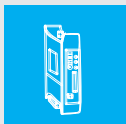
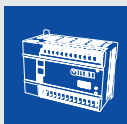
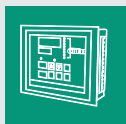


СПЗхх

Примеры настройки обмена

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

Версия: 1.0
Дата: 06.11.2015



Оглавление

1. Описание примера	3
2. Настройки СП307-Р (Modbus TCP Slave)	4
2.1. Сетевые настройки панели	4
2.2. Экран проекта	5
2.2.1. Переключатель с индикацией (BOOL)	6
2.2.2. Цифровой ввод (WORD)	7
2.2.3. Цифровой ввод (REAL)	9
2.2.4. Текстовый ввод	11
3. Настройки ПЛК323 (Modbus TCP Master)	12
3.1. Сетевые настройки контроллера	12
3.2. Создание объединений для REAL и STRING	13
3.3. Программа PLC_PRG	15
3.4. Добавление и настройка компонентов Modbus	17
3.5. Конфигурация задач	22
4. Работа с примером	23

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 **Master**, а панель – **Slave**.

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	Master	Slave
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	ModbusTCPmaster.projectarchive	ModbusTCPslave.txp

Табл. 2. Список используемых в примере переменных

ПЛК323			СП307-Р
Переменные для чтения из панели	Переменные для записи в панель	Тип данных	Регистр
bBoolRead	bBoolWrite	BOOL	PSB300
wWordRead	wWordWrite	WORD	PSW300
ModbusREALread.rReal	ModbusREALwrite.rReal	REAL	PSW310-311
ModbusSTRINGread.sString	ModbusSTRINGwrite.sString	STRING	PSW320-322

2. Настройки СП307-Р (Modbus TCP Slave)

2.1. Сетевые настройки панели

Для того чтобы использовать панель **СП307-Р** в режиме **Modbus TCP Slave**, 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**.

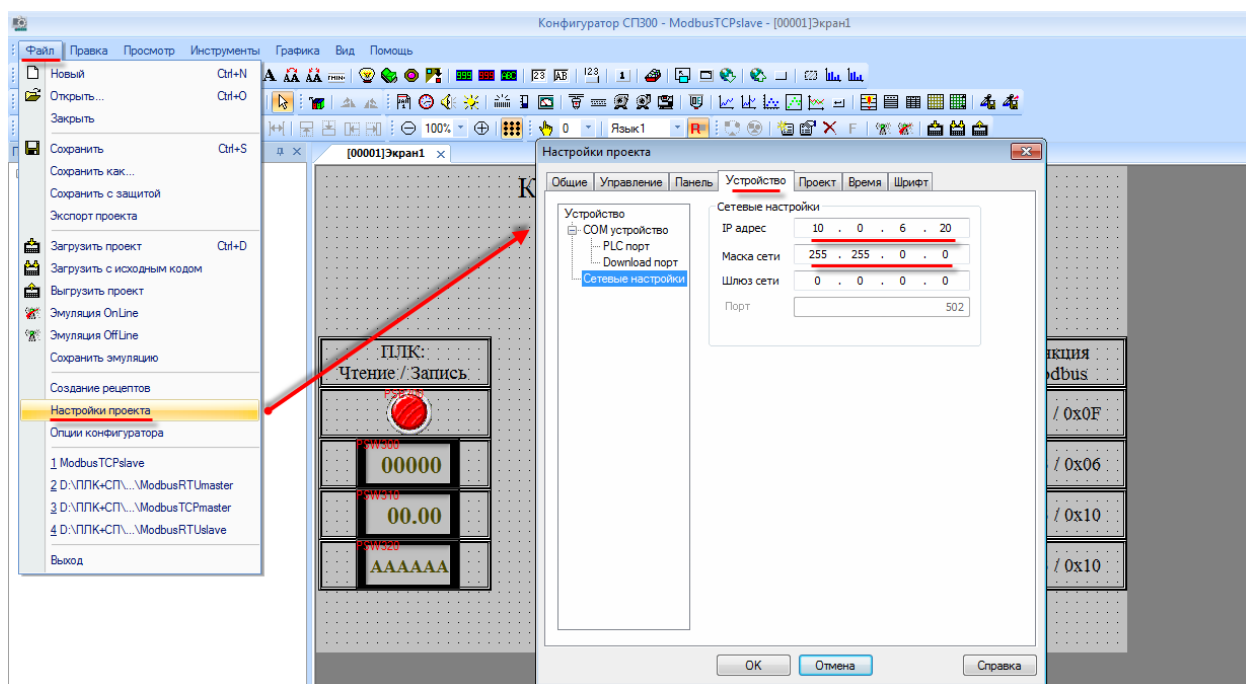


Рис. 1. Сетевые настройки панели (режим **Modbus TCP Slave**)

2.2. Экран проекта

Проект содержит один экран, на котором осуществляется отображение данных, которые записывает ПЛК, и ввод данных, которые ПЛК считывает; при этом и отображение, и ввод для каждого типа данных реализованы через один элемент.

Соответственно, экран содержит четыре активных элемента:

1. **Переключатель с индикацией** с привязанным регистром **PSB300**. ПЛК считывает значение этого регистра в переменную **bBoolRead** и по команде пользователя (по переднему фронту переменной **bBoolWriteTrigger**) записывает в него значение переменной **bBoolWrite**. Обе переменные имеют тип **BOOL**.
2. **Цифровой ввод** с привязанным регистром **PSW300**. ПЛК считывает значение этого регистра в переменную **wWordRead** и по команде пользователя (по переднему фронту переменной **bWordWriteTrigger**) записывает в него значение переменной **wWordWrite**. Обе переменные имеют тип **WORD**.
3. **Цифровой ввод** с привязанными регистрами **PSW310-311**. ПЛК считывает значения этих регистров в переменную **ModbusREALread.rReal** и по команде пользователя (по переднему фронту переменной **bRealWriteTrigger**) записывает в них значение переменной **ModbusREALwrite.rReal**. Обе переменные имеют тип **REAL**.
4. **Текстовый ввод** с привязанными регистрами **PSW320-322**. ПЛК считывает значения этих регистров в переменную **ModbusSTRINGread.sString** и по команде пользователя (по переднему фронту переменной **bStringWriteTrigger**) записывает в них значение переменной **ModbusSTRINGwrite.sString**. Обе переменные имеют тип **STRING**.



Рис. 2. Внешний вид экрана проекта

Настройки элементов приведены ниже.

2.2.1. Переключатель с индикацией (BOOL)

Элемент **Переключатель с индикацией** используется для отображения и ввода значений переменных ПЛК типа **BOOL**. Переменные этого типа в каждый момент времени находятся в одном из двух возможных состояний: **TRUE** или **FALSE** (иными словами, **Включено/Выключено**).



Рис. 3. Внешний вид элемента **Переключатель с индикацией**

В настройках элемента на вкладке **Регистр элемента** выберем регистр **PSB300** (согласно [табл. 2](#)).

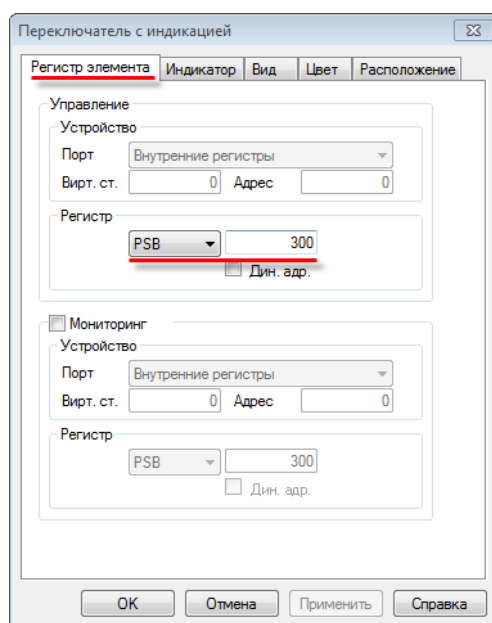


Рис. 4. Настройки элемента **Переключатель с индикацией**, вкладка **Регистр элемента**

На вкладке **Индикатор** для настройки **Переключение бита в состояние** выберем значение **Инверсия**. Это нужно для того, чтобы свободно менять значение элемента с дисплея панели.

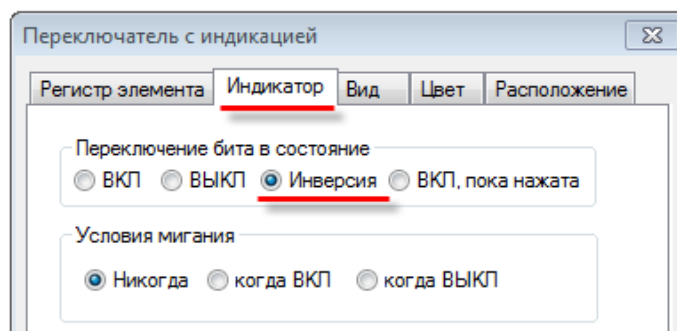


Рис. 5. Настройки элемента **Переключатель с индикацией**, вкладка **Индикатор**

2.2.2. 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.

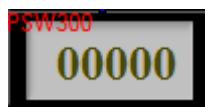


Рис. 6. Внешний вид элемента **Цифровой ввод**

В настройках элемента на вкладке **Регистр элемента** выберем регистр **PSW300** (согласно [табл. 2](#)). **Обратите внимание**, что параметр **Тип** должен иметь значение **WORD**.

