

## Оглавление

1. Описание примера .....	3
2. Основные сведения о протоколе Modbus RTU .....	4
3. ПР200 – Modbus RTU Slave, СП310 – Modbus RTU Master .....	7
3.1. Формулировка задачи.....	7
3.2. Создание проекта для ПР200 .....	8
3.3. Создание проекта для СП310 .....	14
4. СП310 – Modbus RTU Slave, ПР200 – Modbus RTU Master .....	23
4.1. Формулировка задачи.....	23
4.2. Создание проекта для СП310 .....	24
4.3. Создание проекта для ПР200 .....	28
5. Работа с примером.....	35

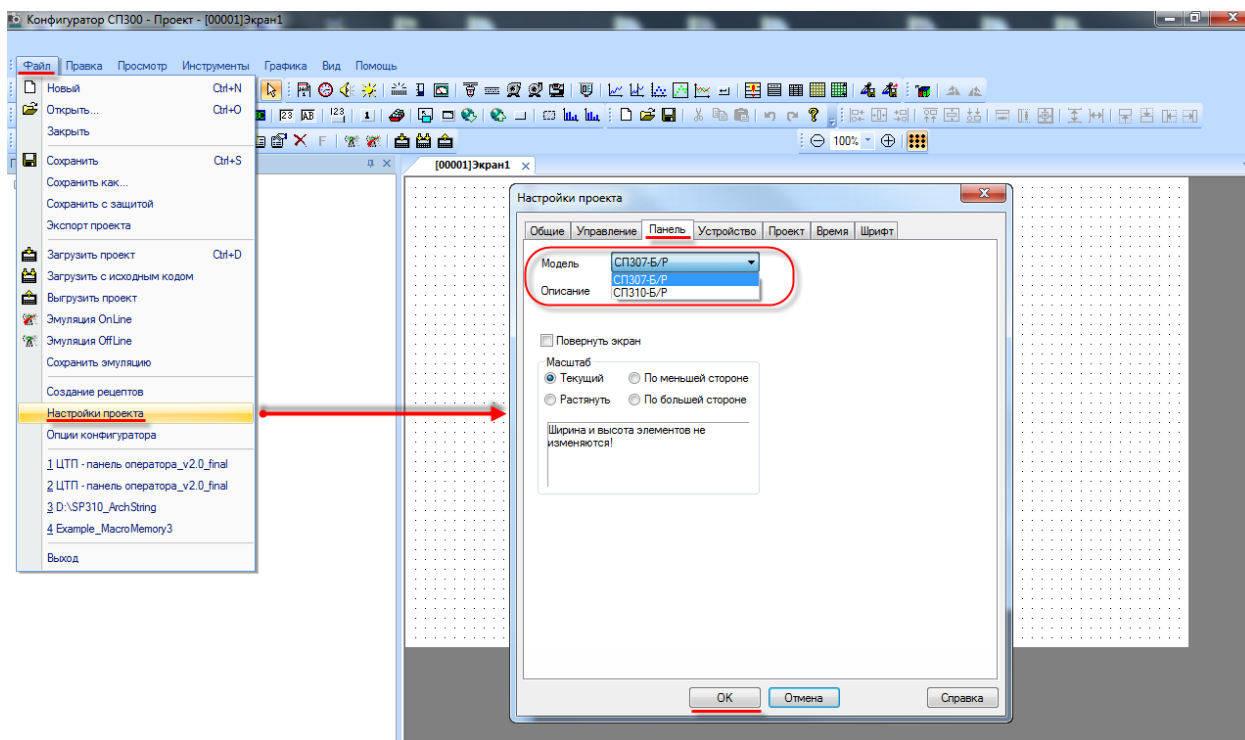
## 1. Описание примера

Этот пример посвящен настройке обмена данными между программируемым реле **ПР200** и панелью оператора **СП310** по протоколу **Modbus RTU**.

Подразумевается, что пользователь ознакомлен с **РЭ** и **РП** на соответствующие приборы и имеет базовые навыки работы с ними. Также перед началом работы с примерами рекомендуется прочитать [п. 2](#) данного документа (основные сведения о протоколе **Modbus**). Рекомендуется изучать документ строго последовательно, поскольку его основные разделы ([п. 3](#) и [п. 4](#)) тесно связаны друг с другом.

Проекты, доступные вместе с примером, созданы для панели оператора **СП310** и реле **ПР200-24.2(4)**. При необходимости загрузить их в другие модификации устройств, необходимо сделать следующее:

1. для панели оператора в программе **Конфигуратор СП300** в меню **Файл** выбрать пункт **Настройки проекта**, после чего на вкладке **Панель** выбрать нужную модификацию панели и нажать **ОК**.



2. для программируемого реле создать отдельный проект для модификации с нужным типом и количеством входов-выходов и настроить его согласно инструкциями из данного документа.

## 2. Основные сведения о протоколе Modbus RTU

Modbus RTU – широко распространенный в промышленности коммуникационный протокол, реализуемый поверх последовательных интерфейсов **RS-232/RS-485** и основанный на архитектуре **Master-Slave** (ведущий-ведомый).

**Master** (мастер, ведущее устройство) является инициатором обмена и может считывать и записывать данные в slave-устройства. **Slave** (слэйв, подчиненное устройство) не может инициализировать обмен.

При работе с Modbus по интерфейсам RS-232/485 в сети может находиться только одно master-устройство и несколько slave-устройств (согласно стандарту – до **32-х** без использования повторителей, до **247-ми** с использованием повторителя после каждых 32-х устройств).

Все устройства в пределах одной сети должно иметь одинаковые сетевые настройки (скорость, кол-во значащих бит, четность, кол-во стоп бит). Все slave-устройства должны иметь **уникальные адреса** в диапазоне **1...247**. Адрес **0** используется для широковещательной рассылки (рассылки, которую получают все slave-устройства). **Очевидно**, что master-устройство не нуждается в адресе, поскольку slave-устройства не могут его опрашивать.

При запросе master-устройство обращается к одной из **областей памяти** slave-устройства с помощью **определенной функции**. **Область памяти** характеризуется типом хранящихся в ней значений (биты/регистры) и типом доступа (только чтение/чтение и запись). Стандарт Modbus определяет 4 области памяти:

Табл. 2.1. Области данных протокола **Modbus**

Область данных	Обозначение	Тип данных	Тип доступа
Coils (Регистры флагов)	0x	BOOL	чтение/запись
Discrete Inputs (Дискретные входы)	1x	BOOL	только чтение
Input Registers (Регистры ввода)	3x	WORD	только чтение
Holding Registers (Регистры хранения)	4x	WORD	чтение/запись

Каждая область памяти состоит из определенного (зависящего от конкретного устройства) количества ячеек. Каждая ячейка имеет уникальный адрес. Для конфигурируемых устройств (таких как ТРМ, ПЧВ и т.д.) производитель предоставляет **карту регистров**, в которой содержится информация о соответствии параметров устройства и их адресов. Для программируемых устройств пользователь формирует такую карту самостоятельно с помощью среды разработки. Существуют устройства, в которых сочетаются оба рассмотренных случая – у их карты регистров есть фиксированная часть, которую пользователь может дополнить в соответствии со своей задачей (очевидно, что адреса ячеек при этом не должны пересекаться).

**Стоит отметить**, что в некоторых устройствах области памяти наложены друг на друга (например, **0x** и **4x**) – т.е. пользователь сможет обращаться разными функциями к одним и тем же ячейкам памяти.

**Функция** определяет операцию (чтение/запись) и область памяти, с которой эта операция будет произведена. Ниже приведен список наиболее часто используемых функций:

Табл. 2.2. Основные функции протокола **Modbus**

Код функции	Имя функции	Выполняемая команда
1 (0x01)	Read Coil Status	Чтение значений из нескольких регистров флагов
2 (0x02)	Read Discrete Inputs	Чтение значений из нескольких дискретных входов
3 (0x03)	Read Holding Registers	Чтение значений из нескольких регистров хранения
4 (0x04)	Read Input Registers	Чтение значений из нескольких регистров ввода
5 (0x05)	Force Single Coil	Запись значения в один регистр флага
6 (0x06)	Preset Single Register	Запись значения в один регистр хранения
15 (0x0F)	Force Multiple Coils	Запись значений в несколько регистров флагов
16 (0x10)	Preset Multiple Registers	Запись значений в несколько регистров хранения

**Обратите внимание**, что нельзя смешивать понятия области памяти и функции. У начинающих пользователей часто возникают проблемы при работе с **input** и **holding** регистрами, поскольку **область памяти holding регистров** имеет обозначение **4х**, а **функция чтения holding регистров** – **0x03** (может интуитивно показаться, что номера области памяти и функции должны совпадать – но это не так).

Ниже приведен фрагмент карты регистров для модуля аналогового ввода **MB110-8A**. В ней для каждого параметра приведен адрес и тип данных (тип данных определяет число ячеек памяти, занимаемых параметром). В таблице не упомянуто, в какой области памяти расположены параметры – но в примечании указано, что обращаться к ним необходимо функциями **0x03** и **0x04** – из чего можно сделать вывод, что области памяти **4х** и **3х** в устройстве наложены друг на друга.

Окончание таблицы В.4

Параметр	Тип	Адрес регистра	
		(hex)	(dec)
Положение десятичной точки в целом значении для входа 2 (знач. <b>DP</b> )	int16	0006	6
Целое значение измерения входа 2 со смещением точки	int16	0007	7
Статус измерения входа 2 (код исключительной ситуации)	int16	0008	8
Циклическое время измерения входа 2	int16	0009	9
Измерение входа 2 в представлении с плавающей точкой	Float32	000A,000B	10,11
...			
Положение десятичной точки в целом значении для входа 8 (знач. <b>DP</b> )	int16	002A	42
Целое значение измерения входа 8 со смещением точки	int16	002B	43
Статус измерения входа 8 (код исключительной ситуации)	int16	002C	44
Циклическое время измерения входа 8	int16	002D	45
Измерение входа 8 в представлении с плавающей точкой	Float32	002E,002F	46,47

**Примечания**

1 Все регистры только на чтение, чтение регистров осуществляется командами 03 или 04 (прибор поддерживает обе команды).

2 При передаче 4-х байтных значений (тип Float 32) старшее слово передается в регистре с меньшим номером.

Рис. 2.1. Фрагмент карты регистров модуля **MB110-8A**

Стоит отметить, что в различных документах идентичные обозначения могут иметь разный смысл в зависимости от контекста. Например, префикс **0x** часто используют как указание на 16-ричную систему счисления, поэтому в одном случае **0x30** может обозначать “30-й бит области памяти **coils**”, а в другом – “адрес 30 в 16-ричной (HEX) системе счисления” (при этом данный адрес может относиться к любой области памяти).

Другой пример необходимости уточнения контекста – принцип адресации регистров. В некоторых случаях в адреса битов/регистров закладывается префикс области памяти, в которых они находятся, например – **30101** (цифра **3** указывает на **input регистры**), **40202** (цифра **4** указывает на **holding регистры**). При этом обычно подразумевается, что адрес **30001** соответствует нулевому input регистру, а **40001** – нулевому holding регистру. Иными словами, при опросе упомянутых регистров (**30101** и **40202**) в настройках master-устройства следует указать, что необходимо обращаться к **input регистру** с адресом **100** и **holding регистру** с адресом **201**.

В то же время существуют устройства, для которых адрес **40202** может являться адресом любой области памяти (например, **coil** номер **40202**). В рамках данного документа энергонезависимые регистры панелей **СПЗхх** имеют адреса типа **10400** – эти адреса являются абсолютными, т.е. в них не заложен префикс области памяти **1x** – хранящиеся в них значения расположены в области памяти **holding регистров**.

Приведенные выше примеры позволяют сделать вывод, что система обозначений для адресов битов/регистров slave-устройств зависит от конкретного производителя (в некоторых случаях – даже для конкретного документа), в связи с чем от читателя требуется четкое понимание контекста используемых обозначений и повышенное внимание к примечаниям, сноскам и т.п.

Ниже приведен пример различных вариантов обозначений для **holding регистра** с адресом **39**:

- регистр **4x39**
- регистр **39**, чтение осуществляется командой **03**
- регистр **0x27**, чтение осуществляется командой **0x03**
- регистр **40040**

Опрос slave-устройства может быть **одиноким** или **групповым**. При **одиночном опросе** master-устройство считывает каждый из параметров slave-устройства отдельной командой. При **групповом опросе** master-устройство считывает одной командой сразу несколько параметров, чьи адреса в карте регистров расположены строго последовательно и не имеют разрывов. Групповой опрос позволяет уменьшить трафик в сети и время, затрачиваемое на опрос устройства, но в некоторых случаях его применение невозможно (или возможно с ограничениями) из-за индивидуальных особенностей устройства.

### 3. PR200 – Modbus RTU Slave, СП310 – Modbus RTU Master

#### 3.1. Формулировка задачи

Настроим обмен между **PR200** и **СП310** по протоколу **Modbus RTU**. PR200 будет выполнять функцию **slave**-устройства, СП310 – **master**-устройства.

Основные характеристики используемых устройств приведены в табл. 3.1. Используемые в примере переменные и их адреса описаны в табл. 3.2.

Табл. 3.1. Характеристики используемых в примере устройств

Устройство	PR200-24.2(4)	СП310
Функция	Slave	Master
Используемый порт	RS-485 (слот 2)	PLC
Настройки обмена	115200, 8 бит, 1 стоп бит, без контроля четности	
Slave ID	1	-
Среда разработки проекта	Owen Logic 1.8.93.7745	Конфигуратор СП300 (V2.D3d-3)
Название файла проекта	PR200_ ModbusExample_OW1893.owl	SP310_ModbusExample.txp

Табл. 3.2. Список используемых в примере переменных

PR200			СП310			Комментарий
Программа		Визуализация	Тип данных	Тип данных	Регистр элемента	
Имя	Адрес	Имя*				
X_S_СП1	512	X_S_ПР1	INT	BOOL	4x512.0	Одиночный запрос; только запись
I_S1	513	I_S1	INT	WORD	4x513	Одиночный запрос
F_S1	514-515	F_S1	FLOAT	FLOAT	4x514	Одиночный запрос
X_S_СП2	520	X_S_ПР2	INT	BOOL	PSW320	Групповой запрос; только чтение
X_S_СП2	521	X_S_ПР2	INT	BOOL	PSW321	Групповой запрос; только чтение
I_S2	522	I_S2	INT	WORD	PSW322	Групповой запрос
I_S3	523	I_S3	INT	WORD	PSW323	Групповой запрос
F_S2	530-531	F_S2	FLOAT	FLOAT	PSW330-331	Групповой запрос
F_S3	532-533	F_S3	FLOAT	FLOAT	PSW332-333	Групповой запрос
F_S4	534-535	F_S4	FLOAT	FLOAT	PSW334-335	Групповой запрос
F_S5	536-537	F_S5	FLOAT	FLOAT	PSW336-337	Групповой запрос

\* - к визуализации привязана либо переменная программы (ее адрес указан), либо внутренняя переменная, у которой нет адреса

Приведенная в табл. 3.2 информация может сразу вызвать ряд вопросов (например, почему в визуализации ПР в некоторых случаях требуются дополнительные переменные, почему типы данных в ПР и панели не соответствуют друг другу, почему отличаются обозначения регистров панели и т.д.). Ответы на них будут приведены по мере создания проекта.

