

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования  
«Витебский государственный технологический университет»

**ПРОЕКТИРОВАНИЕ СИСТЕМ АВТОМАТИЗАЦИИ.  
ПРОГРАММИРОВАНИЕ ПРИБОРНОЙ АВТОМАТИКИ**

**Лабораторный практикум  
для студентов специальности  
6-05-0713-04 «Автоматизация технологических процессов и производств  
(Компьютерная мехатроника)»**

Витебск  
2025

УДК 681.5

Составители:

А. М. Науменко, К. Н. Ринейский, А. М. Самусев, Д. А. Тёмкин

Одобрено кафедрой «Автоматизация производственных процессов»  
УО «ВГТУ», протокол № 11 от 15.05.2025.

Рекомендовано к изданию редакционно-издательским советом  
УО «ВГТУ», протокол № 9 от 28.05.2025.

**ПРОЕКТИРОВАНИЕ СИСТЕМ АВТОМАТИЗАЦИИ. ПРОГРАММИРОВАНИЕ ПРИБОРНОЙ АВТОМАТИКИ:** лабораторный практикум /  
сост. А. М. Науменко, К. Н. Ринейский, А. М. Самусев, Д. А. Тёмкин. –  
Витебск : УО «ВГТУ», 2025. – 77 с.

Лабораторный практикум является руководством к выполнению лабораторных работ по дисциплине «Проектирование систем автоматизации» для студентов специальности 6-05-0713-04 «Автоматизация технологических процессов и производств (Компьютерная мехатроника)», содержит перечень лабораторных работ, освещает теоретические вопросы подготовки к их выполнению, приводит примеры построения автоматизированных систем управления технологическим процессом на основе использования программируемого логического контроллера, сенсорных панелей оператора, программного обеспечения и средств автоматизации.

УДК 681.5

© УО «ВГТУ», 2025

## Содержание

Введение.....	4
1 Установка программного компонента.....	5
1.1 Установка программного компонента для ПЛК.....	5
1.1.1 Установка конфигуратора ПЛК «Owen Configurator».....	5
1.1.2 Установка среды программирования «CoDeSys v3.5».....	9
1.1.3 Установка необходимых таргет-файлов.....	12
1.2 Установка среды программирования сенсорных панелей оператора.....	16
1.2.1 Установка среды программирования «Конфигуратор СП300».....	16
2 Работа в среде программирования «CoDeSys v3.5».....	18
2.1 Знакомство со средой программирования CODESYS 3.5.....	18
2.2 Создание нового проекта. Принципы работы ПЛК. Цикл ПЛК.	
Переменные и типы данных CODESYS.....	20
2.3 Знакомство с языком CFC. Стандартные операторы CODESYS: логика, арифметика.....	25
2.4 Стандартные операторы CODESYS: операторы выбора и ограничения, сравнение. Работа с входами и выходами ПЛК210.....	34
2.5 Настройка связи между ПЛК и CODESYS. Методы отладки программы.	43
3 Работа в среде программирования «Конфигуратор СП300».....	49
3.1 Конфигурирование панели СП3xx.....	49
3.2 Изучение принципов работы элементов «Переключатель» и «Индикатор».....	58
3.3 Изучение принципов работы с численными элементами.....	63
3.4 Изучение принципов работы элементов визуального отображения численных значений.....	67
3.5 Изучение принципов работы панели оператора с контроллером.....	71
Литература.....	76

## **Введение**

Основная цель лабораторных работ – это развитие инженерных навыков по разработке систем автоматизации производственных процессов на основе современных технических средств контроля, управления на основе блочно-модульной автоматики и по созданию прикладного программного обеспечения на основе промышленных языков программирования стандарта МЭК 61131-3.

Программирование приборной автоматики имеет большое значение, так как является одной из важных составляющих в этапах разработки систем управления и формирования комплексных знаний инженера по автоматизации.

Полученные знания должны подготовить студента к выполнению дипломного проекта.

# 1 УСТАНОВКА ПРОГРАММНОГО КОМПОНЕНТА

## 1.1 Установка программного компонента для ПЛК

### 1.1.1 Установка конфигуратора ПЛК «Owen Configurator»

Дистрибутив «Owen Configurator» может быть загружен с сайта ОВЕН в разделе «Каталог продукции → Программное обеспечение, устройства связи → Конфигураторы». Дистрибутив распространяется в виде .zip-архива. Распакуйте архив в отдельную папку и запустите файл установщика с расширением .exe. В появившемся окне выберете язык установки (рис. 1.1).

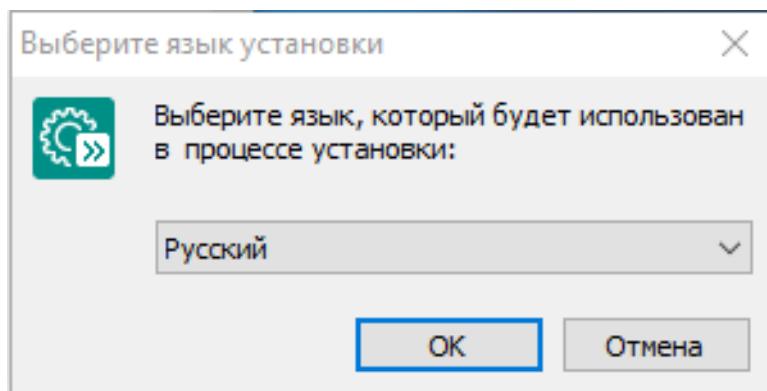


Рисунок 1.1 – Выбор языка установки «Owen Configurator»

В следующем окне (рис. 1.2) нажмите кнопку «Далее>».

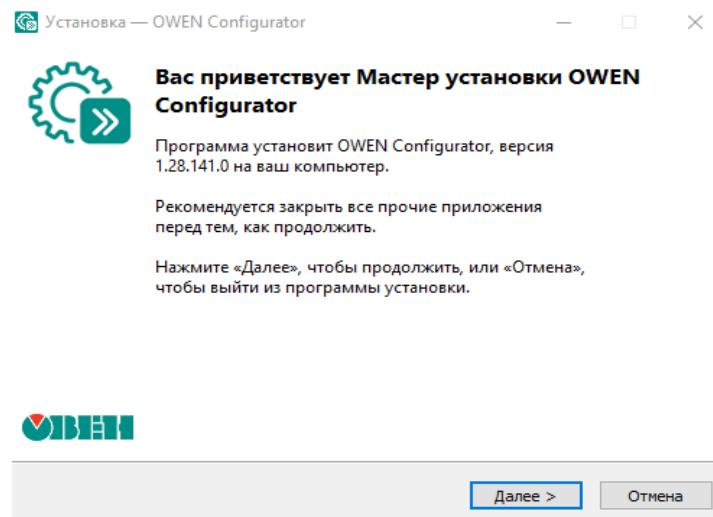


Рисунок 1.2 – Окно установки «Owen Configurator»

В следующем окне (рис. 1.3) ознакомьтесь с текстом лицензионного соглашения, выберите пункт «Я принимаю условия соглашения» и нажмите кнопку «Далее>».

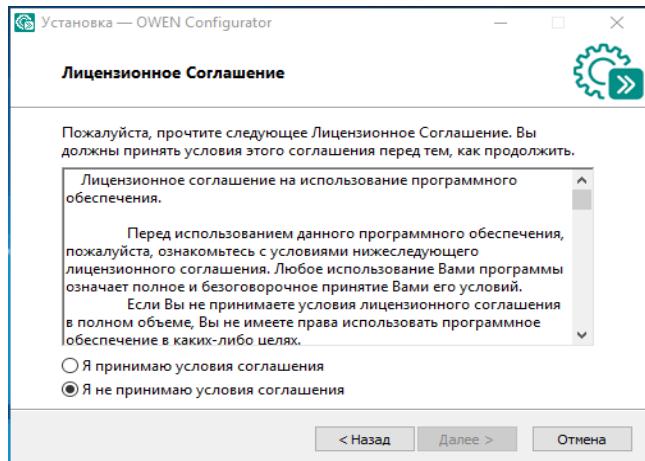


Рисунок 1.3 – Окно лицензионного соглашения

В следующем окне (рис. 1.4–1.5) выберите директорию, в которую будет установлен «Owen Configurator», и нажмите кнопку «Далее>».

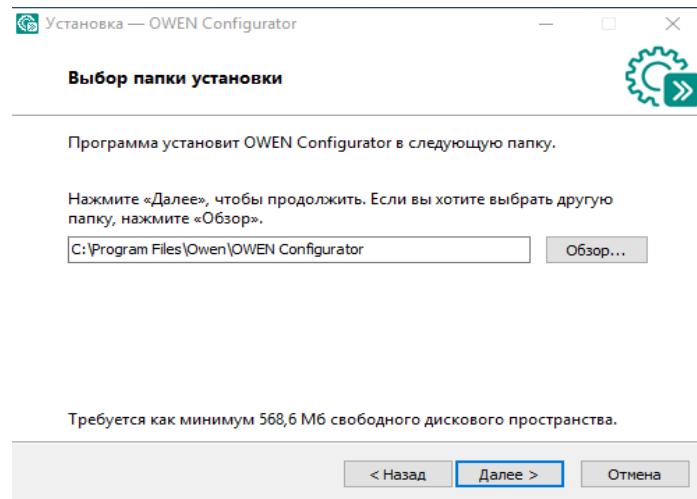


Рисунок 1.4 – Окно директории установки «Owen Configurator»

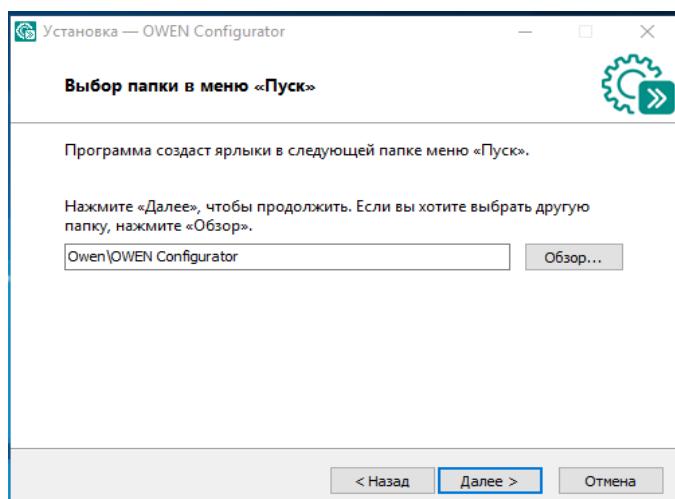


Рисунок 1.5 – Выбор папки в меню «Пуск»

После открытия следующего окна (рис. 1.6) выберете «Создать значок на Рабочем столе» и «Установить драйвер COM-порт для связи с устройствами ОВЕН» и нажмите на кнопку «Далее>».

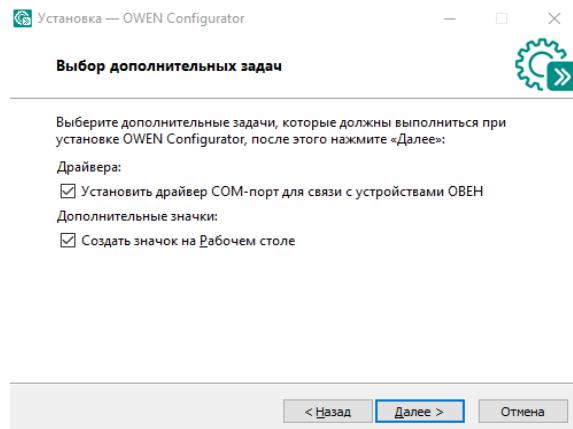


Рисунок 1.6 –Окно установки дополнительных задач «OWEN Configurator»

В предустановочном окне (рис. 1.7) нажмите кнопку «Установить».

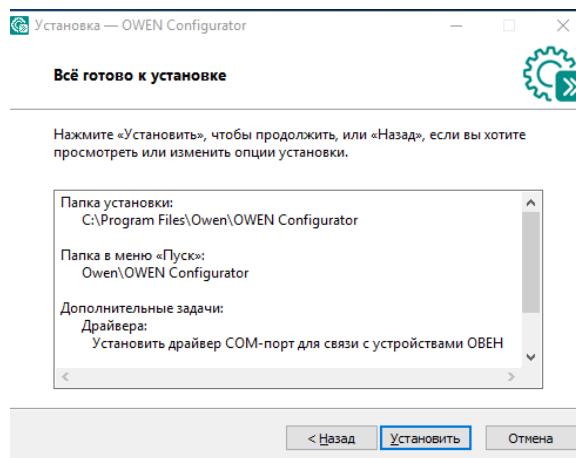


Рисунок 1.7 – Предустановочное окно

В процессе установки дождитесь распаковки файлов (рис. 1.8)

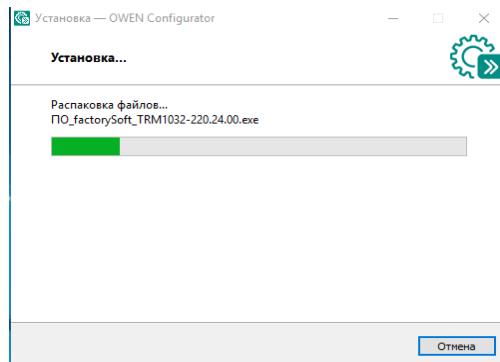


Рисунок 1.8 – Окно установки OWEN Configurator

Если вы выбрали «Установить драйвер COM-порт для связи с устройствами ОВЕН», тогда у вас откроется окно (рис. 1.9) установки «Мастер установки драйверов устройств» нажмите «Далее»».

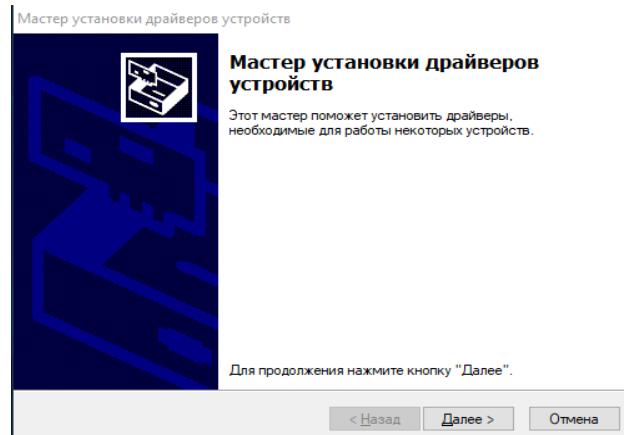


Рисунок 1.9 – Мастер установки драйверов устройств

В окне завершения установки драйверов устройств после установки драйвера устройства (рис. 1.10) нажмите кнопку «Готово».

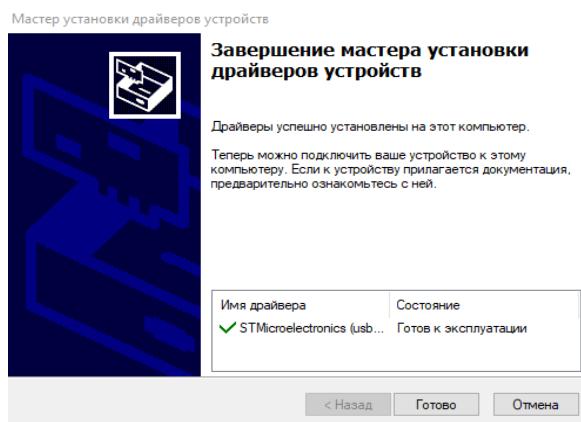


Рисунок 1.10 –Окно завершения установки драйверов устройств

После появиться окно установки «USB Virtual Com Port Driver» (рис. 1.11), в котором нажмите кнопку «Instal».

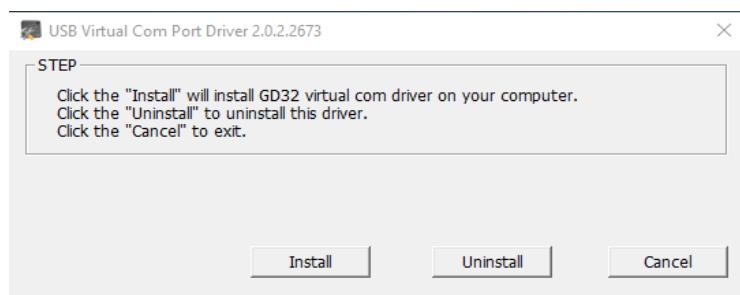


Рисунок 1.11 – Окно установки USB Virtual Com Port Driver

После завершения установки нажмите в окне (рис. 1.12) на кнопку «Завершить». OWEN Configurator успешно установлен.

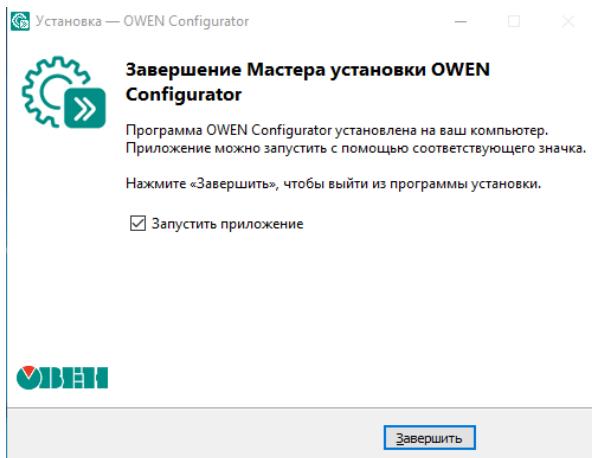


Рисунок 1.12 – Окно завершения установки OWEN Configurator

### 1.1.2 Установка среды программирования «CoDeSys v3.5»

Дистрибутив CODESYS может быть загружен с сайта ОВЕН в разделе «CODESYS V3 → Среда программирования». Дистрибутив распространяется в виде «.zip-архива». Распакуйте архив в отдельную папку и запустите файл установщика с расширением «.exe».

**ВАЖНО! УСТАНОВКА CODESYS ТРЕБУЕТ ПОДКЛЮЧЕНИЯ К ИНТЕРНЕТУ, ПОСКОЛЬКУ В ПРОЦЕССЕ УСТАНОВКИ ЗАГРУЖАЕТСЯ ДОПОЛНИТЕЛЬНОЕ ПО, НЕОБХОДИМОЕ ДЛЯ РАБОТЫ СРЕДЫ. НА ВРЕМЯ УСТАНОВКИ CODESYS ОТКЛЮЧИТЕ ВАШ АНТИВИРУС.**

В появившемся окне нажмите кнопку Install для установки утилиты «CODESYS Installer» (рис. 1.13)

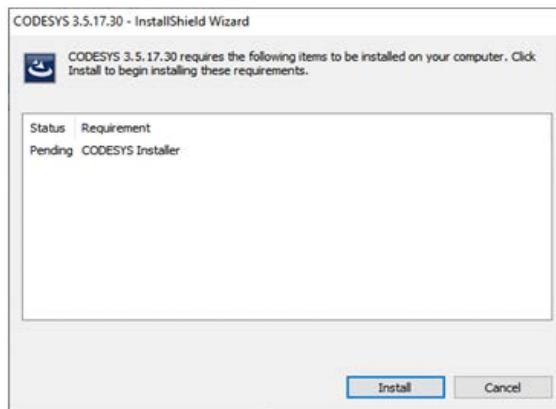


Рисунок 1.13 – Окно установки утилиты CODESYS Installer

В стартовом окне установщика нажмите кнопку «Next>» (рис. 1.14).



Рисунок 1.14 – Стартовое окно установщика

В окне лицензионного соглашения (рис.1.15) ознакомьтесь с текстом лицензионного соглашения, выберите пункт «*I accept the terms in the license agreement*» и нажмите кнопку «*Next>*».



Рисунок 1.15 – Окно лицензионного соглашения

В окне с информацией об основных изменениях в устанавливаемой версии среды (рис. 1.16) ознакомьтесь с информацией об основных изменениях, произошедших в устанавливаемой версии CODESYS, выберите пункт «*I have read the information*» и нажмите кнопку «*Next>*».

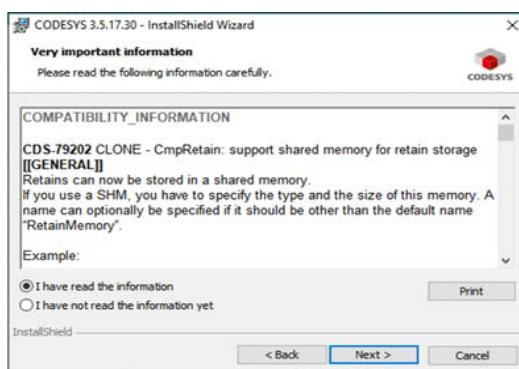


Рисунок 1.16 – Окно с информацией об основных изменениях в устанавливаемой версии среды

В окне выбора директории установки (рис. 1.17) выберите директорию, в которую будет установлен CODESYS, и нажмите кнопку «*Next >*». Каждую версию среды следует устанавливать в отдельную директорию.

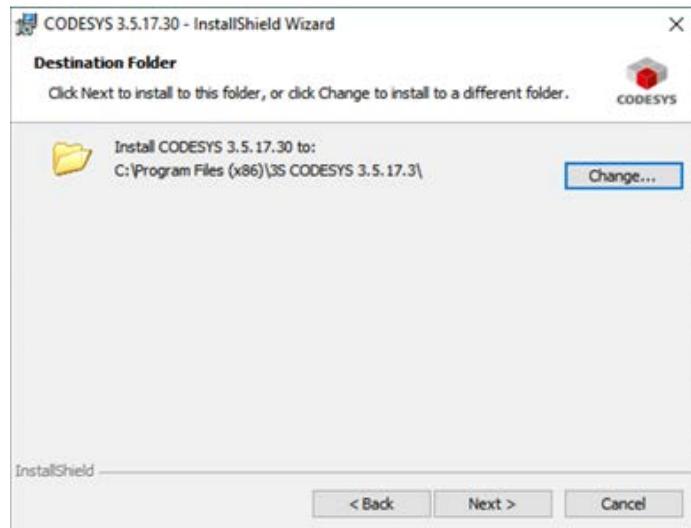


Рисунок 1.17 – Окно выбора директории установки

В окне выбора режима установки CODESYS (рис. 1.18) необходимо выбрать режим установки CODESYS. **Обязательно** выберите режим полной установки (*Complete*), чтобы установить все доступные плагины, и нажмите «*Next >*».

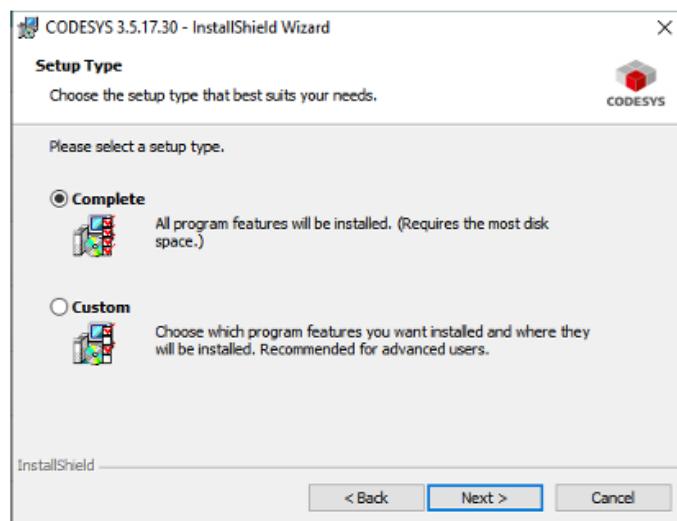


Рисунок 1.18 – Окно выбора режима установки CODESYS

В окне запуска установки CODESYS (рис. 1.19) нажмите кнопку «*Install*» для запуска процесса установки.

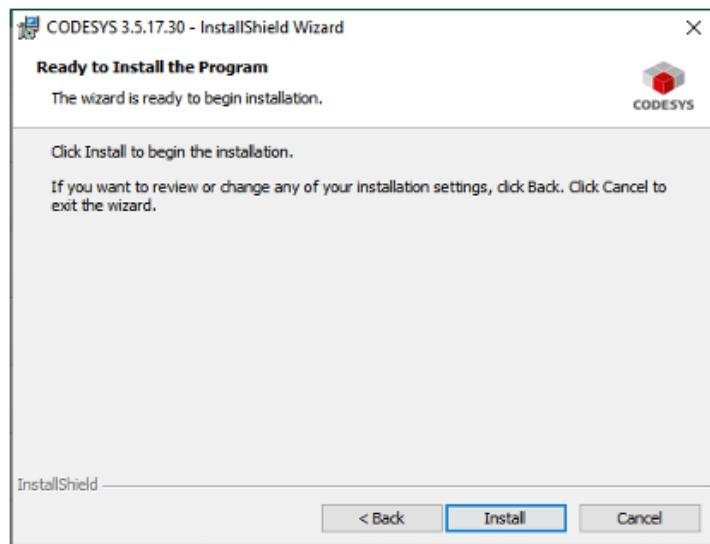


Рисунок 1.19 – Окно запуска установки CODESYS

Процесс установки занимает от 10 минут до часа. После завершения установки появится окно завершения установки CODESYS (рис. 1.20). Нажмите кнопку «*Finish*». Процесс установки CODESYS завершен.

