

# Модули <OPC-UA> подсистемы “Сбор данных” и подсистемы “Транспортные протоколы”

Параметр	Модуль 1	Модуль 2
ID:	OPC-UA	OPC-UA
Имя:	OPC-UA	OPC-UA
Тип:	DAQ	Протокол
Источник:	daq_daq OPC-UA.so	
Версия:	0.5.0	0.5.0
Автор:	Роман Савоченко	
Описание:	Предоставляет реализацию клиентского сервиса OPC UA.	Предоставляет реализацию OPC UA протокола.
Лицензия:	GPL	

## Оглавление

<a href="#">Модули &lt;OPC-UA&gt; подсистемы “Сбор данных” и подсистемы “Транспортные протоколы”</a>	1
<a href="#">Введение</a>	2
<a href="#">1. Протокол OPC UA</a>	3
<a href="#">2. Модуль реализации протокола</a>	4
<a href="#">2.1. Обслуживание запросов по протоколу OPC UA</a>	4
<a href="#">3. Модуль сбора данных</a>	6
<a href="#">3.1. Контроллер данных</a>	6
<a href="#">3.2. Параметры</a>	8

## Введение

OPC (OLE for Process Control) это семейство протоколов и технологий, предоставляющих единый интерфейс для управления объектами автоматизации и технологическими процессами. Создание и поддержку спецификаций OPC координирует международная некоммерческая организация [OPC Foundation](#), созданная в 1994 году ведущими производителями средств промышленной автоматизации.

В виду того, что значительное влияние в организации OPC Foundation имеет корпорация Microsoft, протоколы OPC, до последнего времени, были одноплатформенными и закрытыми, в виду привязки к закрытым технологиям MS Windows. Однако с недавних пор организацией OPC Foundation были созданы такие многоплатформенные интерфейсы, как OPC XML DA и OPC UA. Наибольший интерес из них представляет интерфейс OPC UA, как унифицирующий все интерфейсы ранних версий в рамках открытых и многоплатформенных технологий.

Данный модуль реализует поддержку интерфейса и протокола OPC UA как в виде клиентского сервиса, так и в виде сервера OPC UA. Клиентский сервис OPC UA реализуется одноимённым модулем подсистемы "Сбор данных", а сервер реализуется модулем подсистемы "Протоколы".

В текущей версии данных модулей реализуются бинарная часть протокола и базовые сервисы в небезопасном режиме. В последствии планируется расширение модуля для работы через HTTP/SOAP, поддержки политик безопасности и реализации остальных сервисов OPC UA.

Хотя протокол OPC UA и является многоплатформенным, его спецификация и SDK не являются свободнодоступными, а предоставляются только членам организации OPC Foundation. По этой причине реализация данных модулей столкнулась с значительными препятствиями и проблемами.

Во первый, протокол OPC UA сложен и реализация его вообще без спецификации крайне трудоёмка. По этой причине работы над данными модулями долгое время не начиналась и только благодаря спонсорской помощи одной из организаций-члена OPC Foundation проект OpenSCADA получил документацию спецификации. При этом SDK и исходные тексты ANSIC-API протокола OPC-UA получены небыли по причине не совместимости их лицензии с GPL и как следствие потенциальной угрозой нарушения лицензии при работе с исходными текстами, что могло привести к последующим юридическим проблемам при свободном распространении данных модулей.

Во вторых, даже наличие спецификации не позволяет решить ряд технических вопросов без примера реализации и возможности проверки на рабочем прототипе клиента и сервера OPC UA. Например, именно технические особенности реализации алгоритмов симметричного шифрования и получения ключей для них не позволили закончить реализацию поддержки политик безопасности, на данный момент.

Для отладки функционирования модулей было использовано демонстрационное ПО фирмы [Unified Automation](#) в составе OPC UA клиента - UA Expert и сервера - OPC UA Demo Server.

# 1. Протокол OPC UA

OPC UA это платформо-независимый стандарт с помощью которого системы и устройства различного типа могут взаимодействовать путём отправки сообщений между клиентом и сервером через различные типы сетей. Протокол поддерживает безопасное взаимодействие путём валидации клиентов и серверов, а также противодействия атакам. OPC UA определяет понятие *Сервисы*, которые сервера могут предоставлять, а также сервисы, которые сервер поддерживает для клиента. Информация передаётся в виде типов данных определённые OPC UA и производителем, кроме того сервера определяют объектную модель, для которой клиенты могут осуществлять динамический обзор.

