**СОДЕРЖАНИЕ**

[1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_Toc105094487)

[1.1 Среда разработки – CODESYS 6](#_Toc105094488)

[1.1.1 О CODESYS 6](#_Toc105094489)

[1.1.2 Назначение и области применения 6](#_Toc105094490)

[1.1.3 Особенности CODESYS 7](#_Toc105094491)

[1.1.4 Основные различия CODESYS v3 и v2 7](#_Toc105094492)

[2 ПРАКТИЧЕСКАЯ РАБОТА 9](#_Toc105094493)

[2.1 Описание практической работы 9](#_Toc105094494)

[2.1.1 Название и номер практической работы 9](#_Toc105094495)

[2.1.2 Цель и задание работы 9](#_Toc105094496)

[2.2 Выполнение практической работы 9](#_Toc105094497)

[2.2.1. Создание стандартного проекта в Codesys 3.5 9](#_Toc105094498)

[2.2.2. Добавление визуализации в проект (Рисунок 2) 10](#_Toc105094499)

[2.2.3. Добавление примитивов визуализации 11](#_Toc105094500)

[2.2.5. Добавляем плейсхолдер в имя кнопки (Рисунок 8) 13](#_Toc105094501)

[2.2.6. Объявляем переменную 14](#_Toc105094502)

[2.2.8. Конфигурация действия при нажатии мыши на кнопку 15](#_Toc105094503)

[3 Московский завод «ФИЗПРИБОР» 25](#_Toc105094504)

[ЗАКЛЮЧЕНИЕ](#_Toc105094505)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ](#_Toc105094506)

**1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **1.1 Среда разработки – CODESYS**

### **1.1.1 О CODESYS**

На сегодняшний день CoDeSys (Controller Development System) — это самый популярный в мире аппаратно-независимый комплекс для прикладного программирования ПЛК и встраиваемых контроллеров. Основным его компонентом является среда программирования на языках стандарта МЭК 61131-3. Комплекс работает на компьютере. Программы компилируются в машинный код и загружаются в контроллер. Любую задачу, которая имеет решение в виде программы, можно реализовать в CoDeSys. [1]

### **1.1.2 Назначение и области применения**

Изначально CoDeSys был нацелен на задачи, требующие автономности, надежности и предельного быстродействия при минимизации аппаратных средств. Благодаря этому он вышел далеко за рамки традиционных для МЭК 61131-3 систем ПЛК. Сегодня автомобили, краны, экскаваторы, самосвалы, яхты, печатные машины, деревообрабатывающие станки, литейные и прокатные машины, сборочные автоматы крупнейших мировых брендов включают один или группу встроенных контроллеров с CoDeSys.

На сегодняшний день CoDeSys успешно применяется во всех без исключения областях промышленности. В мире более 350 компаний, изготавливают контроллеры с CoDeSys в качестве штатного инструмента программирования.

Как продукт, CoDeSys ориентирован на изготовителей контроллеров. Разрабатывая новый контроллер, они устанавливают в него систему исполнения CoDeSys Control. Собирают из ее компонентов требуемую конфигурацию, добавляют собственные специфические компоненты и получают собственное инструментальное ПО. [2]

### **1.1.3 Особенности CODESYS**

 В соответствии со стандартом МЭК 61131-3 CODESYS поддерживает 5 языков программирования:

* IL (Instruction List) – язык, по синтаксису схожий с языком низкого уровня Ассемблер.
* ST (Structured Text) – текстовый язык, похожий на Pascal.
* LD (Ladder Diagram) – язык релейно-лестничных схем.
* FBD (Function Block Diagramm) — язык функциональных блоков.
* SFC (Sequental Function Chart) – язык диаграмм, похожих на блок-схемы.

Кроме этих языков CODESYS включает в себя еще один язык – CFC (Continuous Function Chart). Он похож на FBD, но позволяет располагать функциональные блоки свободно на экране и задавать порядок их выполнения.

В CoDeSys V3 впервые в мире была реализована поддержка объектно-ориентированного программирования (ООП) в языках стандарта МЭК 61131-3.

### **1.1.4 Основные различия CODESYS v3 и v2**

* Поддержка элементов объектно-ориентированного программирования (ООП).
* Новый язык программирования UML (Unified Modelling Language), тесно связанный с ООП.
* Сети ПЛК — инструмент управления в одном проекте несколькими контроллерами.
* Управление системами движения (CODESYS SoftMotion).
* Оптимизация программного кода (сложные конструкции типа IF … END\_ IF можно «сворачивать» для упрощения просмотра кода).

# **2 ПРАКТИЧЕСКАЯ РАБОТА**

## **2.1 Описание практической работы**

### **2.1.1 Название и номер практической работы**

Практическая работа вариант номер 1: Основы работы с графикой Coodesys 3.5.

### **2.1.2 Цель и задание работы**

Изучение простейших графических примитивов в графическом редакторе Codesys 3.5 и их основных параметров. Ознакомится с графическим примитивом «Кнопка».

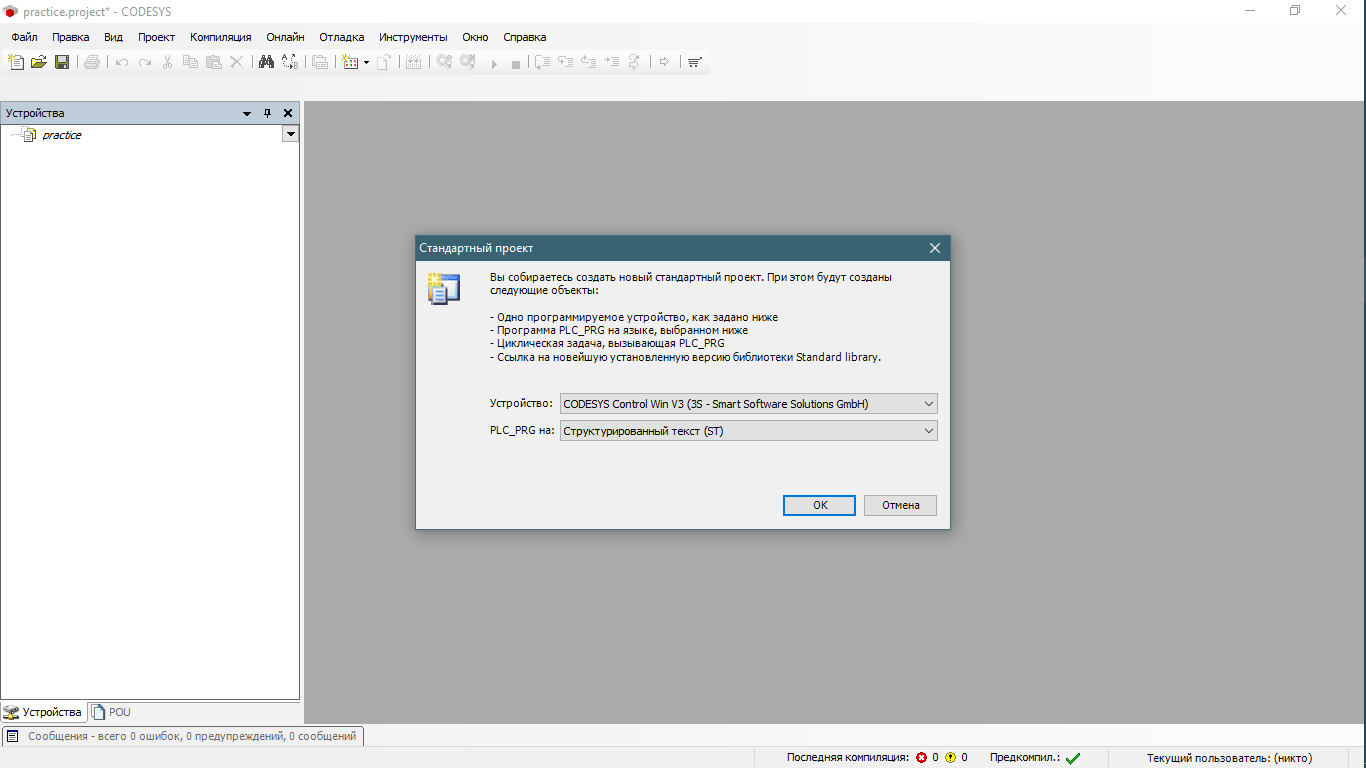
Заданием работы является создание в Codesys 3.5 графического интерфейса с одной кнопкой, при нажатии на которую, счетчик нажатий увеличивается. Счетчик должен быть изображен в виде цифр посредине кнопки.

Далее представлено поэтапное выполнение практической работы варианта №1, включающее изображения, наглядно демонстрирующие основные шаги выполнения практической работы варианта №1, а также пояснения или комментарии к каждому этапу работы.

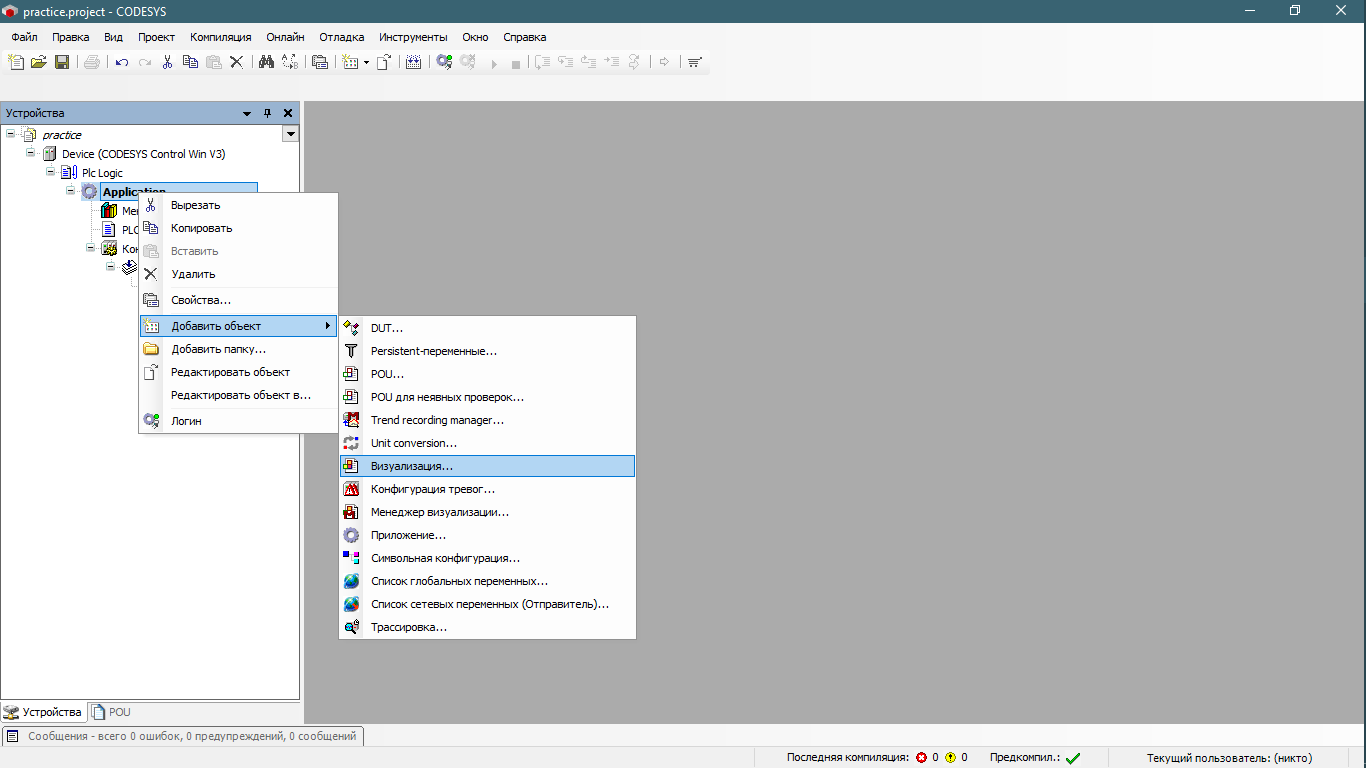
## **2.2 Выполнение практической работы**

### **2.2.1. Создание стандартного проекта в Codesys 3.5**

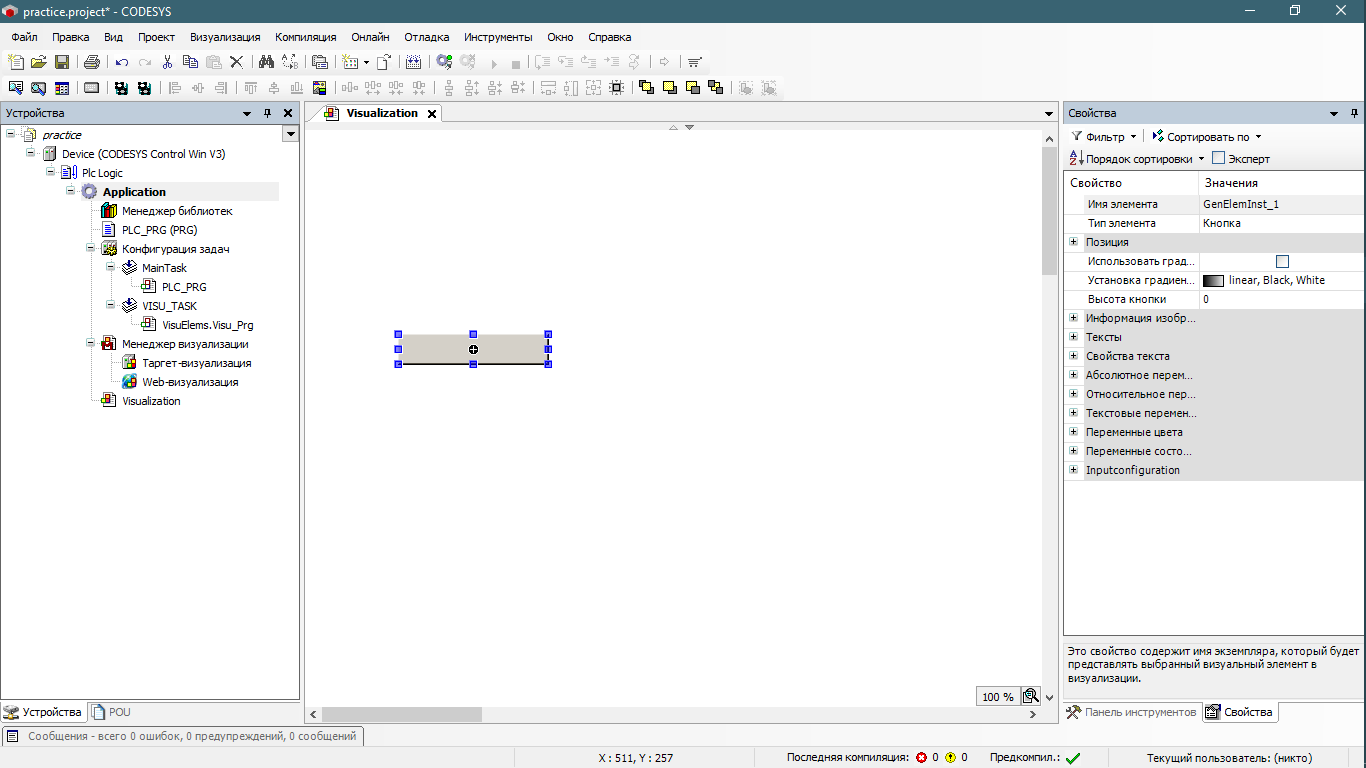
Выбираем устройство, как на Рисунке 1. Язык программирования укажите ST.

