Модульная интегрированная

SCADA КРУГ-2000™

Версия 4.4

ОРС - ТЕХНОЛОГИИ В КРУГ-2000

Руководство Пользователя

Модульная интегрированная SCADA КРУГ-2000™. OPC - технологии в КРУГ-2000.
Руководство Пользователя/1-е изд.

НПФ «КРУГ»

440028, г. Пенза, ул. Титова, 1

Телефон: (841-2) 49-97-75

E-mail: **support@krug2000.ru**

http:// **www.krug2000.ru**

Обозначение документа: КР01.20020W-04.40-И2.1.12

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ

Технология ОРС

Технология **OPC** (**OLE for Process Control**) предназначена для унификации доступа к данным различных систем и устройств.

Технология ОРС базируется на клиент-серверной архитектуре. В рамках ОРС существуют понятия ОРС-сервера и ОРС-клиента.

ОРС-сервер – программа, преобразующая данные из внутреннего формата устройства или системы в формат данных ОРС и передающая их ОРС-клиентам.

Возможна организация двустороннего обмена данными между ОРС-сервером и ОРС-клиентом. ОРС-сервер является источником данных для ОРС-клиентов и может принимать от ОРС-клиентов команды на прием данных.

ОРС-сервер предоставляет ОРС-клиентам свои данные в виде тегов.

Тег – единица данных ОРС-сервера (рисунок В.1)

Данные	**Признак достоверности**	**Метка времени**

Рисунок В.1 -Структура тега

Данные тега могут содержать как значение отдельного параметра какого-либо устройства, так и значения, характеризующие состояние целой системы. Структура данных тега определяется назначением и реализацией ОРС-сервера.
Кроме непосредственно самих данных, тег содержит в себе дополнительную информацию:

Признак достоверности данных — величина, показывающая степень достоверности данных тега. Может принимать 3 значения:

- *OPC_QUATITY_GOOD* – данные достоверны
- *OPC_QUALITY_BAD* – данные недостоверны
- *OPC_QUALITY_UNCERTAIN* – достоверность данных не может быть определена.

Метка времени — дата и время, когда тег был отправлен ОРС-клиенту.
Перед началом обмена данными ОРС-клиент узнает значения каких тегов ОРС-сервер может ему предоставить, выбирает нужные и начинает опрос ОРС-сервера.

ОРС-клиент – программа, принимающая от ОРС-серверов данные в формате ОРС и преобразующая их во внутренний формат устройства или системы. ОРС-клиент является инициатором обмена данных с ОРС-сервером и не может служить источником данных для других ОРС-клиентов.

ОРС-клиент может производить опрос ОРС-сервера двумя способами:

- *Синхронный опрос* – команда на чтение или запись данных тегов ОРС-сервера посылается ОРС-клиентом через жестко заданный промежуток времени. Этот вариант обмена предполагает получение данных через заданный промежуток времени вне зависимости от того, изменились значения в тегах ОРС-сервера или нет
- *Асинхронный опрос* – команда на чтение или запись значений тегов ОРС-сервера посылается ОРС-клиентом после того, как от ОРС-сервера пришло уведомление об изменении значений его тегов. Этот вариант обмена позволяет существенно снизить нагрузку на ОРС-сервер и сетевые узлы, если данные передаются по сети. Асинхронный опрос используется большинством ОРС-клиентов по умолчанию.

Спецификации OPC в SCADA КРУГ-2000

Программное обеспечение SCADA КРУГ-2000 включает модули OPC-клиентов и модули OPC-серверов. Это позволяет SCADA системе, с одной стороны, получать данные от OPC-серверов разнообразных приборов и оборудования, а с другой, предоставлять свои данные OPC-клиентам систем более высокого уровня, например, MES-системам (рисунок В.2).

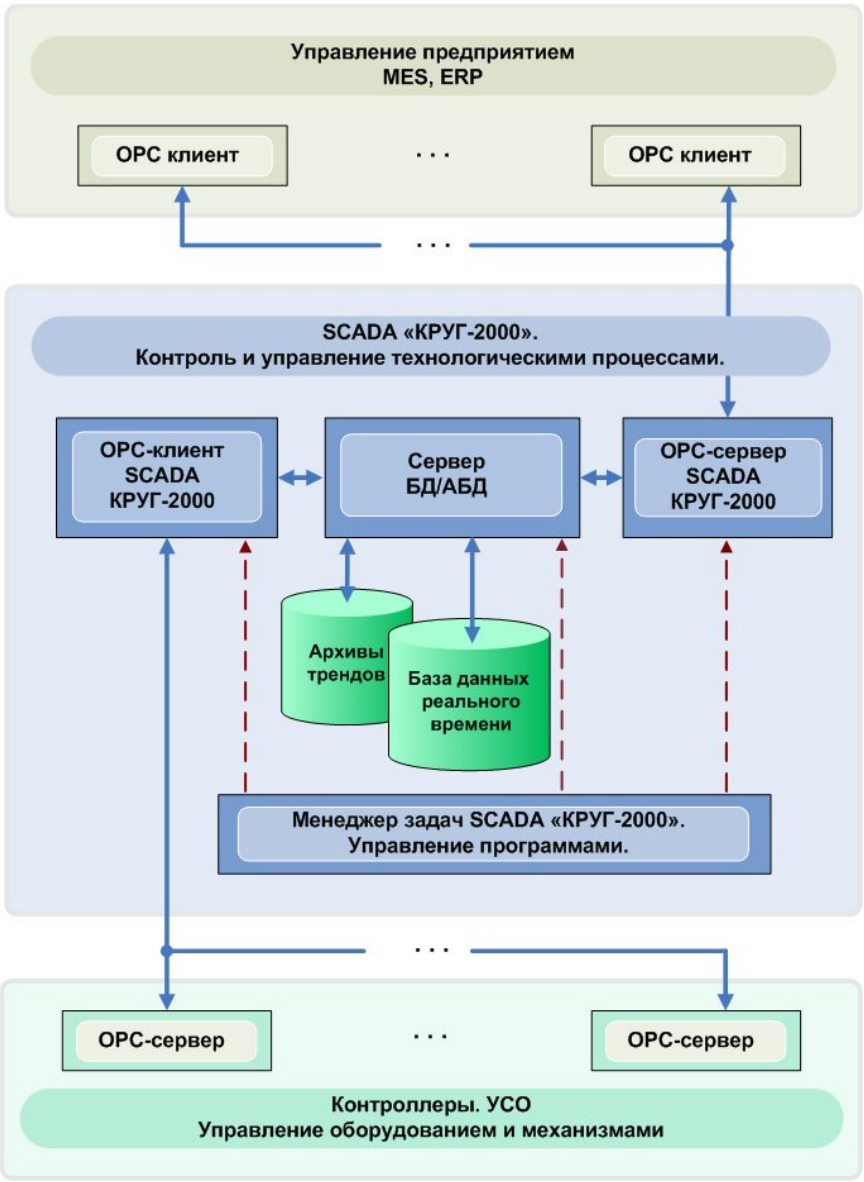


Рисунок В.2 – Программные компоненты OPC-технологии в SCADA КРУГ-2000

SCADA КРУГ-2000 поддерживает следующие *спецификации OPC:*

- ***OPC DA (Data Access)*** –обеспечивает обмен оперативными данными.

  Спецификация OPC Data Access обеспечивает базовые функциональные возможности для доступа к данным (чтение и запись) различных устройств посредством стандартного набора интерфейсов.

  Основное назначение OPC DA состоит в том, чтобы обеспечить интерфейсы для сбора данных при реализации вертикальной информационной архитектуры – доставка данных от OPC-сервера (устройства или SCADA) клиентскому приложению, работающему на компьютере более высокого уровня.

- ***OPC HDA (Historical Data Access)*** **–** предоставляет доступ к историческим данным. Использование этой спецификации позволяет представить архивные данные в универсальном формате, как в простых системах визуализации, так и в сложных SCADA системах.
  Исторические тренды, являясь важным компонентом информационной архитектуры предприятия, должны быть доставлены, как Пользователям, так и клиентским программам, заинтересованным в этой информации.
- ***OPC UA*** **(Unified Architecture)** — спецификация, определяющая передачу данных в промышленных сетях и взаимодействие устройств в них. Спецификация OPC UA совмещает все преимущества предыдущих спецификаций и открывает новые горизонты для применения OPC-технологий. В частности, благодаря тому, что произошел отказ от использования COM-интерфейса, обеспечивается кросс-платформенная совместимость. Новый стандарт позволяет обеспечить более высокий уровень безопасности данных, чем OPC DA, и дает возможность организации передачи информации через сеть интернет.

Подробную информацию о технологии OPC можно получить на сайте OPC Foundation Research Group: www.opcfoundation.org.

1 OPC DATA ACCESS

1.1 OPC DA СЕРВЕР

1.1.1 Общие сведения

OPC DA (OPC Data Access) сервер предоставляет OPC DA клиентам доступ к значениям оперативных данных из базы данных реального времени **SCADA КРУГ 2000**.

Внимание!!!

OPC DA сервер может работать с любыми переменными НЕнулевых каналов без дополнительной настройки базы данных!

OPC DA сервер SCADA КРУГ 2000 поддерживает обмен данными согласно спецификации OPC DA версии 2.0 (**OPC Data Access Specification version 2.0**).

Схема доступа к оперативным данным SCADA КРУГ-2000 с помощью OPC DA сервера представлена на рисунке 1.1

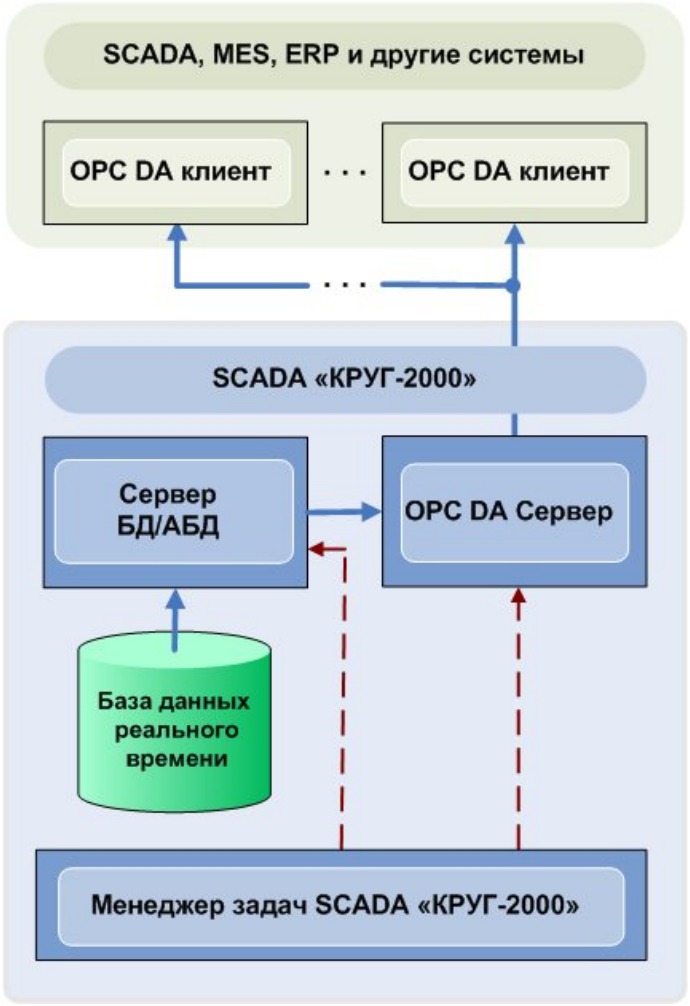


Рисунок 1.1 – Доступ к оперативным данным SCADA КРУГ-2000

1.1.2 Состав OPC DA сервера

OPC DA сервер SCADA КРУГ-2000 включает следующие файлы:

- **kropcserv.exe** – исполняемый файл OPC DA сервера
- **kropcserv.ini** – файл настроек OPC DA сервера
- **krugbrowse.dll** – вспомогательная библиотека для отображения пространства имен тегов OPC DA сервера
- **opcbrowsetest.exe** – демонстрационный OPC-клиент
- **fsserver.dll** – вспомогательная библиотека для работы OPC DA сервера
- **mfc42.dll** – вспомогательная библиотека для работы OPC DA сервера
- **msvcrt.dll** – вспомогательная библиотека для работы OPC DA сервера
- **opccomn_ps.dll** – стандартная библиотека для локального и удаленного обмена данными
- **opcenum.exe** – стандартная утилита для поиска зарегистрированных на машине OPC DA серверов
- **opcproxy.dll** – стандартная библиотека для локального и удаленного обмена данными

1.1.3 Настройка базы данных для работы с OPC DA сервером

☝Внимание!!!

Если Вам необходимо передавать с помощью OPC DA сервера только «физические» переменные, т.е. переменные с ненулевым номером канала связи, то никаких дополнительных настроек БД выполнять не требуется!

Если же Вам необходимо передавать с помощью OPC DA сервера еще и «виртуальные» переменные, т.е. переменные с нулевым номером канала (например, значения данных, рассчитываемые на Станции оператора программой КРУГОЛ или вводимые оператором вручную), то необходимо выполнить дополнительные настройки БД:

- создать дополнительный канал связи «**OPC-сервер**»;
- назначить «виртуальные» переменные созданному каналу связи «OPC-сервер».

В генераторе базы данных в дереве объектов в ветке Система вызовите контекстное меню для ветки Абоненты (рисунок 1.1.1). Создайте абонента (тип: Станция оператора, или Станция архивирования, или Сервер ОБД, или Сервер АБД). Для создания канала на форме описания абонента (рисунок 1.1.2) необходимо левой кнопкой мыши выбрать вкладку «Добавить канал», в появившемся контекстном меню необходимо выбрать тип создаваемого канала – OPC-сервер.

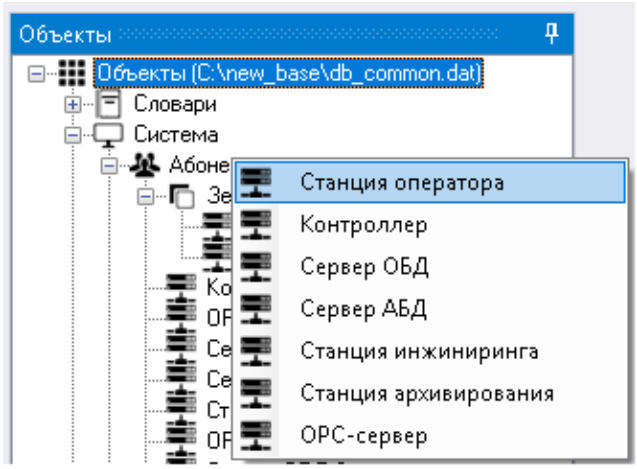


Рисунок 1.1.1

Протокол обмена «ОРС-сервер» – протокол обмена в стандарте OPC DA версии 2.0, обеспечивающий передачу данных из системы КРУГ-2000. Для данного типа протокола обмена допускается описывать один канал связи «ОРС-сервер» независимо от количества ОРС-клиентов.

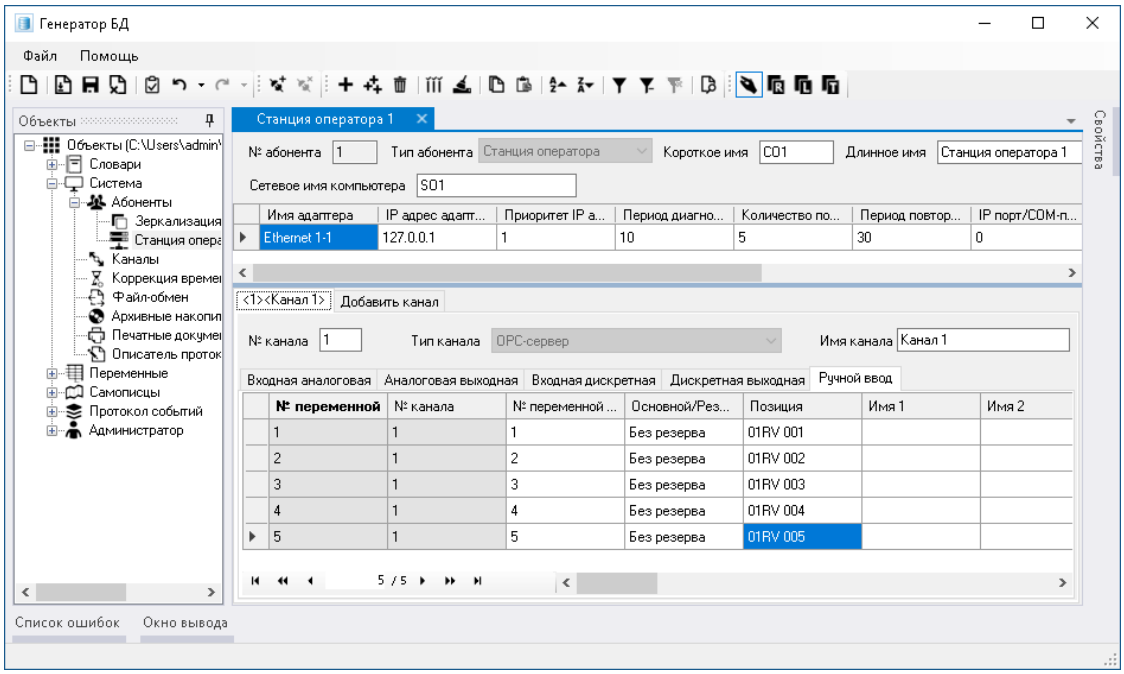


Рисунок 1.1.2 – Настройка канала «ОРС-сервер»

Назначение «виртуальных» переменных каналу связи «ОРС-сервер» выполняется после его создания. При работе с формами описания переменных вы можете назначить необходимое число переменных каналу связи «ОРС-сервер», но не больше максимального количества переменных для Вашей системы (ограничивается лицензией на количество переменных, так как ограничение распространяется на переменные с ненулевым номером канала связи).

На вкладке канала доступен набор вкладок по типам **переменных**, который позволяет назначить каналу связи «ОРС-сервер» все типы переменных, используемые в SCADA КРУГ-2000: входные аналоговые переменные (ВА), входные дискретные переменные (ВД), переменные ручного ввода (РВ), дискретные выходные переменные (ДВ), аналоговые выходные переменные (АВ).

А также для создания новой переменной можно воспользоваться веткой Переменные в дереве объектов. При выборе нужного типа переменных на экран вызывается табличная форма для работы с атрибутами переменной. Записи можно создать кнопками добавления переменной на панели инструментов ( или ) и соответствующим пунктом контекстного меню, вызываемым в области описания переменной.

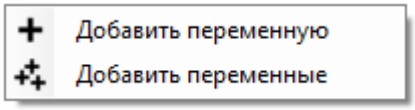


Рисунок 1.1.3 – Контекстное меню для создания переменных

У «виртуальных» переменных, которые Вы хотите использовать при работе с OPC-сервером, атрибут «**№ канала**» надо установить равным № канала связи «OPC-сервер» (при создании на вкладке канала назначается автоматически).

Например, на рисунке 1.1.4 переменные типа РВ №1-5, №11-15 будут доступны для OPC-сервера, т.к. для них № канала отличен от нуля. Другие переменные (№6-10) не доступны OPC-серверу, т.к. принадлежат к каналу с **№ 0**!

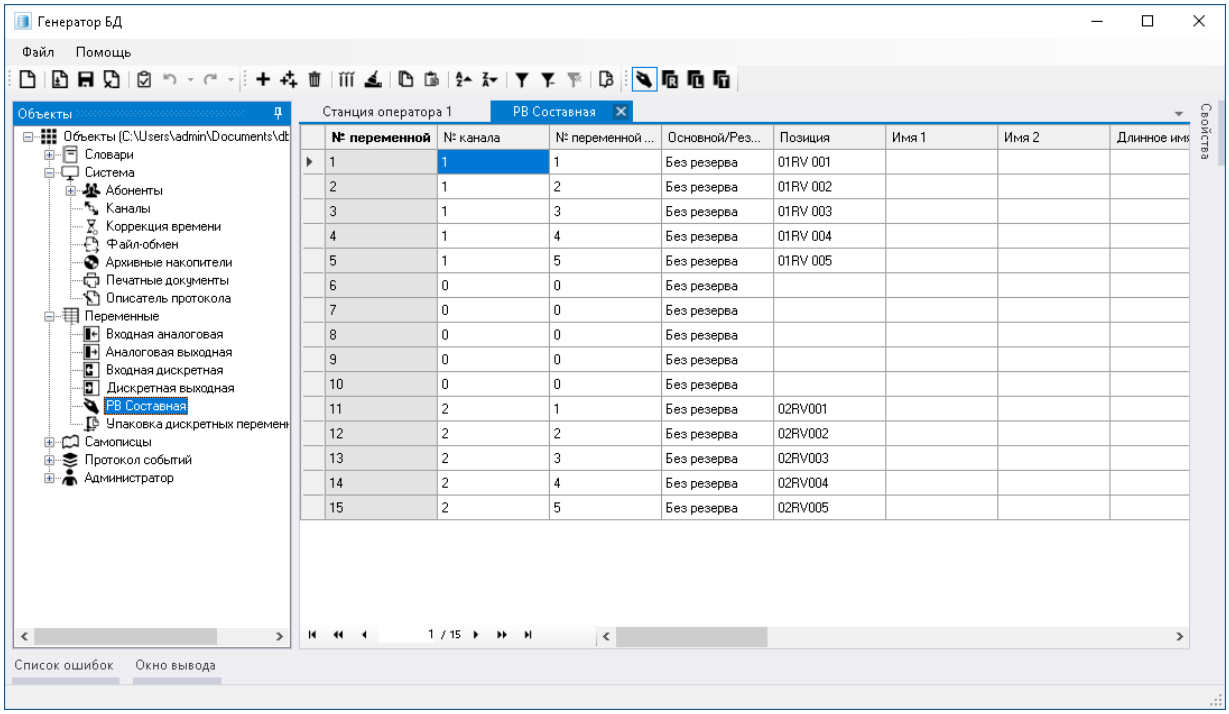


Рисунок 1.1.4 – Переменные ручного ввода, назначаемые каналу связи «OPC-сервер»

1.1.4 Работа с OPC DA сервером

Запуск OPC DA сервера производится автоматически при обращении к нему OPC DA клиента. Однако если в момент обращения не будет обнаружен работающий сервер базы данных (Сервер БД или Сервер АБД), то OPC DA сервер запущен не будет и OPC DA клиент выдаст ошибку соединения с OPC DA сервером.

ВНИМАНИЕ !!!

Всегда запускайте сервер базы данных перед началом работы с OPC DA сервером! OPC-сервер не поддерживает получение данных от Сервера БД, запущенного в режиме «Демо» (о режимах работы сервера БД можно прочитать в книге 8 «Среда исполнения», часть 2 «Программные модули и комплексы», раздел 1.1.1.1 «Параметры запуска Сервера БД»).

Для описания доступа к значениям переменных используйте следующий синтаксис:

ПРЕФИКС.ПОЗИЦИЯ_НОМЕР
где
ПРЕФИКС – краткое наименование типа переменной:
VA – входная аналоговая
VD – входная дискретная
AV – аналоговая выходная
DV – дискретная выходная
RV – ручной ввод
ПОЗИЦИЯ – позиция переменной, например, T112
НОМЕР – номер переменной в базе данных

Пример: VA.T112_**1**

Атрибуты переменных, которые поддерживает OPC DA сервер, описаны в приложениях A, B, C, D, E.

1.2 OPC DA КЛИЕНТ

1.2.1 Общие сведения

OPC DA клиент предназначен для обмена данными SCADA КРУГ-2000 в формате OPC DA со сторонними OPC-серверами.

Отличительной особенностью OPC DA клиента является интеграция с сервером БД SCADA КРУГ-2000, что позволяет упростить передачу данных и сделать связь с OPC DA серверами доступной и понятной для Пользователей.

Для того чтобы обозначить OPC DA сервер, как источник данных для OPC DA клиента, введен **тип абонента** «**OPC-сервер**».

В рамках системы КРУГ-2000 сторонний OPC DA сервер представляет собой канал данных. Канал может принадлежать одному из абонентов системы КРУГ-2000, и с этой точки зрения абонент системы КРУГ-2000 представляет собой компьютер (локальный или удаленный), где работает один или несколько OPC DA серверов. В этом случае один абонент (= компьютер) может иметь в распоряжении столько каналов, сколько OPC DA серверов необходимо опрашивать на данном компьютере.

Настройка OPC DA клиента происходит в несколько этапов:

1. Добавление абонента в систему
2. Создание каналов «OPCDA-клиент» (по числу опрашиваемых OPC DA серверов)
3. «Привязка» (указание соответствия) тегов или атрибутов тегов каждого OPC DA сервера к атрибутам переменных оперативной БД.

1.2.2 Добавление абонента

Для создания канала OPCDA-клиент необходим абонент типа «**OPC-сервер**» с настройками компьютера, где работает один или несколько OPC DA серверов. Если нужного абонента нет, то его необходимо добавить в базу данных. Для этого необходимо в контекстном меню пункта «Абоненты» (Объекты → Система → Абоненты) выбрать соответствующий тип (рисунок 1.2.1). В открывшейся форме описания абонента (рисунок 1.2.2) следует указать **Сетевое имя компьютера** в соответствии с сетевыми настройками операционной системы, установленной на выбранном компьютере (по умолчанию создается абонент с сетевым именем локальной машины – localhost).

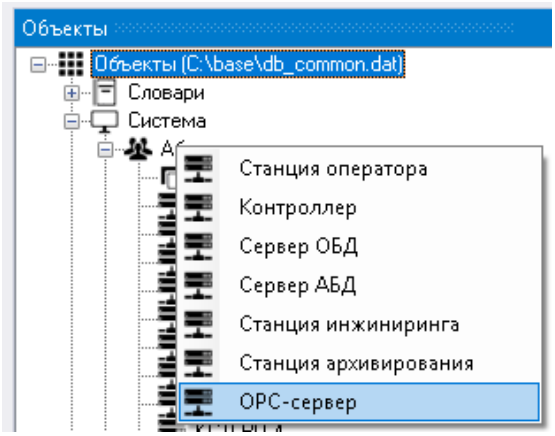


Рисунок 1.2.1 – Контекстное меню создания абонента

После добавления нового абонента в таблице адаптеров абонента следует указать соответствующий ему адаптер. Ключевым здесь является поле «**IP-адрес адаптера**». По умолчанию создается адаптер для локального абонента (компьютера) с IP-адресом – **127.0.0.1.** Для удаленного абонента (компьютера) укажите **IP-адрес компьютера,** заданный в сетевых настройках операционной системы**.**

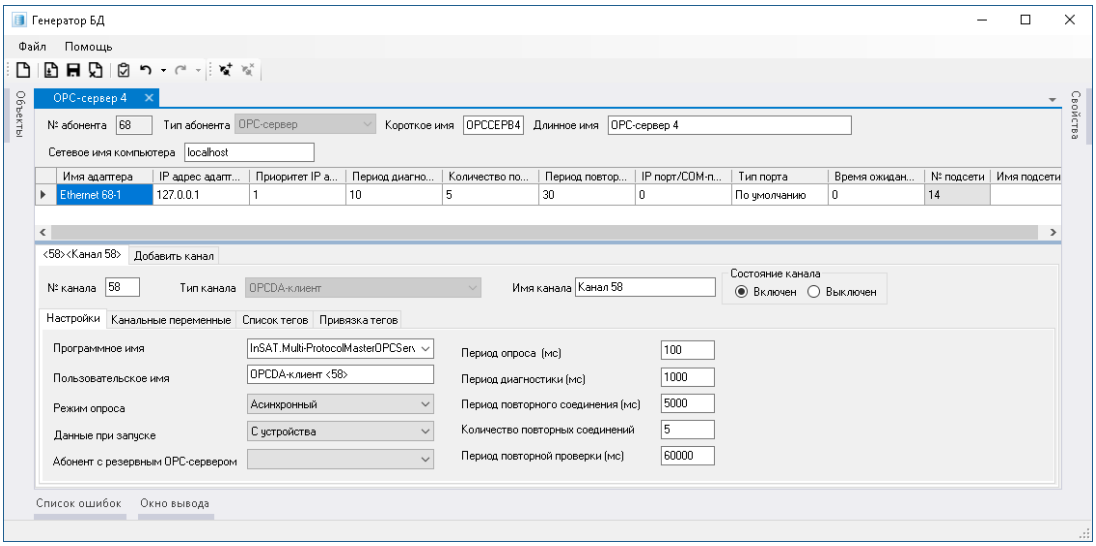


Рисунок 1.2.2 – Форма настройки описания абонента

1.2.3 Создание каналов «OPC DA-клиент»

Чтобы добавить новый канал на форме описания абонента (рисунок 1.2.2) необходимо левой кнопкой мыши выбрать вкладку «Добавить канал», в появившемся контекстном меню (рисунок 1.2.3) необходимо выбрать тип создаваемого канала – OPCDA-клиент.

