

Mictrack Communication Protocol (For B50)



Contents

1. Communication Mode.....	1
1.1 Data type definition.....	1
1.2 Transfer Rules.....	1
1.3 Message definition.....	1
1.3.1 Message Structure.....	1
1.3.2 Start flag and end flag:.....	1
1.3.3 Message Header.....	2
1.3.4 Checksum.....	2
2. LTE Data.....	2
2.1 Universal respond of tracker (tracker to server).....	2
2.2 Universal Respond of server(server to tracker).....	2
2.3 Shake hand message of tracker.....	3
2.4 Tracker Login (tracker to server).....	3
2.4.1 Sample data:.....	3
2.4.2 The parameter description.....	3
2.5 Login Respond (Server to tracker).....	4
2.5.1 Sample data.....	4
2.5.2 decode:	4
2.6 Authentication request.....	4
2.7 Position message (Tracker to server).....	5
2.8 Event Message Report (Tracker to Server).....	8
2.9 Time Sync Request (Tracker to Server).....	9
Message ID: 0x0F01.....	9
2.10 Time Sync Response (Server to Tracker).....	9
2.11 Text message down (server to tracker).....	10
2.12 Text message up (tracker to server).....	10

1. Communication Mode

The protocol support TCP. the platform as a server, the terminal as a client.

1.1 Data type definition

Data Type	Description
BYTE	unsigned char (byte, 8 bit)
WORD	unsigned short (2byte, 16bit)
DWORD	unsigned int (4byte, 32bit)
BYTE[n]	N byte
BCD[n]	8421 code, n byte
STRING	end by 0

1.2 Transfer Rules

The protocol use big-endian network byte order to transfer the words and double words.

The agreement is as follows:

- Byte(BYTE) transmission convention: According to the byte stream transmission;
- Word(WORD) transmission agreement: the first transmission the eight high, and then lower eight;
- Double Word(DWORD) transmission agreement: the first transmission high 24 bit, then high 16 bits, then high 8 bits and the last low eight.

1.3 Message definition

1.3.1 Message Structure

Each message include start flag,header,message body,checksum and end flag.

Start flag	Message header	Message body	Checksum	End flag
0x7e				0x7e

1.3.2 Start flag and end flag:

Use 0x7e as the flag. if the message header, message body and checksum appears 0x7e, have to escape processing as following:

- 0x7e <----> 0x7d followed by a 0x02;
- 0x7d <----> 0x7d followed by a 0x01.

The escaping process is as follows:

Sending message: Message Encapsulation->Compute and Fill Checksum->Escape;

Receiving message: Escaping restore->Verify checksum->Parse message.

Example:

Send a data package 0x30 0x7e 0x08 0x7d 0x55, then encapsulated as follows
0x7e 0x30 7d 0x02 0x08 0x7d 0x01 0x55 0x7e.

1.3.3 Message Header

Item	Type	Length	Description
Message ID	WORD	2	
Message length	WORD	2	Message length
Device ID	BCD[6]	6	Device ID
Serial number	WORD	2	Start for 0 , ++

1.3.4 Checksum

remove field [start flag] [checksum] [end flag]

```
//length is not include field [start flag] [checksum] [end flag]
```

```
char cSum = buff[1];
```

```
for(i=2; i<length;i++)
```

```
{
```

```
    cSum ^= buff [i];
```

2. LTE Data

The following functional classification of protocol description. No special instructions, the default TCP communication.

2.1 Universal respond of tracker (tracker to server)

MessageID: 0x0001

Item	Type	Length	Description
Serial number	Word	2	It is same with request
Respond ID	Word	2	It is same with the request
Result	Byte	1	0: success 1:failed 2:message error 3:Unknown message

2.2 Universal Respond of server(server to tracker)

MessageID: 0x8001

Item	Type	Length	Description
Serial number	Word	2	It is same with request
Respond ID	Word	2	It is same with the request

Result	byte	1	0: success 1:failed 2:message error 3:Unknown message 4.alarm confirm
--------	------	---	---

2.3 Shake hand message of tracker

Message ID : 0x0002, message body is NULL

2.4 Tracker Login (tracker to server)

Message ID: 0x0100

Item	Type	Length	Description
Province ID	Word	2	
City ID	word	2	
Factory ID	Byte	5	
Terminal ID	Byte	20	
Terminal string	Byte	7	
Color of car	byte	1	
Car number	string		

