



CODESYS V3.5

Настройка обмена с верхним уровнем



Руководство пользователя

24.10.2025

версия 3.1

Оглавление

Оглавление	3
1 Цель документа	4
2 Основные сведения о технологии OPC	5
3 Настройка обмена через символьную конфигурацию	7
3.1 Настройка контроллера	7
3.2 Настройка CODESYS OPC Server V3	11
3.3 Особенности настройки OPC UA-сервера	15
3.4 Подключение к облачному сервису OwenCloud	17
4 Настройка обмена по протоколу Modbus	24
4.1 Настройка контроллера	24
4.2 Настройка MasterOPC Universal Modbus Server	29
4.3 Настройка Owen OPC Server	35
5 Подключение OPC-сервера к MasterSCADA 3.x.....	40
5.1 Подключение OPC DA-сервера.....	40
5.2 Подключение OPC UA-сервера.....	43

1 Цель документа

Настоящее руководство описывает настройку обмена данными с верхним уровнем АСУ (SCADA-системами и другим ПО) для контроллеров ОВЕН с использованием технологии [OPC](#). Руководство предназначено для пользователей с базовыми навыками работы в **CODESYS V3.5**, поэтому общие вопросы (например, создание и загрузка проектов) в данном документе не рассматриваются. Базовая информация приведена в руководствах **CODESYS V3.5. Первый старт** и **CODESYS V3.5. FAQ**, которые доступны на сайте [ОВЕН](#) в разделе **CODESYS V3/Документация**.

В документе рассматриваются вопросы подключения контроллеров ОВЕН, программируемых в **CODESYS V3.5**, к SCADA-системе [MasterSCADA 3.x](#) с использованием различных OPC-серверов:

- [CODESYS OPC Server V3](#) (протокол обмена – символьный протокол CODESYS);
- [MasterOPC Universal Modbus Server](#) (протокол обмена – **Modbus**);
- [Owen OPC Server](#) (протокол обмена – **Modbus**);
- встроенный в ПЛК [OPC UA Server](#) (протокол обмена – **OPC UA**).

Кроме того, рассматривается подключение контроллера к облачному сервису [OwenCloud](#).

2 Основные сведения о технологии OPC

Первая версия стандарта **OPC** была опубликована консорциумом OPC Foundation в 1996 году. Целью стандарта являлось создание унифицированного интерфейса для подключения устройств автоматизации к SCADA-системам. В то время в отрасли было относительно немного открытых промышленных протоколов, из-за чего большинство компаний разрабатывали собственные решения. Это, в свою очередь, затрудняло процесс интеграции приборов в SCADA-системы: разработчикам SCADA приходилось либо создавать и поддерживать множество коммуникационных драйверов, либо производители приборов были вынуждены разрабатывать драйвер для каждой SCADA, к которой предполагалось подключать их устройства.

Стандарт OPC основан на технологии **OLE** (*Object Linking and Embedding*), разработанной компанией Microsoft для ОС Windows. Аббревиатура «OPC» означает OLE for Process Control (OLE для управления процессами). В стандарте описывается интерфейс обмена данными между OPC-клиентом (SCADA-системой) и OPC-сервером. OPC-сервер – это специализированное программное обеспечение, установленное на ПК, которое опрашивает подключенные устройства по промышленным протоколам и предоставляет SCADA-системе доступ к данным этих устройств. Таким образом, производителям оборудования достаточно однократно разработать свой OPC-сервер, чтобы обеспечить возможность подключения оборудования к любой SCADA-системе, поддерживающей технологию OPC. Сейчас эту технологию поддерживает практически любая SCADA-система.

Стандарт OPC оказал существенное влияние на рынок промышленной автоматизации. Но с развитием технологий стали проявляться некоторые его недостатки:

- привязка к технологиям Microsoft (OLE, DCOM и т.д.) сделала фактически невозможным использование OPC на других ОС. Увеличение аппаратных характеристик ПЛК привело к желанию запускать OPC-серверы прямо на них – но поскольку значительная часть контроллеров использует ОС на базе Linux, то это желание было неосуществимо;
- сложность настройки связи OPC-сервера с OPC-клиентом, который запущен на другом ПК. Такой вариант подключения требует настройки службы DCOM, что в ряде случаев является довольно сложной задачей;
- отсутствие средств информационной безопасности. В период создания стандарта OPC большинство систем автоматизации были локальными, и аспекты, связанные с удаленным доступом и обеспечением его защиты, практически не рассматривались.

Недостатки классической технологии OPC привели к необходимости разработки нового стандарта. Он получил название **OPC UA** (OPC Unified Architecture). Первая версия нового стандарта была представлена в 2006 году, и с тех пор он постоянно развивается и дополняется.

Ключевыми особенностями нового стандарта являются:

- **кроссплатформенность** – OPC UA не использует проприетарных технологий, поэтому клиент и сервер могут быть запущены на устройствах с любыми ОС. В связи с этим аббревиатура OPC с введением нового стандарта стала расшифровываться как «*Open Platform Communications*»;
- **безопасность** – подключение к серверу может быть защищено логином/паролем и требовать использования сертификатов;
- **удаленный доступ** – сервер и клиент могут располагаться в разных сетях и быть связаны через Интернет с использованием VPN и т. д.;
- **функциональность** – в рамках стандарта описан набор информационных моделей для работы с данными – доступ к оперативным данным, чтение архивов, передача тревог и событий и т. д. Большинство этих моделей были разработаны еще для «классической»

технологии OPC, но в рамках OPC UA для всех них используется единообразный механизм адресации и доступа к данным;

- **удобство настройки** – OPC UA-клиент при подключении к серверу считывает информацию о доступных параметрах и предоставляет ее пользователю. Соответственно, программисту не требуется добавлять и настраивать каждый параметр отдельно, а только отметить параметры, которые нужно использовать;
- принципиальным преимуществом нового стандарта по сравнению с классическим OPC является снятие с OPC-сервера роли шлюза между устройствами автоматизации, использующими промышленные протоколы, и SCADA-системами. Фактически OPC UA сам представляет собой промышленный протокол, который применяется для обмена данными на среднем (контроллеры, панели оператора, модули ввода-вывода и т. д.) и верхнем (SCADA, облачные сервисы и т. д.) уровнях системы автоматизации.

Контроллеры ОВЕН поддерживают следующие варианты настройки обмена с использованием технологии OPC:

1. использование **CODESYS OPC Server V3**, который входит в дистрибутив CODESYS. Преимущество этого варианта – простота настройки обмена в проекте;
2. использование OPC-сервера с поддержкой протокола **Modbus** (например, **Modbus Universal Master OPC Server** от компании [МПС софт](#)). Этот вариант является наиболее сложным в настройке (в частности, из-за необходимости написания кода конвертации типов данных в программе ПЛК); обычно он используется, когда такой OPC-сервер уже входит в состав системы автоматизации;
3. использование протокола **OPC UA**. Этот вариант является таким же простым в настройке, как и вариант 1, но значительно более функциональным (например, поддерживается передача тревог и защищенное соединение).

3 Настройка обмена через символьную конфигурацию

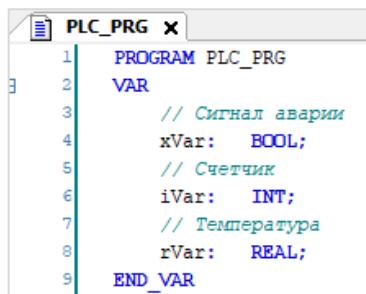
3.1 Настройка контроллера

Символьная конфигурация позволяет настроить обмен с:

- [CODESYS OPC Server V3](#);
- [OPC UA-клиентом](#);
- [облачным сервисом OwenCloud](#);
- устройством, поддерживающим символьный протокол CODESYS (например, его поддерживают панели оператора Weintek).

Ниже приведена инструкция по подготовке проекта с символьной конфигурацией, который будет использоваться в примерах:

1. Следует создать новый проект в **CODESYS V3.5** (*язык программы не имеет значения*).
2. В программе **PLC_PRG** объявить следующие переменные:



```

PROGRAM PLC_PRG
VAR
    // Сигнал аварии
    xVar: BOOL;
    // Счетчик
    iVar: INT;
    // Температура
    rVar: REAL;
END_VAR

```

Рисунок 3.1.1 – Объявление переменных в программе PLC_PRG

3. Добавить в проект компонент **Символьная конфигурация**:

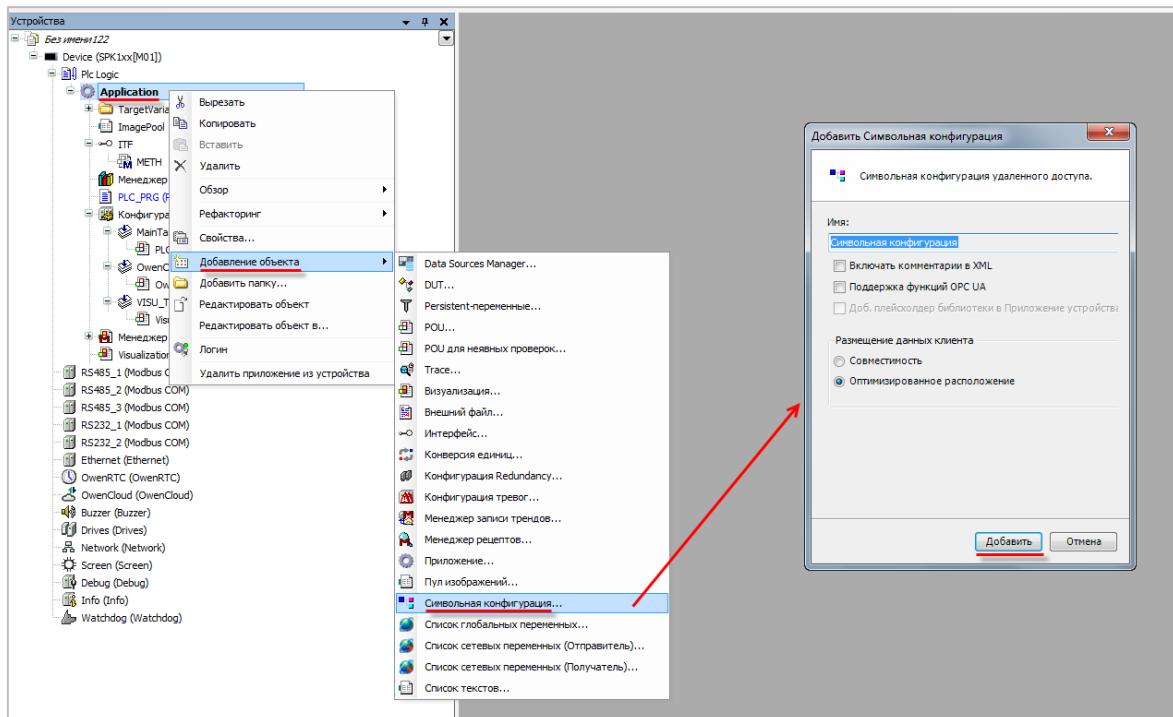


Рисунок 3.1.2 – Добавление компонента Символьная конфигурация

При добавлении компонента пользователь может выбрать следующие настройки:

- Включить комментарии в XML** – если установлена галочка, то в файл символьной конфигурации будут включены комментарии к переменным;
- Поддержка функций OPC UA** – если установлена галочка, то в файл символьной конфигурации добавляется дополнительная информация, необходимая для поддержки функций **OPC UA сервера**. OPC UA сервер поддерживается в следующих контроллерах ОВЕН: **СПК1xx [M01]** (начиная с прошивки **1.1.0611.1056**), **ПЛК2xx**. См. также [п. 3.3](#);
- Размещение данных клиента** – пользователь может выбрать структуру файла символьной конфигурации – совместимую со старыми версиями CODESYS или оптимизированную. Оптимизированная структура поддерживается начиная с **CODESYS V3.5 SP7**. Более подробная информация о различиях в размещении описана в [справке CODESYS](#).

- После добавления компонента **Символьная конфигурация** следует выполнить компиляцию проекта:

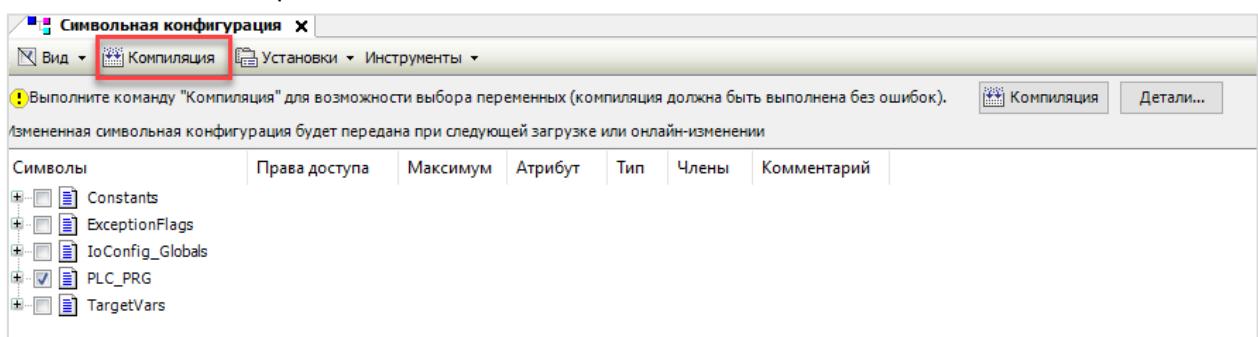


Рисунок 3.1.3 – Кнопка компиляции проекта после создания символьной конфигурации

В случае добавления в проект новых переменных для внесения изменений в символьную конфигурацию требуется предварительно выполнить повторную компиляцию проекта.

Компонент имеет следующие настройки:

Таблица 2.1 – Настройки компонента Символьная конфигурация

Настройка	Описание	Рекомендуемое значение
Вкладка Вид (настройки фильтрации отображаемых переменных)		
Не конфигурируется из проекта	В случае выбора фильтра – в списке будут отображаться переменные проекта, доступные для добавления в символьную конфигурацию	
Не конфигурируется из библиотеки	В случае выбора фильтра – в списке будут отображаться переменные библиотек, доступные для добавления в символьную конфигурацию	
Символы, экспортируемые атрибутами	В случае выбора фильтра – в списке будут отображаться переменные проекта с атрибутом attribute 'symbol'. См. подробнее в справке CODESYS	

Вкладка Установки		
Поддержка функций OPC UA	Если установлена галочка, то в файл символьной конфигурации добавляется дополнительная информация, необходимая для поддержки функций OPC UA сервера . OPC UA сервер поддерживается в следующих контроллерах ОВЕН: СПК1xx [М01] (начиная с прошивки 1.1.0611.1056), ПЛК2xx . См. также п. 3.3 ;	Включено
Включить комментарии в XML	Если установлена галочка, то в XML-файл символьной конфигурации будут включены комментарии к переменным	Включено
Включить флаги узлов в XML	Флаги узлов пространств имен предоставляют дополнительную информацию о расположении узлов. Флаги узлов всегда экспортируются в символьную конфигурацию при включенной поддержке функций OPC UA. Однако можно отключить их экспорт в XML-файл символьной конфигурации, так как у некоторых недоработанных парсеров могут возникнуть ошибки при их разборе	Отключено
Задать комментарии и атрибуты	Команда позволяет детально настроить комментарии и атрибуты, которые будут экспортированы в XML-файл символьной конфигурации. См более подробное описание в справке CODESYS	Все галочки включены
Настроить синхронизацию с МЭК-задачами	См. описание в справке CODESYS	Не настраивать
Расположение	Пользователь может выбрать структуру файла символьной конфигурации – совместимую со старыми версиями или оптимизированную. Оптимизированная структура поддерживается начиная с CODESYS V3.5 SP7 . Более подробная информация о различиях в размещении описана в справке CODESYS .	Оптимизированное расположение
Использовать пустые доп. имена по умолчанию (V2-совместимость)	Опция позволяет создать символьную конфигурацию, совместимую с OPC-сервером из дистрибутива CoDeSys V2.3	Отключено
Включить прямой доступ к I/O	Опция позволяет получить доступ к переменным символьной конфигурации по АТ-адресам. Эта возможность является потенциально опасной и не должно использоваться на этапе эксплуатации (только на этапе наладки)	Отключено
Вызовы в функции, ФБ, методы и программы	Если установлена галочка, то OPC UA-клиент может осуществлять вызов функций, ФБ, методов и программ контроллера, работающего в режиме OPC UA-сервера	Включено
Включить информацию вызов в XML	Если установлена галочка, то в XML-файл символьной конфигурации будет включена информация, необходимая для вызова функций, ФБ, методов и программ	Отключено
Включить наборы символов	Опция позволяет создавать в символьной конфигурации различные наборы символов. Таким образом, разные клиенты символьной конфигурации будут иметь доступ только к определенным переменным контроллера. В	В зависимости от того, требуется ли защитить доступ по OPC UA с

	частности, это позволяет задать логин и пароль для доступа к контроллеру по OPC UA	помощью логина и пароля
Вкладка Инструменты		
Сохранить XML-файл	Команда позволяет сохранить схему (.xsd) символьной конфигурации для импорта в другое ПО	

**ПРИМЕЧАНИЕ**

XML-файл формируется в директории проекта при выполнении команд **Компиляция** или **Генерация кода**. В рамках примеров документа он не требуется (но, например, может требоваться для настройки обмена между контроллером и другим устройством, поддерживающим символьный протокол CODESYS).

- Пометить галочками переменные, которые будут считываться/изменяться клиентом символьной конфигурации (OPC DA-сервером, OPC UA-клиентом, облачным сервисом OwenCloud) и указать для каждой из них права доступа.

Для прав доступа используются следующие пиктограммы:

- только чтение;
- только запись;
- чтение/запись.

**ПРИМЕЧАНИЕ**

Кроме ручного выбора в списке можно добавить переменные в символьную конфигурацию с помощью атрибута attribute 'symbol'. См. подробнее в [справке CODESYS](#).

