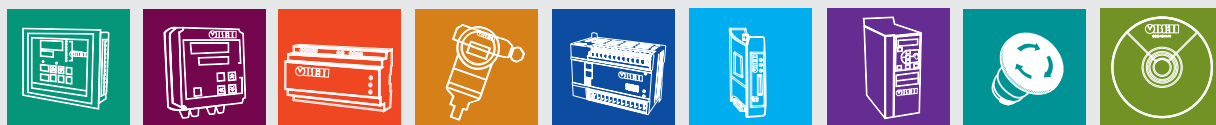


# СПЗхх

## Примеры настройки обмена

ПЛК110 [M02] (Modbus RTU Master) + СП307 (Modbus RTU Slave)

Версия: 1.1  
Дата: 21.02.2017



## Оглавление

1. Описание примера .....	3
2. Настройки СП307 (Modbus RTU Slave).....	4
2.1. Настройки PLC-порта .....	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 (STRING).....	11
3. Настройки ПЛК110 [M02] (Modbus RTU Master) .....	12
3.1. Конфигурация ПЛК .....	12
3.2. Программа PLC_PRG.....	20
3.3. Конфигурация задач.....	23
4. Работа с примером.....	24

## 1. Описание примера

Данный пример посвящен настройке обмена данными между панелью оператора **СПЗхх** и контроллером **ПЛК110 [M02]** по протоколу **Modbus RTU**. В этом примере контроллер выполняет функцию **Master**, а панель – **Slave**.

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. Переменные ПЛК подразделяются на две группы – одна (**Modbus**) используется в **Конфигурации ПЛК** при настройке обмена, вторая (**Программа**) – в коде пользовательской программы. Наличие двух групп переменных связано с тем, что в некоторых ситуациях (например, при работе с переменными типа **REAL**) полученные данные требуется преобразовать в нужный формат.

Табл. 1. Характеристики используемых в примере устройств

Устройство	ПЛК110-32.Р-М [M02]	СПЗ07
Функция	Master	Slave
Используемый порт	RS-485	PLC
Настройки обмена	115200, 8 бит, 1 стоп бит, без контроля четности	
Slave ID	-	1
Таргет	PLC110.32-М v2 (версия 3.08)	-
Среда разработки проекта	CoDeSys 2.3.9.41	Конфигуратор СПЗ00 (V2.D3d-2)
Название файла проекта	ModbusRTUmaster.pro	ModbusRTUslave.txp

Табл. 2. Список используемых в примере переменных

ПЛК110-32.Р-М [M02]				СПЗ07	
Переменные для чтения из панели		Переменные для записи в панель		Тип данных	Регистр
Программа	Modbus Master (конфигурация)	Программа	Modbus Master (конфигурация)		
bBoolRead	bInput	bBoolWrite	bOutput	BOOL	PSB300
wWordRead	wInput	wWordWrite	wOutput	WORD	PSW300
rRealRead	rInput	rRealWrite	rOutput	REAL	PSW310-311
sStringRead	sInput	sStringWrite	sOutput	STRING	PSW320-322

## 2. Настройки СП307 (Modbus RTU Slave)

### 2.1. Настройки PLC-порта

Для того чтобы использовать панель **СП307** в режиме **Modbus RTU Slave**, необходимо в настройках проекта на вкладке **Устройство** настроить режимы работы последовательных портов. В нашем примере мы будем использовать **PLC-порт**. Настроим его в качестве **Modbus Slave** согласно [табл. 1](#): скорость – **115200**, бит данных – **8**, стоп бит – **1**, контроль четности – **нет**. Адрес порта (**Slave ID**) будет равен **1**.

**Обратите внимание** на наличие галочки **Передача данных** – при ее отсутствии ПЛК не сможет считывать данные с панели (но сможет их записывать).

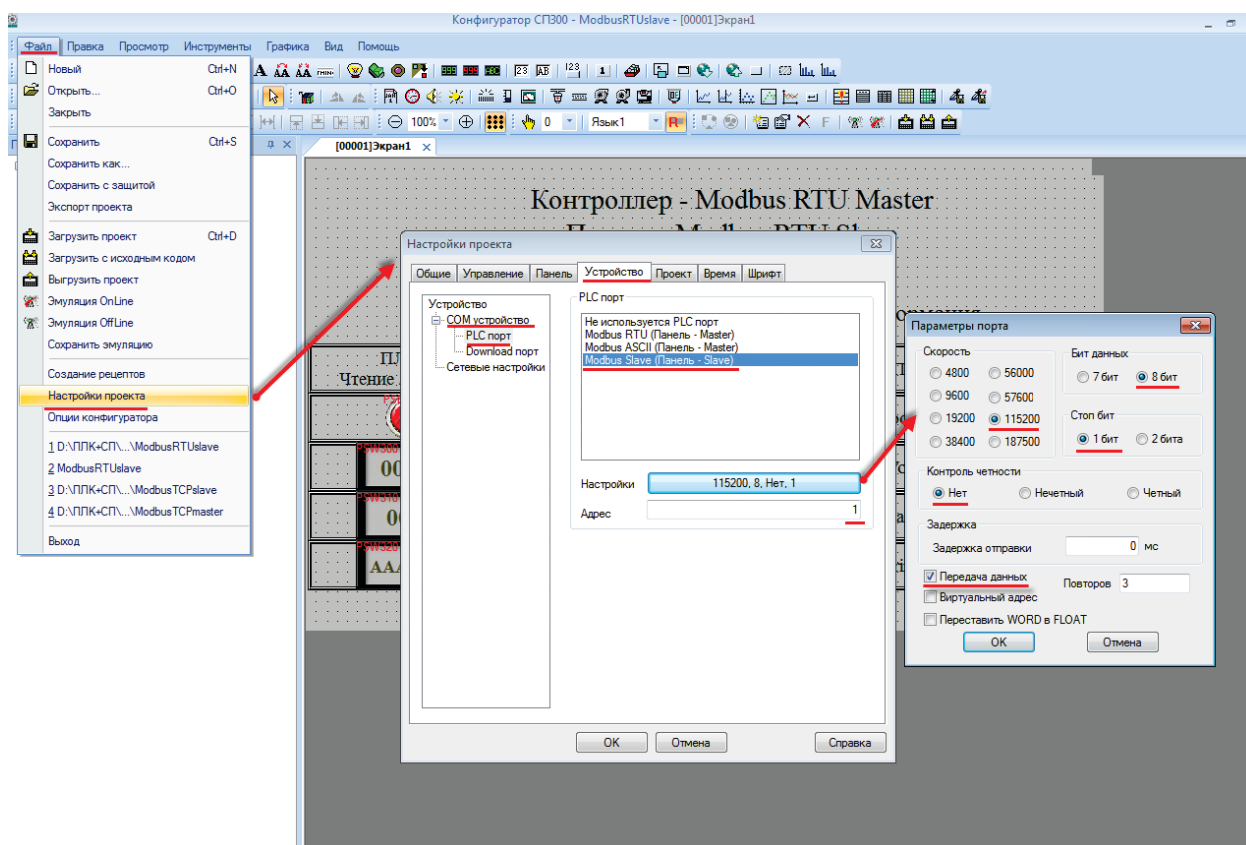


Рис. 1. Настройки **PLC-порта** панели (режим **Modbus RTU Slave**)

## 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 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 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 **rRealRead** и по команде пользователя записывает в них значение переменной **rRealWrite**. Обе переменные имеют тип **REAL**.
4. **Текстовый ввод** с привязанными регистрами **PSW320-322**. ПЛК считывает значения этих регистров в переменную **sStringRead** и по команде пользователя записывает в них значение переменной **sStringWrite**. Обе переменные имеют тип **STRING**.

