

КОНТРОЛЛЕРЫ ЛОГИЧЕСКИЕ ПРОГРАММИРУЕМЫЕ
ОПТИLOGIC L

Быстрый старт MasterScada 4D



РОССИЯ, 305000, Г. КУРСК, УЛ. ЛУНАЧАРСКОГО, 8
WWW.KEAZ.RU

Содержание

1	Среда разработки	3
2	Исполнительная система	4
3	Клиент визуализации.....	4
4	Установка среды разработки MasterSCADA 4D.....	5
5	Описание ярлыков MasterSCADA 4D.....	5
6	Разработка простого проекта для OptiLogic L в MasterSCADA 4D.....	6
7	Создание проекта	6
8	Конфигурирование дерева системы	9
9	Создание логической части проекта	11
10	Создание окна управления	19
11	Запуск проекта.....	28
12	Разработка всплывающего окна.....	33
13	Объектный подход при создании проекта.....	38
14	Операции с библиотеками.....	48

Программирование управляющей программы контроллера осуществляется в платформе для автоматизации и диспетчеризации MasterSCADA 4D.

Состав MasterSCADA 4D:

- Среда разработки
- Исполнительная система, загруженная в контроллер
- Клиент визуализации

В среда разработки создается проект, который запускается в реальном времени под управлением исполнительной системы в контроллере, доступ к графической части проекта осуществляется при помощи клиента визуализации.

1 СРЕДА РАЗРАБОТКИ

Среда разработки является приложением Windows

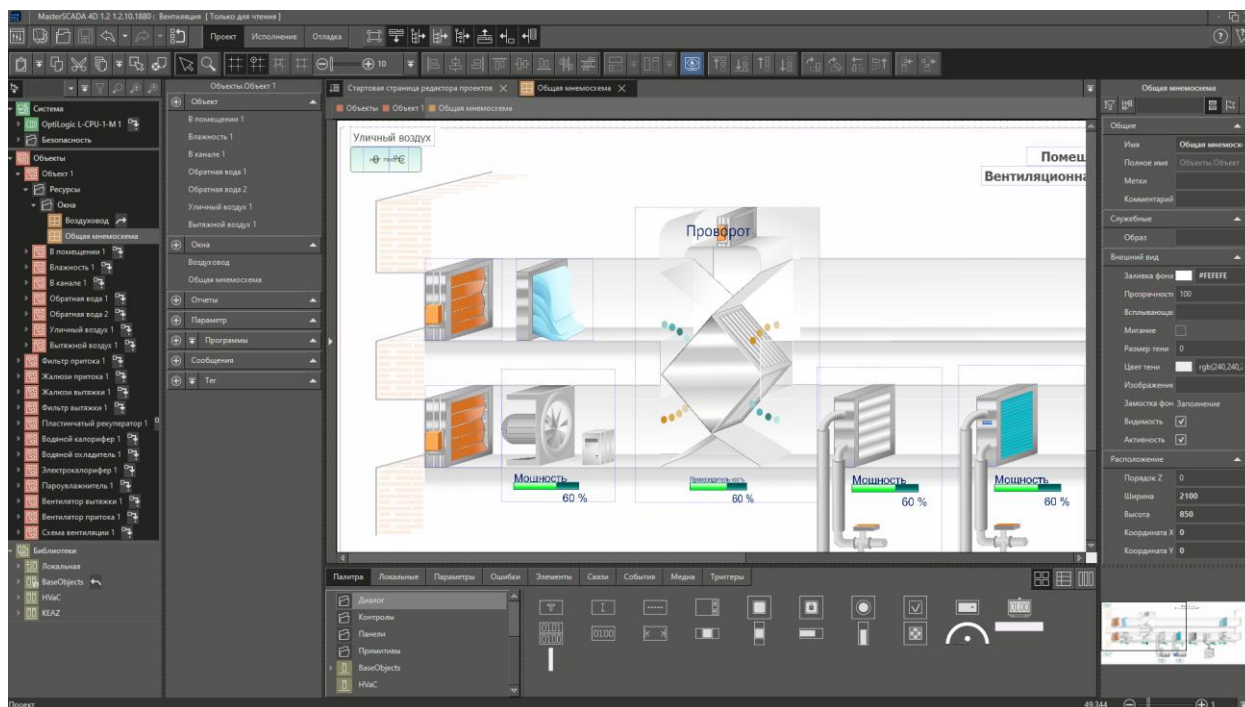


Рисунок 1 – Общий вид среды разработки

Проект создается в рамках единой инструментальной среды. Никаких других инструментов или редакторов не требуется. Основным инструментом для создание элементов проекта (управляющих программ, графических объектов, журналов, трендов и т.д.) является контекстное меню, а для настройки связей между элементами – механизм drag-and-drop. Среда разработки имеет большой набор инструментов для тиражирования, автоматизации рутинных операций, а также online и offline отладки.

Минимальные системные требования:

- операционная система (ОС) – Windows Server 2008 R2 SP1 x64 или Windows 7 SP1 x64;

- процессор – Intel® Core™ i3, 2.3 ГГц
- ОЗУ – 8 Гб
- дисплей – 1280x1024, цвет – 32 бита
- свободное место на диске – 10 Гб
- клавиатура, мышь

Рекомендуемые системные требования:

- ОС – Windows 10 x64 или старше
- процессор – современный многоядерный, не ниже Intel® Core™ i5, 3.4 ГГц
- ОЗУ – не менее 16 Гб
- дисплей – 1920x1080, цветность – 32 бита
- жесткий диск – SSD
- свободное место на диске – 100 Гб
- клавиатура, мышь.

2 ИСПОЛНИТЕЛЬНАЯ СИСТЕМА

В исполнительной системе происходит исполнение созданного проекта.

В рамках одного проекта MasterSCADA 4D позволяет исполнять проект (или части проекта) на:

- Программируемых логических контроллерах
- HMI-панелях
- Промышленных компьютерах
- Архивных и технологических серверах
- Облачных сервисах

Для этих элементов систем управления в MasterSCADA 4D имеются специализированные исполнительные системы (run-time). Они устанавливаются на данные устройства, и по команде среды разработки загружают и запускают на исполнение предназначенные для них части проекта. ПЛК OptiLogic поставляется с уже установленной исполнительной системой.

3 КЛИЕНТ ВИЗУАЛИЗАЦИИ

Для представления графической информации в MasterSCADA используется стандарт HTML5. Для его реализации в состав серверной части большинства исполнительных модулей входит WEB-сервер. Он

формирует страницы формата HTML5, которые визуализирует графический клиент MasterSCADA. В качестве клиента можно использовать любое устройство, имеющее в своем составе web-браузер.

4 УСТАНОВКА СРЕДЫ РАЗРАБОТКИ MASTERSCADА 4D

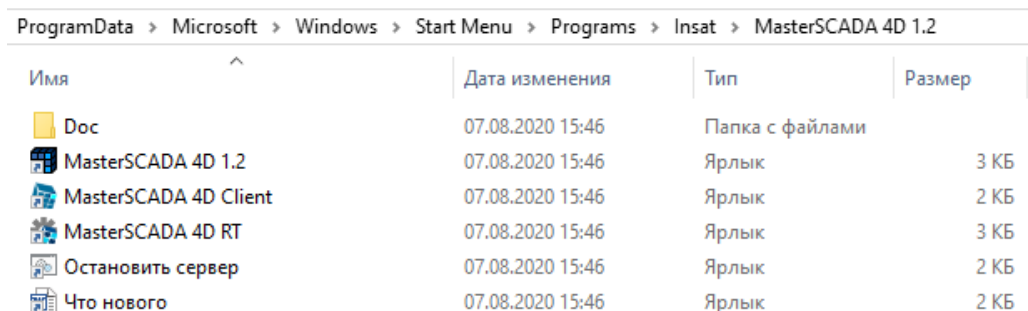
В комплект поставки входят 2 установочных файла:

- MasterSCADA4D_x86.exe – установочный файл для 32-разрядных ОС
- MasterSCADA4D_x64.exe – установочный файл для 64-разрядных ОС

Для установки среды запустите файл MasterSCADA 4D с диска, входящего в комплект поставки, или загруженного с сайта производителя – insat.ru (<http://insat.ru/services/support/demos/>), и следуйте командам инсталлятора.

5 ОПИСАНИЕ ЯРЛЫКОВ MASTERSCADА 4D

По завершению установки мастер создает программную группу Пуск - Все программы - InSAT - MasterSCADA 4D, содержащую следующие пункты меню:



Имя	Дата изменения	Тип	Размер
Дос	07.08.2020 15:46	Папка с файлами	
MasterSCADA 4D 1.2	07.08.2020 15:46	Ярлык	3 КБ
MasterSCADA 4D Client	07.08.2020 15:46	Ярлык	2 КБ
MasterSCADA 4D RT	07.08.2020 15:46	Ярлык	3 КБ
Остановить сервер	07.08.2020 15:46	Ярлык	2 КБ
Что нового	07.08.2020 15:46	Ярлык	2 КБ

Рисунок 2 – Состав меню программной группы

Дос – содержит следующие ярлыки:

- Лицензионное соглашение – открывает лицензионное соглашение
- Разработка драйверов и ФБ в MasterSCADA – открывает файл Разработка драйверов и ФБ в MasterSCADA 4D.docx

• Справка – открывает справку

MasterSCADA 4D – запускает среду разработки

MasterSCADA 4D Client – запускает клиент MasterSCADA 4D

MasterSCADA 4D RT – запускает исполнительную систему

Остановить сервер – останавливает исполнительную систему

Что нового – запускает файл WhatsNew.rtf

6 РАЗРАБОТКА ПРОСТОГО ПРОЕКТА ДЛЯ OPTILOGIC L В MASTERSCADA 4D

Рассмотрим создание проекта, реализующего управление включением и выключением устройства двумя кнопками – ПУСК и СТОП. При нажатии на кнопку ПУСК устройство должно включаться, при нажатии на кнопку СТОП – выключаться.

7 СОЗДАНИЕ ПРОЕКТА

Для того, чтобы начать создание проекта необходимо запустить среду разработки MasterSCADA 4D, выполнив пункт меню Windows Пуск\Все программы\InSAT\MasterSCADA 4D\MasterSCADA 4D.

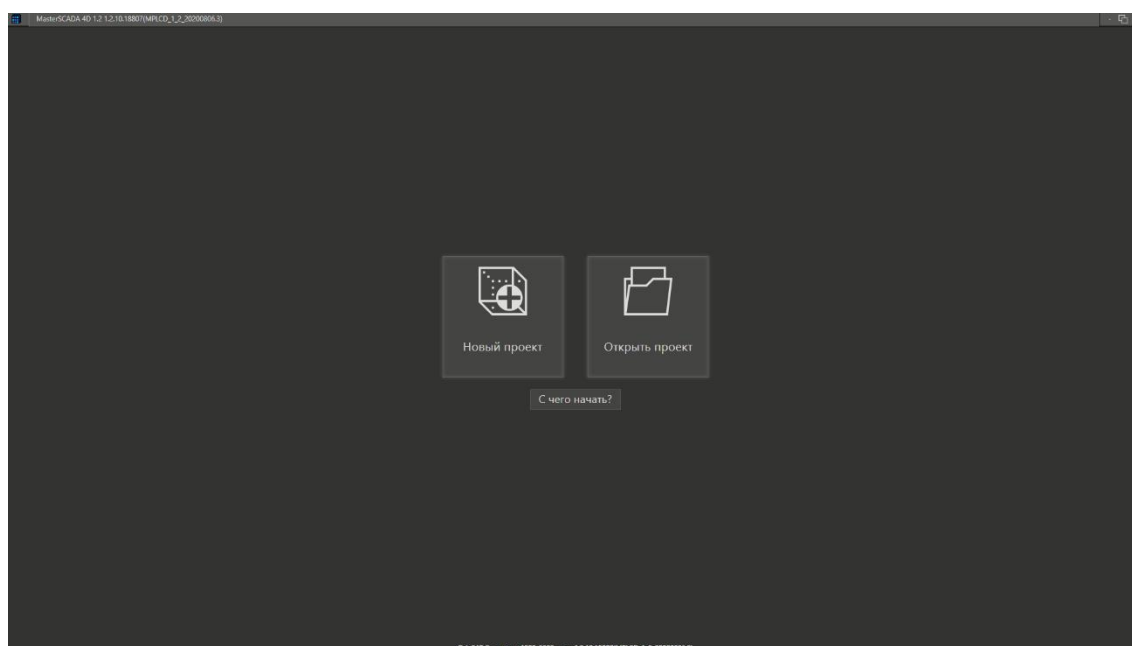


Рисунок 3 – Стартовое окно

После запуска MasterSCADA 4D будет предложено создать новый проект или открыть существующий. Создадим новый проект.

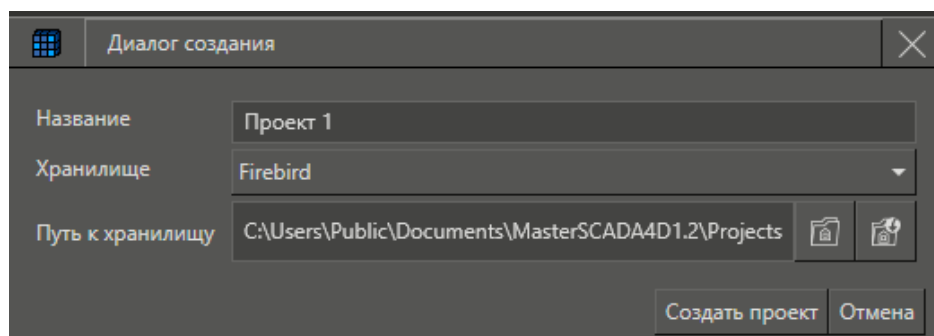


Рисунок 5 – Создание нового проекта

По умолчанию новому проекту будет присвоено имя Проект 1. Оставим его без изменения

После создания проекта откроется интерфейс среды разработки

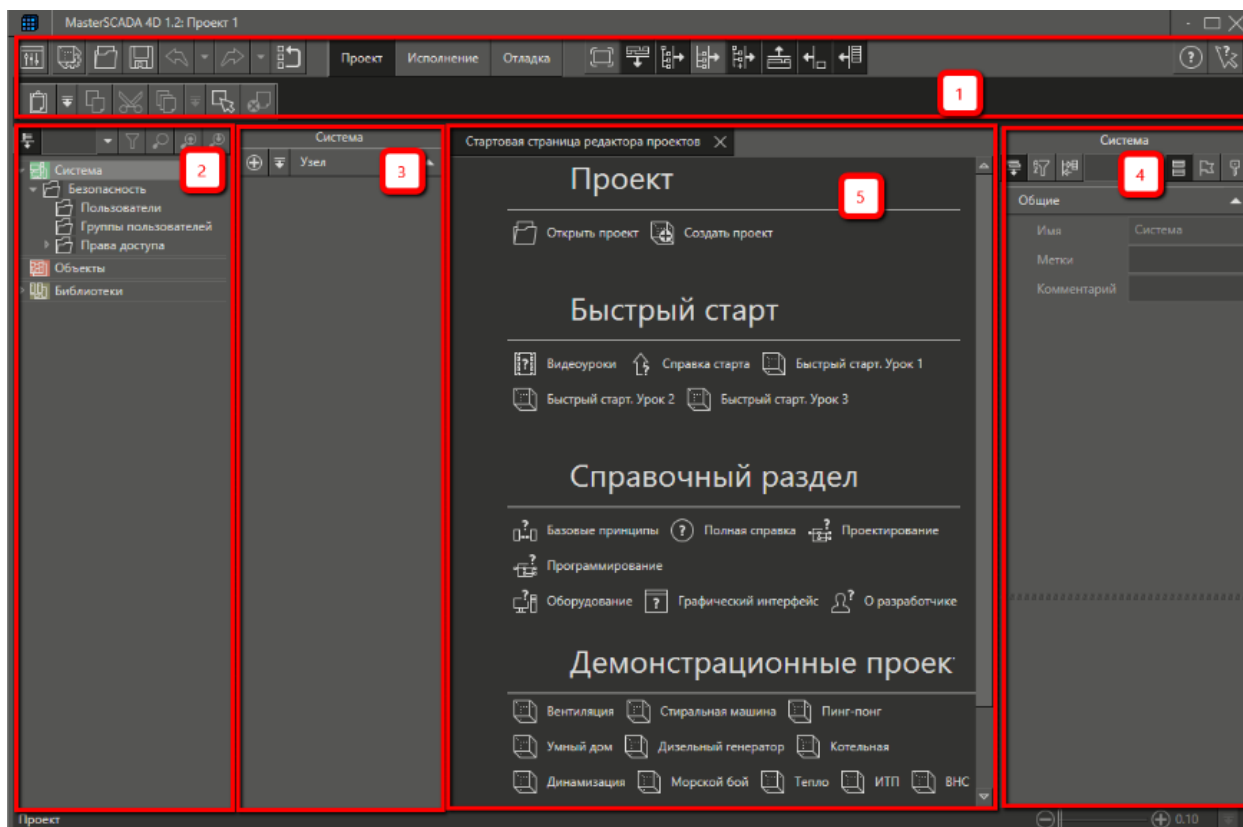


Рисунок 4 – Состав интерфейса

Интерфейс среды разработки состоит из нескольких элементов. В верхней части интерфейса располагается панель инструментов (1). Слева – дерево проекта (2) и контекстная панель (3). В правой части – панель свойств (4). В центральной части интерфейса располагается поле (панель) редакторов (5). В этом поле после запуска отображается стартовая страница с иконками для открытия демонстрационных и учебных проектов.

Дерево проектов состоит из трех частей.

Первая часть – дерево системы. В нем задается физическая конфигурация системы, т.е указываются устройства, протоколы, каналы связи.

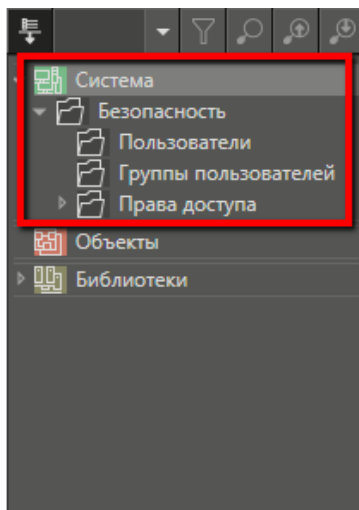


Рисунок 6 – Устройства, протоколы, каналы связи.

Вторая часть – дерево объектов. В этом дереве создаётся логическая структура проекта. По умолчанию, оно не содержит дочерних элементов.

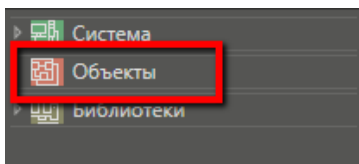


Рисунок 7 – Логическая структура проекта.

Третья часть дерева содержит библиотеки, как поставляемые со средой разработки, так и созданные пользователем.

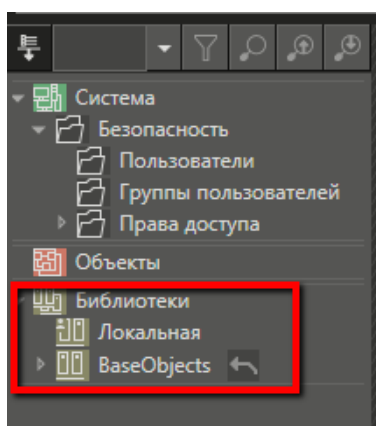


Рисунок 8 – Библиотеки.

Для загрузки проекта в ПЛК OtiLogic необходимо подключить его библиотеку. Для этого нужно в дереве библиотек нажать правой кнопкой мышки на пункт Библиотеки, выбрать в выпадающем списке пункт Подключить библиотеку и далее в выпадающем списке выбрать библиотеку KEAZ и нажать левую кнопку мыши.

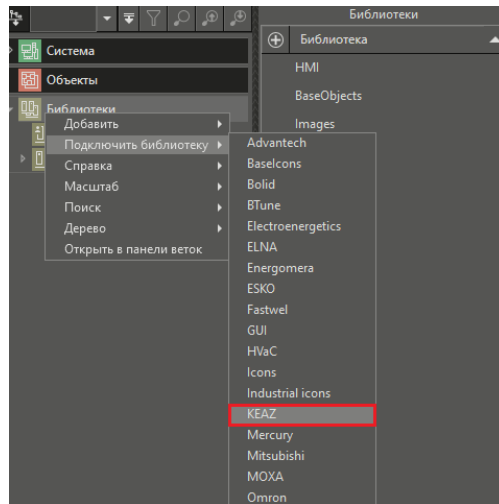


Рисунок 9 – Подключение библиотеки KEAZ.

В результате в дереве библиотек появится библиотека KEAZ с уже сконфигурированными ЦПУ и модулями ввода-вывода.

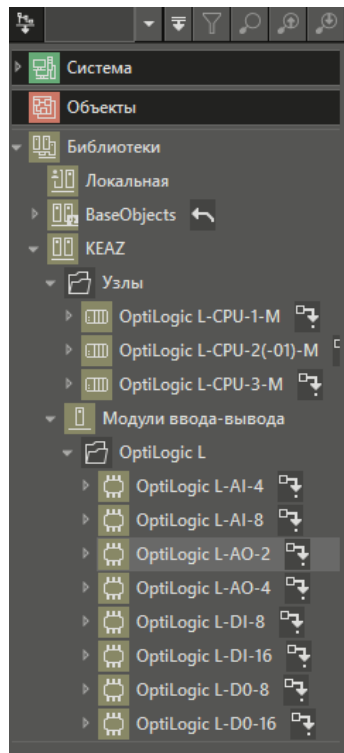


Рисунок 10 – Состав библиотеки KEAZ.

8 КОНФИГУРИРОВАНИЕ ДЕРЕВА СИСТЕМЫ

Построение проекта начинается с конфигурирования Древа системы. Для этого нужно нажать правой кнопкой мыши на любое место Древа системы и из Контекстного меню выбрать устройство, на котором будет исполняться проект. В нашем случае это OptiLogic L-CPU-1-M

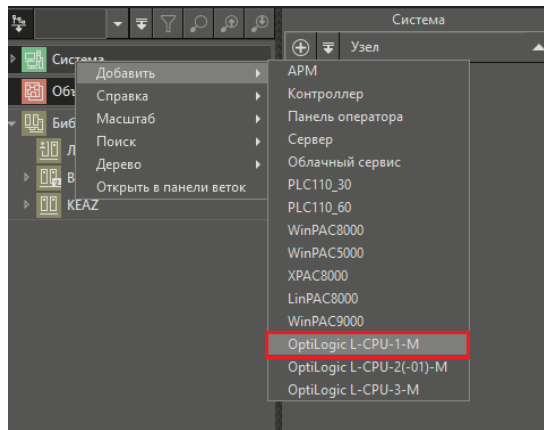


Рисунок 11 – Выбор модуля для добавления в проект.

В дерево добавится ПЛК OptiLogic L-CPU-1-M, содержащий свойственные для него элементы.

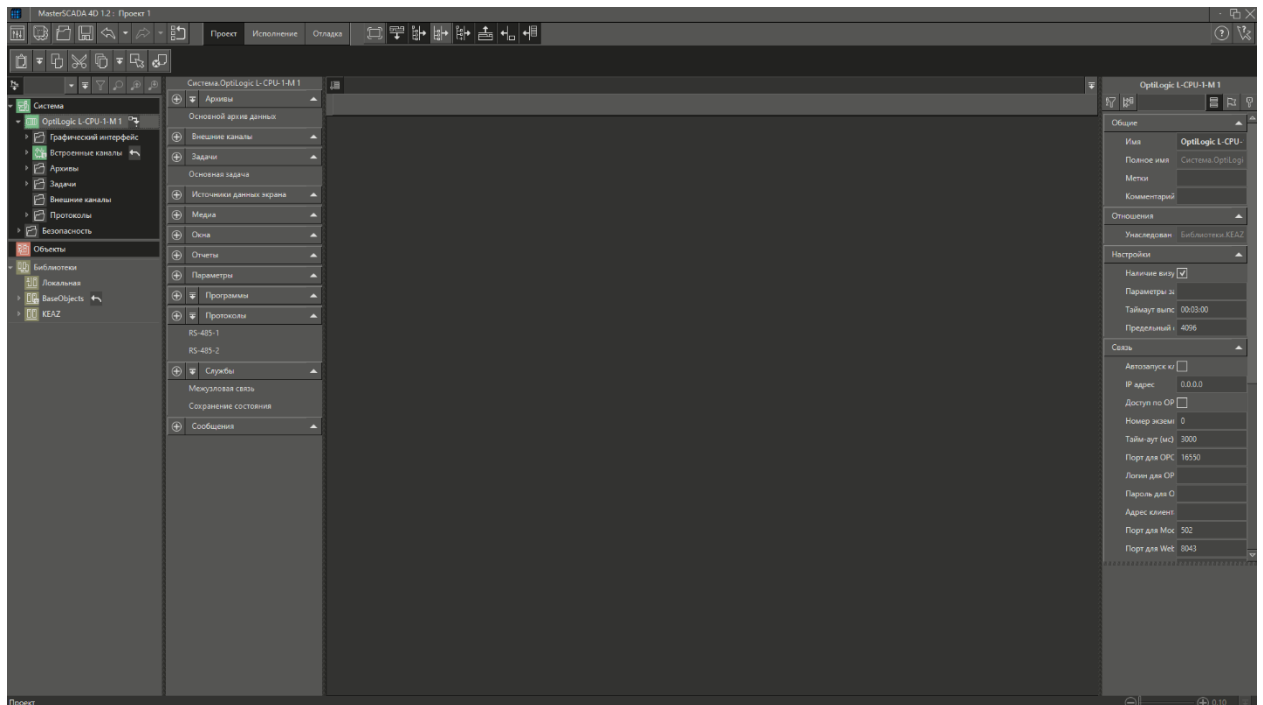


Рисунок 12 – Внешний вид окна для работы с модулем.