2.4.1 Sample data:

```
7e 01 00 00 2d 01 34 23 39 23 30 00 06 00 2c 00 64 31 31 31
31 00 4e 52 30 39 42 30 30 30 30 31 00 00 00 00 00 00 00 00
00 00 39 32 35 35 30 37 32 01 d4 c1 41 35 35 30 37 32 23 7e
```

2.4.2 The parameter description

Item	Description
Start flag	7e
Command ID	01 00
Length	00 2d
Mobile number	01 34 24 39 23 30
Serial number	00 06
Country code	00 2c
City code	00 86
ID	31 31 31 31 00
Device type	4e 52 30 39 42 30 30 30 30 31 00 00 00 00 00 00 00 00 00 00
ID	39 32 35 35 30 37 32

Color	01
Car Number	d4 c1 41 35 35 30 37 32
Checksum	27
End flag	7e

2.5 Login Respond (Server to tracker)

MessageID: 0x8100

Item	Type	Length	Description
Serial number	Word	2	It is same with request
Respond ID	Word	2	It is same with the request
Result	Byte	1	0: success 1: the ID has been login 2: no car in the system
Authentication code	String		it is only valid while the result is0, otherwise is NULL

2.5.1 Sample data

7e 81 00 00 13 01 34 24 30 23 30 00 00 00 02 00 54 52 32 30
31 37 31 30 31 38 30 39 30 30 31 37 ec 7e

2.5.2 decode:

7e
81 00
00 13
01 34 23 39 23 30
00 00
00 02
00
54 52 32 30 31 37 31 30 31 38 30 39 30 30 31 37 Auth code: TR20171018090017
ec
7e

2.6 Authentication request

Message ID: 0x0102

Item	Type	Length	Description
Authentication code	string	Server send to tracker	Device reconnect to server, will send the authentication code

2.7 Position message (Tracker to server)

Message ID: 0x0200

Item	Type	Length	Description
Alarm flag	DWORD	4	Table1
Status	DWORD	4	Table2
Latitude	DWORD	4	degree *10 ⁶ , for example:113.9452 it will be 113945200
Longitude	DWORD	4	Degree*10 ⁶ , same with latitude
Altitude	Word	2	Unit :m
Speed	Word	2	Unit :0.1KM
Direction	Word	2	0~359
Date&time	BCD[6]	6	YYMMDDhhmmss

Table1: Alarm flag define

Bit	Definition	Action
0	Reserved	
1	Over speed alarm	Default 120km/h
2	Tired alarm	
3-6	Reserved	
7	Low power	
8	Remove alarm	
9~14	Reserve	
15	Motion alarm	
16~28	Reserved	
29	Crash alarm	
30-31	Reserved	

Table2: status define

Bit	Definition
0	0:ACC on 1 :ACC off
1	0:GPS not fix, 1:GPS fix
2	0:North Latitude 1:South Latitude
3	0: east longitude 1: west longitude
4~31	Reserve

Extend data:

Extend data will add the end of position data, Variable length, it is include 3 part

Extend data ID	BYTE	1~255
Length	BYTE	
Extend data		Table3:extend data define

Table3 : extend data

Extend data ID	Length	Description
0x01	4	Mileage on car meter 1/10KM
0x02	2	Fuel : 1/10 L
0x03-0x24		Reserved
0x25	4	Extend status
0x2A	2	IO status
0x2B	4	Reserved
0x30	1	Network signal
0x31	1	GNSS satellite quantity
0xE3	6	Data=[0x01,0x02,0x03,0x04,0x05,0x06] Battery Voltage = 0x0304 (0.001V)
0XF3	N	OBD Data,refer to Table4

Table4 : OBD data

Extend data ID	Length	Description
0x0002	2	Speed 0.1km/h
0x0003	2	Engine speed: 0– 3000RPM
0x0004	2	Battery voltage: 0.001V
0x0005	4	Total mileage: 0.1km
0x0006	2	Instantaneous fuel consumption (idling): 0.1L/h
0x0007	2	Instantaneous fuel consumption (driving): 0.1L/100km
0x0008	1	Engine load: 0 – 100 %
0x0009	2	Coolant temperature: -40 – 215 °C

