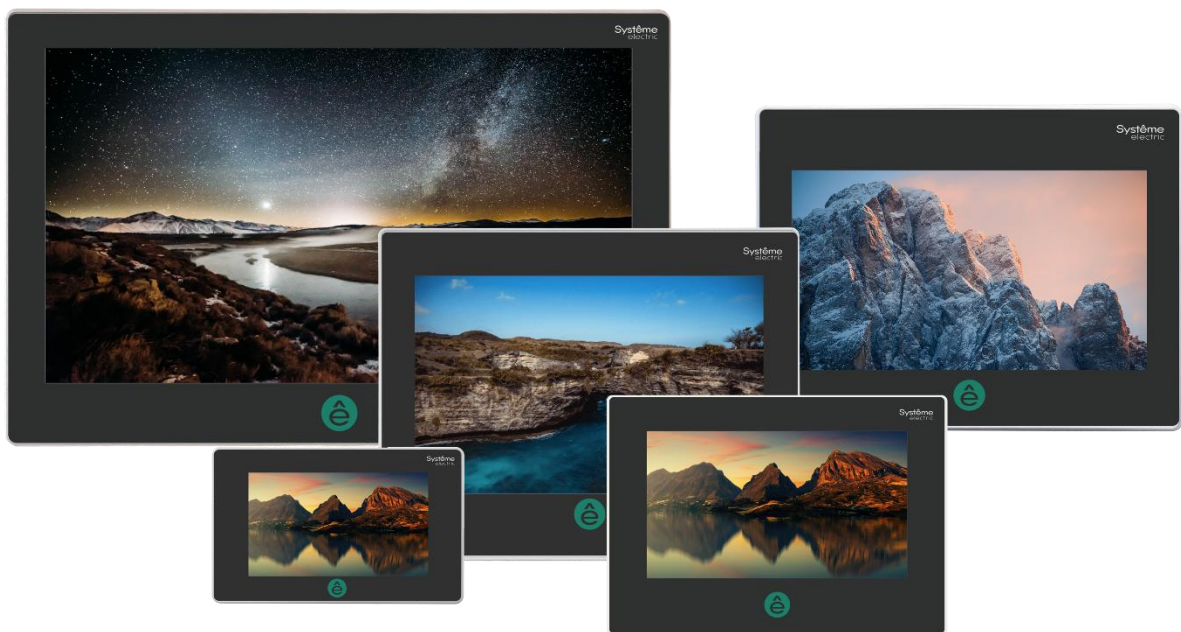


Systeme HMI Studio FAQ



Содержание

Как анимировать расположение, позицию, размер или вращение объекта на экране панели оператора	3
Как импортировать или экспортировать переменные проекта	8
Как настроить конфигурацию проекта.....	13
Как создать внутренние и внешние переменные	17
Как создать новое окно в проекте панели оператора и задать его свойства.....	21
Как создать новый проект? Выбор модели панели оператора.....	31
Как создать основные элементы интерфейса? Кнопки, переключатели, индикаторы, числовые или символьные поля ввода, числовые или символьные дисплеи	36
Какие типы данных поддерживаются?	64
Как настроить связь между панелью HMISGU*** и ПЛК SystemePLC S250	68
Как настроить связь между панелью HMISGU*** и ПЛК SystemePLC S250 по Modbus	74

Как анимировать расположение, позицию, размер или вращение объекта на экране панели оператора

Версия ПО

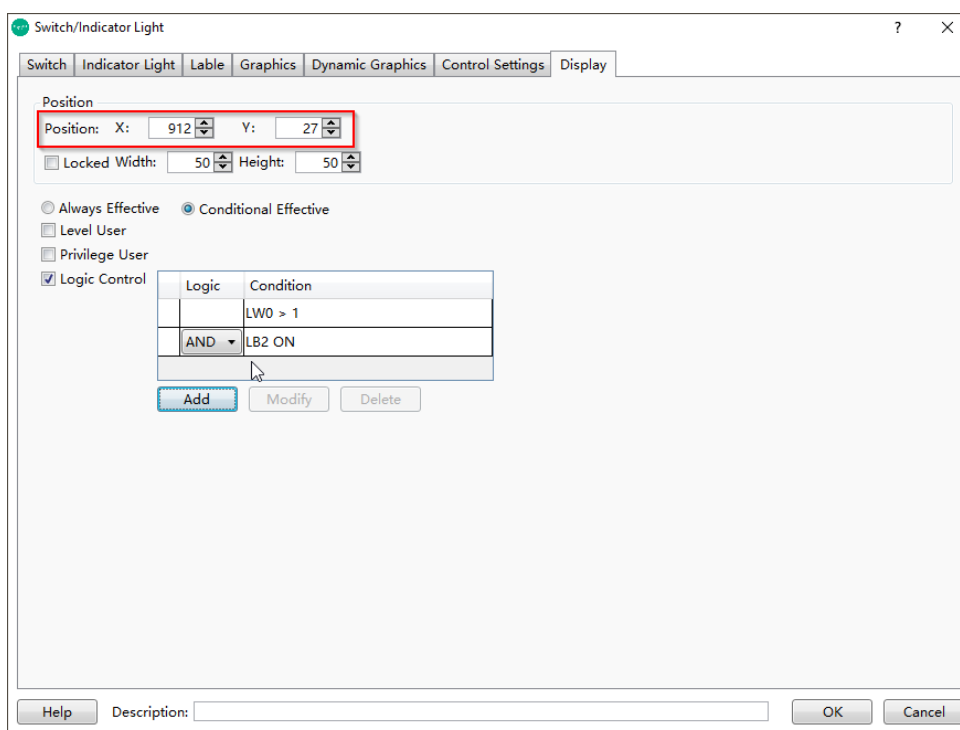
Версия Systeme HMI Studio: 3.0.12783.0

Версия ОС Windows: Windows 10 Pro, 21H2, сборка 19044.2604

В панелях оператора предусмотрено динамическое изменение (Анимация) трех параметров каждого объекта. Это его позиция на экране, размеры и угол поворота.

✓ Позиция

Каждый компонент имеет свойство **«Позиция»{Position}**. Вы можете найти его на вкладке свойств **«Общие»{General}** или на для некоторых компонентов во вкладке **«Отображение»{Display}** свойств компонента.



«X» и «Y» в закладке **«Позиция»{Position}** используются для установки координаты X и координаты Y начальной точки текущего положения компонента в окне. Координата «X» и «Y» — это «Фиксированная точка»{Fixed Point} или «Позиция»{Position}. «Фиксированная точка» определяет фиксированную точку в качестве начальной точки компонента - точку «привязки».

Больше информации по этой теме можно найти в разделе «Общие функции/ Рисование/ Вращение» { General functions/ Drawing/ Rotation}.

Поле «Позиция» определяет точку в верхнем левом углу компонент в качестве отправной точки.

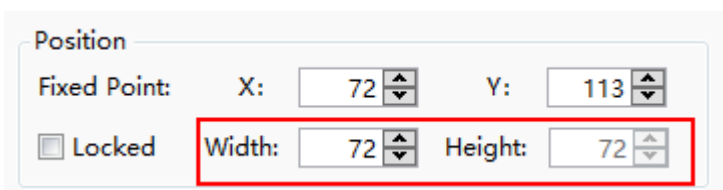
Примечание:

В этом программном обеспечении система координат показана ниже. Исходная точка находится в верхнем левом углу. Ось X является горизонтальным направлением. Положительное направление оси X — вправо. Ось Y имеет вертикальное направление. Положительное направление оси Y — вниз.

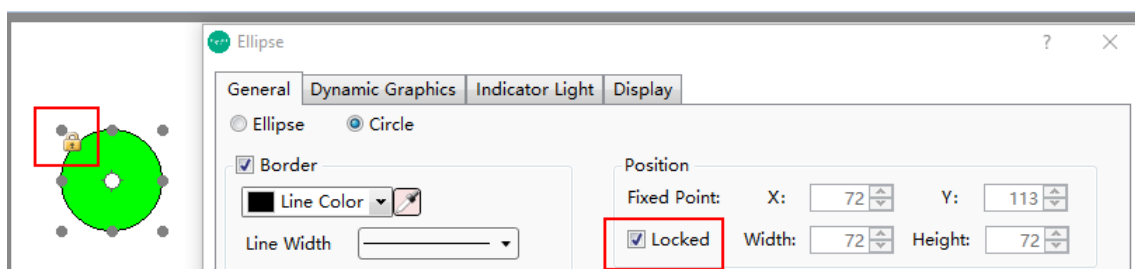


✓ Размеры объекта.

Свойства **«Ширина»{Width}** и **«Высота»{Height}** используются для установки ширины и высоты компонента. Если опция выделена серым цветом и недоступна для редактирования, это означает, что атрибут недоступен. Как показано ниже.



Если Вы отметите опцию «Заблокировано»{Locked}, положение компонента будет зафиксировано. При такой настройке его положение и размер изменить будет нельзя. Метка блокировки будет отображаться в верхнем левом углу компонента, когда вы выбираете его в окне конфигурации.



✓ Вращение{Rotation}.

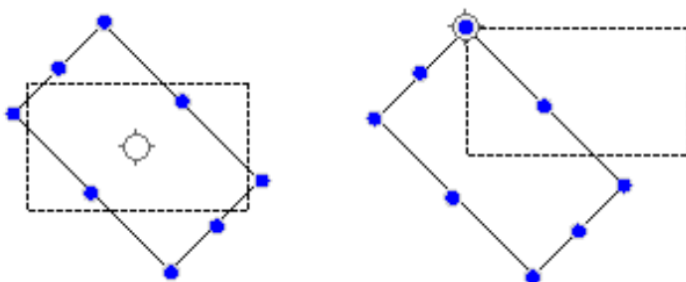
Функция «Вращение» позволяет настроить угол отображения таких компонентов, как статическая графика, векторная графика и другие компоненты.

Эта функция является статической функцией. То есть угол отображения не настраивается в НМІ, если Вы устанавливаете угол отображения в свойстве «Поворот»{RotationAngle}.

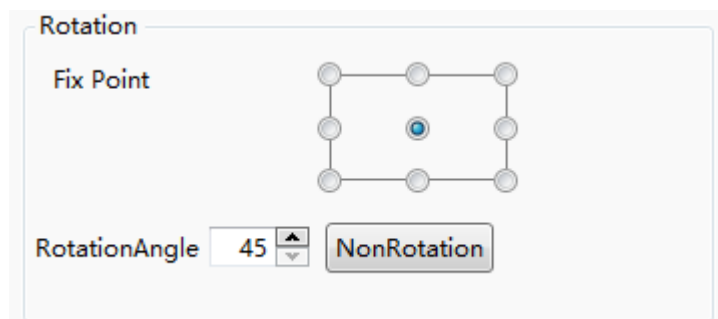
Если вы хотите динамически настроить угол отображения графики, обратитесь к разделу Общие функции/Анимация»{ General functions\Dynamic Graphics}.

- Точка фиксации{Fixed Point}

Каждая фигура имеет девять точек фиксации. Средняя точка фиксации выбирается системой по умолчанию. Вы можете изменить эту точку. Результат поворота будет другим, если фигура вращается вокруг другой фиксированной точки. Например, прямоугольник вращается относительно центральной точки фиксации, а другой прямоугольник вращается относительно верхней левой точки фиксации. Как показано ниже.



- Угол поворота{RotationAngle}.



Угол поворота используется для установки угла поворота компонента по часовой стрелке. Этот диапазон углов составляет 0-360 градусов.

- Не вращать{NonRotation}.

Вы можете быстро установить угол отображения на ноль, нажав кнопку «Не вращать»{NonRotation}.

- ✓ Анимация{Dynamic Graphics}

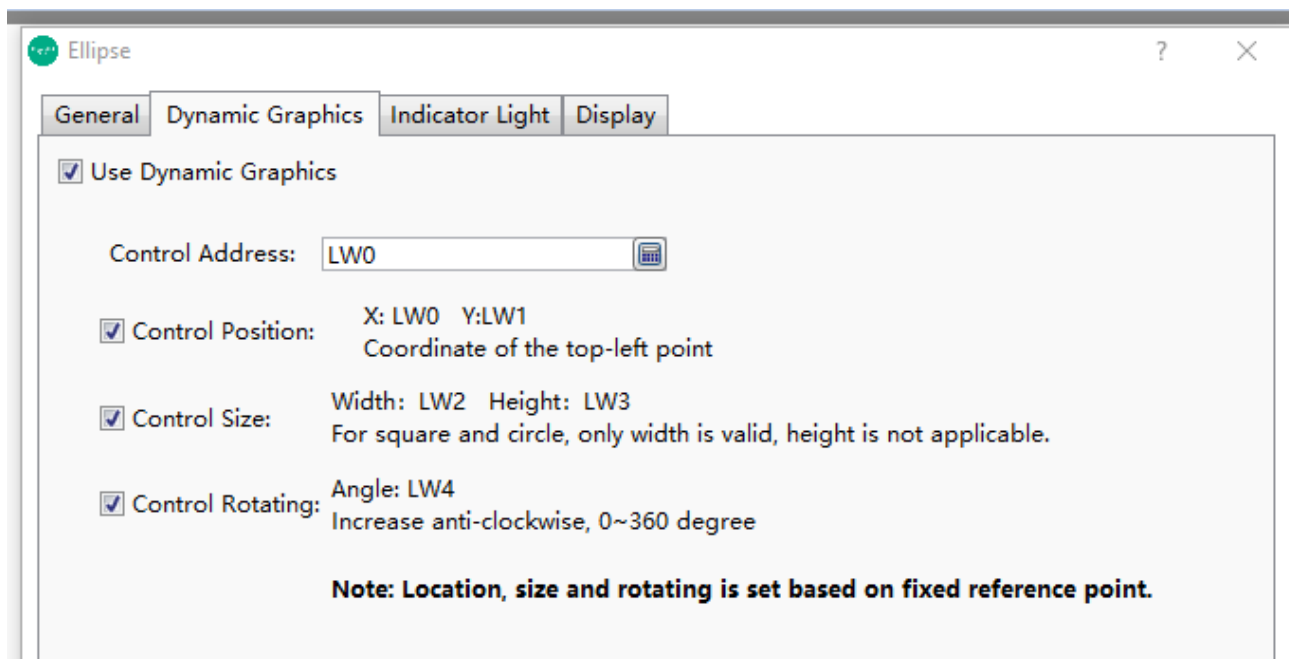
Существует закладка свойств «Анимация» для некоторых типов компонентов, таких как: компонент статического изображения, векторная графика и т. д.

Вы можете использовать функцию «Анимация», если хотите динамически регулировать положение, размер и угол поворота этого компонента (фигуры) во время работы панели HMI.

Функция анимации реализуется за счет использования регистров для динамического управления *положением, размером и углом поворота* фигуры.

- Использовать анимацию {Use Dynamic Graphics}.

Вам необходимо установить флажок «Использовать анимацию» на вкладке свойств «Анимация» {Dynamic Graphics}, если вы хотите использовать эту функцию.



- Адрес управления{Control Address}
Здесь выбирается начальный адрес регистров управления. Как задать адрес регистра управления описано в разделе: Общие функции/Редактор адреса/Стандартный ввод байтового адреса { General function/Address editor/Standard Byte Address Input.}.

- Управление позицией{Control Position}
Параметр «Управление позицией» должен быть отмечен, если вы хотите динамически регулировать положение компонента. Используются два регистра. *Адреса будут установлены и отображены автоматически после того, как будет задан атрибут «Адрес управления».* Эти два регистра будут контролировать абсолютные координаты фиксированной точки компонента на экране.

Система координат сенсорного экрана описана в разделе: Общие функции/Рисование/Позиция{General function/Drawing/Position}.

- Управление размером{Control Size}
Параметр «Управление размером» должен быть отмечен, если вы хотите динамически регулировать размер компонента. Используются два регистра. Адреса будут установлены и отображены автоматически после того, как будет задан атрибут «Адрес управления».

Эти два регистра будут контролировать ширину и высоту компонента. Регистр для управления шириной будет использоваться для компонентов, ширина и высота которых равны, таких как квадрат и круг.

- Управление вращением{Control Rotation}

Параметр «Управление вращением» должен быть отмечен, если вы хотите динамически регулировать угол поворота компонента. Используется один регистр. Адрес будет установлен и отображен автоматически после того, как будет задан атрибут «Адрес управления». Содержимое регистра будет определять угол поворота компонента. Угол увеличивается против часовой стрелке. Диапазон углов составляет 0-360 градусов.

Примечание:

Положение, размер и поворот отсчитываются от точки фиксации компонента. Точка фиксации {Fixed Point} описана в разделе: Общие функции/Рисование/Вращение{ General function/ Drawing/ Rotation}.

Как импортировать или экспортировать переменные проекта

Версия ПО

Версия SystemeHMI Studio: 3.0.12512.0

Версия ОС Windows: Windows 10 Pro, 21H2, сборка 19044.2604

Библиотека Адресных Меток (Address Tag Library).

«Библиотека адресных меток» сохраняет адреса регистров в форме меток (тегов). Это упрощает их использование и изменение адреса для пользователя.

Библиотеку адресных меток можно открыть, щелкнув команду меню «Library/Address Tag Library (Библиотека/Библиотека адресных меток)» или дважды щелкнув «Library/Address Tag Library» в дереве проекта.