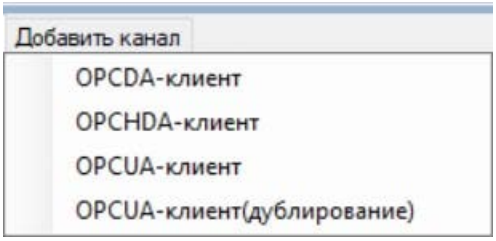


Рисунок 1.2.3 – Контекстное меню создания канала

На вкладке «Настройки» необходимо указать **Программное имя** ОРС-сервера и заполнить дополнительные поля для настройки параметров обмена данными с выбранным ОРС DA сервером. Посредством программного имени ОРС-сервер сообщает о себе операционной системе и ее процессам. В список выбора программного имени заносятся все ОРС-серверы, установленные на компьютере, указанном в свойствах абонента. При раскрытии списка происходит автоматический поиск зарегистрированных ОРС-серверов на компьютере, сетевое имя которого написано в текущем абоненте.

Во время подключения к ОРС DA серверу, может произойти ошибка (рисунок 1.2.4). Обычно это связано с тем, что ОРС DA сервер не настроен на своем компьютере или не настроены опции DCOM.

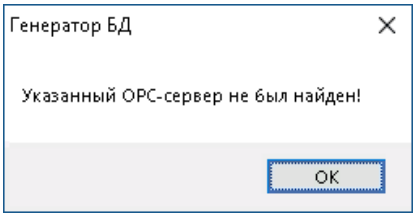


Рисунок 1.2.4 – Окно с сообщением об ошибке соединения с ОРС DA сервером

Рассмотрим подробнее параметры, которые следует настроить для обмена данными с ОРС DA сервером:

- **Пользовательское имя** – имя ОРС-сервера, задаваемое пользователем (255 символов). Именно оно используется при выдаче ОРС-клиентом сообщений в протокол событий. Если пользовательское имя не задано, то в сообщениях протокола событий будет использоваться программное имя ОРС-сервера.
- **Режим опроса** – выбирается из списка: синхронный или асинхронный (по умолчанию). Синхронный доступ к данным ОРС-сервера означает, что чтение будет производиться через определенный период времени, вне зависимости от того, изменились данные или нет. Асинхронный доступ, являясь современным методом, подразумевает, что операция чтения будет производиться только тогда, когда данные изменились, о чем ОРС-сервер уведомит ОРС-клиента.
- **Данные при запуске** – (при включении канала ОРС-клиент или при восстановлении связи с ОРС-сервером после обрыва данные через ОРС-сервер запрашиваются: «с устройства» (по умолчанию) или «с кэша». Значение «с устройства» по умолчанию.
- **Абонент с резервным ОРС-сервером** – (абонент системы КРУГ-2000, где установлен резервный ОРС-сервер. Резервный ОРС-сервер должен иметь точно такое же программное имя, как и основной ОРС-сервер, и иметь идентичный набор тегов. В момент, когда ОРС-клиент после указанного количества повторных соединений не может установить связь с основным ОРС-сервером, происходит переключение на удалённый компьютер, указанный в настройках абонента, выбранного в данном поле. Основному ОРС-серверу при этом присваивается статус резервного, а резервному ОРС-серверу на удалённом компьютере – статус основного.

ВНИМАНИЕ !!!

Резервный OPC DA сервер должен иметь точно такое же программное имя, как и основной OPC DA сервер, а также идентичный набор тегов.

Если необходимо одновременно запрашивать данные с «основного» и «резервного» OPC серверов, то необходимо определить два канала «OPC-клиент».

При использовании нестандартных обработок в канале «OPC-клиент» в качестве времени возникновения события, вызванного нестандартными обработками, будет использоваться метка времени тега OPC DA сервера

- **Период опроса (мс) –** период опроса OPC-сервера при синхронном режиме работы (по умолчанию 100 мс).
- **Период диагностики (мс)** – интервал времени, с которым OPC-клиент будет проверять состояние OPC-сервера (по умолчанию 1 секунда).
- **Период повторного соединения (мс)** – период времени в миллисекундах, использующийся для повторных попыток соединения с OPC-сервером при обрыве связи (по умолчанию 5 секунд).
- **Количество повторных соединений** – количество попыток повторного соединения, по достижении которого попытки возобновить связь прекратятся (по умолчанию 5).
- **Период повторной проверки (мс)** – интервал времени, по истечении которого OPC-клиент снова будет пытаться соединиться с OPC-сервером указанное количество раз в поле «Количество повторных соединений» с периодом, указанным в поле «Период повторного соединения» (по умолчанию 1 минута).

В момент, когда OPC DA клиент после указанного количества повторных соединений не сможет установить связь с основным OPC DA сервером, происходит переключение на компьютер, указанный в настройках абонента как резервный OPC DA сервер. При этом основной OPC DA сервер становится резервным, а резервный – основным. В протокол событий системы выдается сообщение о переходе на резервный OPC DA сервер.

ВНИМАНИЕ!!!

После переключения на резервный OPC DA сервер клиент будет продолжать работу с ним, даже если связь со другим OPC DA сервером восстановлена.
Смена OPC DA сервера происходит только при потере связи с основным на данный момент OPC DA сервером.
OPC DA клиент считает основным тот OPC DA сервер, с которым связь была установлена последним.

1.2.4 Редактирование и привязка тегов OPC DA сервера

1.2.4.1 Редактирование тегов

Генератор базы данных предоставляет возможность редактирования тегов OPC DA сервера. Для редактирования тегов следует на форме канала OPCDA-клиент перейти на вкладку «Список тегов» (рисунок 1.2.5)

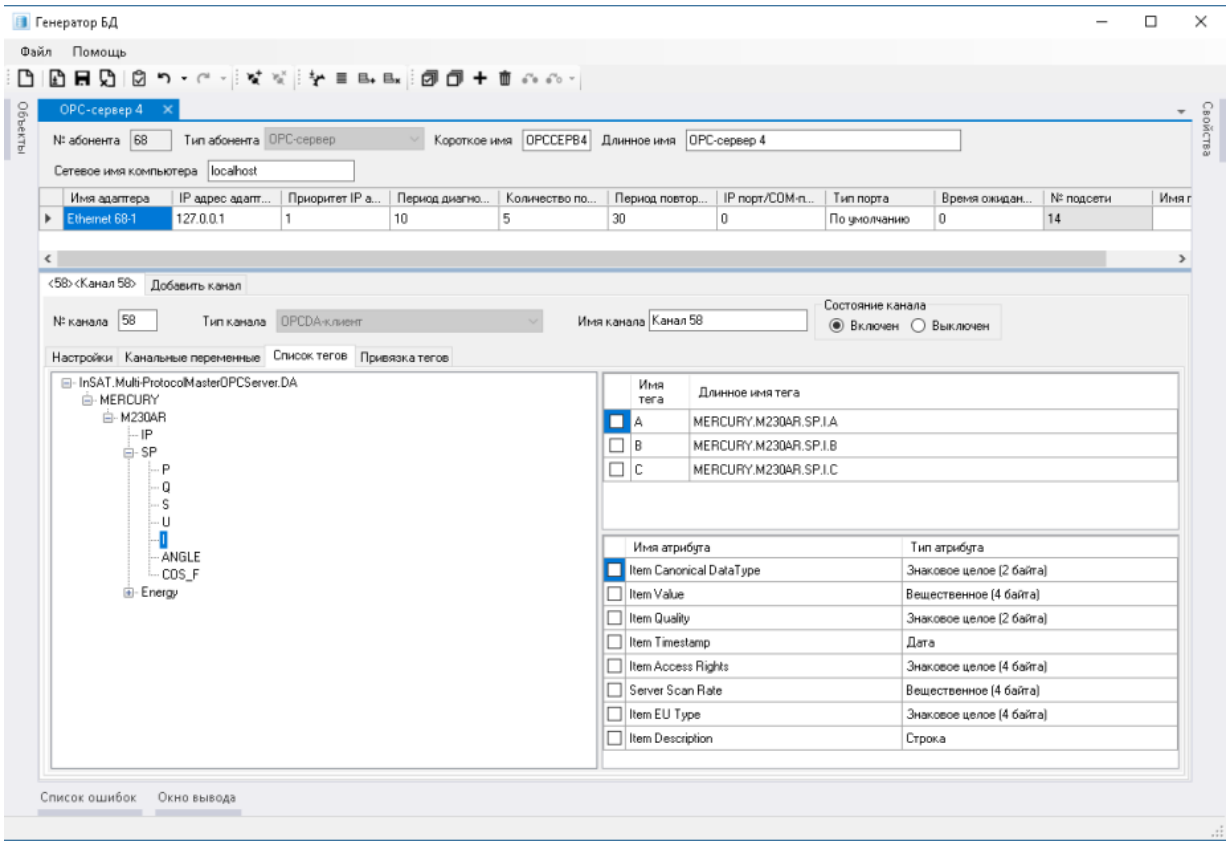


Рисунок 1.2.5 - Форма управления списком тегов OPC DA сервера

Вкладка разделена на две части: область дерева тегов, область описания тегов выбранной ветки дерева и атрибутов выделенного тега для DA.
Для создания привязки необходимо установить флажки напротив тегов и атрибутов тегов для DA (по умолчанию при выборе флажка у тега отмечается атрибут Текущее значение (Item Value)). Дополнительные действия можно осуществить с помощью контекстного меню, которое вызывается с помощью нажатия правой кнопки мыши в соответствующей области.
Контекстное меню области дерева тегов:

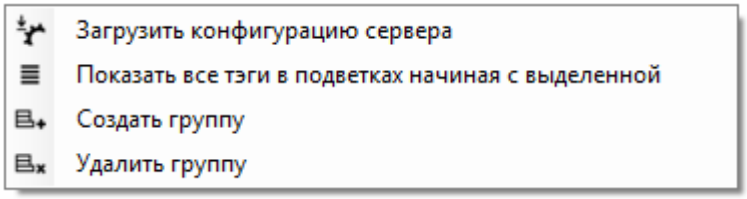


Рисунок 1.2.6 – Контекстное меню области дерева тегов

- **Загрузить конфигурацию сервера** – позволяет обновить дерево тегов с ОРС-сервера.
- **Показать все теги в подветках, начиная с выделенной** – в описании тегов отобразятся все теги выделенной ветки и всех ее подветок.
- **Создать группу –** позволяет создать пользовательскую ветку тегов в текущей выбранной ветке.
- **Удалить группу** – удаляет текущую выбранную ветку.

ВНИМАНИЕ !!!

Удаление тегов или группы тегов (и другие изменения) не влияют на информацию на ОРС-сервере. При следующем открытии диалога все дерево будет зачитано заново с той структурой, что задана на сервере.

Контекстное меню области тегов:

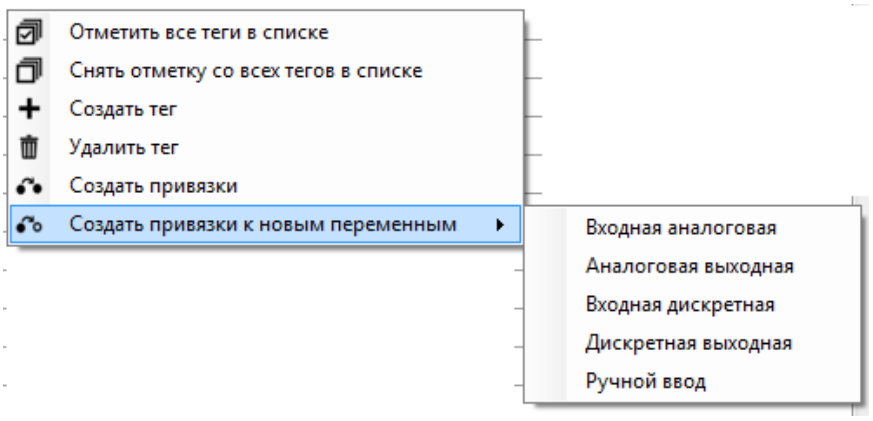


Рисунок 1.2.7 – Контекстное меню области тегов

- **Отметить все теги в списке** – устанавливает флажок выбора у всех тегов текущей выбранной ветки дерева.
- **Снять отметку со всех тегов в списке** – снимает флажок выбора у всех тегов текущей выбранной ветки дерева.
- **Создать тег** – позволяет создать пользовательский тег.
- **Удалить тег** – позволяет удалить выделенный тег.
- **Создать привязки** – создает пустые привязки для тегов и атрибутов, для которых установлен флажок выбора. Дальнейшая привязка осуществляется на вкладке «Привязка тегов».
- **Создать привязки к новым переменным** – создает привязки для тегов и атрибутов, отмеченных флажком, к новым переменным выбранного типа с атрибутами переменных по умолчанию.

ВНИМАНИЕ !!!

Каждая пользовательская группа или пользовательский тег (добавленные вручную) после закрытия вкладки канала OPC не сохраняется. Т.о. при каждом новом открытии формы редактирования тегов после автоматического считывания доступны будут только теги сервера.
Привязки, созданные на основе пользовательских тегов, сохраняются, и их можно просмотреть на форме «Привязка тегов»

1.2.4.2 Привязка тегов

Привязка тегов OPC DA сервера к переменным базы данных необходима для того, чтобы OPC DA клиент знал, какие теги OPC DA сервера следует обрабатывать, и как сопоставить их с переменными базы данных.

Для задания привязки между тегами OPC-сервера и переменными базы данных используют вкладку «**Привязка тегов»** (рисунок 1.2.8).

ВНИМАНИЕ !!!

Назначить тегам в соответствие можно только переменные, принадлежащие текущему выбранному каналу.
Одна строка таблицы привязок описывает одну привязку.

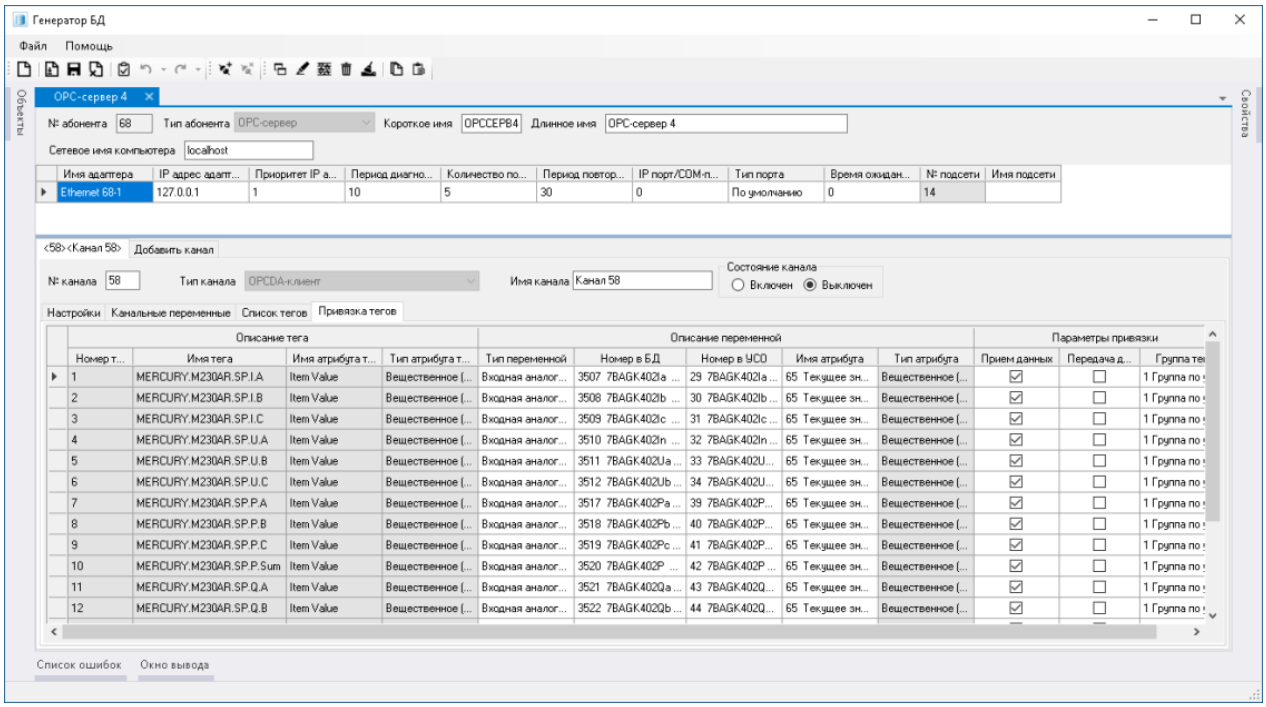


Рисунок 1.2.8 - Привязка тегов ОРС-сервера к переменным БД

Поля в таблице привязок структурированы по разделам.

Описание тега (информационные поля)

- **Номер тега** – уникальный номер для идентификации привязки, задается автоматически
- **Имя тега** – имя опрашиваемого тега. Содержимое поля определяется выбором записей на форме со списком тегов OPC DA-сервера (рисунок 1.2.5)
- **Имя атрибута тега** – имя атрибута опрашиваемого тега. Если у тега нет описания атрибута с ID, равным 2 (обычно это атрибут «Item Value»), то ГБД добавляет «виртуальный атрибут» («Item Value» ID = 2, тип - вариант). Содержимое поля определяется выбором записей на форме со списком атрибутов тега OPC DA-сервера (рисунок 1.2.5)
- **Тип атрибута тега** – целое, вещественное, строковое и т.д..

Описание переменной

- **Тип переменной** – набор типов зависит от наличия в канале переменных соответствующих типов
- **№ в БД** – раскрывающийся список, в котором можно выбрать переменную по общему номеру в БД, по позиции и по длинному имени.
- **№ в УСО** – раскрывающийся список, в котором можно выбрать переменную по номеру в УСО, по позиции и по длинному имени
- **Имя атрибута –** раскрывающийся список, в котором можно выбрать атрибут переменной по его имени и типу (целое, вещественное, строковое и т.д.)
- **Тип атрибута** - тип данных атрибута переменной, заполняется автоматически при выборе имени атрибута. Для логического типа может принимать значения «логическое» (тип данных атрибута по умолчанию - как он задан в паспорте переменной) или «логическое с отрицанием» (2 байта).

Параметры привязки

- **Прием данных** – определяет возможность приема данных от тега к переменной. Если стоит галочка, то прием данных будет работать. По умолчанию для переменных типа ВА и ВД в поле «Прием» стоит галочка, а для переменных ДВ и РВ - не стоит
- **Передача данных** – определяет возможность передачи данных от переменной к тегу. Если стоит галочка, то передача данных будет работать. По умолчанию для переменных типа ВА и ВД в поле «Передача» галочка не стоит, а для переменных ДВ и РВ - стоит. При назначении в привязке атрибута тега, не представляющего текущее значение тега, в поле «Передача» галочка не ставится, а само это поле становится недоступным для редактирования
- **Имя группы тегов** – номер и имя группы тегов. Создать группы тегов можно в форме «Группы тегов» (рисунок 1.2.9), которая открывается при нажатии на одноименную кнопку на панели инструментов. Группы тегов служат для объединения тегов в группы с целью установления каких-либо общих свойств для всех тегов в группе.

☝**ВНИМАНИЕ!!!**

В случае привязки атрибута тега, не представляющего текущее значение, в поле «Передача» галочка не ставится, а само это поле становится недоступным для редактирования. Это связано с тем, что спецификация OPC Data Access версии 2.0, которую использует OPC DA клиент, предусматривает запись данных только для текущего значения тега.

1.2.4.3 Управление привязками

Группы тегов служат для объединения тегов в группы с целью установления каких-либо общих свойств для всех тегов в группе. При нажатии на панели инструментов на кнопку «**Группы тегов**» вызывается одноименная форма (рисунок 1.2.9).

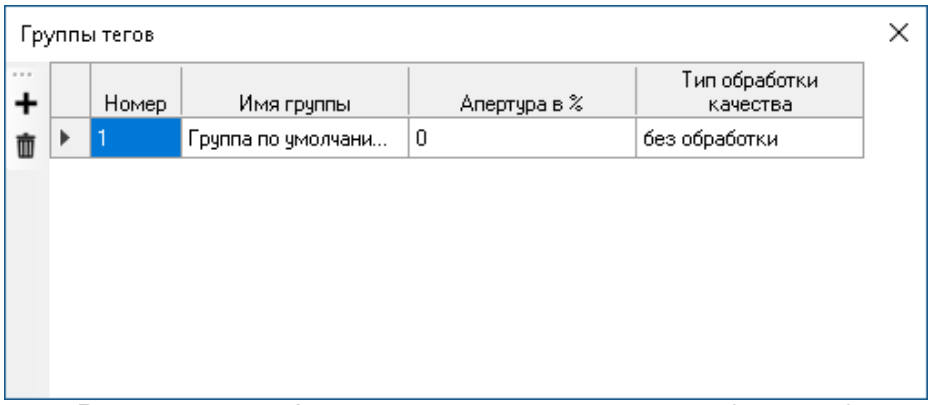


Рисунок 1.2.9 - Форма для задания групп тегов (апертур)

В начале работы OPC DA клиент сформирует группы тегов в OPC DA сервере на основе групп, заданных Пользователем в данной форме, и придаст им свойства, которые указал Пользователь. По умолчанию создается группа, апертура которой равна 0%.

☝**ВНИМАНИЕ!!!**

Апертура используется только при асинхронном опросе OPC-сервера. Обработка апертур осуществляется OPC-сервером. Если OPC-сервер не поддерживает обработку апертур, заданные апертуры для групп тегов действовать не будут.

В зависимости от значения поля **Тип обработки качества** обработка качества тегов, связанных с одной переменной, будет выполняться по различным алгоритмам:

- «**без обработки**» – значение по умолчанию. Обобщенное качество переменной определяется по наихудшему качеству тегов в группе привязок.
- **«с обработкой»** – на определение обобщенного качества по переменной влияет наличие привязки ОРС-тега в группе к атрибуту «Текущее значение» переменной соответственно, для:
  - **ВА** – атрибут №**28** «Текущее значение до преобразования (контроллер)»
  - **ВД** – атрибут №**27** «Текущее значение переменной»
  - **РВВ** – атрибут №**27** «Текущее значение»
  - **РВС** – атрибут №**13** «Текущее значение (Строка)»
  - **РВЛ** – атрибут №**14** «Текущее значение (логич)»
  - **ДВ** – атрибут №**20** «Значение выходной переменной в контроллере»
  - **АВ** – атрибут №**48** «Значение выходного сигнала (аналогового регулятора)»

При наличии такой привязки, определяющим для обобщенного качества является качество данного тега, при его отсутствии – обобщенное качество принимается равным «**GOOD**».

Панель «**Автозаполнение**» (рисунок 1.2.10) служит для автоматического заполнения привязок и вызывается с помощью соответствующей кнопки на панели инструментов .

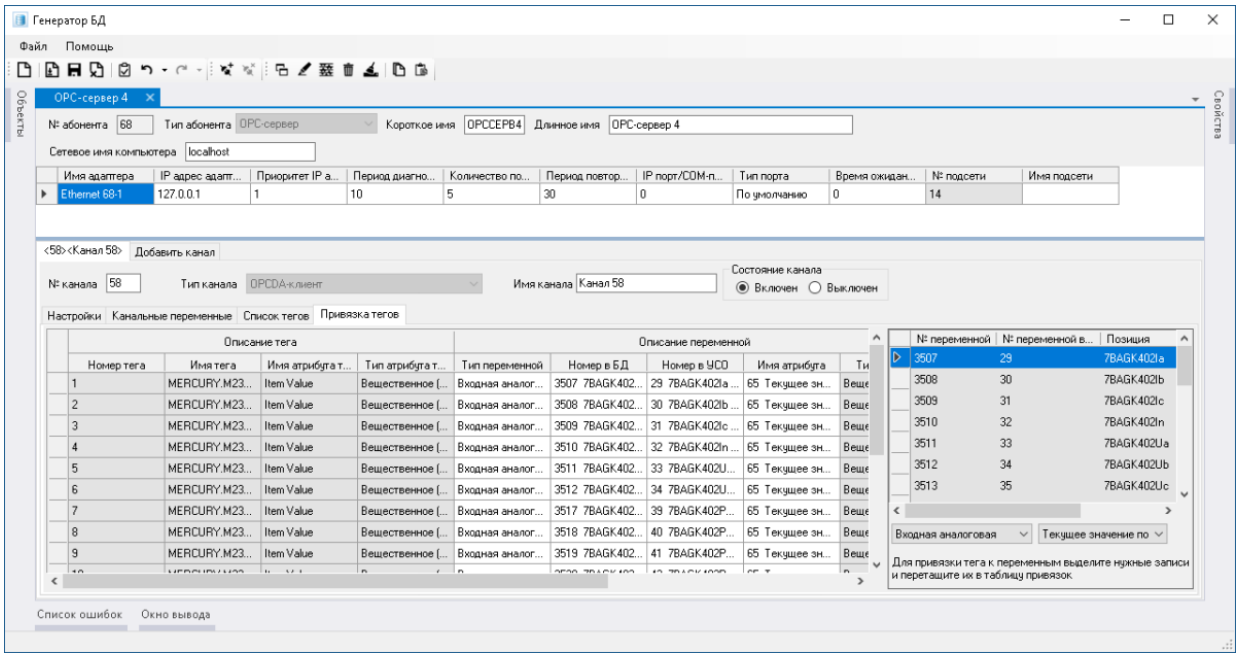


Рисунок 1.2.10 - Форма автозаполнения привязок

Заполнение таблицы привязок происходит с помощью выбора переменных из списка и переноса записей в таблицу привязок путём «захвата» (нажатием и удержанием левой кнопки мыши) выделенных в списке записей выбранного типа переменных и перемещении переменных поверх текущих привязок. Привязка будет осуществлена к атрибуту переменной, выбранному на панели автозаполнения.
В списке автозаполнения доступны только переменные текущего канала.
Направление обмена будет выставлено по типу переменной по умолчанию (для переменных типа ВА и ВД в поле «Прием», для переменных типа ДВ и РВ в поле «Передача»). Группа тегов по умолчанию будет задана №1.

В результате заполнения привязок могут возникнуть ошибочные ситуации (например, соответствия типов значений тегов и привязанных к ним атрибутов переменных). Окно ошибок и предупреждений необходимо для оповещения пользователя о том, что была

допущена ошибка или неточность (рисунок 1.2.11). Ошибка (предупреждение) имеет описание, указывающее на причину возникновения. Одна из проверок – выявление непреобразуемых типов данных. Например, из тега, передающего вещественное значение хотят записывать информацию в метку времени переменной BA, которое является типом «Дата/Время».

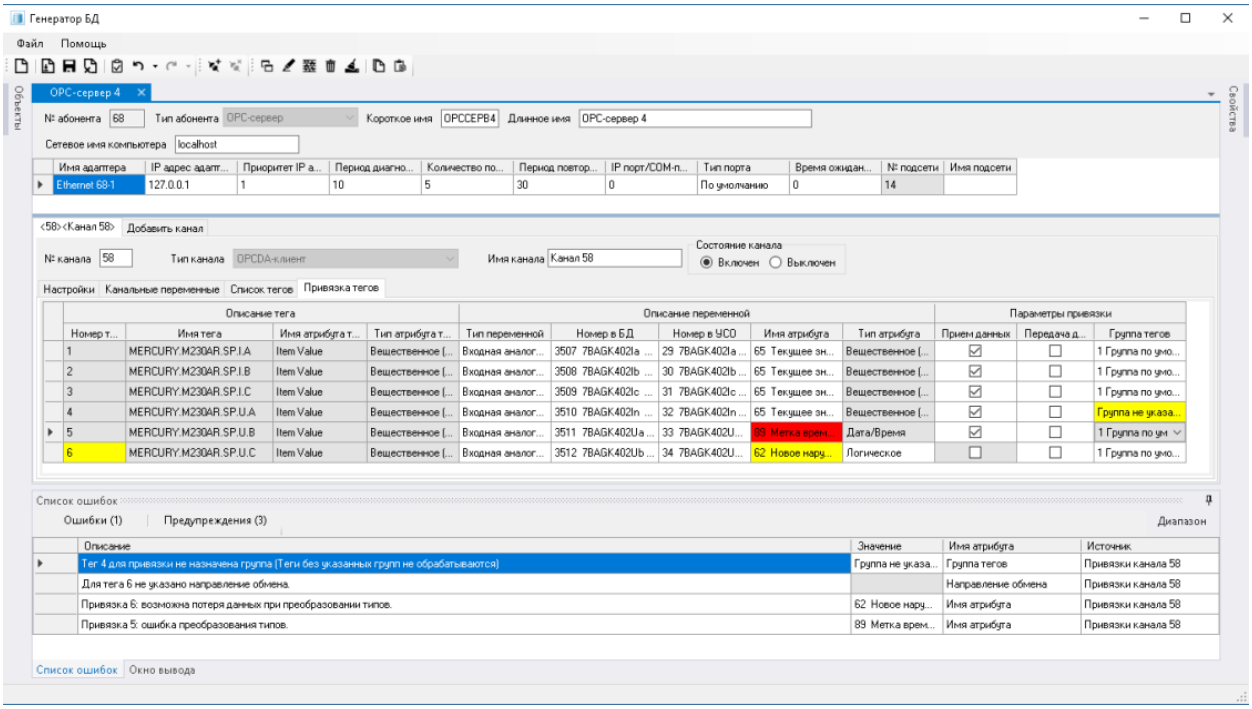


Рисунок 1.2.11 – Вывод сообщений о несоответствии типов

1.2.5 Обновление значений тегов по команде Пользователя

OPC DA клиент предоставляет возможность обновления значений тегов с OPC DA сервера по команде Пользователя.

