

Владимирский государственный университет

М. С. ДЕНИСОВ П. А. ЧЕБОТАРЕВ К. Е. ДАВЫДОВ

**ОСНОВЫ ПРОГРАММИРОВАНИЯ
ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ
В ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ**

Лабораторный практикум

Владимир 2026

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

М. С. ДЕНИСОВ П. А. ЧЕБОТАРЕВ К. Е. ДАВЫДОВ

ОСНОВЫ ПРОГРАММИРОВАНИЯ
ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ
В ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ

Лабораторный практикум

Электронное издание



Владимир 2026

ISBN 978-5-9984-2208-9

© ВлГУ, 2026

УДК 004.312

ББК 32.97

Рецензенты:

Кандидат технических наук, доцент
доцент кафедры технологии машиностроения
Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых
А. В. Жданов

Доктор технических наук, доцент
профессор кафедры автоматизированных технологических систем
Брянского государственного технического университета
Д. И. Петрешин

Издается по решению редакционно-издательского совета ВлГУ

Денисов, М. С. Основы программирования логических контроллеров в промышленной автоматизации [Электронный ресурс] : лаб. практикум / М. С. Денисов, П. А. Чеботарев, К. Е. Давыдов ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2026. – 112 с. – ISBN 978-5-9984-2208-9. – Электрон. дан. (5,97 Мб). – 1 электрон. опт. диск (CD-ROM). – Систем. требования: Intel от 1,3 ГГц ; Windows XP/7/8/10 ; Adobe Reader ; дисковод CD-ROM. – Загл. с титул. экрана.

Посвящен изучению работы с логическими контроллерами фирмы UniLogic и разработке базовых проектов для промышленной автоматизации на примере стеклоформовочного производства.

Предназначен для студентов вузов направления подготовки 15.03.04 «Автоматизация технологических процессов и производства» всех форм обучения.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Ил. 131. Табл. 16. Библиогр.: 10 назв.

ISBN 978-5-9984-2208-9

© ВлГУ, 2026

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. ПРАКТИЧЕСКАЯ СХЕМОТЕХНИКА.	5
ВИДЫ СХЕМ И ИХ КОМПОНЕНТОВ	5
<i>Контрольные вопросы</i>	10
2. ОБЩЕЕ ВВЕДЕНИЕ В РАБОТУ ПЛК	11
<i>Контрольные вопросы</i>	21
3. ВВЕДЕНИЕ В БУЛЕВУ АЛГЕБРУ.....	22
ТАЙМЕРЫ И СЧЕТЧИКИ. ПОДПРОГРАММЫ	22
<i>Контрольные вопросы</i>	31
4. ПРОТОКОЛЫ, ИНТЕРФЕЙСЫ	32
<i>Контрольные вопросы</i>	38
5. АНАЛОГОВАЯ ОБРАБОТКА И РАЗРАБОТКА ЧМИ	39
<i>Контрольные вопросы</i>	45
6. АРХИТЕКТУРА ПЛК. ПОДКЛЮЧЕНИЕ ПЛК.	46
ПОДКЛЮЧЕНИЕ К ПЛК. БЕЗОПАСНОСТЬ В ПЛК	46
<i>Контрольные вопросы</i>	51
ПРАКТИКУМ	52
Лабораторная работа № 1. ОСНОВЫ СХЕМОТЕХНИКИ	52
Лабораторная работа № 2. ЗНАКОМСТВО СО СРЕДОЙ ПРОГРАММИРОВАНИЯ ПЛК UNILOGIC. ВВЕДЕНИЕ В РАБОТУ ПЛК	63
Лабораторная работа № 3. ПРОГРАММИРОВАНИЕ ТАЙМЕРОВ..	75
Лабораторная работа № 4. ПРОГРАММИРОВАНИЕ СЧЕТЧИКОВ	87
Лабораторная работа № 5. РАЗРАБОТКА ТЕСТОВОГО ЭКРАНА НМІ.....	93
ЗАКЛЮЧЕНИЕ.....	110
СПИСОК БИБЛИОГРАФИЧЕСКИХ ИСТОЧНИКОВ	111
РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК	111

ВВЕДЕНИЕ

В настоящее время в промышленности делают большой акцент на внедрении компьютерных систем автоматизации технологических процессов, за счет которых удастся повысить эффективность работы оборудования и, как следствие, улучшить качество производимой продукции.

В лабораторном практикуме представлены материалы, необходимые для создания компьютерных систем автоматизации, начиная со схемотехники, заканчивая программированием логических контроллеров и созданием экрана человеко-машинного интерфейса для непосредственного управления технологическими процессами.

В теоретической части издания представлены практическая схемотехника, общее введение в работу программируемого логического контроллера, основы программирования счетчиков и таймеров, информация по протоколам и интерфейсам связи, которые необходимы для осуществления взаимодействия между средствами автоматизации, способы обработки сигналов с датчиков, а также структуры и архитектуры компьютерных систем управления.

В лабораторных работах рассматриваются программирование контроллеров и НМІ-панелей в программной среде UniLogic на примере стеклоформовочного производства; вопросы построения электрических схем и их осуществление на программном уровне в UniLogic.

1. ПРАКТИЧЕСКАЯ СХЕМОТЕХНИКА. ВИДЫ СХЕМ И ИХ КОМПОНЕНТОВ

Перед тем как начать изучение схемотехники, необходимо понять, зачем нужны схемы. Рассмотрим данный вопрос поэтапно:

– на этапе проектирования – для определения структуры будущего объекта автоматизации (варочной печи, стеклоформовочной машины, конвейерной ленты и т. д.);

– этапе производства – для ознакомления с конструкцией объекта, разработки технологических процессов изготовления, монтажа и контроля изделия (продукции);

– этапе эксплуатации – для определения неисправностей, ремонта и технического обслуживания объекта.

Электрическая схема – документ, содержащий в виде условных обозначений или изображений составные части изделия, действующие при помощи электроэнергии, и их взаимосвязи. На предприятиях в работе чаще всего используют следующие виды схем:

- 1) структурные электрические схемы;
- 2) функциональные электрические схемы;
- 3) принципиальные электрические схемы;
- 4) монтажные схемы;
- 5) кабельные планы;
- 6) топологические электрические схемы;
- 7) мнемоническую схему (SCADA).

Рассмотрим подробнее основные схемы для разработки автоматизированной системы управления технологическим процессом.

Функциональные электрические схемы автоматизации (ФСА) предназначены для разъяснения процессов, происходящих в отдельных функциональных цепях объекта или в объекте в целом.

При различных режимах работы можно разрабатывать несколько функциональных схем.

Принципиальные электрические схемы – схемы, которые определяют состав взаимосвязанных элементов и дают представление о принципах работы описанного продукта или системы.

Монтажные схемы – схемы, которые предоставляют информацию о конструкции какого-либо изделия или системы, а также информацию по сбору элементов изделия или системы между собой. Такие

схемы необходимы для монтажа многих видов средств автоматизации и сборки электрических шкафов управления (ШУ).

Кабельные планы – чертежи, показывающие расположение и марки электрических проводов и кабелей. При разработке кабельных планов действуют лишь общие требования к оформлению конструкторской документации.

Мнемоническая схема – упрощенная модель объекта, облегчающая запоминание его схемы, назначения оборудования, а также управления при различных режимах работы.

Существуют определенные правила выполнения электрических схем:

- схемы выполняют для изделий, находящихся в отключенном положении;
- номера устройствам или же элементам схемы присваивают по порядку начиная с единицы, также используют буквенные выражения, например R1, R2, R3 (для резисторов), C1, C2, C3 (для конденсаторов);
- номера присваивают на схеме в порядке сверху вниз и слева направо;
- обозначения позиций проставляют на схеме с правой стороны или над условными графическими обозначениями (УГО) устройств и (или) элементов.

Рассмотрим условные графические обозначения, которые чаще всего используют при разработке печатных плат и силовых элементов.

Транзистор (рис. 1) – активный полупроводниковый прибор с несколькими электрическими переходами и тремя выводами, предназначенный для усиления сигнала и генерации колебаний.

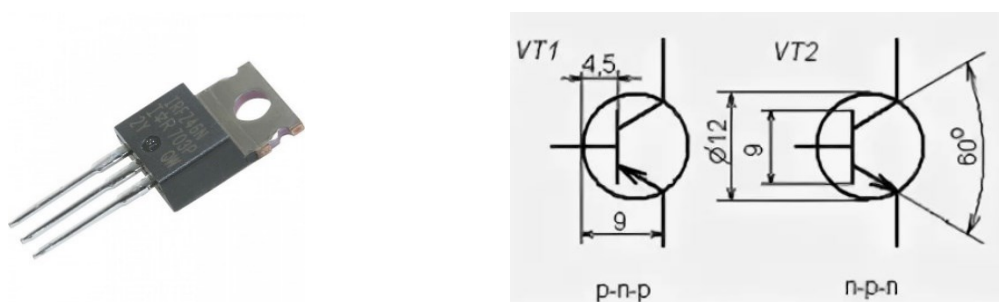


Рис. 1. Транзистор (VT)

Резистор (рис. 2) – элемент электрической цепи (обычно реализованный в виде законченного изделия), основное функциональное назначение которого – оказывать известное активное сопротивление электрическому току.

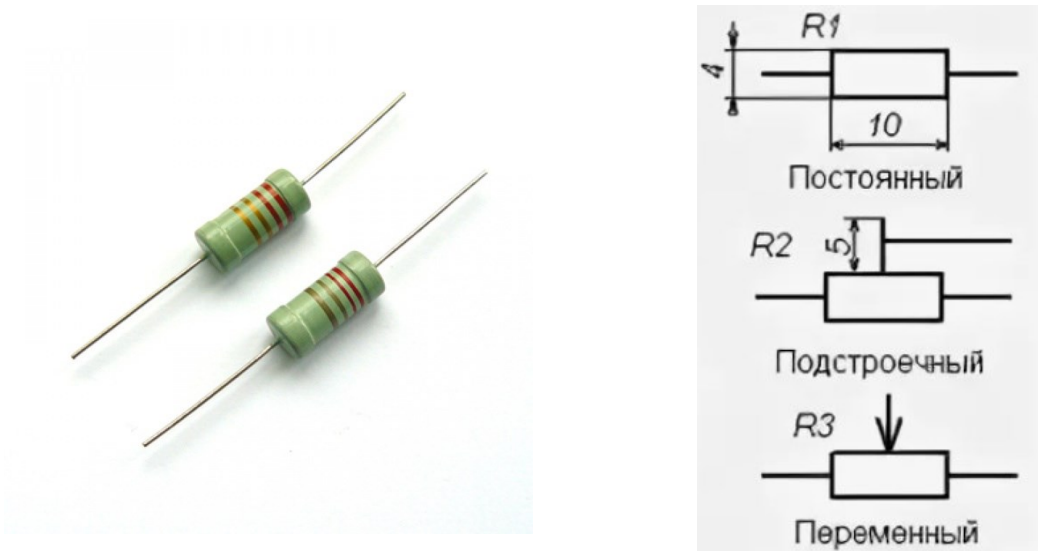


Рис. 2. Резистор (R)

Конденсатор (рис. 3) – электронный компонент, предназначенный для накопления, хранения, фильтрации и регулирования тока в электрических цепях.

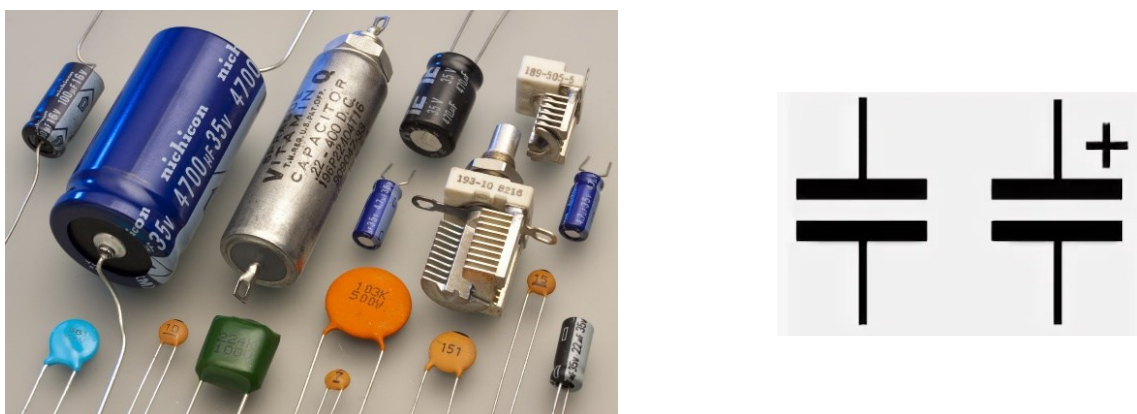


Рис. 3. Конденсатор (C)

Диод в схемотехнике (рис. 4) – активный электрический элемент, проводящий ток только в одном направлении.



Рис. 4. Диод (VD, D)

Интегральная схема (ИС), интегральная микросхема (ИМС), микросхема (рис. 5) – микроэлектронное устройство – электронная схема произвольной сложности (кристалл), изготовленная на полупроводниковой подложке (пластине или пленке) и помещенная в неразборный корпус или входящая в состав микросборки.

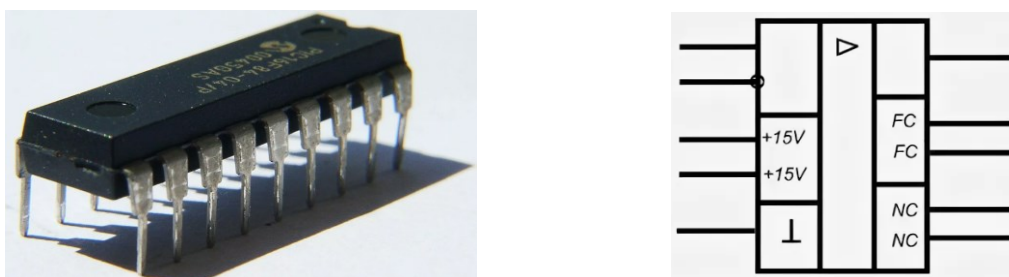


Рис. 5. Микросхемы (DA, D)

Двигатель (рис. 6) – устройство, преобразующее какой-либо вид энергии в механическую работу.

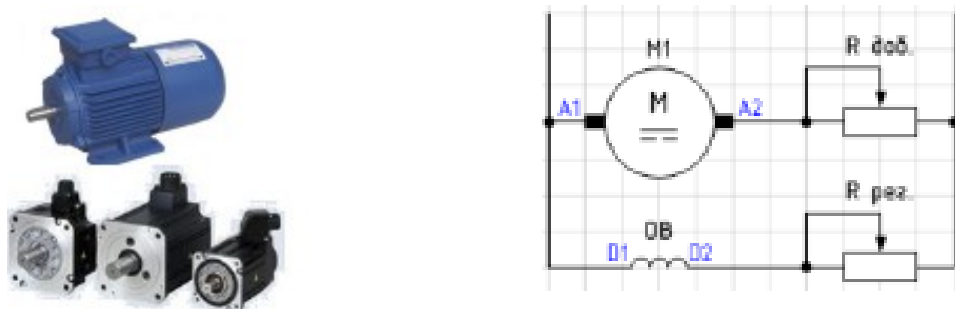


Рис. 6. Двигатель, мотор (M)

Трансформатор (рис. 7) – электрическое устройство, предназначенное для преобразования переменного тока одного напряжения в переменный ток другого напряжения при сохранении частоты.



Рис. 7. Трансформатор (T, TV)

Контактор (рис. 8) – разновидность электромагнитных реле, рассчитанная на большие токи. Предназначен для постоянной коммутации (включения/отключения) электроцепей и обеспечения бесперебойной работы электрического оборудования.



Рис. 8. Электромагнитное реле (K, Y), пускатель и контактор (KM)

Источник электропитания (рис. 9) – устройство, предназначенное для обеспечения питания электроприбора электрической энергией при соответствии требованиям параметров последней (напряжения, тока и т. д.) путем преобразования энергии других источников питания.



Рис. 9. Источник питания (GB)

Выключатель (рис. 10) – электротехническое устройство для размыкания или замыкания электрической цепи.



Рис. 10. Выключатели, переключатели (НО и НЗ)

Контрольные вопросы

1. Что такое электрическая схема?
2. Какие бывают виды схем?
3. Что такое монтажная схема?
4. Что такое резистор?
5. Что такое конденсатор?
6. Что такое интегральные схемы?
7. Какова разница между двигателем и трансформатором?

2. ОБЩЕЕ ВВЕДЕНИЕ В РАБОТУ ПЛК

В данном разделе разберем состав программируемого логического контроллера (ПЛК), основные принципы его работы и функции.

В состав ПЛК входят следующие компоненты (рис. 11):

1) *центральный процессор управления* (ЦПУ, CPU), который чаще всего реализуется на 32-разрядных процессорах. В некоторых случаях ЦПУ могут быть выполнены с использованием однокристальных микроЭВМ;

2) *карта памяти* – флэш-память, на которую записывают коды управляющих программ; служит как для кратковременного, так и для долговременного хранения данных;

3) *блок оперативного запоминающего устройства (ОЗУ)* – блок, в который встраивается карта памяти; необходим для стабилизации флэш-памяти;

4) *сетевые интерфейсы* – устройства для подключения контроллера к другим элементам автоматики; могут быть встроенными или реализованными в виде отдельного модуля;

5) *источник питания* – модуль для питания ПЛК; может быть встроенным или реализованным в виде отдельного модуля.

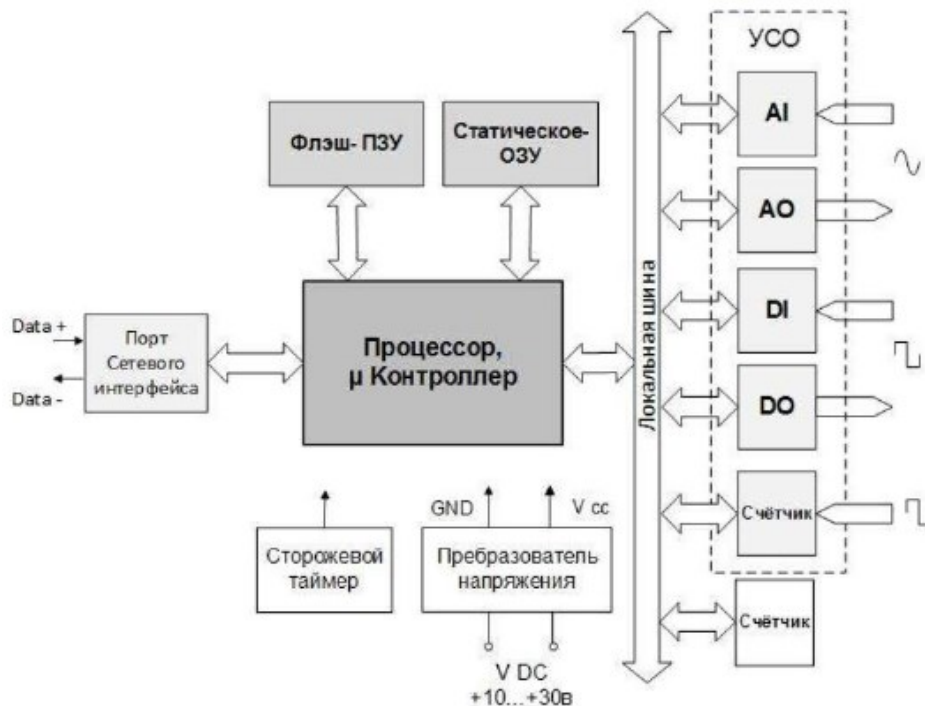


Рис. 11. Взаимодействие процесса и контроллера

Принцип работы ПЛК

Программируемый логический контроллер работает по принципу обработки входных сигналов, управления выходами и выполнения заданных программ управления.

Программируемый логический контроллер работает по циклу, состоящему из четырех фаз (рис. 12).

1. *Получение данных с контрольно-измерительных приборов (опрос входов).* Программируемый логический контроллер производит опрос всех подключенных к нему устройств, принимает входные сигналы, переданные на его входы.

2. *Выполнение управляющей программы.* Пользователь заранее пишет программу управления, загружает ее в ПЛК, после чего программа исполняется.

3. *Установка значений для передачи на исполнительные устройства.* После того как выполняется программа, устанавливает значения на своих выходах.

4. *Дополнительные вспомогательные операции.* На данном этапе выполняются дополнительные операции, такие как обновление программного обеспечения, проверка состояния входов, реализация визуализации и т. д.

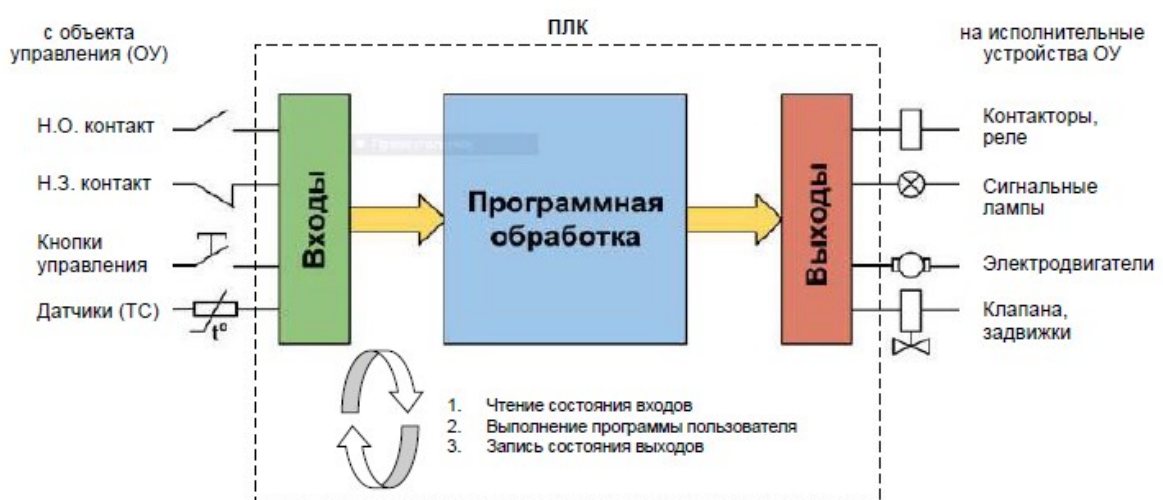


Рис. 12. Принцип работы ПЛК

В качестве периферии для ПЛК выступают входы и выходы, которые делятся на три типа:

1) *аналоговые* – данный вид соединения необходим для ввода и вывода различных непрерывных сигналов, уровней тока и напряжения, которые соответствуют определенной физической величине в каждый момент времени;

2) *дискретные* – данный вид соединения предназначен для ввода и вывода информации с разных устройств и дискретных датчиков в виде кода;

3) *специализированные* – данный вид соединения подходит для взаимодействия со специфическими датчиками, требующими нестандартных уровней сигналов и специальной обработки.

Возможные вариации отображения периферии представлены на рис. 13.



Рис. 13. Вариации отображения периферии

Вместе с ПЛК используют различные модули расширения и специального назначения. По назначению их делят на следующие группы:

- источники напряжения;
- модули для расширения входов и выходов;
- модули ввода/вывода дискретных сигналов;
- модули ввода/вывода аналоговых сигналов;
- модули для реализации интерфейса.

Для взаимодействия контроллера с модулями и осуществления реальной работы ПЛК в технологическом процессе используют различные программные продукты.

Системное программное обеспечение:

- отвечает за контроль аппаратных средств ПЛК;
- отвечает за индикацию работы памяти и тестирование; взаимодействие с источниками питания, модулями ввода/вывода и интерфейсов, часов реального времени и таймеров.

Составная часть системного программного обеспечения – система исполнения программного кода программы.

Код системного программного обеспечения расположен в постоянном запоминающем устройстве (ПЗУ) и может быть изменен только изготовителем программного логического контроллера.

Прикладное программное обеспечение (ППО) создает пользователь ПЛК при помощи какой-либо системы программирования. Перепрограммирование может быть многократным. Код ППО размещается в энергонезависимой памяти.

К основным параметрам программируемого логического контроллера относятся:

- каналы ввода и вывода различных типов;
- вид ЦПУ (такты частота, архитектура, энергопотребление, производительность, нормы литографического процесса);
- количество памяти запоминающего устройства для данных и написанных программ;
- время быстрогодействия реакции по каналам ввода/вывода на перемены сигнала (сигналов), циклов управления;
- устойчивость к температурным и механическим воздействиям, а также к влажности, давлению и попаданию пыли и грязи;
- качество функционирования электроники в ЦПУ, устойчивость к сбоям, резервирование, время наработки на отказ;
- цена.

UniStream PLC – высокопроизводительный и надежный контроллер с новой концепцией – виртуальным человеко-машинным интерфейсом (ЧМИ, HMI) (рис. 14).

Мощный многофункциональный ПЛК предоставляет пользователям расширенную поддержку связи, разнообразные конфигурации ввода/вывода и виртуальный ЧМИ.

Встроенные программируемые контроллеры UniStream предназначены для сложных устройств и проектов автоматизации.

Мощные программируемые контроллеры с уникальными функциями ЧМИ на базе самого эффективного программного обеспечения позволяют выполнять высококлассные проекты в срок, не выходя за рамки бюджета.



Рис. 14. ПЛК UniStream

Общие характеристики ПЛК UniStream

Программируемые логические контроллеры UniStream представляют собой ПЛК, монтируемые на DIN-рейку, со встроенной конфигурацией ввода/вывода.

Серия доступна в трех версиях: расширенной (Pro), стандартной (Standard) и базовой (Basic).

UniStream программируются с помощью UniLogic для настройки оборудования, связи, приложений ПЛК и ЧМИ.

Возможности управления ПЛК UniStream:

- встроенные тренды и измерительные приборы;
- автоматическая настройка пропорционально-интегрально-дифференциального регулирования;
- таблицы с различными данными, их выборки и наборы параметров;
- UniApps: мониторинг, устранение ошибок и неполадок, доступ для редактирования данных, отладка и т. д.;
- безопасность: защита, состоящая из нескольких уровней, со встроенным паролем;
- сигнализирование: встроенная система, промышленные стандарты.

Опции связи:

- встроенные порты: два порта Ethernet;
- один USB-хост, один порт для устройства USB;
- поддержка дополнительных портов CANbus (порт CAN-шины), порты RS232/485.

Протоколы связи:

- промышленные шины: CANopen, CAN Layer2, MODBUS, Ether-NetIP;
- дополнительно: SNMP Agent/Trap, электронная почта, смс, модемы, GPRS/GSM, FTP-сервер/клиент, веб-сервер, SQL и MQTT.
- удаленный доступ через любое устройство, поддерживающее VNC;

Место установки ПЛК:

- необходимо установить устройство на стандартную DIN-рейку;
- по бокам устройства должно быть достаточно места для любых модулей ввода/вывода или модулей связи;
- необходимо надвигать устройство на DIN-рейку, пока зажимы, расположенные в верхней и нижней частях устройства, не защелкнутся;
- при правильной установке устройство располагается на DIN-рейке под прямым углом.

Данное оборудование предназначено исключительно для работы в соответствии с требованиями, установленными для безопасного сверхнизкого напряжения. Все блоки питания в системе должны иметь двойную изоляцию. Выходное напряжение блоков питания должно соответствовать требованиям, установленным для безопасного сверхнизкого напряжения. Нельзя подключать проводник, помеченный маркировкой «Нейтраль» или «Линия» сети переменного тока 110/220 В, к клемме устройства с маркировкой «0V». Нельзя прикасаться к оголенным проводам, находящимся под напряжением. Все операции, связанные с монтажом электропроводки, должны выполняться при выключенном питании.

Для защиты от больших токов в точке подключения источника питания необходимо использовать защиту от перегрузки по току – плавкий предохранитель или автоматический выключатель. Перед подачей питания необходимо перепроверить всю разводку.

Монтаж ПЛК:

Подключение питания к клеммам «+V» и «0V».

Отдельно подключайте каждую точку заземления к монтажной панели металлического шкафа (рис. 15).

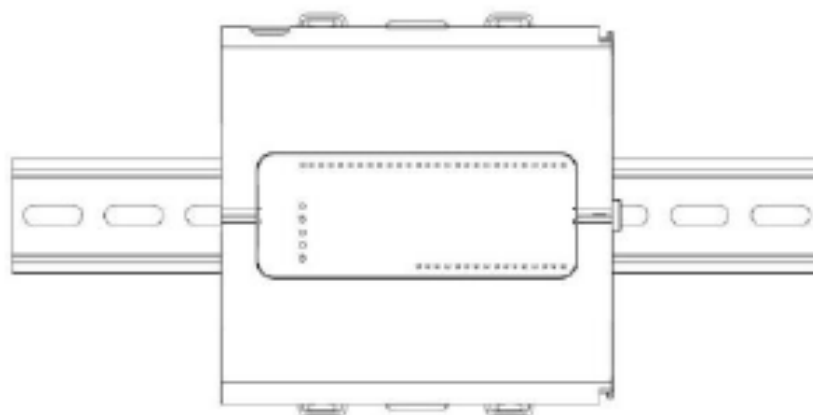


Рис. 15. Установка ПЛК

1. Подключение цифровых входов.

Все 10 цифровых входов имеют общую точку СМ0. Цифровые входы могут быть скреплены проводом как источник или приёмник (рис. 16).

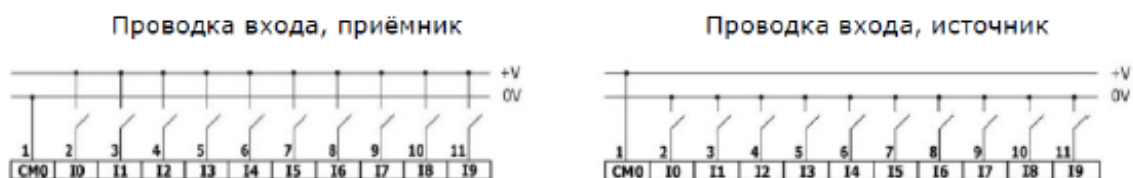


Рис. 16. Проводка приемник/источник

2. Подключение аналоговых входов.

Оба входа имеют общую точку СМ1. Каждый вход предусматривает два режима: напряжение или ток. Режим определяется конфигурацией оборудования в программном приложении UniLogic. Если подключить вход к токовому входу, то в программном приложении необходимо установить его на токовый вход (рис. 17).

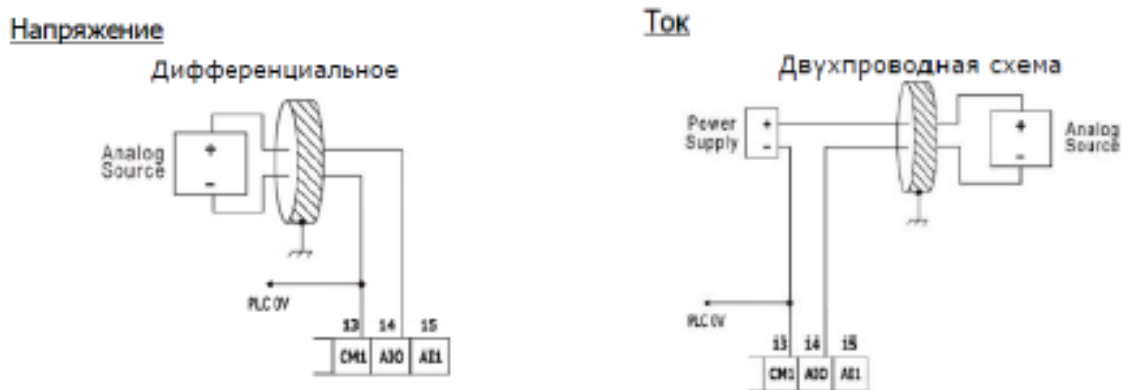


Рис. 17. Передача электрического сигнала

3. Подключение релейных выходов.

Релейные выходы расположены в двух изолированных группах: О0 – О3 разделяют общий обратный провод CM2; О4 – О7 разделяют общий обратный провод CM3.

4. Подключение транзисторных выходов.

Аналоговые выходы не изолированы между собой. Подключение клемм +VO и 0VO показано на рис. 17. Выходы О0 – О11 имеют общий обратный провод 0VO.

Для увеличения срока службы релейных контактов и защиты контроллера от обратной ЭДС необходимо подключать:

- защитный диод параллельно каждой индуктивной нагрузке постоянного тока (DC Load);
- демпфирующую цепь RC, включенную параллельно каждой индуктивной нагрузке переменного тока (AC Load).

Среда программирования ПЛК UniLogic

Программное обеспечение UniLogic предназначено для программирования и обслуживания логических контроллеров серии UniStream фирмы Unitronics (рис. 18).

Применение:

- конфигурирование оборудования: ПЛК, экраны НМІ, модули ввода/вывода и коммуникационные модули (COM);
- конфигурирование коммуникаций;
- построение управляющих приложений на базе релейной логики;
- разработка экранов НМІ.

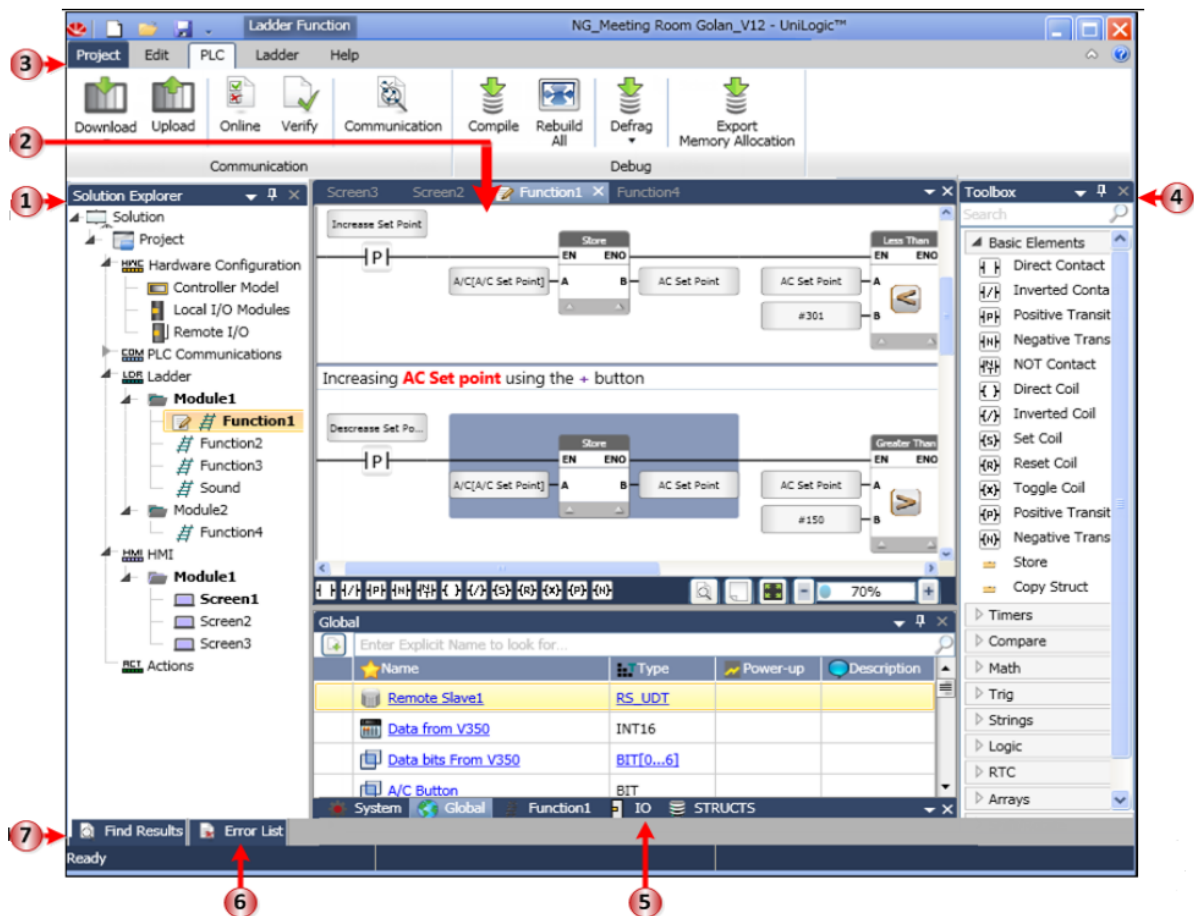


Рис. 18. Интерфейс программного обеспечения

Программирование и отладка технических средств автоматизации (ТСА) для пневматической стеклоформирующей секции могут быть выполнены на специальном учебном стенде, изображенном на рис. 19.

