R-LOGIC RUBEZH

ООО «РУБЕЖ»

СРЕДА РАЗРАБОТКИ ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ R-Logic.Designer

РУКОВОДСТВО ПРОГРАММИСТА

ТШВГ.305659.221 33

Редакция 1

Саратов 2025

Содержание

ВВЕДЕНИЕ **4**

Условные обозначения и сокращения **5**

1 Системные требования **7**

2 Установка **8**

2.1 Установка на Windows 10 ..... **8**

2.2 Установка на Linux **12**

3 Общая информация **14**

4 Основное окно программы **15**

4.1 Меню программы **15**

4.2 Панель инструментов **20**

4.3 Окно проекта **22**

4.4 Окно свойств **23**

4.5 Окно контекстной справки **23**

4.6 Окно сообщений **24**

4.7 Библиотека компонентов **24**

5 Добавление и удаление элементов дерева проекта **27**

5.1 Контекстное меню добавления шасси **27**

5.2 Контекстное меню добавления модулей **27**

5.3 Контекстное меню добавления программных компонентов **27**

5.4 Контекстное меню добавления служб **27**

5.5 Удаление элементов из дерева проекта **28**

6 Работа с проектом **29**

6.1 Создание нового проекта **29**

6.2 Открытие существующего проекта **30**

6.3 Настройка резервирования **30**

6.4 Настройка аппаратной конфигурации **31**

6.5 Создание программных компонентов **37**

6.6 Создание и привязка переменных **38**

6.7 Редакторы **49**

6.8 Конфигурация запуска **59**

6.9 Создание и настройка служб контроллера **59**

7 Загрузка проекта в контроллер 108

8 Средства мониторинга и отладки 111

9 Утилита R-Logic.Загрузчик 118

10 Языки стандарта МЭК 61131-3 121

10.1 Типы данных 121

10.2 Преобразование данных 122

10.3 Конверсия типов данных 127

10.4 Пользовательские типы данных 128

10.5 Язык ST 130

ВВЕДЕНИЕ

Система разработки технологических программ R-Logic.Designer предназначена для создания приложений управления технологическим процессом. R-Logic.Designer включает набор компонентов для обеспечения процесса подготовки исходного кода технологической программы, описания связей переменных с их физическими представлениями (каналами ввода-вывода), компиляции и отладки приложения.

Документ предназначен для разработчиков программного обеспечения, а также для проектировщиков систем контроля и управления. При работе рекомендуется ознакомиться со следующей документацией:

- «Контроллер логический программируемый R-Logic Стандарт. Руководство по эксплуатации»;
- «Контроллер автоматизации программируемый R-Logic Лайт. Руководство по эксплуатации»;
- Настоящее руководство.

Условные обозначения и сокращения

APDU – Application Protocol Data Unit – Блок данных прикладного протокола

ASDU – Application Specific Data Unit – Блок данных прикладного уровня

BCD – Binary Coded Decimal – Двоично-десятичный код

CO – Coils – Дискретные выходы

DA – Data Attributes – Атрибуты данных

DI – Discrete Inputs – Дискретные входы

DHCP – Dynamic Host Configuration Protocol – Протокол динамической настройки узла

DIP-переключатель – Движковый переключатель в корпусе типа DIP (Dual In-line Package)

FBD – Function Block Diagram – Функциональные блоковые диаграммы

HART – Highway Addressable Remote Transducer – Магистральный адресуемый удаленный преобразователь

HR – Holding Registers – Регистры хранения

IED – Intelligent Electronic Device – Интеллектуальное электронное устройство

IOA – Information Object Address – Адрес объекта информации

IR – Input Registers – Входные регистры

LD – Ladder Diagram – Релейно-контактные схемы

MTU – Maximum Transmission Unit – Максимальная единица передачи

NTP – Network Time Protocol – Протокол сетевого времени

PDU – Protocol Data Unit — Блок данных протокола

POU – Program Organization Unit – Компонент организации программ, программный компонент

SCP – Secure Copy – Безопасная копия

SSH (Secure Shell) – Протокол «Безопасная оболочка»

ST – Structured Text – Структурированный текст

АЦП (ADC – Analog-to-Digital Converter) – Аналого-цифровой преобразователь

БД – База данных

БДРВ – База данных реального времени

МК – Микроконтроллер

ПК – Персональный компьютер

ПКМ – Правая кнопка мыши

ПЛК – Контроллер логический программируемый

ПО – Программное обеспечение

ППО – Прикладное программное обеспечение

СТ – Сторожевой таймер

ТВ – Тип взаимодействия

УСО – Устройство связи с объектом

ФБ – Функциональный блок

ФС – Файловая система

ЦАП (DAC – Digital-to-Analog Converter) – Цифро-аналоговый преобразователь

ЦПУ – Центральное процессорное устройство

1 Системные требования

Операционная система: Windows 10, Linux.

Минимальные системные требования должны соответствовать минимальным системным требованиям к операционной системе Windows 10.

Встроенная видеокарта.

Место на диске: не менее 4 ГБ.

Сетевая карта Ethernet (для связи с контроллерами).

2 Установка

2.1 Установка на Windows 10

Удалите R-Logic.Designer предыдущей версии (если установлен). Для установки R-Logic.Designer запустите установочный файл *R-Logic.Designer_setup*. После отображения на экране ПК окна установщика (рисунок 1) следуйте инструкциям мастера установки (рисунки 2-8).

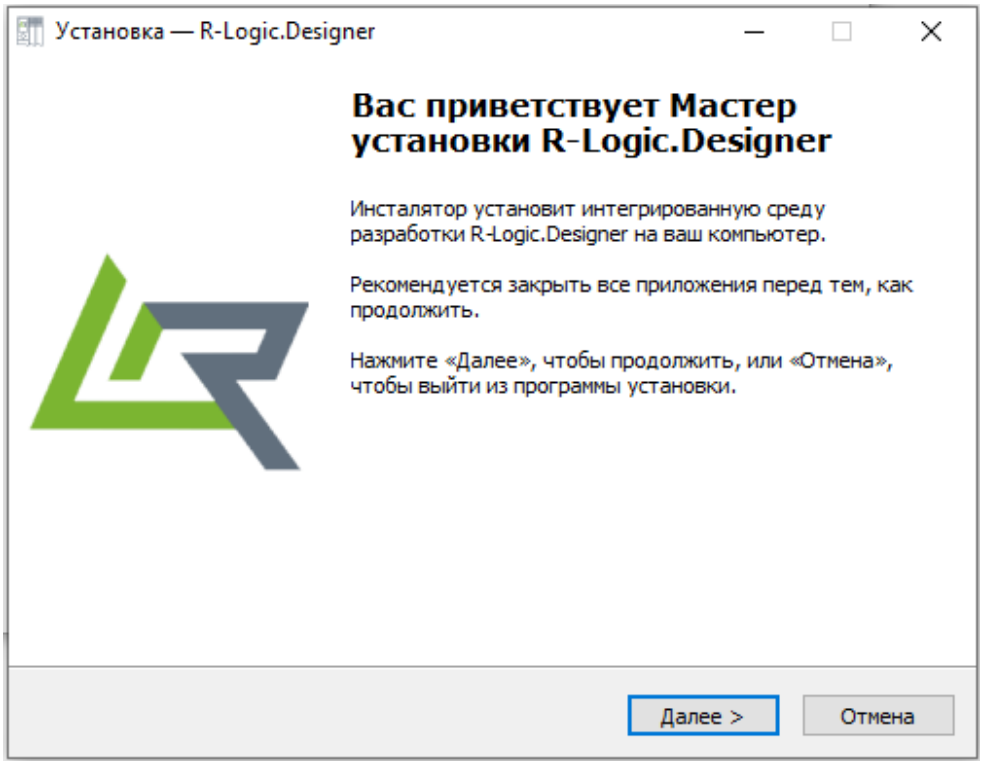

Рисунок 1 – Мастер установки R-Logic.Designer

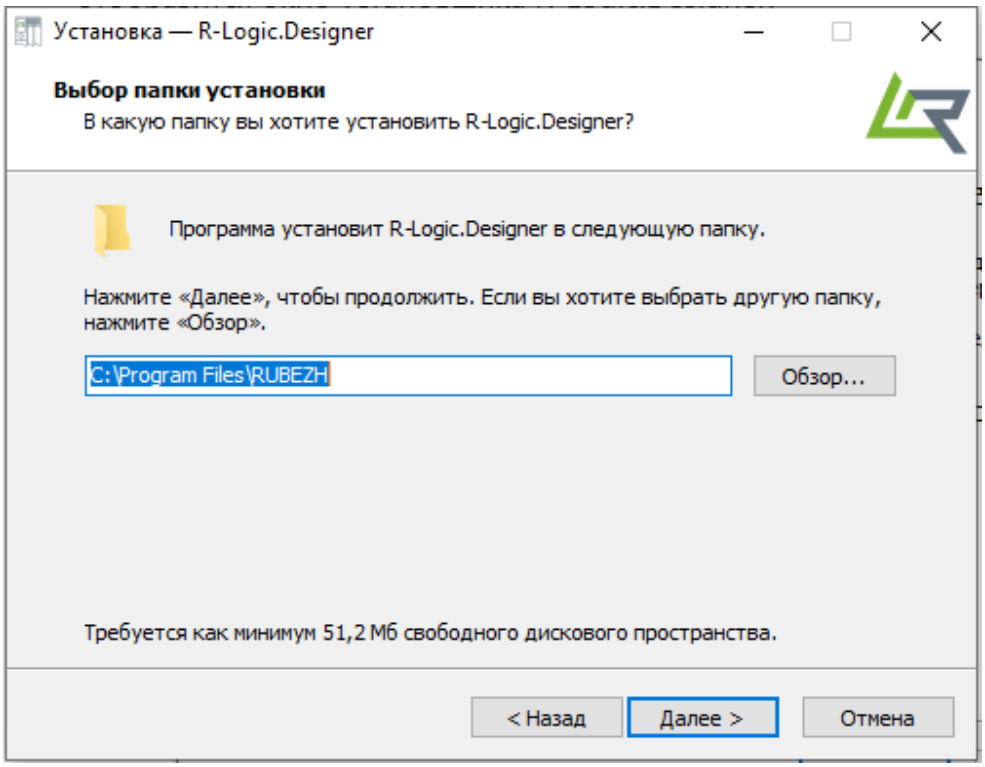

Рисунок 2 – Мастер установки R-Logic.Designer (шаг 1)

При повторной установке в уже существующую папку инсталлятор запросит подтверждение (рисунок 3).

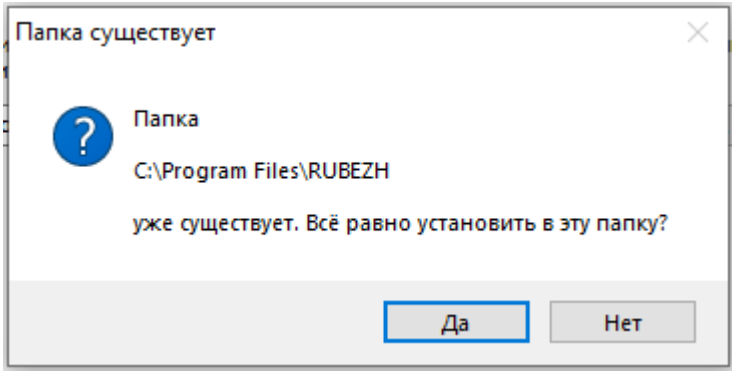

Рисунок 3 – Окно подтверждения пути установки R-Logic.Designer

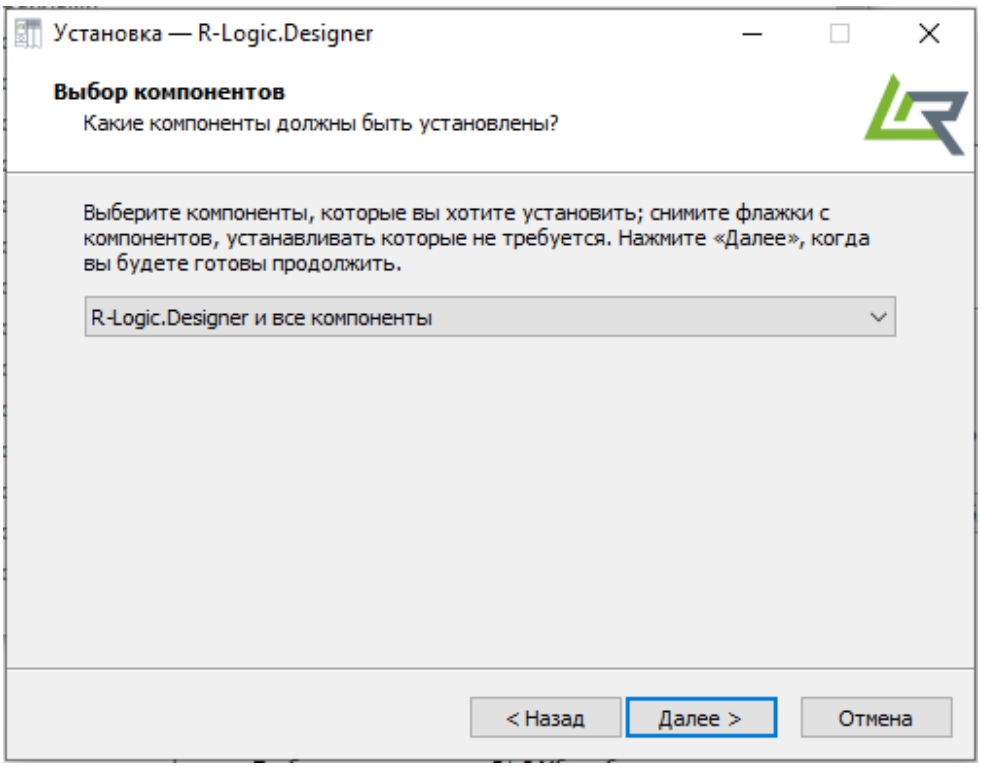

Рисунок 4 – Мастер установки R-Logic.Designer (шаг 2)

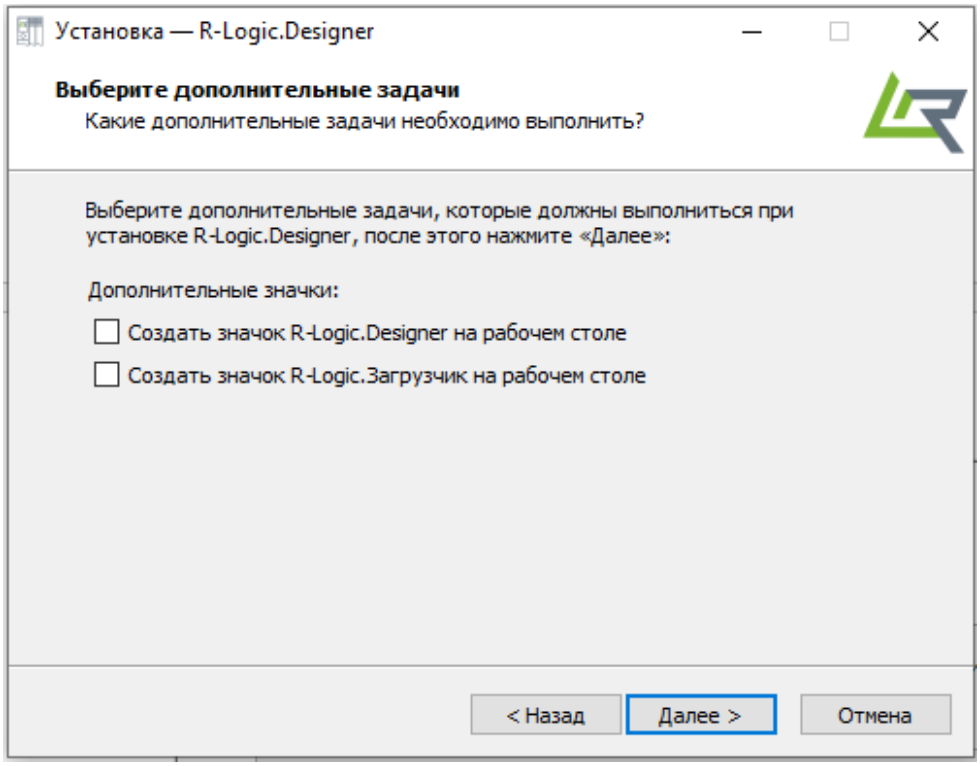

Рисунок 5 – Мастер установки R-Logic.Designer (шаг 3)

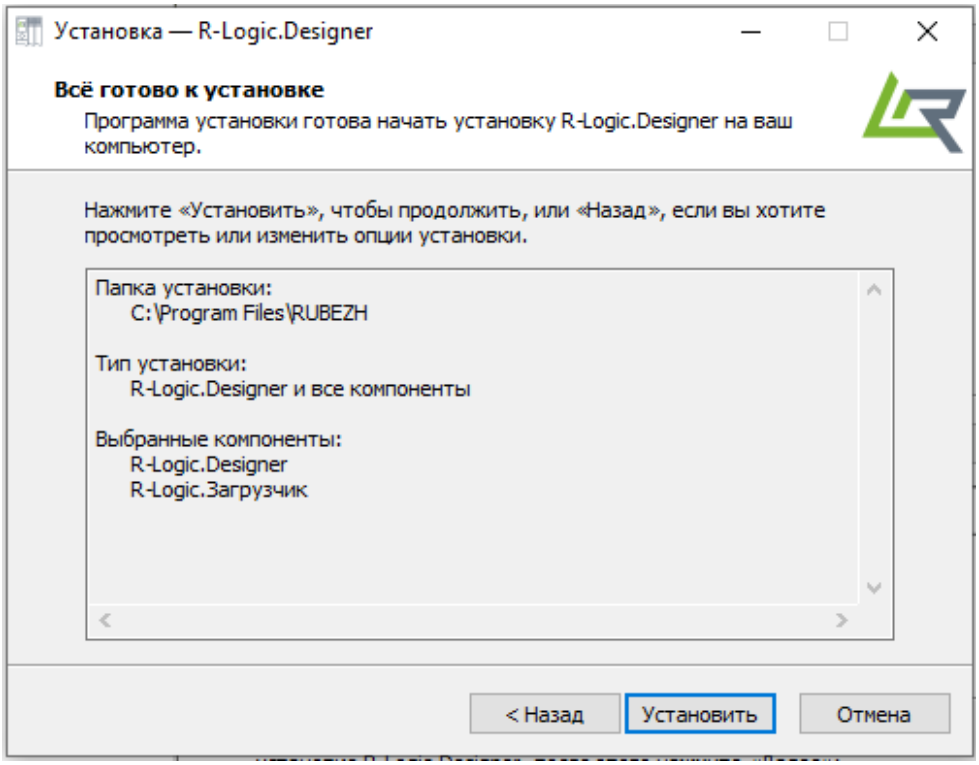

Рисунок 6 – Мастер установки R-Logic.Designer (шаг 4)

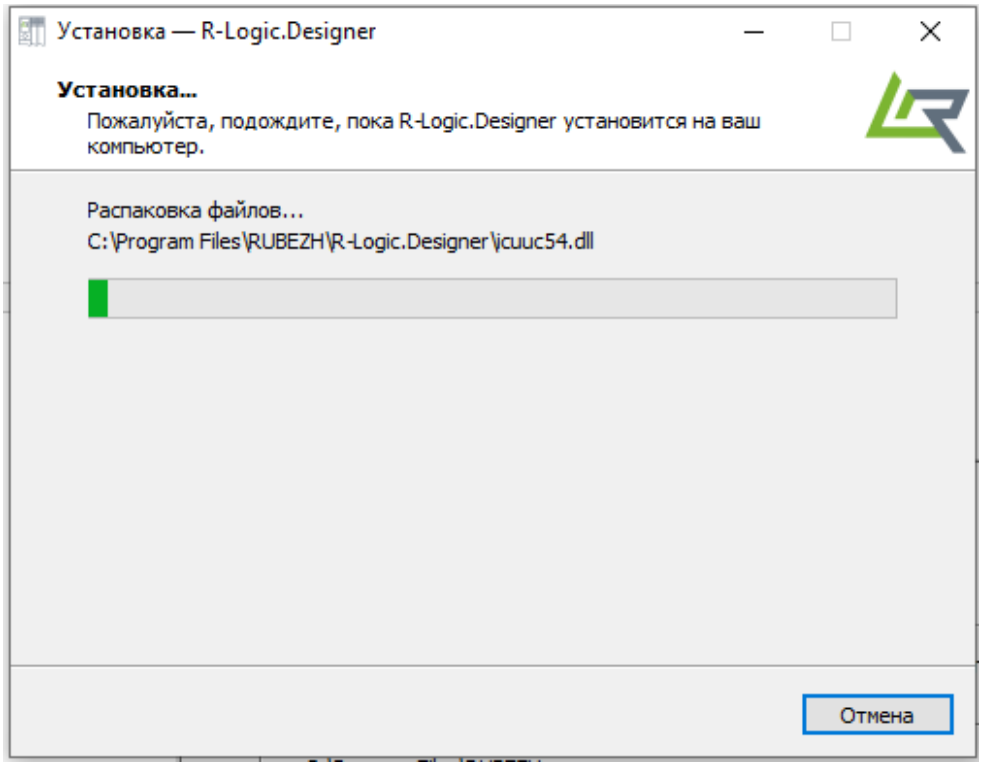

Рисунок 7 – Мастер установки R-Logic.Designer (шаг 5)

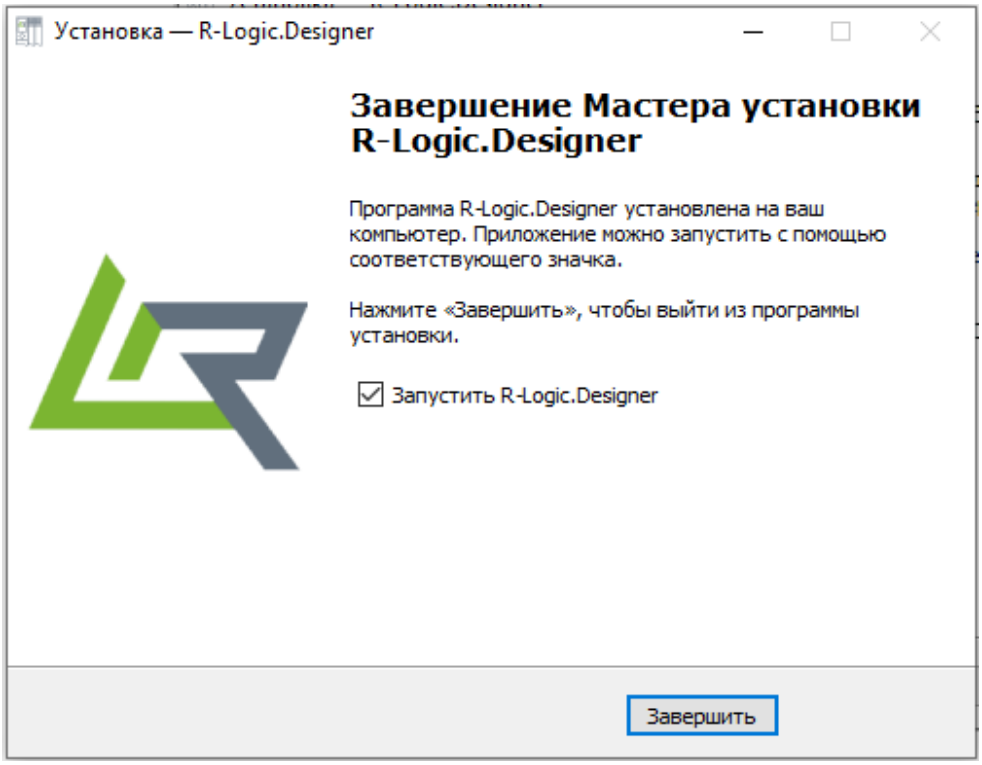

Рисунок 8 – Завершение установки R-Logic.Designer

2.2 Установка на Linux

2.2.1 Подготовка файлов

Скопируйте файлы поставки в домашний каталог пользователя:

```
$ cp ./r-logic.designer*.rpm ~

$ cp ./r-logic.designer_*_platform_pack.tar.xz ~

$ cp ./r-logic.designer_platform_pack_basic.tar.xz ~
```

2.2.2 Удаление ранее установленной версии (при наличии)

Перед обновлением удалите старую версию пакета и каталог в `/opt`:

```
$ sudo rpm -e r-logic.designer

$ sudo rm -rf /opt/R-Logic.Designer
```

2.2.3 Установка

Распакуйте платформенные архивы в корень файловой системы и установите RPM-пакет:

```
$ sudo tar -xvJf ~/r-logic.designer_*_platform_pack.tar.xz -C /

$ sudo tar -xvJf ~/r-logic.designer_platform_pack_basic.tar.xz -C /
```

```
$ sudo apt-get install ~/r-logic.designer-*.rpm
```

2.2.4 Проверка установки

Проверьте версию платформенного пакета:

```
$ cat /opt/R-Logic.Designer/builder2/version.txt
```

3 Общая информация

Среда разработки представляет собой кроссплатформенное приложение с оконным интерфейсом, запускаемое на ПК.

Среда R-Logic.Designer выполняет следующие функции:

- разработка программного обеспечения с использованием языков программирования, описанных в стандарте МЭК 61131-3 (ST, FBD, LD, SFC);
- задание типа и состава оборудования, предназначенного для загрузки и выполнения проекта;
- задание стартовой конфигурации модулей и портов ввода-вывода;
- задание периода исполнения задач;
- мониторинг текущего состояния переменных и портов ввода-вывода;
- конфигурирование коммуникационных интерфейсов;
- конфигурирование информационных протоколов, используемых для передачи данных.

Вся информация проекта сохраняется в одном файле формата XML согласно документу PLCopen Technical Committee 6 XML Formats for IEC 61131-3.

Дополнительно файл проекта содержит следующую информацию:

- конфигурация аппаратного обеспечения;
- конфигурация различных служб.

4 Основное окно программы

Ниже представлена стандартная комбинация окон и вкладок (рисунок 9). Для удобства работы с ПО она может быть изменена пользователем.

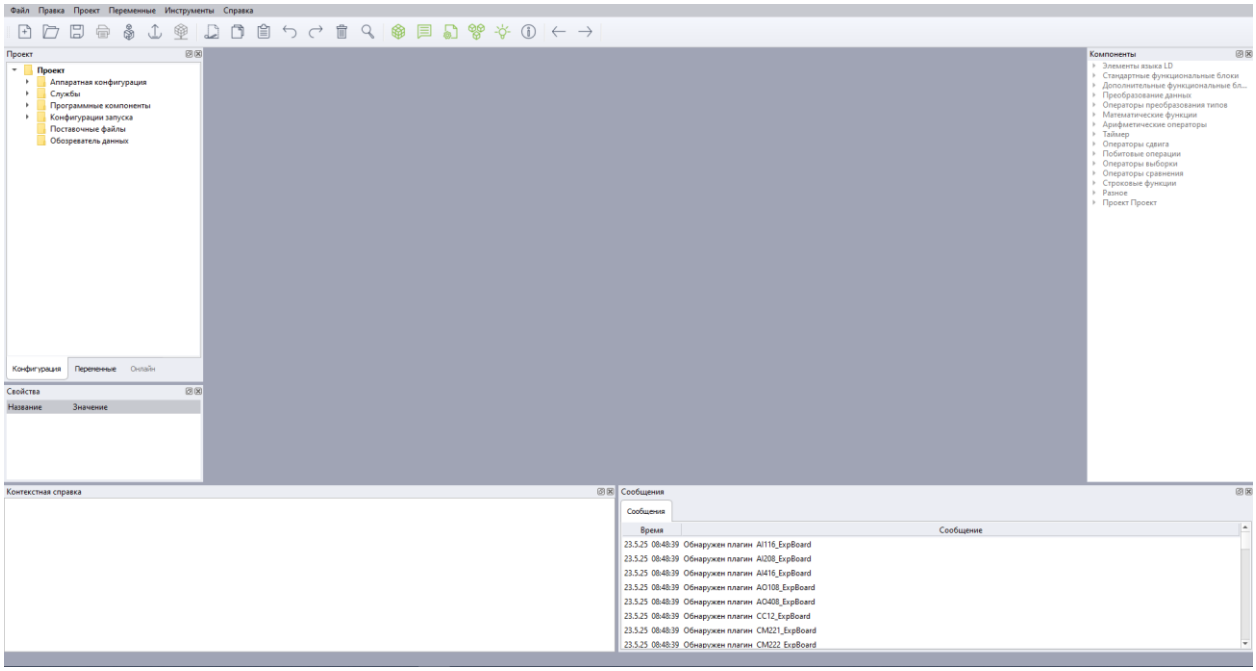

Рисунок 9 – Основное окно программы

4.1 Меню программы

В верхней части основной рабочей области находится главное меню (рисунок 10).

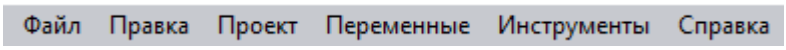

Рисунок 10 – Главное меню программы

Меню программы состоит из:

- меню «Файл»;
- меню «Правка»;
- меню «Проект»;
- меню «Переменные»;
- меню «Инструменты»;
- меню «Справка».

Пункты меню могут быть вызваны с помощью мыши или с помощью «горячих клавиш». Ниже представлено описание пунктов меню и комбинации клавиш.

4.1.1 Меню «Файл»

Меню «Файл» предназначено для работы с проектом (рисунок 11).

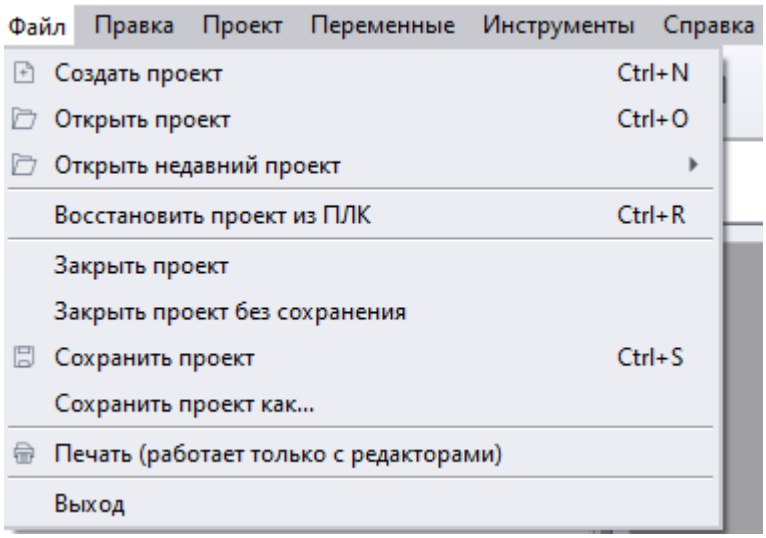

Рисунок 11 – Внешний вид меню «Файл»

Меню «Файл» содержит следующие пункты:

- «Создать проект» – создание нового проекта (Ctrl + N);
- «Открыть проект» – открытие существующего проекта (Ctrl + O);
- «Открыть недавний проект» – быстрое открытие одного из десяти последних сохраненных проектов;
- «Восстановить проект из ПЛК» – считывание ранее загруженного на ПЛК проекта (Ctrl + R).

Окно, открывающееся при выборе функции «Восстановить проект из ПЛК», содержит поля для ввода IP-адреса ПЛК, авторизации пользователя, а также для задания названия и выбора пути сохранения формируемого проекта (рисунок 12).

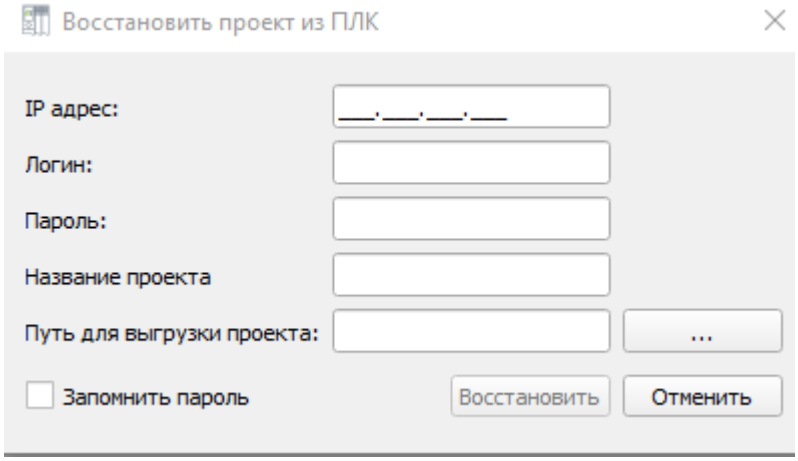

Рисунок 12 – Восстановление проекта из ПЛК

Данные о начале, прерывании, ошибках и завершении процесса восстановления проекта отображаются в окне «Сообщения»:

- «Закрыть проект» – закрытие текущего открытого проекта;
- «Закрыть проект без сохранения»;

- «Сохранить проект» – сохранение текущего проекта (Ctrl + S);
- «Сохранить проект как…» – сохранение текущего проекта с выбором папки и названия файла;
- «Печать» – печать активной программы на принтере (работает только с редакторами);
- «Выход» – закрытие текущего проекта и выход из среды R-Logic.Designer.

4.1.2 Меню «Правка»

Меню «Правка» предназначено для работы с редакторами языков стандарта МЭК 61131-3 (рисунок 13).

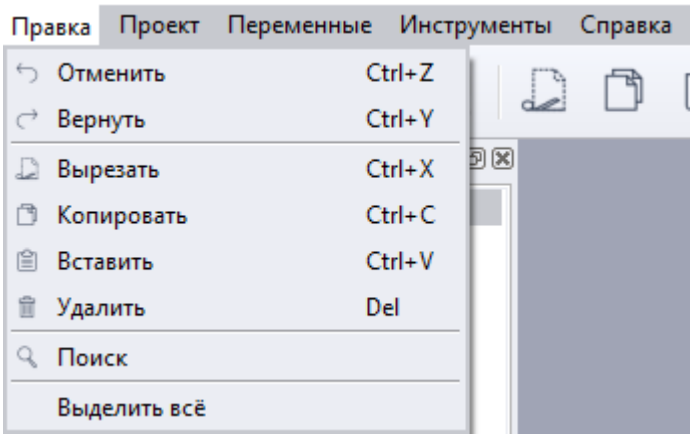

Рисунок 13 – Внешний вид меню «Правка»

Меню «Правка» содержит следующие пункты

- «Отменить» – отмена последнего действия в редакторе (до 50 действий) (Ctrl + Z);
- «Вернуть» – возврат отмененного действия (Ctrl + Y);
- «Вырезать» – перемещение выделенного объекта в буфер обмена (Ctrl + X);
- «Копировать» – копирование выделенного объекта в буфер обмена (Ctrl + C);
- «Вставить» – вставка объекта из буфера обмена в редактор (Ctrl + V);
- «Удалить» – удаление выделенного объекта (Del);
- «Поиск» – вызов окна поиска данных в проекте;
- «Выделить все» – выделение всех элементов в активной вкладке редактора.

В диалоговом окне поиска (рисунок 14) находится выпадающий список «Область», в котором перечислены области поиска:

- «Проект» – поиск в окне «Проект», во вкладке «Конфигурация»;
- «Переменные» – поиск в окне «Проект», во вкладке «Переменные»;
- «Активное окно» – поиск в окне, с которым в данный момент взаимодействует пользователь.

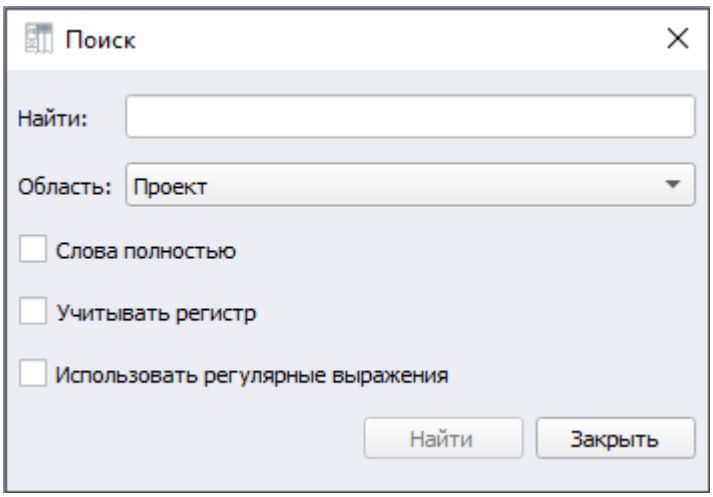

Рисунок 14 – Поиск

Также окно поиска позволяет сузить выборку результатов флажками точного совпадения слов, учета регистра и использования регулярных выражений. Направление поиска осуществляется сверху вниз, а найденный объект выделяется функционалом соответствующего ему окна (рисунок 15). Для перехода между результатами используются кнопки «Найти далее» (F3) и «Найти предыдущее» (Shift + F3). Начало и завершение поиска отображается в окне «Сообщения».

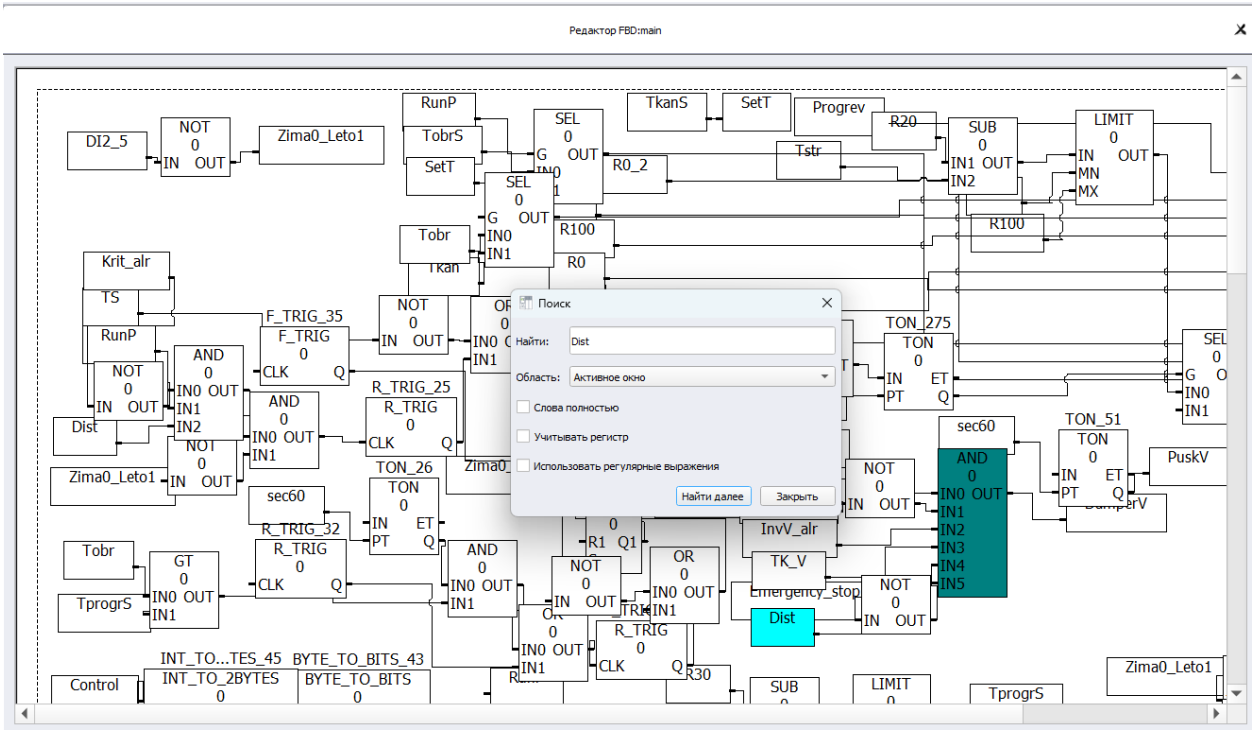

Рисунок 15 – Результат поиска, выделенный голубым цветом

4.1.3 Меню «Проект»

Меню «Проект» предназначено для работы с редакторами языков стандарта МЭК 61131-3 (рисунок 16).

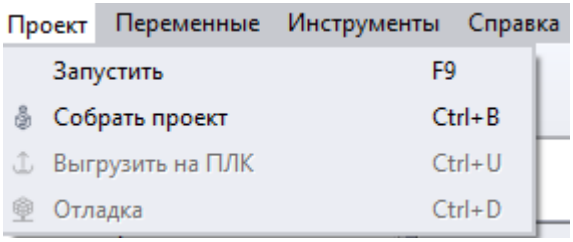

Рисунок 16 – Внешний вид меню «Проект»

Меню «Проект» содержит следующие пункты:

- «Запустить» – запуск последовательно (в случае успешного завершения предыдущего) процессов проверки проекта, сборки проекта, подключения к контроллеру, загрузки проекта и отладки проекта (F9);
- «Собрать проект» – запуск процесса трансляции программных блоков, составленных на языках программирования, в бинарный код, а также проверка правильности трансляции (Ctrl + B);
- «Выгрузить на ПЛК» – загрузка собранного проекта на ПЛК (Ctrl + U);
- «Отладка» – перевод программы в режим «Онлайн» для работы с ПЛК в режиме реального времени (Ctrl + D).

4.1.4 Меню «Инструменты»

Меню «Инструменты» предназначено для настройки программы и настройки отображения окон (рисунок 17).

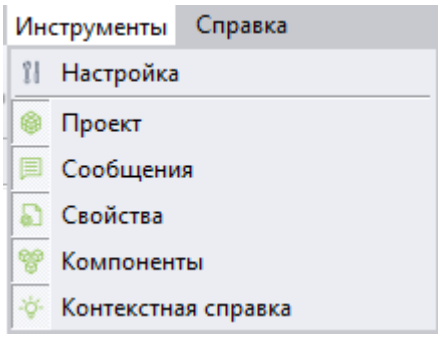

Рисунок 17 – Внешний вид меню «Инструменты»

Меню «Инструменты» позволяет управлять видимостью соответствующих окон.

