

А. И. Новиков
М. А. Воропанова

МОДЕЛИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ

Практикум

Санкт-Петербург
2022

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«Санкт-Петербургский государственный университет
промышленных технологий и дизайна»**
Высшая школа технологии и энергетики
Кафедра информационно-измерительных технологий и систем управления

А. И. Новиков
М. А. Воропанова

МОДЕЛИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ

Практикум

Утверждено Редакционно-издательским советом ВШТЭ СПбГУПТД

Санкт-Петербург
2022

УДК 62-5(07)
ББК 32.965я7
Н 731

Рецензент

кандидат технических наук, доцент, заведующий кафедрой АТПиП ВШТЭ
Санкт-Петербургского государственного университета промышленных технологий и дизайна
Д. А. Ковалёв

Новиков, А. И., Воропанова, М. А.

Н 731 Моделирование систем управления: Практикум / А. И. Новиков, М. А. Воропанова. — СПб.: ВШТЭ СПбГУПТД, 2022. — 46 с.

Практикум соответствует программам и учебным планам дисциплин «Моделирование систем управления», «SCADA-системы», «Математическое моделирование робототехнических систем», «Моделирование объектов управления для робототехнических систем», «Моделирование объектов управления», «Математическое моделирование автоматизированных и цифровых производств», «Математическое моделирование систем автоматического управления», «Интегрированные системы проектирования и управления» для студентов, обучающихся по направлениям подготовки: 01.03.02 «Прикладная математика и информатика», 15.03.04 «Автоматизация технологических процессов и производств», 15.04.04 «Автоматизация технологических процессов и производств» и 27.03.04 «Управление в технических системах». В практикуме собраны упражнения для выполнения лабораторных работ по дисциплинам. Приведены примеры работы со средой разработки CoDeSys 3.5.

Практикум предназначен для подготовки бакалавров и магистров очной и заочной форм обучения. Отдельные разделы могут быть полезны аспирантам и специалистам, работающим в области автоматизации.

УДК 681.3(06)
ББК 32.965я7

© ВШТЭ СПбГУПТД, 2022
© Новиков А.И., Воропанова М.А., 2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ПЕРЕЧЕНЬ ПРИНЯТЫХ ОБОЗНАЧЕНИЙ	4
1. НАЧАЛО РАБОТЫ	5
1.1. Описание контроллера	5
1.2. Установка программного обеспечения.....	6
1.3. Создание проекта.....	7
1.4. Состав проекта	8
Контрольные вопросы	8
2. РАБОТА С ПРОГРАММОЙ НА CFC	10
2.1. Постановка задачи	11
2.2. Пример программы на CFC	12
2.3. Расположение элементов в программе.....	12
Контрольные вопросы	13
3. РАЗРАБОТКА ЭКРАНОВ.....	14
3.1. Редактор экранов	14
3.2. Связывание экрана с программой.....	14
3.3. Дополнительные возможности при создании экранов	17
Контрольные вопросы	21
4. ЗАПУСК ЭМУЛЯЦИИ.....	22
Контрольные вопросы	23
5. СОЗДАНИЕ СОБСТВЕННОГО БЛОКА НА ST	24
5.1. Постановка задачи	25
5.2. Пример программы на ST	25
5.3. Редактор ST	26
5.4. Использование созданного блока.....	27
Контрольные вопросы	27
6. СИСТЕМЫ РЕГУЛИРОВАНИЯ	28
6.1. ПИД-регулятор.....	28
6.2. Модель объекта	31
6.3. Модель системы.....	32
6.4. Ввод и отображение числовых значений	33
6.5. Отображение графиков	36
6.6. Итоговый экран.....	39
6.7. Транспортное запаздывание	39
Контрольные вопросы	41
7. ЛАБОРАТОРНЫЕ РАБОТЫ	41
7.1. Содержание отчета	41
7.2. Лабораторная работа 1	41
7.3. Лабораторная работа 2	41
7.4. Лабораторная работа 3	41
8. КУРСОВАЯ РАБОТА (ПРОЕКТ).....	44
8.1. Содержание курсовой работы (проекта)	44
8.2. Пример задания на курсовую работу (проект)	44
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	46

ВВЕДЕНИЕ

В настоящее время много внимания уделяется технологиям создания программного обеспечения для систем управления промышленной автоматикой, построенных на базе программируемых логических контроллеров (ПЛК), и практическому программированию на языках стандарта Международной электротехнической комиссии (МЭК) 61131-3.

Одной из систем для разработки программного обеспечения для ПЛК является программный комплекс CoDeSys. Он разработан компанией 3S-Smart Software Solutions GmbH (3S). Основным его назначением является программирование ПЛК и промышленных компьютеров в стандарте МЭК 61131-3. CoDeSys является штатным инструментом программирования ПЛК ведущих европейских изготовителей: ABB, Beckhoff, Beck IPC, Berger Lahr, Bosch Rexroth, ifm, Keb, Kontron, Lenze, Moeller, WAGO, Fastwel и др., поддерживая таким образом множество различных контроллеров от различных производителей.

Человек, прошедший обучение по программе, включающей стандарт МЭК 61131-3, сможет работать с ПЛК любой фирмы. В то же время, если он имел ранее опыт работы с любыми ПЛК, его навыки окажутся полезными и существенно упростят изучение новых возможностей.

ПЕРЕЧЕНЬ ПРИНЯТЫХ ОБОЗНАЧЕНИЙ

!!! — Обратить особое внимание

ПЛК — Программируемый логический контроллер

ПО — Программное обеспечение

1. НАЧАЛО РАБОТЫ

1.1. Описание контроллера

Программируемый логический контроллер (ПЛК) ОВЕН СПК110 (рис. 1) является устройством класса человеко-машинного интерфейса со встроенными функциями свободно программируемого контроллера [1].



Рис. 1. Внешний вид СПК110

Отличительной особенностью данного контроллера является то, что панель оператора и ПЛК объединены в единый корпус.

Контроллер работает на встроенной операционной системе Linux. СПК110 имеет два независимых последовательных интерфейса RS-232 и RS-485, поддерживает протоколы Modbus и OWEN. Диагональ графического экрана равна 10 дюймам.

1.2. Установка программного обеспечения

Для работы в среде разработки CoDeSys 3.5 необходимо установить следующие файлы (рис. 2) [2].

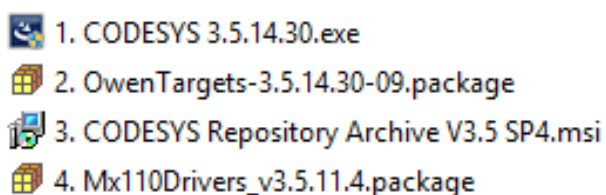


Рис. 2. Файлы, необходимые для выполнения работы

!!! Необходимо использовать указанные версии программного обеспечения (ПО).

После установки в меню Инструменты / «**Менеджер пакетов**» должно отображаться следующее (рис. 3):




Имя	Версия
 CODESYS SoftMotion	4.5.1.0
 OwenTargets	3.5.14.3009
 Модули ввода\вывода компании OWEN	3.5.11.4

Рис. 3. Менеджер пакетов

1.3. Создание проекта

Порядок создания нового проекта следующий.

1. Меню Файл / «Новый проект».
2. Создать «Стандартный проект» (рис. 4).

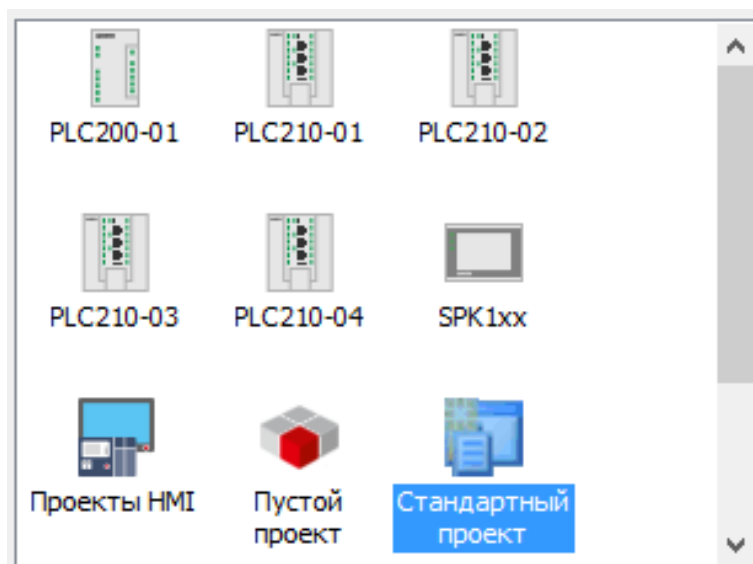


Рис. 4. Создание стандартного проекта

3. Выбрать контроллер **СПК110** и язык **CFC** (рис. 5).

Язык CFC — это модификация языка FBD (т.е. функциональных блоков, как например в Simulink MatLab).

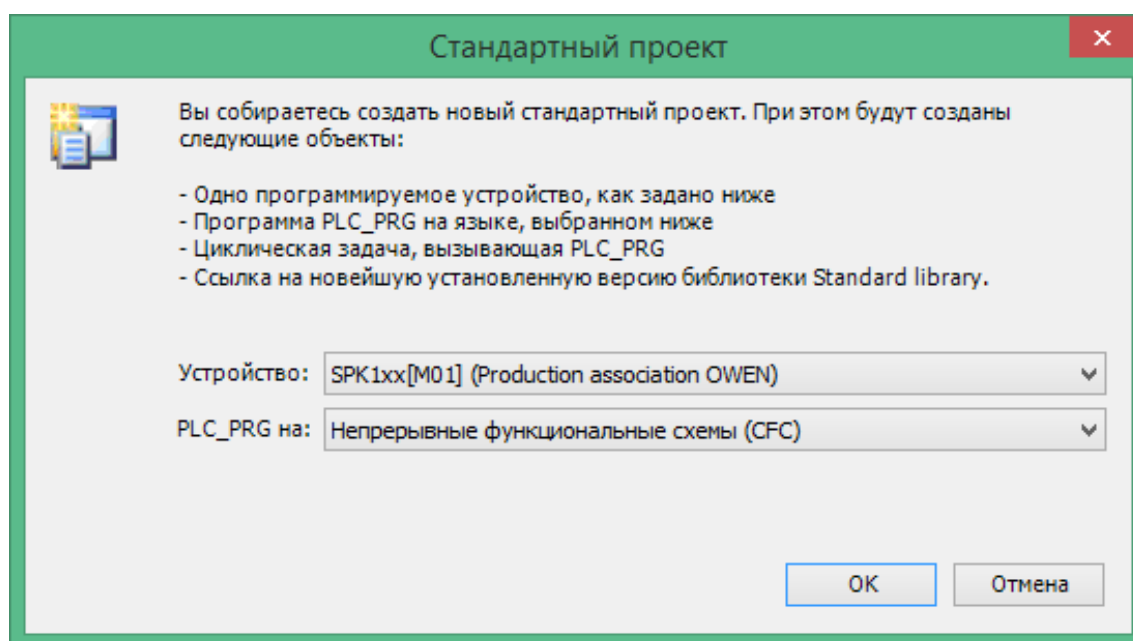


Рис. 5. Выбор контроллера и языка программирования

4. Добавить в проект экран (рис. 6).
5. Запустить Компиляцию (F11) и убедиться, что отсутствуют ошибки.

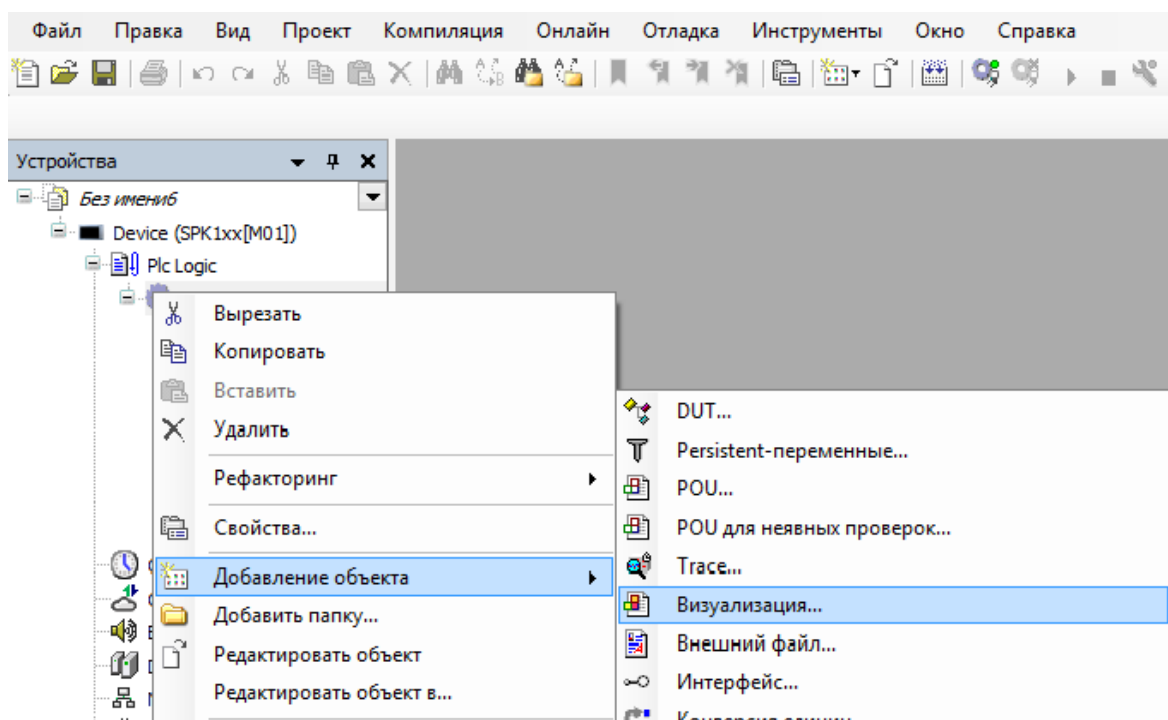


Рис. 6. Добавление экрана в проект

1.4. Состав проекта

Проект состоит из основных объектов, представленных на рис. 7.

!!! Не все из перечисленных объектов создаются автоматически.

!!! Имена **Block1**, **Block2**, **Prog1**, **Visualization1**, **Visualization2** приведены для примера, в реальности нужно давать более осмысленные имена (например, Главный экран лучше назвать **Main**).

Контрольные вопросы

1. Какие технические характеристики ПЛК СПК110 Вы можете перечислить?
2. Что такое CFC?
3. Какие объекты в составе проекта создаются автоматически?
4. Почему имена объектам программы нужно давать осмысленно?