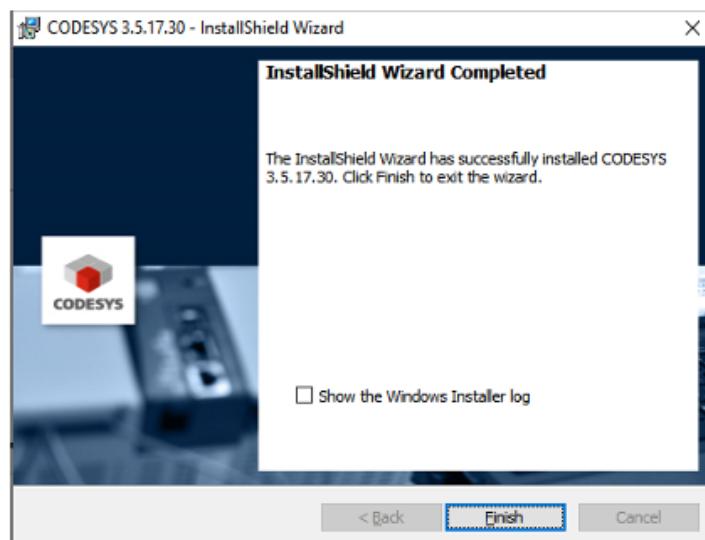


Рисунок 1.20 – Окно завершения установки CODESYS

### 1.1.3 Установка необходимых таргет-файлов

Для установки таргет-файлов скачайте с сайта [owen.ru](http://owen.ru) в разделе «*Сервисное ПО*». Выберете «*Пакет таргет-файлов*» в соответствии с вашим контроллером и датой его выпуска (рис. 1.21).

Сервисное ПО					
Контроллер	Пакет таргет-файллов*	Прошивка	Инструкция	Драйвер USB	История версий
СПК210	3.5.17.33_содержание	3.6.0918.1402			
	3.5.17.31_содержание	2.4.0923.1000 * хелфекс			<a href="#">Перейти</a>
СПК1xx [M01]	3.5.16.32_содержание	1.3.0928.2149			
	3.5.14.30-10_содержание	1.2.0903.1220			
ПЛК210-1x	3.5.17.34_содержание	3.6.0906.1300			
	3.5.17.32_содержание	3.5.1023.2023			
ПЛК210-0x	3.5.17.31_содержание	2.4.0901.0905 * хелфекс			
	3.5.15.32_содержание	1.3.0912.1315			
ПЛК200	3.5.14.30-10_содержание	1.3.0909.1297			
	3.5.17.31_содержание	2.4.2110.1400			
	3.5.16.32_содержание	1.3.1005.2220			<a href="#">Перейти</a>
	3.5.14.30-10_содержание	1.3.0419.1204			

Рисунок 1.21 – Выбор сервисного ПО

После скачивания файла с установщиком откройте CODESYS. Перейдите во вкладку «Инструменты → Менеджер пакетов» (рис. 1.22).

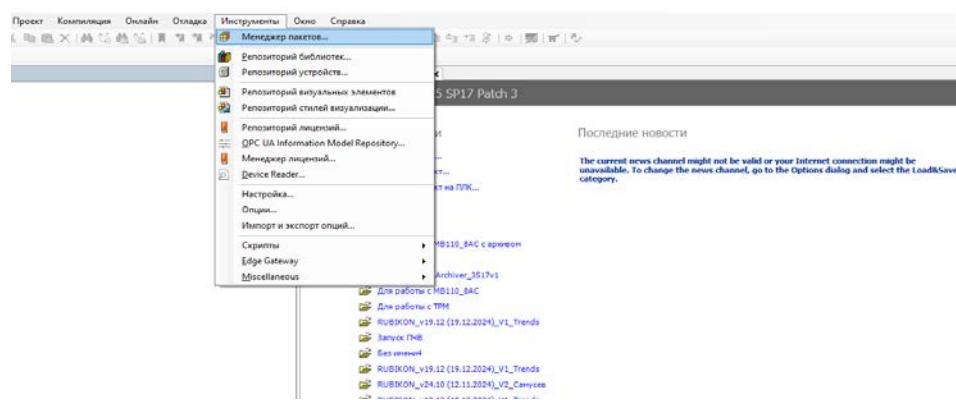


Рисунок 1.22 – Открытие «Менеджера пакетов»

После открытия «Менеджера пакетов» нажмите на кнопку «Установить» (рис. 1.23).

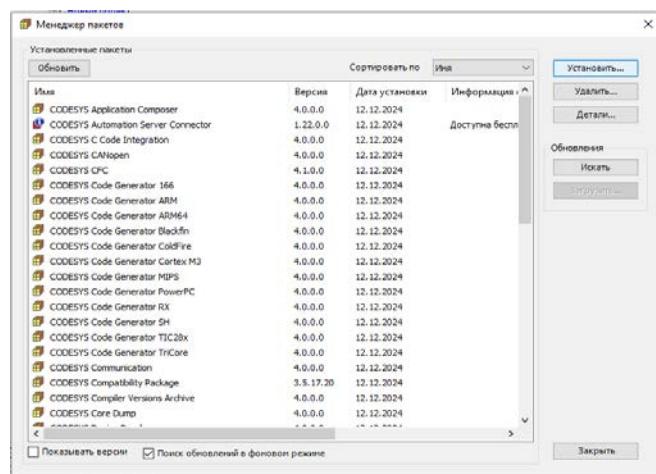


Рисунок 1.23 – Окно «Менеджер пакетов»

Затем откройте файл «owentargets-3.5.17.31.package» (рис. 1.24).

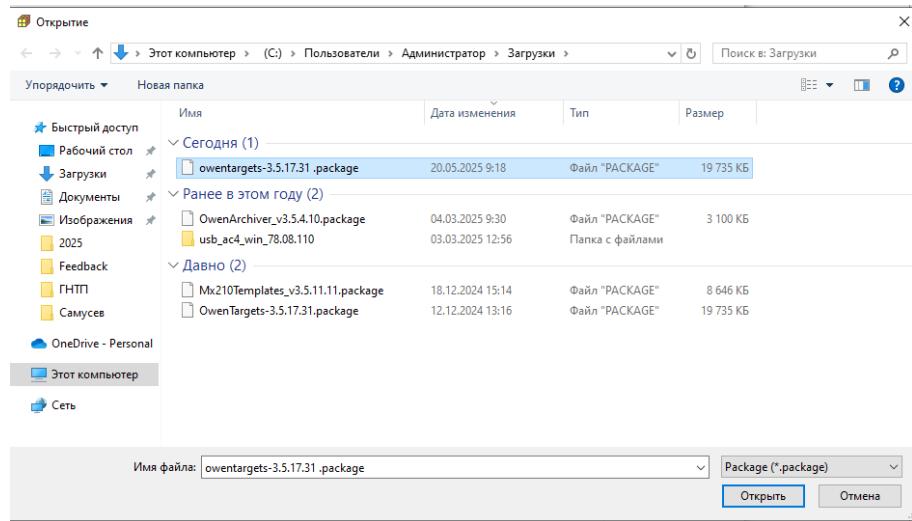


Рисунок 1.24 – Открытие таргет-файла

После этого у Вас появится окно «Проверить подписи пакетов», поставьте галочку на пункте «Не подписанные и самоподписанные пакеты» и нажмите кнопку «OK» (рис. 1.25).

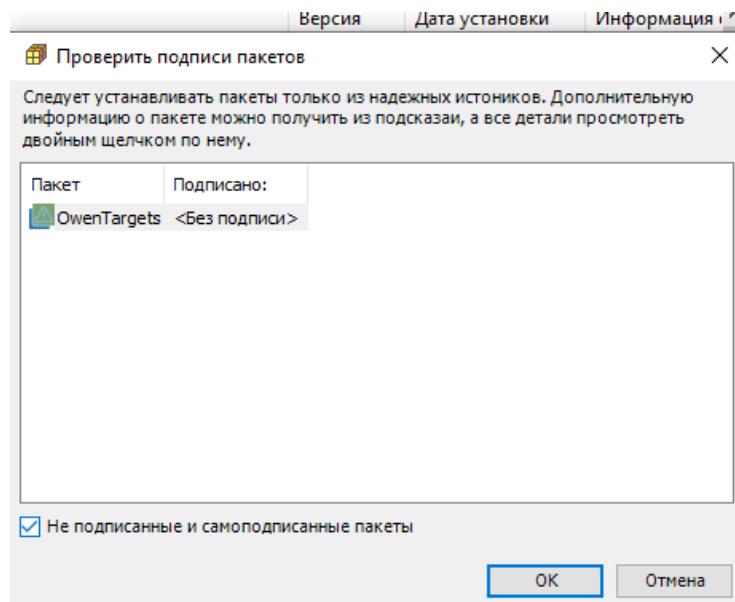


Рисунок 1.25 – Окно «Проверить подписи пакетов»

Затем появиться окно «Установка – Choose Setup Type» выберите пункт «Полная установка» и нажмите «Next >» (рис. 1.26).

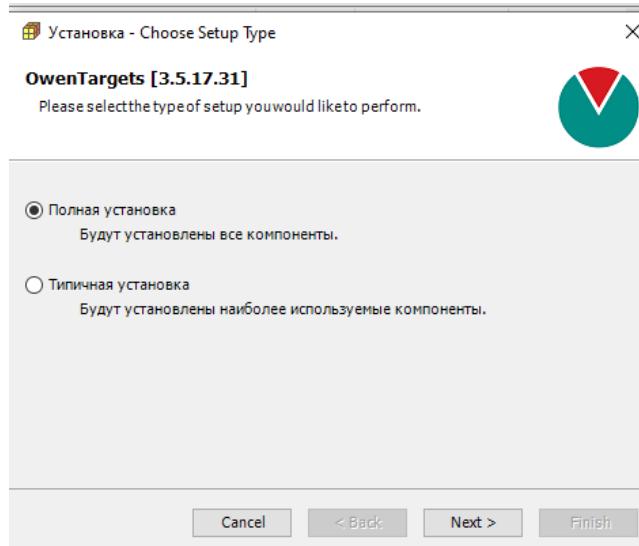


Рисунок 1.26 – «Установка – Choose Setup Type»

В процессе установки таргет-файла у Вас появиться окно «Установка – Шрифты для контроллеров ОВЕН СПКxxx» нажмите кнопку «Установить» (рис. 1.27). После появиться информационное окно о завершении установки шрифтов, где необходимо нажать кнопку «Завершить».

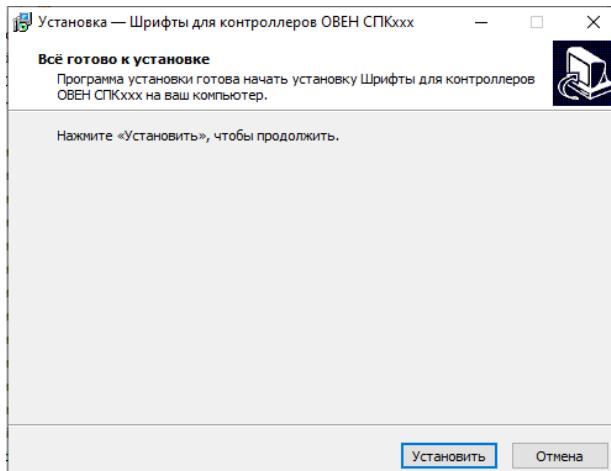


Рисунок 1.27 – Окно «Установка – Шрифты для контроллеров ОВЕН СПКxxx»

После завершения установки таргет-файлов появиться окно «Установка – Setup Completed», нажмите кнопку «Finish» (рис. 1.28). Процесс установки завершен.

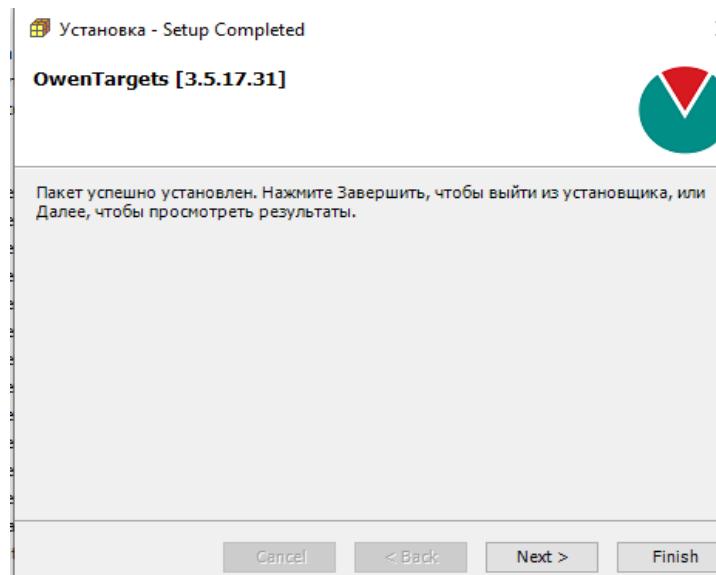


Рисунок 1.28 – Окно «Установка – Setup Completed»

## 1.2 Установка среды программирования сенсорных панелей оператора

### 1.2.1 Установка среды программирования «Конфигуратор СП300»

Для установки таргет-файлов скачайте с сайта [owen.ru](http://owen.ru). Выберете вкладку «*ПО, примеры и документация*», затем выберите «*Конфигуратор СП300 V2.D3k-5*» и нажмите «Скачать» (рис. 1.29), в появившемся окне нажмите «*Я принимаю условия лицензионного соглашения*» и на кнопку «Скачать».

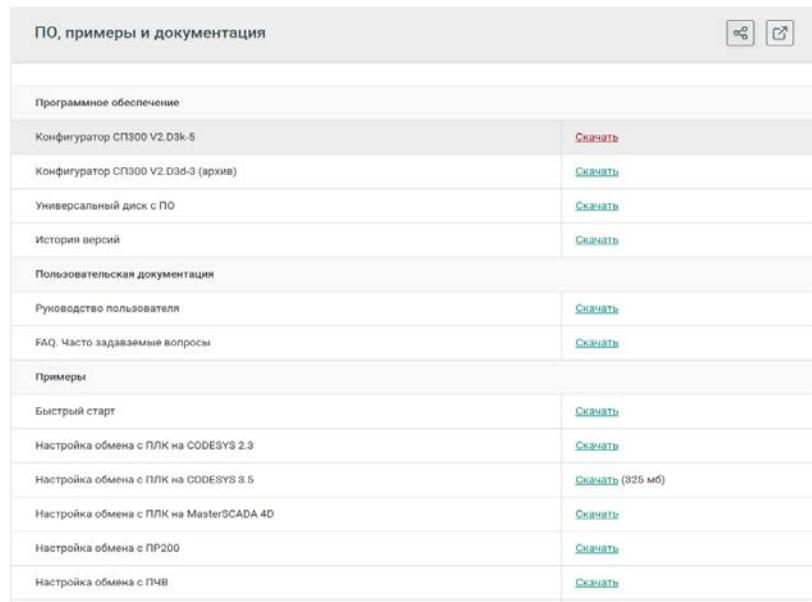


Рисунок 1.29 – Выбор программного обеспечения – Конфигуратор СП300 V2.D3k-5

После скачивания файла «*SP300\_project\_V2.D3k-5.zip*» разархивируйте его и запустите. Перед Вами появится окно «*Установка – КОНФИГУРАТОР СП300*» выберите пункт «*Я принимаю условия соглашения*» и нажмите кнопку «*Далее >*» (рис. 1.30).

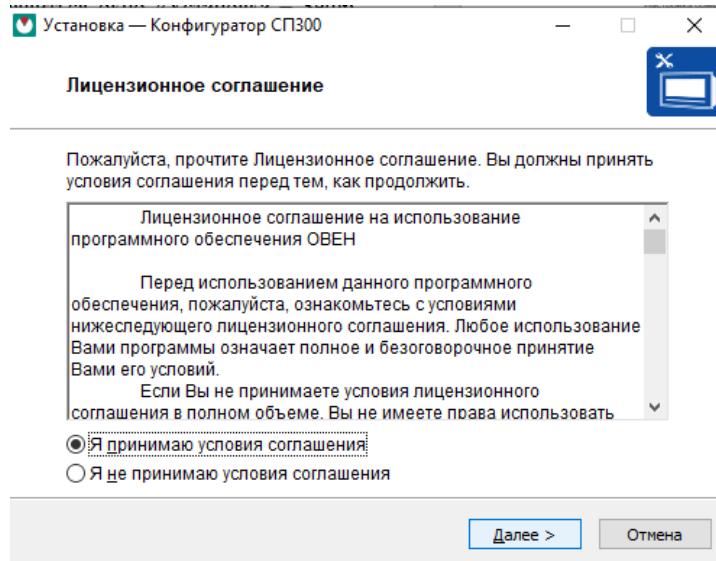


Рисунок 1.30 – Окно «Установка – Конфигуратор СП300»

Затем выберите папку установки и нажмите кнопку «*Далее*», в следующем окне выберите дополнительные задачи, которые должны выполниться при установке Конфигуратор СП300 и нажмите на кнопку «*Установить*». После завершения установки появиться окно о завершении (рис. 1.31).

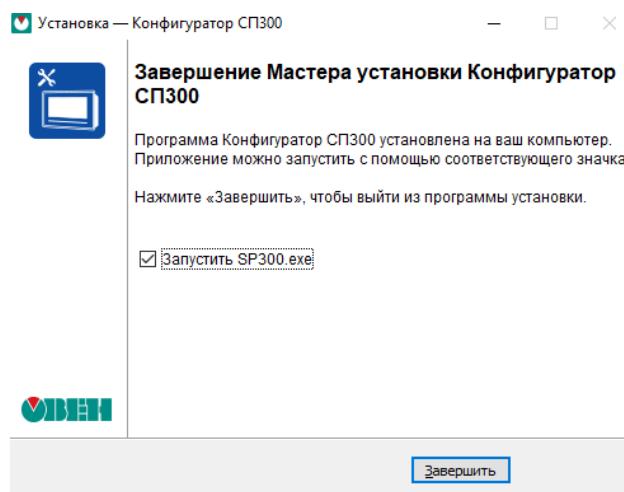


Рисунок 1.31 – Окно завершения установки СП300

## 2 РАБОТА В СРЕДЕ ПРОГРАММИРОВАНИЯ «CODESYS V3.5»

### 2.1 Знакомство со средой программирования CODESYS 3.5

*Цель работы:* ознакомиться со средой программирования CODESYS 3.5

#### Теоретическая часть

CODESYS (Controller Development System) – продукт германской компании 3S-softwear, которая является ведущей системой программирования ПЛК в мире. Среда разработана в соответствии с международным стандартом МЭК (IEC) 61131-3. Стандарт IEC 61131-3 (МЭК 61131-3) – раздел международного стандарта IEC 61131, описывающий языки программирования для программируемых логических контроллеров. IEC 61131-3 – первый независимый от производителя стандартизованный язык программирования для промышленной автоматизации, который предусматривает 2 типа языков программирования: графические и текстовые. К графическим языкам программирования относятся: язык релейных схем – Ladder Diagram (LD), язык функциональных блоковых диаграмм – Function Block Diagram (FBD), язык диаграмм состояний – Sequential Function Chart (SFC), язык непрерывных функциональных схем – Continuous Function Chart (CFC); к текстовым языкам относятся: список инструкций – Instruction List (IL), структурированный текст (Pascal-подобный язык) – Structured Text (ST). Для пользователей программный продукт CODESYS предоставляется бесплатно, его можно скачать с сайта [owen.ru](http://owen.ru).

Для программирования программируемого логического контроллера (далее ПЛК) 2xx серии фирмы «ОВЕН» необходимо использовать среду разработки CODESYS 3.5 Для ПЛК 210-04 CS необходимо использовать среду разработки CODESYS V3.5 SP17 Patch3 . При открытии среды перед пользователем открывается стартовое окно, где отображаются: основные операции, при помощи которых можно создать новый проект, открыть проект и открыть проект на ПЛК; недавние проекты, где отображаются ранее открытые проекты; последние новости (рис. 2.1).

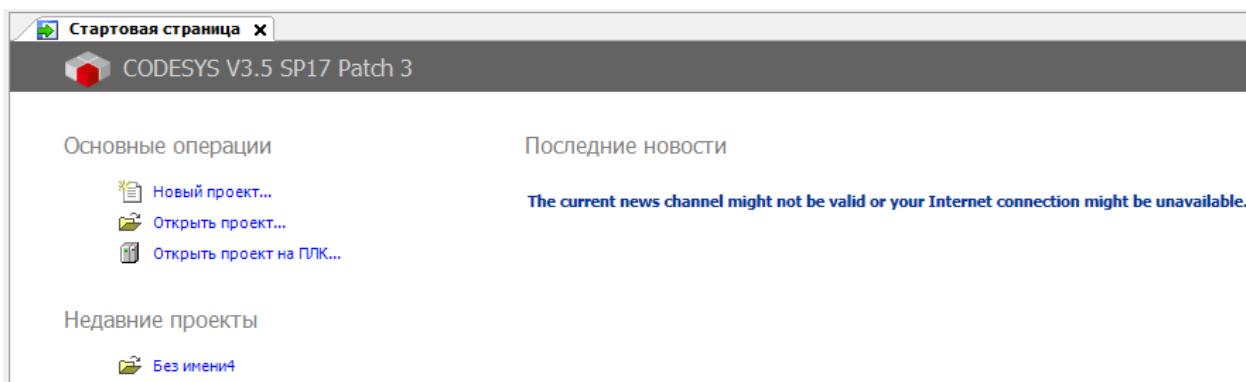


Рисунок 2.1 – Стартовая страница

При создании проекта пользователю подвластны инструменты программирования, инструменты отладки и инструменты создания визуализации (рис. 2.2).

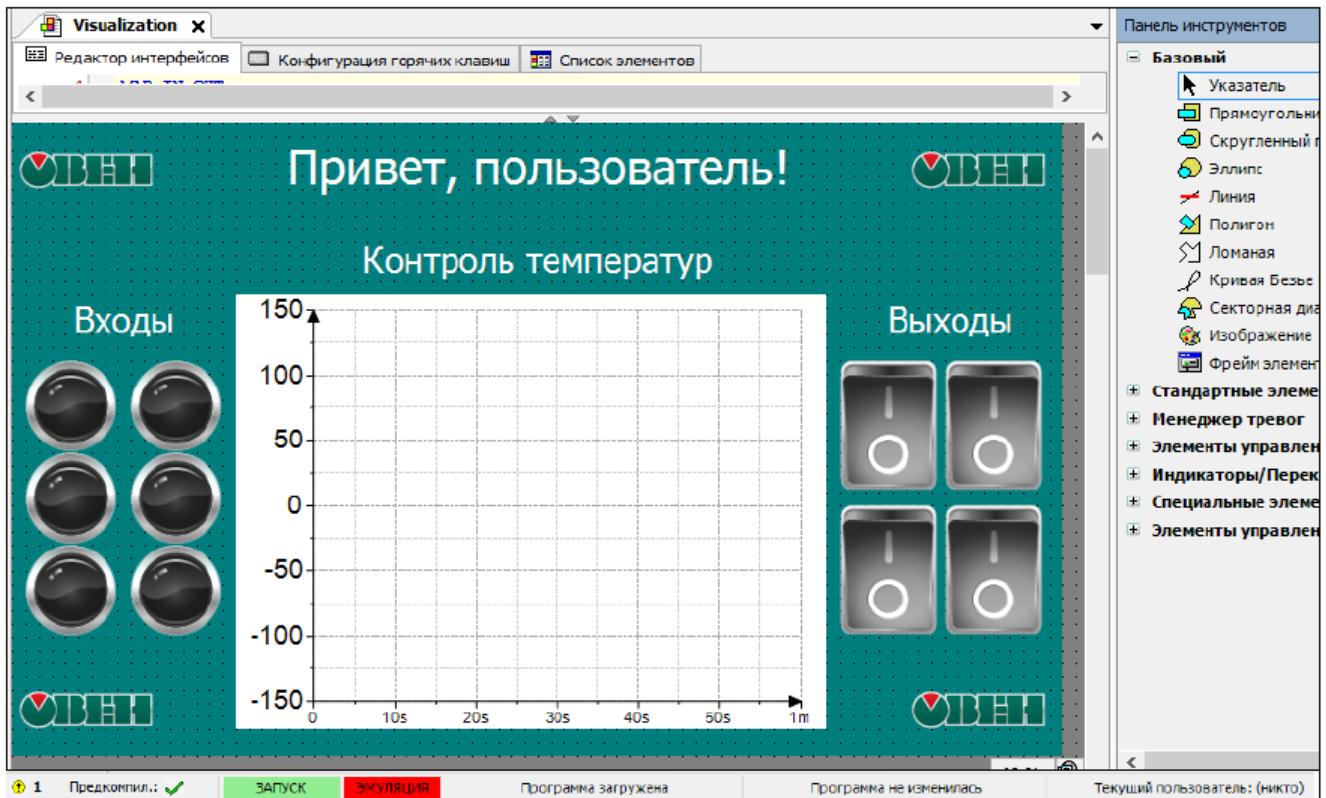


Рисунок 2.2 – Инструменты для создания проекта

**Ход работы. Ознакомление.** В данной лабораторной работе необходимо следующее:

- 1) ознакомиться со средой программирования CODESYS 3.5;
- 2) рассмотреть графические и текстовые языки программирования ПЛК;
- 3) ознакомится с инструментами для создания проекта;
- 4) открыть среду разработки и ознакомиться со стартовой страницей.

