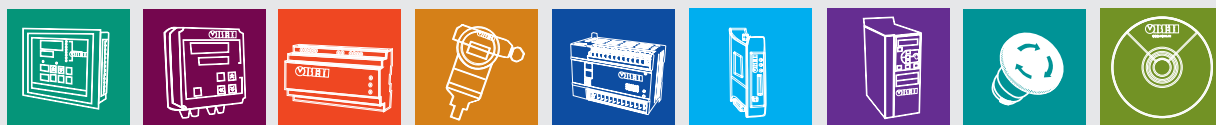


# СПЗхх

## Примеры настройки обмена

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

Версия: 1.1  
Дата: 21.02.2017



## Оглавление

1. Описание примера .....	3
2. Настройки ПЛК110 [M02] (Modbus RTU Slave) .....	4
2.1. Конфигурация ПЛК .....	4
2.2. Программа PLC_PRG .....	8
2.3. Конфигурация задач .....	9
3. Настройки СП307 (Modbus RTU Master) .....	10
3.1. Настройки PLC-порта панели .....	10
3.2. Экран проекта .....	11
3.2.1. Переключатель с индикацией (BOOL) .....	12
3.2.2. Цифровой ввод (WORD) .....	13
3.2.3. Цифровой ввод (REAL) .....	15
3.2.4. Текстовый ввод (STRING) .....	17
4. Работа с примером .....	18

## 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 **Slave**, а панель – **Master**.

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	ПЛК110-32.Р-М [M02]	СП307
Функция	Slave	Master
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 СП300 (V2.D3d-2)
Название файла проекта	ModbusRTUslave.pro	ModbusRTUmaster.txp

Табл. 2. Список используемых в примере переменных

ПЛК110-32.Р-М [M02]		СП307
Переменные, которые читает/записывает панель	Тип данных	Регистр ПЛК
bInputOutput	BOOL	0x0
wInputOutput	WORD	4x1
rInputOutput	REAL	4x2
sInputOutput	STRING	4x4

## 2. Настройки ПЛК110 [M02] (Modbus RTU Slave)

### 2.1. 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 (Slave)**:

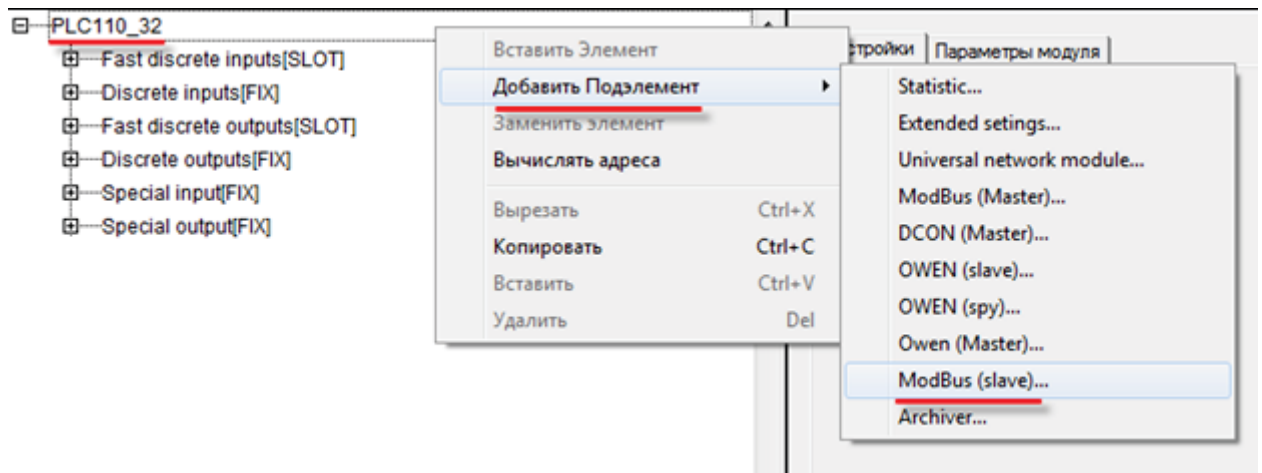


Рис. 1. Конфигурация ПЛК. Добавление **Modbus (Slave)**

В его настройках выберем адрес (**Slave ID**), равный 1 (в соответствии с [табл. 1](#)):

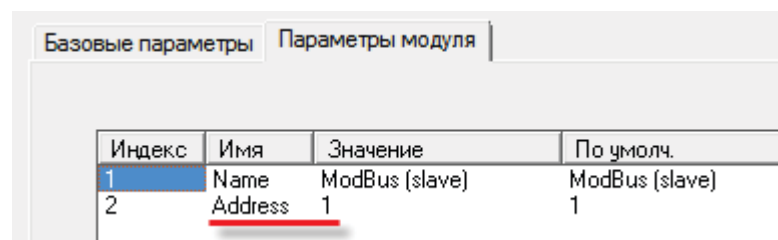


Рис. 2. Конфигурация ПЛК. Настройка **Modbus (Slave)**

Выберем порт ПЛК, который будет использоваться для связи с панелью. Для этого нажмем **ПКМ** на элемент **Modbus (FIX)** и добавим подэлемент **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.

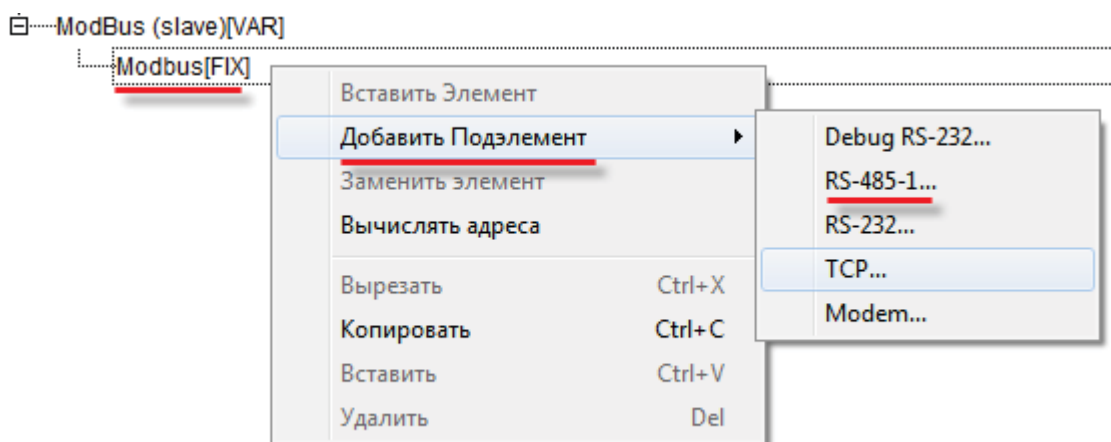


Рис. 3. Конфигурация ПЛК. Добавление подэлемента **RS-485**

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**.