Пункт верхнего «Настройка» вызывает диалоговое окно настроек программы, содержащее:

- вкладку «Настройки среды» с полем «Каталог проектов» и кнопкой выбора пути сохранения проекта;
- вкладку «Настройки доступа» с файлом сертификата и кнопкой выбора файла сертификата.

4.1.5 Меню «Справка»

Меню «Справка» предназначено для вывода информации о программе (рисунок 18).

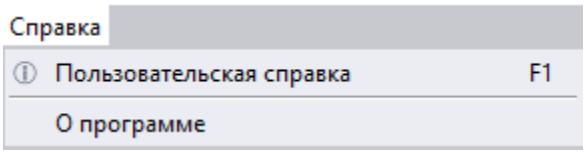

Рисунок 18 – Внешний вид меню «Справка»

Меню «Справка» содержит следующие пункты:

- «Пользовательская справка» – информация о функциях и инструментах R-Logic.Designer (F1);
- «О программе» – информация о версии программы, а также загруженных модулях.

В правом нижнем углу окна пользовательской справки расположены кнопки «Вперед» и «Назад», предназначенные для переключения между открытыми страницами справки согласно истории просмотра. Пользователь также может масштабировать текст, нажав клавишу Ctrl и прокрутив колесико мыши.

4.2 Панель инструментов

Под верхним меню находится панель инструментов (рисунок 19), предназначенная для быстрого вызова команд меню, а также перехода между диалоговыми окнами.

Рисунок 19 – Внешний вид панели инструментов

- «Собрать проект» – запуск процесса трансляции программных блоков, составленных на языках программирования, в бинарный код, а также проверка правильности трансляции (Ctrl + B);
- «Выгрузить на ПЛК» – загрузка собранного проекта на ПЛК (Ctrl + U);
- «Отладка» – перевод программы в режим «Онлайн» для работы с ПЛК в режиме реального времени (Ctrl + D).

Ниже приведено описание функций кнопок панели инструментов (таблица 1).

Таблица 1 – Кнопки панели инструментов

Значок	Название	Функция
	Создать проект	Создать новый проект
	Открыть проект	Открыть существующий проект
	Сохранить проект	Сохранить текущий проект

Значок	Название	Функция
	Печать	Печать текущей программы на принтере
	Собрать проект	Запустить трансляцию программных блоков на инженерных языках в бинарный код; проверить правильность трансляции
	Выгрузить на ПЛК	Загрузить собранный проект на ПЛК
	Отладка	Перевести программу в режим «Онлайн» для работы с ПЛК в режиме реального времени
	Вырезать	Переместить выделенный объект в буфер обмена
	Копировать	Копировать выделенный объект в буфер обмена
	Вставить	Вставить объект из буфера обмена в редактор
	Отменить	Отменить последнее действие в редакторе
	Вернуть	Вернуть отмененное действие в редакторе
	Удалить	Удалить выделенный объект
	Поиск	Вызвать окно поиска данных в проекте
	Проект	Показать/скрыть окно «Проект»
	Сообщения	Показать/скрыть окно «Сообщения»
	Свойства	Показать/скрыть окно «Свойства»
	Компоненты	Показать/скрыть окно «Компоненты»
	Контекстная справка	Показать/скрыть окно «Контекстная справка»

Значок	Название	Функция
	Пользовательская справка	Показать окно «Пользовательская справка»
	Назад	Переход между окнами редактора или диалогами слева направо
	Вперед	Переход между окнами редактора или диалогами справа налево

4.3 Окно проекта

По умолчанию в левой части рабочей области расположено окно «Проект» (рисунок 20). Во вкладке «Конфигурация» содержится дерево проекта, отображающее компоненты (модули, функции, функциональные блоки и др.). Дерево проекта позволяет добавлять и удалять элементы.

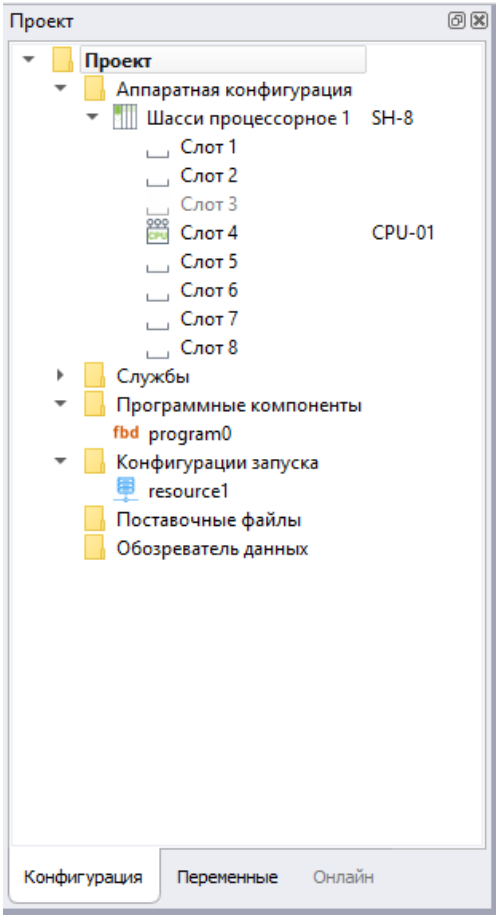

Рисунок 20 – Внешний вид окна «Проект»

Вкладка «Переменные» содержит дерево переменных, представленных в группах «Задачи» и «Программные блоки», которые могут быть выбраны из выпадающего списка фильтрации. Для структур во вкладке «Переменные» представлен узел «Поля».

Вкладка «Онлайн» по умолчанию неактивна и открывается автоматически после перехода в режим «Отладка».

4.4 Окно свойств

По умолчанию окно «Свойства» (рисунок 21) находится в левой части рабочей области приложения под окном «Проект». В нем отображается краткая информациях о свойствах (параметрах) объектов среды разработки R-Logic.Designer с возможностью их изменения после щелчка по соответствующей ячейке.

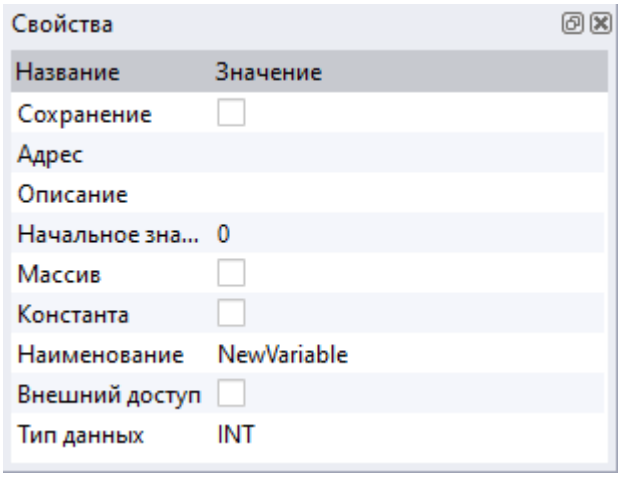

Рисунок 21 – Внешний вид окна «Свойства»

Окно свойств может быть открыто также щелчком по объекту или выбором пункта «Свойства» в меню или через панель инструментов.

4.5 Окно контекстной справки

Окно «Контекстная справка» (рисунок 22), находящееся по умолчанию в левой нижней части рабочей области, отражает информацию из пользовательской справки. Окно контекстной справки может быть открыто нажатием на соответствующий значок на панели инструментов.

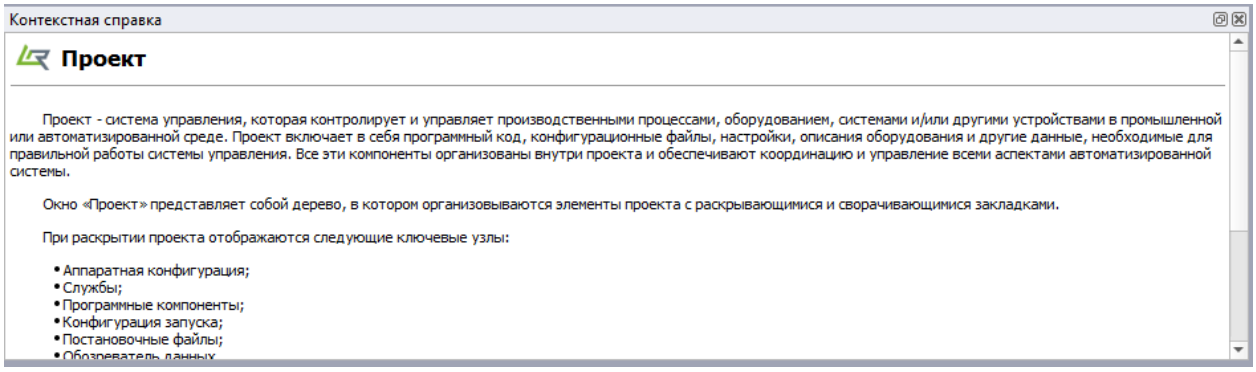

Рисунок 22 – Внешний вид окна «Контекстная справка»

При нажатии на элемент программы в окне «Контекстная справка» отображается информация о нем.

4.6 Окно сообщений

По умолчанию в правой нижней части рабочей области программы находится окно «Сообщения», где отображаются сообщения о запущенных и завершенных процессах в R-Logic.Designer (рисунок 23).

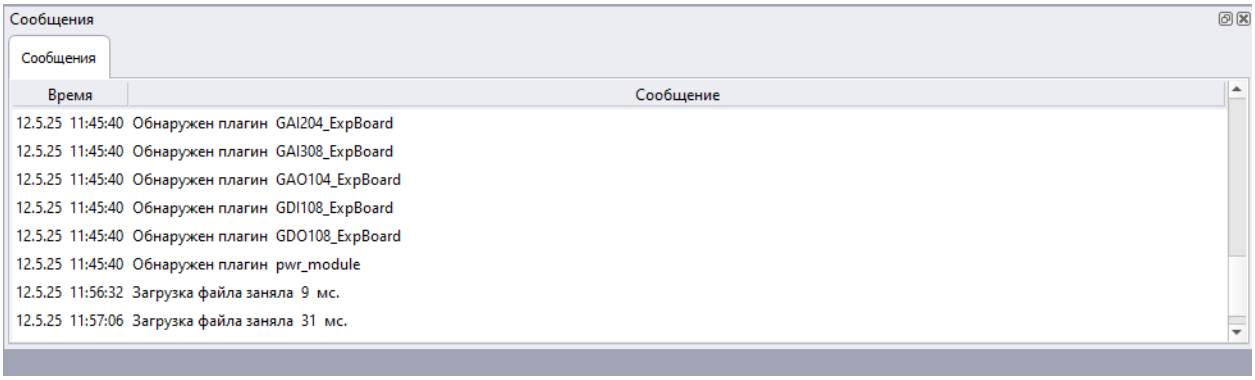

Рисунок 23 – Внешний вид окна «Сообщения»

Во вкладке «Сообщения» представлены сообщения о выполнении фоновых действий. Другие вкладки появляются в окне после появления соответствующих сообщений в процессе работ.

Во вкладке «Сообщения сборки» отображается информация о процессе сборки проекта.

Во вкладке «Отладочные сообщения» отображаются сообщения об обнаружении устройства и подключении к нему, о запуске процесса отладки, изменении значений каналов модулей ввода-вывода, изменении значений переменных.

При невозможности подключения к контроллеру в столбце «Сообщение» отображается текст: «Невозможно подключиться к контроллеру! Ошибка: “…”».

4.7 Библиотека компонентов

Библиотека компонентов (рисунок 24) по умолчанию находится в правой части рабочей области и содержит предварительно сконфигурированные шаблоны объектов: команды релейно-контактной логики, элементы FBD, LD, SFC, ST и др., которые можно добавить в проект.

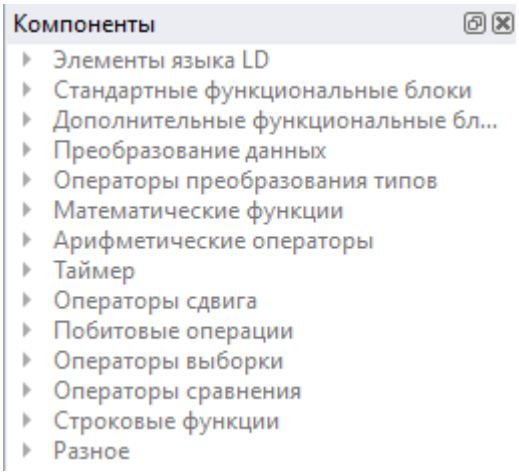

Рисунок 24 – Внешний вид окна «Компоненты»

Созданные программные блоки отображаются в узле «Программные компоненты» в окне проекта, а также в окне «Компоненты», где они сгруппированы по имени проекта с возможностью поиска по названию.

В окне «Компоненты» отображаются только те компоненты, которые соответствуют типу открытого в редакторе программного блока и могут быть в него добавлены. Для этого выделенный компонент переносится в рабочее окно редактора.

В библиотеке компонентов объекты представлены текстовыми названиями, которые при переносе и установке в рабочее окно редактора преобразуются в графическое изображение объекта.

Параметры выбранного содержимого редактора FBD отображаются в окне «Свойства» (таблица 2).

Таблица 2 – Свойства компонентов

Название	Описание
Наименование	Имя компонента
Имя экземпляра	Имя (только для функциональных блоков)
Вход/Выход контроля выполнения	Добавление входа EN и выхода ENO: EN (Enable) – включение/выключение блока – логический ноль на входе EN запрещает вызов блока. Если на EN подаётся логическая единица, блок будет вызван и выполнен; ENO (Enable Out) – индикация успешного исполнения – если блок не выполнен, ENO принимает значение логический ноль, что прерывает выполнение цепи. Если блок выполнился без ошибок, ENO принимает значение логической единицы.

Название	Описание
Номер выполнения	Порядок запуска прикладного ПО начинается с 1 и задается вручную или автоматически при добавлении объекта
Количество входов	Задается вручную, доступен только для некоторых компонентов редактора FBD

5 Добавление и удаление элементов дерева проекта

Пользователь может добавить новые элементы в дерево в окне «Проект» с помощью контекстного меню, вызываемого нажатием ПКМ. Названия созданных элементов могут состоять только из латиницы, цифр и символа «_».

5.1 Контекстное меню добавления шасси

С помощью контекстного меню в узел «Аппаратная конфигурация» могут быть добавлены одно или несколько шасси как показано на рисунке 25.

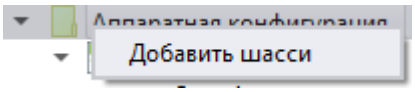

Рисунок 25 – Контекстное меню добавления шасси

5.2 Контекстное меню добавления модулей

С помощью контекстного меню в слоты могут быть добавлены модули различных видов (рисунок 26).

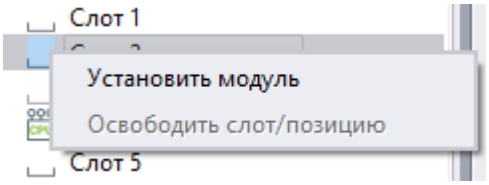

Рисунок 26 – Контекстное меню добавления модуля

Пользователь также может размещать модули двойным щелчком по соответствующему слоту.

5.3 Контекстное меню добавления программных компонентов

С помощью контекстного меню в узел «Программные компоненты» могут быть добавлены функции, функциональные блоки, программы (рисунок 27).

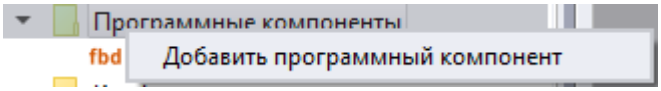

Рисунок 27 – Контекстное меню добавления программного компонента

5.4 Контекстное меню добавления служб

С помощью контекстного меню в узел «Службы» могут быть добавлены службы (рисунок 28).

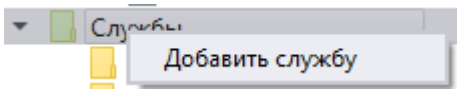

Рисунок 28 – Контекстное меню добавления службы

5.5 Удаление элементов из дерева проекта

Пользователь может удалить элемент дерева проекта, вызвав контекстное меню щелчком ПКМ и выбрав команду «Удалить» (рисунок 29) (для отключения модуля – команду «Освободить слот/позицию» (рисунок 30)).

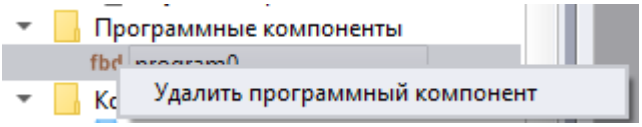

Рисунок 29 – Контекстное меню удаления программного компонента

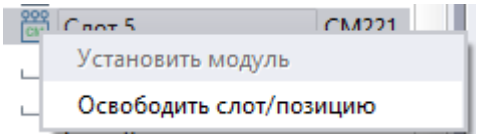

Рисунок 30 – Контекстное меню отключения модуля

6 Работа с проектом

В разделе описаны основные этапы работы в R-Logic.Designer, которые необходимы для создания прикладной программы: создание проекта, настройка аппаратной конфигурации, настройка модулей, добавление программных компонентов, переменных, служб и др.

6.1 Создание нового проекта

Пользователь может создать новый проект с помощью функции главного меню «Файл» – «Создать проект» (рисунок 31) или с помощью соответствующей кнопки на панели управления.

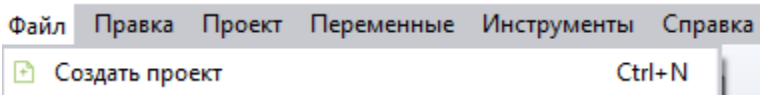

Рисунок 31 – Вызов меню «Создать проект»

Затем открывается диалоговое окно, позволяющее задать название, шаблон и каталог (расположение папки) проекта (рисунок 32).

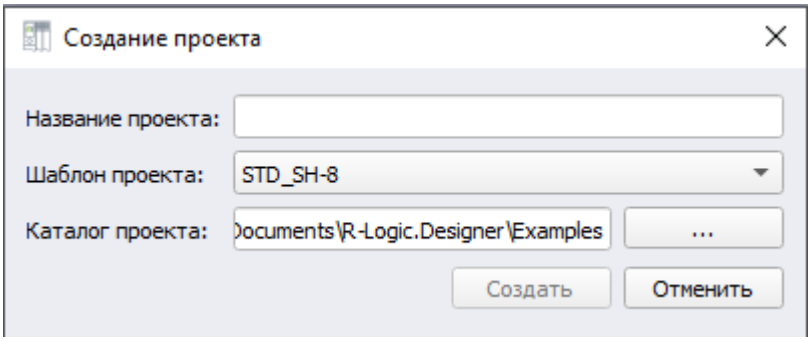

Рисунок 32 – Создание проекта

Шаблоны проекта включают варианты исполнения ПЛК:

- STD_SH-8 – исполнение «Стандарт» с шасси на 8 слотов;
- STD_SH-16 – исполнение «Стандарт» с шасси на 16 слотов;
- Lite – исполнение «Лайт».

При выборе шасси в соответствующем окне отображаются его свойства (таблица 3).

Таблица 3 – Свойства шасси

Название	Описание
Наименование	Условное обозначение шасси с номером (нередактируемое)
Количество слотов	Общее количество слотов, принадлежащее шасси
Резервированное	Установка флажка включает функцию резервирования

При создании ПЛК с несколькими шасси (максимальное допустимое количество – 10) в окне создания проекта необходимо выбрать тип процессорного шасси.

После нажатия кнопки «Создать» свернутое по умолчанию дерево проекта с заданным именем верхнего элемента отображается в окне «Проект» (рисунок 33).

Рисунок 33 – Новый проект программы

6.2 Открытие существующего проекта

Открытие ранее созданного проекта производится с помощью функции главного меню «Файл» – «Открыть проект» (рисунок 34) или соответствующей кнопки панели управления.

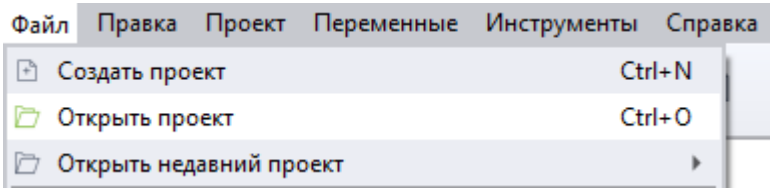

Рисунок 34 – Вызов меню «Открыть проект»

При наведении курсора на пункт меню «Открыть недавний проект» отображается список последних проектов пользователя.

6.3 Настройка резервирования

При включенном резервировании два модуля ЦПУ получают идентичные данные с модулей УСО, т.е. работают как один за счет синхронизации переменных. Управление выходными модулями осуществляется только ведущим ЦПУ. Передача контекста происходит через Ethernet. При этом службы обмена на ведущем и ведомом устройстве работают идентично.

Для настройки резервирования в свойствах процессорного шасси устанавливается флажок «Резервированное» (рисунок 35).

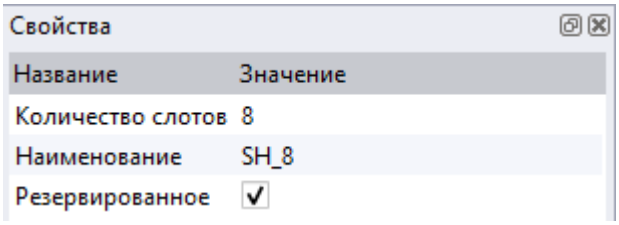

Рисунок 35 – Свойства шасси (выбрано «Резервированное»)

После этого в окне конфигурации модуля ЦПУ во вкладке «Настройка сети» появляется дополнительное поле настройки «ЦПУ Шасси 1Б» (см. п. 6.4.1). По умолчанию при резервировании ведущим является ЦПУ Шасси 1А, а Шасси 1Б – ведомым. Ведущий ЦПУ определяется модулем СР-12 (на Шасси 2) по «индексу здоровья», при этом приоритет

находится у ЦПУ Шасси А. Между ЦПУ А и ЦПУ Б происходит синхронизация значений всех переменных, помеченных в свойствах как внешние (рисунок 36).

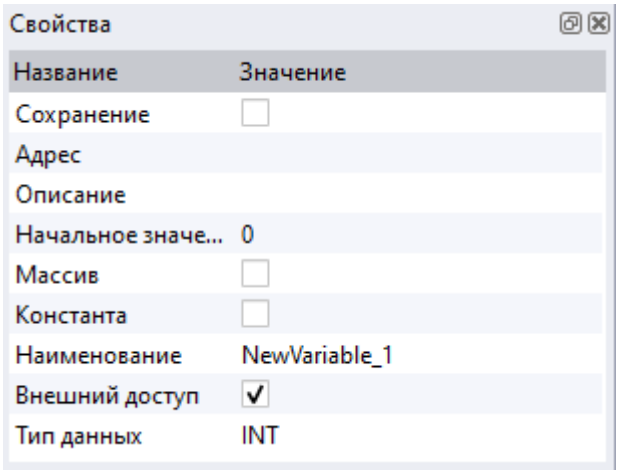

Рисунок 36 – Задание внешнего доступа к переменной

Синхронизация происходит с циклом порядка 20 мс.

Прикладная программа не имеет информации о том, является ли исполняющий ее ПЛК ведущим или ведомым. Роль модулей идентифицируется по параметру модуля ЦПУ «Ведущий А», который равен 1, если ведущий А, и 2, если ведущий Б. Параметр «Обмен контекстом» имеет значение TRUE при успешном обмене контекстом (см. п. 6.9.2).

6.4 Настройка аппаратной конфигурации

Для установки модуля на шасси используется команда «Установить модуль» контекстного меню или двойной щелчок по соответствующему слоту. При этом открывается диалоговое окно (рисунок 37), где модули отображаются по категориям:

- базовые модули (модули питания, ЦПУ, коммуникационный процессор);
- модули ввода-вывода (аналогового и дискретного);
- коммуникационные модули (только для исполнения «Стандарт»).

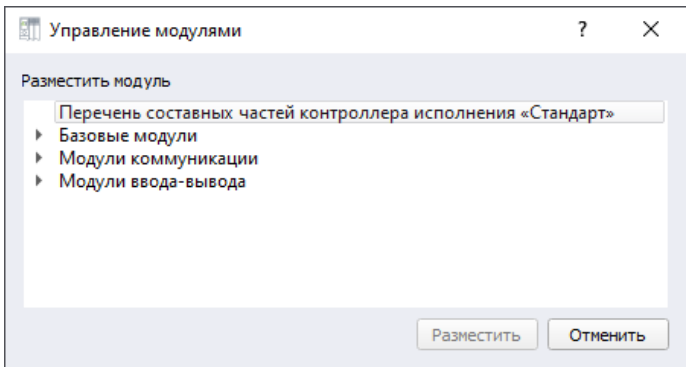

Рисунок 37 – Размещение модулей

После выбора модуля и нажатия кнопки «Разместить» напротив имени слота в окне проекта отобразится условное обозначение модуля. На ПЛК исполнения «Стандарт» адресация модулей определяется позицией на шасси. Для ПЛК исполнения «Лайт» при

сборке модулей на шасси необходимо производить конфигурацию DIP-переключателя (см. ТШВГ.421457.008 РЭ Контроллер автоматизации программируемый R-Logic Лайт. Руководство по эксплуатации).

Настройка модулей СР-12 производится в соответствии с документом ТШВГ.421457.001 РЭ Контроллер логический программируемый R-Logic Стандарт. Руководство по эксплуатации.

При выборе слота в соответствующем окне отображаются его свойства (таблица 4).

Таблица 4 – Свойства слотов

Название	Описание
Расположение	Расположение слота в контроллере
Наименование	Имя и номер слота (нередактируемое)
Модуль	Условное обозначение модуля, установленного в слоте

При добавлении действуют следующие ограничения:

- в шасси может быть установлен только один модуль питания PWR-01;
- PWR-01 занимает два слота и может быть установлен только в 1-й и 2-й слоты шасси. При его добавлении в дерево проекта слот 1 становится неактивным, а напротив слота 2 отображается условное обозначение модуля питания;
- в шасси может быть установлен только один модуль ЦПУ (CPU-01 или CPU-02 для исполнения «Стандарт», GCP-01 для исполнения «Лайт»);
- модули CPU-01 или CPU-02 могут быть установлены только в «Шасси процессорное 1»;
- модули CPU-01 или CPU-02 занимают по два слота и могут быть добавлены только в 3-й и 4-й слоты шасси. При добавлении модуля CPU-01 или CPU-02 в дерево проекта слот 3 становится неактивным, а напротив слота 4 отображается условное обозначение модуля ЦПУ;
- модуль СР-12 может быть расположен только в слоте 5 (при схеме без резервирования);
- при резервировании два модуля СР-12 располагаются на дополнительных шасси в слотах 5 и 6;
- модуль GCP-01 по умолчанию расположен на позиции 1;
- модуль GCP-01 не может быть расположен на позиции 32.

Настройка установленных модулей производится в соответствующих окнах конфигурации. Для этого пользователь дважды щелкает по нужному слоту.

6.4.1 Конфигуратор ЦПУ

При выборе модуля ЦПУ его свойства отображаются в соответствующем окне (таблица 5).

Таблица 5 – Свойства модуля ЦПУ

Название	Описание
Название CPU	Имя процессора (нередактируемое)
Расположение	Локация слота в контроллере
Модуль	Условное обозначение модуля ЦПУ, установленного в слоте

Окно конфигуратора ЦПУ (рисунок 38) открывается через дерево проекта двойным щелчком по соответствующему слоту в узле «Аппаратная конфигурация».

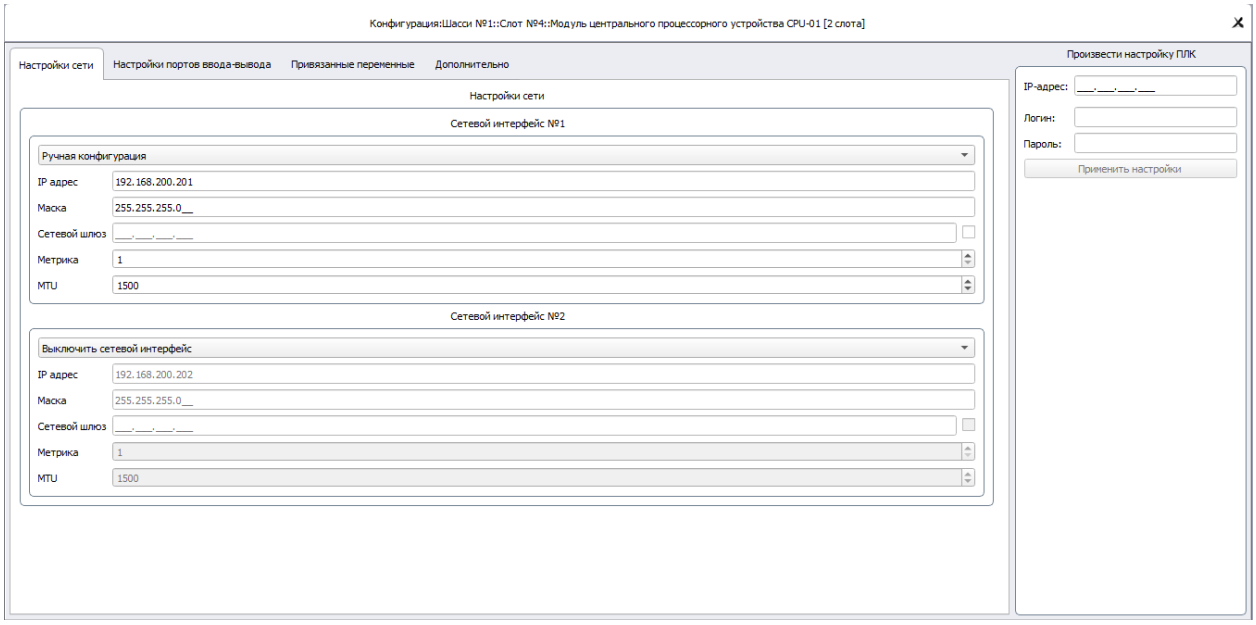

Рисунок 38 – Окно конфигурации ЦПУ

Во вкладке «Настройка сети» представлены настройки двух сетевых интерфейсов. При активном параметре «Резервированное» в свойствах процессорного шасси вкладка содержит панели «ЦПУ Шасси 1А» и «ЦПУ Шасси 1Б» для настройки параметров обоих ЦПУ.

У каждого сетевого интерфейса есть выпадающий список, который содержит конфигурации:

- Ручная конфигурация – позволяет вручную заполнять поля сетевых настроек;
- Выключить сетевой интерфейс;
- Автоматическая конфигурация (DHCP) – автоматическая настройка сетевого интерфейса, при этом поля параметров сетевого интерфейса становятся неактивны.

По умолчанию настройки ЦПУ следующие:

Сетевой интерфейс №1:

- Ручная конфигурация;
- IP-адрес – 192.168.200.201;

- Маска – 255.255.255.0;
- Сетевой шлюз не указан;
- Метрика – 1;
- MTU – 1500.

Сетевой интерфейс №2: Интерфейс отключен.

ВАЖНО! Для сохранения изменений параметров NTP сервера и сетевых интерфейсов в правой части окна конфигуратора («Произвести настройку ПЛК») пользователю необходимо ввести текущий IP-адрес ПЛК, а также логин и пароль для авторизации, затем нажать «Применить настройки». В противном случае новые настройки **не будут** применены на ПЛК.

Во вкладке «Настройка портов ввода-вывода» находятся каналы, разделенные на группы «Дискретные выходы» и «Дискретные входы», у каждого канала в названии находится номер и обозначение входа/выхода.

Окно позволяет настроить для каналов дискретного ввода значение фильтра в мс (от 0 (отключен) до 250).

Во вкладке «Привязанные переменные» информация отображается в табличном виде со столбцами «Задача», «Переменная», «Тип», «Канал», «ID пар.» и «Наименование параметра». В этой таблице приведены переменные, привязанные к модулю, в соответствующих столбцах указаны тип конкретной переменной, название программы, в которой она используется, и номер соответствующего канала.

Во вкладке «Дополнительно» находятся настройки NTP сервера в виде поля ввода «NTP сервер».

При активированной службе резервирования и совпадении IP-адресов ведущего и ведомого ЦПУ кнопка «Применить настройки» неактивна.

6.4.2 Конфигуратор модуля питания

Окно конфигуратора модуля питания (рисунок 39) открывается через дерево проекта двойным щелчком по соответствующему слоту в узле «Аппаратная конфигурация».

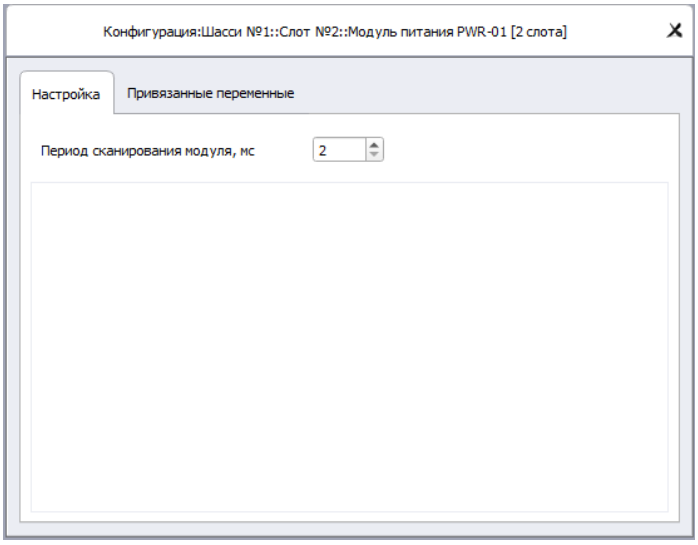

Рисунок 39 – Окно конфигурации модуля питания

Во вкладке «Настройки» пользователь может задать период сканирования модуля (от 2 мс).

Во вкладке «Привязанные переменные» информация отображается в табличном виде со столбцами «Задача», «Переменная», «Тип», «Канал», «ID пар.» и «Наименование параметра». В этой таблице приведены переменные, привязанные к модулю, в соответствующих столбцах указаны тип конкретной переменной, название программы, в которой она используется, и номер соответствующего канала.

6.4.3 Конфигураторы модулей ввода-вывода

Окно конфигуратора модуля ввода (вывода) (рисунок 40) открывается через дерево проекта двойным щелчком по соответствующему слоту в узле «Аппаратная конфигурация».

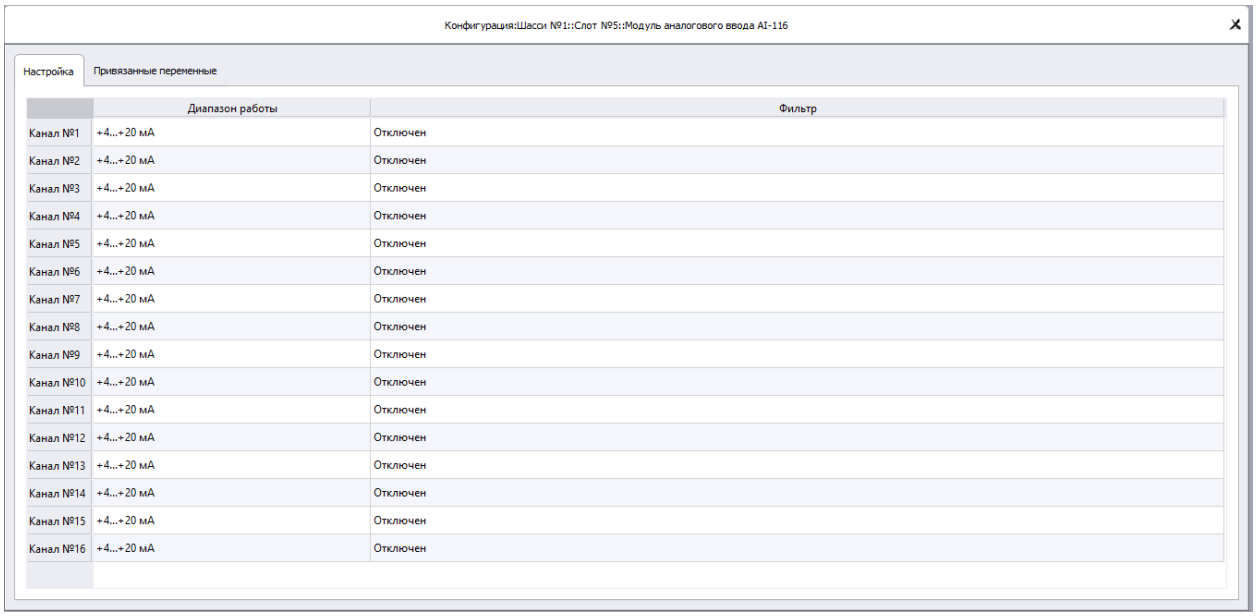

Рисунок 40 – Окно конфигурации модуля ввода (вывода)

Во вкладке «Настройка» отображается таблица со строками, обозначающими канал и его номер по порядку, и столбцами параметров данных каналов.

Настройки различаются в зависимости от типа модуля:

a) Модули аналогового ввода (AI-116, AI-416, AI-208, ...):

Доступен выбор типа измеряемого сигнала, фильтрации 50/60 Гц, а также выбор количества усредняемых измерений (опционально).

Для сигналов термопар доступно задание фиксированного значения компенсации температуры холодного спая.

b) Модули аналогового вывода (AO-108, AO-408, ...):

Доступен выбор диапазона генерируемого сигнала, а также действия при потере связи с ЦПУ – сохранить последнее заданное значение («Предыдущее») или установить фиксированное значение («Установить»).

c) Модули дискретного ввода (DI-124, GDI-104, …):

Доступен выбор размера фильтра от дребезга. Сигналы длительностью меньше, чем значение фильтра, будут отброшены модулем (контрольная лампа сигнала не загорится).

d) Модули дискретного вывода (DO-132, GDO-108, …):

Доступен выбор поведения при потере связи с ЦПУ – сохранить предыдущее значение модуля, перейти в состояние ВКЛ или в состояние ОТКЛ.

После двойного щелчка по ячейке в нужном столбце пользователь может задать доступное значение параметра из выпадающего списка.

Во вкладке «Привязанные переменные» информация отображается в табличном виде со столбцами «Задача», «Переменная», «Тип», «Канал», «ID пар.» и «Наименование параметра». В этой таблице приведены переменные, привязанные к модулю, в соответствующих столбцах указаны тип конкретной переменной, название программы, в которой она используется, и номер соответствующего канала.

6.4.4 Конфигураторы модулей коммуникации

Окно конфигуратора модуля коммуникации (рисунок 41) открывается через дерево проекта двойным щелчком по соответствующему слоту в узле «Аппаратная конфигурация».

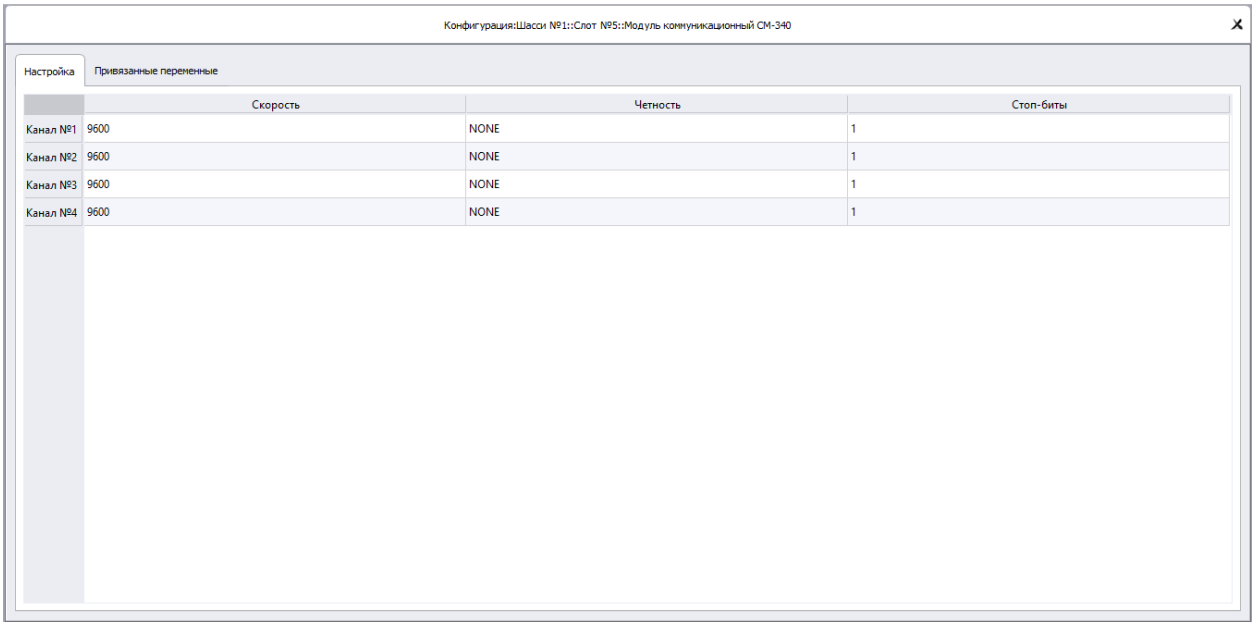

Рисунок 41 – Окно конфигурации модуля коммуникации

Для модулей СМ-340, СМ-341, СМ-342, СМ-440, СМ-441 окно конфигурации содержит вкладку «Привязанные переменные». Настройки портов модулей задаются соответствующими службами.

Настройки порта осуществляются при конфигурировании использующих их служб. Порты имеют следующие настраиваемые параметры:

- скорость (бод);
- четность;
- стоп-биты.

Параметр «Биты данных» всегда равен 8, не настраивается и не отображается.