OPC UA предоставляет совмещение, интегрированного адресного пространства и сервисной модели. Это позволяет серверу интегрировать данные, нарушения (Alarms) и события (Events), историю в этом адресном пространстве, а также предоставлять доступ к ним посредством интегрированных сервисов. Сервисы также предоставляют интегрированную модель безопасности.

OPC UA позволяет серверам предоставлять для клиентов определения типов для доступа к объектам из адресного пространства. OPC UA допускает предоставление данных в различных форматах, включая бинарные структуры и XML-документы. Через адресное пространство клиенты могут запросить у сервера метаданные которые описывают формат данных.

OPC UA добавляет поддержку множественной связности между узлами вместо простого довольствования иерархичностью. Такая гибкость, в комбинации с определением типов, позволяет применять OPC UA для решения задач в широкой проблемной области.

OPC UA спроектирован для обеспечения надёжной выдачи данных. Основная особенность всех OPC серверов способность выдавать данные и события.

OPC UA спроектирован для поддержки широкого диапазона серверов, от простых ПЛК до промышленных серверов. Эти сервера характеризуются широким спектром размеров, производительности, платформ исполнения и функциональной ёмкости. Следовательно OPC UA определяет исчерпывающее множество возможностей, и сервер может имплементировать подмножества этих возможностей. Для обеспечения совместимости OPC UA определяет подмножества, именуемые *Профилями*, которые сервера могут указывать для согласования. Клиенты могут в последствии выполнять обзор профилей сервера и пробрасывать взаимодействие с сервером основанном на профилях.

OPC UA спецификация спроектирована как ядро в слое изолированном от подлежащих компьютерных технологий и сетевых транспортов. Это позволяет OPC UA, при необходимости, расширяться на будущие технологии без отторжения основы дизайна. На данный момент спецификацией определены два способа кодирования данных: XML/text и UA Binary. В дополнение, определено два типа транспортного слоя: TCP и HTTP/SOAP.

OPC UA спроектирован как решение для миграции с OPC клиентов и серверов, которые основаны на Microsoft COM технологиях. OPC COM сервера (DA, HDA и A&E) могут быть легко отражены в OPC UA. Производители могут самостоятельно осуществлять такую миграцию или же рекомендовать пользователям использовать обёртки и конверторы между этими протоколами. OPC UA унифицирует предыдущие модели в едином адресном пространстве с единым множеством сервисов.

## 2. Модуль реализации протокола

Модуль протокола содержит код реализации протокольной части OPC UA как для клиентского, так и для серверного сервиса. Для построения OPC UA сервера достаточно создать входящий транспорт и выбрать в нём модуль данного протокола.

### 2.1. Обслуживание запросов по протоколу OPC UA

Входящие запросы к модулю-протоколу обрабатываются модулем в соответствии со сконфигурированными конечными узлами OPC UA (EndPoints) (рис.1).

