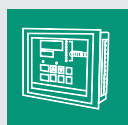


# СПЗхх

## Примеры настройки обмена

ПЛК323 (Modbus TCP Slave) + СП307-Р (Modbus TCP Master)

Версия: 1.0  
Дата: 06.11.2015



## Оглавление

1. Описание примера .....	3
2. Настройки ПЛК323 (Modbus TCP Slave).....	4
2.1. Сетевые настройки контроллера .....	4
2.2. Создание объединений для REAL и STRING .....	5
2.3. Программа PLC_PRG.....	7
2.4. Добавление и настройка компонентов Modbus.....	8
2.5. Конфигурация задач.....	11
3. Настройки СП307-Р (Modbus TCP Master).....	12
3.1. Сетевые настройки панели.....	12
3.2. Экран проекта .....	15
3.2.1. Цифровой ввод (WORD) .....	16
3.2.2. Цифровой ввод (REAL).....	18
3.2.3. Текстовый ввод (STRING).....	20
4. Работа с примером.....	21

## 1. Описание примера

Данный пример посвящен настройке обмена данными между панелью оператора **СП3хх-Р** и контроллером **ПЛК323** по протоколу **Modbus TCP**. В этом примере контроллер выполняет функцию **Slave**, а панель – **Master**.

Обратите внимание, что Ethernet-порт присутствует только у расширенной модификации панели СП3хх (**СП3хх-Р**).

Подразумевается, что пользователь ознакомлен с **РЭ** и **РП** на соответствующие приборы и имеет базовые навыки работы с ними.

Основные характеристики используемых устройств приведены в табл. 1. Используемые в примере переменные описаны в табл. 2.

Табл. 1. Характеристики используемых в примере устройств

Устройство	ПЛК323	СП307-Р
Функция	Slave	Master
Используемый порт	Ethernet	Ethernet
IP-адрес	10.0.6.10	10.0.6.20
Маска подсети	255.255.0.0	
Таргет	OWEN PLAC323.03-CS.WEB (3.5.0.40 [005])	-
Среда разработки проекта	CODESYS 3.5 SP3 Patch2	Конфигуратор СП300 (V2.D3d-2)
Название файла проекта	ModbusTCPslave.projectarchive	ModbusTCPmaster.txp

Табл. 2. Список используемых в примере переменных

ПЛК323			СП307-Р	
Переменные, которые читает панель	Переменные, которые записывает панель	Тип данных	Регистр ПЛК (чтение)	Регистр ПЛК (запись)
wWordWrite	wWordRead	WORD	3x0	4x0
ModbusREALwrite.rReal	ModbusREALread.rReal	REAL	3x1	4x1
ModbusSTRINGwrite.sString	ModbusSTRINGread.sString	STRING	3x3	4x3

**Обратите внимание**, что при необходимости записывать и считывать переменные типа **BOOL**, необходимо в программе ПЛК упаковывать/извлекать их из переменных типа **WORD**.

## 2. Настройки ПЛК323 (Modbus TCP Slave)

### 2.1. Сетевые настройки контроллера

В данном примере мы используем для контроллера сетевые настройки, заданные ему по умолчанию – IP-адрес **10.0.6.10** и маску подсети **255.255.0.0**.

При необходимости изменения сетевых настроек, воспользуйтесь веб-конфигуратором. Для этого введите в браузере IP-адрес контроллера. Login/пароль – **www-data / 54321**.

The screenshot shows a web browser window with the address bar set to 10.0.6.10. The page title is "Настройки ПЛК 323" (PLC 323 Settings) with version 1.10.1. The left sidebar contains links: "Дата/Время", "Сеть" (selected), "Безопасность", "Опции ПЛК", and "Утилиты". The main content area is titled "Сетевые настройки - Основное" (Network Settings - Basic) and includes tabs for "Основное", "GPRS", "VPN", and "DDNS". The "Основное" tab is active, displaying the following settings:

Parameter	Value
Имя хоста *	plc323kis
DHCP	Да <input type="radio"/> Нет <input checked="" type="radio"/>
IP-адрес *	10 . 0 . 6 . 10
IP-маска *	255 . 255 . 0 . 0
Адрес шлюза *	10 . 0 . 1 . 1
DNS1	10 . 10 . 10 . 1
DNS2	10 . 10 . 10 . 15

At the bottom right of the settings area are buttons for "OK" and "Сброс" (Reset).

Рис. 1. Внешний вид **web-конфигуратора** ПЛК323

## 2.2. Создание объединений для REAL и STRING

Протокол **Modbus** предусматривает работу только с двумя типами данных – **BOOL** и **WORD**. Для того чтобы передать или получить данные других типов (в рамках нашего примера – **REAL** и **STRING**), необходимо преобразовать их. В **CODESYS 3.5** наиболее простым способом преобразования данных является использование **объединений (Union)**. Все переменные, входящие в такие объединения, помещаются в одну область памяти. Соответственно, изменение любой из переменных затрагивает все остальные. Иными словами, каждая из переменных объединения интерпретирует данные области памяти согласно своему типу.

Создадим два объединения с названиями **ModbusREAL** и **ModbusSTRING**:

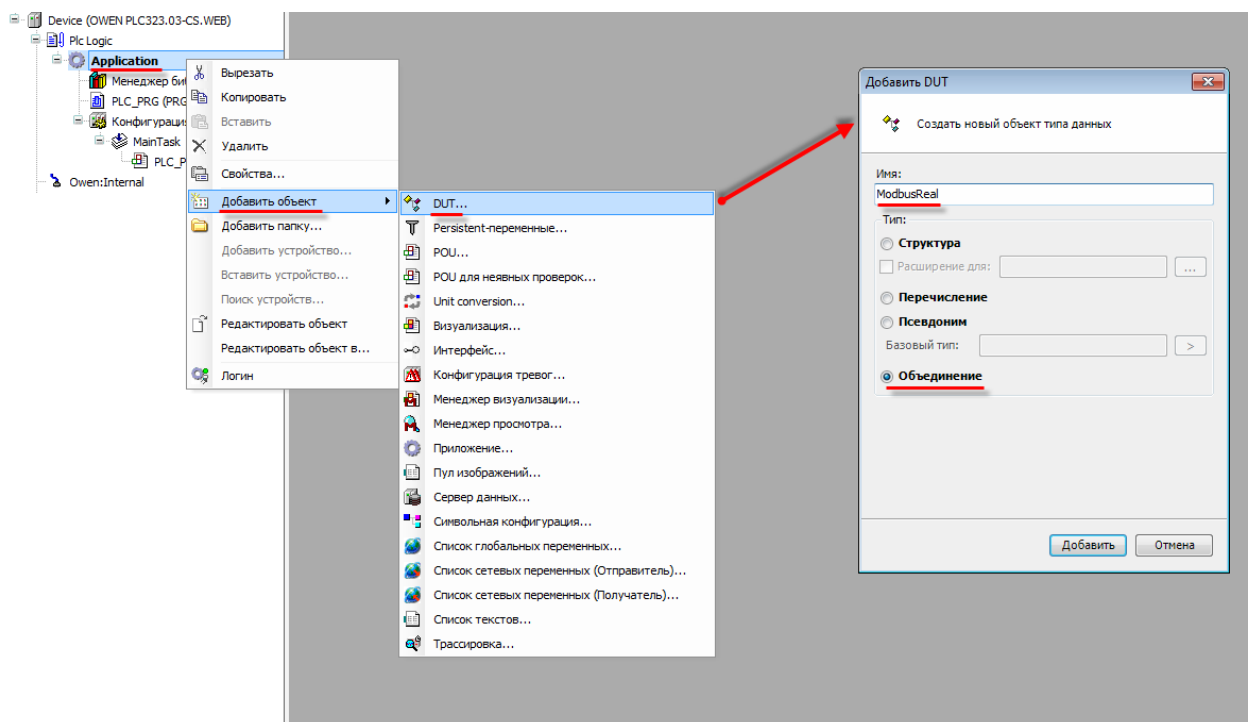


Рис. 2. Добавление объединений в проект

Объявим в объединении **ModbusREAL** массив из двух переменных типа **WORD** и переменную типа **REAL**. Соответственно, массив мы будем использовать [при настройке каналов Modbus](#), а переменную **rReal** – непосредственно в [программе](#).

```
1 TYPE ModbusREAL :  
2 UNION  
3     arrReal :ARRAY [0..1] OF WORD; // переменная для канала Modbus  
4     rReal   :REAL;                 // переменная для программы  
5 END UNION  
6 END_TYPE
```

