CODESYS V3.5

Настройка связи между ПЛК

Руководство пользователя

24.05.2022
версия 3.0

Оглавление

Глоссарий

ПЛК – программируемый логический контроллер.

ПК – персональный компьютер.

ПКМ – правая кнопка мыши.

1 Цель документа

Настоящее руководство описывает настройку обмена данными между контроллерами, программируемыми в среде **CODESYS V3.5**. 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**.

Контроллеры, программируемые в среде CODESYS V3.5, обычно поддерживают несколько промышленных протоколов (например, Modbus). Но если требуется настроить обмен между несколькими ПЛК, то этот способ обычно оказывается не очень удобным – требуется отдельно настраивать оба ПЛК, формировать карту регистров, добавлять код для преобразования типов данных и т. д. Более простым вариантом является использование специфических коммуникационных компонентов CODESYS V3.5 – **сетевых переменных** или **менеджера источников данных**. Они позволяют организовать «бесшовный» обмен между ПЛК с минимальными затратами времени – один контроллер получает доступ к переменным других ПЛК и взаимодействует с ними так, как будто они являются его собственными.

Основное отличие сетевых переменных от менеджера источника данных:

- сетевые переменные используют широковещательную рассылку (UDP broadcast). Они удобны в тех случаях, когда требуется передать общий набор данных от одного ПЛК нескольким другим;
- менеджер источников данных использует TCP-подключение. Он удобен в тех случаях, когда ПЛК должен считывать/записать разные наборы данных из одного или нескольких других ПЛК.

Особенно удобным является добавление всех ПЛК в один проект CODESYS – это позволяет минимизировать время, требуемое на настройку обмена.

2 Сетевые переменные

2.1 Основные сведения о сетевых переменных

Сетевые переменные позволяют организовать обмен между несколькими контроллерами, программируемыми в **CODESYS V3.5** (и **CoDeSys V2.3**), 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 в одном устройстве список читаемых/записываемых переменных и импортировать его в 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 единственное устройство-отправитель 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 в новые устройства соответствующие списки. Обмен сетевыми переменными основан на отправке широковещательных (broadcast) запросах.

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

В п. 2.2 и п. 2.3 рассмотрены настройки компонентов Список сетевых переменных (отправитель) и Список сетевых переменных (получатель).

В п. 2.4 рассмотрен пример обмена сетевыми переменными между контроллерами, программируемыми в **CODESYS V3.5**

В п. 2.5 рассмотрен пример обмена сетевыми переменными между контроллерами, программируемыми в **CoDeSys V2.3** и **CODESYS V3.5**

В п. 2.6 описаны особенности использования сетевых переменных.

В п. 2.7 описан процесс диагностики обмена сетевыми переменными.

2.2 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**:

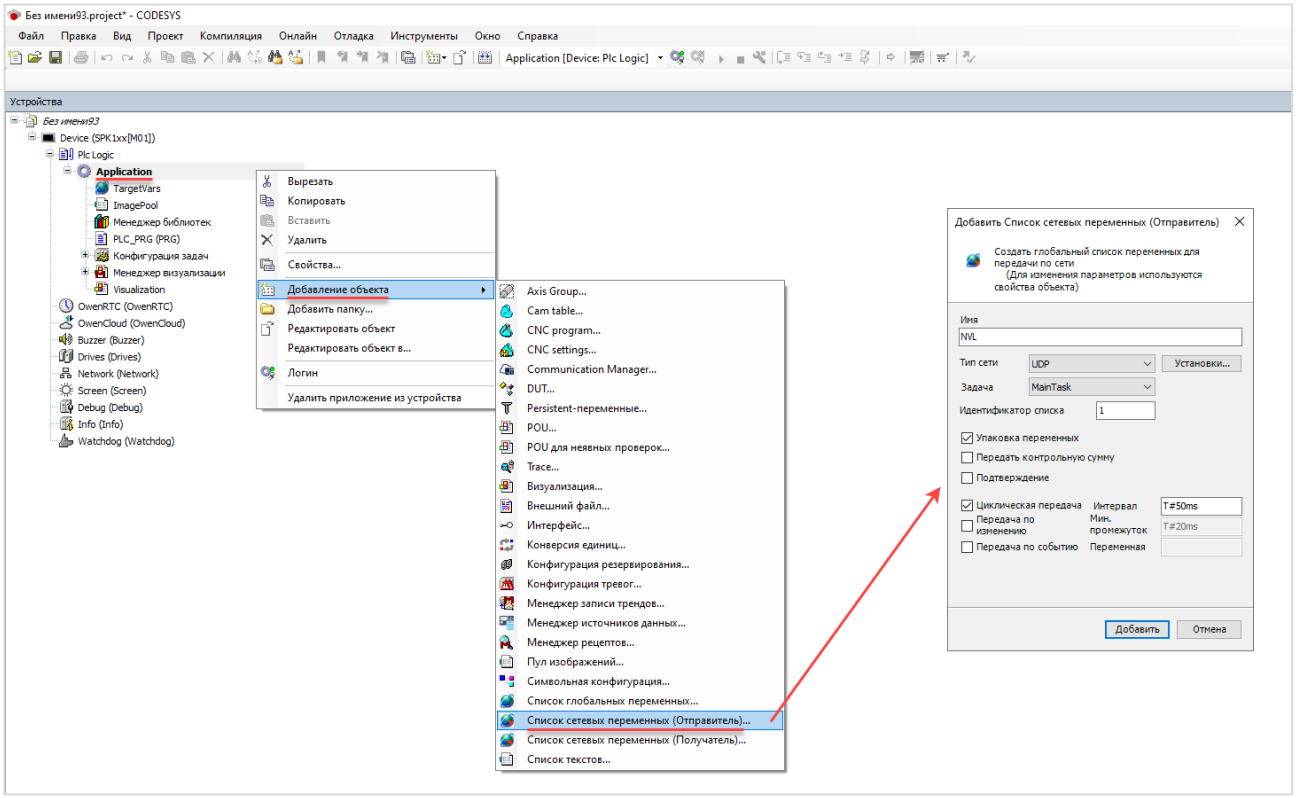

Рисунок 2.2.1 – Добавление компонента Список сетевых переменных (отправитель)

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

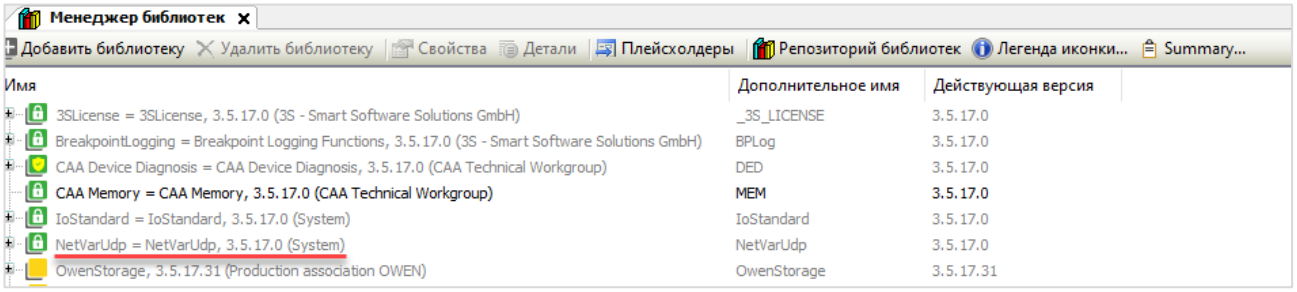

Рисунок 2.2.2 – Библиотека NetVarUdp в Менеджере библиотек

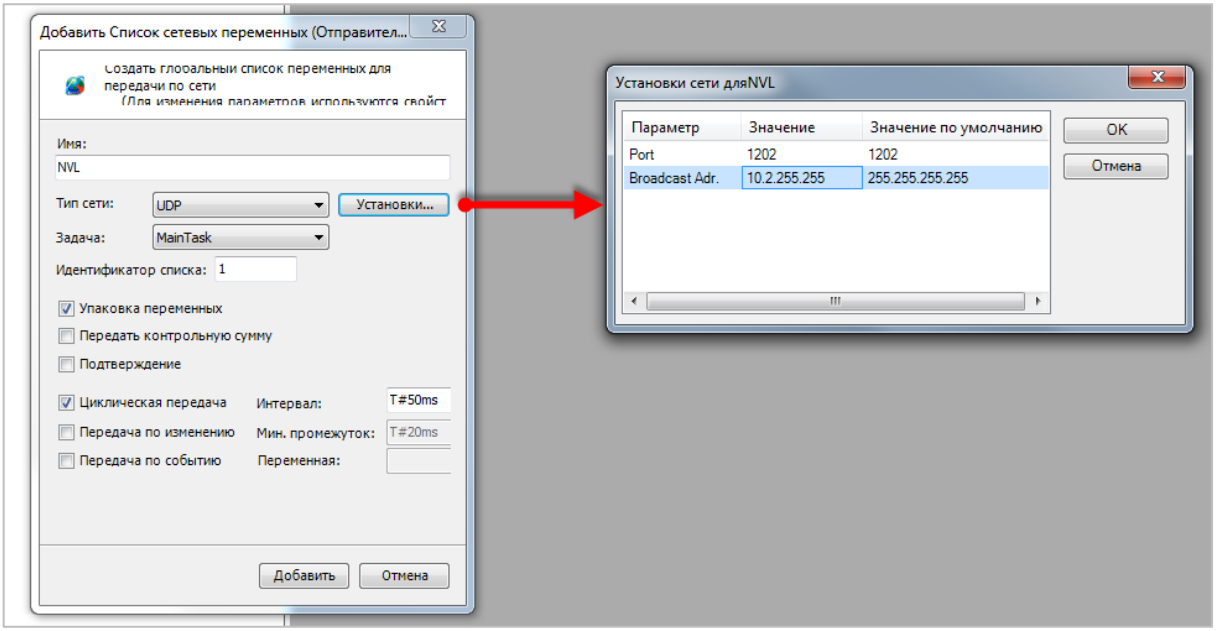

Рисунок 2.2.3 – Настройки компонента Список сетевых переменных (отправитель)

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 настройкам локальной сети.
Например, 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 будет являться любое устройство с IP-адресом **10.2.x.x**) или **10.2.11.255** (в данном случае получателем сетевых переменных будет являться любое устройство с IP-адресом **10.2.11.x**).

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

3. **Задача** – задача, к которой будет привязан процесс обмена сетевыми переменными.
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 – 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 сетевые переменные однократно отправляются вне зависимости от выполнения условий из пп. 8–10.

После создания списка следует наполнить его нужными переменными:

NVL

```
{attribute 'qualified_only'}
VAR_GLOBAL
    iVar:       INT;
    rVar:       REAL;
    adwVar:     ARRAY [0..15] OF DWORD;
END_VAR
```

Рисунок 2.2.4 – 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**.

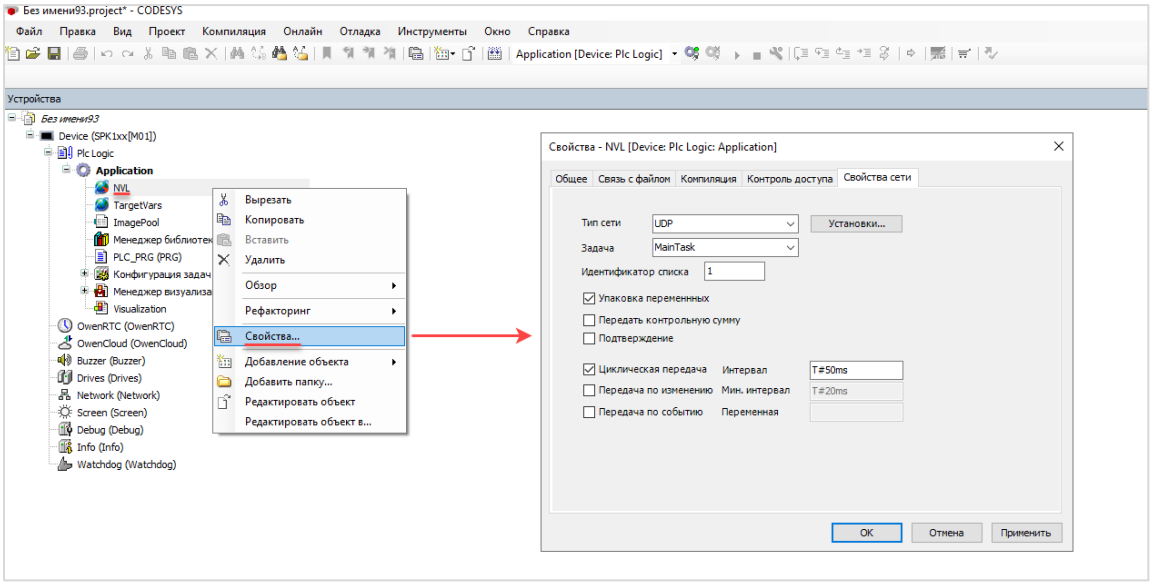

Рисунок 2.2.5 – Изменение настроек списка сетевых переменных

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:

```
<GVL>
  <Declarations><![CDATA[{attribute 'qualified_only'}
VAR_GLOBAL
	iVar:		INT;
	rVar:		REAL;
	adwVar:	ARRAY [0..15] OF DWORD;
END_VAR


]]></Declarations>
  <NetvarSettings Protocol="UDP">
    <ListIdentifier>1</ListIdentifier>
    <Pack>True</Pack>
    <Checksum>False</Checksum>
    <Acknowledge>False</Acknowledge>
    <CyclicTransmission>True</CyclicTransmission>
    <TransmissionOnChange>False</TransmissionOnChange>
    <TransmissionOnEvent>False</TransmissionOnEvent>
    <Interval>T#50ms</Interval>
    <MinGap>T#20ms</MinGap>
    <EventVariable>
    </EventVariable>
    <ProtocolSettings>
      <ProtocolSetting Name="Broadcast Adr." Value="10.2.11.255" />
      <ProtocolSetting Name="Port" Value="1202" />
    </ProtocolSettings>
  </NetvarSettings>
</GVL>
```

Рисунок 2.2.6 – Содержимое файла формата .gvl

2.3 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**:

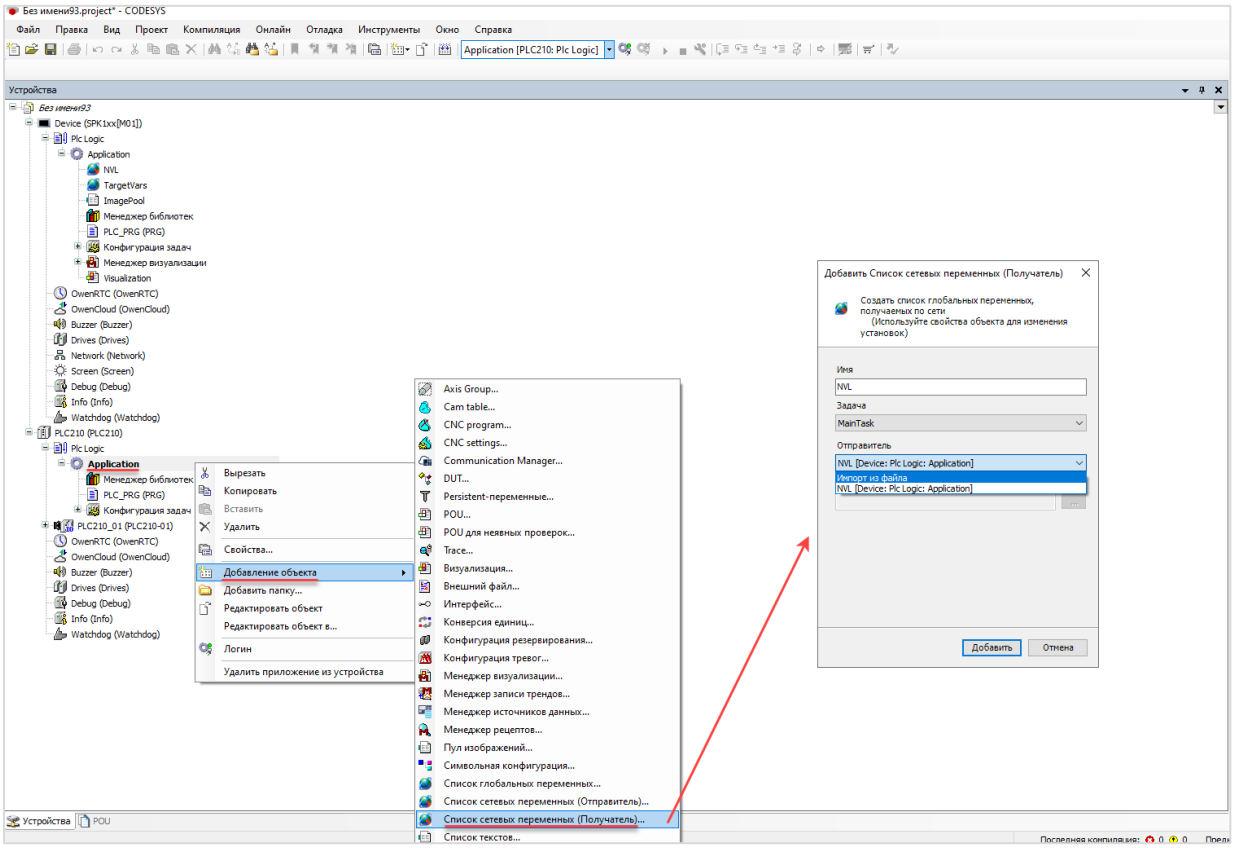

Рисунок 2.3.1 – Добавление компонента Список сетевых переменных (получатель)

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 2.2.6).

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

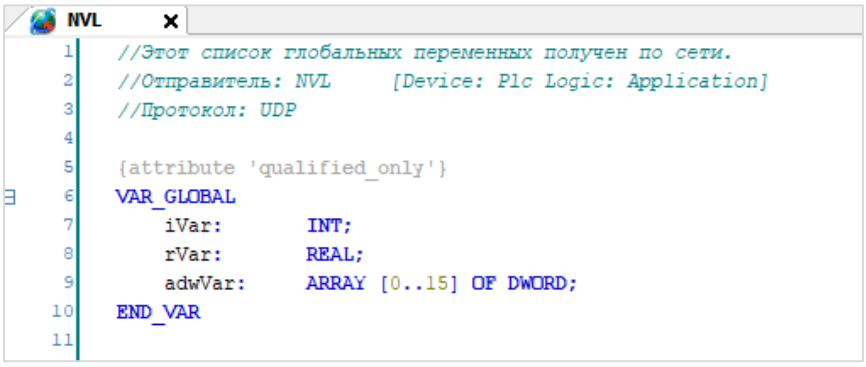

Рисунок 2.3.2 – Добавление компонента Список сетевых переменных (получатель)

Переменные этого списка можно использовать в программе контроллера-получателя – при наличии обмена между устройствами они будут иметь те же значения, что и переменные в аналогичном списке контроллера-отправителя.

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

В качестве примера будет рассмотрен обмен сетевыми переменными между контроллерами **СПК1xx [M01]** и **ПЛК210**.

Пример доступен для скачивания: Example_NetworkVariables_3517v1.zip

В примере также добавлен обмен с ПЛК, программируемом в среде **CoDeSys V2.3** (см. п. 2.5).

Сетевые параметры и используемые переменные приведены в таблице 2.1.