**Требования к содержанию отчёта:**

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Описание среды разработки CODESYS 3.5.
- 4) Стандарт МЭК 61131-3. Примеры языков данного стандарта.
- 5) Инструменты для создания проекта – инструменты программирования, отладки и создания визуализации.
- 6) Вывод.

**Контрольные вопросы:**

- 1) Что такое CODESYS?
- 2) Кто является разработчиком среды программирования?

- 3) Что такое стандарт МЭК 61131-3?
- 4) На каких языках можно программировать ПЛК, исходя из стандарта?
- 5) Какие инструменты необходимы для создания проекта?

## 2.2 Создание нового проекта. Принципы работы ПЛК. Цикл ПЛК. Переменные и типы данных CODESYS

*Цель работы:* ознакомится с принципами и особенностями создания нового проекта; изучить принципы работы ПЛК, циклы ПЛК, переменные и типы данных CODESYS.

### Теоретическая часть

**Создание нового проекта.** Прежде всего нужно дать проекту новое имя, оно же послужит и названием файла проекта. Первый программный компонент (POU – Program Organization Unit) помещается в новый проект автоматически и получает название PLC\_PRG.

**Типы программного компонента.** К программным компонентам в CoDeSys относятся: функция, функциональный блок и программа.

**Функция** (FUNCTION) – программный компонент, имеющий один или более входов, один выход – результат работы функции (выбранный тип данных), не имеющий внутренней памяти, который при каждом запуске работает аналогично и используется для комплексных вычислений.

**Функциональный блок** (FUNCTION\_BLOCK) – программный компонент, имеющий произвольное число входов и выходов и внутреннюю память.

**Программа** (PROGRAM) – программный компонент, подобный функциональному блоку, но имеющий один глобальный экземпляр. **Важно помнить:** все используемые программные компоненты должны вызываться прямо или косвенно из главной программы PLC\_PRG!

**Принципы работы ПЛК. Описание методики создания запросов в CoDeSys.** Для начала необходимо создать новый проект «Файл → Новый проект» (Ctrl+N), в появившемся окне перейти в категорию «Проекты» и выбрать «Стандартный проект», затем идёт выбор и настройка целевой платформы (устройства) и языка реализации основной программы PLC\_PRG (рис. 2.3). В конкретном случае необходимо выбрать устройство **PLC210** (Production association OWEN) и язык непрерывных функциональных схем **CFC**. После выбора устройства и языка необходимо нажать кнопку «OK».

**При создании нового программного компонента, где типом РОУ является «Программа», переименовывать данный тип не рекомендуется (имя остаётся по умолчанию «PLC\_PRG»).**

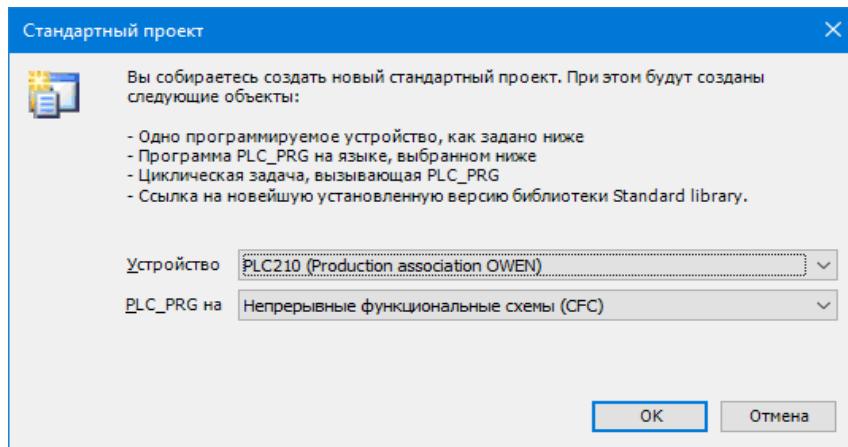


Рисунок 2.3 – Выбор целевой платформы и языка программирования

При созданном проекте пользователь видит три окна: слева – «Дерево проекта», в котором отображаются две вкладки – «Устройства» (где содержится вся информация о проекте: имя файла; устройство (*Device*), в котором расположены настройки (*Application*), *Менеджер библиотек*, основная программа *PLC\_PRG*, *Конфигурация задач*; выбранное оборудование (изначально PLC210\_01), в состав которого входят: соотношения входов/выходов (*LeftSide* и *RightSide*), *OwenRTC*, *OwenCloud*, *Buzzer*, *Drives*, *Debug*, *Info*, *WatchDog*, и вкладка *POU* – настройки среды; по центру – рабочее поле, внизу – отклик программы, где будут отображены ошибки (тип ошибки, место ошибки), предупреждения и сообщения. Для замены оборудования необходимо нажать правой клавишей мыши по старому оборудованию (PLC210\_01) и выбрать пункт «Обновить устройство». Далее появится окно замены (обновления) оборудования (рис. 2.4). Пользователю необходимо выбрать **PLC210\_04** и нажать кнопку «Обновить устройство», затем необходимо нажать кнопку «Закрыть».

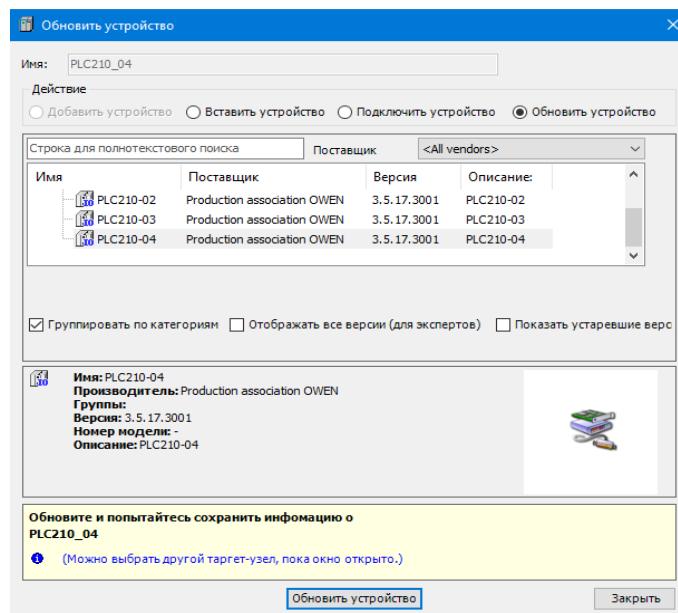


Рисунок 2.4 – Обновление (изменение) целевой платформы (устройства)

Новое устройство отобразиться в дереве проекта с его конфигурацией. Для замены имени устройства (при необходимости) следует нажать на устройство правой клавишей мыши и выбрать пункт «*Рефакторинг → Переименовать в...*», в появившемся окне которого необходимо в поле «*Новое имя*» задать изменённое.

**Настройка установки ПЛК.** Для настройки установок ПЛК необходимо в дереве проекта перейти во вкладку «*Device*» путем двойного нажатия на левую клавишу мыши. В рабочем поле откроется окно «*Device*», в котором расположено 15 вкладок: «*Установка соединения*», «*Приложения*», «*Резервное копирование и восстановление*», «*Файлы*», «*Журнал*», «**Установки ПЛК**», «*Оболочка ПЛК*», «*Пользователи и группы*», «*Права доступа*», «*Символьные права*», «*Licensed Software Metrics*», «*МЭК-объектов*», «*Размещение задачи*», «*Состояние*» и «*Информация*». Перейдем во вкладку «*Установки ПЛК*» (рис. 2.5). У пользователя есть возможность настройки ряда параметров: «*Приложение для обработки I/O*» (по умолчанию – **Application**), «*Установки ПЛК*», «*Опции цикла шины*» и «*Доп. установки*». Необходимо в параметре «*Всегда обновлять переменные*» выбрать «**Включено 2 (всегда в задаче цикла шины)**», в параметре «*Задача цикла шины*» выбрать «**MainTask**», в параметре «*Генерировать переменные фиксации для соотнесения I/O*» выбрать «**Вкл**», в параметре «*Показывать предупреждения I/O как ошибки*» выбрать «**Вкл**».

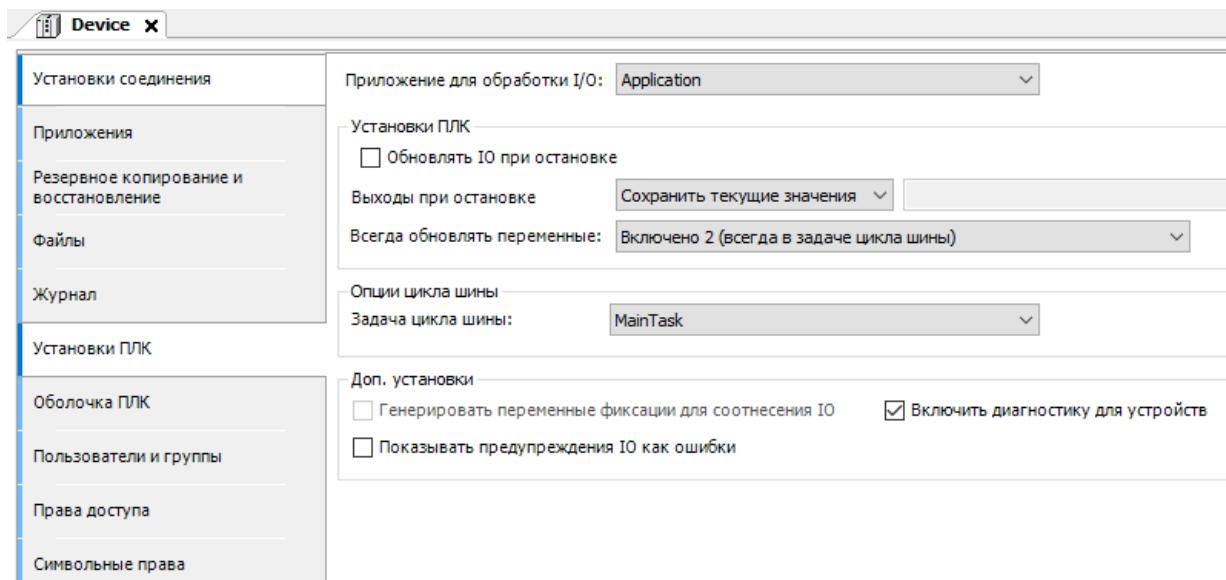


Рисунок 2.5 – Настройка установок ПЛК

Для корректной работы проекта есть возможность изменять время задачи цикла. Для этого необходимо перейти во вкладку «*Конфигурация задач* → *Main Task* → выбрать тип «*Циклическое*» → задать интервал в ms (например, 10ms)» (рис. 2.6).

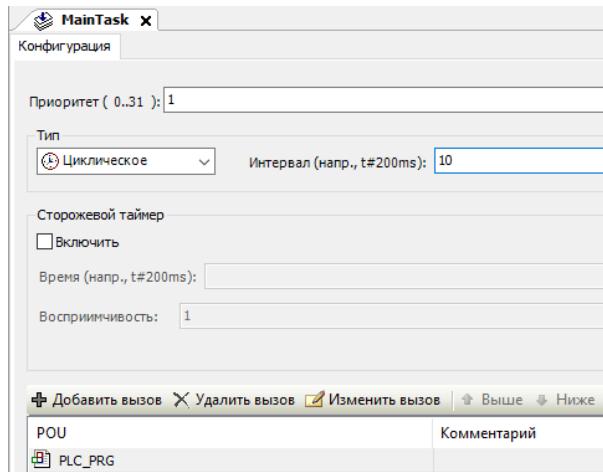


Рисунок 2.6 – Конфигурация задач

**Цикл ПЛК.** Все ПЛК работают по методу периодического опроса входных данных (сканирования). ПЛК опрашивает входы, выполняет пользовательскую программу и устанавливает необходимые значения выходов (рис. 2.7).

Рабочий цикл ПЛК состоит из нескольких фаз:

1. Начало цикла.
2. Чтение состояния входов.
3. Выполнение кода программы пользователя.
4. Запись состояния выходов.
5. Обслуживание аппаратных ресурсов ПЛК.
6. Монитор системы исполнения.
7. Контроль времени цикла.
8. Переход на начало цикла.

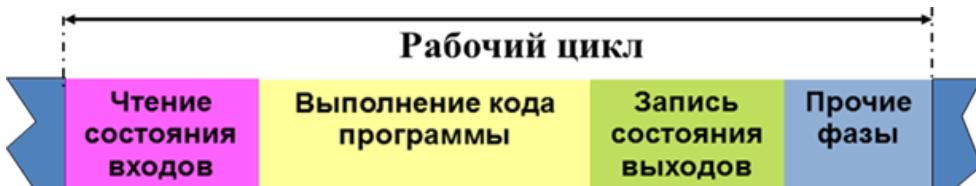


Рисунок 2.7 – Цикл ПЛК

**Типы данных переменных.** Существует два вида переменных: глобальные и локальные.

**Глобальные переменные** – это переменные, связанные с входами и выходами контроллера, которые инициализируются во вкладке «PLC\_210» → *LeftSide* и *RightSide*.

**Локальные переменные** (внутренние, содержащиеся в основном теле проекта) объявляются при помощи сочетания клавиш *Shift+F2*, или вручную в окне объявления переменных (верхнее окно).

**Важно помнить: переменная объявляется только один раз, либо как локальная, либо как глобальная и обладает именем и типом!**

При наименовании переменной следует знать:

- переменная должна содержать буквы и цифры английского алфавита;
- переменная начинается только с буквы, где регистр не учитывается;
- переменная не должна содержать пробелов, но возможно использование одинарного подчеркивания;
- в имени переменной нельзя использовать зарезервированные слова (AND, PROGRAM, VAR и т. д.);
- нельзя использовать название и выходы функциональных блоков (RS, R, S, TP, CTU и т. д.);
- не должна содержать символы в имени (‘, <, ., > и т. д.).

**Типы данных переменных.** Для присвоения переменных используются следующие типы данных: логический (BOOL) – дискретные значения (FALSE...TRUE), вещественный (INT) – целые, отрицательные или положительные числа (-32768...32767), вещественный неотрицательный (WORD) – целые, только положительные числа (0...65535), с плавающей точкой (REAL) – целые и дробные числа в широком диапазоне (-1.2\*10<sup>-38</sup>...3.4\*10<sup>38</sup>), строковый (STRING) – произвольная последовательность символов, временной (TIME) – временные значения (пример T#5h10m35s15ms).

**Горячие клавиши.** Горячие клавиши предназначены для замены некоторых функций, что упрощает работу с проектом. Ниже представлен список основных горячих клавиш:

- **F1** – вызов справки. Для того чтобы вызвать справку для конкретного элемента, необходимо выделить его название и нажать F1.
- **F2** – ассистент ввода. Чтобы вызвать ассистент ввода, необходимо выделить знаки «???» и нажать F2. Помогает быстро и безошибочно присвоить ранее объявленную переменную.
- **Shift + F2** – объявление локальной переменной.
- **F5** – запуск программы.
- **F11** – компиляция. Проверка на ошибки.
- **Ctrl + F7** – записать значение.

## Практическая часть

**Ход работы. Создание проекта в среде CoDeSys.** Пользователю необходимо создать проект в среде CoDeSys, выбрать тип программного компонента (программа) и язык реализации (CFC), добавить необходимые переменные, загрузить проект в ПЛК либо же симулировать его работу, проверить проект на работоспособность.

На примере простой задачи реализуем создание проекта. Допустим, у пользователя есть кнопка и лампочка. При нажатии на кнопку загорается лампочка (рис. 2.8).

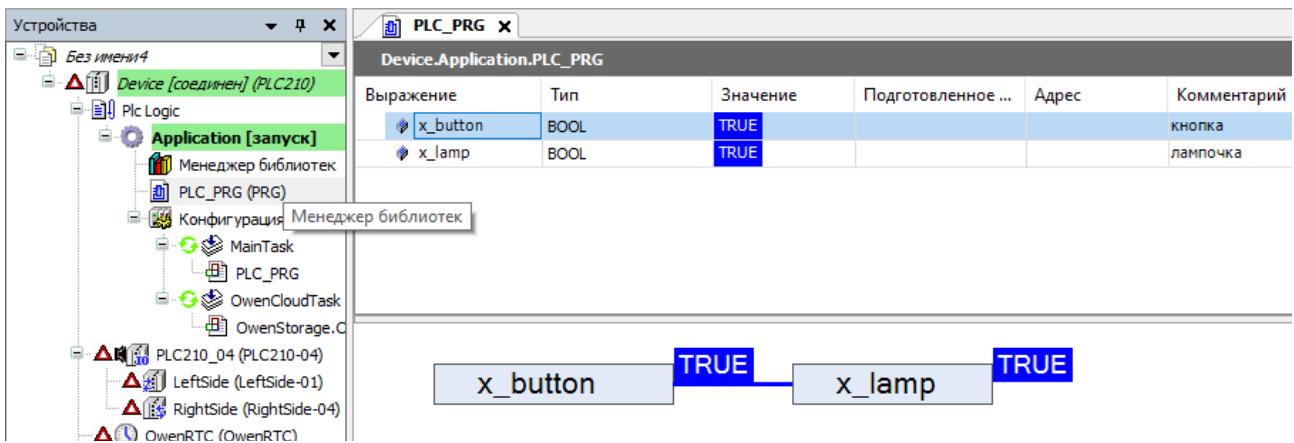


Рисунок 2.8 – Работа лампочки при нажатии на кнопку

Для проверки проекта на ошибки необходимо компилировать проект: проследовать по пути «Компиляция → Генерировать код» либо воспользоваться горячей клавишей «F11». Для эмуляции работы контроллера необходимо перейти во вкладку «Онлайн → Эмуляция» и поставить галочку (рис. 2.9). После этого пользователь работает в режиме эмуляции. Затем, чтобы загрузить проект в контроллер необходимо перейти во вкладку «Онлайн → Логин» (рис.2.9), либо воспользоваться сочетанием горячих клавиш «Alt+F8», после чего проект загрузится в виртуальный контроллер. Далее для запуска проекта необходимо перейти во вкладку «Отладка → Старт» или использовать горячую клавишу «F5», затем пользователь сможет работать с этим проектом.

При старте программы переменные имеют значение «FALSE». Для изменения состояния переменной необходимо курсором перейти на поле «Подготовленное значение», нажать на него один раз левой клавишей мыши. После этого появится изменённое значение (в нашем случае «TRUE»). Чтобы присвоить выбранное значение необходимо нажать сочетание горячих клавиш «Ctrl+F7». Для отключения от проекта необходимо перейти во вкладку «Онлайн → Отключение», либо воспользоваться сочетанием горячих клавиш «Ctrl+F8».

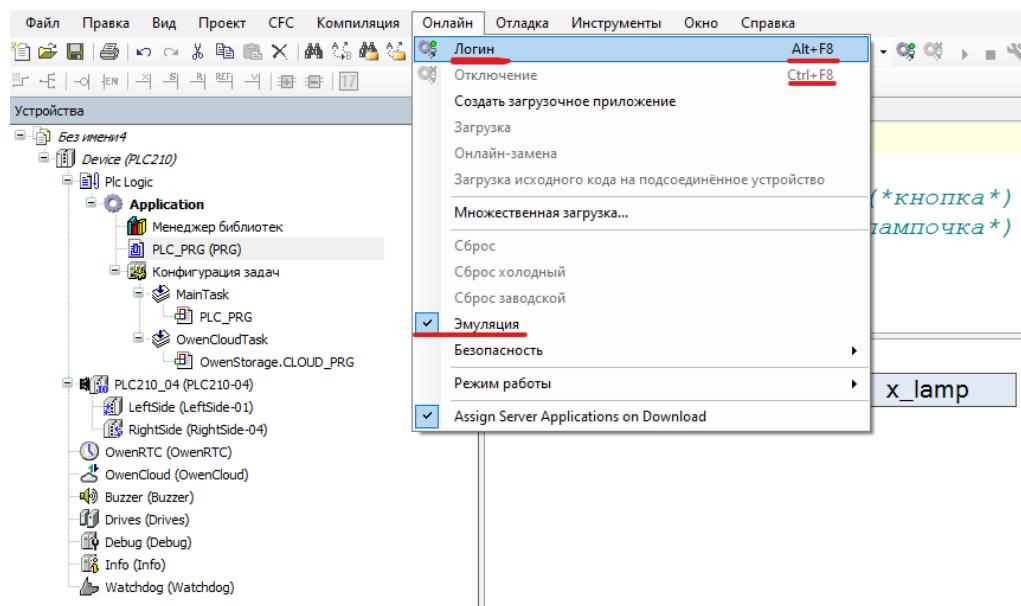


Рисунок 2.9 – Подключение проекта

### Требования к содержанию отчёта:

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Знакомство со средой программирования CoDeSys v3.5.
- 4) Создание нового проекта.
- 5) Типы программного компонента.
- 6) Методика создания запросов в CoDeSys v3.5.
- 7) Типы данных переменных. Горячие клавиши.
- 8) Реализовать методику создания проекта в среде CoDeSys v3.5.
- 9) Вывод.

### Контрольные вопросы:

- 1) Что такое «POU»?
- 2) Что такое «Программа»?
- 3) Что такое «Функциональный блок»?
- 4) Как реализуется методика создания запросов в CoDeSys?
- 5) Каким образом объявляются «Глобальные» переменные?
- 6) Как назначаются «Локальные» переменные?
- 7) Как реализуется настройка установки ПЛК?
- 8) Перечислите типы данных переменных?
- 9) Какие горячие клавиши используются для работы с проектом?

## 2.3 Знакомство с языком CFC. Стандартные операторы CODESYS: логика, арифметика

*Цель работы:* ознакомится с графическим языком программирования CFC, изучить основные характеристики и принципы работы операторов CoDeSys: логики, арифметики.

### Теоретическая часть

**Графический язык программирования CFC.** Язык CFC (Continuous Flow Chart) – ещё один высокоуровневый язык визуального программирования. По сути, CFC – это дальнейшее развитие языка FBD. Этот язык был специально создан для проектирования систем управления непрерывными технологическими процессами (рис. 2.10).

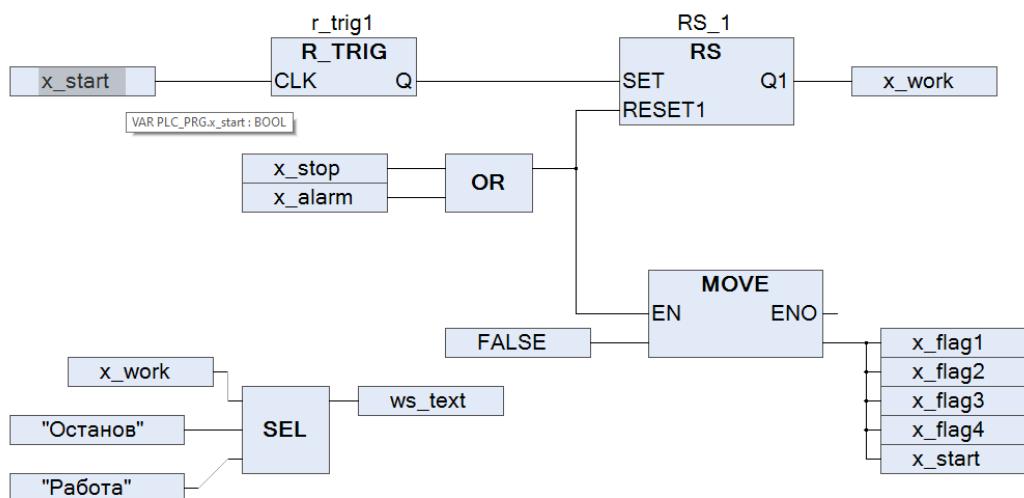


Рисунок 2.10 – Пример программы, реализованной на языке CFC

**Логические операторы CoDeSys.** Логическими операторами, использующимися в CoDeSys, являются: логическое «И», логическое «ИЛИ», логическое «исключающее ИЛИ», логическое «НЕ». Все логические операторы работают со следующими типами данных: BOOL, BYTE, WORD или DWORD.