  
**Рисунок 1 — Первичные параметры стандартного проекта**

### **2.2.2. Добавление визуализации в проект (Рисунок 2)**

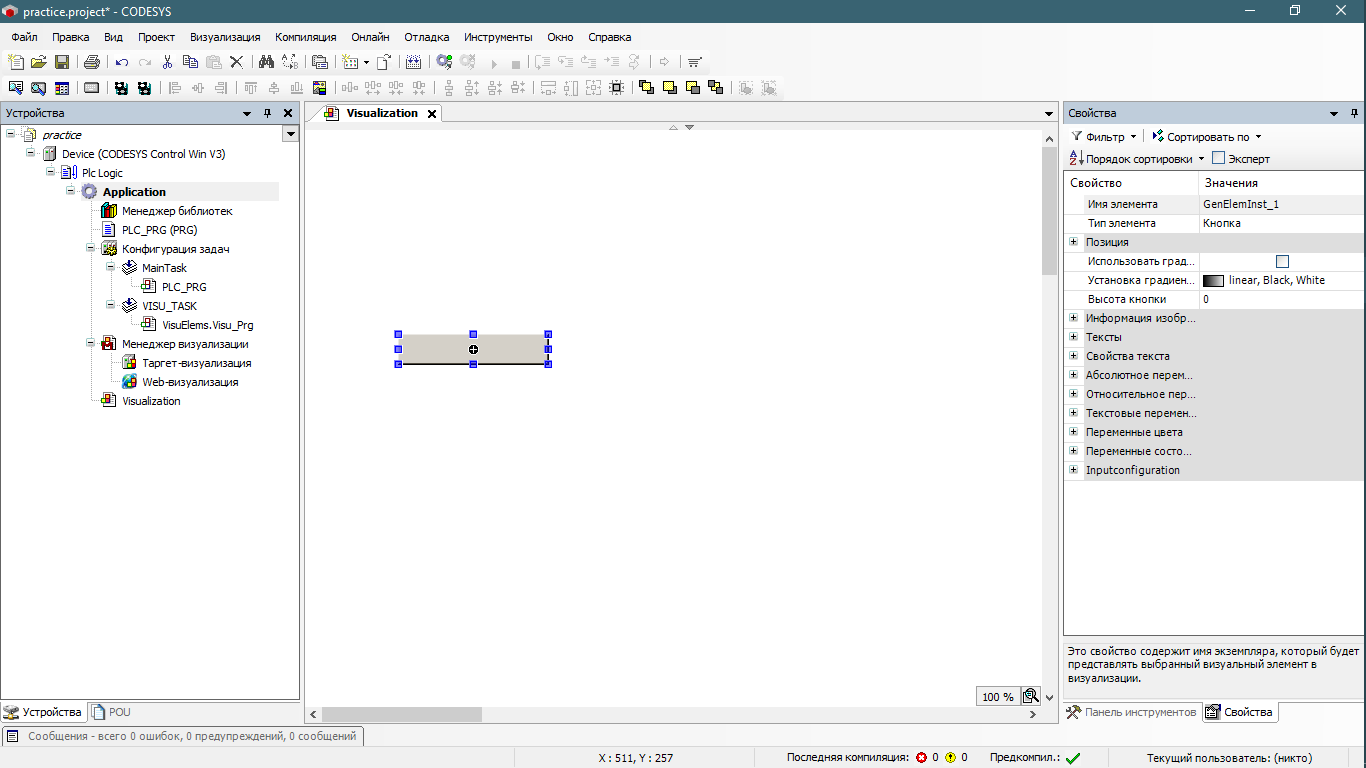
 **Рисунок 2 — Добавление элемента «Визуализация»**

После добавления визуализации в проект появляется пустая вкладка «Visualisation» [4]. Появление вкладки изображено на Рисунке 3.

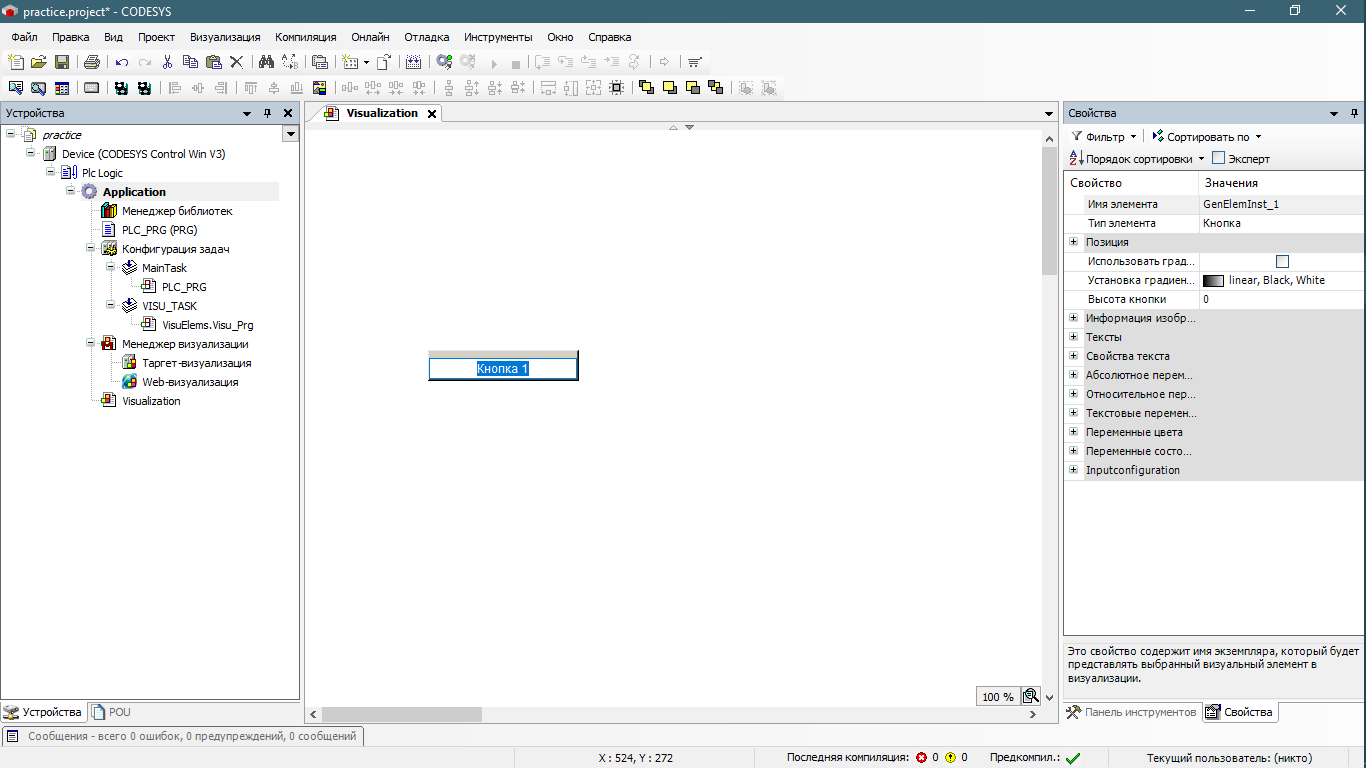
  
**Рисунок 3 — Вид пустой вкладки «Визуализация»**

### **2.2.3. Добавление примитивов визуализации**

В Панели инструментов визуализации – выберем примитив «Кнопка» и перетащим «Кнопку» на рабочую область (Рисунок 4).

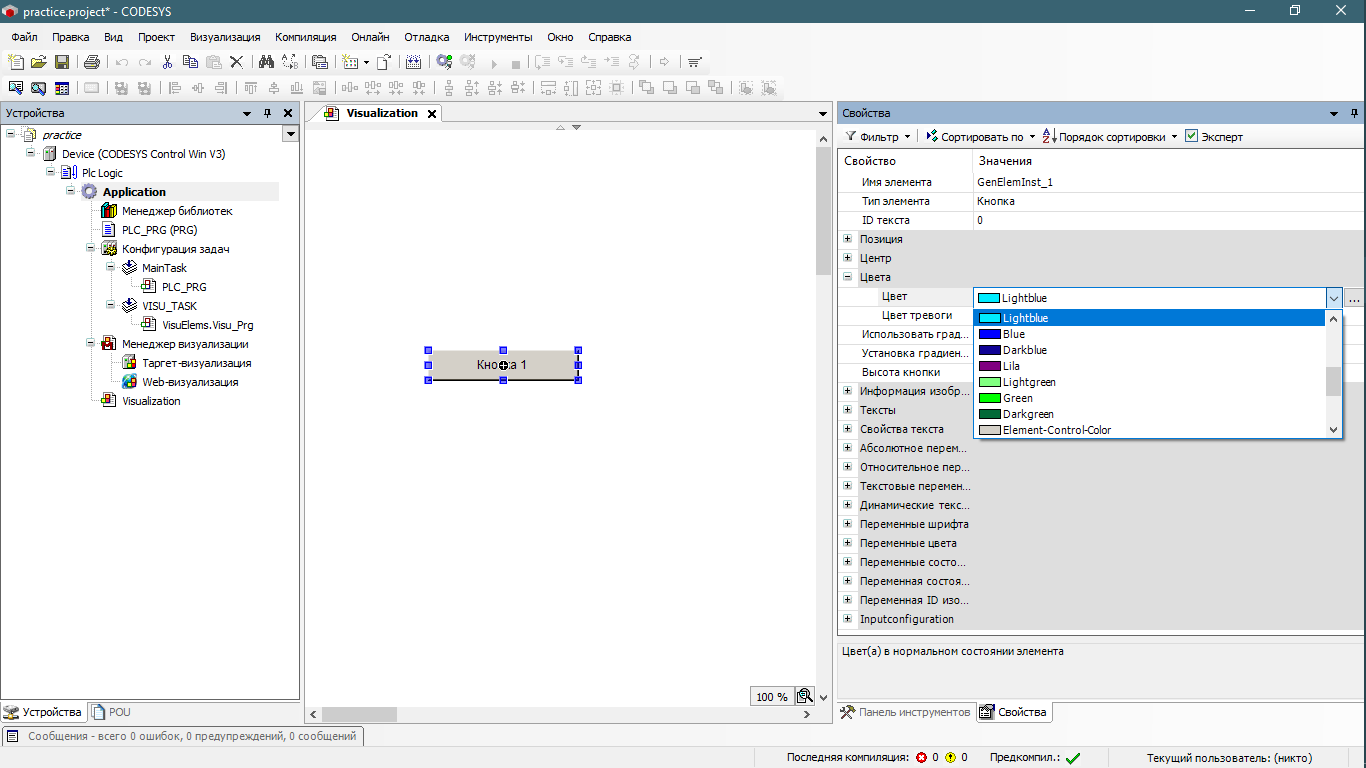
  
**Рисунок 4 — Добавление примитива «Кнопка»**

Зададим «Имя Кнопки» (Рисунок 5).

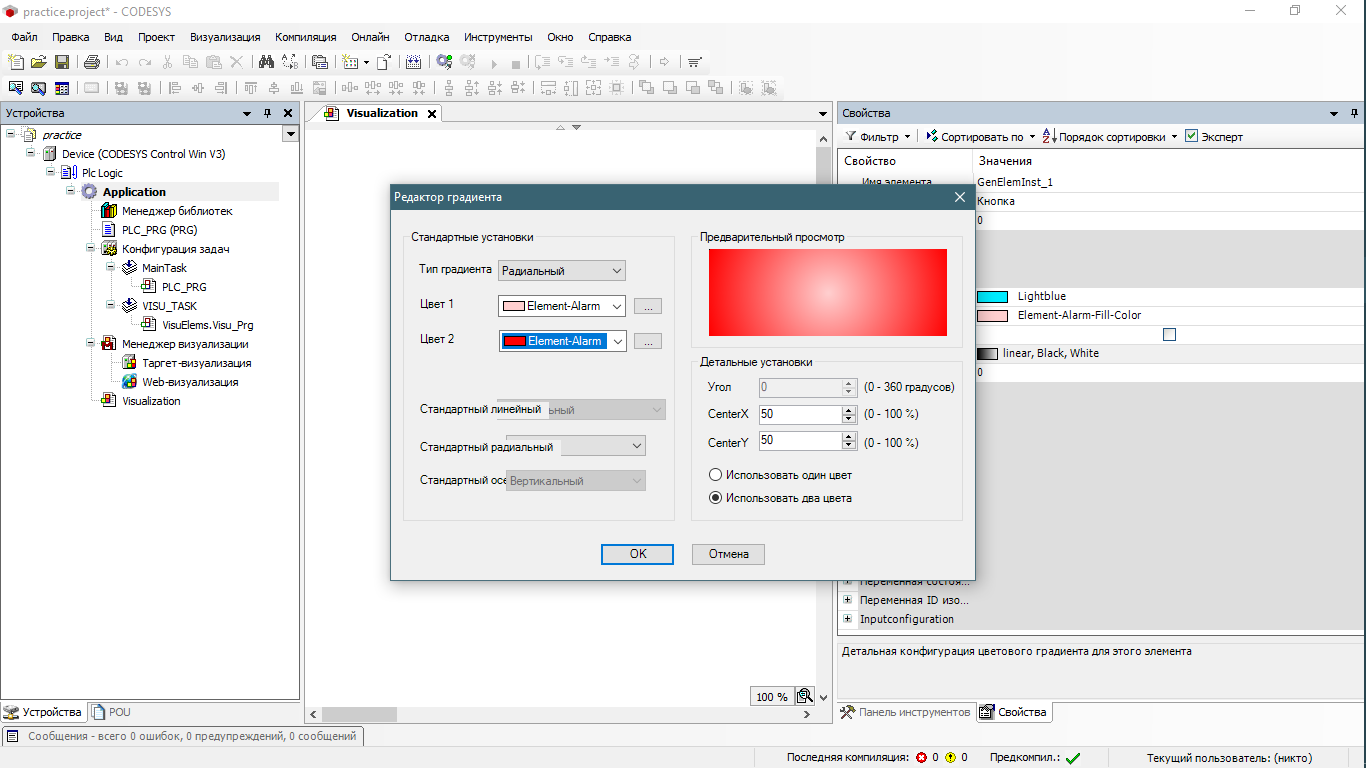
  
**Рисунок 5 — Изменение имени Кнопки**

**2.2.4. Выбор цвета стиля или фиксированного цвета для примитива визуализации**

Выберем примитив «Кнопка1». Активируем Дополнительные свойства. Выберем нужный цвет «Кнопки» (Рисунок 6).