Для модулей СС-12, СМ-221, СМ-222, СМ-223 вкладки «Настройка», «Привязанные переменные» и «Дополнительно» аналогичны вкладкам в окне конфигурации ЦПУ (см. п. 6.4.1).

Во вкладке «Привязанные переменные» информация отображается в табличном виде со столбцами «Задача», «Переменная», «Тип», «Канал», «ID пар.» и «Наименование параметра». В этой таблице приведены переменные, привязанные к модулю, в соответствующих столбцах указаны тип конкретной переменной, название программы, в которой она используется, и номер соответствующего канала.

6.5 Создание программных компонентов

При щелчке ПКМ по узлу «Программные компоненты» появляется команда «Добавить программный компонент», после нажатия на которую выводится диалоговое окно с выбором названия и типа программного блока (рисунок 42).

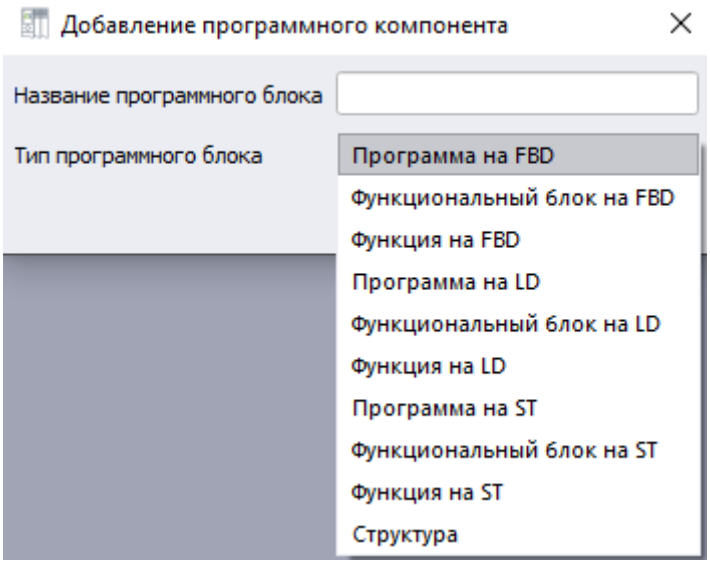

Рисунок 42 – Добавление программного компонента

Для создания программного компонента пользователь выбирает используемый язык (FBD, LD, ST, SFC) из выпадающего списка, задает название программного блока и нажимает кнопку «Добавить». Виды поддерживаемых программных компонентов включают программы, функции, функциональные блоки и структуры.

Созданный программный компонент отображается в дереве проекта и во вкладке «Переменные». Программный блок можно настраивать путем открытия соответствующего редактора по двойному щелчку мыши. Добавленные программы, функциональные блоки, функции и структуры отображаются также в библиотеке компонентов.

При выборе программного компонента в соответствующем окне отображаются его свойства (таблица 6).

Таблица 6 – Свойства программных компонентов

Название	Описание
Наименование	Имя программного компонента/службы
Модуль	Язык написания программного компонента

6.6 Создание и привязка переменных

Вкладка «Переменные» (рисунок 43) содержит дерево переменных, сгруппированных по программным компонентам. Переменные разделяются на следующие типы:

- локальные – переменные, область видимости которых ограничена блоком, в котором они созданы. Значения локальных переменных сохраняются между вызовами;

- разделяемые – ссылки для обращения к глобальным переменным с совпадающим именем и типом. Прочие параметры переменной для такой ссылки не используются;
- входные – переменные, доступные для записи вне блока, внутри блока недоступны для изменения;
- выходные – переменные, доступные для чтения вне блока;
- входные-выходные – переменные, доступные для чтения и записи вне блока.

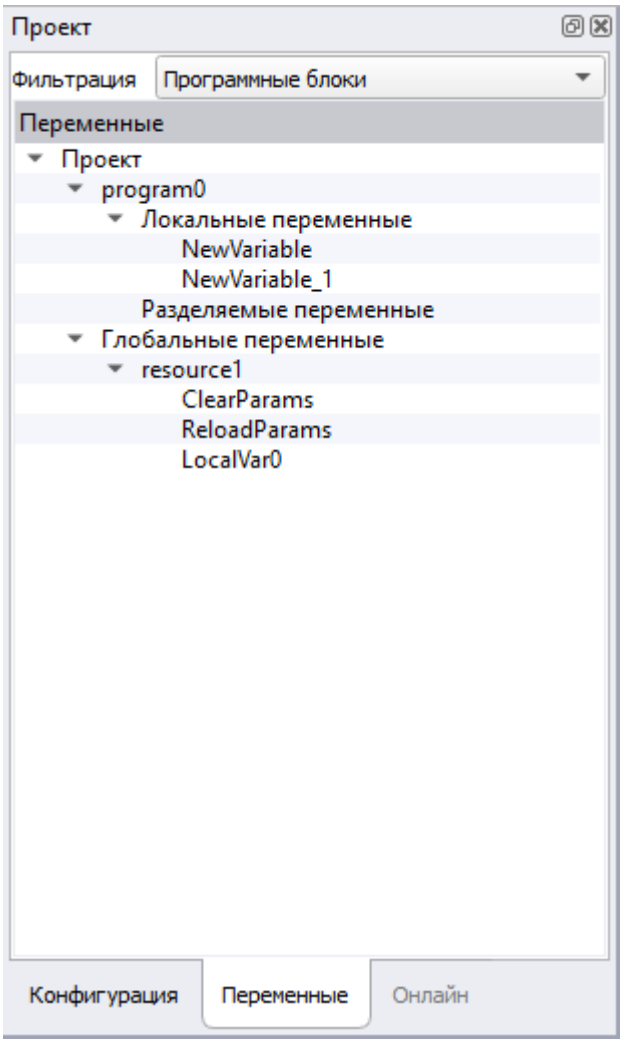

Рисунок 43 – Внешний вид вкладки «Переменные»

При выборе переменной ее свойства (таблица 7) отображаются в соответствующем окне.

Таблица 7 – Свойства переменных

Название	Описание
Сохранение	Активируется флажком для сохранения переменной в память контроллера. Значение сохраняется при «теплом» перезапуске ПЛК.
Адрес	Привязка переменной к системным переменным ПЛК, содержащим данные ввода-вывода и диагностики. Допустимо использовать только для локальных переменных программ или для глобальных переменных.
Описание	Информация о переменной
Начальное значение	Задается пользователем и присваивается переменной при инициализации, при отсутствии флага сохранения или при «холодном» запуске прикладной программы.
Массив	Активируется флажком для обозначения переменной как массива. При активации появляется поле «Количество элементов».
Константа	Обозначение переменной как константы
Наименование	Имя переменной
Внешний доступ	Отображение значения переменной в режиме отладки
Тип данных	Выбор типа данных переменной из списка
Количество элементов	Количество элементов в массиве (только при активации свойства «Массив»)

Добавление переменной осуществляется щелчком ПКМ по соответствующему узлу на вкладке «Переменные» (рисунок 44). При этом доступны команды:

- добавить переменную;
- вставить.

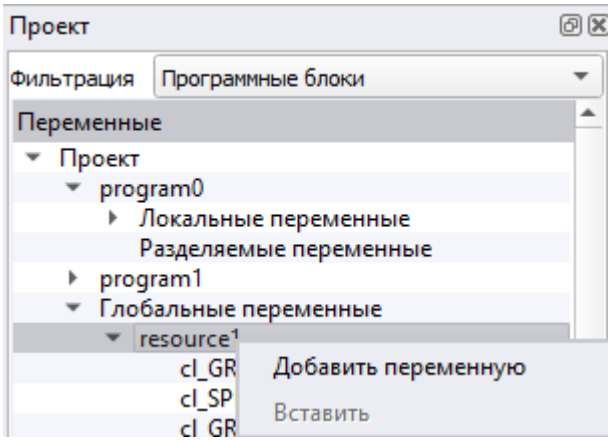

Рисунок 44 – Добавление переменной

Для переменных доступны следующие команды контекстного меню:

- дублировать;
- вырезать;
- копировать;
- удалить.

При создании переменных предусмотрен запрет присвоения одинаковых имен переменным, находящимся в одном и том же POU, а также запрет на использование зарезервированных служебных слов языков МЭК 61131-3 в качестве имен переменных. Если название уже существует в программном компоненте, к названию дубликатов добавляется порядковый номер.

Привязка переменных осуществляется нажатием на кнопку «…» в окне свойств переменной (рисунок 45) в строке «Адрес» (для отображения кнопки следует нажать на ячейку столбца «Значение»).

ВАЖНО! Привязка переменных допустима только для локальных компонентов POU типа «программа» и для глобальных переменных. Недопустимо выполнять привязки для переменных функций, функциональных блоков, структур.

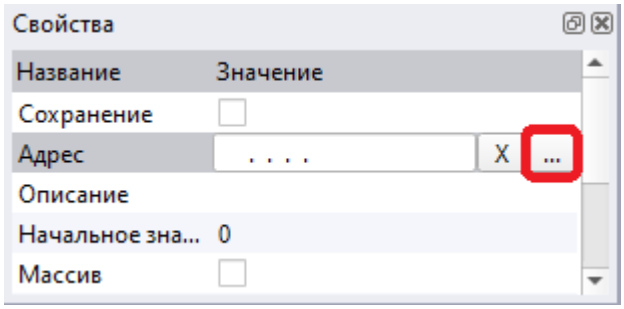

Рисунок 45 – Кнопка привязки переменных

При нажатии кнопки «…» открывается диалоговое окно «Выбор параметра», представленное в табличном виде (рисунок 46). Удаление привязанной переменной производится нажатием на кнопку «X».

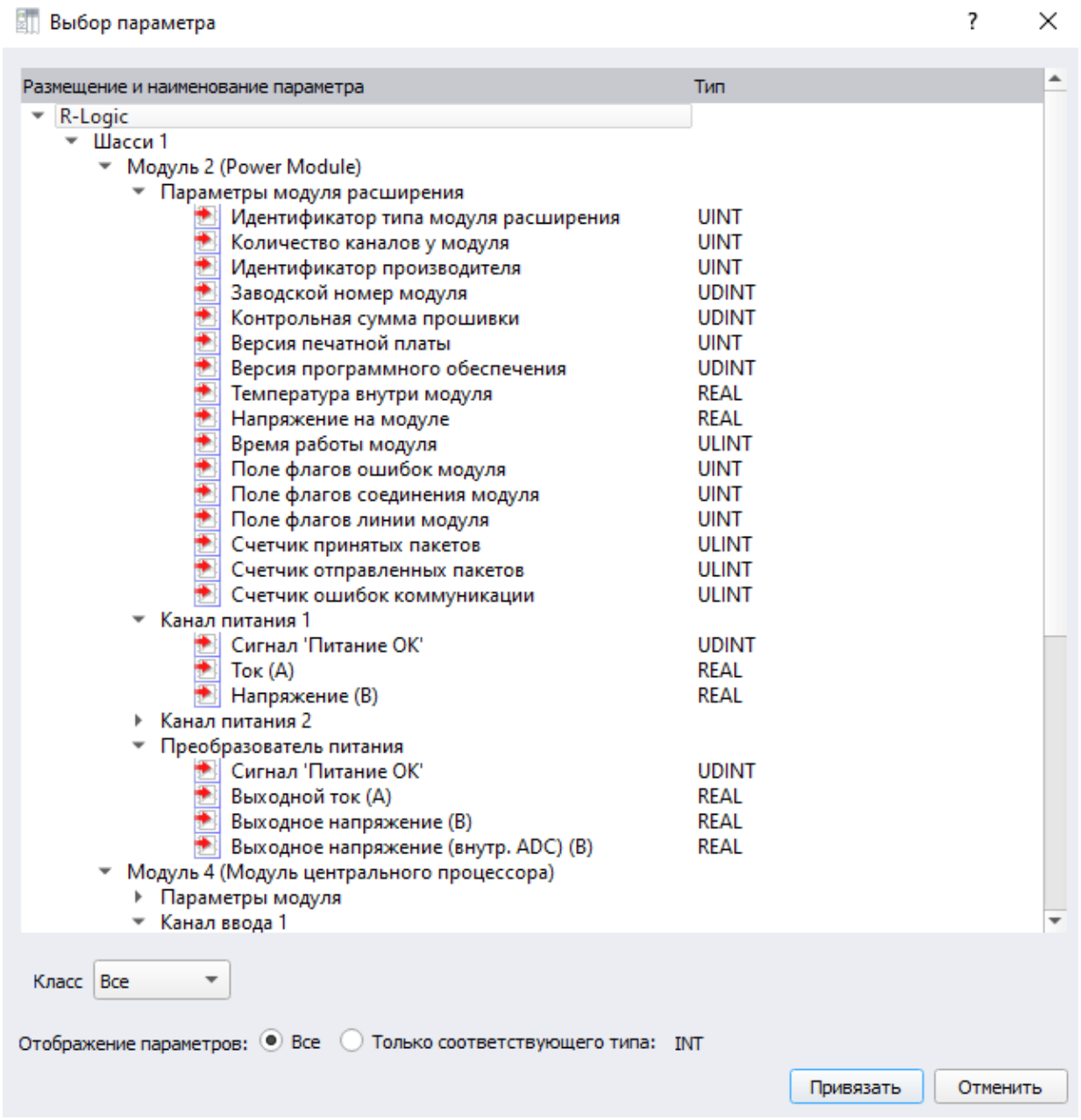

Рисунок 46 – Привязка переменных

В нижней части окна находятся кнопки «Привязать» и «Отмена» – подтверждение и отмена действий в окне соответственно, выбор типа отображаемых в окне параметров и выбор класса отображаемых параметров.

В окне под таблицей располагается выпадающий список «Класс», позволяющий выбрать параметры для отображения в табличной форме: «Все», «Вход», «Выход».

Под выпадающим списком «Класс» находится кнопка контроля типизации.

Контроль типизации происходит автоматически при привязке переменной и может быть отключен. Для этого в нижней части окна в пункте «Отображение параметров» пользователь может нажать переключатель «Все», отвечающую за отображение всех типов параметров (по умолчанию отображаются параметры только соответствующего типа).

При привязке переменной к объекту другого типа появляется запрос на изменение типа переменной после ее привязки к объекту, который пользователь может принять или отклонить.

6.6.1 Создание глобальных переменных

Глобальные переменные отображаются на вкладке «Переменные», на ветке «Глобальные переменные». Использование глобальных переменных осуществляется через разделяемые переменные (VAR EXTERNAL, МЭК 63131-3) по схеме, приведенной на рисунке 47. Для этого в POU создается переменная с идентичным именем и типом.

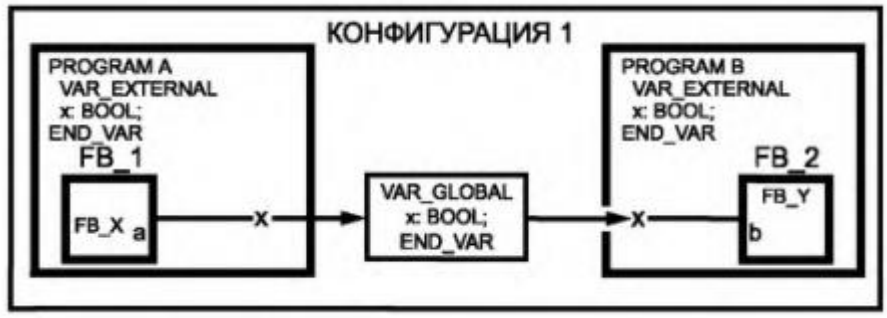

Рисунок 47 – Взаимодействие через глобальные переменные

Глобальные переменные позволяют использовать привязку к системным переменным ПЛК. Значения переменных могут передаваться между программами в одной конфигурации через глобальные переменные, как переменная x, показанная на рисунке 47.

6.6.2 Переменные ПЛК

В таблице 8 приведен список параметров модулей ПЛК, доступных для использования прикладной программой.

Таблица 8 – Описание параметров ПЛК, доступных для использования прикладной программой

Параметр	Тип	Описание
Параметры модулей		
Идентификатор типа модуля расширения (идентификатор модуля)	UINT	Сервисная информация
Количество каналов у модуля	UINT	Сервисная информация
Идентификатор производителя	UINT	Сервисная информация
Заводской номер модуля	UDINT	Сервисная информация

Параметр	Тип	Описание
Контрольная сумма прошивки	UDINT	Сервисная информация
Версия печатной платы	UINT	Сервисная информация
Версия программного обеспечения	UDINT	Сервисная информация
Температура внутри модуля	REAL	Диагностическая информация
Напряжение на модуле	REAL	Диагностическая информация
Время работы модуля	ULINT	Диагностическая информация
Поле флагов соединения модуля	UINT	1 – Нет связи; 2 – Есть связь.
Поле флагов линии модуля	UINT	0 – Не используется; 1 – OFFLINE; 2 – ONLINE. Состояние связи RS-485 R_IBC_1 – 0b0000_0000_0000_00xx; Состояние связи RS-485 R_IBC_2 – 0b0000_0000_0000_xx00; Состояние связи RS-485 R_IBC_3 – 0b0000_0000_00xx_0000; Состояние связи RS-485 R_IBC_4 – 0b0000_0000_xx00_0000; Состояние связи CAN_1 R_IBC – 0b0000_00xx_0000_0000; Состояние связи CAN_2 R_IBC – 0b0000_xx00_0000_0000; Состояние связи RS-485 R_BUS_1 – 0b00xx_0000_0000_0000; Состояние связи RS-485 R_BUS_1 – 0bxx00_0000_0000_0000.
Счетчик принятых пакетов	ULINT	Диагностическая информация
Счетчик отправленных пакетов	ULINT	Диагностическая информация
Счетчик ошибок коммуникации	ULINT	Диагностическая информация

Параметр	Тип	Описание
Поле флагов ошибок модуля (PWR-01)	UINT	Бит 0 – Ошибка EEPROM.
Поле флагов ошибок модуля (CPU-01, CPU-02)	UINT	Бит 0 – Ошибка GPIO.
Поле флагов ошибок модуля (CP-12, CC-12, CM-221, CM-222, CM-223)	UINT	Бит 0 – Ошибка EEPROM; Бит 3 – Ошибка конфигурации.
Поле флагов ошибок модуля (CM-340, CM-341, CM-342, CM-440, CM-441)	UINT	Для каналов: Статусы (флаги) порта N: • 0 – Буфер пуст; • 1 – Буфер частично заполнен; • 2 – Буфер полон; • 0b00xx – состояние приёмного буфера; • 0bxx00 – состояние буфера на отправку. Свободное место порта N – количество байт, свободное в буфере на отправку.
Поле флагов ошибок модуля (модули ввода-вывода)	UINT	Бит 0 – Ошибка EEPROM; Бит 1 – Ошибка соединения; Бит 2 – Отсутствие терминальной панели; Бит 3 – Ошибка конфигурации.
Каналы питания (PWR-01)		
Сигнал «Питание ОК»	UDINT	Состояние канала питания: 0 – Ошибка питания; 1 – ОК.
Ток, А	REAL	Значение тока канала питания
Напряжение	REAL	Значение напряжения канала питания
Преобразователь питания (PWR-01)		
Сигнал «Питание ОК»	UDINT	Состояние преобразователя питания: 0 – Напряжение не входит в диапазон; 1 – Напряжение входит в диапазон от 90 до 110 % от выходного.
Выходной ток, А	REAL	Значение тока преобразователя питания

Параметр	Тип	Описание
Выходное напряжение, В	REAL	Значение напряжения преобразователя питания
Выходное напряжение (внутр. ADC), В	REAL	Значение напряжения преобразователя питания, измеренное с АЦП МК
Каналы ввода (CPU-01, CPU-02, GCP-01)		
Значение на входе	BOOL	Значение канала дискретного ввода
Код ошибки	INT	Состояние канала дискретного ввода
Каналы вывода (CPU-01, CPU-02)		
Значение на выходе	BOOL	Значение канала дискретного вывода
Код ошибки	INT	Состояние канала дискретного вывода
Сторожевой таймер (СТ) (CPU-01, CPU-02)		
Управление работой СТ	INT	1 – Запуск СТ; 2 – Останов СТ; 3 – Обновление СТ.
Интервал срабатывания, мс	INT	Таймаут срабатывания СТ, мс
Пользовательская кнопка (CPU-01, CPU-02)		
Значение состояния «Нажата»	BOOL	0 – Не нажата; 1 – Нажата.
Резервирование (CPU-01, CPU-02)		
Ведущий А	INT	1 – Ведущий А; 2 – Ведущий Б.
Обмен контекстом	BOOL	1 – Есть обмен
Версии (CPU-01, CPU-02)		
Версия ФС ОС Линукс	INT	Сервисная информация
Версия системного ПО	INT	Сервисная информация
Каналы (AI-116, GAI-104)		
Значение на входе, мА | В	REAL	Значение тока или напряжения, подаваемого на ввод
Масштабированное значение на входе, инж. ед.	REAL	Значение тока или напряжения, подаваемого на ввод, в единицах АЦП

Параметр	Тип	Описание
Установить значение коэффициента масштабирования	REAL	Калибровочный коэффициент A
Установить значение коэффициента смещения	REAL	Калибровочный коэффициент B
Код ошибки	UINT	Бит 0 – Ошибка состояния АЦП; Бит 1 – Рассчитанное значение вне диапазона; Бит 2 – Ошибка установки значений; Бит 4 – Ошибка модуля.
Каналы (AI-416)		
Значение на входе, мА	REAL	Значение тока, подаваемого на ввод
Масштабированное значение на входе, инж. ед.	REAL	Значение тока или напряжения, подаваемого на ввод, в единицах АЦП
Установить значение коэффициента масштабирования	REAL	Калибровочный коэффициент A
Установить значение коэффициента смещения	REAL	Калибровочный коэффициент B
Код ошибки	UINT	Бит 0 – Ошибка состояния АЦП; Бит 1 – Рассчитанное значение вне диапазона; Бит 2 – Ошибка установки значений; Бит 3 – Ошибка связи по каналу HART; Бит 4 – Ошибка модуля.
Каналы (AI-208, GAI-204)		
Значение на входе, °C	REAL	Значение температуры термопреобразователя сопротивления и термопары
Код ошибки	INT	Бит 0 – Ошибка состояния АЦП; Бит 1 – Рассчитанное значение вне диапазона; Бит 2 – Ошибка установки значений; Бит 4 – Ошибка модуля.

Параметр	Тип	Описание
Каналы компенсации T° XC (AI-208, GAI-204)		
Текущее значение температуры холодного спая, °C	REAL	Температура компенсации термопары
Код ошибки	INT	Бит 0 – Ошибка состояния АЦП; Бит 1 – Рассчитанное значение вне диапазона; Бит 2 – Ошибка установки значений; Бит 4 – Ошибка модуля.
Каналы (AO-108, AO-408, GAO-104)		
Текущее значение на выходе	REAL	Значение тока или напряжения на выходе
Установить значение на выходе	REAL	Значение тока или напряжения на выходе
Текущее приведенное значение на выходе	REAL	Значение тока или напряжения в единицах ЦАП
Установить приведенное значение на выходе	REAL	Значение тока или напряжения в единицах ЦАП
Установить значение коэффициента масштабирования	REAL	Калибровочный коэффициент A
Установить значение коэффициента смещения	REAL	Калибровочный коэффициент B
Код ошибки	INT	• AO-108, GAO-104: Бит 0 – Ошибка состояния ЦАП; Бит 1 – Рассчитанное значение вне диапазона; Бит 2 – Ошибка установки значений; Бит 4 – Ошибка модуля; • AO-408: Бит 1 – Рассчитанное значение вне диапазона; Бит 2 – Ошибка установки значений; Бит 3 – Ошибка связи по каналу HART; Бит 4 – Ошибка модуля.

Параметр	Тип	Описание
Каналы (DI-124, GDI-108)		
Значение на входе	BOOL	Логическое состояние дискретного сигнала на входе
Код ошибки	INT	Бит 4 – Ошибка модуля
Каналы (DO-132, GDO-108)		
Значение на выходе	BOOL	Логическое состояние дискретного сигнала на выходе
Код ошибки	INT	Бит 4 – Ошибка модуля

6.7 Редакторы

Редакторы обеспечивают возможность редактирования программных компонентов на соответствующих языках.

6.7.1 Редактор FBD

Редактор FBD (рисунок 48) открывается двойным щелчком по программному элементу на языке FBD в дереве проекта. Он обеспечивает для среды R-Logic.Designer возможность работы с программами, функциональными блоками и функциями на языке FBD.

При работе в редакторе FBD доступны изменения свойств поля редактирования (таблица 9).

Таблица 9 – Свойства поля редактирования

Название	Описание
Автоматическая нумерация блоков	Активируется флажком для автоматического присвоения номера блоку при добавлении его в окно редактора
Порядок автоматической нумерации	Нумерация «Сверху вниз, слева направо» и «Слева направо, сверху вниз»
Ориентация страницы	Книжное или альбомное отображение страницы
Размер страницы	Выбор из перечня форматов от A4 до A2
Масштаб просмотра	Выбор масштаба в % от базового значения
Левое, правое, верхнее, нижнее поле, мм	Отступ от левого, правого, верхнего, нижнего края страницы соответственно

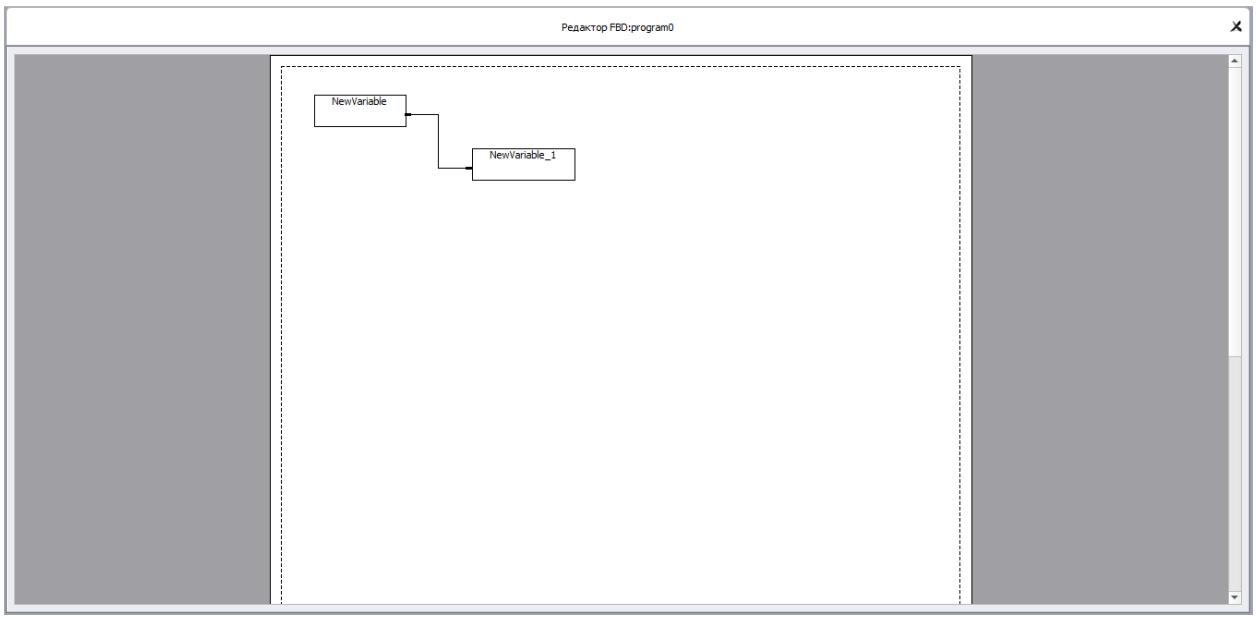

Рисунок 48 – Внешний вид редактора FBD

Разработка программы на языке FBD осуществляется с помощью готовых элементов языка, расположенных в библиотеке.

Для добавления на поле редактирования переменные переносятся с поля переменных, блоки – из библиотеки компонентов. Для передачи результата между ними добавляются связи. Программа исполняется по порядку связей слева направо и сверху вниз.

Переменная в редакторе FBD отображается в виде прямоугольника с названием. У входной переменной есть только связь, расположенная слева (вход); у выходной – только справа (выход); у входной/выходной – вход и выход, расположенные слева и справа соответственно.

После размещения элемента на поле программы пользователю доступны следующие действия:

- выделение элементов, находящихся за границей отображения, доступно посредством пролистывания с помощью колесика мыши. При наведении курсора на горизонтальную полосу прокрутки выполняется горизонтальное пролистывание;
- соединение между элементами выполняется с выходной стороны ко входной (противоположной) стороне функции или функционального блока (блоков). Для соединения элементов пользователь нажимает кнопкой мыши на выход, затем, удерживая кнопку нажатой, перемещает курсор к соответствующему входу, отпускает кнопку мыши, таким образом создавая линию связи;
- при создании связей между блоками производится контроль совместимости типов их входов и выходов. Также исключена возможность соединения входов со входами и выходов с выходами;
- программа контролирует и не допускает объединение нескольких выходов в одной сети;

- при наложении линий связи соединение линий обозначается точкой, пересечение – волной;
- в окне «Свойства» связям могут быть присвоены наименования. При этом контролируется уникальность их идентификаторов, а также не допускается использование служебных слов и наименований переменных и компонентов ПО;
- все блоки редактора имеют порядковый номер выполнения, значение и отображение которого могут быть заданы в настройках программы.

6.7.2 Редактор ST

Редактор ST (рисунок 49) открывается двойным щелчком по соответствующему программному элементу в дереве проекта.

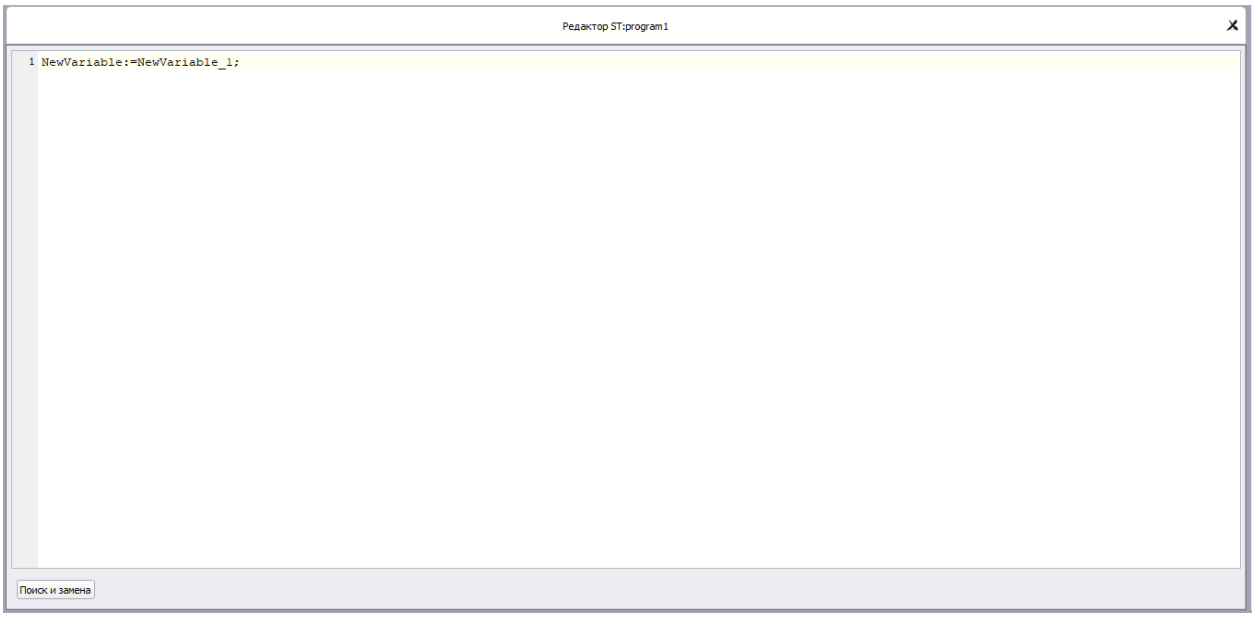

Рисунок 49 – Внешний вид редактора ST

Редактор позволяет размещать на текстовом поле редактирования компоненты библиотеки.

Редактор поддерживает использование «горячих клавиш» (см. п. 4.1.2), соответствующих элементов панели инструментов, а также вызов команд правки по щелчку ПКМ.

В нижней части области редактора под текстовым полем расположена кнопка «Поиск и замена», при нажатии на которую раскрывается панель (рисунок 50), содержащая:

- поле «Поиск», кнопки «Найти», «Предыдущий», «Следующий» и «Очистить»;
- поле «Замена», кнопки «Заменить» и «Заменить все».

Рисунок 50 – Поиск по редактору ST

Поиск с помощью кнопки «Найти» осуществляется со строки №1, первый результат выделяется цветом, процесс поиска при этом останавливается. Для поиска следующего совпадения пользователь может повторно нажать кнопку «Найти».

При нажатии на кнопку «Найти все» программа запускает процесс поиска и останавливает его только после того, как все совпадения будут найдены и выделены.

Кнопки «Предыдущий» и «Следующий» позволяют переключаться между найденными совпадениями.

Для активации поля «Замена» поле «Поиск» должно быть заполнено, а совпадения найдены. Кнопка «Заменить» заменяет найденный и выделенный элемент в текстовом окне редактора, а кнопка «Заменить все» заменяет все выделенные элементы в текстовом окне редактора.

6.7.3 Редактор LD

Редактор LD (рисунок 51) открывается двойным щелчком по соответствующему программному элементу в дереве проекта.

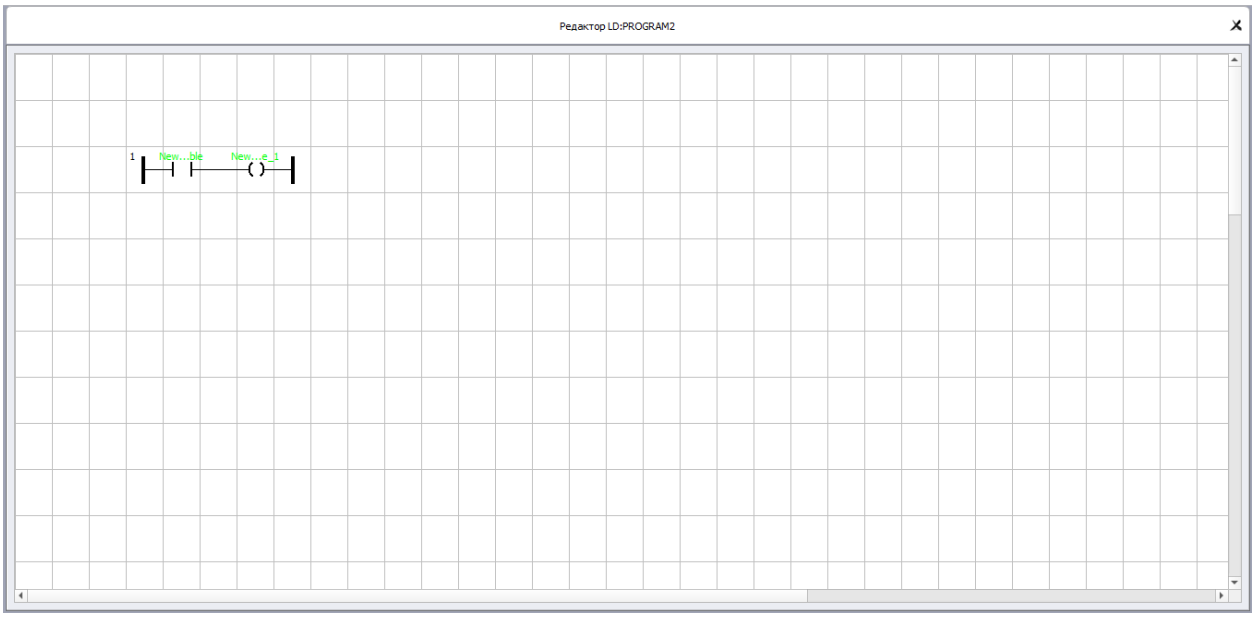

Рисунок 51 – Внешний вид редактора LD

Редактор позволяет размещать на графическом поле редактирования компоненты библиотеки.

Узел библиотеки компонентов «Элементы языка LD» содержит следующие элементы:

- левая и правая шины питания (каждая левая шина имеет порядковый номер);
- горизонтальное, вертикальное, крестовое соединения, а также пересечение без соединения (нумерация сверху вниз);
- верхнее, правое, левое и нижнее Т-образные соединения;
- правый верхний, правый нижний, левый верхний и левый нижний уголки;
- нормально разомкнутый и замкнутый контакты;
- контакты, чувствительные к положительному и отрицательному переходам;

- катушка;
- обратная обмотка;
- устанавливающая катушка (с фиксацией включения);
- сбрасывающая катушка (с фиксацией выключения);
- катушки, чувствительные к положительному и отрицательному переходам.

Поле редактора разбито на прямоугольные ячейки фиксированного размера для размещения элементов по одному на ячейку.

При размещении элемента или перемещении его по полю программы доступны следующие действия:

При переносе элемента над ячейками, в которых он может быть размещен, происходит автоматическое определение ячейки элемента. Если ячейка, над которой в данный момент находится элемент не занята другими элементами (за исключением горизонтальных связей), вокруг нее отображается тонкая **зеленая** рамка. Если ячейка занята, то вокруг нее отображается **красная** рамка;

Ячейки, над которыми не находится элемент, отображаются с тонкой **черной** рамкой. При переносе функционального блока над ячейками, в которых он может быть размещен, происходит автоматическое определение необходимого количества ячеек (по вертикали и горизонтали), в которых он будет размещен. При расположении функционального блока над такой ячейкой ее рамка переходит из **черной** в **зеленую**;

При переносе элемента языка LD или элемента из библиотеки компонентов на свободные ячейки поля редактора LD они размещаются в определенных для них ячейках;

При переносе элемента языка LD на ячейку, занятую горизонтальной связью, элемент заменяет собой соответствующий сегмент горизонтальной связи;

При попытке поместить элемент или блок в занятые ячейки (исключая размещение элемента языка FBD на сегменте горизонтальной связи), операция размещения отменяется (рисунок 52).

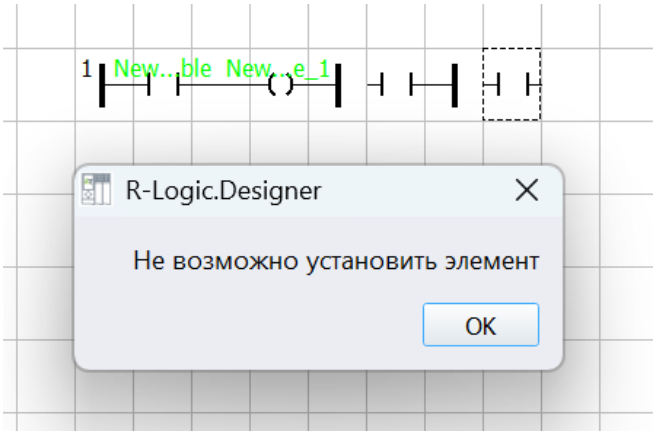

Рисунок 52 – Окно с сообщением «Невозможно установить элемент»

При размещении элементов на поле редактирования связи между компонентами, имеющими соприкосновение по границам, устанавливаются автоматически, т.е. если

компонент, размещенный слева от границы соприкосновения, имеет правую связь, совпадающую по вертикали с левой связью компонента, размещенного справа от границы, такие связи автоматически объединяются.

Привязка переменных к элементам программы LD осуществляется путем задания их названия в соответствующем поле ввода элемента.

В случае задания имени несуществующей переменной автоматически создается переменная с введенным именем. При несовпадении имени переменной с существующими наименованиями при вводе оно выделяется **зеленым** цветом.

При вводе производится проверка типа привязываемой переменной. Если привязываемая переменная имеет несовпадающий тип, ее наименование выделяется **красным** цветом.

6.7.4 Редактор SFC

Редактор SFC открывается двойным щелчком по соответствующему программному элементу в дереве проекта.

Основной функцией редактора является редактирование последовательности функциональных схем. Редактор позволяет работать с дисковой копией последовательности функциональных схем (offline) и контролировать выполнение SFC, запущенных в окне «Контроллер» (online).

Редактор SFC организует блоки логики в блок-схему. В редакторе создаются программы последовательного управления технологическими процессами, описываемыми в форме, близкой к диаграмме состояний.

Редактор обеспечивает возможность работы с программными модулями типа структура.

Для сохранения программы используется XML-файл проекта в соответствии с требованиями, приведенными в документе PLCopen Technical Committee 6 XML Formats for IEC 61131-3.

Редактор взаимодействует с ядром среды R-Logic.Designer посредством механизма передачи сообщений «сигнал-слот».

При работе с программой элементы языка SFC добавляются на графическое поле редактирования.

В редакторе SFC доступны следующие функции:

- «Сохранить программу»;
- команды редактирования:
  - «Копировать»;
  - «Вырезать»;
  - «Вставить»;
  - «Удалить»;
  - «Отменить»;
  - «Возврат отмены»;

    - «Выделить все»;
- «Изменить свойства переменной»;
- «Изменить наименование переменной»;
- «Печать».

При выборе в окне редактора элемента программы редактор SFC предоставляет информацию о его свойствах.

При разбиении на элементы SFC программный компонент ПЛК также разбивается. При отсутствии разбиения он рассматривается как одиночное действие и выполняется под управлением вызывающего объекта.