**Логическое «И».** Логическое «И» представлено элементом «AND». В своей структуре имеет два входа и один выход (рис. 2.11). Логика работы элемента представлена в таблице 2.1.

Таблица 2.1 – Логика работы элемента логического «И»

1-ый вход	2-ой вход	выход
0	0	0
1	0	0
0	1	0
1	1	1

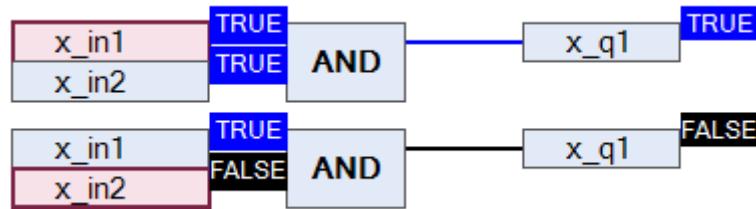


Рисунок 2.11 – Логическое «И»

**Логическое «ИЛИ».** Логическое «ИЛИ» представлено элементом «OR». В своей структуре имеет два входа и один выход (рис. 2.12). Логика работы элемента представлена в таблице 2.2.

Таблица 2.2 – Логика работы элемента логического «ИЛИ»

1-ый вход	2-ой вход	выход
0	0	0
1	0	1
0	1	1
1	1	1

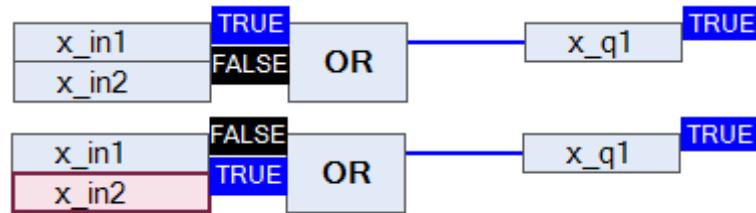


Рисунок 2.12 – Логическое «ИЛИ»

**Логическое «исключающее ИЛИ».** Логическое «исключающее ИЛИ» представлено элементом «XOR». В своей структуре имеет два входа и один выход (рис. 2.13). Логика работы элемента представлена в таблице 2.3.

Таблица 2.3 – Логика работы логического элемента «исключающее ИЛИ»

1-ый вход	2-ой вход	выход
0	0	0
1	0	1
0	1	1
1	1	0

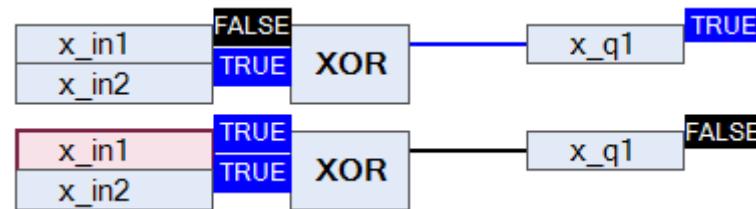


Рисунок 2.13 – Логическое «исключающее ИЛИ»

**Логическое «НЕ».** Логическое «НЕ» представлено элементом «NOT». В своей структуре имеет один вход и один выход (рис. 2.14). Логика работы элемента представлена в таблице 2.4.

Таблица 2.4 – Логика работы логического элемента «НЕ»

вход	выход
0	1
1	0

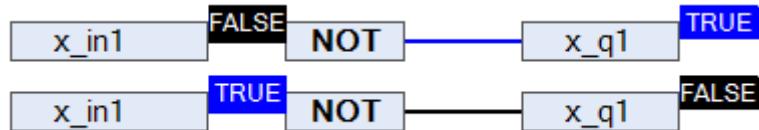


Рисунок 2.14 – Логическое «НЕ»

**Арифметические операторы CoDeSys.** Арифметическими операторами, использующимися в CoDeSys, являются: сложение, умножение, вычитание, деление, остаток от деления. Название блоков арифметических элементов может указываться как буквенно, так и графически.

**Арифметическое «Сложение».** Элемент «Сложение» в CoDeSys может быть представлено как в письменном виде (ADD), так и в графическом (+). В своей первоначальной структуре имеет два входа и один выход (рис. 2.15). Типы входных и выходных данных, с которыми используется элемент: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL, TIME. Принцип работы: первое слагаемое (верхний вход) складывается со вторым слагаемым (нижний вход), в итоге получаем сумму чисел того же типа, что и на входе.

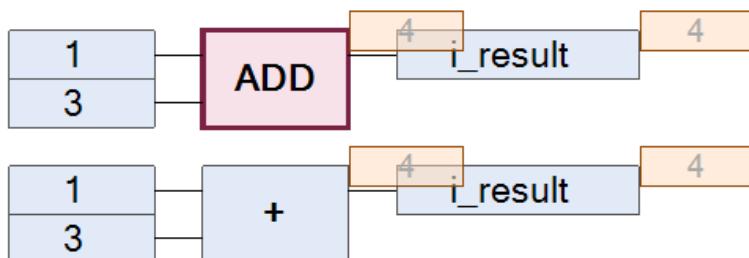


Рисунок 2.15 – Арифметический оператор «Сложение».

**Арифметическое «Умножение».** Элемент «Умножение» в CoDeSys может быть представлено как в письменном виде (MUL), так и в графическом. В своей первоначальной структуре имеет два входа и один выход (рис. 2.16). Типы входных и выходных данных, с которыми используется элемент: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT. Принцип работы: первый множитель (верхний вход) умножается со вторым множителем (нижний вход), в итоге получаем произведение чисел того же типа, что и на входе.

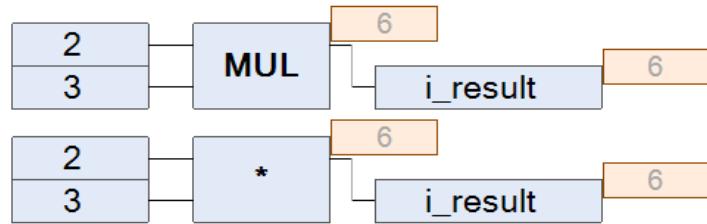


Рисунок 2.16 – Арифметический оператор «Умножение»

**Арифметическое «Вычитание».** Элемент «Вычитание» в CoDeSyS может быть представлено как в письменном виде (SUB), так и в графическом (-). В своей структуре имеет два входа и один выход (рис. 2.17). Типы входных и выходных данных, с которыми используется элемент: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL и LREAL, TIME. Принцип работы следующий: уменьшаемое (верхний вход) вычитает вычитаемое (нижний вход), в итоге получаем разность чисел того же типа, что и на входе.

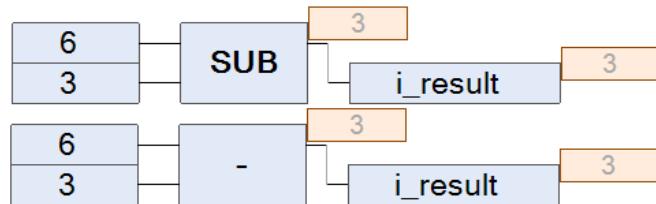


Рисунок 2.17 – Арифметический оператор «Вычитание»

**Арифметическое «Деление».** Элемент «Деление» в CoDeSyS может быть представлено как в письменном виде (DIV), так и в графическом (/). В своей структуре имеет два входа и один выход (рис. 2.18). Типы входных и выходных данных, с которыми используется элемент: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT. Принцип работы следующий: делимое (верхний вход) делится на делитель (нижний вход), в итоге получаем частность чисел того же типа, что и на входе.

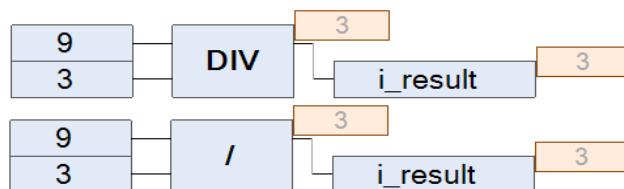


Рисунок 2.18 – Арифметический оператор «Деление».

**Арифметический «Остаток от деления».** Элемент «Остаток от деления» в CoDeSyS обозначается письменном виде (MOD). В своей структуре имеет два входа и один выход (рисунок 2.19). Типы входных и выходных данных, с которыми используется элемент: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT. Принцип работы следующий: делимое (верхний вход) делится на делитель (нижний вход), в итоге получаем следующее:

а) если делимое делится на делитель без остатка (например, 4/2), то в результате на выходе блока мы получим значение, равное 0;

б) если делимое делится на делитель с остатком, то выполняется следующие действия: находится число, которое меньше делителя и делится на делитель без остатка, затем от делимого отнимается это число, в результате получаем искомый результат.

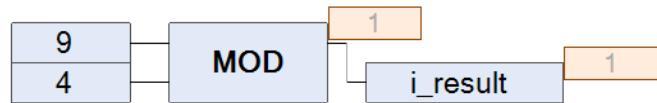


Рисунок 2.19 – Арифметический оператор «Остаток от деления»

## Практическая часть

**Ход работы. Реализация изменения состояния объекта управления с использованием нескольких устройств управления.** Объектом управления выступит лампочка (out1), устройствами управления – кнопки (in7, in8, in9) (рис. 2.20). Пользователь имеет возможность менять состояние объекта управления при помощи трёх вышеупомянутых устройств управления. Объект управления изменяет своё состояние при одновременной работе двух или более устройств управления (рисунок 2.21). Реализовать программу на языке CFC в среде CoDeSys с использованием логических операторов.



Рисунок 2.20 – Изменение состояния объекта управления с использованием нескольких устройств управления



Рисунок 2.21 – Реализация управления над объектом

**Найти решение уравнению.** Пользователю дано четыре входных переменных ( $R1 - R4$ ) и одна выходная переменная ( $R5$ ), который имеет право изменять значения входных переменных.

Вычислить значение выходной переменной, исходя из уравнения (рис. 2.22):

$$R5 = \frac{(R1 - 10)}{(R2 + 11)} \cdot (R3 - 4) + R4.$$

Результат уравнения доступен при нажатии при включении тумблера  $in1$  (рис. 2.23). Для проверки правильного написания уравнения рекомендуется ввести следующие значения для входных параметров:  $R1 = 60$ ,  $R2 = 89$ ,  $R3 = 20$ ,  $R4 = 4$ . Необходимо реализовать данную задачу с использованием арифметических операторов.



Рисунок 2.22 – Исходные данные



Рисунок 2.23 – Результат решения уравнения

### Требования к содержанию отчёта:

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Описание графического языка программирования CFC.
- 4) Логические и арифметические операторы CoDeSys.
- 5) Реализация изменения состояния объекта управления с использованием нескольких устройств управления.
- 6) Найти решение уравнению.
- 7) Вывод.

## **Контрольные вопросы:**

- 1) Графический язык программирования CFC.
- 2) Логические операторы CoDeSys: описание, структура, обозначение, примеры.
- 3) Арифметические операторы CoDeSys: описание, структура, обозначение, примеры.
- 4) Какие типы данных имеют входы и выходы у логических операторов?
- 5) Какие типы данных имеют входы и выходы у арифметических операторов?
- 6) Особенности операторов «Сложение» и «Умножение».

## **2.4 Стандартные операторы CODESYS: операторы выбора и ограничения, сравнение. Работа с входами и выходами ПЛК210.**

*Цель работы:* ознакомится и изучить основные характеристики и принципы работы операторов CoDeSys: операторы выбора и ограничения, сравнения, произвести работу с входами и выходами ПЛК210

### **Теоретическая часть**

**Операторы выбора и ограничения. Селектор.** Селектор – оператор побитной выборки. В своей структуре оператор имеет три входных элемента и один выходной. Первый вход селектора (верхний) имеет тип данных BOOL, остальные входы и выход могут быть любого типа данных. Однако, типы данных у оставшихся входов и выхода должны совпадать. Принцип работы : при подаче на первый вход логического нуля, выходному элементу присваивается значение второго входного элемента; если же на первый вход была подана логическая единица, то выходному элементу присваивается значение третьего входного элемента (рис. 2.24).

**Операторы выбора наибольшего и наименьшего значения.** Оператор выбора наибольшего значения (MAX) – оператор, который из приведённых значений выбирает наибольшее. В своей структуре имеет два входа и один выход (рис. 2.25).

Оператор выбора наименьшего значения (MIN) – оператор, который из приведённых значений выбирает наименьшее. В своей структуре имеет два входа и один выход.

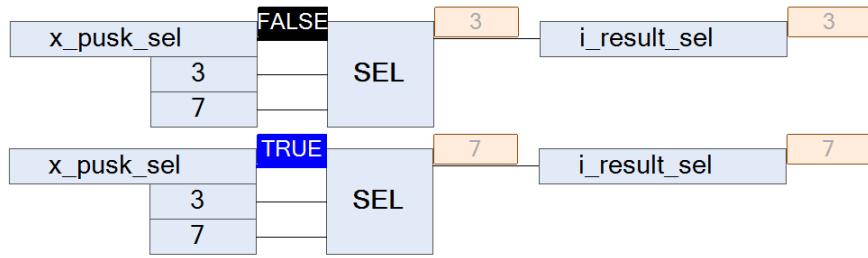


Рисунок 2.24 – Оператор «Селектор»

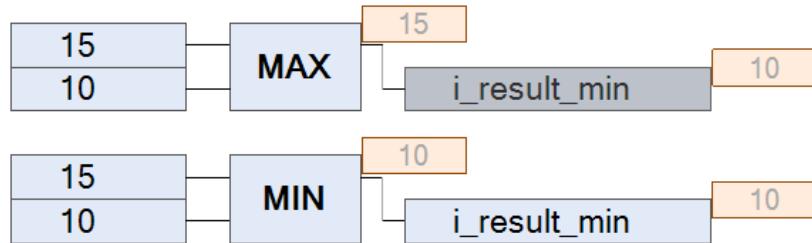


Рисунок 2.25 – Операторы выбора наименьшего и наибольшего значения

**Ограничитель.** Ограничитель (LIMIT) – оператор, который ограничивает значение переменной в заданном промежутке значений, который работает со всеми типами данных. В своей структуре оператор имеет три входа и один выход. Первому входу присваивается начальное значение промежутка, второй вход – переменная, которую мы ограничиваем, третий вход – конечное значение промежутка (рис. 2.26). Если значение лимитированной переменной выходит за указанный промежуток, то выходу присваивается значение промежутка, близкому к заданному.

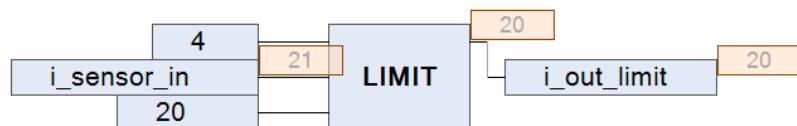


Рисунок 2.26 – Оператор «Ограничитель»

**Оператор «Присвоение».** Присвоение (MOVE) – оператор, который присваивает значение входной переменной выходной. Классический оператор в своей структуре имеет один вход и один выход (рис. 2.27). По сути, классический оператор выполняет функцию прямого присвоения и в своей классической компоновке не используется.

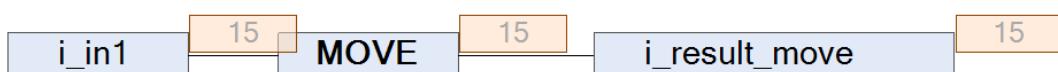


Рисунок 2.27 – Классическое представление оператора сравнения

В качестве оператора блок «Присвоение» используется с дополнительными входами условиями срабатывания (EN/ENO) (рис. 2.28). Вход «EN» служит для условия срабатывания блока присвоения (рис. 2.29) и имеет тип данных BOOL. Принцип работы следующий: если на вход запуска работы блока присвоения (EN) приходит логическая единица, то значению выхода присваивается значения входа, иначе выход имеет свое состояние, которое ему было присвоено ранее. Стоит заметить, что типы данных входного и выходного элемента должны совпадать.

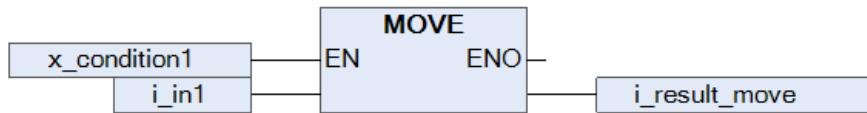


Рисунок 2.28 – Оператор «Присвоение» с дополнительным входом условия

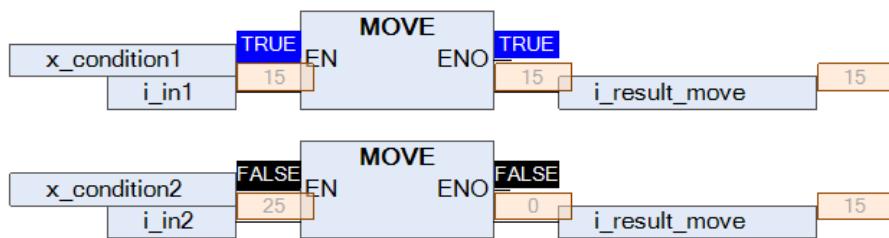


Рисунок 2.29 – Работа оператора «Присвоение»

**Мультиплексор.** Мультиплексор – это устройство с ячейками памяти, в которых хранится информация, в зависимости от выбора ячейки может передаваться на исполнительное устройство. В своей первоначальной структуре мультиплексор имеет 4 входа (один управляющий, три – хранящие данные) и один выход (рисунок 2.30). Верхний вход является управляющим входом, имеющим тип данных INT, который в свою очередь указывает на регистр (ячейку памяти), в которой хранится информация (порядок регистров начинается с «0»). Последующие входы являются ячейками памяти (регистрами), в которых хранится необходимая информация. Типы данных входных элементов (регистров) и выходного элемента должны совпадать.

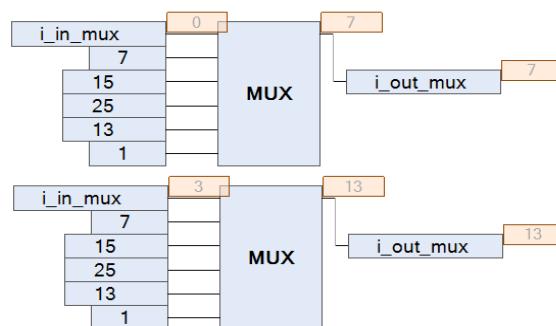


Рисунок 2.30 – Оператор «Мультиплексор»

**Операторы сравнения.** Операторами сравнения в CoDeSys являются: «Больше», «Меньше», «Больше или равно», «Меньше или равно», «Равно», и «Неравно». Операторы сравнения в своей структуре имеют два входа и один выход. На входе оператора переменные могут быть любого типа данных, главное, чтобы типы у этих переменных совпадали. Выходная переменная может иметь только один тип данных: BOOL.

**Оператор сравнения «Больше».** Оператор сравнения «Больше» может обозначаться письменно как «GT», так и символично «>». Работает следующим образом: если значение первого входа больше значения второго – результатом будет логическая единица, иначе – логический ноль (рис. 2.31).

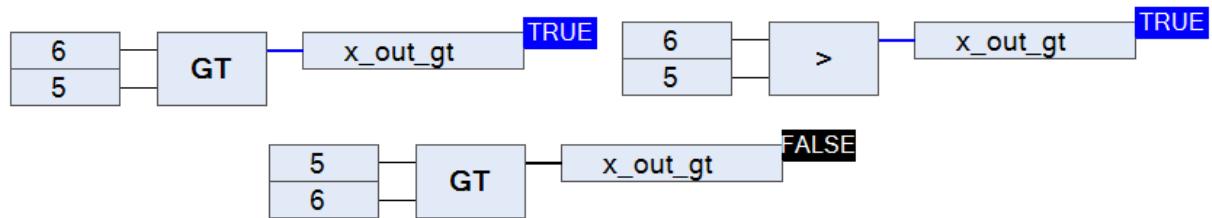


Рисунок 2.31 – Оператор сравнения «Больше»

**Оператор сравнения «Меньше».** Оператор сравнения «Меньше» может обозначаться письменно как «LT», так и символично «<». Работает следующим образом: если значение первого входа меньше значения второго – результатом будет логическая единица, иначе – логический ноль (рис. 2.32).

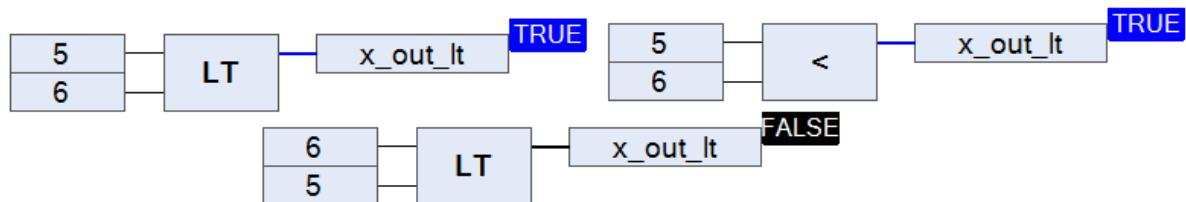


Рисунок 2.32 – Оператор сравнения «Меньше»

**Оператор сравнения «Больше или равно».** Оператор сравнения «Больше или равно» может обозначаться письменно как «GE», так и символично «>=». Работает следующим образом: если значение первого входа больше или равно значению второго – результатом будет логическая единица, иначе – логический ноль (рис. 2.33).

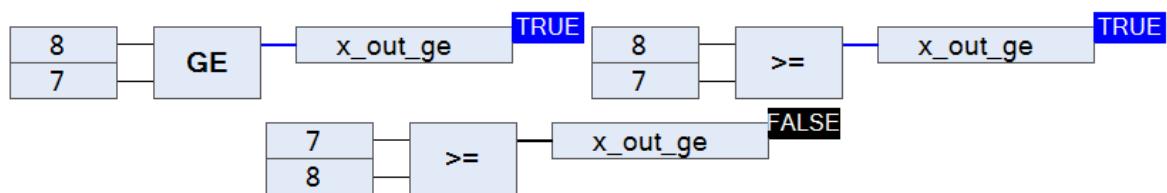


Рисунок 2.33 – Оператор сравнения «Больше или равно»

**Оператор сравнения «Меньше или равно».** Оператор сравнения «Меньше или равно» может обозначаться письменно как «LE», так и символично « $\leq$ ». Работает следующим образом: если значение первого входа меньше или равно значению второго – результатом будет логическая единица, иначе – логический ноль (рис. 2.34).

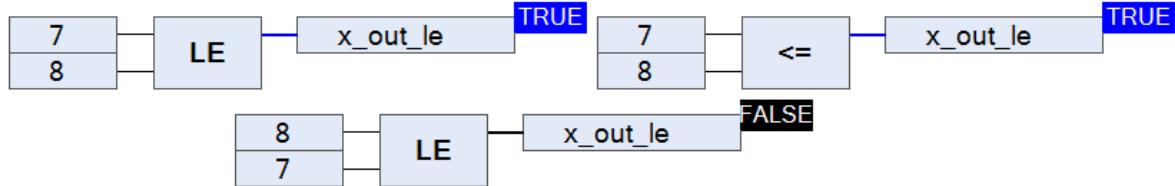


Рисунок 2.34 – Оператор сравнения «Меньше или равно»

**Оператор сравнения «Равно».** Оператор сравнения «Равно» может обозначаться письменно как «EQ», так и символично « $=$ ». Работает следующим образом: если значение первого входа равно значению второго – результатом будет логическая единица, иначе – логический ноль (рис. 2.35).

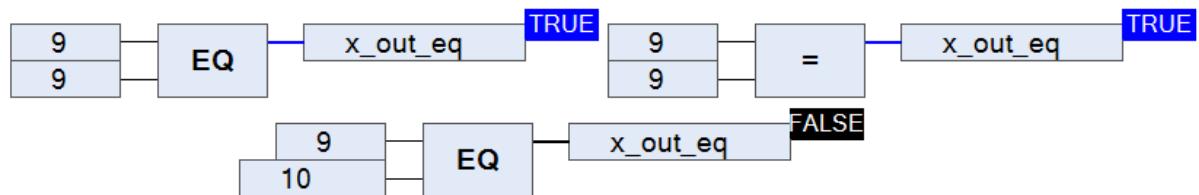


Рисунок 2.35 – Оператор сравнения «Равно»

**Оператор сравнения «Не равно».** Оператор сравнения «Не равно» может обозначаться письменно как «NE», так и символично « $\neq$ ». Выполняет противоположные функции относительно оператора «Равно» (рис. 2.36).