Таблица 2.1 – Сетевые параметры и переменные примера

Параметр	СПК1xx [M01]	ПЛК210
IP-адрес	10.2.11.174	10.2.25.2
Порт UDP	1202	
Broadcast адрес	10.2.255.255	
Названия списков сетевых переменных	SpkToPlc210 (отправление) Plc210ToSpk (получение)	Plc210ToSpk (отправление) SpkToPlc210 (получение)
Идентификаторы списков	1 (отправление) 2 (получение)	2 (отправление) 1 (получение)
Отправляемая сетевая переменная	wSpkToPlc210	wPlc210ToSpk
Получаемая сетевая переменная	wPlc210ToSpk	wSpkToPlc210

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) с названием **SpkToPlc210** с настройками в соответствии с таблицей 2.1:

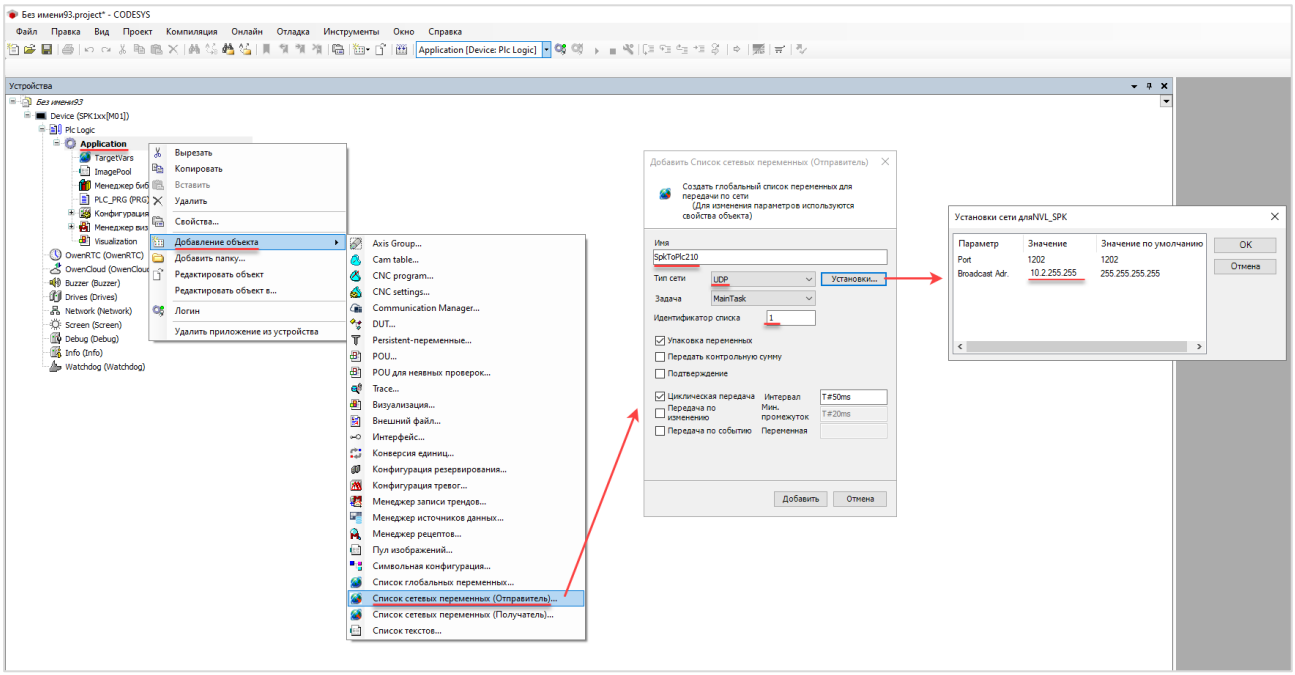

Рисунок 2.4.1 – Добавление и настройка списка отправляемых сетевых переменных для СПК1xx [M01]

В созданном списке объявить переменную **wSpkToPlc210** типа **WORD**:

```
{attribute 'qualified_only'}
VAR_GLOBAL
    wSpkToPlc210: WORD;
END_VAR
```

Рисунок 2.4.2 – Объявление отправляемых сетевых переменных для СПК1xx [M01]

3. Нажать **ПКМ** на имя проекта, использовать команду **Добавить устройство** и выбрать модель контроллера **ПЛК210**:

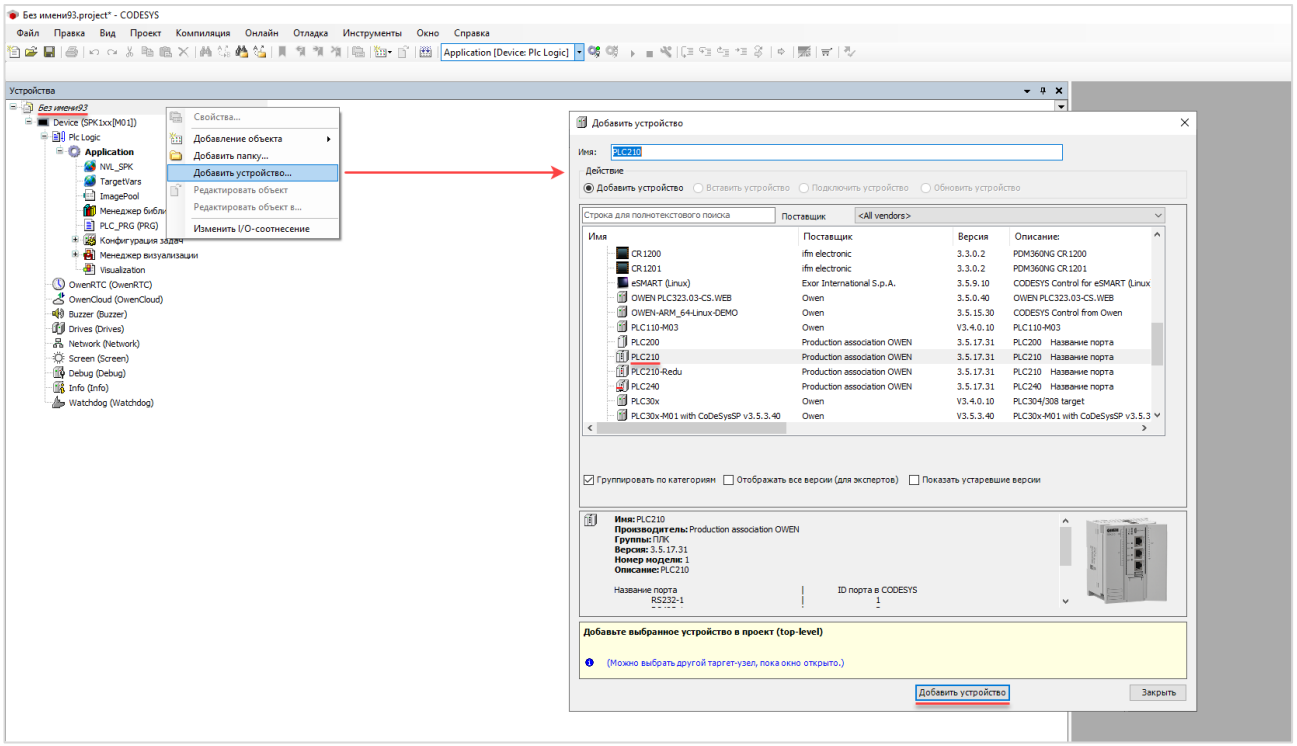

Рисунок 2.4.3 – Добавление в проект ПЛК210

4. Нажать **ПКМ** на компонент **Конфигурация задач** и добавить задачу с названием **MainTask** и настройками по умолчанию:

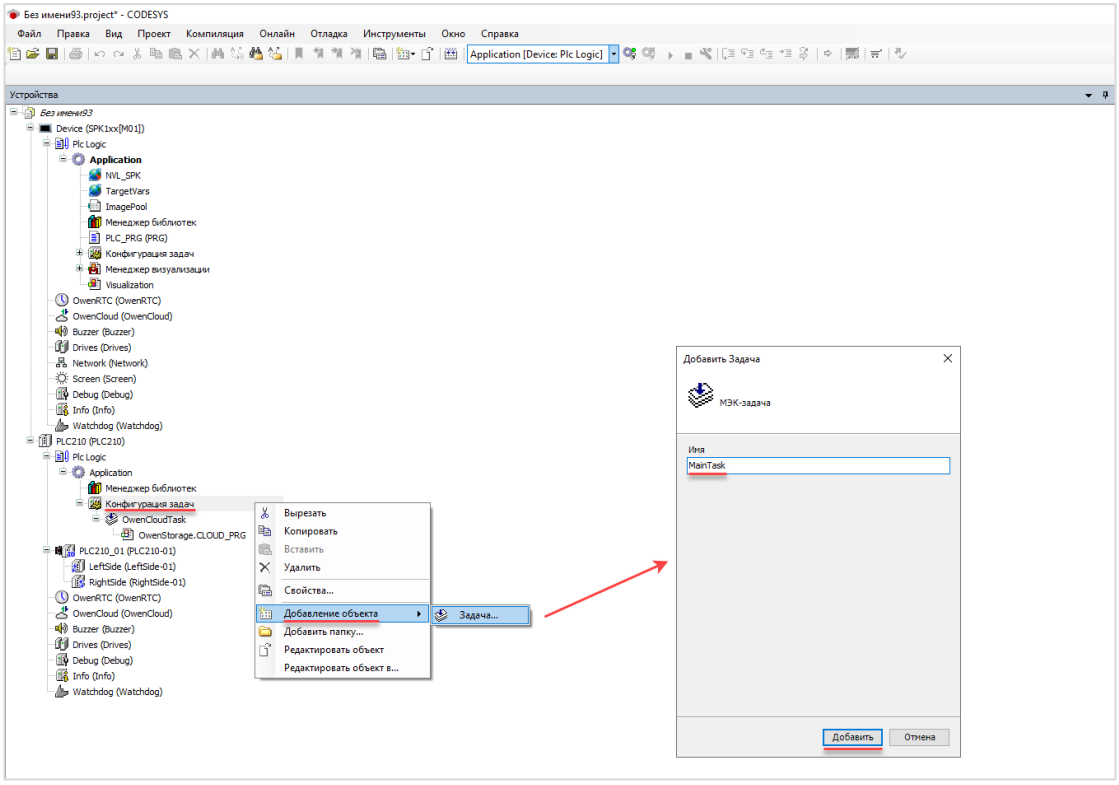

Рисунок 2.4.4 – Создание задачи

5. В устройстве ПЛК210 добавить компонент [Список сетевых переменных (получатель)] с импортом из устройства СПК1xx [M01]:

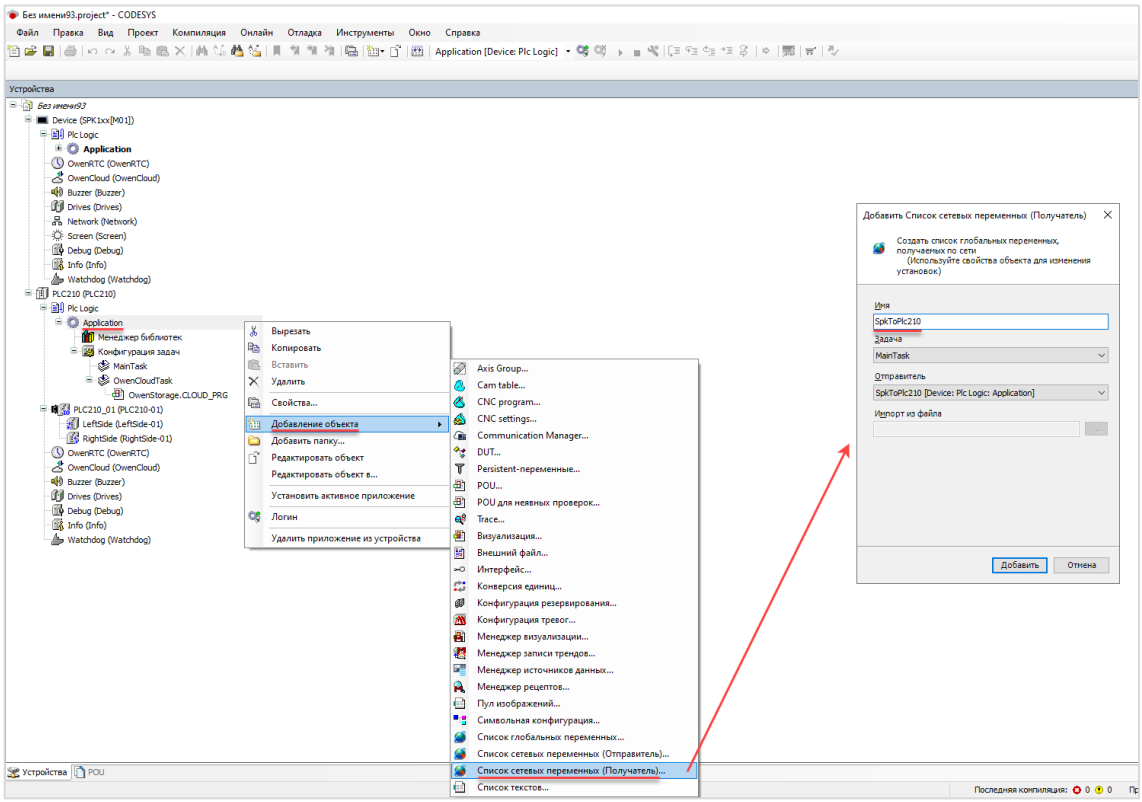

Рисунок 2.4.5 – Добавление списка получаемых сетевых переменных

6. В устройстве **ПЛК210** добавить компонент Список сетевых переменных (отправитель) с настройками в соответствии с таблицей 2.1:

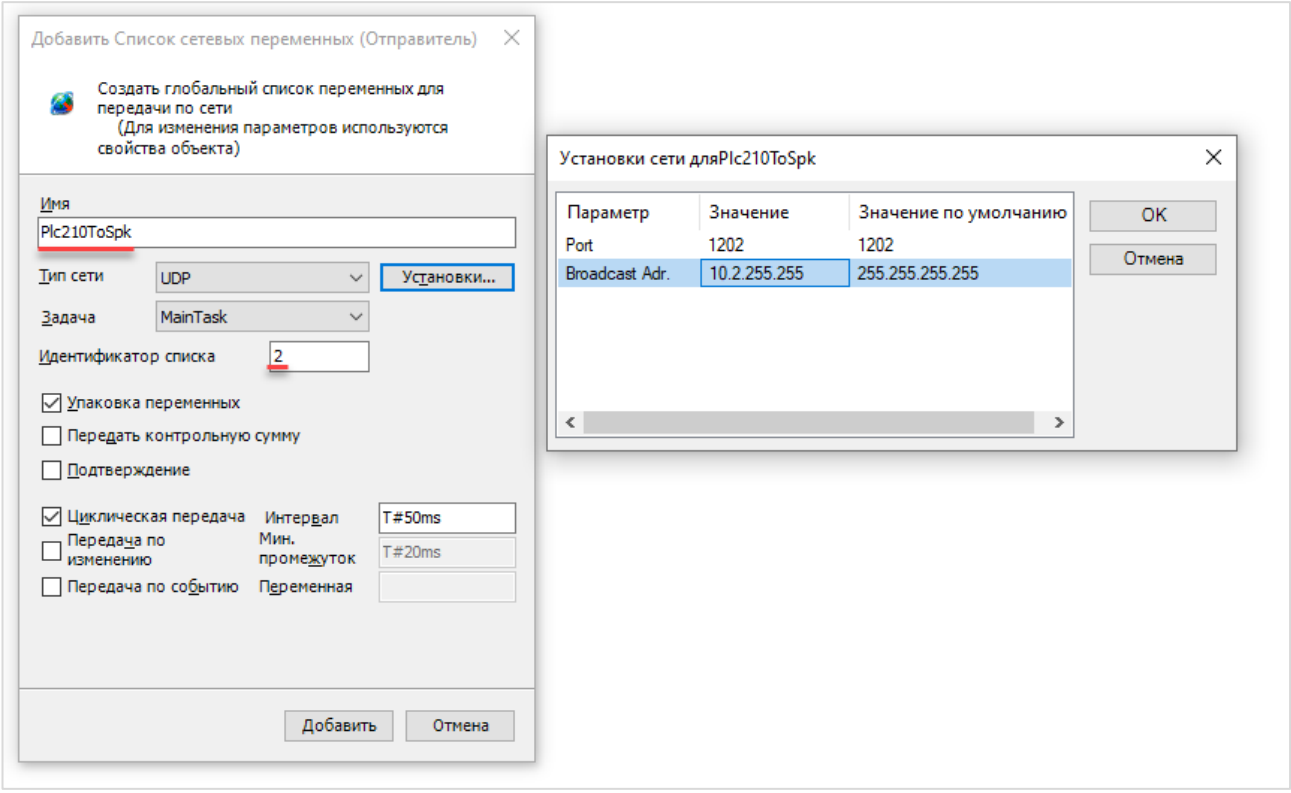

Рисунок 2.4.6 – Настройки списка отправляемых сетевых переменных для СПК1xx [M01]

В созданном списке объявить переменную **wPlc210ToSpk** типа **WORD**:

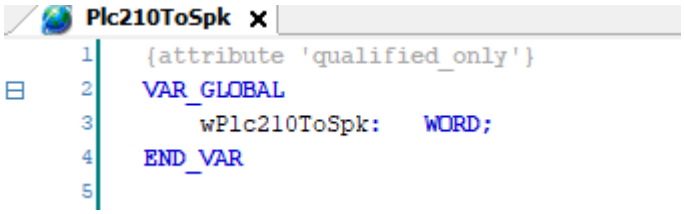

Рисунок 2.4.7 – Объявление отправляемых сетевых переменных для СПК1xx [M01]

7. В устройстве **СПК1xx [M01]** добавить компонент **Список сетевых переменных (получатель)** с импортом из устройства **ПЛК210**:

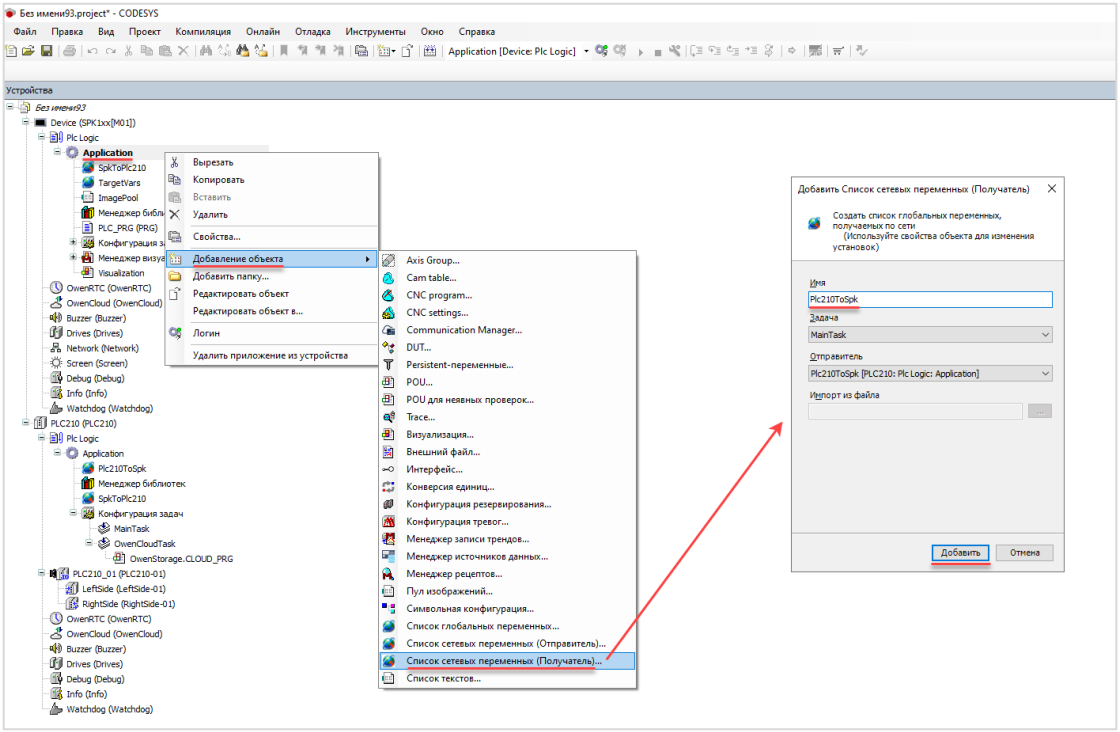

Рисунок 2.4.8 – Импорт списка сетевых переменных из СПК1xx [M01] в ПЛК210

8. Загрузить проекты в оба устройства и запустить их. Чтобы произвести сканирование сети для конкретного устройства – следует сначала выбрать его приложение с помощью выпадающего списка на панели инструментов:

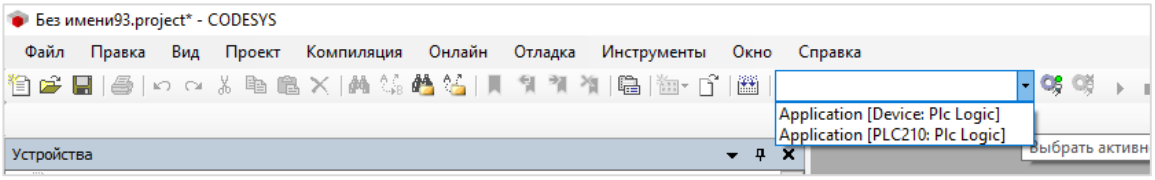

Рисунок 2.4.9 – Выбор активного приложение проекта

9. В устройстве **СПК1xx [M01]** в списке **SpkToPlc210** изменить значение переменной. Проверить, что оно изменилось в соответствующем списке в устройстве ПЛК210.

