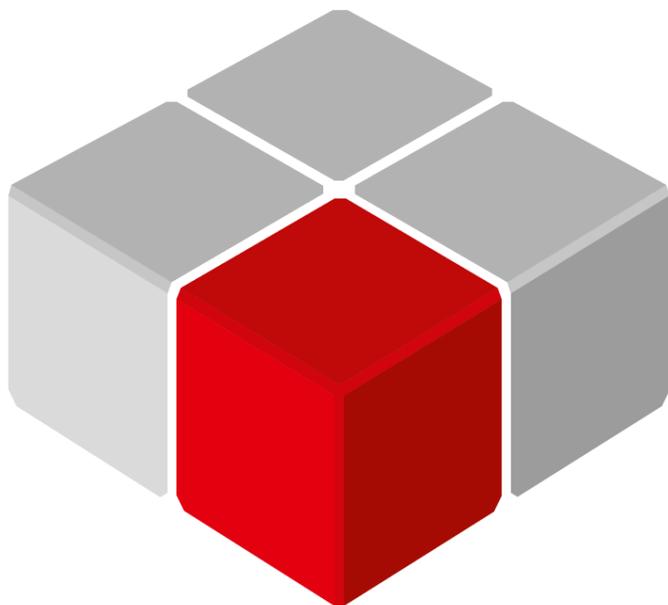




# **CODESYS V3.5**

**Примеры настройки обмена с ПЛК110 [M02]**



**Руководство пользователя**

18.07.2023

версия 3.0

## Оглавление

<b>1</b>	<b>Цель и структура документа</b> .....	<b>3</b>
<b>2</b>	<b>Основные особенности настройки обмена для CDS V3.5 и CDS V2.3</b> .....	<b>4</b>
2.1	CODESYS V3.5: Соответствие нумерации COM-портов контроллера и CODESYS.....	4
2.2	CODESYS V3.5: Использование объединений (UNION).....	4
2.3	CODESYS V3.5: области памяти slave-устройства.....	6
2.4	CoDeSys V2.3: выравнивание данных в Modbus Slave.....	6
<b>3</b>	<b>Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave</b> .....	<b>7</b>
3.1	Описание примера.....	7
3.2	Настройка ПЛК (slave) .....	8
3.3	Настройка СПК (master).....	13
3.4	Работа с примером.....	19
<b>4</b>	<b>Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master</b> .....	<b>21</b>
4.1	Описание примера.....	21
4.2	Настройка СПК (slave) .....	22
4.3	Настройка ПЛК (master).....	26
4.4	Работа с примером.....	35
<b>5</b>	<b>Modbus TCP. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave</b> .....	<b>37</b>
5.1	Описание примера.....	37
5.2	Настройка ПЛК (slave) .....	37
5.3	Настройка СПК (master).....	42
5.4	Работа с примером.....	48
<b>6</b>	<b>Modbus TCP. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master</b> .....	<b>50</b>
6.1	Описание примера.....	50
6.2	Настройка СПК (slave) .....	51
6.3	Настройка ПЛК (master).....	56
6.4	Работа с примером.....	62

## 1 Цель и структура документа

Настоящее руководство описывает настройку обмена данными между контроллерами, программируемыми в **CODESYS V3.5** и **CoDeSys V2.3** по протоколу **Modbus**. Подробные сведения о работе с **Modbus** приведены в других документах – **CODESYS V3.5. Протокол Modbus** и **Руководстве пользователя ПЛК1xx [M02] / Руководство по программированию ПЛК100/150/154**, которые доступны на сайте компании [ОВЕН](#).

Руководство предполагает, что пользователь уже имеет базовые навыки работы с соответствующими приборами.

Контроллеры могут работать как в режиме **Master**, так и в режиме **Slave**. Пользователь должен выбрать режим работы каждого устройства, основываясь на специфике конкретной задачи и используемого оборудования.

Документ содержит 4 примера:

1. [Протокол Modbus RTU. ПЛК \(CDS V3.5\) – master, ПЛК \(CDS V2.3\) – slave.](#)
2. [Протокол Modbus RTU. ПЛК \(CDS V3.5\) – slave, ПЛК \(CDS V2.3\) – master.](#)
3. [Протокол Modbus TCP. ПЛК \(CDS V3.5\) – master, ПЛК \(CDS V2.3\) – slave.](#)
4. [Протокол Modbus TCP. ПЛК \(CDS V3.5\) – slave, ПЛК \(CDS V2.3\) – master.](#)

## 2 Основные особенности настройки обмена для CDS V3.5 и CDS V2.3

### 2.1 CODESYS V3.5: Соответствие нумерации COM-портов контроллера и CODESYS

Во время настройки интерфейсов RS-232/RS-485 в **CODESYS V3.5** следует указывать номера портов. Номера портов перечислены в руководстве **CODESYS V3.5. FAQ**.

### 2.2 CODESYS V3.5: Использование объединений (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 данном 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 тип данных объединение (Union). Во время настройки ПЛК в **CoDeSys V2.3** это не требуется, т. к. у пользователя уже есть готовые модули для каждого типа данных.

**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 приложение **Application** и добавить объект **DUT** типа **объединение** с названием **Real\_Word**:

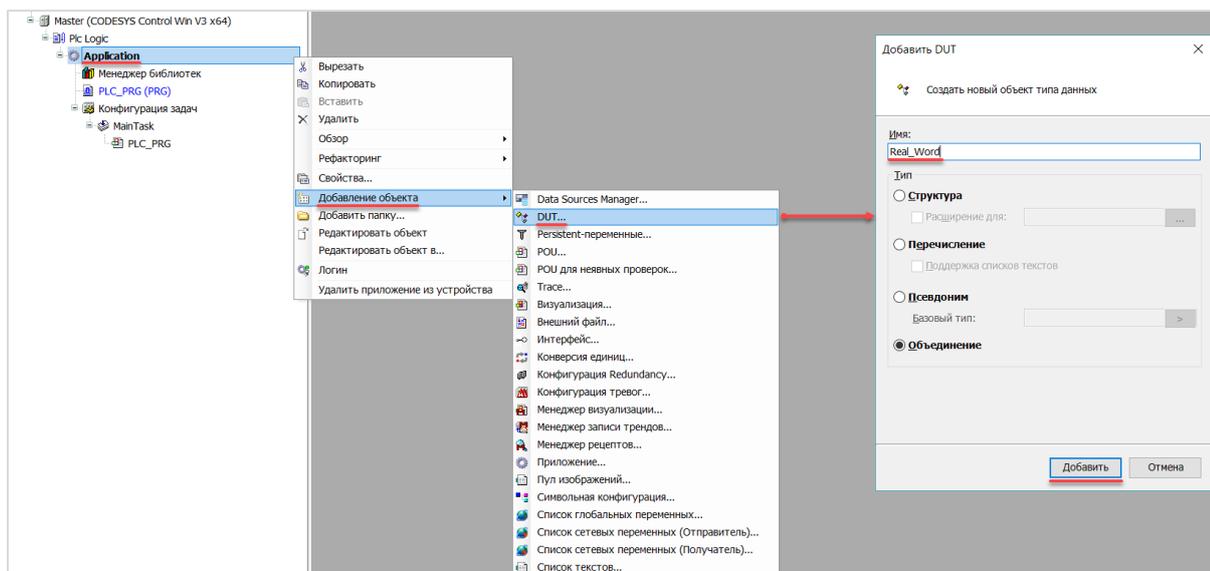


Рисунок 2.1 – Добавление в проект объединения

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

```

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

Рисунок 2.2 – Объявление переменных объединения

- В программе объявить экземпляр объединения **Real\_Word** с названием **\_2WORD\_TO\_REAL**:

```

1  PROGRAM PLC_PRG
2  VAR
3      _2WORD_TO_REAL: Real_Word;
4  END_VAR
5
    
```

Рисунок 2.3 – 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:

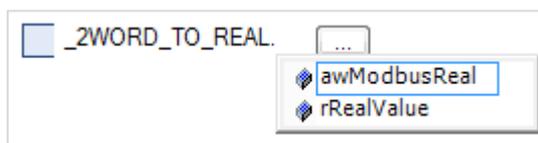


Рисунок 2.4 – Работа с переменными объединения в программе

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

На рисунке 2.5 приведен скриншот значения переменных объединения в процессе работы программы. Во время записи в элементы массива **awModbusReal** значений в HEX, в переменную **rRealValue** будет записано соответствующее им значение с плавающей точкой.

Device.Application.PLC_PRG		
Выражение	Тип	Значение
▣ awModbusReal	ARRAY [0..1] OF WORD	
awModbusReal[0]	WORD	16#CCCD
awModbusReal[1]	WORD	16#3F8C
▣ rRealValue	REAL	1.1
▣ _2WORD_TO_REAL	REAL_WORD	

Рисунок 2.5 – Отображение значений переменных объединения

### 2.3 CODESYS V3.5: области памяти slave-устройства

Во время настройки ПЛК в **CODESYS V3.5** в режиме **Modbus Slave** следует обращать внимание на следующие особенности:

1. **Input Registers** (регистры ввода) доступны только для чтения, **Holding Registers** (регистры временного хранения) – для чтения и записи.
2. **Discrete Inputs** (дискретные входы) доступны только для чтения, **Coils** («катушки», «регистры») – для чтения и записи.
3. **По умолчанию** в реализации CODESYS области памяти битов и регистров наложены друг на друга (общие области для **Coils/Holding регистров** и **Discrete Inputs/Input регистров**). Чтобы разделить указанные области – необходимо установить **Дискретные битовые области** в настройках slave'a.
4. **По умолчанию** в реализации CODESYS значения **holding регистров** и **coils'ов** не могут быть изменены из кода программы контроллера (только со стороны master-устройства). Чтобы holding регистры и coils стали доступны для изменения из кода программы контроллера – необходимо установить галочку **Запись** в настройках slave'a.

### 2.4 CoDeSys V2.3: выравнивание данных в Modbus Slave

Во время настройки ПЛК в **CoDeSys V2.3** в режиме **Modbus Slave** следует выравнивать данные по регистрам (регистр занимает 2 байта, т. е. 16 бит). Следует учитывать, что адреса регистров размещаемых данных должны быть кратны количеству байт этих данных.

В качестве примера в slave-устройстве будет использоваться элемент **8 bits**, два элемента **2 byte** и элемент **float**. Размещать их последовательно нельзя. Выравненные данные в данном случае будет выглядеть следующим образом:

Таблица 2.1 – Карта регистров slave-устройства

Элемент	Комментарий	Адреса регистров	Адреса битов (в абсолютной адресации)
8 bits		0	0...7
8 bits	Добавлен для выравнивания		8...15
2 byte		1	16...31
2 byte		2	32...47
2 byte	Добавлен для выравнивания	3	48...63
float		4	64...79

Более подробная информация о выравнивании содержится в документе **Руководство пользователя ПЛК1xx [M02]**, доступном на сайте компании [ОВЕН](#).

### 3 Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

#### 3.1 Описание примера

Пример посвящен настройке обмена данными между сенсорным панельным контроллером **СПК1хх [M01]** и контроллером **ПЛК110 [M02]** по протоколу **Modbus RTU**. В примере СПК выполняет функцию **Master**, а ПЛК – **Slave**.

Основные характеристики используемых устройств приведены в таблице 3.1.

Используемые в примере переменные описаны в таблице 3.2.

**Таблица 3.1 – Характеристики устройств**

Устройство	СПК1хх [M01]	ПЛК110 [M02]
Функция	Master	Slave
Используемый порт (нумерация на корпусе)	RS-485-1	RS-485 (1)
Настройки обмена	115200, 8 бит, 1 стоп бит, без контроля четности	
Slave ID	–	1
Таргет	3.5.17.31	PLC110.30-M v2 (версия 3.3)
Версия прошивки	2.4.0923.1000	1.2.42
Среда разработки проекта	CODESYS V3.5 SP17 Patch 3	CoDeSys V2.3.9.41
Название файла проекта	ModbusRTUmaster.projectarchive	ModbusRTUslave.pro

**Таблица 3.2 – Список переменных**

СПК1хх[M01] (Master)			ПЛК110 [M02] (Slave)	
Переменные, в которые считываются значения из Slave	Переменные, значения которых записываются в Slave	Тип данных	Переменные ПЛК	Адрес регистра/бита
xVarRead	xVarWrite	BOOL	xVar	0/0
wVarRead	wVarWrite	WORD	wVar	1
rVarRead	rVarWrite	REAL	rVar	2–3
sVarRead	sVarWrite	STRING(6)	sVar	4–6

Проекты примера доступны для скачивания: [Example\\_SpkModbusRtuMaster.zip](#)

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

## 3.2 Настройка ПЛК (slave)

Для настройки ПЛК (slave) следует:

1. Создать новый проект **CoDeSys V2.3** для ПЛК110 с программой **PLC\_PRG** на языке **CFC**.
2. В компоненте **Конфигурация ПЛК** (вкладка **Ресурсы**) настроить регистры Modbus и привязать к ним переменные.

Нажать **ПКМ** на название контроллера (в данном примере – **PLC110\_30**) и добавить подэлемент **Modbus (Slave)**:

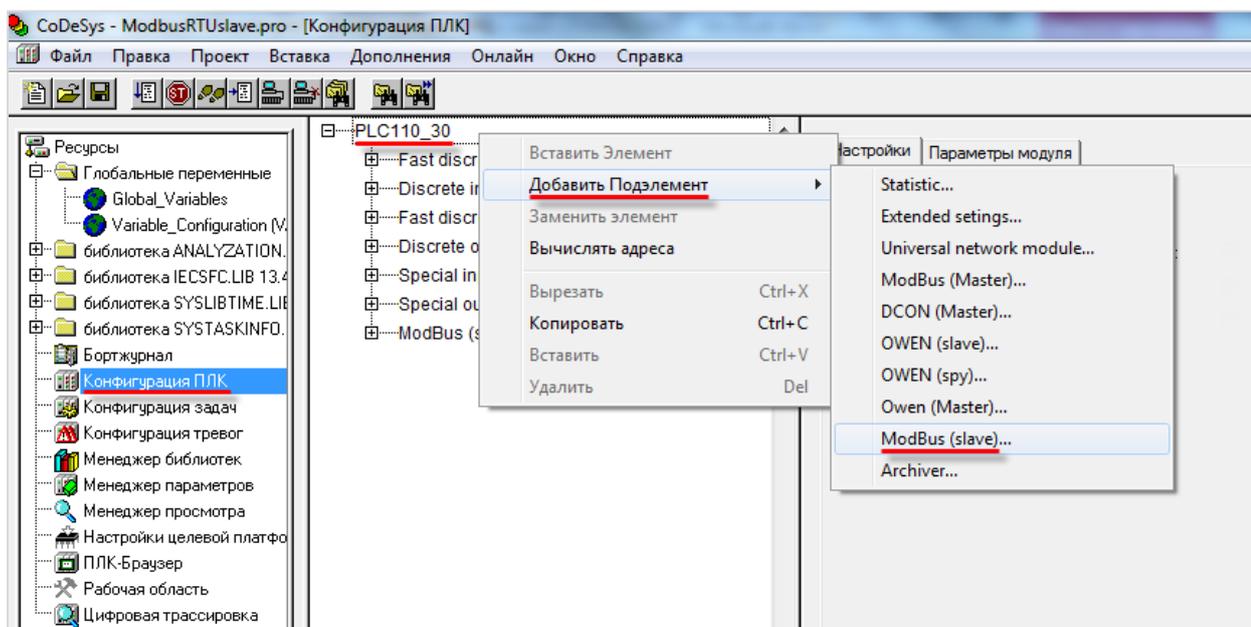


Рисунок 3.1 – Конфигурация ПЛК. Добавление Modbus (Slave)

В его настройках выбрать адрес (**Slave ID**), равный **1** (в соответствии с [таблицей 3.1](#)):

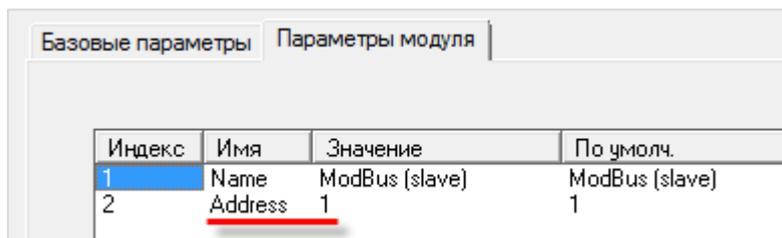


Рисунок 3.2 – Конфигурация ПЛК. Настройка Modbus (Slave)

3. Выбрать порт ПЛК, который будет использоваться для связи с контроллером. Для выбора следует нажать **ПКМ** на элемент **Modbus (FIX)** и добавить подэлемент **RS-485-1** (согласно [таблице 3.1](#)).

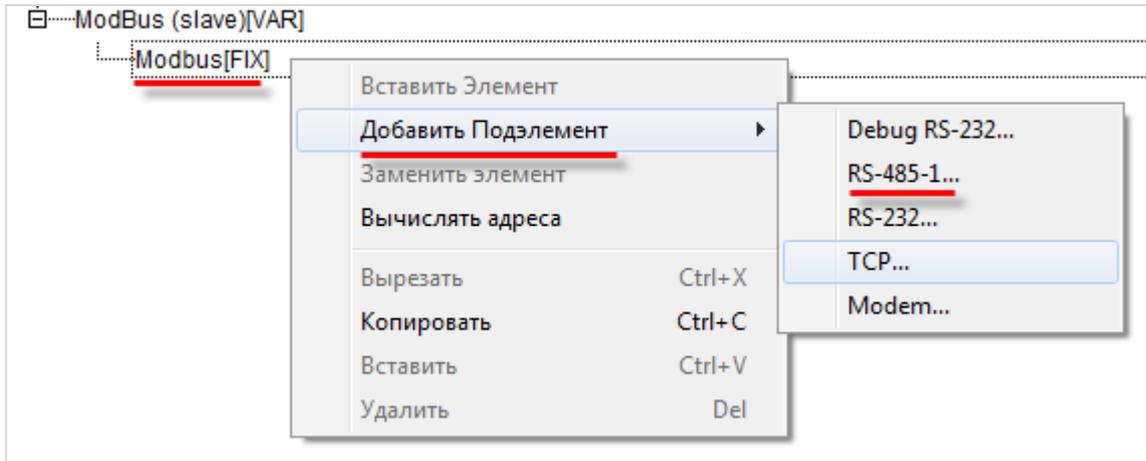


Рисунок 3.3 – Конфигурация ПЛК. Добавление подэлемента RS-485-1

В параметрах подэлемента указать сетевые настройки в соответствии с [таблицей 3.1](#) и используемый протокол – **RTU**.

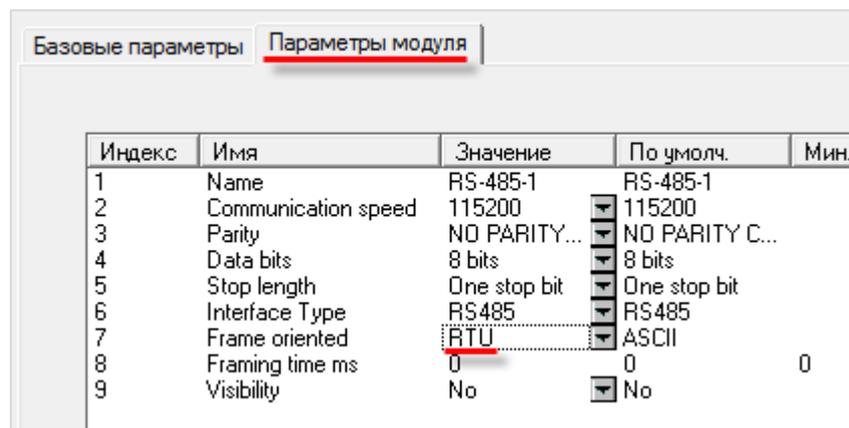


Рисунок 3.4 – Конфигурация ПЛК. Настройки подэлемента RS-485-1

4. Нажать **ПКМ** на элемент **Modbus (Slave)** и добавить следующие подэлементы:
- 8 bits (для BOOL);
  - 8 bits (для обеспечения [выравнивания памяти](#));
  - 2 byte (для WORD);
  - Float (для REAL);
  - 3 элемента 2 byte (для **STRING** из 6 символов).

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

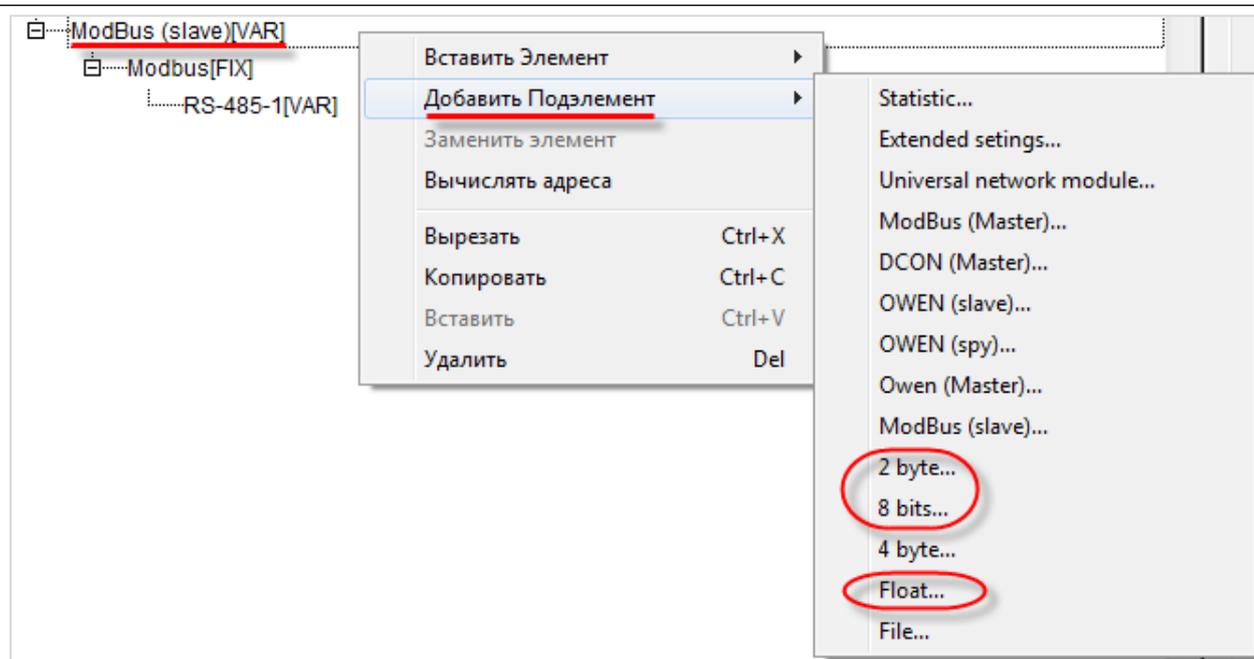


Рисунок 3.5 – Конфигурация ПЛК. Добавление подэлементов

В результате **Конфигурация ПЛК** будет выглядеть следующим образом (см. рисунок 3.6). Затем следует объявить переменные (после ввода имен они автоматически будут добавлены в список глобальных переменных проекта). Для ввода имени переменной два раза нажмите на **AT**.



#### ПРИМЕЧАНИЕ

К регистрам 4–6 переменные не привязываются. Соответствующая им переменная **sVar** будет объявлена в программе **PLC\_PRG** (в пп. 5).

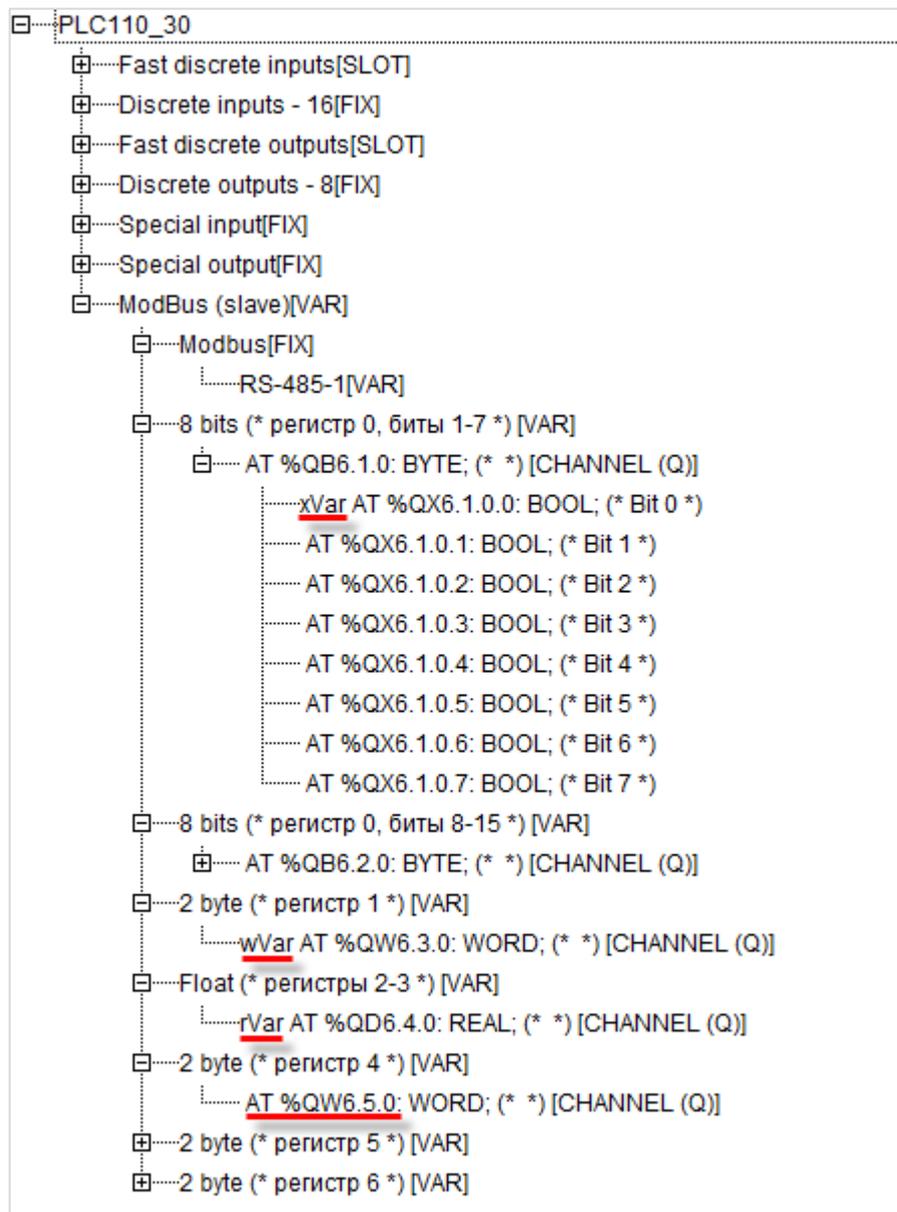


Рисунок 3.6 – Внешний вид Modbus (Slave) с добавленными подэлементами

**ПРИМЕЧАНИЕ**

Нумерация регистров в CODESYS всегда начинается с нуля, каждый регистр физически занимает два байта (16 бит). Поэтому переменная типа **REAL** займет два регистра (с адресами 2 и 3). Переменная типа **STRING**, которой соответствует три **2 byte** элемента, займет регистры с адресами 4–6. Это необходимо учитывать во время настройки master-устройства.

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

5. Программа **PLC\_PRG** будет выглядеть следующим образом:

```
0001 PROGRAM PLC_PRG
0002 VAR
0003   sVar AT %QW6.5.0: STRING(6);           (*собираем STRING переменную из трех WORD [т.е. шести символов],
0004                                           указывая адрес [см. Конфигурация ПЛК] первого из них*)
0005 END_VAR
0006
0007
0008
0009
```



The screenshot shows a software interface for a PLC program. At the top, there is a code editor with the following text: `0001 PROGRAM PLC_PRG`, `0002 VAR`, `0003 sVar AT %QW6.5.0: STRING(6);`, `0004`, `0005 END_VAR`, `0006`, `0007`, `0008`, and `0009`. A comment in Russian explains that the STRING variable is built from three WORDs. Below the code editor is a control panel with a text box containing the instruction: "СПК считывает/записывает значения из ПЛК" and "Изменяйте значения переменных и наблюдайте соответствующие изменения в СПК". Below this are four buttons labeled xVar, wVar, rVar, and sVar.

Рисунок 3.7 – Код программы PLC\_PRG

Настройка ПЛК (**slave**) завершена.



#### ПРИМЕЧАНИЕ

Данный проект не содержит операций и используется только для отображения и ввода значений. Пользователь должен создать программу для реализации необходимых алгоритмов.

### 3.3 Настройка СПК (master)

Для настройки СПК (master) следует:

1. Создать новый проект **CODESYS V3.5** для СПК1хх [M01] с программой **PLC\_PRG** на языке **CFC**.
2. Добавить в проект [объединение](#) с именем **Real\_Word**:

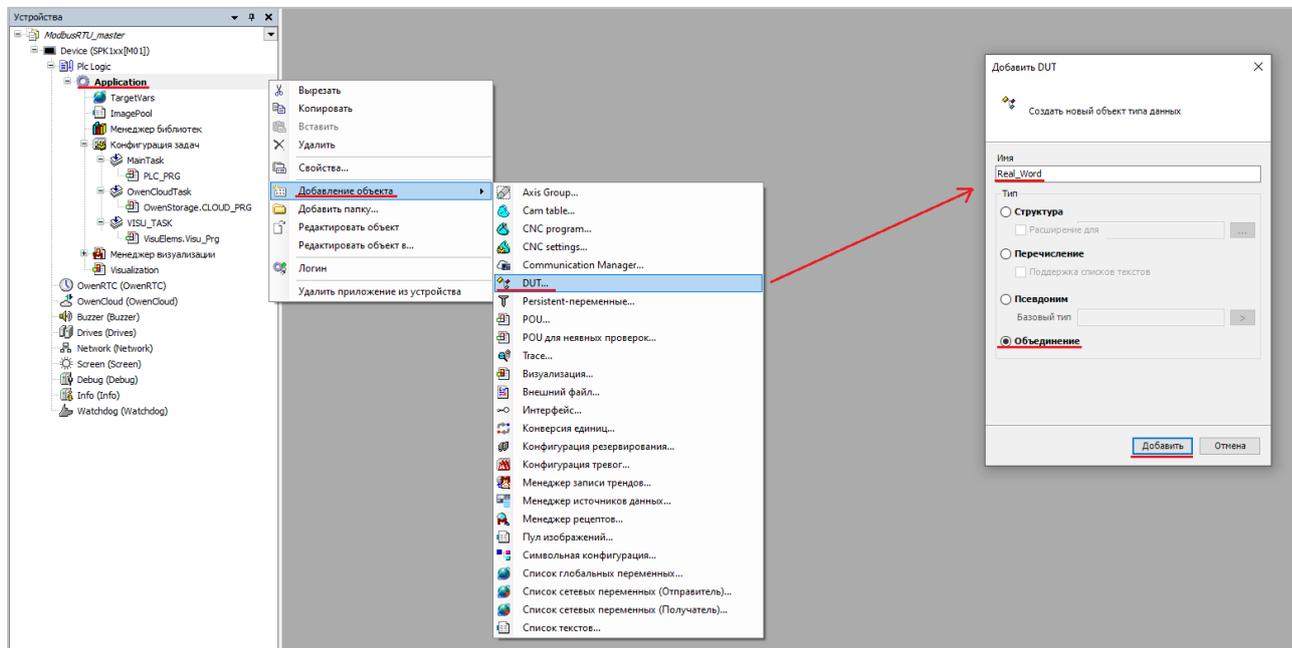


Рисунок 3.8 – Добавление в проект объединения

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

```

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

```

Рисунок 3.9 – Объявление переменных объединения

3. Добавить в проект [объединение](#) с именем **String\_Word**.

В объединении объявить переменную **sStringValue** типа **STRING** и массив **awModbusString** типа **WORD**, содержащий три элемента (**STRING** может содержать до 6 символов, поскольку каждый **WORD** может содержать два символа):

```

1  TYPE String_Word :
2  UNION
3      sStringValue: STRING(6);
4      awModbusString: ARRAY [0..2] OF WORD;
5  END_UNION
6  END_TYPE

```

Рисунок 3.10 – Объявление переменных объединения

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

4. Объявить в программе **PLC\_PRG** девять переменных – 4 из них будут использоваться для отображения данных, считанных из ПЛК, еще 4 – для ввода данных, которые будут записаны в ПЛК. Последняя переменная будет являться триггером записи.

```
1  PROGRAM PLC_PRG
2  VAR
3      (*данные, из считываемые ПЛК*)
4      xVarRead:      BOOL;
5      wVarRead:      WORD;
6      rVarRead:      Real_Word;
7      sVarRead:      String_Word;
8
9      (*данные, записываемые в ПЛК*)
10     xVarWrite:     BOOL;
11     wVarWrite:     WORD;
12     rVarWrite:     Real_Word;
13     sVarWrite:     String_Word;
14
15     xTrigger:      BOOL; // триггер записи
16 END_VAR
```

Рисунок 3.11 – Объявление переменных программы

5. Код программы будет выглядеть следующим образом:

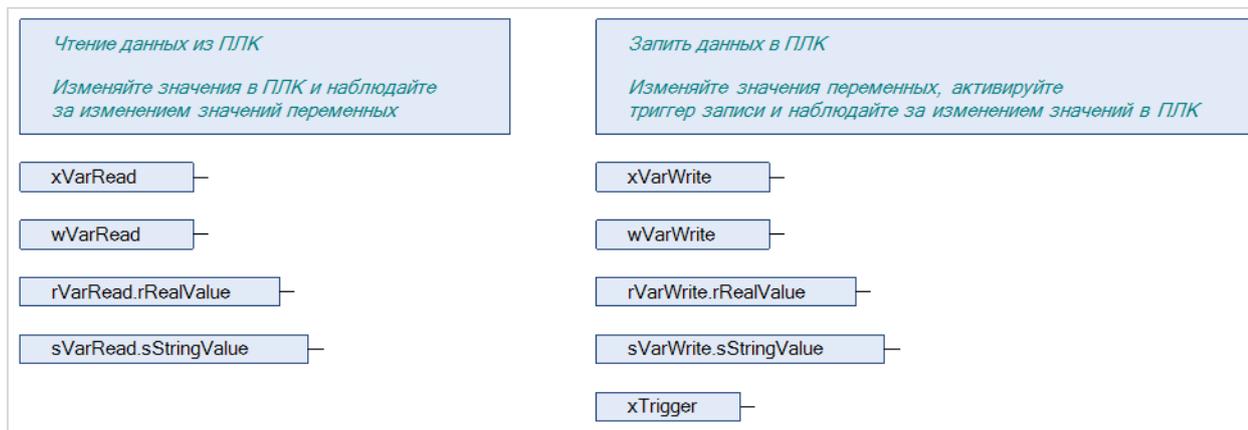


Рисунок 3.12 – Код программы на языке CFC

6. Добавить в проект устройство **Modbus COM**.



#### ПРИМЕЧАНИЕ

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – не может.

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

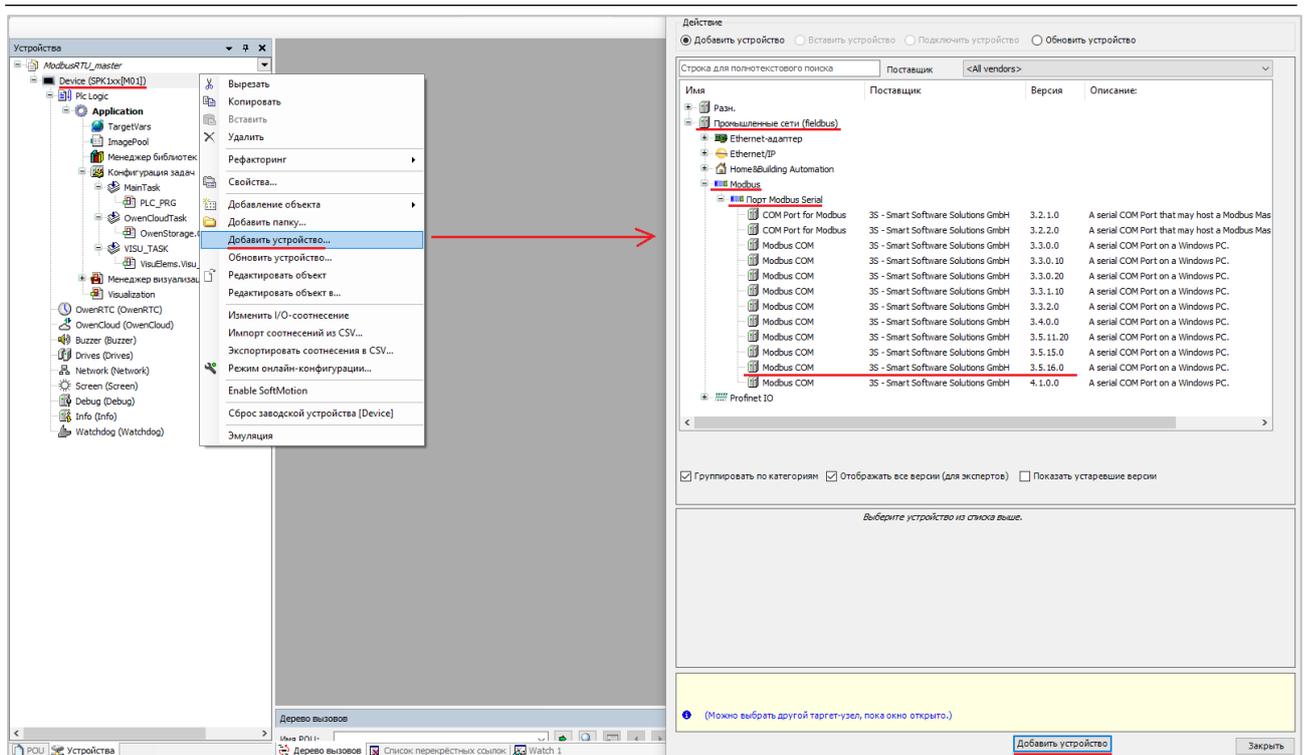


Рисунок 3.13 – Добавление устройства Modbus COM

В конфигурации COM-порта следует указать сетевые настройки в соответствии с [таблицей 3.1](#) и номер порта. Нумерация COM-портов приведена в документе **CODESYS V3.5. FAQ**.

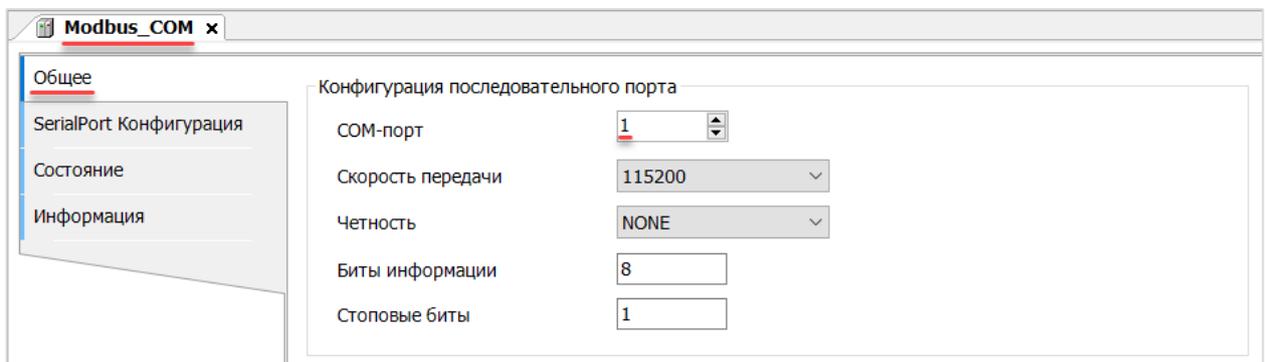


Рисунок 3.14 – Настройки COM-порта COM1

7. В COM-порт добавить компонент **Modbus Master**.



#### ПРИМЕЧАНИЕ

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – не может.

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

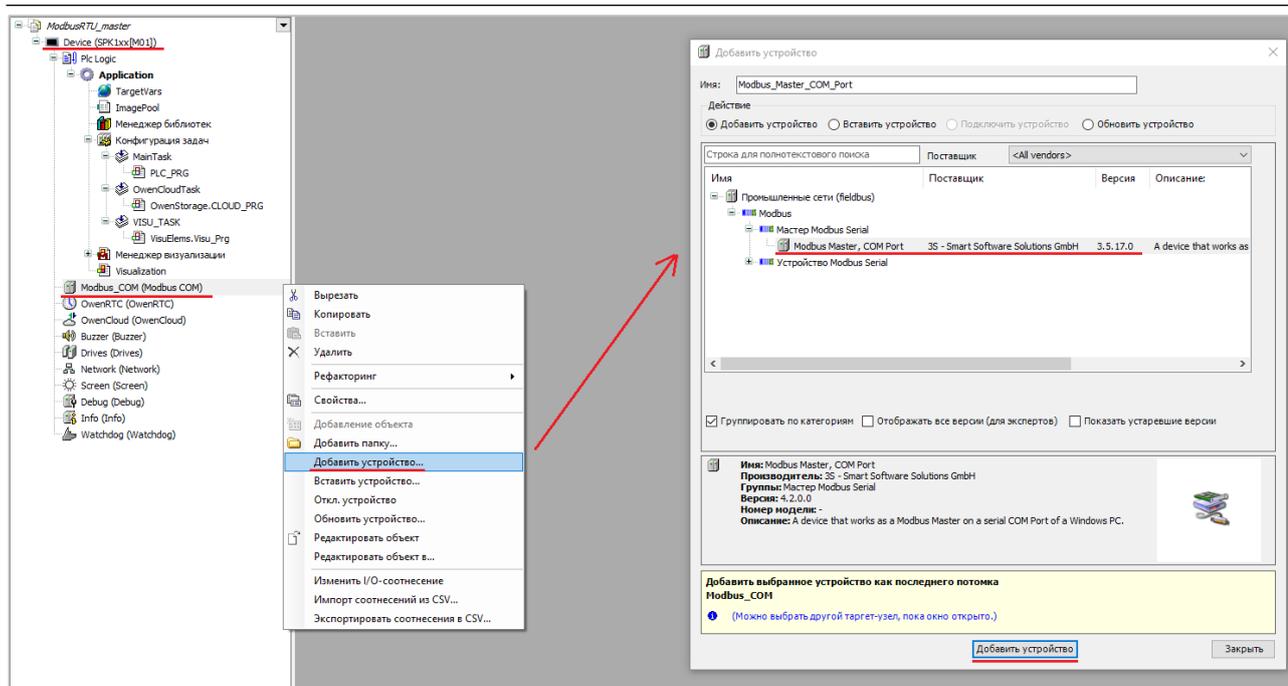


Рисунок 3.15 – Добавление компонента Modbus Master

В настройках компонента поставить галочку **Автоперезапуск соединения**.

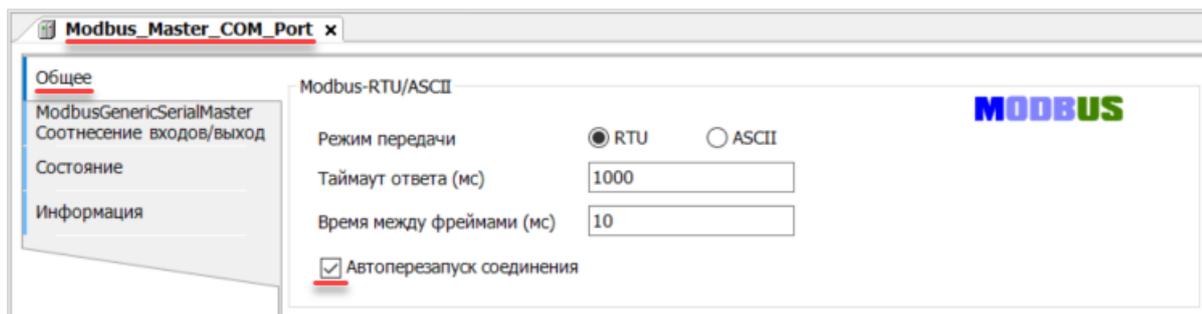


Рисунок 3.16 – Настройка компонентов Modbus Master

8. В Modbus Master добавить компонент Modbus Slave.



#### ПРИМЕЧАНИЕ

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – не может.

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

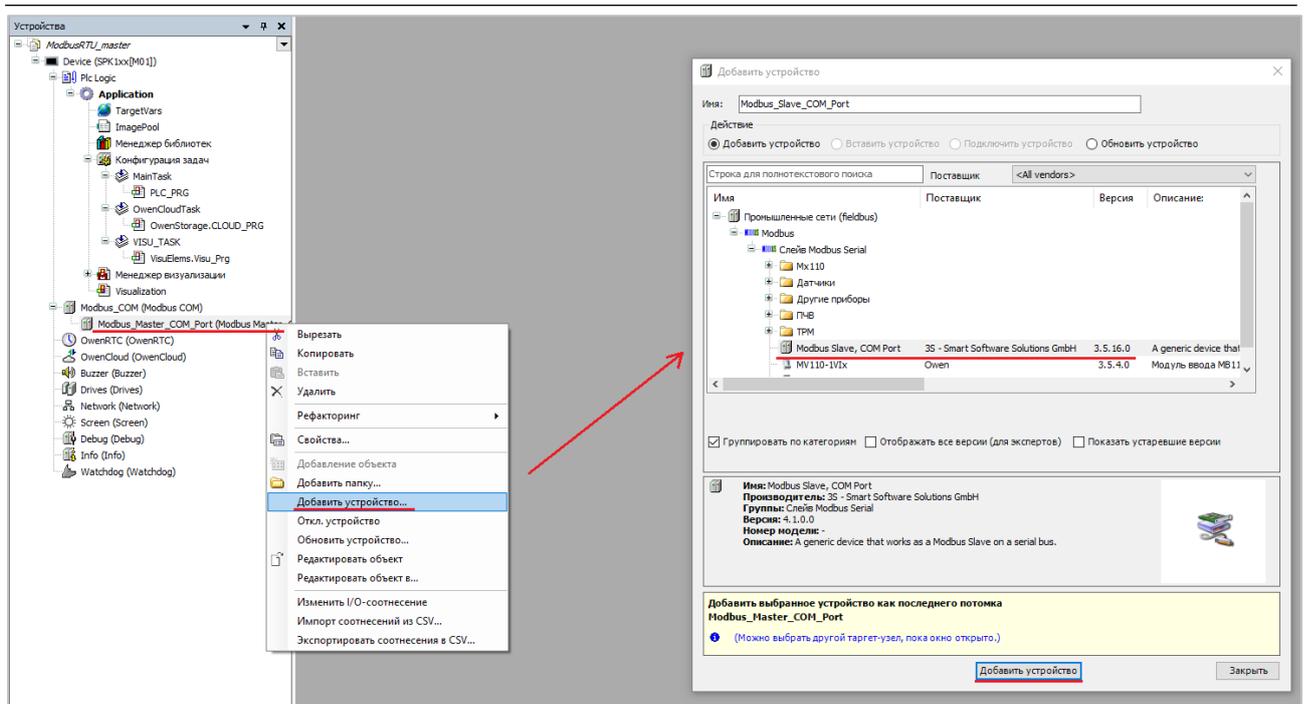


Рисунок 3.17 – Добавление компонента Modbus Slave в проект

В настройках компонента на вкладке **Общие** следует указать адрес slave-устройства в соответствии с [таблицей 3.1.](#)

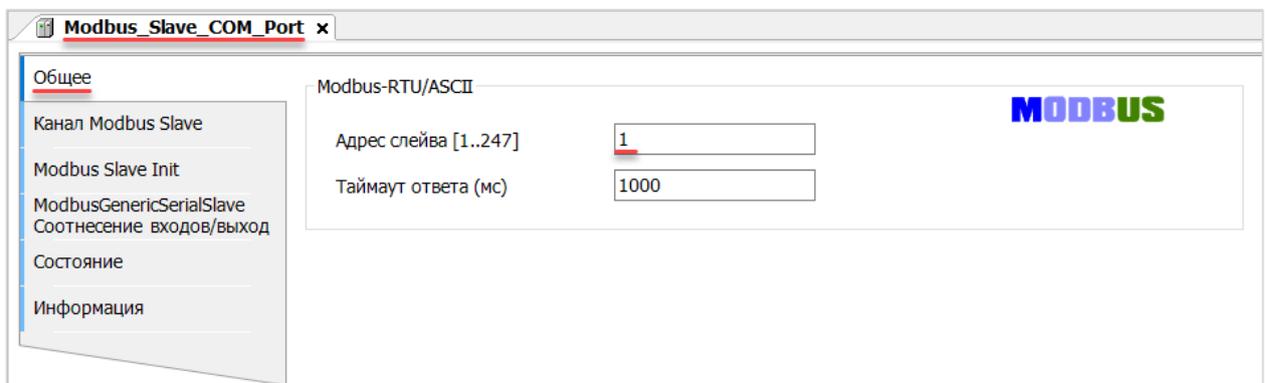


Рисунок 3.18 – Настройки компонента Modbus Slave в проект

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

На вкладке **Канал Modbus Slave** следует создать 8 каналов – 4 из них будут использоваться для чтения переменных, 4 – для записи. Чтение будет осуществляться циклически, запись – по переднему фронту триггера (**RISING\_EDGE**). Используемые функции соответствуют типам данных, адреса регистров настроены согласно [таблице 3.2](#).

Имя	Тип доступа	Триггер	Сдвиг READ	Длина	Обработка ошибок	Сдвиг WRITE	Длина	Комментарий
0 Channel 8	Read Coils (Код функции 01)	Цикл., t#100ms	16#0000	1	Сохранить посл. значение			
1 Channel 1	Read Holding Registers (Код функции 03)	Цикл., t#100ms	16#0001	1	Сохранить посл. значение			
2 Channel 2	Read Holding Registers (Код функции 03)	Цикл., t#100ms	16#0002	2	Сохранить посл. значение			
3 Channel 3	Read Holding Registers (Код функции 03)	Цикл., t#100ms	16#0004	3	Сохранить посл. значение			
4 Channel 4	Write Single Coil (Код функции 05)	Передний фронт				16#0000	1	
5 Channel 5	Write Single Register (Код функции 06)	Передний фронт				16#0001	1	
6 Channel 6	Write Multiple Registers (Код функции 16)	Передний фронт				16#0002	2	
7 Channel 7	Write Multiple Registers (Код функции 16)	Передний фронт				16#0004	3	

Рисунок 3.19 – Настройка каналов Modbus Slave

На вкладке **ModbusGenericSerialSlave Соотнесение входов/выходов** следует привязать к каналам переменные программы в соответствии с [таблицей 3.2](#). У параметра **Всегда обновлять переменные** должно быть выставлено значение **Включено 2**.

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Channel 8	%IB0	ARRAY [0..0] OF BYTE		Read Coils
		Channel 8[0]	%IB0	BYTE		бит 0
Application.PLC_PRG.xVarRead		Bit0	%IX0.0	BOOL		бит 0
		Channel 1	%IW1	ARRAY [0..0] OF WORD		Read Holding Registers
Application.PLC_PRG.wVarRead		Channel 1[0]	%IW1	WORD		регистр 1
Application.PLC_PRG.rVarRead.awModbusRea[0]		Channel 2	%IW2	ARRAY [0..1] OF WORD		Read Holding Registers
Application.PLC_PRG.rVarRead.awModbusRea[1]		Channel 2[0]	%IW2	WORD		регистр 2
		Channel 2[1]	%IW3	WORD		регистр 3
		Channel 3	%IW4	ARRAY [0..2] OF WORD		Read Holding Registers
Application.PLC_PRG.sVarRead.awModbusString[0]		Channel 3[0]	%IW4	WORD		регистр 4
Application.PLC_PRG.sVarRead.awModbusString[1]		Channel 3[1]	%IW5	WORD		регистр 5
Application.PLC_PRG.sVarRead.awModbusString[2]		Channel 3[2]	%IW6	WORD		регистр 6
Application.PLC_PRG.xTrigger		Channel 4	%IX0.0	BIT		Trigger Variable
		Channel 4	%QB1	ARRAY [0..0] OF BYTE		
		Channel 4[0]	%QB1	BYTE		Write Single Coil
Application.PLC_PRG.xVarWrite		Bit0	%IX1.0	BOOL		бит 0
Application.PLC_PRG.xTrigger		Channel 5	%IX2.0	BIT		Trigger Variable
		Channel 5	%QW2	ARRAY [0..0] OF WORD		Write Single Register
Application.PLC_PRG.wVarWrite		Channel 5[0]	%QW2	WORD		регистр 1
Application.PLC_PRG.xTrigger		Channel 6	%IX6.0	BIT		Trigger Variable
		Channel 6	%QW4	ARRAY [0..1] OF WORD		Write Multiple Registers
Application.PLC_PRG.rVarWrite.awModbusRea[0]		Channel 6[0]	%QW4	WORD		регистр 2
Application.PLC_PRG.rVarWrite.awModbusRea[1]		Channel 6[1]	%QW5	WORD		регистр 3
Application.PLC_PRG.xTrigger		Channel 7	%IX12.0	BIT		Trigger Variable
		Channel 7	%QW7	ARRAY [0..2] OF WORD		Write Multiple Registers
Application.PLC_PRG.sVarWrite.awModbusString[0]		Channel 7[0]	%QW7	WORD		регистр 4
Application.PLC_PRG.sVarWrite.awModbusString[1]		Channel 7[1]	%QW8	WORD		регистр 5
Application.PLC_PRG.sVarWrite.awModbusString[2]		Channel 7[2]	%QW9	WORD		регистр 6

Сброс соотнесения Всегда обновлять переменные: **Вкл. 2 (всегда в задаче цикла шины)**

Рисунок 3.20 – Привязка переменных к каналам

Настройка **СПК (master)** завершена.



#### ПРИМЕЧАНИЕ

Проект не содержит операций и используется только для отображения и ввода значений. Пользователь должен создать программу для реализации необходимых алгоритмов.

### 3.4 Работа с примером

Проекты следует загрузить проекты в оба устройства и запустить.

В случае изменения значения переменных в ПЛК, соответствующие изменения будут отображаться в программе СПК:

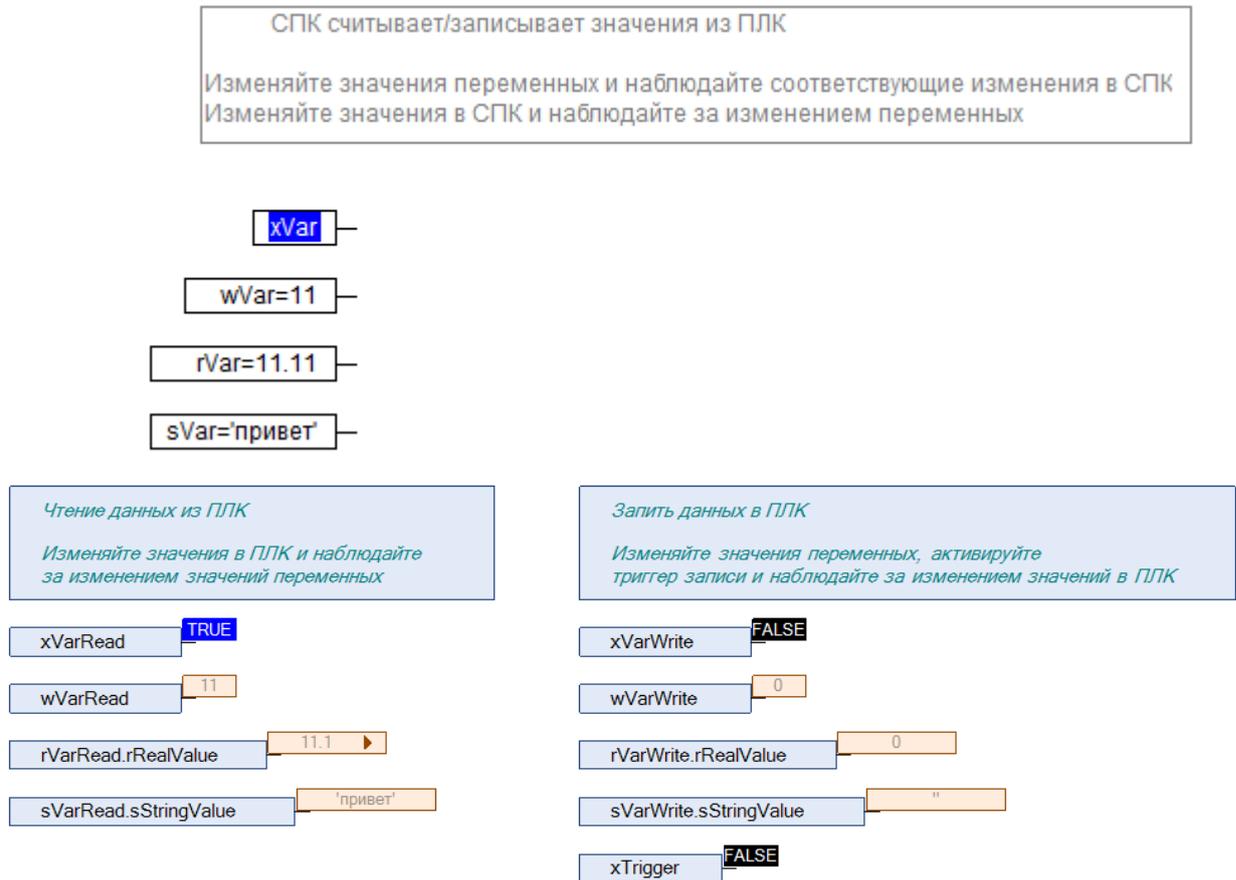


Рисунок 3.21 – СПК считывает данные из ПЛК

### 3. Modbus RTU. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

Затем следует изменить значения **write** переменных СПК и активировать триггер записи. Соответствующие изменения будут отображаться в программе ПЛК. Также новые значения будут считаны в **read** переменные программы СПК.



Рисунок 3.22 – СПК записывает данные в ПЛК

## 4 Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

### 4.1 Описание примера

Пример посвящен настройке обмена данными между сенсорным панельным контроллером **СПК1xx [M01]** и контроллером **ПЛК110 [M02]** по протоколу **Modbus RTU**. В примере СПК выполняет функцию **Slave**, а ПЛК – **Master**.

Основные характеристики используемых устройств приведены в таблице 4.1.

Используемые в примере переменные описаны в таблице 4.2.

**Таблица 4.1 – Характеристики устройств**

Устройство	СПК1xx [M01]	ПЛК110 [M02]
Функция	Slave	Master
Используемый порт (нумерация на корпусе)	RS-485-1	RS-485 (1)
Настройки обмена	115200, 8 бит, 1 стоп бит, без контроля четности	
Slave ID	1	–
Таргет	3.5.17.31	PLC110.30-M v2 (версия 3.3)
Версия прошивки	2.4.0923.1000	1.2.42
Среда разработки проекта	CODESYS V3.5 SP17 Patch 3	CoDeSys V2.3.9.41
Название файла проекта	ModbusRTUslave.projectarchive	ModbusRTUmaster.pro

**Таблица 4.2 – Список переменных**

СПК1xx [M01] (Slave)		ПЛК110 [M02] (Master)	
Переменные, которые считывает Master	Адрес регистра/бита	Тип данных	Переменные ПЛК
xVarFromSPK	0 (Discrete Inputs)	BOOL	xVarFromSPK
wVarFromSPK	1 (Input Registers)	WORD	wVarFromSPK
rVarFromSPK	2–3 (Input Registers)	REAL	rVarFromSPK
sVarFromSPK	4–6 (Input Registers)	STRING(6)	sVarFromSPK
Переменные, которые записывает Master	Адрес регистра/бита	Тип данных	Переменные ПЛК
xVarToSPK	0 (Coils)	BOOL	xVarToSPK
wVarToSPK	1 (Holding Registers)	WORD	wVarToSPK
rVarToSPK	2–3 (Holding Registers)	REAL	rVarToSPK
sVarToSPK	4–6 (Holding Registers)	STRING(6)	sVarToSPK

Проекты примера доступны для скачивания: [Example SpkModbusRtuSlave.zip](#)

## 4.2 Настройка СПК (slave)

Для настройки СПК (slave) следует:

1. Создать новый проект **CODESYS V3.5** для **СПК1xx [M01]** с программой **PLC\_PRG** на языке **CFC**.
2. Добавить в проект **объединение** с именем **Real\_Word**:

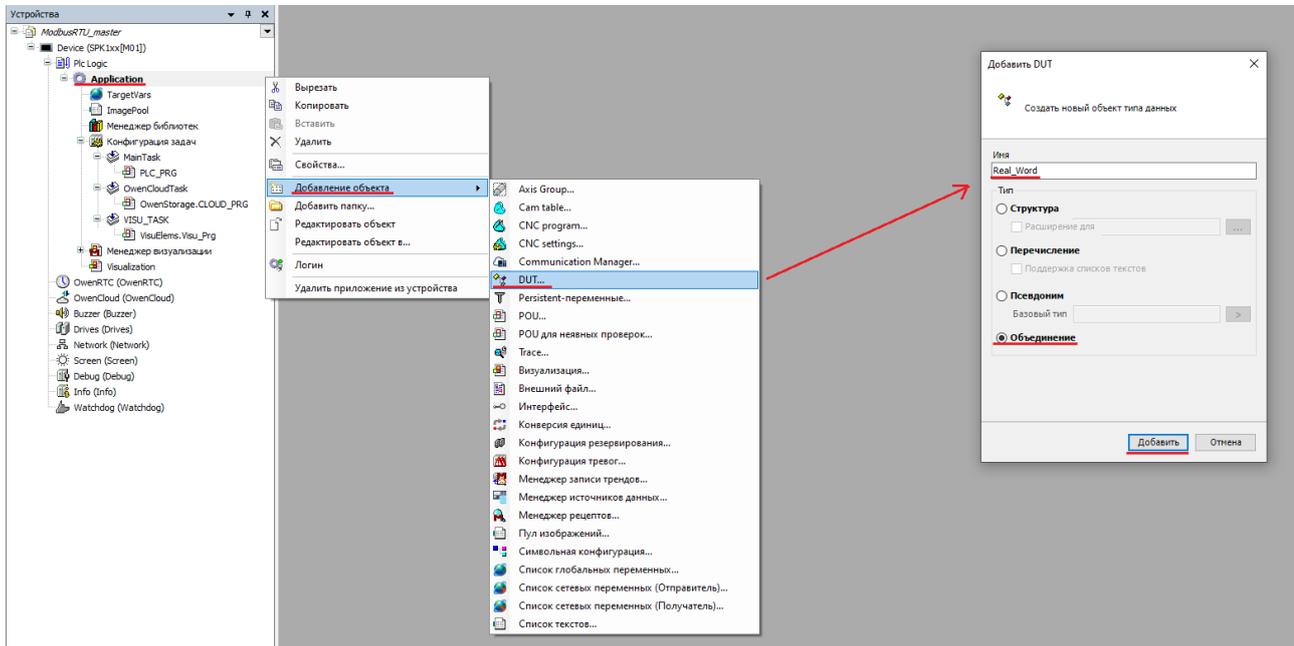


Рисунок 4.1 – Добавление в проект объединения

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

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

Рисунок 4.2 – Объявление переменных объединения

3. Добавить в проект **объединение** с именем **String\_Word**.

В объединении следует объявить переменную **sStringValue** типа **STRING** и массив **awModbusString** типа **WORD**, содержащий три элемента (**STRING** сможет содержать до 6 символов, поскольку каждый **WORD** может содержать два символа):