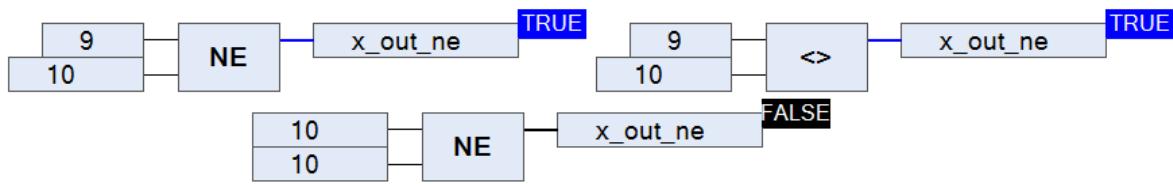


Рисунок 2.36 – Оператор сравнения «Не равно»

**Работа с входами и выходами ПЛК210.** Для работы с входами и выходами ПЛК 210-04 пользователю необходимо инициализировать переменные в блоке объявления переменных (класс VAR). Тип данных переменных, исходя из входных/выходных точек ПЛК210 – логический (**BOOL**). После инициализации переменных в дереве проекта необходимо перейти в категорию выбранного оборудования «PLC210\_04», затем выбрать вкладку «LeftSide» или «RightSide», далее категорию «Соотнесение входов/выходов» (рис. 2.37).

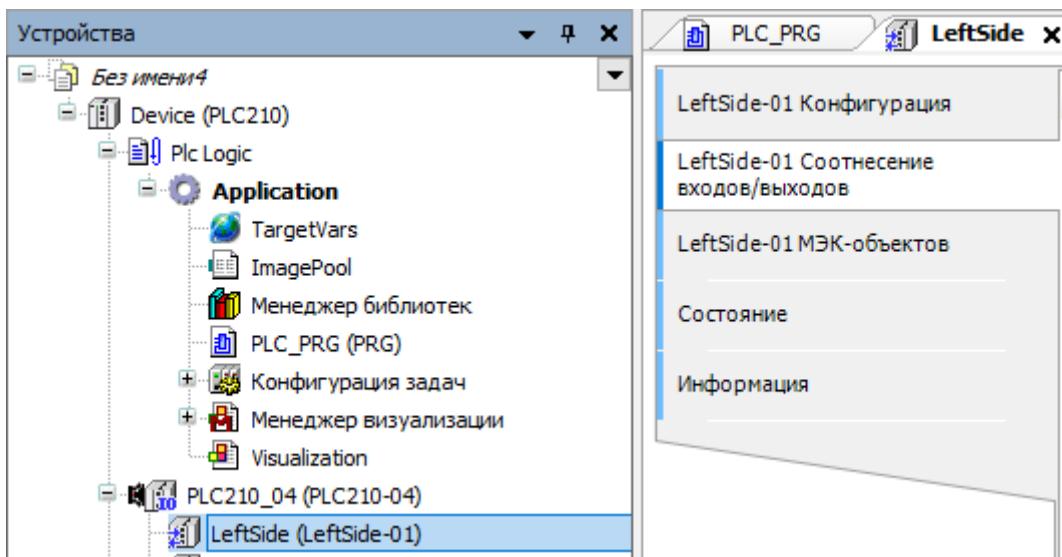


Рисунок 2.37 – Работа с входами/выходами ПЛК210-04

Компоновка входов/выходов ПЛК210-04 следующая: в разделе «LeftSide» во вкладке «LeftSide-01 Соотнесение входов/выходов» расположены: «Битовая маска входов», при раскрытии которой отображаются 12 быстрых входов; «Битовая маска выходов (на запись)», при открытии которой отображаются 4 выхода; «Счётчики входов», «Генераторы импульсов», «Коэффициент заполнения ШИМ», «AB энкодеры», «ABZ энкодеры» (рис. 2.38).

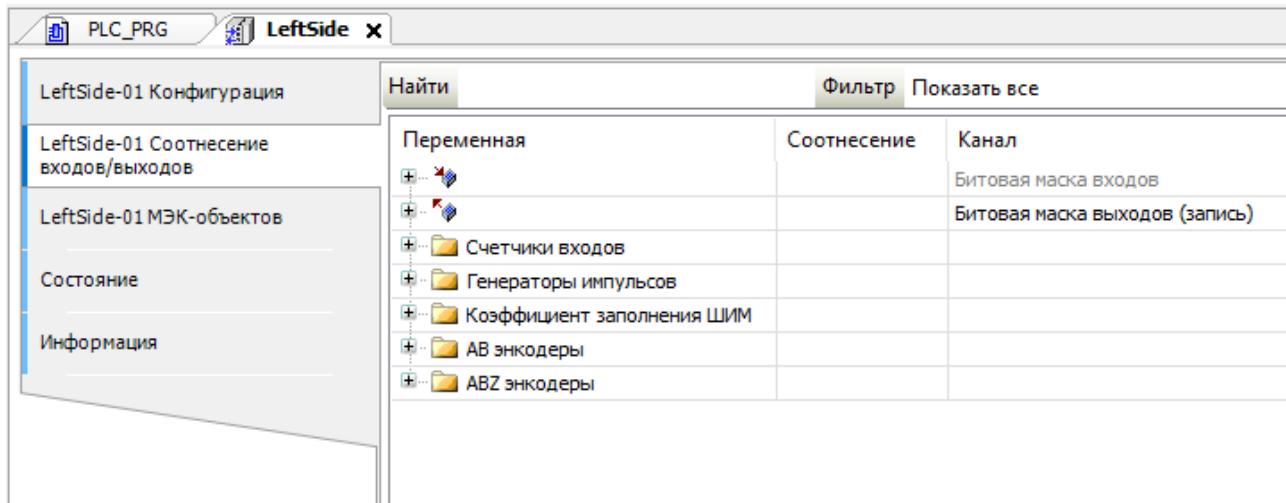
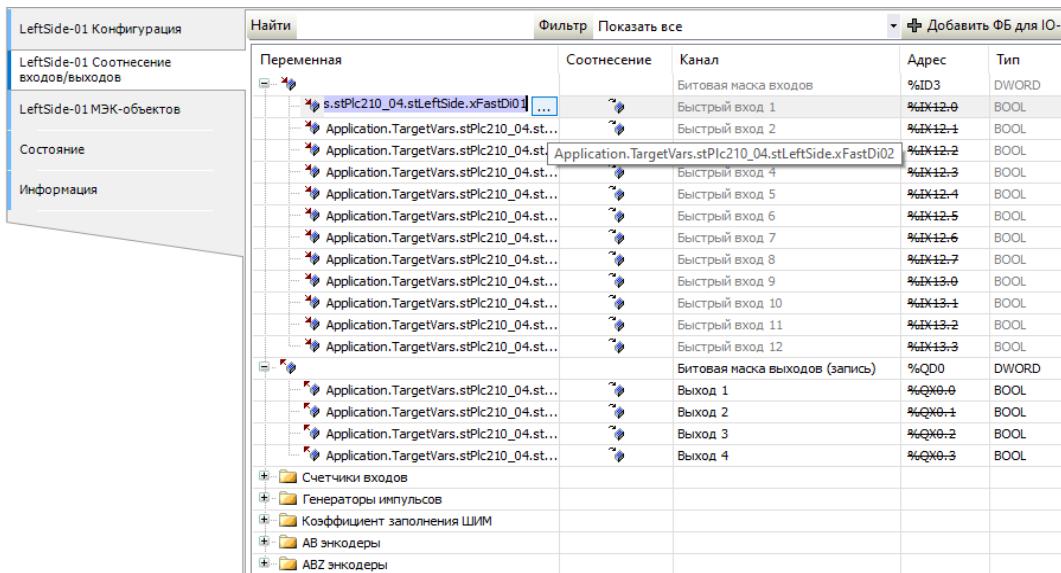


Рисунок 2.38 – Вкладка «LeftSide-01 Соотнесение входов/выходов»

Для добавления входных или выходных переменных пользователю необходимо перейти в соответствующие вкладки: «Битовая маска входов» и «Битовая маска выходов (на запись)» (рисунок 2.38). После открытия необходимой вкладки и выбора работы с соответствующим входом/выходом (например, «Быстрый вход 1» и «Выход 1») необходимо их инициализировать. Для этого следует два раза нажать левой клавишей мыши по столбцу «Переменная» выбранной позиции и с помощью либо «Ассистента ввода» (клавиша «F2»), либо

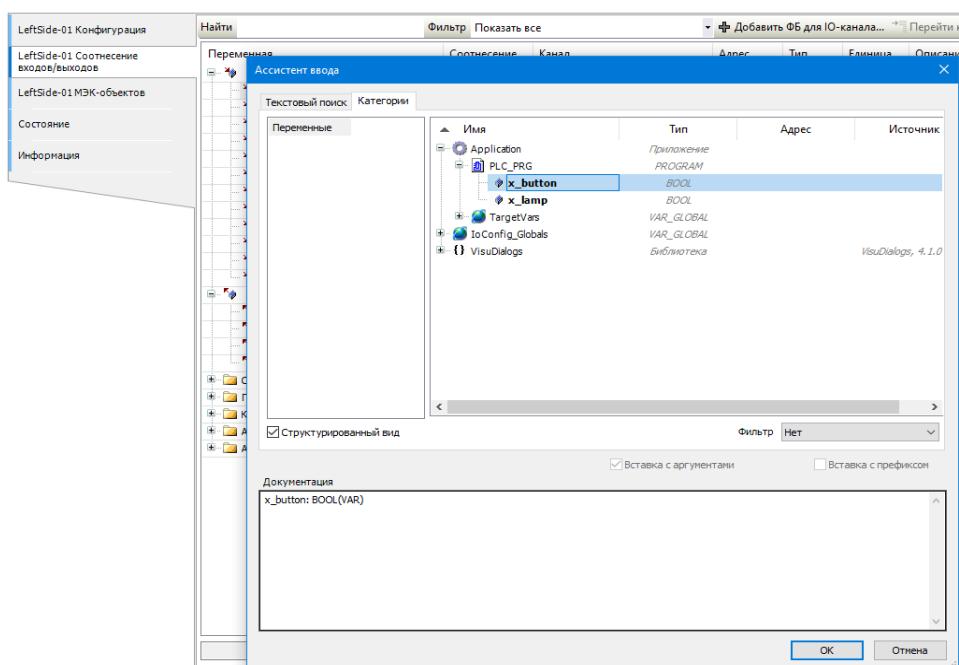
путём нажатия на вкладку «...» инициализировать вход/выход (привязать переменную) (рис. 2.39).



Переменная	Соотнесение	Канал	Адрес	Тип
s.stPlc210_04.stLeftSide.xFastDi01	Битовая маска входов		%ID3	DWORD
Application.TargetVars.stPlc210_04.st...	Быстрый вход 1		%IX12.0	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 2		%IX12.1	BOOL
Application.TargetVars.stPlc210_04.st...	Application.TargetVars.stPlc210_04.stLeftSide.xFastDi02		%IX12.2	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 4		%IX12.3	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 5		%IX12.4	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 6		%IX12.5	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 7		%IX12.6	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 8		%IX12.7	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 9		%IX13.0	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 10		%IX13.1	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 11		%IX13.2	BOOL
Application.TargetVars.stPlc210_04.st...	Быстрый вход 12		%IX13.3	BOOL
	Битовая маска выходов (запись)		%QD0	DWORD
	Выход 1		%QX0.0	BOOL
	Выход 2		%QX0.1	BOOL
	Выход 3		%QX0.2	BOOL
	Выход 4		%QX0.3	BOOL
Счетчики входов				
Генераторы импульсов				
Коэффициент заполнения ШИМ				
AB энкодеры				
ABZ энкодеры				

Рисунок 2.39 – Выбор необходимого входа/выхода

После перехода в «Ассистент ввода» следует выбрать необходимую переменную, которую нужно привязать (рисунок 2.40). После выбора переменной следует нажать кнопку «OK».



Имя	Тип	Адрес	Источник
Application	Приложение		
PLC_PRG	PROGRAM		
x_button	BOOL		
x_lamp	BOOL		
TargetVars	VAR_GLOBAL		
IoConfig_Globals	VAR_GLOBAL		
VisuDialogs	библиотека		VisuDialogs, 4.1.0

Рисунок 2.40 – Выбор переменной к выбранному входу/выходу

Привязанная переменная появится во вкладке «Переменная» инициализированного входа/выхода (рис. 2.41). Инициализация входов/ выходов завершена.

Переменная	Соотнесение	Канал	Адрес	Тип
Application.PLC_PRG.x_button		Быстрый вход 1	%IX12-0	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 2	%IX12-1	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 3	%IX12-2	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 4	%IX12-3	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 5	%IX12-4	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 6	%IX12-5	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 7	%IX12-6	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 8	%IX12-7	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 9	%IX12-8	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 10	%IX12-9	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 11	%IX12-10	BOOL
Application.TargetVars.stPlc210_04.st...		Быстрый вход 12	%IX12-11	BOOL
Битовая маска выходов (запись)			%QD0	DWORD
Application.PLC_PRG.x_lamp		Выход 1	%QX0-0	BOOL
Application.TargetVars.stPlc210_04.st...		Выход 2	%QX0-1	BOOL
Application.TargetVars.stPlc210_04.st...		Выход 3	%QX0-2	BOOL
Application.TargetVars.stPlc210_04.st...		Выход 4	%QX0-3	BOOL

Рисунок 2.41 – Инициализация переменной к выбранному входу/выходу

## Практическая часть

**Ход работы. Реализация конфигурирования объектом управления (вентилятором) в зависимости от влияющих факторов (температуры) с использованием операторов выбора и ограничения, сравнения.** Пользователю дан объект управления – вентилятор. Вентилятор имеет **5 режимов** работы в зависимости от температуры: **1 режим** – скорость вентилятора равна 0, при температуре от 0 до 20  $^{\circ}\text{C}$  включительно (рис. 2.42); **2 режим** – скорость вентилятора равна 1, при температуре от 21 до 40  $^{\circ}\text{C}$  включительно; **3 режим** – скорость вентилятора равна 2, при температуре от 41 до 60  $^{\circ}\text{C}$  включительно (рис. 2.43); **4 режим** – скорость вентилятора равна 3, при температуре от 61 до 80  $^{\circ}\text{C}$  включительно; **5 режим** – скорость вентилятора равна 4, при температуре от 81 до 100  $^{\circ}\text{C}$  включительно.

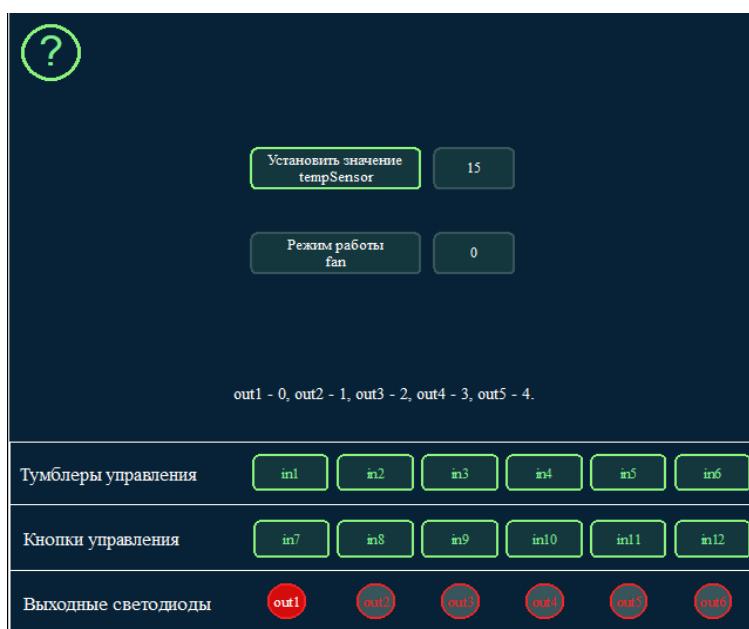


Рисунок 2.42 – Режим работы вентилятора при температуре 15  $^{\circ}\text{C}$

Температура ограничена и находится в промежутке от 0 до 100  $^{\circ}\text{C}$ . Поставленную задачу необходимо реализовать с использованием операторов, изученных ранее.

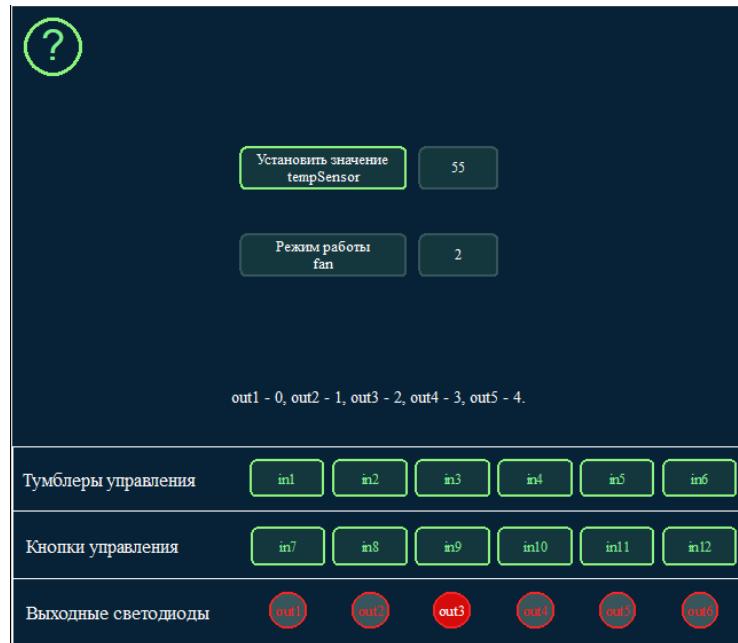


Рисунок 2.43 – Режим работы вентилятора при температуре 55  $^{\circ}\text{C}$

**Дополнительное задание.** В зависимости от режимов работы вентилятора выходные элементы out1-out5 (светодиоды) меняют свое состояние: out1 загорается при первом режиме работы, out2 – при втором, out3 – при третьем, out4 – при четвёртом, out5 – при пятом (рис. 2.4–2.43).

#### Требования к содержанию отчёта:

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Операторы выбора и ограничения CoDeSys.
- 4) Операторы сравнения CoDeSys.
- 5) Работа с входами и выходами ПЛК210.
- 6) Реализация конфигурирования объектом управления (вентилятором) в зависимости от влияющих факторов (температуры) с использованием операторов выбора и ограничения, сравнения.
- 7) Вывод.

#### Контрольные вопросы:

- 1) Операторы выбора и сравнения CoDeSys: описание, структура, обозначение, примеры.

2) Операторы сравнения CoDeSys: описание, структура, обозначение, примеры.

3) Какие типы данных имеют входы и выходы у операторов выбора и ограничения?

4) Какие типы данных имеют входы и выходы у операторов сравнения?

5) Особенности операторов «Мультиплексор» и «Селектор».

## 2.5 Настройка связи между ПЛК и CODESYS. Методы отладки программы

*Цель работы:* ознакомиться и изучить настройку связи между ПЛК и CODESYS, методы отладки программы.

### Теоретическая часть

**Настройка связи между ПЛК и CODESYS.** Узел «Device» (который определяется соответствующим таргет-файлом) должен соответствовать модели подключаемого контроллера. Если был выбран неподходящий тип контроллера – то его можно изменить, нажав в дереве проекта правой кнопкой мыши на узел «Device» и использовать команду «Обновить устройство» (рис. 2.44).

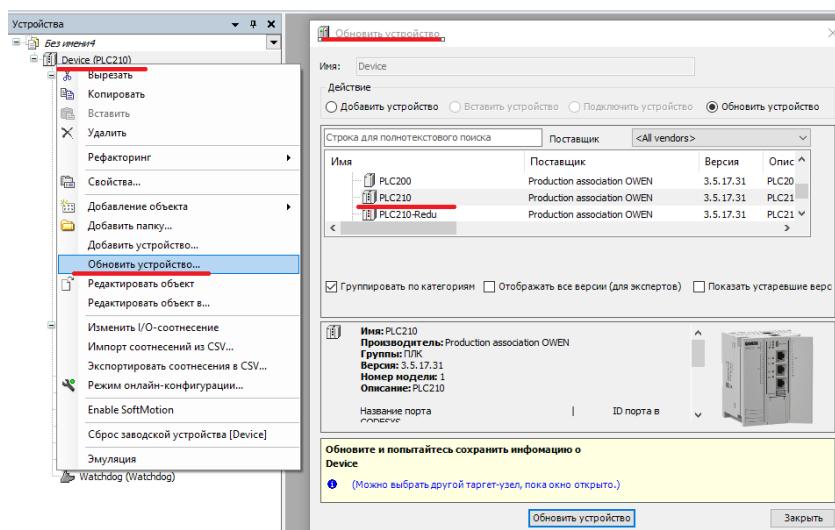


Рисунок 2.44 – Внешний вид окна обновления устройства

Затем следует выбрать из списка нужный таргет-файл. Напомним, что его версия (как и версия среды CODESYS) должна соответствовать версии прошивки контроллера (если вы не знаете эти версии – то используйте «выбира-тор» на сайте ОВЕН). По умолчанию отображаются только самые новые версии установленных таргет-файлов. Для отображения всех доступных версий установите галочку «Отображать все версии». После выбора таргет-файла

следует нажать кнопку «Обновить устройство» и закрыть окно (рис. 2.44). В дереве проекта у узла «*Device*» отобразится название выбранного контроллера.

Затем следует настроить «*Gateway (шлюз)*». Настройка производится однократно – обычно она требуется только в том случае, если на этом ПК раньше не было установлено ни одной версии CODESYS. Два раза нажмите левой кнопкой мыши на узел *Device* и перейдите на вкладку «Установки соединения». Нажмите на кнопку «*Gateway*» и выберите пункт «Добавить gateway» (рис. 2.45).

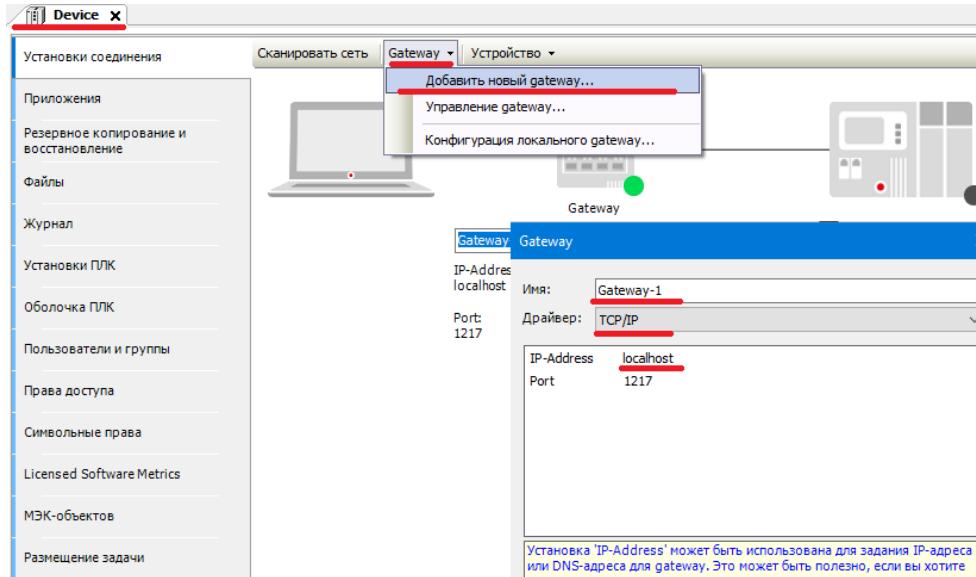


Рисунок 2.45 – Добавление нового gateway (шлюза)

Настройки рекомендуется оставить по умолчанию (имя – *Gateway-1*, IP-адрес – *localhost*). Закройте окно настроек шлюза и нажмите кнопку «*Сканировать сеть*» (рис. 2.46). В появившемся списке следует выбрать нужный контроллер и установить связь, нажав кнопку «*OK*».

