SystemePLC Studio

Программное обеспечение SystemePLC Studio

для интеллектуальных реле SystemePLC SR и ПЛК SystemePLC S172

Быстрый старт

systeme.ru

Système electric

Содержание

Поехали! 4
Где взять ПО и ВОС ПЛК? 4
Создание нового проекта 5
Основы интерфейса 7
Панель инструментов (Toolbar) 8
Вкладки меню (Tabs): содержит четыре основные вкладки 9
Подключение ПЛК к ПК через USB Type-C* 9
Проверка соответствия ВОС ПЛК и версии ПО SystemePLC Studio* 12
Перепрошивка ПЛК* 13
Конфигурация аппаратной части ПЛК 14
Конфигурация контроллера и модулей 15
Связывание переменных с физическими входами/выходами (IO Mapping) 17
Настройка физических входов/выходов (IO Mapping Local): 18
Демонстрационный проект Шаг1 19
Создание глобальных сетевых переменных (Field) 20
Демонстрационный проект Шаг2 21
Основы программирования 21
Интерфейс вкладки Программирование (Programming) 21
Дерево проекта (Project Tree) 23
Типы переменных 23
Задачи (Tasks) 24
Написание программы 25
Панель инструментов Программирование (Programming Toolbar) 28
Советы по программированию 29
Демонстрационный проект Шаг 3 32
Программирование дисплея ПЛК 33
Интерфейс вкладки Дисплей (Display) 33
Панель инструментов Дисплей (Display Toolbar) 34
Создание пользовательского экрана ПЛК 35
Создание пользовательского элемента 35
Основные элементы пользовательских станиц 36

Расширенные функции 39
Создание многостраничного интерфейса 41
Демонстрационный проект Шаг 4 41
Компиляция программы 43
Загрузка программы в ПЛК* 43
Загрузка программы в оффлайн симулятор ПЛК 43
Запуск / Стоп программы ПЛК (Run/Stop) 43
Оффлайн симулятор ПЛК (Simulation Mode) 44
Онлайн мониторинг ПЛК (Live Debug Mode) 45
Окно наблюдения за переменными (Watch Window) 46
Демонстрационный проект Шаг 5 47
Логика работы демонстрационного проекта 48
Дополнительные возможности 49
Используемые сокращения и термины 50

Поехали!

Программное обеспечение SystemePLC Studio представляет собой среду программирования для всех интеллектуальных реле SystemePLC SR и ПЛК SystemePLC S172. ПО поставляется бесплатно и доступно для свободного скачивания.

Важно! При обновлении версии ПО SystemePLC Studio также обновляются ВОС (внутренние операционные системы) [прошивки] ПЛК. Новая ВОС содержит часть реализации нового функционала ПО. Поэтому при использовании новой версии ПО необходимо удостовериться в том, что вы загрузили последнюю версию ВОС в ПЛК.
Инструкцию по загрузке ВОС в ПЛК см. в разделе Перепрошивка ПЛК.
Таблицу совместимости версий ВОС ПЛК и ПО SystemePLC Studio см. в Settings > Help > About.

Важно! Перед установкой новой версии ПО SystemePLC Studio обязательно деинсталлируйте старую версию.

Важно! ПО SystemePLC Studio можно начать полноценно использовать без физического ПЛК. С помощью встроенного оффлайн симулятора ПЛК вы сможете полноценно разработать и испытать написанную программу.

Назначение данного документа
В данном документе будут рассмотрены основные элементы и функции ПО, а также создан простой демонстрационный проект.

Порядок работы с документом

1. Посмотрите видео «SystemePLC Studio Быстрый старт»
2. Следуйте инструкциям по ходу данного документа с заголовками Демонстрационный проект Шаг 1,2,3,... и на выходе у вас получится протестировать созданный проект на симуляторе ПЛК или на физическом ПЛК (при его наличии).

Описание демонстрационного проекта «Нагреватель»
Данный проект реализует простейшую систему управления электронагревателем с гистерезисом для поддержания температуры в помещении по заданной уставке.

- Нагреватель включается, когда температура падает ниже уставки минус гистерезис (SetPoint - Hysteresis)
- Нагреватель выключается, когда температура поднимается выше уставки + гистерезис (SetPoint + Hysteresis)
- В диапазоне между этими порогами состояние нагревателя остается неизменным

Где взять ПО и ВОС ПЛК?

Скачайте последнюю версию ПО SystemePLC Studio и ВОС (внутренние операционные системы / прошивки) ПЛК по одной из ссылок.

Сайт: https://systeme.ru/

- Продукты > Промышленная автоматизация > Программируемые логические контроллеры > SystemePLC SR & S172
- Прямая ссылка: https://systeme.ru/products/systemeplc-sr-systemeplc-s172

ИЛИ
Материалы SystemePLC SR & S172 в облачном хранилище

- /SystemePLC Studio Software/SystemePLC Studio
- Прямая ссылка: https://workspace.systeme.ru/s/7TaicGCSDQgPqMr

ИЛИ
SystemePLC Studio Software

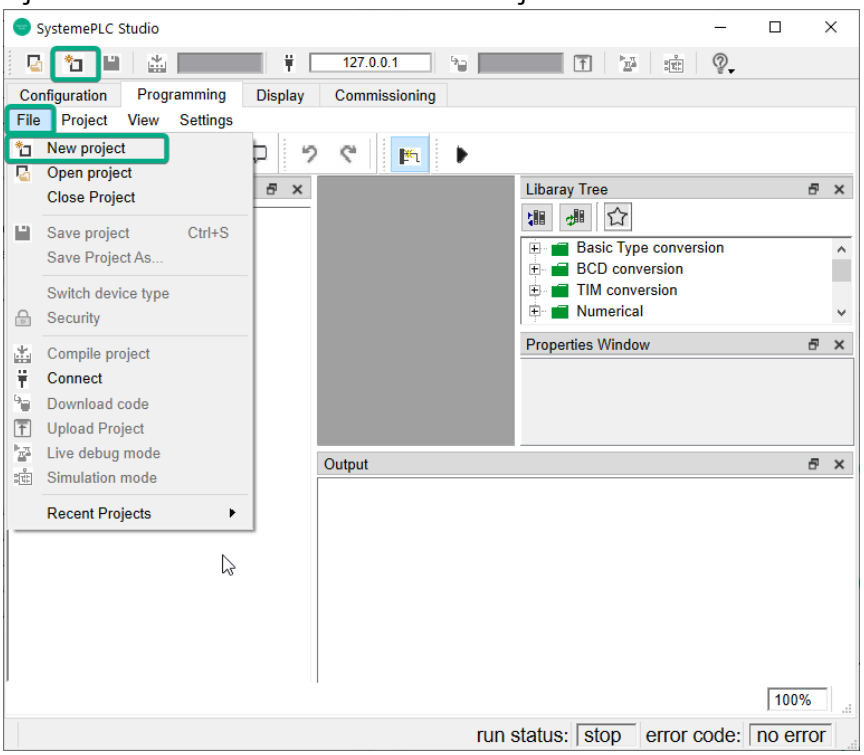
- Прямая ссылка: https://workspace.systeme.ru/s/Xd4LzsPZjd35cGb

Загрузите дистрибутив и следуйте инструкциям установщика.
Для установки требуются права администратора.

Создание нового проекта

Для начала работы с новым проектом необходимо:

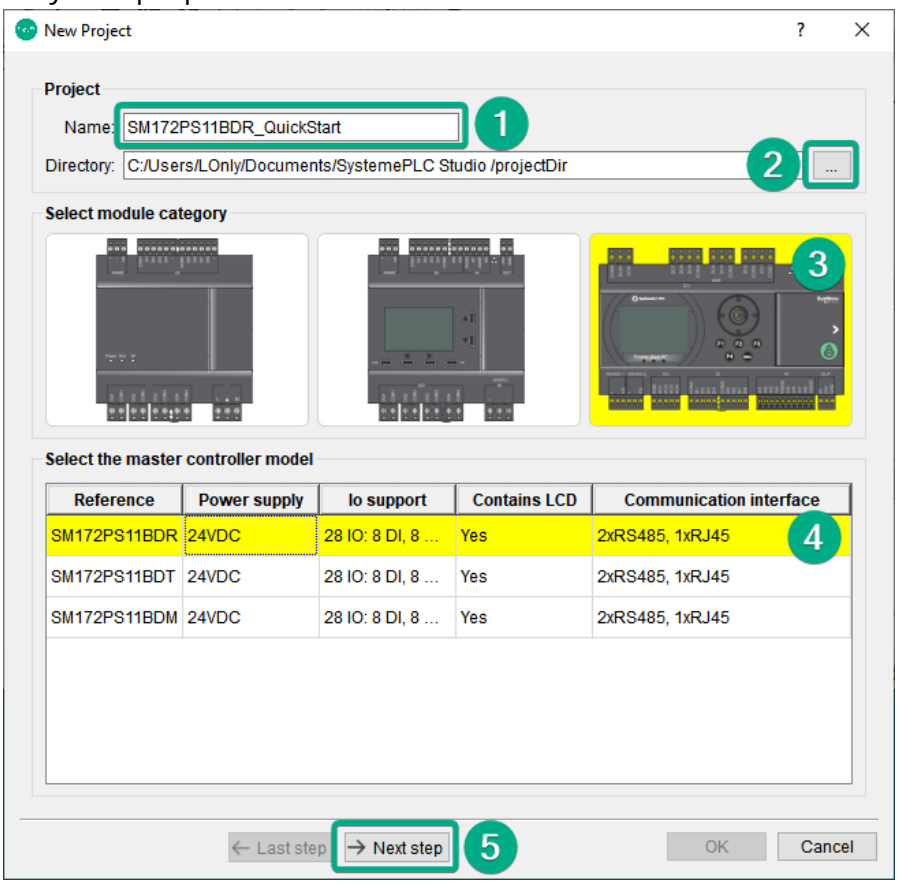
1. В стартовом интерфейсе SystemePLC Studio создать новый проект через меню File > New Project или нажать на значок New Project.



2. В открывшемся диалоговом окне задать:
   a. Имя проекта (Name)
   b. Путь для сохранения проекта (Directory)
   c. Выберите тип контроллера (Select Module category)
   d. Выберите модель контроллера (Select the master controller model)

3. Нажать Next Step

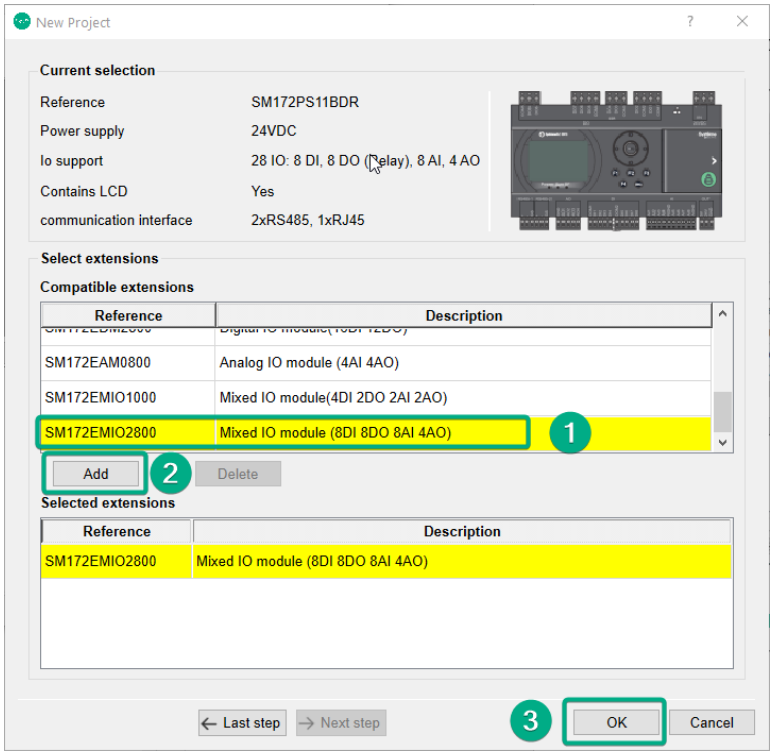
Важно! Имя проекта задается латиницей. Путь сохранения проекта **не должен содержать** кириллицу, китайские иероглифы и символы. Нарушение этого требования приведет к невозможности запуска программы.



Далее можно добавить в конфигурацию модули ввода/вывода. Для этого

4. Выбрать модуль ввода/вывода (**максимум до 7 модулей расширения)**
5. Нажать кнопку Add
6. В завершении нажать OK

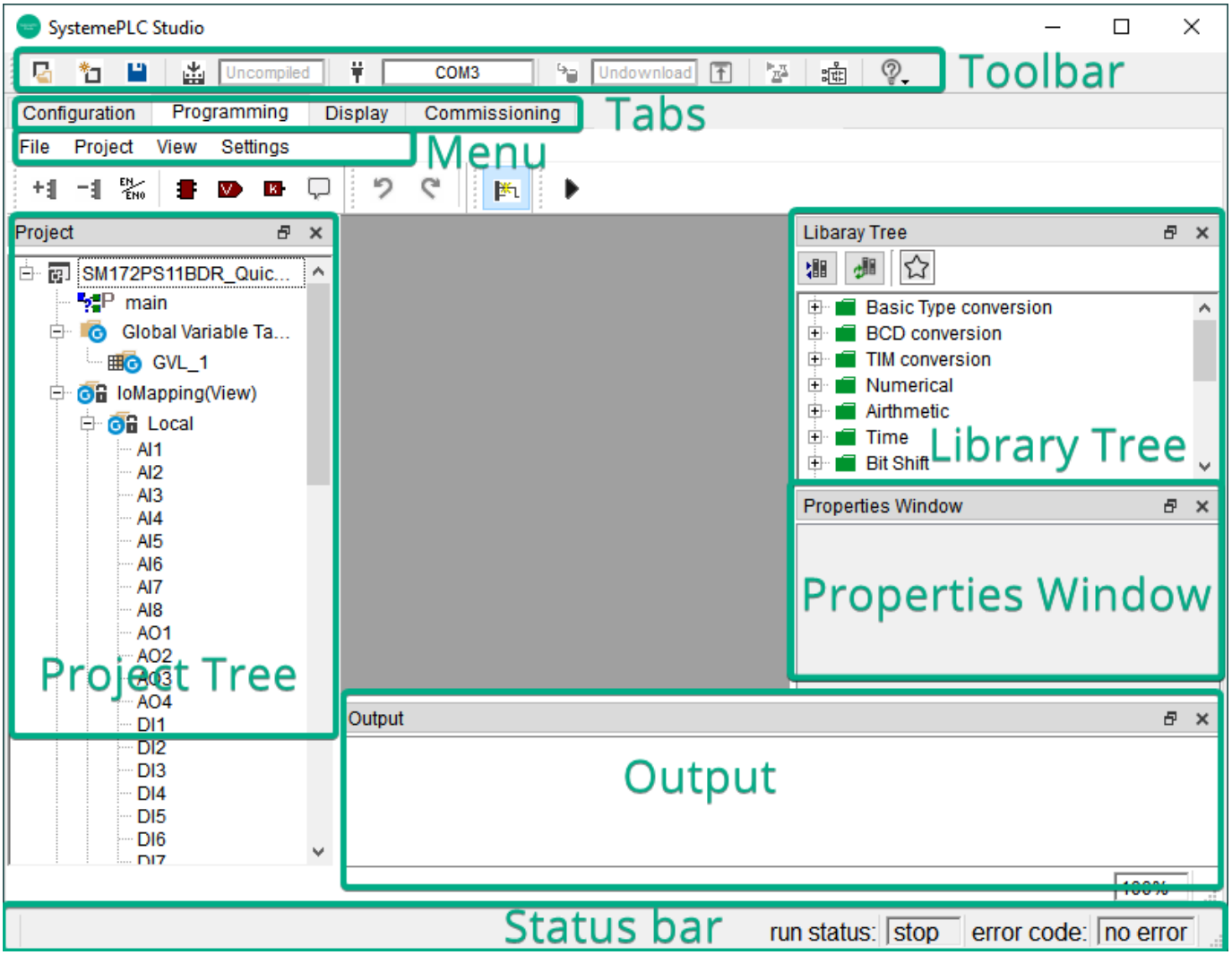
Важно! Добавить модули можно и позже на вкладке Hardware Configuration > Extend. В нашем тестовом проекте не будет модулей расширения, поэтому пропустим этот шаг. Удалите добавленные модули с помощью кнопки Delete (если вы их добавляли).



Основы интерфейса

Базовый интерфейс проекта включает следующие ключевые элементы:

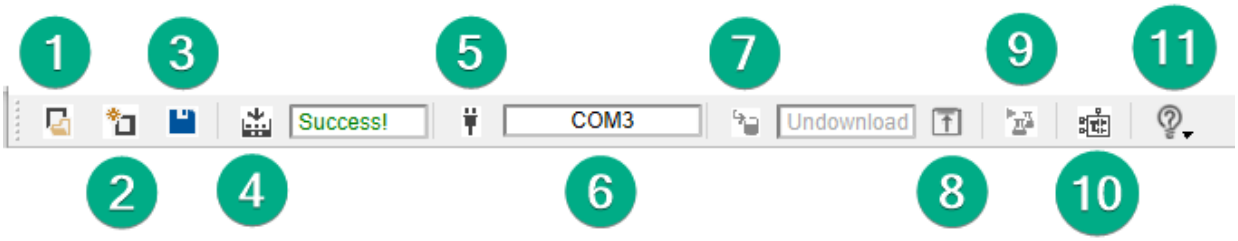
- **Панель инструментов (Toolbar):** обеспечивает быстрый доступ к часто используемым функциям (создание, открытие, сохранение проектов, компиляция, загрузка, соединение, симуляция и др.).
- **Вкладки меню (Tabs):** содержит основные разделы: Configuration (Конфигурация), Programming (Программирование), Display (Отображение), Commissioning (Ввод в эксплуатацию).
- **Пункты меню (Menu):** содержит пункты меню File, Project, View, Settings и др.
- **Дерево проекта (Project Tree):** отображает иерархическую структуру проекта для каждого раздела меню.
- **Библиотека элементов (Library Tree):** содержит библиотеку блоков, их описание, а также
- **Свойства (Properties Window):** показывает свойства или описание выбранных элементов.
- **Поле вывода (Output):** отображает текущий статус проекта (информация о компиляции, загрузке, сохранении).
- **Строка состояния (Status bar):** отображает текущий режим ПЛК (Run/Stop) и код текущей ошибки (error code).



Панель инструментов (Toolbar)

Описание элементов панели инструментов

1. Открытие проекта (Open project)
2. Создание нового проекта (New Project)
3. Сохранение текущего проекта (Save Project)
4. Компиляция проекта (+ результат компиляции проекта справа) [Compile PLC Project]
5. Соединение с ПЛК (Connect to the target)
6. Настройка соединения с ПЛК (Modbus Configuration)
7. Загрузка программы в ПЛК (+ результат загрузки программы в ПЛК справа) [Download PLC code]
8. Выгрузка проекта из ПЛК на ПК (Upload Project)
9. Режим онлайн мониторинга (Live Debug Mode)
10. Режим оффлайн симуляции ПЛК (Simulation Mode)
11. Руководство пользователя (Help) и информация о программе (About)



Вкладки меню (Tabs): содержит четыре основные вкладки

- **Configuration (Конфигурация):** настройка аппаратной части проекта.
  - Определение модели контроллера и модулей расширения.
  - Настройка последовательности модулей на шине.
  - Конфигурация карты регистров Modbus (сетевые переменные для SCADA и панелей оператора).
  - Настройка портов (RS-485, Modbus RTU/Master, Modbus TCP/IP Client).
- **Programming (Программирование):** разработка логики управления.
  - Создание и редактирование программ, функциональных блоков (FB) и функций (FC).
  - Основной язык программирования — CFC (Continuous Function Chart). В будущем планируется добавление LD (Ladder Diagram) и ST (Structured Text).
- **Display (Дисплей):** проектирование интерфейсов для встроенного дисплея контроллера.
  - Создание экранов с элементами ввода, вывода, кнопками и навигацией.
- **Commissioning (Ввод в эксплуатацию):** сервисные функции.
  - Онлайн-мониторинг и изменение параметров контроллера.
  - Диагностика состояния входов/выходов.
  - Настройка сетевых параметров и синхронизации времени.

Подключение ПЛК к ПК через USB Type-C*

**Пропустите этот шаг если у вас нет физического ПЛК.*

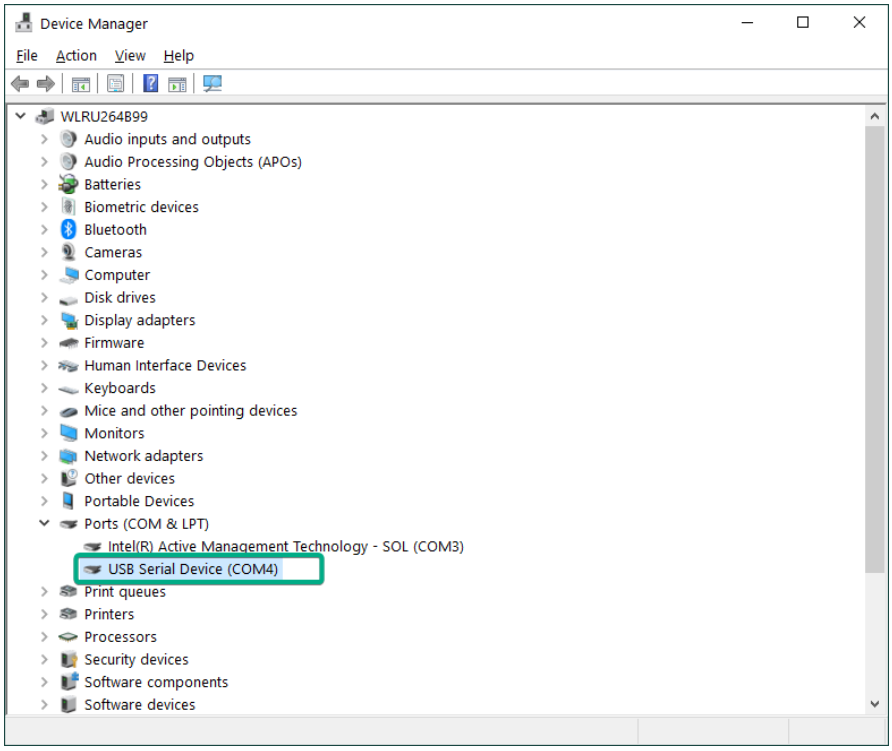
ПО SystemePLC Studio поддерживает два протокола связи с контроллером: **Modbus TCP** и **Modbus RTU**.