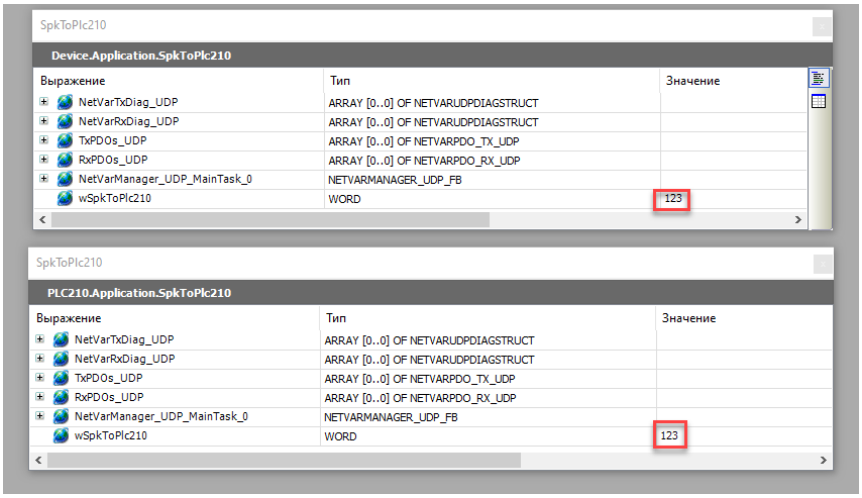

Рисунок 2.4.10 – Передача сетевых переменных из СПК1xx [M01] в ПЛК210

В устройстве **ПЛК210** в списке **Plc210ToSpk** изменить значение переменной. Проверить, что оно изменилось в соответствующем списке в устройстве ПЛК210.

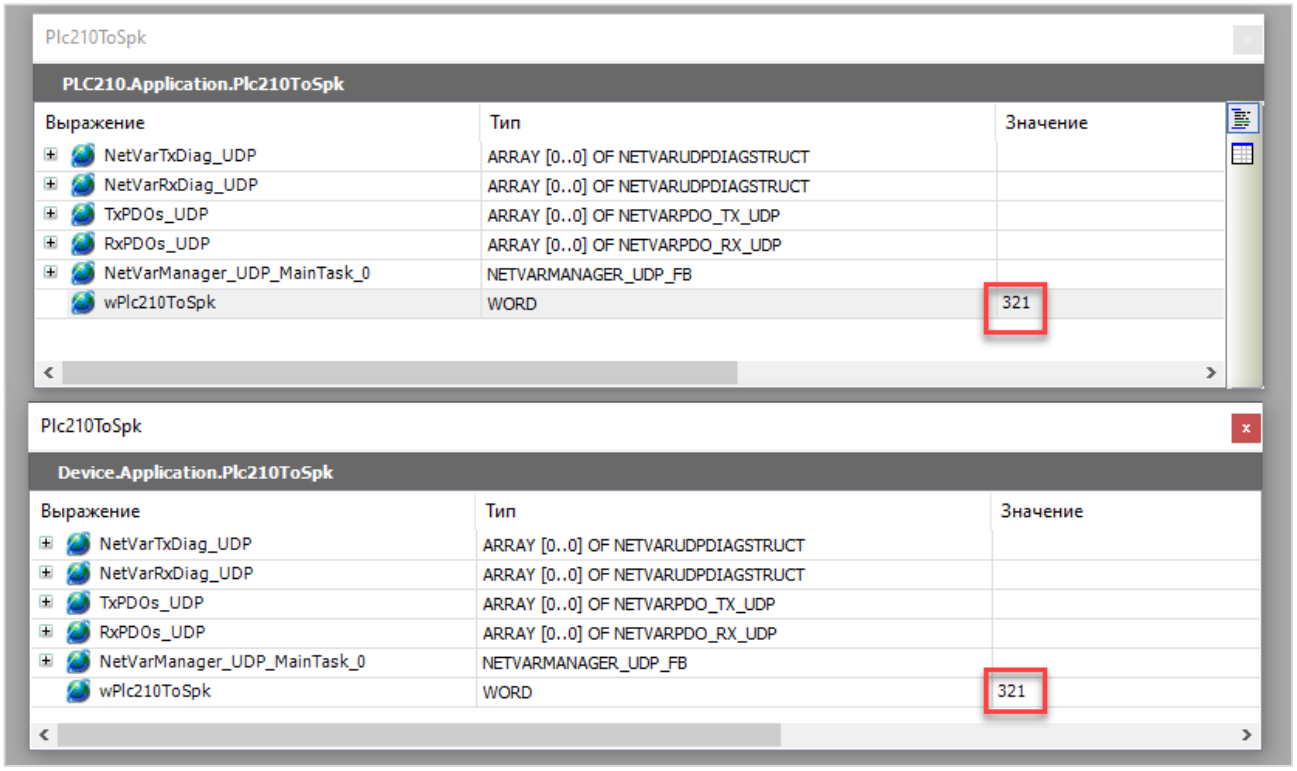

Рисунок 2.4.11 – Передача сетевых переменных из ПЛК210 в СПК1xx [M01]

2.5 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 **СПК1xx [M01]** (программируется в **CODESYS V3.5**) и **ПЛК110 [M02]** (программируется в **CoDeSys V2.3**).
Для СПК1xx [M01] используется пример, созданный в предыдущем пункте.

Пример доступен для скачивания: Example_NetworkVariables_3517v1.zip

Сетевые параметры и используемые переменные приведены в таблице 2.2.

Таблица 2.2 – Сетевые параметры и переменные примера

Параметр	СПК1xx [M01]	ПЛК110 [M02]
IP-адрес	10.2.11.174	10.2.11.176
Порт UDP	1202	
Broadcast адрес	10.2.255.255	
Названия списков сетевых переменных	SpkToPlc210[1] (отправление) Plc110ToSpk (получение)	Plc110ToSpk (отправление) SpkToPlc210 (получение)
Идентификатор списка	1 (отправление) 3 (получение)	3 (отправление) 1 (получение)
Отправляемая сетевая переменная	wSpkToPlc210	wPlc110ToSpk
Получаемая сетевая переменная	wPlc110ToSpk	wSpkToPlc210

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, но если вы используете язык ST – то добавьте в код хотя бы одно выражение (например, оператор «;», иначе проект не скомпилируется).
2. Во вкладке **Ресурсы** открыть узел **Настройки целевой платформы** и во вкладке **Сетевая функциональность** указать поддержку сетевого интерфейса **UDP**:

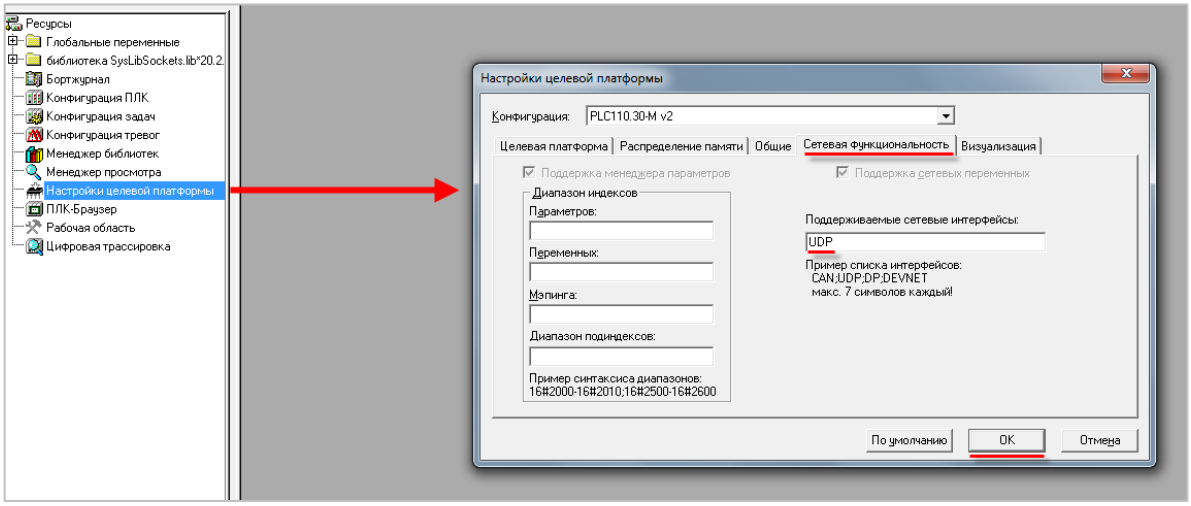

Рисунок 2.5.1 – Включение поддержки сетевых переменных в CoDeSys V2.3

[1] Упоминание Plc210 связано с тем, что на стороне СПК используется проект с такими названиями, созданный в п. 2.4.

3. Во вкладке **Ресурсы** открыть узел **Менеджер библиотек**, нажать **ПКМ** на свободное поле рядом с названиями библиотек, использовать команду **Добавить библиотеку** и добавить в проект библиотеку **NetVarUdp**:

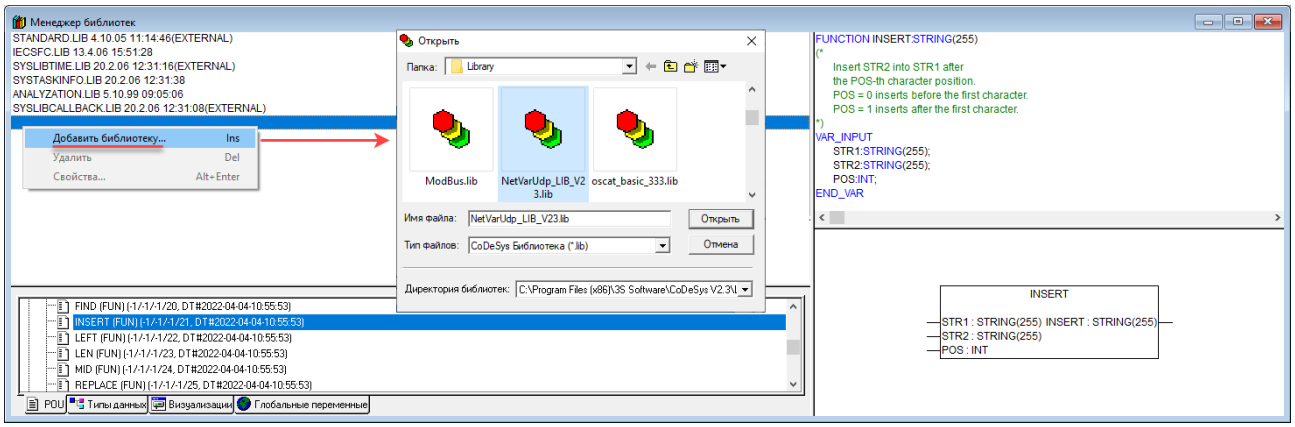

Рисунок 2.5.2 – Добавление библиотеки NetVarUdp в менеджере библиотек

4. Нажать **ПКМ** на папку **Глобальные переменные** и создать список отправляемых сетевых переменных **Plc110ToSpk** с настройками в соответствии с [таблицей 3.2]:

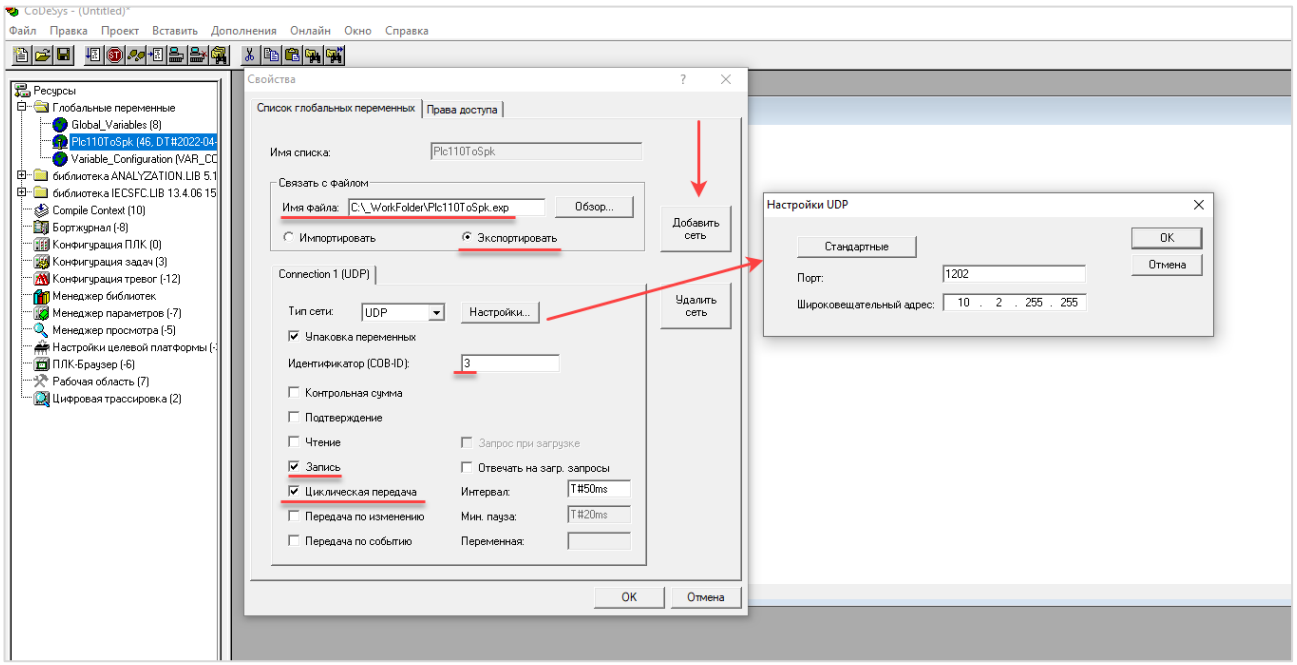

Рисунок 2.5.3 – Настройки списка отправляемых сетевых переменных для ПЛК110 [M02]

В созданном списке объявить переменную **wPlc110ToSpk** типа **WORD**:

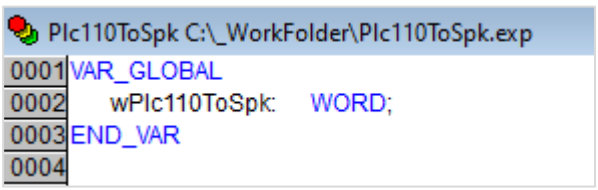

Рисунок 2.5.4 – Объявление отправляемых сетевых переменных для ПЛК110 [M02]

5. Нажать **ПКМ** на папку **Глобальные переменные** и создать список получаемых сетевых переменных **SpkToPlc210** настройками в соответствии с таблицей 3.2 (фактически этот список будет соответствовать списку, который был создан для СПК в п. 2.4):

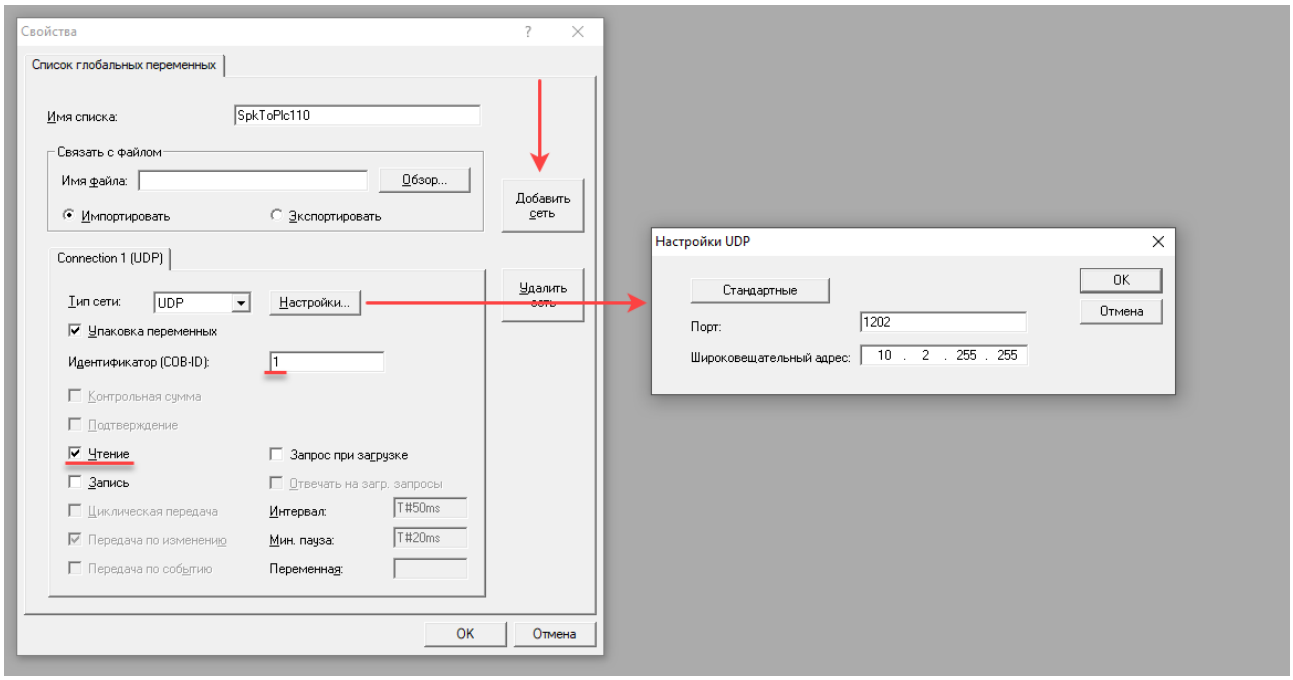

Рисунок 2.5.5 – Настройки списка получаемых сетевых переменных для ПЛК110 [M02]

В созданном списке объявить переменную **wSpkToPlc210** типа **WORD**:

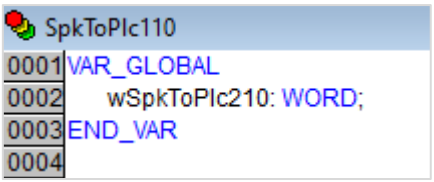

Рисунок 2.5.6 – Объявление получаемых сетевых переменных для ПЛК110 [M02]

6. Открыть проект для **СПК1xx [M01**], созданный в п. 2.4, в среде **CODESYS V3.5** (язык программы не имеет значения, поскольку проект не будет содержать программы).
7. Импорт списка сетевых переменных из CoDeSys V2.3 в CODESYS V3.5 не поддерживается. Поэтому следует создать в **CODESYS V3.5** список отправляемых переменных, аналогичный тому, который был создан в **CoDesys V2.3**, экспортировать его в файл и импортировать в список получаемых переменных. Это процедура описана в следующих подпунктах.

