

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»

Д.Ю. Поленов

Автоматизированные информационно-управляющие системы

Учебно-методическое пособие



Москва
ИЗДАТЕЛЬСТВО
МГТУ им. Н.Э. Баумана
2023

УДК 681.5
ББК 32.81
П49

Издание доступно в электронном виде по адресу
<https://press.bmstu.ru/catalog/item/8107/>

Факультет «Космический»
Кафедра «Системы автоматического управления»

*Рекомендовано Научно-методическим советом
МГТУ им. Н.Э. Баумана в качестве учебно-методического пособия*

Рецензент
канд. техн. наук доцент *Н.Г. Поялков*

Поленов, Д. Ю.

П49 Автоматизированные информационно-управляющие системы : учебно-методическое пособие / Д. Ю. Поленов. — Москва : Издательство МГТУ им. Н. Э. Баумана, 2023. — 66, [4] с. : ил.

ISBN 978-5-7038-6178-3

Дано описание лабораторных работ, выполняемых при изучении курса «Автоматизированные информационно-управляющие системы». Приведены общая методика цикла лабораторных работ, теоретические сведения и вопросы для подготовки к их защите. Представленный материал поможет студентам научиться разрабатывать автоматизированные информационно-управляющие системы, интерфейс оператора, подключать к ним исполнительные устройства.

Для студентов Космического факультета Мытищинского филиала МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки «Управление в технических системах» (уровень бакалавриата). Может быть использовано студентами инженерных факультетов других высших учебных заведений.

УДК 681.5
ББК 32.81



Уважаемые читатели! Пожелания, предложения, а также сообщения о замеченных опечатках и неточностях Издательство просит направлять по электронной почте: info@bmstu.press

ISBN 978-5-7038-6178-3

© МГТУ им. Н.Э. Баумана, 2023
© Оформление. Издательство
МГТУ им. Н.Э. Баумана, 2023

Предисловие

Дисциплина «Автоматизированные информационно-управляющие системы» входит в учебный план основных образовательных программ подготовки студентов Космического факультета Мытищинского филиала МГТУ им. Н.Э. Баумана по направлениям подготовки 27.03.04 «Управление в технических системах» (уровень бакалавриата) и состоит из одного модуля, изучаемого в течение одного семестра. Студенты осваивают теоретический материал и выполняют лабораторные работы, способствующие получению практических навыков в разработке автоматизированных информационно-управляющих систем.

В целях углубления и закрепления теоретических знаний, полученных студентами на лекциях и в процессе самостоятельной работы, в составе данной дисциплины предусмотрен цикл лабораторных работ. Методика проведения лабораторных работ и контроль качества усвоения учебного материала в значительной степени определяются тем, что лабораторные занятия вписаны в электронную образовательную среду обучения студентов.

При выполнении каждой лабораторной работы необходимо заполнить отчет, форма которого представлена в данном учебно-методическом пособии. Процедура защиты лабораторной работы включает в себя демонстрацию работоспособности разработанной модели и ответы на вопросы, примеры которых приведены в конце описания каждой лабораторной работы. Положительной оценкой при защите лабораторной работы является «зачтено».

Пособие предназначено для студентов Космического факультета Мытищинского филиала МГТУ им. Н.Э. Баумана, обучающихся по направлению «Управление в технических системах» (уровень бакалавриата), а также для студентов инженерных факультетов высших учебных заведений.

Список сокращений

АИУС	— автоматизированные информационно-управляющие системы
АРМ	— автоматизированное рабочее место
АСУ ТП	— автоматизированная система управления технологическими процессами
БД	— база данных
ИСПУ	— интегрированная система проектирования и управления
ЛКМ	— левая кнопка мыши
МРВ	— монитор реального времени
ОТ	— отчет тревог
ПК	— персональный компьютер
ПКМ	— правая кнопка мыши
ПЛК	— программируемый логический контроллер
СУБД	— система управления базами данных

Введение

Для реализации интегрированных систем проектирования и управления (ИСПУ) могут быть использованы различные системы и технологии. Одними из наиболее популярных являются SCADA-системы, которые можно комбинировать с интернет-сайтами, мобильными приложениями и создавать смешанные системы.

Назначение SCADA-систем состоит в разработке и обеспечении работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. В ряде случаев SCADA-системы являются частью ИСПУ автоматизированных систем управления технологическими процессами (АСУ ТП), систем мониторинга, автоматизации зданий и т. д. и используются в тех отраслях хозяйства, где необходимо обеспечивать операторский контроль за ходом технологических процессов и производств в режиме реального времени. Данное программное обеспечение устанавливается на персональный компьютер (ПК) или автоматизированное рабочее место (АРМ). Программный код может быть написан на одном из языков программирования либо сгенерирован в среде проектирования.

Основными являются следующие задачи SCADA-систем.

1. Обмен данными с «устройствами связи с объектом» (т. е. с промышленными контроллерами и платами ввода–вывода) в реальном времени через драйверы.
2. Обработка информации в реальном времени.
3. Отображение информации на экране монитора в удобной и понятной для оператора форме.
4. Ведение базы данных в режиме реального времени с технологической информацией.
5. Аварийная сигнализация и управление тревожными сообщениями.
6. Подготовка и генерирование отчетов о ходе технологического процесса.
7. Осуществление сетевого взаимодействия между SCADA-системой и ПК.
8. Обеспечение связи с внешними приложениями (системами управления базами данных, текстовыми процессорами и т. д.).

Рабочее окно SCADA-системы показано на рис. В.1. Оно состоит из двух областей:

- 1) область навигатора проекта;
- 2) рабочая область проекта.

В первой области находится дерево проекта со всеми его элементами. Во второй области отображается «Экран», создаются программы, отображаются элементы, созданные пользователем.

Пиктограммы, которые будут использованы в лабораторных работах, показаны на рис. В.2.

Целью лабораторных работ по дисциплине «Автоматизированные информационно-управляющие системы» является формирование начальных

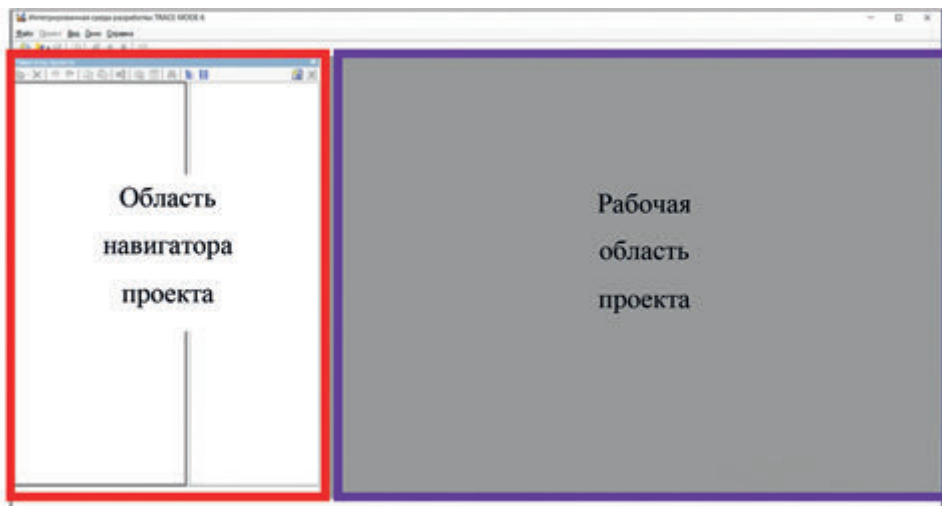


Рис. В.1. Рабочее окно SCADA-системы Trace Mode

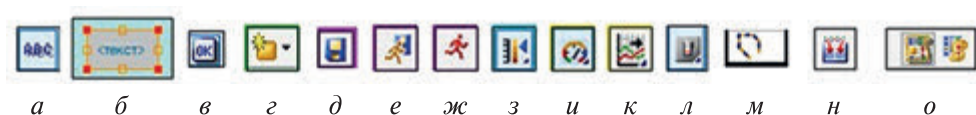


Рис. В.2. Пиктограммы, используемые при выполнении лабораторных работ:

а — элемент «Текст»; *б* — выделение элемента «Текст»; *в* — элемент «Кнопка»; *г* — создать аргумент Экрана; *д* — сохранить; *е* — сохранить для монитора реального времени; *ж* — запустить профайлер; *з* — элемент «Ползунок»; *и* — элемент «Стрелочный прибор»; *к* — элемент «Тренд»; *л* — элемент «Выключатель»; *м* — динамический контур; *н* — компиляция программы; *о* — элемент «Отчет тревог»

практических навыков проведения анализа, исследования технических характеристик объекта управления, технологического процесса и систем.

Для достижения этой цели студенты в процессе занятий должны получить начальные практические навыки по разработке и/или обеспечению работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления.

В результате отработки всех этапов лабораторного цикла студенты должны приобрести знания, умения и навыки, соответствующие компетенциям, установленным основными образовательными программами. После выполнения цикла лабораторных работ студенты должны уметь использовать SCADA-системы для разработки информационно-измерительных и управляющих систем.

Проведение цикла лабораторных работ по дисциплине «Автоматизированные информационно-управляющие системы» предполагает предварительное освоение студентами следующих дисциплин: информатика;

математика; физика; инженерная и компьютерная графика; электротехника и электроника; вычислительные машины, системы и сети; моделирование систем управления; теория автоматического управления; математическая логика и теория автоматов; структуры и алгоритмы обработки данных; системное и прикладное программное обеспечение.

В соответствии с учебным планом на цикл лабораторных работ отводятся 72 часа, из которых 36 часов — на аудиторные (лабораторные) занятия и 36 часов — на самостоятельную работу. Работы относятся к модулю «Основные понятия и принципы проектирования автоматизированных информационно-управляющих систем. Автоматизированные системы управления технологическими процессами в составе АИУС. Автоматизированные системы управления производственными и бизнес-процессами в составе АИУС». Аудиторные занятия, которые проводятся в компьютерном классе, представляют собой девять четырехчасовых лабораторных работ. Отчеты студентов по лабораторным работам планируются в часы занятий.

Лабораторная работа № 1

СОЗДАНИЕ ПРОЕКТА ДИСПЕТЧЕРИЗАЦИИ С ПРИМЕНЕНИЕМ КОМПОНЕНТОВ ВИЗУАЛИЗАЦИИ «КНОПКА», «ПОЛЗУНОК», «ТРЕНД» И «ТЕКСТ»

Цели работы:

- 1) научиться создавать проект и монитор реального времени (МРВ) программного комплекса SCADA-системы;
- 2) научиться создавать Экран и задавать его свойства;
- 3) освоить навыки применения компонентов визуализации «Текст», «Кнопка», «Стрелочный прибор», «Тренд» и «Ползунок»;
- 4) научиться создавать имитаторы входных сигналов разной формы;
- 5) изучить процесс создания и привязки каналов к аргументам Экрана.

План работы

1. Создание проекта.
2. Создание МРВ и Экрана.
3. Создание элемента «Текст».
4. Создание элемента «Кнопка».
5. Создание аргумента Экрана.
6. Создание имитатора синусоидального сигнала.
7. Создание элемента «Стрелочный прибор».
8. Создание элемента «Тренд».
9. Создание элемента «Ползунок».

Создание проекта. В начале работы необходимо выбрать вариант проекта: простой, стандартный или комплексный (рис. 1.1). Эти варианты различаются по своей сложности.

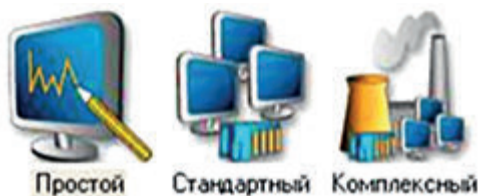


Рис. 1.1. Варианты проекта

Создадим стандартный проект, для чего щелкнем мышью на соответствующей пиктограмме. Открывшееся после этого окно «Навигатор проекта» показано на рис. 1.2.

Создание МРВ и Экрана. Для создания МРВ и Экрана необходимо:

- 1) в окне «Навигатор проекта» щелкнуть правой кнопкой мыши (далее — ПКМ) на пиктограмме «Система», далее выбрать «Создать узел», после чего щелкнуть левой кнопкой мыши (далее — ЛКМ) на строке «RTM» (рис. 1.3). В итоге появится узел RTM_1, соответствующий МРВ, как показано на рис. 1.4;

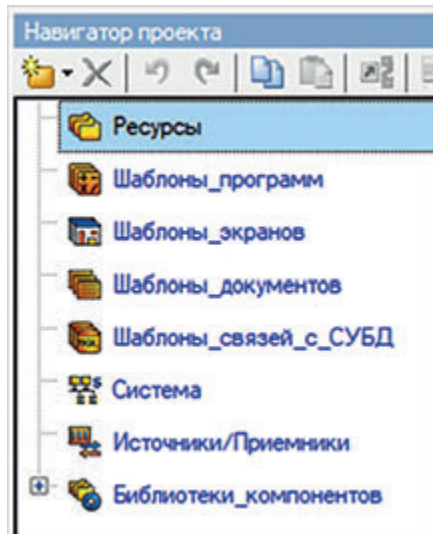


Рис. 1.2. Окно «Навигатор проекта»

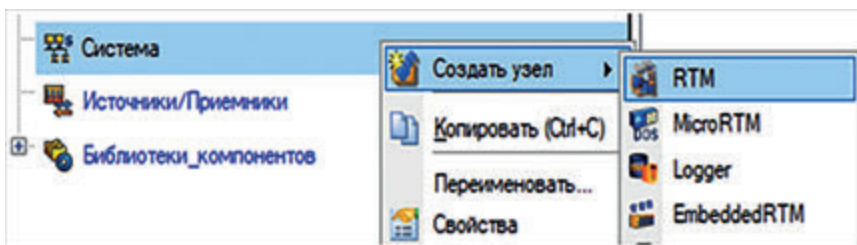


Рис. 1.3. Поэтапное создание МРВ

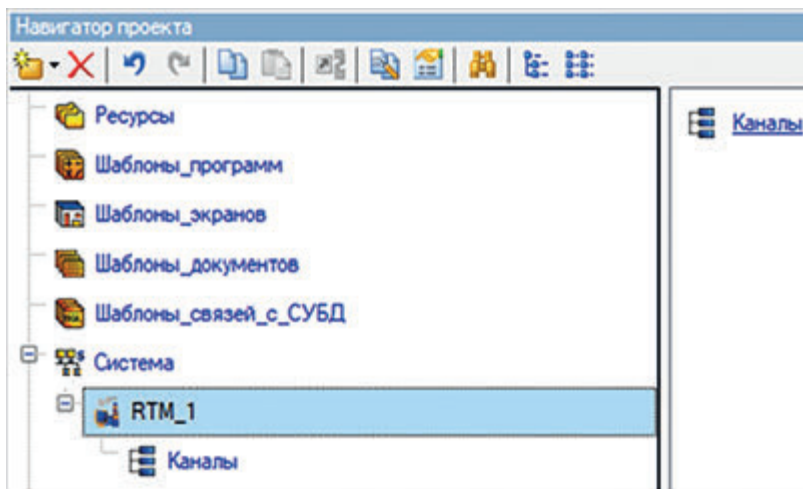


Рис. 1.4. Дерево проекта с созданным МРВ

2) щелкнуть ПКМ на строке RTM_1, выбрать «Создать компонент» и затем «Экран», как показано на рис. 1.5. В итоге будут созданы MPB и Экран, и рядом с группой «Каналы» появится кнопка «Экран#1:1», как показано на рис. 1.6;

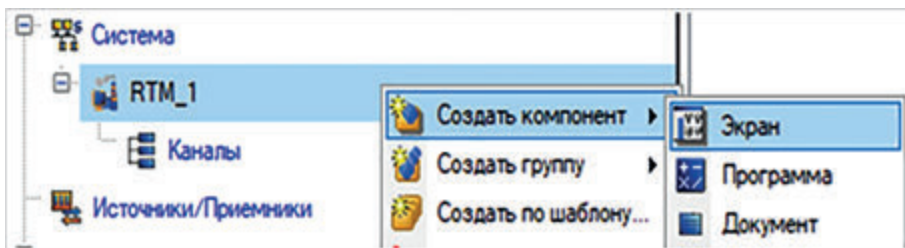


Рис. 1.5. Создание Экрана

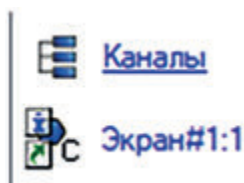



Рис. 1.6. Создание кнопки «Экран#1:1»


3) щелкнуть два раза ЛКМ на кнопке «Экран#1:1», чтобы открыть Экран, затем два раза щелкнуть ЛКМ на свободном месте Экрана, после чего откроется окно «Свойства объекта», показанное на рис. 1.7.

В этом окне можно задавать размеры и фон Экрана, масштабировать его содержимое и т. д. Изменим размер Экрана: вместо 1024x768 зададим 800x600, а в строке «Цвет» выберем в качестве фона голубой, как показано на рис. 1.8.

Создание элемента «Текст». Создадим и разместим на Экране элемент «Текст» и ознакомимся с его свойствами. Для этого на верхней панели на-

жмем ЛКМ на кнопку «Текст»  и, не отпуская ЛКМ, перетащим элемент «Текст» на Экран. Такой метод установки элементов именуется механизмом *drag'n'drop*. На Экране появится созданный элемент «Текст», а в окне «Свойства объекта» откроется список «Свойство», как показано на рис. 1.9.

Обратите внимание на то, что некоторые названия в списке выделены красным цветом и подчеркнуты. Это означает, что данные свойства являются динамическими, т. е. их можно изменять в процессе работы. Например, с помощью свойства «Контур» можно изменять цвет, стиль и толщину линии. Динамическими являются также свойства «Заливка», «Текст», «Цвет текста» и др.

Создание элемента «Кнопка». Для динамического изменения свойств элемента «Текст» добавим на Экран элемент «Кнопка»: нажмем ЛКМ на пиктограмму  на верхней панели и перетащим ее с помощью *drag'n'drop*

Свойства объекта	
Текущий слой: Слой	
Параметры экрана:	
Свойство	Значение
<u>Размеры</u>	1024x768
Масштабировать содержимое	False
<u>Фон</u>	Цвет
<input type="checkbox"/> Цвет	
Положение источника света (%) X	25
Положение источника света (%) Y	25
Код доступа	0x0
Горячая клавиша	Нет
<u>Всплывающее окно</u>	False
Загружать	False
Показать в меню экранов	False
Искать источник событий	False
<u>Верхний объект</u>	False
<u>Нижний объект</u>	False
<u>Левый объект</u>	False
<u>Правый объект</u>	False

Рис. 1.7. Окно «Свойства объекта»



Свойства объекта	
Текущий слой: Слой	
Параметры экрана:	
Свойство	Значение
<u>Размеры</u>	800x600
Масштабировать содержимое	False
<u>Фон</u>	Цвет
<input checked="" type="checkbox"/> Цвет	
Положение источника света (%) X	Основная палитра 
Положение источника света (%) Y	
Код доступа	

Рис. 1.8. Итоговые значения свойств Экрана

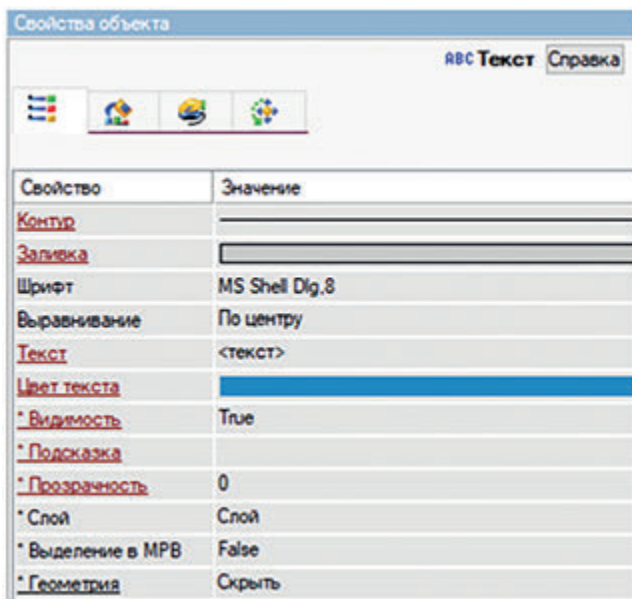


Рис. 1.9. Окно «Свойства объекта» элемента «Текст»

на Экран. В итоге откроется окно, в котором можно задать свойства элемента «Кнопка». Выполним следующие действия:

1) в строке «Текст» окна «Свойства объекта» напишем «Сменить текст», изменим его цвет и размер;

2) в списке «События» щелкнем ПКМ на «MousePress» и выберем строку «Передать значение». В поле «Значение» введем 1, как показано на рис. 1.10.