ПО SystemePLC Studio поддерживает три физических интерфейса для подключения ПК к ПЛК

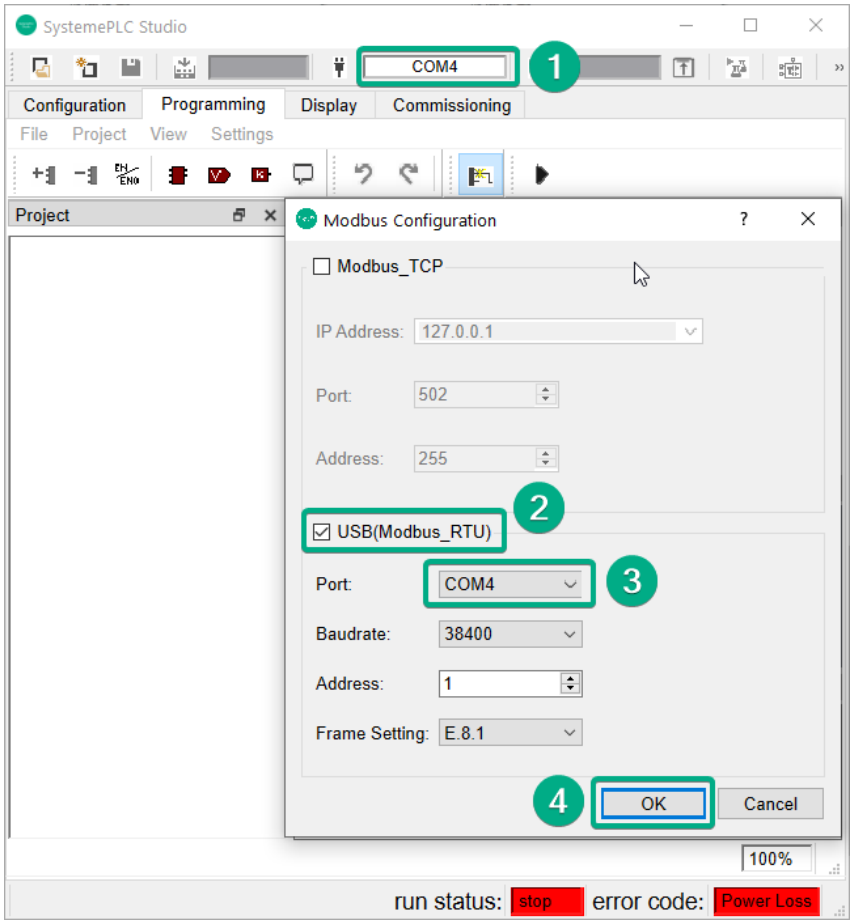
- **Через порт USB Type-C:** используется кабель USB – USB Type-C. Связь осуществляется по протоколу **Modbus RTU**.
- **Через порт RJ45 Ethernet:** используется стандартный сетевой кабель (RJ45). Связь осуществляется по протоколу **Modbus TCP**.
- **Через порт RS485:** используется кабель USB – RS485. Связь осуществляется по протоколу **Modbus RTU**.

Процедура связи через кабель USB Type-C по протоколу Modbus RTU

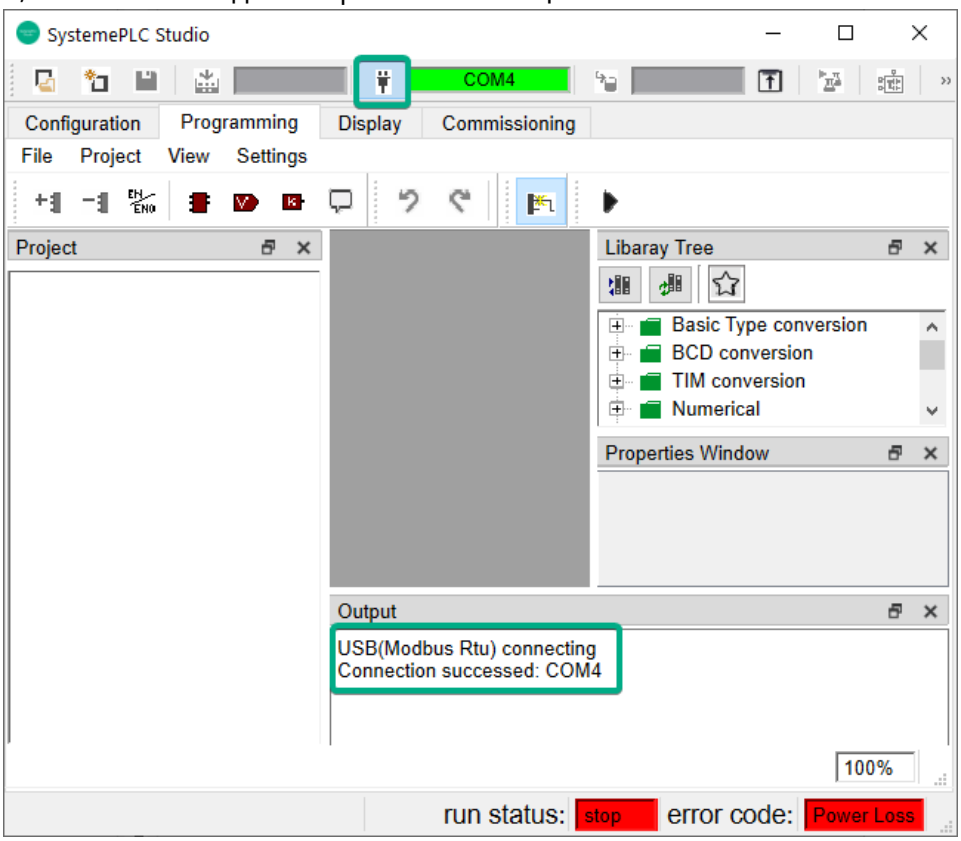
1. Установите на ПК драйвер последовательного порта, соответствующий используемому кабелю (например, USB-to-RS485). Для кабеля USB Type-C Windows может определить драйвер самостоятельно.
2. Подключите ПЛК к ПК кабелем USB Type-C. Подавать внешнее питание на ПЛК при этом не обязательно.
3. Определите, какой **COM-порт** был назначен ПЛК в системе (через **Диспетчер устройств Windows**).



4. В ПО SystemePLC Studio нажмите на окно настройки соединения с устройством
5. В открывшемся диалоговом окне Modbus Configuration выберите USB (Modbus RTU).
6. Задайте параметры:
   a. **Port:** назначенный ПЛК COM-порт из диспетчера устройств Windows (например, COM4).
   b. Остальные параметры можно оставить без изменения.
      (1) **Baudrate:** скорость обмена (должна совпадать с настройкой ПЛК).
      (2) **Address:** адрес устройства.
      (3) **Frame Setting:** настройки кадра (например, E.8.1 — Четность 'Even', 8 бит данных, 1 стоп-бит).



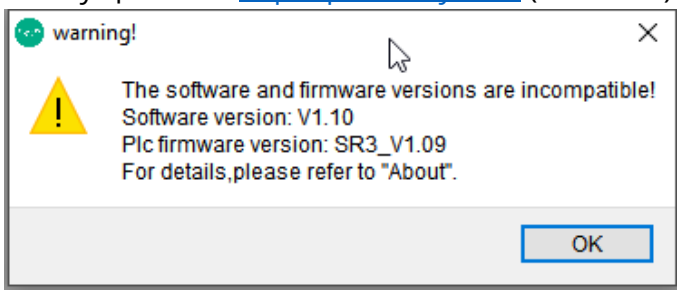
7. Нажмите значок **подключения**. При успешном соединении индикатор станет зеленым, а в окне вывода отобразится сообщение.



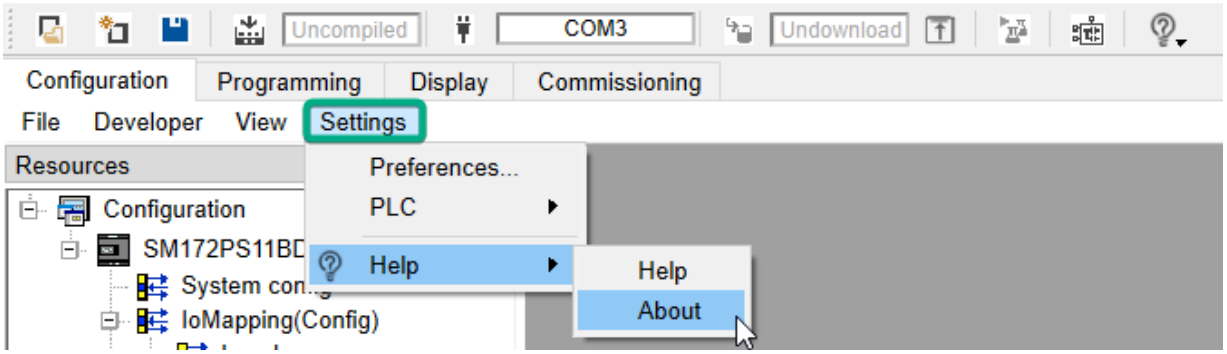
Проверка соответствия ВОС ПЛК и версии ПО SystemePLC Studio*

**Пропустите этот шаг если у вас нет физического ПЛК.*

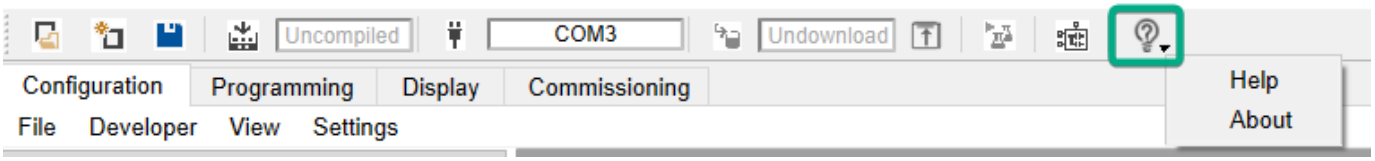
Важно! При первом подключении к ПЛК может появится окно с предупреждением о несоответствии версии ВОС ПЛК текущей версии ПО SystemePLC Studio.
В этом случае необходимо осуществить перепрошивку ПЛК (см. ниже).



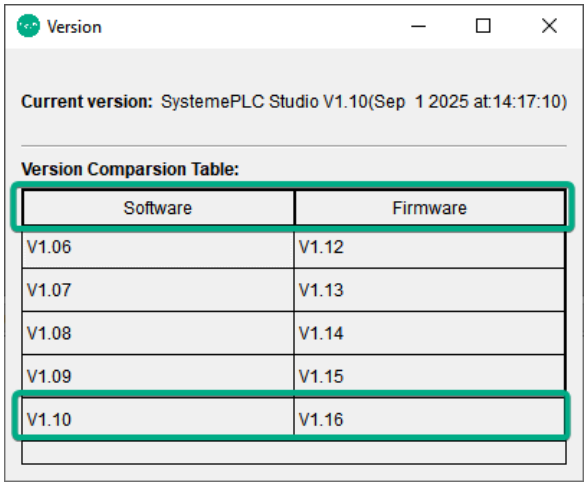
Важно! Посмотреть соответствие версии ВОС ПЛК текущей версии ПО SystemePLC Studio можно выбрав пункт меню Settings > Help > About или нажать на значок «?» > About.



ИЛИ



В открывшемся окне **Version** посмотрите на соответствие версии ВОС ПЛК и установленной версии ПО SystemePLC Studio.



Software	Firmware
V1.06	V1.12
V1.07	V1.13
V1.08	V1.14
V1.09	V1.15
V1.10	V1.16

Перепрошивка ПЛК*

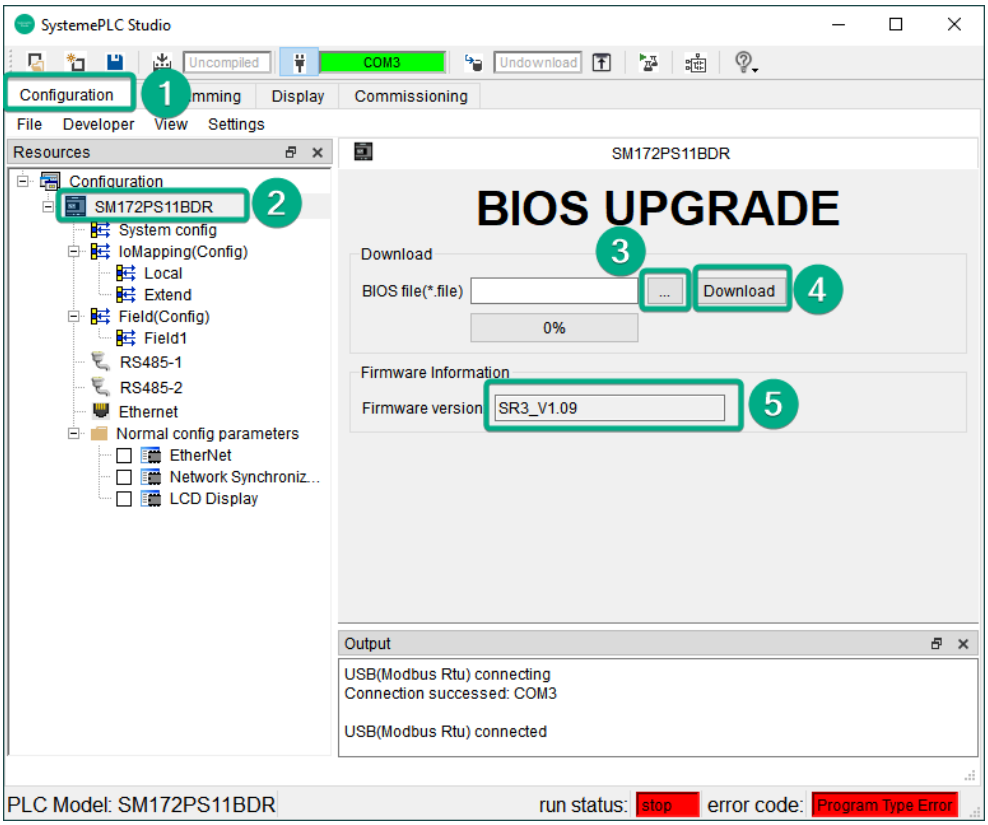
**Пропустите этот шаг если у вас нет физического ПЛК.*

Загрузка внутренней операционной системы (ВОС) [прошивки] устройства SystemePLC SR или S172 может быть выполнена как с подключением к ПК, так и без него (это зависит от модели устройства) через флеш-карту памяти USB Type-A.

Загрузка ВОС с подключением к ПК (доступно для интеллектуальных реле SystemePLC SR и ПЛК SystemePLC S172)

Для установки новой версии ВОС необходимо:

1. Подключиться к обновляемому устройству (интеллектуальному реле или ПЛК). Это можно сделать с помощью:
   a. Кабеля USB (порт USB Type-C);
   b. Кабеля Ethernet (разъём RJ45);
   c. Витой пары (порт RS-485).
2. Выбрать устройство во вкладке Configuration двойным нажатием ЛКМ
3. Выбрать файл с новой версией ВОС
4. Нажать кнопку Download
5. Дождаться загрузки ВОС и автоматической перезагрузки устройства (ПЛК).
6. После повторного подключения к ПЛК убедится, что в поле Firmware version отображается советующая версия ВОС, которую вы загрузили.



Загрузка ВОС через флеш-карту памяти USB Type-A без подключения к ПК (доступна только для ПЛК SystemePLC S172)

Важно! максимальный объем карты памяти до 32 GB; формат: FAT32; размер сектора: 4KB.

Для загрузки новой версии ВОС с помощью флеш-карты USB Type-A необходимо:

1. Задать имя файла ВОС: firmware.bin
2. Записать файл ВОС в корневую папку флеш-карты
3. Подключить внешнее питание ПЛК
4. Вставить в разъем ПЛК USB Type-A флеш-карту памяти, содержащую файл с новой версией ВОС.
5. Нажать любую клавишу на ПЛК.
6. С помощью кнопок «вверх» и «вниз» выбрать раздел USB Flash и нажать кнопку OK.
7. Выбрать Load firmware и нажать OK.
8. Дождаться загрузки ВОС и автоматической перезагрузки устройства (ПЛК).

Конфигурация аппаратной части ПЛК

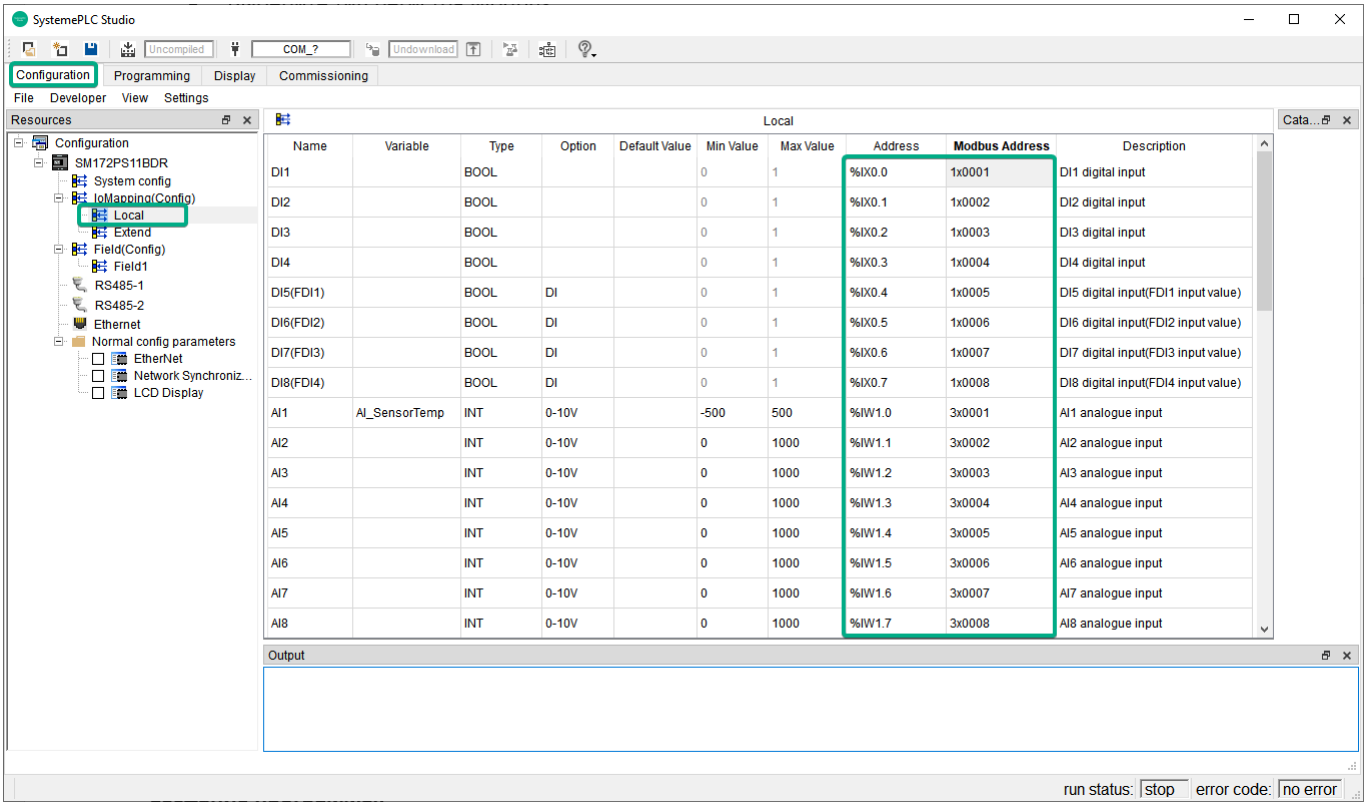
Перейдите на вкладку **Конфигурация (Configuration).**

После создания проекта необходимо выполнить конфигурацию аппаратного обеспечения – процесс, заключающийся в компоновке ПЛК и его модулей в соответствии с их физическими соединениями и задании их моделей и параметров.

Сопоставление входов/выходов (IO Mapping) делится на:

- Локальные входы/выходы (Local IO) – в/в ПЛК
- Расширенные входы/выходы (Extended IO) – в/в модулей расширения

Система автоматически распределяет адреса входов/выходов и соответствующие им адреса Modbus.