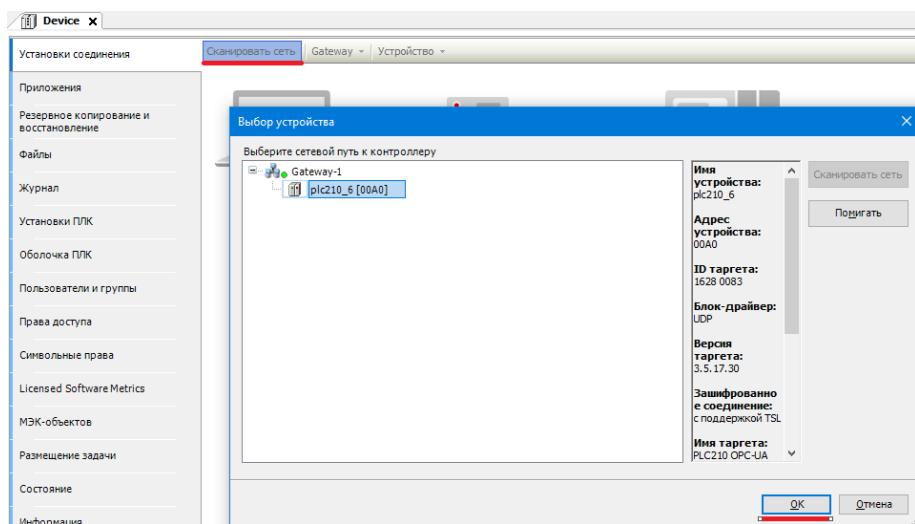


Рисунок 2.46 – Окно сканирования сети

В случае успешной установки связи индикаторы шлюза и контроллера загорятся зеленым (рис. 2.47).

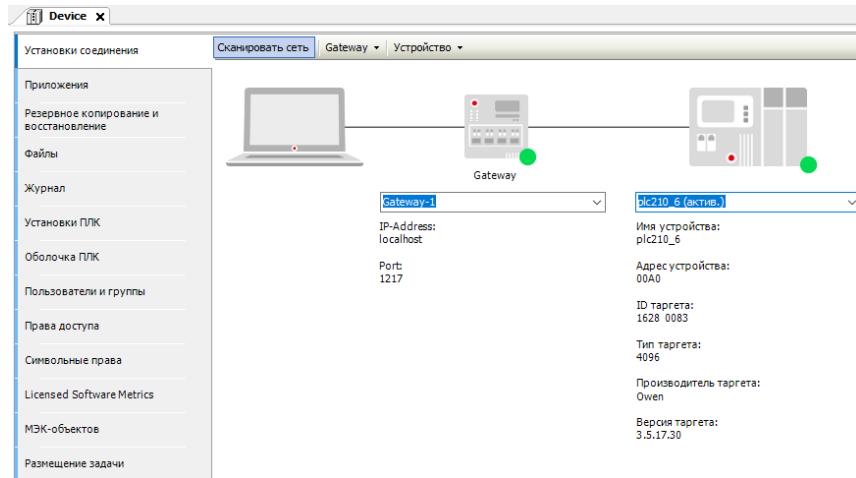


Рисунок 2.47 – Результат успешной установки связи

В ряде случаев (например, на ПК запущен антивирус, который блокирует рассылку широковещательных UDP пакетов или получение пакетов по портам 1740-1743) контроллер может не определиться во время сканирования сети. Тогда следует ввести IP-адрес контроллера вручную и нажать клавишу на клавиатуре «Enter» (рис.2.48). После успешного подключения можно переходить к загрузке проекта.

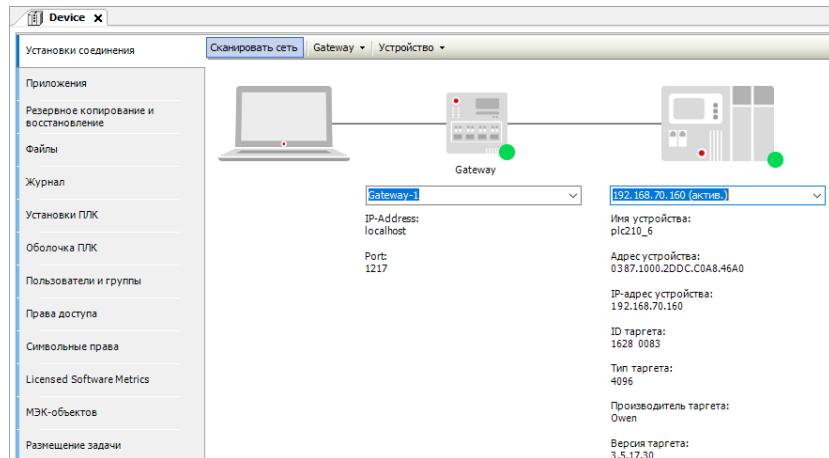


Рисунок 2.48 – Ввод IP-адреса контроллера для подключения без сканирования сети

**Методы отладки программы.** Меню «Отладка» (рис. 2.49) содержит команды для анализа ошибок в коде проекта. Большинство команд доступны только при подключении к контроллеру. Наиболее часто используемые команды:

- «Старт» – команда запуска приложения, обычно выполняется пользователем сразу после загрузки нового приложения;

- «Команды, связанные с точками останова» – позволяют установить в редакторах кода проекта точки останова. При достижении этих точек приложение будет автоматически останавливаться. Это позволяет, например, отследить, выполняются ли определенные ветви кода, определить, какие значения имеют переменные проекта в моменты выполнения определенных условий и т. д.;

- «Команды, связанные с шагами» – если приложение остановлено в точке останова, то с помощью этих команд можно выполнять его фрагментарно, отдельно «проходя» через каждую строку кода;

- «Записать значения» – эта команда позволяет присвоить переменным проекта заданные пользователем значения. Если значение переменной циклически изменяется в самом проекте (например, эта переменная привязана к входу контроллера или подключенного к контроллеру модуля ввода, или же в программе ей присваивается значение другой переменной), то команда «Записать значения» не сработает – вместо нее нужно использовать команду «Фиксировать значения»;

- «Фиксировать значения» – принудительно циклически записывает в переменную заданное пользователем значение. Пример использования – переменная привязана к аналоговому входу контроллера, к которому еще не подключен датчик, и поэтому она имеет значение 0.0. Чтобы сымитировать получаемое от датчика значение – необходимо использовать команду «Фиксировать значения» (если использовать команду «Записать значения», то записанное пользователем значение сразу будет сброшено в 0.0, так как опрос датчика происходит циклически);

- «Освободить значения» – эта команда отключает фиксацию значений выбранных переменных;

- «Режим отображения» – переключает режим отображения целочисленных переменных (доступные режимы: BIN/DEC/HEX).

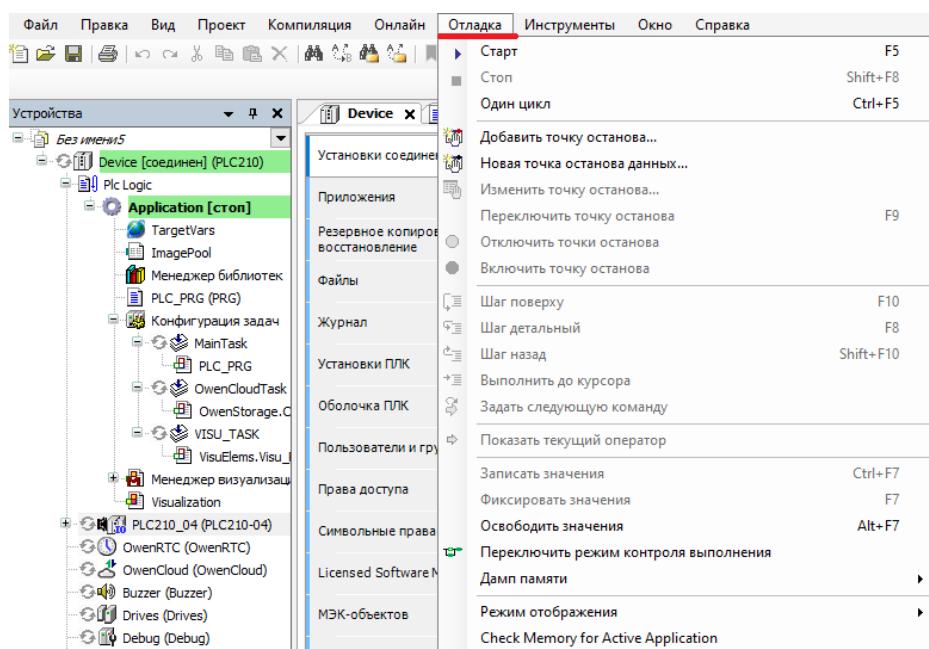


Рисунок 2.49 – Команды меню «Отладка»

Вместе с командами меню «Отладка» удобно использовать дополнительные панели, которые можно открыть в меню «Вид» (рис. 2.50):

- «Просмотр – Watch» – позволяет создать список переменных для наблюдения за их значениями (с возможностью изменения при помощи команд «Записать значения» и «Фиксировать значения»). Переменные могут быть объявлены в разных программных объектах проекта;
- «Просмотр – Наблюдение всех фиксаций» – отображает все переменные проекта, находящиеся в состоянии фиксации;
- «Точки останова» – отображает все точки останова проекта.

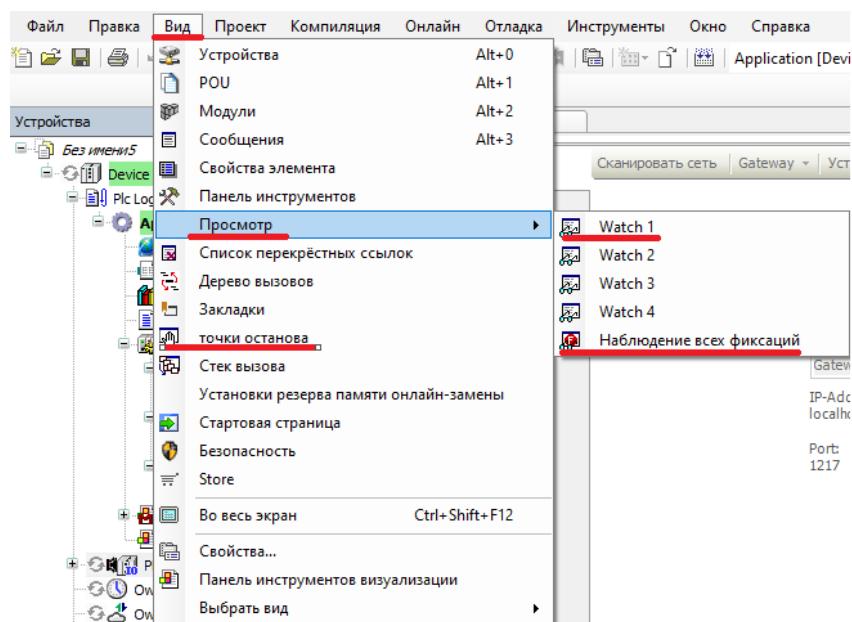


Рисунок 2.50 – Дополнительные панели меню «Вид»

## Практическая часть

**Ход работы.** Откройте предыдущий проект (*Реализация конфигурирования объектом управления (вентилятором) в зависимости от влияющих факторов (температуры) с использованием операторов выбора и ограничения, сравнения.*). Настройте связь между ПЛК и CODESYS. Инициализируйте переменные (привяжите их к реальным входам/выходам ПЛК). Загрузите проект в ПЛК. Поработайте с методами отладки программ.

### Требования к содержанию отчёта:

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Настройка связи между ПЛК и CODESYS.
- 4) Методы отладки программы.

5) Реализация конфигурирования объектом управления (вентилятором) в зависимости от влияющих факторов (температуры) с использованием операторов выбора и ограничения, сравнения.

6) Вывод.

### **Контрольные вопросы:**

- 1) Принципы и отличительные особенности настройки связи между ПЛК и CODESYS.
- 2) Как добавить/подключить устройство к проекту?
- 3) Какие существуют методы отладки программ?
- 4) Особенность команды «Старт».
- 5) Отличие команды «Записать значение» от «Фиксировать значение».

### **3 РАБОТА В СРЕДЕ ПРОГРАММИРОВАНИЯ «КОНФИГУРАТОР СП300»**

#### **3.1 Конфигурирование панели СП3xx**

*Цель работы:* ознакомиться с панелью оператора с сенсорным экраном СП3xx, ознакомиться и изучить методику конфигурирования панели СП3xx.

#### **Теоретическая часть**

Панель оператора с сенсорным экраном СП3xx представляет собой устройство класса «человеко-машинный интерфейс», предназначенное для загрузки управляющей программы (проекта) функционирования программируемого логического контроллера (ПЛК) или др. приборов, к которым подключается панель, мониторинга функционирования и редактирования значений параметров функционирования. Позволяет отображать на экране ход выполнения технологического процесса и редактировать значения параметров, отвечающих за функционирование системы. Логика работы панели СП3xx определяется пользователем в процессе конфигурирования на ПК с использованием программного обеспечения «Конфигуратор СП300».

Панель СП3xx предназначена для выполнения следующих функций:

- отображение состояния управляемого объекта в режиме реального времени, с использованием графических пиктограмм (индикаторы, графики, линейки, условные обозначения оборудования и т. д.);
- отображение сенсорных элементов, при помощи которых оператор осуществляет непосредственное управление функционированием объекта;
- управление функционированием ПЛК и/или других приборов; запись и чтение значений регистров ПЛК и/или других приборов, к которым подключается панель;
- оперативное изменение режима работы (изменение внешнего вида экрана и интерфейса управления, параметров управления и пр.) путем загрузки нового проекта;
- работа в режиме Master или Slave.

Проект функционирования управляемого объекта создается на ПК под конкретную задачу и загружается в энергонезависимую память панели. В процессе работы панели на экран выводятся сообщения о нештатных ситуациях для возможного вмешательства оператора.

На лицевой стороне панели (рис. 3.1) расположен сенсорный графический экран, чувствительный к прикосновениям. Сенсорный экран резистивного типа предназначен для ввода и отображения информации. Управление осуществляется путем нажатия на экран или перемещения по нему пальцем или другим удобным предметом, не наносящим повреждений экрану. Управляющие элементы представлены в диалоговых окнах экрана. Управление аналогично управлению с помощью механических клавиш.

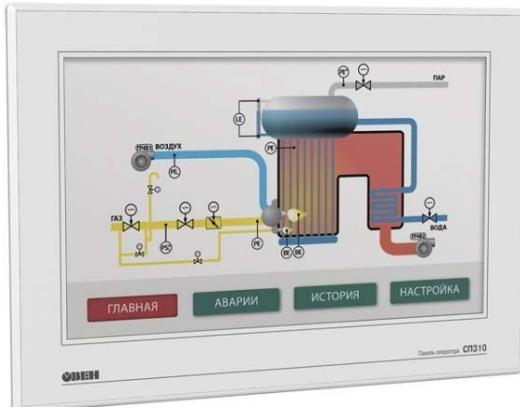


Рисунок 3.1 – Лицевая сторона панели СП3xx

Подключение всех внешних связей осуществляется через разъемные соединения (разъемы), расположенные на задней стороне корпуса (рис. 3.2). На задней стороне панели расположены два разъема DB-9 «PLC» и «Download», USB – B Device и, у расширенных модификаций, Ethernet и USB – A Host. На разъем «PLC» выведены контакты интерфейсов RS-485 и RS-232, предназначенные для подключения приборов. На разъем «Download» выведены контакты интерфейсов RS-485 и RS-232, предназначенные для подключения приборов и загрузки рабочего проекта в панель в режиме принудительной загрузки. Разъем USB – B предназначен для загрузки рабочего проекта в стандартном режиме работы. Разъем USB – A предназначен для подключения USB flash накопителей для ведения архивов, а также импорта файлов (например, рецептов) и загрузки проектов. Разъем Ethernet предназначен для подключения к панели других устройств по протоколу Modbus TCP. На задней стороне корпуса панели расположены клеммы для подключения питания (24 В) и функционального заземления (FG).



Рисунок 3.2 – Задняя сторона панели СП3xx

При работе применяется программа «Конфигуратор СП300» (рис. 3.3), при помощи которой осуществляется запись проекта в панель. Конфигуратор СП300 – это программа, предназначенная для конфигурирования панелей опе-

ратора СП3xx. Она работает под управлением операционных систем MS Windows XP/Vista/7/8/10.

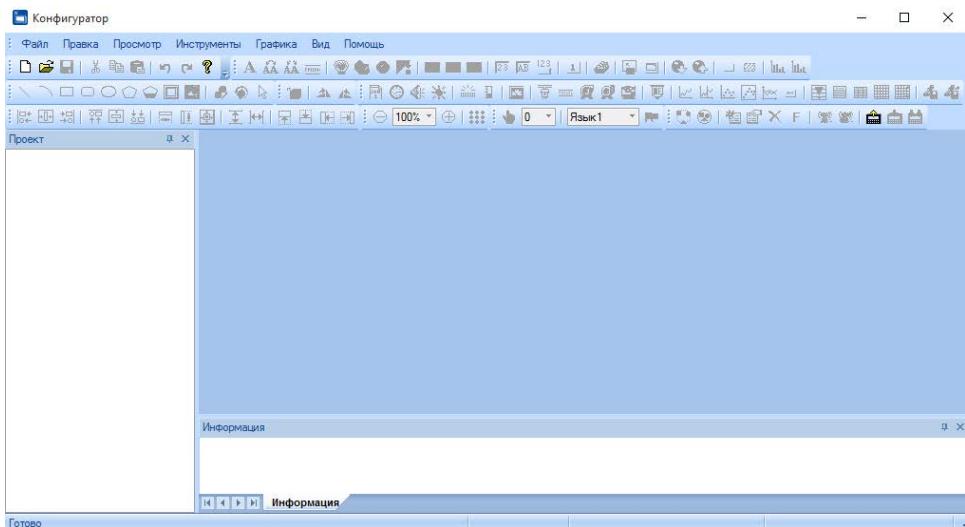


Рисунок 3.3 – Стартовое окно программы «Конфигуратор СП300»

Конфигуратор позволяет формировать и сохранять пользовательские экраны, отображаемые на экране панели в процессе эксплуатации. Пользовательские экраны включают наборы базовых элементов, каждый из которых позволяет выполнять определенную задачу проекта. К базовым элементам экрана относятся буквы (русские или английские) и символы, пиктограммы (индикатор, график, линейка, регистр для отображения / редактирования данных, кнопки переключения экранов и другие элементы). Совокупность экранов образует проект, который можно загрузить в панель или сохранить в виде файла на жестком диске компьютера. После загрузки проекта панель сразу начнет опрашивать регистры ПЛК или других приборов и отображать их значения на дисплее.

Для создания нового проекта необходимо нажать на кнопку «Новый», расположенную в меню «Файл» (рис. 3.4).

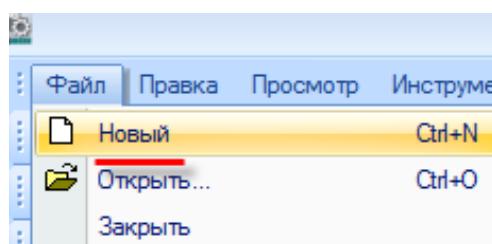


Рисунок 3.4 – Создание нового проекта

В открывшемся диалоговом окне «Панель» (рис. 3.5) необходимо выбрать модификацию панели и ориентацию экрана, после чего нажать кнопку «Далее».

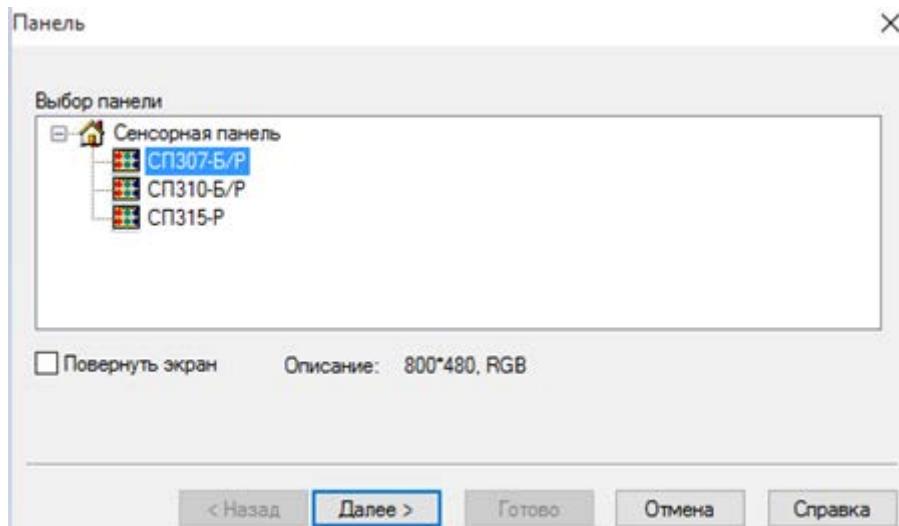


Рисунок 3.5 – Диалоговое окно выбора модификации панели

В диалоговом окне «Устройство» (рис. 3.6 и 3.7) необходимо выбрать режимы работы (Master/Slave) и настройки последовательных портов (вкладки «PLC порт» и «Download порт»), для расширенных модификаций панели (СП3xx-Р) также указать сетевые параметры (вкладка «Сетевые настройки»), после чего нажать кнопку «Далее».

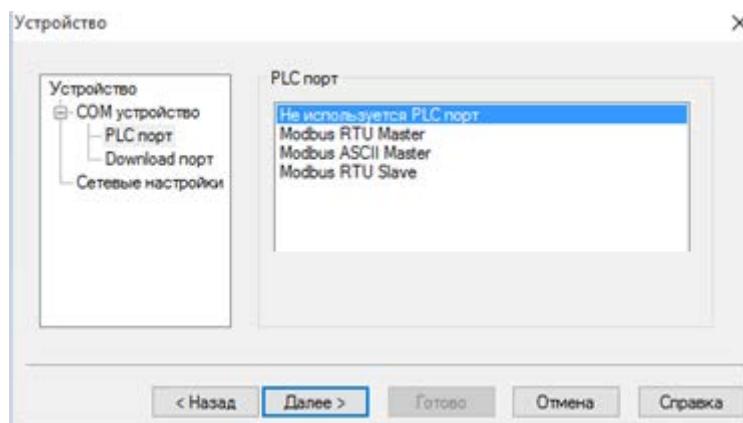


Рисунок 3.6 – Меню «Устройство», вкладка «СОМ устройство»

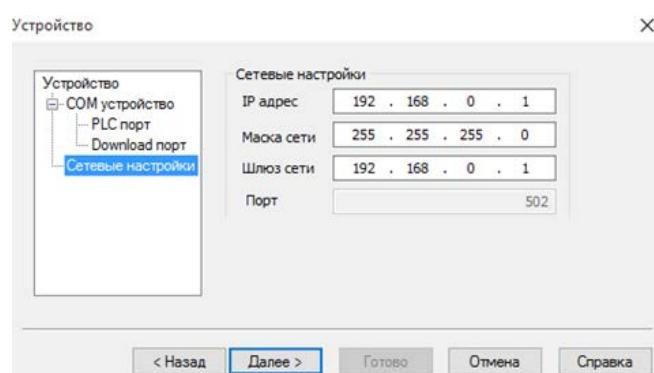


Рисунок 3.7 – Меню «Устройство», вкладка «Сетевые настройки»

В диалоговом окне «Проект» необходимо указать название проекта и опционально – автора и краткое описание (рис. 3.8).

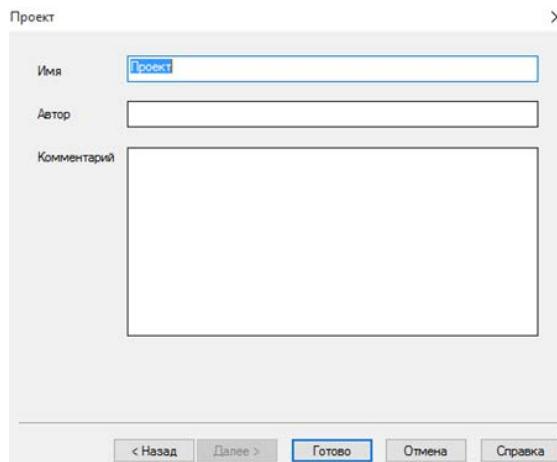


Рисунок 3.8 – Диалоговое окно «Проект»

После нажатия на кнопки «Готово» будет создан новый пустой пользовательский проект, содержащий один экран с названием [00001]Экран1. После создания пустого проекта окно конфигуратора будет выглядеть, как показано на рисунке 3.9.