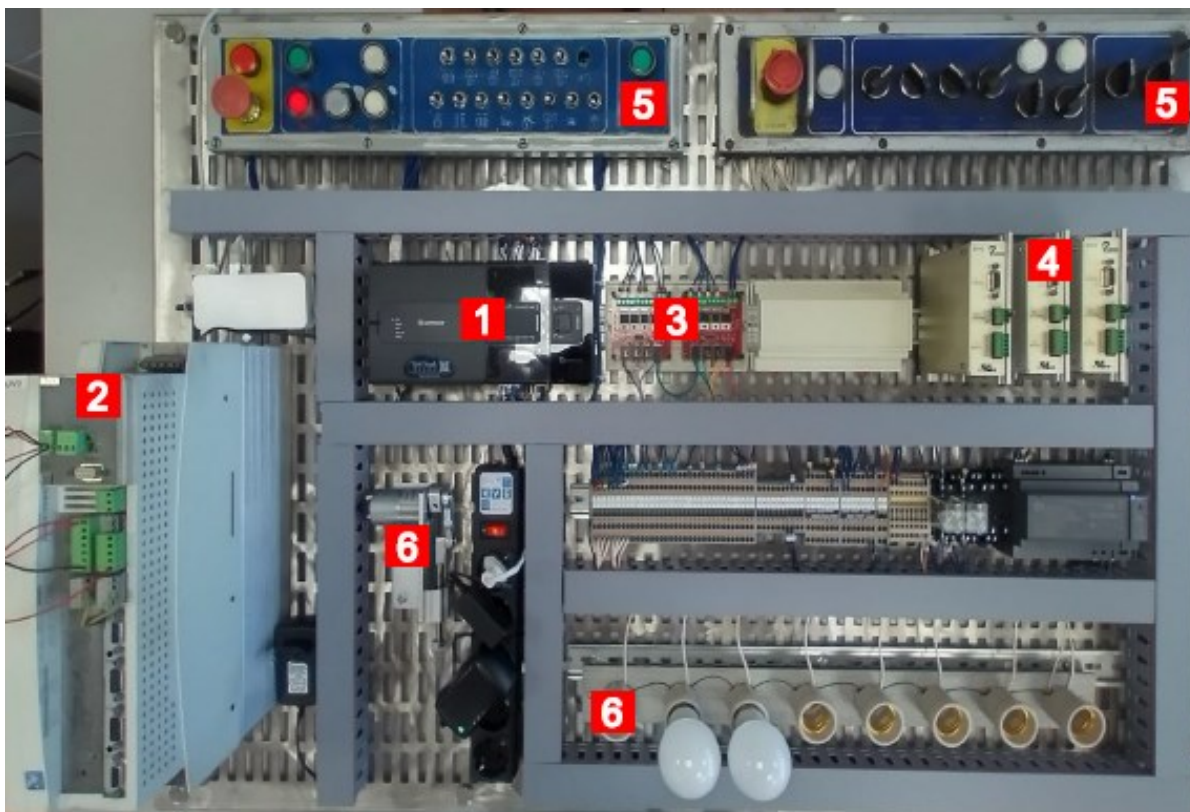


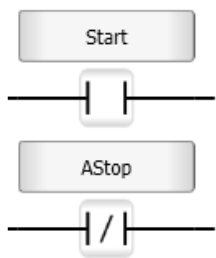
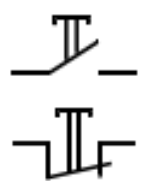
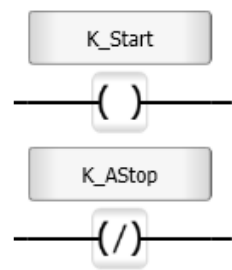
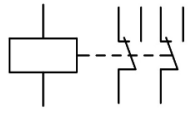
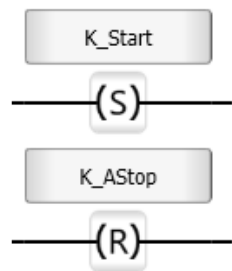
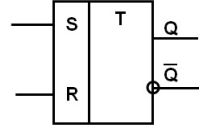
Рис. 19. Учебный стенд

В состав учебного стенда входят:

- 1) эталонный ПЛК;
- 2) сервоусилитель (Lenze);
- 3) драйвер оптогальванический;
- 4) драйвер шагового двигателя;
- 5) панели управления черновой и чистовой формы;
- 6) средства визуализации (лампы и двигатели постоянного тока).

Для управления учебным стендом используют команды релейной логики, представленные в табл. 1.

Команды релейной логики

Название элемента	Описание	Схемное обозначение в UniLogic	Обозначение УГО
Нормально открытый (НО) и нормально закрытый (НЗ) контакт (прямой и инверсный)	Эти команды получают значение из памяти ПЛК или из регистров входов/выходов образа процесса, если тип данных I или Q. Контакт НО замкнут (включен), когда бит равен 1. Контакт НЗ замкнут (включен), когда бит равен 0		
Выходная катушка (прямая, инверсная)	Когда выполняется команда Output, в выходном регистре образа процесса устанавливается выходной бит. При выполнении команды Output указанный бит устанавливается равным потоку сигнала на входе команды Output		
Выходная катушка (RS-триггер)	Когда исполняется команда S или R, устанавливается (включается) или сбрасывается (выключается) указанное количество разрядов (N), начиная со значения, определенного параметром (битом) команды		

Контрольные вопросы

1. Что входит в состав ПЛК?
2. Каков принцип работы ПЛК?
3. Что такое периферия ПЛК?
4. Какие модули расширения используют для ПЛК?
5. Что такое программное обеспечение UniStream?
6. Каковы общие характеристики ПЛК UniStream?
7. Перечислите рекомендации по использованию ПЛК.
8. Как происходят установка и подключение ПЛК?
9. Что входит в интерфейс НМІ-панели?
10. Опишите учебный стенд для стеклоформовочной машины.

3. ВВЕДЕНИЕ В БУЛЕВУ АЛГЕБРУ. ТАЙМЕРЫ И СЧЕТЧИКИ. ПОДПРОГРАММЫ

В булевой алгебре используют две системы счисления:

- непозиционную;
- позиционную.

Непозиционная система счисления представлена в табл. 2.

Таблица 2

Непозиционная система счисления

Римские цифры	I	V	X	L	C	D	M
Значение	1	5	10	50	100	500	1000

Если цифра, стоящая справа, больше цифры, которая находится слева, то из правой цифры вычитают левую цифру:

IV: $1 < 5$, следовательно, $5 - 1 = 4$;

XL: $10 < 50$, следовательно, $50 - 10 = 40$.

Цифры складывают, в том случае, если правая цифра меньше или равна левой цифре:

VI: $5 + 1 = 6$;

VIII: $5 + 1 + 1 + 1 = 8$;

XX: $10 + 10 = 20$.

Позиционные системы счисления представлены в табл. 3.

Таблица 3

Позиционная система счисления

Основание	Название	Обозначение	Алфавит
P = 2	Двоичная	x_2	{0, 1}
P = 3	Троичная	x_3	{0, 1, 2}
P = 8	Восьмеричная	x_8	{0, 1, 2, 3, 4, 5, 6, 7}
P = 10	Десятичная	x_{10}	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
P = 16	Шестнадцатеричная	x_{16}	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

При определении числа в какой-либо системе счисления используют следующую формулу:

$$A_p = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + a_{-m} p^{-m},$$

где A – число в системе счисления p ; p – основание системы счисления, a_i – цифры в системе счисления p ; n – количество целых разрядов числа A ; m – количество дробных разрядов числа A .

Виды систем счисления представлены в табл. 4.

Таблица 4

Виды систем счисления

Системы счисления				
Десятичная	Двоичная	Восьмеричная	Шестнадцатеричная	Двоично-десятичная
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	00010000
11	1011	13	B	00010001
12	1100	14	C	00010010
13	1101	15	D	00010011
14	1110	16	E	00010100
15	1111	17	F	00010101
16	10000	20	10	00010110

В случае программирования таймеров и счетчиков нам необходимо использовать двоичную систему счисления – систему счисления с основанием 2 $\{0, 1\}$:

0 (Ложь, False);

1 (Истина, True).

Так, например, число 12 в двоичной системе будет выглядеть как 1100 (рис. 20).

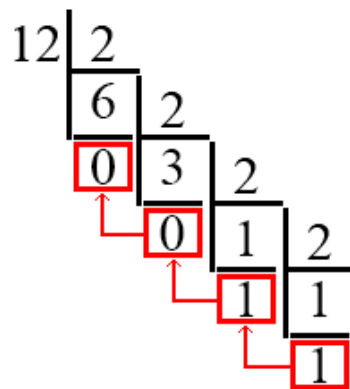


Рис. 20. Число 12 в двоичной системе

Также при программировании таймеров и счетчиков используют различные логические операции, некоторые из них представлены в табл. 5.

Таблица 5

Логические операции для программирования счетчиков и таймеров

Обозначение	Название
$\neg A$	НЕ, отрицание, инверсия
$A \& B, A * B$	И, конъюнкция, логическое умножение
$A \vee B, A + B, A B$	ИЛИ, дизъюнкция, логическое сложение

Приоритет выполнения имеют следующие логические операции: \neg , $\&$, \vee .

Булевы команды (битовые логические команды), используемые в программе UniLogic, представлены в табл. 6.

Булевы команды

Название элемента	Описание	Схемное обозначение в UniLogic															
Элемент И (And)	<p>Эта команда может быть использована для манипулирования сигналами булевой операции И (конъюнкция)</p> <table border="1"> <tr> <td>a</td> <td>b</td> <td>$a \wedge b$</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	a	b	$a \wedge b$	0	0	0	0	1	0	1	0	0	1	1	1	
a	b	$a \wedge b$															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
Элемент ИЛИ (Or)	<p>Эта команда может быть использована для манипулирования сигналами булевой операции ИЛИ (дизъюнкция)</p> <table border="1"> <tr> <td>a</td> <td>b</td> <td>$a \vee b$</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	a	b	$a \vee b$	0	0	0	0	1	1	1	0	1	1	1	1	
a	b	$a \vee b$															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
Элемент УСТАНОВИТЬ БИТ (Set Bit), СБРОСИТЬ БИТ (Reset Bit)	<p>Эта команда может быть использована для манипулирования сигналами булевой операции НЕ (отрицание)</p> <table border="1"> <tr> <td>a</td> <td>$\neg a$</td> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	a	$\neg a$	0	1	1	0										
a	$\neg a$																
0	1																
1	0																

Также при программировании таймеров и счетчиков в программе UniLogic используют операции сравнения. В табл. 7 представлены существующие операции сравнения, в табл. 8 – их расшифровка в программном обеспечении UniLogic.

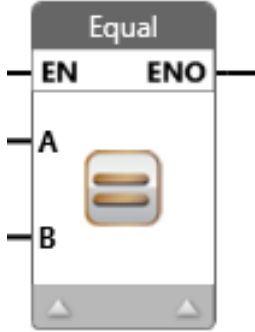
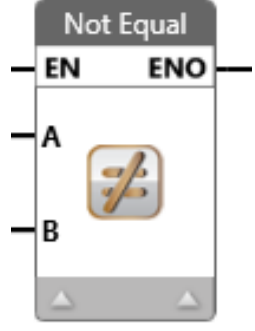
Таблица 7

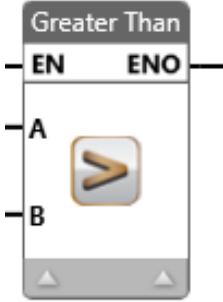
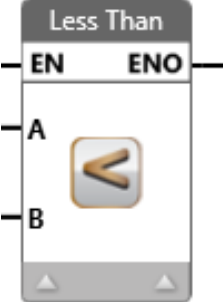
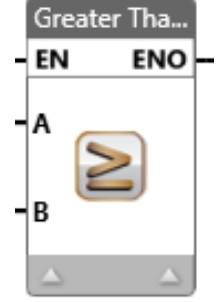
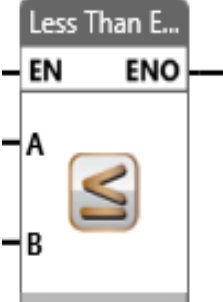
Операции сравнения

Операция	Назначение	Пример
==	Равно	I == 0
!=	Не равно	K != 15
>	Больше, чем	Z > 15.2
<	Меньше, чем	Tab < -132.654
>=	Больше или равно	Z11 >= 0
<=	Меньше или равно	Y2 <= 10

Таблица 8

Операции сравнения в UniLogic

Название элемента	Описание	Схемное обозначение в UniLogic
Значение A = B	Контакт ENO включен, когда сравнение A = B истинно. Сравнить можно только переменные одного типа	 <p>The image shows the UniLogic 'Equal' comparison symbol. It is a rectangular block with a dark grey header labeled 'Equal'. Below the header, there are two terminals: 'EN' on the left and 'ENO' on the right. On the left side, there are two input terminals labeled 'A' and 'B'. The central area contains a yellow square icon with two horizontal bars, representing the equals sign (=). At the bottom, there are two small upward-pointing triangles.</p>
Значение A != B	Контакт ENO включен, когда сравнение A != B истинно. Сравнить можно только переменные одного типа	 <p>The image shows the UniLogic 'Not Equal' comparison symbol. It is a rectangular block with a dark grey header labeled 'Not Equal'. Below the header, there are two terminals: 'EN' on the left and 'ENO' on the right. On the left side, there are two input terminals labeled 'A' and 'B'. The central area contains a yellow square icon with a diagonal slash through it, representing the not-equals sign (≠). At the bottom, there are two small upward-pointing triangles.</p>

Название элемента	Описание	Схемное обозначение в UniLogic
Значение $A > B$	Контакт ENO включен, когда сравнение $A > B$ истинно. Сравнить между собой можно только переменные одного типа	 <p>The symbol is a rectangular box with a grey header labeled 'Greater Than'. Below the header, there are two terminals labeled 'EN' and 'ENO'. On the left side, there are two input terminals labeled 'A' and 'B'. In the center of the box is a large orange greater-than sign (>). At the bottom, there are two small grey triangles pointing upwards.</p>
Значение $A < B$	Контакт ENO включен, когда сравнение $A < B$ истинно. Сравнить можно только переменные одного типа	 <p>The symbol is a rectangular box with a grey header labeled 'Less Than'. Below the header, there are two terminals labeled 'EN' and 'ENO'. On the left side, there are two input terminals labeled 'A' and 'B'. In the center of the box is a large orange less-than sign (<). At the bottom, there are two small grey triangles pointing upwards.</p>
Значение $A \geq B$	Контакт ENO включен, когда сравнение $A \geq B$ истинно. Сравнить можно только переменные одного типа	 <p>The symbol is a rectangular box with a grey header labeled 'Greater Than...'. Below the header, there are two terminals labeled 'EN' and 'ENO'. On the left side, there are two input terminals labeled 'A' and 'B'. In the center of the box is a large orange greater-than-or-equal sign (\geq). At the bottom, there are two small grey triangles pointing upwards.</p>
Значение $A \leq B$	Контакт ENO включен, когда сравнение $A \leq B$ истинно. Сравнить можно только переменные одного типа	 <p>The symbol is a rectangular box with a grey header labeled 'Less Than E...'. Below the header, there are two terminals labeled 'EN' and 'ENO'. On the left side, there are two input terminals labeled 'A' and 'B'. In the center of the box is a large orange less-than-or-equal sign (\leq). At the bottom, there are two small grey triangles pointing upwards.</p>

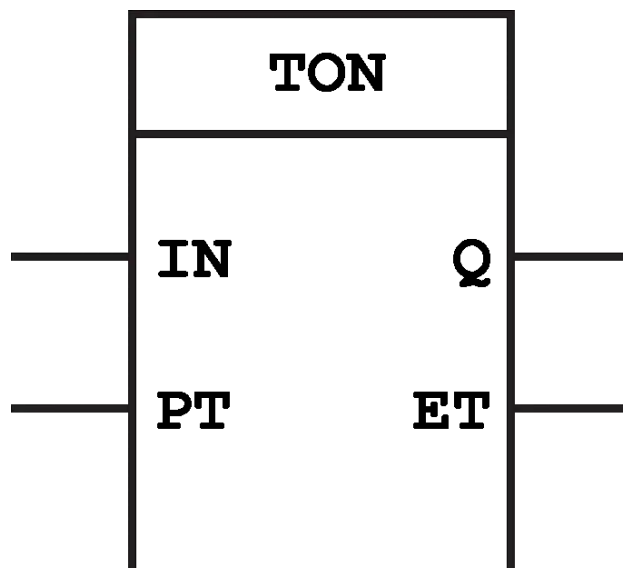


Рис. 22. Функциональный блок TON

Алгоритм работы функционального блока TOF (рис. 23).

1. ЕСЛИ (вход $IN = TRUE$) $\Rightarrow ET = 0$ и $Q = TRUE$.
2. ЕСЛИ $(ET < PT) \& (IN = FALSE) \Rightarrow$ Включение таймера и начало отсчета времени ($ET = ET + 1$).
 - 2.1. Отсчет времени ($ET = ET + 1$).
3. ЕСЛИ $(ET = PT) \& (IN = FALSE) \Rightarrow Q = FALSE$.

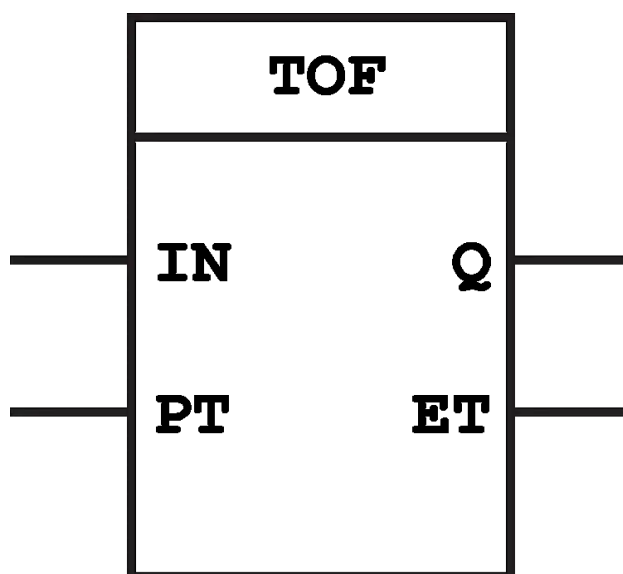


Рис. 23. Функциональный блок TOF

Алгоритм работы функционального блока ТР (рис. 24).

1. ЕСЛИ (вход $IN = FALSE$) $\Rightarrow ET = 0$ и $Q = FALSE$.

2. ЕСЛИ ($ET < PT$) & ($IN = TRUE$) \Rightarrow Включение таймера и начало отсчета времени ($ET = ET + 1$).

2.1. ЕСЛИ (вход $IN = TRUE$) $\Rightarrow Q = TRUE$.

2.2. Отсчет времени ($ET = ET + 1$) \Rightarrow генерация импульсов заданной длительности.

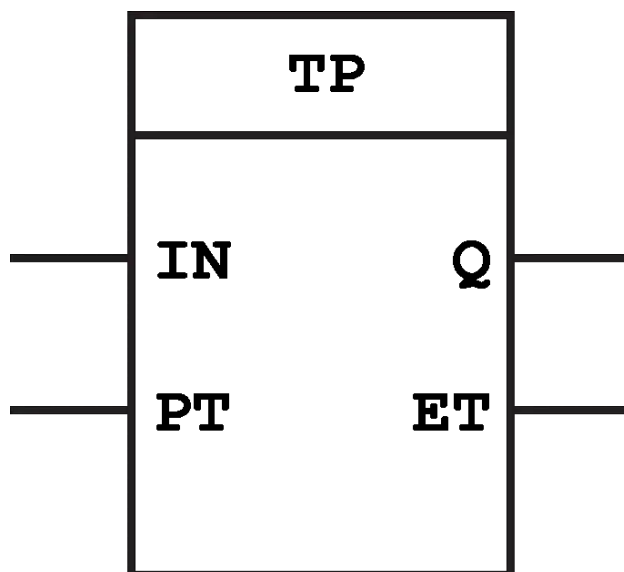


Рис. 24. Функциональный блок ТР

Помимо таймеров используют три типа счетчиков (рис. 25) по ГОСТ Р МЭК 61131-3-2016.



Рис. 25. Три типа счетчиков

Блок для программирования счетчика можно представить следующим образом (рис. 26).

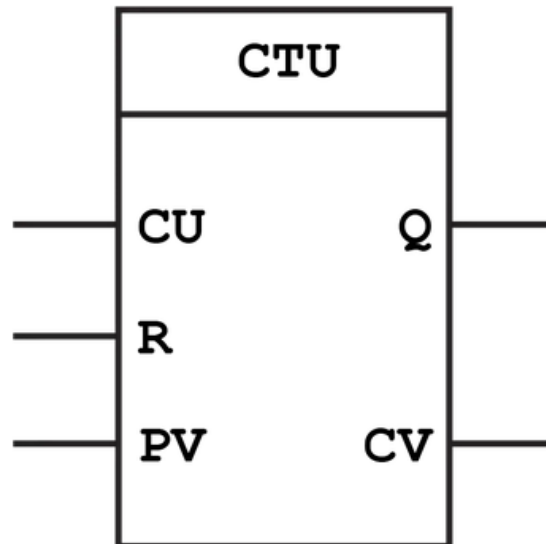


Рис. 26. Блок программирования счетчика

Обозначения: вход **CU (BOOL)** – вход прямого счета; вход **R (BOOL)** – вход сброса прямого счета; вход **PV (BOOL)** – предустановленное значение; выход **Q (BOOL)** – результат работы счетчика; выход **CV (WORD)** – текущее значение счета.

Контрольные вопросы

1. Какие существуют системы счисления в булевой алгебре?
2. Расскажите про программирование таймеров.
3. Расскажите про программирование счетчиков.
4. Какие обозначения таймеров и счетчиков существуют в программном обеспечении UniLogic?

4. ПРОТОКОЛЫ, ИНТЕРФЕЙСЫ

Для решения задач проектирования систем автоматизации необходимо создать условия для сопряжения объектов управления и оборудования, задействованного в технологическом процессе, которые позволят объединить отдельные устройства в систему.

Под **устройством сопряжения** понимают технические средства, которые обеспечивают работу различных элементов автоматики, контрольно-измерительных приборов и исполнительных устройств. Также в виде интерфейса представляют совокупность протоколов и программного обеспечения для организации процесса обмена информацией между функциональными блоками.

Стеком протоколов в автоматизации называют согласованные протоколы разных уровней, которых достаточно для организации сетевого взаимодействия.

На каждом уровне выбирают определенный набор запросов и функций, при помощи которых происходит взаимодействие с близлежащим уровнем – интерфейсом.

Условия и правила для сопряжения двух устройств автоматизации описывают в виде набора действий для всех уровней по отдельности и в общем, которые называют *протоколами*.

Существует огромное количество стеков протоколов, повсеместно применяемых в сетях различных типов. Сюда относятся стеки, которые являются национальными и международными стандартами, и фирменные стеки, получившие популярность за счет востребованности оборудования различных фирм.

Одни из самых популярных стеков протоколов – стек IPX/SPX фирмы Novell; стек TCP/IP, который применяют в сети Интернет на базе операционной системы UNIX; стек OSI, используемый в международных организациях для стандартизации, и различные другие протоколы.

Выделяют несколько уровней стеков протоколов: транспортные, сетевые, прикладные.

1. К *транспортным протоколам* относятся:

– TCP (Transmission Control Protocol – Протокол управления передачей). Протокол стека TCP/IP, отвечает за организованную, надежную и безопасную доставку данных;

– NetBIOS (Базовая сетевая система ввода/вывода). NetBIOS предназначен для организации соединения между компьютерами, а NetBEUI позволяет передавать данные для вышеуказанного соединения.

При использовании транспортных протоколов удастся осуществить безопасную и надежную передачу различных данных между персональными компьютерами, что, в свою очередь, требуется при организации автоматизации на любом производстве.

2. *Сетевые протоколы* необходимы для реализации следующих услуг:

- маршрутизация и адресация информации;
- осуществление повторной передачи данных;
- организация и установление правил для взаимодействия в определенной сетевой среде.

Самые востребованные сетевые протоколы в производстве на данный момент:

- DDP (Datagram Delivery Protocol – Протокол доставки дейтаграмм);
- IP (Internet Protocol – Протокол Internet). Протокол стека TCP/IP, обеспечивающий адресную информацию и информацию о маршрутизации;
- IPX (Internetwork Packet eXchange – Межсетевой обмен пакетами) в NWLink.

3. Прикладные протоколы предназначены для реализации связи между приложениями. К самым популярным прикладным протоколам относятся:

- FTP (File Transfer Protocol – Протокол для передачи файлов) – протокол стека TCP/IP, который используют для реализации услуг по передаче файлов;
- HTTP (Hyper Text Transfer Protocol) – протокол для передачи гипертекста;
- PROFINET – промышленный стандарт для автоматизации, который является открытым. Данный промышленный протокол использует TCP/IP и IT, режим реального времени и стандарт Ethernet. PROFINET имеет модульную концепцию и структуру, с помощью которых пользователи могут осуществлять каскадирование самих функций. Данные функции отличаются в зависимости от типа обмена данными для выполнения высоких требований к скорости.

Все уровни автоматизированной системы независимы относительно друг друга. Модули и блоки, предназначенные для исполнения функций на любом уровне, можно заменить без каких-либо изменений на других уровнях. Каждый уровень отвечает за отдельные функции взаимодействия различных сетевых устройств. Уровни создают иерархическую систему с запросом, который вырабатывается на одном из этих уровней, после чего передается на исполнение ниже по иерархической лестнице. Результаты обработки запроса транслируют информацию на уровень, находящийся выше в цепи автоматизации.

На прикладном уровне от приложений, работающих на персональном компьютере, происходит запрос данных, представляющий собой сообщение, предназначенное для передачи на другой персональный компьютер.

На физическом уровне происходит передача данных по физическим линиям. Для осуществления взаимосвязей между двумя уровнями в системе организуется соединение на физическом уровне, которое называется *интерфейсом*.

С помощью интерфейсов определяется набор возможных для использования сервисов, которые предоставляются для взаимодействия соседних уровней. Обычно для обмена данными используют два контроллера или компьютера. Взаимодействие между компьютерами может быть представлено как связь между одинаковыми, равнозначными уровнями.

Промышленная сеть – это среда для передачи данных, которая включает в себя набор протоколов, предназначенных для обмена данными, а также интерфейс связи. На предприятиях используют промышленные сети вместе с устройствами, которые обслуживают реальный процесс производства.

Промышленные сети делят на следующие виды:

- *локальные сети* – расположены на ограниченной территории;
- *городские сети* – обслуживают территории крупных городов;
- *глобальные сети* – объединяют территориально удаленных пользователей на большой территории.

Промышленные сети делят на два уровня:

- *контроллерные сети* – промышленные сети этого уровня используют для управления процессом производства;
- *сенсорные сети* – применяют для опроса датчиков и управления работой исполнительных устройств.

Для лучшего понимания работы сетей необходимо также знать виды топологий локальных сетей.

Топология локальных сетей «шина» (рис. 27) предполагает подключение всех рабочих станций к общему кабелю (называемому «магистралью» или «шиной»). На концах кабеля находятся терминаторы в целях предотвращения отражения сигнала.

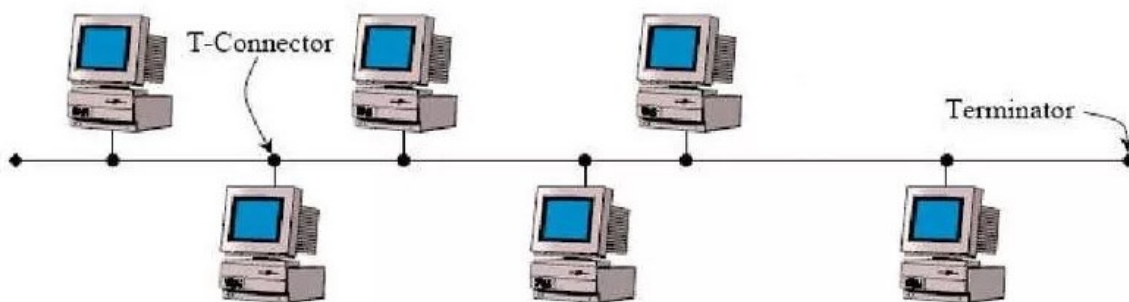


Рис. 27. Топология локальных сетей «шина»

Топология локальных сетей «звезда» (рис. 28) – схема, в которой все компьютеры сети присоединены к центральному узлу.

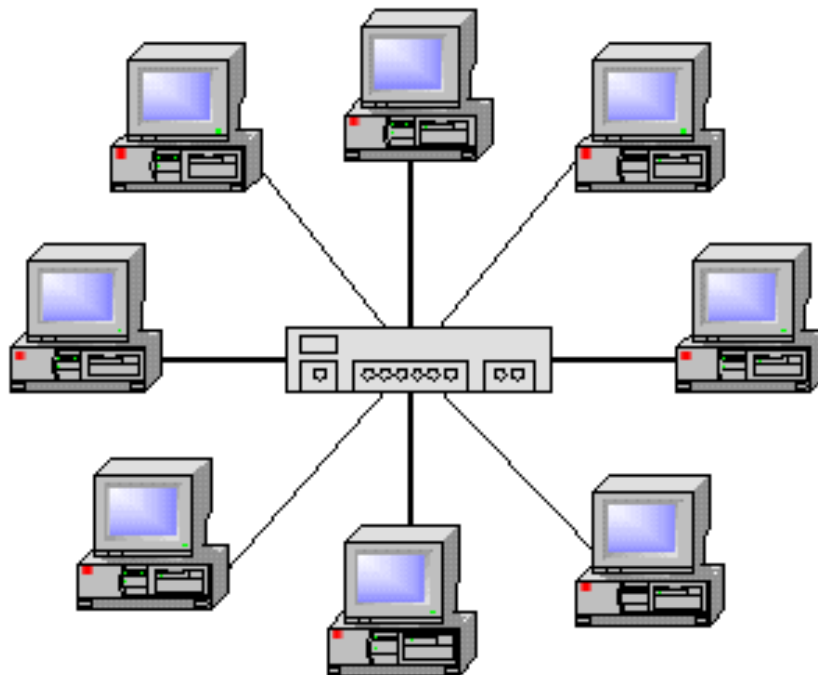


Рис. 28. Топология локальных сетей «звезда»

Топология локальных сетей «кольцо» (рис. 29) – топология, в которой каждый компьютер соединен линиями связи только с двумя другими компьютерами: от одного он только получает информацию, а второму только передает. На каждой линии связи, как и в случае «звезды», работают только один передатчик и один приемник. Это позволяет отказаться от применения внешних терминаторов.

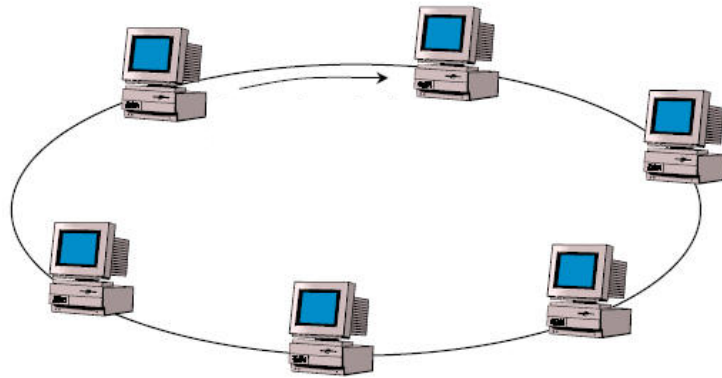


Рис. 29. Топология локальных сетей «кольцо»

Топология локальных сетей «ячеистая или полносвязная структура» (рис. 30) – сеть, в которой каждый компьютер непосредственно связан со всеми остальными. Преимуществом такой топологии является то, что она соединяет все узлы друг с другом. Однако на практике этот вид топологии не применяется, поскольку это крайне дорогой вариант построения сети.

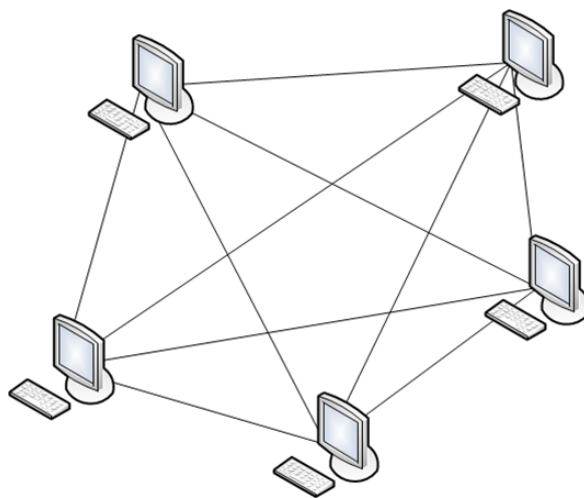


Рис. 30. Топология локальных сетей «ячеистая или полносвязная структура»

Топология локальных сетей «древовидная» (рис. 31) – иерархическая структура, в которой все компьютеры связаны друг с другом. Самый верхний узел в такой топологии известен как корневой узел, а все остальные узлы – его потомки. Существует только один путь между двумя узлами для передачи данных.

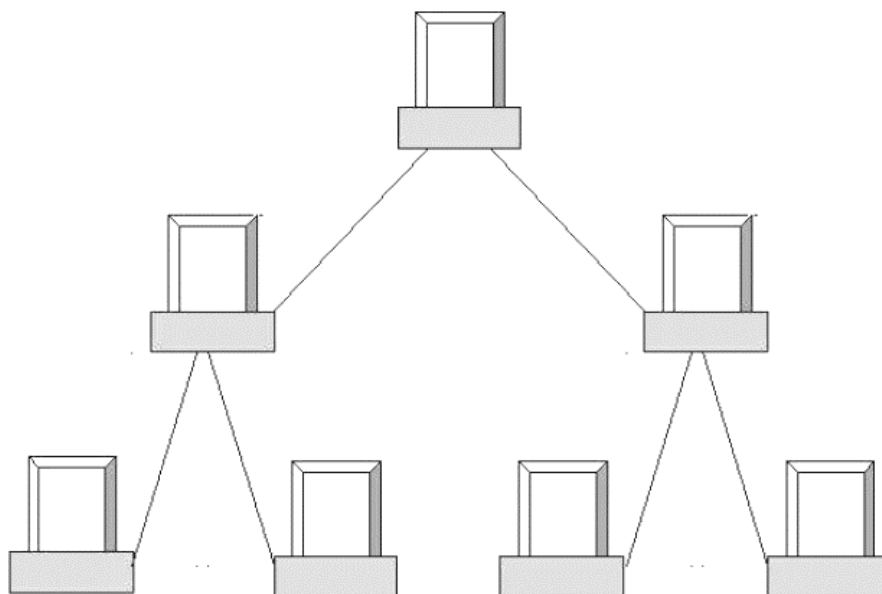


Рис. 31. Топология локальных сетей «древовидная»

Правильная топология позволяет передавать данные быстро и эффективно. Это как с дорогами: если выбрать короткий и свободный маршрут, цель будет достигнута быстрее.

Некоторые топологии обеспечивают высокую надежность сети. Например, если в ячеистой топологии (Mesh) один кабель выйдет из строя, данные смогут передаваться через другие маршруты.

Сети должны расти и изменяться вместе с потребностями пользователя. Хорошо спроектированная топология сети позволяет легко добавлять новые устройства и модернизировать оборудование.

Выбор правильной топологии может сэкономить деньги. Например, в топологии «шина» требуется меньше кабелей, чем в ячеистой топологии, что может существенно уменьшить затраты. Однако иногда лучше вложиться в более дорогую сетевую структуру, чтобы избежать проблем в будущем.

В качестве примера промышленных сетей можно привести архитектуру протокола Modbus, представленную на рис. 32 и 33.