		(signed integer-40)
0x000B	2	MAP: 0 – 500 kPa
0x000C	2	MAT: -40 – 215 °C (signed integer-40)
0x000D	2	Intake flow rate: 0 – 655.35 g/s. (%100)
0x000E	1	Throttle position: 0 – 100 % (DATA * 100 / 255)
0x000f	1	Ignition timing :(DATA * 0.5)-64 degree
0x0050	17	Vehicle VIN
0x0051	n	Fault code
0x0052	4	Trip ID
0x0100	2	Trip mileage (from start engine to stop engine): 0.1km
0x0101	4	Total mileage: 0.1km
0x0102	2	Trip fuel consumption: 0.1L
0x0103	4	Total fuel consumption: 0.1L
0x0104	2	Trip average fuel consumption: 0.1L/100km
0x010c	2	Trip average speed: 0.1km/h
0x010d	2	Trip maximum speed: 0.1km/h
0x010e	2	Trip maximum engine speed: RPM
0x010f	2	Trip maximum coolant temperature, °C , signed integer
0x0110	2	Trip maximum voltage, 0.001V
0x0112	2	Trip rapid accelerate times
0x0113	2	Trip rapid decelerate times
0x0116	2	Trip rapid stop times

Sample data with extend code:

```
7E 0200 00ED 022102106447 0007 00000000 00040003 026174D5 06F2807C 0014
0000 0000 191218151124 0104 00000000 0202 0000 2504 00000000 2A02 0000
2B04 00000000 300115 310106 E306000005780000 F3A7 0002020000 00030203B2
```

```
00040236B0 00050400000000 0006020000 0007020000 00080115 0009020051
000B020000 000C020036 000D020101 000E0138 000F0193 0100020000
01010400000000 0102020000 01030400000000 01040203E7 010D020050
010E020550 010F020051 01100236B0 0112020000 0113020000 0116020000
0050114C56474250383745374847303331343839 005100 00520400000000
010C020000 00530106 11 7E
```

2.8 Event Message Report (Tracker to Server)

Message ID 0x0900

Item	Type	Length	Description
Flag	String	3	F80008 (default)
Event Message Length	WORD	2	
Event Message Content			Table5: Event message content

Table5: Event message

Item	Type	Description
Event type	1 BYTE	0x01 start of trip; 0x02 end of trip
TIME	BCD[6]	YY-MM-DD-hh-mm-ss(GMT+8)
Trip Number	DWORD	Trip unique serial number
Latitude	DWORD	0.000001 degree
Longitude	DWORD	0.000001 degree
Trip Mileage	DWORD	Mileage of this trip(only show in event type 0x02)
Total Mileage	DWORD	Total mileage, 0.01km

Sample Data of event message

```
1.Start of trip
7E0900001C0251942015020270F8000800170120032015480900000056015F6AF406C
687200000129C597E
7E //start flag
0900 //Message ID
001C //length
025194201502 //device ID
0270 //serial number
F80008 //event flag
0017 //event message length
```

```

01 //event type 0x01=start of trip
200320154809 //time 2020-03-20 15:48:09
00000056 //Trip serial number (It should be the same at the start and end of the trip)
015F6AF4 //latitude 23.030516 degree
06C68720 //longitude 113.674016 degree
0000129C //total mileage at the begin of the trip
59 //checksum
7E //end flag
    
```

2.End of trip

```

7E090000200251942015020300F80008001B0220032016120500000056015F6D7D020
6C68710000000010000129EF07E
    
```

```

7E //start flag
0900 //message ID
0020 //length
025194201502 //device ID
0300 //serial number
F80008 //event flag
001B //event message length
02 //event type 0x02=end of trip
200320161205 //time 2020-03-20 16:12:05
00000056 // Trip serial number (It should be the same at the start and end of the trip)
015F6D7D02 //latitude (7D02=7E)
06C68710 //longitude
00000001 //mileage of this trip
0000129E //total mileage at the end of the trip
F0 //checksum
7E //end flag
    
```

2.9 Time Sync Request (Tracker to Server)

Message ID: 0x0F01

2.10 Time Sync Response (Server to Tracker)

Message ID 0x8F01

Item	Type	Length	Description
result	byte	1	0: none 1: time sync success 2: time sync failed
Date&Time	BCD[6]		YYMMDDhhmmss

2.11 Text message down (server to tracker)

Message ID 0x8300

Item	Type	Length	Description
Flag	BYTE	1	
Message Content	String		1024 byte maximum, GBK(ASCII)

2.12 Text message up (tracker to server)

Message ID 0x0300

Item	Type	Length	Description
Message Content	String	1	1024 byte maximum, GBK(ASCII)

mictrack

Shenzhen Mictrack Electronics Co.,Ltd.

Add: B305-306, Kangsheng Electronic Industrial Park, Zhonghua Rd, Longhua District, Shenzhen, China 518131

Tel: +86-755-21014699

Web: www.mictrack.com

Email: info@mictrack.com