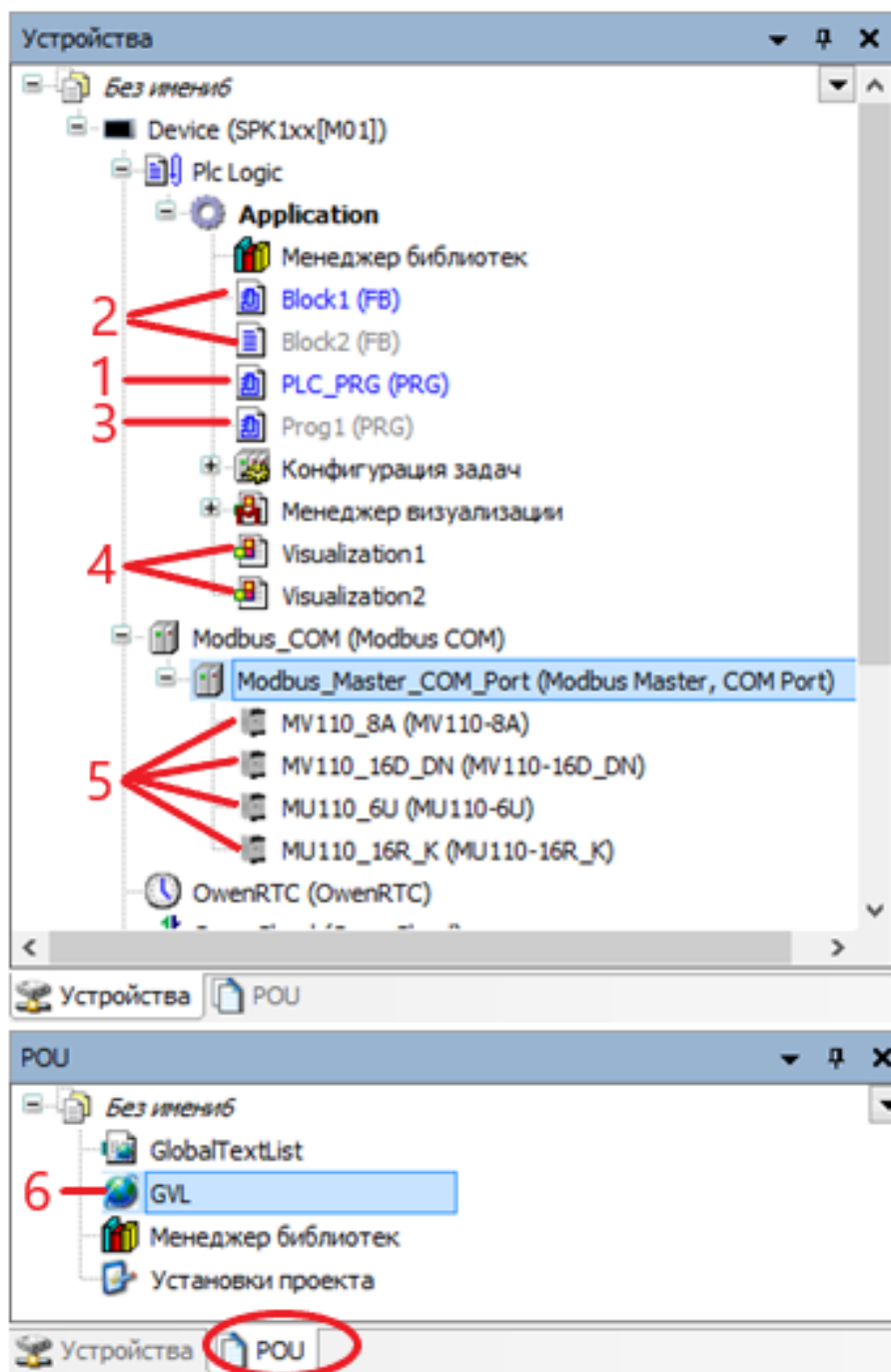


Рис. 7. Состав проекта:

- 1 – главная программа; 2, 3 – функциональные блоки и подпрограммы;
- 4 – экраны; 5 – модули ввода/вывода;
- 6 – список глобальных переменных

2. РАБОТА С ПРОГРАММОЙ НА CFC

Редактор программы CFC имеет внешний вид, показанный на рис. 8, и состоит из:

- графического редактора;
- панели инструментов с добавляемыми блоками (на рис. 8 выделены 4 наиболее используемых блока);
- списка переменных, который может редактироваться вручную, но в большинстве случаев он заполняется автоматически [3].

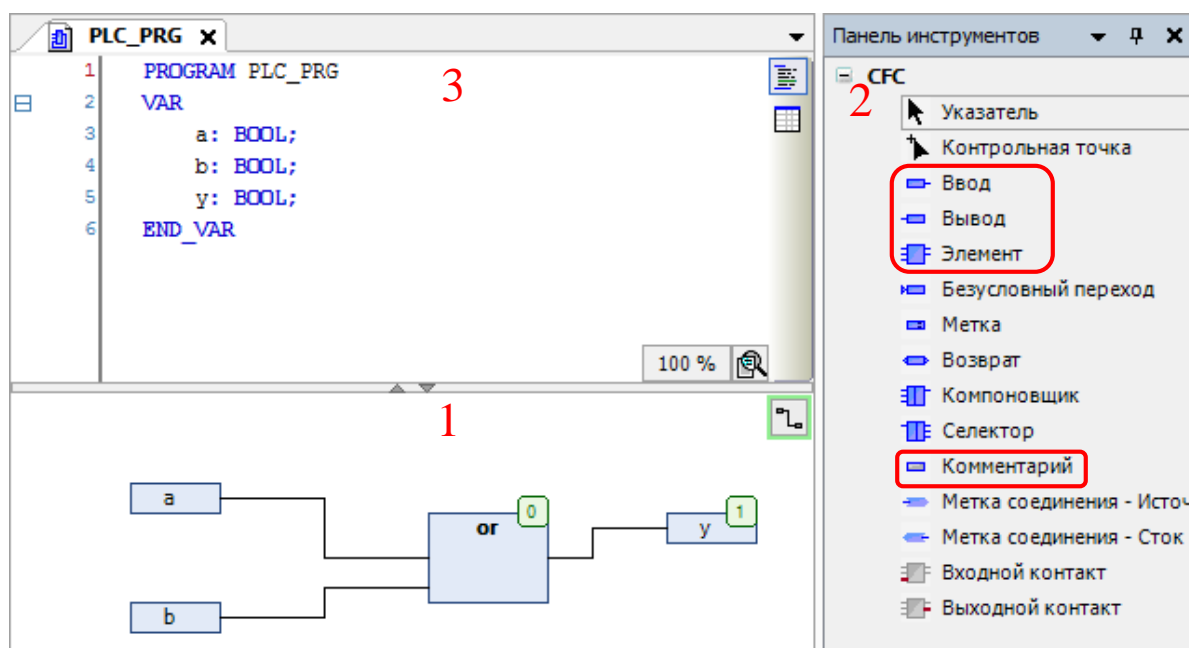


Рис. 8. Внешний вид редактора программы:
1 – графический редактор,
2 – панель инструментов, 3 – список переменных

После добавления «Ввода» (или «Вывода») достаточно написать в нем имя переменной с клавиатуры. Если этого имени еще нет в списке переменных, то программа предложит его добавить, открыв окно Автообъявление (рис. 9).

Аналогично для «Элемента», можно написать имя с клавиатуры (например: **not**, **and**, **or**, **xor**, **abs**, **add**, **mul**, **div** и др.) или нажать многоточие и выбрать из списка (рис. 10).

!!! Некоторые блоки, такие как **ton** или **rs**, требуют также указания имени над блоком (уникального для каждого блока). Эти имена будут добавлены в список переменных.

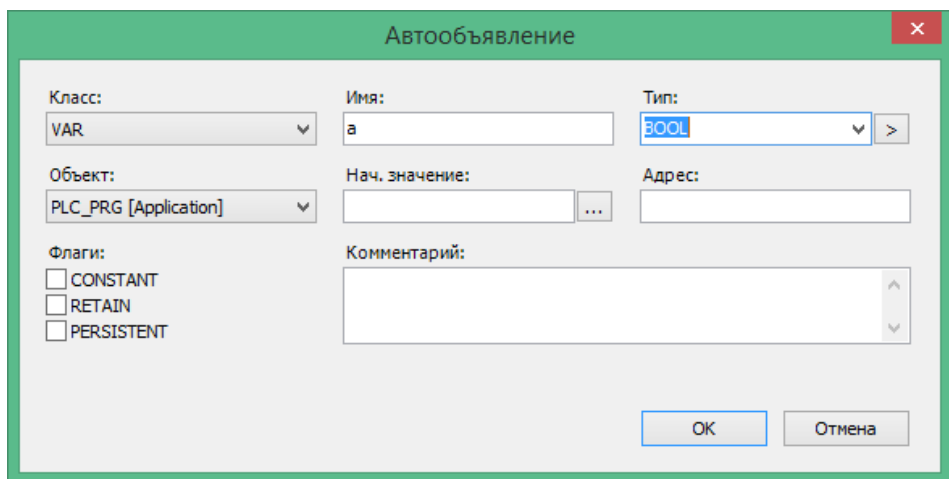


Рис. 9. Добавление переменной с помощью окна Автообъявление

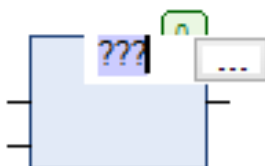


Рис. 10. Добавление имени для «Элемента»

2.1. Постановка задачи

Пусть необходимо написать простейшую программу управления вентиляцией, реализующую следующие действия:

- запуск вентилятора кнопками **ПУСК/СТОП** с экрана оператора;
- задержка включения вентилятора на **3 сек**;
- отключение вентилятора при поступлении сигнала **Пожар**.

В программе необходимо использовать переменные, представленные в табл. 1.

!!! Данная таблица содержит как вводы/выводы контроллера (верхние две строки), так и переменные для связи с экраном (нижние две строки). В реальности такой таблицы дано не будет, и определить количество и тип переменных для связи с экраном должен будет сам программист.

Таблица 1 — Необходимые переменные для создаваемой программы

DI	Fire	Сигнал пожар
DO	Vent	Запуск вентилятора
DI	Start	Кнопка СТАРТ (на экране)
DI	Stop	Кнопка СТОП (на экране)

2.2. Пример программы на CFC

Программа на языке CFC, реализующая описанное ранее задание, выглядит, как показано на рис. 11.

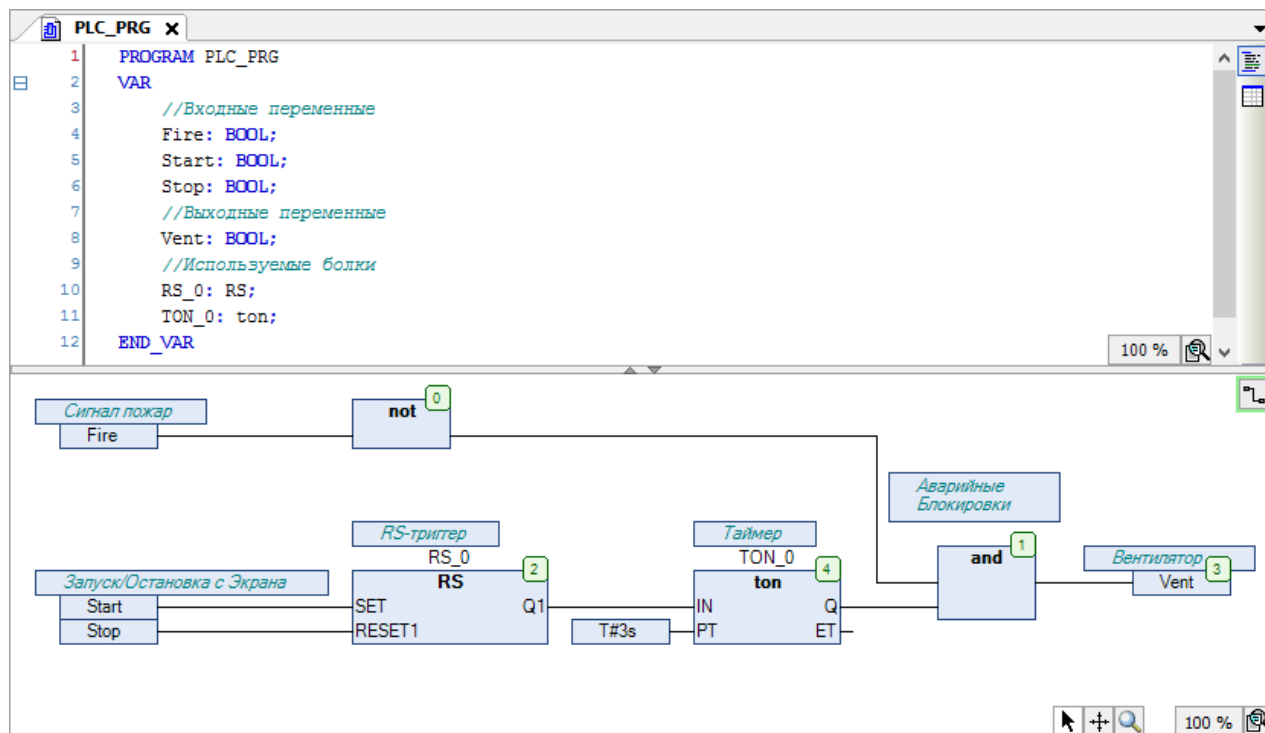


Рис. 11. Внешний вид созданной программы

!!! Комментарии на русском языке обязательны!

2.3. Расположение элементов в программе

При создании программы на языках программирования подобных FBD (в том числе и CFC) следует придерживаться следующих рекомендаций по расположению элементов:

- все входы должны находиться в левой части программы;
- все выходы должны располагаться в правой части программы;
- в верхней части программы следует размещать части программы, относящиеся к авариям;
- элементы программы необходимо располагать в порядке слева направо в следующем порядке: элементы, относящиеся к основной программе; элементы, относящиеся к ручному управлению; элементы блокировок.

Все перечисленные выше рекомендации изображены схематически на рис. 12.

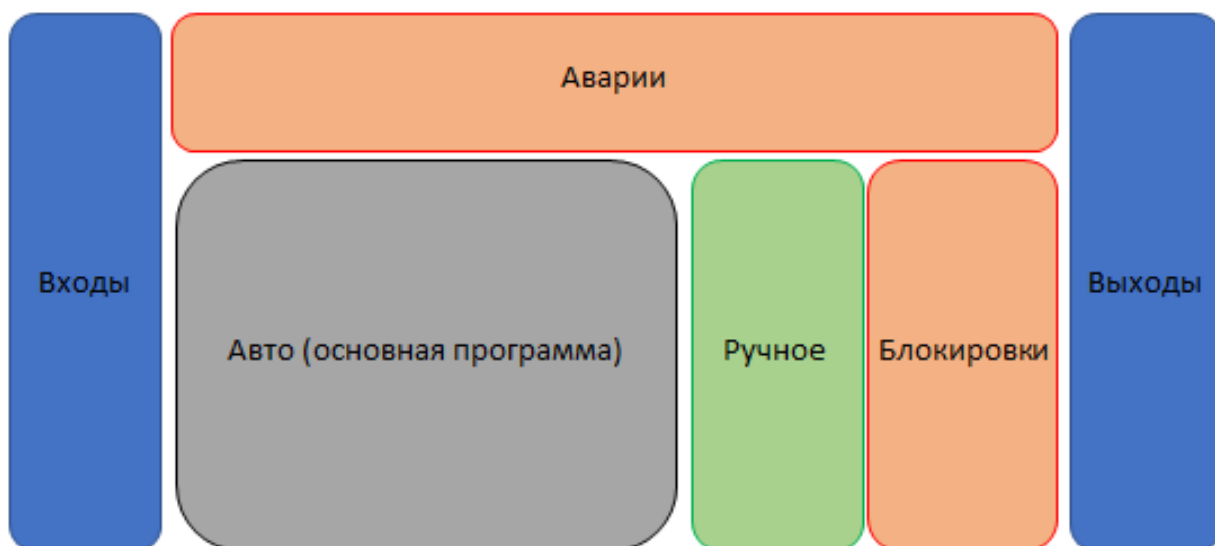


Рис. 12. Расположение элементов программы

Контрольные вопросы

1. Из чего состоит редактор программы CFC?
2. Как добавить и объявить в программе Ввод/ Вывод и Элемент?
3. Для чего нужен RS-триггер в программе на рис. 11?
4. Опишите, почему рекомендуется располагать элементы программы, относящиеся к блокировкам, после элементов, относящихся к ручному управлению?

3. РАЗРАБОТКА ЭКРАНОВ

3.1. Редактор экранов

Необходимо разработать экран, представленный на рис. 13.

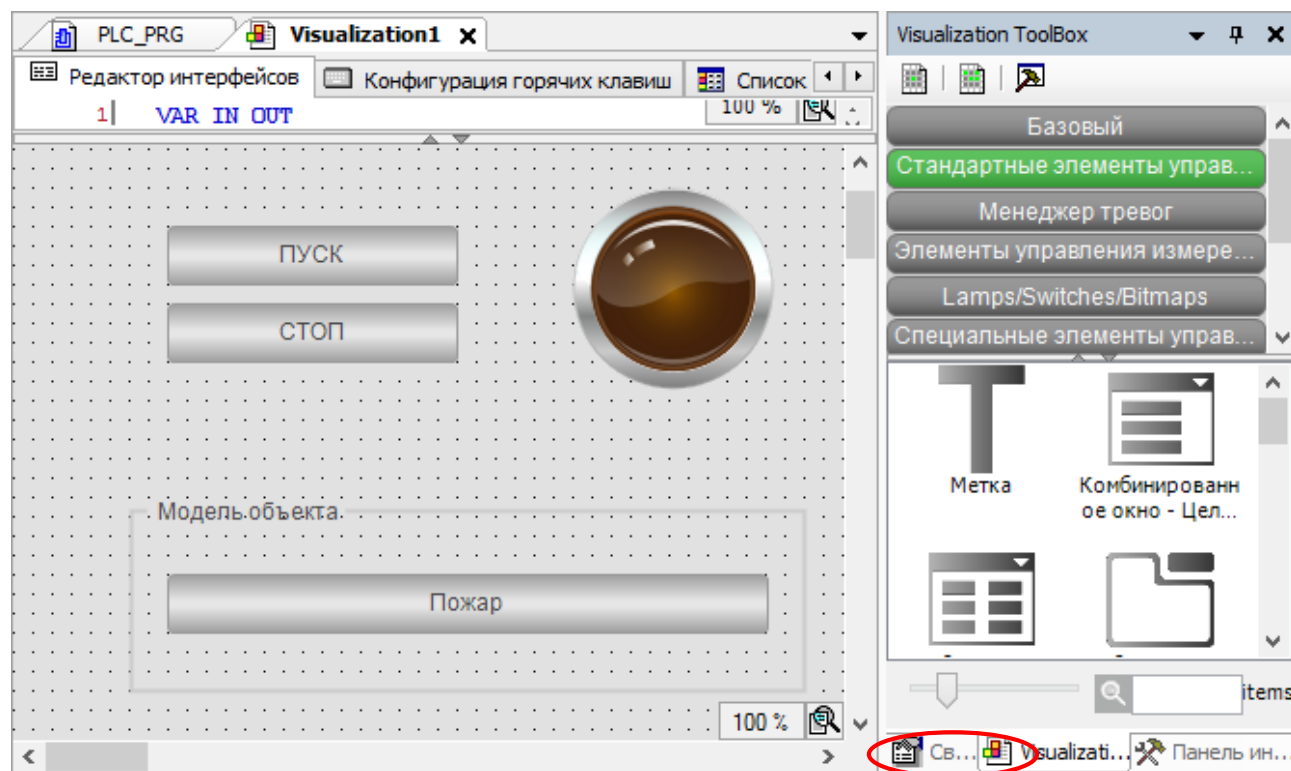


Рис. 13. Пример экрана для создаваемой программы

Он состоит из трех кнопок, индикатора и «группы». Все элементы расположены в панели справа.

