

КОММУНИКАЦИОННЫЙ ПРОТОКОЛ WIALON IPS v.2.0

Версия	Дата	Изменения
2.0	10.2014	<p>В пакет L (пакет логина) добавлено поле «версия протокола».</p> <p>В пакеты L, SD, D, M, I, IT, T, US, UC добавлено поле контрольной суммы crc16.</p> <p>Для пакетов AL, ASD, AD, AM, AI, AIT, AT добавлен код ошибки проверки контрольной суммы.</p>

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	сообщение
\r\n	концевик пакета

Типы пакетов (таблица 1)

Тип	Описание	Отправитель
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 с чёрным ящиком	оборудование
AB	ответ на пакет с чёрным ящиком	сервер
M	сообщение для водителя	оборудование/сервер
AM	ответ на сообщение от водителя	сервер
US	пакет с новой прошивкой	сервер
UC	пакет с файлом конфигурации	сервер



Пакет логина

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

;	разделитель
protocol_version	Версия протокола. Поле должно содержать значение 2.0
imei	уникальный ID трекера, IMEI или серийный номер
password	пароль для доступа к устройству, при отсутствии передаётся NA
crc16	контрольная сумма (см. Приложение 1)

В ответ на пакет логина сервер посылает команду AL:

- "1" – при успешной авторизации на сервер;
- "0" – при отказе сервера от подключения;
- "01" – при ошибке проверки пароля;
- "10" – при ошибке проверки контрольной суммы.

Примеры:

#AL#1\r\n #AL#0\r\n

Сокращенный пакет с данными

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

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

Если поля date и time содержат NA, используется 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.

Примеры:

#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, при отсутствии передается NA
time	время в формате HHMMSS, в UTC, при отсутствии передается NA
lat1;lat2	широта (5544.6025;N), при отсутствии передается NA;NA
lon1;lon2	долгота (03739.6834;E), при отсутствии передается NA;NA
speed	скорость, целое число, км/ч, при отсутствии передается NA
course	курс, целое число, градусы, при отсутствии передается NA
height	высота, целое число, в метрах, при отсутствии передается NA
sats	количество спутников, целое число, при отсутствии передается NA
hdop	снижение точности, дробное число, при отсутствии передается NA
inputs	цифровые входы, каждый бит числа соответствует одному входу, начиная с младшего, целое число, при отсутствии передается NA
outputs	цифровые выходы, каждый бит числа соответствует одному выходу, начиная с младшего, целое число, при отсутствии передается NA
adc	аналоговые входы, дробные числа, через запятую. Нумерация выхода начинается с единицы; Если аналоговых входов нет - передается пустая строка. Пример: 14.77,0.02,3.6
ibutton	код ключа водителя, строка произвольной длины. При отсутствии ключа передается NA
params	набор дополнительных параметров через запятую. Каждый параметр представляет собой конструкцию NAME:TYPE:VALUE NAME – произвольная строка, длиной не более 15 байт; TYPE –тип параметра, 1 –int/long long, 2 –double, 3 –string VALUE – значение в зависимости от типа Для передачи тревожной кнопки используется параметр первого типа с именем «SOS», значение 1 означает нажатие тревожной кнопки. Для передачи текстового сообщения используется параметр третьего типа (string) с именем «text». Данный параметр можно использовать для передачи текстового сообщения от водителя, которое может содержать координаты и другие параметры. Примеры: count1:1:564,fuel:2:45.8,hw:3:V4.5 SOS:1:1
crc16	контрольная сумма (см. Приложение 1)

Если поля date и time содержат значение NA, используется 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 hdpор
- "13" – ошибка получения inputsили outputs
- "14" – ошибка получения adc
- "15" –ошибка получения дополнительных параметров
- "16" – ошибка проверки контрольной суммы.

Примеры: #AD#1\r\n #AD#0\r\n #AD#10\r\n #AD#11\r\n #AD#16\r\n

Пинговый пакет

#P#\r\n

Служит для поддержания активного TCP-соединения с сервером, а также 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 с чёрным ящиком сервер посылает команду AB, где указывается количество зафиксированных сообщений:

Примеры: #AB#3\r\n #AB#0\r\n

или пустая строка, которая означает ошибку проверки контрольной суммы:

#AB#\r\n

Сообщение для водителя

#M#msg;crc16\r\n

Служит для отправки текстового сообщения водителю.

"msg" - непосредственно текст сообщения. Сообщение может отправляться как сервером, так и оборудованием

"crc16" контрольная сумма (см. Приложение 1)

В ответ на сообщение для водителя сервер посылает команду AM:

"1" –сообщение принято

"0" –ошибка принятия сообщения

"01" – ошибка проверки контрольной суммы

Примеры: #AM#1\r\n #AM#0\r\n

Пакет с новой прошивкой

Служит для отправки новой прошивки на трекер.

#US#sz;crc16\r\nBIN

sz	размер бинарных данных пакета (к примеру, 51200 байт)
crc16	контрольная сумма (см. Приложение 1)
BIN	прошивка в бинарном виде

Пакет с файлом конфигурации

Служит для отправки файла конфигурации на трекер

#UC#sz;crc16\r\nBIN

sz	размер файла конфигурации, байт
crc16	контрольная сумма (см. Приложение 1)
BIN	содержимое файла конфигурации



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,
0x0500,0xC5C1,0xC481,0x0440,0xCC01,0x0CC0,0x0D80,0xCD41,0xF00,0xFCF1,0xCE81,0xE40,
0xA00,0xCAC1,0xCB81,0xB40,0xC901,0x09C0,0x0880,0xC841,0xD801,0x18C0,0x1980,0xD941,
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;
}
```