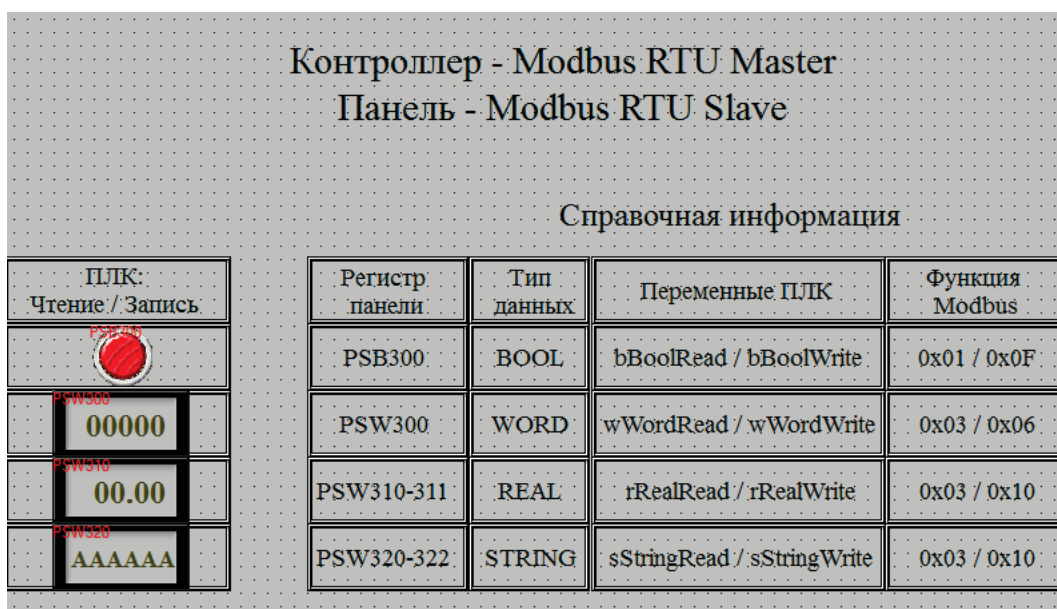


Рис. 2. Внешний вид экрана проекта

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](#)).

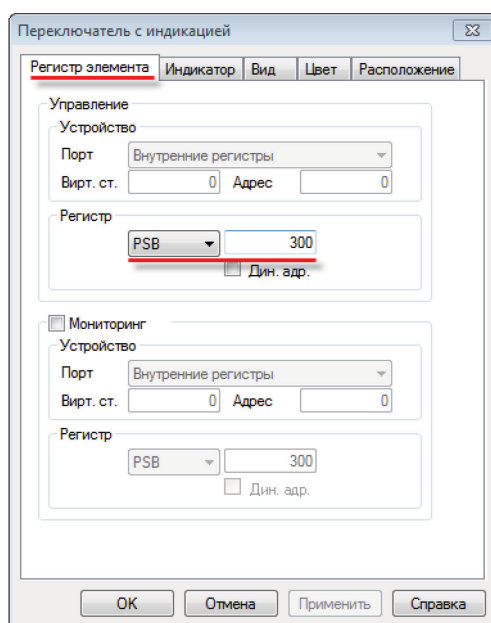


Рис. 4. Настройки элемента **Переключатель с индикацией**, вкладка **Регистр элемента**