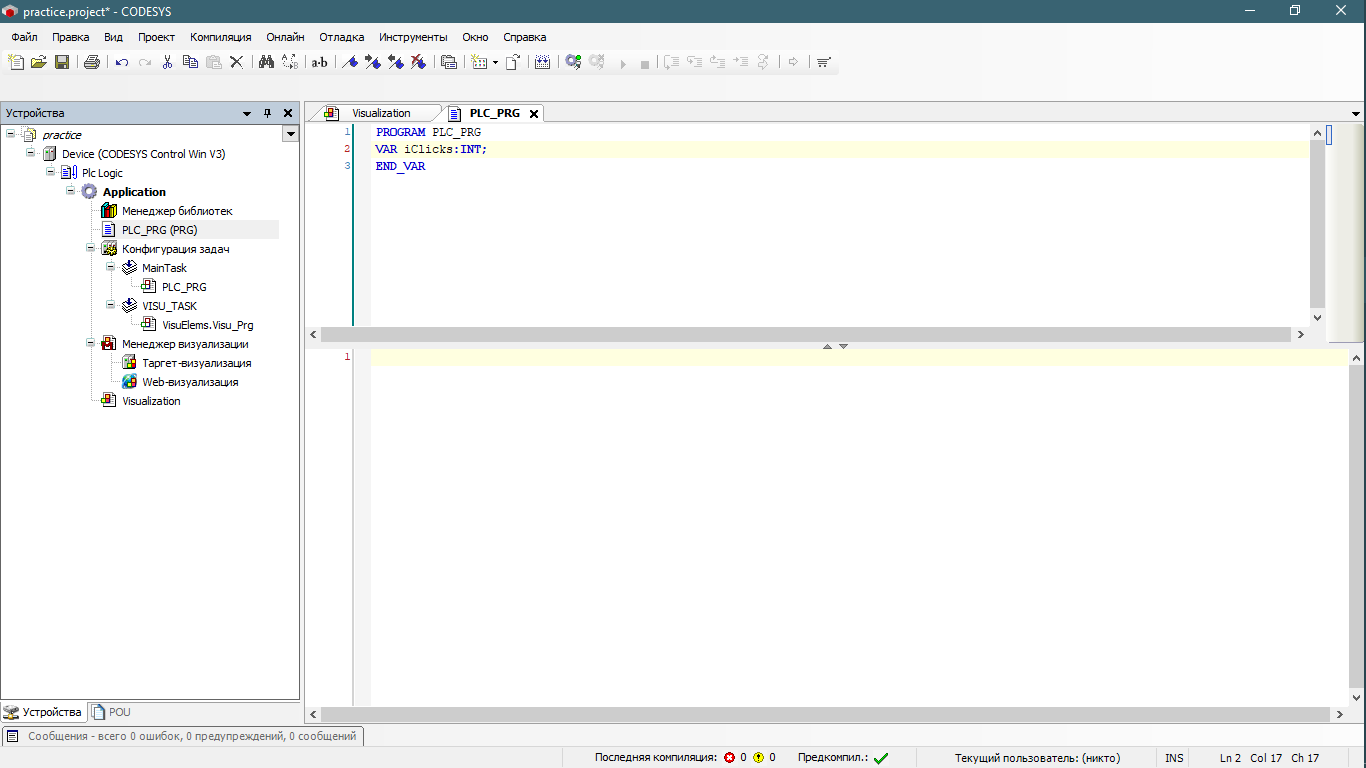
  
**Рис. 6. Выбор цвета «Кнопки»**

Установим «Градиент» цвета (Рисунок 7).

  
**Рисунок 7 — Установка «Градиента цвета»**

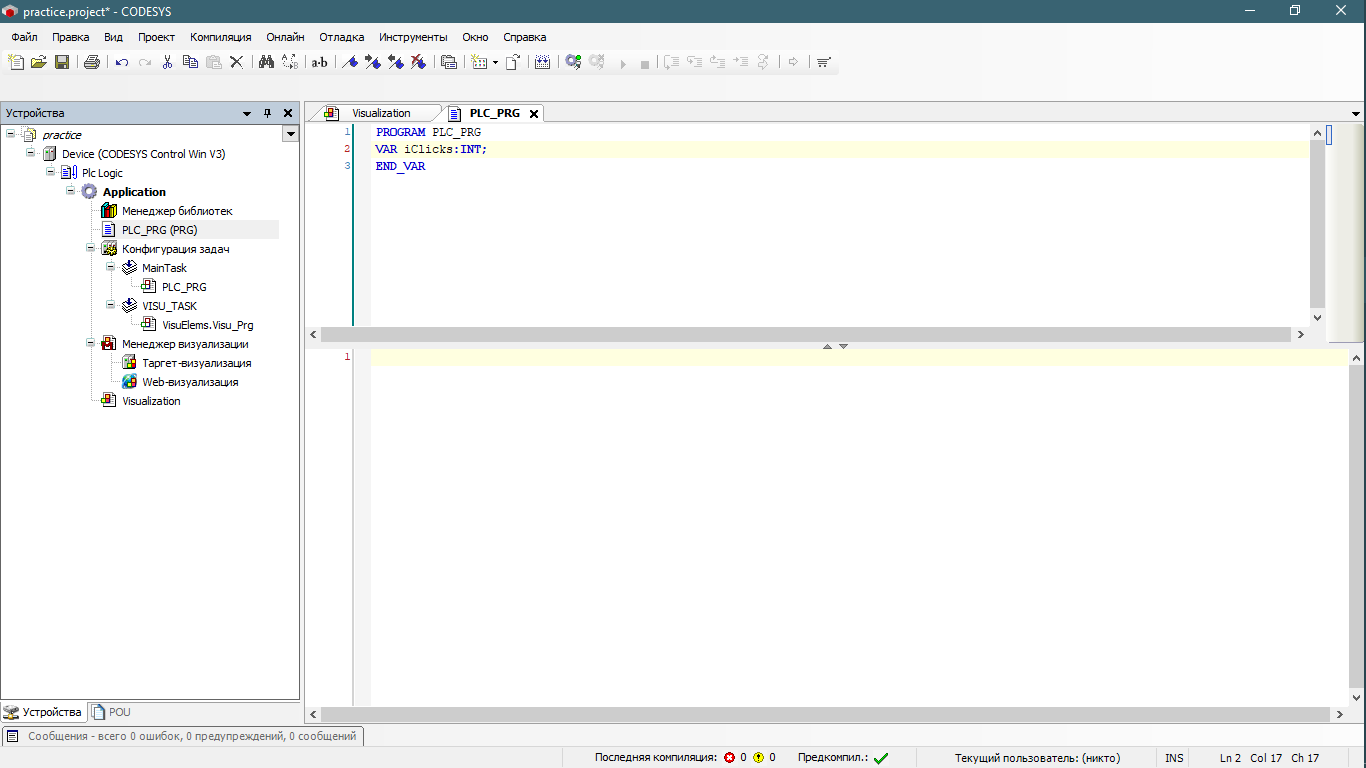
Далее добавим скрипт на ранее добавленную кнопку на языке ST. При нажатии на кнопку счетчик должен прибавлять единицу и выводить результат.

### **2.2.5. Добавляем плейсхолдер в имя кнопки (Рисунок 8)**

  
**Рисунок 8 — Добавление плейсхолдера**

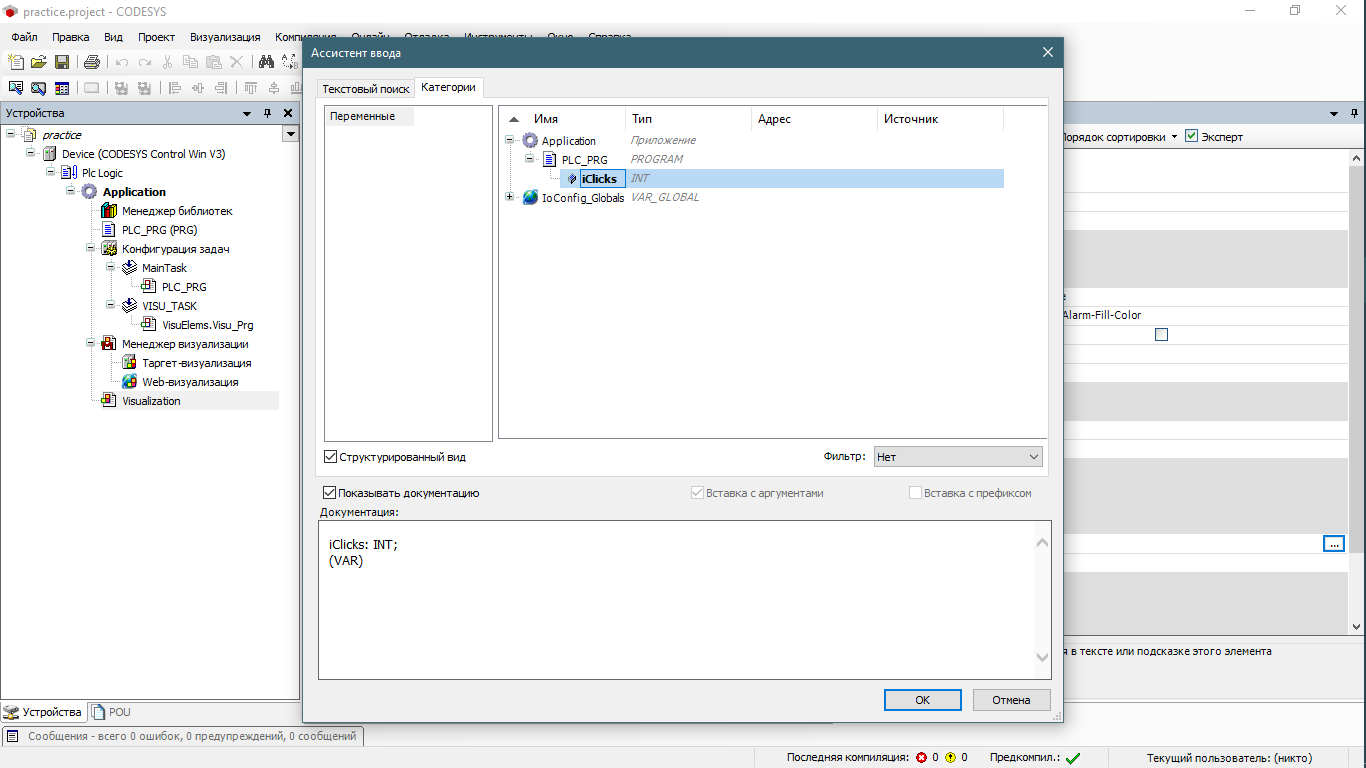
### **2.2.6. Объявляем переменную**

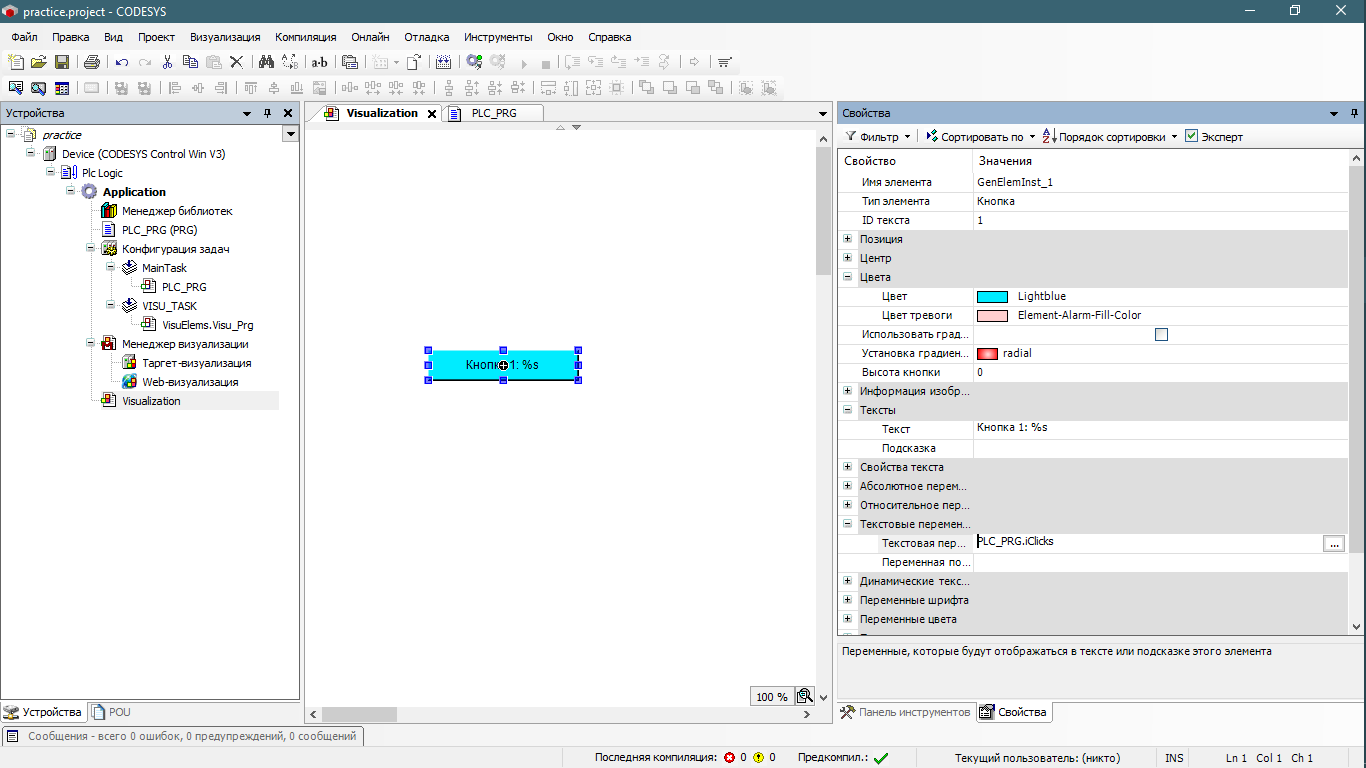
iClicks: INT;

  
**Рисунок 9 — Объявление переменной типа INT**

**2.2.7. Добавление переменной в свойства текста кнопки**

Привязываем переменную к «Кнопке» (Рисунок 10). После добавления переменной она отобразится в свойствах «Кнопки» (Рисунок 11).