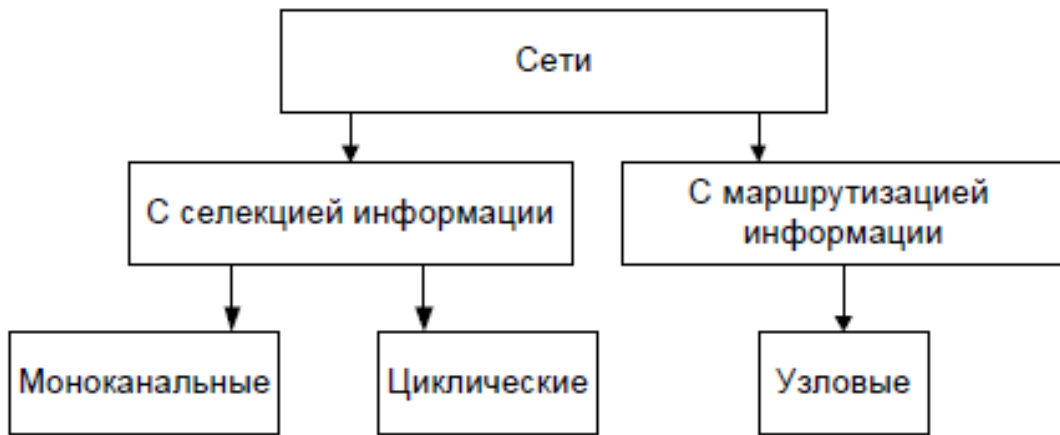


Рис. 32. Общая структура промышленных сетей

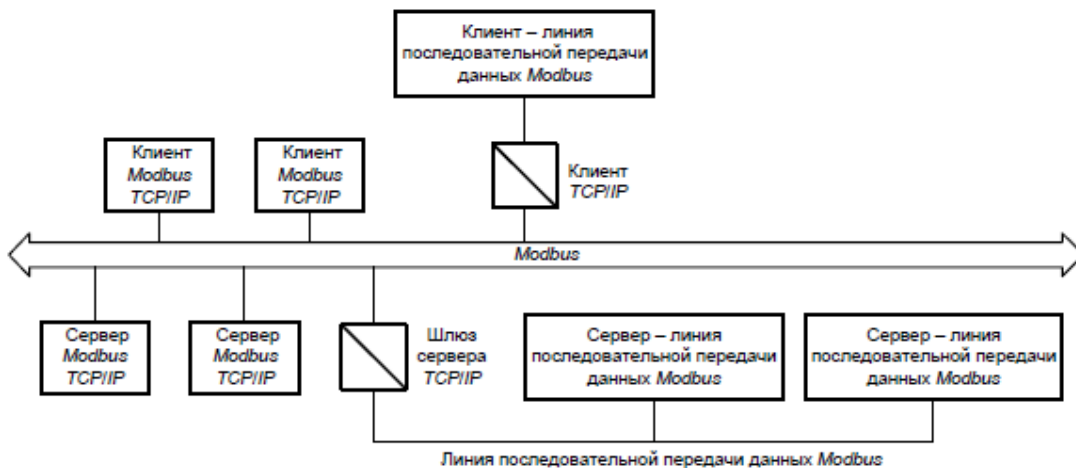


Рис. 33. Архитектура протокола Modbus

Контрольные вопросы

1. Что такое стек протоколов?
2. Назовите примеры стеков протоколов.
3. Какие существуют уровни стеков протоколов?
4. Какие существуют промышленные сети?
5. Назовите топологии локальных сетей.
6. Какие существуют сети Modbus?

5. АНАЛОГОВАЯ ОБРАБОТКА И РАЗРАБОТКА ЧМИ

Аналоговый сигнал – непрерывно меняющееся со временем значение физической величины. Значения аналогового сигнала произвольны в каждый момент времени, поэтому он может быть представлен либо как некая непрерывная функция (зависящая от времени как от переменной), либо как кусочно-непрерывная функция времени.

Аналоговым сигналом можно назвать, например, звуковой сигнал, генерируемый обмоткой электромагнитного микрофона или ламповым акустическим усилителем, поскольку такой сигнал непрерывен и его значения (напряжение или ток) сильно отличаются друг от друга в каждый момент времени.

Аналоговые величины могут иметь бесконечное множество значений в некоторых пределах. Они непрерывны и их значения не могут изменяться скачками. На рис. 34 представлены абстрактный вид аналогового сигнала и некоторые устройства, при применении которых используется данный сигнал.



Рис. 34. Аналоговый сигнал и устройства, работающие на его основе

Существует несколько типов аналоговых сигналов:

- сигналы по току (4 – 20 мА);
- сигналы по напряжению (от 0 – 10 В).

При использовании сигнала по напряжению (от 0 до 10 В) для получения информации с датчика весь диапазон последнего делится на

диапазон напряжения 0 – 10 В (рис. 35). Например, датчик температуры имеет диапазоны $-10\dots+70$ °С. Тогда при -10 °С на выходе датчика будет 0 В, а при $+70$ °С – 10 В. Все промежуточные значения находят из пропорции.

С помощью унифицированного сигнала напряжения можно не только получать данные о физических величинах, но и управлять устройствами. Например, можно привести трехходовой клапан в нужное положение, изменить скорость вращения электродвигателя через частотный преобразователь или мощность нагревателя.

Возьмем для примера электродвигатель, частотой вращения которого управляет частотный преобразователь.

Частоту вращения двигателя задает контроллер сигналом 0 – 10 В, приходящим на аналоговый вход частотника. Частота вращения двигателя может быть от 0 до 50 Гц. Тогда, если в соответствии с алгоритмом контроллер собирается раскрутить двигатель на 25 Гц, он должен подать на вход частотника 5 В.

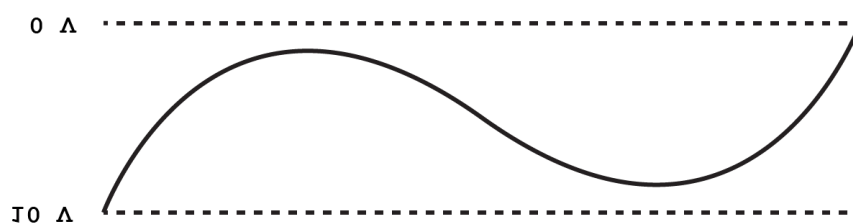


Рис. 35. Аналоговый сигнал 0 – 10 В

Управление различными устройствами с помощью токового сигнала (4 – 20 мА) ничем не отличается от управления с помощью сигнала напряжения. Только в данном случае нужен уже источник не напряжения, а тока.

Если устройство имеет управляющий вход 4 – 20 мА, то им может управлять контроллер или другое интеллектуальное устройство, имеющее соответствующий выход.

Например, мы хотим плавно открывать вентиль, имеющий электропривод со входом 4 – 20 мА. Если подать на вход сигнал тока 4 мА, тогда вентиль будет полностью закрыт, а если подать сигнал тока 20 мА – полностью открыт.

Аналоговый сигнал может иметь любое значение от 4 до 20 мА, пример диапазона сигнала представлен на рис. 36.

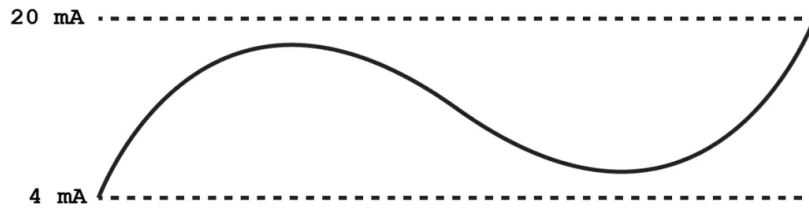


Рис. 36. Аналоговый сигнал 4 – 20 мА

Рассмотрим пример использования датчика с сигналом по току от 4мА до 20мА (рис. 37).

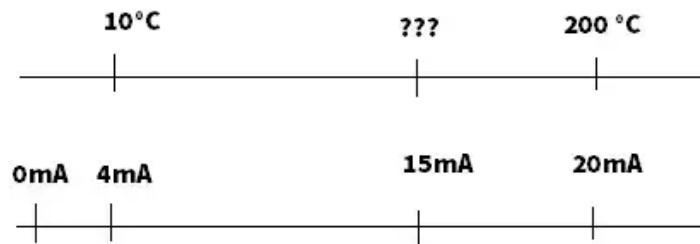


Рис. 37. Аналоговый сигнал 4 – 20 мА

- Из представленной условной схемы можно наблюдать следующее:
- диапазон сигнала по току: от 4 мА до 20 мА;
 - диапазон датчика температуры: от 10 °С до 200 °С;
 - измеренное текущее значение: 15 мА.

Для определения физической величины (значения, которое отобразит датчик) можно использовать формулу

$$P_v = \frac{P_{\max} - P_{\min}}{I_{\max} - I_{\min}} (I_{\text{value}} - I_{\min}) + P_{\min},$$

где P_{\max} – верхняя граница шкалы датчика; P_{\min} – нижняя граница шкалы датчика; I_{\max} – максимальное значение тока с датчика (20 мА); I_{\min} – минимальное значение тока с датчика (4 мА); I_{value} – измеренное значение тока с датчика.

При работе с напряжениями используют формулу преобразования напряжения U в физическую величину P_v

$$P_v = \frac{P_{\max} - P_{\min}}{U_{\max} - U_{\min}} (U_{\text{value}} - U_{\min}) + P_{\min},$$

где P_{\max} – верхняя граница шкалы датчика; P_{\min} – нижняя граница шкалы датчика; U_{\max} – максимальное значение напряжения с датчика (20 мА); U_{\min} – минимальное значение напряжения с датчика (4 мА); U_{value} – измеренное значение напряжения с датчика.

Данные, полученные с датчиков, выводят на HMI. Рассмотрим возможности создания HMI-экранов в программном обеспечении UniLogic. Для начала необходимо рассмотреть общий вид редактора экранов HMI (рис. 38).

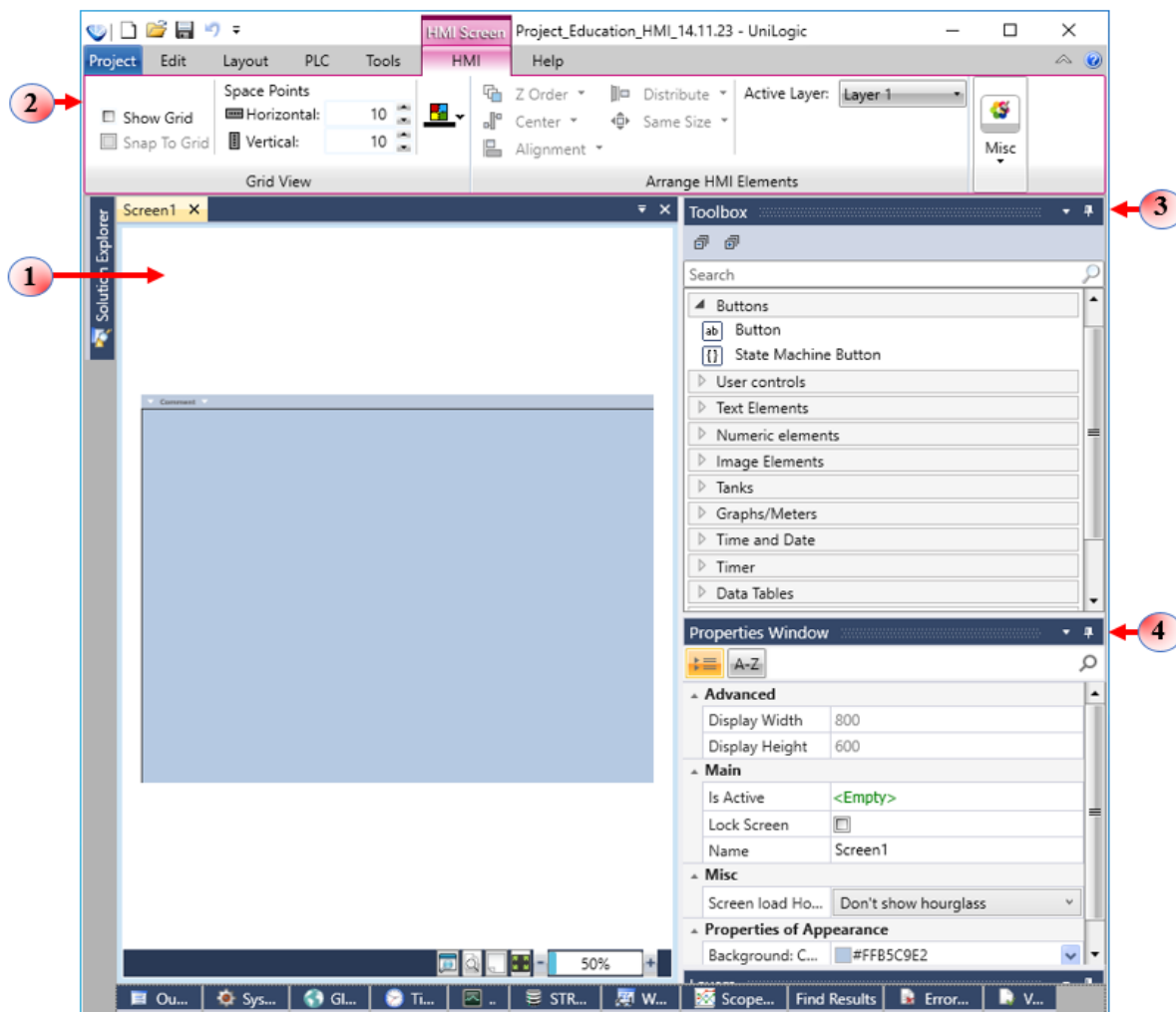


Рис. 38. Редактор HMI-экранов: 1 – редактор HMI (рабочая область для создания экранных форм HMI); 2 – лента (вкладка HMI позволяет упорядочивать элементы, экспортировать скриншоты экранов и выбирать темы); 3 – панель инструментов (Ctrl + T) (рабочая область, на которой размещаются основные элементы для создания HMI); 4 – окно свойств (рабочая область, которая позволяет просмотреть свойства выбранного элемента HMI)

В программном обеспечении UniLogic можно выбрать различные статические элементы мнемосхем. На рис. 39 изображены графические примитивы (линии, эллипсы и другие фигуры).

Также в программном обеспечении UniLogic используют библиотеки графических элементов (рис. 40), в которые входят такие элементы, как кнопки, ползунки, аппараты, механизмы, технологические объекты, переключатели, стрелки и др.



Рис. 39. Графические примитивы

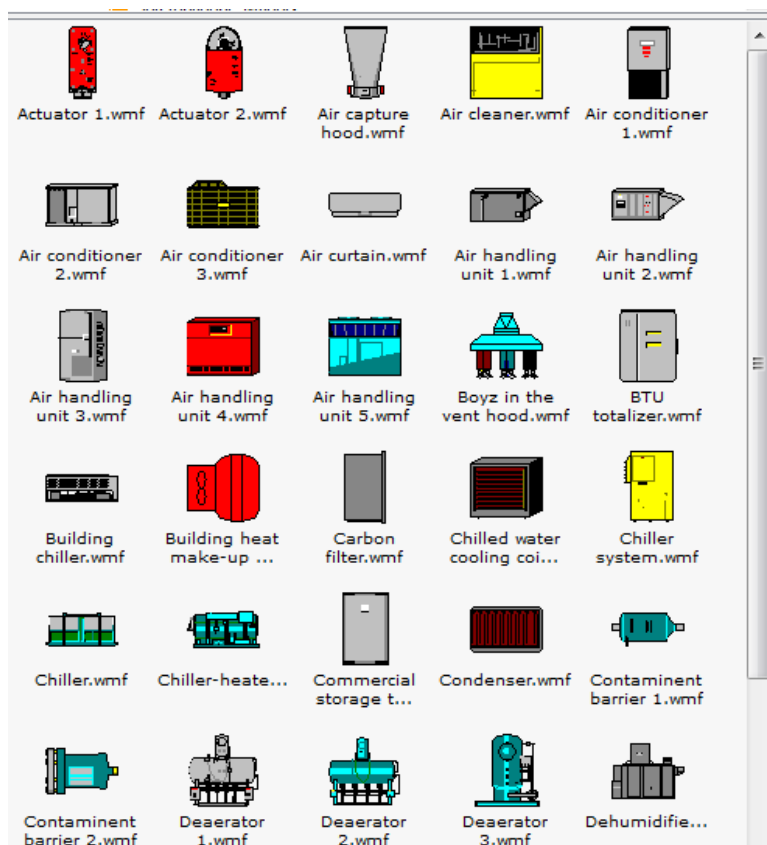
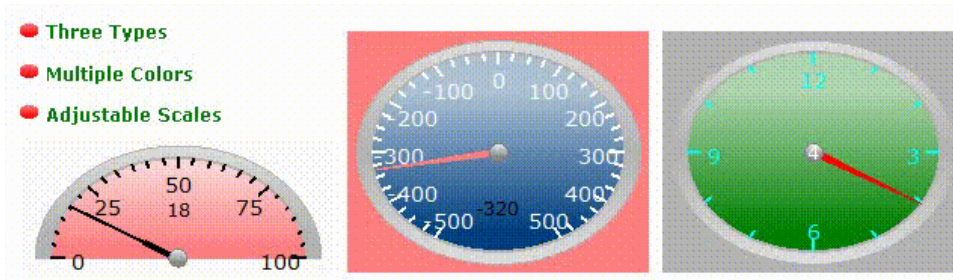


Рис. 40. Библиотеки графических элементов

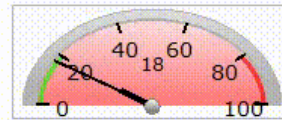
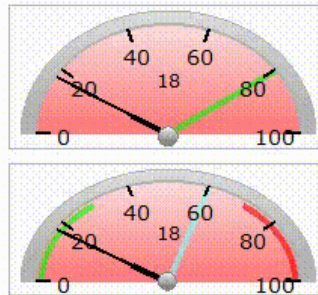
Для визуального отображения данных, полученных во время работы, предусмотрены динамические объекты, которые представляют собой диаграммы, графики, цифровые счетчики и другие элементы (рис. 41).



There is

- Target Line
- Low Range
- High Range

and it could be variables



Target

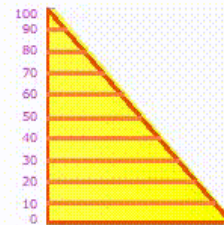
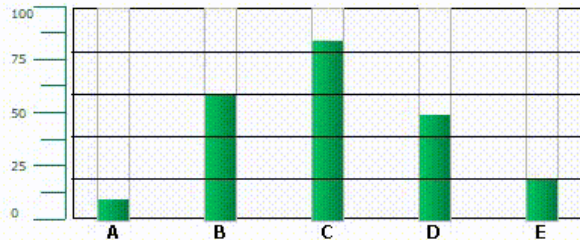
60

Low

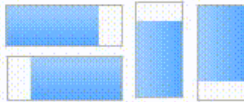
30

High

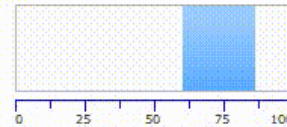
70



• Multiple Direction



• Show Differential



Standard Value

60

Devition Range

15

• Target

60

• Low

45

• High

75

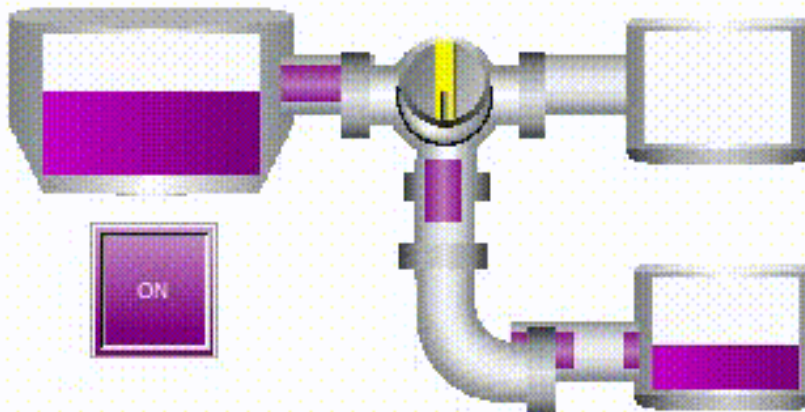


Рис. 41. Динамические объекты

Существует несколько принципов создания интерфейсов:

- лаконичность и наглядность;
- максимальная линейность;
- визуальный акцент на элементы контроля и управления;
- учет человеческого фактора.

Также для создания интерфейсов используют различные способы кодирования сигналов, в основном визуальные (световые) (табл. 9) и звуковые (табл. 10) сигналы.

Таблица 9

Визуальные сигналы

Цвет	Смысловое значение		
	Безопасность людей или оборудования	Состояние процесса	Состояние оборудования
КРАСНЫЙ	Опасность	Критическое состояние	Неисправность
ЖЕЛТЫЙ	Внимание	Переходное (изменение условий или состояние, предшествующее изменению условий)	Переходное (изменение условий или состояние, предшествующее изменению условий)
ЗЕЛЕНый	Безопасность	Нормальное	Нормальное
СИНИЙ	Специальное (может иметь любое значение, кроме функционального, для красного, желтого и зеленого цветов)		
БЕЛЫЙ	Не имеет специального значения		
СЕРЫЙ	Не имеет специального значения		

Таблица 10

Звуковые сигналы

Вид звука	Смысловое значение		
	Безопасность людей или оборудования	Состояние процесса	Состояние оборудования
Протяжный, резкий, усиливающийся	Опасность	Критическое	Неисправное
Прерывистый с постоянным интервалом	Внимание	Переходное	Переходное
Непрерывный с постоянным уровнем	Безопасность	Нормальное	Нормальное
Чередующиеся звуки	Специальное		
Другие звуки	Не имеет специального значения		

Контрольные вопросы

1. Что такое аналоговый сигнал?
2. В каких приборах используется аналоговый сигнал?
3. Расскажите про аналоговый сигнал 0 – 10 В.
4. Расскажите про аналоговый сигнал 4 – 20 мА.
6. Расскажите про статические элементы мнемосхем на примере UniLogic.
7. Расскажите про динамические объекты на примере UniLogic.

6. АРХИТЕКТУРА ПЛК. ПОДКЛЮЧЕНИЕ ПЛК. ПОДКЛЮЧЕНИЕ К ПЛК. БЕЗОПАСНОСТЬ В ПЛК

Программируемый логический контроллер предназначен для управления технологическими процессами, однако без полноценного технического комплекса это управление невозможно. В ПЛК входят контрольно-измерительные приборы, исполнительные устройства, модули ввода и вывода аналоговых и цифровых сигналов, преобразователи сигналов, реле, контакторы, блоки питания и другие элементы автоматики. Рассмотрим основные элементы автоматики и архитектуры систем управления.

В настоящий момент существует большое количество контрольно-измерительных приборов, таких как манометры, термометры, датчики положения и многие другие. Все контрольно-измерительные приборы работают по принципу преобразования какой-либо физической величины в электрический сигнал. В том случае, если сигналы с контрольно-измерительных приборов не согласуются со входами аналого-цифровых преобразователей или не являются стандартными (например, выходное значение датчика имеет напряжение 0 – 100 мВ, а входная величина преобразователя имеет напряжение 0 – 10 В), применяют измерительный преобразователь. Данное устройство позволяет нормализовать значение сигнала, поступающего с контрольно-измерительного прибора, посредством стабилизации величины напряжения до требуемой величины, а также позволяет компенсировать значение погрешности, добиться линейности сигнала. Стандартная практика – совмещение преобразователя для контрольно-измерительного прибора с модулем аналогового ввода.

Измерительные преобразователи зачастую имеют встроенный цифро-аналоговый или аналого-цифровой преобразователь. Для линеаризации характеристик контрольно-измерительного прибора и сокращения погрешностей используется микропроцессор. На данный момент наибольшую популярность имеют цифровые контрольно-измерительные приборы, так как они включают в себя измерительный преобразователь, аналого-цифровой преобразователь и первичный преобразователь физического сигнала в электрический.

Модули аналогового ввода предназначены для передачи аналогового сигнала в компьютер оператора или программируемый логический контроллер, они бывают как универсальные, так и специализированные.

Главный элемент для систем автоматизации – компьютер оператора или ПЛК. Для управления и отображения информации, принимаемой контроллером, используют панели управления. На главный элемент системы автоматизации приходят сигналы с контрольно-измерительных приборов, также с его помощью осуществляется передача информации на устройства ввода (далее – на исполнительный механизм) посредством исполнения заранее записанной и установленной в памяти устройства программы. Для взаимодействия между персональным компьютером или ПЛК в основном используются интерфейсы связи RS-485, RS-232, Ethernet, USB и CAN, однако существуют модули и блоки с портами другого типа, для них подбирают специализированное периферийное устройство. В некоторых случаях модули ввода и вывода реализованы в виде плат, встроенных в компьютер, а для коммуникации используются разъемы шин ISA и PCI, это позволяет устанавливать скорость передачи информации между устройствами выше чем 10 Мбит/с. При этом возникает проблема, связанная с высоким уровнем электромагнитных наводок и конструктивными ограничениями в виде количества каналов, возможных для подключения.

Все вышеперечисленные элементы можно представить в виде структуры автоматизированной системы (рис. 42).

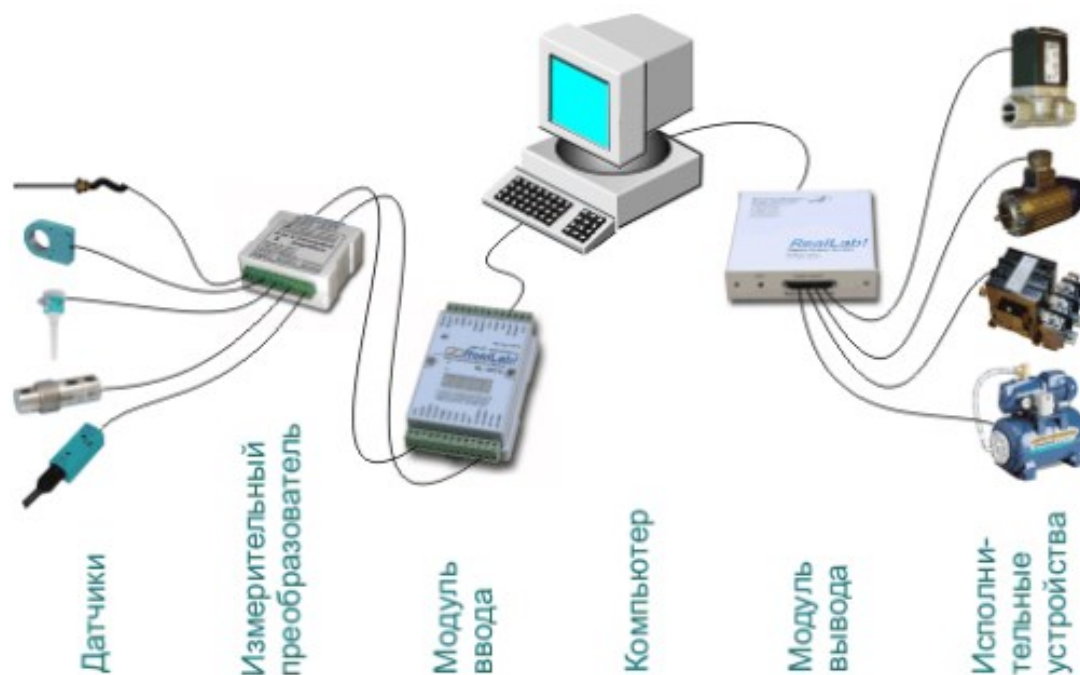


Рис. 42. Структура автоматизированной системы

Под *архитектурой автоматизированной системы управления* на физическом уровне подразумевают модель, состоящую из элементов автоматики, контрольно-измерительных приборов и исполнительных устройств, входящих в систему, а также отображение взаимосвязей между данными компонентами. Архитектура представляет собой взгляд на автоматизированную систему управления в целом, позволяя принять огромное количество конкретных технических решений, подобрать различные компоненты для непосредственной реализации проекта автоматизации.

Для качественного проектирования архитектуры автоматизированной системы управления проектировщик должен предметно знать, какие принципы, законы и функции применяют при использовании технологического объекта управления. Помимо этого необходимо иметь представление о контрольно-измерительных приборах, исполнительных устройствах, автоматике, программных средах и средствах, которые могут быть применены для создания будущей автоматизированной системы управления.

При проектировании архитектуры автоматизированной системы управления обязательно закладываются следующие свойства:

- 1) низкий уровень связанности элементов архитектуры между собой; это необходимо для того, чтобы избежать лишних потоков информации и замыканий контуров автоматического регулирования;
- 2) тестируемость – свойство, необходимое для реализации верного функционирования;
- 3) диагностируемость – свойство, необходимое для обнаружения неисправностей в системе;
- 4) пригодность к ремонту;
- 5) надежность системы;
- 6) безопасность – система должна соответствовать стандартам безопасности ГОСТов;
- 7) простота в эксплуатации и обслуживании системы – не должно быть высоких требований к квалификации персонала, работающего с данной системой;
- 8) защищенность системы;
- 9) экономическая эффективность системы во время ее функционирования.

Архитектуры автоматизированных систем управления отличаются в зависимости от решаемой задачи автоматизации. Одни из таких задач:

1) **мониторинг** – продолжительное измерение сигналов, находящихся внутри автоматизированной системы управления, и контроль потоков информации с последующим сохранением полученных данных в архиве;

2) **автоматическое управление** может быть представлено в нескольких вариантах: система с односторонним управлением без коррекции в режиме реального времени, автоматизированная система управления с наличием контура обратной связи, система управления с адаптивным управлением.

3) **диспетчерское управление** – система автоматизированного управления, подразумевающая наличие оператора или диспетчера, отвечающего за технологический объект управления, при этом взаимодействие специалиста с системой происходит за счет использования ЧМИ.

4) **обеспечение безопасной работы предприятия.**

Распределенные системы применяют в том случае, если автоматизированная система управления и технологический участок, на котором она используется, начинают расширяться. Смысл данной системы заключается в том, что она включает в себя большое количество контроллеров и прочих элементов автоматики, размещенных на разных участках производственной зоны. С использованием распределенных систем автоматизации удастся добиться эффекта, при котором алгоритм работы автоматизированной системы управления становится идентичным работе и структуре технологического объекта автоматизации, а нагрузка на систему распределяется между всеми программируемыми логическими контроллерами, т. е. каждый контроллер связан со своей группой элементов автоматики, контрольно-измерительными приборами и исполнительными устройствами; это позволяет обслуживать каждую часть технологического объекта управления по отдельности.

Распределенная система – система, состоящая из множества устройств, размещенных на рабочей территории производства, каждое из которых не зависит от другого, но взаимодействует с ним для выполнения определенных или общих задач.

Максимальные преимущества распределенной системы достигаются, когда контроллеры работают автономно, а обмен информацией между ними сведен до минимума.

Распределенная система имеет следующие характеристики, отличающие ее от сосредоточенной:

- большее быстродействие благодаря распределению задач между параллельно работающими процессорами;
- повышенную надежность (отказ одного из контроллеров не влияет на работоспособность других);
- большую устойчивость к сбоям;
- более простое наращивание или реконфигурирование системы;
- упрощенную процедуру модернизации;
- большую простоту проектирования, настройки, диагностики и обслуживания благодаря соответствию архитектуры системы архитектуре объекта управления, а также относительной простоте каждого модуля системы;
- улучшенную помехоустойчивость и точность благодаря уменьшению длины линий передачи аналоговых сигналов от датчиков к устройствам ввода;
- меньший объем кабельной продукции, пониженные требования к кабелю и более низкую его стоимость;
- меньшие расходы на монтаж и обслуживание кабельного хозяйства.

На рис. 43 представлен общий вид абстрактной распределенной системы автоматизации.

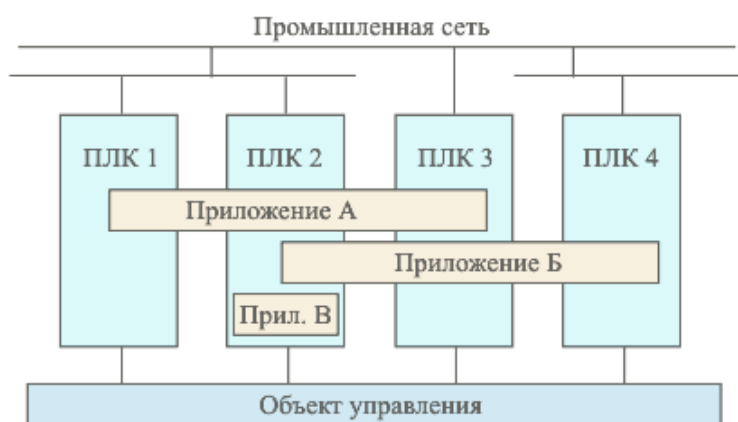


Рис. 43. Абстрактная распределенная система автоматизации

Распределенная система автоматизированного управления позволяет снизить требования к операционным системам реального времени, так как выполняемые операции – параллельно распределенные между ПЛК, параллельно взаимодействующие между собой с учетом того, что на каждом контроллере установлена своя операционная система.

Для эффективного проектирования распределенных систем автоматизации необходимы строгие методы их описания. Требуется также обеспечить совместимость и взаимозаменяемость всех устройств, входящих в систему и выпускаемых разными производителями. Для этих целей был разработан международный стандарт МЭК 61499 «Функциональные блоки для промышленных систем управления».

Особенность функциональных блоков стандарта МЭК 61499 в том, что они учитывают традиционное инициирование выполнения алгоритма не только с помощью тактирования или временного расписания, но и по признаку наступления некоторых событий (событийное управление). Событийное управление используют чаще, чем тактирование; последнее можно рассматривать как его частный случай, заключающийся в периодическом появлении одного и того же события (сигнала тактирования).

Контрольные вопросы

1. Опишите взаимодействие между ПЛК и контрольно-измерительными приборами.
2. Расскажите про интерфейсы для компьютера оператора.
3. Что представляет собой структура автоматизированной системы?
4. Какие требования предъявляют к автоматизированной системе?
5. Каковы задачи автоматизированных систем?
6. Что такое распределительные системы?

ПРАКТИКУМ

Лабораторная работа № 1

ОСНОВЫ СХЕМОТЕХНИКИ

Цель работы: изучить виды схем и условно-графические изображения.

Задания:

- 1) ознакомиться с правилами техники безопасности;
- 2) изучить ГОСТ 2.701-2008;
- 3) изучить условно-графические обозначения.