### 3.2. Создание проекта для ПР200

Для того чтобы использовать ПР200 в режиме **Modbus RTU Slave**, необходимо в меню **Прибор** выбрать пункт **Настройки прибора** и на вкладке **Интерфейсы** настроить режимы работы последовательных портов. В нашем примере мы будем использовать **Слот 2**. Настроим его в качестве **Modbus Slave** согласно [табл. 3.1](#): скорость – **115200**, бит данных – **8**, стоп бит – **1**, контроль четности – **нет**.

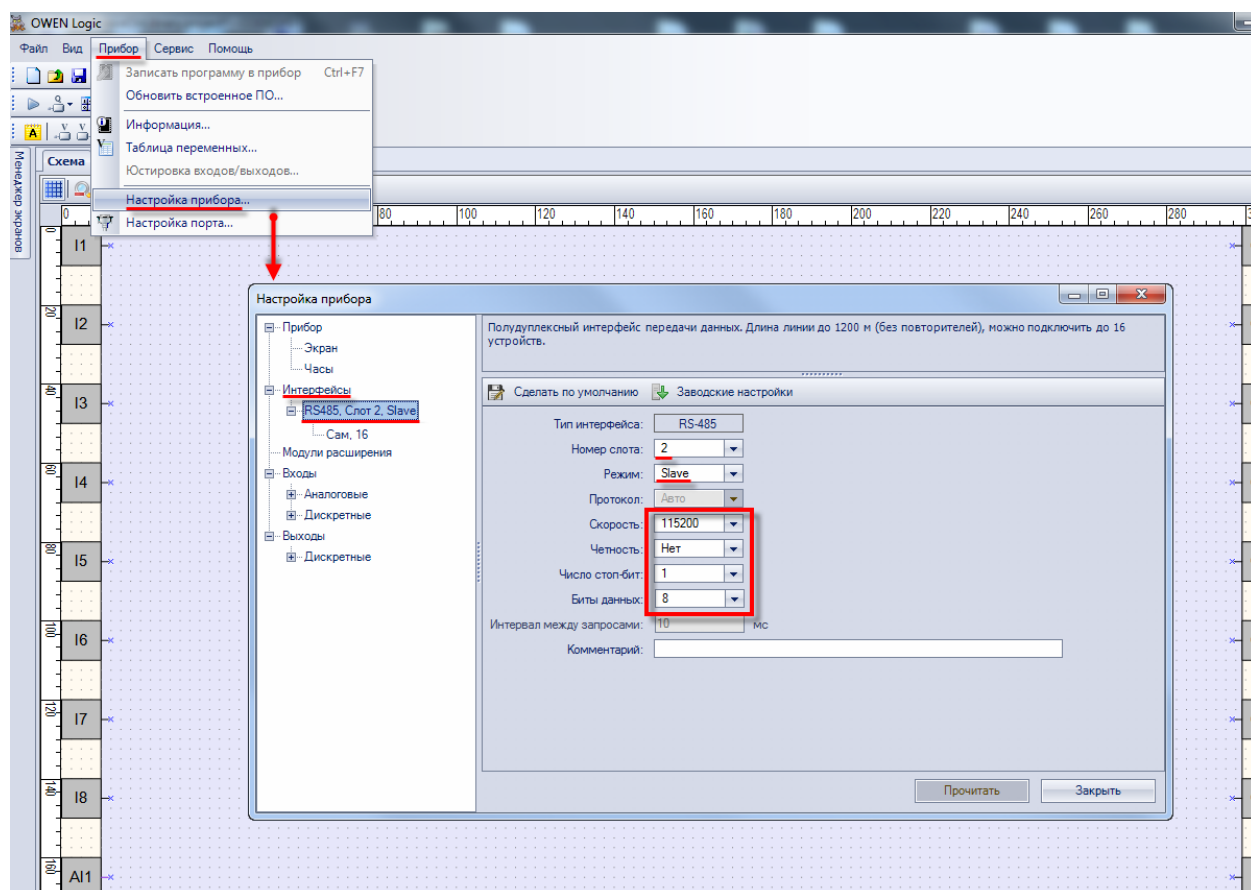


Рис. 3.1. Настройка Слота 2 RS-485 (режим Modbus RTU Slave)

Во вкладке устройства, встроенного в **Слот 2**, укажите его имя и адрес. В соответствии с [табл. 3.1](#) адрес ПР200 имеет адрес **1**.

Создайте в устройстве сетевые переменные в соответствии с [табл. 3.2](#). Таким образом будет сформирована карта регистров панели. Все сетевые переменные являются **holding**-регистрами. **Напомним**, что переменная с плавающей точкой занимает два регистра.

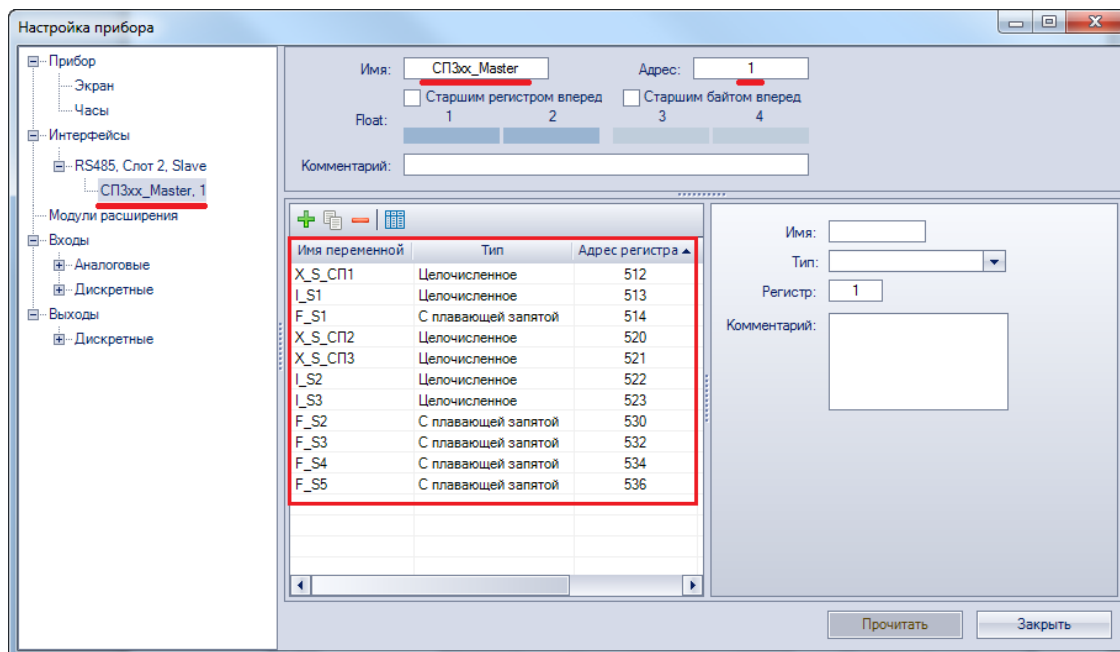


Рис. 3.2. Настройка адреса слота и создание сетевых переменных

**Обратите внимание**, что нельзя создать сетевую переменную типа **BOOL**. Поэтому биты передаются в виде значения типа **INT**, из которого в программе ПР200 извлекается значащий бит.

В меню **Прибор** выберите пункт **Таблица переменных** и добавьте в таблицу три внутренние переменные типа **BOOL** (в соответствии с [табл. 3.2](#)). Несмотря на то, что для каждой создаваемой переменной тип **BOOL** предлагается по умолчанию, необходимо выбирать его с помощью выпадающего списка – иначе переменная не будет добавлена в таблицу.

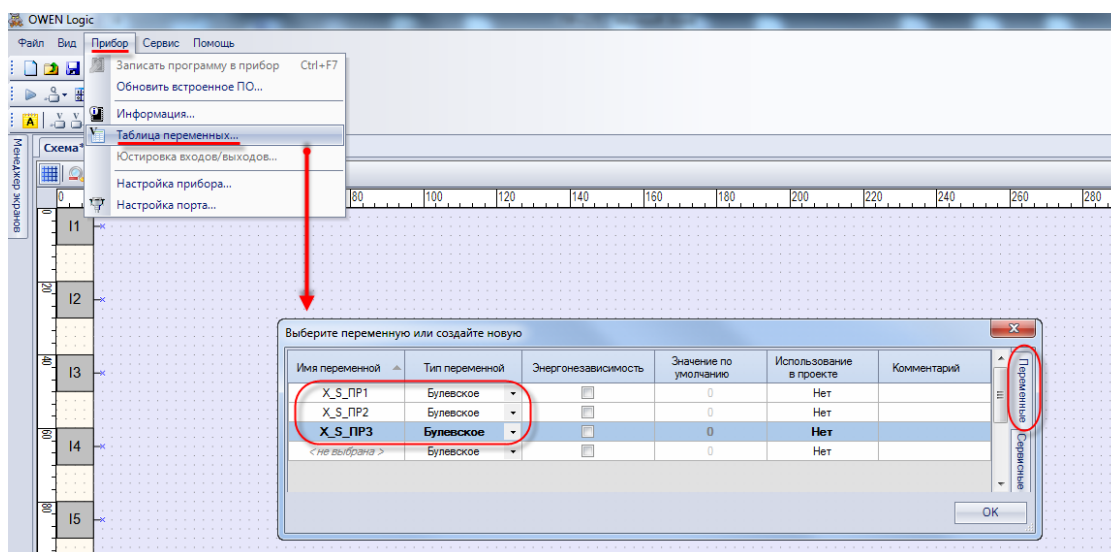


Рис. 3.3. Создание внутренних переменных типа **BOOL**



Разместим созданные сетевые и внутренние переменные на холсте.

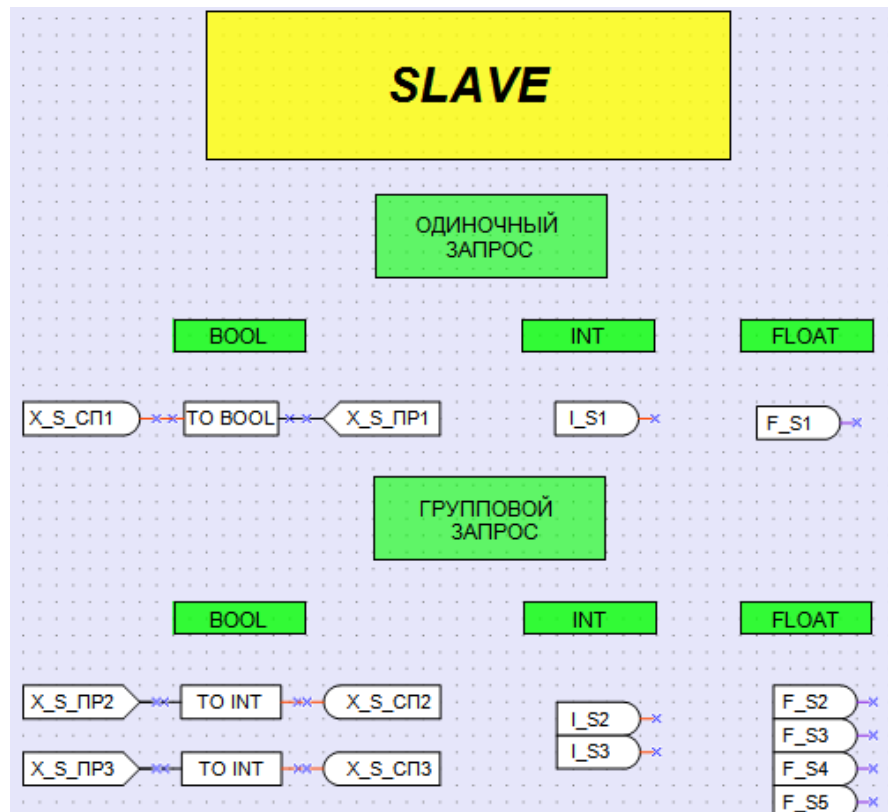






Рис. 3.4. Размещение переменных на холсте

Переменные групп **INT** и **FLOAT** являются входными сетевыми переменными (). Они будут записываться панелью, а также могут быть изменены с дисплея ПР200, после чего панель СП310 считает их новые значения.

Переменная **X\_S\_СП1** также является входной сетевой переменной. **Напомним**, что поскольку в ПР200 для режима **Slave** нет возможности создать сетевую переменную типа **BOOL**, нам придется выполнить преобразование с помощью оператора **TO\_BOOL**. Результат преобразования запишем в выходную внутреннюю переменную **X\_S\_ПР1** () , которую будем использовать на экране визуализации. В данном случае полученное от панели значение не может быть изменено с дисплея ПР200; для реализации этого необходимо настроить однократную запись значения переменной **X\_S\_ПР1** в переменную **X\_S\_СП1** по изменению. Это приведет к чрезмерному усложнению программы, поэтому в примере данный функционал рассмотрен не будет.

Переменные **X\_S\_ПР2** и **X\_S\_ПР3** являются входными внутренними переменными () , значения которых будут изменяться с дисплея ПР200. После этого они будут преобразовываться оператором **TO\_INT** и записываться в выходные сетевые переменные () **X\_S\_СП2** и **X\_S\_СП3**, значения которых будут считываться панелью СП310. При этом изменить их с дисплея панели будет нельзя – поскольку они тут же будут перезаписаны значениями, заданными на дисплее ПР200. Как уже упоминалось, решение подобной проблемы сопряжено с чрезмерным усложнением программы, поэтому в примере данный функционал рассмотрен не будет.

Итак, мы разместили на холсте все необходимые элементы. **Отметим**, что программа не содержит никакого алгоритма – только отображение и изменение значений, которые будут записываться и считываться панелью оператора СП310. Эти значения пользователь может использовать при реализации алгоритма, выполняющего его задачу.

Создадим экран визуализации, на котором будем отображать и изменять упомянутые значения. Откроем **Менеджер экранов**, зададим экрану по умолчанию имя **SLAVE** и увеличим количество его строк до **11-ти**.