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

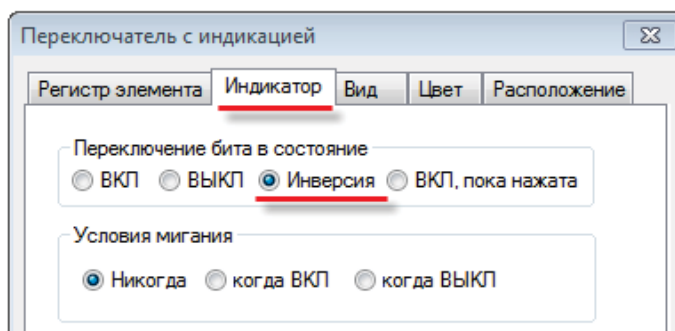


Рис. 5. Настройки элемента **Переключатель с индикацией**, вкладка **Индикатор**

### 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**.

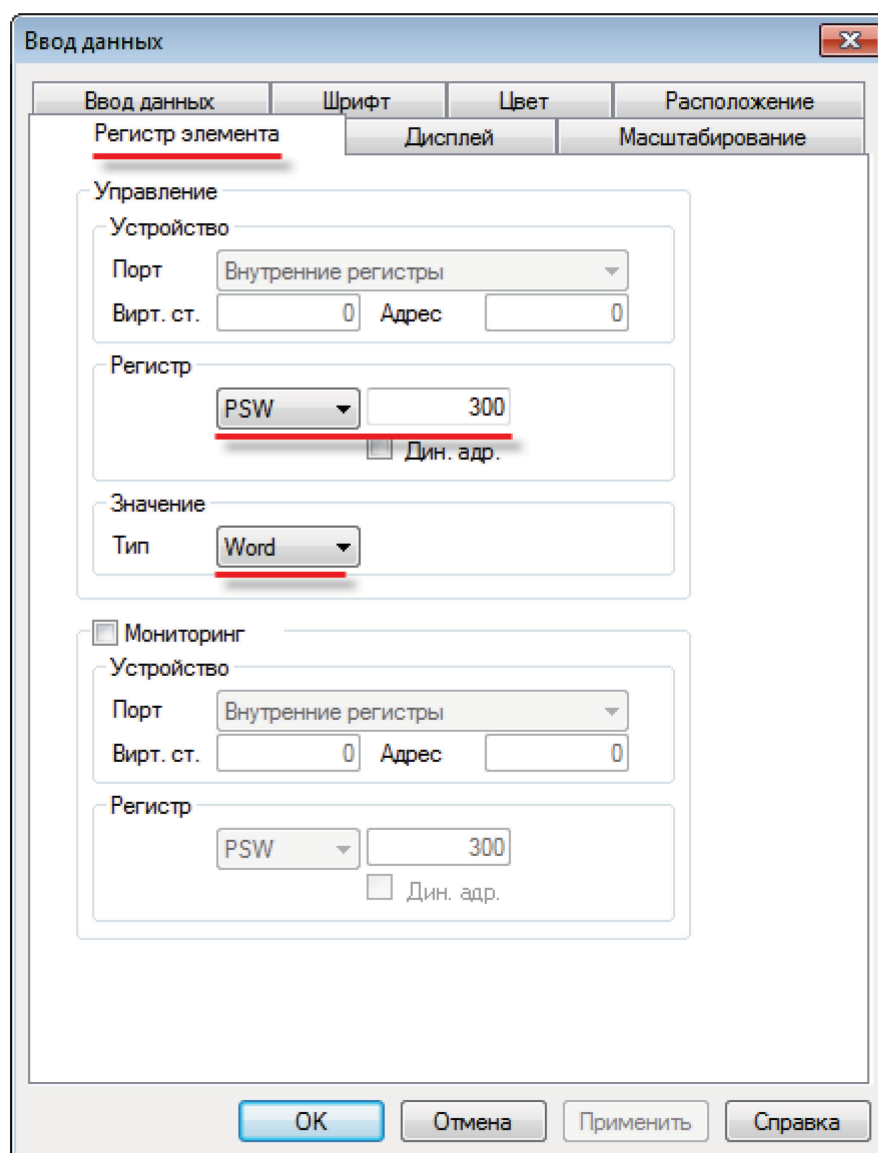


Рис. 7. Настройки элемента **Цифровой ввод**, вкладка **Регистр элемента**

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

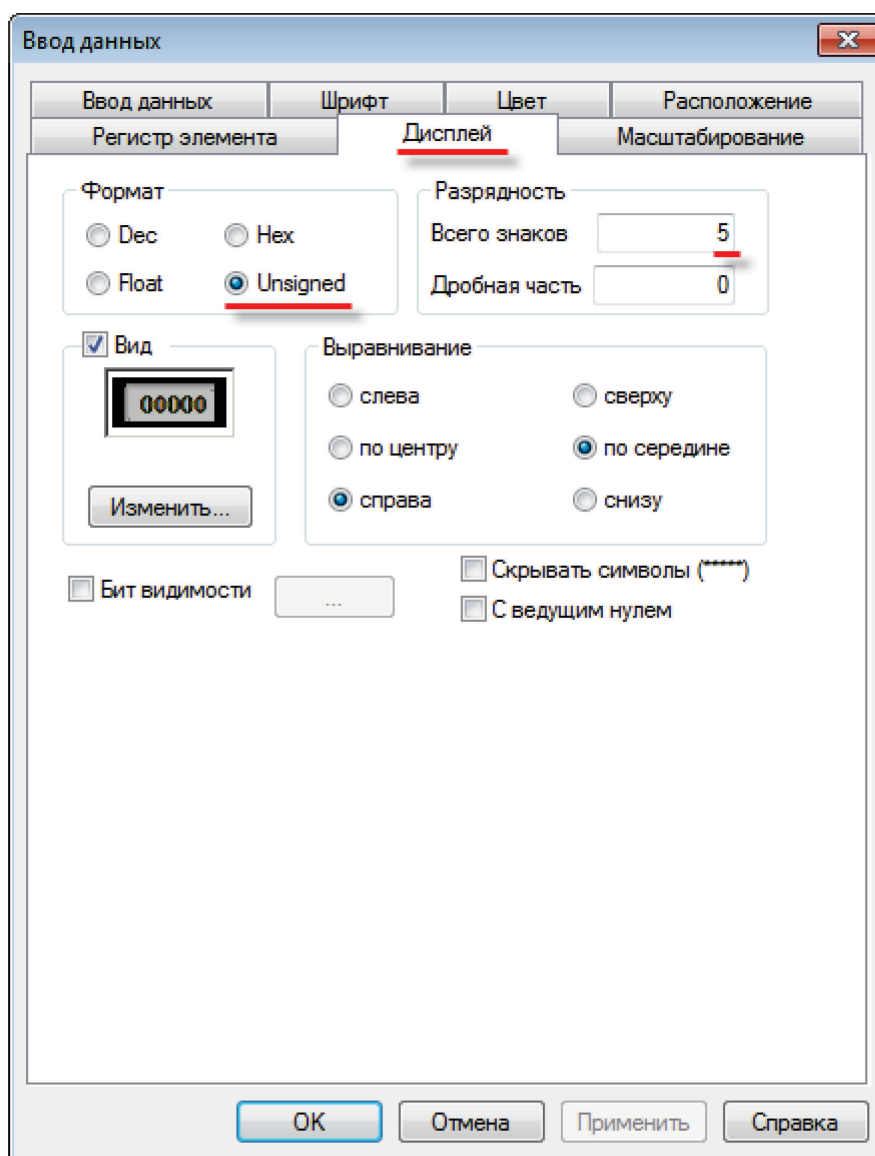


Рис. 8. Настройки элемента **Цифровой ввод**, вкладка **Дисплей**

**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](#).

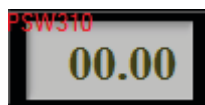


Рис. 9. Внешний вид элемента **Цифровой ввод**

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](#)).

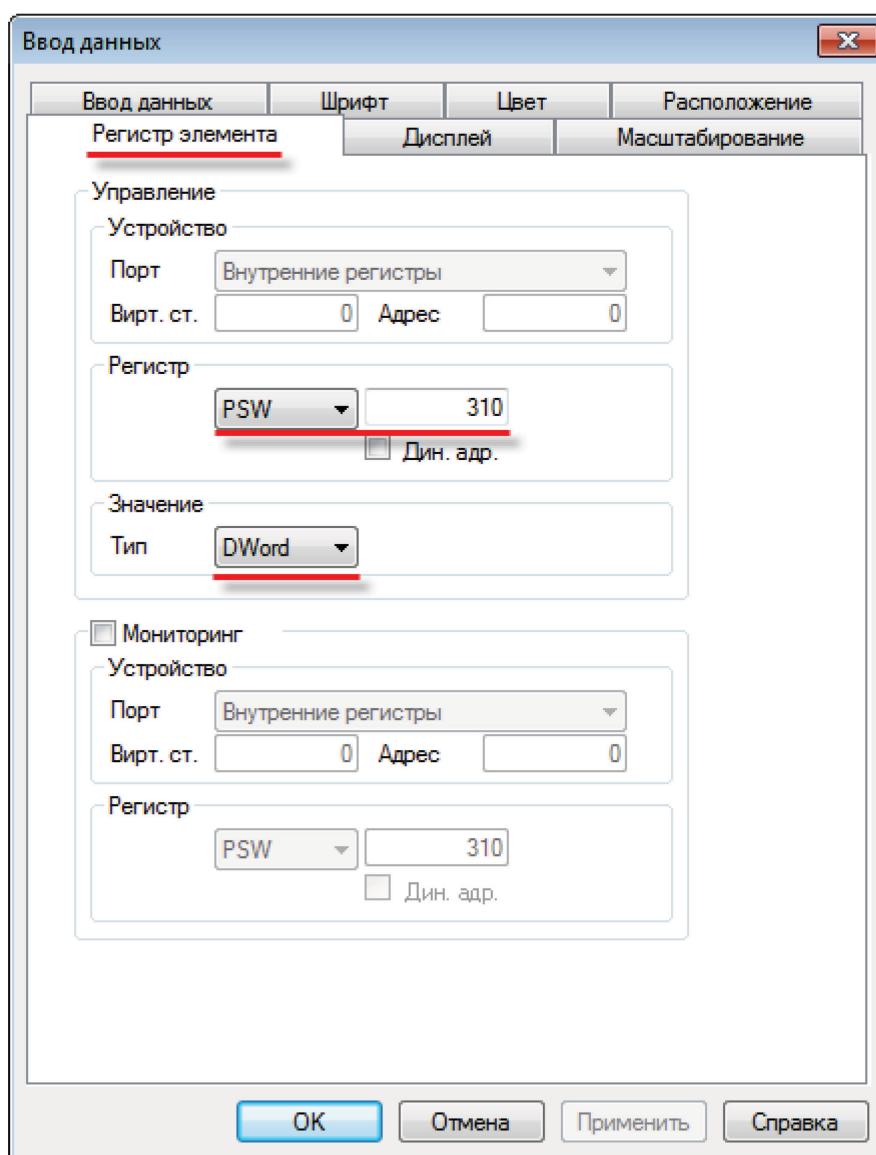


Рис. 10. Настройки элемента **Цифровой ввод**, вкладка **Регистр элемента**

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.