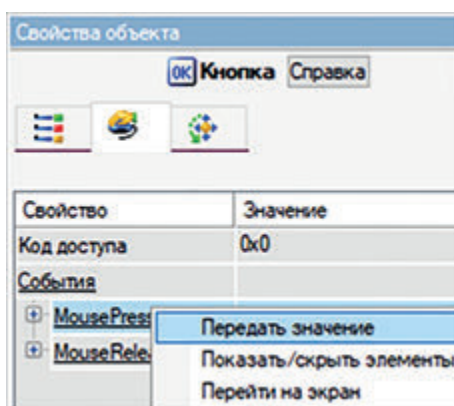


Рис. 1.10. Выбор строки «Передать значение» в окне «Свойства объекта»

Создание аргумента Экрана. Для того чтобы иметь возможность изменять свойства элемента «Текст» по нажатию элемента «Кнопка», необходимо создать один аргумент Экрана. Для этого следует выполнить следующие действия:



1) нажмите на кнопку «Создать аргумент» на нижней панели Экрана (если ее нет, необходимо в меню «Вид» установить соответствующую галочку, как показано на рис. 1.11);

2) присвойте аргументу название: например, «Аргумент_кнопки» (обратите внимание на то, что в названии аргументов должны отсутствовать пробелы);

3) с помощью *drag'n'drop* перетащите созданный аргумент на элемент «Текст»;

4) в свойствах элемента «Кнопка» выберите вкладку «События» и в поле «Результат» добавьте этот же аргумент с помощью *drag'n'drop*;

5) в поле «Восстанавливать значение» выберем «True».

Вернемся к изменению свойств элемента «Текст» и сделаем выбор, показанный на рис. 1.12.

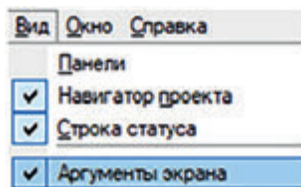


Рис. 1.11. Вызов окна «Аргументы экрана»

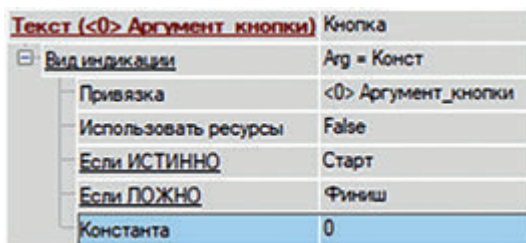


Рис. 1.12. Выбор свойств элемента «Текст» для его динамического изменения

В результате работы с элементами «Текст» и «Кнопка» получим изображения, показанные на рис. 1.13.

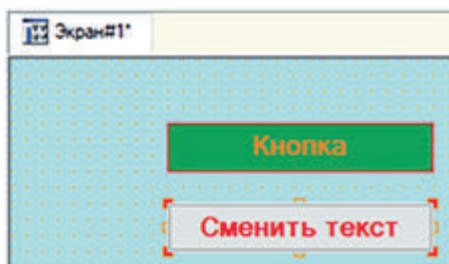





Рис. 1.13. Элементы «Текст» и «Кнопка» в окне Экран#1

Сохраним и запустим созданный проект на исполнение, последовательно нажав следующие кнопки:

- 1) «Сохранить»  на верхней панели;
- 2) «Сохранить для МРВ»  ;
- 3) «Запустить профайлер» , предварительно выбрав RTM_1 в дереве проекта.

Обратите внимание на следующее:

- если кнопка «Сменить текст» в окне профайлера не нажата, в элементе «Текст» написано слово «Старт» (рис. 1.14, а);
- если кнопка «Сменить текст» нажата, в элементе «Текст» написано слово «Финиш» (рис. 1.14, б).

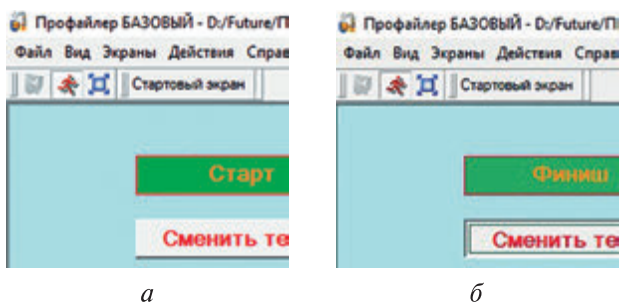


Рис. 1.14. Кнопка «Сменить текст» в исходном (а) и активированном (б) положении

Таким образом выполняется динамическая привязка элементов с помощью созданного аргумента Экрана. Закроем профайлер.

Создание имитатора синусоидального сигнала. Перейдем к созданию имитатора синусоидального сигнала. Для этого щелкнем ПКМ на строке «Источники» дерева проекта, выберем строку «Создать группы», далее — «Генераторы», как показано на рис. 1.15.

Щелкнем ПКМ на созданной группе «Генераторы_1», выберем строку «Создать компонент», далее — «Синусоида», как показано на рис. 1.16.

Теперь необходимо созданный имитатор синусоидального сигнала перенести в каналы МРВ и привязать его к аргументам Экрана. Для этого требуется выполнить следующие действия:

- 1) перетащить пиктограмму созданного имитатора «Синусоида» с помощью *drag'n'drop* в группу «Каналы» узла RTM_1. Результат показан на рис. 1.17;
- 2) добавить аргумент экрана «Синусоида»;
- 3) привязать созданный канал «Синусоида#1» к аргументу Экрана «Синусоида», как показано на рис. 1.18.

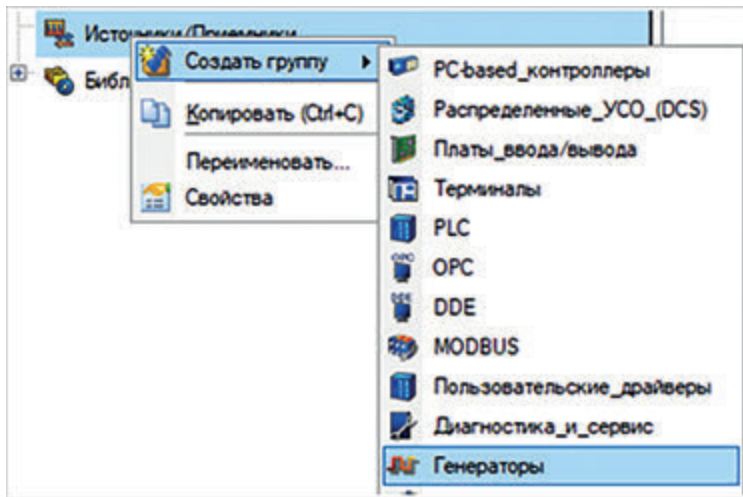


Рис. 1.15. Создание генератора сигналов

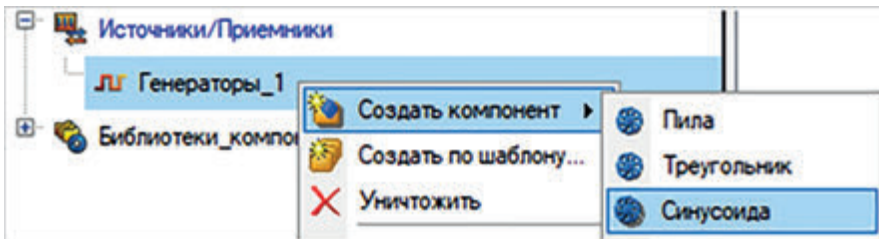


Рис. 1.16. Создание имитатора синусоидального сигнала

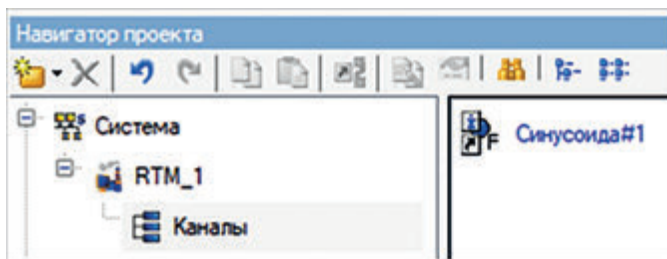



Рис. 1.17. Созданный канал «Синусоида#1» узла RTM_1

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Аргумент_кнопки	↓ IN	REAL		
Синусоида	↓ IN	REAL		Синусоида#1:Реальное значение (Система.RTM_1.Каналы)

Рис. 1.18. Привязка канала «Синусоида#1» к аргументу «Синусоида»

Имитаторы сигналов позволяют подавать различные воздействия на МРВ и тем самым проверять правильность выбранных программных решений.

Создание элемента «Стрелочный прибор». Для того чтобы добавить на Экран элемент «Стрелочный прибор», следует в группе «Приборы» выбрать пиктограмму «Стрелочный прибор»  и с помощью *drag'n'drop* перетащить ее на Экран. Результат показан на рис. 1.19.

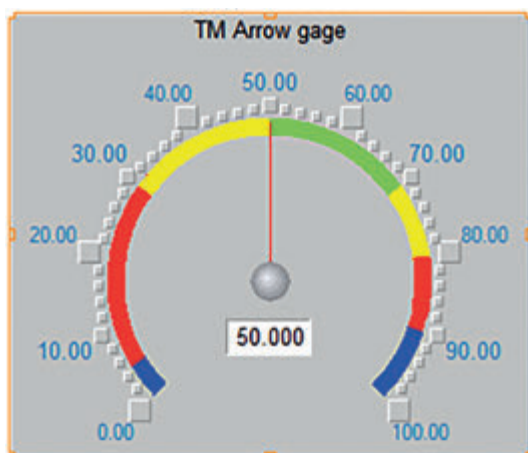


Рис. 1.19. Элемент «Стрелочный прибор»

У данного элемента можно менять стиль отображаемых цифр, границы и цвета интервалов, фон и т. д. Самостоятельно поэкспериментируйте с указанными свойствами.

Привяжем созданный аргумент «Синусоида» к элементу «Стрелочный прибор». Для этого, выделив в окне «Свойства объекта» строку «Отображаемая величина», в поле «Привязка» добавим необходимый аргумент. Результат привязки показан на рис. 1.20.

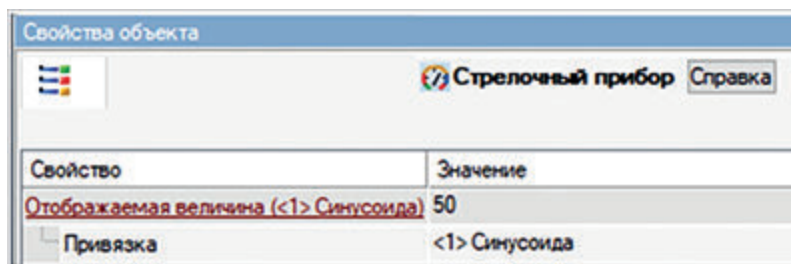




Рис. 1.20. Привязка элемента «Стрелочный прибор» к аргументу «Синусоида» в окне «Свойства объекта»


Сохраним и запустим созданный проект на исполнение, последовательно нажав следующие кнопки на верхней панели:

1) «Сохранить» ;

2) «Сохранить для МРВ» ;

3) «Запустить профайлер» , предварительно выбрав RTM_1 в дереве проекта.

Убедимся в правильности выполненных действий по плавному движению стрелки «Стрелочного прибора». Закроем профайлер.

Создание элемента «Тренд». Добавим на Экран элемент «Тренд» , после чего с помощью *drag'n'drop* перенесем на «Тренд» аргумент Экрана «Синусоида». Результатом будет являться окно, показанное на рис. 1.21.

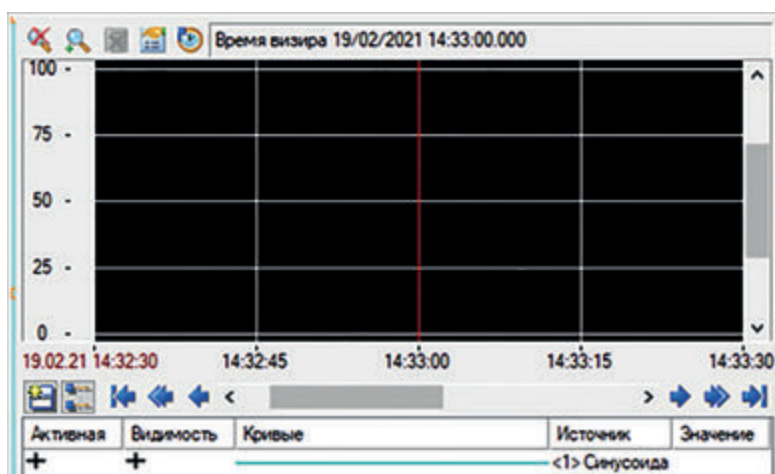



Рис. 1.21. Элемент «Тренд» с добавленным на него аргументом «Синусоида»

Настроим свойства выводимого сигнала. Для этого в окне свойств элемента «Тренд» зайдём в список «Кривые» и активируем строку «Кривая1 (<1> Синусоида)». Зададим цвет и толщину линии выводимого сигнала, как показано на рис. 1.22.

Сохраним и запустим созданный проект на исполнение. Результатом работы должен стать настроенный на элементе «Тренд» синусоидальный сигнал, как показано на рис. 1.23.

Создание элемента «Ползунок». Для создания этого элемента необходимо выбрать пиктограмму «Ползунок»  в группе «Приборы» и с помощью *drag'n'drop* перетащить ее на Экран. Результат показан на рис. 1.24.

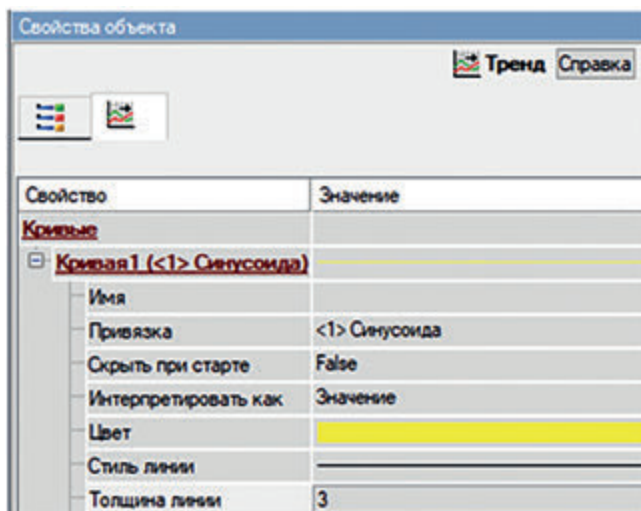


Рис. 1.22. Окно «Свойства объекта» с заданными параметрами линии «Кривая 1» элемента «Тренд»

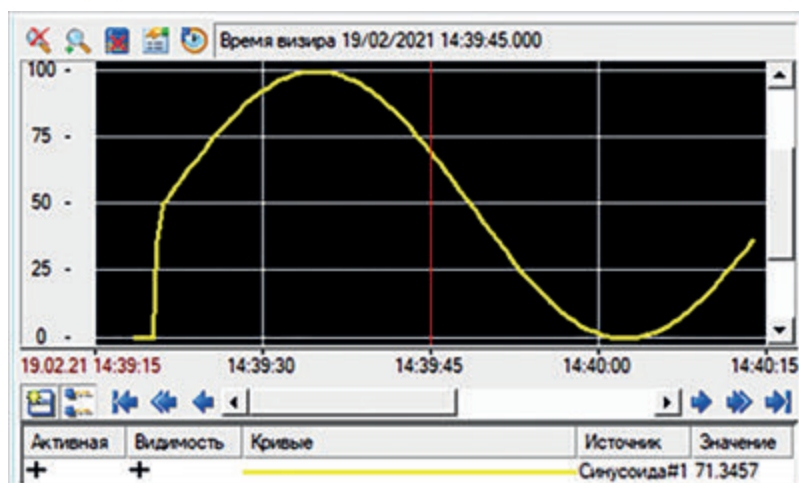


Рис. 1.23. Настроенный синусоидальный сигнал на элементе «Тренд»

Привяжем созданный аргумент «Синусоида» к элементу «Ползунок». Для этого в окне «Свойства объекта» в строках «Отображаемая величина» и «Задаваемая величина» в поля «Привязка» добавим аргумент «Синусоида». Результат показан на рис. 1.25.

Сохраним и запустим созданный проект на исполнение. Убедимся, что элемент «Ползунок» отображает значения синусоидального сигнала.

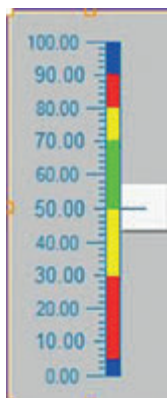


Рис. 1.24. Элемент «Ползунок»

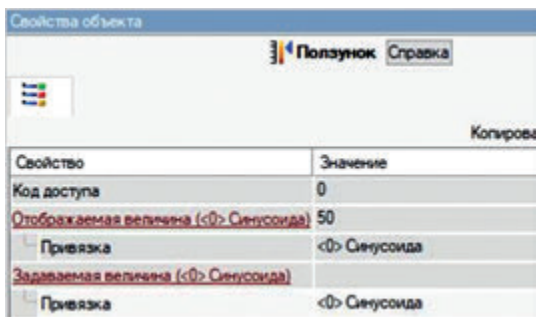


Рис. 1.25. Привязка аргумента «Синусоида» к элементу «Ползунок» в окне «Свойства объекта»

Остановим проект, закроем профайлер. В качестве отчета по лабораторной работе преподавателю предоставляют информацию о созданной АИУС с применением элементов «Кнопка», «Ползунок», «Стрелочный прибор», «Тренд» и «Текст» и о ходе выполнения работы. Студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Задания для самоконтроля и подготовки к защите лабораторной работы № 1

Создайте новый проект и затем на Экране:

- создайте элементы «Кнопка» и «Текст». По нажатию на элемент «Кнопка» должны изменяться цвет фона и текст в элементе «Текст»;
- создайте имитаторы сигналов «Случайное число» и «Пила», привяжите их к вновь созданным элементам «Текст» и «Ползунок» соответственно;
- выведите созданные сигналы на элемент «Тренд», присвойте им названия и измените стиль линии.

Лабораторная работа № 2

СОЗДАНИЕ ГРАФИЧЕСКИХ ОБЪЕКТОВ. ТИРАЖИРОВАНИЕ ИХ НА МОНИТОРЕ РЕАЛЬНОГО ВРЕМЕНИ. ПРИВЯЗКА КАНАЛОВ

Цели работы:

- 1) научиться создавать графические объекты в SCADA-системе;
- 2) освоить работу с группой «Ресурсы» для добавления в проект пользовательских графических объектов;
- 3) освоить тиражирование графических объектов;
- 4) научиться привязывать аргументы графических объектов к существующим аргументам и каналам.

План работы

1. Создание линии.
2. Создание элементов «Выключатель 0» и «Выключатель 1».
3. Создание графических объектов.
4. Тиражирование графических объектов.

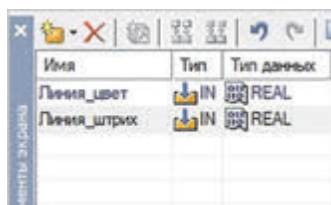



Рис. 2.1. Аргументы Экрана «Линия_цвет» и «Линия_штрих»

Создание линии. В соответствии с алгоритмом, приведенным в лабораторной работе № 1, создадим стандартный проект, узел RTM и Экран, а затем создадим два аргумента Экрана: «Линия_цвет» и «Линия_штрих», как показано на рис. 2.1.