Во вкладке Связь необходимо указать IP-адрес ПЛК, по умолчанию это 172.16.0.125. Для того чтобы узнать IP-адрес вашего ПЛК обратитесь к документации на ПЛК.

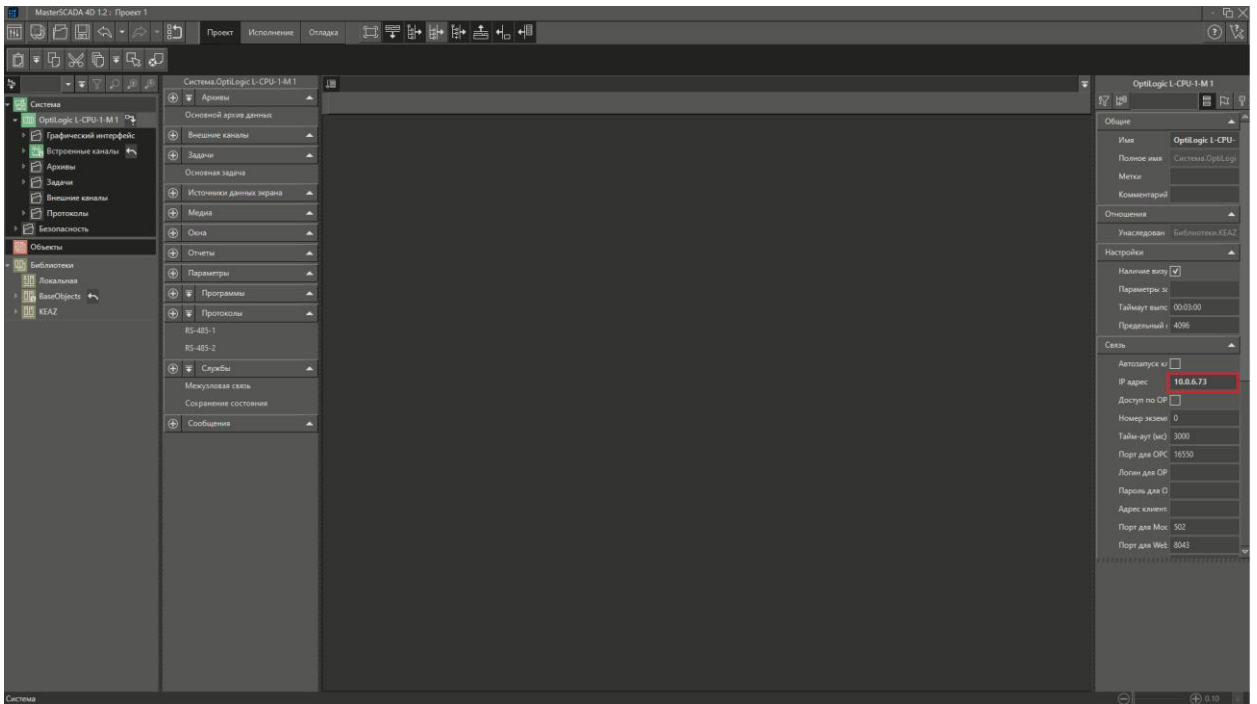


Рисунок 13 – Поле IP-адреса ПЛК.

В рамках поставленной задачи требуется один дискретный выход для управления устройством. ПЛК OptiLogic L-CPU-1-M имеет встроенные каналы ввода-вывода. Нам потребуется один канал дискретного выхода.

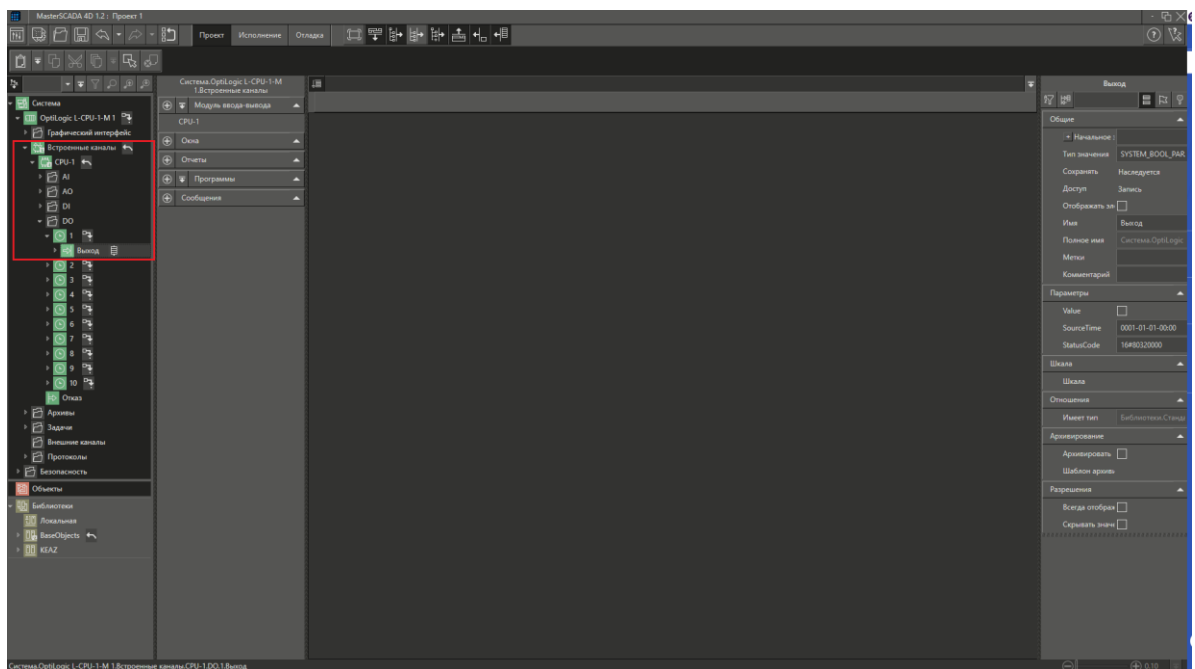


Рисунок 14 – Добавление дискретного канала ПЛК.

На этом конфигурирование физической структуры закончено.

9 СОЗДАНИЕ ЛОГИЧЕСКОЙ ЧАСТИ ПРОЕКТА

Вызовем контекстное меню дерева объектов, нажав правую кнопку мыши, и выполним пункт Добавить-Объект.

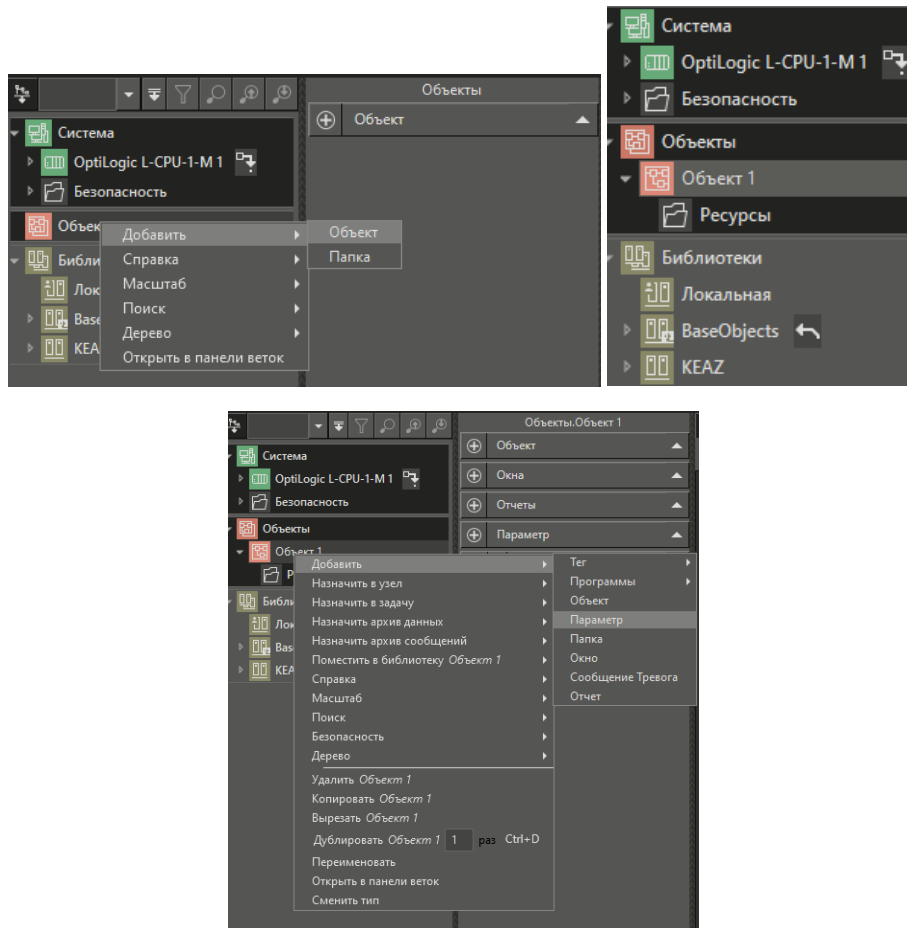


Рисунок 15 – Добавление параметра.

В объект можно добавлять программы для обработки данных, графические окна и другие элементы.

В рамках поставленной задачи у нас будет один управляющий сигнал. Нажмем правой кнопкой мыши на объект, и из контекстного меню выберем Добавить-Параметр

В отрывшемся окне свойств зададим ему имя – Управляющий Выход и укажем тип параметра BOOL (логический)

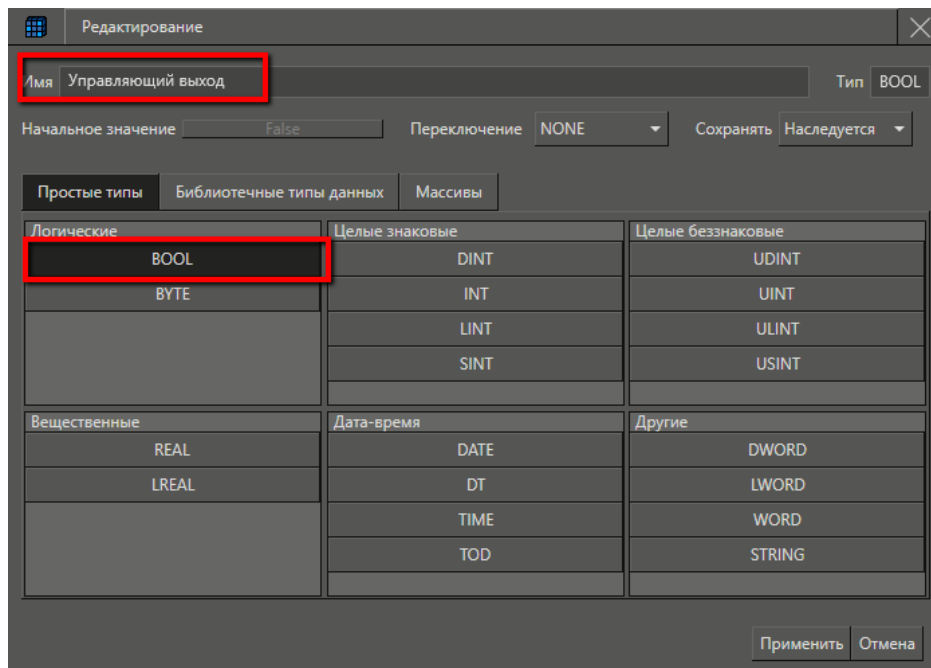


Рисунок 16 – Заведение параметра.

В результате получим:

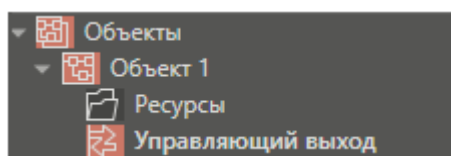


Рисунок 17 – Параметр управляющей программы.

Далее необходимо создать управляющую программу. Для этого воспользуемся контекстной панелью элемента дерева Объект 1. На рисунке ниже показано, что данная панель дает возможность добавить программу на одном из четырех языков стандарта МЭК 61131-3 и языка C#. Выберем программу на языке функциональных блоков – FBD.

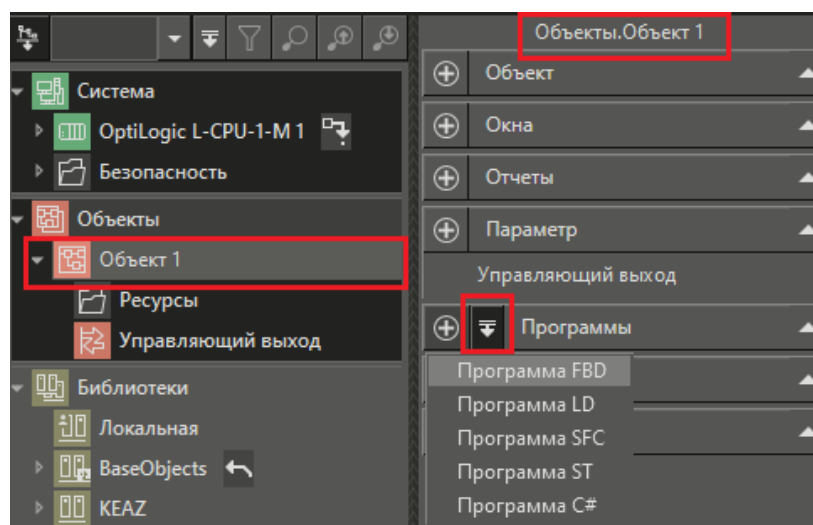


Рисунок 18 – Выбор редактора управляющей программы.

При этом откроется редактор этого языка

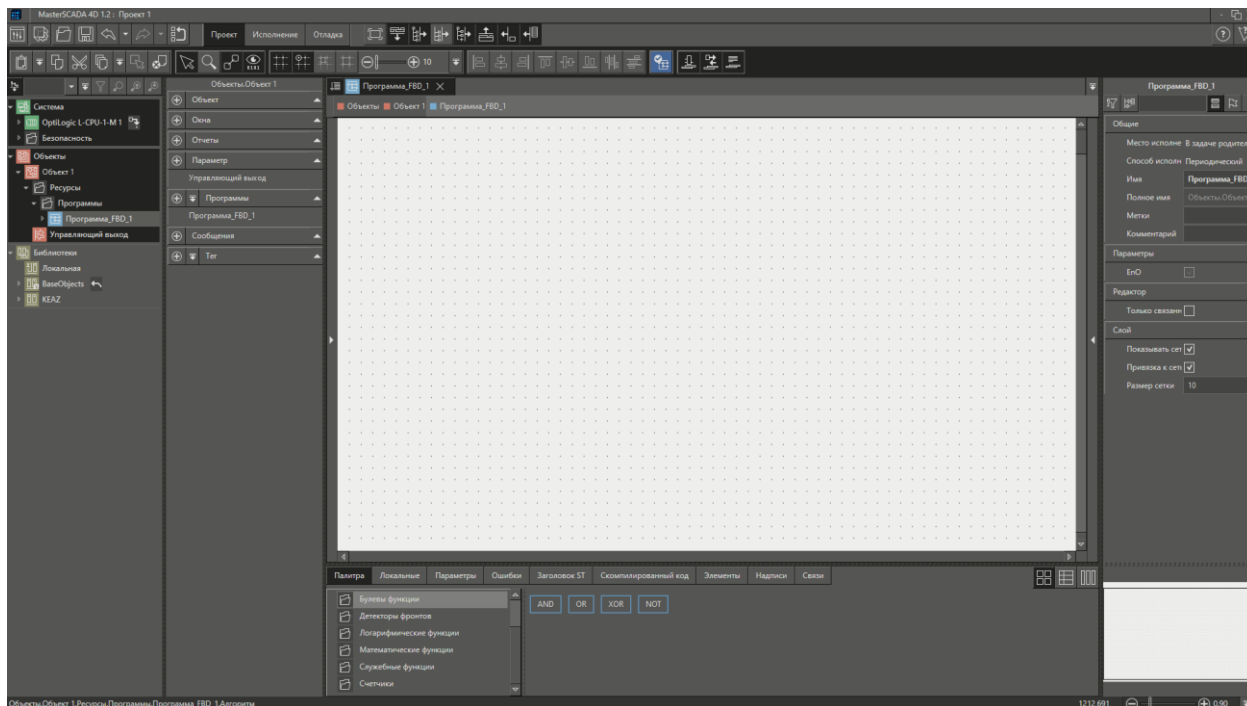


Рисунок 19 – Редактор FBD.

Слева и справа от поля редактирования, по аналогии с электрическими схемами, расположены клеммники. На клеммнике слева находятся входные сигналы, а на клеммнике справа – выходные. Для того чтобы развернуть или свернуть клеммники, необходимо нажать на стрелки, расположенные по краям поля редактирования.

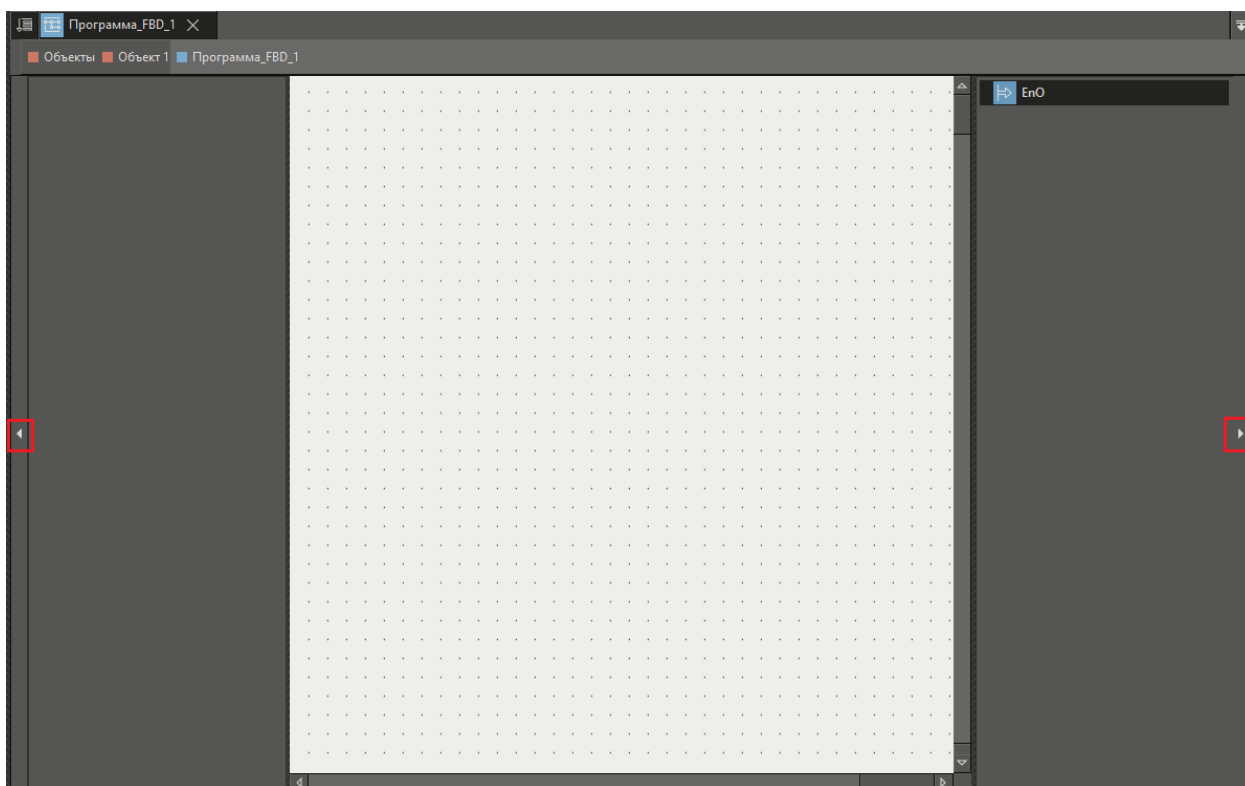


Рисунок 20 – «Клеммники» редактора FBD.

Далее необходимо создать в программе два входных параметра: Пуск и Стоп. Для этого нажмем правой кнопкой мыши в любое место

входного клеммника, и из контекстного меню выберем пункт **Добавить-Параметр**.

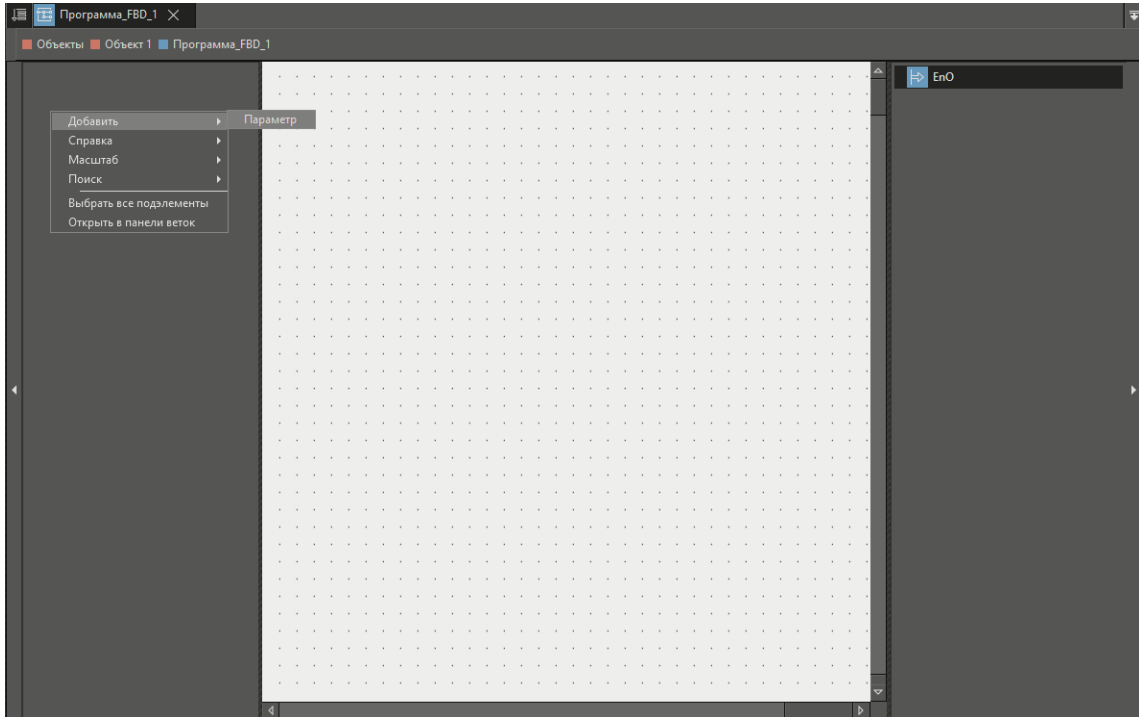


Рисунок 21 – Создание входных параметров.

В открывшемся окне свойств параметра изменим его имя на **Пуск**, а тип – на **BOOL**.