Рис. 3. Объявление переменных объединения **ModbusREAL**

В объединении **ModbusSTRING** объявим массив из трех переменных типа **WORD** (число переменных зависит от количества передаваемых символов; в нашем примере мы работаем с текстовой строкой, которая занимает три регистра панели и может содержать до шести символов, т.к. один регистр содержит два символа) и переменную типа **STRING**. Соответственно, массив мы будем использовать [при настройке каналов Modbus](#), а переменную **sString** – непосредственно в [программе](#).

```
1 TYPE ModbusSTRING :  
2 UNION  
3     arrSTRING :ARRAY [0..2] OF WORD; // переменная для канала Modbus  
4     sString   :STRING(6);           // переменная для программы  
5 END_UNION  
6 END_TYPE
```

Рис. 4. Объявление переменных объединения **ModbusSTRING**

### 2.3. Программа PLC\_PRG

Объявим в программе **PLC\_PRG** следующие переменные:

```
1  PROGRAM PLC_PRG
2  VAR
3      wWordRead      :WORD;      // панель записывает WORD
4      wWordWrite     :WORD;      // панель считывает WORD
5
6      ModbusREALread  :ModbusReal; // панель записывает REAL
7      ModbusSTRINGread :ModbusString; // панель считывает REAL
8
9      ModbusREALwrite  :ModbusReal; // панель записывает STRING
10     ModbusSTRINGwrite :ModbusString; // панель считывает STRING
11 END_VAR
```

Рис. 5. Объявление переменных в программе **PLC\_PRG**

**Read** и **Write** переменные будут записываться/считываться панелью.

Код программы **PLC\_PRG**, написанной на языке **CFC**, выглядит следующим образом:



Рис. 6. Код программы **PLC\_PRG**

В процессе работы проекта пользователь может изменять значения **Write** переменных (непосредственно в **CODESYS**), наблюдая за соответствующими изменениями на дисплее панели (у группы элементов **Панель: чтение**), а также изменять данные на панели (с помощью сенсорного ввода для группы элементов **Панель: запись**) и наблюдать изменения у соответствующих **Read** переменных.

## 2.4. Добавление и настройка компонентов Modbus

Нажмем **ПКМ** на название контроллера (в нашем примере – **Device (OWEN PLC323.03-CS.WEB)**) и добавим компонент **Ethernet**. **Обратите внимание**, что версия компонента не должна превышать версию таргет-файла ПЛК. Для того, чтобы посмотреть все доступные версии компонента, необходимо поставить галочку **Отображать все версии**.

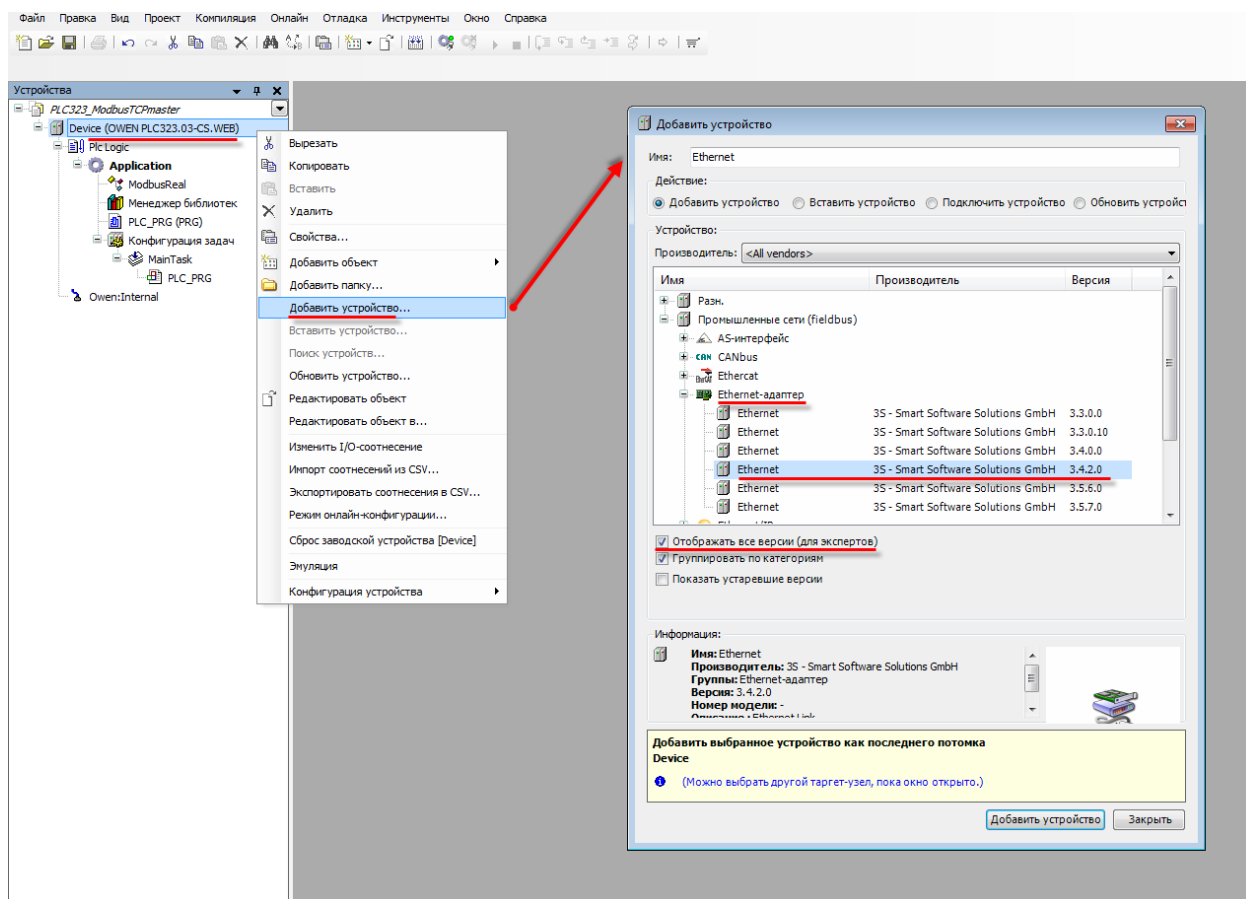


Рис. 7. Добавление компонента **Ethernet**

Этот компонент не нуждается в настройках.



Нажмем **ПКМ** на компонент **Ethernet** и добавим компонент **Master TCP Slave**. **Обратите внимание**, что версия компонента не должна превышать версию таргет-файла ПЛК. Для того, чтобы посмотреть все доступные версии компонента, необходимо поставить галочку **Отображать все версии**.

