



**Руководство пользователя
Драйвер орсиа2mqtt**

Оглавление

1. Ключевые функции	3
1.1 Поддерживаемые возможности орсua2mqtt (клиента)	3
1.2 Функциональность OPCUA клиента	3
2. Функционал и реализация	4
2.1 Описание ключей запуска приложения	4
2.2 Описание файлов конфигурации	4
2.2.1 Описание корневого файла конфигурации	5
2.2.2 Описание раздела mqtt.tls	8
2.2.3 Переменные окружения для настройки OPCUA-серверов	8
2.2.4 Описание файла конфигурации OPCUA – серверов	9
2.2.5 Поддерживаемые скалярные типы данных	21
2.2.6 Поддерживаемые OPC UA атрибуты	21
3. Примеры файлов конфигурации YAML	22

Орсиа2mqtt — это приложение, предназначенное для взаимодействия с серверами OPC UA в роли клиента. Оно позволяет собирать данные с OPC UA серверов и публиковать их в брокере MQTT, форматировать полученные данные.

Приложение предназначено для использования в системах, где необходима интеграция различных устройств OPC UA с брокером MQTT для обработки данных в реальном времени.

Приложение представляет собой бинарный файл под Linux той или иной архитектуры процессора. Может поставляться в виде deb-пакета. Конфигурирование производится через yaml-файлы конфигурации.

1. Ключевые функции

Приложение поддерживает следующие функции:

- Периодическое чтение данных из OPC UA-сервера;
- Подписка на изменение данных на OPC UA-сервере;
- Событийная публикация данных;
- Отправка считанных данных в виде MQTT-топиков в JSON формате в MQTT-брокер;
- Управление орсиа-устройством через управляющие mqtt-топики;
- Гибкое конфигурирование посредством YAML-файлов;
- Загрузка файлов конфигурации устройств относительно расположения исполняемого файла.

1.1 Поддерживаемые возможности орсиа2mqtt (клиента)

Категория	Возможность	Поддержка	Примечание
Encoding	OPC UA Binary	Да	основной режим
Transport	UA-TCP / UA-SC / UA Binary	Да	стандартный протокол
Encryption	None	Да	без шифрования
Encryption	Basic128Rsa15	Да	поддерживается
Encryption	Basic256	Да	поддерживается
Encryption	Basic256Sha256	Да	поддерживается
Encryption	Aes128Sha256RsaOaep	Да	поддерживается
Encryption	Aes256Sha256RsaPss	Да	поддерживается
Auth	Anonymous	Да	без логина/пароля
Auth	Username + Password	Да	стандартная схема
Auth	X.509 Certificate	Да	клиентский сертификат + ключ

1.2 Функциональность OPCUA клиента

Категория	Возможность	Описание
Безопасность	Шифрование каналов	Basic128Rsa15 / Basic256 / Basic256Sha256
Режимы защиты	Sign / SignAndEncrypt	Настраиваются через YAML

Чтение	Периодическое чтение по Nodeld	Чтение значений и атрибутов (Value, DisplayName, Description, DataType, ...)
Чтение по апертуре	Получение измененных данных от OPC UA-сервера	Настраивается через YAML
Запись	Установка значения по MQTT-топику	Запись значений в атрибут Value
Подписки	Создание подписок на узлы	Сервер сам присылает изменения

2. Функционал и реализация

2.1 Описание ключей запуска приложения

Для вызова справки по ключам (аргументам командной строки), поддерживаемым приложением можно использовать **--help**:

```
andromeda-user@host-linux:~$ ./opcua2mqtt_x.x.x.x_linux_amd64 --help
Usage of ./opcua2mqtt_x.x.x.x_linux_amd64:
  -c, --cfgpath string    Путь до файла конфигурации (default "configs/opcua2mqtt/opcua2mqtt.yaml")
  --version               Вывести версию Opсua2Mqtt
  -l, --level_log int     Уровень логирования Opсua2Mqtt (1-4) (default 1)
  -p, --path_log string   Путь до файла логов (если путь не указан, то запись в файл не производится)
```

Если ключ **-c, --cfgpath** не будет передан при запуске, то драйвер будет искать файл конфигурации в пути по умолчанию "configs/opcuat2mqtt/opcua2mqtt.yaml".

Ключи, связанные с логами, могут быть заданы в файле конфигурации, который передается через ключи **-c, --cfgpath**.

ВАЖНО!

Настройки применяются в порядке приоритета (от наивысшего к низшему):

1. Аргумент командной строки;
2. Параметр в файле конфигурации;
3. Значение по умолчанию.

2.2 Описание файлов конфигурации

Конфигурация драйвера задается через параметры командной строки, как описано выше, и в нескольких YAML-файлах: один корневой и несколько файлов конфигурации конкретных OPCUA-серверов. Корневой файл конфигурации содержит общие настройки драйвера, такие как логирование, адрес MQTT-брокера и пути к файлам конфигурации опрашиваемых OPCUA-серверов. Путь к нему как раз указывается через ключ **-c, --cfgpath**. Файлы конфигурации OPCUA-серверов содержат настройки, необходимые для их опроса, подписки и управления. Описание корневого файла конфигурации