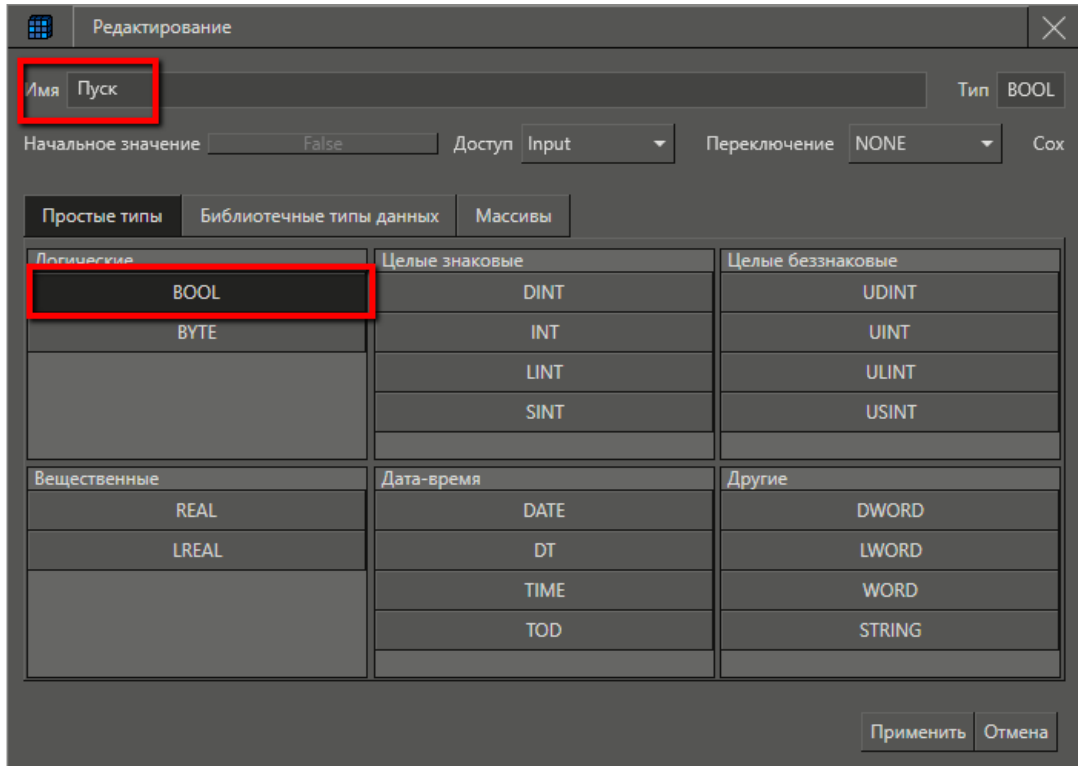


Рисунок 22 – Заведение параметра «Пуск».

Аналогично создадим параметр Стоп.

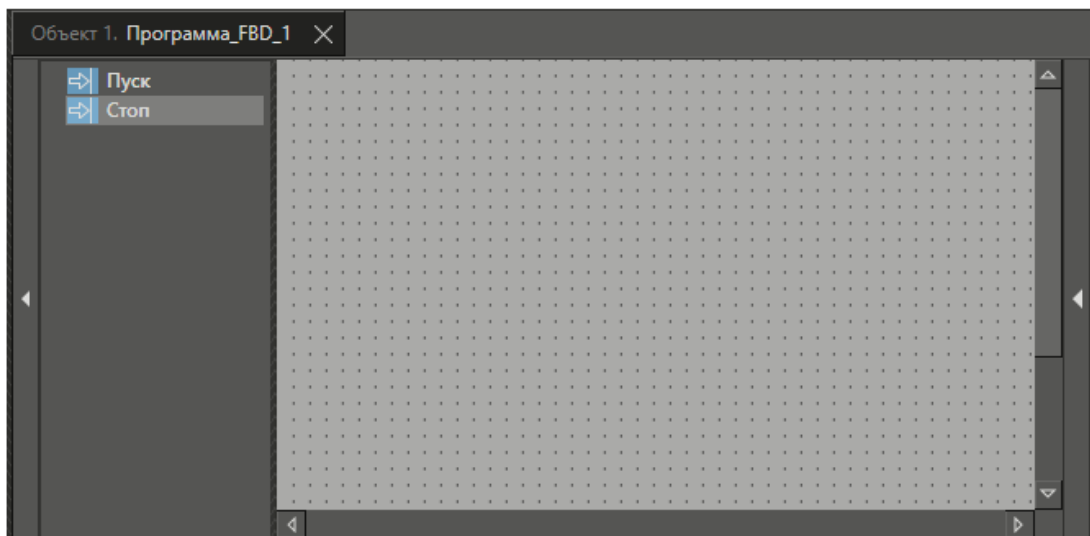
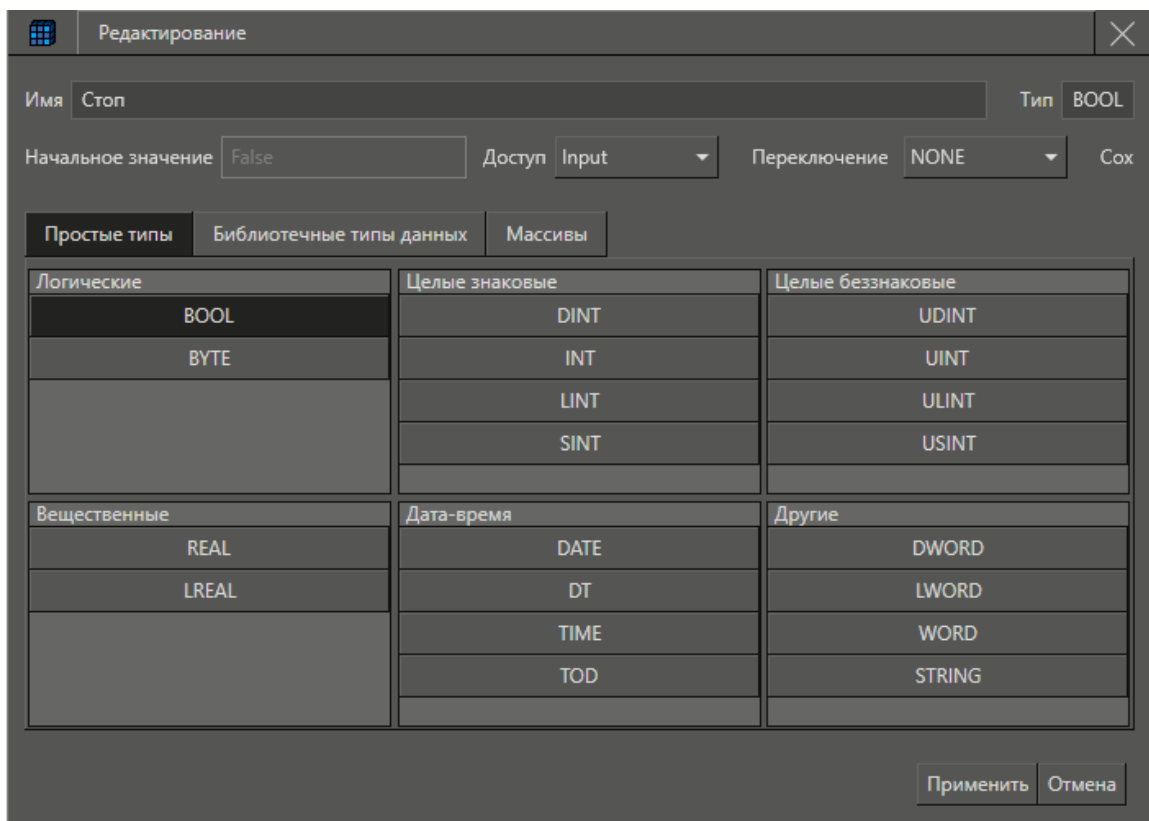


Рисунок 23 – Заведение параметра «Стоп».

Для формирования логики включения используем обычный RS-триггер. Для того чтобы вставить на схему функциональный блок, достаточно просто перетащить его из Палитры в поле редактора. В Палитре найдем раздел Триггеры и перетащим элемент RS на схему.

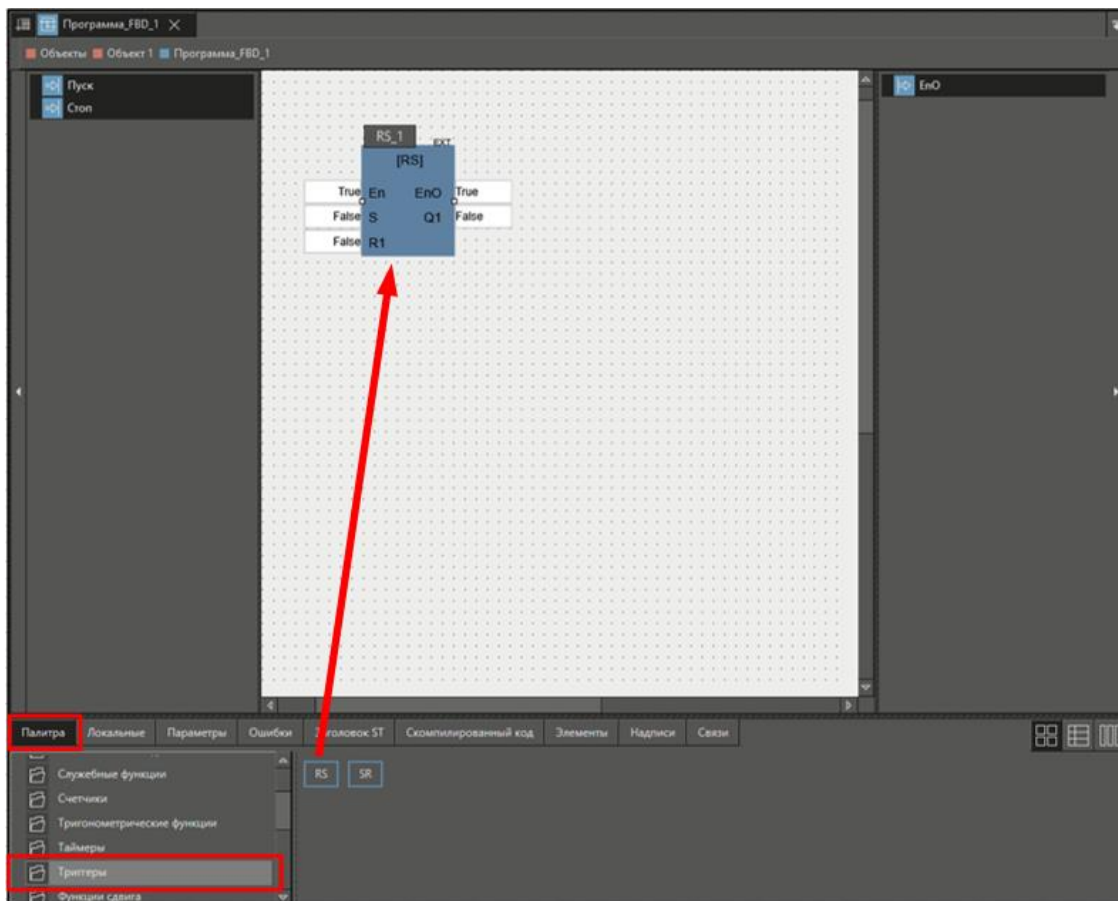


Рисунок 24 – Добавление триггера.

Теперь необходимо подключить входы программы к входам RS-триггера. Для этого нужно нажать левой кнопкой мыши над названием параметра (например, Пуск), и, удерживая кнопку мыши нажатой, перенести указатель на тот параметр, с которым Вы хотите его связать (S), и отпустить над ним кнопку мыши.

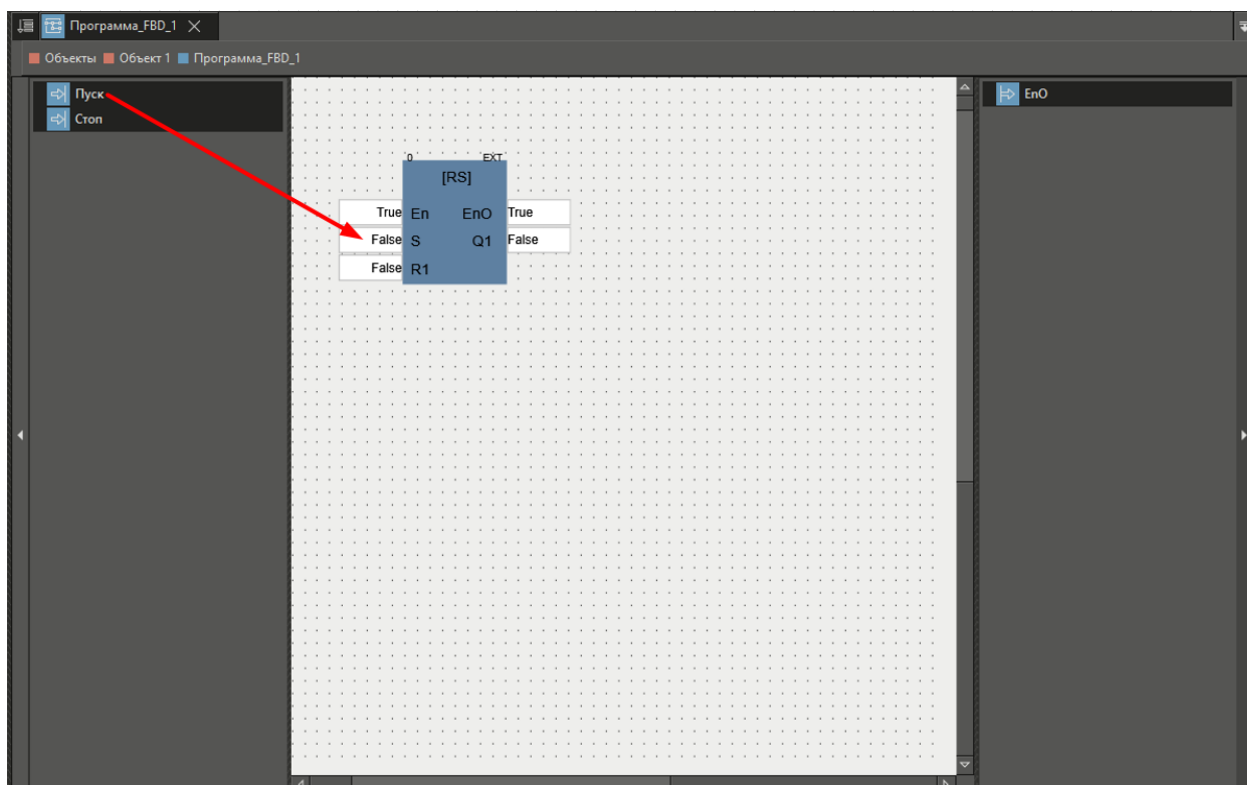


Рисунок 25 – Связь с параметром «Старт».

Аналогичным способом необходимо связать вход параметра Стоп с входом триггера R1:

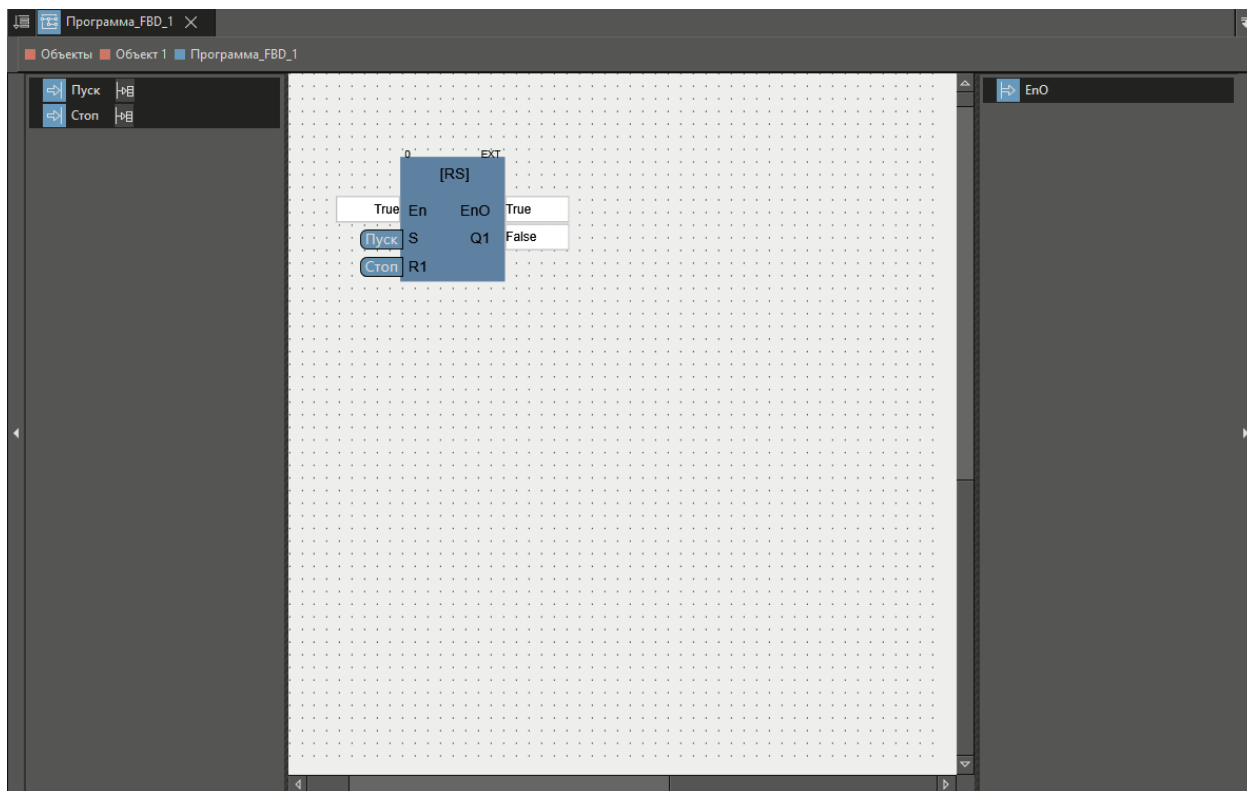


Рисунок 26 – Связь с параметром «Стоп».

Далее необходимо создать выходной параметр. Для этого перетащим параметр объекта Управляющий Выход на выходной клеммник программы, и в результате будет создан выходной параметр программы (по умолчанию – одноимённый), а также связь с ним.

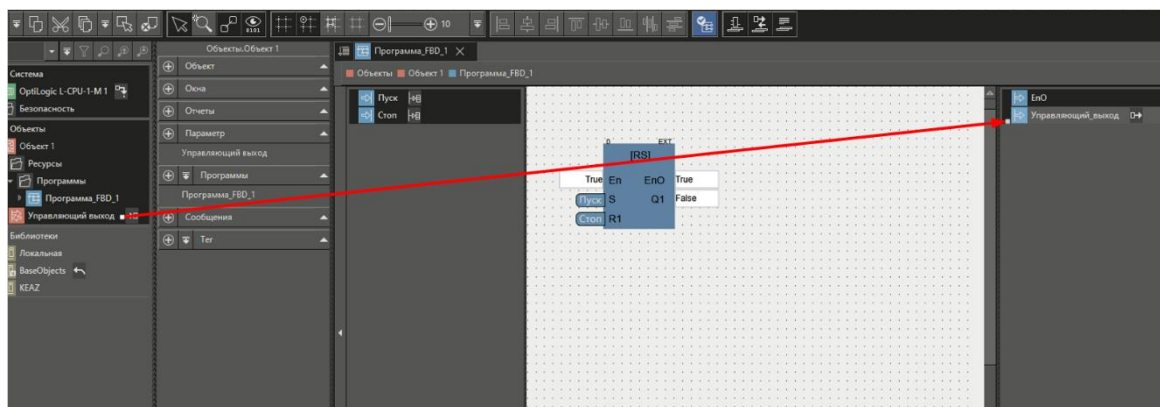


Рисунок 27 – Выходной параметр.

Свяжем выход триггера Q1 с выходным параметром Управляющий Выход.

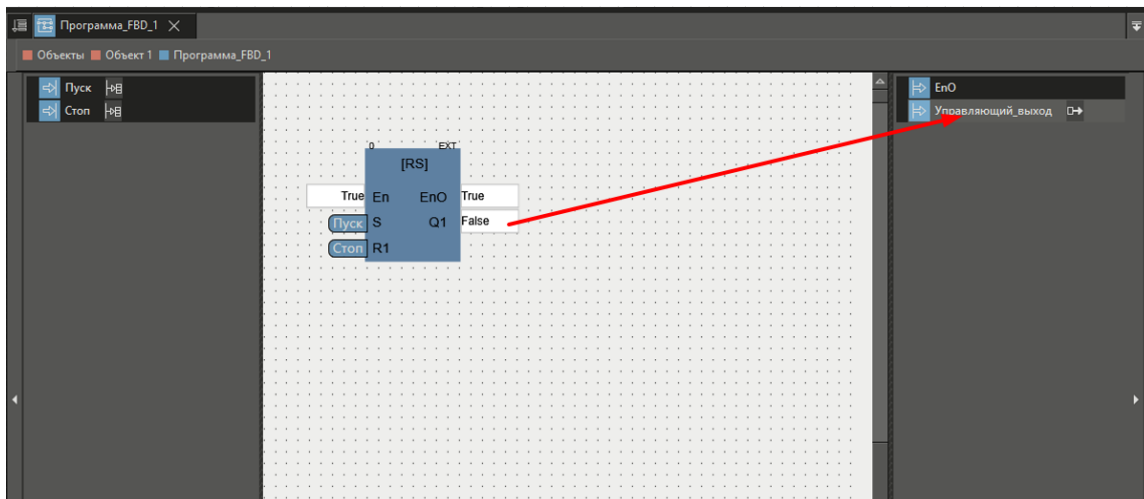


Рисунок 28 – Управляющий выход.

Таким образом, мы сформировали выходной тракт от выхода триггера Q1 к параметру Управляющий Выход.

На этом формирование логики управления закончено. На всякий случай, нажмём кнопку «Проверить» и убедимся, что компиляция пройдет успешно.

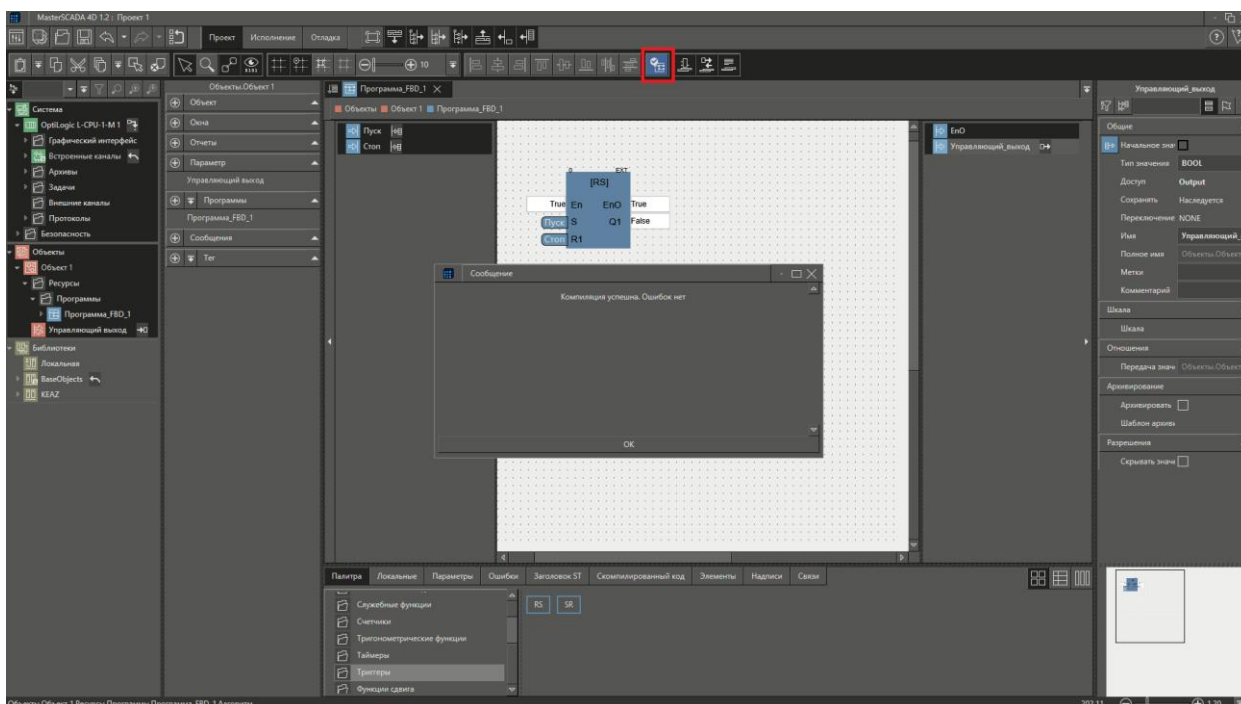


Рисунок 29 – Проверка управляющей программы.

Следующий этап – создание визуализации.