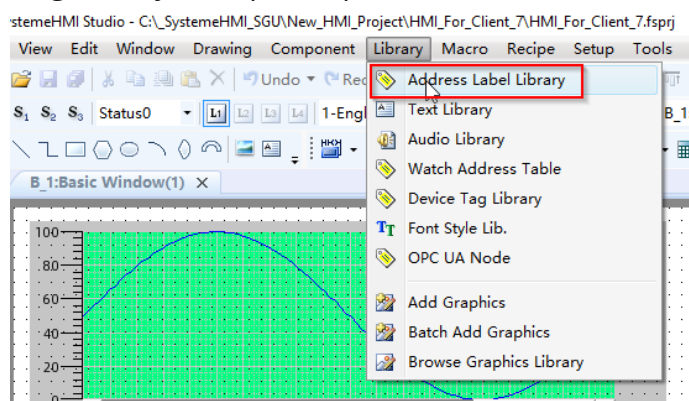
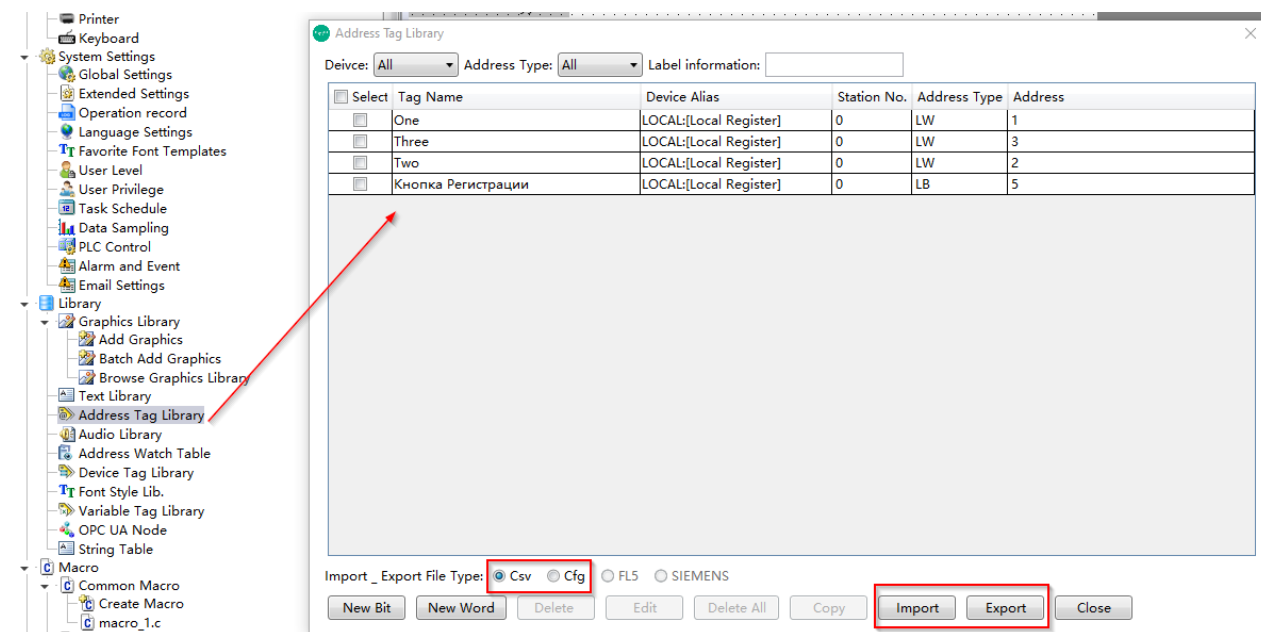


Таблица в «Библиотеке адресных тегов» отображает информацию обо всех адресных

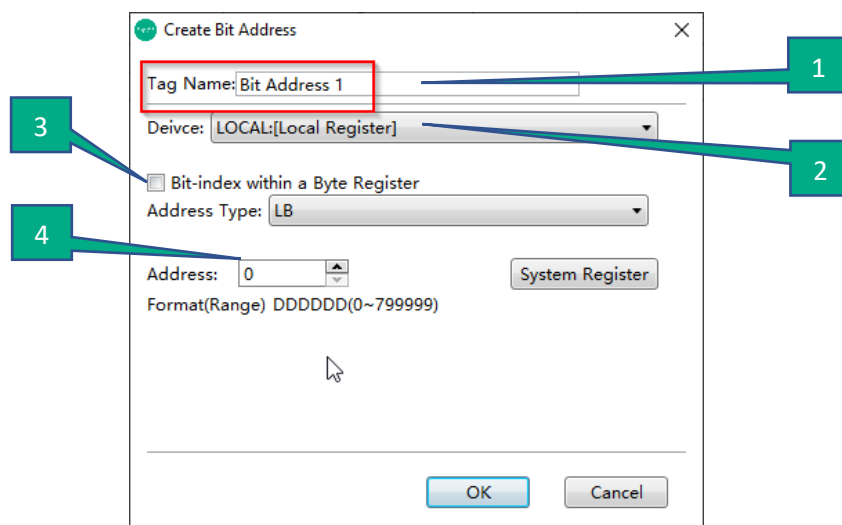


тегах.

Данные из этой библиотеки, с назначенными для них разработчиком приложения именами, могут быть экспортированы во внешний файл с расширением .csv или .cfg для последующего редактирования или импортированы из другого проекта.

Создание меток в Address Tag Library, для используемых в проекте адресов, происходит двумя основными способами:

- New Bit (Новый Bit)



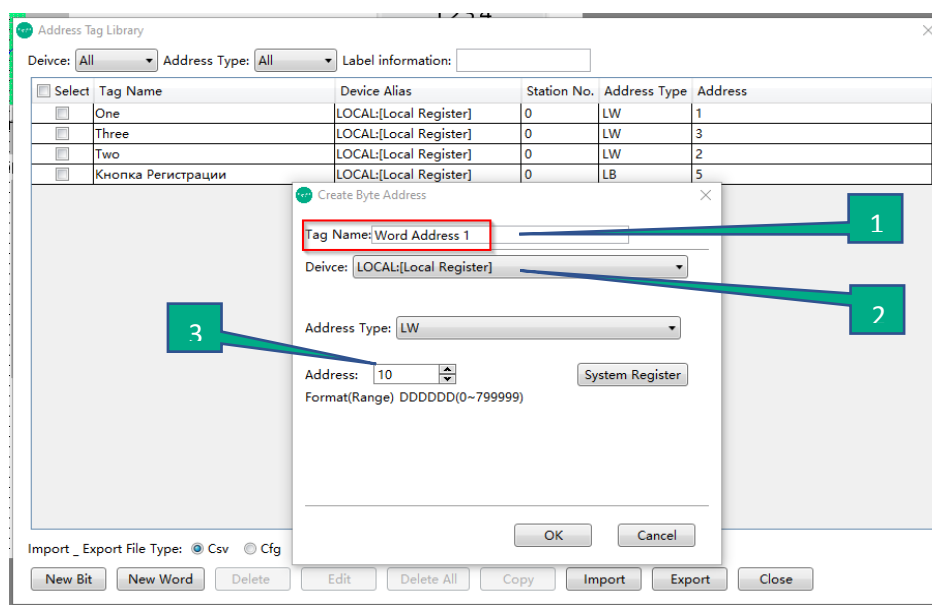
После нажатия кнопки «New Bit» появится диалоговое окно «Create Bit Address

Создать битовый адрес». Здесь вы можете создать новый битовый адрес, соответствующий биту данных. Подробнее о вводе битового адреса Вы можете прочитать в соответствующем разделе.

1. Имя адресной метки.
2. Выбор источника данных - устройство или внутренняя память панели.
3. Адрес бита данных.
4. "Bit-index"- позволяет назначить адресную метку для определенного бита в слове данных.

- New Word (Новый Word)

После нажатия кнопки «New Word» появится диалоговое окно «Create Byte Address» создания новой метки, соответствующей регистру данных. Подробнее о вводе байтового адреса Вы можете прочитать в соответствующем разделе.



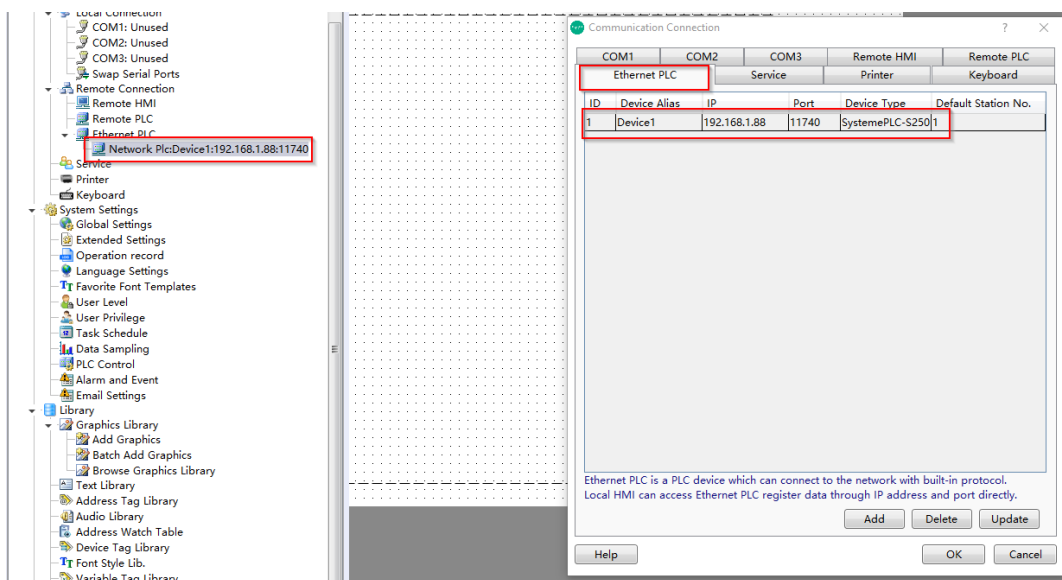
1. Имя адресной метки.
2. Выбор источника данных – удалённое устройство или внутренняя память панели.
3. Адрес регистра данных

- **Delete** - удалить выделенный элемент из библиотеки.
- **Edit** - изменить выделенный элемент.
- **Delete All** - удалить все элементы из библиотеки.
- **Copy** - копировать выделенный элемент.
- **Import** - импортировать элементы библиотеки из файла.
- **Export** - экспортировать элементы библиотеки в файл.

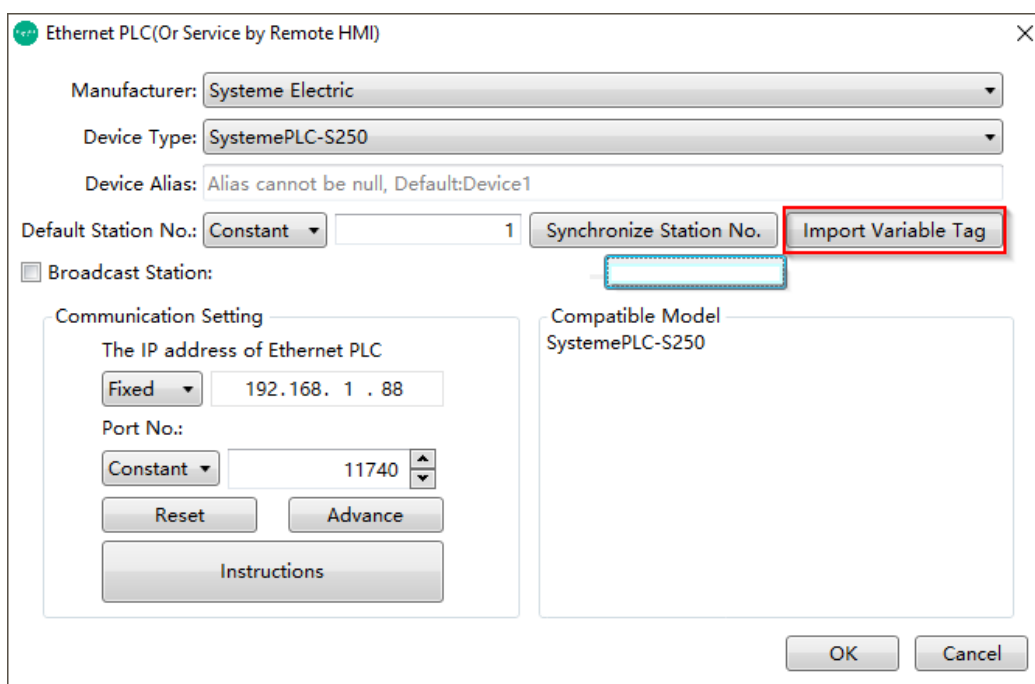
Импорт переменных из PLC (для Advanced панелей оператора).

Усовершенствованные (Advanced) панели оператора: HMISGU70PEA, HMISGU101MEA, HMISGU156ME поддерживают функцию Label Based Access Control (LBAC), включающей в себя:

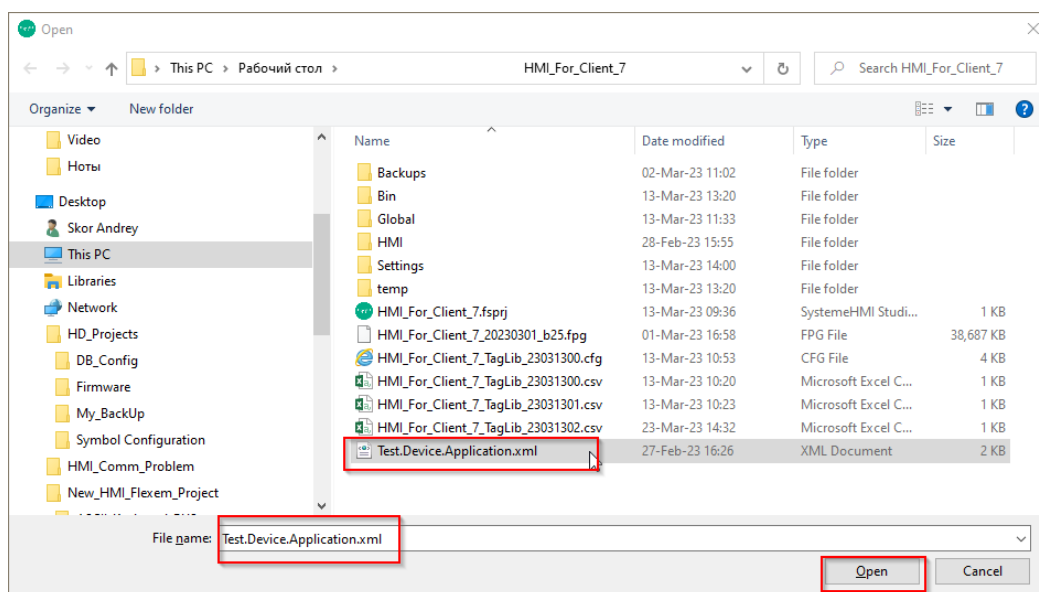
- Прямой импорт тегов ПЛК
- Анализ структуры тегов ПЛК (массив, структура и т.д.)
- Возможность использование массивов текстовых значений тэгов (Array-type Tags) для рецептов.
- и др.



После того как в проекте панели оператора мы выбрали тип подключенного к ней PLC, мы можем сразу импортировать список переменных с помощью файла .xml. На страничке конфигурации подключенного к панели PLC есть кнопка «Import Variable Tag»

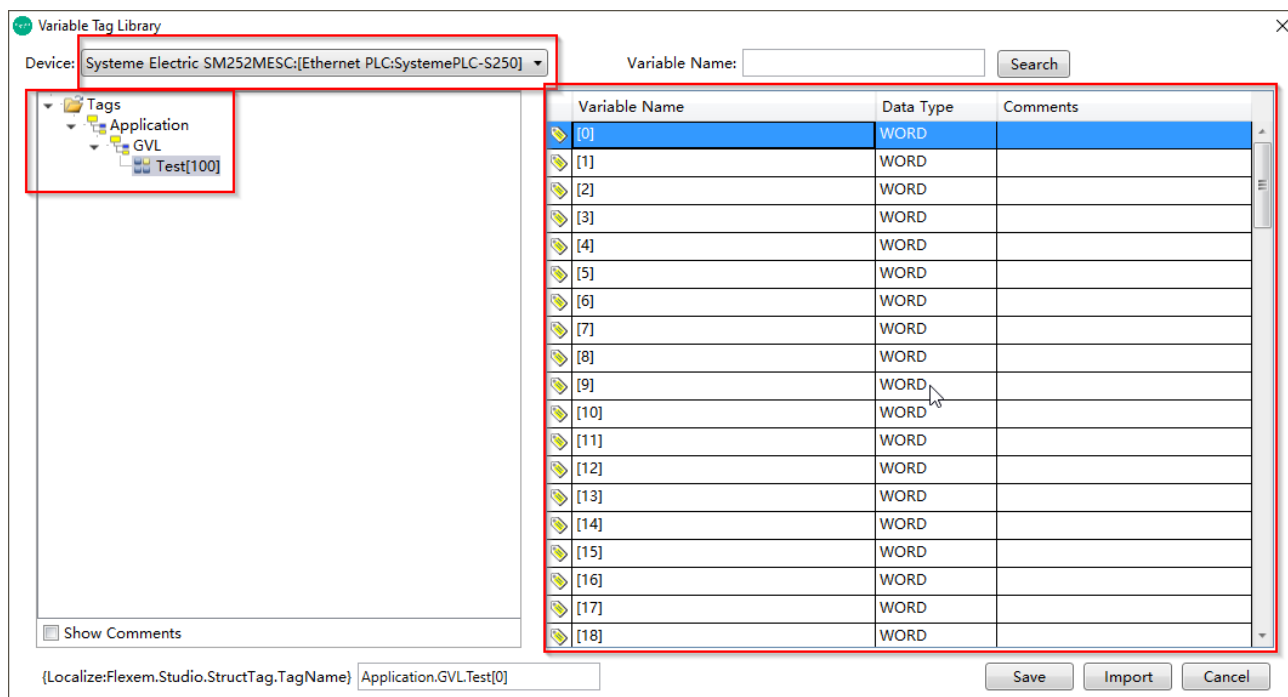


При нажатии на которую открывается окно доступа к файлу .xml, в котором нужно указать путь к требуемому файлу.



После нажатия кнопки Open (Открыть) список подготовленных переменных импортируется в проект панели и сохраняется в библиотеке переменных «Variable Tag Library».

В данном примере мы подготовили к переносу одномерный массив из 100 переменных типа Word.



Все импортированные из файла переменные сохраняются в библиотеке «Variable Tag Library» и становятся доступными для использования в проекте панели оператора.