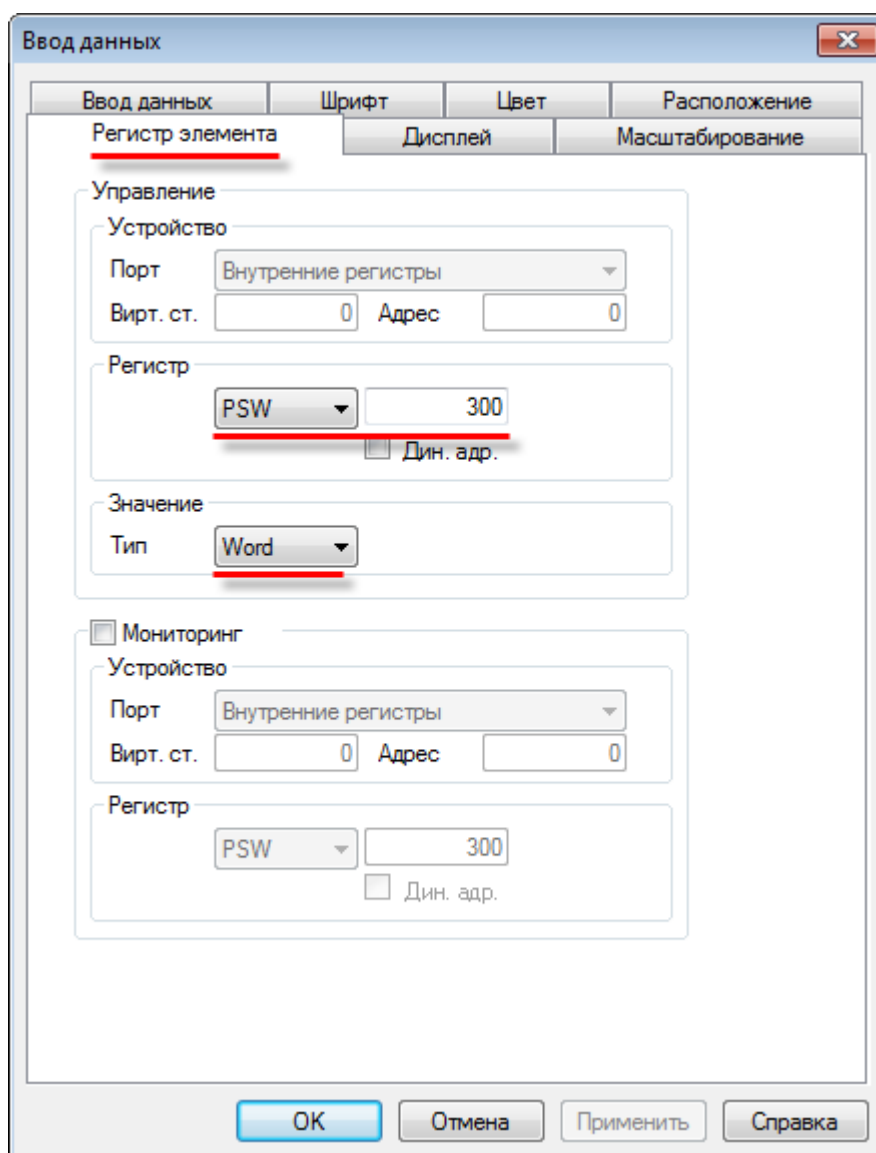


Рис. 7. Настройки элемента **Цифровой ввод**, вкладка **Регистр элемента**

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

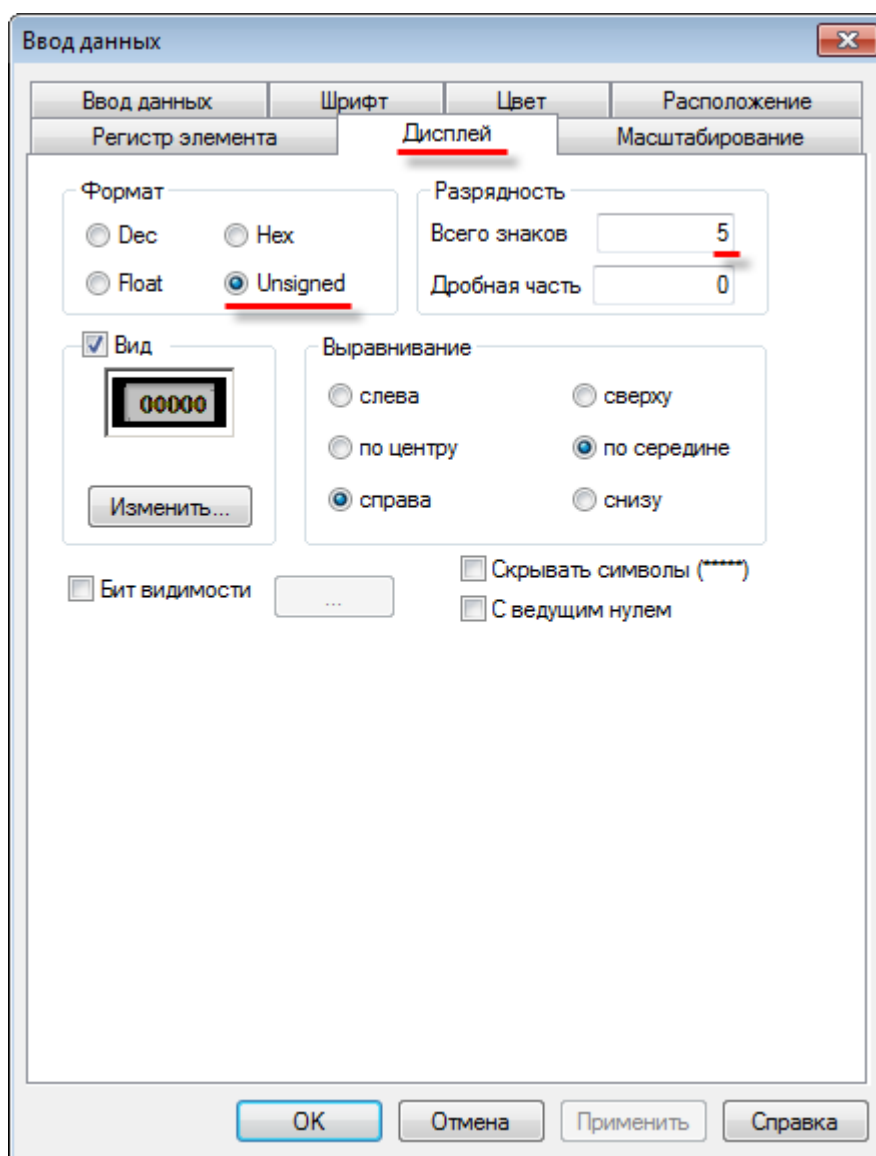


Рис. 8. Настройки элемента **Цифровой ввод**, вкладка **Дисплей**

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**.

2.2.3. 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. 9. Внешний вид элемента **Цифровой ввод**

В настройках элемента на вкладке **Регистр элемента** выберем регистр **PSW310** и тип **DWORD** (поскольку переменные типа **REAL** занимают два регистра). В результате, данные элемента будут храниться в регистрах **PSW310-311** (согласно [табл. 2](#)).

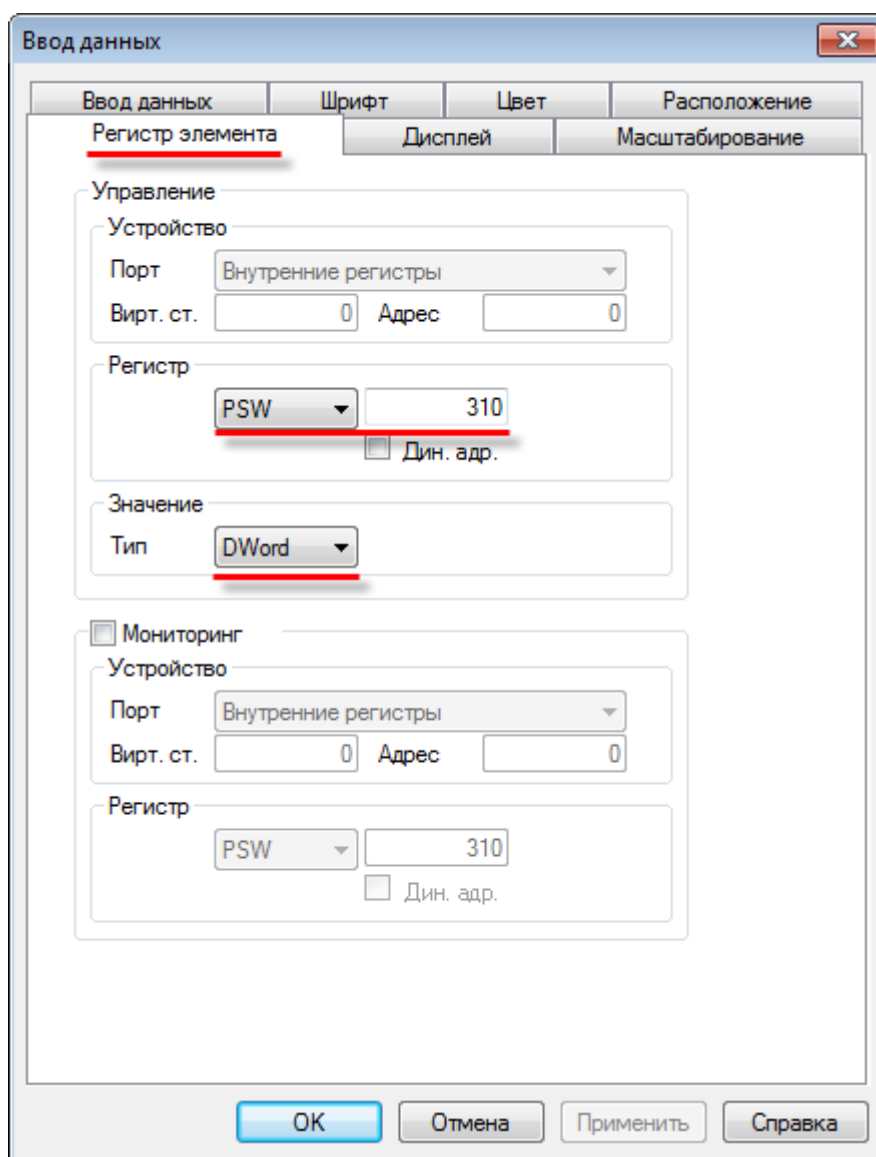


Рис. 10. Настройки элемента **Цифровой ввод**, вкладка **Регистр элемента**

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.

