



Негосударственное частное образовательное
учреждение высшего образования
«Технический университет УГМК»

**ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО
ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ ПО
МОДУЛЮ 7. ЦИФРОВИЗАЦИЯ И АВТОМАТИЗАЦИЯ
ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ**

Направление подготовки	<i>22.04.02 Metallurgy</i>
Направленность (профиль)	<i>Обогащение и подготовка сырья к металлургической переработке</i>
Уровень высшего образования	<i>магистратура</i> <i>(бакалавриат, специалитет, магистратура)</i>
Квалификация выпускника	<i>магистр</i>

Авторы-разработчики: Худяков П.Ю., канд. физ.-мат. наук, Мамонов С.В., канд. техн. наук,
Дементьев А.В.

Рассмотрено на заседании кафедры обогащения полезных ископаемых.

Одобрено Методическим советом университета 18 октября 2021 г., протокол № 6

г. Верхняя Пышма
2021

Задания и методические указания для студентов по выполнению практических работ по модулю 7 Цифровизация и автоматизация технологических процессов.

Практические занятия являются формой аудиторных занятий. Практические работы по модулю имеют целью под руководством преподавателя на практике закрепить и углубление изученного материала и приобретение умений и навыков.

Студентам для лучшего усвоения материала рекомендуется вести запись информации, полученной во время обсуждения вопросов на практических занятиях.

Практические работы по теме модуля: «Основы АСУТП обогатительных фабрик»

Практическая работа № 1. Тема: Системы автоматизированного проектирования

1. Время на выполнение задания – ОФО - 4 часа, ЗФО – 1 час.

2. Удовлетворительным результатом успешного выполнения практической работы считается правильная функциональная схема автоматизации.

3. Оценка теоретических знаний:

Устные вопросы по теме практического задания:

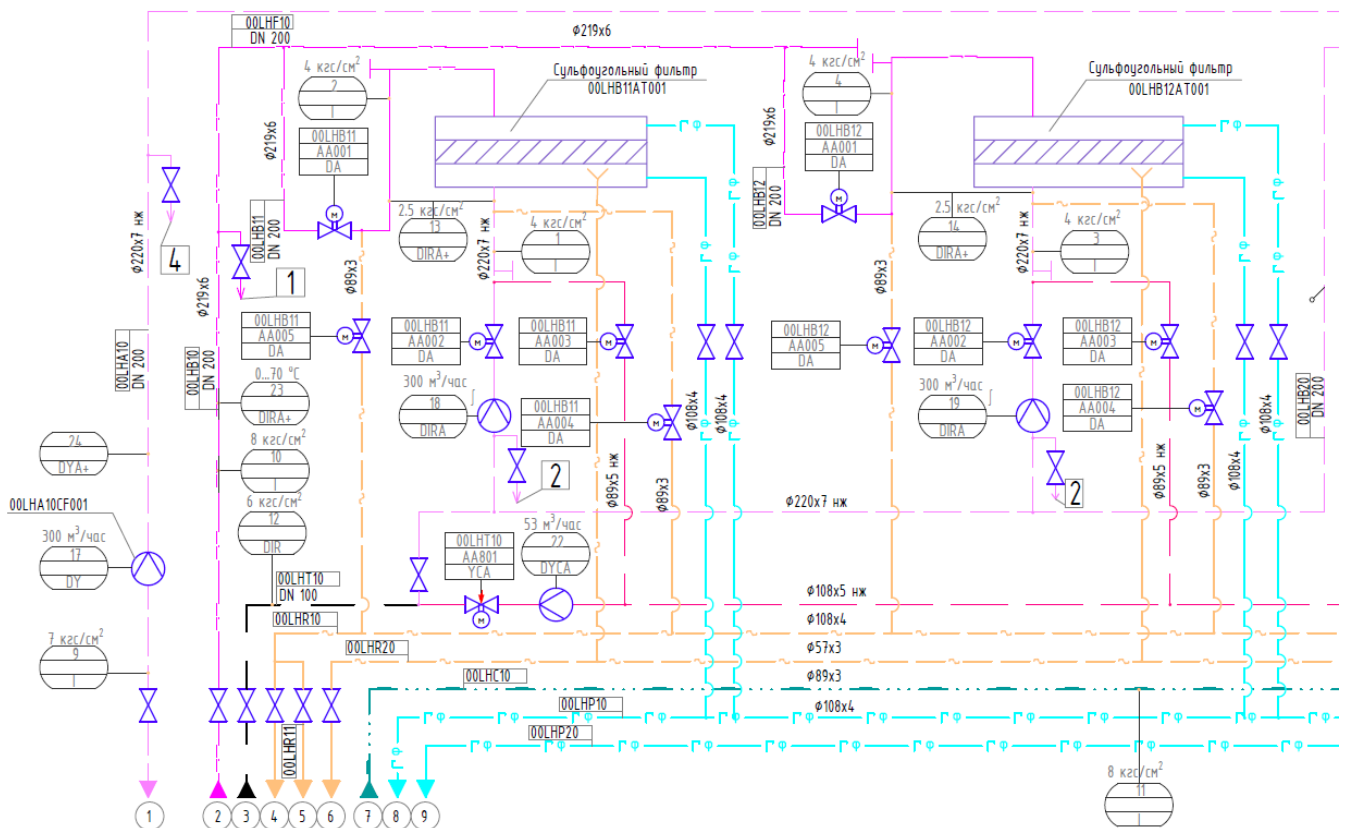
Какие системы кодирования элементов АСУ вы знаете.

Перечислить основные разделы кодов KKS.

Как формируется условное графическое обозначение точки измерения.

Практическое задание (на основе выданной технологической схемы):

- оценить полноту исходных данных;
- выделить основные технологические процессы;
- проанализировать участки схемы;
- выполнить кодирование элементов схемы.



Пример функциональной схемы автоматизации

Результатом успешного выполнения практического задания считается умение обучающегося выполнить правильное кодирование элементов функциональной схемы автоматизации в САПР.

Практическая работа № 2. Тема: Разработка схем электрических принципиальных

1. Время на выполнение задания – ОФО - 4 часа, ЗФО – 1 час.

2. Удовлетворительным результатом успешного выполнения практической работы считается правильная схема электрическая принципиальная.

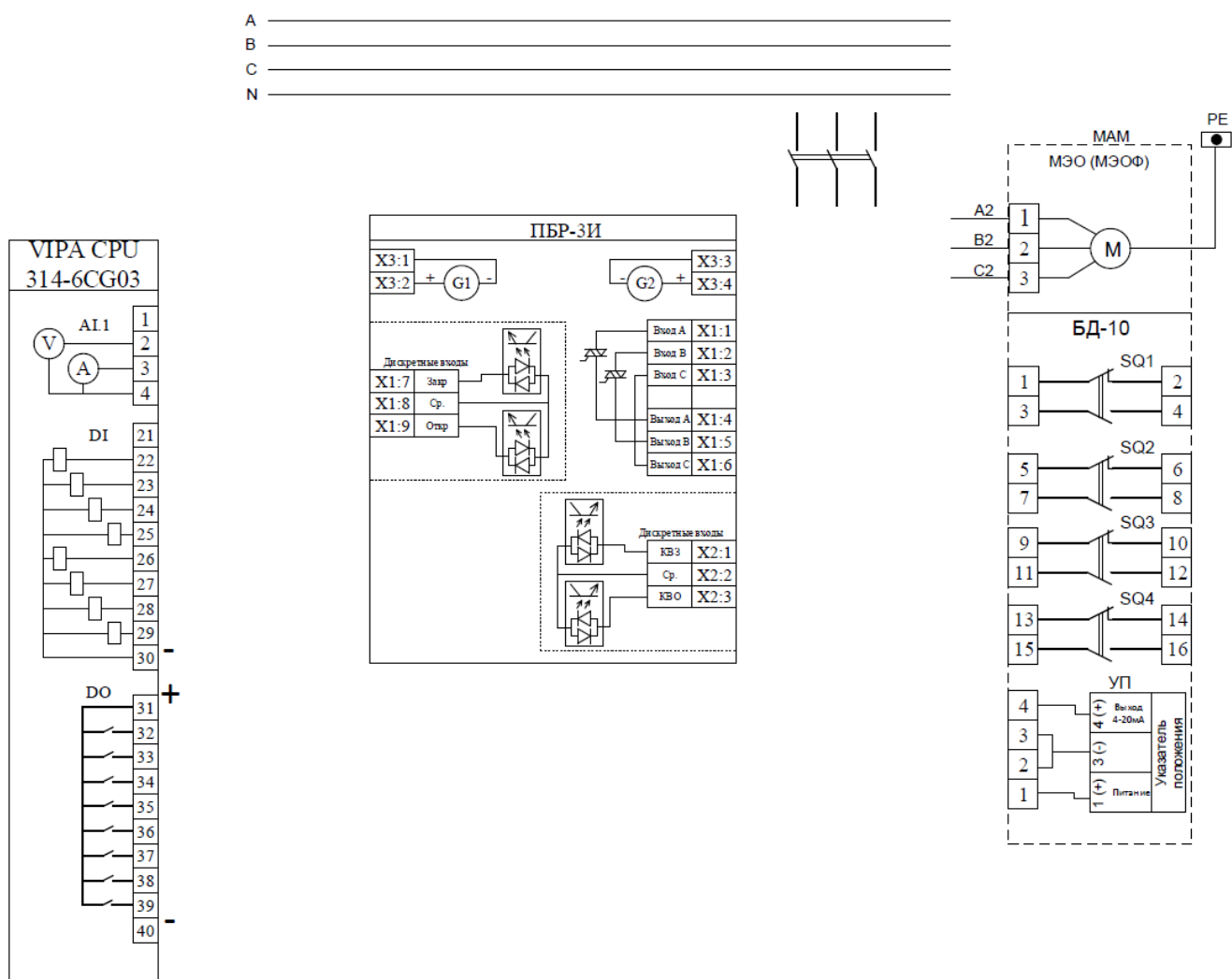
Устные вопросы по теме практического задания:

Что такое схема электрическая принципиальная

Правила разработки схем электрических принципиальных

Практическое задание (на основе выданной технологической схемы):

- оценить полноту исходных данных;
- выделить основные узлы и элементы
- проанализировать участки схемы;
- выполнить соединение элементов схемы в САПР.



Практическая работа № 3. Тема: Расчет коэффициентов калибровочных характеристик

1. Время на выполнение задания – ОФО - 2 часа, ЗФО – 2 часа.

2. Удовлетворительным результатом успешного выполнения практической работы считается правильный расчет коэффициентов.

Устные вопросы по теме практического задания:

Преобразование сигналов датчиков в физические и инженерные величины
Типы сигналов датчиков

Практическое задание (на основе выданной технологической схемы):

- оценить полноту исходных данных;
- выбрать тип характеристики
- выполнить расчет коэффициентов.