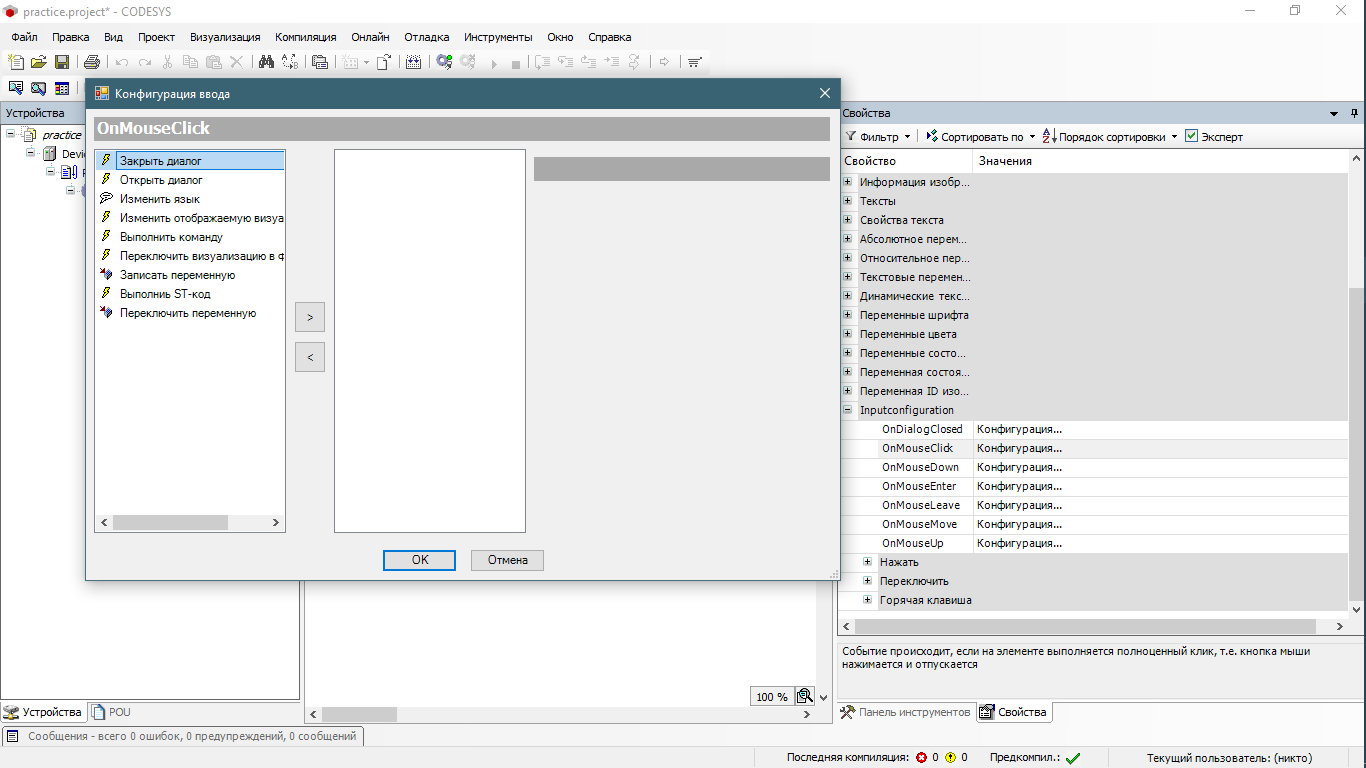
**Рисунок 10 — Добавление переменной в свойства кнопки**

  
**Рисунок 11 — Отображение переменной в свойствах «Кнопки»**

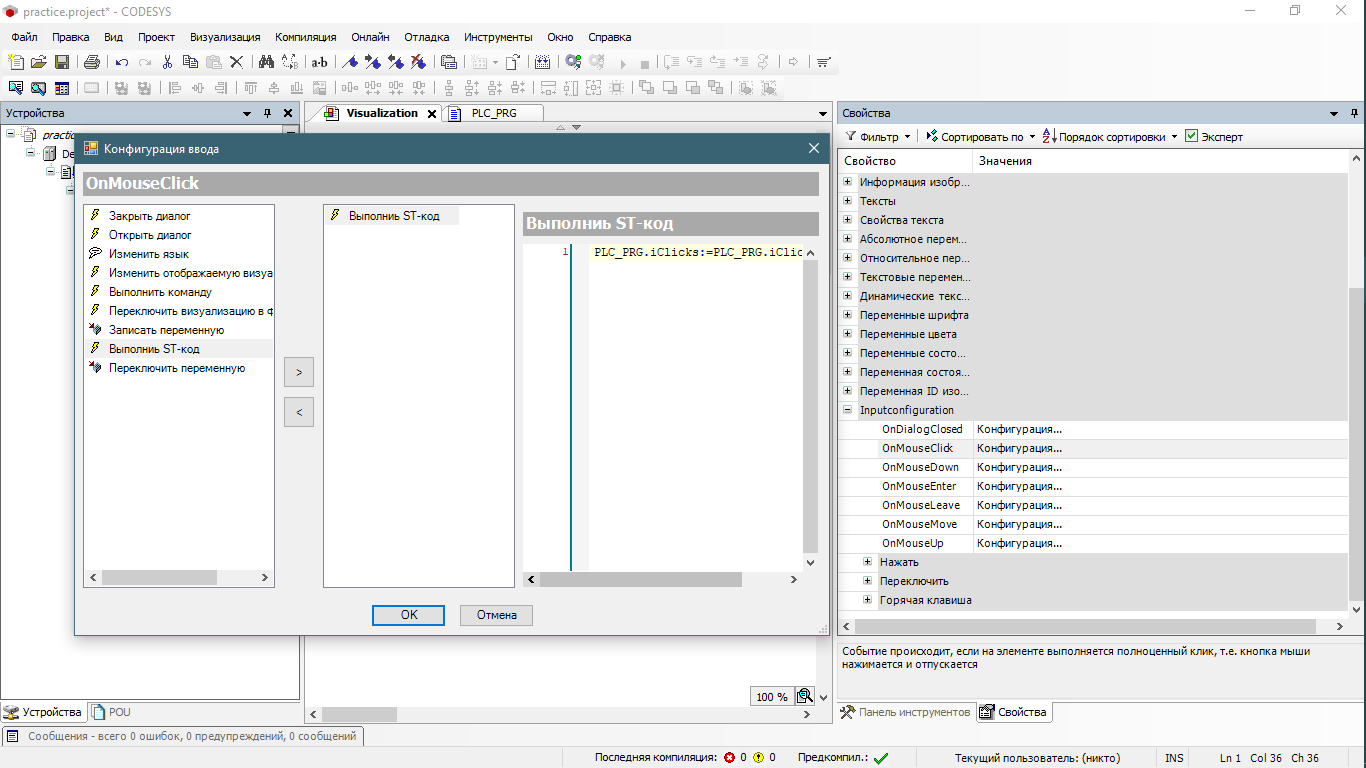
Теперь во время исполнения программы вместо плейсхолдера будет выводиться значение переменной.

### **2.2.8. Конфигурация действия при нажатии мыши на кнопку**

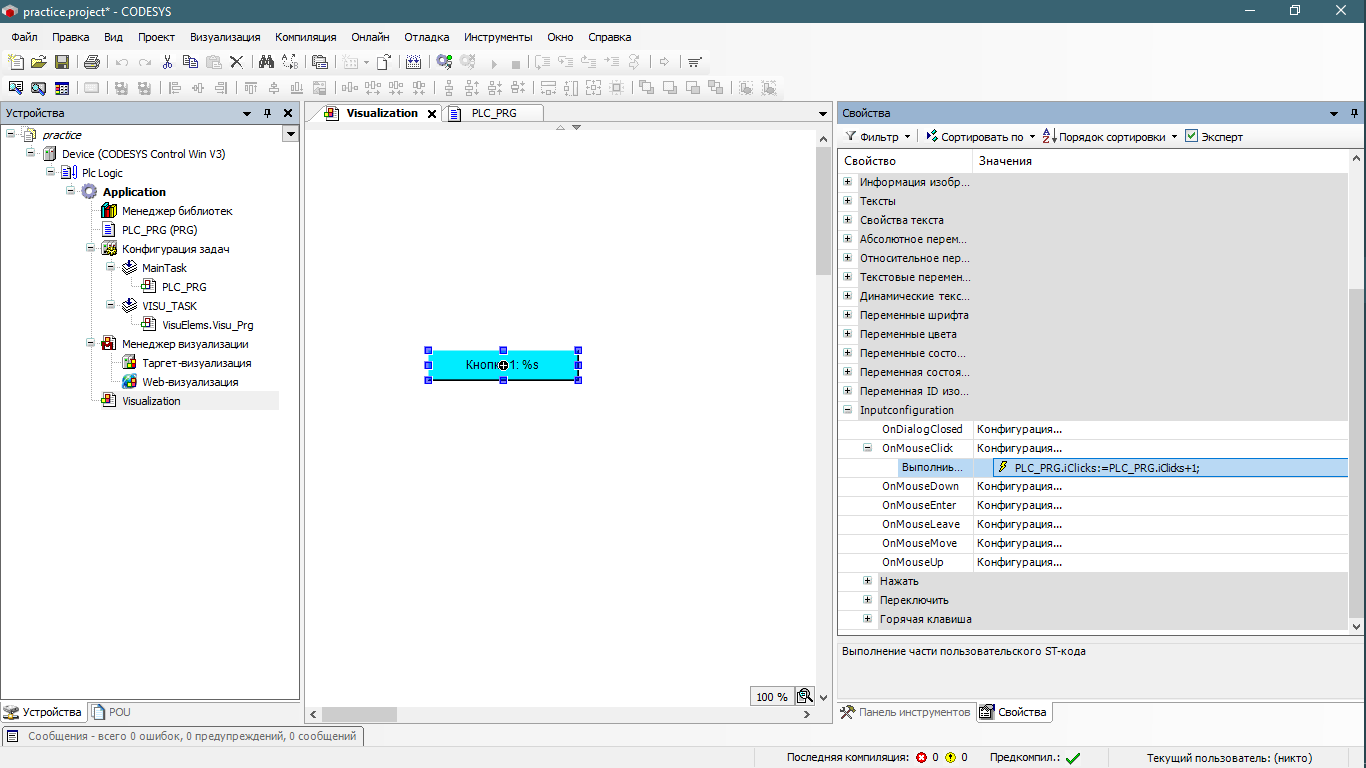
Выбираем действие при нажатии мышки на кнопку (Рисунок 12).

  
**Рисунок 12 — Конфигурация действия при нажатии на кнопку**

Вводим код, который будет выполняться при нажатии на кнопку (Рисунок 13).