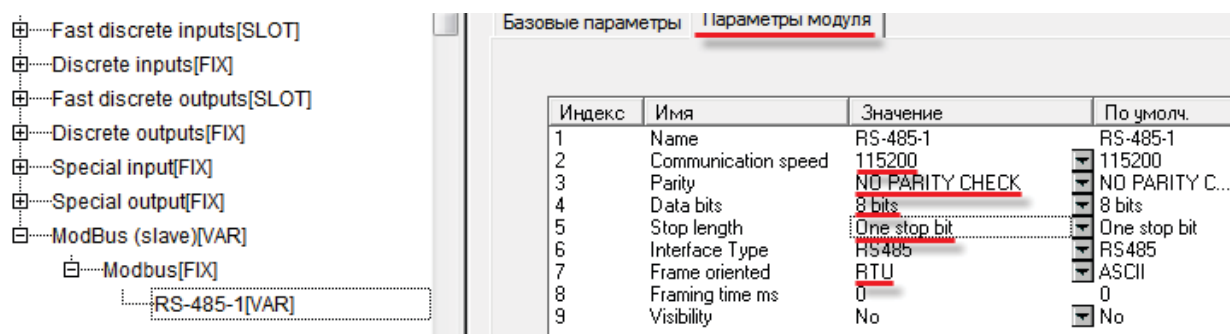


Рис. 4. Конфигурация ПЛК. Настройки порта

Нажмем ПКМ на элемент **Modbus (Slave)** и добавим подэлементы **8 bits** (для **BOOL**), **2 byte** (для **WORD**), **Float** (для **REAL**) и далее – 3 элемента **2 byte** (для **STRING** из 6 символов):

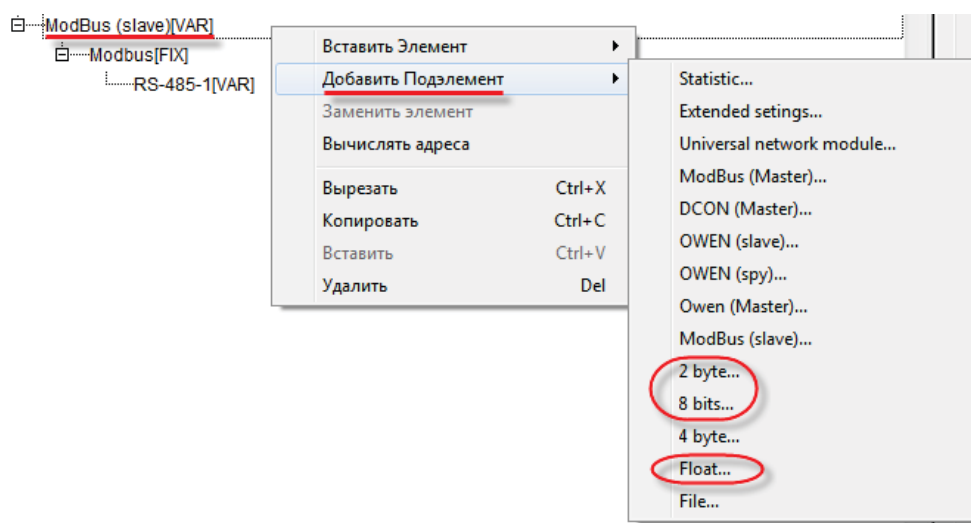


Рис. 5. Конфигурация ПЛК. Добавление каналов

В результате **Конфигурация ПЛК** будет выглядеть следующим образом (см. рис. 6). Привяжем к каналам переменные (после ввода их имен они автоматически будут добавлены в список глобальных переменных проекта). Для ввода имени переменной два раза нажмите на **AT**. **Обратите внимание**, что переменная **slInputOutput** будет объявлена в программе [PLC\\_PRG](#).

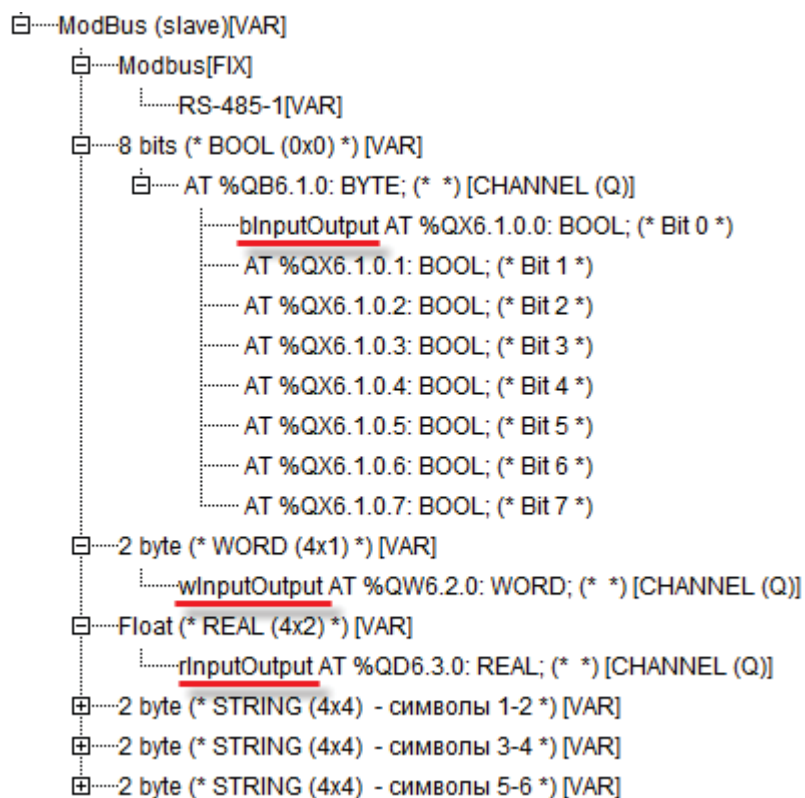


Рис. 6. Внешний вид **Modbus (Slave)** с добавленными каналами

**Обратите внимание,** что нумерация регистров начинается с нуля, при этом каждый регистр физически занимает два байта (в т.ч. и для **8 bits**). В связи с этим, переменная типа **REAL** займет два регистра (2 и 3). Переменная типа **STRING**, которой соответствует три **2 byte** элемента, займет регистры 4 и 5. Это необходимо учитывать при настройке master-устройства.

Подробнее вопросы адресации рассмотрены в **РЭ** на ПЛК.