8. Добавить компонент Список сетевых переменных (отправитель) **Plc110ToSpk** с настройками, соответствующими одноименному списку из **CoDeSys V2.3**:

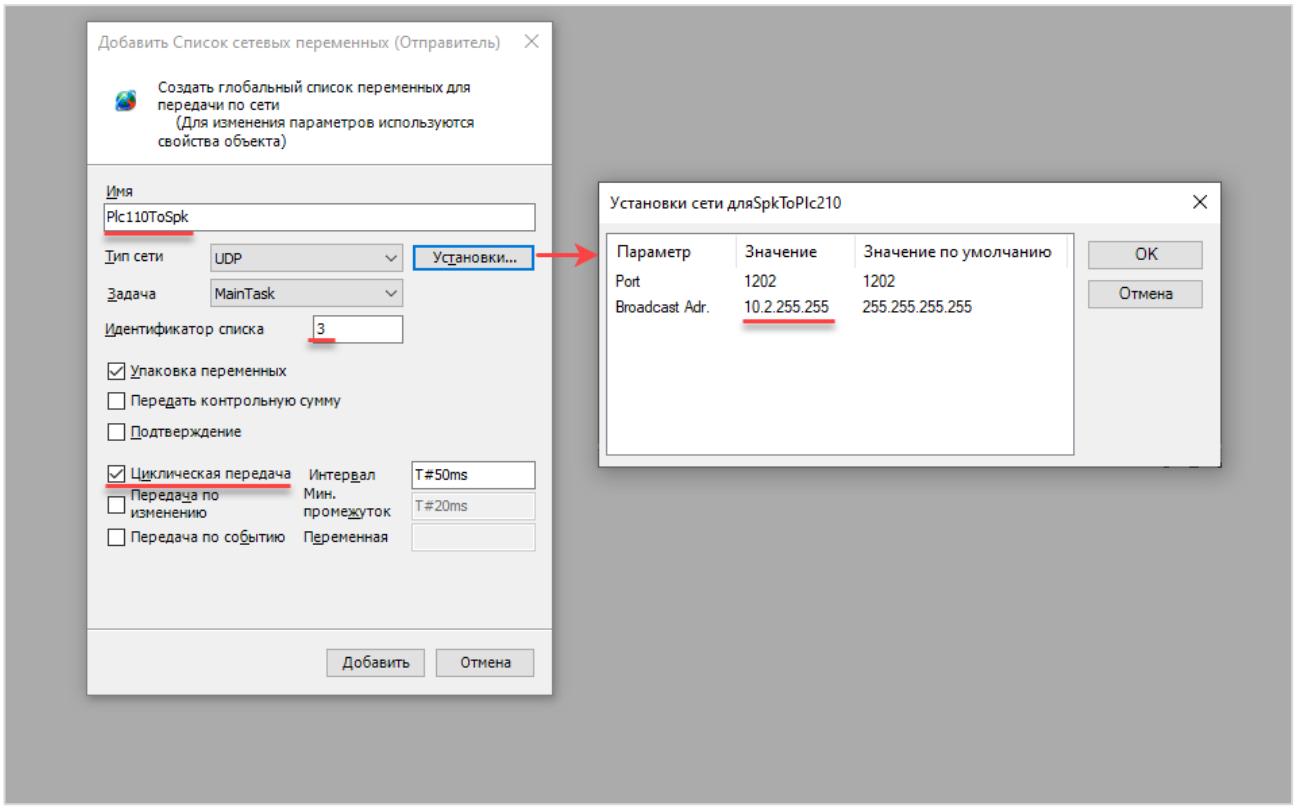

Рисунок 2.5.7 – Настройки «имитационного» списка отправляемых сетевых переменных

В созданном списке объявить переменную **wPlc110ToSpk** типа **WORD** (по аналогии с рисунком 2.5.4);

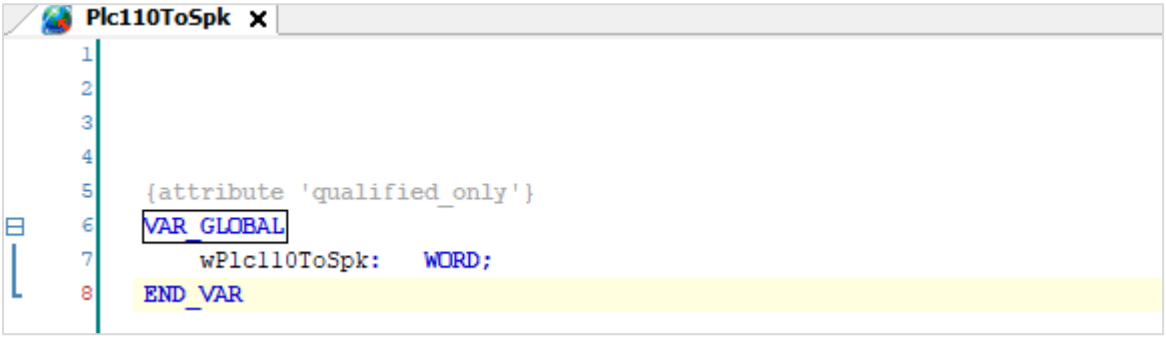

Рисунок 2.5.8 – Объявление отправляемых сетевых переменных в «имитационном» списке

9. Нажать **ПКМ** на список сетевых переменных **Plc110ToSpk**, выбрать пункт **Свойства**, во вкладке **Связь с файлом** выбрать режим **Экспорт перед компиляцией** и указать путь, по которому будет сохранен файл экспорта (имя файла может быть произвольным):

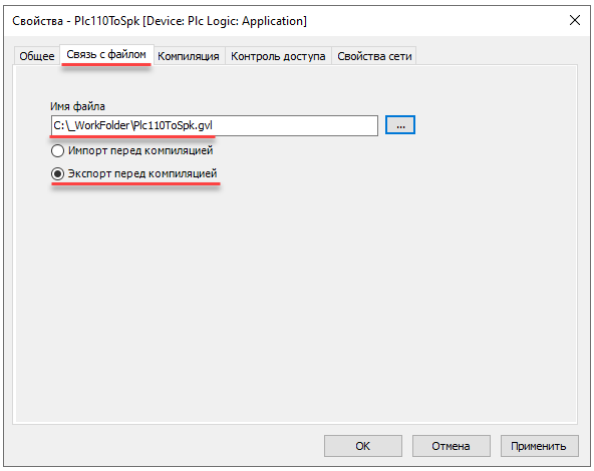

Рисунок 2.5.9 – Настройки экспорт списка сетевых переменных

10. Выполнить команду **Компиляция – Генерировать код**. После этого по указанному пути будет сформирован файл экспорта.

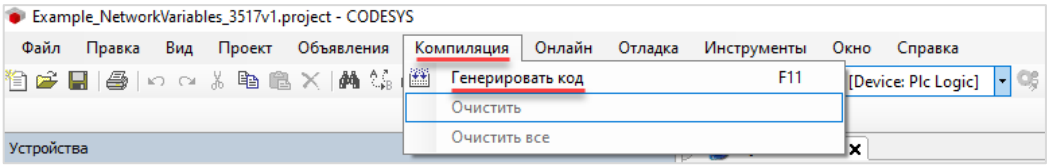

Рисунок 2.5.10 – Экспорт списка сетевых переменных

11. Удалить список сетевых переменных **Plc110ToSpk** из проекта **CODESYS V3.5**.
12. Добавить компонент Список сетевых переменных (получатель) с названием **Plc110ToSpk** и импортировать файл **Plc110ToSpk.gvl**, созданный в пп. 10:

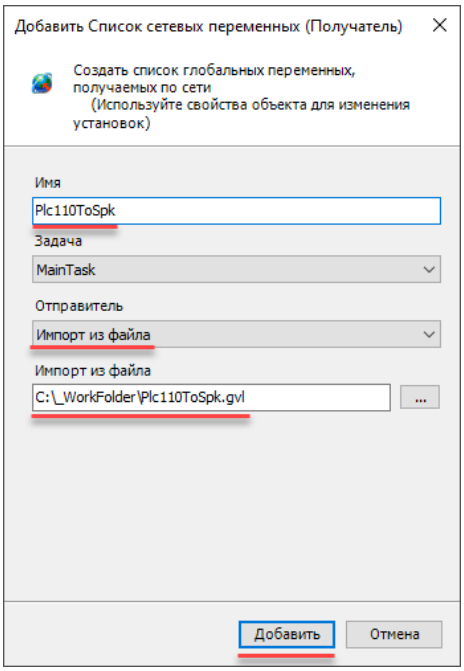

Рисунок 2.5.11 – Импорт списка сетевых переменных

13. Загрузить проекты в оба контроллера и запустить их. Убедиться, что оба контроллера подключены к одной локальной сети.
14. В проекте **CODESYS V3.5** в списке **SpkToPlc210** изменить значение переменной **wSpkToPlc210**. Проверить, что оно изменилось в **CoDeSys V2.3**.

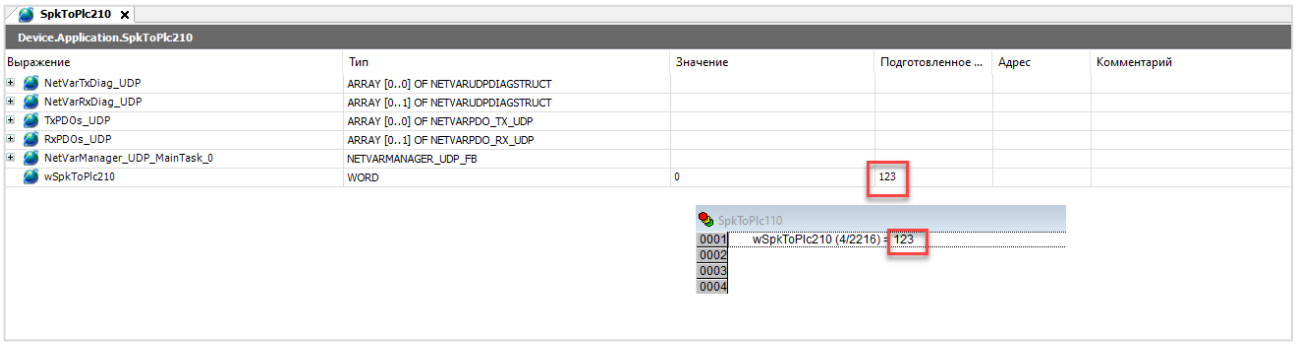

Рисунок 2.5.12 – Передача сетевых переменных из CODESYS V3.5 в CoDeSys V2.3

15. В проекте **CoDeSys V2.3** в списке **Plc110ToSpk** изменить значение переменной **wPlc110ToSpk**. Проверить, что оно изменилось в **CODESYS V3.5**.

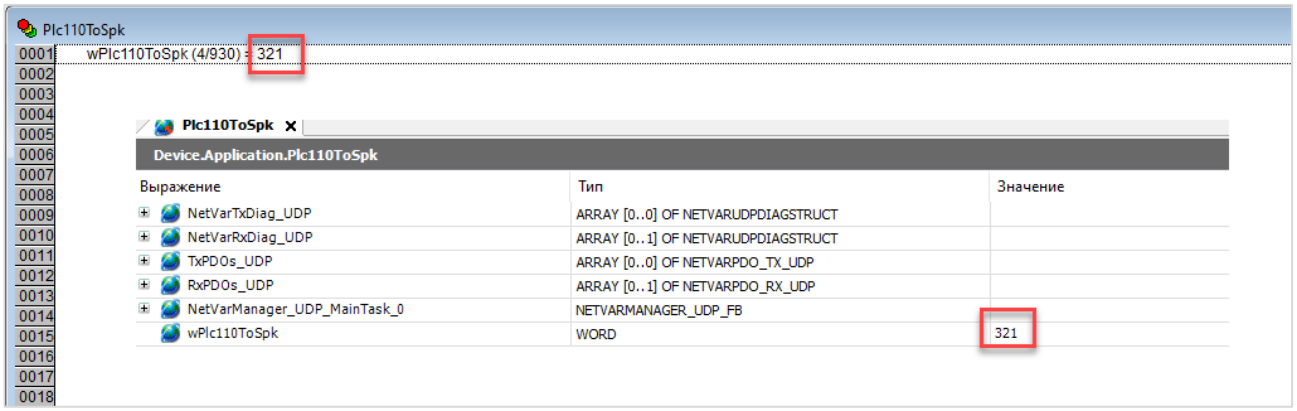

Рисунок 2.5.13 – Передача сетевых переменных из CoDeSys V2.3 в CODESYS V3.5

2.6 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 не должен блокироваться на уровне промежуточного сетевого оборудования.
4. Границы передаваемых массивов должны быть определены только через литералы или константы (но не выражения).
5. Максимальный размер сетевой переменной – 255 байт.
6. Число сетевых переменных в проекте не ограничено.
7. Для обращения к сетевой переменной в коде программы требуется указывать имя списка сетевых переменных. Например, для обращения к переменной **wSpkToPlc210** из списка **SpkToPlc210** требуется указать такое имя: **SpkToPlc210.wSpkToPlc210**.

2.7 Диагностика

Диагностика обмена сетевыми переменными производится с помощью специальных переменных, отображаемых в списках сетевых переменных при онлайн-подключении к контроллеру. В документации CODESYS описание этих переменных отсутствует. Ниже приведены несколько практических советов по их использованию.

Для диагностики на стороне отправителя необходимо в свойствах списка сетевых переменных на вкладке **Свойства сети** установить галочку **Подтверждение** для ожидания подтверждений от получателя.

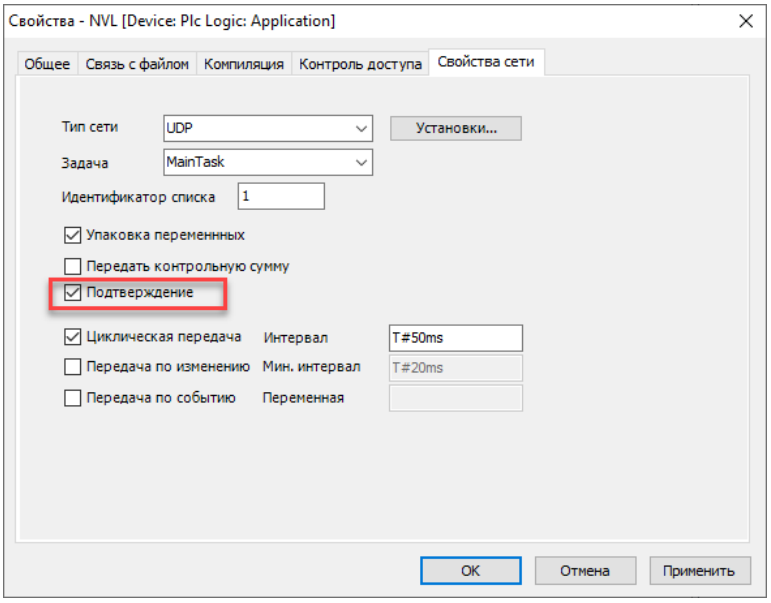

Рисунок 2.7.1 – Установка ожидания подтверждения получения сетевых переменных

В случае обрыва связи в структуре **NetVarTxDiag_UDP** поле **sLastError** примет значение **NetVarUDPError_NOACKNOWLEDGEMENT**, а поле **nErrorCount** будет постоянно увеличиваться.

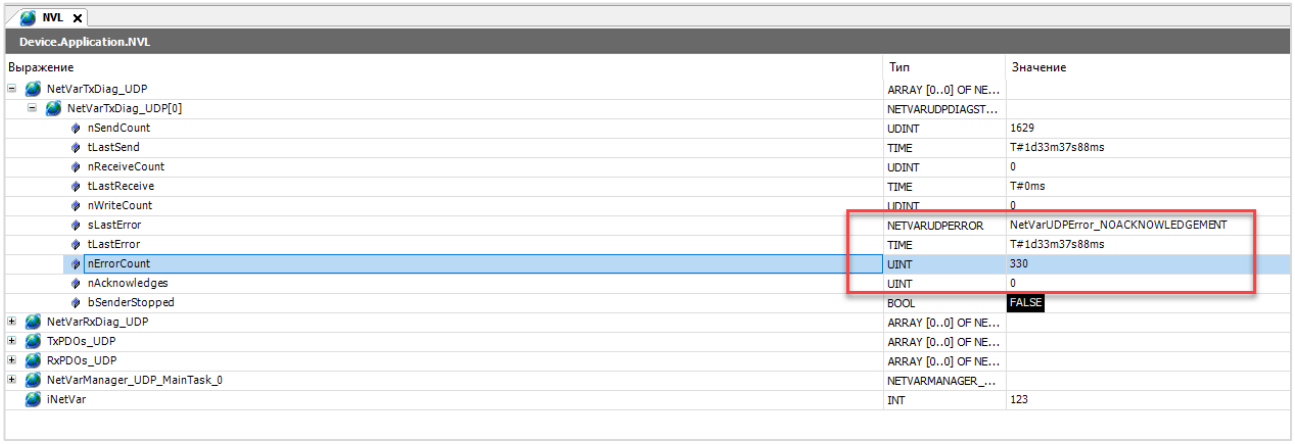

Рисунок 2.7.2 – Диагностика обрыва связи на стороне отправителя

Диагностика связи на стороне получателя: в случае обрыва связи в структуре **NetVarRxDiag_UDP** значение полей **nReceiveCount** и **tLastReceive** «застынут» и перестанут изменяться.

