

# Транспорты AgavaSCADA/AgavaPLC

---



## Содержание

---

### Транспорты Modbus-RTU, Modbus-TCP

#### Узлы

##### TransportModbus-RTU

Доступные дочерние узлы

Свойства

##### TransportModbus-TCP

Доступные дочерние узлы

Свойства

##### ModbusRegister

Доступные дочерние узлы

Свойства

### Транспорт OPC-UA

#### Узлы

##### Транспорт OPC-UA

Свойства

Группа свойств "Транспорт OPC-UA"

Группа свойств "Клиент OPC-UA"

##### Группа тегов

Свойства

##### Тег OPC-UA

Свойства

### Создание и настройка транспорта OPC-UA

Добавление параметров в Транспорт OPC-UA, работающий в режиме "Клиент"

### Транспорт SNMP

#### Создание и настройка транспорта SNMP

Как определить версию SNMP для устройства

Авторизация в версиях SNMP v1 и SNMP v2c

Авторизация в версиях SNMP v3

Добавление параметров в Транспорт SNMP

## 1 Транспорты Modbus-RTU, Modbus-TCP

---

Данные виды транспортов предназначены для приема и передачи данных по протоколу Modbus. Поддерживаются последовательные и Ethernet подключения.

## **1.1 Узлы**

- TransportModbus-RTU - транспорт, работающий по последовательным линиям связи.
- TransportModbus-TCP - транспорт, работающий по ethernet.
- ModbusRegister - источник типа "Регистр Modbus".

### **1.1.1 TransportModbus-RTU**

Узел, представляющий транспорт Modbus, работающий по последовательным линиям связи RS-485, RS-422, RS-232.

#### **1.1.1.1 Доступные дочерние узлы**

- Группа узлов
- ModbusRegister

#### **1.1.1.2 Свойства**

### **1.1.2 TransportModbus-TCP**

Узел, представляющий транспорт Modbus, работающий по линиям связи Ethernet.

#### **1.1.2.1 Доступные дочерние узлы**

- Группа узлов
- ModbusRegister

#### **1.1.2.2 Свойства**

### **1.1.3 ModbusRegister**

Узел, представляющий источник типа "Регистр Modbus".

#### **1.1.3.1 Доступные дочерние узлы**

Нет

#### **1.1.3.2 Свойства**

Наименование свойства	Идентификатор	Тип	Доступ	Описание
Устройство	DeviceAddr	INT	Чтение / запись	
Функция чтения	ReadFunction	INT	Чтение / запись	
Регистр чтения	ReadRegister	INT	Чтение / запись	
Размер регистра	ParamSize	INT	Чтение / запись	
Приоритет	Priority	INT	Чтение / запись	
Функция записи	WriteFunction	INT	Чтение / запись	
Регистр записи	WriteRegister	INT	Чтение / запись	
Тип значения	ValueType	INT	Чтение / запись	
Порядок байт	ByteOrder	INT	Чтение / запись	
Операции после чтения	OperationsOnRead	NODESARRAY	Чтение / запись	
Операции перед записью	OperationsOnWrite	NODESARRAY	Чтение / запись	
Узел для чтения/записи	Sourceld	NODE	Чтение / запись	

Устройство

Функция чтения

Регистр чтения

Размер регистра

Приоритет

Функция записи

Регистр записи

Тип значения

Порядок байт

Операции после чтения

Операции перед записью

Узел для чтения/записи

## 2 Транспорт OPC-UA

---

Протокол **OPC-UA (OPC Unified Architecture)** представляет собой современный стандарт обмена данными в промышленной автоматизации, обеспечивая высокий уровень безопасности, масштабируемость и платформенную независимость. Он позволяет надёжно интегрировать устройства, системы управления и SCADA-приложения, гарантируя единый подход к взаимодействию между различными компонентами инфраструктуры.

В данном разделе описывается **транспорт OPC-UA**, предназначенный для настройки связи по протоколу OPC-UA. Далее будут изложены основные свойства, принципы конфигурации и особенности работы транспортного узла, что поможет пользователям быстро и эффективно интегрировать OPC-UA в свои проекты.