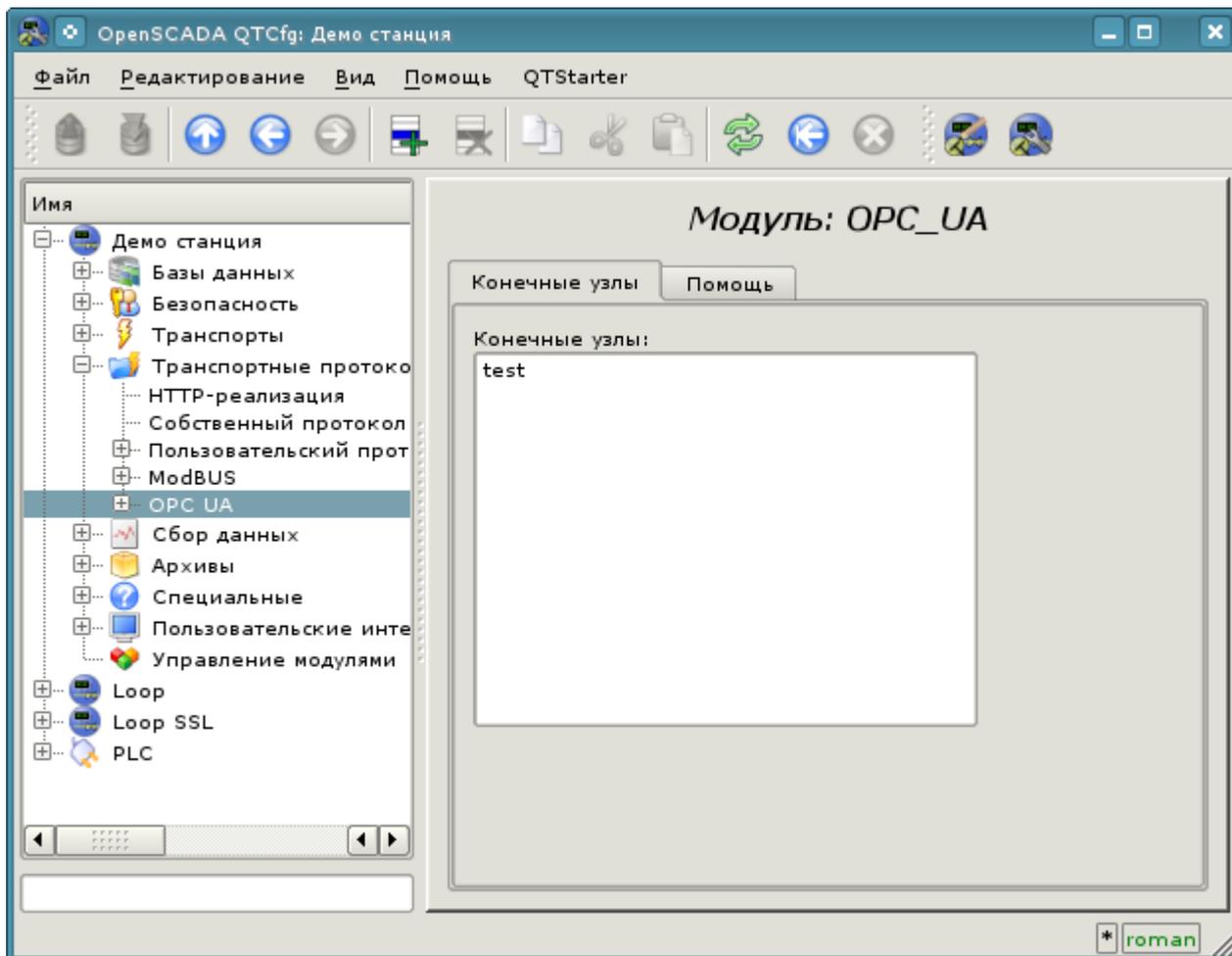


Рис.1. Конечные узлы протокола.

Конечный узел протокола OPC UA это фактически объект сервера OPC UA. Конечные узлы в OPC UA могут быть как локальными, так и удалёнными. Локальный конечный узел предназначен для предоставления ресурсов станции по протоколу OPC UA, в то время как удалённые конечные узлы служат как для выполнения сервиса обзора доступных OPC-UA узлов, так и для шлюзования запросов к удалённым станциям. В данной версии модуля поддерживается только конфигурация локальных конечных узлов.

Общая конфигурация конечного узла осуществляется на главной вкладке страницы конечного узла (рис.2), параметрами:

- Состояние узла, а именно: статус, "Включен" и имя БД, содержащей конфигурацию.
- Идентификатор, имя и описание узла.
- Состояние, в которое переводить узел при загрузке: "Включен".
- Тип кодирования протокола. На данный момент это только "Бинарный".

- URL конечной точки.
- Сертификат сервера в формате PEM.
- Приватный ключ в формате PEM.
- Политики безопасности сервера.

