

Wialon IPS v.2.0

коммуникационный протокол

Коммуникационный протокол Wialon IPS разработан компанией Gurtam для использования в персональных и автомобильных GPS и ГЛОНАСС трекерах, передающих данные на сервер системы спутникового мониторинга по протоколу TCP или UDP.

Изменения

Версия	Дата	Изменения
2.0	10/2014	В пакет L (пакет логина) и в пакеты для передачи по UDP добавлено поле «версия протокола». Добавлены пакеты QT, IT, T, AIT, AT для передачи файлов с тахографов. В пакеты L, SD, D, M, I, IT, T, US, UC добавлено поле контрольной суммы crc16. Для пакетов AL, ASD, AD, AM, AI, AIT, AT добавлен код ошибки проверки контрольной суммы.

Входящие данные TCP протокола

Все данные приходят в текстовом формате и представляют собой пакет следующего вида:

#TP#msg\r\n

#	стартовый байт
TP	тип пакета, описание всевозможных типов указано в табл. 1
#	Разделитель
msg	непосредственно сообщение
\r\n	концевик пакета

Типы пакетов

таблица 1

Тип	Описание	Кто отправляет
L	Пакет логина	оборудование
AL	Ответ на пакет логина	Сервер
D	Пакет с данными	оборудование
AD	Ответ на пакет с данными	Сервер
P	Пинговый пакет	оборудование
AP	Ответ на пинговый пакет	Сервер
SD	Сокращённый пакет с данными	оборудование
ASD	Ответ на сокращённый пакет	Сервер
B	Пакет из чёрного ящика	оборудование
AB	Ответ на пакет из чёрного ящика	Сервер
M	Сообщение для водителя	оборудование/сервер
AM	Ответ на сообщение от водителя	Сервер
QI	Команда запроса фотоизображения	Сервер
I	Пакет с фотоизображением	оборудование
QT	Команда запроса файла с тахографа (ddd-файла)	Сервер
IT	Пакет с информацией о ddd-файле	оборудование
AIT	Ответ на пакет IT	Сервер
T	Пакет с блоком ddd-файла	оборудование
AT	Ответ на пакет T	Сервер
AI	Ответ на пакет с фотоизображением	Сервер
US	Пакет с новой прошивкой	Сервер
UC	Пакет с файлом конфигурации	Сервер

Пакет логина

#L#protocol_version;imei;password;crc16\r\n

Пример: #L#2.0;imei;N/A;BB2B

protocol_version	Версия протокола. Поле должно содержать значение 2.0
;	Разделитель
imei	уникальный ID контроллера, IMEI или серийный номер

password	пароль для доступа к устройству, если отсутствует, то передаётся NA
crc16	контрольная сумма (см. Приложение 1)

В ответ на пакет логина сервер посылает команду AL:
 "1" - если авторизация объекта на сервер прошла успешно
 "0" - если сервер отверг подключение
 "01" - если ошибка проверки пароля
 "10" - если ошибка проверки контрольной суммы

Пример:
 #AL#1\r\n
 #AL#0\r\n

Сокращённый пакет с данными

#SD#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats;crc16\r\n

date	дата в формате DDMMYY, в UTC, если отсутствует, то передаётся NA
time	время в формате HHMMSS, в UTC, если отсутствует, то передаётся NA
lat1;lat2	широта (5544.6025;N), если отсутствует, то передаётся NA;NA
lon1;lon2	долгота (03739.6834;E), если отсутствует, то передаётся NA;NA
speed	скорость, целое число, км/ч, если отсутствует, то передаётся NA
course	курс, целое число, градусы, если отсутствует, то передаётся NA
height	высота, целое число, в метрах, если отсутствует, то передаётся NA
sats	количество спутников, целое число, если отсутствует, то передаётся NA
crc16	Контрольная сумма (см. Приложение 1)

Если поля date и time содержат значение NA - берётся текущее время сервера.