## 2.1 Узлы

- **Транспорт OPC-UA** – обеспечивает связь и обмен данными через протокол OPC-UA.
- **Группа тегов** – узел, предназначенный для логического объединения OPC-UA тегов в единое пространство имен.
- **Тег OPC-UA** – узел, служащий источником данных, который хранит и передает значения.

### 2.1.1 Транспорт OPC-UA

#### 2.1.1.1 Свойства

##### 2.1.1.1.1 Группа свойств "Транспорт OPC-UA"

Эта группа свойств определяет глобальное поведение плагина OPC-UA, задавая параметры для подключения и работы в выбранном режиме.

Группа свойств "Транспорт OPC-UA"

Наименование свойства	Идентификатор	Тип	Доступ	Значение по умолчанию
Режим работы	TransportMode	ENUM	Чтение / запись	Сервер
Адрес	IPAddr	STRING	Чтение / запись	opc.tcp://localhost:4840
Разделитель	TagSeparator	STRING	Чтение / запись	. (Точка)

##### Описание свойств группы "Транспорт OPC-UA" узла Транспорт OPC-UA

- 1. Режим работы.** Определяет режим работы плагина OPC-UA. Допустимые значения:
  - Не определен (ошибка): При попытке построить проект SCADA выдаст ошибку о некорректном параметре.
  - Клиент: В этом режиме плагин функционирует как клиент OPC-UA. Дочерние узлы транспортного узла представляют собой ссылки на удалённые OPC-UA-узлы и/или пространства имен (namespaces), откуда производится считывание данных. Для корректного получения данных с конкретного узла на сервере требуется указать адрес сервера, пространство имен (если задано) и название узла на сервере.
  - Сервер: В режиме сервера плагин предоставляет данные через протокол OPC-UA. Дочерние узлы транспортного узла (которые могут быть как отдельными узлами, так и пространствами имен) описывают данные и группы данных, доступные для чтения внешними клиентами. Клиентские приложения смогут обращаться к этим узлам, чтобы получать данные, предоставляемые сервером.
- 2. Адрес.** Определяет адрес для работы OPC-UA транспорта – как для подключения клиента, так и для конфигурации сервера. Для клиента, адрес указывает на OPC-UA сервер, к которому клиент будет пытаться подключиться. При указании некорректного или недоступного адреса клиент будет автоматически пытаться установить соединение с интервалом в 1 секунду. Для сервера адрес используется для настройки конечных точек (endpoint) OPC-UA сервера. Он задаёт URL, по

которому сервер будет принимать подключения. На адрес накладываются следующие ограничения:

- Строка не может быть пустой.
- Длина строки не должна превышать 512 символов.

3. **Разделитель.** Определяет символ, который разделяет *пространство имен* узлов и *названия* узлов. В качестве значения параметра допускается единственный символ; если символ отсутствует или длина строки превышает 1, SCADA выдаст ошибку сборки проекта.

### 2.1.1.1.2 Группа свойств "Клиент OPC-UA"

Эта группа свойств содержит настройки, определяющие поведение плагина в режиме работы Клиент.

Группа свойств "Клиент OPC-UA"

Наименование свойства	Идентификатор	Тип	Доступ	Значение по умолчанию
Способ опроса	DataRetrievalMethod	ENUM	Чтение / запись	Не определен (ошибка)
Пауза между пакетами	PacketDelay	INT	Чтение / запись	1 мс
Таймаут операций	Timeout	INT	Чтение / запись	5000 мс

### Описание свойств группы "Клиент OPC-UA" узла Транспорт OPC-UA

1. **Способ опроса.** Возможные значения Циклический опрос и Подписка. Параметр определяет способ взаимодействия клиента с сервером для получения актуальной информации. При выборе режима Циклический опрос клиент последовательно запрашивает данные по списку узлов, используя при этом значения свойств Таймаут операций и Пауза между пакетами. Режим Подписка позволяет обозначить интерес клиента к изменениям значений узлов на сервере: при обнаружении изменений сервер отправляет уведомления, которые клиенту остаётся только обработать.
2. **Пауза между пакетами.** Используется только для режима Циклический опрос. Параметр определяет задержку клиента между каждым отдельным запросом, при запросе значений узлов у сервера. Измеряется в миллисекундах. Если значение ноль - задержки нет.
3. **Таймаут операций.** Используется только в режиме Циклический опрос. Параметр задаёт задержку между циклами опроса сервера. Это время в миллисекундах, которое клиент ожидает после завершения цикла опроса сервера, позволяя снизить нагрузку на сервер и обеспечить стабильную работу системы.

