

**А. И. Новиков
М. А. Воропанова**

**АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ
ПРОЦЕССОВ И ПРОИЗВОДСТВ**

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**

**А. И. Новиков
М. А. Воропанова**

АВТОМАТИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ И ПРОИЗВОДСТВ

Практикум

Утверждено Редакционно-издательским советом ВШТЭ СПбГУПТД

Санкт-Петербург
2022

УДК 681.3(06)
ББК 32.965я7
Н731

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.

Н731 Автоматизация технологических процессов и производств: Практикум /
А. И. Новиков, М. А. Воропанова. – СПб.: ВШТЭ СПбГУПТД, 2022. –
71 с.

Практикум соответствует программам и учебным планам дисциплин «Автоматизация технологических процессов и производств», «SCADA-системы», «Моделирование систем управления», «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» 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 очной и заочной форм обучения. Отдельные разделы могут быть полезны магистрам, 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	5
1. НАЧАЛО РАБОТЫ	6
1.1. Описание контроллера	6
1.2. Установка программного обеспечения	7
1.3. Создание проекта	8
1.4. Состав проекта	9
Контрольные вопросы	9
2. ПОДКЛЮЧЕНИЕ МОДУЛЕЙ К КОНТРОЛЛЕРУ	11
2.1. Описание модулей	11
2.2. Добавление модулей к контроллеру	13
2.3. Настройка модулей	15
Контрольные вопросы	16
3. РАБОТА С ПРОГРАММОЙ НА CFC	17
3.1. Постановка задачи	18
3.2. Пример программы на CFC	19
3.3. Расположение элементов в программе	19
Контрольные вопросы	20
4. ОСНОВНЫЕ БЛОКИ CFC	21
4.1. Список ключевых слов	22
4.2. Логические блоки	23
4.3. Триггеры	24
4.4. Задержки включения и выключения	26
4.5. Блок выбора Sel	27
4.6. ПИД-регулятор	27
4.7. Система регулирования	30
4.8. Генераторы Gen	31
4.9. Получение справки по блокам	34
Контрольные вопросы	35
5. РАЗРАБОТКА ЭКРАНОВ	37
5.1. Редактор экранов	37
5.2. Связывание экрана с программой	37
5.3. Ввод и отображение числовых значений	40
5.4. Отображение графиков	43
5.5. Всплывающие окна для ввода значений	46
5.6. Дополнительные возможности при создании экрана	49
Контрольные вопросы	52
6. ЗАПУСК ЭМУЛЯЦИИ	53
Контрольные вопросы	54
7. СОЗДАНИЕ СОБСТВЕННЫХ БЛОКОВ	55
7.1. Постановка задачи	56
7.2. Пример программы на ST	56
7.3. Редактор ST	57

7.4. Использование созданного блока	58
7.5. Создание блока Аперидического звена 1-го порядка	58
7.6. Создание блока Аперидического звена 2-го порядка	60
7.7. Создание блока Интегрирующего звена	61
7.8. Создание блока Транспортного запаздывания	62
Контрольные вопросы	62
8. ЛАБОРАТОРНЫЕ РАБОТЫ	64
8.1. Содержание отчета	64
8.2. Лабораторная работа 1	64
8.3. Лабораторная работа 2	66
8.4. Лабораторная работа 3	66
9. КУРСОВАЯ РАБОТА (ПРОЕКТ)	69
9.1. Содержание курсовой работы (проекта)	69
9.2. Пример задания на курсовую работу (проект)	69
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	71

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. Ряд неординарных решений 3S привел к тому, что CoDeSys стал штатным инструментом программирования ПЛК ведущих европейских изготовителей: ABB, Beckhoff, Beck IPC, Berger Lahr, Bosch Rexroth, ifm, Keb, Kontron, Lenze, Moeller, WAGO, Fastwel и др. Внедрение стандарта послужило фундаментом для создания единой школы подготовки специалистов. 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].

1. CODESYS 3.5.14.30.exe
2. OwenTargets-3.5.14.30-09.package
3. CODESYS Repository Archive V3.5 SP4.msi
4. Mx110Drivers_v3.5.11.4.package

Рис. 2. Файлы, необходимые для выполнения работы

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

После установки в меню Инструменты / «**Менеджер пакетов**» должно отобразиться 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).

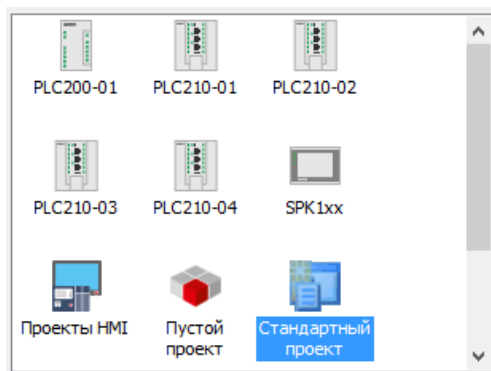


Рис. 4. Создание стандартного проекта

3. Выбрать контроллер **СПК110** и язык **CFC** (рис. 5).
Язык CFC – это модификация языка FBD (т. е. функциональных блоков, как например в Simulink MatLab).

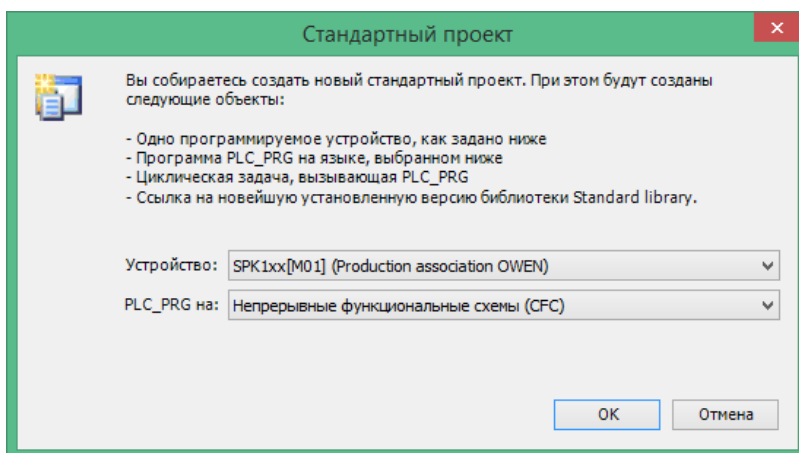


Рис. 5. Выбор контроллера и языка программирования

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

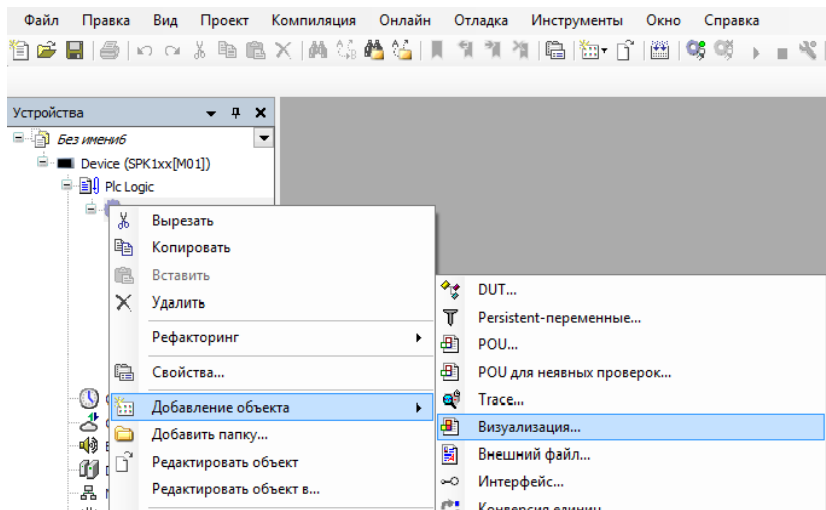


Рис. 6. Добавление экрана в проект

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?

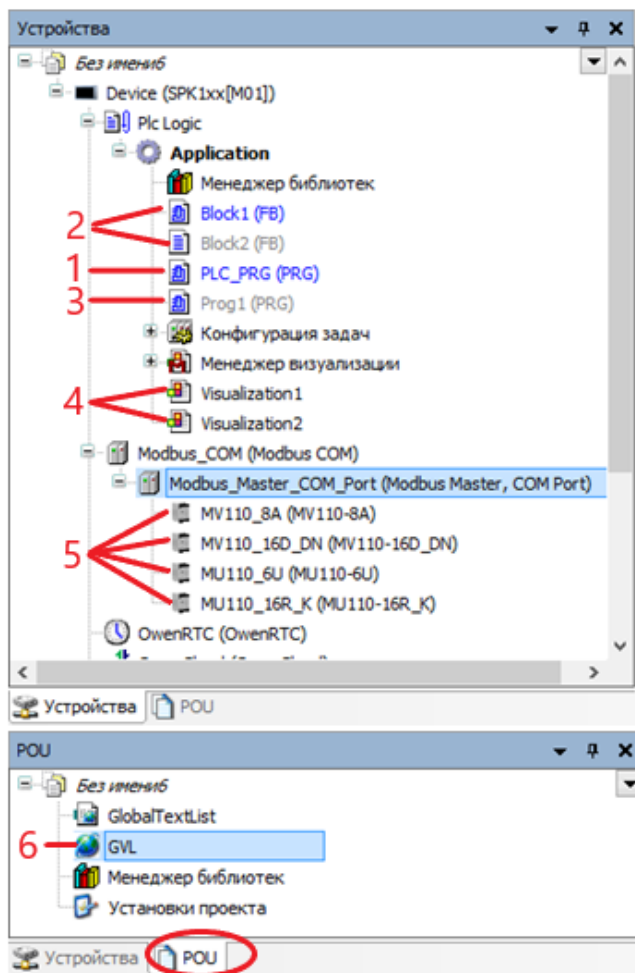


Рис. 7. Состав проекта: 1 – главная программа; 2, 3 – функциональные блоки и подпрограммы; 4 – экраны; 5 – модули ввода / вывода; 6 – список глобальных переменных

2. ПОДКЛЮЧЕНИЕ МОДУЛЕЙ К КОНТРОЛЛЕРУ

2.1. Описание модулей

Контроллер **СПК110** не содержит собственных входов и выходов. Необходимо использовать внешние модули, подключаемые по **ModBus**. Мы будем использовать следующие 4 типа модулей (AI, DI, AO, DO):

МВ110-224.8А, МВ110-224.16Д, МУ110-224.6У, МУ110-224.16Р.

Модули аналогового ввода МВ110-224.8А (рис. 8) предназначены для измерения аналоговых сигналов встроенными аналоговыми входами, преобразования измеренных величин в значение физической величины и последующей передачи этого значения по сети RS-485 [3].



Рис. 8. Модуль МВ110-224.8А

Данные модули имеют 8 аналоговых входов. Они поддерживают сигналы:

- унифицированные: 0...5 мА, 0(4)...20 мА, ±50 мВ, 0...1 В;
- термосопротивления: 50М, Cu50, 50П, Pt50, Ni100, 100М, Cu100, 100П, Pt100, Ni500, 500М, Cu500, 500П, Pt500, Ni1000, 1000М, Cu1000, 1000П, Pt1000;
- термопары: L, J, N, K, S, R, B, T, A-1, A-2, A-3;
- сопротивление: 0...900(2000) Ом (датчик положения задвижки).

Модули дискретного ввода МВ110-224.16Д (рис. 9) предназначены для сбора данных со встроенных дискретных входов и передачи их в сеть RS-485 [4].



Рис. 9. Модуль MB110-224.16Д

Данные модули имеют 16 дискретных входов. Они поддерживают сигналы:

- сухой контакт (не требует внешнего питания);
- датчик n-p-n типа.

Модули аналогового вывода МУ110-224.6У (рис. 10) предназначены для преобразования цифровых сигналов, передаваемых по сети RS-485, в аналоговые сигналы для управления исполнительными механизмами или для передачи сигналов приборам регистрации и самописцам [5].

Данные модули имеют 6 аналоговых выходов 0-10В.



Рис. 10. Модуль МУ110-224.6У

Модули дискретного вывода МУ110-224.16Р (рис. 11) предназначены для управления по сигналам из сети RS-485 встроенными дискретными выходными элементами, используемыми для подключения исполнительных механизмов с дискретным управлением [6].



Рис. 11. Модуль МУ110-224.16Р

Данные модули имеют 16 дискретных выходов типа электромагнитное реле.

2.2. Добавление модулей к контроллеру

Для подключения модулей к контроллеру необходимо [7]:

1. Добавить устройство (рис. 12);

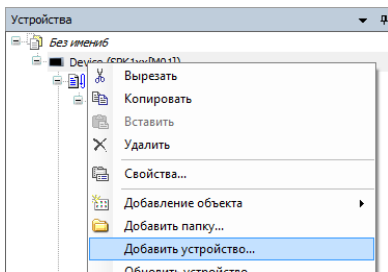


Рис. 12. Добавление устройства

2. Добавить **Modbus_COM** (рис. 13);

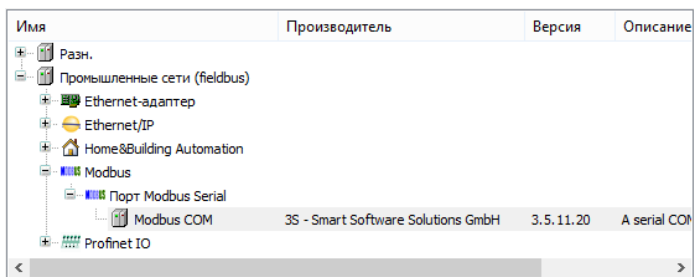


Рис. 13. Добавление Modbus COM

3. Добавить **Modbus_Master_COM_Port** (рис. 14);

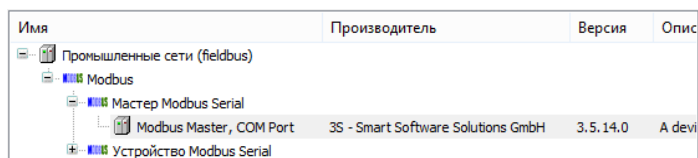


Рис. 14. Добавление Modbus Master COM Port

4. Добавить необходимые модули (например **MB110-224.16Д**) (рис. 15).