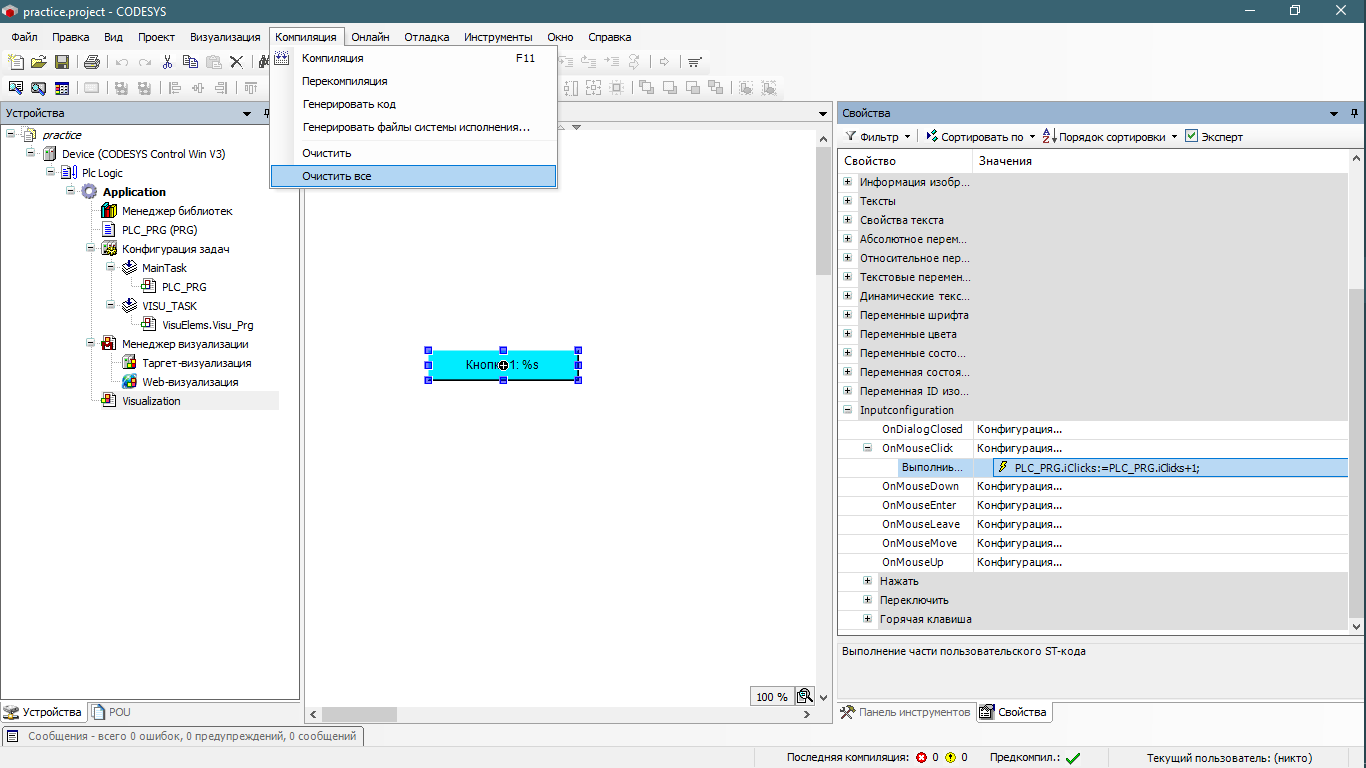
  
**Рисунок 13 — Код программы (действия) «Кнопки»**

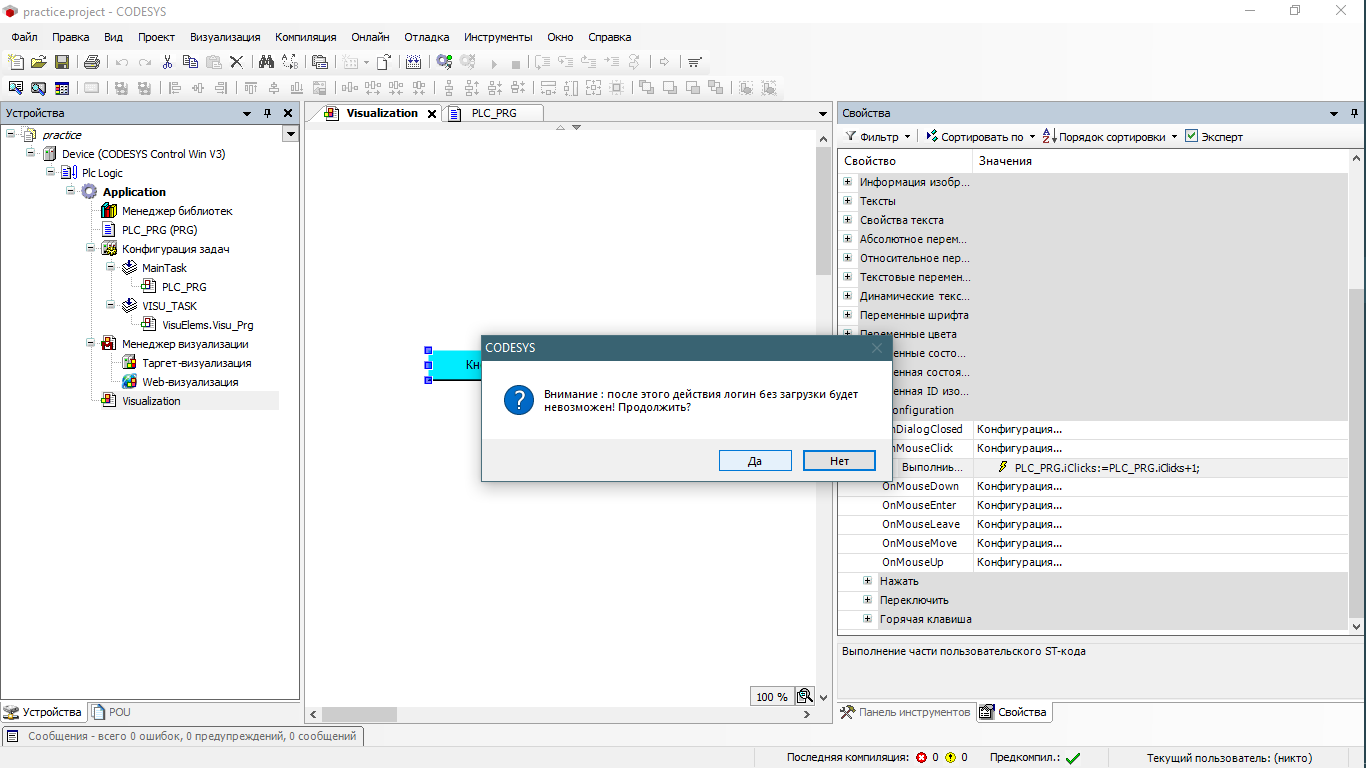
После всех настроек код действия отобразится в свойствах «Кнопки» (Рисунок 14).

  
**Рисунок 14 — Отображение кода действия в свойствах «Кнопки»**

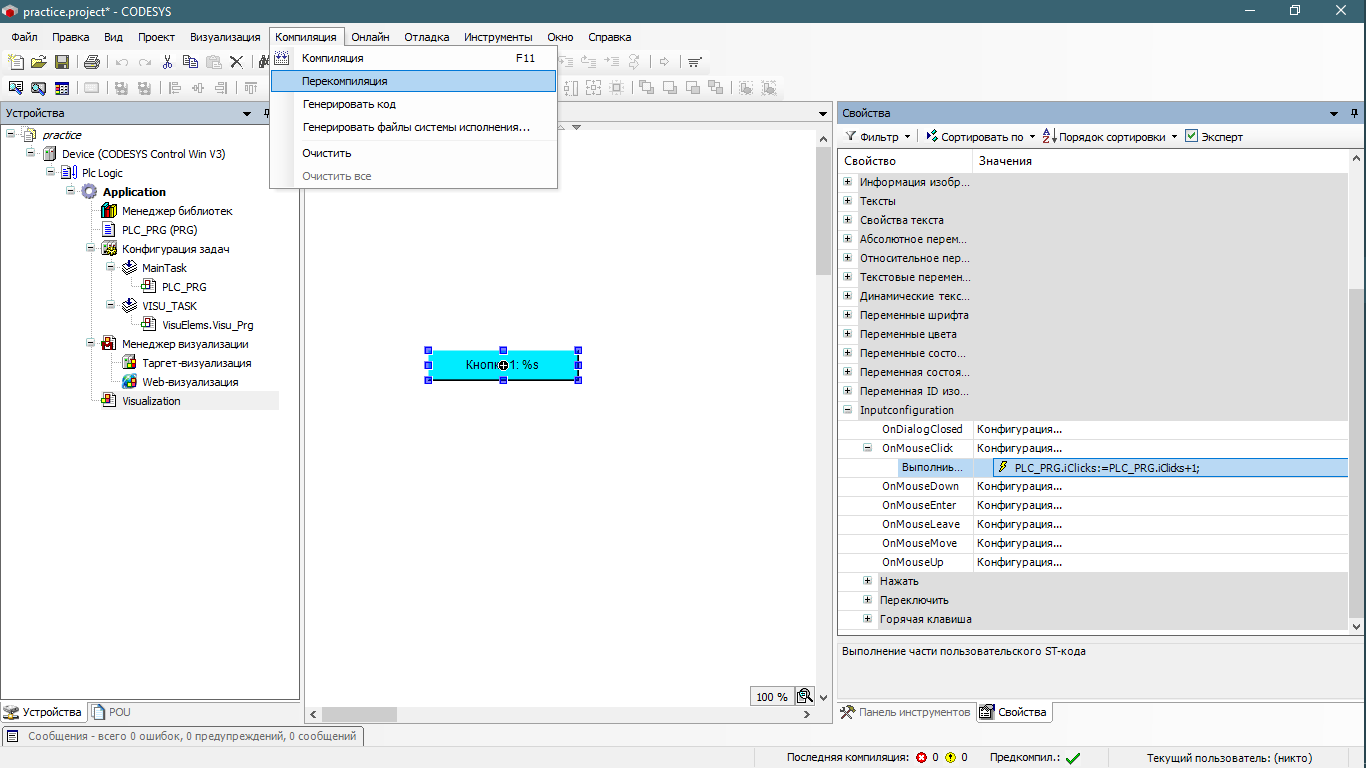
**2.2.9. Компиляция, загрузка и запуск программы**

Перед загрузкой проекта в Soft PLC сделаем предварительную очистку буфера (Рисунок 15), (Рисунок 16).

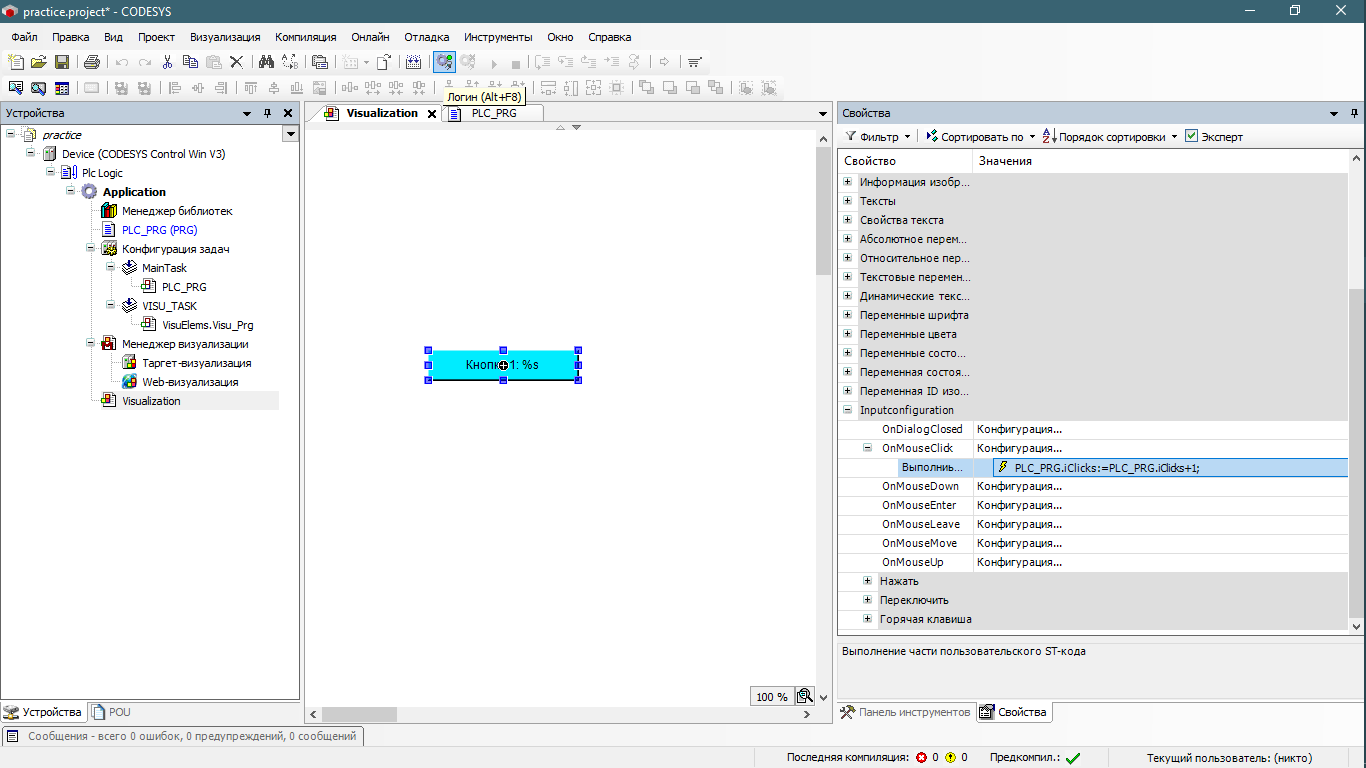
  
**Рисунок 15 — Очистка буфера**

  
**Рисунок 16 — Подтверждение очистки**

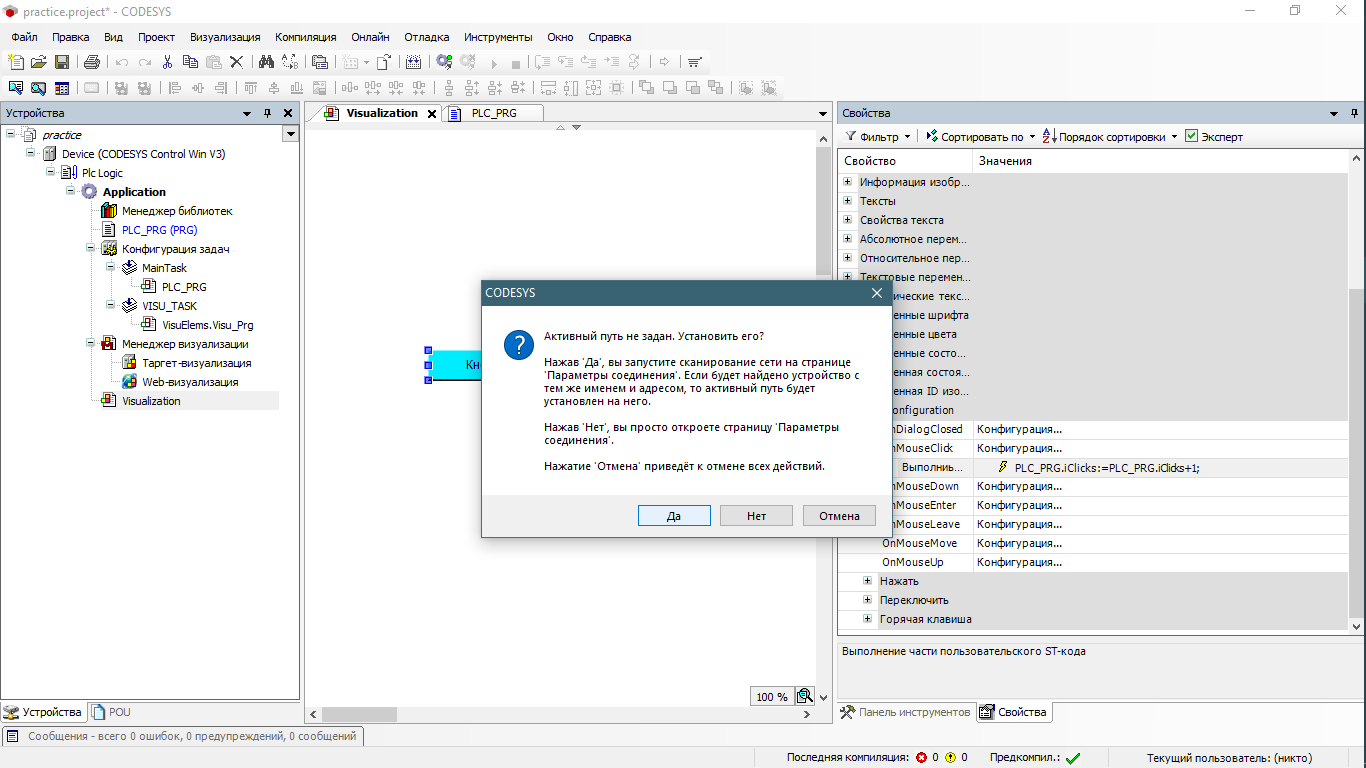
После этого выполним полную перекомпиляцию проекта (Рисунок 17).