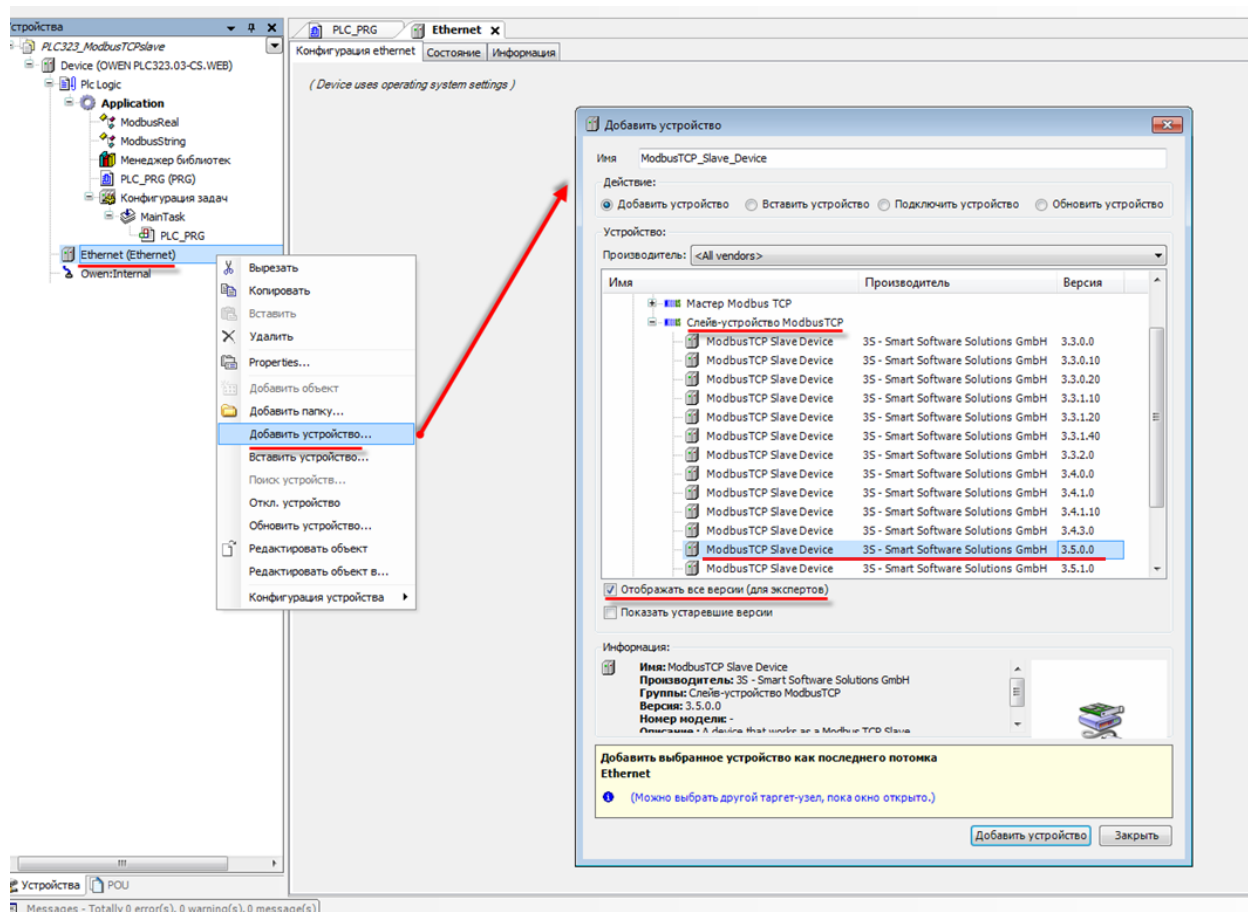


Рис. 8. Добавление компонента **Modbus TCP Slave**

На вкладке **Страница конфигурации** необходимо снять галочку **Таймаут** (поскольку панель записывает данные в панель не циклически, а по команде оператора, то при отсутствии данной галочки данные в ПЛК будут обнуляться) и указать **Slave ID** контроллера (согласно [табл. 1](#) он равен 1).

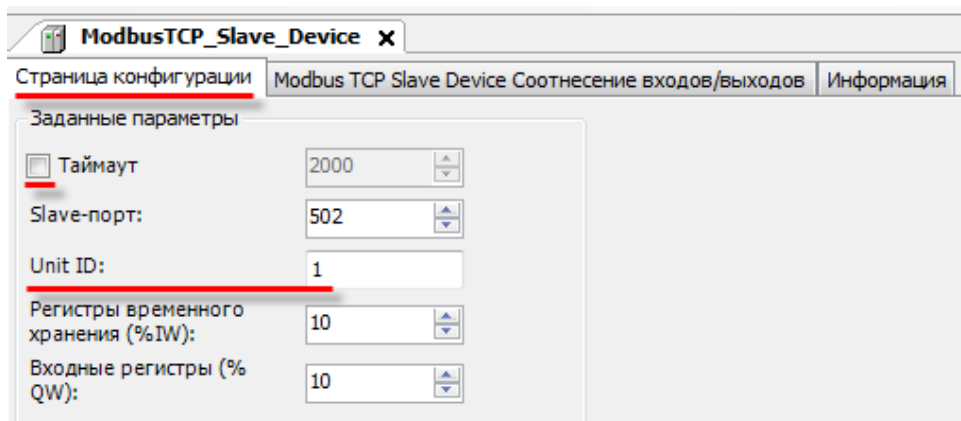


Рис. 9. Настройки вкладки **Modbus TCP Slave**

На вкладке **Modbus TCP Slave Device Соотнесение входов/выходов** привяжем переменные программы к регистрам ПЛК. Панель записывает данные в **Inputs** каналы и считывает данные из **Outputs** каналов.

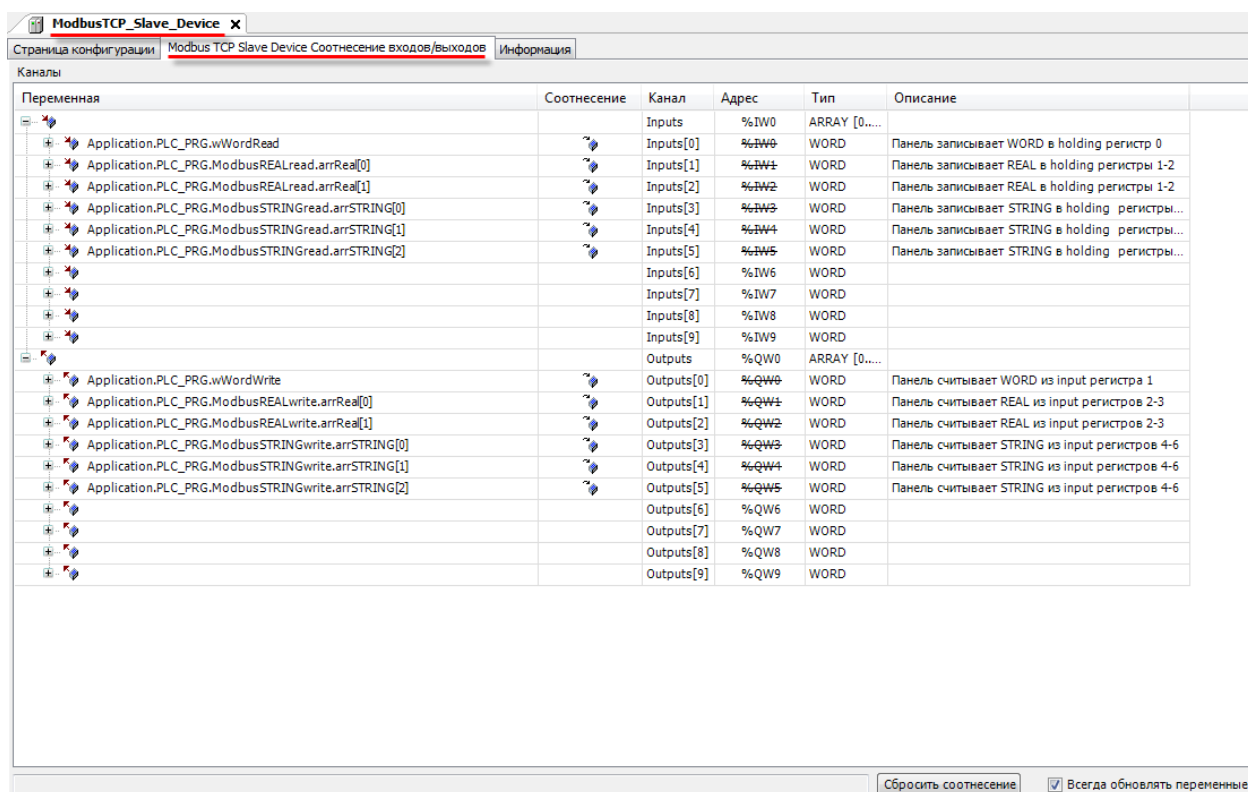


Рис. 10. Настройки вкладки **Modbus TCP Slave Device Соотнесение входов/выходов**

**Обратите внимание**, что для корректного опроса должна быть установлена галочка **Всегда обновлять переменные**.

## 2.5. Конфигурация задач

**Обратите внимание**, что для работы программы **PLC\_PRG**, она должна быть привязана к одной из задач (компонент **Конфигурация задач**).