Редактор SFC содержит следующие элементы:

- шаг – описывает одну операцию. Шаги отделяются друг от друга переходами. Графически представлены в виде прямоугольников с указанием имени;
- переход – обеспечивает переключение активности шагов при выполнении условия перехода или после полного выполнения всех операций на данном шаге. На графическом поле редактирования переход изображается как горизонтальная черта на ветви диаграммы процесса с указанием имени и условия перехода;
- условие перехода – логическое условие, описывающее переход между шагами;
- метка/переход на метку (например, к шагу N) – обозначает начало цепи. Имя метки подчиняется правилам наименования переменных (см. п. 6.6). Цепь может содержать только одну метку и один переход. Графически метка отображается как треугольник с соединительной линией;
- действие – операция, которая должна выполняться на заданном шаге. Может быть описано на языке ST. Может выполняться непрерывно (пока активен шаг), либо единожды – это определяется квалификаторами. Действие графически представлено прямоугольником, разделенным на две части, со входом слева. В левой части прямоугольника отображается наименование, в правой – выбранный классификатор;
- ветвление/сходимость – конструкция, позволяющая описать логику программы на основе чередующихся шагов и условных переходов, а также описать параллельно выполняемые и альтернативные последовательности. Элемент графически представлен в виде горизонтальной линии (альтернативное) или двойной горизонтальной линии (параллельное ветвление/схождение) с отходящими вверх и вниз перпендикулярными линиями.

Действия и условия перехода на языке ST автоматически добавляются в окно проекта в узел программы SFC, которой принадлежат, и доступны только этой программе.

Каждый шаг имеет не более трех соединений: с переходом (сверху и снизу) и с блоком действий (справа). При активации в свойствах флага «Начальный шаг» отключается отображение верхнего соединения, при этом прямоугольник, соответствующий начальному шагу на графическом поле редактирования, выделяется двойной рамкой. При активации

флага «Выход» отключается отображение нижнего соединения шага, а при активации флага «Действие» – правого соединения.

При создании шага в окне проекта во вкладке «Переменные» для узла с именем соответствующего программного компонента на языке SFC автоматически создается структура с именем шага.

Переход имеет следующие свойства:

- поле «Наименование» – соответствует содержанию полей «Привязка» или «Логическое выражение» в зависимости от типа;
- выпадающий список «Тип» содержит элементы «Привязка» и «Логическое выражение»;
- поле «Привязка» с кнопками «…» и «+» появляется при выборе типа «Привязка». При нажатии на кнопку «…» открывается окно выбора элемента типа «условие» из имеющихся в узле программного компонента SFC. После выбора наименование выбранного элемента отображается в поле. При нажатии на кнопку «+» открывается диалоговое окно создания элемента «условие», содержащее поле «Наименование», кнопки «Создать» и «Отменить». После нажатия на кнопку «Создать» созданный элемент типа «условие» отображается в узле программного компонента SFC в окне проекта и в поле редактирования;
- поле «Логическое выражение» появляется при выборе типа «Логическое выражение». При нажатии на кнопку «…» открывается окно выбора переменной, которая отображается затем в поле редактирования.

Условие перехода имеет следующие особенности:

- добавление условия осуществляется через окно свойств элемента «Переход»;
- пересекающимися называются условия, которые могут выполняться одновременно. Например, условия v1 > 10 и v1 < 100 являются пересекающимися. Рекомендуется избегать пересекающихся условий при определении переходов SFC-схемы;
- альтернативные ветви начинаются и заканчиваются собственным условием перехода. Проверка альтернативных условий выполняется слева направо. Если верное условие найдено, то прочие альтернативы не рассматриваются;
- условие перехода, завершающее параллельность, проверяется только в случае, если в каждой параллельной ветви активны последние шаги.

Действие имеет следующие свойства:

- выпадающий список «Классификатор», в котором пользователь может выбрать одно из значений:
  - - (Отсутствует) – Не сохраняется (нулевой классификатор);
  - N – Не сохраняется;
  - R (Сброс переопределения) – действие с классификатором R сбрасывает сохраняемые действия;

- S (Установка (Сохранено)) – действие с классификатором SL начинает выполняться при активации связанного с ним шага и выполняется в каждом цикле ПЛК (даже если шаг уже не является активным) или до его вызова с классификатором R;
- L (Ограничено по времени) – действие с классификатором L начинает выполняться при активации связанного с ним шага и выполняется в каждом цикле ПЛК в течение заданного времени. Если шаг деактивируется до истечения заданного времени, то выполнение действия прекращается;
- D (Отложено) – действие с классификатором D начинает выполняться спустя заданное время после активации шага и выполняется в каждом цикле до тех пор, пока шаг не будет деактивирован. Если шаг деактивируется до истечения заданного времени, то действие ни разу не будет вызвано;
- P (Импульс) – действие с классификатором P выполняется в течение одного (если последнее сканирование не используется) или двух (если последнее сканирование используется) циклов ПЛК при активации связанного с ним шага;
- P0 (Импульс задний фронт) – действие с классификатором P1 выполняется в течение одного цикла ПЛК при активации связанного с ним шага;
- P1 (Импульс передний фронт) – действие с классификатором P0 выполняется в течение одного цикла ПЛК при деактивации связанного с ним шага;
- SD (Сохранено и отложено по времени) – действие с классификатором SD начинает выполняться спустя заданное время после активации шага и выполняется в каждом цикле ПЛК до тех пор, пока не будет вызвано с классификатором R. Если действие деактивируется до истечения заданного времени, то действие ни разу не будет выполнено;
- DS (Отложено и сохранено) – действие с классификатором DS начинает выполняться спустя заданное время после активации шага (если шаг к этому моменту остается активен) и выполняется в каждом цикле ПЛК до тех пор, пока не будет вызвано с классификатором R. Если действие вызывается с классификатором R до истечения заданного времени, то его код ни разу не будет выполнен;
- SL (Сохранено и ограничено по времени) – действие с классификатором SL начинает выполняться при активации связанного с ним шага и выполняется в каждом цикле ПЛК в течение заданного времени (даже если шаг уже не является активным) или до его вызова с классификатором R;

- поле «Длительность» отвечает за задание значения времени, применение которого обусловлено классификатором (доступно при выборе классификаторов L, D, SD, DS, SL);
- поле «Привязка» с кнопками «…» и «+». При нажатии на кнопку «…» открывается окно выбора элемента типа «действие» из имеющихся в узле программного компонента SFC. После выбора наименование выбранного элемента отображается в поле. При нажатии на кнопку «+» открывается диалоговое окно создания элемента «действие», содержащее поле «Наименование», кнопки «Создать» и «Отменить». После нажатия на кнопку «Создать» созданный элемент типа

«действие» отображается в узле программного компонента SFC в окне проекта и в поле редактирования.

Добавление действий в шаг производится через перенос действия в окно редактора и подключение к выбранному шагу. Добавление последующих действий осуществляется путем нажатия в окне редактора на треугольные кнопки, доступные при нажатии на объект типа «действие».

Для элемента «метка» выбор шага для перехода определяется в окне свойств в поле «Переход к шагу» при нажатии на кнопку «…». В открывшемся окне отображаются все шаги программы, которые пользователь может выбрать, а также находится поле поиска.

При параллельном ветвлении/схождении может быть только одно условие перехода, при альтернативном – разные условия. По умолчанию при добавлении в область редактора отображается альтернативное ветвление с двумя выходами.

Ветвление/схождение имеет следующие свойства:

- выпадающий список «Тип», содержащий выбор между альтернативными и параллельными ветвлениями и схождениями;
- поле «Количество линий соединения» – по умолчанию содержит значение 2.

При создании программ на языке SFC не допускается соединять два шага непосредственно (они всегда должны быть разделены переходом), а также соединять два перехода (они всегда должны быть разделены шагом).

Эволюции активных состояний шагов происходят вдоль направлений связей, когда они вызваны очисткой одного или большего числа переходов. Переход разрешается, когда все предшествующие шаги, соединенные с соответствующим символом перехода направленными связями, активны. Пересечение перехода происходит, когда он разрешен, а соответствующее условие перехода истинно.

Очистка перехода вызывает деактивацию (сброс) всех непосредственно предшествующих шагов, соединенных с соответствующим символом перехода, с последующей активацией всех непосредственно последующих шагов.

Когда очистка перехода приводит к активации нескольких шагов одновременно, то последовательности, к которым принадлежат такие шаги, называются параллельными. После их параллельной активации эволюция таких последовательностей становится независимой. Дивергенция и конвергенция параллельных последовательностей обозначается двойной горизонтальной линией.

Несколько переходов, которые могут быть очищены параллельно, очищаются параллельно в пределах временных ограничений реализации конкретного ПЛК и значений по приоритету.

Испытание условий перехода преемника активного шага не выполняется, пока влияние активации шага не распространится по программному компоненту.

6.8 Конфигурация запуска

Отправной точкой исполнения ППО ПЛК является программный компонент типа «программа», сконфигурированный на исполнение в задаче. Существуют задачи с периодическим исполнением, кратно указанному интервалу, а также задачи с запуском по событию (триггеру). Для корректной работы ППО требуется как минимум одна задача периодического исполнения с назначенным экземпляром программы.

Вкладка «Конфигурация» по умолчанию содержит элемент «resource1», при двойном щелчке по которому открывается окно «Конфигурация запуска программы: resource1» (рисунок 53).

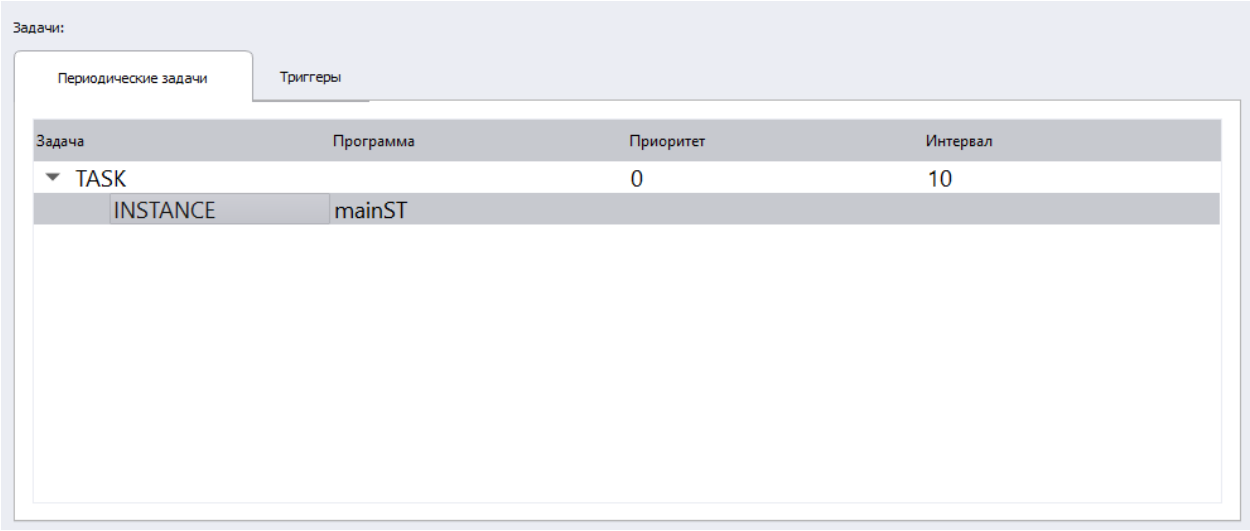

Рисунок 53 – Окно конфигурации запуска программ

Управление задачами и экземплярами программ происходит через вызов контекстного меню:

- при нажатии ПКМ по пустой строке таблицы появляется команда «Добавить задачу».

При нажатии ПКМ по существующей задаче появляется список команд:

- «Удалить задачу»;
- «Добавить экземпляр технологической программы» – при выборе команды имя программного блока отображается в столбце «Программа».

Наименования задач и экземпляров технологических программ могут содержать только символы латинского алфавита и цифры, их имена также должны быть уникальными.

Выбор программного блока для соответствующей задачи во вкладке «Периодические задачи» осуществляется посредством выпадающего списка (появляется при двойном щелчке по наименованию POU).

6.9 Создание и настройка служб контроллера

ПЛК R-Logic поддерживает обмен информацией между контроллером и сторонним оборудованием по стандартным протоколам Modbus RTU, Modbus TCP, МЭК 60870-5-101,

МЭК 60870-5-104, BACnet, HART, а также возможность реализации собственного протокола обмена поверх интерфейсов RS-485 и RS-232. Данный функционал реализуется службами обмена. При этом допускается создание нескольких экземпляров служб.

Выбор режима работы (ведущий/ведомый) службы происходит при создании экземпляра службы обмена. Процесс конфигурирования настроек исполнения и выбора устройства для всех служб идентичный.

Для протоколов обмена, работающих через интерфейс RS-485 или RS-232, выбирается модуль, его адрес на шине ПЛК (шасси, слот), порт модуля, а также параметры обмена. Окно конфигуратора службы сервера Modbus представлено на рисунке 54.

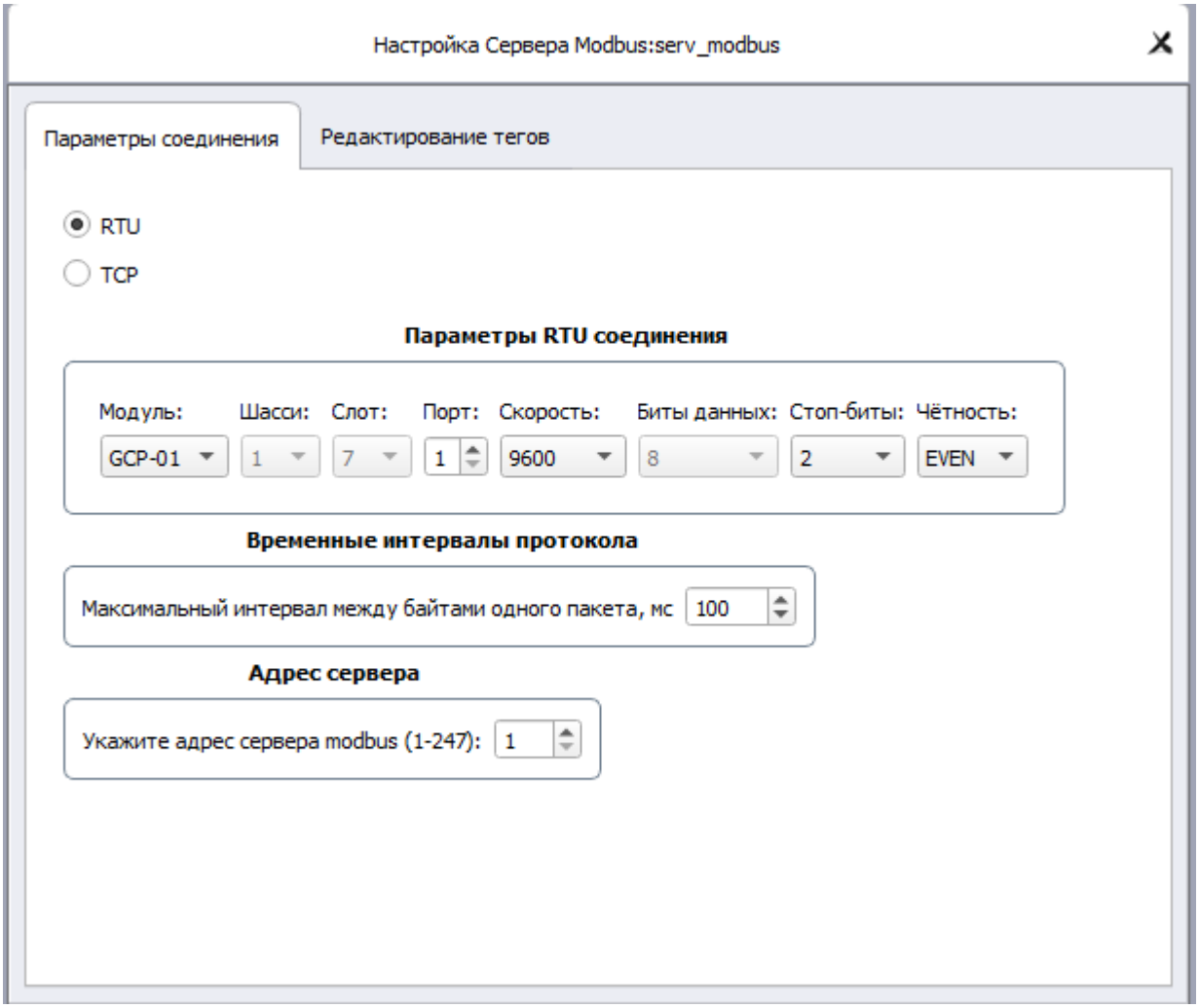

Рисунок 54 – Окно настройки службы Modbus-сервера (RTU-соединение)

6.9.1 Настройка и работа со службами Modbus (TCP, RTU)

Служба сервера протокола Modbus позволяет предоставлять данные по протоколам связи Modbus TCP и Modbus RTU.

Реализация протокола обеспечивает работу стандартных функций 01, 02, 03, 04, 05, 06, 10, 16, а также для функций от 65 до 72 и от 100 до 110 служба предоставляет

возможность получения буфера данных запроса и отправки буфера данных сформированного ответа.

Выбор интерфейса передачи данных (Ethernet/RS-485/RS-232) вместе с типом протокола (TCP/RTU) происходит в процессе конфигурирования и может быть изменен в дальнейшем.

Формирование запросов/ответов на стандартные функции происходит автоматически по сформированной таблице тегов. Для формирования этой таблицы пригодны переменные проекта простых типов, за исключением BYTE. Также допустимо использовать массивы элементов.

6.9.1.1 Конфигурирование службы обмена по протоколу Modbus в режиме «ведомого»

ВАЖНО! При создании экземпляра службы необходимо указать тип «Серверы Modbus».

На вкладке «Параметры соединения» пользователь указывает адрес устройства на шине и параметры соединения.

На вкладке «Редактирование тегов» (рисунок 55) происходит формирование адресной таблицы. Для этого в правой части экрана присутствуют элементы для добавления переменных проекта в таблицу, перемещения тегов по таблице, а также механизм импорта/экспорта.

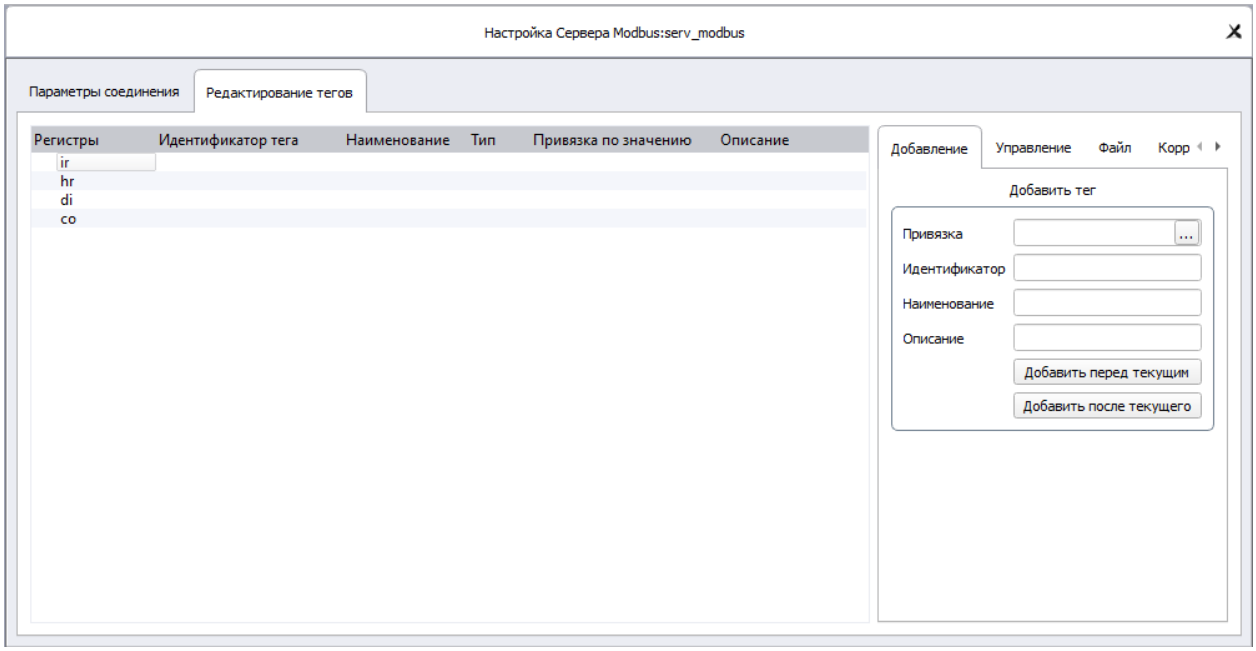

Рисунок 55 – Окно настройки службы Modbus-сервера (вкладка «Редактирование тегов»)

Добавление переменных в перечень обмена происходит следующим образом:

- нажать кнопку «…» и выбрать переменную проекта;
- выбрать строку в таблице и нажать одну из кнопок «Добавить»;

- тег будет добавлен возле выбранной строчки. Адрес назначается автоматически с учетом размера предыдущей переменной (включая массивы);
- при необходимости ручной установки адреса необходимо сделать двойной клик по полю адреса и ввести значение. Адреса последующих регистров будут пересчитаны автоматически.

При необходимости смены порядка уже добавленных тегов на вкладке «Управление» расположены кнопки «Вверх», «Вниз» для перемещения выбранного тега по таблице.

Удаление выбранной строки происходит через контекстное меню или кнопкой «Удалить» на вкладке «Управление».

Присутствует функция импорта/экспорта таблицы тегов в формат XML. Импорт может быть совмещен с созданием экземпляров переменных и их привязкой.

На вкладке «Коррекция» расположены инструменты для группового управления привязками тегов.

После проведения настройки службы, формирования перечня тегов и привязки к ним переменных технологической программы значения тегов могут быть предоставлены вышестоящим устройствам.

6.9.1.2 Работа с запросами, для которых не реализована автоматическая обработка

Для функций протокола Modbus от 65 до 72 и от 100 до 110 служба предоставляет технологическому приложению возможность получения буфера данных запроса и отправки буфера данных сформированного ответа.

Для этого при создании элемента службы сервера Modbus в библиотеке компонентов проекта создается функциональный блок с соответствующим названием (рисунок 56).

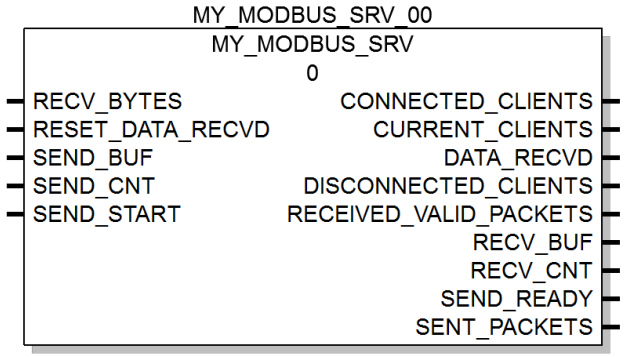

Рисунок 56 – Внешний вид функционального блока экземпляра службы сервера протокола Modbus

Ниже даны входные и выходные переменные функциональных блоков, их типы данных и описание (таблица 10, таблица 11).

Таблица 10 – Перечень входных переменных функционального блока службы сервера Modbus

Мнемоника	Тип	Значение по умолчанию	Описание
RECV_BYTES	INT	254	Количество байт, которое может быть размещено в буфере приема. Меньше или равно размерности массива принимаемых байтов.
RESET_DATA_ RECVD	BOOL	0	При значении «1» производится сброс флага приема пакета DATA_RECVD
SEND_BUF	BYTE [254]		Массив отправляемых байтов
SEND_CNT	INT	254	Количество байт с начала массива SEND_BUF, которые передаются в порт
SEND_START	BOOL	0	Сигнал отправки пакета. При значении «1» происходит передача информационного пакета.

Таблица 11 – Перечень выходных переменных функционального блока службы сервера Modbus

Мнемоника	Тип	Описание
CONNECTED_ CLIENTS	ULINT	Значение счетчика количества подключений клиентов
CURRENT_ CLIENTS	ULINT	Количество соединений, установленных клиентами в настоящее время
DATA_RECVD	BOOL	Значение «1» после получения целостного пакета данных. Сбрасывается после установки флага RESET_DATA_RECVD.
DISCONNECTED_ CLIENTS	ULINT	Значение счетчика количества отключений клиентов
RECEIVED_ VALID_PACKETS	ULINT	Значение счетчика полученных пакетов (от всех клиентов)
RECV_BUF	BYTE [254]	Массив, содержащий принятые данные
RECV_CNT	INT	Количество байт данных в принятом пакете

Мнемоника	Тип	Описание
SEND_READY	BOOL	Готовность службы к передаче данных
SENT_PACKETS	ULINT	Значение счетчика отправленных пакетов (всем клиентам)

Последовательность действий технологического приложения при обработке запросов с пользовательскими функциями:

шаг 1: Приложение осуществляет мониторинг флага DATA_RECVD функционального блока службы сервера Modbus;

шаг 2: Если флаг равен «1», технологическое приложение использует данные из RECV_BUF и RECV_CNT для подготовки ответа на запрос;

шаг 3: После использования данных из RECV_BUF технологическое приложение устанавливает флаг RESET_DATA_RECVD, что приводит к сбросу флага DATA_RECVD;

шаг 4: Подготовленный ответ размещается в SEND_BUF;

шаг 5: Устанавливается входная переменная SEND_CNT в соответствии с количеством байт в подготовленном буфере;

шаг 6: Если установлен флаг SEND_READY, устанавливается флаг SEND_START;

шаг 7: Приложение переходит к шагу 1.

6.9.1.3 Конфигурирование службы обмена по протоколу Modbus в режиме «ведущего»

ВАЖНО! При создания экземпляра службы необходимо указать тип «Клиент Modbus».

На вкладке «Параметры соединения» пользователь указывает адрес устройства на шине и параметры соединения.

Конфигурирование тегов обмена реализована идентично режиму «ведомый» и описано в п. 6.9.1.1.

Реализация службы клиента Modbus позволяет управлять чтением/записью из прикладной программы (таблица 12).

Таблица 12 – Атрибуты, управляющие процессом чтения и записи и отражающие его статус

Мнемоника	Описание параметра
PRIORITY	Приоритет записи для тегов, описанных в секциях HR и CO. Формирование запросов на запись тегов при их одновременной модификации технологическим приложением происходит в порядке возрастания значения приоритета.
GROUP	Номер группы, в которую включен тег
STATUS_WRITE	Наименование привязанной переменной статуса операции записи тега. Тип переменной – INT
STATUS_READ	Наименование привязанной переменной статуса операции чтения тега. Тип переменной – INT
EXTRA_READ_CMD	Наименование привязанной переменной флага необходимости экстренного чтения
EXTRA_WRITE_CMD	Наименование привязанной переменной флага необходимости экстренной записи

После проведения настройки службы, формирования перечня тегов и привязки к ним переменных технологическая программа может получать значения тегов и управлять чтения и записью отдельных тегов. Чтобы управлять работой службы в целом, при создании элемента службы в библиотеке компонентов создается ФБ с соответствующим названием (рисунок 57).

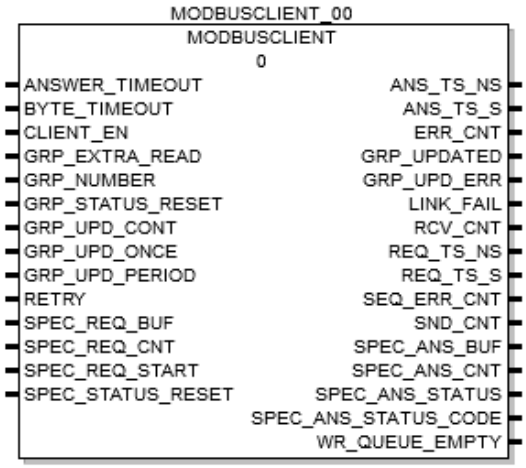

Рисунок 57 – Внешний вид блока управления экземпляром службы клиента Modbus

Ниже представлены список входных и выходных переменных блоков, их типы данных и описание (таблица 13, таблица 14).

Таблица 13 – Перечень входных переменных функционального блока управления службой клиента Modbus

Мнемоника	Тип	Значение по умолчанию	Описание
CLIENT_EN	BOOL	TRUE	«TRUE» разрешает выполнение операций функционального блока. Иначе операции блока не выполняются.
Управление опросом			
GRP_NUMBER	INT	0	Номер опрашиваемой группы
GRP_UPD_CONT	BOOL	TRUE	Выполнение циклического опроса группы
GRP_UPD_ PERIOD	INT	1000	Периодичность циклического опроса группы, мс
GRP_UPD_ONCE	BOOL	FALSE	Выполнение однократного опроса группы
GRP_EXTRA_ READ	BOOL	FALSE	Выполнение экстренного опроса группы
GRP_STATUS_ RESET	BOOL	FALSE	Сброс параметров статуса выполнения опроса
Произвольные запросы			
SPEC_REQ_BUF	BYTE [253]	[0..0]	Массив данных произвольного запроса
SPEC_REQ_CNT	BYTE	0	Длина произвольного запроса
SPEC_REQ_ START	BOOL	FALSE	Выполнение произвольного запроса
Настройки обмена по протоколу Modbus			
BYTE_TIMEOUT	INT	Из диалога настройки	Максимальный интервал времени между байтами одного пакета, мс
ANSWER_ TIMEOUT	INT	Из диалога настройки	Максимальное время ожидания ответного пакета, мс
RETRY	INT	Из диалога настройки	Количество повторных попыток выполнения запросов, шт

Таблица 14 – Перечень выходных переменных функционального блока управления службой клиента Modbus

Мнемоника	Тип	Описание
Состояние связи		
LINK_FAIL	BOOL	Флаг «Устройство не на связи» (1 – устройство не на связи, 0 – устройство на связи)

Мнемоника	Тип	Описание
RCV_CNT	UINT	Количество пакетов, полученных от подчиненного устройства
SND_CNT	UINT	Количество пакетов, отправленных подчиненному устройству
REQ_TS_S	UINT	Метка времени отправки последнего запроса к подчиненному устройству (в секундах, начиная с полуночи по локальному времени 1 января 1970 года)
REQ_TS_NS	UINT	Наносекундная часть метки времени отправки последнего запроса к подчиненному устройству
ANS_TS_S	UINT	Метка времени получения последнего ответа, полученного от подчиненного устройства (в секундах, начиная с полуночи по локальному времени 1 января 1970 года)
ANS_TS_NS	UINT	Наносекундная часть метки времени получения последнего ответа от подчиненного устройства
ERR_CNT	UINT	Количество коммуникационных ошибок
SEQ_ERR_CNT	UINT	Количество последовательных неуспешных запросов
Статус обмена		
WR_QUEUE_ EMPTY	BOOL	Флаг «Очередь записи пуста»
Управление опросом		
GRP_UPDATED	BOOL	Обновление группы завершено
GRP_UPD_ERR	BOOL	Ошибка обновления группы
Произвольные запросы		
SPEC_ANS_BUF	BYTE []	Массив ответа подчиненного устройства на произвольный запрос
SPEC_ANS_CNT	UINT	Длина полученного ответа на произвольный запрос
SPEC_ANS_ STATUS	BOOL	Флаг успешного выполнения произвольного запроса
SPEC_ANS_ STATUS_CODE	UINT	Код статуса выполнения произвольного запроса (таблица 15)

Таблица 15 – Коды статуса выполнения произвольного запроса

Код	Описание значения
00	Код может устанавливаться технологической программой с целью обнаружения обновления значения со стороны службы клиента Modbus
01	Модификация регистров тега успешно произведена

Код	Описание значения
-20	Ошибка превышения времени ответа
-21	Ошибка контрольной суммы ответа

6.9.1.4 Получение технологической программой информации о статусе выполнения операций чтения и записи отдельных информационных тегов БД подчиненного устройства по протоколу Modbus (типы взаимодействия ТВ1, ТВ2, ТВ3, ТВ4, ТВ5)

Каждый тег информационной базы Modbus подчиненного устройства из секций HR и CO имеет возможность привязки переменной статуса операции записи (по ТВ4 и ТВ5). Тип переменной статуса – INT. Служба клиента Modbus после выполнения запроса на запись регистров тега записывает в переменную статуса код статуса выполнения запроса (таблица 16). Часть кодов статуса представляет собой отрицательное значение полученного кода исключения протокола Modbus.

Каждый тег информационной базы Modbus подчиненного устройства из секций IR, DI, HR и CO имеет возможность привязки переменной статуса операции чтения (по ТВ1, ТВ2, ТВ3). Тип переменной – INT. Служба клиента Modbus после выполнения запроса на чтение регистров тега записывает в переменную статуса код статуса выполнения запроса согласно таблице 16. Часть кодов статуса представляет собой отрицательное значение полученного кода исключения протокола Modbus.

Для того, чтобы получить информацию об окончании процесса записи регистров Modbus подчиненного устройства (по ТВ4 и ТВ5), после изменения привязанной переменной технологическое приложение самостоятельно отслеживает изменение привязанной переменной значения тега и при этом производит обнуление привязанной переменной статуса. Последующий переход переменной в ненулевое значение означает, что служба клиента Modbus закончила выполнение запроса и записала в переменную код статуса его выполнения.

Для того, чтобы получить информацию об окончании процесса чтения регистров Modbus подчиненного устройства (по ТВ1, ТВ2, ТВ3) и об обновлении привязанной переменной значения тега, технологическое приложение производит обнуление привязанной переменной статуса чтения. Последующий переход переменной в ненулевое значение означает, что служба клиента Modbus закончила выполнение запроса чтения тега и записала в переменную код статуса его выполнения.

Таблица 16 – Коды статуса выполнения запроса обновления тега Modbus

Код	Описание значения
00	Код может устанавливаться технологической программой с целью обнаружения обновления значения со стороны службы клиента Modbus
01	Чтение или модификация регистров тега успешно произведены

Код	Описание значения
-01	Неверная функция
-02	Неверный адрес данных
-03	Неверное значение данных
-04	Общий сбой устройства сервера
-05	Подтверждение приема данных
-06	Устройство сервера занято
-08	Ошибка контроля четности памяти
-10	Путь к шлюзу недоступен
-11	Ошибка связи шлюза с конечным устройством
-20	Ошибка превышения времени ответа
-21	Ошибка контрольной суммы ответа
-22	Иная ошибка
-23	Ошибка идентификатора пакета (только для Modbus TCP)
-24	Ошибка длины пакета

6.9.1.5 Получение технологической программой информации об опустошении очереди запросов записи данных в регистры подчиненного устройства по протоколу Modbus (по ТВ4, ТВ5)

При организации взаимодействия с подчиненными устройствами, для того чтобы организовать последовательную передачу команд или запись параметров с максимальной скоростью, программа может передавать очередные команды сразу после передачи предыдущих. Информация о том, что направленная устройству команда уже передана, может быть получена путем анализа флага «Очередь записи пуста» (WR_QUEUE_EMPTY) из состава выходных переменных функционального блока управления службой. Значение флага устанавливается службой клиента Modbus равным «1» в случае отсутствия текущих невыполненных запросов записи регистров в устройство. Значение флага устанавливается службой клиента Modbus равным «0» в случае наличия сформированных (с использованием ТВ4 и ТВ5), но невыполненных запросов записи регистров в устройство.

6.9.1.6 Запись информационных тегов в регистры подчиненного устройства по протоколу Modbus в определенном порядке (по ТВ4 и ТВ5)

При изменении значений переменных технологической программы, привязанных к информационным тегам базы Modbus, служба клиента Modbus производит обновление соответствующих регистров подключенного устройства. По умолчанию обновление регистров происходит в порядке возрастания их номера. Для изменения порядка

обновления регистров каждому информационному тегу регистров HR и CO может быть задан приоритет записи. При создании тега приоритет равен «1». Среда разработки R-Logic.Designer позволяет задать значение приоритета. При выполнении запросов на изменение регистров в первую очередь выполняются запросы с более высоким значением приоритета.

Значение приоритета используется как при выполнении операций плановой записи (ТВ4), так и при выполнении операций экстренной записи (ТВ5).

6.9.1.7 Управление процессом опроса подчиненных устройств (по ТВ1 и ТВ2) от технологической программы

Технологическая программа может управлять процессом опроса подключенных устройств и производить однократное или циклическое получение отдельных информационных блоков из их информационных баз.

Для этого каждый информационный тег назначается в несколько групп (минимум один тег в группе). Все теги в описании информационной БД устройства включаются в группу с номером 0. Служба клиента Modbus предоставляет возможность управления опросом групп.

Входные параметры функционального блока, задействованные при работе с группами тегов:

- номер опрашиваемой группы GRP_NUMBER;
- выполнение циклического опроса группы GRP_UPD_CONT;
- периодичность циклического опроса группы GRP_UPD_PERIOD, (мс);
- выполнение однократного опроса группы GRP_UPD_ONCE (происходит при изменении технологической программой значения переменной с «0» на «1»);
- выполнение однократного экстренного опроса группы GRP_EXTRA_READ;
- сброс параметров статуса выполнения опроса GRP_STATUS_RESET;

Выходные параметры функционального блока, задействованные при работе с группами тегов:

- обновление группы завершено GRP_UPDATED;
- ошибка обновления группы GRP_UPD_ERR.

Для обновления определенной группы тегов ее номер присваивается параметру GRP_NUMBER, а с помощью параметров GRP_UPD_CONT, GRP_UPD_PERIOD, GRP_UPD_ONCE выбирается периодический или однократный опрос группы. Если в информационной базе отсутствуют теги группы с указанным номером, то выполняется обновление всей базы целиком, как если бы обновлялась группа с номером 0.

Параметр «Обновление группы завершено» принимает значение «1» после выполнения попытки обновления всех информационных тегов группы. Параметр «Ошибка обновления группы» принимает значение «1» в случае наличия неуспешно выполненных запросов обновления тегов, иначе параметр принимает значение «0». В случае значения «1»

параметра «Сброс параметров статуса выполнения опроса» оба параметра статуса принимают значение «0».

6.9.1.8 Экстренный опрос отдельных тегов информационной БД подчиненных устройств (ТВ3)

Экстренный опрос производится для быстрого обновления ограниченного количества тегов информационной БД подчиненного устройства, т.к. обладает более высоким приоритетом по сравнению с плановым опросом.

Для каждого тега из секций IR, DI, HR и CO информационной базы Modbus подчиненного устройства может быть указана переменная флага требования экстренного опроса чтения (тип переменной – BOOL).

Служба клиента Modbus в процессе работы осуществляет мониторинг флагов требования экстренного чтения и производит выполнение соответствующих запросов при значении флага «1».

После выполнения операции чтения значение соответствующего флага требования экстренного чтения устанавливается службой клиента Modbus равным «0».

Мониторинг флагов принудительного опроса производится при одновременном выполнении следующих условий:

- очередь запросов записи данных в регистры подчиненного устройства пуста;
- очередь запросов на экстренное чтение регистров пуста;
- одновременно не производится выполнение транзакции протокола Modbus в рамках планового опроса подчиненного устройства.

При наличии флага экстренного опроса выполнение планового опроса для всех устройств, находящихся на используемой линии связи, прерывается после окончания текущей транзакции протокола Modbus. При этом, если информационная база устройства была считана не полностью, считанные данные не используются.

6.9.1.9 Экстренная запись отдельных тегов информационной БД подчиненных устройств (ТВ5)

Экстренная запись обеспечивает быструю запись ограниченного количества тегов информационной БД подчиненного устройства.

При выполнении экстренной записи выполнение планового опроса для всех устройств, находящихся на используемом канале связи, прерывается. Выполнение операций записи при этом не отменяется. В случае, если на один канал связи подключено несколько устройств, работа с устройствами происходит в следующем порядке:

шаг 1: Полностью выполняются все сформированные запросы на запись для устройства, на котором начато их исполнение;

шаг 2: Выполняются все запросы на экстренную запись регистров соответствующего устройства;

шаг 3: Продолжается прерванная работа клиента Modbus с очередным устройством.

Для каждого тега из секций HR и CO информационной базы Modbus может быть указана переменная флага требования экстренной записи (тип переменной – BOOL).

Служба клиента Modbus осуществляет мониторинг флагов экстренной записи и производит выполнение соответствующих запросов при значении флага «1».

После выполнения операции записи тега значение соответствующего флага необходимости экстренной записи устанавливается службой клиента Modbus равным «0».

Мониторинг флагов экстренной записи производится при одновременном выполнении следующих условий:

- закончено исполнение очереди запросов записи данных в регистры очередного подчиненного устройства на линии;
- очередь запросов на экстренную запись регистров пуста;
- в настоящее время не производится выполнение транзакции запроса к подчиненному устройству в рамках планового опроса.

При наличии флага экстренной записи выполнение планового опроса для всех устройств, находящихся на используемой линии связи, прерывается после окончания текущей транзакции протокола Modbus. Если при этом опрос устройства не полностью завершен, считанные данные не используются. В случае, если на линии связи несколько устройств, на которых должны быть выполнены плановые запросы записи, производится ожидание полного выполнения запросов на устройстве, где оно уже начато, затем выполняются все экстренные запросы записи для всех устройств на линии связи, затем продолжается выполнение запросов из штатной очереди записи.

6.9.1.10 Получение технологической программой подтверждения факта записи информационного тега в устройство

С целью реализации сложных алгоритмов процессов обмена технологическая программа может получать информацию о статусе выполнения запросов записи отдельных информационных тегов.

Для получения информации о статусе выполнения записи (с использованием TB4 или TB5) технологической программой производится следующая последовательность действий:

шаг 1: Модифицируются значения тегов;

шаг 2: Значения привязанных переменных статуса записи устанавливаются равными «0»;

шаг 3: Если используется TB5, значения привязанных переменных флага экстренной записи устанавливаются равными «1»;

шаг 4: Ожидается (в течение ограниченного таймаутом времени) появление ненулевого значения привязанной переменной статуса записи;

шаг 5: Если после истечения таймаута в переменной статуса записи наблюдается нулевое значение, диагностируется неисправность работы службы клиента Modbus;

шаг 6: Если значение переменной статуса записи не нулевое, статус операции записи анализируется согласно таблице 16. Если значение равно «1», запись успешно произведена.

При необходимости последовательной подтвержденной записи тегов в определенном порядке шаги 1-6 выполняются для переменной, модифицируемой в первую очередь, затем для переменной, модифицируемой во вторую очередь и т.д.

