Однофазный WiFi счетчик.
Описание формата настроек
Версия 1.0.1

Содержание

Protobuf 1

proto-файл 1

Работа с настройками 2

Права доступа 3

Protobuf

(https://developers.google.com/protocol-buffers)

Protocol Buffers — протокол сериализации (передачи) структурированных данных, предложенный Google как эффективная бинарная альтернатива текстовому формату XML. Разработчики сообщают, что Protocol Buffers проще, компактнее и быстрее, чем XML, поскольку осуществляется передача бинарных данных, оптимизированных под минимальный размер сообщения

proto-файл

WiFi счетчик использует protobuf v3.1.0. Работа других версий не гарантируется.

Описание формата настроек описывается proto-файлом с расширением *.proto. Например, основное сообщение настроек:

```
message MeterSettings
{
  required uint32 version = 1;

  optional DeviceSettings device = 2;
  optional CalendarSettings calendar = 3;
  optional LogSettings log = 4;
  optional TouSettings tou = 5;
  optional LoadProfileSettings load_profile = 6;
  optional CalibrationSettings calibration = 8;
  optional DisplaySettings display = 9;
  optional KeysSettings keys = 10;
  optional KeysNetworkAccessSettings keys_network_access = 11;
  optional VpnClientSettings vpn_client = 12;
  optional WifiPasswordSettings wifi_password = 13;
  optional RelaySettings relay = 14;
  optional PQMSettings pqm = 15;
  optional NetworkSettings network = 16;
  optional MetrologySettings metrology = 17;
  optional KeysMacSettings keys_mac = 18;
  optional DemandSettings demand = 20;
```

```
  optional NotificationsSettings notifications = 21;
  optional LimiterSettings limiter = 22;
  optional OpticalPortSettings opticalport = 23;
  optional BillingSettings billing = 24;
}
```

Чтобы скомпилировать настройки с вашим языком программирования, необходимо подключить библиотеку protobuf и использовать специальный protobuf компилятор. Компиляция настроек Protobuf была протестирована со следующими языками программирования: C/C++/JS/Lua/Python.

C компиляция:

```
protoc -I settings/proto/ Settings.proto --nanopb_out=settings/proto/
```

C++ компиляция:

```
protoc -I settings/proto/ Settings.proto --cpp_out=settings/proto/
```

JavaScript компиляция:

```
protoc -I settings/proto/ Settings.proto --js_out=import_style=commonjs,binary:settings/proto/
```

Lua компиляция:

```
protoc -I settings/proto/ Settings.proto --lua_out=settings/proto/
```

Python компиляция:

```
protoc -I settings/proto/ Settings.proto --python_out=settings/proto/
```

После этого вы получите файлы с описанием настроек для вашего языка программирования. Так же библиотека protobuf предоставляет возможность сериализации данных в JSON-объект и обратно.

Работа с настройками

Настройки устройства разделены на группы (DeviceSettings, CalendarSettings, LogSettings…). Каждая группа может быть записана отдельно (отмечена как «опциональная» в главном сообщении «MeterSettings»). Но внутри группы должны быть все настройки (даже необязательные поля), в противном случае, отсутствие настройки в группе будет означать ее деактивацию. Например, если вы записываете настройки **limiter** без опционального поля **ovp_limit** - это означает, что настройка **ovp_limit** будет отключена.

Для изменения отдельных полей используется принцип read–modify–write.

Группы **DeviceSettings**, **CalibrationSettings** предназначны только для чтения. Запись в эти группы невозможна.

При записи настроек необходимо указывать поле **version** в основном теле настроек. Значение - любое число.

С устройства считываются сериализованные настройки. Эти двоичные данные должны быть разобраны сгенерированным парсером (например, функцией ParseFromArray() в C++). Далее вы можете работать с настройками.

Чтобы загрузить и применить настройки, их нужно сериализовать (например, с помощью функции SerializeToArray() в C++). При загрузке необходимо учитывать, что размер блока данных COAP - 128 байт.

Единицы измерения, ограничения в размере и прочие особенности описаны в комментариях *.proto файла.

Права доступа

Таблица соответствия уровня доступа и ресурсов представлена ниже

Режим Чтения (GET запросы)

Ресурс	Гость	Биллинг	Сервис	Администратор
device	•	•	•	•
calibration		•	•	•
calendar		•	•	•
log		•	•	•
tou		•	•	•
load_profile		•	•	•
instrumentation_profile		•	•	•
display		•	•	•
keys				
network		•	•	•
keys_network_access				
vpn_client			•	•
wifi_password				•
pqm		•	•	•
relay		•	•	•
tampering		•	•	•
metrology		•	•	•
demand		•	•	•
keys_mac				
notifications		•	•	•
limiter		•	•	•
opticalport		•	•	•
counters		•	•	•

Режим Записи (PUT/POST/DELETE запросы)

Ресурс	Гость	Биллинг	Сервис	Администратор
device				
calibration				
calendar			•	•
log				•
tou			•	•
load_profile			•	•
instrumentation_profile			•	•
display			•	•
keys				•
network			•	•
keys_network_access			•	•
vpn_client			•	•
wifi_password				•
pqm			•	•
relay			•	•
tampering				•
metrology				
demand			•	•
keys_mac				•
notifications			•	•
limiter			•	•

Ресурс	Гость	Биллинг	Сервис	Администратор
opticalport				
counters			•	•