## 2.2. Программа PLC\_PRG

В программе **PLC\_PRG** мы будем оперировать переменными, объявленными в **Конфигурации ПЛК**. Помимо этого, необходимо объявить переменную типа **STRING** с указанием адреса первого из группы регистров **Конфигурации ПЛК**, которые она занимает (см. рис. 6-7):

```
0001 PROGRAM PLC_PRG
0002 VAR
0003   sInputOutput AT %QW6.4.0: STRING(6);           (*собираем STRING переменную из трех WORD [т.е. шести символов],
0004                                                    указывая адрес [см. Конфигурация ПЛК] первого из них*)
0005 END_VAR
```

Рис. 7. Объявление переменных в программе **PLC\_PRG**

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



Рис. 8. Код программы **PLC\_PRG**

В процессе работы проекта пользователь может изменять значения 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 переменных.



### 2.3. Конфигурация задач

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

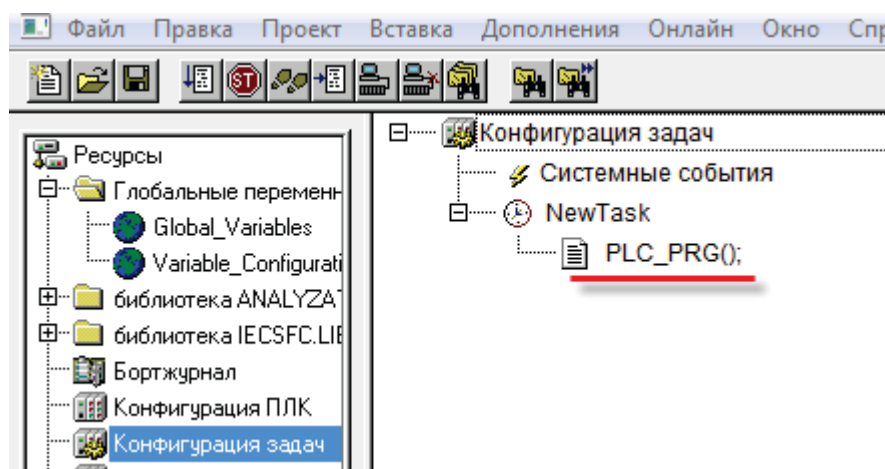


Рис. 9. Привязка программы **PLC\_PRG** к задаче

### 3. Настройки СП307 (Modbus RTU Master)

#### 3.1. Настройки PLC-порта панели

Для того чтобы использовать панель **СП307** в режиме **Modbus RTU Master**, 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 панель не сможет записывать данные в ПЛК (но сможет их считывать).

**Обратите внимание** на наличие галочки **Переставить WORD в FLOAT** – при ее отсутствии переменная типа **REAL** будет отображаться некорректно.

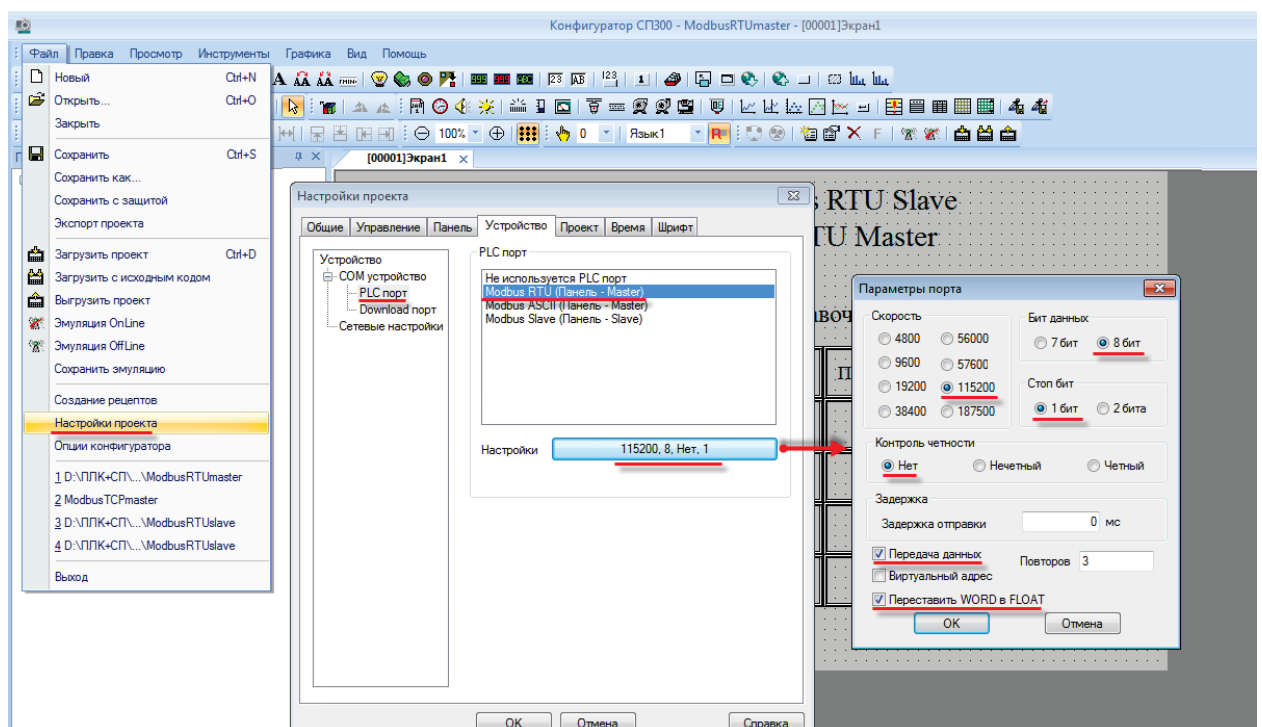


Рис. 10. Настройки **PLC-порта** панели (режим **Modbus RTU Master**)

### 3.2. Экран проекта

Проект содержит один экран, на котором осуществляется отображение данных, которые считываются из ПЛК, и ввод данных, которые записываются в ПЛК; 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 ПЛК **0x0**. Панель считывает значение этого регистра, а также записывает значение в этот регистр (т.е. в переменную **bInputOutput** типа **BOOL**) по команде пользователя.
2. **Цифровой ввод** с привязанным регистром ПЛК **4x1**. Панель считывает значение этого регистра, а также записывает значение в этот регистр (т.е. в переменную **wInputOutput** типа **WORD**) по команде пользователя.
3. **Цифровой ввод** с привязанными регистрами ПЛК **4x2**. Панель считывает значение этого регистра, а также записывает значение в этот регистр (т.е. в переменную **rInputOutput** типа **REAL**) по команде пользователя.
4. **Текстовый ввод** с привязанными регистрами ПЛК **4x4**. Панель считывает значение этого регистра, а также записывает значение в этот регистр (т.е. в переменную **sInputOutput** типа **STRING**) по команде пользователя.