3.2. Связывание экрана с программой

1. Выделив кнопку **ПУСК**, следует выбрать в её свойствах **OnMouseDown** (нажатие кнопки мыши) (рис. 14).

2. Затем необходимо привязать к кнопке выполнение кода на языке **ST** (рис. 15).

В качестве кода для **OnMouseDown** указана команда включения переменной:

```
PLC_PRG.Start := 1;
```

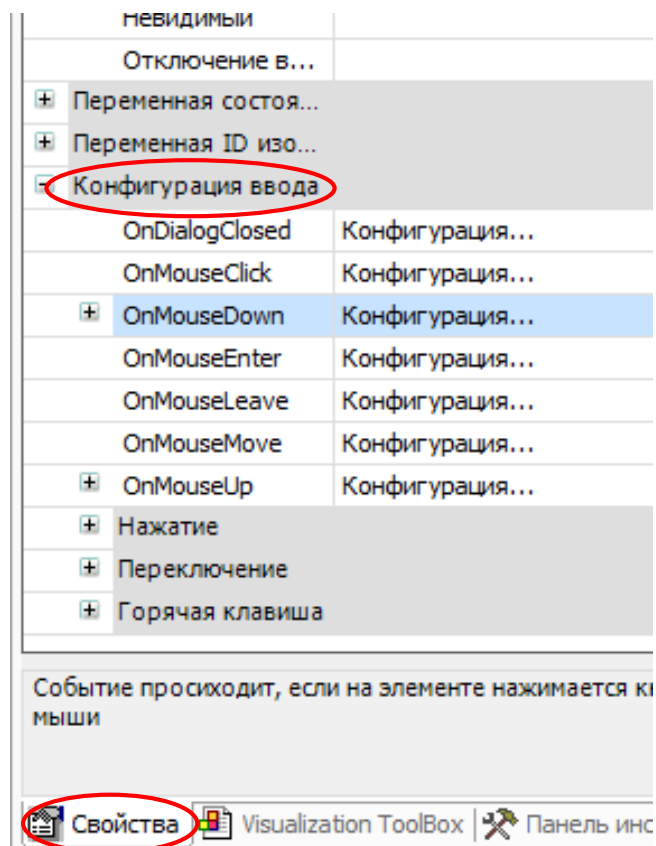


Рис. 14. Выбор свойства OnMouseDown

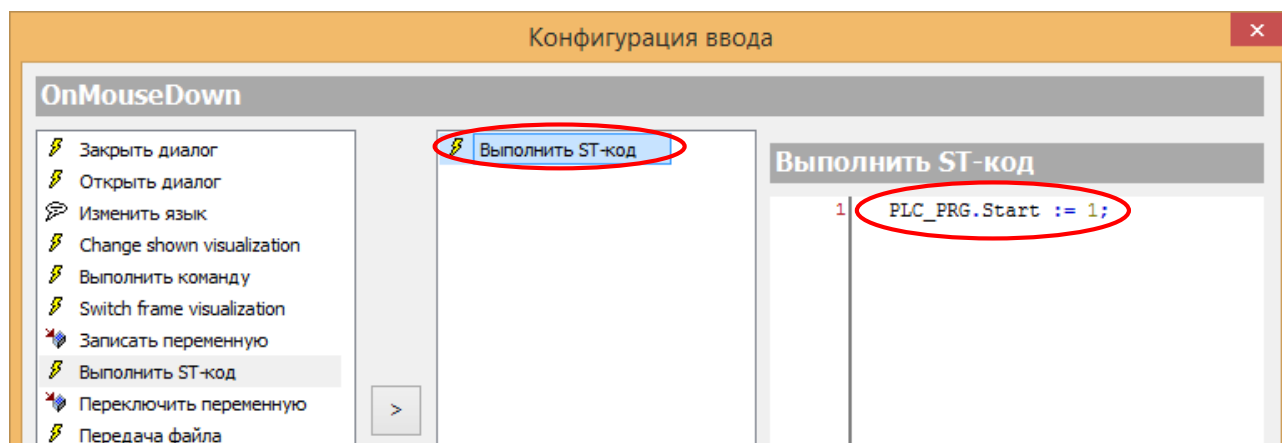


Рис. 15. Привязка кода на ST к кнопке

Аналогично для **OnMouseUp** (отпускание кнопки мыши) следует указать команду выключения этой же переменной:

PLC_PRG.Start := 0;

3. Необходимо выделить кнопку **СТОП** и проделать те же действия, за исключением того, что имя переменной будет **PLC_PRG.Stop**.

4. После этого нужно выделить **Индикатор** и в его свойствах указать переменную **PLC_PRG.Vent** (рис. 16).

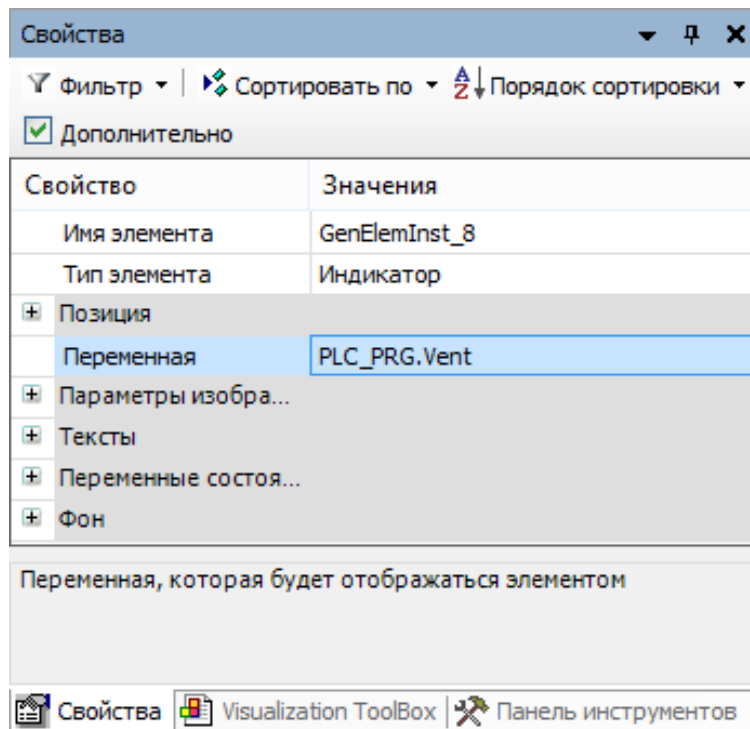


Рис. 16. Задание переменной в свойствах индикатора

5. Затем нужно выделить кнопку **Пожар** и для её свойства **OnClick** указать команду, изменяющую (при каждом нажатии) значение переменной на противоположное:

PLC_PRG.Fire := not PLC_PRG.Fire;

Для этой же кнопки следует задать отображение текущего значения переменной цветом (рис. 17)

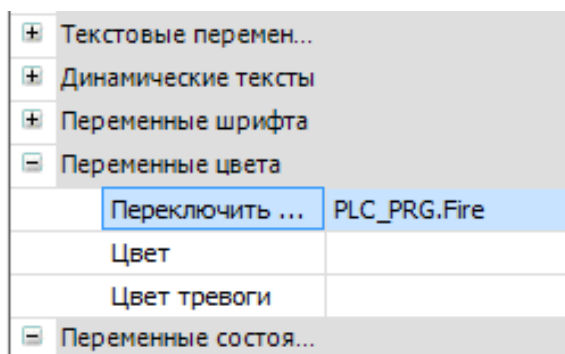


Рис. 17. Задание отображения текущего значения переменной цветом

Также для этой кнопки необходимо изменить «Цвет тревоги» на красный (рис. 18).

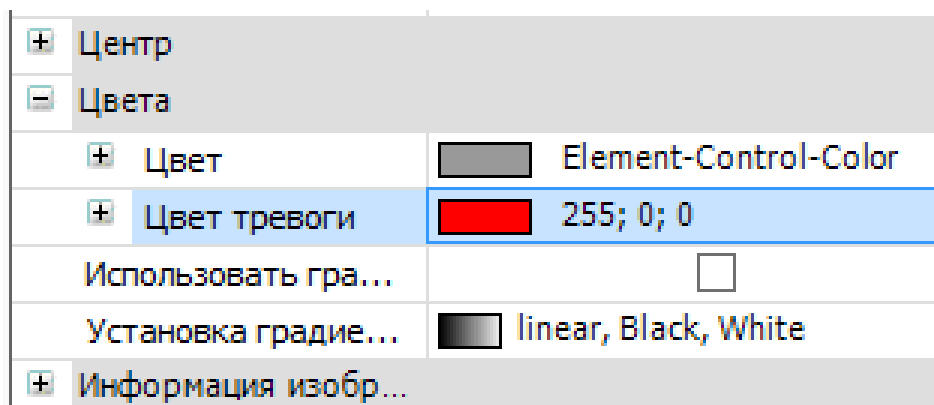


Рис. 18. Выбор цвета тревоги для кнопки

3.3. Дополнительные возможности при создании экранов

Достаточно часто экран оператора должен содержать технологическую схему управляемого процесса. Изобразить ее с помощью имеющихся в библиотеке элементов визуализации может быть сложно, долго или вообще невозможно.

В данной ситуации можно вставить на экран готовое изображение. Для этого необходимо выполнить следующие действия:

1. Добавить **Пул изображений** (рис. 19).
 2. В появившемся окне можно изменить имя **Пула изображений** (или оставить его по умолчанию) и нажать кнопку **Добавить**.
 3. В окне пула изображений в появившейся таблице дважды кликнуть левой кнопкой мыши в ячейку **Имя файла** (рис. 20).
 4. В появившемся списке найти необходимое изображение и задать настройки, как показано на рис. 21.
 5. Нажать кнопку **ОК**.
 6. После этого в таблице **Пула** появится название изображения в проекте (**ID**), название изображения на компьютере (**Имя файла**), само изображение и тип ссылки (рис. 22).
 7. Для добавления загруженного изображения на экран в качестве фона необходимо кликнуть правой кнопкой мыши по экрану визуализации и выбрать в появившемся списке пункт **Фон** (рис. 23).
 8. В появившемся окне поставить галочку **Изображение**, после чего добавить необходимое изображение из списка (рис. 24).
- После проделанных операций появится фон экрана оператора, поверх которого можно будет размещать различные элементы управления (рис. 25).