Вариант	Тип измеряемого параметра и его пределы
1	Температура, датчик для измерения – термопара ХК, среда – пар, температура 530 - 550°C, давление 21 МПа, регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
2	Давление, датчик для измерения – Метран 150, среда – пар, температура 545°C, давление 15-20 МПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
3	Расход, датчик для измерения – Метран, среда – пар, температура 545°C, расход 5-15 т/ч, давление 21 МПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
4	Уровень, датчик для измерения – Метран 150, среда – вода, давление 21 МПа, уровень 5-10 м, температура 545°C, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)
5	Температура, датчик для измерения – термопара ХА, среда – вода, температура 300-400 °С, давление 600 кПа, регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
6	Давление, датчик для измерения – Метран 150, среда – вода, температура 400 °С, давление 300-600 кПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
7	Расход, датчик для измерения – Метран-350SFA, среда – вода, температура 400°C, расход 2-10 т/ч, давление 600 кПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
8	Уровень, датчик для измерения – Метран 150, среда – вода, давление 600 кПа, уровень 1-2 м, температура 400 °С, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)
9	Температура, датчик для измерения – термопара ХК, среда – воздух, температура 100-200 °С, давление 300 кПа, регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
10	Давление, датчик для измерения – Метран 150, среда – воздух, температура 200 °С, давление 100 - 300 кПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
11	Расход, датчик для измерения – Метран-350SFA, среда – воздух, температура 200 °С, расход 100-200 кг/ч, давление 300 кПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
12	Уровень, датчик для измерения – Метран 150, среда – вода, давление 300 кПа, уровень 5 - 7 м температура 200 °С, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)

Вариант	Тип измеряемого параметра и его пределы
13	Температура, датчик для измерения – термопара ХА, среда – пар, температура 400-500 °С, давление 100 кПа, регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
14	Давление, датчик для измерения – Метран 150, среда – пар, температура 500 °С, давление 50 - 100 кПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
15	Расход, датчик для измерения – Метран-350SFA, среда – вода, температура 500 °С, расход 50-100 кг/ч, давление 100 кПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
16	Уровень, датчик для измерения – Метран 150, среда – вода, давление 100 кПа, уровень 10 - 12 м температура 500 °С, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)
17	Температура, датчик для измерения – термопара ХК, среда – вода, температура 150 - 250 °С, давление 10 МПа, регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
18	Давление, датчик для измерения – Метран 150, среда – вода, температура 250 °С, давление 8 - 10 МПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
19	Расход, датчик для измерения – Метран-350SFA, среда – вода, температура 250 °С, расход 50-100 т/ч, давление 10 МПа, регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
20	Уровень, датчик для измерения – Метран 150, среда – вода, давление 10 МПа, уровень 25 - 30 м температура 250 °С, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)
21	Температура, датчик для измерения – термопара ХА, среда – воздух, температура 430 - 500 °С, давление - 50 кПа регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
22	Давление, датчик для измерения – Метран 150, среда – воздух, температура 500 °С, давление - 50... 0 кПа регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
23	Расход, датчик для измерения – Метран-350SFA, среда – воздух, температура 500 °С, расход 17-26 кг/ч давление 50 кПа регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
24	Уровень, датчик для измерения – Метран 150, среда – вода, давление - 50 кПа уровень 3-6 м температура 500 °С, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)
25	Температура, датчик для измерения – термопара ХК, среда – пар, температура 100 - 150 °С, давление 100 кПа, регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
26	Давление, датчик для измерения – Метран 150, среда – пар, температура 150 °С, давление 100 - 150 кПа регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
27	Расход, датчик для измерения – Метран-350SFA, среда – вода, температура 150 °С, расход 13 -27 т/ч давление 150 кПа регулирование давления производится за счет

Вариант	Тип измеряемого параметра и его пределы
	изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
28	Уровень, датчик для измерения – Метран 150, среда – вода, давление 150 кПа уровень 0,5 - 2 м температура 150 °С, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)
29	Температура, датчик для измерения – термопара ХА, среда – вода, температура 50 - 100 °С, давление 0 кПа регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
30	Давление, датчик для измерения – Метран 150, среда – вода, температура 90 °С, давление 0 - 20 кПа регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
31	Расход, датчик для измерения – Метран-350SFA, среда – вода, температура 90 °С, расход 92- 134 т/ч давление 0 кПа регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
32	Уровень, датчик для измерения – Метран 150, среда – вода, давление 0 кПа уровень 3 - 9 м температура 90 °С, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)
33	Температура, датчик для измерения – термопара ХК, среда – воздух, температура 130 - 250 °С, давление 200 кПа регулирование температуры производится впрыском воды в коллектор (управление электродвигателем регулирующего клапана)
34	Давление, датчик для измерения – Метран 150, среда – воздух, температура 250 °С, давление 120 - 200 кПа регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
35	Расход, датчик для измерения – Метран-350SFA, среда – воздух, температура 250 °С, расход 44 - 58 кг/ч давление 200 кПа регулирование давления производится за счет изменения проходного сечения канала (управление электродвигателем регулирующего клапана)
36	Уровень, датчик для измерения – Метран 150, среда – вода, давление 200 кПа уровень 4 - 7 м температура 250 °С, регулирование уровня производится за счет изменения расхода воды в бак (управление электродвигателем регулирующего клапана)

Практическая работа № 4. Тема: Разработка алгоритмов управления технологическим оборудованием

1. Время на выполнение задания – ОФО - 6 часов, ЗФО – 4 часа.
2. Удовлетворительным результатом успешного выполнения практической работы считается алгоритм, выполняющий требуемые функции.

3. Оценка теоретических знаний:

Устные вопросы по теме практического задания:

Что такое исполнительный механизм

Способы управления ИМ

Виды сигналов

Практическое задание (на основе выданной технологической схемы):

- оценить полноту исходных данных;
- выбрать тип вариант схемы управления
- выполнить разработку и отладку алгоритма.

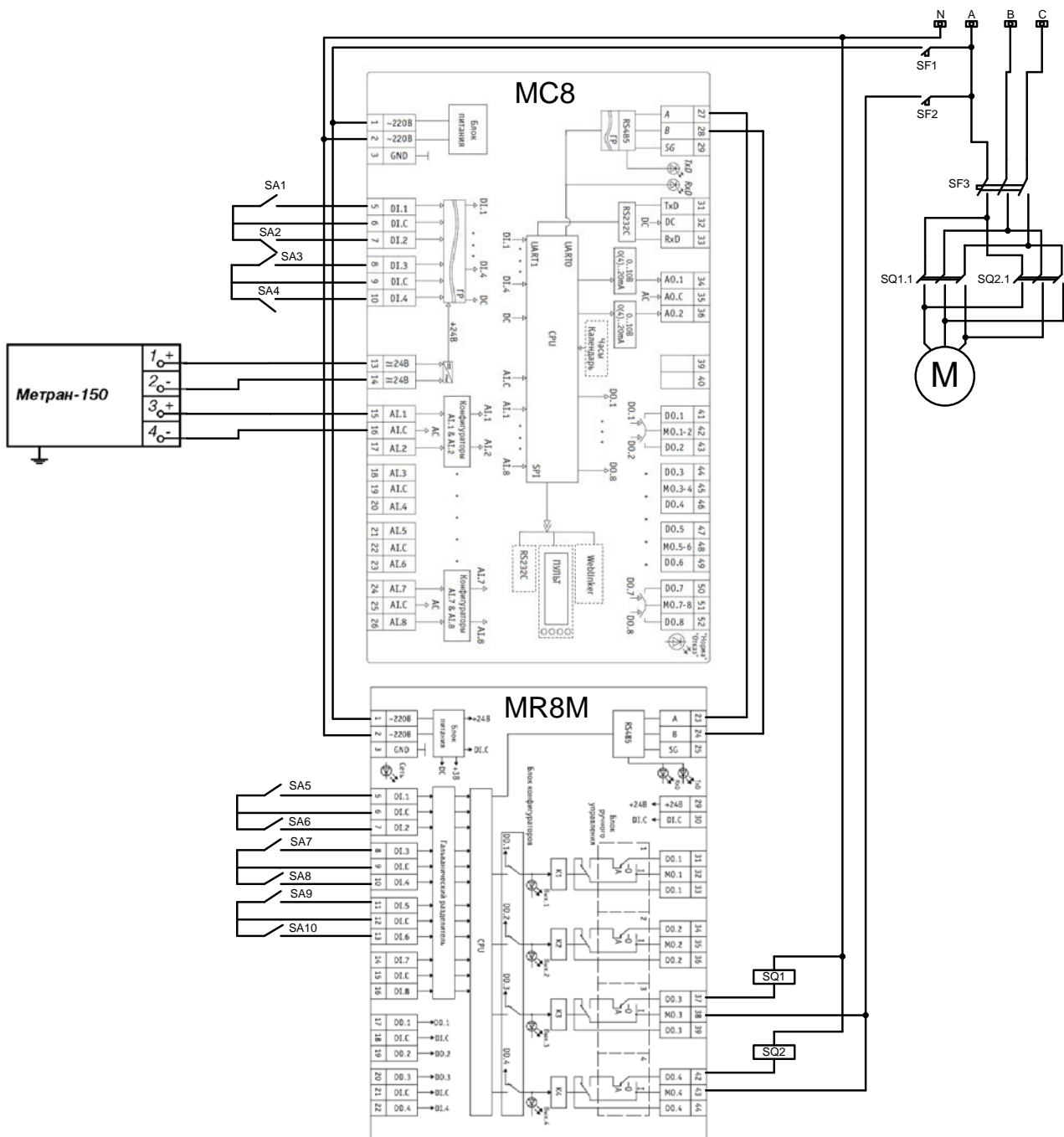


Рисунок 4.1 – Схема электрическая принципиальная

Практическая работа № 5. Тема: Вывод данных в SCADA-систему

1. Время на выполнение задания – ОФО - 6 часов, ЗФО – 4 часа.
2. Удовлетворительным результатом успешного выполнения практической работы считается корректная передача данных.

3. Оценка теоретических знаний:

Устные вопросы по теме практического задания:

Практическое задание (на основе выданной технологической схемы):

- оценить полноту исходных данных;
- выбрать тип канала связи
- выполнить разработку алгоритма и передачу данных.

Для отображения параметров технологического процесса АРМ, и для удаленного управления процессом используются SCADA-системы. В данной главе рассмотрим основные принципы работы со SCADA-системой TraceMode 6 [13].

TraceMode работает под Windows и Linux. В состав программы входят встроенные драйверы для более чем 2511 ПЛК и УСО. Все драйверы поставляются со средой разработки. Система поддерживает масштабируемость от 16 до 1 000 000 точек ввода/вывода, имеются специальные технологии работы с крупными проектами.

В системе существуют средства разработки систем телемеханики. Система реализует адаптивную самонастройку ПИД регуляторов, основанную на оригинальной российской патентованной технологии.

В процессе работы над проектом в SCADA – системе TraceMode выполняются следующие операции:

- создание экранных форм;
- написание программ;
- подключение к контроллеру;
- настройка связи с СУБД.

При запуске среды разработки открывается пустое окно IDE (рисунок 5.1).