Для обновлений значений используется атрибут **№42** «**ResBt**» таблицы «**Канал**» базы данных. Алгоритм использования этого атрибута OPC DA клиентом следующий:

- По умолчанию, значение атрибута **№42** равно **0**
- OPC DA клиент периодически (**1 раз в секунду**) проверяет значение этого атрибута
- Если значение атрибута **№42** равно **1**, то OPC-клиент осуществляет обновление значений тегов с OPC-сервера. После выполнения обновления значение атрибута автоматически сбрасывается в **0**.

Таким образом, для управления обновлением тегов по команде Пользователя следует:

- Создать в Генераторе динамики на мнемосхеме графический элемент, используемый для управления обновлением тегов, например, кнопку
- Назначить для этого элемента функцию реакции «**Установить значение**» на событие «**Нажатие левой клавиши мыши**»
- В окне свойств элемента в поле «**Приемник**» назначить ссылку на переменную «**System\Канал\ResBt\n**», где **n** – номер канала
- В поле «**Записываемое значение**» задать **1** (рисунок 1.2.19).

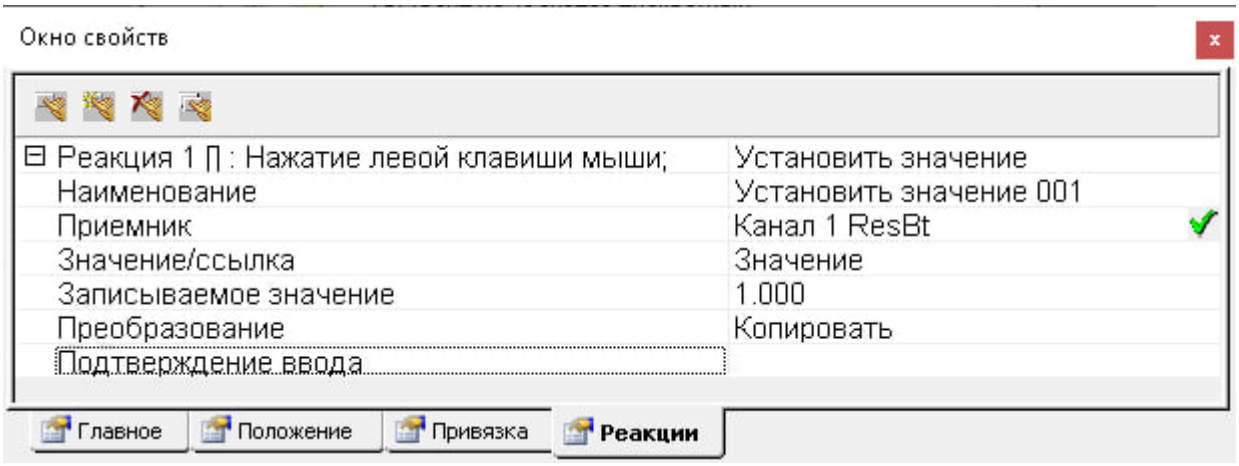


Рисунок 1.2.19 – Окно свойств Генератора динамики. Назначение функции реакции

При работе автоматизированной системы, нажав на созданную таким образом кнопку, Пользователь изменит значение атрибута **№42** на **1** и этим инициирует обновление значений атрибутов тегов с OPC DA сервера.

ВНИМАНИЕ!!!

Если атрибут №42 не используется для обновления значений атрибутов тегов, то OPC DA клиент автоматически обновляет все значения атрибутов тегов при включении канала или восстановлении связи по каналу после обрыва.

1.2.6 Обработка атрибута «Метка времени»

Если сервер Базы Данных запущен с параметром «**–timechange**», то OPC-клиент будет обрабатывать «метку времени» следующим образом:

1) В атрибут «**Метка времени**» переменной записывается значение «метки времени» тэга от источника данных, привязанного к данной переменной Сервера БД
2) При передаче значения атрибута переменной от OPC-клиента в тег OPC-сервера в атрибут «**Метка времени**» переменной записывается текущее время OPC-клиента.

ВНИМАНИЕ!!!

Если Сервер БД запущен без параметра «-timechange», то OPC-клиент «метку времени» не обрабатывает.

2 OPC HISTORICAL DATA ACCESS

2.1 OPC HDA СЕРВЕР

2.1.1 Общие сведения

OPC HDA (OPC Historical Data Access) сервер – это специализированный OPC-сервер, предоставляющий исторические данные технологического процесса (оперативные и архивные значения трендов SCADA КРУГ-2000).
Для работы с OPC HDA серверами используются специализированные OPC-клиенты – OPC HDA клиенты, ориентированные на работу с историческими данными.

OPC HDA сервер SCADA КРУГ-2000 предоставляет OPC HDA клиентам доступ к оперативным и архивным значениям трендов КРУГ-2000, к любым другим данным доступ не предоставляется.

☝**ВНИМАНИЕ!!!**

Параметры OPC HDA сервера SCADA КРУГ-2000 загружаются автоматически и Пользователем НЕ настраиваются.
OPC HDA сервер SCADA КРУГ-2000 обеспечивает только чтение значений трендов, формирование и изменение значений трендов сервер не поддерживает.
OPC HDA сервер предоставляет доступ к значениям трендов при наличии работающего сервера базы данных КРУГ-2000 (Сервер БД/АБД)

☝**ВНИМАНИЕ!!!**

OPC HDA сервер SCADA КРУГ-2000 будет запущен только в случае разрешения этого в Вашем аппаратном ключе защиты!

Для организации доступа к архивным значениям трендов требуется наличие Сервера АБД, запуск которого также должен быть разрешён в Вашем аппаратном ключе защиты.

OPC HDA сервер системы КРУГ 2000 поддерживает обмен данными согласно спецификации OPC HDA версии 1.2 (**OPC Historical Data Access Specification. version 1.2**).

Схема доступа к оперативным и архивным значениям трендов SCADA КРУГ-2000 с помощью OPC HDA сервера представлена на рис. 2.1

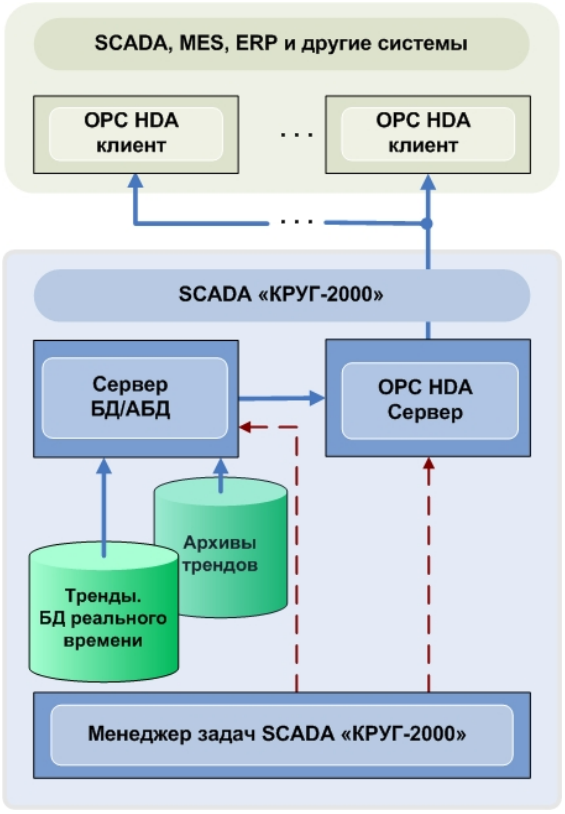


Рисунок 2.1 – Доступ к историческим данным SCADA КРУГ-2000

2.1.2 Состав OPC HDA сервера

OPC HDA сервер системы КРУГ-2000 включает следующие файлы:

- **KrOPCHDAServer.exe**- исполняемый файл OPC HDA сервера
- **KrugHDAsvr.dll** – вспомогательная библиотека для работы OPC HDA сервера
- **opccomn_ps.dll** – стандартная библиотека для локального и удалённого обмена данными
- **opcenum.exe** – стандартная утилита для поиска зарегистрированных на компьютере OPC-серверов
- **opcproxy.dll** – стандартная библиотека для локального и удалённого обмена данными
- **opc_aeps.dll** – стандартная библиотека для локального и удалённого обмена данными
- **opchda_ps.dll** – стандартная библиотека для локального и удалённого обмена данными.

2.1.3 Работа с OPC HDA сервером

Запуск OPC HDA сервера производится автоматически при обращении к нему OPC HDA клиента либо с помощью Менеджера задач SCADA КРУГ-2000.
Если в момент обращения OPC HDA клиента не будет обнаружен работающий Сервер базы данных, то OPC HDA сервер запущен не будет и OPC HDA клиент не сможет соединиться с OPC HDA сервером.

☝**ВНИМАНИЕ!!!**

Всегда запускайте Сервер базы данных перед началом работы с OPC HDA сервером! OPC HDA сервер не поддерживает получение данных от Сервера БД, запущенного в режиме «Демо» (о режимах работы сервера БД можно прочитать в книге 8 «Среда исполнения», часть 2 «Программные модули и комплексы», раздел 1.1.1.1 «Параметры запуска Сервера БД»).

Для описания доступа к значениям трендов используйте следующий синтаксис:

<ТИП>.<ПОЗИЦИЯ>_<НОМЕР1>.R<НОМЕР2>P<НОМЕР3>

где **ТИП** – краткое наименование типа переменной:

VA – входная аналоговая;
VD – входная дискретная
AV –аналоговая выходная;
DV – дискретная выходная;
RV – ручной ввод

ПОЗИЦИЯ – позиция переменной, например, T112
НОМЕР1 – номер переменной в базе данных
НОМЕР2 – номер самописца в базе данных
НОМЕР3 – номер пера в самописце

Например,

VA.T112_1.R1P12

DV.K15_25.R3P24

2.1.4 Доступ к историческим данным SCADA КРУГ-2000 версий 3.0 и 2.5.

☝**ВНИМАНИЕ!!!**

OPC HDA сервер может работать только в среде SCADA КРУГ-2000 версии 4.0 и выше

Для того чтобы OPC HDA сервер получил доступ к историческим данным SCADA КРУГ-2000 версии 3.0 (или версии 2.5) должны быть выполнены следующие условия:

- SCADA КРУГ-2000 версии 3.0 или 2.5 установлена на отдельном компьютере
- В SCADA КРУГ-2000 версии 3.0 или 2.5 функционирует Сервер архивной базы данных (Сервер АБД). OPC HDA серверу исторические данные этой системы будут недоступны, если ее сервер БД работает в демо- или оперативном режиме
- В SCADA КРУГ-2000 версии 4.1 выше настроен удаленный доступ к базе данных сервера SCADA КРУГ-2000 версии 3.0 или 2.5 (описание настройки приведено в Руководстве Пользователя «Введение в КРУГ-2000» в разделе «Свойства проекта и конфигурации клиентов»).

При выполнении этих условий OPC HDA сервер будет получать значения трендов из системы, построенной на основе SCADA КРУГ-2000 версии 3.0 или 2.5.

2.2 OPC HDA КЛИЕНТ

2.2.1 Общие сведения

OPC HDA клиенты – это специализированные OPC-клиенты, ориентированные на работу с историческими данными, которые могут быть получены от OPC HDA сервера.

OPC HDA клиент SCADA КРУГ-2000 (далее OPC HDA клиент) предназначен для получения данных от сторонних OPC HDA серверов и передачи этих данных в событийные тренды сервера БД SCADA КРУГ-2000.

В рамках SCADA КРУГ-2000 сторонний OPC HDA сервер представляет собой канал данных. Канал может принадлежать одному из абонентов SCADA КРУГ-2000, и с этой точки зрения абонент представляет собой компьютер (локальный или удалённый), где работает один или несколько OPC HDA серверов. В этом случае один абонент (= компьютер) может иметь в распоряжении столько каналов, сколько необходимо опрашивать OPC HDA серверов на данном компьютере.

Настройка OPC HDA клиента в Генераторе базы данных происходит в несколько этапов:

1. Добавление абонента в систему
2. Создание каналов данных (по числу опрашиваемых OPC HDA серверов)
3. Указание соответствия между тегами каждого OPC HDA сервера и перьями событийных самописцев системы КРУГ-2000.

ВНИМАНИЕ!!!

OPC HDA клиент может быть запущен Менеджером задач SCADA КРУГ-2000.
Для корректной работы OPC HDA клиента нужно, чтобы работал Сервер БД.
Если используется резервирование серверов БД SCADA КРУГ-2000, то OPC HDA клиент необходимо запускать с параметром «–dde:», чтобы настроить его на обмен данными с локальным сервером БД.
Если статус Сервера БД «резервный», то клиент остаётся в оперативной памяти и не предпринимает никаких действий до тех пор, пока сервер БД не приобретёт статус «Основной» или клиент не получит команду завершения работы.

2.2.2 Добавление абонента.

Для создания канала OPCHDA-клиент необходим абонент типа «**OPC-сервер**» с настройками компьютера, где работает один или несколько OPC HDA серверов. Если нужного абонента нет, то его необходимо добавить в базу данных. Для этого необходимо в контекстном меню пункта «Абоненты» (Объекты → Система → Абоненты) выбрать соответствующий тип (рисунок 2.2.1). В открывшейся форме описания абонента (рисунок 2.2.2) следует указать **Сетевое имя компьютера** в соответствии с сетевыми настройками операционной системы, установленной на выбранном компьютере (по умолчанию создается абонент с сетевым именем локальной машины – localhost).

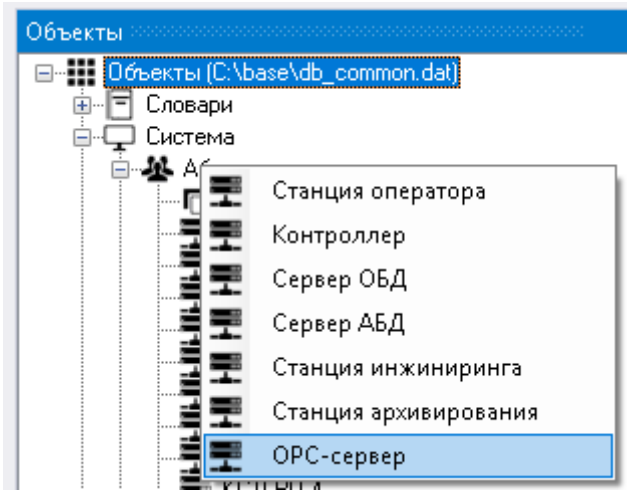


Рисунок 2.2.1 – Контекстное меню создания абонента

После добавления нового абонента в таблице адаптеров абонента следует указать соответствующий ему адаптер. Ключевым здесь является поле «**IP-адрес адаптера**». По умолчанию создается адаптер для локального абонента (компьютера) с IP-адресом – **127.0.0.1.** Для удаленного абонента (компьютера) укажите **IP-адрес компьютера,** заданный в сетевых настройках операционной системы**.**

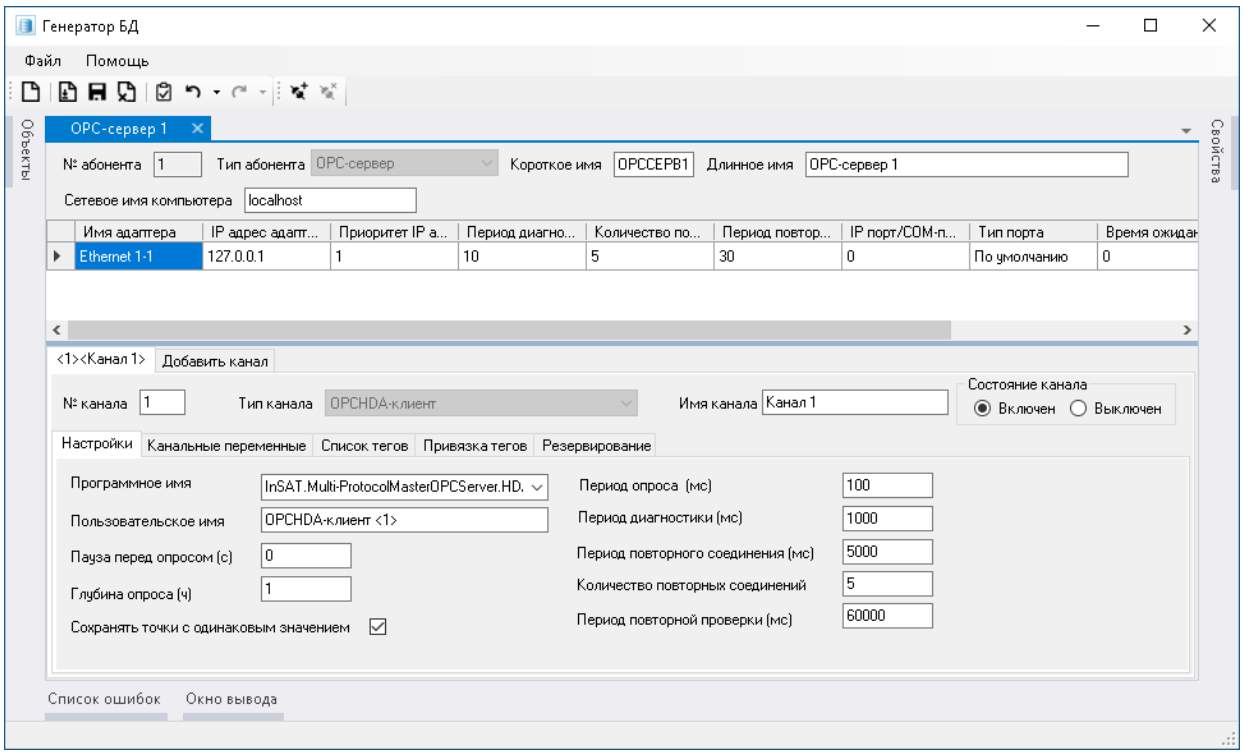


Рисунок 2.2.2 – Форма настройки описания абонента

2.2.3 Создание и настройка каналов

Чтобы добавить новый канал на форме описания абонента (рисунок 2.2.2) необходимо левой кнопкой мыши выбрать вкладку «Добавить канал», в появившемся контекстном меню (рисунок 2.2.3) необходимо выбрать тип создаваемого канала – OPCHDA-клиент.

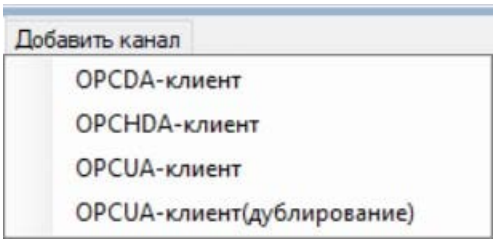


Рисунок 2.2.3 – Контекстное меню создания канала

На вкладке «Настройки» необходимо указать **Программное имя** OPC-сервера и заполнить дополнительные поля для настройки параметров обмена данными с выбранным OPC HDA сервером. Посредством программного имени OPC-сервер сообщает о себе операционной системе и ее процессам. В список выбора программного имени заносятся все OPC-серверы, установленные на компьютере, указанном в свойствах абонента. При раскрытии списка происходит автоматический поиск зарегистрированных OPC-серверов на компьютере, сетевое имя которого написано в текущем абоненте.

Во время подключения к OPC HDA серверу, может произойти ошибка (рисунок 2.2.4). Обычно это связано с тем, что OPC HDA сервер не настроен на своем компьютере или не настроены опции DCOM.

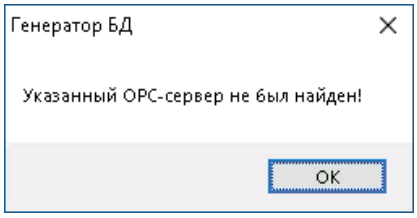


Рисунок 2.2.4 – Окно с сообщением об ошибке соединения с OPC HDA сервером

Рассмотрим подробнее параметры, которые следует настроить для обмена данными с OPC HDA сервером:

- **Пользовательское имя** – имя OPC HDA сервера, задаваемое Пользователем. Длина поля 255 символов. Это поле служит для более понятного Пользователям SCADA КРУГ - 2000 названия OPC HDA сервера (не всегда удобно оперировать его программным именем). Именно оно используется OPC HDA клиентом при формировании сообщений в роллинг (протокол событий). Если Пользовательское имя не задано, то в сообщениях роллинга будет использоваться программное имя
- **Пауза перед опросом** – при разрешенном опросе группы тегов и выполнении одного из условий (старт OPC-HDA-клиента/включение канала связи/смене статуса сервера БД/ восстановление связи по каналу), а также, если канал связи включен и получено разрешение опроса группы тегов, то перед опросом выполняется пауза. Значение данного параметра задается в свойствах канала и может принимать одно из следующих значений:

  «**-1**» – однократный опрос тегов не выполняется. В этом случае дальнейший опрос тегов выполняется согласно заданному режиму опроса группы тегов: для синхронного режима опроса канала – инициативно или с наступлением очередного периода опроса; для асинхронного режима опроса канала – при наличии изменения

  «**0**» – паузы перед опросом тегов нет, происходит однократный запрос данных (для синхронного и асинхронного режимов опроса канала) и далее опрос тегов выполняется согласно заданным режимам опроса группы тегов

  «**>0**» – после восстановления связи и выдержки паузы по каналу (при старте системы/включении канала/смене статуса/после обрыва связи) происходит однократный запрос данных (для синхронного и асинхронного режимов опроса

канала) и далее опрос тегов выполняется согласно заданным режимам опроса группы тегов. В данной версии не используется

- **Глубина опроса** – максимальный период возврата клиента в историю данных HDA сервера, начиная от текущего момента. Значение по умолчанию – **1 час**
- **Сохранять точки с одинаковыми значениями** – значение по умолчанию "да". Т.е. клиент по умолчанию будет сохранять в самописец точки с одинаковыми значениями и качеством. Если выбрано "нет", то будут сохраняться только те точки, у которых изменилось значение или качество.
- **Период опроса (мс) –** период опроса OPC-сервера при синхронном режиме работы (по умолчанию 100 мс).
- **Период диагностики** – интервал времени (в миллисекундах), с которым OPC HDA клиент будет проверять состояние связи с OPC HDA сервером. По умолчанию – **1 секунда**
- **Период повторного соединения** – период времени (в миллисекундах) повторных попыток для соединения с OPC HDA сервером при обрыве связи. Период повторного соединения играет роль, когда с OPC HDA сервером по какой-то причине не удается установить или поддерживать связь. В этом случае по прошествии указанного промежутка времени будет осуществлена попытка восстановить связь. Если она окончилась неудачно, то через этот же промежуток времени будет осуществлена повторная попытка и т.д. до тех пор, пока число попыток не станет равным *количеству повторных соединений*. О результатах попыток установления связи с серверами сообщается в роллинг. Значение по умолчанию – **5 секунд**
- **Количество повторных соединений** – количество попыток повторного соединения, по достижению которого попытки возобновить связь прекратятся (значение по умолчанию **5).**
- **Период повторной проверки** – интервал времени (в миллисекундах), по истечению которого, OPC HDA клиент снова будет пытаться соединиться с OPC HDA сервером (с периодом, указанным в поле «Период повторного соединения», и количеством попыток, указанным в поле «Количество повторных соединений»). По умолчанию – **1 минута**.

2.2.4 Редактирование и привязка тегов OPC HDA сервера

2.2.4.1 Редактирование тегов

Генератор базы данных предоставляет возможность редактирования тегов OPC HDA сервера.

Для редактирования тегов следует на форме канала OPCHDA-клиент перейти на вкладку «Список тегов» (рисунок 2.2.5).

Вкладка разделена на две части: область дерева тегов и область описания тегов выбранной ветки дерева.

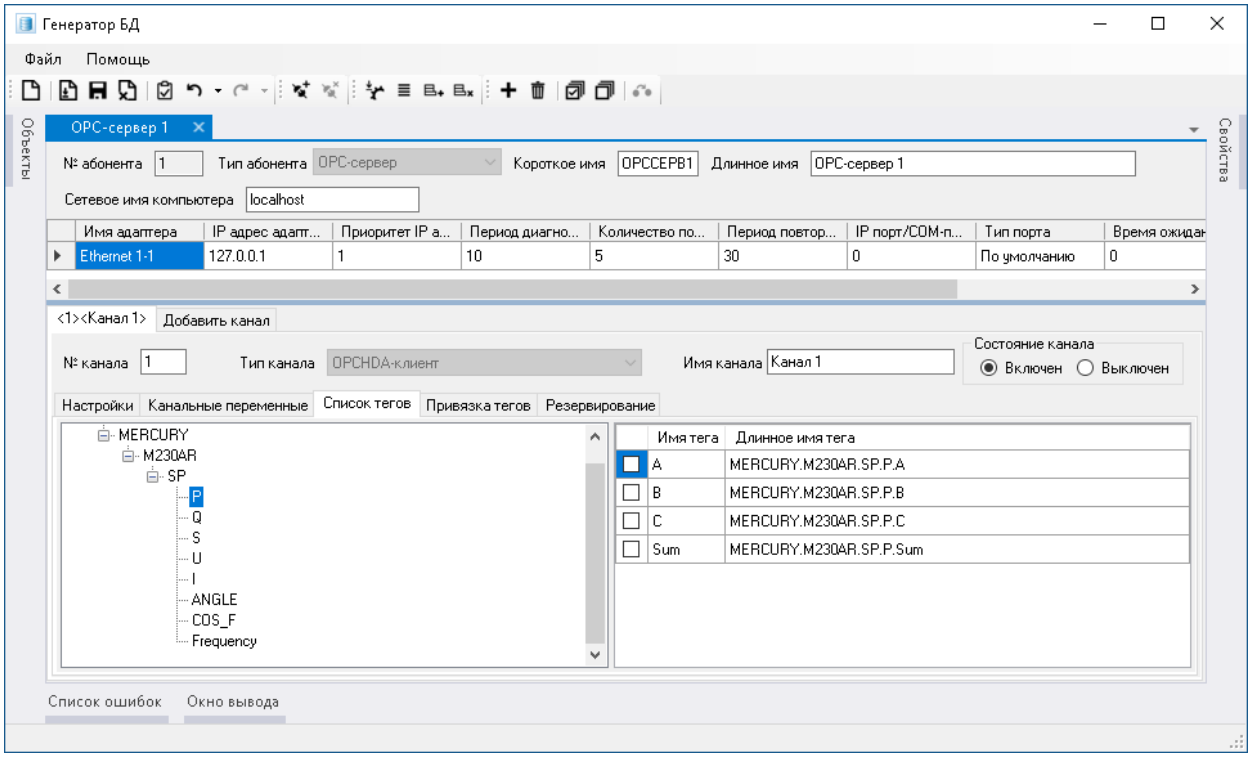


Рисунок 2.2.5 - Форма управления списком тегов OPC HDA сервера

Для создания привязки необходимо установить флажки напротив тегов. Дополнительные действия можно осуществить с помощью контекстного меню, которое вызывается с помощью нажатия правой кнопки мыши в соответствующей области.
Контекстное меню области дерева тегов:

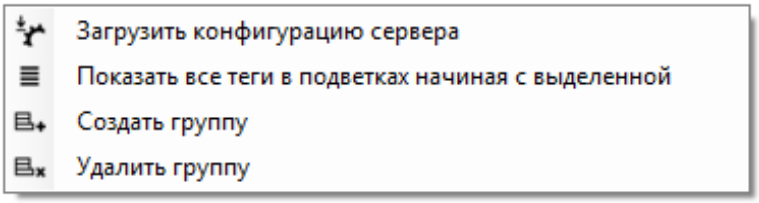


Рисунок 2.2.6 – Контекстное меню области дерева тегов

- **Загрузить конфигурацию сервера** – позволяет обновить дерево тегов с OPC-сервера.
- **Показать все теги в подветках, начиная с выделенной** – в описании тегов отобразятся все теги выделенной ветки и всех ее подветок.
- **Создать группу –** позволяет создать пользовательскую ветку тегов в текущей выбранной ветке.
- **Удалить группу** – удаляет текущую выбранную ветку.