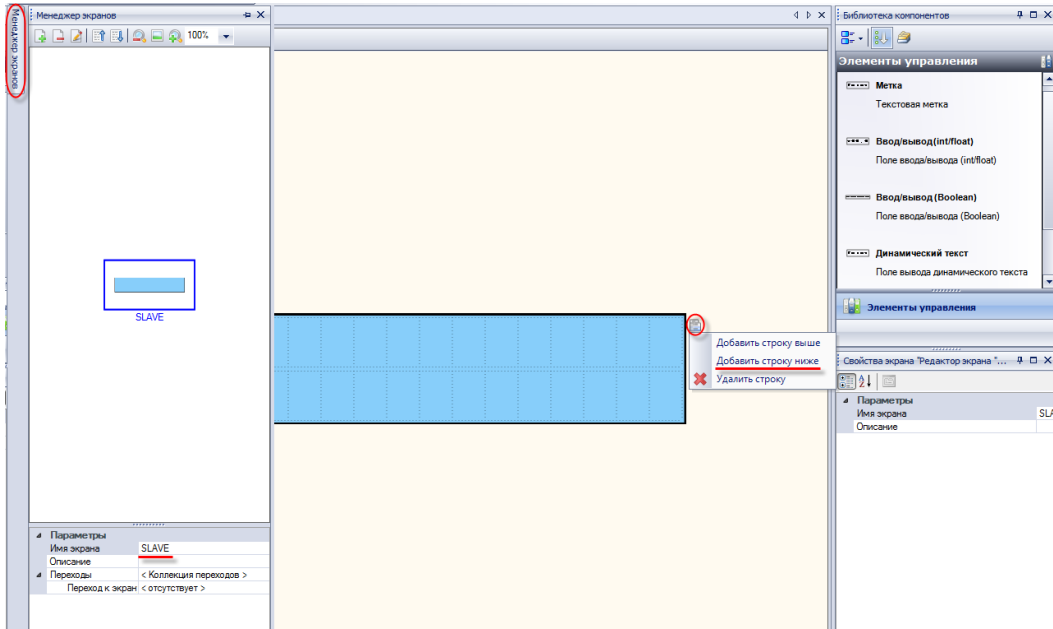


Рис. 3.5. Добавление строк на экран визуализации

Экран будет содержать 3 элемента **Ввод/вывод (Boolean)**, 8 элементов **Ввод/вывод (int/float)**, а также 4 **Метки** с поясняющими надписями. Настройки элементов приведены ниже.

				S L A V E			
B O O L				I n t			
X 1 = Н е т				I 1 = 0 0 0 0			
X 2 = Н е т				I 2 = 0 0 0 0			
X 3 = Н е т				I 3 = 0 0 0 0			
				F L O A T			
				F 1 = + 0 0 , 0 0			
				F 2 = + 0 0 , 0 0			
				F 3 = + 0 0 , 0 0			
				F 4 = + 0 0 , 0 0			
				F 5 = + 0 0 , 0 0			

Рис. 3.6. Внешний вид экрана визуализации **SLAVE**

К элементу **X1** привяжем внутреннюю переменную **X\_S\_ПР1** и зададим тексты для ее значений **TRUE/FALSE**. Для параметра **Редактируемо** установим значение **Нет** (см. комментарии к рис 3.4).

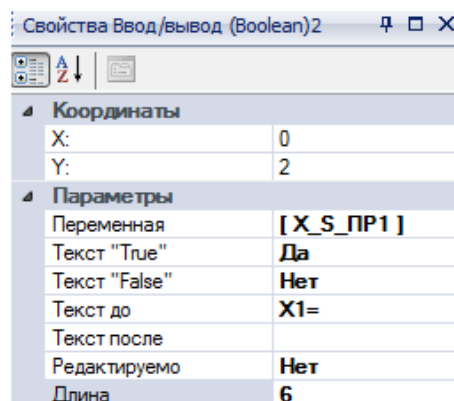
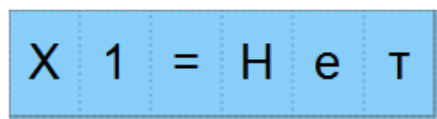


Рис. 3.7. Настройка элемента Ввод/вывод (Boolean) **X1**

К элементам **X2** и **X3** привяжем переменные **X\_S\_ПР2** и **X\_S\_ПР3**, настроив их по аналогии с элементом **X1**. Для их параметров **Редактируемо** установим значение **Да** для возможности изменения значения переменной с дисплея ПР200.

К элементу **I1** привяжем сетевую переменную **I\_S1** и зададим ей тип **Целый**. Для параметра **Редактируемо** установим значение **Да** для возможности изменения значения переменной с дисплея ПР200.

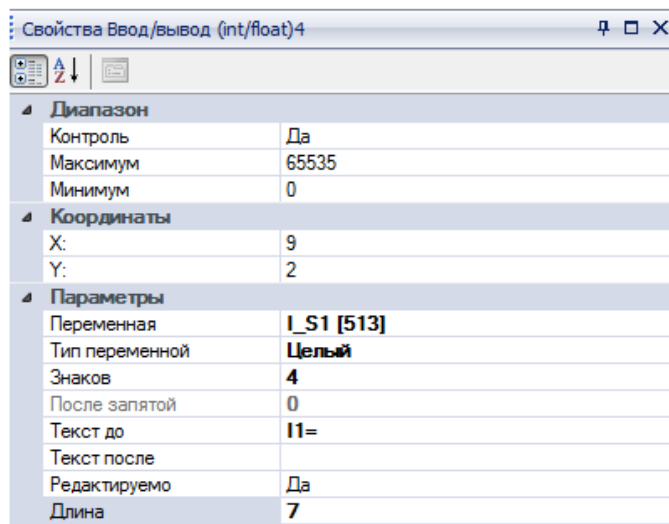
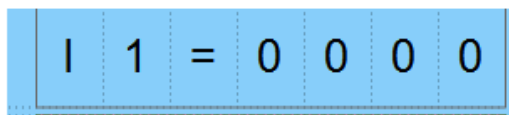


Рис. 3.8. Настройка элемента Ввод/вывод (int/float) **I1**

К элементам **I2** и **I3** привяжем переменные **I\_S2** и **I\_S3**, настроив их по аналогии с элементом **I1**.

К элементу **F1** привяжем сетевую переменную **F\_S1** и зададим ей тип **Вещественный**. Для параметра **Редактируемо** установим значение **Да** для возможности изменения значения переменной с дисплея ПР200.

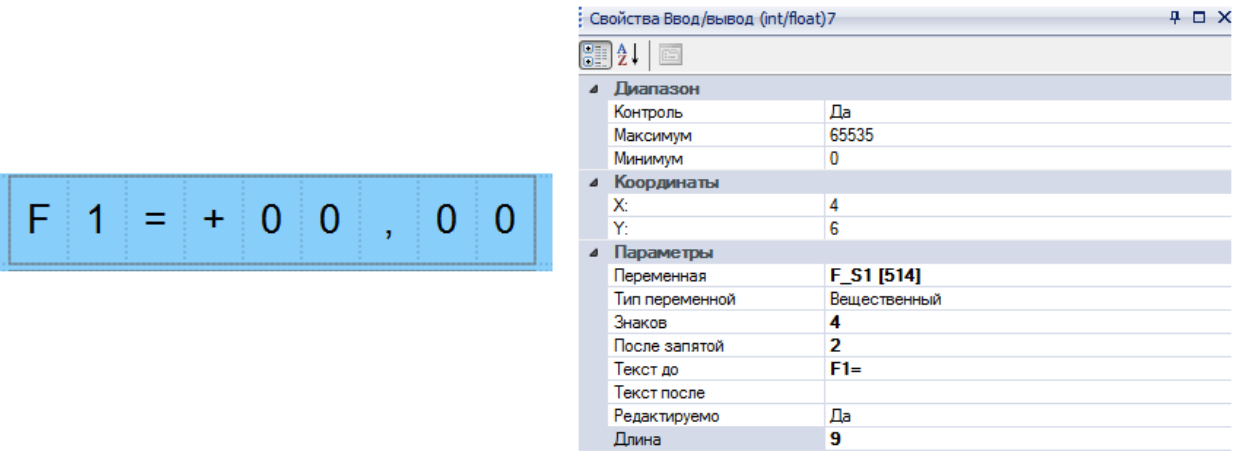


Рис. 3.9. Настройка элемента Ввод/вывод (int/float) **F1**

К элементам **F2**, **F3**, **F4** и **F5** привяжем переменные **F\_S2**, **F\_S3**, **F\_S4** и **F\_S5**, настроив их по аналогии с элементом **F1**.

На этом создание проекта для ПР200 завершено.

### 3.3. Создание проекта для СП310

Для того чтобы использовать **СП310** в режиме **Modbus RTU Master** необходимо в меню **Файл** выбрать пункт **Настройки проекта** и на вкладке **Устройство** настроить режимы работы последовательных портов. В нашем примере мы будем использовать **PLC-порт**. Настроим его в качестве **Modbus RTU Master** согласно [табл. 3.1](#): скорость – **115200**, бит данных – **8**, стоп бит – **1**, контроль четности – **нет**. Поставим галочку **Передача данных** для возможности записи в регистры ПР200 и галочку **Переставить WORD в FLOAT** для автоматического выбора используемого для панели порядка регистров у значений с плавающей точкой, считанных по Modbus.

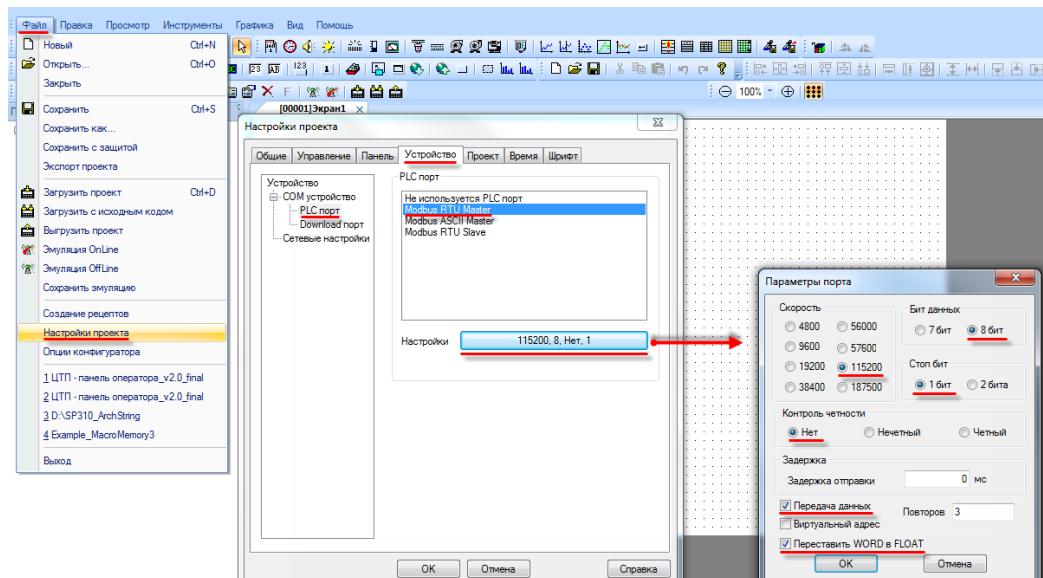


Рис. 3.10. Настройка **PLC-порта** (режим **Modbus RTU Master**)

Добавим на экран панели элемент **Переключатель с индикацией**, два элемента **Индикатор**, восемь элементов **Цифровой ввод** и две **Функциональные области**. Настройки элементов приведены ниже. Пояснительные надписи сделаны с помощью элементов **Статический текст**.

СП307 - Modbus RTU Master (порт PLC)			ПР200 - Modbus RTU Slave (слот 2) адрес 1					
Опрос с помощью элементов			Групповой опрос с помощью функциональной области (копирование данных в регистры панели)					
Регистр ПР	Тип	Элемент	Регистр ПР	Тип	Элемент	Регистр ПР	Тип	Элемент
512 (X_S_СП1)	INT		520 (X_S_СП2)	INT		530-531 (F_S2)	FLOAT	
513 (I_S1)	INT		521 (X_S_СП3)	INT		532-533 (F_S3)	FLOAT	
514-515 (F_S1)	FLOAT		522 (I_S2)	INT		534-535 (F_S4)	FLOAT	
			523 (I_S3)	INT		536-537 (F_S5)	FLOAT	

Рис. 3.11. Внешний вид экрана панели

В примере рассмотрены два способа опроса slave-устройства – с помощью элементов визуализации (через **одиночные запросы**) и с помощью функциональных областей (**групповые запросы**).

При опросе через элементы пользователь настраивает для каждого элемента порт, к которому подключено slave-устройство, адрес устройства, область памяти slave-устройства и адрес опрашиваемого регистра/бита из этой области памяти. Панель при этом автоматически выбирает нужную функцию Modbus. **Стоит заметить**, что чтение происходит циклически, а запись - однократно при нажатии на элемент. Считанные значения сразу отображаются на дисплее, не попадая во внутреннюю память панели – т.е. возможности с ними работать (например, отмасштабировать с помощью макроса или опросить с помощью OPC-сервера) у пользователя нет.

При опросе через функциональную область панель считывает/записывает сразу группу последовательно расположенных регистров slave-устройства в свою внутреннюю память, после чего можно отобразить значения этих регистров с помощью элементов панели, указав в них соответствующие адреса внутренней памяти. Поскольку значения попадают во внутреннюю память панели, у пользователя есть возможность обрабатывать их и считывать по другим интерфейсам.

Для начала рассмотрим реализацию опроса через элемента.

Элемент **Переключатель с индикацией** будет использоваться для записи значения в сетевую переменную PR200 X\_S\_CP1. В его свойствах на вкладке **Регистр элемента** выберем **PLC-порт** и адрес PR200 – напомним, что он равен **1** (см. рис. 3.2). Укажем область памяти **4x** (holding-регистры), адрес регистра **512** и адрес бита **00**. Таким образом, элемент будет производить запись в нулевой бит **512-го** регистра PR200, которому соответствует сетевая переменная **X\_S\_CP1** типа **INT**. С помощью оператора **TO\_BOOL** из этой переменной будет выделен значащий нулевой бит, который будет записан в внутреннюю переменную **X\_S\_PR1** – она будет отображена на дисплее PR200.