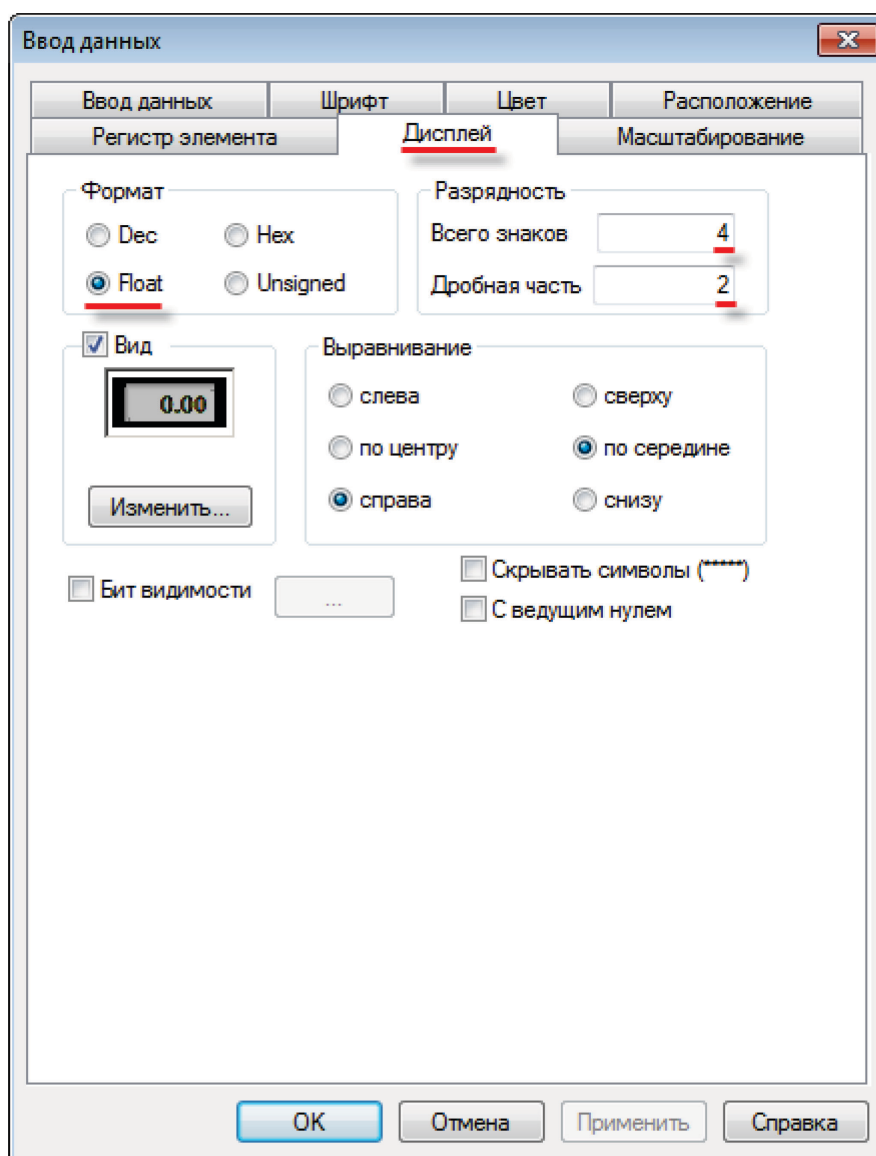


Рис. 11. Настройки элемента **Цифровой ввод**, вкладка **Дисплей**

## 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.

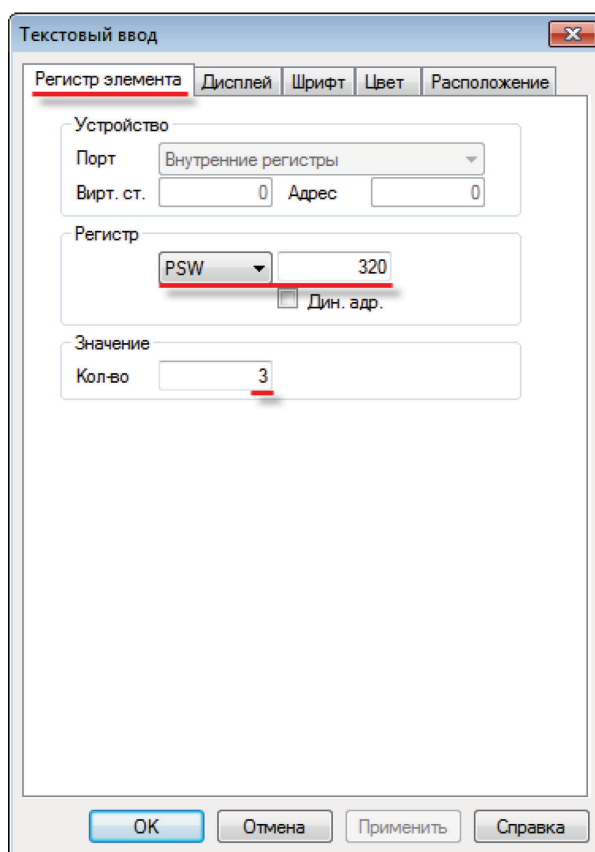


Рис. 13. Настройки элемента **Текстовый ввод**, вкладка **Регистр элемента**

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

### 3. Настройки ПЛК110 [M02] (Modbus RTU Master)

#### 3.1. Конфигурация ПЛК

В компоненте **Конфигурация ПЛК** (вкладка **Ресурсы**) пользователь производит настройку каналов Modbus и привязывает к ним переменные.

Нажмем **ПКМ** на название контроллера (в нашем примере - **PLC110\_32**) и добавим подэлемент **Modbus (Master)**:

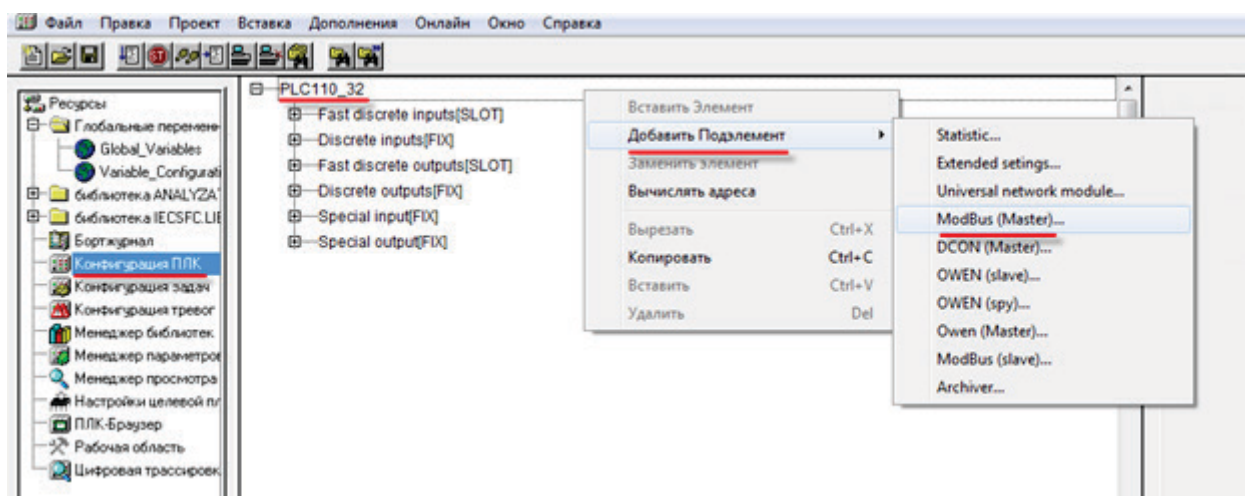


Рис. 14. Конфигурация ПЛК. Добавление **Modbus (Master)**

Этот элемент не нуждается в настройках.

Выберем порт ПЛК, который будет использоваться для связи с панелью. Для этого в элементе **Modbus (Master)** нажмем **ПКМ** на порт **Debug RS-232** и в контекстном меню выберем команду **Заменить элемент**. В нашем примере мы используем **RS-485**; при необходимости, можно использовать **RS-232** или **Debug RS-232** – это повлияет только на порт, к которому необходимо будет подключить панель, и кабель, с помощью которого будут соединены панель и ПЛК.

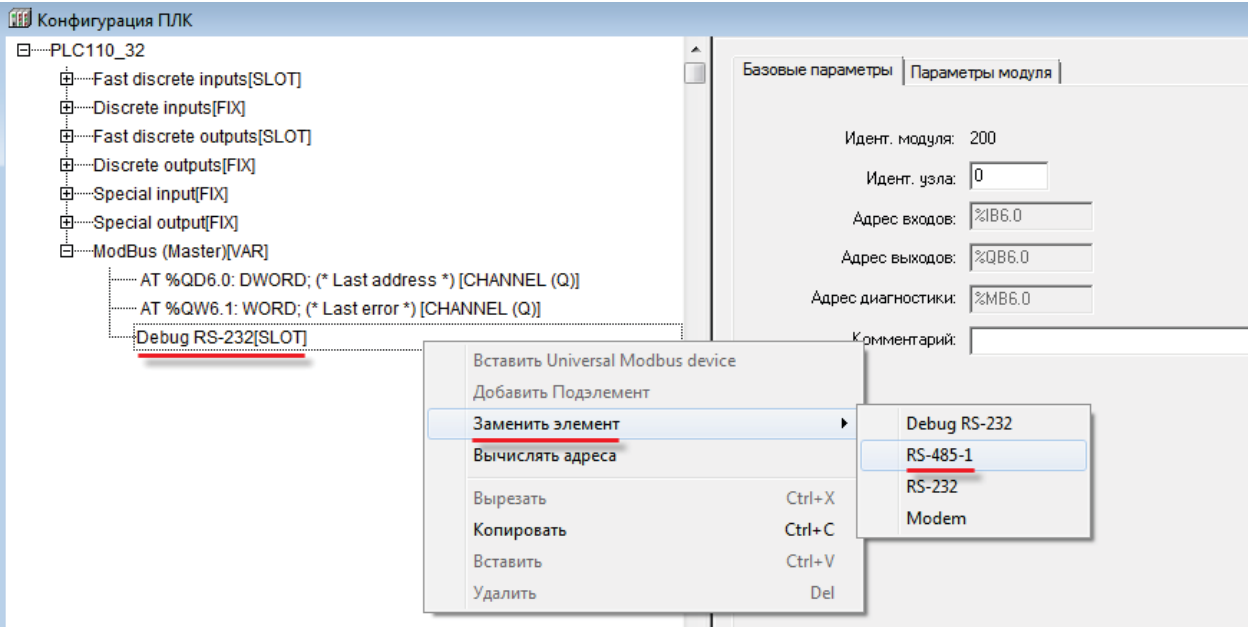


Рис. 15. Конфигурация ПЛК. Выбор порта

Настройки порта по умолчанию соответствуют тем настройкам, которые мы задали панели (согласно [табл. 1](#)): скорость – **115200**, бит данных – **8**, стоп бит – **1**, контроль четности – **нет**. Для параметра **Frame Oriented** выберем значение **RTU**.