**ПРИМЕЧАНИЕ**

В случае подключения к [OwenCloud](#) параметры с типом доступа **Только чтение** добавляются в группу опроса **Оперативные**, параметры с типом доступа **Чтение и запись** – в группы **Конфигурационные** и **Управляемые**.

Символы	Права доступа	Максимум	Атрибут	Тип	Члены	Комментарий
Constants						
ExceptionFlags						
IoConfig_Globals						
PLC_PRG	<input checked="" type="checkbox"/>			INT		Счетчик
iVar	<input checked="" type="checkbox"/>			REAL		Температура
rVar	<input checked="" type="checkbox"/>			BOOL		Сигнал аварии
xVar	<input checked="" type="checkbox"/>					
TargetVars						

Рисунок 3.1.4 – Выбор переменных в компоненте Символьная конфигурация

Настройка проекта завершена (на предупреждение «Задано 3 переменных, которые не используются в МЭК-коде» не следует обращать внимание – в рамках примера эти переменные действительно не используются в программе контроллера). Загрузите проект в контроллер.

3.2 Настройка CODESYS OPC Server V3

До релиза версии **CODESYS V3.5 SP17** OPC-сервер **CODESYS OPC Server V3** входил в дистрибутив 32-битных версий CODESYS. Начиная с версии **CODESYS V3.5 SP17** он исключен из дистрибутива, но может быть загружен отдельно из [CODESYS Store](#) или по [прямой ссылке](#).

До релиза версии **CODESYS V3.5 SP12** OPC-сервер **CODESYS OPC Server V3** распространялся бесплатно. Начиная с версии **CODESYS V3.5 SP12** использование OPC-сервера требует приобретения лицензии. Тем не менее, сборки OPC-сервера из ранних версий CODESYS могут использоваться и при работе со свежими версиями CODESYS. Пакет таргет-файлов ОВЕН включает в себя скрипт, который заменяет конфигурационные файлы OPC-сервера на эти же файлы из старых версий среды.

Таким образом:

- если вы используете версию **CODESYS < V3.5 SP17**, то после установки пакета таргет-файлов ОВЕН вы сможете работать с **CODESYS OPC Server V3** без активации лицензии;
- если вы используете версию **CODESYS V3.5 SP17**, то вам потребуется установить **CODESYS OPC Server V3** отдельно ([ссылка](#)) и убедиться, что в директории установки вашей версии CODESYS появилась директория **CODESYS OPC Server 3** (если она не появилась – *то следует скопировать ее туда самостоятельно из директории, в которую она была установлена*). После установки пакета таргет-файлов ОВЕН вы сможете работать с **CODESYS OPC Server V3** без активации лицензии.

Для настройки OPC-сервера **CODESYS OPC Server V3** следует:

1. Запустить приложение **OPC Configurator** (из меню **Пуск** или папки **CODESYS OPC Server 3**, расположенной в директории установки **CODESYS**).

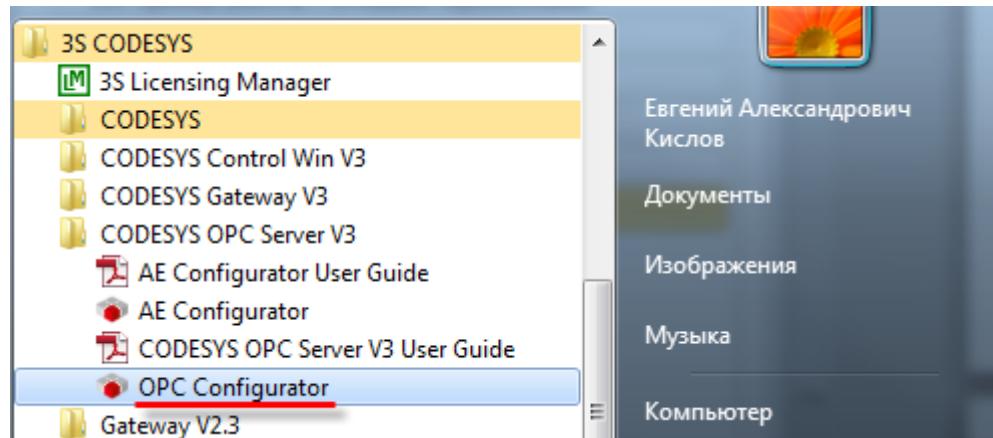


Рисунок 3.2.1 – Запуск приложения OPC Configurator

2. Нажать ПКМ на узел **Server** и в контекстном меню выбрать команду **Append PLC**:

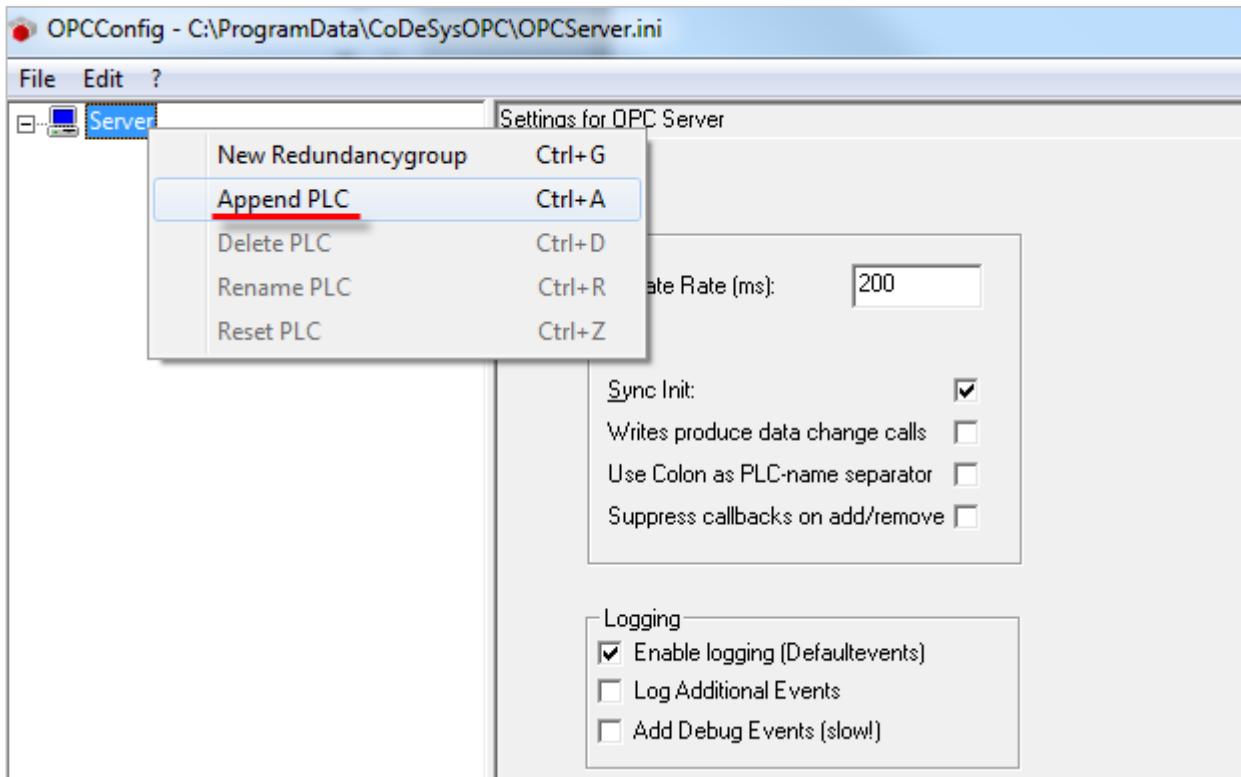


Рисунок 3.2.2 – Добавление контроллера в OPC-сервер

3. На вкладке **PLC1** указать интерфейс, по которому будут связаны контроллер и OPC-сервер – **GATEWAY3 (Ethernet)**.

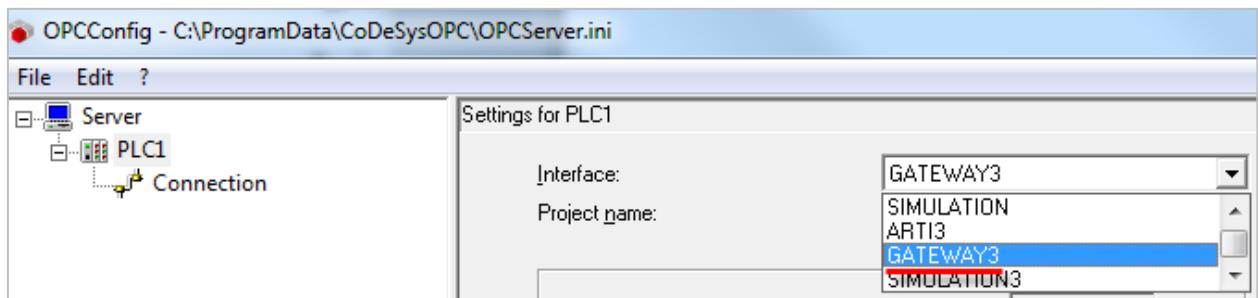


Рисунок 3.2.3 – Выбор интерфейса связи контроллера и OPC-сервера

4. На вкладке **Connection** нажать кнопку **Edit** и указать IP-адрес контроллера.

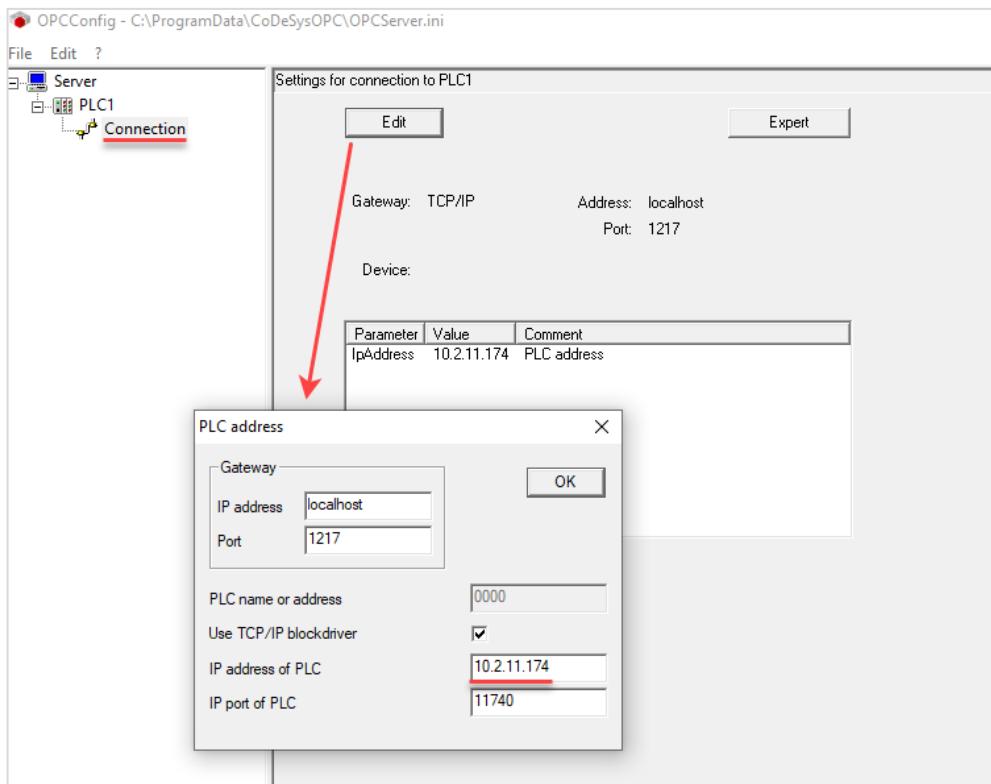


Рисунок 3.2.4 – Указание IP-адреса контроллера

5. Сохранить настройки OPC-сервера:

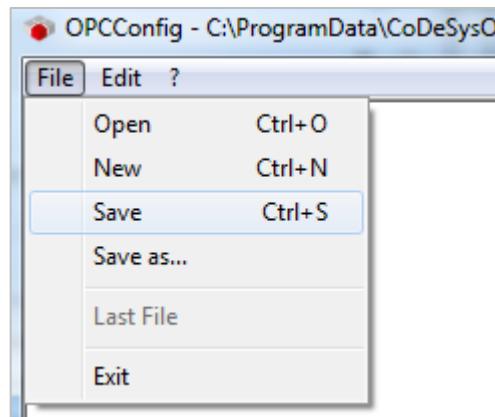


Рисунок 3.2.5 – Сохранение настроек OPC-сервера

Настройка OPC-сервера завершена. Приложение **OPC Configurator** можно закрыть.

Теперь можно переходить к подключению OPC-сервера к SCADA-системе – см. [п. 5.1](#).

**ПРИМЕЧАНИЕ**

При добавлении OPC-сервера **CODESYS OPC Server V3** в SCADA-систему может возникнуть следующая ошибка:

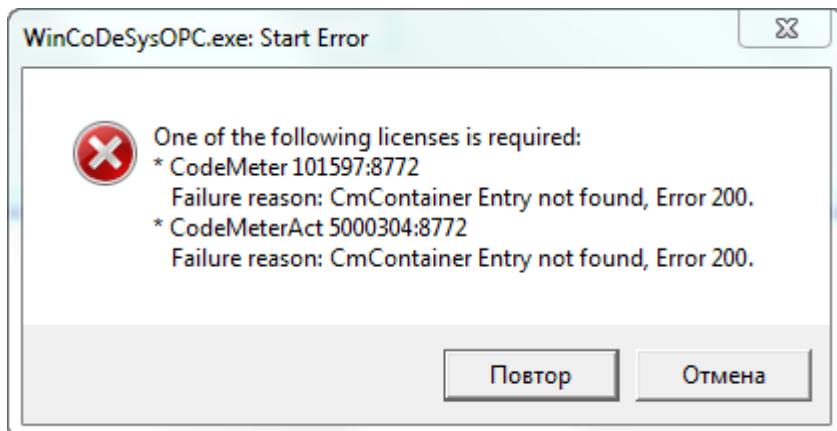


Рисунок 3.2.6 – Ошибка при добавлении OPC-сервера CODESYS OPC Server V3 в SCADA-систему

Это означает, что используемая версия OPC-сервера требует лицензирования. См. информацию в начале пункта.

**ПРИМЕЧАНИЕ**

CODESYS OPC Server V3 использует для подключения к контроллеру символьный протокол CODESYS. Максимальное количество одновременных подключений по этому протоколу – 4, при этом онлайн-подключение к контроллеру из среды CODESYS занимает одно подключение. Для контроллеров ПЛК210-1х и СПК210 количество доступных подключений можно увеличить путем редактирования конфигурационного файла:

1. Подключитесь к контроллеру с помощью утилиты [WinSCP](#) (протокол – **SCP**, пароль – совпадает с паролем web-конфигуратора).
2. Перейдите в рабочую директорию (**ПЛК210: /root/CODESYS, СПК210: /home/root/CODESYS_WRK**).
3. Откройте файл **CODESYSControl_User.cfg**.
4. Найдите в файле секцию **CmpChannelServer** и отредактируйте значения её параметров:
 - **MaxChannels** определяет максимальное количество подключений по символьному протоколу;
 - **BufferSize** определяет общий размер буфера (в байтах), используемого всеми подключениями. Рекомендуется рассчитывать его значение по формуле $\text{MaxChannels} \cdot 1000000$.
5. Сохраните файл.
6. Перезагрузите контроллер, чтобы внесенные изменения вступили в силу.

3.3 Особенности настройки OPC UA-сервера

Для использования OPC UA-сервера достаточно при добавлении [символьной конфигурации](#) установить галочку **Поддержка функций OPC UA**. Других настроек в общем случае не требуется. После [п. 3.1](#) можно сразу переходить к подключению OPC UA-сервера к SCADA-системе – см. [п. 5.2](#).

Дополнительная информация и ссылки:

- порт OPC UA-сервера по умолчанию – **4840** (может быть изменен, см информацию ниже);
- поддерживается аутентификация через логин/пароль и сертификат безопасности. См. [видеопример](#) и [информацию по ссылке](#);
- начиная с версии **V3.5 SP17** в полном объеме (начиная с **V3.5 SP11** – с ограничениями) поддерживается профиль **OPC UA Alarm&Conditions** для передачи тревог из компонента **Конфигурация тревог** в OPC UA-клиент. Единственное, что для этого требуется – добавить в проект библиотеку **CmpOPCUAProviderAlarmConfiguration**. См. [видеопример](#) (начиная с 16:35) и [информацию по ссылке](#);
- начиная с версии **V3.5 SP17** поддерживается профиль **OPC UA Methods** для вызова функций, ФБ, методов и программ контроллера со стороны OPC UA-клиента. См. [видеопример](#) (начиная с 12:38);
- профиль **OPC UA Historical Access** в данный момент не поддерживается;
- вопросы производительности OPC UA-сервера рассмотрены в п. 2.1.2.1 и 2.1.2.2 [CODESYS Public FAQ.pdf](#).



ВНИМАНИЕ

В текущей версии системы исполнения CODESYS, используемой в контроллерах ОВЕН (**3.5.17.30**), присутствуют ошибки, связанные с работой CODESYS OPC UA-сервера:

1. Когда OPC UA-клиент закрывает сессию с удалением подписок (с помощью запроса **CloseSessionRequest** с параметром **DeleteSubscriptions = TRUE**), то CODESYS OPC UA-сервер не удаляет подписки.

2. CODESYS OPC UA-сервер не обрабатывает параметр **requestedLifetimeCount** (время, через которое подписка должна быть удалена в случае отсутствия запросов от OPC UA-клиента).