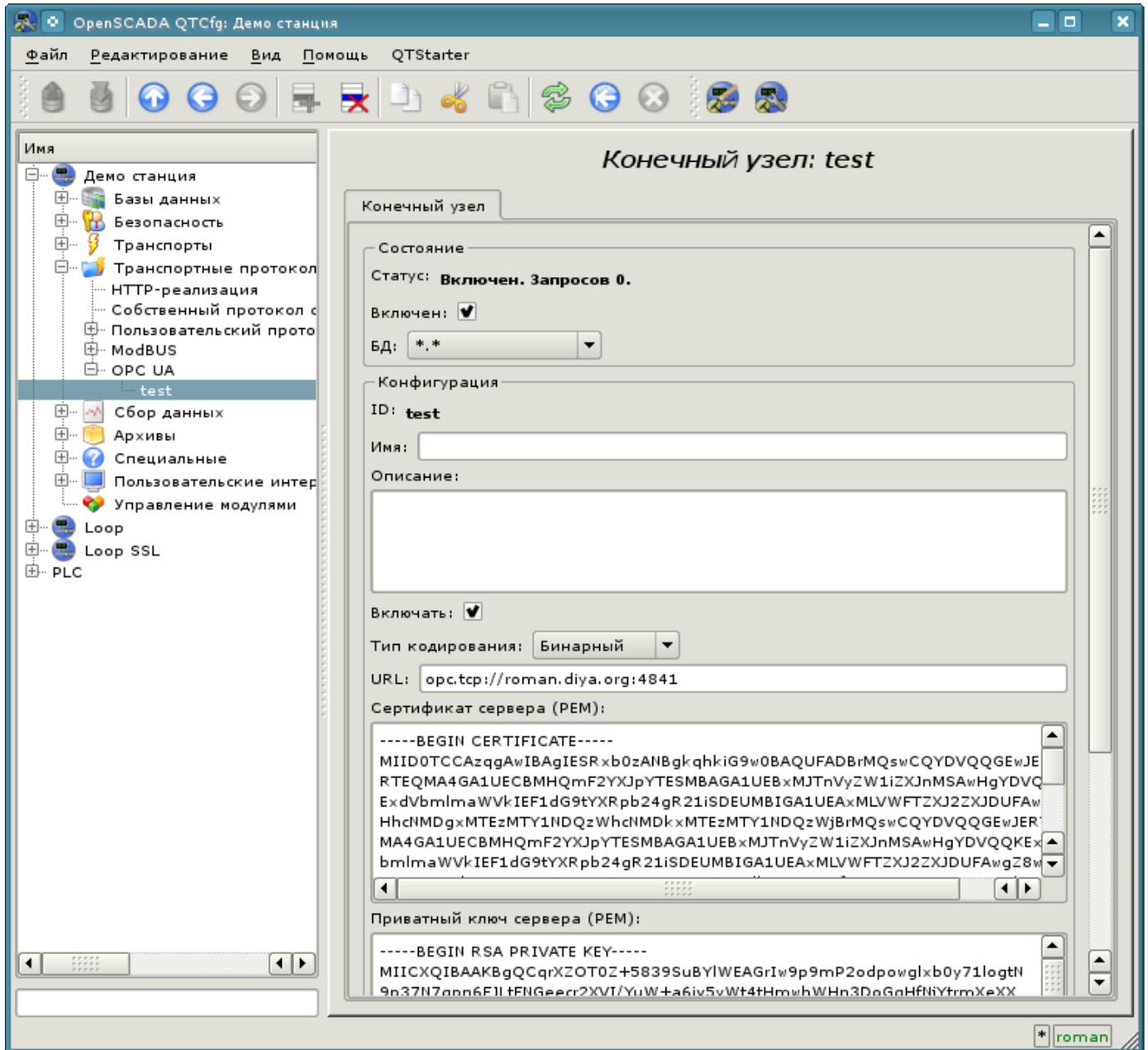


Рис.2. Главная вкладка страницы конечного узла.

### 3. Модуль сбора данных

Модуль сбора данных предоставляет возможность опроса и записи атрибутов значения(13) узлов типа "Переменная".

#### 3.1. Контроллер данных

Для добавления источника данных OPC UA создаётся и конфигурируется контроллер в системе OpenSCADA. Пример вкладки конфигурации контроллера данного типа изображен на рис.3.