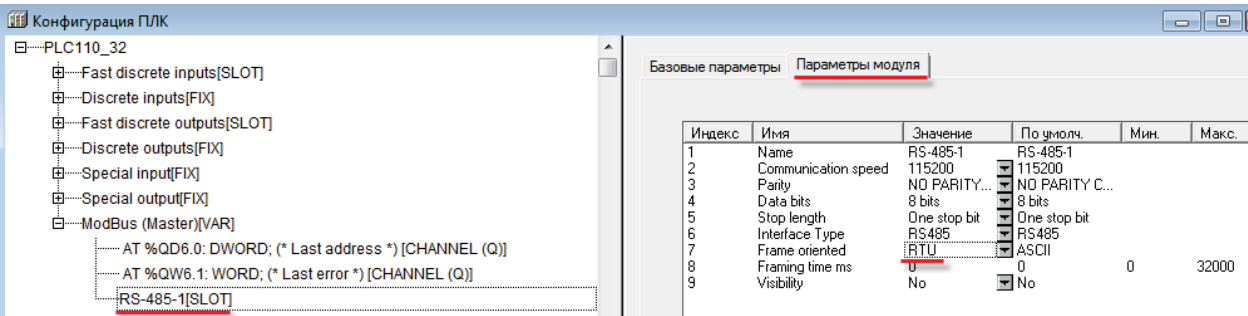


Рис. 16. Конфигурация ПЛК. Настройки порта

Нажмем **ПКМ** на элемент **Modbus (Master)** и добавим два подэлемента **Universal Modbus Device**:

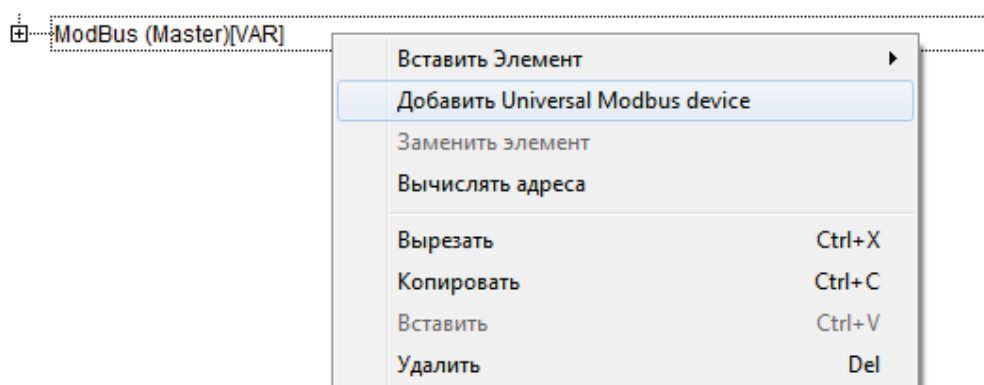


Рис. 17. Конфигурация ПЛК. Добавление **Universal Modbus Device**

Один из них будет использоваться для чтения значений из панели, второй – для записи.

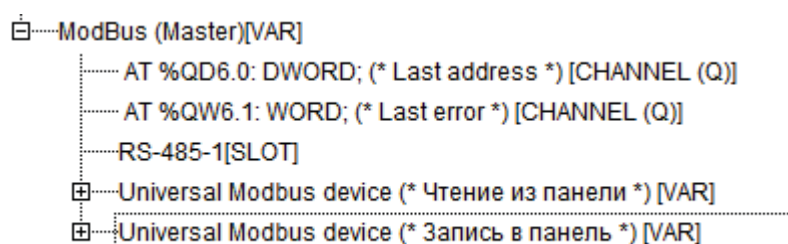


Рис. 18. Внешний вид **Конфигурации ПЛК** после добавления двух **Universal Modbus Device**

В настройках элемента **Чтение из панели** укажем тип связи (**Serial**), **Slave ID** панели (**1**) и режим опроса (**By poll time**, т.е. циклически).

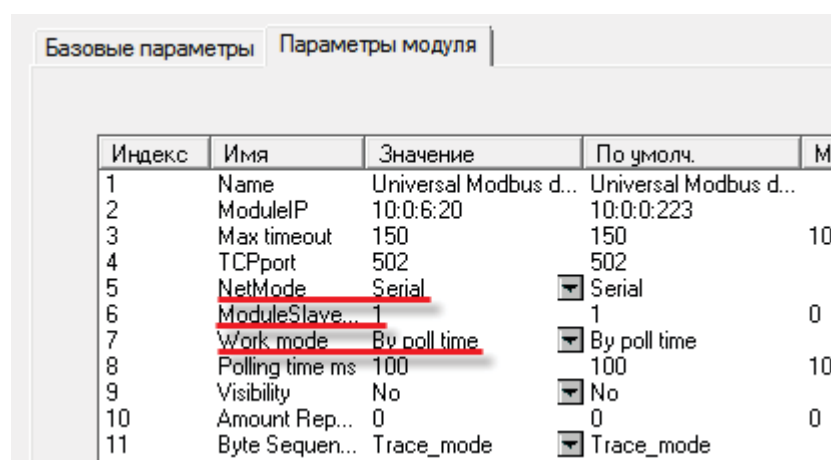


Рис. 19. Настройки **Universal Modbus Device (Чтение из панели)**

В настройках элемента **Запись в панель** укажем тип связи (**Serial**), **Slave ID** панели (**1**) и режим опроса 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	

Рис. 20. Настройки **Universal Modbus Device (Запись в панель)**

Нажмем **ПКМ** на элемент **Universal Modbus Device (Чтение из панели)** и добавим в него подэлементы **8 bit Input Module**, **Register Input Module**, **Real Input Module** и **String Input Module**.

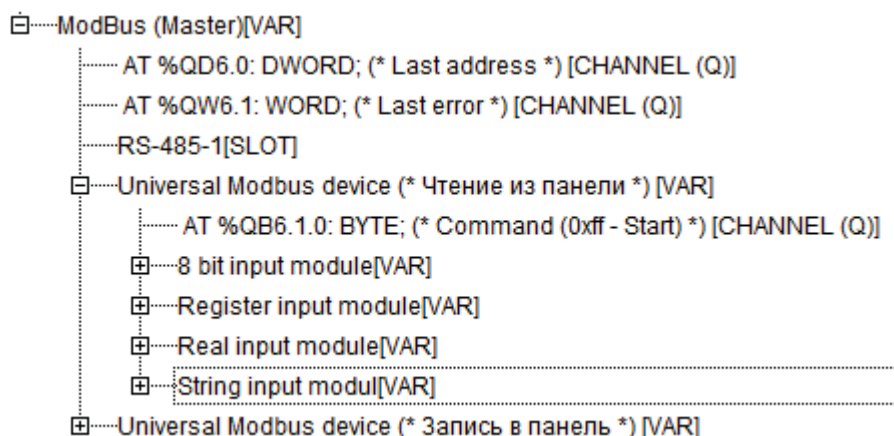


Рис. 21. **Universal Modbus Device (Чтение из панели)** с добавленными **Input** модулями

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

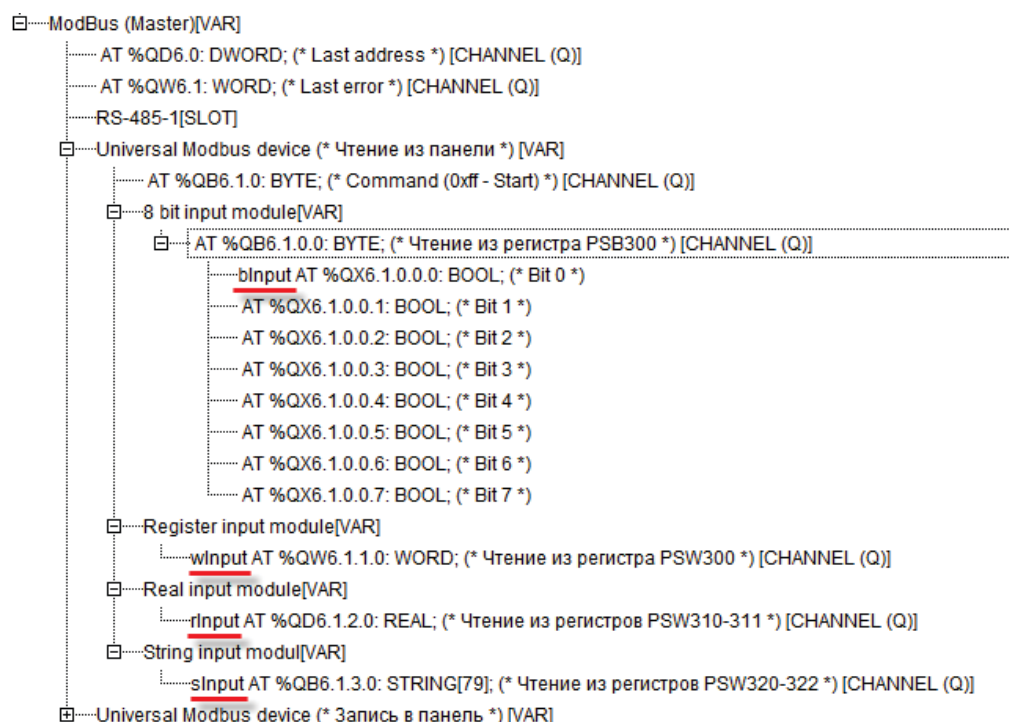


Рис. 22. Привязка переменных к каналам

Настройки модулей (используемые регистры панели и функции Modbus) приведены ниже.