Ход работы

1. Рассмотрим схемы:

- 1) Электрическая схема (рис. 44);

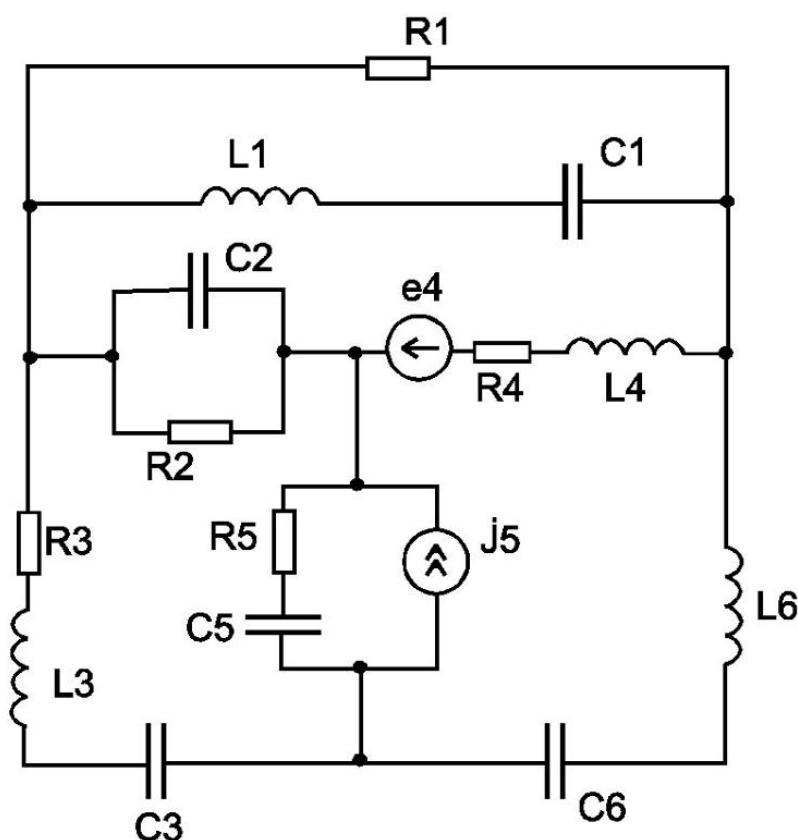


Рис. 44. Электрическая схема

2) структурная схема (рис. 45);



Рис. 45. Структурная схема

3) функциональная схема (рис. 46);

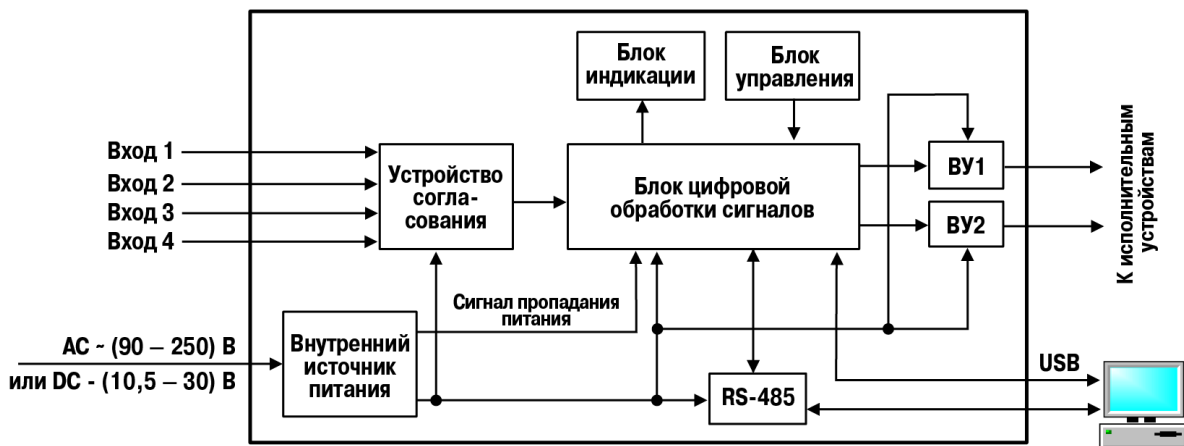


Рис. 46. Функциональная схема

4) соединительная схема (монтажная) (рис. 47);

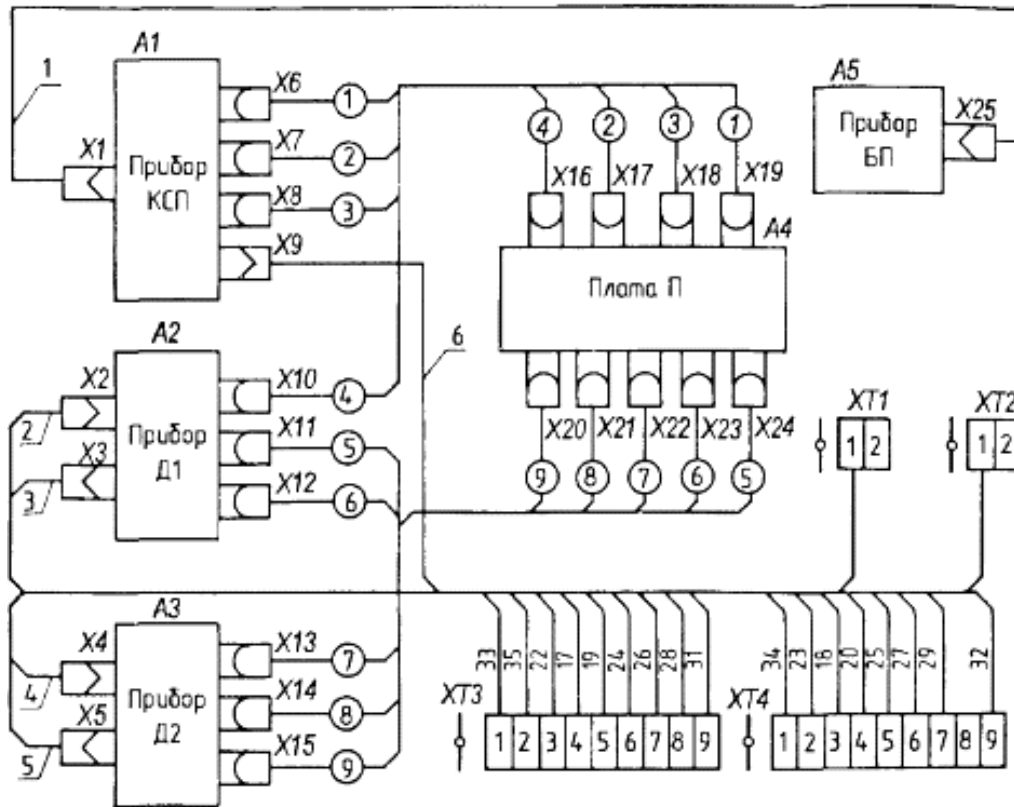


Рис. 47. Схема соединений

2. Изучим условно графические обозначения в электрических схемах:

1) разъемы, клеммы, контакты (табл. 11);

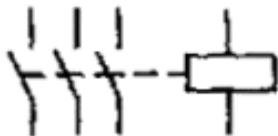

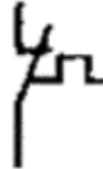


Таблица 11

Условно-графические обозначения разъемов, клемм, контактов

Наименование		Условно-графическое обозначение	
Контакт замыкающий выключателя			
Однополюсный	Трехполюсный		

Наименование	Условно-графическое обозначение
Контакт замыкающий выключателя трехполюсный с автоматическим срабатыванием максимального тока	
Контакт замыкающий нажимного кнопочного выключателя без самовозврата, с размыканием и возвратом элемента управления автоматический	
Контакт замыкающий нажимного кнопочного выключателя без самовозврата, с размыканием и возвратом элемента управления посредством вторичного нажатия кнопки	
Контакт замыкающий нажимного кнопочного выключателя без самовозврата, с размыканием и возвратом элемента управления посредством вытягивания кнопки	
Контакт замыкающий нажимного кнопочного выключателя без самовозврата, с размыканием и возвратом элемента управления посредством отдельного привода (пример нажатия кнопки – сброс)	
Разъединитель трехполюсный	
Выключатель-разъединитель трехполюсный	
Выключатель ручной	


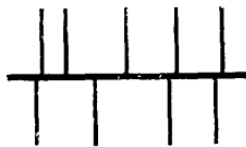
Окончание табл. 11

Наименование	Условно-графическое обозначение
Выключатель электромагнитный (реле)	
Выключатель концевой с двумя отдельными цепями	
Выключатель термический саморегулирующий	
Выключатель инерционный	
Переключатель ртутный трехконечный	



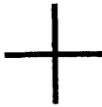

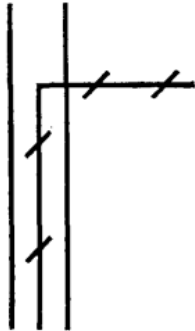
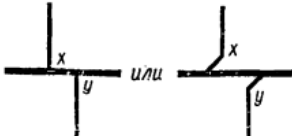
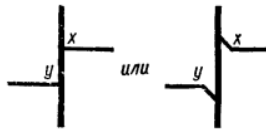


2) провода и шины (табл. 12);


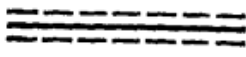




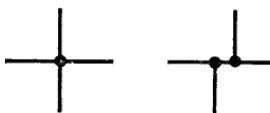
Таблица 12

Условно-графические обозначения проводов и шин

Наименование	Условно-графическое обозначение
Линия электрической связи (провод, кабель, шина)	
Графическое слияние линий электрической связи в линию групповой связи	

Продолжение табл. 12

Наименование	Условно-графическое обозначение
Графическое разветвление (слияние) линий групповой связи	
Графический излом линии групповой связи	
Графическое пересечение линий групповой связи	
Графическое пересечение линии групповой связи с линией электрической связи	
Если при выполнении схем автоматическим способом линии групповой связи выполняют не утолщенными, то для графического отделения этих линий от пересекающихся с ними или параллельных им линий электрической связи на линию групповой связи наносят наклонные штрихи	
Обозначение линий электрической связи, графически сливаемых и расположенных вертикально	
Обозначение линий электрической связи, графически сливаемых и расположенных горизонтально	
Линия экранирования	
Экранирование группы элементов	

Наименование	Условно-графическое обозначение
Экранирование группы линий электрической связи	
Линия электрической связи экранированная	
Обрыв линии электрической связи	
Заземление	
Корпус (машины, аппарата, прибора)	
Линии электрической связи с одним ответвлением	
Линии электрической связи с двумя ответвлениями	

3) аккумуляторы и батареи (рис. 48 – 50).



Рис. 48. Аккумулятор



Рис. 49. Батарея аккумуляторов



Рис. 50. Заземления

2. Силовые элементы:
 1) Насосы и моторы (табл. 13);



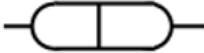
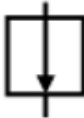

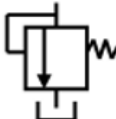
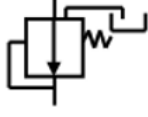


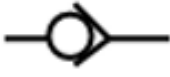
Таблица 13

Условно-графические обозначения насосов и моторов

Наименование	Условно-графическое обозначение
Насос с регулируемой производительностью с постоянным направлением потока	
Насос с регулируемой производительностью с реверсивным потоком	
Компрессор	
Вакуум-насос	
Гидромотор	
Гидромотор нерегулируемый с постоянным направлением потока	

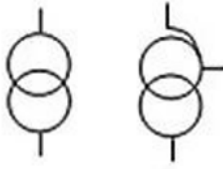
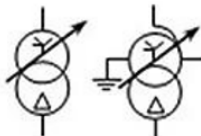
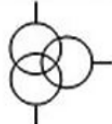
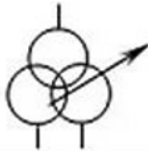





- 2) гидравлические элементы (табл. 14);

Условно-графические обозначения гидравлических элементов

Наименование	Условно-графическое обозначение
Бак под атмосферным давлением	
Фильтр для жидкости	
Гидравлический аккумулятор	
Регулирующий элемент нормально открытый	
Регулирующий элемент нормально закрытый	
Клапан предохранительный	
Клапан редукционный	
Дроссель регулируемый	
Гидрозамок	
Обратный клапан	





3) Трансформаторы (табл. 15);

Условно-графические обозначения трансформаторов

Наименование	Условно-графическое обозначение
<p>Трансформатор (автотрансформатор) силовой. Общее обозначение. <i>Примечание.</i> Внутри окружности допускается размещение квалифицирующих символов и дополнительной информации. Допускается увеличение диаметра окружности</p>	
<p>Трансформатор и автотрансформатор с регулированием напряжения под нагрузкой (РПН) с указанием схемы соединений обмоток</p>	
<p>Трансформатор силовой трехобмоточный. Трансформатор собственных нужд основного напряжения</p>	
<p>Трансформатор силовой двухобмоточный с расщеплением обмотки низшего напряжения на две, с РПН</p>	
<p>Обмотка (одной фазы) трансформатора, дросселя. Начало обмотки указывается точкой</p>	
<p>Трансформатор напряжения</p>	
<p>Два однофазных трансформатора напряжения, соединенных в открытый треугольник</p>	
<p>Трансформатор напряжения трехфазный, трехобмоточный. Трансформатор напряжения обходной системы шин</p>	
<p>Трансформатор тока измерительный</p>	

4) выключатель, переключатель (табл. 16)

Условно-графические обозначения выключателей, переключателей

Наименование	Условно-графическое обозначение
Включатель путевой однополюсный	
Включатель кнопочный нажимной с замыкающим контактом	
Включатель кнопочный нажимной с размыкающим контактом	
Контакт электротеплового реле	
Выключатель трехполюсный с автоматическим возвратом	
Контакт для коммутации силовой цепи (контактора, пускателя) замыкающий	

Содержание работы:

- 1) цель работы;
- 2) задания;
- 3) ход работы;
- 4) выводы;
- 5) средства, используемые при выполнении лабораторной работы.

Средства, используемые при выполнении лабораторной работы:

- 1) методические указания к выполнению лабораторных работ;
- 2) данные, предоставленные преподавателем во время занятия;
- 3) при проведении анализа допускается использование глобальной сети Интернет.

Лабораторная работа № 2

ЗНАКОМСТВО СО СРЕДОЙ ПРОГРАММИРОВАНИЯ ПЛК UNILOGIC. ВВЕДЕНИЕ В РАБОТУ ПЛК

Цель работы: освоить среду программирования ПЛК UniLogic.

Задания:

- 1) запуск среды UniLogic и создание нового решения;
- 2) настройка аппаратных средств;
- 3) настройка связи с ПЛК;
- 4) разработка программы на языке программирования LD.

Ход работы

1. Запуск среды UniLogic и создание нового решения.

Запускаем программу с помощью ярлыка на рабочем столе (рис. 51).

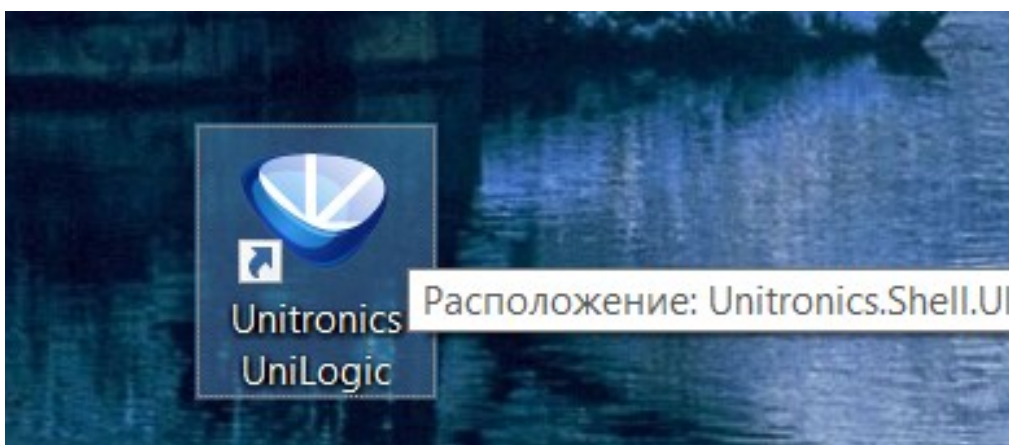


Рис. 51. Запуск программы

Для создания нового решения во вкладке «Проект» нажимаем «Новый» (рис. 52).

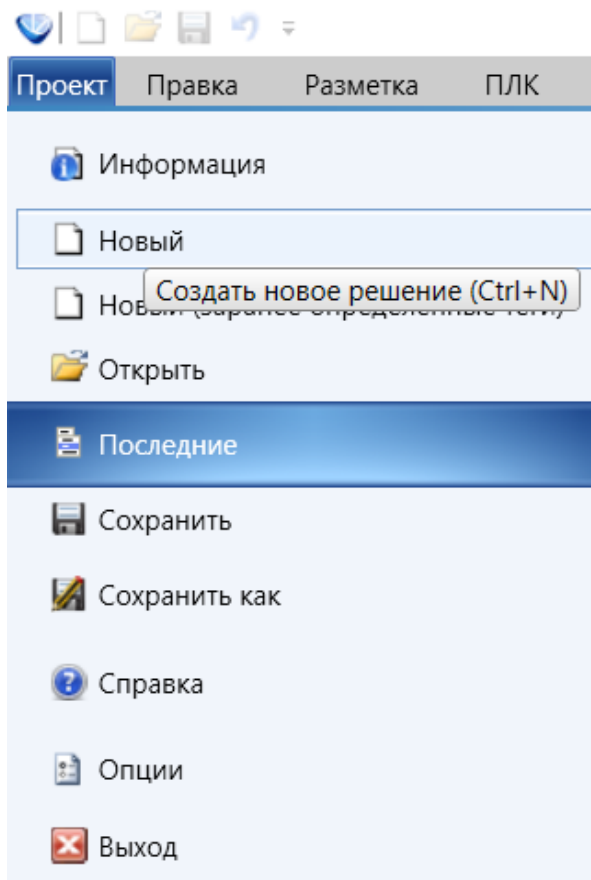


Рис. 52. Создание нового проекта

Далее указываем имя файла проекта и его местоположение (рис. 53). Путь можно указать через кнопку «Обзор». Задаем имя файла – «Практика 2», местоположение – C:\Users\user\Documents.

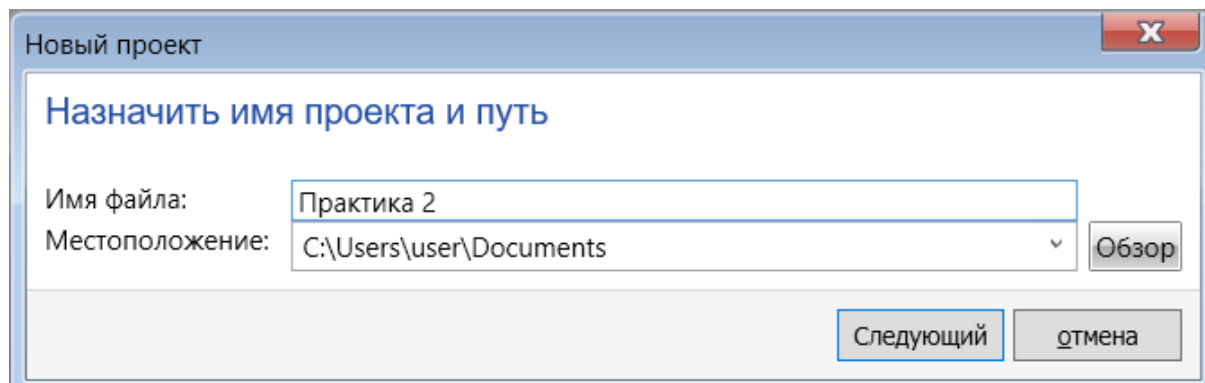


Рис. 53. Указываем имя файла проекта и его местоположение

Выбираем модель контроллера Standard USC-x5-B1 (рис. 54).

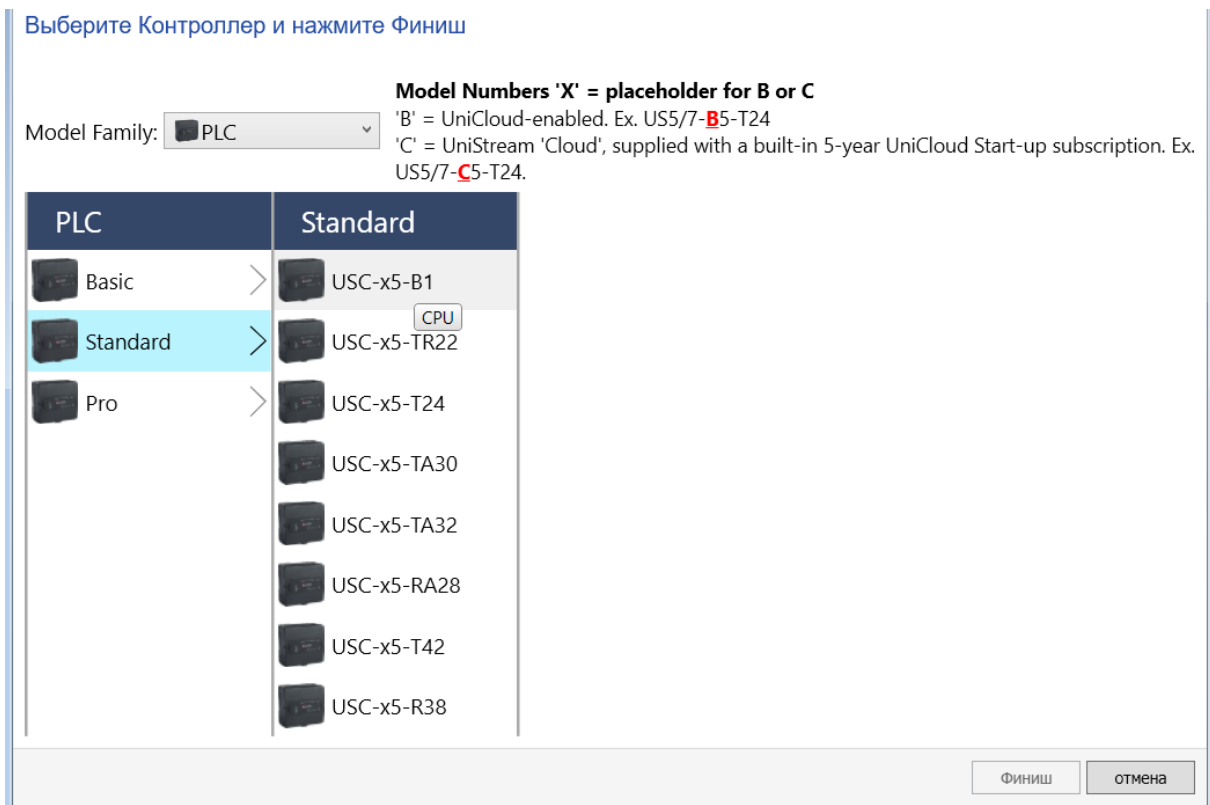


Рис. 54. Выбор модели контроллера

Выбрав модель контроллера, подбираем к нему модули ввода и вывода. Для этого нажимаем в проводнике решений на локальные I/O- и COM-модули и в панели инструментов выбираем модули UID-W1616T-0 и UID-0800N (рис. 55).



Рис. 55. Выбор модулей ввода и вывода

2. Настройка аппаратных средств.

Выбрав модули, производим их настройку: нажимаем ЛКМ на UID-W1616T-0 – откроется окно свойств (рис. 56); в нем выбираем 8 входов (0 – 7) и 8 входов (8 – 15).



Рис. 56. Окно свойств и настройка модуля UID-W1616T-0

После настраиваем модуль UID-0800N; необходимо выбрать на всех входах с AI8 вход 0 по AI8 вход 7, тип электричества от 0 до 20 мА (рис. 57).

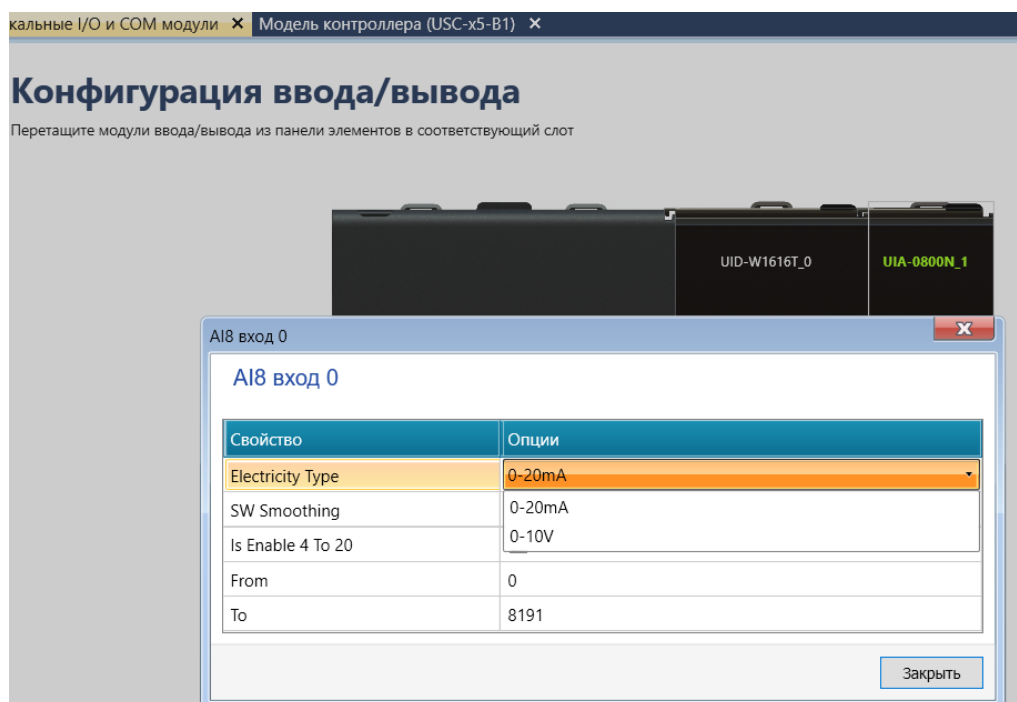


Рис. 57. Настройка модуля UID-0800N

3. Настройка связи с ПЛК.

Подключение к ПЛК с помощью ПК осуществляем следующим образом: в проводнике «Решение» нажимаем ЛКМ на Panel Ethernet (рис. 58).

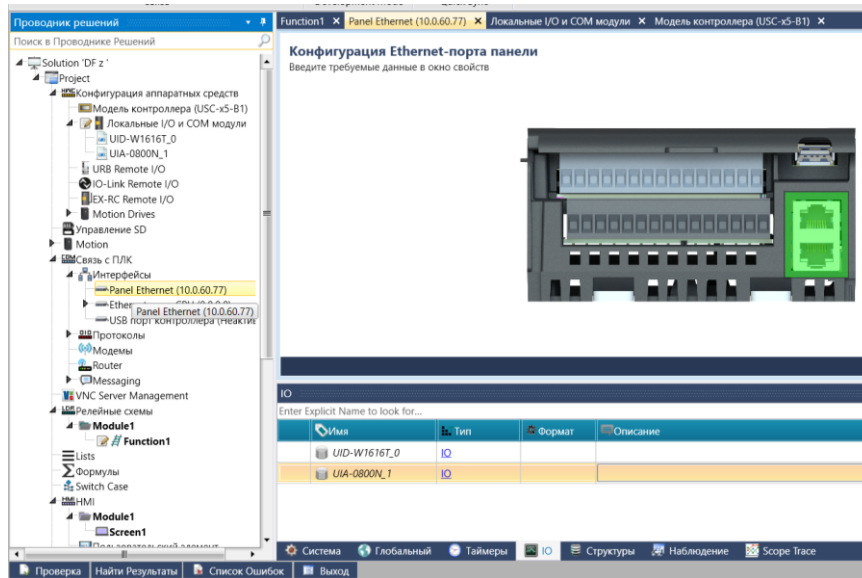


Рис. 58. Вкладка Panel Ethernet

Открыв данную вкладку на правой стороне, в строке «IP-адрес» вводим «10.0.60.77», в строке «Маска подсети» всё оставляем без изменений и в строке «Шлюз» вводим «10.0.60.1». В верхнем левом углу нажимаем ЛКМ на связь между ПК и ПЛК. Появляется окно с выбором подключения ПЛК по Ethernet или USB. Выбираем подключение по Ethernet и вводим IP-адрес 10.0.60.77. (рис. 59). Подключение к ПЛК выполнено.

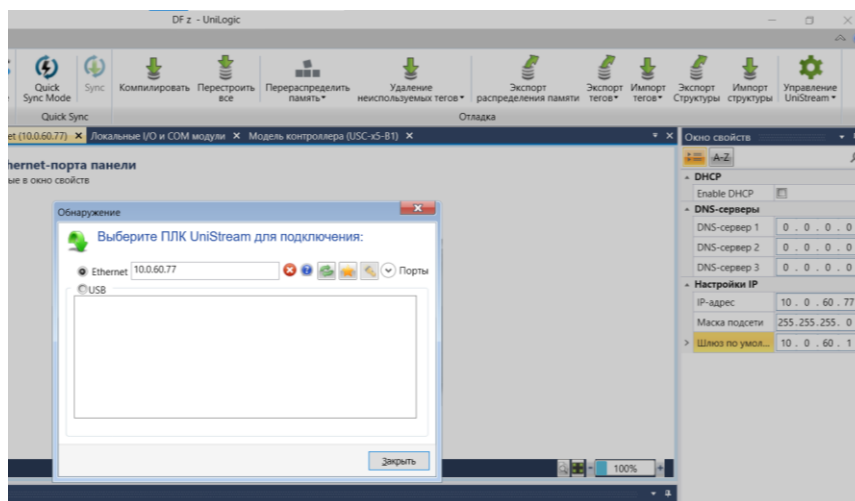


Рис. 59. Ввод IP-адреса контроллера

4. Разработка программы на языке программирования LD.

Для начала разработки программы открываем в проводнике решений вкладку Function1 (рис. 60).

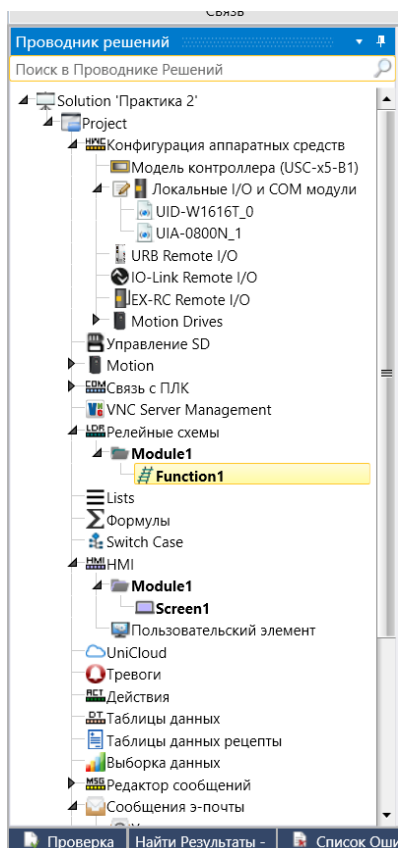


Рис. 60. Вкладка Function1

Для первого звена добавляем комментарий. Для этого нажимаем ПКМ на первом звене и выбираем во всплывающем окне Open Rung Comment (рис. 61), называем его «Первая программа».

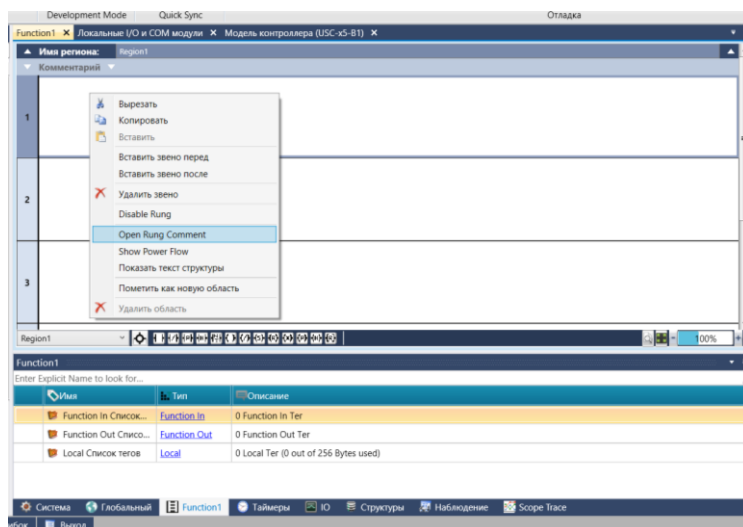


Рис. 61. Комментарий

Для создания программы используем логические элементы, представленные на панели инструментов (рис. 62).

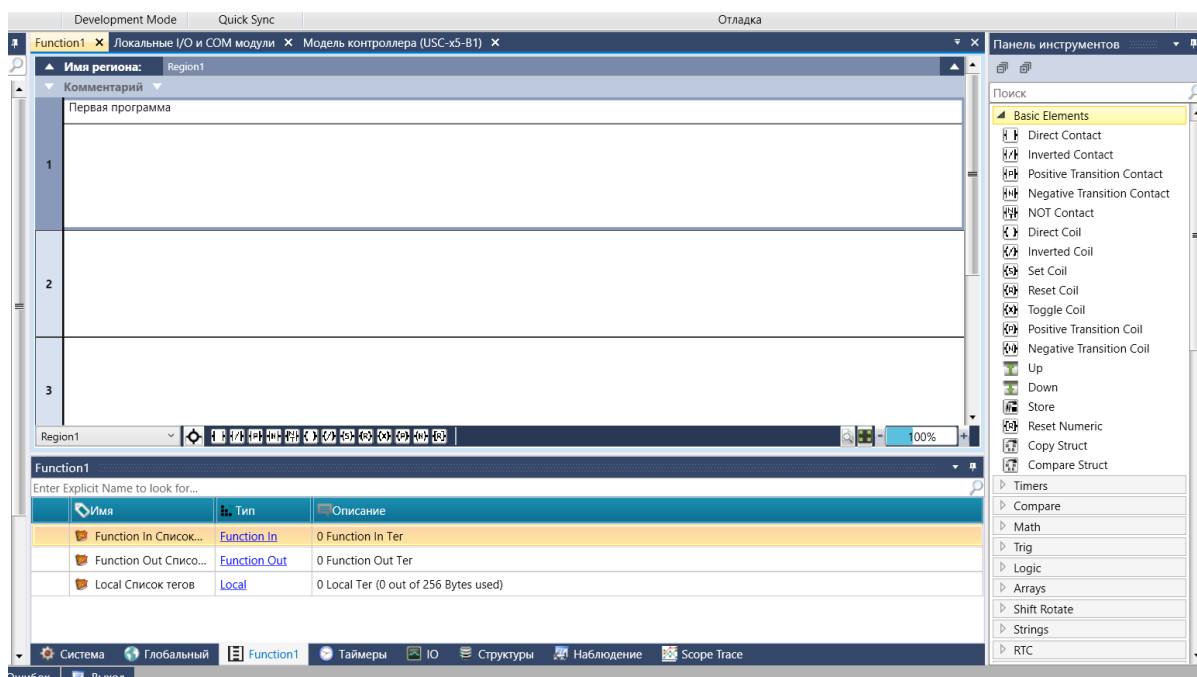


Рис. 62. Логические элементы

Первая программа будет выполнять функции включения прямой катушки с блокировкой и отключения катушки. Для этого переносим в первое звено инверсный контакт (нормально закрытый контакт) (рис. 63) и называем его Stop.

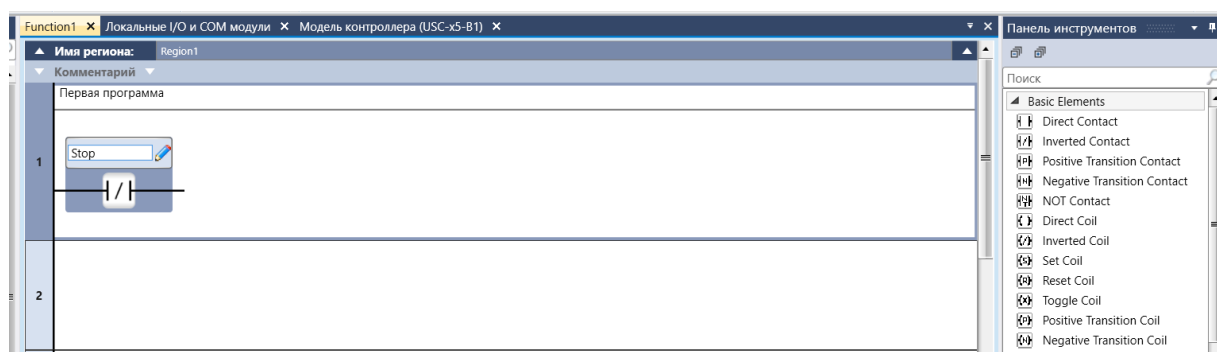


Рис. 63. Добавление инверсного контакта

Красная волнистая линия говорит о том, что тег не настроен. Для его настройки нажимаем ЛКМ на карандаш возле названия тега (рис. 64).

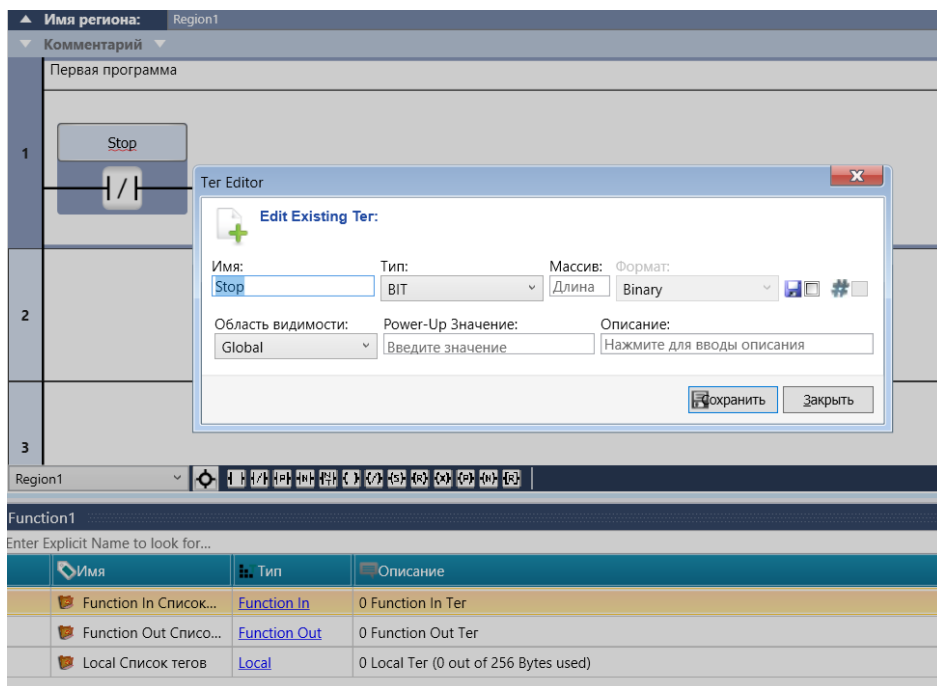


Рис. 64. Настройка тега

В открывшемся окне тип, массив, формат, область применения и power-up-значение оставляем без изменения, в описании вводим: «Кнопка стоп», затем нажимаем «Сохранить», чтобы завершить настройку тега. К инверсному контакту присоединяем прямой контакт и называем его Start, а в описании вводим: «Кнопка старт» (рис. 65).