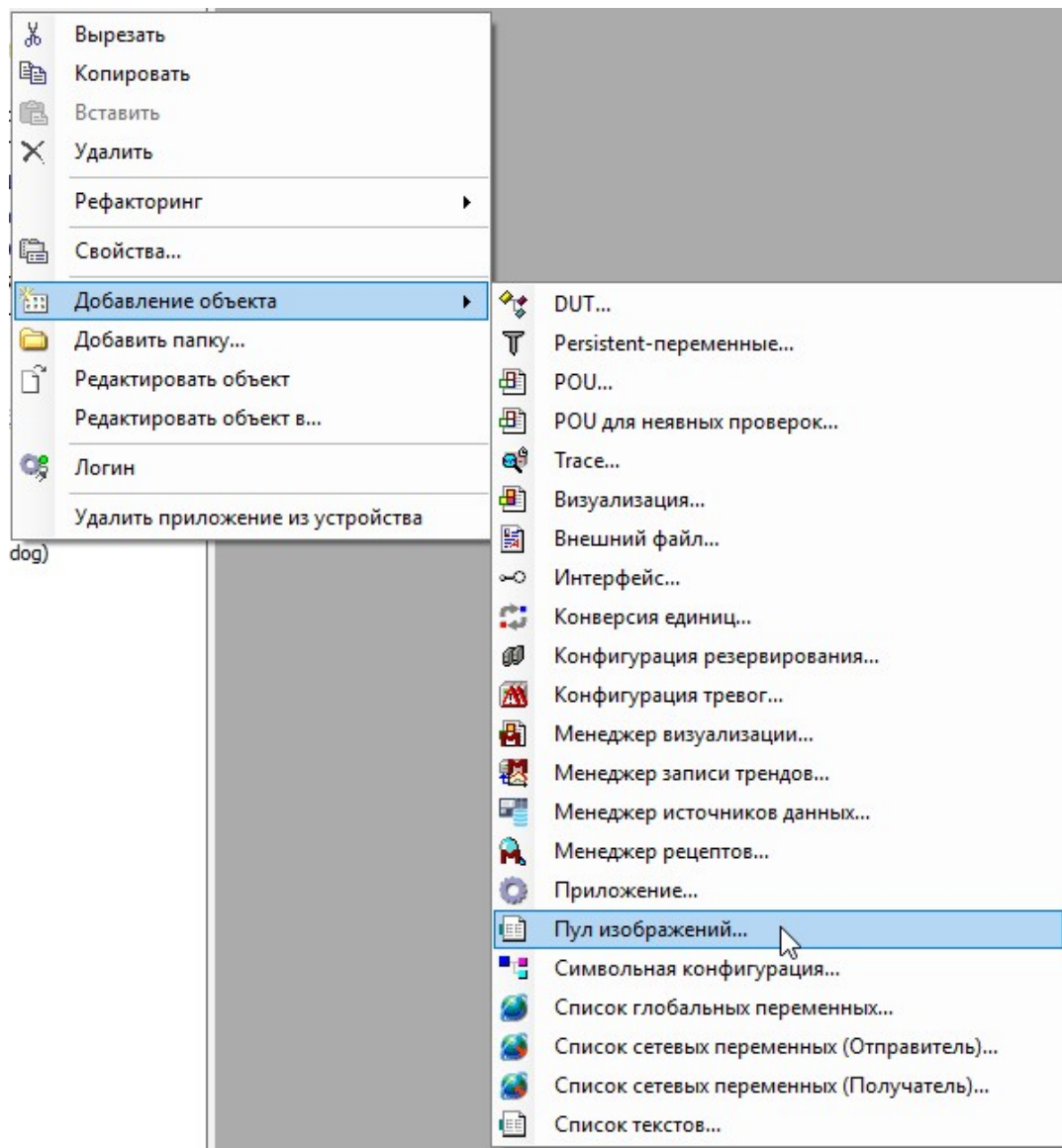


Рис. 19. Добавление Пула изображений

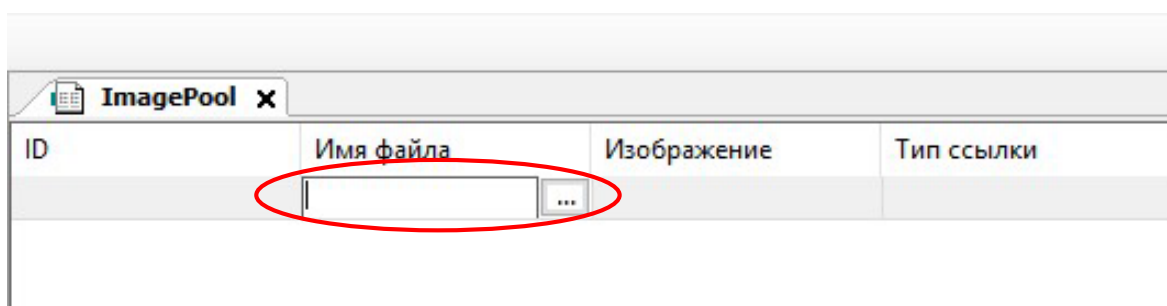


Рис. 20. Выбор ячейки Имя файла

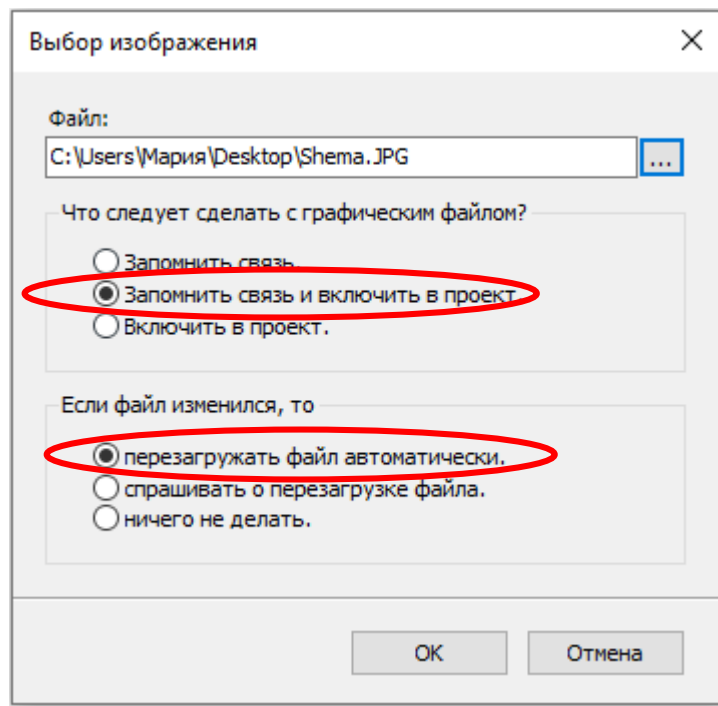


Рис. 21. Выбор изображения и его настроек

ID	Имя файла	Изображение	Тип ссылки
Shema	Shema.JPG		Embedded and link to file

Рис. 22. Пул изображений, после выбора изображения

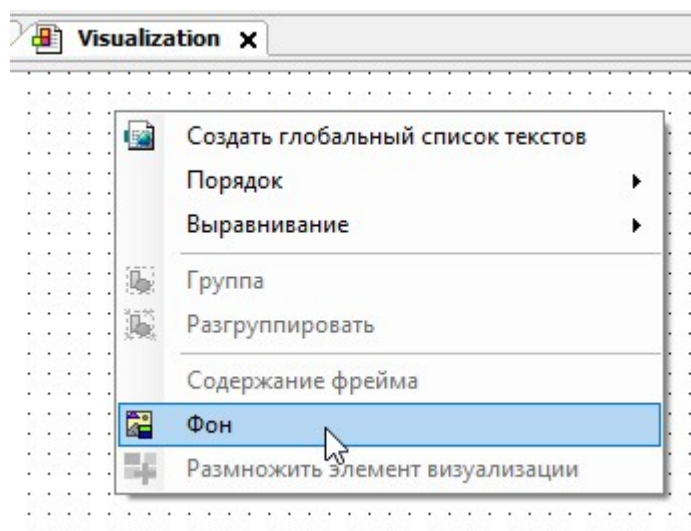


Рис. 23. Выбор пункта Фон

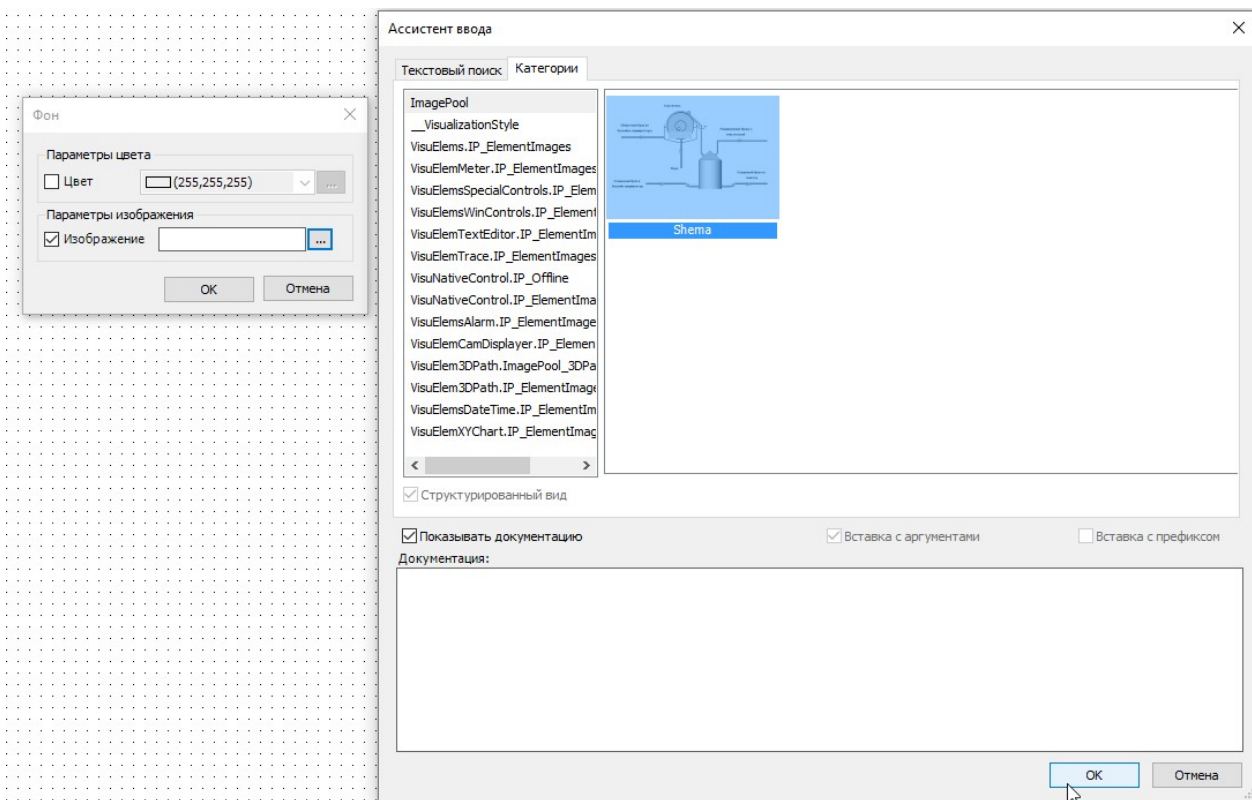


Рис. 24. Добавление изображения в качестве фона

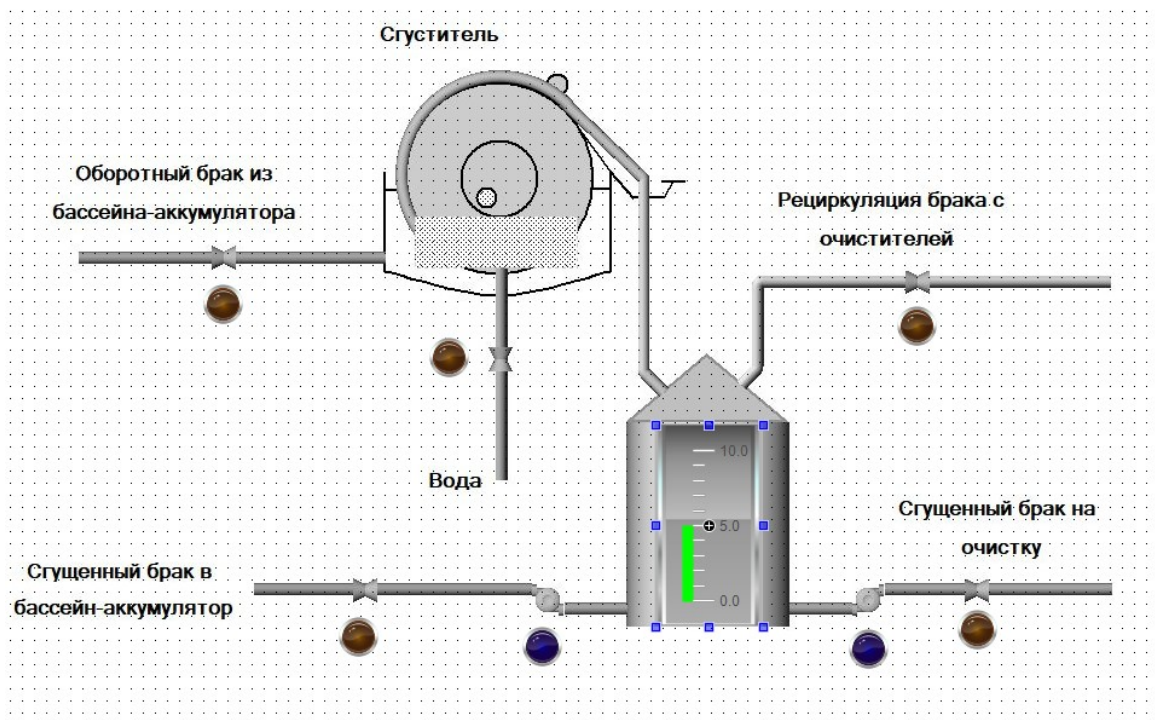


Рис. 25. Экран оператора с фоном

Добавлять графические файлы можно не только для создания фона экрана [4]. Их возможно использовать как статические или динамические изображения (с помощью элемента Изображение), как динамические (переключаемые по значению логической переменной) изображения с помощью элемента Переключатель изображений, как пиктограммы для Журнала тревог, как фоновые изображения для некоторых элементов (Отображение линейки, Кнопка).

!!! Для того, чтобы использовать изображение для всех перечисленных ситуаций, имя изображения должно быть написано латинскими буквами.

Контрольные вопросы

1. Как привязать к кнопке выполнение кода на языке ST?
2. Каким кодом на ST значение переменной меняется на противоположное при нажатии на кнопку?
3. Как изменить цвет тревоги у кнопки?
4. За счет чего связаны кнопки ПУСК/ СТОП с Индикатором?
5. Приведите примеры ситуаций, когда может быть необходимо использовать добавляемый в проект графический файл.

4. ЗАПУСК ЭМУЛЯЦИИ

Для запуска режима эмуляции необходимо проделать следующие действия:

1. Включить режим **Эмуляция** (рис. 26).

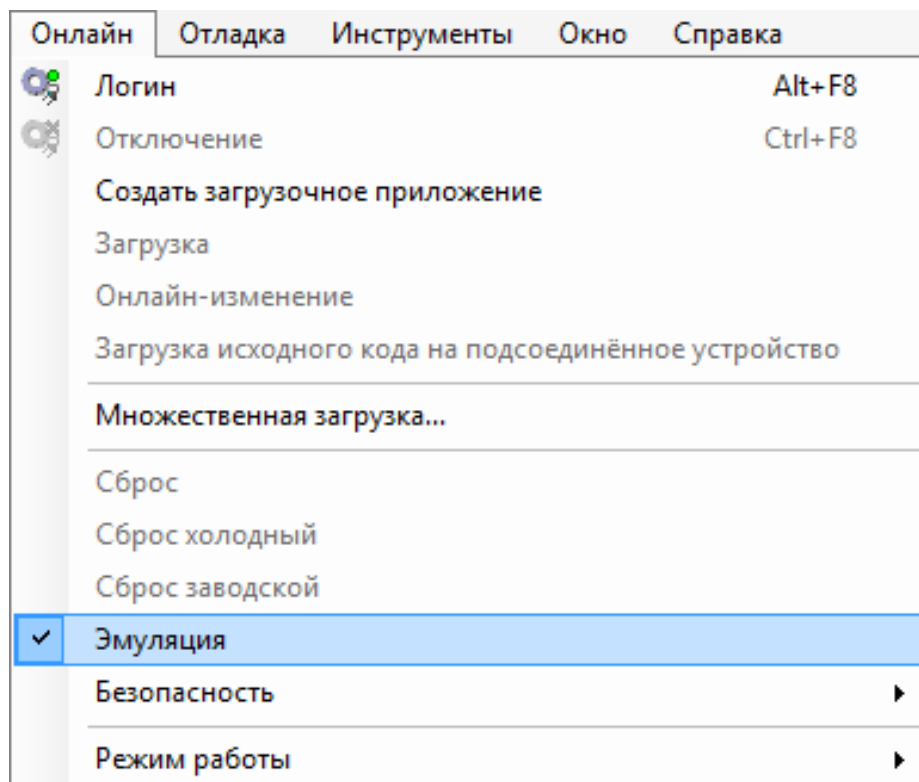


Рис. 26. Включение режима эмуляции

2. Выполнить команду **Логин** (Alt+F8).
3. Выполнить команду **Старт** (F5) (рис. 27)

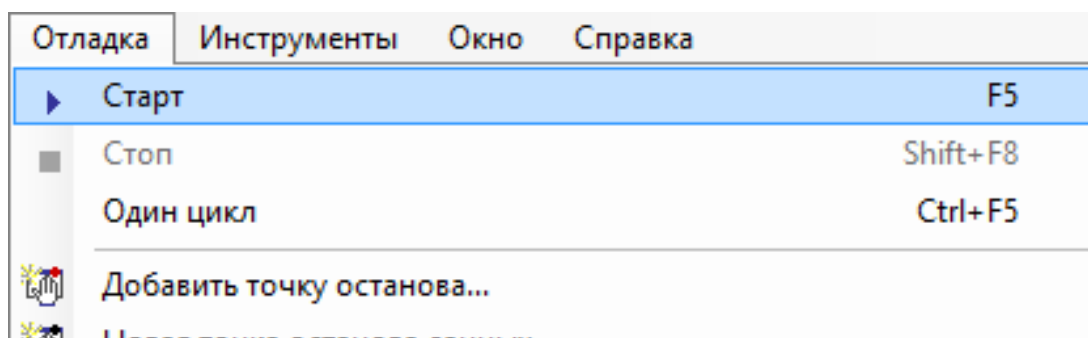


Рис. 27. Команда Старт

4. Открыть созданный ранее экран и проверить работу программы (рис. 28).

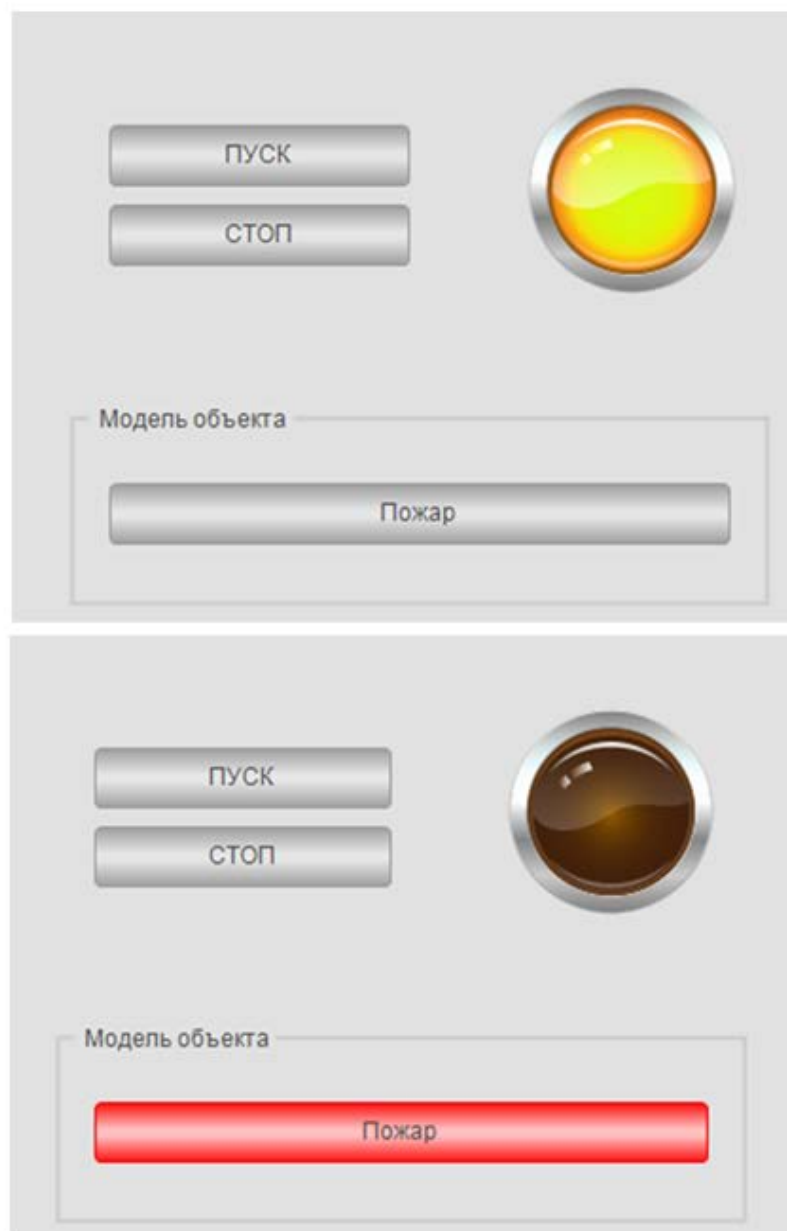


Рис. 28. Проверка работы программы

Контрольные вопросы

1. Как включить режим Эмуляция?
2. Какие «горячие клавиши» выполняют команду Логин?
3. Какую команду выполняют «горячие клавиши» F5?
4. Как изменить значение переменной в режиме Эмуляции без использования экранов?