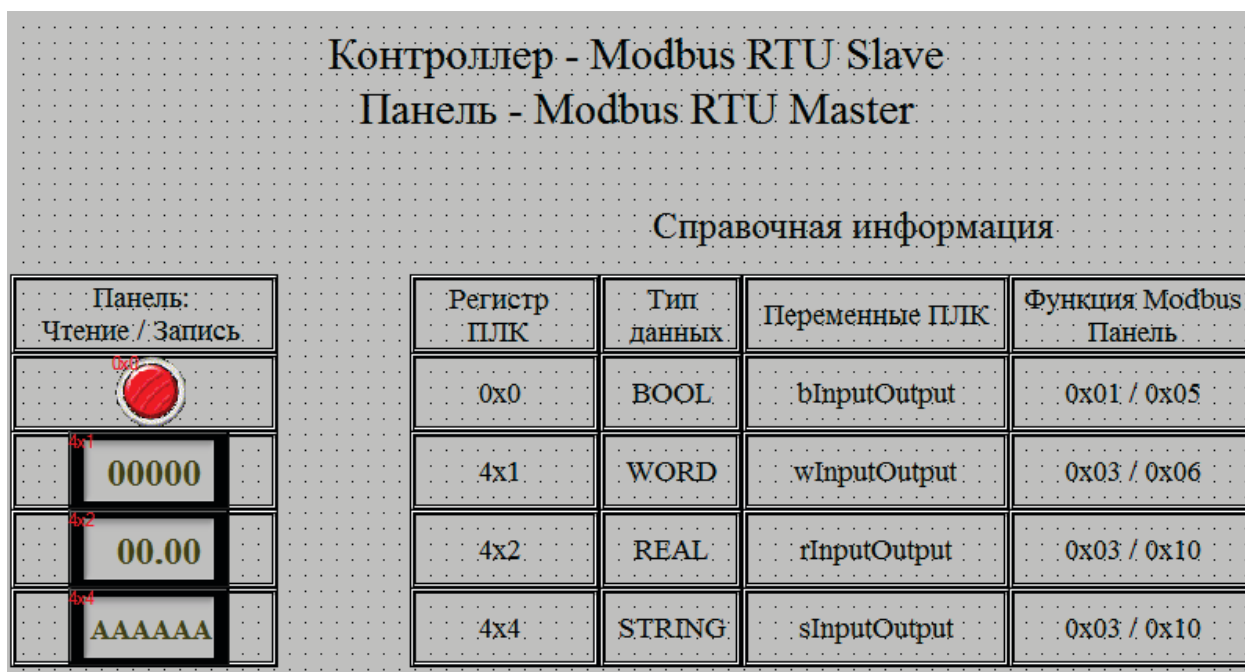


Рис. 11. Внешний вид экрана проекта

Настройки элементов приведены ниже.

### 3.2.1. 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. 12. Внешний вид элемента **Переключатель с индикацией**

В настройках элемента на вкладке **Регистр элемента** выберем устройство **ПЛК110-32**, укажем для него адрес **1** и регистр **0x0** (согласно [табл. 2](#)):

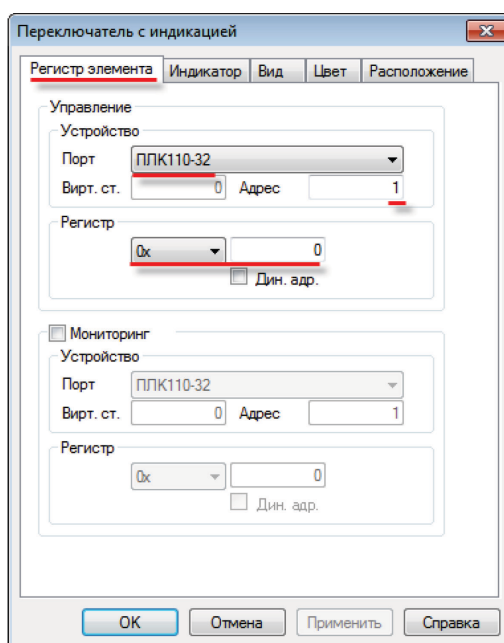


Рис. 13. Настройки элемента **Переключатель с индикацией**, вкладка **Регистр элемента**

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

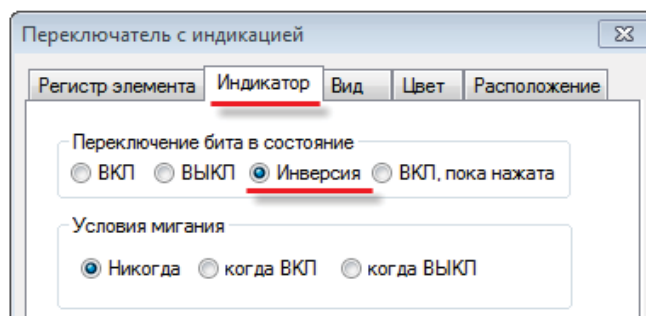


Рис. 14. Настройки элемента **Переключатель с индикацией**, вкладка **Индикатор**

### 3.2.2. 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.

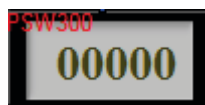


Рис. 15. Внешний вид элемента **Цифровой ввод**

В настройках элемента на вкладке **Регистр элемента** выберем устройство **ПЛК110-32**, укажем для него адрес **1** и регистр **4x1** (согласно [табл. 2](#)). **Обратите внимание**, что параметр **Тип** должен иметь значение **WORD**.