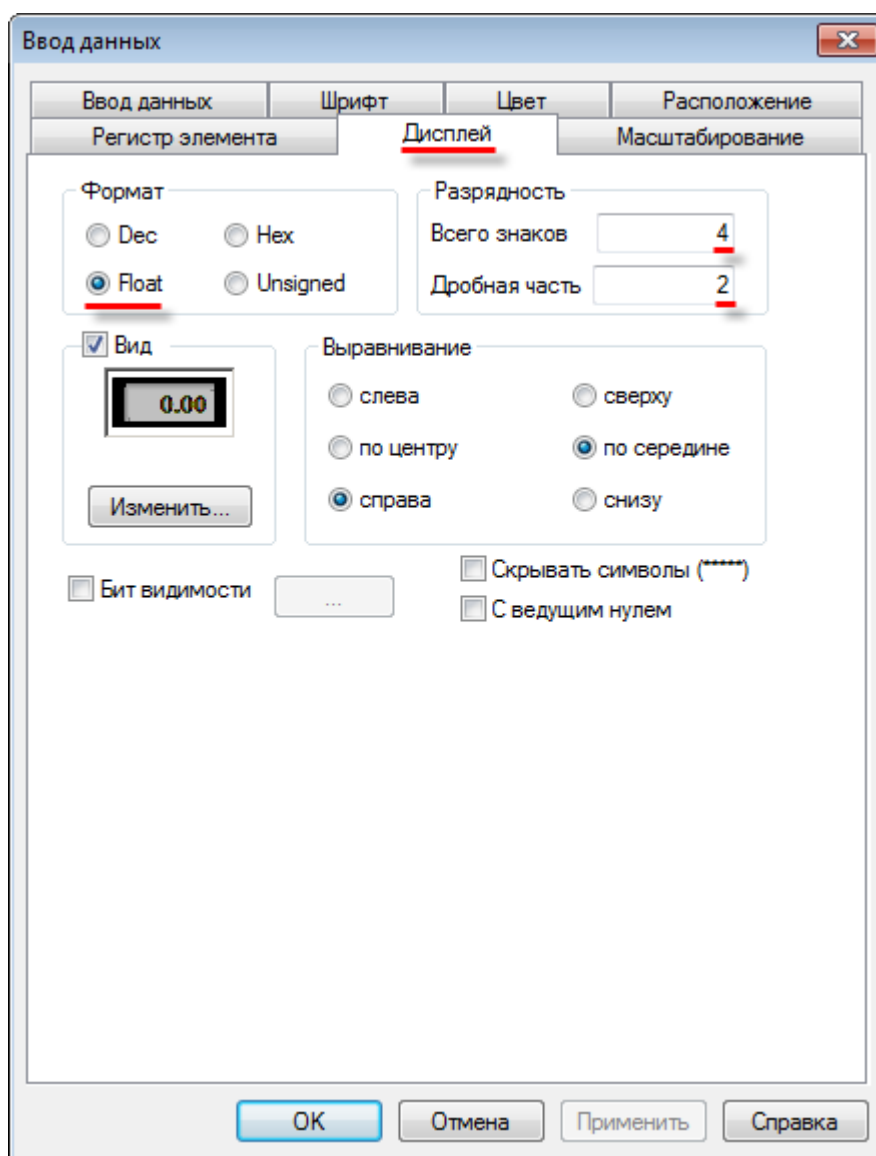


Рис. 11. Настройки элемента **Цифровой ввод**, вкладка **Дисплей**

2.2.4. Текстовый ввод (STRING)

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



Рис. 12. Внешний вид элемента **Текстовый ввод**

В настройках элемента на вкладке **Регистр элемента** выберем регистр **PSW320** и кол-во регистров **3**. В результате, данные элемента будут храниться в регистрах **PSW320-322** (согласно [табл. 2](#)). Каждый регистр может содержать два символа, поэтому с помощью элемента будет осуществляться отображение и ввод текстовых строк длиной до **6** символов.

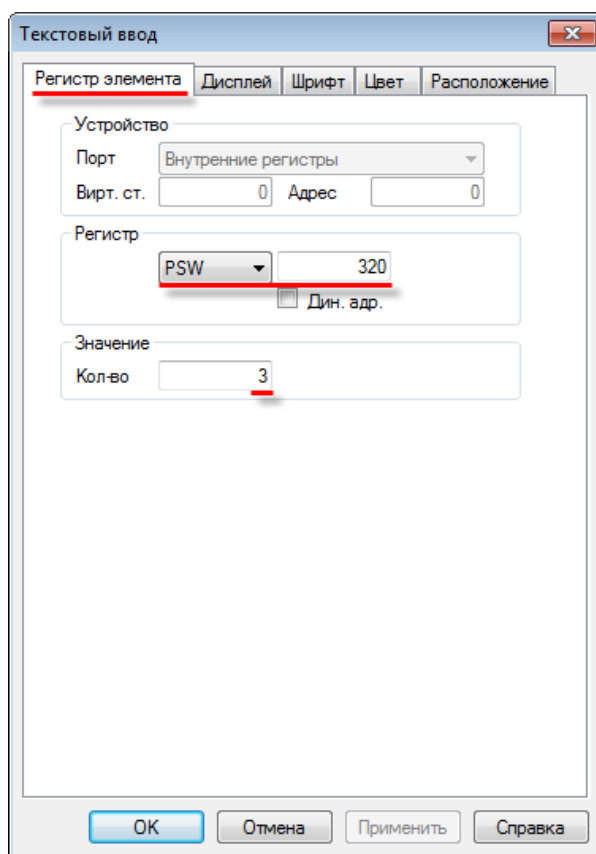


Рис. 13. Настройки элемента **Текстовый ввод**, вкладка **Регистр элемента**

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

3. Настройки ПЛК323 (Modbus TCP Master)

3.1. 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 displaying '10.0.6.10'. The browser's address bar also shows 'Сервисы' and 'Webvisu'. The main title of the page is 'Настройки ПЛК 323' with a version number 'ver. 1.10.1' in the top right corner. On the left side, there is a navigation menu with links: 'Дата/Время', 'Сеть', 'Безопасность', 'Опции ПЛК', and 'Утилиты'. The 'Сеть' link is currently selected. The main content area is titled 'Сетевые настройки - Основное' and contains the following settings:

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, there are two buttons: 'OK' and 'Сброс'.

Рис. 14. Внешний вид **web-конфигуратора** ПЛК323

3.2. 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**:

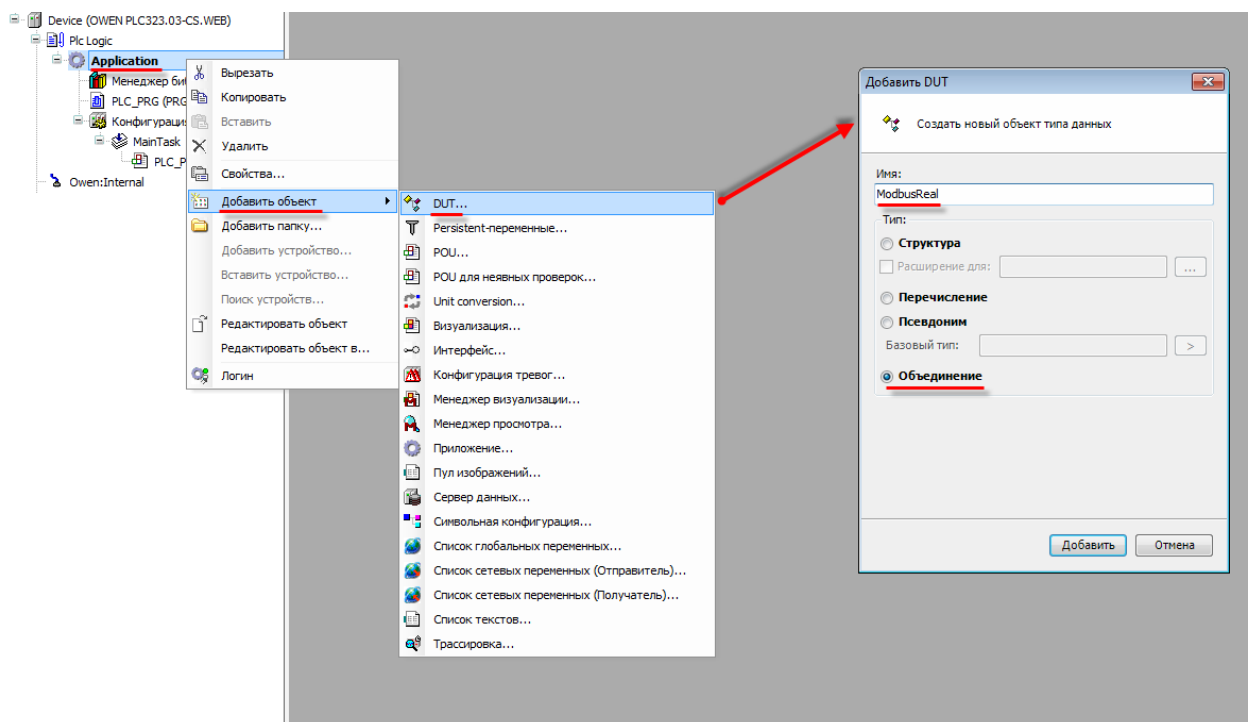


Рис. 15. Добавление объединений в проект

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. 16. 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. 17. Объявление переменных объединения **ModbusSTRING**