Разместим на Экране элемент «Линия»,

выбрав пиктограмму  на верхней панели и перетащив ее на Экран. В открывшемся окне «Свойства объекта» зададим свойства элемента «Линия», как показано на рис. 2.2:

- в строке «Вид индикации» выберем Arg = Конст;
- сделаем привязку к аргументу «Линия_цвет»;
- установим значение Константы, равное 1;
- зададим толщину, равную 10.

Указанные действия подразумевают смену цвета линии с зеленого на красный в соответствии со значением аргумента «Цвет_линии».

Создание элементов «Выключатель 0» и «Выключатель 1». Для проверки работоспособности созданного элемента добавим на Экран элемент «Вы-

ключатель 0»  и зададим его свойства, как показано на рис. 2.3.

Проверим работоспособность созданного элемента, правильность привязки каналов и заданных свойств. Для этого сохраним и запустим проект на исполнение. На Экране должно быть видно, как при установке «Выключателя 0» в положение «ВЫКЛ» цвет линии становится красным (рис. 2.4, а), при положении «ВКЛ» — зеленым (рис. 2.4, б). Остановим проект, закроем профайлер.

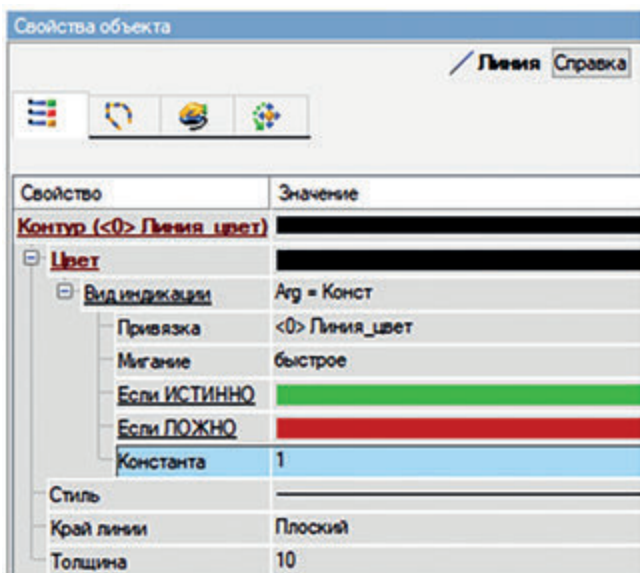


Рис. 2.2. Свойства элемента «Линия»

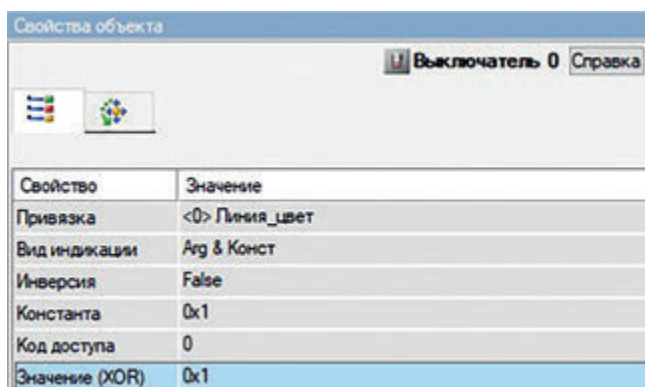


Рис. 2.3. Установка свойств элемента «Выключатель 0»



Рис. 2.4. Демонстрация работоспособности динамической привязки элемента «Выключатель 0» к элементу «Линия»

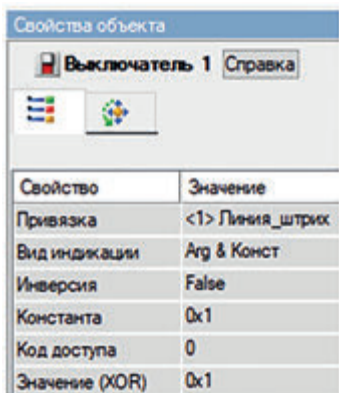
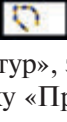


Рис. 2.5. Установка свойств элемента «Выключатель 1»

Теперь ознакомимся со свойством «Динамический контур» элемента «Линия». Для этого с помощью кнопки  перейдем во вкладку «Динамический контур», затем в списке «Свойство» выберем строку «Привязка», далее в списке «Значение» — аргумент «Линия_штрих». Добавим на Экран новый элемент «Выключатель 1» и для параметров в списке «Свойство» установим значения, как показано на рис. 2.5.

Проверим работоспособность созданных элементов, правильность привязки каналов и заданных свойств. Для этого сохраним и запустим созданный проект на исполнение. На Экране должно быть видно, как при установке «Выключателя 1» в положение «ВЫКЛ» цвет контура линии становится серым (рис. 2.6, а), а при положении «ВКЛ» появляются красные двигающиеся штрихи (рис. 2.6, б). Остановим проект, закроем профайлер.



Рис. 2.6. Демонстрация работоспособности динамической привязки выключателя и контура линии

Создание графических объектов. Для создания графических объектов щелкнем ПКМ на строке «Ресурсы» дерева проекта, выберем строку «Создать группу», далее — «Графические элементы», как показано на рис. 2.7.

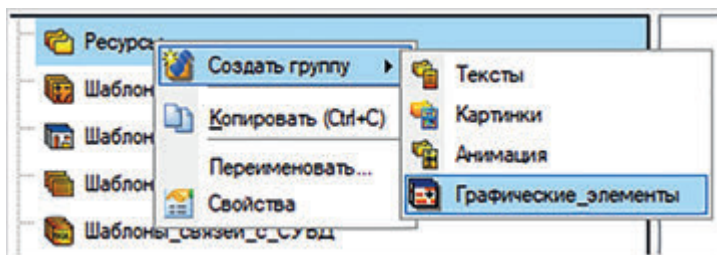


Рис. 2.7. Процесс создания группы для графических элементов

После этого щелкнем ПКМ на пиктограмме созданной группы, выберем строку «Создать компонент», далее — «Графический объект», как показано на рис. 2.8. Будет создано окно графического объекта.

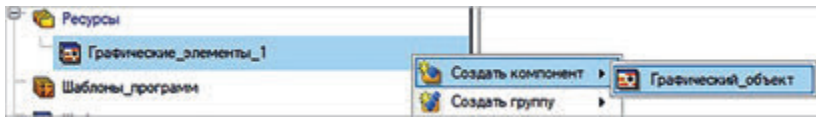


Рис. 2.8. Процесс создания графического объекта

Перейдем в окно «Графический_объект_1», дважды щелкнув на нем ЛКМ, откроем окно свойств, аналогичное окну свойств Экрана узла RTM. Увеличим разрешение окна. Добавим в окно с помощью пиктограмм на верхней панели элементы «Трубопровод», «Стрелочный прибор» и «Линия». Создадим аргументы «Температура» и «Подача_воды» и привяжем их к двум последним элементам соответственно. При этом аргумент «Подача_воды» будет влиять на свойство линии «Динамический контур». В результате получится графический объект, подобный показанному на рис. 2.9. Сохраним графический объект.

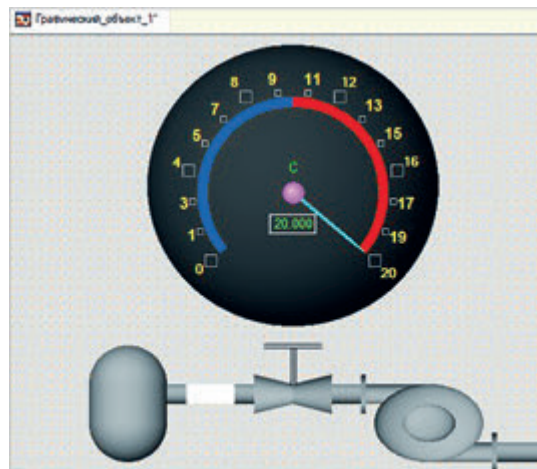


Рис. 2.9. Графический объект с использованием элементов «Трубопровод», «Стрелочный прибор» и «Линия»

Тиражирование графических объектов. Перейдем на Экран и с помощью *drag'n'drop* дважды перетащим туда один и тот же элемент «Графический_объект_1», после чего добавим следующие элементы:

1) на Экран элемент «Емкость» и соединим его с добавленными графическими объектами;

2) в узел RTM_1 два источника сигналов: «Пила» и «Треугольник». Создадим для них каналы и аргументы Экрана, к которым сделаем привязку этих источников. Привяжем созданные аргументы каналов «Пила» и «Треугольник» к аргументам «Температура» графических объектов;

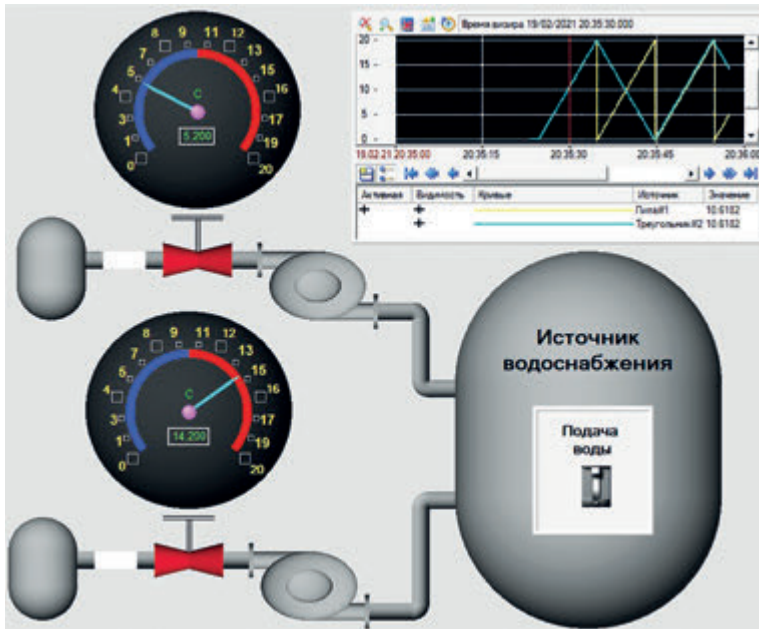


Рис. 2.10. Иллюстрация работы системы водоснабжения без подачи воды, но с контролем температуры

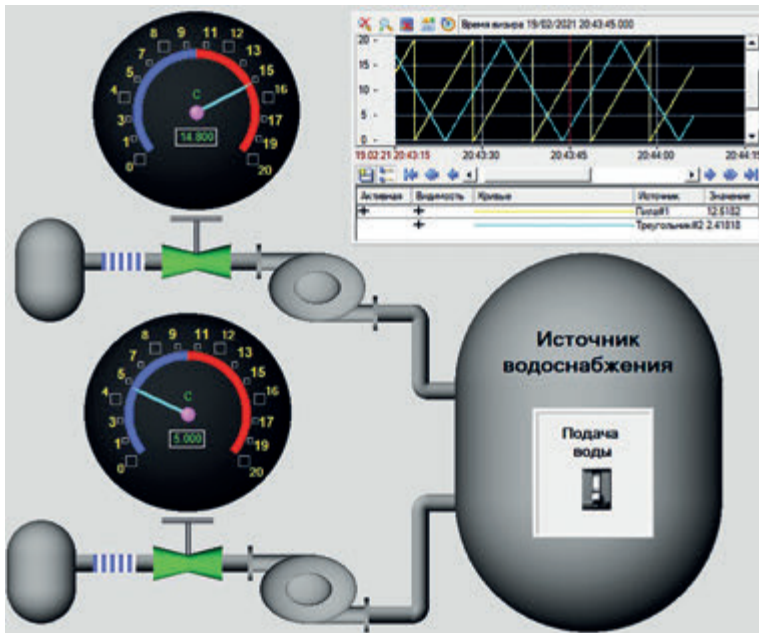


Рис. 2.11. Иллюстрация работы системы водоснабжения с подачей воды и контролем температуры

3) на Экран элемент «Выключатель 2» для включения имитации подачи воды в трубопровод. Создадим на Экране аргумент «Вода» и привяжем его к соответствующему аргументу («Подача_воды») обоих графических объектов;

4) на Экран элемент «Тренд» и перетащим на него аргументы «Температура» обоих графических объектов.

Проверим работоспособность созданных элементов, правильность привязки каналов и заданных свойств. Для этого сохраним и запустим созданный проект на исполнение. Результаты работы показаны на рис. 2.10 и 2.11.

Следует обратить внимание на то, что при верхнем положении «Выключателя 2» клапаны трубопровода окрашены в красный цвет и в трубопроводе отсутствует индикация подачи воды (динамический контур линии окрашен в белый цвет), как показано на рис. 2.10. При нижнем положении «Выключателя 2» клапаны трубопровода окрашены в зеленый цвет и в трубопроводе присутствует индикация подачи воды (динамический контур линии окрашен в синий и белый цвета), как на рис. 2.11. На элементах «Тренд» и «Стрелочный прибор» показаны текущие значения температуры воды в трубопроводе.

В качестве отчета по лабораторной работе преподавателю предоставляют информацию о созданной АИУС с применением графических объектов и о ходе выполнения работы, при этом студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Задание для самоконтроля и подготовки к защите лабораторной работы № 2

Разработайте Экран, содержащий элементы управления и контроля за горячим и холодным водоснабжением, а также за электроснабжением четырехквартирного таунхауса. Экран должен содержать четыре области (по числу квартир). Во всем доме для диспетчера должны быть предусмотрены выключатели воды и света.

Лабораторная работа № 3

СОЗДАНИЕ ПРОГРАММЫ РАБОТЫ ДЕКАДНОГО ЧЕТЫРЕХРАЗЯДНОГО СЧЕТЧИКА НА ЯЗЫКЕ ST

Цели работы:

- 1) ознакомиться с основными принципами создания программ на языке ST (техно ST);
- 2) ознакомиться с основными типами данных, операторами и конструкциями языка ST;
- 3) научиться разрабатывать код программы на языке ST;
- 4) научиться связывать аргументы программы с аргументами Экрана.

План работы

1. Создание программы на языке ST.
2. Разработка кода программы.
3. Привязка аргументов элемента «Программа#1» к аргументам Экрана.
4. Вывод результатов работы элемента «Программа#1» на элемент «Тренд».

Создание программы на языке ST. В соответствии с алгоритмом, приведенным в лабораторной работе № 1, создадим стандартный проект, узел RTM и Экран. Щелкнув ПКМ на созданном узле RTM_1, выберем строку «Создать компонент», далее — «Программа», как показано на рис. 3.1.

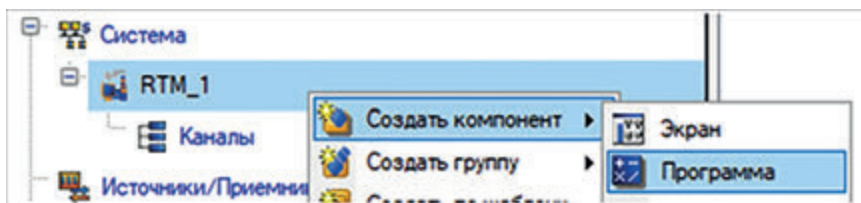



Рис. 3.1. Процесс добавления элемента «Программа» в узел RTM_1

В результате в узле RTM_1 должен появиться соответствующий элемент, как показано на рис. 3.2. Щелкнув два раза ЛКМ на созданном элементе «Программа#1:2», откроем список «Структура программы», как показано на рис. 3.3.



Рис. 3.2. Элемент «Программа#1:2» в узле RTM_1

В списке «Структура программы» содержится перечень аргументов, переменных, функций и т. д.

Выберем строку «Аргументы» и, нажав кнопку , создадим следующие аргументы: «Arg1», «Arg2», «Сум», «Разн» с типами IN, IN, OUT, OUT соответственно, т. е. два аргумента, а также аргументы их суммы и разности. Результат показан на рис. 3.4.

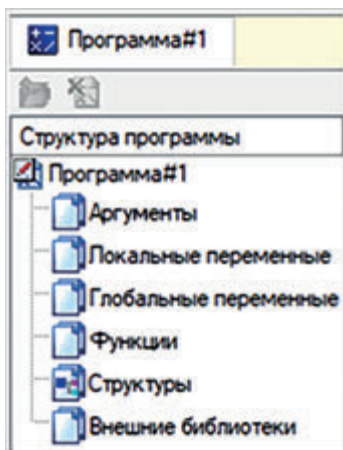


Рис. 3.3. Список «Структура программы»

Имя	Тип	Тип данных
Arg1	IN	REAL
Arg2	IN	REAL
Сум	OUT	REAL
Разн	OUT	REAL

Рис. 3.4. Перечень созданных аргументов с указанием их типов

В списке «Структура программы» выберем строку «Программа», после чего появится окно «Выбор Языка Программирования». Выберем первый пункт «ST программа», как показано на рис. 3.5, и нажмем кнопку «Принять».

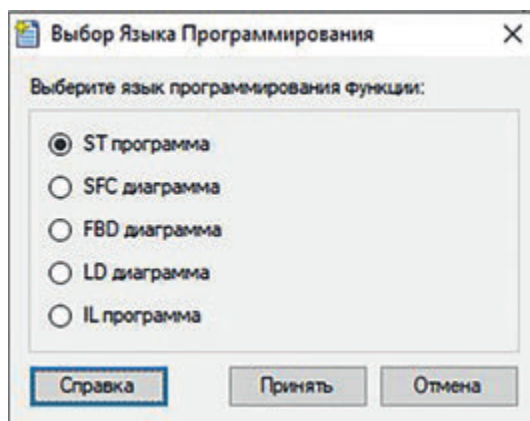


Рис. 3.5. Окно «Выбор Языка Программирования»

Открывшееся окно предназначено для написания кода программы. Обратите внимание на то, что серые области окна защищены от редактирования. Для написания программного кода предназначена белая область.

Разработка кода программы. Напишем код, позволяющий суммировать (выходной аргумент «Сум») и вычитать (выходной аргумент «Разн») поступающие на вход программы аргументы (входные аргументы «Arg1» и «Arg2»). В результате код программы должен выглядеть примерно так, как показано на рис. 3.6.

```

PROGRAM
VAR_INPUT Apr1 : REAL; END_VAR
VAR_INPUT Apr2 : REAL; END_VAR
VAR_OUTPUT Сум : REAL; END_VAR
VAR_OUTPUT Разн : REAL; END_VAR

Сум = Apr1 + Apr2;
Разн = Apr1 - Apr2;

END_PROGRAM

```

Рис. 3.6. Программный код для суммирования и вычитания двух аргументов

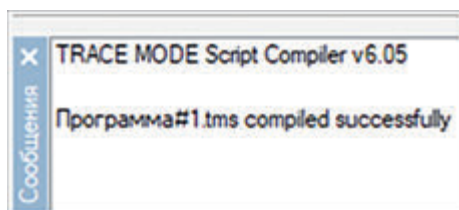


Рис. 3.7. Сообщение об отсутствии ошибок в программе

«Создать компонент», далее — «Пила» и «Треугольник»;

2) привяжем их к каналам узла RTM_1;

3) создадим четыре аргумента Экрana: «Пила», «Треугольник», «Сум», «Разн»;

4) привяжем к аргументам «Пила» и «Треугольник» Экрana каналы «Пила» и «Треугольник» узла RTM_1;

5) привяжем к аргументам «Пила» и «Треуг» элемента «Программа#1» созданные аргументы «Пила» и «Треугольник» Экрana;

6) перейдем в элемент «Программа#1» в строку «Аргументы»;

7) для аргумента «Арг1» установим привязку к аргументу Экрana «Пила», как показано на рис. 3.8;

8) аналогичным образом привяжем второй аргумент «Арг2» к каналу «Треугольник» Экрana, аргумент «Сум» — к «Сум», «Разн» — к «Разн», как показано на рис. 3.9;

9) проверим элемент «Программа#1» на наличие ошибок. Отсутствие ошибок означает, что привязка аргументов элемента «Программа#1» к аргументам Экрana выполнена успешно;

10) добавим на Экрana элемент «Тренд». Выведем на элемент «Тренд» кривые всех четырех аргументов Экрana.

Вывод результатов работы элемента «Программа#1» на элемент «Тренд». Для проверки работоспособности созданного проекта, правильности работы и привязки аргументов программы сохраним и запустим созданный проект на исполнение. Результат работы показан на рис. 3.10.