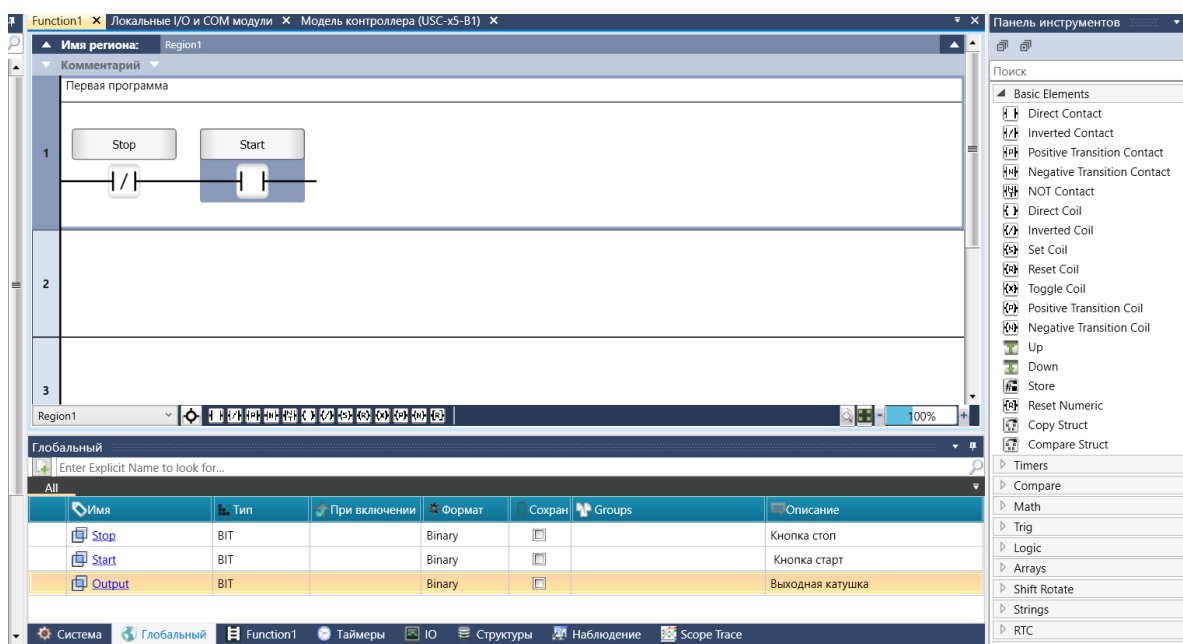


Рис. 65. Добавление прямого контакта

После добавляем к прямому контакту катушку и называем ее Output, а в описании вводим: «Выходная катушка» (рис. 66).

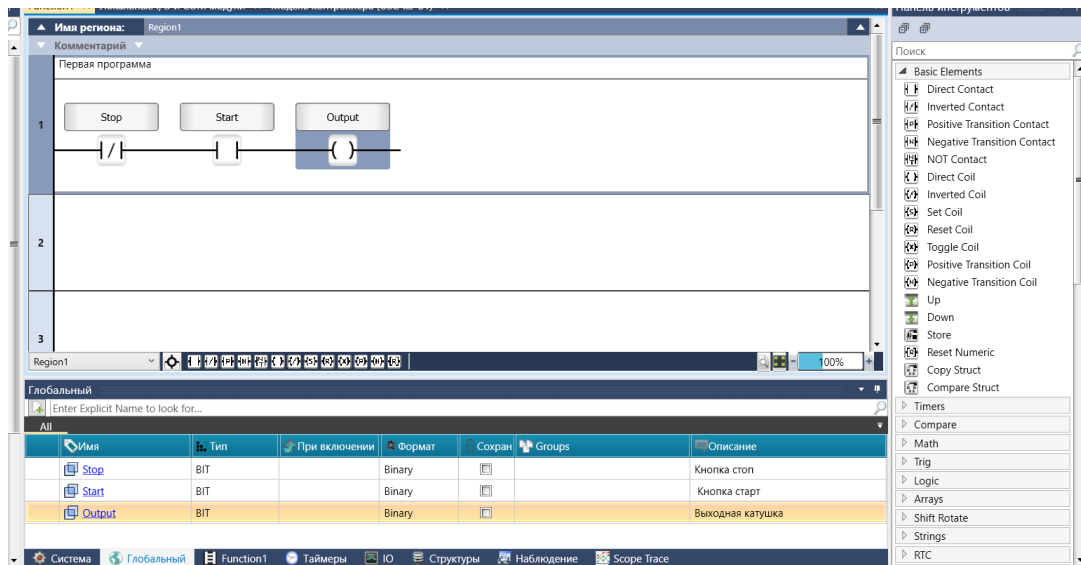


Рис. 66. Добавление катушки

Для того чтобы при отжати кнопки «Старт» цепь не прерывалась, добавляем блокировку для кнопки «Старт». Для этого добавляем прямой контакт параллельно кнопке «Старт», название выбираем из выплывающего списка Output (рис. 67).

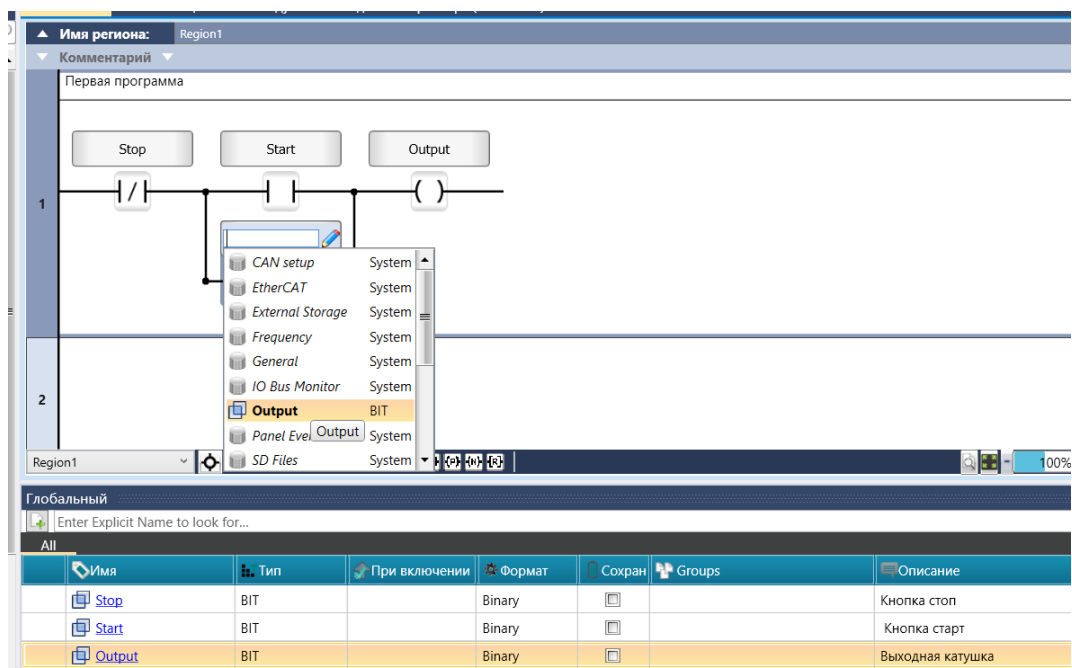


Рис. 67. Добавление блокировки для кнопки «Старт»

Для зажигания лампы по нажатию кнопки «Старт» связываем выход Output с контроллером. Для этого переходим во вкладку О/І (рис. 68).

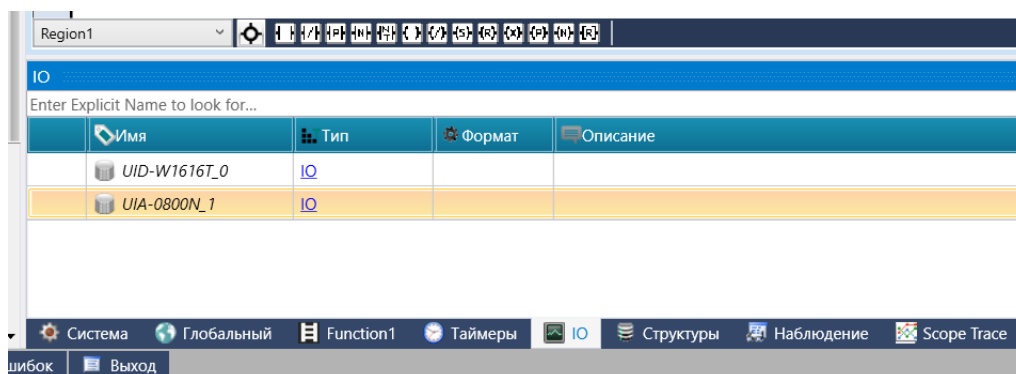


Рис. 68. Вкладка IO

Перейдя во вкладку, выбираем тип данных IO в строке UID-W1616T_0 и в этой вкладке выбираем тип данных BIT[0...15] (рис. 69).

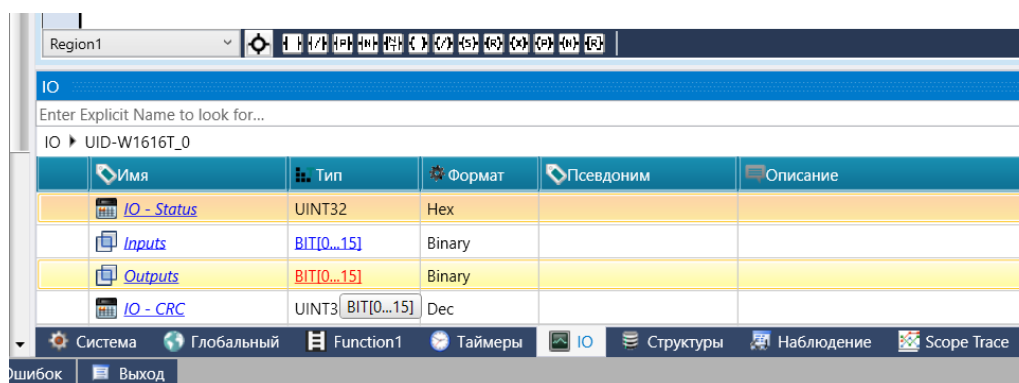


Рис. 69. Выбор типа данных UID-W1616T_0

В строке Outputs_0 в столбце «Псевдоним» вводим: Output_real и в описании вводим: «Нагрузка – лампа» (рис. 70).

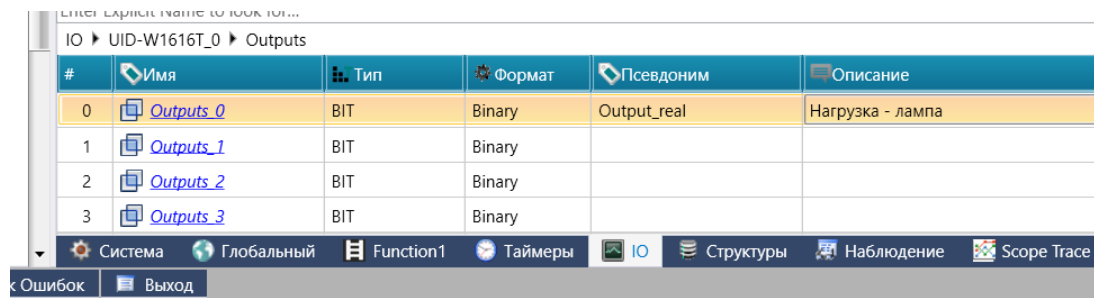


Рис. 70. Настройка выхода контроллера

После добавляем параллельно катушке еще одну, чтобы связать ее с тегом Outputs_0 (рис. 71). Название выбираем из выплывающего списка UID-W1616T_0.

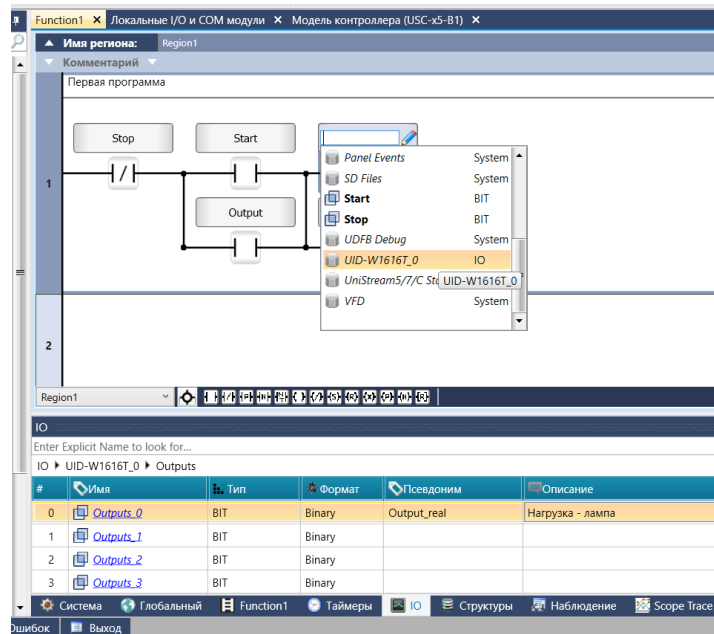


Рис. 71. Добавление катушки и ее привязка к тегу UID-W1616T_0

Связываем катушку с выходом контроллера Output_0 (рис. 72).

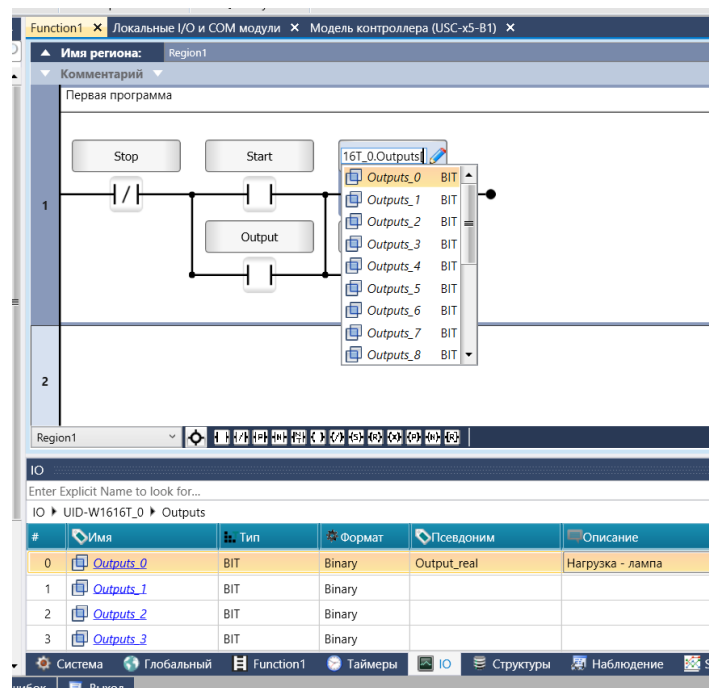


Рис. 72. Связывание катушки с выходом контроллера

Для проверки разработанной программы ее необходимо скомпилировать. Для этого нажимаем в ленте: «Компилировать» и ждем результата проверки (рис. 73).

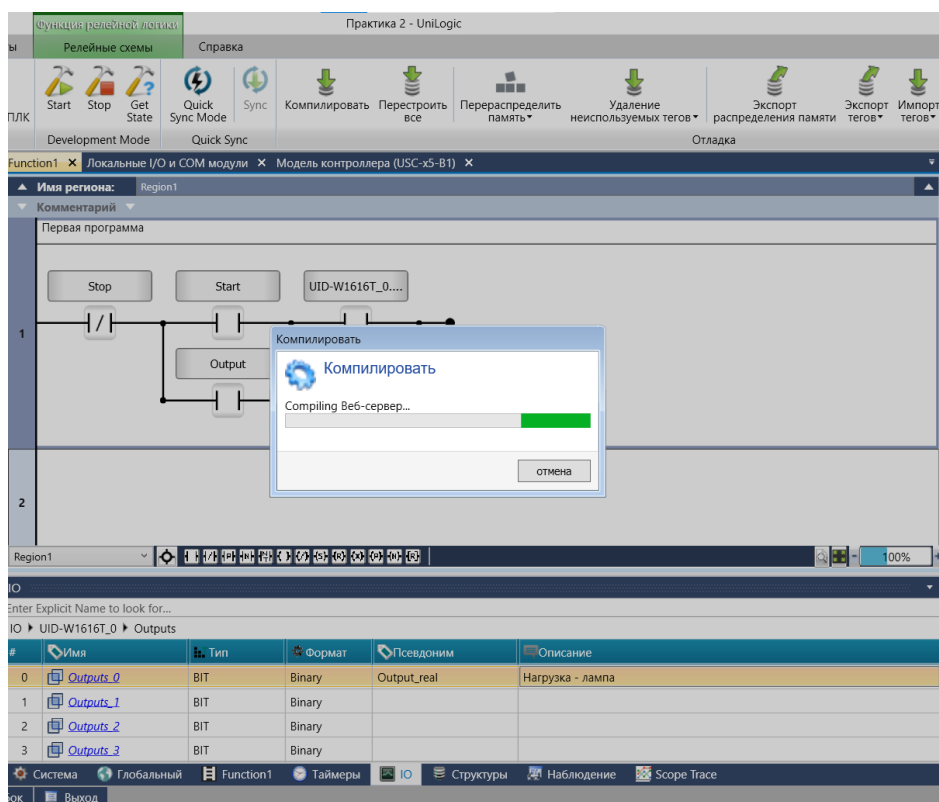


Рис. 73. Проверка программы

Содержание работы:

- 1) цель работы;
- 2) задания;
- 3) ход работы;
- 4) выводы;
- 5) средства, используемые при выполнении лабораторной работы.

Средства, используемые при выполнении лабораторной работы:

- 1) методические указания к выполнению лабораторных работ;
- 2) данные, предоставленные преподавателем во время занятия;
- 3) при проведении анализа допускается использование глобальной сети Интернет.

Лабораторная работа № 3

ПРОГРАММИРОВАНИЕ ТАЙМЕРОВ

Цель работы: изучение принципа работы таймеров и счетчиков.

Задания:

- 1) разработка программы включения лампы с задержкой 5 с;
- 2) разработка программы выключения лампы с задержкой 1 ч;
- 3) разработка программы включения и выключения лампы периодически 10 с;
- 4) разработка программы сброса импульсного таймера каждые 15 с.

Ход работы

1. Разработка программы задержки на включение лампы в программе UniLogic.

Для разработки программы открываем UniLogic. Выбираем стандартный ПЛК и необходимые модули к нему, как в лабораторной работе № 1. Выбрав модули, в проводнике решений находим Function1. Нажимаем на данную вкладку ПКМ и во всплывающем окне выбираем «Rename функции» (рис. 74).

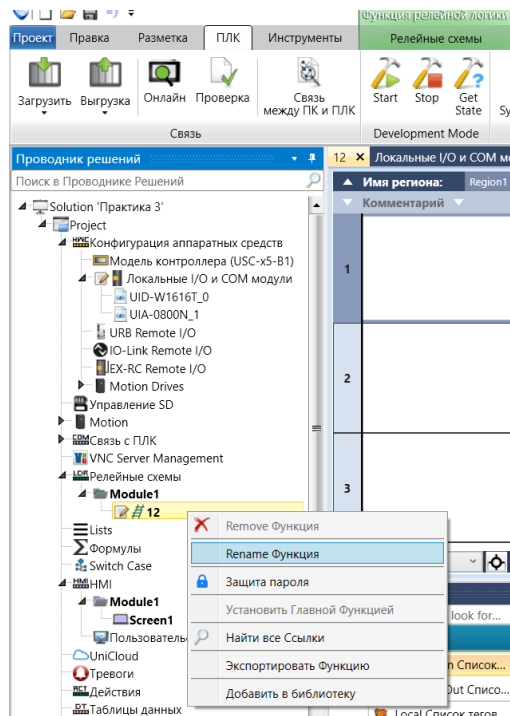


Рис. 74. Rename функции

Данная функция позволяет назвать вкладку. Называем вкладку «Таймеры» (рис. 75).

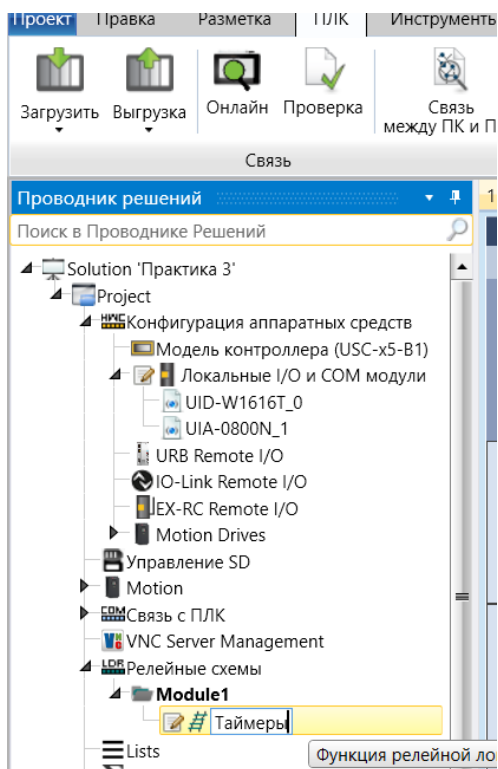


Рис. 75. Название вкладки

Для каждой из четырех программ будет использоваться отдельный регион (рис. 76).

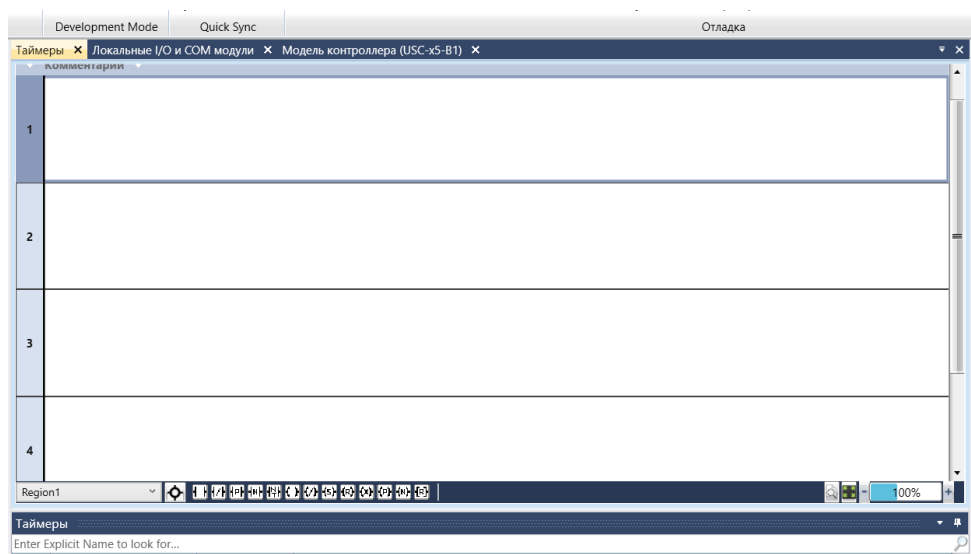


Рис. 76. Регион

Создаем для каждого региона комментарий, в котором будет указываться название создаваемой программы. В первом комментарии пишем: «Таймер с задержкой включения (Timer on Delay)» (рис. 77).

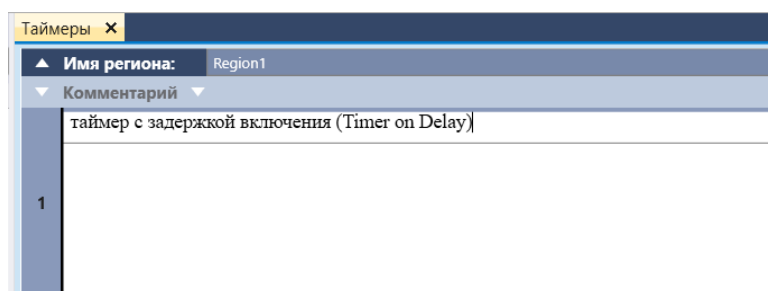


Рис. 77. Название первого региона

Во втором комментарии пишем: «Таймер с задержкой выключения» (Timer off Delay)» (рис. 78).

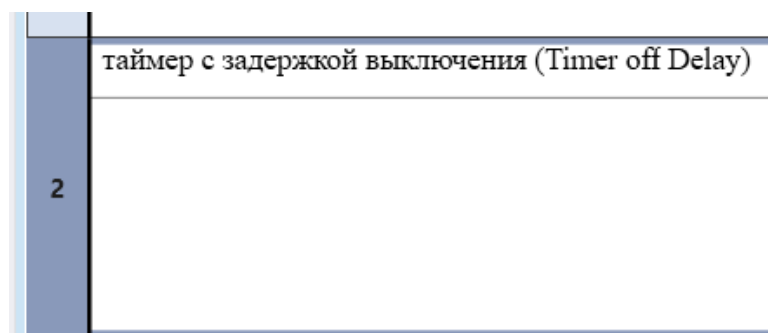


Рис. 78. Название второго региона

В третьем комментарии пишем: «Импульсный таймер (Pulse Timer)» (рис. 79).

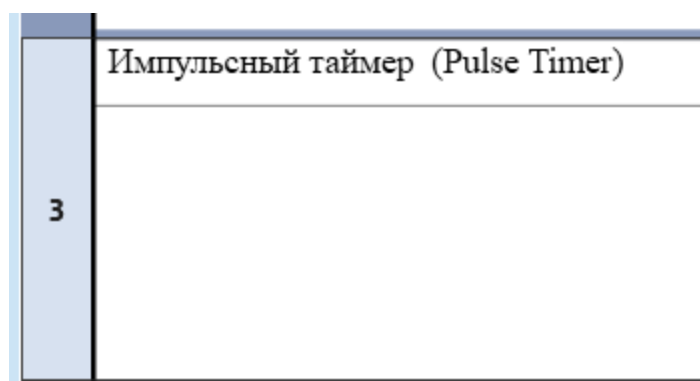


Рис. 79. Название третьего региона

В четвертом комментарии пишем: «ФБ сброса таймера (Timer Reset)» (рис. 80).

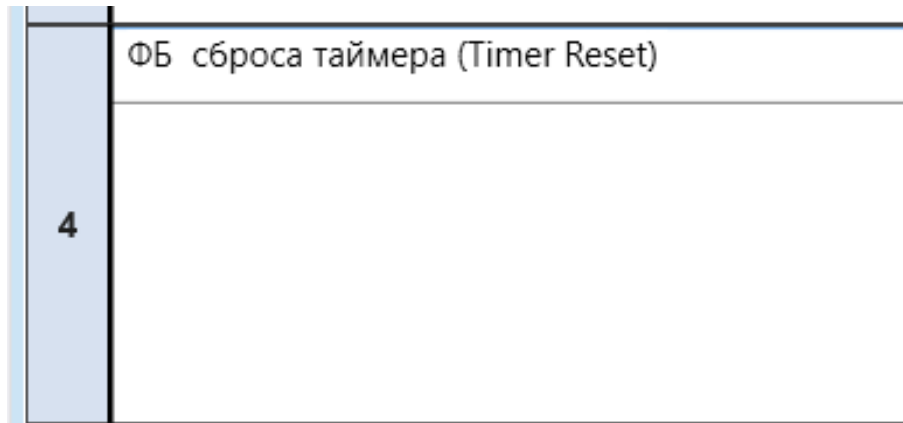


Рис. 80. Название четвертого региона

В комментариях указывается, какая программа будет разрабатываться в данном регионе. Приступим к разработке первой программы. В панели инструментов открываем вкладку Timers (рис. 81).

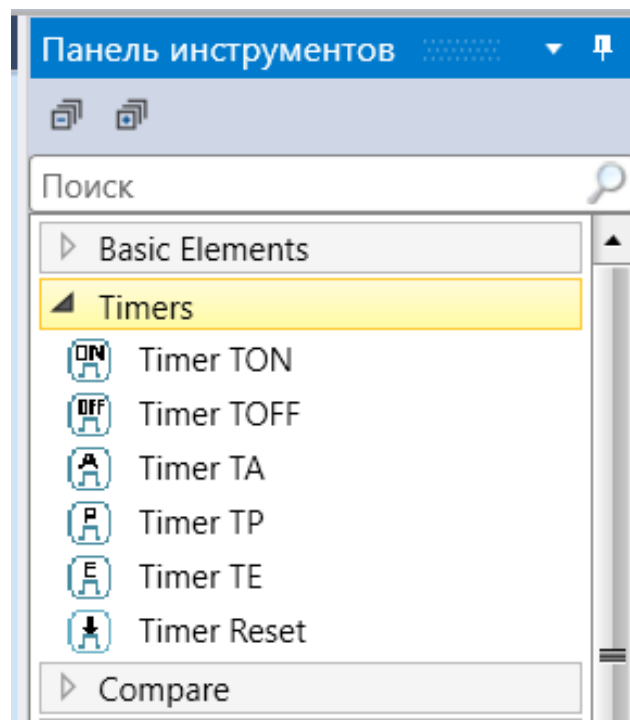


Рис. 81. Вкладка Timers

Из вкладки Timers перетаскиваем Timer NON в первый регион (рис. 82).

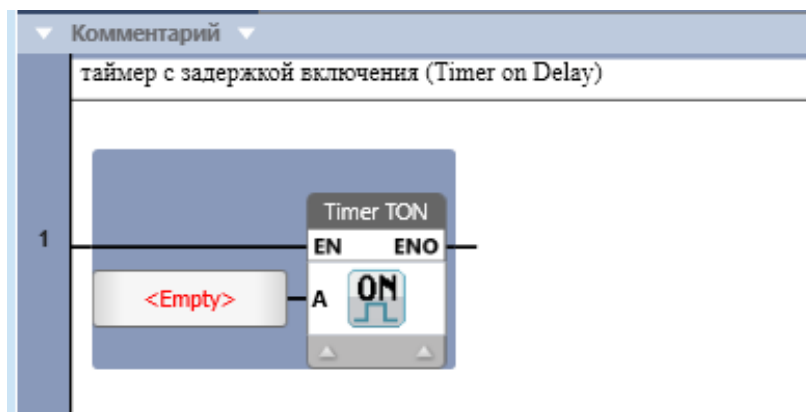


Рис. 82. Таймер с задержкой включения

Для данного таймера создаем тег. Для этого открываем вкладку «Таймеры» в панели тегов и нажатием ЛКМ создаем новый тег (рис. 83).

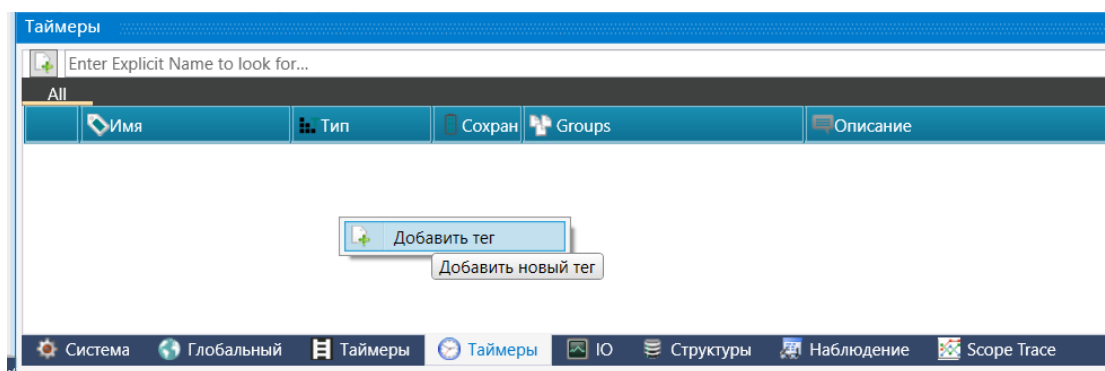


Рис. 83. Создание тега для таймера с задержкой включения

Называем тег TON_1, остальное оставляем без изменений и сохраняем (рис. 84).

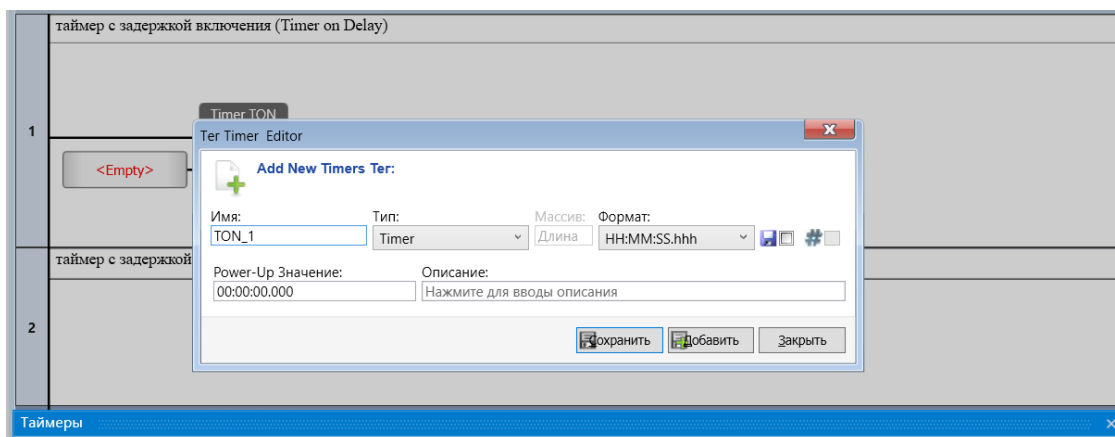


Рис. 84. Настройка тега для таймера с задержкой включения

Добавляем выход таймера (это нормально открытый контакт) и привязываем его к выходу таймера (рис. 85).

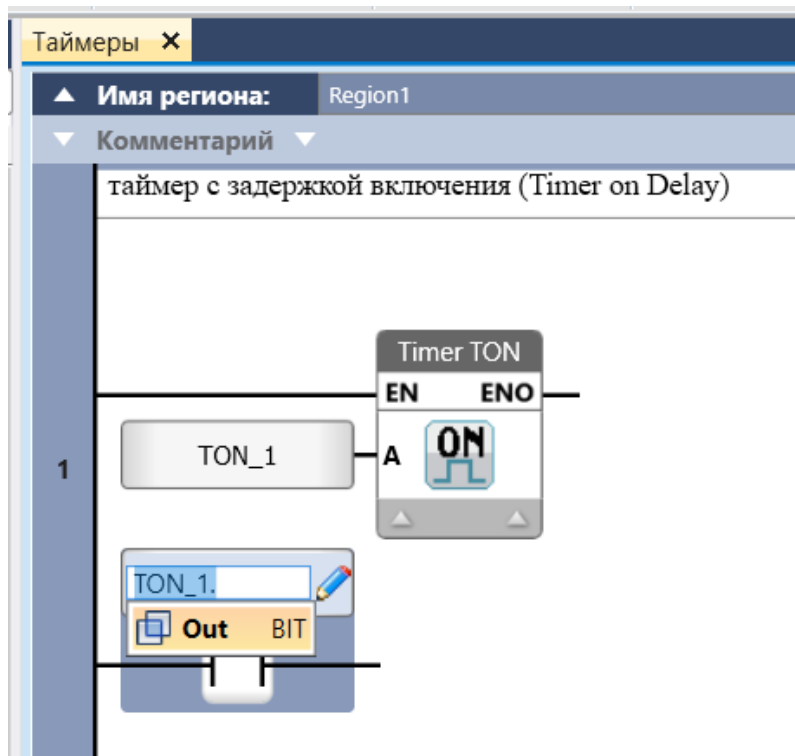


Рис. 85. Привязка нормально открытого контакта к выходу таймера

Выставляем время задержки таймера на включение. Задержка будет составлять 5 с. Для этого открываем сначала вкладку «Таймеры», затем вкладку «Тип таймера» и устанавливаем значение 5 с (рис. 86).

Имя	Тип	При включении	Формат	Псевдоним	Сохран	Оп
Preset	UINT32	00:00:05.000	HH:MM:SS.hhh		<input type="checkbox"/>	
Current	UINT32	00:00:00.000	HH:MM:SS.hhh		<input type="checkbox"/>	
Out	BIT		00:00:00.000		<input type="checkbox"/>	

Рис. 86. Выставление задержки таймера

К выходу таймера добавляем индукционную катушку – лампу, которая будет загораться через 5 с после включения таймера (рис. 87).