В ответ на сокращённый пакет с данными сервер посылает команду ASD:

- “-1” – ошибка структуры пакета
- “0” – некорректное время
- “1” – пакет успешно зафиксировался
- “10” – ошибка получения координат
- “11” – ошибка получения скорости, курса или высоты
- “12” – ошибка получения количества спутников
- “13” – ошибка проверки контрольной суммы

Пример:

```
#ASD#1\r\n
#ASD#0\r\n
#ASD#10\r\n
#ASD#11\r\n
#ASD#12\r\n
```

Пакет с данными

```
#D#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats;hdop;inputs;outputs;adc;ibutton;params;crc16\r\n
```

date	дата в формате DDMMYY, в UTC, если отсутствует, то передаётся NA
time	время в формате HHMMSS, в UTC, если отсутствует, то передаётся NA
lat1;lat2	широта (5544.6025;N), если отсутствует, то передаётся NA;NA
lon1;lon2	долгота (03739.6834;E), если отсутствует, то передаётся NA;NA
speed	скорость, целое число, км/ч, если отсутствует, то передаётся NA
course	курс, целое число, градусы, если отсутствует, то передаётся NA
height	высота, целое число, в метрах, если отсутствует, то передаётся NA
sats	количество спутников, целое число, если отсутствует, то передаётся NA
hdop	снижение точности, дробное число, если отсутствует, то передаётся NA

inputs	цифровые входы, каждый бит числа соответствует одному входу, начиная с младшего, целое число, если отсутствует, то передаётся NA
outputs	цифровые выходы, каждый бит числа соответствует одному выходу, начиная с младшего, целое число, если отсутствует, то передаётся NA
adc	аналоговые входы, дробные числа, через запятую. Нумерация выхода начинается с единицы, передается пустая строка, если нету никаких аналоговых входов. Пример: 14.77,0.02,3.6
ibutton	код ключа водителя, строка произвольной длины. В случае отсутствия ключа передаётся NA
params	<p>набор дополнительных параметров через запятую. Каждый параметр представляет собой конструкцию NAME:TYPE:VALUE</p> <p>NAME – произвольная строка, длиной не более 15 байт</p> <p>TYPE – тип параметра, 1 -int/long long, 2 – double, 3 – string</p> <p>VALUE – значение в зависимости от типа</p> <p>Для передачи тревожной кнопки используется параметр первого типа с именем «SOS», значение 1 означает нажатие тревожной кнопки.</p> <p>Для передачи текстового сообщения используется параметр третьего типа (string) с именем «text».</p> <p>Данный параметр можно использовать для передачи текстового сообщения от водителя, которое может содержать координаты и другие параметры.</p> <p>Примеры: count1:1:564,fuel:2:45.8,hw:3:V4.5 SOS:1:1</p>
crc16	контрольная сумма (см. Приложение 1)

Если поля date и time содержат значение NA – берётся текущее время сервера.

В ответ на пакет с данными сервер посылает команду AD:
“-1” – ошибка структуры пакета

- “0” – некорректное время
- “1” – пакет успешно зафиксировался
- “10” – ошибка получения координат
- “11” – ошибка получения скорости, курса или высоты
- “12” – ошибка получения количества спутников или hdrop
- “13” – ошибка получения inputs или outputs
- “14” – ошибка получения adc
- “15” – ошибка получения дополнительных параметров
- “16” – ошибка проверки контрольной суммы

Пример:

```
#AD#1\r\n  
#AD#0\r\n  
#AD#10\r\n  
#AD#11\r\n
```

Пинговый пакет

```
#P#\r\n
```

Служит для поддержания активного TCP соединения с сервером, а так же для проверки работоспособности канала.

В ответ на пинговый пакет сервер посылает команду AP:

Пример:

```
#AP#\r\n
```

Пакет из чёрного ящика

```
#B#msg|crc16\r\n
```

Пакет с чёрным ящиком предназначен для передачи несколько сообщений одновременно.

“msg” представляет собой несколько тел сокращённых или полных пакетов (без указания типа), разделённых между собой символом ‘|’.