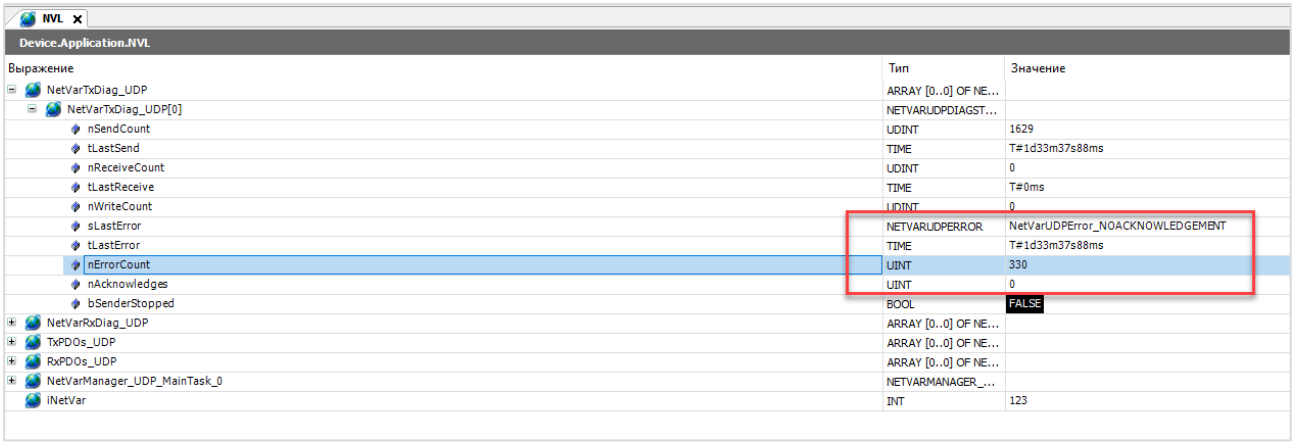

Рисунок 2.7.3 – Диагностика обрыва связи на стороне получателя

В версии **CODESYS V3.5 SP17** значения переменных диагностики в списке сетевых переменных могут не отображаться:

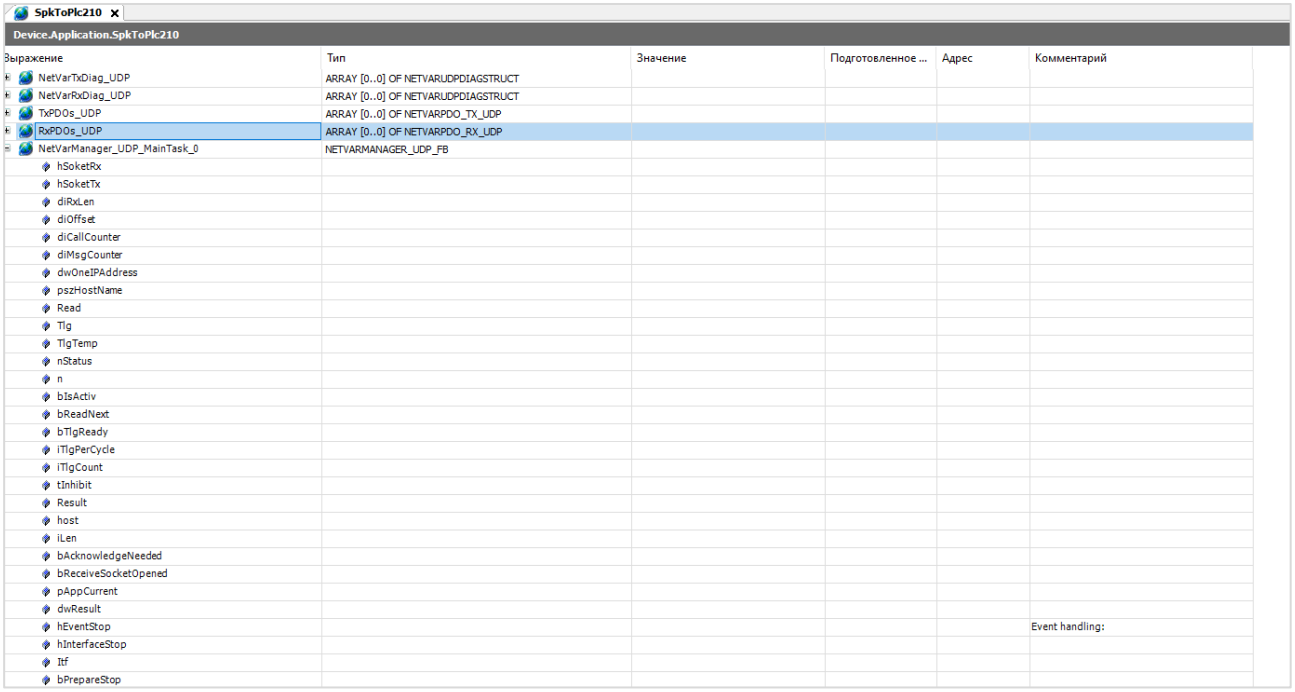

Рисунок 2.7.4 – Проблема с отображением значений переменных диагностики в CODESYS V3.5 SP17

В этом случае следует добавить их в список просмотра (**Вид – Просмотр – Watch**):

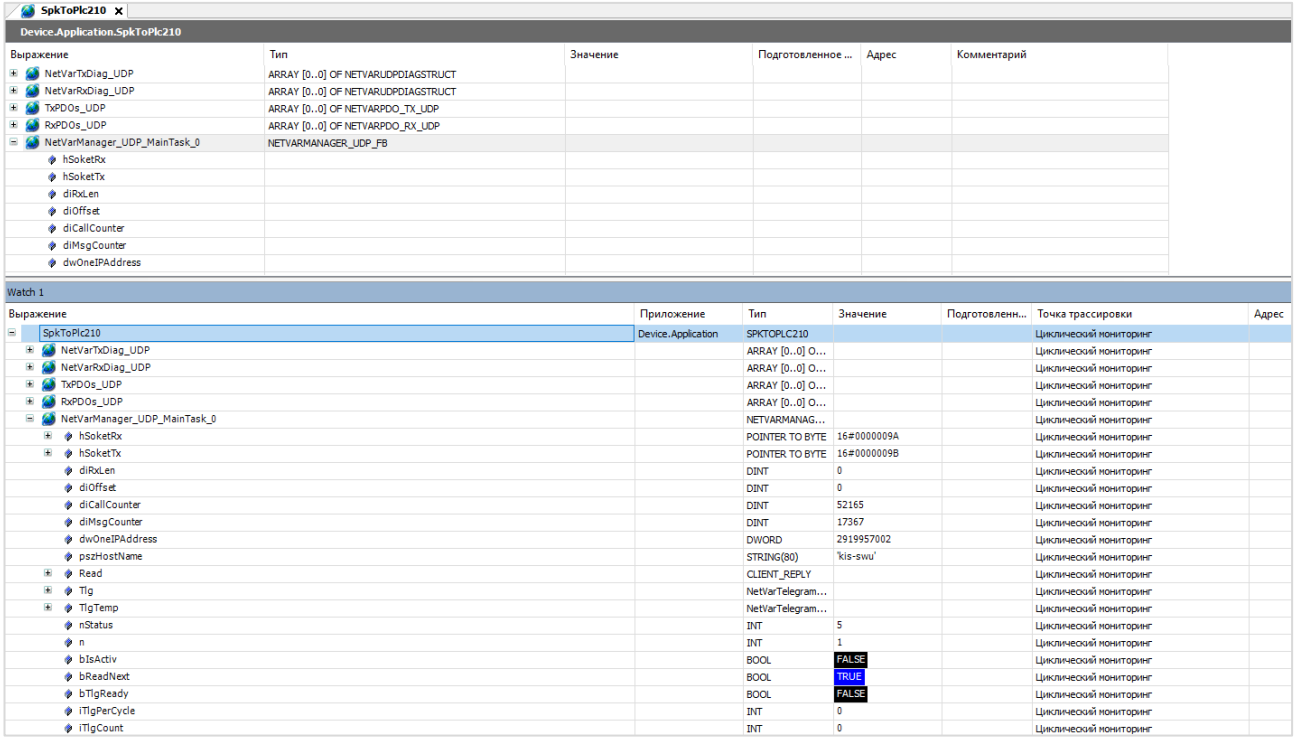

Рисунок 2.7.5 – Отображения значений переменных диагностики в окне просмотра

Переменные диагностики можно использовать в коде программы (при этом компоненты диагностики не будут предлагаться для автодополнения – требуется ввести их имена вручную):

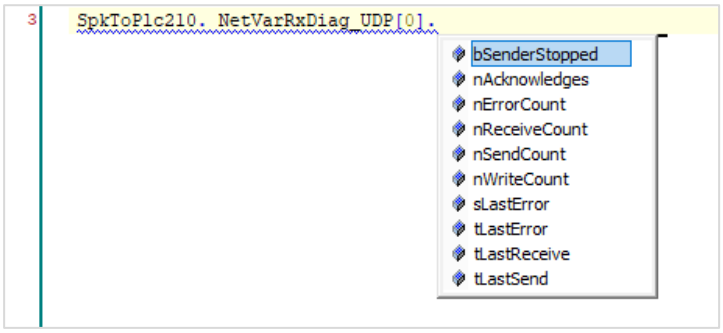

Рисунок 2.7.6 – Доступ к переменным диагностики в коде программы

.

3 Менеджер источников данных

3.1 Основные сведения о менеджере источников данных

Менеджер источников данных позволяет организовать обмен между несколько контроллерами, программируемыми в **CODESYS V3.5.** В рамках данного механизма опроса контроллер может выполнять роль источника данных (удаленного устройства) или менеджера данных (локального устройства) – но не может совмещать эти роли.

В приложении удаленных устройств (серверов) не требуется каких-либо дополнительных настроек для использования этого механизма обмена.

В приложении локальных устройств (клиентов) должен быть добавлен и настроен компонент **Менеджер источников данных**.

Контроллеры, которые участвуют в обмене, должны находиться в одной локальной сети и иметь разные сетевые имена (hostname) – в том числе, на этапе настройки обмена. Настроить обмен для разных ПЛК «по отдельности» не получится, потому что для установки связи требуется выполнить операцию сканирования сети – соответственно, контроллеры должны обнаруживаться при сканировании, а для этого в сети должны быть разрешены широковещательные UDP-запросы (UDP broadcast). Крайне рекомендуется проконтролировать, что ПК, на котором выполняется сканирование сети, был подключен только к одной локальной сети (имел только один активный сетевой интерфейс) – иначе на этапе конфигурирования обмена адреса локальных устройств могут быть определены некорректно, что приведет к невозможности подключения к ним менеджера источника данных.

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

В п. 3.2 приведен пример настройки обмена с использованием менеджера источника данных и описание этого компонента.

В п. 3.3 описан частный случай настройки обмена с удаленным устройством с использованием переменной с адресом устройства.

В п. 3.4 приведена информация по использованию источников данных в визуализации.

В п. 3.5 описан процесс диагностики обмена с удаленным устройством.

3.2 Пример настройки обмена

В качестве примера будет рассмотрен обмен между контроллерами **СПК1xx [M01]** и **ПЛК210**. ПЛК210 будет использоваться в роли удаленного устройства, СПК1xx [M01] – локального устройства. Пример доступен для скачивания: Example_Datasources_3517v1.projectarchive

Для настройки обмена с помощью менеджера источника данных следует:

1. Объявить в проекте удаленного устройства переменные, к которым требуется организовать доступ со стороны локального устройства. В рамках примера используются переменные типа **WORD**, **REAL** и **STRING**:

```
PLC_PRG
2  VAR
3      wVar:   WORD;
4      rVar:   REAL;
5      sVar:   STRING;
6  END_VAR
7
```

Рисунок 3.2.1 – Объявление переменных удаленного устройства

2. Подключиться к ПЛК210 (узел **Device** – вкладка **Установки соединения**) и загрузить в него проект.
3. Нажать **ПКМ** на имя проекта, использовать команду **Добавить устройство** и выбрать модель контроллера **СПК1xx [M01]**:

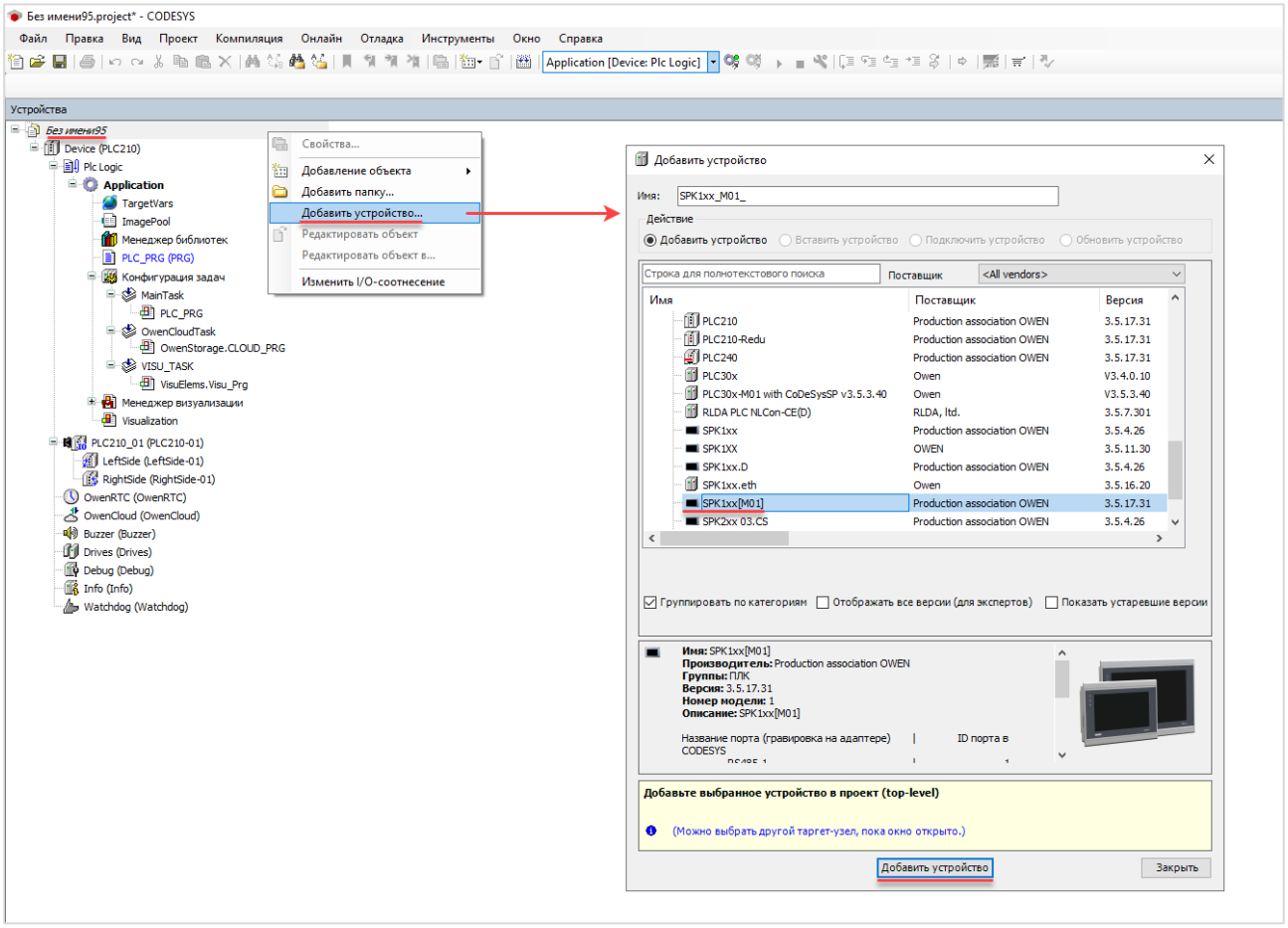

Рисунок 3.2.2 – Добавление в проект CODESYS локального устройства

4. Нажать **ПКМ** на узел **Application** и добавить программу **SPK_PRG**.

ВНИМАНИЕ

Менеджер источников данных импортирует в проект объекты удаленного устройства. Так как в примере программа в ПЛК210, в которой объявлены переменные, называется **PLC_PRG**, то в проекте СПК1xx [M01] не должно быть программы с таким же названием (иначе возникнут ошибки компиляции).

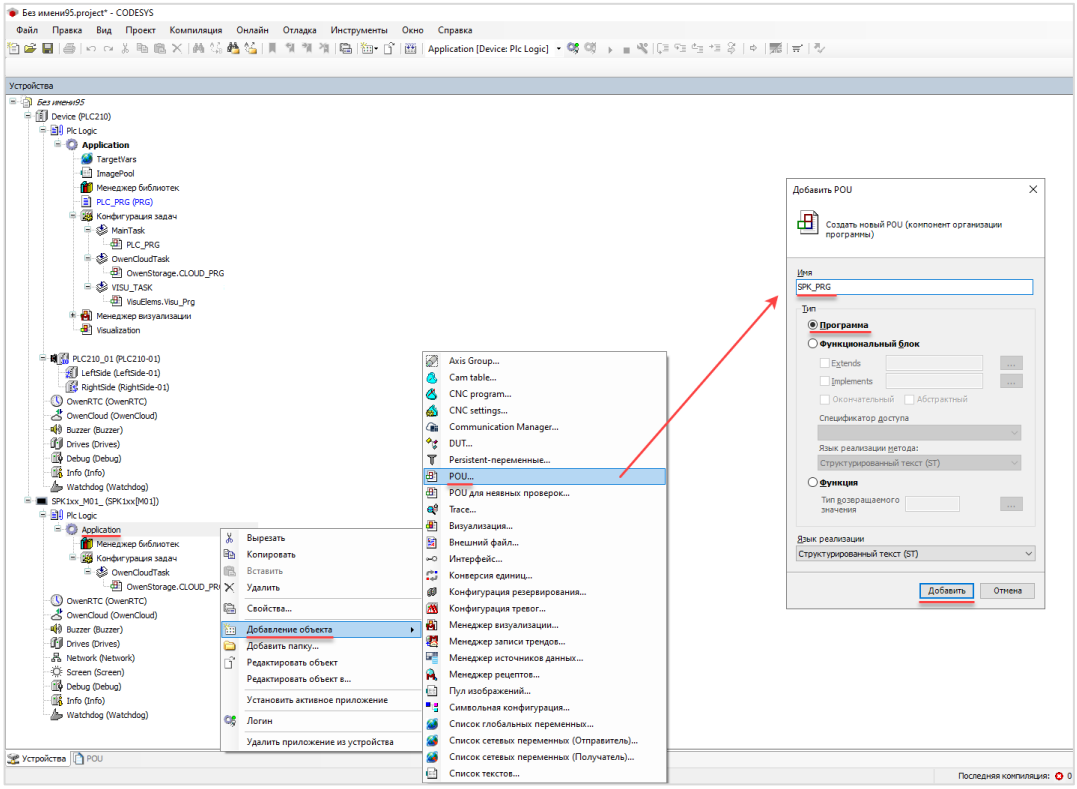

Рисунок 3.2.3 – Добавление программы SPK_PRG

5. В программе **SPK_PRG** объявить два набора переменных – один из них будет соответствовать значениям переменных, считываемым из удаленного устройства, второй – использоваться для записи в удаленное устройство новых значений по команде, представленной переменной **xWriteData**.Код программы будет добавлен в пп. 15 после настройки менеджера источников данных.

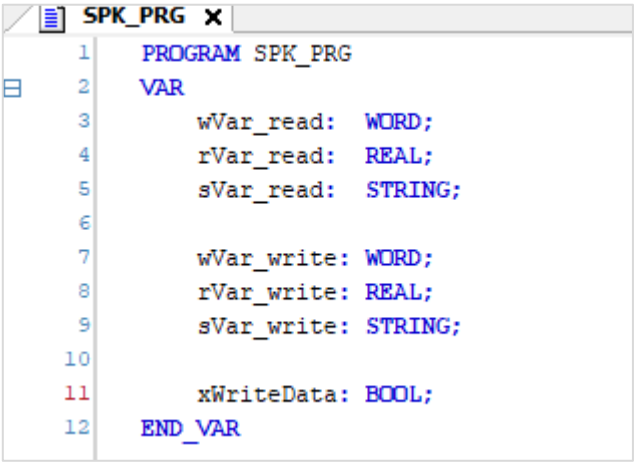

Рисунок 3.2.4 – Объявление переменных программы SPK_PRG

6. Нажать **ПКМ** на компонент **Конфигурация задач** и добавить задачу с названием **MainTask** и настройками по умолчанию.

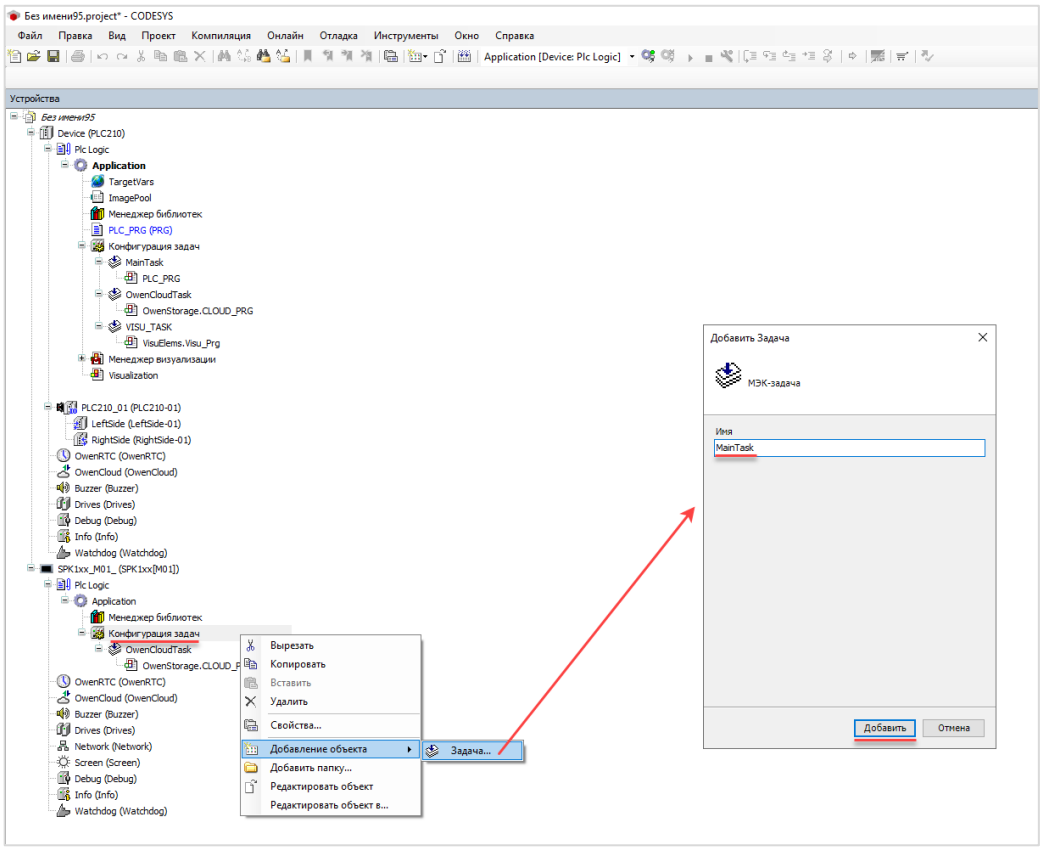

Рисунок 3.2.5 – Добавление задачи MainTask

7. В настройках созданной задачи добавить вызов программы **SPK_PRG**.