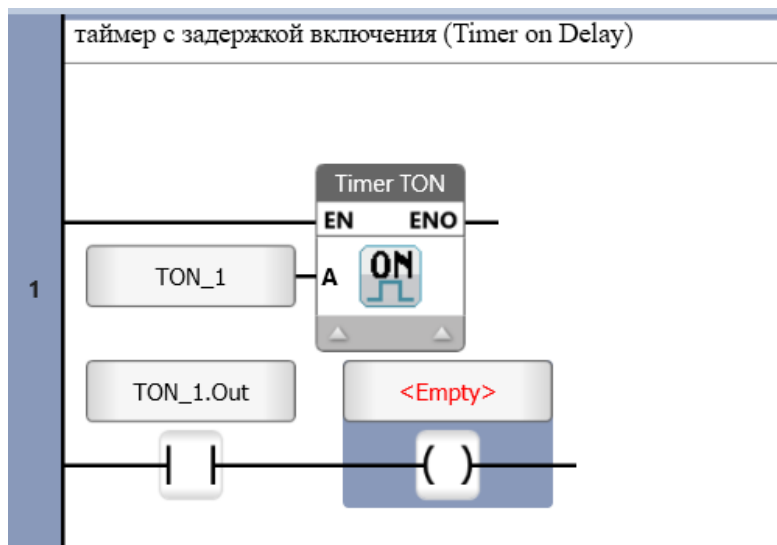


Рис. 87. Добавление лампы

Для добавленной лампы создаем и привязываем глобальный тег. Для этого в панели тегов нажимаем на вкладку «Глобальный». Щелчком ЛКМ добавляем новый тег, называем его Q1 (рис. 88) и привязываем его к лампе (рис. 89).

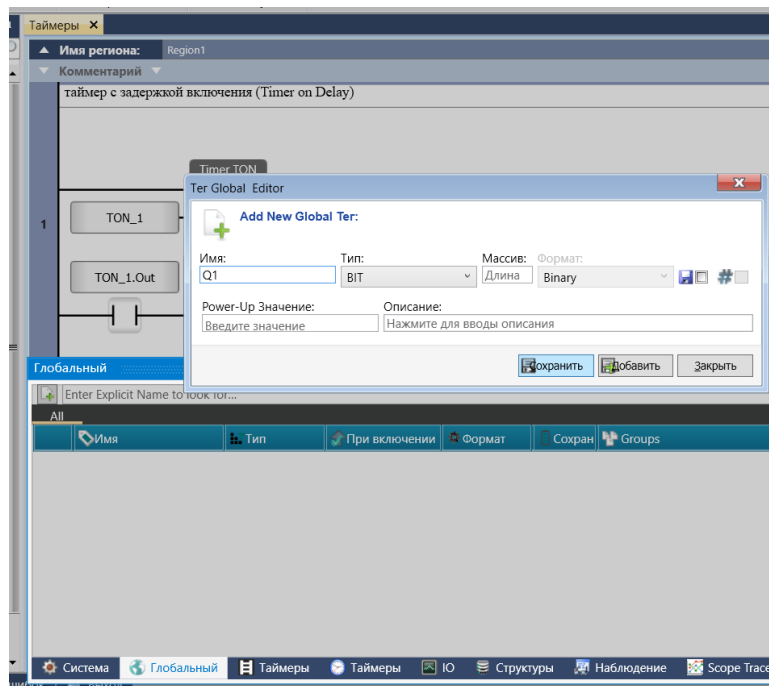


Рис. 88. Добавление тега для лампы

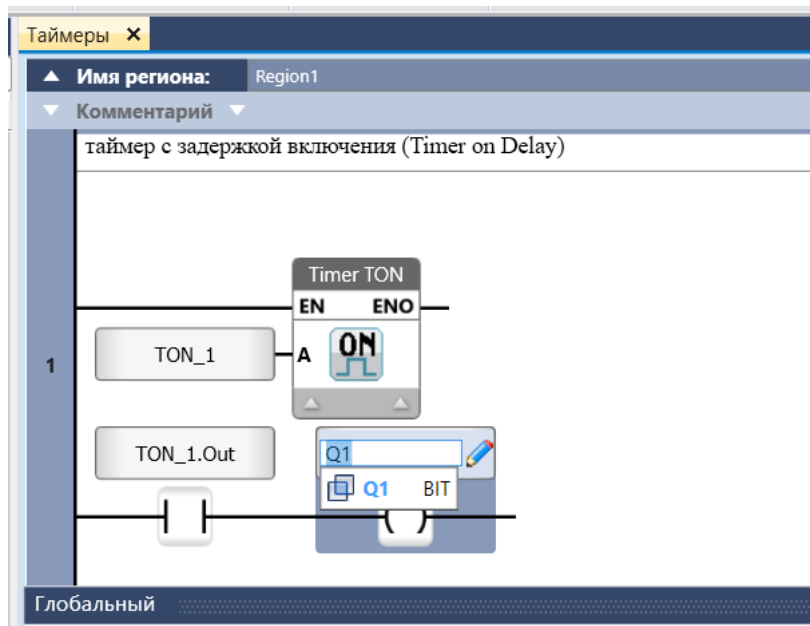


Рис. 89. Привязка тега к лампе

Для включения таймера добавляем кнопку «Старт». Создаем для нее тег под названием «Старт лампа» и привязываем его к кнопке (рис. 90).

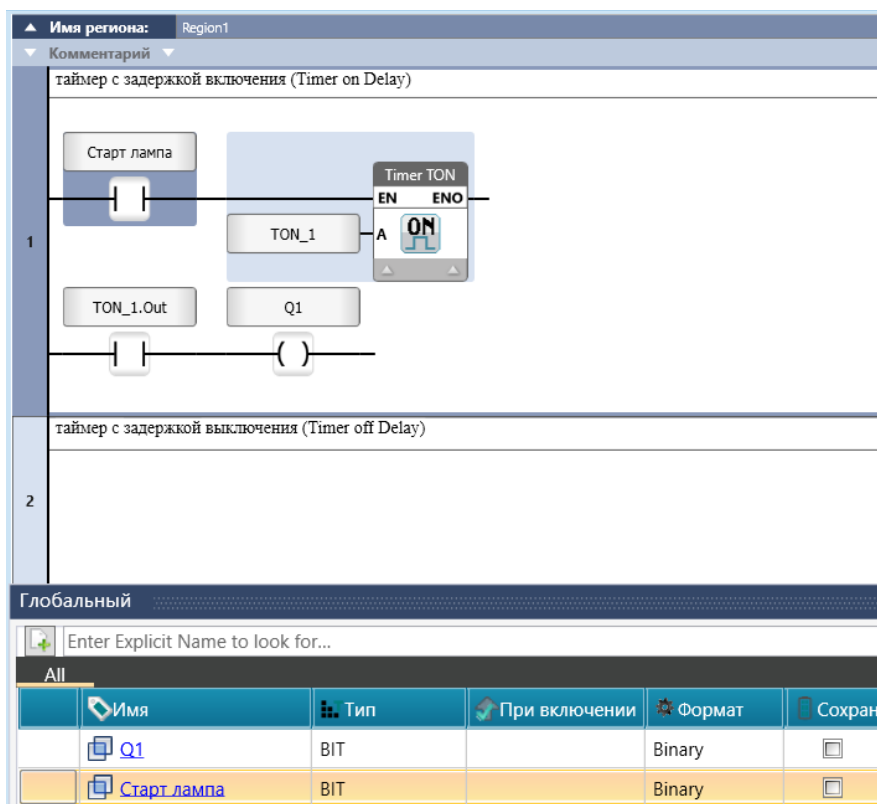


Рис. 90. Добавление кнопки, создание тега и привязка его к кнопке

2. Разработка программы выключения лампы с задержкой 1 ч. Добавляем все необходимые логические элементы во второй регион (рис. 91).

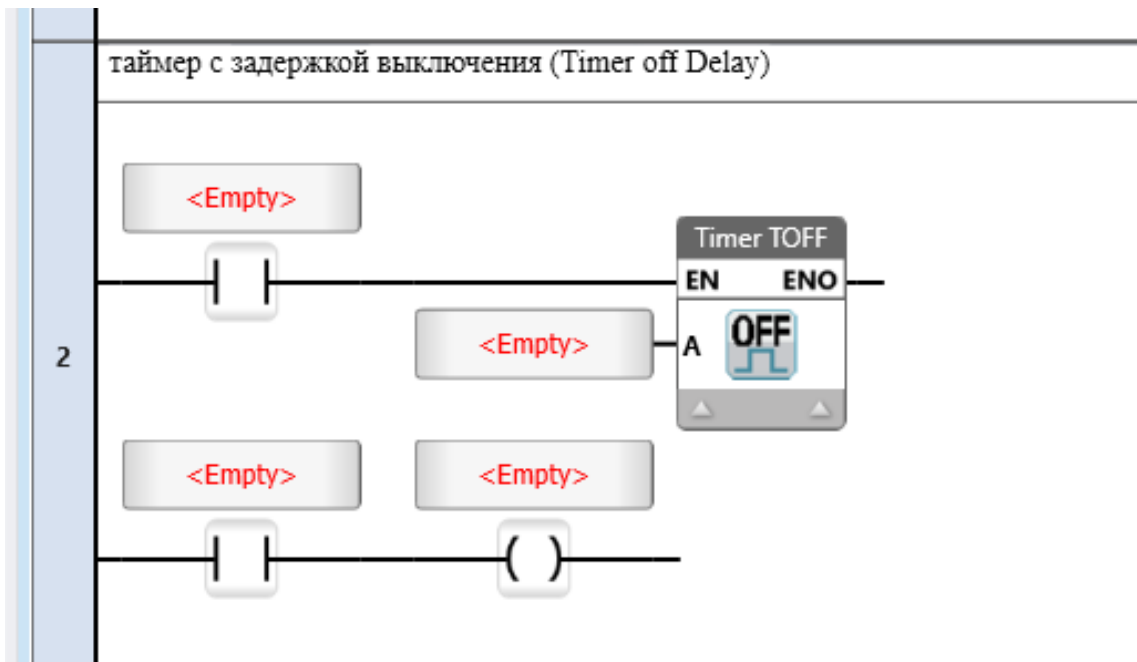


Рис. 91. Добавление элементов релейной логики для разработки программы выключения лампы с задержкой 1 ч

Добавив логические элементы, создаем теги и привязываем их к логическим элементам (рис. 92).

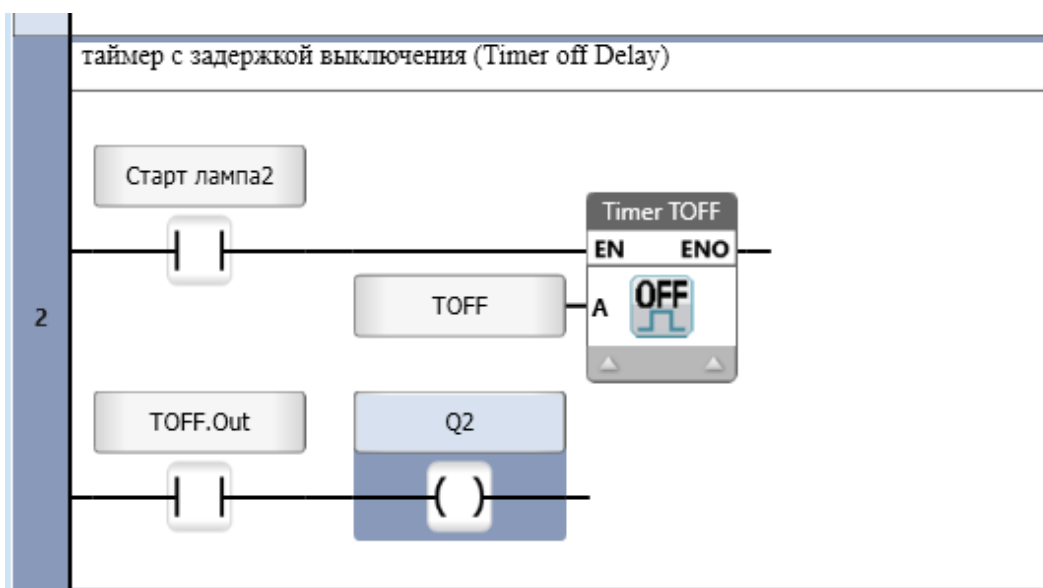


Рис. 92. Создание и привязка логических элементов

Для выключения лампы через час необходимо настроить таймер и выставить в нем время. Для этого делаем все то же, что и на рис. 86, но выставляем время 1 ч (рис. 93).

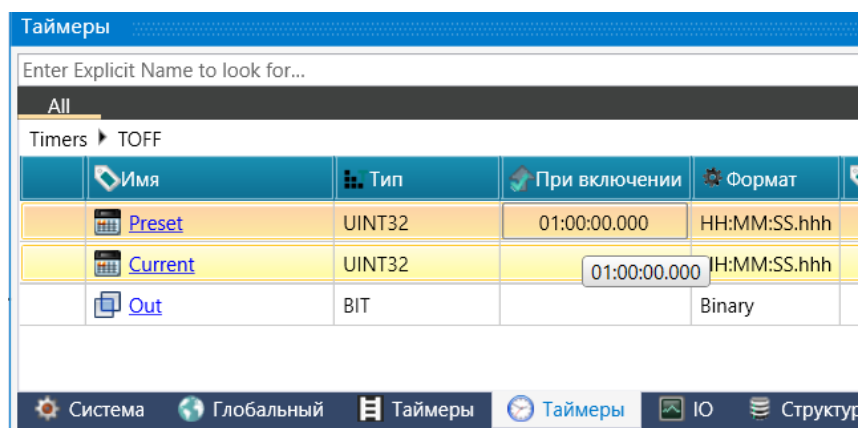


Рис. 93. Настройка таймера на выключение

3. Разработка программы включения и выключения лампы периодичностью 10 с.

Для разработки данной программы необходимо проделать те же действия, что и на рис. 91, заменив Timer TOFF на Timer TP (рис. 94).

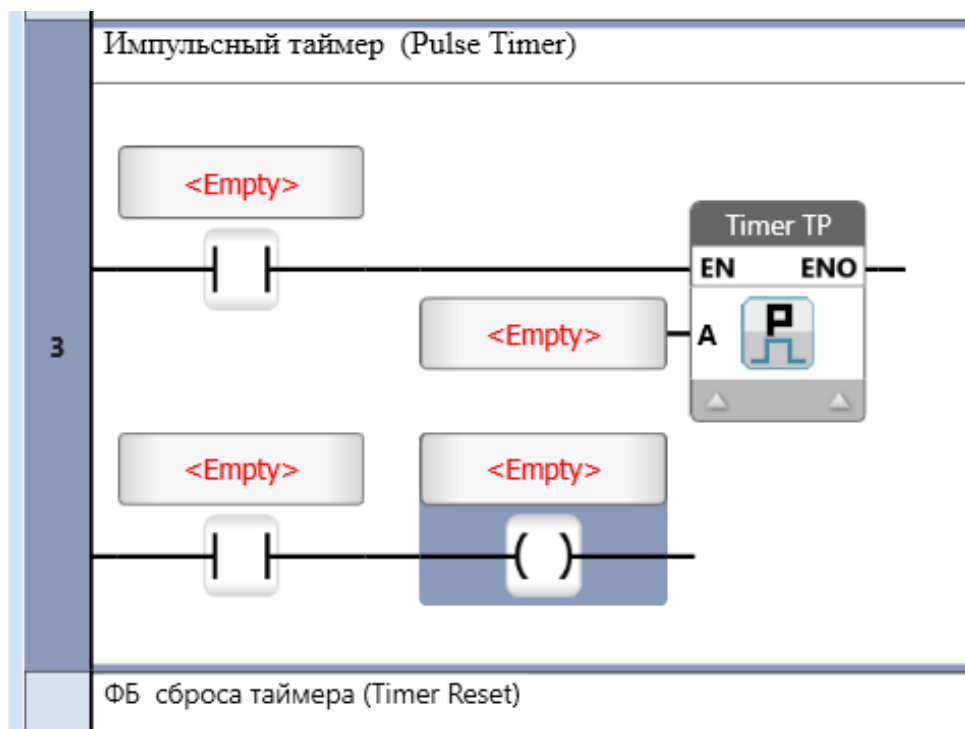


Рис. 94. Добавление элементов релейной логики для разработки программы включения и выключения лампы периодичностью 10 с

После добавления логических элементов создаем и привязываем теги к ним. А также задаем значение, через которое лампа будет включаться и выключаться (рис. 95).

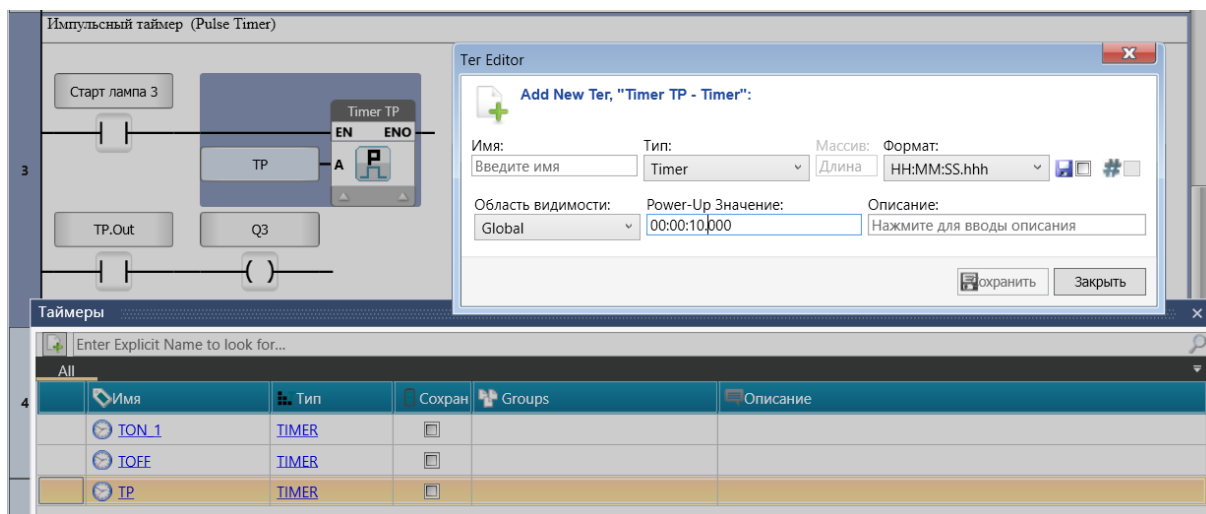


Рис. 95. Создание тегов, их привязка и настройка таймера на включение и выключение лампы каждые 10 с

4. Разработка программы сброса импульсного таймера каждые 15 с. Переносим Timer Reset из панели инструментов в четвертый регион (рис. 96).

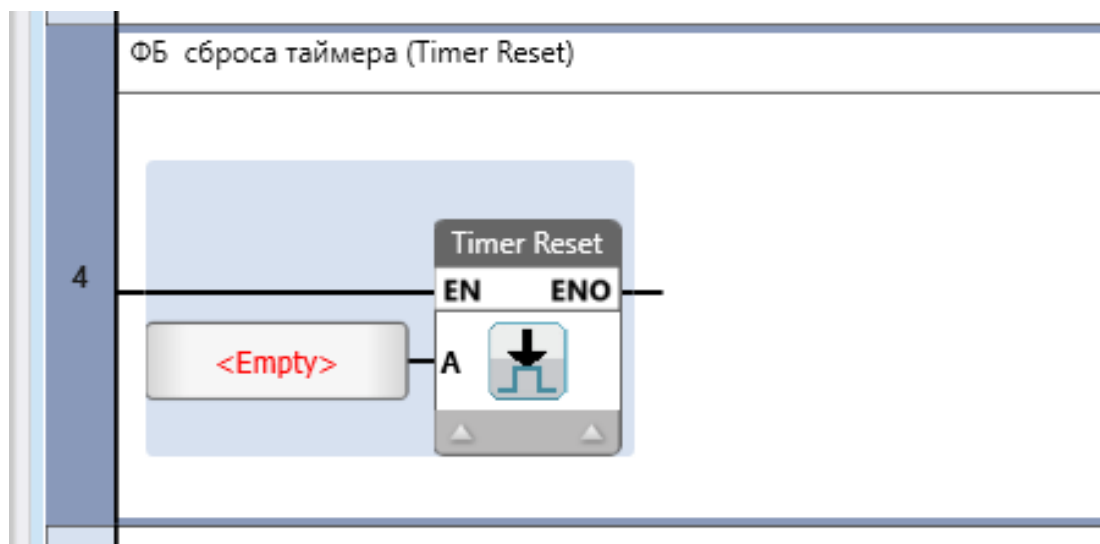


Рис. 96. Таймер сброса

Добавляем кнопку «Старт», создаем для нее тег и привязываем кнопку к тегу (рис. 97).

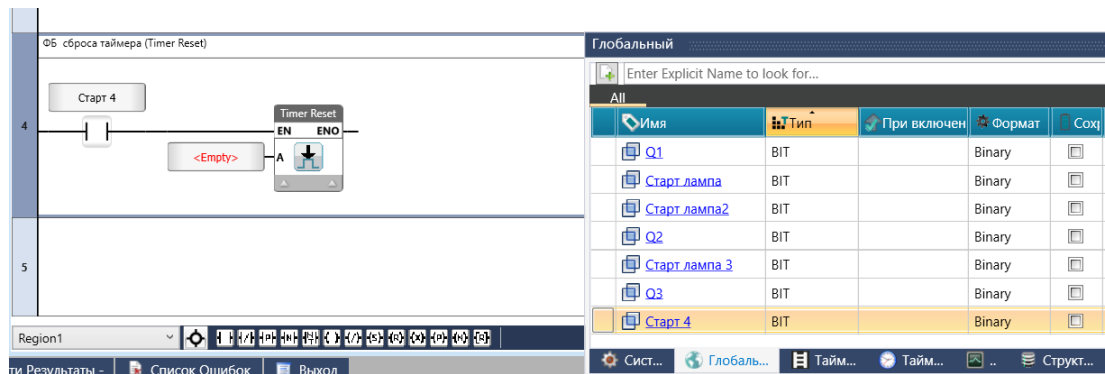


Рис. 97. Программирование кнопки «Старт»

Привязываем таймер сброса к импульсному таймеру и тем самым сбрасываем импульсный таймер за один цикл сканирования (рис. 98).

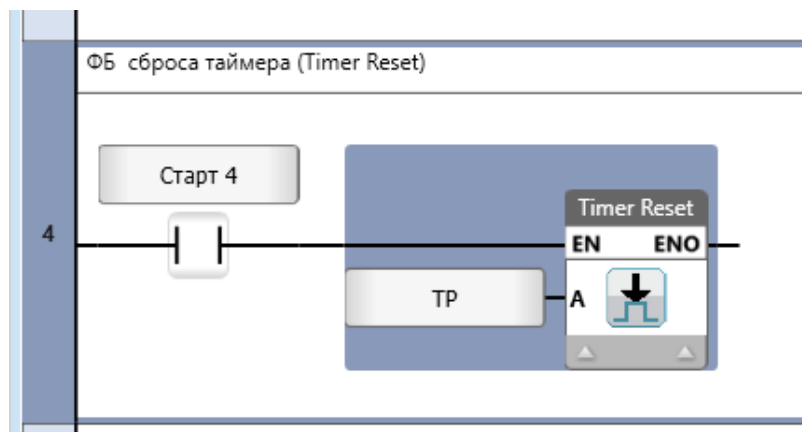


Рис. 98. Настройка таймера сброса

Содержание работы:

- 1) цель работы;
- 2) задания;
- 3) ход работы;
- 4) выводы;
- 5) средства, используемые при выполнении лабораторной работы.

Средства, используемые при выполнении лабораторной работы:

- 1) методические указания к выполнению лабораторных работ;
- 2) данные, предоставленные преподавателем во время занятия;
- 3) при проведении анализа допускается использование глобальной сети Интернет.

Лабораторная работа № 4

ПРОГРАММИРОВАНИЕ СЧЕТЧИКОВ

Цель работы: изучение принципа работы счетчиков.

Задания:

1) реализовать расчет выходного состояния (X) секции на основе статусов трех входов, представленных на схеме (рис. 99);

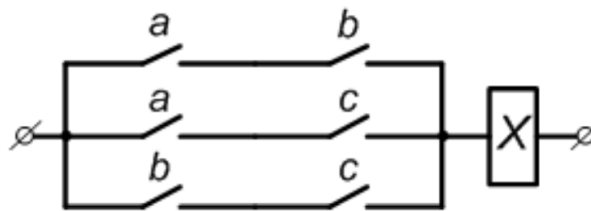


Рис. 99. Схема статусов трех входов

2) изучить битовые логические команды. Для этого из панели инструментов во вкладке Logic зажимаем ЛКМ на логической операции And и переносим ее в первый регион.

Ход работы

Для начала создаем первую ветку А и Б; для этого в панели инструментов во вкладке Logic зажимаем ЛКМ на логической операции And и переносим ее в первый регион (рис. 100).

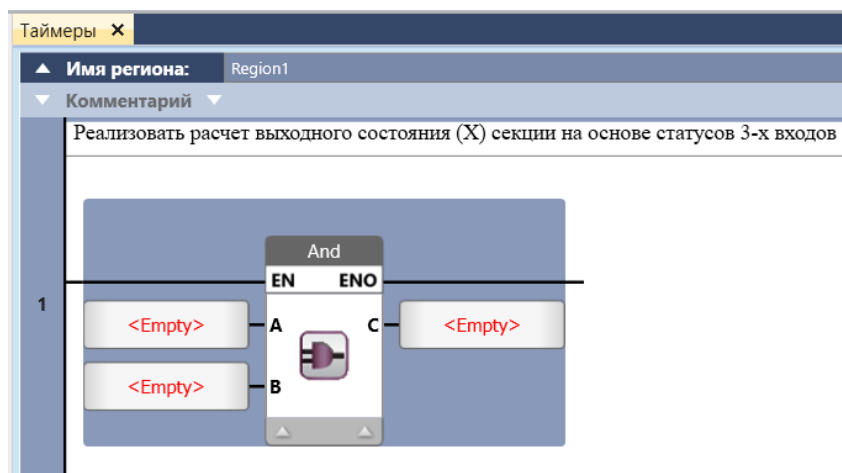


Рис. 100. Логическая операция And

После создаем теги А, Б и А_Б, настраиваем их тип на INT8 и осуществляем их привязку к логической операции And, как показано на рис. 101 – 103.

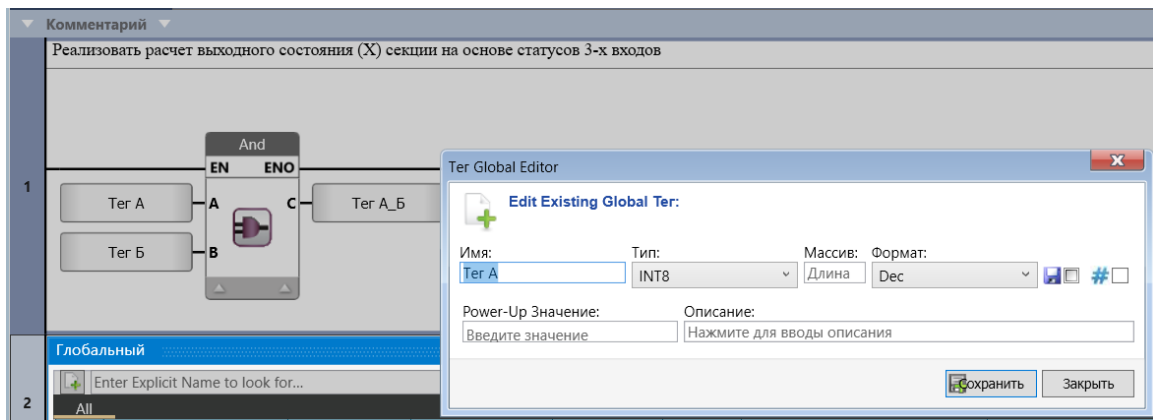


Рис. 101. Создание тега А

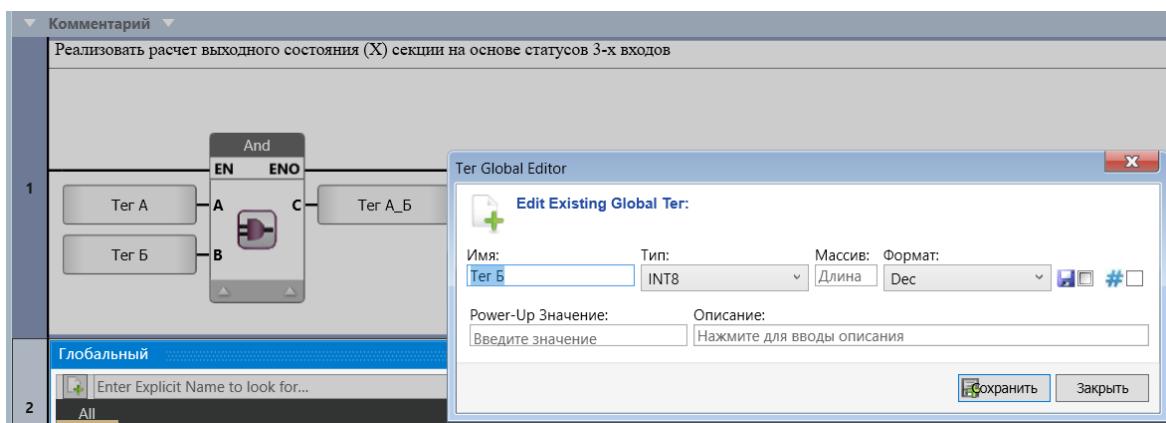


Рис. 102. Создание тега Б

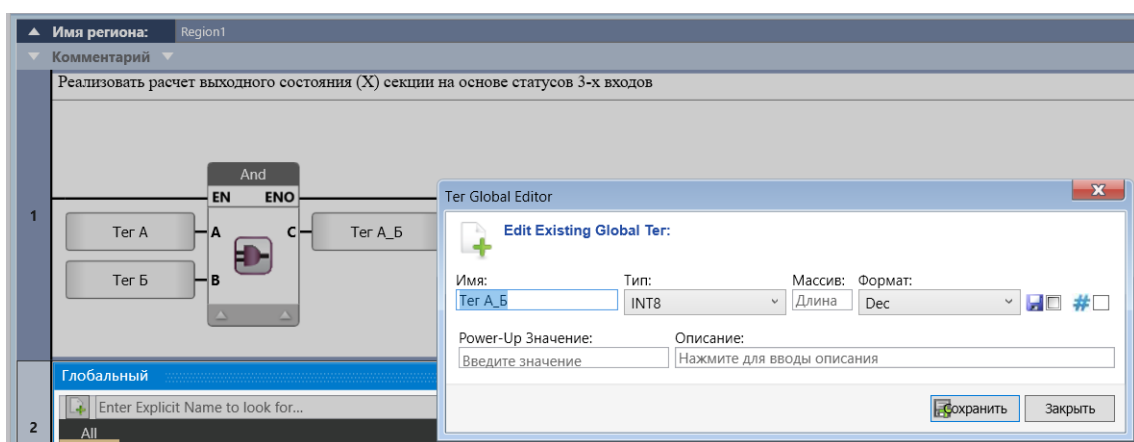


Рис. 103. Создание тега А_Б

Последовательно к ветке А и Б прикрепляем ветку Б и С. Для этого из панели инструментов во вкладке logic зажимаем ЛКМ на логической операции And и переносим ее в первый регион (рис. 104).

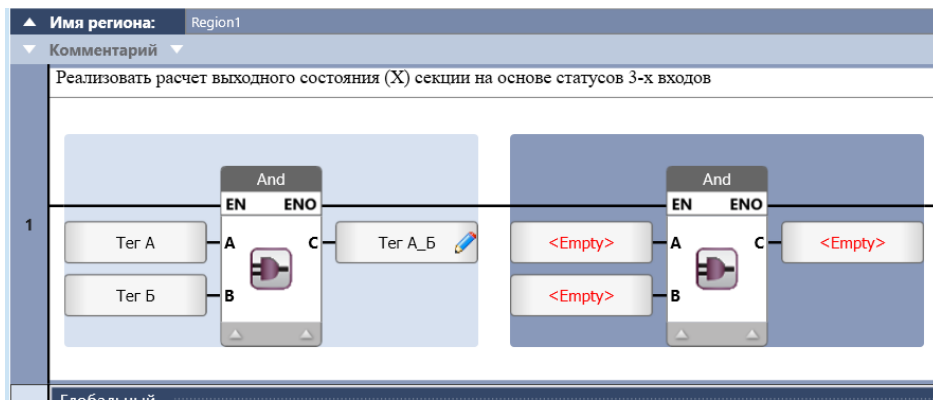


Рис. 104. Присоединение ветки Б и С

Теперь создаем для ветки Б и С теги С, Б_С, настраиваем их тип на INT8 и привязываем их к логической операции And (рис. 105, 106).

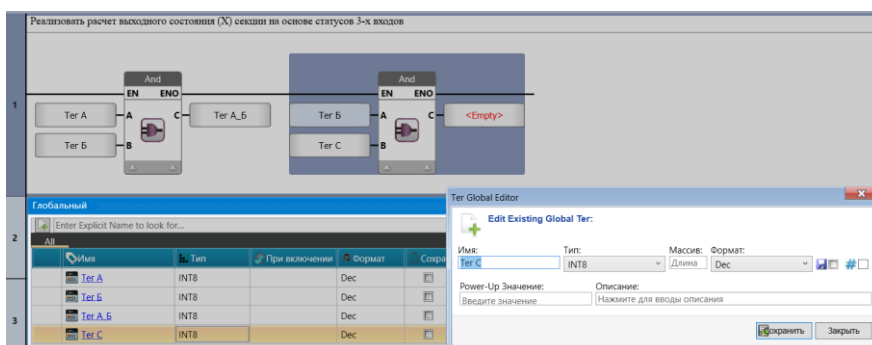


Рис. 105. Тег С

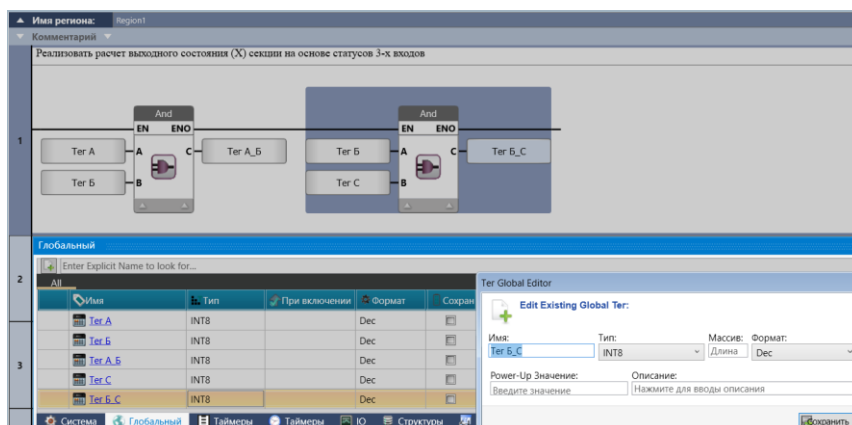


Рис. 106. Тег Б_С

Последовательно к ветке Б и С прикрепляем ветку А и С. Для этого из панели инструментов во вкладке logic зажимаем ЛКМ на логической операции And и переносим ее в первый регион (рис. 107).

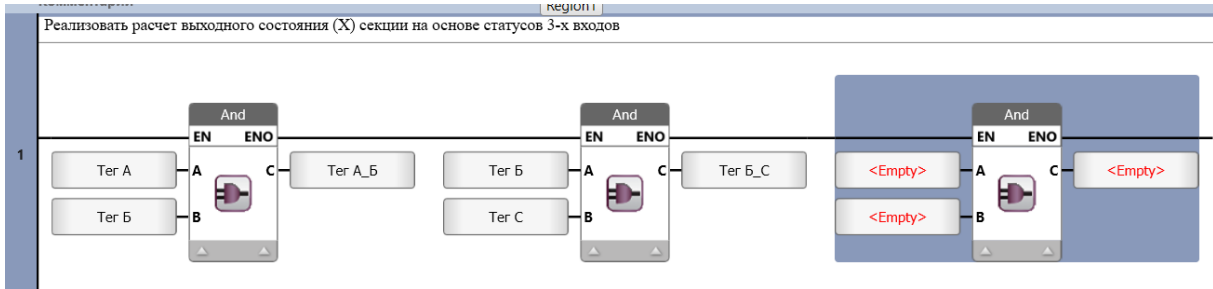


Рис. 107. Присоединение ветки А и С

Затем создаем для ветки А и С тег А_С, настраиваем его тип на INT8 и привязываем его к логической операции And (рис. 108).