2.2.1 Описание корневого файла конфигурации

Содержимое файла конфигурации выглядит следующим образом:

```
# =====
# Версия: 1.0.0.0
# =====
орсуа:
  enabled: true #Включение драйвера; boolean true|false; [ОБЯЗАТЕЛЬНЫЙ]

  mqtt:
    broker: "tcp://127.0.0.1:1883" #Адрес MQTT брокера; string URL tcp|ssl|unix;
[ОБЯЗАТЕЛЬНЫЙ]; По умолчанию: tcp://localhost:1883
    client_id: "орсуа2mqtt" #MQTT client_id; string; [НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию:
орсуа2mqtt
    max_retries: 5000 #Макс. попыток подключения; int >=0 (0=бесконечно);
[НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию: 5000
    retry_interval: 5 #Интервал переподключения (сек); int >=1; [НЕОБЯЗАТЕЛЬНЫЙ]; По
умолчанию: 5
    clean_session: true #Чистая MQTT-сессия; boolean; [НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию: true
    autoreconnect: true #Автопереподключение; boolean; [НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию:
true
    keepalive: 30 #MQTT keepalive (сек); int [1..240]; [НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию: 30
    ping_timeout: 10 #Таймаут ответа keepalive (сек); int [1..60]; [НЕОБЯЗАТЕЛЬНЫЙ]; По
умолчанию: 10
    connect_timeout: 30 #Таймаут первого соединения (сек); int [1..60]; [НЕОБЯЗАТЕЛЬНЫЙ];
По умолчанию: 30
    pubQoS: 0 #MQTT QoS (0|1|2); int; [НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию: 0
    retained: false #Retained-сообщения; boolean; [НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию: false
    # username: "mqtt_user" #Логин MQTT; string; [НЕОБЯЗАТЕЛЬНЫЙ]
    # password: "mqtt_password" #Пароль MQTT; string; [НЕОБЯЗАТЕЛЬНЫЙ]

    # tls:
    #   enabled: true #Включение TLS; boolean; [ОБЯЗАТЕЛЬНЫЙ при наличии секции]
    #   ca_cert: "certs/mqtt/ca.pem" #CA сертификат; string path; [ОБЯЗАТЕЛЬНЫЙ если
insecure_skip_verify=false]
    #   cert_file: "certs/mqtt/client.pem" #Клиентский сертификат; string path;
[НЕОБЯЗАТЕЛЬНЫЙ]
    #   key_file: "certs/mqtt/client_key.pem" #Клиентский ключ; string path;
[НЕОБЯЗАТЕЛЬНЫЙ]
    #   insecure_skip_verify: false #Отключение проверки TLS; boolean; [НЕОБЯЗАТЕЛЬНЫЙ];
По умолчанию: false

  logger:
    level_log: 1 #Уровень логирования (1=INFO,2=WARN,3=DEBUG,4=ERROR); int;
[НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию: 1
    path_log: "logs/орсуа2mqtt.log" #Файл логов; string path; [НЕОБЯЗАТЕЛЬНЫЙ]
    path_json_log: "logs/орсуа2mqtt.json" #JSON-лог; string path; [НЕОБЯЗАТЕЛЬНЫЙ]
    max_size_log: 5 #Макс. размер лога (МБ); int [1..5]; [НЕОБЯЗАТЕЛЬНЫЙ]; По умолчанию: 5
```

```

max_backups_log: 100 #Кол-во резервных логов; int [1..100]; [НЕОБЯЗАТЕЛЬНЫЙ]; По
умолчанию: 100
max_age_log: 30 #Максимальный возраст лог-файлов (дни); int [1..30]; [НЕОБЯЗАТЕЛЬНЫЙ];
По умолчанию: 30

devices:
  device_1:
    file: "орсуа_server_1.yaml" #Файл конфигурации OPC UA устройства; string path;
[ОБЯЗАТЕЛЬНЫЙ]
  device_2:
    file: "орсуа_server_2.yaml" #Файл конфигурации OPC UA устройства; string path;
[ОБЯЗАТЕЛЬНЫЙ]

```

Раздел `mqtt` описывает настройки подключения к mqtt-брокеру. Все поля являются необязательными и могут не указываться, в таком случае будут приняты значения по умолчанию;

Раздел `logger` описывает настройки логирования работы приложения. Все поля являются необязательными и могут не указываться, в таком случае будут приняты значения по умолчанию; Примечание: настройки параметров `level_log` и `path_log` доступны через аргументы командной строки. Запись логов в файл происходит только в том случае, если был указан путь логов.

Раздел `devices` описывает условные имена опрашиваемых орсуа-устройств и пути к их файлам конфигурации. Имена устройств должны быть уникальными. Относительные пути к файлам устройств указываются относительно текущего файла конфигурации, а не рабочей директории.

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
<code>enabled</code>	Режим работы	Обязательный	Служит для включения, отключения работы драйвера	<code>true, false</code>	—
<code>mqtt.broker</code>	MQTT-брокер	Обязательный, если задан раздел <code>mqtt</code>	Адрес MQTT-брокера, на который будут отправляться топики с данными из орсуа-устройств.	<code>tcp://, unix://, sst://</code> Примеры: <code>tcp://localhost:1883</code> <code>unix:///var/run/mosquitto/mosquitto.sock</code>	127.0.0.1:1883
<code>mqtt.client_id</code>	Идентификатор клиента	Необязательный	Значение <code>client_id</code> , с которым драйвер подключается к MQTT-брокеру.	Строка	<code>орсуа2mqtt</code>
<code>mqtt.max_retries</code>	Количество повторов попыток подключения к брокеру	Необязательный	Общее количество попыток подключения к MQTT-брокеру. В случае, если число попыток превысит <code>max_retries</code> , драйвер завершится с соответствующей ошибкой.	<code>int32</code> , которое строго больше 0	5000
<code>mqtt.retry_interval</code>	Интервал между попытками подключения к брокеру	Необязательный	Время в секундах, которое должно пройти перед тем, как будет совершена очередная попытка подключения к MQTT-брокеру.	<code>int32</code> , которое строго больше 1	5
<code>mqtt.clean_session</code>	Чистая сессия с брокером	Необязательный	Флаг очистки сессии. Если <code>true</code> , брокер удаляет данные о подписках и сообщениях после отключения клиента, если <code>false</code> — сохраняет.	<code>true, false</code>	<code>true</code>
<code>mqtt.autoreconnect</code>	Переподключение к брокеру при	Необязательный	Флаг, определяющий переподключение к MQTT-брокеру при потере уже установленного соединения.	<code>true, false</code>	<code>true</code>