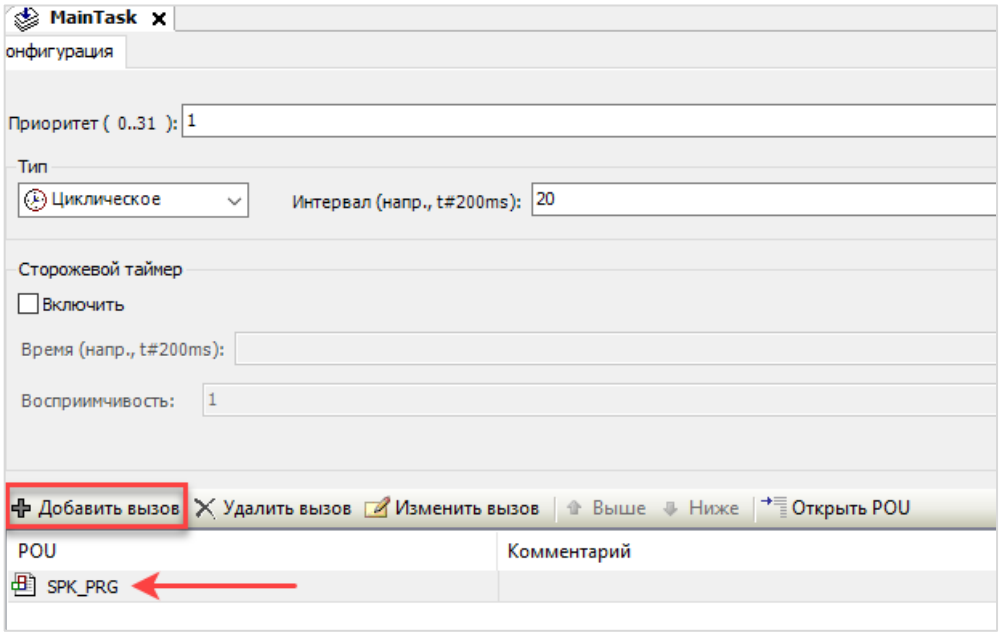

Рисунок 3.2.6 – Привязка программы SPK_PRG к задаче MainTask

8. Нажать **ПКМ** на узел **Application** и добавить в проект визуализацию (без нее не получится скомпилировать проект для СПК).

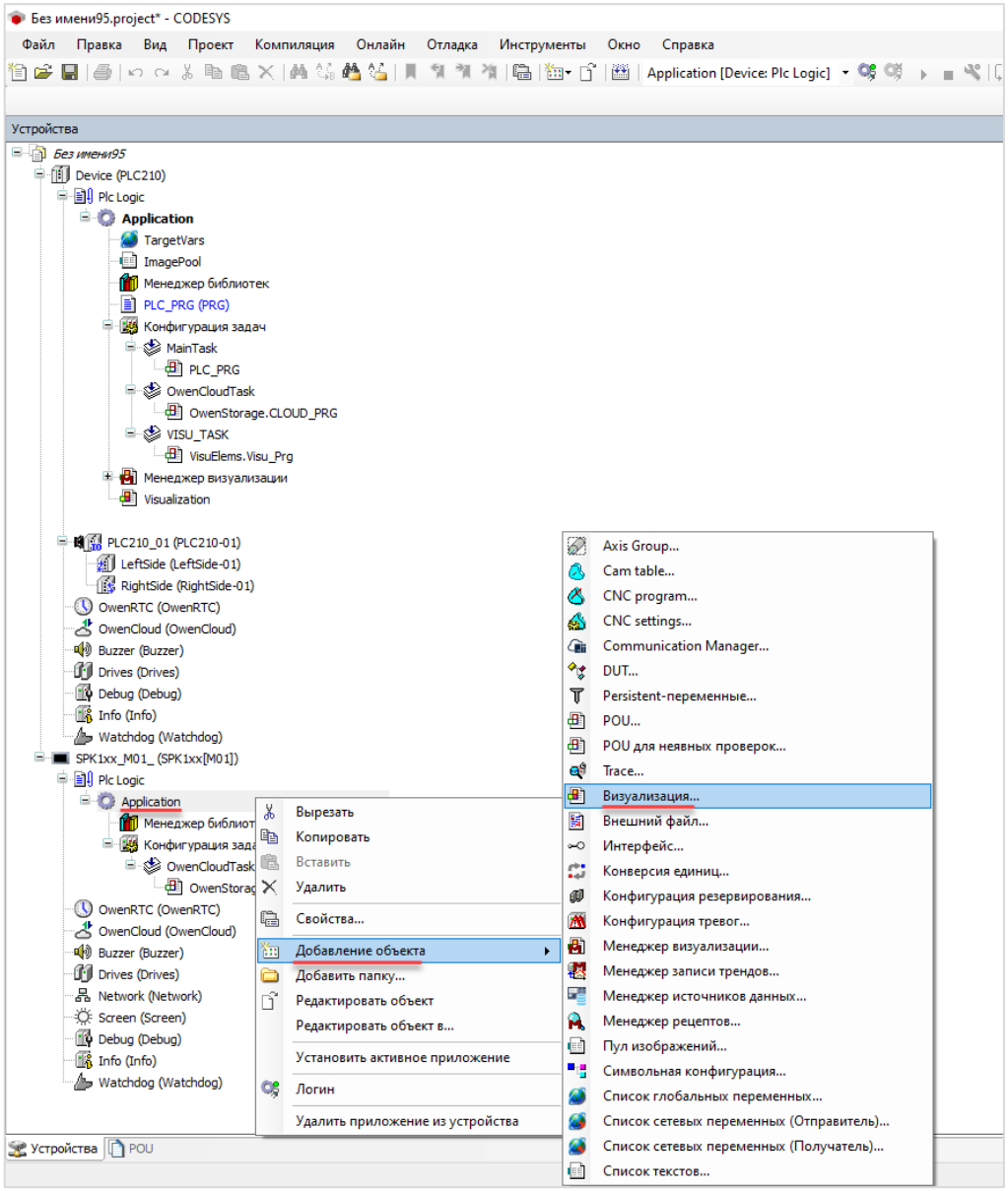

Рисунок 3.2.7 – Добавление в проект визуализации

9. Нажать **ПКМ** на узел **Application** и добавить в проект **Менеджер источников данных**.

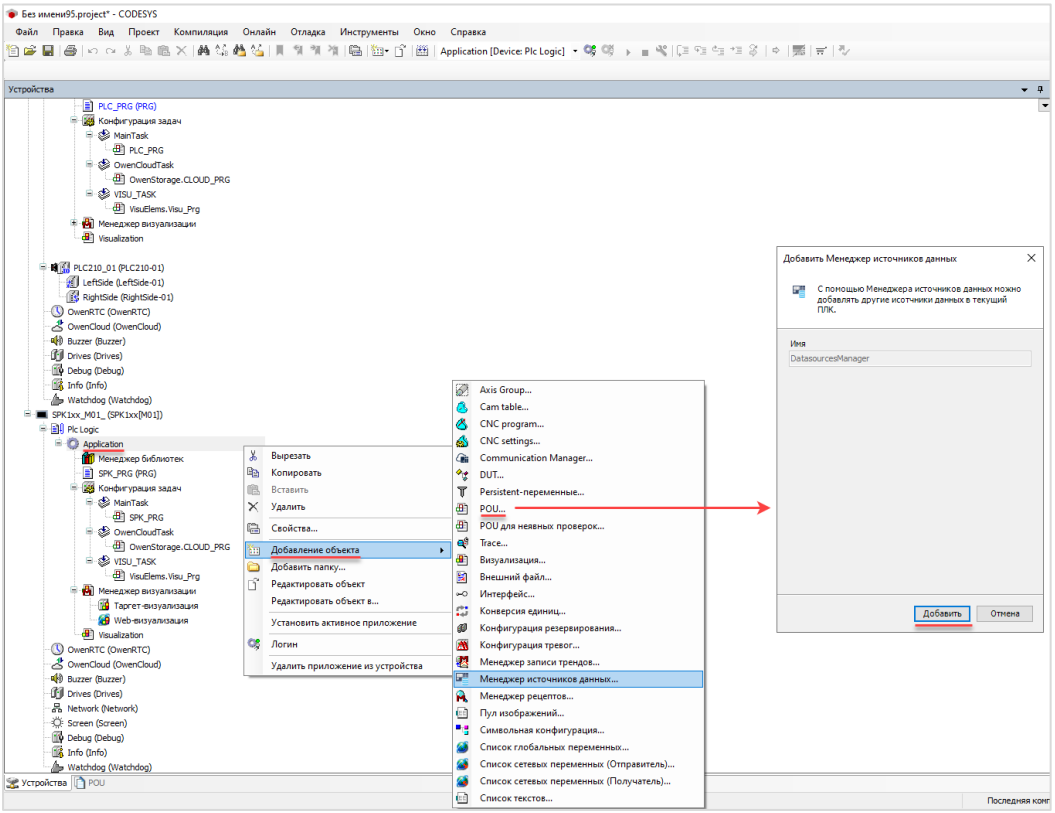

Рисунок 3.2.8 – Добавление менеджера источника данных

10. Нажать **ПКМ** на компонент **Менеджер источников данных** и добавить источник данных с названием **Datasource**. В рамках примера используется один источник данных, в случае необходимости – можно добавить несколько источников данных (каждый источник данных соответствует одному опрашиваемому менеджером контроллеру).

ВНИМАНИЕ

Контроллеры ОВЕН поддерживают только один тип источника данных – **CODESYS ApplicationV3**.

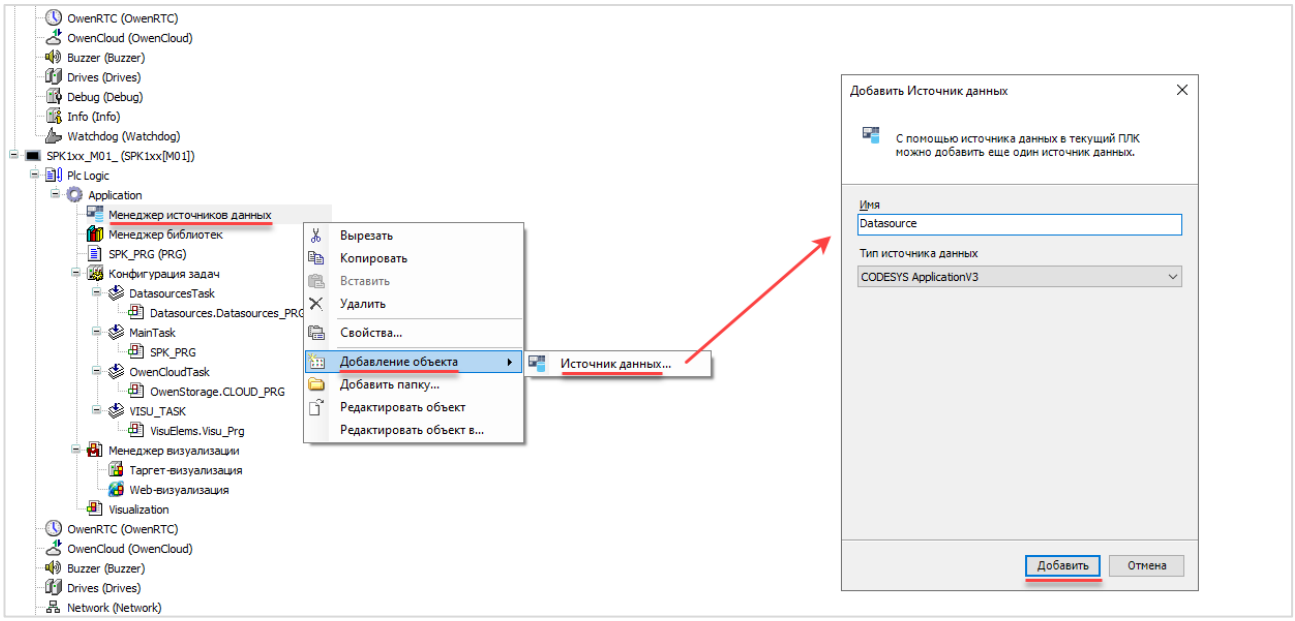

Рисунок 3.2.9 – Добавление источника данных

11. В появившемся окне указать параметры удаленного устройства:

Выберите текущий проект – если удаленное устройство добавлено в этом же проекте CODESYS (как в примере), то следует указать вариант **Текущий проект** и выбрать в окне приложение нужного устройства (в рамках примера – приложение ПЛК210). Если же удаленное устройство создано в отдельном проекте CODESYS, то следует выбрать вариант **Другой проект** и указать путь к файлу проекта.

Целевое устройство – определение сетевых настроек удаленного устройства. Возможные варианты:

- **Автоматическая конфигурация** – конфигурация будет считана автоматически, если удаленное устройство добавлено в том же проекте (как в примере) и обнаружено при сканировании сети. Предварительно в это устройство должен быть загружен проект (см. пп. 2). Этот способ является наиболее простым и рекомендуется использовать именно его;
- **Конфигурация вручную** – конфигурация задается пользователем с помощью специальной переменной (см. п. 3.3) или с помощью сканирования сети по заданным параметрам (см. информацию в онлайн-справке CODESYS).

Конфигурация логина – если в удаленном устройстве настроено управление пользователями визуализации, то требуется указать логин и пароль, которые будут использоваться для подключения;

Размер буфера связи по умолчанию – недокументированная настройка, значение которой не рекомендуется изменять.

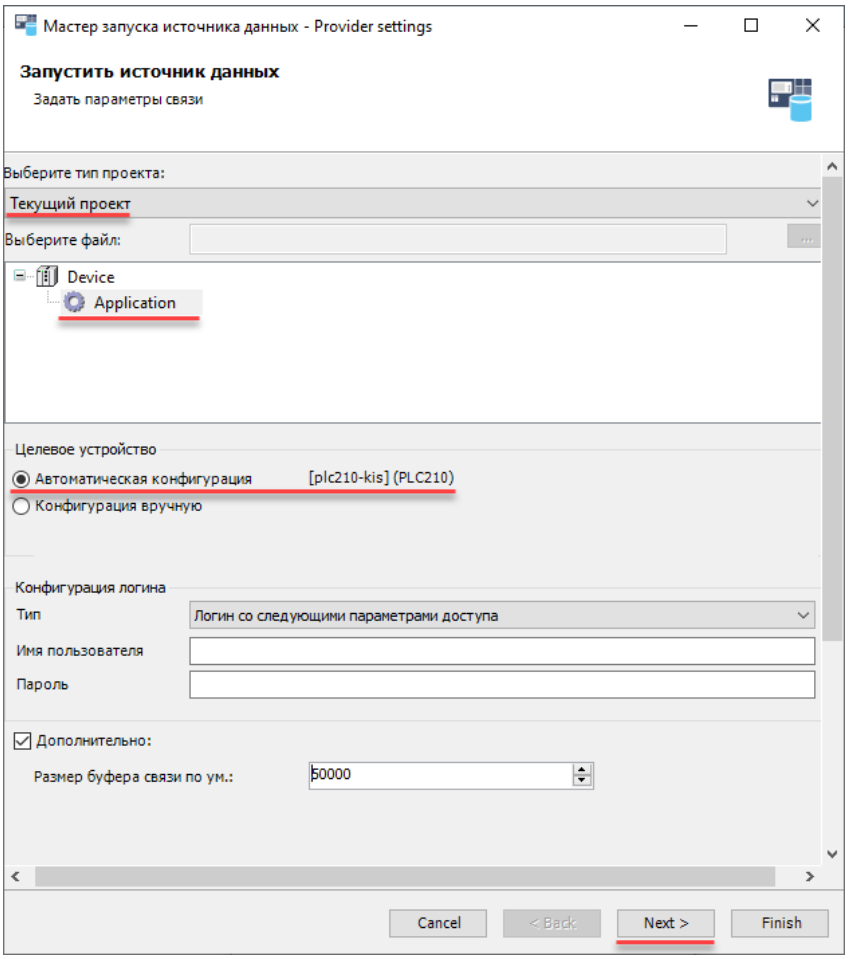

Рисунок 3.2.10 – Настройки подключения к источнику данных

После ввода настроек следует нажать **Next** и выбрать нужные переменные источника данных:

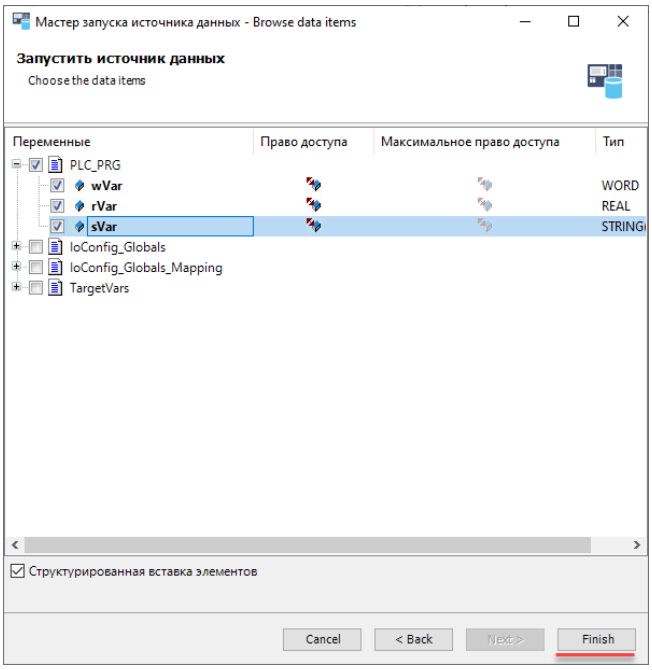

Рисунок 3.2.11 – Выбор переменных удаленного устройства

Для завершения настройки следует нажать **Finish**.

12. У добавленного в проект источника данных доступны следующие вкладки:

Переменные – на этой вкладке отображаются переменные, импортированные из удаленного устройства. Столбец **Создать или соотнести** позволяет выбрать, как будет производиться импорт – с помощью автоматического создания в проекте локального устройства нужных переменных ( ) или ручного соотнесения переменных удаленного и локального устройства со строгим ( ) или нестрогим ( ) совпадением типов. Подробнее см. в онлайн-справке CODESYS. Рекомендуется использовать автоматический импорт.

В столбце **Обновлять всегда** выбирается режим обновления переменных. Если переменные используются в визуализации (см. также п. 3.4), то рекомендуется не устанавливать галочку – в этом случае обновление переменных будет происходить автоматически при изменении их значений. Если переменные используются только в коде программы (как в примере) – то следует установить галочку для циклического обновления данных. Также возможно обновление данных по команде в коде программы – см. статью в онлайн-справке CODESYS.

Команда **Обновить переменные** позволять выполнить повторный импорт переменных из удаленного устройства (например, если его переменные изменились).