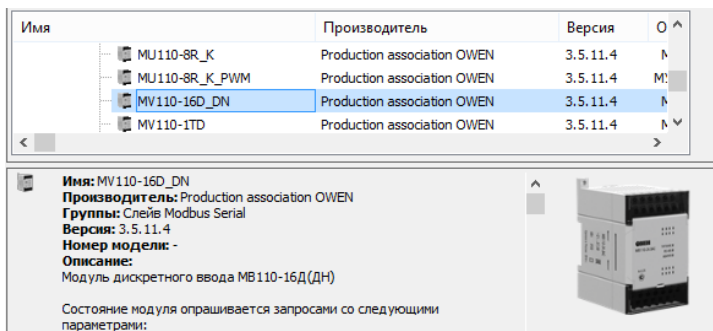


Рис. 15. Добавление необходимых модулей

2.3. Настройка модулей

1. Для объекта **Modbus_COM** задать скорость **115200**, четность **НЕТ**, номер порта **COM3**, как показано на рис. 16.

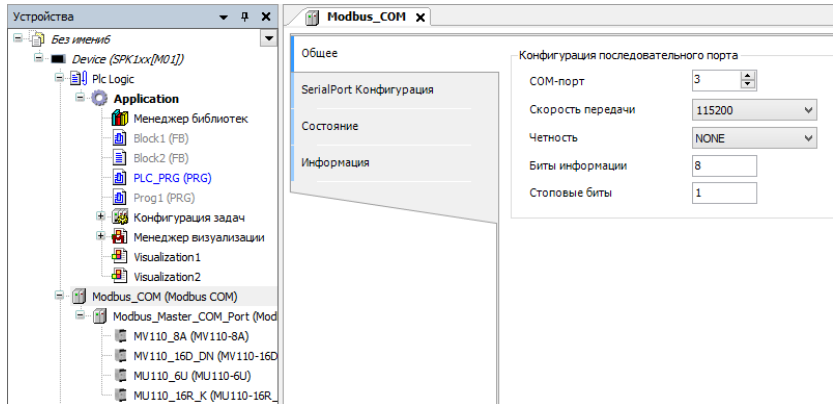


Рис. 16. Настройки объекта Modbus COM

2. Для первого из модулей (**MV110_8A**) необходимо указать адрес **1** (рис. 17).

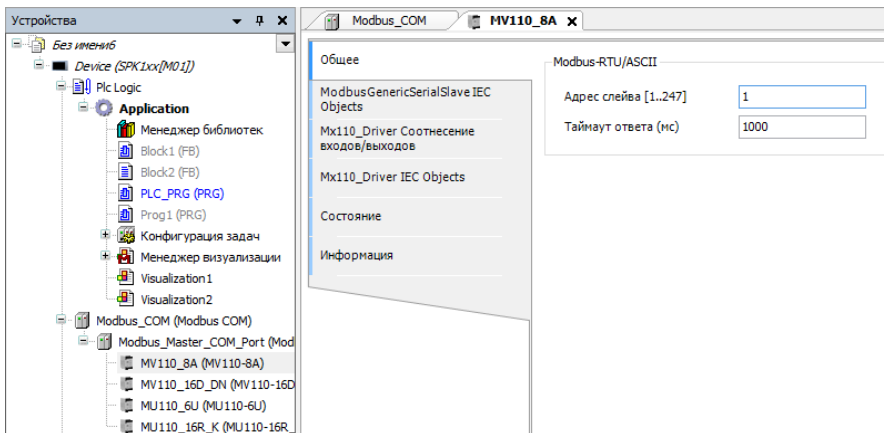


Рис. 17. Адрес для первого модуля MV110_8A

3. Для остальных модулей задать адреса **2, 3, 4**.

Контрольные вопросы

1. Для чего нужны внешние модули к контроллеру СПК110?
2. Какие параметры порта Вы знаете?
3. За что отвечает параметр порта «четность»?
4. По какому протоколу будет осуществляться обмен данными между внешними модулями и контроллером СПК110?

3. РАБОТА С ПРОГРАММОЙ НА CFC

Редактор программы **CFC** имеет внешний вид, показанный на рис. 18, и состоит из:

- графического редактора;
- панели инструментов с добавляемыми блоками (на рис. 18 выделены 4 наиболее используемых блока);
- списка переменных, который может редактироваться вручную, но в большинстве случаев он заполняется автоматически [8].

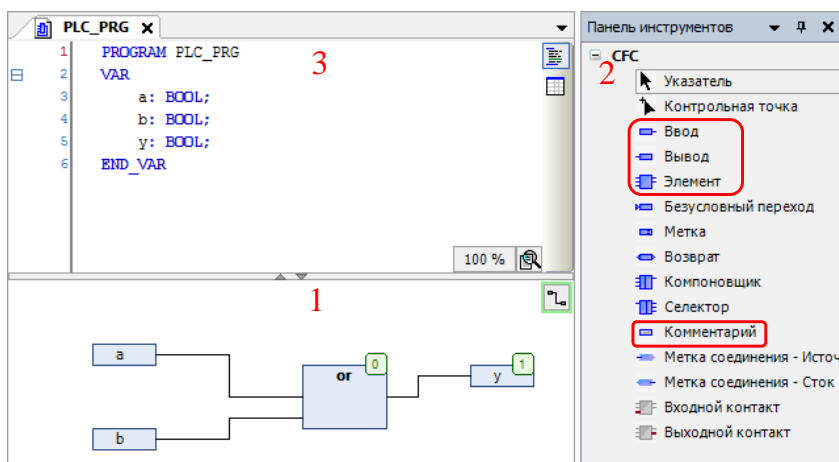


Рис. 18. Внешний вид редактора программы: 1 – графический редактор, 2 – панель инструментов, 3 – список переменных

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. 19).

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. 20).

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

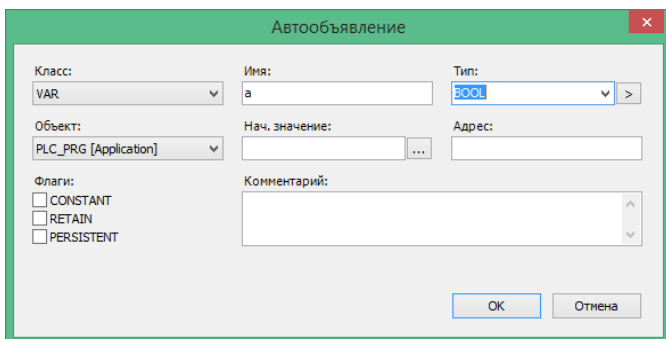


Рис. 19. Добавление переменной с помощью окна Автообъявление

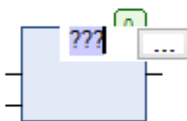


Рис. 20. Добавление имени для «Элемента»

3.1. 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 входы / выходы 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)

3.2. Пример программы на CFC

Программа на языке CFC, реализующая описанное ранее задание, выглядит, как показано на рис. 21.

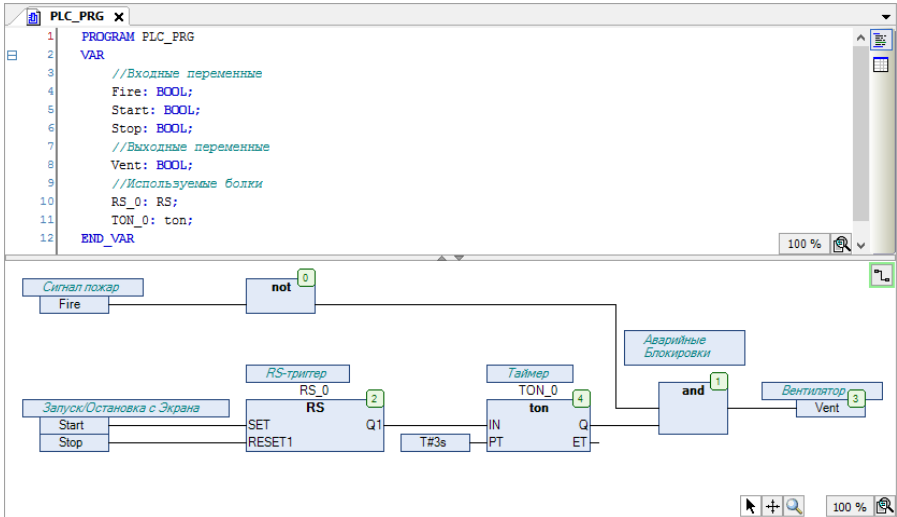


Рис. 21. Внешний вид созданной программы

!!! Комментарии на русском языке обязательны!

3.3. 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 следующей последовательности: элементы, относящиеся к основной программе; элементы, относящиеся к ручному управлению; элементы блокировок.

Все перечисленные выше рекомендации изображены схематически на рис. 22.

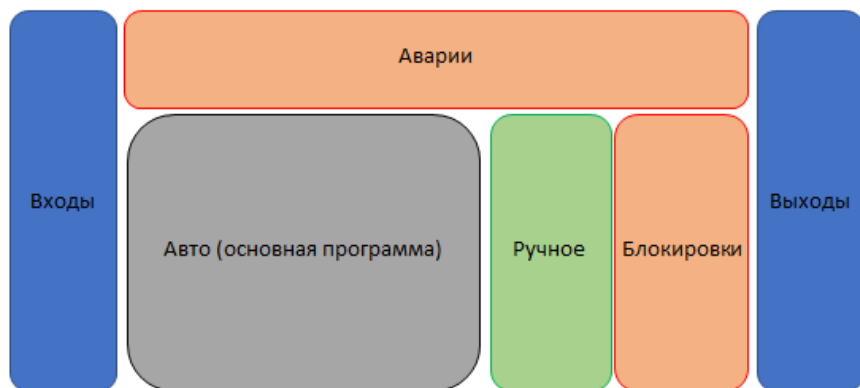


Рис. 22. Расположение элементов программы

Контрольные вопросы

1. Из чего состоит редактор программы CFC?
2. Как добавить и объявить в программе Ввод / Вывод и Элемент?
3. Какие виды сигналов управления Вы знаете? Приведите примеры использования каждого из них.
4. Опишите, почему рекомендуется располагать элементы программы, относящиеся к блокировкам, после элементов, относящихся к ручному управлению?

4. ОСНОВНЫЕ БЛОКИ СFC

Чтобы вызвать окно Ассистент ввода, необходимо щелкнуть правой кнопкой мыши по Элементу и в появившемся меню выбрать пункт Ассистент ввода (рис. 23).

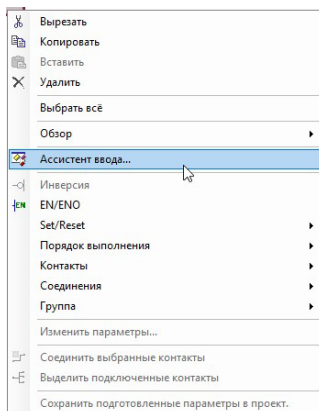


Рис. 23. Вызов окна Ассистента ввода

Окно Ассистента ввода выглядит, как показано на рис. 24.

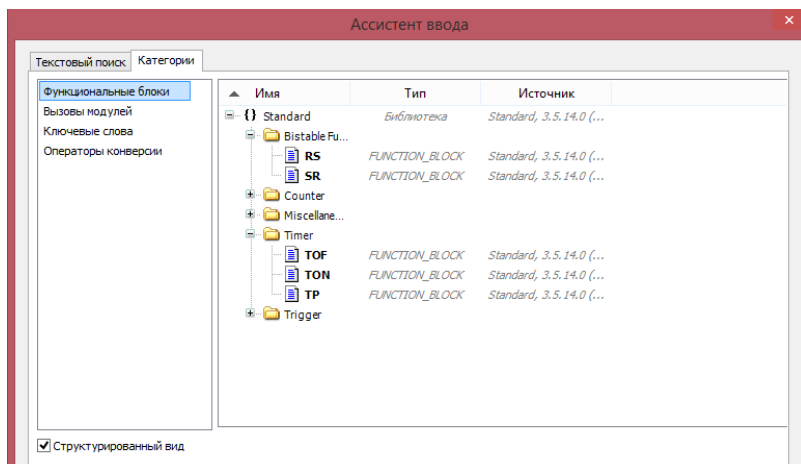


Рис. 24. Окно Ассистента ввода

4.1. Список ключевых слов

Ключевые слова находятся в одноименном разделе окна Ассистента ввода (см. рис. 24).

Список наиболее важных ключевых слов представлен в табл. 2.

