ROPE GPS TRACKER PROTOCOL

I. EXPLAIN THE CONCEPT OF GLOSSARY

Supported Model

M528S, M508S, M508A, M508G

Packet Head and Packet End

The packet head means the beginning of the package.
The packet end means the end of the package.

Packet Length

Packet length means the length of the data content,It is from the next byte of the packet length until the packet end.

Calibration（XOR）

From the packet head of the data stream XOR with the next byte,until the previous byte of the calibration byte

Pseudo IP Address

Using special algorithm change the ID to Pseudo IP Address,the algorithm in Appendix.

II、SERVER TO TRACKER

Locate Immediately--30

[Function] force the tracker to upload current location immediately
[Direction] from server to tracker
[Format]

24	24	30	00	06						0D
Packet head		Main order	Packet length		Pseudo IP Address				calibration	Packet end

Check Tracker Parameters/Status--31

[Function] Check Tracker Parameters/Status
[Direction] from server to tracker
[Format]

24	24	31	00	06						0D
Packet head		Main order	Packet length		Pseudo IP Address				calibration	Packet end

Location Report Interval Setting--34

[Function] Set the timing interval of the location data from tracker.
[Direction] from server to tracker
[Format]

24	24	34	00	06											0D
Packet head		Main order	Packet length		Pseudo IP Address				Parameter 1		Parameter 2		Parameter 3	calibration	Packet end

Parameter1:double byte,hexadecimal form,means the ACC on time horizon(form 1 second to 18.2 hours).
Parameter2:double byte,hexadecimal form,means the ACC off time horizon(from 1 second to 18.2 hours).
Parameter3:single byte(choosing)when close the heartbeat functio,this instruction doesn't include the parameter 3 the packet length is 0x0A,the heartbeat time is 31 seconds or 61 seconds or 91 seconds.

Suspend Current Alarm--37

[Function] Suspend the tracker to keep uploading the alarm message
[Direction] from server to tracker
[Format]

24	24	37	00	06						0D
Packet head		Main order	Packet length		Pseudo IP Address				calibration	Packet end

Enable Engine--38

[Function] Enable engine
[Direction] from server to tracker
[Format]

24	24	38	00	06						0D
Packet head		Main order	Packet length		Pseudo IP Address				calibration	Packet end

Stop Engine--39

[Function] Stop engine
[Direction] from server to tracker
[Format]

24	24	39	00	07						0D
Packet head		Main order	Packet length		Pseudo IP Address				calibration	Packet end

Send Setting Command—3A

[Function] Send setting command
[Direction] from server to tracker
[Format]

24	24	3A	00	N					N		0D
Packet head		Main order	Packet length		Pseudo IP Address				Setting Command	calibration	Packet end

The setting command is ASCII encoding value.

Telephone monitor--3E

[Function] telephone monitor
[Direction]from server to tracker
[Format]

24	24	3E	00	6+X					X		0D
Packet head		Main order	Packet length		Pseudo IP Address				Return number	calibration	Packet end

return number：compressed BCD code，the odd number makes up F,the even number makes up FF.
Example：13812345678 is 13H 81H 23H 45H 67H 8FH　62664307 is 62H 66H 43H 07H FFH
return number: not fixed-length.
introduction: After on-hook, cancels the monitor automatically.

Set Speeding Alarm Threshold--3F

[Function] Set Speeding Alarm Threshold
[Direction] from server to tracker
[Format]

24	24	3F	00	7							0D
Packet head		Main order	Packet length		Pseudo IP Address				Setting value	calibration	Packet end

Setting value　is the upper limit threshold value of the overs peed alarm,when the setting value is 0,the over speed alarm function is closed（form 0 to 255 km/hour）single byte, hexadecimal form.

Camera Controlling--55

[Function] Set/Control camera
[Direction] From server to tracker
[Format]

24	24	55	00	0A										0D
Packet head		Main order	Packet length		Pseudo IP Address				Command Type	Param 1	Param 2	Camera ID	calibration	Packet end

[Explanation]

Command Type	Param1	Param2	Camera ID	Function
01H	00H=Take photo immediately	Picture resolution: Value=1,2,3,4 1=best resolution 4=worst resolution.	1,2	Take Photo

Start Remotely Firmware Update--62

[Function] Inform tracker to update firmware remotely
[Direction] from server to tracker
[Format]

24	24	62	00	06						0D
Packet head		Main order	Packer length		Pseudo IP Address				calibration	Packet end

Tracker will reply with 87 command and sub command 01 after ready to do update.