```
String_Word x
1  TYPE String_Word :
2  UNION
3      sStringValue: STRING(6);
4      awModbusString: ARRAY [0..2] OF WORD;
5  END_UNION
6  END_TYPE
```

Рисунок 4.3 – Объявление переменных объединения

4. Объявить в программе **PLC\_PRG** 8 переменных – 4 из них будут считываться ПЛК, 4 – записываться ПЛК. **Следует обратить внимание** на [п. 2.3](#).

```

1  PROGRAM PLC_PRG
2  VAR
3      (*данные, считываемые ПЛК*)
4      xVarFromSPK:   BOOL;
5      wVarFromSPK:   WORD;
6      rVarFromSPK:   Real_Word;
7      sVarFromSPK:   String_Word;
8
9      (*данные, записываемые ПЛК*)
10     xVarToSPK:      BOOL;
11     wVarToSPK:      WORD;
12     rVarToSPK:      Real_Word;
13     sVarToSPK:      String_Word;
14 END_VAR
    
```

Рисунок 4.4 – Объявление переменных программы

5. Код программы будет выглядеть следующим образом:

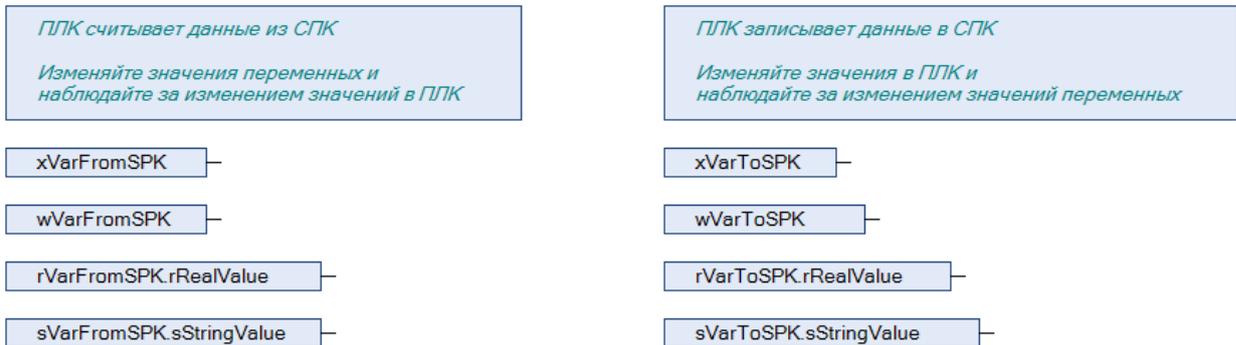


Рисунок 4.5 – Код программы на языке CFC

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

#### 6. Добавить в проект устройство **Modbus COM**.



#### ПРИМЕЧАНИЕ

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – не может.

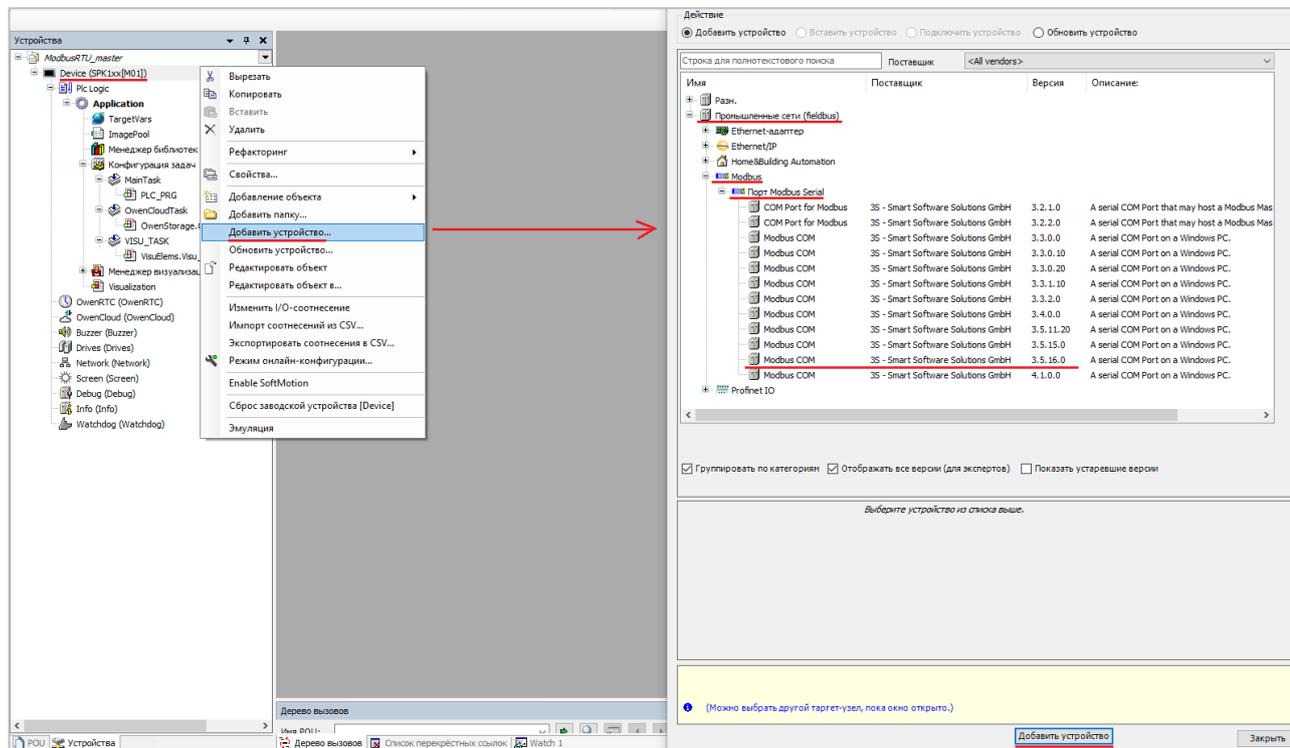


Рисунок 4.6 – Добавление устройства Modbus COM

В конфигурации COM-порта следует указать сетевые настройки в соответствии с [таблицей 4.1](#) и номер порта. Нумерация COM-портов приведена в документе **CODESYS V3.5. FAQ**.

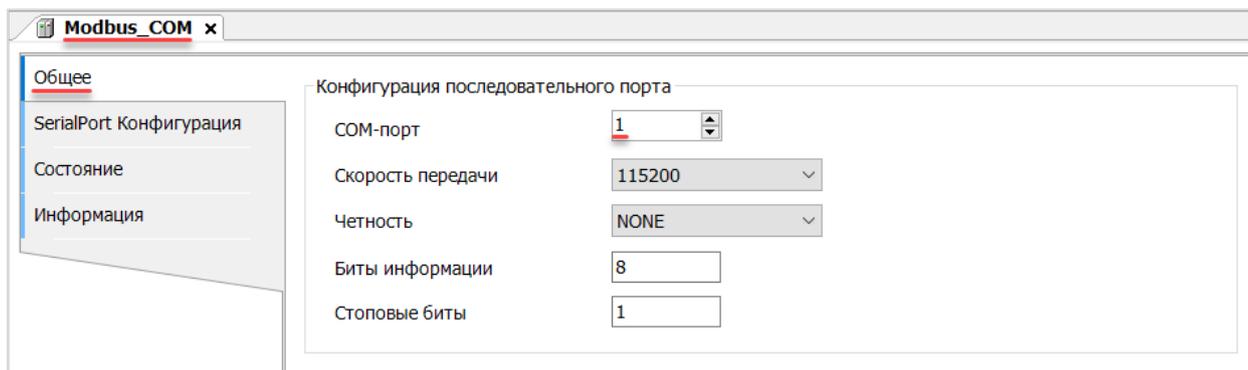


Рисунок 4.7 – Настройки COM-порта COM1

#### 7. В COM-порт добавить компонент **Modbus Serial Device**.



#### ПРИМЕЧАНИЕ

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – не может.

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

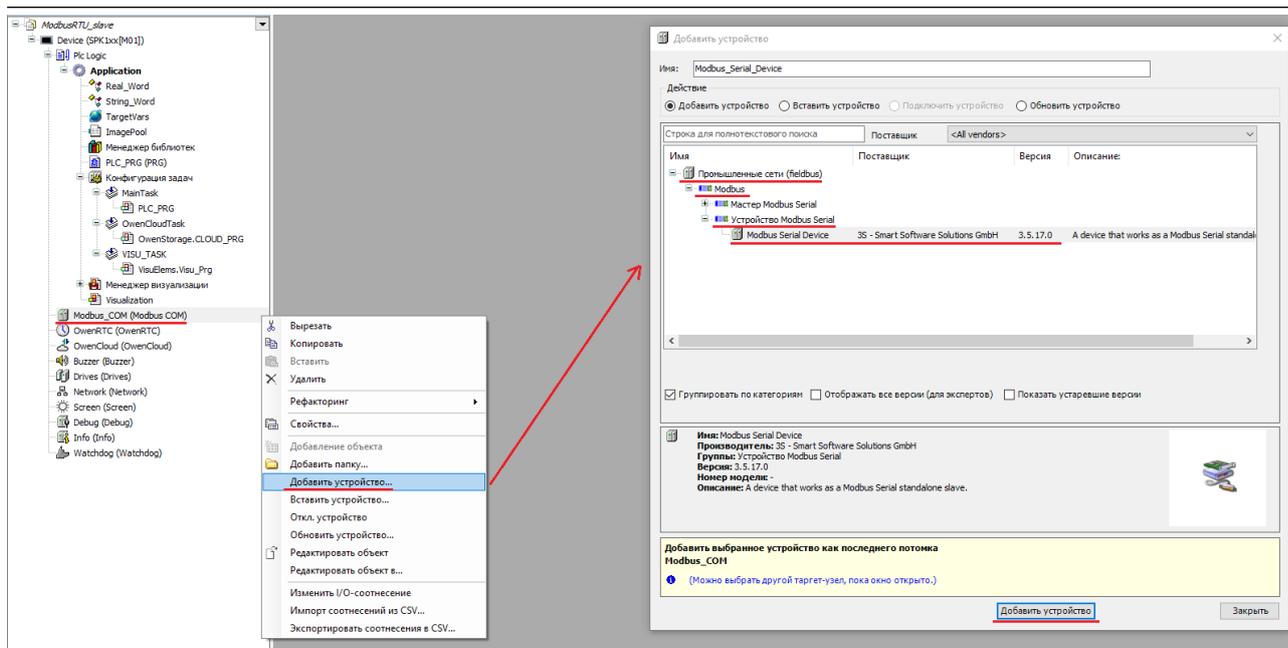


Рисунок 4.8 – Добавление компонента Modbus Serial Device

В настройках компонента на вкладке **Modbus Serial Device** следует указать адрес slave-устройства (1 в соответствии с [таблицей 4.1](#)).

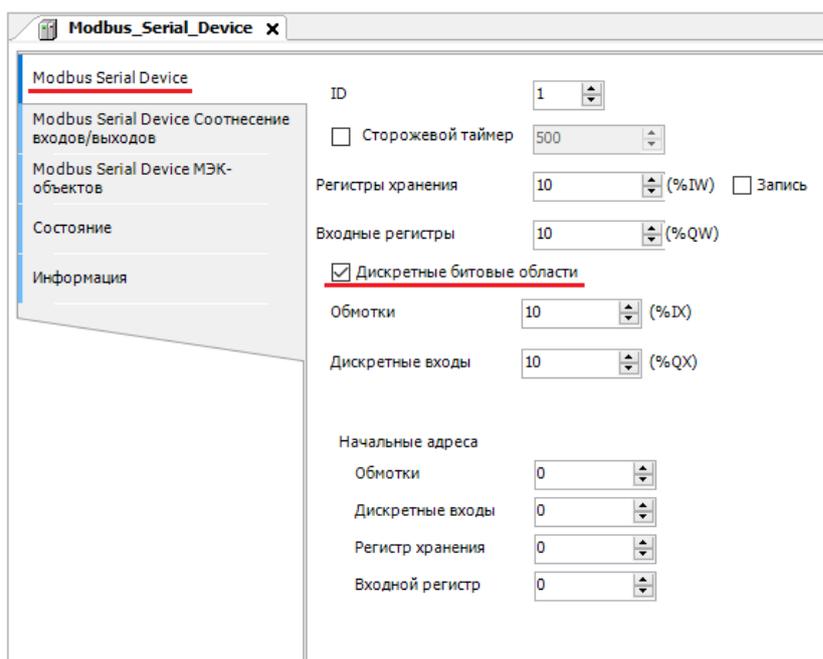


Рисунок 4.9 – Настройки компонента Modbus Serial Device

На вкладке **Modbus Serial Device** **Соотнесение входов/выходов** следует привязать к регистрам переменные программы в соответствии с [таблицей 4.2](#).



#### ПРИМЕЧАНИЕ

Следует обратить внимание на порядок WORD для переменных типа REAL.

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

Переменная	Соотнесение	Канал	Адрес	Тип	Единица
		Регистры временного хранения	%IW0	ARRAY [0..9] OF WORD	
		Регистры временного хранения[0]	%IW0	WORD	
Application.PLC_PRG.wVarToSPK		Регистры временного хранения[1]	%IW1	WORD	
Application.PLC_PRG.rVarToSPK.awModbusReal[1]		Регистры временного хранения[2]	%IW2	WORD	
Application.PLC_PRG.rVarToSPK.awModbusReal[0]		Регистры временного хранения[3]	%IW3	WORD	
Application.PLC_PRG.sVarToSPK.awModbusString[0]		Регистры временного хранения[4]	%IW4	WORD	
Application.PLC_PRG.sVarToSPK.awModbusString[1]		Регистры временного хранения[5]	%IW5	WORD	
Application.PLC_PRG.sVarToSPK.awModbusString[2]		Регистры временного хранения[6]	%IW6	WORD	
		Регистры временного хранения[7]	%IW7	WORD	
		Регистры временного хранения[8]	%IW8	WORD	
		Регистры временного хранения[9]	%IW9	WORD	
		Входные регистры	%QW0	ARRAY [0..9] OF WORD	
		Входные регистры[0]	%QW0	WORD	
Application.PLC_PRG.wVarFromSPK		Входные регистры[1]	%QW1	WORD	
Application.PLC_PRG.rVarFromSPK.awModbusReal[1]		Входные регистры[2]	%QW2	WORD	
Application.PLC_PRG.rVarFromSPK.awModbusReal[0]		Входные регистры[3]	%QW3	WORD	
Application.PLC_PRG.sVarFromSPK.awModbusString[0]		Входные регистры[4]	%QW4	WORD	
Application.PLC_PRG.sVarFromSPK.awModbusString[1]		Входные регистры[5]	%QW5	WORD	
Application.PLC_PRG.sVarFromSPK.awModbusString[2]		Входные регистры[6]	%QW6	WORD	
		Входные регистры[7]	%QW7	WORD	
		Входные регистры[8]	%QW8	WORD	
		Входные регистры[9]	%QW9	WORD	

Рисунок 4.10 – Привязка переменных к регистрам slave-устройства (holding и input регистры)

		Обмотки	%IB20	ARRAY [0..1] OF BYTE	
		Обмотки[0]	%IB20	BYTE	
Application.PLC_PRG.xVarToSPK		Bit0	%IX20.0	BOOL	
		Bit1	%IX20.1	BOOL	
		Bit2	%IX20.2	BOOL	
		Bit3	%IX20.3	BOOL	
		Bit4	%IX20.4	BOOL	
		Bit5	%IX20.5	BOOL	
		Bit6	%IX20.6	BOOL	
		Bit7	%IX20.7	BOOL	
		Обмотки[1]	%IB21	BYTE	
		Дискретные входы	%QB20	ARRAY [0..1] OF BYTE	
		Дискретные входы[0]	%QB20	BYTE	
Application.PLC_PRG.xVarFromSPK		Bit0	%QX20.0	BOOL	
		Bit1	%QX20.1	BOOL	
		Bit2	%QX20.2	BOOL	
		Bit3	%QX20.3	BOOL	
		Bit4	%QX20.4	BOOL	
		Bit5	%QX20.5	BOOL	
		Bit6	%QX20.6	BOOL	Всегда обновлять переменные Вкл. 2 (всегда в задаче цикла шины)
		Bit7	%QX20.7	BOOL	
		Дискретные входы[1]	%QB21	BYTE	

Рисунок 4.11 – Привязка переменных к регистрам slave-устройства (coils и discrete inputs)

У параметра **Всегда обновлять переменные** должно быть выставлено значение **Включено 2**.

Настройка СПК (slave) завершена.

### 4.3 Настройка ПЛК (master)

Для настройки ПЛК (master) следует:

1. Создать новый проект **CoDeSys V2.3** для ПЛК110 с программой **PLC\_PRG** на языке **CFC**. Нажать **ПКМ** на название контроллера (в данном примере – **PLC110\_30**) и добавить подэлемент **Modbus (Master)**:

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

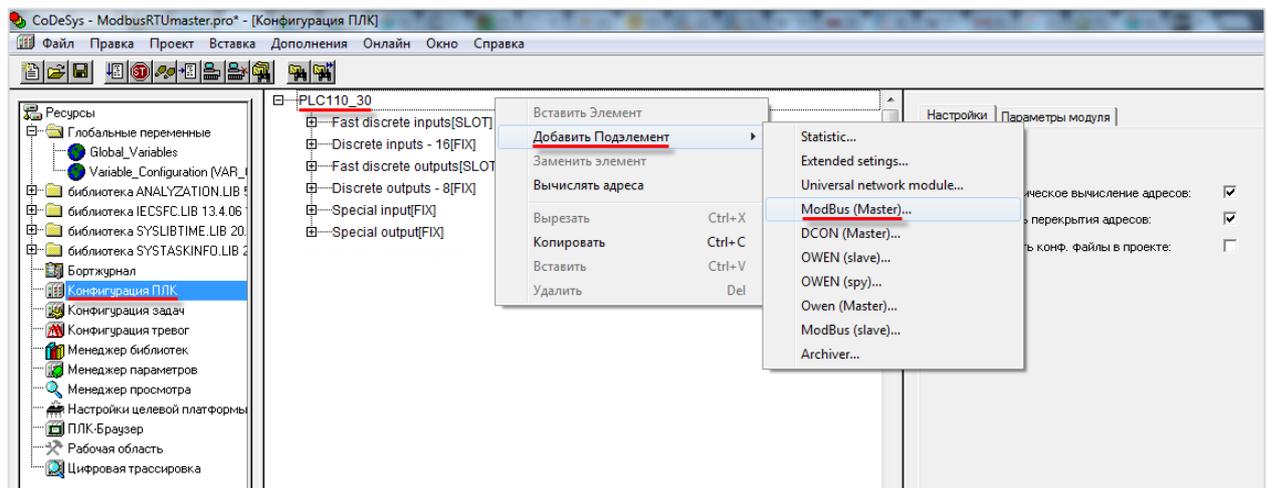


Рисунок 4.12 – Конфигурация ПЛК. Добавление Modbus (Master)

Данный элемент не нуждается в настройках.

2. Выбрать порт ПЛК, который будет использоваться для связи с СПК. Для этого в элементе **Modbus (Master)** следует нажать ПКМ на порт **Debug RS-232** и в контекстном меню выбрать команду **Заменить элемент**. В данном примере используется порт **RS-485-1**.

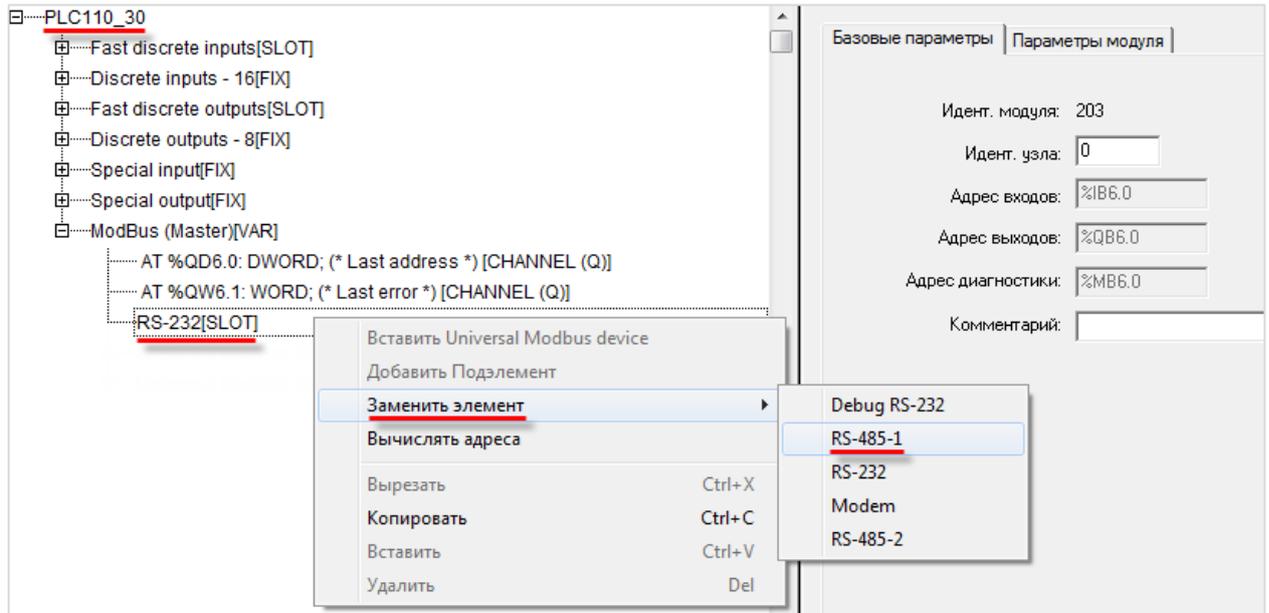


Рисунок 4.13 – Конфигурация ПЛК. Выбор порта