### 2.1.2 Группа тегов

#### 2.1.2.1 Свойства

Узел Группа тегов реализует функциональность *пространств имен* (namespaces) протокола OPC-UA. Он используется для назначения OPC-UA тегов в определённое пространство имен. Чтобы узел Тег OPC-UA корректно ассоциировался с заданным

пространством, его необходимо сделать дочерним узла, обозначающего это пространство имен.

### Группа свойств "Группа OPC"

Наименование свойства	Идентификатор	Тип	Доступ	Значение по умолчанию
Пространство имен	NamespaceIndex	UCHAR	Чтение / запись	1

### Описание свойства "Пространство имен" узла "Группа тегов"

Свойство "Пространство имен" определяет числовой индекс пространства имен, используемого для формирования полного имени OPC-UA тегов. Допустимый диапазон значений: в диапазоне от 1 до 255. Если задано значение 0, при сборке проекта появится сообщение об ошибке. Если введено число больше 255, лишние разряды будут обрезаны для попадания в допустимый диапазон (например, значение 256 преобразуется в 25, значение 4851 — в 48).

Если числовой индекс пространства имен, заданный на клиенте, не совпадает с индексом, зарегистрированным на сервере, то узлы, принадлежащие данной группе, будут полностью недоступны для обмена данными между клиентом и сервером. Для корректного взаимодействия клиента и сервера необходимо, чтобы совпадали не только числовые индексы пространств имен, но и имена узлов, обозначающих эти пространства. Например, если у клиента пространство имен с индексом 3 именуется Dynamic, а у сервера — Dynamic (из-за опечатки), получение данных из OPC-UA узлов в этом пространстве станет невозможным.

## 2.1.3 Тег OPC-UA

### 2.1.3.1 Свойства

Узел Тег OPC-UA содержит свойства, определяющие поведение OPC-UA тега, включая тип данных, операции обработки значений и привязку к узлу в древовидной структуре проекта.

#### Свойства узла "Тег OPC-UA"

Наименование свойства	Идентификатор	Тип	Доступ	Значение по умолчанию
Тип значения	ValueType	ENUM	Чтение / запись	UInt16
Операции после чтения	OperationsOnRead		Чтение / запись	Отсутствует
Операции перед записью	OperationsOnWrite		Чтение / запись	Отсутствует
Узел для чтения/записи	Sourceld		Чтение / запись	Отсутствует
Права доступа	AccessRights	ENUM	Чтение / запись	Только чтение

### Описание свойств узла Тег OPC-UA

1. **Тип значения.** Определяет тип данных, с которыми работает тег. Допустимые значения:

- Не определен (ошибка): При сборке проекта SCADA выдаст диалоговое окно с ошибкой, информируя о невалидном значении параметра.
- BOOL: Логический тип, принимающий значения true или false.
- Int16: Целое число со знаком, диапазон от -32768 до 32767.

- Int32: Целое число со знаком, диапазон от -2147483648 до 2147483647.
- UInt16: Целое число без знака, диапазон от 0 до 65535.
- UInt32: Целое число без знака, диапазон от 0 до 4294967295.
- Float32: Число с плавающей запятой (32-битное, стандарт IEEE-754).
- String: Текстовая строка.