5. СОЗДАНИЕ СОБСТВЕННОГО БЛОКА НА ST

Для создания собственного блока на языке ST необходимо добавить в проект новую программу (рис. 29).

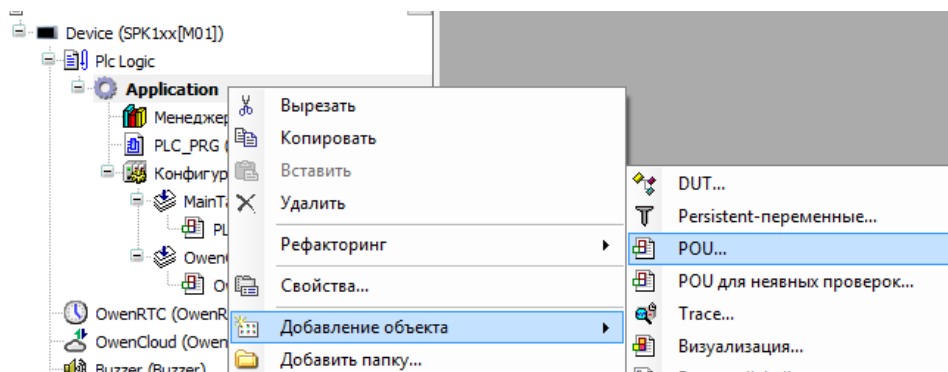


Рис. 29. Добавление в проект новой программы

После этого следует указать настройки, представленные на рис. 30.

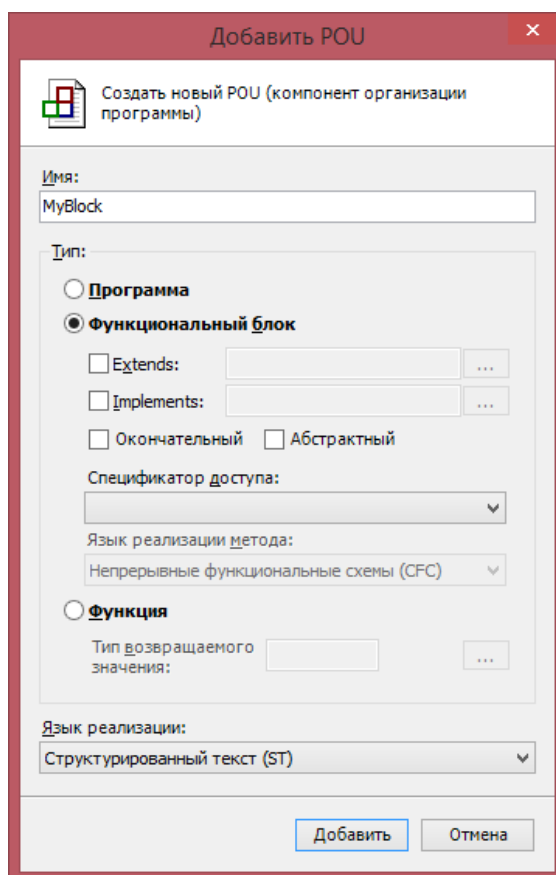


Рис. 30. Настройки для создания собственного блока

5.1. Постановка задачи

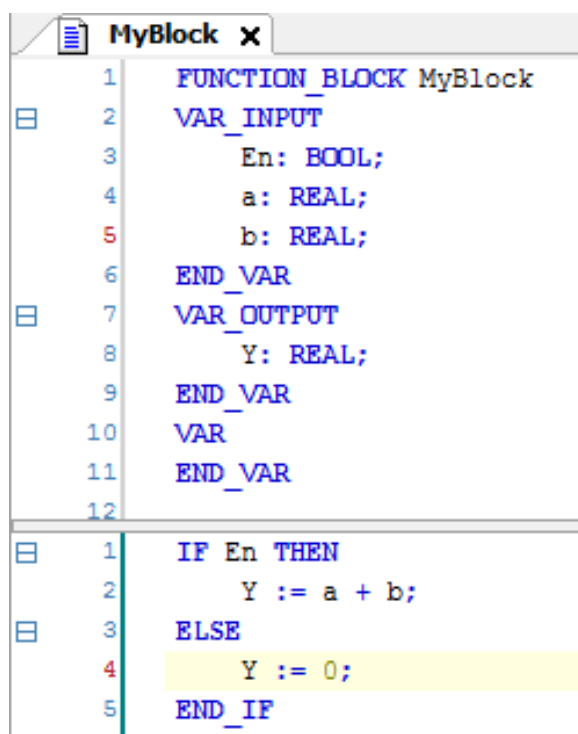
Необходимо написать собственный блок, реализующий следующий алгоритм:

- На выход **Y** выводится сумма входов **a** и **b**;
- Если на входе **En** (Enable, Разрешить) значение **False**, то на выходе **Y** всегда ноль (не зависимо от **a** и **b**).

!!! Переменные **a**, **b** и **Y** имеют тип **REAL**, а переменная **En** имеет тип **BOOL**.

5.2. Пример программы на ST

Программа на языке **ST**, реализующая описанное ранее задание представлена на рис. 31.



```
MyBlock x
1  FUNCTION_BLOCK MyBlock
2  VAR_INPUT
3      En: BOOL;
4      a: REAL;
5      b: REAL;
6  END_VAR
7  VAR_OUTPUT
8      Y: REAL;
9  END_VAR
10 VAR
11 END_VAR
12
1  IF En THEN
2      Y := a + b;
3  ELSE
4      Y := 0;
5  END_IF
```

Рис. 31. Программа для создаваемого блока

5.3. Редактор ST

Редактор **ST** похож на редактор **CFC**, в верхней части объявляются переменные, в нижней части пишется программа (но уже не графическая, а текстовая).

Сначала в нижней части пишется код программы. Для автоматического объявления переменных необходимо выполнить действие, показанное на рис. 32.

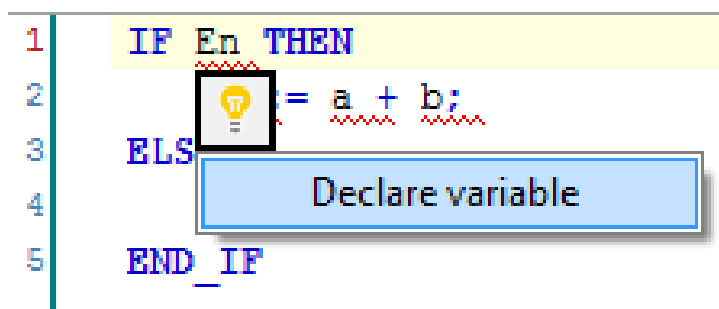


Рис. 32. Действие, выполняемое для автоматического объявления переменных

Стоит обратить внимание, что блоки, кроме переменных, имеют входы и выходы (**VAR_INPUT** и **VAR_OUTPUT**) (рис. 33).

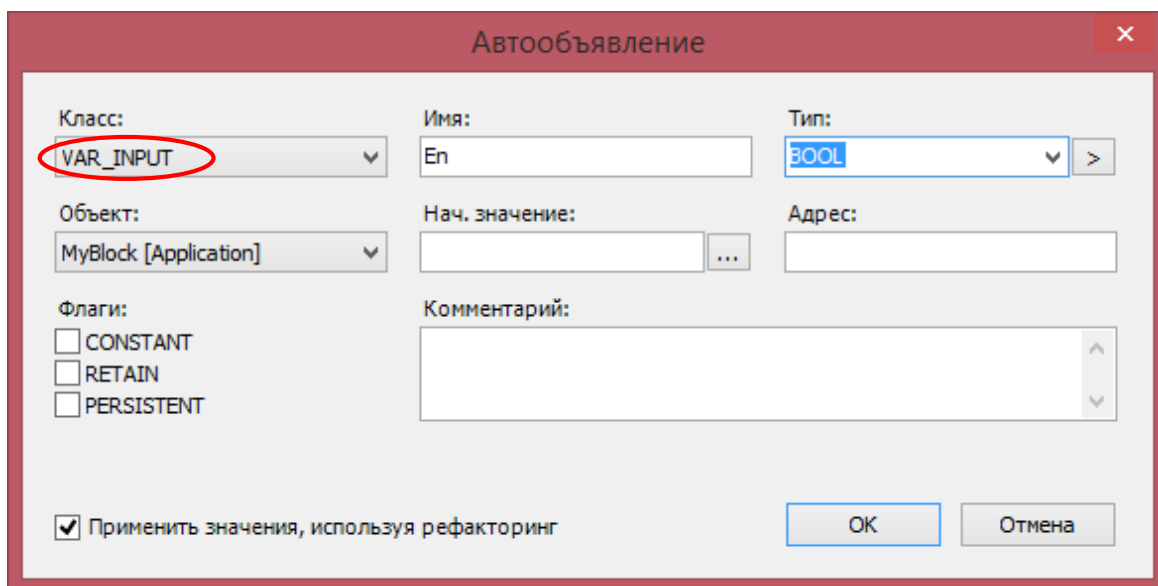


Рис. 33. Задание входов и выходов блоков

5.4. Использование созданного блока

Теперь можно использовать созданный блок в главной программе, например так, как показано на рис. 34.

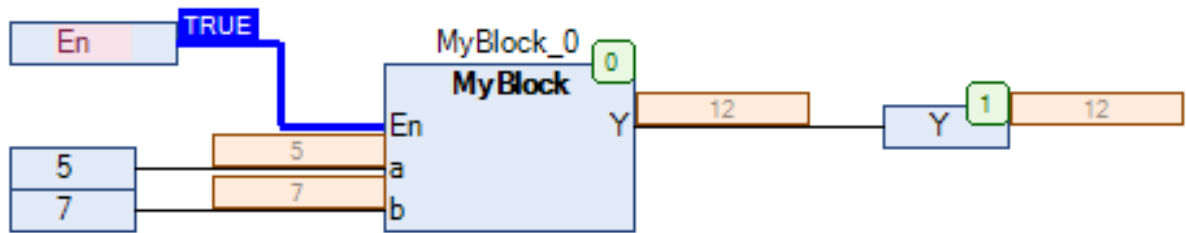


Рис. 34. Использование созданного блока

!!! Рисунок сделан при запущенной Эмуляции.

Контрольные вопросы

1. Как создать собственный блок в CoDeSys?
2. Почему переменные в программе на рис. 31 имеют такие типы данных?
3. Как включить автоматическое объявление переменных?
4. Почему при создании блока нужно указать класс переменной (вход или выход)?

6. СИСТЕМЫ РЕГУЛИРОВАНИЯ

6.1. ПИД-регулятор

Для того, чтобы обеспечить регулирование температуры, необходимо использовать блок ПИД-регулятора. Для этого необходимо выполнить следующие действия.

1. Создать новый проект (как описано в п. 1.2).
2. В **PLC_PRG** добавить новый элемент и задать ему имя **pid** (рис. 35).

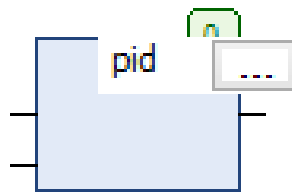


Рис. 35. Элемент pid

После нажатия клавиши Enter и создания переменной, блок принимает вид, показанный на рис. 36.

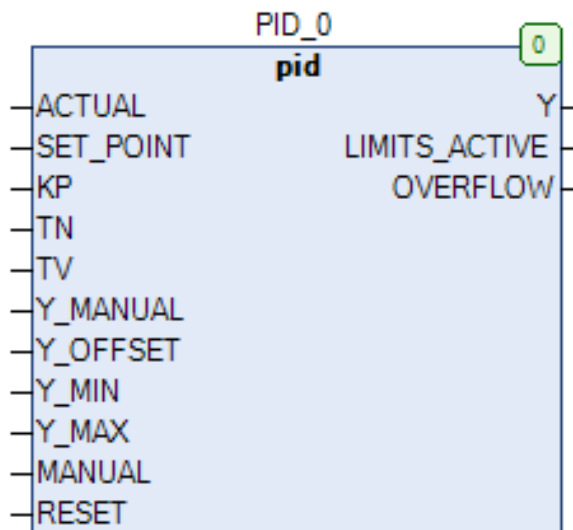


Рис. 36. Вид блока pid

Описание входов и выходов блока PID представлено в табл. 2.

Таблица 2 – Описание входов и выходов блока PID

	Имя	Тип	Описание
Вход	ACTUAL	REAL	Текущее значение, измеряемый технологический параметр, датчик
	SET-POINT	REAL	Требуемое значение, заданное значение
	KP	REAL	Пропорциональный коэффициент P
	TN	REAL	Интегральный коэффициент I (сек)
	TV	REAL	Дифференциальный коэффициент D (сек). <i>Если установлено значение 0, то работает как ПИ-регулятор</i>
	Y_MANUAL	REAL	Значение выходной переменной Y при ручном управлении (MANUAL=TRUE)
	Y_OFFSET	REAL	Смещение выходной переменной Y
	Y_MIN	REAL	Минимальное значение для выходной переменной Y
	Y_MAX	REAL	Максимальное значение для выходной переменной Y
	MANUAL	BOOL	Ручное управление, отключение регулятора
	RESET	BOOL	Сброс регулятора (накопленной интегральной составляющей)
Выход	Y	REAL	Выход ПИД-регулятора, управляющее воздействие
	LIMITS_ACTIVE	BOOL	Выход значения Y за заданные пределы (Y_MIN и Y_MAX)
	OVERFLOW	BOOL	Переполнение интегральной части

3. Запустить компиляцию, в результате которой будет получено две ошибки (рис. 37). Это связано с тем, что к проекту не подключена требуемая библиотека.

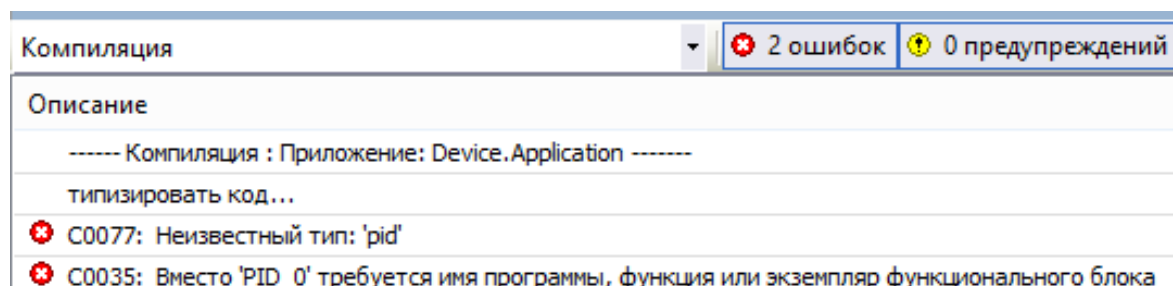


Рис. 37. Ошибки компиляции

4. Открыть менеджер библиотек (рис. 38).