ВНИМАНИЕ!!!

Удаление тегов или группы тегов (и другие изменения) не влияют на информацию на OPC-сервере. При следующем открытии диалога все дерево будет зачитано заново с той структурой, что задана на сервере.

Контекстное меню области тегов:

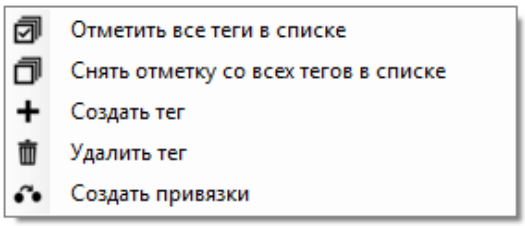


Рисунок 2.2.7 – Контекстное меню области тегов

- **Отметить все теги в списке** – устанавливает флажок выбора у всех тегов текущей выбранной ветки дерева.
- **Снять отметку со всех тегов в списке** – снимает флажок выбора у всех тегов текущей выбранной ветки дерева.
- **Создать тег** – позволяет создать пользовательский тег.
- **Удалить тег** – позволяет удалить выделенный тег.
- **Создать привязки** – создает пустые привязки для тегов, для которых установлен флажок выбора. Дальнейшая привязка осуществляется на вкладке «Привязка тегов».

ВНИМАНИЕ !!!

Каждая пользовательская группа или пользовательский тег (добавленные вручную) после закрытия вкладки канала ОРС не сохраняется. Т.о. при каждом новом открытии формы редактирования тегов после автоматического считывания доступны будут только теги сервера.
Привязки, созданные на основе пользовательских тегов, сохраняются, и их можно просмотреть на форме «Привязка тегов»

2.2.4.2 Привязка тегов

Для того чтобы OPC HDA-клиент знал какие теги OPC-HDA сервера следует обрабатывать и как сопоставить их с перьями самописцев, необходимо произвести привязку тегов OPC HDA-сервера к перьям базы данных.
Для задания привязки между тегами OPC-сервера и перьями базы данных используют вкладку «**Привязка тегов»** (рисунок 2.2.8).

Работа с привязками осуществляется в таблице привязок, поля которой структурированы по разделам:
Описание тега:

- **№** – уникальный номер для идентификации привязки, задается автоматически**.**
- **Имя** – имя опрашиваемого тега. Добавлять свое имя тега нельзя. Список определяется выбором записей на форме со списком тегов OPC HDA-сервера (рисунок 2.2.5)

Описание пера:

- **Самописец** – номер и имя самописца в БД. Выбирается из раскрывающегося списка

ВНИМАНИЕ !!!

В списке самописцев присутствуют только самописцы для выбранного канала.

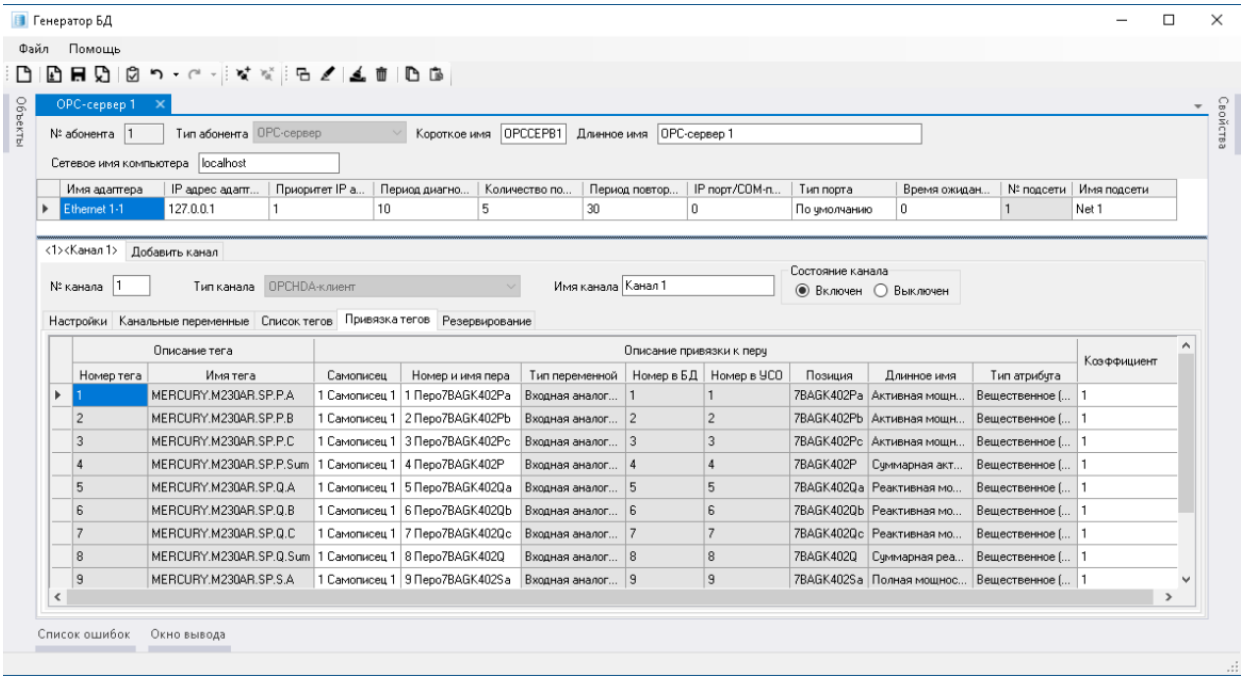


Рисунок 2.2.8 – Форма привязок тегов OPC HDA сервера к перьям самописцев БД

- **№ и имя пера** – номер и имя пера самописца, в которое будет производиться запись данных тега OPC HDA-сервера. Выбирается из раскрывающегося списка.
- **Тип переменной, № в БД, № в УСО, Позиция и Длинное имя переменной –** атрибуты переменной, соответствующей выбранному перу. Нередактируемые информационные поля.
- **Тип атрибута –** тип данных атрибута переменной, являющейся источником пера (информационное поле), заполняется при выборе пера
- **Коэффициент –** величина, на которую умножаются значения тега OPC HDA сервера перед записью в выбранное перо. При этом время и качество тега OPC HDA сервера остаются неизменными.

ВНИМАНИЕ !!!

Источником пера являются переменные, привязанные к текущему каналу OPC HDA, выбранному в самописце. В случае необходимости визуализировать дополнительные параметры пера, например такие как «единицы измерения», «позиция», будут использоваться атрибуты переменной, связанной с данным пером.

Панель «**Автозаполнение**» (рисунок 2.2.9) служит для автоматического заполнения привязок тегов OPC HDA сервера к существующим перьям выбранного самописца БД и вызывается с помощью соответствующей кнопки на панели инструментов . Заполнение таблицы привязок происходит с помощью выбора перьев из списка и переноса записей в таблицу привязок путём «захвата» (нажатием и удержанием левой кнопки мыши) выделенных в списке записей перьев выбранного самописца, и перемещением их поверх текущих привязок.

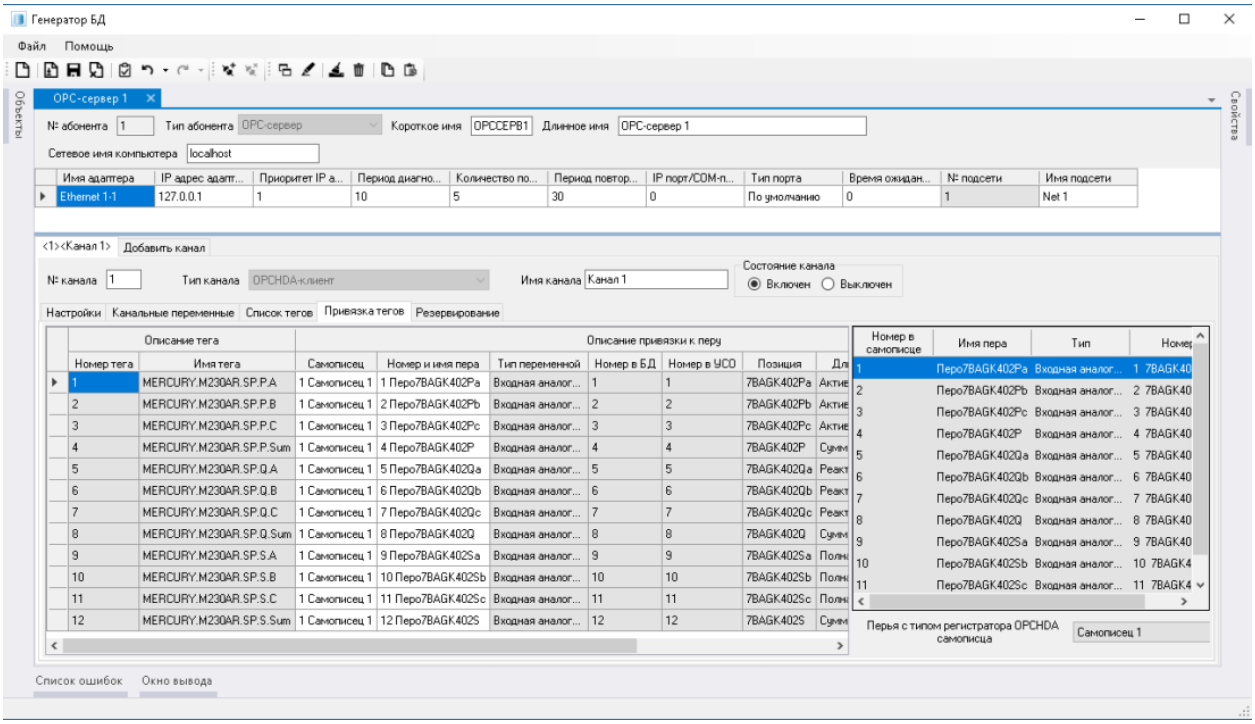


Рисунок 2.2.9 - Форма автозаполнения привязок

2.2.5 Получение значений тегов по команде Пользователя

OPC HDA клиент предоставляет возможность получения значений тегов с OPC HDA сервера по команде Пользователя.

Для получения значений используется атрибут №**42** «**ResBt**» таблицы «**Канал**» базы данных.

Алгоритм использования этого атрибута OPC HDA клиентом следующий:

- По умолчанию, значение атрибута **№42** равно **0**
- OPC HDA клиент периодически (**1 раз в секунду**) проверяет значение этого атрибута
- Если значение атрибута №**42** равно **1**, то OPC HDA клиент осуществляет получение значений тегов с OPC HDA сервера. После выполнения операции значение атрибута автоматически сбрасывается в **0**.

Таким образом, для управления получением тегов по команде Пользователя следует:

- Создать в Генераторе динамики на мнемосхеме графический элемент, используемый для управления обновлением тегов, например, кнопку
- Назначить для этого элемента функцию реакции «**Установить значение**» на событие «**Нажатие левой клавиши мыши**»
- В окне свойств элемента в поле «**Приемник**» назначить ссылку на переменную «**System\Канал\ResBt\n**», где **n** – номер канала
- В поле «**Записываемое значение**» задать **1** (рисунок 2.2.10).

При работе автоматизированной системы, нажав на созданную таким образом кнопку, Пользователь изменит значение атрибута №**42** на **1** и этим инициирует получение значений тегов с OPC HDA сервера.

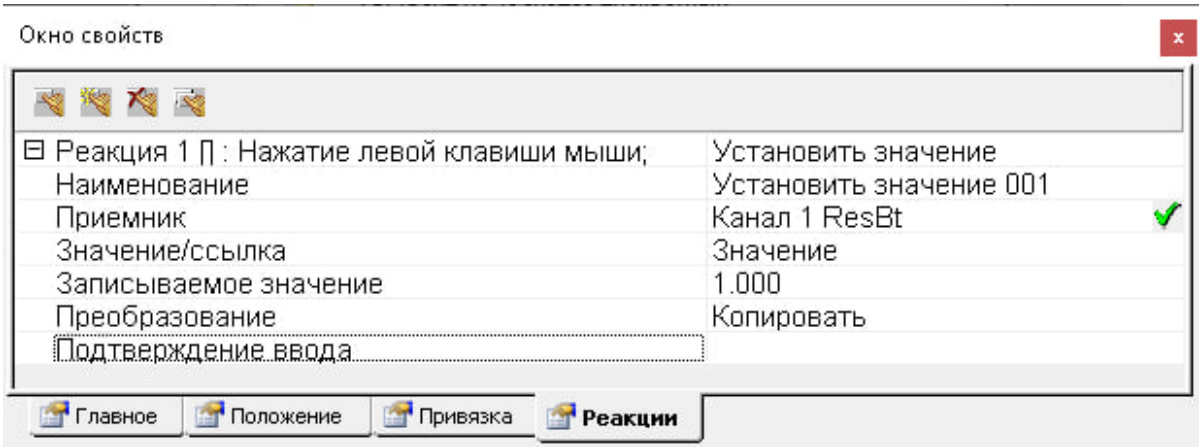


Рисунок 2.2.10 – Окно свойств Генератора динамики. Назначение функции реакции

2.2.6 Параллельный опрос нескольких OPC HDA серверов

Начиная с версии SCADA 4.0 СПО1 хотфикс 1 возможен параллельный опрос нескольких OPC HDA серверов. Эта функция может быть полезна, например, в случае большого времени получения значений тегов сервера.

Для реализации параллельного опроса в строке запуска OPC HDA клиента добавлен параметр **ch=<номер канала>**, который определяет номера каналов связи с OPC HDA серверами, с которыми будет работать OPC HDA клиент.

Например,

KrOPCHDAClient.exe -ch=12
OPC HDA клиент будет работать с каналом №12.

KrOPCHDAClient.exe -ch=12, 18, 19
OPC HDA клиент будет работать с каналами №№ 12, 18 и 19.

Для того чтобы настроить параллельный опрос необходимо выполнить следующие действия:

1. В **Менеджере Задач** добавить нужное количество процессов **KrOPCHDAClient.exe**. Для дальнейшей настройки эти процессы должны иметь разные имена
2. Добавить эти процессы в приложение
3. Задать в параметрах запуска процессов с помощью параметра **ch** необходимые номера каналов.

Подробно работа с процессами, приложениями и проектами описана в книге 8 «Среда исполнения», часть 2 «Программные модули и комплексы», пункт 3 «Менеджер задач. Конфигурирование приложений».

2.2.7 Настройка резервирования

В случае резервирования OPC HDA серверов возникает задача «переключения» OPC HDA клиента к одному из них в зависимости от наличия связи между OPC-клиентом и OPC HDA сервером.
Настройка переключения OPC HDA клиента к одному из резервируемых OPC HDA серверов осуществляется с помощью Генератора базы данных следующим образом:

- В дереве объектов в пункте «Абоненты» выбрать ветку с именем абонента типа OPC-сервер, в котором описан OPCHDA-канал или создать такого абонента

- В появившейся форме выбрать вкладку с описанием нужного канала типа OPCHDA-клиент или при добавлении канала выбрать тип канала «**OPC HDA клиент**», задать необходимые параметры описания канала и обмена с OPC HDA сервером
- Перейти на вкладку «**Резервирование**» (рисунок 2.2.11) и задать требуемые параметры переключения OPC HDA клиента:
  - **«Абонент с резервным OPC HDA сервером»** – абонент автоматизированной системы, на котором установлен «резервный» OPC HDA сервер. Для правильной и корректной работы канала «OPC HDA клиент» OPC HDA сервер на данном абоненте должен иметь такое же программное имя и идентичный набор тегов, что и OPC HDA сервер, определенный как «основной» (заданный в определении канала «OPC HDA клиент»).
    Имя абонента выбирается из списка всех «длинных имен абонентов», определенных в базе данных системы. Исключением является абонент, который определен в полях «Номер и имя абонента» при определении канала «OPC HDA клиент».

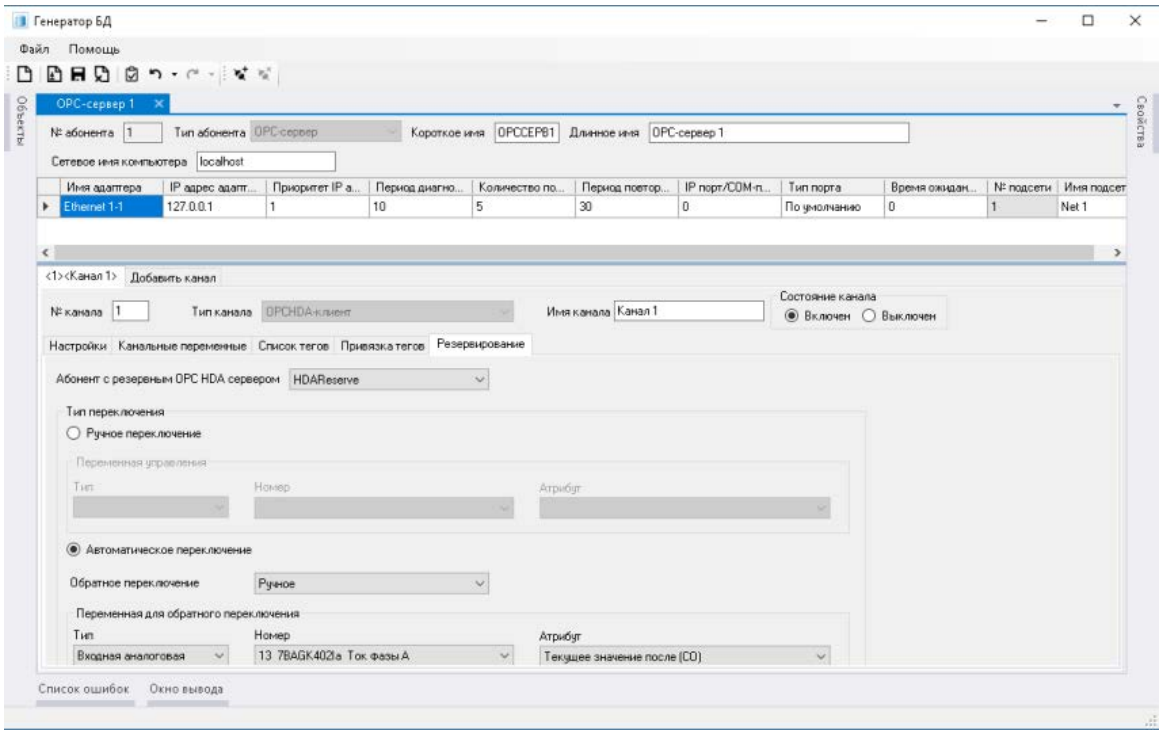


Рисунок 2.2.11 – Форма настройки резервирования

  - **«Тип переключения»** – группа для задания параметров переключения OPC HDA клиента. Группа определяет один из двух взаимоисключающих режимов «**Ручное переключение**» или «**Автоматическое переключение**» и соответствующие им параметры.
    Если параметры заданы, но недоступны для редактирования, Генератор базы данных отображает их значения на форме для информирования пользователя
  - «**Переменная управления**» – определяет переменную базы данных, с помощью которой в реальном времени будет осуществляться выбор абонента с OPC HDA сервером, к которому будет подключаться OPC HDA клиент. Переменная управления определяется с помощью следующих атрибутов:

    Тип – тип переменной базы данных. Выбирается из списка типов переменных, определенных в базе данных системы

    Номер – номер в БД и позиция переменной выбранного типа

Атрибут – атрибут выбранной переменной (тип атрибута только числовой и логический)

- «**Обратное переключение**» – определяет режим переключения для возврата опроса на абонента, заданного по умолчанию как «основной» для данного канала «OPC HDA клиент». Данное поле доступно только при выборе типа «Автоматическое переключение»
- «**Переменная для обратного переключения**» – определяет переменную базы данных, с помощью которой в реальном времени будет осуществляться возврат на абонента с OPC HDA сервером, определенного, как «основной». Атрибуты переменной аналогичны атрибутам переменной управления. В качестве переменной для обратного переключения можно выбрать любую переменную базы данных.

Режим работы «Резервирование»

Для определения алгоритма переключения OPC HDA клиента следует:

- Выбрать имя абонента в списке значений поля «**Абонент с резервным OPC HDA сервером**» (рисунок 2.2.11)

  ☝ВНИМАНИЕ!!!

  Задать параметры переключения OPC HDA клиентов можно только в том случае, если выбран «Абонент с резервным OPC HDA сервером»

- Выбрать тип переключения:
  - «**Ручное переключение**» – переключение на «резервный» OPC HDA сервер и обратно происходит в зависимости от значения переменной, заданной в поле «Переменная управления». Таким образом, пользователь, изменяя значение этой переменной, имеет возможность управлять выбором абонента, с которого OPC HDA клиент получает значения тегов OPC HDA сервера
  - «**Автоматическое переключение**» – переключение на «резервный» OPC HDA сервер происходит автоматически при обрыве связи с «основным» OPC HDA сервером (определенном в свойствах канала «OPCHDA-клиент»)
- **Если выбран тип «Автоматическое переключение»,** то задать параметры «обратного переключения»:
  - Тип обратного переключения «**ручное**» и «**Переменная для обратного переключения**» – возврат на «основной» OPC HDA сервер произойдет только по команде пользователя (с помощью изменения значения переменной для обратного переключения)
  - Тип обратного переключения «**автоматическое**» – обратное переключение на «основной» OPC HDA сервер происходит автоматически при восстановлении связи с OPC HDA сервером на данном абоненте.

☝ВНИМАНИЕ!!!

Если устройство некорректно работает при подключении к нему нескольких OPC HDA серверов, то использовать «автоматический» режим переключения не рекомендуется

Режим работы «Нет резервирования»

Для отключения «механизма переключения» OPC HDA клиентов необходимо в поле «Абонент с резервным OPC HDA сервером» выбрать пустую строку.

В этом случае группа «**Тип переключения**» недоступна для редактирования/изменения и переключение не происходит.

3 OPC UNIFIED ARCHITECTURE

3.1 OPC UA СЕРВЕР

3.1.1 Общие сведения

OPC UA (OPC Unified Architecture) сервер – это специализированный OPC-сервер, предоставляющий оперативные и исторические данные технологического процесса.
Для работы с OPC UA серверами используются специализированные OPC UA клиенты, ориентированные на работу с оперативными и историческими данными.

ВНИМАНИЕ!!!

- **Параметры OPC UA сервера SCADA КРУГ-2000 загружаются автоматически из файла настроек OPCUAServer.ini в корневом каталоге SCADA КРУГ-2000 и могут настраиваться Пользователем.**
- **OPC UA сервер может работать с любыми переменными НЕнулевых каналов без дополнительной настройки базы данных!**
- **OPC UA сервер обеспечивает только чтение значений оперативных тегов и трендов, формирование и изменение значений оперативных тегов и трендов сервер не поддерживает.**
- **OPC UA сервер предоставляет доступ к значениям оперативных тегов и трендов при наличии работающего сервера базы данных КРУГ-2000 (Сервер БД/АБД).**
- **OPC UA сервер будет запущен только в случае разрешения этого в Вашем аппаратном ключе защиты!**
- **Для организации доступа к архивным значениям трендов требуется наличие Сервера АБД, запуск которого также должен быть разрешён в Вашем аппаратном ключе защиты.**

OPC UA сервер системы КРУГ 2000 поддерживает обмен данными согласно спецификации OPC UA версии 1.2 (**OPC Unified Architecture Specification. version 1.2**).

3.1.2 Состав OPC UA сервера

OPC UA сервер системы КРУГ-2000 включает следующие файлы:

- **OPCUAServer.exe** – исполняемый файл
- **OPCUAServer.exe.log** – лог-файл для вывода ошибок
- **OPCUAServer.ini**– файл настроек

3.1.3 Работа с OPC UA сервером

Запуск OPC UA сервера производится с помощью Менеджера задач SCADA КРУГ-2000. Если в момент запуска OPC UA сервера не будет обнаружен работающий Сервер базы данных, то OPC UA сервер запущен не будет. В лог-файле при этом будет выведено сообщение об ошибке (рисунок 3.1.1).

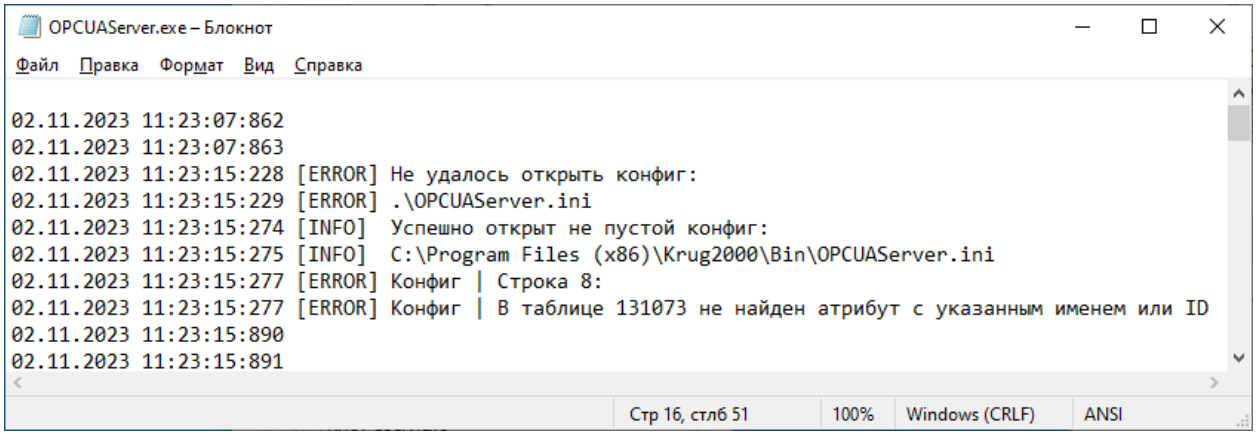


Рисунок 3.1.1 - Сообщения OPC UA сервера

ВНИМАНИЕ!!!

Всегда запускайте Сервер базы данных перед началом работы с OPC UA сервером! OPC UA сервер поддерживает получение данных от Сервера БД, запущенного в режиме «Демо» (о режимах работы сервера БД можно прочитать в книге 10 «Среда исполнения», часть 2 «Программные модули и комплексы», раздел 1.1.2 «Параметры запуска Сервера БД»).

Параметры OPC UA сервера могут настраиваться Пользователем в файле настроек OPCUAServer.ini в корневом каталоге SCADA КРУГ-2000.

По умолчанию ini-файл содержит записи для текущих значений каждого из типов переменных БД.