Настройки порта по умолчанию соответствуют тем настройкам, которые были заданы СПК (согласно [таблице 4.1](#)): скорость – **115200**, бит данных – **8**, стоп бит – **1**, контроль четности – **нет**. Для параметра **Frame Oriented** следует выбрать значение **RTU**.

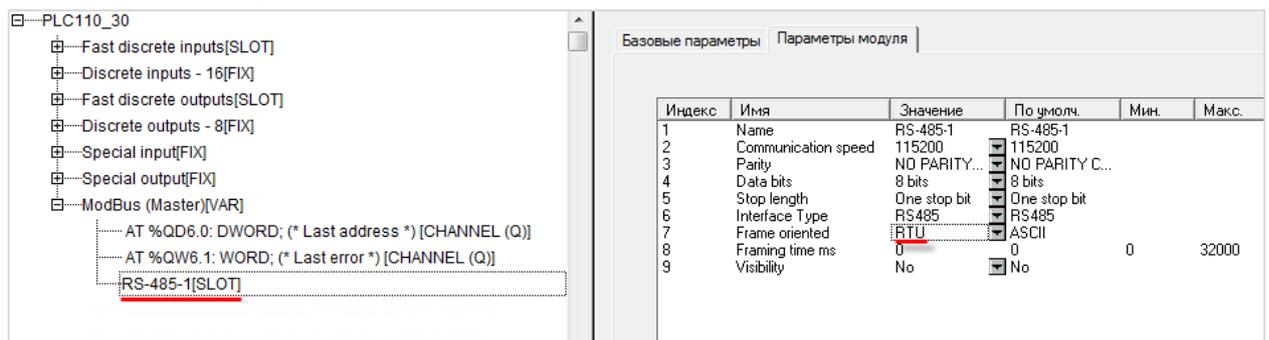


Рисунок 4.14 – Конфигурация ПЛК. Настройки порта

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

3. Нажать **ПКМ** на элемент **Modbus (Master)** и добавить два подэлемента **Universal Modbus Device**:

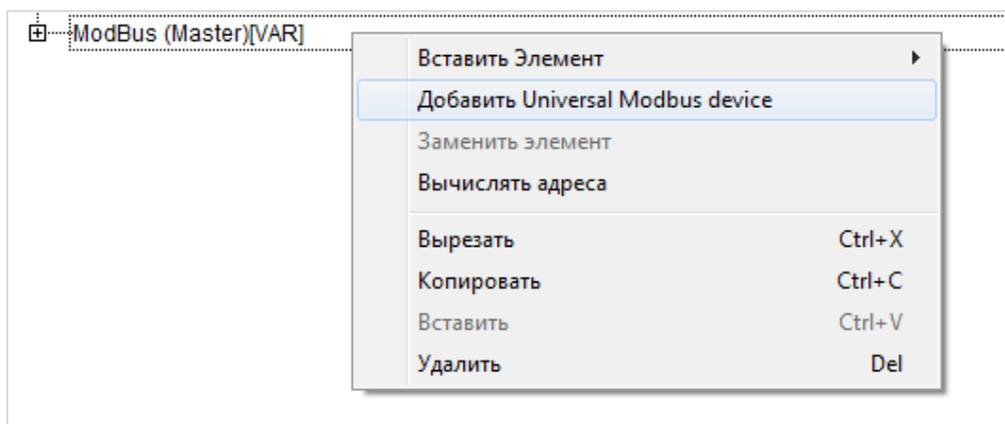


Рисунок 4.15 – Конфигурация ПЛК. Добавление Universal Modbus Device

Один из подэлементов будет использоваться для чтения значений из СПК, второй – для записи.

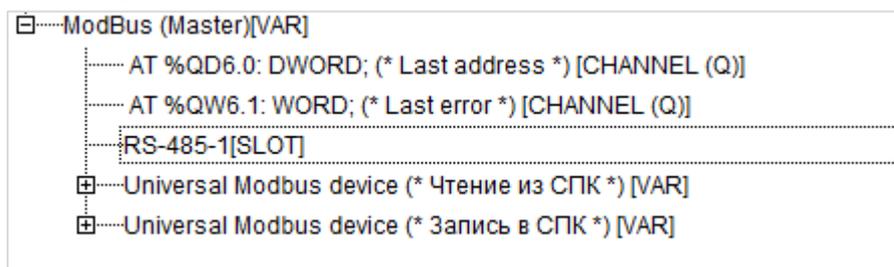


Рисунок 4.16 – Внешний вид Конфигурации ПЛК после добавления двух Universal Modbus Device

В настройках элемента **Чтение из СПК** следует указать тип связи (**Serial**), **Slave ID** СПК (в соответствии с [таблице 4.1](#) он равен 1) и режим опроса (**By poll time**, т. е. циклически).

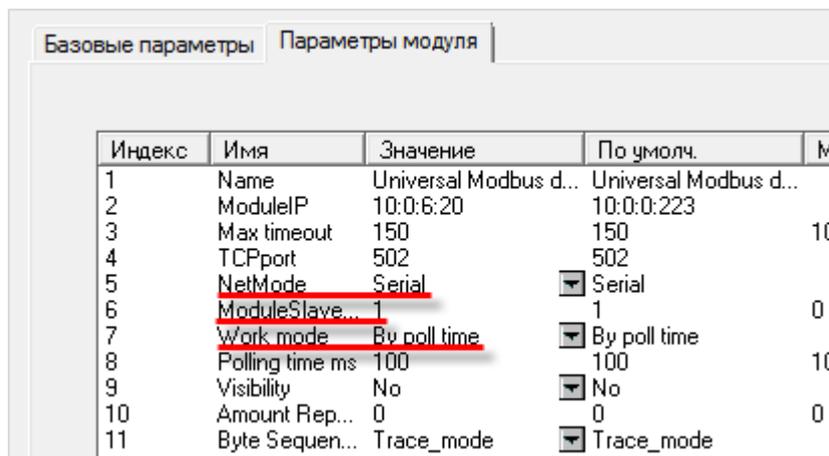


Рисунок 4.17 – Настройки Universal Modbus Device (Чтение из СПК)

В настройках элемента **Запись в СПК** следует указать тип связи (**Serial**), **Slave ID** СПК (в соответствии с [таблицей 4.1](#) он равен 1) и режим опроса (**By value change**, т. е. спорадически).

Базовые параметры		Параметры модуля		
Индекс	Имя	Значение	По умолч.	Мин
1	Name	Universal Modbus d...	Universal Modbus d...	
2	ModuleIP	10:0:6:20	10:0:0:223	
3	Max timeout	150	150	10
4	TCPport	502	502	
5	NetMode	Serial	Serial	
6	ModuleSlave...	1	1	0
7	Work mode	By value change	By poll time	
8	Polling time ms	100	100	10
9	Visibility	No	No	
10	Amount Rep...	3	0	0
11	Byte Sequen...	Trace_mode	Trace_mode	

Рисунок 4.18 – Настройки Universal Modbus Device (Запись в СПК)

Нажать ПКМ на элемент **Universal Modbus Device (Чтение из СПК)** и добавить в него подэлементы **Register Input Module**, **Real Input Module** и **String Input Module**.

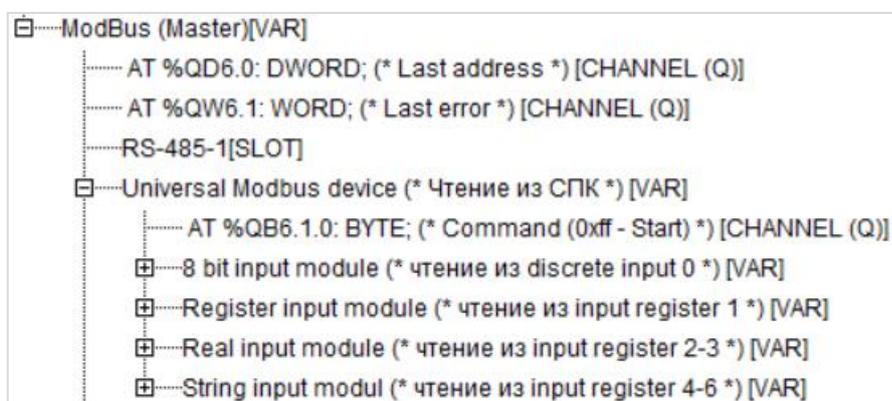


Рисунок 4.19 – Universal Modbus Device (Чтение из СПК) с добавленными Input модулями

Затем следует привязать к каждому из каналов переменную (после ввода ее имени она автоматически будет добавлена в список глобальных переменных проекта). Для ввода имени переменной следует два раза нажать на **AT**.

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

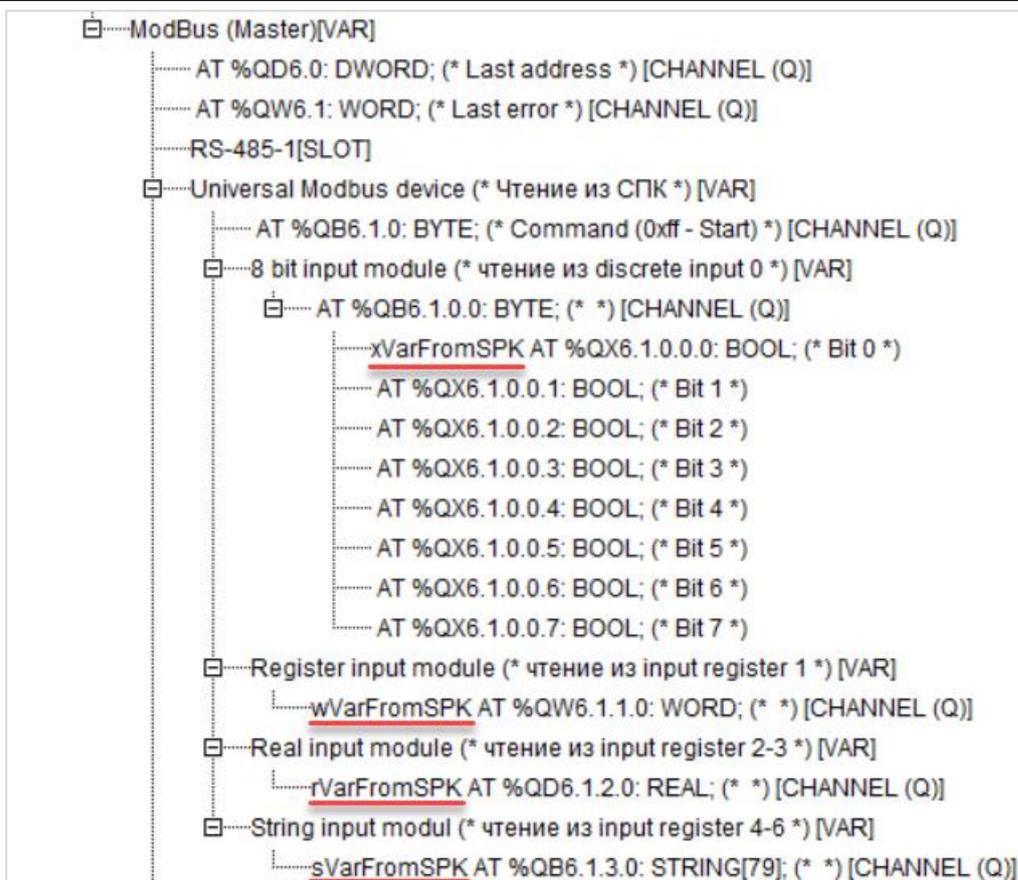


Рисунок 4.20 – Привязка переменных к каналам

Настройки модулей (используемые регистры СПК в соответствии с [таблицей 4.1](#) и функции Modbus) приведены ниже.



#### ПРИМЕЧАНИЕ

Во время работы с переменными, занимающими несколько регистров СПК (тип **REAL** и **STRING**), указывается только первый из группы регистров.

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	8 bit input module	8 bit input module
2	Register address	0	0
3	Command	Read inputs status (0x02)	Read inputs status (0x02)
8	Visibility	No	No

Рисунок 4.21 – Параметры 8 bit Input Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	Register input module	Register input module
2	Register Address	1	0
3	Command	Read input registers (0x04)	Read holding Registers (0x03)
8	Visibility	No	No

Рисунок 4.22 – Параметры Register Input Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	float input module	float input module
2	Register Address	2	0
3	Command	Read input registers (0x04)	Read holding Registers (0x03)
8	Visibility	No	No

Рисунок 4.23 – Параметры Real Input Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	String input module	String input module
2	Command	Read input registers (0x04)	Read bytes (0x70)
3	Register address	4	0
4	Amount byte	6	80
8	Visibility	No	No

Рисунок 4.24 – Параметры String Input Module

Затем следует нажать **ПКМ** на элемент **Universal Modbus Device (Запись в СПК)** и добавить в него подэлементы **Register Output Module**, **Real Output Module** и **String Output Module**.

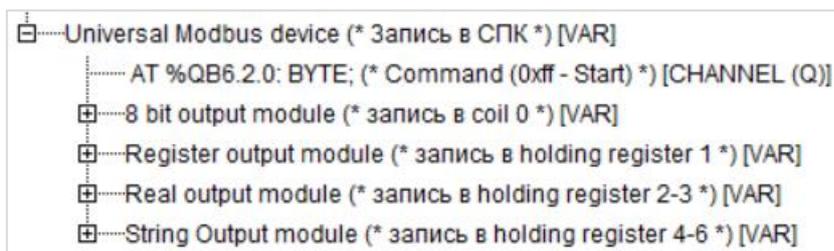


Рисунок 4.25 – Universal Modbus Device (Запись в СПК) с добавленными Output модулями

Следует привязать к каждому из каналов переменную (после ввода ее имени она автоматически будет создана в проекте как глобальная). Для ввода имени переменной следует два раза нажать на **АТ**.

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

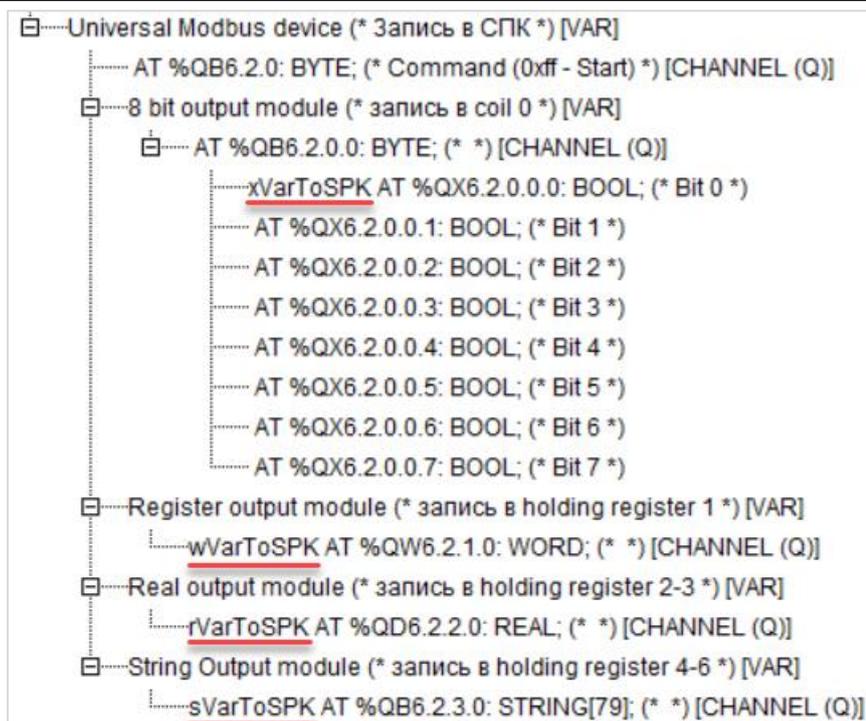


Рисунок 4.26 – Привязка переменных к каналам

Настройки модулей (используемые регистры СПК в соответствии с [таблицей 4.1](#) и функции Modbus) приведены ниже.

Во время работы с переменными, занимающими несколько регистров СПК (тип **REAL** и **STRING**), указывается только первый из группы регистров.

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	8 bit output module	8 bit output module
2	Register Address	0	0
3	Command	Force multiply coils (0x0f)	Force multiply coils (0x0f)
8	Visibility	No	No

Рисунок 4.27 – Параметры 8 Bit Output Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	Register	Register
2	Register Address	1	0
3	Command	Preset singl register (0x06)	Preset singl register (0x06)
8	Visibility	No	No

Рисунок 4.28 – Параметры Register Output Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	float output module	float output module
2	Register Address	2	0
3	Command	Preset multiple Registers (0x10)	Preset multiple Registers (0x10)
8	Visibility	No	No

Рисунок 4.29 – Параметры Real Output Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	String output module	String output module
2	Command	Preset multiple Registers (0x10)	Preset singl register (0x06)
3	Register address	4	0
4	Amounth byte	6	80
8	Visibility	No	No

Рисунок 4.30 – Параметры String Output Module

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

5. Программа **PLC\_PRG** будет выглядеть следующим образом:

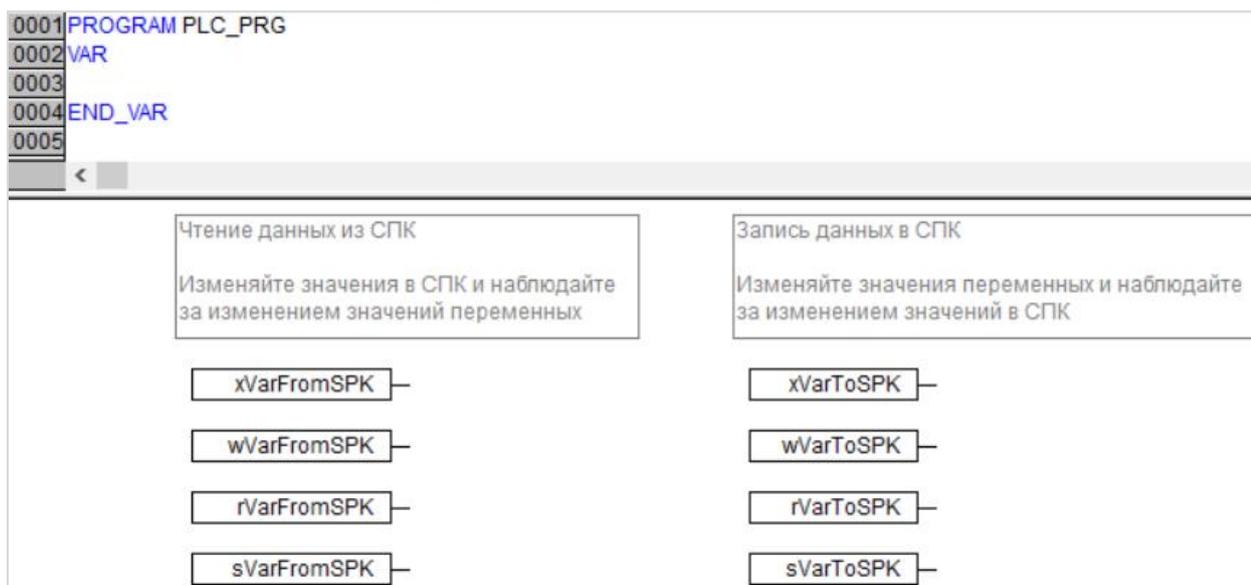


Рисунок 4.31 – Код программы **PLC\_PRG**

Настройка **ПЛК (master)** завершена.



#### ПРИМЕЧАНИЕ

Проект не содержит операций и используется только для отображения и ввода значений. Пользователь должен создать программу для реализации необходимых алгоритмов.

## 4.4 Работа с примером

Проекты следует загрузить в оба устройства и запустить.

В случае изменения значения **ToSPK** переменных в ПЛК, соответствующие изменения будут отображаться в программе СПК:

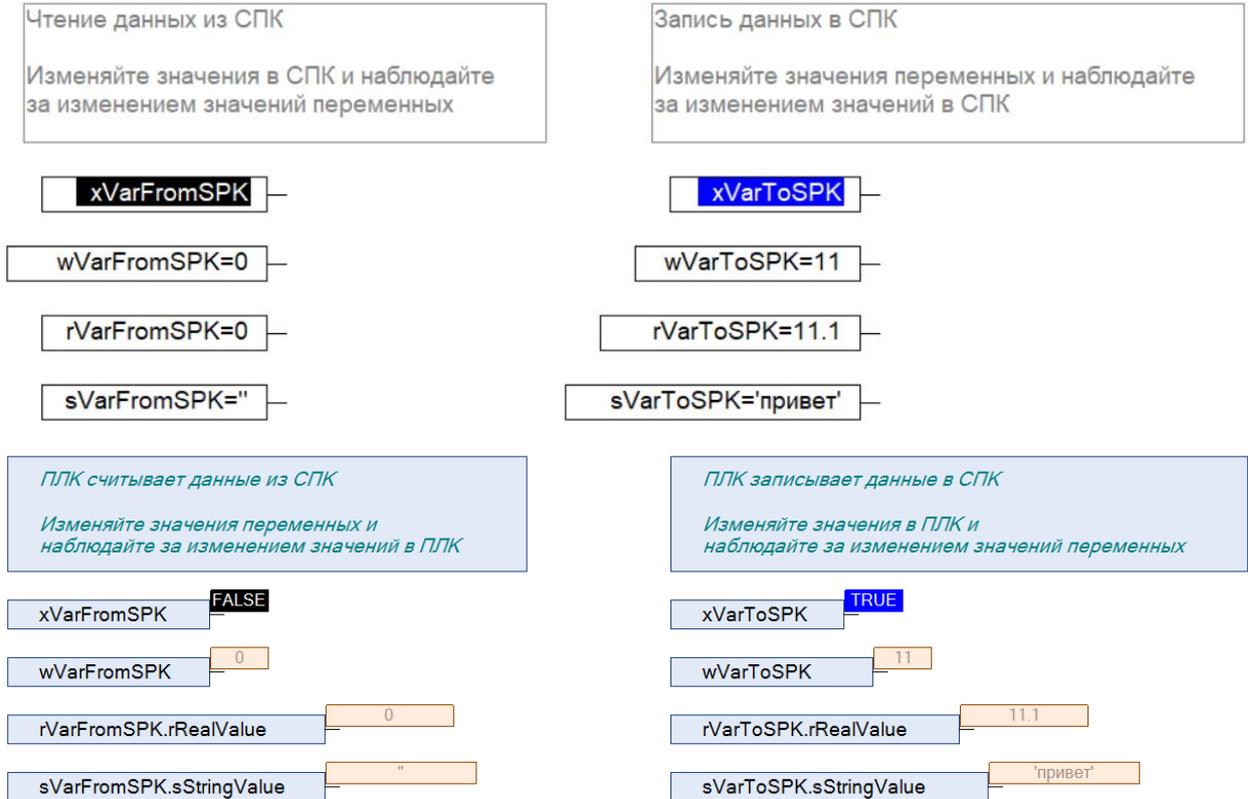


Рисунок 4.32 – ПЛК записывает данные в СПК

#### 4. Modbus RTU. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

В случае изменения значения **FromSPK** переменных СПК, соответствующие изменения будут отображаться в программе ПЛК.

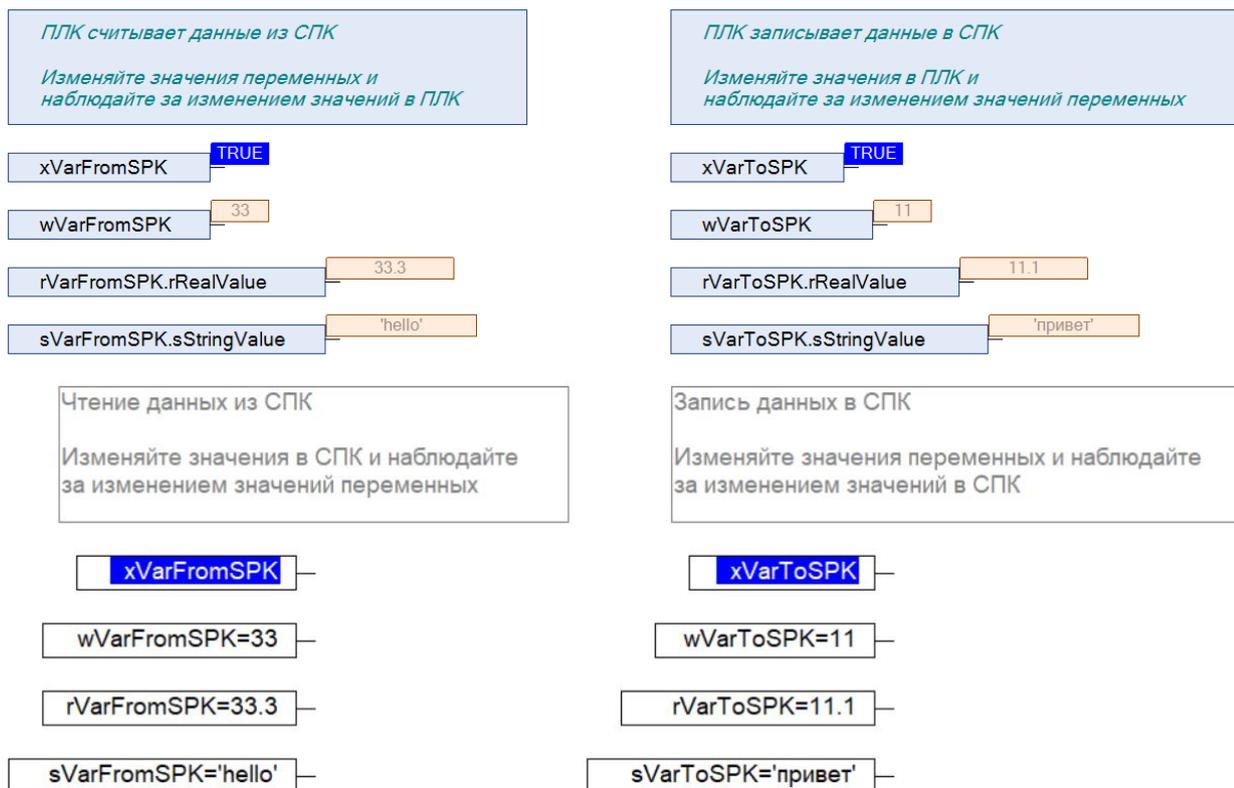


Рисунок 4.33 – ПЛК считывает данные из СПК

## 5 Modbus TCP. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

### 5.1 Описание примера

Пример посвящен настройке обмена данными между сенсорным панельным контроллером **СПК1xx [M01]** и контроллером **ПЛК110 [M02]** по протоколу **Modbus TCP**. В этом примере СПК выполняет функцию **Master**, ПЛК – **Slave**.

Основные характеристики используемых устройств приведены в таблице 5.1.

Используемые в примере переменные описаны в таблице 5.2.

**Таблица 5.1 – Характеристики устройств**

Устройство	СПК1xx [M01]	ПЛК110 [M02]
Функция	Master	Slave
IP адрес	10.2.11.20	10.2.11.10
Маска подсети	255.255.0.0	
Порт TCP	502	
Slave ID	–	1
Таргет	3.5.17.31	PLC110.30-M v2 (версия 3.3)
Версия прошивки	2.4.0923.1000	1.2.42
Среда разработки проекта	CODESYS V3.5 SP17 Patch 3	CoDeSys V2.3.9.41
Название файла проекта	ModbusTCPmaster.projectarchive	ModbusTCPslave.pro

**Таблица 5.2 – Список переменных**

СПК1xx [M01] (Master)			ПЛК110 [M02] (Slave)	
Переменные, в которые считываются значения из Slave	Переменные, значения которых записываются в Slave	Тип данных	Переменные ПЛК	Адрес регистра/бита
xVarRead	xVarWrite	BOOL	xVar	0/0
wVarRead	wVarWrite	WORD	wVar	1
rVarRead	rVarWrite	REAL	rVar	2–3
sVarRead	sVarWrite	STRING(6)	sVar	4–6

Проекты примера доступны для скачивания: [Example\\_SpkModbusTcpMaster.zip](#)

### 5.2 Настройка ПЛК (slave)

Для настройки ПЛК (slave) следует:

1. Создать новый проект **CoDeSys V2.3** для **ПЛК110** с программой **PLC\_PRG** на языке **CFC**.
2. В компоненте **Конфигурация ПЛК** (вкладка **Ресурсы**) настроить регистры Modbus и привязать к ним переменные.

Нажать **ПКМ** на название контроллера (в примере – **PLC110\_30**) и добавить подэлемент **Modbus (Slave)**:

## 5. Modbus TCP. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

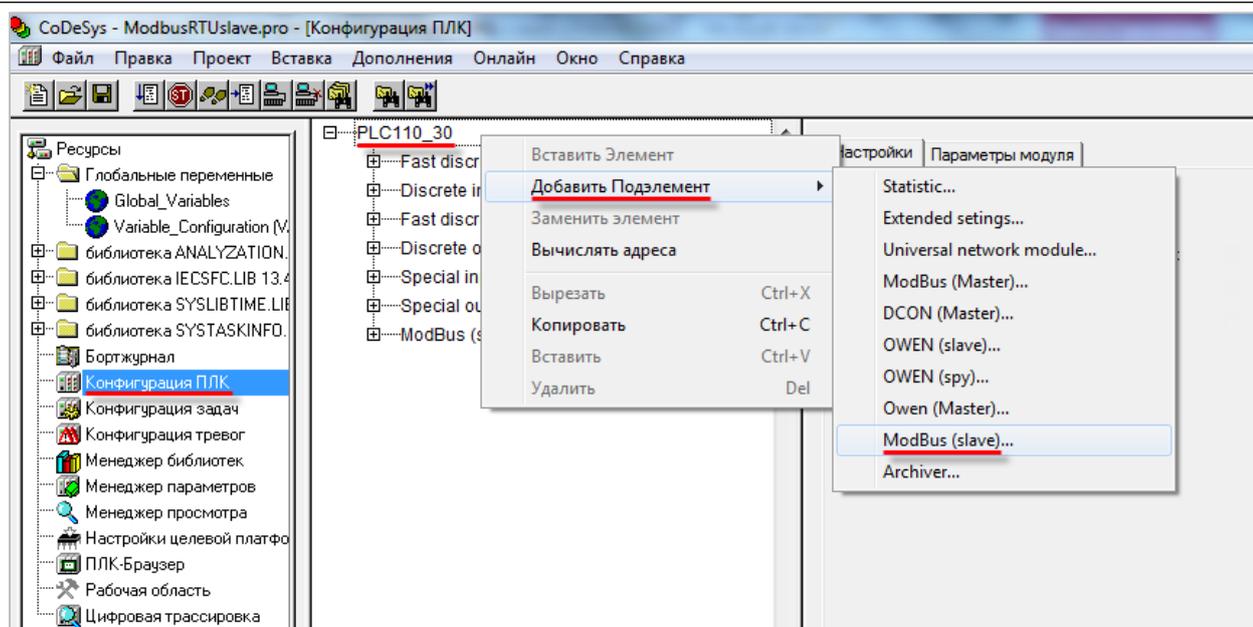
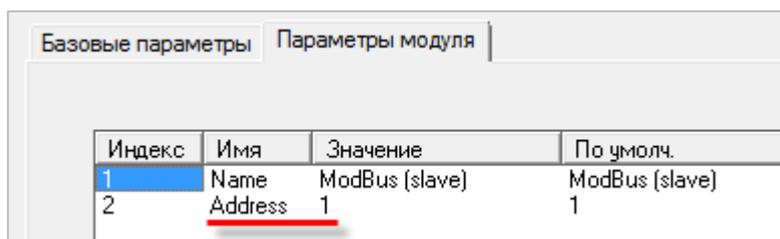


Рисунок 5.1 – Конфигурация ПЛК. Добавление Modbus (Slave)

В его настройках следует выбрать адрес (Slave ID), равный 1 (в соответствии с [таблицей 5.1](#)):



Индекс	Имя	Значение	По умолч.
1	Name	ModBus (slave)	ModBus (slave)
2	Address	1	1

Рисунок 5.2 – Конфигурация ПЛК. Настройка Modbus (Slave)

3. Выбрать порт ПЛК, который будет использоваться для связи с СПК. Для этого следует нажать ПКМ на элемент **Modbus (FIX)** и добавим подэлемент **TCP**.

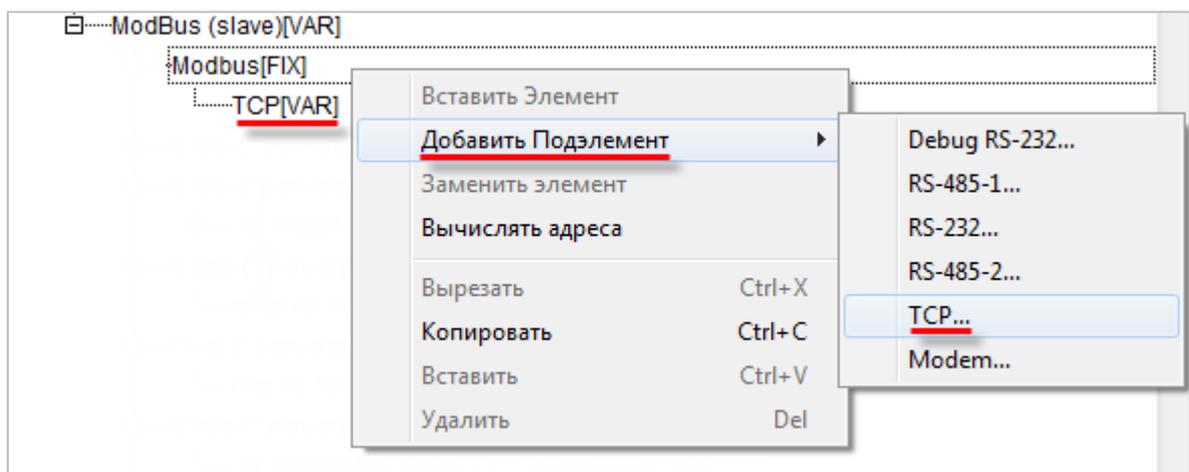


Рисунок 5.3 – Конфигурация ПЛК. Добавление подэлемента TCP

В параметрах подэлемента следует указать используемый TCP порт – в соответствии с [таблицей 5.1](#) это порт **502**.

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	TCP	TCP
2	RemotePort	502	502
3	Visibility	No	<input type="checkbox"/> No

Рисунок 5.4 – Конфигурация ПЛК. Настройки подэлемента TCP

4. Нажать **ПКМ** на элемент **Modbus (Slave)** и добавить следующие подэлементы:

- 8 bits (для BOOL);
- 8 bits (для обеспечения [выравнивания памяти](#));
- 2 byte (для WORD);
- Float (для REAL);
- 3 элемента 2 byte (для **STRING** из 6 символов).

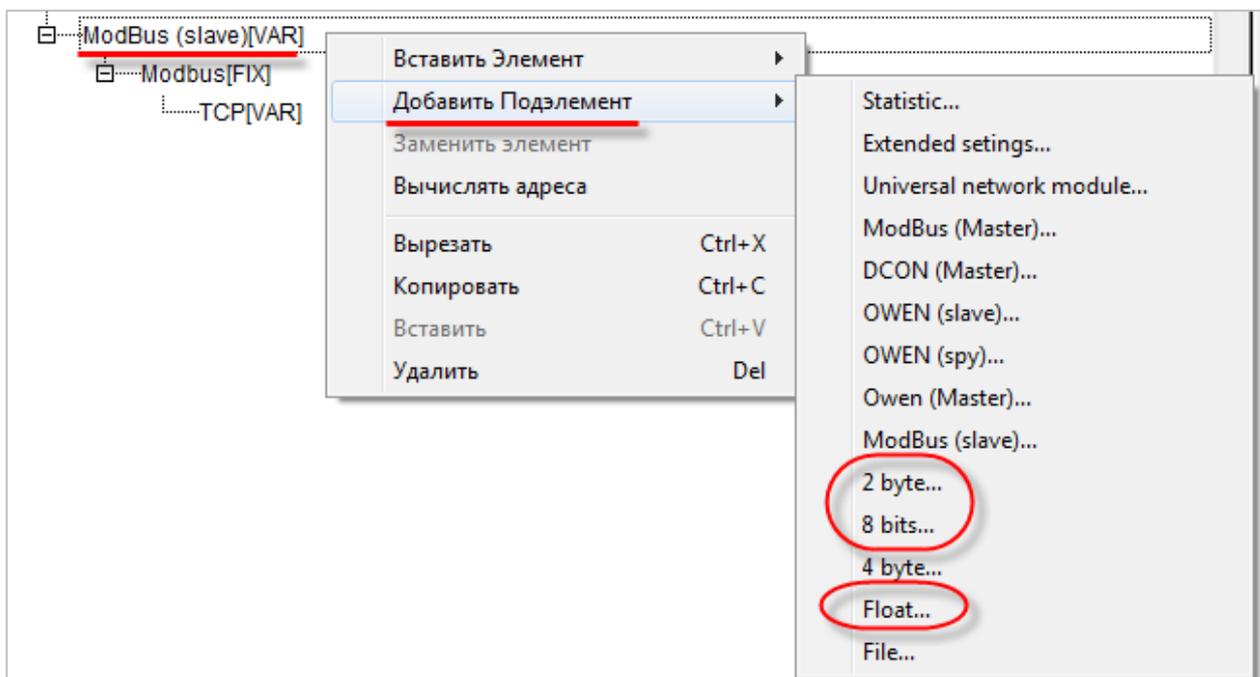


Рисунок 5.5 – Конфигурация ПЛК. Добавление подэлементов

В результате **Конфигурация ПЛК** будет выглядеть следующим образом (см. рисунок 5.6). Затем следует объявить переменные (после ввода имен они автоматически будут добавлены в список глобальных переменных проекта). Для ввода имени переменной следует два раза нажать на **AT**.



**ПРИМЕЧАНИЕ**

К регистрам 4–6 не привязывается никаких переменных. Соответствующая им переменная **sVar** будет объявлена в программе **PLC\_PRG** (в пп. 5).

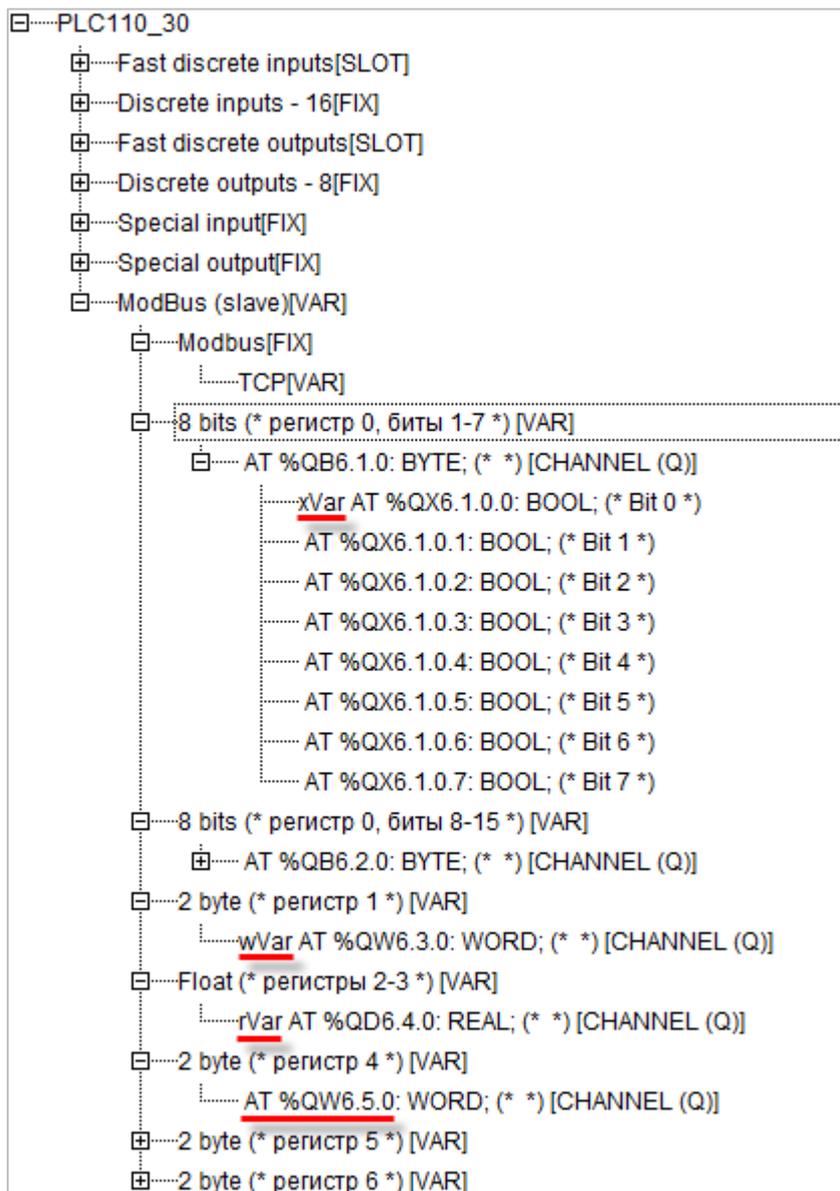


Рисунок 5.6 – Внешний вид Modbus (Slave) с добавленными подэлементами



**ПРИМЕЧАНИЕ**

Нумерация регистров в среде CODESYS всегда начинается с нуля, каждый регистр физически занимает два байта (16 бит). Поэтому переменная типа **REAL** займет два регистра (с адресами 2 и 3). Переменная типа **STRING**, которой соответствует три **2 byte** элемента, займет регистры с адресами 4–6, что следует учитывать при настройке master-устройства.

Подробнее вопросы адресации рассмотрены в **Руководстве пользователя ПЛК**.

5. Программа **PLC\_PRG** будет выглядеть следующим образом:

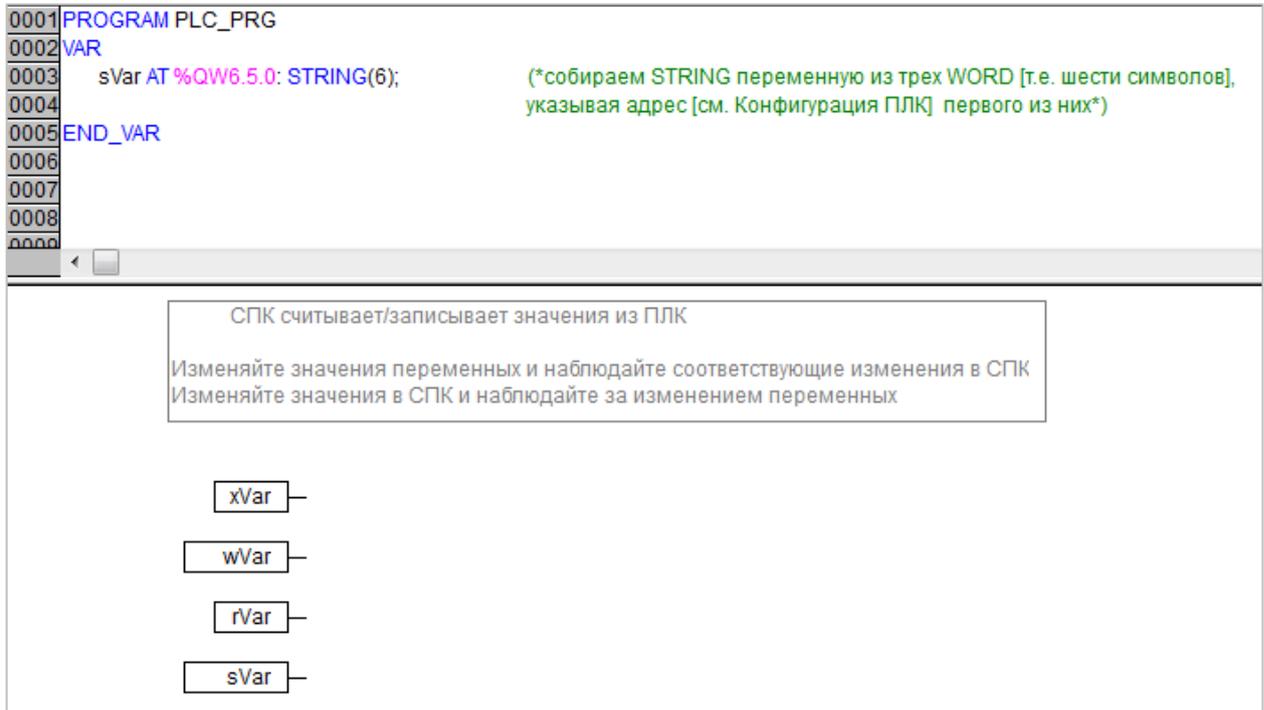


Рисунок 5.7 – Код программы **PLC\_PRG**

Настройка ПЛК (**slave**) завершена.



**ПРИМЕЧАНИЕ**

Проект не содержит операций и используется только для отображения и ввода значений. Пользователь должен создать программу для реализации необходимых алгоритмов.

### 5.3 Настройка СПК (master)

Для настройки СПК (master) следует:

1. Создать новый проект **CODESYS V3.5** для **СПК1xx [M01]** с программой **PLC\_PRG** на языке **CFC**.
2. Добавить в проект **объединение** с именем **Real\_Word**:

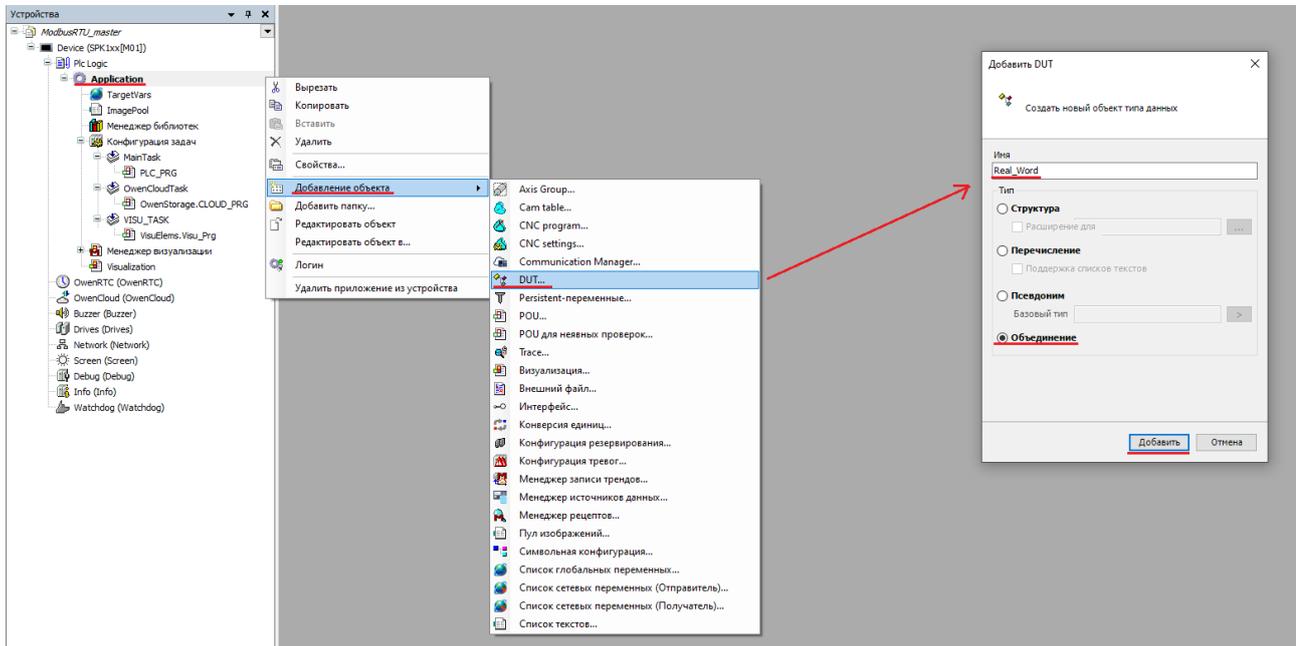


Рисунок 5.8 – Добавление в проект объединения

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

