

I. GPRS Data Setup and Operation

It is very important to understand the basic operation of GPRS network and how the data is sent. We suggest you to check with your GSM service provider for the charges and availability of GPRS data transfer. Charges for sending data are different in every country in the world.

For sending data we use TCP/IP protocol. This protocol uses complete handshaking (No data can get lost or corrupted while sending).

It is very cost effective to send data with GPRS as 10,000 bytes of data will only cost about 10 cents, but sending 1 text message (only 160 bytes) will cost about 20 cents (cost depends on country and GSM service provider). GPRS is more than 124 times cheaper than using SMS messages!

i. Before you can use GPRS

Before you can use GPRS you need to find out from your GSM provider the following information:

1. ISP login and password name.
2. APN (Access Point Name)

If GPRS is not available by your GSM provider then enable SMS only.

II. Command Activation Codes and Data Format

When iTrac receives SMS messages it will only accept messages that come from the 1st mobile phone number preset in call parties.

Note: it needs a space between c1 and x; c2 and x; c3 and x; c5 and x

Functions and options that can be set remotely by text messages(SMS)!

iTrac only listens command code that comes from the 1st preset number in Called parties for the following cases and confirm message will be sent back after one below command codes

c1 x → (x=0 Output 1 OFF
 x=1 Output 1 ON)

c2 x → (x=0 Output 2 OFF
 x=1 Output 2 ON)

c3 x → (x=0 Remote Controller Disable
 x=1 Remote Controller Enable)

c5 x → (x=0 Arm OFF
 x=1 Arm ON)

c8 → Reset

a confirm message will be sent back to 1st preset number after one of command codes above send successful to unit.

Ex: c10k

Data Format from Control centre to iTrac !

c1 x

func:Output1 Control

'x': 1=On, 0=Off

Ack:\$c11

c2 x

func:Output2 Control

'x': 1=On, 0=Off

Ack:\$c11

c3 x

func:Remote Controller

'x': 1=Enable, 0=Disable

Ack:\$c11

c4 x

func:Remote Control with Mobile Phone

'x': 1=Enable, 0=Disable

Ack:\$c11

c5 x

func:Arm Setup

'x': 1=Arm, 0=Disarm

Ack:\$c11

c6 x

func:Data send through GPRS

'x': 1=Go, 0=Stop

Ack:\$c11

c7 x

func:Data send through SMS

'x': 1=Go, 0=Stop

Ack:\$c11

c8

func:Reset

Ack:\$c8

c10

func:Get current position

Ack:\$c23

c11

func:Get command execution result

Ack:\$c11

c12

func:Get general setting

Ack:\$c12

c13

func:Get Call Center setting

Ack:\$c13

\$c20,SMSS,SMSTime,GPRSS,GPRSTime,DataTime
,SOSTEL1,SOSTEL2,SOSTEL3,TEL1,TEL2,TEL3
,In1AS,In2AS,VBAS,BBAS,#

func:General setting

SMSS(Use SMS Select): 1=On, 0=Off

SMSTime(SMS time interval for sending data): 0~9999(Sec)

GPRSS(Use GPRS Select): 1=On, 0=Off

GPRSTime(GPRS time interval for sending data): 0~9999(Sec)

DataTime(Time interval for saving data into memory): 0~9999(Sec)

SOSTEL1(Called parties for SOS phone number 1): 20 digits

SOSTEL2(Called parties for SOS phone number 2): 20 digits

SOSTEL3(Called parties for SOS phone number 3): 20 digits

TEL1(Called parties for event phone number 1): 20 digits

TEL2(Called parties for event phone number 2): 20 digits

TEL3(Called parties for event phone number 3): 20 digits

In1AS(Send SMS when Input 1 activated select): 1=On, 0=Off

In2AS(Send SMS when Input 2 activated select): 1=On, 0=Off

VBAS(Send SMS when Vehicle Battery is under low limit select): 1=On, 0=Off

BBAS(Send SMS when Backup Battery is under low limit select): 1=On, 0=Off

\$c21,APNName,APNID,APNPwd,DIPS,IP1,Port1,IP2,Port2,CCPN,#

func:Call Center setting

APNName: 16 digits

APNID: 16 digits

APNPwd(APN Password):16 digits

DIPS(Default IP Address select): 1=Enable, 0=Disable

IP1(First IP): 0~255.0~255.0~255.0~255

Port1(First TCP Port): 0~65535

IP2(Second IP): 0~255.0~255.0~255.0~255

Port2(Second TCP Port): 0~65535

CCPN(Call Center Phone Number): 20 digits

\$c22,time1,time2,#

func:Get record

time1:Data Dump start time(yymmdd)

time2:Data Dump end time(yymmdd)

Ack:\$c22

Data Format from iTrac to Control Centre !

