CODESYS V3.5

Настройка связи с верхним уровнем

Руководство пользователя

01.12.2018
версия 2.0

Оглавление

Глоссарий

ПЛК – программируемый логический контроллер.

ПК – персональный компьютер.

1 Цель документа. Способы связи с верхним уровнем

Настоящее руководство описывает настройку обмена данными с верхним уровнем АСУ (SCADA-системами) для контроллеров ОВЕН. 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 руководстве рассмотрены следующие функции **CODESYS V3.5**:

1. Использование web-визуализации. Данный компонент позволяет создавать экраны визуализации, с которыми можно будет работать на ПК (планшете, телефоне) через web-браузер. Клиент визуализации и контроллер должны находиться в одной сети (локальной или созданной с помощью VPN и др. технологий). Если пользователю не требуется «сложная» визуализация и обработка/хранение данных на ПК, то можно использовать web-визуализацию вместо SCADA-системы.

   Преимущества:
   - легкость настройки и простота использования;
   - не требуется использования дополнительного ПО;
   - бесплатность (требуется контроллер с поддержкой web-визуализации).

   Недостатки:
   - дополнительная нагрузка на контроллер;
   - возможности визуализации ограничены функционалом CODESYS;
   - отсутствие возможности обрабатывать и хранить данные на ПК.

2. Использование сетевых переменных. Данный компонент позволяет в несколько кликов настроить обмен данными между контроллерами, программируемыми в **CoDeSys V2.3/CODESYS V3.5** и находящимися в одной локальной сети.

   Преимущества:
   - легкость настройки и простота использования;
   - бесплатность (требуется контроллер с Ethernet).

   Недостатки:
   - все контроллеры, участвующие в обмене, должны программироваться в **CODESYS V3.5** или **CoDeSys V2.3.**

3. Передача данных в SCADA-систему с помощью OPC-серверов. Использование OPC-сервера позволяет собирать данные с различных устройств по разным протоколам обмена, после чего передавать их в SCADA-систему для обработки и визуализации. В рамках документа рассмотрена связь контроллера с четырьмя OPC-серверами: CODESYS OPC Server V3, Master OPC Universal Modbus Server, Lectus Modbus OPC/DDE Server и ОВЕН OPC (новый).

   Преимущества:
   - интегрированный комплекс ПО (OPC + SCADA) для сбора данных с различных устройств с последующим отображением, обработкой и архивированием;
   - поддержка различных протоколов, наличие готовых конфигураций приборов.

   Недостатки:
   - в большинстве случаев требуется приобретение соответствующего ПО;
   - сложность настройки;
   - требуется наличие АРМ.

4. Передача данных в облачный сервис OwenCloud. Облачный сервис применяется для удаленного мониторинга, управления и хранения архивов данных приборов, используемых в системах автоматизации. Подключение приборов к сервису осуществляется по интерфейсам **RS-485** (с помощью специальных сетевых шлюзов) или **Ethernet** (в этом случае требуется подключение приборов к сети с доступом к Интернету).

   Преимущества:
   - легкость настройки и простота использования;
   - основные функции бесплатны;
   - доступ к данным из любой точки мира через web-интерфейс или мобильное приложение;
   - рассылка аварийных сообщений и push-уведомлений.

   Недостатки:
   - - функционал сервиса уступает возможностям SCADA-систем.

2 Web-визуализация

Web-визуализация – это компонент **CODESYS V3.5**, который позволяет просматривать экраны визуализации контроллера в web-браузере.

ПРИМЕЧАНИЕ
Используемый web-браузер должен поддерживать **HTML5**.

Рисунок 2.1 – Пример работы с web-визуализацией в браузере

Web-визуализация автоматически включается в проект при добавлении компонента **Менеджер визуализации** (или создания первого экрана визуализации):

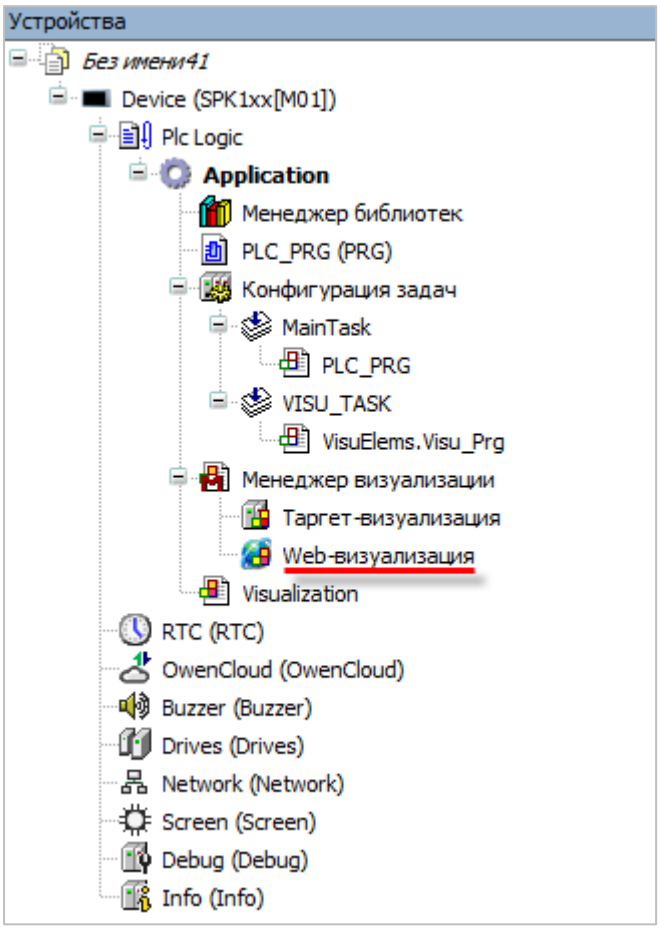


Рисунок 2.2 – Компонент web-визуализация в дереве проекта

В проект может быть добавлено несколько экземпляров компонента (например, в случае необходимости создания нескольких веб-страниц).

Компонент обладает следующими настройками:

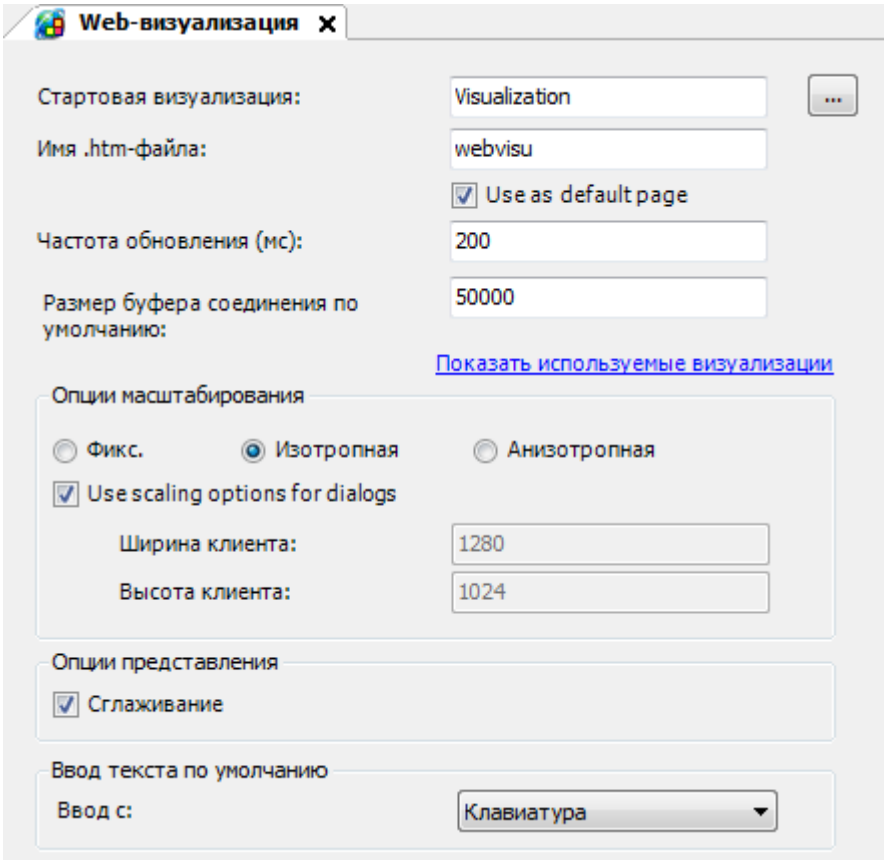


Рисунок 2.3 – Настройки компонента Web-визуализация

Стартовая визуализация – позволяет выбрать экран визуализации, который будет отображен клиентом web-визуализации при открытии данной веб-страницы.

Имя .htm-файла – название страницы, которое указывается в web-адресе визуализации. По умолчанию имя страницы – **webvisu**, адрес web-визуализации:

http://<IP-адрес контроллера>:8080/webvisu.htm

Use as default page – в случае наличия галочки в адресе может отсутствовать имя web-страницы:

http://<IP-адрес контроллера>:8080

Галочка может быть установлена только в одном из компонентов web-визуализации.

Частота обновления – позволяет настроить частоту обновления экранов визуализации на клиенте web-визуализации. Значение задается в миллисекундах.

Размер буфера соединения по умолчанию – максимальный размер буфера данных (в байтах), передаваемых клиенту web-визуализации. Рекомендуется не изменять значение этого параметра.

Подгонка размера – позволяет выбрать режим масштабирования экранов визуализации клиентом web-визуализации:

Фиксированная (Fixed) – позволяет задать фиксированный размер визуализации в пикселях;

Изотропная (Isotropic) – экран визуализации будет ***масштабироваться*** до размеров дисплея клиента web-визуализации ***с сохранением*** соотношения сторон;

Анизотропная (Anisotropic – экран визуализации будет ***масштабироваться*** до размеров дисплея клиента web-визуализации ***без сохранения*** соотношения сторон.

Use scaling options for dialog – если установлена галочка, то выполняется масштабирование диалогов (режим определяется настройкой **Подгонка размера)**.

Сглаживание – позволяет включить сглаживание элементов. Данная настройка улучшает внешний вид элементов, но может привести к падению производительности и нежелательным графическим артефактам (например, у всплывающих подсказок и невидимых элементов).

Ввод текста по умолчанию – позволяет определить основное устройство ввода для клиента web-визуализации – сенсорный экран или клавиатуру.

Подробная информация о разработке экранов визуализации для контроллеров ОВЕН приведена в документе **CODESYS V3.5. Визуализация**, доступном на диске с ПО из комплекта поставки и сайте компании ОВЕН в разделе **CODESYS V3/Документация**.

3 Сетевые переменные

3.1 Основные сведения о сетевых переменных

Сетевые переменные позволяют организовать обмен между несколькими контроллерами, программируемыми в **CODESYS V3.5**, по протоколу UDP, который работает поверх **Ethernet**. Соответственно, все контроллеры, участвующие в обмене, должны находиться в одной локальной сети. В настройках сетевого оборудования должна быть отключена блокировка UDP-пакетов.

Альтернативный вариант – организовать обмен по **Modbus TCP**. В данном случае пользователь должен добавлять в проект соответствующие компоненты (Ethernet, Modbus TCP Master, Modbus TCP Slave), настраивать их, разбираться в используемых функциях и адресации регистров. Преимуществом использования сетевых переменных является простота их настройки – достаточно создать на одном устройстве список читаемых/записываемых переменных и импортировать его на другом. В то же время, протокол **UDP** по сравнению с **TCP** обладает рядом недостатков (см., например, соответствующую статью на Wikipedia). Часть недостатков может быть компенсирована настройками **CODESYS** (контроль CRC, подтверждение получения).

В рамках каждого списка сетевых переменных, обмен происходит только в одном направлении. То есть у любого списка есть устройство-отправитель (в каждый момент времени – только одно) и устройства-получатели (их может быть несколько). Каждое устройство может содержать несколько списков отправляемых и несколько списков получаемых сетевых переменных.

Связь между устройством-отправителем и устройством-получателем определяется следующими параметрами:

1. **Порт**, через который осуществляется передача UDP-пакетов.
2. **Адрес рассылки** – пул адресов, на которые отправляются UDP-пакеты.
3. **Идентификатор списка** – номер используемого списка сетевых переменных.

Вышеперечисленные параметры должны быть идентичными для отправителя и всех получателей. Каждый из списков переменных устройства должен иметь уникальный идентификатор.

В случае масштабирования системы пользователю требуется только добавить на новые устройства соответствующие списки. Обмен сетевыми переменными может производиться по любой схеме маршрутизации (unicast, multicast, broadcast) или их комбинации.

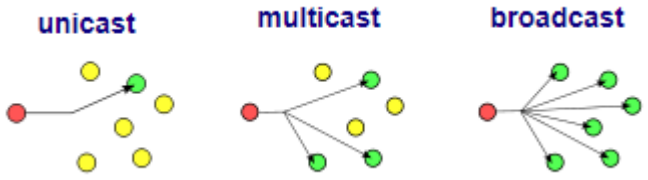


Рисунок 3.1 – Схемы маршрутизации UDP

CODESYS V3.5 позволяет в пределах одного проекта создавать программы сразу для нескольких контроллеров, что также упрощает процесс разработки.

В п. 3.2 и п. 3.3 рассмотрены настройки компонентов Список сетевых переменных (отправитель) и Список сетевых переменных (получатель).

В п. 3.4 рассмотрен пример обмена сетевыми переменными между контроллерами, программируемыми в **CoDeSys V2.3** и **CODESYS V3.5**

В п. 3.5 рассмотрен пример обмена сетевыми переменными между контроллерами, программируемыми в **CODESYS V3.5**

В п. 3.6 описаны особенности использования сетевых переменных.

3.2 Добавление и настройка компонента «Список сетевых переменных (отправитель)»

Для добавления в проект компонента **Список сетевых переменных (отправитель)** следует в дереве проекта нажать **ПКМ** на узел **Application** и в контекстном меню выбрать команду **Добавление объекта**:

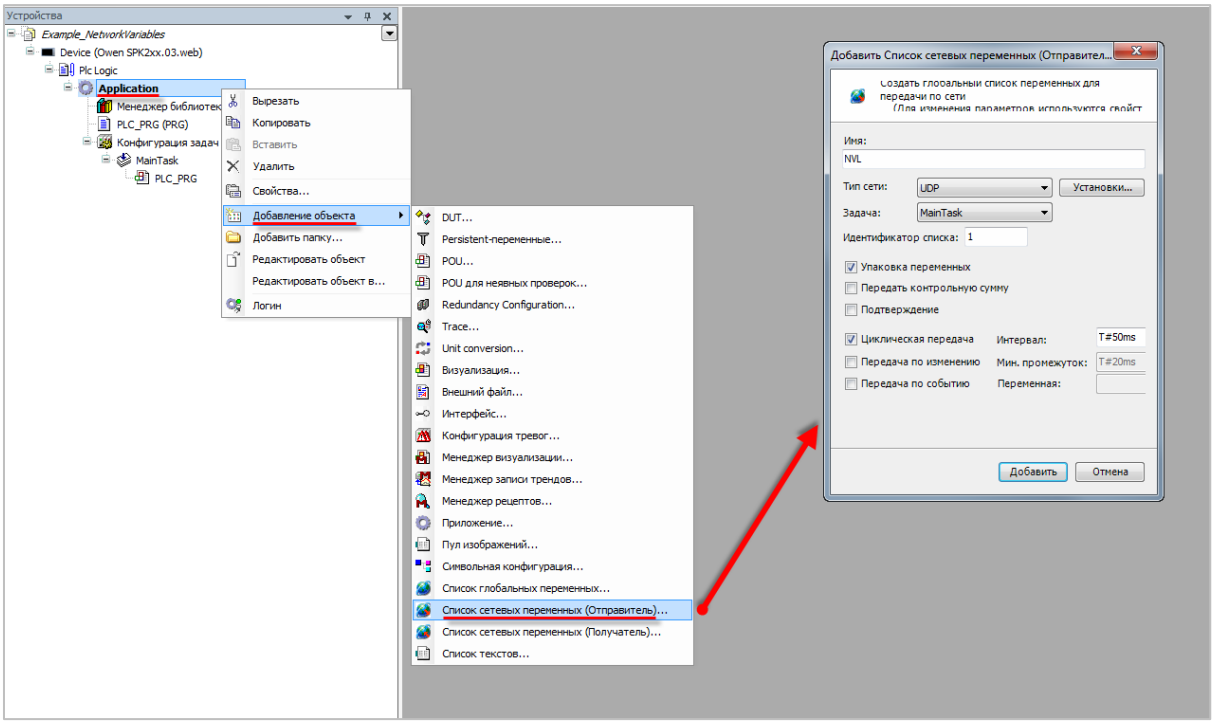


Рисунок 3.2 – Добавление компонента Список сетевых переменных (отправитель)

После создания списка в проект будет автоматически добавлена библиотека **NetVarUdp**:

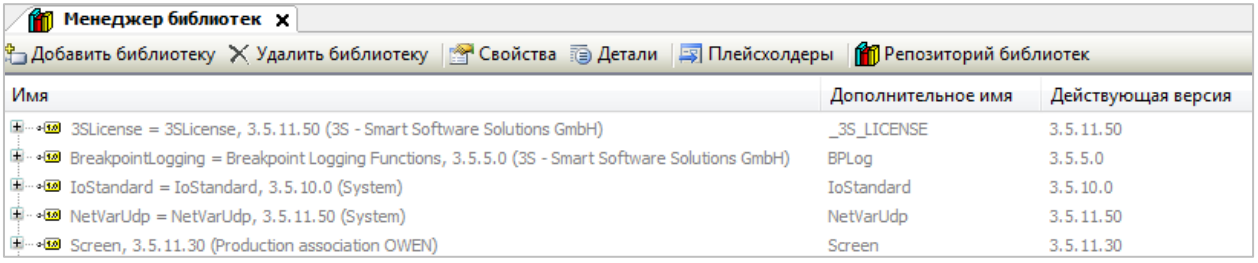


Рисунок 3.3 – Библиотека NetVarUdp в Менеджере библиотек

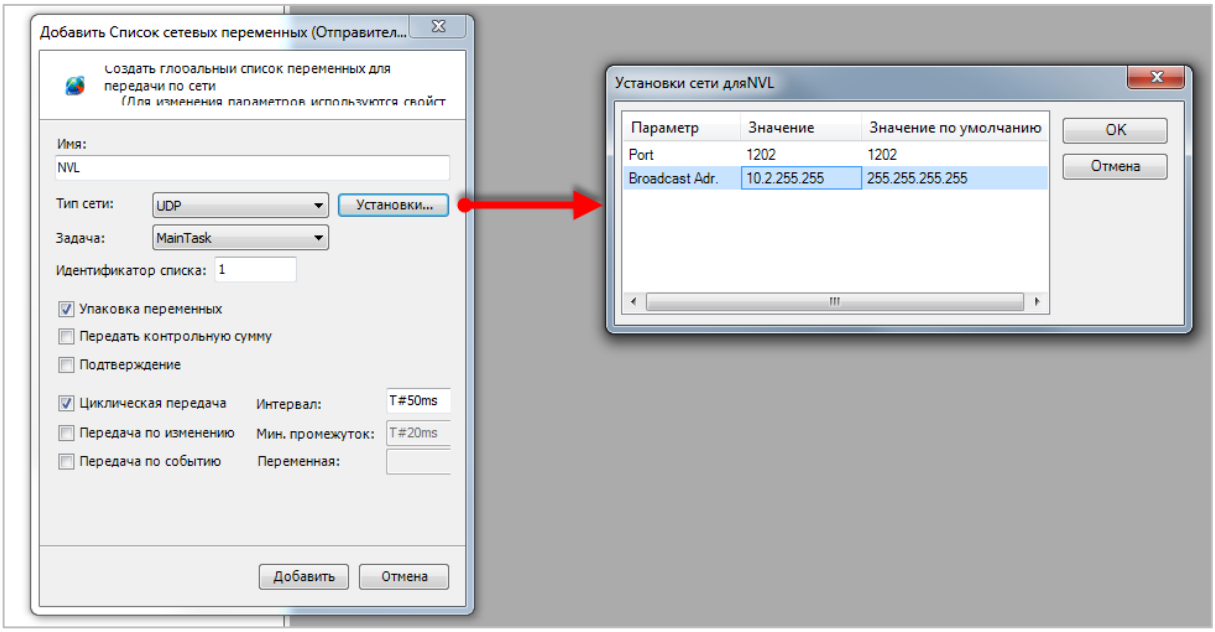


Рисунок 3.4 – Настройки компонента Список сетевых переменных (отправитель)

Настройки компонента:

1. **Тип сети** – протокол, используемый для передачи сетевых переменных. В данный момент поддерживается только протокол **UDP**.
2. **Установки** – в данном меню выбирается **порт** контроллера и **адрес широковещательной рассылки** (Broadcast address).

ПРИМЕЧАНИЕ
Рекомендуется использовать номер порта по умолчанию (**1202**).

ПРИМЕЧАНИЕ
Адрес рассылки должен соответствовать локальной сети.
Если контроллер имеет IP-адрес **10.2.11.10**, то адрес рассылки может быть задан как **10.2.255.255**. В данном случае получателем сетевых переменных может являться любое устройство с IP-адресом **10.2.x.x**.

ПРИМЕЧАНИЕ
В случае использования адреса рассылки по умолчанию (**255.255.255.255**) обмен сетевыми переменными будет невозможен.

3. **Задача** – задача, к которой привязан процесс обмена сетевыми переменными. Рекомендуется выбирать задачу с наименьшим временем цикла.
4. **Идентификатор списка** – номер данного списка.

ПРИМЕЧАНИЕ
В пределах одного устройства для каждого списка сетевых переменных (как отправляемых, так и получаемых) должен использоваться уникальный идентификатор.

5. **Упаковка переменных** – если галочка установлена, то переменные будут упаковываться в пакеты (датаграммы), размер которых будет определяться настройками сети. В противном случае каждая переменная отправляется отдельным пакетом.
6. **Передавать контрольную сумму** – если галочка установлена, то в пакет будет добавлена контрольная сумма. Устройство-получатель будет отбрасывать пакеты с несовпадающей контрольной суммой.
7. **Подтверждение передачи** – если галочка установлена, то отправитель будет ждать подтверждения получения на каждый отправленный пакет. Если подтверждение отсутствует, то в переменных диагностики будет выставлен соответствующий флаг.

Выбор режима передачи сетевых переменных:

8. **Циклическая передача** – в данном режиме сетевые переменные будут передаваться с заданным интервалом времени.
9. **Передача по изменению** – в данном режиме сетевые переменные будут передаваться в случае изменения их значений. Пользователь должен выбрать минимальный интервал времени между двумя передачами (т. е. если в пределах этого интервала значение переменной изменилось, то она все равно не будет отправлена до его истечения).
10. **Передача по событию** – в данном режиме сетевые переменные будут передаваться по переднему фронту заданной логической переменной.

ПРИМЕЧАНИЕ

При загрузке контроллера происходит сетевые переменные однократно отправляются вне зависимости от выполнения условий из пп. 8–10.

После создания списка следует наполнить его нужными переменными:

NVL

```
1  VAR_GLOBAL
2      wVar1: WORD;
3      rVar1: REAL;
4      aValues: ARRAY [0..1] OF INT;
5      Struct1: MyStruct;
6  END_VAR
```

Рисунок 3.5 – Объявление сетевых переменных

Если необходимо изменить настройки созданного списка, то следует нажать на него **ПКМ** и в контекстном меню выбрать пункт **Свойства**, после чего перейти на вкладку **Свойства сети**.

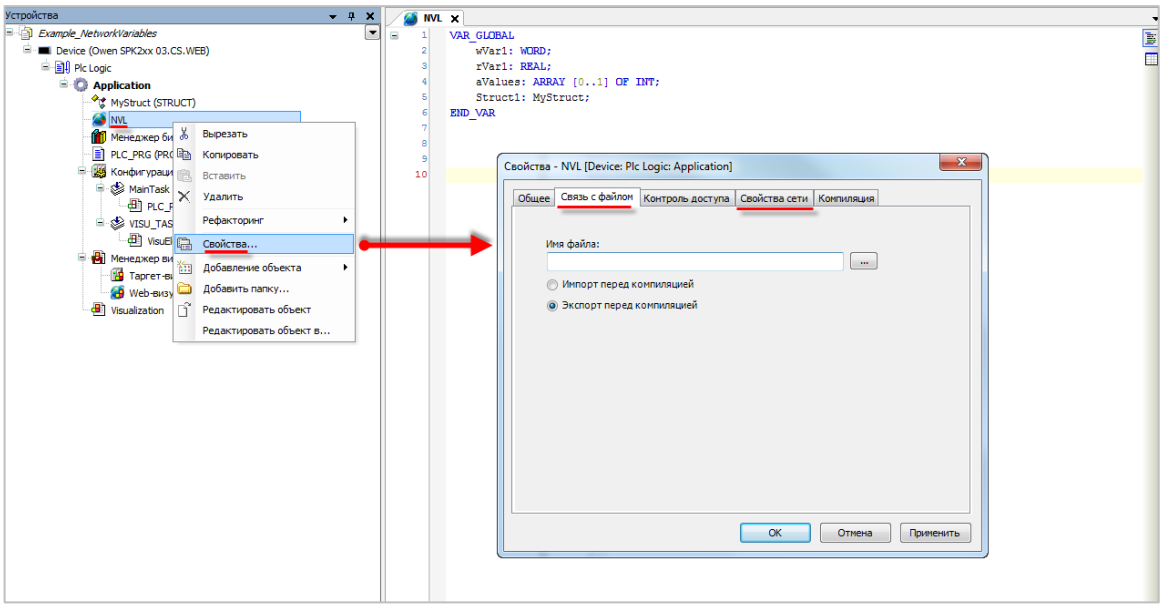


Рисунок 3.6 – Изменение настроек списка сетевых переменных

Во вкладке **Связь с файлом** можно указать путь к файлу, в который будет экспортирован (или из которого будет импортирован) список глобальных переменных. Экспорт/импорт происходит после компиляции проекта. Экспортированный список можно импортировать в компонент **Список сетевых переменных (получатель)** другого контроллера.

Экспортированный список представляет собой файл формата **.gvl**, который содержит сетевые переменные и сетевые настройки. Файл можно открыть любым текстовым редактором:

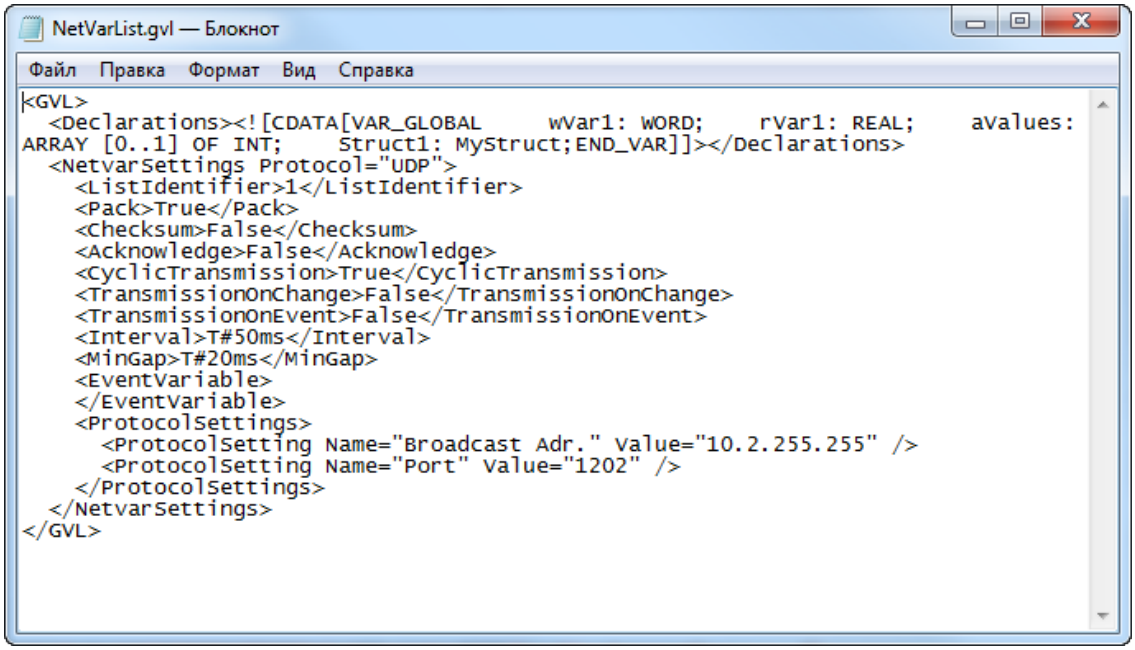


Рисунок 3.7 – Содержимое файла формата .gvl

3.3 Добавление и настройка компонента «Список сетевых переменных (получатель)»

Для добавления в проект компонента **Список сетевых переменных (получатель)** следует в дереве проекта нажать **ПКМ** на узел **Application** и в контекстном меню выбрать команду **Добавление объекта**:

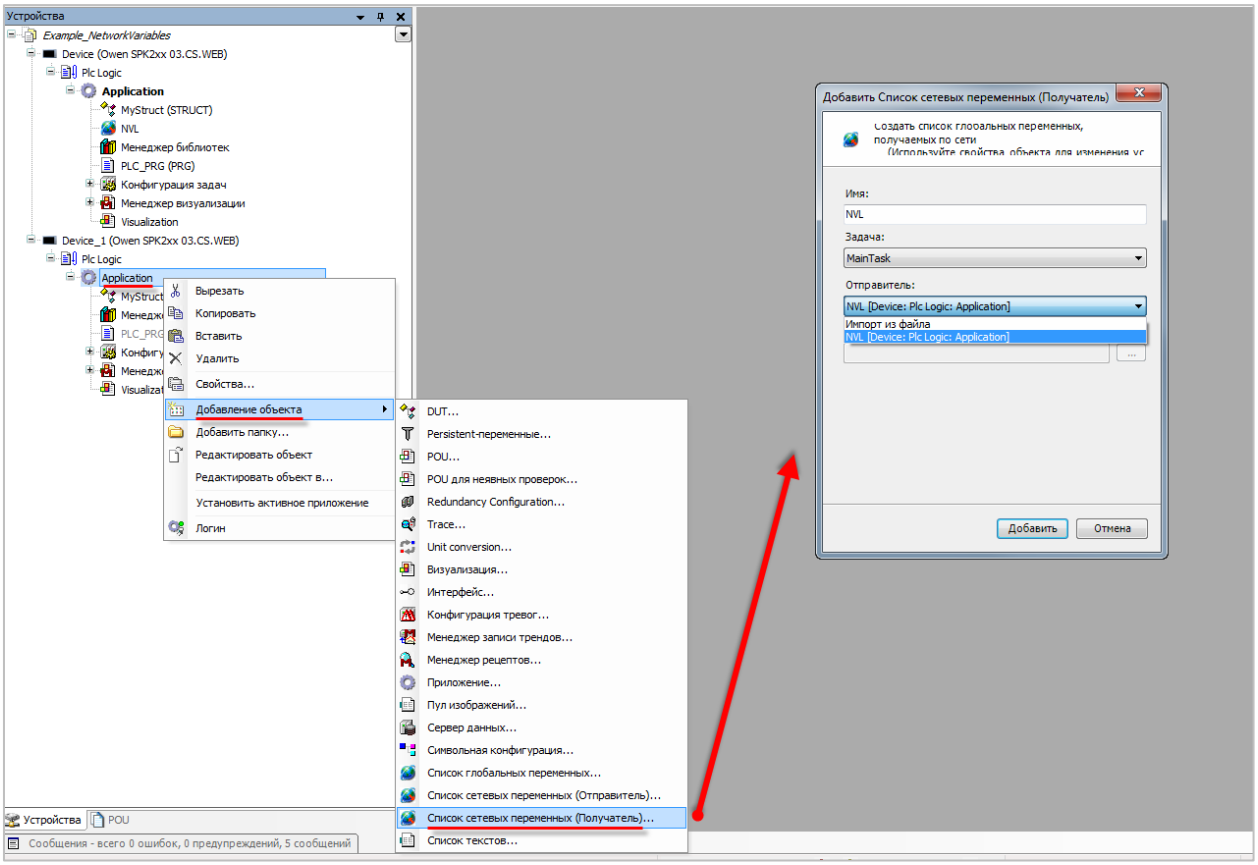


Рисунок 3.8 – Добавление компонента Список сетевых переменных (получатель)

При добавлении компонента пользователь должен указать, откуда будет импортирован список сетевых переменных, созданный на устройстве-отправителе – из другого устройства проекта или же из файла формата **.gvl** (см. рисунок 3.7).

