



RUT955  
wialon gateway

Version 1.0

Николай Котцов  
Апрель 2022

## Содержание

1	Информация о руководстве	2
2	Возможности и настройки конфигов	2
3	Модули	4
4	Установка	5
5	Настройка RUT устройства для передачи данных	5

## 1 Информация о руководстве

Данное руководство описывает основные функциональные возможности RUT955 wialon gateway, его установку и настройку. Данный шлюз служит посредником между вервером обработки данных и устройствами с котоых приходят эти данные. Полученные данные имеют разные форматы, поэтому они конвертируются в wialon формат. Схема работы предствалена в конце файла.

## 2 Возможности и настройки конфигов

В первую очередь необходимо собрать данные. Для этого сущесвует система модулей и их конфигов. В версии 1.0 "из коробки"имеется 3 модуля MQTT, Modbus и кастомный модуль. Данные модули с их нстройками будут рассмотрены ниже.

- Для того чтобы желаемый модуль работал, необходимо указать его в конфиге модулей. Где необходимо указать имя модуля и путь к его более детальным настройкам.

rut\_modules\_config.json

```
{
  "modules": [
    {
      "name": "mqtt",
      "module_config_path": "module_mqtt_config.json"
    },
    {
      "name": "custom",
      "module_config_path": "module_custom_1_config.json"
    }
  ]
}
```

- После чего полученные данные попадают в обработчик данных, где они преобразуются в подходящий формат(в будущих версиях конвертация в готовый вид будет происходить в блоке wialon client, здесь же данные будут конвертироваться в два набора) В данный момент есть настройка таймера отправки конвертированных данных и размер буфера для канала источников данных.

rut\_data\_processing\_service\_config.json

```
{
  "data_source_channel_size": 1000,
  "ticker_time": 10
}
```

- В котловом к отправке виде данные либо отправляются на сервер, либо сохраняются в буфер при отсутствии соединения с сервером. Для корректной работы этого блока необходимо настроить подключение к серверу. Адрес сервера, логин и пароль(логин, он же id может использоваться без пароля). Также путь к файлу буферизации, в случае необходимости можно указать на не пустой файл, тогда при запуске первым делом будут отправлены данные с буфера.

rut\_wialon\_client\_config.json

```
{
  "wialon_server_address": "192.168.100.107:11113",
  "connection_type": "tcp",
  "data_buffer_path": "buffer.buf",
  "login": "admin",
  "password": "admin"
}
```

- Также в случае необходимости можно настроить путь ко всем описанным выше конфигам в файле.

rut\_gateway\_config\_paths.json

```
{
  "wialon_client_config_path": "rut_wialon_client_config.json",
  "data_processing_service_config_path": "rut_data_processing_service_config.json",
  "modules_config_path": "rut_modules_config.json"
}
```

### 3 Модули

- MQTT

mqtt модуль может сделать множество подписок на разные топики различных брокеров. Модуль ожидает на вход одноуровневую json структуру, где по имени поля выделит необходимые данные. Конфигурация данного модуля состоит из создания описания клиентов, где необходимо указать ip:port брокера, список желаемых топиков и названия 3 полей в которых будет храниться необходимая информация. В случае если брокер не может отправлять тип переданного значения, автоматически будет обрабатываться как string и иметь код 3(для wialon).

```
{
  "clients": [
    {
      "ip_port": "127.0.0.1:18883",
      "subscription_list": [
        "#"
      ],
      "data_format": {
        "data_name_field": "req_name",
        "data_type_field": "type",
        "data_value_field": "data_value"
      }
    }
  ]
}
```

- Modbus

В данный момент модуль может работать, однако файл конфигурации слишком сложный, поэтому было принято решение получать данные через rut. Однако при необходимости можно работать напрямую с протоколом Modbus.

- Custom

Настраиваемый модуль. В данный момент работает так, что все получаемые данные на определенный порт отправляются дальше по цепочке. В конфиге необходимо мод определения ip - auto | manual , указать ip и порт. В случае автоматического определения, ip будет определен автоматически

```
{
  "list": [
    {
      "mode": "auto",
      "ip": "none",
      "port": "32211"
    }
  ]
}
```

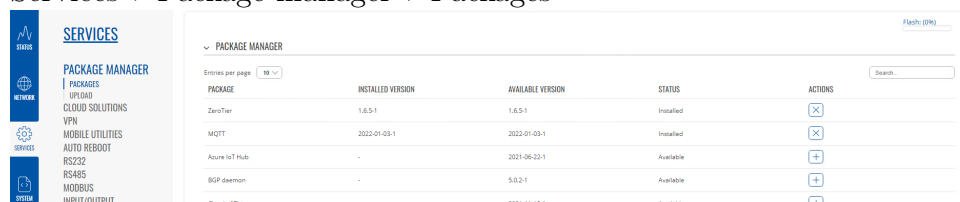
## 4 Установка

подробный гайд по установке и настройке системы через ssh

## 5 Настройка RUT устройства для передачи данных

Для работы с протоколом MQTT, необходимо установить пакет MQTT, который можно найти по пути

- Services » Package manager » Packages



PACKAGE	INSTALLED VERSION	AVAILABLE VERSION	STATUS	ACTIONS
ZeroTier	1.6.5-1	1.6.5-1	Installed	[X]
MQTT	2022-01-03-1	2022-01-03-1	Installed	[X]
Azure IoT Hub	-	2021-06-22-1	Available	[+]
BGP daemon	-	5.0.2-1	Available	[+]
Cloud of Things	-	2021-11-12-1	Available	[+]

- В окне поиска задать MQTT

- Среди полученных результатов найти интересующий, нажать на + , после чего подтвердить.