	потере соединения				
mqtt.Keepalive	Интервал отправки служебных сообщений протокола MQTT для мониторинга состояния соединения	Необязательный	Интервал времени (в секундах), в течение которого драйвер должен обмениваться сообщениями с MQTT-брокером, чтобы подтвердить, что соединение остается активным.	int32 в диапазоне: [1;240]	30
mqtt.ping_timeout	Таймаут ожидания ответа на сообщение keepalive	Необязательный	Время ожидания ответа на сообщение keepalive. В секундах.	int32 в диапазоне: [1;60]	10
mqtt.connect_timeout	Время ожидания установления первого соединения с брокером	Необязательный	Время ожидания установления первого соединения с брокером. В секундах.	int32 в диапазоне: [1;60]	30
mqtt.publish_qos	Уровень гарантии доставки сообщения	Необязательный	QoS (Quality of Service) - гарантии доставки сообщений: QoS 0 ("хоть бы раз", доставка не гарантируется), QoS 1 (доставка "как минимум один раз", возможны дубли) и QoS 2 ("ровно один раз", доставка без дубликатов)	byte из: 0, 1, 2	0
mqtt.retained	Сохранение топика в брокере	Необязательный	Флаг, если установлен в true - брокер сохраняет последнее отправленное сообщение топика и доставляет его любому новому подписчику, даже если он подключился позже.	true, false	false
mqtt.username	Логин	Необязательный	Логин для подключения к MQTT-брокеру	Строка	—
mqtt.password	Пароль	Необязательный. При наличии mqtt.username - обязательный.	Пароль для подключения к MQTT-брокеру	Строка	—
mqtt.tls	Настройки безопасности	Необязательный	Настройки безопасного подключения к MQTT-брокеру.	п.2.2.2	—
logger.level_log	Уровень логирования	Необязательный	Уровень логирования.	int32, 1-4 (1-info, 2-warn, 3-debug, 4-error).	1
logger.path_log	Путь к лог-файлу	Необязательный	Путь к файлу логов. Если путь указан - лог пишется в файл. Иначе - без записи логов в файл.	Строка	—
logger.path_json_log	Путь к json лог-файлу	Необязательный	Путь к файлу логов JSON. Если путь указан - лог пишется в JSON файл. Иначе - без записи JSON логов в файл.	Строка	—
logger.max_size_log	Размер лог-файла	Необязательный	Максимальный размер лог-файла в мегабайтах.	1-5	5
logger.max_backups_log	Количество резервных копий	Необязательный	Максимальное количество резервных копий логов.	1-100	100
logger.max_age_log	Возраст логов	Необязательный	Максимальный возраст лог-файлов в днях.	1-30	30
devices	Конфигурация и устройств	Обязательный	Описывает условные имена опрашиваемых OPC UA-серверов и пути к их файлам конфигурации. Имена устройств должны быть уникальными. В случае совпадения имен, конфигурация будет использоваться для последнего по порядку заданного устройства(сервера).	—	—

2.2.2 Описание раздела mqtt.tls

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
enabled	Режим работы	Обязательный, если задан данный раздел	Служит для включения, отключения безопасного подключения к MQTT-брокеру.	true, false	—
mqtt.tls.ca_cert	Сертификат CA	Обязательный, если mqtt.tls.insecure_skip_verify задан false	Путь к файлу корневого сертификата удостоверяющего центра	Строка	""
mqtt.tls.cert_file	Клиентский сертификат	Обязательный, если указан mqtt.tls.key_file	Путь к файлу клиентского сертификата для взаимной аутентификации	Строка	""
mqtt.tls.insecure_skip_verify	Пропуск проверки сертификата	Обязательный	Если задан true, то полностью отключает проверку подлинности сертификата сервера, позволяя приложению устанавливать соединение даже с самоподписанными, просроченными или недействительными сертификатами.	true, false	false
mqtt.tls.key_file	Клиентский ключ	Обязательный, если указан mqtt.tls.cert_file	Путь к файлу приватного ключа клиента	Строка	""

2.2.3 Переменные окружения для настройки OPCUA-серверов

Для безопасного хранения секретов (учётных данных OPC UA-устройства для auth_type: UserName) используется ввод через файл переменных окружения (ENV). Перед стартом драйвера необходимо обеспечить инициализацию этих переменных: при ручном запуске — с помощью команды source path/to/.env, а при использовании systemd — добавив параметр EnvironmentFile=/path/to/.env в файл сервиса opcua2mqtt.service

Названия переменных окружения для конкретного OPC UA-устройства должны подчиняться строгим правилам наименования:

- OPCUA_DEVICE_<DEVICE_NAME>_USERNAME
- OPCUA_DEVICE_<DEVICE_NAME>_PASSWORD

При этом, <DEVICE_NAME> является device.name в верхнем регистре.

Приоритет конфигурирования: ENV → затем YAML

Пример файла с настройками /mnt/data/etc/andromeda_embedded/configs/opcua2mqtt/.opcua2mqtt.env с правами 600 root:root

```
OPCUA_DEVICE_OPCUA_EXAMPLE_SERVER_USERNAME=""
OPCUA_DEVICE_OPCUA_EXAMPLE_SERVER_PASSWORD=""
```