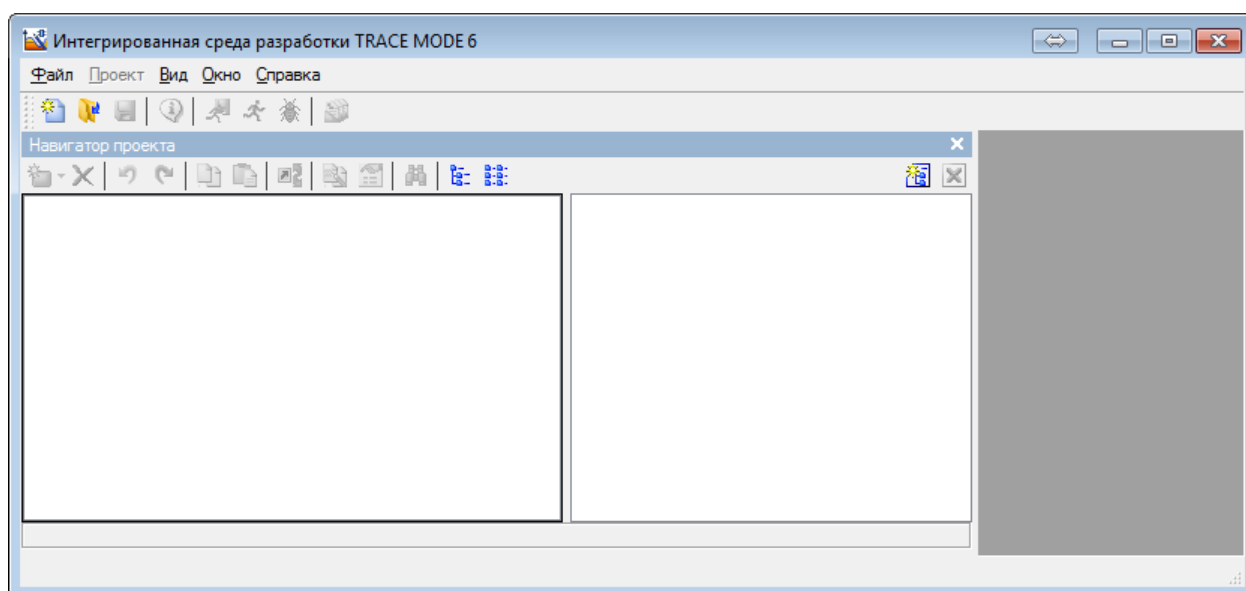


Рисунок 5.1 – Окно среды разработки TraceMode 6

Приступим к работе над проектом. Начнем с создания экранных форм. Для ускорения процесса разработки проекта и лучшего понимания принципов системы необходимо ознакомиться с разделом «Быстрый старт» справочной системы среды разработки TraceMode 6 [14].

Создадим новый проект. При выборе типа проекта укажем «Простой» (рисунок 5.2).

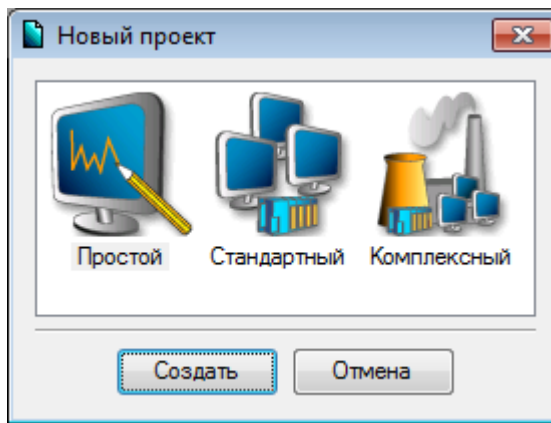


Рисунок 5.2 – Создание нового проекта TraceMode

При создании такого типа проектов в навигаторе будут указаны минимально необходимые элементы проекта (рисунок 5.3).

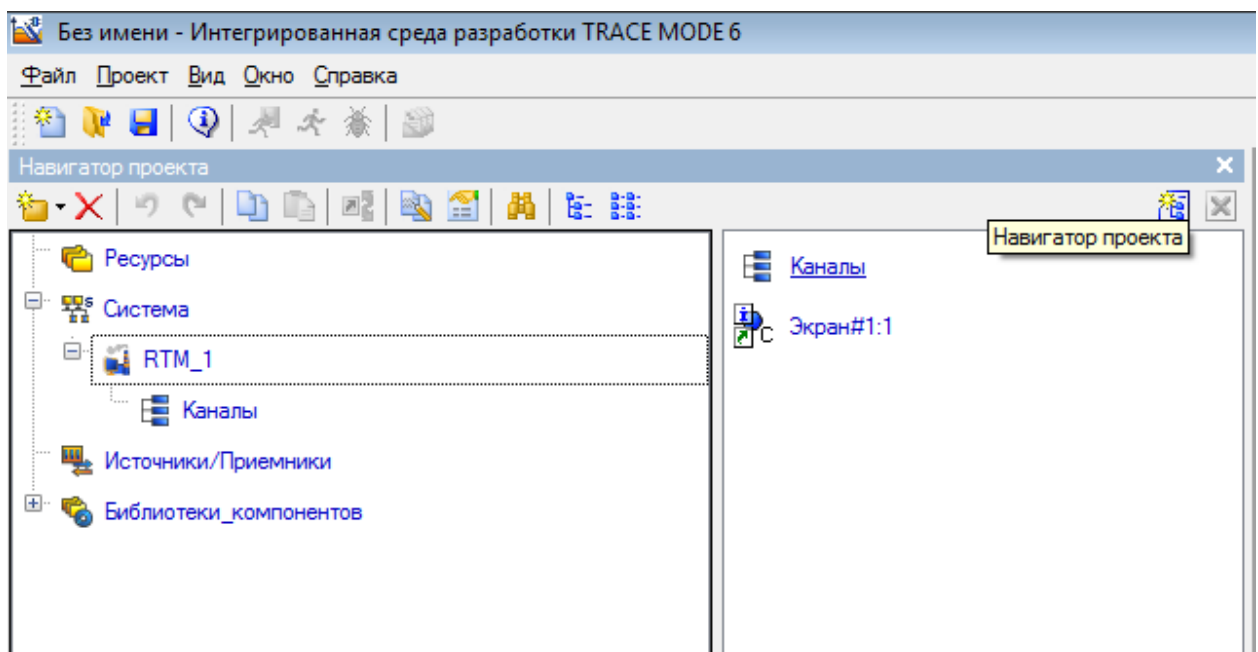


Рисунок 5.3 – Навигатор проекта

Теперь можно приступить к редактированию существующих экранных форм или созданию новых.

1.1. Создание экранных форм

В ветке RTM навигатора проекта уже присутствует элемент *Экран#1:1*, для создания дополнительных экранов нажмем правой кнопкой мыши в правой части навигатора проекта и выберем пункт *Создать компонент > Экран* (рисунок 5.4).

В результате данного действия будет создан второй экран с именем *Экран#2:2*. Откроем первый экран на редактирования двойным кликом.

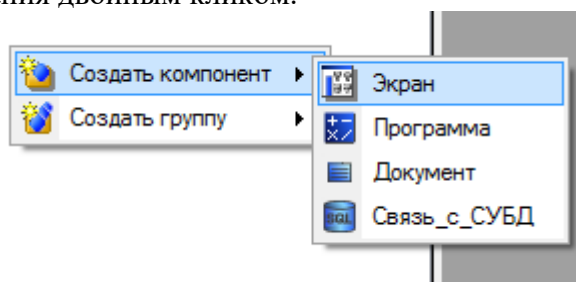


Рисунок 5.4 – Создание нового экрана

На панели инструментов графического редактора (Рисунок 5.5) выберем необходимые для создания мнемосхемы инструменты и фигуры.



Рисунок 5.5 – Панель инструментов графического редактора

Параметры экрана настраиваются в меню «Свойства объекта» (рисунок 5.6). Имеется возможность изменять размеры, фон, и источник света полотна для создания мнемосхемы.

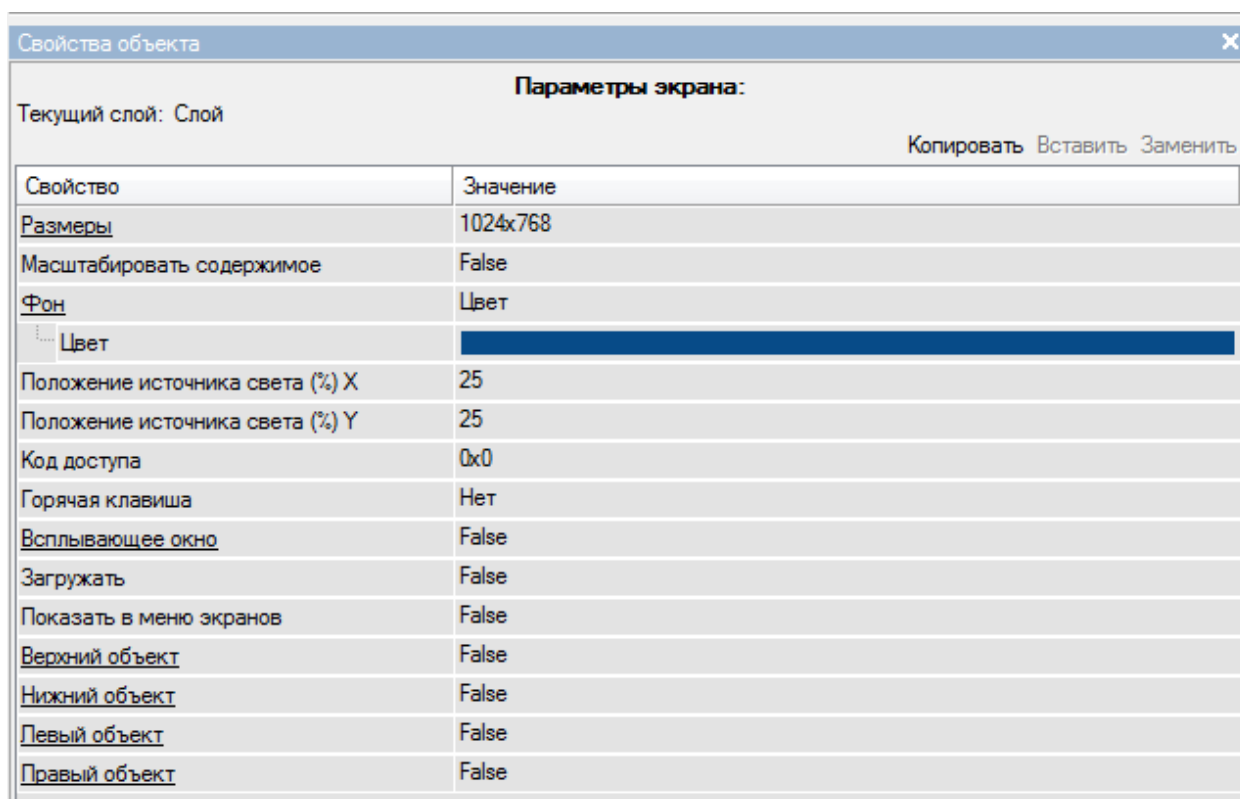


Рисунок 5.6 – Свойства экрана

Для отображения значений, полученных от контроллера, воспользуемся полем текстового вывода. Каждое поле привязываем к необходимому аргументу технологического параметра, полученного с полевого уровня.