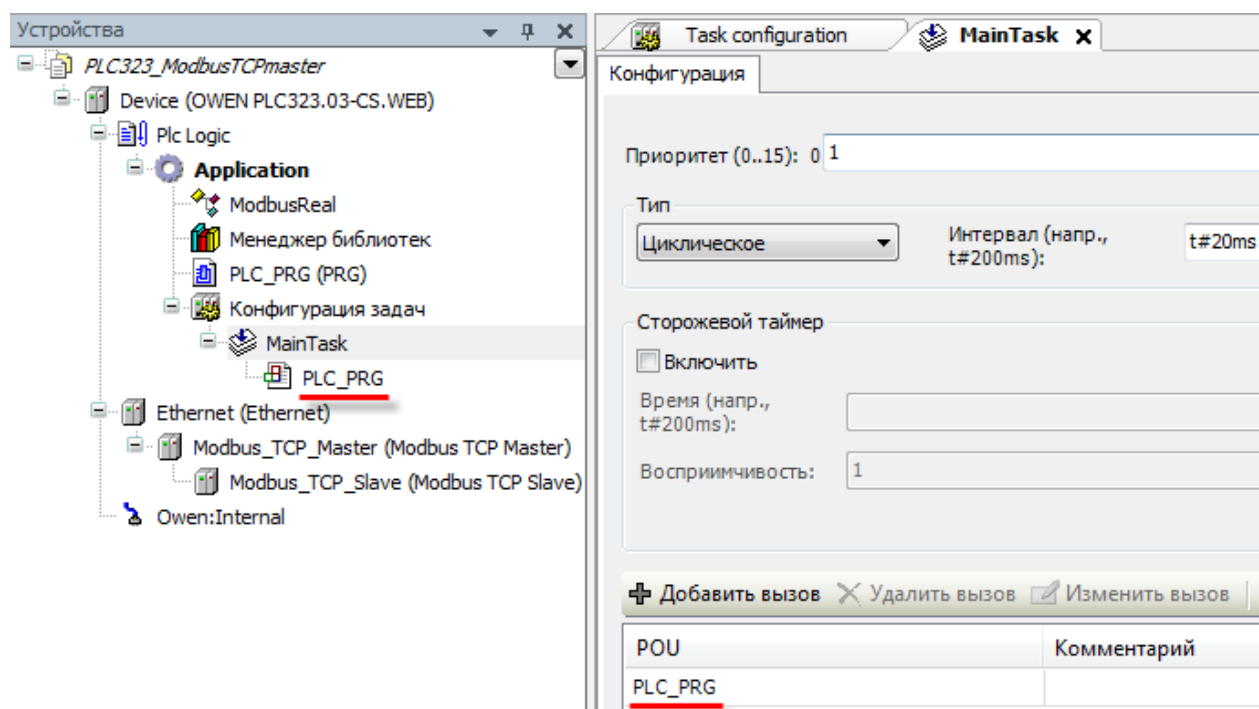


Рис. 11. Привязка программы **PLC\_PRG** к задаче

### 3. Настройки СП307-Р (Modbus TCP Master)

#### 3.1. Сетевые настройки панели

Для того чтобы использовать панель **СП307-Р** в режиме **Modbus TCP Master**, в первую очередь необходимо в настройках проекта на вкладке **Устройство** задать **IP-адрес** панели и **маску подсети**. В соответствии с [табл. 1](#) зададим IP-адрес **10.0.6.20** и маску подсети **255.255.0.0**.

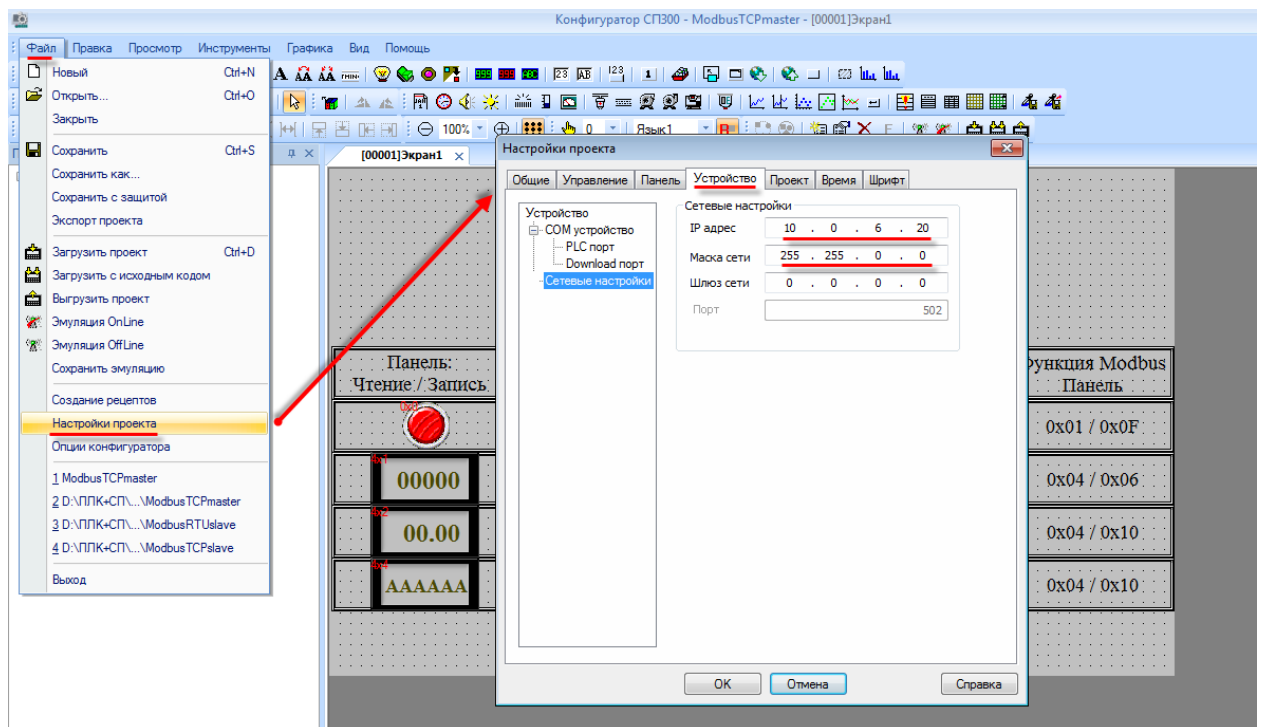


Рис. 12. Сетевые настройки панели

Нажмем **ПКМ** на вкладку **Сетевые настройки** и добавим slave-устройство с названием **ПЛК323**:

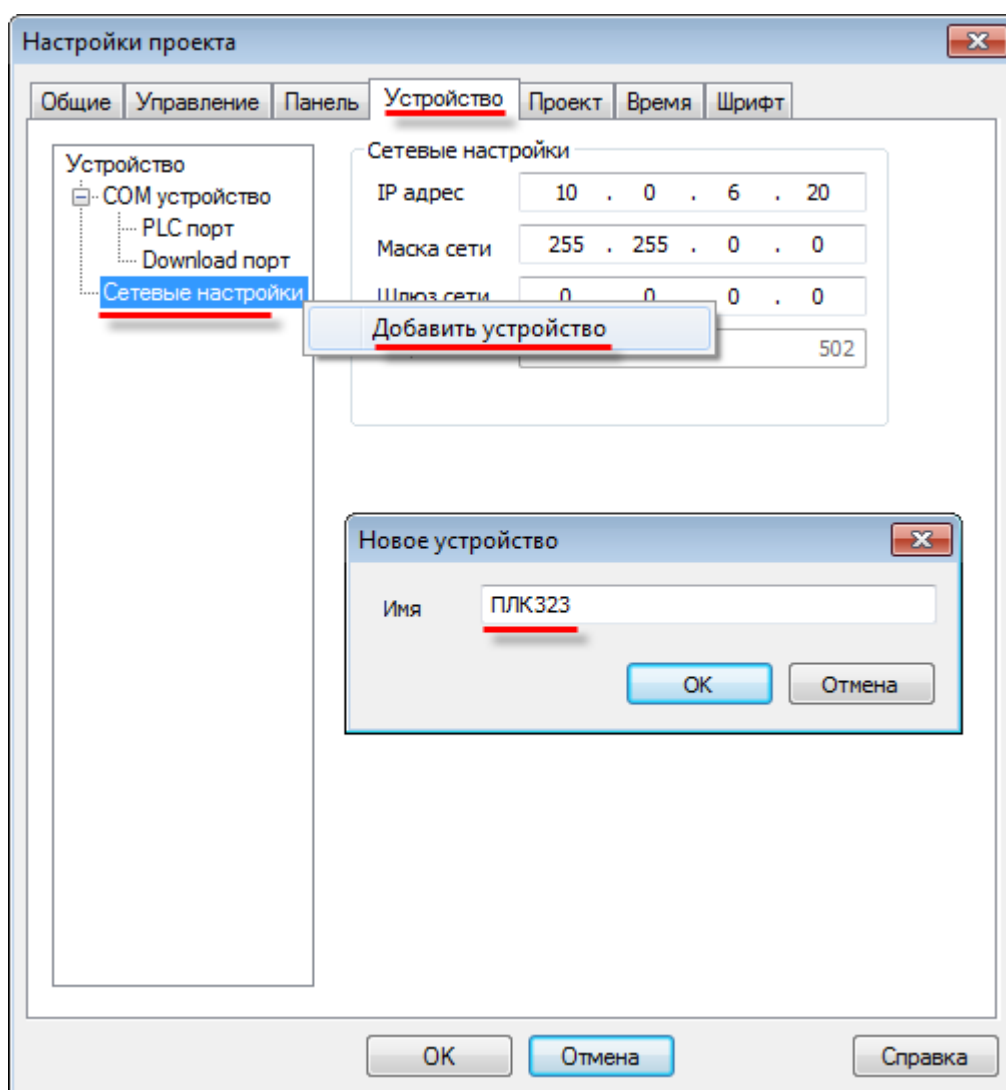


Рис. 13. Добавление **TCP Slave** устройства

В настройках slave-устройства укажем **IP-адрес** ПЛК (**10.0.6.10** согласно [табл. 1](#)) и используемый протокол (**TCP**):