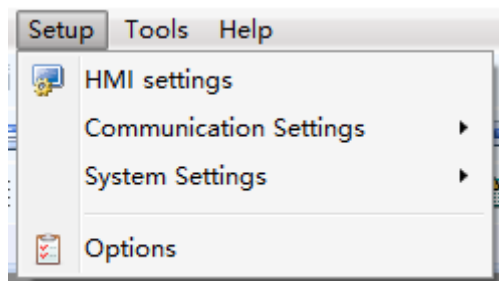
Как настроить конфигурацию проекта

Версия ПО

Версия SystemeHMI Studio: 3.0.12512.0

Версия ОС Windows: Windows 10 Pro, 21H2, сборка 19044.2604

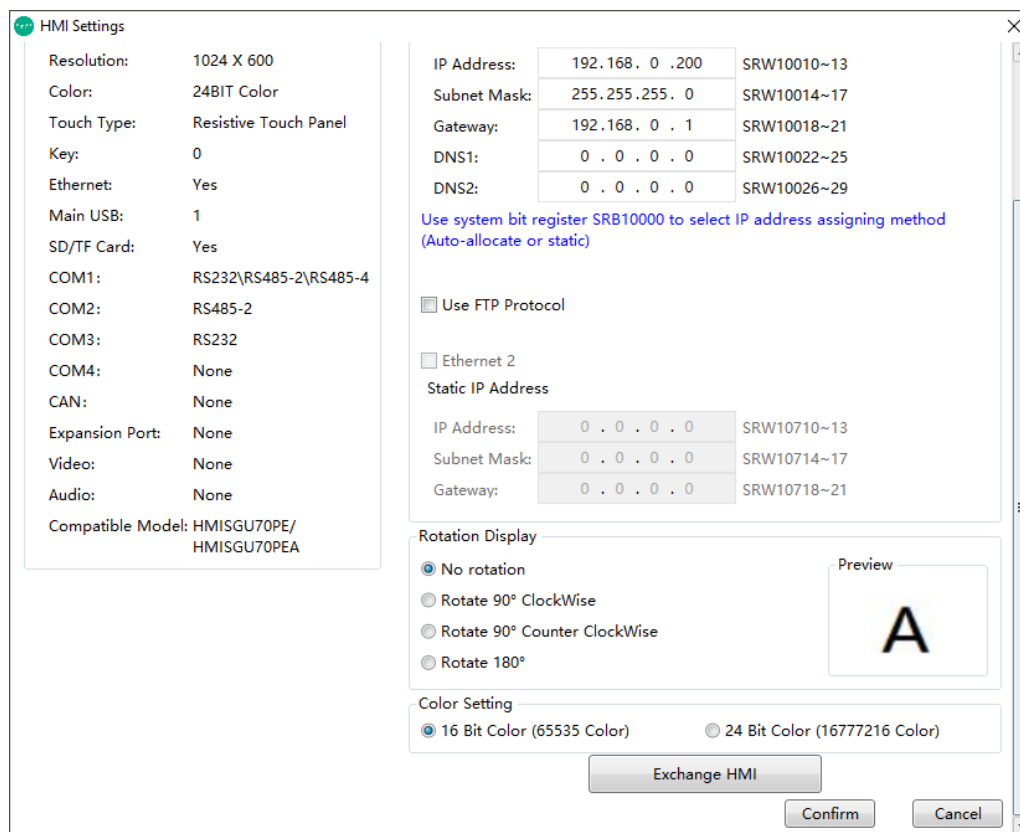
- A. В разделе **Setup > HMI Settings** предоставляется возможность выполнить основные настройки в процессе создания нового проекта, однако в случае необходимости, можно изменить настройки в любой момент времени



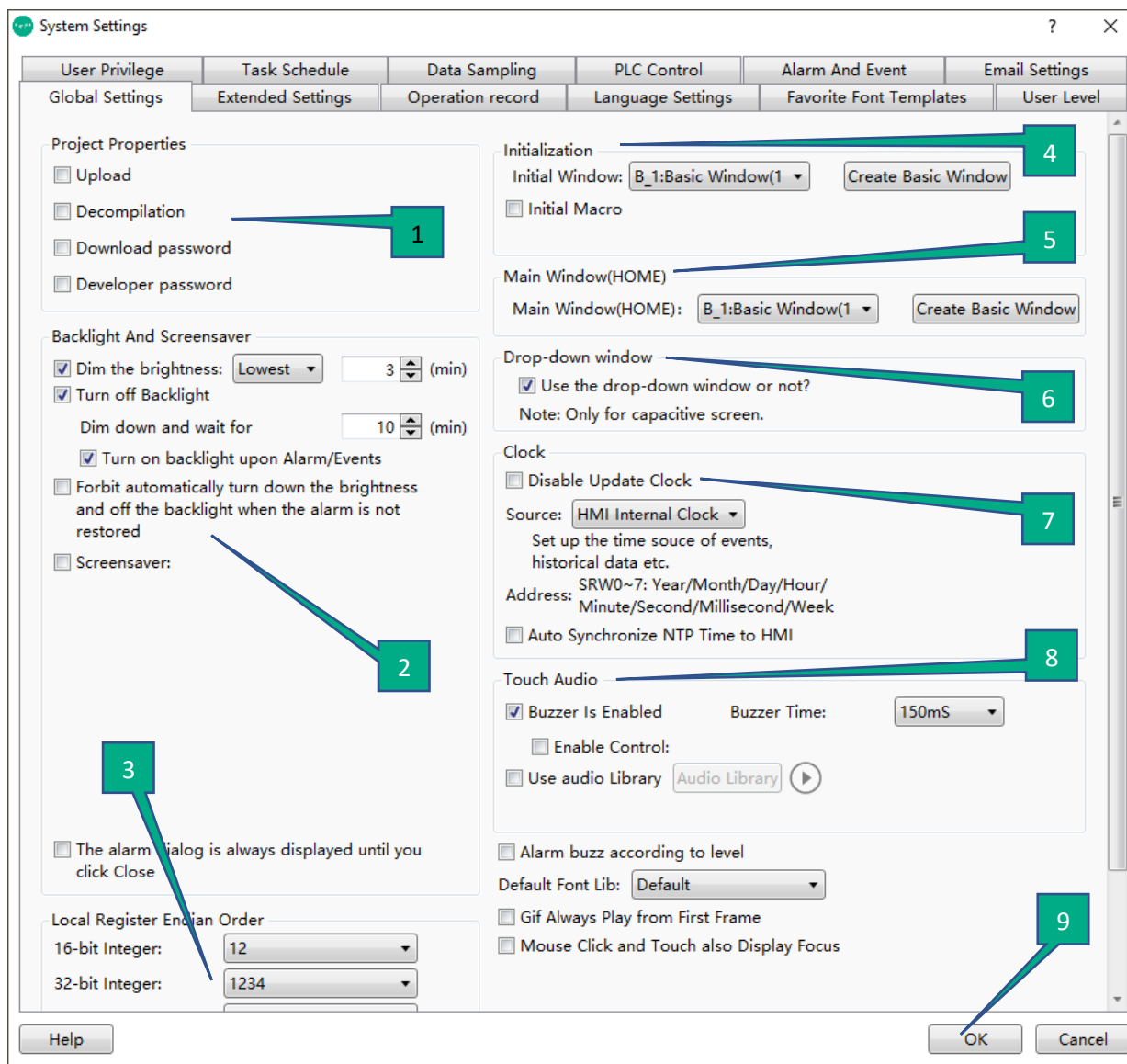
В случае необходимости переноса проекта с одного типа панели на другой, можно воспользоваться функцией "**Exchange HMI (Сменить тип HMI)**". Для этого щелкните левой клавишей мыши по кнопке «**Exchange HMI**», расположенной в нижней части экрана и в появившемся окне выберите новый тип панели из выпадающего списка.

Если разрешение экранов старой и новой панелей оператора не совпадают, то в процессе замены все экраны проекта будут

Перерисованы автоматически в масштабе кратном соотношению разрешений двух панелей, но может потребоваться и ручная корректировка визуальных элементов и шрифтов подписей.



В. Раздел **Setup > System Setting (Настройка Системы) > Global Setting (Общие настройки)**



1. Project Properties - Настройки защиты проекта от несанкционированного доступа или изменения:

- **Upload** (Выгрузка) - опция защиты проекта от выгрузки из панели оператора.

- **Decompilation** (Декомпиляция) - опция защиты выгруженного проекта от декомпиляции.

- **Download Password** (Защита паролем на загрузку проекта) - опция защиты загруженного в панель проекта от перезаписи новым.

Для активации желаемой опции установите галочку напротив и в появившемся поле введите пароль, который будет необходим в дальнейшем при попытке выполнить запрещенное действие.

- **Developer Password** - Активация парольной защиты разработчика проекта.

2. Backlight and Screensaver -Настройки подсветки и экранной заставки.

При активации опций затенения экрана и отключения подсветки, необходимо задать время бездействия сенсора до момента активации функции. Для экрана-заставки, также задается время бездействия и выбирается окно проекта из списка существующих, которое будет активировано при окончании временного интервала.

Выход из режима отображения экрана-заставки можно защитить паролем, для этого активируется соответствующая опция и определяется уровень доступа, которым должен обладать пользователь для совершения операции.

3. Local Register Endian Order -Настройки порядка расположения байт в регистрах памяти при работе с данными 16 и 32 разрядов

4. Initialization - Настройки процесса инициализации при запуске проекта. В данном разделе выбирается стартовое окно и макрос, который будет выполнен при начальной загрузке панели и старте проекта. И окно, и макрос выбираются из выпадающего списка доступных, поэтому они должны быть добавлены в проект до настройки данной опции.

5. Main Window (HOME) - Настройка окна по умолчанию, которое будет отображено при вызове функций перехода к домашнему окну.

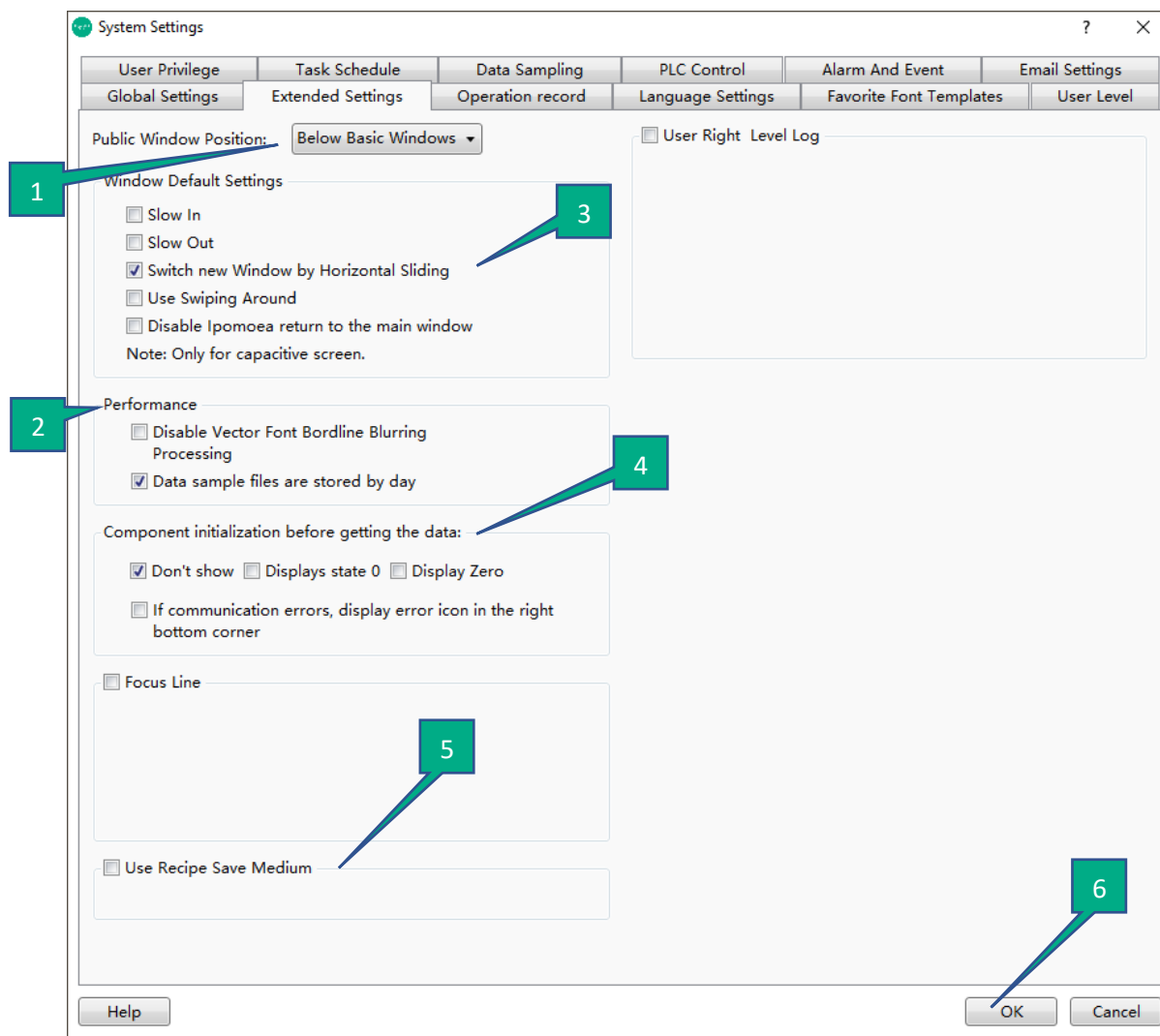
6. Drop-down window - Настройка выпадающих меню. Установите галочку если хотите использовать данную опцию интерфейса, при этом следует помнить, что работа данной опции может быть не корректной на панелях с резистивным сенсором экрана ввиду особенности конструкции.

7. Clock - Настройка встроенных часов. По умолчанию в проекте используются встроенные в панель часы реального времени, однако можно также выбрать часы подключенного оборудования в качестве источника времени. При этом должно быть настроено хотя бы одно внешнее подключение.

8. Touch Audio - Настройки зуммера используются для управления встроенным источником звука, он может быть включен или отключен в зависимости от активных опций, а также его состояние может зависеть от состояния внешнего регистра если необходимо оперативно менять его состояние.

9. Кнопка «OK» для сохранения произведённых настроек и закрытия этого окна.

C. Раздел **Setup > System Setting (Настройка Системы) > Extended Setting (Расширенные настройки).**



1. Public Window Position - Настройка отображения окна шаблона и соответственно порядка обработки событий касания экрана.

- **Below Basic Window** - Под основным окном - окно шаблон размещено под основным окном и элементы основного окна перекрывают элементы окна шаблона.

- **Above Basic Window** - Окно шаблон размещено над основным окном и элементы основного окна перекрывают элементы окна шаблона.

2. Performance - Настройка производительности интерфейса. Для повышения производительности можно отключить опцию обработки векторных шрифтов.

3. Window Default Setting - Настройка опций управления экраном. (Для панелей с ёмкостным сенсорным экраном).

4. Component Initialization before getting the data - Настройка начального отображения компонента, внешний вид которого зависит от внешнего условия или данных.

5. Use Recipe Save Media - Использовать опцию сохранения рецептов на перечисленные носители (HMI, SD Card, USB1).

6. Кнопка «ОК» для сохранения произведённых настроек и закрытия этого окна.

Как создать внутренние и внешние переменные

Версия ПО

Версия SystemeHMI Studio: 3.0.12512.0

Версия ОС Windows: Windows 10 Pro, 21H2, сборка 19044.2604

Прежде всего попытаемся пояснить, что, подразумевается под внутренними и внешними переменными применительно к проектам для панелей оператора.

Переменная - это область памяти имеющая адрес или имя, ссылку на которую можно использовать внутри проекта панели оператора для осуществления доступа к данным.

Переменные могут быть связаны с областями памяти разных устройств.

Под *внутренней* (локальной, Local) памятью, в нашем описании, подразумевается область памяти хранения данных в самой Панели оператора SystemeHMI. А под *внешней* (удалённой, Remote) подразумевается область памяти того устройства, которое подключено к панели оператора (например, удалённого ПЛК).

Основываясь на этом определении, переменные тип которых задан как *внутренние*, располагаются в памяти самой панели оператора, используются самой панелью во время работы загруженного в неё проекта, но не подлежат передачи по каналам коммуникаций с другими устройствами, а *внешние* переменные - это переменные, значения которых также может быть использовано в процессе функционирования проекта, но содержимое которых передается панелью в удаленные устройства или получается из них по тем или иным коммуникационным каналам. *Внешние* переменные всегда «привязаны» к тому или иному внешнему устройству.

В процессе создания проекта панели оператора, в качестве внутренних переменных или внутренних регистров, Вы можете использовать все типы регистров (LW, RW, SRW, LB, SRB).

Описание типов переменных (Регистров LW, RW, SRW, LB, SRB) смотри в статье «Какие типы данных поддерживаются?»

При двойном щелчке правой кнопкой мыши на каком-либо графическом элементе, помещённого на экран проекта панели оператора, открывается окно задания переменной для управления тем или иным свойством этого элемента.

Вот, например, для графического элемента ввода числовых значений мы определяем внутреннюю переменную типа Word (LW) с адресом (например 1). Именно в этой переменной будет сохраняться значение числа, которое мы будем вводить на экране панели оператора.