3.3. Программа PLC_PRG

Объявим в программе **PLC_PRG** следующие переменные:

```
1  PROGRAM PLC_PRG
2  VAR
3      bBoolRead      :BOOL;      // чтение BOOL из регистра панели PSB300
4      bBoolWrite     :BOOL;      // запись BOOL в регистр панели PSB300
5
6      wWordRead       :WORD;      // чтение WORD из регистра панели PSW300
7      wWordWrite      :WORD;      // запись WORD в регистр панели PSW300
8
9      ModbusREALread  :ModbusReal; // объединение для чтения REAL (PSW310-311) по Modbus
10     ModbusSTRINGread :ModbusString; // объединение для записи REAL (PSW310-311) по Modbus
11
12     ModbusREALwrite  :ModbusReal; // объединение для чтения STRING (PSW320-322) по Modbus
13     ModbusSTRINGwrite :ModbusString; // объединение для записи STRING (PSW320-322) по Modbus
14
15
16     bBoolWriteTrigger :BOOL;      // триггер для записи BOOL переменной
17     bWordWriteTrigger :BOOL;      // триггер для записи WORD переменной
18     bRealWriteTrigger :BOOL;      // триггер для записи REAL переменной
19     bStringWriteTrigger :BOOL;    // триггер для записи STRING переменной
20 END_VAR
```

Рис. 18. Объявление переменных в программе **PLC_PRG**

Read и **Write** переменные будут использоваться, соответственно, для чтения/записи данных в панель. Запись значений **Write** переменных в панель происходит по переднему фронту соответствующих **Trigger** переменных. Такая реализация позволяет использовать один элемент панели как для передачи данных, введенных с ее сенсорного дисплея, в ПЛК, так и отображения полученных из ПЛК данных. **Обратите внимание**, что такой подход использован только в рамках примера, и в значительном числе ситуаций проще использовать для отображения и записи данных отдельные элементы.

Код программы **PLC_PRG**, написанной на языке **CFC**, выглядит следующим образом:

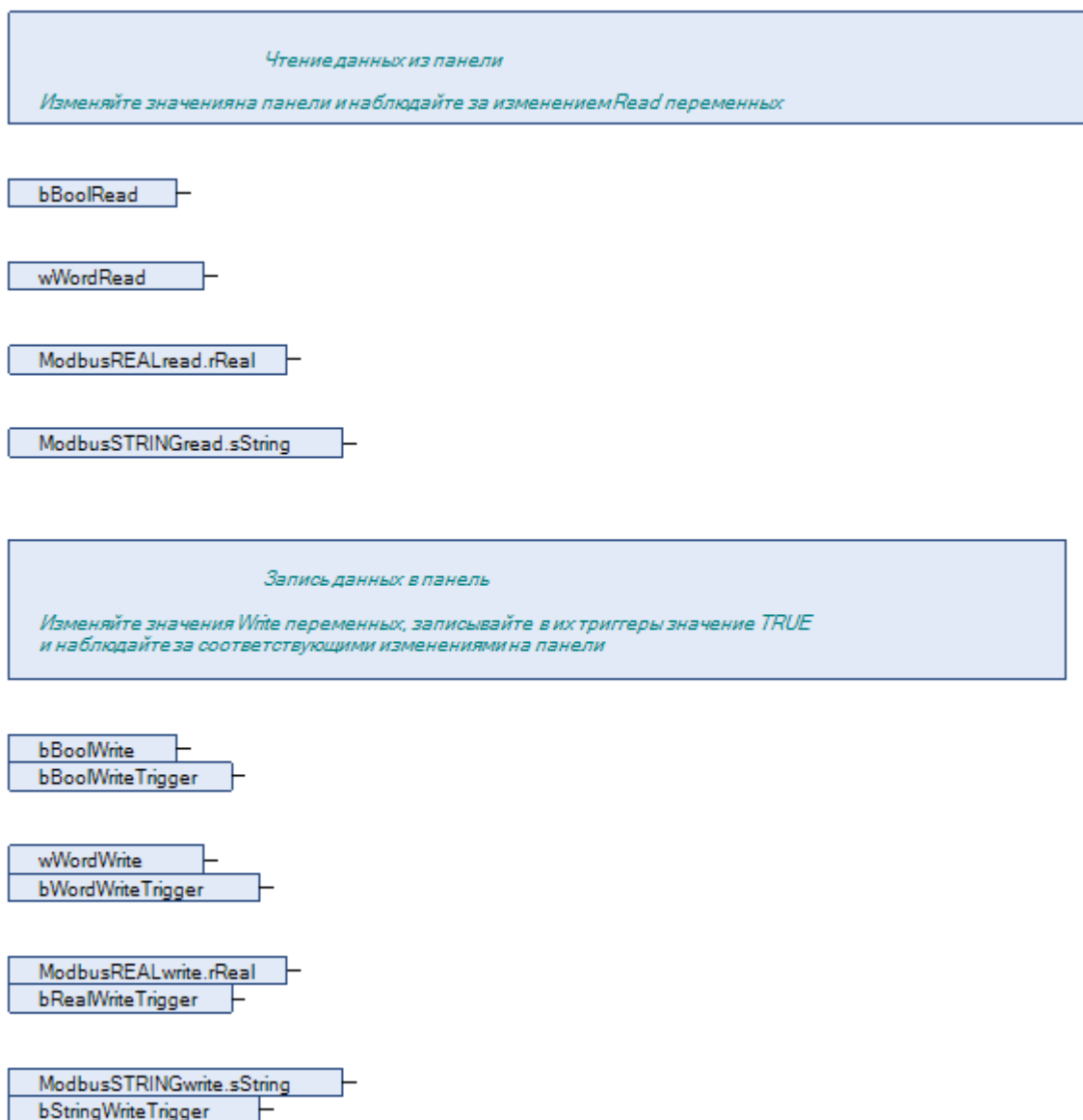


Рис. 19. Код программы **PLC_PRG**

В процессе работы проекта пользователь может изменять значения **Write** переменных (непосредственно в **CODESYS**) и записывать их в панель по переднему фронту соответствующих **Trigger** переменных, наблюдая за изменениями на дисплее панели, а также изменять данные на панели (с помощью сенсорного ввода) и наблюдать изменения у соответствующих **Read** переменных.

3.4. 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**.

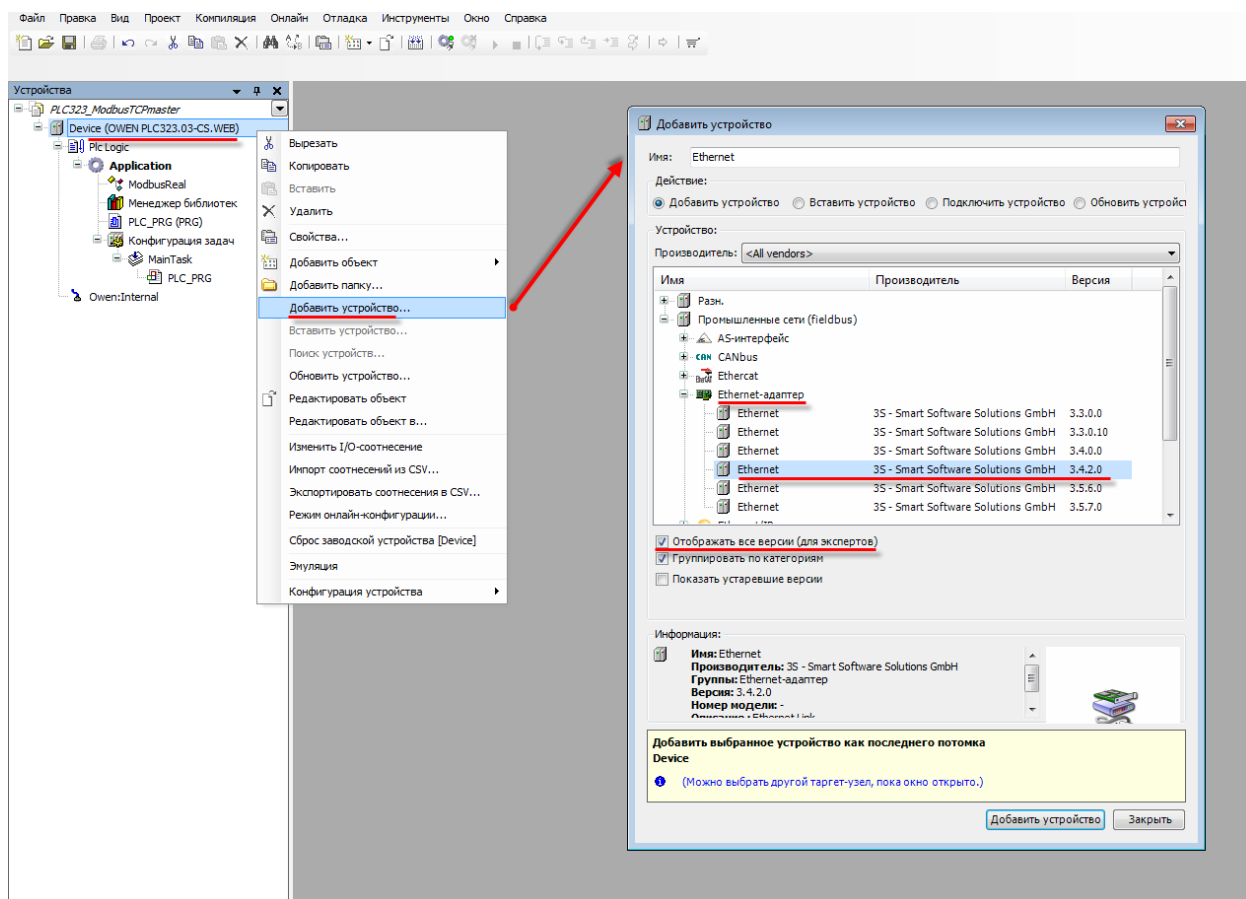


Рис. 20. Добавление компонента **Ethernet**

Этот компонент не нуждается в настройках.