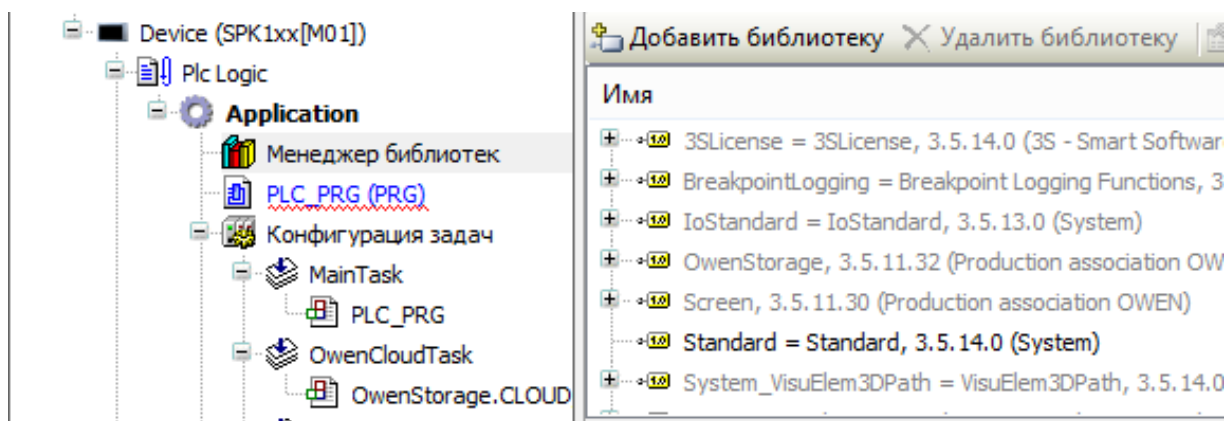


Рис. 38. Менеджер библиотек

!!! Активные библиотеки выделены черным, а неактивные – серым.

5. Нажать кнопку «Добавить библиотеку» и добавить библиотеку «Util» (рис. 39).

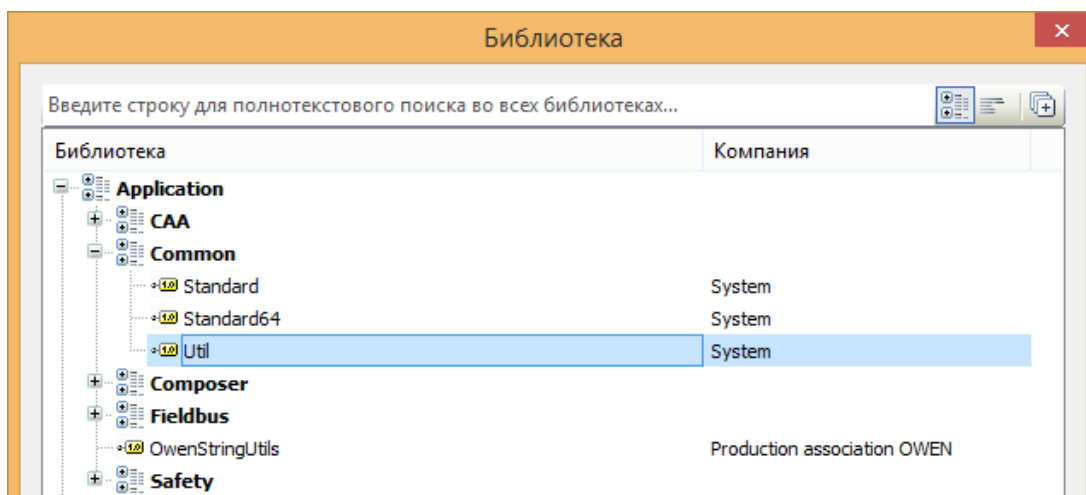


Рис. 39. Добавление библиотеки Util

6.2. Модель объекта

Математической моделью объекта регулирования температуры (а также концентрации и многих других) является апериодическое звено 1-го порядка (или больших порядков) с запаздыванием или без него. Наиболее подробная информация о динамических звеньях имеется в [5].

Простейшим вариантом будет следующий блок, реализованный на языке ST (рис. 40).

```
Obj x
1  FUNCTION_BLOCK Obj
2  VAR_INPUT
3      In: REAL;
4      K: REAL := 1;
5      T: REAL := 10;
6  END_VAR
7  VAR_OUTPUT
8      Out: REAL;
9  END_VAR
10 VAR
11 END_VAR
1  Out := K*(0.02/T)*In + (1-0.02/T)*Out;
```

Рис. 40. Модель объекта на языке ST

Данный блок в CFC будет выглядеть, как показано на рис. 41.

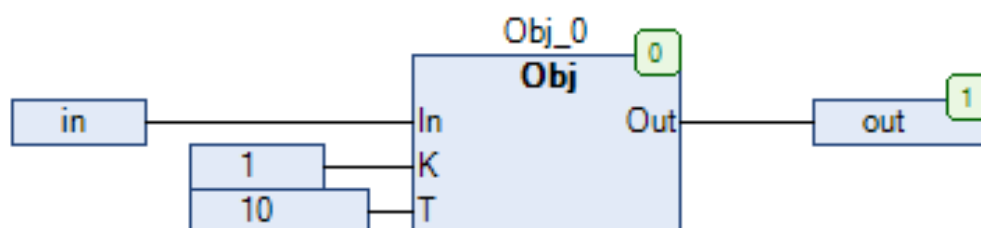


Рис. 41. Внешний вид созданного блока

При подаче на вход (In) блока единичного ступенчатого воздействия, будет получен график переходного процесса, представленный на рис. 42.

Стоит отметить, что взятая на рис. 40 константа 0,02 зависит от интервала, выбранного в MainTask (рис. 43).

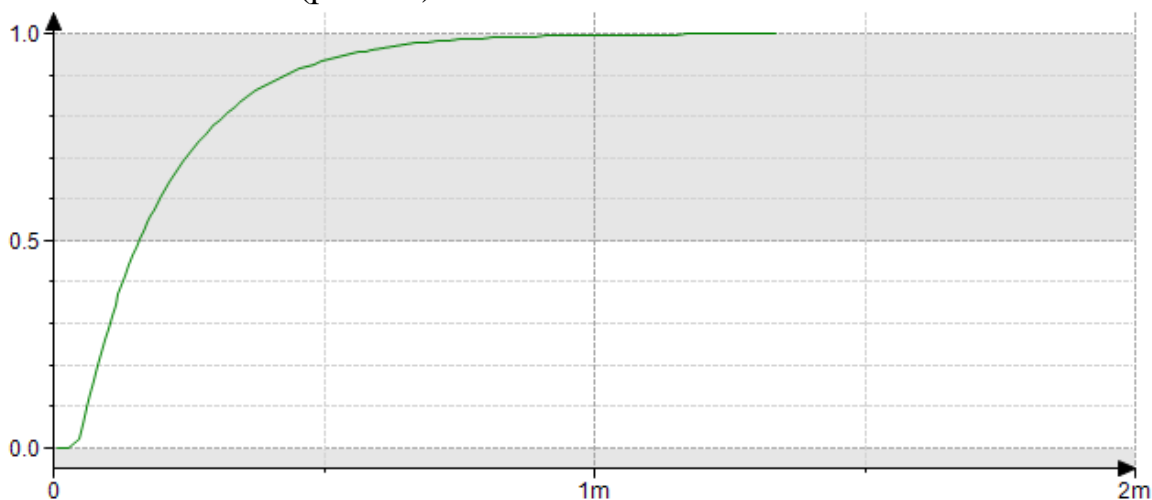


Рис. 42. График переходного процесса объекта

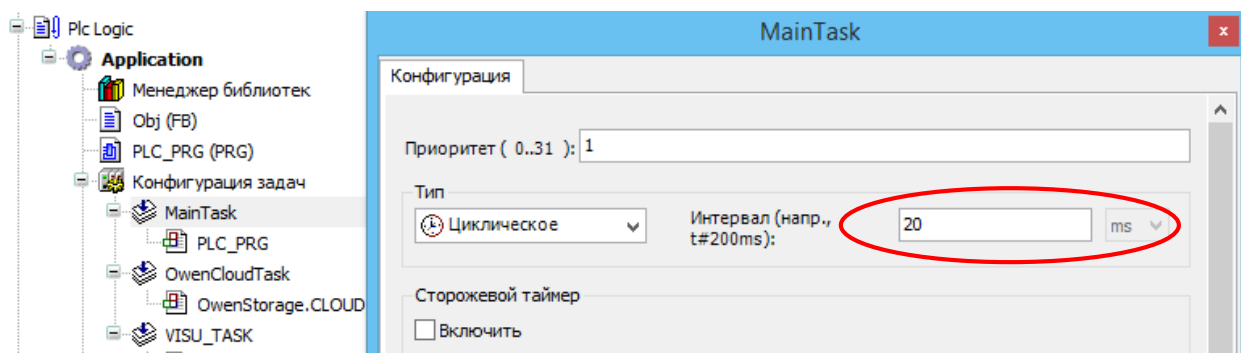


Рис. 43. Интервал в MainTask

6.3. Модель системы

На рис. 44 представлена модель системы регулирования температуры, состоящей из блоков pid и Obj, описанных в пунктах 6.1 и 6.2.

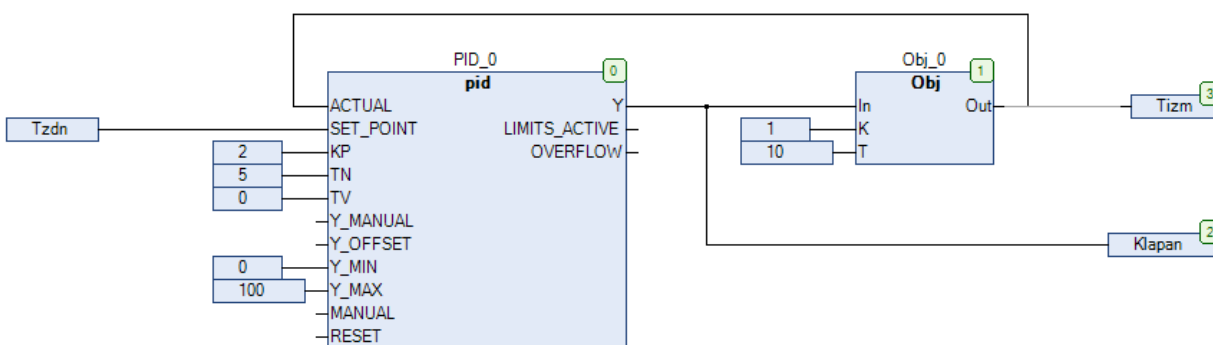


Рис. 44. Модель системы регулирования температуры

6.4. Ввод и отображение числовых значений

Для входного значения необходимо добавить элемент «Управление вращением» (рис. 45).

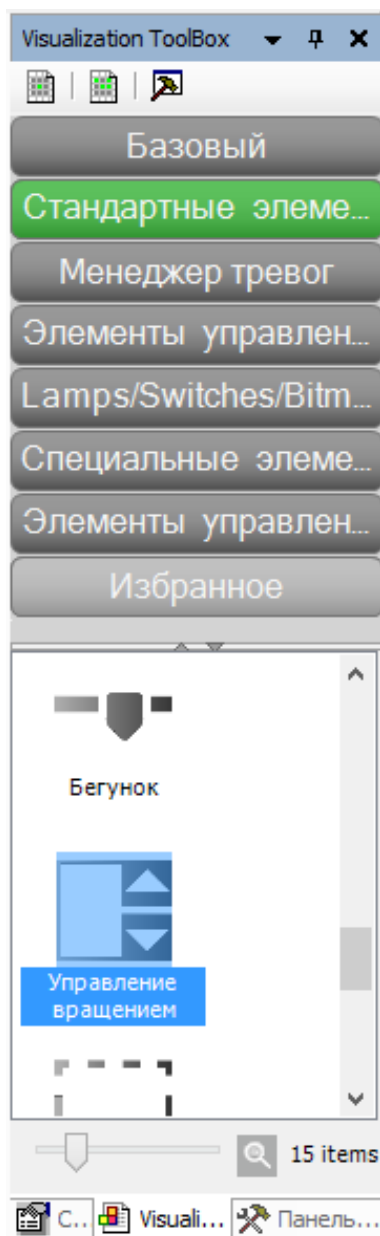


Рис. 45. Элемент «Управление вращением»

В его свойствах следует указать требуемую входную переменную (рис. 46).

Шаг изменения вводимого с помощью элемента «Управление вращением» значения не обязан быть равным единице. Для его изменения необходимо отредактировать параметр «интервал», задав его, например, 0,5.

Свойства	
Фильтр	Сортировать по
Порядок сортировки	<input checked="" type="checkbox"/> Дополнительно
Свойство	Значения
Имя элемента	GenElemInst_3
Тип элемента	Управление вращением
Позиция	
X	120
Y	20
Ширина	150
Высота	30
Переменная	PLC_PRG.Tzdn
Числовой формат	
Интервал	1

Рис. 46. Задание входной переменной для элемента «Управление вращением»

Для выходного значения температуры нужно добавить элемент «Прямоугольник» (рис. 47).

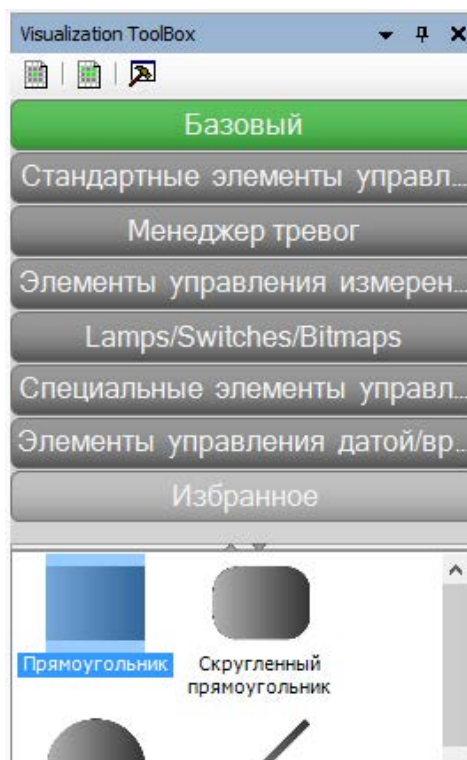


Рис. 47. Элемент «Прямоугольник»

В его свойствах необходимо указать требуемую переменную и формат отображаемого значения, как показано на рис. 48.

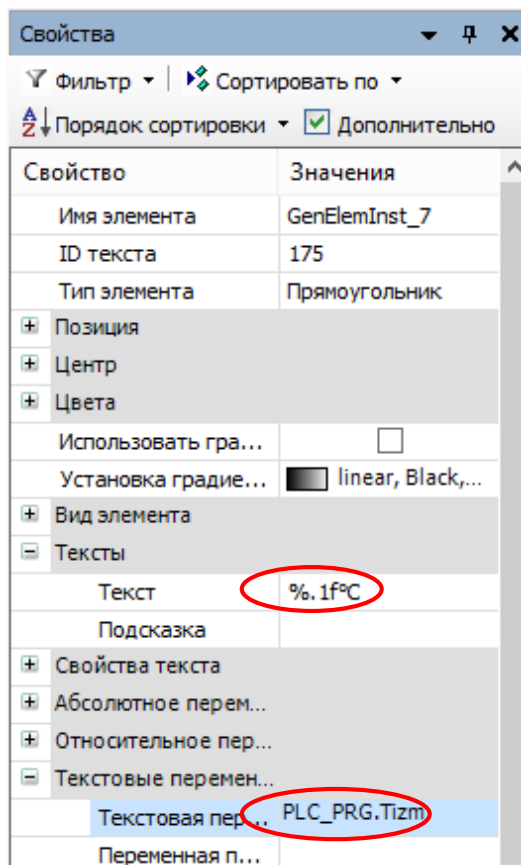


Рис. 48. Задание переменной и формата отображаемого значения для элемента «Прямоугольник»

!!! Описанный способ отображения значения может быть применен не только к Прямоугольнику, но и к некоторым другим элементам, например, к Кнопке.

Для выходного значения процента открытия клапана необходимо проделать аналогичные действия, указав в качестве единиц измерения проценты. В итоге будут получены три значения, представленные на рис. 49.

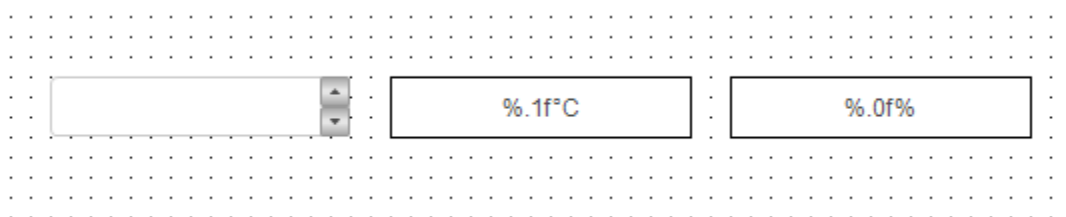


Рис. 49. Отображение значений

6.5. Отображение графиков

Для добавления графика используется элемент «Трассировка» (рис. 50).