В результате этого список отправителя (включая все сетевые настройки) будет импортирован на устройство-получатель. Никаких дополнительных настроек не требуется.

3.4 Настройка обмена сетевыми переменными между контроллерами, программируемыми в CoDeSys V2.3 и CODESYS V3.5

В качестве примера будет рассмотрен обмен сетевыми переменными между контроллерами **ПЛК110 [M02]** (программируется в **CoDeSys V2.3**) и **СПК1xx [M01]** (программируется в **CODESYS V3.5**).
Пример доступен для скачивания: Example_NetVars23_35.zip

Сетевые параметры и используемые переменные приведены в таблице 3.1.

Таблица 3.1 – Сетевые параметры и переменные примера

Параметр	СПК1xx [M01]	ПЛК110 [M02]
IP-адрес	10.2.11.170	10.2.20.172
Порт UDP	1202	
Broadcast адрес	10.2.255.255	
Идентификатор списка	1 (отправление) 2 (получение)	2 (отправление) 1 (получение)
Отправляемая сетевая переменная	wVar3523	wVar2335
Получаемая сетевая переменная	wVar2335	wVar3523

Для настройки обмена через сетевые переменные следует:

1. Создать новый проект для **ПЛК110 [M02]** в среде **CoDeSys V2.3** (язык программы не имеет значения, поскольку проект не будет содержать программы).
2. Во вкладке **Ресурсы** открыть узел **Настройки целевой платформы** и во вкладке **Сетевая функциональность** указать поддержку сетевого интерфейса **UDP**:

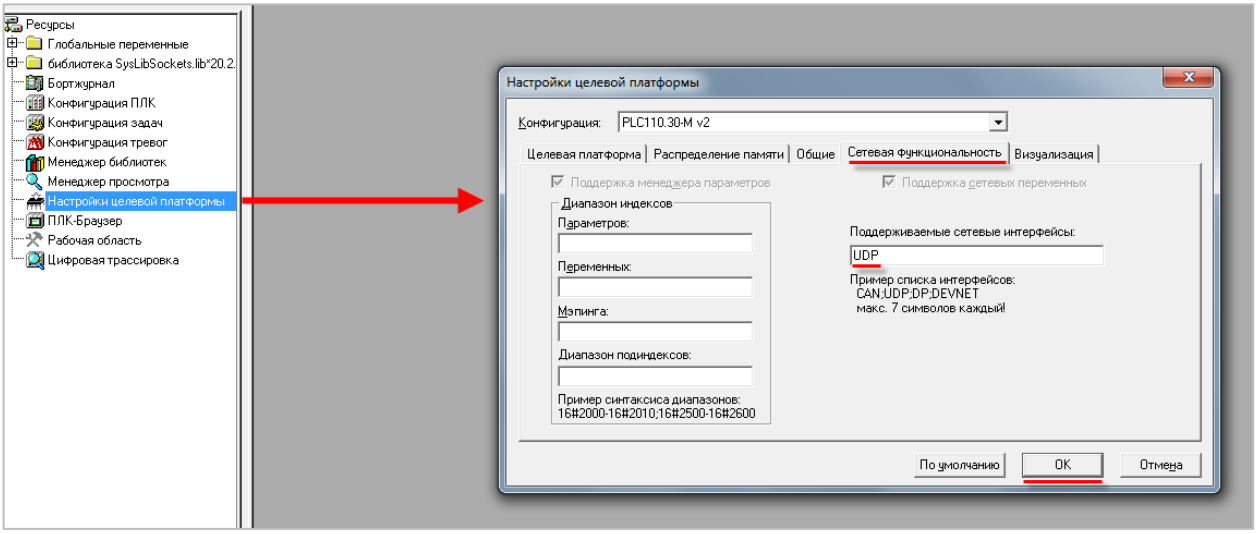


Рисунок 3.9 – Включение поддержки сетевых переменных в CoDeSys V2.3

3. Нажать **ПКМ** на папку **Глобальные переменные** и создать список получаемых сетевых переменных **From35_To23** с настройками в соответствие с таблицей 3.1:

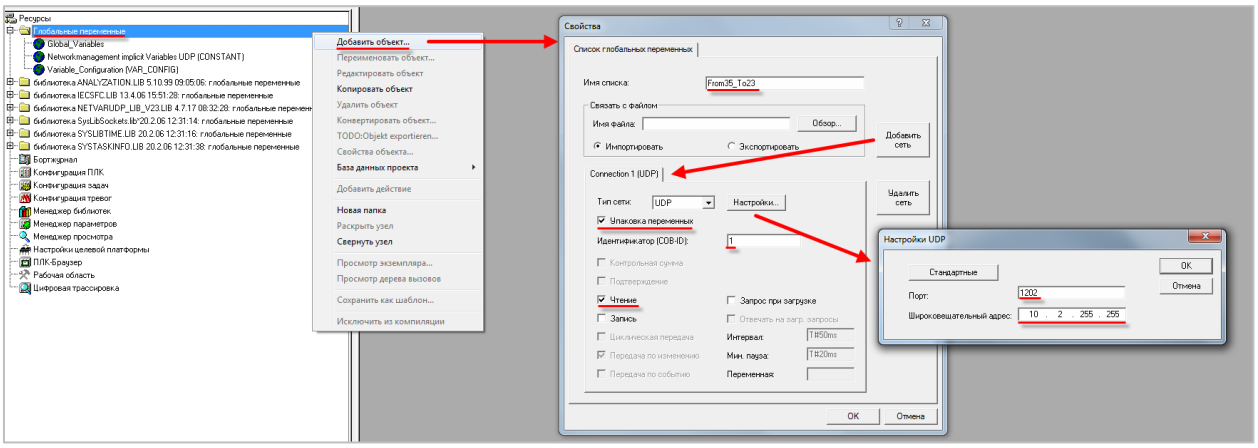


Рисунок 3.10 – Настройки списка получаемых сетевых переменных для ПЛК110 [M02]

В созданном списке объявить переменную **wVar3523** типа **WORD**:

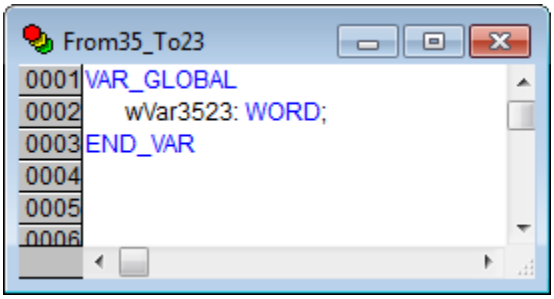


Рисунок 3.11 – Объявление получаемых сетевых переменных для ПЛК110 [M02]

4. Нажать **ПКМ** на папку **Глобальные переменные** и создать список отправляемых сетевых переменных **From23_To35** с настройками в соответствии с таблицей 3.1:

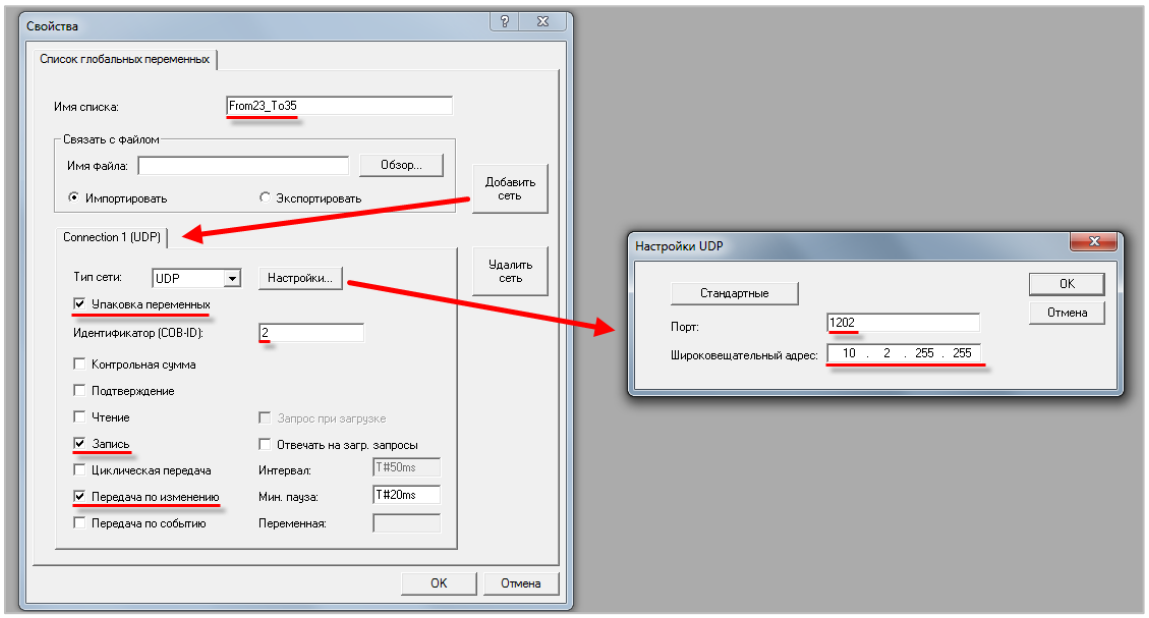


Рисунок 3.12 – Настройки списка отправляемых сетевых переменных для ПЛК110 [M02]

В созданном списке объявить переменную **wVar2335** типа **WORD**:

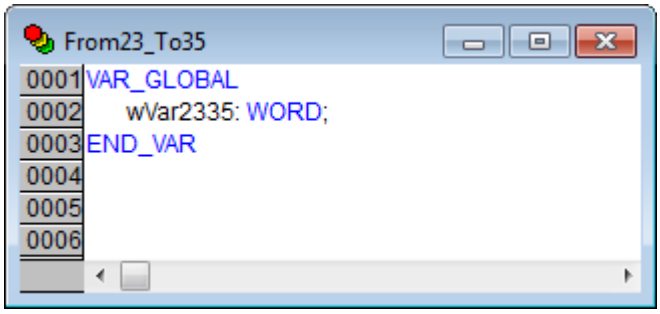


Рисунок 3.13 – Объявление отправляемых сетевых переменных для ПЛК110 [M02]

5. Создать новый проект для **СПК1xx [M01]** в среде **CODESYS V3.5** (язык программы не имеет значения, поскольку проект не будет содержать программы).
6. Добавить компонент Список сетевых переменных (отправитель) **From35_To23** с настройками в соответствии с таблицей 3.1:

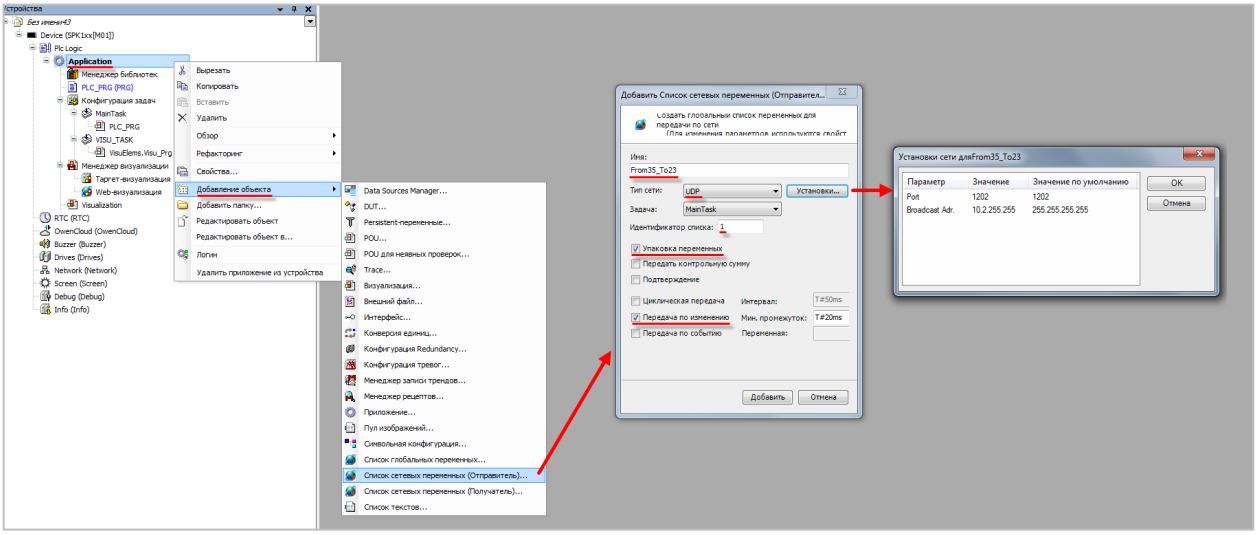


Рисунок 3.14 – Настройки списка отправляемых сетевых переменных для СПК1xx [M01]

В созданном списке объявить переменную **wVar3523** типа **WORD**:

Рисунок 3.15 – Объявление отправляемых сетевых переменных для СПК1xx [M01]

ПРИМЕЧАНИЕ

CODESYS V3.5 не поддерживает импорт списка получаемых сетевых переменных из CoDeSys V2.3. Ниже описана процедура обхода этого ограничения.

7. Добавить компонент Список сетевых переменных (отправитель) **From23_To23** с настройками, соответствующими одноименному списку из **CoDeSys V2.3**:

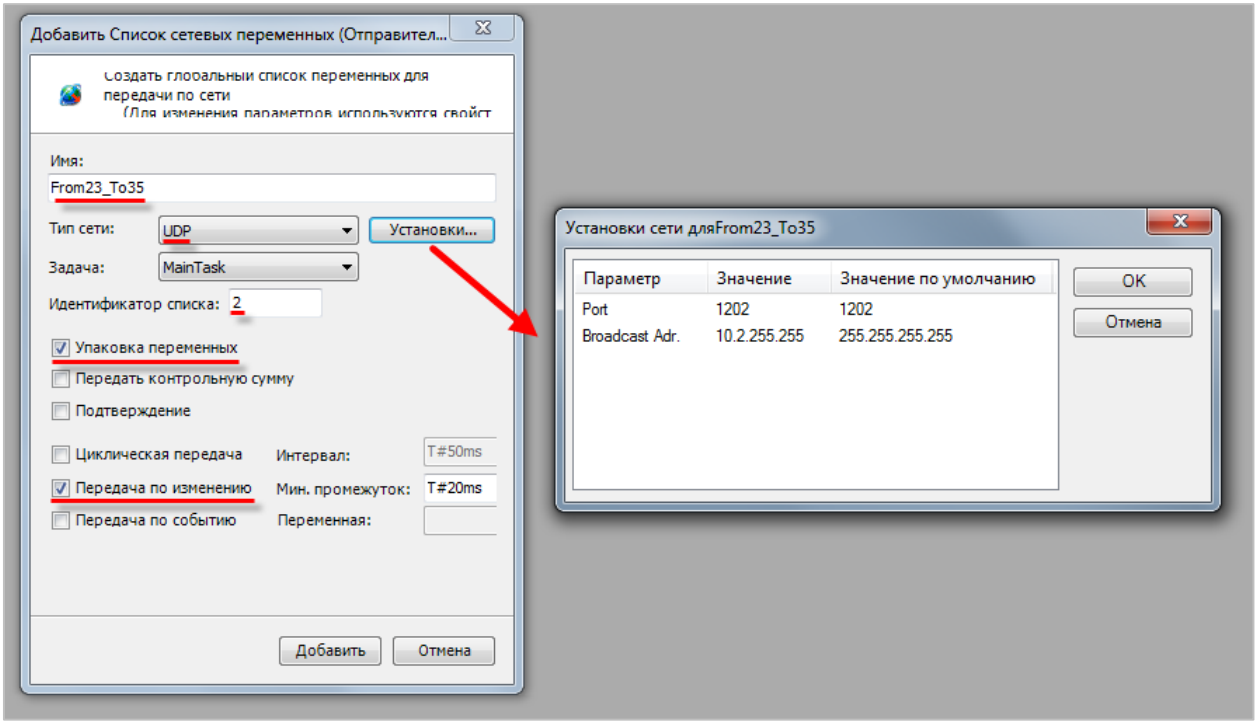


Рисунок 3.14 – Настройки «имитационного» списка отправляемых сетевых переменных

В созданном списке объявить переменную **wVar2335** типа **WORD**:

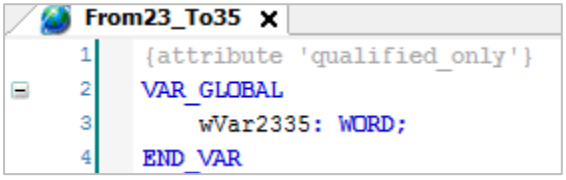


Рисунок 3.15 – Объявление отправляемых сетевых переменных

8. Нажать **ПКМ** на список сетевых переменных **From23_To35**, выбрать команду **Свойства** и во вкладке **Связь с файлом** указать имя экспортного файла сетевых переменных **From23_To35.gvl**:

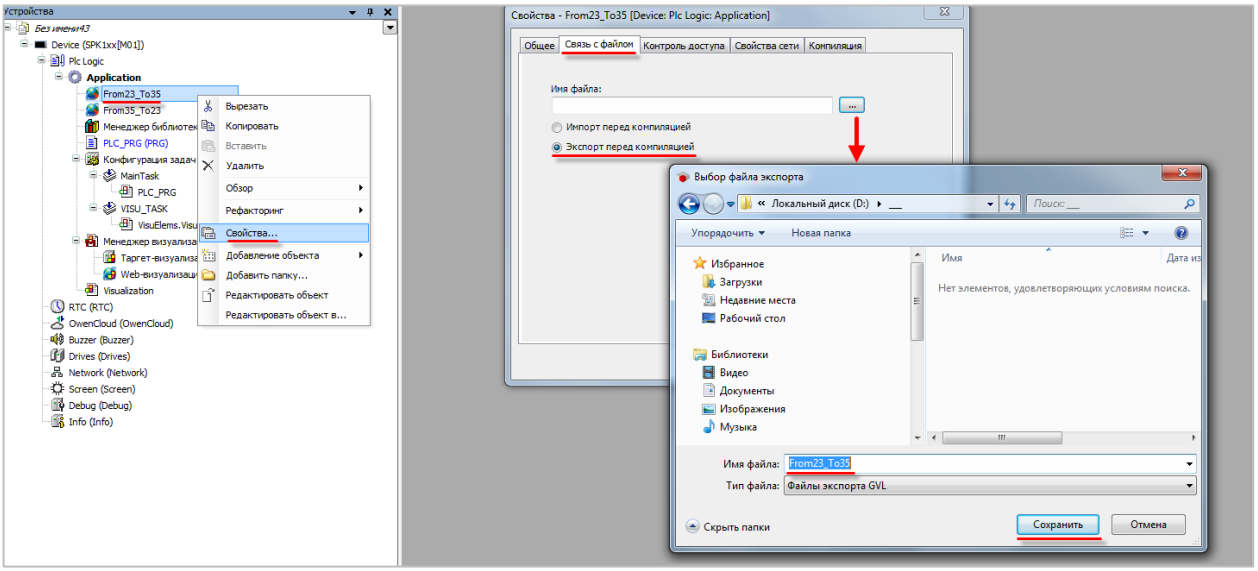


Рисунок 3.16 – Экспорт списка сетевых переменных

9. Удалить список сетевых переменных **From23_To35** из проекта **CODESYS V3.5**.
10. Добавить компонент Список сетевых переменных (получатель) с названием **From23_To23** и импортировать файл **From23_To35.gvl**, созданный в пп. 8:

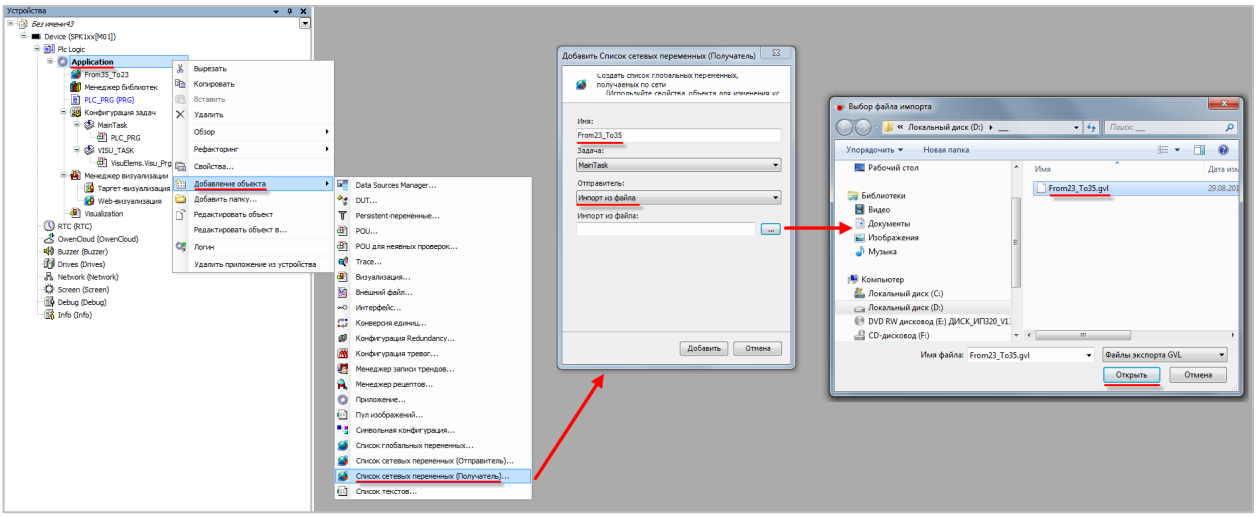


Рисунок 3.17 – Импорт списка сетевых переменных

11. Загрузить проекты в оба контроллера и запустить их. Убедиться, что оба контроллера подключены к одной локальной сети.

12. В проекте **CODESYS V3.5** в списке **From35_To23** изменить значение переменной **wVar3523**. Проверить, что оно изменилось в **CoDeSys V2.3**.

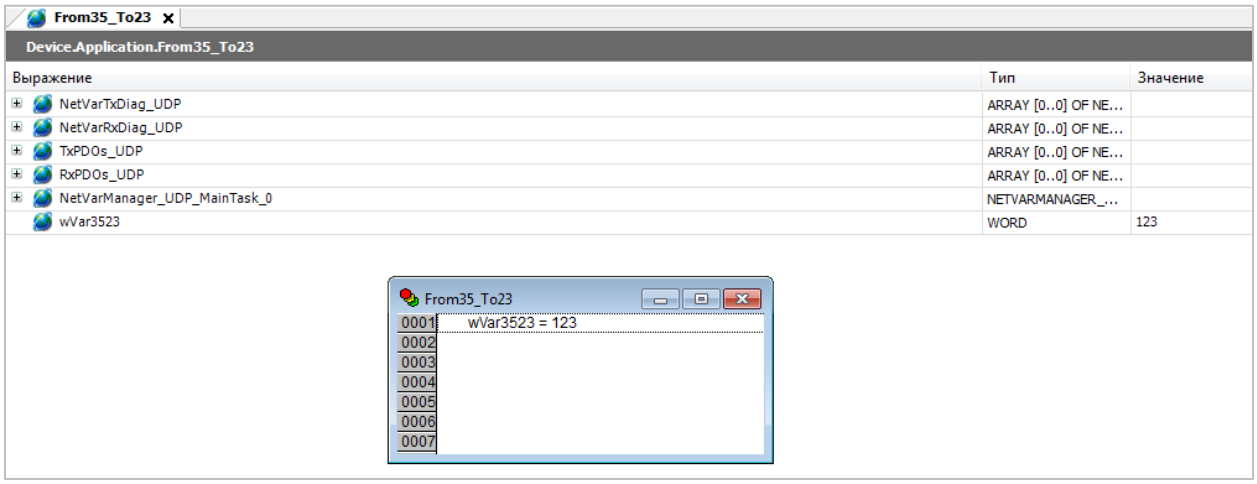


Рисунок 3.18 – Передача сетевых переменных из CODESYS V3.5 в CoDeSys V2.3

13. В проекте **CoDeSys V2.3** в списке **From23_To35** изменить значение переменной **wVar2335**. Проверить, что оно изменилось в **CODESYS V3.5**.

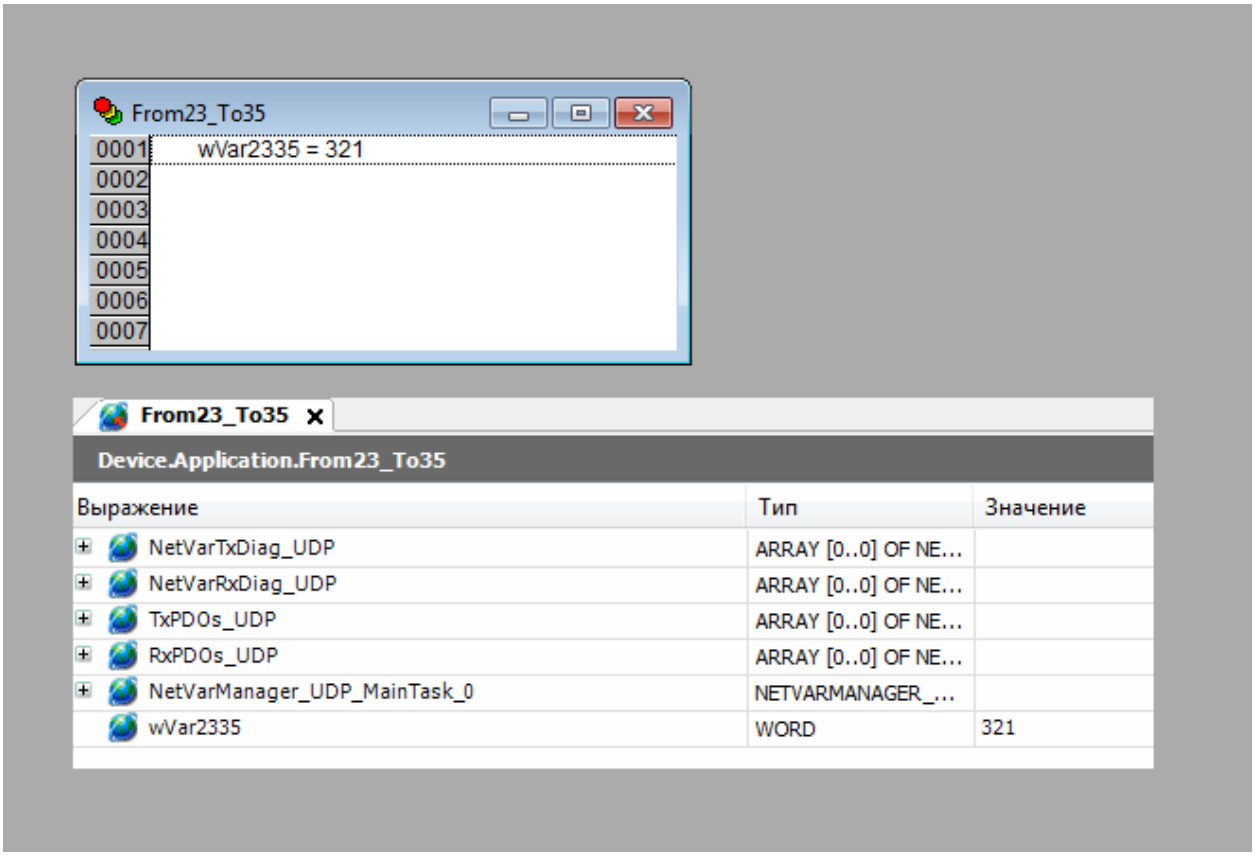


Рисунок 3.19 – Передача сетевых переменных из CoDeSys V2.3 в CODESYS V3.5

3.5 Настройка обмена сетевыми переменными между контроллерами, программируемыми в CODESYS V3.5

В качестве примера будет рассмотрен обмен сетевыми переменными между контроллером **СПК1xx [M01]** и виртуальным контроллером **CODESYS Control Win V3**.

Пример доступен для скачивания: Example_NetVars35_35.projectarchive

Сетевые параметры и используемые переменные приведены в таблице 3.2.