10 СОЗДАНИЕ ОКНА УПРАВЛЕНИЯ

Для того чтобы добавить в проект окно визуализации необходимо выбрать в контекстном меню объекта пункта Добавить-Окно

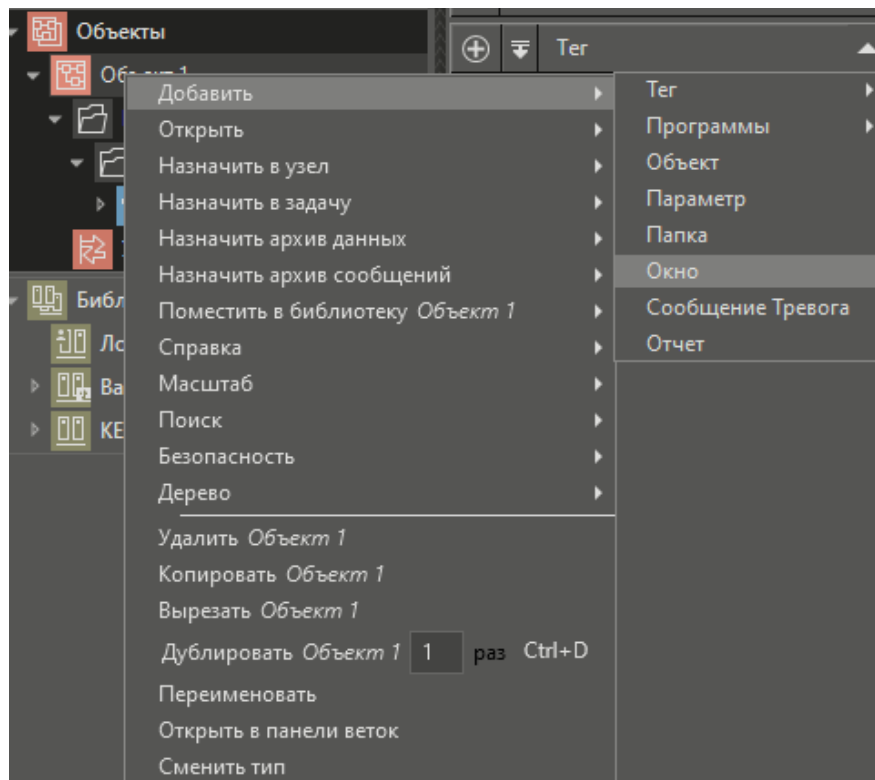


Рисунок 30 – Создание окна управления.

При этом у объекта добавится окно, которое сразу откроется в поле редактора графического интерфейса

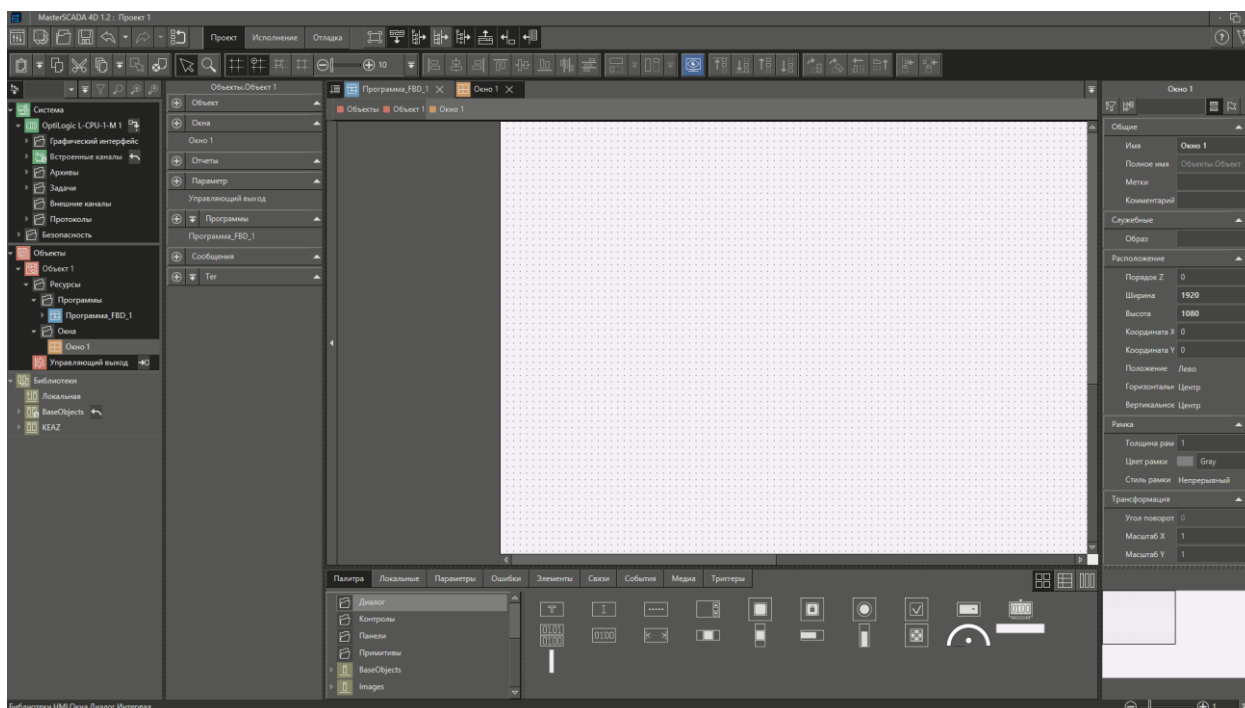


Рисунок 31 – Редактор графического интерфейса.

Наша задача заключается в том, чтобы разместить в этом окне две кнопки – Пуск и Стоп, нажатие которых будет передаваться на соответствующие входные параметры программы, а также создать индикацию состояния Управляющего выхода.

Из группы Параметры созданной программы необходимо перетащить правой кнопкой мыши параметр Пуск в окно. Из открывшегося контекстного меню выберем пункт Вставить. Кнопка.

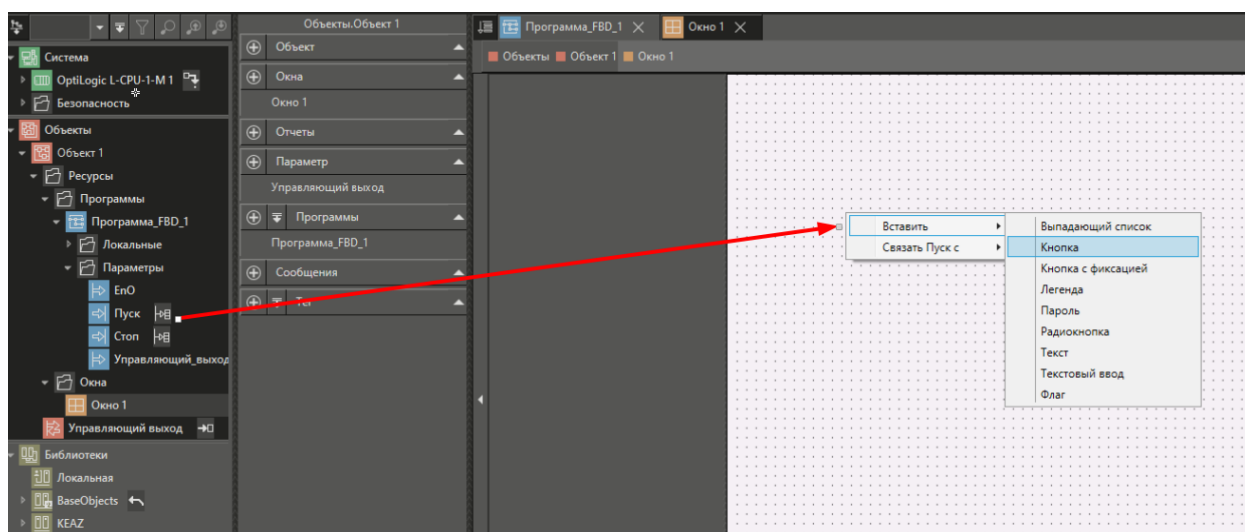


Рисунок 32 – Добавление Кнопки.

В окно будет вставлена кнопка с уже установленными связями между параметром Пуск и свойством кнопки Нажата. По нажатию кнопки параметр Пуск будет устанавливаться в TRUE, а при отпусканию кнопки этот же параметр будет установлен в False. Таким образом, параметр Пуск будет отражать состояние кнопки. Текст кнопки будет совпадать с текстом параметра, при желании его можно изменить в соответствующем пункте меню.

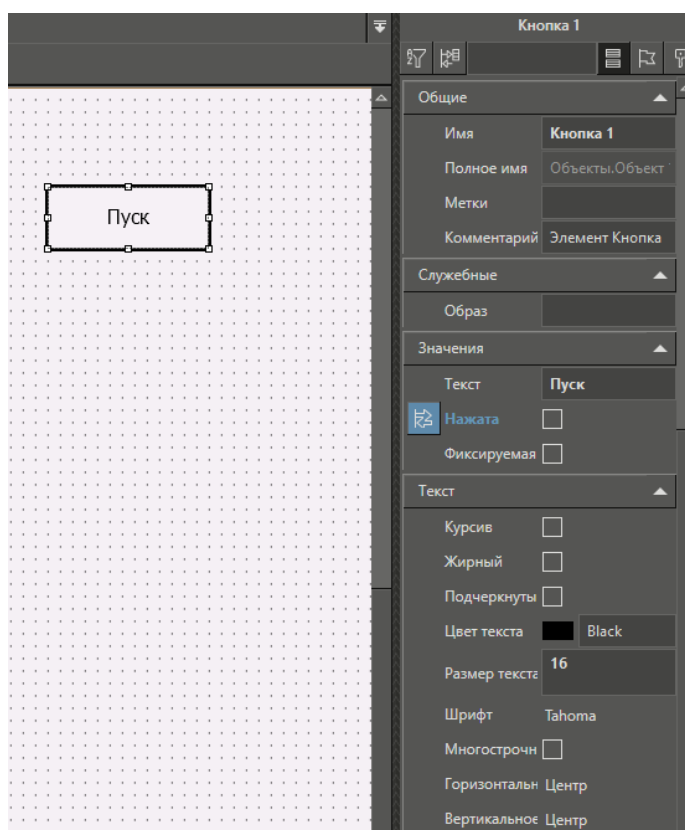


Рисунок 33 – Текст кнопки.

Аналогичные действия произведем с параметром Стоп.

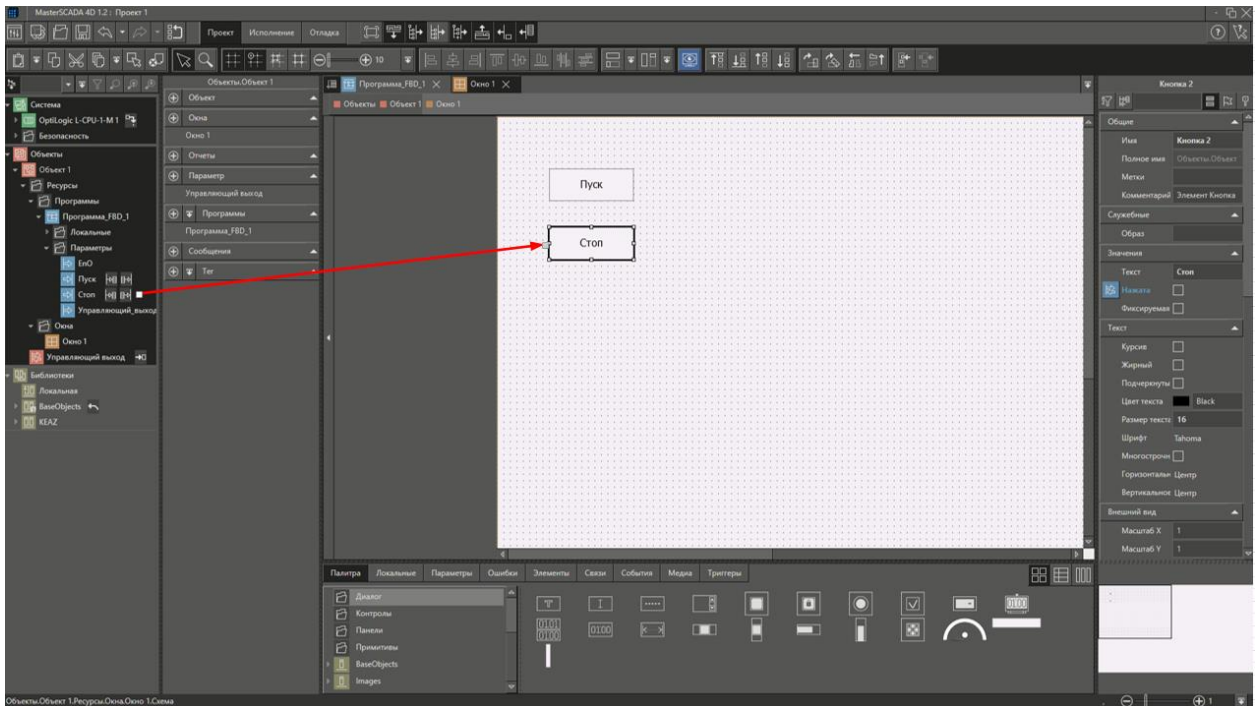


Рисунок 34 – Кнопка Стоп.

Далее необходимо реализовать отображение состояния Управляющего входа. Для этого нужно перейти в Легенду графического редактора, открыть вкладку Палитра, выбрать категорию Прimitives и перетащить левой кнопкой мыши элемент Овал в окно.

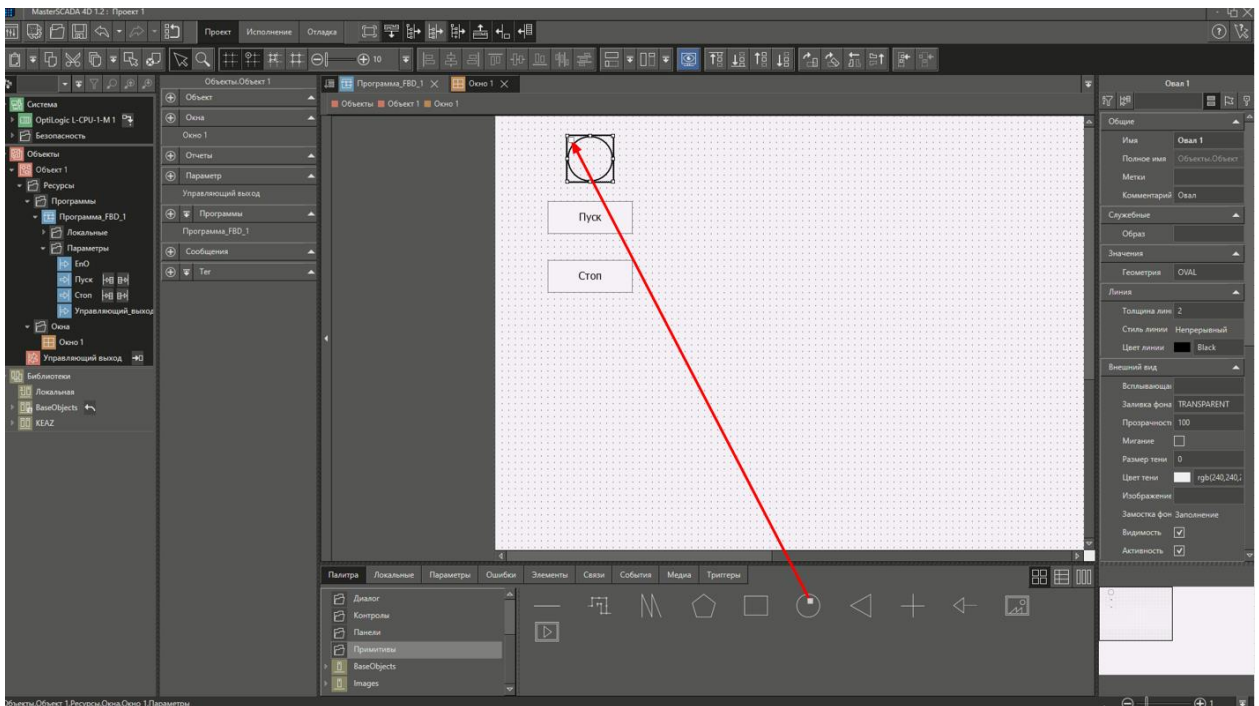


Рисунок 35 – Примитив для индикации.

Цвет овала должен отображать состояние входного сигнала: включен или выключен.

Для этого необходимо параметр объекта Управляющий выход перетащить мышкой на свойство овала Заливка фона.

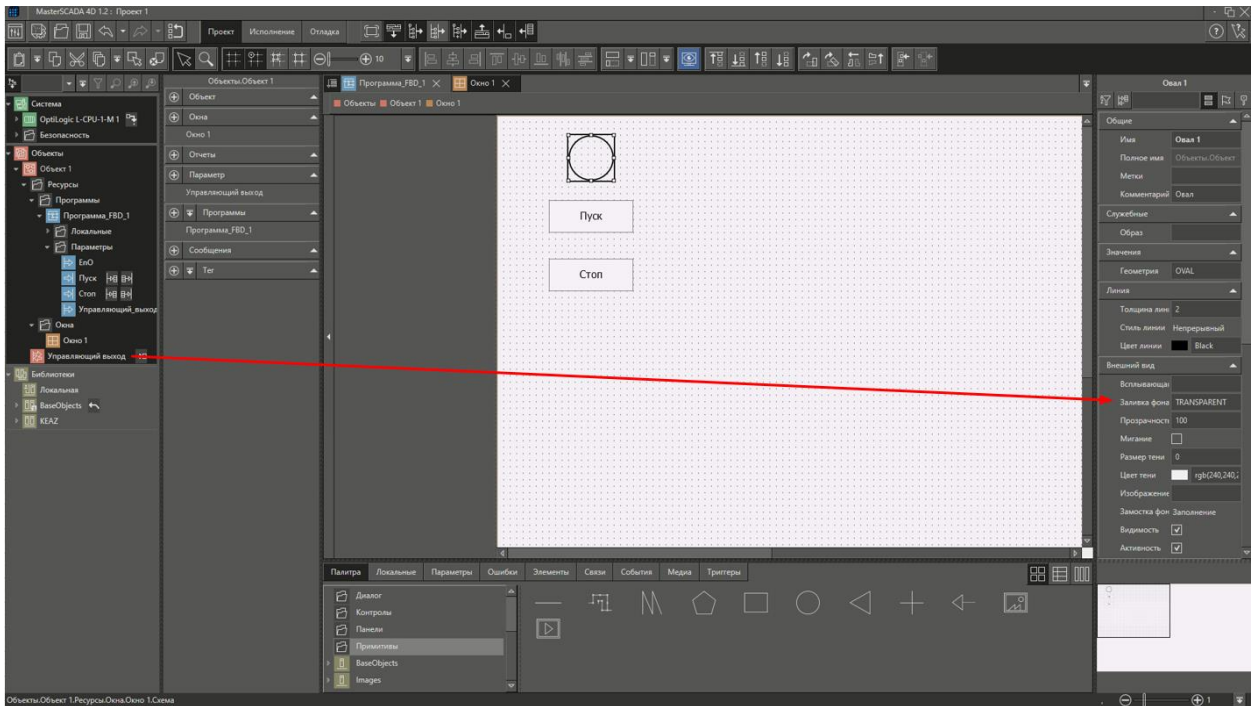




Рисунок 36 – Заливка фона примитива.

Справа от свойства появится значок связи . При нажатии на него левой кнопкой мыши откроется дополнительная информация о созданной связи. Нажмём на значок 

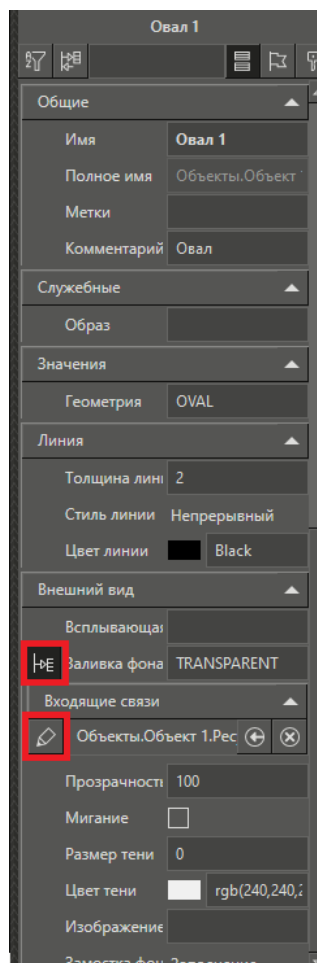


Рисунок 37 – Заливка фона примитива.

Откроется диалоговое окно:

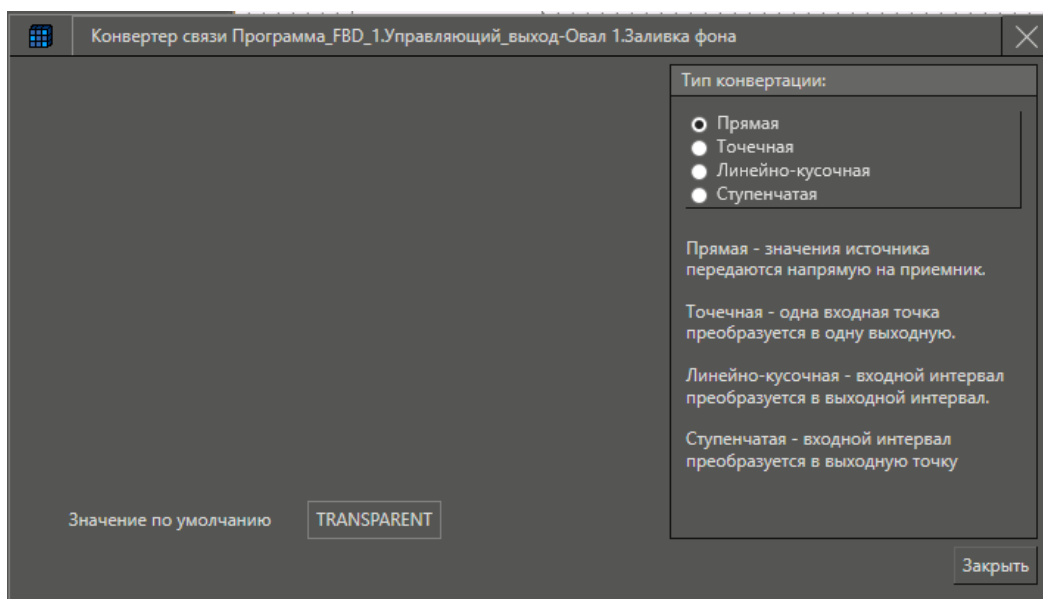


Рисунок 38 – Параметры связи с примитивом.

На любую связь в MasterSCADA 4D можно назначить конвертацию. Выберем точечный тип конвертации. Назначим так, чтобы значению False соответствовал белый цвет (выберем из стандартных цветов WHITE). Для того чтобы кнопки выбора стали доступны, необходимо нажать правой кнопкой мыши напротив нужного значения Источника в поле Приемник.

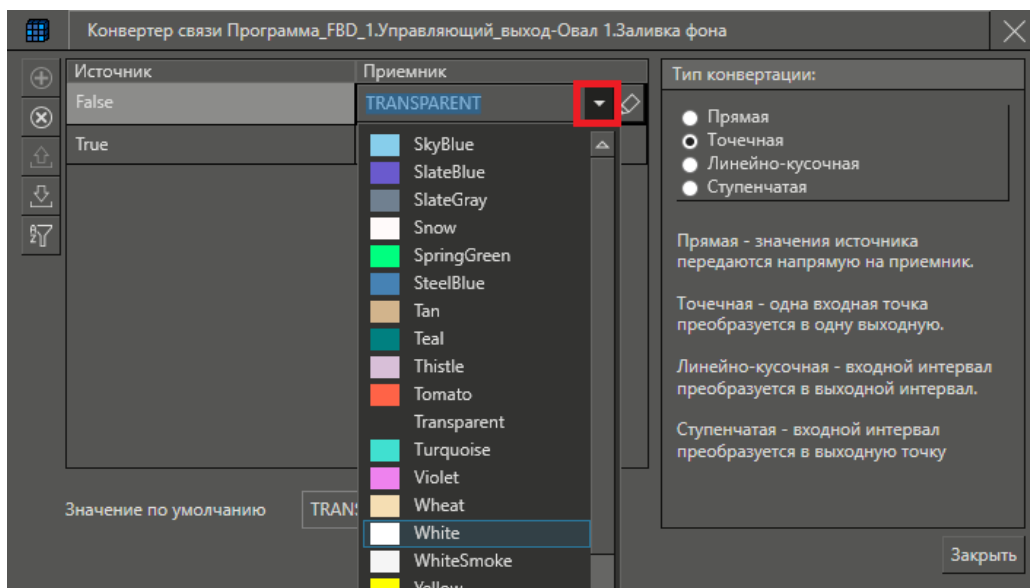


Рисунок 39 – Задание цвета значению False.

Для значения TRUE выберем зеленый цвет из палитры.