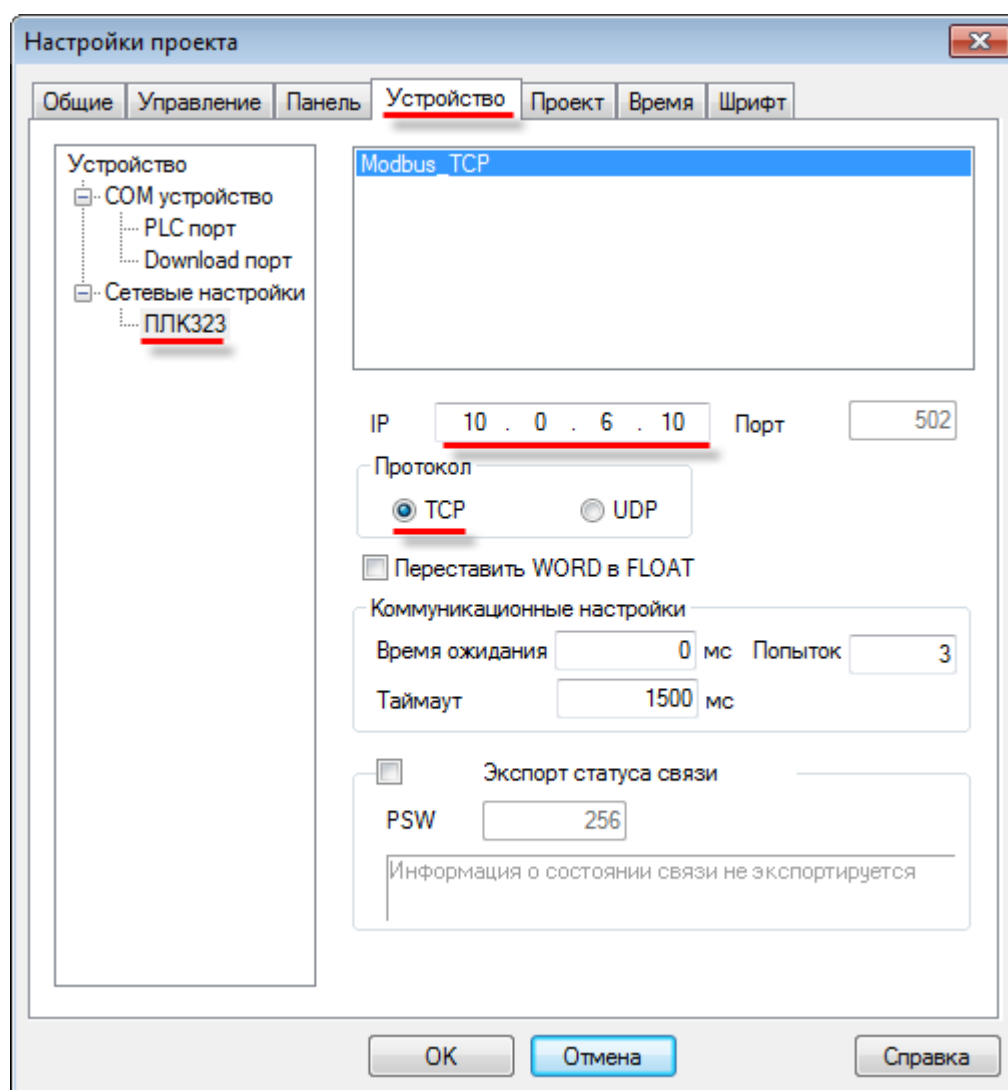


Рис. 14. Настройки **TCP Slave** устройства

### 3.2. Экран проекта

Проект содержит один экран, на котором осуществляется отображение данных, которые считываются из ПЛК, и ввод данных, которые записываются в ПЛК.

Соответственно, экран содержит шесть активных элементов:

1. **Цифровой ввод** с привязанным регистром **ПЛК 3х1**. Панель считывает значение этого регистра, к которому привязана переменная ПЛК **wWordWrite**.
2. **Цифровой ввод** с привязанным регистром **ПЛК 3х2**. Панель считывает значение этого регистра, к которому привязана переменная ПЛК **ModbusREALwrite.rReal**.
3. **Текстовый ввод** с привязанным регистром **ПЛК 3х4**. Панель считывает значение этого регистра, к которому привязана переменная ПЛК **ModbusSTRINGwrite.sString**.
4. **Цифровой ввод** с привязанным регистром **ПЛК 4х1**. Панель записывает значение в этот регистр, к которому привязана переменная ПЛК **wWordRead**.
5. **Цифровой ввод** с привязанным регистром **ПЛК 4х2**. Панель записывает значение в этот регистр, к которому привязана переменная ПЛК **ModbusREALread.rReal**.
6. **Текстовый ввод** с привязанным регистром **ПЛК 4х4**. Панель записывает значение в этот регистр, к которому привязана переменная ПЛК **ModbusSTRINGread.sString**.

Контроллер - Modbus TCP Slave					
Панель - Modbus TCP Master					
Справочная информация					
Панель: Чтение	Панель: Запись	Регистр ПЛК	Тип данных	Переменные ПЛК	Функция Modbus Панель
3х0 00000	4х0 00000	3х0 / 4х0	WORD	wWordWrite / wWordRead	0х04 / 0х06
3х1 00.00	4х1 00.00	3х1 / 4х1	REAL	ModbusREALwrite.rReal / ModbusREALread.rReal	0х04 / 0х10
3х3 AAAAAA	4х3 AAAAAA	3х3 / 4х3	STRING	ModbusSTRINGwrite.sString / ModbusSTRINGread.sString	0х04 / 0х10

Рис. 15. Внешний вид экрана проекта

Настройки элементов приведены ниже.

### 3.2.1. Цифровой ввод (WORD)

Элемент **Цифровой ввод (WORD)** используется для отображения и ввода значений переменных ПЛК типа **WORD**. Переменные этого типа являются **целочисленными** и могут принимать значение от 0 до 65535.



Рис. 16. Внешний вид элементов **Цифровой ввод**

В настройках элементов на вкладке **Регистр элемента** выберем устройство **ПЛК323**, укажем для него адрес **1** и регистры **3x0** (чтение) и **4x0** (запись) [согласно [табл. 2](#)]. **Обратите внимание**, что параметр **Тип** должен иметь значение **WORD**.

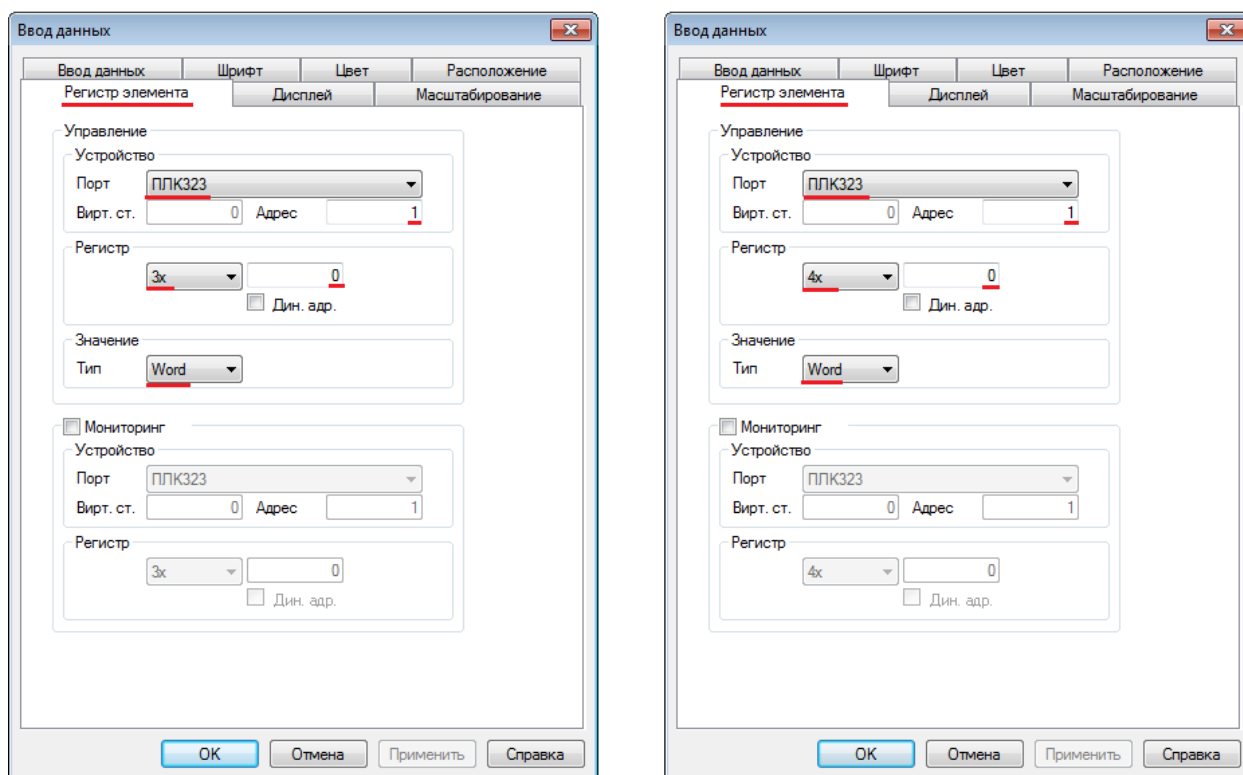


Рис. 18. Настройки элемента **Цифровой ввод**, вкладка **Регистр элемента**

На вкладке **Дисплей** выберем формат **Unsigned** (беззнаковый). Для параметра **Разрядность/Всего знаков** зададим значение **5**, поскольку переменные типа **WORD** не способны принимать шестизначные значения.