```

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

Рисунок 5.9 – Объявление переменных объединения

3. Добавить в проект **объединение** с именем **String\_Word**.

В объединении следует объявить переменную **sStringValue** типа **STRING** и массив **awModbusString** типа **WORD**, содержащий три элемента (**STRING** может содержать до 6 символов, поскольку каждый **WORD** может содержать два символа):

```

1  TYPE String_Word :
2  UNION
3      sStringValue:  STRING(6);
4      awModbusString: ARRAY [0..2] OF WORD;
5  END_UNION
6  END_TYPE
    
```

Рисунок 5.10 – Объявление переменных объединения

4. Объявить в программе **PLC\_PRG** девять переменных – 4 из них будут использоваться для отображения данных, считанных из ПЛК, еще 4 – для ввода данных, которые будут записаны в ПЛК. Последняя переменная будет являться триггером записи.

```

1  PROGRAM PLC_PRG
2  VAR
3      (*данные, из считываемые ПЛК*)
4      xVarRead:      BOOL;
5      wVarRead:      WORD;
6      rVarRead:      Real_Word;
7      sVarRead:      String_Word;
8
9      (*данные, записываемые в ПЛК*)
10     xVarWrite:     BOOL;
11     wVarWrite:     WORD;
12     rVarWrite:     Real_Word;
13     sVarWrite:     String_Word;
14
15     xTrigger:      BOOL; // триггер записи
16 END_VAR
    
```

Рисунок 5.11 – Объявление переменных программы

5. Код программы будет выглядеть следующим образом:

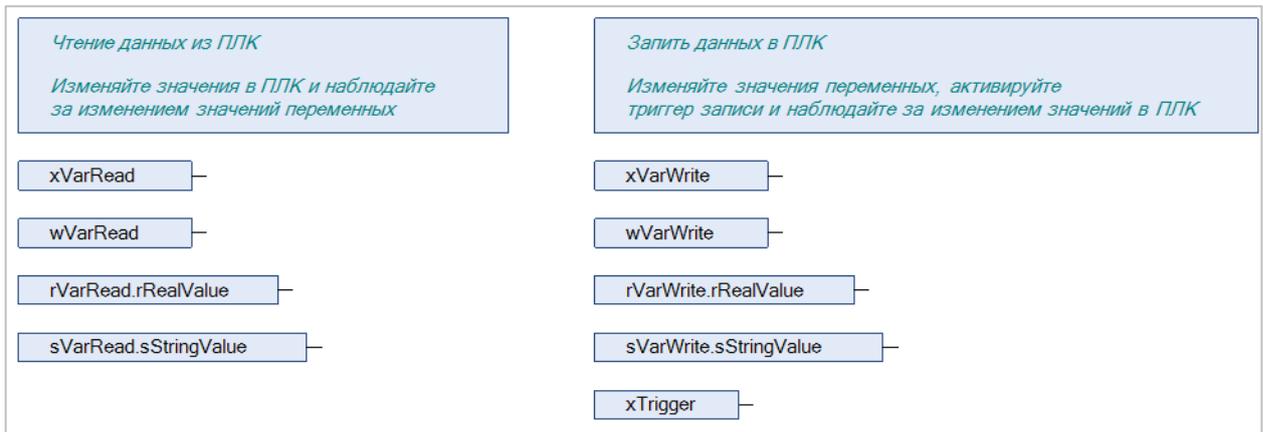


Рисунок 5.12 – Код программы на языке CFC

6. Добавить в проект компонент **Ethernet**.



**ПРИМЕЧАНИЕ**

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – может.

## 5. Modbus TCP. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

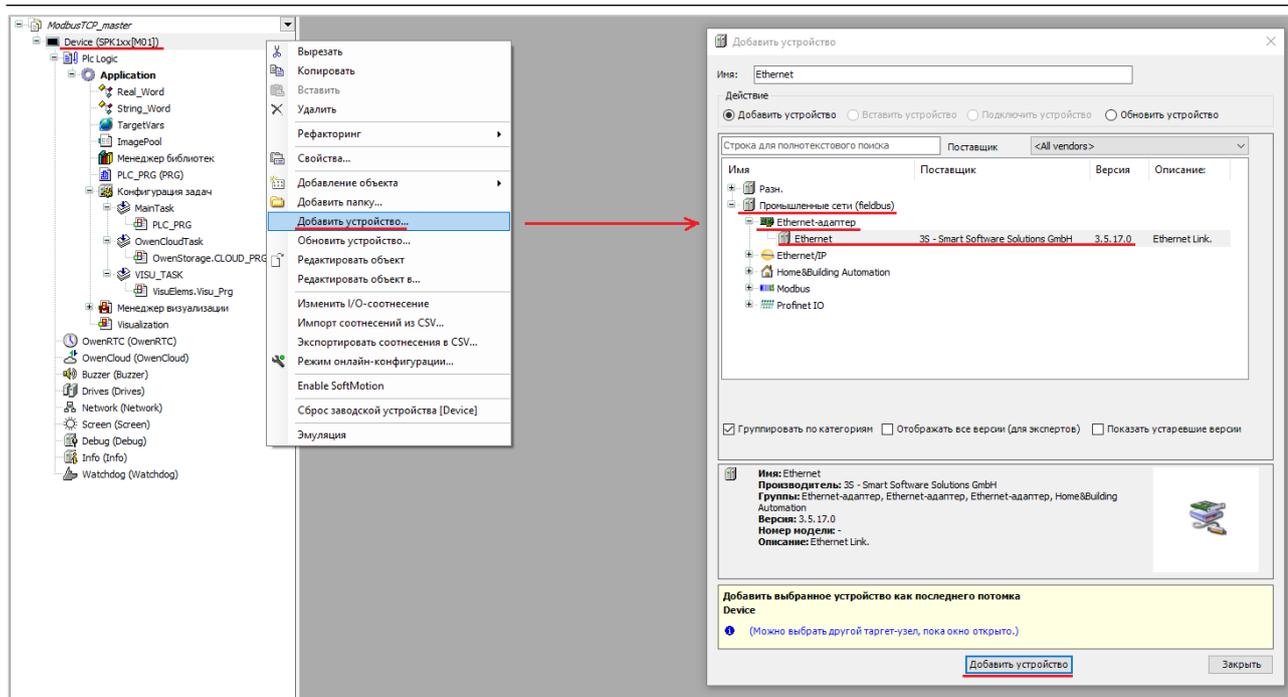


Рисунок 5.13 – Добавление компонента Ethernet

В конфигурации **Ethernet** следует указать сетевые настройки в соответствии с [таблицей 5.1](#).

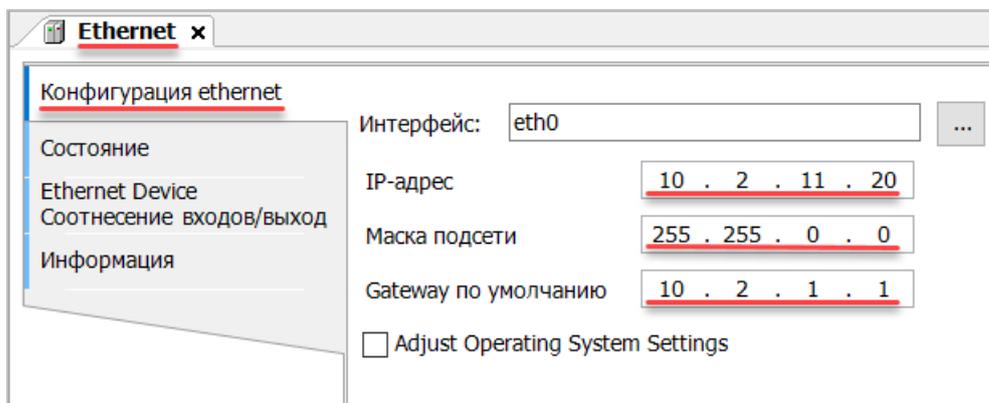


Рисунок 5.14 – Настройки компонента Ethernet

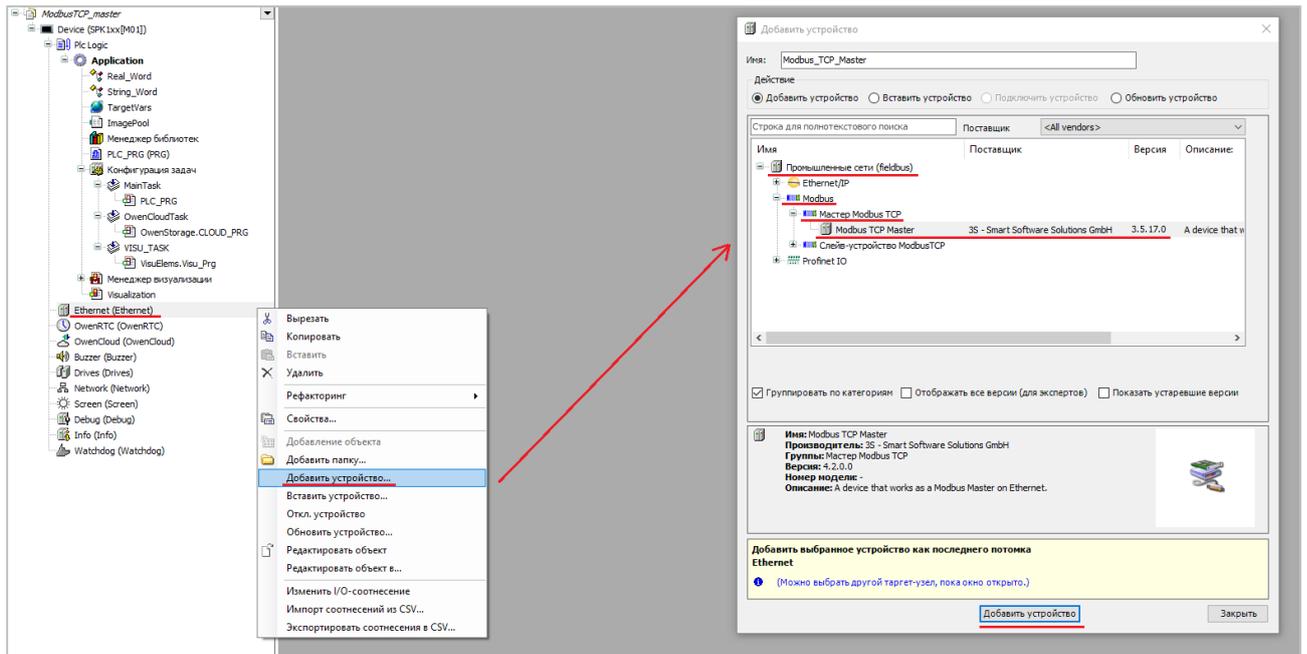
7. В компонент **Ethernet** следует добавить компонент **Modbus TCP Master**.



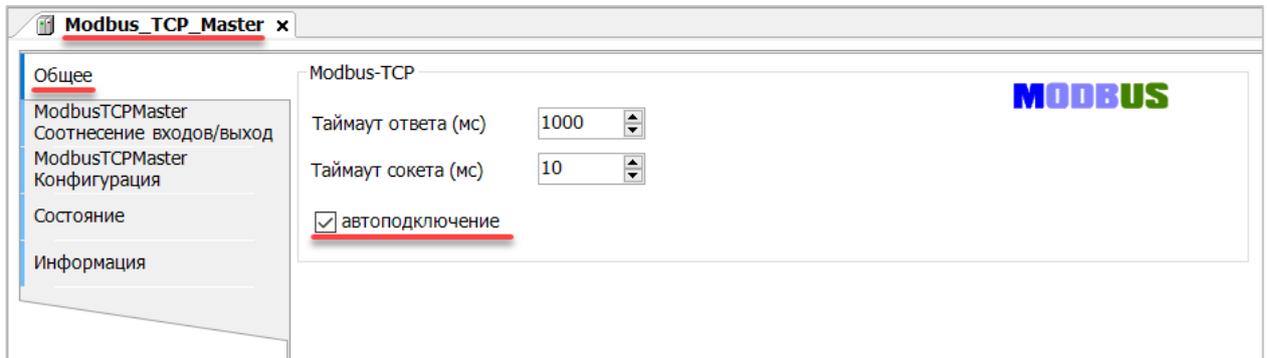
### ПРИМЕЧАНИЕ

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – может.

## 5. Modbus TCP. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave



В настройках компонента следует поставить галочку **Автоподключение**.



## 5. Modbus TCP. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

### 8. В компонент **Modbus TCP Master** следует добавить компонент **Modbus TCP Slave**.



#### ПРИМЕЧАНИЕ

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – может.

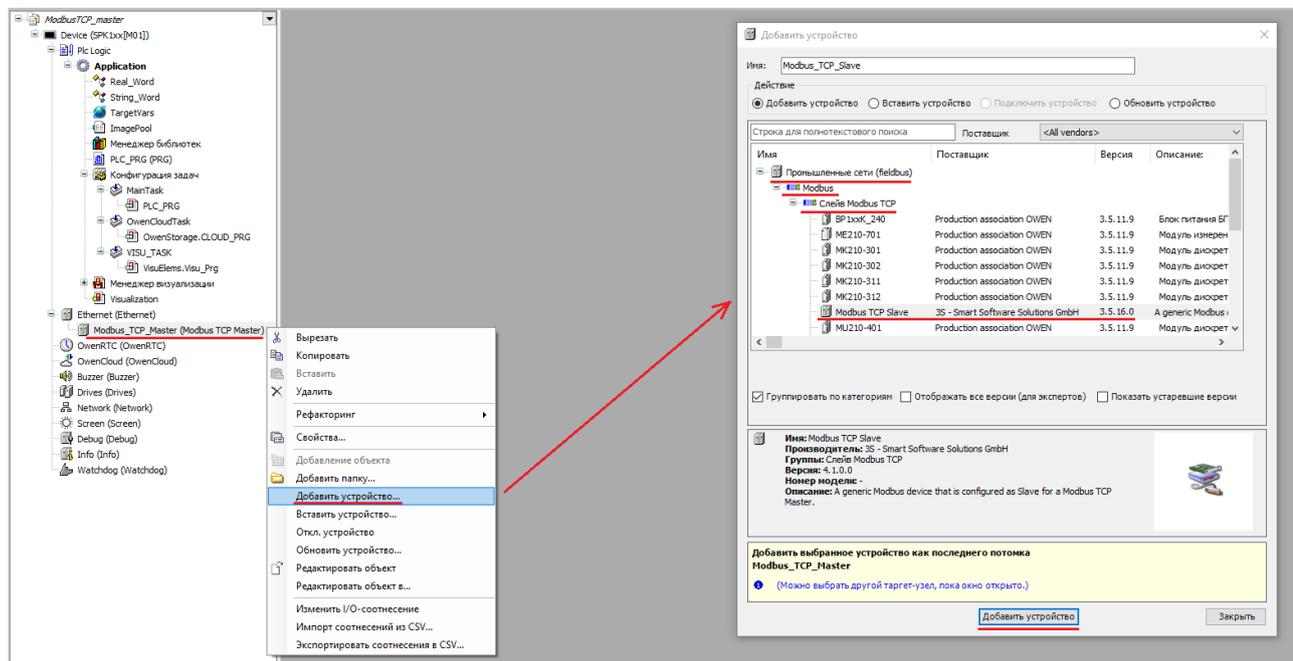


Рисунок 5.17 – Добавление компонента Modbus TCP Slave в проект

В настройках компонента на вкладке **Общее** следует указать IP-адрес, Unit ID и порт slave-устройства в соответствии с [таблицей 5.1](#).

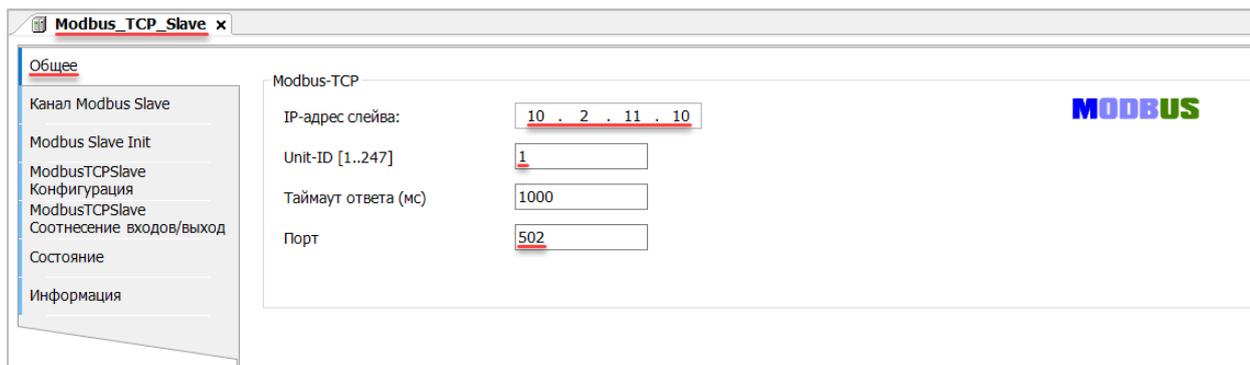


Рисунок 5.18 – Настройки компонента Modbus Slave в проект

Во вкладке **Канал Modbus Slave** следует создать 8 каналов – 4 из них будут использоваться для чтения переменных, 4 – для записи. Чтение будет осуществляться циклически, запись – по переднему фронту триггера (**RISING\_EDGE**). Используемые функции соответствуют типам данных, адреса регистров настроены согласно [таблице 5.2](#).

## 5. Modbus TCP. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

Имя	Тип доступа	Триггер	Сдвиг READ	Длина	Обработка ошибок	Сдвиг WRITE	Длина
0 Channel 0	Read Coils (Код функции 01)	Цикл., t#100ms	16#0000	1	Сохранить посл. значение		
1 Channel 1	Read Holding Registers (Код функции 03)	Цикл., t#100ms	16#0001	1	Сохранить посл. значение		
2 Channel 2	Read Holding Registers (Код функции 03)	Цикл., t#100ms	16#0002	2	Сохранить посл. значение		
3 Channel 3	Read Holding Registers (Код функции 03)	Цикл., t#100ms	16#0004	3	Сохранить посл. значение		
4 Channel 4	Write Single Coil (Код функции 05)	Передний фронт				16#0000	1
5 Channel 5	Write Single Register (Код функции 06)	Передний фронт				16#0001	1
6 Channel 6	Write Multiple Registers (Код функции 16)	Передний фронт				16#0002	2
7 Channel 7	Write Multiple Registers (Код функции 16)	Передний фронт				16#0004	3

Рисунок 5.19 – Настройка каналов Modbus Slave

Во вкладке **ModbusTCPSlave Соотнесение входов/выходов** следует привязать к каналам переменные программы в соответствии с [таблицей 5.2](#). У параметра **Всегда обновлять переменные** должно быть выставлено значение **Включено 2**.

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Channel 0	%IB0	ARRAY [0..0] OF BYTE		Read Coils
		Channel 0[0]	%IB0	BYTE		Бит 0
Application.PLC_PRG.xVarRead		Bit0	%BX0.0	BOOL		Бит 0
		Channel 1	%IW1	ARRAY [0..0] OF WORD		Read Holding Registers
Application.PLC_PRG.wVarRead		Channel 1[0]	%IW1	WORD		Регистр 1
		Channel 2	%IW2	ARRAY [0..1] OF WORD		Read Holding Registers
Application.PLC_PRG.rVarRead.awModbusRea[0]		Channel 2[0]	%IW2	WORD		Регистр 2
Application.PLC_PRG.rVarRead.awModbusRea[1]		Channel 2[1]	%IW3	WORD		Регистр 3
		Channel 3	%IW4	ARRAY [0..2] OF WORD		Read Holding Registers
Application.PLC_PRG.sVarRead.awModbusString[0]		Channel 3[0]	%IW4	WORD		Регистр 4
Application.PLC_PRG.sVarRead.awModbusString[1]		Channel 3[1]	%IW5	WORD		Регистр 5
Application.PLC_PRG.sVarRead.awModbusString[2]		Channel 3[2]	%IW6	WORD		Регистр 6
Application.PLC_PRG.xTrigger		Channel 4	%QB0.0	BIT		Trigger Variable
		Channel 4	%QB1	ARRAY [0..0] OF BYTE		Write Single Coil
Application.PLC_PRG.xVarWrite		Channel 4[0]	%QB1	BYTE		Бит 0
		Bit0	%BX1.0	BOOL		Бит 0
Application.PLC_PRG.xTrigger		Channel 5	%QW2.0	BIT		Trigger Variable
		Channel 5	%QW2	ARRAY [0..0] OF WORD		Write Single Register
Application.PLC_PRG.wVarWrite		Channel 5[0]	%QW2	WORD		Регистр 1
Application.PLC_PRG.xTrigger		Channel 6	%QW6.0	BIT		Trigger Variable
		Channel 6	%QW4	ARRAY [0..1] OF WORD		Write Multiple Registers
Application.PLC_PRG.rVarWrite.awModbusRea[0]		Channel 6[0]	%QW4	WORD		Регистр 2
Application.PLC_PRG.rVarWrite.awModbusRea[1]		Channel 6[1]	%QW5	WORD		Регистр 3
Application.PLC_PRG.xTrigger		Channel 7	%QX12.0	BIT		Trigger Variable
		Channel 7	%QW7	ARRAY [0..2] OF WORD		Write Multiple Registers
Application.PLC_PRG.sVarWrite.awModbusString[0]		Channel 7[0]	%QW7	WORD		Регистр 4
Application.PLC_PRG.sVarWrite.awModbusString[1]		Channel 7[1]	%QW8	WORD		Регистр 5
Application.PLC_PRG.sVarWrite.awModbusString[2]		Channel 7[2]	%QW9	WORD		Регистр 6

Сброс соотнесения    Всегда обновлять переменные: **Вкл. 2 (всегда в задаче цикла шины)**

МЖ-объекты

Переменная	Соотнесение	Тип
Modbus_TCP_Slave		ModbusTCPSlave

= Создать новую переменную   
 = Соотнести с существующей переменной

Рисунок 5.20 – Привязка переменных к каналам



### ПРИМЕЧАНИЕ

Проект не содержит операций и используется только для отображения и ввода значений. Пользователь должен создать программу для реализации необходимых алгоритмов.

Настройка СПК (master) завершена.

## 5.4 Работа с примером

Проекты следует загрузить в оба устройства и запустить их.