Таблица 2 – Список ключевых слов

Ключевое слово	Назначение
ABS	Модуль числа. Например: $ABS(-5) = 5$
ACOS	Арккосинус числа
ADD	Сложение. Например: $5 + 7 = 12$
AND	Логическое «И»
ASIN	Арсинус числа
ATAN	Арктангенс числа
COS	Косинус числа
DIV	Деление (для Real), например: $14/5 = 2.8$. Целочисленное деление (для Int), возвращает целую часть частного, например: $14 \text{ div } 5 = 2$
EQ	«Равно»
EXP	Экспоненциальная функция e^x
EXPT	Возведение одной переменной в степень другой. Например: $2^3 = 8$
GE	«Больше или равно»
GT	«Больше, чем»
LE	«Меньше или равно»
LIMIT	Установить пределы. Например: $Y = \text{limit}(0, X, 100)$, тогда Y всегда будет лежать в интервале $[0, 100]$
LN	Натуральный логарифм числа
LOG	Десятичный логарифм числа
LT	«Меньше, чем»
MAX	Максимум. Например: $\text{max}(5, 7) = 7$
MIN	Минимум. Например: $\text{min}(5, 7) = 5$
MOD	Целочисленное деление, возвращает остаток частного. Например: $14 \text{ mod } 5 = 4$
MUL	Умножение. Например: $2*2 = 4$
MUX	Мультиплексор
NE	«Не равно»
NOT	Логическое «НЕ»
OR	Логическое «ИЛИ»
ROL	Побитовое вращение операнда влево
ROR	Побитовое вращение операнда вправо
SEL	Выбор, переключатель

Ключевое слово	Назначение
SHL	Побитовый сдвиг операнда влево. Например: $3 \text{ shl } 1 = 6$
SHR	Побитовый сдвиг операнда вправо. Например: $3 \text{ shr } 1 = 1$
SIN	Синус числа
SIZEOF	Определение числа байт, занимаемых переменной
SQRT	Квадратный корень числа. Например: $\text{sqrt}(36) = 6$
SUB	Вычитание. Например: $12 - 7 = 5$
TAN	Тангенс числа
TRUNC	Округление до целой части. Конверсия типа данных REAL в тип данных DINT
TRUNC_INT	Округление до целой части. Конверсия типа данных REAL в тип данных INT
XOR	Логическое «ИСКЛЮЧАЮЩЕЕ ИЛИ»

4.2. Логические блоки

К логическим блокам в CoDeSys 3.5 относятся блоки: not, or, and, xor. Ниже приведены таблицы истинности для каждой из перечисленных логических функций (рис. 25 – рис. 28)

a	NOT a
0	1
1	0

Рис. 25. Таблица истинности для логической функции «NOT»

a	b	a OR b
0	0	0
0	1	1
1	0	1
1	1	1

Рис. 26. Таблица истинности для логической функции «OR»

a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1

Рис. 27. Таблица истинности для логической функции «AND»

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Рис. 28. Таблица истинности для логической функции «XOR»

Законы алгебры логики

Операции с константами:

$$\begin{aligned}
 A + 0 &= A, & A \cdot 0 &= 0, \\
 A + 1 &= 1, & A \cdot 1 &= A, \\
 A + \bar{A} &= 1, & A \cdot \bar{A} &= 0.
 \end{aligned}$$

Повторения:

$$A + A = A, \quad A \cdot A = A.$$

Поглощения:

$$\begin{aligned}
 A + A \cdot B &= A, & A \cdot (A + B) &= A, \\
 A + \bar{A} \cdot B &= A + B, & A \cdot (\bar{A} + B) &= A \cdot B.
 \end{aligned}$$

Склеивания:

$$A \cdot B + \bar{A} \cdot B = B, \quad (A + B) \cdot (\bar{A} + B) = B.$$

Двойственности (закон де Моргана):

$$\overline{A + B} = \bar{A} \cdot \bar{B}, \quad \overline{A \cdot B} = \bar{A} + \bar{B}.$$

Двойного отрицания:

$$\overline{\bar{A}} = A.$$

4.3. Триггеры

В стандартной библиотеке CoDeSys 3.5 имеется два вида триггеров: с приоритетом установки (SR-триггер) и с приоритетом сброса (RS-триггер).

На рис. 29 представлены блоки RS и SR триггеров.



Рис. 29. Блоки RS и SR триггеров

Блок SR-триггера используется для переключения с фиксацией состояния при поступлении коротких импульсов на соответствующий вход. Работу поясняет приведенная на рис. 30 диаграмма и таблица истинности.

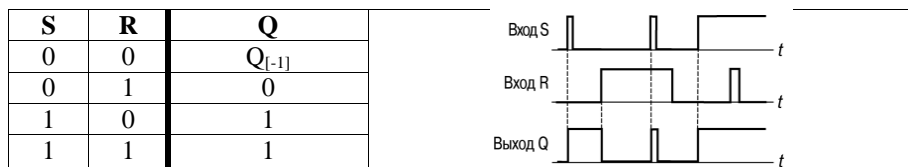


Рис. 30. Диаграмма работы и таблица истинности SR-триггера

По переднему фронту сигнала на входе S (Set, установить) на выходе Q блока появится логическая «1», т. е. выход включается. По переднему фронту сигнала на входе R (Reset, сбросить) на выходе Q блока появится логический «0», т. е. выход выключается.

При одновременном поступлении сигналов на входы R и S приоритетным является сигнал входа S [9].

Обозначение $Q_{[-1]}$ означает «предыдущее» значение выхода. При этом обозначение Q (без индекса) означает «текущее» значение выхода.

Аналогичным образом, но с приоритетом по входу Reset работает и RS-триггер. Работу RS-триггера поясняют приведенные на рис. 31 диаграмма и таблица истинности.

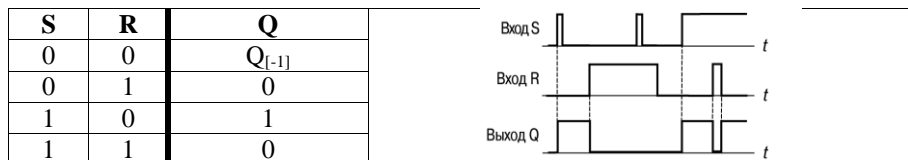


Рис. 31. Диаграмма работы и таблица истинности RS-триггера

При одновременном поступлении сигналов на входы, приоритетным является сигнал входа R [9].

4.4. Задержки включения и выключения

Для того чтобы обеспечить задержку включения или выключения, используются блоки ton (рис. 32) и tof (рис. 33) соответственно.

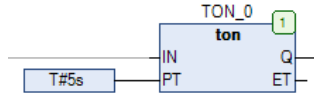


Рис. 32. Внешний вид блока задержки включения ton

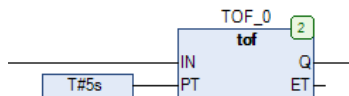


Рис. 33. Внешний вид блока задержки выключения tof

Принцип работы блока ton заключается в следующем. Когда вход получает значение TRUE, начинается отсчет заданного времени, по истечении которого значение TRUE поступает и на выход. На рис. 34 представлен график работы блока ton.

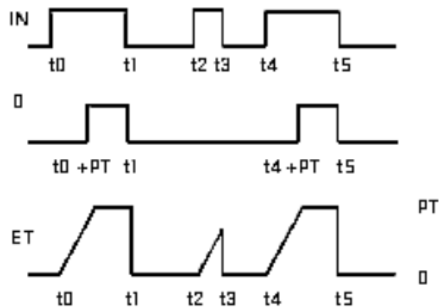


Рис. 34. Графическое представление работы TON

Аналогичным образом, но на значение FALSE работает и блок tof (рис. 35).

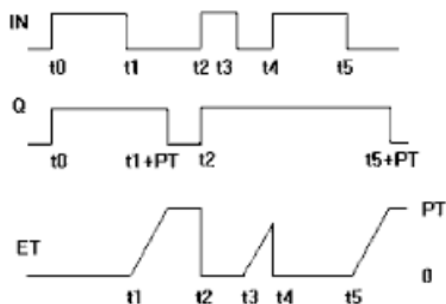


Рис. 35. Графическое представление работы TOF

4.5. Блок выбора Sel

Блок Sel (Select), представленный на рис. 36, осуществляет переключение между двумя вариантами In0 и In1 (см. пояснения на рис. 37). При этом выбор активной линии осуществляется с помощью первого сигнала G.

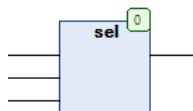


Рис. 36. Внешний вид блока sel

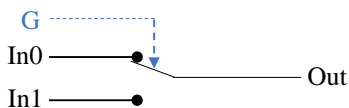


Рис. 37. Принцип работы блока sel

4.6. ПИД-регулятор

Для того чтобы обеспечить работу системы регулирования, необходимо использовать блок ПИД-регулятора. Для этого требуется выполнить следующие действия.

1. Создать новый проект (как описано в п. 1.3).
2. В **PLC_PRG** добавить новый элемент и задать ему имя **pid** (рис. 38).

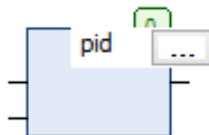


Рис. 38. Элемент pid

После нажатия клавиши Enter и создания переменной, блок принимает вид, показанный на рис. 39.

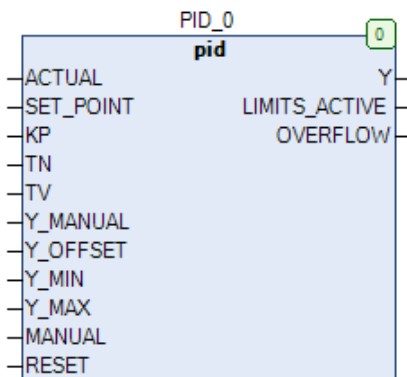


Рис. 39. Вид блока pid

Описание входов и выходов блока PID представлено в табл. 3.

3. Запустить компиляцию, в результате которой будет получено две ошибки (рис. 40). Это связано с тем, что к проекту не подключена требуемая библиотека.

4. Открыть менеджер библиотек (рис. 41).

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

5. Нажать кнопку «Добавить библиотеку» и добавить библиотеку «Util» (рис. 42).

Таблица 3 – 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
LIMITS_ACTIVE		BOOL	Выход значения Y за заданные пределы (Y_MIN и Y_MAX)
OVERFLOW		BOOL	Переполнение интегральной части

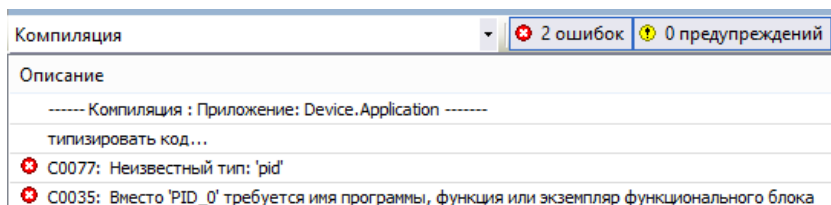


Рис. 40. Ошибки компиляции

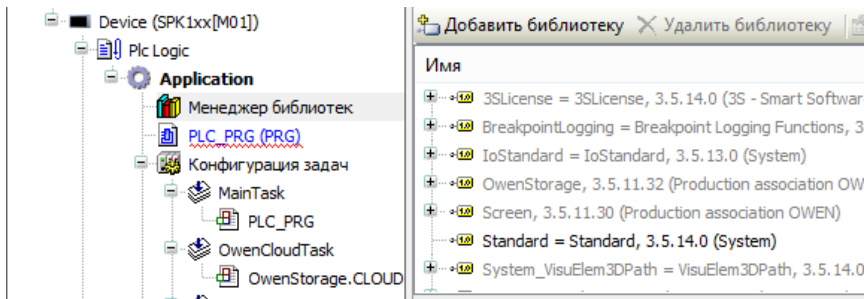


Рис. 41. Менеджер библиотек

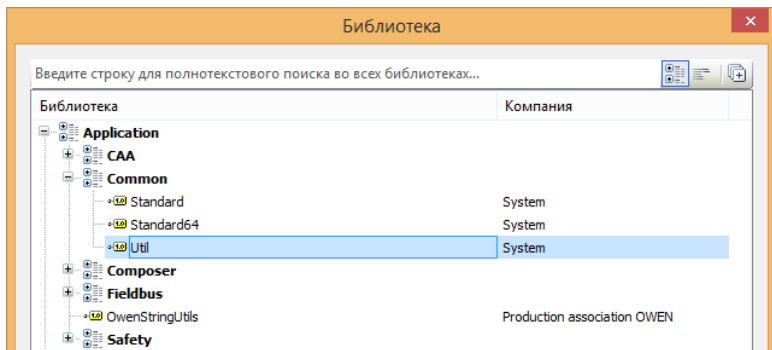


Рис. 42. Добавление библиотеки Util

4.7. Система регулирования

На рис. 43 представлен пример модели системы регулирования температуры, состоящей из блоков pid (ПИД-регулятор, описанный выше в пункте 4.6) и блока Obj (варианты объектов описаны далее в пунктах 7.5 – 7.8).

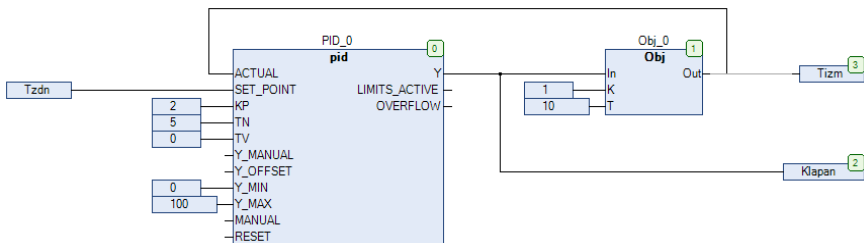


Рис. 43. Модель системы регулирования температуры

4.8. Генераторы Gen

Функция GEN применяется для генерации типичных периодических функций. В табл. 4 и табл. 5 представлено описание входных и выходных переменных для блока gen (рис. 44).

Таблица 4 – Описание входов блока gen

Переменная	Тип данных	Начальное значение	Описание
MODE	GEN_MODE		<p>Генерируемая функция</p> <ul style="list-style-type: none"> – TRIANGLE: треугольная функция от -AMPLITUDE до +AMPLITUDE – TRIANGLE_POS: треугольная функция от 0 до +AMPLITUDE – SAWTOOTH_RISE: Пилообразная функция, увеличивающаяся от -AMPLITUDE до +AMPLITUDE – SAWTOOTH_FALL: Пилообразная функция, уменьшающаяся от -AMPLITUDE до +AMPLITUDE – RECTANGLE: Прямоугольная функция, переключающаяся с -AMPLITUDE на +AMPLITUDE – SINE: Синусоидальная функция – COSINE: Косинусоидная функция
ASE	BOOL		<ul style="list-style-type: none"> – FALSE: Период функции, относящийся к номерам вызовов (CYCLES) – TRUE: Период функции, относящийся к времени (PERIOD)

Переменная	Тип данных	Начальное значение	Описание
PERIOD	TIME	TIME#1s0ms	Время периода; релевантно, только если BASE = TRUE
CYCLES	INT	1000	Количество вызовов за период; релевантно, только если BASE = FALSE
AMPLITUDE	INT		Амплитуда функции
RESET	BOOL		Сброс функционального блока

Таблица 5 – Описание выходов блока gen

Переменная	Тип данных	Описание
OUT	INT	Переменная генерируемой функции

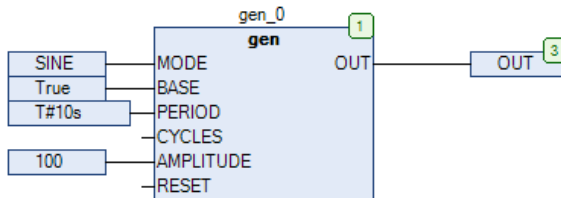


Рис. 44. Блок генератора синуса

На рис. 45 представлен график, полученный с помощью блока генератора синуса.

