

WireGeo UDP API

Данный документ описывает протокол, которым пользуются мобильные приложения, контроллеры и Wi-Fi устройства WireGeo, для того чтобы обмениваться показаниями датчиков и командами внутри локальной сети. Данный протокол может быть также использован для интеграции оборудования WireGeo со сторонними системами.

Основные положения

1. Все сообщения отправляются через широковещательные UDP запросы (то есть на адрес 255.255.255.255), порт у всех устройств один и тот же, фиксированный - **43260**.
2. Все сообщения закодированы в JSON.
3. Протокол предназначен главным образом для нерегулярной отправки коротких сообщений, которые предназначены от одного устройства другому устройству, нескольким устройствам или неопределенной группе устройств, каждое сообщение содержит или команду (лампочка X, включить свет), или состояние одной или нескольких переменных устройства (датчик 1, температура - X, влажность - Y, освещенность - Z), или предназначено для того, чтобы устройство могло заявить о своем существовании другим устройствам (я - реле с номером X, у меня есть одна переменная, принимающая команды вкл/выкл и 1 переменная, показывающая температуру). Протокол не предназначен для слишком частой (чаще раз в 10 секунд) отправки больших пакетов (больше десятка переменных). Во всех случаях нужно стараться, чтобы размер пакета был не больше 576 байт (тогда он гарантированно не будет фрагментирован промежуточным сетевым оборудованием).
4. Однозначным идентификатором устройства-отправителя и устройства-получателя сообщения является ID устройства (dev-id). Для устройств WireGeo данный ID является уникальным и «защитым» с завода, другие устройства могут использовать произвольную цифробуквенную последовательность не длиннее 13 символов.
5. Протокол предполагает, что имена полей не содержат пробелов и символов разметки, поэтому всегда пишутся без кавычек. В случае если отправитель пакета считает, что значение поля содержит пробелы или символы разметки JSON, такое значение должно быть взято в кавычки. В примерах ниже пакеты содержат пробелы и символы переноса только в целях повышения наглядности. Отправитель пакета не должен включать в него лишние пробелы и символы переноса строк. Однако получатель не должен рассчитывать на то, что пакет не содержит лишних пробелов и символов переноса строк.
6. Разделитель десятичных знаков в дробных числах - **точка**.

Описания типов и полей сообщений

Сообщение Intro

Предназначено для обнаружения одних устройств другими устройствами. Например, есть шлюз и есть умное реле. Нажали на шлюзе кнопку “Обучение”, шлюз на 60 секунд вошел в режим обучения. Нажали на реле кнопку “Обучение”. Реле отправляет сообщение Intro, шлюз его получает, и с этого момента он знает ID этого устройства и может отправлять команды этому устройству напрямую. При нажатии на кнопку “Обучение” сам шлюз однократно отправляет “в эфир” своё сообщение Intro, чтобы те устройства (или мобильные приложения), которым это нужно, могли узнать его ID.

Поля:

msg: intro - константа, определяющая тип сообщения как Intro (обязательное)

dev-id - ID устройства, отправившего сообщение (обязательное)

dev-type - тип или марка устройства (может отсутствовать)

dest-id - ID устройства, которому предназначено сообщение (может отсутствовать)

Остальные поля могут присутствовать или отсутствовать:

dev-name-en - англоязычное “человеческое” название устройства (для интерфейса)

dev-name-ru - русскоязычное “человеческое” название устройства (для интерфейса)

var - имя переменной устройства dev, значение которой передается

val - передаваемое значение этой переменной

varn - “человеческое” русскоязычное название переменной

varne - “человеческое” англоязычное название переменной

vat - тип переменной (см. справочник ниже). При отсутствии данного поля определение типа переменной остается на усмотрение получателя. Тип также определяет, является ли переменная только для чтения, для записи, или для чтения/записи.

var-num - в случае, если в пакете передается несколько переменных, число переменных

varX - имя переменной X устройства dev, которое передается, где X - 1, 2, 3 и т.д. - натуральное число, начинающееся с 1.

valX - новое значение переменной X, где X - натуральное число, начинающееся с 1.

varnX - “человеческое” русскоязычное название переменной X

varneX - “человеческое” англоязычное название переменной X

vatX - тип переменной X (см. справочник ниже). При отсутствии данного поля определение типа переменной остается на усмотрение получателя.