В случае изменения значения переменных в ПЛК, соответствующие изменения будут отображаться в программе СПК:

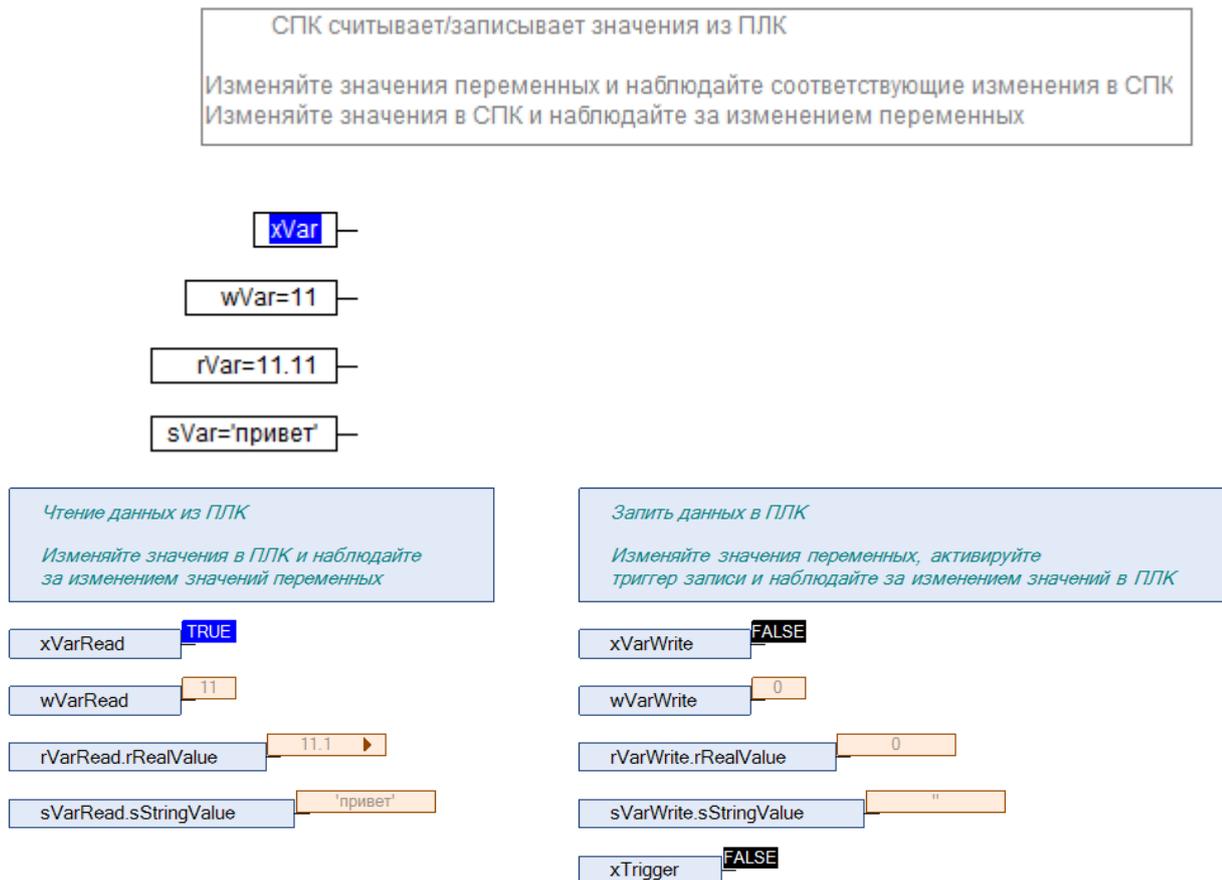


Рисунок 5.21 – СПК считывает данные из ПЛК

## 5. Modbus TCP. ПЛК (CDS V3.5) – master, ПЛК (CDS V2.3) – slave

Затем следует сохранить изменения **write** переменных СПК и активировать триггер записи. Соответствующие изменения будут отображаться в программе ПЛК. Также новые значения будут считаны в **read** переменные программы СПК.

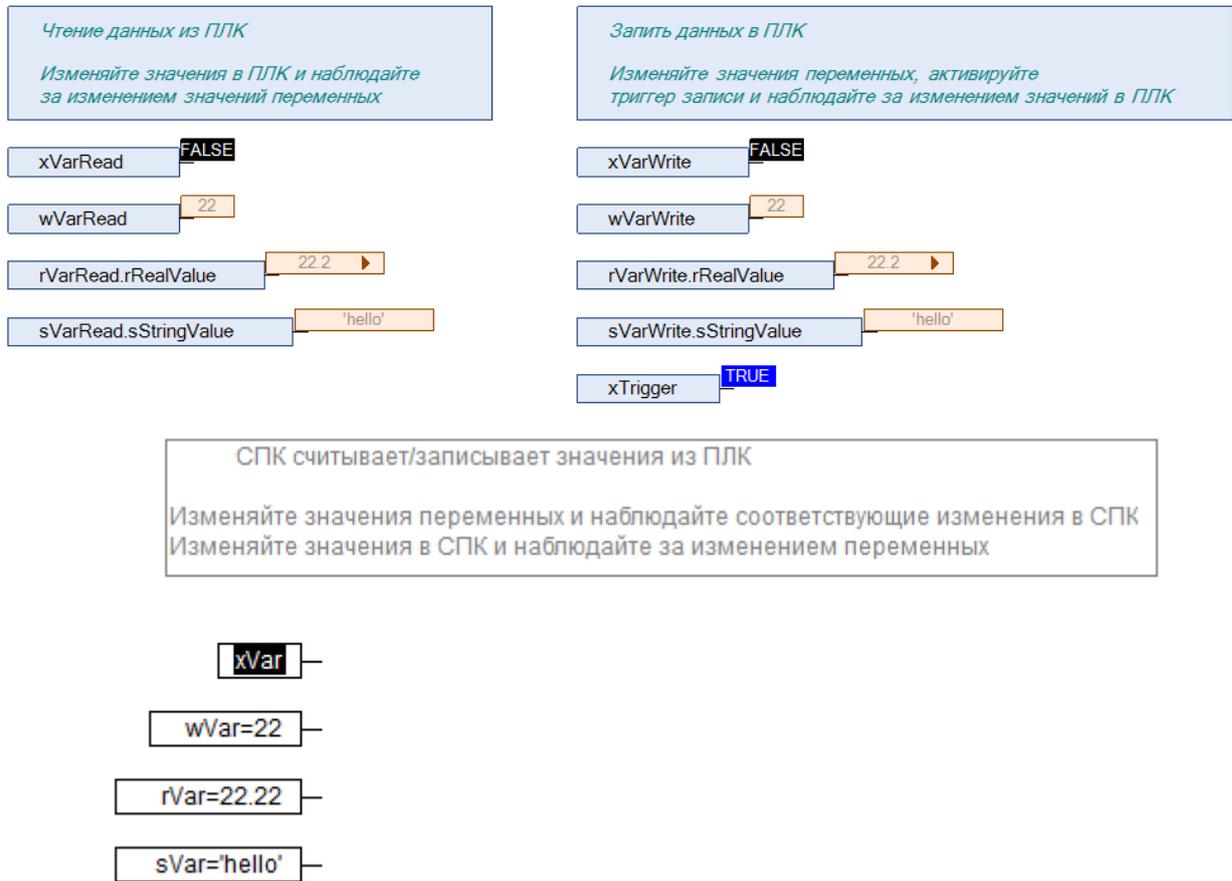


Рисунок 5.22 – СПК записывает данные в ПЛК

## 6 Modbus TCP. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

### 6.1 Описание примера

Пример посвящен настройке обмена данными между сенсорным панельным контроллером **СПК1хх [01]** и контроллером **ПЛК110 [M02]** по протоколу **Modbus TCP**. В примере СПК выполняет функцию **Slave**, ПЛК – **Master**.

Основные характеристики используемых устройств приведены в таблице 6.1.

Используемые в примере переменные описаны в таблице 6.2.

**Таблица 6.1 – Характеристики устройств**

Устройство	СПК1хх [M01]	ПЛК110 [M02]
Функция	Slave	Master
IP адрес	10.2.11.20	10.2.11.10
Маска подсети	255.255.0.0	
Порт TCP	502	
Slave ID	1	–
Таргет	3.5.17.31	PLC110.30-M v2 (версия 3.3)
Версия прошивки	2.4.0923.1000	1.2.42
Среда разработки проекта	CODESYS V3.5 SP17 Patch 3	CoDeSys V2.3.9.41
Название файла проекта	ModbusTCPslave.projectarchive	ModbusTCPmaster.pro

**Таблица 6.2 – Список переменных**

СПК1хх [M01] (Slave)			ПЛК110 [M02] (Master)
Переменные, которые считывает Master	Адрес регистра/бита	Тип данных	Переменные ПЛК
xVarFromSPK	0 (Discrete Inputs)	BOOL	xVarFromSPK
wVarFromSPK	1 (Input Registers)	WORD	wVarFromSPK
rVarFromSPK	2–3 (Input Registers)	REAL	rVarFromSPK
sVarFromSPK	4–5 (Input Registers)	STRING(6)	sVarFromSPK
Переменные, которые записывает Master	Адрес регистра/бита	Тип данных	Переменные ПЛК
xVarToSPK	0 (Coils)	BOOL	xVarToSPK
wVarToSPK	1 (Holding Registers)	WORD	wVarToSPK
rVarToSPK	2–3 (Holding Registers)	REAL	rVarToSPK
sVarToSPK	4–6 (Holding Registers)	STRING(6)	sVarToSPK

Проекты примера доступны для скачивания: [Example\\_SpkModbusTcpSlave.zip](#)

## 6.2 Настройка СПК (slave)

Для настройки СПК (slave) следует:

1. Создать новый проект **CODESYS V3.5** для СПК1хх [M01] с программой **PLC\_PRG** на языке **CFC**.
2. Добавить в проект [объединение](#) с именем **Real\_Word**:

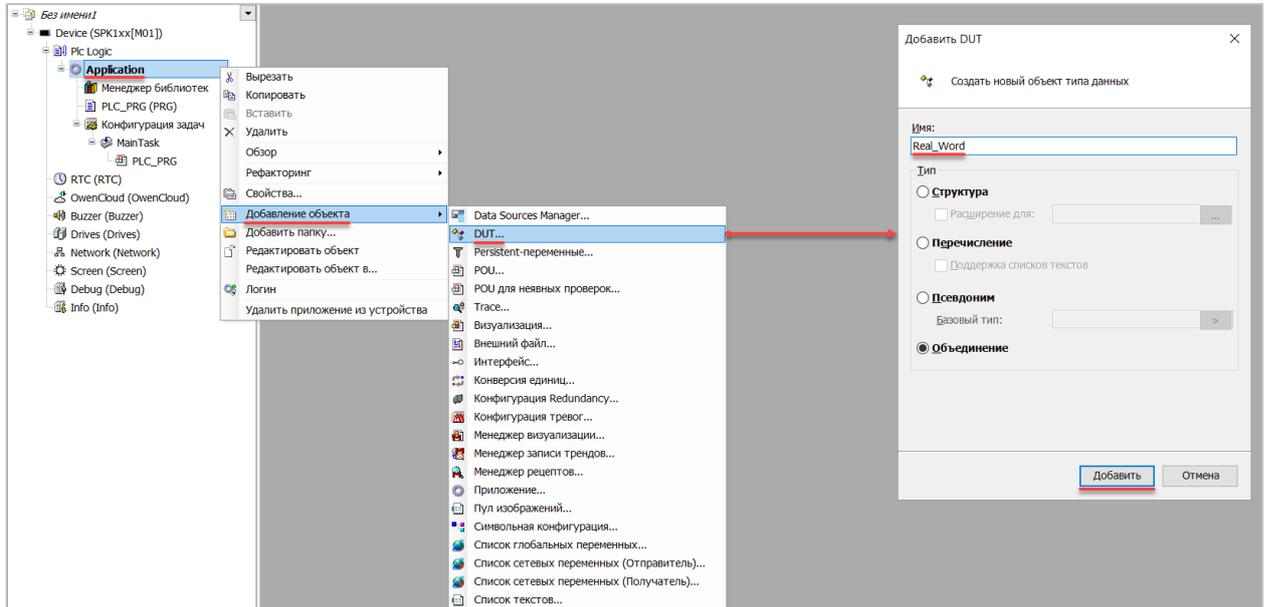


Рисунок 6.1 – Добавление в проект объединения

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

```

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

```

Рисунок 6.2 – Объявление переменных объединения

3. Добавить в проект [объединение](#) с именем **String\_Word**.

В объединении следует объявить переменную **sStringValue** типа **STRING** и массив **awModbusString** типа **WORD**, содержащий три элемента (**STRING** сможет содержать до 6 символов, так как каждый **WORD** может содержать два символа):

```

1  TYPE String_Word :
2  UNION
3      sStringValue:   STRING(6);
4      awModbusString: ARRAY [0..2] OF WORD;
5  END_UNION
6  END_TYPE

```

Рисунок 6.3. – Объявление переменных объединения

6. Modbus TCP. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

4. Объявить в программе **PLC\_PRG** 8 переменных – 4 из них будут считываться ПЛК, 4 – записываться ПЛК. **Следует обратить внимание** на [п. 2.3](#).

```
PLC_PRG x
1 PROGRAM PLC_PRG
2 VAR
3     (*данные, считываемые ПЛК*)
4     xVarFromSPK:   BOOL;
5     wVarFromSPK:   WORD;
6     rVarFromSPK:   Real_Word;
7     sVarFromSPK:   String_Word;
8
9     (*данные, записываемые ПЛК*)
10    xVarToSPK:     BOOL;
11    wVarToSPK:     WORD;
12    rVarToSPK:     Real_Word;
13    sVarToSPK:     String_Word;
14 END_VAR
```

Рисунок 6.4 – Объявление переменных программы

5. Код программы будет выглядеть следующим образом:

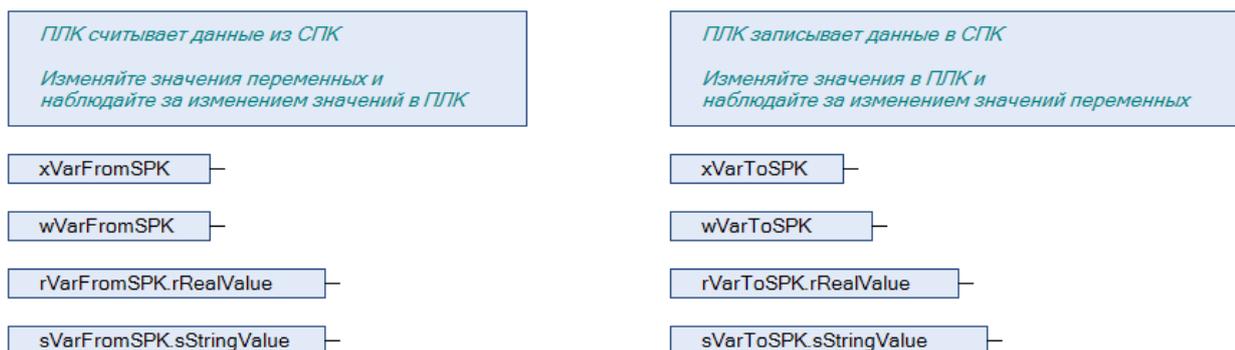


Рисунок 6.5 – Код программы на языке CFC

6. Добавить в проект компонент **Ethernet**.



**ПРИМЕЧАНИЕ**

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – может.

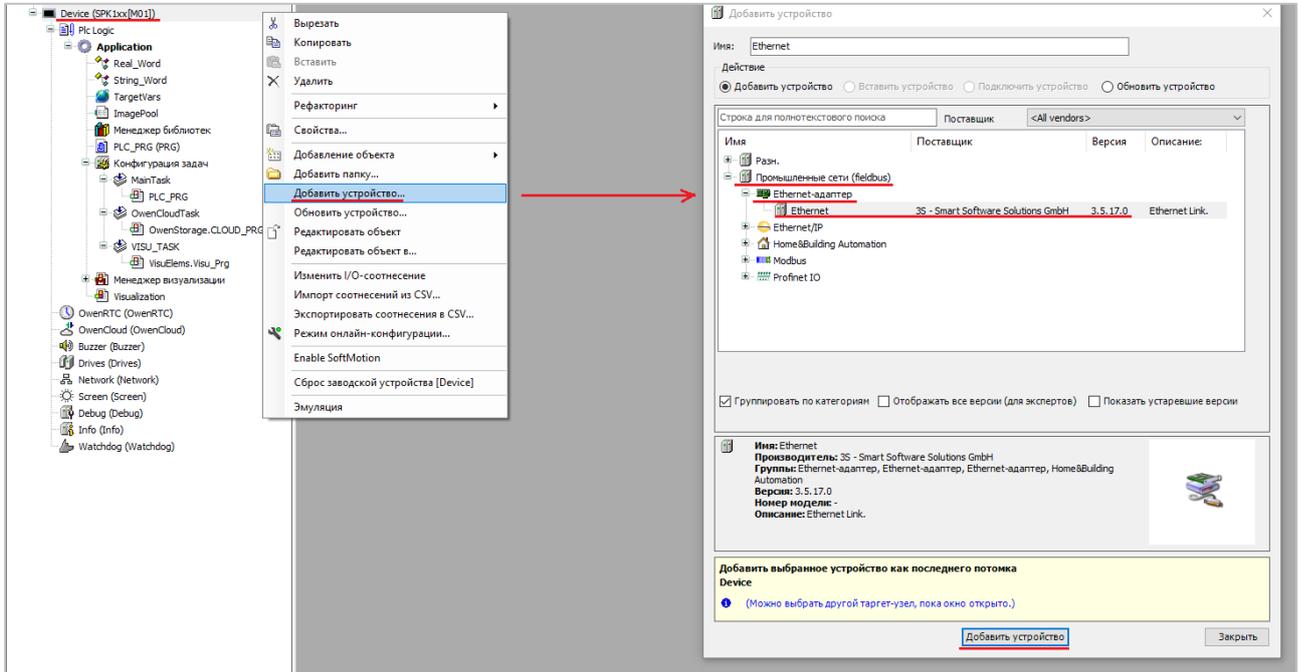


Рисунок 6.6 – Добавление компонента Ethernet

В конфигурации **Ethernet** следует указать сетевые настройки в соответствии с [таблицей 6.1](#).

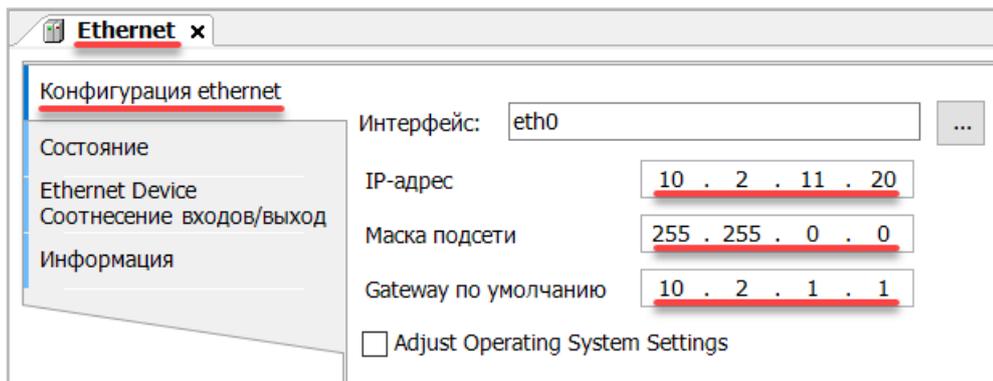


Рисунок 6.7 – Настройки компонента Ethernet

## 6. Modbus TCP. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

### 7. В компонент Ethernet добавить компонент Modbus TCP Slave Device.



#### ПРИМЕЧАНИЕ

В версии CODESYS V3.5 SP17 и выше версия компонента может превышать версию таргет-файла. В более ранних версиях среды – может.

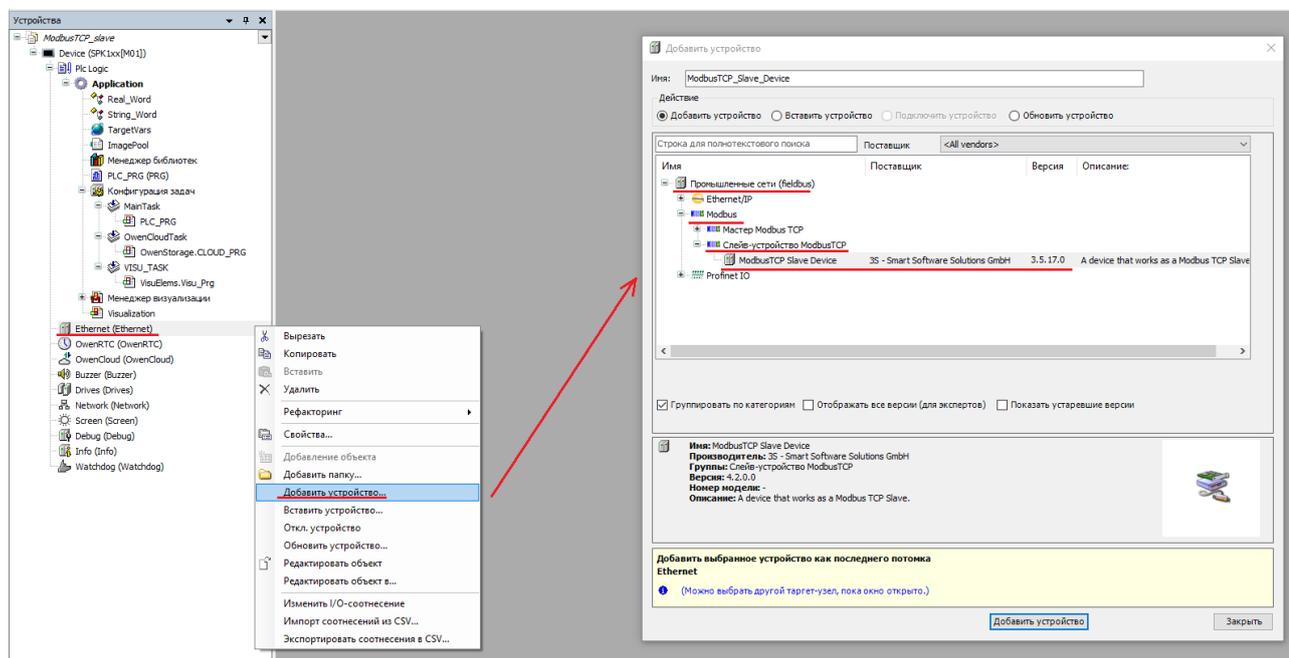


Рисунок 6.8 – Добавление компонента Modbus TCP Slave Device

В настройках компонента на вкладке **Страница конфигурации** можно задать настройки slave-устройства (в соответствии с [таблицей 6.1](#)).

- **Slave-порт** – номер порта, используемого для обмена (по умолчанию – **502**);
- **Unit ID** – адрес slave-устройства (по умолчанию slave отвечает на запросы с любым адресом);
- **Регистры временного хранения** – число holding регистров slave'a;
- **Входные регистры** – число input регистров slave'a.
- **Регистры** – число coils slave'a.
- **Дискретные входы** – число discrete inputs slave'a.

**Обратите внимание**, что по умолчанию в реализации CODESYS значения **holding регистров** и **coils'ов** не могут быть изменены из кода программы контроллера (только со стороны master-устройства).

Чтобы holding регистры и coils стали доступны для изменения из программы контроллера – необходимо установить галочку **Запись** в настройках компонента **Modbus TCP Slave Device**.

**Обратите внимание**, что по умолчанию в реализации CODESYS области памяти битов и регистров наложены друг на друга (общие области для **Coils/Holding регистров** и **Discrete Inputs/Input регистров**).

Чтобы разделить указанные области – необходимо установить галочку **Дискретные битовые области** в настройках компонента **Modbus TCP Slave Device**.

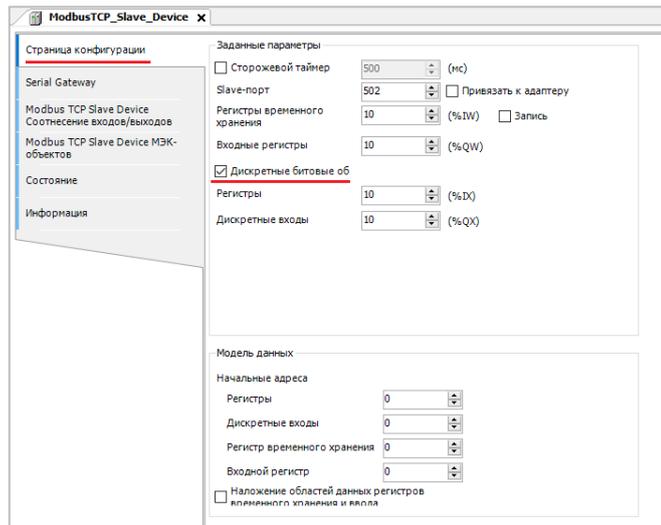


Рисунок 6.9 – Настройки компонента Modbus TCP Slave Device

Во вкладке **Modbus Serial Device Соотнесение входов/выходов** следует привязать к регистрам переменные программы в соответствии с [таблицей 6.2](#). **Следует обратить внимание** на порядок **WORD** для переменных типа **REAL**.

У параметра **Всегда обновлять переменные** должно быть выставлено значение **Включено 2**.

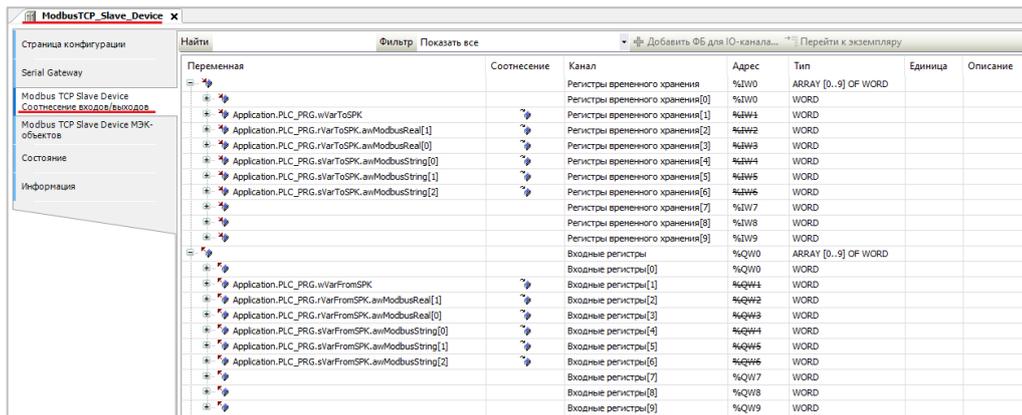


Рисунок 6.10 – Привязка переменных к регистрам slave-устройства (holding and input регистры)

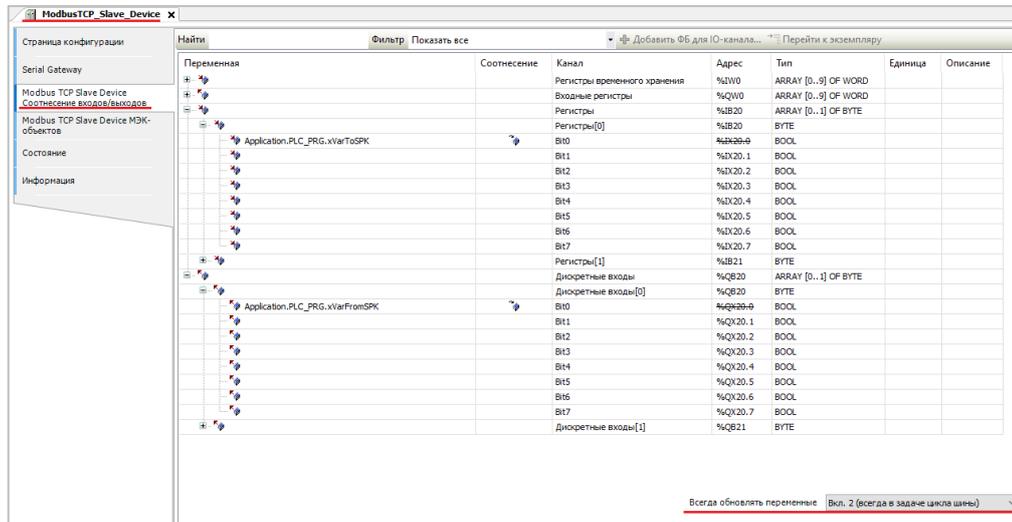


Рисунок 6.11 – Привязка переменных к регистрам slave-устройства (coils and discrete inputs)

Настройка СПК (slave) завершена.