После написания кода программы его необходимо проверить на наличие ошибок. Это действие выполняется с помощью соответствующей



кнопки на верхней панели. Если код программы не содержит ошибок, в окне «Сообщения» (в нижнем левом углу Экрana) появится соответствующая надпись, как показано на рис. 3.7.

Если код неверный, необходимо найти ошибки. Отладку программы можно выполнить с помощью соответствующих кнопок на верхней панели Экрana.

Привязка аргументов элемента «Программа#1» к аргументам Экрana. Для привязки аргументов элемента «Программа#1» к аргументам Экрana выполним следующие действия:

1) создадим группу «Генераторы_1» (см. рис. 1.15), выберем строку

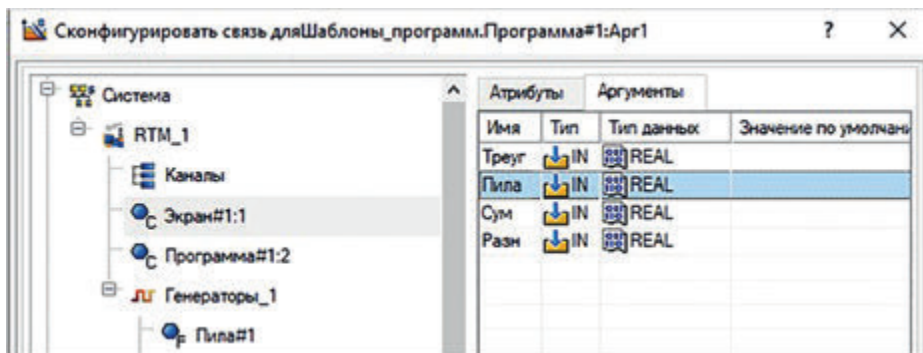


Рис. 3.8. Привязка аргумента «Arg1» к аргументу Экрана

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Arg1	IN	REAL		Экран#1:1:Пила (Система.RTM_1)
Arg2	IN	REAL		Экран#1:1:Треуг (Система.RTM_1)
Сум	OUT	REAL		Экран#1:1:Сум (Система.RTM_1)
Разн	OUT	REAL		Экран#1:1:Разн (Система.RTM_1)

Рис. 3.9. Итоговый результат привязки аргументов элемента «Программа#1»

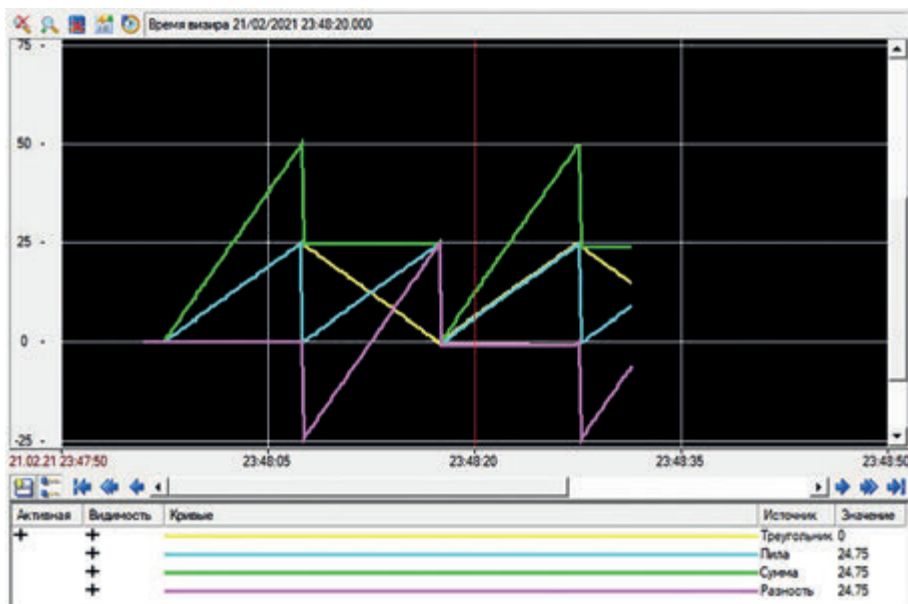


Рис. 3.10. Изображение сигналов: входные аргументы (желтый и голубой цвета), их сумма (зеленый) и разность (розовый)

Для создания декадного четырехразрядного счетчика изменим аргументы и скорректируем программный код. Полученный результат показан на рис. 3.11.

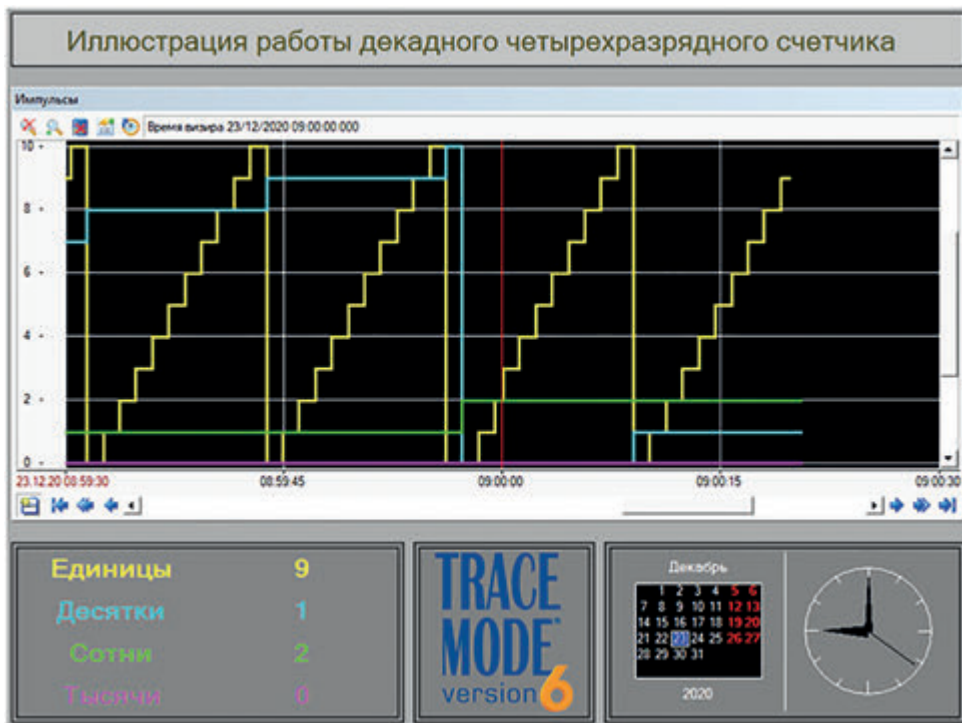


Рис. 3.11. Иллюстрация работы программы декадного четырехразрядного счетчика

В качестве отчета по лабораторной работе преподавателю предоставляются информация о созданной АИУС с применением программы, написанной на языке ST, и о ходе выполнения работы. Студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Задание для самоконтроля и подготовки к защите лабораторной работы № 3

Напишите программу на языке ST, которая отслеживает значение температуры помещения. Исходные данные и пояснения:

- 1) расположите на Экране два элемента «Текст», «Выключатель 4» и «Тренд»;
- 2) изменение температуры в диапазоне от 0 до 40 °С должна имитировать синусоида от источника «Генераторы»;

3) при температуре на входе в диапазоне от 20 до 30 °С в первом элементе «Текст» должна появляться надпись «НОРМА» и «Выключатель 4» должен гореть зеленым светом. Вне указанного диапазона температур в первом элементе «Текст» должна появляться надпись «НЕ НОРМА», а «Выключатель 4» должен гореть красным светом;

4) во втором элементе «Текст» должно записываться текущее значение температуры (в целых числах). Если оно за пределами нормы, надпись должна быть красной, в противном случае — зеленой;

5) выведите на элемент «Тренд» текущие значения температуры и добавьте две кривые, обозначающие заданные диапазоны.

Лабораторная работа № 4

СОЗДАНИЕ ПРОГРАММЫ РАБОТЫ ДЕКАДНОГО ЧЕТЫРЕХРАЗЯДНОГО СЧЕТЧИКА НА ЯЗЫКЕ FBD

Цели работы:

- 1) научиться работать с FBD-блоками «Счетчик», «Статистика» и «Фильтрация пиков»;
- 2) научиться разрабатывать программы на языке FBD;
- 3) научиться связывать аргументы программы языка FBD с аргументами Экрана.

План работы

1. Создание программы на языке FBD.
2. Привязка аргументов «Программы#1» к аргументам Экрана.
3. Вывод результатов работы «Программы#1» на элемент «Тренд».

Создание программы на языке FBD. В соответствии с алгоритмом, приведенным в лабораторной работе № 1, создадим стандартный проект, узел RTM и Экран. Щелкнув ПКМ на созданном узле RTM_1, выберем строку «Создать компонент», далее — «Программа», как показано на рис. 4.1.

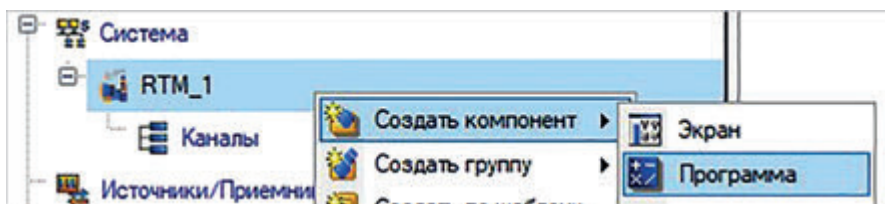


Рис. 4.1. Процесс добавления элемента «Программа» в узел RTM_1



Рис. 4.2. Элемент «Программа#1:2» в узле RTM_1

В результате в узле RTM_1 должен появиться соответствующий элемент, как показано на рис. 4.2.

Щелкнув два раза ЛКМ на созданном элементе «Программа#1:2», откроем окно «Программа#1», как показано на рис. 4.3.

Привязка аргументов «Программы#1» к аргументам Экрана. Список «Структура программы» представляет собой перечень аргументов, переменных, функций и т. д. Выберем строку «Аргументы» и создадим следующие аргументы: «Arg1», «Arg2», «Сум», «Разн» с типами IN, IN, OUT, OUT соответственно, т. е. первый и второй аргументы, а также аргументы их суммы и разности. Результат показан на рис. 4.4.

В списке «Структура программы» (рис. 4.3) выберем первую строку «Программа#1», после чего появится окно «Выбор Языка Программирования». Выберем пункт «FBD диаграмма» и нажмем кнопку «Принять».

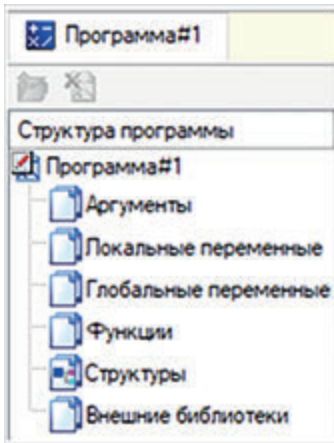


Рис. 4.3. Окно «Программа#1»

Имя	Тип	Тип данных
Arg1	IN	REAL
Arg2	IN	REAL
Сум	OUT	REAL
Разн	OUT	REAL

Рис. 4.4. Перечень созданных аргументов с их типами

Откроется окно для создания программы на языке FBD, как показано на рис. 4.5.

В области выбора группы блоков «Логические» выберем четыре блока счетчика STU и перетащим их в область создания программы, после чего

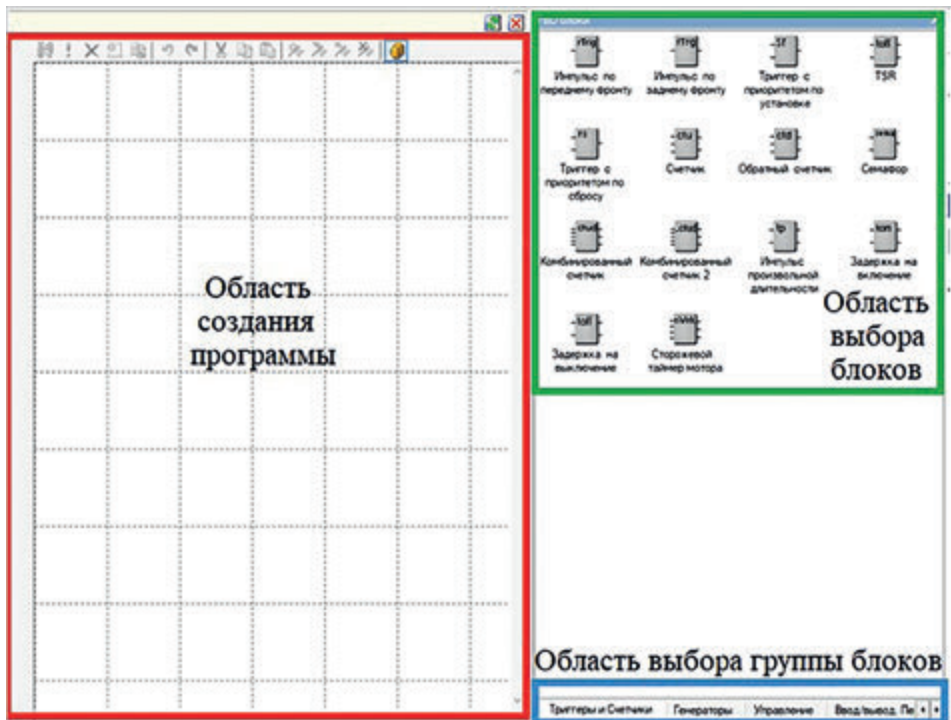


Рис. 4.5. Окно создания программы на языке FBD

добавим на Экран входы и выходы аргументов «Программа#1», представленные на рис. 4.6. Созданные аргументы показаны на рис. 4.7.

Добавим аргументы Экрана и привяжем их к аргументам созданной «Программы#1», как показано на рис. 4.8.

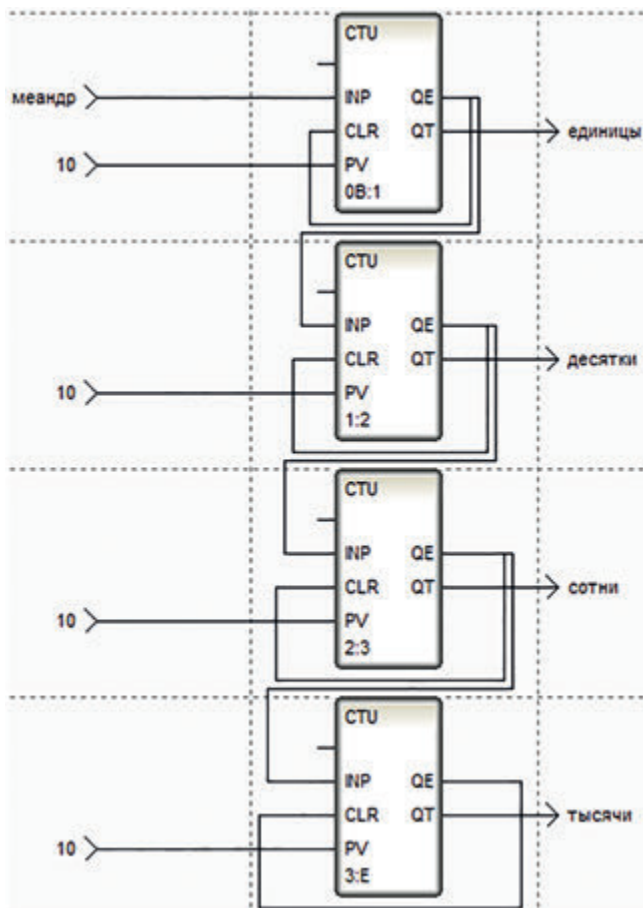


Рис. 4.6. Созданная «Программа#1» декадного четырех-разрядного счетчика

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
меандр	IN	UINT		Экран#1:2:меандр (Система.RTM_1)
установка	IN	REAL	3	
единицы	OUT	REAL		
десятки	OUT	REAL		
сотни	OUT	REAL		
тысячи	OUT	REAL		

Рис. 4.7. Созданные аргументы «Программы#1»

Имя	Тип	Тип данных	Значение по умолчанию	Привязка
Уставка	OUT	INT	10	
меландр	IN	UINT		Битовый_меландр#1:Реальное значение (Система.RTM_1.Генераторы_1)
выход	IN	INT		Программа#1:1:единицы (Система.RTM_1)
десятки	IN	REAL		Программа#1:1:десятки (Система.RTM_1)
сотни	IN	REAL		Программа#1:1:сотни (Система.RTM_1)
тысячи	IN	REAL		Программа#1:1:тысячи (Система.RTM_1)

Рис. 4.8. Созданные аргументы Экрана и их привязки

После создания «Программы#1» и привязки аргументов необходимо ее проверить на наличие ошибок с помощью соответствующей кнопки проверки



. Если программа без ошибок, в окне «Сообщения» (нижний левый угол Экрана) появится соответствующая надпись, как показано на рис. 4.9. Если есть ошибки, необходимо доработать программу.

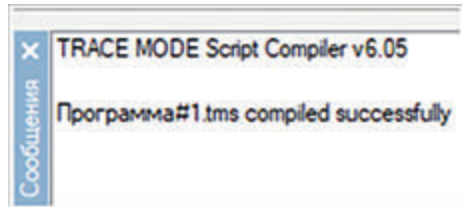


Рис. 4.9. Сообщение об успешной проверке «Программы#1»

Вывод результатов работы «Программы#1» на элемент «Тренд». Для проверки работоспособности созданного проекта, правильности работы и привязки аргументов программы сохраним и запустим созданный проект на исполнение.

В качестве отчета по лабораторной работе преподавателю предоставляют информацию о созданной АИУС с применением программы, написанной на языке FBD, и о ходе выполнения работы. Студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Задания для самоконтроля и подготовки к защите лабораторной работы № 4

1. Создайте программу на языке FBD с алгебраическим блоком «Статистика». Ознакомьтесь с его описанием в справке. Создайте новый имитатор сигнала «Случайное число». С помощью блока «Статистика» рассчитайте среднее, максимальное и минимальное значения для созданного имитатора сигнала «Случайное число». Выведите результат на созданный элемент «Тренд».

2. Спроектируйте информационно-измерительную систему, не пропускающую помехи, уровень которых вы будете самостоятельно задавать с помощью элемента «Ползунок». Результаты выведите на элемент «Тренд». Необходимый алгебраический блок языка FBD найдите в области выбора групп блоков, на вкладке «Управление» окна создания программы.

Лабораторная работа № 5

РАБОТА С БАЗОЙ ДАННЫХ

Цели работы:

1) научиться организовывать связь базы данных (БД) со SCADA-системой;

2) освоить работу с мастером *отправки SQL-запросов* в БД.

План работы

1. Создание связи SCADA-системы с БД.

2. Создание запросов считывания информации из БД.

3. Создание запроса отправки информации в БД.

Создание связи SCADA-системы с БД. В соответствии с алгоритмом, приведенным в лабораторной работе № 1, создадим стандартный проект, узел RTM и Экран. Щелкнем ПКМ на строке «Шаблоны_связей_с_СУБД» дерева проекта, выберем строку «Создать компонент», а затем — «Связь с СУБД», как показано на рис. 5.1.

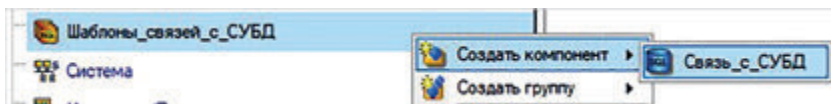


Рис. 5.1. Создание связи SCADA-системы с БД

Создадим БД: внесем в БД три текстовых поля («Фамилия», «Имя» и «Отчество») и одно числовое («Возраст»). Сделаем несколько записей. Сохраним и закроем БД.

После создания БД необходимо наладить с ней связь в SCADA-системе. Для этого зайдём в созданный в дереве проекта элемент «Связь_с_СУБД», дважды щелкнув на нём ЛКМ. В открывшемся окне «База_данных#1» нажмём кнопку «Администратор ODBC», затем кнопку «Добавить» для добавления необходимого драйвера. Выберем «Microsoft Access Driver (*.mdb, *.accdb)», как показано на рис. 5.2, и нажмём кнопку «Готово».