6.9.1.11 Получение технологической программой подтверждения факта обновления значения информационного тега

Для реализации сложных алгоритмов процессов обмена технологическая программа может получать информацию о статусе выполнения запросов чтения отдельных информационных тегов.

Для получения информации о статусе выполнения чтения одного или нескольких информационных тегов из подчиненного устройства (по ТВ1, ТВ2 или ТВ3) технологической программой выполняется следующая последовательность действий:

шаг 1: Значения привязанных переменных статуса чтения устанавливаются равными «0»;

шаг 2: Если используется ТВ2, значение входной переменной GRP_EXTRA_READ управляющего функционального блока устанавливается равным «1»;

шаг 3: Если используется ТВ3, значение привязанной переменной флага экстренного чтения устанавливается равным «1»;

шаг 4: Ожидается (в течение ограниченного таймаутом времени) появление ненулевого значения привязанной переменной статуса чтения;

шаг 5: Если после истечения таймаута в переменной статуса записи наблюдается нулевое значение, диагностируется неисправность работы службы клиента Modbus;

шаг 6: Если значение переменной статуса чтения не нулевое, статус операции чтения анализируется согласно таблице 16. Если значение равно «1», чтение успешно произведено.

Шаги 1-3 выполняются за один такт работы технологической программы.

6.9.1.12 Формирование произвольных запросов к подчиненным устройствам

При отправке произвольных информационных пакетов подключенному устройству служба поддерживает адресацию устройств и контроль целостности пакета. Этот тип взаимодействия может использоваться для выполнения функций протокола Modbus, не поддерживаемых службой клиента напрямую.

Для передачи подобных запросов и приема ответных пакетов в управляющем функциональном блоке находится ряд переменных:

- входные:
  - массив данных произвольного запроса SPEC_REQ_BUF;
  - длина произвольного запроса SPEC_REQ_CNT;

- выполнение произвольного запроса SPEC_REQ_START;

- выходные:
  - массив ответа подчиненного устройства на произвольный запрос SPEC_ANS_BUF;
  - длина полученного ответа на произвольный запрос SPEC_ANS_CNT;
  - статус выполнения произвольного запроса SPEC_ANS_STATUS.

Служба клиента Modbus обеспечивает расчет контрольной суммы CRC и выполнение функции временного разделения пакетов Modbus. Пользователю доступно конструирование запросов и разбор ответов.

Если произвольные запросы выполняются без остановки периодического опроса подчиненного устройства, произвольный запрос выполняется по окончании текущего цикла периодического опроса.

6.9.1.13 Выполнение операций службой клиента Modbus

Цикл операций выполняется службой клиента Modbus в процессе взаимодействия с подключенными устройствами в следующем порядке:

шаг 1: Определяется необходимость выполнения операций экстренной записи в регистры подчиненного устройства по ТВ5;

шаг 2: В случае необходимости последовательно выполняются операции экстренной записи по ТВ5;

шаг 3: Определяется необходимость экстренного выполнения произвольного запроса к подключенному устройству по ТВ7;

шаг 4: В случае необходимости выполняется экстренный произвольный запрос по ТВ7. Результат запроса записывается в БДРВ;

шаг 5: Определяется необходимость выполнения операций экстренного чтения регистров подчиненного устройства по ТВ2 или ТВ3;

шаг 6: В случае необходимости последовательно выполняются операции экстренного чтения по ТВ2, результаты записываются в БДРВ, приложение переходит к шагу 1 по окончании операций чтения;

шаг 7: Определяется необходимость выполнения плановых операций записи в регистры подчиненного устройства по ТВ4;

шаг 8: В случае необходимости последовательно выполняются операции записи по ТВ4;

шаг 9: Определяется необходимость выполнения планового произвольного запроса к подчиненному устройству по ТВ6;

шаг 10: В случае необходимости выполняется произвольный запрос по ТВ6. Результат запроса записывается в БДРВ;

шаг 11: Определяется необходимость выполнения планового опроса по ТВ1 (по сигналам ФБ, по таймеру или в случае прерывания предыдущей попытки опроса);

шаг 12: В случае необходимости последовательно выполняются операции планового чтения по ТВ1. При выполнении операции после каждой транзакции производится мониторинг обновления БДРВ и определяется необходимость выполнения операций экстренного чтения/записи регистров подчиненного устройства ТВ2, ТВ3, ТВ5 или отправки произвольного запроса по ТВ7;

шаг 13: При необходимости взаимодействия ТВ2, ТВ3, ТВ5 или ТВ7 прерывается плановый опрос и выполняются операции экстренного взаимодействия в последовательности, аналогичной шагам 1-6;

шаг 14: Приложение переходит к шагу 1.

6.9.1.14 Задание временных параметров работы, обработка неуспешно выполненных запросов, определение статуса «Устройство не на связи» (LINK_FAIL) для подключенного устройства

Для изменения временных характеристик опроса подчиненного устройства по протоколу Modbus во время исполнения технологического приложения в качестве входных переменных управляющего функционального блока добавляют следующие параметры:

- максимальный интервал времени между байтами одного пакета BYTE_TIMEOUT, мс;
- максимальное время ожидания ответного пакета ANSWER_TIMEOUT, мс;
- количество повторных попыток выполнения запросов RETRY, шт.

Также эти параметры находятся в окне конфигурирования клиента Modbus. Значения параметров, используемые по умолчанию в переменных управляющего блока, определяются значениями, введенными в окне конфигурирования.

В случае неудачного выполнения запроса служба клиента Modbus производит повторные попытки указанное количество раз.

Служба реализует алгоритм определения отсутствия связи с подключенным устройством согласно выражению: ЕСЛИ «Количество последовательных неуспешных запросов» БОЛЬШЕ, чем «Количество повторных попыток выполнения запросов», ТО устанавливается статус «Устройство не на связи». Если статус установлен, значение соответствующей выходной переменной блока управления LINK_FAIL записывается равным «1», иначе – «0».

6.9.1.15 Примеры взаимодействия технологического приложения и службы клиента протокола Modbus

Начальные условия для всех примеров: в окне конфигурирования клиента Modbus заданы настройки порта и настройки обмена по протоколу Modbus, а также привязки переменных проекта к информационным тегам.

Примеры

1 Регулярное автоматическое вычитывание информации из подчиненного устройства (ТВ1).

Начальные условия: установлено периодическое чтение информационной базы Modbus. Группы для тегов не заданы. Переменные статуса для тегов не заданы. Входные переменные управляющего блока не привязаны (используются значения по умолчанию). Выходная переменная LINK_FAIL привязана к переменной проекта, индицирующей состояние связи.

Работа системы: служба клиента Modbus периодически осуществляет вычитывание всех значений информационных тегов из регистров подчиненного устройства. Приложение использует в работе переменные, значения в которых обновляются после каждого опроса устройства. Приложение отслеживает статус обмена с помощью переменной, привязанной к флагу LINK_FAIL.

2 Запись информации в подчиненное устройство (ТВ1).

Начальные условия: в секциях HR или CO перечня тегов заданы информационные теги, к которым привязаны переменные приложения.

Работа системы: аналогично примеру 1. При модификации технологической программой переменных, привязанных к тегам из секций HR и CO, службой клиента Modbus автоматически генерируются запросы на их модификацию в устройстве, т.е. приложению необходимо только модифицировать значение. Весь процесс записи значений регистров происходит автоматически.

3 Получение информации о статусе выполнения операции чтения информационных тегов (ТВ1).

Начальные условия: те же, что в примере 1, за исключением того, что к тегам, для которых нужно осуществлять мониторинг статуса чтения, привязаны переменные статуса чтения.

Работа системы: аналогично примеру 1. Для отслеживания статуса или момента выполнения операции чтения технологическое приложение устанавливает значение привязанной к тегу переменной статуса чтения равным «0» и осуществляет ее мониторинг. Появление ненулевого значения означает, что служба клиента Modbus осуществила попытку опроса соответствующих регистров, и текущее значение отражает статус этой операции. Приложение использует в логике своей работы значение статуса чтения. Далее, при необходимости отслеживания окончания следующего опроса, операция повторяется.

4 Получение информации о статусе выполнения операции записи информационных тегов (ТВ4).

Начальные условия: те же, что в примере 2, за исключением того, что к тегам, для которых нужно осуществлять мониторинг статуса записи, привязаны переменные статуса записи.

Работа системы: аналогично примеру 2. Для отслеживания статуса или момента выполнения операции записи технологическое приложение после модификации привязанной переменной значения тега устанавливает значение привязанной переменной статуса записи тега равным «0» и осуществляет ее мониторинг. Появление ненулевого значения означает, что служба клиента Modbus осуществила попытку записи соответствующих регистров, и текущее значение отражает статус этой операции. Приложение использует в логике своей работы значение статуса записи. При необходимости отслеживания следующей записи тега операция повторяется.

5 Управляемое чтение информации из подчиненного устройства (ТВ1, ТВ2) путем последовательного опроса групп тегов.

Начальные условия: в зависимости от нужд приложения теги распределены в несколько групп. Переменные статуса для тегов не заданы. Ко входной переменной GRP_NUMBER привязан выход блока или другая переменная, определяющая номер опрашиваемой группы. Ко входной переменной GRP_UPD_ONCE привязан выход блока или другая переменная, используемая

технологическим приложением для формирования сигнала вычитывания групп. Технологическое приложение осуществляет мониторинг выходных переменных блока GRP_UPDATED и GRP_UPD_ERR с целью получения информации о статусе обмена.

Работа системы: программа технологического приложения построена таким образом, что осуществляет циклический перебор номеров всех имеющихся групп тегов. Для чтения очередной группы технологическое приложение устанавливает в значение переменной GRP_NUMBER нужный номер группы и устанавливает значение переменной GRP_UPD_ONCE равным «1» на один такт работы. Начинается опрос указанной группы. Для осуществления экстренного чтения группы (ТВ2) необходимо установить равным «1» значение входной переменной GRP_EXTRA_READ. В случае, если переменная GRP_EXTRA_READ равна «0», производится плановое чтение (ТВ1).

При этом значения переменных статуса чтения группы GRP_UPDATED и GRP_UPD_ERR равны «0». Технологическое приложение анализирует значения переменных GRP_UPDATED и GRP_UPD_ERR. Значение переменной GRP_UPDATED становится равным «1», если группа была успешно прочитана. Значение переменной GRP_UPD_ERR становится равным «1», если при чтении группы возникла ошибка. После этого технологическое приложение устанавливает в значение переменной GRP_NUMBER номер следующей группы, и процесс повторяется.

6 Экстренное чтение значения информационного тега с анализом статуса операции (ТВ3).

Начальные условия: для тега заданы привязанные переменные статуса чтения STATUS_READ и необходимости экстренного чтения EXTRA_READ_CMD.

Работа системы: для выполнения операции технологическое приложение в одном такте своей работы устанавливает значение переменной EXTRA_READ_CMD равным «1» и обнуляет STATUS_READ. Затем ожидается появление ненулевого значения переменной STATUS_READ и проводится анализ статуса выполнения операции. Если операция выполняется успешно, обновленное значение привязанной переменной значения тега используется в логике приложения, иначе – производится индикация ошибки.

Операция может быть одновременно выполнена на любом количестве тегов, находящихся в информационной базе. Для этого для всех обновляемых тегов необходимо задать собственные привязанные переменные статуса чтения и необходимости чтения и производить описанную выше операцию.

7 Экстренная запись значения информационного тега с анализом статуса операции (ТВ5).

Начальные условия: тег задан в секции HR или CO. Для тега заданы привязанные переменные статуса записи STATUS_WRITE и необходимости экстренной записи EXTRA_WRITE_CMD.

Работа системы: для выполнения операции технологическое приложение в одном такте своей работы устанавливает значение переменной EXTRA_WRITE_CMD равным «1» и обнуляет STATUS_WRITE. Далее ожидается появление ненулевого значения переменной STATUS_WRITE и проводится анализ статуса выполнения операции. Статус операции используется в логике работы технологического приложения.

Операция может быть одновременно выполнена на любом количестве тегов, находящихся в секциях HR и CO информационной базы. Для этого для всех записываемых тегов необходимо задать собственные привязанные переменные статуса записи и необходимости записи и производить описанную выше операцию.

6.9.2 Настройка и работа со службой обмена по последовательным портам

Служба обмена обеспечивает отправку массивов данных по последовательным портам связи, автоматически поддерживая обеспечение и контроль целостности данных путем расчета и добавления к пакету контрольных сумм и преамбул.

Окно настройки службы обмена по последовательным портам (рисунок 58) содержит панели «Параметры соединения» и «Параметры протокола».

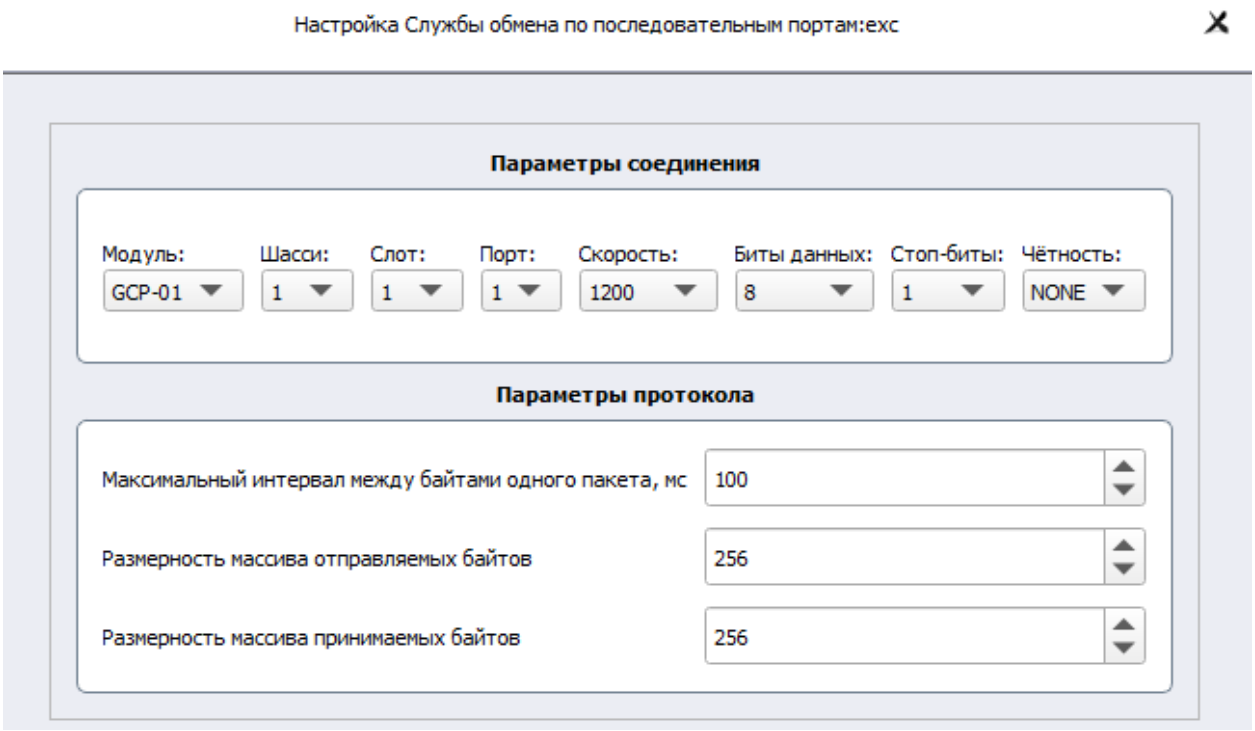

Рисунок 58 – Окно настройки службы обмена по последовательным портам

Панель «Параметры соединения» содержит следующие выпадающие списки:

- Модуль;
- Шасси;
- Слот;
- Порт;
- Скорость;
- Биты данных;
- Стоп-биты;
- Четность.

Панель «Параметры протокола» содержит поля:

- Максимальный интервал между байтами одного пакета, мс;
- Размерность массивов отправляемых байтов;
- Размерность массивов принимаемых байтов.

6.9.2.1 Универсальная служба обмена по последовательному порту

Для обеспечения возможности обмена с устройствами, работающими по протоколам связи без прямой поддержки, в ПЛК реализована универсальная служба обмена по последовательному порту. Такая служба позволяет отправлять массивы данных по последовательным интерфейсам связи, автоматически поддерживая обеспечение и контроль целостности данных путем расчета и добавления к пакету контрольных сумм и преамбул. Пользователь также может задать тип обеспечения целостности данных.

Универсальная служба позволяет выполнять операции обмена с любым количеством последовательных портов ПЛК путем создания для каждого порта отдельного экземпляра службы. Также последовательный порт может быть поочередно использован несколькими экземплярами службы, работающими с различными настройками.

Конфигурирование службы происходит на этапе старта или загрузки ПО контроллера. В процессе работы службы настройки параметров работы последовательного порта не изменяются.

Ниже приведен список параметров службы, задаваемых на этапе конфигурации (таблица 17).

Таблица 17 – Настраиваемые параметры службы

Наименование параметра	Мнемоника	Тип	Значение по умолчанию	Описание
Номер шасси	RACK	UINT	1	Номер шасси ПЛК, используемого для соединения
Номер слота	SLOT	UINT	1	Номер слота ПЛК, используемого для соединения
Номер порта	PORT	UINT	1	Номер порта ПЛК, используемого для соединения
DTR	DTR	BOOL	0	Состояние DTR (параметр активен для портов, имеющих этот сигнал)
RTS	RTS	BOOL	0	Состояние RTS (параметр активен для портов, имеющих этот сигнал)
Четность	PARITY	STRING	NONE	NONE – без бита четности; ODD – контроль четности (проверка на нечетность);

Наименование параметра	Мнемоника	Тип	Значение по умолчанию	Описание
				EVEN – контроль четности (проверка на четность).
Режим порта	MODE	STRING	RS-485	Режим работы порта. Допустимы значения RS-485 и RS-232.
Скорость работы порта	BAUDRATE	INT	9600	Скорость работы порта. Выбирается из списка.
Кол. бит	DATABITS	USINT	8	Количество бит в передаваемом байте. Может принимать значения 8 или 7.
Кол. стоповых бит	STOPBITS	USINT	1	Количество стоповых бит. Может принимать значения 1 и 2.
Таймаут между байтами, мс	BYTE_ TIMEOUT	INT	50	Интервал времени, используемый для разделения входных пакетов. Если в течение данного интервала на порт не поступают данные, служба считает пакет полученным.
Разрешение работы службы	EN_SERVICE	BOOL	1	Управление функционированием службы: 1 - функционирование разрешено, порт открыт; 0 – функционирование запрещено, порт закрыт.

Наименование параметра	Мнемоника	Тип	Значение по умолчанию	Описание
Размерность массива отправляемых байтов	SEND_DIM	UINT	256	Используется для типизации входа SEND_BUF функционального блока и для создания внутреннего буфера
Размерность массива принимаемых байтов	RECV_DIM	UINT	256	Используется для типизации выхода RECV_BUF функционального блока и для создания внутреннего буфера

Служба допускает управление собственным функционированием. Если служба функционирует, она открывает порт для использования и осуществляет обработку отправляемой и получаемой информации. По окончании функционирования порт закрывается, отправка и прием данных не производятся.

Если системе требуется работать с устройствами, подключенными к одному порту и имеющими разные настройки порта, то для этого создаются разные экземпляры службы с отличающимися настройками. Для работы с устройством должно быть включено функционирование соответствующего экземпляра службы и выключено функционирование остальных экземпляров, использующих тот же порт, т.к. одновременное функционирование нескольких экземпляров службы на одном порте не допускается.

Для взаимодействия технологической программы со службой используется соответствующий функциональный блок (рисунок 59). Блок может быть использован в программах, написанных на языках программирования стандарта МЭК 61131-3. В среде R-Logic.Designer для каждого экземпляра службы в библиотеке проекта автоматически создается функциональный блок с соответствующим наименованием.

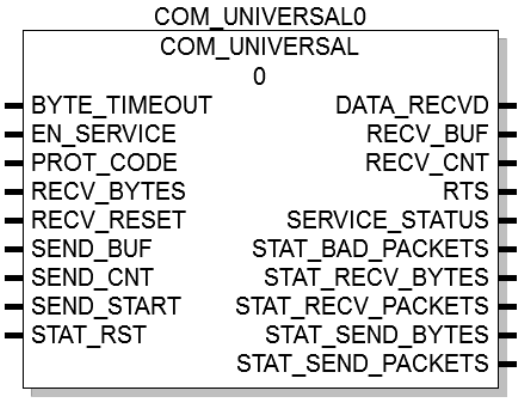

Рисунок 59 – Функциональный блок универсальной службы обмена по последовательному порту

Ниже приведен список входных и выходных параметров функционального блока (таблица 18, таблица 19).

Таблица 18 – Входные параметры функционального блока

Наименование параметра	Мнемоника	Тип	Значение по умолчанию	Описание
Разрешение работы службы	EN_SERVICE	BOOL	В соответствии с настроечным параметром	Управление функционированием службы: 1 - функционирование разрешено, порт открыт; 0 – функционирование запрещено, порт закрыт.
Массив отправляемых байтов данных	SEND_BUF	BYTE [X]	[0..0]	Массив, содержащий данные, подлежащие передаче. Размерность массива X задается на этапе создания экземпляра службы.
Количество отправляемых байтов данных	SEND_CNT	UINT	0	Количество байт с начала массива SEND_BUF, которые будут переданы в порт
Сигнал отправки пакета	SEND_ START	BOOL	0	1 – происходит передача информационного пакета

Наименование параметра	Мнемоника	Тип	Значение по умолчанию	Описание
Ожидаемое количество байт при приеме	RECV_ BYTES	UINT	Соответствует размерности массива	Используется для получения заданного количества байт из порта. Если количество принятых байт равно значению этого параметра, служба считает пакет полученным. Если параметр равен нулю, чтение из порта не производится.
Таймаут между байтами, мс	BYTE_ TIMEOUT	UINT	Соответствует настроечному параметру	Интервал времени, используемый для разделения входных пакетов. Если в течение данного интервала на порт не поступают данные, служба считает пакет полученным.
Код типа протокола передачи	PROT_CODE	UINT	0	Обеспечение контроля целостности пакетов на уровне службы. Коды приведены в ниже (таблица 20).
Сигнал сброса принятой информации	RECV_ RESET	BOOL	0	1 – накопленная в буфере приема информация удаляется
Сигнал сброса счетчиков статистики	STAT_RESET	BOOL	0	1 – производится сброс накопленной статистической информации

Таблица 19 – Выходные параметры функционального блока

Наименование параметра	Мнемоника	Тип	Описание
Код статуса функционирования службы	SERVICE_ STATUS	USINT	0 – служба не запущена; 1 – служба запущена, функционирование разрешено; 2 – служба запущена, функционирование не разрешено; 5 – служба запущена, ошибка открытия порта. В случае расширения списка кодов он будет обновлен в справочной системе R-Logic.Designer.
Массив полученных байтов данных	RECV_BUF	BYTE [X]	Массив, содержащий принятые данные. Размерность массива X задается на этапе создания экземпляра службы.
Сигнал получения пакета данных	DATA_RECVD	BOOL	Устанавливается равным «1» на один цикл исполнения технологической программы после получения целостного пакета данных
Количество полученных байтов в пакете	RECV_CNT	UINT	Количество байт данных в принятом пакете (исключая преамбулы и контрольные суммы, используемые для контроля целостности)
Сигнал готовности к передаче	RTS	BOOL	Возможность передачи пакета. Устанавливается равным «1» после опустошения выходного буфера. Обнуляется после получения сигнала отправки.

Наименование параметра	Мнемоника	Тип	Описание
Количество полученных пакетов	STAT_SEND_ PACKETS	ULINT	Общее количество отправленных пакетов. Обнуляется по сигналу STAT_RESET.
Количество отправленных пакетов	STAT_RECV_ PACKETS	ULINT	Общее количество полученных целостных пакетов. Обнуляется по сигналу STAT_ RESET.
Количество полученных поврежденных пакетов	STAT_BAD_ PACKETS	ULINT	Общее количество полученных пакетов с нарушением целостности. Обнуляется по сигналу STAT_ RESET.
Количество полученных байтов данных	STAT_SEND_ BYTES	ULINT	Общее количество отправленных байтов с учетом заголовков и контрольных сумм. Обнуляется по сигналу STAT_ RESET.
Количество отправленных байтов данных	STAT_RECV_ BYTES	ULINT	Общее количество полученных байтов с учетом заголовков и контрольных сумм, без учета целостности пакетов. Обнуляется по сигналу STAT_ RESET.

Таблица 20 – Коды типов протоколов передачи

Код	Протокол
0	Контроль целостности не производится
1	CRC8 в конце пакета, расчет по всем байтам пакета
2	CRC16 в конце пакета, расчет по всем байтам пакета
3	CRC32 в конце пакета, расчет по всем байтам пакета
4	HART, преамбула 8 байт 0xFF и контрольная сумма XOR по всем байтам кроме преамбулы в конце пакета

Функционирование службы происходит по следующему алгоритму:

шаг 1: Служба проверяет значение флага управления функционированием EN_SERVICE. Если значение равно «1», служба открывает порт с заданными настройками и переходит к шагу 2. Иначе служба производит закрытие ранее открытого порта. Выполнение пункта повторяется. Обновляется код состояния службы SERVICE_STATUS;

шаг 2: Служба проверяет значение сигнала отправки пакета SEND_START. Если он равен «1», то служба производит сброс сигнала готовности к передаче RTS, выполняет формирование передаваемого пакета с использованием указанного в параметре SEND_CNT количества байт из буфера SEND_BUF согласно коду протокола PROT_CODE. Сформированный пакет отправляется в коммуникационный порт. Обновляется статистическая информация в параметрах STAT_SEND_BYTES, STAT_SEND_PACKETS. По окончании отправки устанавливается сигнал готовности к передаче RTS;

шаг 3: Служба осуществляет чтение из порта. Если количество полученных байтов равно заданному в параметре RECV_BYTES или таймаут выполнения операции превышает значение, заданное в параметре BYTE_TIMEOUT, то пакет считается полученным. Производится анализ его целостности согласно коду протокола PROT_CODE. Если пакет целостный и отсутствует сигнал сброса принятой информации RECV_RESET (равен «0»), его содержимое размещается в буфере RECV_BUF, устанавливается значение количества полученных байтов RECV_CNT, устанавливается равным «1» значение сигнала получения пакета данных DATA_RECVD. Обновляется статистическая информация в выходных параметрах STAT_RECV_BYTES, STAT_RECV_PACKETS, STAT_BAD_PACKETS;

шаг 4: Служба проверяет значение сигнала сброса счетчиков статистической информации STAT_RESET и, если оно равно «1», обнуляет счетчики STAT_SEND_BYTES, STAT_SEND_PACKETS, STAT_RECV_BYTES, STAT_RECV_PACKETS и STAT_BAD_PACKETS;

шаг 5: Приложение переходит к шагу 1.

6.9.3 Настройка и работа со службой публикации

Служба публикации по умолчанию находится в дереве проекта. Двойным щелчком по ней открывается окно настройки. Окно содержит вкладки «Параметры соединения» и «Конфигурация».

Вкладка «Параметры соединения» (рисунок 60) используется для настройки подключения ПЛК к внешнему устройству. Она содержит поле «Корневая тема публикации», в котором может быть задана тема, в которую необходимо добавить переменные. Для контроллера и внешнего устройства темы должны быть одинаковыми, также предусмотрена возможность регулирования уровня вложенности тем путем добавления «/» между словами.

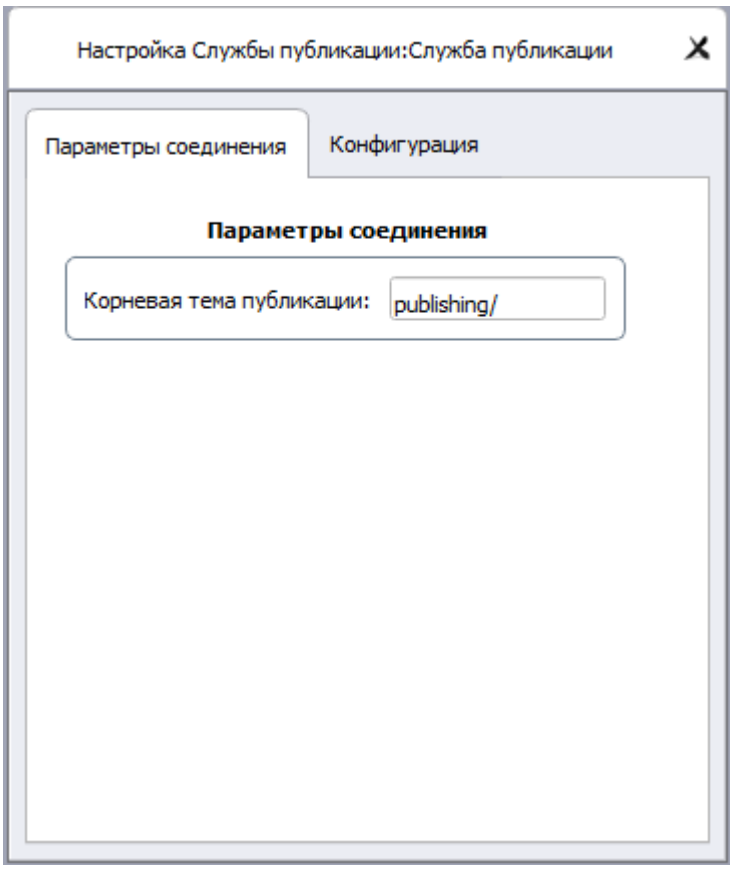

Рисунок 60 – Окно настройки службы публикации (вкладка «Параметры соединения»)

Вкладка «Конфигурация» (рисунок 61) позволяет добавлять переменные в тему.

В левой части вкладки «Конфигурация» находится таблица со столбцами:

- «Тема сообщений» – обеспечивает уникальность (уровень вложенности регулируется добавлением «/» между словами);
- «Тип публикации» – выбор между бинарным и строковым;
- «Направление» – выбор между чтением и записью;
- «Имя переменной»;
- «Тип переменной»;
- «Описание переменной».

При нажатии ПКМ по строке таблицы появляется команда «Удалить».

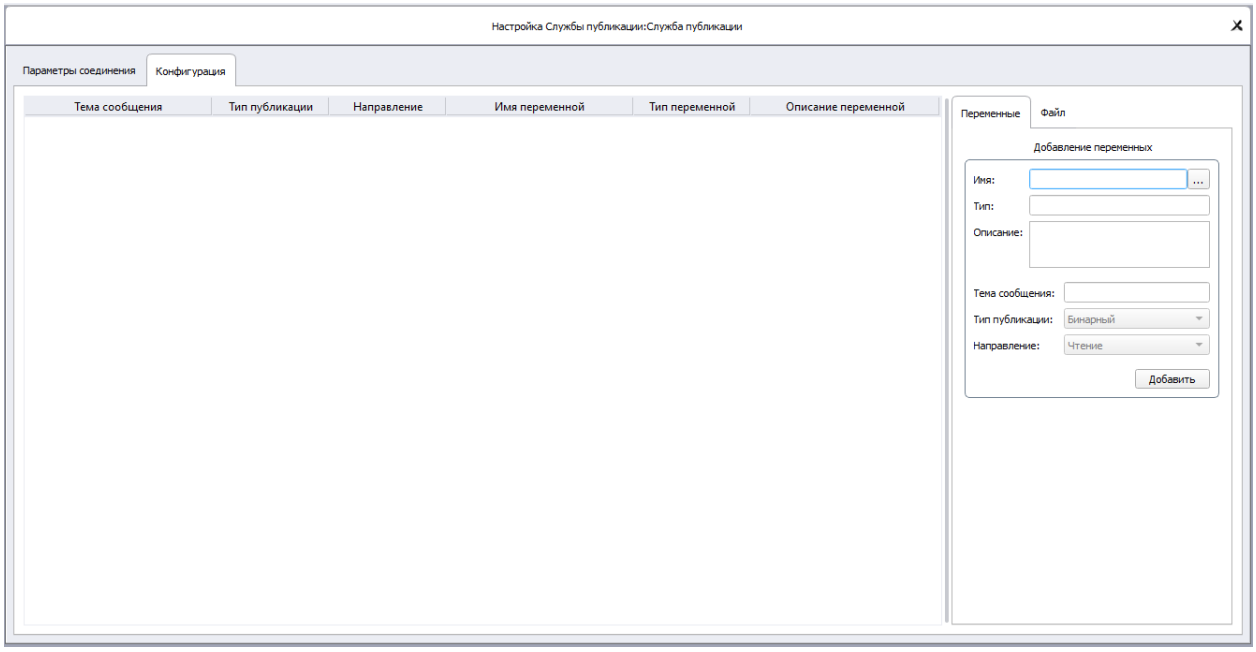

Рисунок 61 – Окно настройки службы публикации (вкладка «Конфигурация»)

В правой части вкладки «Конфигурация» располагаются вкладки «Переменные» и «Файл».

Вкладка «Переменные» позволяет добавить переменные. В ней находятся:

- поле «Имя» с кнопкой «…» – при нажатии открывается диалоговое окно «Список переменных», в котором пользователь выбирает добавляемую переменную;
- поля «Тип» и «Описание» – автоматически заполняются при заполнении поля «Имя»;
- поле «Тема сообщения» – по умолчанию соответствует полю «Имя»; если заданная тема сообщения уже добавлена в табличную форму, то поле выделяется красным цветом;
- выпадающий список «Тип публикации» с выбором между бинарным и строковым;
- выпадающий список «Направление» с выбором между чтением и записью;
- кнопка «Добавить».

Вкладка «Файл» содержит кнопки «Импортировать файл» и «Экспортировать файл», которые отвечают за импорт и экспорт файлов описания службы публикаций соответственно.

6.9.4 Настройка и работа со службой BACnet клиент

В окне службы BACnet клиент (рисунок 62) находятся вкладки «Параметры соединения» и «Конфигурация».

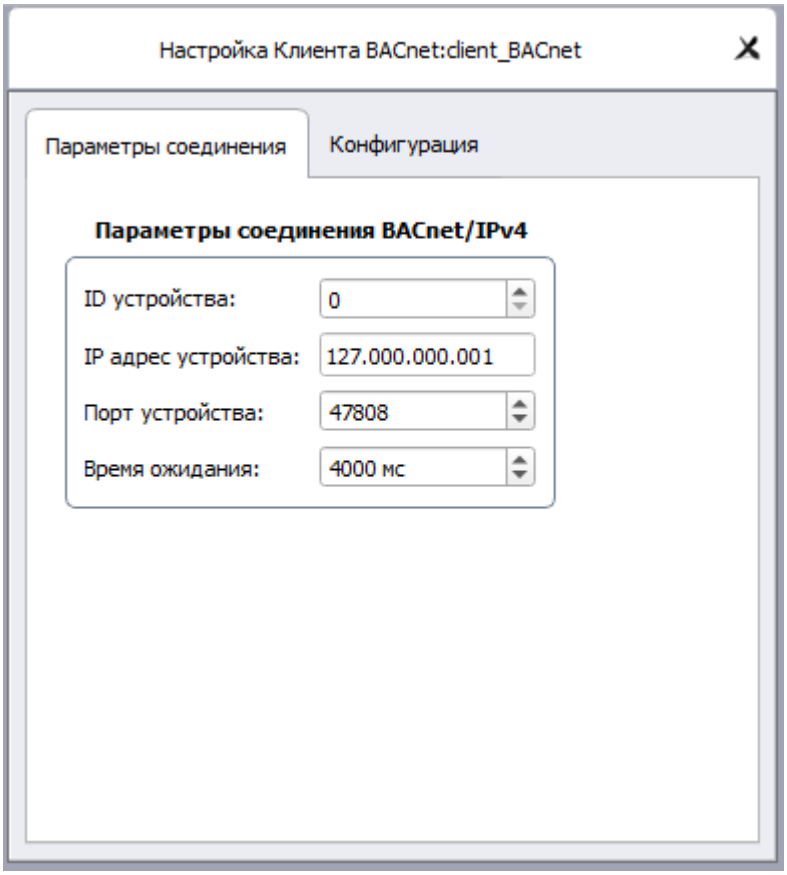

Рисунок 62 – Окно настройки службы BACnet клиент (вкладка «Параметры соединения»)

Вкладка «Параметры соединения» содержит настраиваемые поля:

- «ID устройства»;
- «IP адрес устройства»;
- «Порт устройства»;
- «Время ожидания».

Для работы с устройством по протоколу BACnet необходимо, чтобы устройство содержало объекты.

В левой части вкладки «Конфигурация» (рисунок 63) находится табличная форма со столбцами:

- «Тип объекта» – ID объекта, заложенного в BACnet;
- «Экземпляр объекта» – номер объекта в сети;
- «Идентификатор свойства» – справочная информация;
- «Значение свойства» – полное значение или элемент массива;
- «Индекс массива свойства» – активен только при выборе элемента массива;
- «Направление» – выбор между чтением и записью;
- «Описание переменной».

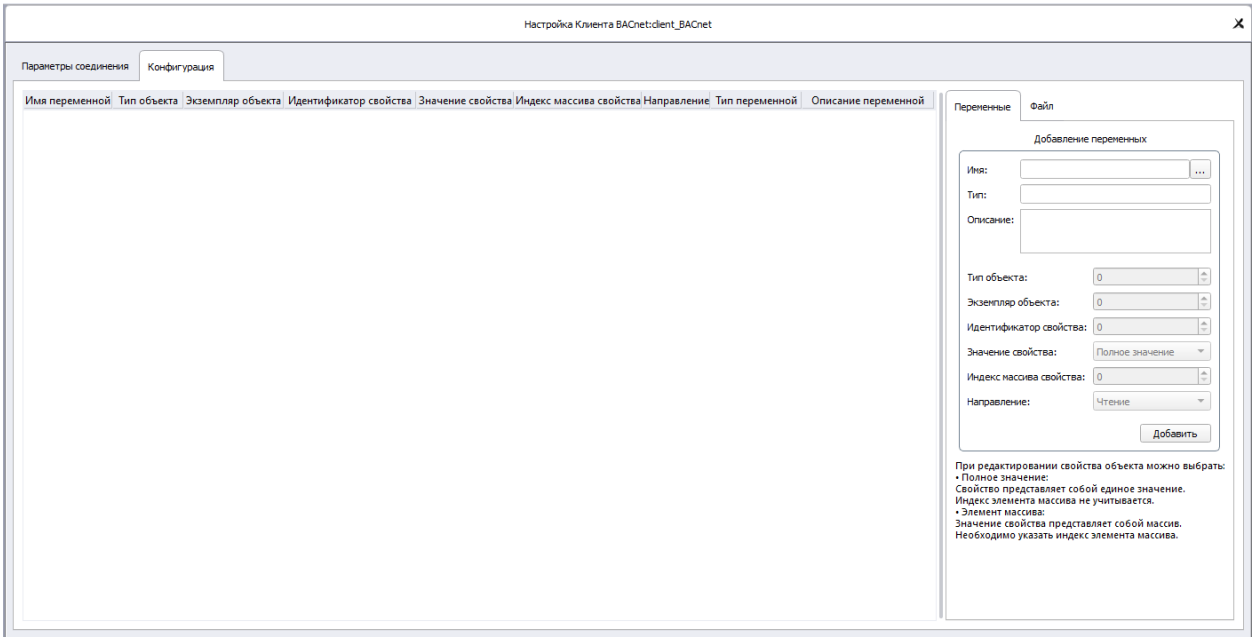

Рисунок 63 – Окно настройки службы BACnet клиент (вкладка «Конфигурация»)

В правой части вкладки «Конфигурация» располагаются вкладки «Переменные» и «Файл».

Вкладка «Переменные» содержит панель «Управление переменными», в которой находятся:

- поле «Имя» с кнопкой «…» – при нажатии открывается диалоговое окно «Список переменных», в котором пользователь выбирает добавляемую переменную;
- поля «Тип» и «Описание» – автоматически заполняются при заполнении поля «Имя переменной»;
- поле «Тип объекта»;
- поле «Экземпляр объекта»;
- поле «Идентификатор свойства»;
- выпадающий список «Значение свойства» – выбор между вариантами «Полное значение» и «Элемент массива»;
- поле «Индекс массива свойства»;
- выпадающий список «Направление» – выбор между чтением и записью;
- кнопка «Добавить».

На вкладке «Файл» находятся кнопки «Импортировать файл» и «Экспортировать файл» для работы с файлами описания клиента BACnet.

По нажатию ПКМ на строку в таблице появляется команда «Удалить».

6.9.5 Настройка и работа со службой управления сохраняемыми параметрами

В левой части окна конфигуратора службы сохранения (рисунок 64) находится таблица со столбцами:

- «Смещение» – номер байта, с которого переменная начинает записываться в памяти и который может служить индикатором для конкретной переменной;
- «Название» – имя переменной;
- «Тип»;
- «Начальное значение»;
- «Текущее значение»;
- «Описание».