**Обратите внимание**, что работа с переменными типа **BOOL** осуществляется через **8 Bit Input Module**. При этом нулевому биту соответствует заданный регистр модуля (см. рис. 22), первому – заданный регистр+1 (PSB301), второму – PSB302 и т.д.

**Обратите внимание**, что при работе с переменными, занимающими несколько регистров панели (тип **REAL** и **STRING**), указывается только первый из группы регистров.

Инде...	Имя	Значение	По умолч.
1	Name	8 bit input module	8 bit input module
2	Register a...	300	0
3	Command	Read coils status (...)	Read inputs status (0x02)
8	Visibility	No	No

Рис. 23. Параметры **8 Bit Input Module**

Инде...	Имя	Значение	По умолч.
1	Name	Register input module	Register input module
2	Regist...	300	0
3	Comma...	Read holding Register...	Read holding Registers ..
8	Visibility	No	No

Рис. 24. Параметры **Register Input Module**



Инде...	Имя	Значение	По умолч.
1	Name	float input module	float input module
2	Regist...	310	0
3	Comma...	Read holding Register...	Read holding Registers ...
8	Visibility	No	No

Рис. 25. Параметры **Real Input Module**

Базовые параметры			
Параметры модуля			
Индекс	Имя	Значение	По умолч.
1	Name	String input module	String input m...
2	Command	Read holding Registers (0...	Read bytes (0...
3	Register a...	320	0
4	Amount ...	6	80
8	Visibility	No	No

Рис. 26. Параметры **String Input Module**

Нажмем **ПКМ** на элемент **Запись в панель** и добавим в него подэлементы **8 bit Output Module**, **Register Output Module**, **Real Output Module** и **String Output Module**.

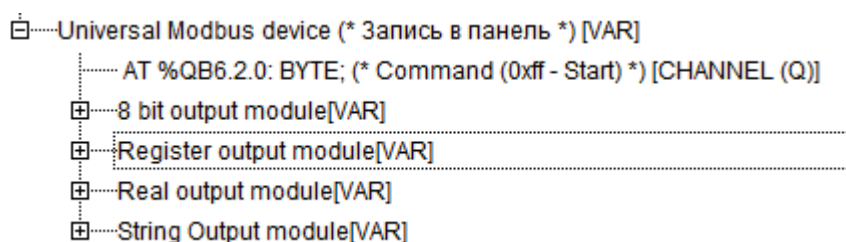


Рис. 27. **Universal Modbus Device (Запись в панель)** с добавленными **Output** модулями

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

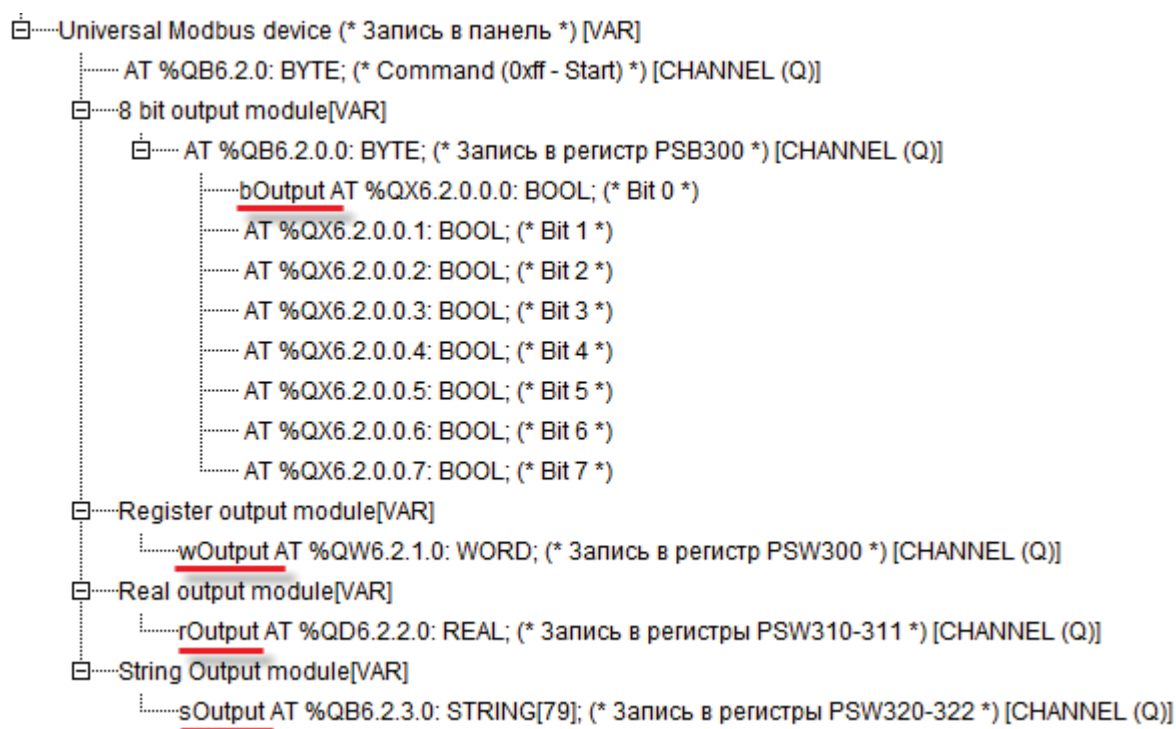


Рис. 28. Привязка переменных к каналам

Настройки модулей (используемые регистры панели и функции Modbus) приведены ниже.

**Обратите внимание**, что работа с переменными типа **BOOL** осуществляется через **8 Bit Output Module**. При этом нулевому биту соответствует заданный регистр модуля (см. рис. 28), первому – заданный регистр+1 (PSB301), второму – PSB302 и т.д.

**Обратите внимание**, что при работе с переменными, занимающими несколько регистров панели (тип **REAL** и **STRING**), указывается только первый из группы регистров.

Инде...	Имя	Значение	По умолч.
1	Name	8 bit output module	8 bit output mod
2	Register ...	300	0
3	Command	Force multiply coils (0...	Force multiply c
8	Visibility	No	No

Рис. 29. Параметры **8 Bit Output Module**

Базовые параметры			
Параметры модуля			
Инде...	Имя	Значение	По умолч.
1	Name	Register	Register
2	Register...	300	0
3	Command	Preset singl register (...)	Preset singl regi
8	Visibility	No	No

Рис. 30. Параметры **Register Output Module**

Индекс	Имя	Значение	По умолч.
1	Name	float output module	float output mod
2	Regist...	310	0
3	Comm...	Preset multiple Registe...	Preset multiple F
8	Visibility	No	No

Рис. 31. Параметры **Real Output Module**

Индекс	Имя	Значение	По умолч.
1	Name	String output module	String output
2	Comma...	Preset multiple Register...	Preset singl re
3	Registe...	320	0
4	Amount...	6	80
8	Visibility	No	No

Рис. 32. Параметры **String Output Module**

### 3.2. Программа PLC\_PRG

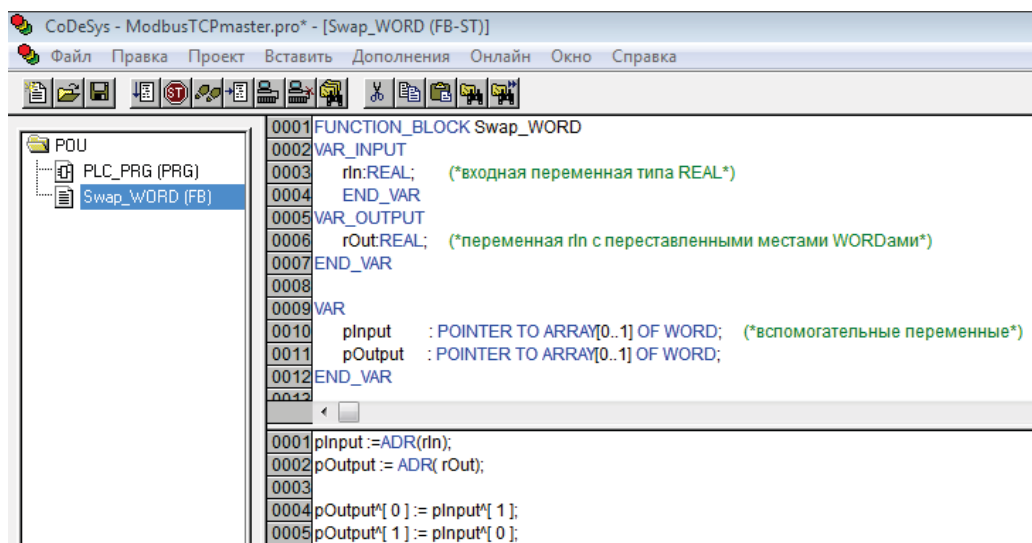
Данные, считанные через **Конфигурацию ПЛК**, присваиваются локальным переменным программы **PLC\_PRG**. В свою очередь, значения, заданные в программе, передаются в **Конфигурацию ПЛК** и записываются в панель. Этот подход использован только на уровне примера – в реальных проектах зачастую удобнее пользоваться в программе переменными из **Конфигурации ПЛК**.