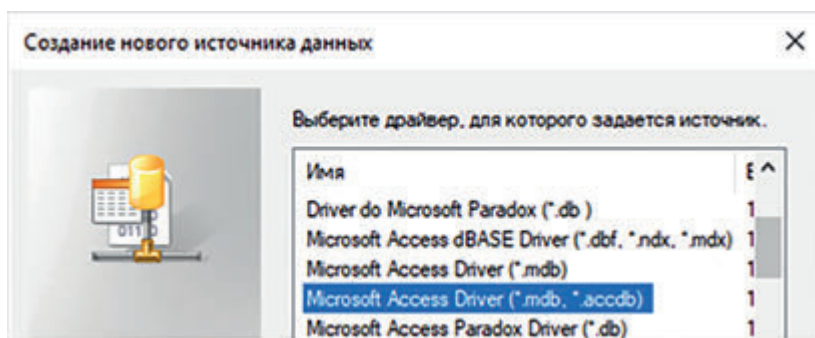


Рис. 5.2. Окно выбора драйвера

Заполним поле «Имя источника данных» в окне установки драйвера, как показано на рис. 5.3, затем с помощью кнопки «Выбрать...» выберем созданную ранее БД, как показано на рис. 5.4, и нажмем кнопку «ОК».

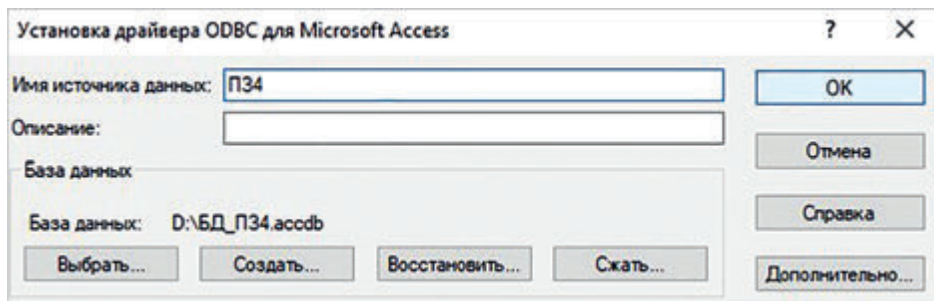


Рис. 5.3. Заполнение поля «Имя источника данных»

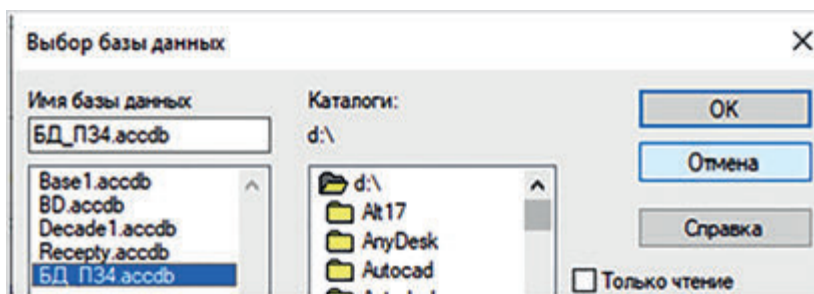


Рис. 5.4. Выбор созданной БД

Далее в окне «База_данных#1» (рис. 5.5) выберем вкладку «Подключение», а затем нажмем кнопку «Проверка», после чего в поле «Отчет» увидим появившуюся надпись об успешном подключении к БД.

В окне «База_данных#1» выберем вкладку «Аргументы» и создадим:

1) четыре аргумента БД типа OUT для считывания информации из БД в SCADA-систему. В колонке «Тип данных» присвоим аргументам соответствующие типы: трем — текстовые STRING и одному — числовое UNIT;

2) четыре аргумента БД типа IN для записи информации в БД из SCADA-системы. В колонке «Тип данных» присвоим аргументам соответствующие типы: трем — текстовые STRING и одному — числовое UNIT, как показано на рис. 5.6.

Создание запросов считывания информации из БД. Создадим первый SQL-запрос: выбор (SELECT) элементов из БД. Для этого нажмем кнопку «Мастер» в окне «База_данных#1», выберем тип запроса «SELECT» и нажмем кнопку «Next». В появившемся окне выберем «Таблица1», как показано на рис. 5.7.

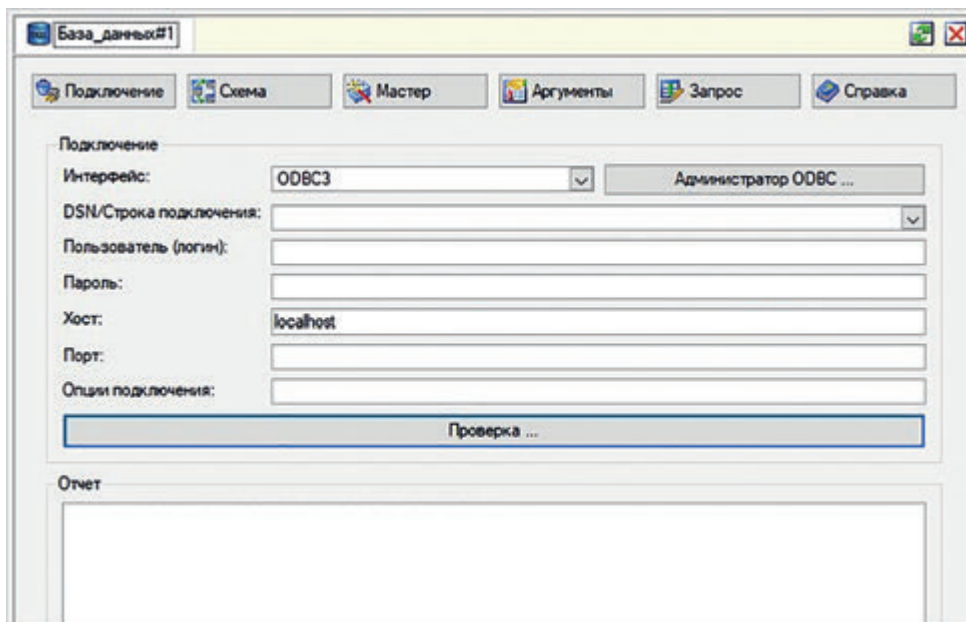


Рис. 5.5. Окно «База_данных#1»

Имя	Тип	Тип данных	Знач	Привл	флаг	Гру	Ед.и.	Комментарий
фамилия	OUT	ab	STRING				...	из БД в ТМ
Имя	OUT	ab	STRING				...	из БД в ТМ
Отчество	OUT	ab	STRING				...	из БД в ТМ
Возраст	OUT	UINT	UINT				...	из БД в ТМ
фамилия_в	IN	ab	STRING				...	в БД из ТМ
Имя_в	IN	ab	STRING				...	в БД из ТМ
Отчество_в	IN	ab	STRING				...	в БД из ТМ
Возраст_в	IN	UINT	UINT				...	в БД из ТМ

Рис. 5.6. Созданные аргументы для работы с БД

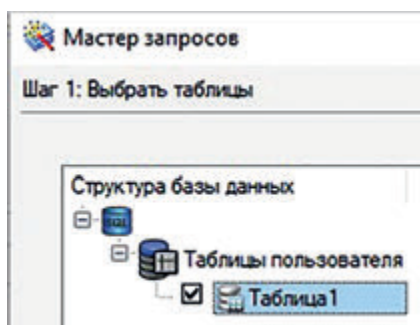



Рис. 5.7. Выбор таблицы при создании запроса SELECT

Нажмем кнопку «Next», выберем интересующие нас поля и нажмем кнопку «Next». Далее привяжем столбец таблицы к одноименному аргументу БД типа IN, нажав кнопку «Привязать», после чего нажмем кнопки «Next» и «Finish». В результате в окне запросов сформируется запрос. Нажав кнопку «Выполнить запрос»  в поле «Отчет», увидим результат работы созданного запроса.

Создадим второй SQL-запрос: вставка (INSERT) элементов из БД. Для этого нажмем кнопку «Мастер» в окне «База_данных#1», выберем тип запроса «INSERT» и нажмем кнопку «Next». В появившемся окне выберем поле «Таблица1» и нажмем кнопку «Next». Далее выберем поле «Возраст», сделаем к нему привязку численного значения во вкладке «Константа», как показано на рис. 5.8, и нажмем кнопку «Next».

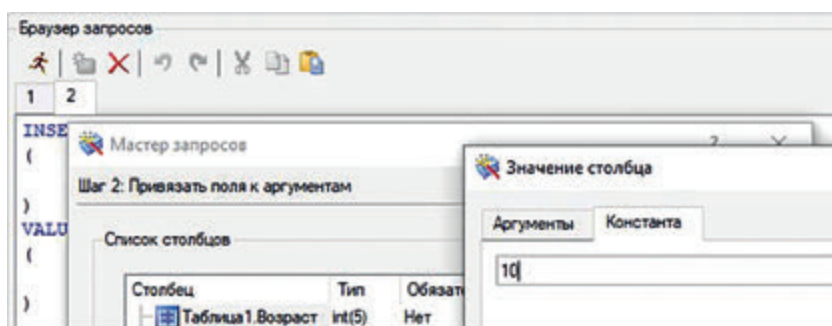



Рис. 5.8. Заполнение поля «Константа» аргумента «Возраст»

Нажав кнопку «Выполнить запрос», увидим результат работы запроса в созданной БД и убедимся в том, что новая строка с введенной цифрой появилась в поле «Возраст».

Создадим третий SQL-запрос: выбор (SELECT) элементов из БД по нажатию элемента «Выключатель 0». Для этого создадим на Экране аргумент «БД_1» типа OUT и UINT, затем привяжем аргумент «БД_1» к реальному значению канала БД. Далее нажмем кнопку «Мастер» в окне «База_данных#1» и создадим один запрос с номером 1 на выбор (SELECT) элементов. Выберем поле «Таблица1», затем поле «Имя» и привяжем к нему аргумент «Имя» типа OUT. Добавим условие: выберем поле «Возраст», нажмем кнопку «Установить левый операнд», выберем вкладку «Константа», введем определенное значение возраста и нажмем кнопку «Установить правый операнд». После этого зададим условие («=», «<>», «<», «>» и т. д.) сравнения левого и правого операндов. Результат показан на рис. 5.9.

Проверим работоспособность созданного запроса в ручном режиме, нажав кнопку «Выполнить запрос» . Результат работы запроса получим в поле «Отчет», как показано на рис. 5.10.

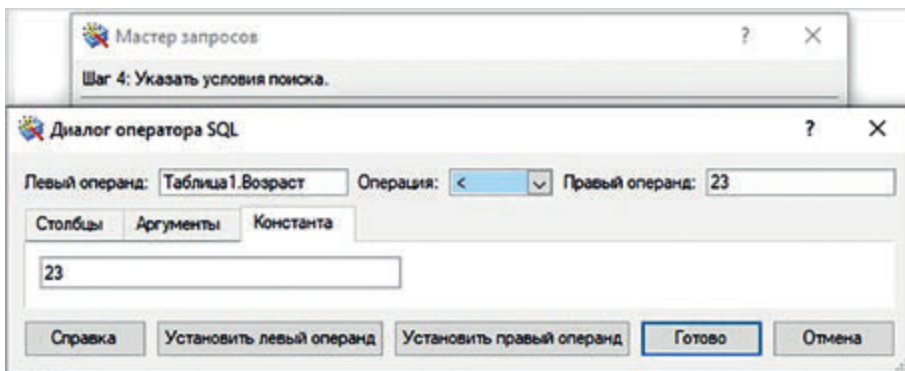


Рис. 5.9. Заданные условия SQL-запроса

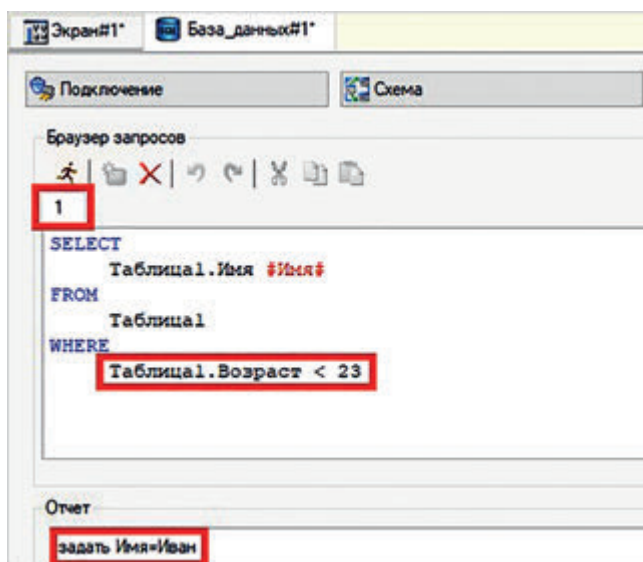


Рис. 5.10. Результат работы запроса SELECT с добавлением условий выбора

Проанализируем данные, представленные на рис. 5.10:

- 1) номер запроса — «1»;
- 2) в SQL-запросе установлено условие «Таблица1.Возраст < 23» под служебным словом WHERE;
- 3) в поле «Отчет» выведено имя «Иван».

Далее создадим на Экране «Выключатель 0» и привяжем к нему созданный ранее аргументу «БД_1». Затем создадим на Экране два элемента «Текст» и привяжем к одному из них аргумент «БД_1», а ко второму — аргумент «Имя» (тип OUT, STRING) БД.

Для проверки работоспособности созданного проекта, правильности работы и привязки аргументов программы сохраним и запустим созданный проект на исполнение.

Создание запроса отправки информации в БД. Создадим четвертый SQL-запрос: добавление (INSERT) элементов в БД по нажатию элемента «Выключатель 0». Для этого создадим в БД еще одно числовое поле и будем добавлять туда из SCADA-системы числа по нажатию выключателя. Источником чисел будет являться генератор случайных чисел.

Нажмем кнопку «Мастер» в окне «База_данных#1» и создадим новый запрос со вторым номером. Добавим на Экран еще один элемент «Выключатель 0» и выполним настройки, как показано на рис. 5.11.

Создадим два новых аргумента Экрана: «В_БД_цифру» (тип OUT, UINT) и «Цифра» (тип IN, UINT). К аргументу «В_БД_цифру» привяжем реальное значение БД, а к аргументу «Цифра» — канал случайного числа, созданного с помощью генератора.

Добавим на Экран два элемента «Текст» и привяжем созданные аргументы для диагностики записей в БД и отслеживания значения показаний канала случайного числа.

Для проверки работоспособности созданного проекта, правильности работы и привязки аргументов программы сохраним и запустим созданный проект на исполнение.

В качестве отчета по лабораторной работе преподавателю предоставляют информацию о созданной АУИС с применением БД, реализованной с помощью СУБД, и о ходе выполнения работы. Студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Свойство	Значение
Привязка	<2> В_БД_цифру
Вид индикации	Arg & Констант
Инверсия	False
Константа	0x2
Код доступа	0
Значение (XOR)	0x2

Рис. 5.11. Настройки элемента «Выключатель 0»

Задание для самоконтроля и подготовки к защите лабораторной работы № 5

Разработайте Экран диспетчера, содержащий элементы контроля давления в трубопроводе. Разместите на Экране элемент «Тренд», на котором должны отображаться максимальное, минимальное и текущее значения давления. Должно быть предусмотрено выполнение следующих условий:

- если значение давления больше максимально допустимого, оно автоматически должно быть добавлено в БД с указанием времени;
- если значение давления меньше минимально допустимого, оно вручную (например, по нажатию элемента «Выключатель») добавляется в БД с указанием времени.

Для привлечения внимания диспетчера к происшествию на Экране должны быть установлены соответствующие индикаторы.

Лабораторная работа № 6

ОТЧЕТ ТРЕВОГ И ЗАПУСК ПРИЛОЖЕНИЙ WINDOWS

Цели работы:

- 1) научиться генерировать отчеты тревог (ОТ);
- 2) научиться документировать ОТ в текстовый файл;
- 3) научиться запускать приложения Windows.

План работы

1. Генерирование ОТ.
2. Документирование ОТ в файл.
3. Запуск приложения «Блокнот» в Windows.

Генерирование отчета тревог. В соответствии с алгоритмом, приведенным в лабораторной работе № 1, создадим стандартный проект, узел RTM и Экран.

В начале работы создадим генератор синусоиды, для чего воспользуемся группой «Источники» и компонентом «Синусоида». Перенесем его в каналы RTM_1. Щелкнем ПКМ на RTM_1 и выберем строку «Редактировать», как показано на рис. 6.1.

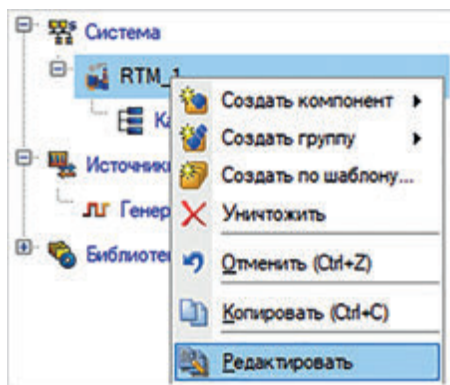


Рис. 6.1. Выбор строки «Редактировать»

Перейдем во вкладку «Отчет тревог / Дамп / Параметры» и зададим следующие данные (рис. 6.2):

1) в поле «Имя файла» введем название будущего ОТ (имя файла должно иметь в длину более четырех символов и желательно расширение *.txt);

2) в поле «Состояние» установим TRUE;

3) в поле «Максимум записей» зададим 50;

4) поле «Формат даты» оставим по умолчанию.

Перейдем в окно свойств созданного канала «Синусоида#1» и в группе

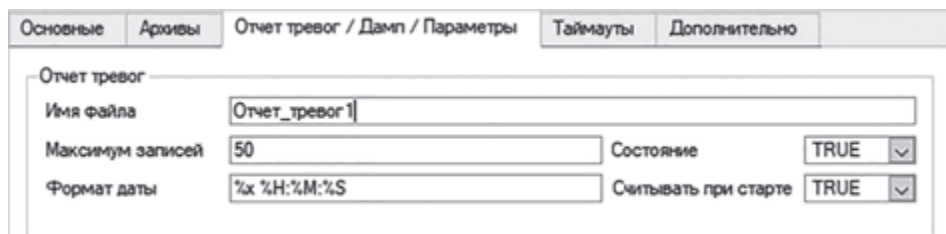


Рис. 6.2. Вкладка «Отчет тревог / Дамп / Параметры»

«Системные» выберем «Архивация». Там установим галочку в поле «Отчет тревог».

Щелкнув ПК на узле RTM_1, создадим группу «Словари сообщений», как показано на рис. 6.3.

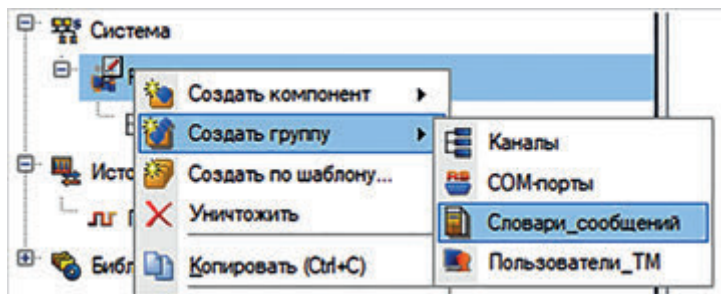


Рис. 6.3. Процесс создания словаря сообщений

Создадим словарь для каналов типа Float, как показано на рис. 6.4.

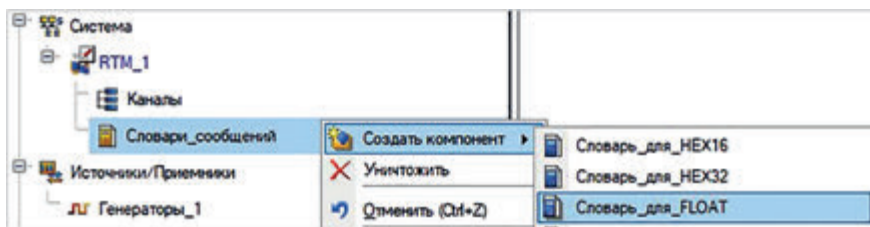


Рис. 6.4. Процесс создания словаря для каналов типа Float

Подобный словарь нужен в связи с тем, что мы будем работать с синусоидой, являющейся как раз каналом типа Float. Щелкнув два раза ЛКМ на созданном словаре, зайдем в раздел его свойств. В поле «Направление» можно выбирать для словаря разные установки и их варианты, например:

- AR — Генерация отчета тревог;
 - G — Графический экран;
 - Prn — Вывод на принтер;
 - GSM — Отправка СМС;
 - Net — Отправка в сеть;
 - Play — Проигрывание звукового файла с расширением .wav
- и т. д.