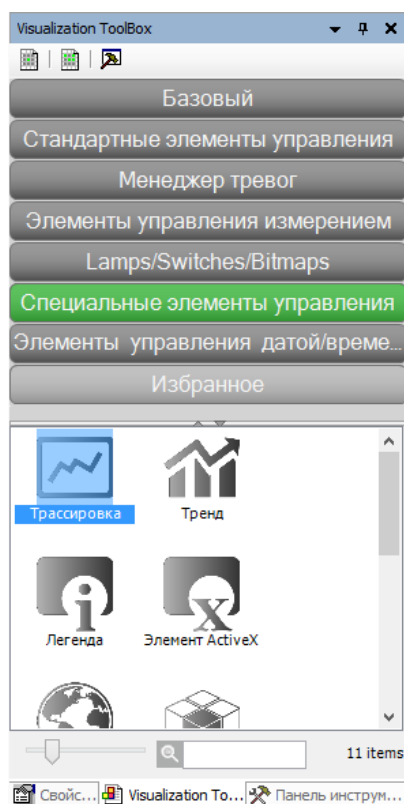


Рис. 50. Элемент «Трассировка»

В свойствах графика нужно выбрать «Трассировка» (рис. 51).

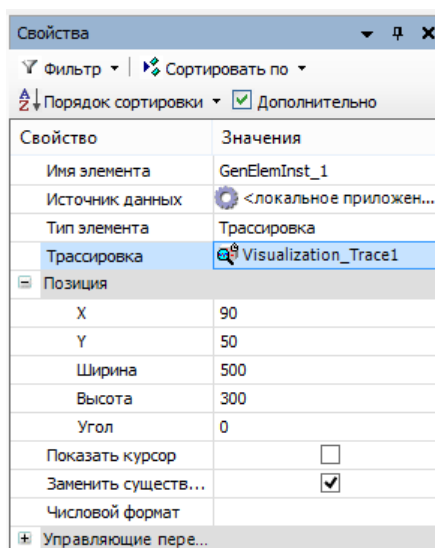


Рис. 51. Свойство графика «Трассировка»

В качестве «Задачи» нужно выбрать MainTask, а затем добавить переменную (рис. 52).

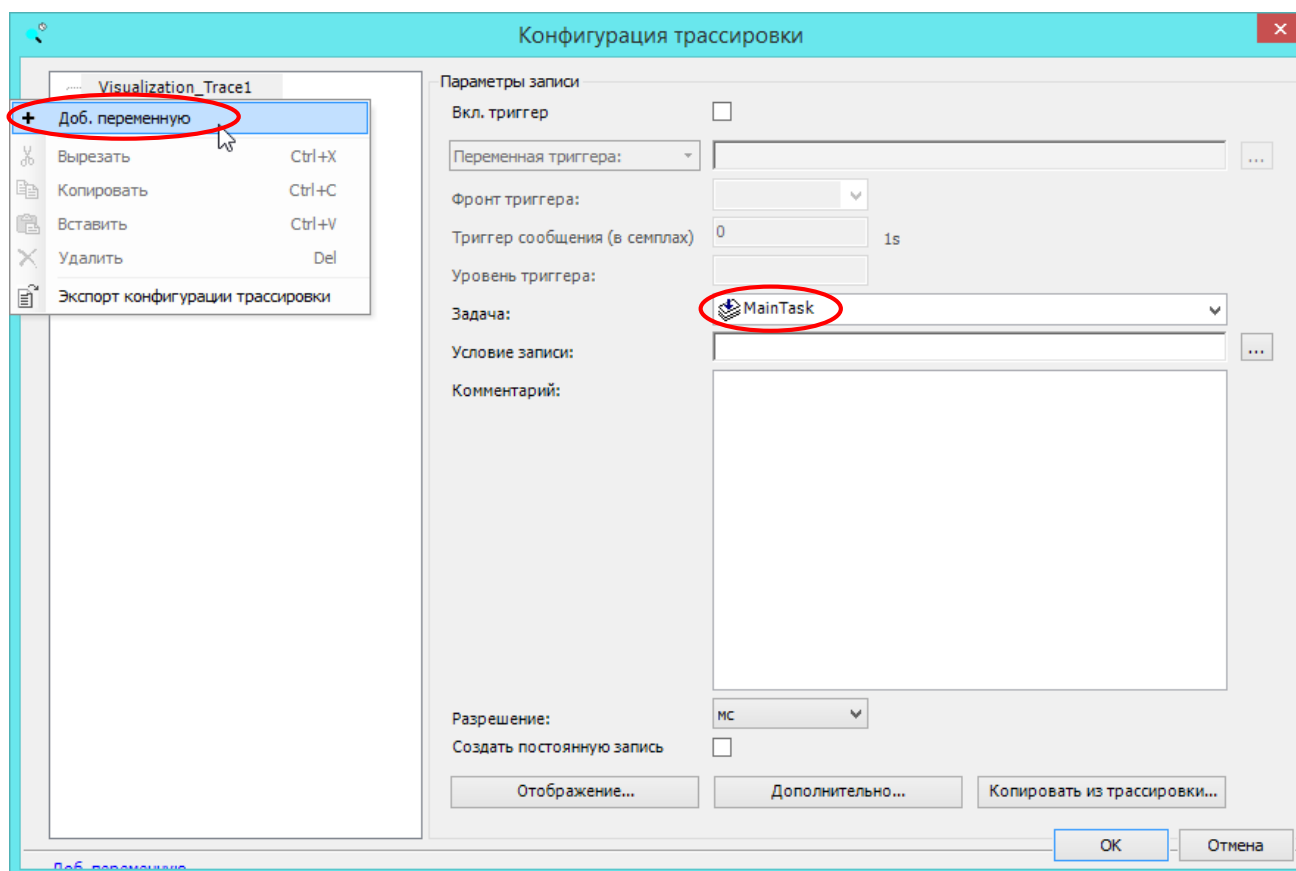


Рис. 52. Выбор задачи

Всего нужно добавить две переменные, представленные на рис. 53. Одна из них — это заданное значение, другая — измеренное.

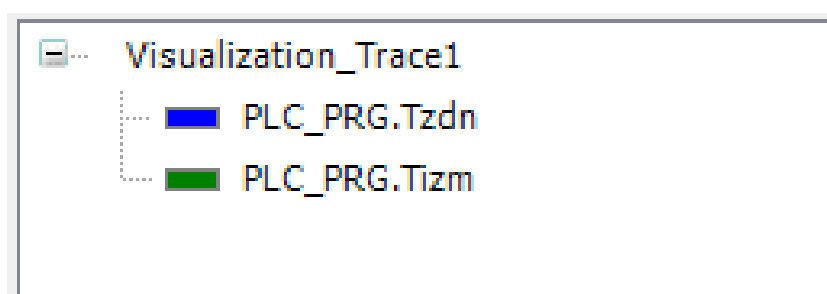


Рис. 53. Добавляемые переменные

Также следует провести настройку Отображения (рис. 54) и Дополнительных параметров (рис. 55).

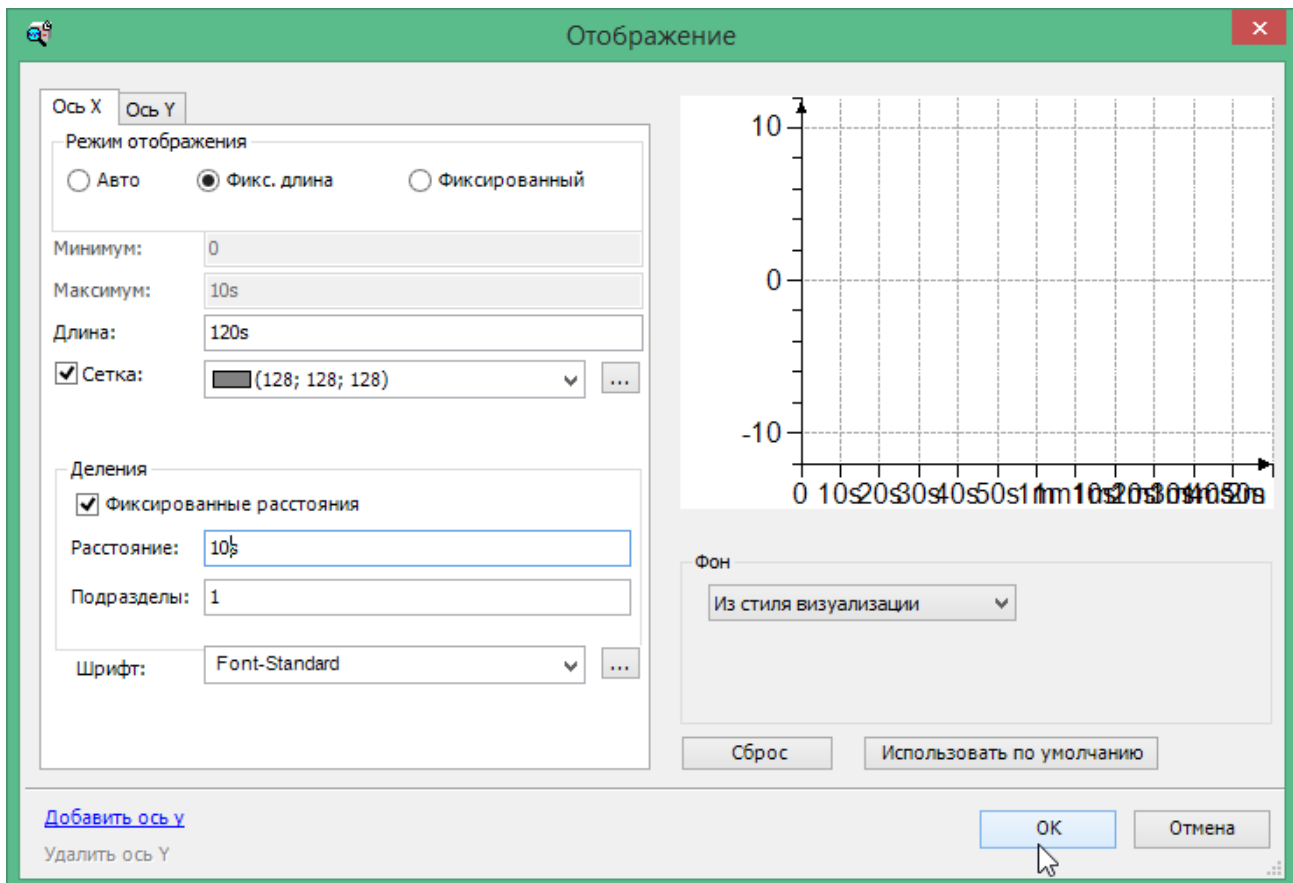


Рис. 54. Настройка отображения

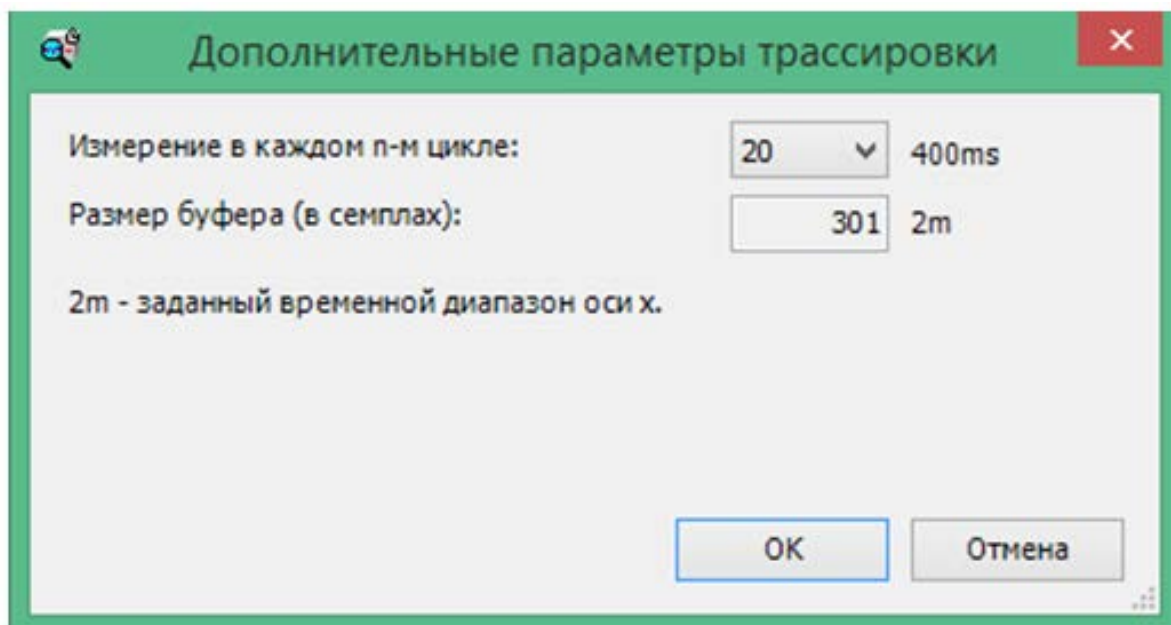


Рис. 55. Настройка дополнительных параметров

6.6. Итоговый экран

На рис. 56 изображен экран, полученный в итоге.

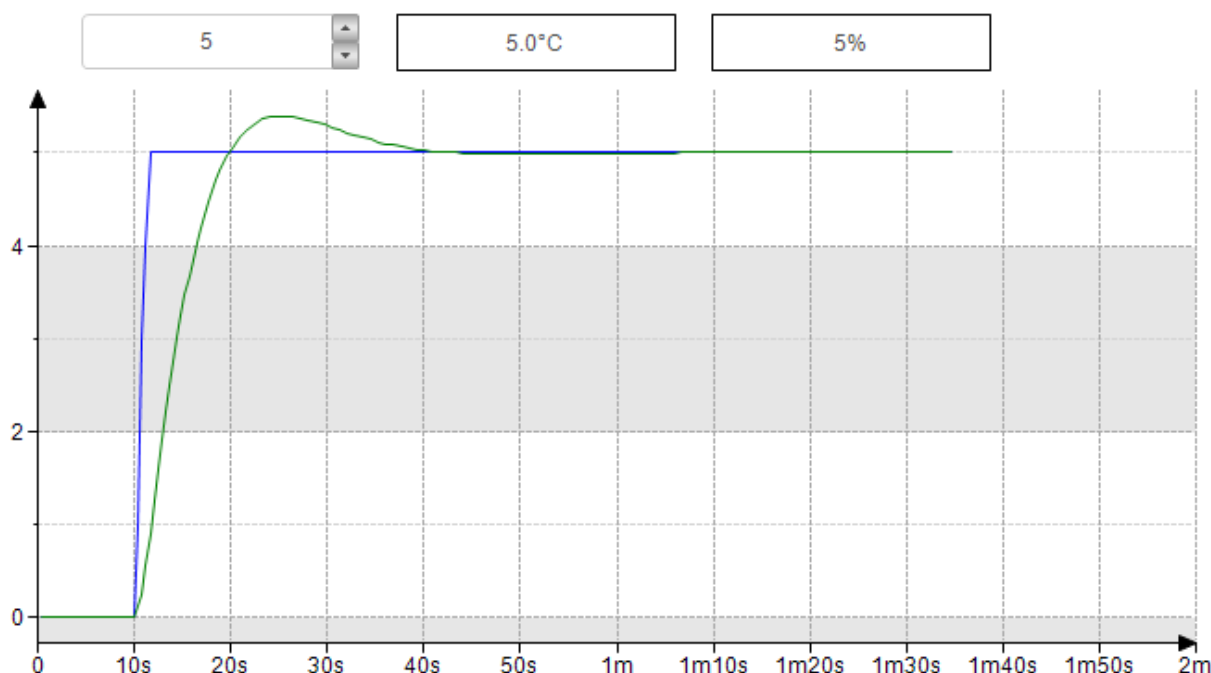


Рис. 56. Итоговый экран

!!! Стоит отметить, что полученный график не оптимален, т.к. настройка коэффициентов регулятора не проводилась.

6.7. Транспортное запаздывание

Кроме систем без запаздывания (например, как на рис. 56), также используются системы с запаздыванием (чистое транспортное запаздывание). Для реализации блока запаздывания необходимо организовать массив, хранящий все значения на заданном интервале запаздывания.

Программа, реализующая блок транспортного запаздывания (Delay), представлена на рис. 57.