Создадим мнемосхему, аналогичную разработанной для панели оператора (рисунок 5.7).

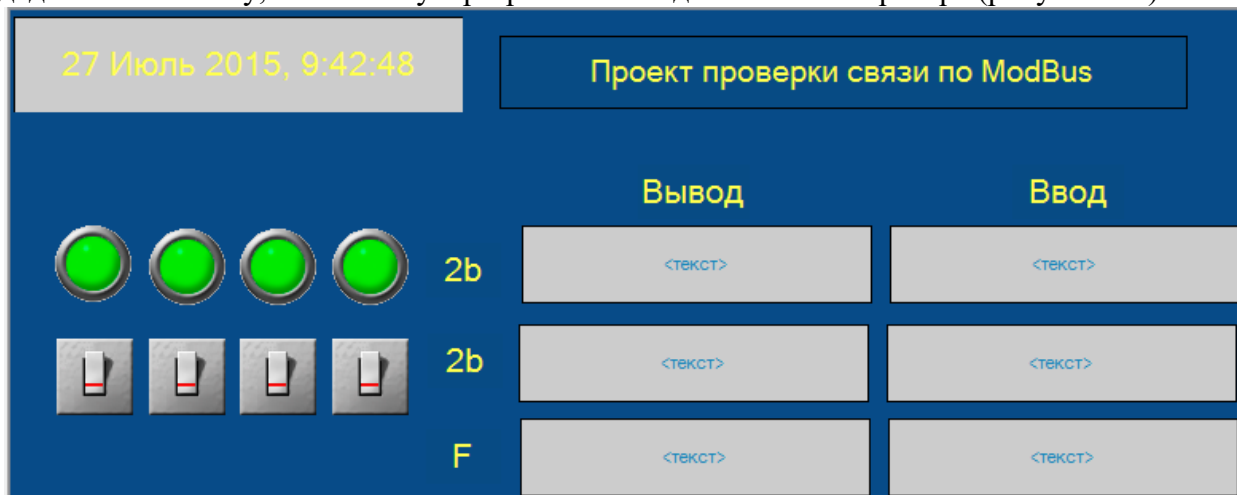


Рисунок 5.7 – Мнемосхема проекта

Настроим параметры объектов окна, для этого двойным кликом откроем элемент на редактирование (рисунок 5.8).

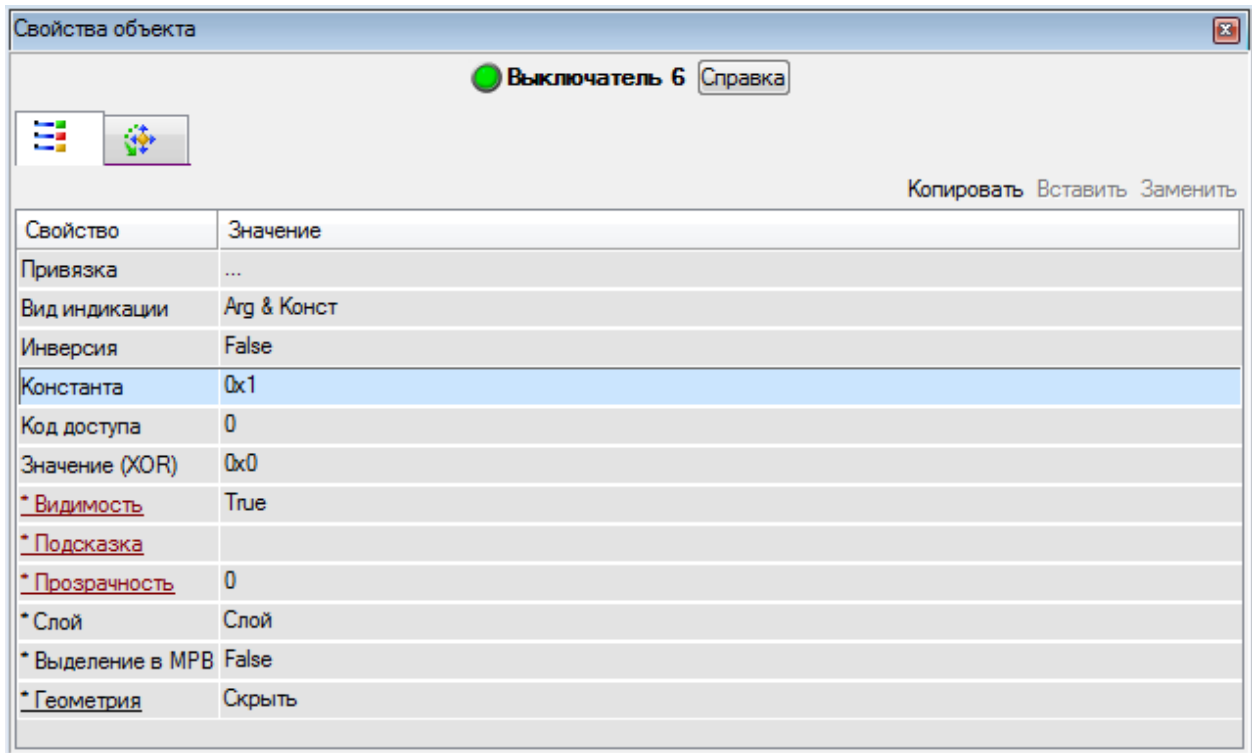



Рисунок 5.8 – ГЭ выключатель 6

В свойствах элемента укажем значение константы 0x1 и выполним привязку к тэгу окна. Нажмем на поле *Значение* свойства *Привязка* и в появившемся окне создадим новый тэг нажатием на кнопку , зададим имя DI1, укажем тип и тип данных (рисунок 5.9).

Аналогичным образом настроим оставшиеся 4 лампы и привяжем к ним созданные тэги. Аналогичным образом производится привязка и настройка выключателей. Для выключателя необходимо создать тэг с типом OUT, тип данных BOOL и указать кроме значения константы значение (XOR). Подробно о свойствах элементов написано в справочной системе среды TraceMode [7].

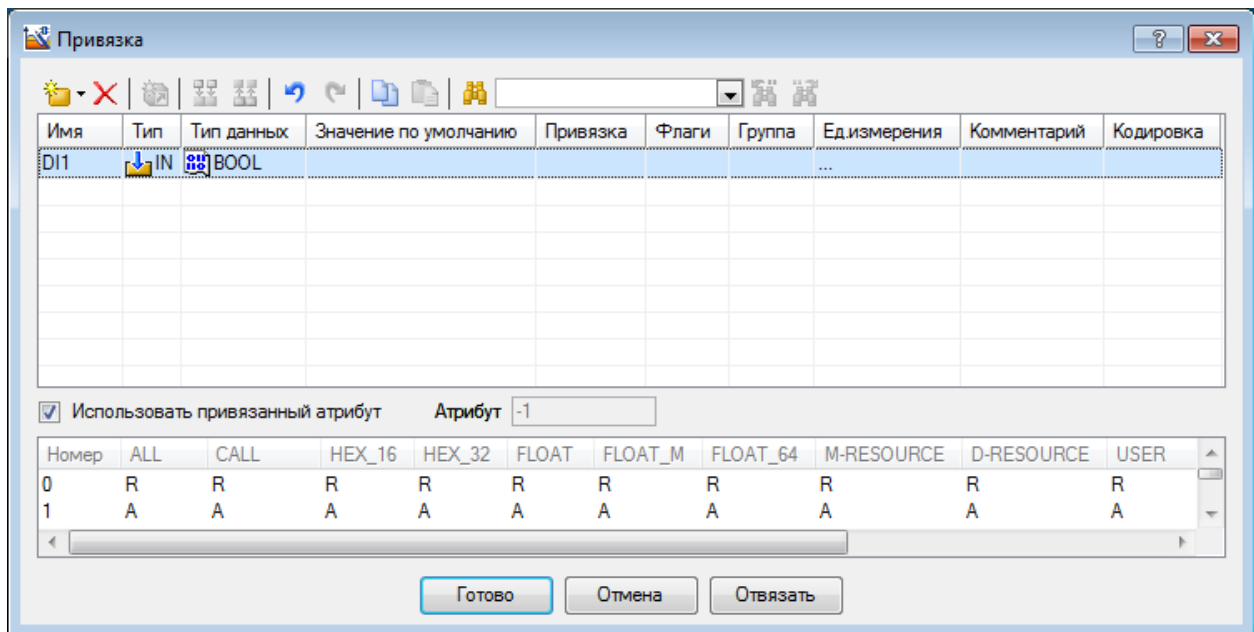


Рисунок 5.9 – Привязка тэга к ГЭ

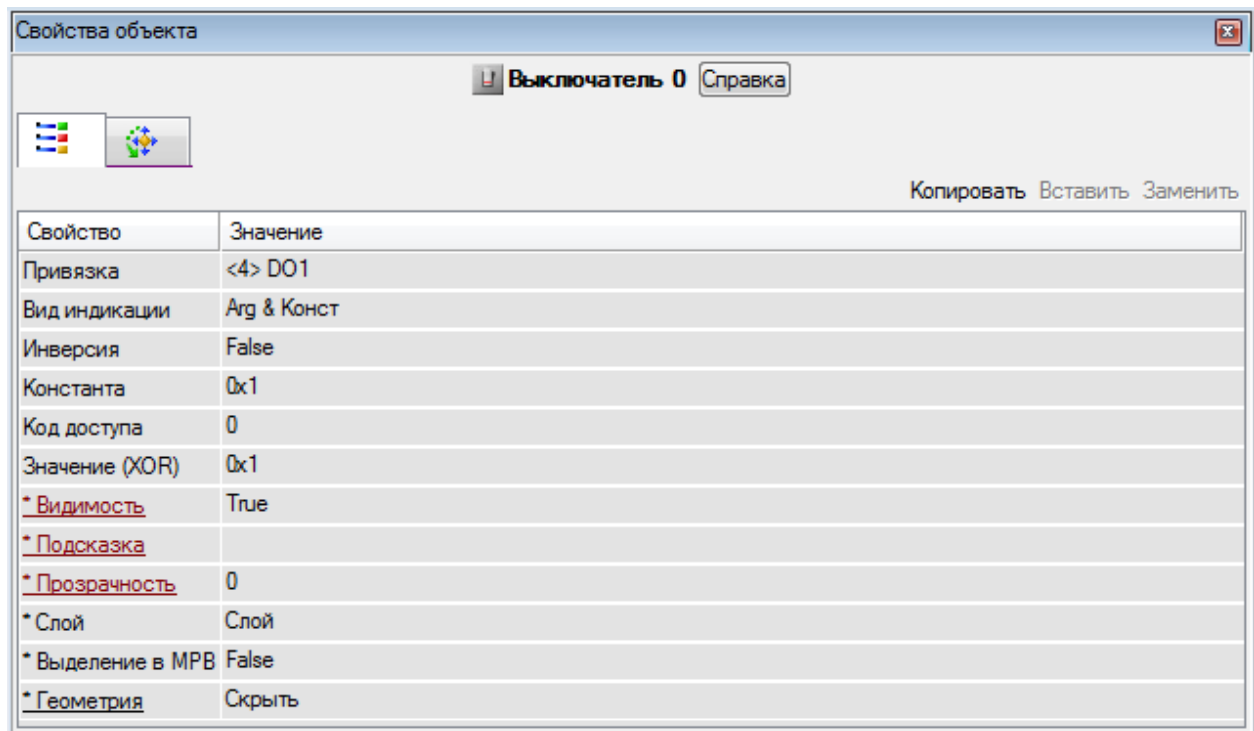


Рисунок 5.10 – Задание параметров ГЭ выключатель

После того как привязки дискретных сигналов выполнены приступим к работе с аналоговыми величинами. Для этого откроем на редактирование свойства поля текст и выполним привязку поля к вновь созданному тэгу экрана АИ1 с типом *IN* и типом данных *INT*.