Таким образом, если OPC UA-клиент регулярно разрывает и восстанавливает соединение с CODESYS OPC UA-сервером (например, из-за проблем с каналом связи или перезагрузки OPC UA-клиента), то начинается «утечка» подписок. Текущее число подписок CODESYS OPC UA-сервера можно отследить по системному тегу **Server – ServerDiagnostics – ServerDiagnosticsSummury – CurrentSubscriptionCount**. При достижении некоторого порогового значения (в рамках тестов оно было приблизительно равно **580**), CODESYS OPC UA-сервер перестает передавать в рамках подписок актуальные данные (отдаются ранее «застывшие» значения).

Эта проблема не проявляется, если вместо подписки (**Subscription service**) использовать циклический опрос (**Read service**).

Если в рамках вашей системы нет возможности использовать циклический вопрос и предполагается вероятность регулярного разрыва и восстановления связи между OPC UA-клиентом и CODESYS OPC UA-сервером, то настоятельно рекомендуется отказаться от использования CODESYS OPC UA-сервера и реализовать обмен с контроллером по другому протоколу (например, Modbus TCP).



ПРИМЕЧАНИЕ

Для контроллеров ПЛК210-1x и СПК210 значения конфигурационных параметров OPC UA сервера можно изменить путем редактирования конфигурационного файла:

1. Подключитесь к контроллеру с помощью утилиты [WinSCP](#) (протокол – **SCP**, пароль – совпадает с паролем web-конфигуратора).

2. Перейдите в рабочую директорию (**ПЛК210: /root/CODESYS, СПК210: /home/root/CODESYS_WRK**).

3. Откройте файл **CODESYSControl_User.cfg**.
4. Найдите в файле секции **CmpOPCUA** и **CmpOPCUAProviderIecVarAccess**, и добавьте нужные параметры/отредактируйте значения существующих:
 - **NetworkPort** – номер сетевого порта, используемого OPC UA сервером;
 - **MaxNodesPerRead** – максимальное количество объектов в одном запросе чтения;
 - **SessionsMax** – максимальное количество одновременных подключений к серверу;
 - **ItemMinSamplingRate** – минимальный период обновления для запрашиваемых клиентом данных;
 - **CustomNodeName** – имя контроллера, которое будет отображаться в OPC UA клиенте. Для изменения имени следует удалить в начале строки символ комментария (;);
 - остальные доступные для добавления параметры описаны в [этой статье](#).
5. Сохраните файл.
6. Перезагрузите контроллер, чтобы внесенные изменения вступили в силу.

3.4 Подключение к облачному сервису OwenCloud

Облачный сервис OwenCloud не имеет никакого отношения к технологии ОРС, но так как его настройка тоже выполняется через [символьную конфигурацию](#) – то разумно рассмотреть пример подключения к нему контроллера в рамках данного документа.

Для подключения контроллеров ОВЕН, программируемых в **CODESYS V3.5**, к сервису **OwenCloud** не требуется использования сетевых шлюзов линейки [Пx210](#). Доступ к облачному сервису осуществляется через подключение контроллера к локальной сети с доступом в Интернет.



ПРИМЕЧАНИЕ

Для контроллеров **СПК1xx [M01]** подключение к **OwenCloud** через символьную конфигурацию поддерживается начиная с прошивки **1.1.0611.1056**. В более ранних версиях использовалось подключение через Modbus TCP – этот способ описан в предыдущих версиях документа и не поддерживается в актуальных прошивках. Для контроллеров **ПЛК2xx** подключение к **OwenCloud** поддерживается только через символьную конфигурацию.

Для подключения к **OwenCloud** следует:

1. Проверить сетевые настройки контроллера. В web-конфигураторе на вкладке **Сеть/Интерфейсы** для интерфейса, через который осуществляется подключение к OwenCloud, должен быть задан IP-адрес шлюза и DNS-сервера (например, [Google Public DNS](#)). Узнать адрес шлюза и локального DNS-сервера можно у сетевого администратора.

Интерфейсы - LAN

На этой странице вы можете настроить сетевые интерфейсы. Вы можете объединить несколько интерфейсов в мост, выбрав опцию 'Объединить мост' и введя список интерфейсов, разделенных пробелами. Вы также можете использовать [VLAN](#)-обозначения вида ИНТЕРФЕЙС.НОМЕР VLAN (напр.: eth0.1).

Общие настройки

Состояние	Устройство: br-lan Время работы: 5д 17ч 5м 28с MAC-адрес: 40:BD:32:E5:67:61 Получение (RX): 4.09 ГБ (53652993 пакетов) Передача (TX): 152.54 МБ (2211469 пакетов) IPv4: 10.2.11.177/16
Протокол	Статический адрес
IPv4-адрес	10.2.11.177
Маска сети IPv4	255.255.0.0
IPv4-адрес шлюза	10.2.1.1
Широковещательный IPv4-адрес	
Использовать собственные DNS сервера	8.8.8.8

Рисунок 3.4.1 – Настройка IP-адреса шлюза и DNS-сервера в web-конфигураторе

Если контроллер имеет корректные сетевые настройки, то при выполнении пинг-запроса (вкладка Сеть/Диагностика) для адреса **gate.owencloud.ru** будут получены ответы:

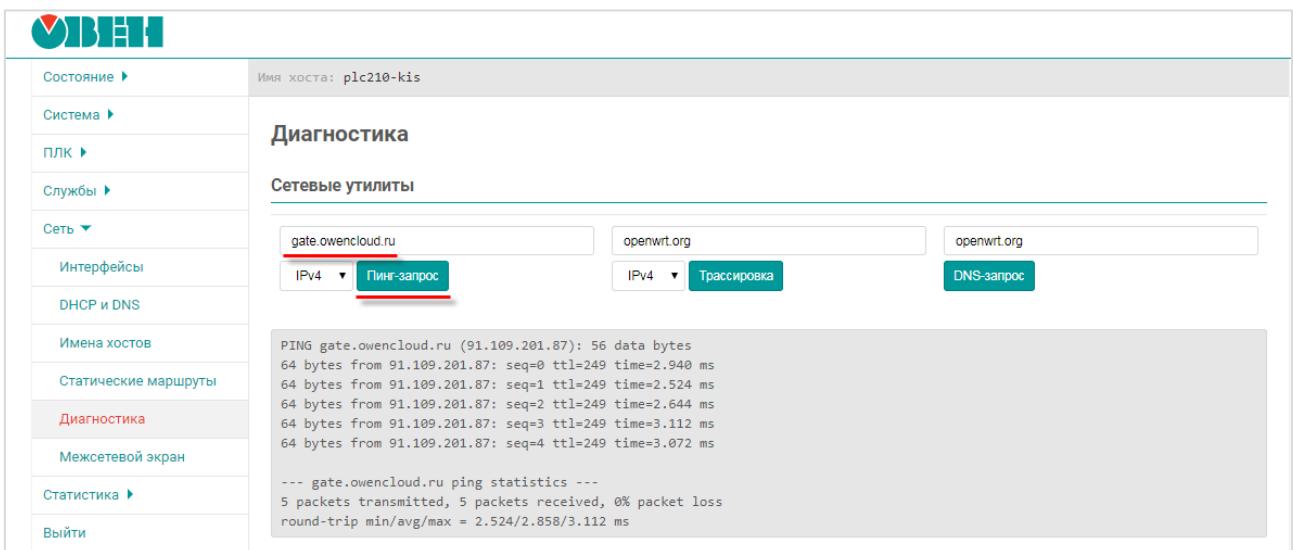


Рисунок 3.4.2 – Результат выполнения пинг-запроса

В случае отсутствия ответа следует проверить сетевые настройки контроллера и коммутационного оборудования, к которому он подключен.



ПРИМЕЧАНИЕ

Для связи с OwenCloud контроллер устанавливает **клиентское** соединение с ресурсом **gate.owencloud.ru** по порту **26502**.

2. В CODESYS создать проект с символьной конфигурацией согласно [п. 3.1](#).
3. В узле **OwenCloud** на вкладке **Конфигурация** указать пароль, которым будут шифроваться передаваемые данные. Этот пароль потребуется при добавлении прибора в облачный сервис. На вкладке **Соотнесение входов/выходов** можно привязать переменные для диагностики связи с **OwenCloud**.

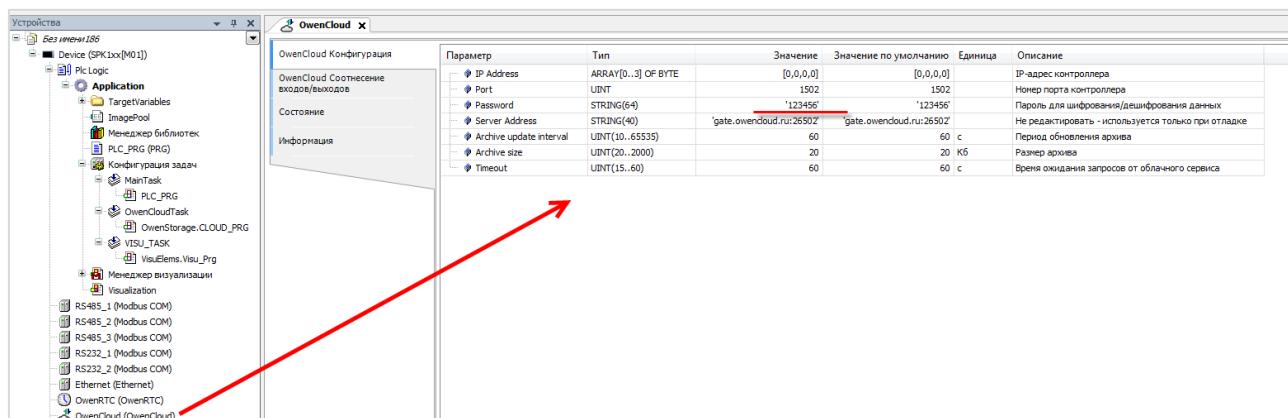


Рисунок 3.4.3 – Выбор пароля для шифрования данных

Таблица 3.1 – Описание каналов узла OwenCloud

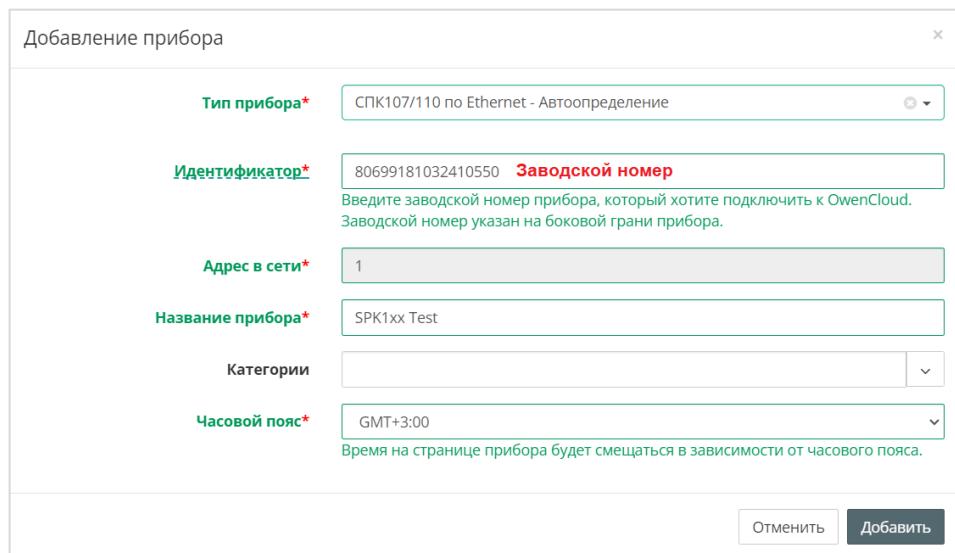
Канал	Тип	Описание
Вкладка Конфигурация		
Значения конфигурационных параметров задаются в проекте CODESYS. К ним не могут быть привязаны переменные		
IP Address	ARRAY [0..3] OF BYTE	IP-адрес интерфейса контроллера, через который осуществляется связь с OwenCloud . Значение 0.0.0.0 означает, что для связи используются все интерфейсы
Port	UINT	Порт контроллера, через который осуществляется связь с OwenCloud
Password	STRING(64)	Пароль шифрования данных, который также указывается в OwenCloud при добавлении контроллера
Server Address	STRING(40)	URL сервера OwenCloud . Параметр используется только при отладке, поэтому его значение следует редактировать только по рекомендации технической поддержки ОВЕН
Archive update interval	UINT (10...65535)	Период записи данных в архив (в секундах). Архив вычитывается облачным сервисом после разрыва и восстановления связи с контроллером. В архив включаются параметры символьной конфигурации с типом доступа Только чтение
Archive size	UINT (20...2000)	Размер архива в килобайтах. Для записи одной переменной (включая метку времени) используется от 20 до 34 байт (в зависимости от типа переменной)
Timeout	UINT(15...60)	Таймаут ожидания запросов от OwenCloud , который используется для детектирования отсутствия связи
Log level	UINT(0..7)	Уровень лога отладки. 0 – записываются только основные сообщения, 7 – детализация по каждой транзакции. Посмотреть лог можно в web-конфигураторе (Состояние/Журналы/Системный журнал)
Вкладка Соотнесение входов/выходов		
Тип доступа канала Enable OwenCloud : чтение и запись		
Тип доступа остальных каналов: только чтение		
OwenCloud enabled	BOOL	Флаг «запущен сервис связи с OwenCloud »
Folder Error	BOOL	Ошибка превышения максимального количества папок в проекте. Под «папкой» в данном контексте подразумевается элемент пространства имен в символьной конфигурации – то есть если в символьной конфигурации привязаны переменные одной программы, то это соответствует одной папке, а если переменные пяти разных программ – то пяти папкам. Максимально допустимое число папок – 100
Symbol Error	BOOL	Ошибка превышения максимального количества переменных, привязанных в символьной конфигурации. Максимально допустимое число переменных – 1000
No Symbol Config	BOOL	TRUE – в проекте отсутствует компонент Символьная конфигурация , который необходим для обмена с OwenCloud , или в символьной конфигурации не выбрано ни одной переменной
Warning Big Symbol Name	BOOL	Флаг наличия в символьной конфигурации как минимум одной переменной, длина имени или комментария которой превышает 32 символа. Наличие таких переменных может привести к ошибке связи с облачным сервисом. Значение канала формируется в момент установки связи с облачным сервисом

Warning Symbol Name	STRING(80)	Имя первой из обнаруженных переменных, длина имени или комментария которой превышает 32 символа. Значение канала формируется в момент установки связи с облачным сервисом
Symbol Count	UINT	Количество переменных, выделенных галочками в символьной конфигурации. Значение канала формируется в момент установки связи с облачным сервисом
Status	OwenTypes. CLOUD_STATUS	Статус связи с облачным сервисом. Тип канала – перечисление CLOUD_STATUS из библиотеки OwenTypes . Возможные значения: IDLE – выполняется установка соединения с OwenCloud; COMM_OK – наличие обмена данными с OwenCloud; COMM_ERROR – отсутствие обмена данными с OwenCloud в течение таймаута; NO_COMM – связь с OwenCloud отключена (канал Enable OwenCloud имеет значение FALSE);
Enable OwenCloud	BOOL	TRUE – включить сервис связи с OwenCloud, FALSE – отключить сервис связи с OwenCloud. Значение по умолчанию: TRUE

4. Подключиться к контроллеру и загрузить в него проект с символьной конфигурацией.
5. Зайти на главную страницу сервиса **OwenCloud**. Если вы еще не зарегистрированы в сервисе – необходимо пройти [процедуру регистрации](#).
6. Перейти на страницу **Администрирование**, открыть вкладку **Приборы**, нажать кнопку

() и указать следующие настройки:

- **Тип прибора** – выбрать тип, соответствующий подключаемому прибору (с режимом **Автоопределение**);
- **Идентификатор** – ввести заводской номер прибора (указан на корпусе прибора и отображается в web-конфигураторе на вкладке **Система/Состояние**);
- **Название прибора** – ввести название прибора;
- **Категории** – выбрать категории, к которым будет принадлежать прибор;
- **Часовой пояс** – указать часовой пояс, в котором находится прибор.



Добавление прибора

Тип прибора* СПК107/110 по Ethernet - Автоопределение

Идентификатор* 80699181032410550 **Заводской номер**
Введите заводской номер прибора, который хотите подключить к OwenCloud. Заводской номер указан на боковой грани прибора.

Адрес в сети* 1

Название прибора* SPK1xx Test

Категории

Часовой пояс* GMT+3:00
Время на странице прибора будет смещаться в зависимости от часового пояса.

Отменить Добавить

Рисунок 3.4.4 – Окно добавления прибора

Нажать кнопку **Добавить**.

7. На вкладке **Общие данные/Базовые настройки** следует ввести пароль из пп. 2:

Управление прибором: SPK1xx Test

Общие данные Настройки событий Настройки параметров

Базовые настройки Расположение на карте

Текущий идентификатор	80699181032410550
Тип прибора	СПК107/110 по Ethernet - Автоопределение
Новый идентификатор	
Пароль	123456
Название прибора*	SPK1xx Test
Категории	
Часовой пояс*	GMT+3:00

Время на странице прибора будет смещаться в зависимости от часового пояса.

Рисунок 3.4.5 – Ввод пароля шифрования данных

8. Следует нажать на пиктограмму , чтобы перейти к просмотру значений параметров прибора. Список переменных контроллера будет автоматически выгружен в **OwenCloud**.

Это может занять до нескольких минут. После появления статуса связи нажмите **F5**, чтобы обновить страницу.

9. Изменить значения переменных в CODESYS и наблюдать соответствующие изменения в **OwenCloud**. В случае необходимости изменить значения из облачного сервиса следует перейти на вкладку **Запись параметров**.