Name	Variable	Type	Option	Default Value	Min Value	Max Value	Address	Modbus Address	Description
DI1		BOOL			0	1	%IX0.0	1x0001	DI1 digital input
DI2		BOOL			0	1	%IX0.1	1x0002	DI2 digital input
DI3		BOOL			0	1	%IX0.2	1x0003	DI3 digital input
DI4		BOOL			0	1	%IX0.3	1x0004	DI4 digital input
DI5(FDI1)		BOOL	DI		0	1	%IX0.4	1x0005	DI5 digital input(FDI1 input value)
DI6(FDI2)		BOOL	DI		0	1	%IX0.5	1x0006	DI6 digital input(FDI2 input value)
DI7(FDI3)		BOOL	DI		0	1	%IX0.6	1x0007	DI7 digital input(FDI3 input value)
DI8(FDI4)		BOOL	DI		0	1	%IX0.7	1x0008	DI8 digital input(FDI4 input value)
AI1	AI_SensorTemp	INT	0-10V		-500	500	%IW1.0	3x0001	AI1 analogue input
AI2		INT	0-10V		0	1000	%IW1.1	3x0002	AI2 analogue input
AI3		INT	0-10V		0	1000	%IW1.2	3x0003	AI3 analogue input
AI4		INT	0-10V		0	1000	%IW1.3	3x0004	AI4 analogue input
AI5		INT	0-10V		0	1000	%IW1.4	3x0005	AI5 analogue input
AI6		INT	0-10V		0	1000	%IW1.5	3x0006	AI6 analogue input
AI7		INT	0-10V		0	1000	%IW1.6	3x0007	AI7 analogue input
AI8		INT	0-10V		0	1000	%IW1.7	3x0008	AI8 analogue input

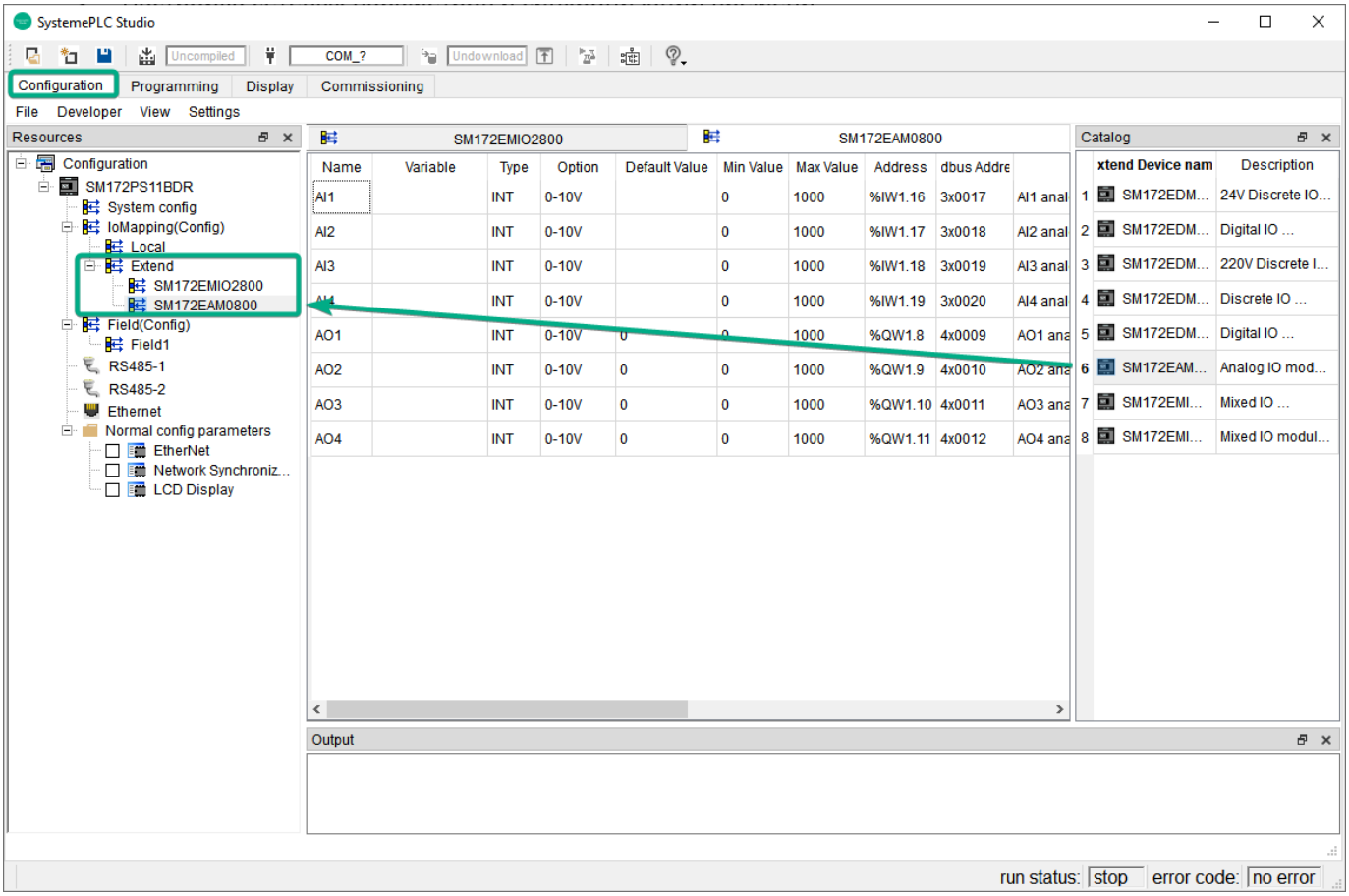
Конфигурация контроллера и модулей

В **ресурсах проекта (Resources)** можно добавить необходимые модули расширения через IoMapping (Config) > Extend

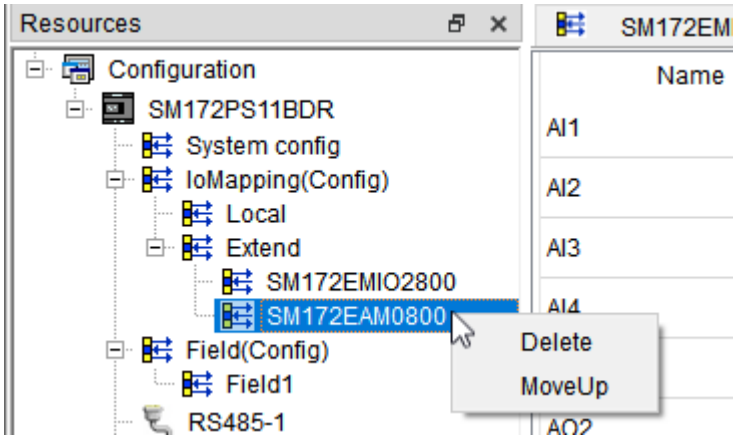
Каждый контроллер SR2 или S172 поддерживает подключение до **7 модулей расширения**.

Для этого перейдите на вкладку **Конфигурация (Configuration)**

1. Нажмите ЛКМ на папку IoMapping (Config) > Extend
2. Перетащите ЛКМ необходимые модули из каталога в папку Extend



3. Определите их порядок на шине в соответствии с их физическим порядком подключения.
   a. Нажмите на модуль ПКМ > MoveUp или ПКМ > MoveDown для изменения порядка подключения
   b. Нажмите на модуль ПКМ > Delete чтобы удалить его из конфигурации

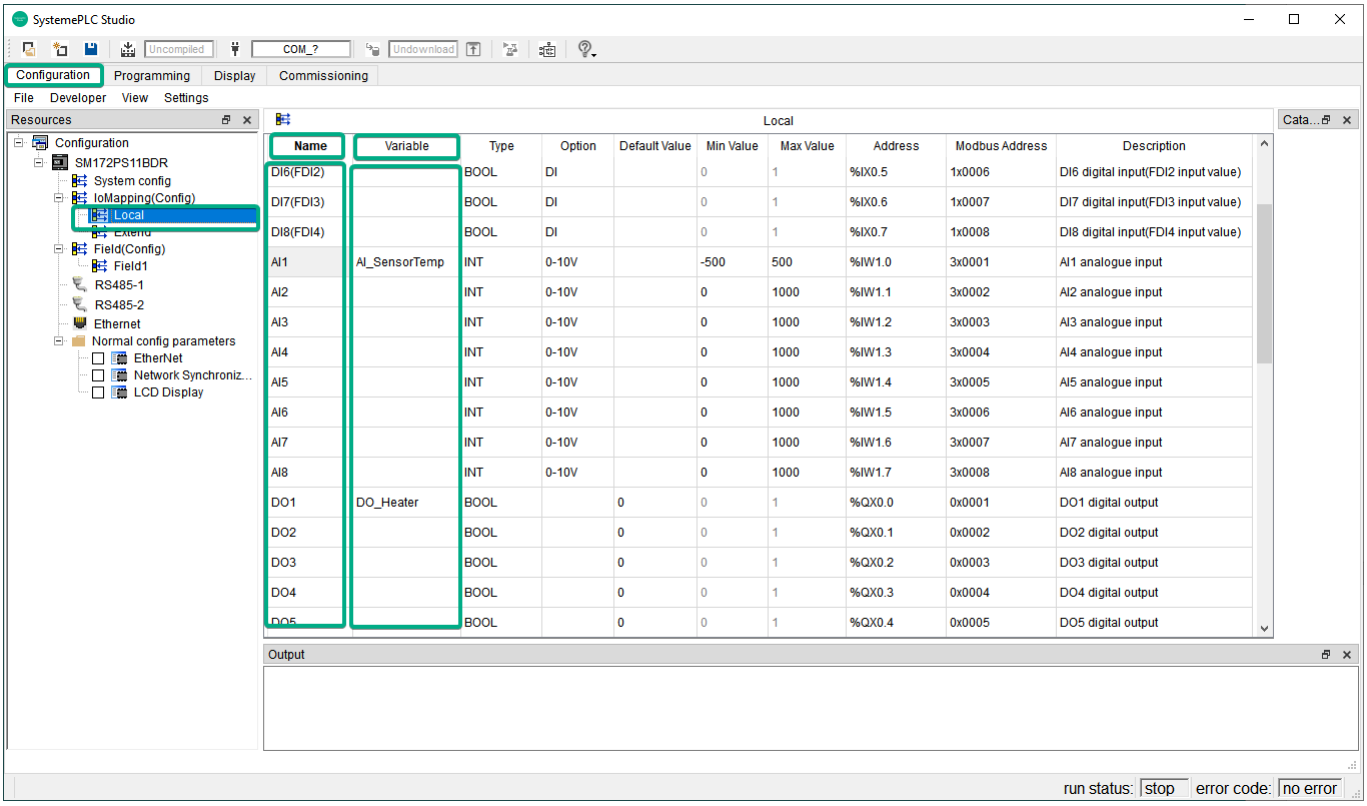


Важно! В нашем тестовом проекте не будет модулей расширения, поэтому пропустим этот шаг. Если вы уже добавили модули, то удалите их из проекта.

Связывание переменных с физическими входами/выходами (IO Mapping)

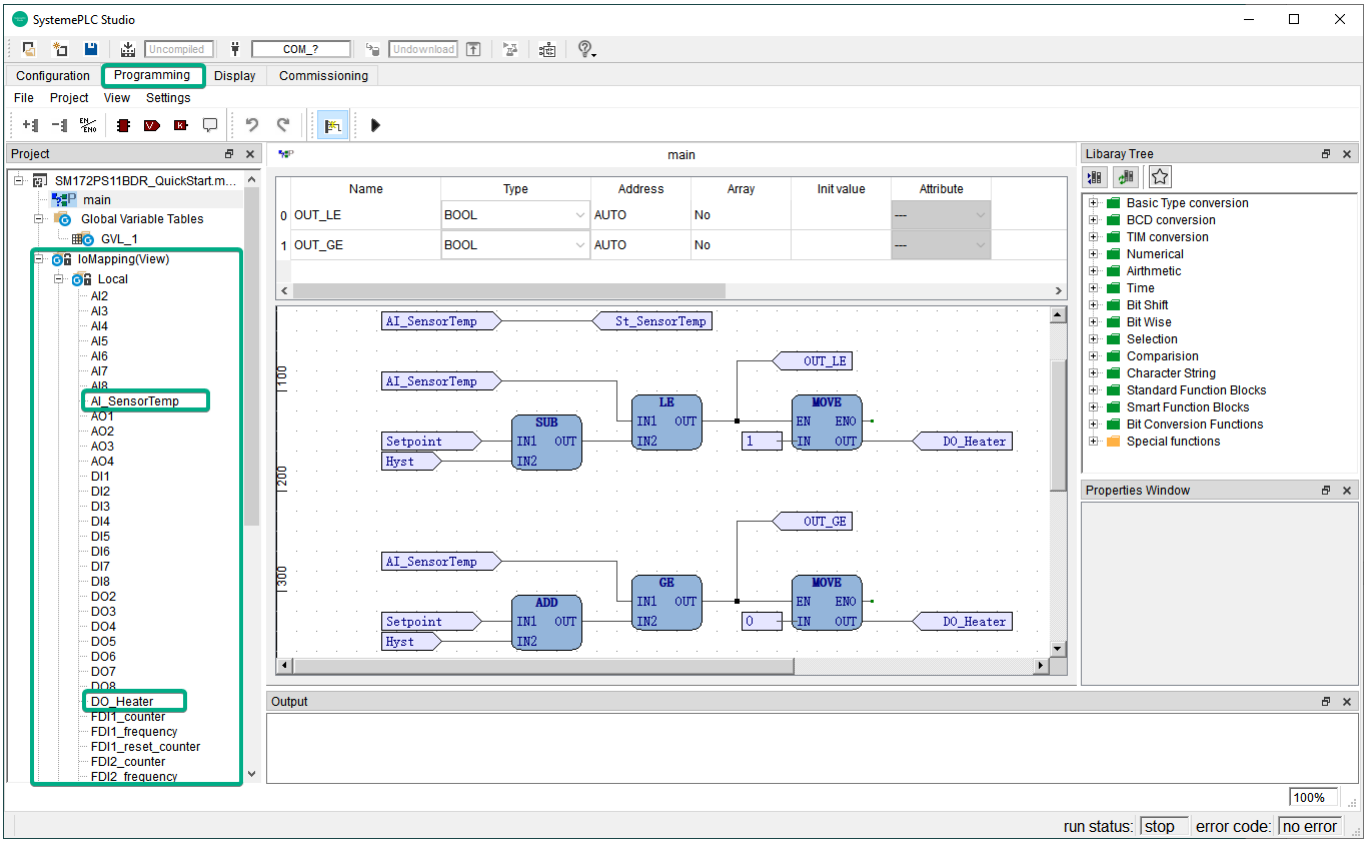
Для ассоциации переменных в программе с физическими входами/выходами необходимо задать имя переменной в таблице IoMapping (Config) > Local

- Каждый вход/выход имеет имя переменной по умолчанию, заданный автоматически в поле **Name**
- В поле **Variable** можно задать пользовательские имена соответствующим переменным (например, зададим входу AI1 имя AI_SensorTemp, а выходу DO1 имя DO_Heater)



Name	Variable	Type	Option	Default Value	Min Value	Max Value	Address	Modbus Address	Description
DI6(FDI2)		BOOL	DI		0	1	%IX0.5	1x0006	DI6 digital input(FDI2 input value)
DI7(FDI3)		BOOL	DI		0	1	%IX0.6	1x0007	DI7 digital input(FDI3 input value)
DI8(FDI4)		BOOL	DI		0	1	%IX0.7	1x0008	DI8 digital input(FDI4 input value)
AI1	AI_SensorTemp	INT	0-10V		-500	500	%IW1.0	3x0001	AI1 analogue input
AI2		INT	0-10V		0	1000	%IW1.1	3x0002	AI2 analogue input
AI3		INT	0-10V		0	1000	%IW1.2	3x0003	AI3 analogue input
AI4		INT	0-10V		0	1000	%IW1.3	3x0004	AI4 analogue input
AI5		INT	0-10V		0	1000	%IW1.4	3x0005	AI5 analogue input
AI6		INT	0-10V		0	1000	%IW1.5	3x0006	AI6 analogue input
AI7		INT	0-10V		0	1000	%IW1.6	3x0007	AI7 analogue input
AI8		INT	0-10V		0	1000	%IW1.7	3x0008	AI8 analogue input
DO1	DO_Heater	BOOL		0	0	1	%QX0.0	0x0001	DO1 digital output
DO2		BOOL		0	0	1	%QX0.1	0x0002	DO2 digital output
DO3		BOOL		0	0	1	%QX0.2	0x0003	DO3 digital output
DO4		BOOL		0	0	1	%QX0.3	0x0004	DO4 digital output
DO5		BOOL		0	0	1	%QX0.4	0x0005	DO5 digital output

После установления связи соответствующие переменные IO автоматически появятся в разделе Programming > IoMapping (View) и могут быть напрямую перетянуты в область программирования для вызова.



Важно! В программе можно использовать физические входы с именами по умолчанию из поля Name (например: AI2, AI3, DI1, DO1 и др.) или задать им пользовательские имена в поле Variable.

Настройка физических входов/выходов (IO Mapping Local):

В таблице IoMapping (Config) > Local часть параметров для физических входов/выходов задаются автоматически, а часть можно задать вручную при необходимости.

Задаются автоматически по умолчанию для входа/выхода

- Name – имя входа/выхода
- Type – тип входа/выхода
- Address – адрес во внутренней памяти ПЛК
- Modbus Address – Modbus адрес
- Description – описание

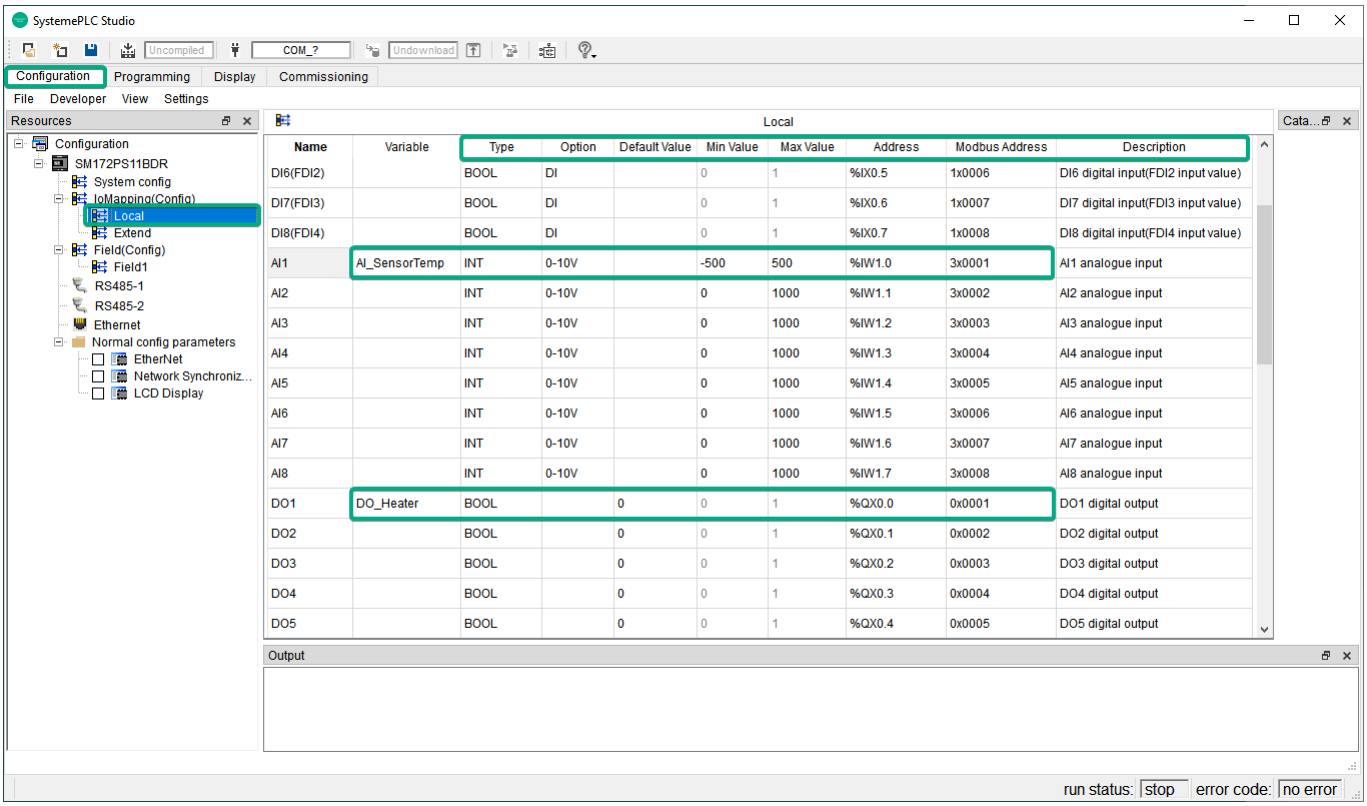
Можно задать вручную для входа/выхода

- Variable – пользовательское имя переменной
- **Option** – тип сигнала для входа/выхода (0-10В, 4-20мА или др.)
- Default Value – значение по умолчанию
- Min Value – минимально возможное принимаемое значение
- Max Value – максимально возможное принимаемое значение

Демонстрационный проект Шаг1

Для демонстрационного проекта зададим

- Для аналогового входа AI1: Variable: AI_SensorTemp; Option: 0-10 V; Min Value = -500; Max Value = 500.
- Для дискретного выхода DO1: Variable: AI_SensorTemp.



Name	Variable	Type	Option	Default Value	Min Value	Max Value	Address	Modbus Address	Description
DI6(FDI2)		BOOL	DI		0	1	%IX0.5	1x0006	DI6 digital input(FDI2 input value)
DI7(FDI3)		BOOL	DI		0	1	%IX0.6	1x0007	DI7 digital input(FDI3 input value)
DI8(FDI4)		BOOL	DI		0	1	%IX0.7	1x0008	DI8 digital input(FDI4 input value)
AI1	AI_SensorTemp	INT	0-10V		-500	500	%IW1.0	3x0001	AI1 analogue input
AI2		INT	0-10V		0	1000	%IW1.1	3x0002	AI2 analogue input
AI3		INT	0-10V		0	1000	%IW1.2	3x0003	AI3 analogue input
AI4		INT	0-10V		0	1000	%IW1.3	3x0004	AI4 analogue input
AI5		INT	0-10V		0	1000	%IW1.4	3x0005	AI5 analogue input
AI6		INT	0-10V		0	1000	%IW1.5	3x0006	AI6 analogue input
AI7		INT	0-10V		0	1000	%IW1.6	3x0007	AI7 analogue input
AI8		INT	0-10V		0	1000	%IW1.7	3x0008	AI8 analogue input
DO1	DO_Heater	BOOL		0	0	1	%QX0.0	0x0001	DO1 digital output
DO2		BOOL		0	0	1	%QX0.1	0x0002	DO2 digital output
DO3		BOOL		0	0	1	%QX0.2	0x0003	DO3 digital output
DO4		BOOL		0	0	1	%QX0.3	0x0004	DO4 digital output
DO5		BOOL		0	0	1	%QX0.4	0x0005	DO5 digital output

Важно! Для аналоговых сигналов сразу настройте тип сигнала (0-10В, 4-20мА или др.) и масштабирование (Scaling) с помощью задания Min Value и Max Value.
Например, для датчика температуры -50°C...+50°C, который передает сигнал 0-10 В, задайте: MinValue = -500, MaxValue = 500.