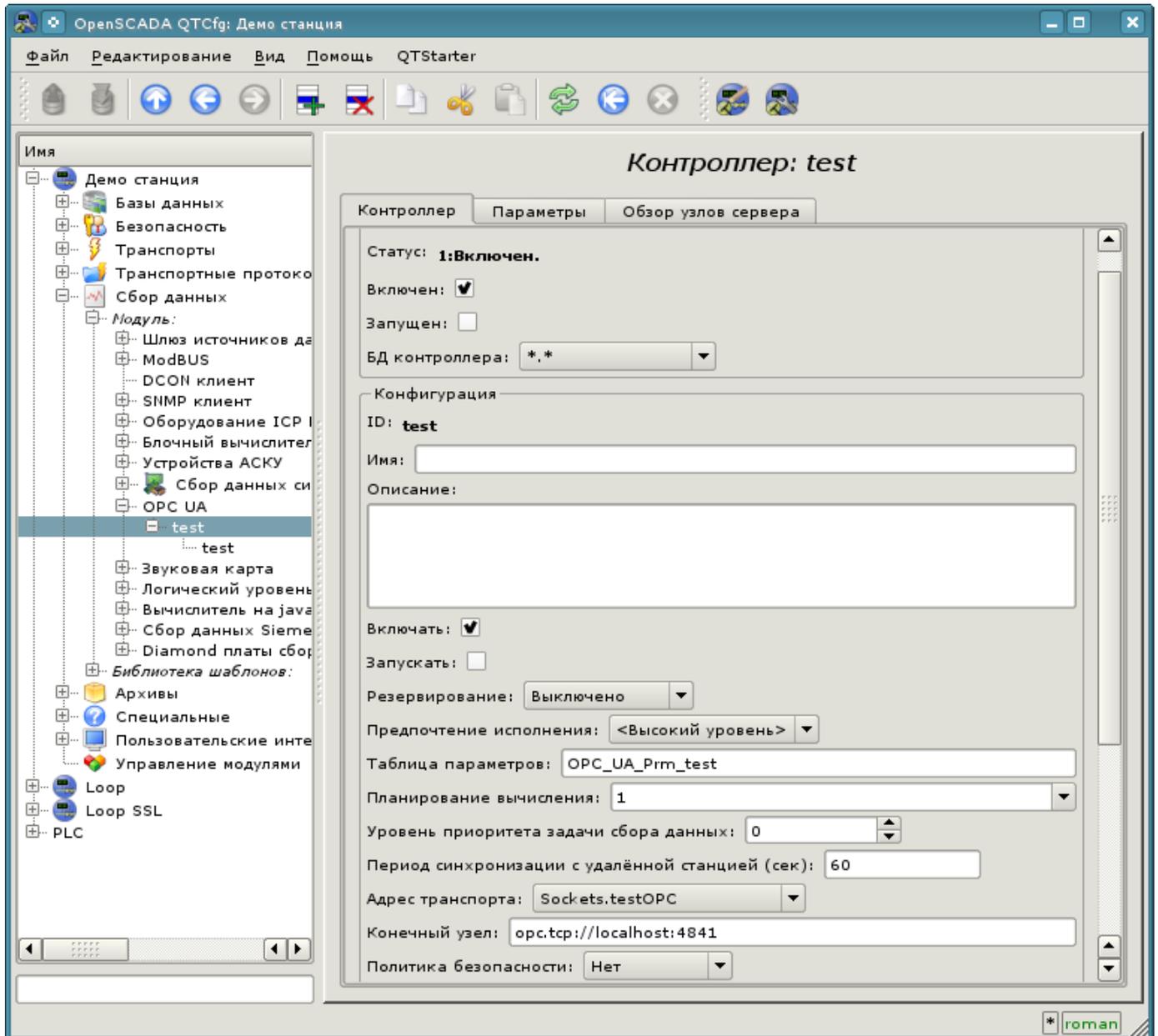


Рис.3. Вкладка конфигурации контроллера.

С помощью этой вкладки можно установить:

- Состояние контроллера, а именно: Статус, "Включен", "Запущен" и имя БД, содержащей конфигурацию.
- Идентификатор, имя и описание контроллера.
- Состояние, в которое переводить контроллер при загрузке: "Включен" и "Запущен".
- Режим горизонтального резервирования и предпочтение исполнения данного контроллера.

- Имя таблицы для хранения конфигурации параметров контроллера.
- Параметры планировщика исполнения задачи сбора и её приоритетность.
- Период синхронизации конфигурации атрибутов параметров с удалённой станцией.
- Адрес исходящего транспорта из списка сконфигурированных исходящих транспортов в подсистеме "Транспорты" OpenSCADA.
- URL конечного узла удалённой станции.
- Политика безопасности и режим безопасности сообщения.
- Сертификат клиента в формате PEM.
- Ограничение числа атрибутов в параметре, для режима импорта всех атрибутов принадлежащих объекту.