Numeric Input ? X

General | Number Format | Keyboard Setting | Font | Graphics | Dynamic Graphics | Control Settings | Communication | Display

Operation Attribute: ☐ Numeric Display ☒ Numeric Input ☐ Characters Display ☐ Characters Input

Display Mode: ☐ Password

☐ Reading And Writing Address Is Different

Read Address:

☐ Use Address Tag

☐ Use Recipe Index

Device: LOCAL:[Local Register]

Address Type: LW

Address: 1 System Register

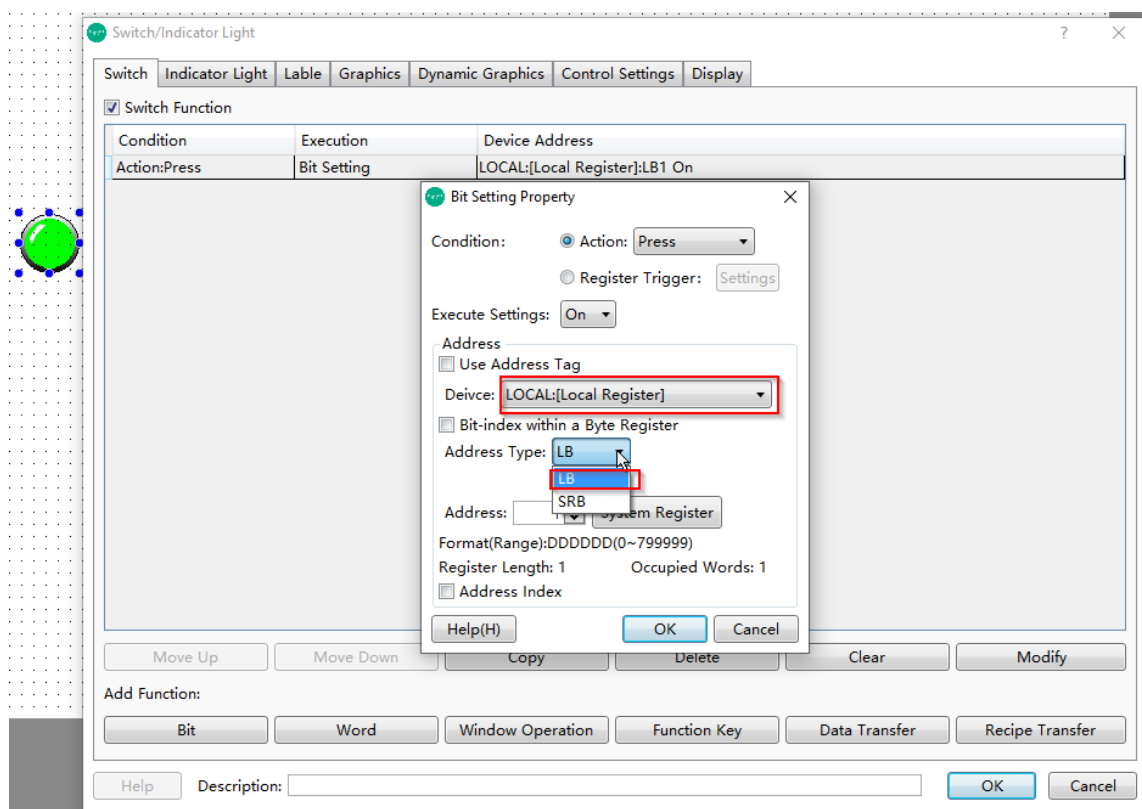
Format(Range):DDDDDD(0~799999)

Register Length: 1 Occupied Words: 1

☐ Address Index

Help Description: OK Cancel

А для хранения значения состояния графического элемента «Переключатель» мы определяем внутреннюю переменную типа Bit (LB) у которой может быть только два состояния «0» и «1», они будут соответствовать состоянию этого «переключателя» - «Включен» или «Выключен».

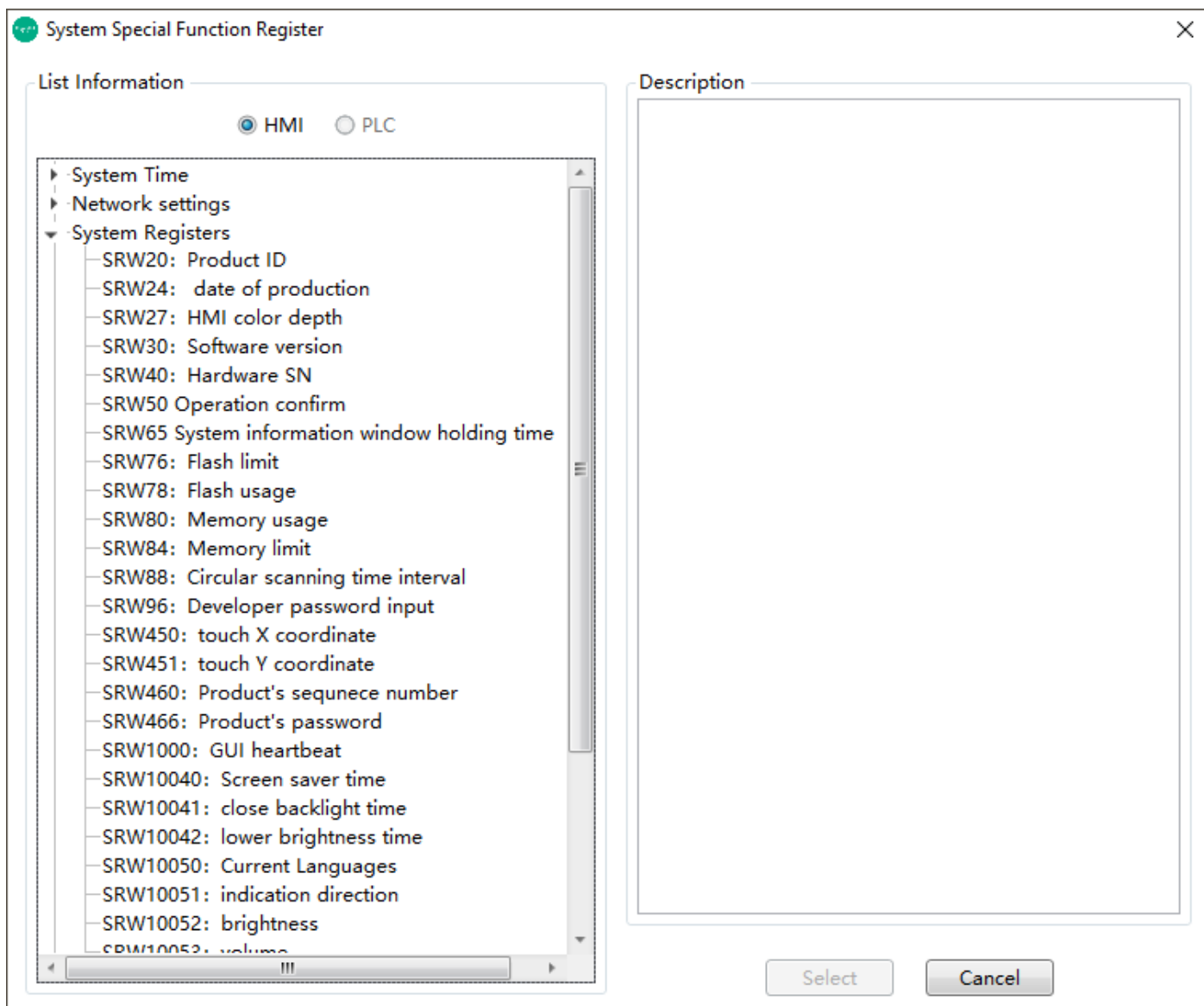


В среде программирования «SystemeHMI Studio» в качестве переменных можно использовать

«Специальные системные регистры функций» (**System Special Function Register**) за которыми производителем панелей закреплены разнообразные действия (функции) которые могут потребоваться пользователям панелей при их повседневной эксплуатации.

Кнопка **System Register** (Системный регистр), которая присутствует во всех меню задания переменных, открывает список **System Special Function Register** (Системный специальный Регистр функций), для выбора конкретной функции, закреплённой за этим регистром SRW.

Например изменение параметров коммуникационных портов, вывод на экран системного времени, смена языка интерфейса, изменение яркости экрана и многое другое в различных подразделах полного перечня регистров (List Information).



Как создать новое окно в проекте панели оператора и задать его свойства

Версия ПО

Версия SystemeHMI Studio: 3.0.12512.0

Версия ОС Windows: Windows 10 Pro, 21H2, сборка 19044.2604

Проект панели оператора может состоять из нескольких графических окон, в которых расположены элементы управления и индикации. Количество окон, элементов управления и индикации, их расположение на экранах пользователь определяет самостоятельно в зависимости от требований к проекту.

Перед тем как пользователь приступит к задаче управления всеми имеющимися в его проекте окнами, необходимо все эти окна создать.

В SystemeHMI Studio следующие типы окон.

- Окно **Power-on Screen** – Начальный экран. Появляется однократно после подачи питания на панель оператора и может содержать, например заставку с логотипом компании.
- Окно **Public (Common) Window** – Графические элементы, помещённые на это окно, будут отображаться на всех остальных окнах.
- Окно **Base Window** – Основной тип окон, которых может быть несколько в проекте.

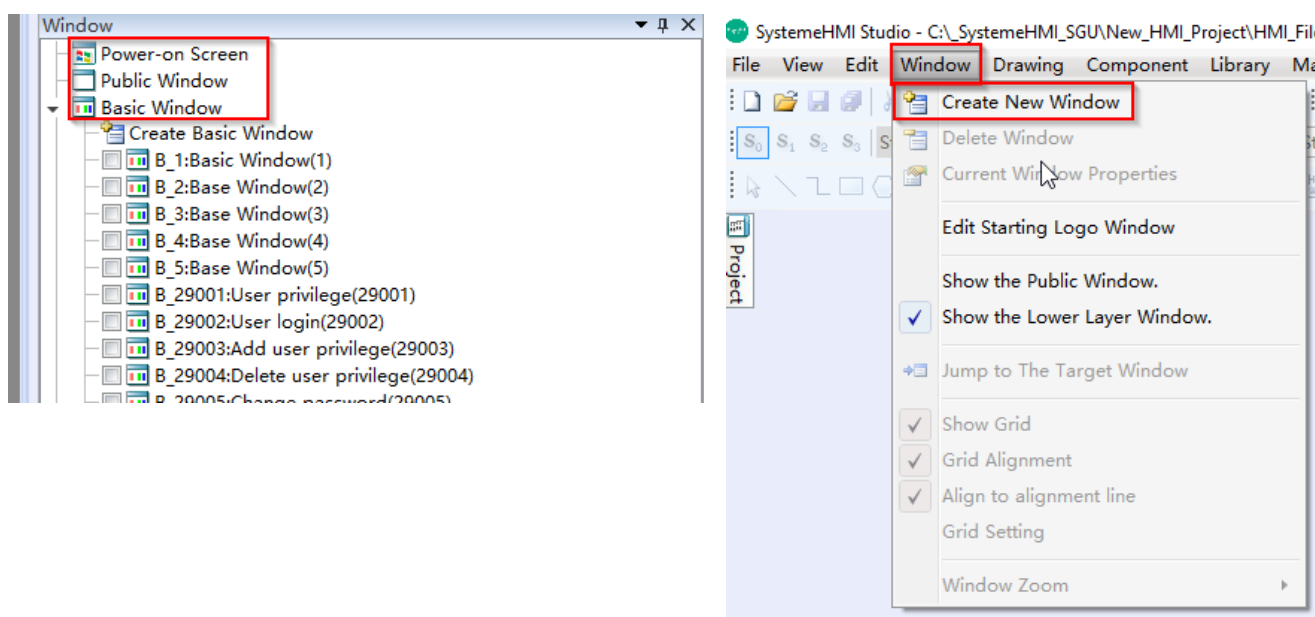
Основное окно (Base Window) может быть выведено на экран в виде всплывающего окна. Для этого предусмотрена опция "Popup Window".

- В качестве специальных окон, в проекте пользователь может использовать:

Окно **Клавиатуры {Keyboard Window}** (для ввода символов или числовых значений с экрана панели) и

Системное окно {System Window} (для вывода информационных сообщений).

В этой главе мы будем говорить о создании и настройке окон **Base Window**.

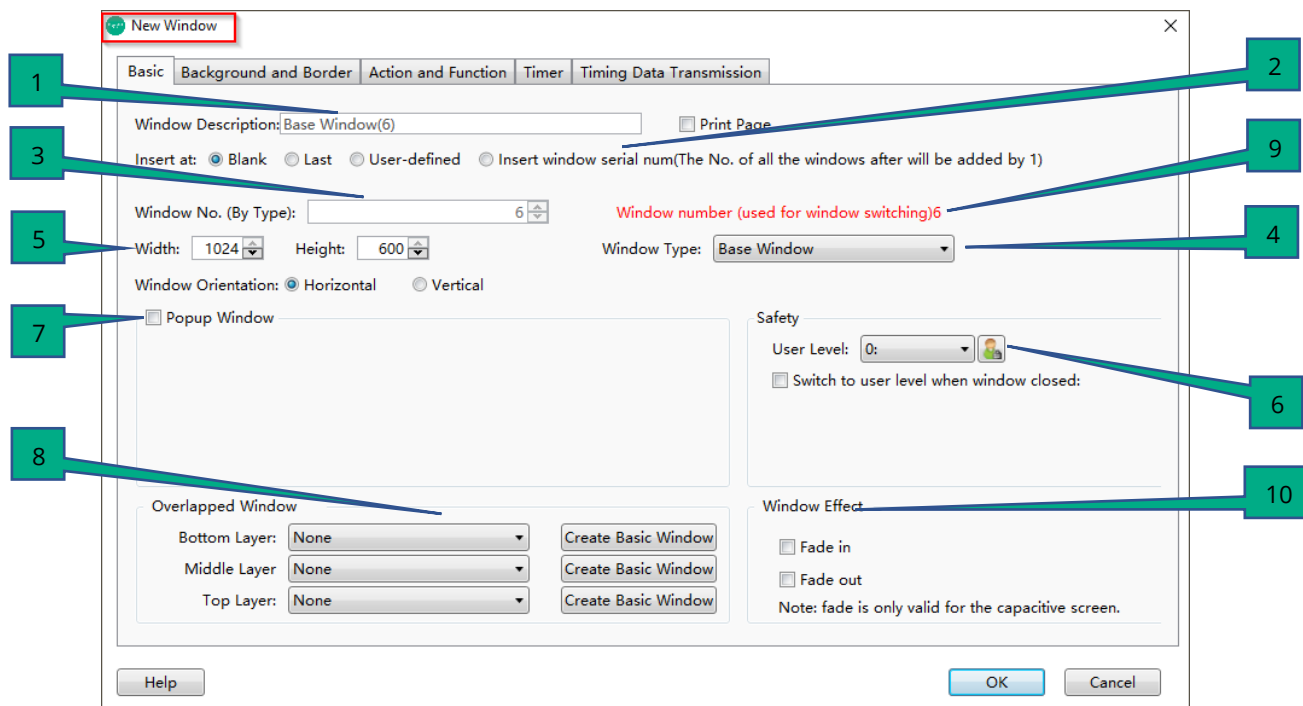


Для создания нового окна необходимо открыть закладку *Window* и выбрать опцию **Create New Window (Создать новое окно)**.

В результате открывается диалог создания нового окна (*New Window*). Все доступные настройки и функции объединены в несколько закладок в верхней части диалогового окна.

✓ Основные (Basic) настройки.

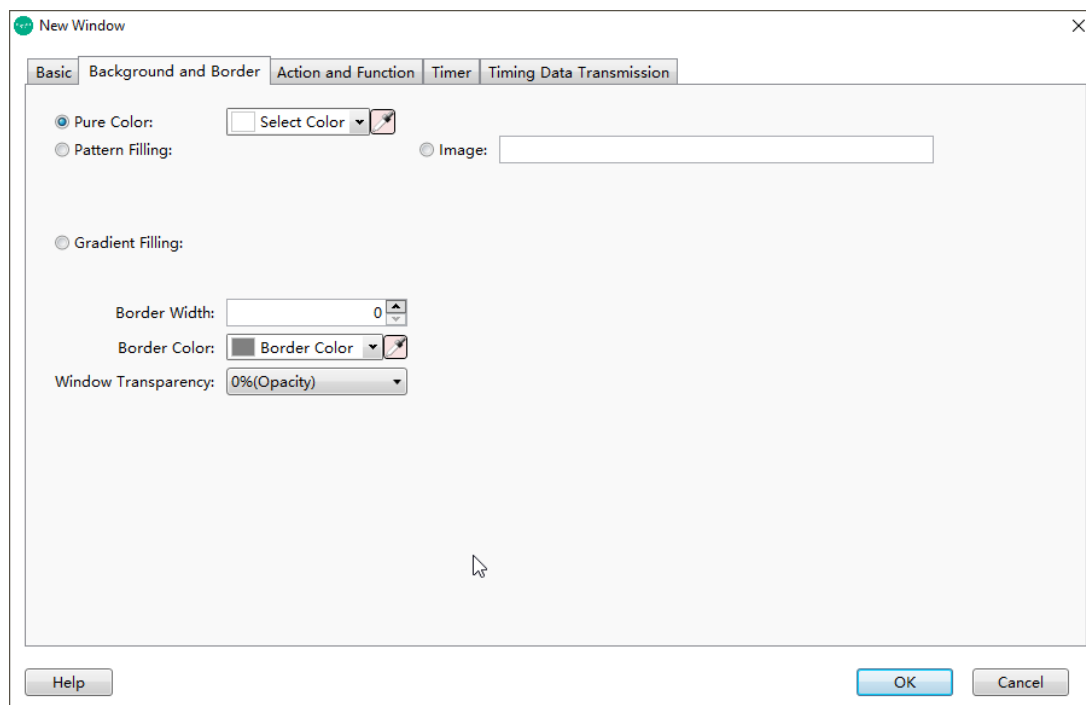
Вкладка используется для настройки основных свойств нового окна. Внешний вид и назначение основных полей представлено на рисунке.



- Имя или краткое описание нового окна для упрощения идентификации окон при редактировании проекта.
- Порядок нумерации окон при добавлении нового.
 - Blank** – Номер окну присваивается автоматически из числа наименьшего свободного номера.
 - Last** – Окну присваивается номер следующий за наибольшим номером окна, уже существующего в проекте.
 - User-defined** – Номер окну присваивает пользователь.
 - Insert window serial num** - Все последующие номера окон за присвоенным этому окну будут увеличены на единицу.
- Поле задания номера вручную для данного типа окна.
- Поле выбора типа создаваемого окна. После сохранения окна, оно будет расположено в соответствующем разделе Менеджера окон.
- Поле задания размера окна в пикселях.
- Настройка безопасности. Поле выбора необходимого уровня доступа для пользователя к элементам этого окна. Активируйте опцию *Switch to user level when window closed* (При закрытии окна установить уровень доступа), если необходимо сменить текущий уровень доступа при закрытии окна, и выбрать уровень, который будет активирован.
- Основное окно может быть выведено на экран в виде *Всплывающего окна (Popup Window)* в заданными пользователем позиции экрана (*Показать в.. {Show At..}*), своим заголовком (*Показать заголовок {Show Title}*) и ограниченной этим всплывающим окном активной областью сенсора (*Монополизировать {Monopolize}*).
- Опция *Наложения окон {Overlapped Window}* (до трёх вспомогательных окон в своём слое поверх создаваемого Основного окна). Данная опция полезна к применению, если в нескольких окнах используются идентичные элементы управления или стили оформления.
- Номер, который будет присвоен окну при их переключении {Window number used for window switching}.
- Визуальные эффекты {Window Effect}* – *Плавное появление окна {Fade in}*, *Плавное исчезновение {Fade out}*. Применимы только для ёмкостных типов экранов панелей оператора.