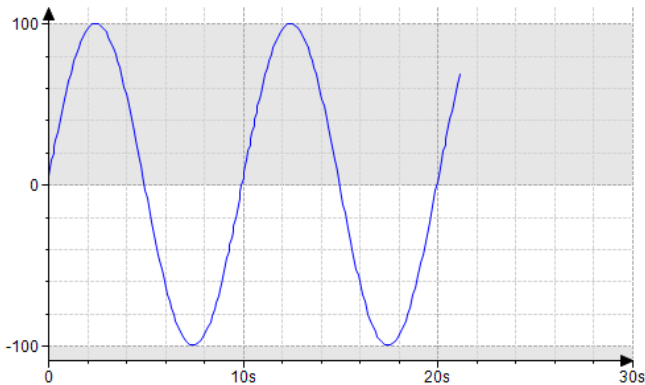


Рис. 45. Итоговый график синуса

Если запустить компиляцию, то будет получено две ошибки (рис. 46). Это связано с тем, что к проекту не подключена требуемая библиотека.

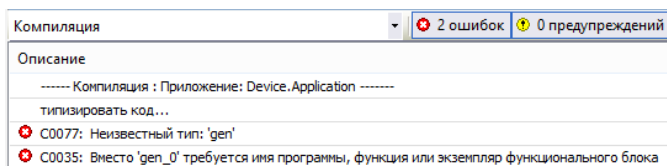


Рис. 46. Ошибки компиляции

Необходимо открыть менеджер библиотек (рис. 47).

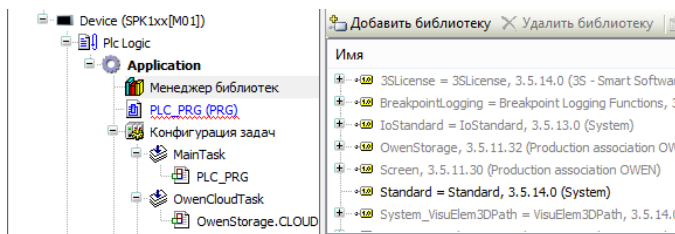


Рис. 47. Менеджер библиотек

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

Нажать кнопку «Добавить библиотеку» и добавить библиотеку «Util» (рис. 48).

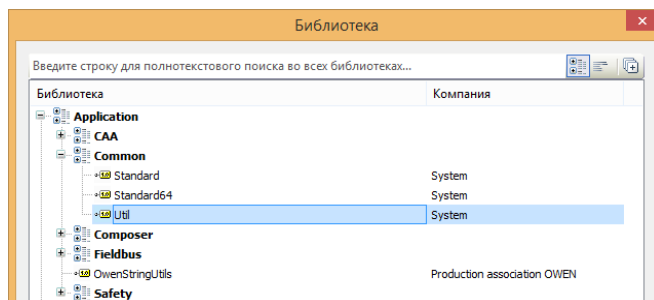


Рис. 48. Добавление библиотеки Util

4.9. Получение справки по блокам

Чтобы получить справку в CoDeSys 3.5, в том числе по блокам, необходимо выбрать пункт меню Справка и перейти в раздел Указатель (или воспользоваться «горячими» клавишами Ctrl+Shift+F2) (рис. 49).

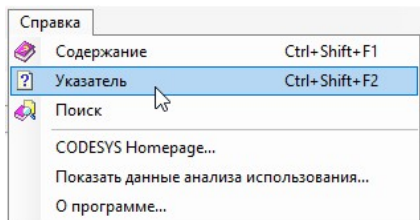


Рис. 49. Вызов указателя

После этого откроется окно онлайн-справки. В зависимости от наличия соединения с Интернет данное окно может открываться либо внутри программы (рис. 50), либо в браузере (рис. 51).

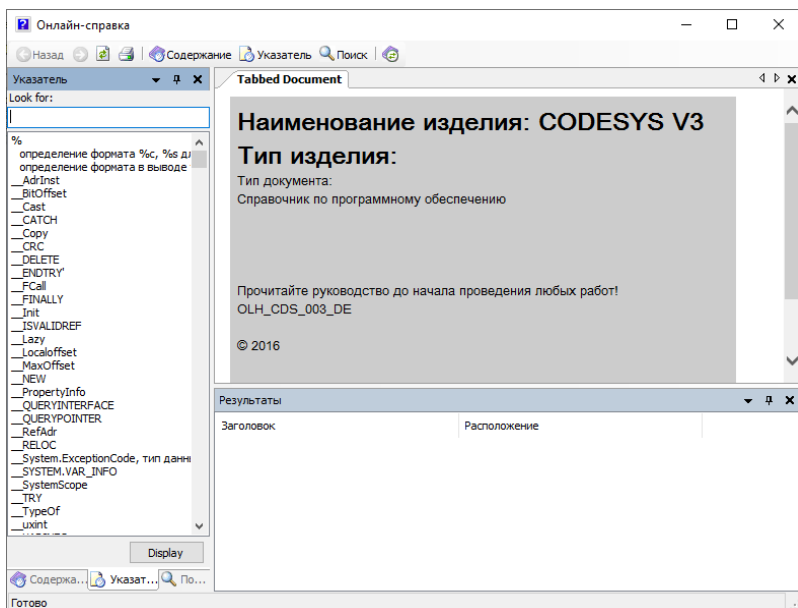


Рис. 50. Окно онлайн-справки внутри программы

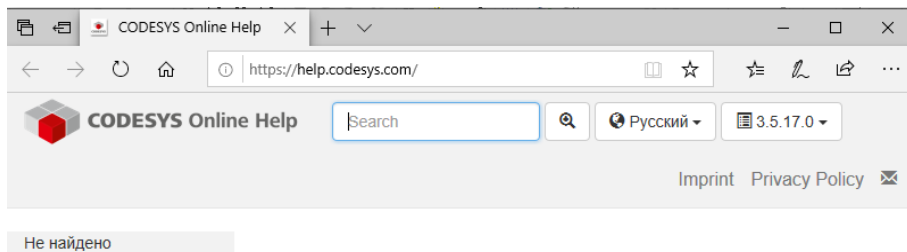
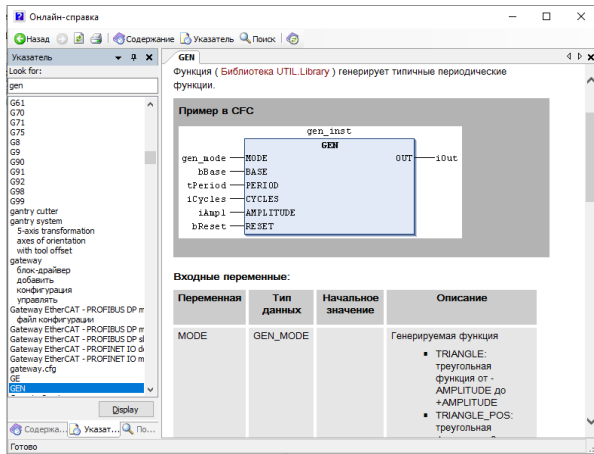


Рис. 51. Окно онлайн-справки в браузере

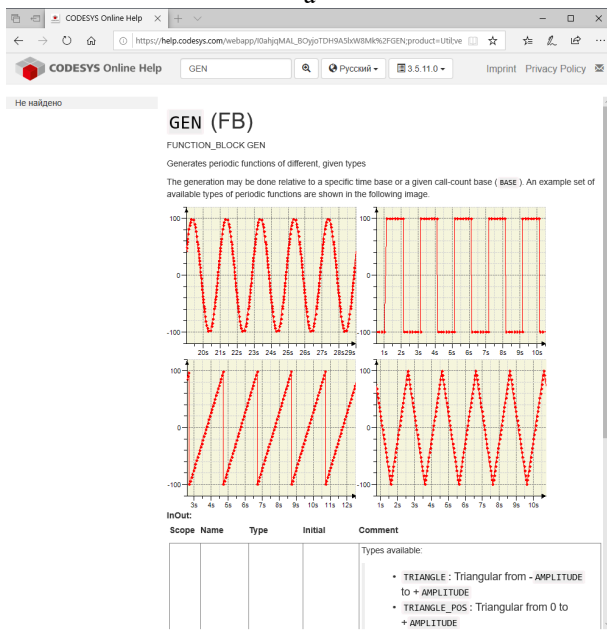
Чтобы найти необходимую информацию по блоку, требуется в строке поиска ввести имя блока. После чего на экране появится вся необходимая информация (рис. 52а, б).

Контрольные вопросы

1. Перечислите основные логические блоки, которые Вы знаете. Напишите основные законы алгебры логики.
2. В чем отличие триггера RS от SR?
3. В какой ситуации необходимо использовать блок ton, а в какой – tof? Приведите примеры.
4. Как работает блок sel в CoDeSys 3.5? В каких случаях он может быть использован?
5. Как Вы думаете, какие входы и выходы блока регулятора pid необходимо задать обязательно для работы системы регулирования, а какие используются дополнительно для большего удобства использования системы регулирования?
6. Какие функции формируются генератором gen? Перечислите их.



а



б

Рис. 52. Отображение информации в окне справки: а – внутри программы; б – в браузере

5. РАЗРАБОТКА ЭКРАНОВ

5.1. Редактор экранов

Необходимо разработать экран, представленный на рис. 53.

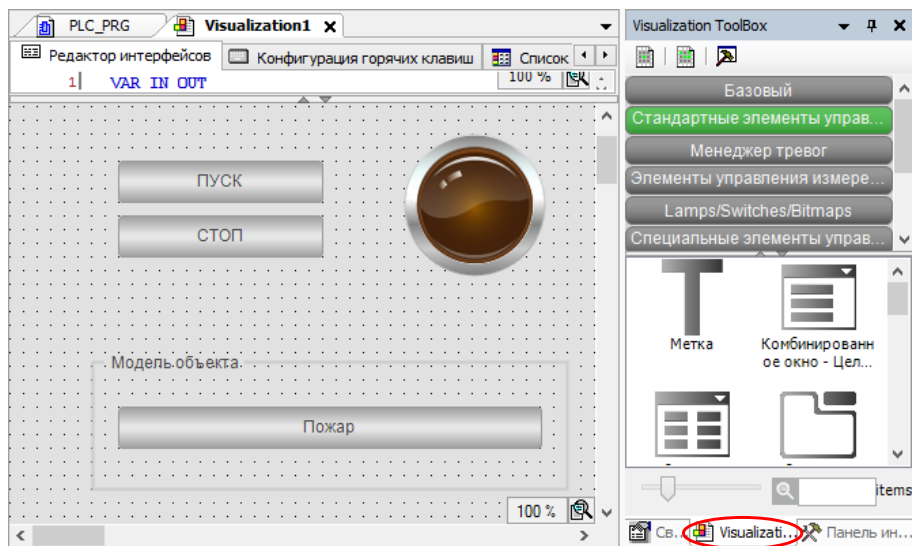


Рис. 53. Пример экрана для создаваемой программы

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

5.2. Связывание экрана с программой

1. Выделив кнопку **ПУСК**, следует выбрать в ее свойствах **OnMouseDown** (нажатие кнопки мыши) (рис. 54).

2. Затем необходимо привязать к кнопке выполнение кода на языке **ST** (рис. 55).

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

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

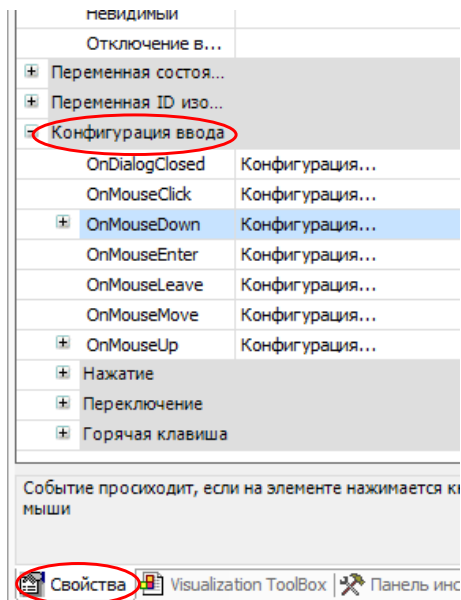


Рис. 54. Выбор свойства OnMouseDown

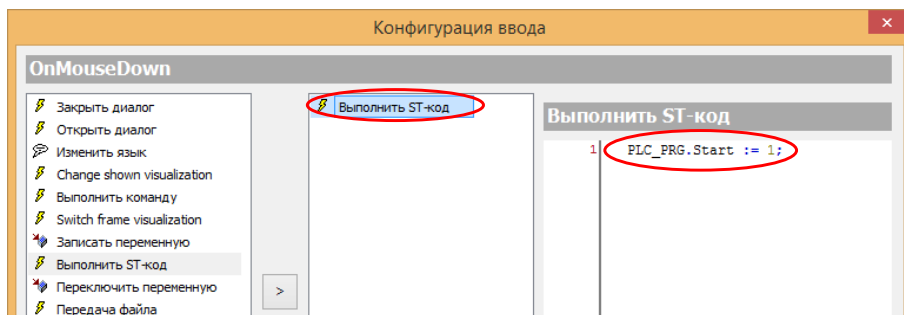


Рис. 55. Привязка кода на ST к кнопке

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. 56).

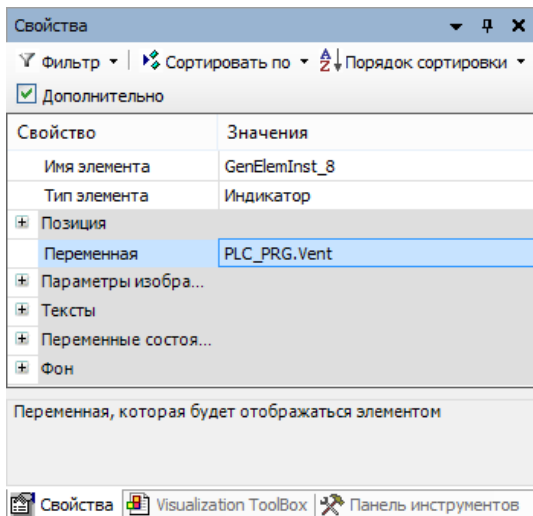


Рис. 56. Привязка переменной в свойствах индикатора

5. Затем нужно выделить кнопку **Пожар** и для ее 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. 57).

