

Wialon IPS v.1.1 Communication protocol

IPS communication protocol was developed by Gurtam for personal and vehicle GPS/GLONASS trackers, which send data to satellite monitoring system server over TCP or UDP.

Incoming data (TCP)

All data is sent and received in plain text format over TCP protocol and has the following format:

#TP#msg\r\n

#	Start byte
TP	Type of packet, all possible types are listed in table 1
#	Separator
msg	Message
\r\n	End of message, <CR><LF> symbols (0D0A in HEX)

Packet types

Type	Description	Sender
L	Login	Tracker
AL	Answer to login	Server
D	Data packet	Tracker
AD	Answer to data packet	Server
P	Ping(heartbeat) packet	Tracker
AP	Answer to ping (heartbeat packet)	Server
SD	Short data packet	Tracker
ASD	Answer to short data packet	Server
B	Blackbox packet	Tracker
AB	Answer to blackbox packet	Server
M	Message to driver	Tracker/Server
AM	Reply to message from driver	Server

I	Packet with photo	Tracker
AI	Reply to packet with photo	Server
US	Packet with new firmware	Server
UC	Packet with configuration file	Server

Login packet

#L#imei;password\r\n

imei	Controller unique ID, IMEI, or serial number
;	Separator
password	Password for access to device, if no password, then NA is sent

Server sends answer to login packet, AL:

“1” – authorization successful

“0” – connection rejected by server

“01” – error checking password

Example:

#AL#1\r\n

#AL#0\r\n

Short data packet

#SD#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats\r\n

date	Date in UTC format, DDMMYY
time	Time in UTC format, HHMMSS
lat1;lat2	Latitude (5544.6025;N)
lon1;lon2	Longitude (5544.6025;N)
speed	Speed, integer, km/h, if no data, NA is sent
course	Course, integer, degrees, if no data, NA is sent
height	Height, , integer, m, if no data, NA is sent
sats	Number of satellites, integer, if no data, NA is sent

Server sends ASD packet as an answer to SD packet:

“-1” – packet structure error

“0” – incorrect time

“1” – packet successfully registered

“10” – error getting coordinates

“11” – error getting height, speed or course

“12” – error getting amount of satellites

Example:

#ASD#1\r\n

#ASD#0\r\n

#ASD#10\r\n

#ASD#11\r\n

#ASD#12\r\n

Data packet

#D#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats;hdop;inputs;outputs;adc;
ibutton;params\r\n

date	Date in UTC format, DDMMYY
time	Time in UTC format, HHMMSS
lat1;lat2	Latitude (5544.6025;N)
lon1;lon2	Longitude (5544.6025;N)
speed	Speed, integer, km/h, if no data, NA is sent
course	Course, integer, degrees, if no data, NA is sent
height	Height, , integer, m, if no data, NA is sent
sats	Number of satellites, integer, if no data, NA is sent
hdop	Horizontal Dilution of Precision, double, if no data, NA is sent
inputs	Digital inputs, each bit corresponds to one digital input beginning from the LSB, integer, if no data, NA is sent
outputs	Digital outputs, each bit corresponds to one digital output beginning from the LSB, integer, if no data, NA is sent
adc	Analog inputs, fractional numbers separated by comma, if no data, empty string is send. Input numbering begins from 1 (adc1..adcN). Example 14.77,0.02,3.6
ibutton	Driver key code, custom length string. If no data, NA is sent
params	Set of additional parameters separated by comma. Each parameter has the following format: NAME:TYPE:VALUE NAME – custom string TYPE – parameter type, 1 – int, 2 – double, 3 – string VALUE – parameter value, depends on type To send panic button use parameter with 1 type named “SOS”, 1 mean panic button was pressed. Example: count1:1:564,fuel:2:45.8,hw:3:V4.5

Server sends AD packet as an answer to D packet:

“-1” – packet structure error

“0” – incorrect time

“1” – packet successfully registered

“10” – error getting coordinates

“11” – error getting height, speed or course

“12” – error getting amount of satellites or HDOP

“13” – error getting inputs or outputs

“14” – error getting adc

“15” – error getting additional parameters

Example:

#AD#1\r\n

#AD#0\r\n

#AD#10\r\n

#AD#11\r\n

...

#AD#16\r\n

Ping (heartbeat) packet

#P# \r\n

This packet is used for keeping active TCP-connection to server and checking channel availability.

Server sends AP packet as an answer to P packet:

Example:

#AP#\r\n

Blackbox packet

#B#msg \r\n

Blackbox packet is used for transmission of several messages at one time.

In this case “msg” contains several packets (SD or D), separated by “|” symbol.

Server sends AB packet as an answer to B packet, where number of registered messages is specified:

Example:

```
#AB#5\r\n
#AB#0\r\n
```

Message to driver

```
#M#msg \r\n
```

Serves for sending a text message to driver. “msg” is the actual text. Message can be sent from either server or tracker.

Server sends AM command as a reply to message from driver.

“1” – message received
 “0” – error receiving message

Example:

```
#AM#1\r\n
#AM#0\r\n
```

Packet with photo

That type of packet is used for sending photos to Wialon's server. The whole photo is separated into equivalent blocks and then each of them is sent to the server. Recommended size of one single block is 512 bytes. If server fails to receive any image block, it breaks the connection. In this case we recommend to reduce the size of blocks down to 256 or 128 bytes.

```
##sz;ind;count;date;time;name\r\nBIN
```

sz	size of packet's binary data (i.e., 512 bytes)
ind	serial number of transmitting block (numbering starts from zero)
count	the whole number of transmitted blocks
date	data in DDMMYY format, in UTC
time	time in HHMMSS format, in UTC
name	name of transmitting photo
BIN	photo's binary block

Server sends AI command as a reply to packet with photo:

```
#AI#ind;0/1\r\n
```

"1" – packet with block of photo is received

"0" – error receiving packet

#AI#1\r\n – photo is completely received and saved in Wialon

Example:

#AI#3;0\r\n

#AI#16;1\r\n

#AI#1\r\n

Packet with new firmware

Serves for sending new firmware to tracker.

#US#sz\r\nBIN

sz	Size of firmware's binary data
BIN	Firmware in binary mode

Packet with configuration file

Serves for sending configuration file to tracker.

#UC#sz\r\nBIN

sz	Size of configuration file, bytes
BIN	Content of configuration file

Incoming data (UDP)

All data is sent and received in plain text format and have the same structure as in TCP protocol, but with adding of IMEI in the beginning of packet. For example, short data packet will look in the following way:

imei#SD#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats\r\n