Само сообщение может быть отнесено к одной из следующих категорий (см. *category.tmc*), задающих степень его ответственности (список «Категория»):

- <W> — Предупреждение;
- <E> — Ошибка;

- <A> — Тревога;
 - <M> — Сообщение;
 - < > — Без категории
- и т. д.

Создадим сообщение типа «AR+G» категорий <W> Предупреждение, <E> Ошибка и <A> Тревога. Напишем текстовые сообщения (например, NORMA) в соответствующих полях, как показано на рис. 6.5.

Направление	Категория	Текст
AR+G	<> Без категории	NORMA
AR+G	<W> Предупреждение	>
AR+G	<W> Предупреждение	<
AR+G	<A> Тревога	>>
AR+G	<A> Тревога	<<
AR+G	<E> Ошибка	>>>
AR+G	<E> Ошибка	<<<

Рис. 6.5. Запись сообщений

Зайдем в свойства канала «Синусоида#1», дважды щелкнув ЛКМ на канале «Синусоида#1». Поставим галочку в поле «Использовать». Указанные границы ВП, ВА, ВГ и др. соответствуют категориям тревог, приведенным на рис. 6.5.


Установим значения границ:

- ВП (верхний предел) = 90;
- ВА (верхняя аварийная граница) = 80;
- ВГ (верхняя предупредительная граница) = 70;
- НГ (нижняя предупредительная граница) = 30;
- НА (нижняя аварийная граница) = 20;
- НП (нижний предел) = 10.

Результат настройки канала «Синусоида#1» показан на рис. 6.6.

В окне свойств канала «Синусоида#1» выберем созданный словарь «Для_FLOAT#1» в поле «Индекс аварийного словаря».

Перейдем на Экран. Создадим на Экране элементы «Строка ОТ» и

«ОТ узла» , аргумент Экрана «Синусоида» и привяжем к нему канал «Синусоида#1». В элементы «Строка ОТ» и «ОТ узла» добавим аргумент Экрана «Синусоида», затем создадим элемент «Тренд» и выведем на него аргумент Экрана «Синусоида».

Для проверки работоспособности созданного проекта, правильности работы и привязки аргументов сохраним и запустим созданный проект на исполнение. Результат работы созданного проекта показан на рис. 6.7.

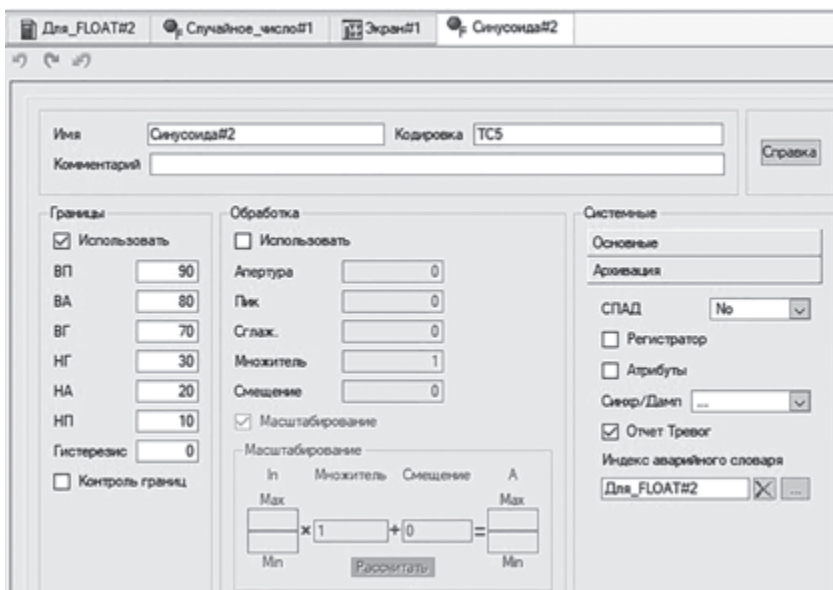


Рис. 6.6. Настройка канала «Синусоида#1»

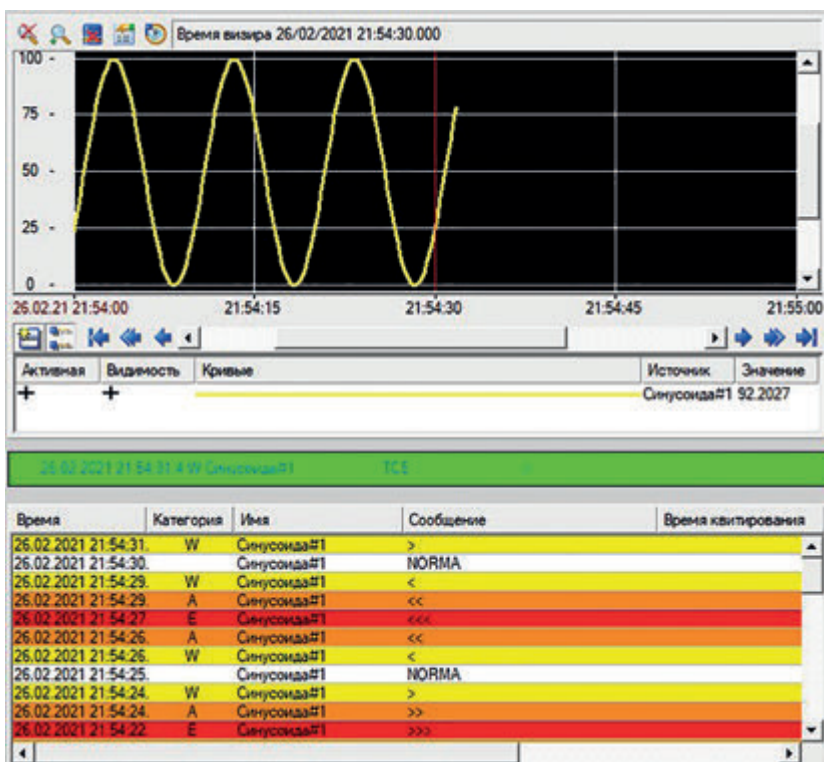


Рис. 6.7. Результат работы созданного проекта

Верхняя часть рисунка — изображение аргумента Экрана «Синусоида», средняя часть — элемент «строка ОТ», показывающая текущее сообщение ОТ; нижняя часть — элемент «ОТ узла» с выводимыми туда сообщениями ОТ, окрашенными в заданный во вкладке «Цвета» цвет.

Документирование ОТ в файл. Зайдем в файл «Отчет_тревог.txt», имя которого было задано в начале работы. Данный файл находится в папке с проектом и содержит последние 50 (как было задано) записей ОТ, часть которых показана на рис. 6.8.

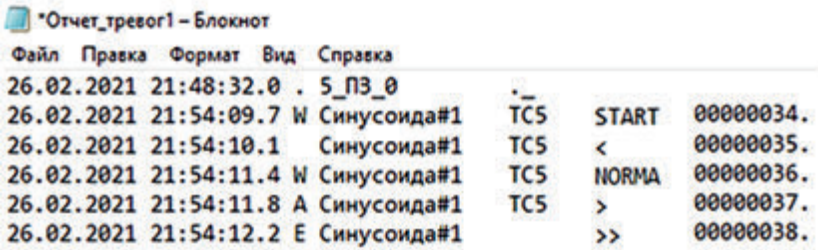


Рис. 6.8. Содержание текстового файла ОТ

Строка сообщения в текстовом файле ОТ содержит поля, разделенные пробелами:

Date Time Category Name Coding Text UserID T_ack N

Здесь **Date Time** — поле вывода даты и времени; **Category** — категория сообщения, 1 знакоместо; **Name** — для сообщения по каналу — имя канала; **Coding** — кодировка и время изменения канала (31 знакоместо); **Text** — текст сообщения или пользовательский комментарий (48 знакомест); **UserID** — идентификатор пользователя, квитирававшего сообщение (5 знакомест). Сообщения могут квитироваться с помощью графических элементов; **T_ack** — время квитиования сообщения в формате **DD_НН:ММ:SS**; **N** — порядковый номер строки в ОТ в формате HEX (8 символов).

Поясним, в чем заключается квитиование. Это специализированная функция — подтверждение оператором того, что он видел появившееся сообщение о тревоге. Квитиовать можно посредством:

- нажатия комбинаций «горячих» клавиш ЛКМ + *Ctrl*;
- отправки цифры 4 в 46-й атрибут канала.

Создадим соответствующую кнопку для активации функции квитиования, для чего создадим новый аргумент Экрана «Квитиование». Привяжем к нему 46-й атрибут канала «Синусоида». По нажатию кнопки (рис. 6.9) будем передавать значение 4, как показано на рис. 6.10.

Для проверки работоспособности созданного проекта, правильности работы и привязки аргументов сохраним и запустим созданный проект на исполнение.

Стоит отметить, что при нажатии кнопки «Квитиование» в элементе «ОТ узла» в соответствующем столбце появится время квитиования, как

Рис. 6.9. Передача значения по нажатию кнопки

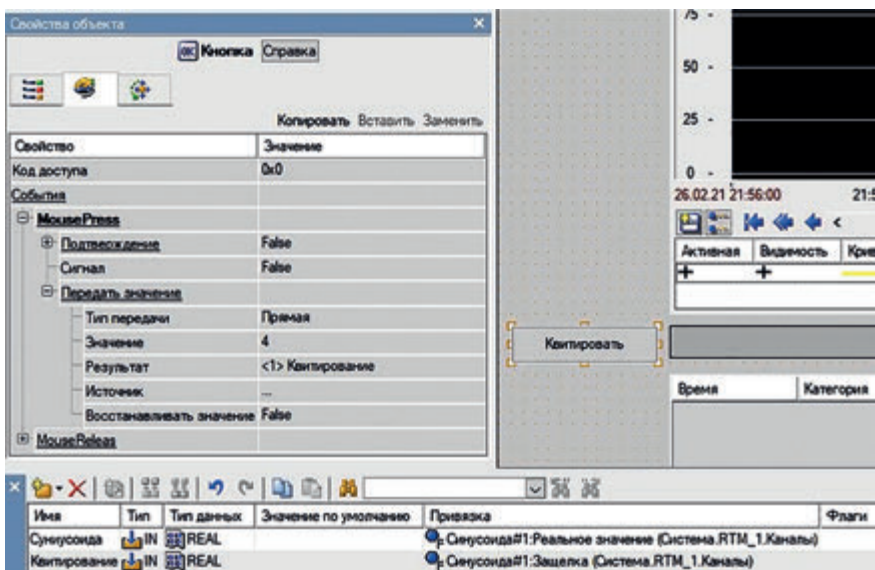
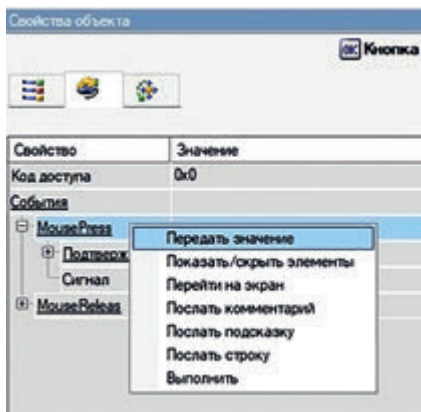


Рис. 6.10. Установка параметров кнопки «Квितिование»

показано на рис. 6.11. В файле «Отчет_тревог1.txt» также появится запись о квितिовании.

Запуск приложения Блокнот Windows. Создадим в узле RTM_1 новый канал типа CALL, щелкнем на нем ПКМ и зайдем в его свойства во вкладке

«Аргументы». С помощью кнопки создания аргумента создадим аргумент ARG_000 типа IN, REAL, как показано на рис. 6.12.

Перейдем во вкладку «Атрибуты канала» CALL#2, нажав кнопку. В поле «Тип вызова» выберем строку «Ехес», как показано на рис. 6.13.

Время	Категория	Имя	Сообщение	Время квитирования
26.02.2021 22:16:55		Сенусонда#1	NORMA	22:16:55
26.02.2021 22:16:55	W	Сенусонда#1	<	
26.02.2021 22:16:54	A	Сенусонда#1	<<	
26.02.2021 22:16:52	E	Сенусонда#1	<<<	
26.02.2021 22:16:52	A	Сенусонда#1	<<	
26.02.2021 22:16:52	W	Сенусонда#1	<	
26.02.2021 22:16:50		Сенусонда#1	NORMA	22:16:51

Рис. 6.11. Отображение времени квитирования в элементе «ОТ узла» при нажатии кнопки «Квитирование»

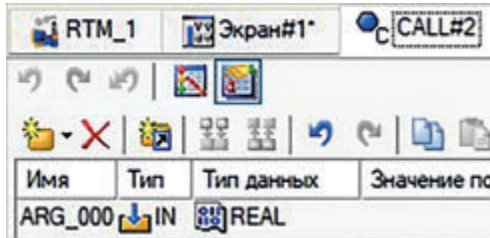


Рис. 6.12. Создание аргумента канала CALL#2

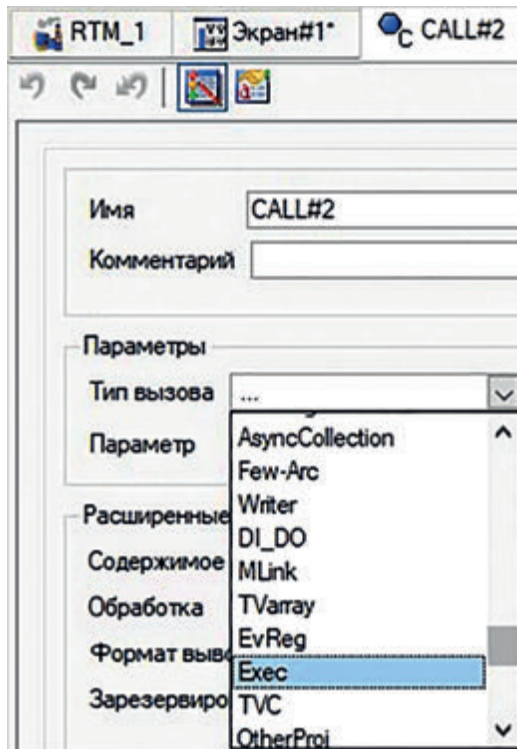


Рис. 6.13. Создание аргумента канала CALL#2

Укажем имя «notepad» и адрес «C:\Windows» открываемого приложения «Блокнот» в поле «Комментарий», как показано на рис. 6.14.

Запустим профайлер, откроем вкладку «Вид», выберем строку «Панель МРВ», после чего откроется окно «Службная панель МРВ». Найдем в нем строку с названием нашего приложения «notepad» и отправим на его вход «In» единицу, как показано на рис. 6.15. В итоге откроется приложение «Блокнот».

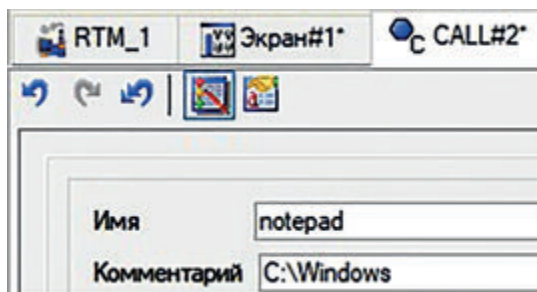


Рис. 6.14. Указание имени и адреса приложения «Блокнот»

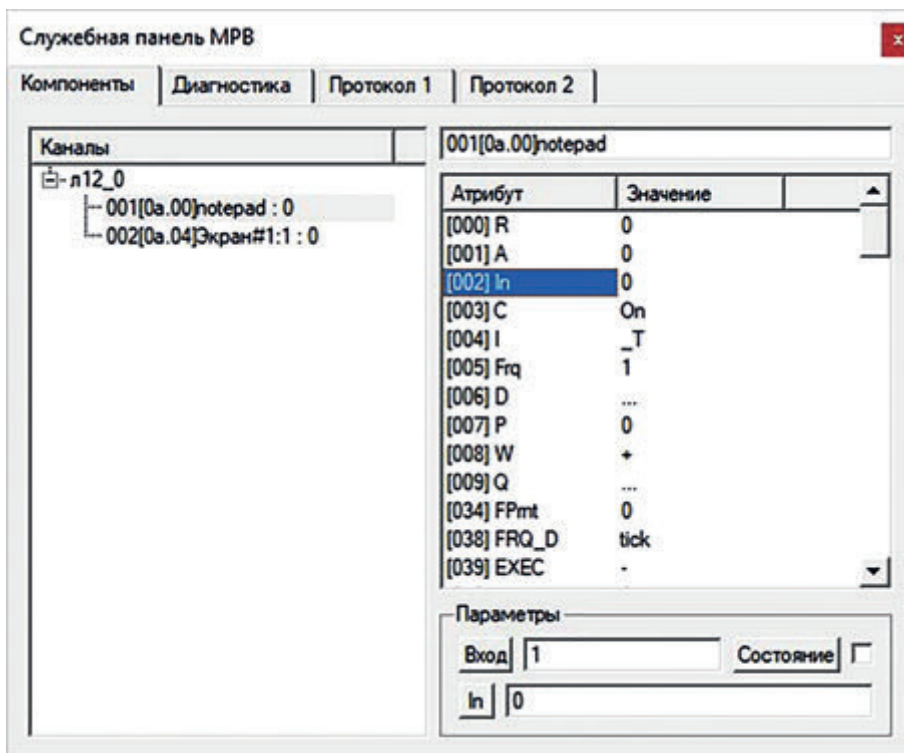


Рис. 6.15. Указание имени и адреса открываемого приложения «Блокнот»

В качестве отчета по лабораторной работе преподавателю предоставляют информацию о созданной АИУС с применением ОТ, запуском приложения *Windows* и о ходе выполнения работы. Студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Задание для самоконтроля и подготовки к защите лабораторной работы № 6

Создайте Экран для контроля за входным сигналом. Для этого выполните следующие действия:

- добавьте на Экран каналы, содержащие источники сигналов «Синусоида» и «Случайное число»;
- выведите созданные каналы на элемент «Тренд»;
- выведите на Экран элементы «строка ОТ» и «ОТ узла», поместив в них созданные источники сигналов;
- во время работы RTM сохраняйте ОТ в файл. Количество строк в файле будет равно вашему варианту в Списке студентов + 30;
- добавьте на Экран кнопки вызова блокнота, браузера, калькулятора, квитиования;
- проверьте работоспособность созданного узла RTM.

Лабораторная работа № 7

ОРГАНИЗАЦИЯ СВЯЗИ ПЛК С АРМ ДИСПЕТЧЕРА ПО ПРОТОКОЛУ MODBUS RTU

Цели работы:

1) научиться конфигурировать систему в соответствии с выбранными параметрами протокола;

2) научиться работать с регистрами чтения и записи протокола Modbus RTU;

3) научиться налаживать связь программируемого логического контроллера (ПЛК) с персональным компьютером (ПК) по протоколу Modbus RTU и интерфейсу RS-485.

План работы

1. Сборка схемы установки для демонстрации взаимодействия ПК с ПЛК.

2. Конфигурирование ПЛК.

3. Настройка узла RTM_1, проверка работоспособности собранной установки.

Программируемый логический контроллер — это промышленный компьютер, адаптированный под управление производственными процессами, которые осуществляются с помощью сборочных линий, роботизированных устройств.

Сборка схемы установки для демонстрации взаимодействия ПК с ПЛК. В соответствии с алгоритмом, приведенным в лабораторной работе № 1, создадим стандартный проект, узел RTM и Экран.

Соберем схему установки, как показано на рис. 7.1.