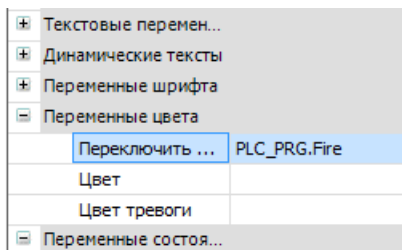


Рис. 57. Задание отображения текущего значения переменной цветом

Также для этой кнопки необходимо изменить «Цвет тревоги» на красный (рис. 58).

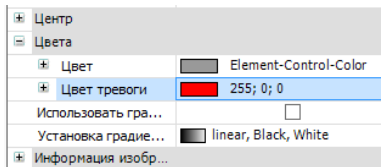


Рис. 58. Выбор цвета тревоги для кнопки

5.3. Ввод и отображение числовых значений

Для входного значения необходимо добавить элемент «Управление вращением» (рис. 59).

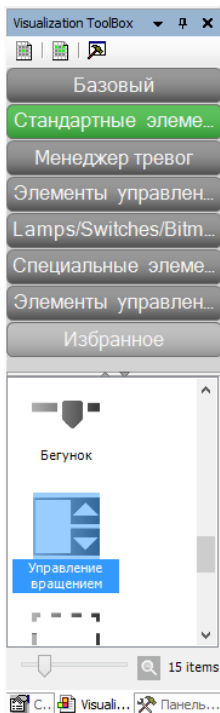
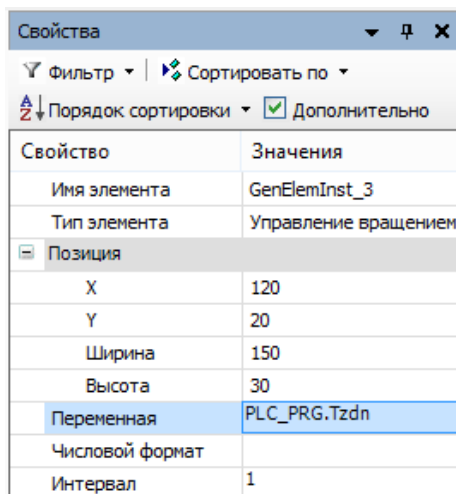


Рис. 59. Элемент «Управление вращением»

В его свойствах следует указать требуемую входную переменную (рис. 60).



Свойство	Значения
Имя элемента	GenElemInst_3
Тип элемента	Управление вращением
[-] Позиция	
Х	120
У	20
Ширина	150
Высота	30
Переменная	PLC_PRG.Tzdn
Числовой формат	
Интервал	1

Рис. 60. Задание входной переменной для элемента «Управление вращением»

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.

Для выходного (отображаемого) значения нужно добавить элемент «Прямоугольник» (рис. 61).

В его свойствах необходимо указать требуемую переменную и формат отображаемого значения, как показано на рис. 62.

!!! 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. 63.

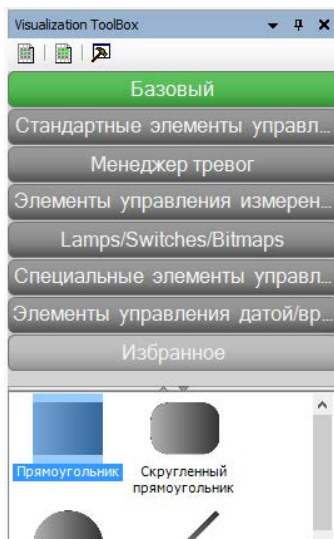


Рис. 61. Элемент «Прямоугольник»

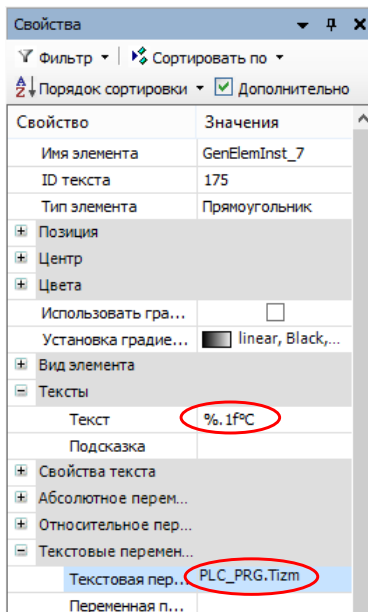


Рис. 62. Задание переменной и формата отображаемого значения для элемента «Прямоугольник»

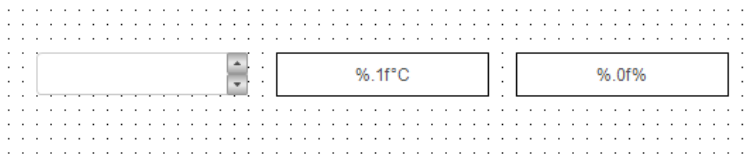


Рис. 63. Отображение значений

5.4. Отображение графиков

Для добавления графика используется элемент «Трассировка» (рис. 64).

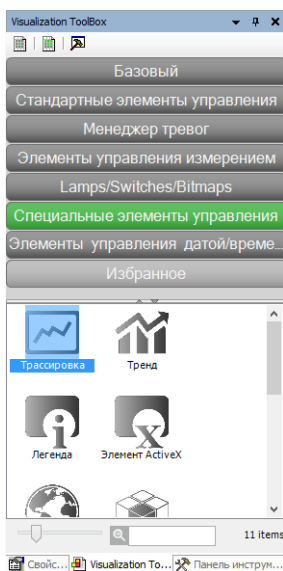


Рис. 64. Элемент «Трассировка»

В свойствах графика нужно выбрать «Трассировка» (рис. 65).

В качестве «Задачи» нужно выбрать MainTask, а затем добавить переменную (рис. 66).

Всего нужно добавить две переменные, представленные на рис. 67. Одна из них – это заданное значение, другая – измеренное.

Также следует провести настройку Отображения (рис. 68) и Дополнительные параметры (рис. 69).

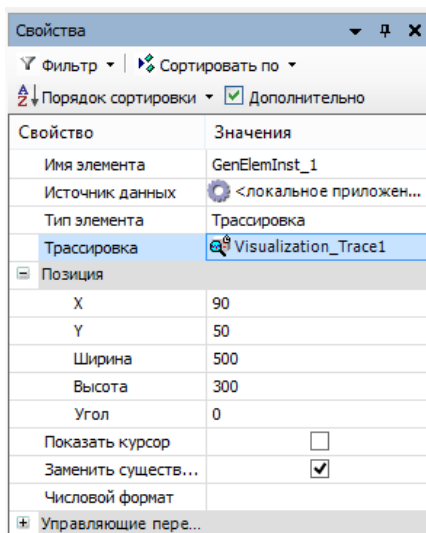


Рис. 65. Свойство графика «Трассировка»

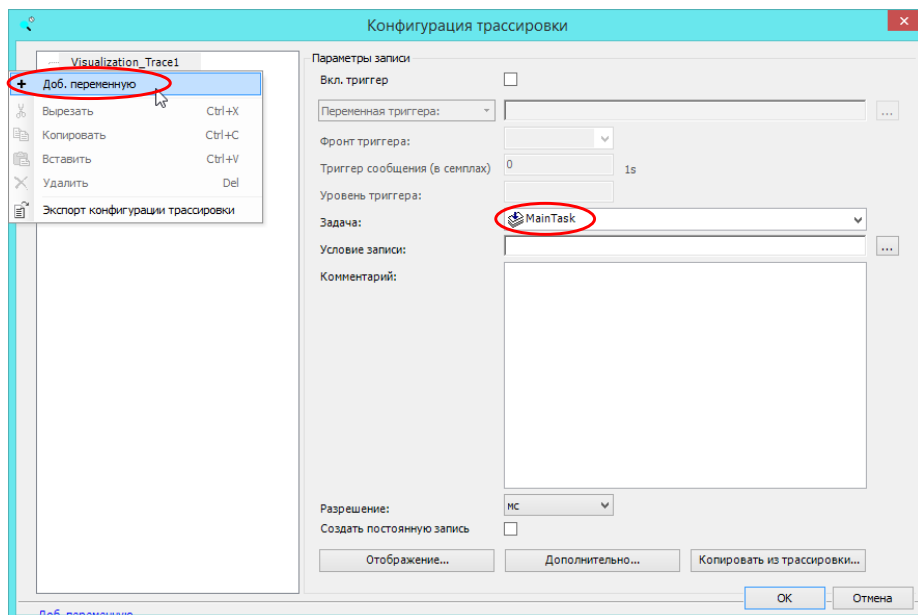


Рис. 66. Выбор задачи

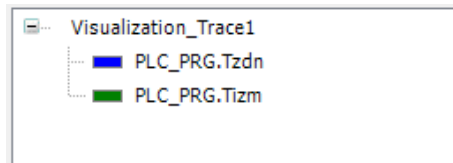


Рис. 67. Добавляемые переменные

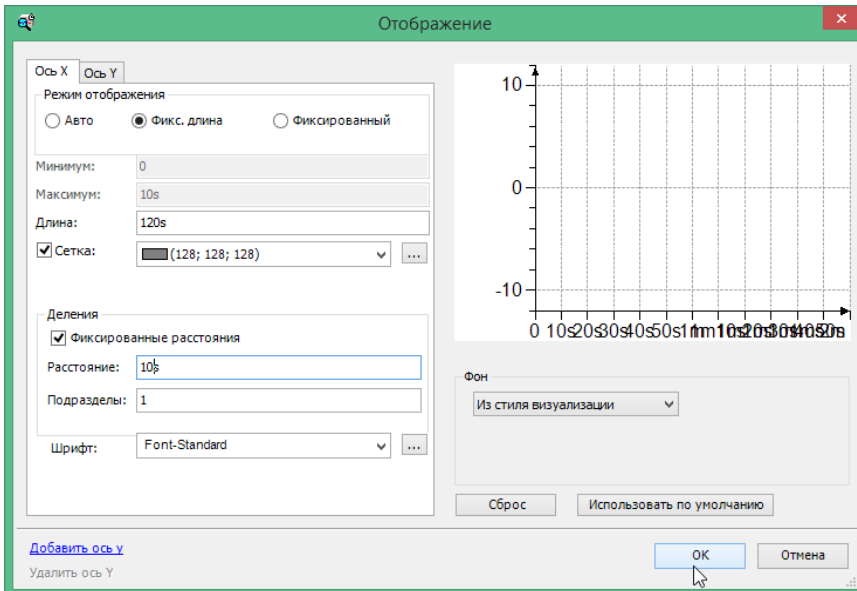


Рис. 68. Настройка отображения

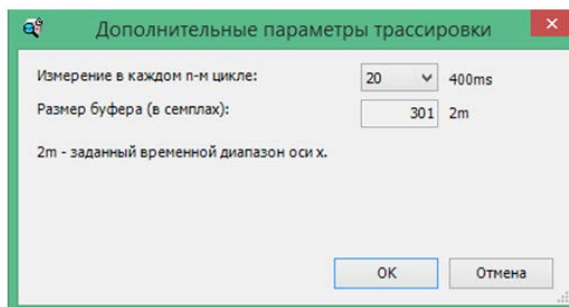


Рис. 69. Настройка дополнительных параметров

На рис. 70 изображен экран, полученный в итоге.

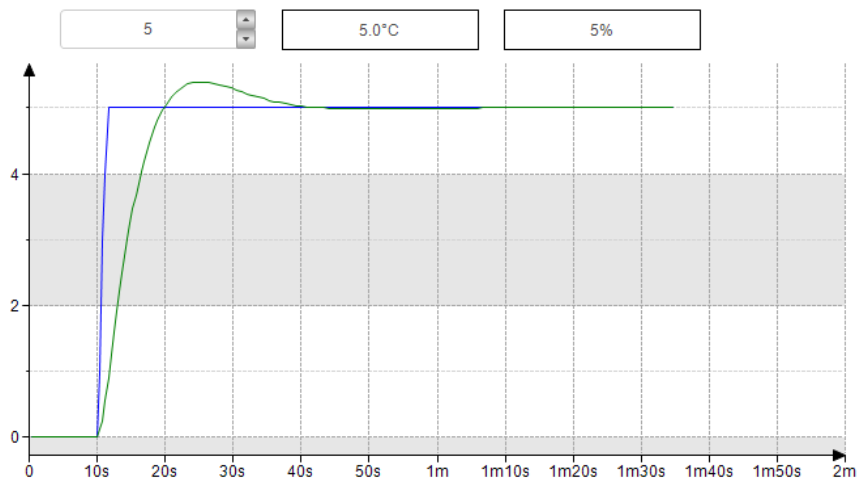


Рис. 70. Итоговый экран

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

5.5. Всплывающие окна для ввода значений

Всплывающее окно для ввода значений можно настроить на щелчок кнопкой мыши `OnMouseClicked` (рис. 71). Данное окно может быть задано, например, для прямоугольника, отображающего заданное значение уровня. Настройки окна представлены на рис. 72.

В итоге при щелчке мышкой по полю с заданным значением уровня появится клавиатура, представленная на рис. 73.