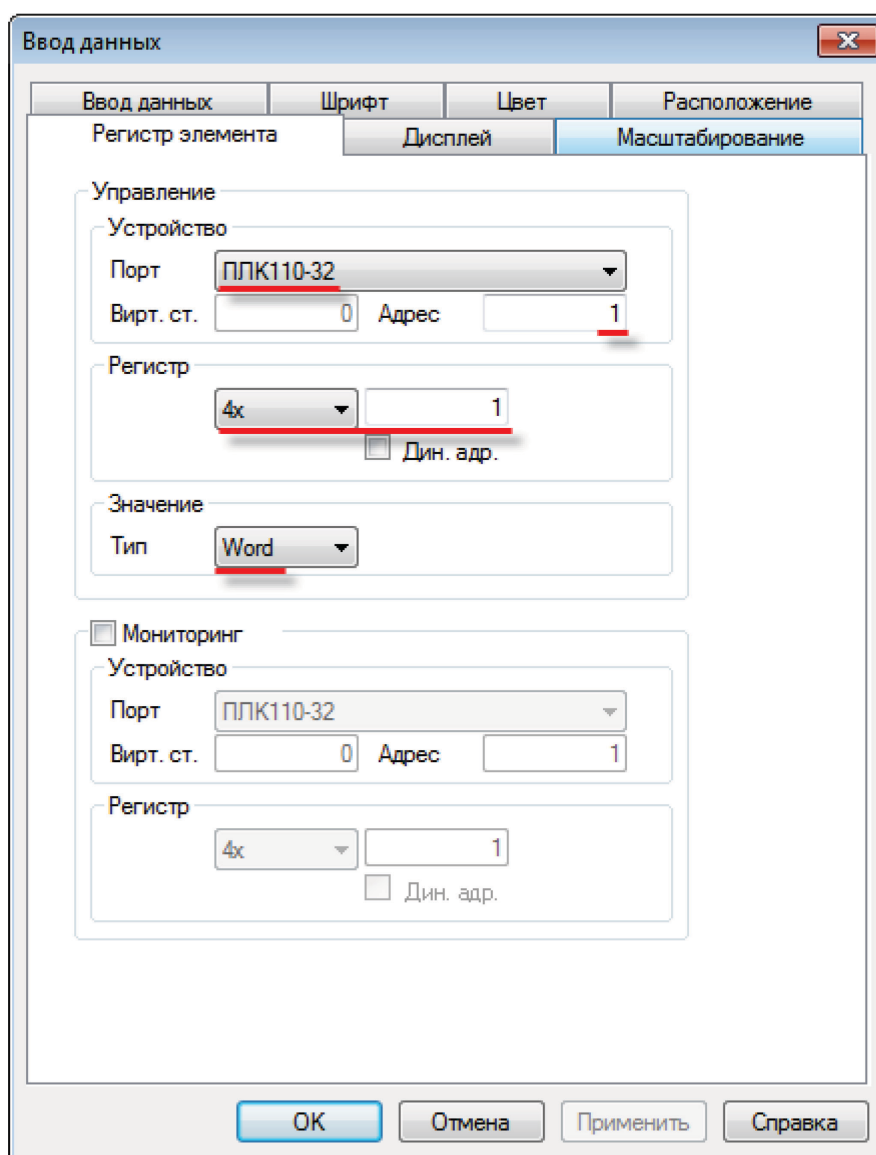


Рис. 16. Настройки элемента **Цифровой ввод**, вкладка **Регистр элемента**

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

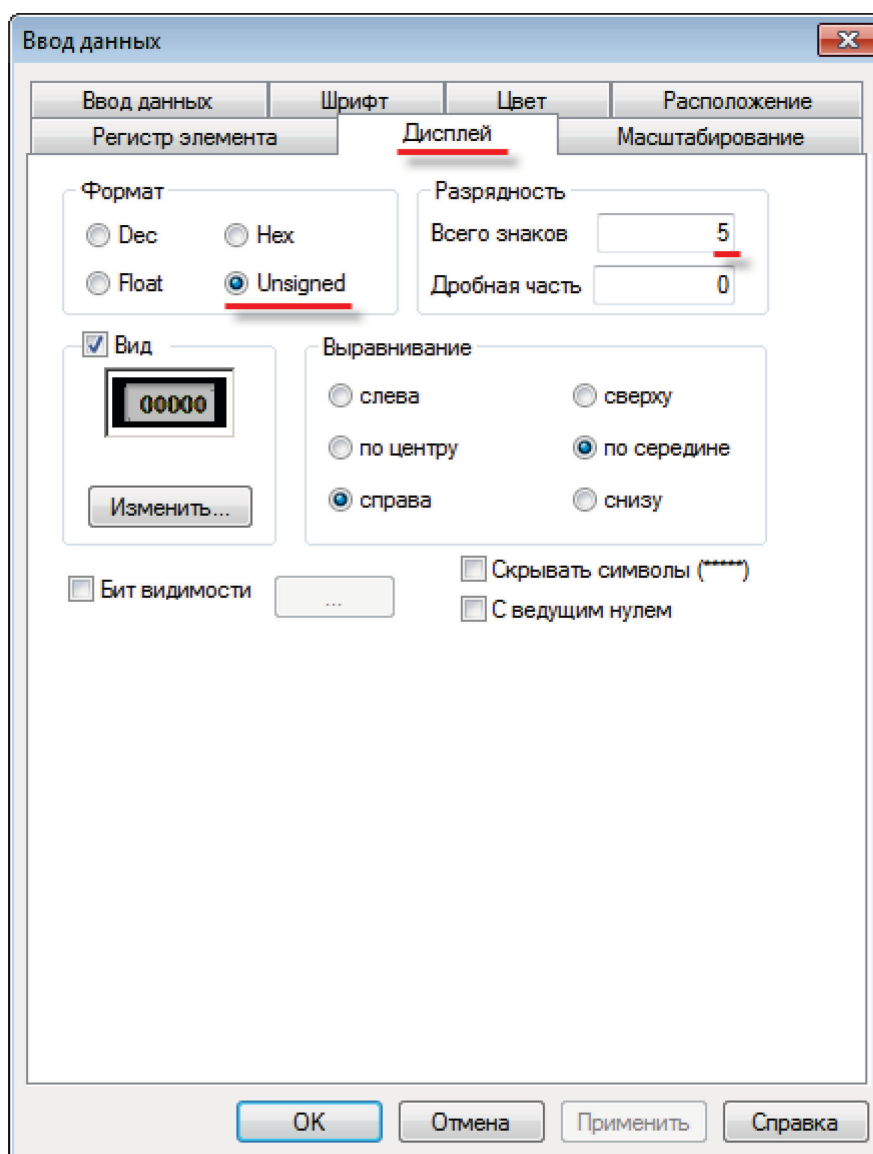


Рис. 17. Настройки элемента **Цифровой ввод**, вкладка **Дисплей**

**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**.

### 3.2.3. 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. 18. Внешний вид элемента **Цифровой ввод**

В настройках элемента на вкладке **Регистр элемента** выберем устройство **ПЛК110-32**, укажем для него адрес **1**, регистр **4x2** и тип **DWORD** (поскольку переменные типа **REAL** занимают два регистра). В результате, элемент будет обращаться к 2 и 3 регистру ПЛК.