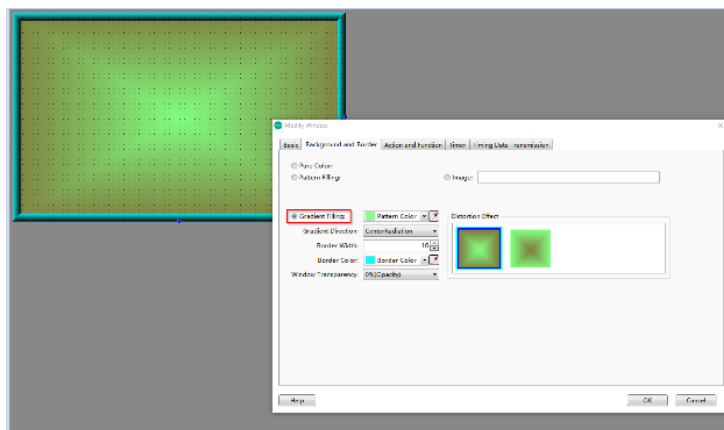
✓ **Оформление {Background and Border}.**

Вкладка предназначена для оформления внешнего вида окна. Для придания вашему окну желаемого вида доступны следующие инструменты:



- **Базовый цвет {Pure Color}** – Выбор основного тона для заливки области окна.
- **Заполнение текстурой {Pattern Filling}** – Выбор рисунка текстуры её цвета и цвета фона на котором она будет отображаться.
- **Изображение {Image}** – Позволяет поместить в окно требуемое Вам графическое изображение или фото из файла графики различных типов (.bmp, .jpeg, .dib, .rle, .jpg, .jpe, .jfif, .gif, .emf, .tiff, .png и др.)
 - Изображение можно выбрать из файла или из Библиотеки графики.
- **Заливка градиентом {Gradient Filling}** – Выбор цвета и направления градиентной заливки объекта.
- **Ширина границы (рамки) {Border Width}**
- **Цвет границы (рамки) {Border Color}**

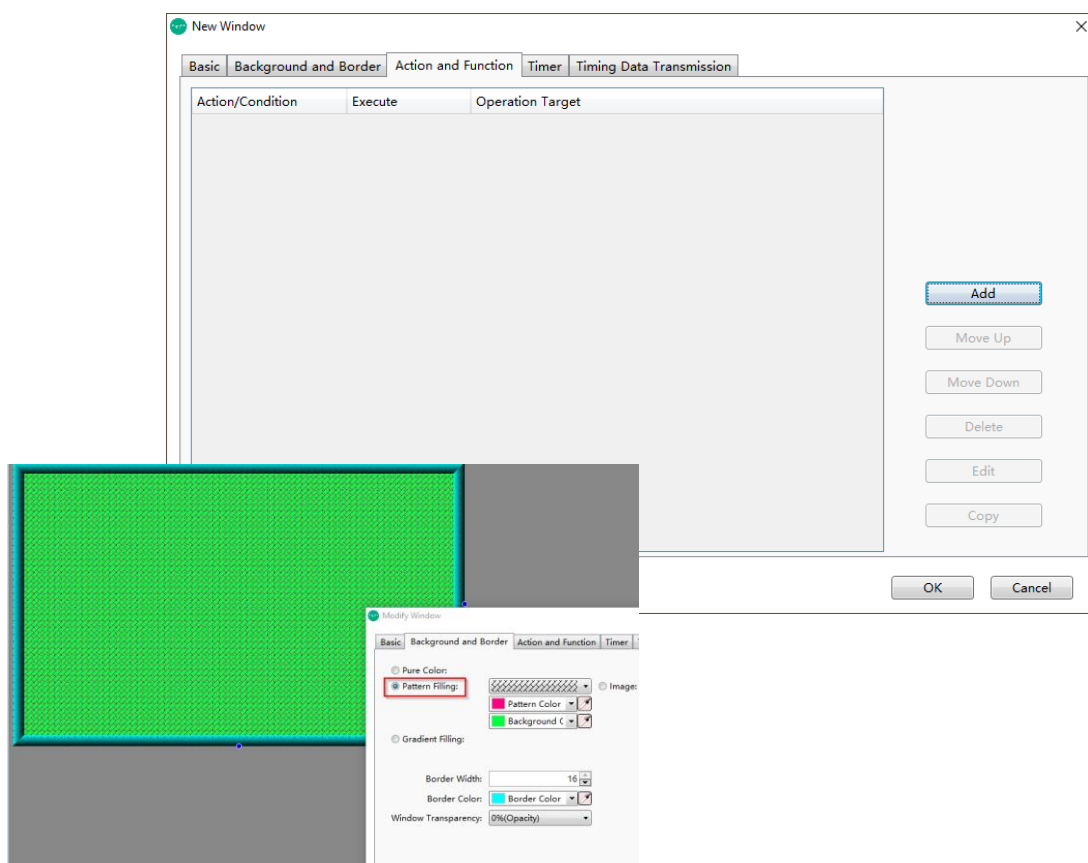
- **Прозрачность окна {Window Transparency}.**



✓ **Действия и функции {Action and Function}.**

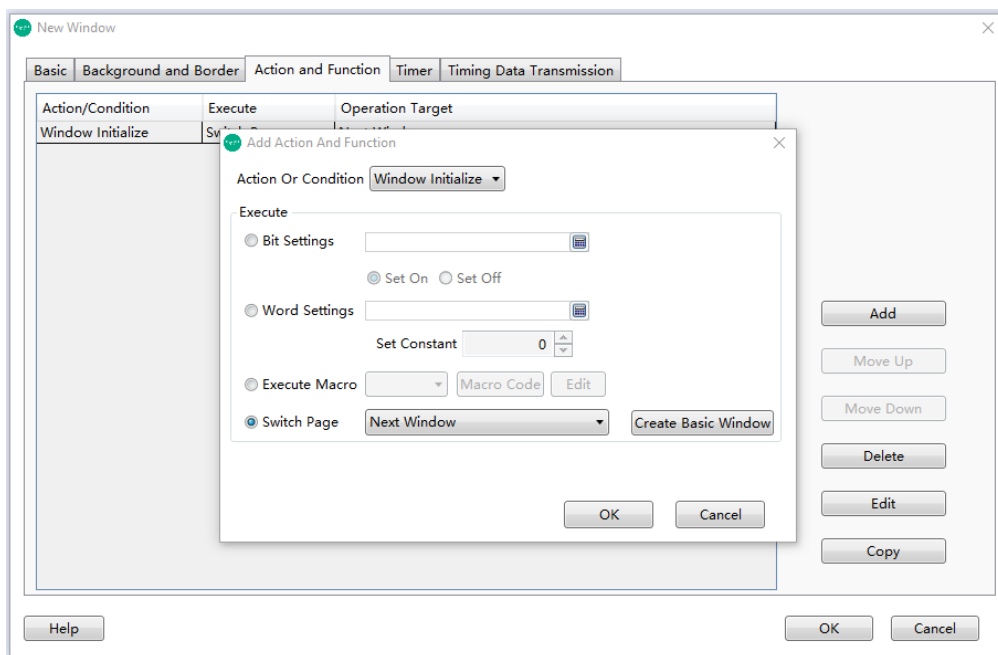
Вкладка предназначена для задания действий, которые должны быть выполнены при открытии или закрытии окна в процессе выполнения проекта.

Действие или функцию можно назначить нажатием на кнопку *Добавить {Add}*.



В открывшемся диалоге «Действие или Функцию {Action and Function}» можно задать:

- Как **Событие или Условие {Action Or Condition}** для окна, при наступлении которого, будет выполнено указанные ниже действия:



Bit- изменить {Bit Settings}. Изменить указанную Битовую переменную (LBx) или Бита по номеру в регистре Word (LWxx.y). Где "xx"- номер регистра, а "y"- номер бита в этом регистре.

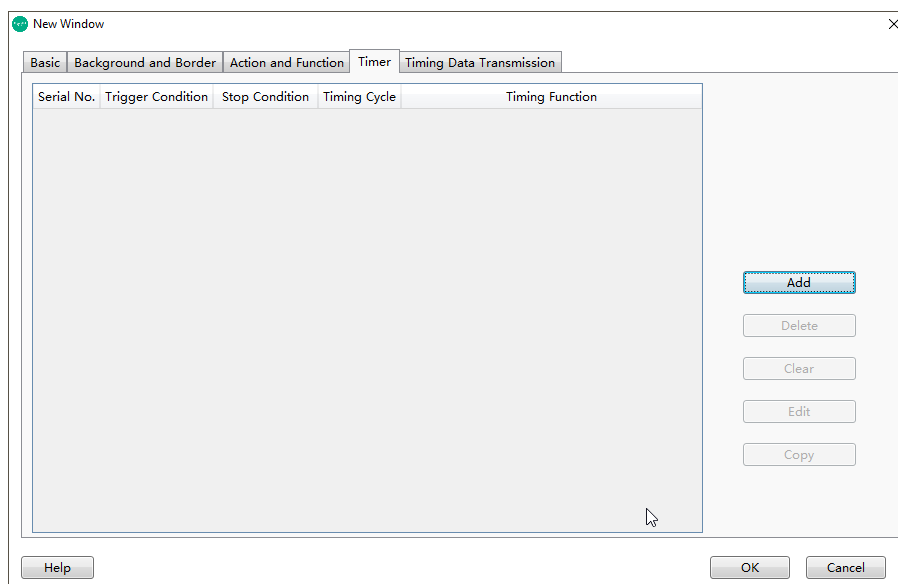
Word- изменить {Word Settings}. Изменить значения слова данных Word (задать константу).

Выполнить макрос {Execute Macro} - Выполнить макрос из числа настроенных в проекте.

Переключить окно {Switch Page} - Открыть заданное окно проекта

✓ Таймеры

Вкладка предназначена для создания и настройки таймеров, по условию срабатывания которых может выполняться определенное действие.

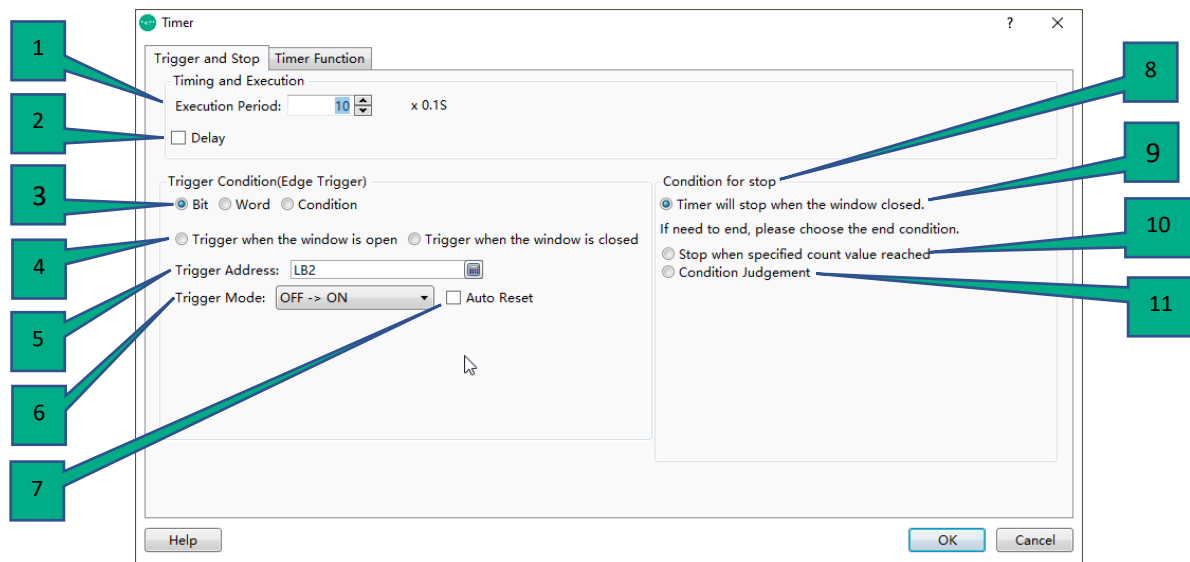


Кнопкой **Добавить {Add}** создаём новый таймер и переходим к заданию времени его работы и условий запуска.

✓ Запуск и остановка таймера.

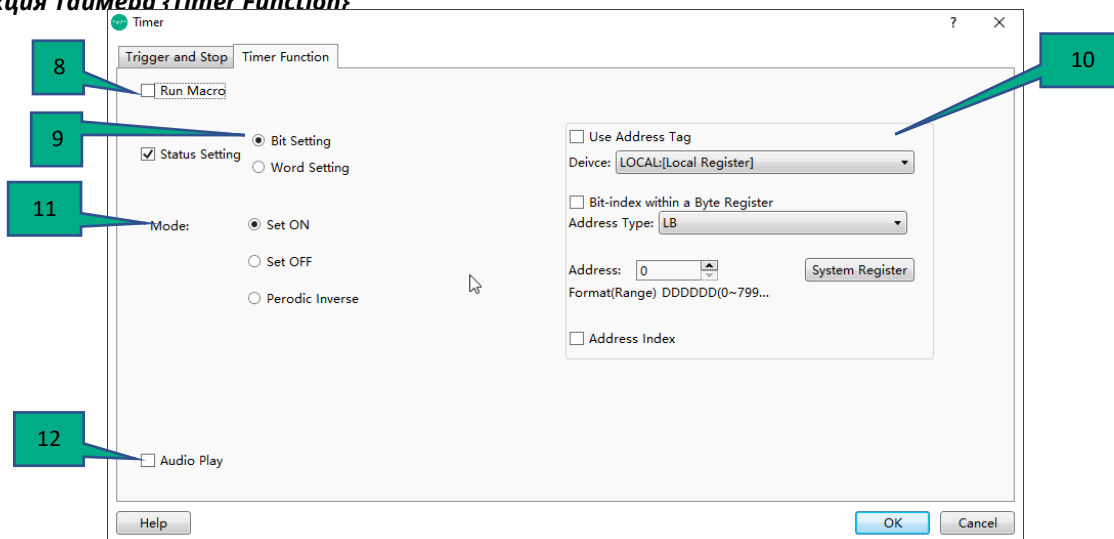
Область **Timing and Execution** Задание периода времени и выполнение запуска таймеров.

1. **Период: {Execution Period}** -Время работы таймера.
2. **Задержка {Delay}** - Задержка начала работы таймера с момента наступления события определенного, как условие для запуска.

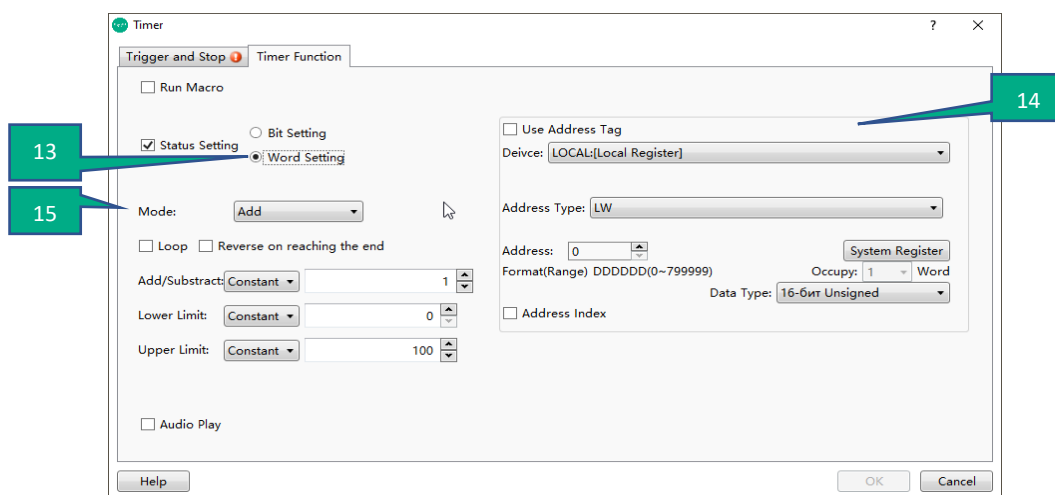


3. Область **Условие вызова {Trigger Condition}** – Запуск таймера по фронту изменения {Edge Trigger} битового сигнала {Bit}, содержимого регистра {Word}, Условие {Condition} -Задание логической цепи связанных между собой условий.
4. Условие запуска таймера.
Вызывать, если окно открыто {Trigger when the window is open};
Вызывать, если окно закрыто {Trigger when the window is closed};
5. **Адрес {Trigger Address}** – Адрес переменной, по которой происходит запуск (вызов) таймера.
6. **Режим вызова {Trigger Mode}** для битовой переменной – При каком изменении значения битовой переменной производить запуск таймера.
 - OFF->ON - по переднему фронту изменения состояния переменной (0 -> 1)
 - ON->OFF - по заднему фронту изменения состояния переменной (1 -> 0)
 - ON<-> OFF – по любому из фронтов изменения состояния переменной (0 -> 1 или 1 -> 0)
7. **Авто сброс {Auto Reset}** -Автоматический сброс таймера.