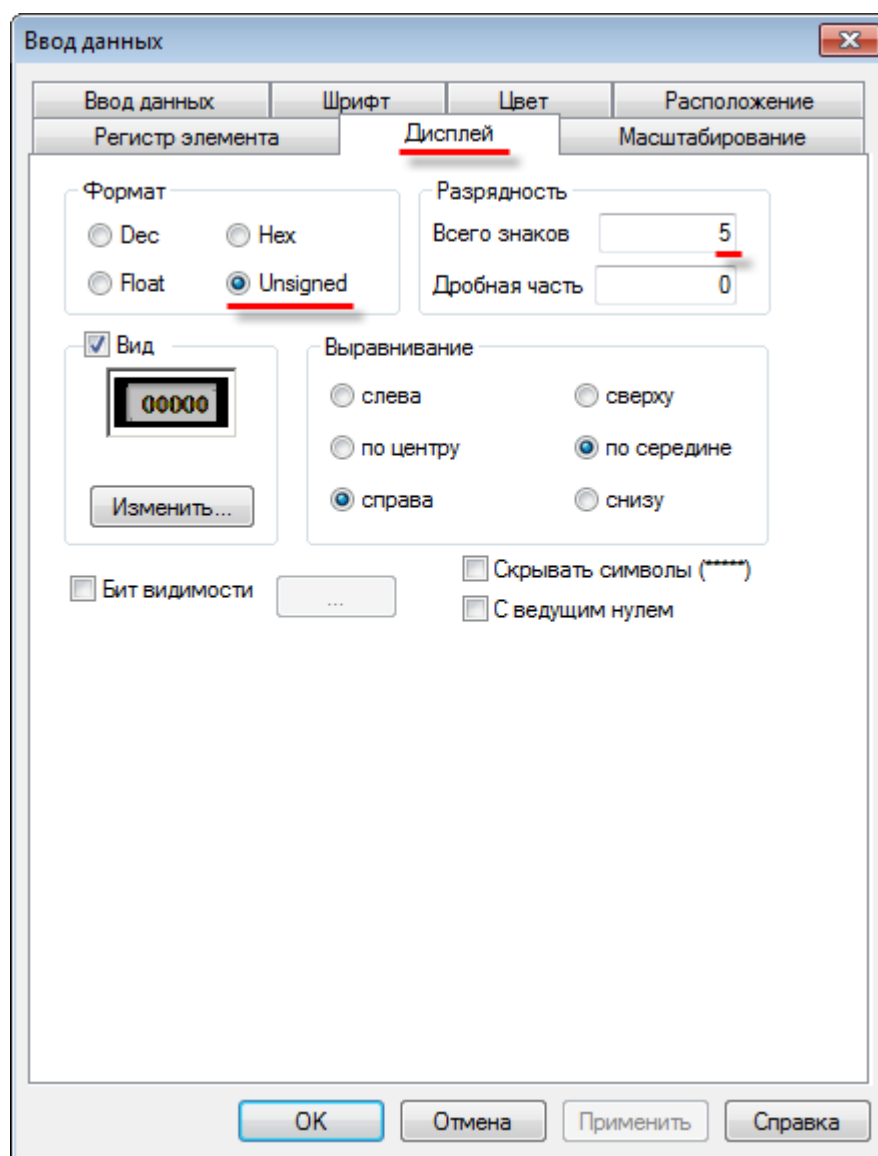


Рис. 17. Настройки элемента **Цифровой ввод**, вкладка **Дисплей**

**Обратите внимание**, что при необходимости работать с отрицательными целыми числами следует использовать формат **Dec**, а в ПЛК преобразовать полученное значение стандартной функцией **WORD\_TO\_INT** и записывать в переменную типа **INT**.

### 3.2.2. Цифровой ввод (REAL)

Элемент **Цифровой ввод (REAL)** используется для отображения и ввода значений переменных ПЛК типа **REAL**. Переменные этого типа используются для работы с [числами с плавающей точкой](#).



Рис. 18. Внешний вид элементов **Цифровой ввод**

В настройках элементов на вкладке **Регистр элемента** выберем устройство **ПЛК323**, укажем для него адрес **1** и регистры **3x1** (чтение) и **4x1** (запись) [согласно [табл. 2](#)] и тип **DWORD** (поскольку переменные типа **REAL** занимают два регистра). В результате, элемент будет обращаться к 2 и 3 регистру ПЛК.

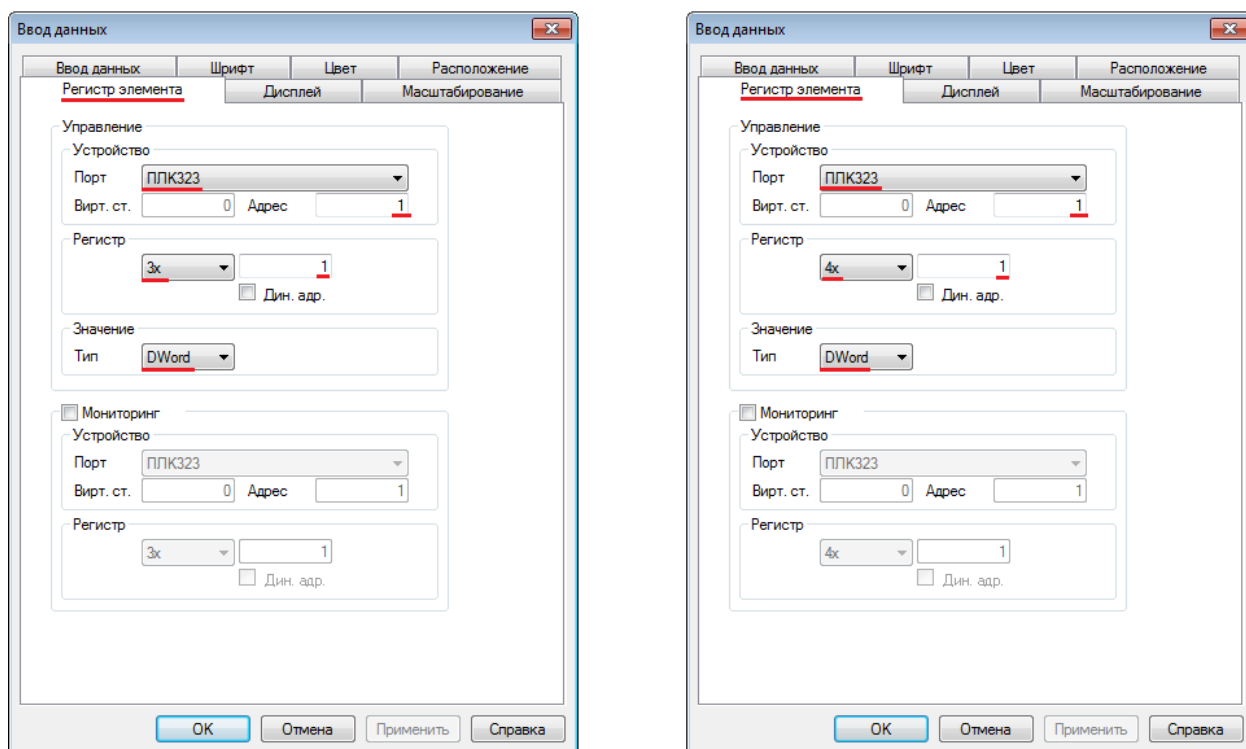


Рис. 19. Настройки элемента **Цифровой ввод**, вкладка **Регистр элемента**

На вкладке **Дисплей** выберем формат **Float** (с плавающей точкой). Для параметров **Разрядность/Всего знаков** и **Разрядность/Дробная часть** зададим значения **4** и **2**, т.е. элемент сможет отображать значения от -99.99 до 99.99.