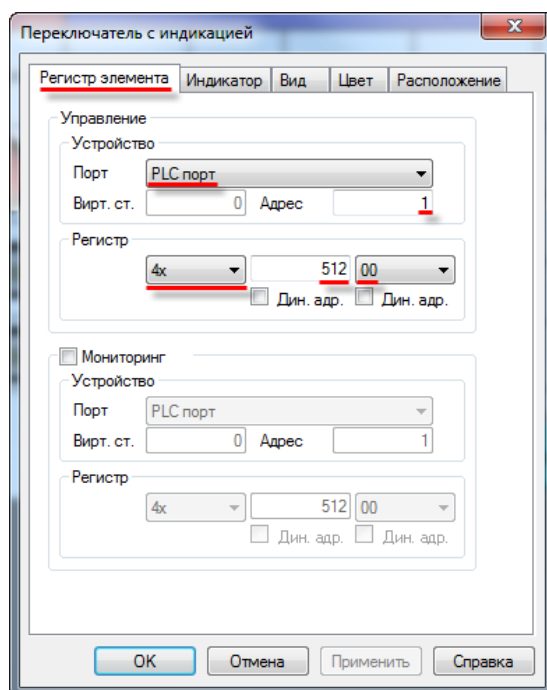


Рис. 3.12. Настройка элемента **Переключатель с индикацией**

Элемент **Цифровой ввод** (INT) будет использоваться для чтения и записи сетевой переменной ПР200 **I\_S1**. В его свойствах на вкладке **Регистр элемента** выберем **PLC-порт** и адрес **1**. Укажем область памяти **4x** (holding-регистры) и адрес регистра **513**. Таким образом, элемент будет производить опрос **513-ro** регистра ПР200, которому соответствует сетевая переменная **I\_S1** типа **INT**.

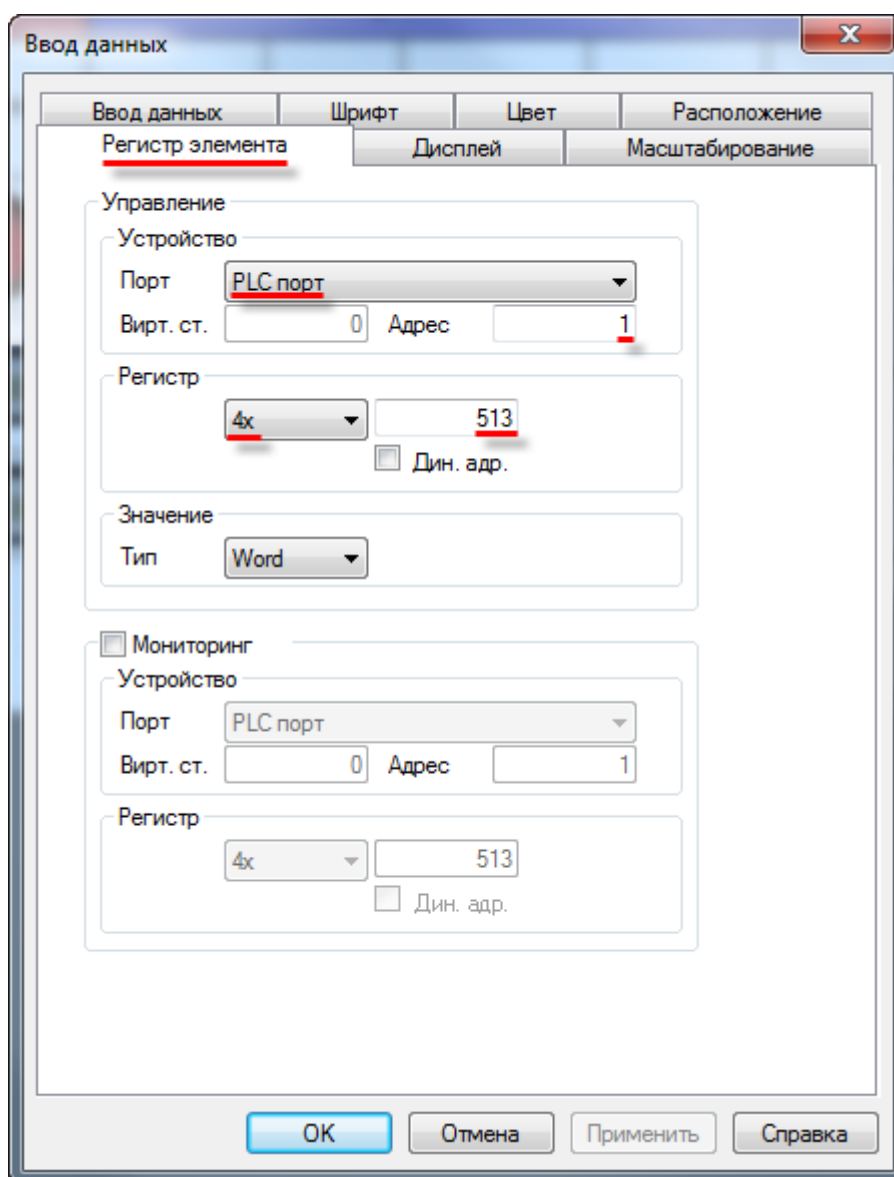


Рис. 3.13. Настройка элемента **Цифровой ввод** (INT)

Элемент **Цифровой ввод (FLOAT)** будет использоваться для чтения и записи сетевой переменной ПР200 **I\_F1**. В его свойствах на вкладке **Регистр элемента** выберем **PLC-порт** и адрес **1**. Укажем область памяти **4x** (holding-регистры), адрес регистра **514** и тип **DWORD** (поскольку значение с плавающей точкой занимает два регистра). Таким образом, элемент будет производить опрос **514-515-го** регистра ПР200, которым соответствует сетевая переменная **F\_S1** типа **FLOAT**.

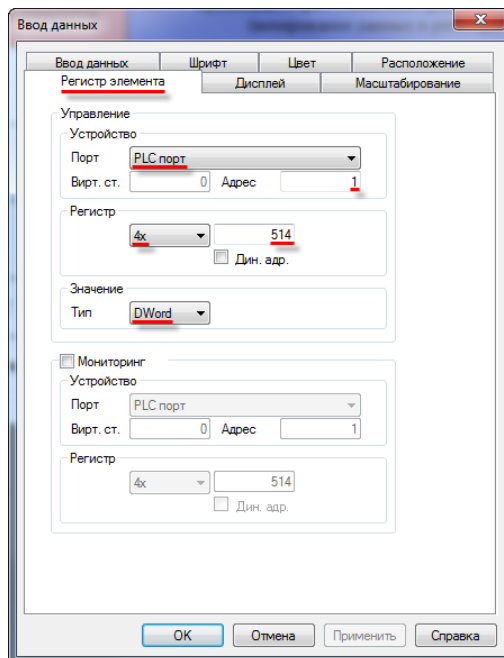


Рис. 3.14. Настройка элемента **Цифровой ввод (FLOAT)**, вкладка **Регистр элемента**

На вкладке **Дисплей** укажем формат отображаемых элементом значений (**Float**) и число знаков после запятой.

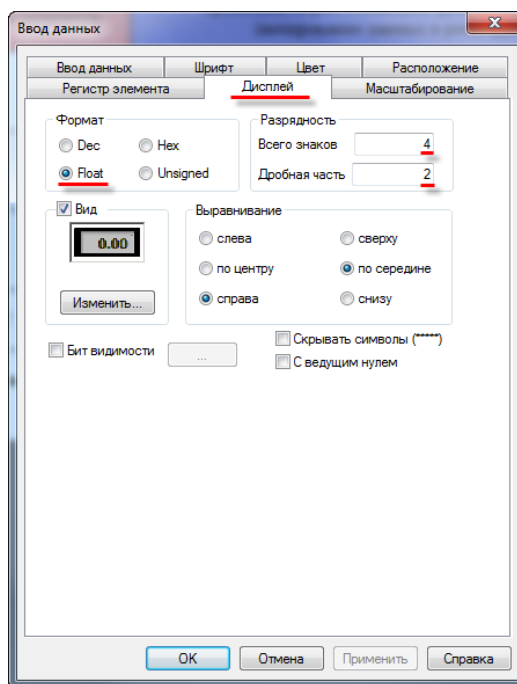


Рис. 3.15. Настройка элемента **Цифровой ввод (FLOAT)**, вкладка **Дисплей**



Перейдем к настройке **группового опроса**. Он будет осуществляться с помощью двух функциональных областей – одна из них будет реализовать чтение, другая – запись.

Настроим функциональную область для чтения. Выберем для нее режим запуска **Цикл** с частотой 1 секунда – соответственно, раз в секунду СП310 будет опрашивать ПР200.

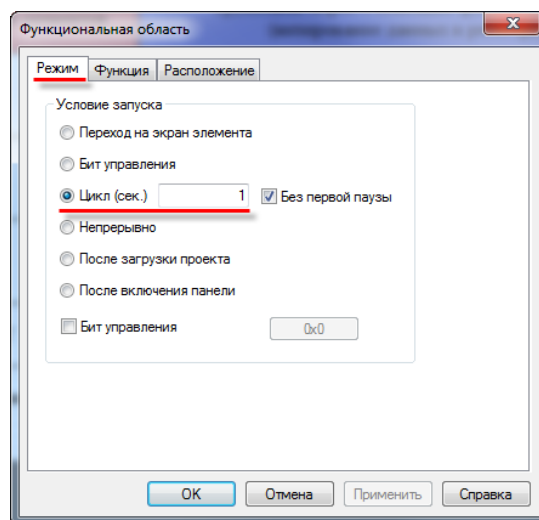


Рис. 3.16. Настройка элемента **Функциональная область** (чтение)

На вкладке **Функция** добавим два действия **Копировать группу регистров**. Одно будет использоваться для группового опроса **INT** переменных ПР200, второе – для **FLOAT** переменных. **Необходимо отметить**, что каждая из команд группового опроса ПР200 должна запрашивать/записывать **не более 13-ти регистров** (т.е. не более **13-ти INT переменных** или **6-ти FLOAT переменных**). Это ограничения вызвано размером сетевого буфера ПР200. Для первого действия укажем **PLC-порт**, адрес **1**, адрес источника – область памяти **4х** ПР200, адрес первого опрашиваемого регистра – **520**, адрес назначения – **внутренние регистры** панели, адрес первого регистра назначения – **PSW320**, количество опрашиваемых регистров – **4**. Таким образом, данное действие будет копировать регистры ПР200 с адресами **520-523** во внутренние регистры панели **PSW320-323**.

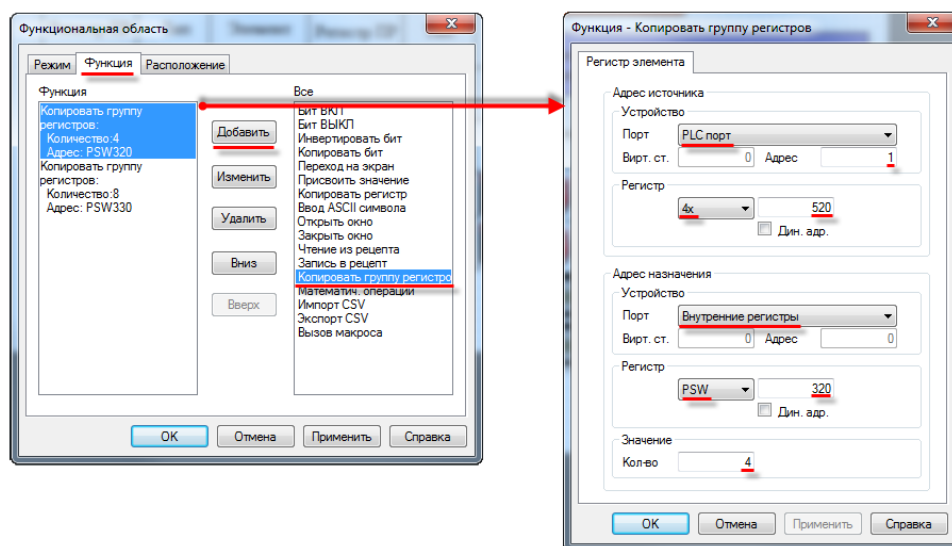


Рис. 3.17. Настройка элемента **Функциональная область** (чтение), вкладка **Функция**, действие 1

Для второго действия укажем адрес источника – область памяти **4х** PR200, адрес первого опрашиваемого регистра – **530**, адрес назначения – **внутренние регистры** панели, адрес первого регистра назначения – **PSW330**, количество опрашиваемых регистров – **8**. Таким образом, данное действие будет копировать регистры PR200 с адресами **530-537** в памяти во внутренние регистры панели **PSW330-337**.

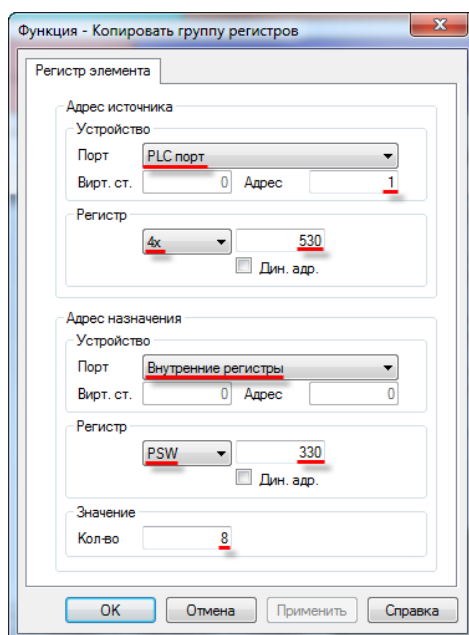


Рис. 3.18. Настройка элемента **Функциональная область** (чтение), вкладка **Функция**, действие 2

Теперь настроим вторую функциональную область, которая будет производить групповую запись в регистры PR200. Выберем для нее режим запуска **Бит управления** с привязанным битом **PSB256**. Запись будет происходить однократно по переднему фронту данного бита. **Очевидно**, мы не можем в данном случае запускать функциональную область циклически, так как это лишит нас возможности изменять значения с дисплея PR200 (т.к. они будут сразу перезаписаны панелью).

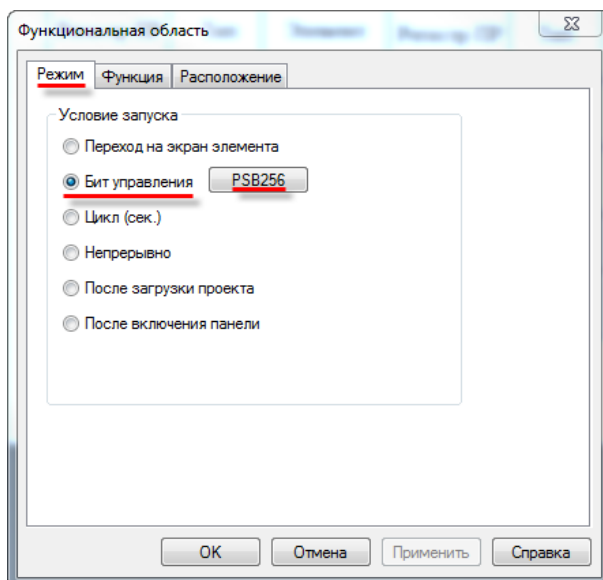


Рис. 3.19. Настройка элемента **Функциональная область** (запись)

На вкладке **Функция** добавим два действия **Копировать группу регистров**, а после них – действие **Бит ВЫКЛ** с привязанным битом **PSB256**. Таким образом, после каждой операции записи бит **PSB256** будет сбрасываться в **FALSE**, тем самым отключая функциональную область. Включаться же он будет при вводе новых значений в элементы панели (см. рис. 3.23, параметр **Бит уведомления**).