Имя	Тип	При включении	Формат	Со
Ter Б	INT8		Dec	
Ter А_Б	INT8		Dec	
Ter С	INT8		Dec	
Ter Б_С	INT8		Dec	
A_С	INT8		Dec	

Ter Global Editor

Edit Existing Global Ter:

Имя: Тип: Массив: Формат:

Power-Up Значение: Описание:

Рис. 108. Тег А_С

Для реализации выходного параметра X необходимо ввести логические операции ИЛИ, поэтому параллельно к трем веткам прикрепляем логическую операцию ИЛИ. Для этого из панели инструментов во вкладке Logic зажимаем ЛКМ на логической операции Or и переносим ее в первый регион (рис. 109).

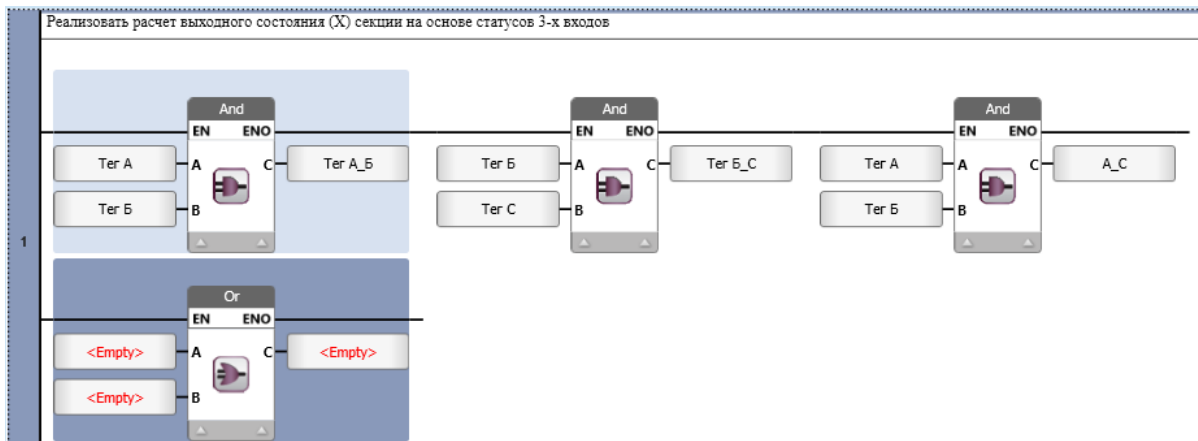


Рис. 109. Присоединение логической операции A_B или B_C

Затем создаем для логической операции A_B или B_C тег $A_B\&B_C$, настраиваем его тип на INT8 и привязываем его к логической операции And (рис. 110).

Реализовать расчет выходного состояния (X) секции на основе статусов 3-х входов

Имя	Тип	Массив	Формат	Power-Up	Значение	Описание
Ter A_B	INT8		Dec	<input type="checkbox"/>		
Ter C	INT8		Dec	<input type="checkbox"/>		
Ter B_C	INT8		Dec	<input type="checkbox"/>		
A_C	INT8		Dec	<input type="checkbox"/>		
A_B&B_C	INT8		Dec	<input type="checkbox"/>		

Рис. 110. Тег $A_B\&B_C$

Последовательно к логической операции A_B или B_C прикрепляем логическую операцию $A_B\&B_C$ или A_C . Для этого из панели инструментов во вкладке Logic зажимаем ЛКМ на логической операции Or и переносим ее в первый регион (рис. 111).

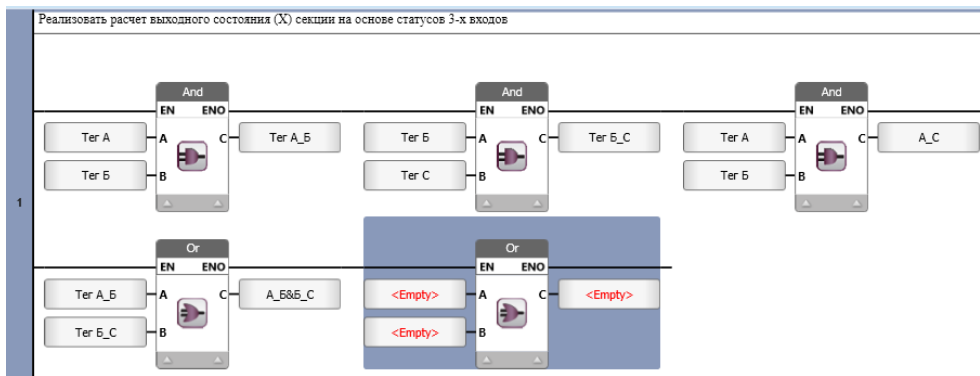


Рис. 111. Присоединение логической операции $A_B \& B_C$ или A_C

Затем создаем для логической операции $A_B \& B_C$ или A_C тег «Выход X», настраиваем его тип на INT8 и привязываем его к логической операции $A_B \& B_C$ или A_C (рис. 112).

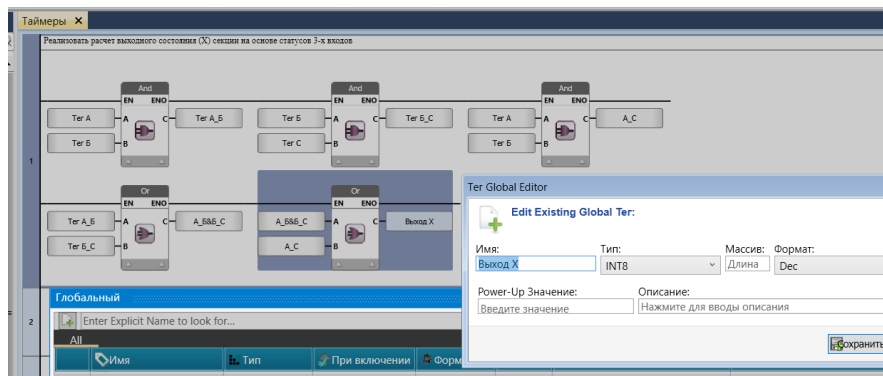


Рис. 112. Тег «Выход X»

После проект компилируем и при отсутствии ошибок сохраняем.

Содержание работы:

- 1) цель работы;
- 2) задания;
- 3) ход работы;
- 4) выводы;
- 5) средства, используемые при выполнении лабораторной работы.

Средства, используемые при выполнении лабораторной работы:

- 1) методические указания к выполнению лабораторных работ;
- 2) данные, предоставленные преподавателем во время занятия;
- 3) при проведении анализа допускается использование глобальной сети Интернет.

Лабораторная работа № 5

РАЗРАБОТКА ТЕСТОВОГО ЭКРАНА HMI

Цель работы: освоить навыки разработки экрана HMI.

Задание: разработать тестовый экран HMI.

Ход работы

В программном обеспечении Unitronics UniLogic во вкладке «Проводник решений» открываем HMI. В этой вкладке выбираем Module1, нажимаем на нее ЛКМ и выбираем ПКМ «Add Экран» (рис. 113).

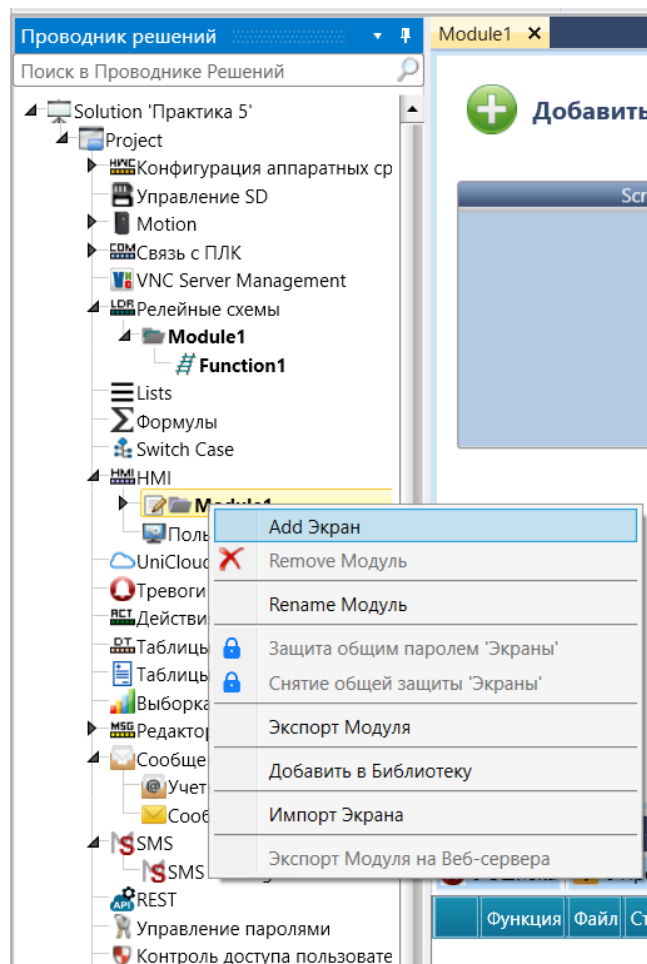


Рис. 113. Добавление в проект экран HMI

В разделе «Свойства внешнего вида» выбираем цветовую палитру будущего экрана HMI (рис. 114).

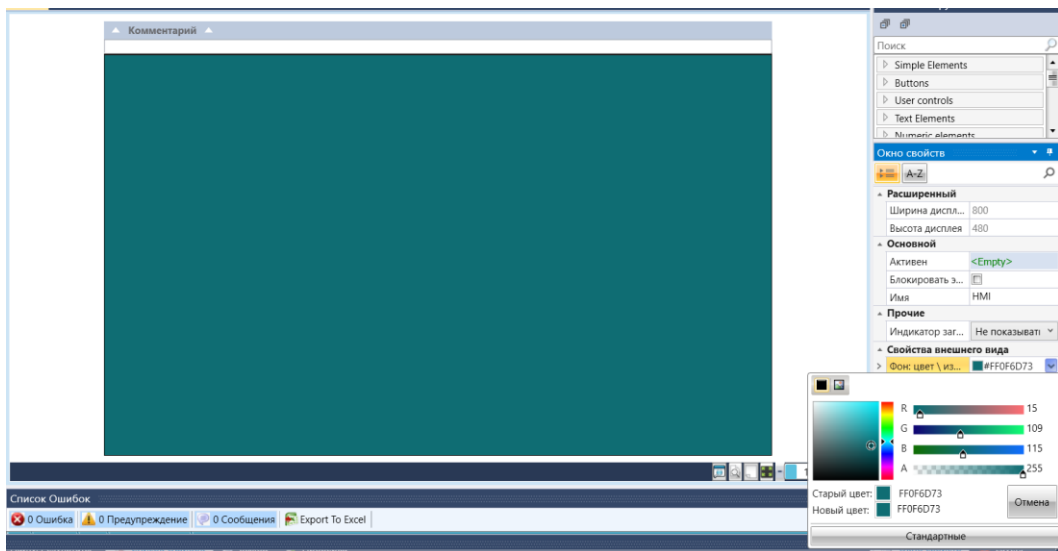


Рис. 114. Выбор цвета экрана HMI

Разрабатываем верхний заголовок экрана HMI. Для этого выбираем свойства внешнего вида элементов Rectangle; цвет: $R = 15$, $G = 109$, $B = 115$; расположение элемента – угол = 0; позицию по горизонтали = 0; позицию по вертикали = 0, размер: высота = 50, ширина = 800 (рис. 115).

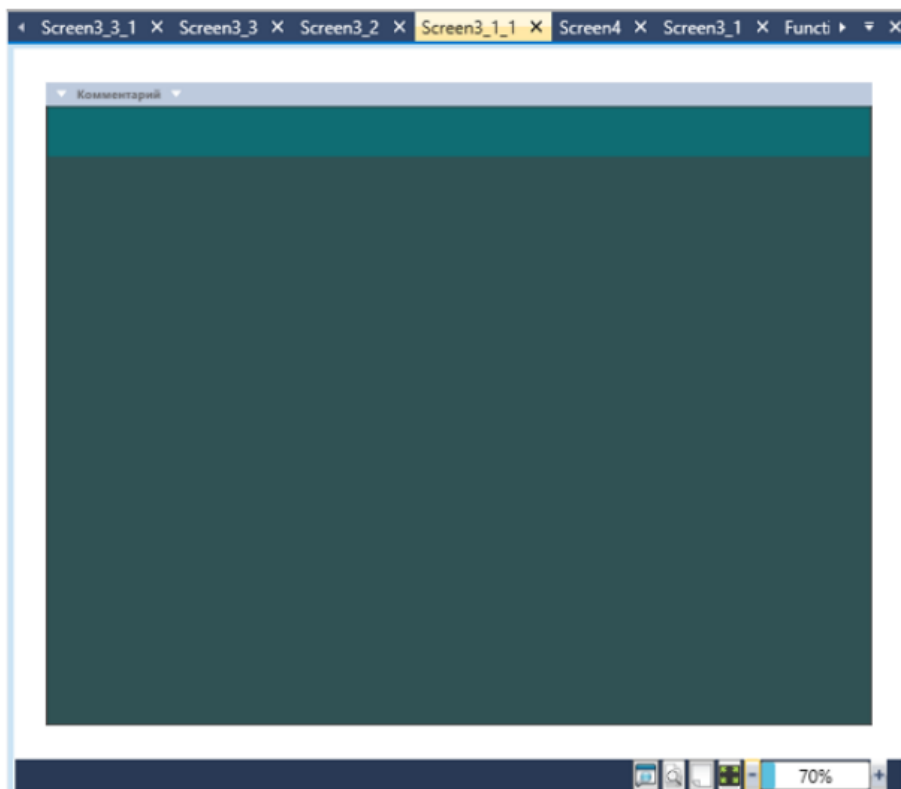


Рис. 115. Разрабатываем верхний заголовок экрана HMI

После открываем источник внешнего вида элемента Line, тем самым создавая линию. Выбираем цвет линии: R = 40, G = 127, B = 106; расположение элемента: X1 = 799, Y1 = 53, X2 = 0, Y2 = 53; размер: толщина линии = 10 (рис. 116).

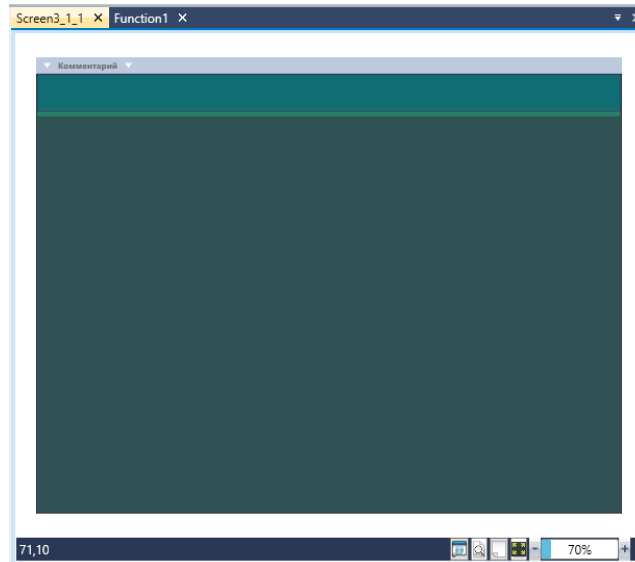


Рис. 116. Верхний заголовок экрана HMI

Для того чтобы создать текстовый элемент в заголовке, выбираем источник элемента, а в нем – фиксированный текст. Вводим следующий текст: «Управление одним выходом ПЛК от двух кнопок». Для редактирования текста выбираем фиксированный текст, а в нем шрифт Bitstream Vera Sans, размер 22. Для изменения цвета текста сначала выбираем фон, затем цвет\изображение и transparent (рис. 117).

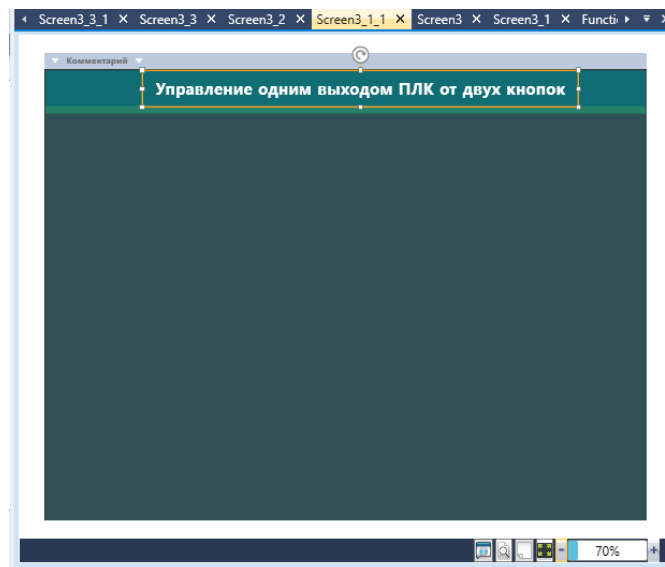


Рис. 117. Редактирование текста верхнего заголовка экрана HMI

После создаем пять форм для HMI и задаем свойства внешнего вида элемента Rectangle, размер и расположение элемента Rectangle.

Для формы № 1:

- свойства внешнего вида элемента Rectangle:

фон: цвет\изображение: Transparent;

цвет границы: белый;

Толщина границы: 1;

- размер и расположение элемента Rectangle:

угол = 0;

ширина = 562;

высота = 386;

позиция по горизонтали = 10;

позиция по вертикали = 89 (рис. 118).

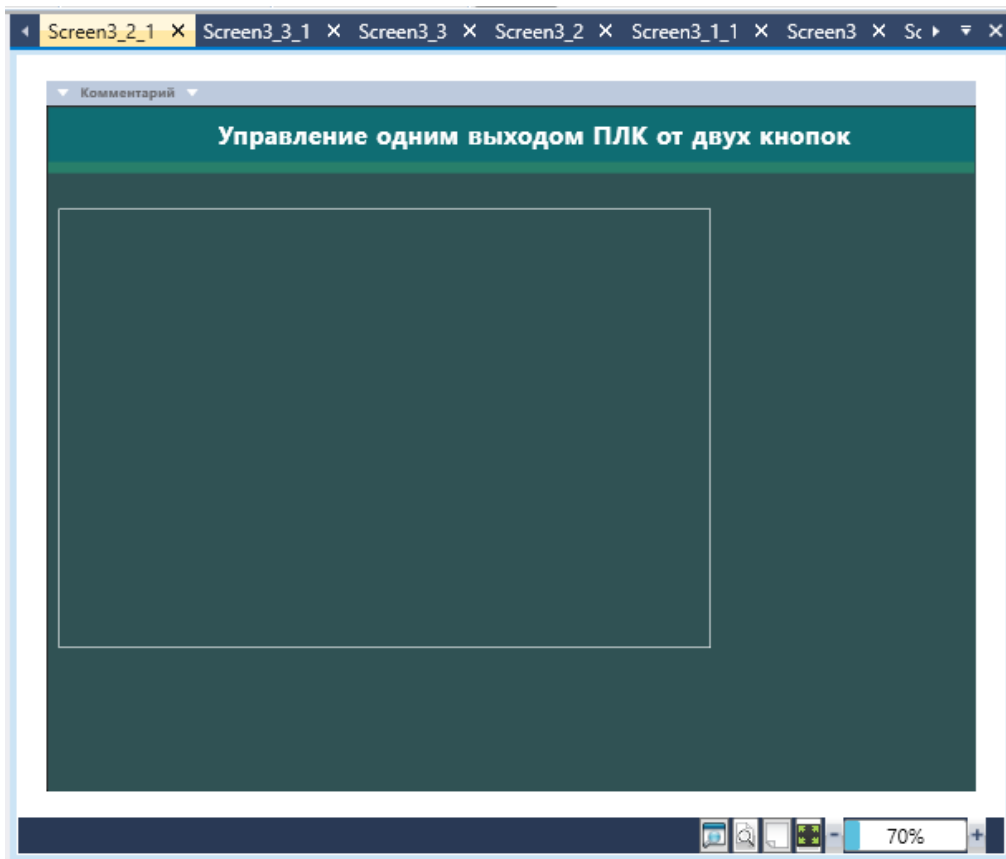


Рис. 118. Создание и редактирование формы № 1

Для формы № 2:

- Свойства внешнего вида элемента Rectangle:

фон: цвет\изображение: Transparent;

- цвет границы: белый;
толщина границы: 1;
- размер и расположение элемента Rectangle:
угол = 0;
ширина = 785;
высота = 86;
позиция по горизонтали = 10;
позиция по вертикали = 504 (рис. 119).

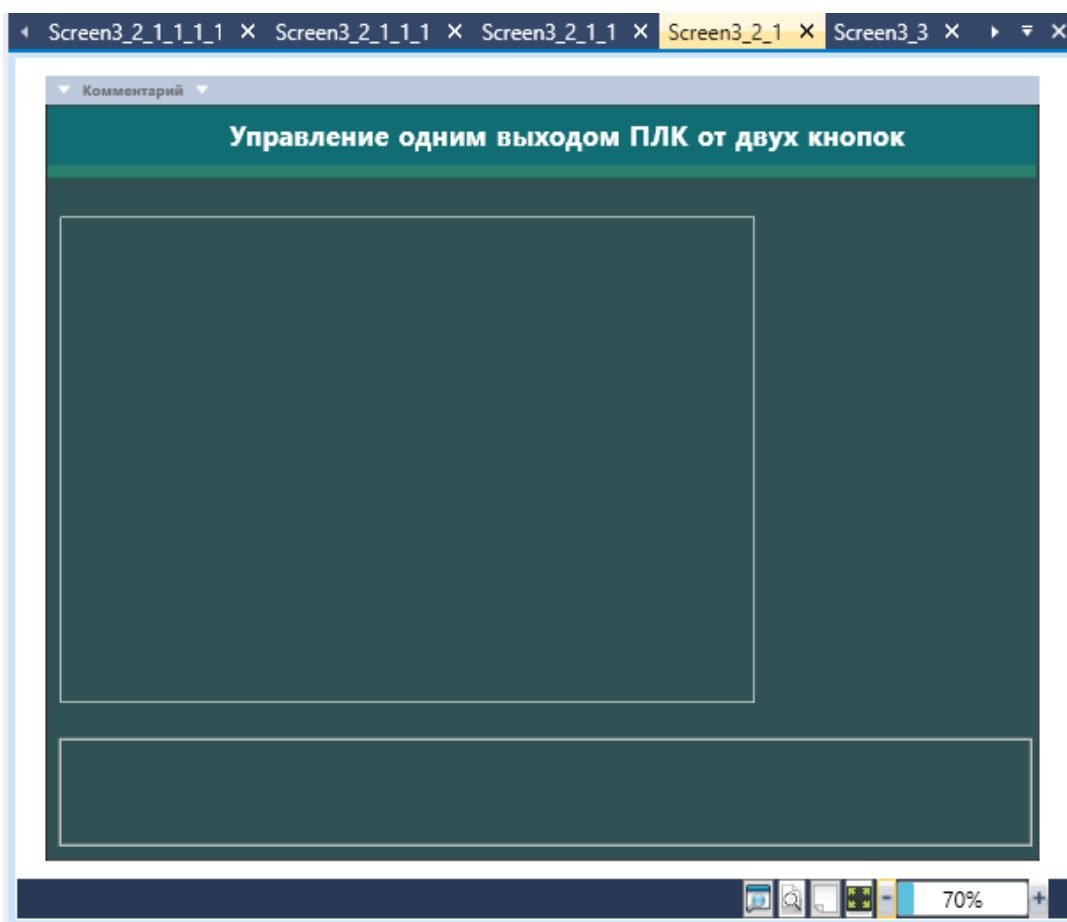


Рис. 119. Создание и редактирование формы № 2

Для формы № 3:

- свойства внешнего вида элемента Rectangle:
фон: цвет\изображение: Transparent;
цвет границы: #FFBFBFBF;
толщина границы: 2;
- размер и расположение элемента Rectangle:
угол = 0;

ширина = 194;
высота = 138;
позиция по горизонтали = 596;
позиция по вертикали = 337 (рис. 120).

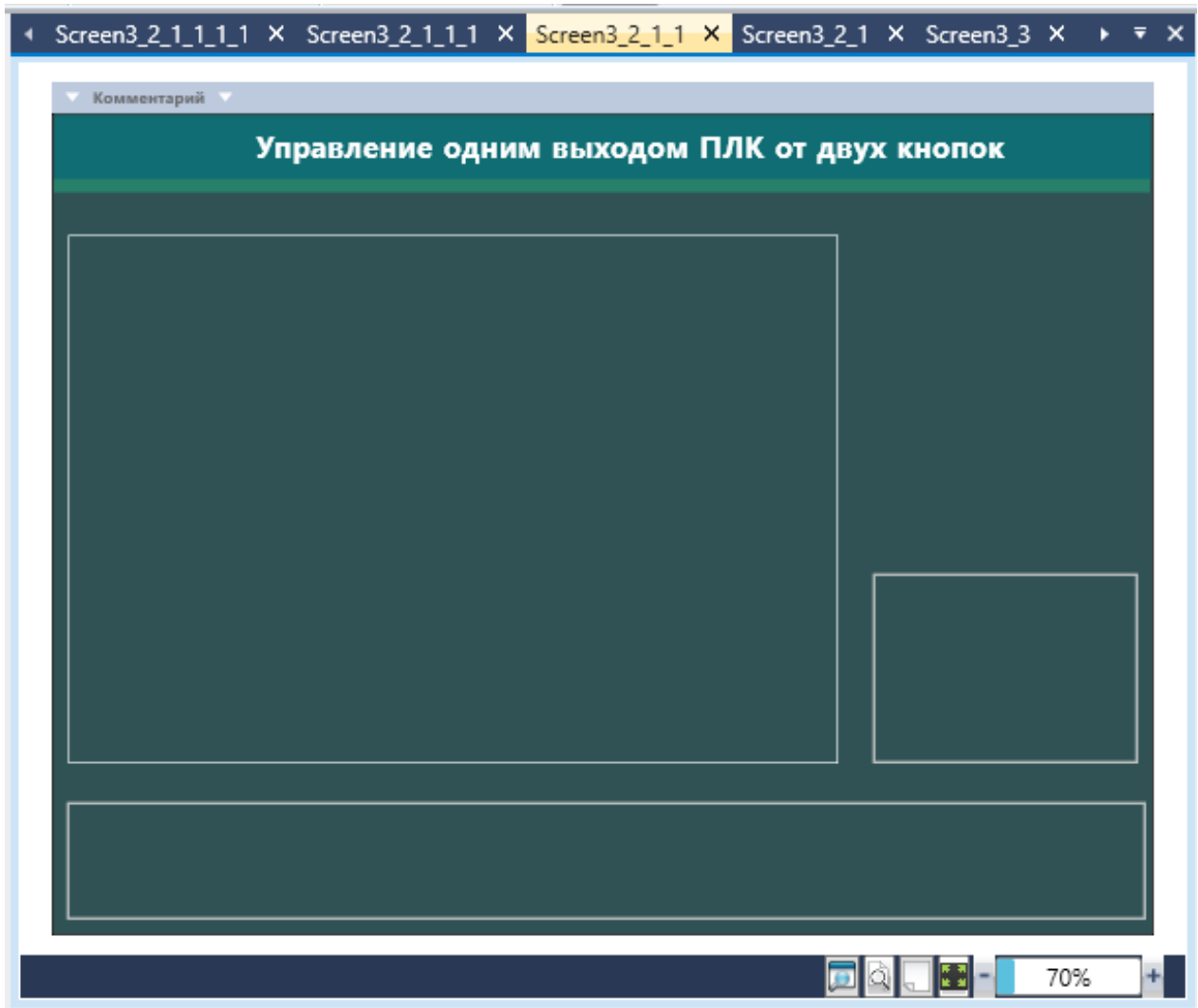


Рис. 120. Создание и редактирование формы № 3

Для формы № 4:

- свойства внешнего вида элемента Rectangle:
фон: цвет\изображение: Transparent;
цвет границы: #FFBFBFBF;
толщина границы: 2;
- размер и расположение элемента Rectangle:
угол = 0;
ширина = 194;

высота = 78;
позиция по горизонтали = 596;
позиция по вертикали = 229 (рис. 121).

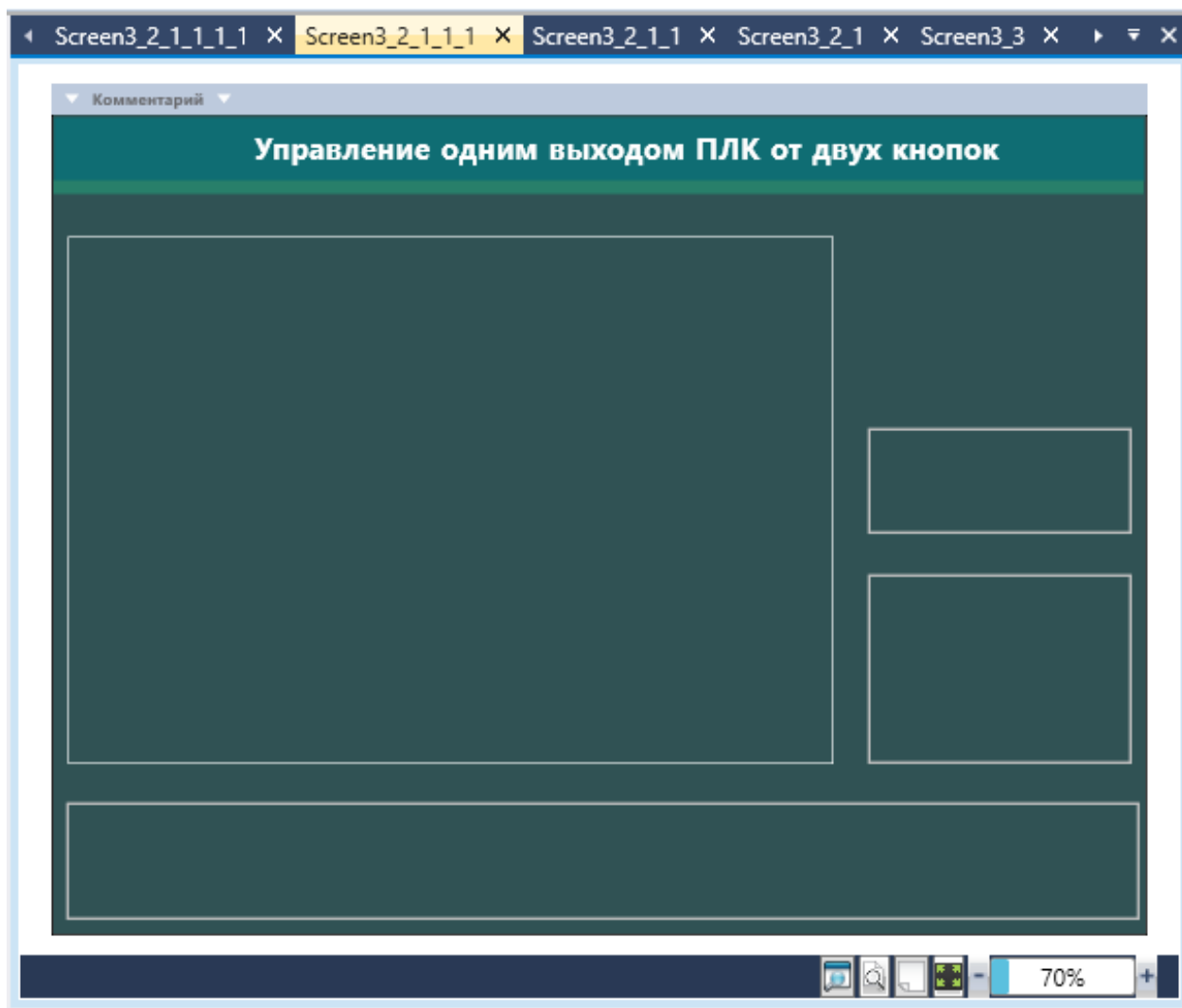


Рис. 121. Создание и редактирование формы № 4

Для формы № 5:

- свойства внешнего вида элемента Rectangle:
фон: цвет\изображение: Transparent;
цвет границы: #FFBFBFBF;
толщина границы: 2;
- размер и расположение элемента Rectangle:
угол = 0;
ширина = 194;
высота = 106;

позиция по горизонтали = 596;
позиция по вертикали = 89 (рис. 122).

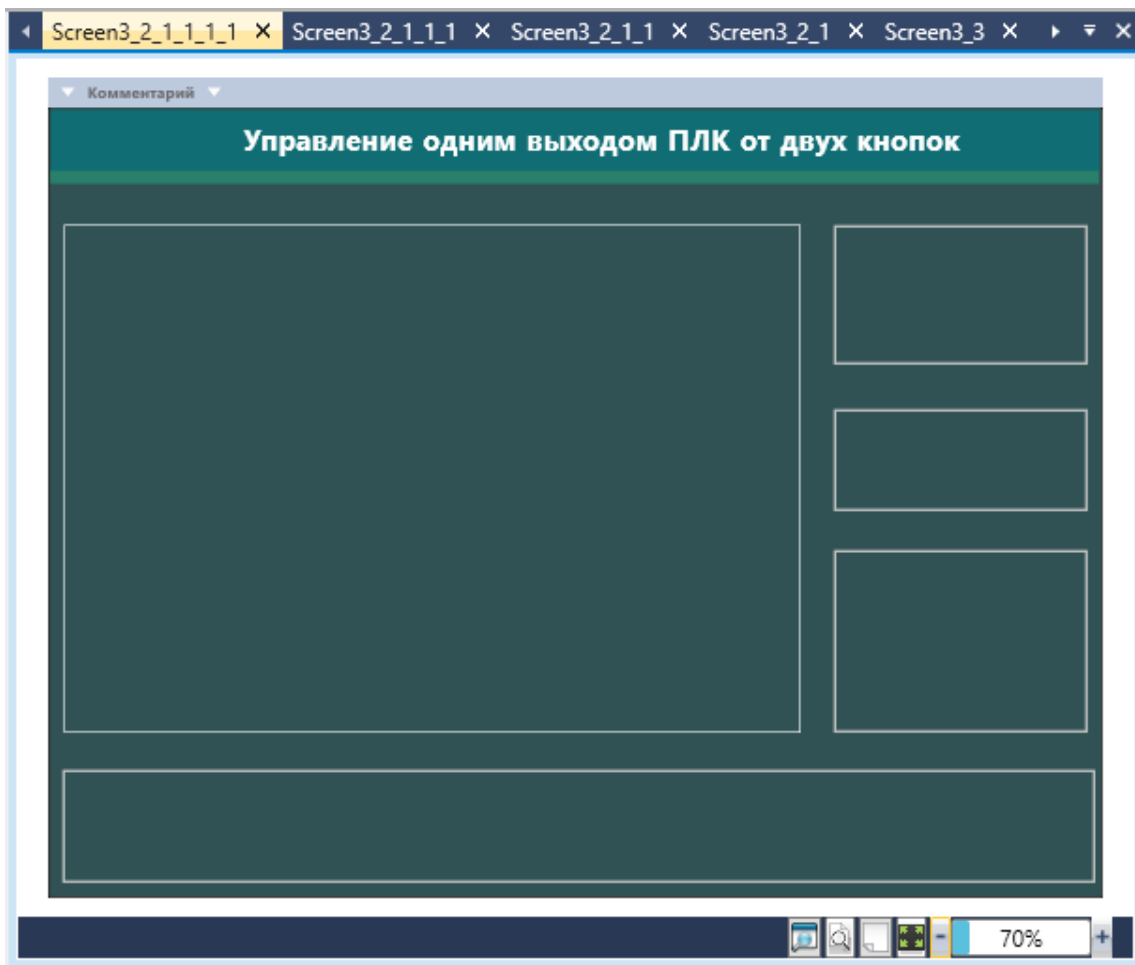


Рис. 122. Создание и редактирование формы № 5

Создаем поясняющие надписи к пяти формам.

Поясняющая надпись к форме № 5:

- источник элемента «Фиксированный текст»:

Текст: «Управление»;

- свойства внешнего вида элемента «Фиксированный текст»:

фон: цвет\изображение: белый;

цвет границы: черный;

толщина границы: 0;

выравнивание: по центру;

- размер и расположение элемента «Фиксированный текст»:

угол = 0;

ширина = 103;

высота = 34;
позиция по горизонтали = 612;
позиция по вертикали = 80 (рис. 123).