Таблица 3.2 – Сетевые параметры и переменные примера

Параметр	СПК1xx [M01]	CODESYS Control Win V3
IP-адрес	10.2.11.170	10.2.5.60
Порт UDP	1202	
Broadcast адрес	10.2.255.255	
Идентификатор списка	1 (отправление) 2 (получение)	2 (отправление) 1 (получение)
Отправляемая сетевая переменная	wSpkToWin	wWinToSpk
Получаемая сетевая переменная	wWinToSpk	wSpkToWin

Для настройки обмена через сетевые переменные следует:

1. Создать новый проект для **СПК1xx [M01]** в среде **CODESYS V3.5** (язык программы не имеет значения, поскольку проект не будет содержать программы).
2. Добавить компонент Список сетевых переменных (отправитель) **SpkToWin** с настройками в соответствии с таблицей 3.2:

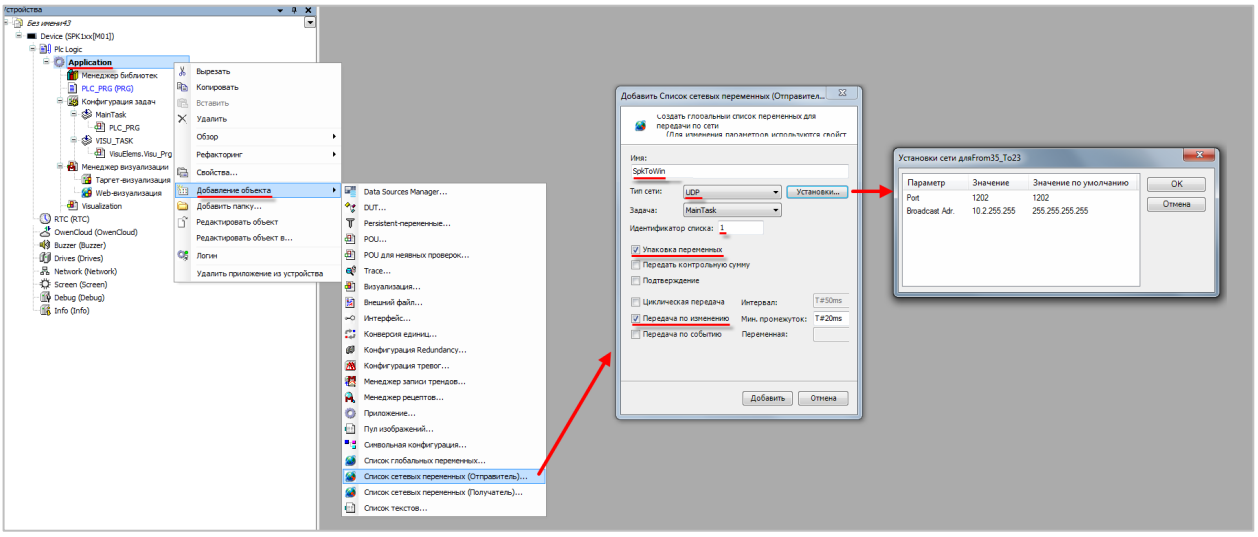


Рисунок 3.20 – Настройки списка отправляемых сетевых переменных для СПК1xx [M01]

В созданном списке объявить переменную **wSpkToWin** типа **WORD**:

```
SpkToWin
1  {attribute 'qualified_only'}
2  VAR_GLOBAL
3      wSpkToWin: WORD;
4  END_VAR
```

Рисунок 3.21 – Объявление отправляемых сетевых переменных для СПК1xx [M01]

3. Нажать **ПКМ** на узел **Device** и выполнить команду **Копировать**:

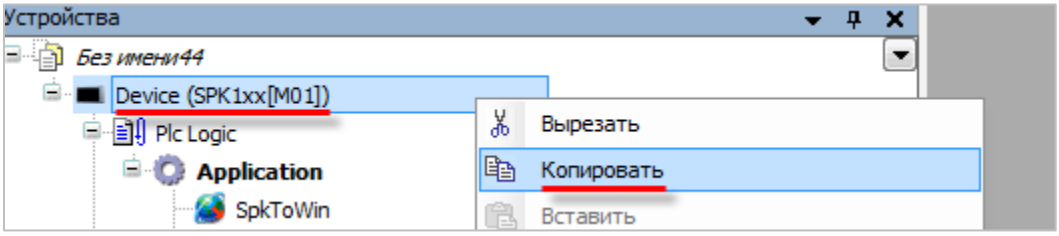


Рисунок 3.22 – Копирование устройства

4. Нажать **ПКМ** на название проекта и выполнить команду **Вставить**:

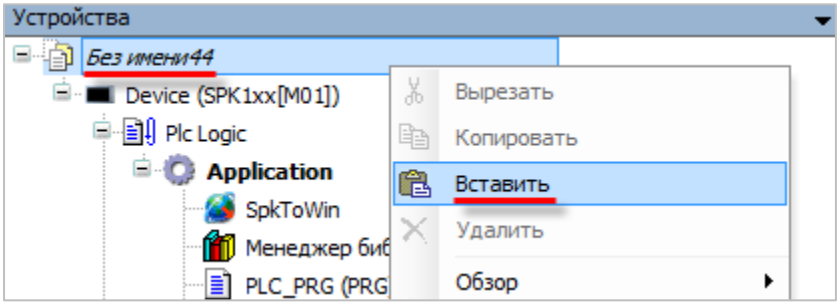


Рисунок 3.23 – Вставка устройства

5. В появившемся устройстве **Device_1** заменить таргет-файл СПК на таргет-файл виртуального контроллера **CODESYS Control Win V3** (**ПКМ** на узел **Device** – **Обновить устройство**).

Удалить в устройстве **Device_1** список сетевых переменных **SpkToWin**.

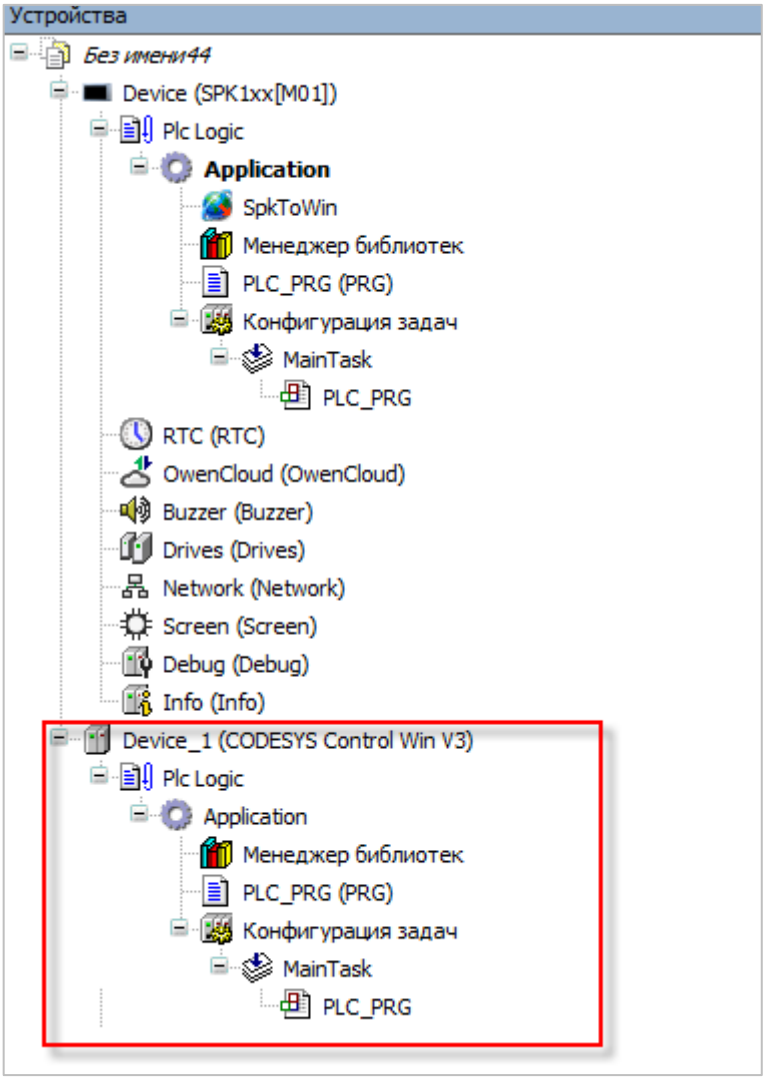


Рисунок 3.24 – Внешний вид устройства Device_1 в дереве проекта

6. В устройстве **Device_1** добавить компонент Список сетевых переменных (получатель) с импортом из устройства **Device**:

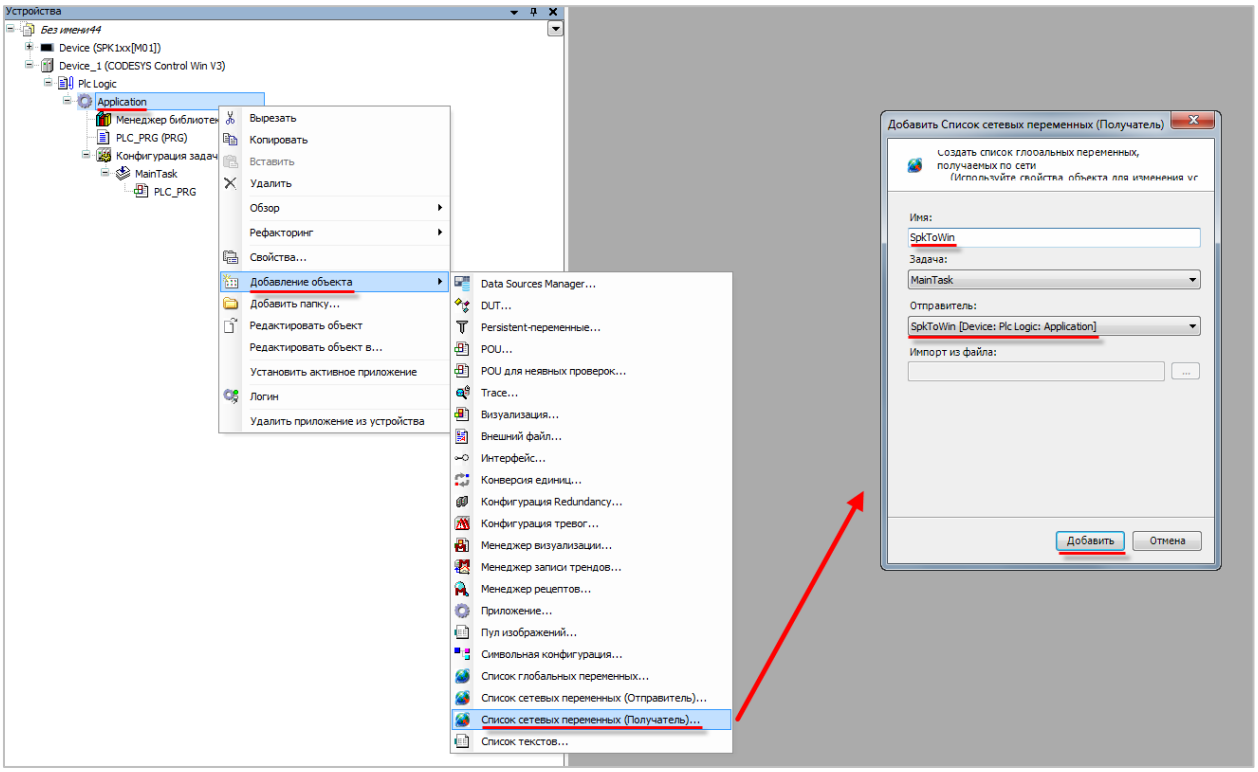


Рисунок 3.25 – Импорт сетевых переменных в устройство Device_1

7. В устройстве **Device_1** добавить компонент Список сетевых переменных (отправитель) **WinToSpk** с настройками в соответствии с таблицей 3.2:

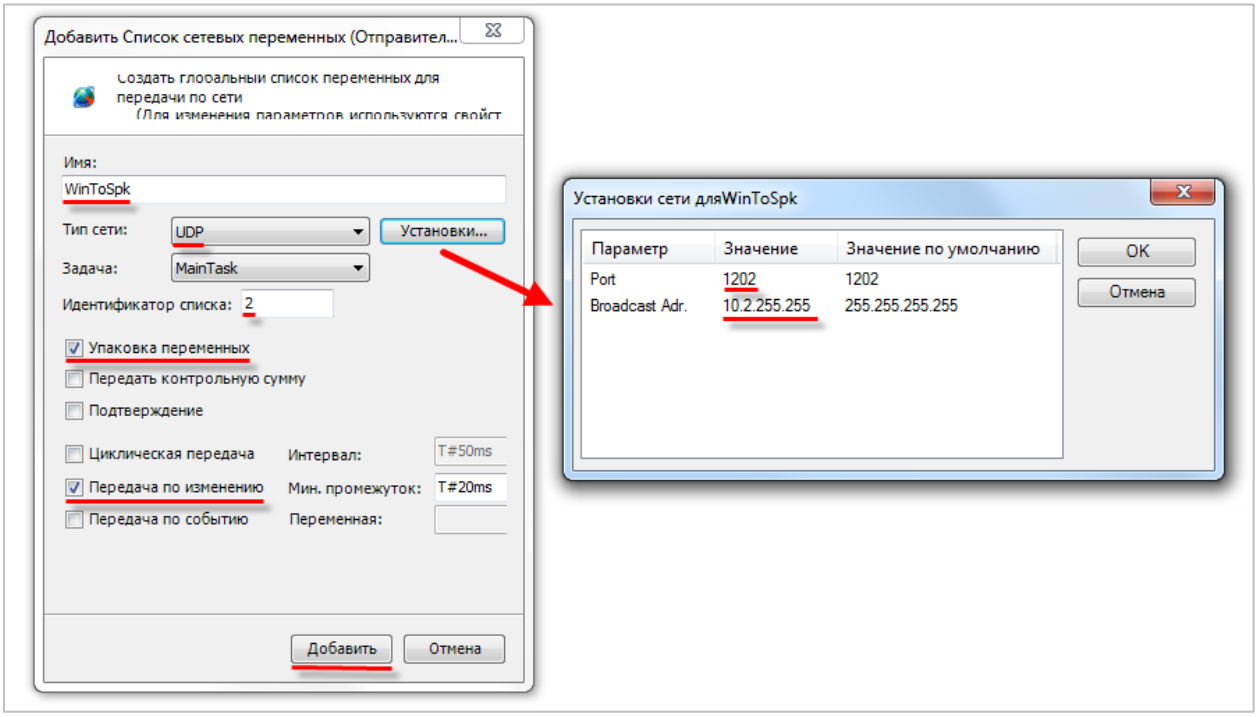


Рисунок 3.26 – Настройки списка отправляемых сетевых переменных для СПК1xx [M01]

В созданном списке объявить переменную **wWinToSpk** типа **WORD**:

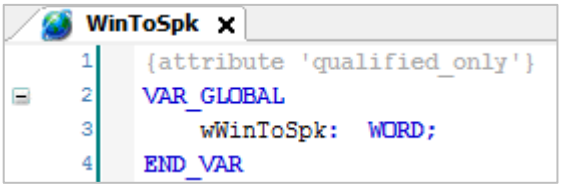


Рисунок 3.27 – Объявление отправляемых сетевых переменных для СПК1xx [M01]

8. В устройстве **Device** добавить компонент **Список сетевых переменных (получатель)** с импортом из устройства **Device_1**:

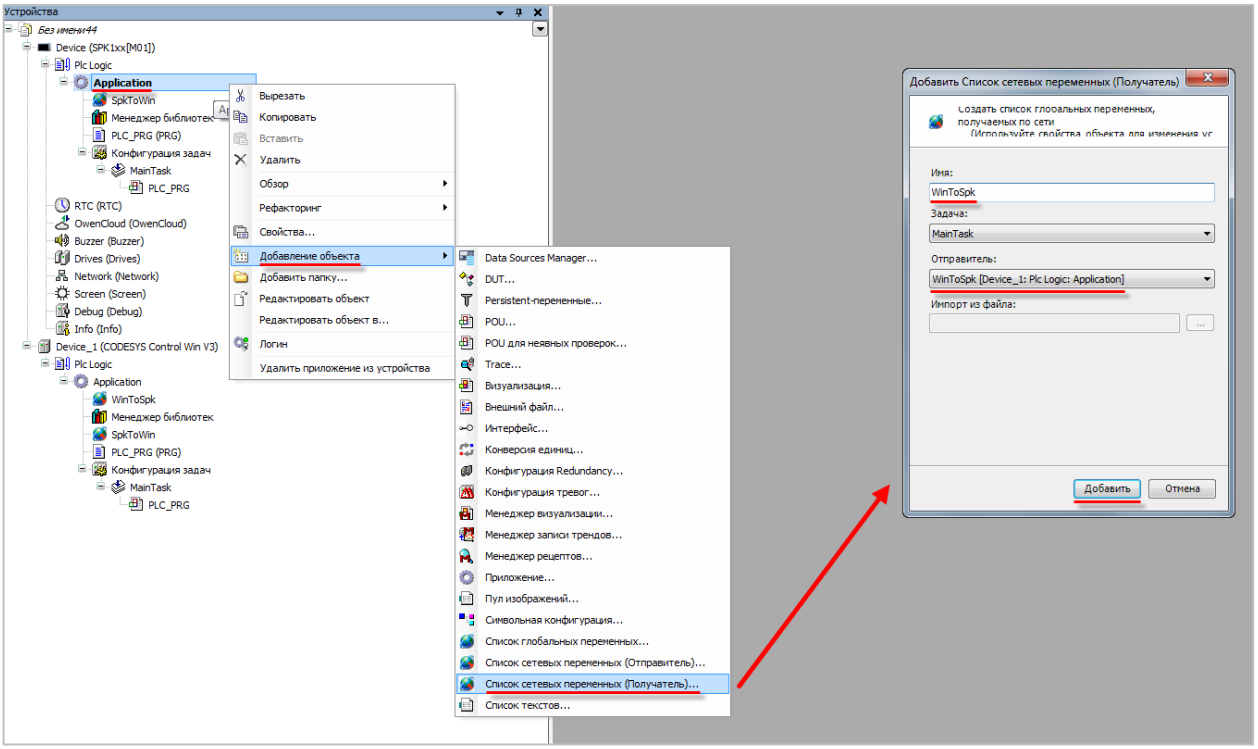


Рисунок 3.28 – Импорт сетевых переменных в устройство Device_1

9. Запустить виртуальный контроллер с помощью иконки в системном трее:

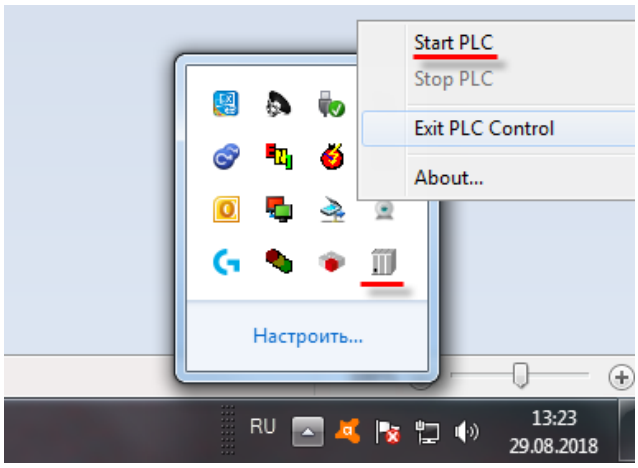


Рисунок 3.29 – Запуск виртуального контроллера

10. Загрузить проекты в оба устройства и запустить их. Чтобы сделать устройство активным следует нажать **ПКМ** на узел **Application** и выбрать команду **Установить активное приложение**:

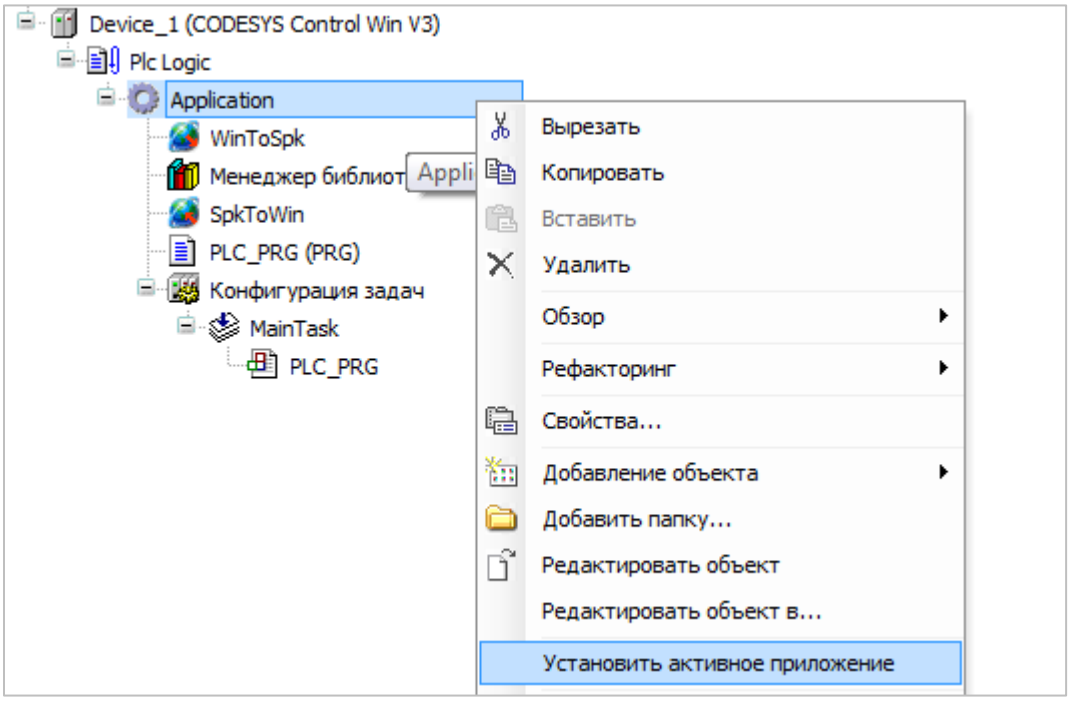


Рисунок 3.30 – Выбор активного приложение проекта

11. В устройстве **Device** в списке **SpkToWin** изменить значение переменной **wSpkToWin**. Проверить, что оно изменилось в устройстве Device_1.

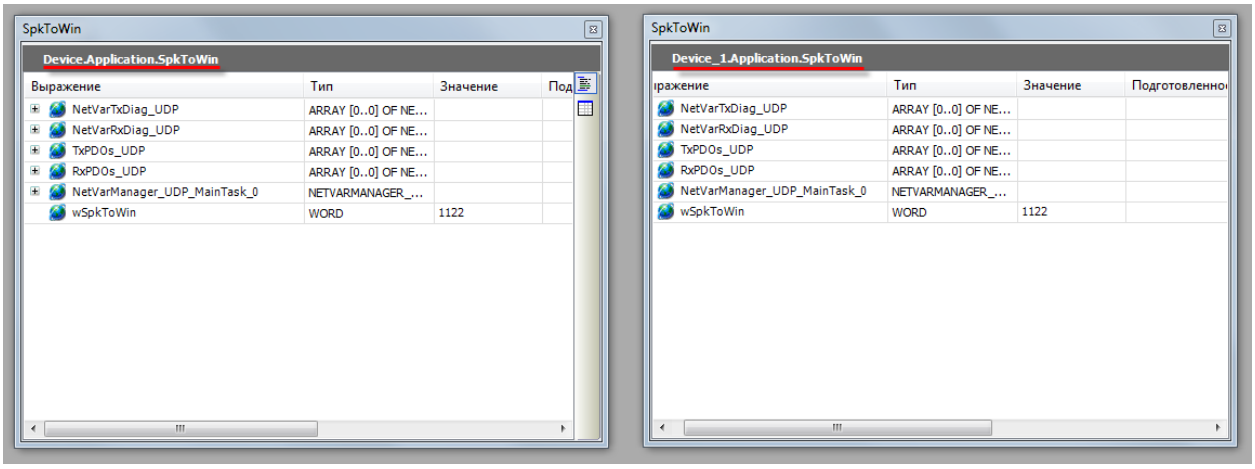


Рисунок 3.31 – Передача сетевых переменных из контроллера ОВЕН в виртуальный контроллер

12. Сделать активным устройство **Device_1** (см пп. 9) и в списке **WinToSpk** изменить значение переменной **wWinToSpk**. Проверить, что оно изменилось в устройстве **Device**.

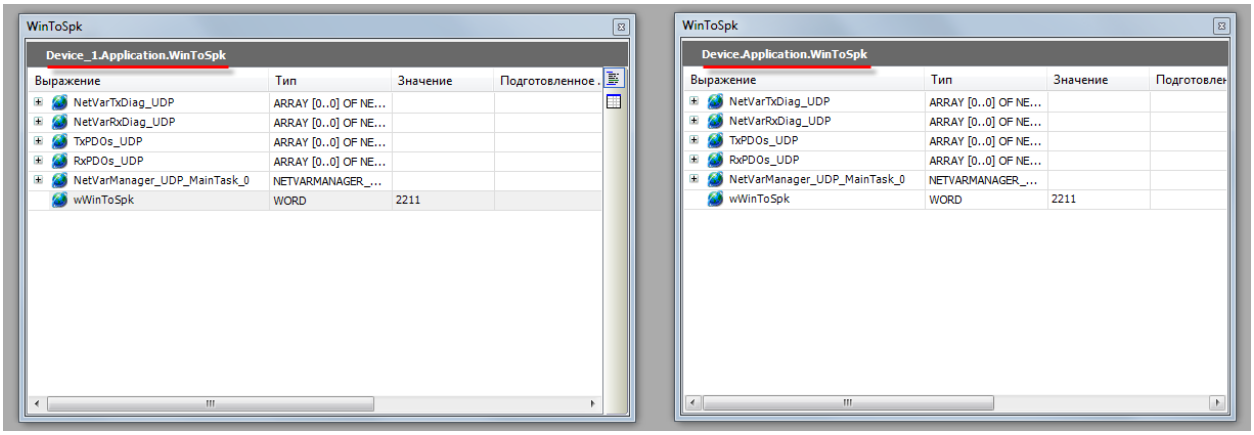


Рисунок 3.32 – Передача сетевых переменных из виртуального контроллера в контроллер ОВЕН

3.6 Особенности использования сетевых переменных

1. В случае использования адреса рассылки по умолчанию (**255.255.255.255**) обмен сетевыми переменными будет невозможен.
2. Каждый список сетевых переменных должен иметь уникальный идентификатор (ID).
3. Используемый для обмена сетевыми переменными порт не должен применяться для других целей и блокироваться сетевыми экранами (Firewall, Brandmauer).
4. Только одно приложение устройства может использовать сетевые переменные.
5. Границы передаваемых массивов должны быть определены только через литералы или константы (но не выражения).
6. Максимальный размер сетевой переменной – 255 байт.
7. Число сетевых переменных в проекте не ограничено.

4 Связь со SCADA-системой через OPC-сервер

4.1 Основные сведения об OPC

OPC – стандарт семейства программных технологий, предоставляющих единый интерфейс для управления объектами автоматизации и технологическими процессами. Одной из наиболее распространенных в настоящее время спецификаций является OPC DA (**d**ata **a**ccess), которая используется для обмена данными между различными устройствами.

Главной целью разработки стандарта OPC являлось обеспечение возможности интеграции средств автоматизации, функционирующих на разных платформах, в разных промышленных сетях и производимых различными фирмами. В настоящее время, OPC-сервер является неотъемлемым компонентом практически любой продвинутой АСУ и используется для сбора данных и их последующей передачи в SCADA-систему.

В настоящем руководстве рассматриваются вопросы подключения контроллеров ОВЕН, программируемых в CODESYS V3.5, к SCADA-системе MasterSCADA с использованием различных OPC-серверов:

- CODESYS V3 OPC (протокол **Gateway** поверх Ethernet);
- MasterOPC Universal Modbus Server (протокол **Modbus TCP**);
- Lectus Modbus OPC/DDE Server (протокол **Modbus TCP**);
- ОВЕН OPC (протокол **Modbus TCP**).

Сравнительные характеристики OPC-серверов приведены в таблице 4.1.

Типичная структурная схема обмена приведена на рисунке 4.1:

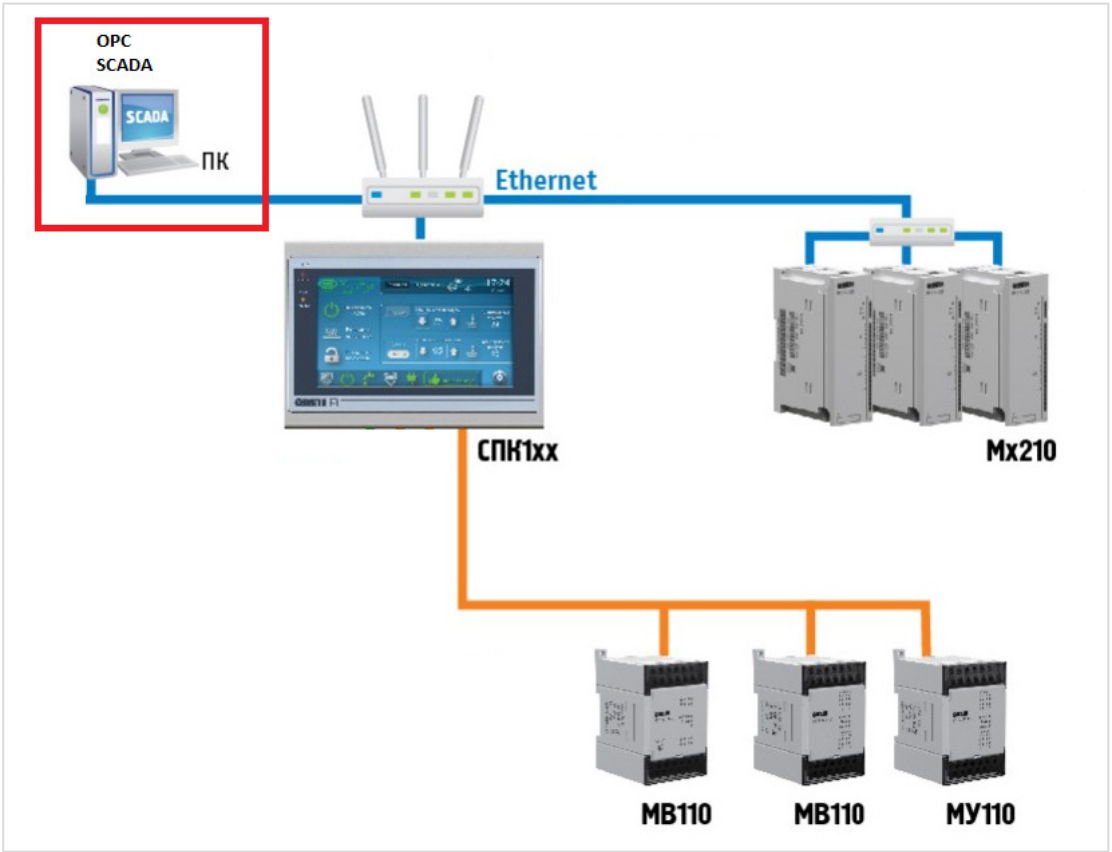


Рисунок 4.1 – Структурная схема связи контроллера и SCADA-системы через OPC-сервер

Таблица 4.1 – Сравнительные характеристики OPC-серверов

Функция	CODESYS OPC Server V3	MasterOPC Universal Modbus Server	Lectus OPC	ОВЕН OPC
Modbus RTU	-	+	+	+
Modbus ASCII	-	+	+	+
Modbus TCP	-	+	+	+
Чтение архивов ПЛК (20-я функция Modbus)	-	+	+	-
Визуальный контроль значений переменных	-	+	+	+
Поддержка скриптов	-	+	+	-
Работа с SQL-сервером	-	+	+	-
Поддержка OwenCloud	-	-	-	+
Готовые конфигурации для приборов ОВЕН	-	+-[1]	+-[1]	+
Экспорт таблицы переменных из OWEN Logic	-	-	-	+
Модель распространения	Бесплатный	Платный[2]	Платный	Бесплатный

Методика настройки обмена контроллера и SCADA-системы через OPC-сервер:

1. Настройка контроллера (настройка **символьной конфигурации** или настройка компонента **Modbus Slave**).
2. Настройка OPC-сервера.
3. Подключение OPC-сервера к SCADA-системе.