SPK SWU

Параметры Таблицы Графики События Запись параметров Конфигурации

Параметр

Код параметра	Значение
UID1073741829	1
UID1073741827	11
UID1073741828	22.330

Экспорт в Excel

Рисунок 3.4.6 – Просмотр параметров прибора

10. При импорте переменных в облачный сервис в качестве имен используются **комментарии** (русскоязычные комментарии поддерживаются). В случае отсутствия комментария в качестве имени параметра в облачном сервисе используется имя переменной из CODESYS. Для возможности импорта комментариев в качестве имен следует в **установках** символьной конфигурации выбрать пункт **Задать комментарии и атрибуты** и установить все галочки:

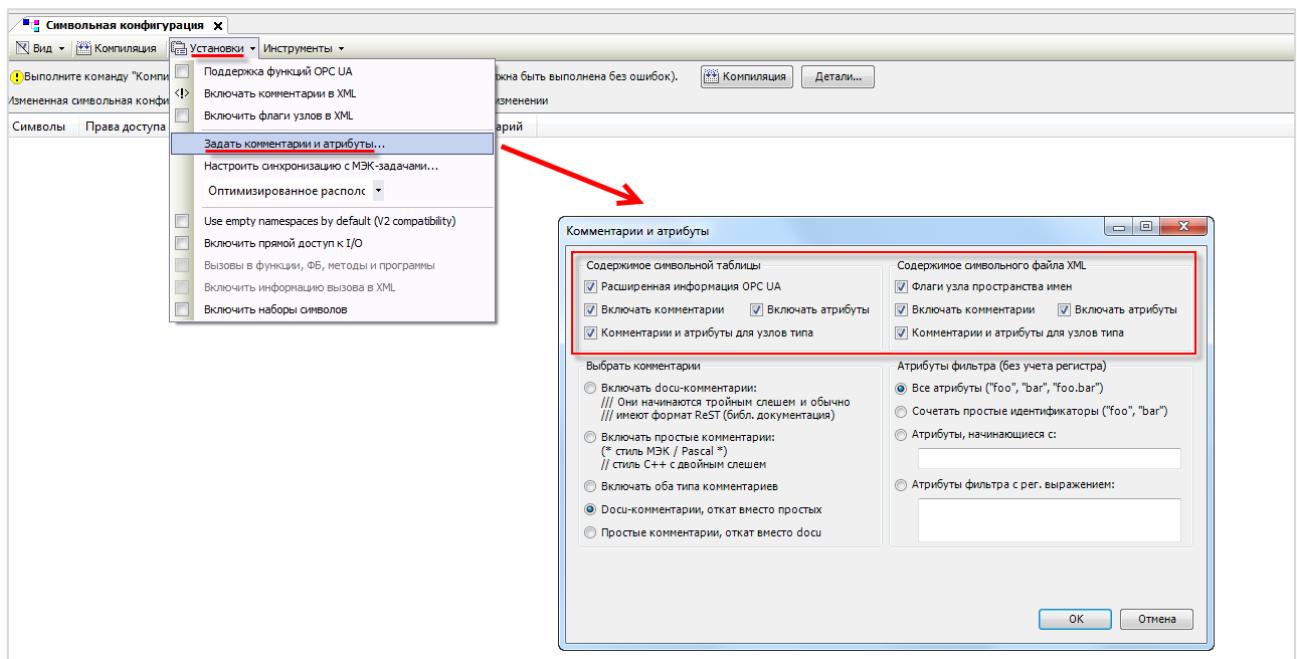


Рисунок 3.4.7 – Настройка импорта комментариев переменных OwenCloud

Для изменения названия параметров в OwenCloud следует открыть меню **Управление прибором** и перейти на вкладку **Настройки параметров**. Для изменения имени параметра следует нажать пиктограмму . В этом же меню можно настроить отображение параметра на графиках, в таблицах и событиях. Для изменения названия папки следует нажать на пиктограмму .

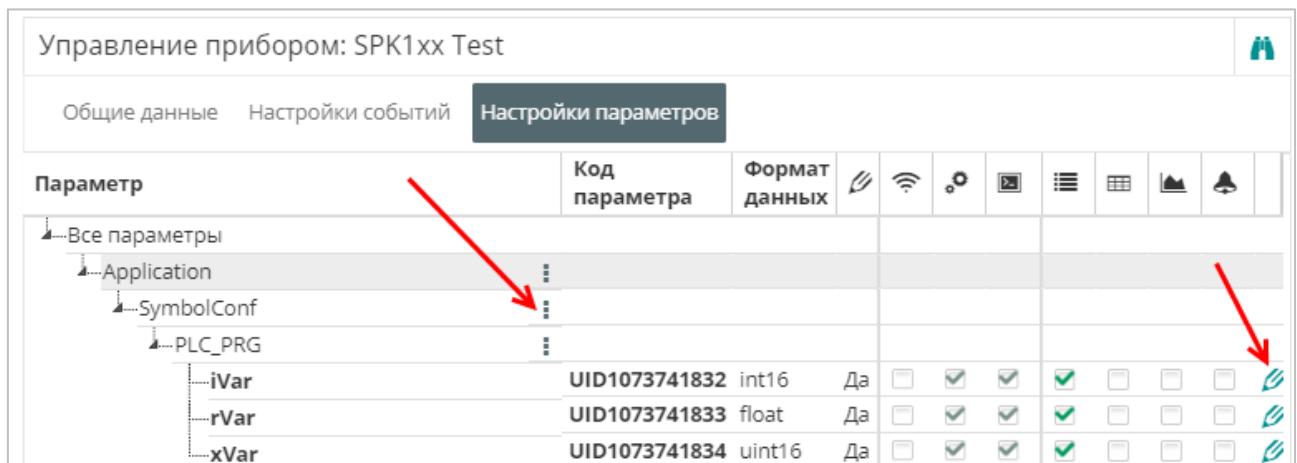


Рисунок 3.4.8 – Просмотр параметров прибора

**ПРИМЕЧАНИЕ**

Количество допустимых параметров контроллера, импортируемых в OwenCloud, ограничено **1000**. При превышении этого значения часть параметров не будет импортирована и в узле **OwenCloud** на вкладке **Соотнесение входов-выходов** канал **Symbol error** примет значение **TRUE**.

**ПРИМЕЧАНИЕ**

Количество папок в конфигурации ограничено **100**. Под папкой подразумевается пространство имен в пути к параметру – например, имя программы. При превышении этого значения параметры из некоторых папок не будут импортированы и в узле **OwenCloud** на вкладке **Соотнесение входов-выходов** канал **Folder error** примет значение **TRUE**.

**ПРИМЕЧАНИЕ**

Поддерживается импорт только элементарных типов данных (за исключением STRING, WSTRING, DT, DATE, TOD, TIME, LTIME). Импорт перечислений, структур и их элементов, ФБ и их элементов, указателей, ссылок и т. п. не поддерживается.

**ПРИМЕЧАНИЕ**

Максимальная поддерживаемая длина комментария/имени переменной при импорте в OwenCloud – **32 символа**. В случае превышения этого значения лишние символы будут отсечены. См. также каналы диагностики **Warning Big Symbol Name** и **Warning Symbol Name** в [таблице 3.1](#).

**ПРИМЕЧАНИЕ**

Параметры с типом доступа **Только чтение** добавляются в OwenCloud в группу опроса **Оперативные**, параметры с типом доступа **Чтение и запись** – в группы **Конфигурационные** и **Управляемые**.

4 Настройка обмена по протоколу Modbus

4.1 Настройка контроллера

В ряде случаев требуется настроить обмен между контроллером и OPC-сервером по протоколу **Modbus**. Обычно в этом случае контроллер используется в режиме **Modbus Slave**, а OPC-сервер выполняет роль **Modbus Master'a**.

В рамках примера рассматривается настройка обмена со следующими OPC-серверами:

- [MasterOPC Universal Modbus Server](#);
- [Owen OPC Server](#).

Настройка с другими OPC-серверами по протоколу Modbus производится аналогичным образом.

Ниже приведена обзорная инструкция по подготовке проекта с **Modbus TCP Slave**, который будет использоваться в примерах. Более подробная информация по настройке компонентов Modbus приведена в документе **CODESYS V3.5. Протокол Modbus**.

1. Следует создать новый проект **CODESYS V3.5** (язык программы не имеет значения).
2. Добавить в проект объединение с именем **Real_Word** (это связано с тем, что стандартные Modbus-компоненты CODESYS поддерживают только привязку переменных типа **BOOL** и **WORD**):

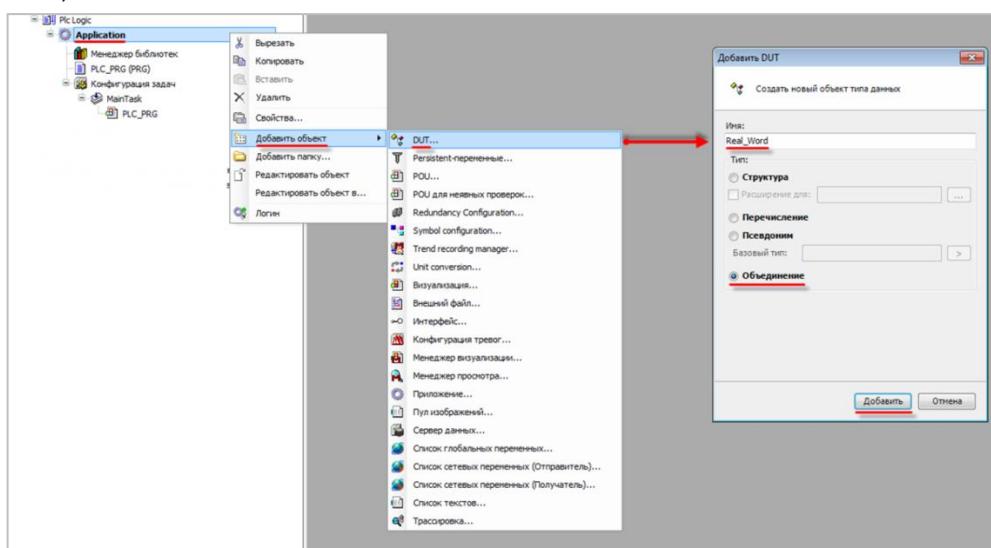


Рисунок 4.1.1 – Добавление в проект объединения

В объединении объявить переменную **rRealValue** типа **REAL** и массив **awModbusReal** типа **WORD**, содержащий два элемента:

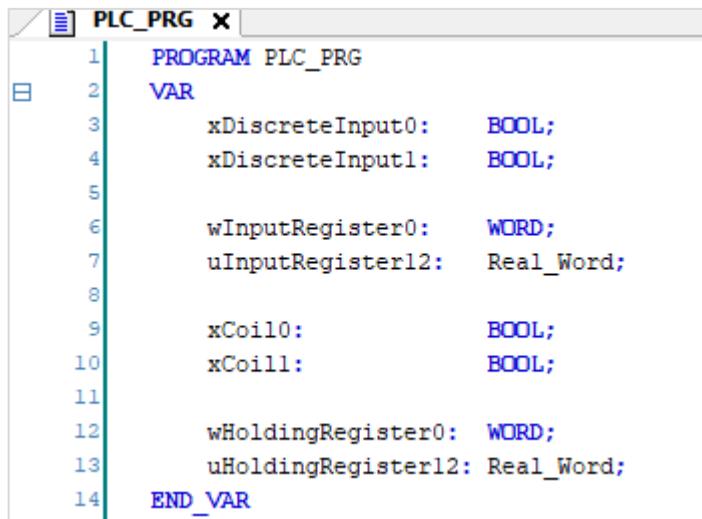
```

Real_Word x
1  TYPE Real_Word : 
2    UNION
3      rRealValue      :REAL;
4      awModbusReal   :ARRAY [0..1] OF WORD;
5    END_UNION
6  END_TYPE

```

Рисунок 4.1.2 – Объявление переменных объединения

3. В программе **PLC_PRG** объявить следующие переменные:



```

1 PROGRAM PLC_PRG
2
3     xDiscreteInput0:      BOOL;
4     xDiscreteInput1:      BOOL;
5
6     wInputRegister0:      WORD;
7     uInputRegister12:     Real_Word;
8
9     xCoil0:                BOOL;
10    xCoil1:                BOOL;
11
12    wHoldingRegister0:    WORD;
13    uHoldingRegister12:   Real_Word;
14
15 END_VAR

```

Рисунок 4.1.3 – Объявление переменных программы PLC_PRG

4. Добавить в проект компонент **Ethernet**.



ПРИМЕЧАНИЕ

Версия компонента не должна превышать версию таргет-файла контроллера.

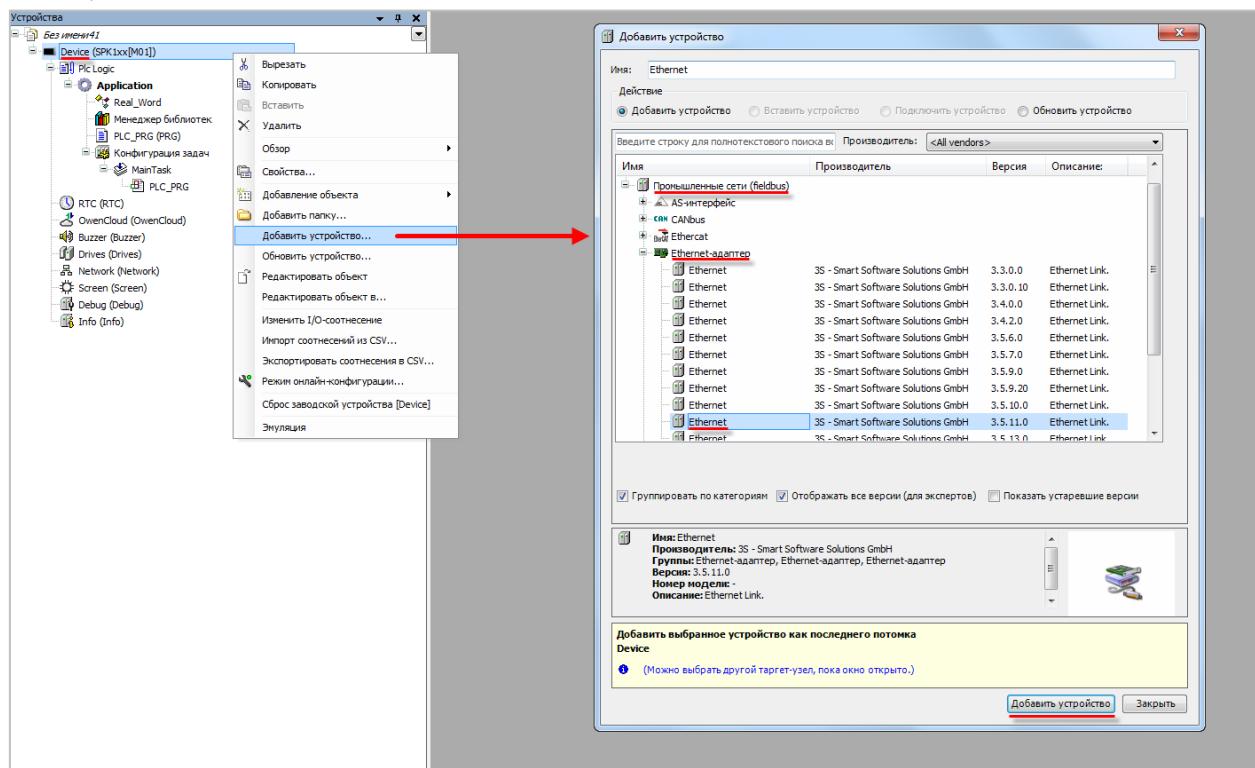


Рисунок 4.1.4 – Добавление компонента Ethernet

Затем следует установить соединение с контроллером на вкладке **Device**.

Для этого нужно на вкладке **Конфигурация Ethernet** выбрать нужный сетевой адаптер контроллера:

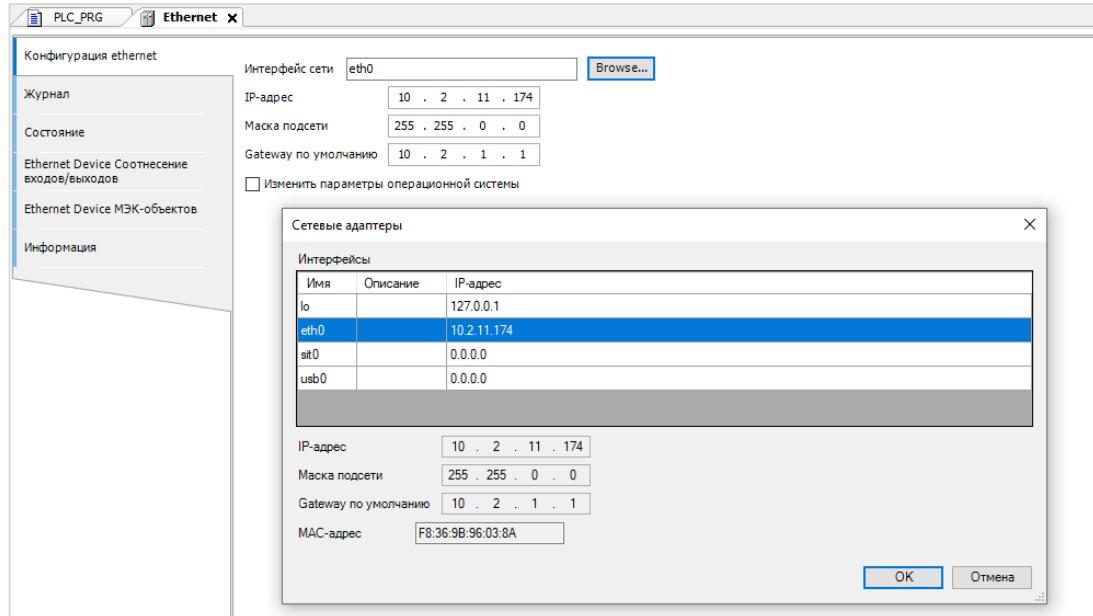


Рисунок 4.1.5 – Настройки компонента Ethernet

5. В компонент **Ethernet** добавить компонент **Modbus TCP Slave Device**.



ПРИМЕЧАНИЕ

Версия компонента не должна превышать версию target-файла контроллера.