Конфигурирование ПЛК. Сконфигурируем входы и выходы ПЛК, как показано на рис. 7.2. Обратим внимание на области, выделенные прямоугольными рамками — именно их необходимо настроить: два входа (*fdi0* и *fdi1*) и четыре выхода (*do0* — *do3*).

Настроим протокол Modbus RTU в соответствии с рис. 7.3–7.5.

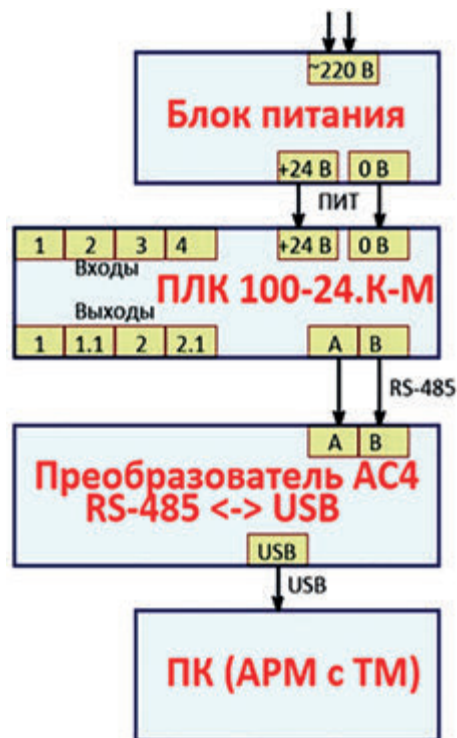


Рис. 7.1. Схема установки

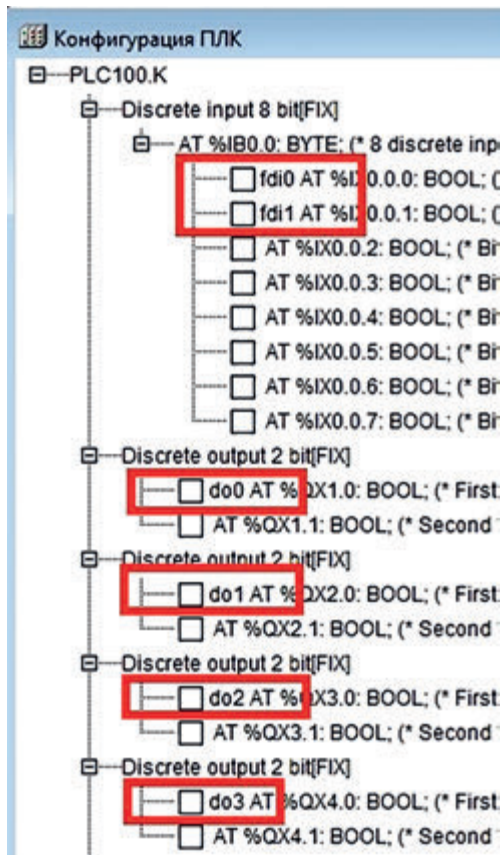


Рис. 7.2. Конфигурирование входов и выходов ПЛК

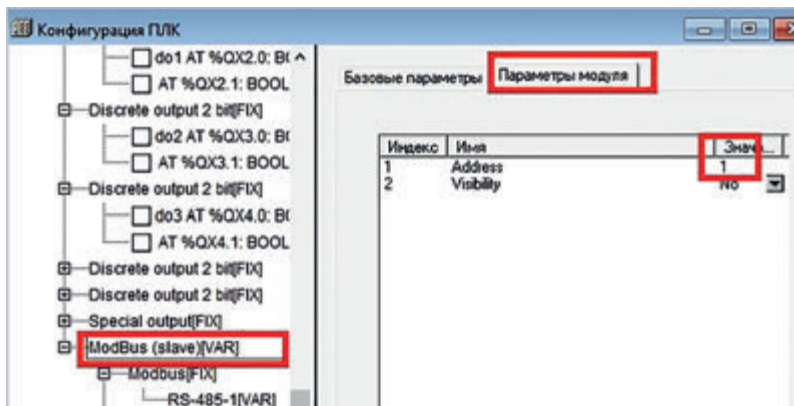


Рис. 7.3. Окно «Конфигурация ПЛК» для настройки Modbus RTU

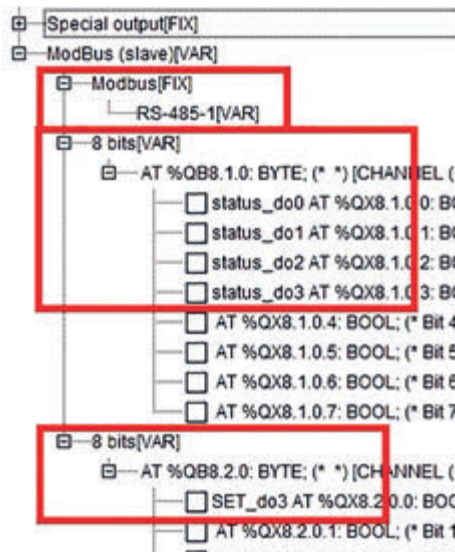


Рис. 7.4. Настройка регистров Modbus RTU

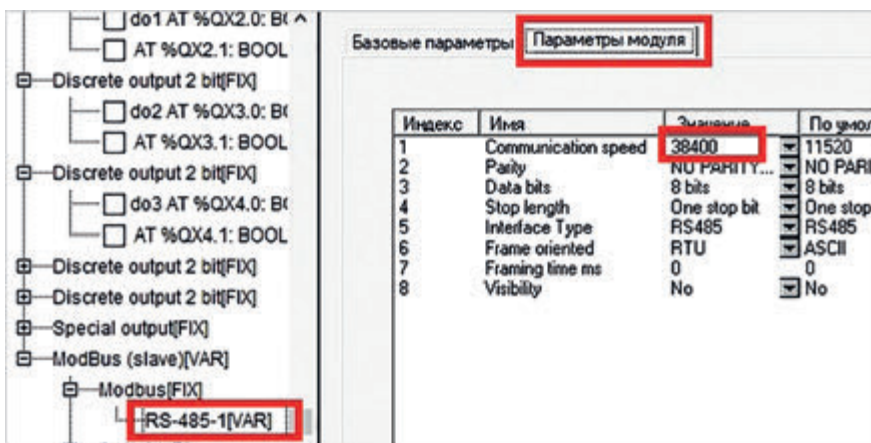


Рис. 7.5. Настройка скорости передачи данных с помощью интерфейса RS-485 для работы по Modbus RTU

Разработаем схему взаимодействия входов и выходов ПЛК, как показано на рис. 7.6.

Настройка узла RTM_1, проверка работоспособности собранной установки. В окне «Навигатор проекта» узла RTM_1 добавим модуль работы с Modbus RTU, как показано на рис. 7.7.

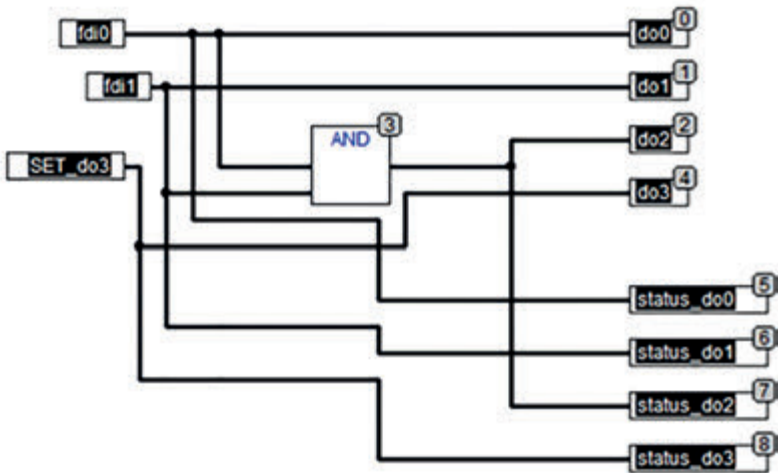


Рис. 7.6. Схема взаимодействия входов и выходов ПЛК

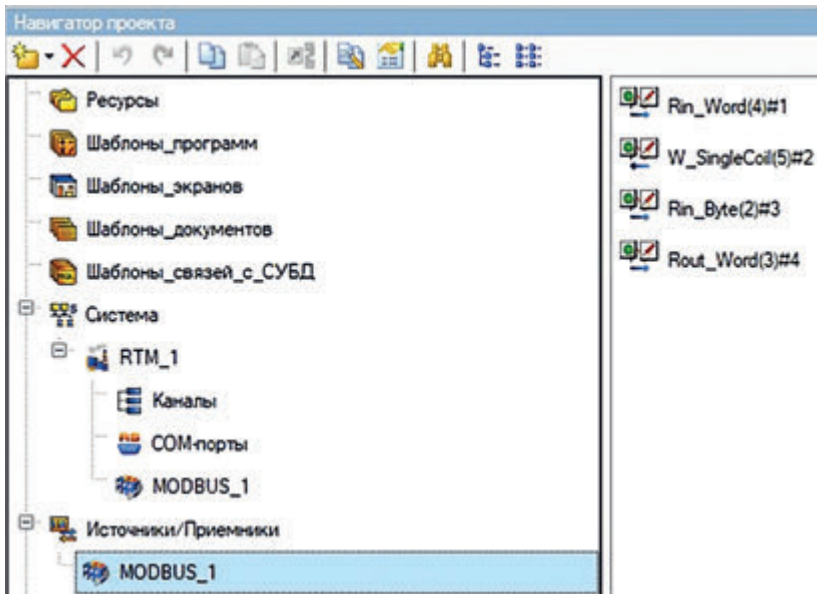


Рис. 7.7. Добавление модуля работы с Modbus RTU

Создадим регистры для работы с Modbus RTU, как показано на рис. 7.8. Сконфигурируем созданные регистры, как показано на рис. 7.9 и 7.10.

По аналогии сконфигурируем остальные регистры. Перенесем созданные регистры Modbus RTU в каналы. Создадим Экран. В нем создадим аргументы и привяжем их, как показано на рис. 7.11.

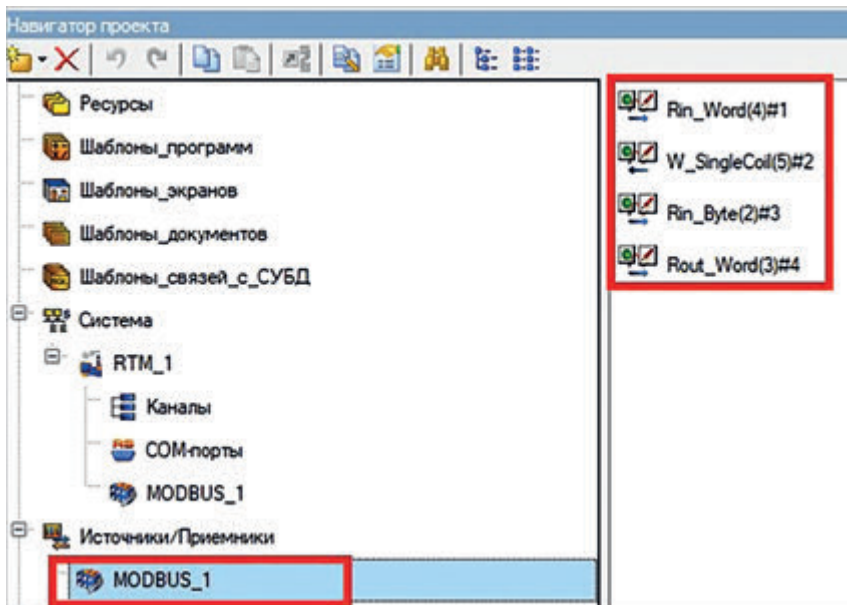


Рис. 7.8. Регистры для работы с Modbus RTU

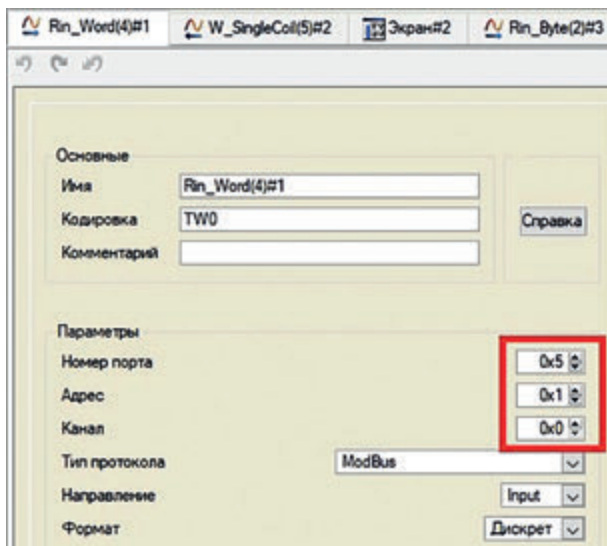


Рис. 7.9. Конфигурирование регистра чтения

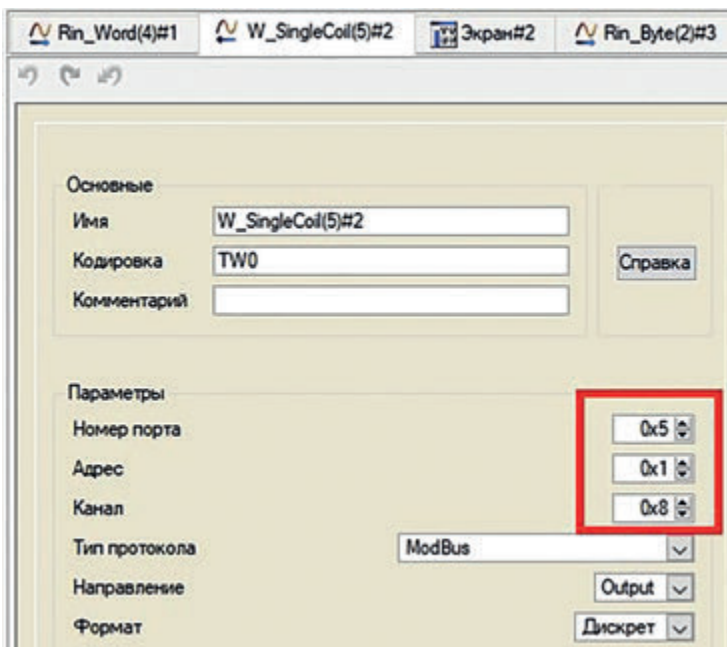


Рис. 7.10. Конфигурирование регистра записи

Имя	Тип	Тип данных	Привязка
status_do0	IN	BOOL	16 Rin_Word(4)#1:Бит 1 (Система.RTM_1.MODBUS_1)
status_do1	IN	BOOL	16 Rin_Word(4)#1:Бит 2 (Система.RTM_1.MODBUS_1)
status_do2	IN	BOOL	16 Rin_Word(4)#1:Бит 3 (Система.RTM_1.MODBUS_1)
status_do3	IN	BOOL	16 Rin_Word(4)#1:Бит 4 (Система.RTM_1.MODBUS_1)
Set_do3	IN/OUT	BOOL	16 W_SingleCoil(5)#2:Входное значение (Система.RTM_1.MODBUS_1)
st_do0_1	IN	BOOL	16 Rin_Byte(2)#3:Бит 1 (Система.RTM_1.MODBUS_1)
st_do0_2	IN	BOOL	16 Rout_Word(3)#4:Бит 1 (Система.RTM_1.MODBUS_1)

Рис. 7.11. Аргументы Экрана и их привязки

Создадим четыре элемента «Текст» и четыре элемента «Выключатель 4». Сделаем к ним привязку созданных аргументов Экрана и сконфигурируем, как показано на рис. 7.12.

Создадим на Экране элемент «Выключатель 0», сконфигурируем его в соответствии с рис. 7.13.

Сконфигурируем COM-порт для работы по RS-485 и Modbus RTU, как показано на рис. 7.14. Помните: номер COM-порта необходимо узнать в диспетчере устройств.

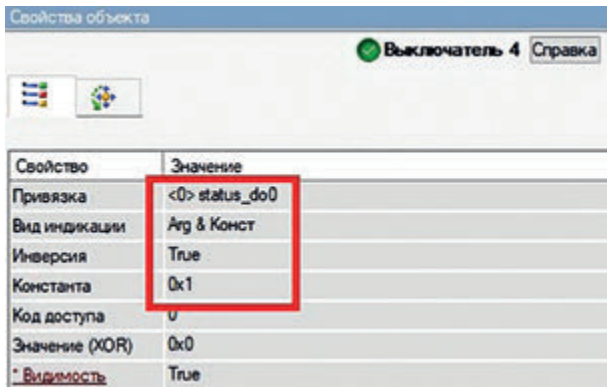


Рис. 7.12. Привязка созданных аргументов Экрана

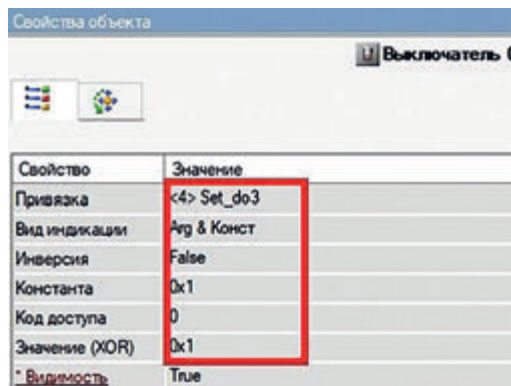


Рис. 7.13. Конфигурирование элемента «Выключателя 0»

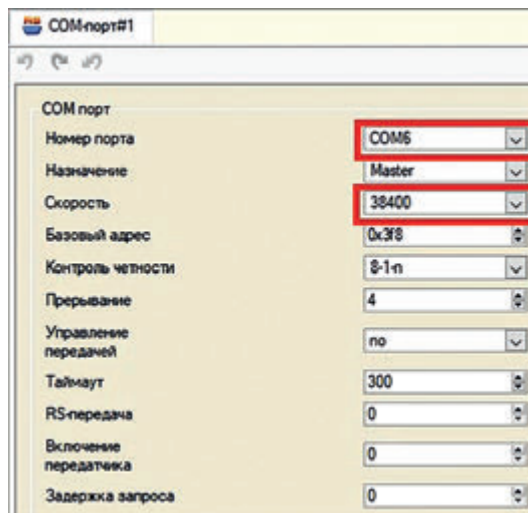


Рис. 7.14. Конфигурирование COM-порта

Проверим работоспособность созданного проекта и для этого запустим эмуляцию портов. Убедимся, что все входы/выходы находятся в выключенном положении. Подадим «питание» к одному входу и пронаблюдаем, как изменяются его состояние и состояние соответствующего выхода. Включим профайлер и проконтролируем состояние созданных элементов «Выключатель 4». Переведем тумблер элемента «Выключатель 0» в другое состояние и зафиксируем изменения в эмуляции портов на ПЛК.

В качестве отчета по лабораторной работе преподавателю предоставляются информацию о созданной АУИС с применением ПЛК и реализации связи с ПК по протоколу Modbus RTU, а также о ходе выполнения работы. Студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Задание для самоконтроля и подготовки к защите лабораторной работы № 7

Доработайте созданный проект:

- воспользовавшись группой «Источники» и компонентом «Синусоида», создайте генератор синусоиды и перенесите его в канал RTM1. Установите диапазон допустимых значений. При значениях менее допустимого должен загораться один выход, при значениях больше максимального — второй. При нажатии на элемент «Выключатель 0» должна отключаться вся система сигнализации;
- добавьте на Экран элементы индикации перехода системы за граничные значения.

Лабораторная работа № 8

РАЗРАБОТКА СИСТЕМЫ ДИСПЕТЧЕРИЗАЦИИ, ПОЛУЧАЮЩЕЙ И ОТОБРАЖАЮЩЕЙ ТЕКУЩИЕ ЗНАЧЕНИЯ ОСВЕЩЕННОСТИ ПОМЕЩЕНИЯ И ТЕМПЕРАТУРЫ ОКРУЖАЮЩЕГО ВОЗДУХА, ПОЛУЧЕННЫЕ ПО ПРОТОКОЛУ MODBUS

Цели работы:

- 1) научиться разрабатывать схемы подключения датчиков температуры и освещенности к SCADA-системе;
- 2) научиться считывать данные внешних устройств по протоколу Modbus и выводить их на Экран оператора.