[1] Только для некоторых приборов.
[2] Доступна бесплатная версия на 32 тега

4.2 CODESYS OPC Server V3

4.2.1 Настройка контроллера

CODESYS OPC Server V3 – самый простой с точки зрения настройки OPC-сервер для организации обмена с контроллером, так как он интегрирован в среду разработки и позволяет автоматически импортировать переменные проекта.

Для настройки контроллера следует:

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

```
PLC_PRG
1  PROGRAM PLC_PRG
2  VAR
3      xVar: BOOL;    // логическое значение
4      iVar: INT;     // целое число
5      rVar: REAL;    // число с плавающей точкой
6  END_VAR
```

Рисунок 4.2.1 – Объявление переменных программы PLC_PRG

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

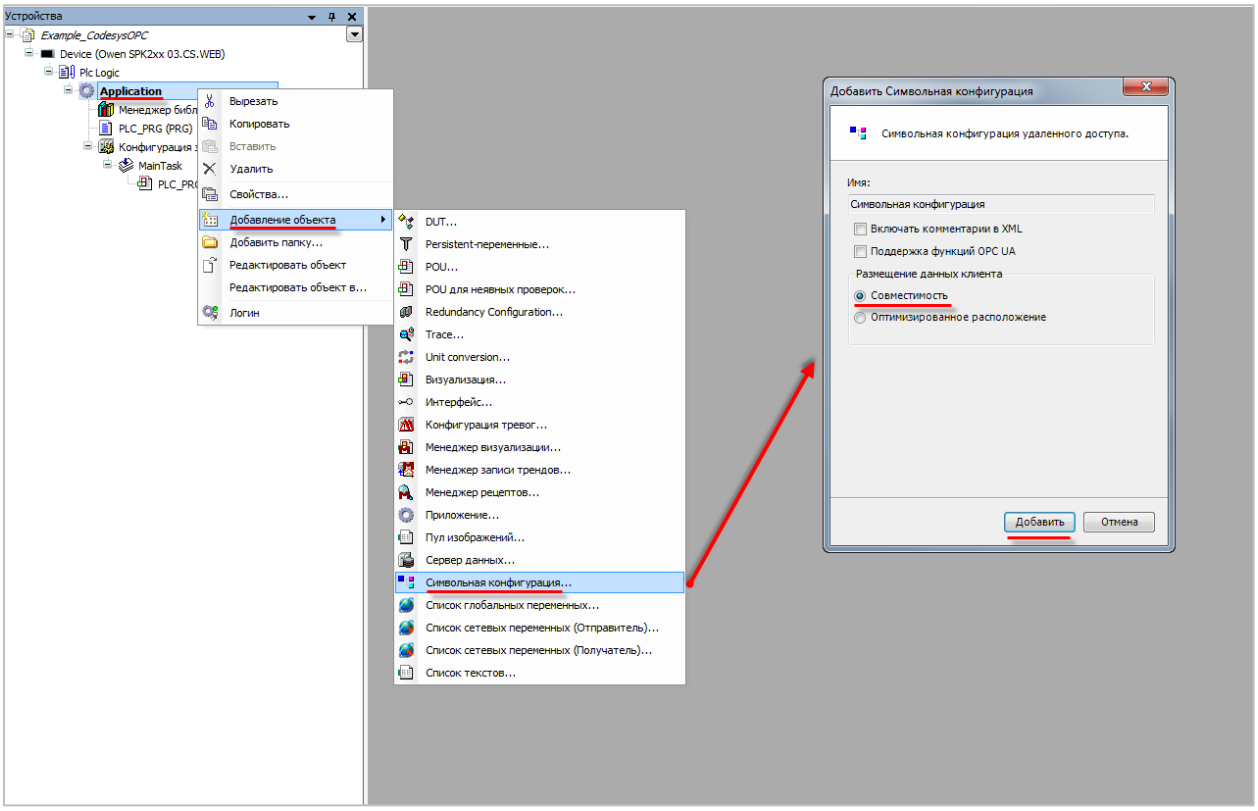


Рисунок 4.2.2 – Добавление компонента Символьная конфигурация

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 является дополнительным платным компонентом CODESYS, который ***в данный момент не поддерживается контроллерами ОВЕН***;

Размещение данных клиента – пользователь может выбрать структуру файла символьной конфигурации – совместимую со старыми версиями или оптимизированную. Оптимизированная структура поддерживается начиная с **CODESYS V3.5 SP7**. Рекомендуется всегда использовать совместимую структуру.

Дополнительные настройки компонента описаны в справке CODESYS.

4. После добавления компонента **Символьная конфигурация** следует выполнить компиляцию проекта:

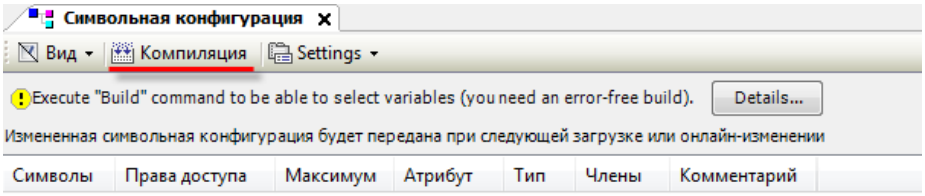


Рисунок 4.2.3 – Кнопка компиляции проекта после создания символьной конфигурации

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

5. Пометить галочками переменные, которые будут считываться/изменяться OPC-сервером и указать для каждой из них права доступа (со стороны OPC-сервера).
   Для прав доступа используются следующие пиктограммы:

   – только чтение;

   – только запись;

   – чтение/запись.

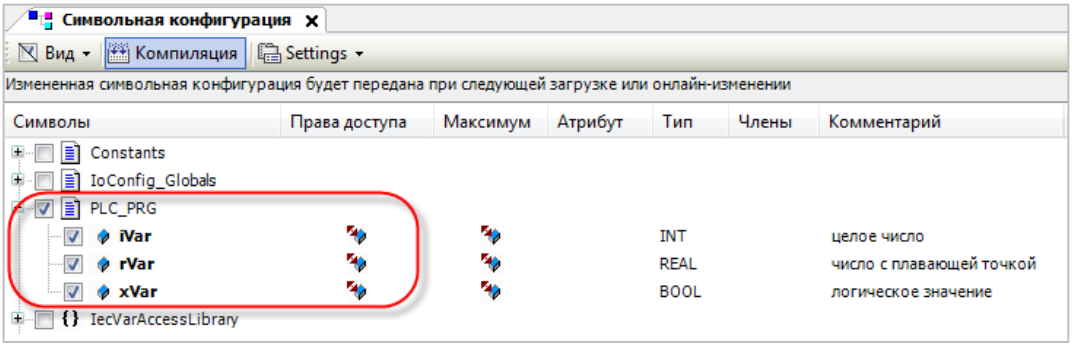


Рисунок 4.2.4 – Добавление компонента Символьная конфигурация

Настройка контроллера завершена. После загрузки проекта следует выполнить команду **Создать загрузочное приложение** из меню **Онлайн**.

4.2.2 Настройка OPC-сервера

Для настройки OPC-сервера следует:

1. Запустить приложение **OPC Configurator** (из **меню Пуск** или папки **CODESYS OPC Server V3**, расположенной в директории установки **CODESYS**).

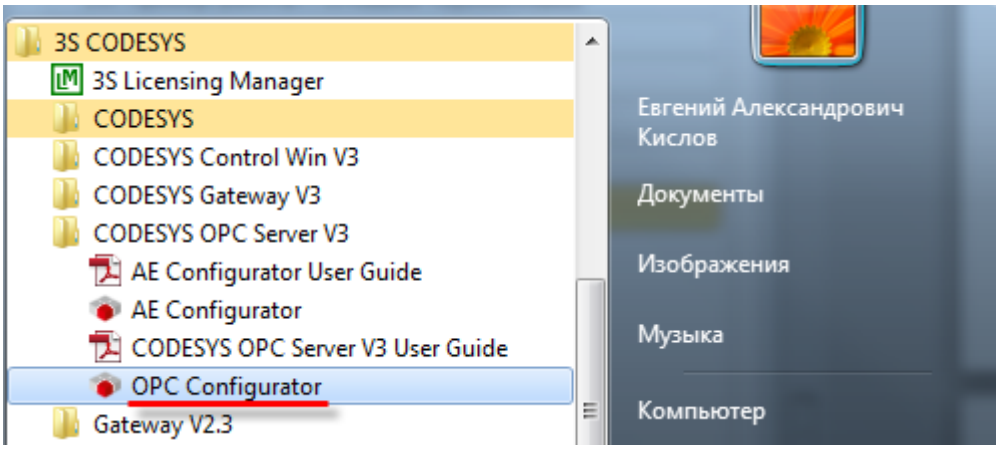


Рисунок 4.2.5 – Запуск приложения OPC Configurator

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

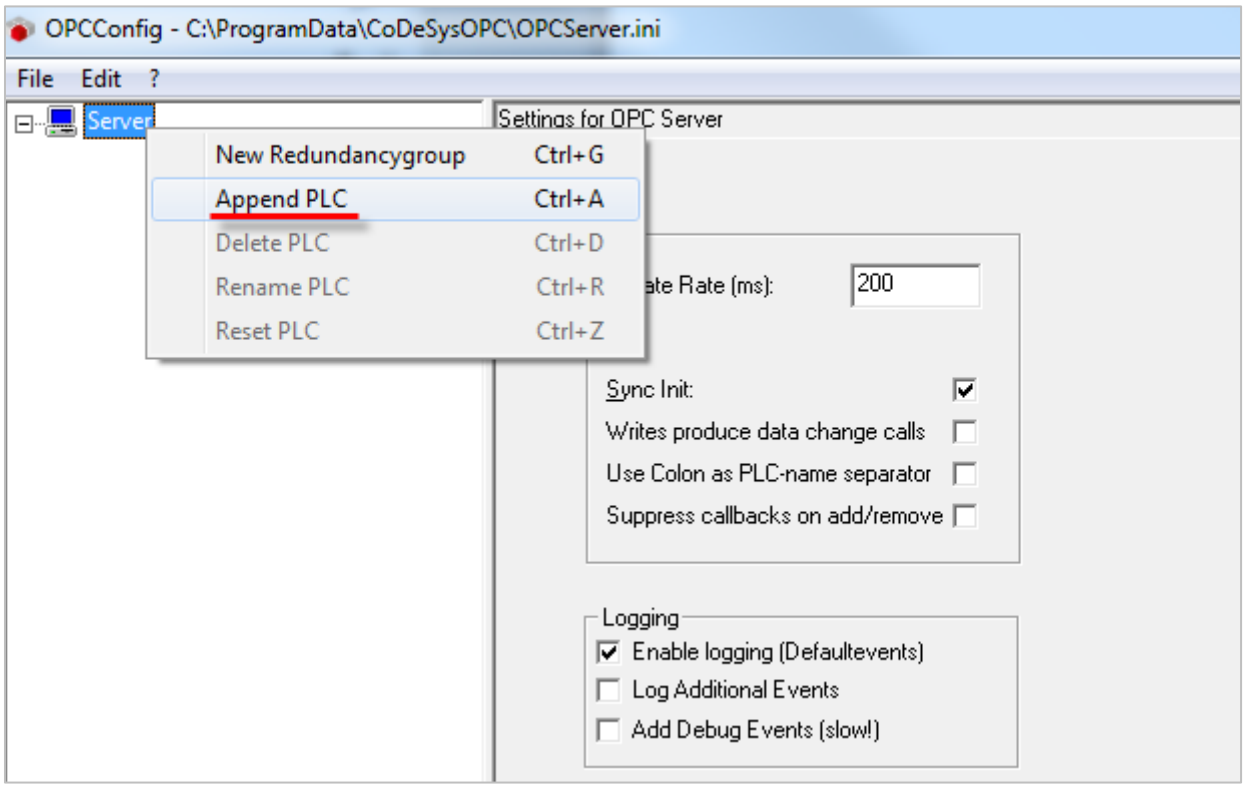


Рисунок 4.2.6 – Добавление контроллера в OPC-сервер

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

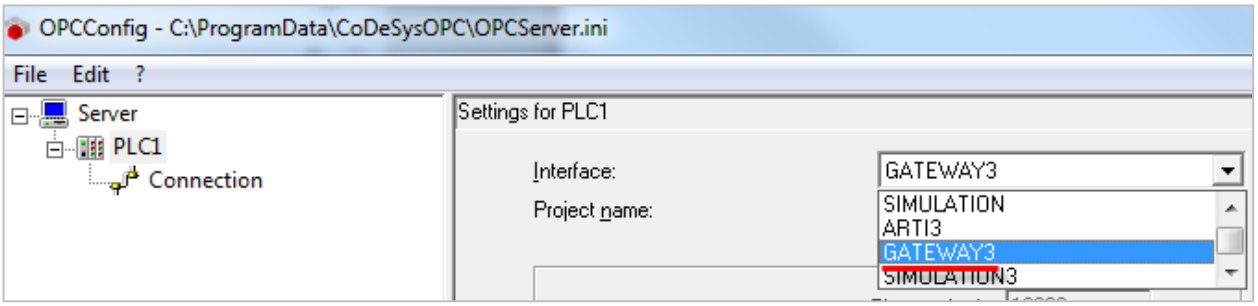


Рисунок 4.2.7 – Выбор интерфейса связи контроллера и OPC-сервера

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

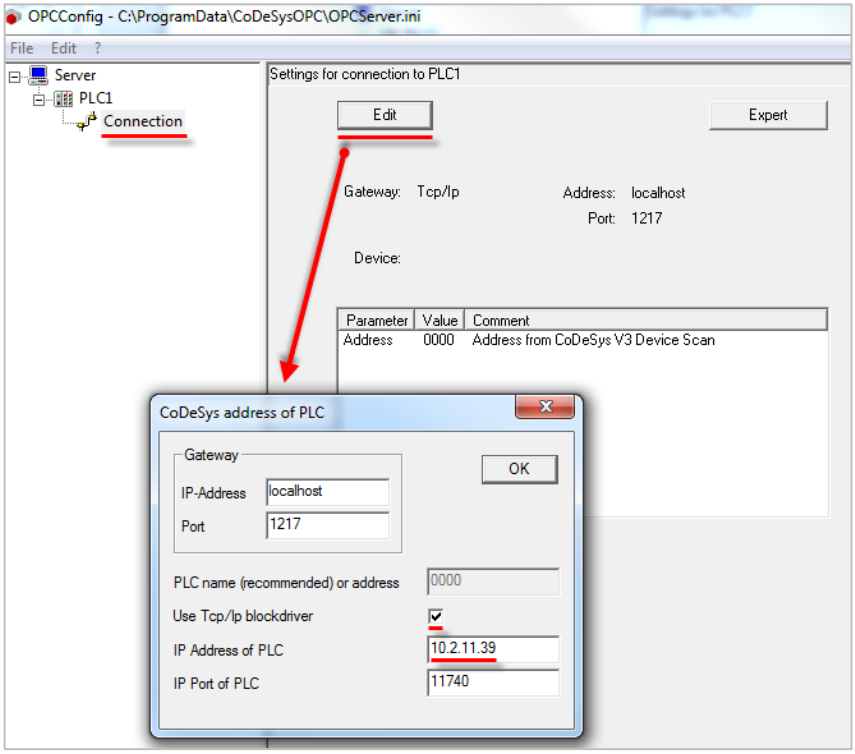


Рисунок 4.2.8 – Указание IP-адреса контроллера

5. Сохранить настройки OPC-сервера:

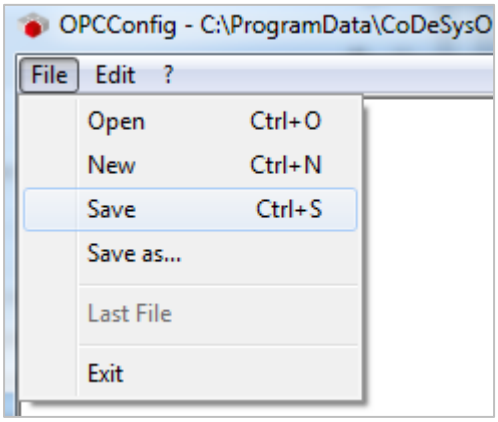


Рисунок 4.2.9 – Сохранение настроек OPC-сервера

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

Затем следует загрузить проект, созданный в п. 4.2.1, в контроллер и убедиться, что контроллер находится в одной локальной сети с OPC-сервером. После загрузки можно переходить к п. 4.6.

4.3 MasterOPC Universal Modbus Server

4.3.1 Настройка контроллера

Для настройки контроллера в режиме **Modbus TCP Slave** следует:

1. Создать новый проект **CODESYS V3.5** (язык программы не имеет значения).
2. Добавить в проект объединение с именем **Real_Word**:

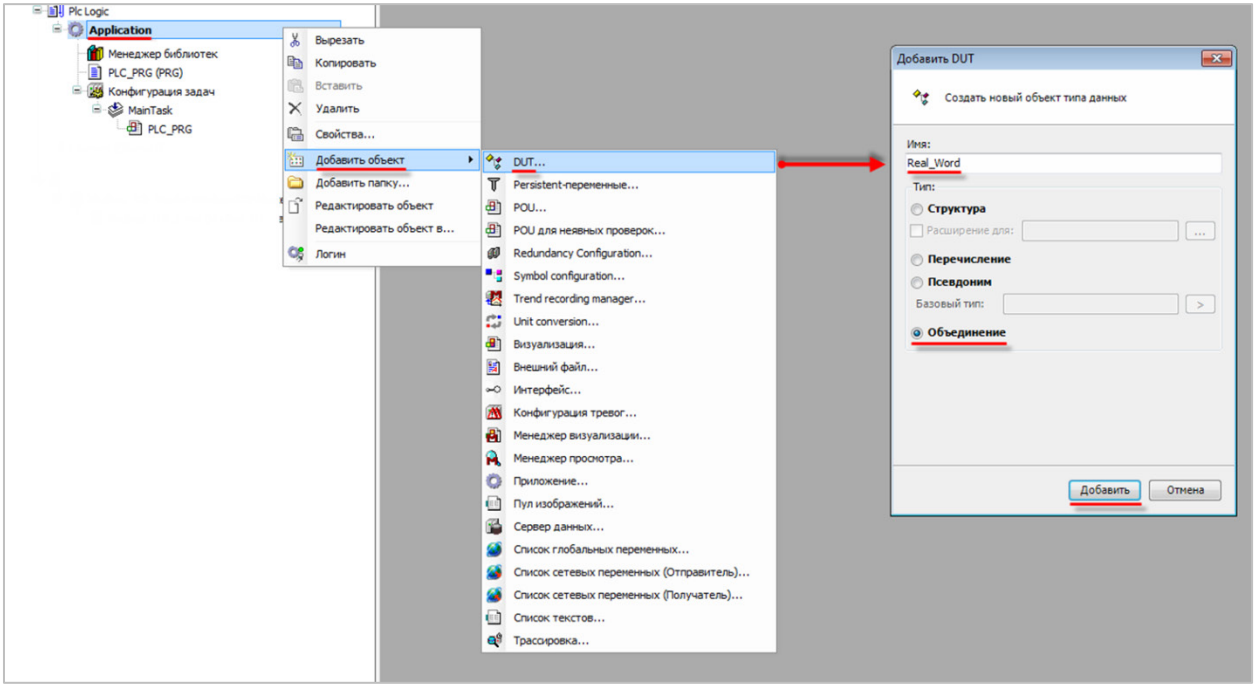


Рисунок 4.3.1 – Добавление в проект объединения

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

Real_Word

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

Рисунок 4.3.2 – Объявление переменных объединения

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

PLC_PRG

```
1   PROGRAM PLC_PRG
2   VAR
3       (* данные, которые считывает OPC-сервер *)
4       xVar_OpcRead:           BOOL;
5       wBitMask_OpcRead:       WORD;
6       wVar_OpcRead:           WORD;
7       _rVar_OpcRead:          Real_Word;
8
9       (* данные, которые записывает OPC-сервер *)
10      xVar_OpcWrite:          BOOL;
11      wBitMask_OpcWrite:      WORD;
12      wVar_OpcWrite:          WORD;
13      _rVar_OpcWrite:         Real_Word;
14  END_VAR
```

Рисунок 4.3.3 – Объявление переменных программы PLC_PRG

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

ПРИМЕЧАНИЕ
Версия компонента не должна превышать версию таргет-файла контроллера.

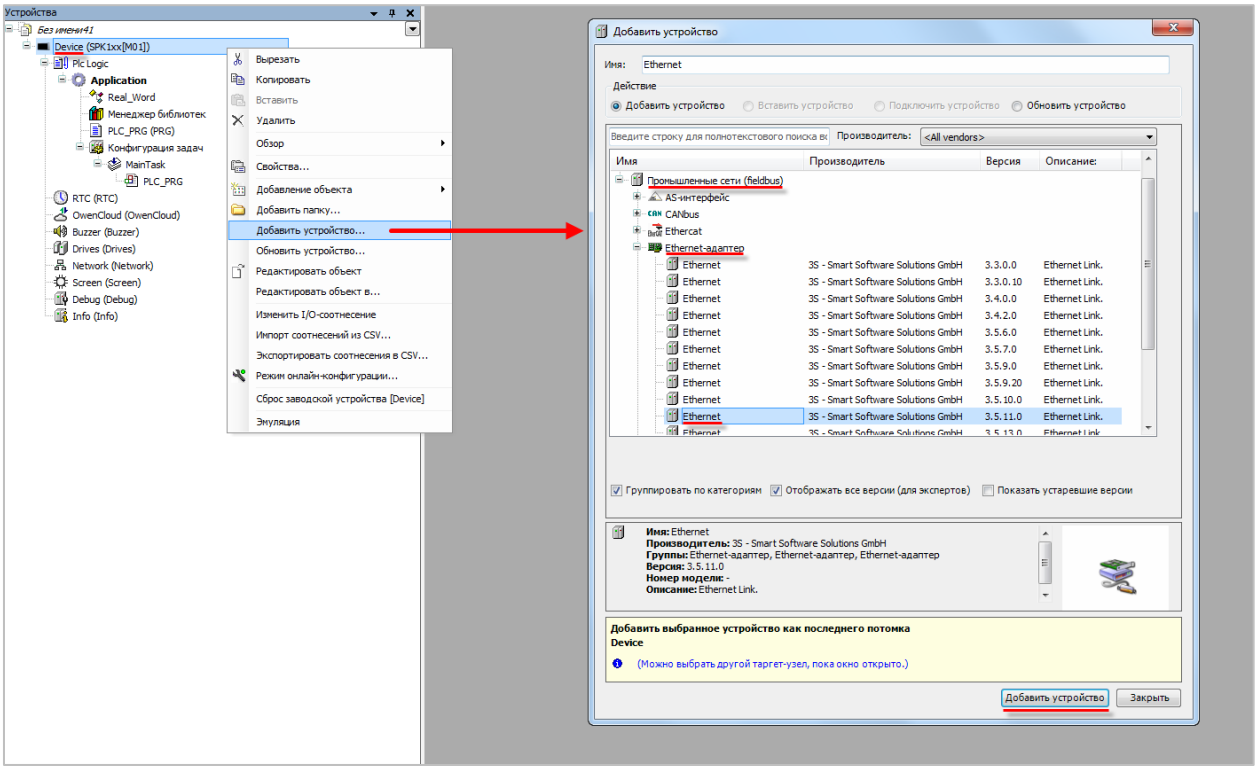


Рисунок 4.3.4 – Добавление компонента Ethernet

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

На вкладке **Конфигурация Ethernet** выбрать адаптер с IP-адресом, совпадающим с IP-адресом контроллера:

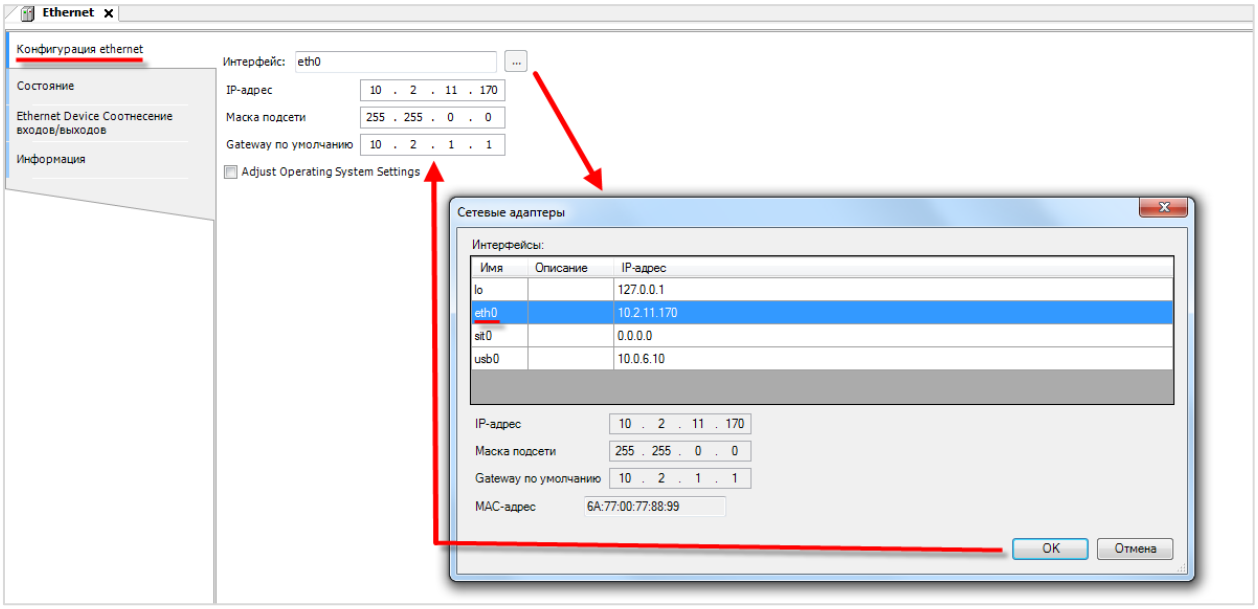


Рисунок 4.3.5 – Настройки компонента Ethernet

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

ПРИМЕЧАНИЕ
Версия компонента не должна превышать версию target-файла контроллера.

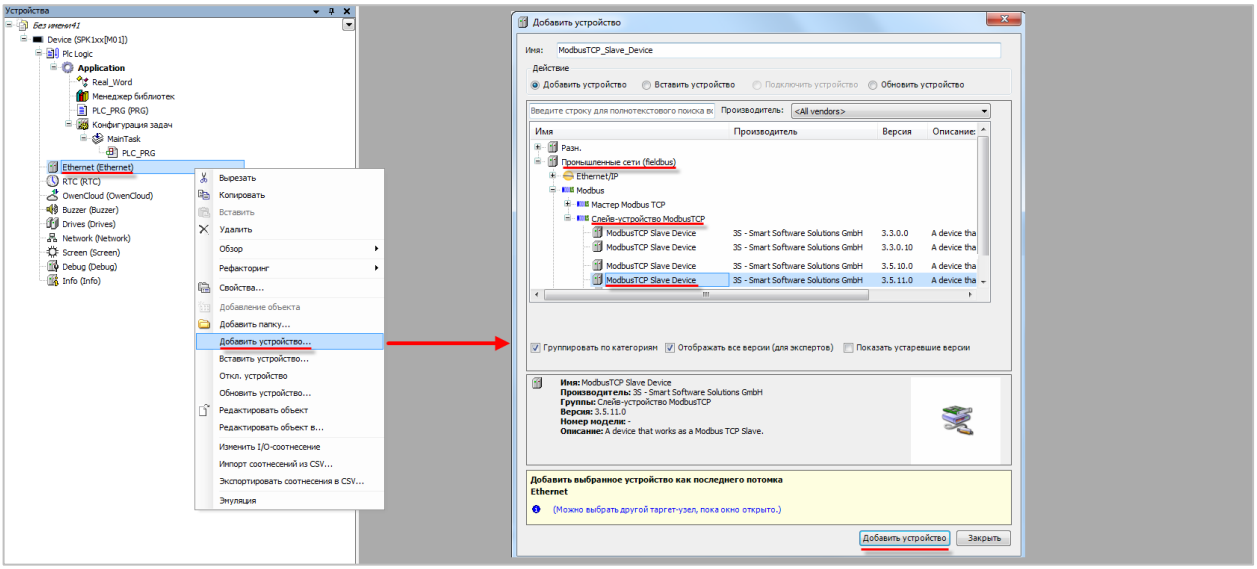


Рисунок 4.3.6 – Добавление компонента Modbus TCP Slave Device

В настройках компонента на вкладке **Страница конфигурации** указать адрес slave-устройства (в примере используется адрес **1**) и порт для Modbus TCP (**502**).