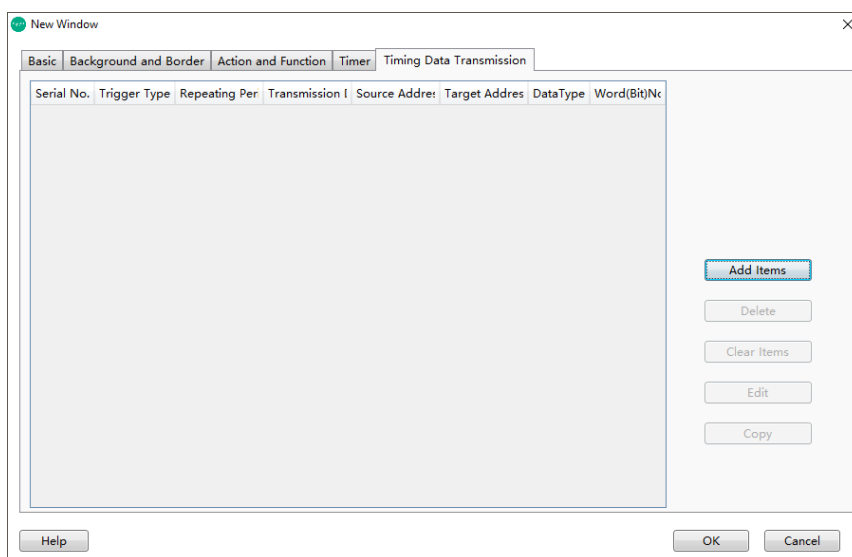
✓ Функция Таймера {Timer Function}



8. **Запустить Макрос {Run Macro}** – если опция выбрана запускает выбранный Макрос при активации таймера.
9. **Изменить переменную {Status Setting}** – Выбор типа переменной, которая будет менять своё состояние при срабатывании таймера. **Bit-изменить {Bit Setting}** – выбор битовой переменной или **Word - настройку {Word setting}** выбор для изменения переменной Word см. ниже.
10. Область задания адреса и параметров для выбранной битовой переменной:
 - область хранения выбранной переменной (внутренняя память панели (LOCAL) или удаленное устройство (Device *))
 - её метки (опция **Использовать адресную метку {Use Address Tag}**)
 - типа битовой переменной (LB или SRB)
 - адрес битовой переменной **Адрес {Address}** или номера конкретного бита по номеру в указанном регистре **{Bit-index within a Byte Register}**.
 - адрес для индексной переменной **Индекс {Address Index}** (если используется).
11. Выбор типа операции с выбранной битовой переменной **Режим {Mode}**:
 - **Установить в «1» {Set ON}** (включить)
 - **Установить в «0» {Set OFF}** (отключить)
 - **Инвертировать {Periodic Inverse}** (переключить)
12. Название аудио {Audio Play} – выбор аудио файла из соответствующей Библиотеки для проигрывания файла по таймеру. (Для панелей с audio выходом).



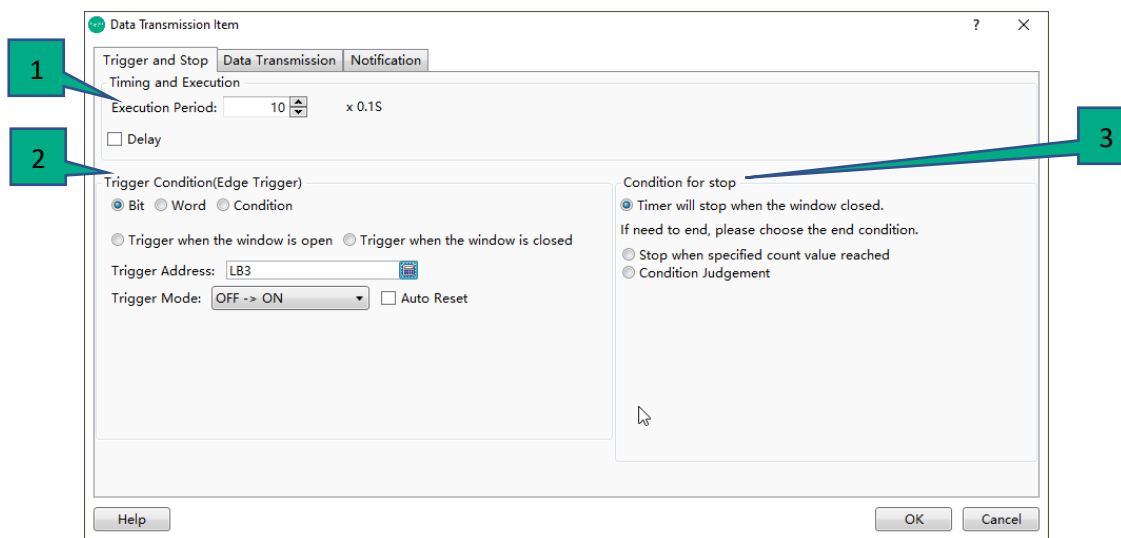
13. **Изменить переменную {Status Setting}** – Выбор типа переменной, которая будет менять своё состояние при срабатывании таймера. **Word - настройку {Word setting}**.
 14. Область задания адреса и параметров выбранной переменной Word.
 15. Изменение переменной Word по таймеру дополнены такими режимами, как Сложить {ADD}, Вычесть {Subtract} и Константа {Constant}.
- Опции {Run Macro}, {Status Setting} и {Audio Play} могут быть выбраны одновременно. Таймер способен контролировать выполнение всех трёх опций. Однако порядок их выполнения не определён.



✓ **Синхронная передача данных {Timing Data Transmission}.**

Отдельные значения переменных или пакет данных могут быть переданы по времени. Действие может запускаться или выполняться периодически. Этот вкладка аналогична вкладке **Таймера {Timer}**. Вам можно добавить этот компонент в настройки создаваемого окна. Если вы хотите глобальное выполнение, вы можете добавить его в Фоновое окно {Public Window}.

• **Запуск и остановка {Trigger and Stop}**



Эта вкладка используется для управления режимами выполнения компонентов. Режимы «Запуск и остановка» включают

1. «Период выполнения» {Execution Period},

2. «Условие запуска» {Trigger Condition}

3. «Условие остановки» {Condition for stop}.

- **Период выполнения { Execution Period}** и задержка начала работы **{Delay}** исчисляются с момента наступления события определенного, как условие для запуска.

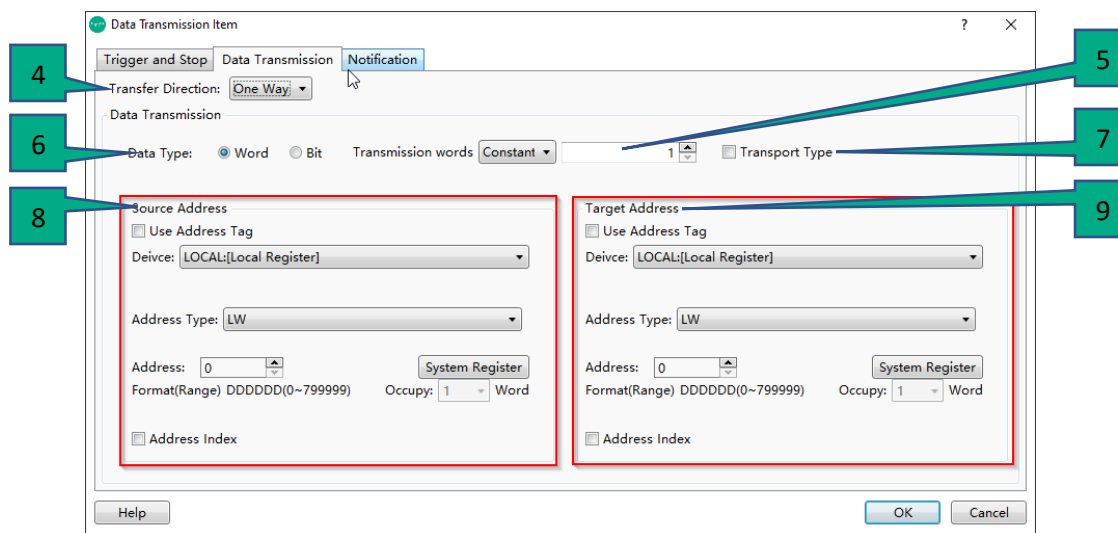
- **«Условие запуска» {Trigger Condition}** - поле настройки условия начала передачи данных. Может быть выбрано состояние бита, значение слова или логическое условие. Также в качестве условия выполнения таймера может быть указано состояние окна: открыто или закрыто.

- **«Условие остановки» {Condition for stop}** - поле настройки условия завершения передачи данных. В качестве условия остановки передачи данных может быть выбрано событие закрытия окна,

Остановить при достижении значения {Stop when specified count value reached} по условию выполнения заданного количества циклов с момента запуска, либо логическое или арифметическое условие

• **Передача данных {Data Transmission}**

На этой вкладке задаем тип и объем передаваемых данных, а также адреса Источника {Source} и Приемника {Target} этих данных.



4. **Направление передачи {Transfer Direction}** – Односторонняя, Двухсторонняя.

5. **Количество {Transmission words}** – Слов или бит задаваемое константой или переменной.

6. **Тип передаваемых данных {Data Type}** – Word (16 битное слово) или Bit (бит).

7. **Формат данных {Transport Type} для Word** - 16-бит Integer, 32-бит Integer, 32-бит Float.

8. **Адрес источника данных {Source Address}**.

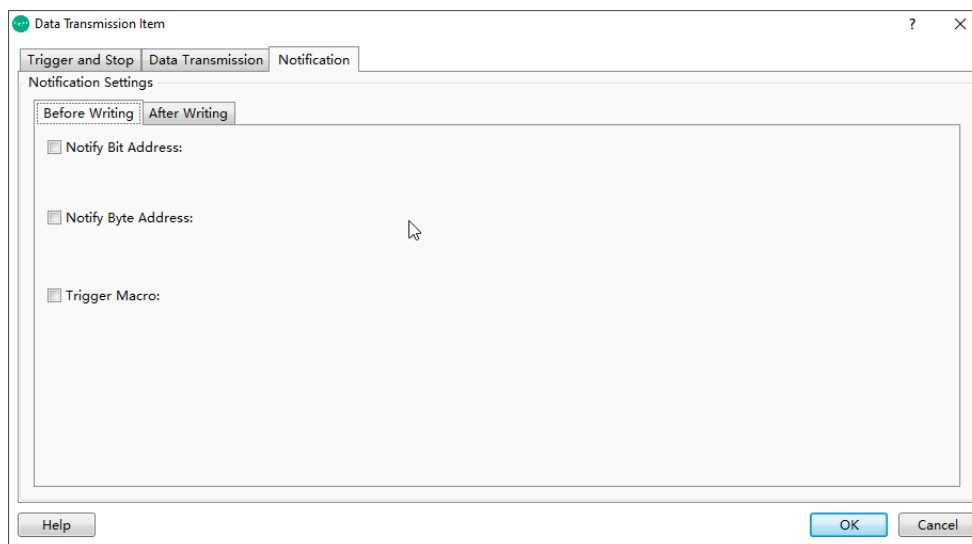
9. **Адрес приемника данных {Target Address}**.

В случае передачи нескольких Слов или Бит, в поле **Адрес** задается начальный адрес блока данных в памяти, адреса всех остальных переменных рассчитываются автоматически по порядку с увеличением на 1.

- **Уведомления {Notification}.**

На этой вкладке предоставляется возможность настроить уведомления о начале {Before Writing} или завершении {After Writing} процесса передачи (записи в память) данных.

Уведомления, как До записи данных, так и После записи могут быть переданы:



- Изменением состояния бита (Записать «1» {Write On} или Записать «0» {Write Off}. Для этого выбираем опцию

Объявить адрес бита {Notify Bit Address}.

- Записью требуемого предустановленного значения по требуемому адресу переменной. Для этого выбираем опцию **Объявить адрес Байта {Notify Byte Address}**.
- Запуском макроса {Trigger Macro}.

Для завершения настроек нажмите кнопку «ОК».

Как создать новый проект? Выбор модели панели оператора

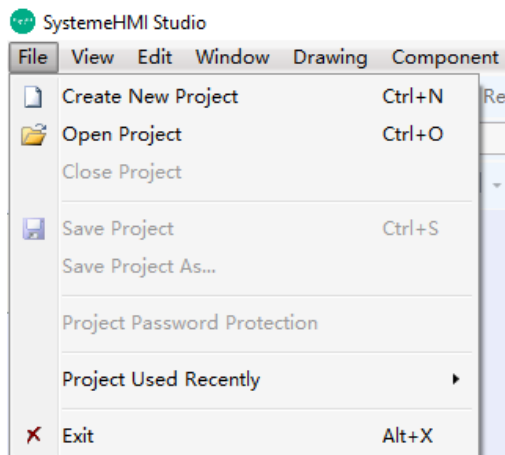
Версия ПО

Версия SystemeHMI Studio: 3.0.12512.0

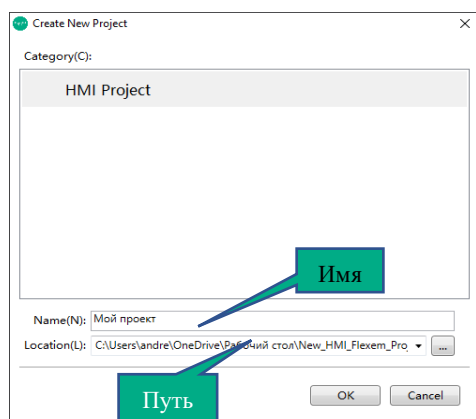
Версия ОС Windows: Windows 10 Pro, 21H2, сборка 19044.2604

Для создания нового проекта необходимо:

1. Выбрать закладку **File > Create New Project** (горячая клавиша CTRL+N).



2. В открывшемся окне ввести имя вашего будущего проекта "**Name(N)**" и путь к папке, в которой он будет сохранён "**Location(L)**".



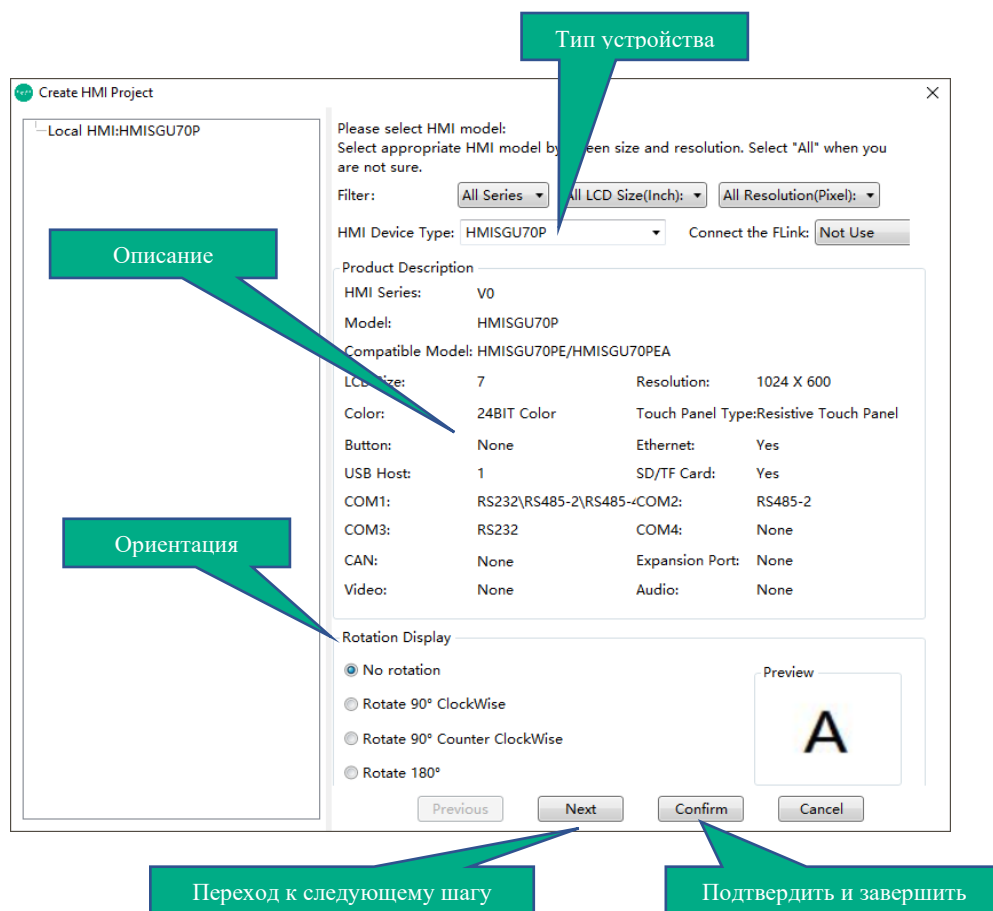
3. Выбрать тип используемой панели из выпадающего списка доступного оборудования, при этом в поле "**Product Description**" отобразится краткое техническое описание модели. При необходимости можно изменить ориентацию экрана панели.

Необходимо учесть, что панели оператора серии: HMISGU70PEA, HMISGU101MEA, HMISGU156ME являются усовершенствованными (Advanced) и поддерживают функцию **Label Based Access Control (LBAC)**, включающей в себя:

- Прямой импорт тегов ПЛК
- Анализ структуры тегов ПЛК (массив, структура и т.д.)
- Возможность использование массивов текстовых значений тэгов (Array-type Tags) для рецептов
- Настройка отчетов в соответствии с выбранным форматом с помощью макрокоманды
- Передача по протоколу FTP сигналов тревоги, записей операций, пользовательских отчетов
- Перенос (импорт/экспорт) рецептов между проектами
- Быстрое копирование/замена/редактирование рецептов
-

Эта функция поддерживается в ПЛК серии SystemePLC S250 от Systeme Electric, и поэтому при работе с указанными выше панелями оператора, можно в качестве типа подключенного к панели контроллера выбирать в настройках соединения по Ethernet порту именно **Manufacturer > Systeme Electric** и **Device Type > SystemePLC-S250**.

Во всех остальных случаях рекомендуем, в качестве типа подключенного к панели ПЛК, использовать **Manufacturer > Modbus Compatible** с присущей спецификации Modbus адресацией регистров.