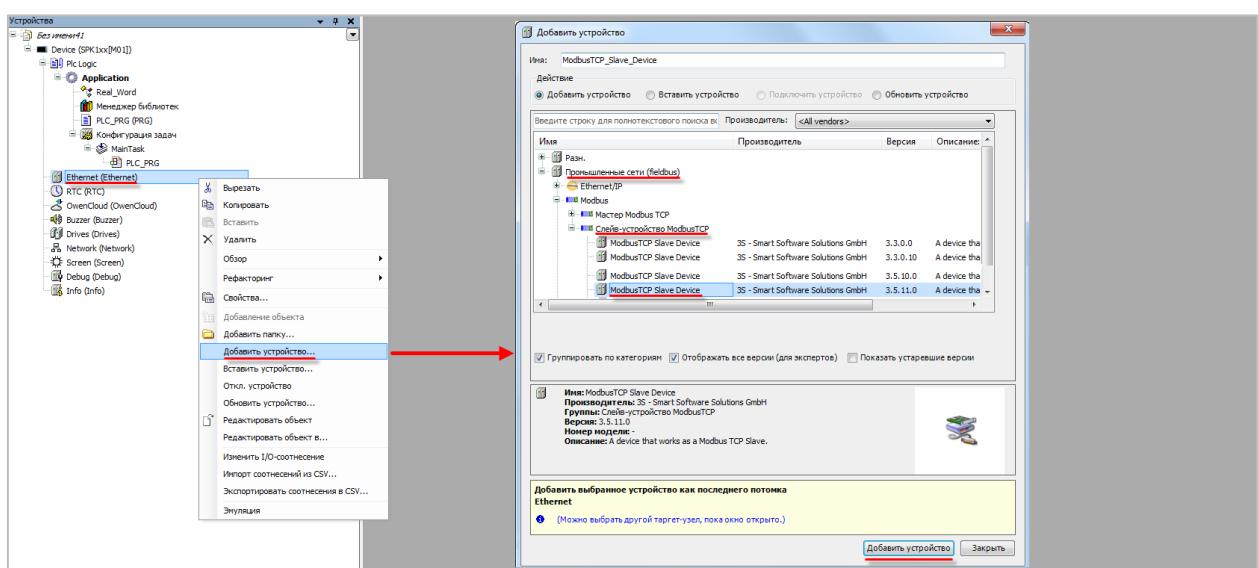


Рисунок 4.1.6 – Добавление компонента Modbus TCP Slave Device

В настройках компонента на вкладке **Страница конфигурации** следует установить галочки **Запись** (для возможности изменения coils и holding-регистров из программы контроллера) и **Дискретные битовые области** (для выделения coils и discrete Inputs в отдельные области памяти – по умолчанию они наложены на области holding-регистров/input-регистров соответственно).

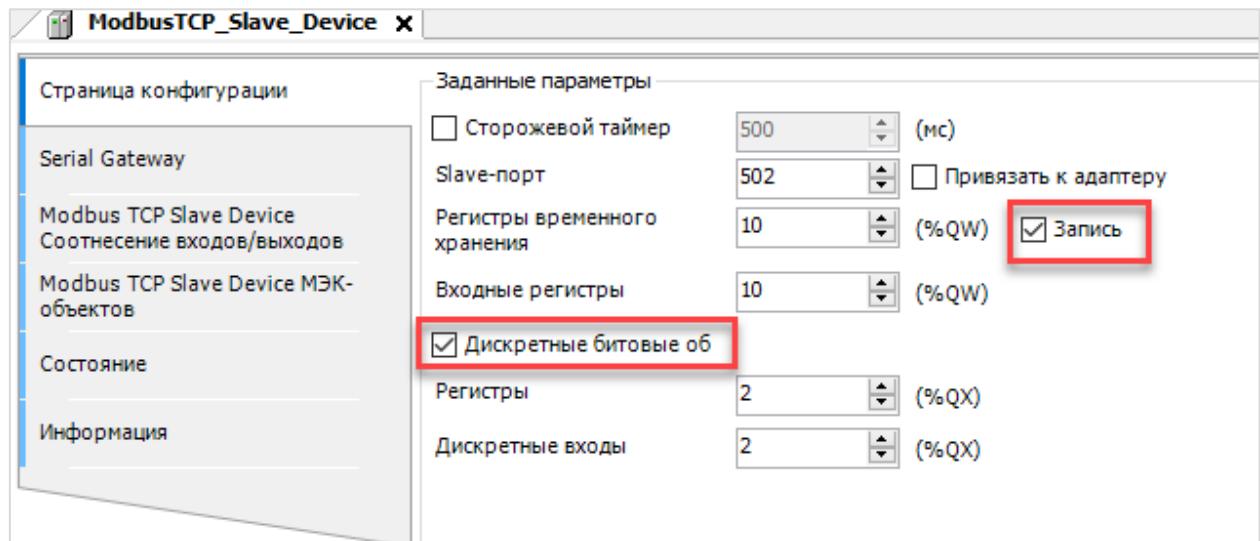


Рисунок 4.1.7 – Настройки компонента Modbus TCP Slave Device

На вкладке **Modbus TCP Slave Device Соотнесение входов/выходов** привязать к регистрам переменные программы. У параметра **Всегда обновлять переменные** следует установить значение **Включено 2**.

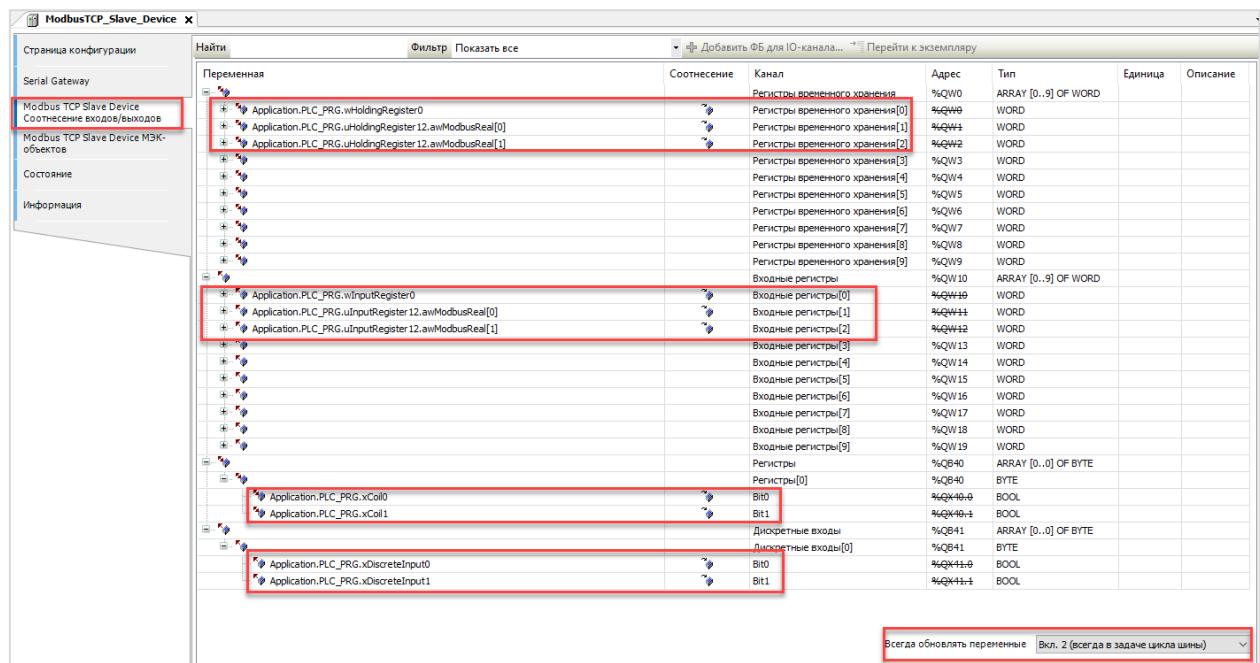


Рисунок 4.1.8 – Привязка переменных к регистрам

В результате в контроллере будет сформирована следующая карта регистров:

Таблица 4.1 – Карта регистров контроллера

Область памяти	Адрес	Переменная	Тип переменной
Holding-регистры	0	wHoldingRegister0	WORD
	1-2	rHoldingRegister12	REAL (Real_Word)
Input-регистры	0	wHoldingRegister0	WORD
	1-2	rHoldingRegister12	REAL (Real_Word)
Coils	0	xCoil0	BOOL
	1	xCoil1	BOOL
Discrete Inputs	0	xDiscreteInput0	BOOL
	1	xDiscreteInput1	BOOL

Более подробно вопросы настройки и особенности работы компонента **Modbus Slave** рассмотрены в документе **CODESYS V3.5. Протокол Modbus**.

Созданный в данном пункте проект доступен для скачивания: [Example_OpcModbus.zip](#)



ПРИМЕЧАНИЕ

В рамках примера рассматривается обмен по протоколу **Modbus TCP**. В случае необходимости использовать протокол **Modbus RTU** следует вместо компонентов **Ethernet** и **Modbus TCP Slave Device** использовать компоненты **Modbus COM** и **Modbus Serial Slave Device**. Более подробная информация приведена в руководстве **CODESYS V3.5. Протокол Modbus**.



ПРИМЕЧАНИЕ

В режиме отладки значения переменных, привязанных к области coils и holding-регистров, можно изменить только с помощью команды **Фиксировать значения** (но не **Записать значения**). После записи фиксацию можно отключить. Это связано с особенностями работы компонента **Modbus Slave Device** при установленной галочке **Запись**.

4.2 Настройка MasterOPC Universal Modbus Server

Для настройки OPC-сервера следует:

1. Установить и запустить [MasterOPC Universal Modbus Server](#).
2. Нажать **ПКМ** на узел **Server** и добавить коммуникационный узел. В его настройках указать тип **TCP/IP** и сетевые настройки (**IP-адрес** и **порт**). Сетевые настройки должны соответствовать настройкам контроллера (см. [п. 4.1](#), рисунки 4.1.5 и 4.1.7).

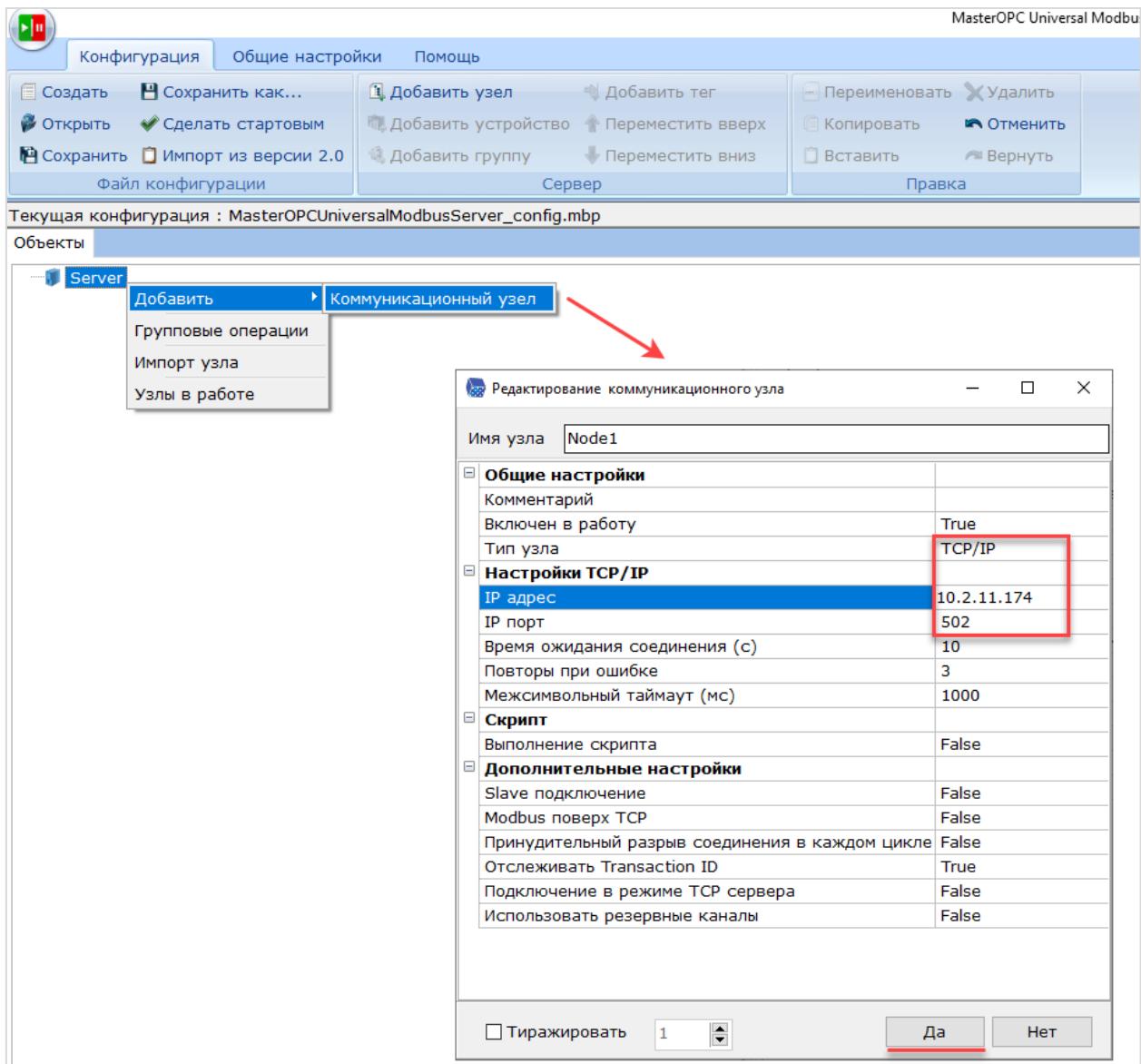


Рисунок 4.2.1 – Добавление коммуникационного узла

3. Нажать **ПКМ** на коммуникационный узел и добавить устройство. В настройках устройства указать адрес (если контроллер программируется в версии **CODESYS V3.5 SP16 Patch 3**, следует обязательно указать для **Modbus TCP Slave** адрес **0** или **255** (см. [подробности](#)); в более старых и новых версиях CODESYS можно указать любой адрес). По умолчанию период опроса устройства составляет 1000 мс – в случае необходимости можно изменить это значение.

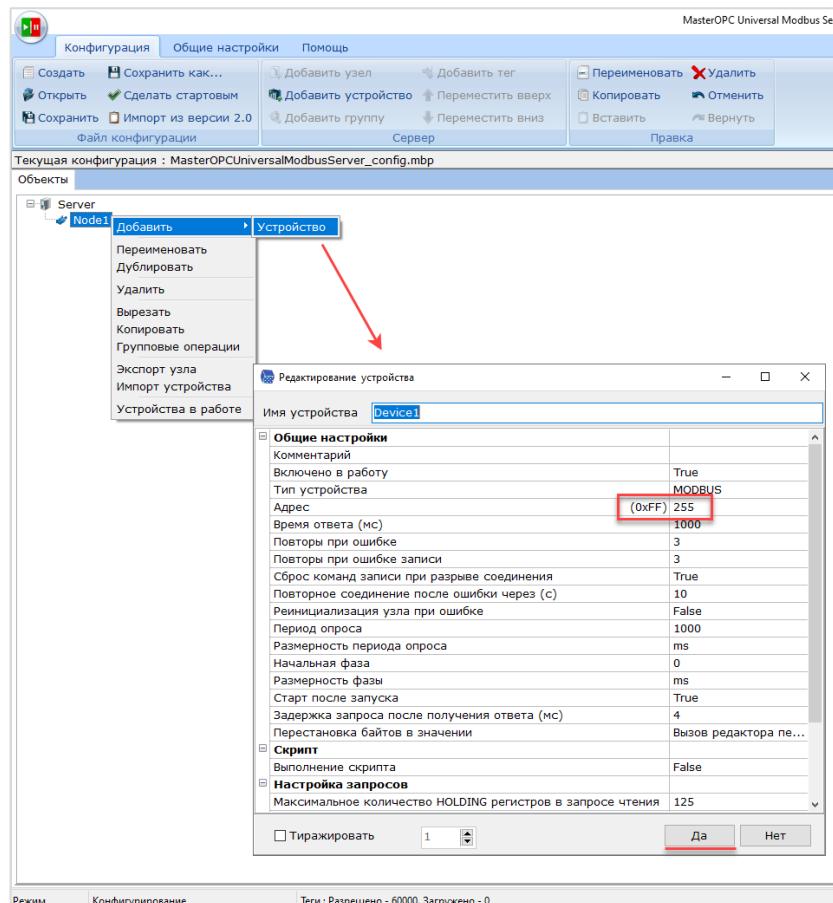


Рисунок 4.2.2 – Добавление устройства

4. Нажать **ПКМ** на устройство и добавить 8 тегов в соответствии с [таблицей 4.1](#). Настройки тегов приведены ниже.