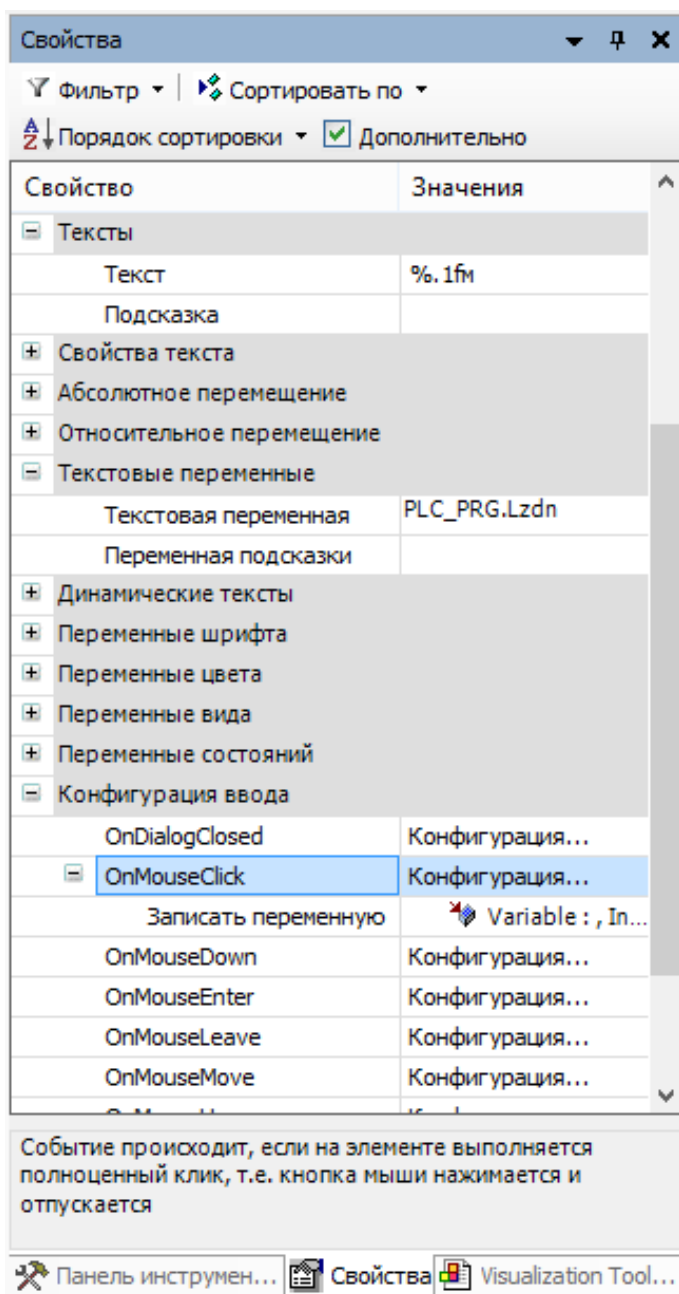


Рис. 71. Добавление OnMouseClicked

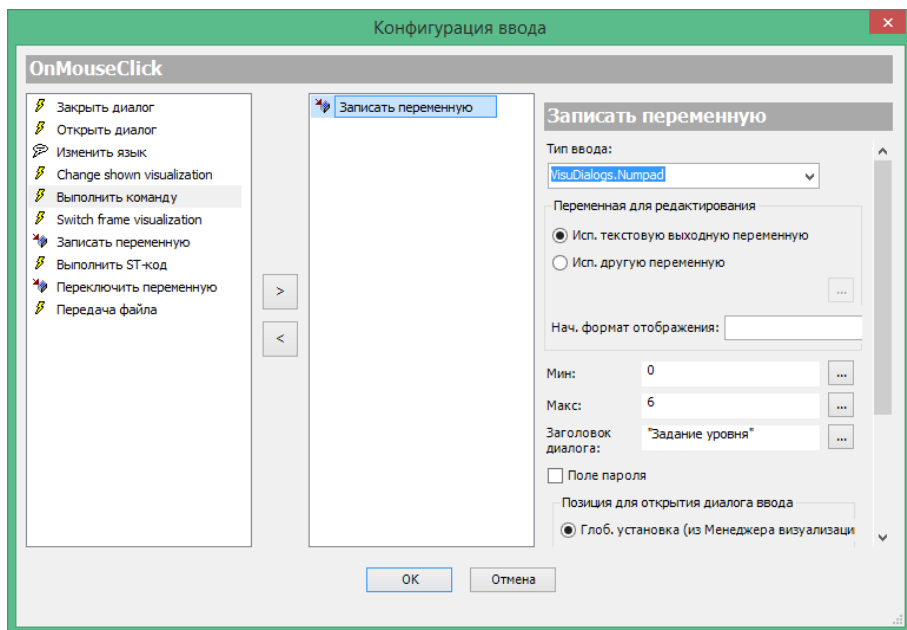


Рис. 72. Настройка ввода значения

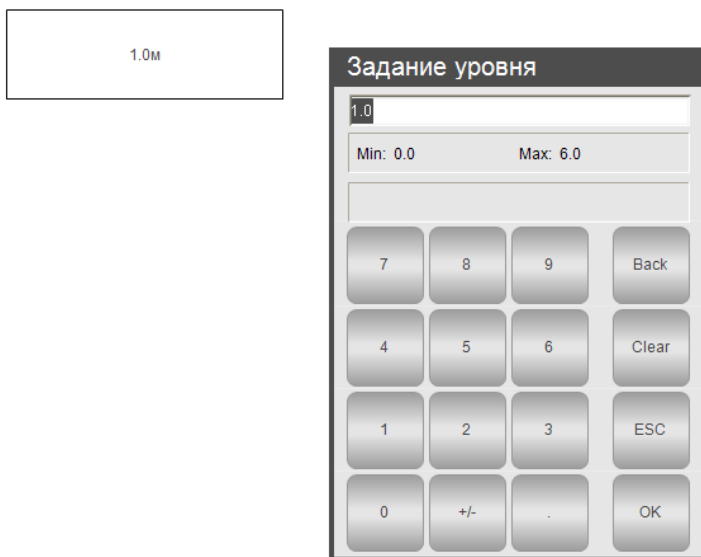


Рис. 73. Пример ввода значения

5.6. Дополнительные возможности при создании экрана

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. 74).

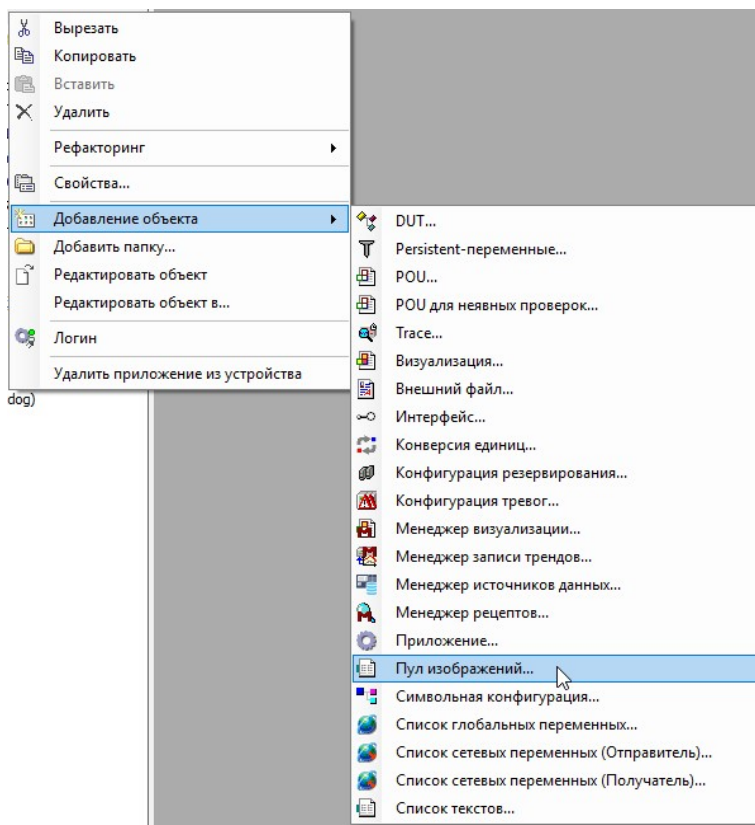


Рис. 74. Добавление Пула изображений

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. 75).

ID	Имя файла	Изображение	Тип ссылки

Рис. 75. Выбор ячейки Имя файла

4. В появившемся списке найти необходимое изображение и задать настройки, как показано на рис. 76.

Выбор изображения

Файл:
C:\Users\Мария\Desktop\Shema.JPG

Что следует сделать с графическим файлом?

Запомнить связь
 Запомнить связь и включить в проект.
 Включить в проект.

Если файл изменился, то

перезагружать файл автоматически.
 спрашивать о перезагрузке файла.
 ничего не делать.

OK Отмена

Рис. 76. Выбор изображения и его настроек

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. 77).


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

Рис. 77. Пул изображений, после выбора изображения

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. 78).

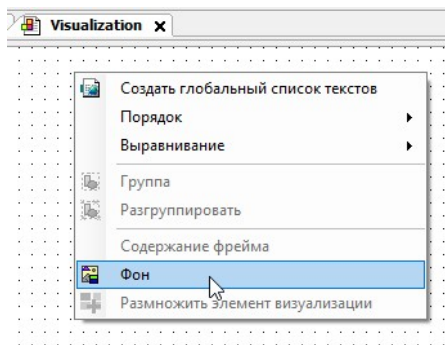


Рис. 78. Выбор пункта Фон

8. В появившемся окне поставить галочку **Изображение**, после чего добавить необходимое изображение из списка (рис. 79).

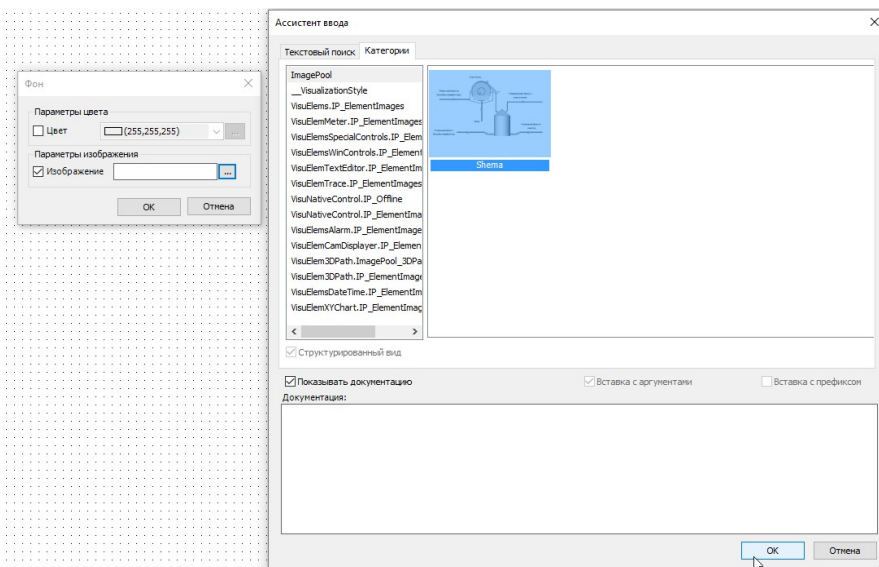


Рис. 79. Добавление изображения в качестве фона

После проделанных операций появится фон экрана оператора, поверх которого можно будет размещать различные элементы управления (рис. 80).

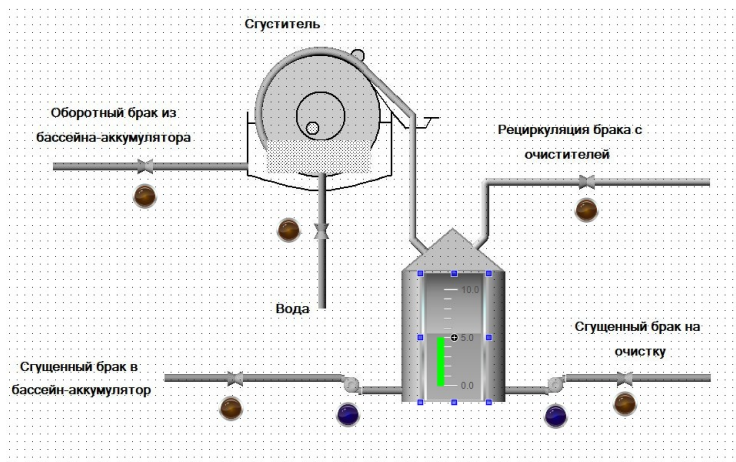


Рис. 80. Экран оператора с фоном

Добавлять графические файлы можно не только для создания фона экрана [10]. 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.
6. Как создать всплывающее окно для ввода значений?

6. ЗАПУСК ЭМУЛЯЦИИ

Для запуска режима эмуляции необходимо проделать следующие действия:

1. Включить режим **Эмуляция** (рис. 81).

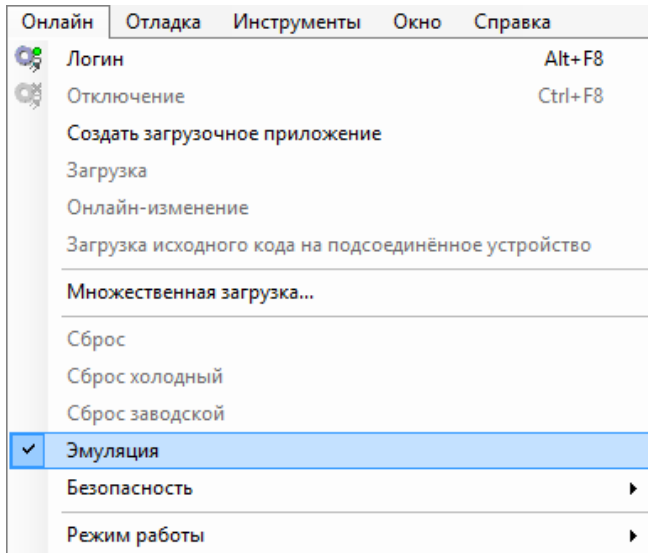


Рис. 81. Включение режима эмуляции

2. Выполнить команду **Логин** (Alt+F8).
3. Выполнить команду **Старт** (F5) (рис. 82)

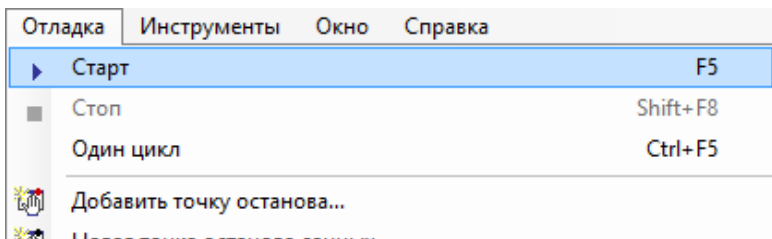


Рис. 82. Команда Старт

4. Открыть созданный ранее экран и проверить работу программы (рис. 83).