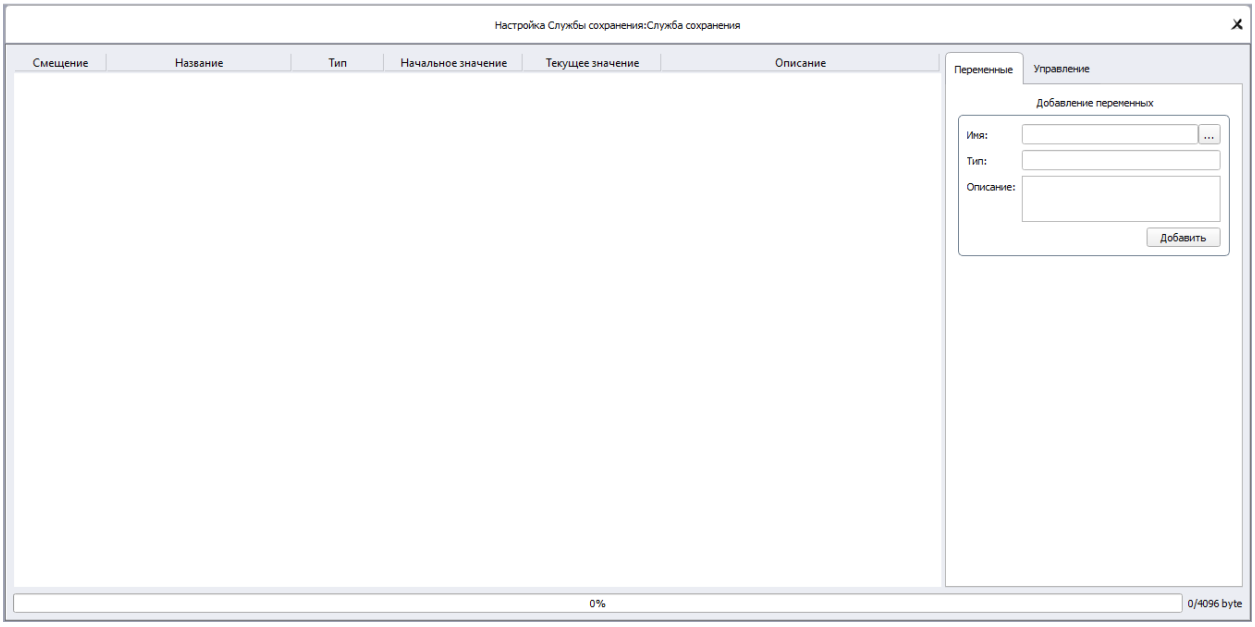

Рисунок 64 – Окно настройки службы управления сохраняемыми параметрами

Окно службы управления сохраняемыми параметрами разделено на две части.

В таблице отображаются только те переменные, у которых в свойствах напротив параметра «Сохранение» активирован флажок.

В правой части окна находятся вкладки «Переменные» и «Управление».

Вкладка «Переменные» позволяет добавить переменную. Она содержит:

- поле «Имя» с кнопкой «…» – при нажатии открывается диалоговое окно «Список переменных», в котором пользователь выбирает добавляемую переменную;
- поля «Тип» и «Описание» – автоматически заполняются при заполнении поля «Имя переменной»;
- кнопку «Добавить».

Во вкладке «Управление» находятся кнопки:

- кнопка «Прочитать значение из ПЛК» – замена значений переменных из таблицы на значения переменных, записанных на ПЛК, при совпадении имен (переменные из ПЛК, не отраженные в таблице, игнорируются);

- кнопка «Записать значения на ПЛК» – замена значений переменных, записанных на ПЛК, на значения переменных из таблицы, при совпадении имен (переменные из таблицы, не отраженные в ПЛК, игнорируются);
- кнопка «Экспортировать в файл» – сохранение XML-файла с обозначенными в таблице переменными;
- кнопка «Импортировать из файла» – восстановление переменных из XML-файла.

В нижней части окна расположен индикатор заполнения памяти контроллера сохраненными переменными, при малом количестве свободной памяти появляется предупреждающее сообщение.

6.9.6 Диалог конфигурирования протокола МЭК 61850

Вкладка «Параметры соединения» (рисунок 65) окна настройки сервера МЭК 61850 содержит:

- раздел «Параметры протокола MMS» с полями для ввода IP-адреса и номера порта;
- раздел «Импорт файла МЭК 61850».

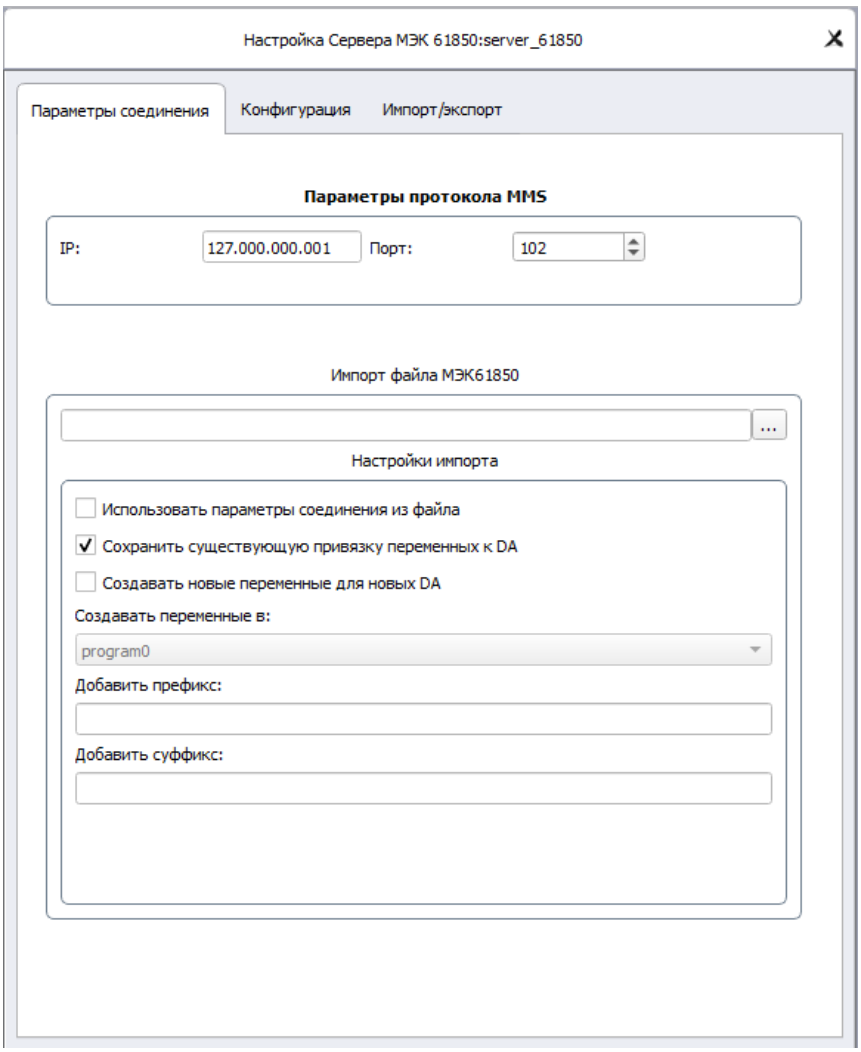

Рисунок 65 – Окно настройки элемента службы «Серверы МЭК 61850»

Для настройки по протоколу МЭК 61850 пользователь задает IP-адрес сервера. Им определяется сетевой интерфейс, который будет использован для сервера. Возможно два варианта конфигурации:

1) если указан IP-адрес «0.0.0.0», сервер будет прослушивать все сетевые интерфейсы;

2) если указан IP-адрес одного из сетевых интерфейсов, сервер будет доступен только через указанный IP.

Затем задается номер порта (по умолчанию 102).

Конфигурирование дерева тегов по протоколу МЭК 61850 производится путем импорта файла с расширением .icd. Перед этим задаются флажками настройки:

- «Использовать параметры соединения из файла» – при активированном флажке после импорта IP-адрес и номер порта изменятся на соответствующие значения из файла;
- «Сохранить существующую привязку переменных к DA» (по умолчанию активирован) – при повторном импорте файла для существующих переменных сохраняется привязка;
- «Создавать новые переменные для DA» – для всех импортированных элементов протокола создаются переменные в программе, выбранной из выпадающего списка ниже, с автоматической привязкой. Имена переменных формируются конкатенацией пути к атрибуту данных. При указании суффикса и/или префикса эти строки добавляются к имени.

Импортируемый файл с расширением .icd содержит описания характеристик и технических возможностей устройства в соответствии с протоколом МЭК 61850 и составляется с использованием сторонних программ, например, ICD Designer. Импорт осуществляется нажатием на кнопку «…», после чего открывается окно выбора файлов с компьютера. После выбора путь к импортируемому файлу появится в соответствующем поле, и автоматически произойдет операция импорта.

После импорта во вкладке «Переменные» в узле соответствующей программы появятся все переменные, соответствующие загруженному ICD-файлу.

Во вкладке «Конфигурация» (рисунок 66) содержится модель логических узлов со вложенными в них переменными, заданная импортируемым файлом (раздел «Модель данных»). Для конфигурирования необходимо произвести привязку каждого из узлов к переменным в теле программы. Для привязки переменных пользователь щелчком мыши выбирает нужный узел, нажимает кнопку «…» в разделе «Привязанные переменные» в правой нижней части основного рабочего окна и выбирает соответствующую по типу переменную, при необходимости также задает описание переменной.

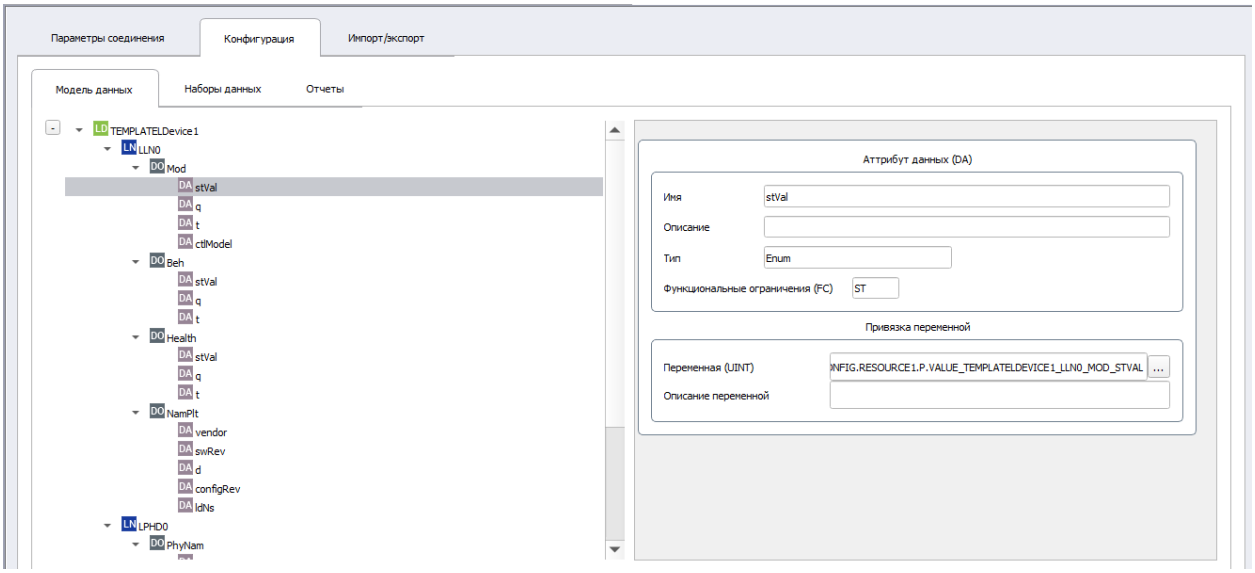

Рисунок 66 – Модель данных элемента службы «Серверы МЭК 61850»

В разделе «Наборы данных» (рисунок 67) представлены наборы данных, созданные при конфигурировании импортированного ICD-файла.

Набор данных DATA-SET – это упорядоченная группа объектных ссылок ObjectReferences данных DATA или атрибутов данных DataAttributes (называемых элементами набора данных), организованных как отдельный комплект для удобства клиента. Предназначены для оптимизации передачи данных, а также для определения значений данных DATA или атрибутов данных DataAttributes, которые должны передаваться в случае изменения значения одного из элементов.

В разделе «Отчеты» при нажатии на отчет (в дереве IED-устройства обозначен как «RP») появляется панель «Контрольный блок отчета (RCB)», содержащая:

- поле «Идентификатор (Report ID)» – имя отчета;
- поле «Набор данных (Data Set)» – путь к набору данных, данные которого будут передаваться в отчете;
- поле «Период сохранности (IntgPd), мс» – временной период, используемый для выдачи отчета о сохранности (отчет о сохранности включает значения всех элементов соответствующего набора данных);
- поле «Время буффера (BufTm), мс» – интервал времени для буферизации внутренних уведомлений, вызванных изменением данных, изменением качества, обновлением данных блоком управления буферизованным отчетом BRCB, для включения в один отчет;
- поле «Прич. отправки (TrgOps)» – причины отправки отчета (изменение данных, изменение качества, обновление данных, сохранность, общий опрос);
- поле «Опции (OptFlds)» – включенные в отчет опциональные поля (sequence-number, report-timestamp, reason-for-inclusion, data-set-name, data-reference, buffer-overflow или entrylD).

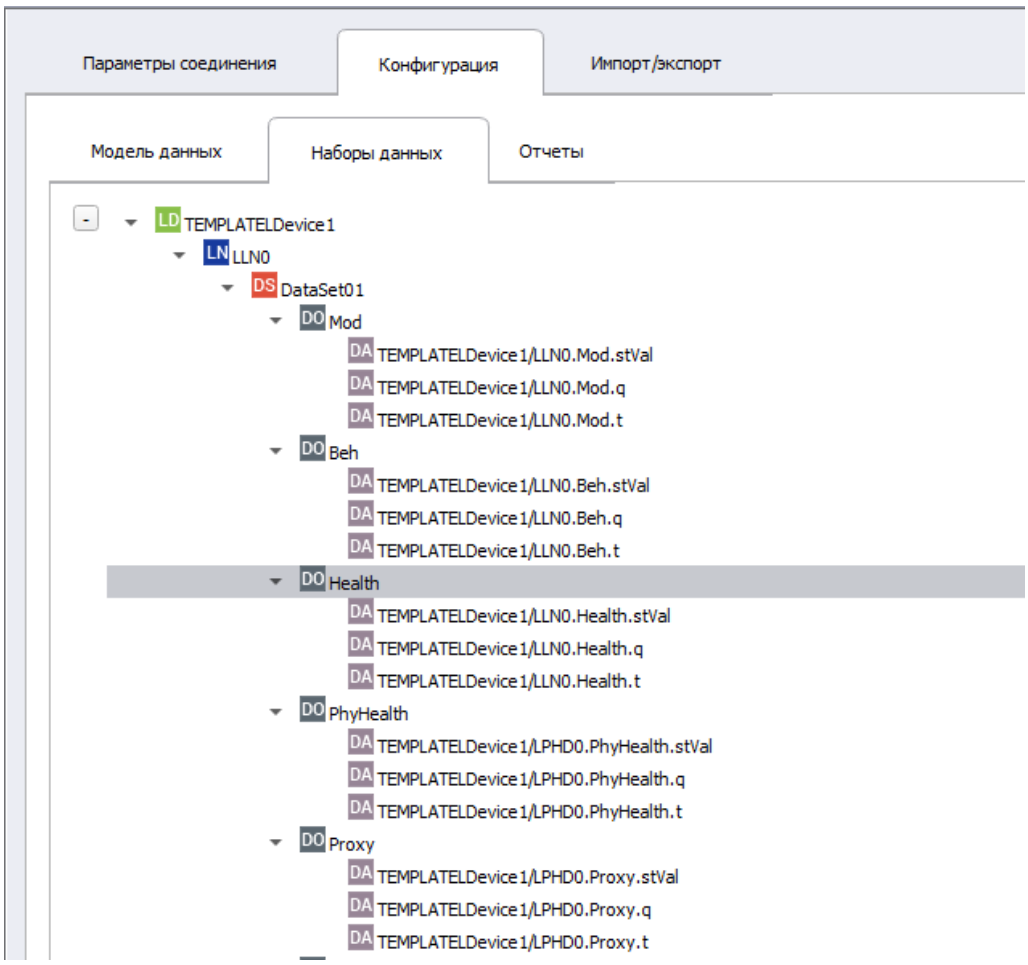

Рисунок 67 – Наборы данных элемента службы «Серверы МЭК 61850»

ВАЖНО! Для корректной передачи данных наличие набора данных обязательно.

Вкладка «Импорт/экспорт» позволяет экспортировать ICD-файл из проекта, выбрав путь для его сохранения.

6.9.7 Диалог конфигурирования протоколов МЭК 60870-5-101 и МЭК 60870-5-104

Протоколы МЭК 60870-5-101 и МЭК 60870-5-104 обеспечивают передачу массивов данных по последовательным портам связи, автоматически поддерживая обеспечение и контроль целостности данных путем расчета и добавления к пакету контрольных сумм и преамбул.

В дереве проекта возможно добавление нескольких элементов служб «Клиенты МЭК 60870-5-101», «Серверы МЭК 60870-5-101», «Клиенты МЭК 60870-5-104» и «Серверы МЭК 60870-5-104».

Окно настройки службы (рисунок 68) содержит вкладки «Параметры соединения», «Контроль», «Измерения» и «Импорт/экспорт».

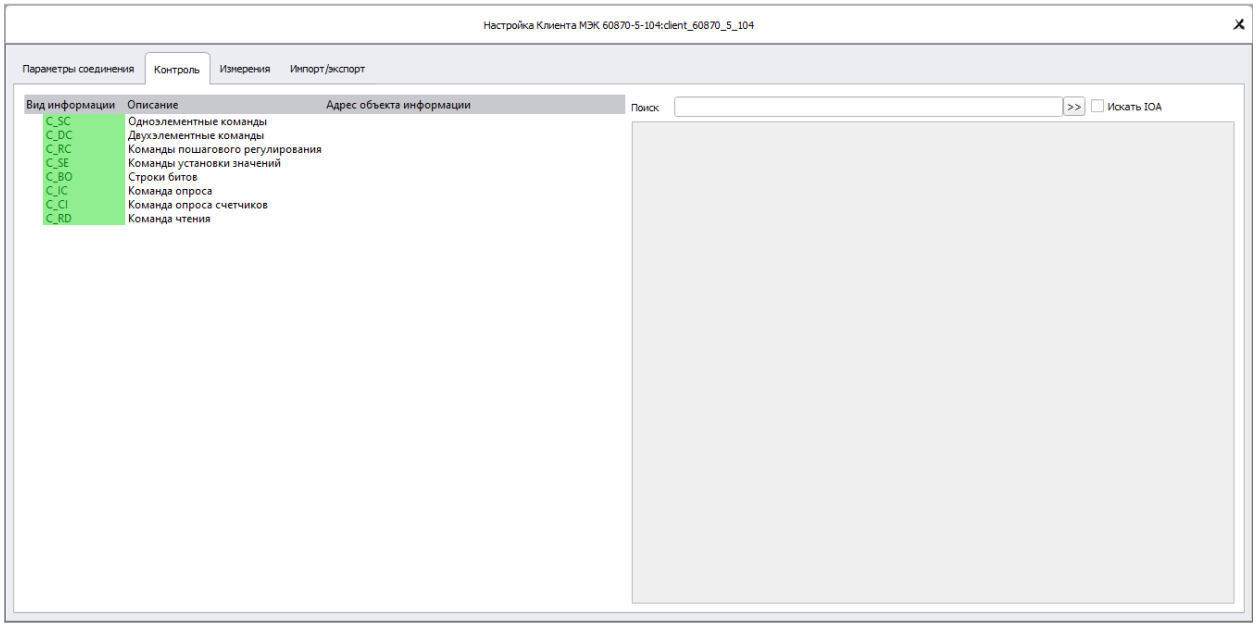

Рисунок 68 – Окно настройки элемента службы «Клиенты МЭК 60870-5-104»

Вкладка «Параметры соединения» служб «Клиенты МЭК 60870-5-101» и «Серверы МЭК 60870-5-101» содержит (рисунок 69) следующие выпадающие списки:

- Шасси;
- Слот;
- Порт;
- Скорость.

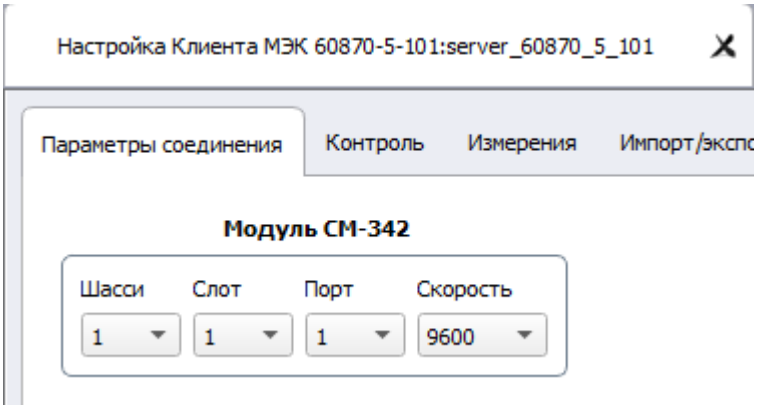

Рисунок 69 – Вкладка «Параметры соединения» окна настройки службы по протоколу МЭК 60870-5-101

Вкладка «Параметры соединения» служб «Клиенты МЭК 60870-5-104» и «Серверы МЭК 60870-5-104» содержит (рисунок 70):

- выпадающие списки «Шасси» и «Слот», задающие расположение коммуникационного модуля для настройки соединения;
- поле «IP адрес» для ввода IP-адреса устройства, от которого разрешается принимать входящие запросы;

- поле «Порт»;
- поле «Адрес станции»;
- поля задания параметров:
  - t0 – таймаут при установлении соединения – время ожидания ответа на STARTDT;
  - t1 – таймаут при посылке или тестировании APDU («нет ответа на команду») – превышение времени ожидания ответа на команду приводит к разрыву соединения;
  - t2 – таймаут отправки подтверждения – подтверждение о приеме пакета S-PDU отсылается либо по истечении этого таймера, либо по приеме предельного количества PDU, указанного в параметре W. При этом t2 < t1;
  - t3 – таймаут неактивности сервера – в случае долгого простоя как вышестоящее, так и подчиненное устройства могут отсылать блоки тестирования для проверки связи. Если в течение интервала времени t3 нет никакого обмена, то подчиненное устройство может закрыть соединение;
  - K – количество неподтвержденных посланных PDU – если было отослано количество PDU, указанное в параметре K, и не пришло подтверждение, то отсылка приостанавливается (ожидается подтверждение);
  - W – количество принятых PDU, которые необходимо подтверждать.
- поле «Длина очереди сообщений» – размер очереди неотправленных данных, которая заполняется при отсутствии связи с вышестоящим устройством по основному и резервному каналам. Если размер очереди меньше планируемого количества элементов данных, то часть данных не будет отправлена.

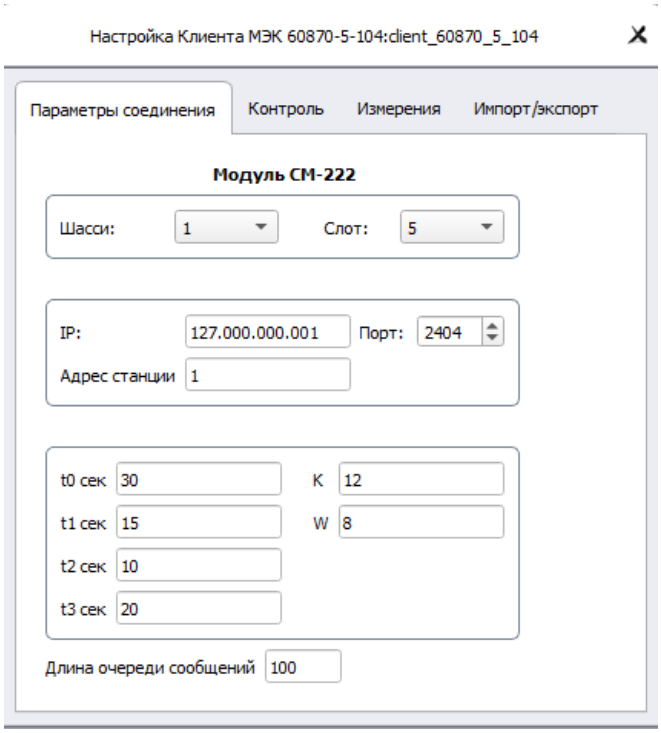

Рисунок 70 – Вкладка «Параметры соединения» окна настройки службы по протоколу МЭК 60870-5-104

IP-адресом определяется сетевой интерфейс, который будет использован для сервера. Возможно два варианта конфигурации:

1) если указан IP-адрес «0.0.0.0», сервер будет прослушивать все сетевые интерфейсы;

2) если указан IP-адрес одного из сетевых интерфейсов, сервер будет доступен только через указанный IP.

Затем задается номер порта (по умолчанию 2404).

Вкладки «Контроль» и «Измерения» представлены в виде таблиц. Столбец «Вид информации» содержит обозначение, столбец «Описание» – расшифровку вида информации, а столбец «Адрес объекта информации» – заданный при добавлении ASDU адрес. При нажатии на ячейку открывается окно настройки соответствующего объекта информации (рисунок 71).

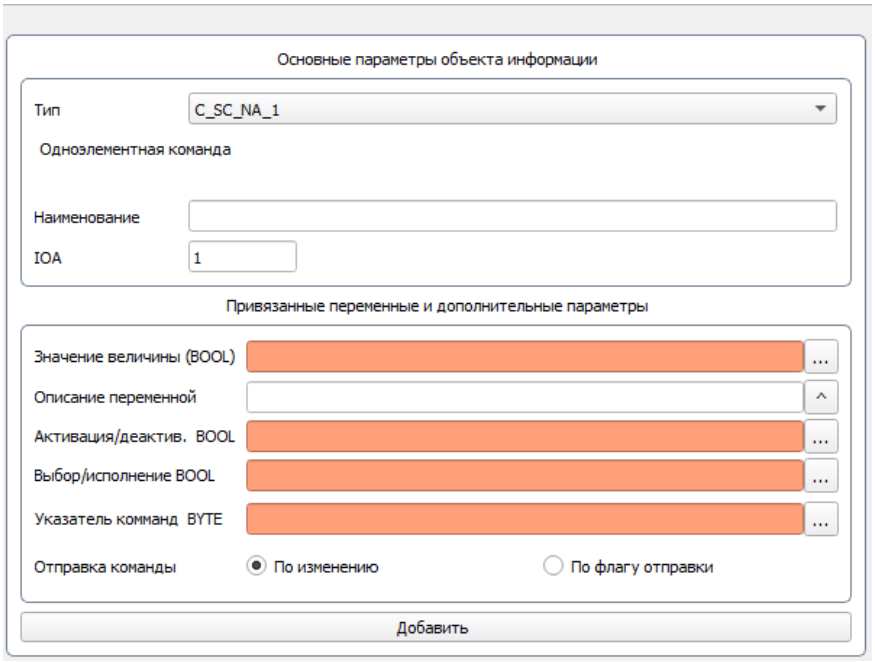

Рисунок 71 – Окно настройки объекта информации (вкладка «Контроль»)

Раздел «Основные параметры объекта информации» содержит выпадающий список «Тип», в котором перечислены типы идентификаторов, относящиеся к выбранному виду информации. В том же разделе пользователь может задать наименование объекта и адрес объекта информации.

Раздел «Привязанные переменные и дополнительные параметры» позволяет привязывать переменные ППО для передачи значений по указанному адресу. При выборе существующего объекта информации в нижней части окна появляется кнопка «Удалить».

В правой части вкладок «Контроль» и «Измерения» находится поле поиска, которое при нажатии на кнопку «>>» и активации флажка «Искать IOA» запускает поиск по адресу объекта информации.

В общем случае к информационному объекту может быть привязано несколько переменных.

При попытке привязать переменную ПЛК несоответствующего типа появляется предупреждение. Поле, отображающее наименование переменной, в каждый момент времени контролирует тип переменной, наличие ее в программе и уведомляет пользователя о обнаруженных нарушениях путем изменения цвета заливки.

Вкладка «Импорт/экспорт» (рисунок 72) содержит кнопки «Импортировать файл» и «Экспортировать файл», позволяющие выбрать расположение XML-файла для импорта и путь экспорта соответственно.

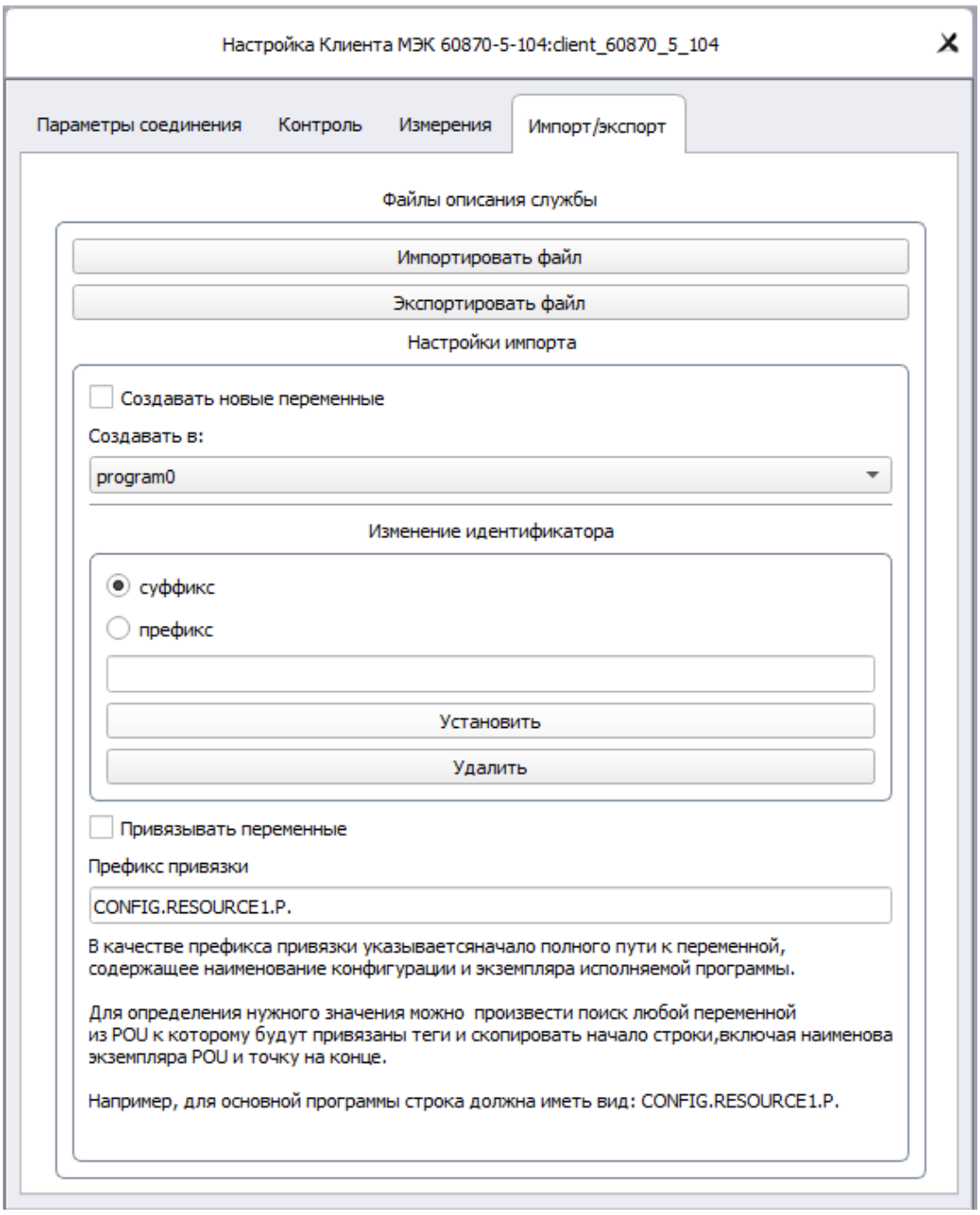

Рисунок 72 – Вкладка «Импорт/экспорт» окна настройки службы

В разделе «Настройки импорта» пользователь может активировать флажок «Создавать новые переменные», что позволяет автоматически создавать новые переменные при импорте файла, а также выбрать из выпадающего списка программу для их создания.

В разделе «Изменение идентификатора» для создания нескольких однотипных экземпляров переменных пользователь может задать суффикс или префикс. Активация флажка «Привязывать переменные» позволяет автоматически привязывать переменные при импорте файла.

Также может быть задан префикс привязки.

Настройка служб серверов производится аналогично настройке служб клиентов.

6.9.8 Конфигурирование обмена по протоколу HART

Обмен по протоколу HART производится посредством модулей AI-416 (ведущий) и AO-408 (ведомый).

Для этого пользователь добавляет службу обмена через контекстное меню (рисунок 73).

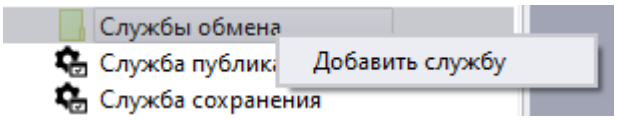

Рисунок 73 – Добавление службы обмена

Затем выбирает модуль, его позицию и номер канала (порт), по которому требуется производить обмен (рисунок 74).

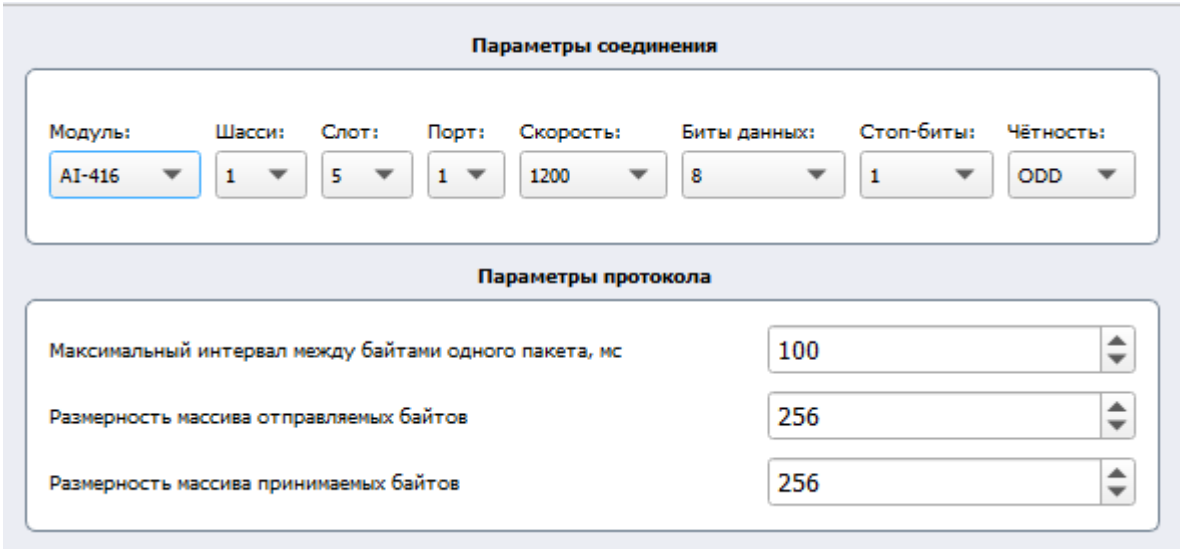

Рисунок 74 – Окно настройки элемента службы обмена по протоколу HART

При выборе модулей AI-416 или AO-408 параметры скорости, количества битов данных, стоп-битов и четность зафиксированы в соответствии с требованиями протокола HART и недоступны для изменения: скорость составляет 1200 бит/с, 8 бит данных, нечетная (ODD) четность и 1 стоп-бит (стандарт BELL 202).

После создания службы обмена в дереве компонентов проекта появится соответствующий элемент.

Для информационного взаимодействия со службой обмена по протоколу HART необходимо перенести элемент как функциональный блок в прикладную программу из библиотеки компонентов и произвести конфигурирование вызова ФБ (установку значений его входов), после чего произвести обработку значений выходов.

Формирование и обработка пакетов обмена по протоколу производится средствами прикладной программы в соответствии с требованиями стандарта протокола HART, служба обмена обеспечивает только трансляцию данных.

Пример прикладной программы расположен в примерах программ. Перечень переменных приведен в таблице 21.

Таблица 21 – Перечень переменных

Мнемоника	Тип	Описание
Входы:		
EN_SERVICE	BOOL	Разрешение работы службы: • 1 – функционирование разрешено, порт открыт; • 0 – функционирование запрещено, порт закрыт.
SEND_BUF	BYTE[X]	Массив отправляемых байтов данных (размерность массива X задается на этапе создания экземпляра службы)
SEND_CNT	UINT	Количество отправляемых байтов данных
SEND_START	BOOL	Сигнал отправки пакета: • 1 – происходит передача информационного пакета.
RECV_BYTES	UINT	Ожидаемое количество байт при приеме. Если количество принятых байт равно значению этого параметра, служба считает пакет полученным. Если параметр равен нулю, чтение из порта не производится.
BYTE_TIMEOUT	UINT	Таймаут между байтами, мс (если в течение данного интервала на порт не поступают данные, служба считает пакет полученным)
PROT_CODE	UINT	Код типа протокола передачи: • 0 – Контроль целостности не производится; • 1 – CRC8 в конце пакета, расчет по всем байтам пакета; • 2 – CRC16 в конце пакета, расчет по всем байтам пакета;

Мнемоника	Тип	Описание
		• 3 – CRC32 в конце пакета, расчет по всем байтам пакета; • 4 – HART, преамбула 8 байт 0xFF и контрольная сумма XOR по всем байтам кроме преамбулы в конце пакета.
RECV_RESET	BOOL	Сигнал сброса принятой информации: • 1 – накопленная в буфере приема информация удаляется.
STAT_RESET	BOOL	Сигнал сброса счетчиков статистики: • 1 – производится сброс.
Выходы:		
SERVICE_STATUS	USINT	Код статуса функционирования службы: • 0 – служба не запущена; • 1 – служба запущена, функционирование разрешено; • 2 – служба запущена, функционирование не разрешено; • 5 – служба запущена, ошибка открытия порта.
RECV_BUF	BYTE[X]	Массив полученных байтов данных (размерность массива X задается на этапе создания экземпляра службы)
DATA_RECVD	BOOL	Сигнал получения пакета данных: • 1 – получен целостный пакет данных (выставляется на один цикл исполнения технологической программы).
RECV_CNT	UINT	Количество полученных байтов в пакете (исключая преамбулы и контрольные суммы, используемые для контроля целостности)
RTS	BOOL	Сигнал готовности к передаче: • 1 – после опустошения выходного буфера; • 0 – после получения сигнала отправки.
STAT_SEND_PACKETS	ULINT	Количество полученных пакетов (обнуляется по сигналу STAT_RESET)
STAT_RECV_PACKETS	ULINT	Количество отправленных пакетов (обнуляется по сигналу STAT_RESET)
STAT_BAD_PACKETS	ULINT	Количество полученных поврежденных пакетов (обнуляется по сигналу STAT_RESET)

Мнемоника	Тип	Описание
STAT_SEND_BYTES	ULINT	Количество полученных байтов данных (обнуляется по сигналу STAT_RESET)
STAT_RECV_BYTES	ULINT	Количество отправленных байтов данных (обнуляется по сигналу STAT_RESET)

6.9.9 Диалог конфигурирования протокола OPC UA

Сервер OPC UA может быть добавлен в дерево проекта при нажатии ПКМ на соответствующий элемент служб и выборе команды «Добавить службу» (рисунок 75).

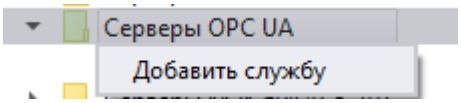

Рисунок 75 – Контекстное меню добавления службы сервера OPC UA

Окно службы «Серверы OPC UA» (рисунок 76) содержит вкладки «Параметры соединения», «Адресное пространство» и «Импорт/экспорт».