2. **Операции после чтения.** Содержит список операций, которые применяются к значению тега сразу после его чтения с сервера и перед передачей клиенту. Операции выполняются последовательно, от первой к последней в списке. Результат каждой операции используется как вход для следующей. Если список операций оставлен пустым, считанное значение будет передано клиенту без изменений. Список возможных операций:

- Умножение
- Скрипт C++
- Сложение
- Битовое AND
- Битовое OR
- Битовое SHR
- Битовое SHL
- Логическое И
- Логическое ИЛИ
- Логическое НЕ

3. **Операции перед записью.** Содержит список операций, применяемых к значению тега перед тем, как сервер запишет его после получения от клиента. Список операций идентичен списку для *Операций после чтения*. Операции выполняются последовательно, как описано выше. Если список операций оставлен пустым, сервер запишет переданное от клиента значение без изменений.

4. **Узел для чтения/записи.** Определяет ссылку на узел в древовидной структуре проекта SCADA, который хранит значение тега. Используется только в режиме работы Сервер. Значение должно быть выбрано из существующих узлов дерева проекта. Если указанный узел не найден или его тип значения несовместим с заданным типом тега, в Журнал событий будет записано сообщение о недопустимом типе.

5. **Права доступа.** Определяют режим доступа к тегу. Допустимые значения:

- Не определен (ошибка): При наличии такого значения во время сборки проекта SCADA выдаст ошибку, информируя о невалидном параметре.
- Только чтение: Тег доступен только для чтения; запись запрещена.
- Чтение и запись: Тег доступен для операций чтения и записи.

## 2.2 Создание и настройка транспорта OPC-UA

Для использования транспорта OPC-UA в проекте должна присутствовать транспортная система. Добавьте в систему узел типа Транспорт OPC UA.

Выберите режим работы транспорта:

- Клиент - транспорт работает как ведущий и выполняет запрос значений параметров у сервера.
- Сервер - транспорт работает в режиме "ведомый" и отвечает на запросы клиентов.

Задайте свойствам транспорта необходимые значения.

## 2.3 Добавление параметров в Транспорт OPC-UA, работающий в режиме "Клиент"

Для добавления параметров (OPC-UA тегов) в транспорт необходимо знать их имена. Определение имен параметров, предоставляемых OPC UA сервером, можно произвести путем просмотра дерева параметров сервера с помощью доступных OPC-UA клиентов, например:

- Unified Automation UaExpert (бесплатный).

После определения имен интересующих параметров необходимо добавить в транспорт нужные параметры в виде древовидной структуры в соответствии со структурой параметров в сервере.

Для примера продемонстрируем порядок настройки транспорта на примере публичного OPC UA сервера по адресу `opc.tcp://milo.digitalpetri.com:62541/milo`

Данный сервер предоставляет большой набор параметров, среди которых есть динамически меняющиеся, которые можно найти с помощью упомянутых выше OPC-UA клиентов:

```
┌-----┐
| Root/Objects/Dynamic/RandomDouble |
| Root/Objects/Dynamic/RandomFloat  |
| Root/Objects/Dynamic/RandomInt32  |
| Root/Objects/Dynamic/RandomInt64  |
└-----┘
```

С помощью клиента определяем, что интересующие нас параметры имеют следующие идентификаторы:

```
┌-----┐
| ns=2;s=Dynamic/RandomDouble      |
| ns=2;s=Dynamic/RandomFloat       |
| ns=2;s=Dynamic/RandomInt32       |
| ns=2;s=Dynamic/RandomInt64       |
└-----┘
```

Добавим в транспорт узел типа Группа тегов, установим его свойства следующим образом:

- Пространство имен = 2;
- Имя = Dynamic.

Добавим в данную группу узлы типа Тег OPC-UA с именами RandomDouble,



RandomFloat, RandomInt32, RandomInt64 и установим в их свойствах соответствующий тип значения.

После сохранения, компиляции и запуска проекта транспорт будет запрашивать у сервера добавленные теги.

## 3 Транспорт SNMP

---

Протокол **SNMP (Simple Network Management Protocol)** широко используется для мониторинга и управления сетевыми устройствами, такими как маршрутизаторы, коммутаторы, серверы и принтеры. Он позволяет собирать информацию о состоянии устройств и выполнять удалённые настройки. Для эффективного использования SNMP в AgavaSCADA необходимо правильно настроить **транспорт SNMP** и его параметры. В следующих разделах описывается процесс создания и настройки транспорта SNMP, а также добавления необходимых параметров для обеспечения надежного взаимодействия с управляемыми устройствами.