Значения аналогового входа передается в формате INT (целое число). Поэтому для работы с дробными значениями температуры (например, +23,3 C) будет удобно задать масштабирование в 10 раз больше диапазона температуры (от -500 до +500). Это позволит избежать дополнительных операций преобразования типов REAL > INT и INT > REAL.
Тогда

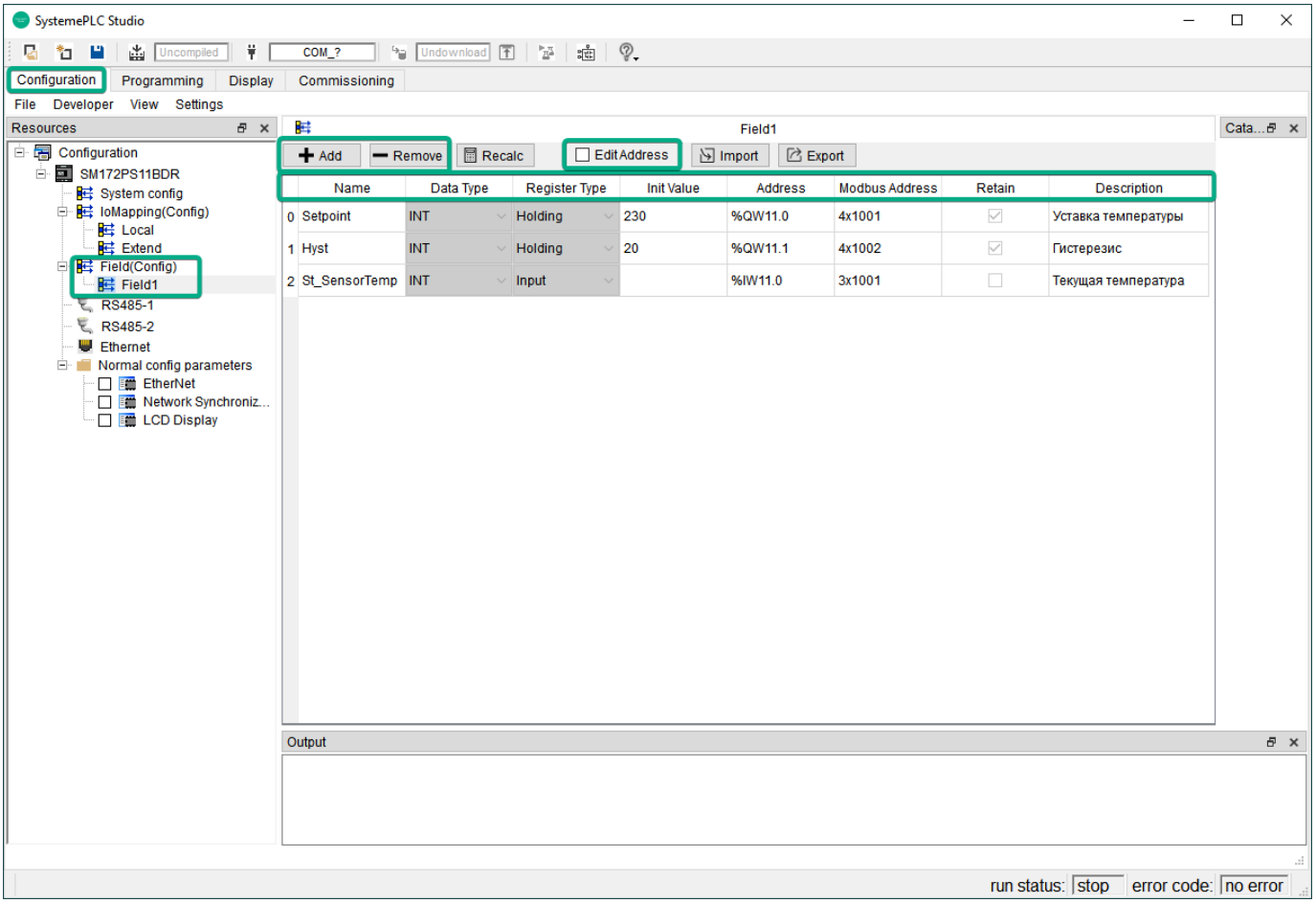
- 0 V на входе будет соответствовать -500 (-50 C)
- 5 V на входе будет равно 0 (0 C)
- 10 V на входе будет соответствовать +500 (+50 C)

Создание глобальных сетевых переменных (Field)

Глобальные сетевые переменные доступны извне, например из SCADA-систем, панели оператора или другого ПЛК.

Создание глобальных сетевых переменных (Field)

1. Перейдите в Configuration > Field (Config) > Field1
2. Задайте имя переменной
3. Укажите тип переменной (INT, BOOL и т.д.)
4. Выберите тип регистра Modbus
   a. Input (только для чтения, адресация 3xxxx)
   b. Holding (для чтения и записи, адресация 4xxxx)
5. Значение по умолчанию (Init Value) будет присваиваться переменной при старте ПЛК
6. Адрес во внутренней памяти ПЛК (Address) и адрес Modbus (Modbus Address) назначаются автоматически и последовательно
   a. При необходимости Modbus Address можно задавать вручную, для этого необходимо поставить галку Edit Address
7. Опция Retain: позволяет сохранять значение переменной после сброса питания (энергонезависимая flash-память). Рекомендуется для уставок.
8. В поле Description можно задать пользовательское описание переменной



Демонстрационный проект Шаг2

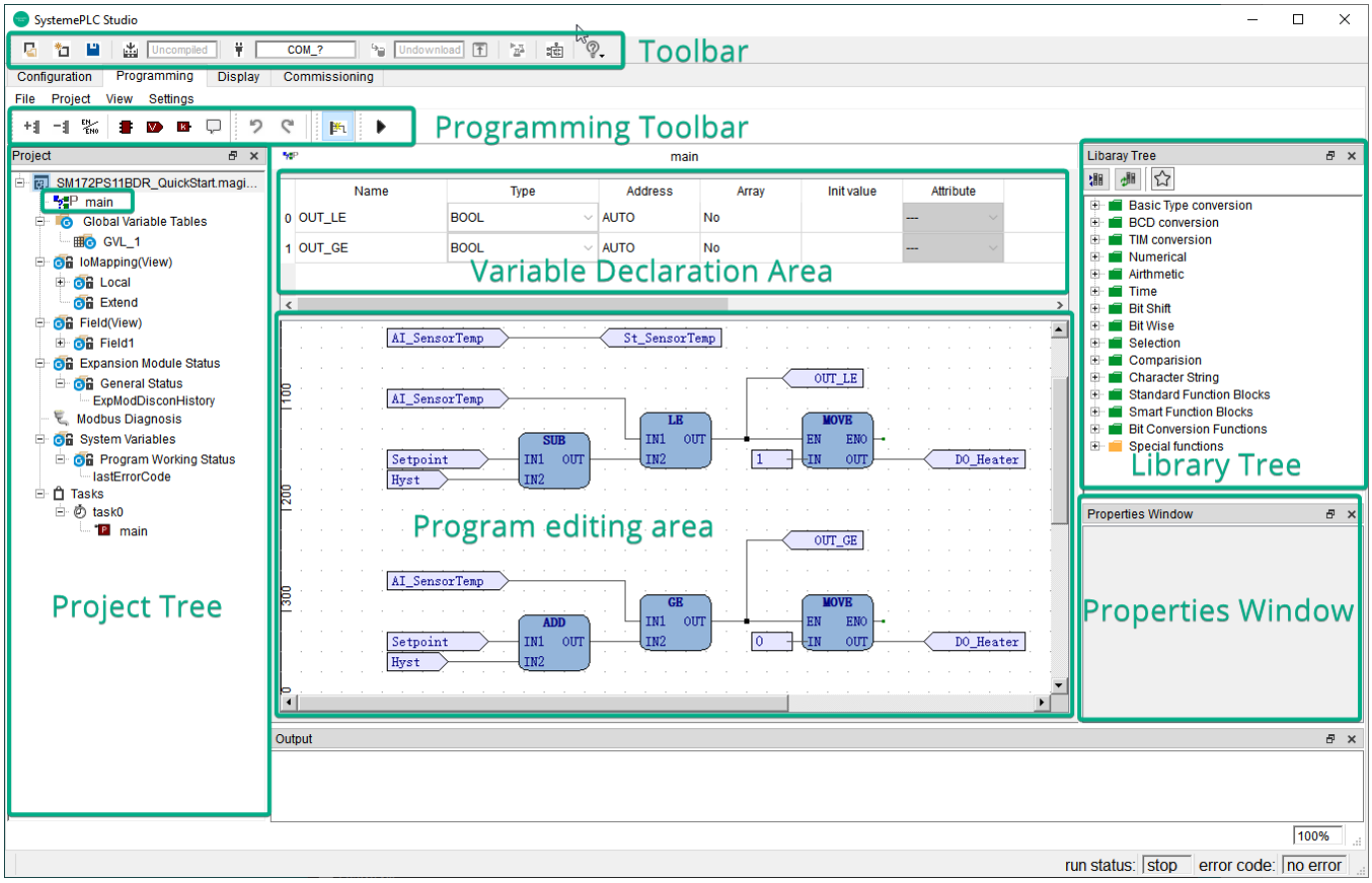
Для демонстрационного проекта создадим следующие переменные (см. скриншот выше)

- Name: Setpoint; Data type: INT; Register Type: Holding; Init Value: 230; Retain: On; Description: «Уставка температуры».
- Name: Hyst; Data type: INT; Register Type: Holding; Init Value: 20; Retain: On; Description: «Гистерезис».
- Name: St_SensorTemp; Data type: INT; Register Type: Input; Init Value: 230; Retain: On; Description: «Текущая температура».

Основы программирования

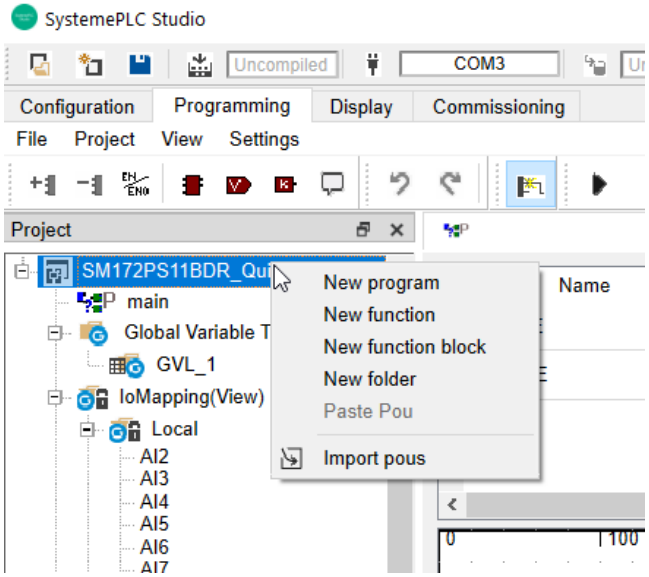
Интерфейс вкладки Программирование (Programming)

- **Панель инструментов (Toolbar):** обеспечивает быстрый доступ к часто используемым функциям (создание, открытие, сохранение проектов, компиляция, загрузка, соединение, симуляция и др.).
- **Панель инструментов программирования (Programming Toolbar):** обеспечивает быстрый доступ к часто используемым инструментам (функциональные блоки, переменные и др.), действиям (Undo/Redo), режимам работы ПЛК (Run/Stop) и др.
- **Основная программа (Main Program):** Каждый проект по умолчанию содержит программу **main**, которую можно редактировать в разделе Programming > Main.
- **Область объявления переменных (Variable Declaration Area):** позволяет вручную изменять имена локальных переменных, их типы данных и другие атрибуты.
- **Область написания программного кода (Program editing area)** служит для написания логики программы с помощью программных элементов и связей.
- **Библиотека элементов (Library Tree):** содержит библиотеку блоков, их описание, а также
- **Свойства (Properties Window):** показывает свойства или описание выбранных элементов.



Для структурирования кода можно создавать программы (Program operation Unit [POU]), пользовательские функциональные блоки (FB), пользовательские функции (Function), а также папки (Folder) для их систематизации и структурирования. Для этого

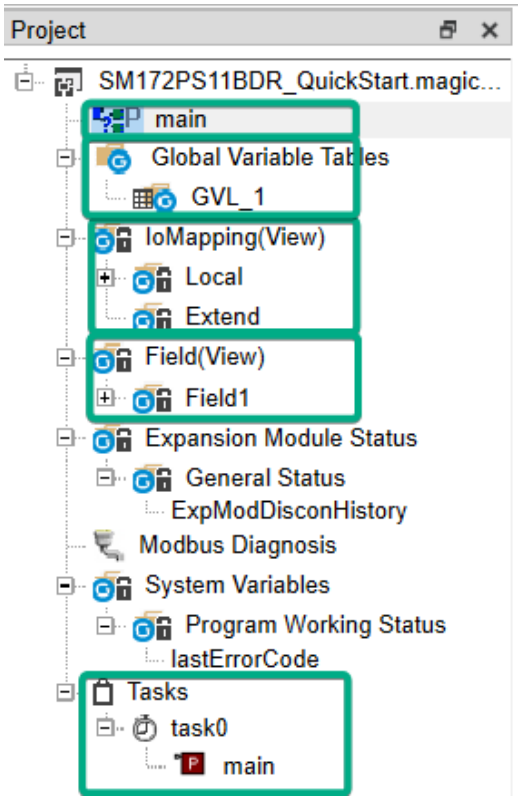
- Нажмите ПКМ на имени проекта в Project Tree и выберите соответствующую опцию
- Для вызова созданного блока в основной программе main достаточно перетянуть созданный блок в область программирования



Дерево проекта (Project Tree)

Дерево проекта (Project Tree) содержит следующие элементы

- Programs: содержит все программы (POU). Программа Main создается по умолчанию
- IO Mapping (View): Переменные, привязанные к физическим входам/выходам (только просмотр и использование в программах)
- Field (View): сетевые глобальные переменные (только просмотр и использование в программах)
- Global Variable Table: глобальные переменные без адресации Modbus (создание и использование в программах)
- Tasks: настройка задач выполнения программ
- И другие элементы



Типы переменных

В SystemePLC Studio различают следующие типы переменных:

1. **Связанные с физическими входами/выходами (IO Mapping)**
   a. Имеют адрес Modbus (доступны извне)
   b. Доступны во всех программах проекта.
   c. Объявляются в Configuration > IoMapping (Config)
2. **Глобальные сетевые (Field)**
   a. Имеют адрес Modbus (доступны извне). Ограничение ~2000 адресных переменных.
   b. Доступны во всех программах проекта
   c. Объявляются в Configuration > Field (Config)
3. **Глобальные (Global Variable Table)**

   a. **Не имеют адреса Modbus. Экономят лимит глобальных сетевых переменных.**
   b. Доступны во всех программах проекта
   c. Объявляются в Programming > Global Variables Table
4. **Локальные (Local)**
   a. Не имеют адреса Modbus
   b. Доступны только внутри своей программы
   c. Объявляются внутри каждой программы (например, Programming > main)

Важно! Переименование переменных. После использования переменной в программе ее переименование в Configuration или Programming **не обновляет** имя в коде автоматически. Необходимо вручную изменить имя во всех местах использования, иначе возникнет ошибка компиляции.

Задачи (Tasks)

Задачи служат для настройки порядка выполнения программ.

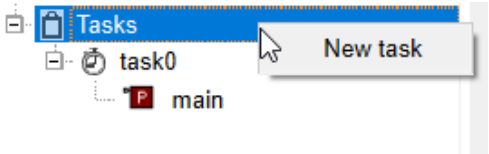
- Приоритет выполнения задач и программ внутри них регулируется кнопками Move Up и Move Down
- В одну задачу можно добавить несколько программ; они выполняются последовательно сверху вниз

Типы задач (Tasks)

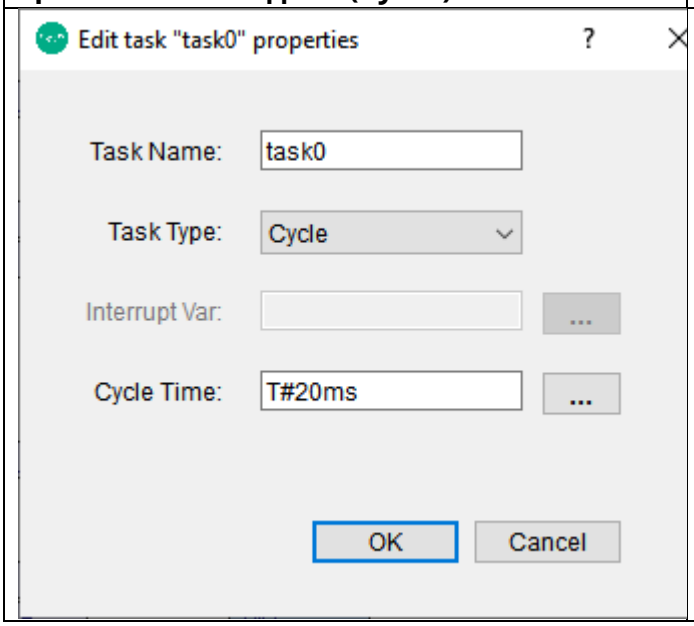
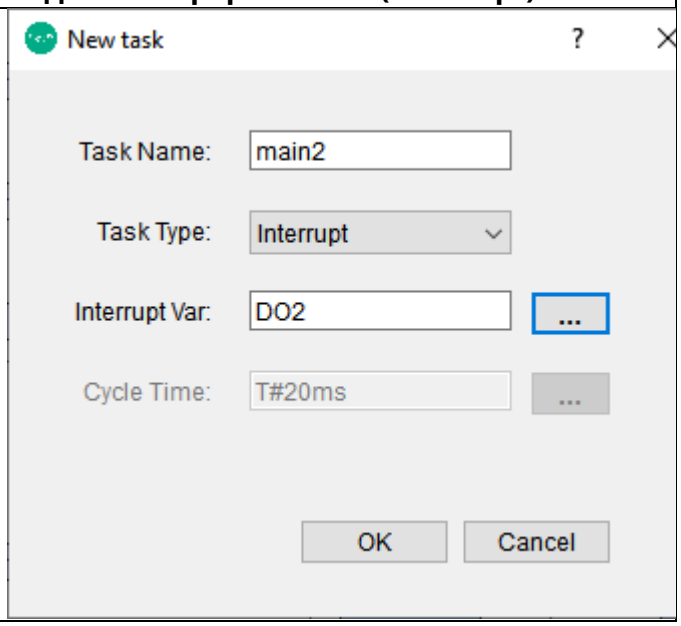
- Циклическая задача (Cyclic): программа выполняется постоянно с заданным циклом (Cycle Time)
- Задача по прерыванию (Interrupt): программа выполняется однократно по переднему фронту булевой переменной

Создание задач

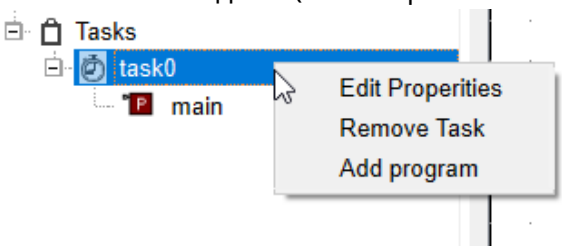
1. Для создания задач нажмите ПКМ по папке Tasks > New task



2. Задайте параметры задачи:
   a. Name: имя задачи
   b. Task Type: тип задачи (Cyclic или Interrupt)
   c. Cycle Time (только для Cyclic Task): время выполнения циклической задачи (выполняется через каждый заданные интервал времени)
   d. Interrupt Var (только для Interrupt Task) – имя переменной по которой необходимо выполнить задачу по прерыванию

Циклическая задача (Cyclic)	Задача по прерыванию (Interrupt)
	

3. Нажмите по имени задачи ПКМ для
   a. изменения ее свойств (Edit Properties)
   b. удаления (Remove task)
   c. добавления программы в задачу (Add Program)
   d. изменение порядка выполнения задачи (Move Up / Move Down)

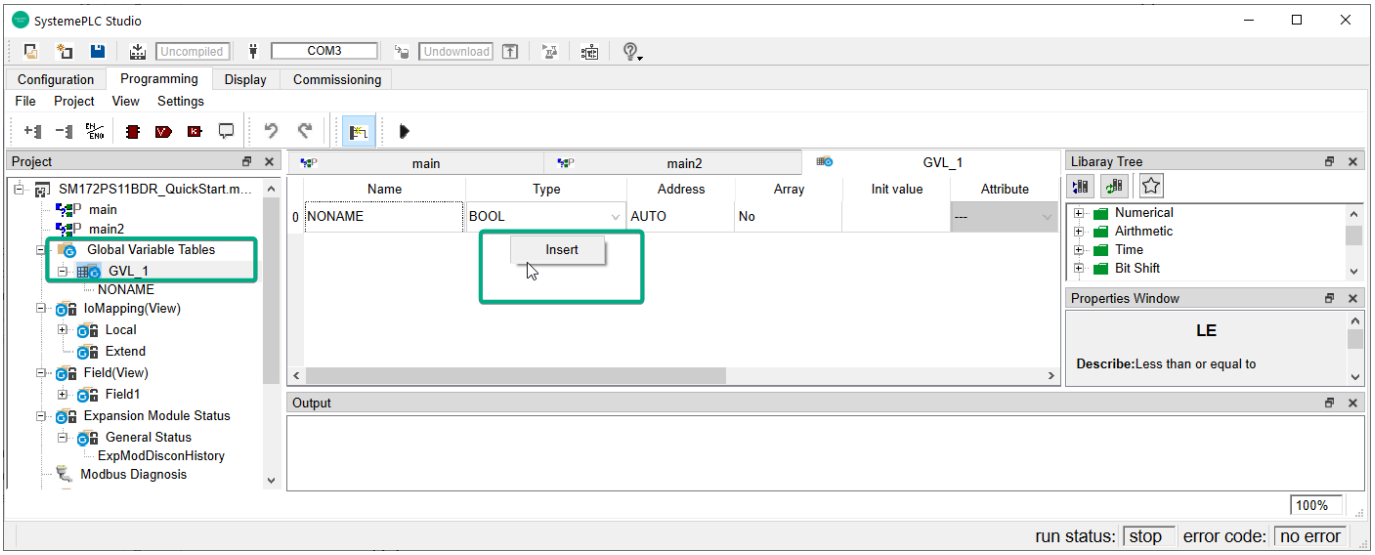


Написание программы

Создание Глобальные переменных (Global Variable Table)

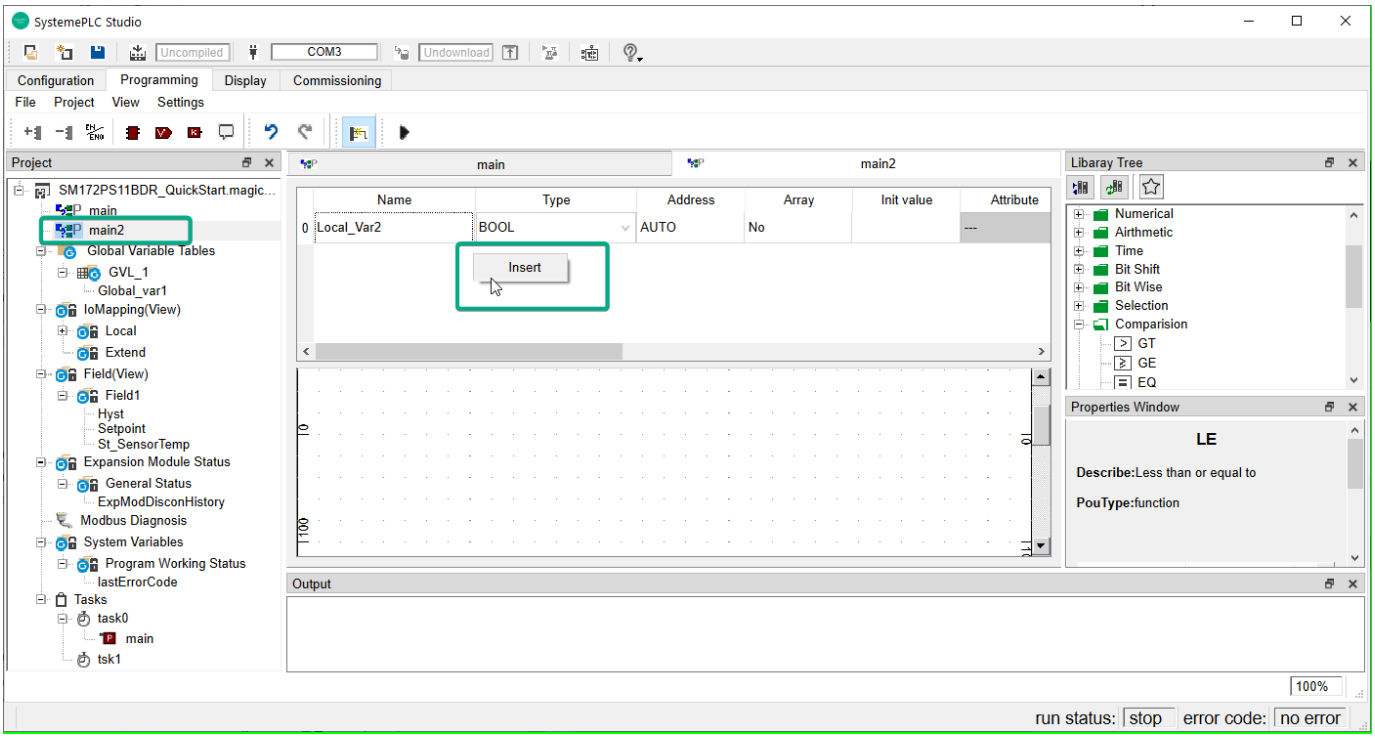
1. Создать лист глобальных переменных: ПКМ на папке Global Variables Table > Append
2. Открыть лист глобальных переменных: Двойной щелчок ЛКМ на папке глобальных переменных (например, GLV1)
3. Создать глобальную переменную: ПКМ в Области объявления переменных (Variable Declaration Area) > Insert

Далее откройте программу и перетаскивайте глобальную переменную ЛКМ из папки GLV1 в Область написания программного кода (Program editing area).