Send Firmware Package to Tracker--63

[Function] Send firmware package to tracker
[Direction] from server to tracker
[Format]

24	24	63	00	N							256 bytes		0D
Packet head		Main order	Packer length		Pseudo IP Address				Package Index		Package Data	calibration	Packet end

The package index should start from 1.

Set Alert Phone Number--66

[Function] set the terminal's service mobile number
[Direction] from server to tracker
[Format]

24	24	66	00	19					0D		0D
Packet head		Main order	Packer length		Pseudo IP Address				Data packet	calibration	Packet end

Data packet format:"< mobile number>" mobile: 11 byte,ASICC code beginning is '<' symbole,the end is '>' symbol.

Reset the mileage—6B

[Function] reset the mileage
[Direction]from server to tracker
[Format]

24	24	6B	00	06							XOR	0D
Packet head		Main order	Packet length		Pseudo IP Address				parameter		calibration	Packet end

[Explanation]
parameter：the value of the mileage(unit:km)

Set Tracker Time Zone—6C

[Function] change time zone
[Direction] from server to tracker
[Format]

24	24	6C	00	08							XOR	0D
Packet head		Main order	Packet length		Pseudo IP Address				parameter		calibration	Packet end

[Explanation]
Parameters:
Issue th differential value based on GMT, (unit: hours:minutes,)
remarks:
minute follows hour, the high order of hour is to decide whether plus or minus, if the high order is O, it means the timezone should be adjust as GMT+time parameters
e.g.:
the 8 area east: parameters is 0x08 0x1E, current time= GMT+8hours30minutes

the 11 area west: parameters is 0x8b 0x30, current time=GMT-11hours 48minutes

Set iButton Permission--94

[Function] Set permission for swiping ibutton
[Direction] From server to tracker
[Format]

24	24	94	00	N+7						N	XOR	0D
header		Main command	length		Pseudo IP				Sub Command	content	calibration	Packet end

Sub Command=6: Set tracker to anti theft mode
Sub Command=7: Cancel anti theft mode
Sub Command=10: Set ibutton ID which will be allow to start the car after swipe, can only set maximum 3 ID, each ibutton ID must be 12 bytes, for example:

If driver Jack's ibuttton ID is 13082D45, pls mind the driver name that you want to set should be no longer than 10 bytes, then if you set only Jack can start the car after swipe the ibutton, the content should be:

<Jack,000013082D45><><>

If you want to clear the setting, the content should be:

<><><>

Sub Command=11: Get ibutton ID setting and whether tracker is in anti theft mode

III、KEEP CONNECTION ALIVE

Heartbeat Data to Tracker--21

[Function] The center send the “link confirm instruction” from server to tracker when received the linkage information of the vehicle。

[Direction] from server to tracker

[Format]

24	24	21	00	05					0D
Packet head		Main order	Packet length		Calibration value	Main order	Slave order	calibration	Packet end

[Explanation]

Calibration value: It is the value of the return data of the vehicle.(1 byte)

Main order:It is the main order of the return data of the vehicle (1 byte);

Slave order:It is the slave order of the return data of the vehicle (1 byte);

IV TRACKER TO SERVER

Heart Beat Data To Server--21

[Function] The terminals send connecting information From tracker to server.