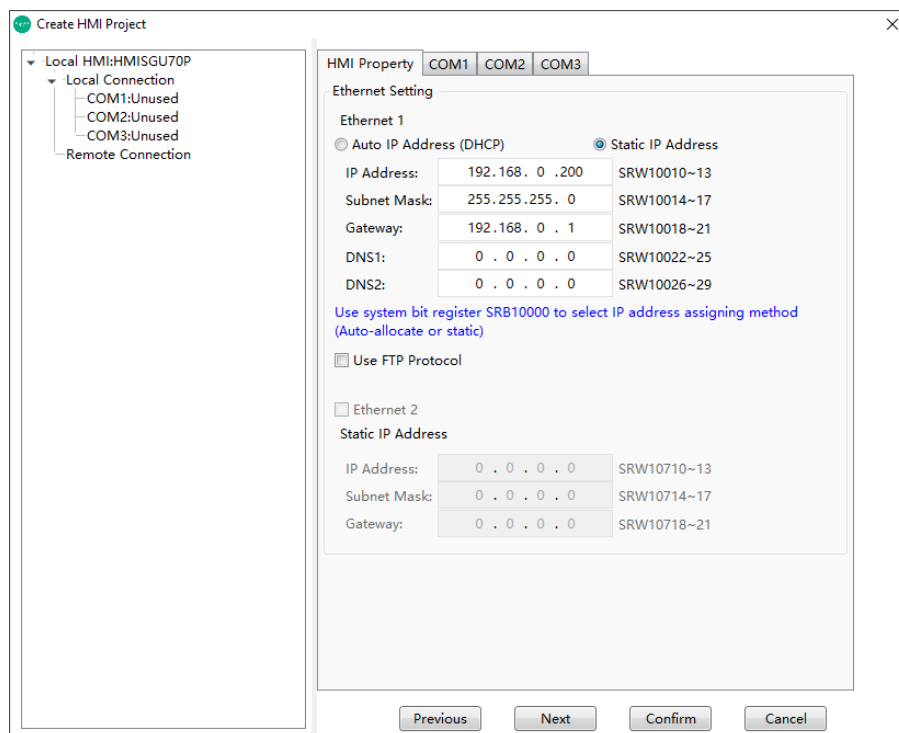


Кнопка «**Next**» служит для перехода к следующему шагу

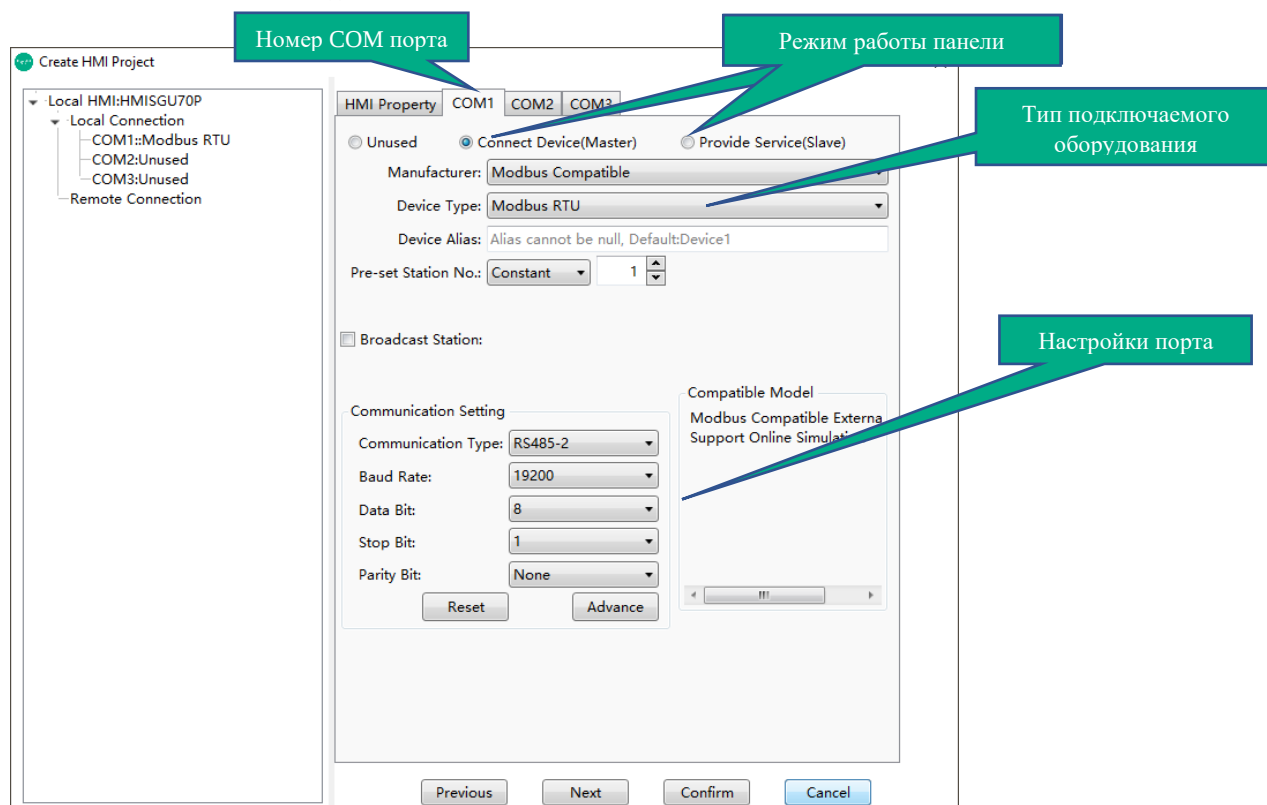
Кнопка «**Confirm**» позволяет подтвердить сделанный выбор и прервать диалог создания нового проекта. В последнем случае вернуться к настройкам можно будет в разделе главного меню "**Setup > HMI settings**" или через соответствующие пункты в окне проекта.

4. Для перехода к следующему шагу нажмите кнопку «**Next**».

На вкладке «**HMI Property**» нового экрана в разделе «**Ethernet Setting**» можно задать IP адрес панели оператора, маску подсети, IP адрес шлюза и DNS серверов (при необходимости). Выбрать способ назначения IP адреса для панели (Статический или присвоенный DHCP сервером автоматически).

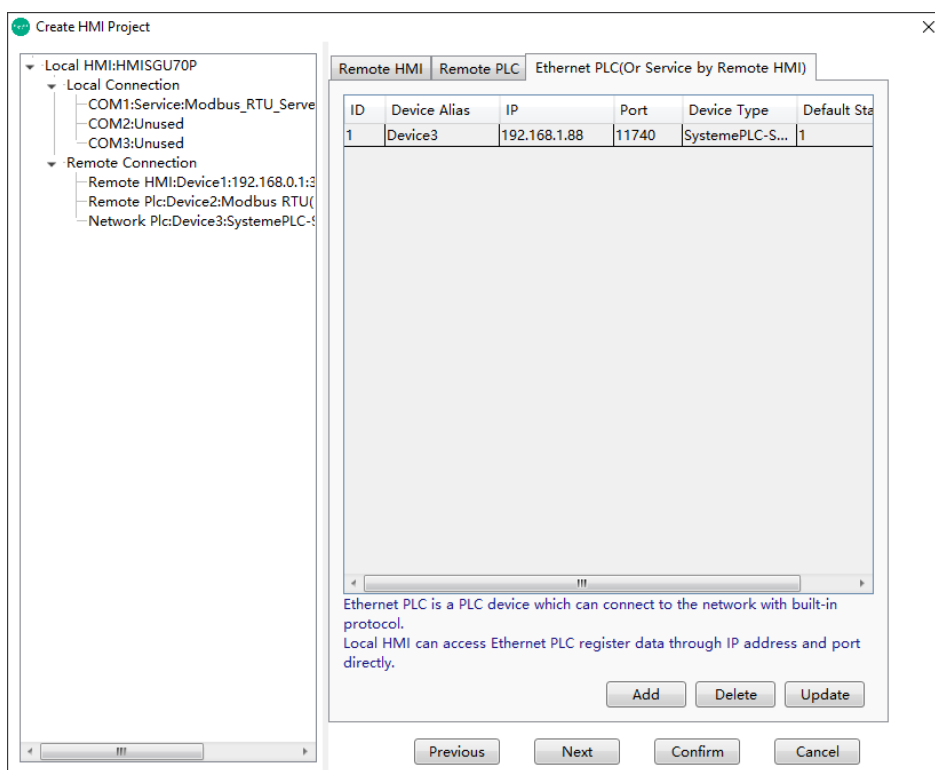


5. Вкладки COM1-COM4 используются для задания параметров связи по последовательным портам. Здесь определяется режим работы панели (Master-Slave), производитель и тип подключаемого оборудования-контроллера, а также его адрес по умолчанию. В нижней части окна задаются параметры порта: скорость передачи данных, количество бит данных, количество стоповых бит и т.д.

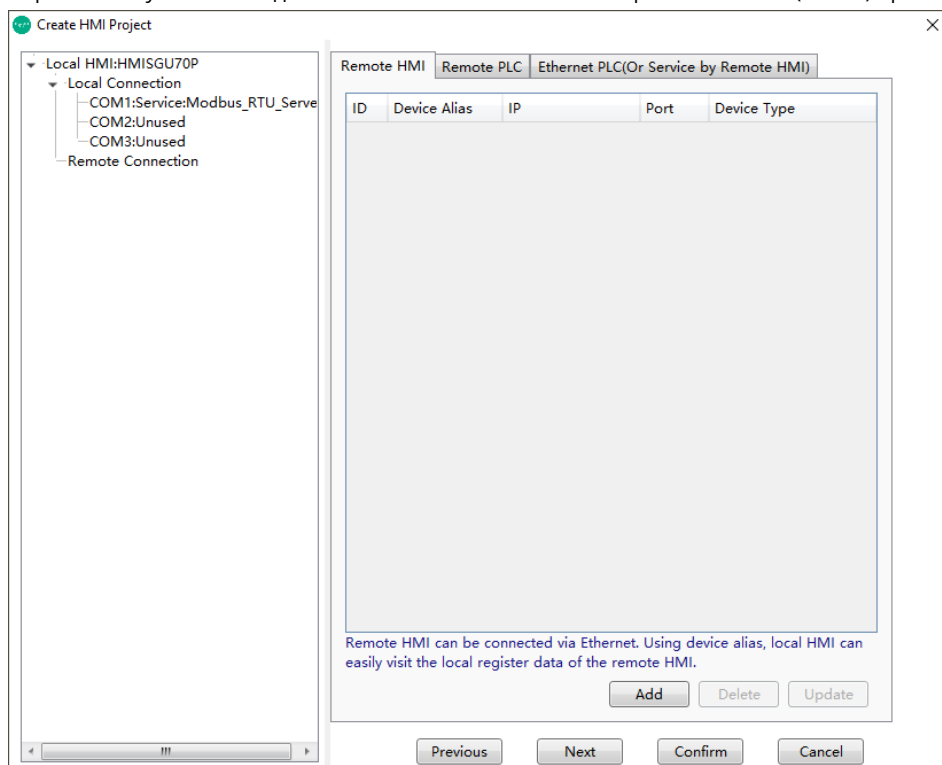


6. Переход на следующий шаг (кнопка NEXT) позволяет настроить подключение панели с помощью сетей Ethernet.
- Вкладка **"Remote HMI"** служит для настройки подключения к удаленной панели оператора, что позволяет использовать регистры хранения данных удаленной панели в локальном проекте.

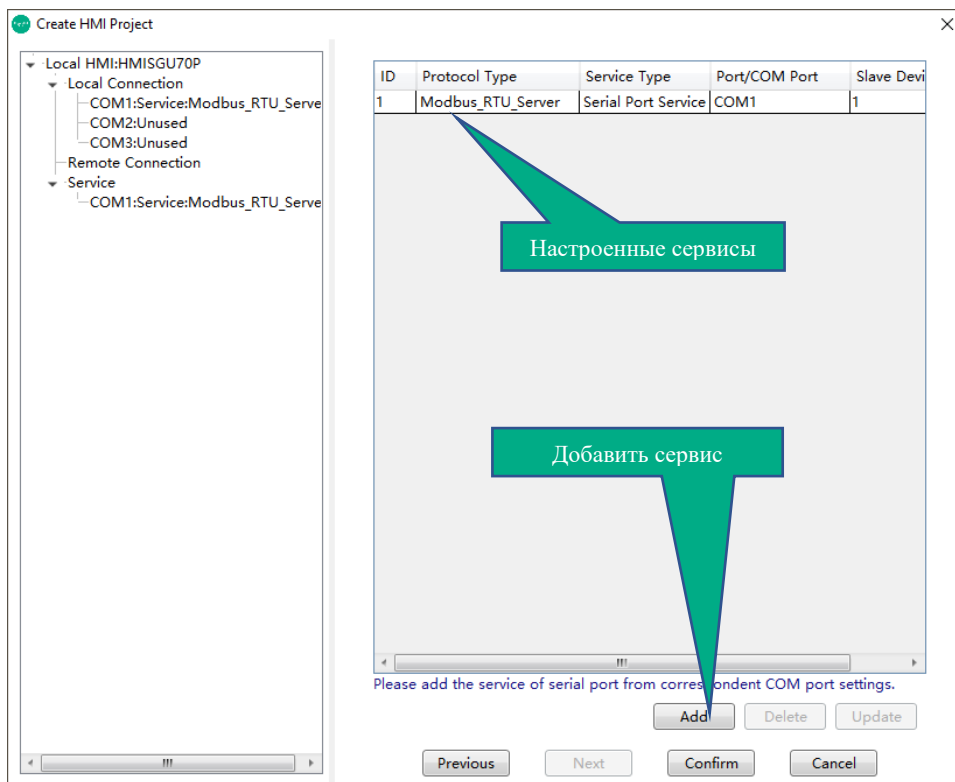
- Вкладка "**Remote PLC**" позволяет настроить подключение к удаленному ПЛК через COM порт удаленной панели оператора (Remote HMI), которая, в данном случае, выполняет роль шлюза. При таком подключении в локальном проекте появляется возможность работать напрямую с регистрами памяти удаленного ПЛК.
- Третья вкладка "**Ethernet PLC or Service by Remote HMI**" позволяет настроить прямое подключение к ПЛК, в случае если ПЛК имеет встроенный порт Ethernet и объединен в одну сеть с панелью оператора.



- Последнее окно диалога создания проекта позволяет просмотреть все порты, для которых выбрана роль "Slave", а также используемые протоколы и предоставляемые ресурсы. Также на данном этапе можно настроить дополнительные сервисы в случае необходимости использования панели в режиме "Slave" (Server) при подключении через Ethernet.



8. Нажмите кнопку «Confirm» для завершения диалога конфигурирования параметров панели оператора и открытия главного окна проекта. Все готово для начала создания вашего проекта, который в последствии будет загружен в указанную Вами панель и успешно работать.



Как создать основные элементы интерфейса? Кнопки, переключатели, индикаторы, числовые или символьные поля ввода, числовые или символьные дисплеи

Версия ПО

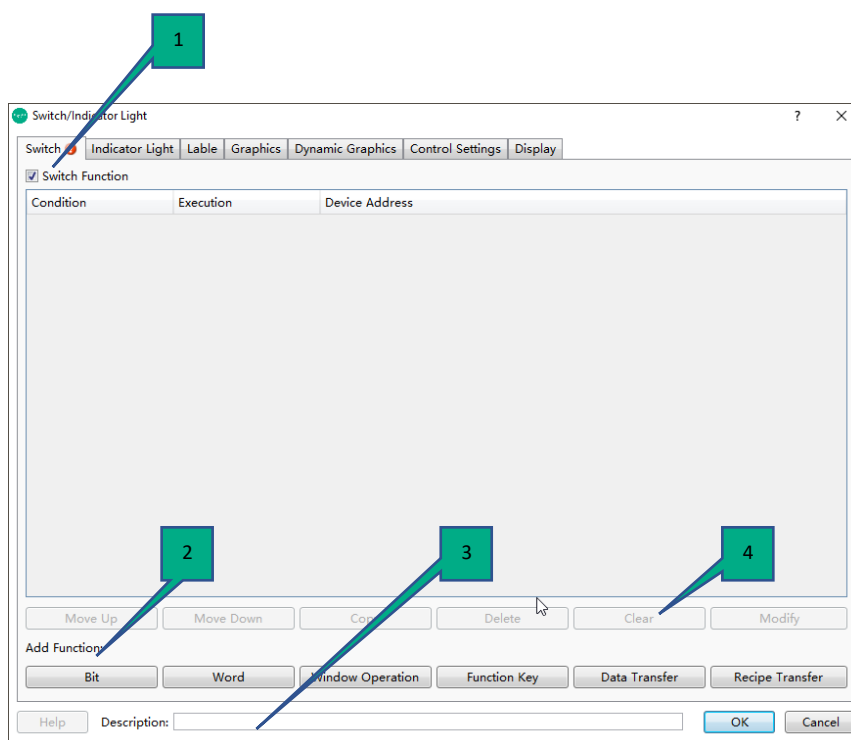
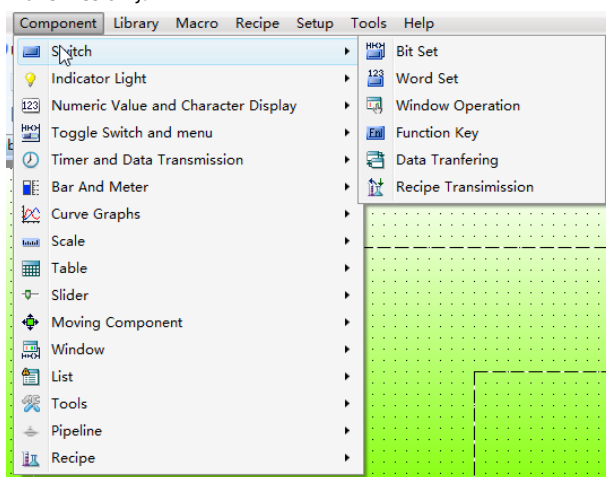
Версия SystemeHMI Studio: 3.0.12512.0

Версия ОС Windows: Windows 10 Pro, 21H2, сборка 19044.2604

✓ Настройка Графических элементов {Component}

Клавиша (Кнопка, Переключатель) {Switch} – это один из компонентов который пользователь может поместить в создаваемую область окна. Компонент Switch включает в себя следующие возможные действия:

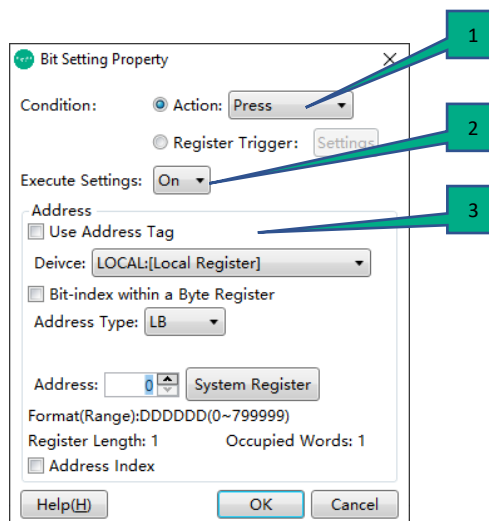
- Установка бита (Bit - изменить {Bit Set}),
- Установка, задание слова (Word - изменить { Word Set }),
- Операции с окнами ({Window Operation},
- Функции {Function Key }, -
- Передачу данных {Data Transferring }
- Передачу рецепта {Recipe Transmission }.