\$c8

func:Restart command receive

\$c11,Out1,Out2,RC,RCM,Arm,DSG,DSS,Out1W,#

func:Command execution result

Out1(Output1): 1=On, 0=Off

Out2(Output2): 1=On, 0=Off
RC(Remote Controller): 1=Enable, 0=Disable
RCM(Remote Control with Mobile Phone): 1=Enable, 0=Disable
Arm: 1=Arm, 0=Disarm
DSG(Data send through GPRS): 1=Go, 0=Stop
DSS(Data send through SMS): 1=Go, 0=Stop
Out1W(Output1 On Wait): 1=Waiting, 0=free

\$c12,SMSS,SMSTime,GPRSS,GPRSTime,DataTime
,SOSTEL1,SOSTEL2,SOSTEL3,TEL1,TEL2,TEL3
,In1AS,In2AS,VBAS,BBAS,#

func:General setting result

SMSS(Use SMS Select): 1=On, 0=Off
SMSTime(SMS time interval for sending data): 0~9999(Sec)
GPRSS(Use GPRS Select): 1=On, 0=Off
GPRSTime(GPRS time interval for sending data): 0~9999(Sec)
DataTime(Time interval for saving data into memory): 0~9999(Sec)
SOSTEL1(Called parties for SOS phone number 1): 20 digits
SOSTEL2(Called parties for SOS phone number 2): 20 digits
SOSTEL3(Called parties for SOS phone number 3): 20 digits
TEL1(Called parties for event phone number 1): 20 digits
TEL2(Called parties for event phone number 2): 20 digits
TEL3(Called parties for event phone number 3): 20 digits
In1AS(Send SMS when Input 1 activated select): 1=On, 0=Off
In2AS(Send SMS when Input 2 activated select): 1=On, 0=Off
VBAS(Send SMS when Vehicle Battery is under low limit select): 1=On, 0=Off
BBAS(Send SMS when Backup Battery is under low limit select): 1=On, 0=Off

\$c13,APNName,APNID,APNPwd,DIPS,IP1,Port1,IP2,Port2,CCPN,#

func: Call Center setting result

APNName: 16 digits
APNID: 16 digits
APNPwd(APN Password):16 digits
DIPS(Default IP Address select): 1=Enable, 0=Disable
IP1(First IP): 0~255.0~255.0~255.0~255
Port1(First TCP Port): 0~65535
IP2(Second IP): 0~255.0~255.0~255.0~255
Port2(Second TCP Port): 0~65535
CCPN(Call Center Phone Number): 20 digits

\$c14,Date,Time,Event,Lat,N/S,Lon,E/W,MSLA,Units,Status,Speed,Course,#

func: GPS Data & event feed back

Date: dd(date)mm(month)yy(year)
Time: hh(hour)mm(minute)ss.sss(second)
Event: 0=Normal
1=S.O.S
2=Input1 activated
3=Input2 activated
4=Vehicle Battery is under 11.3V
5=Vehicle Battery is under 10.4V
6=Vehicle Battery is under 4.2V

7=Backup Battery is under 3.5V
Lat(Latitude): 0000.0000~9000.0000 (ddmm.mmmm)
N/S: N=North, S=South
Lon(Longitude): 00000.0000~18000.0000 (dddmm.mmmm)
E/W: E=East, W=West
MSLA(MSLAltitude): +/-0.0~18000.0
Units: M=Meter
Status: A=Data valid, V=Data invalid
Speed(Speed over ground in knots): 0.0~99999.999
Course(Course over ground in degrees): 0.0~359.99

\$c15

func: Record dump stop receive

\$c23,Date,Time,Event,Lat,N/S,Lon,E/W,MSLA,Units,Status,Speed,Course,#

func:GPS Data & event feed back

Date: dd(date)mm(month)yy(year)

Time: hh(hour)mm(minute)ss.sss(second)

Event: 0=Normal

1=S.O.S

2=Input1 actived

3=Input2 actived

4=Vehicle Battery is under 11.3V

5=Vehicle Battery is under 10.4V

6=Vehicle Battery is under 4.2V

7=Backup Battery is under 3.5V

Lat(Latitude): 0000.0000~9000.0000 (ddmm.mmmm)

N/S: N=North, S=South

Lon(Longitude): 00000.0000~18000.0000 (dddmm.mmmm)

E/W: E=East, W=West

MSLA(MSLAltitude): +/-0.0~18000.0

Units: M=Meter

Status: A=Data valid, V=Data invalid

Speed(Speed over ground in knots): 0.0~99999.999

Course(Course over ground in degrees): 0.0~359.99

How to convert ddmm.mmmm and dddmm.mmm to D.d or DMS

Example: 2500.1472 is in ddmm.mmmm format

To convert to D.d

Divide mm.mmmm by 60 to get .d ($0.1472/60 = 0.0024$)

Add .d to D to get D.d -> 25.0024 (Decimal degrees)

To convert to DMS

Multiply .mmmm by 60 to get S ($0.1472 * 60 = 8$)

25 degrees , 0 minutes and 8 Seconds