[Direction] From tracker to server

[Format]

24	24	21	00	06						0D
Packet head		Main order	Packet length		Pseudo IP Address				calibration	Packet end

Camera Reply Command--56

[Function] reply to command 55 from server

[Direction] from tracker to server

[Explanation]

24	24	56	00	L					N bytes		0D
Packet head		Main order	Packet length		Pseudo IP Address				Reply content	calibration	Packet end

Reply Content Explanation:

Reply content	Function
01H+Total package quantity	Inform the server that the camera have taken photo and total picture data package quantity
07H	Have upload all picture data package
09H	Reset camera

Picture data Package--57

[Function] Upload picture data to server
[Direction] from tracker to server
[Format]

24	24	57	00	519						512 bytes		0D
Packet head		Main order	Packet length		Pseudo IP Addresses				Package Index	Picture package data	calibration	Packet end

[Explanation]

Package Index: start from zero.
Picture data: image data

RFID Card Data--72

[Function] upload RFID card swiping data to server
[Direction] from tracker to server
[Format]

24	24	72	00	7+N						N		0D
Packet head		Main order	Packet length		Pseudo IP Address				02H	RFID data	calibration	Packet end

RFID Data Example：

24 24 72 00 22 0C A2 32 83 02 30 30 30 35 33 39 34 34 36 37 0D 2A 41 11 04 07 11 58 52 02 23 41 79 11 35 26 13 24 0D

Parsing result:

24 24 Packet head
72 Main order
00 22 Packet length
0C A2 32 83 Pseudo IP Addressess
02 sub signal, means it is RFID data
30 30 30 35 33 39 34 34 36 37 ASCII code RFID card ID
0D 2A 0x0D mark the end of the RFID card ID, 0x2A is reserved byte
41 0x41 means the tracker is located by GPS at current
11 04 07 11 58 52 Date&Time
02 23 41 79 Latitude
11 35 26 13 Longitude
24 calibration
0D Packet end

Position data--80

[Function] The terminals send the position data From tracker to server.
[Direction] From tracker to server
[Format]

24	24	80	00	23					N Byte		0D
Packet head		Main order	Packet length		Pseudo IP Address				Position data & Extend Data	calibration	Packet end

Position Data Format

6 bytes	4 bytes	4 bytes	2 bytes	2 bytes	1 byte	1 byte	1 byte	4 bytes	4 bytes
Date Time	latitude	longitude	speed	angle	GPS Status	Digital Input	Ignition Status	Analog Input	Mileage

Date&time format：

Date range：
Year range: 00~99, month range: 1~12, day range:1~31
Hour range: 00~23, minute range: 00~59, second range: 00~59
Example:2009-10-25 15:10:03
0x09 0x10 0x25 0x15 0x10 0x03

Latitude format：
latitude range：00.000 minute 0 degree——59.999 minutes 89 degrees ；
Data form is BCD form,but the highest bit is sign bit。“positive”means“north latitude”，“negative”means“south latitude”；
[example]： Latitude 37.901 Minutes 30 Degrees south：
3H，03H，79H，01H

Longitude：
longitude form：00.000 minute 0 degree——59.999 minutes 179 degrees；
Data form is BCD form,but the highest bit is sign bit。“positive”means“east longitude”，“negative”means“west longitude”；
[example]：Longitude 45.608 Minutes 130 Degrees west
13H，04H，56H，08H

Speed：
speed range：0——9999km/hour
data form is BCD form。
[example]： 120km/hour：
01H，20H

Angle：
rang：000——359 degree
data form is BCD form, Due north is 0 degrees, clockwise counts。
unit：degree。
[example]： 154 degree：
01H，54H

GPS status：
position status is a single byte：

D7	D6	D5	D4	D3	D2	D1	D0

D7	**Mark flag**
0	delocalization
1	position

D6	**Differential positioning**
0	no
1	yes

D5	**D4**	**introduction**

X	X	reserved

D3	D2	D1	D0	satellite
0	0	0	0	range：0—15
1	1	1	1	

Digital Input Format：

D7	
D6	1：default system use
D5	Digital input 1（1：full 0：no）
D4	Antenna short(0,normal,<> 0 short)
D3	Antenna open(0,normal,<> 0 open)
D2	Digital input 2（0：open 1：close）
D1	Digital input 3（0：open 1：close）
D0	Digital input 4（0：open 1：close）

Ignition Status:

1：ignition ON

0：ignition OFF

Analog input:

The first 2 bytes are oil resistance,the latter two are voltage value.

example：50 ohm is 01F4

example:0000H is 0 V; 0B3AH is 11.58 V

Voltage conversion step:

1. The 16 bits of the first two byte were changed to the interger of the voltage.
2. The last two bytes were change to decimal.

mileage：

The current mileage of the vehicle, unit is meter

Extend Data Format

2 byte	2 byte	1 byte	1 byte	N byte
Temperature 1	Reserve bytes	Harsh alarm byte	Sub Signal	Sub Data

2 byte	1 byte	1 byte	1 byte	1 byte	N byte
Temperature 1	Rotation Sensor Status	Reserve byte	Harsh alarm byte	Sub Signal	Sub Data

Temperature 1：

Temperature Sign:

0: Temperatures of zero degrees and above

1: Sub-zero temperature

Temperature Sensor1 Value:

0xFF means there is no temperature sensor detected

Example:

0x00 0x1E= 30℃

0x01 0x1E= -30℃

Rotation Sensor Status:

0x00= Stopped

0x01= Stirring

0x02= Unloading

Harsh Alarm Byte:

0x01 = harsh acceleration alarm

0x02= harsh braking alarm

0x04= harsh cornering alarm

0x10= Collision Alarm

0x20= Rollover alarm

Sub Signal:

0x03: sub data is the data of temperature sensor 2,3,4 and weight sensor value, total length of sub data is 8 bytes.

sub data format for sub signal 0x03:

Temperature Sign	Temperature 2	Temperature Sign	Temperature 3	Temperature Sign	Temperature 4	2 bytes
Temperature Sensor 2		Temperature Sensor 3		Temperature Sensor 4		Weight Sensor Value

Temperature data from Sensor 2,3,4 is the same format as temperature 1

Weight Sensor value is hexadecimal data which should be converted to be decimal data.

0x06: sub data is the data of temperature and Humidity from temperature and humidity integrated sensor, the total length of sub data is 6 bytes.

sub data format for sub signal 0x06:

Package length	Temperature	Temperature Integer part	Temperature decimal part	Humidity Integer part	Humidity decimal part
0x05	0:positive 1:negative				

Reply to Command 30--81

[Function] Reply to command 30

[Direction] From tracker to server

[Format]

24	24	81	00	23					（24Byte）		0D
Packet head		Main order	Packet length		Pseudo IP Address				Position data	calibration	Packet end

Alarm Data--82

[Function] Terminal send the alarm data From tracker to server

[Direction] From tracker to server

[Format]

24	24	82	00	25					（24Byte）				0D
Packet head		Main order	Packet length		Pseudo IP Address				Position data	Alarm data		calibration	Packet end

[Explanation]

Alarm data(double Hexadecimal byte)

The first byte

D7	D6	D5	D4	D3	D2	D1	D0

D7	
D6	oil change alarm
D5	Cross the border alarm
D4	Over voltage alarm
D3	Under voltage alarm
D2	The people of overload alarm
D1	Overtime driving alarm
D0	Enter in to the border alarm

The second alarm data

D7	D6	D5	D4	D3	D2	D1	D0

D7	The alarm of the door was opened illegally
D6	Start the vehicle illegally
D5	Vibration alarm
D4	Center enable the terminal to alarm
D3	Power failure alarm
D2	Parking alarm
D1	Over speed alarm

D0	Emergency alarm

1:alarm
0:no alarm

Tracker Parameters and Status--83

[Function] Tracker parameters and status, reply to command 31
[Direction] From tracker to server
[Format]

24	24	83	00	1E					24Byte		0D
Packet head		Main order	Packet length		Pseudo IP Address				Stauts data	calibration	Packet end

Format of Status data:

6 bytes									
sampling time	Alarm status	Whether to locate	Sampling type	Sampling value		Sending type	Car stop setting	Over speed setting

Phone limit	Area node limit	Safe setting	Long time driving	AA		BB	CC	none

AA:sampling value(acc off)
BB:emergency alarm switch
CC:The related of photograph

name	length	description
Sampling time	6	BCD code
Alarm stauts	2	Double Hexadecimal byte
Whether to locate	1	1 yes 0 no