Для поля ввода, расположенного справа от поля АИ1, выполним аналогичную привязку, создадим две переменных АО1 и АО1_I, одна будет использоваться для записи значения в регистр ModBus, вторая для отображения значения регистра. Драйвер ModBus в TraceMode реализован таким образом, что мы не можем в одном тэге драйвера обращаться как на чтение так и на запись.

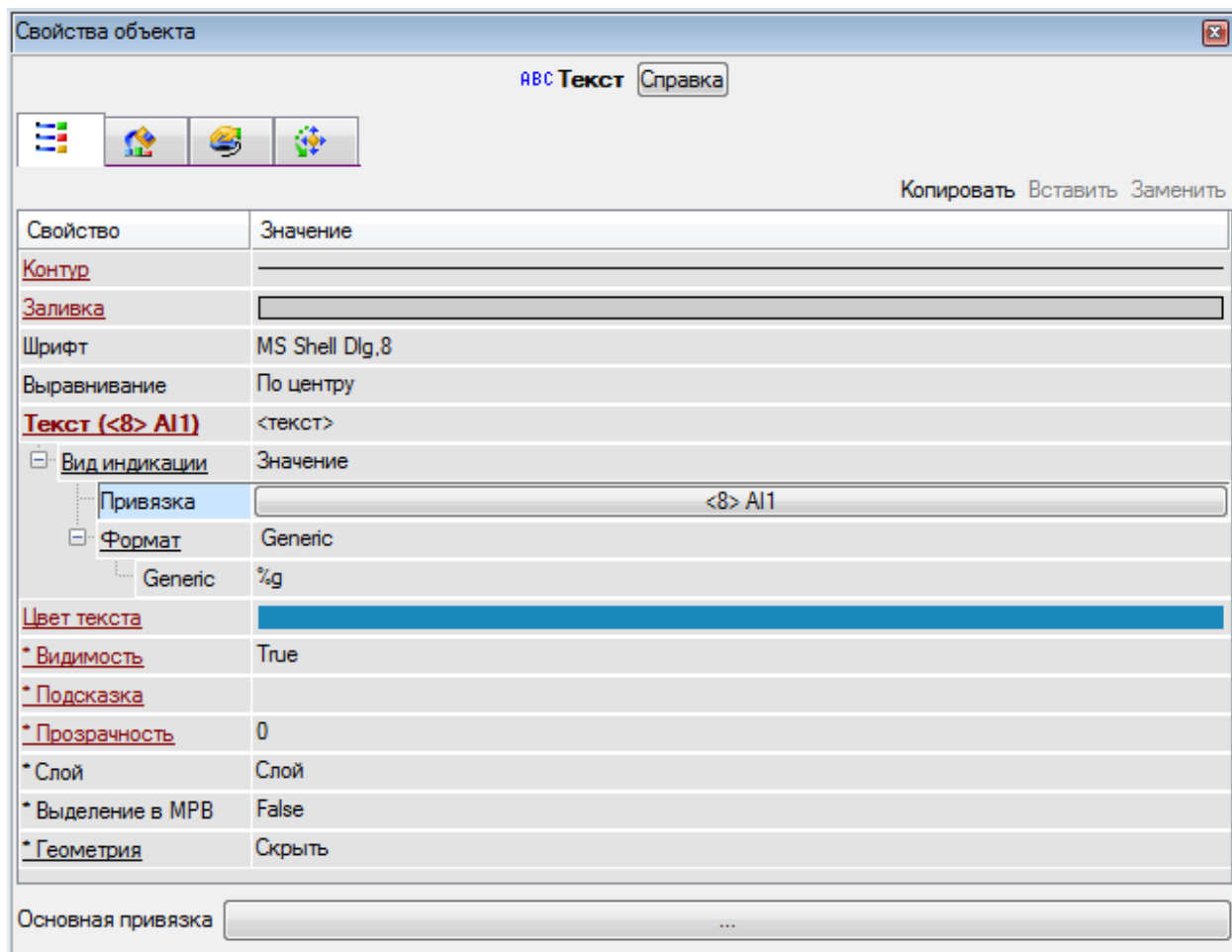


Рисунок 5.11 – Задание параметров ГЭ текст

Переменная AO1 будет выходной с типом *INT*, а переменная AO1_I будет входной с тем же самым типом. Привязку поля ГЭ текст выполним именно к AO1_I, т.к. именно это значение нужно будет отображать. Для настройки ввода значения перейдем на вкладку события и в разделе *MousePress* создадим элемент *Передать значение* нажатием правой кнопкой мыши (рисунок 5.12).

Выберем тип передачи *Ввести и передать* и привяжем элемент к тэгу (рисунок 5.13). При такой настройке элемента в поле текст будет выводиться считанное значение регистра, а при нажатии на текст будет открываться окно для ввода значения интересующего нас регистра.

По аналогии с панелью оператора в первых двух переменных мы будем передавать целочисленные значения.

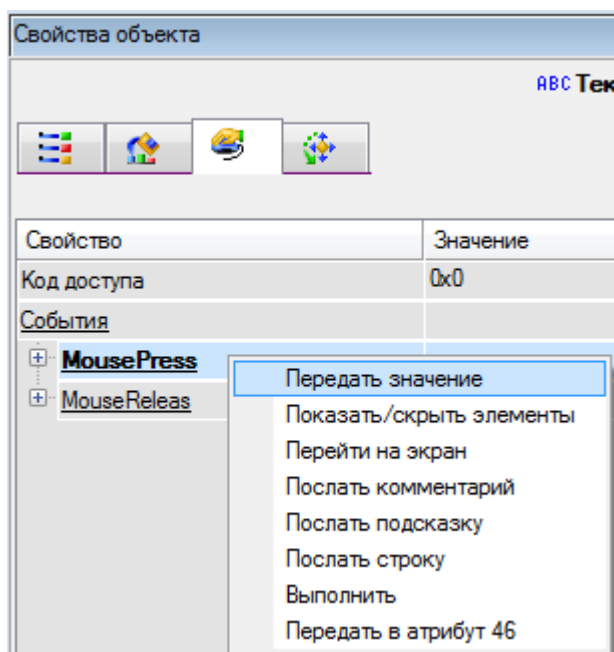


Рисунок 5.12 – Настройка поля ввода

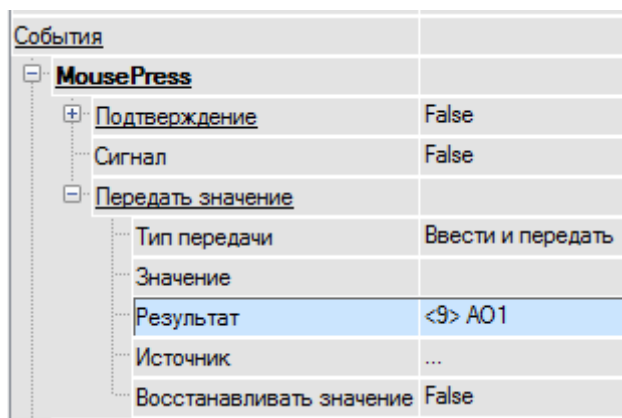


Рисунок 5.13 – Настройка события «Передать значение»

Для следующих двухбайтных переменных выполним аналогичные действия, но изменим тип переменных тэгов на *REAL* (рисунок 5.14).

AI2	IN	REAL							
AO2	OUT	REAL							
AO2_I	IN	REAL							

Рисунок 5.14 – Набор тэгов для переменных с фиксированной точкой

Для последнего набора аналоговых сигналов выполним привязку значений аналогичным второму набору образом.

После того как все переменные созданы приступим к настройке ModBus.

1.2. Подключение к контроллеру

Так как PLC поддерживает обмен по протоколу ModbusTCP, то для организации запроса данных в слое Источники/Приемники создадим группу компонентов-источников MODBUS_1(рисунок 5.15).

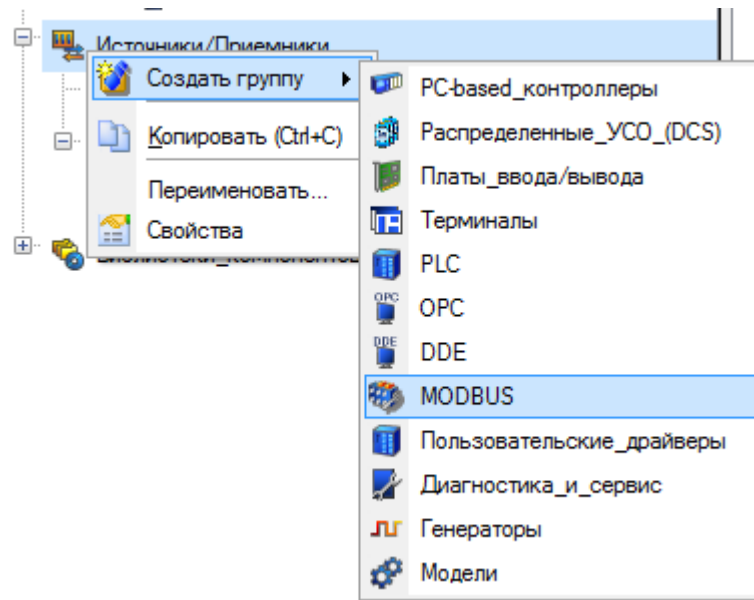


Рисунок 5.15 – Создание группы компонентов источника/приемники

В группе MODBUS_1 создадим следующий набор компонентов:

- Rin_Word(4) – для чтения 16 дискретных входов ModBus;
- W_Word(6) – для записи 16 дискретных выходов ModBus;
- Rin_Word(4) – для чтения AI1;
- W_Word(6) – для записи AO1;
- Rin_Word(4) – для чтения состояния AO1;
- Rin_Word(4) – для чтения AI2;
- W_Word(6) – для записи AO2;
- Rin_Word(4) – для чтения состояния AO2;
- Rin_Float(4) – для чтения AI3;
- W_Float(16) – для записи AO3;
- Rin_Float(4) – для чтения состояния AO3;

Отредактируем компоненты-источники с учетом адресации ПЛК и расположения параметров в его адресном пространстве. Для чтения дискретных сигналов изменим имя канала, для удобства установим его в *D11-16(4x:0)*, зададим комментарий и настроим адресацию. Адрес устройства у нас *1*, тип протокола *TCP/IP*, канал будет *0* т.к. мы считываем регистр *4x:0*, порт оставляем по умолчанию и строке IP-адрес указываем адрес нашего контроллера.