План работы

1. Создание Экрана и размещение на нем элементов визуализации.
2. Конфигурирование протокола Modbus.
3. Подключение внешних устройств к ПК и проверка работоспособности собранной системы.

Создание Экрана и размещение на нем элементов визуализации. В соответствии с алгоритмом, приведенным в лабораторной работе № 1, создадим стандартный проект, узел RTM и Экран.

Изменим время пересчета для дальнейшего плавного отображения значений параметров подключенных внешних устройств. Для этого щелкнем ПКМ на узле RTM_1 и в появившемся окне выберем строку «Редактировать». В новом появившемся окне во вкладке «Основные» установим время пересчета: «Период» — 1, «Разрешение» — 0,01.

Добавим на Экран элементы «Ползунок», «Текст» и два «Стрелочных прибора», затем создадим аргументы Экрана:

- 1) «Диод_вых» типа IN, USINT для отработки подачи команд со SCADA-системы на внешние устройства (на светодиод);
- 2) «temp» типа IN, USINT для отработки приема данных в SCADA-системе от внешних устройств (с датчика температуры и освещенности).

Привяжем эти аргументы к созданным элементам: «Диод_вых» — к элементу «Ползунок», «temp» — к элементам «Стрелочный прибор».

Конфигурация протокола Modbus. Создадим компонент COM-порты в RTM_1 и сконфигурируем его. Для этого щелкнем ПКМ на RTM_1, выберем строку «Создать_группу» и затем элемент «COM-порты». Сконфигурируем созданный элемент «COM-порт1», как показано на рис. 8.1.

Зайдем в «Источники» и создадим группу «MODBUS», как показано на рис. 7.7. Там создадим два компонента: W_Float(16) и ROUT_Word(3). Перенесем их в каналы RTM_1, привяжем W_Float (16) к аргументу Экрана «Диод_вых», а ROUT_Word(3) — к аргументу Экрана «temp».

Установим соответствующие параметры регистров чтения и записи, как показано на рис. 8.2 и рис. 8.3.

Сконфигурируем COM-порт, как показано на рис. 7.14.

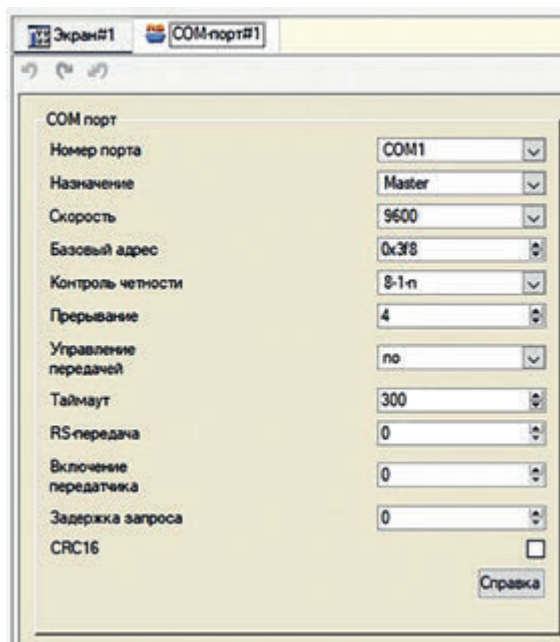


Рис. 8.1. Конфигурирование элемента «COM-порт1»

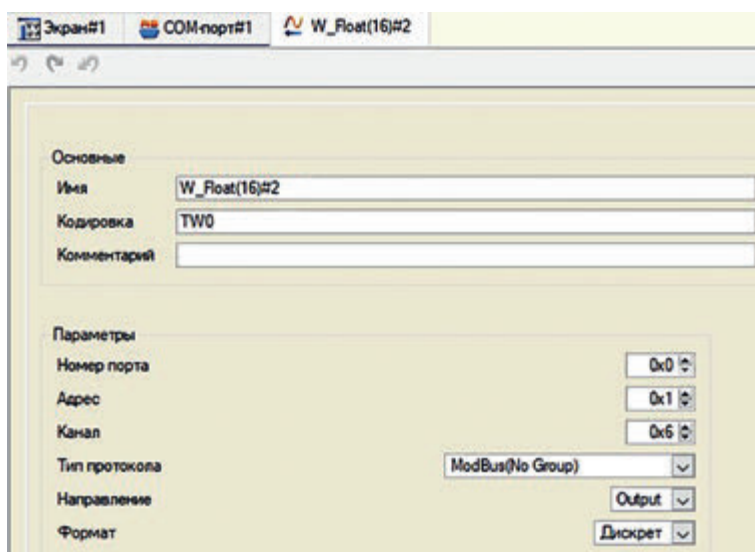


Рис. 8.2. Установка параметров регистра чтения

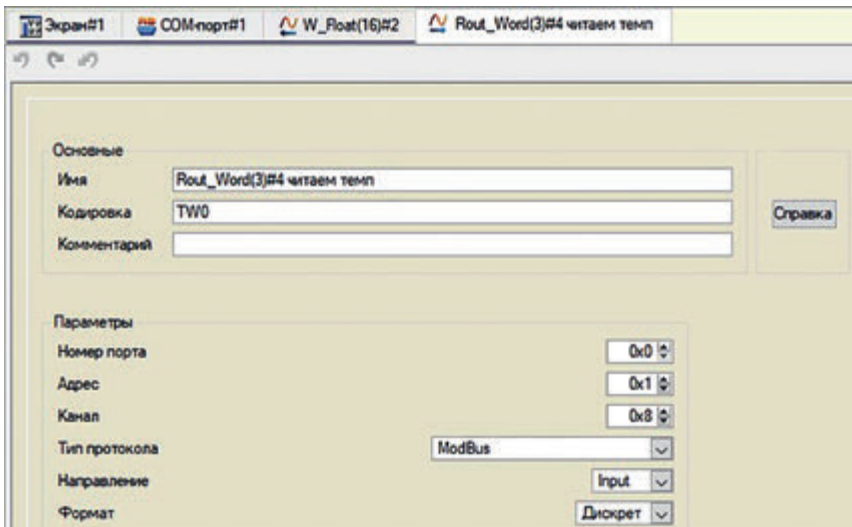


Рис. 8.3. Установка параметров регистра записи

Подключение внешних устройств к ПК и проверка работоспособности собранной системы. Подключим к ПК внешнее устройство, например плату Arduino nano. Подключим светодиод к входу D9 платы Arduino nano, затем датчик температуры DS18B20 к входу D8 платы Arduino nano.

Запустим профайлер и проверим работоспособность созданного проекта (рис. 8.4).



Рис. 8.4. Система отслеживания текущих показателей температуры и освещенности помещения

В качестве отчета по лабораторной работе преподавателю предоставляют информацию о созданной АИУС, получающей и отображающей текущие значения освещенности помещения и температуры окружающего воздуха, полученные по протоколу Modbus, и о ходе выполнения работы. Студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Задания для самоконтроля и подготовки к защите лабораторной работы № 8

Создайте новый проект, в котором будет включаться светодиод сигнала тревоги при превышении заданного значения температуры. Для визуализации процесса выведите на элемент «Тренд» температуру и ее допустимый диапазон.

Лабораторная работа № 9

ПРОЕКТИРОВАНИЕ ОПЕРАТОРСКОГО ИНТЕРФЕЙСА СРЕДСТВАМИ MASTERSCADА ПО ТЕМЕ КУРСОВОГО ПРОЕКТА

Цели работы:

- 1) научиться создавать главный узел системы — автоматизированное рабочее место (АРМ);
- 2) освоить навыки применения элементов «СолнцеСнежинка», «Воздуходувка», «Кнопка» и «Текст»;
- 3) изучить процесс создания и привязки параметров к свойствам элементов.

План работы

1. Создание главного узла — системы в MasterSCADA.
2. Отработка эмуляции проекта.
3. Создание проекта «Кондиционер».

Создание главного узла — системы в MasterSCADA. Универсальная SCADA-система MasterSCADA предназначена для автоматизации и диспетчеризации объектов во всех отраслях промышленности. Объект в MasterSCADA — именованная совокупность графического представления технологического объекта, его параметров, алгоритмов контроля и управления, окон управления и других доступных элементов проекта (в том числе других объектов).

Щелчком ПКМ на элементе «Система», выберем строку «Добавить» и строку «АРМ», как показано на рис. 9.1.

Создадим элементы «Объект 1» и «Окно 1», последний элемент сделаем стартовым, как показано на рис. 9.2.

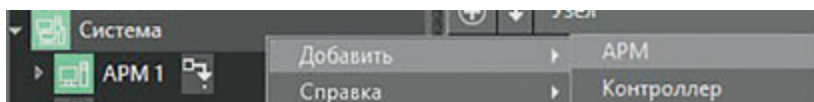


Рис. 9.1. Создание АРМ

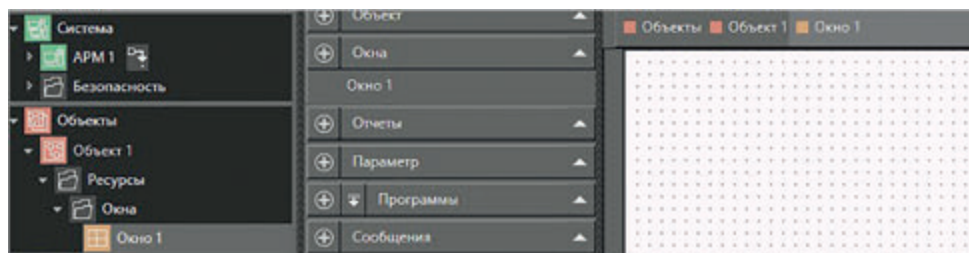


Рис. 9.2. Создание элементов «Объект 1» и «Окно 1»

Добавим в созданный «Объект 1» два параметра вещественного типа и свяжем их между собой (первый будет источником второго). Для этого щелкнем ПКМ на элементе «Объект 1», выберем строку «Добавить», затем «Параметр», как показано на рис. 9.3.

Перетащим с помощью ПКМ на Экран элемент «Параметр 1», добавим на Экран элемент «Текст», привяжем к нему второй параметр. Результат показан на рис. 9.4.

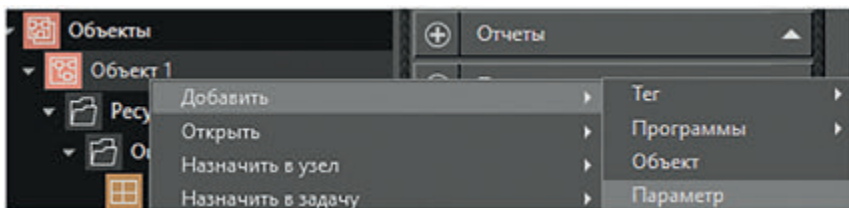


Рис. 9.3. Процесс создания параметров



Рис. 9.4. Элементы «Параметр 1» и «Текст»



Отработка эмуляции проекта. Включим эмуляцию с помощью кнопки , введем значение «Параметра 1» в поле «ввод текста», оно появится в элементе «Текст». Добавим элементы «Инкремент» и «Стрелочный прибор», включим эмуляцию , после чего убедимся в работоспособности созданного проекта. Результат должен соответствовать представленному на рис. 9.5.





Рис. 9.5. Демонстрация работоспособности инкремента на Экране

Создание проекта «Кондиционер». Создадим пять параметров: «Влажность» и «Температура» с типом REAL, «Вкл» и «Отказ» с типом BOOL, «Сезон» с типом INT. Добавим на Экран элементы:

Вкл — элемент «Кнопка с фиксацией» (ПКМ с помощью *drag'n'drop*);

Влажность и Температура — элемент «Текст» с форматом {f2} % и С соответственно (см. Справку F1);

Сезон — элемент «Текст» с точечным типом конвертации.

Включим эмуляцию , введем значение в параметр «Влажность», которое появится в текстовом поле. Добавим элементы «Инкремент» и «Стрелочный прибор», включим эмуляцию , после чего убедимся в работоспособности созданного проекта.

Добавим на Экран элемент «Воздуходувка» на вкладке «Палитра» категории «BaseObjects» (рис. 9.6).

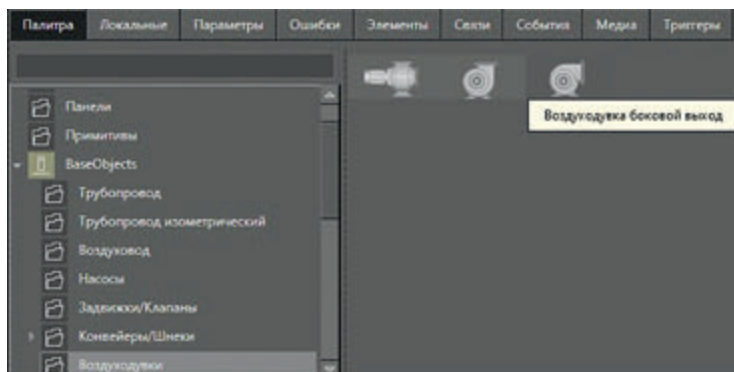


Рис. 9.6. Элемент «Воздуходувка»

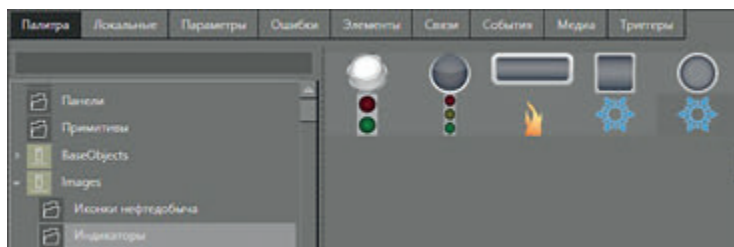


Рис. 9.7. Элементы библиотеки «Images» вкладки «Палитра»

Добавим на Экран элемент «СолнцеСнежинка» из библиотеки «Images» вкладки «Палитра», как показано на рис. 9.7.

Параметр «Вкл» свяжем со свойствами «Состояние» и «Вращение» элемента «Воздуходувка» (исходящая связь). Параметр «Сезон» свяжем со свойством «Сезон» элемента «СолнцеСнежинка», установив точечную конвертацию: 1 — Summer, 2 — Winter. Окно конвертера связей показано на рис. 9.8.

Свяжем параметр «Отказ» со свойством «Общая авария» элемента «Воздуходувка» и выберем «Точечная» в списке «Тип конвертации», как показано на рис. 9.9.

В результате будет создано окно оператора, показанное на рис. 9.10.

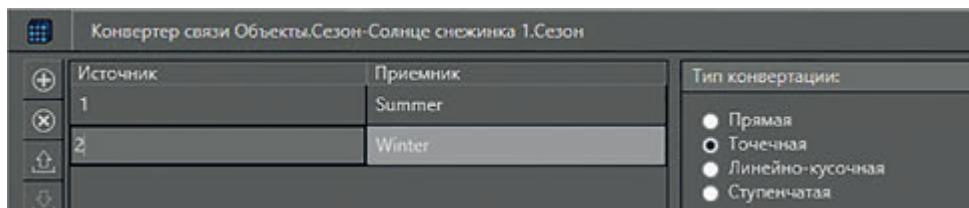


Рис. 9.8. Окно конвертера связей

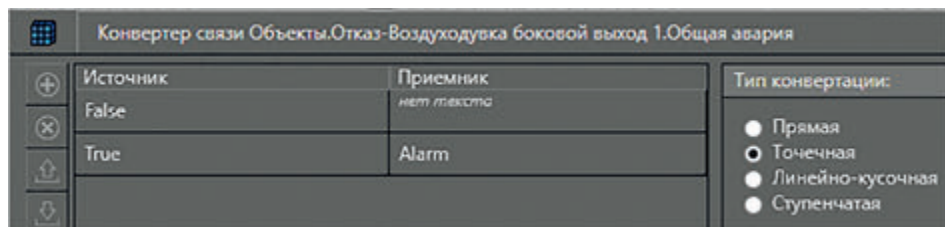


Рис. 9.9. Связь параметра «Отказ» со свойством «Общая авария» элемента «Воздуходувка»



Рис. 9.10. Окно оператора

Изменим значения параметров «Отказ» и «Сезон» и проверим работоспособность системы. В окне должны появиться значок солнца и красная рамка вокруг элемента «Воздуходувка» — сработает сигнализация ALARM. Результат работы системы приведен на рис. 9.11.

В качестве отчета по лабораторной работе преподавателю предоставляется информация о созданной АИУС с разработанным операторским интер-



Рис. 9.11. Результат работы системы

фейсом средствами MasterSCADA и о ходе выполнения работы. Студенты должны продемонстрировать работоспособность созданной системы. Требования к отчету приведены в конце настоящего пособия.

Задание для самоконтроля и подготовки к защите лабораторной работы № 9

Доработайте проект «Кондиционер»:

- добавьте на Экран элемент «Тренд» из палитры категории «Контролы»;
- выведите на элемент «Тренд» параметр «Влажность»;
- включите эмуляцию, убедитесь в работоспособности системы, изменив значение влажности.

Примечание: для изменения типа графика на ступенчатый необходимо открыть список свойств элемента «Тренд», в строке «Перья» выбрать необходимое перо, найти поле «Тип графика» и выбрать «Ступенчатый».

Состав отчета о выполнении лабораторной работы

1. Цели лабораторной работы.

В этом разделе студент приводит перечень целей лабораторной работы, указанных преподавателем.

2. Ход выполнения работы.

В этом разделе приводится пошаговое описание процесса выполнения лабораторной работы с прикреплением изображений, демонстрирующих промежуточные результаты, полученные в ходе выполнения работы.

3. Выводы.

В этом разделе описываются и анализируются те результаты, которые были получены в процессе выполнения лабораторной работы.

Литература

Одинокое В.В., Хабибулина Н.Ю. Автоматизированные информационно-управляющие системы : учебное пособие. Томск: Томский государственный университет систем управления и радиоэлектроники, 2014. 129 с. Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. URL: <https://www.iprbookshop.ru/72068.html>. Режим доступа: для авторизированных пользователей.

Сумительнов В.Н. Автоматизированные информационно-управляющие системы: учебное пособие. М.: ГОУ ВПО МГУЛ, 2007. 163 с.

Содержание

Предисловие	3
Список сокращений	4
Введение	5
Лабораторная работа № 1. Создание проекта диспетчеризации с применением компонентов визуализации «Кнопка», «Ползунок», «Тренд» и «Текст»	8
Лабораторная работа № 2. Создание графических объектов. Тиражирование их на мониторе реального времени. Привязка каналов	20
Лабораторная работа № 3. Создание программы работы декадного четырехразрядного счетчика на языке ST	26
Лабораторная работа № 4. Создание программы работы декадного четырехразрядного счетчика на языке FBD	32
Лабораторная работа № 5. Работа с базой данных	36
Лабораторная работа № 6. Отчет тревог и запуск приложений Windows ..	42
Лабораторная работа № 7. Организация связи ПЛК с АРМ диспетчера по протоколу Modbus RTU	51
Лабораторная работа № 8. Разработка системы диспетчеризации, получающей и отображающей текущие значения освещенности помещения и температуры окружающего воздуха, полученные по протоколу Modbus	59
Лабораторная работа № 9. Проектирование операторского интерфейса средствами MasterSCADA по теме курсового проекта	63
Состав отчета о выполнении лабораторной работы	68
Литература	69

Учебное издание

Поленов Дмитрий Юрьевич

Автоматизированные информационно-управляющие системы

Оригинал-макет подготовлен в Издательстве МГТУ им. Н.Э. Баумана.

В оформлении использованы шрифты Студии Артемия Лебедева.

Подписано в печать 20.11.2023. Формат 70×100/16.
Усл. печ. л. 5,69. Тираж 176 экз. Изд. № 1281-2023.

Издательство МГТУ им. Н.Э. Баумана.
105005, г. Москва, улица 2-я Бауманская, д. 5, стр. 1.
info@bmstu.press
<https://press.bmstu.ru>

Отпечатано в типографии МГТУ им. Н.Э. Баумана.
105005, г. Москва, улица 2-я Бауманская, д. 5, стр. 1.
baumanprint@gmail.com