Для первого действия **Копировать группу регистров** укажем **PLC-порт**, адрес **1**, адрес источника – внутренняя память панели **4x**, адрес первого записываемого регистра – **PSW322**, адрес назначения – область памяти **4x** PR200, адрес первого регистра назначения – **522**, количество записываемых регистров – **2**. Таким образом, данное действие будет копировать регистры панели **PSW322-323** в регистры PR200 **522-523**.

Вполне предсказуем вопрос – почему мы записываем только два регистра, ведь считываем с PR200 четыре? Напомним, что в регистры PR200 **520** и **521** попадают значения типа **BOOL**, изменяемые с дисплея PR200 и конвертируемые оператором **TO\_INT**. Поскольку присвоение выполняется циклически, то даже если панель отправит в соответствующие регистры значения, они сразу будут перезаписаны значениям с дисплея PR200. Как уже упоминалось, решение этой проблемы излишне усложнило бы пример, поэтому мы ограничимся записью только целочисленных переменных.

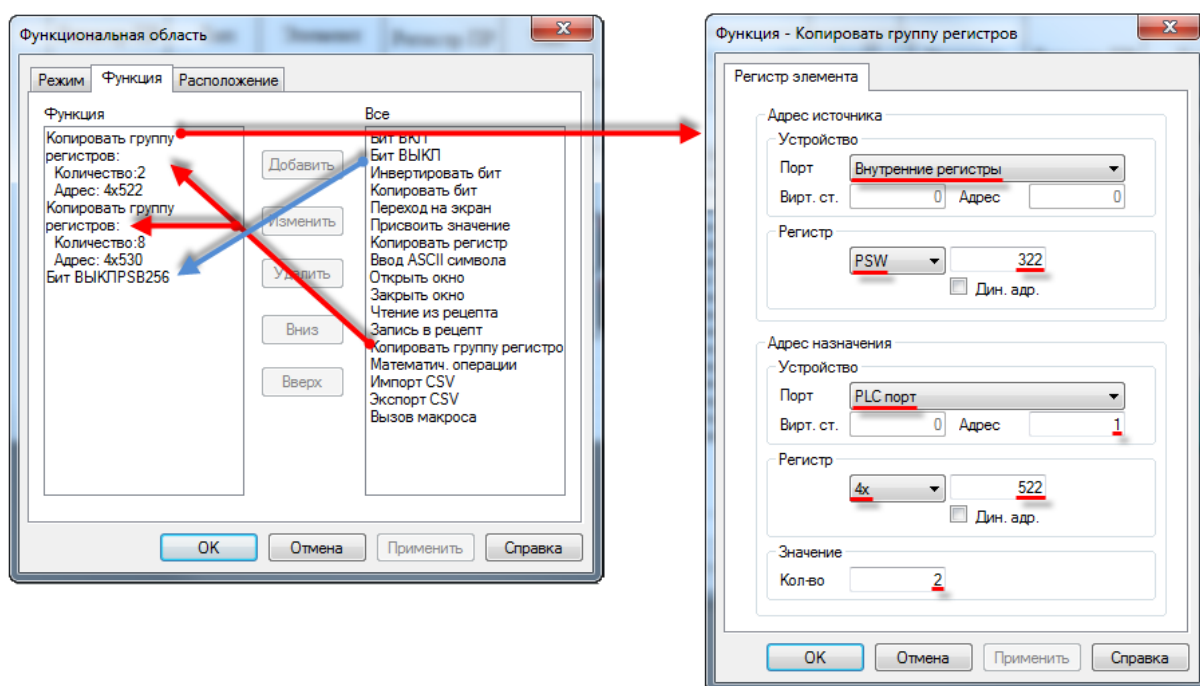


Рис. 3.20. Настройка элемента **Функциональная область** (чтение), вкладка **Функция**, действие 1

Для второго действия укажем адрес источника – **внутренняя память** панели, адрес первого записываемого регистра – **330**, адрес назначения – область памяти **4x** PR200, адрес первого регистра назначения – **530**, количество записываемых регистров – 8. Таким образом, данное действие будет копировать регистры панели **PSW330-337** в регистры PR200 **530-537**.

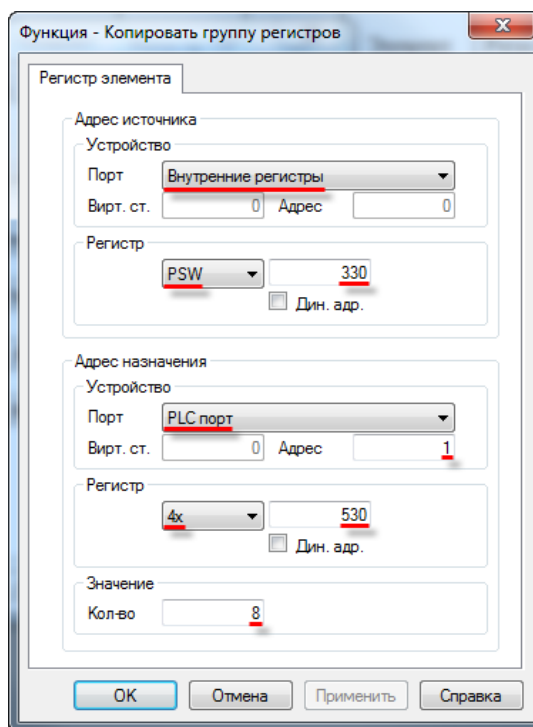


Рис. 3.21. Настройка элемента **Функциональная область** (запись), вкладка **Функция**, действие 2

Теперь осталось только настроить элементы, с помощью которых будут отображаться и изменяться значения регистров, связанных с **групповым опросом**.

К элементам **Индикатор** привяжем регистры **PSW320** и **PSW321** (для обоих – бит **00**).

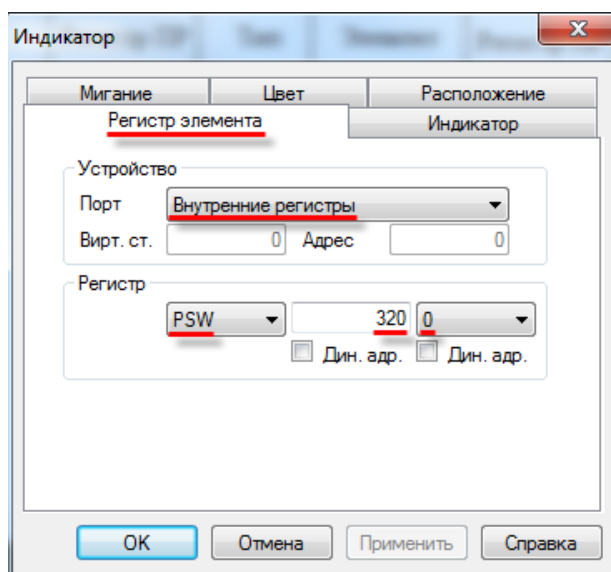


Рис. 3.22. Настройка элемента **Индикатор** (X\_S\_СП2)

К элементам **Цифровой ввод (INT)** привяжем регистры **PSW322** и **PSW323**. На вкладке **Ввод данных** поставим галочку **Бит уведомления** с привязанным битом **PSB256**. При вводе значения с помощью элемента этот бит примет значение **TRUE**, что приведет к запуску функциональной области (см. рис. 3.19), которая произведет запись в регистры ПР200 и вернет бит в состояние **FALSE** (т.е. запись будет произведена однократно).

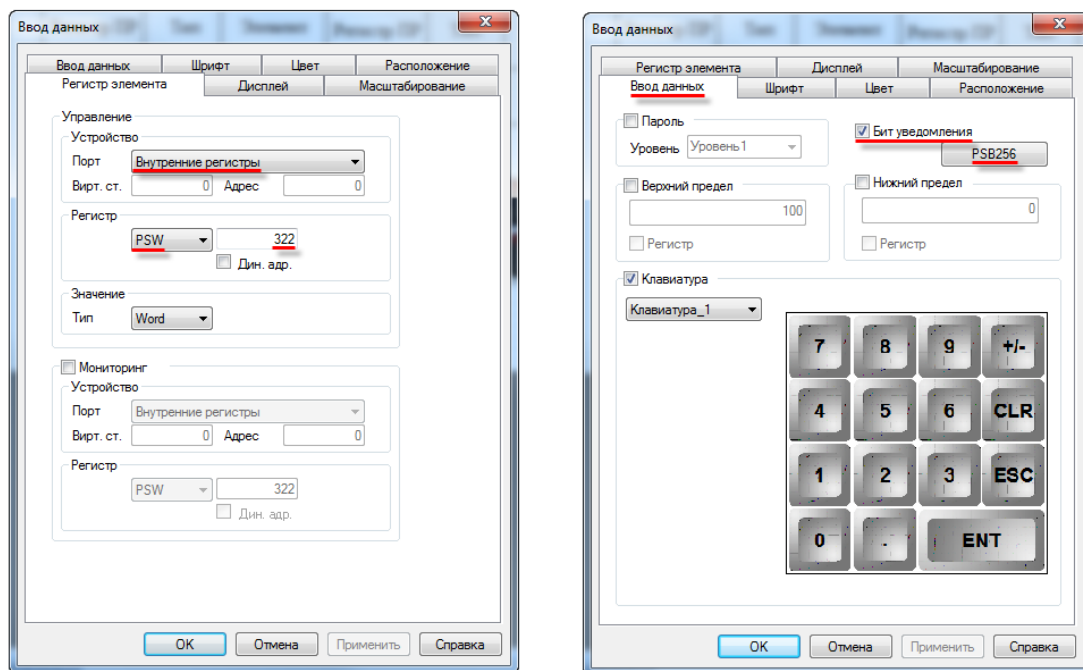


Рис. 3.23. Настройка элемента **Цифрой ввод (I\_S2)**

К элементам **Цифровой ввод (FLOAT)** привяжем регистры **PSW330**, **332**, **334** и **336** с типом **DWORD**. На вкладке **Дисплей** укажем формат **Float**. На вкладке **Ввод данных** поставим галочку **Бит уведомления** с привязанным битом **PSB256**. При вводе значения с помощью элемента этот бит примет значение **TRUE**, что приведет к запуску функциональной области (см. рис. 3.19), которая произведет запись в регистры ПР200 и вернет бит в состояние **FALSE** (т.е. запись будет произведена однократно).

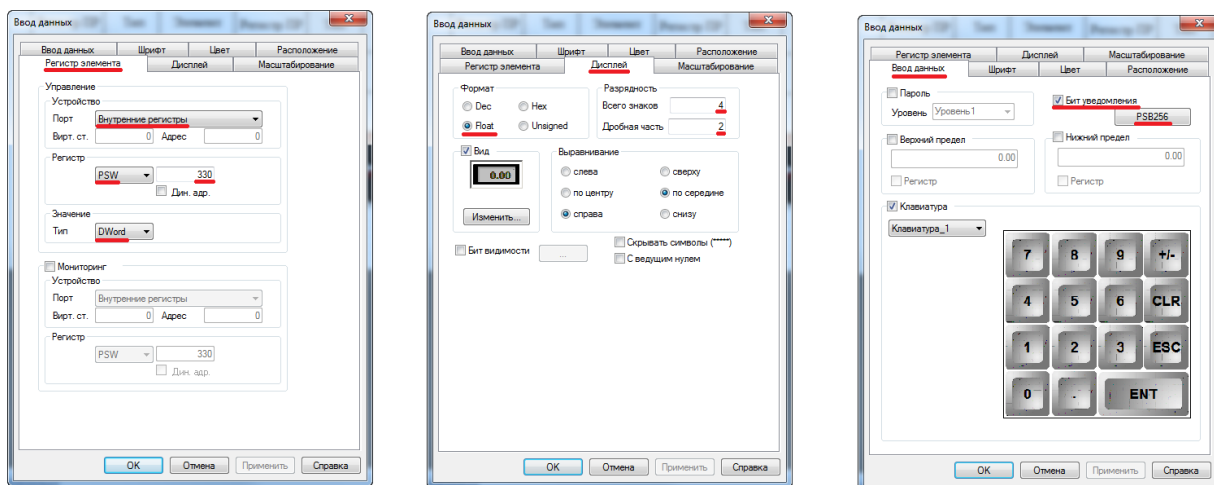


Рис. 3.24. Настройка элемента **Цифрой ввод (F\_S2)**

На этом создание проектов завершено. Описание работы с примером приведено в [п. 5](#).

## 4. СП310 – Modbus RTU Slave, ПР200 – Modbus RTU Master

### 4.1. Формулировка задачи

Настроим обмен между **ПР200** и **СП310** по протоколу **Modbus RTU**. ПР200 будет выполнять функцию **master**-устройства, СП310 – **slave**-устройства.

Основные характеристики используемых устройств приведены в табл. 4.1. Используемые в примере переменные и их адреса описаны в табл. 4.2.

Табл. 4.1. Характеристики используемых в примере устройств

Устройство	ПР200-24.2(4)	СП310
Функция	Master	Slave
Используемый порт	RS-485 (слот 1)	Download
Настройки обмена	115200, 8 бит, 1 стоп бит, без контроля четности	
Slave ID	-	1
Среда разработки проекта	Owen Logic 1.8.93.7745	Конфигуратор СП300 (V2.D3d-3)
Название файла проекта	PR200_ ModbusExample_OW1893.owl	SP310_ModbusExample.txp

Табл. 4.2. Список используемых в примере переменных

СП310	Тип данных	ПР200		Комментарий
Бит/Регистр элемента		Имя переменной	Функции Modbus (чтение/запись)	
PSB400	BOOL	X_M1	0x01 / 0x05	
PSB401	BOOL	X_M2	0x01 / 0x05	
PSB402	BOOL	X_M3	0x01 / 0x05	
PSB403	BOOL	X_M4	0x01 / 0x05	
PSW400	WORD	I_M1	0x03 / 0x06	
PSW401	WORD	I_M2	0x03 / 0x06	
PSW402	WORD	I_M3	0x03 / 0x06	
PFW400	WORD	I_M4	0x03 / 0x06	энергонезависимая
PSW410-411	FLOAT	F_M1	0x03 / 0x10	
PSW412-413	FLOAT	F_M2	0x03 / 0x10	
PSW414-415	FLOAT	F_M3	0x03 / 0x10	
PFW410-411	FLOAT	F_M4	0x03 / 0x10	энергонезависимая

## 4.2. Создание проекта для СП310

Для того чтобы использовать **СП310** в режиме **Modbus RTU Slave** необходимо в меню **Файл** выбрать пункт **Настройки проекта** и на вкладке **Устройство** настроить режимы работы последовательных портов. В нашем примере мы будем использовать **Download-порт**. Настроим его в качестве **Modbus RTU Slave** согласно [табл. 4.1](#): скорость – **115200**, бит данных – **8**, стоп бит – **1**, контроль четности – **нет**, адрес slave-устройства – **1**. Вспомним, что в [п. 3.2](#) мы выбрали такой же адрес при настройке ПР200 как slave-устройства – но поскольку для обмена с ПР200 панель использует другой порт (т.е. другую линию связи), то мы можем использовать и этот адрес.