2.2.4 Описание файла конфигурации OPCUA – серверов

Содержимое файла конфигурации выглядит следующим образом:

```
# =====
# Версия: 1.0.0.0
# =====

# -----
# СЕКЦИЯ: device - Информация об устройстве
# -----
device:
  enabled: true # [НЕОБЯЗАТЕЛЬНЫЙ] Включение/выключение
устройства, по умолчанию: true
  name: "PLC_Device_001" # [ОБЯЗАТЕЛЬНЫЙ] Уникальное имя
устройства
  root_topic_mqtt: "/devices/plc001" # [ОБЯЗАТЕЛЬНЫЙ] Корневой MQTT топик для
публикации данных, без wildcards

# -----
# СЕКЦИЯ: connection - Настройки подключения к OPC UA серверу
# -----
connection:
  endpoint: "opc.tcp://192.168.1.100:4840" # [ОБЯЗАТЕЛЬНЫЙ] URL OPC UA сервера

  # Таймауты (секунды)
  dial_timeout: 10 # [НЕОБЯЗАТЕЛЬНЫЙ] Таймаут установки
TCP-соединения (1-30), по умолчанию: 10
  request_timeout: 30 # [НЕОБЯЗАТЕЛЬНЫЙ] Таймаут запросов к
серверу (1-60), по умолчанию: 30
  session_timeout: 300 # [НЕОБЯЗАТЕЛЬНЫЙ] Время жизни сессии
при отсутствии активности (60-3600), по умолчанию: 300
  retry_interval: 10 # [НЕОБЯЗАТЕЛЬНЫЙ] Интервал между
попытками первого подключения (1-60), по умолчанию: 10
  reconnect_interval: 5 # [НЕОБЯЗАТЕЛЬНЫЙ] Интервал между
попытками переподключения (1-60), по умолчанию: 5
  max_retry_attempts: 0 # [НЕОБЯЗАТЕЛЬНЫЙ] Максимальное
количество попыток подключения (0 = бесконечно), по умолчанию: 0
  lifetime: 300 # [НЕОБЯЗАТЕЛЬНЫЙ] Время жизни токена
для secure channel (60-3600), по умолчанию: 300

  # Безопасность
  security_mode: "None" # [ОБЯЗАТЕЛЬНЫЙ] Режим безопасности:
None | Sign | SignAndEncrypt
  security_policy: "None" # [ОБЯЗАТЕЛЬНЫЙ] Политика: None |
Basic128Rsa15 | Basic256 | Basic256Sha256 | Aes128Sha256RsaOaep | Aes256Sha256RsaPss

  # Аутентификация
  auth_type: "Anonymous" # [ОБЯЗАТЕЛЬНЫЙ] Тип: Anonymous |
UserName | Certificate
```

```
auth_policy_id: "anonymous" # [ОБЯЗАТЕЛЬНЫЙ] ID политики
аутентификации на сервере (зависит от конфигурации OPC UA сервера)
auth_security_policy: "None" # [НЕОБЯЗАТЕЛЬНЫЙ] Политика безопасности
для аутентификации, если не указано - берется из security_policy

# Учетные данные (для auth_type: UserName)
# username: "opcua_user" # [НЕОБЯЗАТЕЛЬНЫЙ] Логин, обязателен
если auth_type = UserName
# password: "opcua_pass" # [НЕОБЯЗАТЕЛЬНЫЙ] Пароль, обязателен
если auth_type = UserName

# Идентификация приложения
application_name: "opcua2mqtt_client" # [НЕОБЯЗАТЕЛЬНЫЙ] Имя приложения
application_uri: "urn:opcua2mqtt:client" # [НЕОБЯЗАТЕЛЬНЫЙ] URI приложения
product_uri: "urn:opcua2mqtt" # [НЕОБЯЗАТЕЛЬНЫЙ] URI продукта

# Сертификаты клиента (для security_mode: Sign/SignAndEncrypt)
# client_cert_file: "certs/client_cert.pem" # [НЕОБЯЗАТЕЛЬНЫЙ] Путь к клиентскому
сертификату, обязателен если security_mode != None
# client_key_file: "certs/client_key.pem" # [НЕОБЯЗАТЕЛЬНЫЙ] Путь к приватному
ключу клиента, обязателен если security_mode != None

# Сертификаты пользователя (для auth_type: Certificate)
# user_cert_file: "certs/user_cert.pem" # [НЕОБЯЗАТЕЛЬНЫЙ] Путь к сертификату
пользователя, обязателен если auth_type = Certificate
# user_key_file: "certs/user_key.pem" # [НЕОБЯЗАТЕЛЬНЫЙ] Путь к ключу
пользователя, обязателен если auth_type = Certificate

# Сертификат сервера (для верификации)
# server_cert_file: "certs/server_cert.pem" # [НЕОБЯЗАТЕЛЬНЫЙ] Путь к серверному
сертификату для проверки подлинности

# -----
# СЕКЦИЯ: data_model - Модель данных (чтение и запись)
# -----
data_model:

# -----
# ПОДСЕКЦИЯ: read - Конфигурация чтения данных
# -----
read:

# -----
# poll - Группы периодического опроса
# -----
poll:
  - read_period_ms: 1000 # [ОБЯЗАТЕЛЬНЫЙ] Период опроса в
миллисекундах (1-3600000)
    timestamps_to_return: "Source" # [НЕОБЯЗАТЕЛЬНЫЙ] Метки времени: Source
| Server | App, по умолчанию: Source
```

```
parameters:
  - parameter_name: "Temperature" # [ОБЯЗАТЕЛЬНЫЙ] Уникальное имя
параметра для MQTT топика, без wildcards
    precision: 2 # [НЕОБЯЗАТЕЛЬНЫЙ] Округление (-1 = без
округления, 0-10), по умолчанию: -1
    transform: "0.1:0" # [НЕОБЯЗАТЕЛЬНЫЙ] Трансформация: "a:b"
-> y=a*x+b, по умолчанию: "1:0"
  node:
    node_id: "ns=2;s=Sensors.Temperature" # [ОБЯЗАТЕЛЬНЫЙ] NodeID узла в OPC UA
    attribute: "Value" # [НЕОБЯЗАТЕЛЬНЫЙ] Атрибут узла: Value |
DataType | DisplayName и т.д., по умолчанию: Value
    index_range: "0" # [НЕОБЯЗАТЕЛЬНЫЙ] Диапазон индексов для
массивов, например: "0" или "0:9"
    quality_filter:
      good_only: false # [НЕОБЯЗАТЕЛЬНЫЙ] Фильтр качества: true
= только Good статус, по умолчанию: false

  - parameter_name: "Pressure"
    node:
      node_id: "ns=2;s=Sensors.Pressure"

- read_period_ms: 5000 # Вторая poll-группа с другим периодом
опроса
parameters:
  - parameter_name: "Status"
    node:
      node_id: "ns=2;s=Device.Status"

# -----
# subscribe - Группы подписки на изменения
# -----
subscribe:
  - publishing_interval_ms: 1000 # [ОБЯЗАТЕЛЬНЫЙ] Интервал публикации (0
= минимальный на сервере, 0-3600000)
    timestamps_to_return: "Source" # [НЕОБЯЗАТЕЛЬНЫЙ] Метки времени: Source
| Server | App, по умолчанию: Source
    lifetime_count: 30 # [НЕОБЯЗАТЕЛЬНЫЙ] Время жизни подписки
в количествах публикаций (0 = по умолчанию 3*keep_alive_count)
    keep_alive_count: 10 # [НЕОБЯЗАТЕЛЬНЫЙ] Keep-alive каждые N
публикаций (0-100), по умолчанию: 10
    max_notifications_per_publish: 0 # [НЕОБЯЗАТЕЛЬНЫЙ] Макс. уведомлений на
публикацию (0 = без ограничений, 0-1000), по умолчанию: 0
    global_pub_time_s: 300 # [НЕОБЯЗАТЕЛЬНЫЙ] Периодический опрос
всех параметров независимо от фильтров (0 = отключено, 0-14400), по умолчанию: 300
  parameters:
    - parameter_name: "AlarmCounter" # [ОБЯЗАТЕЛЬНЫЙ] Имя параметра
      precision: 0 # [НЕОБЯЗАТЕЛЬНЫЙ] Без десятичных знаков
      node:
        node_id: "ns=2;s=Alarms.Counter"
      quality_filter:
```

```
    good_only: true # Публиковать только при Good статусе
  monitoring:
    sampling_interval_ms: 500 # Частый сэмплинг для алармов
    queue_size: 10 # Большая очередь
    discard_oldest: false # Не отбрасывать старые значения
    filter:
      trigger: "StatusValueTimestamp" # Триггер по статусу, значению и времени
    aperture:
      deadband_type: "none" # Без deadband - публиковать каждое
изменение

- parameter_name: "FlowRate"
  precision: 3
  transform: "0.001:0" # Преобразование из мл/с в л/с
  node:
    node_id: "ns=2;s=Process.FlowRate"
  monitoring:
    sampling_interval_ms: 1000
    filter:
      trigger: "StatusValue"
    aperture:
      deadband_type: "percent" # Процентный deadband
      deadband_value: 2.0 # Публиковать при изменении >2%

# -----
# ПОДСЕКЦИЯ: write - Конфигурация записи данных
# -----
write:
- parameter_name: "SetPoint" # [ОБЯЗАТЕЛЬНЫЙ] Имя параметра для MQTT
команды, без wildcards
  command_topic: "on" # [НЕОБЯЗАТЕЛЬНЫЙ] Суффикс командного
топика, по умолчанию: "on"
  node:
    node_id: "ns=2;s=Controls.SetPoint" # [ОБЯЗАТЕЛЬНЫЙ] NodeID узла для записи
    data_type: "Double" # [ОБЯЗАТЕЛЬНЫЙ] Тип данных: Boolean |
SByte | Byte | Int16 | UInt16 | Int32 | UInt32 | Int64 | UInt64 | Float | Double | String
| DateTime | ByteString
    index_range: "" # [НЕОБЯЗАТЕЛЬНЫЙ] Диапазон индексов для
массивов
  on_write_feedback:
    immediate_readback: true # [НЕОБЯЗАТЕЛЬНЫЙ] Немедленное чтение
после записи для подтверждения, по умолчанию: true

- parameter_name: "Enable"
  command_topic: "set" # Команда будет приходить в топик:
/devices/plc001/controls/Enable/set
  node:
    node_id: "ns=2;s=Controls.Enable"
    data_type: "Boolean"
  on_write_feedback:
```

```
immediate_readback: true
```

```
- parameter_name: "ConfigArray"
  node:
    node_id: "ns=2;s=Config.Array"
    data_type: "Int32"
    index_range: "0:9"
```