Sampling type	1	1 fixed time 0 fixed distance
Sampling value	2	Double Hexadecimal byte
Sending type	1	1 point send 2 silence
Car stop setting	1	0 is no setting the others is the correct value.single byte(Hexadecimal)
Overspeed setting	1	0 is no setting ther others is the correct value.single byte(Hexadecimal)
Phone limit	1	1 group phone limit 0 no limited
Area node limit	1	1 limit 0 no limit
Safe setting	1	
long time driving(two bytes)	2	Long time driving
Sampling value(acc off)	2	Double byte(Hexadecimal)
Emergency alarm switch	1	1opened，0 closed
The related of photograph	1	
reserved	1	reserved

Reply for Setting Command—84

[Function] Reply to setting command 3A
[Direction] from server to tracker
[Format]

24	24	84	00	N					N		0D
Packet head		Main order	Packet length		Pseudo IP Address				Reply for Setting Command	calibration	Packet end

The setting command is ASCII encoding value.

Tracker Answer Data--85

[Function] The terminal send the correct data when receive the center's instruction
[Direction] From tracker to server

[Format]

24	24	85	00	0B					（5Byte）		0D
Packet head		Main order	Packet length		Pseudo IP Address				Answer data	calibration	Packet end

[Explanation]

- The format of return data

Main order	Slave order	Success or failure	reserved	

- Main order (single byte)(Hexadecimal)
- Slave order (single byte)(Hexadecimal)
- Success or failure BCD code 1 success 0 failure

Tracker Firmware Update Command--87

Firmware Package Download Requisition command 87--01

[Function] Download firmware file package from server, the tracker will keep requesting the package until finish.

[Direction] From tracker to server

[Format]

24	24	87	00	09					01				0D
Packet head		Main order	Package Length		Pseudo IP Address				Sub Command	Package Index		Calibration	Packet end

If the package index is 300, hexadecimal format is 01 2C, the package index data in command should be 2C 01.

Firmware Update Success 87--02

[Function] The tracker upload this command to server after the firmware update success

[Direction] From tracker to server

[Format]

24	24	87	00	09					02		0D

Packet head	Main order	Package Length	Pseudo IP Address	Sub Command	Calibration	Packet end

Firmware Update Failed 87--03

[Function] The tracker upload this command to server after the firmware update failed
[Direction] From tracker to server
[Format]

24	24	87	00	09					03		0D
Packet head		Main order	Package Length		Pseudo IP Address				Sub Command	Calibration	Packet end

The Blind Area data of GPRS--8E

[Function] When the GPRS signal is not connected,the position data was stored,when the GPRS signal is available, the terminal send the stored data to the center.
[Format]

24	24	8E	00	23					（24Byte）		0D
Packet head		Main order	Packet length		Pseudo IP Address				Position data	calibration	Packet end

The 42nd byte of command 8E indicate the alarm as bellow:
0x01=harsh acceleration alarm
0x02=harsh braking alarm
0x04=harsh cornering alarm
0x10=crashing alarm
0x20=roll over alarm
0x4=towed away alarm
0x80=SOS

iButton Tag Swiping Data/Reply for iButton Setting--94

[Function] Reply the command 94 from server

[Direction] From tracker to server

[Format]

24	24	94	00	N+7						N	XOR	0D
Packet head		Main command	length		Pseudo IP				Sub Command	content	calibration	Packet end

Sub Command=6: Set tracker to anti theft mode successfully

Sub Command=7: Set tracker to anti theft mode failed

Sub Command=8： Cancel anti theft mode successfully

Sub Command=10：

Content format: Status+<Jack,000013082D45>

Status=1 means tracker is on anti theft mode

Status=0 means tracker is not on anti theft mode

Status=2 reserved status

Jack is driver name, 000013082D45 is ibutton ID

Sub Command=11: ibutton swiping data

Content format: Status+DateTime+Coordinates+<Jack,000013082D45>

Status=1 means tracker is on anti theft mode

Status=0 means tracker is not on anti theft mode

Status=2 reserved status

DateTime:

If data is 0x11 0x04 0x07 0x11 0x58 0x52, then Date and time should be 2011-04-07 11:58:52

Coordinates:

Locate flag	Latitude				Longitude			

Locate flag: 1 GPS located，0 No GPS signal

Latitude and Longitude is the same format as command 80

Sub Command=12: none permitted ibutton swiping data

Content format：DateTime+Coordinates+ibutton ID

DateTime:

If data is 0x11 0x04 0x07 0x11 0x58 0x52, then Date and time should be 2011-04-07 11:58:52

Coordinates:

Locate flag	Latitude				Longitude			

Locate flag: 1 GPS located，0 No GPS signal

Latitude and Longitude is the same format as command 80

Appendix

How to change the SIM card number to Pseudo IP Address Method：

1. Cut off the first bit of the SIM card number
   Example 13512345006 ， cut off the first bit is 3512345006
2. Get 5 froup number from the 10 number
   Example: get 35 12 34 50 06 from 3512345006
3. Take the latter four numbers of this group,and change them in to binary form
   Example:
   a) The latter four numbers of group of 35 12 34 50 06 is 12 34 50 06
   b) The binary of 12 is 00001100
   c) The binary of 34 is 00100010
   d) The binary of 50 is 00110010
   e) The binary of 06 is 00000110
   f) So the group number is 01100 00100010 00110010 00000110
4. Get the first froup number in second step and subtract 30,then change it into binary,at last add it in to the four number separately which from the third step
   a) The first number of 35 12 34 50 06 is 35
   b) 35 – 30 = 5
   c) The binary of 5 is 0101
   d) Use the first number of third step,then adds 0 on the top digit(0 is the first bit of the 5 which is binary),at last get 00001100, change 00001100 to decimal is 12.
   e) Use the second number of third step,then adds 1 on the top digit(1 is the second bit of the 5 which is binary),at last get 10100010, change 10100010 to decimal is 162.
   f) Use the third number of third step,then adds 0 on the top digit(0 is the third bit of the 5 which is binary),at last get 00110010, change 00110010 to decimal is 50
   g) Use the fourth number of third step,then adds 1 on the top digit(1 is the fourth bit of the 5 which is binary),at last get 10000110, change 10000110 to decimal is 134
5. Get the Pseudo IP Address
   The Pseudo IP Address of 13512345006 is **12.162.50.134**

Here I give you a example to tell you how to reply the command data from tracker, this is the data that you received:
24 24 80 00 25 37 E3 00 A0 13 01 18 07 28 35 02 41 71 15 05 43 86 88 00 00 00 00 C0 47 00 07 A8 19 28 01 16 A0 1D 00 FF E2 0D

Header: 24 24
Main order: 80
Length: 00 25
Pseudo IP Addressess: 37 E3 00 A0
Date: 13 01 18
Time: 07 28 35
Latitude: 02 41 71 15
Longitude: 05 43 86 88
Speed: 00 00
Angle: 00 00
GPS status: C0
Detection setting status: 47
Ignition status: 00
Oil resistance: 07 A8
Voltage: 19 28
Mileage(meter): 01 16 A0 1D
Temperature: 00 FF
Calibration: E2
Footer: 0D

Software should reply:
24 24 21 00 05 E2 80 00 46 0D
Header: 24 24
Main order: 21
Length: 00 05
Calibration of the receiving data: E2
Main order of the receiving data: 80
Slave order of the receiving data (if no such byte in receiving data, please use 0x00): 00
Calibration: 46
Footer: 0D