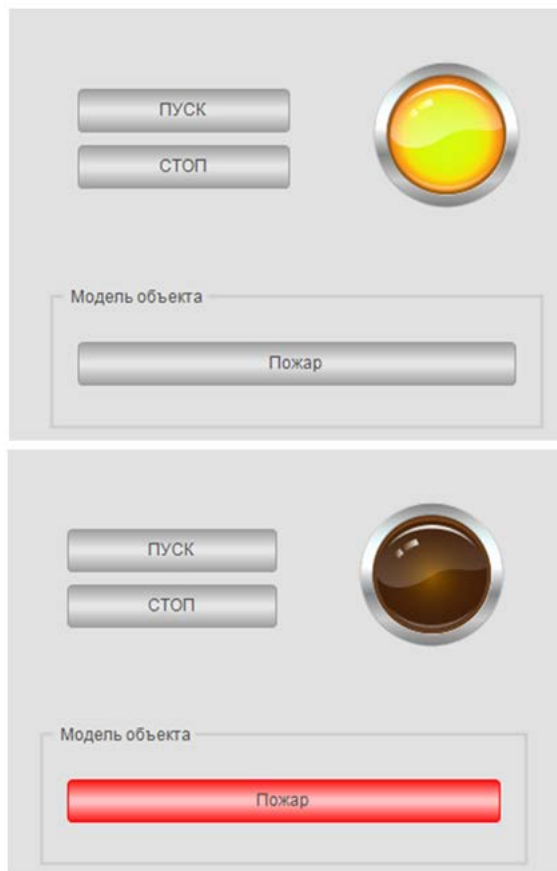


Рис. 83. Проверка работы программы

Контрольные вопросы

1. Как включить режим Эмуляция?
2. Какие «горячие клавиши» выполняют команду Логин?
3. Какую команду выполняют «горячие клавиши» F5?

Как изменить значение переменной в режиме Эмуляции без использования экранов?

7. СОЗДАНИЕ СОБСТВЕННЫХ БЛОКОВ

Для создания собственного блока на языке ST необходимо добавить в проект новую программу (рис. 84).

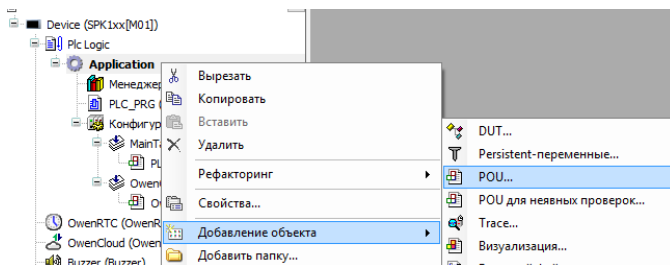


Рис. 84. Добавление в проект новой программы

После этого следует указать настройки, представленные на рис. 85.

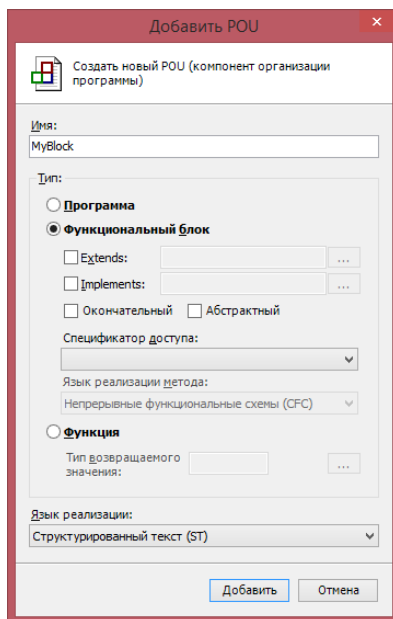


Рис. 85. Настройки для создания собственного блока

7.1. Постановка задачи

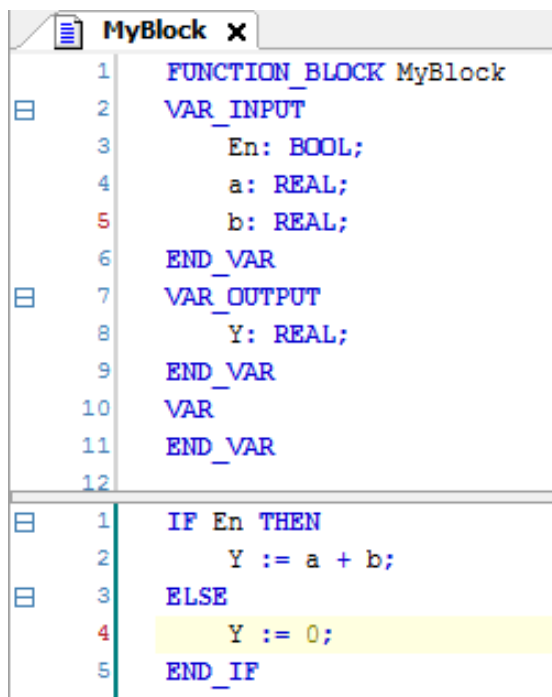
Необходимо написать собственный блок, реализующий следующий алгоритм:

- 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**.

7.2. Пример программы на ST

Программа на языке **ST**, реализующая описанное ранее задание, представлена на рис. 86.



```
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
13 IF En THEN
14     Y := a + b;
15 ELSE
16     Y := 0;
17 END_IF
```

Рис. 86. Программа для создаваемого блока

7.3. 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. 87.

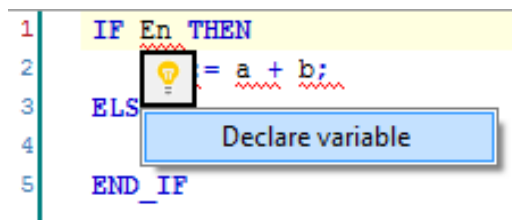


Рис. 87. Действие, выполняемое для автоматического объявления переменных

Стоит обратить внимание, что блоки, кроме переменных, имеют входы и выходы (**VAR_INPUT** и **VAR_OUTPUT**) (рис. 88).

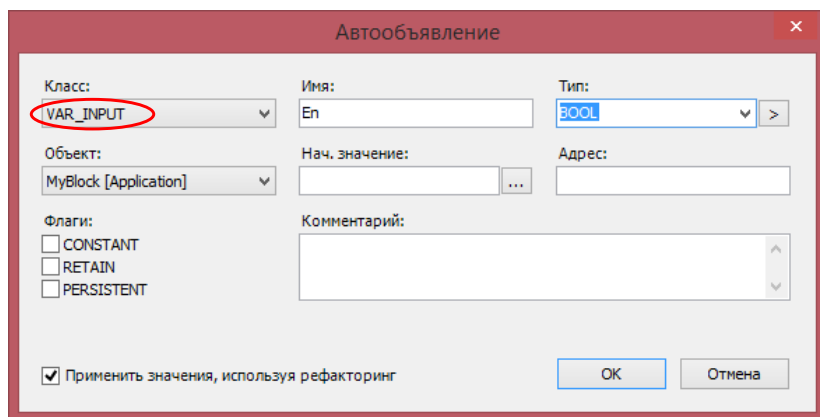


Рис. 88. Задание входов и выходов блоков

7.4. Использование созданного блока

Теперь можно использовать созданный блок в главной программе, например так, как показано на рис. 89.

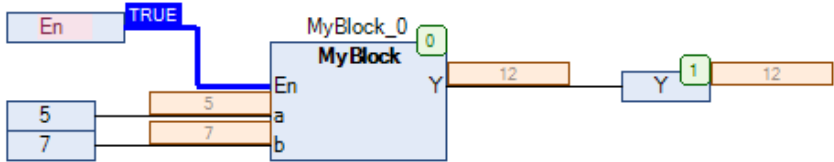


Рис. 89. Использование созданного блока

!!! Рисунок сделан при запущенной Эмуляции.

7.5. Создание блока Аperiodического звена 1-го порядка

Математической моделью объекта регулирования температуры (а также концентрации и многих других) является аperiodическое звено 1-го порядка (или больших порядков) с запаздыванием или без него. Наиболее подробная информация о динамических звеньях имеется в [11].

Простейшим вариантом будет следующий блок, реализованный на языке ST (рис. 90).

```
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. 90. Аperiodическое звено 1-го порядка на языке ST

Данный блок в CFC будет выглядеть, как показано на рис. 91.

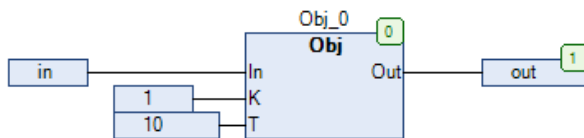


Рис. 91. Внешний вид созданного блока

При подаче на вход (In) блока единичного ступенчатого воздействия будет получен график переходного процесса, представленный на рис. 92.

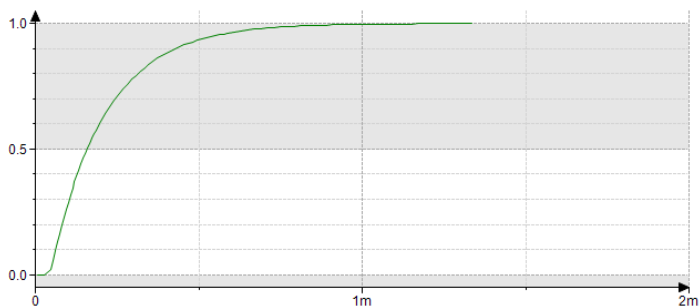


Рис. 92. График Аперидического звена 1-го порядка

Стоит отметить, что взятая на рис. 89 константа 0,02 зависит от интервала, выбранного в MainTask (рис. 93).

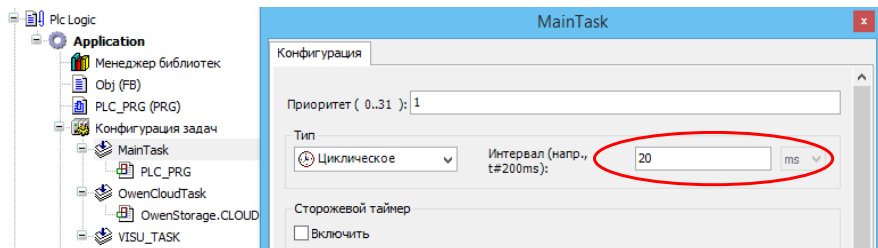


Рис. 93. Интервал в MainTask

7.6. Создание блока Аперiodического звена 2-го порядка

Для получения звена 2-го порядка достаточно соединить последовательно (рис. 94) два звена 1-го порядка, полученных ранее.

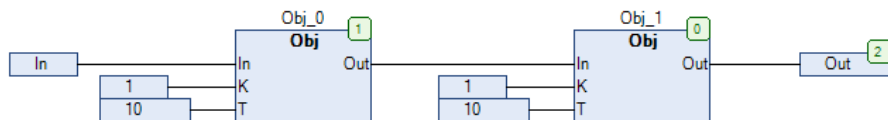


Рис. 94. Создание Аперiodического звена 2-го порядка

!!! Стоит обратить внимание, что имена Obj_0 и Obj_1 должны быть различными.

График данного Аперiodического звена 2-го порядка представлен на рис. 95.

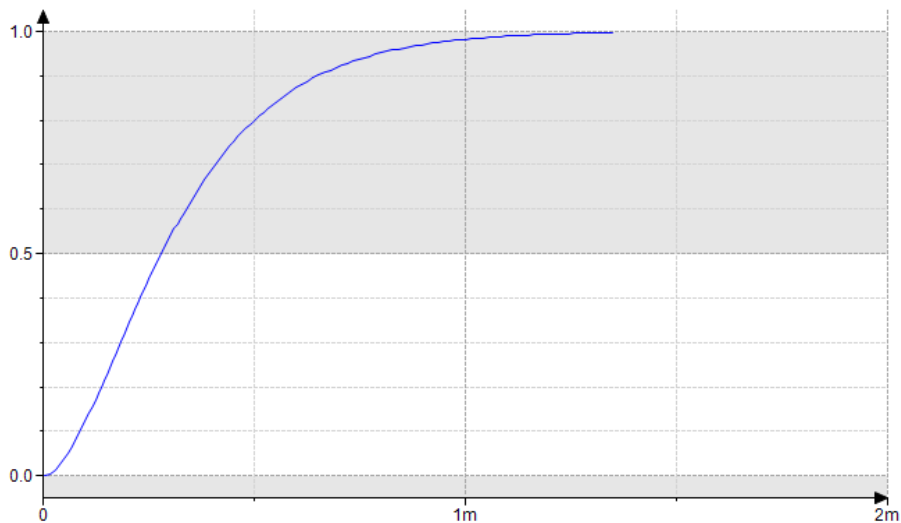


Рис. 95. График Аперiodического звена 2-го порядка

7.7. Создание блока Интегрирующего звена

Математической моделью объекта регулирования уровня является интегрирующее звено с запаздыванием или без него.

Реализуем данный блок на языке ST (рис. 96).

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

Рис. 96. Интегрирующее звено на языке ST

Стоит отметить, что взятая на рис. 96 константа 0,02 зависит от интервала, выбранного в MainTask (рис. 93 п. 7.5).

Данный блок в CFC будет выглядеть, как показано на рис. 97.

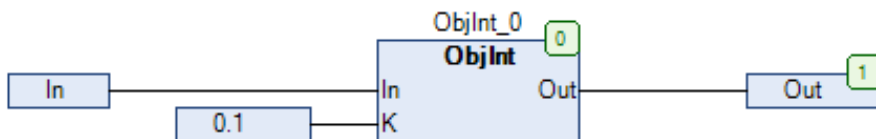


Рис. 97. Внешний вид созданного блока

При подаче на вход (In) блока единичного ступенчатого воздействия будет получен график переходного процесса, представленный на рис. 98.

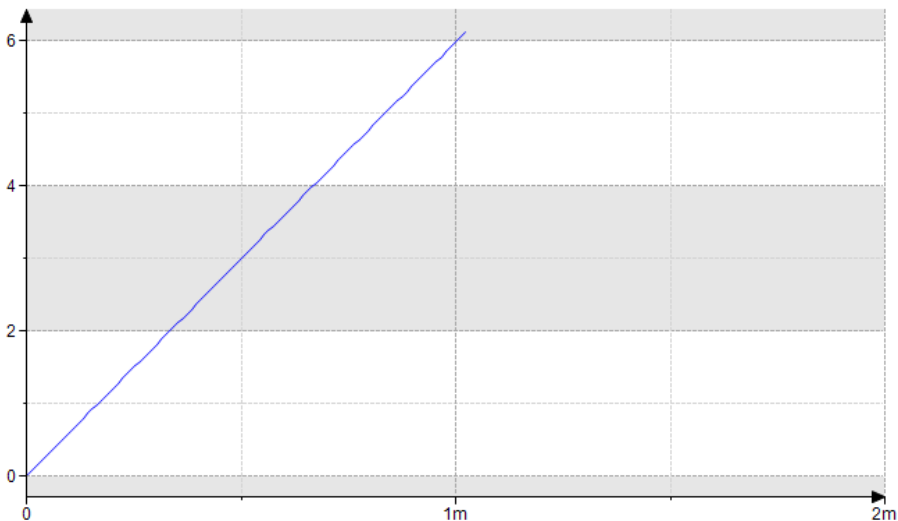


Рис. 98. График Интегрирующего звена

7.8. Создание блока Транспортного запаздывания

Кроме систем без запаздывания (например, как на рис. 43 п. 4.7), 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. 99.

Пример использования блока в главной программе на языке CFC показан на рис. 100. Данный пример реализует задержку на 10 секунд, что проиллюстрировано на графике на рис. 101.

Контрольные вопросы

1. Как создать собственный блок в CoDeSys?
2. Почему переменные в программе на рис. 89 имеют такие типы данных?
3. Как включить автоматическое объявление переменных?
4. Почему при создании блока нужно указать класс переменной (вход или выход)?
5. Как строится блок аperiodического звена второго порядка? Почему? Свой ответ обоснуйте, используя знания о динамических звеньях.
6. Что следует учесть при написании кода ST при создании блока интегрирующего звена?