Без дополнительных настроек будет формироваться дерево тегов для текущих значений всех типов переменных.

```
# Входная аналоговая
Type1.1=Item Value,65,OPC_READABLE;

# Входная дискретная
Type2.1=Item Value,45,OPC_READABLE;

# Аналоговая выходная
Type3.1=Item Value,107,OPC_READABLE;

# Дискретная выходная
Type4.1=Item Value,46,OPC_READABLE;

# PB Составная
Type5.1=Item Value,11,OPC_READABLE;
```

Рисунок 3.1.2 -Описание для текущих значений всех типов переменных

Структура сформированного дерева тегов для OPC-клиента имеет следующий синтаксис:

Operative.<ТИП>.K<НОМЕР1>.<ПОЗИЦИЯ>_<НОМЕР2>.<ИМЯ АТРИБУТА>

ТИП – краткое наименование типа переменной:

VA – входная аналоговая; **VD** – входная дискретная;

AV – аналоговая выходная; **DV** – дискретная выходная; **RV** – ручной ввод

НОМЕР1 – номер канала переменной

ПОЗИЦИЯ – позиция переменной, например, TIC01

НОМЕР2 – номер переменной в базе данных

ИМЯ АТРИБУТА – имя атрибута переменной, передаваемого в качестве тега

Например,

Operative.VA.K28.АСУЭ2_7.Item Value
Operative.DV.K2.TIC01_25.Item Value

Для описания доступа к значениям трендов используйте следующий синтаксис:

Historical.<ТИП>.K<НОМЕР1>.<ПОЗИЦИЯ>_<НОМЕР2>.R<НОМЕР3>P<НОМЕР4>

ТИП – краткое наименование типа переменной:

VA – входная аналоговая; **VD** – входная дискретная;

AV – аналоговая выходная; **DV** – дискретная выходная; **RV** – ручной ввод

НОМЕР1 – номер канала переменной

ПОЗИЦИЯ – позиция переменной, например, TIC01

НОМЕР2 – номер переменной в базе данных

НОМЕР3 – номер самописца в базе данных

НОМЕР4 – номер пера в самописце

Например,

Historical.VA.K28.АСУЭ2_7.R5P2
Historical.DV.K2.TIC01_25.R9P12

3.1.4 Файл настроек OPC UA-сервера

Настройка OPC UA-сервера осуществляется с помощью файла ***OPCUAServer.ini***, который расположен в корневом каталоге SCADA КРУГ-2000.
В ini-файле можно задать интервал опроса (обновления) данных от источников данных:

```
# Интервал опроса (мс)
SamplingInterval=1000
```

Ini-файл содержит описания типов и атрибутов. Описание типов начинается со служебного слова [Types]:

```
[Types]

Type1=Variable\Входная аналоговая;VA;Позиция+RecordID;
```

```
Type2=Variable\Входная дискретная;VD;Позиция+RecordID;
Type3=Variable\Аналоговая выходная;AV;Позиция+RecordID;
Type4=Variable\Дискретная выходная;DV;Позиция+RecordID;
Type5=Variable\PB Составная;RV;Позиция+RecordID;
```

Можно добавить описание своего типа, для этого необходимо использовать следующий шаблон:
Type<1>=<2>\<3>;<4>;<5>;
<1> - порядковый номер типа,
<2> - имя реестра,
<3> - имя таблицы,
<4> - псевдоним таблицы для отображения в дереве тегов,
<5> - псевдоним записи таблицы для отображения ветки в дереве тегов

Описание атрибутов начинается со служебного слова [Attributes]

```
[Attributes]

# Входная аналоговая
Type1.1=Item Value,65,OPC_READABLE;

# Входная дискретная
Type2.1=Item Value,45,OPC_READABLE;

# Аналоговая выходная
Type3.1=Item Value,107,OPC_READABLE;

# Дискретная выходная
Type4.1=Item Value,46,OPC_READABLE;

# PB Составная
Type5.1=Item Value,11,OPC_READABLE;
```

Если необходимо сделать опрос для других атрибутов переменных, то необходимо добавить в ini-файл строку, сформированную по следующему шаблону:
<1>.<2>=<3>,<4>,<5>;
<1> - тип, значение которого определено выше в ini-файле,
<2> - порядковый номер записи/тега указанного типа (например, для ВА Type1.1= … Type1.2= … Type1.3= …),
<3> - пользовательское имя для тега, которое будет выводиться в дереве тегов в OPC-клиенте,
<4> - номер атрибута переменной в базе данных, значение которого будет выводиться в атрибуте тега,
<5> - разрешение на чтение и/или запись (на чтение - OPC_READABLE, на чтение и запись - OPC_READABLE+OPC_WRITEABLE)

Так же в ini-файле приведены примеры заполнения типа и атрибута:

```
#                          Пример типа
#  ------------------------------------------------------------
#  |   Type1=Variable\Входная аналоговая;VA;Позиция+RecordID;  |
#  ------------------------------------------------------------
#  |       Type1          |                Тип                |
#  |      Variable        |            Реестр в БД            |
#  | Входная аналоговая   |            Таблица в БД           |
#  |         VA           |            Префикс тега           |
#  |  Позиция+RecordID    |              Имя тега             |
#  ------------------------------------------------------------

#                        Пример атрибута
#  ------------------------------------------------------------
#  |     Type1.1=Item Value,65,OPC_READABLE+OPC_WRITEABLE;    |
#  ------------------------------------------------------------
#  |           Type1.1          |     Тип, порядковый номер   |
#  |         Item Value         |          Имя атрибута       |
#  |            65              |     Номер атрибута (в БД)   |
#  | OPC_READABLE+OPC_WRITEABLE |    Доступ (чтение, запись)  |
#  ------------------------------------------------------------
```

3.2 OPC UA КЛИЕНТ

3.2.1 Общие сведения

OPC UA клиент предназначен для обмена данными SCADA КРУГ-2000 в формате OPC UA со сторонними OPC-серверами.

Для того чтобы обозначить OPC UA сервер, как источник данных для OPC UA клиента, введен **тип абонента** «**OPC-сервер**».

В рамках системы КРУГ-2000 сторонний OPC UA сервер представляет собой канал данных. Канал может принадлежать одному из абонентов системы КРУГ-2000, и с этой точки зрения абонент системы КРУГ-2000 представляет собой компьютер (локальный или удаленный), где работает один OPC UA сервер.

Настройка OPC UA клиента происходит в несколько этапов:

1 Добавление абонента в систему

2 Создание каналов «OPCUA-клиент» или «OPCUA-клиент(дублирование)»

3 «Привязка» (указание соответствия) тегов или атрибутов тегов каждого OPC UA сервера к атрибутам переменных оперативной БД или к событийным перьям для исторических тегов.

3.2.2 Добавление абонента

Для создания канала OPCUA-клиент необходим абонент типа «**OPC-сервер**» с настройками компьютера, где работает один OPC UA сервер. Если нужного абонента нет, то его необходимо добавить в базу данных. Для этого необходимо в контекстном меню пункта «Абоненты» (Объекты → Система → Абоненты) выбрать соответствующий тип (рисунок 3.1). В открывшейся форме описания абонента (рисунок 3.2) следует указать **Сетевое имя компьютера** в соответствии с сетевыми настройками операционной

системы, установленной на выбранном компьютере (по умолчанию создается абонент с сетевым именем локальной машины – localhost).

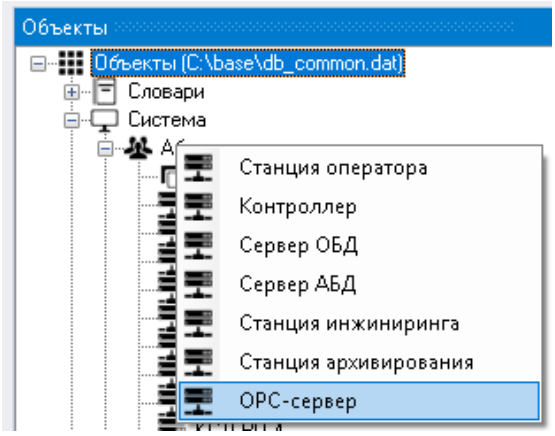


Рисунок 3.1 – Контекстное меню создания абонента

После добавления нового абонента в таблице адаптеров абонента следует указать соответствующий ему адаптер. Ключевым здесь является поля «**IP-адрес адаптера**» и «**IP-порт**». По умолчанию создается адаптер для локального абонента (компьютера) с IP-адресом – **127.0.0.1** и номер порта 0 для подключения к серверу. Для удаленного абонента (компьютера) укажите **IP-адрес компьютера,** заданный в сетевых настройках операционной системы.

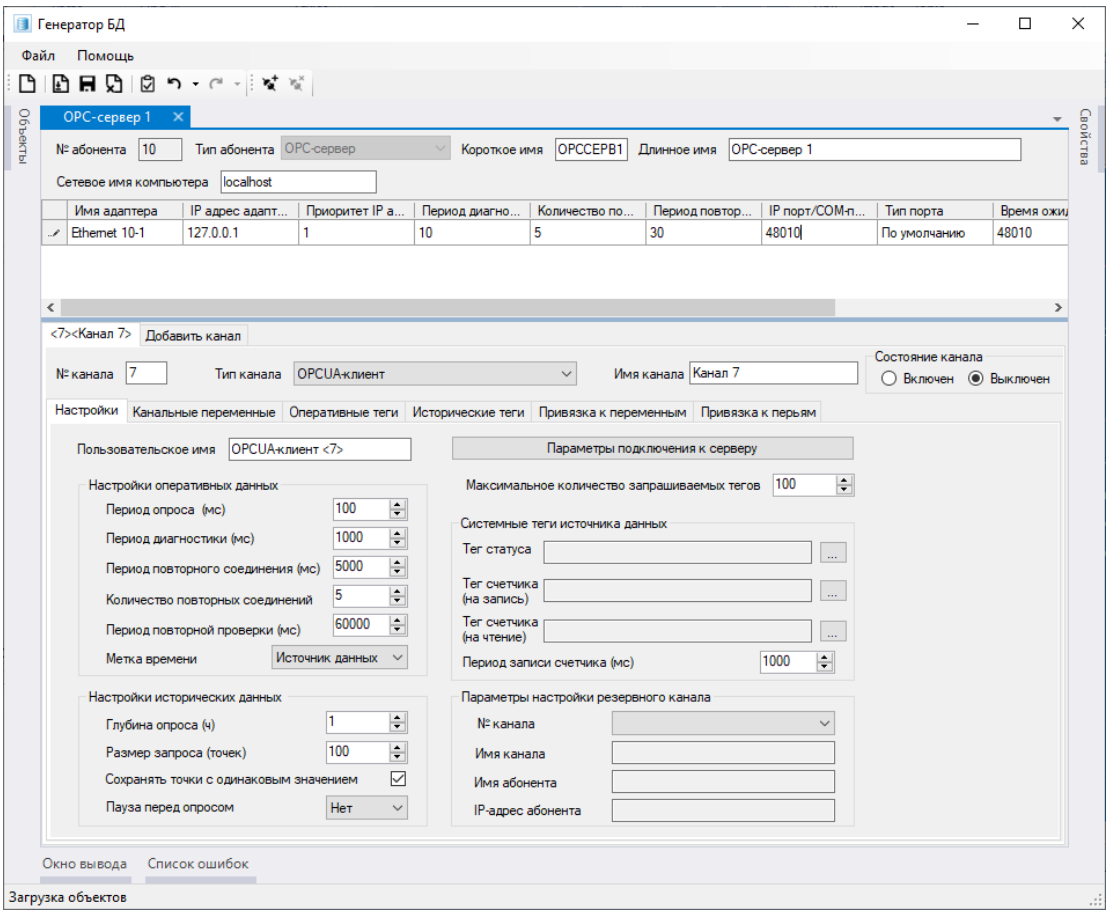


Рисунок 3.2 – Форма настройки описания абонента

Параметры, которые следует настроить для обмена данными с OPC UA сервером:

- **Пользовательское имя** – имя OPC-сервера, задаваемое пользователем (255 символов). Именно оно используется при выдаче OPC-клиентом сообщений в протокол событий. Если пользовательское имя не задано, то в сообщениях протокола событий будет использоваться программное имя OPC-сервера.
- **Максимальное количество запрашиваемых тегов** – максимальный размер массива запрашиваемых тегов в одном запросе чтения данных.

Настройки оперативных данных:

- **Период опроса (мс)** – период опроса OPC-сервера при синхронном режиме работы (по умолчанию 100 мс).
- **Период диагностики (мс)** – интервал времени, с которым OPC-клиент будет проверять состояние OPC-сервера (по умолчанию 1 секунда).
- **Период повторного соединения (мс)** – период времени в миллисекундах, использующийся для повторных попыток соединения с OPC-сервером при обрыве связи (по умолчанию 5 секунд).
- **Количество повторных соединений** – количество попыток повторного соединения, по достижении которого попытки возобновить связь прекратятся (по умолчанию 5).
- **Период повторной проверки (мс)** – интервал времени, по истечении которого OPC-клиент снова будет пытаться соединиться с OPC-сервером указанное количество раз в поле «Количество повторных соединений» с периодом, указанным в поле «Период повторного соединения» (по умолчанию 1 минута).
- **Метка времени** – используемая при обработке переменной метка времени выбирается из выпадающего списка источников времени:
  - ✓ «Источник данных» (по умолчанию) – метка времени значения от источника данных,
  - ✓ «OPCUA-сервер» – метка времени регистрации значения OPCUA-сервером,
  - ✓ «Компьютер» – метка времени Сервера БД при записи значения тега в атрибут переменной

Настройки исторических данных:

- **Глубина опроса (ч)** – максимальный период возврата клиента в историю данных OPCUA-сервера, начиная от текущего момента.
- **Размер запроса (точек)** – максимальное количество значений, возвращаемых в текущем опросе одного тега исторических данных (по умолчанию 100)
- **Сохранять точки с одинаковыми значениями** – по умолчанию "да". Т.е. клиент по умолчанию будет сохранять в самописец точки с одинаковыми значениями и качеством. Если выбрано "нет", то будут сохраняться только те точки, у которых изменилось значение или качество.
- **Пауза перед опросом** – при выполнении одного из условий (старт OPCUA-клиента/включение канала/смена статуса сервера БД/восстановление связи по каналу), выполняется пауза перед опросом, в соответствии с выбранным из выпадающего списка значением:
  - ✓ «Нет» – паузы перед опросом тегов нет, происходит однократный запрос всех данных и далее в соответствии с режимом опроса данных по группам,
  - ✓ «Да» – однократный запрос тегов не выполняется. В дальнейшем для каждой группы тегов опрос выполняется в соответствии с режимом и периодом опроса данных по группам

Системные теги источника данных:
Для изменения значения источника данных необходимо нажать кнопку рядом с соответствующим полем и в списке выбрать тег или пустую строку (если нужно обнулить значение).

- **Тег статуса -** текстовое поле содержит номер записи в таблице привязок к переменным и полное имя тега из привязки. Текущее значение тега статуса источника данных (1-Основной / 0-Резервный) используется Сервером БД для определения текущего статуса канала связи в соответствии с алгоритмом резервирования каналов.
- **Тег счетчика (на запись) -** текстовое поле содержит номер записи в таблице привязок к переменным и полное имя тега из привязки. В выбранный тег UA-клиент отправляет значение ***Тега счетчика (на чтение)*** с инкрементом на 1 с ***Периодом записи счетчика,*** указанном в соответствующем поле. Если ***Тег счетчика (на чтение)*** не задан, то используется «внутренне» значение UA-клиента.
- **Тег счетчика (на чтение) -** текстовое поле содержит номер записи в таблице привязок к переменным и полное имя тега из привязки. Указывается тег, значение которого принимается с UA-сервера для последующей записи в ***Тег счетчика (на запись)***.
- **Период записи счетчика (мс)** - период записи значения ***Тега счетчика (на запись)*** в UA сервер.

Параметры настройки резервного канала:
- **№ канала** – выбирается из списка, который формируется из каналов этого же типа. В качестве резервного канала может быть выбран канал, который еще не используется в резервировании каналов. Если текущий канал указывался в качестве резервного для других каналов, данное поле недоступно для редактирования.
- **Пользовательское имя резервного канала, имя абонента резервного канала и IP-адрес абонента резервного канала –** информационные поля и недоступны для редактирования.

3.2.3 Создание каналов «OPC UA-клиент»

Чтобы добавить новый канал на форме описания абонента (рисунок 3.2) необходимо левой кнопкой мыши выбрать вкладку «Добавить канал», в появившемся контекстном меню (рисунок 3.3) необходимо выбрать тип создаваемого канала – OPC UA-клиент или OPC UA-клиент(дублирование).

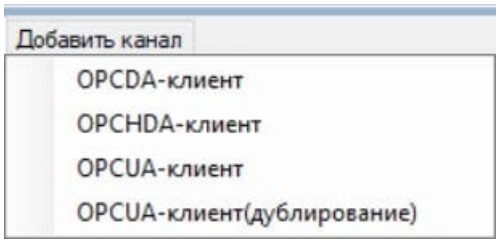


Рисунок 3.3 – Контекстное меню создания канала

На вкладке «Настройки» необходимо определить **параметры подключения к OPC-серверу** (выбрать конфигурацию OPC UA сервера).

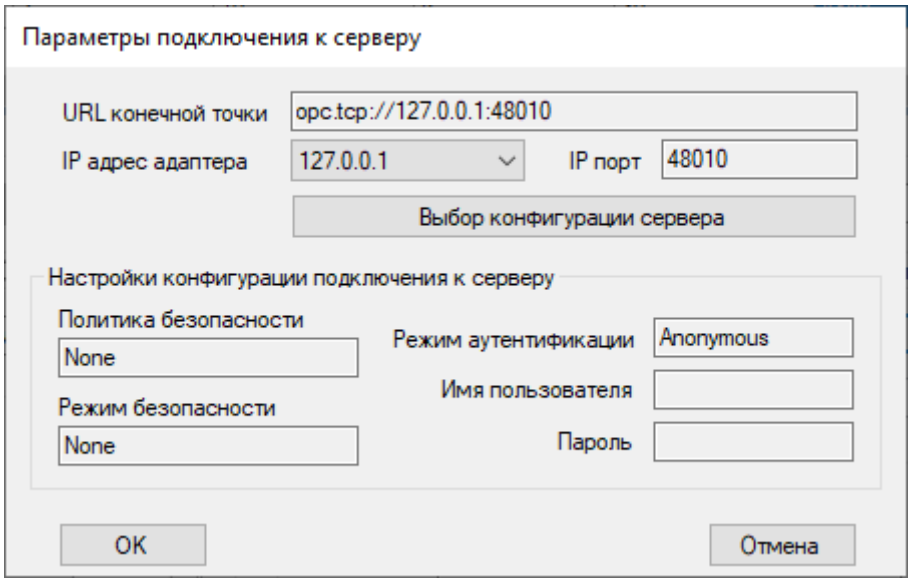


Рисунок 3.4 – Параметры подключения к OPC UA серверу

- **URL конечной точки** – имя сервера, с которым будет выполняться соединение (в том числе при выполнении теста соединения). Формируется из настроек адаптера абонента: IP адрес и IP порт (рисунок 3.5).

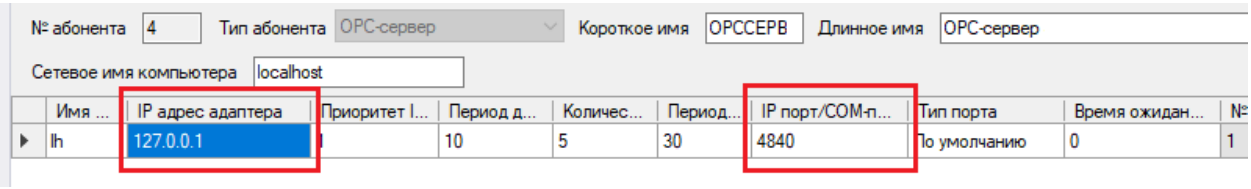


Рисунок 3.5 – Настройки адаптера абонента

- **IP адрес адаптера** - выбор из выпадающего списка адаптеров с приоритетом 1 и 2 для абонента канала, для которого будет сформирована строка подключения к серверу.
- **IP порт** – информационное поле со значением IP порта для выбранного адаптера.
- кнопка **Выбор конфигурации сервера** - при нажатии на кнопку проверяется соединение с абонентом канала с заданными параметрами подключения. При удачном соединении выводится форма **Конфигурация сервера**, в которой выбирается требуемая конфигурация из списка доступных на данном абоненте. После аутентификации пользователя происходить подключение к UA серверу.

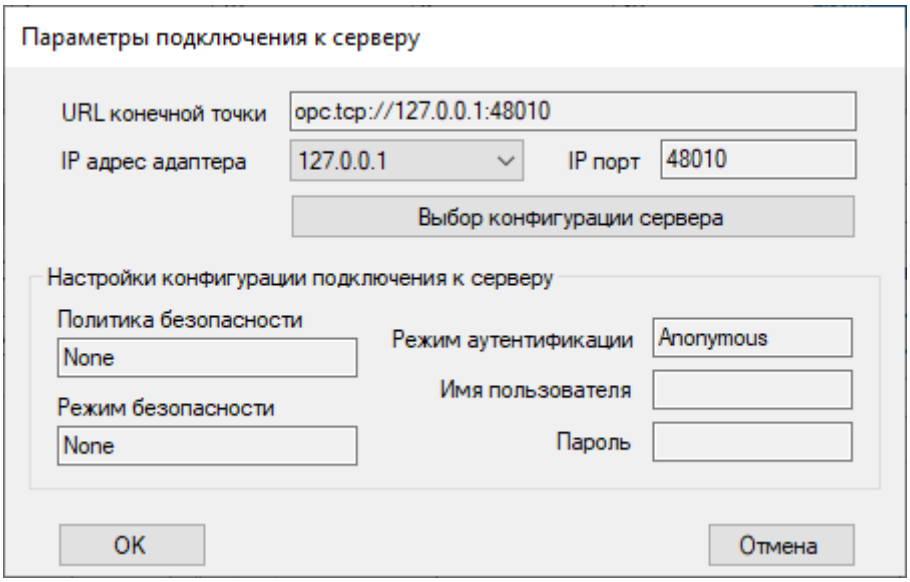
При успешной установке соединения заполняются информационные поля: политика безопасности, режим безопасности, режим аутентификации, имя пользователя и пароль. При нажатии на кнопку **ОК** настройки подключения будут сохранены в базе. При нажатии кнопки **Отмена** - изменение настроек не будет сохранено в базе.

Выбор конфигурации и сертификаты. Порядок действий.

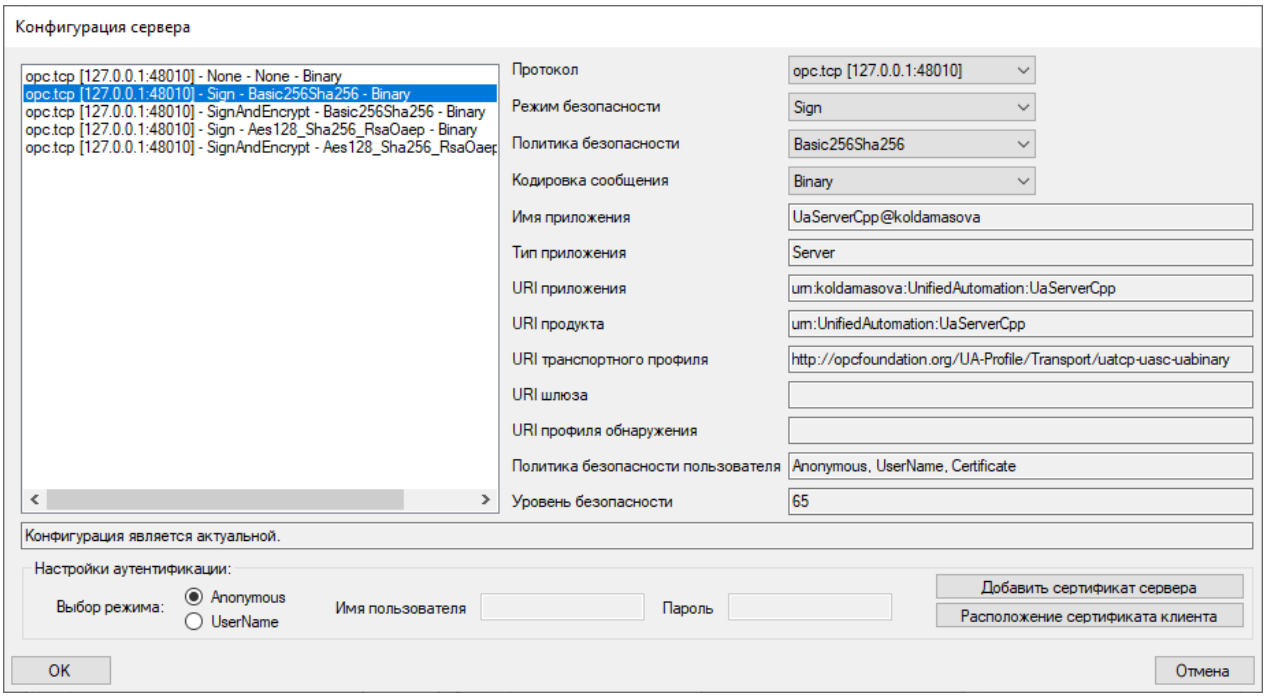
Предварительно должен быть создан абонент OPC-сервер и заполнены настройки адаптера: IP адрес и IP порт. Так же должен быть создан канал UA-клиент или UA-клиент(дублирование). Далее, для настройки подключения к UA серверу необходимо:

1) В настройках канала перейти в «Параметры подключения к серверу» (нажать соответствующую кнопку)

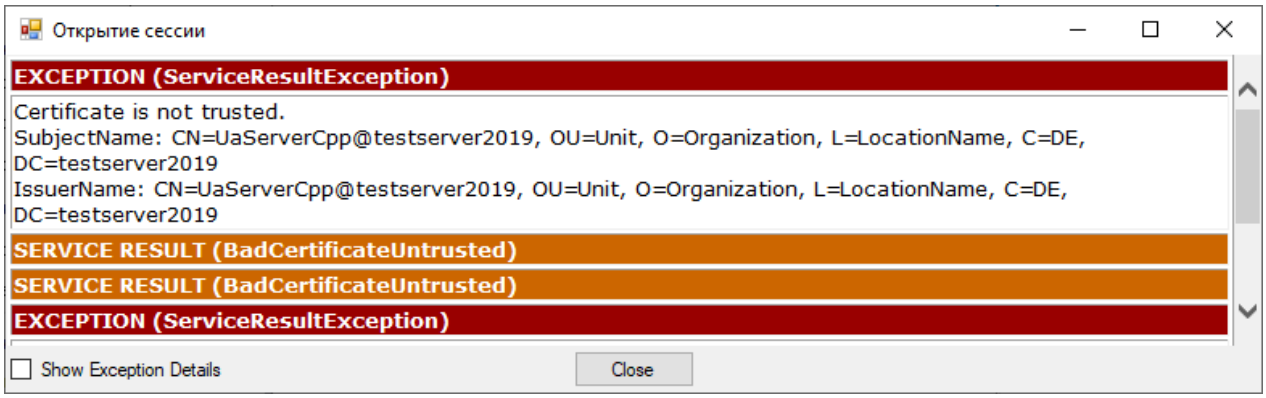
2) Перейти к выбору конфигурации сервера (нажать кнопку «Выбор конфигурации сервера»)



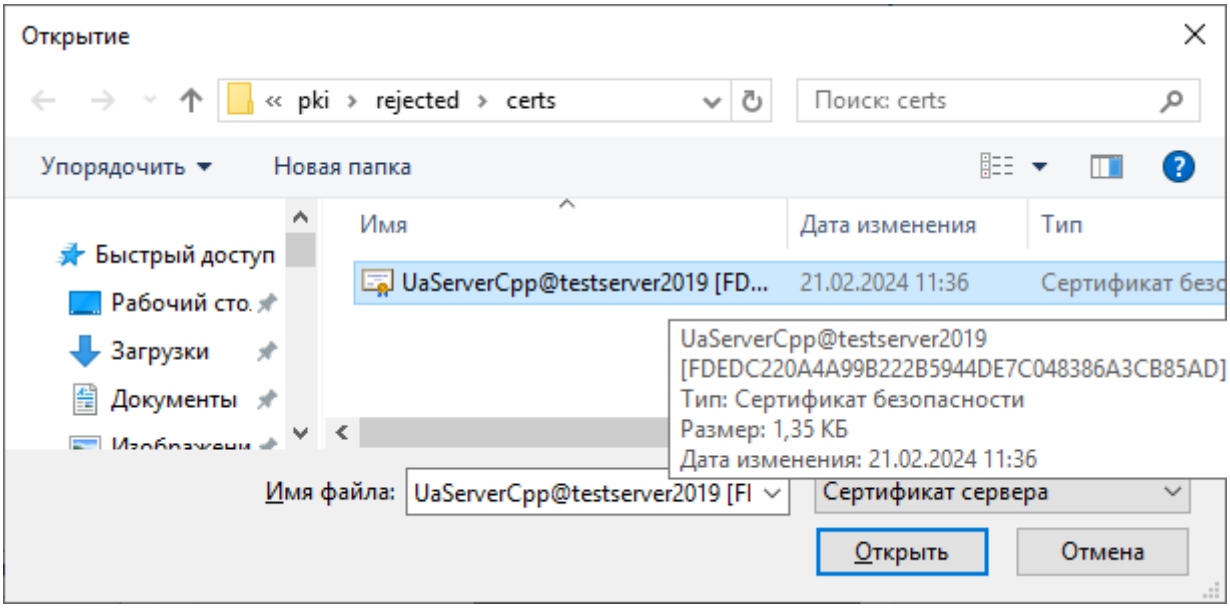
3) Выбрать конфигурацию. Выбрать режим аутентификации (при необходимости ввести логин и пароль). Нажать кнопку «ОК».