Для всех остальных тэгов выполняется аналогичная настройка с указанием корректного значения адреса канала.

В случае работы с аналоговыми сигналами следует выбирать нужный формат.

Результующий перечень регистров приведен на рисунке 5.17.

Основные

Имя: DI1-16(4x:0)

Кодировка: TW0

Комментарий: Дискретные входы

Справка

Параметры

Номер порта: 0x0

Адрес: 0x1

Канал: 0x0

Тип протокола: TCP/IP

Направление: Input

Формат: Дискрет

IP-адрес: 192.168.1.10

Рисунок 5.16 – Редактирование компонента Rin_Word(4)

- DI1-16(4x:0)
- DO1-16(4x:1)
- AI1(4x:2)
- AO1(4x:3)
- AO1_I(4x:3)
- AI2(4x:4)
- AO2(4x:5)
- AO2_I(4x:5)
- AI3(4x:6)
- AO3(4x:8)
- AO3_I(4x:8)

Рисунок 5.17 – Перечень регистров ModBus

Для корректной работы TraceMode с регистрами типа Float необходимо после сохранения проект открыть папку, где расположен проект, найти подпапку *RTM_1* и в ней создать файл с именем *modbus.set* в файле вписать цифру 2 и сохранить.

Такая конфигурация связана с тем, что TraceMode поддерживает отклонения от стандарта ModBus при работе с переменными FLOAT. По этому, для выбора порядка следования байтов нужно вручную создать текстовый файл *modbus.set* в папке узла. Файл должен содержать целое число от

0 до 3, указывающее следующий порядок байтов (байт 0 – младший байт мантииссы, байт 3 – байт знака и порядка):

- 0 – 3-2-1-0 (данный порядок следования байтов устанавливается в отсутствие файла modbus.set);
- 1 – 0-1-2-3;
- 2 – 1-0-3-2 (порядок, установленный для корректной работы);
- 3 – 2-3-0-1.

1.3. Написание программ

Для удобства работы с дискретными сигналами нам потребует распаковать слово в биты и упаковать биты дискретных выходов в слово.

Для этого создадим новый компонент узла RTM_1. Выберем в качестве компонента программу (рисунок 5.18).

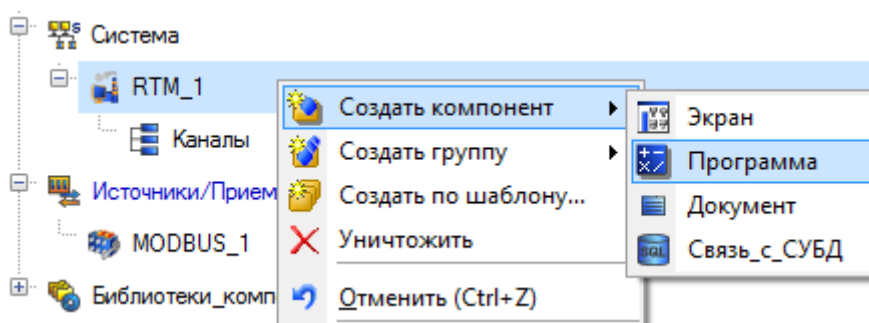



Рисунок 5.18 – Создание новой программы

Двойным кликом мыши на компоненте откроем окно редактора программ и, выделив пункт *Аргументы*, перейдем в табличный редактор аргументов. Создадим аргументы для данной программы руководствуясь требуемым функционалом (рисунок 5.19).

После определения входных и выходных аргументов приступим непосредственно к разработке программы. Для этого в структуре программы выделим строку *Программа#1* и в появившемся диалоге выбора языка программирования укажем *FBD диаграмма* (рисунок 5.20).

В открывшемся окне редактора программ выберем иконку для доступа к библиотекам

функциональных блоков , далее выбирая необходимые блоки, перетаскиваем их в рабочее поле редактора, группируем, определяем внутренние связи между входами и выходами блоков, назначаем привязки к аргументам. Готовая программа выглядит следующим образом (рисунок 5.21).

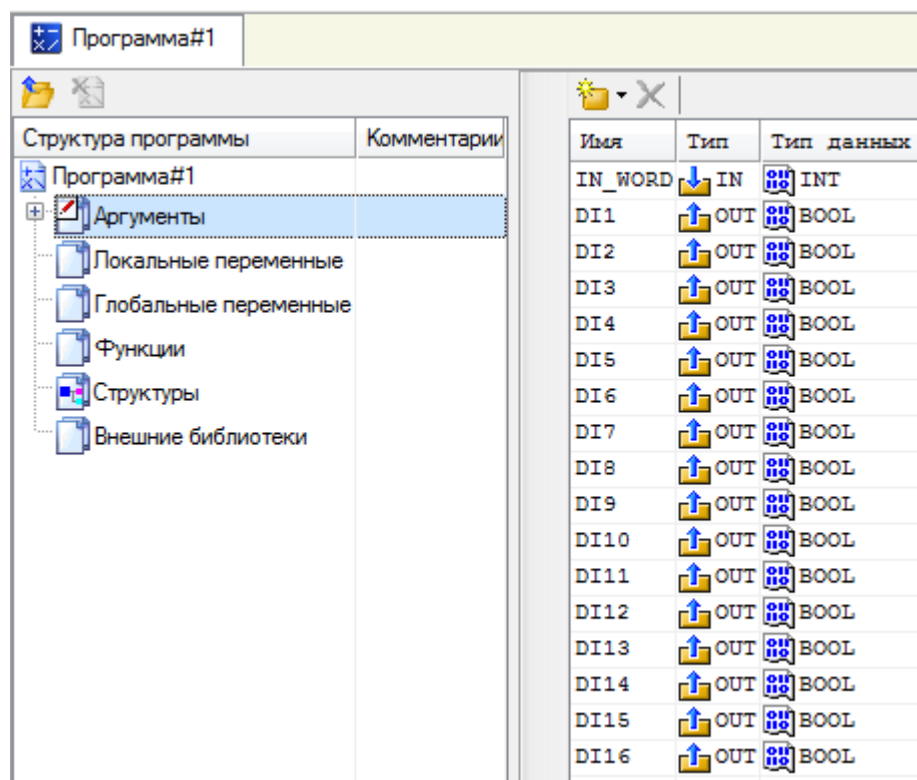


Рисунок 5.19 – Аргументы программы

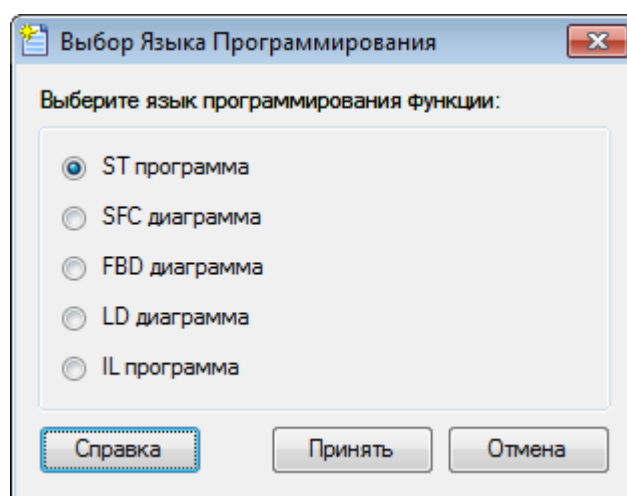



Рисунок 5.20 – Выбор языка программы

Созданную программу необходимо откомпилировать, для этого нажмем на кнопку компиляция . Для отображения результатов компиляции нажмите в на вкладке *Вид > Сообщения*. В случае успешной компиляции вы должны увидеть текст, аналогичный приведенному на рисунке 5.22. Закроем редактор программ и переименуем программу нажатием правой кнопкой мыши в дереве компонентов узла и выбрав элемент *Переименовать*. Зададим программе имя «Распаковка слова в биты».

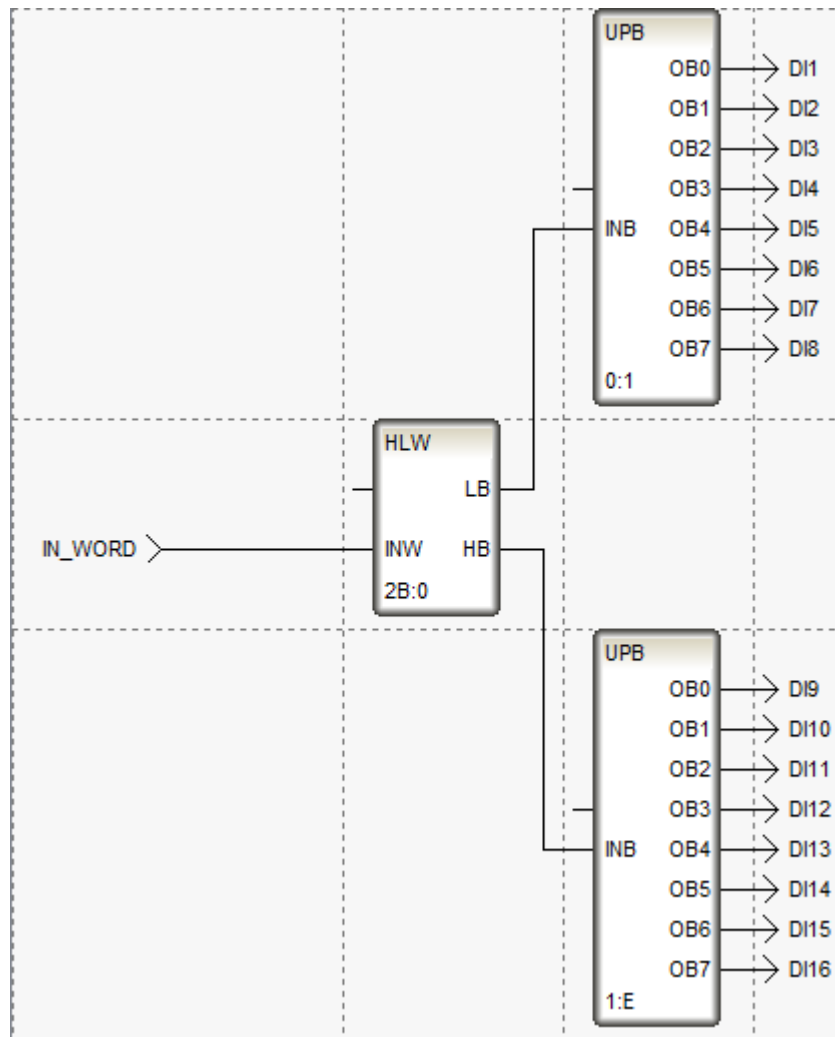


Рисунок 5.21 – Программа распаковки слова

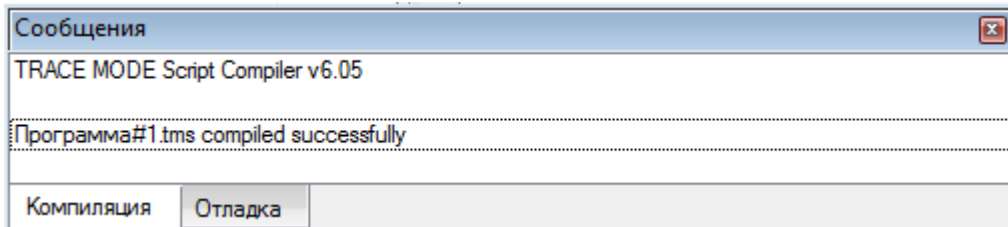


Рисунок 5.22 – Компиляция проекта

Аналогичным образом создается программа упаковки битов в слова. Создадим новую программу, назовем ее «Упаковка битов в слова». Укажем аргументы и установим линии связи между блоками. Результирующим вариантом программы будет диаграмма, приведенная на рисунке 5.23.