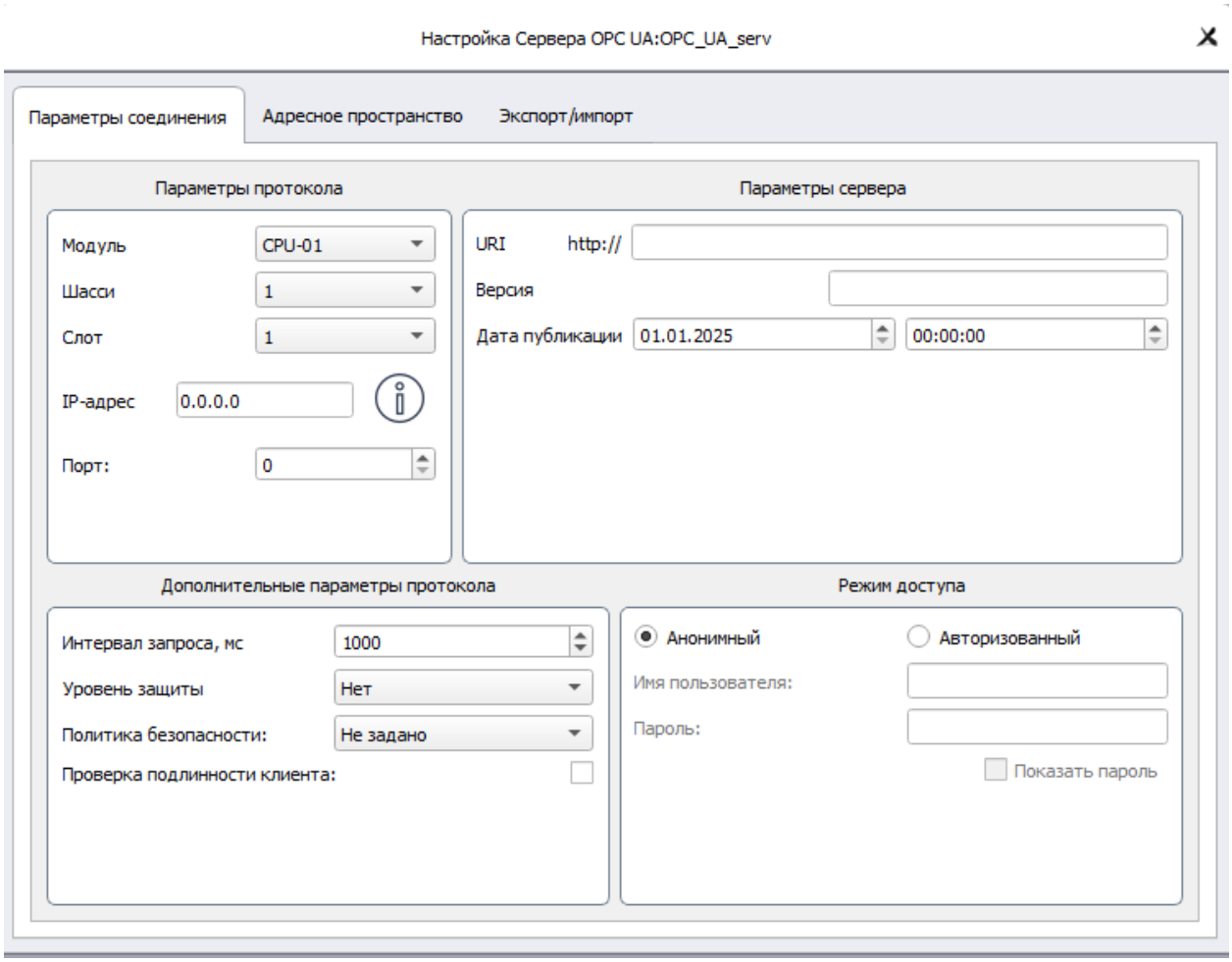

Рисунок 76 – Окно настройки элемента службы сервера OPC UA

Вкладка «Параметры соединения» включает:

- панель «Параметры протокола»:
  - выпадающий список «Модуль»;
  - выпадающий список «Шасси»;
  - выпадающий список «Слот»;
  - поле «IP-адрес» со значком ⓘ, при наведении курсора на который появляется информация о привязке сервера к сетевым интерфейсам. По умолчанию в поле введен адрес 0.0.0.0, привязывающий сервер ко всем имеющимся сетевым интерфейсам. Для привязки к конкретному сетевому интерфейсу пользователю необходимо ввести в поле его IP-адрес;
  - поле «Порт» с кнопками «▲» и «▼» позволяет пользователю вводить числа от 1 до 65535.
- панель «Дополнительные параметры протокола»:
  - поле «Интервал запроса, мс» с кнопками «▲» и «▼» позволяет пользователю вводить числа (значение по умолчанию 1000 мс);
  - выпадающий список «Уровень защиты», содержащий элементы «Нет», «Подпись» и «Подпись и шифрование»;
  - флаг «Проверка подлинности клиента», при активации которого пользователю необходимо загрузить сертификат клиента в ПЛК с помощью утилиты R-Logic.Загрузчик;
  - выпадающий список «Политика безопасности», содержащий элементы «Не задано», «Basic128RSA15», «Basic256» и «Basic256SHA256».
- панель «Режим доступа»:
  - радиокнопки «Анонимный» и «Авторизованный»;
  - поле «Имя пользователя», в котором пользователь может ввести логин для авторизации;
  - поле «Пароль», в котором пользователь может ввести пароль для авторизации.
- панель «Параметры сервера»:
  - поле «URI», в котором пользователь может вписать необходимые символы стандарта UTF-8;
  - поле «Версия», в котором пользователь может вписать необходимые символы стандарта UTF-8;
  - поле «Дата публикации», в котором пользователь может вписать необходимую дату формата «ДД.ММ.ГГГГ» и «ЧЧ.ММ.СС».

Вкладка «Адресное пространство» (рисунок 77) включает:

- область «Информационная модель», в которой представлено дерево, состоящее из узлов типа объект и переменная;
- кнопки, расположенные под областью «Информационная модель»:
  - новая папка;
  - новый тег;
  - вверх;
  - вниз.

- панель «Атрибуты» со следующими параметрами выбранного в области «Информационная модель» узла:
  - поле «Идентификатор узла (NodeID)», заполненное уникальным числовым значением от 0 до 4294967295. Для внесения изменений пользователь может нажать кнопку «…», открывающую окно редактирования, которое содержит поле «Введите новый идентификатор» и кнопки «Применить» и «Отменить». При выборе уже существующего номера идентификатора появляется надпись «Ошибка! Такой код уже занят!»;
  - поле «Класс узла» (неизменяемое);
  - поле «Определение типа» (неизменяемое);
  - поле «Поисковое имя (BrowseName)», которое пользователь может заполнить символами стандарта UTF-8 с помощью кнопки «…», при нажатии на которую открывается окно редактирования, содержащее поле «Введите новое поисковое имя» и кнопки «Применить» и «Отменить». Если поле не заполнено, оно выделяется лососевым цветом. При переключении на другой узел появляется сообщение о необходимости заполнения поля или, если имя не прошло контроль уникальности, изменения значения поля;
  - поле «Отображаемое имя (DisplayName)»;
  - поле «Описание»;
  - выпадающий список «Право доступа», содержащий элементы «Читать» и «Читать/записать»;
  - выпадающий список «Тип данных», отображающийся только для класса узла типа «переменная» и содержащий типы данных OPC UA с обозначением соответствующих типов данных в среде R-Logic.Designer;
  - флаг «Массив», отображающийся только для узла типа «переменная», при активации преобразует узел в массив;
  - поле «Размерность массива» (отображается только при активном флаге «Массив»);
  - флаг «Значение по умолчанию», при активации которого отображаются поле для ввода значений и кнопки, отличающиеся в зависимости от выбранного типа данных и активации флага «Массив».
- панель «Привязанные переменные и дополнительные параметры», содержащую:
  - поле «Привязка переменной» с кнопкой «…», при нажатии на которую открывается окно «Список переменных», позволяющее выбрать привязываемую переменную, и кнопкой «x» очистки поля привязанной переменной (только определенных для тегов, не являющихся матрицами);
  - поле «Статус качества» с кнопкой «…», при нажатии на которую открывается окно «Список переменных», позволяющее выбрать привязываемую переменную, и кнопкой «x» очистки поля привязанной переменной. Ниже поля отображается описание переменной (при наличии) (только для определенных тегов, не являющихся матрицами);
  - поле «Временная метка источника» с кнопкой «…», при нажатии на которую открывается окно «Список переменных», позволяющее выбрать привязываемую переменную, и кнопкой «x» очистки поля привязанной

переменной. Ниже поля отображается описание переменной (при наличии) (только для определенных тегов, не являющихся матрицами). Без привязки при работе службы значение метки автоматически получается с RTS при изменении значения передаваемой переменной Value или ее качества.

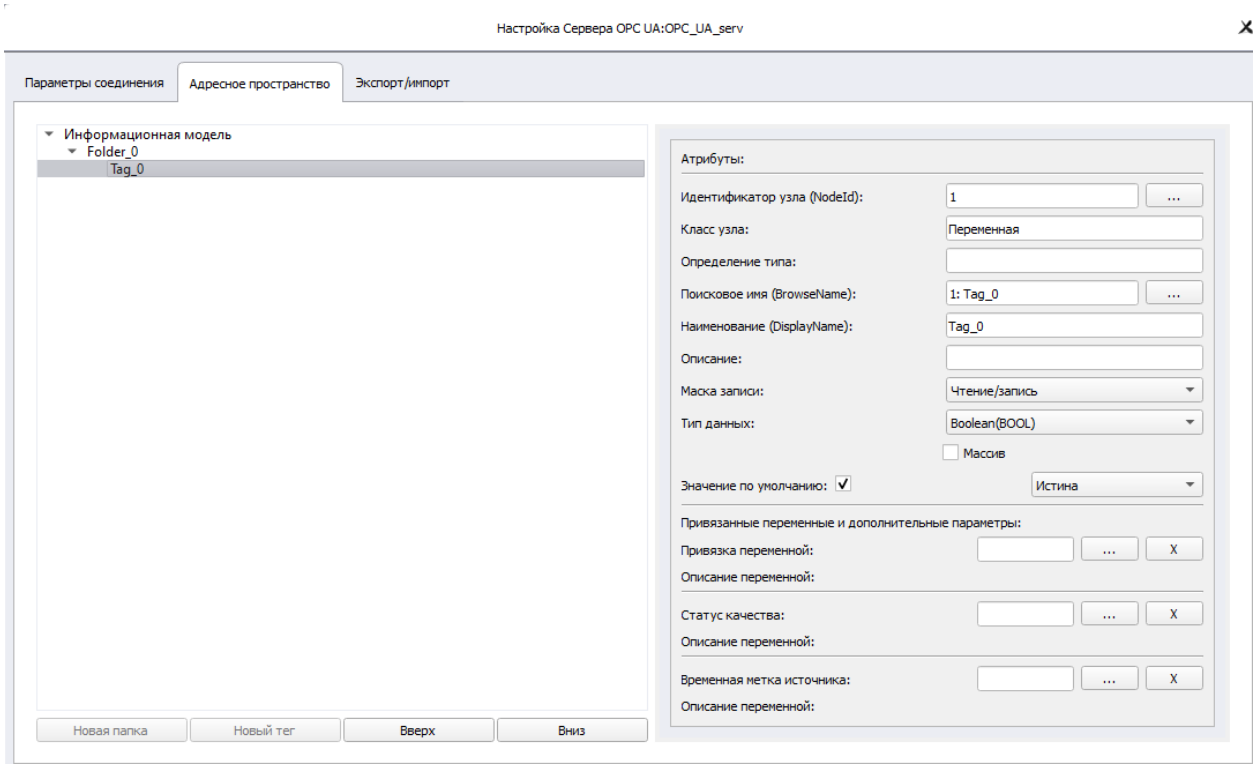

Рисунок 77 – Вкладка «Адресное пространство» окна настройки службы сервера OPC UA

Вкладка «Экспорт/импорт» (рисунок 78) включает:

- панель «Экспорт/импорт модели OPC UA», содержащую:
  - кнопку «Экспорт модели в xml», при нажатии на которую открывается окно выбора пути сохранения экспортируемого файла;
  - кнопку «Импорт структуры модели из внешнего xml», при нажатии на которую открывается окно выбора расположения импортируемого файла. При выборе файла для импорта появляется предупреждение: «При импорте файла текущая информационная модель будет утеряна. Будут импортированы только те узлы, типы которых поддерживаются средой разработки. Вы уверены, что хотите импортировать выбранный файл?» и кнопки «Принять» и «Отменить».
- панель «Экспорт/импорт данных о привязках переменных модели OPC UA», содержащую:
  - кнопку «Экспорт данных о привязках переменных к узлам модели в xml», при нажатии на которую открывается окно выбора пути сохранения экспортируемого файла;
  - кнопку «Импорт данных о привязках из внешнего xml», при нажатии на которую открывается окно выбора расположения импортируемого файла.

При этом производится проверка наличия тегов и переменных, контроль совпадения тегов и привязываемых к ним переменных, а также проверка совпадения размерности (для массивов). В результате выводятся соответствующие сообщения, общее число проваленных проверок соответствия и формируется файл несоответствий. После завершения проверок открывается окно подтверждения импорта.

Рисунок 78 – Вкладка ««Экспорт/импорт»» окна настройки службы сервера OPC UA

7 Загрузка проекта в контроллер

Загрузка проекта производится нажатием на кнопку панели инструментов «Выгрузить на ПЛК», либо командой «Выгрузить на ПЛК» во вкладке «Проект». Для загрузки проекта на ПЛК также может быть использована утилита R-Logic.Загрузчик (см. п. 9).

Для этого пользователь может нажать кнопку «Собрать проект» на панели инструментов (рисунок 79), либо выбрать соответствующий пункт меню «Проект» (рисунок 80), после чего происходит компиляция, статус которой отображается во вкладке «Сообщения сборки» окна «Сообщения» (рисунок 81).

Пункт «Запустить» меню «Проект» (горячая клавиша F9) запускает последовательно процессы сборки, выгрузки и отладки.

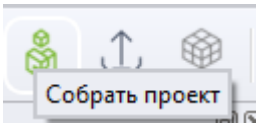

Рисунок 79 – Кнопка «Собрать проект» на панели инструментов

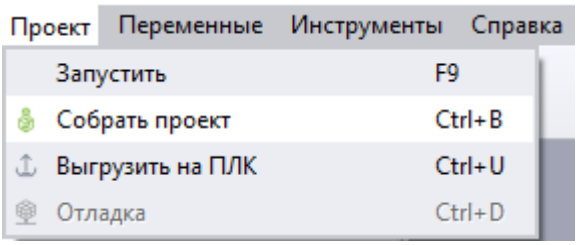

Рисунок 80 – Меню «Проект»

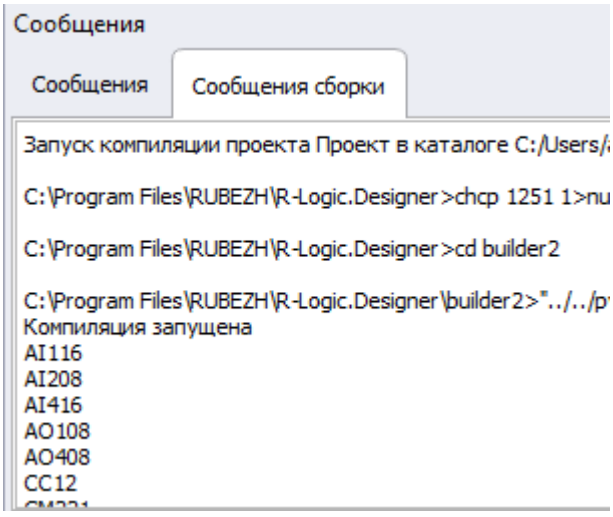

Рисунок 81 – Статус компиляции проекта во вкладке «Сообщения сборки»

После успешной компиляции становится доступной кнопка «Выгрузить на ПЛК» (рисунок 82) и соответствующий пункт меню «Проект» (рисунок 80).

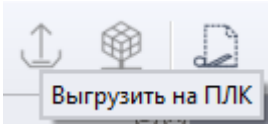

Рисунок 82 – Кнопка «Выгрузить на ПЛК»

После нажатия на кнопку или команду выгрузки проекта появляется диалоговое окно «Подключение к контроллеру» (рисунок 83). В выпадающем списке пользователь выбирает сетевой интерфейс, а также авторизуется в полях «Логин» и «Пароль». Активация флажка «Незащищенное подключение» отменяет проверку соответствия IP-адреса сертификату. После нажатия кнопки «Подключиться» запускается процесс удаленного подключения к ПЛК.

ВАЖНО! В открывшемся при выгрузке окне «Подключение к контроллеру» пользователь может указать один из доступных сетевых интерфейсов, настройки которого задаются в конфигураторе ЦПУ во вкладке «Настройки сети» (см. п. 6.4.1).

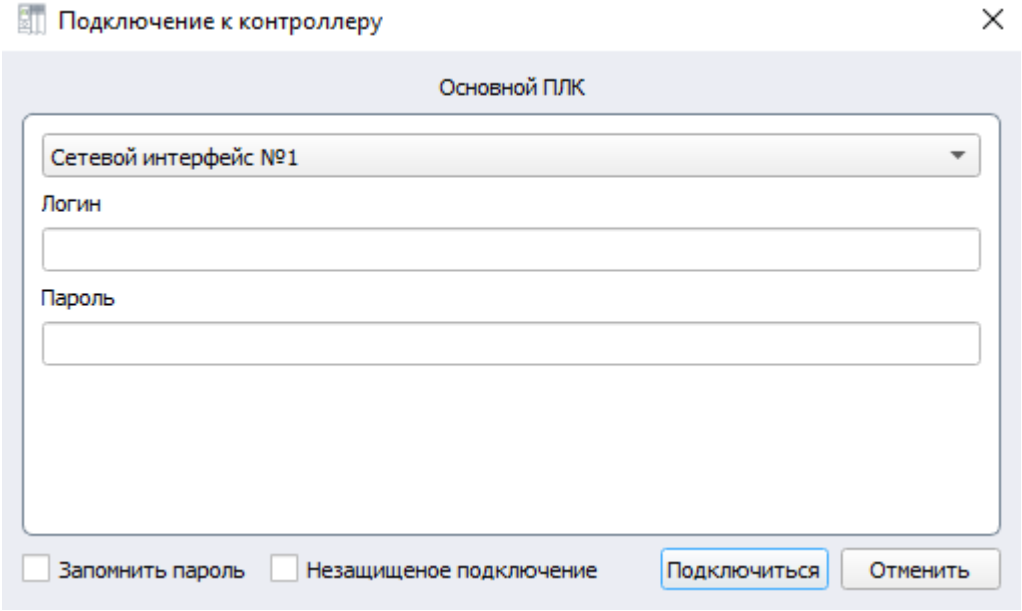

Рисунок 83 – Окно подключения к контроллеру

При резервировании окно «Подключение к контроллеру» (рисунок 84) содержит панели «Основной ПЛК» и «Резервный ПЛК», на которых пользователь выбирает сетевой интерфейс и авторизуется отдельно для каждого из ПЛК.

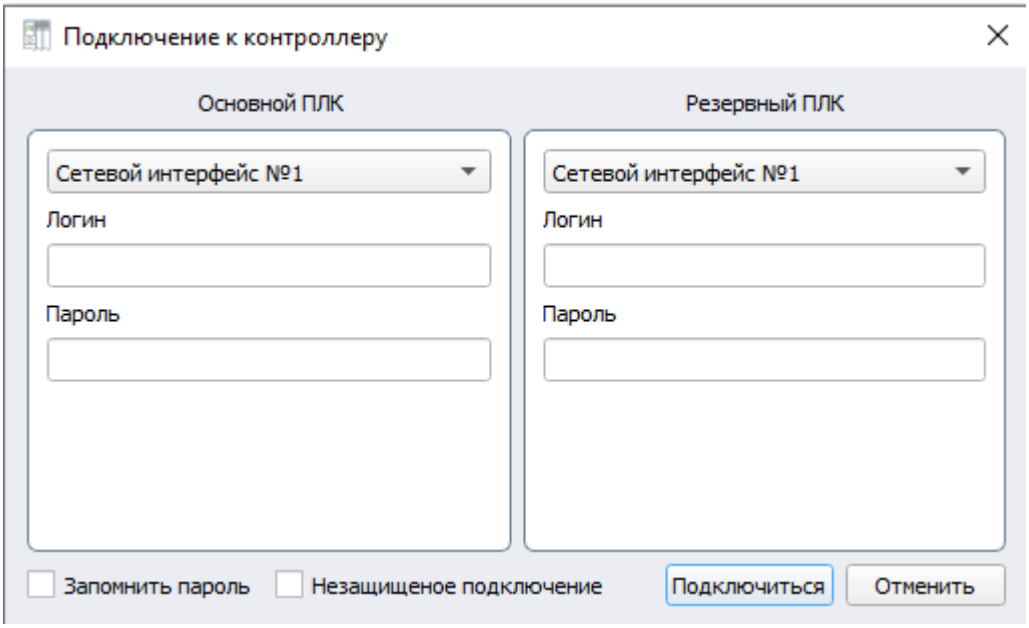

Рисунок 84 – Окно «Подключение к контроллеру» (при включенном резервировании)

После нажатия кнопки «Подключиться» появляется предупреждение.

8 Средства мониторинга и отладки

Отладка осуществляется после успешной сборки и загрузки проекта. При подключении сверяется контрольная сумма проекта, формируемая при компиляции. При обнаружении несовпадения контрольной суммы появляется всплывающее окно (рисунок 85), перехода в режим отладки не происходит.

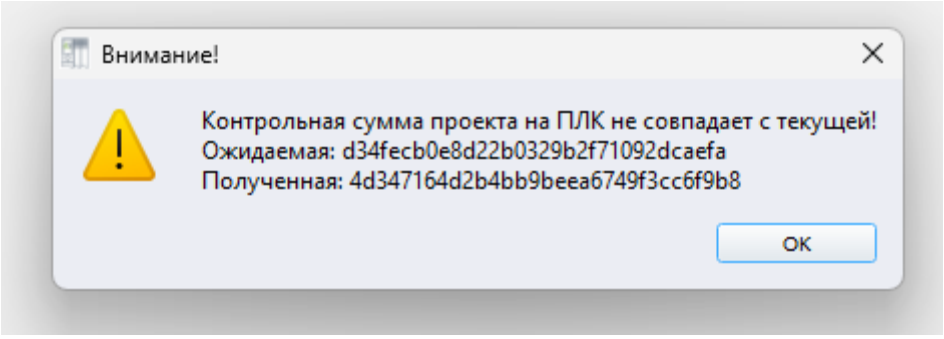

Рисунок 85 – Предупреждение о несовпадении контрольной суммы

После нажатия кнопки «Отладка» (рисунок 86) или выбора соответствующего пункта меню «Проект» (рисунок 80) открывается диалоговое окно «Подключение к контроллеру».

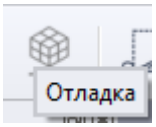

Рисунок 86 – Кнопка «Отладка»

При успешном подключении к контроллеру автоматически открывается вкладка «Онлайн» в окне «Проект», разворачивается окно «Отладка» и скрывается окно «Компоненты».

Мониторинг осуществляется после перевода программы в режим «Онлайн» («Отладка») (рисунок 87).

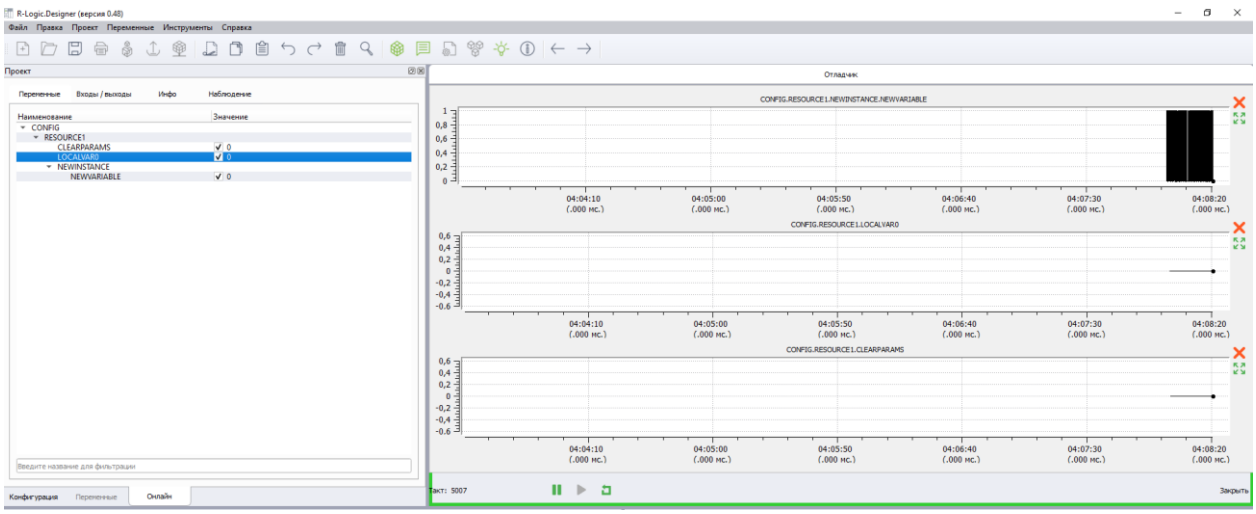

Рисунок 87 – Окно отладки

В режиме «Онлайн» пользователь в реальном времени может наблюдать за процессом работы контроллера и по необходимости вносить изменения.

В нижней части окна расположено поле для фильтрации отображаемых переменных (рисунок 88).

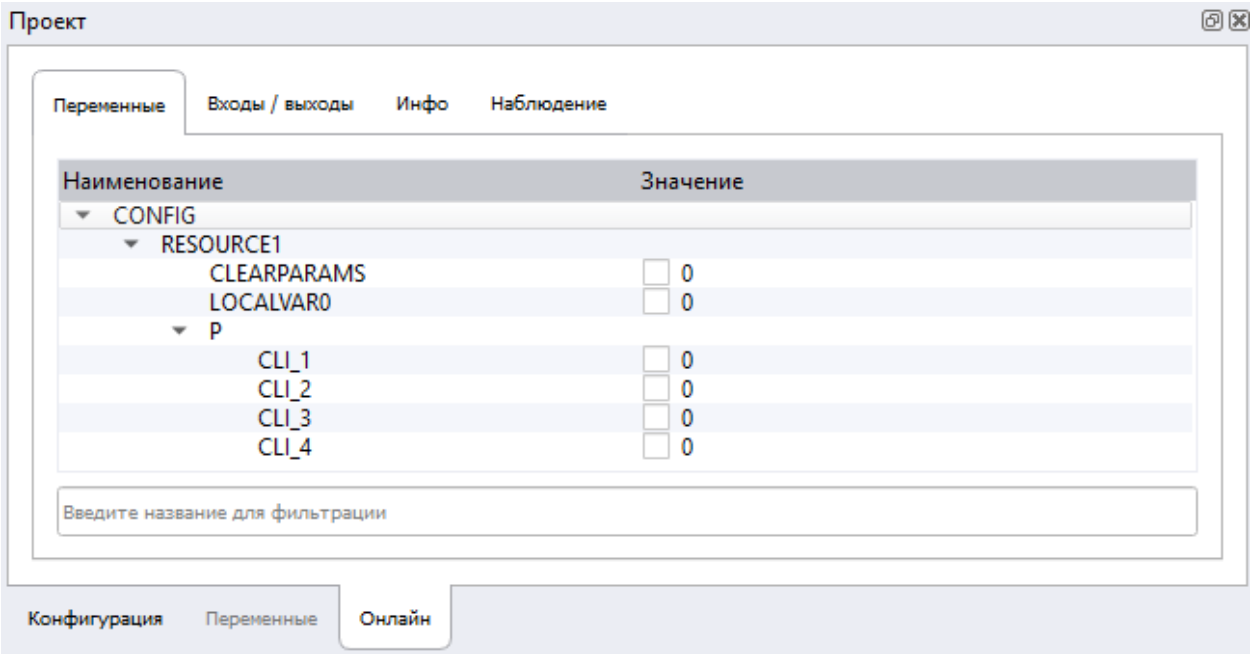

Рисунок 88 – Окно настройки отображения переменных в режиме «Онлайн»

Во вкладке «Переменные» находится дерево переменных. Напротив каждой из них отображается значение, которое может быть изменено пользователем, а также флажок, отвечающий за отображение на графике в окне «Отладка».

Для отображения переменных в режиме отладки необходимо поставить флажок «Внешний доступ» в свойствах переменной (рисунок 89), а затем осуществить пересборку и загрузку проекта на ПЛК.

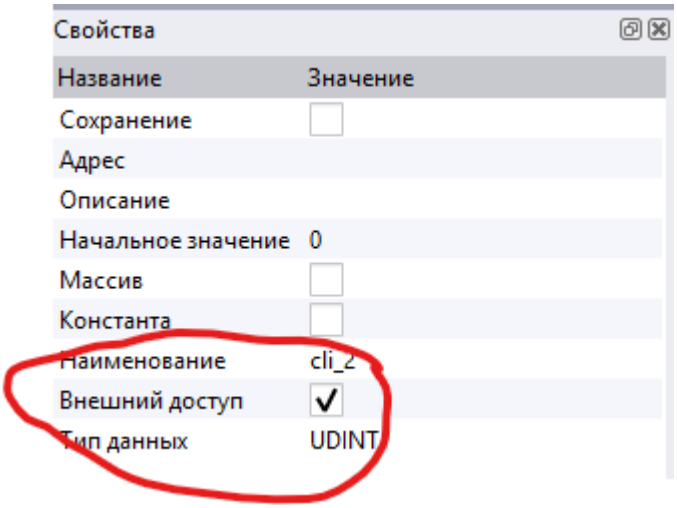

Рисунок 89 – Настройка отображения переменной в окне «Отладка»

При нажатии ПКМ на тег появляется команда «Наблюдать», при ее активации переменная отображается во вкладке «Наблюдение». Удаление тега из отслеживания производится через контекстное меню во вкладке «Наблюдение».

Во вкладке «Входы/выходы» (рисунок 90) отображаются имена установленных в контроллер модулей, в скобках рядом с именем модуля отображаются номер слота и номер шасси, в которые установлен модуль. Ошибки (статусы) каналов приведены в таблице 22, ошибки модулей приведены в таблице 23.

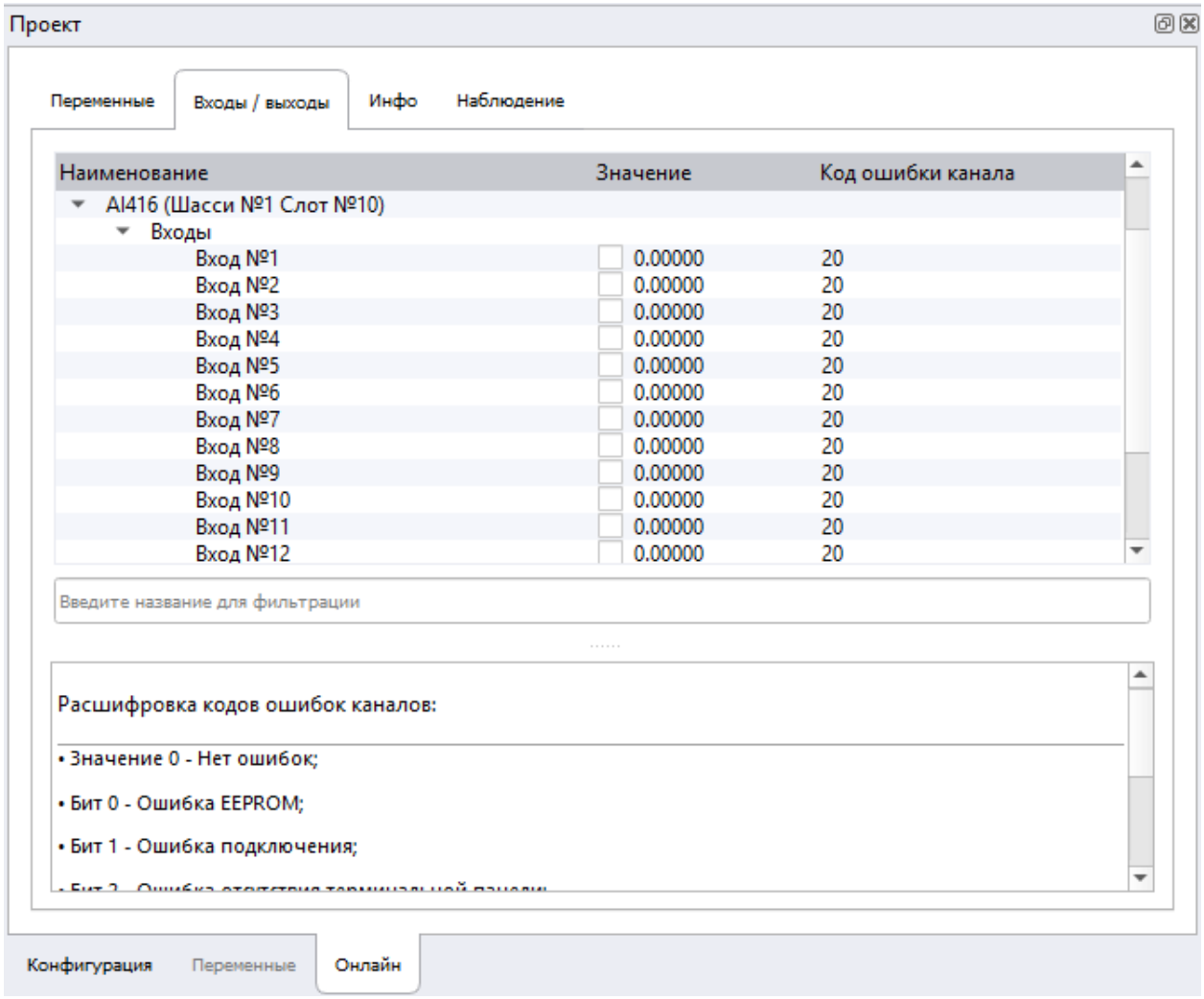

Рисунок 90 – Вкладка «Входы/выходы» (режим «Онлайн»)

Таблица 22 – Ошибки (статусы) каналов

Бит	Ошибка (статус)	Описание ошибки (статуса)
0	ERROR_CHIP_STATUS	Ошибка состояния АЦП/ЦАП
1	ERROR_VALUE_OUT_OF_RANGE	Рассчитанное значение вне диапазона
2	ERROR_SETTING_VALUE	Ошибка установки значений
3	ERROR_UNIQUE	Ошибка связи по каналу HART
4	ERROR_MODULE	Ошибка модуля (одна или несколько ошибок, перечисленных в таблице 23)

Таблица 23 – Ошибки модулей

Бит	Ошибка	Описание ошибки
0	ERROR_EEPROM	Ошибка EEPROM
1	ERROR_CONNECTION	Ошибка соединения
2	ERROR_CONTROL_CABLE	Отсутствие терминальной панели
3	ERROR_CONFIGURATION	Ошибка конфигурации

При нажатии на имя модуля раскрывается список с соответствующими характеристиками модуля, типом и количеством каналов.

ВАЖНО! При наличии привязки к выбранному входу/переменной в ППО установленное вручную значение будет заменено на значение, предоставляемое прикладной программой.

Напротив каждого канала находится флажок, отвечающий за отображение графика изменения переменной в окне «Отладка», и значение на канале, которое может быть задано пользователем.

Окно «Отладка» отражает зависимости изменения значений элементов с возможностью закрыть, свернуть или развернуть соответствующие графики (рисунок 91).

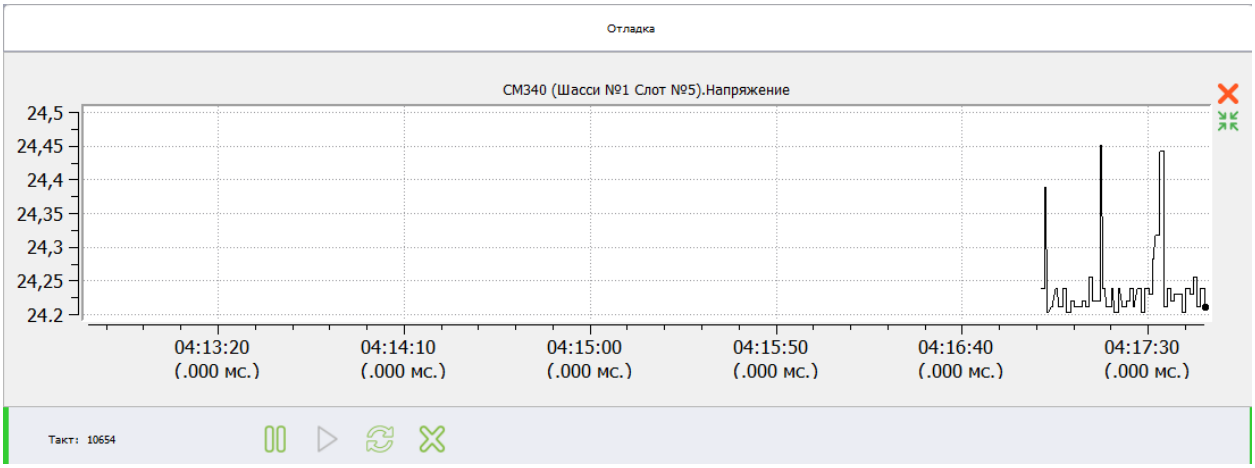

Рисунок 91 – График изменения напряжения

В нижней части окна находятся кнопки приостановки, остановки, запуска, отключения и очистки полей графиков. Также в окне «Отладка» находится счетчик тактов, а в правом верхнем углу расположена кнопка «Закрыть».

По горизонтальной оси графика отображается временная шкала, по вертикальной – числовая.

Вкладка «Инфо» (рисунок 92) содержит раскрывающийся список шасси с подключенными к ним модулями. При выборе модуля раскрывается список параметров с информацией по модулю. Напротив каждого параметра находится флажок, отвечающий за отображение параметра на графике в окне «Отладка», а также поле со значением параметра.

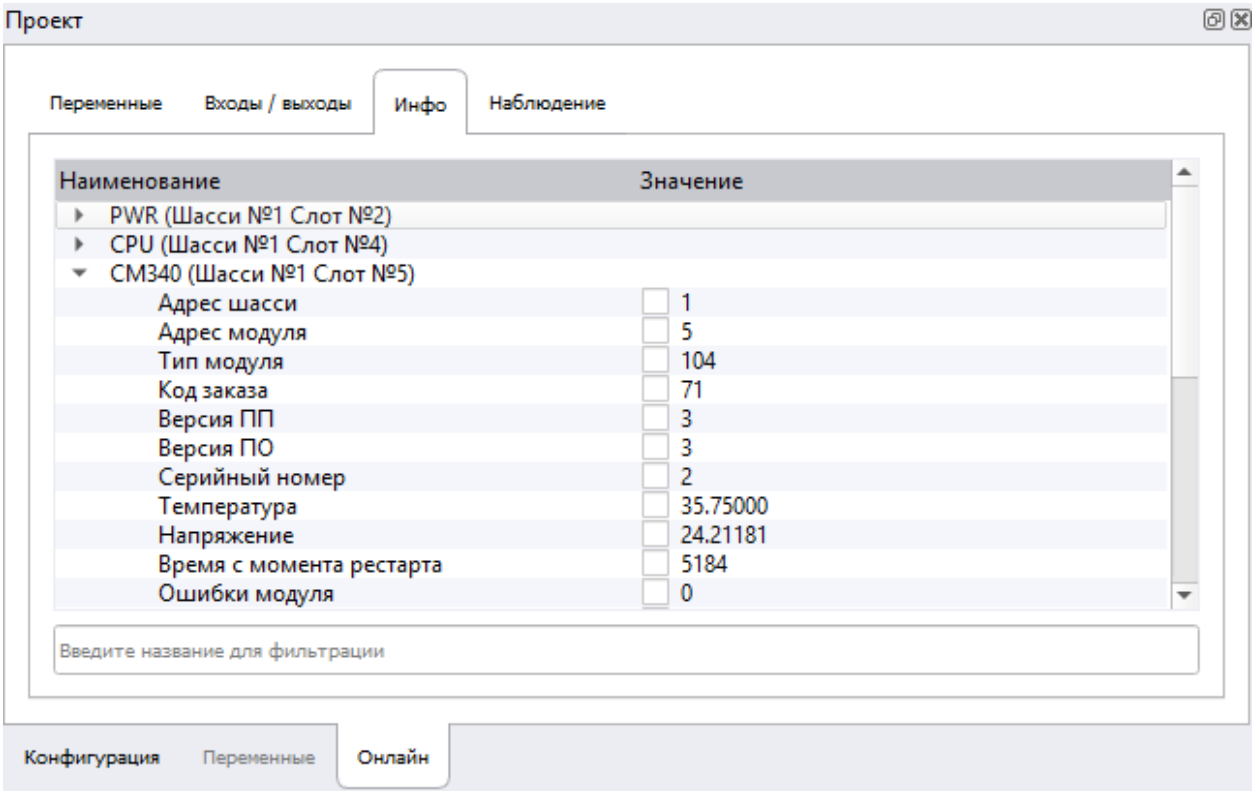

Рисунок 92 – Вкладка «Инфо» (режим «Онлайн»)

Параметры модулей перечислены в таблице 8.

Вкладка «Наблюдение» (рисунок 93) содержит таблицу, в которой отображаются переменные, входы и выходы, для которых активирована функция «Наблюдать», со столбцами:

- «Наименование» – наименования переменных, входов и выходов;
- «Размещение» – программы или модули, к которым принадлежат переменные, входы или выходы;
- «Значение» – значения переменных, входов и выходов.

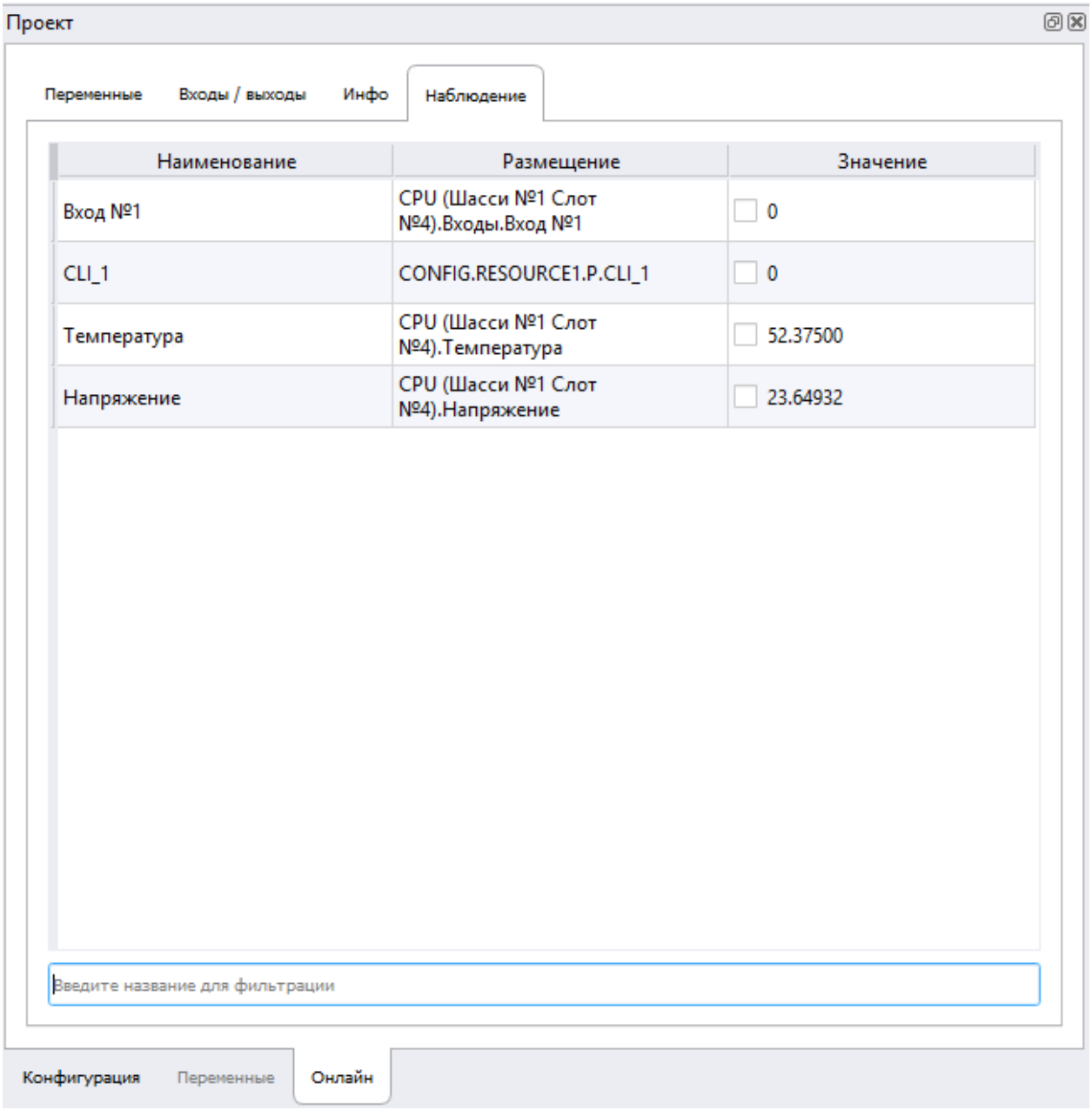

Рисунок 93 – Вкладка «Наблюдение» (режим «Онлайн»)

Для добавления отслеживаемых параметров необходимо нажать ПКМ на переменную, вход, выход или диагностическую информацию и выбрать на появившуюся команду «Наблюдать» (рисунок 94).