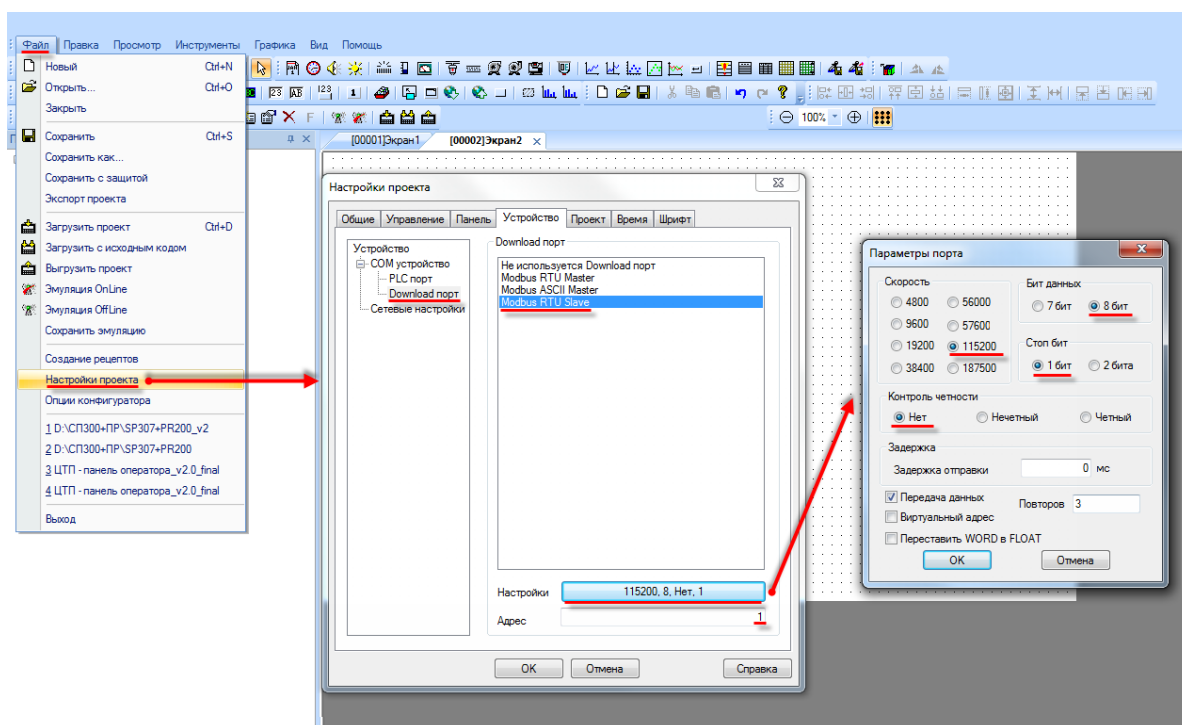


Рис. 4.1. Настройка **Download-порта** (режим **Modbus RTU Slave**)

Добавим в проект, созданный в [п. 3.3](#) еще один экран визуализации.

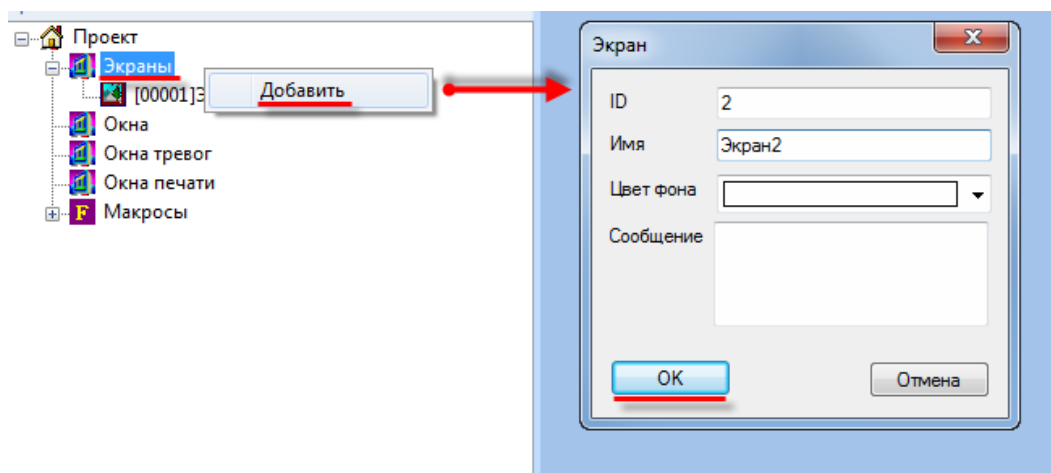


Рис. 4.2 Добавление в проект экрана визуализации

Добавим на экран панели четыре элемента **Переключатель с индикацией**, восемь элементов **Цифровой ввод** и два элемента **Переход на экран**. Переключатели и элементы ввода будут использоваться для отображения значений, полученных от master-устройства, и их изменения. Настройки элементов приведены ниже. Пояснительные надписи сделаны с помощью элементов **Статический текст**.

ПП200 - Modbus RTU Master (слот 1)				СП307 - Modbus RTU Slave (порт Download) адрес 1				
Бит панели	Тип	Элемент	Регистр панели	Тип	Элемент	Регистр панели	Тип	Элемент
400 (X_M1)	PSB	PSB400	400 (I_M1)	PSW WORD	PSW400	410-411 (F_M1)	PSW DWORD	PSW410
401 (X_M2)	PSB	PSB401	401 (I_M2)	PSW WORD	PSW401	412-413 (F_M2)	PSW DWORD	PSW412
402 (X_M3)	PSB	PSB402	402 (I_M3)	PSW WORD	PSW402	414-415 (F_M3)	PSW DWORD	PSW414
403 (X_M4)	PSB	PSB403	400 (I_M4)	PFW WORD	PFW400	410-411 (F_M4)	PFW DWORD	PFW410

Функциональная область

Функциональная область

СП - мастер

СП - слэйв

Рис. 4.3 Внешний вид экрана панели

Поскольку в нашем проекте (созданном в п. 3.3) появился еще один экран визуализации, то необходимо предусмотреть возможность переходов между ними. Для этого будет использовать элемент **Переход на экран**. В настройках кнопки перехода **СП-мастер** на вкладке **Действие** укажем экран **1**, режим **Без ввода пароля**. Таким образом, нажатие на эту кнопку будет возвращать нас на первый экран проекта. Кнопка **СП –слэйв** может не настраиваться – она служит только для отображения названия текущего экрана, выделенного заданным цветом.

Переход на экран

Действие

Кнопка

Цвет

Расположение

ID экрана

1

Режим

Без ввода пароля

С вводом пароля

Клавиатура

Клавиатура\_60

OK

Отмена

Применить

Справка

Рис. 4.4. Настройка элемента **Переход на экран**



Соответственно, на первом экране проекта потребуется добавить кнопку перехода на второй экран с **ID=2**.

Вспомним также о функциональных областях, расположенных на первом экране. При переходе на второй экран они приостановят свою работу. Если необходимо, чтобы работа функциональной области происходила в независимости от открытого экрана, необходимо нажать на нее **ПКМ** и в контекстном меню элемента выбрать пункт **Сделать глобальным**. После этого в редакторе область будет отображаться на всех экранах проекта.

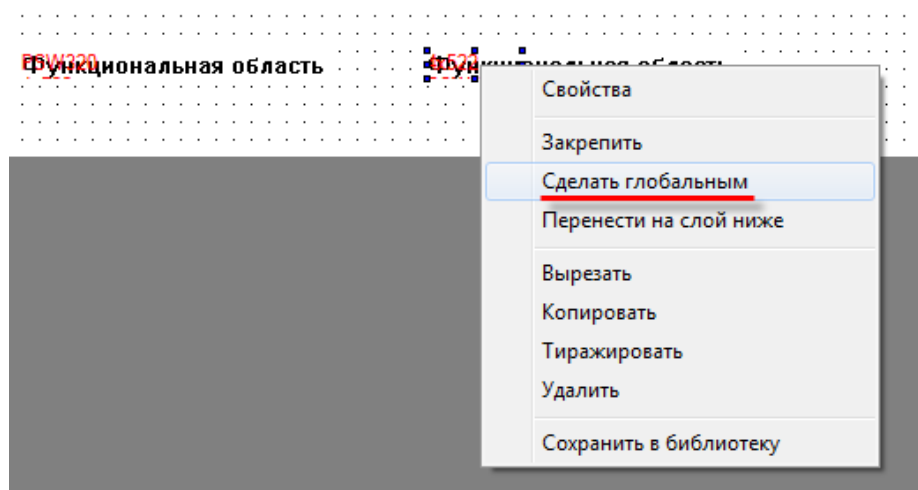


Рис. 4.5. Создание глобальной функциональной области

Настроим элементы второго экрана панели (см. рис. 4.3). К элементам **Переключатели с индикацией** привяжем биты панели **PSB400, 401, 402 и 403**:

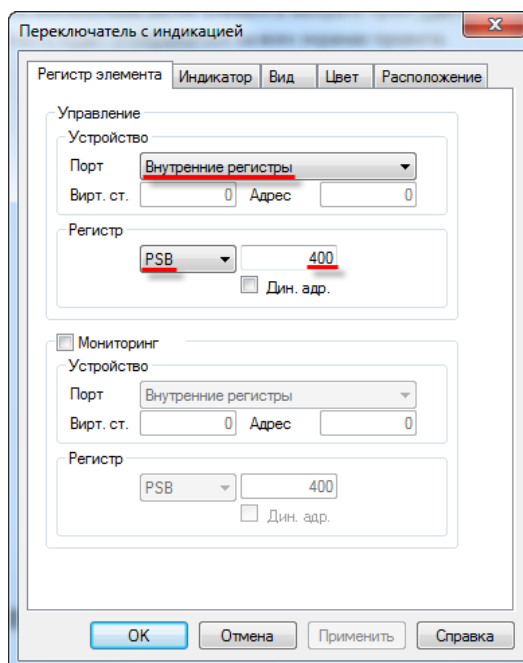


Рис. 4.6. Настройки элемента **Переключатель с индикацией (PSB400)**. Остальные элементы настраиваются по аналогии

К элементам **Цифровой ввод (WORD)** привяжем регистры панели **PSW400, 401, 402** и **PFW400**. PFW400 является **энергонезависимым** регистром, который сохраняет свое значение после перезагрузки панели. **Обратите внимание**, что области памяти **PSB, PSW** и **PFW** являются независимыми – поэтому при работе с ними можно использовать совпадающие адреса.

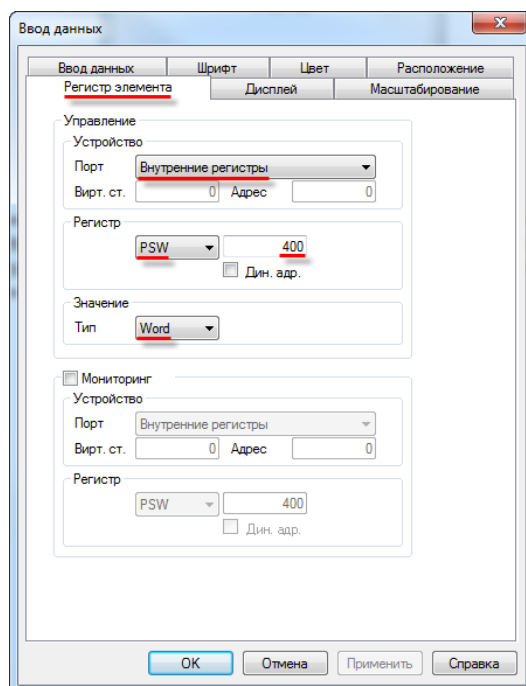


Рис. 4.7. Настройки элементов **Цифрой ввод (WORD)**

К элементам **Цифровой ввод (DWORD)** привяжем регистры **PSW410, PSW412, PSW414** и **PFW410**. Укажем для них тип **DWORD**. На вкладке **Дисплей** выберем тип **Float** и настроим количество знаков после запятой.

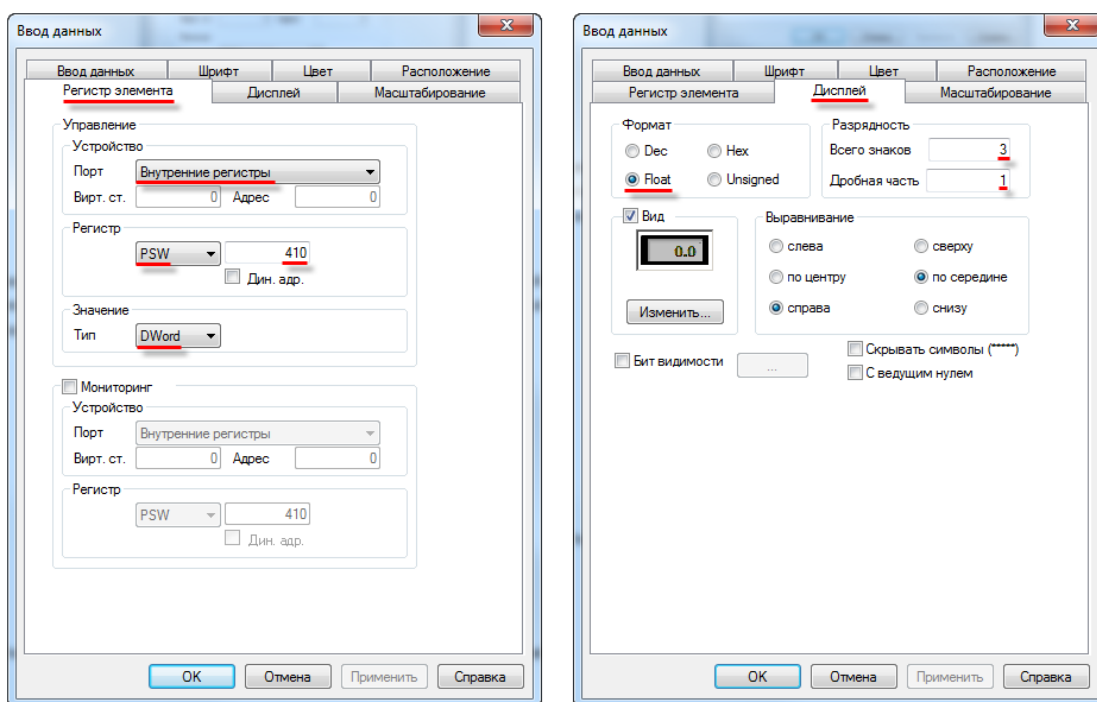


Рис. 4.8. Настройки элементов **Цифрой ввод (DWORD)**

Таким образом, мы сформировали карту регистров панели. **Отметим**, что регистры **PSW** и **PFW** являются **holding**-регистрами, биты **PSB** – ячейками (**coils**).