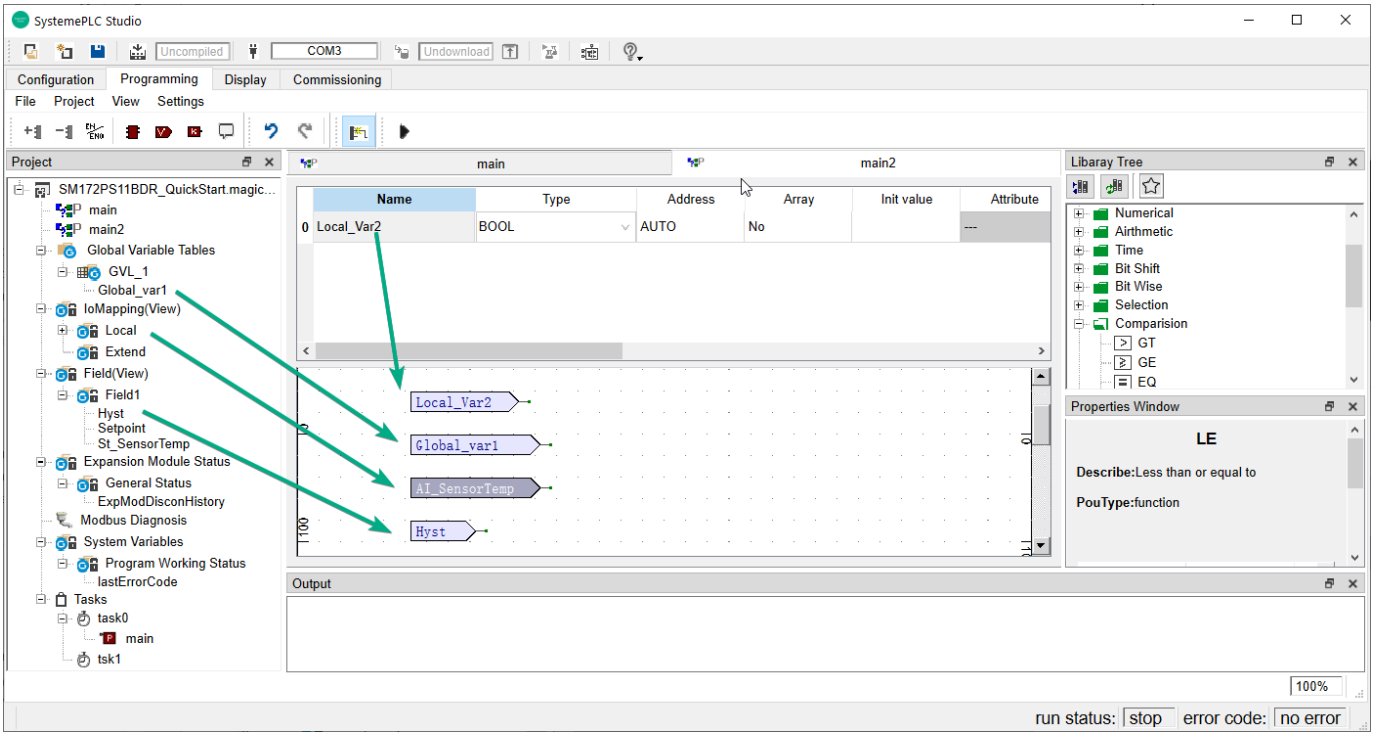
Создание локальные переменных

- Создать локальную переменную: находясь внутри программы (например, main2), ПКМ в Области объявления переменных (Variable Declaration Area) > Insert

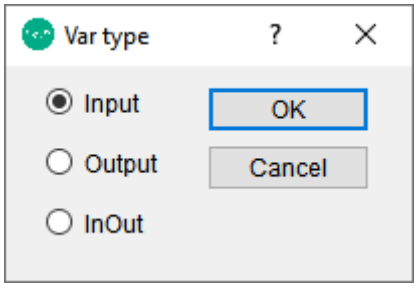


Добавление переменных в программу

- Перетаскивайте переменные из Дерева проекта (Project Tree) в Область написания программного кода (Program editing area)
- Перетаскивайте локальные переменные из Области объявления переменных (Variable Declaration Area) в Область написания программного кода (Program editing area).

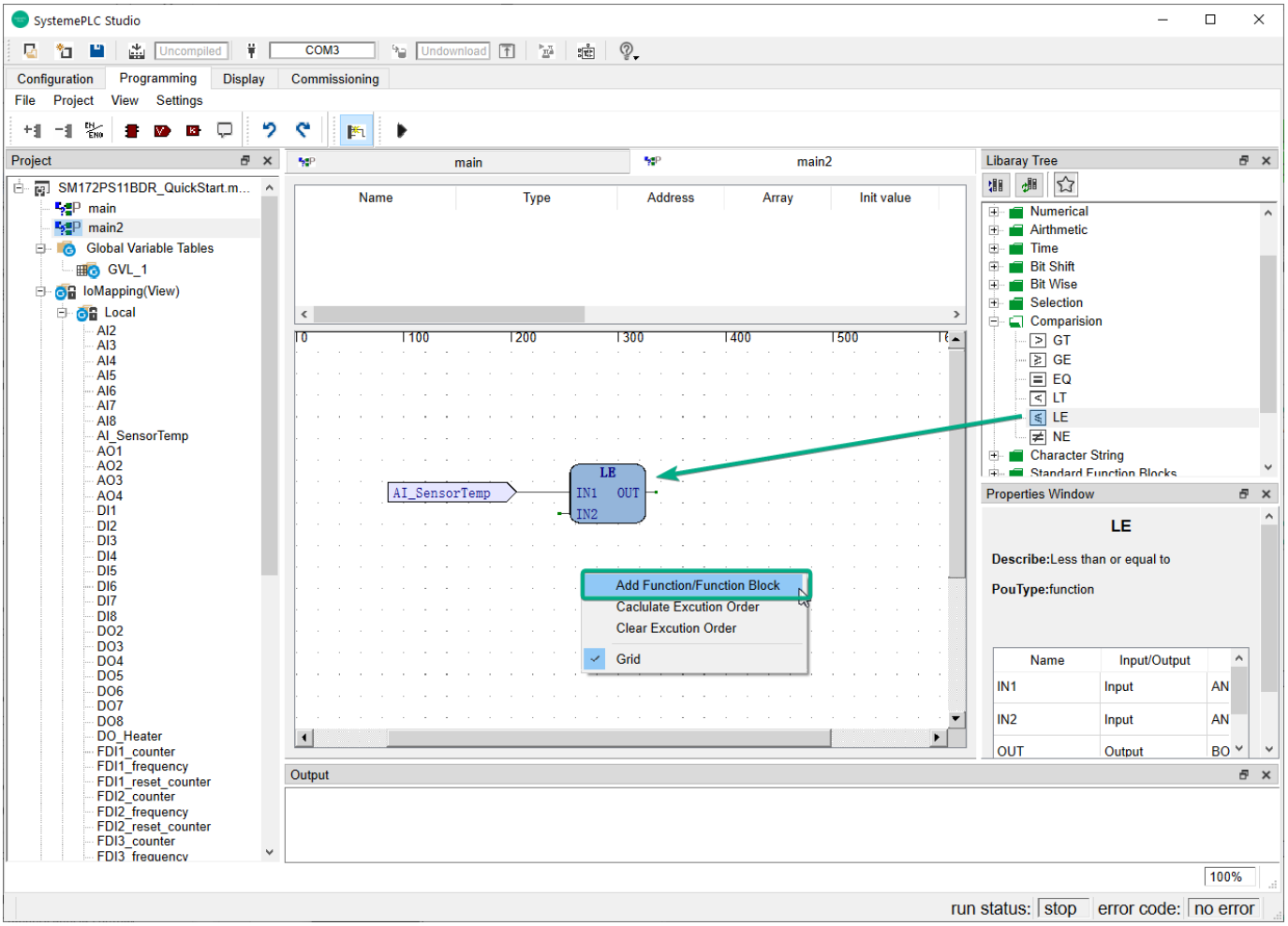


- При добавлении переменной укажите направление данных
  - Input: значение передается в переменную
  - Output: значение передается из переменной
  - Input and Output: значение передается в переменную и из нее



Добавление функциональных блоков

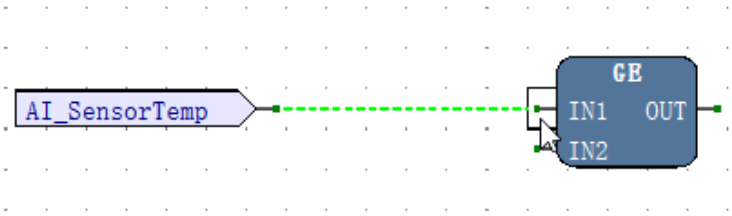
- Перетаскивайте блоки из Библиотека элементов (Library Tree) в Область написания программного кода (Program editing area)
- Вставляйте блоки через контекстное меню вставки
  - Нажмите ПКМ в области написания программного кода > Add Function/Function Block



Создание линии связи

Для создания линии связи

- Нажмите ЛКМ на зеленую точку входа или выхода элемента (переменной, функционального блока и др.)
- Протяните линию ко входу или выходу следующего элемента
- Снова нажмите ЛКМ на входе или выходе элемента

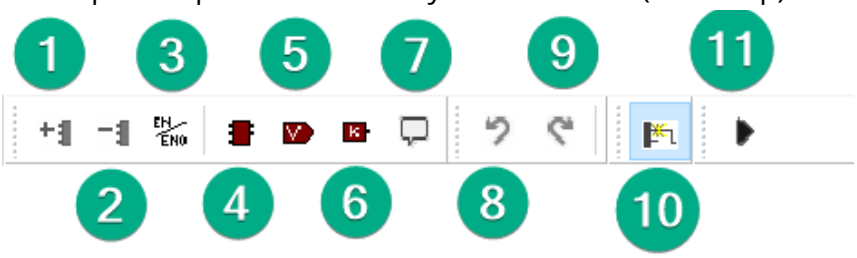


Панель инструментов Программирование (Programming Toolbar)

Добавляйте (функциональные блоки, переменные, константы, комментарии и др.) с помощью кнопок из **Панели инструментов программирования (Programming Toolbar)**

- **1:** увеличить количество входов функционального блока (Increase pins)
- **2:** уменьшить количество входов функционального блока (Decrease pins)
- **3:** включить/выключить пины EN/ENO функционального бока
  - Пин EN (Enable) активирует блок.

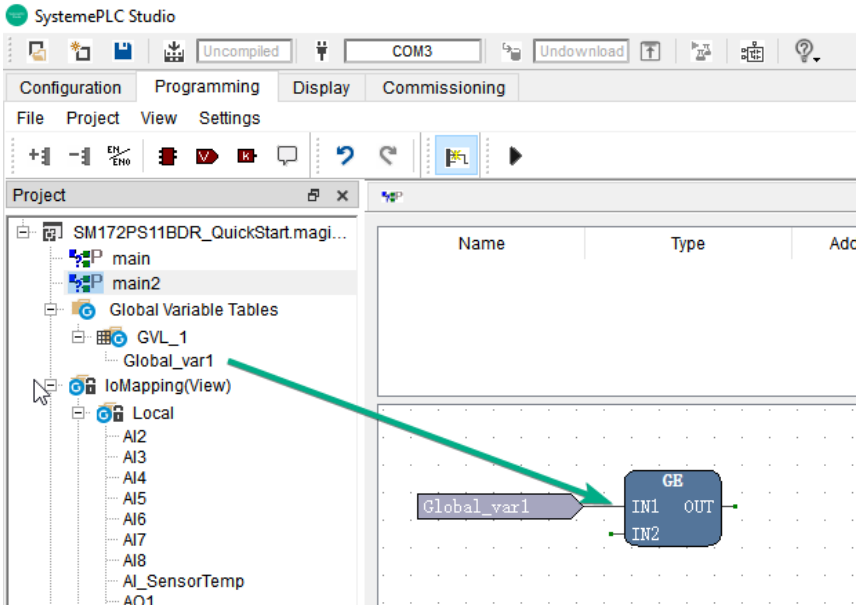
- Выход ENO (Enable Output) можно соединять с EN последующих блоков для создания цепочек выполнения
- Нажмите на одну из кнопок и далее ЛКМ в Область написания программного кода (Program editing area)
  - **4:** вставить функциональный блок (Block)
  - **5:** вставить переменную (Variable)
  - **6:** вставить константу (Constant)
  - **7:** вставить комментарий (Comment)
- **8:** отметить действие (Undo) [Ctrl+Z]
- **9:** повторить действие (Undo) [Ctrl+Y]
- **10:** включить/выключить функцию авто-соединение (Auto Connect).
  - Если включено, то переменная вблизи входов/выходов функционального блока сразу подключится к нему, иначе останется неподключенной.
- **11:** переключить режим работы ПЛК Запуск/Остановка (Run/Stop)



Советы по программированию

Авто-соединение переменной с функциональным блоком при перетаскивании ее из дерева проекта

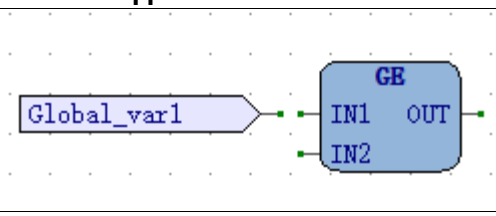
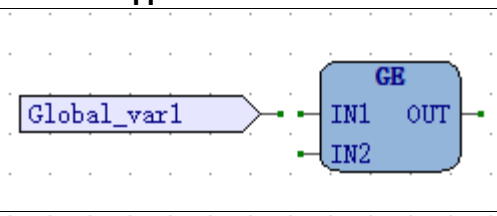
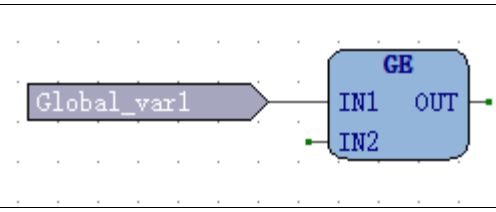
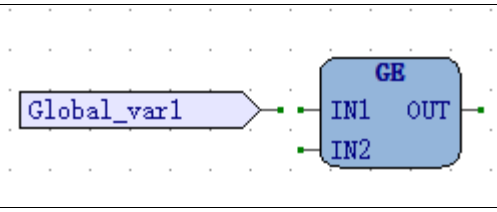
- Перетаскивайте переменные из дерева проекта в область написания программного кода **на зеленую точку входа или выхода** функционального блока. Переменная будет автоматически подсоединена к блоку.



Авто-соединение (Auto Connect) переменной с функциональным блоком в области написания программного кода

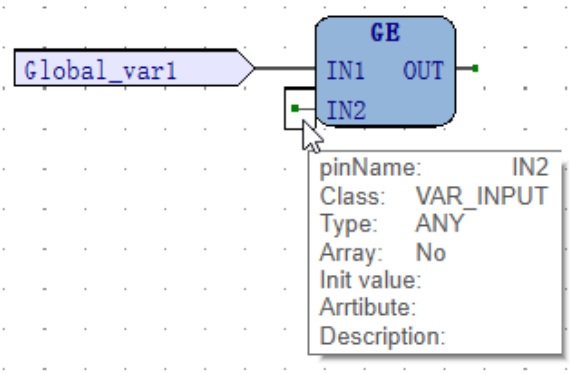
- Включите функцию авто-соединение (Auto Connect) в Панели инструментов программирования (Programming Toolbar) [включено по умолчанию]

  - В области написания программного кода перетаскивайте переменные вплотную ко входу/выходу функционального блока (Шаг1). Тогда контакты переменной автоматически будут подключены к функциональному блоку (Шаг2).
  - Без включения авто-соединения переменная останется неподключенной (Шаг1 и Шаг2). Линию связи необходимо будет создать вручную.

Шаг	Авто-соединение включено	Авто-соединение выключено
1		
2		

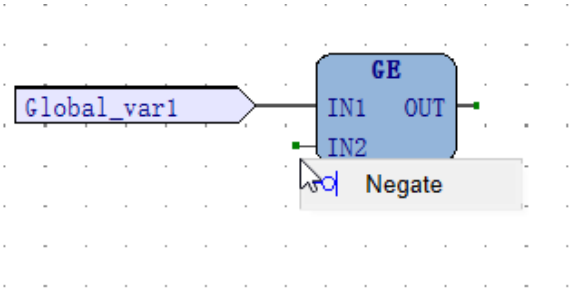
Информация о контактах

- При наведении курсора мыши на контакт (pin) функционального блока появляется всплывающая подсказка с подробной информацией о нем.



Инверсия контакта функционального блока

- Наведите курсор мыши на контакт (pin) функционального блока
- Нажмите ПКМ > Negate

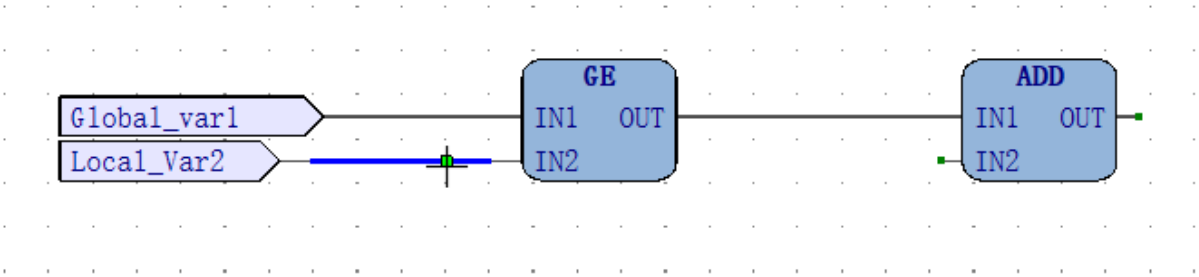


Создание ветвлений (Branchers)

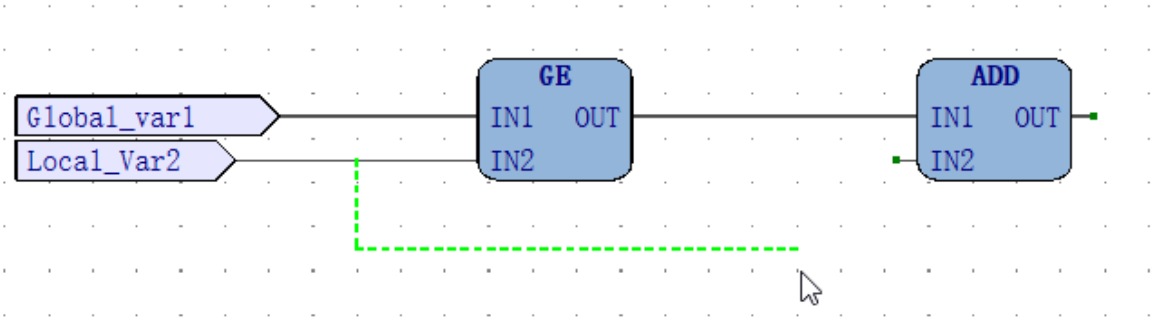
Точки ветвления можно создавать непосредственно на линии соединения переменных и/или блоков.