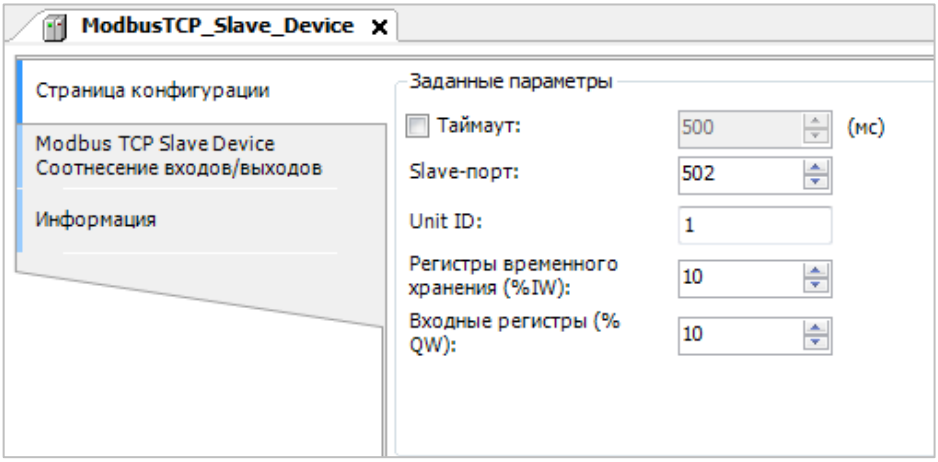


Рисунок 4.3.7 – Настройки компонента Modbus TCP Slave Device

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

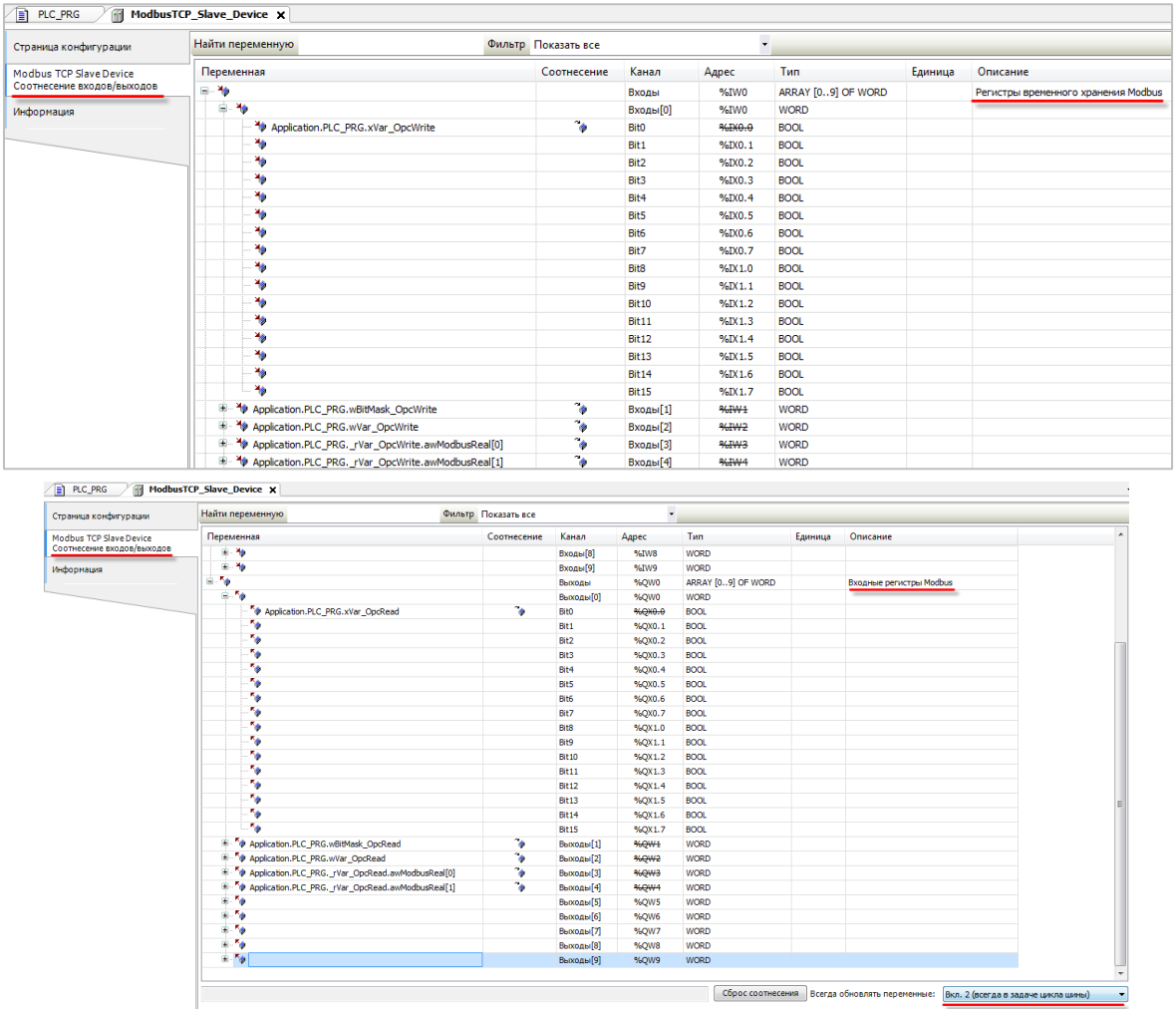


Рисунок 4.3.8 – Привязка переменных к регистрам

ПРИМЕЧАНИЕ

Holding регистры (к которым привязаны **OPC_write** переменные), могут записываться мастер-устройством (в данном случае – OPC-сервером), но не могут быть изменены из программы контроллера. **Input** регистры (к которым привязаны **OPC_read** переменные) могут быть изменены из программы, но не могут быть записаны OPC-сервером.

В результате в контроллере будет сформирована следующая карта регистров:

Таблица 4.2 – Карта регистров контроллера

Область памяти	Адрес регистра	Переменная	Тип переменной
Coils Holding Registers	0x0 (бит 0)	xVar_OpcWrite	BOOL
	0x1	wBitMask_OpcWrite	WORD (битовая маска)
	0x2	wVar_ OpcWrite	WORD
	0x3 – 0x4	_rVar_ OpcWrite…	ARRAY [0..1] OF WORD / REAL
Discrete Inputs Input Registers	0x0 (бит 0)	xVar_OpcRead	BOOL
	0x1	wBitMask_OpcRead	WORD (битовая маска)
	0x2	wVar_ OpcRead	WORD
	0x3 – 0x4	_rVar_ OpcRead…	ARRAY [0..1] OF WORD / REAL

Более подробно вопросы настройки **Modbus Slave** (в том числе **Modbus Serial Slave**) рассмотрены в документе **CODESYS V3.5. Modbus**.

Созданный в данном пункте проект доступен для скачивания: Example_OPC.zip

Пример создан в среде **CODESYS V3.5 SP11 Patch 5** и подразумевает запуск на **СПК1xx [M01]** с таргет-файлом **3.5.11.x.** В случае необходимости запуска проекта на другом устройстве следует изменить таргет-файл в проекте (**ПКМ** на узел **Device** – **Обновить устройство).**

ПРИМЕЧАНИЕ

В рамках примера битовая маска (регистр 1) используется для демонстрации чтения/записи группы переменных типа BOOL с помощью одного регистра. Таким образом можно уменьшить объем передаваемых данных. В примере используется только нулевой бит битовой маски.

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. Modbus**.

4.3.2 Настройка OPC-сервера

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. 4.3.1, пп. 4–5).

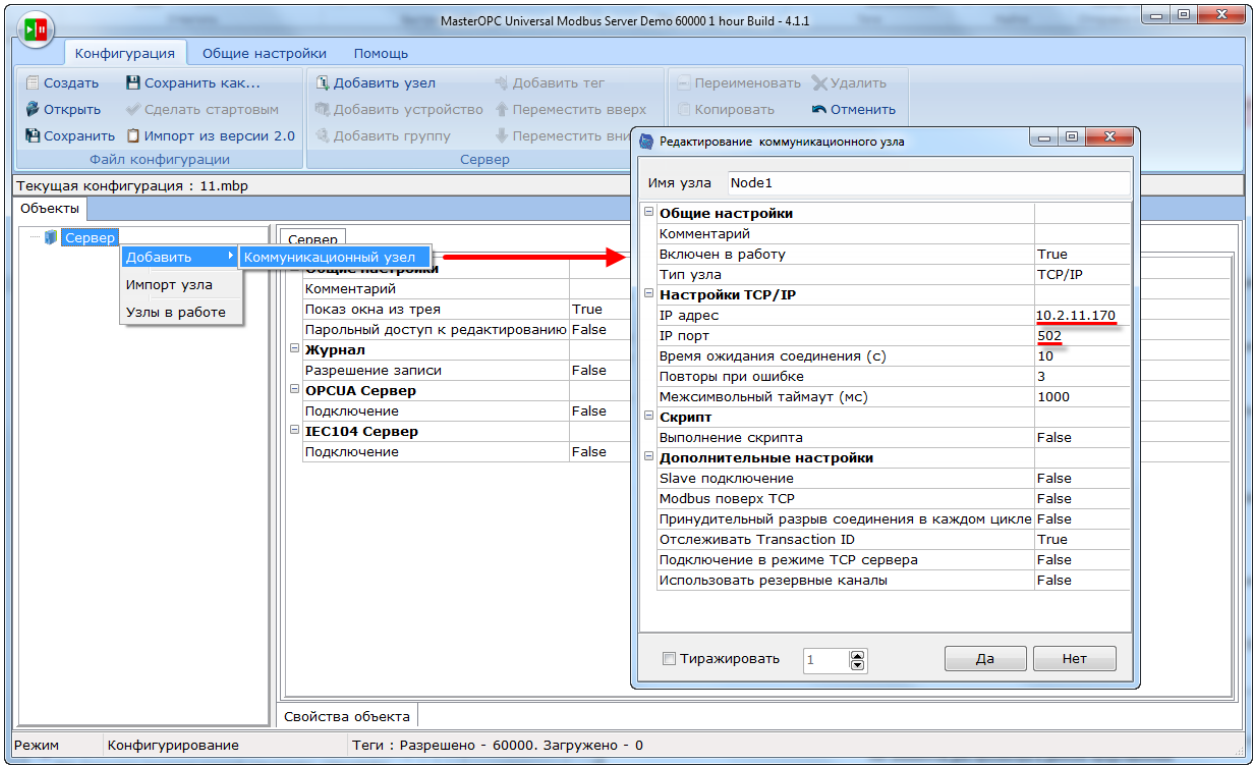


Рисунок 4.3.9 – Добавление коммуникационного узла

3. Нажать **ПКМ** на коммуникационный узел и добавить устройство с названием. В настройках устройства указать адрес **1** (в соответствии с адресом контроллера, см. п. 4.3.1, пп. 5). По умолчанию период опроса составляет 1000 мс – в случае необходимости следует уменьшить это значение.

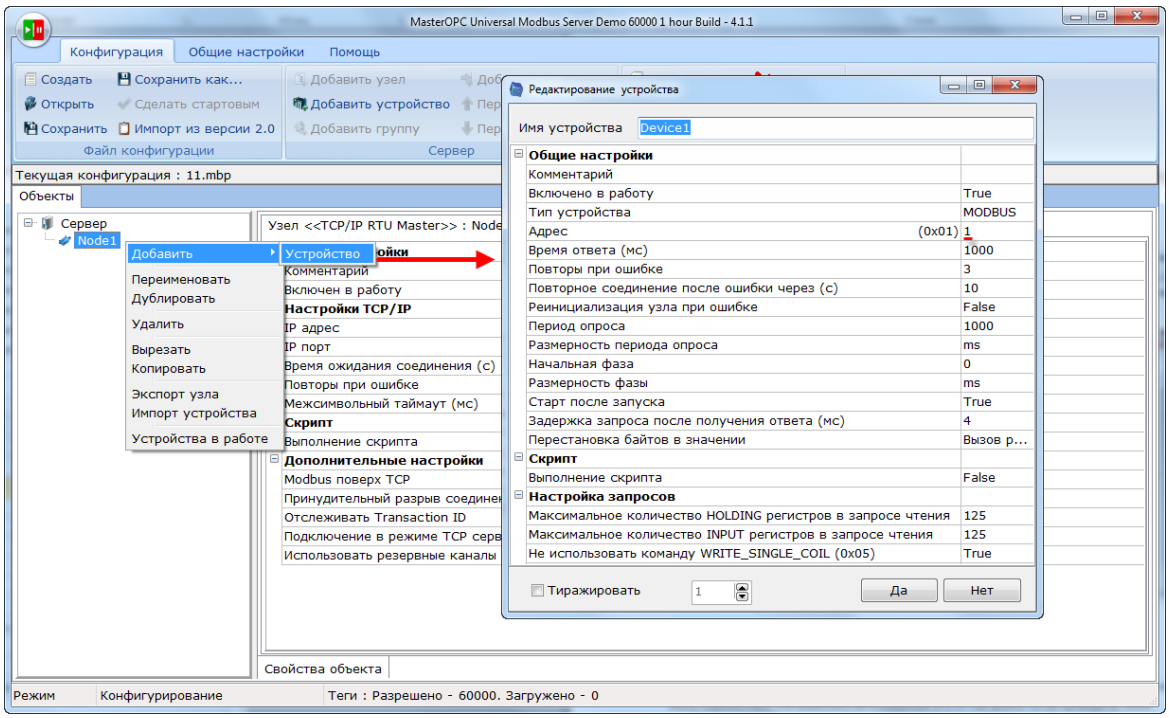


Рисунок 4.3.10 – Добавление устройства

4. Нажать **ПКМ** на устройство и добавить 8 тегов. Число тегов соответствует числу переменных, считываемых/записываемых в контроллере. Настройки тегов приведены ниже.