Запись в элементы массива 0-9

2.2.4.1 Раздел device

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
enabled	Включено	Необязательный	Флаг включения/отключения опроса устройства.	true, false	true
name	Имя устройства	Обязательный	Логическое имя устройства, используется в логах, валидации. Должно быть уникально.	Строка	—
root_topic_mqtt	Корневой топик MQTT	Обязательный	Часть полного MQTT-топика в структуре /devices/\${root_topic_mqtt}/controls/\${parameter_name}.	Строка, без символов '+' и '#'	—

2.2.4.2 Раздел connection

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
endpoint	Адрес	Обязательный	Адрес, через который клиентское приложение взаимодействует с сервером или сервисом для отправки запросов и получения данных.	Строка. Например: opc.tcp://DevMac.local:53530/OPCUASimulationServer	—
dial_timeout	Таймаут установки соединения	Необязательный	Таймаут установки TCP-соединения с OPCUA сервером. Задается в секундах.	int32 в диапазоне: [1;30]	10
request_timeout	Таймаут запросов	Необязательный	Таймаут запросов к серверу. Задается в секундах.	int32 в диапазоне: [1;60]	30
session_timeout	Таймаут сессии	Необязательный	Время жизни сессии в секундах, в течение которой она считается открытой при отсутствии активности со стороны клиента(opcua2mqtt).	int32 в диапазоне: [60;3600]	300
retry_interval	Интервал между попытками	Необязательный	Интервал между попытками первого подключения к серверу. Задается в секундах.	int32 в диапазоне: [1;60]	10
max_retry_attempts	Количество попыток подключения	Необязательный	Максимальное количество попыток подключения к серверу, после которых приложение останавливается.	int32 >=0; 0 - бесконечно	0
lifetime	Время жизни токена для secure channel	Необязательный	Таймаут действия токена защищенного канала. В случае, если клиент не обновляет токен на сервере до его истечения, происходит разрыв сессии и защищенного канала.	int32 в диапазоне: [60;3600]	300
reconnect_interval	Интервал между попытками	Необязательный	Интервал между попытками переподключения.	int32 в диапазоне: [1;60]	5

	переключе ния				
security_mode	Режим безопасности	Обязательный	Режим безопасности. Отвечает за защиту сообщений на уровне Secure Channel	<p>Одно из следующих значений:</p> <p>None - без подписи и шифрования</p> <p>Sign - сообщения подписываются (защита целостности и подлинности, но не секретности)</p> <p>SignAndEncrypt - сообщения подписываются и шифруются (целостность + конфиденциальность)</p>	—
security_policy	Политика безопасности	Обязательный	Политики безопасности, определяет каким способом канал между клиентом(орсua2mqtt) и сервером будет шифроваться, подписываться (для целостности) и аутентифицироваться.	<p>Одно из следующих значений:</p> <p>None</p> <p>Basic128Rsa15</p> <p>Basic256</p> <p>Basic256Sha256</p> <p>Aes128Sha256RsaOaep</p> <p>Aes256Sha256RsaPss</p>	—
auth_type	Тип пользовательской аутентификации	Обязательный	Определяет тип пользовательской аутентификации.	<p>Одно из следующих значений:</p> <p>Anonymous - без логина/пароля/сертификата</p> <p>UserName - логин/пароль</p> <p>Certificate - пользовательский сертификат</p>	—
auth_policy_id	Строковый идентификатор	Обязательный	Строковый идентификатор политики аутентификации пользователя (UserTokenPolicy), который задаётся на стороне сервера.	<p>Строка</p> <p>Примеры для Prosys OPC UA</p>	—

	политики аутентификации пользователя			сервера: - "anonymous" - "certificate_basic256" - "certificate_basic256sha256" - "username_basic256" - "username_basic256sha256" Значение зависит от конфигурации конкретного OPC UA сервера.	
auth_security_policy	Пользовательская политика безопасности	Обязательный	Пользовательская политика безопасности для защиты пользовательского токена (логин/пароль или сертификат пользователя), т.е. не для транспорта	Одно из следующих значений: None Basic128Rsa15 Basic256 Basic256Sha256 Aes128Sha256RsaOaep Aes256Sha256RsaPss	Если не указан, берется из security_policy
username	Имя пользователя для подключения к OPC UA серверу	Обязательный при auth_type = UserName (из YAML или ENV), иначе - необязательный	Логин для аутентификации на OPC UA-сервере. Если auth_type = UserName , должно быть задано либо в файле конфигурации, либо через ENV (OPCUA_DEVICE_<NAME>_USERNAME) .	Строка	—
password	Пароль пользователя для подключения к OPC UA серверу	Обязательный при auth_type = UserName (из YAML или ENV)	Пароль для username. При auth_type = UserName обязателен (файл конфигурации или ENV OPCUA_DEVICE_<NAME>_PASSWORD). Рекомендуется задавать только через ENV, YAML — только для теста.	Строка	—
application_name	Имя OPC UA-клиента	Необязательный	Читаемое имя клиента, которое видит OPC UA-сервер в своих сессиях/логах.	Строка	""
application_uri	URI приложения	Необязательный	Уникальный идентификатор клиента в формате URI(urn:...). Используется сервером для идентификации приложения;	Строка	""
product_uri	URI продукта	Необязательный	Идентификатор продукта/ПО (urn:opcua2mqtt). Нужен в основном для "визитки" клиента на сервере (диагностика, аудит).	Строка	""
client_certificate	Сертификат	Обязательный при	Путь к сертификату OPC UA-клиента (драйвера). Нужен, когда шифруется/подписывается транспорт	Строка	—