Here I give you some function in C# to help you parse the data:

```
/// <summary>
/// Convert Pseudo IP Addressess to System No
/// </summary>
```

```
    /// <param name="ip"> Pseudo IP Addressess, if you receive 0x37, 0xE3, 0x00, 0x A0, then please give
param value55.227.0.160</param>
    /// <returns>System No</returns>
    public string IPToNumber(string ip)
    {
        long iHight = 0;
        string sTemp = null;
        try
        {
            string[] sIp = ip.Split(new char[] { '.' }, 4);
            byte[] bSimNo = new byte[sIp.Length];
            for (int i = 0; i < bSimNo.Length; i++)
            {
                bSimNo[i] = Convert.ToByte(sIp[i]);
            }
            if ((bSimNo[0] & 0x80) != 0)
            {
                iHight = iHight + 8;
            }
            bSimNo[0] = Convert.ToByte(bSimNo[0] & 0x7F);
            if ((bSimNo[1] & 0x80) != 0)
            {
                iHight = iHight + 4;
            }
            bSimNo[1] = Convert.ToByte(bSimNo[1] & 0x7F);
            if ((bSimNo[2] & 0x80) != 0)
            {
                iHight = iHight + 2;
            }
            bSimNo[2] = Convert.ToByte(bSimNo[2] & 0x7F);
            if ((bSimNo[3] & 0x80) != 0)
            {
                iHight = iHight + 1;
            }
            bSimNo[3] = Convert.ToByte(bSimNo[3] & 0x7F);
            sTemp = "1" + Convert.ToString(30 + iHight) + bSimNo[0].ToString("00")
                + bSimNo[1].ToString("00") + bSimNo[2].ToString("00") + bSimNo[3].ToString("00");
        }
        catch
        {
            sTemp = null;
        }
        return sTemp;
    }
```

```
/// <summary>
/// Convert System No to Pseudo IP Addressess
/// </summary>
/// <param name="sim">System No, for example: 13555990032</param>
/// <returns> Pseudo IP Addressess</returns>
public string NumToIp(string sim)
{
    string[] Temp = new string[4];
    int iHigt;
    switch (sim.Length)
    {
        case 11:
            Temp[0] = sim.Substring(3, 2);
            Temp[1] = sim.Substring(5, 2);
            Temp[2] = sim.Substring(7, 2);
            Temp[3] = sim.Substring(9, 2);
            iHigt = Convert.ToInt32(sim.Substring(1, 2)) - 30;
            break;
        case 10:
            Temp[0] = sim.Substring(2, 2);
            Temp[1] = sim.Substring(4, 2);
            Temp[2] = sim.Substring(6, 2);
            Temp[3] = sim.Substring(8, 2);
            iHigt = Convert.ToInt32(sim.Substring(0, 2)) - 30;
            break;
        case 9:
            Temp[0] = sim.Substring(1, 2);
            Temp[1] = sim.Substring(3, 2);
            Temp[2] = sim.Substring(5, 2);
            Temp[3] = sim.Substring(7, 2);
            iHigt = Convert.ToInt32(sim.Substring(0, 1));
            break;
        default:
            switch (sim.Length)
            {
                case 8:
                    return NumToIp("140" + sim);
                case 7:
                    return NumToIp("1400" + sim);
                case 6:
                    return NumToIp("14000" + sim);
                case 5:
                    return NumToIp("140000" + sim);
```

```
                    case 4:
                        return NumToIp("1400000" + sim);
                    case 3:
                        return NumToIp("14000000" + sim);
                    case 2:
                        return NumToIp("140000000" + sim);
                    case 1:
                        return NumToIp("1400000000" + sim);
                    default:
                        return "";
                }

        }
        int[] sIp = new int[4];

        if ((iHigt & 0x08) != 0)
            sIp[0] = Convert.ToInt32(Temp[0]) | 128;
        else
            sIp[0] = Convert.ToInt32(Temp[0]);

        if ((iHigt & 0x04) != 0)
            sIp[1] = Convert.ToInt32(Temp[1]) | 128;
        else
            sIp[1] = Convert.ToInt32(Temp[1]);

        if ((iHigt & 0x02) != 0)
            sIp[2] = Convert.ToInt32(Temp[2]) | 128;
        else
            sIp[2] = Convert.ToInt32(Temp[2]);

        if ((iHigt & 0x01) != 0)
            sIp[3] = Convert.ToInt32(Temp[3]) | 128;
        else
            sIp[3] = Convert.ToInt32(Temp[3]);

        return sIp[0] + "." + sIp[1] + "." + sIp[2] + "." + sIp[3];
    }

    /// <summary>
    /// Get calibration value
    /// </summary>
    /// <param name="tmp">the whole package data</param>
    /// <param name="len">the length which will be used to calculate the calibration value </param>
    /// <returns> calibration value </returns>
```

```
public static byte GetCheckXor(byte[] tmp, int len)
{
	byte A = 0;
	for (int i = 0; i < len; i++)
	{
		A ^= tmp[i];
	}
	return A;
}
```