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

```
0001 PROGRAM PLC_PRG
0002 VAR
0003   bBoolRead   :BOOL;           (*чтение BOOL из регистра панели PSB300*)
0004   bBoolWrite  :BOOL;           (*запись BOOL в регистр панели PSB300*)
0005
0006   wWordRead    :WORD;           (*чтение WORD из регистра панели PSW300*)
0007   wWordWrite   :WORD;           (*запись WORD в регистр панели PSW300*)
0008
0009   rRealRead    :REAL;           (*чтение REAL из регистров панели PSW310-311*)
0010   rRealWrite   :REAL;           (*запись REAL в регистры панели PSW310-311*)
0011
0012   Swap_WORD_panel:Swap_WORD;    (*экземпляр ФБ для переворота WORD в REAL*)
0013
0014   sStringRead  :STRING(6);       (*чтение STRING из регистра панели PSW320-322*)
0015   sStringWrite :STRING(6);       (*запись STRING в регистры панели PSW320-322*)
0016 END_VAR
```

Рис. 33. Объявление переменных в программе **PLC\_PRG**

Восемь переменных (**Read** и **Write**) соответствуют восьми переменным (**Input** и **Output**) из **Конфигурации ПЛК**; функциональный блок **SWAP\_WORD** используется для перестановки **WORD** в переменной типа **REAL**, поскольку порядок **WORD** в переменных с плавающей точкой в панели инвертирован по сравнению с **ПЛК**. Код функционального блока приведен ниже:



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

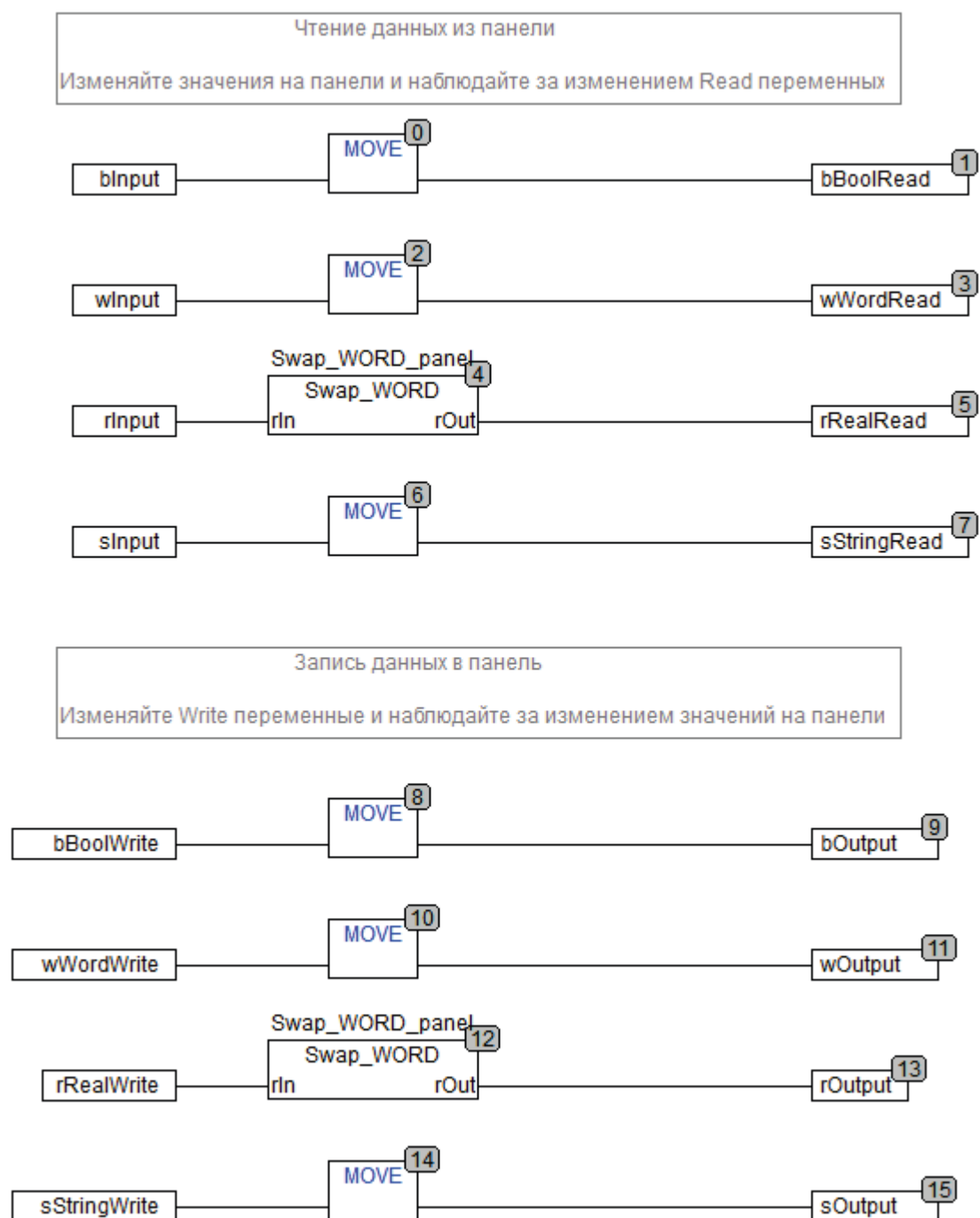


Рис. 35. Код программы **PLC\_PRG**

Программа **PLC\_PRG** каждый цикл выполняет следующие действия:

1. значение переменной **bInput** присваивается переменной **bBoolRead**;
2. значение переменной **wInput** присваивается переменной **wWordRead**;
3. значение переменной **rInput** обрабатывается с помощью функционального блока **Swap\_WORD** (**WORD** в **REAL** меняются местами для корректного отображения в ПЛК; это связано с тем, что порядок **WORD** в панели инвертирован по сравнению с ПЛК) и записывается в переменную **rRealRead**;
4. значение переменной **sInput** присваивается переменной **sStringRead**;
5. значение переменной **bBoolWrite** присваивается переменной **bOutput**;
6. значение переменной **wWordWrite** присваивается переменной **wOutput**;
7. значение переменной **rRealWrite** обрабатывается с помощью функционального блока **Swap\_WORD** (**WORD** в **REAL** меняются местами для корректного отображения в панели; это связано с тем, что порядок **WORD** в панели инвертирован по сравнению с ПЛК) и записывается в переменную **rOutput**;
8. значение переменной **sStringWrite** присваивается переменной **sOutput**.