### 6.3 Настройка ПЛК (master)

Для настройки ПЛК (master) следует:

1. Создать новый проект **CoDeSys V2.3** для ПЛК110 с программой **PLC\_PRG** на языке **CFC**. Нажать **ПКМ** на название контроллера (в данном примере – **PLC110\_30**) и добавить подэлемент **Modbus (Master)**:

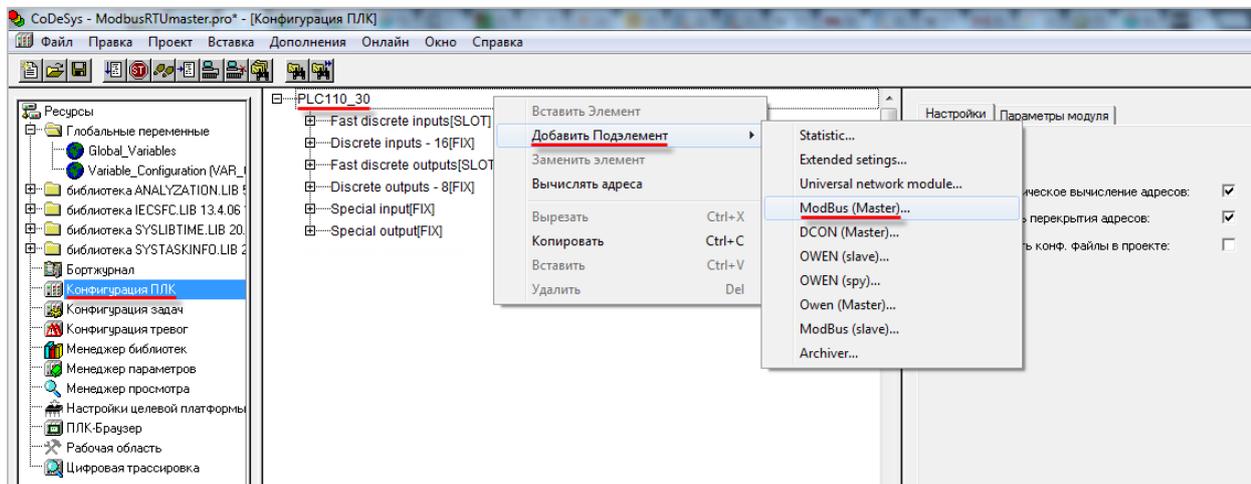


Рисунок 6.12 – Конфигурация ПЛК. Добавление Modbus (Master)

Элемент не нуждается в настройках.

2. Нажать **ПКМ** на элемент **Modbus (Master)** и добавить два подэлемента **Universal Modbus Device**:

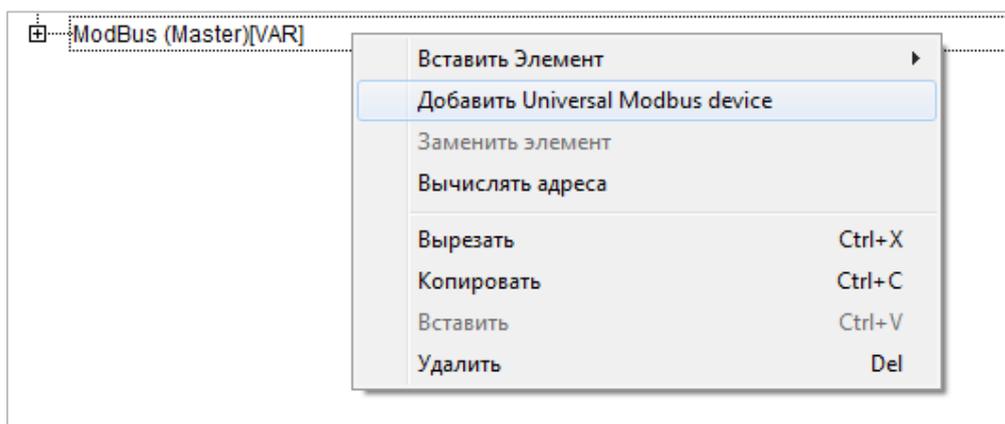


Рисунок 6.13 – Конфигурация ПЛК. Добавление Universal Modbus Device

Один из подэлементов будет использоваться для чтения значений из СПК, второй подэлемент – для записи.

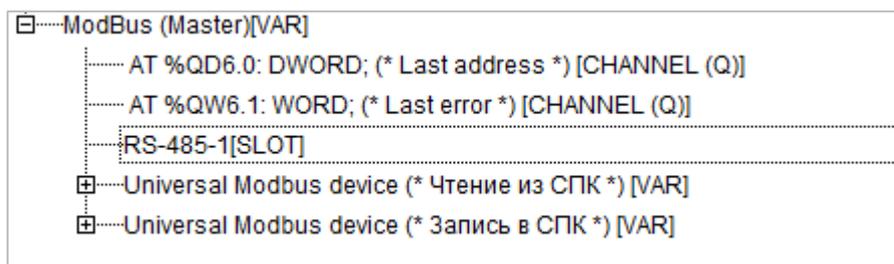


Рисунок 6.14 – Внешний вид Конфигурации ПЛК после добавления двух Universal Modbus Device

В настройках элемента **Чтение из СПК** следует указать тип связи (**TCP**), IP-адрес, порт и **Slave ID** СПК (в соответствии с [таблицей 6.1](#)) и режим опроса (**By poll time**, т. е. циклически).

Базовые параметры		Параметры модуля		
Индекс	Имя	Значение	По умолч.	Мин.
1	Name	Universal Modbus d...	Universal Modbus d...	
2	ModuleIP	10.2.11.20	10.0.0.223	
3	Max timeout	150	150	10
4	TCPport	502	502	
5	NetMode	TCP	Serial	
6	ModuleSlave...	1	1	0
7	Work mode	By poll time	By poll time	
8	Polling time ms	100	100	10
9	Visibility	No	No	
10	Amount Rep...	0	0	0
11	Byte Sequen...	Trace_mode	Trace_mode	

Рисунок 6.15 – Настройки Universal Modbus Device (Чтение из СПК)

В настройках элемента **Запись в СПК** следует указать тип связи (**TCP**), IP-адрес, порт и **Slave ID** СПК (в соответствии с [таблицей 6.1](#)) и режим опроса (**By value change**, т. е. спорадически).

Базовые параметры		Параметры модуля		
Индекс	Имя	Значение	По умолч.	
1	Name	Universal Modbus d...	Universal Modbus d...	
2	ModuleIP	10.2.11.20	10.0.0.223	
3	Max timeout	150	150	
4	TCPport	502	502	
5	NetMode	TCP	Serial	
6	ModuleSlave...	1	1	
7	Work mode	By value change	By poll time	
8	Polling time ms	100	100	
9	Visibility	No	No	
10	Amount Rep...	0	0	
11	Byte Sequen...	Trace_mode	Trace_mode	

Рисунок 6.16 – Настройки Universal Modbus Device (Запись в СПК)

Затем следует нажать **ПКМ** на элемент **Universal Modbus Device (Чтение из СПК)** и добавить в него подэлементы **Register Input Module**, **Real Input Module** и **String Input Module**.

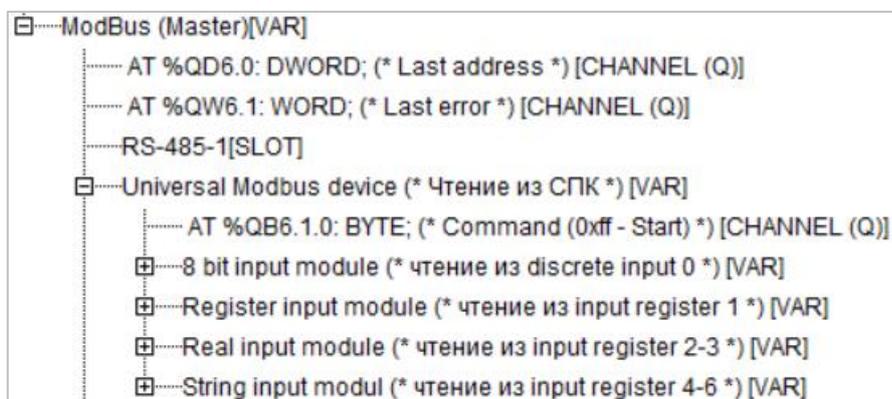
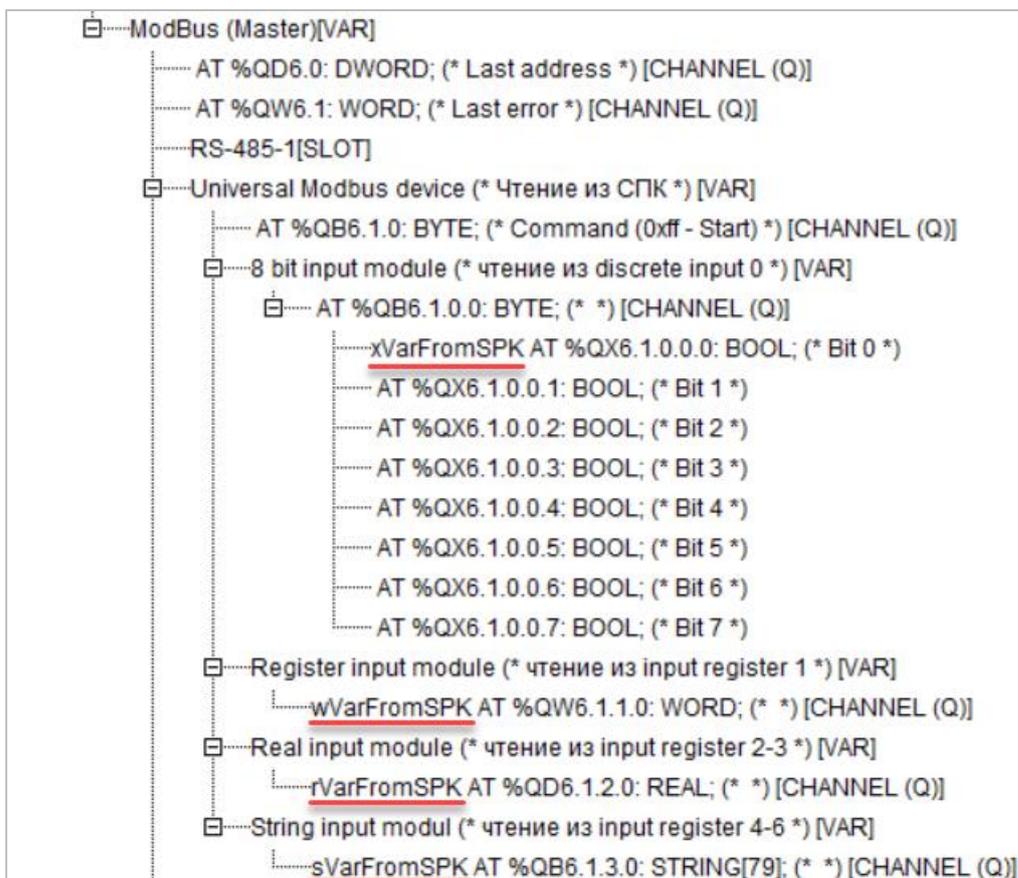


Рисунок 6.17 – Universal Modbus Device (Чтение из СПК) с добавленными Input модулями

## 6. Modbus TCP. ПЛК (CDS V3.5) – slave, ПЛК (CDS V2.3) – master

К каждому из каналов следует привязать переменную (после ввода ее имени она автоматически будет добавлена в список глобальных переменных проекта). Для ввода имени переменной следует два раза нажать на **AT**.



**Рисунок 6.18 – Привязка переменных к каналам**

Настройки модулей (используемые регистры СПК в соответствии с [таблицей 6.1](#) и функции Modbus) приведены ниже.



### ПРИМЕЧАНИЕ

Во время работы с переменными, занимающими несколько регистров СПК (тип **REAL** и **STRING**), указывается только первый из группы регистров.

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	8 bit input module	8 bit input module
2	Register address	0	0
3	Command	Read inputs status (0x02)	Read inputs status (0x02)
8	Visibility	No	No

**Рисунок 6.19 – Параметры 8 bit Input Module**

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	Register input module	Register input module
2	Register Address	1	0
3	Command	Read input registers (0x04)	Read holding Registers (0x03)
8	Visibility	No	No

Рисунок 6.20 – Параметры Register Input Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	float input module	float input module
2	Register Address	2	0
3	Command	Read input registers (0x04)	Read holding Registers (0x03)
8	Visibility	No	No

Рисунок 6.21 – Параметры Real Input Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	String input module	String input module
2	Command	Read input registers (0x04)	Read bytes (0x70)
3	Register address	4	0
4	Amount byte	6	80
8	Visibility	No	No

Рисунок 6.22 – Параметры String Input Module

Нажать ПКМ на элемент **Universal Modbus Device (Запись в СПК)** и добавить в него подэлементы **Register Output Module**, **Real Output Module** и **String Output Module**.

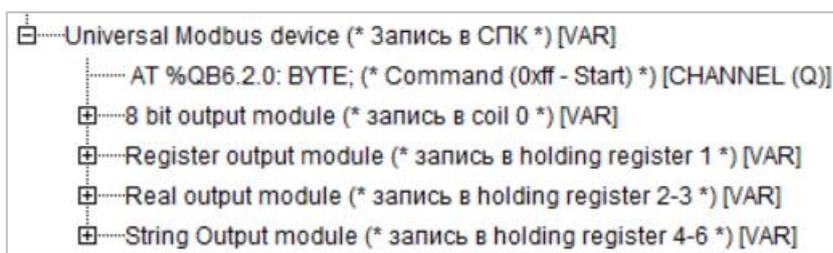


Рисунок 6.23 – Universal Modbus Device (Запись в СПК) с добавленными Output модулями

К каждому из каналов следует привязать переменную (после ввода ее имени она автоматически будет создана в проекте как глобальная). Для ввода имени переменной два раза нажмите на **АТ**.

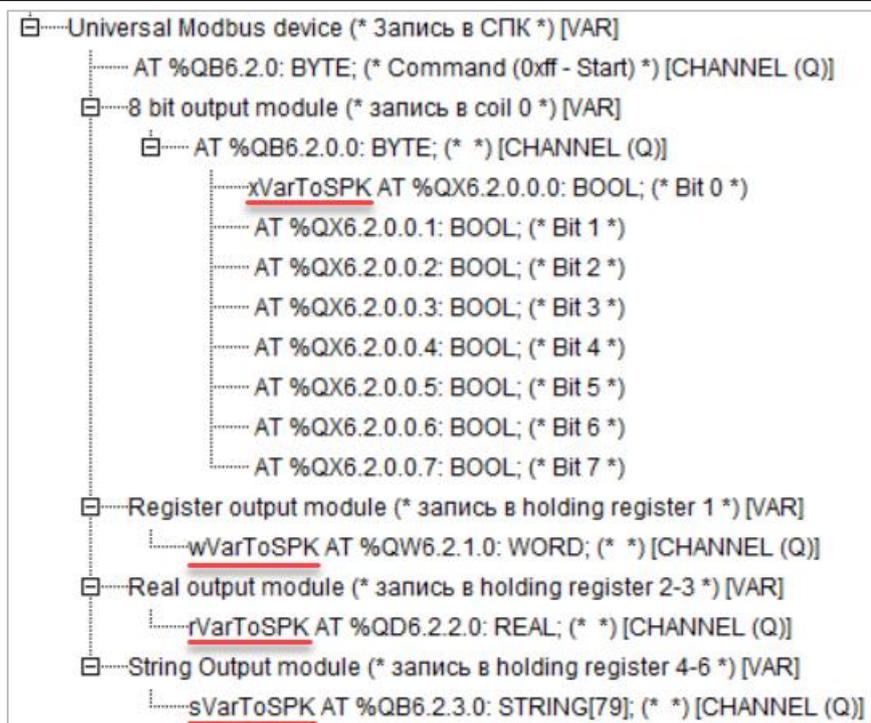


Рисунок 6.24 – Привязка переменных к каналам

Настройки модулей (используемые регистры СПК в соответствии с [таблицей 6.1](#) и функции Modbus) приведены ниже.



**ПРИМЕЧАНИЕ**

Во время работы с переменными, занимающими несколько регистров СПК (тип **REAL** и **STRING**), указывается только первый из группы регистров.

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	8 bit output module	8 bit output module
2	Register Address	0	0
3	Command	Force multiply coils (0x0f)	Force multiply coils (0x0f)
8	Visibility	No	No

Рисунок 6.25 – Параметры 8 Bit Output Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	Register	Register
2	Register Address	1	0
3	Command	Preset singl register (0x06)	Preset singl register (0x06)
8	Visibility	No	No

Рисунок 6.26 – Параметры Register Output Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	float output module	float output module
2	Register Address	2	0
3	Command	Preset multiple Registers (0x10)	Preset multiple Registers (0x10)
8	Visibility	No	No

Рисунок 6.27 – Параметры Real Output Module

Базовые параметры		Параметры модуля	
Индекс	Имя	Значение	По умолч.
1	Name	String output module	String output module
2	Command	Preset multiple Registers (0x10)	Preset singl register (0x06)
3	Register address	4	0
4	Amonth byte	6	80
8	Visibility	No	No

Рисунок 6.28 – Параметры String Output Module

3. Программа PLC\_PRG будет выглядеть следующим образом:

```

0001 PROGRAM PLC_PRG
0002 VAR
0003
0004 END_VAR
0005
0006
0007

```

Чтение данных из СПК

Изменяйте значения в СПК и наблюдайте за изменением значений переменных

xVarFromSPK

wVarFromSPK

rVarFromSPK

sVarFromSPK

Запись данных в СПК

Изменяйте значения переменных и наблюдайте за изменением значений в СПК

xVarToSPK

wVarToSPK

rVarToSPK

sVarToSPK

Рисунок 6.29 – Код программы PLC\_PRG

**ПРИМЕЧАНИЕ**

Проект не содержит операций и используется только для отображения и ввода значений. Пользователь должен создать программу для реализации необходимых алгоритмов.

Настройка ПЛК (master) завершена.

## 6.4 Работа с примером

Проекты следует загрузить в оба устройства и запустить.

В случае изменения значения **ToSPK** переменных в ПЛК, соответствующие изменения будут отображаться в программе СПК:

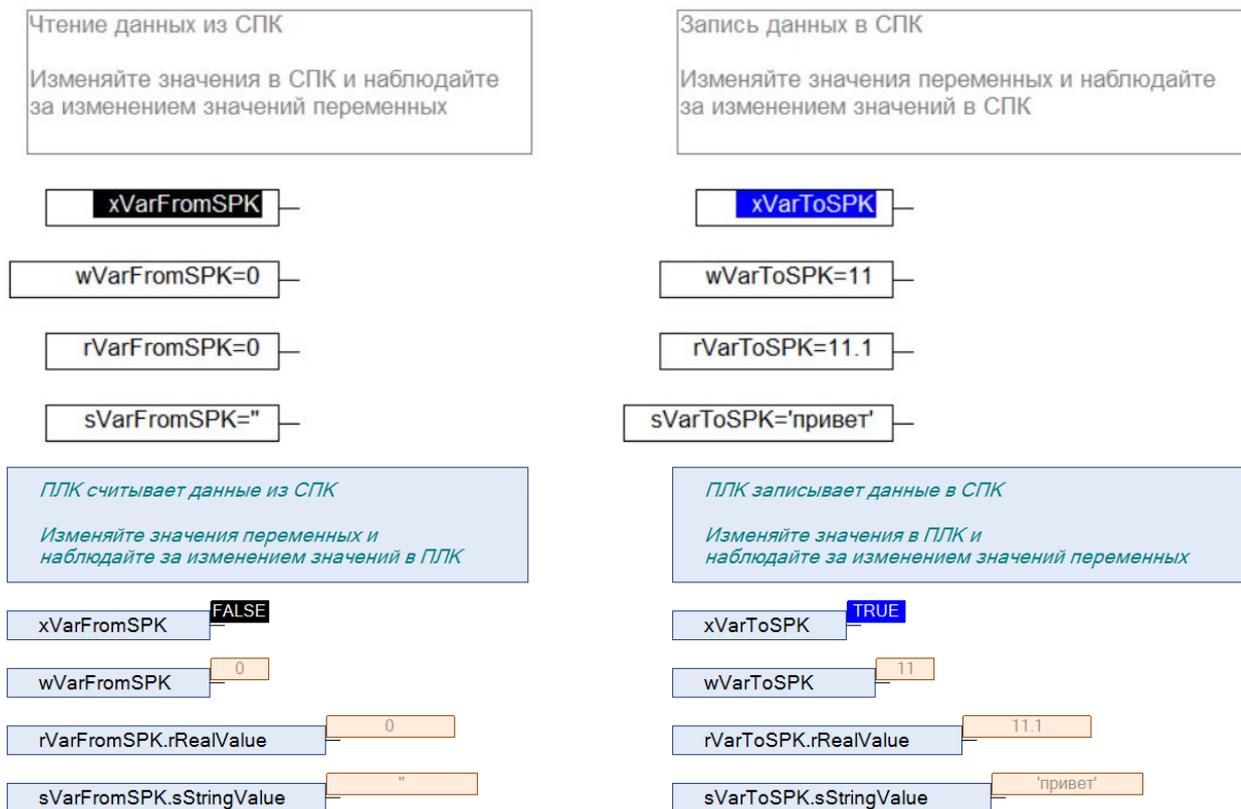


Рисунок 6.30 – ПЛК записывает данные в СПК

В случае изменения значения **FromSPK** переменных СПК, соответствующие изменения будут отображаться в программе ПЛК.

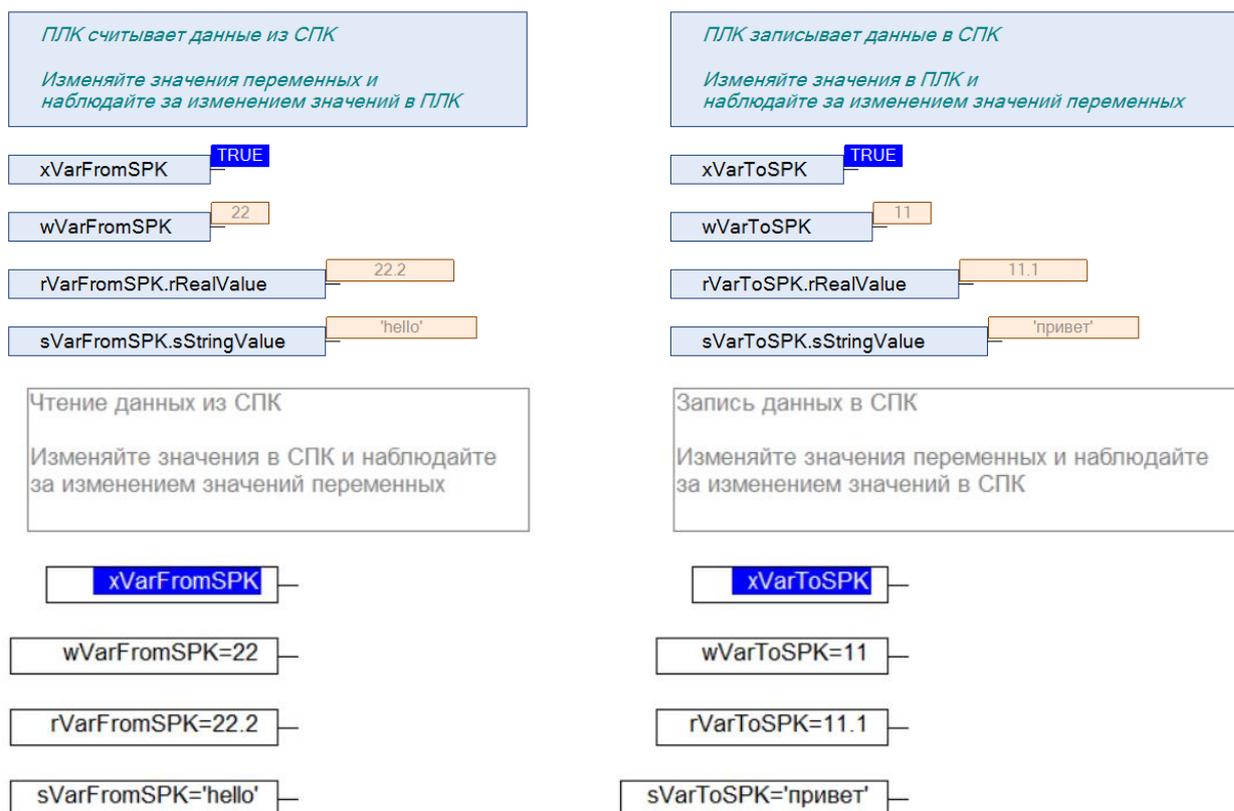


Рисунок 6.31 – ПЛК считывает данные из СПК