	приложение (клиента)	security_mode = Sign или SignAndEncrypt , иначе - необязательный	(Secure Channel). При Sign/SignAndEncrypt должен быть задан вместе с client_key_file ; при None может быть пустым.		
client_key_file	Приватный ключ приложения	Обязательный при security_mode = Sign или SignAndEncrypt , иначе - необязательный	Путь к приватному ключу, соответствующему client_cert_file . Используется для установления защищённого канала. Обязателен при Sign/SignAndEncrypt ; при None не нужен.	Строка	—
user_cert_file	Сертификат пользователя	Обязательный при auth_type = Certificate , иначе - необязательный	Сертификат "пользователя" для аутентификации по сертификату (auth_type = Certificate). Используется на уровне идентификации пользователя.	Строка	—
user_key_file	Приватный ключ пользователя	Обязательный при auth_type = Certificate , иначе - необязательный	Приватный ключ, соответствующий user_cert_file . Нужен только при auth_type = Certificate .	Строка	—
server_cert_file	Сертификат сервера OPC UA	Обязательный, если security_mode не None или auth_type не Anonymous , иначе - необязательный	Сертификат OPC UA-сервера.	Строка	—

2.2.4.3 Особенности конфигурирования режима безопасности и аутентификации

В связи с особенностью работы OPC UA протокола, существуют следующие особенности конфигурирования:

- Если **security_mode** - "None", то **security_policy** обязана быть "None";
- Если **security_mode** - не "None", то **security_policy** не может быть "None";
- Если **security_mode** не "None", то **client_cert_file** и **client_key_file** должны быть указаны;
- **security_policy** должен строго соответствовать конфигурации используемого endpoint'a сервера;
- Если **auth_type** - "UserName", то требуется **username** и **password** (из конфига или из ENV);
- Если **auth_type** - "Certificate", то требуется **user_cert_file** и **user_key_file**, а также **server_cert_file** (для проверки сервера).

2.2.4.4 Раздел **data_model.read**

Содержит в себе два подраздела: **subscribe** (для подписки на изменение параметров внутри OPC UA сервера) и **poll** (для периодического чтения параметров орсua-устройства):

Раздел **subscribe** структурно представляет собой группы сигналов, объединенные по значению их периода публикации, т.е. одна группа сигналов публикуется по периоду 1000ms, вторая группа по периоду 2000ms и тд. Должна быть указана минимум одна группа. Каждая группа имеет следующие поля:

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
publishing_interval_ms	Интервал публикации	Обязательный	Интервал публикации группы со стороны OPCUA сервера. Задается в миллисекундах.	Целое число в диапазоне: [0; 3600000]	—

timestamps_to_return	Источник временной метки	Необязательный	Источник временной метки при насыщении payload мета-информацией.	Одно из следующих значений: Source - время из источника; Server - время сервера; App - время ОС, на которой запущен драйвер(орсуа2 mqtt)	Source
lifetime_count	Время жизни подписки (по количеству публикаций)	Необязательный	Количество публикаций, после которых подписка считается “устаревшей”.	int32 >=0; 0 - бесконечно	Если не указано, но есть keep_alive_count — автоматически ставится 3 * keep_alive_count
keep_alive_count	Интервал keep-alive	Необязательный	Каждые N публикаций отправлять keep-alive сообщение серверу.	int32 в диапазоне: [0; 100].	10
max_notifications_per_publish	Лимит уведомлений на публикацию	Необязательный	Максимальное количество уведомлений (data change notifications), которое сервер может включить в один пакет публикации (Publish).	int32 в диапазоне: [0; 1000]; 0 - бесконечно	0
global_pub_time_s	Интервал периодического опроса всех параметров	Необязательный	Интервал (в секундах) периодического опроса всех параметров группы независимо от фильтров subscribe. Используется обычный poll	int32 в диапазоне: [0; 14400]; 0 - отключено	300
parameters	Список параметров	Обязательный	Список параметров орсуа-устройства, которые будут опрашиваться, либо на которые будет совершена подписка с указанным периодом и публиковаться в MQTT-топики. Каждый параметр имеет собственную структуру. Подробнее смотреть далее в п.2.2.4.5	Подробнее смотреть далее в п.2.2.4.5	—

Раздел poll структурно представляет собой группы сигналов, объединенные по значению их периода опроса, т.е. одна группа сигналов опрашивается по периоду 1000ms, вторая группа по периоду 2000ms и тд. Необходимо указать минимум одну группу для опроса. Каждая группа имеет следующие поля:

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
read_period_ms	Интервал опроса	Обязательный	Интервал опроса группы. Задается в миллисекундах.	int32 в диапазоне: [1; 3600000]	—
timestamps_to_return	Источник временной метки	Необязательный	Источник временной метки при насыщении payload мета-информацией.	Одно из следующих значений:	Source

				Source - время из источника; Server - время сервера; App - время ОС, на которой запущен драйвер(орсуа2 mqtt)	
parameters	Список параметров	Обязательный	Список параметров орсуа-устройства, которые будут опрашиваться, либо на которые будет совершена подписка с указанным периодом и публиковаться в MQTT-топики. Каждый параметр имеет собственную структуру. Подробнее смотреть далее в п.2.2.4.5	Подробнее смотреть далее в п.2.2.4.5	—

