

Wialon Combine

Коммуникационный бинарный протокол Wialon Combine 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.

Спецификации:

- *Порядок байт осуществляется в формате Big-Endian.*
- *(Имя_поля)* — Расширяемое поле 1 байт. Старший бит указывает на наличие 2-го байта.*
- *(Имя_поля)** — Расширяемое поле 2 байта. Старший бит указывает на наличие дополнительных 2-х байт.*
- *Все данные приходят в бинарном формате.*
- *Передача данных осуществляется по протоколу TCP и UDP.*

Содержание:

Общая структура данных.....	3
Обязательный ответ сервера.....	3
Формат команды на устройств.....	4
Пакет Login.....	4
Пакет Keep-Alive.....	5
Пакет Data.....	5
Тип подпакета Custom Parameters.....	6
Тип подпакета Position Data.....	7
Тип подпакета I/O.....	7
Тип подпакета Picture.....	8
Тип подпакета LBS Parameters.....	8
Тип подпакета Fuel Parameters.....	9
Тип подпакета Temperature Parameters.....	9
Тип подпакета CAN Parameters.....	9
Тип подпакета Counter Parameters.....	10
Тип подпакета Analog Parameters (ADC).....	10
Тип подпакета Driver code Parameters.....	10
Тип подпакета Tacho File.....	11
Тип подпакета Driver message.....	11
CRC 16.....	11

Ver. 1.0.4

Общая структура данных

Bytes:	2	1	2	2-4			2
Section:	Head	Type*	Seq	Len**	Login (for UDP)	Data	CRC16

Head — 0x2424

Type* :

0 — Login

1 — Data

2 — Keep-Alive

Seq — Порядковый номер (циклический 0 — 65535).

Len** — Длина поля Data.

Data — Полезные данные. Зависит от типа пакета.

Login (for UDP) — Поле передаются при использовании UDP.

CRC16 — Контрольная сумма. Рассчитывается от начала заголовка (head) до последнего байта полезных данных (data).

Обязательный ответ сервера

Сервер отвечает на каждый полученный пакет:

Bytes:	2	1	2
Section:	Head	Code	Seq

Head — 0x4040

Code — Код ответа.

Seq — порядковый номер полученного пакета.

Коды ответа:

0	Пакет успешно зарегистрирован
1	Ошибка авторизации
2	Неверный пароль
3	Пакет не зарегистрирован

4	Ошибка CRC
255	Команда на устройство

Формат команды на устройство

Bytes:	2	1	2-4	4	1-2		2
Section:	Head	Code	Len**	Time	Type*	Data	CRC16

Head — 0x4040

Code — 0xFF

Len — Длина пакета (Поля **Time**, **Type**, **Data**)

Time — Время отправки сообщения

Type — Тип команды

Data — Дополнительные параметры команды

CRC16 — Контрольная сумма. Рассчитывается от начала заголовка (head) до последнего байта полезных данных (data)

Типы команд

0	Custom command
---	----------------

Пакет Login

Пакет **Login** имеет следующий вид:

Bytes:	1	1		
Section:	Protocol version*	Flags	ID	Pwd

Protocol version* (Версия протокола): сейчас 1.

Flags (Битовое поле):

Старшие 4 бита отвечают за тип и размер поля **ID**.

Младшие 4 бита отвечают за тип и размер поля **Pwd**.

Типы **ID**:

1 — unsigned short (2 байта)

2 — unsigned int (4 байта)

- 3 — unsigned long (8 байт)
- 4 — String (последний байт 0x00)

Типы **Pwd**:

- 0 — пароль отсутствует
- 1 — unsigned short (2 байта)
- 2 — unsigned int (4 байта)
- 3 — unsigned long (8 байт)
- 4 — String (последний байт 0x00)

Пакет Keep-Alive

Содержит только 3 первых поля пакета (Head, Type, Seq)

Пакет Data

Пакет данного типа может содержать в себе несколько сообщений.

Каждое сообщение содержит время, длину, а так же набор подзаписей. В общем виде сообщение имеет следующий вид:

Bytes:	4	1	1-2		...	1-2	
Section:	Time	Count	Sub-record type*	Sub-record	...	Type sub-record N	Sub-record N

Time — Время формирования сообщения.

Count — Количество подзаписей.

Subrecord type* — Поле содержащее код типа подзаписи.

Реализованы типы подзаписей:

- 0 — Custom Parameters
- 1 — Position Data
- 2 — I/O Data
- 3 — Picture

- 4 — LBS Parameters
- 5 — Fuel Parameters
- 6 — Temperature Parameters
- 7 — CAN Parameters
- 8 — Counter Parameters
- 9 — Analog Parameters (ADC)
- 10 — Driver code Parameters
- 11 — Tacho File
- 12 — Driver message

Subrecord — Структура данных. Набор полей подзаписей зависит от ее типа.

Тип подзаписи Custom Parameters

Набор данных произвольных полей. Подзапись имеет следующий вид:

Bytes:	1-2	
Section:	Count*	Params

Count* — количество произвольных полей в подзаписе.

Params — набор нумерованных параметров. Каждый параметр будет зарегистрирован как param№. Имеет следующий вид:

Bytes:	1-2	1	
Section:	№*	Type sensor	Value

№* — Номер датчика.