С целью облегчения идентификации узлов на удалённой станцией, а также выбора их для вставки в параметре контроллера, в объекте контроллера предусмотрена вкладка навигации по узлам удалённой станции, где можно пройти по дереву объектов и ознакомиться с их атрибутами (рис.4).

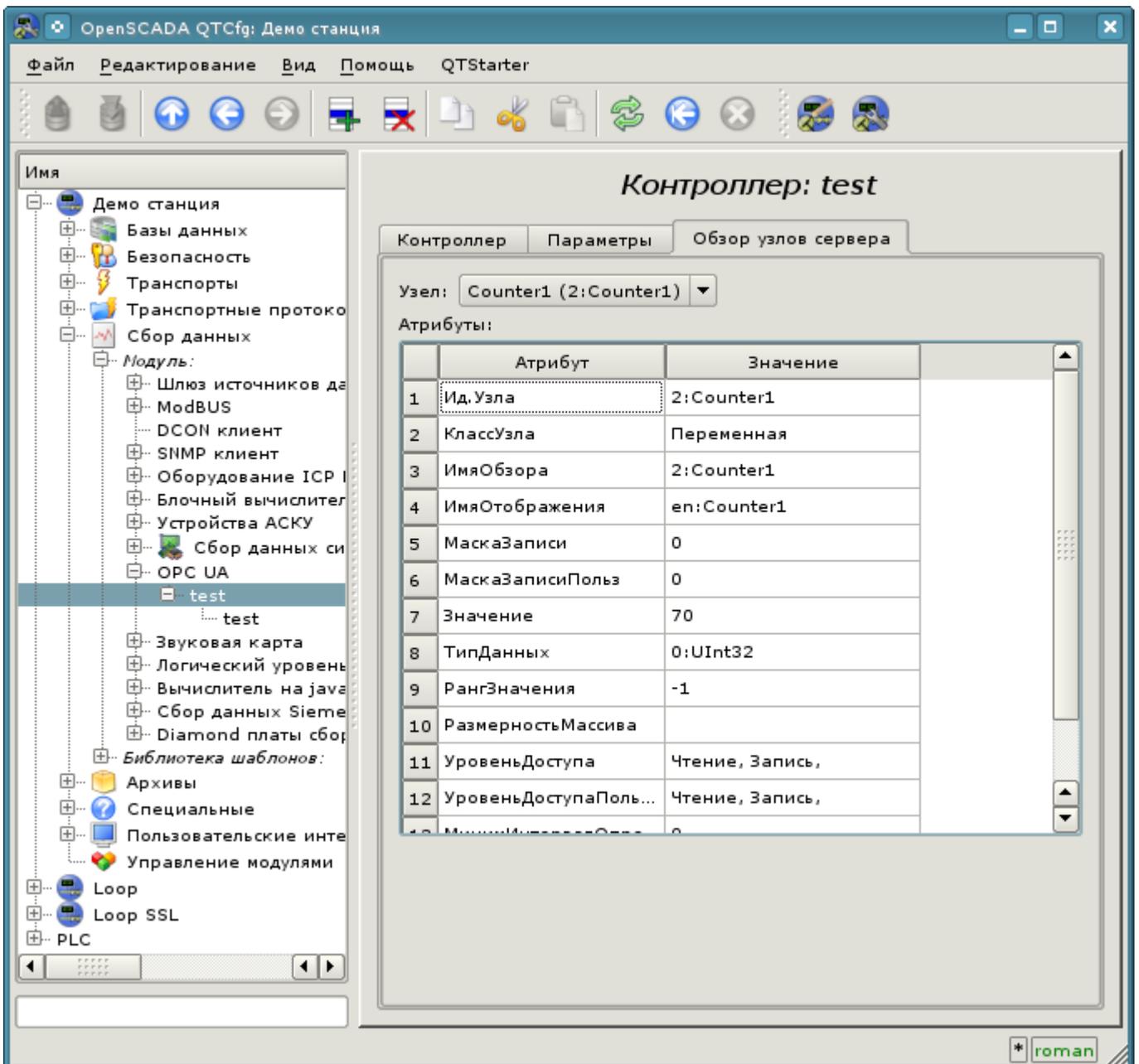


Рис.4. Вкладка "Обзор узлов сервера" страницы контроллера.