  
**Рисунок 17 — Перекомпиляция проекта**

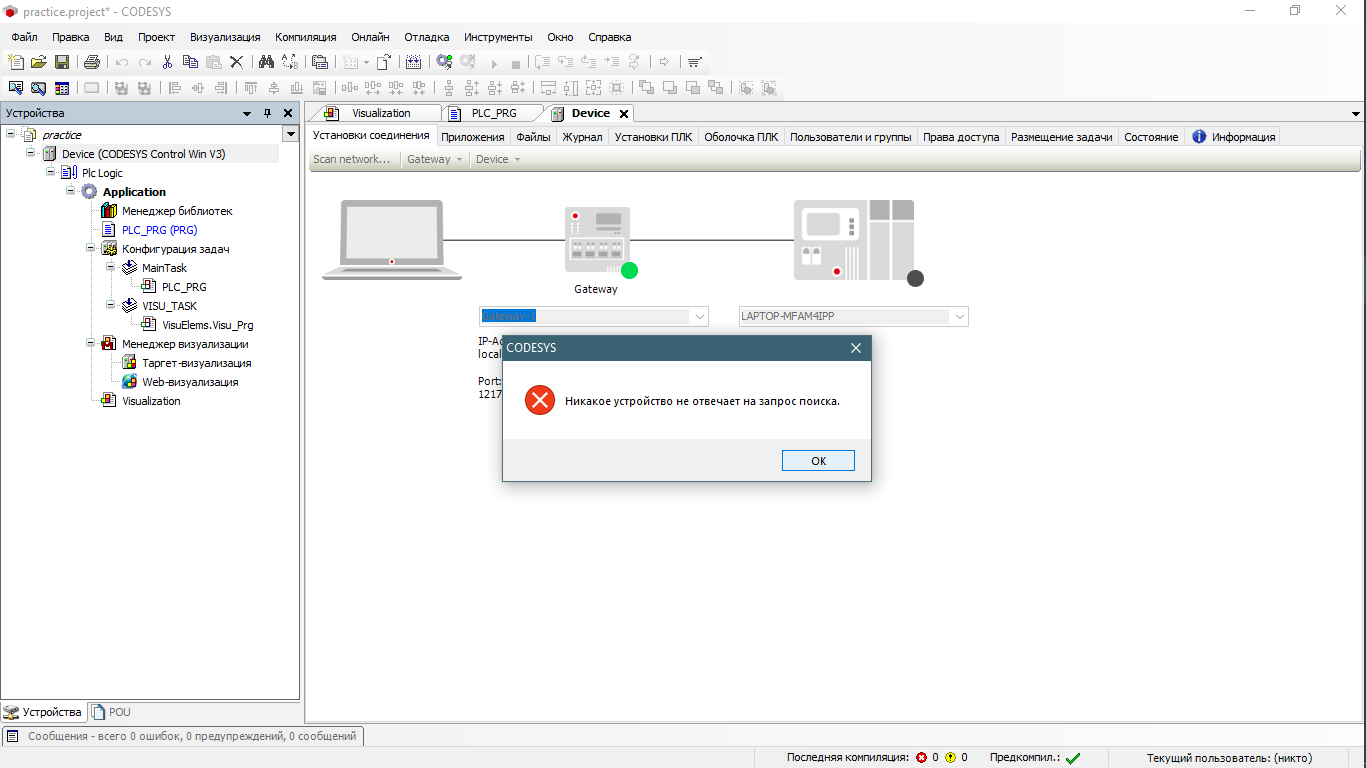
Далее проверим связь (подключение) с ПЛК. (Рисунок 18).

  
**Рисунок 18 — Проверка подключения к ПЛК**

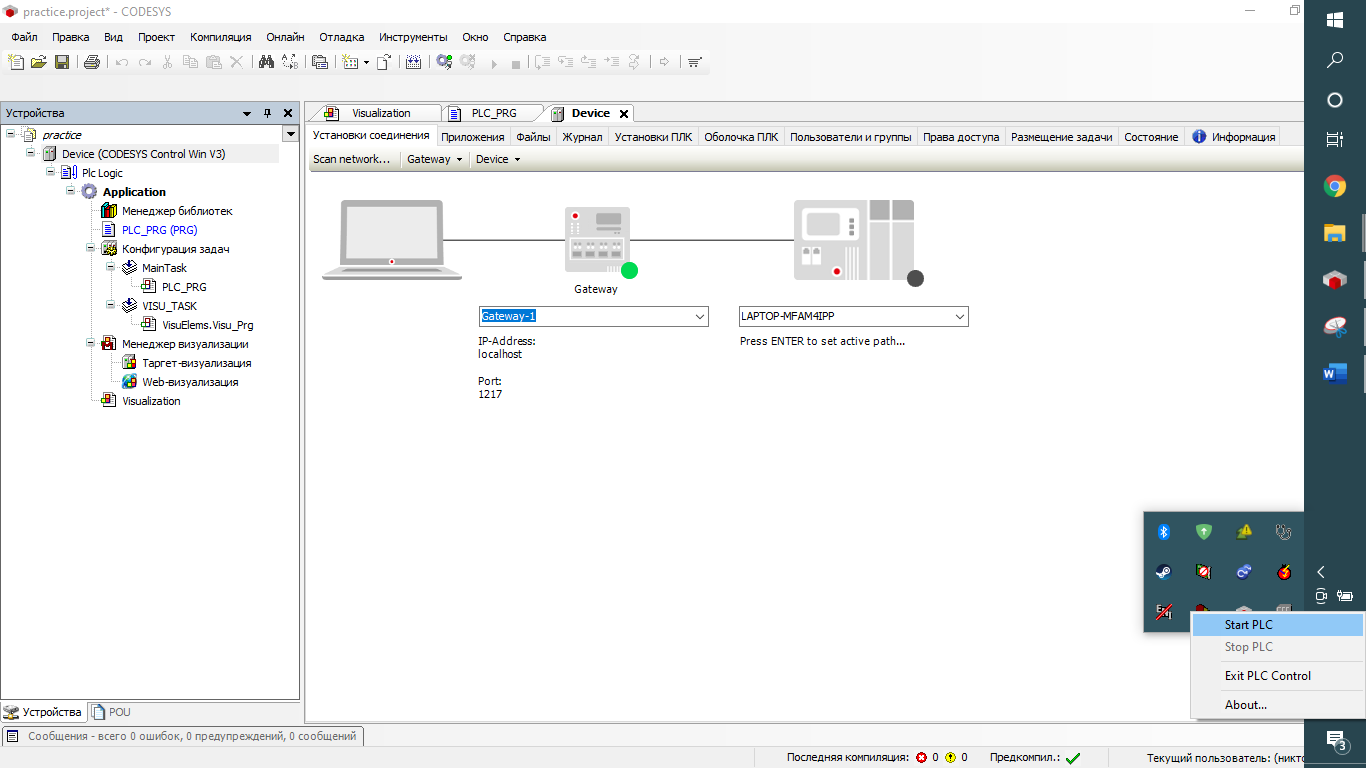
Затем просканируем сеть на наличие PLC (Рисунок 19).

  
**Рисунок 19 — Сканирование сети**

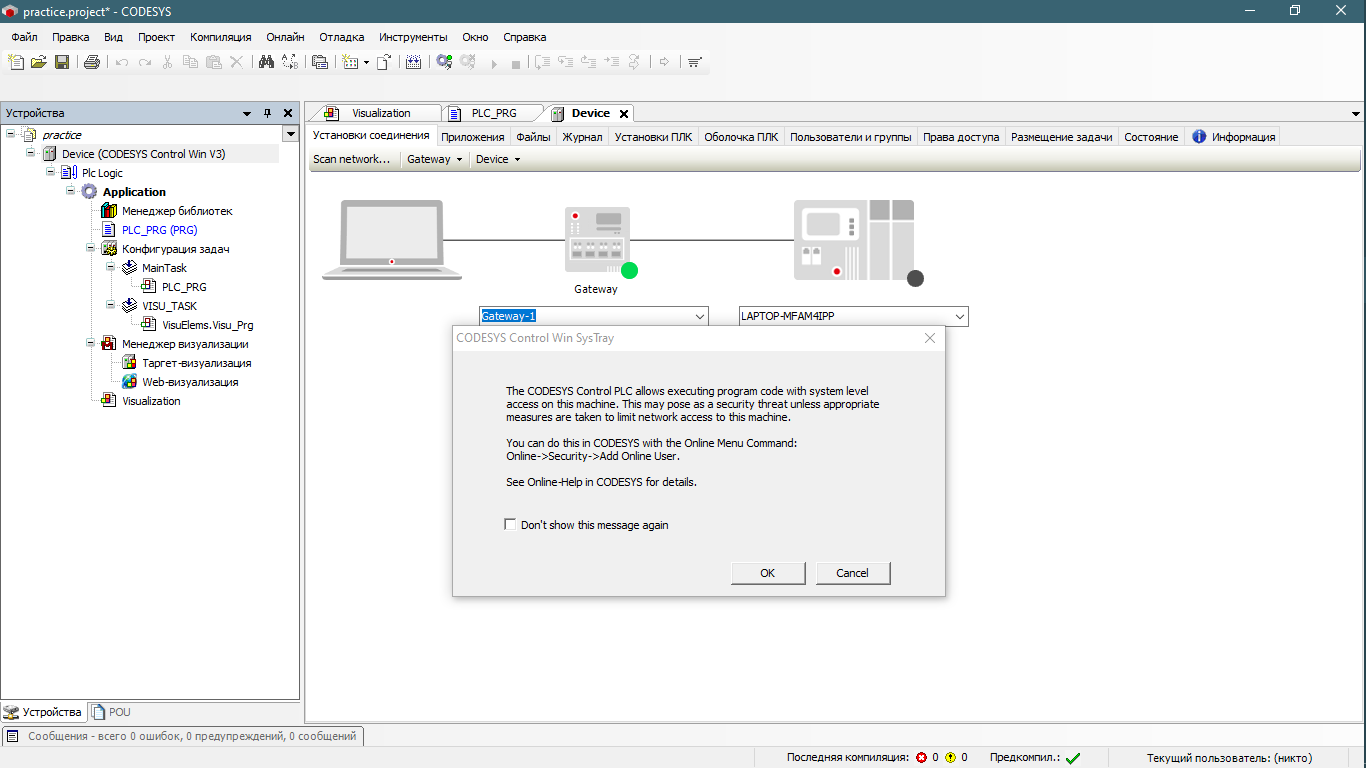
В случае отсутствия активных подходящих устройств на экране будет выдано предупреждение (Рисунок 20).

  
**Рисунок 20 — Индикация отсутствия подключения**

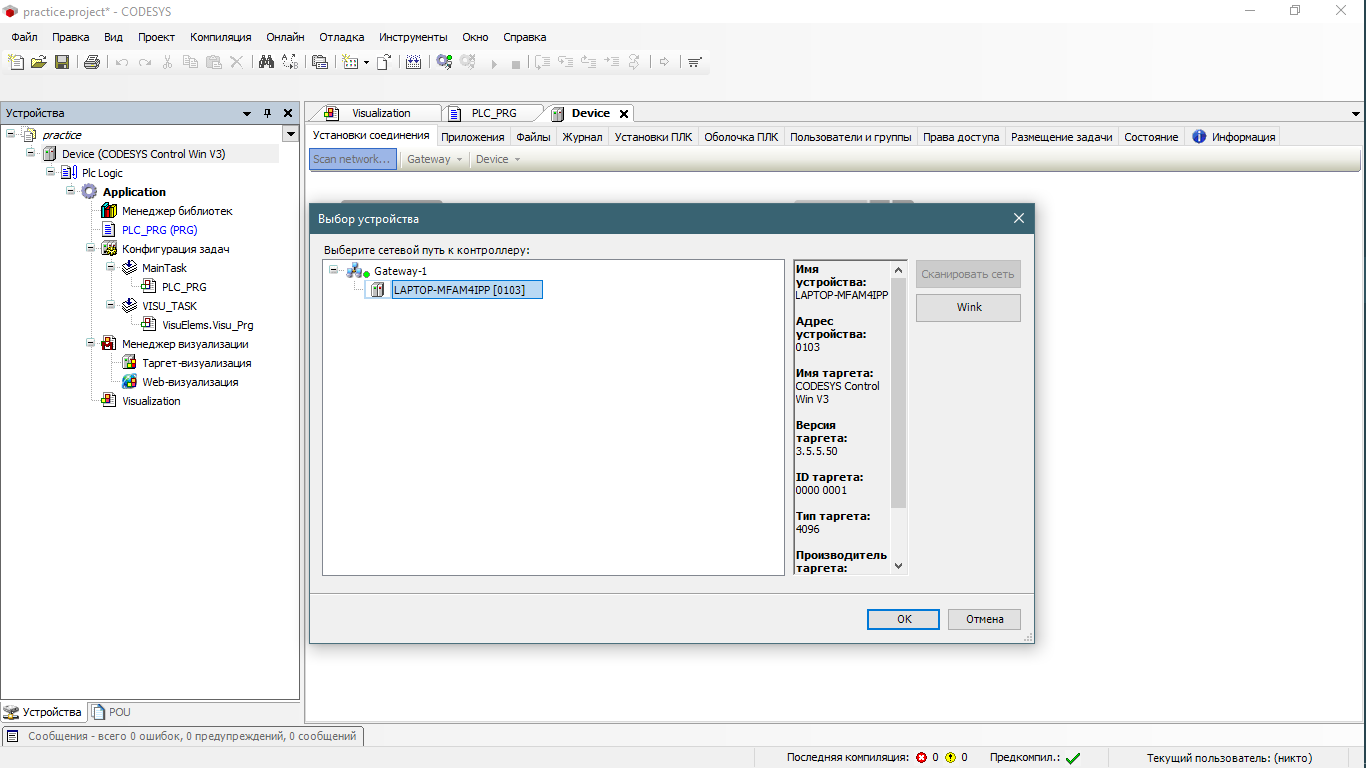
В нашем случае необходимо запустить SoftPLC. Для этого включим Soft PLC через панель задач (Рисунок 21).