Пример сообщения:

```
{  
msg: intro,  
dev-type: 'WF10T',  
dev-name-en: "WiFi Relay WF10T",  
dev-name-ru: "WiFi Реле WF10T",  
dev-id:473387431542,  
var-num: 2,  
var1: power,
```

```
val1: 1,  
vat1: bool,  
vanr1: "Реле",  
vane1: "Relay",  
var2: temp,  
val2: "-1",  
vat2: float,  
vanr2: "Температура",  
vane2: "Temperature",  
}
```

Сообщение Status

Предназначено для передачи другим устройствам состояния своих переменных. Устройство отправляет это сообщение по собственной инициативе, когда изменились какие-то значения его собственных переменных. Устройство само принимает решение, кому и что отправлять, а также принимает на себя ответственность не спамить своими сообщениями общий "эфир".

Поля:

msg: stat - константа, определяющая тип сообщения как Status (обязательное)

dev-id - ID устройства, отправившего сообщение (обязательное)

dest-id - ID устройства, которому предназначено сообщение (может отсутствовать)

Остальные поля могут присутствовать или отсутствовать:

var - имя переменной устройства dev, значение которой передается

val - передаваемое значение этой переменной

var-num - в случае, если в пакете передается несколько переменных, число переменных

varX - имя переменной X устройства dev, которое передается, где X - 1, 2, 3 и т.д. - натуральное число, начинающееся с 1.

valX - новое значение переменной X, где X - натуральное число, начинающееся с 1.

Пример сообщения:

```
{  
msg: stat,  
dev-id:473387431542,  
var: temp,  
val: "9.5"  
}
```

Сообщение Command

Предназначено для отдачи команды другому устройству.

Поля:

msg: cmnd или cmndrep - константа, определяющая тип сообщения как Command (обязательное). Если значение = cmndrep, то это повторная команда.

dev-id - ID устройства, отправившего сообщение (может отсутствовать)

dest-id - ID устройства, которому предназначено сообщение (обязательное)

Остальные поля могут присутствовать или отсутствовать:

var - имя переменной устройства dest, которую нужно изменить

val - новое значение этой переменной

var-num - в случае, если в пакете передается несколько переменных, число переменных

varX - имя переменной X устройства dest, которую нужно изменить, где X - 1, 2, 3 и т.д. - натуральное число, начинающееся с 1.

valX - новое значение переменной X, где X - натуральное число, начинающееся с 1.

seqnum - случайное 4-байтное число, которое однозначно определяет сообщение в последовательности сообщений. Если у двух сообщений seqnum одинаковые, значит это одинаковые сообщения, и второе сообщение является повтором ранее отправленного сообщения с тем же содержанием. Контроллер, устройство и или мобильное приложение может отправить несколько одинаковых сообщений, чтобы повысить вероятность доставки. В этом случае для второго и последующих сообщений msg должно быть равно cmndrep. Это сделано с целью обратной совместимости. Если более старое устройство получит пакет с msg=cmndrep, о существовании которого оно не знает, то оно его проигнорирует.

req - специальный запрос устройству (request). Определены следующие специальные запросы:

“stat” - в ответ на данный запрос устройство должно отправить сообщение типа status с текущими значениями всех своих переменных.

“assign” - данный запрос отправляет контроллер обнаруженному им устройству, для того, чтобы устройство переключилось из прямого подчинения облаку в подчинение к контроллеру. После получения данного запроса устройство больше не должно отправлять обновленные данные в облако, вместо этого оно должно отправлять данные только отправившему данную команду контроллеру.

“unassign” - после получения данного запроса устройство должно переключиться с подчинения контроллеру на подчинение облаку.

Пример сообщения:

```
{
```

```
msg: cmnd,
```

```
dev-id:7472723,  
dest-id:473387431542,  
var-num: 2,  
var1: settemp1,  
val1: "-12.4"  
var2: power1,  
val2: 1  
}
```

Примечание: в пакете может передаваться или одна пара var и val, или же var-num и указанное в нем количество пар varX и valX.