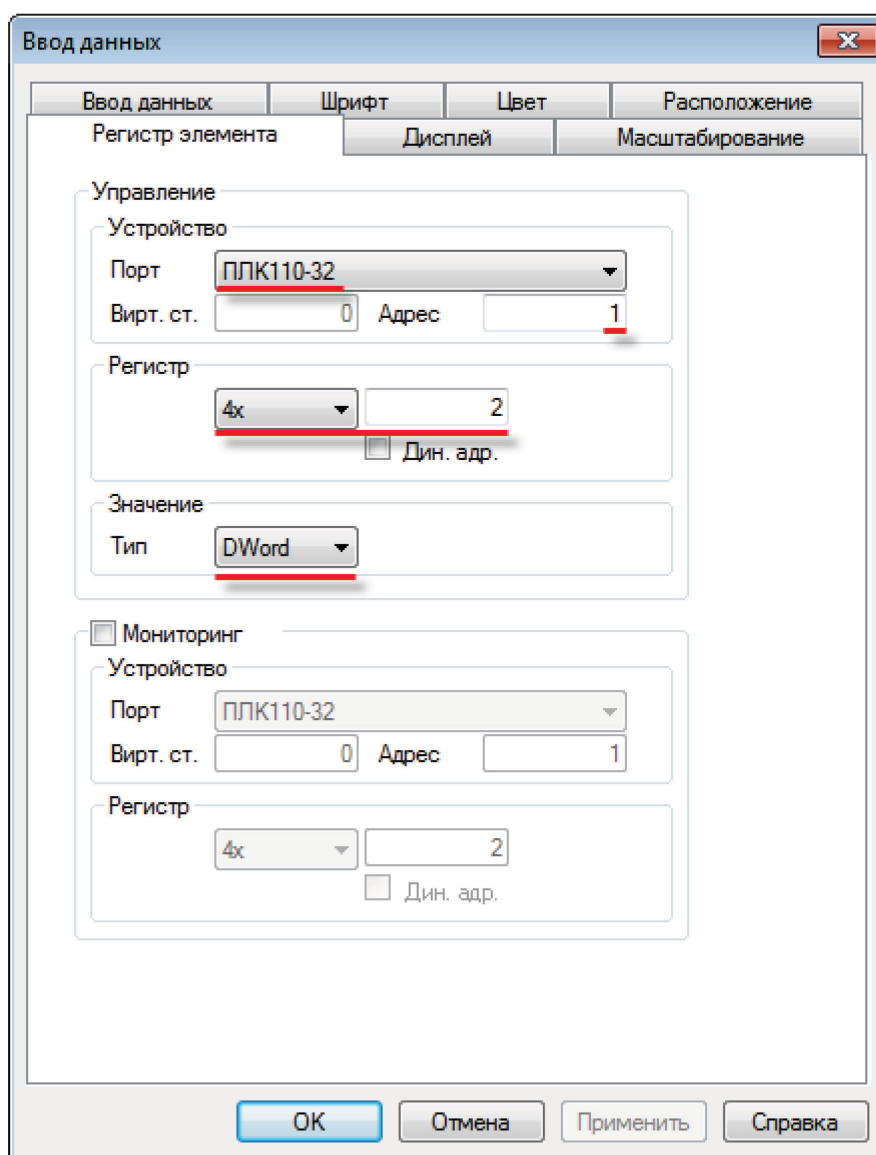


Рис. 19. Настройки элемента **Цифровой ввод**, вкладка **Регистр элемента**

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.

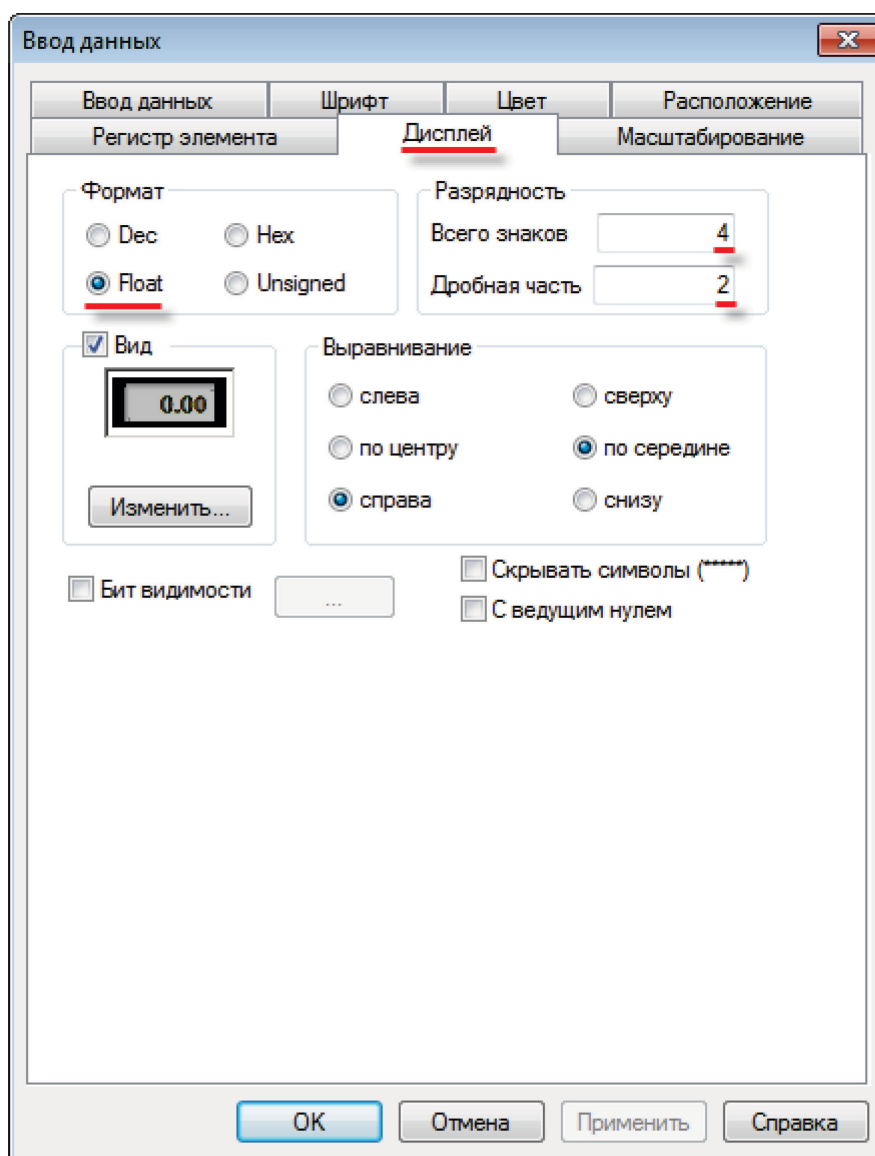


Рис. 20. Настройки элемента **Цифровой ввод**, вкладка **Дисплей**



### 3.2.4. Текстовый ввод (STRING)

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



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

В настройках элемента на вкладке **Регистр элемента** выберем устройство **ПЛК110-32**, укажем для него адрес **1**, регистр **4x4** и кол-во регистров **3**. В результате, элемент будет обращаться к трем регистрам ПЛК, начиная с регистра **4** (т.е. к регистрам **4-6**). Каждый байт может содержать два символа, поэтому с помощью элемента будет осуществляться отображение и ввод текстовых строк длиной до **6** символов.