Type sensor —

Поле указывающее тип данных в Value. Имеет следующую структуру (только для целочисленных типов):

Bits:	3	5
Section:	10**X	Type sensor

Для типов 8 и более первые 3 бита всегда равны 0.

10X** — Степень числа 10. Число на которое будет разделено значение параметра **Value**.

Type sensor:

- 0 — unsigned byte (1 байт)
- 1 — unsigned short (2 байта)
- 2 — unsigned int (4 байта)
- 3 — unsigned long (8 байта)
- 4 — signed byte (1 байт)
- 5 — signed short (2 байта)
- 6 — signed int (4 байта)
- 7 — signed long (8 байта)
- 8 — float (4 байта)
- 9 — double (8 байт)
- 10 — String (последний байт 0x00)

Value — значение датчика

Тип подзаписи Position Data

Навигационные данные:

Bytes:	4	4	2	2	2	1	2
Section:	Lat	Lon	Speed	Course	Height	Sats	Hdop

Lat — Широта. Тип signed int. Разделенный на 1000000.

Lon — Долгота. Тип signed int. Разделенный на 1000000.

Speed — Скорость (км/ч).

Course — Направление движения (градусы 0 — 360).

Height — Высота над уровнем моря. Тип signed int.

Sats — количество видимых спутников.

Hdop — Horizontal dilution of precision. Умноженный на 100.

Показывает точность передаваемых устройством координат, чем меньше значение данного параметра, тем более достоверными являются координаты.

Тип подзаписи I/O

Битовое поле. Значения цифровых входов и выходов. Каждый бит числа соответствует одному входу или выходу:

Bytes:	4	4
Section:	Inputs	Outputs

Тип подзаписи **Picture**

Часть фотоизображения, снятого камерой прибора.

Bytes:	1	2-4	1		Len
Section:	Ind*	Len**	Count*	Name	Bin

Ind* — Порядковый номер передаваемого блока (нумерация с нуля).

Len** — Размер блока фотоизображения.

Count* — Номер последнего блока при нумерации с нуля.

Name — Имя передаваемого изображения. Текстовое поле заканчивающееся 0x00.

Bin — Бинарный блок изображения

Тип подзаписи **LBS Parameters**

Bytes:	1	
Section:	Count	LBS param

Count — Количество структур **LBS param**

LBS param:

Bytes:	2	2	2	2	2	2
Section:	MCC	MNC	LAC	Cell ID	Rx level	TA

MCC — Mobile Country Code - код страны.

MNC — Mobile Network Code - код сотовой сети.

LAC — Local Area Code — код локальной зоны. Локальная зона — это совокупность БС, которые обслуживаются одним BSC — контроллером базовых станций.

Cell ID — Идентификатор соты, присваивается оператором каждому сектору каждой БС.

LAC — Local Area Code — код локальной зоны. Локальная зона — это совокупность БС, которые обслуживаются одним BSC — контроллером базовых станций.

Rx level — Уровень принимаемого по данному каналу радиосигнала на входе в приёмник GSM-модема.

TA — Timing Advance - параметр компенсации времени прохождения сигнала от GSM-модема до БС. Фактически означает расстояние до БС.

Тип подзаписи Fuel Parameters

Bytes:	1	
Section:	Count	Fuel (аналог структуры Params)

Count — Количество структур Fuel

(* Все параметры данного поля будут зарегистрированы с именем Fuel№ *)

Тип подзаписи Temperature Parameters

Bytes:	1	
Section:	Count	Temp (аналог структуры Params)

Count — Количество структур Temp

(* Все параметры данного поля будут зарегистрированы с именем temp№ *)

Тип подзаписи CAN Parameters

Bytes:	1	
Section:	Count	Can (аналог структуры Params)

Count — Количество структур Can

(* Все параметры данного поля будут зарегистрированы с именем can№ *)

Тип подзаписи Counter Parameters

Bytes:	1	
Section:	Count	Counter (аналог структуры Params)

Count — Количество структур Counter

(* Все параметры данного поля будут зарегистрированы с именем counter№ *)

Тип подзаписи Analog Parameters (ADC)

Bytes:	1	
Section:	Count	ADC (аналог структуры Params)

Count — Количество структур Can

(* Все параметры данного поля будут зарегистрированы с именем can№ *)

Тип подзаписи Driver code Parameters

Bytes:	1	
Section:	Count	Driver code (аналог структуры Params)

Count — Количество структур **Driver code**

(* Все параметры данного поля будут зарегистрированы с именем driver_code№ *)

Тип подзаписи Tacho File

Файл, записанный тахографом.

Bytes:	1	2-4	1	Len
Section:	Ind*	Len**	Count*	Bin

Ind* — Порядковый номер передаваемого блока (нумерация с нуля).

Len** — Размер блока файла.

Count* — Номер последнего блока при нумерации с нуля.

Bin — Бинарный блок файла с тахографа.

Тип подзаписи Driver message

Сообщение водителю.

Bytes:	Endian 0x00
Section:	Text

Text — Сообщение водителю. Строка заканчивающаяся 0x00.

CRC 16 (Пример кода на языке C):

```
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,
    0x6600,0xA6C1,0xA781,0x6740,0xA501,0xA5C0,0x6480,0xA441,
    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,
    0x5A00,0x9AC1,0x9B81,0x5B40,0x9901,0x99C0,0x5880,0x9841,
    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;
}
```