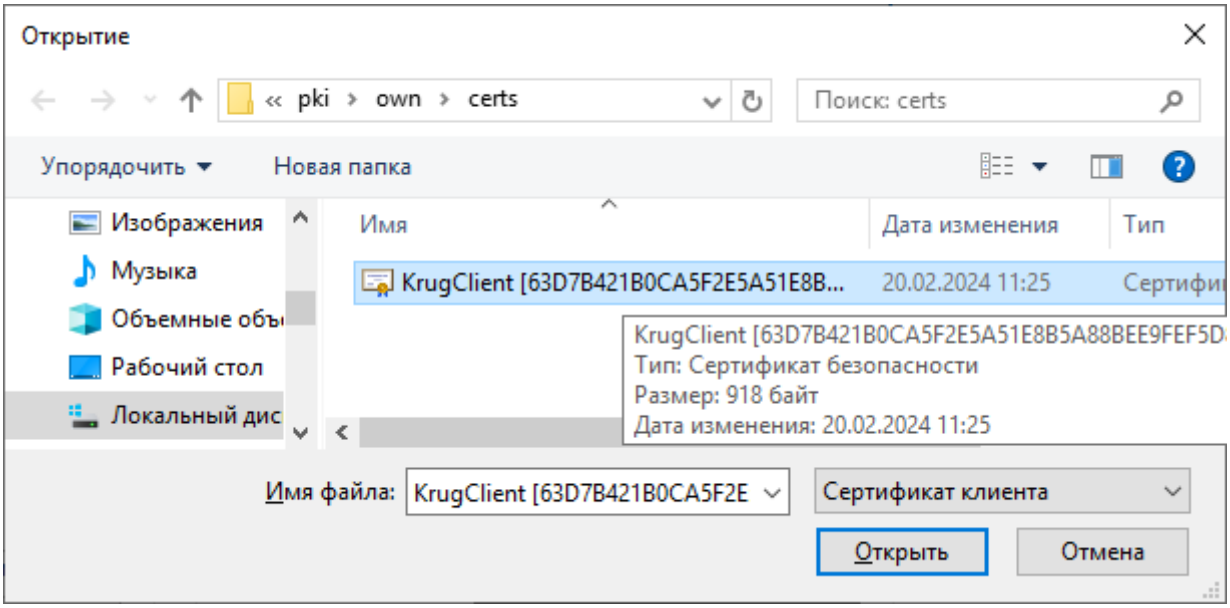
Если при попытке соединения будет выдано сообщение, что сертификат необходимо добавить в доверенные, то необходимо закрыть сообщение (кнопка «Close») и продолжить настройку.



4) В окне выбора конфигурации сервера нажать кнопку **Добавить сертификат сервера.** В окне открытия файла **выбрать** сертификат сервера для переноса его в каталог доверенных сертификатов. Перенос выбранного файла будет осуществлён автоматически при нажатии кнопки «Открыть».

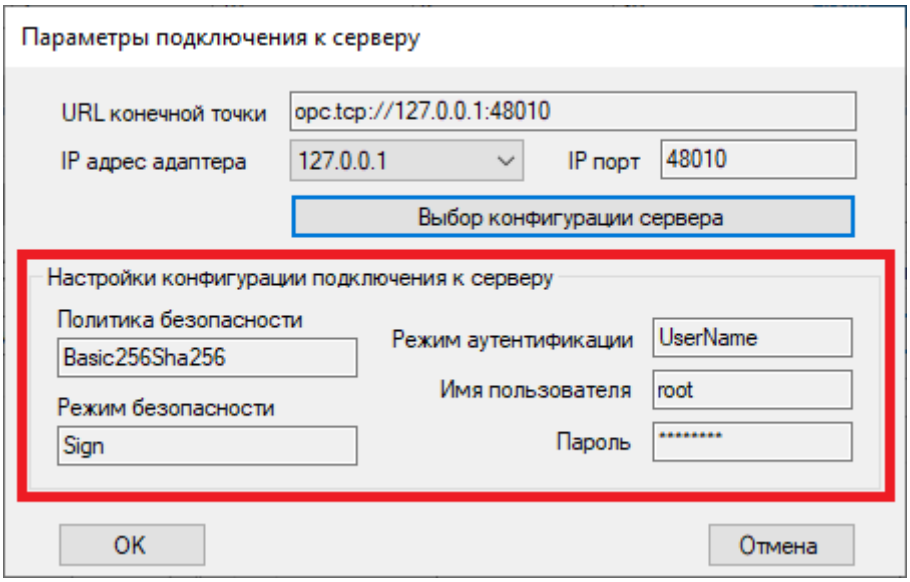


5) В окне выбора конфигурации сервера нажать кнопку **Расположение сертификата клиента**. В открывшемся окне можно скопировать файл сертификата, чтобы добавить его в папку доверенных сертификатов сервера. (*Расположение папок с сертификатами смотрите в документации к OPC UA серверу)*



6) После выполненных настроек (обмен сертификатами между сервером и клиентом) необходимо перейти к выбору конфигурации сервера и при необходимости ввести логин и пароль. Нажать кнопку «ОК».

7) Если соединение выполнено корректно, то при переходе в окно «Параметры подключения к серверу» в соответствии с выбранной конфигурацией будут заполнены поля: политика безопасности, режим безопасности, режим аутентификации, имя пользователя и пароль.



3.2.4 Редактирование и привязка тегов OPC UA сервера

3.2.4.1 Выбор оперативных или исторических тегов

Генератор базы данных предоставляет возможность работы с оперативными и историческими тегами OPC UA сервера.
При первом открытии вкладки со списком тегов загружается дерево тегов с OPC-сервера, выбранного на вкладке «Настройки» (рисунок 3.3). Данная вкладка предназначена для выбора OPC-тегов для дальнейшей привязки к переменным или перьям базы данных.

Вкладка разделена на две части: область дерева тегов, область описания тегов выбранной ветки дерева(рисунок 3.5 и рисунок 3.6).

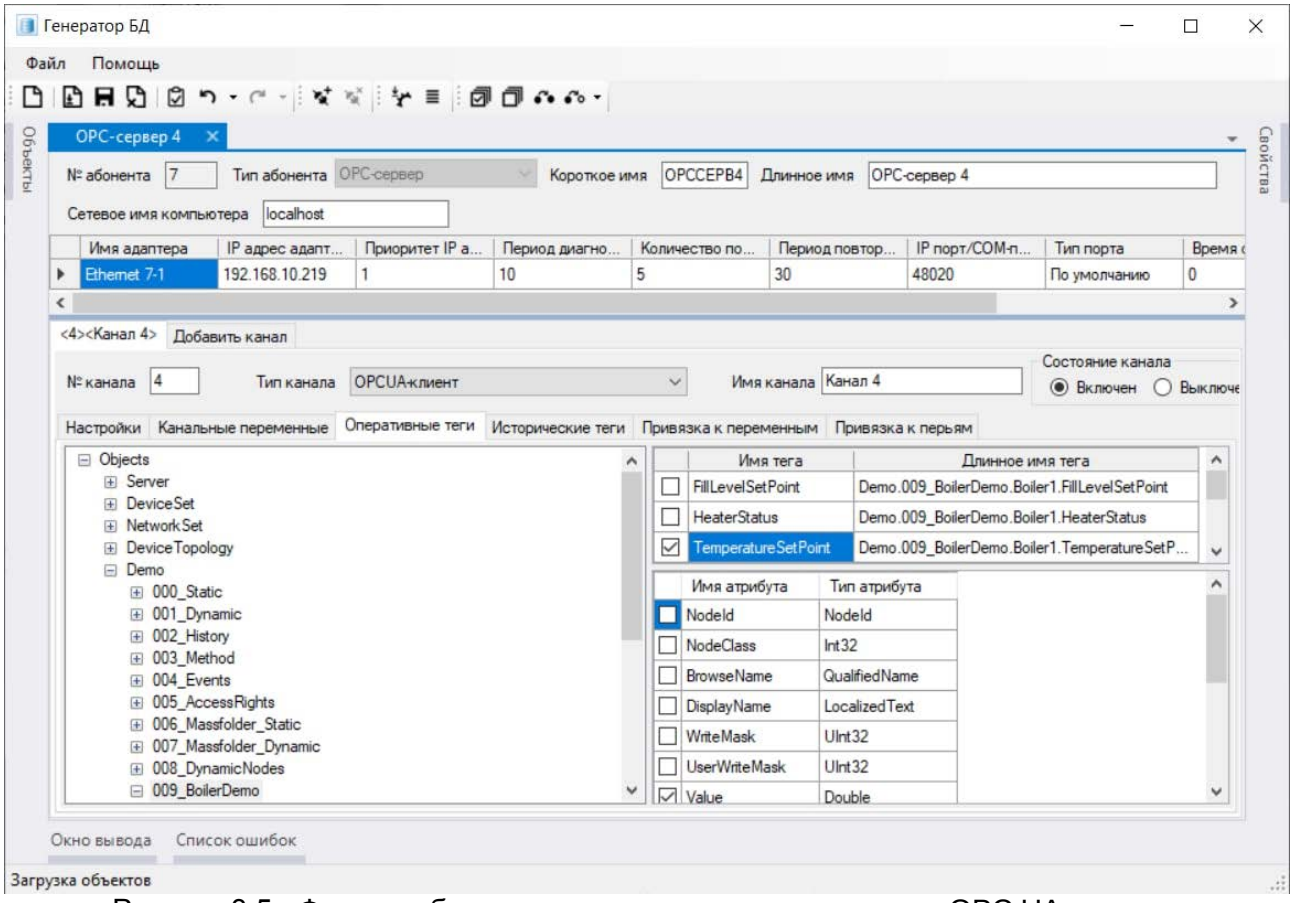


Рисунок 3.5 - Форма работы со списком **оперативных** тегов OPC UA сервера

Для создания привязки необходимо установить флажки напротив тегов. Дополнительные действия можно осуществить с помощью контекстного меню, которое вызывается с помощью нажатия правой кнопки мыши в соответствующей области.

Контекстное меню области дерева тегов:

- **Загрузить конфигурацию сервера –** позволяет обновить дерево тегов с OPC-сервера.
- **Показать все теги в подветках, начиная с выделенной** – в описании тегов отобразятся все теги выделенной ветки и всех ее подветок.

Контекстное меню области тегов:

- **Отметить все теги в списке** – устанавливает флажок выбора у всех тегов текущей выбранной ветки дерева.
- **Снять отметку со всех тегов в списке** – снимает флажок выбора у всех тегов текущей выбранной ветки дерева.
- **Создать привязки** – создает пустые привязки для тегов, для которых установлен флажок выбора. Дальнейшая привязка осуществляется на вкладке «Привязка к переменным» или на вкладке «Привязка к перьям» .

Для оперативных тегов доступен пункт меню

Создать привязки к новым переменным – создает привязки для тегов, отмеченных флажком, к новым переменным выбранного типа с атрибутами переменных по умолчанию.

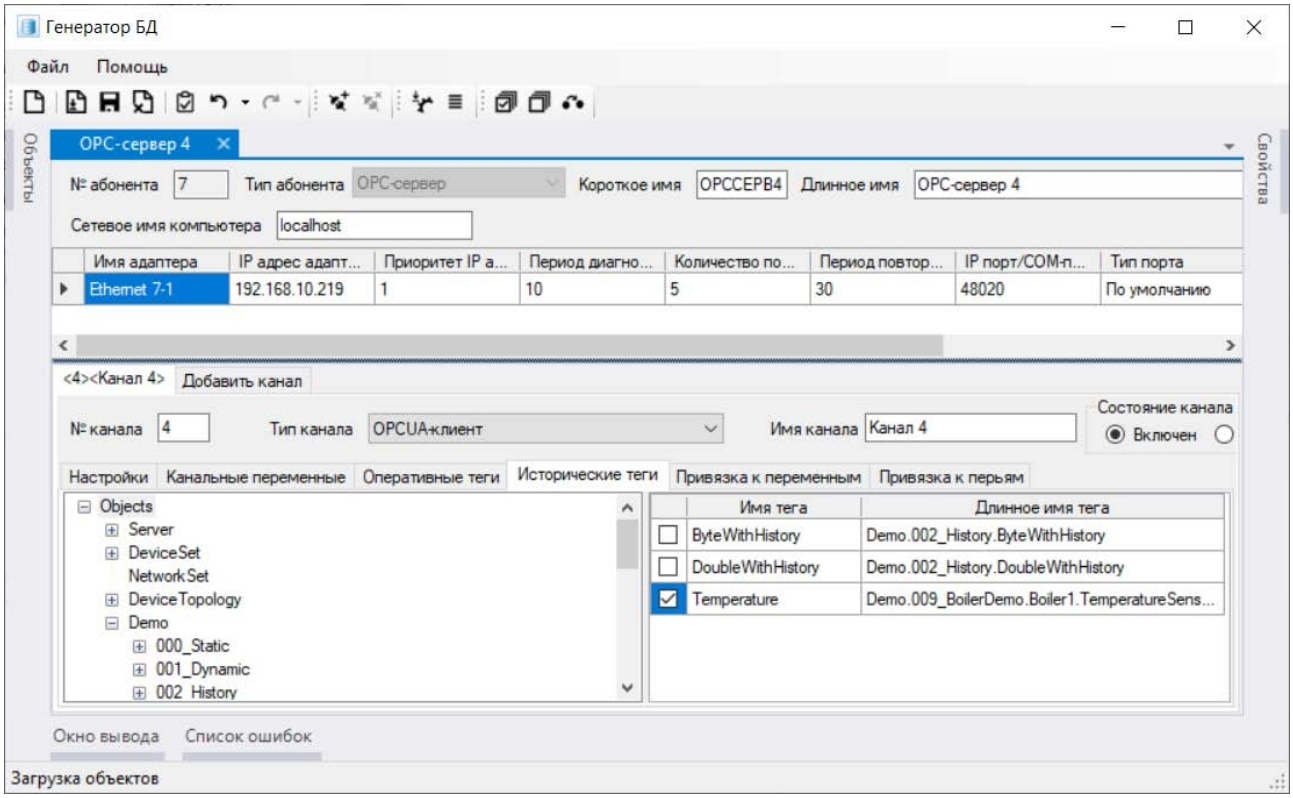


Рисунок 3.6 - Форма работы со списком **исторических** тегов OPC UA сервера

3.2.4.2 Привязка оперативных тегов к переменным

Привязка тегов OPC UA сервера к переменным базы данных необходима для того, чтобы OPC UA клиент знал, какие теги OPC сервера следует обрабатывать, и как сопоставить их с переменными базы данных.

Для задания привязки между тегами OPC-сервера и переменными базы данных используют вкладку «**Привязка к переменным»** (рисунок 3.7).

ВНИМАНИЕ !!!

Назначить тегам в соответствие можно только переменные, принадлежащие текущему выбранному каналу.
Одна строка таблицы привязок описывает одну привязку.

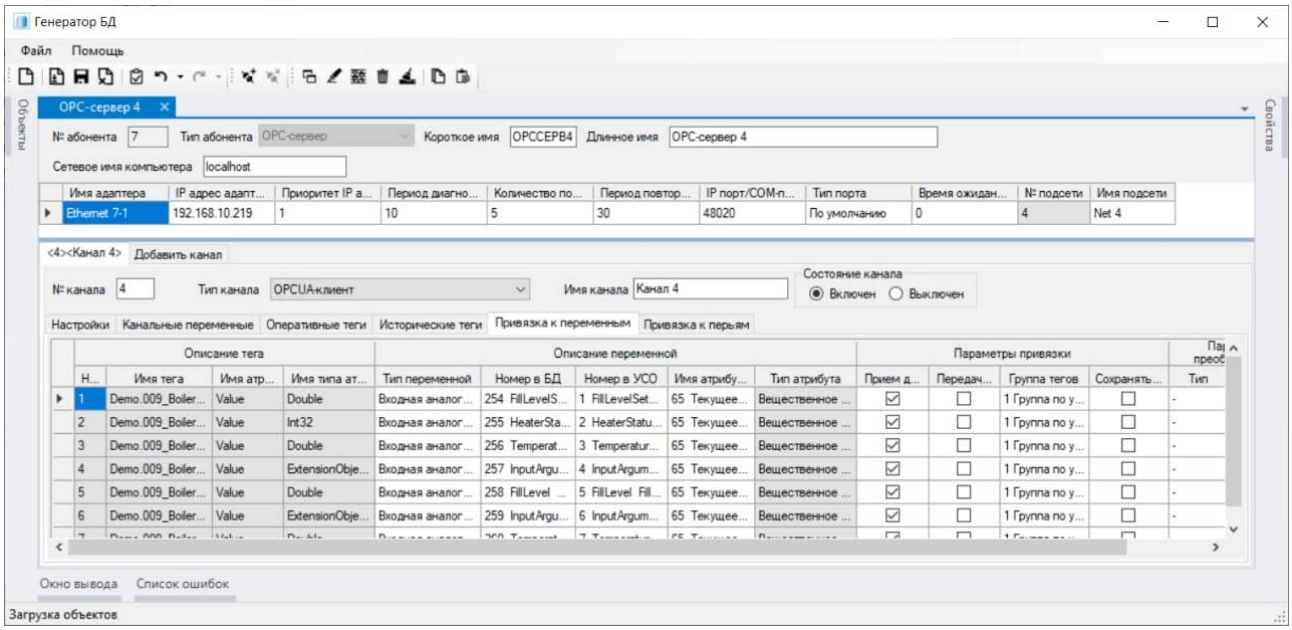


Рисунок 3.7 - Привязка тегов OPC-сервера к переменным БД

Поля в таблице привязок структурированы по разделам.

Описание тега

- **Номер тега –** уникальный номер для идентификации привязки, задается автоматически
- **Имя тега** – имя опрашиваемого тега. Добавлять свое имя тега нельзя. Содержимое списка определяется выбором записей на форме со списком тегов OPC UA-сервера
- **Атрибут** – имя атрибута опрашиваемого тега.
- **Имя типа атрибута** – название типа атрибута выбранного тега.

Описание переменной

- **Тип переменной** – набор типов зависит от наличия в канале переменных соответствующих типов
- **№ в БД** – раскрывающийся список, в котором можно выбрать переменную по общему номеру в БД, по позиции и по длинному имени.
- **№ в УСО** – раскрывающийся список, в котором можно выбрать переменную по номеру в УСО, по позиции и по длинному имени
- **Имя атрибута –** раскрывающийся список, в котором можно выбрать атрибут переменной по его имени и типу (целое, вещественное, строковое и т.д.)
- **Тип атрибута** - тип данных атрибута переменной, заполняется автоматически при выборе имени атрибута. Для логического типа может принимать значения «логическое» (тип данных атрибута по умолчанию - как он задан в паспорте переменной) или «логическое с отрицанием» (2 байта).

Параметры привязки

- **Прием данных** – определяет возможность приема данных от тега к переменной. Если стоит галочка, то прием данных будет работать. По умолчанию для переменных типа ВА и ВД в поле «Прием» стоит галочка, а для переменных ДВ и РВ - не стоит
- **Передача данных** – определяет возможность передачи данных от переменной к тегу. Если стоит галочка, то передача данных будет работать. По умолчанию для переменных типа ВА и ВД в поле «Передача» галочка не стоит, а для

переменных ДВ и РВ - стоит. При назначении в привязке атрибута тега, не представляющего текущее значение тега, в поле «Передача» галочка не ставится, а само это поле становится недоступным для редактирования

- **Группа тегов** (апертура) – номер (0 – группы нет) и имя группы тегов. Создать группы тегов можно в форме «Группы тегов», которая открывается при нажатии на одноименную кнопку на панели инструментов.
- **Сохранять качество** - если галочка установлена, то качество получаемого значения тега записывается в атрибут "Качество" соответствующей переменной

☝**ВНИМАНИЕ !!!**

В случае привязки атрибута тега, не предоставляющего текущее значение, в поле «Передача» галочка не ставится, а само это поле становится недоступным для редактирования.

Параметры преобразование

- Тип функции преобразования значения OPC UA-тега.
- N1 N2 параметры настройки функции преобразования

Типы функции преобразования:
<прочерк> - преобразование не выполняется,
***Bit** - из целой части значения тега выделяется бит с указанным номером – параметр N1 (диапазоны допустимых значений 0-31),*
***Byte** - из целой части значения тега выделяется байт с указанным номером - параметр N1 (диапазон допустимых значений 0-3),*
***Word** - из целой части значения тега выделяется слово (2байт) с указанным номером (диапазон допустимых значений 0-1),*
***Mask** - наложение маски (параметр N1) на целую часть значения тега со смещением результата вправо побитно (параметр N2),*
***Pulse** - сброс значения тега после выполнения команды управления (записи в OPCUA-сервер) через паузу в мсек (параметр N1) в значение, задаваемое параметром N2 (для вещественного типа атрибута тега – сброс в значение N2/1000).*

В контекстном меню для привязок доступна функция **дублирования** привязки. При выборе этого пункта меню будет создана копия записи (имя тега, атрибут и тип атрибута).

Группы тегов служат для объединения тегов в группы с целью установления каких-либо общих свойств для всех тегов в группе. Форма «**Группы тегов**» вызывается при нажатии соответствующей кнопки на панели инструментов.

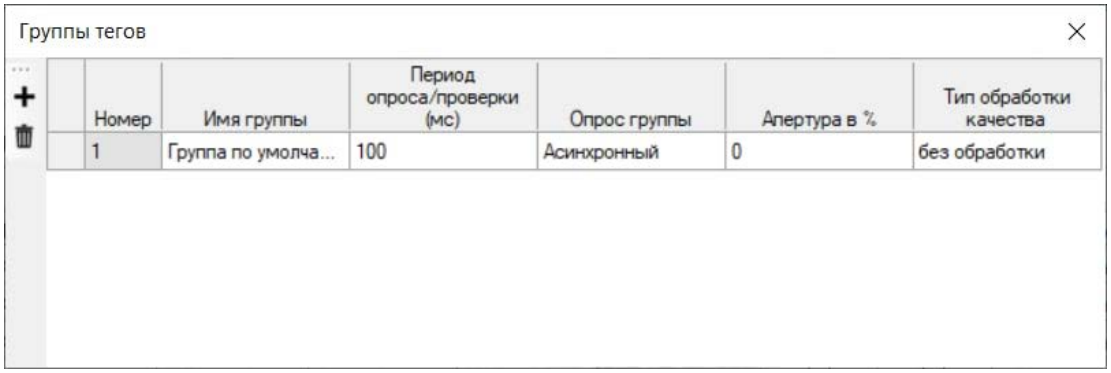


Рисунок 3.8 – Форма для задания групп тегов

☝**ВНИМАНИЕ!!!**

Апертура используется только при асинхронном опросе ОРС-сервера. По умолчанию все теги ОРС сервера образуют одну группу, апертура которой равна 0%. Обработка апертур осуществляется ОРС-сервером. Если ОРС-сервер не поддерживает обработку апертур, заданные апертуры для групп тегов действовать не будут.

- **Период опроса/проверки (мс)** - период опроса оперативных данных для группы тегов при синхронном режиме опроса / период проверки OPCUA-сервером изменения качества или значений оперативных данных на величину апертуры при асинхронном режиме опроса (по умолчанию 100 мс).
- **Опрос группы** – выбирается из выпадающего списка значений Асинхронный (по умолчанию), Синхронный.
- **Апертура в %** - величина изменения значения тега, вызывающая механизм его отправки подписчику при асинхронном режиме опроса (по умолчанию 0).

В зависимости от значения поля **Тип обработки качества тегов** обработка качества тегов, связанных с одной переменной, будет выполняться по различным алгоритмам:
- «**без обработки»** – значение по умолчанию. Обобщенное качество переменной определяется по наихудшему качеству тегов в группе привязок.
- «**с обработкой»** – на определение обобщенного качества по переменной влияет наличие привязки ОРС-тега в группе к атрибуту «Текущее значение» переменной соответственно.

Для:
- ВА – атрибут №28 «Текущее значение до преобразования (контроллер)»
- ВД – атрибут №27 «Текущее значение переменной»
- РВВ – атрибут №27 «Текущее значение»
- РВС – атрибут №13 «Текущее значение (Строка)»
- РВЛ – атрибут №14 «Текущее значение (логич)»
- ДВ – атрибут №20 «Значение выходной переменной в контроллере»
- АВ – атрибут №48 «Значение выходного сигнала (аналог.р-ра)».

При наличии такой привязки определяющим для обобщенного качества является качество данного тега, при его отсутствии – обобщенное качество принимается равным «GOOD».

Кнопка «**Автозаполнение**» – служит для вызова панели автоматического заполнения привязок. Заполнение таблицы привязок происходит с помощью выбора переменных из списка и переноса записей в таблицу привязок путём «захвата» (нажатием и удержанием левой кнопки мыши) выделенных в списке записей выбранного типа переменных и перемещении переменных поверх текущих привязок. Привязка будет осуществлена к атрибуту переменной, выбранному на панели автозаполнения.
В списке автозаполнения доступны только переменные текущего канала.
Направление обмена будет выставлено по типу переменной по умолчанию (для переменных типа ВА и ВД в поле «Прием», для переменных типа ДВ и РВ в поле «Передача»). Группа тегов по умолчанию будет задана №1.

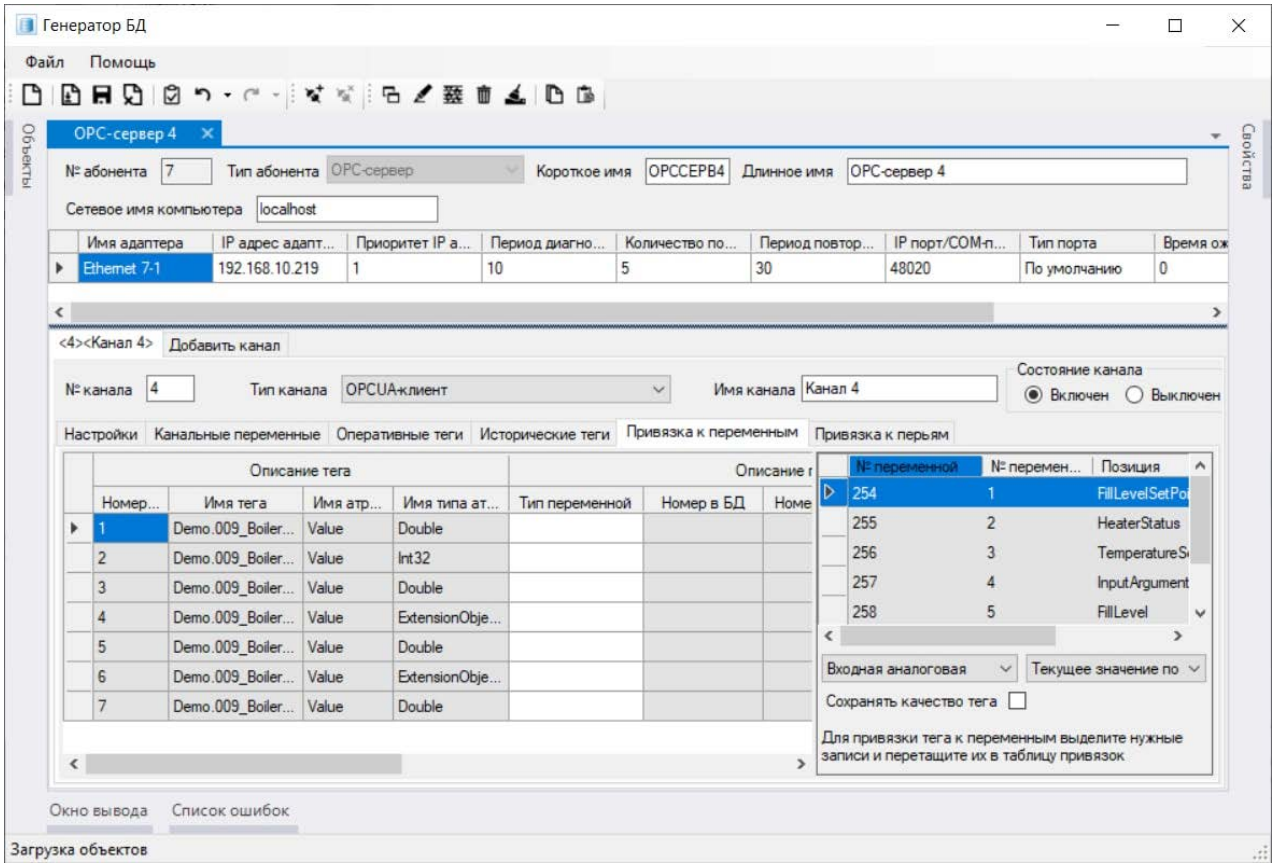


Рисунок 3.9 – Панель автозаполнения оперативных тегов

3.2.4.3 Привязка исторических тегов к перьям

Для того чтобы OPCUA-клиент знал какие теги OPCUA-сервера следует обрабатывать и как сопоставить их с перьями самописцев, необходимо произвести привязку тегов OPCUA-сервера к перьям базы данных. Для задания привязки между тегами OPC-сервера и перьями базы данных используют вкладку «**Привязка к перьям».**

Работа с привязками осуществляется в таблице привязок (рисунок 3.10), поля которой структурированы по разделам:

Описание тега:

- **Номер тега** – уникальный номер для идентификации привязки, задается автоматически**.**
- **Имя тега** – имя опрашиваемого тега. Добавлять свое имя тега нельзя. Список определяется выбором записей на форме со списком тегов OPCUA-сервера

Описание пера:

- **Самописец** – номер самописца и имя самописца в БД. Выбирается из раскрывающегося списка

ВНИМАНИЕ !!!