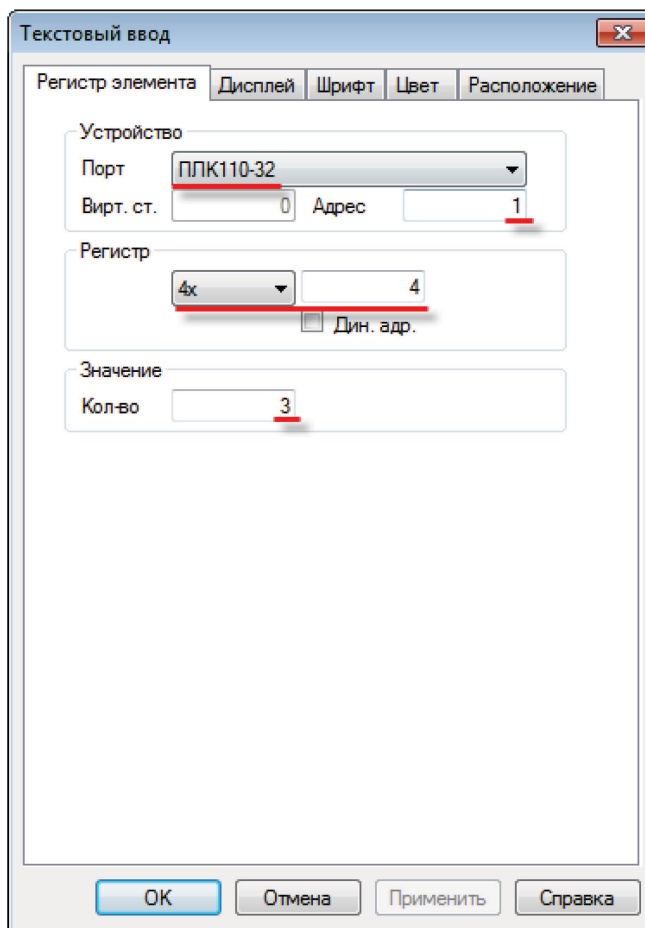


Рис. 22. Настройки элемента **Текстовый ввод**, вкладка **Регистр элемента**

**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 **ModbusRTUslave.pro** в **CoDeSys**, подключитесь к ПЛК, загрузите проект и запустите его:

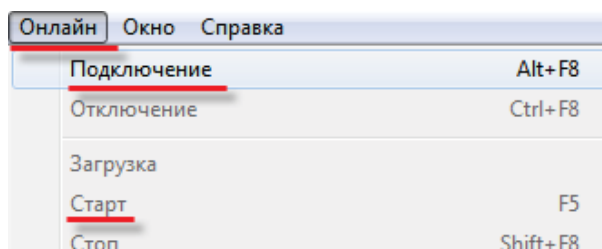


Рис. 23. Загрузка и запуск проекта для ПЛК в **CoDeSys**

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

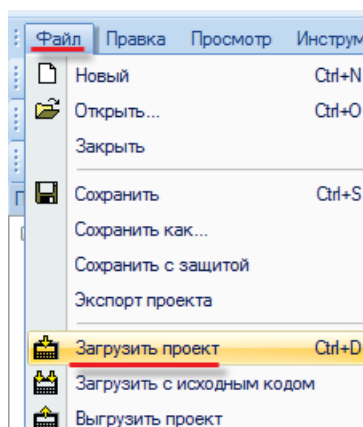


Рис. 24. Загрузка проекта в панель

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

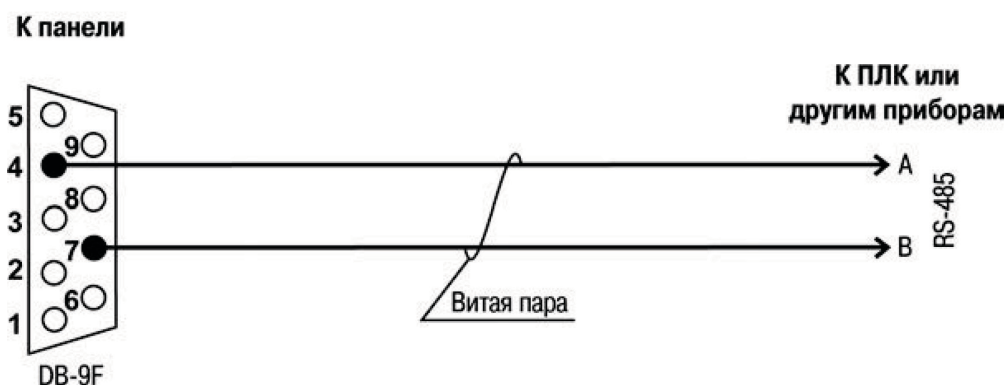


Рис. 25. Соединительный кабель для связи панели с ПЛК по интерфейсу **RS-485**

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

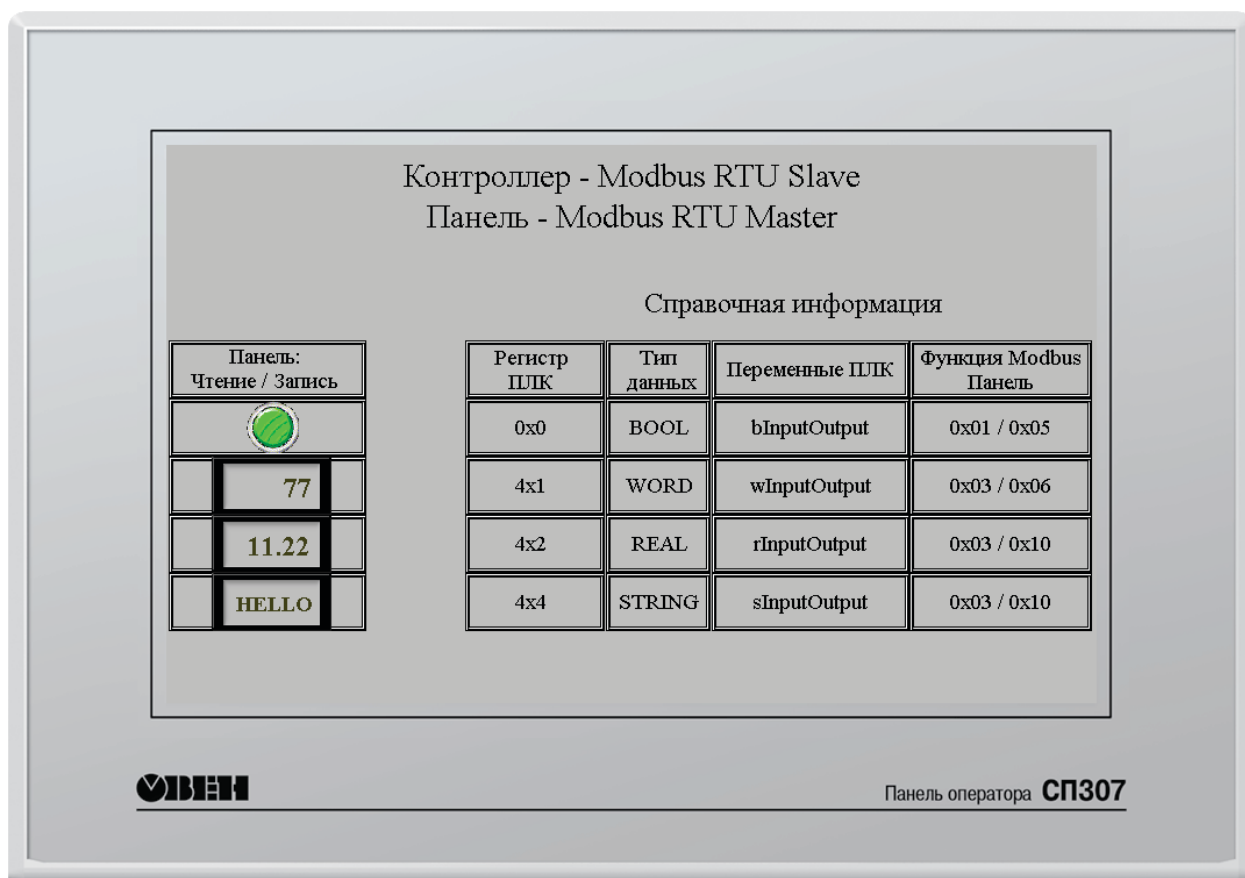


Рис. 26. Ввод значений с помощью сенсорного дисплея панели

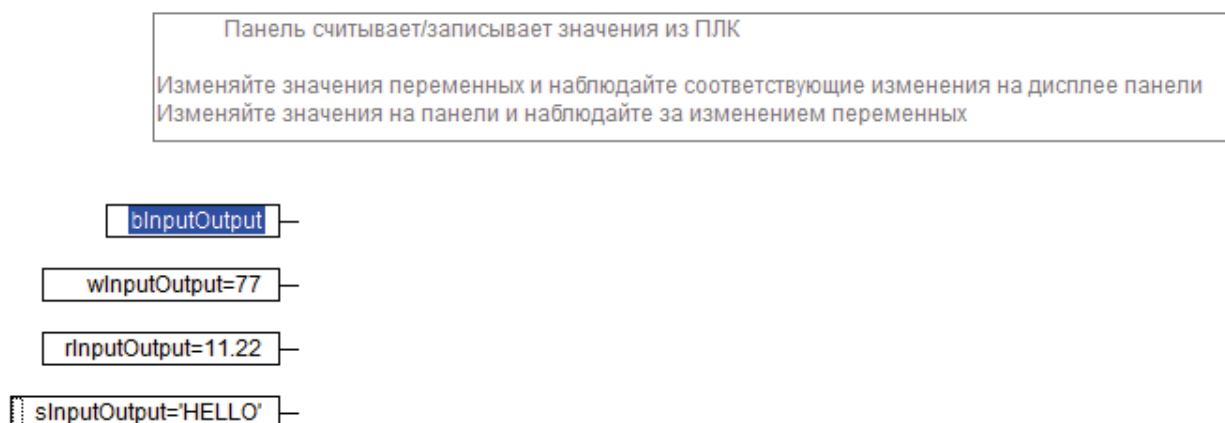


Рис. 27. Отображение введенных на панели значений в **CoDeSys**

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.28. Изменение значений переменных в **CoDeSys**

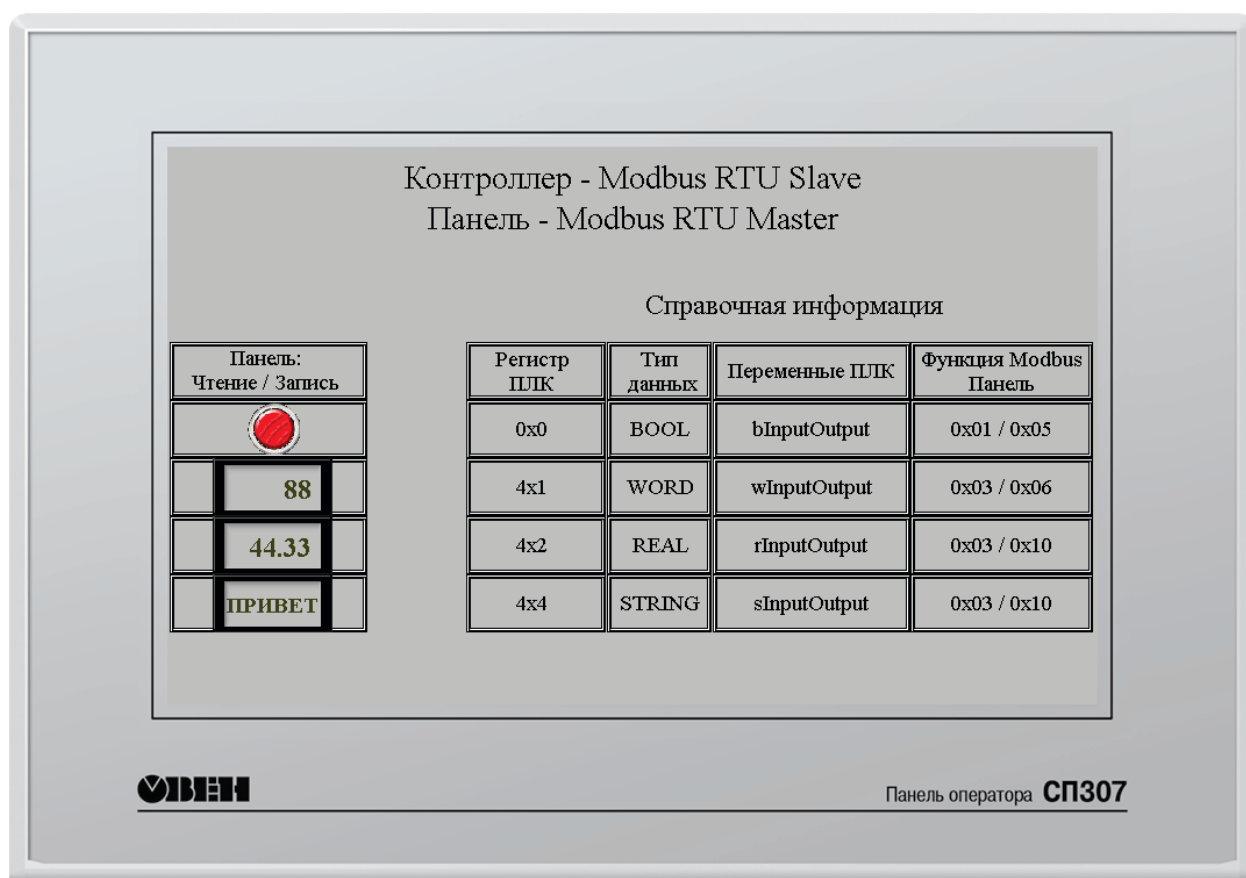


Рис. 29. Отображение введенных в **CoDeSys** значений на панели