Для этого

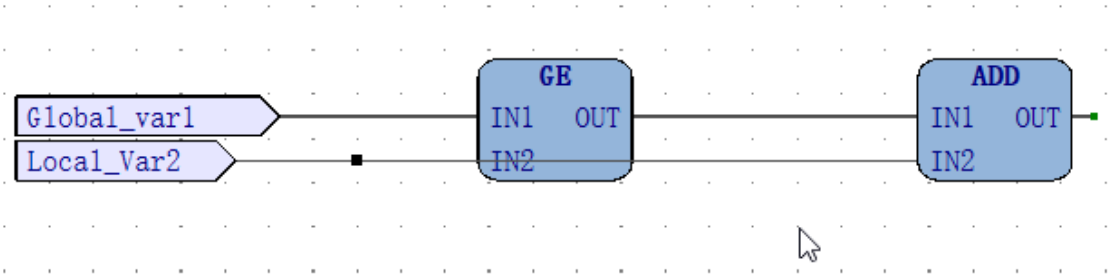
- Наведите курсор на линию, пока не появится зеленая точка



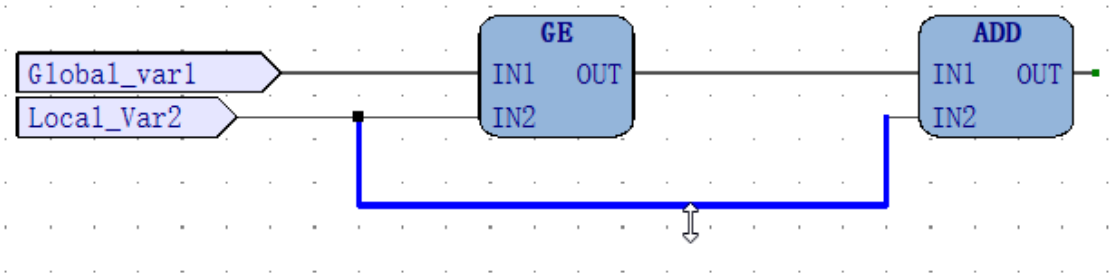
- Нажмите на ней ЛКМ и перетащите в нужное место для создания нового соединения



- Подключите ее к другому входу/выходу функционального блока



- Двигайте линию соединения с помощью ЛКМ для лучшей визуализации



Демонстрационный проект Шаг 3

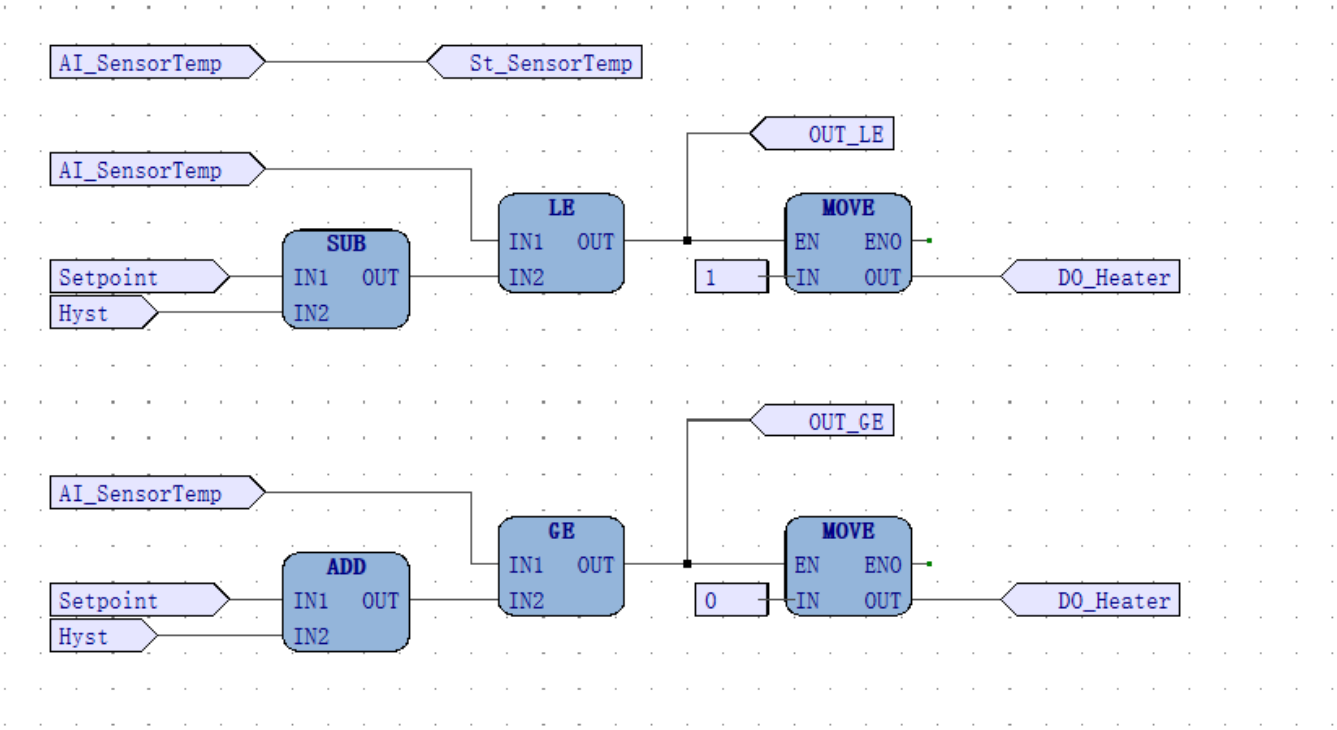
Выполните последовательность действий

1. Откройте программу на вкладке Программирование (Programming) > main
2. Создайте локальные переменные в программе main
   a. Name: OUT_LE; Type: BOOL;
   b. Name: OUT_GE; Type: BOOL;

main

	Name	Type	Address	Array	Init value	Attribute
0	OUT_LE	BOOL	AUTO	No		---
1	OUT_GE	BOOL	AUTO	No		---

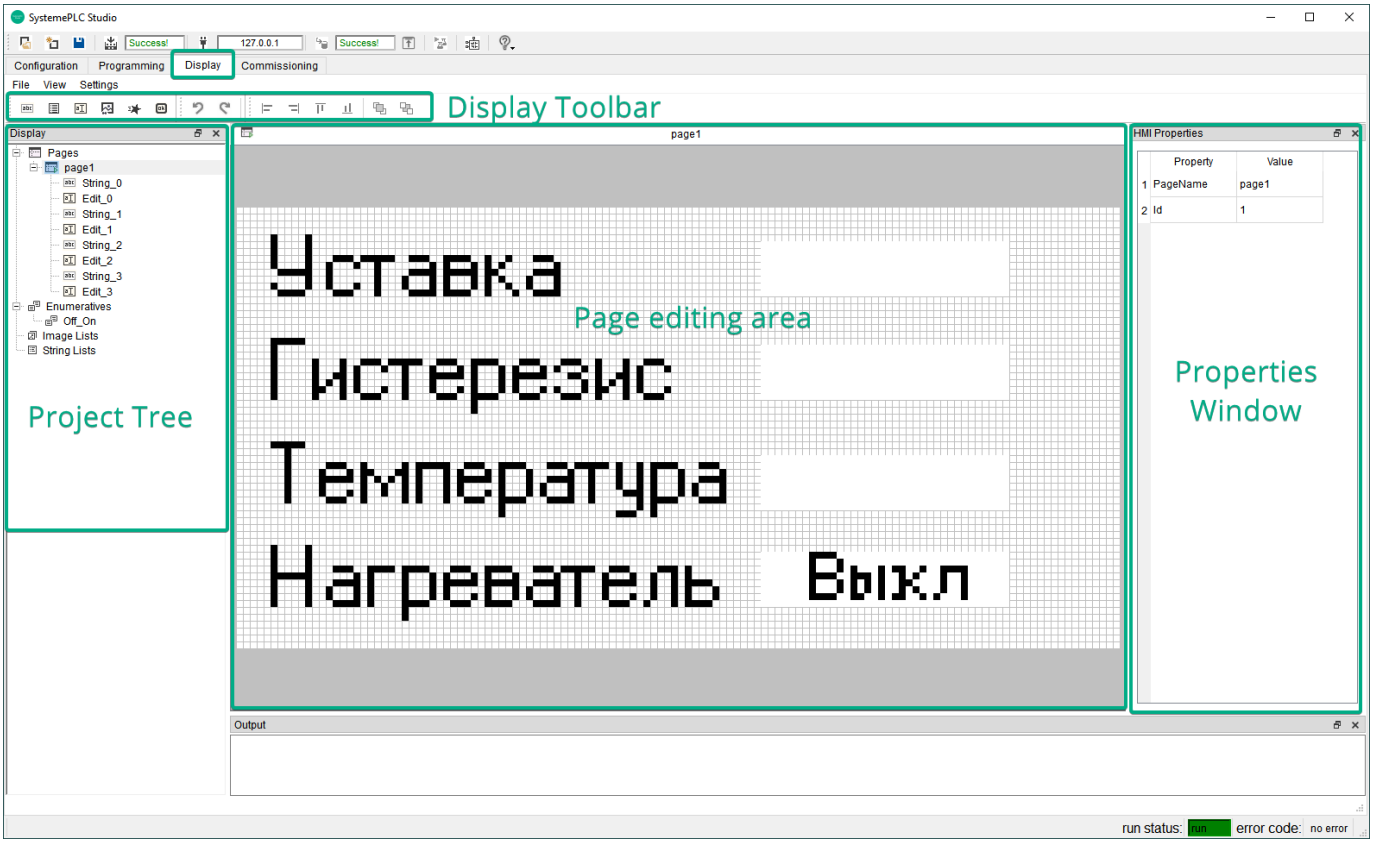
3. Создайте блок схему как показано на рисунке в программе main
   a. Перетащите переменные из Дерева проекта (Project Tree) в Область написания программного кода (Program editing area).
   b. Перетащите локальные переменные из Области объявления переменных (Variable Declaration Area) в Область написания программного кода (Program editing area).
   c. Перетащите функциональные блоки из Библиотека элементов (Library Tree)
      (1) Arithmetic > SUB; ADD; MOVE;
      (2) Comparison > LE; GE;
   d. Включите pin EN/ENO на блоке MOVE для условного присваивания значений (ПКМ на блоке > Enable EN/ENO pins)
   e. Перетащите две константы (1 и 0) на входы (IN) блоков MOVE
   f. Соедините все элементы линиями соединения как показано на рисунке



Программирование дисплея ПЛК

Интерфейс вкладки Дисплей (Display)

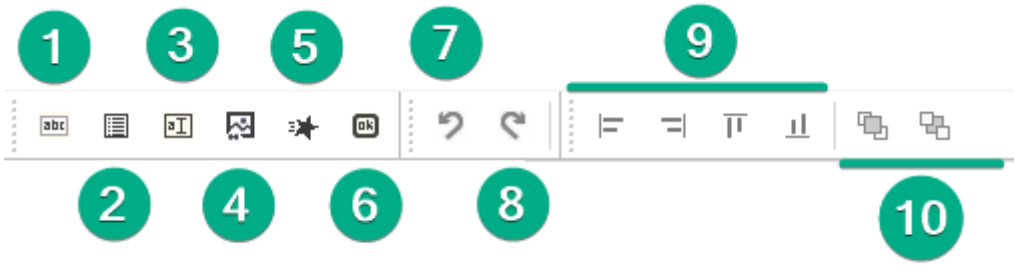
- **Панель инструментов Дисплей (Display Toolbar):** обеспечивает быстрый доступ к часто используемым элементам для создания пользовательских экранов (текст, поле ввода/вывода и др.) и функциям их компоновки.
- **Дерево проекта (Project Tree):** отображает иерархическую структуру пользовательских экранов и элементов на них.
- **Область создания пользовательских экранов (Page editing area)** служит для расположения элементов на пользовательских экранах.
- **Свойства (Properties Window):** позволяет задавать значение свойств выбранного элемента на пользовательском экране.



Панель инструментов Дисплей (Display Toolbar)

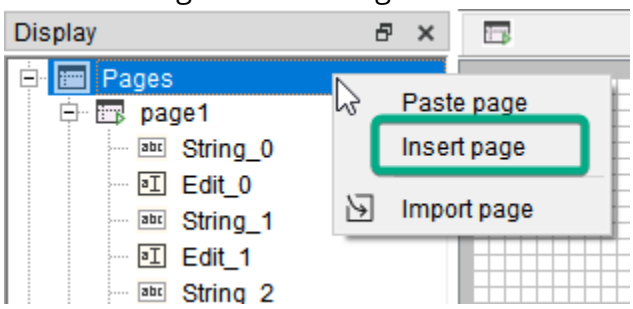
Добавляйте элементы на пользовательский дисплей с помощью кнопок из панели инструментов Дисплей (Display Toolbar)

- **1:** вставить статический текст (Insert Static)
- **2:** вставить список строк (Insert Sting List)
- **3:** вставить поле ввода/вывода (Insert Edit)
- **4:** вставить рисунок (Insert Image)
- **5:** вставить анимацию (Insert Animation)
- **6:** вставить кнопку (Insert Button)
- **7:** отметить действие (Undo) [Ctrl+Z]
- **8:** повторить действие (Undo) [Ctrl+Y]
- **9:** функции выравнивания нескольких элементов (по левому\правому краю и по верху\низу элемента) [Align Left\Right; Align Top\Bottom]
- **10:** функции показа элемента поверх или под другими элементами (Bring to Front \ Send to back)

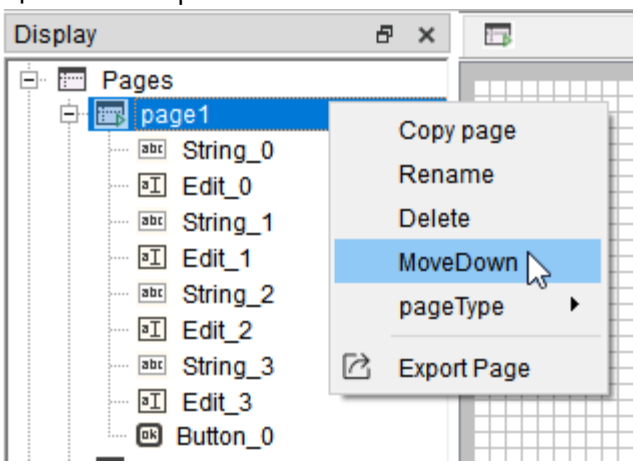


Создание пользовательского экрана ПЛК

В дереве проекта нажмите ПКМ на Pages > Insert Page



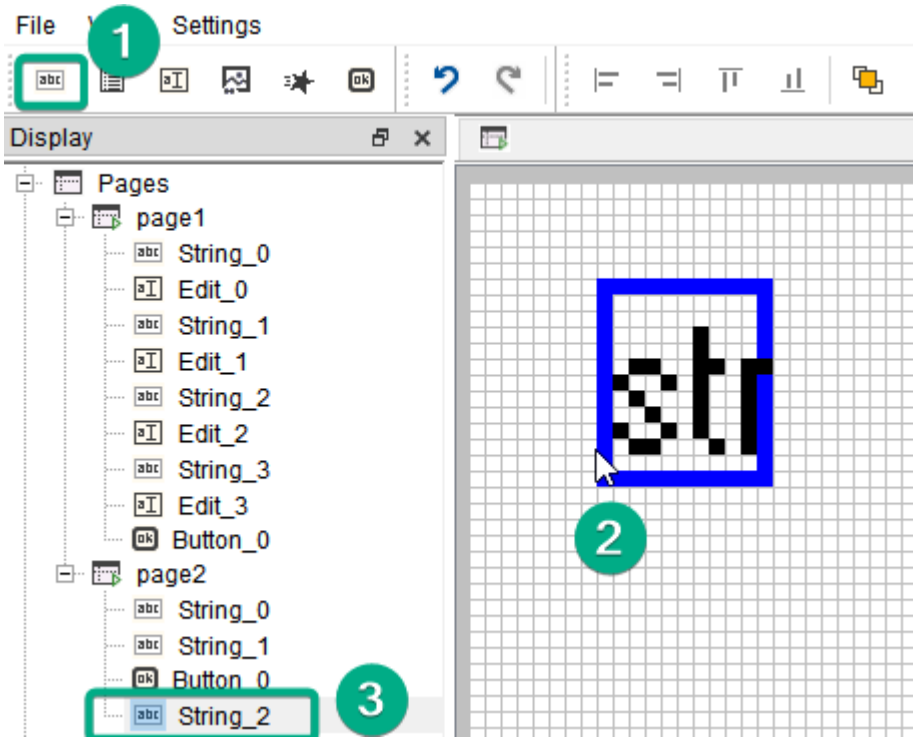
Изменяйте порядок пользовательских экранов в дереве проекта кнопками: ПКМ на пользовательской странице > Move Up/Move Down



Создание пользовательского элемента

Для создание нового элемента на пользовательском экране

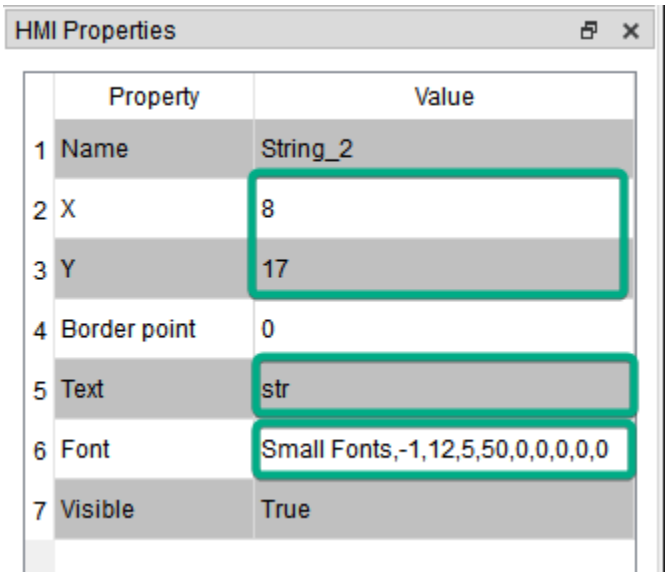
1. Нажмите ЛКМ на нужный элемент в панели инструментов Дисплей (Display Toolbar)
2. Нажмите ЛКМ в области создания пользовательских экранов (Page editing area). Место нажатия ЛКМ определит левый нижний угол элемента.
3. Элемент отобразится на пользовательском экране и появится в дереве проекта



Основные элементы пользовательских станиц

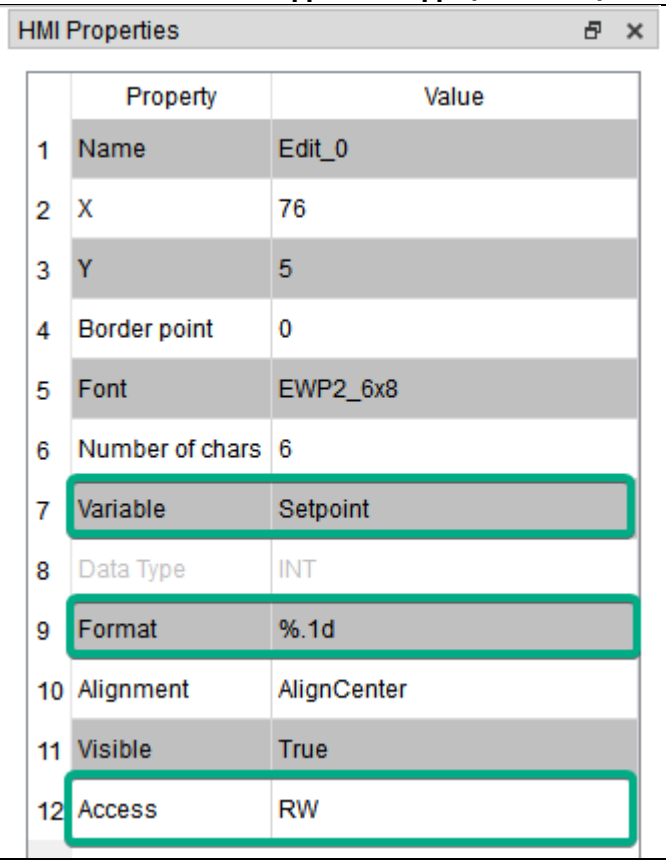
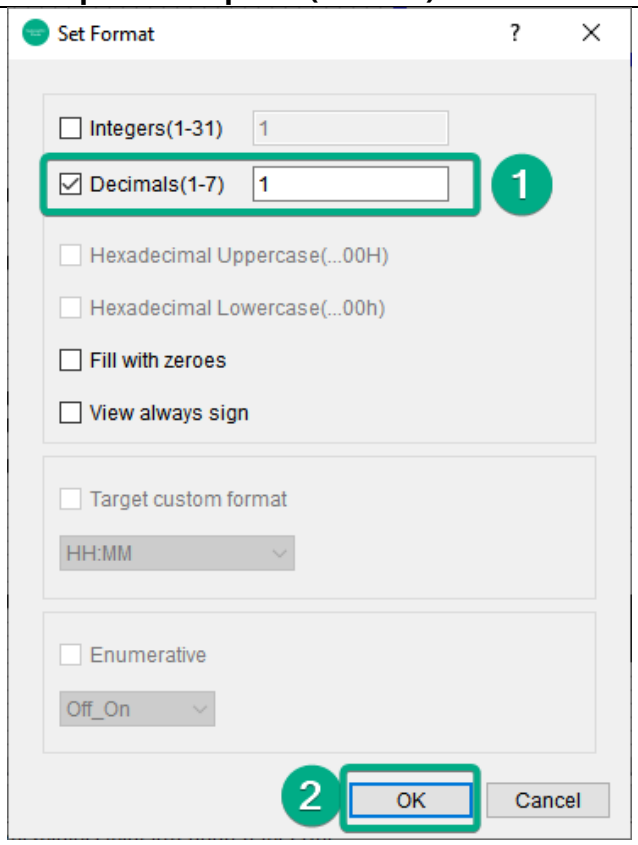
Статический текст (Static Text)

- В поле Text в окне свойств можно вводить любой текст на русском или английском языке
- Доступно изменение шрифта, его размера, начертания (жирный, подчеркнутый) и выравнивания
- Элемент можно перемещать мышью или точно позиционировать, задавая координаты левого верхнего угла



Поле ввода/вывода (Edit Box)

- В свойстве Variable выбирается переменная из проекта (доступны все глобальные переменные, включая входы и выходы). Например, Setpoint.
- Ключевая настройка Формат (**Format**)
  - Integer: отображение целого числа. Можно задать количество цифр.
  - Decimal: отображение числа с плавающей точкой. Указывается количество знаков после запятой. Например, значение переменной 230 при формате Decimal и 1 знаке после запятой будет отображено как 23.0.
- Настройка **Доступ (Access)**
  - Read/Write (RW): позволяет и читать значение переменной с экрана, и записывать его в контроллер.
  - Read Only (RO): разрешает только чтение значения (режим мониторинга).