Осталось настроить ПР200 в режиме Modbus RTU Master, реализовав обращение к соответствующим битам/регистрам панели.

### 4.3. Создание проекта для ПР200

Для того чтобы использовать **ПР200** в режиме **Modbus RTU Master**, необходимо в меню **Прибор** выбрать пункт **Настройки прибора** и на вкладке **Интерфейсы** добавить в проект новый интерфейс RS-485. Выберем для него **Слот 1**, режим **Master** и установим сетевые настройки согласно [табл. 4.1](#): скорость – **115200**, бит данных – **8**, стоп бит – **1**, контроль четности – **нет**.

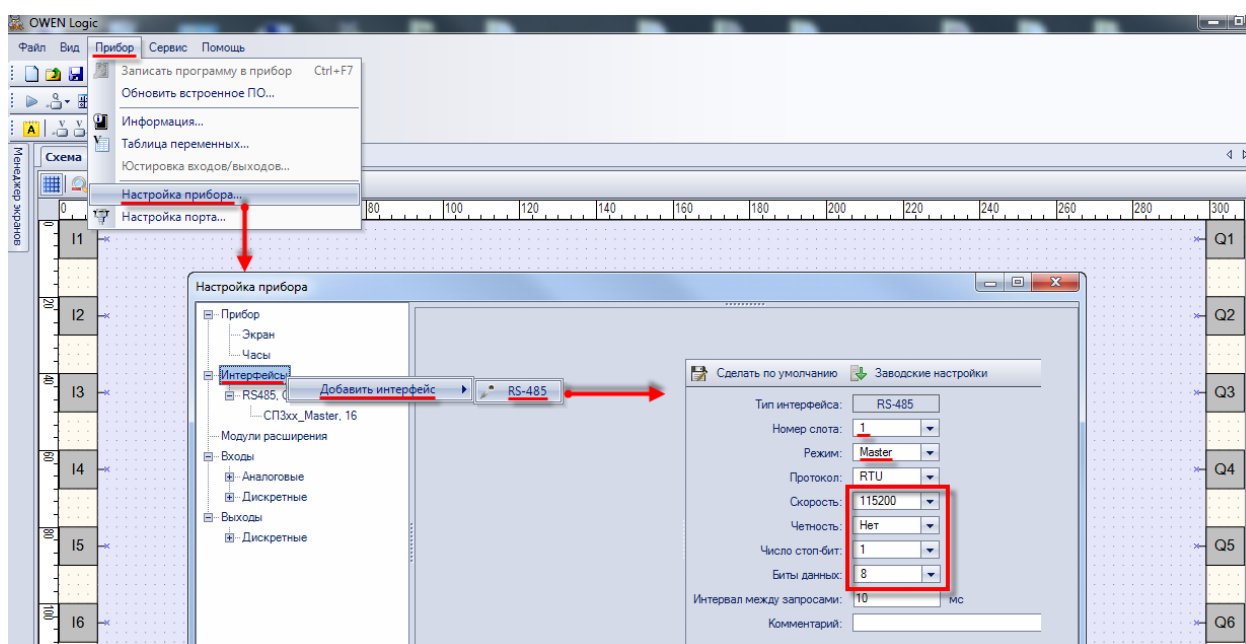


Рис. 4.9. Настройка **Слота 1 RS-485** (режим **Modbus RTU Slave**)

В настройках слота укажем адрес панели – **1**. Вспомним, что в [п. 3.2](#) мы выбрали такой же адрес при настройке ПР200 как slave-устройства – но поскольку для обмена с ПР200 панель использует другой порт (т.е. другую линию связи), то мы можем использовать и этот адрес.

Добавим сетевые переменные, указав для них имена, типы, адреса и функции Modbus в соответствии с [табл. 4.2](#). Для каждой переменной поставим галочку **Запись по изменению**.

**Необходимо обратить внимание** на следующие моменты:

1. в панели СП310 номера бит задаются в явном виде: **PSB400, PSB401** и т.д. При настройке их опроса в среде Owen Logic необходимо указать номер регистра и номер бита. Для того чтобы определить эти значения, необходимо разделить битовый адрес из панели на 16 (т.е. на число бит в одном регистре). Частное будет являться номером регистра, остаток от деления – номером бита. Поясним вышесказанное на практическом примере для PSB400:

$$\text{Номер регистра в OL (PSB400)} = \text{Частное} \left( \frac{400}{16} \right) = 25$$

$$\text{Номер бита в OL (PSB400)} = \text{Остаток от деления} \left( \frac{400}{16} \right) = 0$$

Приведем пример для PSB444 (уточним, что в примере этот бит не используется):

$$\text{Номер регистра в OL (PSB444)} = \text{Частное} \left( \frac{444}{16} \right) = 27$$

$$\text{Номер бита в OL (PSB444)} = \text{Остаток от деления} \left( \frac{444}{16} \right) = 12$$

2. для **энергозависимых** регистров панели **PFW** задается смещение **10000** (PFW400 – регистр 10400 и т.д.).

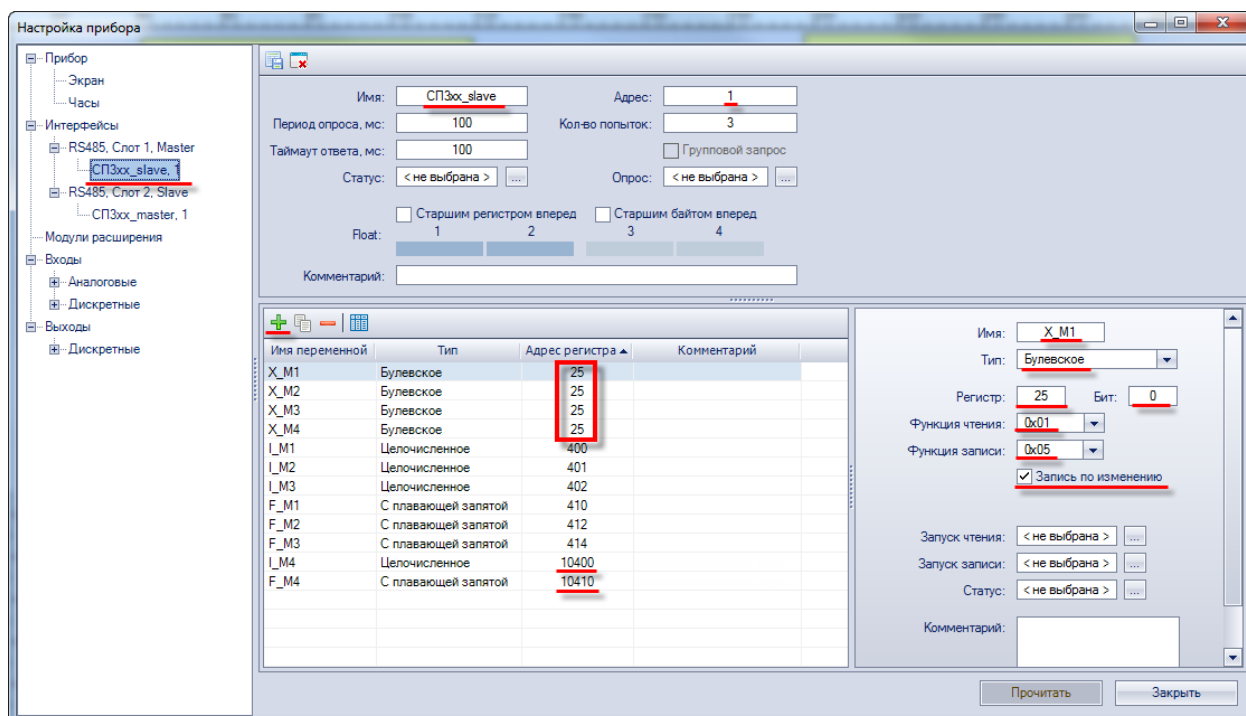


Рис. 4.10. Настройка опроса slave-устройства в Owen Logic

Добавим созданные сетевые переменные на холст и скопируем их значения во внутренние переменные:

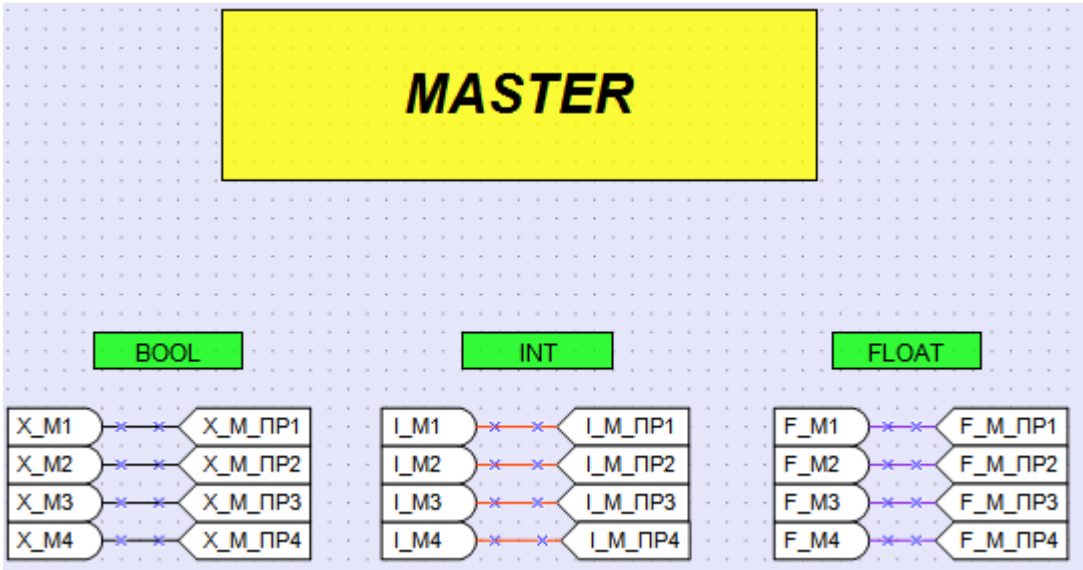


Рис. 4.11. Размещение переменных на холсте

Выберите переменную или создайте новую

Имя переменной	Тип переменной	Энергонезависимость	Значение по умолчанию	Использование в проекте	Комментарий
< не выбрана >	Булевское	<input type="checkbox"/>	0	Нет	
F_M_ПР1	С плавающей за...	<input type="checkbox"/>	0	Да	
F_M_ПР2	С плавающей за...	<input type="checkbox"/>	0	Да	
F_M_ПР3	С плавающей за...	<input type="checkbox"/>	0	Да	
F_M_ПР4	С плавающей за...	<input type="checkbox"/>	0	Да	
I_M_ПР1	Целочисленное	<input type="checkbox"/>	0	Да	
I_M_ПР2	Целочисленное	<input type="checkbox"/>	0	Да	
I_M_ПР3	Целочисленное	<input type="checkbox"/>	0	Да	
I_M_ПР4	Целочисленное	<input type="checkbox"/>	0	Да	
X_S_ПР1	Булевское	<input type="checkbox"/>	0	Да	
X_S_ПР2	Булевское	<input type="checkbox"/>	0	Да	
X_S_ПР3	Булевское	<input type="checkbox"/>	0	Да	
X_M_ПР1	Булевское	<input type="checkbox"/>	0	Да	
X_M_ПР2	Булевское	<input type="checkbox"/>	0	Да	
X_M_ПР3	Булевское	<input type="checkbox"/>	0	Да	
X_M_ПР4	Булевское	<input type="checkbox"/>	0	Да	

Рис. 4.12. Таблица внутренних переменных. Суффиксом **М** помечены переменные, в которые копируются значения, считанные из панели

Итак, мы разместили на холсте все необходимые элементы. **Отметим**, что программа не содержит никакого алгоритма – только отображение и изменение значений, которые будут считываться и записываться в панель оператора **СП310**. Эти значения пользователь может использовать при реализации алгоритма, выполняющего его задачу.

Создадим экран визуализации, на котором будем отображать и изменять упомянутые значения. Откроем **Менеджер экранов** и добавим новый экран с названием **MASTER**.