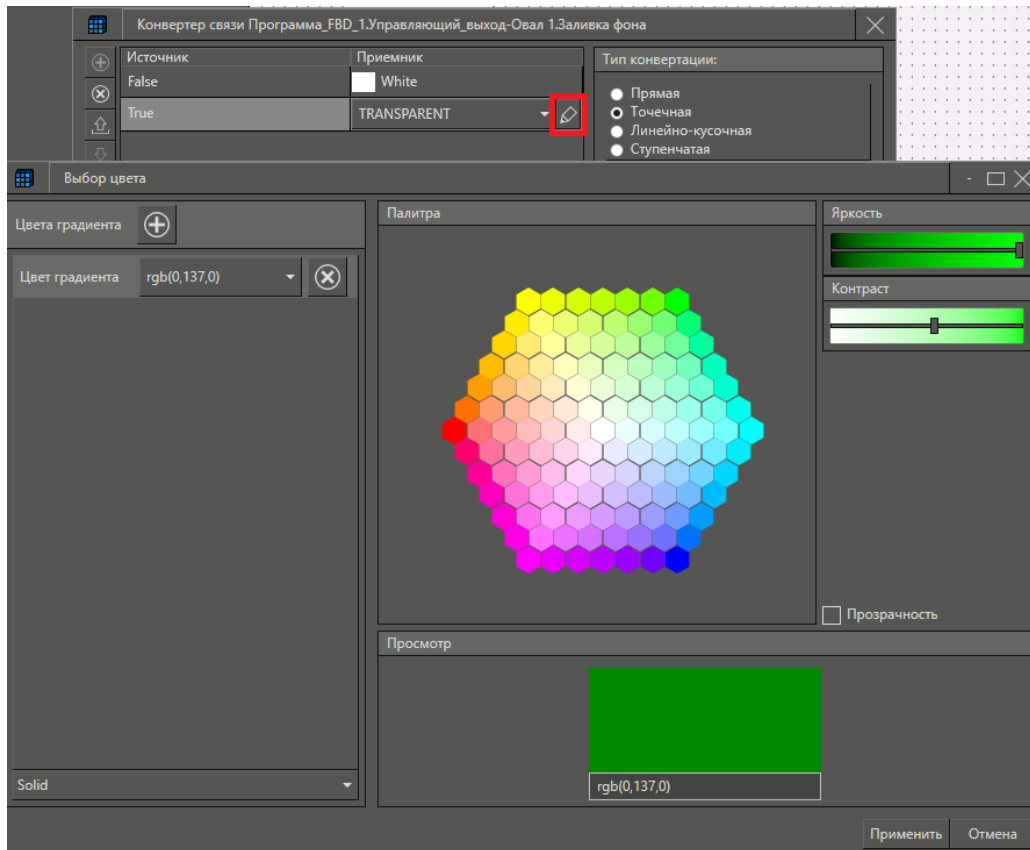


Рисунок 40 – Задание цвета значению TRUE.

После этого закроем окно конвертации.

Теперь добавим изображение насоса. Для этого выберем любой насос из библиотеки насосов, входящих в библиотеку Общая, и перетащим его в окно.

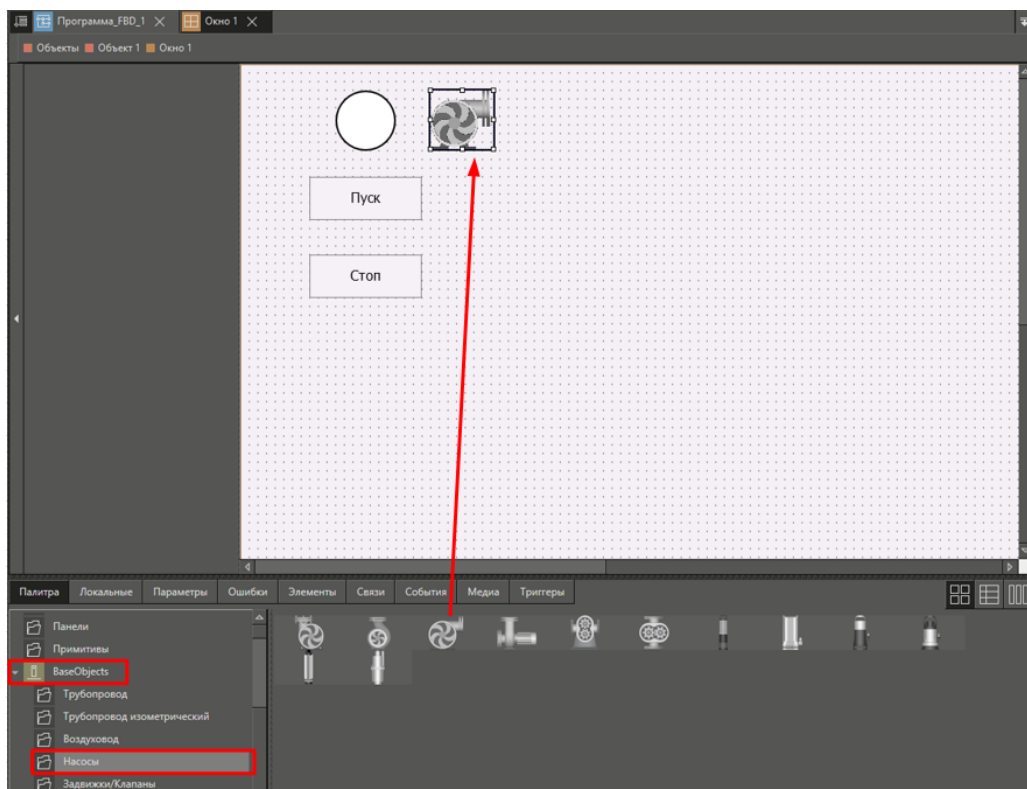


Рисунок 41 – Добавление стандартного примитива.

Наконец, свяжем параметр объекта Управляющий выход со свойствами насоса Вращение и Работа.

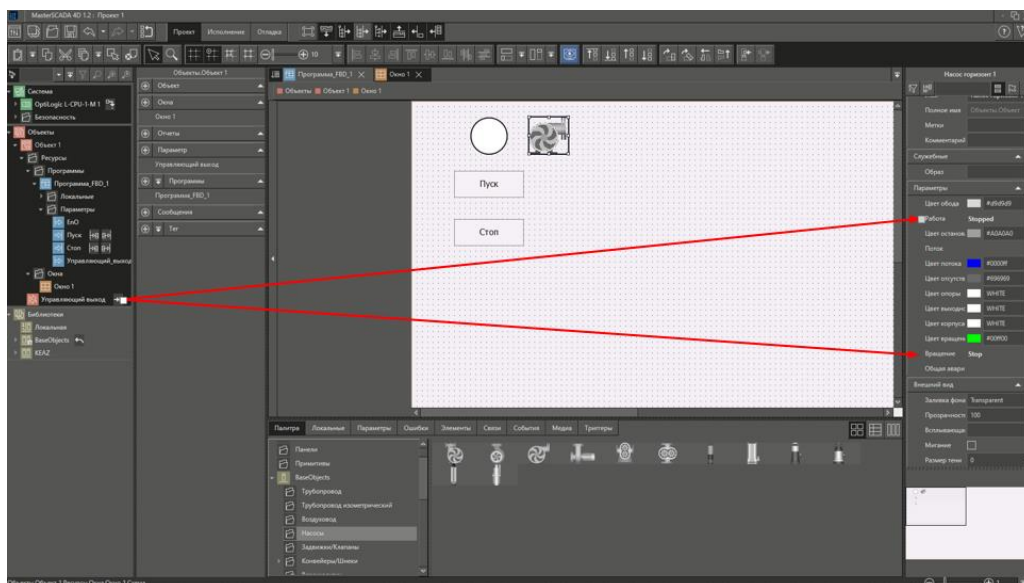


Рисунок 42 – Свойства стандартного примитива.

На этом формирование визуализации завершено.

Теперь назначим объект для исполнения в узле контроллера. Это необходимо сделать потому, что узлов в проекте может быть много, и любой объект может быть назначен для исполнения в любом из них. Для этого нажмем правой кнопкой мыши на объект, и из контекстного меню выберем Назначить в узел, а затем выберем название узла. В нашем случае узел это OptiLogic L-CPU-1-M1

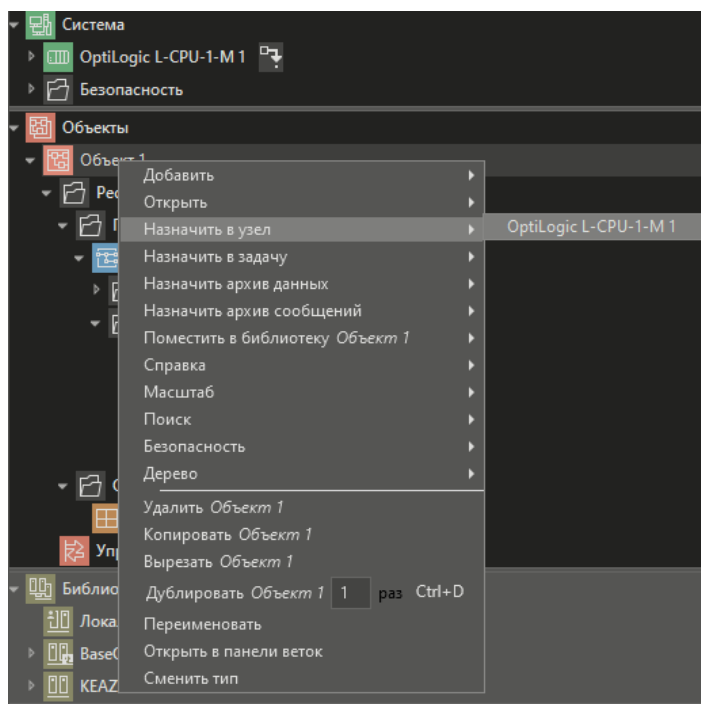


Рисунок 43 – Свойства стандартного примитива.

Следующим шагом нужно назначить стартовое окно. В нашем проекте пока есть только одно окно. Но таких окон может быть много, и необходимо указать, какое из них будет стартовым. Для этого нажмем правой кнопкой мыши на это окно, из контекстного меню выберем

Назначить стартовым окном, и укажем, где именно стартовое окно будет исполняться. В нашем случае есть только один возможный вариант.

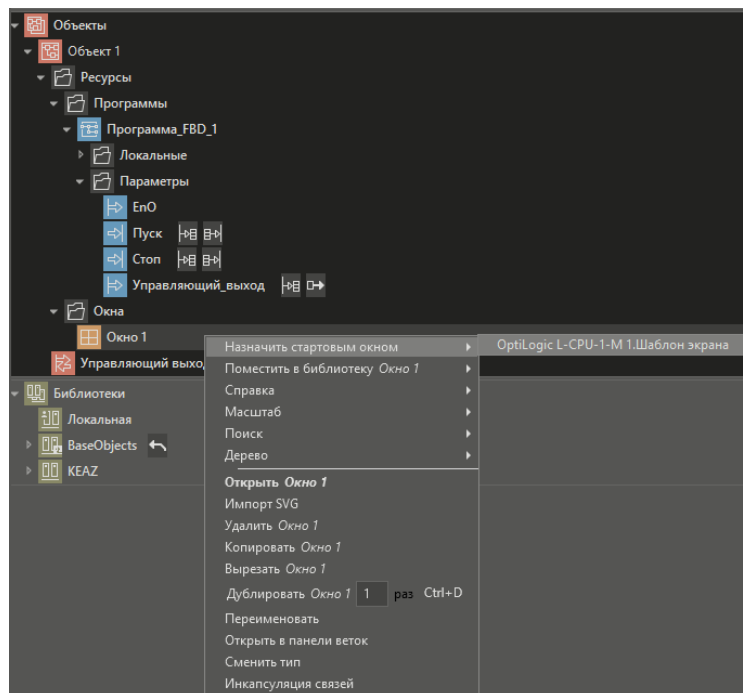


Рисунок 44 - Назначение окна стартовым.

Итак, получился функционально законченный объект, содержащий программу управления, средства визуализации и имеющий один параметр для связи этого объекта с внешним миром. В нашем случае нам требуется при помощи этого параметра управлять дискретным выходом модуля. Для этого перетащим левой кнопкой мыши параметр объекта Управляющий выход на дискретный выход DO.1 встроенного канала OptiLogic L-CPU-1-M1. При этом создастся связь, передающая значение из параметра Управляющий выход в канал дискретного выхода.

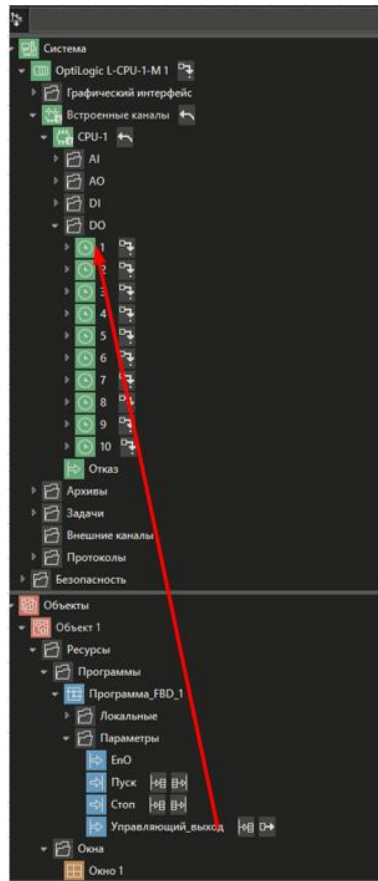


Рисунок 45 – Связь параметра с дискретным выходом.

11 ЗАПУСК ПРОЕКТА

Сначала перейдём на вкладку *Исполнение* панели инструментов и нажмём кнопку *Проверить*:

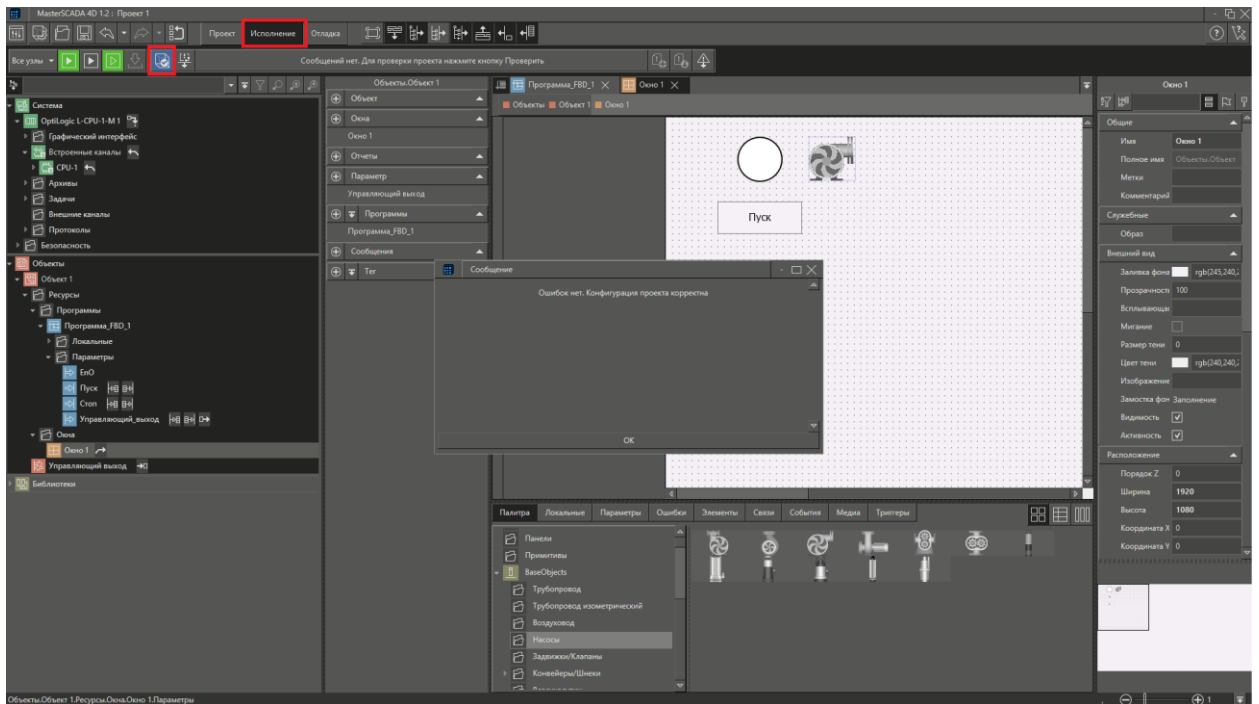


Рисунок 46 – Проверка исполнительной программы.

Затем, убедившись в отсутствии ошибок проекта, запустим проект на исполнение в режиме эмуляции. При этом загрузка в контроллер производиться не будет, а исполнительная система будет эмулироваться на том же компьютере, на котором производилась разработка проекта.

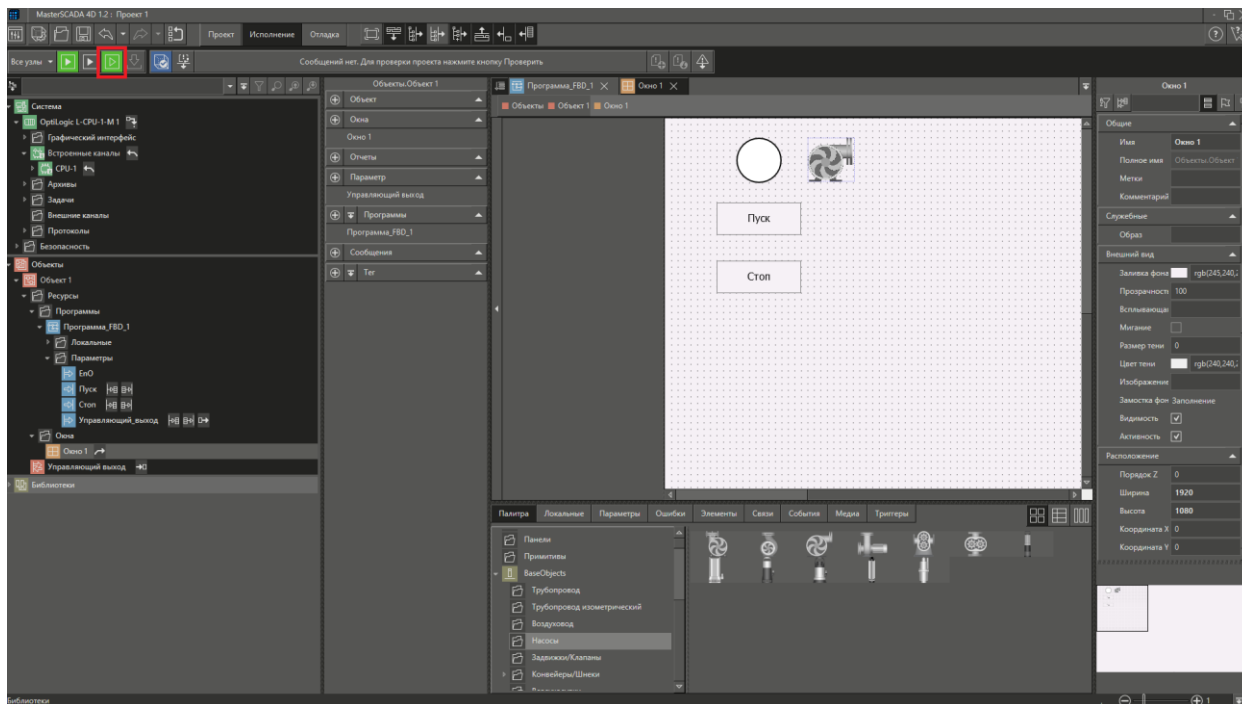


Рисунок 47 – Эмуляция исполнительной программы.

Если режим исполнения запускается впервые, то появится окно, предлагающее для отображения в режиме исполнения на выбор либо клиент визуализации, разработанный компанией "ИнСАТ", либо браузер, назначенный на компьютере по умолчанию.

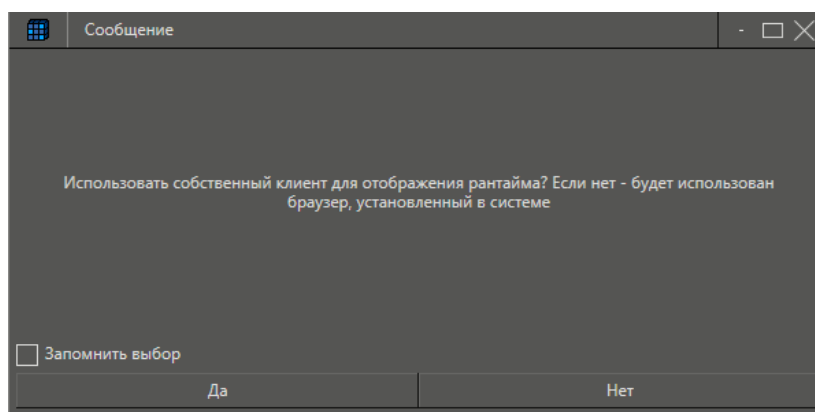



Рисунок 48 – Выбор варианта отображения рантайма.

В нашем случае выбор может быть любым.

После успешного запуска проекта на исполнение, рядом с названием узла появится знак , и откроется встроенный клиент визуализации (или браузер) с созданной мнемосхемой.

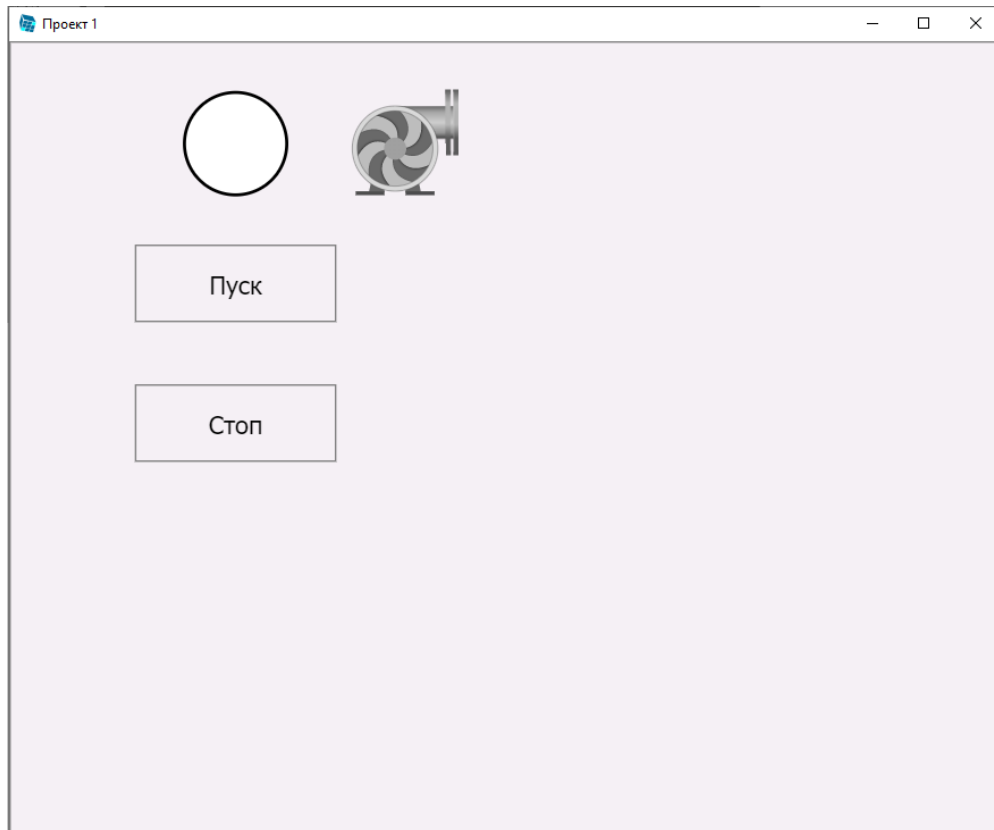


Рисунок 49 – Тестовая мнемосхема.

В среде разработки рядом с именами параметров появятся их текущие значения.