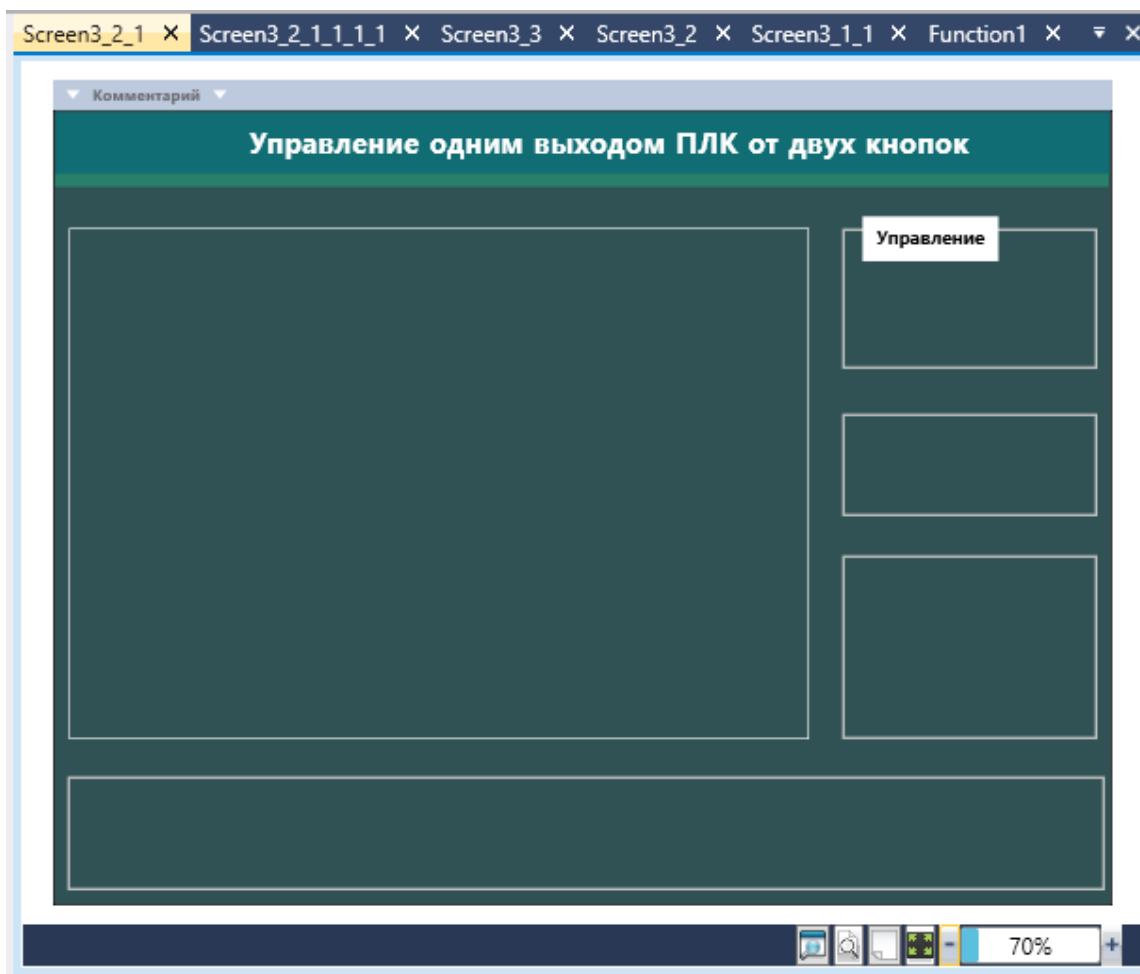


Рис. 123. Создание и редактирование поясняющей надписи к форме № 5

Поясняющая надпись к форме № 4:

- источник элемента «Фиксированный текст»:
текст: «Статус»;
- свойства внешнего вида элемента «Фиксированный текст»:
фон: цвет\изображение: белый;
цвет границы: черный;
толщина границы: 0;
выравнивание: по центру;
размер и расположение элемента «Фиксированный текст»:
угол = 0;
ширина = 103;

высота = 34;
позиция по горизонтали = 612;
позиция по вертикали = 218 (рис. 124).

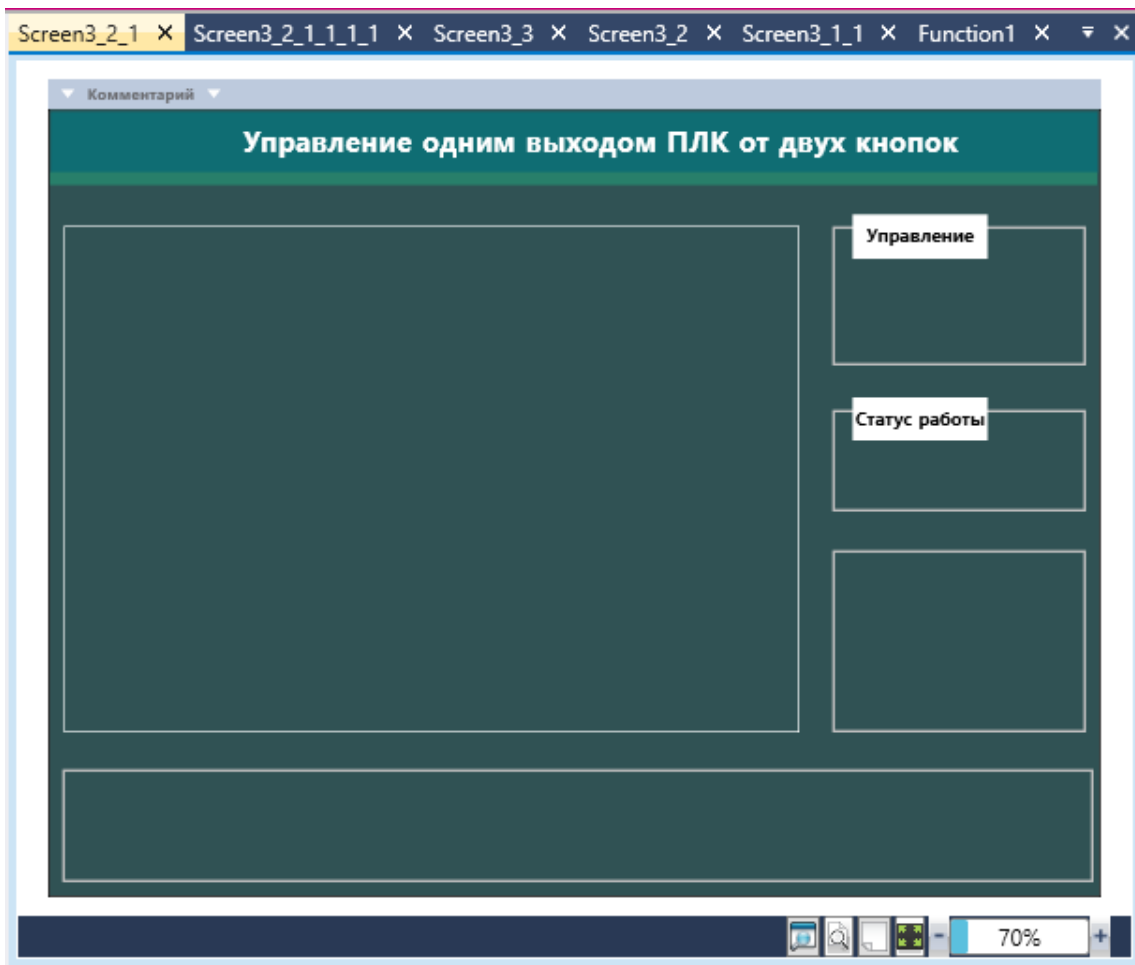


Рис. 124. Создание и редактирование поясняющей надписи к форме № 4

Поясняющая надпись к форме № 3:

- источник элемента «Фиксированный текст»:
текст: «...»;
- свойства внешнего вида элемента «Фиксированный текст»:
фон: цвет\изображение: белый;
цвет границы: черный;
толщина границы: 0;
выравнивание: по центру;
- размер и расположение элемента «Фиксированный текст»:
угол = 0;
ширина = 103;

высота = 34;
позиция по горизонтали = 612;
позиция по вертикали = 327 (рис. 125).

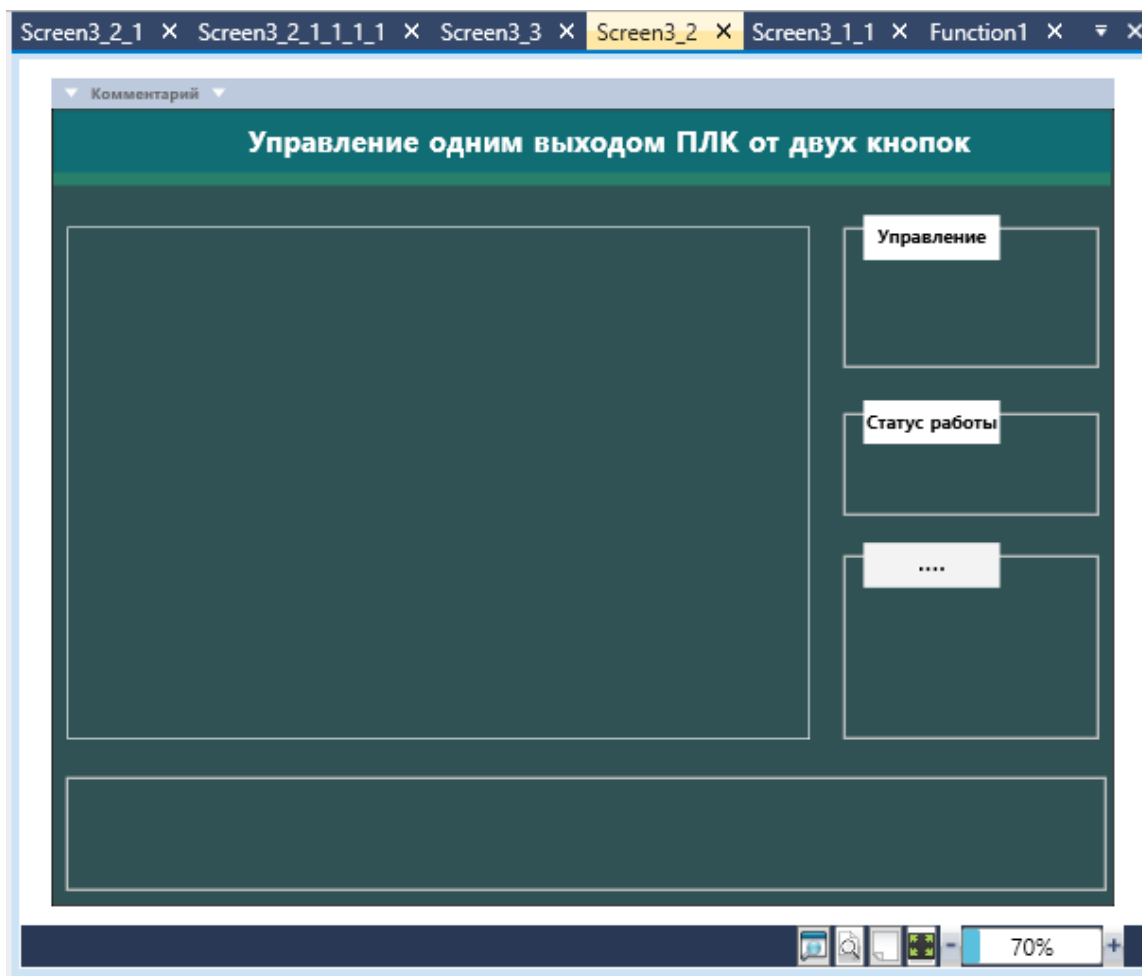


Рис. 125. Создание и редактирование поясняющей надписи к форме № 3

Создаем в форме № 5 кнопку «Старт». Для этого выбираем атрибуты элемента в нем:

- Button:
шрифт: Bitstream Vera Sans, 18;
текстовая метка: «Старт»;
- свойства внешнего вида элемента Button:
цвет заливки фона:
R = 15, G = 109, B = 115;
цвет границы: Transparent;
стиль: Button1;
выравнивание: по центру;

- размер и расположение элемента «Фиксированный текст»:
 угол = 0;
 ширина = 69;
 высота = 36;
 позиция по горизонтали = 618;
 позиция по вертикали = 130 (рис. 126).

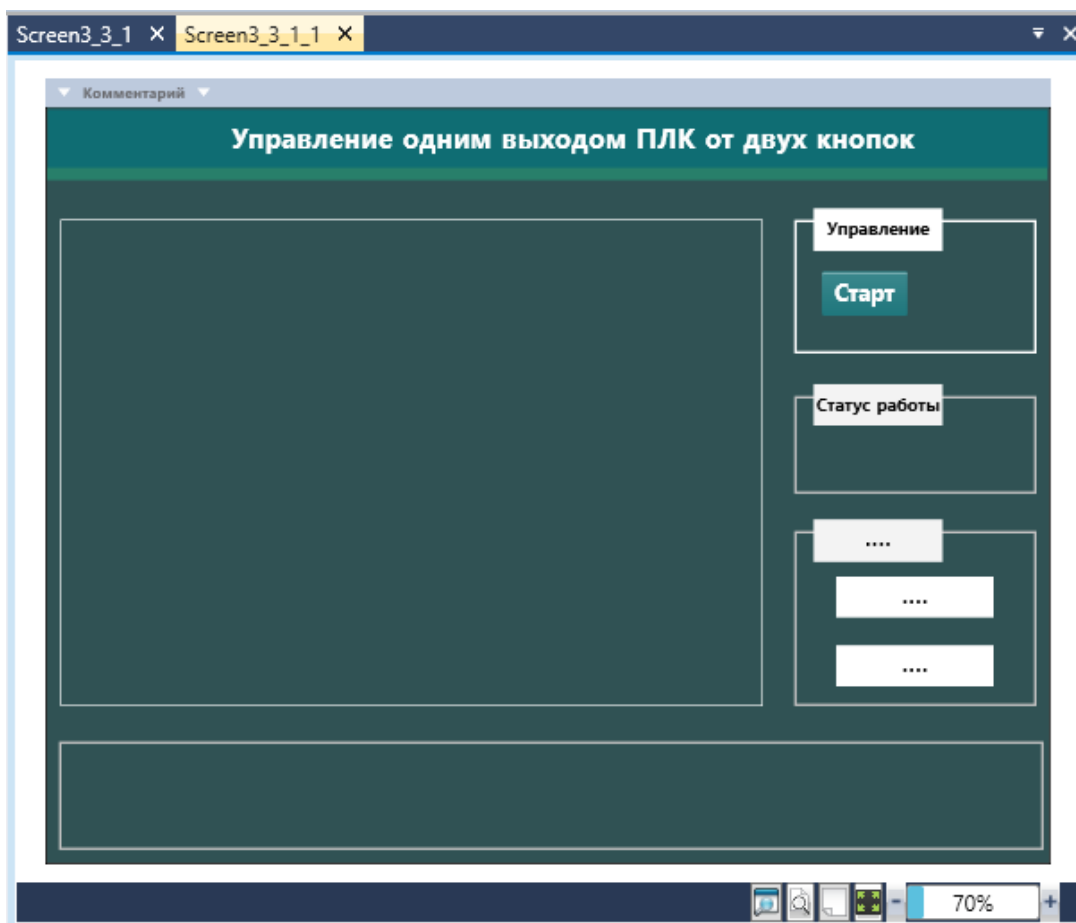


Рис. 126. Создание и редактирование кнопки «Старт»

Создаем в форме № 5 кнопку «Стоп». Для этого выбираем атрибуты элемента в нем:

- Button:
 шрифт: Bitstream Vera Sans, 18;
 текстовая метка: «Стоп»;
- свойства внешнего вида элемента Button:
 цвет заливки фона:
 R = 15, G = 109, B = 115;
 цвет границы: Transparent;

- стиль: Button1;
выравнивание: по центру;
- размер и расположение элемента «Фиксированный текст»:
угол = 0;
ширина = 69;
высота = 36;
позиция по горизонтали = 707;
позиция по вертикали = 130 (рис. 127).

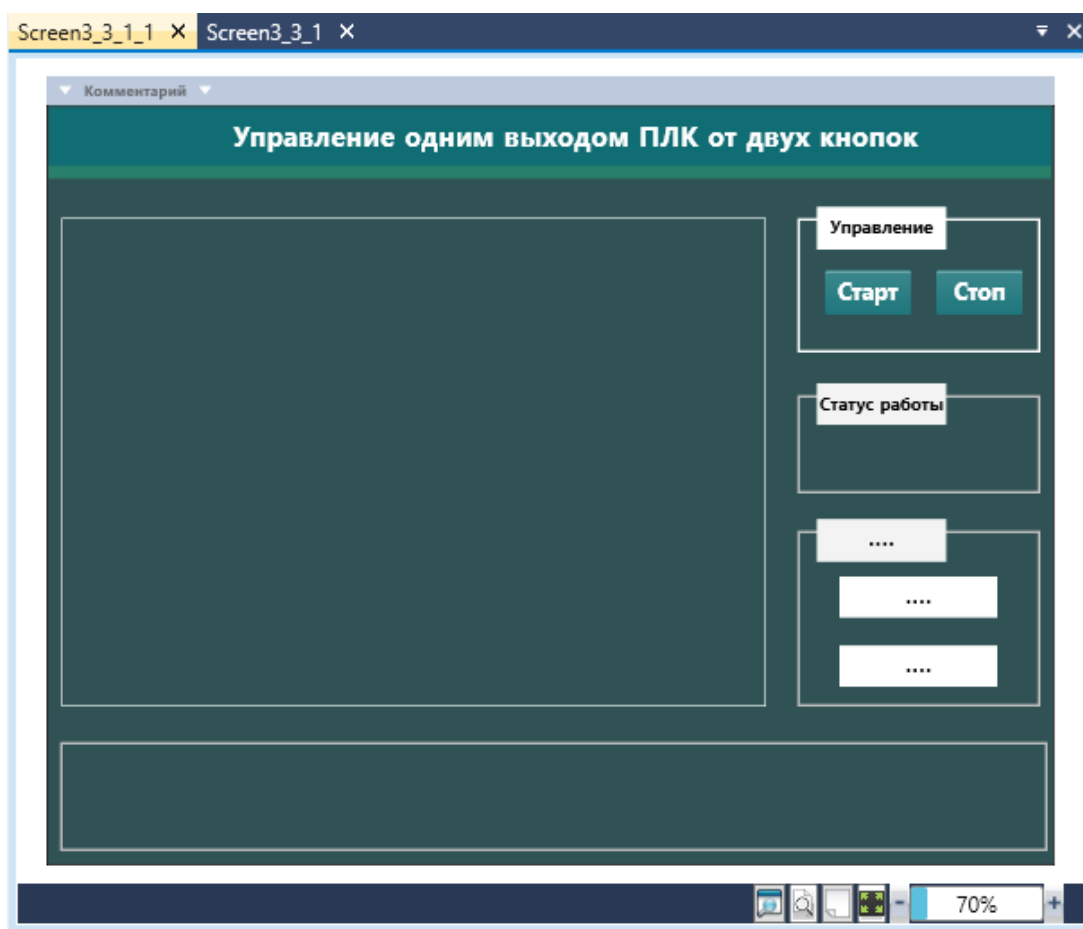


Рис. 127. Создание и редактирование кнопки «Стоп»

Для того чтобы в форме № 4 показывался статус работы, необходимо его проиндексировать. Для этого выбираем свойства внешнего вида элемента в нем:

- двоичная текстовая переменная:
цвет границы: Transparent;
толщина границы: 0;
выравнивание: по центру;

- размер и расположение элемента «Фиксированный текст»:
 - угол = 0;
 - ширина = 126;
 - высота = 33;
 - позиция по горизонтали = 630;
 - позиция по вертикали = 264 (рис. 128).

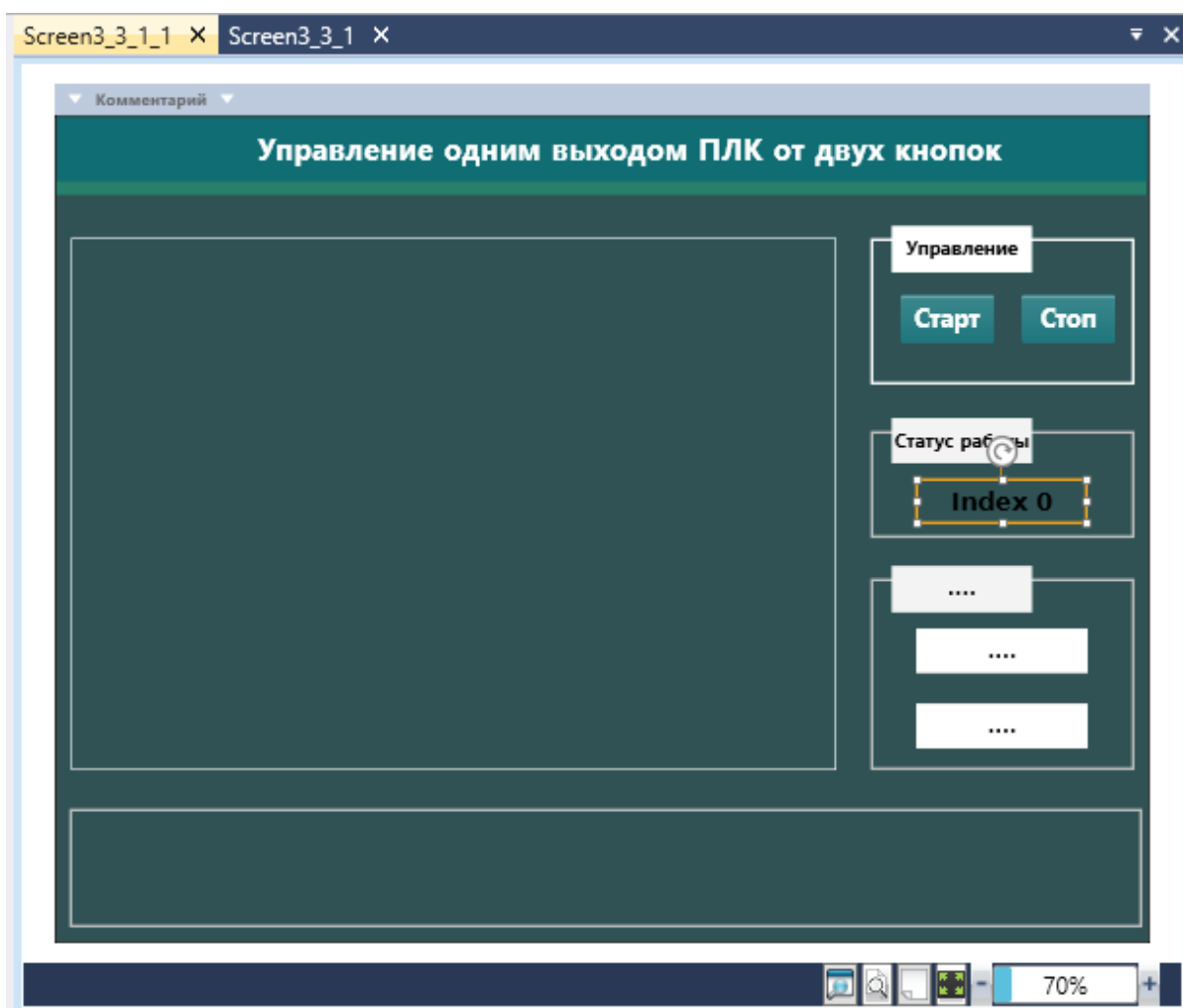


Рис. 128. Индексация статуса работы

В разделе «Источник текста» указываем текст для логической операции 1 и 0.

Например:

0 (ВЫКЛ) – Остановлен;

Цвет – (белый/серый);

1 (ВКЛ) – В работе;

Цвет: R = 35, G = 232, B = 124 (рис. 129).

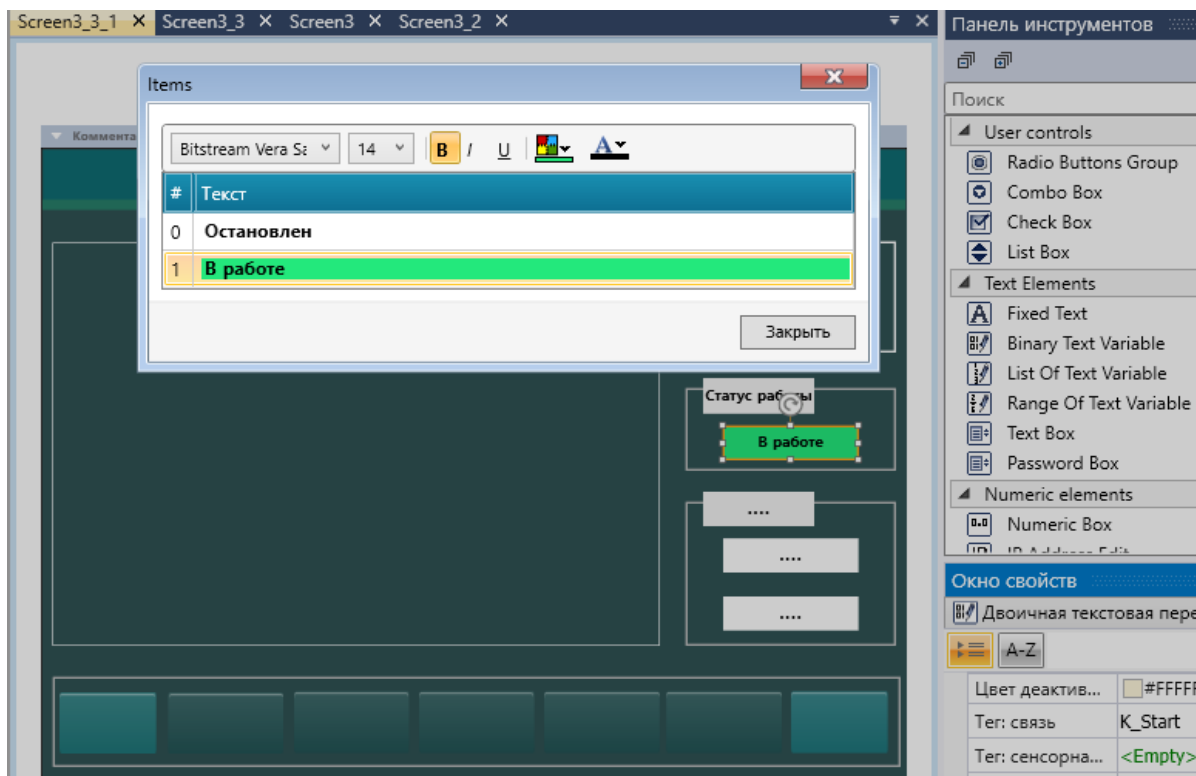


Рис. 129. Настройка текста для индексации статуса работы

Привязываем теги к кнопкам «Старт» и «Стоп». Для этого в атрибутах элемента в Button ЛКМ нажимаем на «Коллекция», добавляем два новых действия и редактируем их (рис. 130).

Действие:

- 1) установить бит;
- 2) сброс бита.

Тег/экран/файл:

- 1) название тега кнопки старт/стоп;
- 2) название тега кнопки старт/стоп.

Триггер:

- 1) нажато;
- 2) нажато.

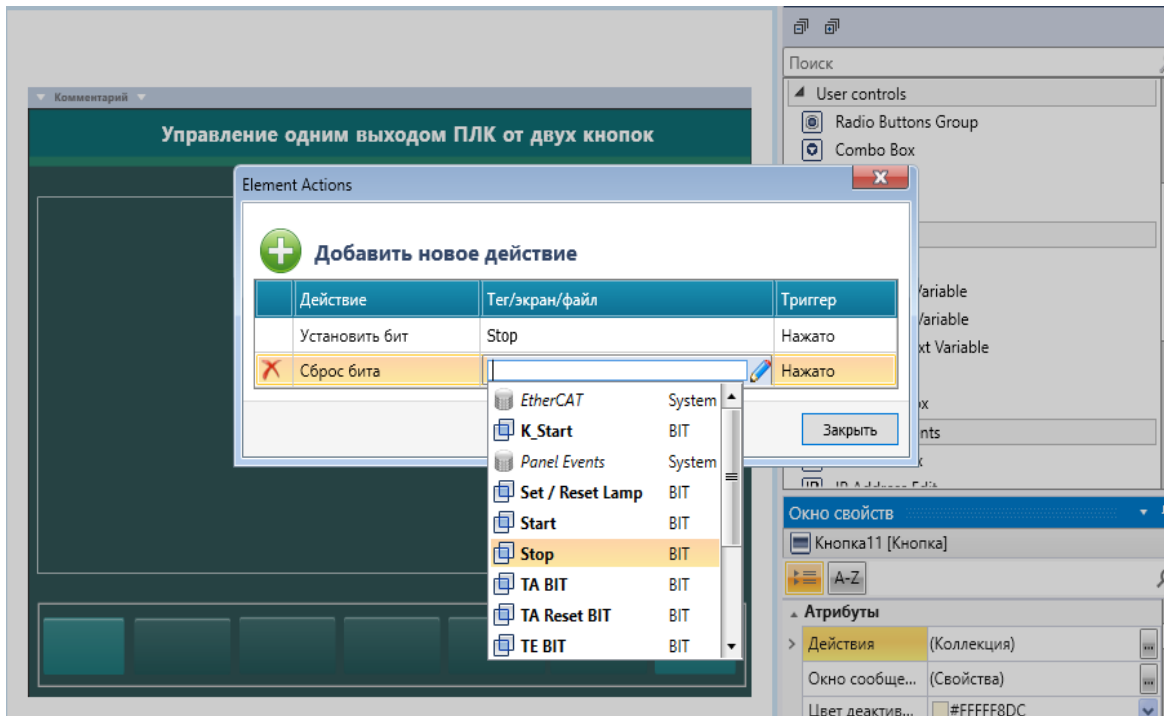


Рис. 130. Привязка тегов к кнопкам «Старт» и «Стоп»

Добавляем изображения на экран HMI (рис. 131).

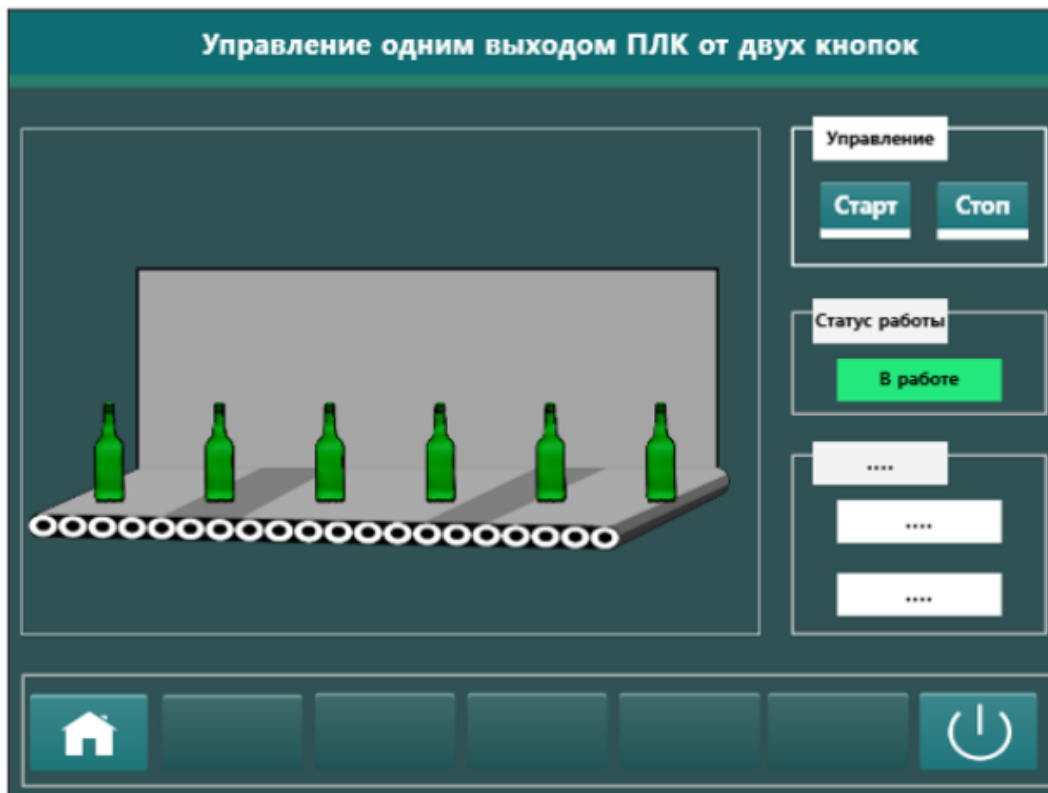


Рис. 131. Добавление изображений на экран HMI

Содержание работы:

- 1) цель работы;
- 2) задания;
- 3) ход работы;
- 4) выводы;
- 5) средства, используемые при выполнении лабораторной работы.

Средства, используемые при выполнении лабораторной работы:

- 1) методические указания к выполнению лабораторных работ;
- 2) данные, предоставленные преподавателем во время занятия;
- 3) при проведении анализа допускается использование глобальной сети Интернет.

ЗАКЛЮЧЕНИЕ

В лабораторном практикуме рассмотрены вопросы, связанные с практической схемотехникой, программированием логических контроллеров UniLogic и разработкой человеко-машинных интерфейсов.

Наиболее подробно представлено программирование ПЛК и интерфейсов на примере стеклоформовочного оборудования.

В издании описаны виды промышленных и локальных сетей, топологии соединений, сети Modbus, типы аналоговых сигналов для измерения показаний с датчиков, статические и динамические элементы мнемосхем, а также представлены задачи, требования, структура и архитектура компьютерных систем управления.

Значительное внимание в лабораторном практикуме уделено условно-графическим обозначениям разъемов, клемм, контактов, аккумуляторов, батарей, насосов, моторов, выключателей, переключателей, гидравлических элементов.

Издание призвано помочь студентам приобрести навыки использования языка программирования LD, настройки программируемого логического контроллера, а также навыки программирования логических элементов, таймеров, счетчиков и разработки человеко-машинного интерфейса.

СПИСОК БИБЛИОГРАФИЧЕСКИХ ИСТОЧНИКОВ

1. Автоматизированное проектирование технологических процессов механической обработки заготовок на станках с ЧПУ : учеб. пособие / Ю. И. Самсонов [и др.] ; Ульянов. гос. техн. ун-т. – Ульяновск : Изд-во УлГТУ, 2000. – 84 с. – ISBN 5-89146-132-3.
2. Денисов, М. С. Системы числового программного управления : лаб. практикум / М. С. Денисов ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2021. – 112 с. – ISBN 978-5-9984-1412-1.
3. Нильсон, Н. Принципы искусственного интеллекта / Н. Нильсон. – М. : Мир, 1985. – 374 с.
4. Норенков, И. П. Информационная поддержка наукоемких изделий. CALS-технологии / И. П. Норенков, П. К. Кузьмик. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2002. – 320 с. – ISBN 5-7038-1962-8.
5. Рассказчиков, Н. Г. Компьютерные системы управления : учеб. пособие / Н. Г. Рассказчиков ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2010. – 155 с. – ISBN 978-5-9984-0017-9.

РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Беккер, В. Ф. Технические средства автоматизации. Интерфейсные устройства и микропроцессорные средства : учеб. пособие / В. Ф. Беккер. – 2-е изд., стер. – М. : РИОР : ИНФРА-М, 2023. – 152 с. – ISBN 978-5-369-01198-0.
2. Варфоломеева, Т. Н. Структуры данных и основные алгоритмы их обработки : учеб. пособие / Т. Н. Варфоломеева. – 2-е изд., стер. – М. : ФЛИНТА, 2023. – 159 с. – ISBN 978-5-9765-3691-3.
3. Денисов, М. С. Компьютерные системы управления: лаб. практикум / М. С. Денисов, И. В. Румянцев, П. А. Чеботарев ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2024. – 132 с. – ISBN 978-5-9984-1976-8.
4. Иванов, В. К. Управление, моделирование и датчики мехатронных систем : учеб. пособие / В. К. Иванов. – М. ; Вологда : Инфра-Инженерия, 2025. – 212 с. – ISBN 978-5-9729-2590-2.
5. Шишов, О. В. Технические средства автоматизации и управления : учеб. пособие / О. В. Шишов. – М. : ИНФРА-М, 2024. – 396 с. – ISBN 978-5-16-019970-2.

Учебное электронное издание

ДЕНИСОВ Максим Сергеевич
ЧЕБОТАРЕВ Петр Александрович
ДАВЫДОВ Кирилл Евгеньевич

ОСНОВЫ ПРОГРАММИРОВАНИЯ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ
В ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ

Лабораторный практикум

Редактор Е. А. Платонова
Технические редакторы Ш. Ш. Амирсейидов, Н. В. Пустовойтова
Компьютерная верстка П. А. Некрасова
Корректор Н. В. Пустовойтова
Выпускающий редактор А. А. Амирсейидова

Системные требования: Intel от 1,3 ГГц; Windows XP/7/8/10; Adobe Reader;
дисковод CD-ROM.

Тираж 9 экз.

Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых.
600000, Владимир, ул. Горького, 87.