Свойства Поля ввода/вывода (Edit Box)	**Настройка Формат (Format)**
	

Кнопка (Button)

- В свойстве Text/img задается отображаемый текст на кнопке или рисунок
- Основная настройка **Действие (Action)**
  - Open Page: открывает другую страницу.
    - В Action Parameter выбирается целевая страница.
  - Invert Bool: инвертирует значение булевой переменной.
    - Каждое нажатие меняет TRUE на FALSE и наоборот.
    - В Action Parameter выбирается булева переменная.
  - Close: закрывает текущую страницу. Используется для возврата в предыдущее меню, так как страницы открываются "слоями".
  - None: кнопка не выполняет действий.

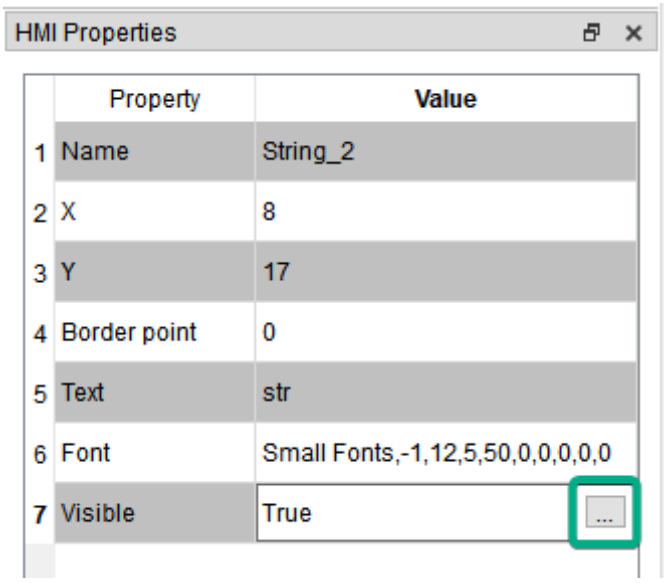
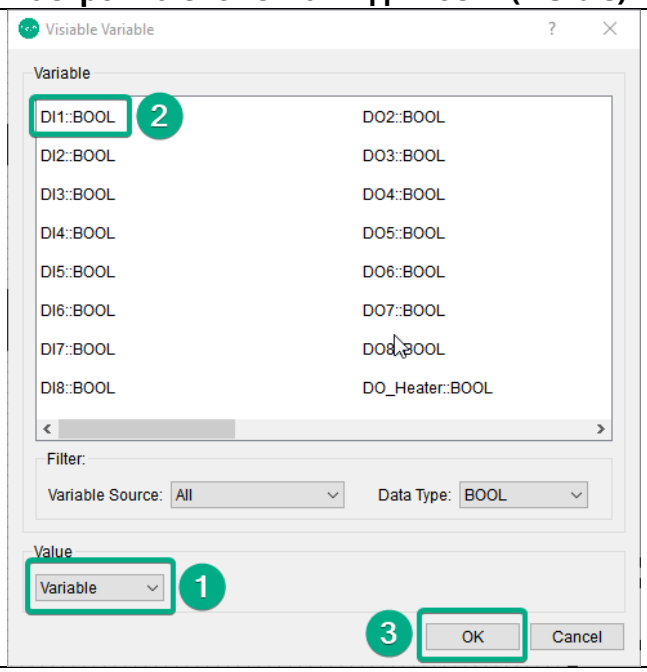
HMI Properties

	Property	Value
1	Name	Button_0
2	X	117
3	Y	53
4	Border point	1
5	Width	10
6	Height	10
7	Text/img	>
8	Font	EWP2_6x8
9	Visible	True
10	Action	OpenPage
11	Alignment	AlignCenter
12	Action par	page2

Расширенные функции

Управление видимостью элемента

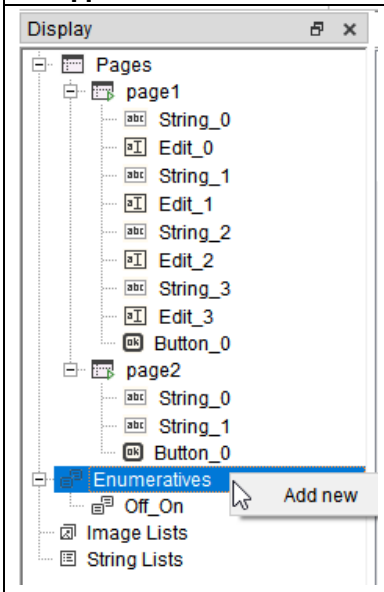
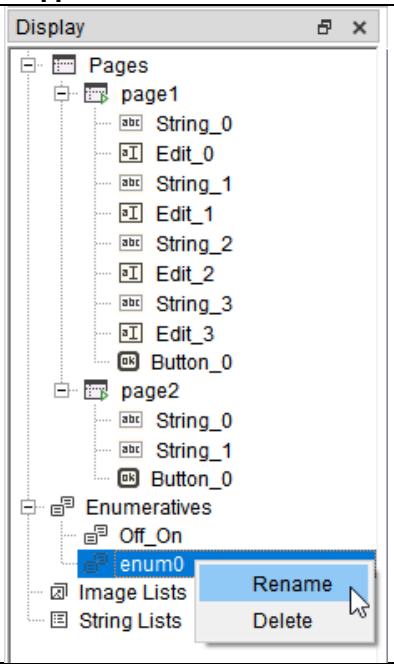
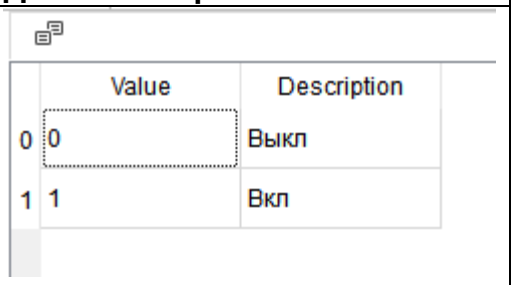
- С помощью изменения свойства элемента **Видимость (Visible)** [по умолчанию равно TRUE], элемент можно сделать видимым условно
- Для этого в свойстве элемента Visible выбирается значение Value: Переменная (Variable). Например, дискретный вход DI1 или выход DO_Heater.
- Элемент будет отображаться на экране только тогда, когда значение этой переменной равно TRUE

Свойство элемента Видимость (Visible)	Настройка свойства Видимость (Visible)
	

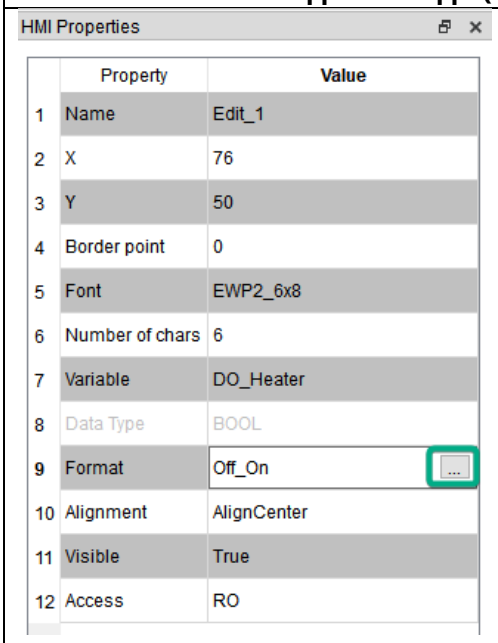
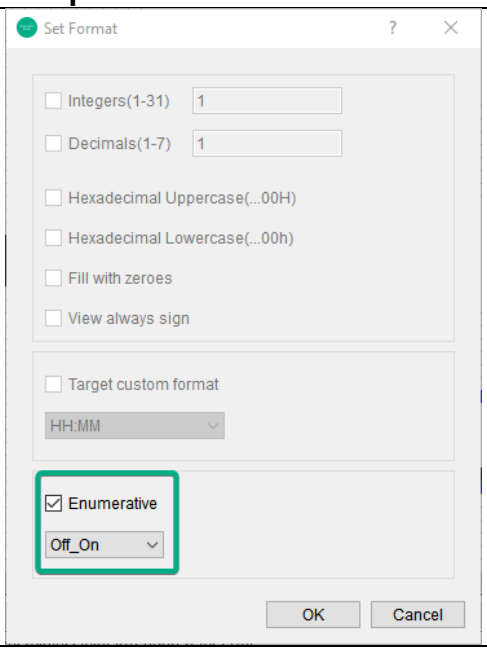
Работа с перечислениями (Enumerative)

Перечисление (Enumerative) позволяет заменять числовые значения переменных в поле ввода/вывода Edit на текстовые статусы

- Для этого необходимо создать новый Enumerative через дерево проекта: ПКМ на Enumeratives > Add new
- Новому Enumerative задается имя. ПКМ на созданном Enumerative > Rename
- Внутри Enumerative добавляются строки, где числовому значению ставится в соответствие текст (например, 0 > "Выкл", 1 > "Вкл"). Для этого
  - Двойной щелчок ЛКМ на новом Enumerative
  - В открывшемся окне с именем Enumerative в области создания пользовательских экранов (Page editing area), нажать ПКМ > Append (добавить строки)

Создать Enumerative	Задать имя Enumerative	Добавить строки в Enumerative
		

- Далее после создания Enumerative, в свойстве поля ввода/вывода (Edit Box): формат (Format) выбирается имя созданного Enumerative (вместо Integer или Decimal).
- В результате, вместо цифр 0 или 1 в окне поля ввода/вывода (Edit Box) будет отображаться "Выкл" или "Вкл".

Свойства поля ввода/вывода (Edit Box)	Выбрать имя Enumerative
	

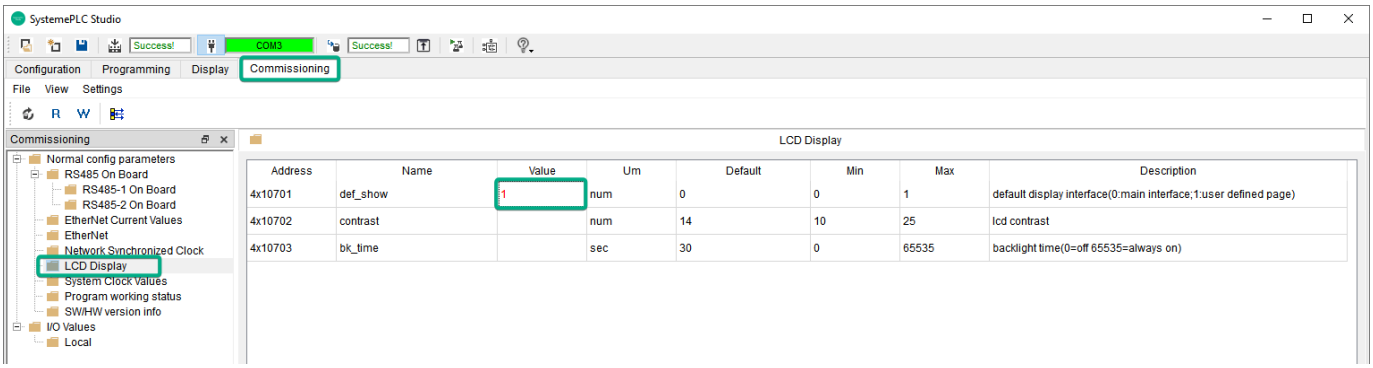
Создание многостраничного интерфейса

Для создания многостраничного интерфейса

- Создайте дополнительные страницы (Page 2, Page 3 и т.д.).
- Навигация между ними организуется с помощью кнопок с действием Open Page.
- Для возврата на предыдущий экран используются кнопки с действием Close.

Назначение стартовой страницы

- Во вкладке **Commissioning** в разделе **LCD Display** задать значение переменной **def_show**. Это влияет на то, какой экран будет показан при старте контроллера.
  - 0: Системное меню контроллера.
  - 1 и далее: Пользовательская страница. Порядковый номер страницы определяется ее позицией в списке (регулируется кнопками Move Up/Move Down в дереве проекта).

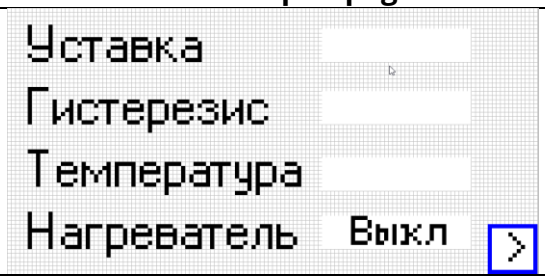
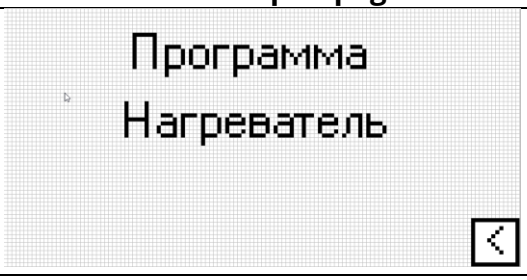


Демонстрационный проект Шаг 4

Создание пользовательских экранов для дисплея ПЛК

Выполните последовательность действий.

Создайте два пользовательских экрана: page1 (уже создан по умолчанию) и page2.
Итоговый вид экранов после добавление всех элементов.

Пользовательский экран page1	**Пользовательский экран page2**
	

Добавьте на пользовательские экраны следующие элементы

1. **Статический текст (Static Text)** «Уставка», «Гистерезис», «Температура», «Нагреватель» на page1 и «Программа», «Нагреватель» на Page2
   a. Задайте свойство текст (Text) каждого элемента соотвественно
2. **Поля ввода/вывода (Edit Box)** для переменных

a. SetPoint. Настроен на чтение и запись (Access: Read/Write). Формат отображения – Decimal с одной цифрой после запятой (чтобы 230 отображалось как 23.0).
b. Hyst. Настроен на чтение и запись (Access: Read/Write). Формат отображения – Decimal с одной цифрой после запятой.
c. St_SensorTemp. Настроен на чтение (Access: Read Only). Формат отображения – Decimal с одной цифрой после запятой.
d. DO_Heater. Настроен на чтение (Access: Read Only). Для него создан и применен Enumerative, который преобразует числовые значения 0 и 1 в текстовые статусы "Выкл" и "Вкл".

Свойства SetPoint

HMI Properties

	Property	Value
1	Name	Edit_0
2	X	76
3	Y	5
4	Border point	0
5	Font	EWP2_6x8
6	Number of chars	6
7	Variable	Setpoint
8	Data Type	INT
9	Format	%.1d
10	Alignment	AlignCenter
11	Visible	True
12	Access	RW

Свойства Hyst

HMI Properties

	Property	Value
1	Name	Edit_3
2	X	76
3	Y	20
4	Border point	0
5	Font	EWP2_6x8
6	Number of chars	6
7	Variable	Hyst
8	Data Type	INT
9	Format	%.1d
10	Alignment	AlignCenter
11	Visible	True
12	Access	RW

Св-ва St_SensorTemp

HMI Properties

	Property	Value
1	Name	Edit_2
2	X	76
3	Y	36
4	Border point	0
5	Font	EWP2_6x8
6	Number of chars	6
7	Variable	St_SensorTemp
8	Data Type	INT
9	Format	%.1d
10	Alignment	AlignCenter
11	Visible	True
12	Access	RO

Свойства DO_Heater

HMI Properties

	Property	Value
1	Name	Edit_1
2	X	76
3	Y	50
4	Border point	0
5	Font	EWP2_6x8
6	Number of chars	6
7	Variable	DO_Heater
8	Data Type	BOOL
9	Format	Off_On
10	Alignment	AlignCenter
11	Visible	True
12	Access	RO

3. **Кнопка (Button) для переключения между экранами page1 и page2**. Добавьте элемент кнопка на page1 и на page2 соответственно
   a. В свойства каждой кнопки (Button) задайте
      (1) ширину (Width) = 10; высоту (Height) = 10.
      (2) Текст (Text/img) задайте в виде символа «>» для page1 и «<» для page2
      (3) На странице page1 для кнопки задайте Action – OpenPage, а в Action par – Page2
      (4) На странице page2 для кнопки задайте Action – OpenPage, а в Action par – Page1

Свойства Button_0 на page1

HMI Properties

	Property	Value
1	Name	Button_0
2	X	117
3	Y	53
4	Border point	1
5	Width	10
6	Height	10
7	Text/img	>
8	Font	EWP2_6x8
9	Visible	True
10	Action	OpenPage
11	Alignment	AlignCenter
12	Action par	page2

Свойства Button_0 на page2

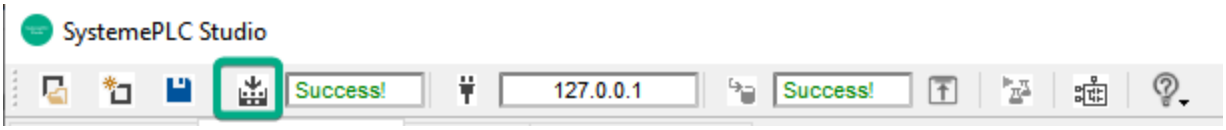
HMI Properties

	Property	Value
1	Name	Button_0
2	X	117
3	Y	53
4	Border point	1
5	Width	10
6	Height	10
7	Text/img	<
8	Font	EWP2_6x8
9	Visible	True
10	Action	OpenPage
11	Alignment	AlignCenter
12	Action par	page1

Компиляция программы

Выполните последовательность действий

1. **Сохранение проекта:** сохраните текущий проект (Save Project)
   a. Сохранение проекта перед компиляцией опционально и не обязательно
2. **Компиляция:** выполните компиляцию проекта (Compile Project).
   a. Компиляция (Compile) проверяет проект на ошибки
3. **Окно вывода (Output Window)** отобразит результаты. Отсутствие ошибок указывает на успешную компиляцию.



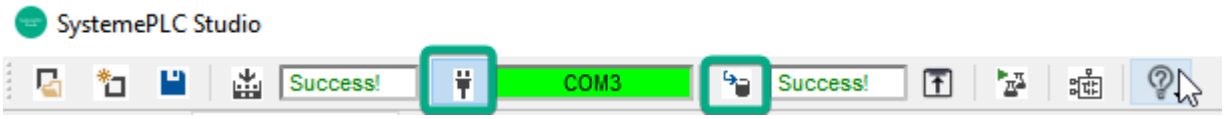
Загрузка программы в ПЛК*

**Пропустите этот шаг если у вас нет физического ПЛК.*

Выполните последовательность действий

1. **Подключение:** установите соединение с ПЛК (см. [Подключение ПЛК к ПК через USB Type-C]).
2. **Загрузка:** нажмите значок **загрузки (Download code)** для передачи программы в ПЛК.

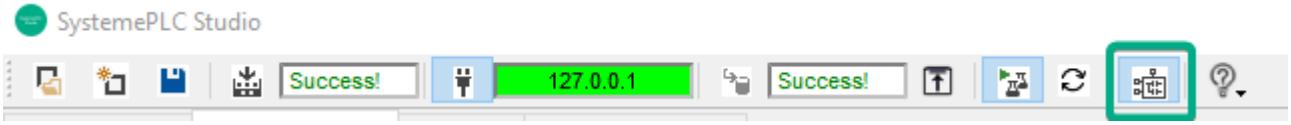
Примечание: загрузка (Download) программы в ПЛК или оффлайн симулятор автоматически запускает компиляцию



Загрузка программы в оффлайн симулятор ПЛК

Выполните последовательность действий

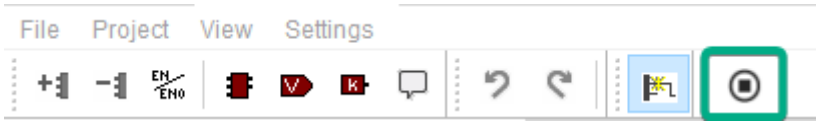
1. **Запуск оффлайн симулятора:** нажмите кнопку Simulation Mode (Офф-лайн симулятор ПЛК)
2. Программа будет автоматически скомпилирована
3. Программа будет автоматически загружена в симулятор ПЛК



Запуск / Стоп программы ПЛК (Run/Stop)

Запуск программы:

- Для запуска / остановки ПЛК (Run/Stop) нажмите переключатель режима работы ПЛК (PLC Run/Stop)
- Если ПЛК находится в **режиме автозапуска**, программа выполняется сразу после успешной загрузки.
- Если ПЛК находится в **режиме останова**, необходимо перевести его в режим автозапуска через экранное меню на дисплее контроллера.



Оффлайн симулятор ПЛК (Simulation Mode)

Офф-лайн симулятор ПЛК (Simulation Mode) эмулирует работу реального ПЛК, запуская виртуальный контроллер.

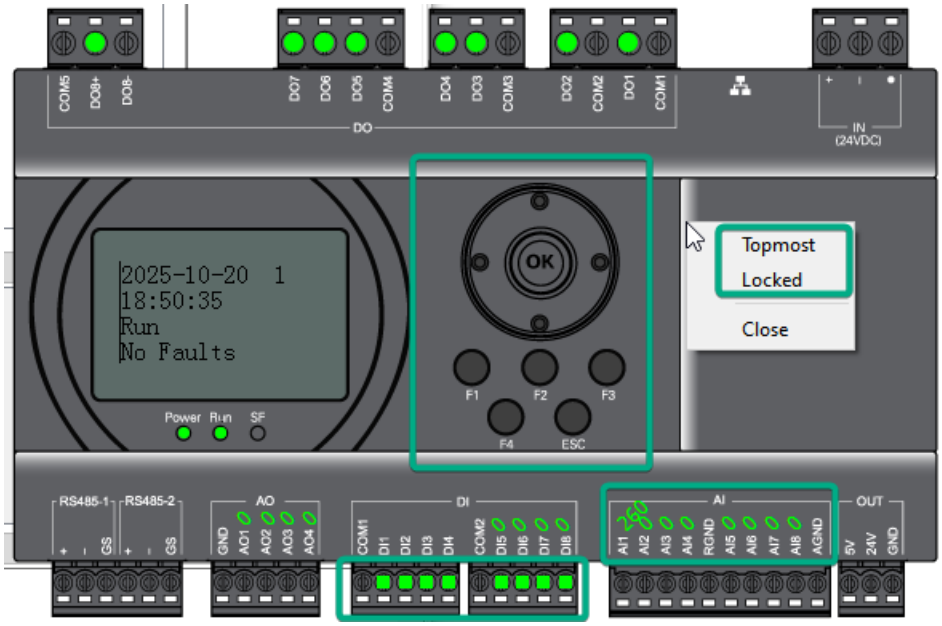
С помощью офф-лайн симулятора ПЛК можно

- Задать значения дискретных и аналоговых входов
- В реальном времени отслеживать выполнение логики программы ПЛК
- Визуально наблюдать значения на дискретных и аналоговых выходах ПЛК
- Создать, протестировать и отладить программу без подключения к физическому ПЛК

Примечание: в текущей версии симулятор ПЛК поддерживает только основной контроллер. Модули расширения симуляцию не поддерживают, поэтому в результатах симуляции отображаются выходные сигналы только основного ПЛК.