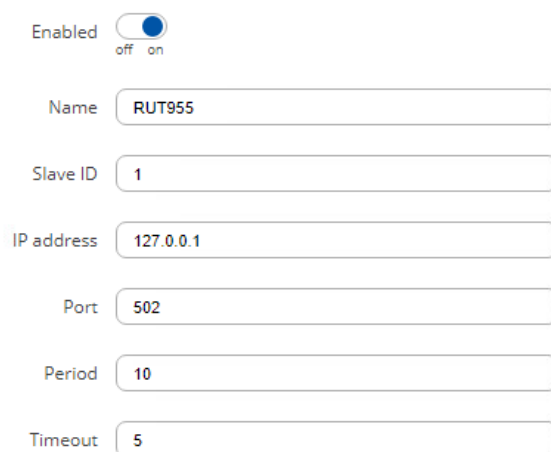
После установки всего необходимого, нужно все настроить. Во вкладке SERVICES

- Modbus
  - TCP Slave - служит для предоставления данных TCP мастеру. В случае необходимости, можно также настроить удаленный доступ. Настройка Slave, необходимо установить параметры в следующий вид

Более подробная информация о настраиваемых полях<sup>1</sup>

- TCP Master - служит для сбора информации с TCP Slave. Можно создать несколько подключений с интересующими настройками. Общие настройки для всех подключений

- \* Enabled - on
- \* Name - имя подключения
- \* SlaveID - ID слейва с которого нужно собирать данные
- \* IP addr - адрес slave устройства
- \* Port - порт устройства
- \* Period - задержка между запросами в секундах
- \* Timeout - Максимальное время ожидания ответа



Enabled ☒ off on

Name

Slave ID

IP address

Port

Period









Timeout

После чего нужно создать сами запросы. Они создаются в соответствии с таблицей регистров. Для создания запроса необходимо

<sup>1</sup>[https://wiki.teltonika-networks.com/view/RUT955\\_Modbus](https://wiki.teltonika-networks.com/view/RUT955_Modbus)

в пункте ADD NEW REQUEST » NEW REQUEST NAME задать имя и нажать на ADD. После чего задать формат данных, функцию, номер первого регистра, кол-во регистров. Все эти данные должны быть в документации опрашиваемого устройства. [reg map TCP Slave](#)<sup>2</sup>

✓ REQUESTS CONFIGURATION

NAME	DATA TYPE	FUNCTION	FIRST REGISTER NUMBER	REGISTER COUNT / VALUES	BRACKETS	
system_uptime	32bit INT, Byte order 1,2,3,4	Read holding registers (3)	2	2	Use brackets	 
mobile_signal_strength	32bit INT, Byte order 1,2,3,4	Read holding registers (3)	4	2	Use brackets	 
system_temperature	32bit INT, Byte order 1,2,3,4	Read holding registers (3)	6	2	Use brackets	 
system_hostname	ASCII	Read holding registers (3)	8	16	Use brackets	 

- RS485
- RS232

#### • MQTT

- MQTT Gateway - шлюз между собранными данными и распределителем данных.

Для его настройки нужно просто включать. Настройки по умолчанию подходят.

SERVICES » MODBUS » MQTT Gateway. После чего полученные данные можно будет направить в MQTT Broker, для этого нужно в SERVICES » [DATA TO SERVER](#)<sup>3</sup> создать нового отправителя, после чего настроить его.

- \* Data source - MODBUS data
- \* Protocol - MQTT
- \* JSON - По примеру сделать json макет
- \* Segment count - 1
- \* Send as object- on
- \* URL|HOST|ConS - 127.0.0.1
- \* Prot - 18883

<sup>2</sup>[https://wiki.teltonika-networks.com/view/RUT955\\_Modbus](https://wiki.teltonika-networks.com/view/RUT955_Modbus)

<sup>3</sup>[https://wiki.teltonika-networks.com/view/RUT955\\_Data\\_to\\_Server](https://wiki.teltonika-networks.com/view/RUT955_Data_to_Server)



#### ADVANCED SENDER SETTINGS

Enable ☒

Name

Data source

Protocol

JSON format
 

```

      {"date": "%t", "dateDHYHMS": "%d", "modbusSlaveIp": "%p",
      "modbusSlaveId": "%i", "slaveName": "%n", "startRegNum": "%s",
      "fullRegAddr": "%z", "req_name": "%r", "data_value": "%a"}
    
```

Date (Linux timestamp) - %t	Date (binary Linux timestamp, UINT32BE) - %T
Date (Day/Month/Year Hour:Minute:Second) - %d	MODBUS slave ID - %i
MODBUS slave IP - %p	Binary Modbus slave ID (UINT8) - %l
Start register number - %s	Binary start register address (UINT16BE) - %S
Register data (JSON object) - %a	Full MODBUS register address - %z
Request name - %r	Slave name - %n

Segment count

Send as object ☒

URL / Host / Connection string

Port

Keepalive

Topic

Client ID

Period

QoS

Data filtering

Retry on fail ☐

Use TLS ☐

Use Credentials ☐

- MQTT Broker - mqtt broker.
- Для настройки брокера нужно открыть SERVICES » MQTT » MQTT Broker, где нужно
  - \* MQTT Broker - on
  - \* Local Port - 18883
  - \* Remote Access - off

MQTT Broker ☒ off on

Local Port

Enable Remote Access ☐ off on