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

### 3.3. Конфигурация задач

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

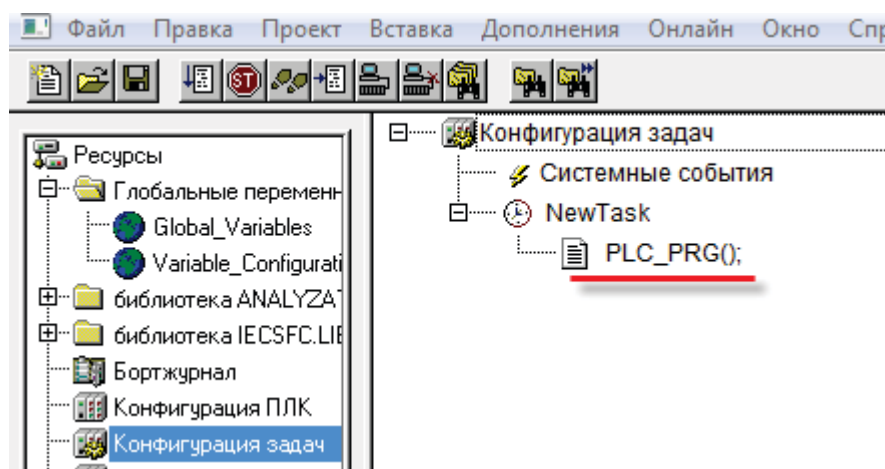


Рис. 36. Привязка программы **PLC\_PRG** к задаче

## 4. Работа с примером

1. Откройте проект **ModbusRTUmaster.pro** в **CoDeSys**, подключитесь к ПЛК, загрузите проект и запустите его:

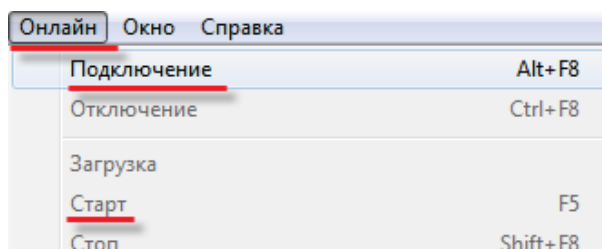


Рис. 37. Загрузка и запуск проекта для ПЛК в **CoDeSys**

2. Откройте **ModbusRTUslave.txp** в программе **Конфигуратор СПЗ00** и загрузите его в панель:

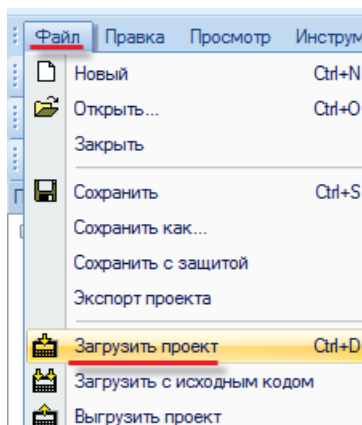


Рис. 38. Загрузка проекта в панель

3. Соедините панель и ПЛК с помощью кабеля (схемы соединительных кабелей приведены в **РЭ** панели СПЗxx):

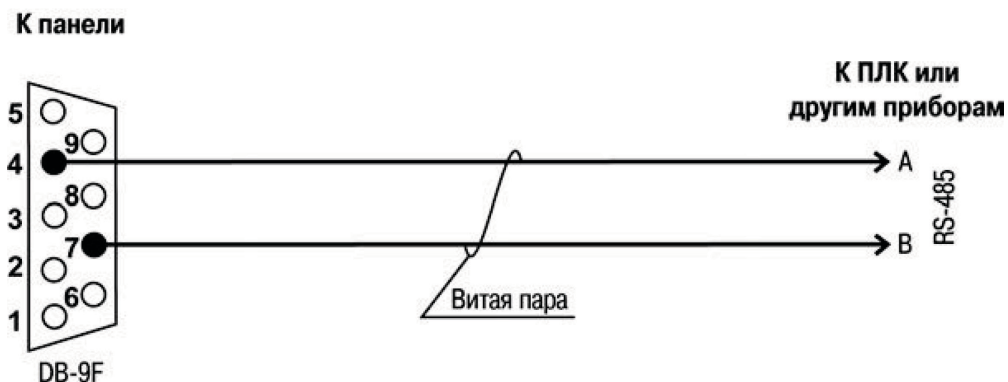


Рис. 39. Соединительный кабель для связи панели с ПЛК по интерфейсу **RS-485**



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



Рис. 40. Ввод значений с помощью сенсорного дисплея панели

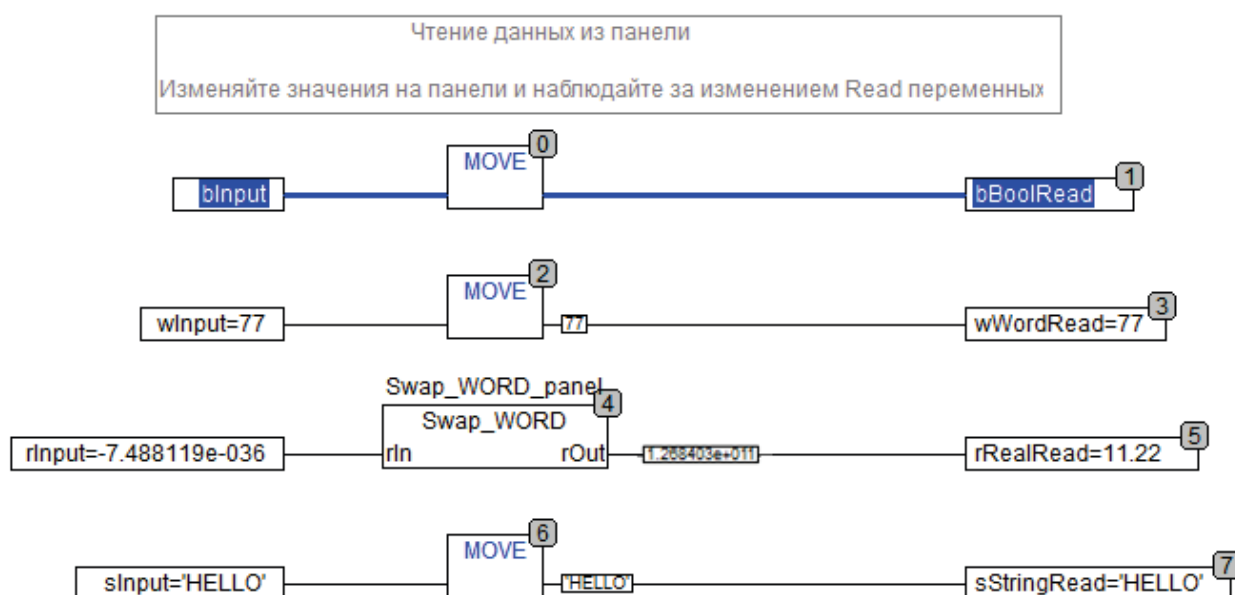


Рис. 41. Отображение введенных на панели значений в **CoDeSys**

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

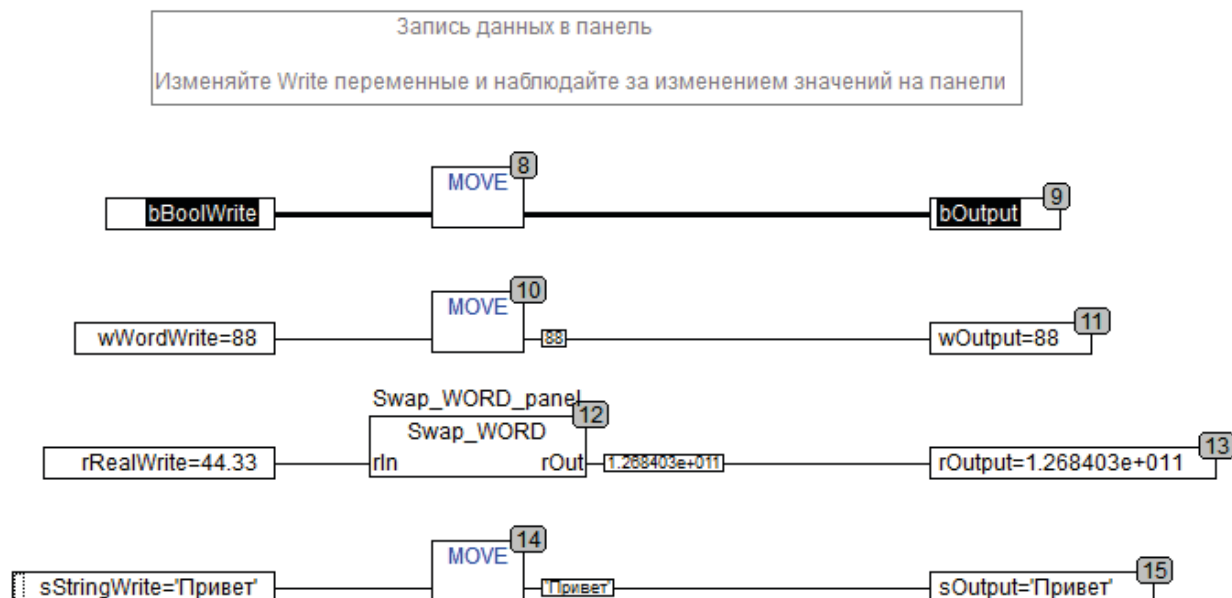


Рис. 42. Изменение значений переменных в **CoDeSys**



Рис. 43. Отображение введенных в **CoDeSys** значений на панели