“crc16” контрольная сумма (см. Приложение 1)

Пример:

```
#B#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats|  
date;time;lat1;lat2;lon1;lon2;speed;course;height;sats|  
date;time;lat1;lat2;lon1;lon2;speed;course;height;sats|crc16\r\n
```

В ответ на пакет из чёрного ящика сервер посылает команду АВ, где указывается количество зафиксированных сообщений:

Пример:

```
#AB#3\r\n  
#AB#0\r\n
```

или пустая строка, которая означает ошибку проверки контрольной суммы.

Пример:

```
#AB#\r\n
```

Сообщение для водителя

```
#M#msg;crc16\r\n
```

Служит для отправки текстового сообщения водителю.

“msg” - непосредственно текст сообщения. Сообщение может посылаться как сервером так и оборудованием.

“crc16” - контрольная сумма (см. Приложение 1)

В ответ на сообщение для водителя сервер посылает команду АМ:

“1” - сообщение принято

“0” - ошибка принятия сообщения

“01” - ошибка проверки контрольной суммы

Пример:

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

Команда для запроса фотоизображения

#QI#\r\n

Применяется для запроса фотоизображения с оборудования.

Пакет с фотоизображением

Служит для отправки фотоизображений на сервер Wialon. Всё изображение разбивается на блоки байт, каждый из которых отправляется на сервер с помощью этого пакета.

Рекомендуемый размер блока - до 50 килобайт. Если сервер не может получить какой-нибудь из блоков изображения - он разрывает соединение, в этом случае рекомендуем уменьшить размер блоков.

#I#sz;ind;count;date;time;name;crc16\r\nBIN

sz	размер бинарных данных пакета (к примеру, 51200 байт)
ind	порядковый номер передаваемого блока (нумерация с нуля)
count	номер последнего блока при нумерации с нуля
date	дата в формате DDMMYY, в UTC
time	время в формате HHMMSS, в UTC
name	имя передаваемого изображения
crc16	Контрольная сумма (см. Приложение 1)
BIN	бинарный блок изображения размера sz

В ответ на пакет с фотоизображением сервер посылает команду AI:

#AI#NA;0\r\n означает ошибку структуры пакета

#AI#ind;result\r\n

ind - порядковый номер передаваемого блока

result – результат обработки пакета:

“1” – если пакет с блоком изображения принят

“0” – если ошибка принятия пакета

“01” – если ошибка проверки контрольной суммы

#AI#1\r\n – изображение полностью принято и сохранено в Wialon

Пример:

Трекер: #I#51200;0;1;070512;124010;sample.jpg;crc16\r\nBIN

Сервер: #AI#0;1\r\n

Трекер: #I#28923;1;1;070512;124010;sample.jpg;crc16\r\nBIN

Сервер: #AI#1;1\r\n

Сервер: #AI#1\r\n

Команда для запроса файла с тахографа

#QT#driverid\r\n

Применяется для запроса ddd-файла с тахографа.

driverid	строка идентифицирующая водителя
----------	----------------------------------

Пакет с информацией о ddd-файле

Пакет должен быть передан перед отправкой ddd-файла.

#IT#date;time;driverid;code;count;crc16\r\n

date	Дата в формате DDMMYY, в UTC
time	Время в формате HHMMSS, в UTC
driverid	Строка идентификации водителя
code	Код ошибки. Если ошибки отсутствуют, передается

	пустая строка
count	Общее количество блоков ddd-файла
crc16	Контрольная сумма для части date;time;driverid;code;count;

На пакет #IT# сервер посылает ответ:

#AIT#state\r\n

state	<p>“1” - если пакет принят</p> <p>“0” - если ошибка принятия пакета</p> <p>“01” - если ошибка проверки контрольной суммы</p>
-------	--

После передачи пакета #IT# должны следовать пакеты #T# содержащие блоки ddd-файла. Формат пакета #T# описан далее.

На сервере файл сохраняется с именем вида driverid_yyyymmdd_hhmmss.ddd

Пакет с блоком ddd-файла

#T#code;sz;ind;crc16\r\nBIN

code	Код ошибки. Если ошибки отсутствуют, передается пустая строка
sz	Размер бинарных данных пакета в байтах
ind	Порядковый номер передаваемого блока (нумерация с нуля)
crc16	Контрольная сумма для бинарного блока BIN

BIN	Бинарный блок файла размера sz
-----	--------------------------------

На каждый пакет сервер посылает ответ:

#AT#ind;state\r\n

ind	Порядковый номер полученного блока
state	“1” – если пакет принят “0” – если ошибка принятия пакета “01” – если ошибка проверки контрольной суммы

Когда ddd-файл полностью принят и сохранен, сервер отвечает

#AT#1\r\n

Все пакеты с блоками ddd-файла должны быть переданы в том же TCP-соединении, что и пакет #IT#.