2.2.4.5 Раздел parameters

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
parameter_name	Имя параметра	Обязательный	Имя параметра. Используется в структуре топика /devices/{root_topic_mqtt}/controls/{parameter_name}/{command_topic}. Если значение parameter_name начинается с /devices/, то root_topic_mqtt не используется, а итоговый топик формируется как \${parameter_name}/{command_topic}.	Строка. Не должно содержать символы + и #	—
precision	Точность округления	Необязательный	Кол-во знаков после запятой; Не применяется к строковым и булевым типам данных. Значение '-1' отключает функцию округления. Значение будет передано с тем количеством знаком, с каким получено из сервера.	-1:10	-1
transform	Формула преобразования	Необязательный	Формула для преобразования значения $y = (a * x + b)$; x - исходное значение, y - преобразованное значение для записи, Формат: "a:b" (например, "0.1:0" означает $y = 0.1 * x + 0$) По умолчанию: "1:0" (без преобразования)	—	1:0
node.node_id	Идентификатор узла OPC UA	Обязательный	Строковый идентификатор узла на OPC UA-сервере. В рамках раздела data_model.write должен быть уникальным.	Строка вида: "ns=x;i=y", где ns - namespace(пространство имен), y - номер узла в указанном пространстве имен Например: "ns=3;i=1609", "ns=2;s=Motor.RunCmd"	—
node.attribute	Атрибут узла	Необязательный	Атрибут OPC UA для чтения. Допускаются только поддерживаемые атрибуты, которые описаны в пункте 2.2.6	Любое значение из описанных в пункте 2.2.6	Value
quality_filter.good_only	Только Good-качество	Необязательный	Если true, публикуются только значения с хорошим статусом качества (Good);	true, false	false

monitoring	Параметры мониторинга (только для раздела subscribe)	Необязательный	Подробнее смотреть далее в п. 2.2.4.6	Подробнее смотреть далее в п. 2.2.4.6	—
------------	--	----------------	---------------------------------------	---------------------------------------	---

2.2.4.6 Раздел parameters.monitoring

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
sampling_interval_ms	Интервал выборки	Необязательный	Определяет частоту, с которой сервер измеряет значение для узлов и атрибутов, на которые подписан драйвер;	int32 в диапазоне: [0; 3600000]. 0 — решение принимает сервер.	1000
queue_size	Размер очереди значений	Необязательный	Максимальное число значений, хранимых в очереди на сервере до отправки.	int32 в диапазоне: [1; 100].	1
discard_oldest	Отбрасывать старые значения	Необязательный	Если true, при переполнении очереди новые значения вытесняют старые; если false, новые отбрасываются.	true, false	true
filter.trigger	Тип триггера изменения данных.	Необязательный	Определяет, когда сервер будет отправлять значение.	Одно из следующих значений: Status - при изменении статуса качества StatusValue - при изменении статуса или значения StatusValueTimestamp - при изменении статуса, значения или метки времени.	StatusValue
aperture_deadband_type	Тип апертур	Необязательный	Тип апертур, который будет использовать OPC UA сервер.	Одно из следующих значений: absolute - абсолютное значение (deadband_value в единицах измерения) percent - процент от	absolute

				диапазона измерения none - без апертуры	
aperture.de adband_val ue	Значение апертуры	Необязательный	Числовое значение порога; изменения меньше этого порога не приводят к уведомлениям.	float(число с плавающей точкой)	0.0

2.2.4.7 Раздел data_model.write

Раздел write представлен как список параметров орсиа-устройства с дополнительным полем command_topic.

Общий принцип следующий: на основе параметров, приведенных в root_topic_mqtt, command_topic и parameter_name формируется топик управления `/devices/{root_topic_mqtt}/controls/{parameter_name}/{command_topic}`, на который подписывается драйвер на mqtt-брокере. При получения этого топика от брокера, его значение payload записывается в соответствующее свойство орсиа-узла:

Описание в YAML	Название	Обязательный / Необязательный параметр	Описание	Допустимые значения	Значения по умолчанию
parameter_name	Имя параметра	Обязательный	Имя параметра. Используется в структуре топика <code>/devices/{root_topic_mqtt}/controls/{parameter_name}/{command_topic}</code> . Если значение parameter_name начинается с <code>/devices/</code> , то root_topic_mqtt не используется, а итоговый топик формируется как <code>{parameter_name}/{command_topic}</code> .	Строка. Не должно содержать символы + и #	—
command_topic	Суффикс топика управления	Необязательный	Значение, используемое в структуре управляющих топиков <code>/devices/{root_topic_mqtt}/controls/{parameter_name}/{command_topic}</code> .	Строка. Не должно содержать символы + и #	on
node.node_id	Идентификатор узла OPC UA	Обязательный	Строковый идентификатор узла на OPC UA-сервере. В рамках раздела data_model.write должен быть уникальным.	Строка вида: <code>"ns=x;i=y"</code> , где ns - namespace(пространство имен), y - номер узла в указанном пространстве имен Например: <code>"ns=3;i=1609"</code> , <code>"ns=2;s=Motor.RunCmd"</code>	—
node.data_type	Тип данных OPC UA	Обязательный	Тип данных значения, которое будет записано в узел. Используется для приведения входящего payload к нужному OPC UA типу. Поддерживаемые типы описаны в пункте 3.2.3	Одно из следующих значений, описанных в пункте 3.2.3	—

node.on_write_feedback.immediate_readback	Обратная связь при записи	Необязательный	Значение определяет, будет ли драйвер сразу после успешной записи повторно читать значение из OPC UA и публиковать его в MQTT как подтверждение/фактический результат записи.	true, false	false
---	---------------------------	----------------	---	-------------	-------

ВАЖНО!

В драйвере orsua2mqtt запись возможна только в атрибут Value, поэтому параметр attribute в разделе Write не работает. Запись автоматически будет выполняться только в свойство Value объекта, в отличие от раздела Read, где из orsua узла можно читать не только Value.

2.2.5 Поддерживаемые скалярные типы данных

Описание в YAML	Описание	Размер (байт)
boolean	Булева переменная (0 или 1)	1
sbyte	Знаковый байт	1
int16	16-битное знаковое целое число	2
uint16	16-битное беззнаковое целое число	2
int32	32-битное знаковое целое	4
uint32	32-битное беззнаковое целое	4
int64	64-битное знаковое целое	8
uint64	64-битное беззнаковое целое	8
float	32-разрядное число с плавающей запятой	4
double	64-разрядное число с плавающей запятой	8
string	Строка символов	—
bytestring	Строка байтов	—
datetime	Дата и время	—
guid	Глобально уникальный идентификатор	—

2.2.6 Поддерживаемые OPC UA атрибуты

Описание в YAML	Описание
nodeid	Уникальный индикатор узла
nodeclass	Тип узла(Объект, Переменная, Метод и т.д.)
browseName	"Человекочитаемое" имя для просмотра
displayName	Локализованный текст, используемый клиентами для отображения имени узла пользователю
description	Описание
writemask	Отображает возможности клиента по записи атрибутов узла. Атрибут WriteMask не учитывает права доступа пользователя.
userwritemask	Определяет, какие атрибуты узла разрешено записывать определенному пользователю (или роли пользователя).