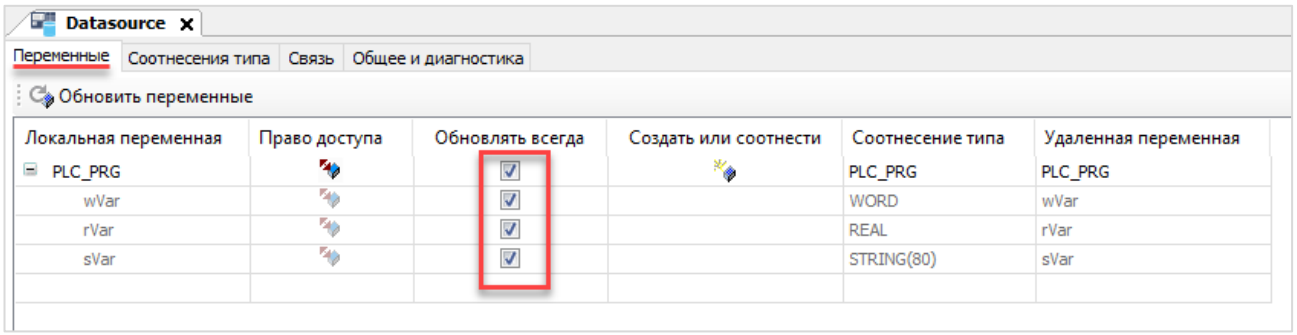

Рисунок 3.2.12 – Настройки источника данных, вкладка Переменные

Соотнесения типа – на этой вкладке отображаются типы импортированных данных. Пользователь может указать соответствие между типами объектов удаленного и локального устройства. Это полезно, например, если в удаленном и локальном устройстве используются одни и те же структуры/перечисления – в этом случае автоматическое создание ( ) невозможно, потому что в результате в проекте локального устройства будут присутствовать два объекта с совпадающими названиями, что приведет к ошибкам компиляции. Вместо этого надо соотнести ( ) структуру удаленного устройства с аналогичной структурой локального устройства.
Подробнее см. в онлайн-справке CODESYS.

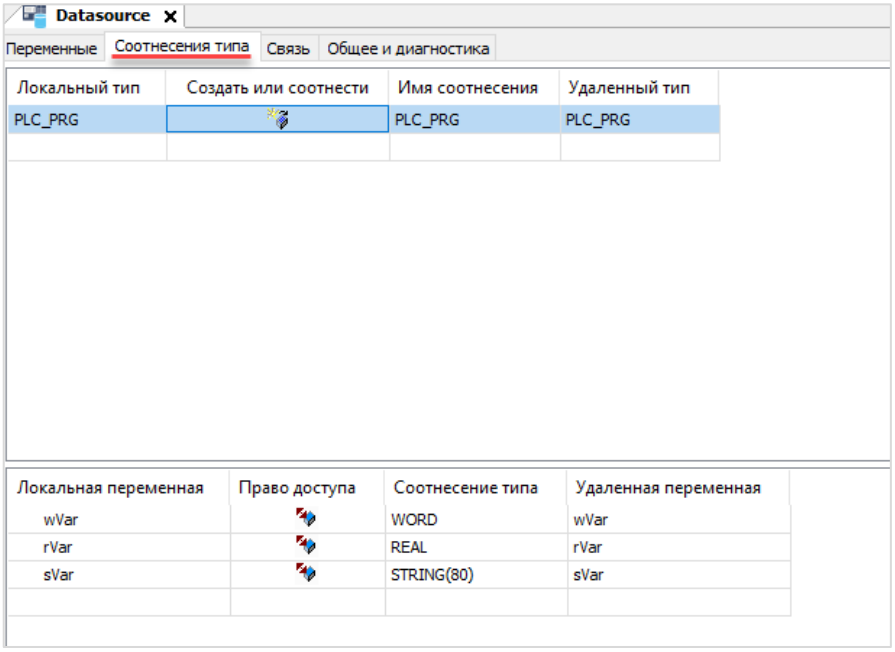

Рисунок 3.2.13 – Настройки источника данных, вкладка Соотнесение типа

Связь – на этой вкладке доступны коммуникационные настройки (их список аналогичен рисунку 3.2.10).

Общее и диагностика – на этой вкладке можно настроить период обновления данных. При онлайн-подключении к контроллеру на этой вкладке отображается статус соединения и код ошибки. См. описание переменных диагностики в п. 3.5.

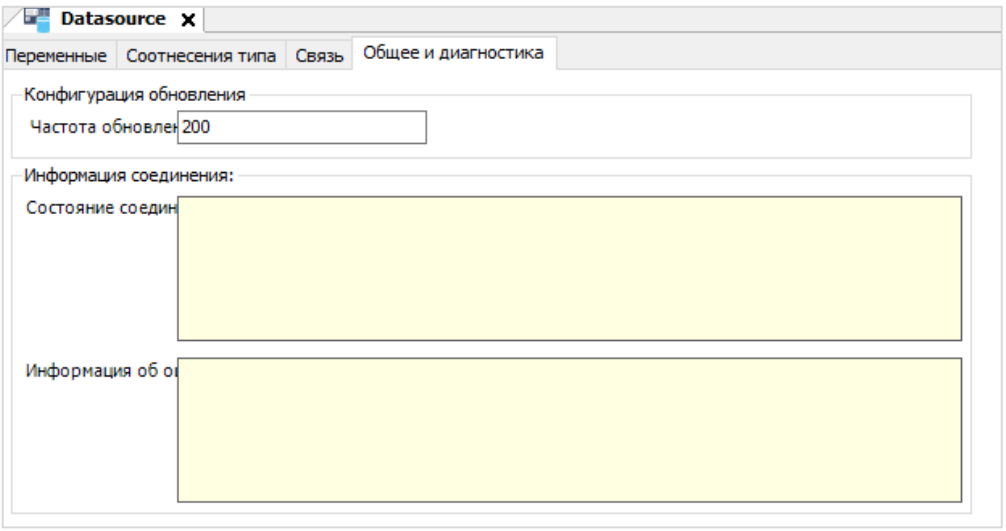

Рисунок 3.2.14 – Настройки источника данных, вкладка Общее и диагностика

13. После добавления и настройки источника данных в проект автоматически будут добавлены:
    - библиотеки **DatasourceAppV3** и **Datasources** (например, они используются для диагностики обмена и настройки подключения из кода программы);
    - задача **DatasourcesTask**;
    - папка **DataSources_Objects** с импортированными объектами и переменными удаленного устройства.

Настоятельно не рекомендуется редактировать эти компоненты.

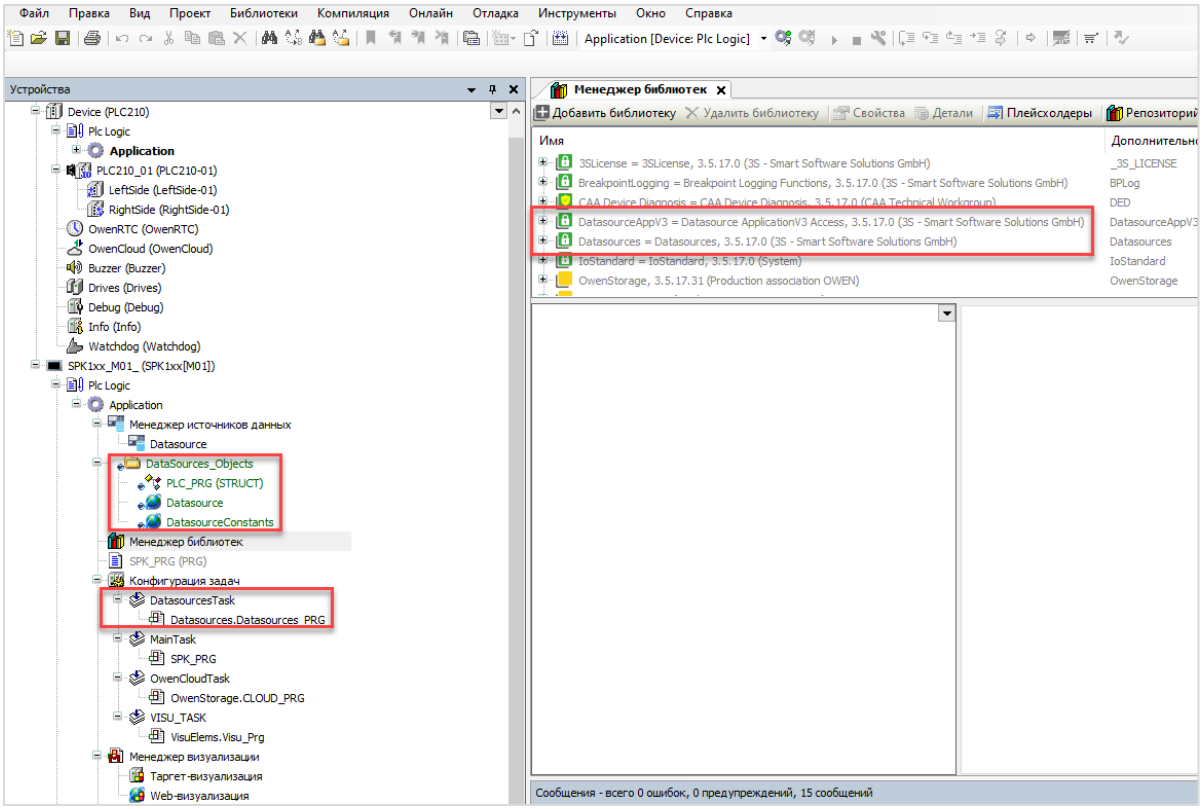

Рисунок 3.2.15 – Автоматические добавленные компоненты менеджера источника данных

14. Для доступа к переменным удаленного устройства следует в коде программе обратиться к объекту нужного источника данных (в примере он называется **Datasource** – см. пп. 10) и в выпадающем списке выбрать нужный объект и его переменные:

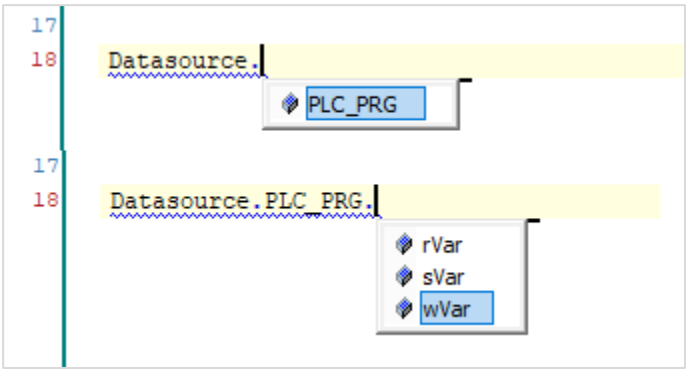

Рисунок 3.2.16 – Обращение к переменным удаленного устройства

15. В рамках примера в программу **SPK_PRG** будет добавлен следующий код:

```
1   // чтение значений из источника данных
2   wVar_read := Datasource.PLC_PRG.wVar;
3   rVar_read := Datasource.PLC_PRG.rVar;
4   sVar_read := Datasource.PLC_PRG.sVar;
5
6   // запись значений в источник данных
7   IF xWriteData THEN
8
9       Datasource.PLC_PRG.wVar := wVar_write;
10      Datasource.PLC_PRG.rVar := rVar_write;
11      Datasource.PLC_PRG.sVar := sVar_write;
12
13      xWriteData := FALSE;
14
15  END_IF
16
```

Рисунок 3.2.17– Код программы SPK_PRG

16. Осталось загрузить проекты в оба устройства и запустить их. Чтобы произвести сканирование сети для конкретного устройства – следует сначала выбрать его приложение с помощью выпадающего списка на панели инструментов:

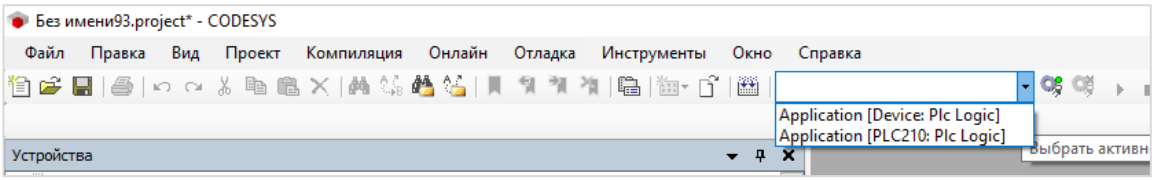

Рисунок 3.2.18 – Выбор активного приложение проекта

17. В случае успешной установки связи (это может занять некоторое время, которое прямо пропорционально числу опрашиваемых переменных) рядом с узлом источника данных отобразится соответствующая пиктограмма:

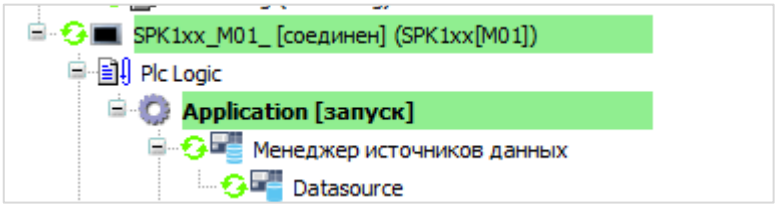

Рисунок 3.2.19 – Пиктограмма успешной установки связи с удаленным устройством

18. В устройстве **ПЛК210** следует изменить значения переменных и проверить, что в устройстве **СПК1xx [M01]** изменились значения переменных с постфиксом **_read**.

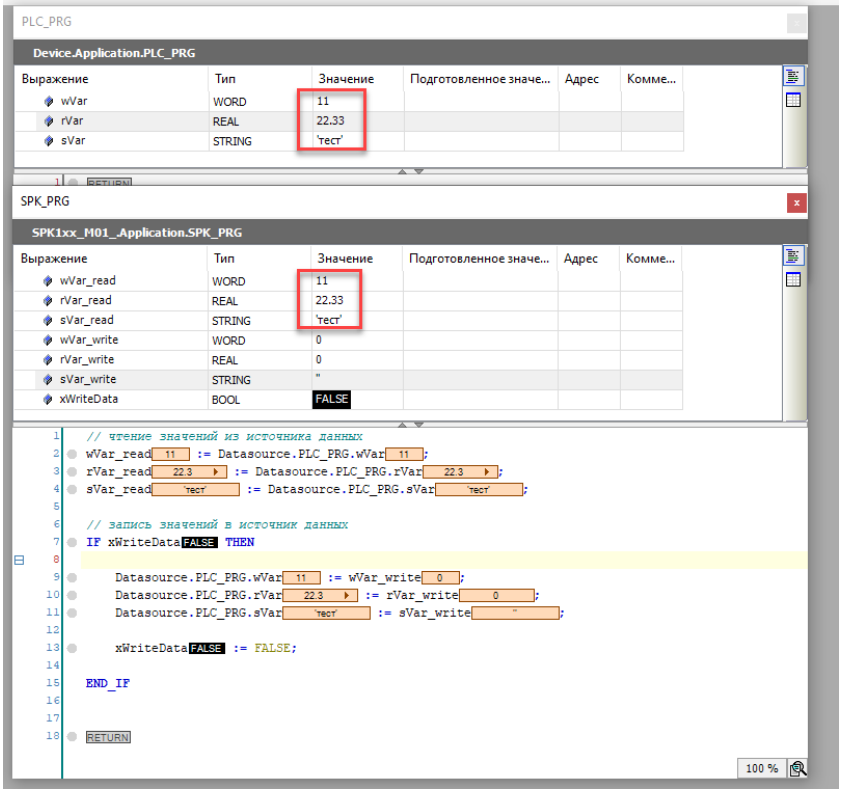

Рисунок 3.2.20 – Чтение значений из удаленного устройства

В устройстве **СПК1xx [M01]** следует изменить значения переменных с постфиксом **_write**, присвоить **TRUE** переменной **xWriteData** и проверить, что изменились значения переменных в устройстве **ПЛК210**.

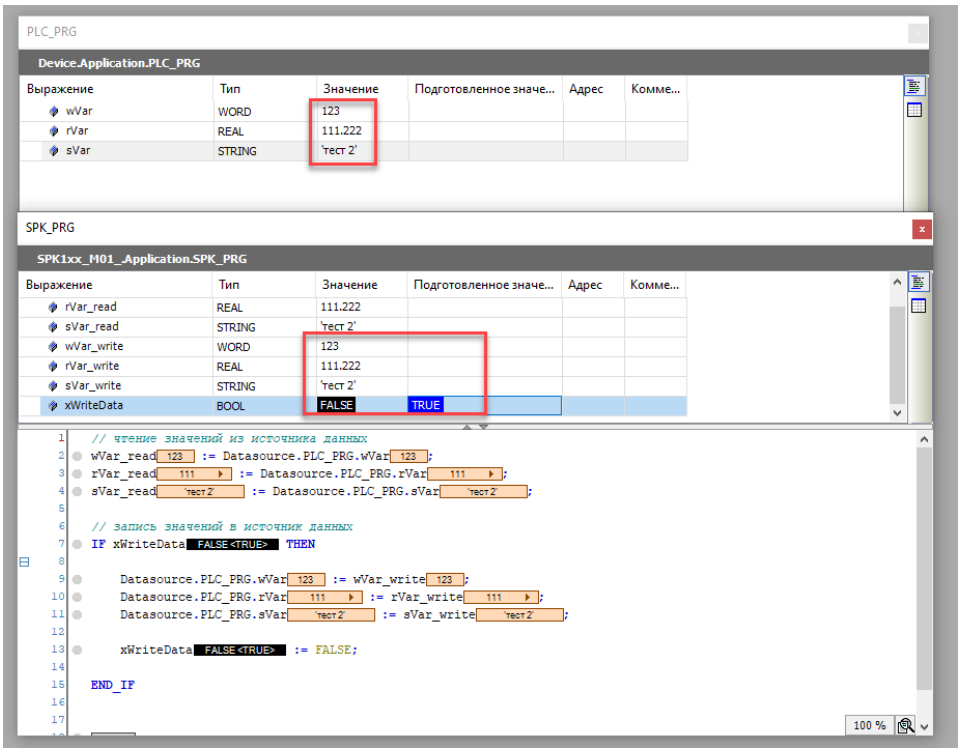

Рисунок 3.2.21 – Запись значений в удаленное устройство

3.3 Настройка обмена с использованием переменной

В п. 3.2 рассматривалась автоматическая конфигурация обмена с удаленным устройством. Альтернативным вариантом является настройка обмена через переменную типа **DatasourceAppV3.ConnectionSetup** (структуру). Эту переменную следует объявить в коде программы и инициализировать ее поля:

```
12
13      stDatasourceSetup:  DatasourceAppV3.ConnectionSetup;
14  END_VAR
```

```
1  // имя источника данных, отображаемое при сканировании сети
2  stDatasourceSetup.stNodeAddress := '0000.1902';
3  stDatasourceSetup.xDataValid := TRUE;
```

Рисунок 3.3.1 – Объявление экземпляра структуры DatasourceAppV3.ConnectionSetup и присвоение значений ее полям

В поле **stNodeAddress** следует записать адрес устройства, отображаемый на вкладке **Установки соединения**. После этого следует присвоить полю **xDataValid** значение **TRUE**.

Обратите внимание, что сканирование сети должно производиться именно с того ПК, с которого выполняется загрузка проекта в контроллер.