Пакет с новой прошивкой

Служит для отправки новой прошивки на контроллер.

#US#sz;crc16\r\nBIN

sz	размер бинарных данных прошивки
crc16	контрольная сумма (см. Приложение 1)
BIN	прошивка в бинарном виде

Пакет с файлом конфигурации

Служит для отправки файла конфигурации на контроллер.

#UC#sz;crc16\r\nBIN

sz	размер файла конфигурации, байт
----	---------------------------------

crc16	контрольная сумма (см. Приложение 1)
BIN	Содержимое файла конфигурации

Входящие данные UDP протокола

Все данные приходят в текстовом формате и представляют собой такую же структуру как и в TCP протоколе с добавлением в начале пакета версии протокола и imei устройства. Поле версии протокола должно содержать значение "2.0".

К примеру, сокращённый пакет с данными будет выглядеть следующим образом:

```
2.0;imei#SD#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats;crc16\r\n
```

Сжатие данных

При передаче данных на сервер может использоваться сжатие исходных текстовых пакетов протокола Wialon IPS. Его целесообразно использовать при передаче больших пакетов #B# для экономии трафика.

Для сжатия используется алгоритм DEFLATE кроссплатформенной библиотеки **zlib** (<http://www.zlib.net/>, [RFC 1951](#)).

Поддерживаются оба транспортных протокола – TCP и UDP. Серверные ответы всегда остаются несжатыми ввиду их небольшого размера.

Структура контейнера со сжатым пакетом:

Размер	1 байт	2 байта	Указанный размер
Содержимое	Признак сжатия – байт 0xFF	Размер сжатых данных (little-endian, 16-bit integer)	Блок сжатых бинарных данных указанного размера, передаётся как есть

Контейнер должен содержать только один пакет текстового протокола.

При использовании сжатия необходимость в символах `\r\n` на конце каждого пакета текстового протокола отпадает, и они могут быть опущены на передающей стороне перед сжатием. Одновременно могут передаваться как сжатые, так и обычные пакеты протокола Wialon IPS.

Пример пакета со сжатием:

Исходный текст пакета	#L#imei;password
Полные данные сжатого пакета (27 байт, в HEX виде)	FF180078DA53F651CECC4DCDB42E482C2E2ECF2F4A01002D1C05E5 FF – признак сжатия 1800 – размер сжатых данных, 24 (0x18) байта 78DA... – сжатые данные

Приложение 1

Поле `crc16` должно содержать значение контрольной суммы в шестнадцатеричном виде в формате `big-endian`, без ведущих нулей слева, например:

`AA13BB`, что в десятичном формате представляет собой число 11146171.

В пакетах `L`, `SD`, `D`, `B`, `M` для расчета `crc16` берется часть пакета между `#TP#` и полем `crc16`.

Пример пакета:

```
#SD#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats;crc16\r\n
```

В данном случае `crc16` рассчитывается для следующей части пакета:

```
date;time;lat1;lat2;lon1;lon2;speed;course;height;sats;
```

Пример пакета:

```
#B#date;time;lat1;lat2;lon1;lon2;speed;course;height;sats|  
date;time;lat1;lat2;lon1;lon2;speed;course;height;sats|crc16\r\n
```

В данном случае `crc16` рассчитывается для следующей части пакета:

```
date;time;lat1;lat2;lon1;lon2;speed;course;height;sats|  
date;time;lat1;lat2;lon1;lon2;speed;course;height;sats|
```

В пакетах `I`, `US`, `UC`, `T` для расчета `crc16` берется поле `BIN`.