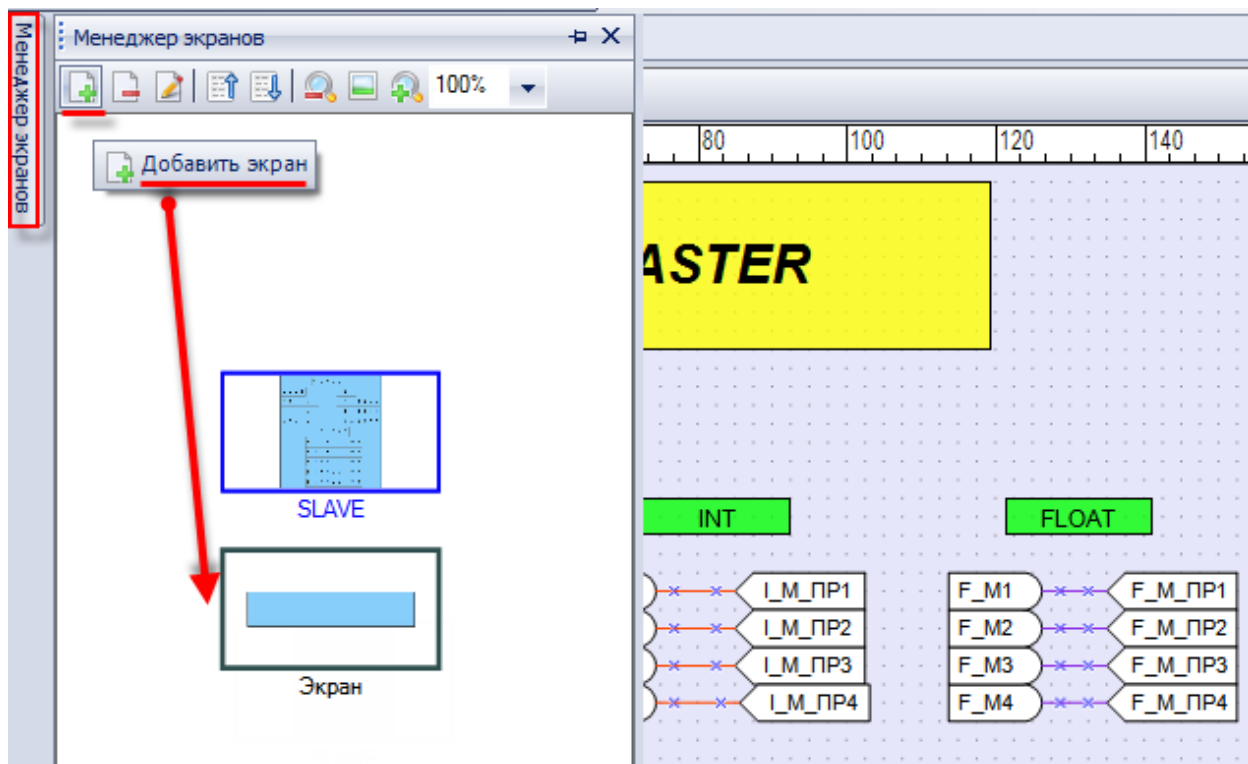


Рис. 4.13. Добавление нового экрана

Поскольку в проекте теперь два экрана, необходимо настроить переходы между ними. Выделим экран **MASTER**, в его параметрах нажмем на вкладку **Переход к экрану** и настроим переход на экран **SLAVE** при нажатии сочетания клавиш **Alt+Вверх**.

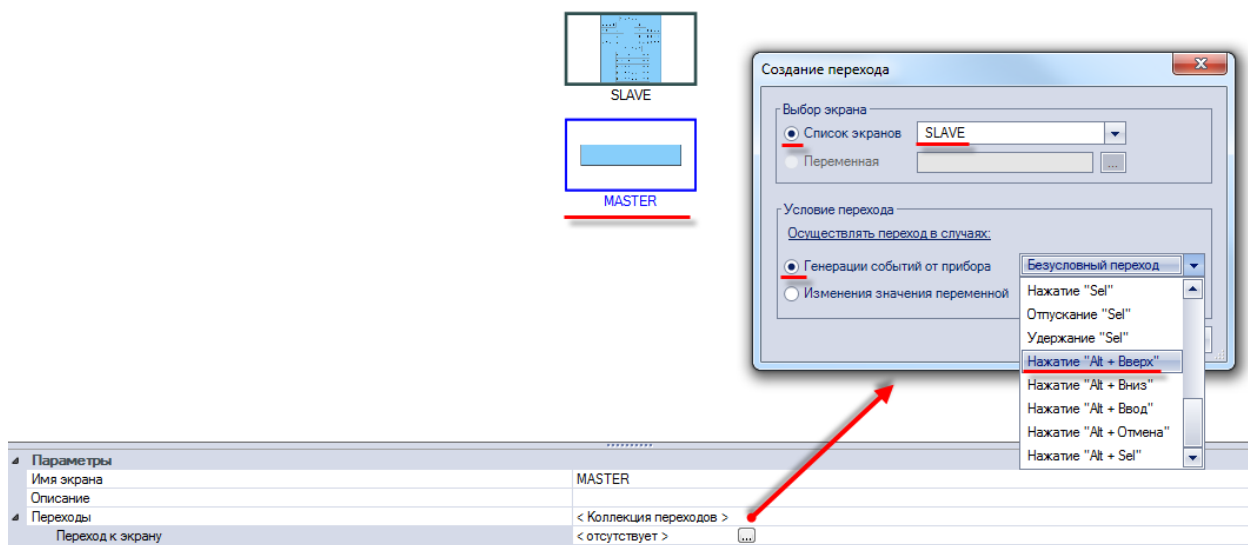


Рис. 4.14. Настройка перехода на экран **SLAVE** с экрана **MASTER**

Для экрана **SLAVE** аналогичным образом настроим переход на экран **MASTER** при нажатии сочетания клавиш **Alt+Вниз**.

Теперь можно перейти к наполнению экрана **MASTER** элементами. Увеличим размер экрана до **12-ти** строк. Экран будет содержать 4 элемента **Ввод/вывод (Boolean)**, 8 элементов **Ввод/вывод (int/float)**, а также 4 **Метки** с поясняющими надписями. Настройки элементов приведены ниже.

The screenshot shows a graphical user interface for a system named "MASTER". The interface is organized into several sections:

- MASTER**: The main title at the top center.
- BOOL**: A section for Boolean inputs/outputs, containing four elements labeled **X 1 = Н е т**, **X 2 = Н е т**, **X 3 = Н е т**, and **X 4 = Н е т**.
- I n t**: A section for integer inputs/outputs, containing four elements labeled **I 1 = 0 0 0 0**, **I 2 = 0 0 0 0**, **I 3 = 0 0 0 0**, and **I 4 = 0 0 0 0**.
- F L O A T**: A section for floating-point inputs/outputs, containing four elements labeled **F 1 = + 0 0 , 0**, **F 2 = + 0 0 , 0**, **F 3 = + 0 0 , 0**, and **F 4 = + 0 0 , 0**.

The interface is displayed on a light blue grid background. On the right side, there is a vertical toolbar with icons for various functions like saving, deleting, and zooming.

Рис. 4.15. Внешний вид экрана **MASTER**

К элементу **X1** привяжем сетевую переменную **X\_M1** и зададим тексты для ее значений **TRUE/FALSE**. Для параметра **Редактируемо** поставим значение **Да**.

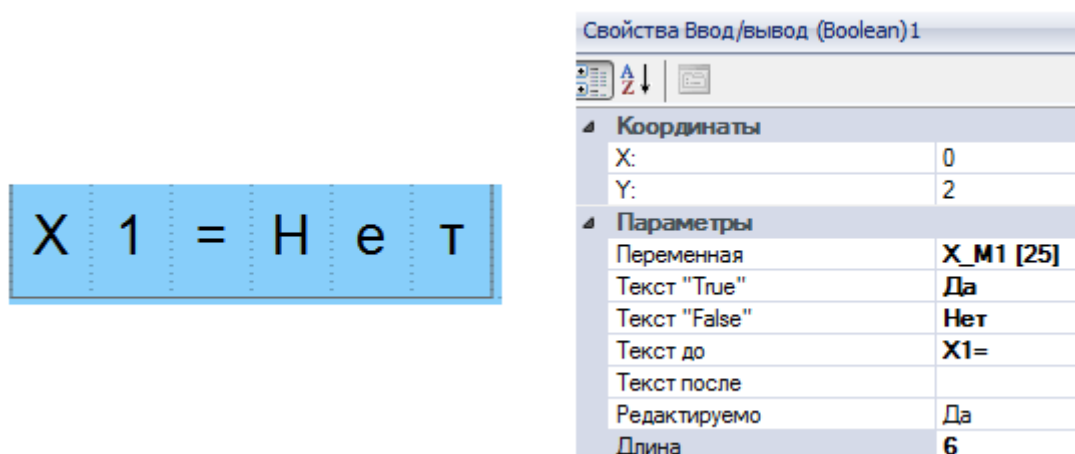


Рис. 4.16. Настройка элемента **Ввод/вывод (Boolean) X1**

К элементам **X2** и **X3** привяжем переменные **X\_M2** и **X\_M3**, настроив их по аналогии с элементом **X1**.

К элементу **I1** привяжем сетевую переменную **I\_S1** и зададим ей тип **Целый**. Для параметра **Редактируемо** поставим значение **Да**.

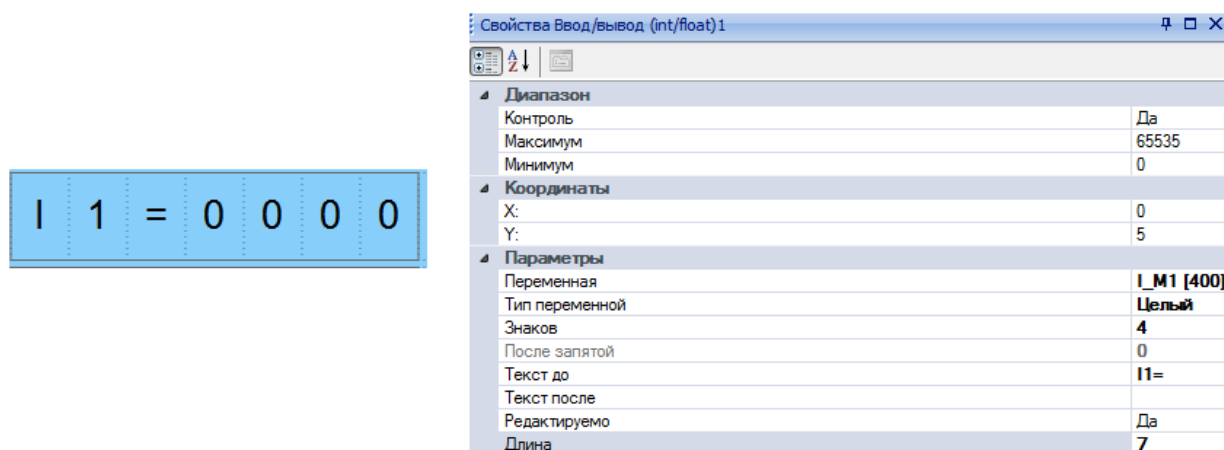


Рис. 4.17. Настройка элемента **Ввод/вывод (int/float) I1**

К элементам **I2** и **I3** привяжем переменные **I\_S2** и **I\_S3**, настроив их по аналогии с элементом **I1**.



К элементу **F1** привяжем сетевую переменную **F\_S1** и зададим ей тип **Вещественный**. Для параметра **Редактируемо** поставим значение **Да**.

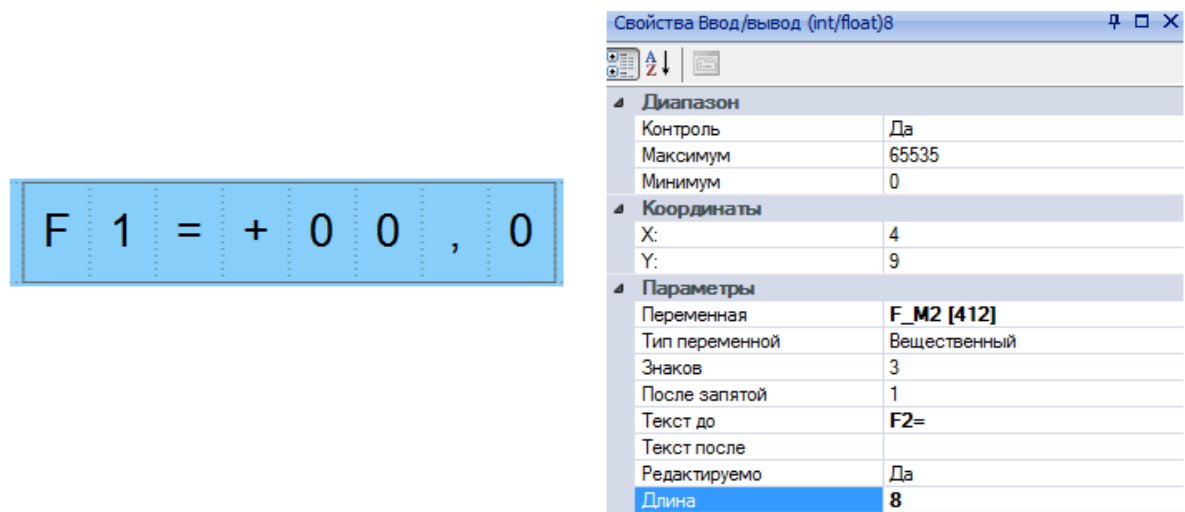


Рис. 4.18. Настройка элемента **Ввод/вывод (int/float) F1**

К элементам **F2**, **F3**, **F4** и **F5** привяжем переменные **F\_S2**, **F\_S3**, **F\_S4** и **F\_S5**, настроив их по аналогии с элементом **F1**.

На этом создание проекта для ПР200 завершено.

## 5. Работа с примером

Соединим порты **ПР200-24.2(4)** и **СП310** между собой. **Слот 1** ПР200 будет соединен с **Download-портом СП310**, а **слот 2** – с PLC-портом. Распиновка кабеля RS-485 для панели приведена на рис. 5.1 (у обоих портов панели идентичная распиновка).

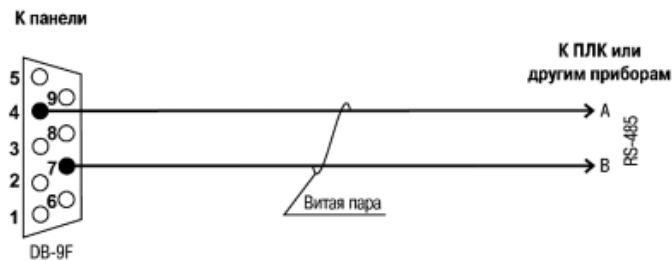


Рис. 5.1. Распиновка кабеля RS-485 для панели оператора СП3хх

Загрузим проекты в оба устройства.

По умолчанию на **СП310** отображается экран **Master**, на **ПР200** – экран **Slave**. Изменяйте значения на дисплее панели и наблюдайте соответствующие изменения на дисплее ПР200. Изменяйте значения на дисплее ПР200 и наблюдайте соответствующие изменения на дисплее панели. Выбор изменяемой переменной осуществляется с помощью кнопки **SEL** (из числа элементов, отображаемых в данный момент на дисплее), изменение значения – с помощью стрелок, переключение изменяемых разрядов – с помощью комбинации **Alt+стрелки вверх/вниз**. Выход из режима ввода значений осуществляется с помощью кнопки **OK**, после чего можно осуществлять прокрутку экрана с помощью стрелок.

Переключите экраны на **Slave** (для **СП310**) и **Master** (для **ПР200**). Переключение экранов панели осуществляется с помощью кнопок внизу экрана, на ПР200 – с помощью комбинации **ALT+стрелки вверх/вниз**. Изменяйте значения на дисплее ПР200 и наблюдайте соответствующие изменения на дисплее панели.



Рис. 5.2. Работа с примером. Ввод значений переменных **X1** и **I1**