### 3.1 Создание и настройка транспорта SNMP

1. В дереве проекта, правой кнопкой мыши выделить группу узлов, представляющие транспорты проекта, выбрать Добавить узел -> Транспорт SNMP, в результате чего в дереве проекта будет создан узел транспорта SNMP.
2. В свойствах созданного транспорта необходимо указать IP-адрес устройства, версию SNMP протокола, используемого устройством, и необходимую информацию для получения доступа к устройству (community-строку для SNMP v2c или имя пользователя и пароль для SNMP v3).

#### 3.1.1 Как определить версию SNMP для устройства

Версию SNMP можно определить, обратившись к документации или настройкам устройства. Если доступ к этой информации ограничен, можно попробовать последовательно опросить устройство с использованием разных версий SNMP и определить, на какую версию оно откликается.

#### 3.1.2 Авторизация в версиях SNMP v1 и SNMP v2c

Авторизация осуществляется с помощью **community-строк**, которые действуют как простые пароли. Community-строка передается в открытом виде и может быть установлена на значения по умолчанию, такие как **public** для операций чтения или **private** для операций записи. Администратор системы может установить собственные значения для community-строк, используемых устройством. Безопасность в этих версиях протокола минимальна.

#### 3.1.3 Авторизация в версиях SNMP v3

Предоставляет усовершенствованные механизмы безопасности, включая аутентификацию и шифрование. Авторизация требует имя пользователя, а также может потребовать пароль аутентификации и пароль шифрования. Данная версия

протокола поддерживает различные уровни безопасности:

- **noAuthNoPriv**: Только имя пользователя без аутентификации и шифрования.
- **authNoPriv**: Аутентификация без шифрования.
- **authPriv**: Полная аутентификация и шифрование данных.

Пример корректно заданного Транспорта SNMP

Свойство	Значение
Версия SNMP	SNMP v2c
IP адрес	172.16.3.22
Имя пользователя	public
Пароль	

### 3.2 Добавление параметров в Транспорт SNMP

1. В дереве проекта найти и правой кнопкой мыши выделить узел транспорт SNMP и выбрать Добавить узел -> Параметр SNMP, в результате чего в дереве проекта будет создан узел параметра SNMP.
2. В свойствах узла, представляющего параметр SNMP, помимо группы свойств Основные, доступна также группа свойств Параметр SNMP, содержащая свойства, специфичные для параметра транспорта: OID, Тип значения и Права доступа.
  - **OID (Object Identifier)** — это уникальный идентификатор объекта управления в MIB (Management Information Base). Он представляет собой последовательность чисел, разделенных точками, например: 1.3.6.1.2.1.1.1.0. OID можно получить несколькими способами: обратиться к документации производителя, которая часто содержит список доступных OID и их описание; использовать MIB-файлы — специальные файлы, предоставляемые производителем устройства и содержащие все OID и их свойства; или воспользоваться сторонними утилитами (например SNMP Walk, позволяющая просмотреть все доступные OID на устройстве).
  - **Тип значения** определяет тип данных, которые будет получен и/или отправлен при взаимодействии с устройством по определенному OID.
  - **Права доступа** — свойство определяющее, какие операции могут быть выполнены с данным OID — только чтение или чтение и запись.

Пример корректно заданного Параметра SNMP

Свойство	Значение
Тип значения	String
Права доступа	Чтение и запись
OID	1.3.6.1.2.1.1.1.0

После корректного задания свойств **Транспорта SNMP** и добавления **параметров SNMP** для данного транспорта, пользователь AgavaSCADA сможет эффективно использовать передаваемую информацию узлами-параметрами SNMP. Это позволит осуществлять мониторинг и анализ данных устройств, интегрированных в сеть.

---

Источник — [http://docs.kb-agava.ru/index.php?title=Транспорты\\_AgavaSCADA/AgavaPLC&oldid=2633](http://docs.kb-agava.ru/index.php?title=Транспорты_AgavaSCADA/AgavaPLC&oldid=2633)

---

Эта страница в последний раз была отредактирована 18 декабря 2024 в 11:05.