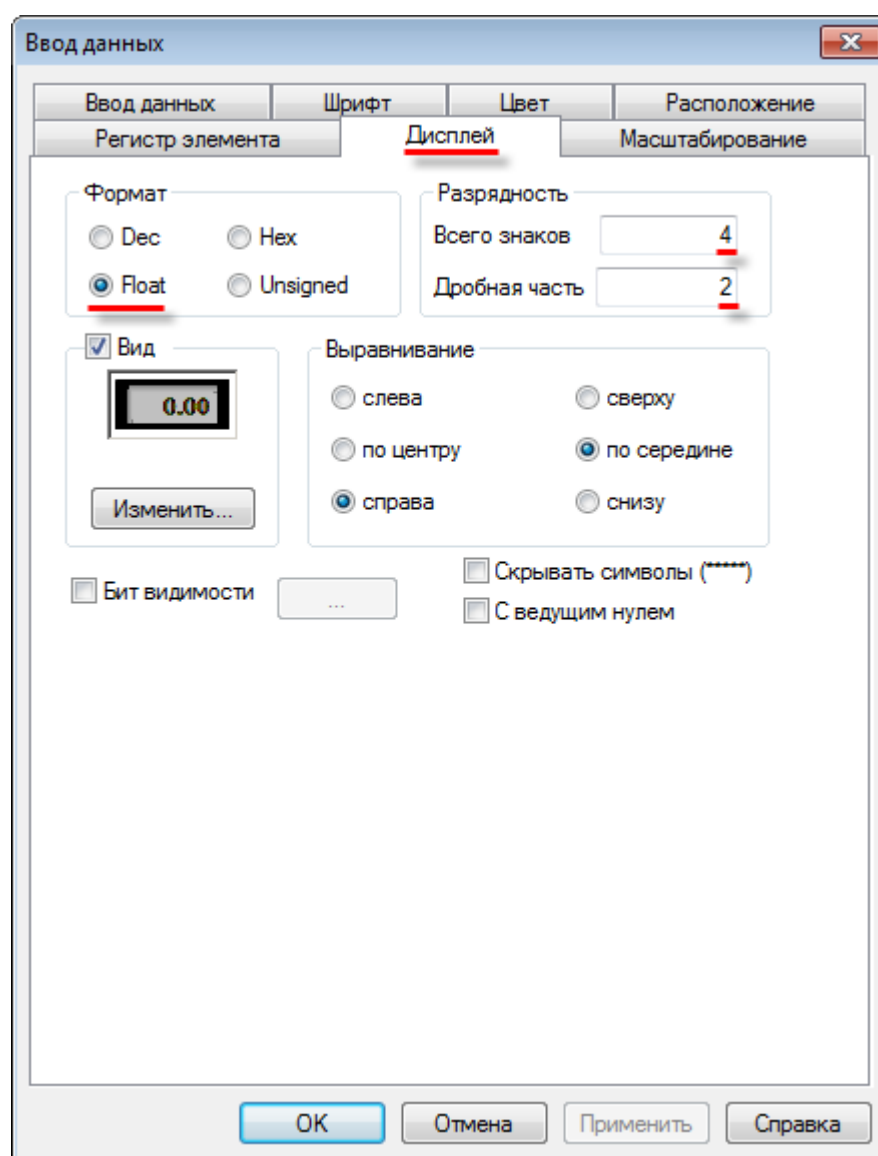


Рис. 20. Настройки элемента **Цифровой ввод**, вкладка **Дисплей**

### 3.2.3. Текстовый ввод (STRING)

Элемент **Текстовый ввод** используется для отображения и ввода значений переменных ПЛК типа **STRING**. Переменные этого типа представляют собой текстовые строки.



Рис. 21. Внешний вид элементов **Текстовый ввод**

В настройках элементов на вкладке **Регистр элемента** выберем устройство **ПЛК323**, укажем для него адрес **1**, регистры **3х3** (чтение) и **4х3** (запись), а также кол-во регистров **3**. В результате, элемент будет обращаться к трем байтам ПЛК, начиная с регистра **4**. Так как регистр ПЛК занимает два байта, то элемент будет обращаться к регистрам 4 и 5. Каждый байт может содержать два символа, поэтому с помощью элемента будет осуществляться отображение и ввод текстовых строк длиной до **6** символов.

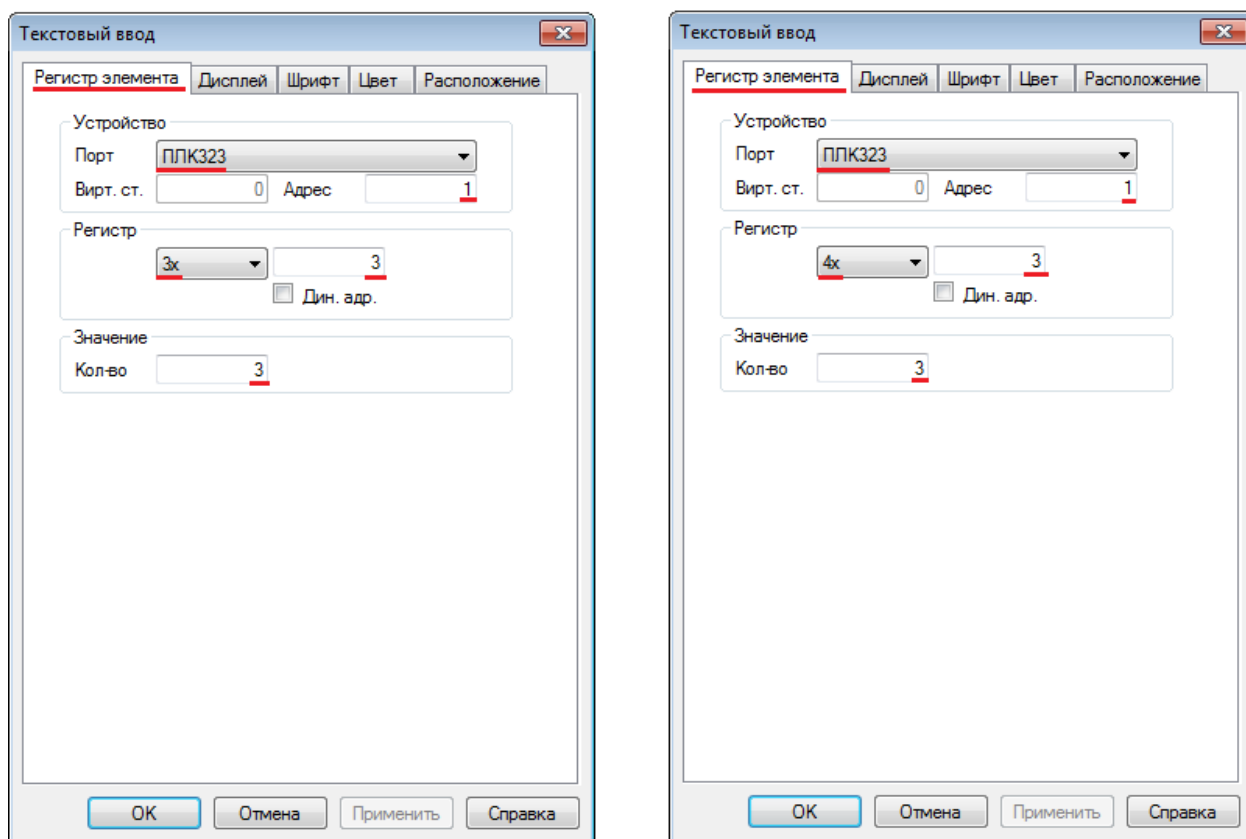


Рис. 22. Настройки элемента **Текстовый ввод**, вкладка **Регистр элемента**

**Обратите внимание**, что при необходимости ввода с дисплея панели символов кириллицы следует на вкладке **Дисплей** в параметре **Клавиатура** выбрать клавиатуру **60043**.

## 4. Работа с примером

1. Откройте проект **ModbusTCPslave.projectarchive** в **CODESYS**, подключитесь к ПЛК, загрузите проект и запустите его:

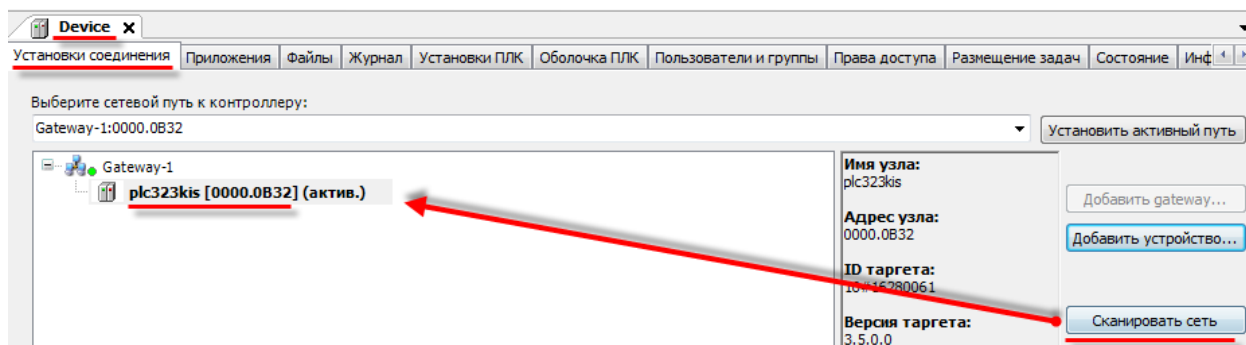


Рис. 23. Сканирование сети

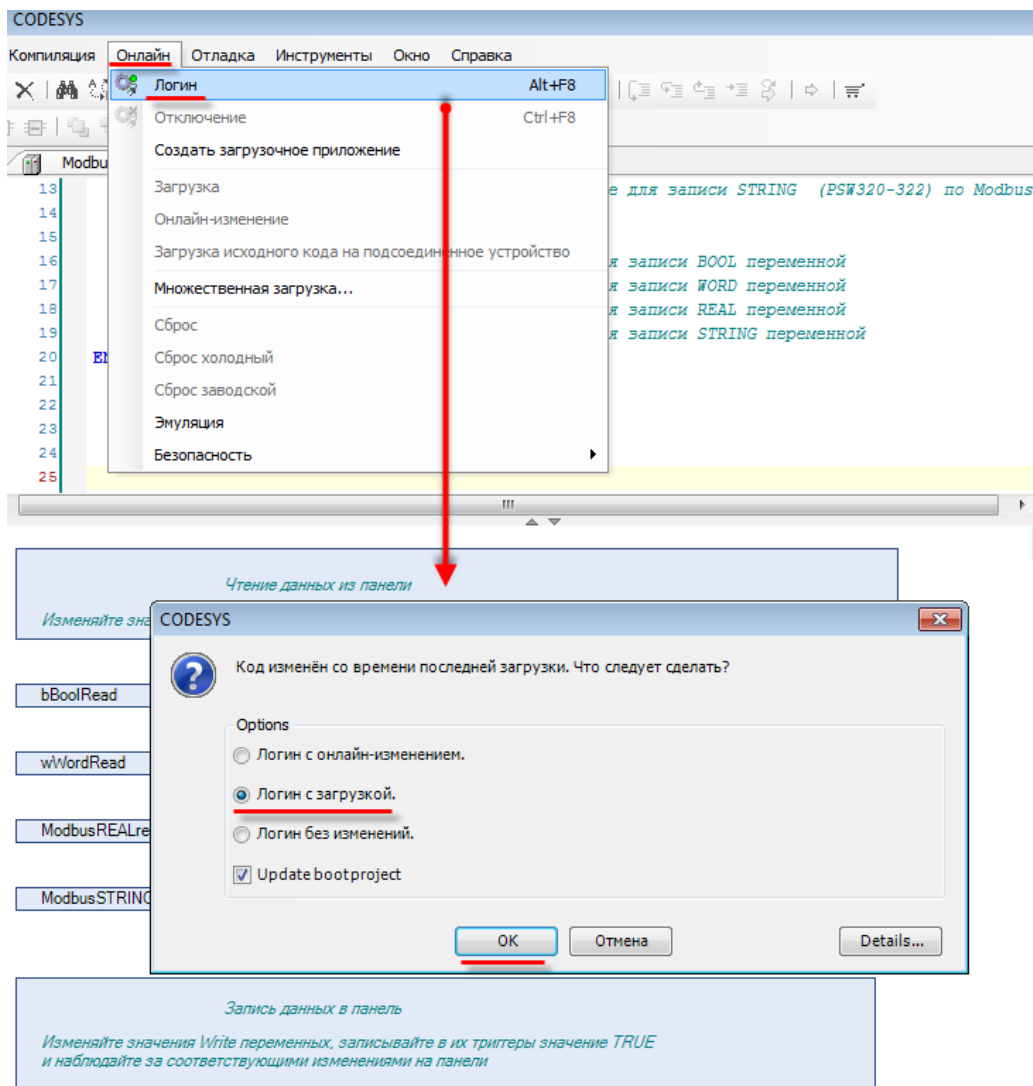


Рис. 24. Загрузка и запуск проекта для ПЛК в **CODESYS**

2. Откройте **ModbusTCPmaster.txp** в программе **Конфигуратор СП300** и загрузите его в панель:

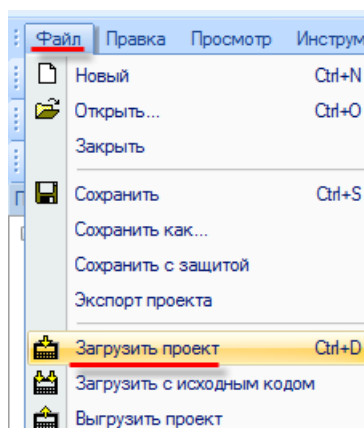


Рис. 25. Загрузка проекта в панель

3. соедините Ethernet-порты панели и ПЛК с помощью Ethernet-кабеля:



Рис. 26. Внешний вид Ethernet-кабеля

4. введите данные с помощью сенсорного дисплея панели и наблюдайте, как они будут записаны в переменные программы **PLC\_PRG**:



Рис. 27. Ввод значений с помощью сенсорного дисплея панели

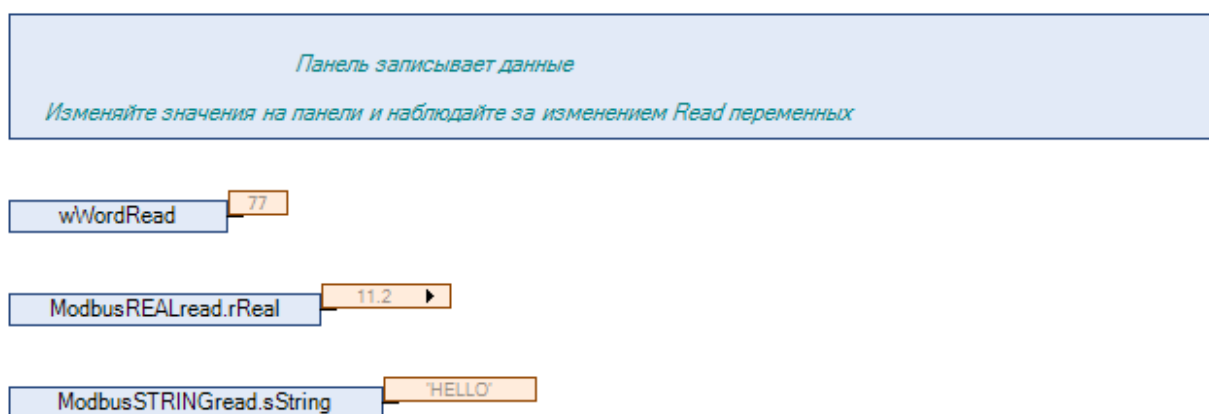


Рис. 28. Отображение введенных на панели значений в **CODESYS**

5. введите новые значения переменных в программе **PLC\_PRG**, запишите их с помощью комбинации клавиш **Ctrl+F7** и наблюдайте, как они будут считаны панелью и отображены на ее дисплее:

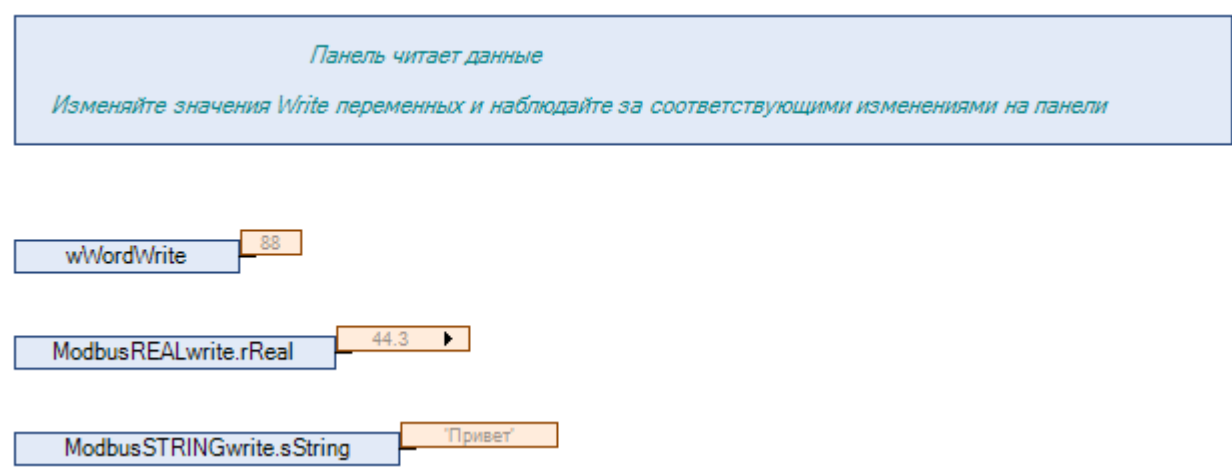


Рис. 29. Изменение значений переменных в **CODESYS**



Рис. 30. Отображение введенных в **CODESYS** значений на панели