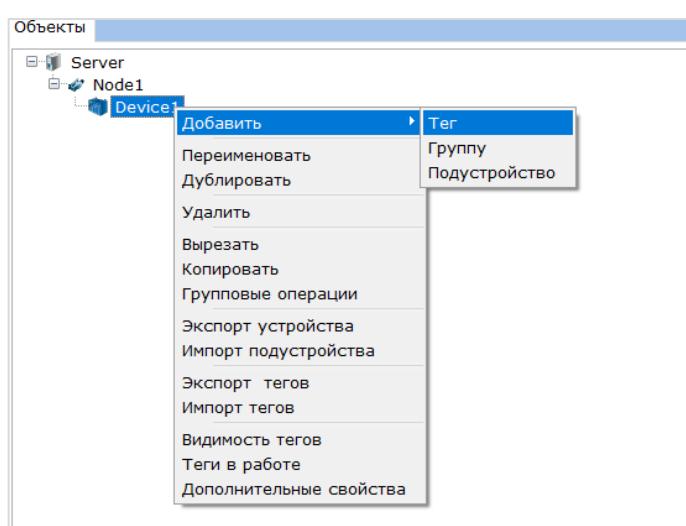


Рисунок 4.2.3 – Добавление тегов в OPC-сервер

Тег <<HOLDING_REGISTERS>> : wHoldingRegister0	
Общие настройки	
Комментарий	
Включен в работу	True
Адрес	(0x0000) 0
Тип данных в устройстве	uint16
Тип данных в сервере	uint32
Тип доступа	ReadWrite
Использовать перестановку байтов устройства	True
Последний тег в групповом запросе	False
Пересчет (A*X + B)	False
Скрипт	
Разрешение выполнения скрипта после чтения	False
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Извлечение бита из данных	False
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
Принудительная запись командой 6	False
HDA	
HDA доступ	False

Рисунок 4.2.4 – Настройки тега wHoldingRegister0

Тег <<HOLDING_REGISTERS>> : rHoldingRegister12	
Общие настройки	
Комментарий	
Включен в работу	True
Адрес	(0x0001) 1
Тип данных в устройстве	float
Тип данных в сервере	float
Тип доступа	ReadWrite
Использовать перестановку байтов устройства	False
Перестановка байтов в значении	10325476
Последний тег в групповом запросе	False
Пересчет (A*X + B)	False
Скрипт	
Разрешение выполнения скрипта после чтения	False
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Извлечение бита из данных	False
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
Принудительная запись командой 6	False
HDA	
HDA доступ	False

Рисунок 4.2.5 – Настройки тега wHoldingRegister12 (порядок байт во Float отличается в ПЛК и OPC, поэтому требуется перестановка)

Тег <<INPUT_REGISTERS>> : wInputRegister0	
Общие настройки	
Комментарий	
Включен в работу	True
Адрес	(0x0000) 0
Тип данных в устройстве	uint16
Тип данных в сервере	uint32
Тип доступа	ReadOnly
Использовать перестановку байтов устройства	True
Последний тег в групповом запросе	False
Пересчет (A*X + B)	False
Скрипт	
Разрешение выполнения скрипта после чтения	False
Дополнительно	
Извлечение бита из данных	False
HDA	
HDA доступ	False

Рисунок 4.2.6 – Настройки тега wInputRegister0

Тег <<INPUT_REGISTERS>> : rInputRegister12	
Общие настройки	
Комментарий	
Включен в работу	True
Адрес	(0x0001) 1
Тип данных в устройстве	float
Тип данных в сервере	float
Тип доступа	ReadOnly
Использовать перестановку байтов устройства	False
Перестановка байтов в значении	10325476
Последний тег в групповом запросе	False
Пересчет (A*X + B)	False
Скрипт	
Разрешение выполнения скрипта после чтения	False
Дополнительно	
Извлечение бита из данных	False
HDA	
HDA доступ	False

Рисунок 4.2.7 – Настройки тега wInputRegister12 (порядок байт во Float отличается в ПЛК и OPC, поэтому требуется перестановка)

Тег <<COILS>> : xCoil0	
Общие настройки	
Комментарий	
Включен в работу	True
Адресс	(0x0000) 0
Тип данных в устройстве	bool
Тип данных в сервере	bool
Тип доступа	ReadWrite
Скрипт	
Разрешение выполнения скрипта после чтения	False
Разрешение выполнения скрипта перед записью	False
Дополнительно	
Наличие отдельного регистра записи	False
Чтение сразу после записи	False
Сброс команды записи	True
HDA	
HDA доступ	False

Рисунок 4.2.8 – Настройки тегов xCoil0 и xCoil1 (для xCoil1 – адрес 1)

Тег <<DISCRETE_INPUTS>> : xDiscreteInput0	
Общие настройки	
Комментарий	
Включен в работу	True
Адресс	(0x0000) 0
Тип данных в устройстве	bool
Тип данных в сервере	bool
Тип доступа	ReadOnly
Скрипт	
Разрешение выполнения скрипта после чтения	False
HDA	
HDA доступ	False

Рисунок 4.2.9 – Настройки тегов xDiscreteInput0 и xDiscreteInput1 (для xDiscreteInput1 – адрес 1)

Для проверки связи можно запустить OPC-сервер. Если OPC уже подключен к SCADA-системе, то он будет автоматически запущен при старте проекта SCADA.

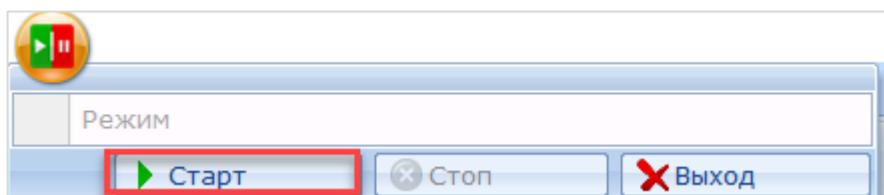


Рисунок 4.2.10 – Команда запуска OPC-сервера

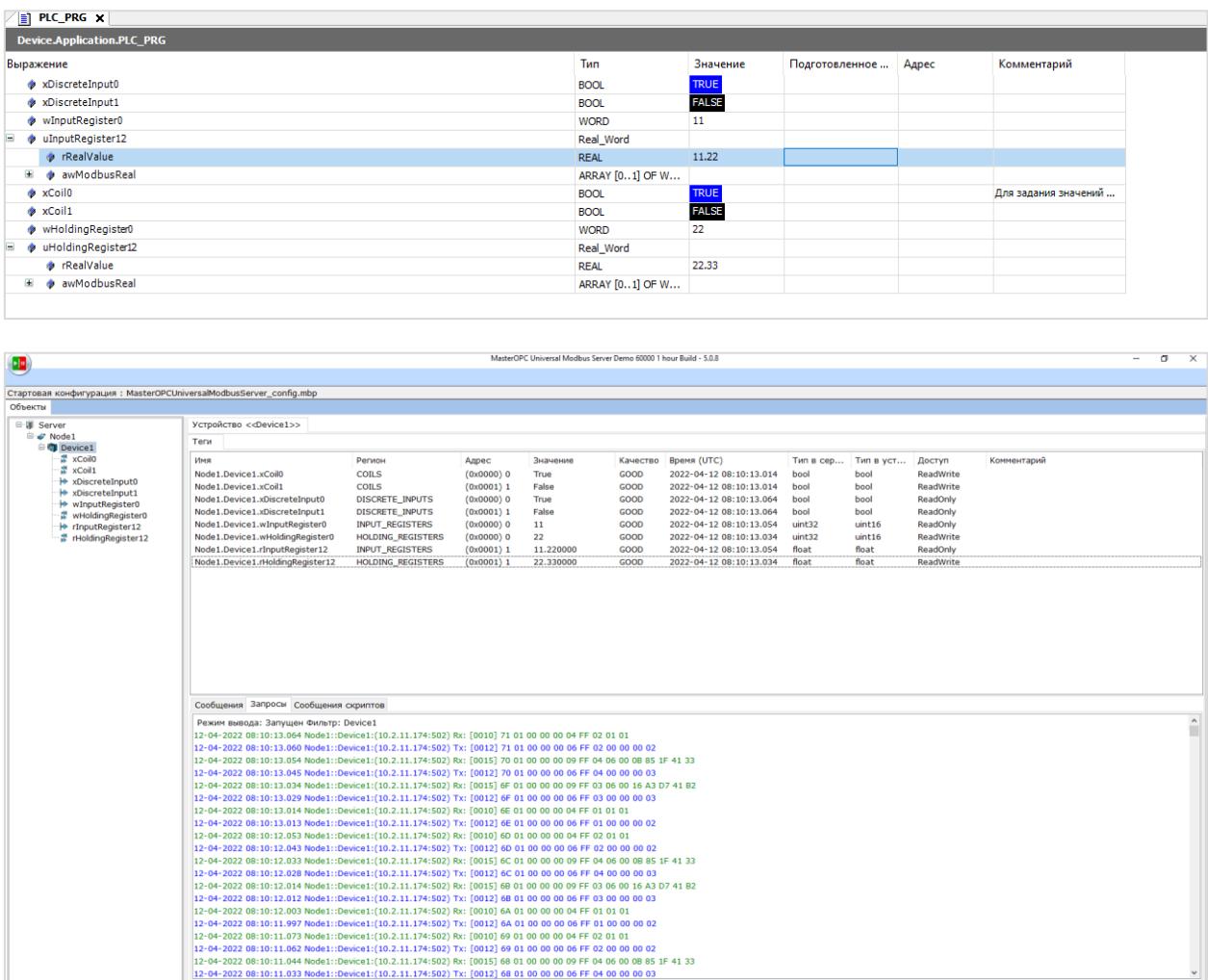


Рисунок 4.2.11 – Успешный обмен между ПЛК и ОРС-сервером

**ПРИМЕЧАНИЕ**

Совместное использование **MasterOPC Universal Modbus Server** и библиотеки **OwenCommunication** для реализации в ПЛК **Modbus TCP Slave** позволяют организовать считывание с ПЛК файлов архивов и передачу их в SCADA-систему с помощью технологии OPC HDA.

См. следующие ссылки:

- пункт **СПК1xx [M01] (Modbus TCP Slave)** – чтение файлов с помощью 20 функции **Modbus** в документе **CODESYS V3.5 Протокол Modbus**;
- демонстрацию данного функционала в рамках [вебинара про библиотеку OwenCommunication](#);
- [описание формата архивных файлов](#);
- [пример создания архивного файла](#).

4.3 Настройка Owen OPC Server

Для настройки OPC-сервера следует:

1. Установить и запустить [Owen OPC Server](#).
2. Нажать **ПКМ** на узел **Сервер** и добавить коммуникационный узел. В его настройках указать тип **Modbus TCP/IP**.

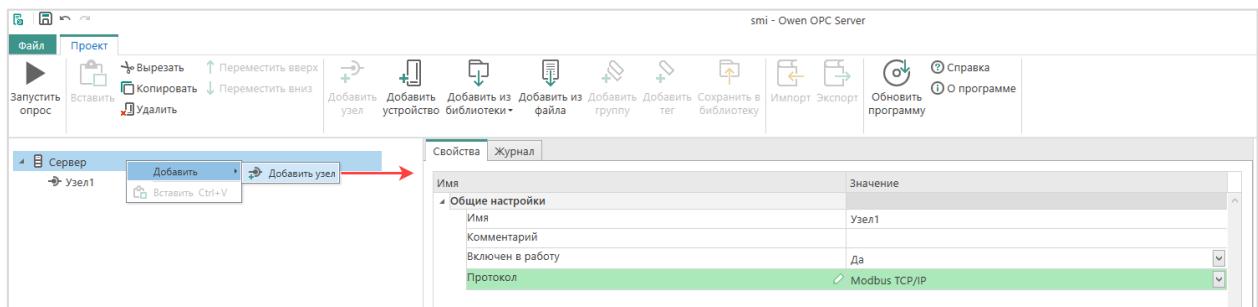


Рисунок 4.3.1 – Добавление коммуникационного узла

3. Нажать **ПКМ** на коммуникационный узел и добавить устройство. В настройках устройства указать сетевые настройки (**IP-адрес** и **порт** – см. [п. 4.1](#), рисунки 4.1.5 и 4.1.7) и адрес (если контроллер программируется в версии **CODESYS V3.5 SP16 Patch 3**, следует обязательно указать для **Modbus TCP Slave** адрес **0** или **255** (см. [подробности](#)); в более старых и новых версиях CODESYS можно указать любой адрес). По умолчанию период опроса устройства составляет 1000 мс – в случае необходимости можно изменить это значение.

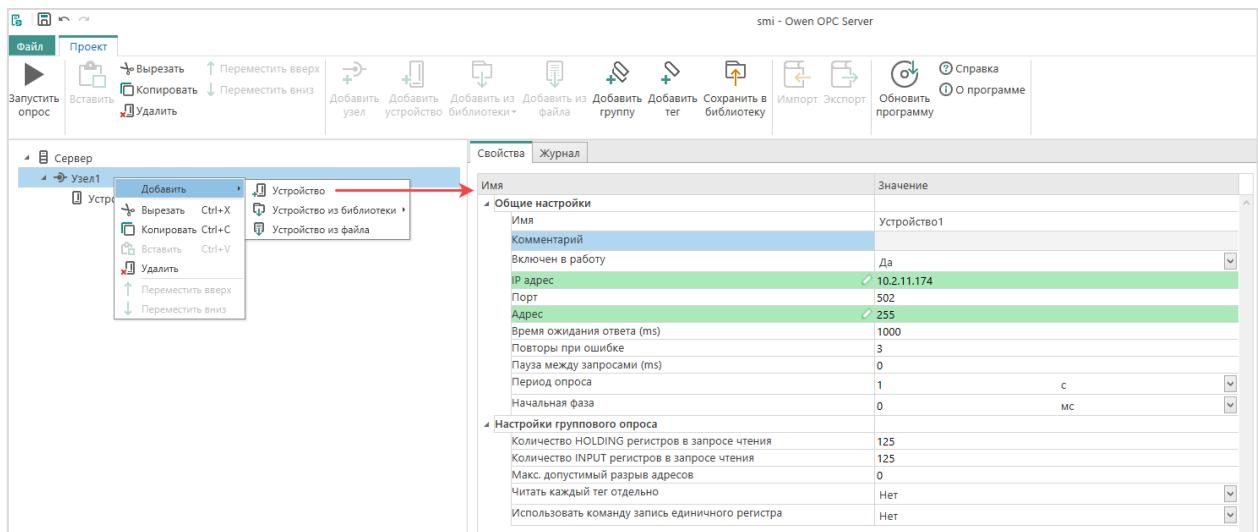


Рисунок 4.3.2 – Добавление устройства

4. Нажать **ПКМ** на устройство и добавить 8 тегов в соответствии с [таблицей 4.1](#). Настройки тегов приведены ниже.

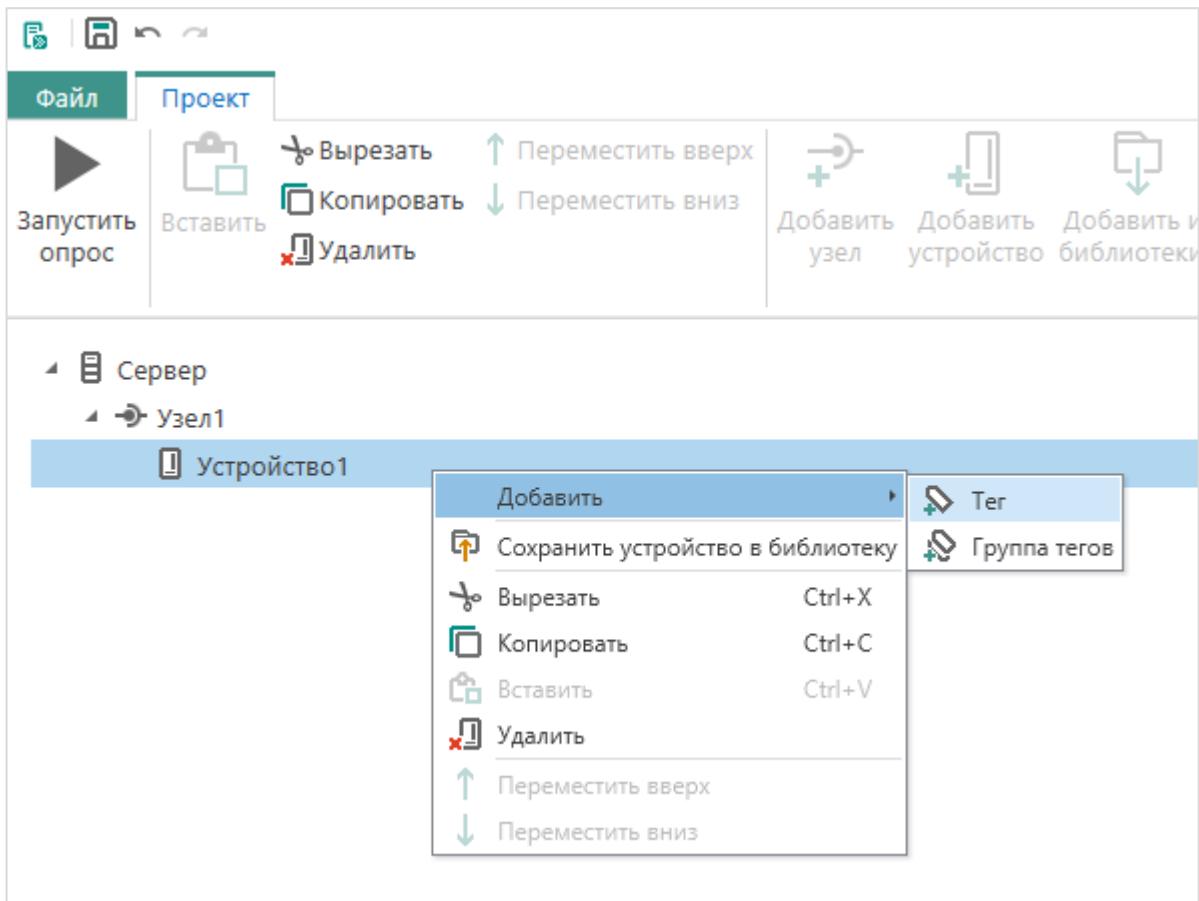


Рисунок 4.3.3 – Добавление тегов в OPC-сервер

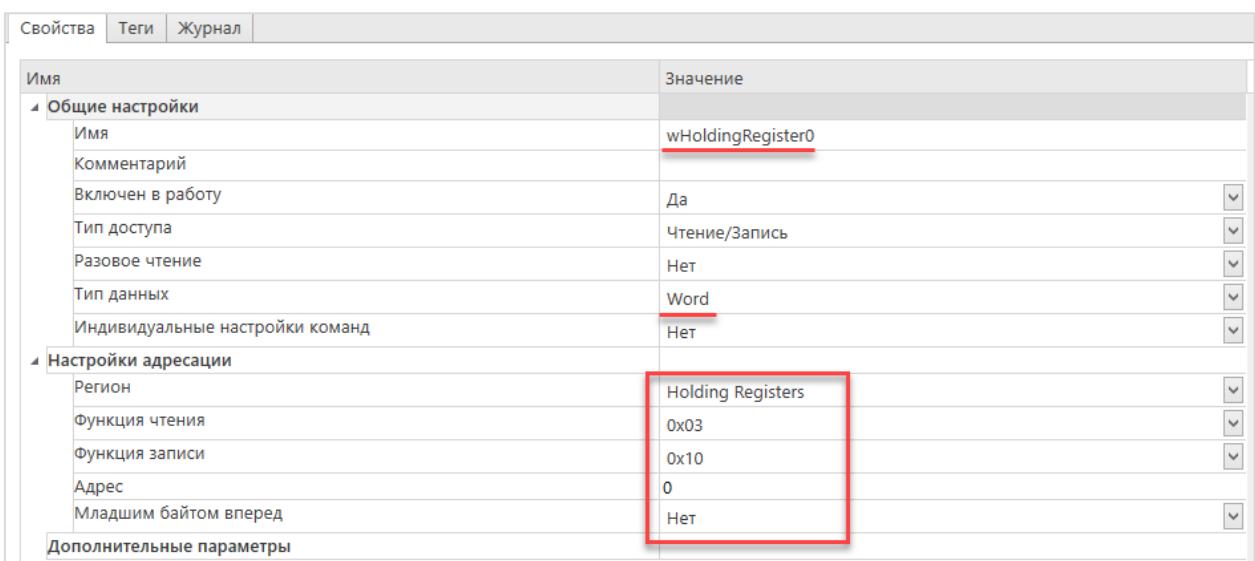


Рисунок 4.3.4 – Настройки тега wHoldingRegister0

Свойства	Теги	Журнал
Имя		Значение
▲ Общие настройки		
Имя		rHoldingRegister12
Комментарий		
Включен в работу		Да
Тип доступа		Чтение/Запись
Разовое чтение		Нет
Тип данных		Float
Индивидуальные настройки команд		Нет
▲ Настройки адресации		
Регион		Holding Registers
Функция чтения		0x03
Функция записи		0x10
Адрес		1
Младшим байтом вперед		Нет
Младшим регистром вперед		Да
Дополнительные параметры		