В списке самописцев присутствуют только самописцы для выбранного канала.

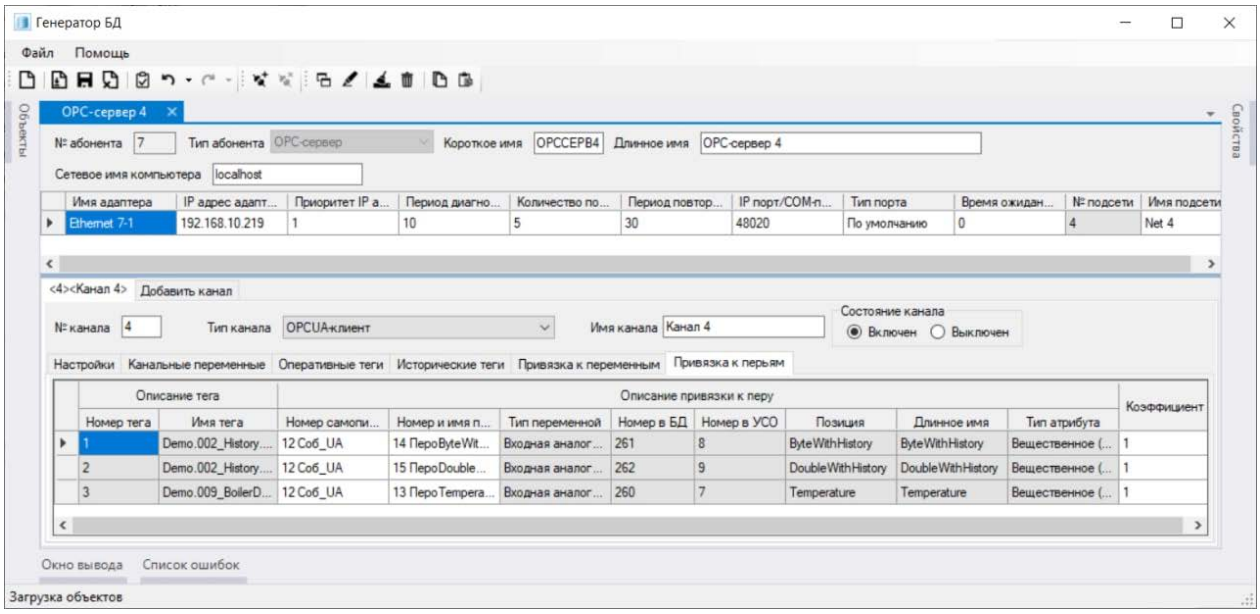


Рисунок 3.10 - Привязка исторических тегов ОРС-сервера к перьям БД

- **№ и имя пера** – номер и имя пера самописца, в которое будет производиться запись данных тега OPCUA-сервера. Выбирается из раскрывающегося списка.
- **Тип переменной, № в БД, № в УСО, Позиция и Длинное имя переменной –** атрибуты переменной, соответствующей выбранному перу. Нередактируемые информационные поля
- **Тип атрибута –** тип данных атрибута переменной, являющейся источником пера (информационное поле), заполняется при выборе пера

- **Коэффициент –** величина, на которую умножаются значения тега ОРС UA сервера перед записью в выбранное перо. При этом время и качество тега ОРС UA сервера остаются неизменными.

ВНИМАНИЕ !!!

Источником пера являются переменные, привязанные к текущему каналу ОРС UA, выбранному в самописце. В случае необходимости визуализировать дополнительные параметры пера, например такие как «единицы измерения», «позиция», будут использоваться атрибуты переменной, связанной с данным пером.

Кнопка «**Автозаполнение**» – служит для вызова панели автоматического заполнения привязок тегов OPCUA-сервера к существующим перьям выбранного самописца БД. Заполнение таблицы привязок происходит с помощью выбора перьев из списка и переноса записей в таблицу привязок путём «захвата» (нажатием и удержанием левой кнопки мыши) выделенных в списке записей перьев выбранного самописца, и перемещением их поверх текущих привязок.

4 КРУГ OPC Toolkit

4.1 Назначение и функции

Библиотека функций «КРУГ OPC Toolkit» предназначена для разработки OPC-клиентов.

Библиотека скрывает в себе механизмы OPC и COM (Component Object Model), которые использует OPC-клиент в своей работе.
Разработчик OPC-клиента путем вызова экспортируемых библиотекой функций получает доступ к данным OPC-серверов, не заботясь о реализации механизмов доступа OPC, COM и DCOM (Distributed COM). Это позволяет сократить время разработки OPC-клиента и уделить большее внимание другим аспектам его работы.

В функции «**КРУГ OPC Toolkit**» входит:
- Поиск установленных в операционной системе OPC-серверов
- Получение пространства имен (списка) тегов выбранного OPC-сервера
- Синхронный и асинхронный доступ к данным выбранных тегов OPC-сервера
- Доступ к значениям атрибутов (свойств) выбранного тега (если таковые имеются).

Все вышеперечисленные действия могут выполняться как на локальном, так и на удаленном компьютере.

Работа с библиотекой предполагает знание Пользователем основных положений технологии OPC.

4.2 Комплект поставки

В каталоге, куда Вы установили «**КРУГ OPC Toolkit**», находятся:
- Папка **Common**, содержащая стандартные объекты и файлы OPC:
  - **OpcError.h** – файл с описанием стандартных кодов операций OPC-серверов
  - **OPCProps.h** – файл с описанием стандартных идентификаторов атрибутов тегов OPC-серверов.

- Папка **Example**, содержащая проект демонстрационного OPC-клиента.

- Файл **KrugOPCDemoClient.exe** – исполняемый файл демонстрационного OPC-клиента, реализованного в среде программирования Microsoft Visual C++.
- Файл **OPCClientDemoDelphi.exe** – исполняемый файл демонстрационного OPC-клиента, реализованного в среде программирования Borland Delphi.
- Файл **KrugOPCToolkit.dll** – библиотека «КРУГ OPC Toolkit».
- Файл **KrugOPCToolkit_Exports.h** – файл описания экспортируемых функций «КРУГ OPC Toolkit».
- Файл **KrugOPCToolkit.lib** – lib-файл для подключения «КРУГ OPC Toolkit» к проекту в среде программирования Visual C++.
- Файл **KrugOPCToolkitRegistration.exe** – программа для регистрации «КРУГ OPC Toolkit».

Для обеспечения работы механизмов OPC в системную директорию **Windows** в папку **system32** будут записаны и зарегистрированы в реестре Windows следующие объекты:
- **OpcEnum.exe** – утилита для поиска зарегистрированных в системе OPC-серверов

- **OPCComn_ps.dll** - объект для работы с DCOM
- **OPCProxy.dll** – объект для работы с DCOM.

При деинсталляции «КРУГ OPC Toolkit» эти объекты не будут удаляться и разрегистрироваться во избежание конфликта с другими OPC-приложениями.

4.3 Схема использования

Библиотека «КРУГ OPC Toolkit» может быть подключена к проекту OPC-клиента двумя способами:

- с помощью поставляемого ***LIB-файла***. Если вы используете язык программирования C/C++, то подключите к вашему проекту файл с описанием функций (***KrugOPCToolkit_Exports.h***) и добавьте в этап «линковки» проекта LIB-файл (***KrugOPCToolkit.lib***). LIB-файл создан в среде программирования Microsoft Visual C++ 6.0;
- с помощью вызова функции WinAPI ***LoadLibrary().*** В этом случае для каждой из функций библиотеки нужно будет вызвать функцию WinAPI ***GetProcAddress()***. Этот метод универсален и может быть применен в любых проектах на базе операционной системы Windows.

Для удобства использования в файле *KrugOPCToolkit_Exports.h* все функции объявлены как для подключения первым способом, так и вторым.

При взаимодействии с OPC-серверами следует придерживаться правил, установленных стандартами OPC. Поэтому, применяя «КРУГ OPC Toolkit», необходимо использовать следующую схему вызова функций (описание функций приведено в подразделе 4.4):

1) Инициализировать библиотеку вызовом функции ***Initialize()***;
2) Циклическим вызовом функции ***GetNextServerName()*** получить имена OPC-серверов, установленных в операционной системе.
3) Вызовом функции ***Connect()*** установить соединение с выбранным OPC-сервером.
4) Циклическим вызовом функции ***GetNextServerItem()*** получить имена тегов OPC-сервера.
5) Пока соединение с OPC-сервером не закрыто, можно получать данные атрибутов какого-либо тега по следующей схеме:
   - циклическим вызовом функции ***GetNextItemAttribute()*** получить идентификаторы доступных атрибутов выбранного тега;
   - получить данные выбранных атрибутов вызовом функции ***ReadItemAttributes()***.
6) Открыть доступ к данным тегов OPC-сервера с помощью функции ***OpenAccess()***.
7) Для асинхронного доступа к данным OPC-сервера вызвать функцию ***AdviseItems().*** В качестве одного из параметров *AdviseItems* передать указатель на функцию обратного вызова (callback). Эту функцию «КРУГ OPC Toolkit» будет вызывать при получении уведомления об изменении данных OPC-сервера.
8) Для синхронного доступа к данным OPC-сервера вызывать функцию ***ReadItems()*** или ***ReadItemsOption()*** с заданной периодичностью.
9) Для записи данных в теги OPC-сервера вызывать функцию ***WriteItems()***.
10) При завершении асинхронного доступа к данным OPC-сервера вызвать функцию ***UnadviseItems()*** для отмены подписки и освобождения ресурсов OPC-сервера.
11) При завершении опроса OPC-сервера вызвать функцию ***CloseAccess()*** для закрытия доступа к данным OPC-сервера и освобождения его ресурсов.
12) При завершении работы с OPC-сервером вызвать функцию ***Disconnect()*** для корректного отсоединения от OPC-сервера и освобождения его ресурсов.
13) Завершить работу с библиотекой вызовом функции ***Uninitialize()***.

Если необходимо опрашивать несколько серверов одновременно следует повторить пункты 3 – 12 для каждого ОРС-сервера.
Пример использования функций «КРУГ ОРС Toolkit» описан в пункте 4.5.

4.4 Описание экспортируемых функций

Все функции имеют тип возвращаемого значения ***HRESULT*.**

Коды возврата функции проверяются с помощью макросов COM ***SUCCEEDED*** (успешное завершение) и ***FAILED*** (ошибочное завершение).

Стандартные коды возврата для ОРС-серверов приведены в поставляемом файле ***OpcError.h.***

Стандартные идентификаторы атрибутов тегов ОРС-сервера приведены в поставляемом файле ***OPCProps.h.***

Текущие значения тегов ОРС-сервера имеют часто используемый в COM-приложениях тип данных ***VARIANT*,** конечный вид которого задается с помощью перечисления ***VARTYPE***.

HRESULT **Initialize** ()
Назначение: Начать работу с библиотекой **Возвращаемая величина**: код результата операции.

HRESULT **GetNextServerName** ([in] LPWSTR **wszRemoteMashineName**, [in,out] LPWSTR **wszServerName**)
Назначение: Получить имя из списка зарегистрированных в операционной системе ОРС-серверов. В параметрах используются UNICODE-строки **Параметры**: **wszRemoteMashineName** — сетевое имя машины, на которой необходимо найти ОРС-серверы; **wszServerName** — имя ОРС-сервера. **Возвращаемая величина**: код результата операции. **Комментарии:** Если **wszRemoteMashineName** — пустая строка, то попытка найти ОРС-серверы будет предпринята на локальной машине. Параметр **wszServerName** используется как входной и выходной. Если параметр **wszServerName** — пустая строка, то в нем будет возвращено первое имя в списке. Если параметр **wszServerName** — имя предыдущего полученного сервера, то в нем будет возвращено следующее имя и т.д. Если в параметре **wszServerName** возвращена пустая строка, то достигнут конец списка. Все параметры должны быть инициализированы Пользователем.

HRESULT **GetNextServerNameA** ([in] LPSTR **szRemoteMashineName**,
[in,out] LPSTR **szServerName**)

Назначение: Получить имя из списка зарегистрированных в операционной системе OPC-серверов
В параметрах используются ANSI-строки

Параметры:
szRemoteMashineName — сетевое имя машины, на которой необходимо найти OPC-серверы;
szServerName — имя OPC-сервера.

Возвращаемая величина:
код результата операции.

Комментарии:
Если **szRemoteMashineName** — пустая строка, то попытка найти OPC-серверы будет предпринята на локальной машине.
Параметр **szServerName** используется как входной и выходной. Если параметр **szServerName** — пустая строка, то в нем будет возвращено первое имя в списке. Если параметр **szServerName** — имя предыдущего полученного сервера, то в нем будет возвращено следующее имя и т.д. Если в параметре **szServerName** возвращена пустая строка, то достигнут конец списка.
Все параметры должны быть инициализированы Пользователем.

HRESULT **Connect** ([in] LPWSTR **wszRemoteMashineName**,
[in,out] LPWSTR **wszServerName**, [out] DWORD ***pdwConnectID**)

Назначение: Осуществить соединение с OPC-сервером
В параметрах используются UNICODE-строки

Параметры:
wszRemoteMashineName — сетевое имя машины, на которой предполагается работа OPC-сервера;
wszServerName — имя OPC-сервера;
pdwConnectID — идентификатор соединения.

Возвращаемая величина:
код результата операции.

Комментарии:
Если wszRemoteMashineName — пустая строка, то попытка установить связь будет предпринята на локальной машине.
Параметр wszServerName можно получить с помощью функции GetNextServerName() или задать заранее известное имя сервера.
Параметр pdwConnectID — уникальный идентификатор, который используется большинством функций библиотеки для однозначного определения места нахождения и имени OPC-сервера.
Все параметры должны быть инициализированы Пользователем.

HRESULT **ConnectA** ([in] LPSTR **szRemoteMashineName**,
[in,out] LPSTR **szServerName**, [out] DWORD ***pdwConnectID**)

Назначение: Осуществить соединение с ОРС-сервером
В параметрах используются ANSI-строки

Параметры:

szRemoteMashineName — сетевое имя машины, на которой предполагается работа ОРС-сервера;
szServerName — имя ОРС-сервера;
pdwConnectID — идентификатор соединения.

Возвращаемая величина:

код результата операции.

Комментарии:

Если **szRemoteMashineName** — пустая строка, то попытка установить связь будет предпринята на локальной машине.
Параметр **szServerName** можно получить с помощью функции **GetNextServerName()** или задать заранее известное имя сервера.
Параметр **pdwConnectID** — уникальный идентификатор, который используется большинством функций библиотеки для однозначного определения места нахождения и имени ОРС-сервера.
Все параметры должны быть инициализированы Пользователем.

HRESULT **GetNextServerItem** ([in] DWORD **dwConnectID**,
[out] LPWSTR **wszItemName**)

Назначение: Получить имя из списка тегов ОРС-сервера
В параметрах используются UNICODE-строки

Параметры:

dwConnectID — идентификатор соединения;
wszItemName — имя тега.

Возвращаемая величина:

код результата операции.

Комментарии:

Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Параметр **wszItemName** используется как входной и выходной. Если параметр **wszItemName** — пустая строка, то в нем будет возвращено первое имя в списке. Если параметр **wszItemName** — имя предыдущего полученного тега, то в нем будет возвращено следующее имя и т.д. Если в параметре **wszItemName** возвращена пустая строка, то достигнут конец списка.
Все параметры должны быть инициализированы Пользователем.

HRESULT **GetNextServerItemA** ([in] DWORD **dwConnectID**,
[out] LPSTR **szItemName**)

Назначение: Получить имя из списка тегов OPC-сервера
В параметрах используются ANSI-строки

Параметры:

dwConnectID — идентификатор соединения;
szItemName — имя тега.

Возвращаемая величина:

код результата операции.

Комментарии:

Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Параметр **szItemName** используется как входной и выходной. Если параметр **szItemName** — пустая строка, то в нем будет возвращено первое имя в списке. Если параметр **szItemName** — имя предыдущего полученного тега, то в нем будет возвращено следующее имя и т.д. Если в параметре **szItemName** возвращена пустая строка, то достигнут конец списка.
Все параметры должны быть инициализированы Пользователем.

HRESULT **OpenAccess** ([in] DWORD **dwConnectID**, [in]DWORD **dwCount**,
[in,size_is(dwCount)] LPWSTR ***pwszItemsNames**,
[in,size_is(dwCount)] VARTYPE ***pvtItemsTypes**,
[in] DWORD **dwRequestedUpdateRate**,
[in] float **fPercentDeadband**,
[out,size_is(dwCount)] VARTYPE ***pvtNativeTypes**,
[out] DWORD ***pdwRevisedUpdateRate**,
[out,size_is(dwCount)] HRESULT ***pErrors**,
[out,size_is(dwCount)] DWORD ***pdwItemsIDs**,
[out] DWORD ***pdwAccessID**)

Назначение: Открыть доступ к данным выбранных тегов OPC-сервера
В параметрах используются UNICODE-строки

Параметры:

dwConnectID — идентификатор соединения;
dwCount — количество выбранных тегов;
pwszItemsNames — массив имен выбранных тегов;
pvtItemsTypes — массив требуемых типов данных для тегов;
dwRequestedUpdateRate — запрашиваемая частота обновления данных в OPC-сервере в миллисекундах
fPercentDeadband — апертура (диапазон, при выходе за который текущего значения, считается, что текущее значение изменилось; иначе – не изменилось) в % от максимального диапазона изменения значения тега. Обычно используется для подавления шумов (мелких хаотических отклонений) и «дребезга» при опросе тегов.
pvtNativeTypes — массив «родных» типов данных тегов OPC-сервера;
pdwRevisedUpdateRate — частота обновления данных, установленная OPC-сервером, в миллисекундах;
pErrors — массив результатов операций для каждого тега;
pdwItemsIDs — массив идентификаторов тегов;
pdwAccessID — идентификатор доступа.

Возвращаемая величина:

HRESULT **OpenAccess** ([in] DWORD **dwConnectID**, [in]DWORD **dwCount**, [in,size_is(dwCount)] LPWSTR ***pwszItemsNames**, [in,size_is(dwCount)] VARTYPE ***pvtItemsTypes**, [in] DWORD **dwRequestedUpdateRate**, [in] float **fPercentDeadband**, [out,size_is(dwCount)] VARTYPE ***pvtNativeTypes**, [out] DWORD ***pdwRevisedUpdateRate**, [out,size_is(dwCount)] HRESULT ***pErrors**, [out,size_is(dwCount)] DWORD ***pdwItemsIDs**, [out] DWORD ***pdwAccessID**)
код результата операции.

Комментарии:

Идентификатор соединения **dwConnectID** возвращает функция **Connect().**

Имена тегов **pwszItemsNames** можно получить с помощью функции **GetNextServerItem.**

Типы данных для тегов **pvtItemsTypes** задаются в соответствии с требованиями ОРС-клиента (например, вещественное значение — VT_R4, VT_R8, целое значение VT_I4, логическое значение — VT_BOOL и т.д.), т.е. клиент сообщает ОРС-серверу о том, в каком виде он хотел бы получать данные.

Также клиент указывает ОРС-серверу с какой частотой (**dwRequestedUpdateRate)** сервер должен обновлять данные для клиента.

Сервер, в свою очередь, возвращает ОРС-клиенту какие типы данных у его тегов были изначально (**pvtNativeTypes)** , а также какую частоту обновления данных для ОРС-клиента он в действительности установил (**pdwRevisedUpdateRate**).

ОРС-сервер возвращает массив результатов операции для каждого тега **pErrors**. Если результат операции для тега неуспешен (например, сервер не смог преобразовать данные из «родного» типа в запрашиваемый), то соответствующий тегу элемент массива **pErrors** содержит код ошибки, и все последующие операции с этим тегом приведут к неудаче.

«КРУГ ОРС Toolkit» каждому тегу присваивает уникальные идентификаторы **pdwItemsIDs** (в пределах данного открытого доступа), которые потом использует при передаче данных для точного указания на теги.

Параметр **pdwAccessID** — уникальный идентификатор (в пределах данного установленного соединения), который используется функциями доступа к данным для однозначного определения параметров доступа.

Все параметры должны быть инициализированы Пользователем.

HRESULT **OpenAccessA** ([in] DWORD **dwConnectID**, [in]DWORD **dwCount**,
[in,size_is(dwCount)] LPSTR ***pszItemsNames**,
[in,size_is(dwCount)] VARTYPE ***pvtItemsTypes**,
[in] DWORD **dwRequestedUpdateRate**,
[in] float **fPercentDeadband**,
[out,size_is(dwCount)] VARTYPE ***pvtNativeTypes**,
[out] DWORD ***pdwRevisedUpdateRate**,
[out,size_is(dwCount)] HRESULT ***pErrors**,
[out,size_is(dwCount)] DWORD ***pdwItemsIDs**,
[out] DWORD ***pdwAccessID**)

Назначение: Открыть доступ к данным выбранных тегов OPC-сервера
В параметрах используются ANSI-строки

Параметры:

dwConnectID — идентификатор соединения;
dwCount — количество выбранных тегов;
pszItemsNames — массив имен выбранных тегов;
pvtItemsTypes — массив требуемых типов данных для тегов;
dwRequestedUpdateRate — запрашиваемая частота обновления данных в OPC-сервере в миллисекундах
fPercentDeadband — апертура (диапазон, при выходе за который текущего значения, считается, что текущее значение изменилось; иначе – не изменилось) в % от максимального диапазона изменения значения тега. Обычно используется для подавления шумов (мелких хаотических отклонений) и «дребезга» при опросе тегов.
pvtNativeTypes — массив «родных» типов данных тегов OPC-сервера;
pdwRevisedUpdateRate — частота обновления данных, установленная OPC-сервером, в миллисекундах;
pErrors — массив результатов операций для каждого тега;
pdwItemsIDs — массив идентификаторов тегов;
pdwAccessID — идентификатор доступа.

Возвращаемая величина:

код результата операции.

Комментарии:

Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Имена тегов **pszItemsNames** можно получить с помощью функции **GetNextServerItem.**
Типы данных для тегов **pvtItemsTypes** задаются в соответствии с требованиями OPC-клиента (например, вещественное значение — VT_R4, VT_R8, целое значение VT_I4, логическое значение — VT_BOOL и т.д.), т.е. клиент сообщает OPC-серверу о том, в каком виде он хотел бы получать данные.
Также клиент указывает OPC-серверу с какой частотой (**dwRequestedUpdateRate)** сервер должен обновлять данные для клиента.
Сервер, в свою очередь, возвращает OPC-клиенту какие типы данных у его тегов были изначально (**pvtNativeTypes)** , а также какую частоту обновления данных для OPC-клиента он в действительности установил (**pdwRevisedUpdateRate**).
OPC-сервер возвращает массив результатов операции для каждого тега **pErrors**. Если результат операции для тега неуспешен (например, сервер не смог преобразовать данные из «родного» типа в запрашиваемый), то соответствующий тегу элемент массива **pErrors** содержит код ошибки, и все последующие операции с этим тегом приведут к неудаче.

HRESULT **OpenAccessA** ([in] DWORD **dwConnectID**, [in]DWORD **dwCount**,
[in,size_is(dwCount)] LPSTR ***pszItemsNames**,
[in,size_is(dwCount)] VARTYPE ***pvtItemsTypes**,
[in] DWORD **dwRequestedUpdateRate**,
[in] float **fPercentDeadband**,
[out,size_is(dwCount)] VARTYPE ***pvtNativeTypes**,
[out] DWORD ***pdwRevisedUpdateRate**,
[out,size_is(dwCount)] HRESULT ***pErrors**,
[out,size_is(dwCount)] DWORD ***pdwItemsIDs**,
[out] DWORD ***pdwAccessID**)

«КРУГ OPC Toolkit» каждому тегу присваивает уникальные идентификаторы **pdwItemsIDs** (в пределах данного открытого доступа), которые потом использует при передаче данных для точного указания на теги.
Параметр **pdwAccessID** — уникальный идентификатор (в пределах данного установленного соединения), который используется функциями доступа к данным для однозначного определения параметров доступа.
Все параметры должны быть инициализированы Пользователем.

HRESULT **AdviseItems** ([in] DWORD **dwConnectID**, [in] DWORD **dwAccessID**,
[in] void ***pCallbackFunc**)

Назначение: Подписаться на получение данных от OPC-сервера
Параметры:
dwConnectID — идентификатор соединения;
dwAccessID — идентификатор доступа;
pCallbackFunc — указатель на функцию обратного вызова (callback).

Возвращаемая величина:
код результата операции.
Комментарии:
Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Идентификатор доступа **dwAccessID** возвращает функция **OpenAccess().**
В параметре **pCallbackFunc** следует передать указатель на функцию, которую «КРУГ OPC Toolkit» будет вызывать при получении уведомления от OPC-сервера об изменении данных. Описание функции смотри ниже.

void **KrugOPCCallbackFunc** (DWORD **dwConnectID**, DWORD **dwAccessID**,
HRESULT **hrMaserError**, DWORD **dwCount**, DWORD
***pdwItemsIDs**, VARIANT ***pvValues**, WORD ***pwQualities**,
FILETIME ***pftTimeStamps**, HRESULT ***pErrors**)

Назначение: Функция обратного вызова (callback) для «КРУГ OPC Toolkit». Вызывается библиотекой в OPC-клиенте при асинхронном опросе OPC-сервера.
Параметры:
dwConnectID — идентификатор соединения;
dwAccessID — идентификатор доступа;
hrMaserError — общий результат операции чтения данных. Также используется для передачи признака завершения работы OPC-сервера;
dwCount — количество переданных тегов;
pdwItemsIDs — массив идентификаторов тегов;
pvValues — массив текущих значений тегов;
pwQualities — массив признаков достоверности данных;

pftTimeStamps — массив меток времени тегов;
pErrors — массив результатов операции чтения для каждого тега.

Комментарии:

OPC-клиент должен объявить и определить эту функцию для того, чтобы осуществлять асинхронный опрос OPC-сервера. «КРУГ OPC Toolkit» будет вызывать эту функцию всякий раз, когда от OPC-сервера придет уведомление об изменении данных, и будет передавать в параметрах данные тегов и результаты операций. Для однозначного определения источника данных следует использовать параметры **dwConnectID** и **dwAccessID**, а принадлежность данных тегам определяется массивом их идентификаторов **pdwItemsIDs**.

Если в параметре «КРУГ OPC Toolkit» возвращает значение *S_OPC_SERVER_SHUTDOWN* (см. файл *KrugOPCToolkit_Exports.h*), это означает, что OPC-сервер готовится завершить свою работу, и OPC-клиент должен принять все меры по корректному отсоединению от сервера и сигнализации об этом.

HRESULT **ReadItems** ([in] DWORD **dwConnectID**, [in] DWORD **dwAccessID**, [out] DWORD ***pdwItemsIDs**, [out] VARIANT ***pvValues**, [out] WORD ***pwQualities**, [out] FILETIME ***pftTimeStamps**, [out] HRESULT ***pErrors**)

Назначение: Получить данные тегов от OPC-сервера

Параметры:

dwConnectID — идентификатор соединения;
dwAccessID — идентификатор доступа;
pdwItemsIDs — массив идентификаторов тегов OPC-сервера;
pvValues — массив данных тегов;
pwQualities — массив признаков достоверности данных тегов.
pftTimeStamps — массив меток времени тегов.
pErrors — массив результатов операции для каждого тега.

Возвращаемая величина:

код результата операции.

Комментарии:

Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Идентификатор доступа **dwAccessID** возвращает функция **OpenAccess().**
Все выходные массивы имеют длину, равную значению параметра **dwCount** при открытии доступа функцией **OpenAccess()**.
Определить принадлежность данных тегам можно с помощью массива идентификаторов тегов **pdwItemsIDs**.
Все параметры должны быть инициализированы Пользователем.