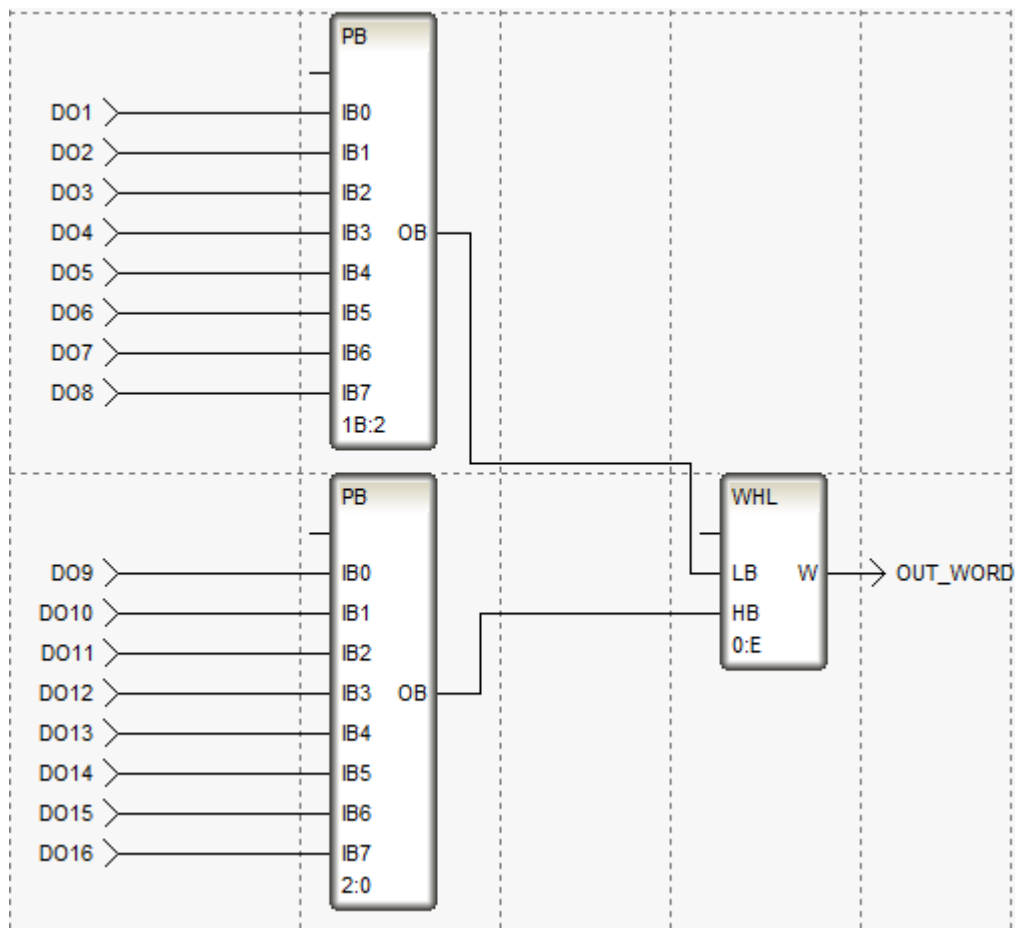



Рисунок 5.23 – Программа упаковки битов в слово

1.4. Привязка тэгов к источникам

Откроем свойства экрана для привязки тэгов (рисунок 5.24). Во вкладке *Аргументы* свойств экрана выделим тэг DI1. При помощи кнопки  создадим переменные узла для каждого из тэгов экрана (рисунок 5.25).

Для привязки дискретных входов откроем свойства программы «Распаковка слова в биты» и во вкладке *Аргументы* произведем создание переменной IN_WORD и привязку переменных узла RTM_1 ко входам программы (рисунок 5.26). Привязка производится двойным нажатием на входе с последующим выбором нужной переменной (рисунок 5.27).

Дискретные выходные сигналы будем привязывать к программе упаковки. Помимо привязки создадим переменную OUT_WORD для выходной переменной программы.

Далее с помощью функции Drag & Drop привяжем регистры ModBas к созданным переменным узла RTM_1 (аналоговые входы/выходы, IN_WORD и OUT_WORD).

Отроем на редактирование созданные переменные. Для каналов AI2 и AO2_I укажем множитель 0,01 (рисунок 5.28). А для канала AO2 коэффициент 100.

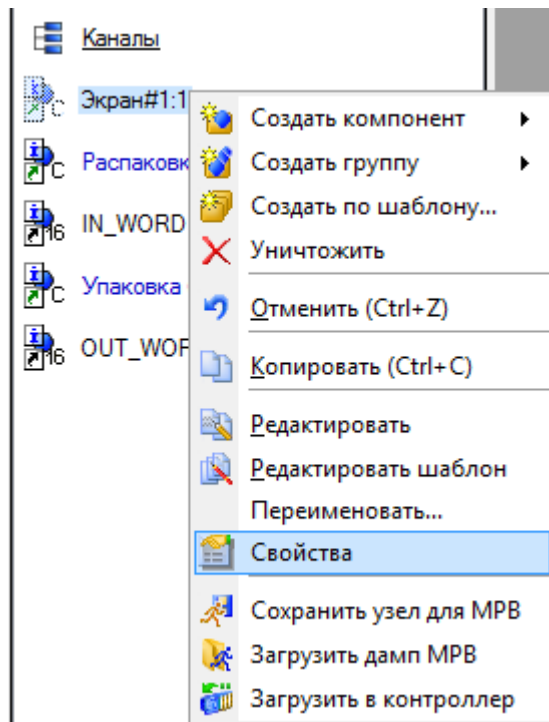


Рисунок 5.24 – Открытие свойств экрана

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
DI1	IN	BOOL		DI1:Реальное значение (Система.RTM_1)
DI2	IN	BOOL		DI2:Реальное значение (Система.RTM_1)
DI3	IN	BOOL		DI3:Реальное значение (Система.RTM_1)
DI4	IN	BOOL		DI4:Реальное значение (Система.RTM_1)
DO1	OUT	BOOL		DO1:Входное значение (Система.RTM_1)
DO2	OUT	BOOL		DO2:Входное значение (Система.RTM_1)
DO3	OUT	BOOL		DO3:Входное значение (Система.RTM_1)
DO4	OUT	BOOL		DO4:Входное значение (Система.RTM_1)
AI1	IN	INT		AI1:Реальное значение (Система.RTM_1)
AO1	OUT	INT		AO1:Входное значение (Система.RTM_1)
AO1_I	IN	INT		AO1_I:Реальное значение (Система.RTM_1)
AI2	IN	REAL		AI2:Реальное значение (Система.RTM_1)
AO2	OUT	REAL		AO2:Входное значение (Система.RTM_1)
AO2_I	IN	REAL		AO2_I:Реальное значение (Система.RTM_1)
AI3	IN	REAL		AI3:Реальное значение (Система.RTM_1)
AO3	OUT	REAL		AO3:Входное значение (Система.RTM_1)
AO3_I	IN	REAL		AO3_I:Реальное значение (Система.RTM_1)

Рисунок 5.25 – Создание переменных для тэгов экрана

На этом конфигурация экранной формы заканчивается.

Также рассмотрим вторую технологию связи с контроллером – OPC. Но для выполнения данного этапа проекта необходимо ознакомиться с главой 8 и выполнить указанные там действия.

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
IN_WORD	IN	INT		IN_WORD:Реальное значение (Система.RTM_1)
DI1_R	OUT	BOOL		DI1:Реальное значение (Система.RTM_1)
DI2_R	OUT	BOOL		DI2:Реальное значение (Система.RTM_1)
DI3_R	OUT	BOOL		DI3:Реальное значение (Система.RTM_1)
DI4_R	OUT	BOOL		DI4:Реальное значение (Система.RTM_1)
DI5	OUT	BOOL		