Рисунок 4.3.5 – Настройки тега wHoldingRegister12 (порядок байт во Float отличается в ПЛК и OPC, поэтому требуется перестановка)

Свойства	Теги	Журнал
Имя		Значение
▲ Общие настройки		
Имя		wInputRegister0
Комментарий		
Включен в работу		Да
Тип доступа		Только чтение
Разовое чтение		Нет
Тип данных		Word
Индивидуальные настройки команд		Нет
▲ Настройки адресации		
Регион		Input Registers
Функция чтения		0x04
Функция записи		0
Адрес		Нет
Младшим байтом вперед		
Младшим регистром вперед		
Дополнительные параметры		

Рисунок 4.3.6 – Настройки тега wInputRegister0

Свойства		Теги	Журнал
Имя	Значение		
▪ Общие настройки			
Имя	rInputRegister12		
Комментарий			
Включен в работу	Да		
Тип доступа	Только чтение		
Разовое чтение	Нет		
Тип данных	Float		
Индивидуальные настройки команд	Нет		
▪ Настройки адресации			
Регион	Input Registers		
Функция чтения	0x04		
Функция записи			
Адрес	1		
Младшим байтом вперед	Нет		
Младшим регистром вперед	Да		
Дополнительные параметры			

Рисунок 4.3.7 – Настройки тега rInputRegister12 (порядок байт во Float отличается в ПЛК и ОРС, поэтому требуется перестановка)

Свойства		Теги	Журнал
Имя	Значение		
▪ Общие настройки			
Имя	xCoil0		
Комментарий			
Включен в работу	Да		
Тип доступа	Чтение/Запись		
Разовое чтение	Нет		
Тип данных	Boolean		
Индивидуальные настройки команд	Нет		
▪ Настройки адресации			
Регион	Coils		
Функция чтения	0x01		
Функция записи	0x0F		
Адрес	0		
Дополнительные параметры			

Рисунок 4.3.8 – Настройки тегов xCoil0 и xCoil1 (для xCoil1 – адрес 1)

Свойства		Теги	Журнал
Имя	Значение		
▪ Общие настройки			
Имя	xDiscreteInput0		
Комментарий			
Включен в работу	Да		
Тип доступа	Только чтение		
Разовое чтение	Нет		
Тип данных	Boolean		
Индивидуальные настройки команд	Нет		
▪ Настройки адресации			
Регион	Discrete Inputs		
Функция чтения	0x02		
Функция записи			
Адрес	0		
Дополнительные параметры			

Рисунок 4.3.9 – Настройки тегов xDiscreteInput0 и xDiscreteInput1 (для xDiscreteInput1 – адрес 1)

Для проверки связи можно запустить OPC-сервер. Если OPC уже подключен к SCADA-системе, то он будет автоматически запущен при старте проекта SCADA.

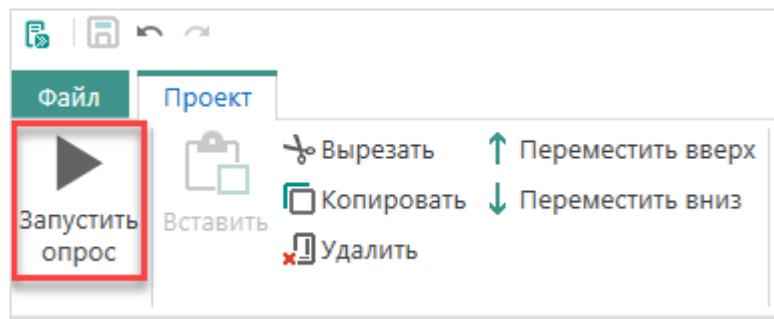


Рисунок 4.3.10 – Команда запуска OPC-сервера

Выражение	Тип	Значение	Подготовленное ...	Адрес	Комментарий
xDiscreteInput0	BOOL	TRUE			
xDiscreteInput1	BOOL	FALSE			
wInputRegister0	WORD	11			
uInputRegister12	Real_Word				
rRealValue	REAL	11.22			
awModbusReal	ARRAY [0..1] OF W...				
xCoil0	BOOL	TRUE			
xCoil1	BOOL	FALSE			
wHoldingRegister0	WORD	22			
uHoldingRegister12	Real_Word				
rRealValue	REAL	22.33			
awModbusReal	ARRAY [0..1] OF W...				

Имя	Адрес	Значение	Тип данных	Качество	Комментарий
Устройство1.wHoldingRegister0	Holding Registers [0]	22	Word	GOOD	
Устройство1.HoldingRegister12	Holding Registers [1]	22.33	Float	GOOD	
Устройство1.wInputRegister0	Input Registers [0]	11	Word	GOOD	
Устройство1.iInputRegister12	Input Registers [1]	11.22	Float	GOOD	
Устройство1.xCoil0	Coils [0]	True	Boolean	GOOD	
Устройство1.xCoil1	Coils [1]	False	Boolean	GOOD	
Устройство1.xDiscreteInput0	Discrete Inputs [0]	True	Boolean	GOOD	
Устройство1.xDiscreteInput1	Discrete Inputs [1]	False	Boolean	GOOD	

Рисунок 4.3.11 – Успешный обмен между ПЛК и OPC-сервером

5 Подключение OPC-сервера к MasterSCADA 3.x

5.1 Подключение OPC DA-сервера

Для подключения OPC DA-сервера к **MasterSCADA 3.x** следует:

1. Запустить [MasterSCADA 3.x](#) и создать новый или открыть существующий проект.
2. Нажать **ПКМ** на узел **Система** и добавить **Компьютер** (*если он отсутствует в проекте*).

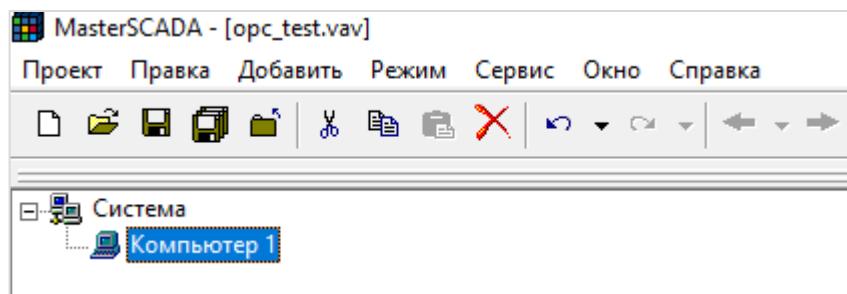


Рисунок 5.1.1 – Добавление компьютера в проект SCADA

3. Нажать **ПКМ** на узел **Компьютер**, использовать команду **Вставить OPC-сервер** и выбрать нужный OPC DA-сервер:

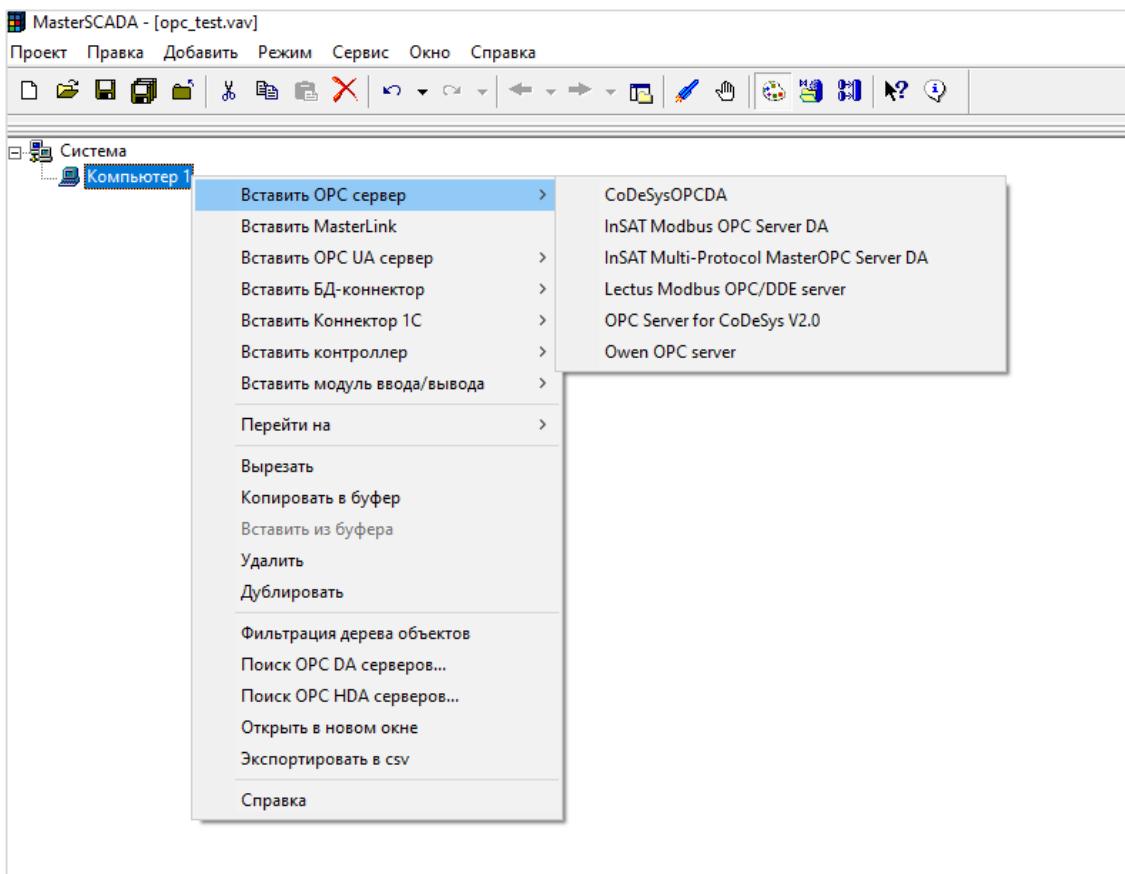


Рисунок 5.1.2 – Добавление OPC-сервера

4. Нажать **ПКМ** на добавленный OPC-сервер и использовать команду **Вставить – OPC переменные** или **Все переменные и группы**. В случае использования команды **OPC переменные** потребуется в открывшемся окно выделить нужные переменные.

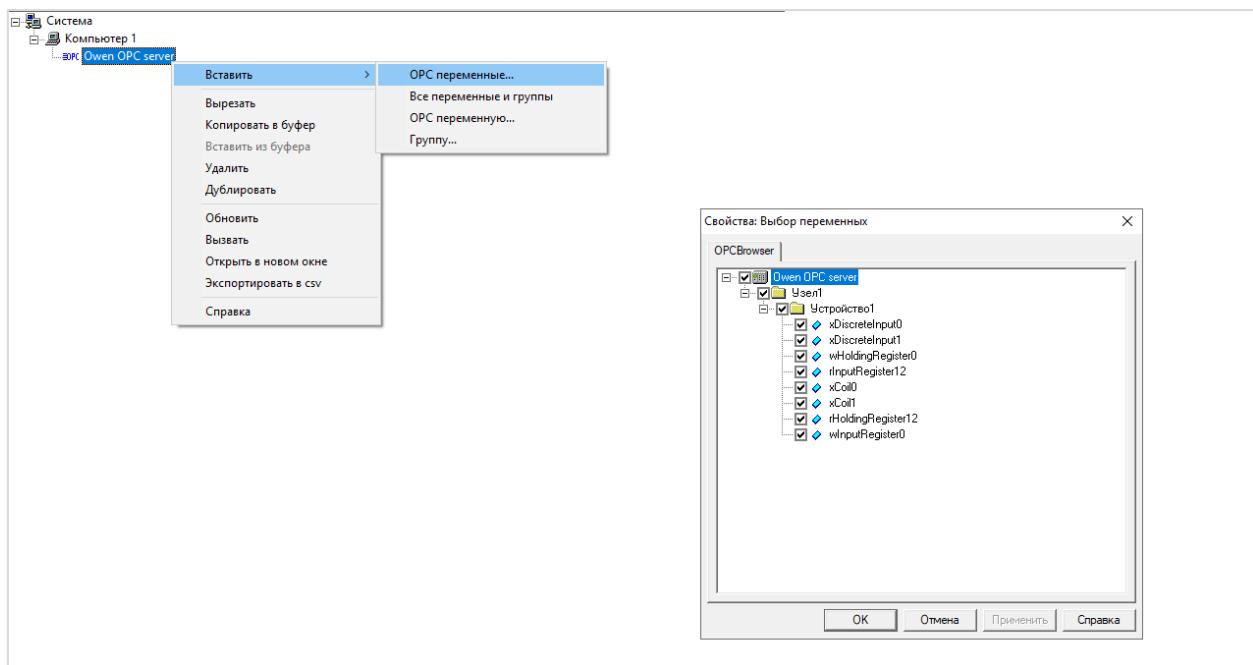


Рисунок 5.1.3 – Импорт тегов OPC-сервера

5. В результате теги OPC будут добавлены в дерево системы. Для проверки связи с OPC следует запустить на исполнение проект SCADA.

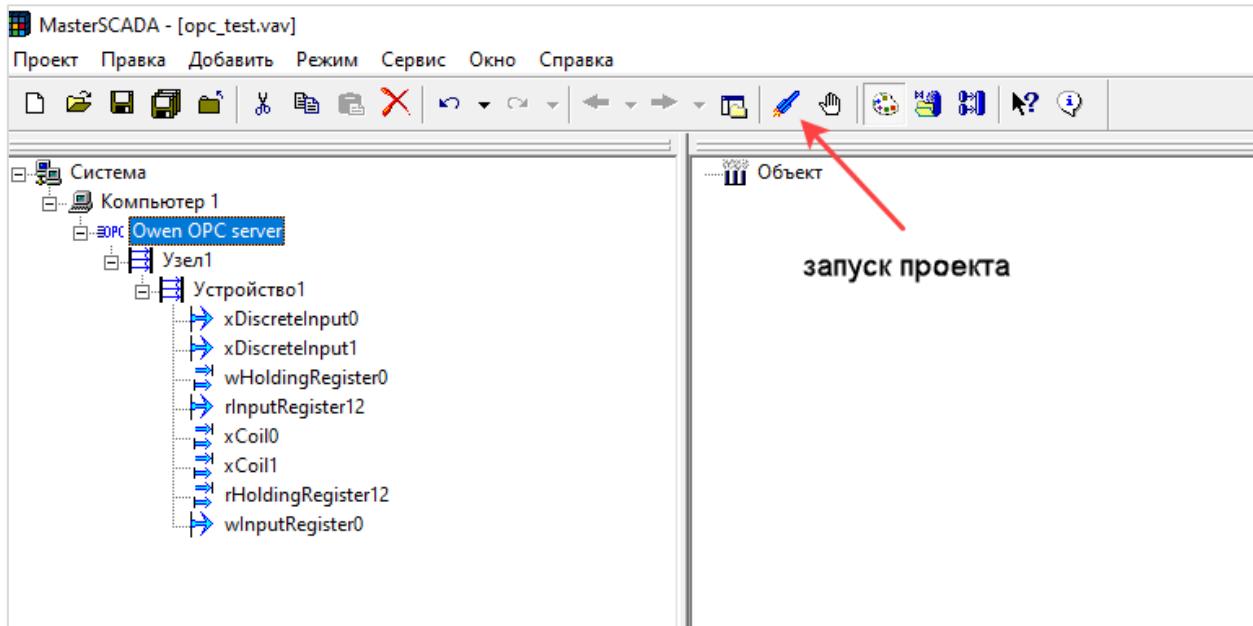


Рисунок 5.1.4 – Отображение добавленных тегов в дереве системы

Для изменения значения тега следует два раза нажать **ЛКМ** на его текущее значение.