Пример пакета:

```
#I#51200;0;1;070512;124010;sample.jpg;crc16\r\nBIN
```

В данном случае `crc16` рассчитывается для поля `BIN`.

Пример кода на языке `C` для расчета `crc16`:

```
static const unsigned short crc16_table[256] =  
{  
    0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,  
    0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
```

```
0xCC01,0x0CC0,0x0D80,0xCD41,0x0F00,0xCFC1,0xCE81,0x0E40,
0x0A00,0xCAC1,0xCB81,0x0B40,0xC901,0x09C0,0x0880,0xC841,
0xD801,0x18C0,0x1980,0xD941,0x1B00,0xDBC1,0xDA81,0x1A40,
0x1E00,0xDEC1,0xDF81,0x1F40,0xDD01,0x1DC0,0x1C80,0xDC41,
0x1400,0xD4C1,0xD581,0x1540,0xD701,0x17C0,0x1680,0xD641,
0xD201,0x12C0,0x1380,0xD341,0x1100,0xD1C1,0xD081,0x1040,
0xF001,0x30C0,0x3180,0xF141,0x3300,0xF3C1,0xF281,0x3240,
0x3600,0xF6C1,0xF781,0x3740,0xF501,0x35C0,0x3480,0xF441,
0x3C00,0xFCC1,0xFD81,0x3D40,0xFF01,0x3FC0,0x3E80,0xFE41,
0xFA01,0x3AC0,0x3B80,0xFB41,0x3900,0xF9C1,0xF881,0x3840,
0x2800,0xE8C1,0xE981,0x2940,0xEB01,0x2BC0,0x2A80,0xEA41,
0xEE01,0x2EC0,0x2F80,0xEF41,0x2D00,0xEDC1,0xEC81,0x2C40,
0xE401,0x24C0,0x2580,0xE541,0x2700,0xE7C1,0xE681,0x2640,
0x2200,0xE2C1,0xE381,0x2340,0xE101,0x21C0,0x2080,0xE041,
0xA001,0x60C0,0x6180,0xA141,0x6300,0xA3C1,0xA281,0x6240,
0x6600,0xA6C1,0xA781,0x6740,0xA501,0x65C0,0x6480,0xA441,
0x6C00,0xACC1,0xAD81,0x6D40,0xAF01,0x6FC0,0x6E80,0xAE41,
0xAA01,0x6AC0,0x6B80,0xAB41,0x6900,0xA9C1,0xA881,0x6840,
0x7800,0xB8C1,0xB981,0x7940,0xBB01,0x7BC0,0x7A80,0xBA41,
0xBE01,0x7EC0,0x7F80,0xBF41,0x7D00,0xBDC1,0xBC81,0x7C40,
0xB401,0x74C0,0x7580,0xB541,0x7700,0xB7C1,0xB681,0x7640,
0x7200,0xB2C1,0xB381,0x7340,0xB101,0x71C0,0x7080,0xB041,
0x5000,0x90C1,0x9181,0x5140,0x9301,0x53C0,0x5280,0x9241,
0x9601,0x56C0,0x5780,0x9741,0x5500,0x95C1,0x9481,0x5440,
0x9C01,0x5CC0,0x5D80,0x9D41,0x5F00,0x9FC1,0x9E81,0x5E40,
0x5A00,0x9AC1,0x9B81,0x5B40,0x9901,0x59C0,0x5880,0x9841,
0x8801,0x48C0,0x4980,0x8941,0x4B00,0x8BC1,0x8A81,0x4A40,
0x4E00,0x8EC1,0x8F81,0x4F40,0x8D01,0x4DC0,0x4C80,0x8C41,
0x4400,0x84C1,0x8581,0x4540,0x8701,0x47C0,0x4680,0x8641,
0x8201,0x42C0,0x4380,0x8341,0x4100,0x81C1,0x8081,0x4040
};

unsigned short crc16 (const void *data, unsigned data_size)
{
    if (!data || !data_size)
        return 0;

    unsigned short crc = 0;
    unsigned char* buf = (unsigned char*)data;

    while (data_size--)
        crc = (crc >> 8) ^ crc16_table[(unsigned char)crc ^ *buf++];

    return crc;
}
```