Рисунок 5.26 – Привязка DI

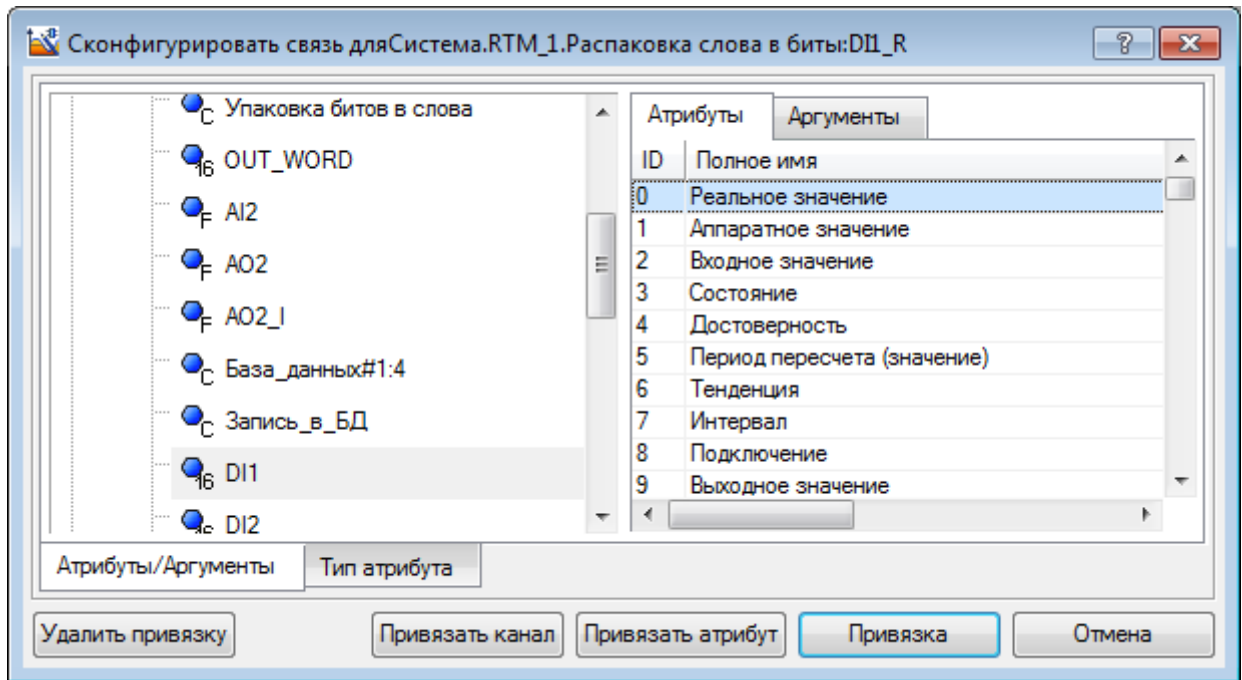


Рисунок 5.27 – Выбор переменной при привязке

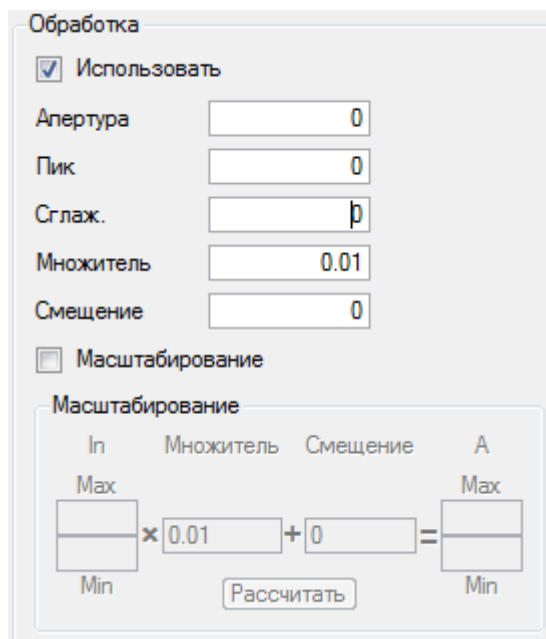


Рисунок 5.28 – Масштабирование сигнала

1.5. Работа с OPC

После того как OPC-сервер сконфигурирован и запущен можно приступать к настройке связи и выбору тэгов. Для этого разделе *Источники/Приемники* создадим группу OPC (рисунок 5.28). В группе добавим новый OPC-сервер (рисунок 5.29) и приступим к созданию тэгов (рисунок 5.30). Откроем тэг на редактирование, зададим ему нужное имя и нажмем на кнопку *Обзор*. В браузере OPC (рисунок 5.31) выберем интересующий нас сервер и в правой части окна укажем тэг. Созданный тэг может быть привязан к атрибутам экрана аналогично регистрам ModBus.

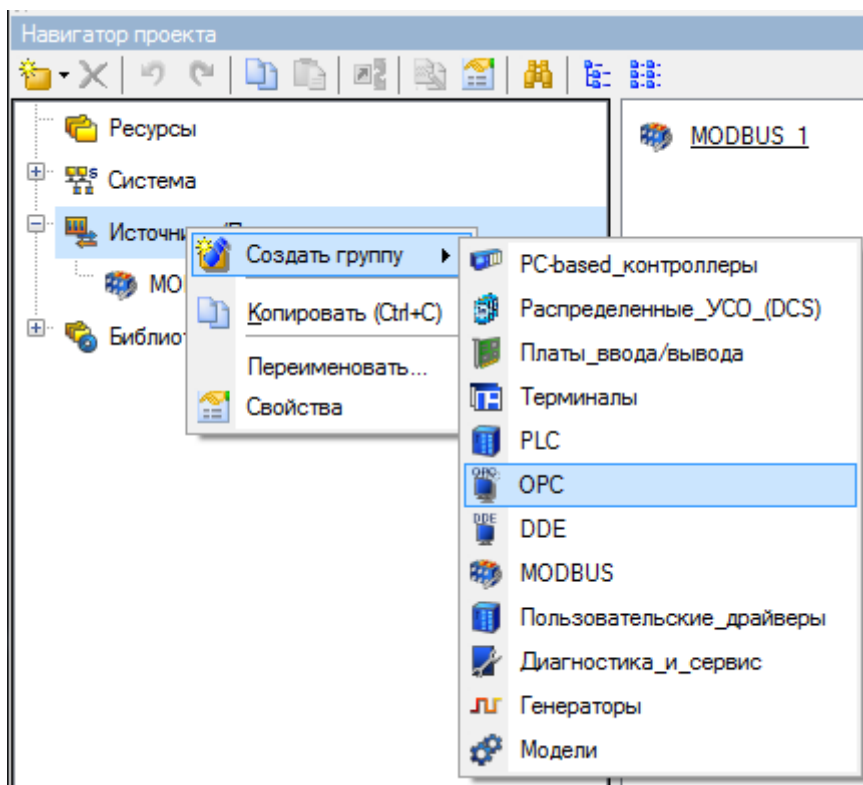


Рисунок 5.28 – Создание группы источников OPC

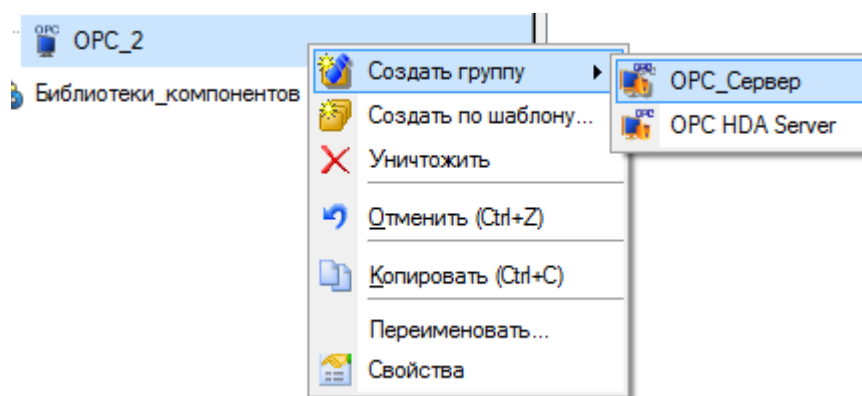


Рисунок 5.29 – Добавление OPC-сервера

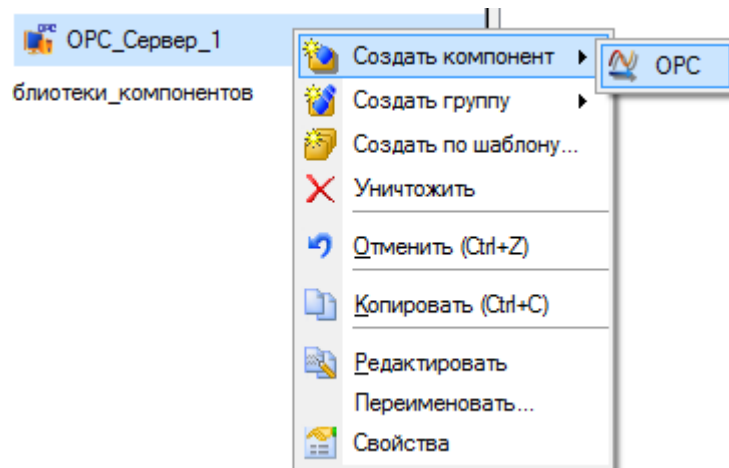


Рисунок 5.30 – Создание тэга ОПС

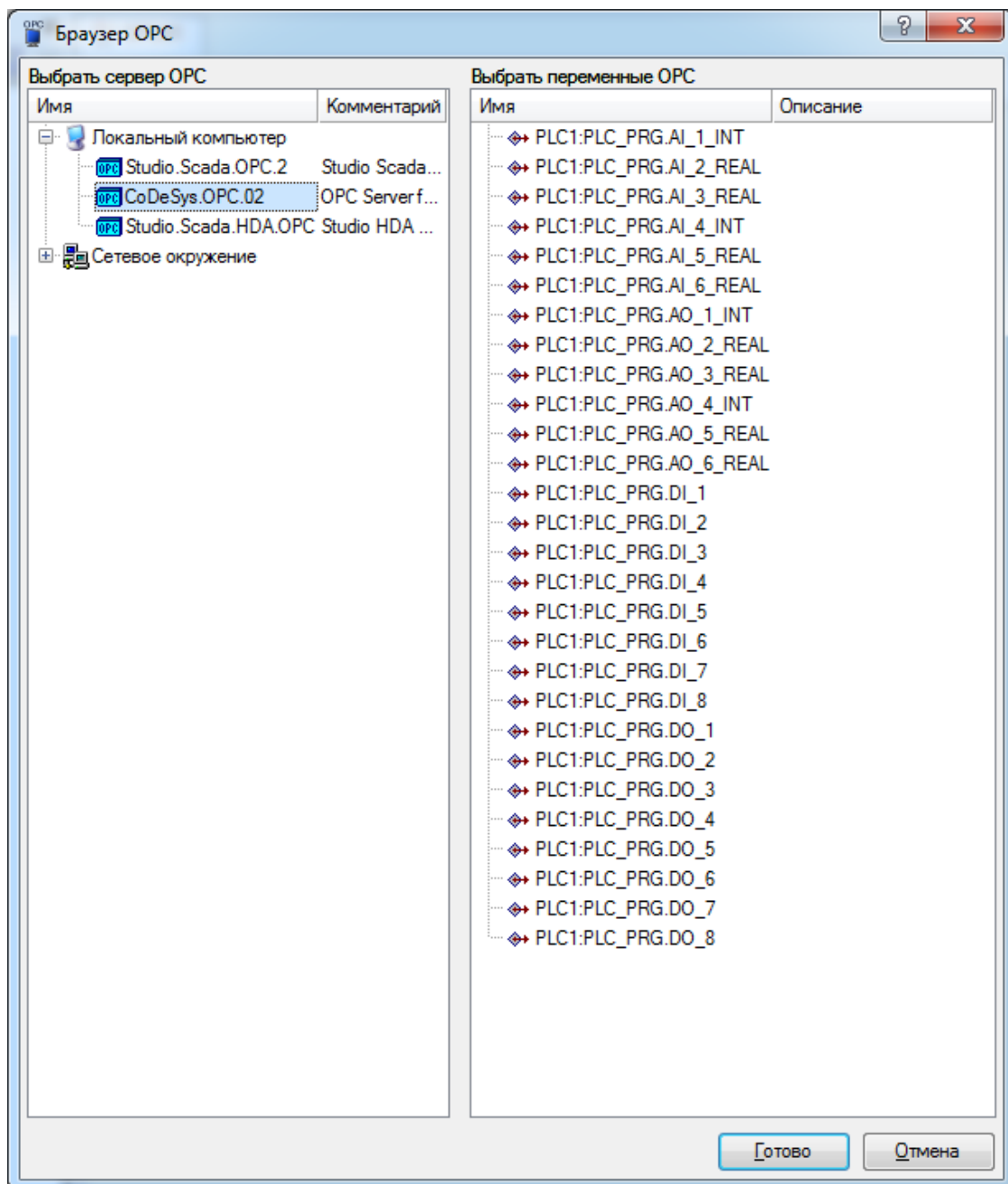


Рисунок 5.31 – Браузер ОПС