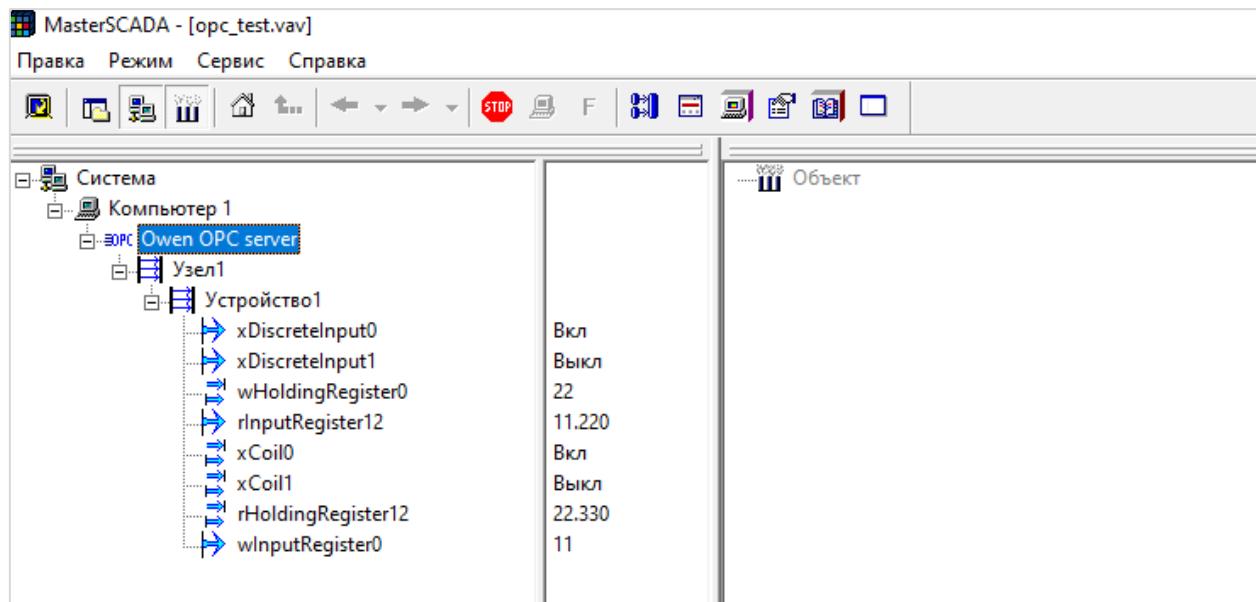


Рисунок 5.1.5 – Успешный обмен между SCADA и OPC-сервером

5.2 Подключение OPC UA-сервера

Для подключения OPC UA-сервера к **MasterSCADA 3.x** следует:

1. Запустить [MasterSCADA 3.x](#) и создать новый или открыть существующий проект.
2. Нажать **ПКМ** на узел **Система** и добавить **Компьютер** (если он отсутствует в проекте).

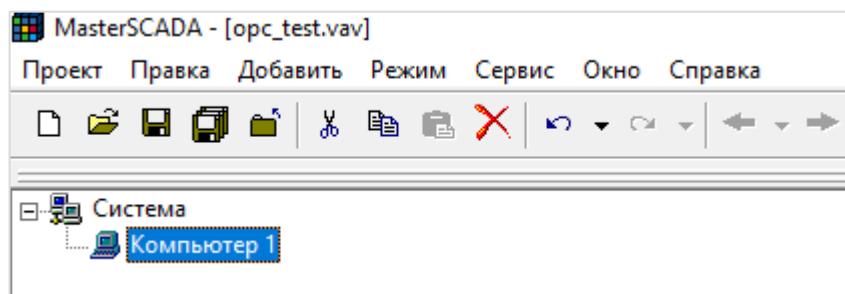


Рисунок 5.2.1 – Добавление компьютера в проект SCADA

3. Нажать **ПКМ** на узел **Компьютер** и использовать команду **Вставить OPC UA сервер**:

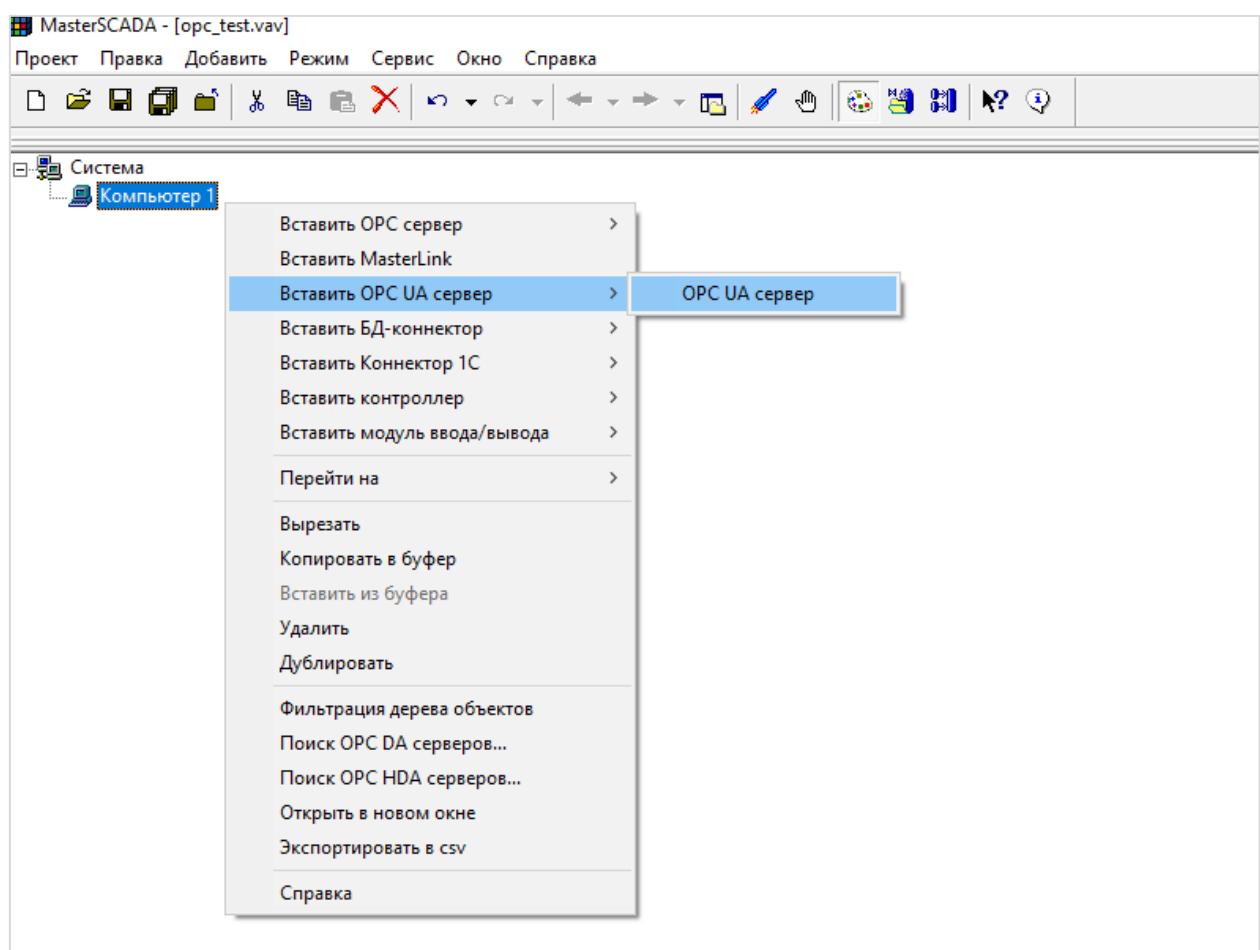


Рисунок 5.2.2 – Добавление OPC UA-сервера

4. Перейти на вкладку **Настройки**, нажать кнопку **Настройки** и указать IP-адрес контроллера и порт OPC UA-сервера (4840). Нажать кнопку **Ок**. В случае необходимости защищенного подключения на этой же вкладке можно выбрать настройки безопасности и аутентификации (эти же настройки должны быть сделаны в проекте ПЛК).

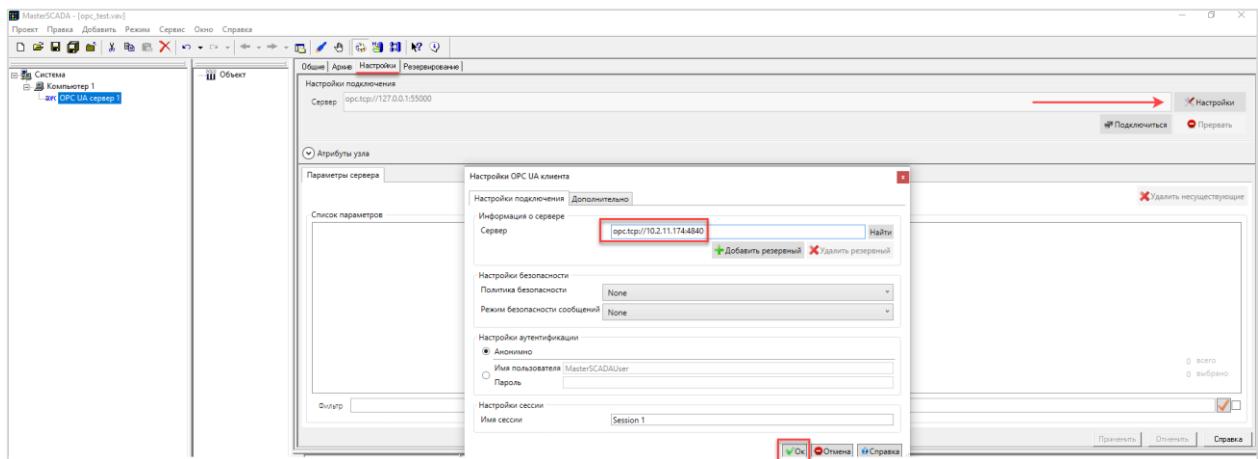


Рисунок 5.2.3 – Настройки подключения к OPC UA-серверу

5. Нажать кнопку **Подключиться**, дождаться импорта тегов, выделить нужные теги галочками и нажать **Применить**.

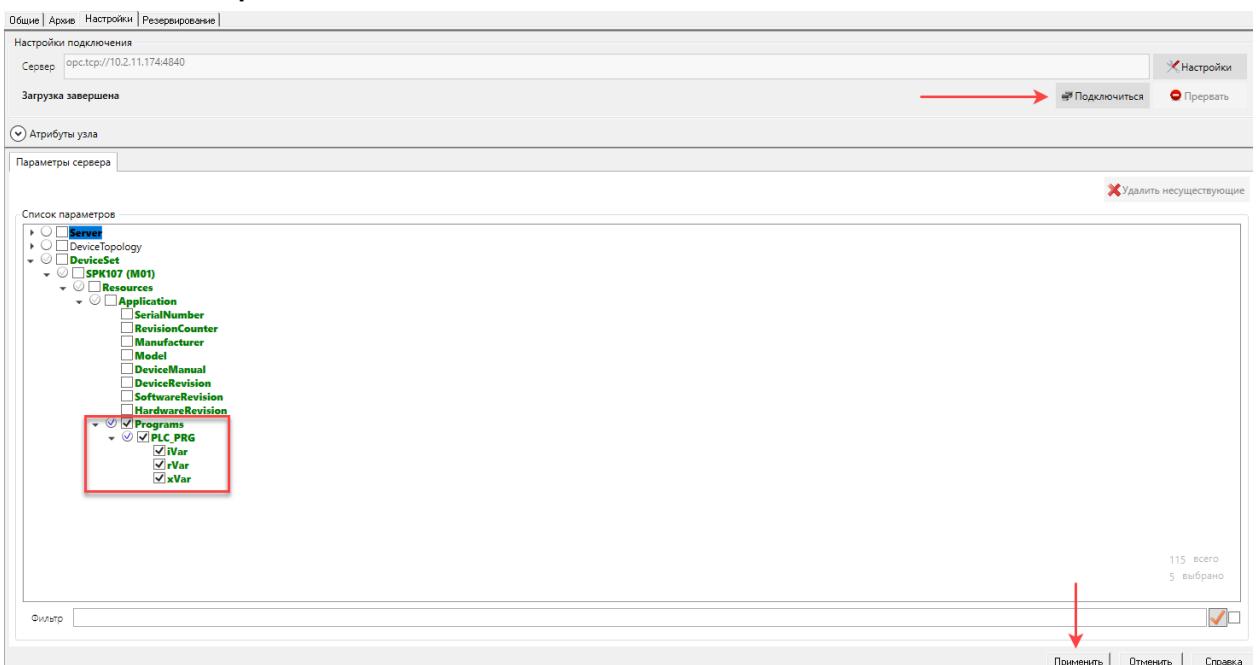


Рисунок 5.2.4 – Импорт тегов OPC UA-сервера

6. В результате теги OPC будут добавлены в дерево системы. Для проверки связи с OPC следует запустить на исполнение проект SCADA.

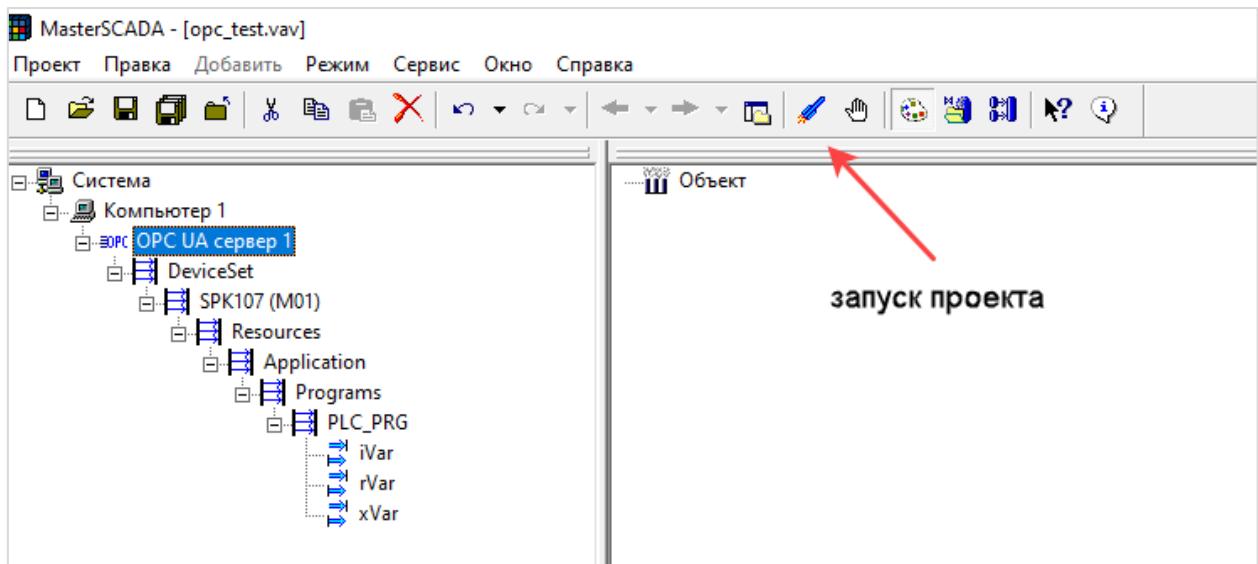


Рисунок 5.2.5 – Отображение добавленных тегов в дереве системы

Для изменения значения тега следует два раза нажать **ЛКМ** на его текущее значение.

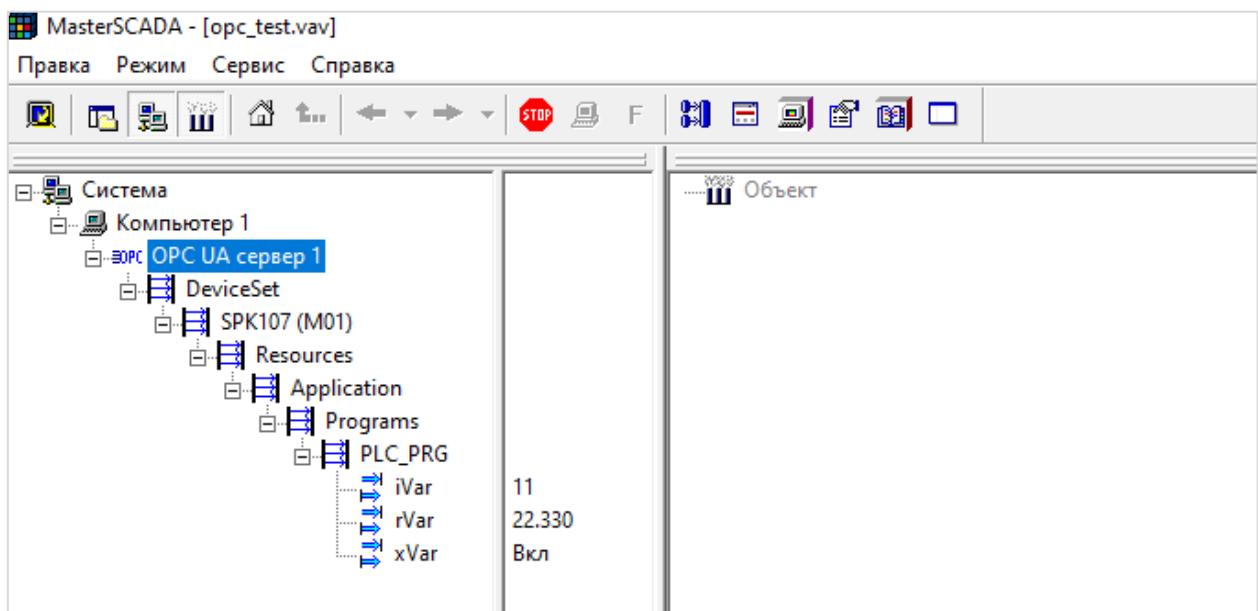


Рисунок 5.2.6 – Успешный обмен между SCADA и OPC-сервером