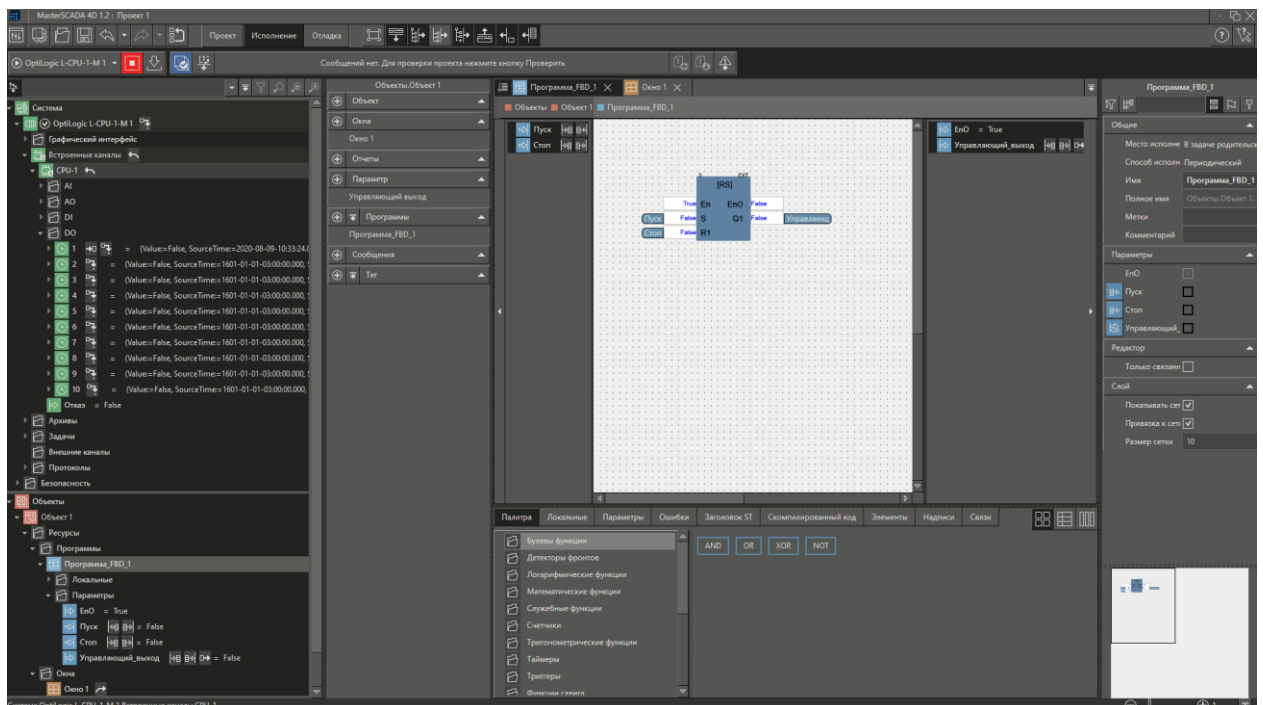


Рисунок 50 – Текущие значения параметров.

Нажмём кнопку Пуск в среде исполнения. Управляющий выход примет значение TRUE, это же значение поступит на дискретный выход модуля, индикатор на мнемосхеме окрасится в зелёный цвет, и начнет вращаться насос.

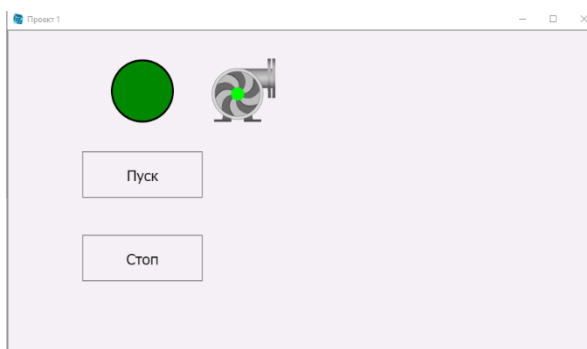
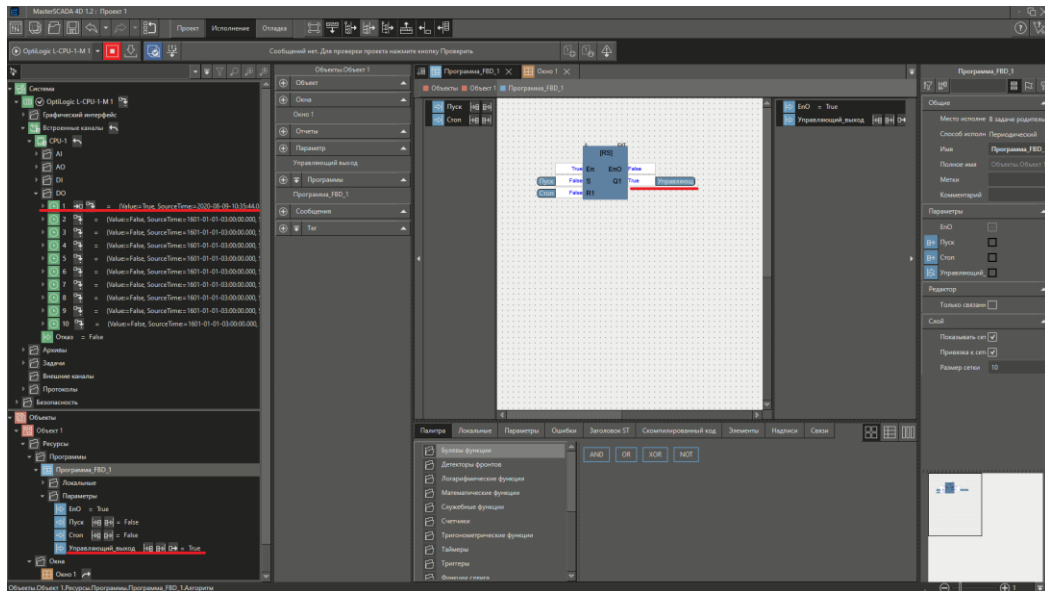


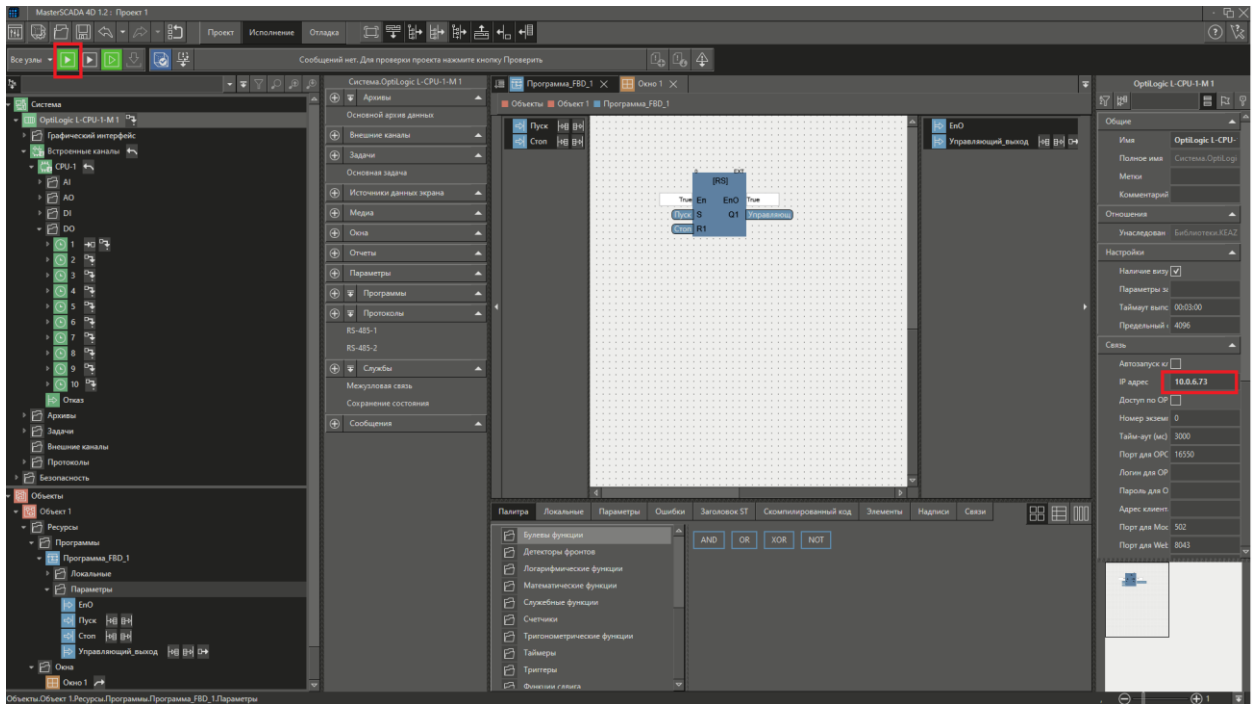
Рисунок 51 – Значения параметров по нажатию «Пуск».

Нажмем кнопку Стоп. Управляющий выход примет значение False, индикатор на мнемосхеме окрасится в белый цвет, а насос остановится.

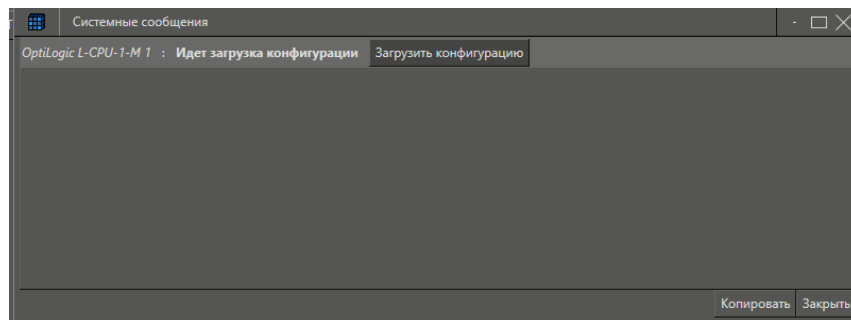
За счёт чего это происходит? Пока кнопка Пуск нажата, значение параметра Пуск на входе программы равно TRUE. Это значение поступает на вход S триггера, устанавливая выход триггера в состояние TRUE. После отпускания кнопки триггер остаётся в прежнем состоянии до тех пор, пока не будет нажата кнопка Стоп – это приведёт к появлению значения TRUE на входе R триггера и, соответственно, к установке выхода триггера в состояние False.

Теперь, если Ваш компьютер доступен по сети с других устройств, откройте браузер на другом устройстве, наберите в адресной строке IP-адрес компьютера, на котором Вы запустили режим исполнения, и убедитесь, что Вы можете управлять насосом совершенно одинаково и в любой последовательности с любого из устройств.

Теперь загрузим конфигурацию в контроллер. Для этого отключим режим исполнения, выделим в дереве контроллер, убедимся, что указан его IP-адрес и нажмем кнопку Подключить.



*Рисунок 52 – Проверка корректности указания IPадреса.
Должна произойти загрузка конфигурации в контроллер.*



*Рисунок 53 – Окно подтверждения загрузки конфигурации.
В адресной строке браузера наберем IP-адрес контроллера, с
указанием порта 8043 (порт должен быть открыт в файрволе)*

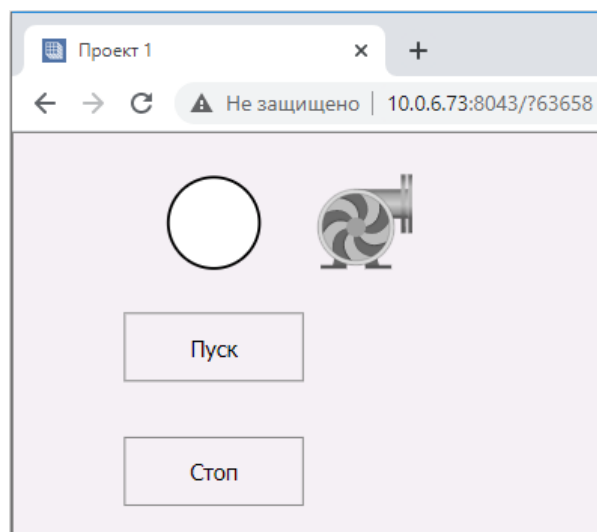


Рисунок 54 – Внешний вид формы в браузере.

При нажатии кнопки Пуск, насос включится, индикатор загорится зеленым и загорится светодиод выхода DO1 контроллера.

12 РАЗРАБОТКА ВСПЛЫВАЮЩЕГО ОКНА

Реализуем управление насосом в отдельном всплывающем окне.

Сначала добавим в Объект 1 ещё одно окно. Для этого в контекстной панели объекта нажмем соответствующую кнопку:

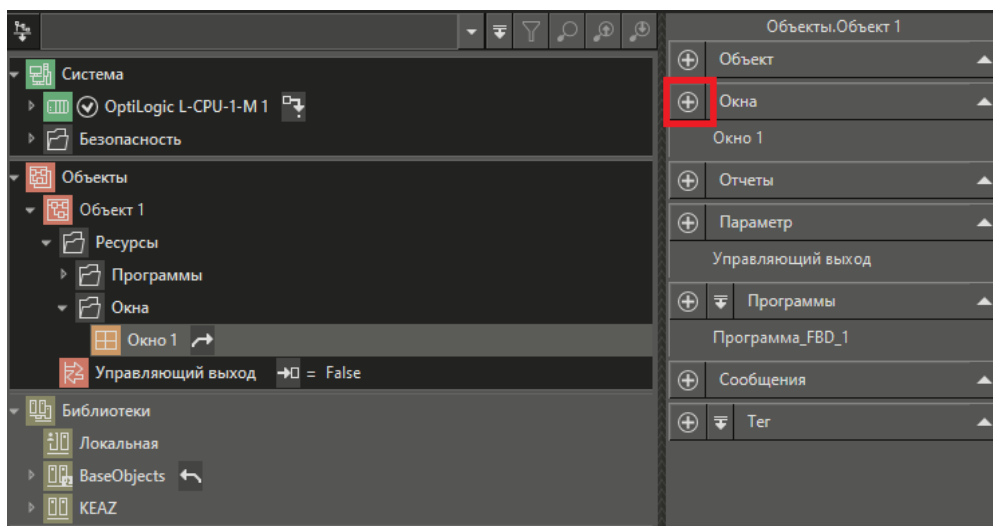


Рисунок 55 – Добавление окна управления.

Назовем новый элемент Окно управления. Для изменения имени, присвоенного добавленному элементу по умолчанию, нужно воспользоваться панелью свойств. После создания окна автоматически откроется графический редактор:

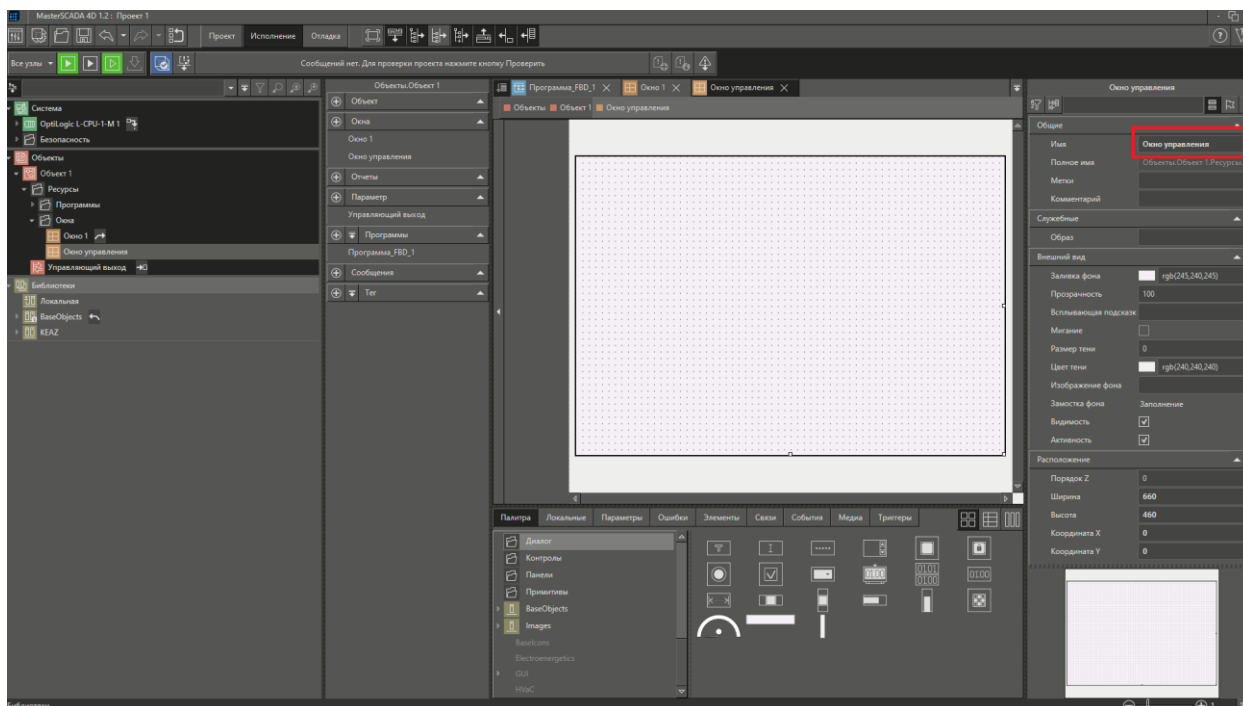


Рисунок 56 – Графический редактор.

Перенесём в него элементы управления и индикации из основного окна.

Для этого перейдём на вкладку *Окно 1*, при помощи мыши выделим обе кнопки и индикатор, и вырежем эти элементы, либо нажав комбинацию клавиш *Ctrl+X*, либо при помощи кнопки *Вырезать* на панели инструментов, либо при помощи контекстного меню выделенных элементов.

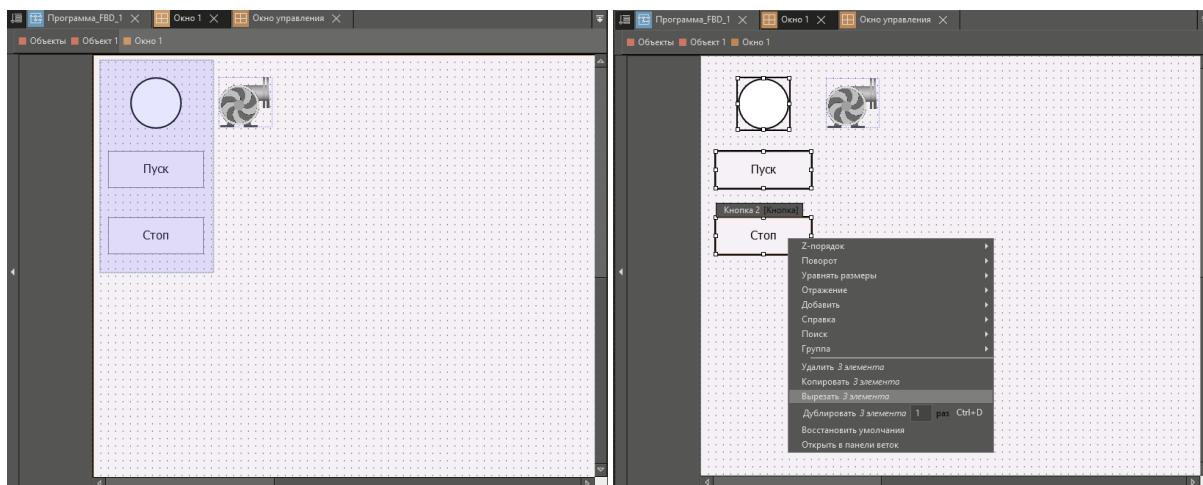


Рисунок 57 – Элементы управления.

Теперь перейдём на вкладку *Окно управления* и, либо через контекстное меню *Вставить*, либо нажав комбинацию клавиш *Ctrl+V*, вставим туда вырезанные элементы.

При этом появится диалоговое окно восстановления связей:

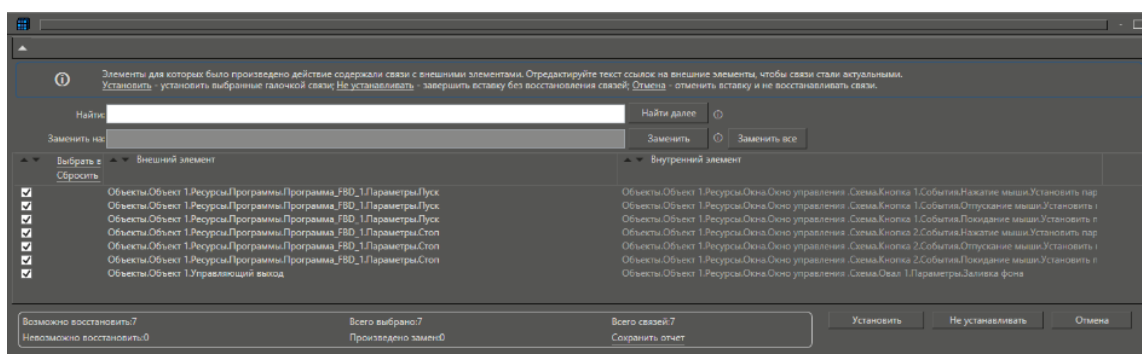


Рисунок 58 – Окно восстановления связей.

Нажмем кнопку *Установить* и получим результат:

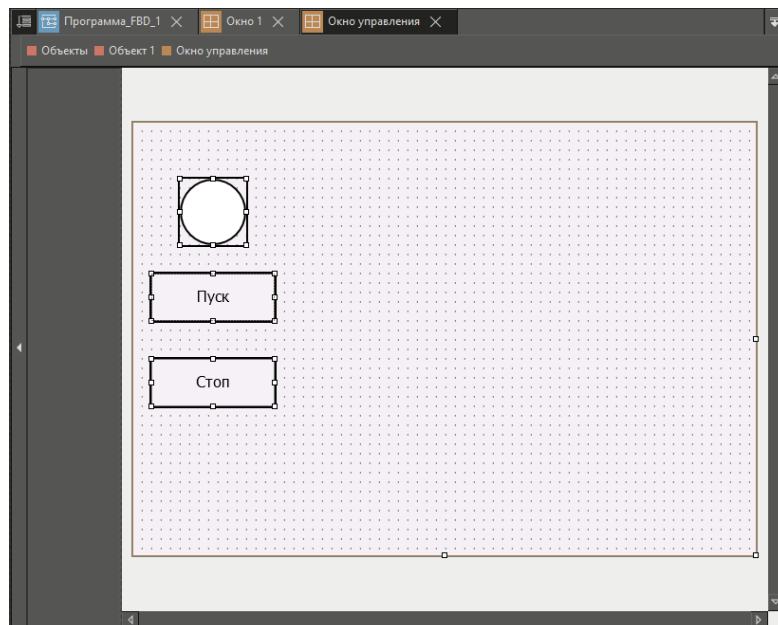


Рисунок 59 – Результат восстановления связей.

Затем уменьшим размер окна, потянув влево правую границу окна и вверх - нижнюю:

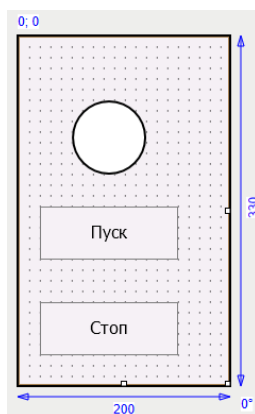


Рисунок 60 – Масштабирование окна.

Теперь нам предстоит сделать так, чтобы это окно открывалось по клику мыши на изображении насоса.

Для этого перейдём на вкладку Окно 1, которое является основным. Выделим в окне изображение насоса, нажав на него левой кнопкой

мыши. Панель свойств переведём в режим отображения событий:

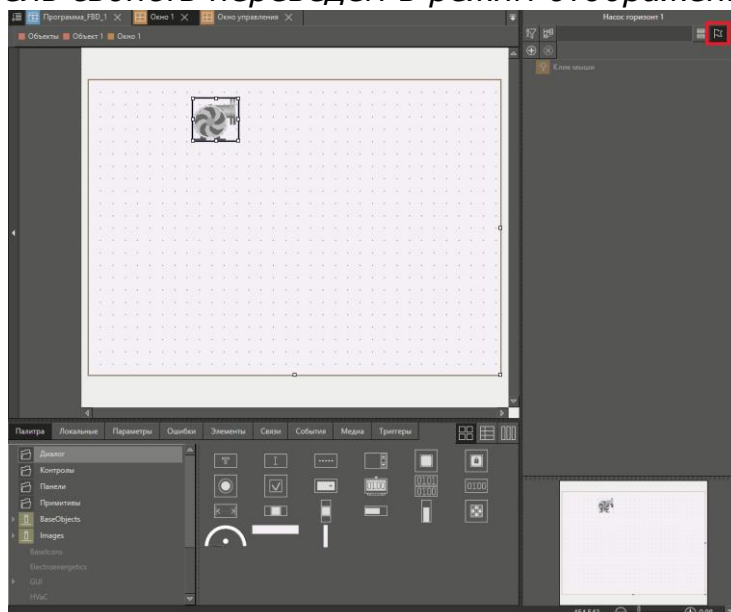


Рисунок 61 – Режим отображения событий.

Нажмем правой кнопкой мыши на событие Клик мыши. Из контекстного меню выберем Добавить, а из появившегося списка доступных действий – Открыть всплывающее окно:

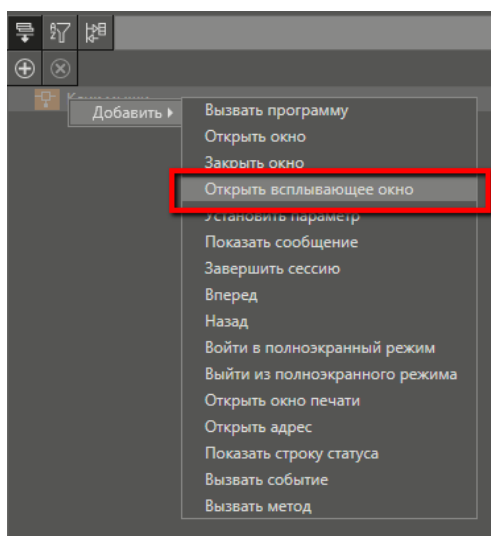


Рисунок 62 – Добавление действий.