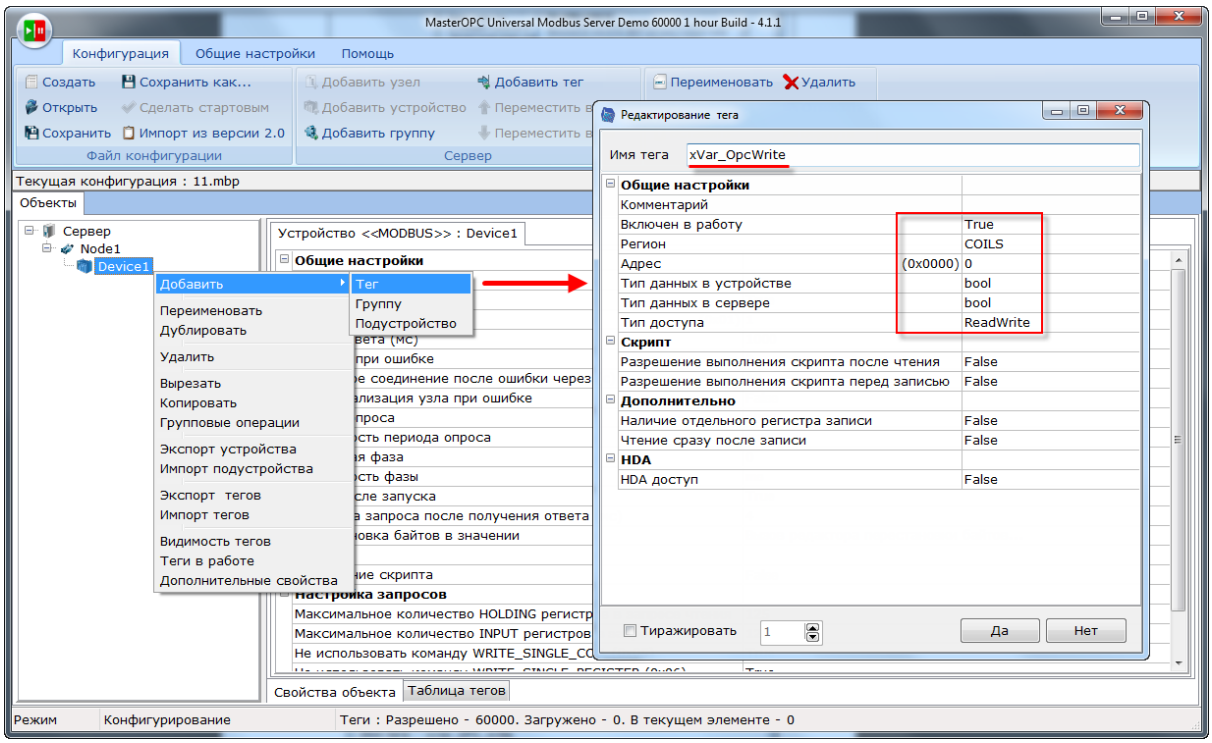


Рисунок 4.3.11 – Добавление и настройка тега xVar_OpcWrite

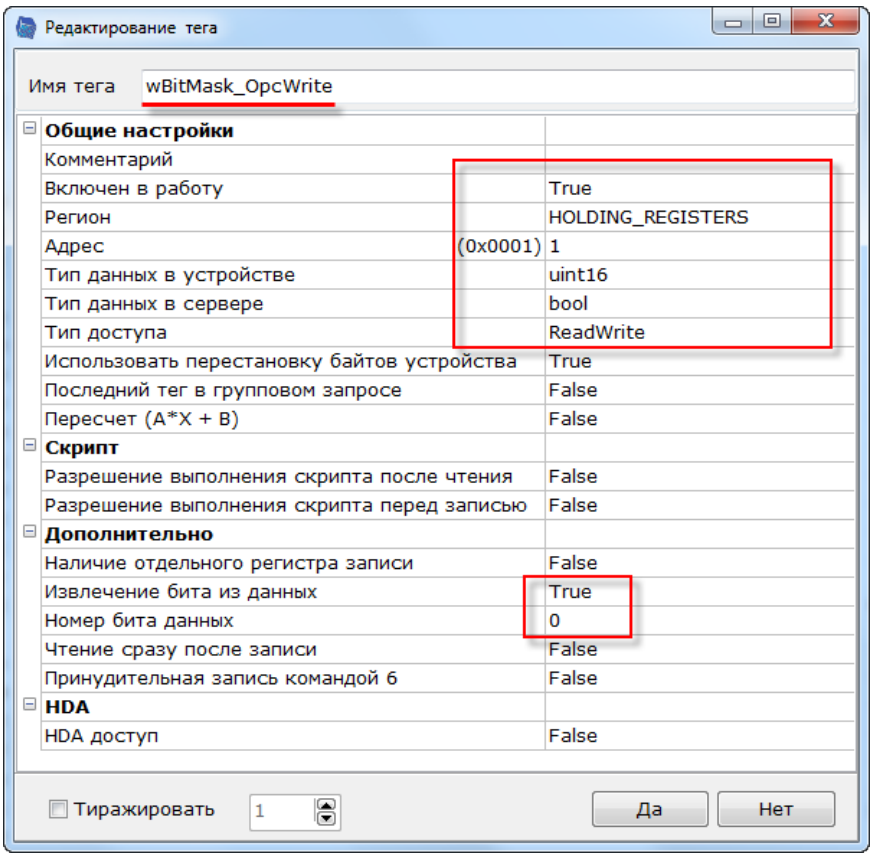


Рисунок 4.3.12 – Настройки тега wBitMask_OpcWrite

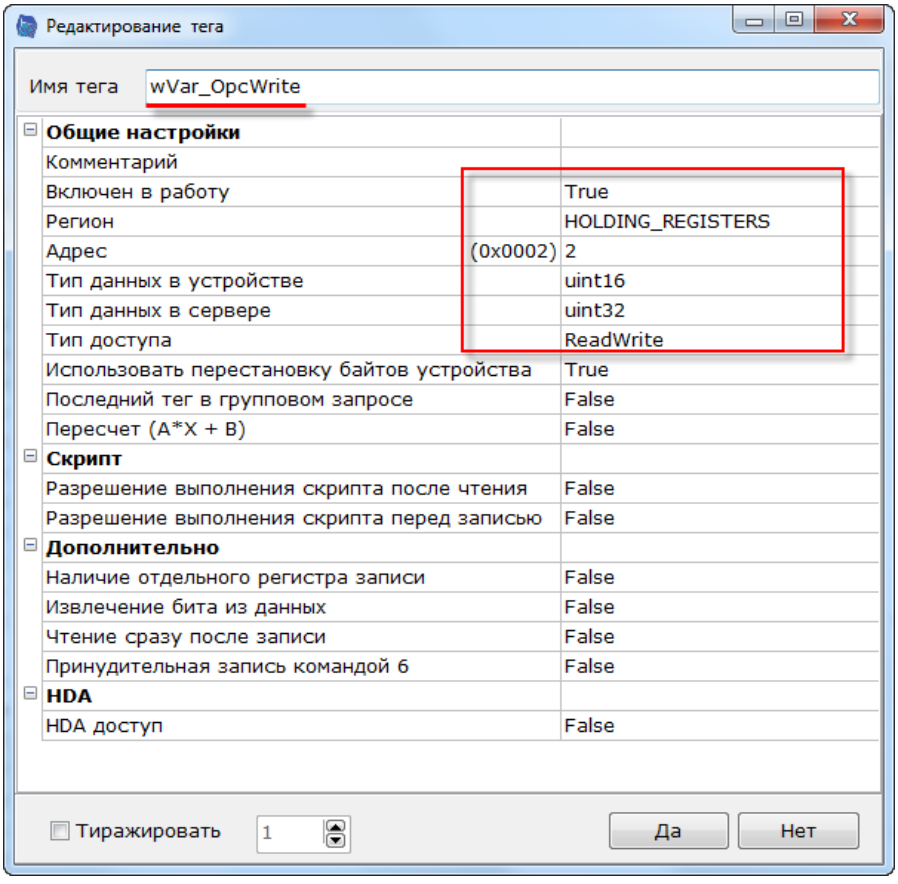


Рисунок 4.3.13 – Настройки тега wVar_OpcWrite

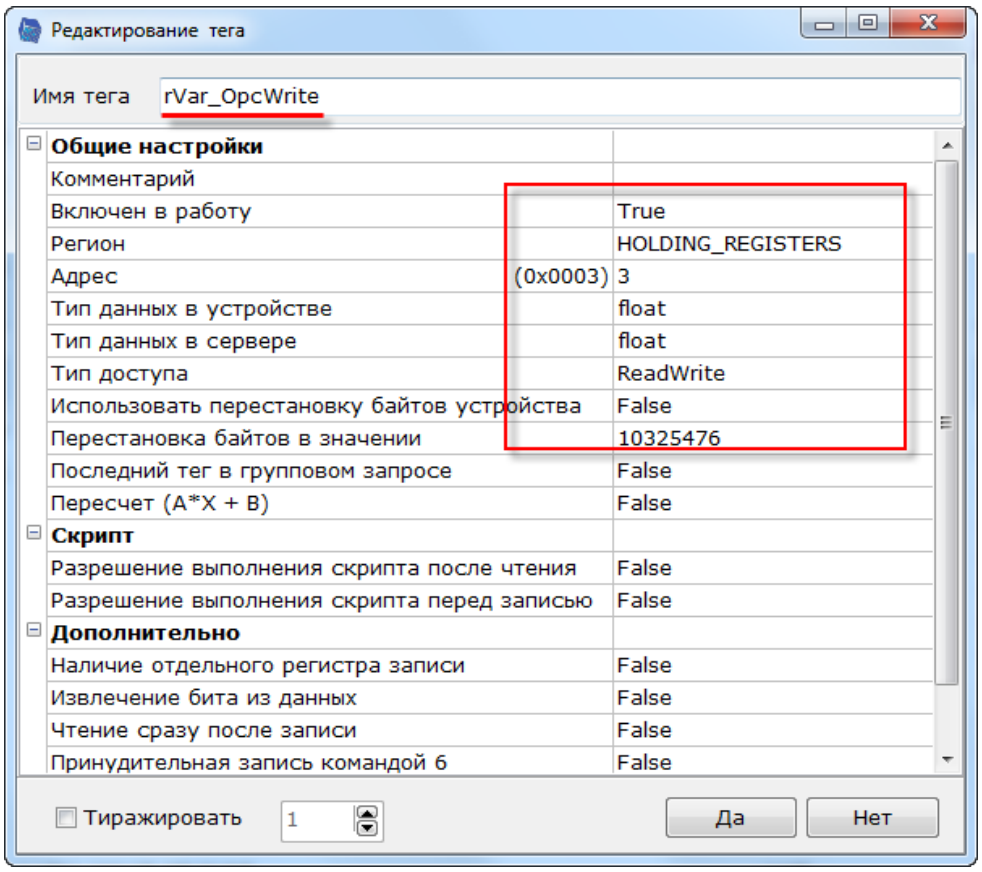


Рисунок 4.3.14 – Настройки тега rVar_OpcWrite

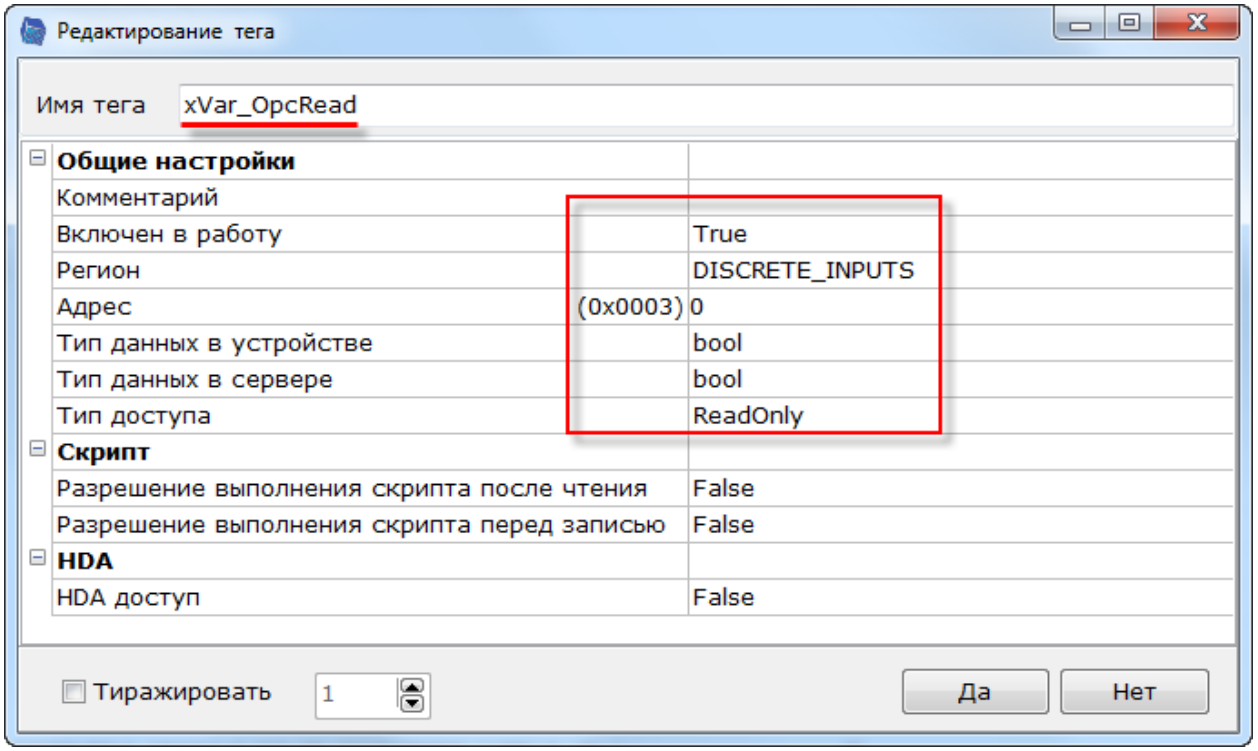


Рисунок 4.3.15 – Настройки тега xVar_OpcRead

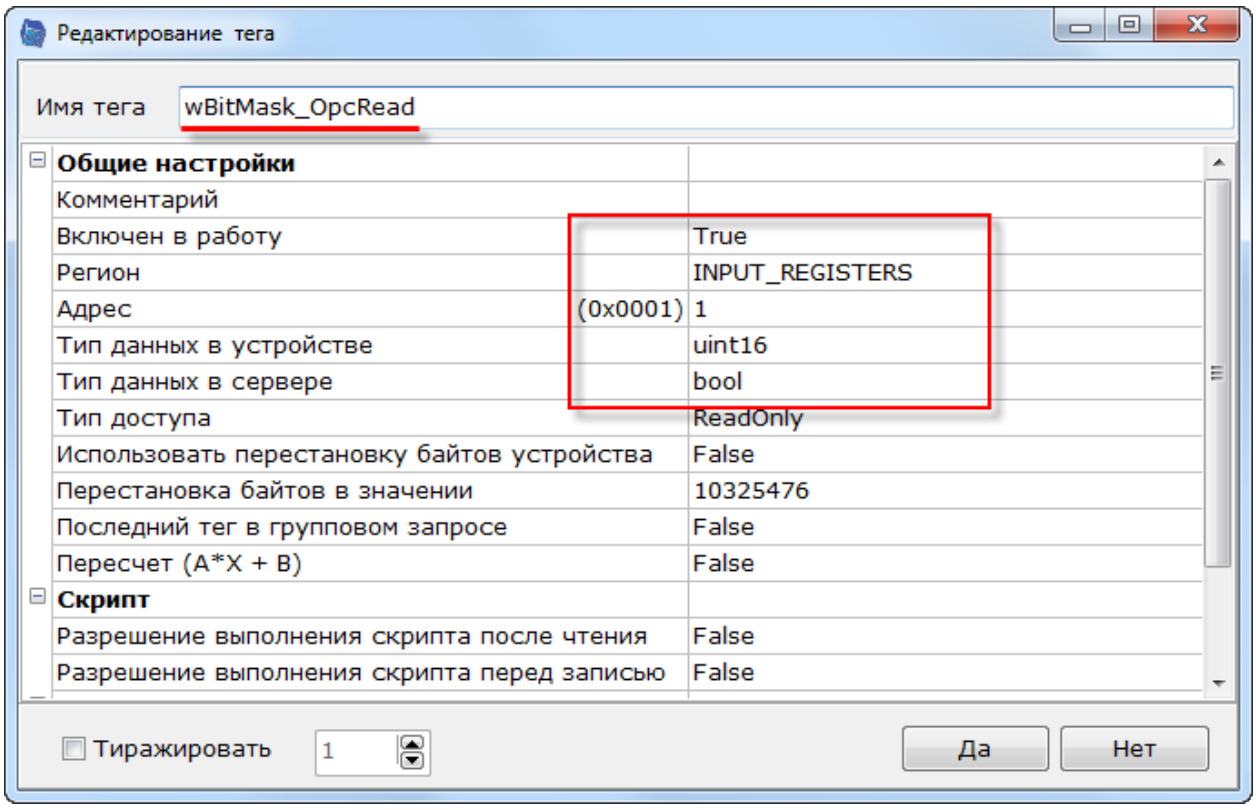


Рисунок 4.3.16 – Настройки тега wBitMask_OpcRead

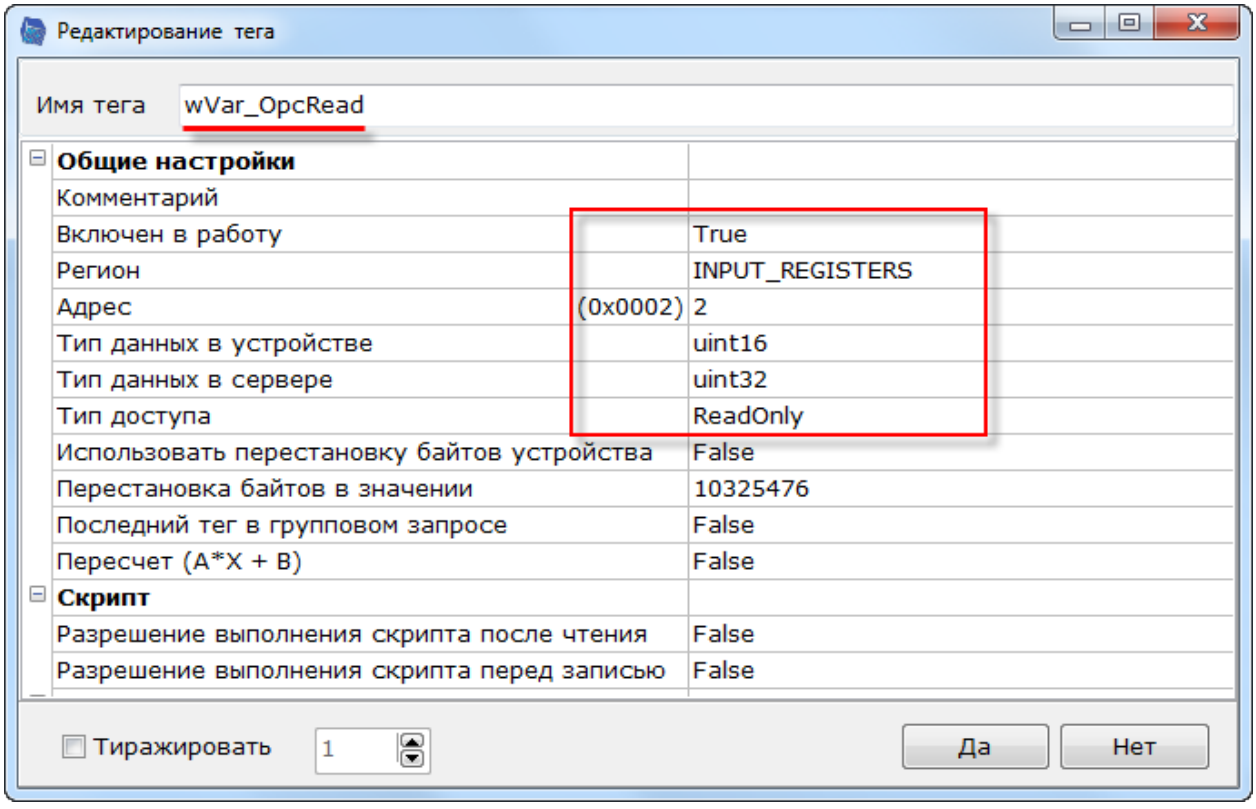


Рисунок 4.3.17 – Настройки тега wVar_OpcRead

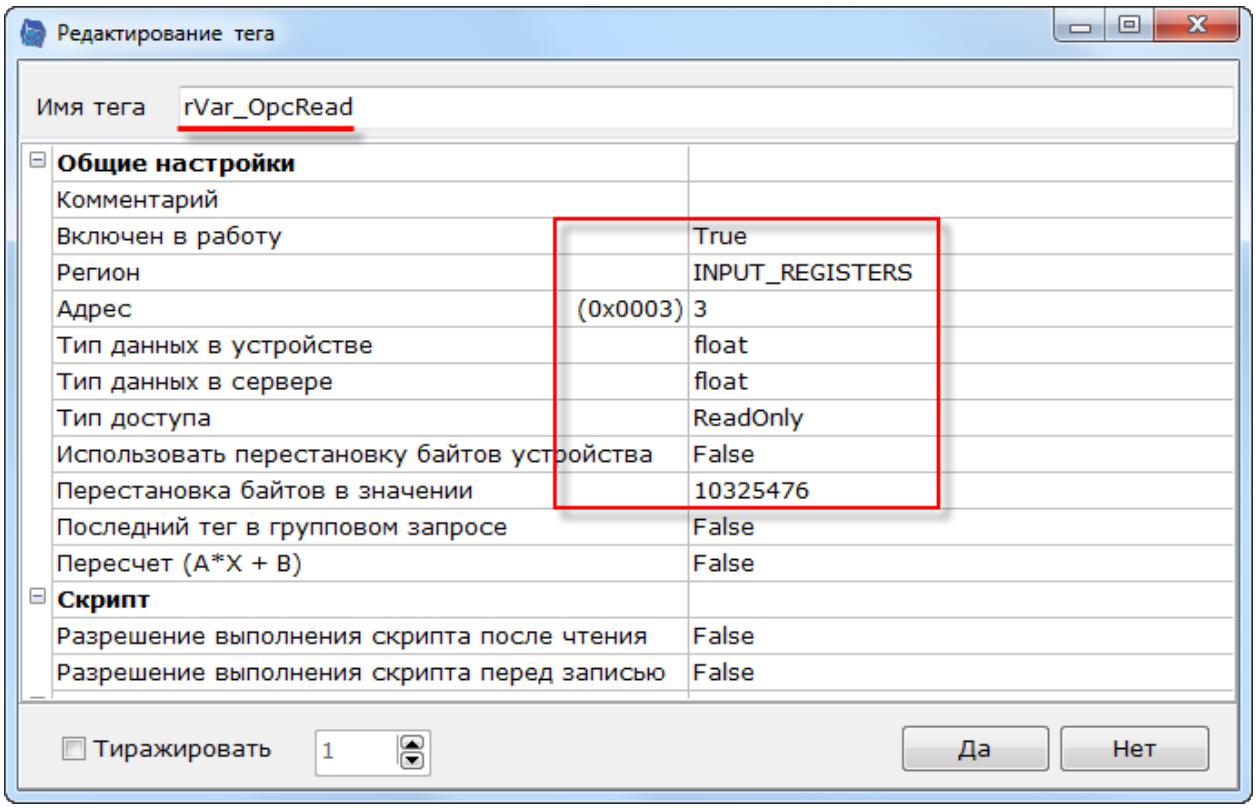


Рисунок 4.3.18 – Настройки тега rVar_OpcRead

5. После добавления и настройки тегов сохранить конфигурацию OPC-сервера:

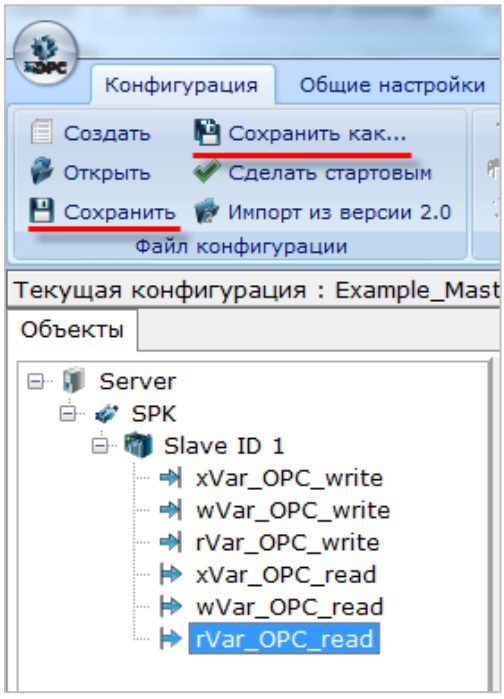


Рисунок 4.3.19 – Сохранение конфигурации OPC-сервера

Затем следует закрыть OPC-сервер. Запускать его не требуется, так как SCADA-система производит этот процесс автоматически.

Затем следует загрузить проект, созданный в п. 4.3.1 в контроллер и убедиться, что контроллер находится в одной локальной сети с OPC-сервером. После загрузки можно переходить к п. 4.6.

Созданная в пункте конфигурация доступна для скачивания: Example_OPC.zip

4.4 Lectus Modbus OPC/DDE Server

4.4.1 Настройка контроллера

Для настройки контроллера следует создать проект согласно п. 4.3.1.

4.4.2 Настройка OPC-сервера

Для настройки OPC-сервера следует:

1. Установить и запустить Lectus Modbus OPC/DDE Server.
2. Нажать **ПКМ** на вкладку **Текущие данные** и добавить узел с названием **OPC_read**. В его настройках указать используемый тип подключения (**TCP клиент**), протокол (**Modbus TCP**), IP-адрес контроллера (в соответствии с п. 4.3.1, пп. 5), порт и адрес устройства (в соответствии с п. 4.3.1, пп. 6), а также функцию **Modbus**, используемую для чтения данных – **04** (**Read Input Registers**).
   Указывать функцию записи не имеет смысла, так как **input** регистры предназначены только для чтения. По умолчанию период опроса составляет 1 секунду – в случае необходимости следует уменьшить это значение.

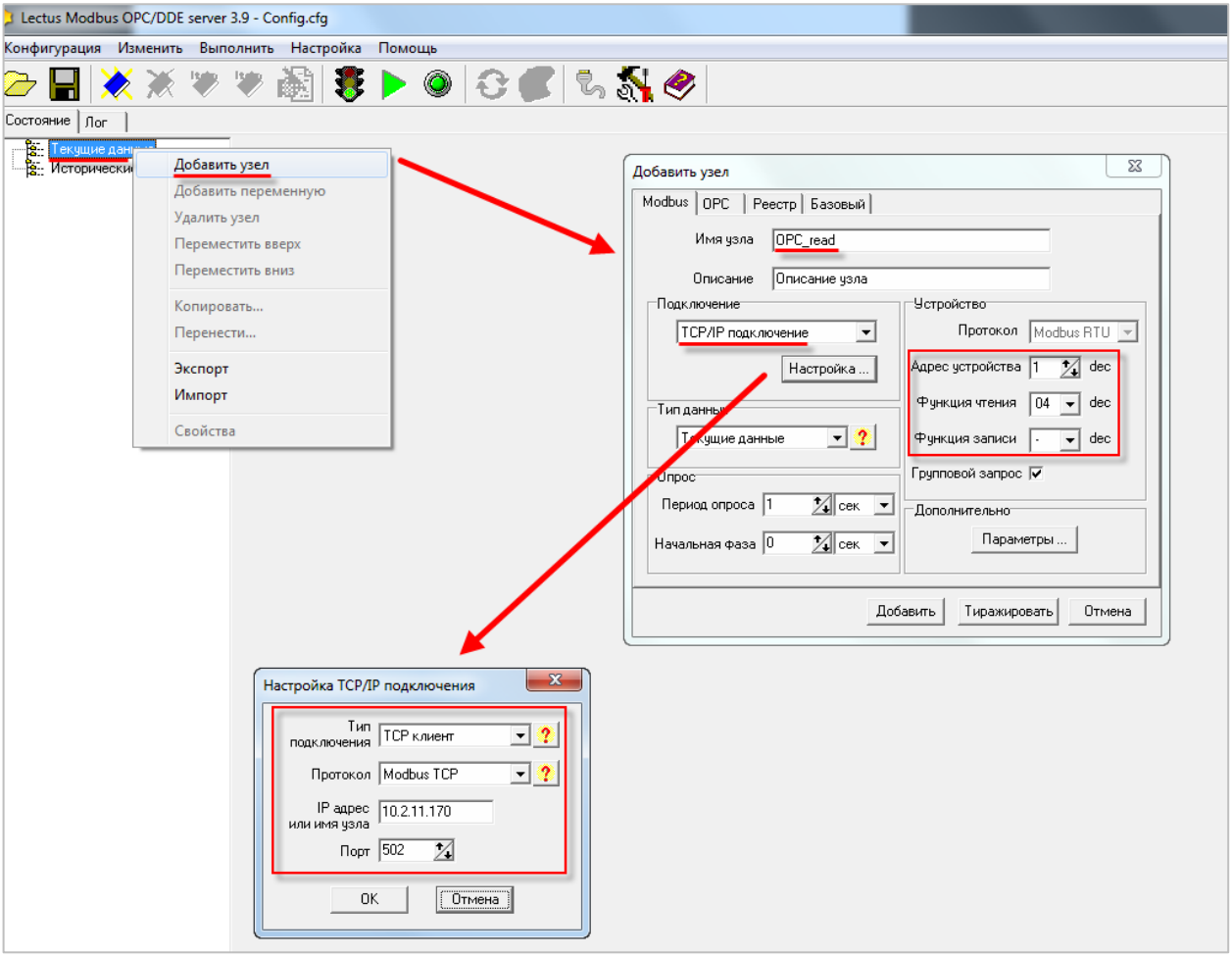


Рисунок 4.4.1 – Добавление узла OPC_read

3. Нажать **ПКМ** на узел **OPC_read** и добавить четыре переменные. Настройки переменных приведены ниже:

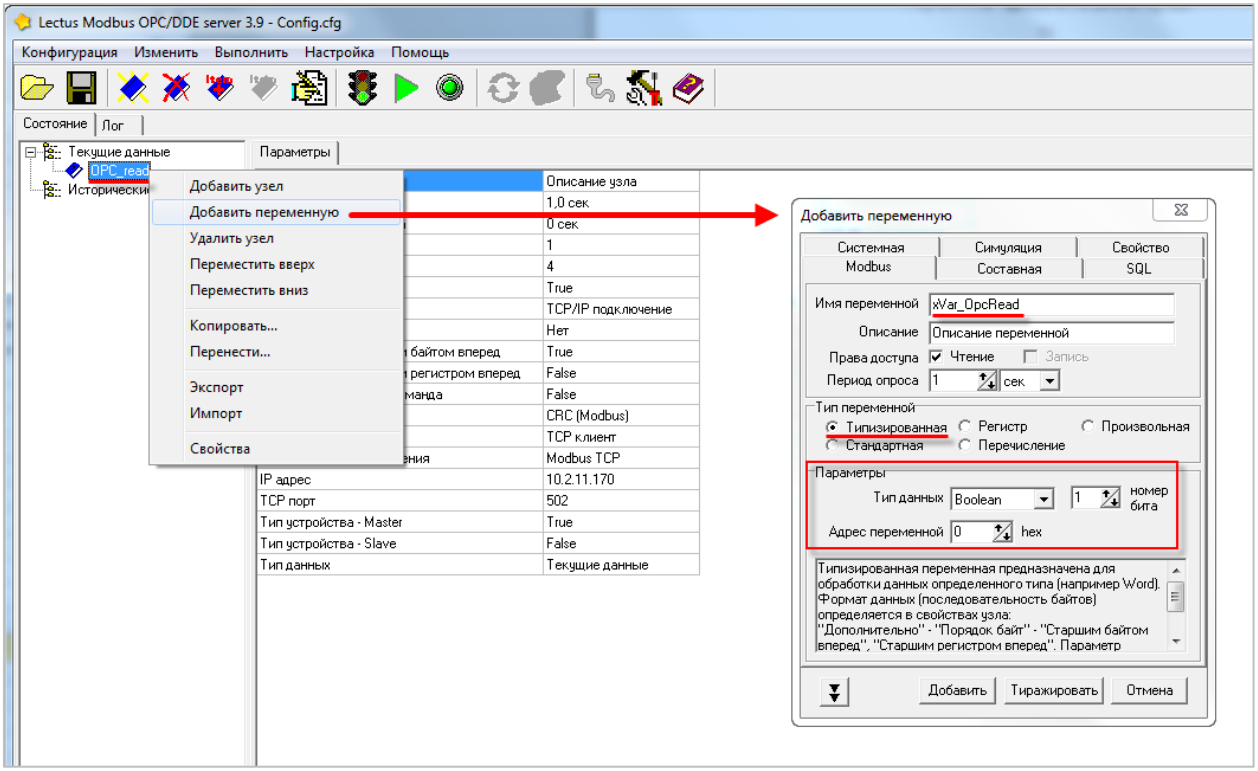


Рисунок 4.4.2 – Добавление и настройка переменной xVar_OpcRead

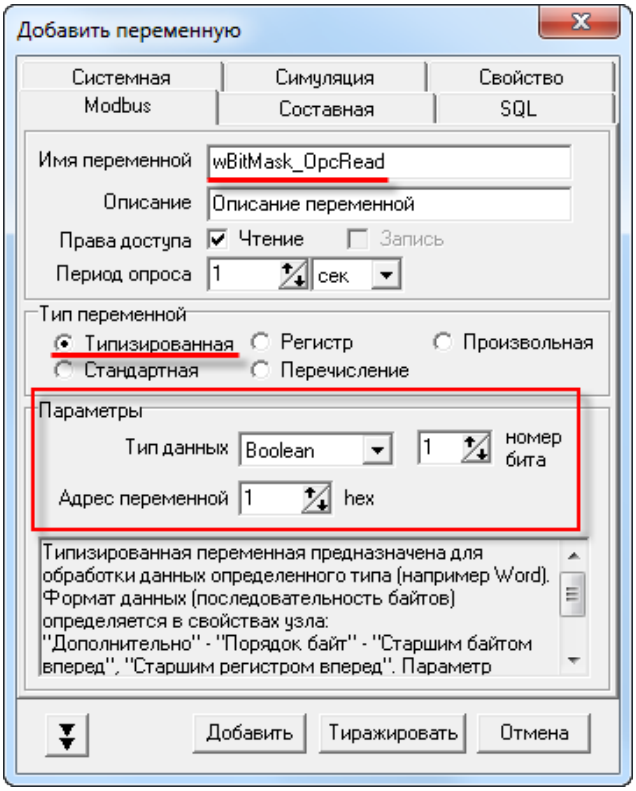


Рисунок 4.4.3 – Добавление и настройка переменной wBitMask_OpcRead

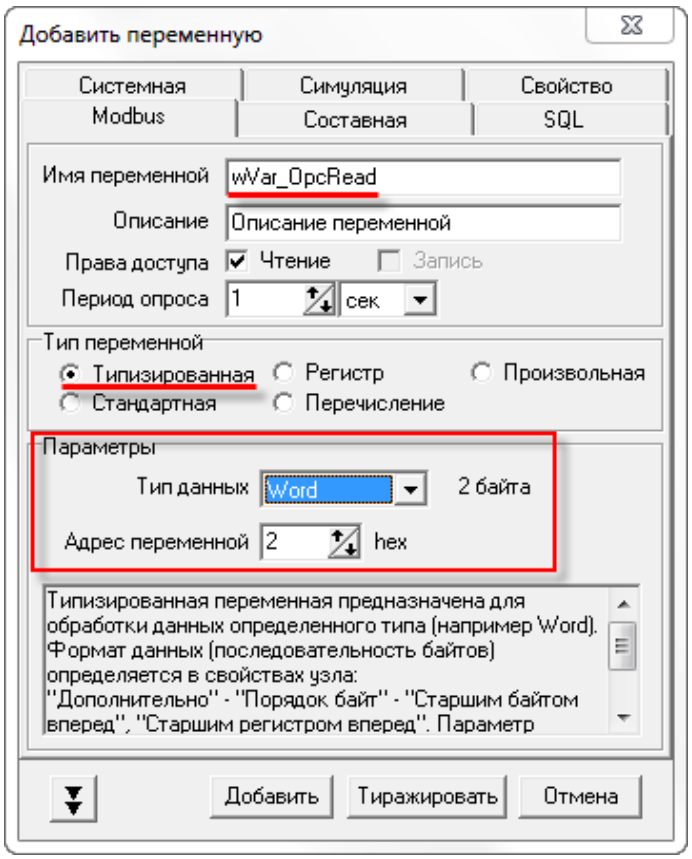


Рисунок 4.4.4 – Настройки переменной wVar_OpcRead

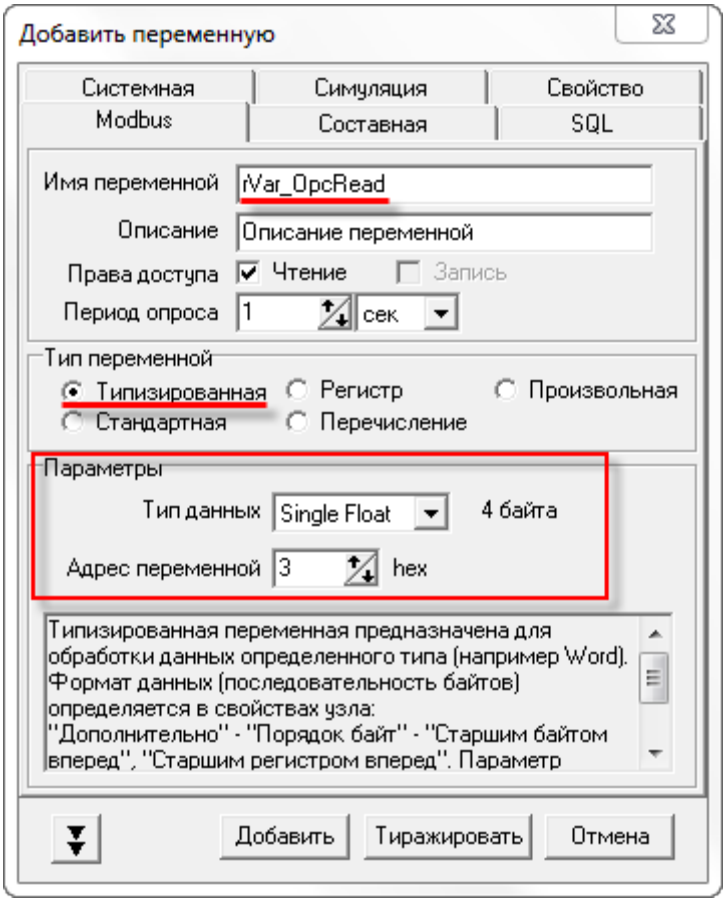


Рисунок 4.4.5 – Настройки переменной rVar_OpcRead

4. Нажать **ПКМ** на вкладку **Текущие данные** и добавить узел с названием **OPC_write**

В его настройках указать:

- используемый тип подключения (**TCP клиент**),
- протокол (**Modbus TCP**),
- IP-адрес контроллера (в соответствии с п. 4.3.1, пп. 5),
- порт и адрес устройства (в соответствии с п. 4.3.1, пп. 6),
- функции **Modbus**, используемые для чтения данных – **03** (**Read Holding Registers**) и записи данных – **16** (**Preset Multiple Registers**).

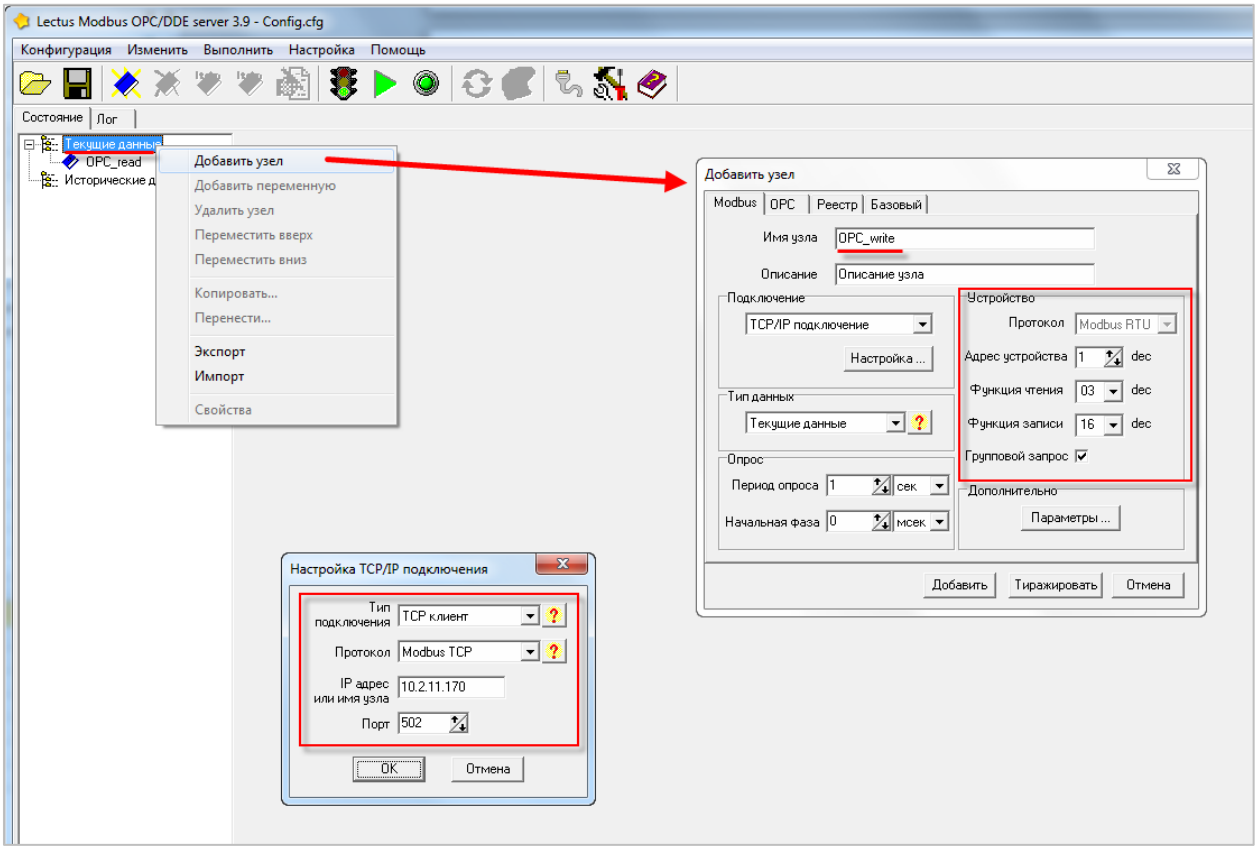


Рисунок 4.4.6 – Добавление узла OPC_write

5. Нажать **ПКМ** на узел **OPC_write** и добавить четыре переменные. Настройки переменных приведены ниже:

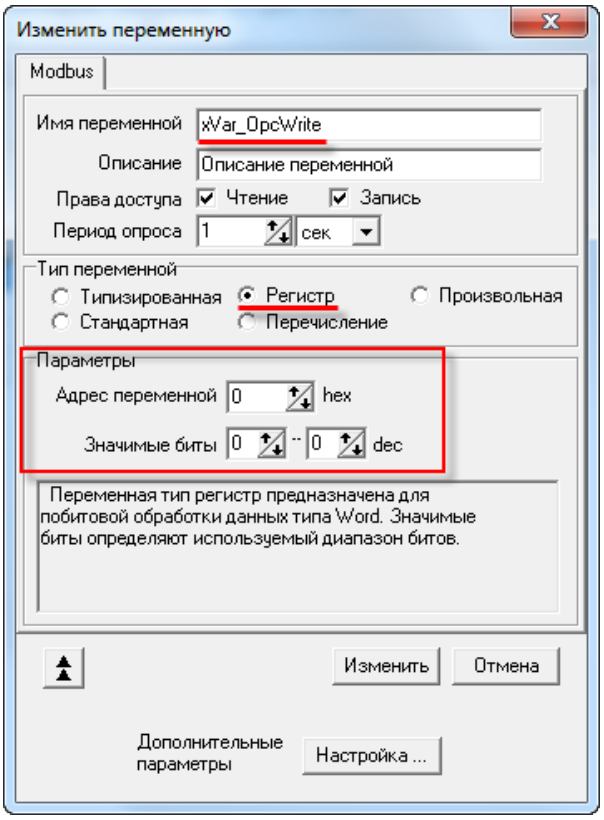


Рисунок 4.4.7 – Добавление и настройки переменной xVar_OpcWrite

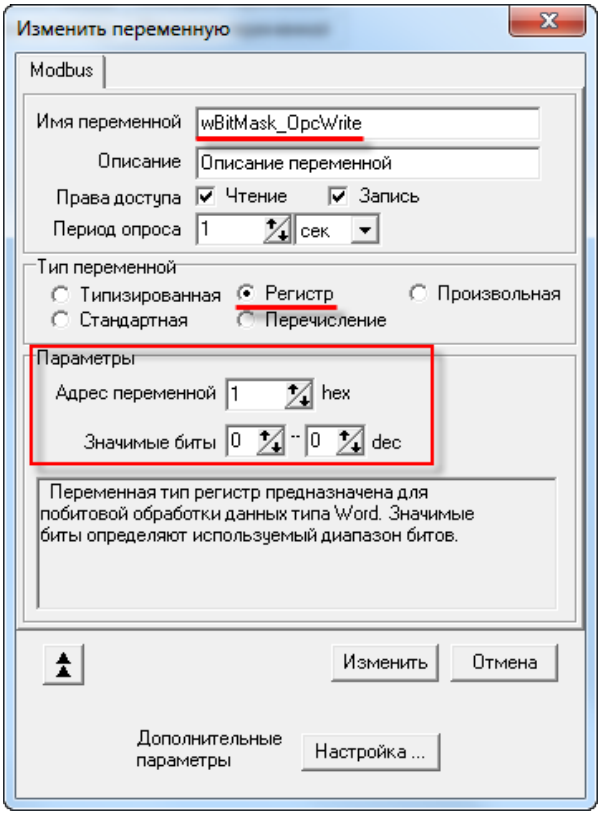


Рисунок 4.4.8 – Настройки переменной wBitMask_OpcWrite

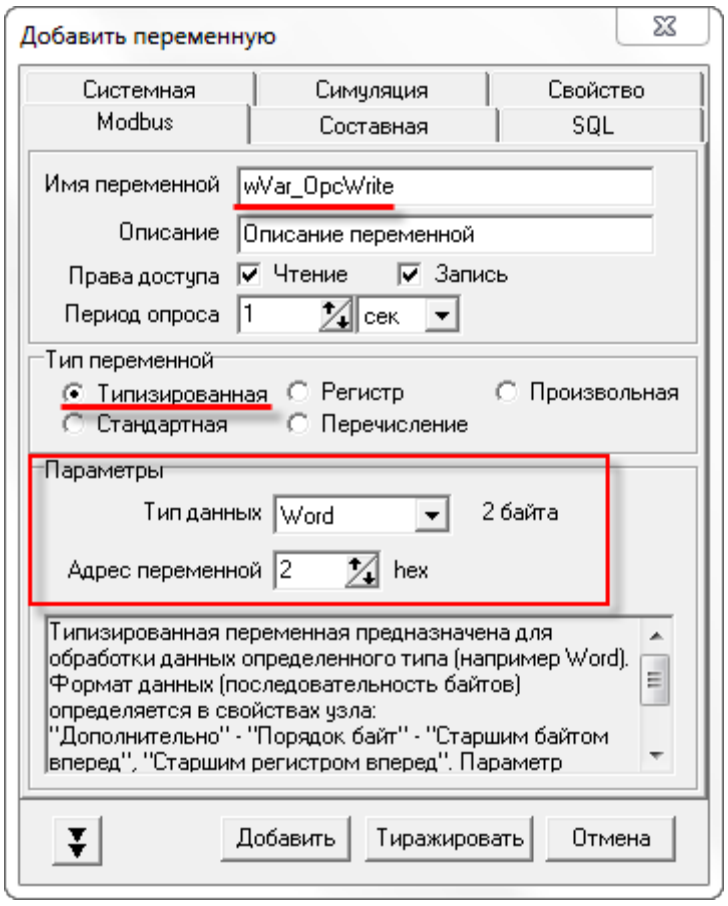


Рисунок 4.4.9 – Настройки переменной wVar_OpcWrite

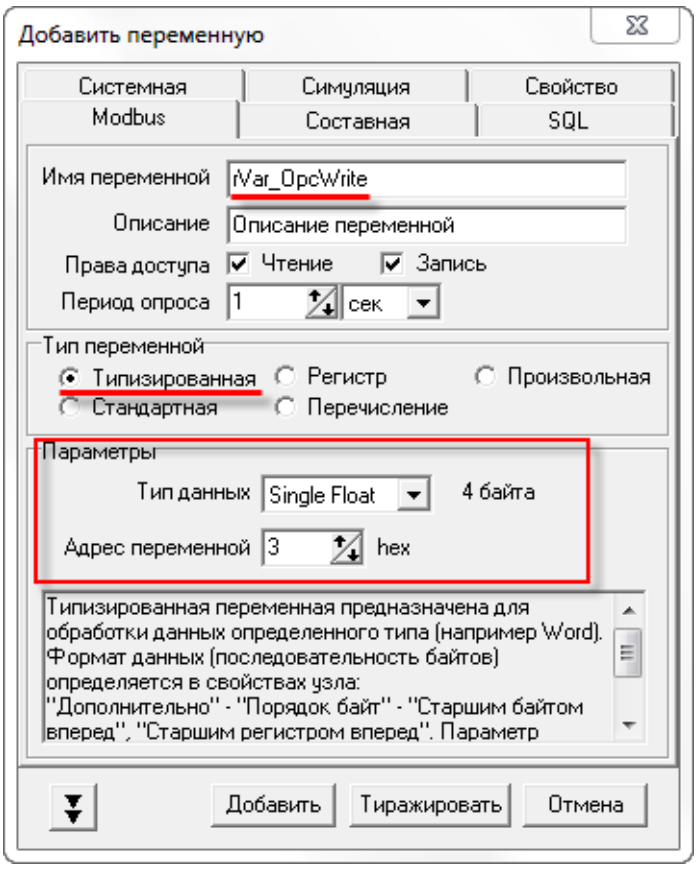


Рисунок 4.4.10 – Настройки переменной rVar_OpcWrite

6. После добавления и настройки переменных сохранить конфигурацию OPC-сервера:

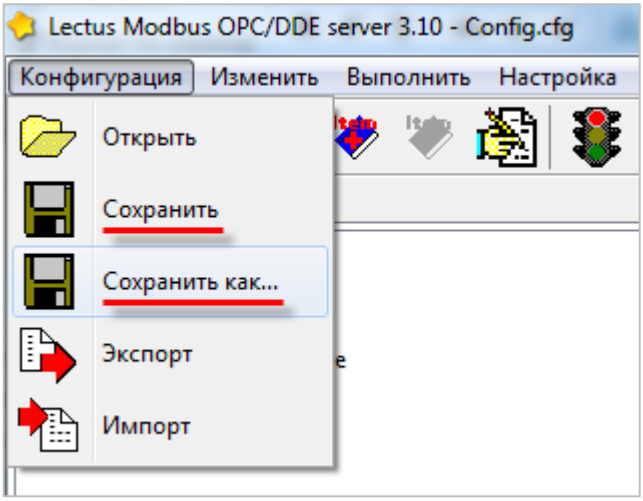


Рисунок 4.4.11 – Сохранение конфигурации OPC-сервера

Затем следует закрыть OPC-сервер. Запускать его не требуется, поскольку SCADA-система делает это автоматически.