HRESULT **ReadItemsOptions** ([in]DWORD **dwSource**, [in] DWORD **dwConnectID**,
[in] DWORD **dwAccessID**, [out] DWORD *__pdwItemsIDs__,
[out] VARIANT *__pvValues__, [out] WORD *__pwQualities__,
[out] FILETIME *__pftTimeStamps__, [out] HRESULT *__pErrors__)

Назначение: Получить данные тегов от OPC-сервера с возможностью выбора источника данных (чтение с устройства или чтение из кэша OPC-сервера)

Параметры:

dwSource — источник данных;
dwConnectID — идентификатор соединения;
dwAccessID — идентификатор доступа;
pdwItemsIDs — массив идентификаторов тегов OPC-сервера;
pvValues — массив данных тегов;
pwQualities — массив признаков достоверности данных тегов.
pftTimeStamps — массив меток времени тегов.
pErrors — массив результатов операции для каждого тега.

Возвращаемая величина:

код результата операции.

Комментарии:

Параметр **dwSource** может принимать значения *OPC_FROM_CACHE* (чтение из кэша) и *OPC_FROM_DEVICE* (чтение с устройства, см. файл *KrugOPCToolkit_Exports.h*). Источником данных может выступать устройство, подключенное к OPC-серверу или кэш OPC-сервера. Чтение значений тегов из кэша требует меньше времени. Чтение с устройства позволяет получить последние данные на момент запроса. Вызов функции с параметром *OPC_FROM_CACHE* идентичен вызову функции *ReadItems().*
Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Идентификатор доступа **dwAccessID** возвращает функция **OpenAccess().**
Все выходные массивы имеют длину, равную значению параметра **dwCount** при открытии доступа функцией **OpenAccess()**.
Определить принадлежность данных тегам можно с помощью массива идентификаторов тегов **pdwItemsIDs**.
Все параметры должны быть инициализированы Пользователем.

HRESULT **WriteItems** ([in] DWORD **dwConnectID**, [in] DWORD **dwAccessID**,
[in]DWORD **dwCount**,
[in,size_is(dwCount)] DWORD *__pdwItemsIDs__, [in,size_is(dwCount)]
VARIANT *__pvValues__, [out,size_is(dwCount)] HRESULT *__pErrors__)

Назначение: Поместить данные в OPC-сервер

Параметры:

dwConnectID — идентификатор соединения;
dwAccessID — идентификатор доступа;
dwCount — количество тегов, в которые нужно поместить данные.
pdwItemsIDs — массив идентификаторов тегов.
pvValues — значения, которые нужно поместить в теги;
pErrors — список результатов операции для каждого тега.

Возвращаемая величина: код результата операции.

Комментарии:

Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Идентификатор доступа **dwAccessID** возвращает функция **OpenAccess().**
Однозначное соответствие передаваемых данных тегам задается массивом идентификаторов тегов **pdwItemsIDs.**
Все параметры должны быть инициализированы Пользователем.

HRESULT **GetServerStatus**([in] DWORD **dwConnectID**)

Назначение: Получить информацию о состоянии OPC-сервера

Параметры:

dwConnectID — идентификатор соединения;

Возвращаемая величина:

Статус OPC-сервера. Если вызов функции успешен, то OPC-сервер функционирует нормально. Иначе — данные OPC-сервера недоступны для клиента.

Комментарии:

Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Эта функция может использоваться для контроля за состоянием OPC-сервера при асинхронном опросе.

HRESULT **GetNextItemAttribute** ([in] DWORD **dwConnectID**,
[in] LPWSTR **wszItemName**,
[in,out] LPWSTR **wszAttributeName**,
[out] VARTYPE ***pvtAttributeType**,
[out] DWORD ***pdwAttributeID**)

Назначение: Получить описание атрибута тега из списка
В параметрах используются UNICODE-строки

Параметры:

dwConnectID — идентификатор соединения;
wszItemName — имя тега;
wszAttributeName — имя атрибута;
pvtAttributeType — тип данных атрибута;
pdwAttributeID — идентификатор атрибута.

Возвращаемая величина:

код результата операции.

Комментарии:

Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Параметр **wszAttributeName** используется как входной и выходной. Если в параметре **wszAttributeName** передается пустая строка, то в нем будет возвращен первый идентификатор атрибута в списке. Если параметр **wszAttributeName** — имя предыдущего полученного атрибута, то в нем будет возвращен следующий и т.д. Если в параметре **wszAttributeName** возвращена пустая строка, то достигнут конец списка.
Возвращаемый идентификатор атрибута **pdwAttributeID** используется для получения данных атрибута в функции **ReadItemAttributes()** .
Все параметры должны быть инициализированы Пользователем.

HRESULT **GetNextItemAttributeA** ([in] DWORD **dwConnectID**,
[in] LPSTR **szItemName**,
[in,out] LPSTR **szAttributeName**,
[out] VARTYPE ***pvtAttributeType**,
[out] DWORD ***pdwAttributeID**)

Назначение: Получить описание атрибута тега из списка
В параметрах используются ANSI-строки

Параметры:
dwConnectID — идентификатор соединения;
szItemName — имя тега;
szAttributeName — имя атрибута;
pvtAttributeType — тип данных атрибута;
pdwAttributeID — идентификатор атрибута.

Возвращаемая величина:
код результата операции.

Комментарии :
Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Параметр **szAttributeName** используется как входной и выходной. Если в параметре **szAttributeName** передается пустая строка, то в нем будет возвращен первый идентификатор атрибута в списке. Если параметр **szAttributeName** — имя предыдущего полученного атрибута, то в нем будет возвращен следующий и т.д. Если в параметре **szAttributeName** возвращена пустая строка, то достигнут конец списка.
Возвращаемый идентификатор атрибута **pdwAttributeID** используется для получения данных атрибута в функции **ReadItemAttributes()** .
Все параметры должны быть инициализированы Пользователем.

HRESULT **ReadItemAttributes** ([in] DWORD **dwConnectID**,
[in] LPWSTR **wszItemName**, [in] DWORD **dwCount**,
[in,size_is(dwCount)] DWORD ***pdwAttributesIDs**,
[out,size_is(dwCount)] VARIANT ***pvAttributesValues**,
[out,size_is(dwCount)] HRESULT ***pErrors**)

Назначение: Получить данные атрибутов тега
В параметрах используются UNICODE-строки

Параметры:
dwConnectID — идентификатор соединения;
wszItemName — имя тега;
dwCount — количество запрашиваемых атрибутов;
pdwAttributesIDs — массив идентификаторов атрибутов;
pvAttributesValues — массив значений атрибутов;
pErrors — массив результатов операции для каждого тега.

Возвращаемая величина:
код результата операции.

Комментарии:
Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Однозначное соответствие данных атрибутам задает массив идентификаторов атрибутов **pdwAttributesIDs**.
Все параметры должны быть инициализированы Пользователем.

HRESULT **ReadItemAttributesA** ([in] DWORD **dwConnectID**,
[in] LPSTR **szItemName**, [in] DWORD **dwCount**,
[in,size_is(dwCount)] DWORD ***pdwAttributesIDs**,
[out,size_is(dwCount)] VARIANT ***pvAttributesValues**,
[out,size_is(dwCount)] HRESULT ***pErrors**)

Назначение: Получить данные атрибутов тега
В параметрах используются ANSI-строки

Параметры:
dwConnectID — идентификатор соединения;
szItemName — имя тега;
dwCount — количество запрашиваемых атрибутов;
pdwAttributesIDs — массив идентификаторов атрибутов;
pvAttributesValues — массив значений атрибутов;
pErrors — массив результатов операции для каждого тега.

Возвращаемая величина:
код результата операции.

Комментарии:
Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Однозначное соответствие данных атрибутам задает массив идентификаторов атрибутов **pdwAttributesIDs**.
Все параметры должны быть инициализированы Пользователем.

HRESULT **UnadviseItems** ([in] DWORD **dwConnectID**, [in] DWORD **dwAccessID**)

Назначение: Отменить подписку на получение данных от OPC-сервера

Параметры:
dwConnectID — идентификатор соединения;
dwAccessID — идентификатор доступа.

Возвращаемая величина:
код результата операции.

Комментарии:
Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Идентификатор доступа **dwAccessID** возвращает функция **OpenAccess().**
После вызова этой функции «КРУГ OPC Toolkit» прекращает асинхронный опрос OPC-сервера.

HRESULT **CloseAccess** ([in] DWORD **dwConnectID**, [in] DWORD **dwAccessID**)

Назначение: Закрыть доступ к данным OPC-сервера

Параметры:
dwConnectID — идентификатор соединения;
dwAccessID — идентификатор доступа.

Возвращаемая величина:
код результата операции.

Комментарии:
Идентификатор соединения **dwConnectID** возвращает функция **Connect().**
Идентификатор доступа **dwAccessID** возвращает функция **OpenAccess().**
После вызова этой функции доступ к данным OPC-сервера будет невозможен.

HRESULT **Disconnect (**[in] DWORD **dwConnectID)**
Назначение: Осуществить отсоединение от ОРС-сервера **Параметры**: **dwConnectID** — идентификатор соединения; **Возвращаемая величина**: код результата операции. **Комментарии:** Идентификатор соединения **dwConnectID** возвращает функция **Connect().** После вызова этой функции ОРС-сервер становится недоступным для ОРС-клиента.

HRESULT **Uninitialize ()**
Назначение: Завершение работы с КРУГ ОРС Toolkit **Возвращаемая величина**: код результата операции.

4.5 Пример использования

4.5.1 Пример использования КРУГ ОРС Toolkit в среде программирования Visual C++

В качестве примера использования функций библиотеки в комплект поставки входит открытый код проекта демонстрационного ОРС-клиента в среде программирования Visual C++ 6.0.

Проект создан на основе библиотеки классов MFC.

Интерфейс клиента представляет собой диалоговое окно, на котором размещены элементы управления, каждый из которых связан с вызовом той или иной функции библиотеки ***KrugOPCToolkit.dll*** (рисунок 4.1).

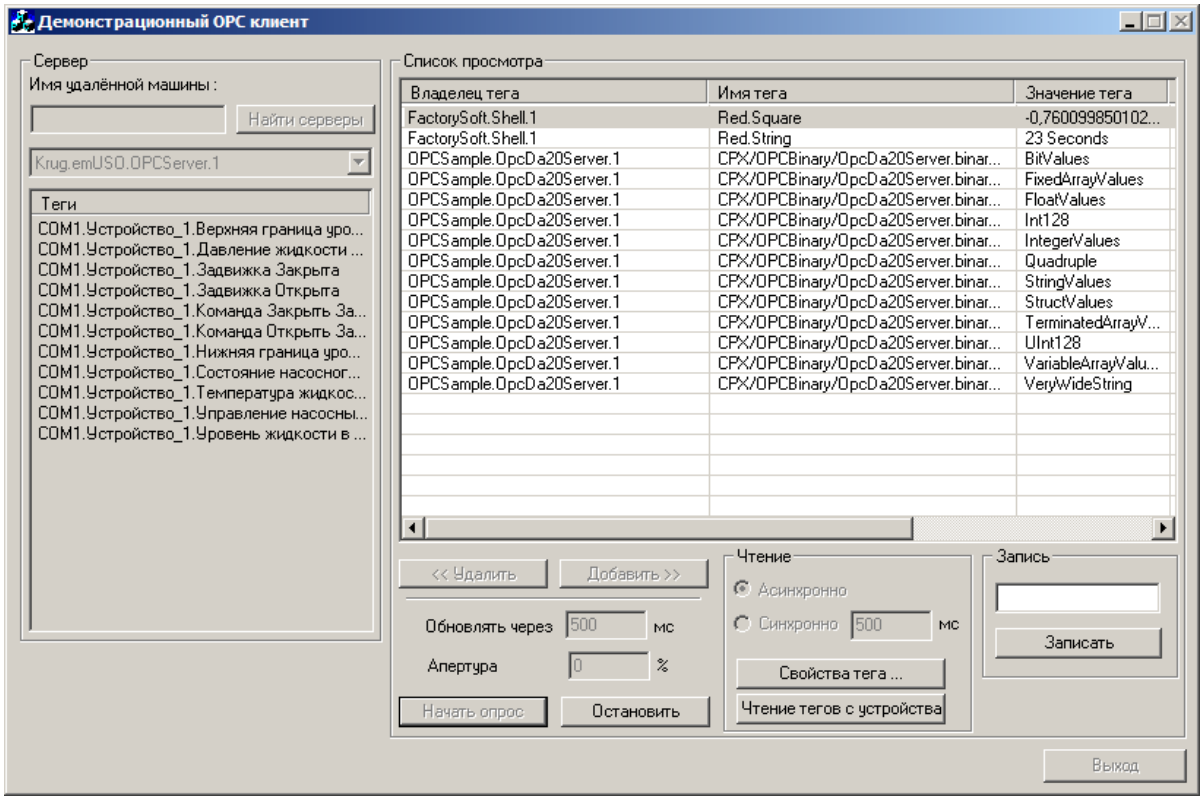


Рисунок 4.1 – Общий вид демонстрационного OPC-клиента

При старте демонстрационный OPC клиент загружает КРУГ OPC Toolkit и вызывает функцию ***Initialize().***

Краткое описание элементов управления:

- Кнопка «Найти серверы» и поле ввода «Имя удаленной машины». Эти два элемента управления служат для поиска установленных OPC-серверов на локальной или удаленной машине. При нажатии кнопки «Найти серверы» происходит циклический вызов функции ***GetNextServerName()***, и в нижележащий выпадающий список заносятся имена найденных OPC-серверов. Если поле ввода «Имя удаленной машины» пусто — будет произведен поиск OPC-серверов на локальной машине
- Выпадающий список с именами найденных OPC-серверов. При выборе одного из пунктов этого списка с указанным OPC-сервером устанавливается связь вызовом функции ***Connect()*** и запрашиваются имена его тегов циклическим вызовом функции ***GetNextServerItem()***. Полученные имена тегов заносятся в список «**Теги**»
- Кнопки «**Добавить**» и «**Удалить**». При их помощи можно сформировать список тегов для опроса. В список могут входить теги разных OPC-серверов
- Кнопка «**Свойства тега**» инициирует циклический вызов функции ***GetNextItemAttribute()***, затем данные атрибутов выбранного тега получаются путем вызова функции ***ReadItemAttributes()***. Полученная информация отображается в отдельном диалоговом окне «**Свойства тега**» (рисунок 4.2). При необходимости данные атрибутов тега можно обновить нажатием кнопки «**Обновить**», которая осуществляет повторный вызов функции ***ReadItemAttributes()***

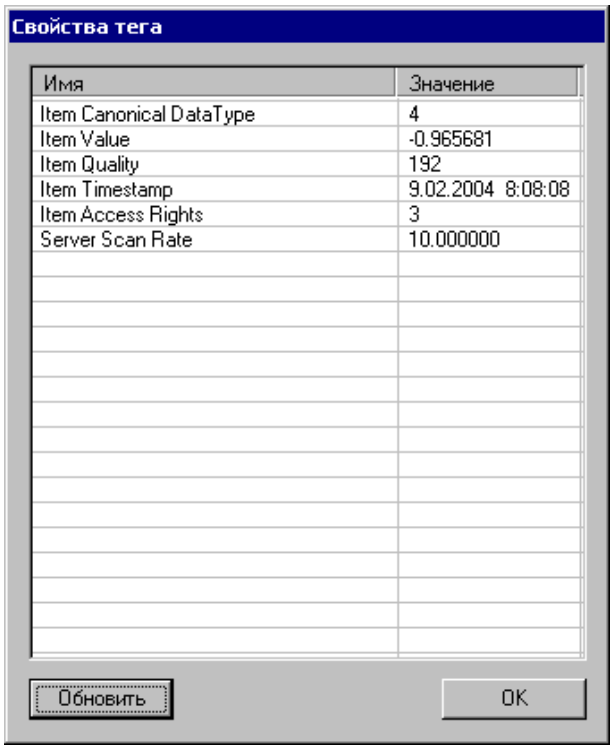


Рисунок 4.2 – диалоговое окно «Свойства тега»

- Поле ввода «**Обновлять через…**» задает запрашиваемую ОРС-клиентом частоту обновления данных в миллисекундах
- Кнопки «**Асинхронно**» и «**Синхронно**» (рисунок 4.1) задают тип опроса ОРС-серверов. Поле ввода рядом с кнопкой «Синхронно» задает период синхронного опроса в миллисекундах
- кнопка «**Начать опрос**» открывает доступ к данным ОРС-серверов вызовами функции ***OpenAccess()*,** и, в зависимости от выбранного режима опроса, инициирует асинхронный (вызов функции ***AdviseItems()***) или синхронный (вызов функции ***ReadItems()*** по таймеру) опрос ОРС-серверов. Для асинхронного опроса в клиенте определена функция обратного вызова ***OPCServersCallback()***, которая при каждом вызове обновляет данные ОРС-клиента и отображает изменения в списке опрашиваемых тегов. При асинхронном опросе периодически определяется статус ОРС-сервера вызовом функции ***GetServerStatus()*** .Если сервер становится недоступным, связь с ним прерывается
- Кнопка «**Чтение тегов с устройства**» обновляет значения тегов ОРС-сервера, запрашивая их с подключенного устройства. При нажатии на кнопку вызывается функция ***ReadItemsOption(OPC_FROM_DEVICE, …)*** для всех опрашиваемых тегов. Функцию с параметром ***OPC_FROM_DEVICE*** удобно применять для обновления значений тегов с ОРС-сервера по команде Пользователя
- Поле ввода «**Запись**» и кнопка «**Записать**» инициируют передачу значения, введенного в поле ввода «Запись» в выбранный тег ОРС-сервера. Это делается с помощью вызова функции ***WriteItems()***
- Кнопка «**Остановить**» прекращает доступ к данным ОРС-сервера с помощью вызова функции ***CloseAccess()***, перед этим вызвав функцию ***UnadviseItems()*,** если происходил асинхронный опрос;
- Кнопка «**Выход**» прекращает связь со всеми ОРС-серверами вызовами функций ***Disconnect()*** и завершает работу с КРУГ ОРС Toolkit вызовом функции ***Uninitialize()***.

Библиотека подключена к проекту универсальным способом (через вызов функций WinAPI ***LoadLibrary()*** и ***GetProcAddress()***), при этом использовалось описание функций, данное в файле ***KrugOPCToolkit_Exports.h*** , подключенном к проекту.

Весь механизм OPC-клиента реализован в трех классах:
класс ***CKrugOPCDemoClientDlg*** скрывает в себе реализацию диалогового окна OPC-клиента;
класс ***COPCServer*** реализует связь между OPC-клиентом и «КРУГ OPC Toolkit»;
класс ***CTagPropertiesDlg*** реализует функции диалогового окна «Свойства тега».

Для более подробного изучения работы демонстрационного клиента смотрите комментарии в коде программы.

4.5.2 Пример использования КРУГ OPC Toolkit в среде программирования Delphi

В качестве примера использования функций библиотеки в комплект поставки входит открытый код проекта демонстрационного OPC-клиента в среде программирования Delphi 7.0.

Интерфейс клиента представляет собой диалоговое окно, на котором размещены элементы управления, каждый из которых связан с вызовом той или иной функции библиотеки ***KrugOPCToolkit.dll*** (рисунок 4.3).

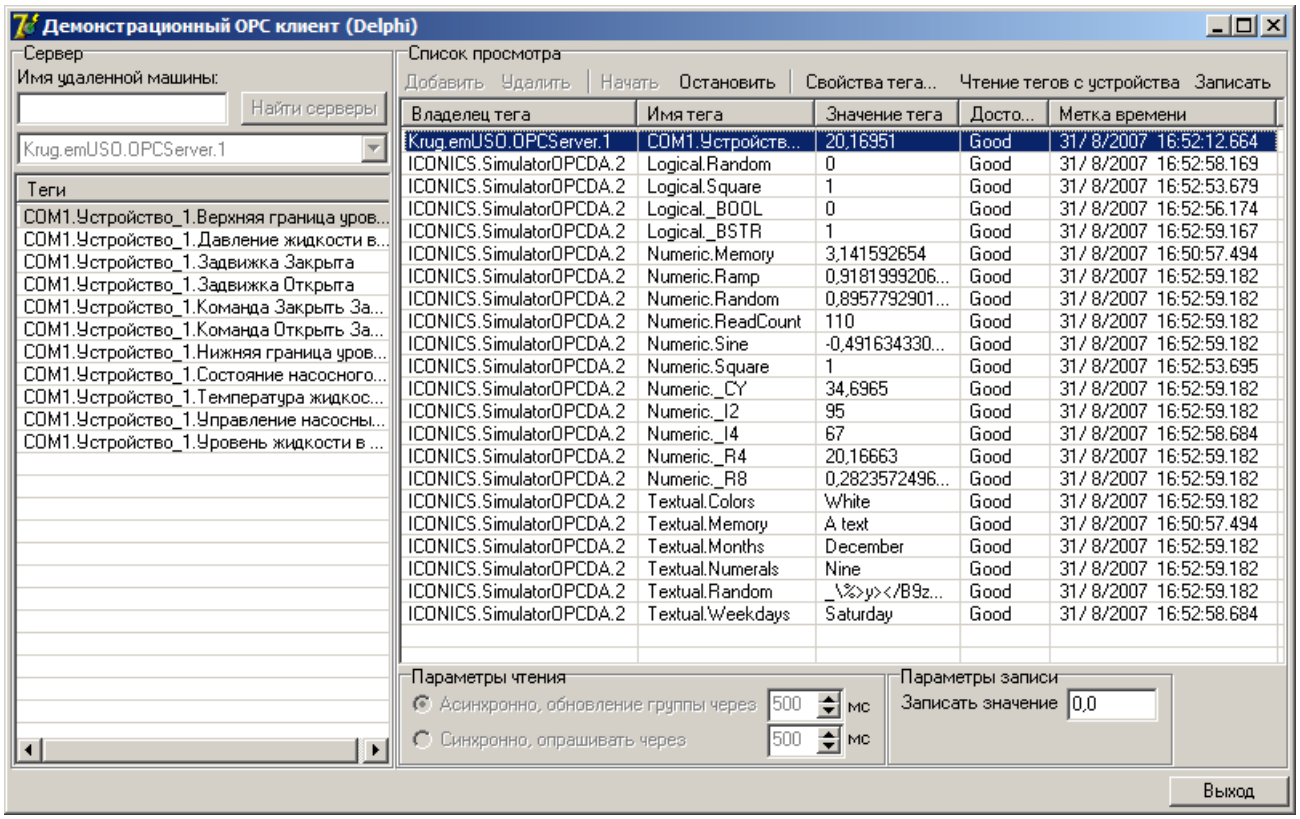


Рисунок 4.3 – Общий вид демонстрационного OPC-клиента

Описание элементов управления диалогового окна демонстрационного OPC-клиента аналогично описанию в разделе 4.5.1. Окно «**Свойства тега**» приведено на рисунке 4.4.

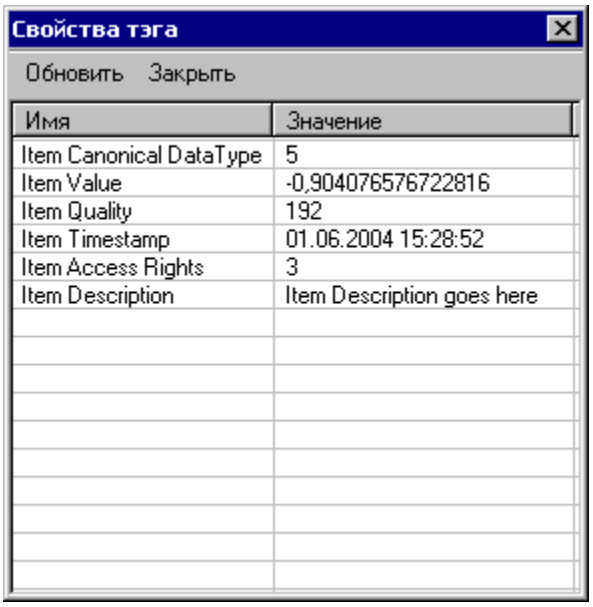


Рисунок 4.4 – Диалоговое окно «Свойства тега»

Библиотека подключена к проекту универсальным способом (через вызов функций WinAPI ***LoadLibrary()*** и ***GetProcAddress()***), при этом использовалось описание функций, данное в файле ***KrugOPCToolkit_Exports.h.***

Для более подробного изучения работы демонстрационного клиента смотрите комментарии в коде программы.

4.6 Регистрация

В стандартную поставку входит только ознакомительная версия «КРУГ OPC Toolkit», имеющая следующие ограничения для одного запуска:

- Максимальное количество произведенных соединений (вызовов функции *Connect()*) – 50. Все последующие соединения установлены не будут
- Максимальное время связи с OPC-серверами – 20 минут. По истечении этого времени связь со всеми OPC-серверами будет прекращена
- Максимальное число опрашиваемых OPC-серверов – 2. Доступ к данным большего количества OPC-серверов одновременно буден невозможен
- Максимальное количество одновременно опрашиваемых тегов – 60. Будут опрашиваться только первые 60 из общего числа выбранных тегов.

Для того, чтобы снять ограничения ознакомительной версии, необходимо использовать программу для регистрации «КРУГ OPC Toolkit», которая входит в комплект поставки (***KrugOPCToolkitRegistration.exe***). В предлагаемом программой диалоговом окне предлагается ввести имя Пользователя, имя компании и регистрационный ключ (рисунок 4.5).

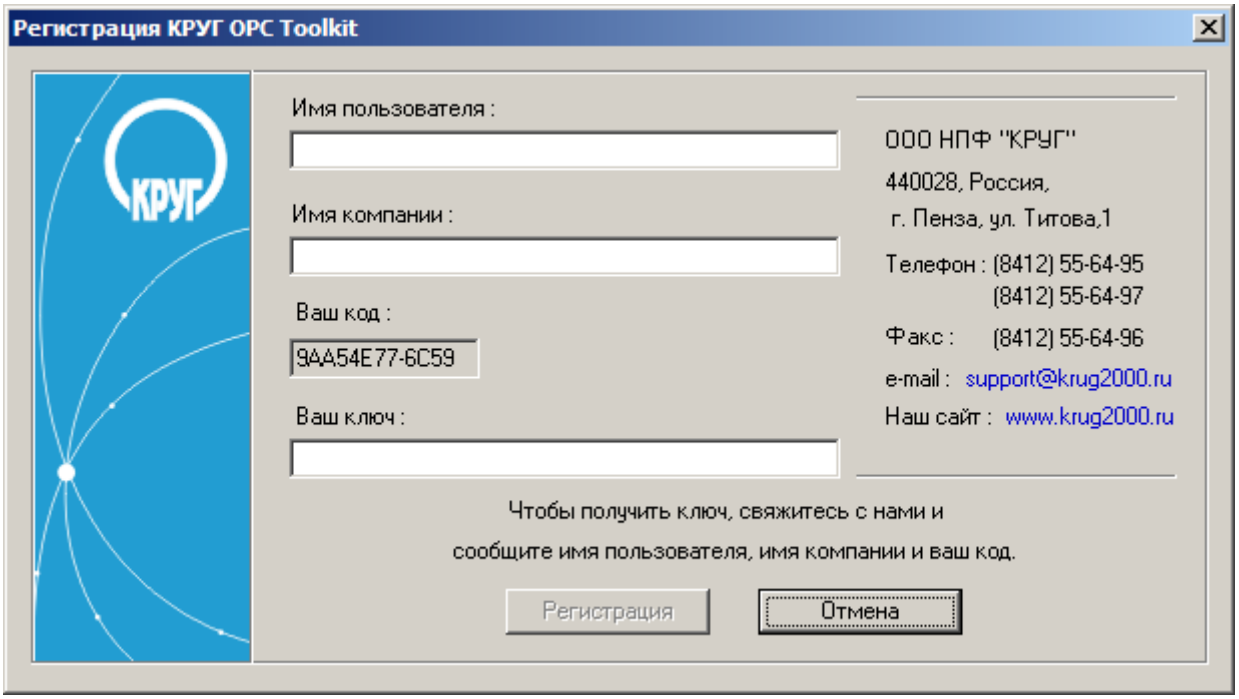


Рисунок 4.5 – Диалоговое окно регистрации «КРУГ OPC Toolkit»

Чтобы получить значение регистрационного ключа, необходимо связаться с разработчиком «КРУГ OPC Toolkit» НПФ «КРУГ»и передать любым удобным путем (по телефону, по факсу, по электронной почте) данные, которые введены для регистрации:

- имя Пользователя
- имя компании
- ваш код.

После процедуры регистрации в НПФ «КРУГ» Вам будет сообщен ключ для полнофункционального использования «КРУГ OPC Toolkit».