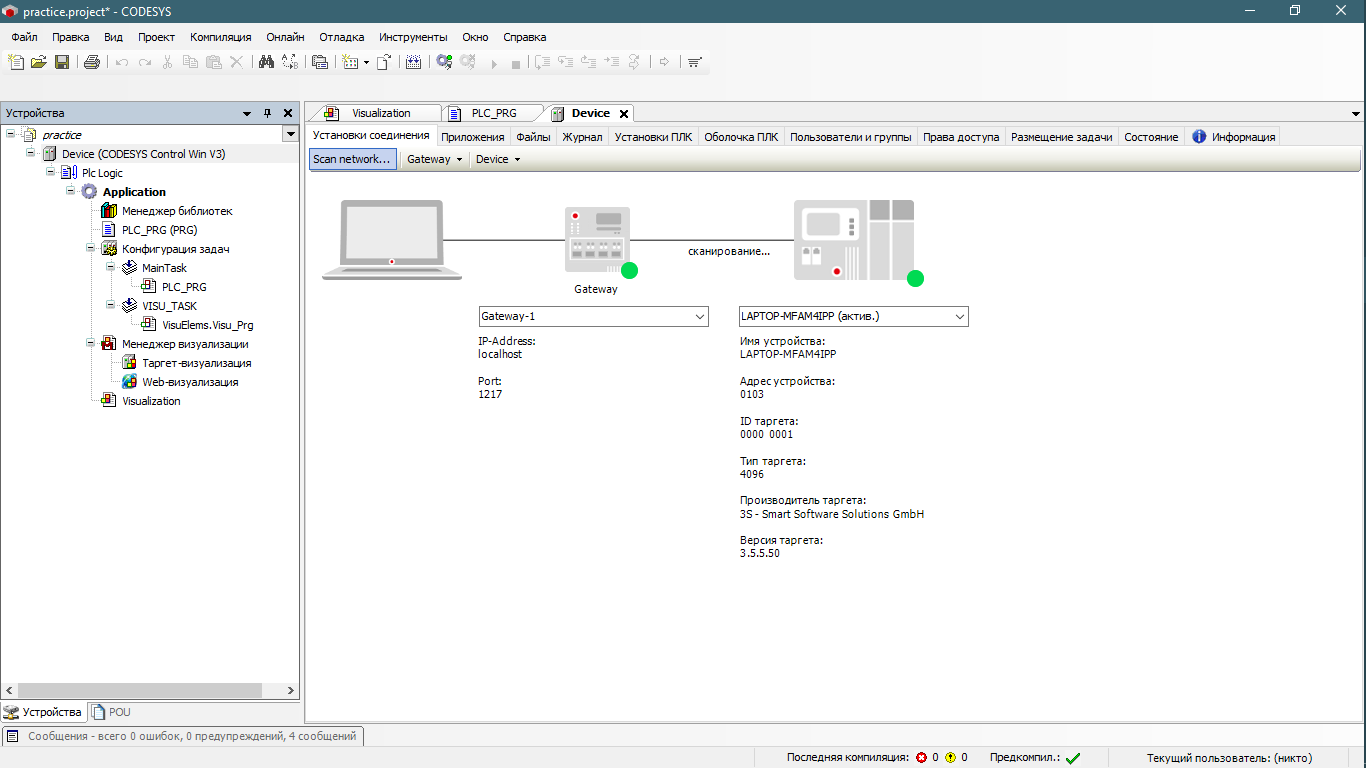
  
**Рисунок 21 — Старт Soft PLC**

После запуска подтвердим данное действие (Рисунок 22).

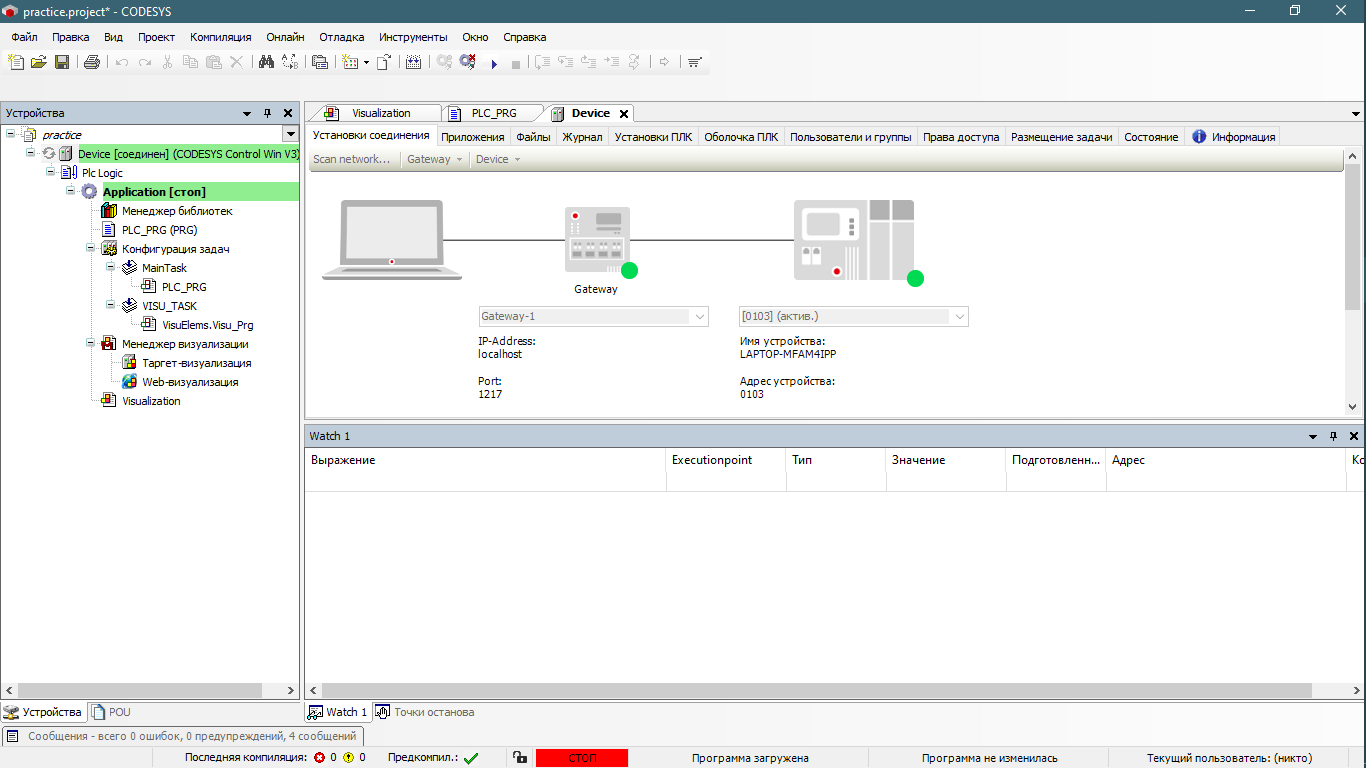
  
**Рисунок 22 — Подтверждение старта Soft PLC**

После запуска SoftPLC необходимо заново просканировать сеть и убедиться в его активности (Рисунок 23), (Рисунок 24).

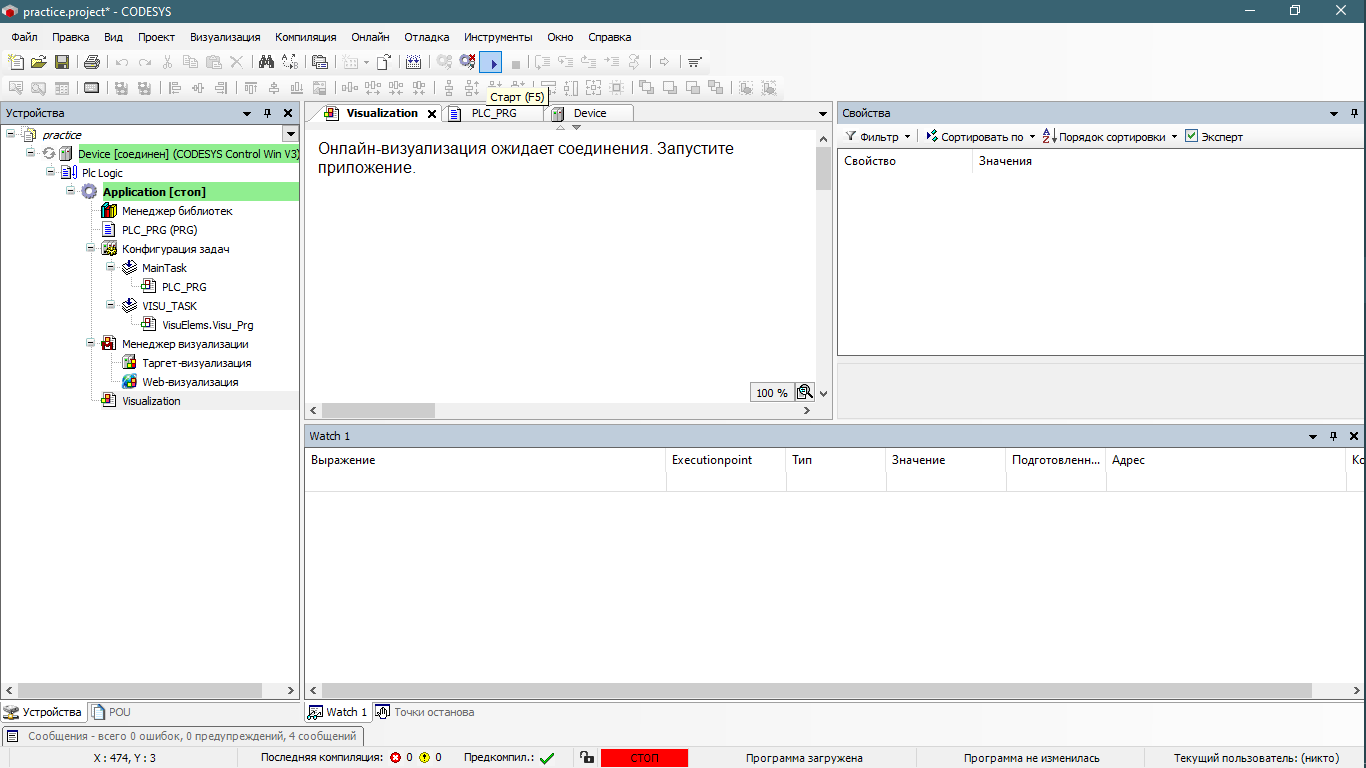
  
**Рисунок 23 — Сканирование сети**

  
**Рисунок 24 — Индикация активного Soft PLC**

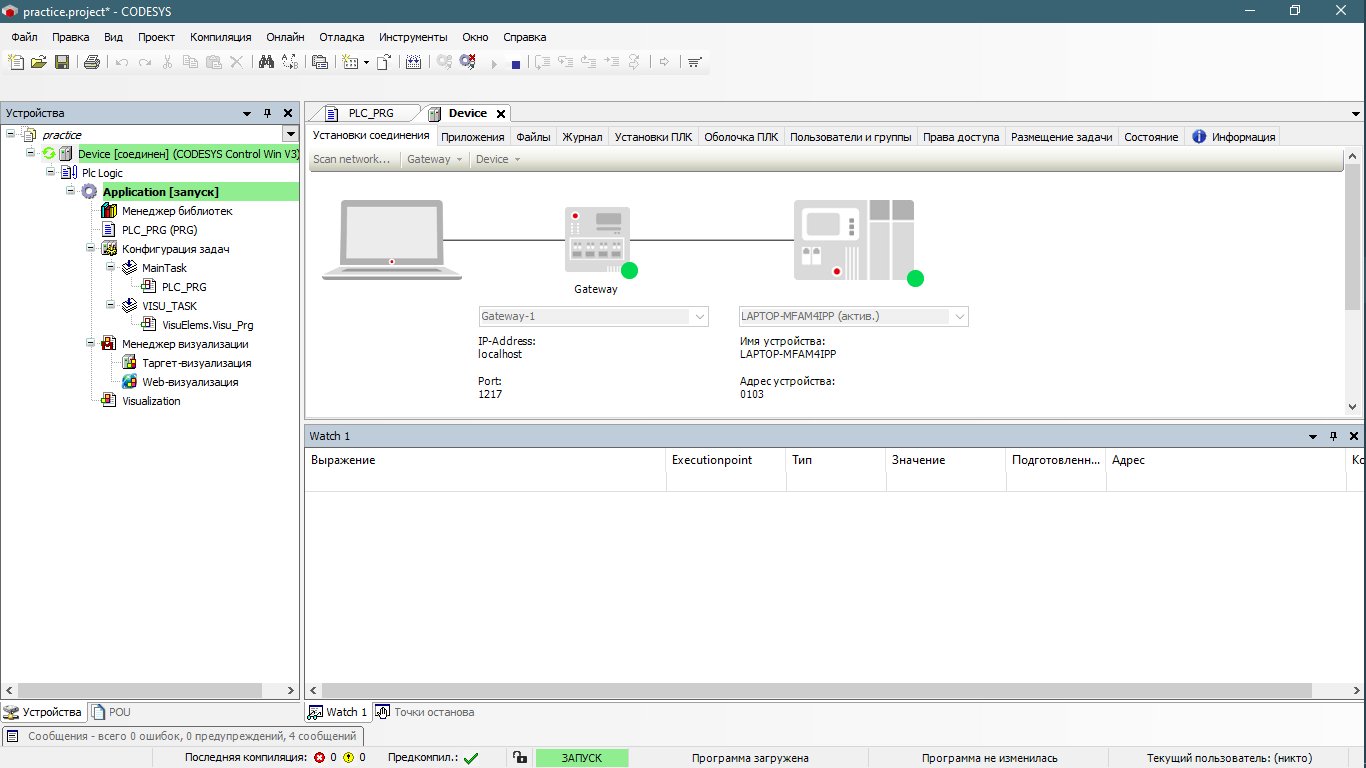
Теперь SoftPLC активен и можно произвести подключение. После подключения будет предложено обновить код в SoftPLC кодом из проекта. После удачной прогрузки проекта в SoftPLC программа в контроллере будет находиться в состоянии «стоп» (Рисунок 25). [5]