Затем следует загрузить проект, созданный в п. 4.3.1 в контроллер и убедиться, что контроллер находится в одной локальной сети с OPC-сервером. После загрузки можно переходить к п. 4.6.

Созданная в пункте конфигурация доступна для скачивания: Example_OPC.zip

4.5 ОВЕН OPC

4.5.1 Настройка контроллера

Для настройки контроллера следует создать проект согласно п. 4.3.1.

4.5.2 Настройка OPC-сервера

Для настройки OPC-сервера следует:

1. Установить и запустить ОВЕН OPC.
2. Нажать **ПКМ** на узел **Сервер** и добавить узел с названием по умолчанию (**Узел1**). В его настройках указать используемый протокол (**Modbus TCP**).

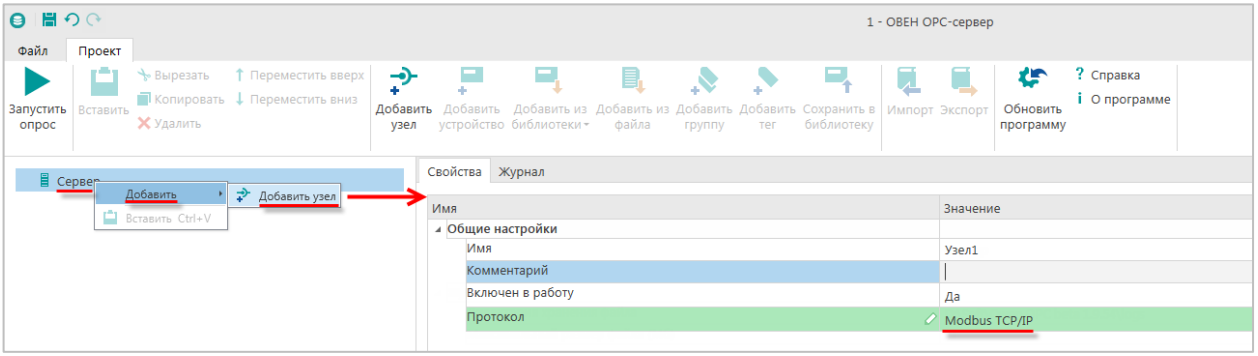


Рисунок 4.5.1 – Добавление и настройка узла

3. Нажать **ПКМ** на узел **Узел1** и добавить устройство. В настройках указать сетевые параметры (**IP-адрес** и **порт**). Сетевые настройки должны соответствовать настройкам контроллера (см. п. 4.3.1, пп. 4–5).

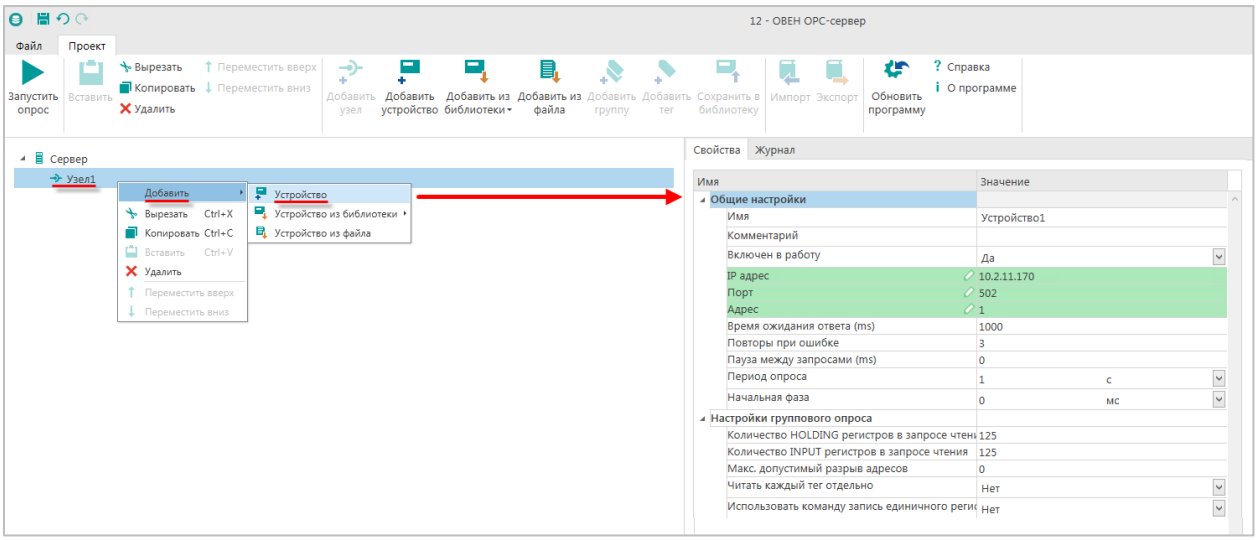


Рисунок 4.5.2 – Добавление коммуникационного узла

4. Нажать **ПКМ** на устройство и добавить 8 тегов. Число тегов соответствует числу переменных, считываемых/записываемых в контроллере. Настройки тегов приведены ниже.

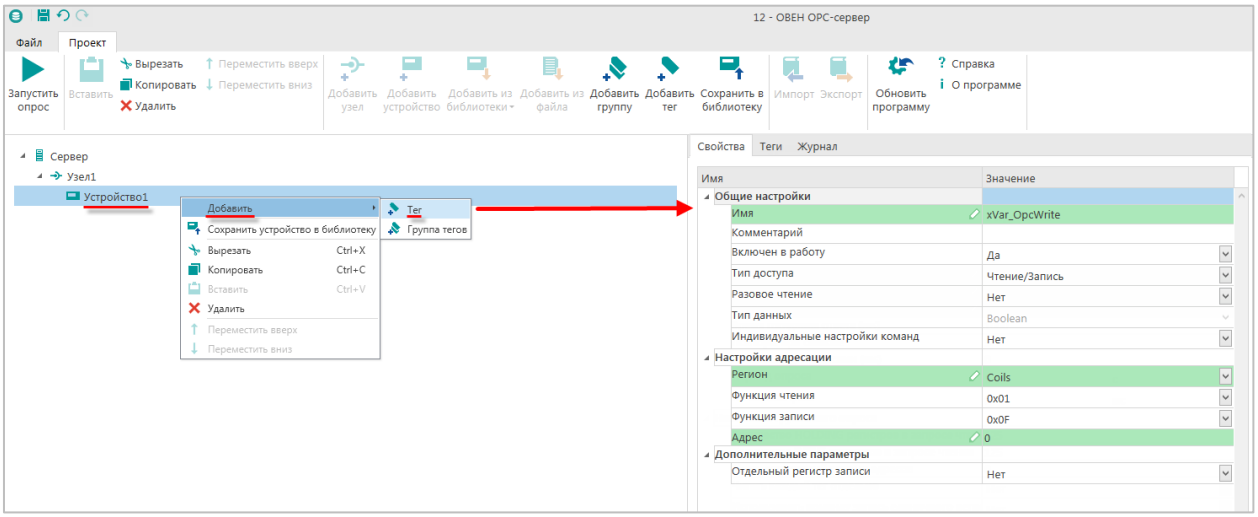


Рисунок 4.5.3 – Добавление и настройка тега xVar_OpcWrite

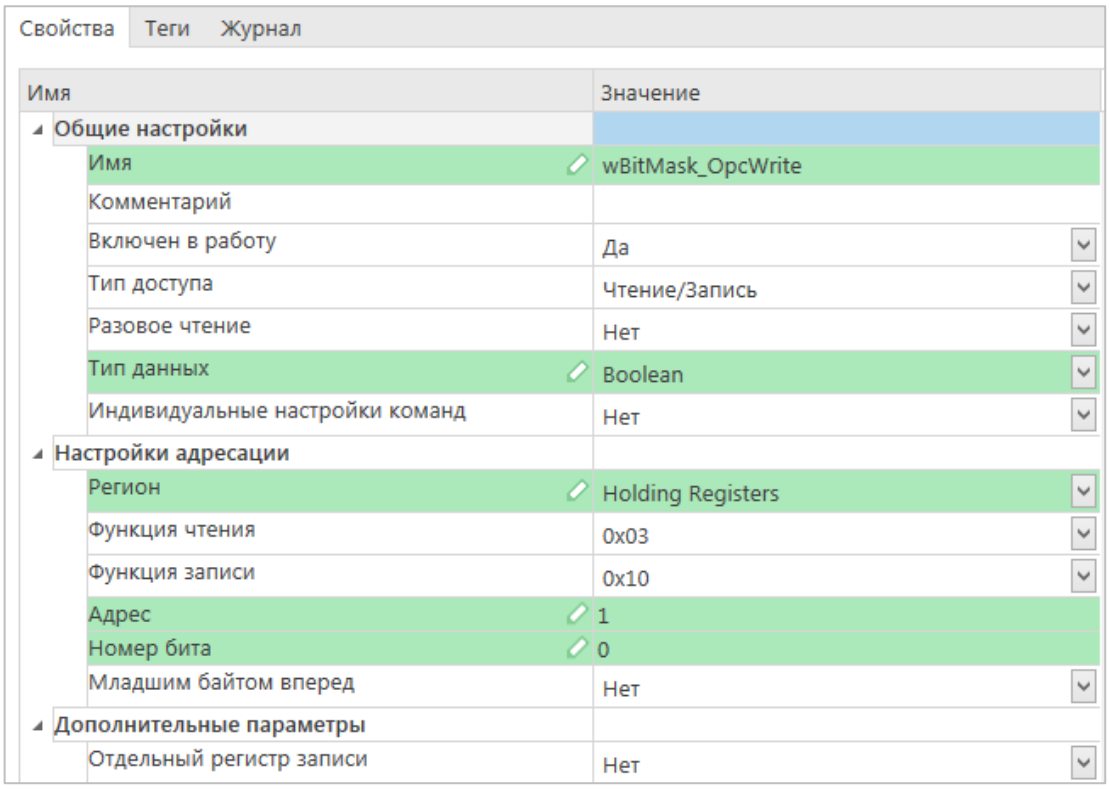


Рисунок 4.5.4 – Настройки тега wBitMask_OpcWrite

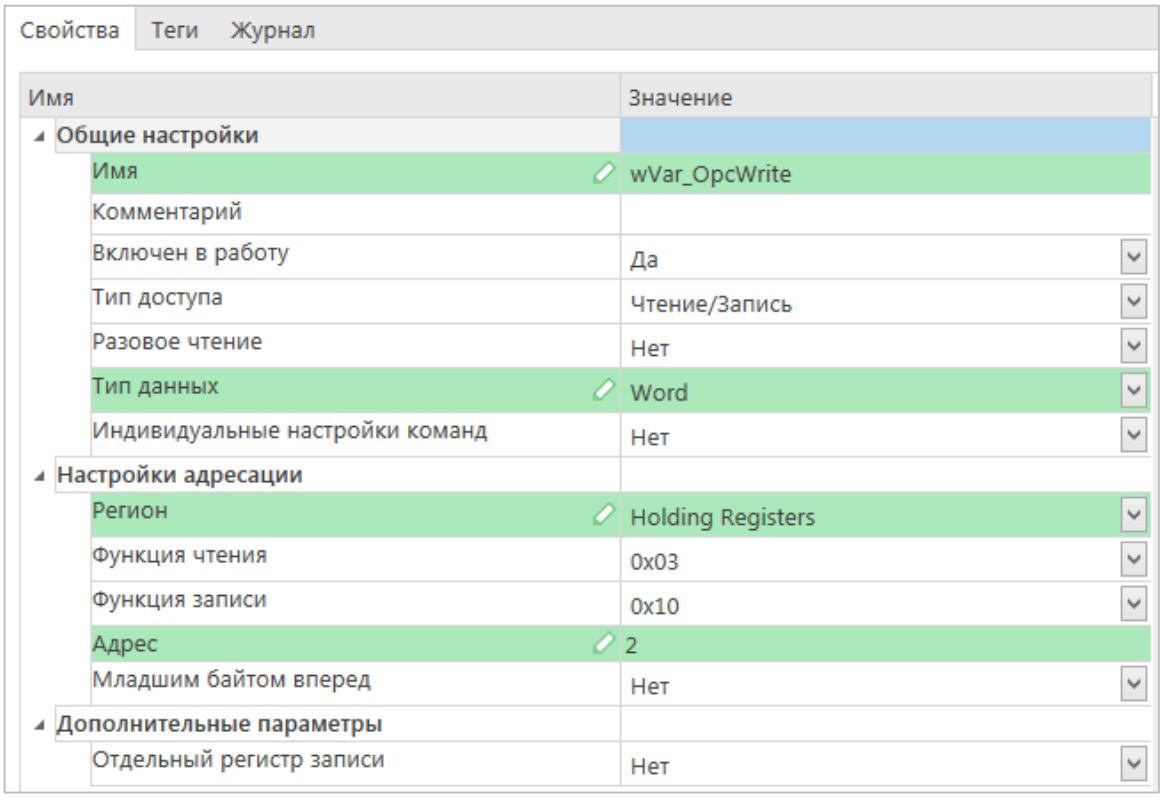


Рисунок 4.5.5 – Настройки тега wVar_OpcWrite

Свойства | Теги | Журнал

Имя	Значение
Общие настройки	
Имя	rVar_OpcWrite
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	3
Младшим байтом вперед	Нет
Младшим регистром вперед	Да
Дополнительные параметры	
Отдельный регистр записи	Нет

Рисунок 4.5.6 – Настройки тега rVar_OpcWrite

Свойства | Теги | Журнал

Имя	Значение
Общие настройки	
Имя	xVar_OpcRead
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 4.5.7 – Настройки тега xVar_OpcRead

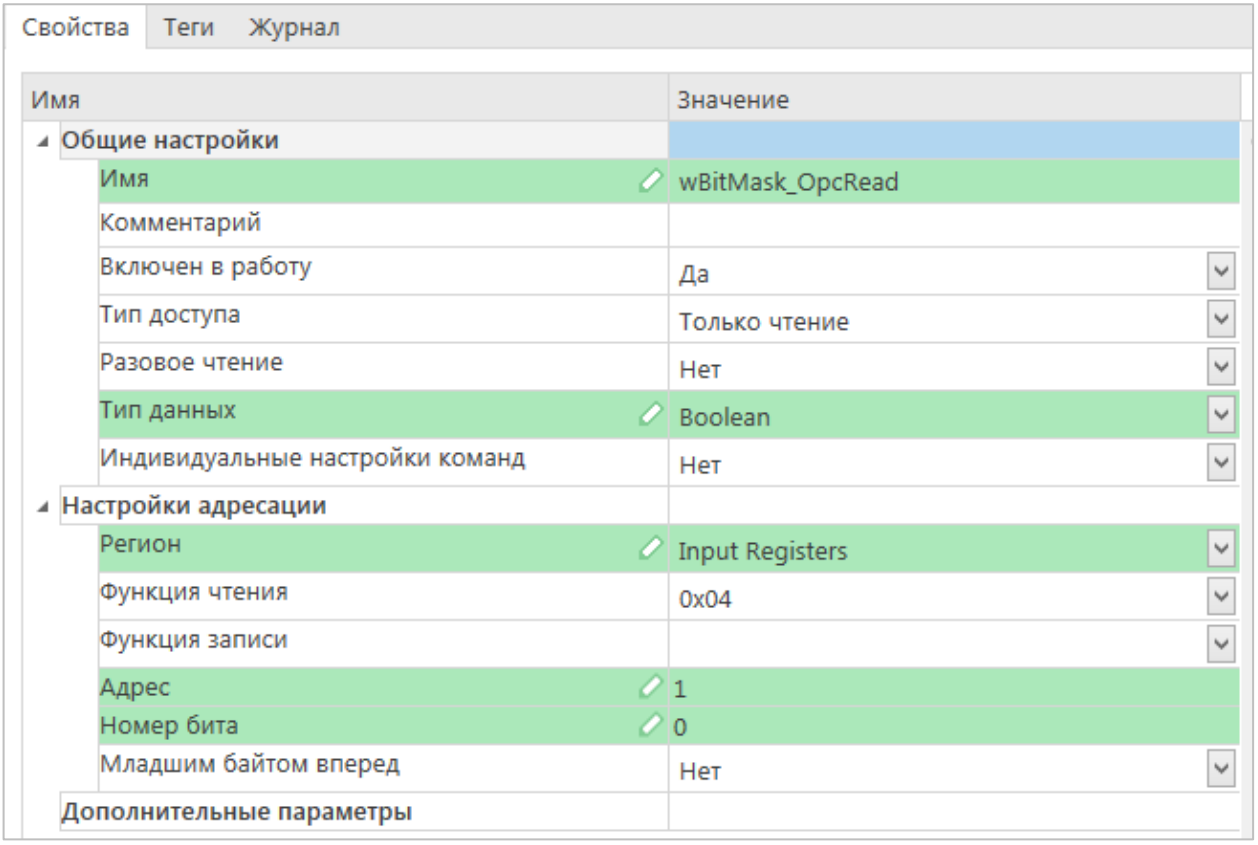


Рисунок 4.5.8 – Настройки тега wBitMask_OpcRead

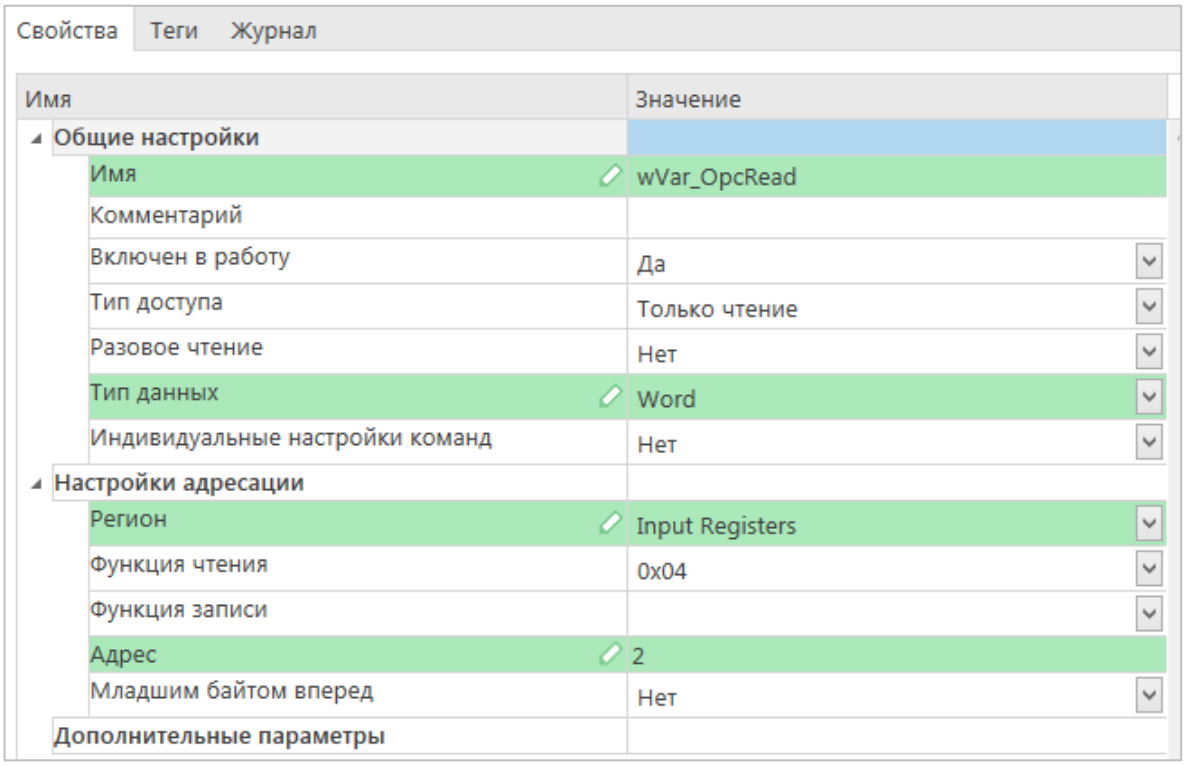


Рисунок 4.5.9 – Настройки тега wVar_OpcRead

Свойства | Теги | Журнал

Имя	Значение
Общие настройки	
Имя	rVar_OpcRead
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	3
Младшим байтом вперед	Нет
Младшим регистром вперед	Да
Дополнительные параметры	

Рисунок 4.5.10 – Настройки тега rVar_OpcRead

5. После добавления и настройки тегов сохранить конфигурацию OPC-сервера:

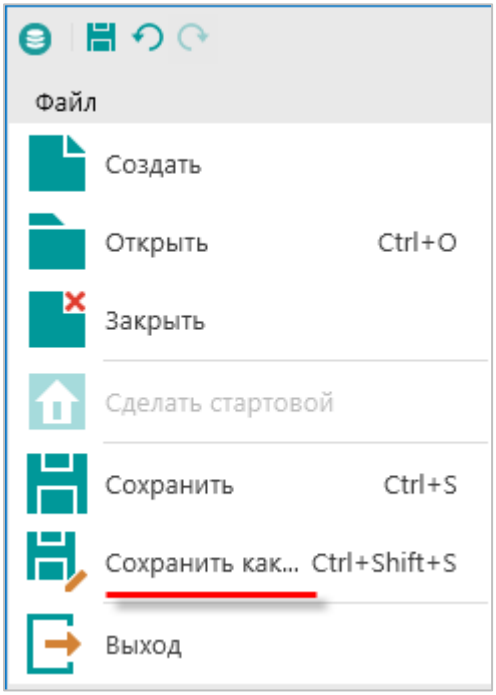


Рисунок 4.5.11 – Сохранение конфигурации OPC-сервера

Затем следует закрыть OPC-сервер. Запускать его не требуется, так как SCADA-система делает это автоматически.

Затем следует загрузить проект, созданный в п. 4.3.1 в контроллер и убедиться, что контроллер находится в одной локальной сети с OPC-сервером. После загрузки можно переходить к п. 4.6.

Созданная в пункте конфигурация доступна для скачивания: Example_OPC.zip

4.6 Подключение OPC-сервера к SCADA-системе

После настройки OPC-сервера следует подключить его к SCADA-системе. В качестве примера будет использована система MasterSCADA.

Для подключения OPC-сервера к SCADA-системе следует:

1. Запустить **MasterSCADA**. Нажать **ПКМ** на узел **Система** и добавить узел **Компьютер**:

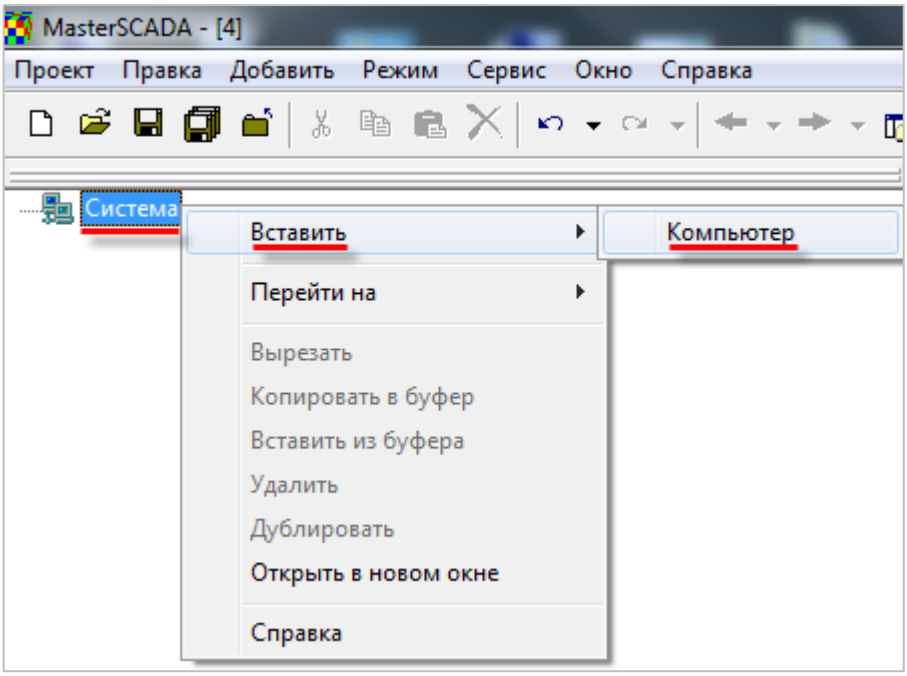


Рисунок 4.6.1 – Добавление узла Компьютер

2. Нажать **ПКМ** на узел **Компьютер** и выбрать команду **Вставить OPC-сервер**:

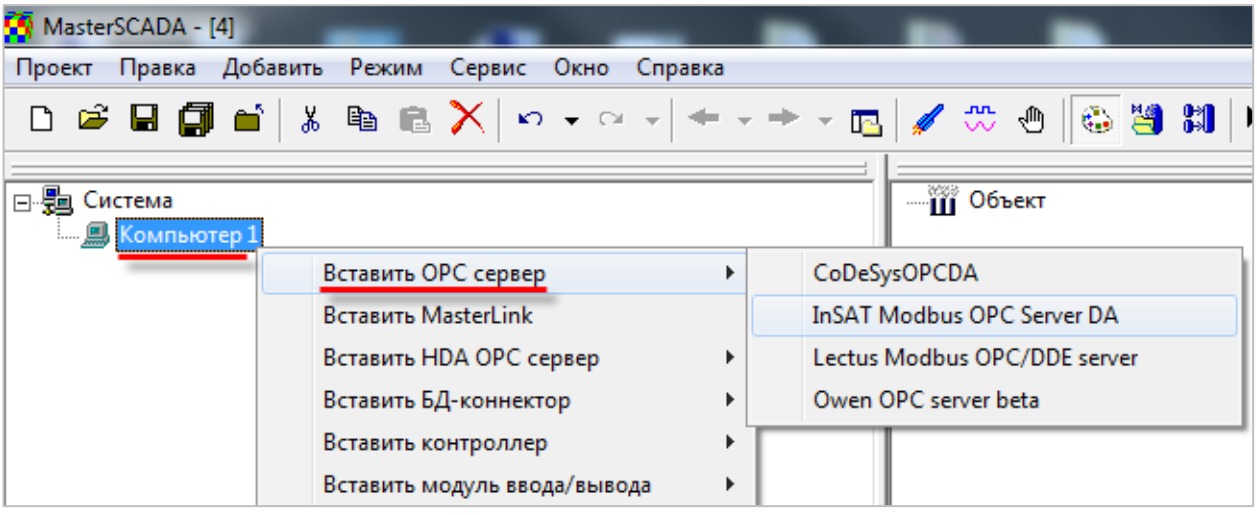


Рисунок 4.6.2 – Выбор OPC-сервера

Если нужный OPC-сервер не отображается в списке, то следует выполнить команду **Поиск OPC DA серверов**:

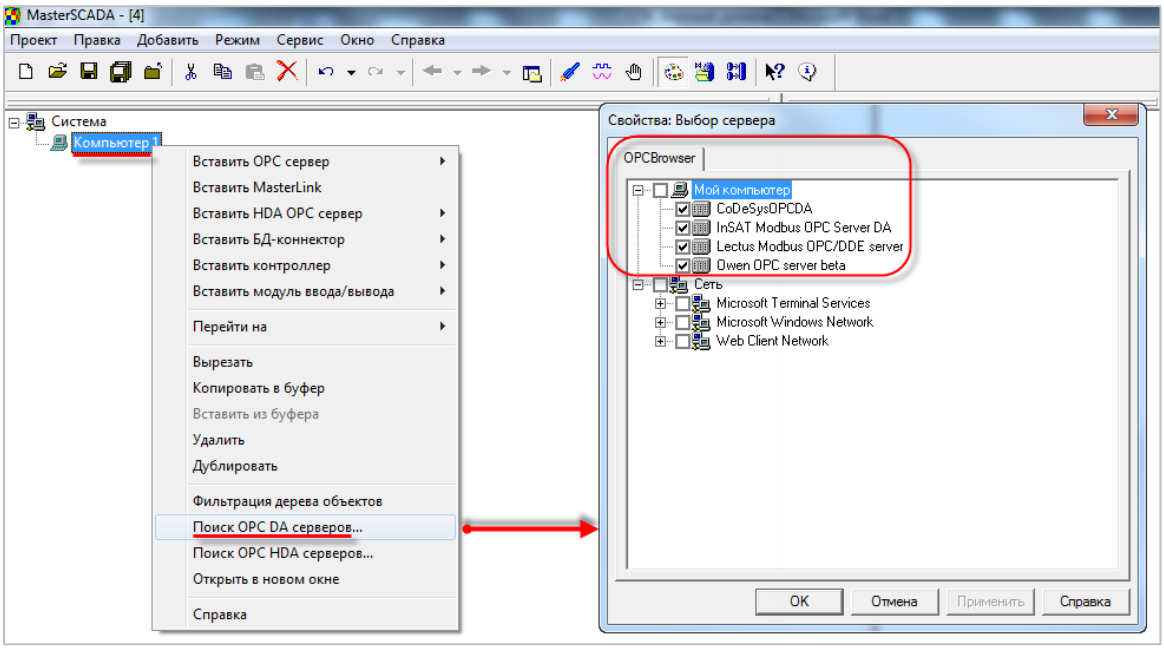


Рисунок 4.6.3 – Поиск установленных OPC-серверов

3. После добавления OPC-сервера следует нажать **ПКМ** на его название и выбрать команду **Вставить OPC переменные**.
   В появившемся диалоговом окне следует пометить галочками нужные переменные. Можно также выбрать папку – тогда в проект будут добавлены все переменные данной папки.