```

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
13 n := TRUNC(t*50);
14
15 Out := A[n];
16 FOR i := 0 TO n-1 DO
17     A[n-i] := A[n-i-1];
18 END_FOR
19 A[0] := In;

```

Рис. 99. Блок транспортного запаздывания

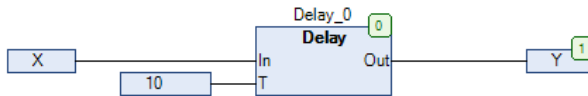


Рис. 100. Использование блок транспортного запаздывания

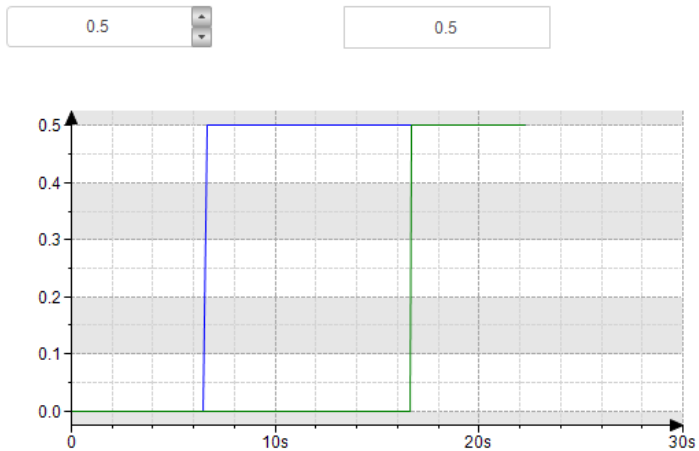


Рис. 101. Задержка сигнала на 10 секунд

8. ЛАБОРАТОРНЫЕ РАБОТЫ

8.1. 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.

8.2. Лабораторная работа 1

Создать экран оператора, визуализирующий четыре основные логические операции и логическое выражение в соответствии с вариантом табл. 6.

Таблица 6 – Варианты

№ п/п	Логическое выражение
1	$\overline{A + B \cdot (\overline{C} \oplus A)}$
2	$(\overline{A \cdot \overline{B}} \oplus C) + A$
3	$(\overline{C} \oplus \overline{A}) + B \cdot C$
4	$A \cdot B \cdot (C \oplus \overline{B}) + A$
5	$\overline{A \oplus B \oplus C} + A \cdot C$
6	$\overline{B} + A \cdot (C \oplus \overline{A})$
7	$C \cdot \overline{A} + \overline{B} \oplus \overline{C}$
8	$\overline{A + B \cdot (C \oplus B)} \cdot \overline{A}$
9	$B \oplus (A \cdot C + \overline{A})$
10	$\overline{A \oplus C \oplus \overline{B}} + B \cdot C$
11	$(\overline{B} \cdot A + B) \oplus \overline{C}$
12	$C + A \cdot \overline{B} \oplus \overline{A}$
13	$A \oplus (\overline{B} + \overline{A} \cdot C)$
14	$\overline{A \cdot B} + (\overline{A} \oplus C)$
15	$B \cdot (A \cdot C) \oplus \overline{B} + C$
16	$C + (B \oplus \overline{A} \cdot B \cdot \overline{C})$
17	$(A + B + C) \oplus (A \cdot \overline{B})$
18	$A + (\overline{B \cdot C} \oplus \overline{A})$
19	$(C \oplus A) \cdot \overline{B} + \overline{B}$
20	$\overline{B \cdot \overline{A} + C} \oplus C$

Необходимо использовать элементы управления с различным дизайном. Создаваемый экран оператора должен содержать заголовок, ФИО и группу студента, эмблему ВШТЭ и анимированный логотип CoDeSys.

Группы элементов необходимо подписывать.

Шрифт должен быть не меньше 14 кегля.

Пример экрана оператора в режиме эмуляции представлен на рис. 102.

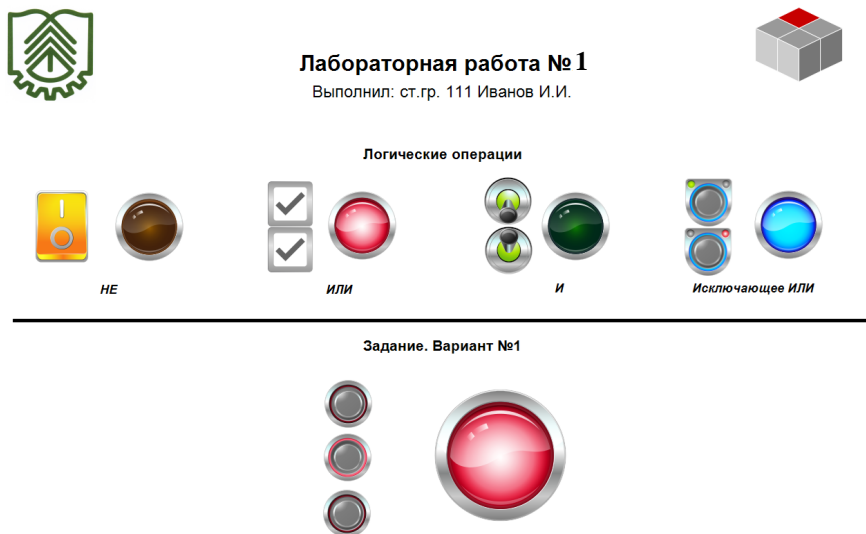


Рис. 102. Пример экрана оператора лабораторной работы 1

Пояснения по выполнению лабораторной работы:

- создать проект, как это описано в главе 1;
- добавить в программу блок **or**, на входы блока **or** подключить переменные **a** и **b**, на выход переменную **y** (см. главу 3 и рис. 18);
- на экране разместить индикатор и два переключателя из раздела «Lamps / Switches / Bitmaps» (см. рис. 53);
- в свойствах индикатора и переключателей привязать (см. рис. 56) переменные **a**, **b** и **y**;
- запустить эмуляцию (см. главу 6) и проверить работу.

8.3. Лабораторная работа 2

Повторить программу на языке СFC и Экран, описанные в примерах (в п. 3.1 – 3.2 и п. 5.1 – 5.2), а также проверить работу Эмуляции (глава 6).

Добавить модули дискретного ввода / вывода (DI и DO из главы 2), подключить их к переменным программы. Организовать запоминание аварии Fire и кнопку сброса аварий на экране. Дополнить программу аварией замерзания. Добавить в систему вентиляции заслонку (вентилятор запускается через 5 секунд после начала открытия заслонки, заслонка закрывается через 3 секунды после остановки вентилятора).

Реализовать систему регулирования температуры воздуха (см. п. 4.6, 4.7, 7.5) и создать отдельный экран с графиками текущей и заданной температур (см. п. 5.4). При отключении вентилятора, система регулирования температуры также должна отключаться. Для переходов между экранами, необходимо добавить соответствующие кнопки.

При реализации программы, необходимо руководствоваться расположением элементов, представленным на рис. 22.

Создаваемый экран оператора должен содержать заголовок, ФИО и группу студента, эмблему ВШТЭ и прочие элементы оформления, как в предыдущей лабораторной работе.

8.4. Лабораторная работа 3

Разработать систему регулирования уровня в баке. Пример экранов системы показан на рис. 103. Передаточная функция объекта регулирования определяется в соответствии с вариантом (табл. 7).

!!! Разрабатываемая система не является реальной.

Все клапаны должны иметь ручное управление. Система должна иметь блокировки и сигнализации по предельным значениям уровня.

Создаваемый экран оператора должен содержать заголовок, ФИО и группу студента, эмблему ВШТЭ и прочие элементы оформления, как в предыдущих лабораторных работах.



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

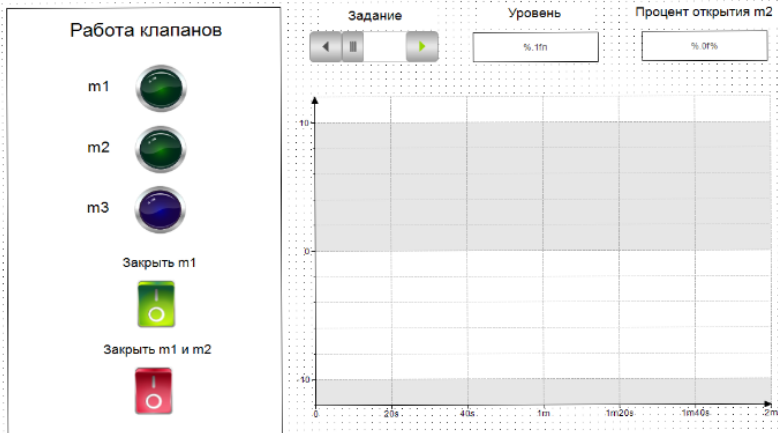


Главный экран



График

Экран График



На Главный

Рис. 103. Пример экранов оператора лабораторной работы 3

Таблица 7 – 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

9. КУРСОВАЯ РАБОТА (ПРОЕКТ)

9.1. 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 ее численными характеристиками.
7. Заключение / выводы.
8. 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 об их функции.

9.2. 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. 104).

Технологическая схема необходима прежде всего для разработки экрана оператора (необходимо перерисовать данную схему в CoDeSys на Главном экране проекта).

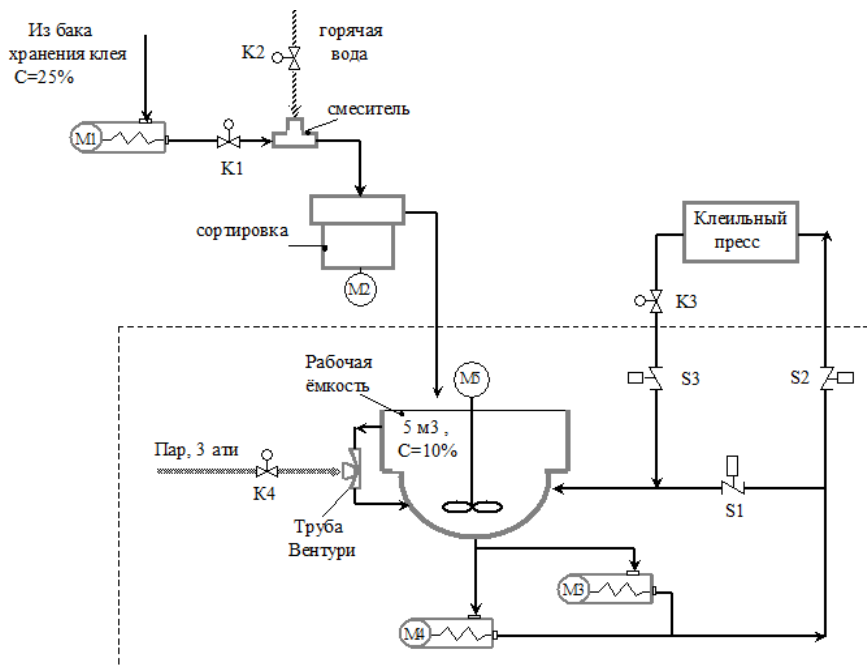


Рис. 104. Пример технологической схемы

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ОВЕН: СПК110 контроллер с сенсорным экраном 10.2 для локальных систем [Электронный ресурс]. – URL: <https://owen.ru/product/spk110> (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. ОВЕН: Модули аналогового ввода с универсальными входами (с интерфейсом RS-485) MB110 [Электронный ресурс]. – URL: https://owen.ru/product/moduli_analogovogo_vvoda_s_universal_nimi_vhodami_s_interfejsom_rs_485 (дата обращения: 20.01.2022).
4. ОВЕН: Модули дискретного ввода (с интерфейсом RS-485) MB110 [Электронный ресурс]. – URL: https://owen.ru/product/moduli_diskretnogo_vvoda_s_interfejsom_rs_485 (дата обращения: 20.01.2022).
5. ОВЕН: Модули аналогового вывода (с интерфейсом RS-485) MY110 [Электронный ресурс]. – URL: https://owen.ru/product/moduli_analogovogo_vivoda_s_interfejsom_rs_485 (дата обращения: 20.01.2022).
6. ОВЕН: Модули дискретного вывода (с интерфейсом RS-485) MY110 [Электронный ресурс]. – URL: https://owen.ru/product/moduli_diskretnogo_vivoda_s_interfejsom_rs_485 (дата обращения: 20.01.2022).
7. Использование модулей ввода вывода Mx110 / Видео из YouTube, 17:14, загружено «ОВЕН. Производственное Объединение», ноябрь 09, 2017, <https://www.youtube.com/watch?v=0m5vbe7z6Ek>.
8. 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>.
9. ОВЕН, Среда программирования Owen Logic. Руководство пользователя [Текст] – Версия 07. – М., 2011. – 47 с.
10. 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).
11. 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**

Автоматизация технологических процессов и производств

Практикум

Редактор и корректор А. А. Чернышева
Техн. редактор Д. А. Романова

Темплан 2021 г., поз. 5260

Подписано к печати 14.06.2022.	Формат 60x84/16.	Бумага тип № 1.
Печать офсетная.	Печ. л. 4,4.	Уч.-изд. л. 4,4.
Тираж 30 экз.	Изд. № 5260.	Цена «С».
		Заказ №

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