### 3.2. Параметры

Модуль сбора данных предоставляет только один тип параметров - "Стандарт". Дополнительным конфигурационным полем параметра данного модуля (рис.5) является перечень узлов OPC UA. Атрибут в этом перечне записывается следующим образом: [ns:id].

Где:

- *ns* - область имён, числом;
- *id* - идентификатор узла, числом или строкой.

Пример:

- '0:84' - корневая директория;
- '3:BasicDevices2' - узел базовых устройств в области имён 3.

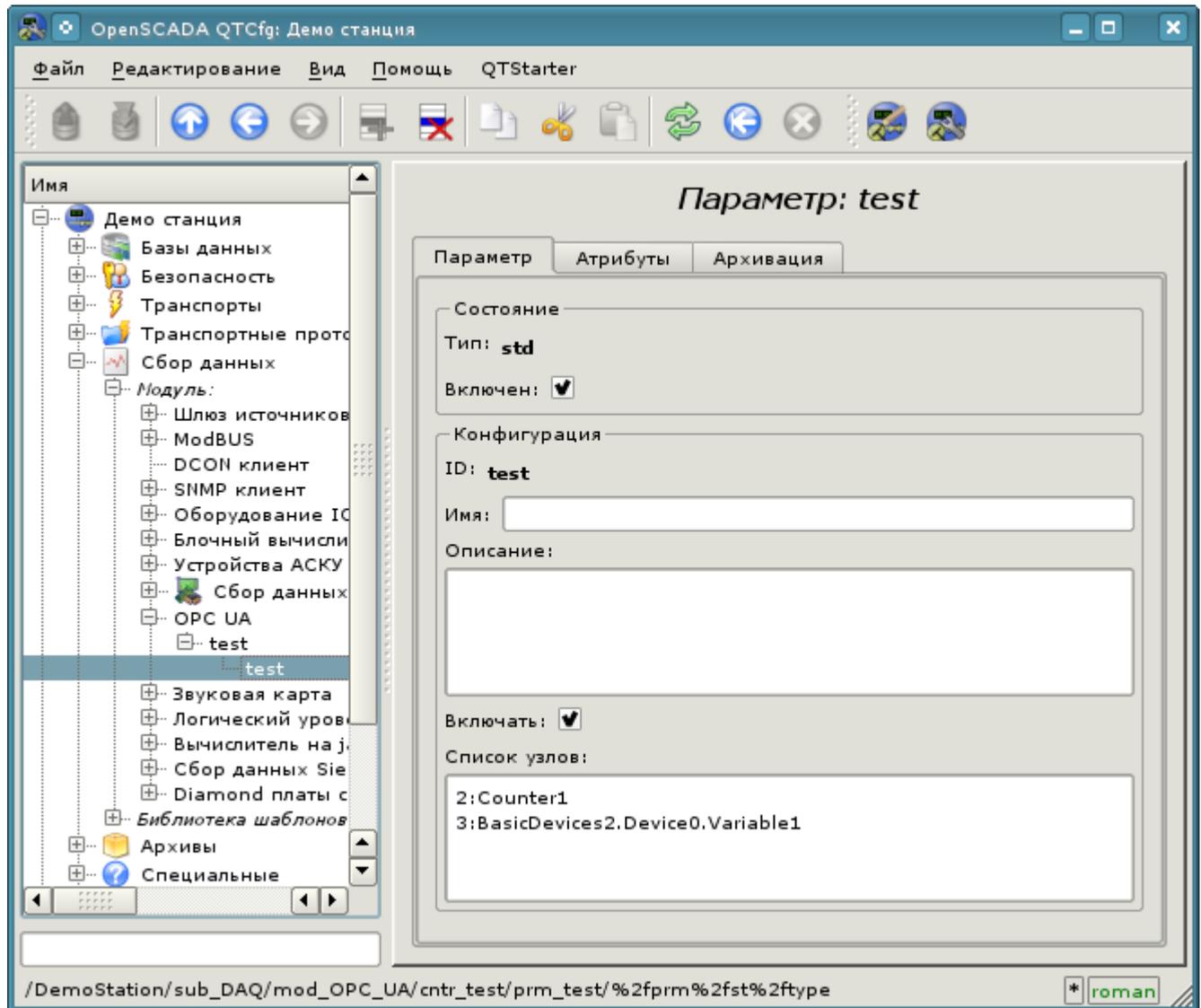


Рис.5. Вкладка конфигурации параметра.