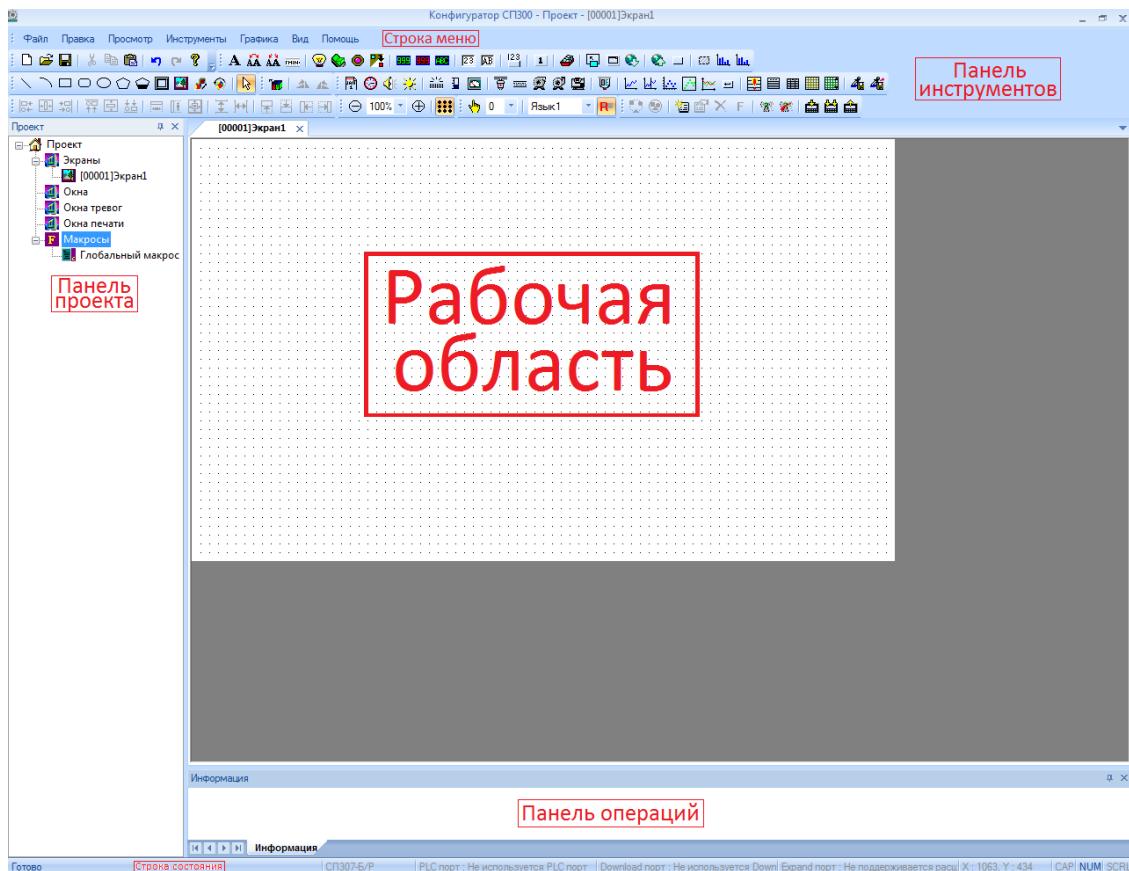


Рисунок 3.9 – Внешний вид интерфейса «Конфигуратор СП300»

На экране расположены следующие компоненты:

1. «Строка меню» содержит набор меню, используемых при работе над проектом.
2. «Панель инструментов» содержит набор ярлыков, дублирующих наиболее часто используемые пункты меню.
3. «Панель проекта» содержит древовидную структуру используемых в проекте экранов и макросов.
4. «Рабочая область» содержит открытые в данный момент компоненты. Используется для создания экранов оператора, настройки графических элементов, разработки макросов и т. д. Переключение между экранами может осуществляться с помощью вкладок, расположенных в верхней части рабочей области.
5. «Панель операций» содержит информацию о последних операциях, произведенных с элементами.
6. «Строка состояний» содержит информацию о текущем состоянии конфигуратора и портов панели.

Панель «Проект» содержит древовидную структуру используемых в проекте экранов и макросов (рис. 3.10).

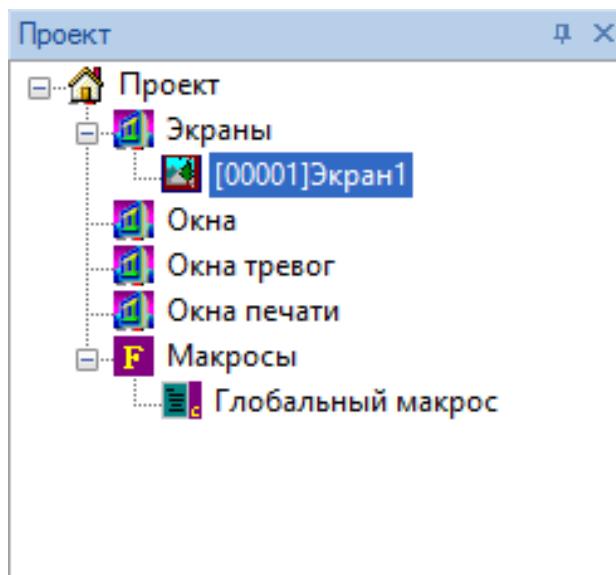


Рисунок 3.10 – Внешний вид панели проекты

Во вкладке «Проект» расположены следующие окна:

«Экраны» – обособленные области отображения графических элементов, размер которых соответствует размеру дисплея панели.

«Окна» – вспомогательные экраны, открываемые поверх основных. Таким экраном, например, может являться окно ввода уставок.

«Окна тревог» – вспомогательные экраны, открываемые поверх основных. В отличие от обычных окон, к этим окнам привязаны биты управления. Если бит принимает значение ВКЛ (1), то на дисплее панели отображается соответствующее окно. Если бит принимает значение ВЫКЛ (0), то окно не исчезает; для закрытия окна необходимо, чтобы его сквигировал оператор.

«Окно печати» – вспомогательные экраны, открываемые поверх основных. В отличие от обычных окон, к этим окнам привязаны биты, управляющие печатью их содержимого.

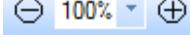
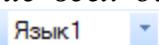
«Макросы» – пользовательские скрипты на языке С, привязываемые к графическим элементам «Функциональная кнопка»/«Функциональная область».

Ярлыки панели инструментов конфигуратора (рис. 3.11) дублируют команды меню.



Рисунок 3.11 – Ярлыки панели инструментов

Первая строка ярлыков содержит команды меню «Файл» (1–3), «Правка» (4–8), «Справка» (9) и команды добавления графических элементов из меню «Инструменты» (10–33). Вторая строка ярлыков содержит команды добавления статических графических элементов из меню «Графика» (1–8) и «Инструменты» (9–10, 15–40), «открытия библиотеки элементов» (12) и вращения элемента «Изображение» (13–14). Третья строка ярлыков содержит «команды операции с группами элементов» (1–11) и «перемещения элементов» (12–15).

Оставшиеся ярлыки:  – изменение масштаба экрана проекта;  – включение/отключение сетки;  – переключение всех битовых элементов;  – выбор отображаемого кадра анимации;  – выбор отображаемого языка проекта;  – включение/отключение отображения номе-ров регистров, привязанных к элементам;  – сделать элемент глобальным (отображается на всех экранах проекта «насквозь») /локальным (отображается только на данном экране);  – создание/переименование/удаление экрана или окна проекта;  – настройки шрифта элемента;  – включение offline/online эмуляция;  – загрузка проекта в панель/ загрузка про-екта с исходным кодом/ выгрузка про-екта из панели в конфигуратор.

Для отладки проекта необязательно загружать его в панель – можно воспользоваться режимом эмуляции и проверить работу проекта на ПК. «Offline эмуляция» позволяет проверить работу проекта без подключения внешних устройств; «Online эмуляция», помимо этого, позволяет проверить обмен со slave-устройствами (при условии, что они подключены к пользовательскому ПК или находятся в одной локальной сети с ним).

Работа макросов, некоторых системных регистров и протокола Modbus ASCII в эмуляции не поддерживается. Также эмуляция не поддерживает работу с PFW регистрами с номерами выше 4095. Время действия online эмуляции ограничено получасом, крайне рекомендуется отключить ее до истечения этого времени.

Память панели подразделяется на оперативную и энергонезависимую. Содержимое оперативной памяти не сохраняется после перезагрузки панели. Содержимое энергонезависимой памяти – сохраняется. Тип данных характеризует количество памяти, выделяемое под хранение значения. В то же время, формат данных определяет его представление (табл. 3.1).

Таблица 3.1 – Карта распределения адресов памяти панели

Название области памяти	PSB	PSW	PFW
Тип памяти	оперативная	оперативная	энергонезависимая
Тип данных	BOOL	WORD	WORD
Занимаемое место	1 бит	16 бит	16 бит
Диапазон доступных адресов	256 – 1023	256 – 4095	256 – 4095
Пример адресов	Адрес в панели Адрес по Modbus	PSB300 0x300	PSW300 4x300
Функции Modbus, применяемые к адресам	1 (0x01), 5 (0x05), 15 (0x0F)	3 (0x03), 6 (0x06), 16 (0x10)	4x10300

Адреса 0 – 255 каждой из областей памяти являются системными. Некоторые из них доступны для пользователя. Область памяти PCW используется для выполнения сервисных программ и не должна использоваться пользователем. Не рекомендуется без необходимости часто записывать данные в область PFW, так как энергонезависимая flash-память имеет ограниченный ресурс перезаписи.

В регистрах PSW/PFW могут храниться данные типа WORD и DWORD. Каждое значение WORD занимает один регистр. Каждое значение типа DWORD занимает два последовательно расположенных регистра (табл. 3.2).

Таблица 3.2 – Диапазон возможных значений для различных форматов данных типа WORD и DWORD

Диапазон возможных значений	WORD	DWORD
Dec	-32768...32767	-2147483648...2147483647
Hex	0...FFFF	0...FFFFFF
Float	не поддерживается	1.175494351 · 10 <sup>-(38)</sup> ... 3.402823466 · 10 <sup>(38)</sup>
Unsigned	0...65535	0...4294967295

В зависимости от модификации панелей оператора СП3xx могут меняться поддерживаемые протоколы передачи данных (табл. 3.3).

Таблица 3.3 – Список поддерживаемых протоколов панели СП3xx

Протокол	Master	Slave
Modbus RTU	СП3xx-Б/СП3xx-Р	СП3xxБ/СП3xx-Р
Modbus ASCII	СП3xx-Б/СП3xx-Р	
Modbus TCP	СП3xx-Р	СП3xx-Р

В случае, когда панель работает в режиме Modbus Master, ее элементы могут считывать/записывать значения из регистров других устройств (табл. 3.4).

Таблица 3.4 – Типы регистров протокола Modbus

Обозначение	Тип регистров	Тип данных	Тип доступа	Функции Modbus
0x	Coils	BOOL	чтение/запись	1 (0x01), 5 (0x05)
1x	Discrete inputs	BOOL	только чтение	2 (0x02)
3x	Input registers	WORD	только чтение	4 (0x04)
4x	Holding registers	WORD	чтение/запись	3 (0x03), 6 (0x06), 16 (0x10)

При опросе бит (0x и 1x) панель создает групповые запросы с кратностью адресов и количества опрашиваемых регистров в 8 бит. При опросе регистров (3x и 4x) панель создает групповые запросы, если разрыв между опрашиваемыми регистрами не превышает 4.

В режиме Modbus Slave все регистры панели являются holding регистрами, все биты – ячейками (coils). Функции 02 (Read Discrete Inputs) и 04 (Read Input Registers) не поддержаны.

## Практическая часть

**Ход работы.** Создать пустой проект с использованием программного обеспечения «Конфигуратор СП300». Ознакомьтесь с возможностями и функционалом программного обеспечения «Конфигуратор СП300».

### Требования к содержанию отчёта:

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Создание проекта с использованием программного обеспечения «Конфигуратор СП300» .
- 4) Возможности и функционал программного обеспечения «Конфигуратор СП300».
- 5) Вывод.

## Контрольные вопросы:

- 1) Для чего предназначена панель оператора СП3xx?
- 2) Какое программное обеспечение применяется для конфигурирования панели СП3xx?
- 3) Из чего состоит внешний вид интерфейса «Конфигуратор СП300»?
- 4) Какие окна содержатся во вкладке «Проект»?
- 5) Какие адреса содержит карта распределения адресов памяти панели?

## 3.2 Изучение принципов работы элементов «Переключатель» и «Индикатор»

*Цель работы:* ознакомиться и изучить работы элементов «Переключатель» и «Индикатор».

### Теоретическая часть

**ИНДИКАТОР.** Элемент Индикатор используется для отображения состояния привязанного бита (ВКЛ/ВЫКЛ) (рис. 3.12).

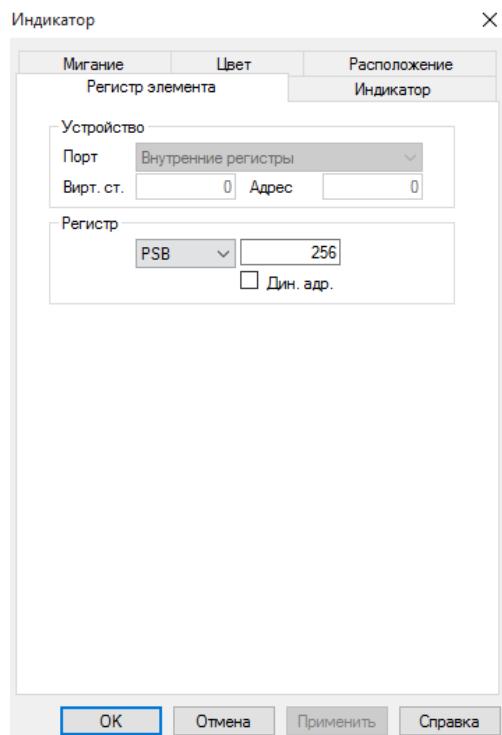


Рисунок 3.12 – Параметры элемента «Индикатор», вкладка «Регистр элемента»

Вкладка «Регистр элемента» содержит:

- 1) «*Порт устройства*, в котором храниться регистр, привязанный к данному элементу»:
  - внутренние регистры – к элементу привязан регистр панели;
  - PLC порт – к элементу привязан регистр устройства, подключенного к PLC-порту панели;
  - Download порт – к элементу привязан регистр устройства, подключенного к Download-порту панели;
  - <Название сетевого устройства> – к элементу привязан регистр устройства, подключенного к Ethernet порту панели (только для СП3xx-P).
- 2) «*Адрес*» – номер устройства (Slave ID).
- 3) «*Регистр*» тип (тип памяти панели или тип регистра Modbus) и номер регистра.

На вкладке «Расположение» задаются размеры элемента, его координаты и регистры, значения которых определяют перемещение элемента по горизонтальной и вертикальной оси в пикселях. Галочка «Закрепить» позволяет зафиксировать элемент на экране (аналогична одноименной команде контекстного меню).

Вкладка «*Индикатор*» содержит:

- 1) «*Вид*», которая включает в себя:
  - ВКЛ – настройка включенного состояния элемента;
  - ВЫКЛ – настройка отключенного состояния элемента;
  - Изменить – выбор изображения элемента из шаблонов элемента (шаблон определяет оба состояния);
  - Библиотека – выбор изображения элемента из библиотеки элементов (для каждого из состояний);
  - Сохранить – сохранение текущего изображения элемента в библиотеку элементов;
  - Скрыть кнопку – при наличии галочки, элемент становится невидим для пользователя, но при этом сохраняет свою функциональность.
- 2) «*Использовать текст*» – текст элемента (различный для состояний ВКЛ и ВЫКЛ), настройки его шрифта, стиля и выравнивания. При наличии галочки «Мультиязычный», текст элемента задается в меню «Языки».
- 3) «*Бит видимости*» – при наличии галочки можно выбрать бит, значение которого будет определять видимость элемента (ВЫКЛ – элемент невидим, ВКЛ – элемент виден).

Вкладка «*Мигание*» содержит:

- 1) «*Условие мигания*», которое включает:
  - Никогда – элемент никогда не мигает;
  - когда ВКЛ – элемент мигает во включенном (ВКЛ) состоянии;
  - когда ВЫКЛ – элемент мигает во включенном (ВЫКЛ) состоянии.
- 2) «*Скорость*»: скорость мигания индикатора.

**ПЕРЕКЛЮЧАТЕЛЬ.** Элемент «Переключатель» используется для изменения состояния выбранного бита (ВКЛ/ВЫКЛ). Уникальным параметром элемента «Переключатель» является вкладка «Действие» (рис. 3.13).

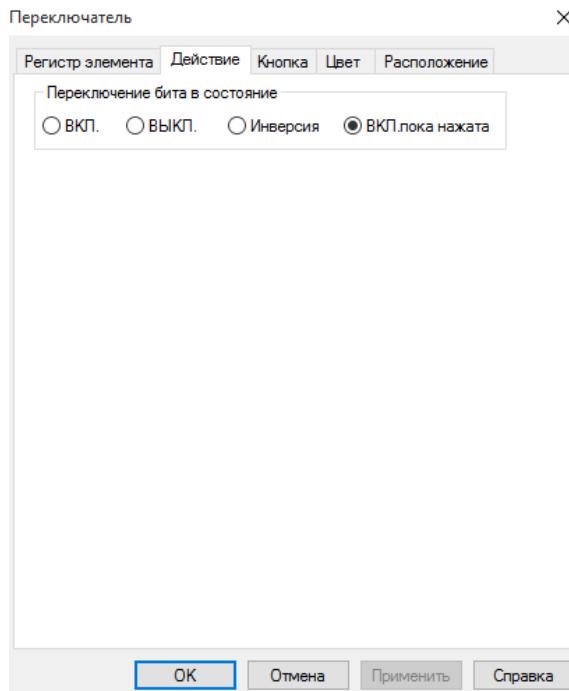


Рисунок 3.13 – Параметры элемента «Переключатель», вкладка «Действие»

Вкладка «Действие» содержит действие, выполняемое при нажатии на элемент:

- ВКЛ – переключение бита в состояние ВКЛ;
- ВЫКЛ – переключение бита в состояние ВЫКЛ;
- Инверсия – переключение элемента в состояние, противоположное текущему;
- ВКЛ, пока нажата – переключение бита в состояние ВКЛ на время нажатия элемента, после отпускания – переключение в состояние ВЫКЛ.

### Практическая часть

**Ход работы. «Работа с элементами «Индикатор» и «Переключатель».** Создайте новый пустой пользовательский проект. Расположите на «Экран1» три индикатора (💡 «Индикатор») и четыре переключателя (💡 «Переключатель»).

Настройка первого индикатора на работу при удержании кнопки управления. Настройки индикатора: «Порт» – «Внутренние регистры», «Регистр» – «PSB 256», «Индикатор» – заменить стандартное изображение индикатора.

Настройки соответствующего переключателя: «Порт» – «Внутренние регистры», «Регистр» – «PSB 256», «Переключение бита в состояние ВКЛ» – «пока нажата», кнопка «Заменить стандартное изображение индикатора», убрать галочку «Текст».

Настройка второго индикатора на переключение состояния по нажатию на управляющую кнопку с миганием во включенном состоянии. Настройки индикатора: «Порт» – «Внутренние регистры», «Регистр» – «PSB 257». «Мигание» – когда ВКЛ. «Индикатор» – заменить стандартное изображение индикатора.

Настройки соответствующего переключателя: «Порт» – «Внутренние регистры», «Регистр» – «PSB 257». Переключение бита в состояние «Инверсия». Кнопка «Заменить стандартное изображение индикатора», убрать галочку «Текст».

Настройка третьего индикатора на включение по нажатию на одну управляющую кнопку выключение и мигание по нажатию на другую. Настройки индикатора: «Порт» – «Внутренние регистры», «Регистр» – «PSB 258». «Мигание» – когда ВЫКЛ. Индикатор «Заменить стандартное изображение индикатора».

Настройки соответствующего переключателя на выключение: «Порт» – «Внутренние регистры», «Регистр» – «PSB 258». Переключение бита в состояние «ВЫКЛ». Кнопка «Заменить стандартное изображение индикатора», заменить текст «Нажата/Отжата» на «ВЫКЛ».

Настройки соответствующего переключателя на включение: «Порт» – «Внутренние регистры», «Регистр» – «PSB 258». Действие «ВКЛ». Кнопка «Заменить стандартное изображение индикатора».

Пример расположения элементов изображен на рисунке 3.14.



Рисунок 3.14 – Расположение элементов при изучении принципов работы «Переключатель» и «Индикатор»

Запустите  «offline эмуляцию» и исследуйте принцип работы элементов (рис. 3.15). По окончанию работы, закройте окно эмуляции.



Рисунок 3.15 – Эмуляция работы при изучении принципов работы «Переключатель» и «Индикатор»

**Требования к содержанию отчёта:**

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Элемент «Переключатель».
- 4) Элемент «Индикатор».
- 5) Реализация работы с элементами «Индикатор» и «Переключатель».
- 6) Вывод.

**Контрольные вопросы:**

- 1) Какие параметры элемента «Индикатор» содержатся во вкладке «Регистр элемента»?
- 2) Какие параметры элемента «Переключатель» содержатся во вкладке «Регистр элемента»?
- 3) Какие параметры элемента «Переключатель» содержатся во вкладке «Действия»?
- 4) Какие параметры элемента «Индикатор» содержатся во вкладке «Индикатор»?
- 5) Какие параметры элемента «Индикатор» содержатся во вкладке «Мигание»?

### 3.3 Изучение принципов работы с численными элементами

*Цель работы:* ознакомится и изучить работы элементов «Цифровой дисплей» и «Цифровой ввод».

#### Теоретическая часть

**ЦИФРОВОЙ ДИСПЛЕЙ.** Элемент «Цифровой дисплей» используется для отображения численного значения привязанного регистра (рис. 3.16).

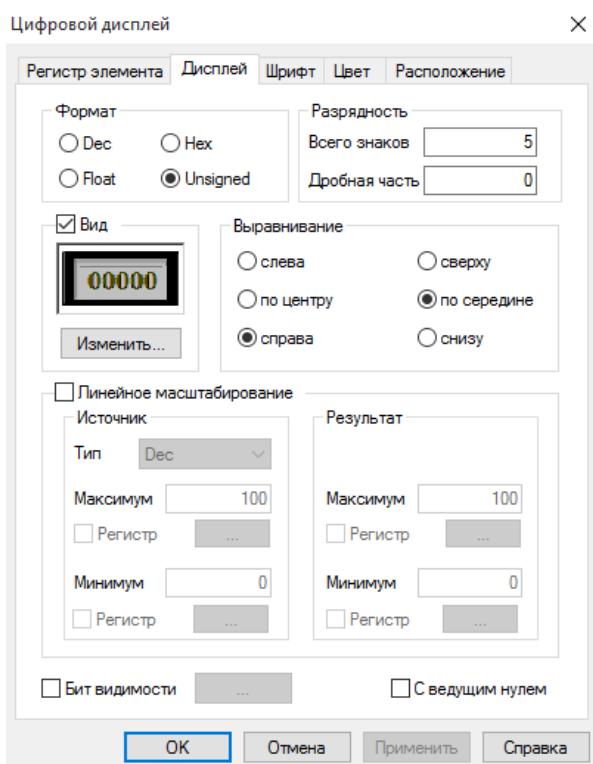


Рисунок 3.16 – Параметры элемента «Цифровой дисплей», вкладка «Дисплей»

Рассмотрим уникальные параметры элемента «Цифровой дисплей». Вкладка «Действие» содержит:

- 1) поле «Формат» – формат данных отображаемых элементом:
  - Dec – целочисленный знаковый;
  - Float – с плавающей точкой;
  - Hex – шестнадцатеричный;
  - Unsigned – целочисленный беззнаковый (только положительные значения).
- 2) Поле «Разрядность» – полное количество знаков, отображаемых элементом, и количество знаков, отображаемых после запятой.

3) Поле «*Вид*» – при отсутствии галочки, элемент не имеет фона. При наличии – можно выбрать фоновое изображение из шаблонов элемента с помощью кнопки «Изменить».

4) Поле «*Выравнивание*» – горизонтальное и вертикальное выравнивание области вывода элемента.

5) Поле «*Линейное масштабирование*» – для исходных данных (значений регистра, привязанного к элементу), указывается тип и верхний/нижний предел. Для результата (отображаемого значения) указывается верхний/нижний предел. Вместо фиксированных значений можно указать соответствующие регистры.

6) Поле «*Бит видимости*» – при наличии галочки можно выбрать бит, значение которого будет определять видимость элемента (ВЫКЛ – элемент невидим, ВКЛ – элемент виден).

7) Поле «*С ведущим нулем*» – при наличии галочки, перед значением регистра отображаются ведущие нули.

**ОПЕРАЦИИ С ДАННЫМИ.** Элемент «*Операции с данными*» используется для выполнения арифметических операций со значением регистра по нажатию на элемент. Рассмотрим уникальные параметры элемента «*Операция с данными*» (рис. 3.17).