Раскроем подробные параметры этого действия. Укажем, какое именно окно будет открываться по клику мыши. Для этого достаточно перетащить требуемое окно (в нашем случае это Окно управления) на параметр Окно действия. В результате этого название окна появится в качестве значения параметра Окно.

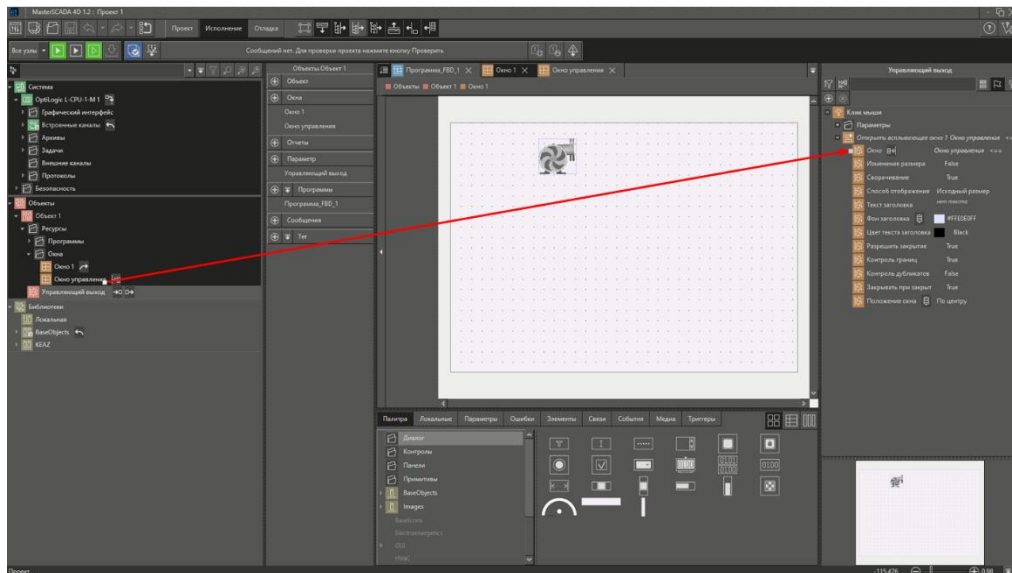


Рисунок 63 – Работа с окном действий.

Переместим элемент Насос левее:

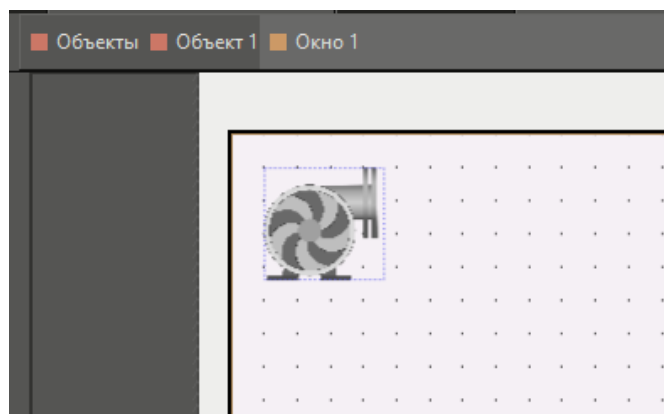


Рисунок 64 – Размещение элемента «Насос».

На этом конфигурирование всплывающего окна завершено

Запустите проект на исполнение (предварительно проверив на ошибки) в режиме эмуляции, либо загрузите в контроллер.

В клиенте визуализации нажмите левой кнопкой мыши на изображении насоса - откроется всплывающее окно с кнопками Пуск, Стоп и индикатором состояния. Нажмем кнопку Пуск – насос включится.

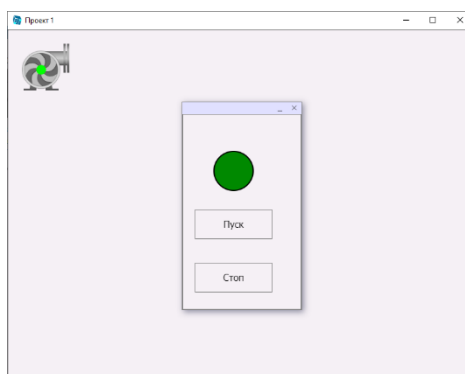


Рисунок 65 – Новый вариант формы.

Нажмем кнопку Стоп – насос остановится. Органы управления вынесены в отдельное окно.

13 ОБЪЕКТНЫЙ ПОДХОД ПРИ СОЗДАНИИ ПРОЕКТА

Для начала рассмотрим, что представляет собой объект. В MasterSCADA 4D объект – это именованная совокупность, объединяющая разнородные элементы проекта и связи между ними.

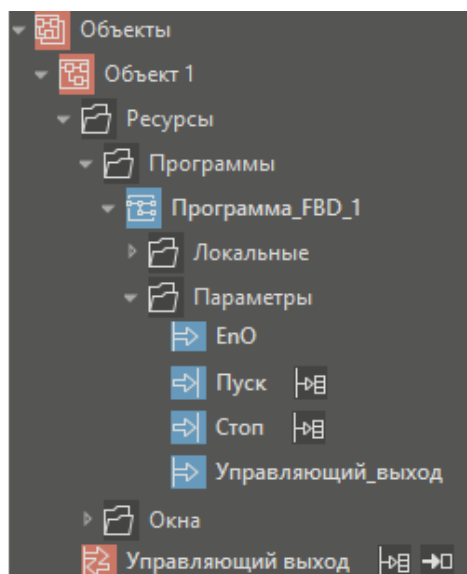


Рисунок 66 – Дерево объектов.

В объект могут входить программы, окна, отчеты и другие элементы, в том числе и вложенные объекты. Различные параметры этих элементов могут быть связаны между собой. Сам объект тоже имеет параметры – они предназначены для его связи с "внешним миром". Основное назначение этого принципа организации объекта – облегчение процедуры тиражирования. Копия такого объекта создаётся с сохранением всей внутренней структуры, включая межэлементные связи внутри объекта. Разработчику останется лишь подключить внешние.

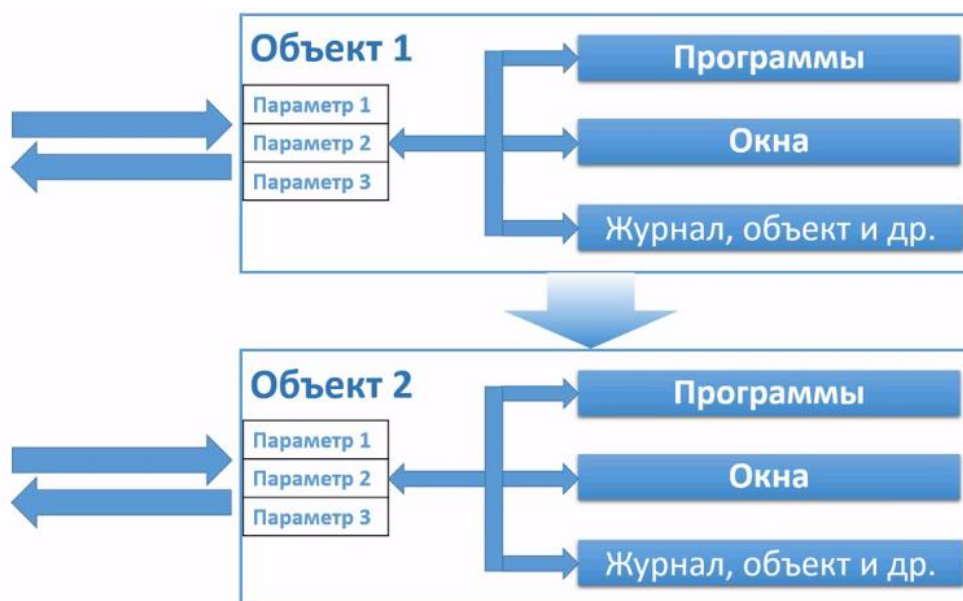


Рисунок 67 – Объекты и его параметры.

Теперь посмотрим как это можно применить на практике.

Ранее мы создали объект, состоящий из окна с изображением насоса, окна управления этим насосом, вызываемого по нажатию мышью на изображении насоса, и программы, реализующей логику управления насосом. Объект имеет один параметр для связи с внешним миром – это управляющий выход, сигнал с которого поступает на физический дискретный выход модуля контроллера.

А теперь зададимся целью создать насосную станцию, состоящую из нескольких насосов.

Переименуем созданный ранее объект в Насос 1.

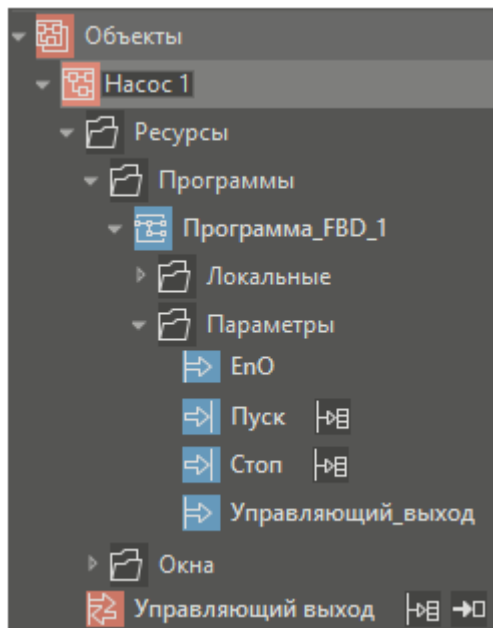


Рисунок 68 – Переименование объекта.

В дереве объектов создадим ещё один объект и назовём его Насосная станция.

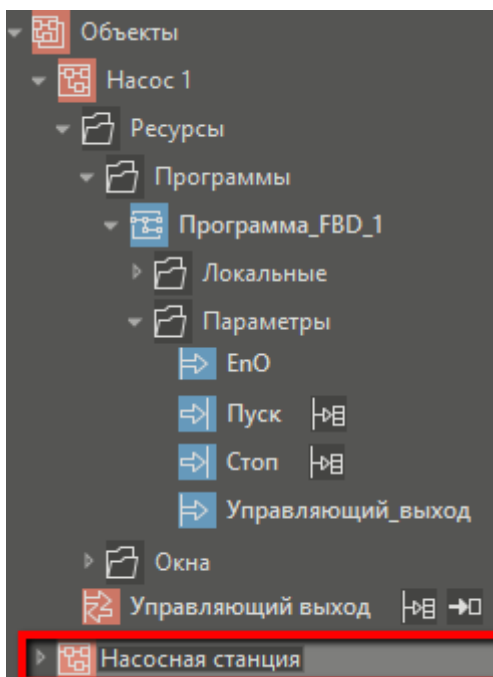


Рисунок 69 – Создание объекта «Насосная станция».

Теперь перетащим левой кнопкой мыши насос в насосную станцию. У нас получился корневой объект Насосная станция, у которого есть вложенный объект Насос 1.

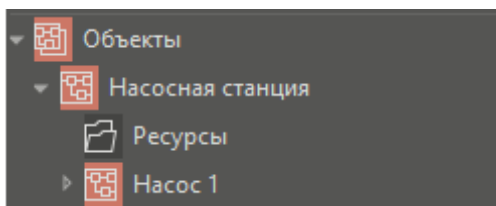


Рисунок 70 – Корневой объект «Насосная станция».

Перенесём его в библиотеку Локальная.

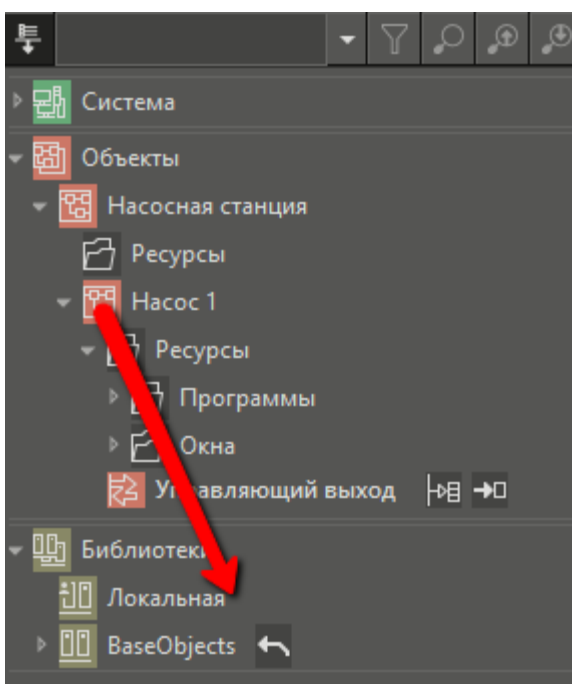


Рисунок 71 – Перенос объекта в локальную библиотеку.

Теперь насос стал библиотечным элементом, а насос в дереве объектов – его наследником, о чём нам говорит специальный значок рядом с именем и всплывающая подсказка.

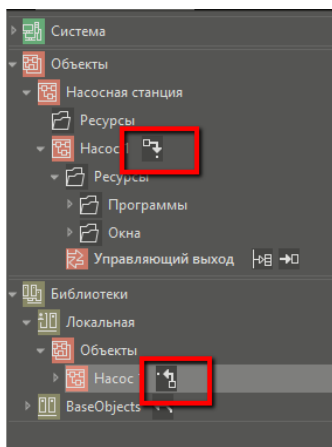


Рисунок 72 – Библиотечный объект.

Перетащим библиотечный насос в насосную станцию. Получаем два насоса.

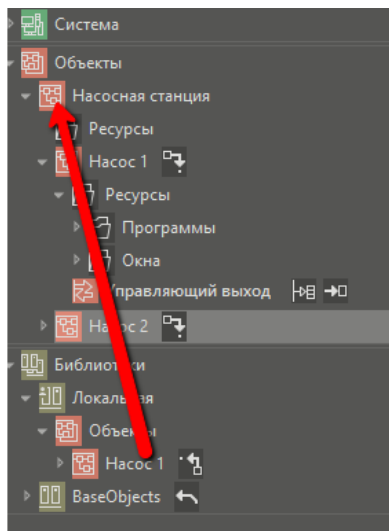


Рисунок 73 – Дублирование библиотечных объектов.

Повторим эту процедуру ещё раз – теперь насосная станция включает уже три насоса, каждый из которых имеет один параметр для связи с внешними элементами или устройствами (Управляющий выход), а внутри содержит окна и программу управления.

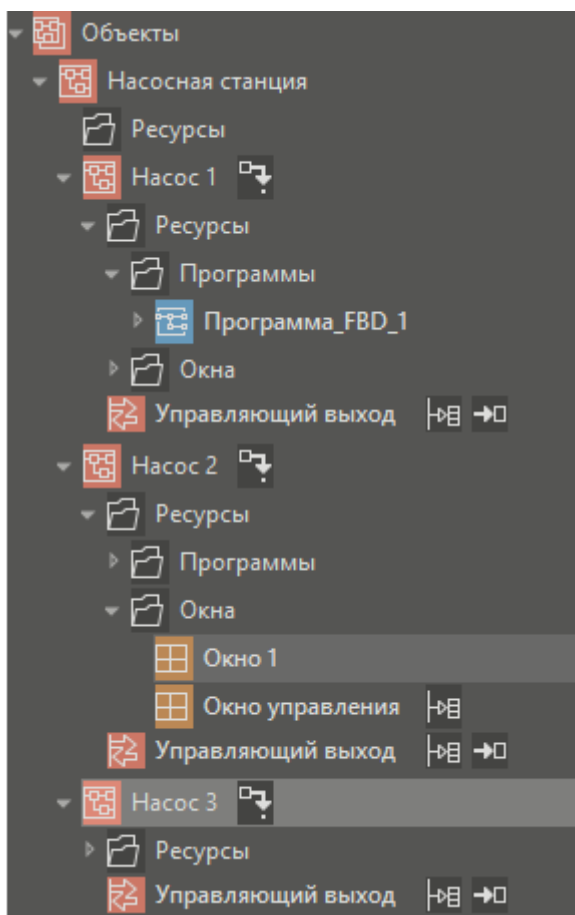


Рисунок 74 – Наследование объектов.

Все насосы в дереве объектов являются наследниками библиотечного. Это означает, что изменение чего-либо в библиотечном насосе отразится во всех его наследниках, то есть будет унаследовано.

Например, раскроем дерево библиотечного насоса до содержимого папки Окна и переименуем Окно 1 в Изображение Насоса:

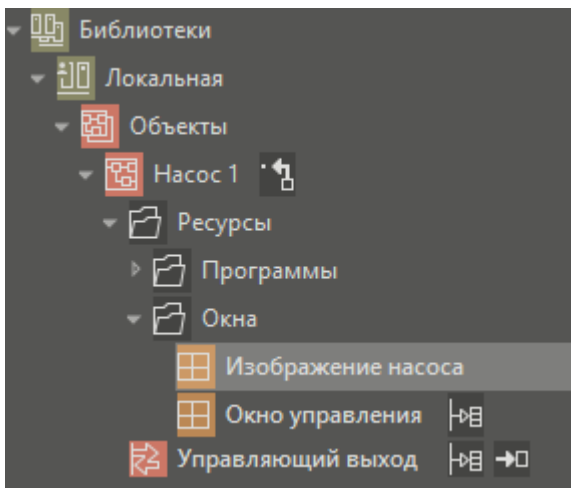


Рисунок 75 – Переименование параметров наследованных объектов.

Увидим, что в дереве объектов это окно тоже переименовалось:

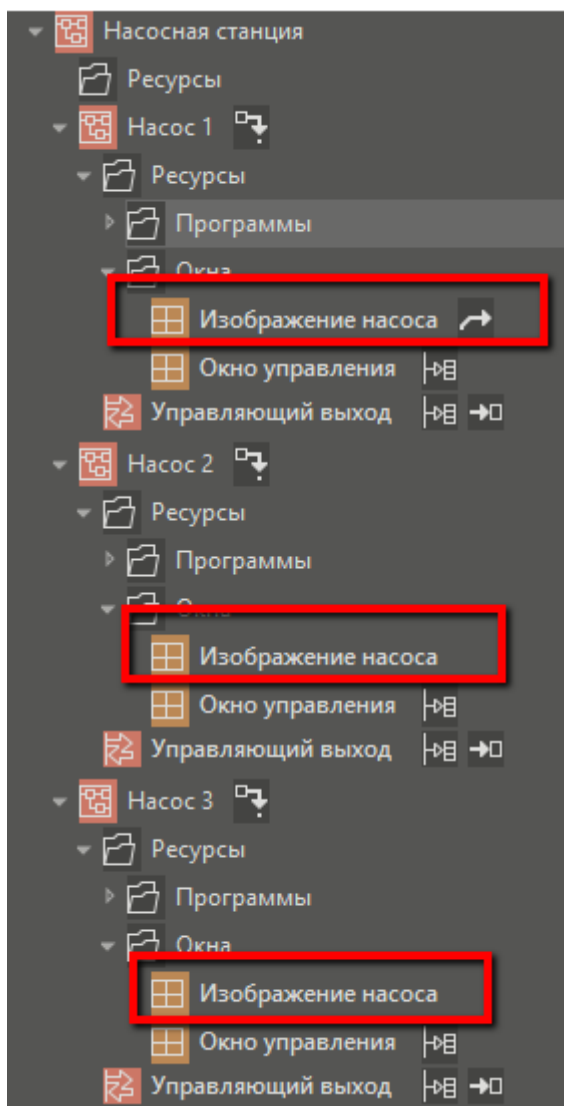


Рисунок 76 – Результат переименования параметров наследованных объектов.

То есть изменение унаследовалось. Строго говоря, изменения будут наследоваться не всегда, а только до тех пор, пока не будет отредактирован сам наследник, и он не станет отличаться от библиотечного элемента. После этого наследование изменений будет сохраняться только для части наследника, совпадающей с библиотечной. Подробнее эти механизмы мы рассмотрим отдельно в других уроках.

А сейчас внесём ещё некоторые изменения. Откроем окно Изображение Насоса из библиотеки и сделаем его компактнее (окно наследуемого насоса закроем):

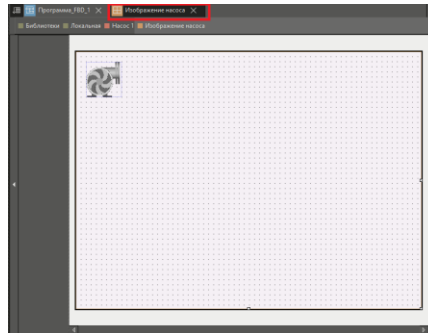


Рисунок 77 – Масштабирование.

Добавим надпись, отражающую состояние насоса. Для этого перетащим правой кнопкой мыши параметр Управляющий выход из библиотечного насоса (мы сейчас работаем с библиотечным элементом), и из открывшегося контекстного меню выберем Вставить.Текст:

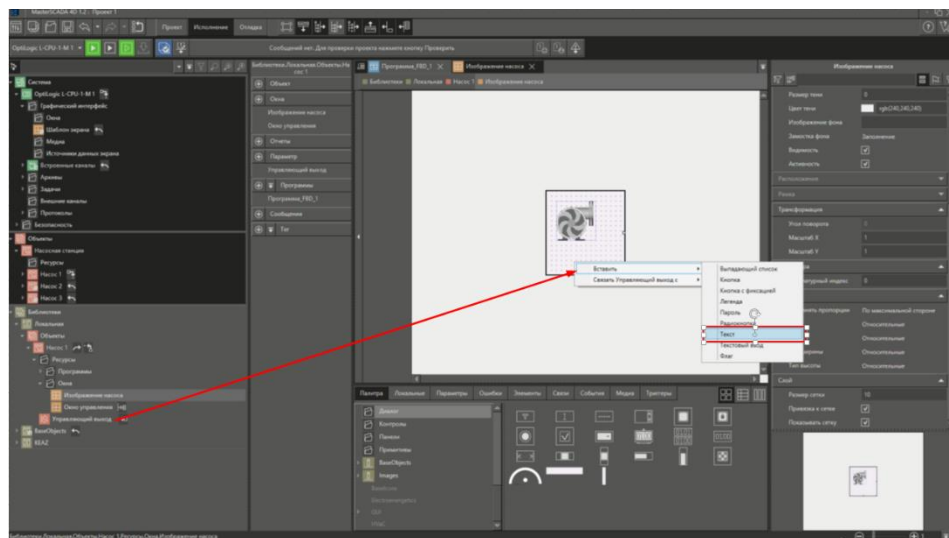


Рисунок 78 – Добавление текста.

Очень часто рядом с изображением объекта отображают какой-нибудь основной параметр его работы, а иногда и несколько. Для клапана – это его положение; для двигателя с частотным преобразователем – частота, и так далее. Мы выведем просто состояние. Перейдем на панель свойств элемента Текст, откроем конвертор на связи параметра Текст. Выберем точечную конвертацию, и для значения False укажем Останов, а для значения TRUE – Работа.