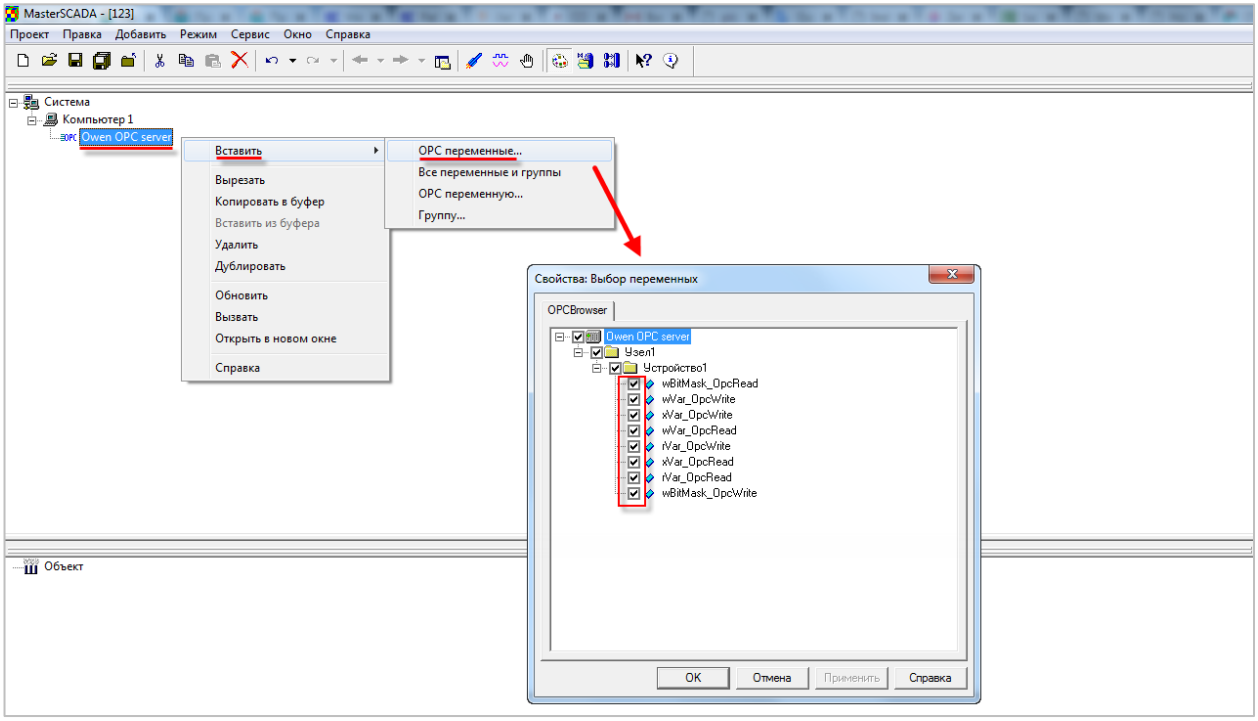


Рисунок 4.6.4 – Добавление переменных OPC-сервера

4. Запустить проект на исполнение. Подразумевается, что в контроллер загружен и запущен нужный проект, настроена связь между контроллером и ПК, на котором установлен OPC-сервер.

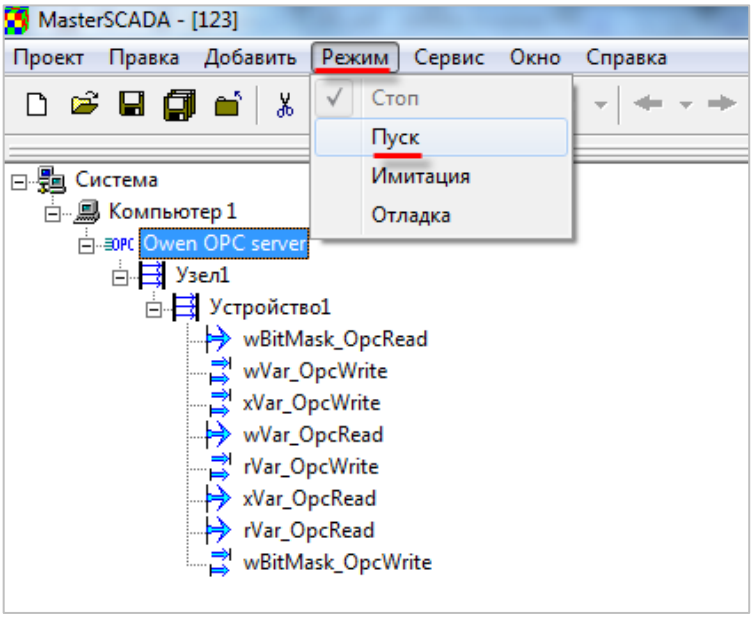


Рисунок 4.6.5 – Запуск проекта MasterSCADA на исполнение

5. В редакторе **CODEYS** следует изменить значения **OpcRead** переменных и наблюдать соответствующие изменения в **MasterSCADA**.

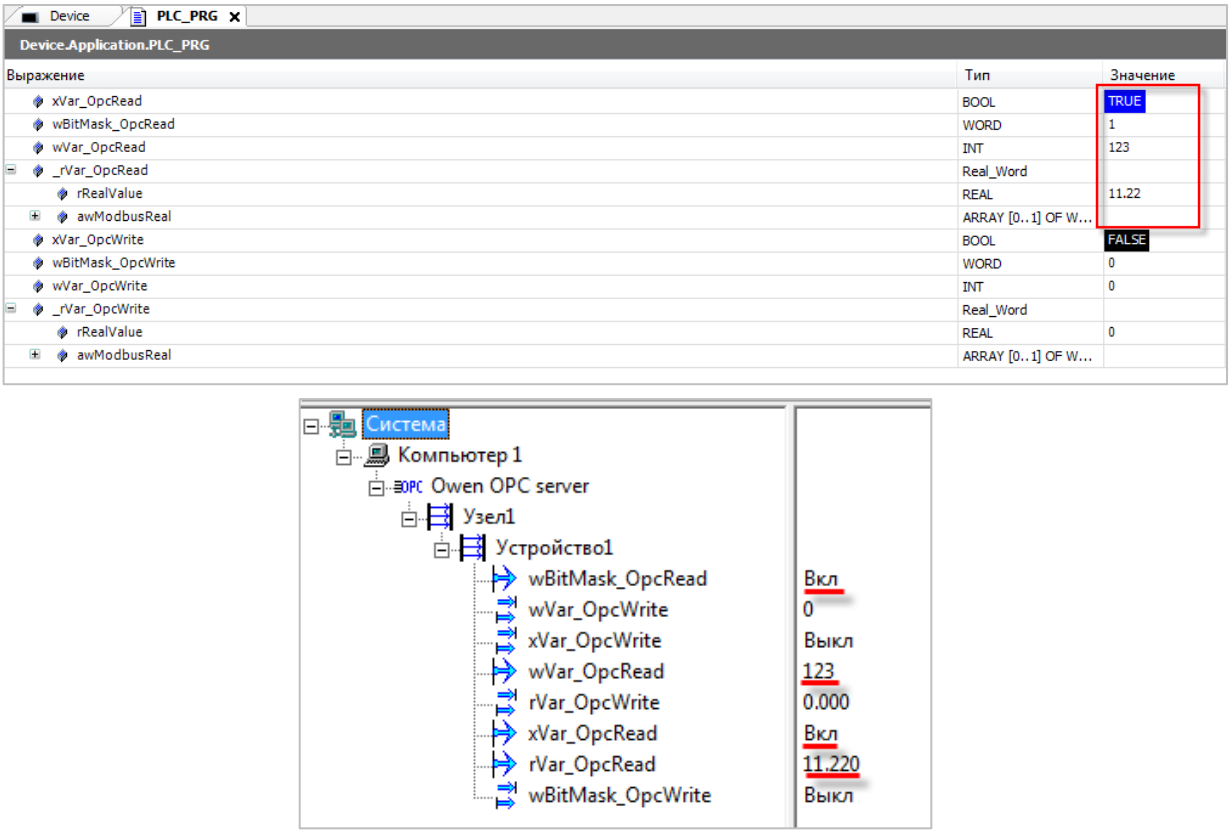


Рисунок 4.6.6 – Считывание данных из контроллера в MasterSCADA

6. В **MasterSCADA** следует изменить значения **OpcWrite** переменных и наблюдать соответствующие изменения в редакторе **CODESYS.**

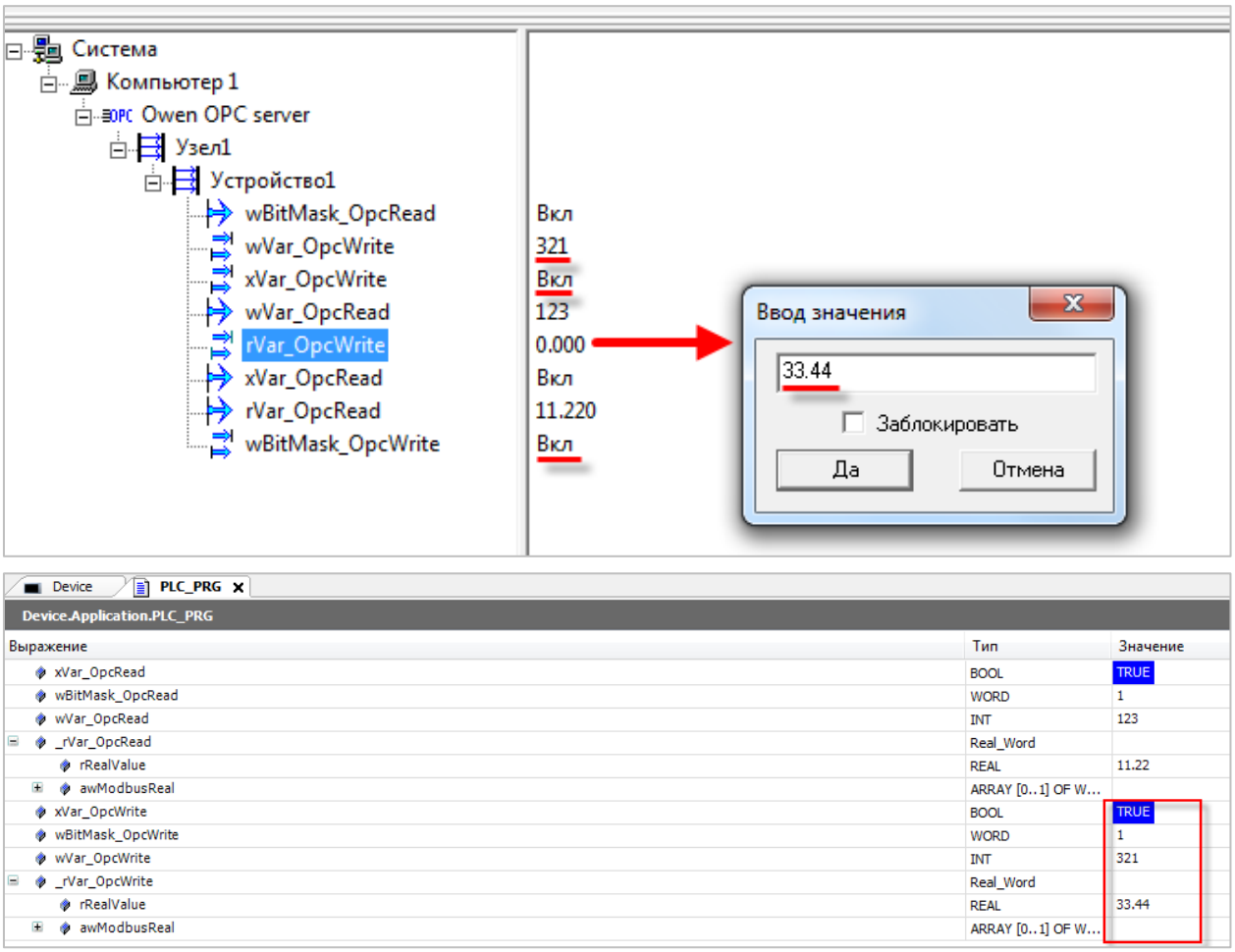


Рисунок 4.6.7 – Запись данных из MasterSCADA в контроллер

5 Облачный сервис OwenCloud

Облачный сервис OwenCloud применяется для удаленного мониторинга, управления и хранения архивов данных приборов, используемых в системах автоматизации. Приборы подключаются к сервису по интерфейсам **RS-485** (с помощью специальных сетевых шлюзов) или **Ethernet** (в этом случае требуется подключение приборов к сети с доступом к Интернету).

Для подключения контроллеров, программируемых в **CODESYS V3.5,** к сервису **OwenCloud** не требуется наличие сетевых шлюзов линейки Пх210. Доступ к облачному сервису осуществляется через подключение контроллера к локальной сети с доступом в Интернет. Для передачи данных используется протокол **Modbus TCP**.

Для подключения к **OwenCloud** следует:

1. Для настройки контроллера создать проект согласно п. 4.3.1.
2. В конфигурации компонента **Modbus TCP Slave Device** выбрать номер порта, который будет использоваться для связи с облачным сервисом (например, **1502**). Также можно указать количество доступных input- и holding-регистров Modbus.

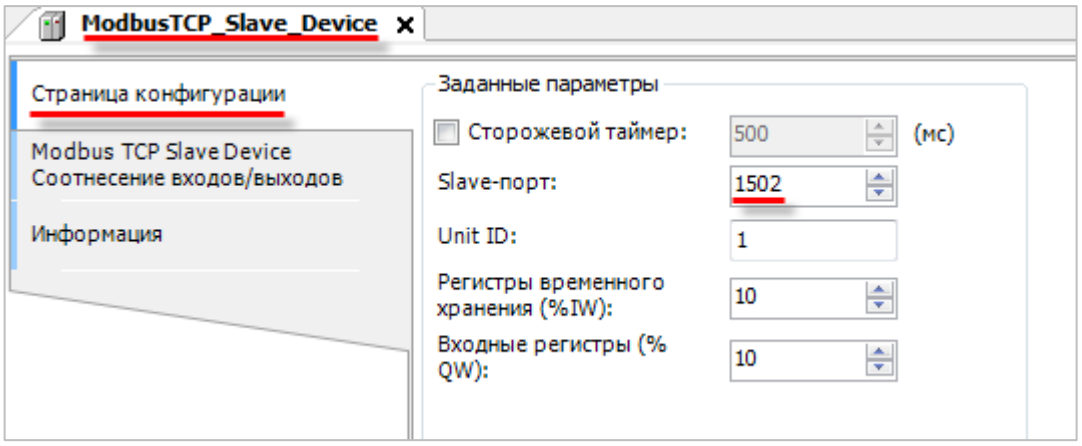


Рисунок 5.1 – Выбор порта для связи с OwenCloud в компоненте Modbus TCP Slave Device

3. В узле **OwenCloud** следует указать номер порта, заданный в компоненте **Modbus TCP Slave Device** (см. рисунок 5.1) и ввести токен прибора.
   На данном этапе токен отсутствует – он будет получен в пп. 6.

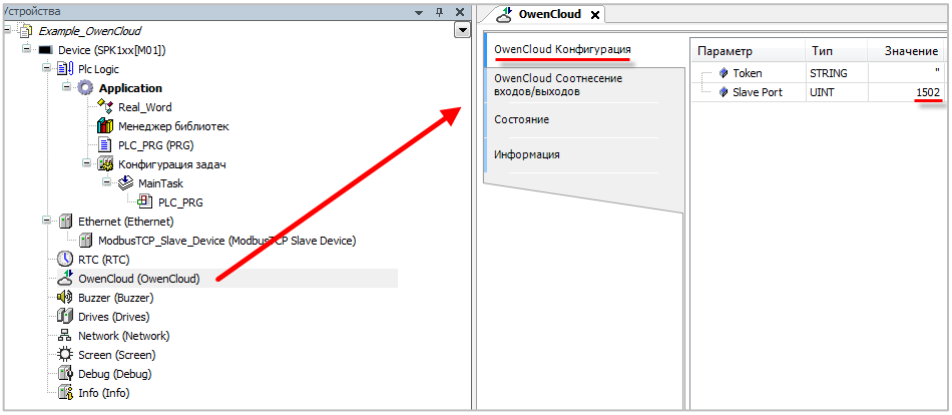
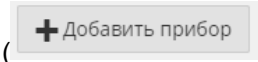


Рисунок 5.2 – Настройки узла OwenCloud

4. Зайти на главную страницу сервиса **OwenCloud**. Если вы еще не зарегистрированы в сервисе – необходимо пройти процедуру регистрации.
5. Перейти на страницу Администрирование, открыть вкладку **Приборы**, нажать кнопку **Добавить прибор** (

) и указать следующие настройки:
   - **Идентификатор** – ввести MAC-адрес контроллера (указан на корпусе прибора и в узле **Network** таргет-файла);
   - Тип прибора – выбрать тип Произвольное устройство Modbus;
   - **Заводской номер** – указать заводской номер прибора (заполнять необязательно);
   - **Название прибора** – ввести название прибора;
   - **Категории** – выбрать категории, к которым будет принадлежать прибор;
   - **Часовой пояс** – указать часовой пояс, в котором находится прибор.

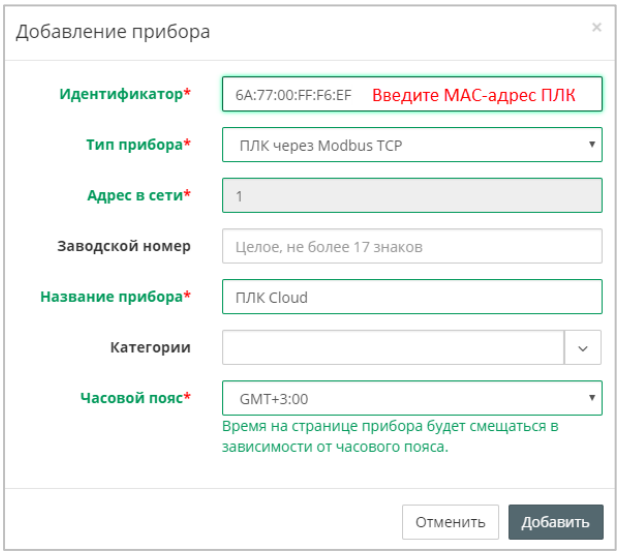


Рисунок 5.3 – Окно добавления прибора

Нажать кнопку Добавить.

6. На вкладке **Общие/Общие настройки** будет отображаться токен ПЛК. Токен следует скопировать в **CODESYS V3.5** в настройках узла **OwenCloud**:

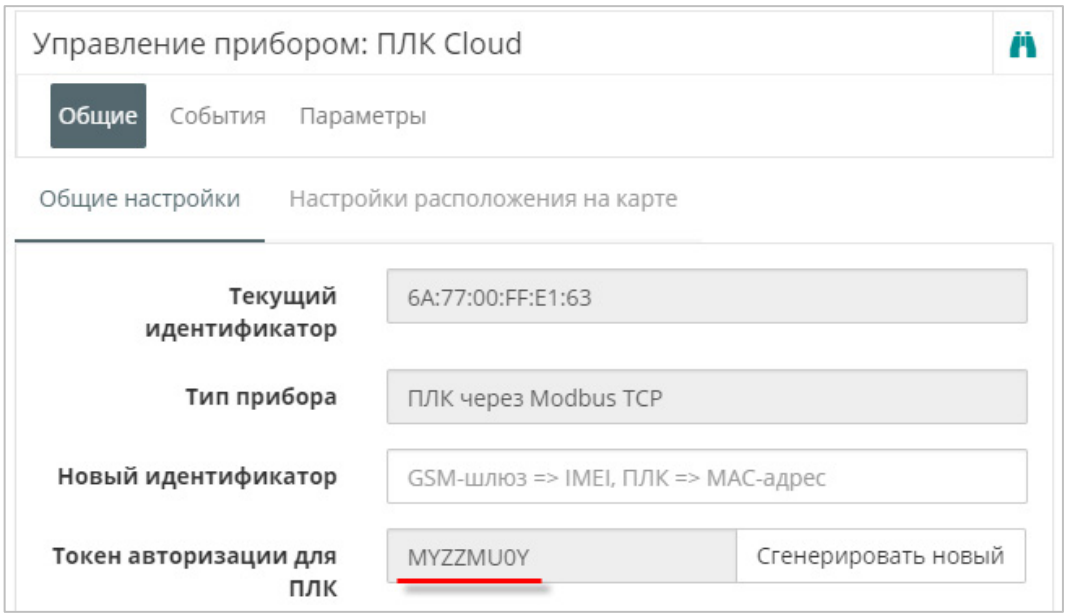


Рисунок 5.4 – Копирование токена из OwenCloud

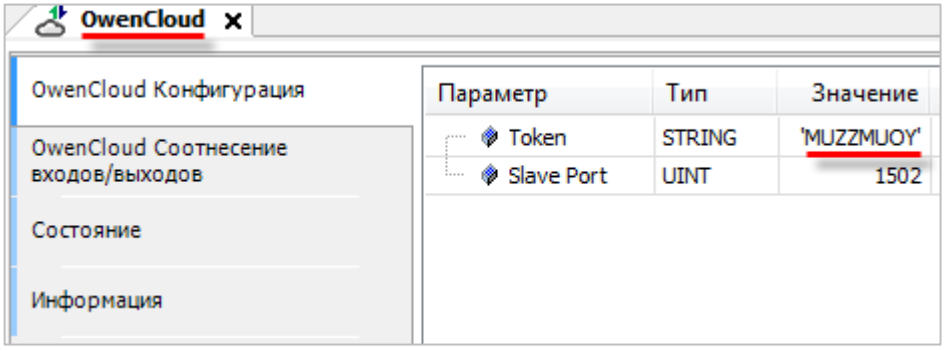


Рисунок 5.5 – Ввод токена в CODESYS V3.5

7. На вкладке **Параметры/Настройки параметров Modbus** добавить параметры в соответствии с рисунком 5.6.

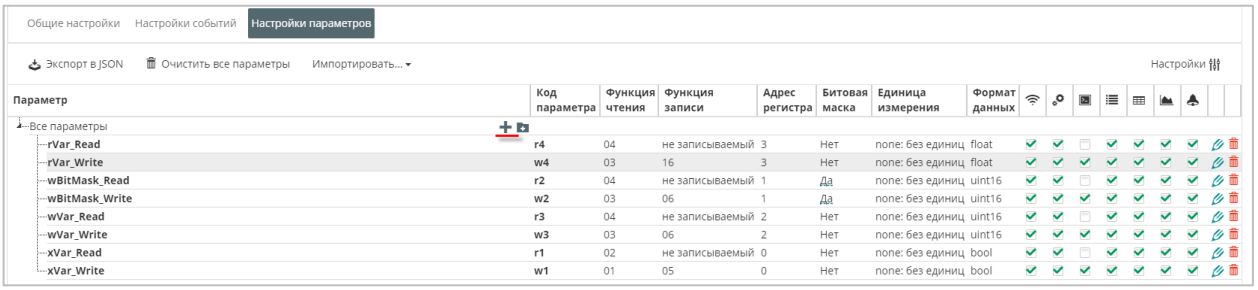


Рисунок 5.6 – Настройка параметров Modbus

ПРИМЕЧАНИЕ

Для параметров типа **Float** следует установить галочку **Младшим регистром вперед**.

8. Нажать на пиктограмму , чтобы перейти к просмотру значений параметров прибора. 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.

PLC_PRG

Device.Application.PLC_PRG

Выражение	Тип	Значение
xVar_OpcRead	BOOL	TRUE
wBitMask_OpcRead	WORD	1
wVar_OpcRead	INT	123
_rVar_OpcRead	Real_Word	
rRealValue	REAL	11.22
awModbusReal	ARRAY [0..1] OF W...	
xVar_OpcWrite	BOOL	FALSE
wBitMask_OpcWrite	WORD	0
wVar_OpcWrite	INT	0
_rVar_OpcWrite	Real_Word	

Параметры | Таблицы | Графики | События | Запись параметров

Параметр	Код параметра	Значение
Все параметры		
rVar_Read	r4	11.22
rVar_Write	w4	0.00
wBitMask_Read	r2	1
wBitMask_Write	w2	0
wVar_Read	r3	123
wVar_Write	w3	0
xVar_Read	r1	1
xVar_Write	w1	0

Рисунок 5.7 – Просмотр параметров прибора

Приложение А. Использование объединение (Union)

Стандарт **Modbus** предусматривает только два типа данных, участвующих в обмене – **BOOL** и **WORD**. Достаточно часто возникает потребность передать данные других типов, например, **REAL** и **STRING**. В этом случае на устройстве, которое отправляет данные, следует преобразовать их в последовательность **WORD** регистров. Соответственно, на устройстве, получающем данные, должно быть выполнено обратное преобразование. Наиболее простой способ сделать это в **CODESYS V3.5** – использовать объединения.

Объединение (UNION) представляет собой пользовательский тип данных, все переменные которого расположены в одной области памяти. Таким образом, переменные различных типов будут представлять различную интерпретацию одних и тех же данных. Для конвертации достаточно записать значение в одну из переменных объединения и считать его из другой.

Для конвертации значения с плавающей точкой, хранящегося в двух **WORD**, в переменную типа **REAL** следует:

1. Нажать **ПКМ** на узел **Application** и добавить объект **DUT** типа **объединение** с названием **Real_Word**:

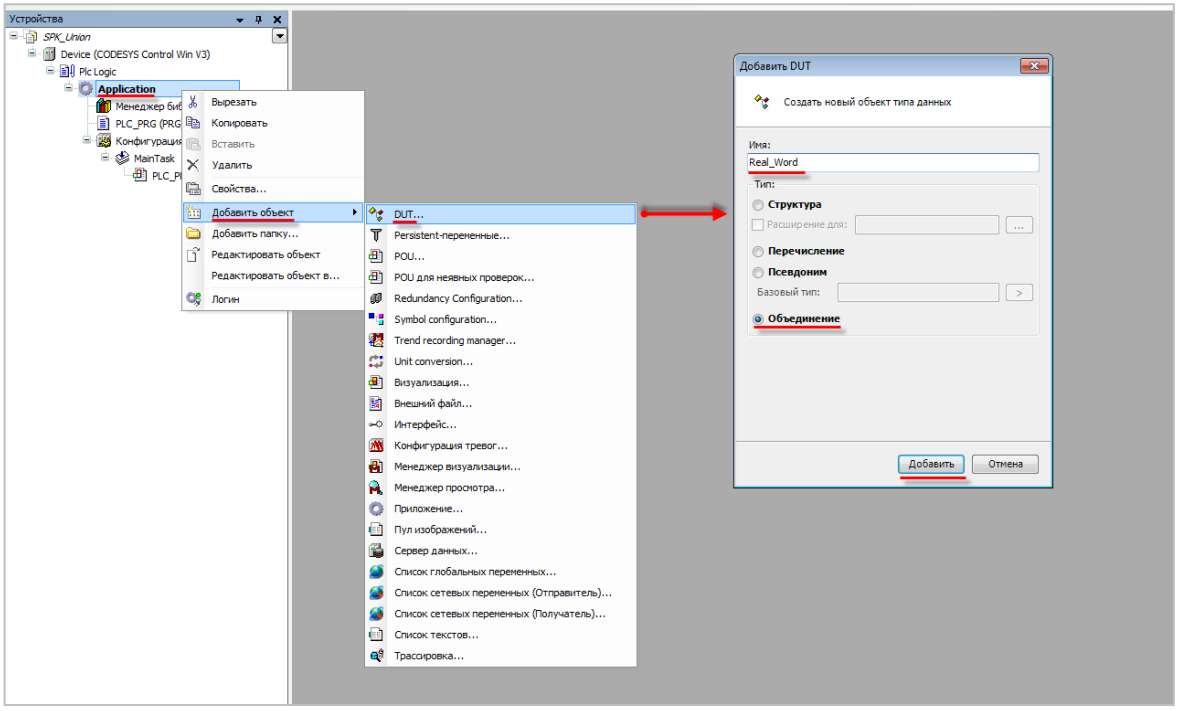


Рисунок А.1 – Добавление в проект объединения

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

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

Рисунок А.2 – Объявление переменных объединения

3. В программе объявить экземпляр объединения **Real_Word** с названием **_2WORD_TO_REAL**:

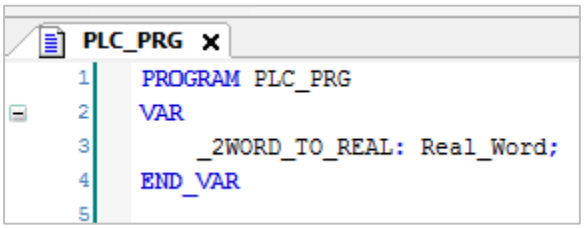


Рисунок А.3 – Объявление экземпляра объединения в программе

Для использования переменных объединения в нужном месте программы следует ввести имя экземпляра объединения и нажать точку, после чего выбрать из списка нужную переменную:

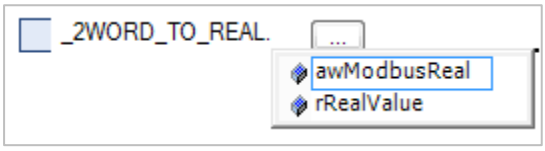


Рисунок А.4 – Работа с переменными объединения в программе

Переменные массива **awModbusReal** будут привязаны к регистрам во время настройки **Modbus**, а переменная **rRealValue** будет использоваться в программе для работы со значением с плавающей точкой.