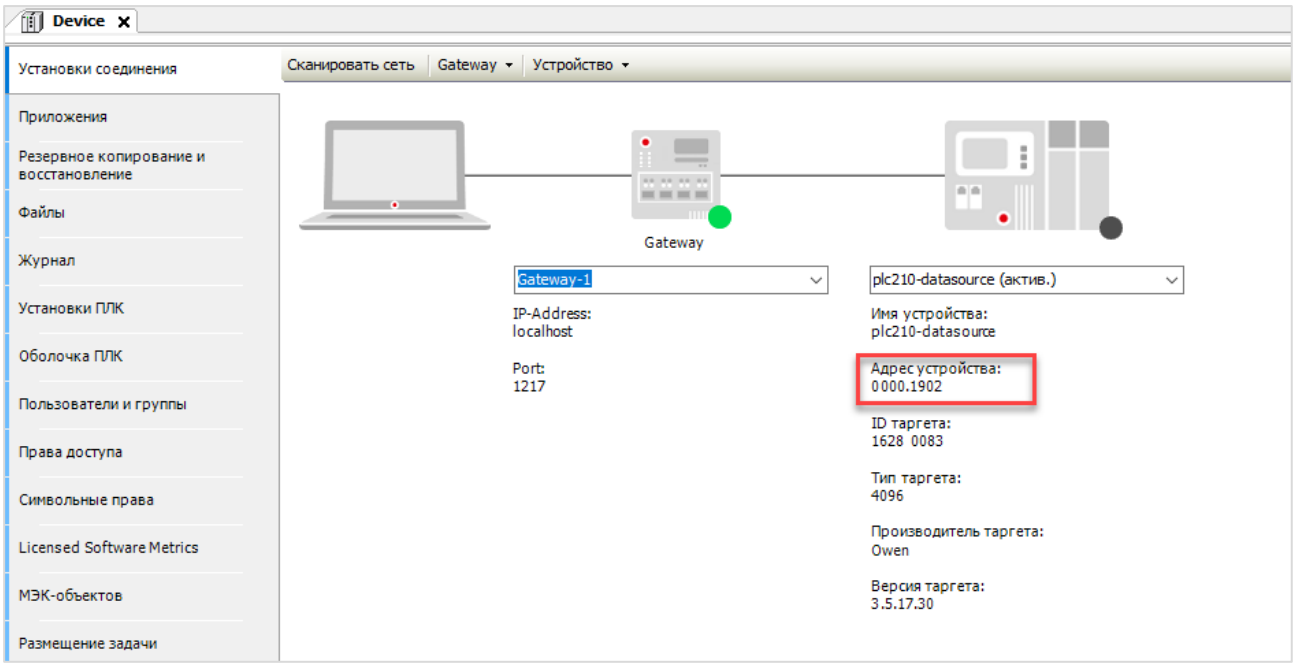

Рисунок 3.3.2 – Отображение адреса устройства

Теперь при настройке связи с удаленном устройством можно выбрать режим **Конфигурация вручную/Динамически из переменной устройства** и привязать объявленную переменную:

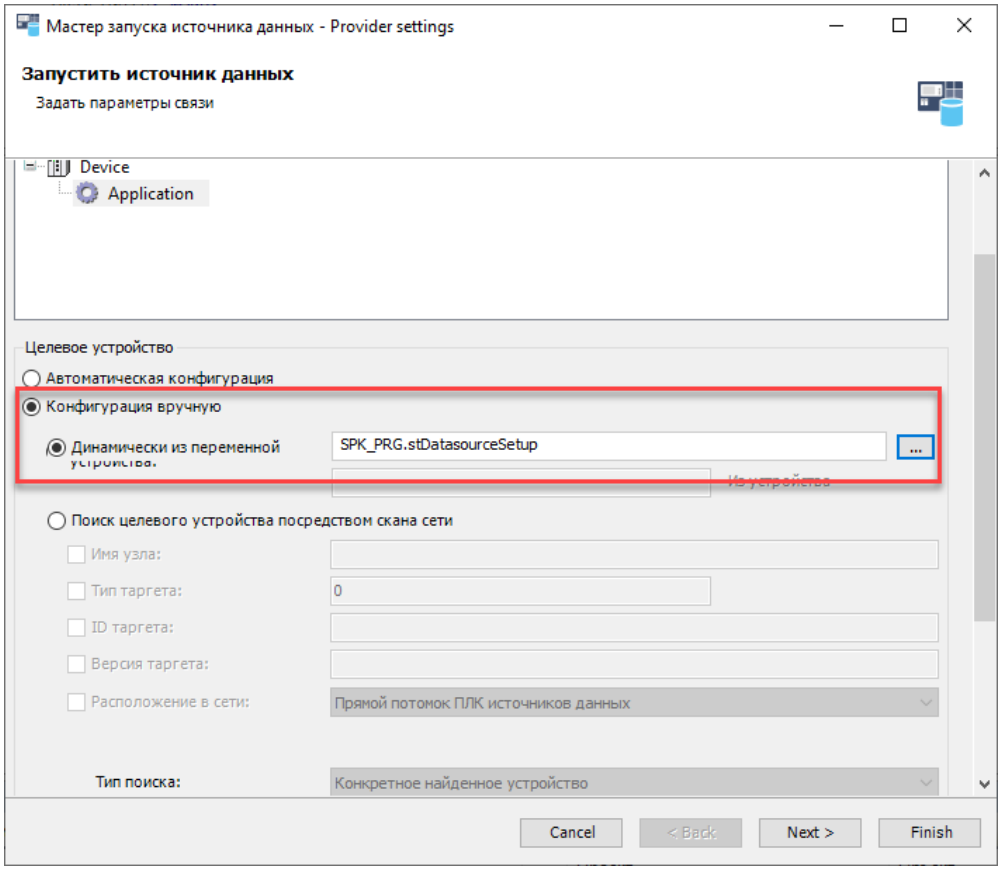

Рисунок 3.3.3 – Привязка переменной устройства в окне настройки обмена

Подключение к удаленному устройству с заданным адресом будет произведено автоматические при запуске проекта. Чтобы отключить связи – следует присвоить полю **xDataValid** значение **FALSE**.

3.4 Использование источников данных в визуализации

Использование менеджера источника данных позволяет без дополнительных настроек получать данные для некоторых элементов визуализации.

Элементы **Таблица тревог** и **Баннер тревог** могут отображать тревоги удаленного устройства. Для этого следует:

- добавить в проект локального устройства компонент **Конфигурация тревог**, настроить его и загрузить проект в контроллер;
- добавить в проект локального устройства компонент **Конфигурация тревог**;
- добавить в конфигурацию тревог локального устройства узел **Удаленные тревоги** (**Remote Alarms**);
- в визуализации для элемента **Таблица тревог** или **Баннер тревог** выбрать источник данных и приложение удаленного устройства.

Других настроек не требуется.

См. видеопример.

Для отображения в таблице тревог названия конкретного удаленного устройства, от которого получена тревога, можно добавить столбец **Удаленное устройство**.

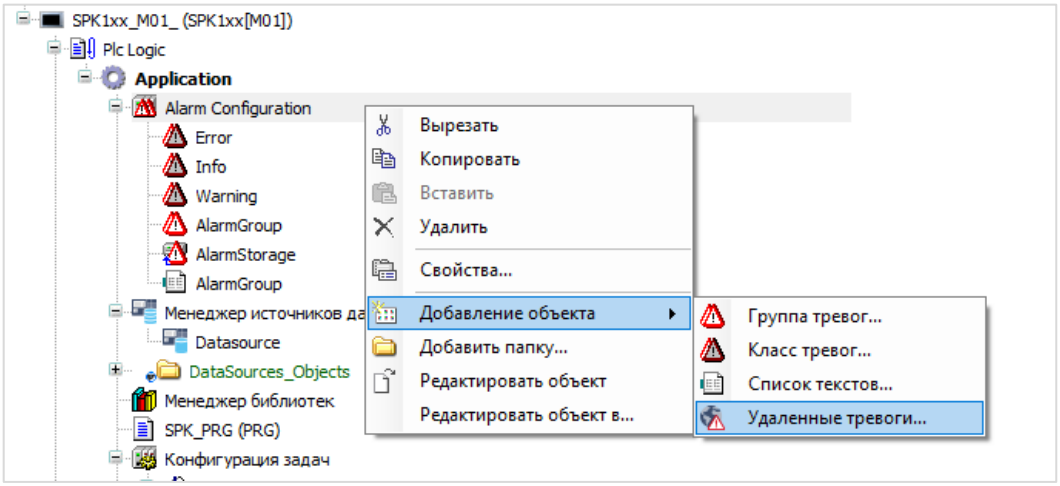

Рисунок 3.4.1 – Добавление узла Удаленные тревоги

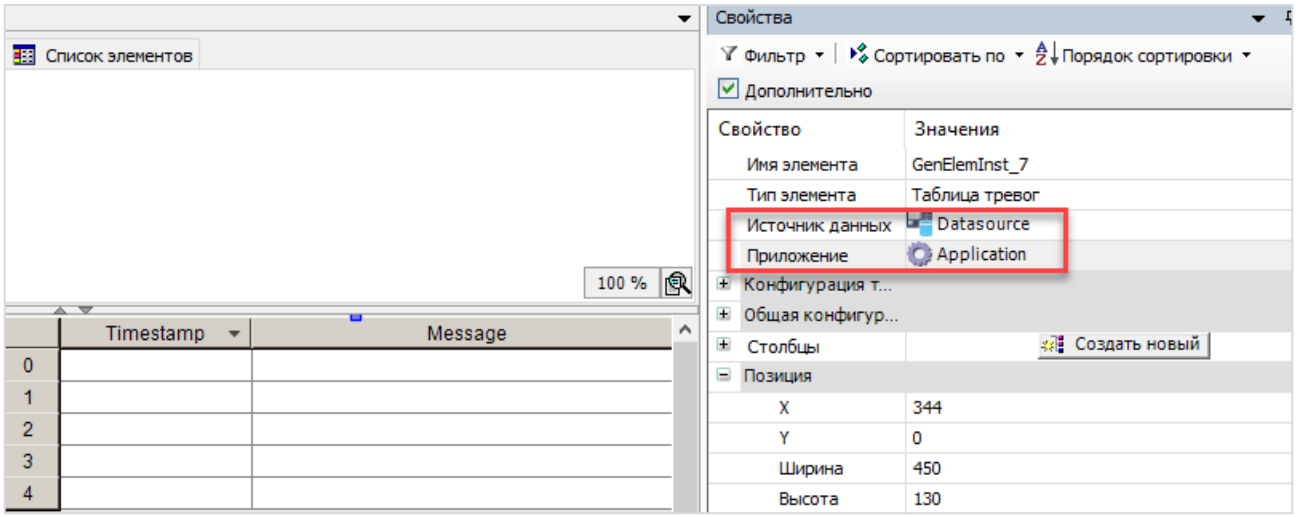

Рисунок 3.4.2 – Выбор источника данных в таблице тревог

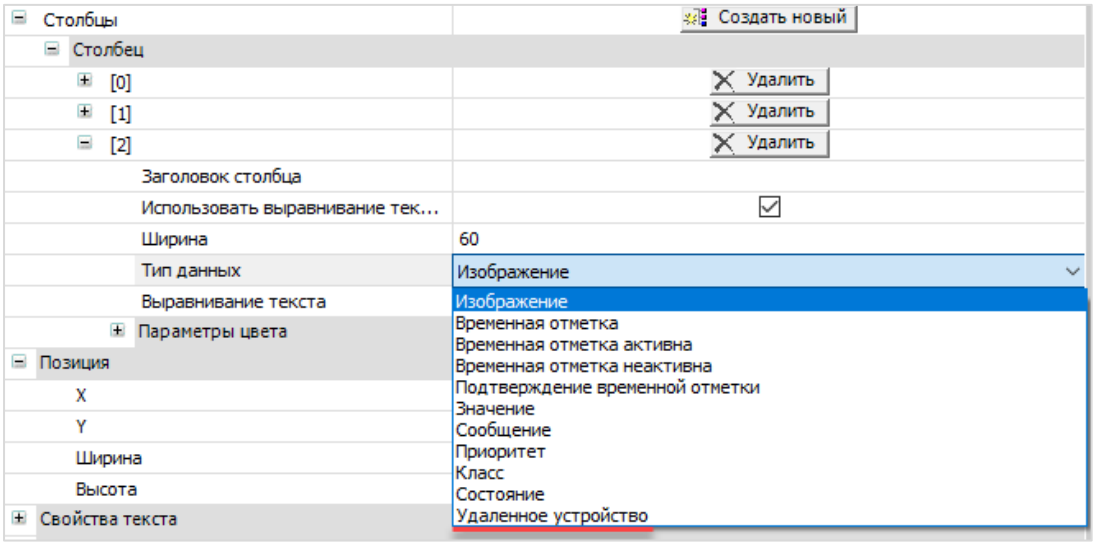

Рисунок 3.4.3 – Добавление столбца с именем удаленного устройства в таблицу тревог

Элементы **Трассировка** и **Тренд** могут отображать значения удаленного устройства. Для этого следует в их настройках выбрать источник данных и приложение удаленного устройства. Для тренда название записей тренда (TrendRecording) в удаленном и локальном устройстве должны совпадать. Других настроек не требуется (в частности, не надо импортировать в локальное устройство переменные тренда удаленного устройства и добавлять их на тренд локального устройства – данные будут считываться автоматически и без этого).

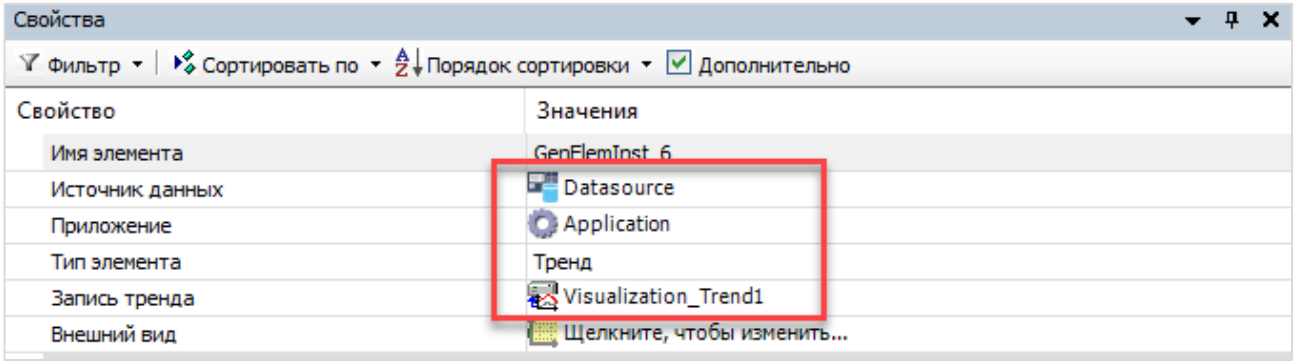

Рисунок 3.4.4 – Выбор источника данных в тренде

Менеджер рецептов синхронизирует рецепты локального и удаленного устройства, если хотя бы одна переменная рецепта добавлена в источник данных.

ПРИМЕЧАНИЕ

Значения переменных источника данных, напрямую привязанные к элементам визуализации (например, прямоугольникам) локального устройства, не будут отображаться в сервисной визуализации CODESYS (в редакторе визуализации CODESYS при онлайн-подключении к контроллеру).

3.5 Диагностика

Диагностику обмена с локальным устройством в коде программы можно произвести с помощью глобальных системных переменных:

- **g_Datasources.<имя_источника_данных>Error** (тип: Datasources.DataSourceError);
- **g_Datasources.<имя_источника_данных>State** (тип: Datasources.DataSourceMonitoringState).

Имя источника данных можно увидеть в дереве проекта (по умолчанию для первого источника данных это имя – **Datasource**).

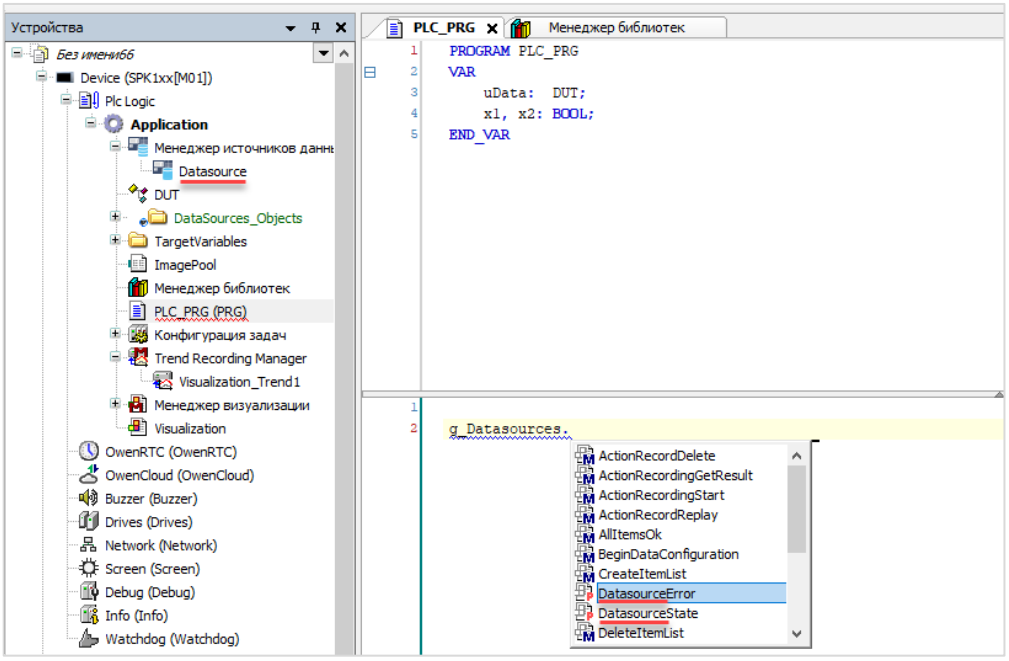

Рисунок 3.5.1 – Обращение к переменным диагностики в коде программы

Перечисления **DataSourceError** и **DataSourceMonitoringState** объявлены в библиотеке **Datasources**, которая автоматически добавляется в проект вместе с менеджером источников данных.

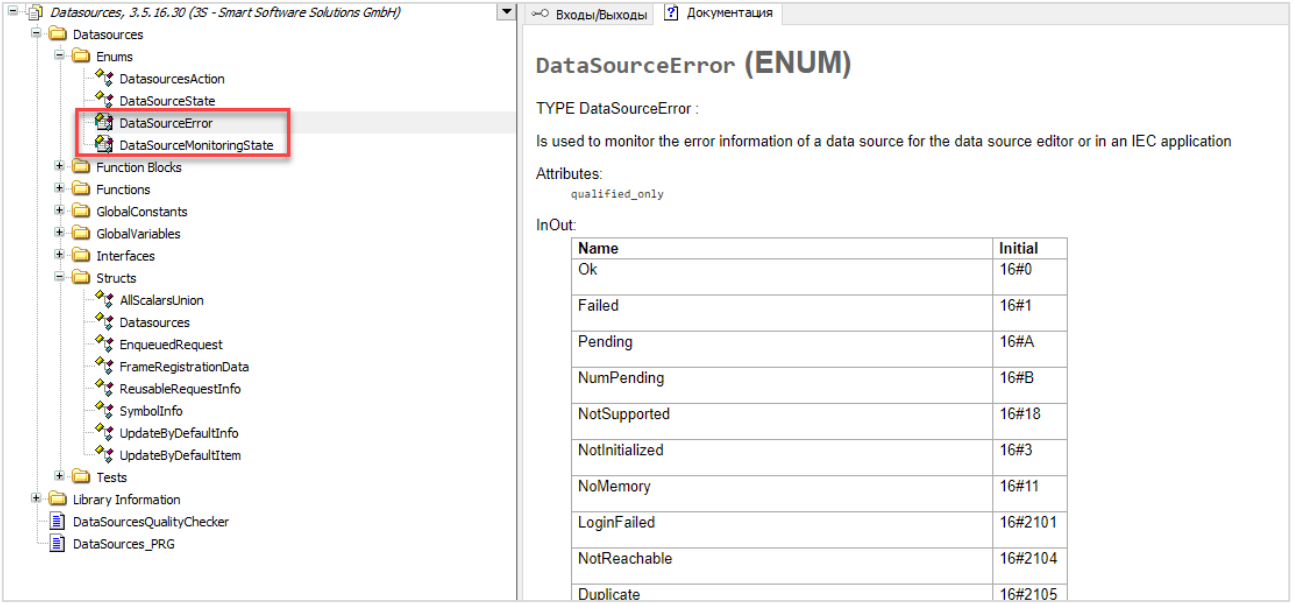

Рисунок 3.5.2 – Перечисления переменных диагностики в библиотеке Datasources