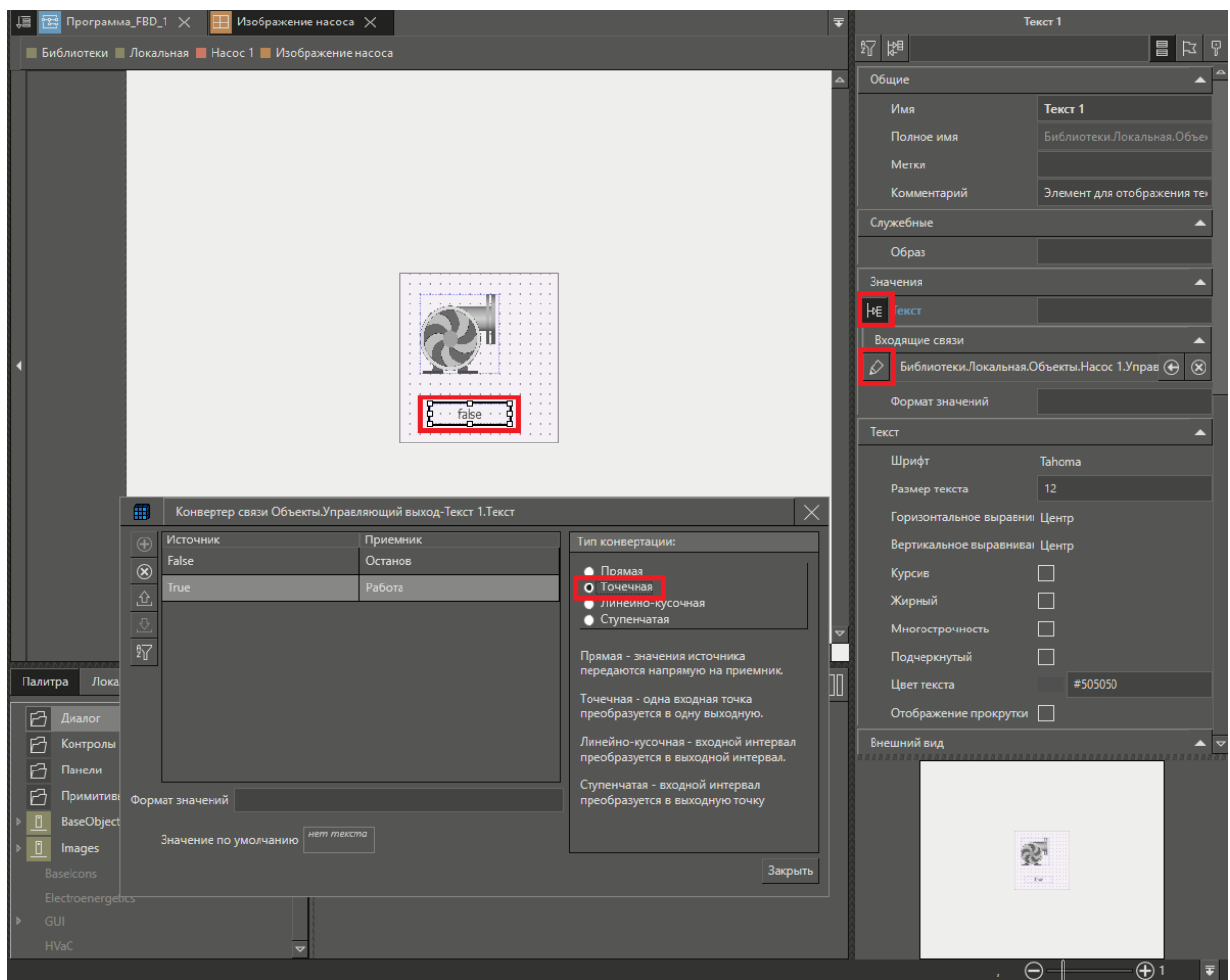


Рисунок 79 – Добавление основного параметра.

Закроем конвертор.

Уменьшим размер окна до размеров картинке, добавим в насосной станции окно и назовём его Мнемосхема:

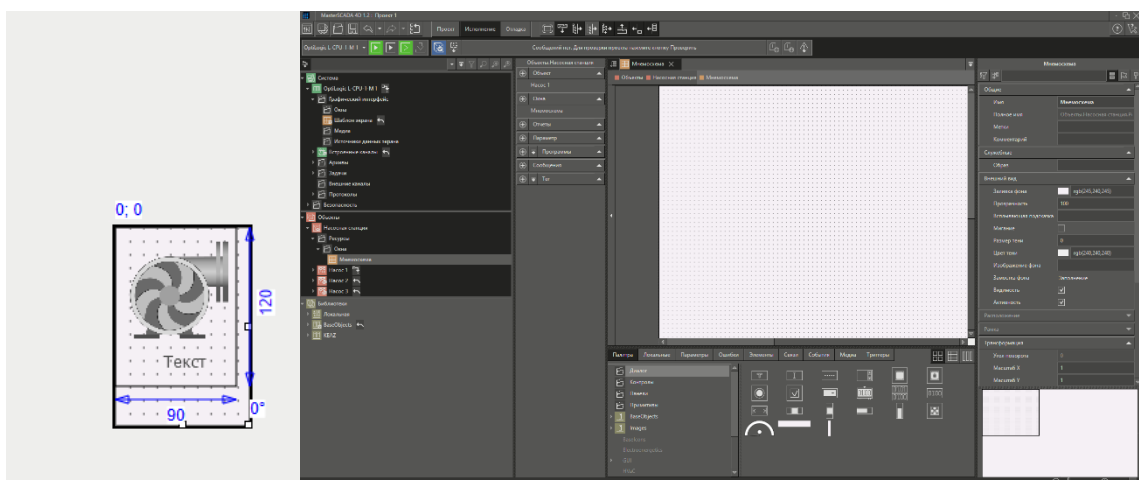


Рисунок 80 – Мнемосхема корневого объекта.

Назначим Мнемосхему стартовым окном узла при помощи контекстного меню:

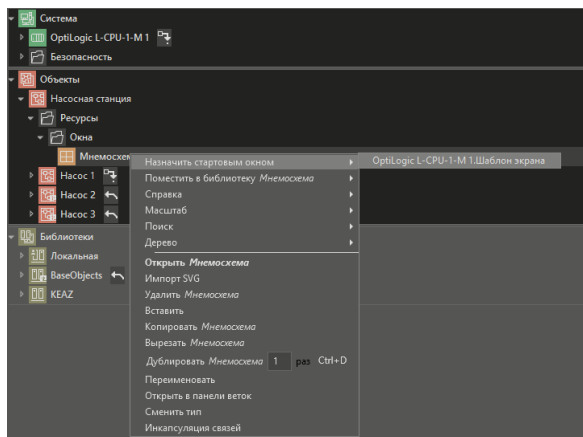


Рисунок 81 – Назначение мнемосхемы стартовым окном.

Теперь перетащим поочередно все насосы в окно:

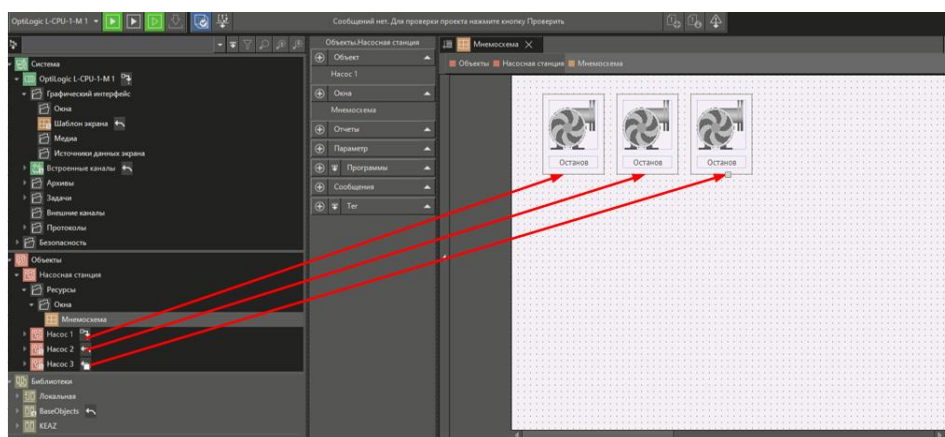


Рисунок 82 – Добавление объектов на мнемосхему корневого объекта.

Свяжем выходы новых насосов с физическими выходами ПЛК OptiLogic. Связывание удобнее производить, открыв деревья связываемых элементов в параллельных окнах. Нажмем правой кнопкой мыши на объект Насосная станция, и из контекстного меню выберем Открыть в панели веток:

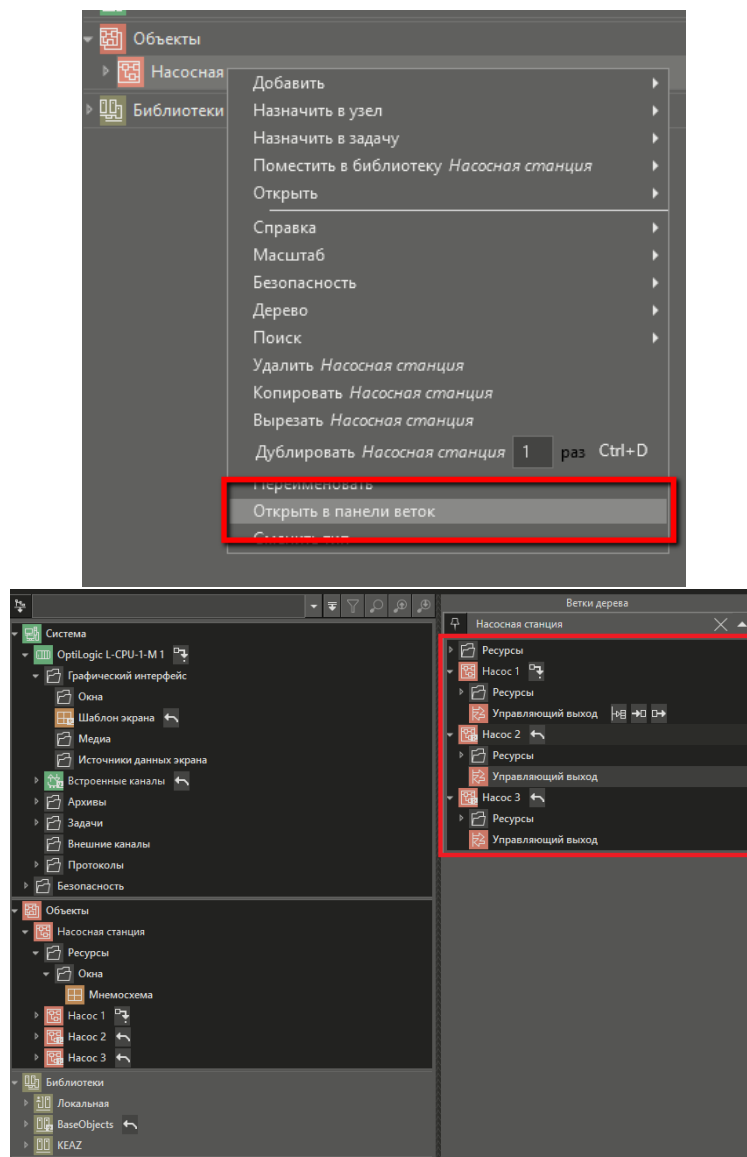


Рисунок 83 – Вызов панели веток.

Первый насос у нас уже связан. Теперь свяжем параметр Управляющий выход второго и третьего насоса с последующими дискретными выходами ПЛК OptiLogic:

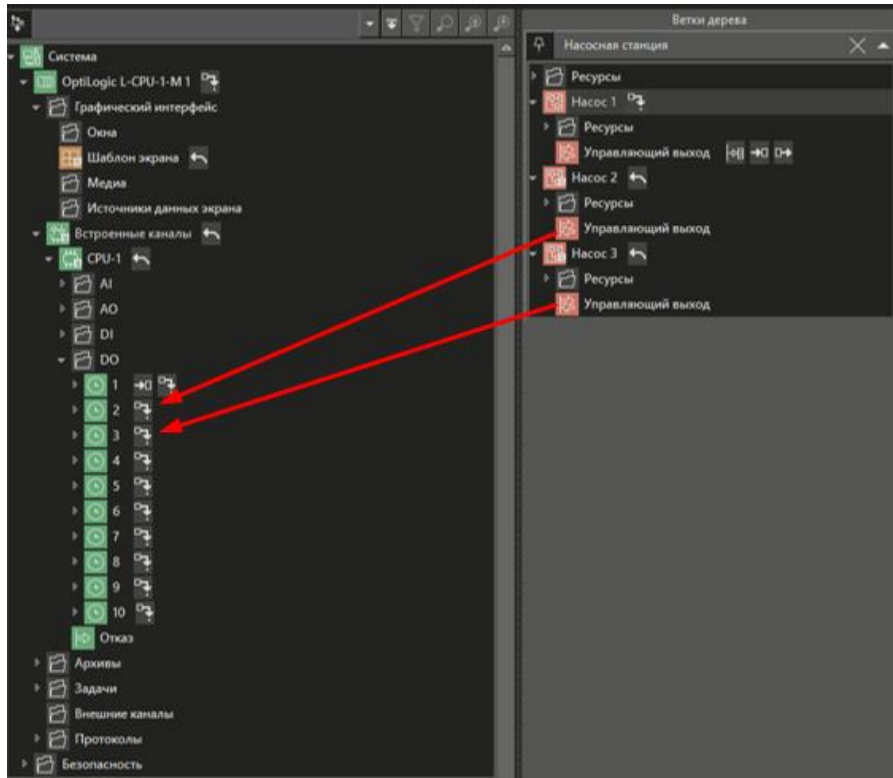


Рисунок 84 – Расстановка связей

Теперь запустим режим исполнения. В клиенте визуализации мы увидим созданную нами насосную станцию с тремя насосами, каждый из которых управляется независимо от других. Насосы можно включать или выключать в любой последовательности. Можно также закрывать ненужные окна, а затем снова их открывать и т.п.

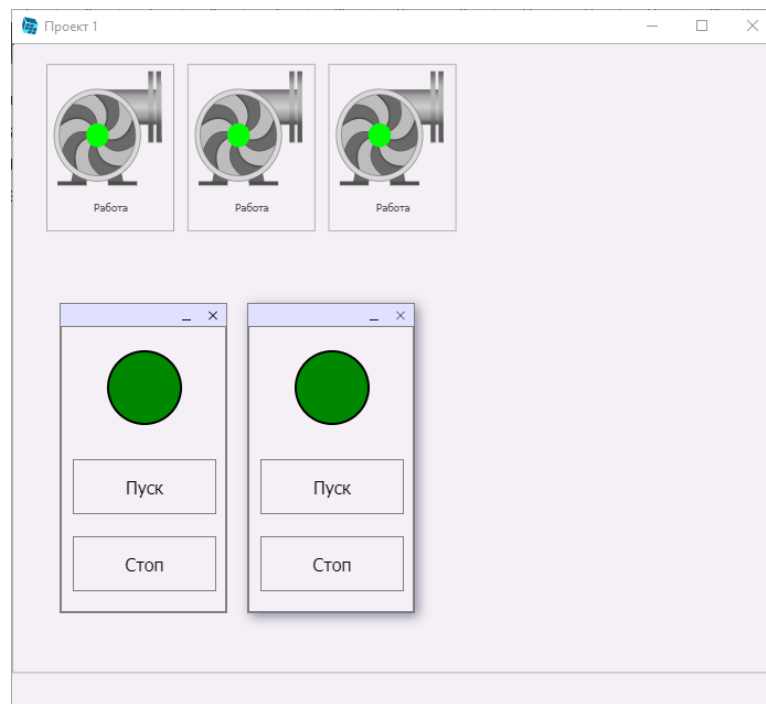


Рисунок 85 - Внешний вид формы корневого объекта

14 ОПЕРАЦИИ С БИБЛИОТЕКАМИ

В рамках создания проекта мы поместили тиражируемый элемент Насос в библиотеку Локальная. Библиотека Локальная доступна только в том проекте, в котором она была создана. Если же требуется, чтобы библиотечный элемент был доступен для использования и в других проектах, то необходимо создать его в другой библиотеке - в уже имеющейся или в новой. Нажмем правой кнопкой мыши на элемент Библиотеки и выберем пункт Добавить - Библиотека:

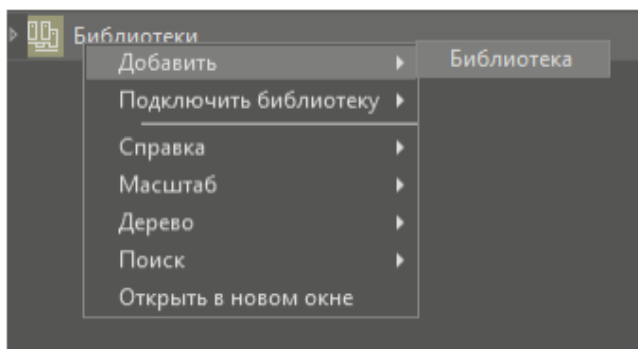


Рисунок 86 – Добавление объекта «Библиотека»

Назовём эту библиотеку Мои насосы. Скопируем созданный нами библиотечный насос из библиотеки Локальная и вставим его в библиотеку Мои насосы:

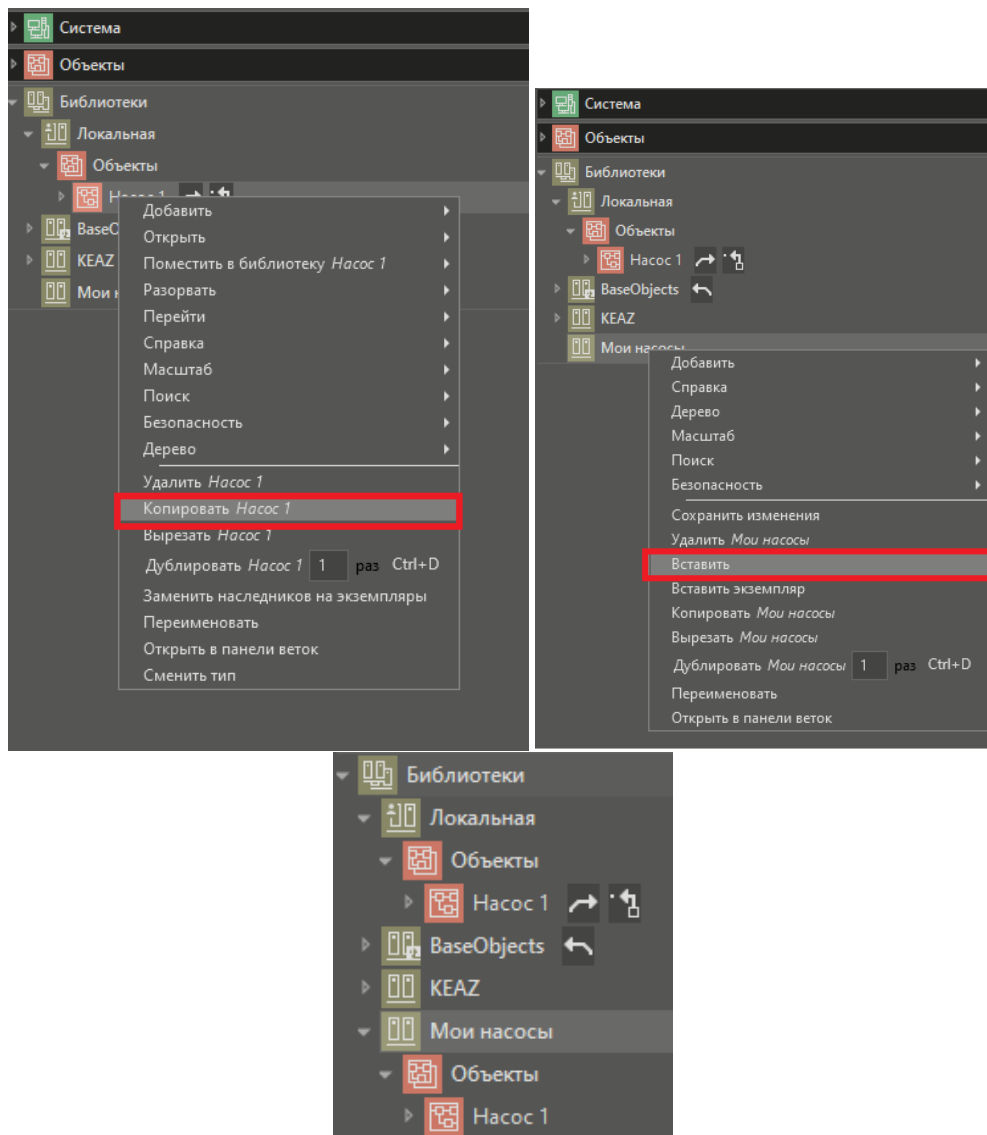


Рисунок 87 – Переименование объекта «Библиотека»

Затем нужно нажать правой кнопкой мыши на название библиотеки и выбрать пункт контекстного меню Сохранить изменения:

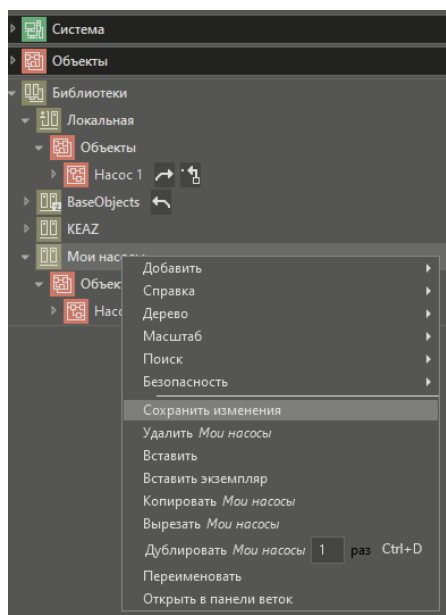


Рисунок 88 – Сохранение изменений

После этого библиотека сохранится на диске и будет доступна для подключения из других проектов через контекстное меню Подключить библиотеку. В данный момент она недоступна для подключения, т.к. уже подключена. Для того чтобы это продемонстрировать, удалим библиотеку Мои насосы из проекта через контекстное меню Удалить:

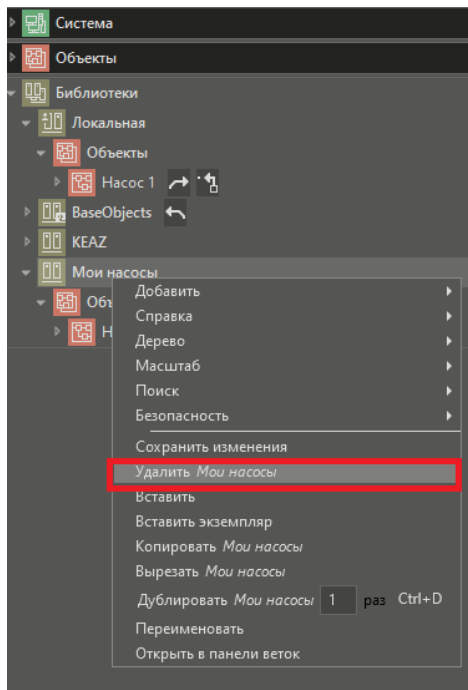


Рисунок 89 – Удаление библиотеки

Теперь подключим её к проекту: для этого нажмем правую кнопку мыши, выберем пункт Подключить библиотеку, и в открывшемся списке доступных библиотек выберем Мои насосы:

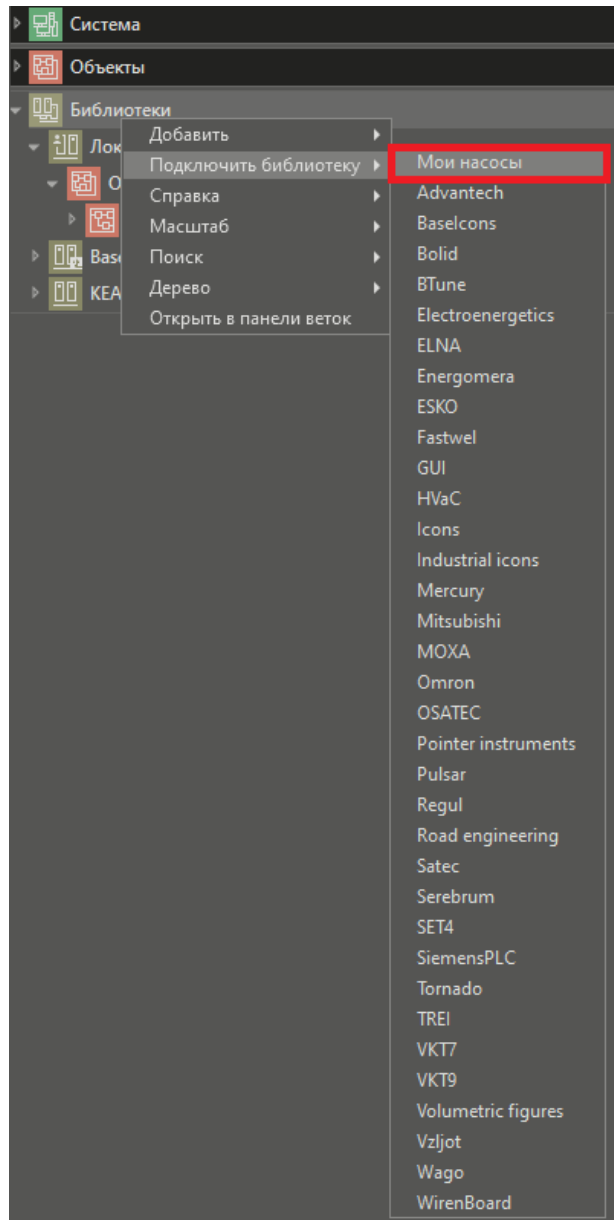


Рисунок 90 – Подключение пользовательской библиотеки к проекту

Библиотека подключилась и теперь её можно использовать.

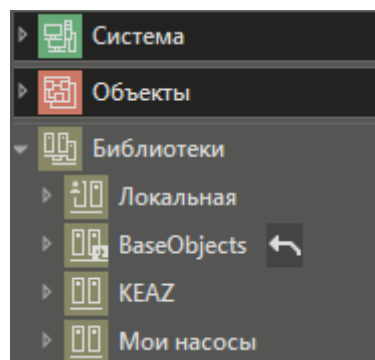


Рисунок 91 – Корректно подключенная библиотека