Нажмем **ПКМ** на компонент **Ethernet** и добавим компонент **Modbus TCP Master**. **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**.

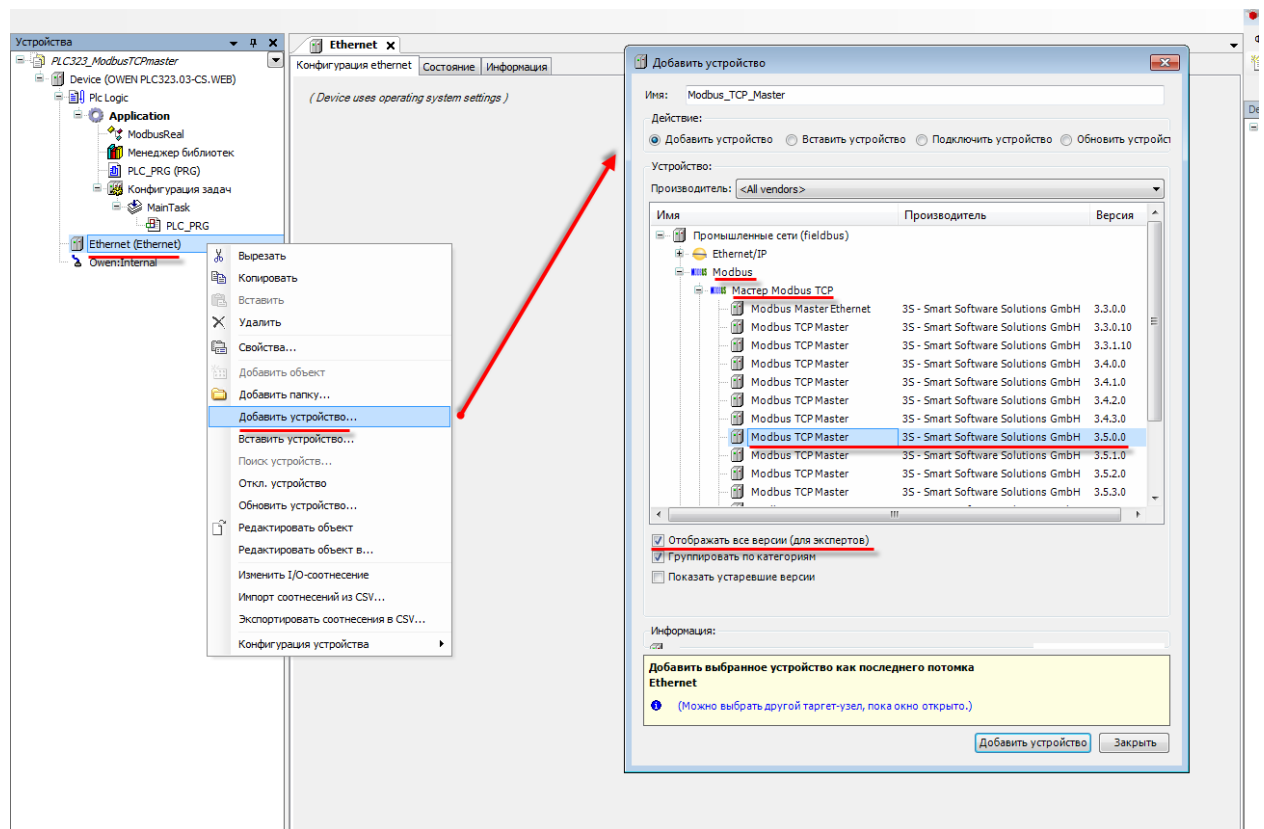


Рис. 21. Добавление компонента **Modbus TCP Master**

На вкладке **Конфигурация ModbusTCP Master** поставим галочку **Автопереподключение**:

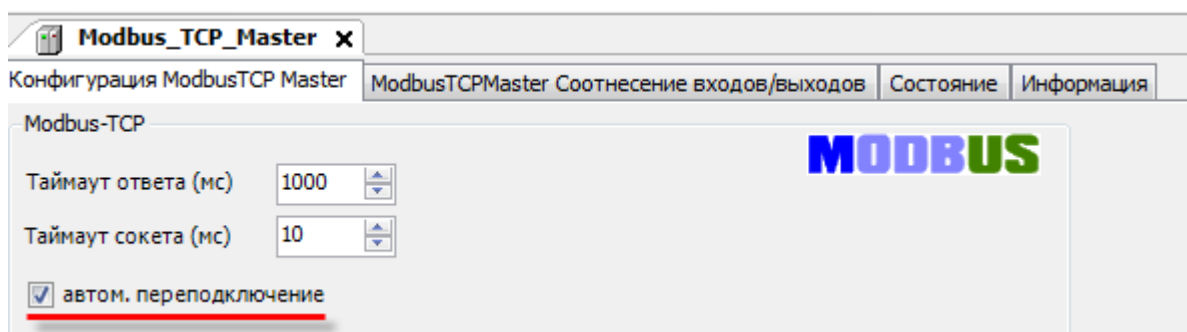


Рис. 22. Настройки вкладки **Конфигурация ModbusTCP Master**

Нажмем **ПКМ** на компонент **Modbus TCP Master** 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**.

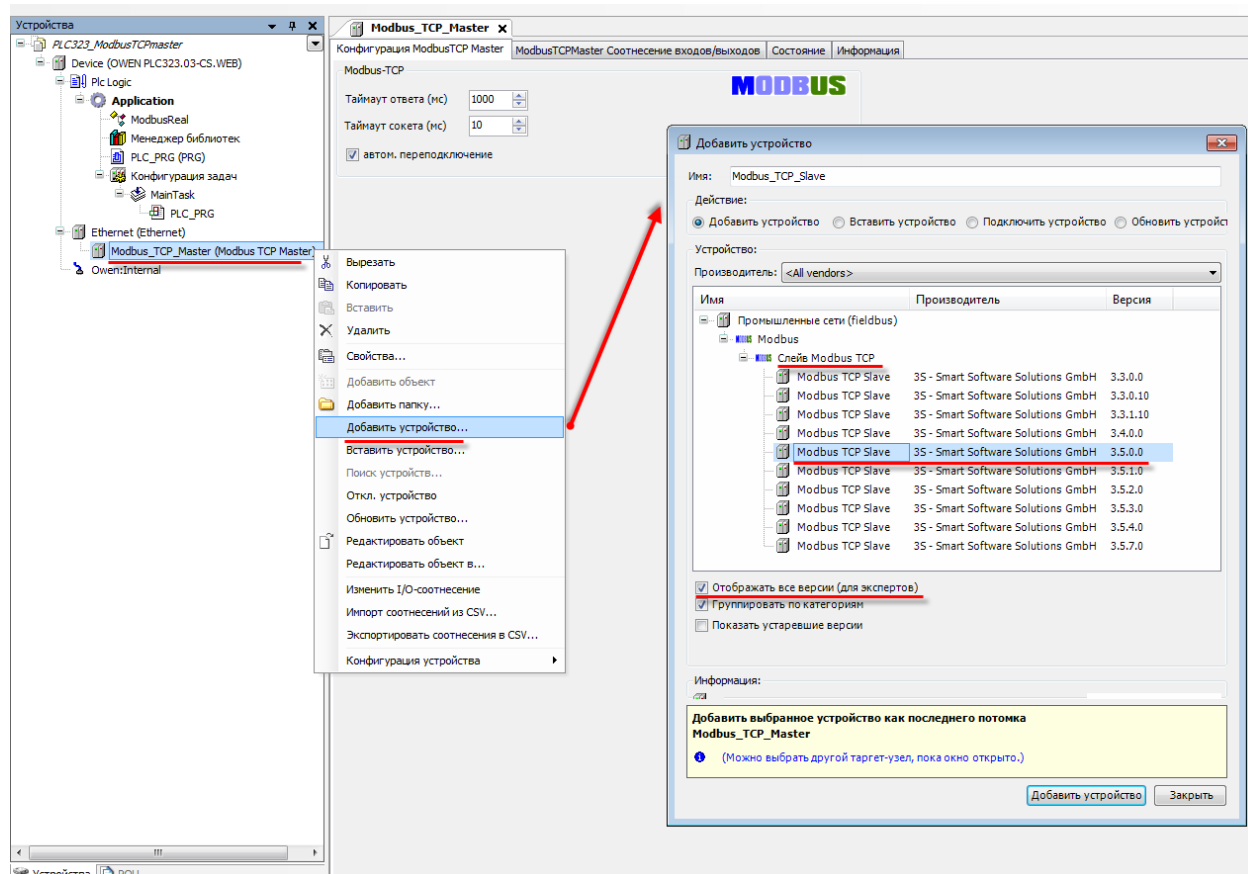


Рис. 23. Добавление компонента **Modbus TCP Slave**

На вкладке **ModbusTCP Slave** укажем **IP-адрес** панели (**10.0.6.20** согласно [табл. 1](#)). Поле **Unit ID** оставим пустым, поскольку панели не задается **Slave ID**.



Рис. 24. Настройки вкладки **ModbusTCP Slave**

На вкладке **Канал Modbus Slave** создадим необходимые нам каналы. Число каналов будет совпадать с общим числом считываемых/записываемых переменных.