В соответствии с указанным списком узлов выполняется опрос и создание атрибутов параметра (рис.6).

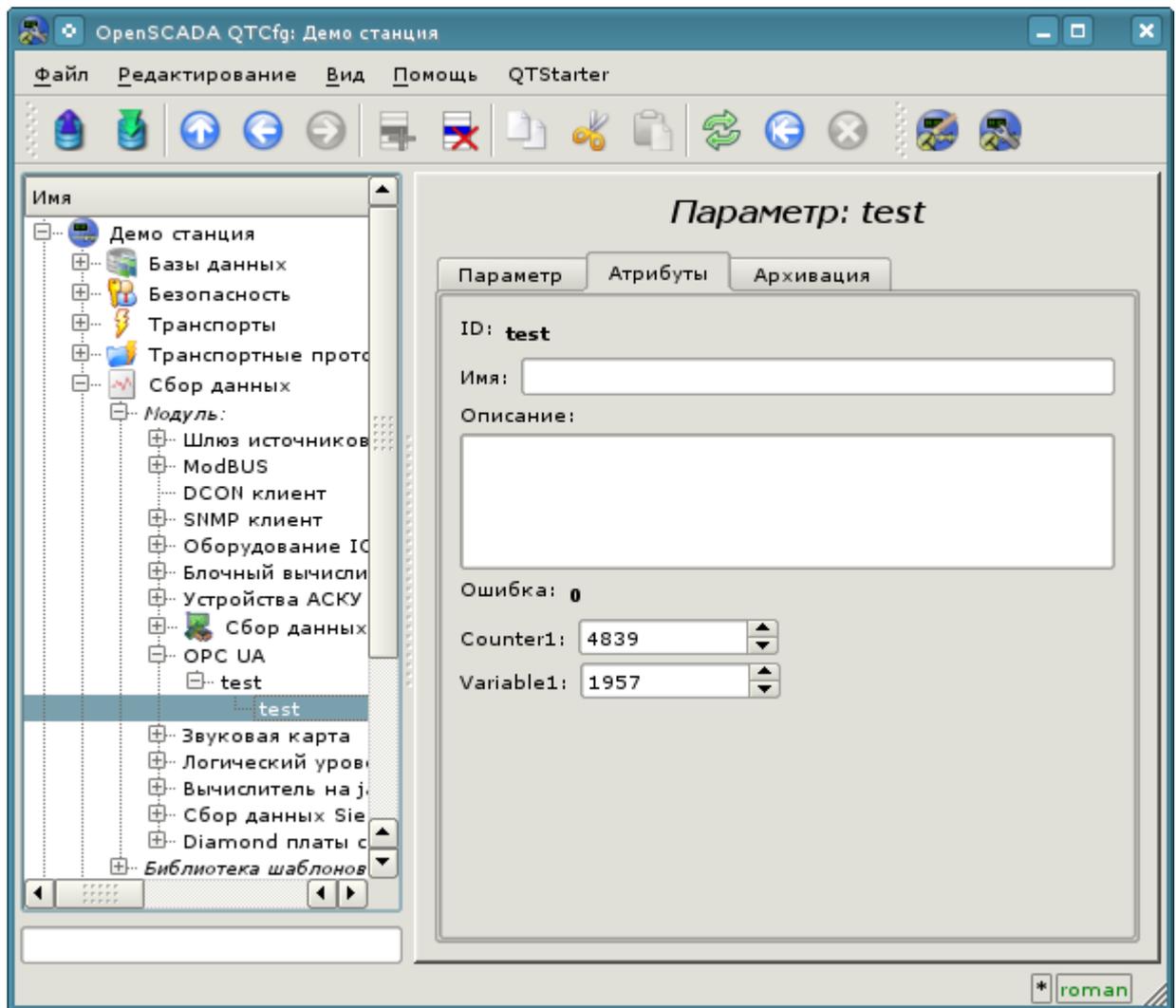


Рис.6. Вкладка атрибутов параметра.