Пример использования блока в главной программе на языке CFC показан на рис. 58. Данный пример реализует задержку на 10 секунд, что проиллюстрировано на графике на рис. 59.


```

Delay x
1  FUNCTION_BLOCK Delay
2  VAR_INPUT
3      In, T: REAL;
4  END_VAR
5  VAR_OUTPUT
6      Out: REAL;
7  END_VAR
8  VAR
9      n, i: DINT;
10     A: Array[0..5000] of real;
11 END_VAR
12
1  n := TRUNC(t*50);
2
3  Out := A[n];
4  FOR i := 0 TO n-1 DO
5      A[n-i] := A[n-i-1];
6  END_FOR
7  A[0] := In;

```

Рис. 57. Блок транспортного запаздывания

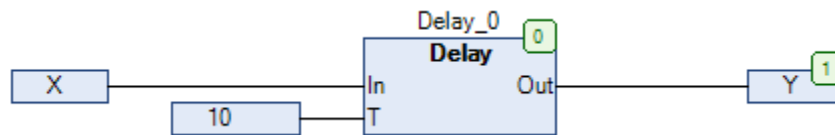


Рис. 58. Использование блок транспортного запаздывания

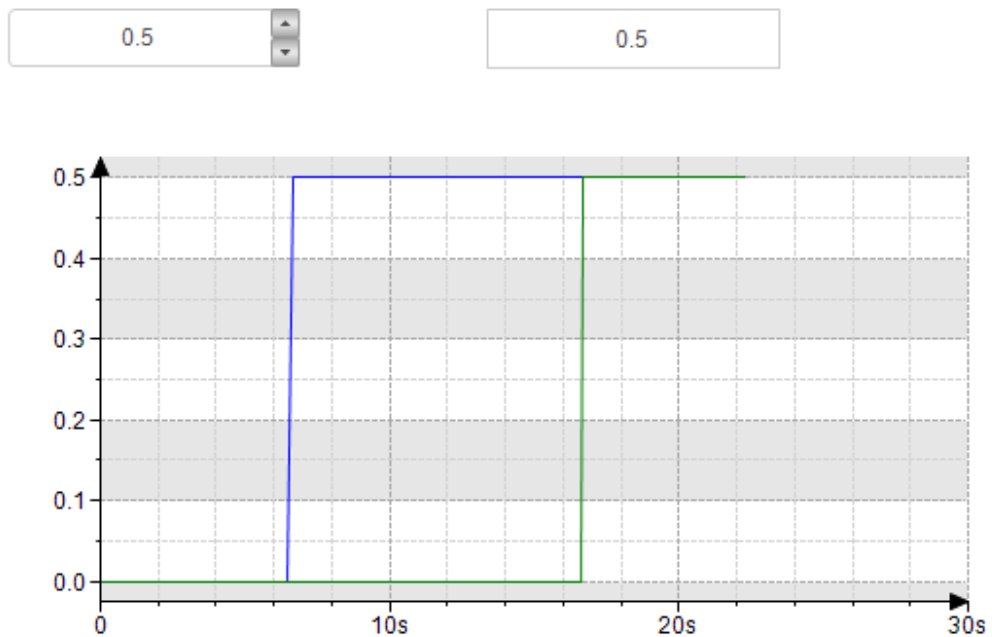


Рис. 59. Задержка на 10 секунд

Контрольные вопросы

1. Какие входы и выходы блока PID используются в разработанной программе и для чего?
2. Что Вы знаете об апериодическом звене 1-го порядка?
3. Почему константа 0,02 в программе на рис. 40 соответствует значению интервала 20 на рис. 43?
4. Почему с выхода блока Obj сигнал подается на вход PID?
5. Как задается формат отображаемого значения и переменной?
6. Расскажите, как и в каком порядке подбираются коэффициенты для PID-регулятора.
7. Как, имея блок апериодического звена 1-го порядка, получить апериодическое звено 2-го порядка?

7. ЛАБОРАТОРНЫЕ РАБОТЫ

7.1. Содержание отчета

Лабораторная работа должна содержать следующие обязательные части:

1. Титульный лист.
2. Содержание (оглавление).
3. Код программы на языках CFC (или др. языках по согласованию с преподавателем).
4. Экраны оператора.
5. Заключение/выводы.
6. Список литературы.

7.2. Лабораторная работа 1

Повторить программу на языке CFC и Экран, описанные в примерах (главы 2, 3), а также проверить работу Эмуляции (глава 4).

7.3. Лабораторная работа 2

Разработать систему регулирования температуры (программу и экран), описанную в главе 6.

7.4. Лабораторная работа 3

Разработать проект для поддержания заданного уровня жидкости в резервуаре. Резервуар имеет три трубы с клапанами. По первой трубе с клапаном m1 жидкость подается в резервуар. Открытие или закрытие m1 задается пользователем с экрана оператора. Если m1 закрыт, то регулирование уровня жидкости не производится.

По второй трубе с клапаном m_2 жидкость вытекает из резервуара. Положение m_2 изменяется автоматически в соответствии с программой.

Третья труба с клапаном m_3 является дренажной. Клапан на ней открывается только при условии, что оба клапана m_1 и m_2 закрыты.

Регулирование уровня жидкости в резервуаре должно обеспечиваться с помощью регулятора. Передаточная функция объекта регулирования определяется в соответствии с вариантом (табл. 3).

Таблица 3 — Варианты

№ п/п	Уровень, л	Время достижения уровня (без регулирования), с
1	3	10
2	4	15
3	5	20
4	6	25
5	7	30
6	3	15
7	4	20
8	5	25
9	6	30
10	7	10
11	3	20
12	4	25
13	5	30
14	6	10
15	7	15
16	3	25
17	4	30
18	5	10
19	6	15
20	7	20

!!! Разрабатываемая система не является реальной.

На экране разрабатываемого проекта обязательно должны иметься:

1. Название лабораторной работы.
2. ФИО и группа студента.
3. Мнемосхема.
4. Элемент трассировки.
5. Элементы для задания и отображения необходимых величин.

Группы элементов необходимо подписывать. Шрифт должен быть 14 кегля. Необходимо помнить о том, что экран оператора должен быть наглядным и не перегруженным информацией. Одним из способов достижения этого требования является размещение отображаемой информации на нескольких экранах (рис. 60).



Лабораторная работа №3
Регулирование уровня
Выполнил: ст. гр. 111 Иванов И.И.



Главный экран



График

Экран График

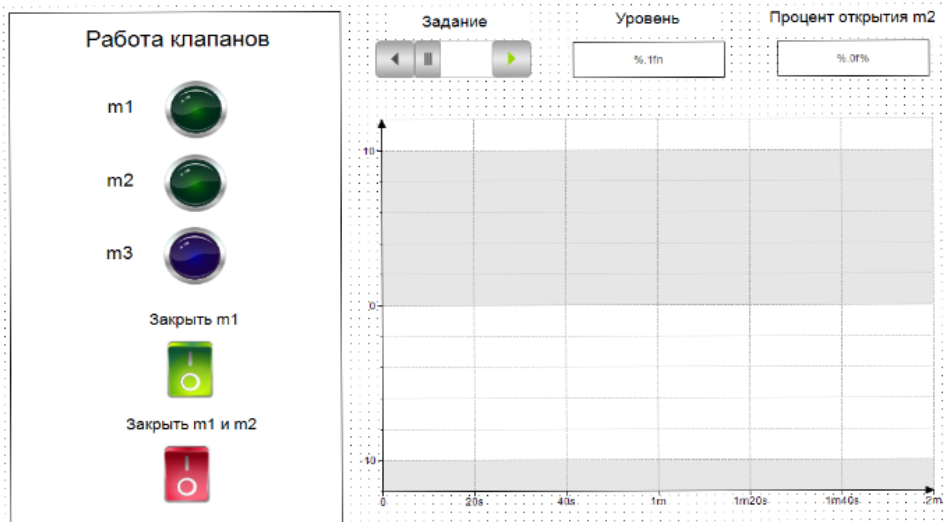


Рис. 60. Пример экранов оператора лабораторной работы 3

8. КУРСОВАЯ РАБОТА (ПРОЕКТ)

8.1. Содержание курсовой работы (проекта)

Курсовая работа (проект) должна содержать следующие обязательные части:

1. Титульный лист.
2. Задание.
3. Содержание (оглавление).
4. Код программы на языках CFC и ST (или др. языках по согласованию с преподавателем).
5. Экраны для тестирования программы (не менее двух экранов).
6. Описание получения модели объекта с её численными характеристиками.
7. Настройку системы регулирования и оценку качества её работы.
8. Заключение/выводы.
9. Список литературы.

Страницы работы обязательно должны быть пронумерованы. В оглавлении должны быть указаны номера страниц. Программа (на любом языке) должна содержать комментарии. Все входные и выходные переменные должны иметь понятные имена, а также подробные комментарии о их функции.

8.2. Пример задания на курсовую работу (проект)

Курсовая работа (проект) выполняется по индивидуальному заданию, выданному преподавателем (по согласованию с преподавателем, студент может использовать для выполнения работы материалы, собранные при прохождении практики или полученные по месту работы). Индивидуальное задание содержит технологическую схему (рис. 61) и график переходного процесса (разгонную кривую) для одного из контуров технологического процесса (рис. 62).

Технологическая схема необходима прежде всего для разработки экрана оператора (необходимо перерисовать данную схему в CoDeSys на Главном экране проекта). Разгонная кривая необходима для получения модели объекта (определить тип звена и его коэффициенты). График, как правило, содержит две кривые, входную (в виде ступеньки) и выходную. Выходная может содержать шумы, их необходимо отбросить.

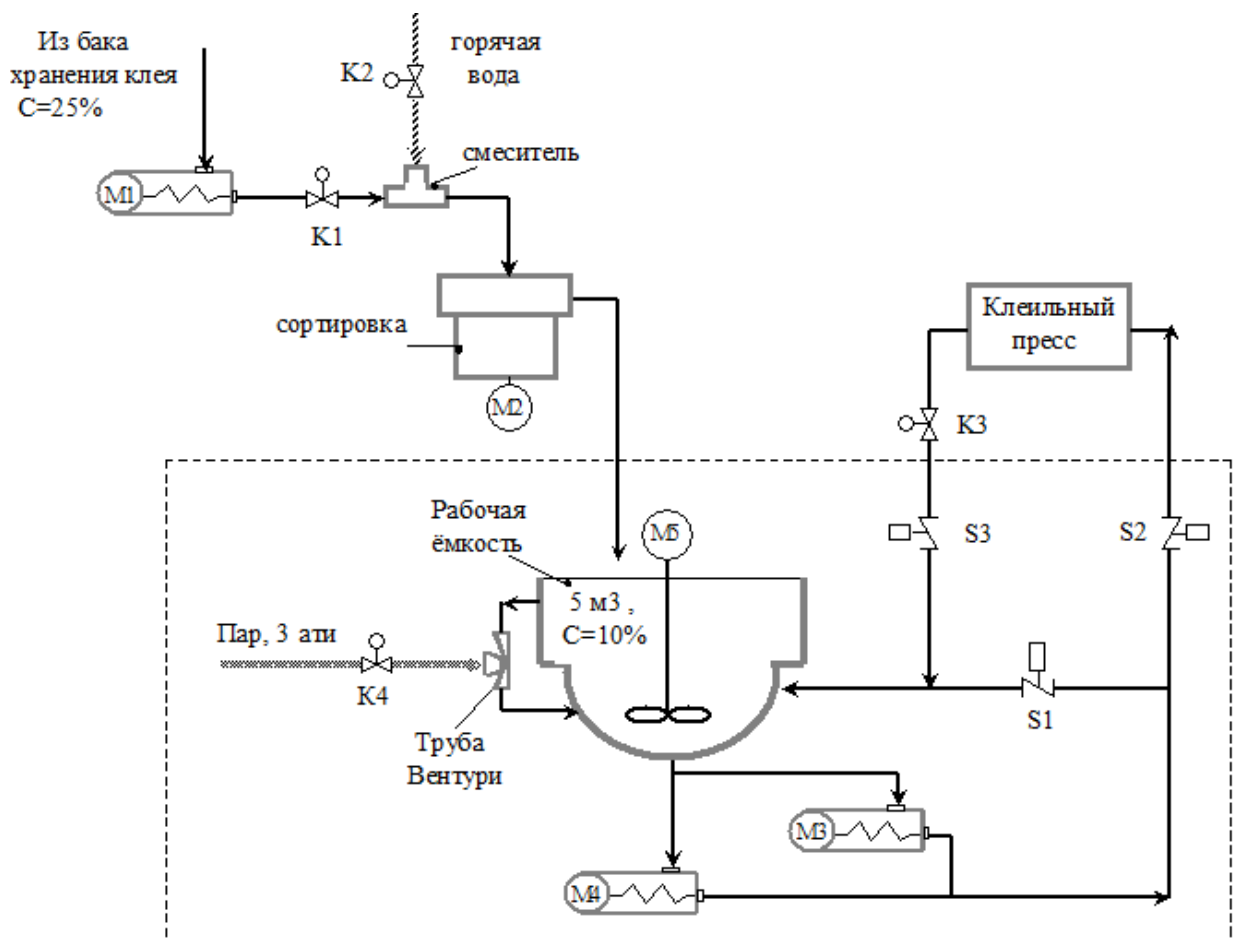


Рис. 61. Пример технологической схемы

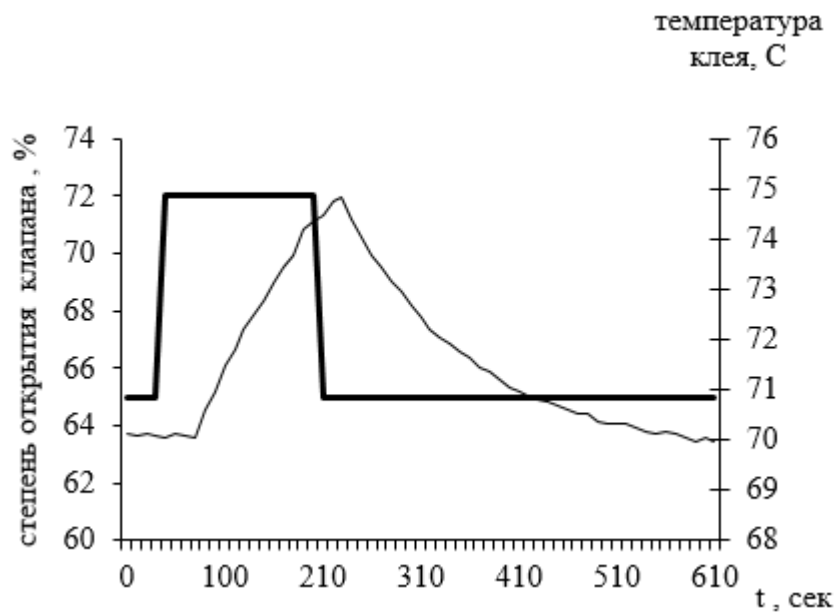


Рис. 62. Пример разгонной кривой

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ОВЕН: СПК110 контроллер с сенсорным экраном 10.2 для локальных систем [Электронный ресурс]. – URL: <https://owen.ru/product/srk110> (дата обращения 04.01.2022).
2. Установка программного обеспечения и загрузка проекта в прибор / Видео из YouTube, 12:06, загружено «ОВЕН. Производственное Объединение», ноябрь 09, 2017, <https://www.youtube.com/watch?v=gnO555SF5kA>.
3. Создание простейшего алгоритма и визуализации / Видео из YouTube, 13:55, загружено «ОВЕН. Производственное Объединение», ноябрь 09, 2017, <https://www.youtube.com/watch?v=YBaWIXs-UxM>.
4. ОВЕН. CODESYS 3.5 FAQ – Руководство пользователя версия 2.4 [Электронный ресурс]. – URL: https://ftp.owen.ru/CoDeSys3/11_Documentation/03_3.5.11.5/CDSv3.5_Faq_v.2.4.pdf (дата обращения 10.01.2022).
5. Бесекерский, В. А., Попов, Е. П. Теория систем автоматического управления [Текст] / В. А. Бесекерский, Е. П. Попов. – Изд. 4-е, перераб. и доп. – СПб.: Профессия, 2003. – 752 с.

Учебное издание

**Новиков Александр Игоревич
Воропанова Мария Андреевна**

Моделирование систем управления

Практикум

Редактор и корректор Е. О. Тарновская
Техн. редактор Е. О. Тарновская

Темплан 2021 г., поз. 5261

Подписано к печати 17.05.2022.

Формат 60x84/16.

Бумага тип № 1.

Печать офсетная.

Печ. л. 2,9.

Уч.-изд. л. 2,9.

Тираж 20 экз.

Изд. № 5261.

Цена «С».

Заказ №

Ризограф Высшей школы технологии и энергетики СПбГУПТД,
198095, Санкт-Петербург, ул. Ивана Черных, 4.