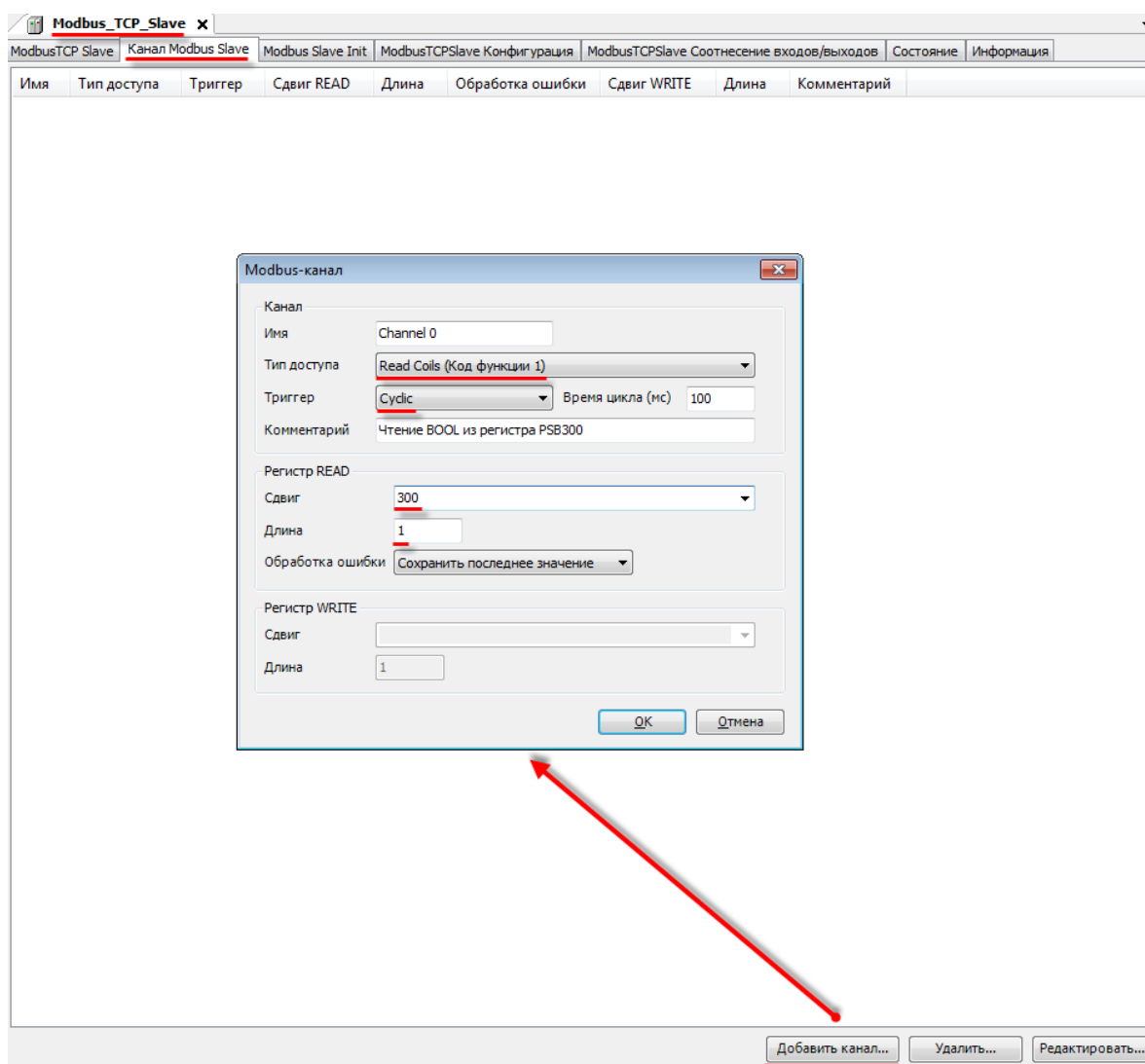


Рис. 25. Добавление каналов для **Modbus TCP Slave**

Тип доступа определяет функцию **Modbus**, применяемую к заданным регистрам.

Триггер определяет, каким образом производится работа с регистрами – циклически или по фронту логической переменной. В данном примере ПЛК циклически считывает значения из панели в **Read** переменные, и записывает **Write** переменные в панель по переднему фронту **Trigger** переменных.

Сдвиг представляет собой номер первого из группы регистров, к которым применяется заданная функция. Номера используемых регистров приведены в [табл. 2](#). **Обратите внимание**, что после добавления канала номер регистра будет отображаться в шестнадцатеричном формате.

Длина представляет собой количество регистров, к которым применяется заданная функция. Переменные **BOOL** и **WORD** занимают один регистр, **REAL** – 2, **STRING** (в рамках нашего примера) – 3.

После добавления всех каналов компонент должен выглядеть следующим образом:

Modbus TCP Slave x								
ModbusTCP Slave Канал Modbus Slave Modbus Slave Init ModbusTCPSlave Конфигурация ModbusTCPSlave Соотнесение входов/выходов Состояние Информация								
Имя	Тип доступа	Триггер	Сдвиг READ	Длина	Обработка ошибки	Сдвиг WRITE	Длина	Комментарий
Channel 0	Read Coils (Код функции 01)	CYCLIC, t#100ms	16#012C	1	Сохранить последнее значение			
Channel 1	Read Holding Registers (Код функции 03)	CYCLIC, t#100ms	16#012C	1	Сохранить последнее значение			
Channel 2	Read Holding Registers (Код функции 03)	CYCLIC, t#100ms	16#0136	2	Сохранить последнее значение			
Channel 3	Read Holding Registers (Код функции 03)	CYCLIC, t#100ms	16#0140	3	Сохранить последнее значение			
Channel 4	Write Single Coil (Код функции 05)	RISING_EDGE				16#012C	1	
Channel 5	Write Multiple Registers (Код функции 16)	RISING_EDGE				16#012C	1	
Channel 6	Write Multiple Registers (Код функции 16)	RISING_EDGE				16#0136	2	
Channel 7	Write Multiple Registers (Код функции 16)	RISING_EDGE				16#0140	3	

Рис. 26. Каналы Modbus TCP Slave

На вкладке **ModbusTCPSlave Соотнесение входов/выходов** привяжем к каналам переменные программы (с помощью двойного нажатия ЛКМ на ячейках столбца **Переменная**):

Modbus TCP Slave x							
ModbusTCP Slave Канал Modbus Slave Modbus Slave Init ModbusTCPSlave Конфигурация ModbusTCPSlave Соотнесение входов/выходов Состояние Информация							
Каналы							
Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание	
Application.PLC_PRG.bBoolRead		Channel 0	16#012C	BIT		Чтение BOOL из регистра PSB300	
Application.PLC_PRG.wWordRead		Channel 1	16#0136	WORD		Чтение WORD из регистра PSW300	
Application.PLC_PRG.ModbusREALread.arrReal[0]		Channel 2	16#0136	WORD		Чтение WORD из регистра PSW310 (REAL)	
Application.PLC_PRG.ModbusREALread.arrReal[1]		Channel 2	16#0138	WORD		Чтение WORD из регистра PSW311 (REAL)	
Application.PLC_PRG.ModbusSTRINGread.arrSTRING[0]		Channel 3	16#0140	WORD		Чтение WORD из регистра PSW320 (STRING)	
Application.PLC_PRG.ModbusSTRINGread.arrSTRING[1]		Channel 3	16#0142	WORD		Чтение WORD из регистра PSW321 (STRING)	
Application.PLC_PRG.ModbusSTRINGread.arrSTRING[2]		Channel 3	16#0144	WORD		Чтение WORD из регистра PSW322 (STRING)	
Application.PLC_PRG.bBoolWriteTrigger		Channel 4	16#012C	BIT		Триггер для записи BOOL	
Application.PLC_PRG.bBoolWrite		Channel 4	16#012C	BIT		Запись BOOL в регистр PSB300	
Application.PLC_PRG.bWordWriteTrigger		Channel 5	16#0136	BIT		Триггер для записи WORD	
Application.PLC_PRG.wWordWrite		Channel 5	16#0136	WORD		Запись WORD в регистр PSW300	
Application.PLC_PRG.bRealWriteTrigger		Channel 6	16#0136	BIT		Триггер для записи REAL	
Application.PLC_PRG.ModbusREALwrite.arrReal[0]		Channel 6	16#0136	WORD		Запись WORD в регистр PSW310 (REAL)	
Application.PLC_PRG.ModbusREALwrite.arrReal[1]		Channel 6	16#0138	WORD		Запись WORD в регистр PSW311 (REAL)	
Application.PLC_PRG.bStringWriteTrigger		Channel 7	16#0140	BIT		Триггер для записи STRING	
Application.PLC_PRG.ModbusSTRINGwrite.arrSTRING[0]		Channel 7	16#0140	WORD		Запись WORD в регистр PSW320 (STRING)	
Application.PLC_PRG.ModbusSTRINGwrite.arrSTRING[1]		Channel 7	16#0142	WORD		Запись WORD в регистр PSW321 (STRING)	
Application.PLC_PRG.ModbusSTRINGwrite.arrSTRING[2]		Channel 7	16#0144	WORD		Запись WORD в регистр PSW322 (STRING)	