isabstract	Является ли ObjectType абстрактным.
symmetric	Определяет, имеет ли связь между двумя узлами одно и то же значение независимо от направления.
inversename	Предоставляет удобочитаемое имя для связи при просмотре с точки зрения TargetNode обратно к SourceNode.
containsnoloops	Указывает, гарантирует ли сервер отсутствие циклических путей при следовании по прямым ссылкам в пределах данного View.
eventnotifier	Определяет, может ли узел использоваться для подписки на события в реальном времени или для доступа к историческим данным о событиях.
value	Представляет фактические данные или информацию, которые содержит узел.
datatype	Формат и синтаксис данных, хранящихся в атрибуте Value узла.
datatypedefinition	Предоставляет формальное, машиночитаемое описание внутренней структуры типа данных.
valuerank	Определяет структуру атрибута Value, указывая, является ли он скалярным значением или массивом.
arraydimensions	Определяет точную длину каждой размерности массива.
accesslevel	Определяет операции, которые клиенту разрешено выполнять над значением узла, такие как чтение, запись или подписка.
useraccesslevel	Указывает конкретные уровни доступа, которыми обладает текущий вошедший в систему пользователь для доступа к значению узла.
minimumsamplinginterval	Сообщает клиенту, насколько быстро сервер способен опрашивать базовый источник данных на предмет изменений.
historizing	Указывает, собирает ли и хранит ли сервер в данный момент исторические данные для этой конкретной переменной.
executable	Указывает, способен ли сервер в данный момент выполнять (исполнять) логику, связанную с этим методом.
userexecutable	Указывает, имеет ли текущий пользователь необходимые разрешения для вызова этого метода.

3. Примеры файлов конфигурации YAML

Далее приведены примеры конфигурации драйвера opcua2mqtt:

Корневой файл конфигурации opcua2mqtt.yaml:

```
opcua:
  enabled: true
mqtt:
  broker: "unix:///var/run/mosquitto/mosquitto.sock"
  client_id: "opcua2mqtt"
  keepalive: 120
  reconnect_delay: 5
  max_reconnect_delay: 300
logger:
```

```
level_log: 1
path_log: logs/opcua2mqtt.log
path_json_log: logs/opcua2mqtt.json
max_size_log: 1
max_backups_log: 10
max_age_log: 10
devices:
  device_1:
    file: opcua_server_1.yaml
  device_2:
    file: opcua_server_2.yaml
```

Файл конфигурации device_1(opcua_server_1.yaml):

```
device:
  enabled: true
  name: "opcua_server_1"
  root_topic_mqtt: "/devices/opcua_server_1"           # Базовый топик MQTT

connection:
  endpoint: "opc.tcp://10.10.10.1:53530/OPCUA/SimulationServer"
  timeout: 30                                           # Таймаут операций (общий)
  dial_timeout: 10                                     # Таймаут установки TCP соединения
  request_timeout: 30                                  # Таймаут запросов к серверу
  session_timeout: 300                                 # Таймаут сессии

# Retry и reconnect механизмы
retry_interval: 10                                     # Интервал между попытками подключения
max_retry_attempts: 0                                 # 0 = бесконечные попытки
auth_policy_id: "anonymous"
auth_security_policy: "None"
security_mode: "None"
security_policy: "None"
auth_type: "Anonymous"

application_name: "opcua2mqtt Client - Server1"
application_uri: "urn:opcua2mqtt:client:server1"
product_uri: "urn:opcua2mqtt"

data_model:
  read:
    subscribe:
      - publishing_interval_ms: 1000
        global_pub_time_s: 300
        timestamps_to_return: "Source"
        parameters:
          - parameter_name: Counter1
            node:
```

```
node_id: ns=17;i=1003
monitoring:
  filter:
    trigger: "StatusValue"
    aperture:
      deadband_type: "absolute"
      deadband_value: 0.5
- publishing_interval_ms: 2000
  global_pub_time_s: 300
  timestamps_to_return: "Source"
  parameters:
    - parameter_name: Counter2
      node:
        node_id: ns=17;i=1004
        monitoring:
          sampling_interval_ms: 1000
          queue_size: 10
          discard_oldest: true
          filter:
            trigger: "StatusValue"
            aperture:
              deadband_type: "none"
write:
- parameter_name: Counter1
  command_topic: "on"
  node:
    node_id: ns=17;i=1003
    data_type: "Double"
  on_write_feedback:
    immediate_readback: true
- parameter_name: Counter2
  command_topic: "on"
  node:
    node_id: ns=17;i=1004
    data_type: "Double"
  on_write_feedback:
    immediate_readback: true
```

Файл конфигурации device_2(opcua_server_2.yaml):

```
device:
  enabled: true
  name: "opcua_server_2"
  root_topic_mqtt: "/devices/opcua_server_2"           # Базовый топик MQTT

connection:
  endpoint: "opc.tcp://10.10.10.2:53530/OPCUA/SimulationServer"
  timeout: 30                                         # Таймаут операций (общий)
  dial_timeout: 10                                    # Таймаут установки TCP соединения
```

```
request_timeout: 30 # Таймаут запросов к серверу
session_timeout: 300 # Таймаут сессии

# Retry и reconnect механизмы
retry_interval: 10 # Интервал между попытками подключения
max_retry_attempts: 0 # 0 = бесконечные попытки
auth_policy_id: "anonymous"
auth_security_policy: "None"
security_mode: "None"
security_policy: "None"
auth_type: "Anonymous"

application_name: "opcua2mqtt Client - Server2"
application_uri: "urn:opcua2mqtt:client:server2"
product_uri: "urn:opcua2mqtt"

data_model:
  read:
    subscribe:
      - publishing_interval_ms: 1000
        global_pub_time_s: 300
        timestamps_to_return: "Source"
        parameters:
          - parameter_name: Counter1
            node:
              node_id: ns=17;i=1003
            monitoring:
              filter:
                trigger: "StatusValue"
                aperture:
                  deadband_type: "absolute"
                  deadband_value: 0.5
          - publishing_interval_ms: 2000
            global_pub_time_s: 300
            timestamps_to_return: "Source"
            parameters:
              - parameter_name: Counter2
                node:
                  node_id: ns=17;i=1004
                monitoring:
                  sampling_interval_ms: 1000
                  queue_size: 10
                  discard_oldest: true
                  filter:
                    trigger: "StatusValue"
                    aperture:
                      deadband_type: "none"

  write:
    - parameter_name: Counter1
```

```
command_topic: "on"
node:
  node_id: ns=17;i=1003
  data_type: "Double"
on_write_feedback:
  immediate_readback: true
- parameter_name: Counter2
command_topic: "on"
node:
  node_id: ns=17;i=1004
  data_type: "Double"
on_write_feedback:
  immediate_readback: true
```



Лист внесения изменений

Версия документа	Дата публикации	Авторы изменений	Описание изменений
rev_01	29.12.2025	Тимонин А.А.	Сформирован документ