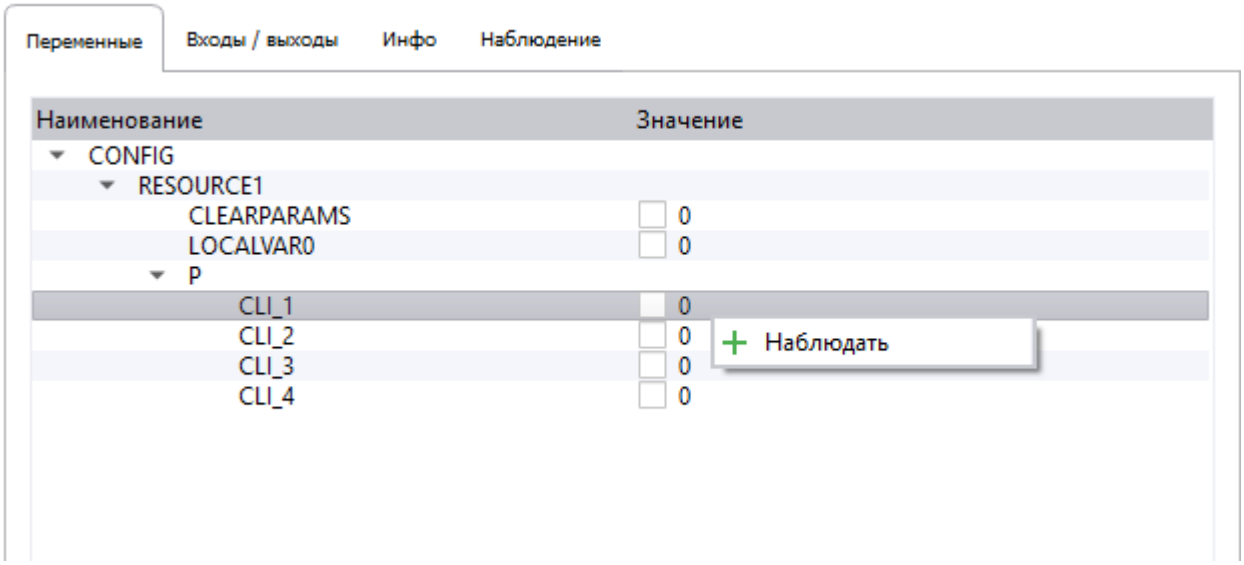

Рисунок 94 – Выбор параметров для отслеживания

9 Утилита R-Logic.Загрузчик

Загрузчик осуществляет копирование пакета установочного прикладного ПО на контроллер, а также развертывание пакета прикладного ПО на ПЛК согласно конфигурационному файлу.

Для доставки данных на контроллер загрузчик использует протокол SCP (Secure CoPy) – утилиту для безопасного копирования данных по протоколу SSH.

При запуске утилиты R-Logic.Загрузчик пользователю доступны четыре функции:

- загрузить проект;
- восстановить проект из ПЛК;
- обновление системного ПО;
- установка MQTT TLS сертификатов.

Для загрузки проекта без резервирования пользователь авторизуется – вводит логин и пароль, выбирает расположение загружаемого файла нажатием на кнопку «…», вводит IP-адрес контроллера и нажимает кнопку «Загрузить».

При включенном резервировании дополнительно пользователь задает базовые роли (ведущий и ведомый) для ПЛК с соответствующим IP-адресом (рисунок 95) и загружает общий файл проекта отдельно на каждый из контроллеров.

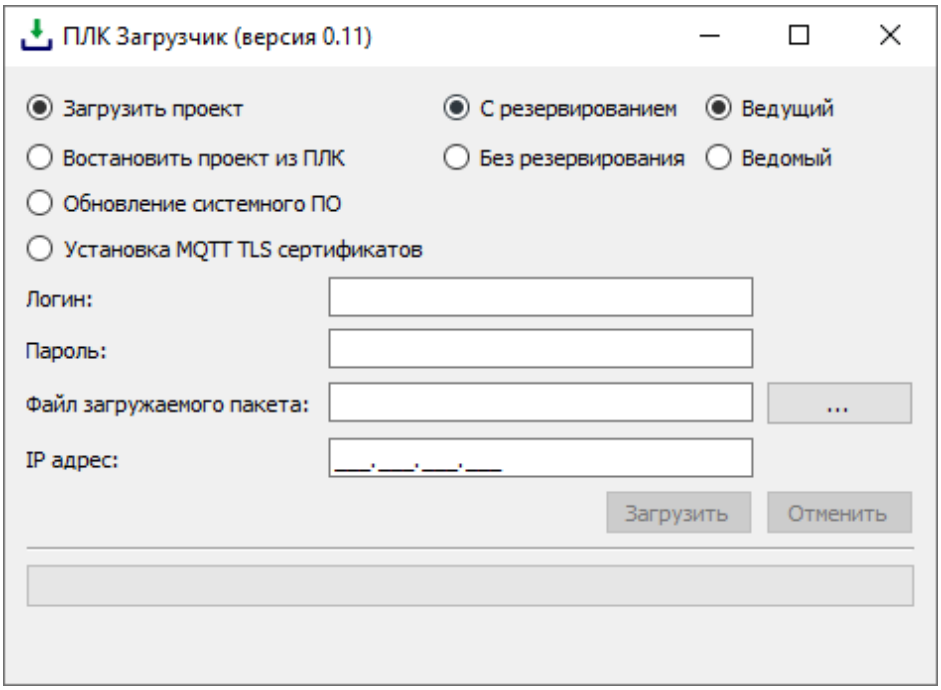

Рисунок 95 – Загрузка проекта с резервированием

Восстановление проекта из ПЛК происходит после авторизации, задания IP-адрес контроллера, пути для сохранения выгружаемого проекта (рисунок 96) и нажатия кнопки «Сохранить».

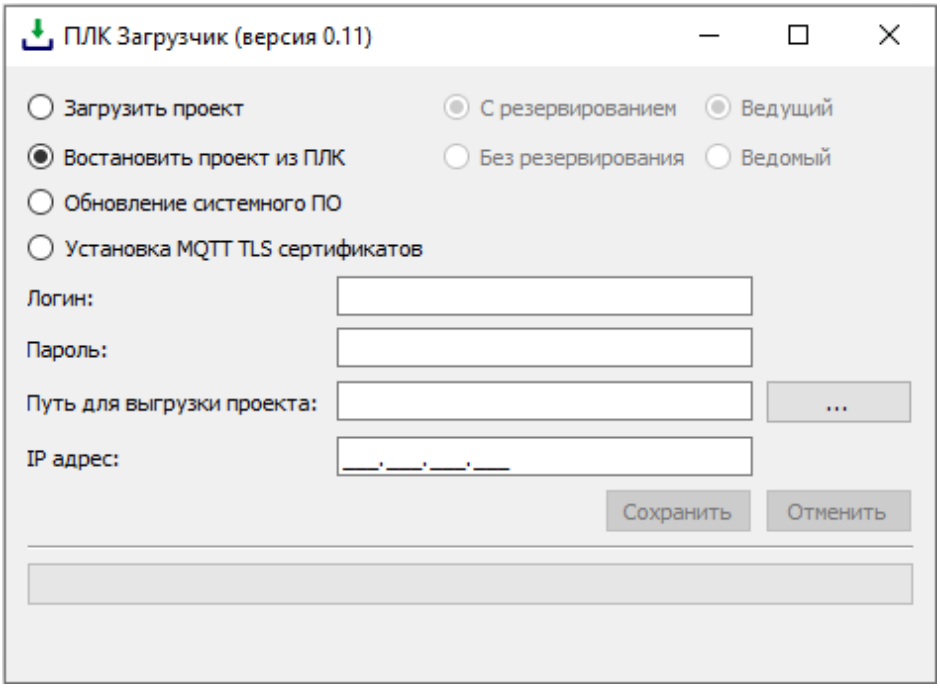

Рисунок 96 – Восстановление проекта из ПЛК

Аналогично после авторизации, задания IP-адреса контроллера и выбора пути загружаемого файла нажатием кнопки «Выгрузить» производится обновление системного ПО (рисунок 97).

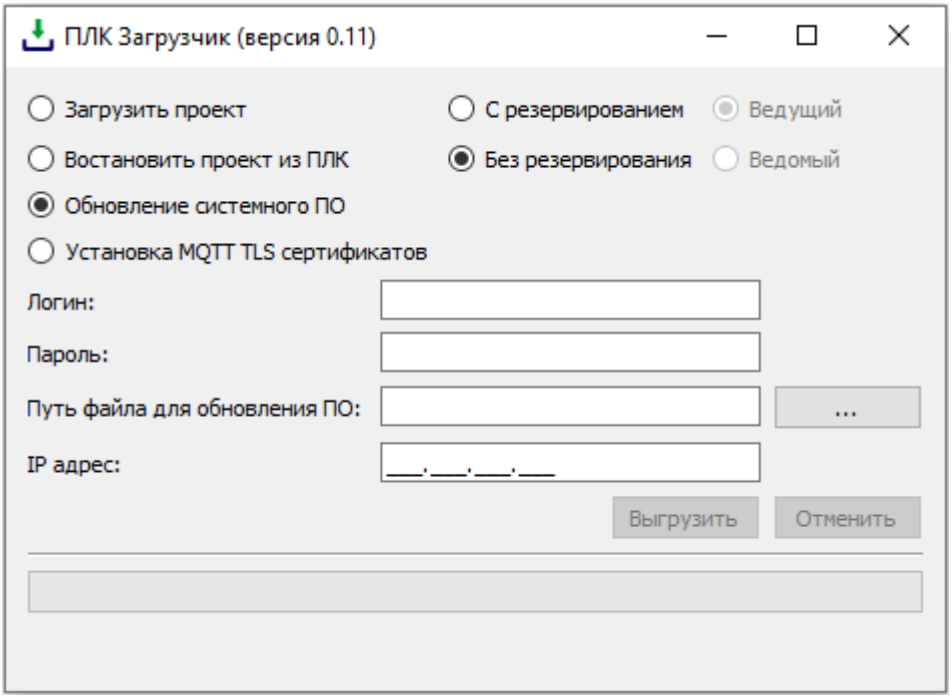

Рисунок 97 – Обновление системного ПО

MQTT TLS сертификаты необходимы для обеспечения работоспособности системы отладки. При их установке (рисунок 98) пользователь авторизуется, задает IP-адрес

контроллера, выбирает с помощью кнопки «…» и загружает (кнопка «Выгрузить») три файла сертификатов:

- публичный ключ корневого сертификата;
- публичный ключ сертификата ПЛК;
- приватный ключ сертификата ПЛК.

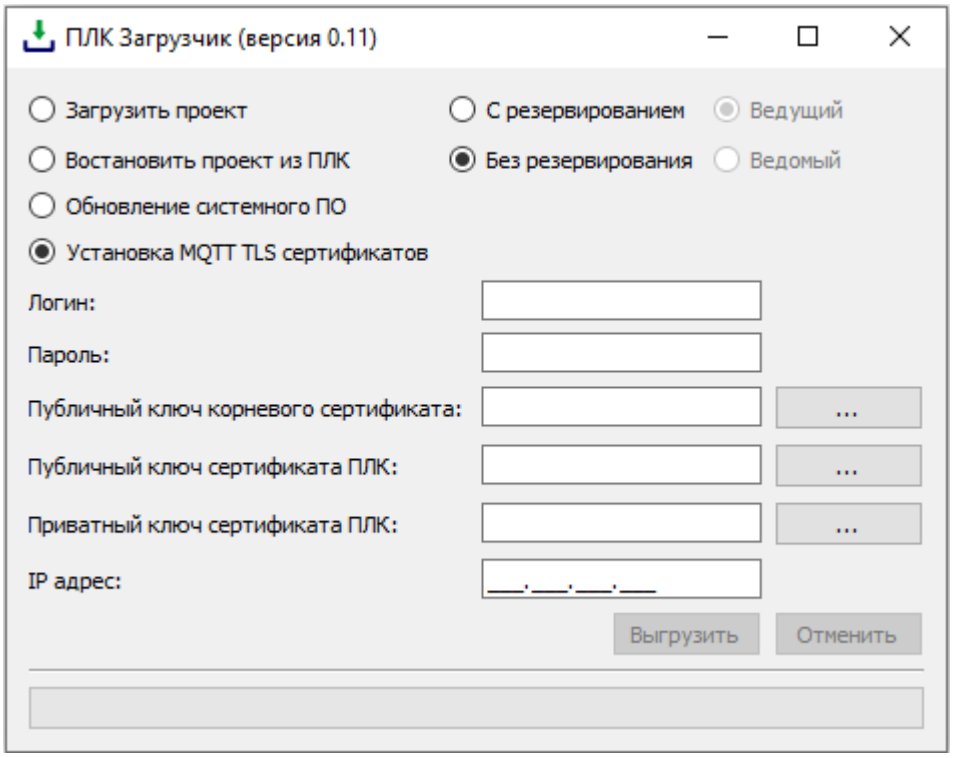

Рисунок 98 – Установка MQTT TLS сертификатов

10 Языки стандарта МЭК 61131-3

10.1 Типы данных

В таблице 24 представлены элементарные типы данных стандарта МЭК 61131-3.

Таблица 24 – Элементарные типы данных МЭК 61131-3

Номер	Описание	Ключевое слово	Диапазон значений/начальное значение	Длина (бит)
1	Логический	BOOL	0, FALSE	1 [1)]
2	Короткое целое	SINT	от -128 до 127	8
3	Целое	INT	от -32768 до 32767	16
4	Двойное целое	DINT	от -2147483648 до 2147483647	32
5	Длинное целое	LINT	от -2^{63} до $2^{63}-1$	64
6	Короткое целое без знака	USINT	от 0 до 255	8
7	Целое без знака	UINT	от 0 до 65535	16
8	Двойное целое без знака	UDINT	от 0 до 4294967295	32
9	Двойное целое без знака	ULINT	от 0 до $2^{64}-1$	64
10	Действительные числа	REAL	0.0	32
11	Действительные числа двойной точности	LREAL	0.0	64
12	Битовая строка длины 8	BYTE	от 0 до 255	8
13	Битовая строка длины 16	WORD	от 0 до 65535	16
14	Битовая строка длины 32	DWORD	от 0 до 4294967295	32
15	Битовая строка длины 64	LWORD	от 0 до $2^{64}-1$	64
[1)] Возможные значения переменных этого типа данных: 0 и 1, соответствующие ключевым словам FALSE и TRUE соответственно.				

В таблице 25 представлены типы данных времени.

Таблица 25 – Типы данных времени

Тип данных	Нижний предел	Верхний предел	Объем памяти	Разрешение
TIME	T#0d0h0m0s0ms	T#49d17h2m47s295ms	64 бит	Миллисекунды

Тип данных	Нижний предел	Верхний предел	Объем памяти	Разрешение
TOD	TOD#0:0:0.000 00:00:00.000	TOD#23:59:59.999 23:59:59.999	64 бит	Миллисекунды
DATE	D#1970-1-1 01/01/70	DATE#2106-2-7 07 февраля 2106 г.	64 бит	День
DT	DT#1979-1-1-00:00:00 01/01/1970 00:00:00	DT#2106-2-7-6:28:15 07/02/2106 06:28:15	64 бит	Миллисекунды
Примечание – Начальным значением типов DT, TOD, DATE по умолчанию назначается момент времени и даты создания переменной.				

10.2 Преобразование данных

Ниже, в таблице 26, представлены функции преобразования, обеспечивающие упаковку и распаковку данных.

Таблица 26 – Функции преобразования основных типов данных

Тип	Наименование	Входные	Выходные	Описание
FB	WORD_TO_2BYTES	IN : WORD	HIBYTE, LOWBYTE : BYTE	Разделяет WORD на два байта
FB	INT_TO_2BYTES	IN : INT	HIBYTE, LOWBYTE : BYTE	Разделяет INT на два байта
FB	UINT_TO_2BYTES	IN : UINT	HIBYTE, LOWBYTE : BYTE	Разделяет UINT на два байта
FB	DWORD_TO_4BYTES	IN : DWORD	BYTE_1, BYTE_2, BYTE_3, BYTE_4 : BYTE	Разделяет DWORD на четыре байта
FB	DINT_TO_4BYTES	IN : DINT	BYTE_1, BYTE_2, BYTE_3, BYTE_4 : BYTE	Разделяет DINT на четыре байта
FB	UDINT_TO_4BYTES	IN : UDINT	BYTE_1, BYTE_2,	Разделяет UDINT на четыре байта

Тип	Наименование	Входные	Выходные	Описание
			BYTE_3, BYTE_4 : BYTE	
FB	REAL_TO_4BYTES	IN : REAL	BYTE_1, BYTE_2, BYTE_3, BYTE_4 : BYTE	Разделяет REAL на четыре байта
FB	BDWORD_TO_4BYTES	IN : DWORD	BYTE_1, BYTE_2, BYTE_3, BYTE_4 : BYTE	Разделяет BDWORD на четыре байта
FB	BDINT_TO_4BYTES	IN : DINT	BYTE_1, BYTE_2, BYTE_3, BYTE_4 : BYTE	Разделяет BDINT на четыре байта
FB	BUDINT_TO_4BYTES	IN : UDINT	BYTE_1, BYTE_2, BYTE_3, BYTE_4 : BYTE	Разделяет BUDINT на четыре байта
FB	BREAL_TO_4BYTES	IN : REAL	BYTE_1, BYTE_2, BYTE_3, BYTE_4 : BYTE	Разделяет BREAL на четыре байта
FC	BO_12_21_WORD	IN : WORD	WORD	Выполняет перестановку байт
FC	BO_12_21_INT	IN : INT	INT	Выполняет перестановку байт
FC	BO_12_21_UINT	IN : UINT	UINT	Выполняет перестановку байт
FC	BO_1234_3412_DWORD	IN : DWORD	DWORD	Выполняет перестановку байт
FC	BO_1234_3412_DINT	IN : DINT	DINT	Выполняет перестановку байт

Тип	Наименование	Входные	Выходные	Описание
FC	BO_1234_3412_UDINT	IN : UDINT	UDINT	Выполняет перестановку байт
FC	BO_1234_3412_REAL	IN : REAL	REAL	Выполняет перестановку байт
FC	BO_1234_4321_DWORD	IN : DWORD	DWORD	Выполняет перестановку байт
FC	BO_1234_4321_DINT	IN : DINT	DINT	Выполняет перестановку байт
FC	BO_1234_4321_UDINT	IN : UDINT	UDINT	Выполняет перестановку байт
FC	BO_1234_4321_REAL	IN : REAL	REAL	Выполняет перестановку байт
FC	BO_1234_2143_DWORD	IN : DWORD	DWORD	Выполняет перестановку байт
FC	BO_1234_2143_DINT	IN : DINT	DINT	Выполняет перестановку байт
FC	BO_1234_2143_UDINT	IN : UDINT	UDINT	Выполняет перестановку байт
FC	BO_1234_2143_REAL	IN : REAL	REAL	Выполняет перестановку байт
FC	BYTES_TO_WORD	BYTE_1:BYTE; BYTE_2:BYTE	WORD	Выполняет упаковку байтов
FC	BYTES_TO_INT	BYTE_1:BYTE; BYTE_2:BYTE	INT	Выполняет упаковку байтов

Тип	Наименование	Входные	Выходные	Описание
FC	BYTES_TO_UINT	BYTE_1:BYTE; BYTE_2:BYTE	UINT	Выполняет упаковку байтов
FC	BYTES_TO_BWORD	BYTE_1:BYTE; BYTE_2:BYTE	WORD	Выполняет упаковку байтов
FC	BYTES_TO_BINT	BYTE_1:BYTE; BYTE_2:BYTE	INT	Выполняет упаковку байтов
FC	BYTES_TO_BUINT	BYTE_1:BYTE; BYTE_2:BYTE	UINT	Выполняет упаковку байтов
FC	BYTES_TO_DWORD	BYTE_1:BYTE; BYTE_2:BYTE; BYTE_3:BYTE; BYTE_4 : BYTE	DWORD	Выполняет упаковку байтов
FC	BYTES_TO_DINT	BYTE_1:BYTE; BYTE_2:BYTE; BYTE_3:BYTE; BYTE_4 : BYTE	DINT	Выполняет упаковку байтов
FC	BYTES_TO_UDINT	BYTE_1:BYTE; BYTE_2:BYTE; BYTE_3:BYTE; BYTE_4 : BYTE	UDINT	Выполняет упаковку байтов
FC	BYTES_TO_REAL	BYTE_1:BYTE; BYTE_2:BYTE; BYTE_3:BYTE; BYTE_4 : BYTE	REAL	Выполняет упаковку байтов
FC	BYTES_TO_BDWORD	BYTE_1:BYTE; BYTE_2:BYTE; BYTE_3:BYTE; BYTE_4 : BYTE	DWORD	Выполняет упаковку байтов
FC	BYTES_TO_BDINT	BYTE_1:BYTE; BYTE_2:BYTE; BYTE_3:BYTE; BYTE_4 : BYTE	DINT	Выполняет упаковку байтов

Тип	Наименование	Входные	Выходные	Описание
FC	BYTES_TO_BUDINT	BYTE_1:BYTE; BYTE_2:BYTE; BYTE_3:BYTE; BYTE_4 : BYTE	UDINT	Выполняет упаковку байтов
FC	BYTES_TO_BREAL	BYTE_1:BYTE; BYTE_2:BYTE; BYTE_3:BYTE; BYTE_4 : BYTE	REAL	Выполняет упаковку байтов
FB	BYTE_TO_BITS	IN : BYTE	BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7 : BOOL	Выполняет распаковку битов
FC	BITS_TO_BYTE	BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7 : BOOL	BYTE	Выполняет упаковку битов
FC	UINT_BIT_SET	IN : UINT BITNUM:INT BITVAL:BOOL	UINT	Выполняет установку указанного бита в слове
FC	WORD_BIT_SET	IN : WORD BITNUM:INT BITVAL:BOOL	WORD	Выполняет установку указанного бита в слове

Двоично-десятичный код – форма записи десятичных рациональных чисел, где каждая цифра десятичного числа записывается в виде четырёхбитного двоичного числа (тетрады). Базовым типом данных для BCD является WORD/DWORD. Наиболее распространенным BCD-кодом является код NBCD (натуральный двоично-десятичный код, который называют кодом 8421, по значениям степеней двойки), в котором каждая двоичная тетрада принимает значения от 0000_2 до 1001_2, что соответствует ряду десятичных цифр от 0 до 9. В таблице 27 приведены функции преобразования формата и типа данных BCD.

Таблица 27 – Функции преобразования данных типа BCD

Наименование функции преобразования	Входной тип	Выходной тип
UDINT_TO_BCD	UDINT	LWORD

Наименование функции преобразования	Входной тип	Выходной тип
UINT_TO_BCD	UINT	WORD
USINT_TO_BCD	USINT	WORD
BCD_TO_UDINT	WORD	UDINT
BCD_TO_UINT	WORD	UINT
BCD_TO_ULINT	WORD	ULINT
BCD_TO_USINT	WORD	USINT
ULINT_TO_BCD	ULINT	LWORD

10.3 Конверсия типов данных

Оператор <data type>_TO_<data type> используется для конверсии типов данных в другие типы данных. Особенности конверсии типов данных приведены в таблице функций приведенные в таблице объединены по родовым типам данных, где REAL_TO_ (ANY_SIGNED) – любая из функций REAL_TO_INT, REAL_TO_SINT, REAL_TO_DINT, и т. д.

ВАЖНО! При преобразовании из большего типа данных в меньший (сужающее преобразование) будет потеряна информация при превышении входного значения над границами выходного типа.

ВАЖНО! Если входное значение оператора конверсии типа находится вне диапазона типа выходных данных, результат операции не определен. Это происходит, например, если входное значение для конверсии из LREAL в DINT является отрицательным.

Таблица 28. Названия функций приведенные в таблице объединены по родовым типам данных, где REAL_TO_ (ANY_SIGNED) – любая из функций REAL_TO_INT, REAL_TO_SINT, REAL_TO_DINT, и т. д.

ВАЖНО! При преобразовании из большего типа данных в меньший (сужающее преобразование) будет потеряна информация при превышении входного значения над границами выходного типа.

ВАЖНО! Если входное значение оператора конверсии типа находится вне диапазона типа выходных данных, результат операции не определен. Это происходит, например, если входное значение для конверсии из LREAL в DINT является отрицательным.

Таблица 28 – Особенности конверсии типов данных

Функция	Особенности преобразования
REAL_TO_ (ANY_SIGNED)	Происходит математическое округление, после чего производится сужающее преобразование. При выходе значения за границы значений типа – результат не определен.

Функция	Особенности преобразования
LREAL_TO_ (ANY_SIGNED)	
REAL_TO_ (ANY_UNSIGNED) LREAL_TO_ (ANY_UNSIGNED) REAL_TO_ (ANY_BIT)	Происходит математическое округление, после чего производится сужающее преобразование. Отрицательные входные значения преобразуются в 0. При выходе значения за границы значений типа – результат не определен.
REAL_TO_ (ANY_DATE) REAL_TO_TIME	Происходит математическое округление, после чего целочисленное значение интерпретируется как миллисекунды.
REAL_TO_LREAL	Расширяющее преобразование, сохраняющее значение.
(ANY_INT)_TO_ (ANY_REAL)	Расширяющее преобразование, сохраняющее значение.
(ANY_UNSIGNED) _TO_ (ANY_SIGNED)	Расширяющее преобразование, сохраняющее значение. Сужающее преобразование – при выходе значения за границы значений типа результат не определен.
(ANY) _TO_BOOL	0 = FALSE, остальное TRUE
(ANY_DATE) _TO_ (ANY_NUM)	Значение интерпретируется как миллисекунды. Сужающее преобразование – при выходе значения за границы значений типа результат не определен.
(ANY_INT) _TO_ (ANY_BIT)	При расширяющем преобразовании – двоичная передача в самые правые байты адресата, остальные байты = 0. При выходе значения за границы значений типа назначения – результат не определен.

10.4 Пользовательские типы данных

Среда разработки R-Logic.Designer поддерживает следующие типы пользовательских данных: массивы простых типов данных (INT, REAL, WORD и др.) и структурированный тип данных.

Объявление массивов описано в п. 6.6.

Объявление структурированного типа происходит через добавление программного компонента, как показано на рисунке 99.

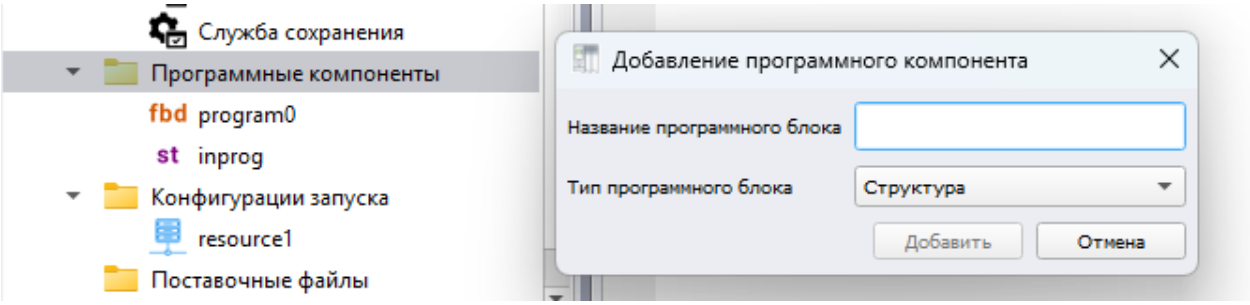

Рисунок 99 – Объявление структуры

После чего созданный элемент появляется в программных компонентах (рисунок 100).

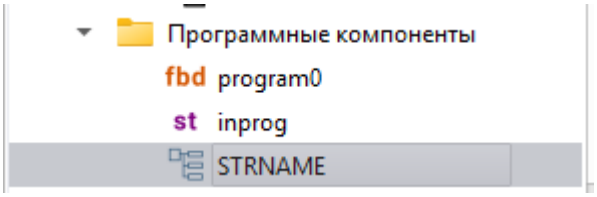

Рисунок 100 – Структура в дереве проекта

Объявление элементов структуры во вкладке «Переменные» происходит аналогично созданию переменных (рисунок 101).

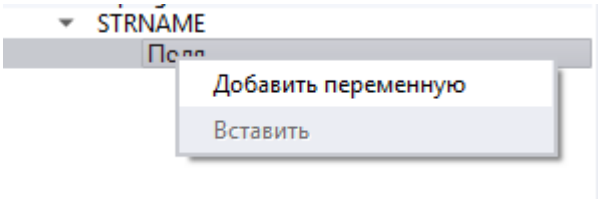

Рисунок 101 – Контекстное меню добавления переменной

Для использования структурного типа необходимо создать переменную в программном компоненте и выбрать созданный тип (рисунок 102).

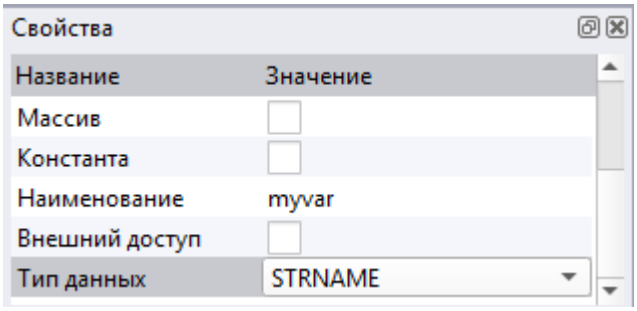

Рисунок 102 – Задание типа данных переменной

Обращение к элементам (полям) структуры производится следующим образом:

имя_экземпляра.имя_поля, где имя экземпляра – наименование созданной переменной.

10.5 Язык ST

ST (Structured Text) – это текстовый язык программирования стандарта МЭК 61131-3. ST может использоваться в программах, в теле функции или функционального блока, а также для описания действия и перехода внутри элементов SFC.

Пробелы и метки табуляции не влияют на синтаксис языка ST. Выражения в ST выполняются справа налево и выглядят следующим образом:

```
[variable] := [value];
```

Выражения состоят из операндов и операторов. Операндом является литерал, переменная, структурированная переменная, компонент структурированной переменной, обращение к функции или прямой адрес. В конце выражения ставится символ «;», обеспечивающий разделение выражений.

10.5.1 Литералы

В R-Logic.Designer отсутствуют неявные преобразования типов и применяются правила написания числовых литералов, перечисленные в таблице 29.

Таблица 29 – Числовые литералы

Номер	Описание	Пример	Объяснение
1	Целый литерал	-12, 0, 123_4, +986	
2	Действительный литерал	0.0, 0.4560, 3.14159_26	
3	Действительные литералы с экспонентой	-1.34E-12, -1.34e-12 1.0E+6, 1.0e+6 1.234E6, 1.234e6	
4	Двоичный литерал	2#1111_1111 2#1110_0000	Литерал с основанием 2 для представления: десятичного числа 255 десятичного числа 224
5	Восьмеричные литералы	8#377 8#340	Литерал с основанием 8 для представления: десятичного числа 255 десятичного числа 224
6	Шестнадцатеричный литерал	16#FF или 16#ff 16#E0 или 16#e0	Литерал с основанием 16 для представления: десятичного числа 255 десятичного числа 224
7	Логический ноль и единица	0 или 1	
8	Логические FALSE и TRUE	FALSE или TRUE	
9	Типизированный литерал	INT#-123	Представление INT десятичного значения -123

		INT#16#7FFF	Представление INT десятичного значения 32767
		WORD#16#AFF	Представление WORD шестнадцатиричного значения 0AFF
		WORD#1234	Представление WORD десятичного значения 1234=16#4D2
		UINT#16#89AF	Представление UINT шестнадцатиричного значения 89AF
		BOOL#0	
		BOOL#1	
		BOOL#FALSE	
		BOOL#TRUE	
Примечания 1 – Ключевые слова FALSE и TRUE соответствуют логически значениям 0 и 1 соответственно. 2 – Свойство 5 «Восьмеричные литералы» не рекомендуется.			

Литералы продолжительности времени описаны в таблице 30.

Таблица 30 – Литералы продолжительности времени

Номер	Описание	Пример
Сокращения продолжительности времени		
1a	d	День
1b	h	Час
1c	m	Минута
1d	s	Секунда
1e	ms	Миллисекунда
Литералы продолжительности времени без символов подчеркивания		
2a	короткий префикс	T#14ms T#-14ms T#14.7m t#12h4m34ms
2b	длинный префикс	TIME#14ms TIME#-14ms time#14.7s
Литералы продолжительности времени с символами подчеркивания		
3a	короткий префикс	t#25h_15m t#5d_14h_12m_18s_3.5ms

Номер	Описание	Пример
3b	длинный префикс	TIME#25h_15m

Литералы даты и времени суток представлены в таблице 31.

Таблица 31 – Литералы даты и времени суток

Номер	Описание	Пример
1a	Литерал даты (длинный префикс)	DATE#1984-06-25, date#2010-09-22
1b	Литерал даты (короткий префикс)	D#1984-06-25
3a	Литерал времени суток (длинный префикс)	TIME_OF_DAY#15:36:55.36
3b	Литерал времени суток (короткий префикс)	TOD#15:36:55.36
5a	Литерал даты и времени (длинный префикс)	DATE_AND_TIME#1984-06-25-15:36:55.360
5b	Литерал даты и времени (короткий префикс)	DT#2025-07-09-06:55:36.640;

10.5.2 Операторы

Ниже приведены операторы языка ST с пояснениями согласно МЭК 61131-3 (таблица 32).

Таблица 32 – Операторы языка ST

Номер	Описание	Примеры
1	Присваивание переменная := выражение;	
1.1	Переменная и выражение простого типа данных	`A:= B; CV:= CV+1; C:= SIN(X);`

1.2	Переменная и выражение типа данных, определяемого пользователем.	`A_Struct1, B_Struct1: Struct1;`
		`C_Array1,D_Array1 : ARRAY [0..10] OF INT;`
		
		`A_Struct1:= B_Struct1;`
		`C_Array1 := D_Array1;`

1.3	Обращение к элементам типов данных, определенных пользователем.	`A_Struct1.intvar:=`
		`B_Struct2.intvar_n;`
		
		`C_Array1[1] := D_Array2[3];`

Номер	Описание	Примеры
1.4	Экземпляры типа функционального блока	`A_Instance1:= B_Instance1;`
2	Вызов функции	
2.1	Вызов функции	`Var_A:=ABS(-17);`
2.2	Вызов функционального блока с именованными аргументами и использование выходной переменной функционального блока.	`CMD_TMR(IN:= bln1, PT:= T#300ms);` `A:= CMD_TMR.Q;`
2.3	Вызов ФБ с позиционным указанием аргументов.	`FB_INST.M1(17);`
3	Выход из исполняемого программного компонента.	`RETURN;`
	Выбор	

4	IF … THEN
	…
	
	ELSIF … THEN
	…
	ELSE
	…
	END_IF;
		`D:= B*B - 4.0*A*C;`
		`IF D < 0.0`
		
		`THEN NROOTS:= 0;`
		`ELSIF D = 0.0`
		`THEN`
		`NROOTS:= 1;`
		`X1:= - B/(2.0*A);`
		`ELSE`
		`NROOTS:= 2;`
		
		`X1:= (- B +`
		`SQRT(D))/(2.0*A);`
		
		`X2:= (- B -`
		`SQRT(D))/(2.0*A);`
		`END_IF;`

5	CASE … OF
	…
	ELSE …
	END_CASE
		`A:=2;`
		`CASE A OF`
		`1,5: B:=1;`
		`2: B:=2;`
		`3: B:=3;`
		`ELSE`
		`B:=10;`
		`END_CASE;  (*B=2*)`

	Итерация	
6	FOR … TO … BY … DO …	`FOR I:= 1 TO 100 BY 2 DO` `IF  I>10 THEN`

Номер	Описание	Примеры
	END_FOR	EXIT; (*Выход из итерации*) END_IF; J:=J+1; END_FOR; (* J=5, I=11 *)
7	WHILE … DO … END_WHILE	J:= 1; WHILE J <= 100 DO J:= J+2; END_WHILE;
8	REPEAT … UNTIL … END_REPEAT	J:= -1; REPEAT J:= J+2; UNTIL J = 101 END_REPEAT;
9	Выход из итерации	EXIT; (см. также в свойстве 6)
10	Разделитель выражений	;
11	Комментарий	(* ЭТО комментарий *)

Операторы выбора (IF и CASE) выбирают один (или группу) составляющих операторов для выполнения на основе указанного условия.

Операторы итерации (WHILE, REPEAT, FOR) указывают, что группа связанных операторов должна выполняться повторно. Операторы WHILE и REPEAT не должны использоваться для достижения межпроцессной синхронизации, например, как «цикл ожидания» с внешне определяемым условием завершения. Для этой цели должны использоваться элементы SFC. Ошибка возникает, если оператор WHILE или REPEAT используется в алгоритме, для которого удовлетворение условия завершения цикла или выполнение оператора EXIT не может быть гарантировано. Оператор FOR используется, когда число итераций может быть определено заранее; в противном случае используются конструкции WHILE или REPEAT. При работе с WHILE и REPEAT стоит помнить об особенности исполнения циклов прикладной программы – в них нельзя ожидать изменений «извне».

10.5.2.1 IF

Оператор IF (см. таблицу 32, п. 4) указывает, что группа операторов выполняется, только если связанное логическое выражение при вычислении принимает значение «1» (TRUE). Если условие является ложным, то или оператор не выполняется, или выполняется группа операторов, следующая за ключевым словом ELSE (или ключевым словом ELSIF, если связанное логическое условие является истинным). Завершение группы операторов выполняется ключевым словом END_IF. Символ разделения операторов «;» используется после вложенных операторов и после END_IF.

10.5.2.2 CASE

Оператор CASE включает выражение, которое вычисляет переменную простого типа данных («селектор»), и перечень групп операторов, причем каждая группа маркируется одним или большим числом литералов или поддиапазонов, в зависимости от того, что применимо. Типы данных таких маркеров должны соответствовать типу данных переменной селектора, т.е. переменная селектора должна быть сравнимой с маркерами. Это указывает на то, что выполняется первая группа операторов, один из диапазонов которых содержит вычисленное значение селектора. Если значение селектора находится вне диапазона для любого из случаев, то выполняется последовательность операторов, следующая за ключевым словом ELSE (при ее наличии в операторе CASE). В противном случае ни одна из последовательностей операторов не выполняется.

10.5.2.3 FOR

Оператор FOR указывает, что последовательность операторов выполняется повторно, до ключевого слова END_FOR, в то время как последовательность значений присваивается переменной управления циклом FOR. Переменная управления, начальное значение и конечное значение должны быть выражениями одного и того же целого типа (например, SINT, INT или DINT) и не должны изменяться ни в каком из повторяющихся операторов. Оператор FOR приращивает переменную управления вверх или вниз от начального до конечного значения в приращениях, определяемых значением выражения. Если конструкция BY пропускается, то значение приращения по умолчанию приравнивается к 1.

Тест на условие завершения выполняется в начале каждой итерации, так что последовательность операторов не выполняется, если значение переменной управления превышает конечное значение, то есть значение переменной управления больше, или, соответственно, меньше конечного значения, если значение инкремента положительное, или, соответственно, отрицательное. Итерация завершается, когда значение переменной управления находится вне диапазона, заданного конструкцией TO. Пример использования оператора FOR приведен в таблице 32.

10.5.2.4 WHILE

Оператор WHILE вызывает выполнение последовательности операторов до ключевого слова END_WHILE. Операторы выполняются повторно до тех пор, пока связанное логическое выражение станет ложным. Если выражение изначально ложное, то группа операторов вообще не выполняется.

10.5.2.5 REPEAT

Оператор REPEAT вызывает последовательность операторов до ключевого слова UNTIL для выполнения повторно (и, по крайней мере, однократно), пока связанное логическое условие является истинным.

10.5.2.6 RETURN

Оператор RETURN обеспечивает ранний выход из функции, функционального блока или программы (например, как результат оценки оператора IF). Перед выходом из функции необходимо полученное значение присвоить выходной переменной с именем функции.

10.5.2.7 EXIT

Оператор EXIT используется для завершения итераций до удовлетворения условия завершения. Когда оператор EXIT размещается внутри вложенных итеративных конструкций, выход происходит от внутреннего цикла, в котором размещен EXIT, то есть управление переходит к следующему оператору после терминатора первого цикла (END_FOR, END_WHILE или END_REPEAT), за которым следует оператор EXIT.