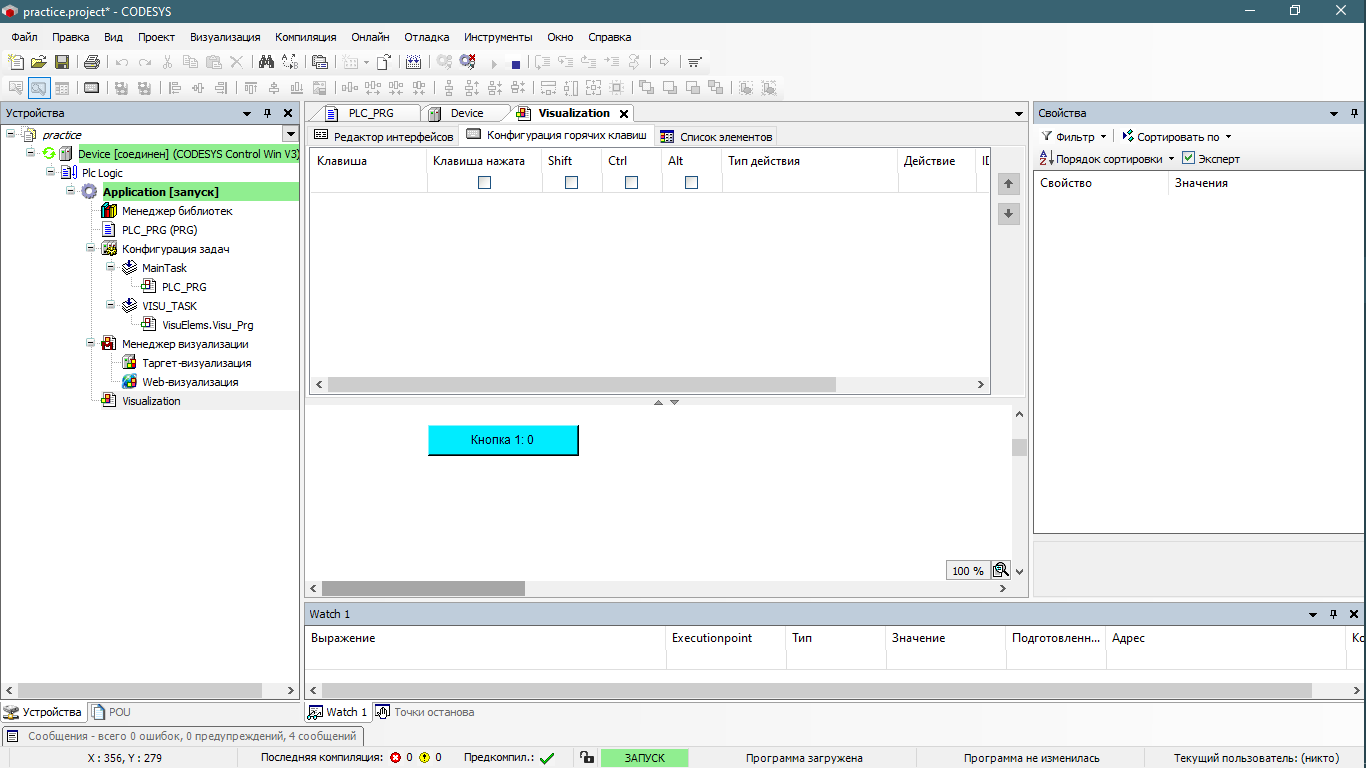
  
**Рисунок 25 — Индикация активного подключения**

Произведем запуск программы в SoftPLC (Рисунок 26).

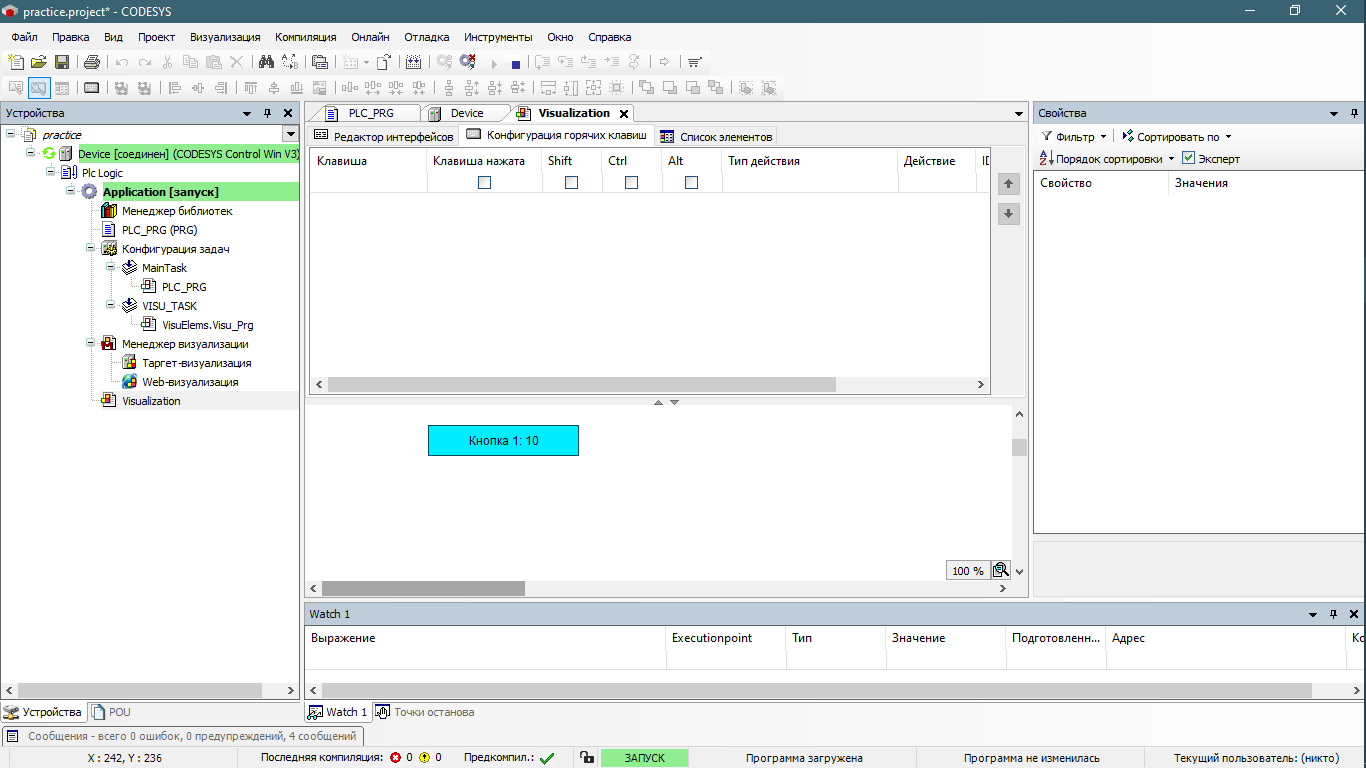
  
**Рисунок 26 — Запуск программы**

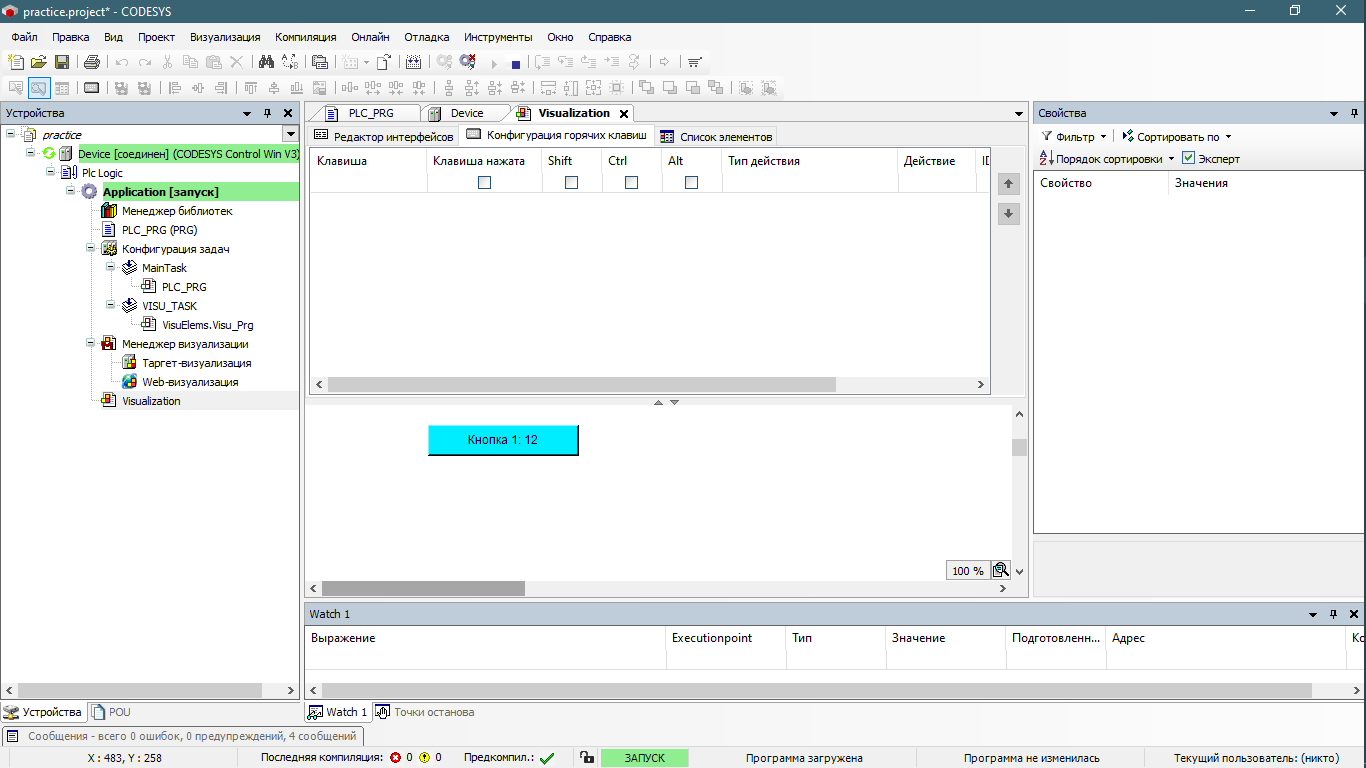
После запуска программы ее статус изменится на «запуск» (Рисунок 27), (Рисунок 28).

  
**Рисунок 27 — Активная программа визуализации (1)**

  
**Рисунок 28 — Активная программа визуализации (2)**

Для проверки работы программы нажимаем на кнопку, счетчик увеличивается (Рисунок 29), (Рисунок 30).

  
**Рисунок 29 — Увеличение счетчика при нажатии на кнопку (1)**

  
**Рисунок 30 — Увеличение счетчика при нажатии на кнопку (2)**

# **3 Московский завод «ФИЗПРИБОР»**

Ознакомительная практика проводилась при содействии Московского завода «ФИЗПРИБОР».

Завод «ФИЗПРИБОР» специализируется на разработке и производстве высококачественных технических средств для построения автоматизированных систем контроля управления технологическими и производственными процессами высокой степени надёжности для атомных электростанций и предприятий гражданской промышленности. Завод производит контроллеры низовой автоматики, работающие как на «жёсткой» логике в современном исполнении, так и программно-аппаратные логические контроллеры, комплексы распределенных систем управления и системы противоаварийной защиты с высокими характеристиками надежности и безопасности. [3]



**Группа КВБО-01-21 с представителем «ФИЗПРИБОРА»**

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной практической работы мы научились:

• Создавать стандартный проект в Codesys 3.5

• Добавлять визуализацию в проект

• Добавлять примитивы визуализации

• Добавлять плейсхолдер в имя кнопки

• Объявлять переменные

• Конфигурировать действия при нажатии на кнопку

• Настраивать и запускать SoftPLC

• Компилировать, загружать и запускать программу в PLC

Итогом выполненной практической работы стал графический интерфейс с одной кнопкой, при нажатии на которую, счетчик нажатий увеличивается.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Avtprom.ru – Статья – CODESYS – повседневный инструмент программиста ПЛК ― URL: <http://savtprom.ru/article/codesys-–-povsednevnyi-instrumen>
2. Официальный сайт CODESYS – Codesys.com ― URL: <https://www.codesys.com>
3. Официальный сайт завода «ФИЗПРИБОР» ― URL: <https://fizpribor.ru/>
4. Визуализация в среде CODESYS - URL: <https://aip.com.ru/article/vizualizatsiya_v_srede_codesys>
5. Д.В. Пастушенков, С.В. Золоторёв "CoDeSys SP RTE: SoftPLC + возможности реального времени в среде Windows"