Рис. 27. Привязка переменных к каналам

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

3.5. Конфигурация задач

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

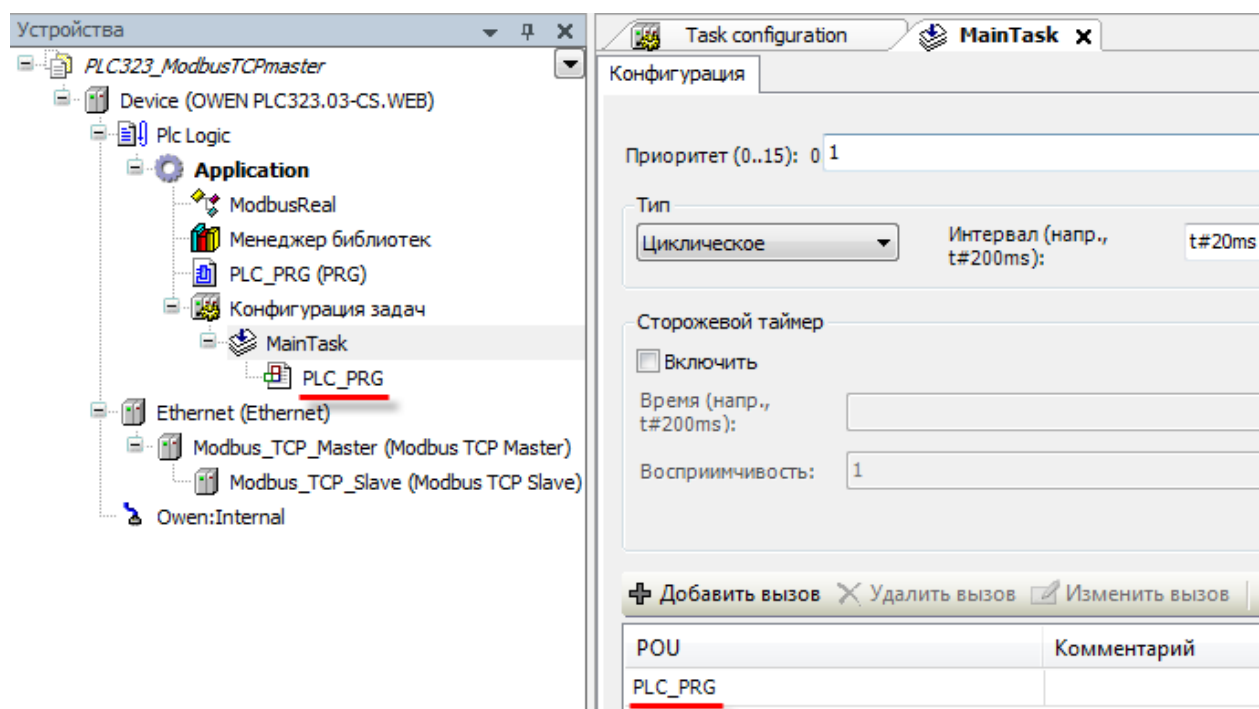


Рис. 28. Привязка программы **PLC_PRG** к задаче

4. Работа с примером

1. Откройте проект **ModbusTCPmaster.projectarchive** в **CODESYS 3.5 SP3 Patch2**, подключитесь к ПЛК, загрузите проект и запустите его:

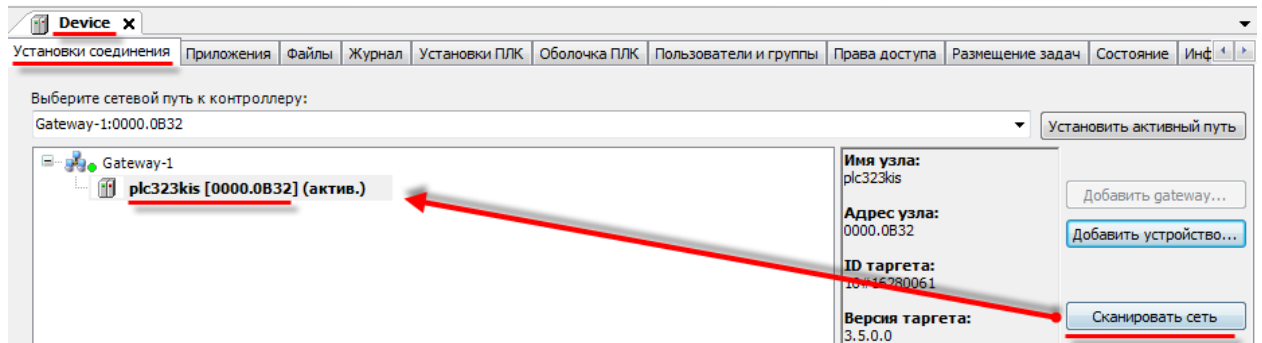


Рис. 29. Сканирование сети

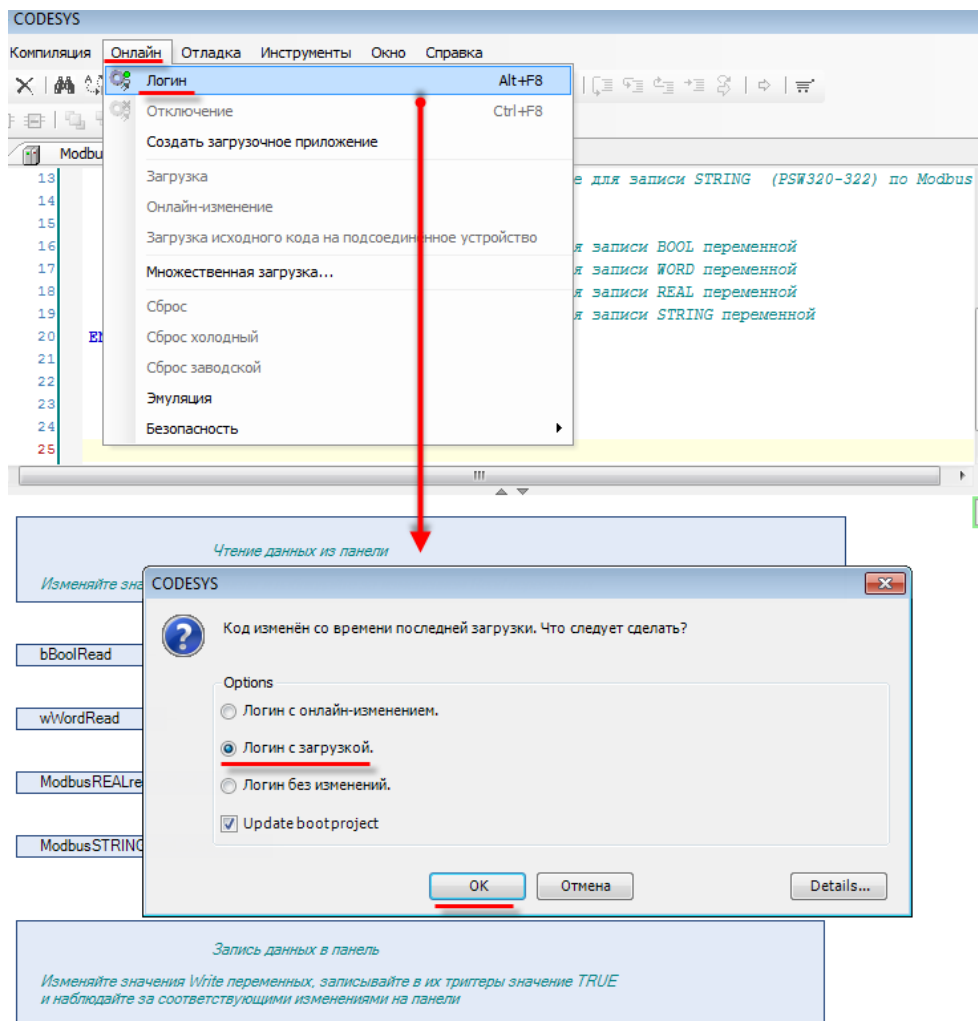


Рис. 30. Загрузка и запуск проекта для ПЛК в CODESYS

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

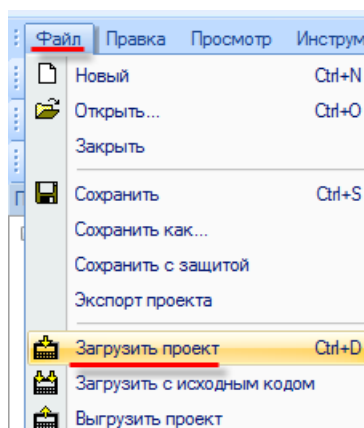


Рис. 31. Загрузка проекта в панель

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



Рис. 32. Внешний вид Ethernet-кабеля

4. введите данные с помощью сенсорного дисплея панели и наблюдайте, как они будут считаны в **Read** переменные в программе **PLC_PRG**:

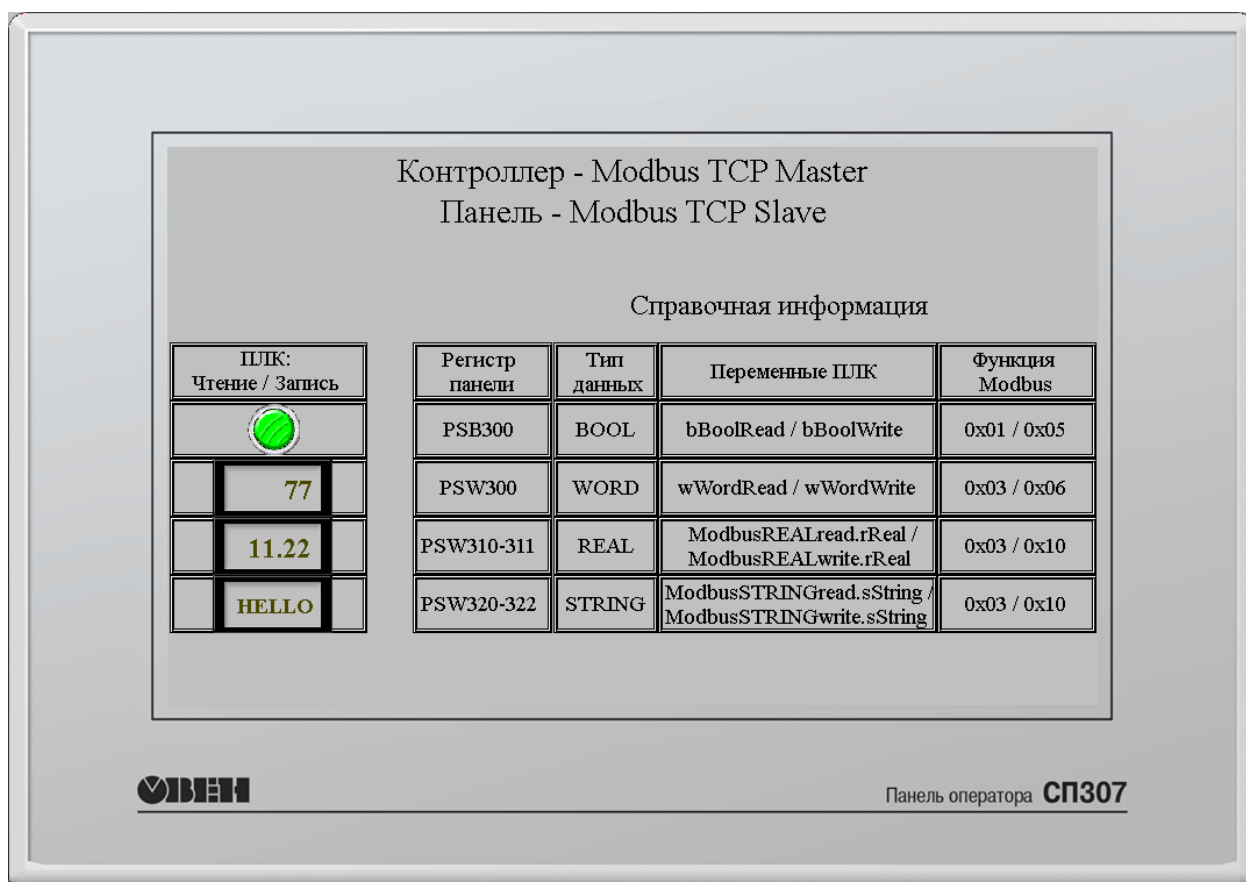


Рис. 33. Ввод значений с помощью сенсорного дисплея панели

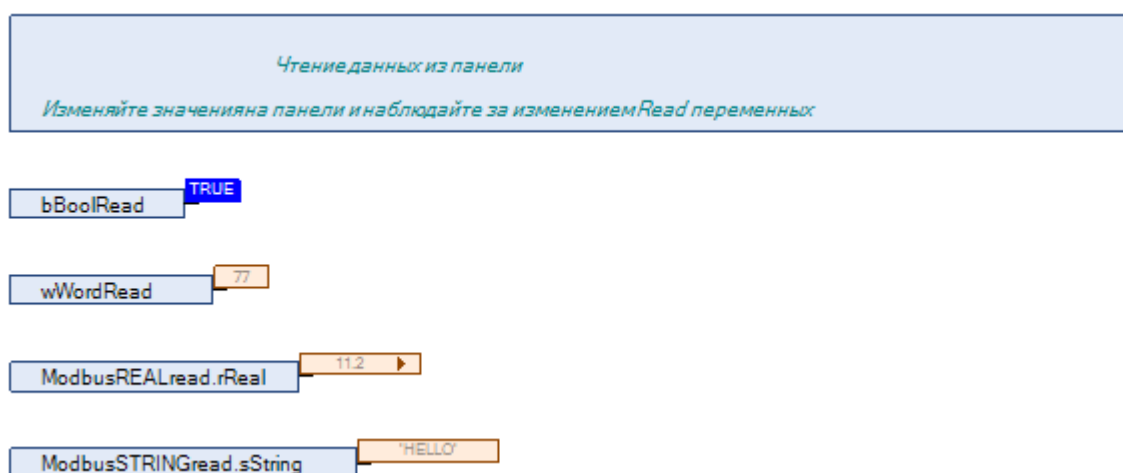


Рис. 34. Отображение введенных на панели значений в **CODESYS**

5. подготовьте новые значения **Write** переменных в программе **PLC_PRG**, установите триггеры записи в положение **TRUE**, запишите их с помощью комбинации клавиш **Ctrl+F7** и наблюдайте, как они будут записаны в панель и отображены на ее дисплее:

Device.Application.PLC_PRG					
Expression	Type	Value	Prepared value	Address	Comment
bBoolRead	BOOL	TRUE			чтение BOOL из регистра панели PSB300
bBoolWrite	BOOL	FALSE	FALSE		запись BOOL в регистр панели PSB300
wWordRead	WORD	77			чтение WORD из регистра панели PSW300
wWordWrite	WORD	0	88		запись WORD в регистр панели PSW300
ModbusREALread	ModbusReal				объединение для чтения REAL (PSW310-311)
arrReal	ARRAY [0..1] OF WORD				переменная для канала Modbus
rReal	REAL	11.22			переменная для программы
ModbusSTRINGread	ModbusString				объединение для записи REAL (PSW310-311)
ModbusREALwrite	ModbusReal				объединение для чтения REAL (PSW310-311)
arrReal	ARRAY [0..1] OF WORD				переменная для канала Modbus
rReal	REAL	0	44.33		переменная для программы
ModbusSTRINGwrite	ModbusString				объединение для записи STRING (PSW310-311)
arrSTRING	ARRAY [0..2] OF WORD				переменная для канала Modbus
sString	STRING(6)		'Привет'		переменная для программы
bBoolWriteTrigger	BOOL	FALSE	TRUE		триггер для записи BOOL переменной
bWordWriteTrigger	BOOL	FALSE	TRUE		триггер для записи WORD переменной
bRealWriteTrigger	BOOL	FALSE	TRUE		триггер для записи REAL переменной
bStringWriteTrigger	BOOL	FALSE	TRUE		триггер для записи STRING переменной

Ctrl+F7

Запись данных в панель
 Изменяйте значения Write переменных, записывайте в их триггеры значение TRUE и наблюдайте за соответствующими изменениями на панели

bBoolWrite	FALSE
bBoolWriteTrigger	TRUE
wWordWrite	88
bWordWriteTrigger	TRUE
ModbusREALwrite.rReal	44.33
bRealWriteTrigger	TRUE
ModbusSTRINGwrite.sString	'Привет'
bStringWriteTrigger	TRUE

Рис. 34. Изменение значений переменных в **CODESYS**

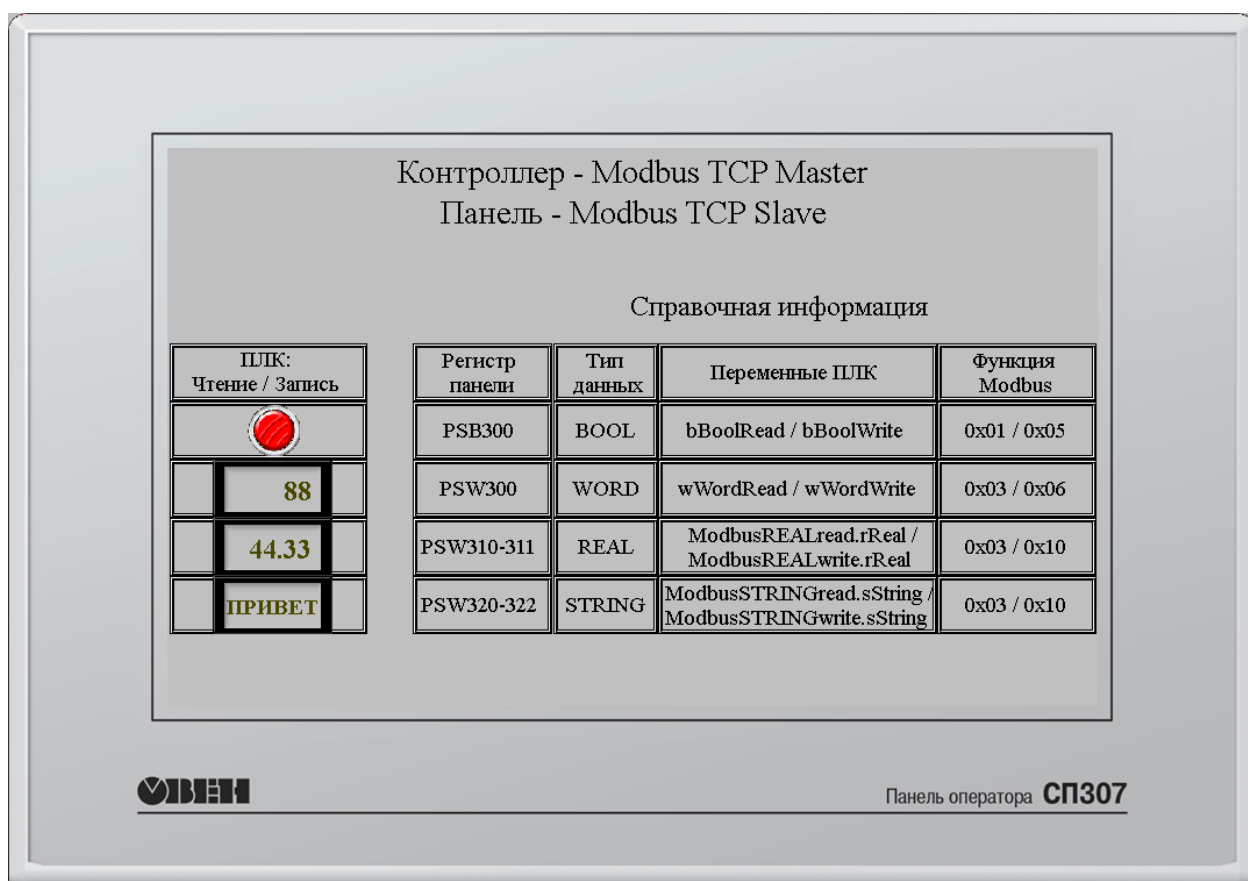


Рис. 35. Отображение введенных в **CODESYS** значений на панели