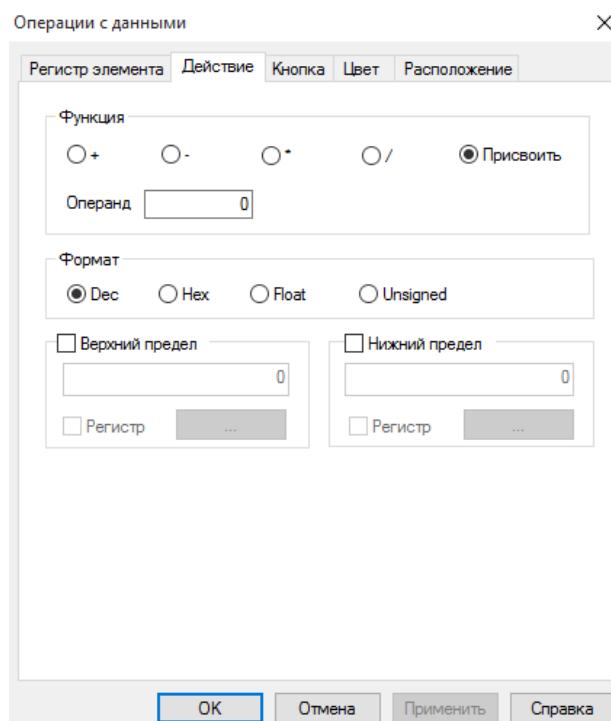


Рисунок 3.17 – Параметры элемента «Операции с данными», вкладка «Действие»

Вкладка «*Действие*» содержит:

1) Поле «Функция» – арифметическая операция, производимая над значением привязанного регистра с помощью задаваемого значения операнда.

2) Поле «Формат» – формат данных отображаемых элементом:

- Dec – целочисленный знаковый;
- Float – с плавающей точкой;
- Hex – шестнадцатеричный;
- Unsigned – целочисленный беззнаковый (только положительные значения).

3) Поле «Верхний/нижний предел» – верхний/нижний предел значений, которые можно ввести с помощью элемента.

## Практическая часть

**Ход работы. «Изучение принципов работы с численными элементами».** Создайте новый пустой пользовательский проект. Расположите на Экран1 Цифровой дисплей  и 5 элементов Операция с данными .

Настройка цифрового дисплея на отображение данных. Элемент «Цифровой дисплей»: «Порт – внутренние регистры», «Регистр – PSW 256», «Значение – Dword», «Формат – Float», «Разрядность – всего знаков 5, дробная часть 1», «Индикатор – заменить стандартное изображение индикатора».

Настройка 1-й кнопки, по нажатию на которую, устанавливается определенное значение регистра. Элемент «Операция с данными»: «Порт – внутренние регистры», «Регистр – PSW 256», «Значение – Dword», «Функция – Присвоить», «Операнд – 50.0», «Формат – Float», «Текст – Нач. Значение».

Настройка 2-й кнопки, по нажатию на которую, значение регистра уменьшается на 1. Элемент «Операция с данными»: «Порт – внутренние регистры», «Регистр – PSW 256», «Значение – Dword», «Функция – «–»», «Операнд – 1.0», «Формат – Float», «Верхний предел – поставить галочку и установить значение 100.0», «Нижний предел – поставить галочку и установить значение 0.0», «Текст – «–1»».

Настройка 3-й кнопки, по нажатию на которую, значение регистра уменьшается на 10. Элемент «Операция с данными»: «Порт – внутренние регистры», «Регистр – PSW 256», «Значение – Dword», «Функция – «–»», «Операнд – 10.0», «Формат – Float», «Верхний предел – поставить галочку и установить значение 100.0», «Нижний предел – поставить галочку и установить значение 0.0», «Текст – «–10»».

Настройка 4-й кнопки, по нажатию на которую, значение регистра увеличивается на 1. Элемент «Операция с данными»: «Порт – внутренние регистры», «Регистр – PSW 256», «Значение – Dword», «Функция – «+»», «Операнд – 1.0», «Формат – Float», «Верхний предел – поставить галочку и установить значение 100.0», «Нижний предел – поставить галочку и установить значение 0.0», «Текст – «+1»».

Настройка 5-й кнопки, по нажатию на которую, значение регистра увеличивается на 10. Элемент «Операция с данными»: «Порт – внутренние регистры

стры», «Регистр – PSW 256», «Значение – Dword», «Функция – «+»», «Операанд – 10.0», «Формат – Float», «Верхний предел – поставить галочку и установить значение 100.0», «Нижний предел – поставить галочку и установить значение 0.0», «Текст – «+10»».

Пример расположения элементов изображен на рисунке 3.18.

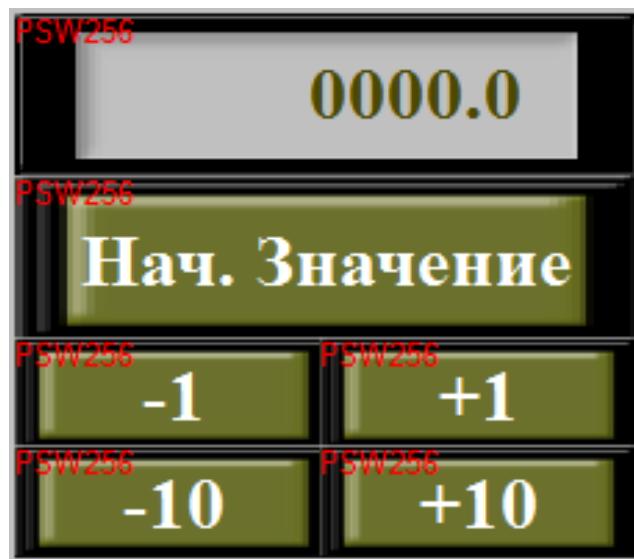


Рисунок 3.18 – Расположение элементов при изучении принципов работы с численными элементами

Запустите offline эмуляцию и исследуйте принцип работы элементов (рис. 3.19). По окончанию работы, закройте окно эмуляции.



Рисунок 3.19 – Эмуляция работы при изучении принципов работы с численными элементами

### **Требования к содержанию отчёта:**

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Элемент «Цифровой дисплей».
- 4) Элемент «Операция с данными».
- 5) Реализация изучения принципов работы с численными элементами.
- 6) Вывод.

### **Контрольные вопросы:**

- 1) Какие параметры элемента «Цифровой дисплей» содержатся во вкладке «Дисплей»?
- 2) Какие форматы данных отображаемых элементов содержит поле «Формат»?
- 3) Для чего необходимо поле «Разрядность» элемента «Цифровой дисплей»?
- 4) Какие параметры элемента «Операции с данными» содержатся во вкладке «Действие»?
- 5) Для чего необходимо поле «Функция» элемента «Операции с данными»?

### **3.4 Изучение принципов работы элементов визуального отображения численных значений**

*Цель работы:* изучение методики и принципов работы элементов визуального отображения численных значений.

#### **Теоретическая часть**

**ЛИНЕЙКА.** Элемент «Линейка» используется для визуального отображения численного значения привязанного регистра совместно со шкалой (рис. 3.20).

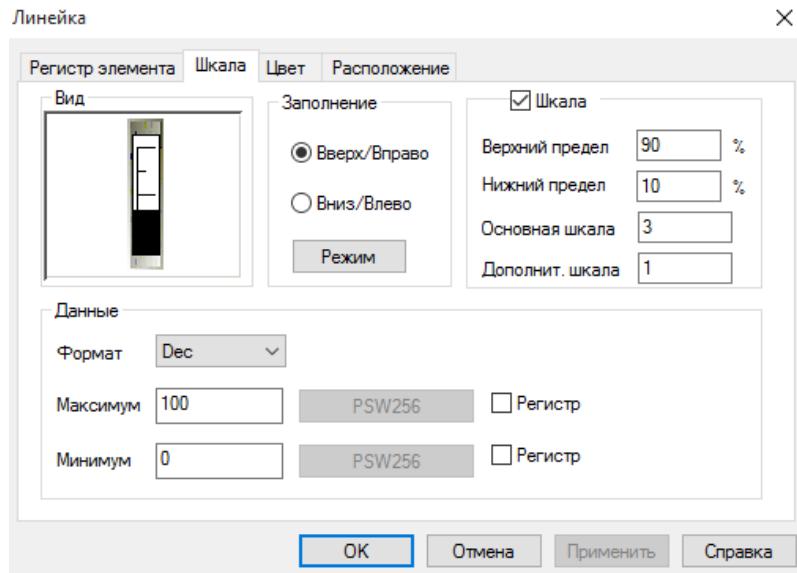


Рисунок 3.20 – Параметры элемента «Линейка», вкладка «Шкала»

Вкладка «Шкала» содержит:

1. Поле «*Вид*» – ориентация элемента (горизонтальная/вертикальная) и направление заполнения.
2. Поле «*Шкала*» – при наличии галочки, можно настроить границы шкалы (в процентах от размера элемента) и количество ее делений.
3. Поле «*Данные*» – тип отображаемых данных и верхний/нижний предел значений, отображаемых переменных. Вместо фиксированных значений можно указать соответствующие регистры.

**АНАЛОГОВЫЙ ДИСПЛЕЙ.** Элемент «*Аналоговый дисплей*» используется для отображения на экране панели значения привязанного регистра в аналоговом виде. Рассмотрим уникальные параметры элемента «*Аналоговый дисплей*» (рис. 3.21).

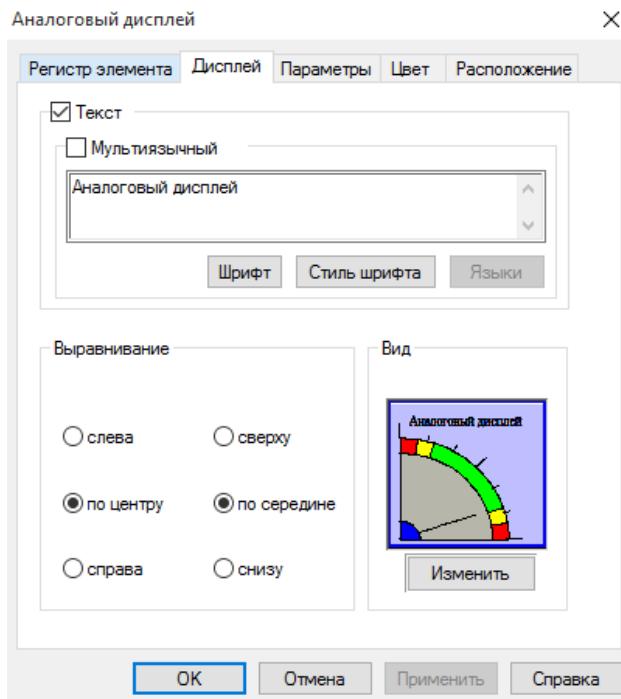


Рисунок 3.21 – Параметры элемента «Аналоговый дисплей», вкладка «Дисплей»

Вкладка «Дисплей» содержит:

- 1) Поле «Текст» – текст элемента, настройки его шрифта и стиля. При наличии галочки «Мультиязычный», текст элемента задается в меню «Языки».
- 2) Поле «Выравнивание» – горизонтальное и вертикальное выравнивание текста элемента.
- 3) Поле «Вид» – фоновое изображение элемента.

Вкладка «Параметры» содержит:

- 1) Поле «Формат» – формат отображаемых данных.
  - Dec – целочисленный знаковый;
  - Float – с плавающей точкой;
  - Hex – шестнадцатеричный;
  - Unsigned – целочисленный беззнаковый (только положительные значения).
- 2) Поле «Направление» – положение точки отсчета элемента.
- 3) Поле «Шкала» – толщина указателя и количество делений основной и дополнительной шкал.
- 4) Поле «Пределы» – выбор диапазона отображаемых значений, а также значений верхней/нижней границы предупреждения и аварии. Значения предупреждений и аварий задаются не диапазонами, а расстояниями от точки отсчета элемента.

## Практическая часть

**Ход работы.** «*Изучение принципов работы элементов визуального отображения численных значений*». Откройте проект, созданный в пункте 3.3. Расположите на Экран1 элемент «Линейка»  и элемент «Аналоговый дисплей» .

Настройка элемента «Линейка» на отображение данных: «Порт – внутренние регистры», «Регистр – PSW 256», «Значение – Dword», «Формат – Float», «Верхний предел – 100 %», «Нижний предел – 0 %», «Основная шкала – 10», «Дополнит. шкала – 1».

Настройка элемента «Аналоговый дисплей» на отображение данных: «Порт – внутренние регистры», «Регистр – PSW 256», «Значение – Dword», «Текст – убрать галочку», «Формат – Float», «Основная шкала – 10», «Дополнит. шкала – 1».

Пример расположения элементов изображен на рисунке 3.22.

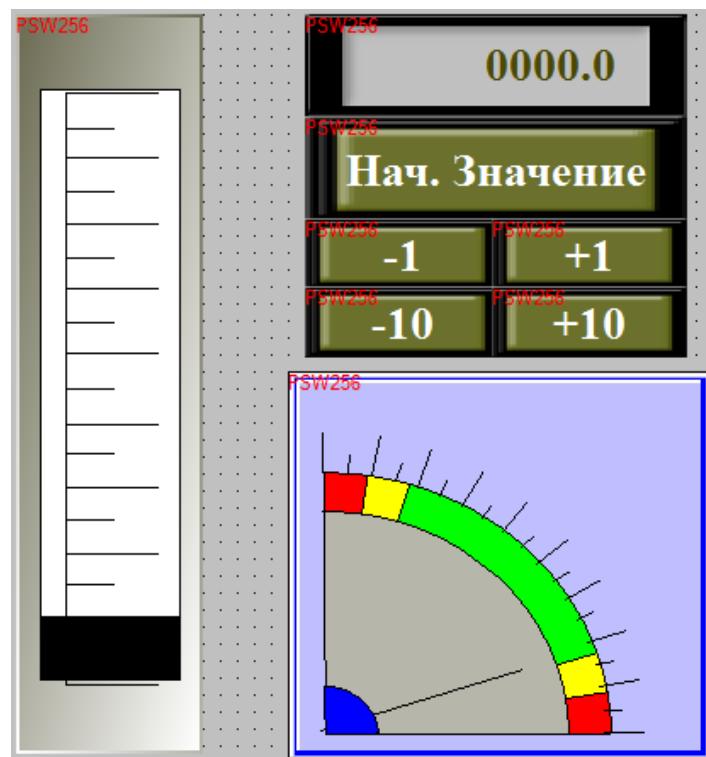


Рисунок 3.22 – Расположение элементов при изучении принципов работы с численными элементами

Запустите  offline эмуляцию и исследуйте принцип работы элементов (рис. 3.23). По окончанию работы закройте окно эмуляции.

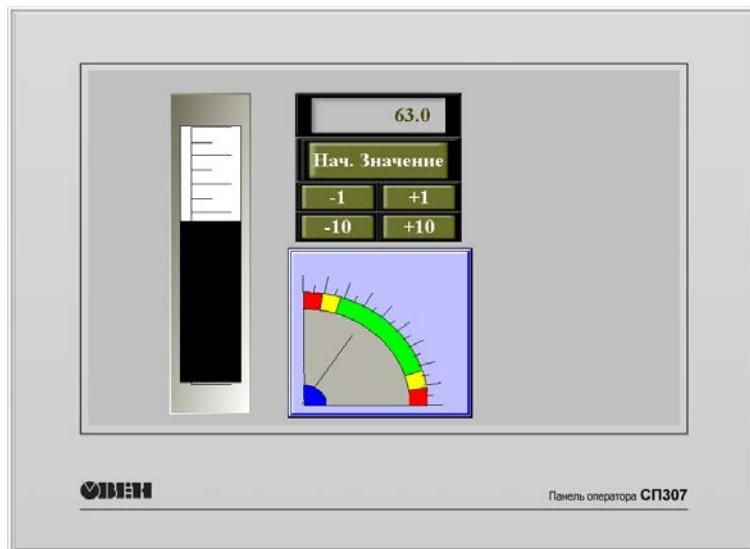


Рисунок 3.23 – Эмуляция работы при изучении принципов работы с численными элементами

**Требования к содержанию отчёта:**

- 1) Название лабораторной работы.
- 2) Цель работы.
- 3) Элемент «Линейка».
- 4) Элемент «Аналоговый дисплей».
- 5) Реализация изучения принципов работы элементов визуального отображения численных значений.
- 6) Вывод.

**Контрольные вопросы:**

- 1) Какие параметры элемента «Линейка» содержатся во вкладке «Шкала»?
- 2) Что содержит поле «Данные» элемента «Линейка»?
- 3) Какие параметры элемента «Аналоговый дисплей» содержатся во вкладке «Дисплей»?
- 4) Для чего необходимо поле «Пределы» элемента «Аналоговый дисплей»?

### 3.5 Изучение принципов работы панели оператора с контроллером.

*Цель работы:* изучение методики и принципов работы панели оператора с контроллером.

## Теоретическая часть

**Настройка ПЛК.** Проект для контроллера создан в среде CODESYS V3.5 SP17 Patch 3. Для контроллера используется таргет-файл версии 3.5.17.31. Посмотреть версию таргет-файла можно в узле «Device» на вкладке «Информация». На этой же вкладке отображаются ID COM-портов контроллера, которые будут использованы при настройке компонента Modbus COM.

Для изменения таргет-файла (например, при использовании другой версии прошивки или другой модели контроллера) следует нажать ПКМ на узел «Device» и выбрать команду «Обновить устройство». Таблица соответствия версий прошивок и таргет-файлов приведена в документе CODESYS V3.5. FAQ.

Для изменения модификации ПЛК (в примере используется модификация PLC210-01) следует нажать ПКМ на узел PLC210-XX и выбрать команду «Обновить устройство». В рамках примера для конвертации данных используется библиотека «OwenCommunication». Библиотека доступна для скачивания на официальном сайте ОВЕН в разделе CODESYS V3/Библиотеки и компоненты. Для установки библиотеки CODESYS следует выбрать пункт «Инструменты», открыть «Репозиторий библиотек», нажать кнопку «Установить» и выбрать скачанный файл библиотеки.

Затем следует добавить компонент «Modbus COM». Для этого следует нажать ПКМ на узел «Device», выбрать команду «Добавить устройство» и добавить компонент «Modbus COM», расположенный во вкладке «Промышленные сети» (fieldbus)/Modbus/Порт Modbus Serial.

После этого на вкладке «Общее» следует задать идентификатор и настройки COM-порта. В конкретном случае используется порт RS-485-1 с настройками 115200-8-N-1. Идентификатор данного порта – 5 (идентификаторы COM-портов отображаются на вкладке «Device – Информация»). Затем следует нажать ПКМ на узел «Modbus COM», выбрать команду «Добавить устройство» и добавить компонент «Modbus Serial Device», расположенный во вкладке «Промышленные сети» (fieldbus)/Modbus/Устройство Modbus Serial.

В настройках компонента на вкладке «Modbus Serial Device» можно задать настройки slave-устройства:

- ID – адрес slave-устройства (Slave ID), в соответствии с таблицей 1 задается равным 1;

- Сторожевой таймер – если установлена галочка, то при отсутствии опросов мастера в течение заданного таймаута происходит обнуление регистров slave'a;

- Регистры временного хранения – число holding регистров slave'a;
- Входные регистры – число input регистров slave'a;
- Обмотки – число coils slave'a;
- Дискретные входы – число discrete inputs slave'a.

**Обратите внимание**, что по умолчанию в реализации CODESYS значения holding регистров не могут быть изменены из программы контроллера (только со стороны master-устройства).

Чтобы holding регистры и coils стали доступны для изменения из программы контроллера, необходимо установить галочку «Запись».

**Обратите внимание**, что по умолчанию в реализации CODESYS области памяти битов и регистров наложены друг на друга (общие области для Coils/Holding регистров и Discrete Inputs/Input регистров).

Чтобы разделить указанные области, необходимо установить галочку «Дискретные битовые области». В рамках примера эта галочка установлена. На вкладке «Modbus Serial Device» «Соотнесение входов-выходов» следует привязать переменные программы. Настройка ПЛК завершена.

**Настройка СП3xx.** Для того чтобы использовать панель СП3xx в режиме «Modbus RTU Master» необходимо в настройках проекта на вкладке «Устройство» для нужного COM-порта выбрать режим «Modbus RTU Master» и указать его настройки. В соответствии с примером используется PLC-порт с настройками 115200-8-N-1. Настройка панели завершена.

## Практическая часть

**Ход работы. «Реализация совместной работы СП3xx и ПЛК».** Подключите контроллер ПЛК210-04 CS к панели оператора СП310. В контроллер загрузите соответствующую программу.

Создайте новый пустой пользовательский проект. Выполните настройки проекта: *СОМ устройство – Modbus RTU Master, скорость – 115200, бит данных 8 бит, контроль четности – отсутствует, изменить порядок регистров – установить галочку*.

**Настройка Экран 1.** Разместить 4 элемента Индикатор  для отображения состояния реальных входов контроллера ПЛК210-04 CS:

1-й индикатор: *Порт – PLC порт, Адрес – 1, Функция – 4x, Регистр – 0, Бит 00;*

2-й индикатор: *Порт – PLC порт, Адрес – 1, Функция – 4x, Регистр – 0, Бит 01;*

3-й индикатор: *Порт – PLC порт, Адрес – 1, Функция – 4x, Регистр – 0, Бит 02;*

4-й индикатор: *Порт – PLC порт, Адрес – 1, Функция – 4х, Регистр – 0, Бит 03.*

Разместить 4 элемента Переключатель с индикацией  для управления состоянием реальных выходов контроллера ПЛК210-04 CS:

1-й переключатель с индикацией, работающий в режиме тумблера: *Порт – PLC порт, Адрес – 1, Функция – 4х, Регистр – 0, Бит 16;.*

2-й переключатель с индикацией, работающий в режиме тумблера: *Порт – PLC порт, Адрес – 1, Функция – 4х, Регистр – 0, Бит 17;.*

3-й переключатель с индикацией, работающий в режиме тумблера: *Порт – PLC порт, Адрес – 1, Функция – 4х, Регистр – 0, Бит 18;.*

4-й переключатель с индикацией, работающий в режиме тумблера: *Порт – PLC порт, Адрес – 1, Функция – 4х, Регистр – 0, Бит 19.*

Разместить 2 элемента Цифрового дисплея  для отображения численных значений, переданных из контроллера ПЛК210-04 CS:

1-й цифровой дисплей: *Порт – PLC порт, Адрес – 1, Функция – 4х, Регистр – 2, Тип – DWord, Формат – Float, Всего знаков – 4, Дробная часть – 1;*

1-й цифровой дисплей: *Порт – PLC порт, Адрес – 1, Функция – 4х, Регистр – 4, Тип – DWord, Формат – Float, Всего знаков – 4, Дробная часть – 1.*

Пример расположения элементов изображен на рисунке 3.24.



Рисунок 3.24 – Расположение элементов на Экран1

Запустите  offline эмуляцию и исследуйте принцип работы элементов (рис. 3.25). По окончанию работы, закройте окно эмуляции.



Рисунок 3.25 – Эмуляция работы программы

## Литература

1. Денисенко, В. В. Компьютерное управление технологическим процессом, экспериментом, оборудованием / В. В. Денисенко. – Москва : Горячая линия-Телеком, 2009. – 608 с.
2. Петров, И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / И. В. Петров ; под ред. В. П. Дьяконова. – Москва : СОЛООН-Пресс, 2004. – 256 с.
3. Промышленные сети передачи данных / Р. К. Нургалиев [и др.] // Вестник Казанского технологического университета. – 2013. – Т. 16, № 1. – С. 252–255.

Учебное издание

# ПРОЕКТИРОВАНИЕ СИСТЕМ АВТОМАТИЗАЦИИ. ПРОГРАММИРОВАНИЕ ПРИБОРНОЙ АВТОМАТИКИ

## Лабораторный практикум

Науменко Андрей Михайлович  
Ринейский Константин Николаевич  
Самусев Артем Михайлович  
Тёмкин Даниил Александрович

Редактор *Р.А. Никифорова*  
Корректор *А.С. Прокопюк*  
Компьютерная верстка *А.М. Науменко*

---

Подписано к печати 01.09.2025. Усл. печ. листов 4.8.  
Уч.-изд. листов 6.0. Заказ № 164

Учреждение образования «Витебский государственный технологический университет»  
210038, г. Витебск, Московский пр., 72.  
Отпечатано на ризографе учреждения образования  
«Витебский государственный технологический университет».  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий № 1/172 от 12 февраля 2014 г.  
Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий № 3/1497 от 30 мая 2017 г.