Работа с оффлайн симулятором ПЛК

- Двойной щелчок ЛКМ по DI позволит переключить дискретный вход: 0 > 1 или 1 > 0
  - Красный цвет: включен (1)
  - Зеленый цвет: выключен (0)
- Двойной щелчок ЛКМ по AI – откроет окно задания значений аналоговых входов
- ЛКМ на кнопках OK, стрелки, F1,...,F4 – работа с экраном ПЛК
- Значения DO и AO выводятся по результату работы логики программы. Задавать их из оффлайн симулятора нельзя.
- Нажмите ПКМ > Topmost для закрепление офф-лайн симулятор ПЛК поверх других окон
- Нажмите ПКМ > Locked для закрепления позиции офф-лайн симулятор ПЛК на экране

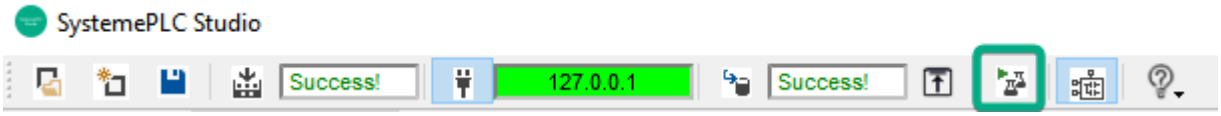


Онлайн мониторинг ПЛК (Live Debug Mode)

Онлайн мониторинг ПЛК (Live Debug Mode) позволяет

- наблюдать за выполнением программы в реальном времени
- отслеживать состояния переменных непосредственно в области написания программного кода (Program editing area)
- следить за логикой выполнения программы

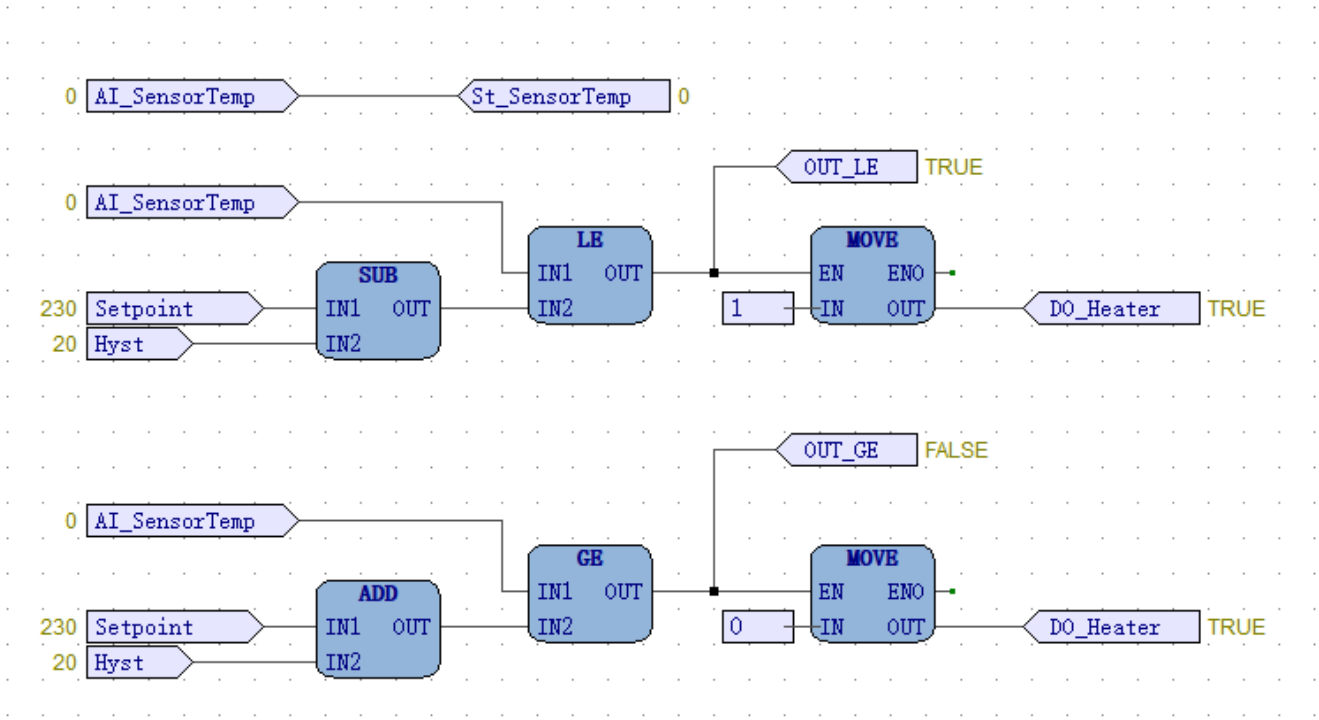
Для включения онлайн-мониторинга ПЛК нажмите значок Live Debug Mode на панели инструментов программирования (Programming Toolbar).



Важно!

- Онлайн мониторинг ПЛК (Live Debug Mode) может использоваться как с подключенным реальным ПЛК, так и совместно с офф-лайн симулятором ПЛК.
- Для внесения изменений в программу необходимо отключить онлайн мониторинг ПЛК

Примечание: для отслеживания промежуточных значений в программе создавайте локальные переменные и соединяйте их с выходами блоков (например: OUT_LE, OUT_GE).

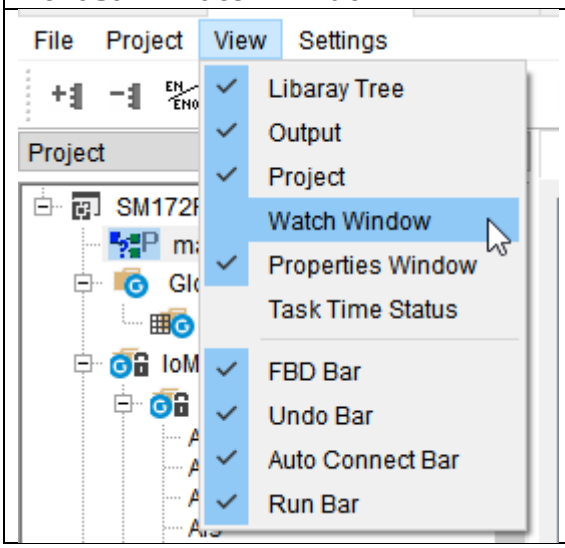
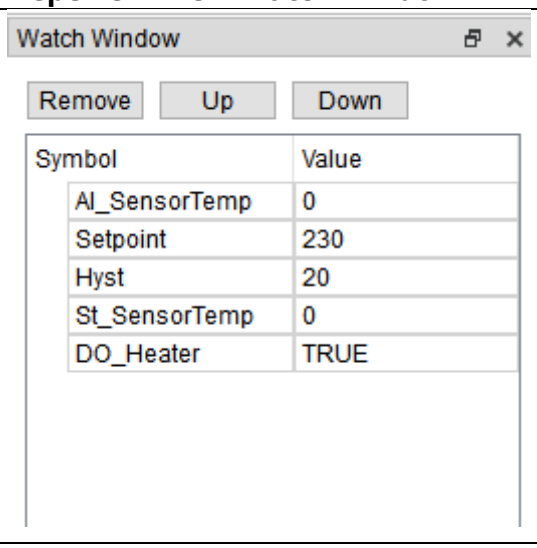


Окно наблюдения за переменными (Watch Window)

Окно наблюдения за переменными (Watch Window) позволяет наблюдать и изменять значения переменных в реальном времени.

Для использования Watch Window

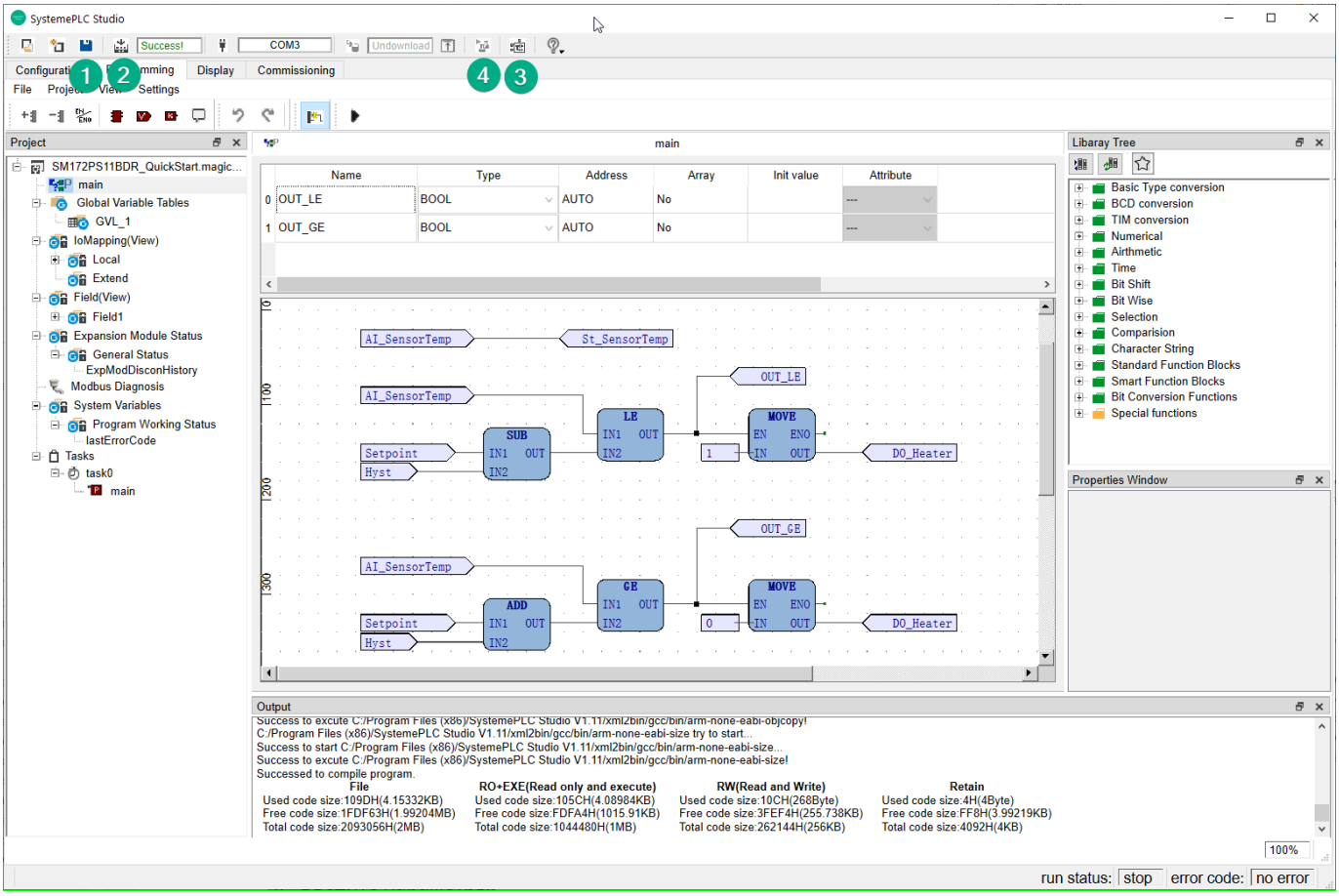
- Нажмите пункт меню View > Watch Window
- Перетащите переменные из дерева проекта (Project Tree) в Watch Window
- Изменяйте значения переменных в реальном времени прямо в окне Watch Window и наблюдайте за логикой исполнения программы (например, Setpoint и Hyst).

Показать Watch Window	**Переменные в Watch Window**
	

Демонстрационный проект Шаг 5

Выполните последовательность действий

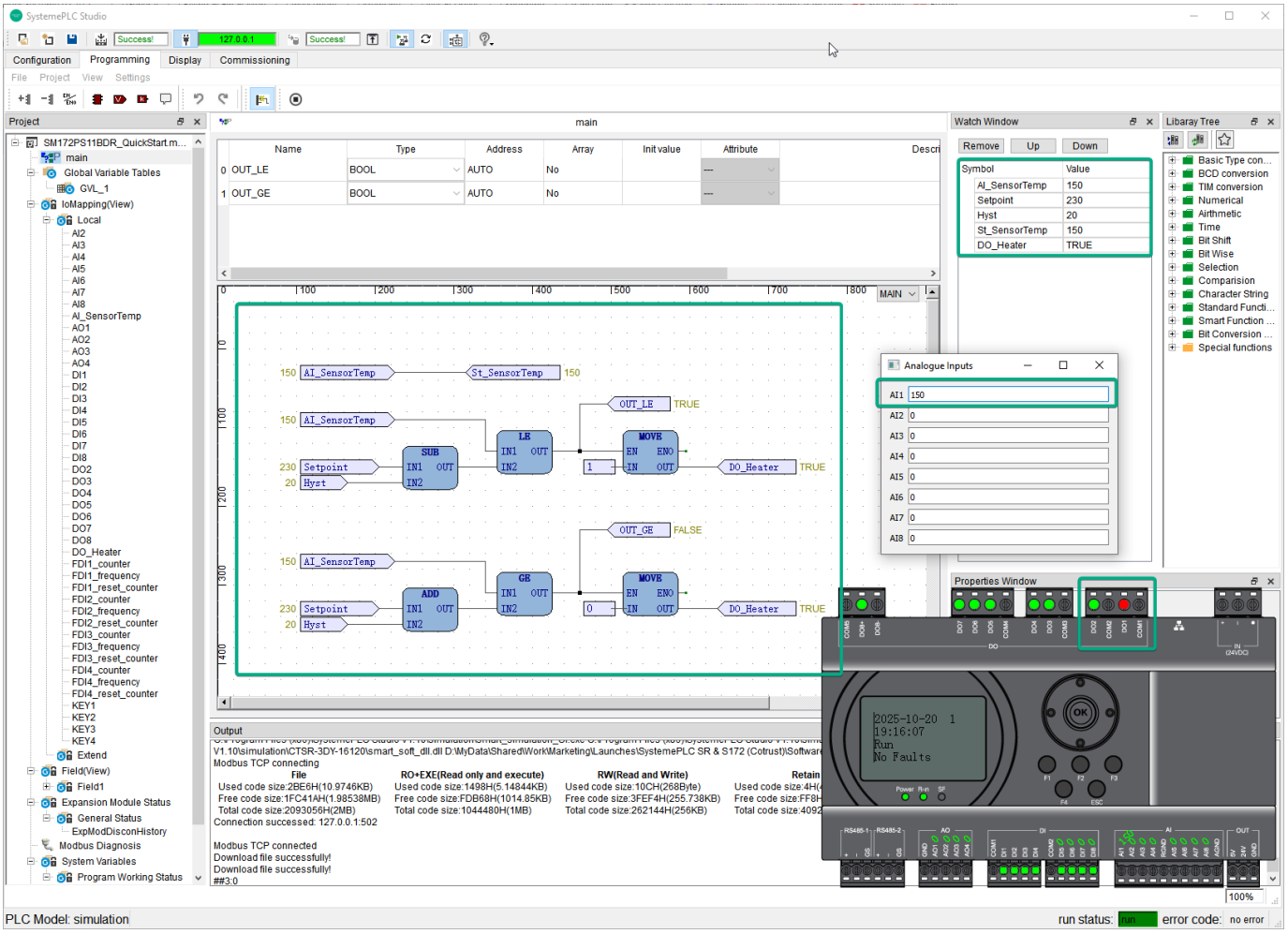
1. Сохраните проект
2. Скомпилируйте получившуюся программу
   a. Окно вывода (Output Window) отобразит результаты. Отсутствие ошибок указывает на успешную компиляцию.
3. Запустите офф-лайн симулятор ПЛК (Simulation Mode)
4. Включите онлайн мониторинг ПЛК (Live Debug Mode)



Работа в оффлайн симуляторе ПЛК демонстрационного проекта

1. Задавайте значение аналогового входа AI1 (переменная AI_SensorTemp)
   a. С помощью задания AI1 из оффлайн симулятора ПЛК
      (1) Дважды щелкните ЛКМ на AI1 на оффлайн симуляторе ПЛК
      (2) Введите значение для AI1
   b. С помощью пользовательского экрана ПЛК
      (1) На симуляторе ПЛК нажмите кнопку OK, далее стрелками вверх/вниз выберите Custompages, нажмите OK. Появятся пользовательские экраны.
      (2) Стрелками вверх/вниз выберите поле Уставка.
      (3) Стрелками вперед/назад переместитесь на нужную цифру
      (4) Стрелками вверх/вниз измените значение уставки (например, с 23.0 на 23.5) и нажмите Ok.

2. Перетащите из дерева проекта в Watch Window переменные AI_SensorTemp, St_SensorTemp, SetPoint, Hyst и DO_Heater для дальнейшего мониторинга.
   a. Наблюдайте за изменениями значений переменных в Watch Window
3. Наблюдайте за исполнением логики программы (с включенным онлайн мониторингом) и значениями переменных в области написания программного кода
4. Наблюдайте за переменной DO_Heater (привязанной к дискретному выходу DO1) на оффлайн симуляторе ПЛК



Логика работы демонстрационного проекта

Логика программы «Нагреватель» заключается в следующем:

1. **Формирование зон срабатывания**
   a. Вычисляется значение уставки минус гистерезис (SetPoint - Hysteresis). Это нижний порог включения нагревателя
   b. Вычисляется значение уставки плюс гистерезис (SetPoint + Hysteresis). Это верхний порог выключения нагревателя
2. **Сравнение и управление выходом**
   a. **Условие ВКЛЮЧЕНИЯ:** если текущая температура (St_SensorTemp) **меньше или равна** нижнему порогу (SetPoint - Hysteresis), то на выходе блока сравнения устанавливается логическая 1 (TRUE).

   b. **Условие ВЫКЛЮЧЕНИЯ:** если текущая температура (St_SensorTemp) **больше или равна** верхнему порогу (SetPoint + Hysteresis), то на выходе соответствующего блока сравнения устанавливается логическая 1 (TRUE)
3. **Условное присваивание с помощью блоков MOVE:**
   a. Для управления выходом DO_Heater используются два блока MOVE
   b. **Первый блок MOVE** передает значение 1 (TRUE) на выход DO_Heater, но только если его вход EN активирован сигналом от условия ВКЛЮЧЕНИЯ
   c. **Второй блок MOVE** передает значение 0 (FALSE) на выход DO_Heater, но только если его вход EN активирован сигналом от условия ВЫКЛЮЧЕНИЯ
   d. **Примечание.** Важность использования блока MOVE с включенными пинами EN/ENO. Без использования управляемого присваивания последний по порядку выполнения блок MOVE всегда бы перезаписывал значение выхода DO_Heater. Использование пина EN гарантирует, что блок срабатывает только при выполнении своего условия, что позволяет организовать приоритетность команд.

Итоговая логика

- Нагреватель включается, когда температура падает ниже (SetPoint - Hysteresis)
- Нагреватель выключается, когда температура поднимается выше (SetPoint + Hysteresis)
- В диапазоне между этими порогами состояние нагревателя остается неизменным

Дополнительные возможности

Защита проекта: через пункт меню File > Security > Encryption можно установить пароль на проект. Выгруженный из контроллера зашифрованный проект нельзя будет открыть без пароля.

Импорт/Экспорт: ПО позволяет экспортировать и импортировать отдельные программы (POU), пользовательские блоки и функции, а также конфигурацию IO Mapping для использования в других проектах.

Watchdog -время выполнения программ ПЛК

- Системный вотчдог контроллера составляет 27 секунд. Если программа не завершает цикл за это время, контроллер перезагружается
- Текущее время выполнения задач (Tasks) можно посмотреть через пункт меню View > Task Time Status

Используемые сокращения и термины

Сокращения

- PC – персональный компьютер (ПК)
- PLC – программируемый логический контроллер (ПЛК)
- LMB (Left Mouse Button) –левая кнопка мыши (ЛКМ)
- RMB (Right Mouse Button) –правая кнопка мыши (ПКМ)
- IO (Inputs/Outputs) – входы/выходы (в/в) ПЛК и модулей расширения
- POU (Program Operation Unit) – программа пользователя
- FB (Function Block) – функциональный блок
- Function Block Diagram (FBD) – язык функциональных блоков
- Continuous Function Chart (CFC) – язык непрерывных функциональных схем
- ВОС – внутренняя операционная система ПЛК [прошивка ПЛК]

Термины частей интерфейса

- Toolbar – панель инструментов
- Tabs – вкладки меню: содержит основные разделы: Configuration (Конфигурация), Programming (Программирование), Display (Отображение), Commissioning (Ввод в эксплуатацию).
- Menu – пункты меню
- Project Tree – дерево проекта
- Library Tree – библиотека элементов
- Properties Window – свойства
- Output – поле вывода
- Status bar – строка состояния
- Programming Toolbar – панель инструментов программирования
- Variable Declaration Area – область объявления переменных
- Program editing area – область написания программного кода
- Display Toolbar – панель инструментов Дисплей
- Page editing area – область создания пользовательских экранов
- Run/Stop – режим работы ПЛК (пуск/стоп)
- Simulation Mode – оффлайн симулятор ПЛК
- Live Debug Mode – онлайн мониторинг ПЛК
- Watch Window – окно наблюдения за переменными

Социальные сети

systemeelectric_official

youtube.com/c/SystemeElectric

vk.com/Systemeelectric

Systeme Electric

www.systeme.ru

Наши бренды

Systême electric

Dēkraft

Механотроника

Systême soft