1. Активация/Деактивация функции Клавиши {Switch}
2. Поле **Добавить функцию {Add Function}**. Назначение элементу новой функции или нескольких функций, если окно добавления не открылось автоматически.
3. **Описание {Description}**. Задание текстового описания каждому элементу, для удобства навигации в проекте.
4. Поле управления порядком выполнения функций, если их задано несколько для этого элемента. Элементы управления списком функций активируются при выделении одной из них.

✓ **Функции {Add Function}**

✓ **Bit – изменить {Bit Setting}**



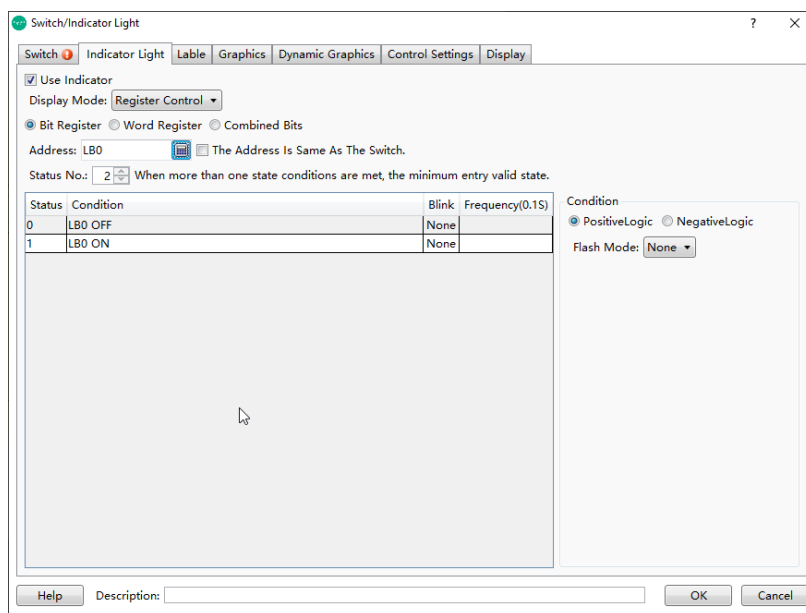
«Bit Setting» — это компонент, с помощью которого происходит управление внутренним битовым адресом HMI или битовым адресом любого контроллера, подключенного к панели HMI.

1. **Условие {Condition}: Событие, {Action}**, по условию наступления которого произойдет выбранное действие.
 - **Нажать {Press}** – Действие произойдет при нажатии на экранный элемент (например кнопку)
 - **Отпустить {Release}** – Действие произойдет при прекращении нажатия.
 - **Register Trigger** – Действие произойдет при любом изменении содержимого указанного регистра или выполнении заданного логического условия.
2. **Выполнить {Execute Setting}**. Указанное действие будет выполнено при наступлении условия из п.1.
 - **Установить [1] {On}** - Установить заданный бит в состояние логической единицы.
 - **Установить [0] {Off}** - Установить заданный бит в состояние логического нуля.
 - **Инвертировать {Inverse}** – Инвертировать состояние указанного бита.
 - **Сброс {Reset}** - Присвоить указанному биту значение логического нуля.
 - **Импульс [1] {On Pulse}** – Установить бит в состояние логической единицы на указанное время **{Pulse Width} * 0,1** в секундах.
 - **Импульс [0] {Off Pulse}** - Установить бит в состояние логического нуля на указанное время **{Pulse Width} * 0,1** в секундах.
3. Адрес выбранного битового регистра (Принадлежность регистра к устройству, Тип адреса, Адрес)

✓ **Индикатор {Indicator Light}**

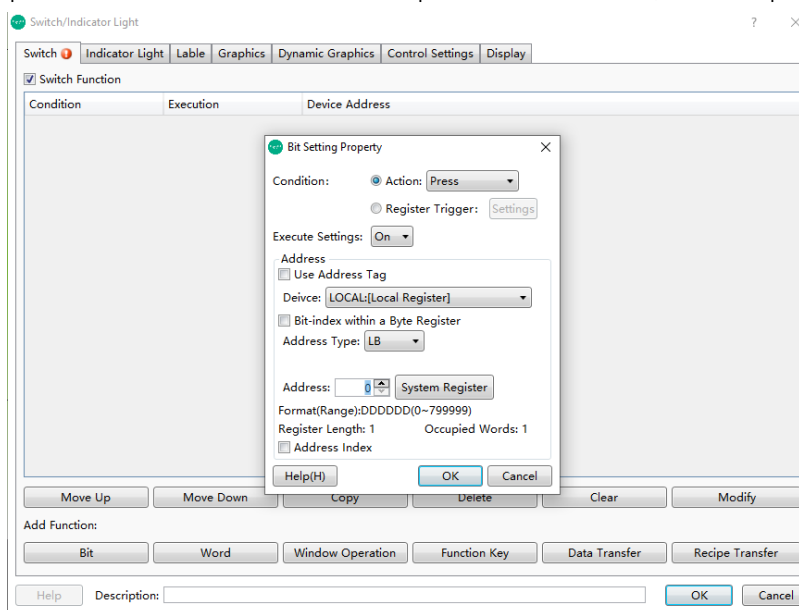
Компонент «Bit Setting» имеет и другие страницы свойств. Например, на странице свойств **Индикатор {Indicator Light}** мы рекомендуем установить флажок «Индикатор» {Use Indicator} и задать адрес переменной для возможности наблюдения за состоянием компонента «Bit Setting», если на странице свойств «Оформление» {Graphics} установлен флажок «Использовать графику» {Use Graphics}.

В следующем кратком примере показан процесс установки в HMI внутреннего адреса LB100 с атрибутом «Инвертировать»



{Inverse}.

В меню программного обеспечения нажмите «Component/ Switch/ Bit Set», чтобы открыть следующий диалог.



Действие по умолчанию — «Нажать» {Press}. Настройка выполнения по умолчанию «Вкл.» {On}. Адрес по умолчанию — «LB0». Мы преобразуем их в требуемые следующим образом:

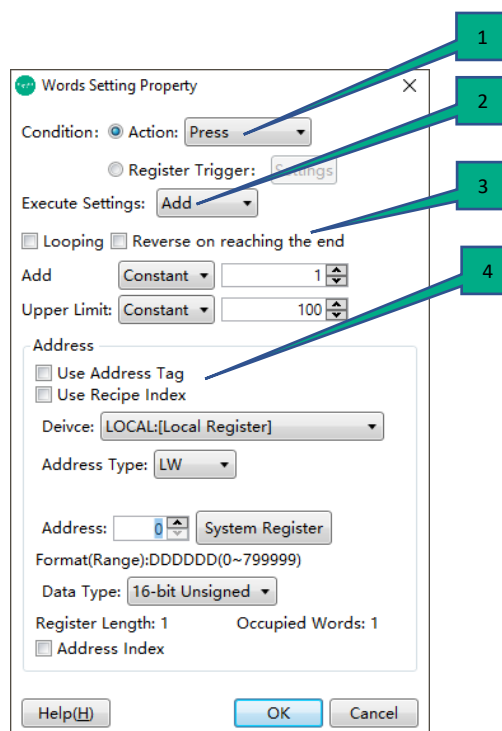
Нажмите «OK», чтобы добавить действие операции в список «Переключение» {Switch}.

Condition	Execution	Device Address
Action:Press	Bit Setting	LOCAL:[Local Register]:LB100 Inverse

В списке мы можем ясно видеть атрибут выполнения (Inverse) и адрес бита (LB100) с которым нужно работать. Нажмите «OK» в диалоговом окне «Переключение» {Switch}, чтобы поместить компонент в окно панели HMI, которое сейчас редактируем. Таким образом, простой компонент «Bit Setting» готов.

✓ **Word- изменить {Word Setting}**

Функция изменяет значение слова данных при нажатии или отпуске сенсора экрана в области графического элемента (например изображения кнопки на редактируемом окне).



1. **Условие {Condition}: Событие, {Action}**, по условию наступления которого произойдет выбранное действие.
- **Нажать {Press}** – Действие произойдет при нажатии на экранный элемент (например изображение кнопки)
- **Отпустить {Release}** – Действие произойдет при прекращении нажатия.
- **Register Trigger** – Действие произойдет при любом изменении содержимого указанного регистра или выполнении заданного логического условия.
2. **Выполнить {Execute Setting}**. Указанное действие (с заданными для каждой операции атрибутами) будет выполнено при наступлении условия из п.1.
3. Параметры необходимые для выбранного действия.
4. Адрес регистра типа Word, с которым будут производиться указанные действия.

Доступны следующие варианты действий **{Execute Setting}**:

- **Сложить {Add}** - выполнить операцию сложения
Назначенные данные будут добавлены к значению назначенного регистра Word.
Действие «Добавить» содержит следующие параметры:
 - **Добавить {Add}** – в этом параметре указывается какое значение необходимо добавить к заданному регистру.
При каждом действии, указанном в «Action», установленные данные в параметре «Add» будут добавляться и записываться в регистр слов.
 - **Верхний предел {Upper Limit}**
Это верхний предел используемого регистра слов. Когда результат достигает верхнего предела операции, дальнейшая операция «Добавить» выполняться уже не будет.
 - **Цикл {Looping}**
Если установлен флажок «Цикл», появится параметр «Нижний предел» {Lower Limit}. Операция будет продолжаться до значения «Верхний предел», а повториться со значения указанного в «Нижнем пределе». Например, если параметр «Нижний предел» установлен на «0», параметр «Добавить» установлен на «1», а «Верхний предел» установлен на «100», то компонент «Word setting» (назначенный регистр Word) будет действительным, когда он достигнет значения «100». При следующих операциях содержимое назначенного регистра Word примет значение 0, 1, 2, 3 и т. д.
 - **Реверс по достижении конца {Reverse on reaching the end}**
Если установлен флажок «Реверс при достижении конца», операция будет продолжена на верхнем пределе, а результат изменится на уменьшение от верхнего предела к нижнему пределу. Когда результат достигнет нижнего предела, операция изменится на сложение.

«Добавить», «Верхний предел» и «Нижний предел» по умолчанию настроены на работу с константами {Constant}. Их также можно установить и на работу с переменными {Variable}. Обратите внимание, что тип данных регистра «Variable» должен соответствовать типу адреса компонента «Word Setting».

- **Вычитать {Subtract}** - выполнить операцию вычитания
Назначенные данные будут вычитаться из значения регистра назначенного слова до указанного «Нижнего предела» {Lower Limit}.

Атрибут «Вычитать» содержит следующие параметры:

Words Setting Property

Condition: ☒ Action: Press

☐ Register Trigger: Settings

Execute Settings: Subtract

☐ Looping ☐ Reverse on reaching the end

Subtract: Constant 1

Lower Limit: Constant 0

Address

☐ Use Address Tag

☐ Use Recipe Index

Device: LOCAL:[Local Register]

Address Type: LW

Address: 0 System Register

Format(Range): DDDDDD(0~799999)

Data Type: 16-bit Unsigned

Register Length: 1 Occupied Words: 1

☐ Address Index

Help(H) OK Cancel

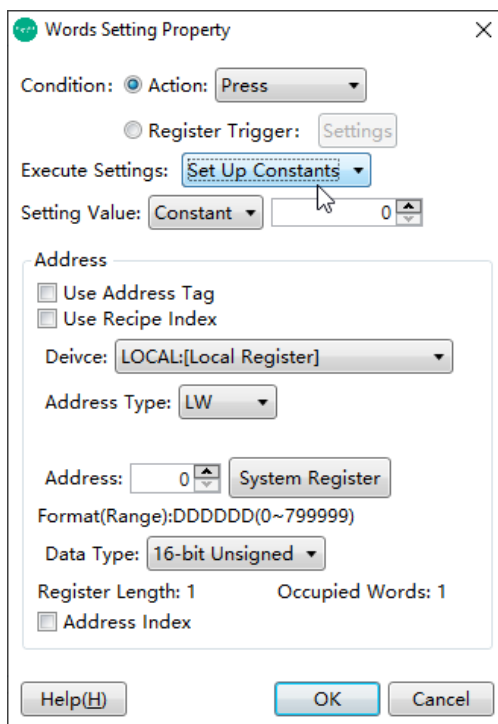
- **Вычитать {Subtract}**
При каждом действии, определённом в «Action», данные, определяемые в параметре «Subtract», будут вычитаться из регистра слов.
- **Нижний предел {Lower Limit}**
Это нижний предел используемого регистра слов. Когда результат достигает нижнего предела операции, дальнейшая операция «Вычитание» не может быть выполнена.
- **Цикл {Looping}**
Если установлен флажок «Цикл», появится опция «Верхний предел». Операция будет продолжена на нижнем пределе, а вычитание будет производиться из верхнего предела. Например, если «Нижний предел» установлен на «0», «Вычитание» установлен на «1», а верхний предел установлен на «100», компонент набора слов будет действительным, если его вычитать до «0». При следующих операциях он превратится в 100, 99, 98, 97 и так далее.
- **Реверс по достижении конца {Reverse on reaching the end}**
Если установлен флажок «Обратный ход при достижении конца», операция будет продолжена на нижнем пределе, а результат изменится на добавление от нижнего предела к верхнему пределу. Когда результат достигнет верхнего предела, операция изменится на вычитание.
- **Увеличить {Increase}** - циклически увеличивать значение в регистре с заданным интервалом
Значение результата будет увеличиваться, если на графический элемент (компонент) на экране с настройкой «Увеличить» будет осуществляться нажатие. Увеличение прекратится, если нажатие на графический элемент будет прекращено или значение достигнет верхнего предела «Upper Limit».

Кроме описанных выше атрибутов "Looping", "Reverse on reaching the end", "Add" и "Upper Limit" компонент с параметром «Увеличение» имеет атрибуты «Выполнить без задержки», «Время задержки» и «Время выполнения».

- Выполнить без задержки {Immediately Execute "Increase/Decrease Action"},
Этот атрибут имеют компоненты с параметрами «Увеличение» и «Уменьшение». Если он отмечен, операция «Увеличение» или «Уменьшение» будет выполняться сразу, не дожидаясь нажатия на компонент.
- Время задержки {Delaying Time}
Если атрибут «Выполнить без задержки» действие увеличения/уменьшения не отмечен, то атрибут «Время задержки» активен. Время по умолчанию установлено — 0,1 с и может быть изменено, максимальное время задержки — 1,5 с. Если время равно 0,1 с, это означает, что действие «Увеличить» будет отложено на 0,1 с для выполнения после нажатия на компонент.
- Время выполнения {Executing Time}
«Время выполнения» означает время для однократного выполнения действия «Увеличить». Диапазон времени составляет от 0,1 с до 1,5 с.
- **Уменьшить {Decrease}** - циклически уменьшать значение в регистре с заданным интервалом
Функция «Уменьшение» аналогична атрибуту «Увеличение». Значение результата будет продолжать уменьшаться, если на компонент осуществляется нажатие. Компонент «Уменьшение» также имеет атрибуты «Немедленно выполнить действие увеличения/уменьшения», «Время задержки» и «Время выполнения».

- **Задать значение {Set Up Constant}** - записать значение или константу в регистр.

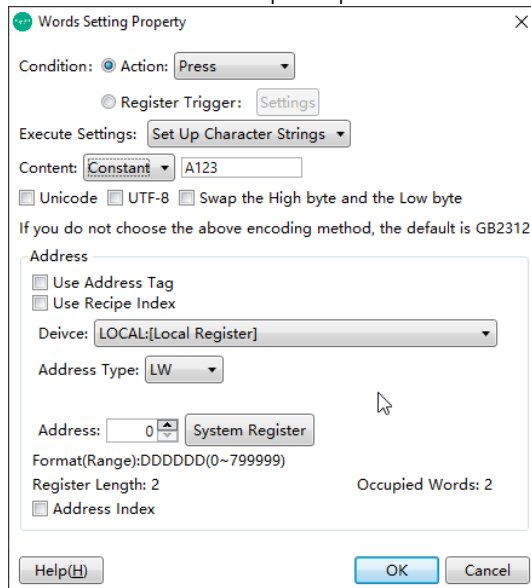
Данные записываются в указанный регистр. Данные (Значение настройки {Setting Value}) могут быть назначены как



«Постоянная» -константа {Constant}, так и значением «Переменной» {Variable}.

На рисунке выше в регистр LW0 HMI записывается константа «0».

- **Задать строку {Set Up Character String}** - записать текстовую строку в регистр (в несколько регистров подряд) Эта функция аналогична функции «Задать значение». С её помощью строка символов может быть записана в регистр слова. Например, строка символов «A123» записывается в регистры LW0 и LW1 HMI.



Настройка строк символов имеет три необязательных атрибута: «UNICODE», «UTF-8» и «Поменять местами старшие и младшие байты». Если флажок «UNICODE» или «UTF-8» установлен, это означает, что строка символов, записываемая в регистры, закодирована в UNICODE или в UTF-8. Строка обычно отображается только в «UNICODE». Если установлен флажок «Поменять местами старшие и младшие байты» {Swap the High byte and Low byte}, старшие и младшие байты в строке символов будут заменены перед записью в регистр слов.

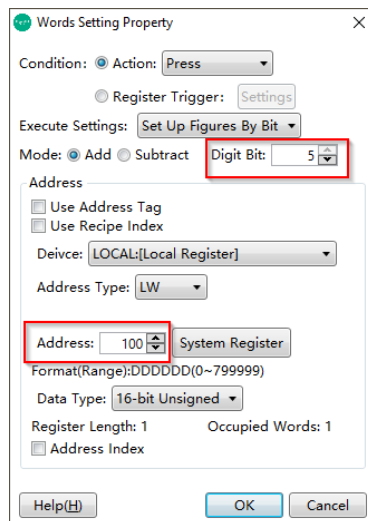
- **Изменить разряд {Set Up Figures By Bit}** - изменить количество единиц, десятков, сотен и т.д. в значении

Эта функция имеет два режима: «Добавить» и «Вычитать» (с помощью этой функции каждый бит значения может быть добавлен или вычтен).

Эта функция имеет атрибут Тип Данных {Data Type}. Это может быть «16-битное число без знака», «32-битное число без знака» или «64-битное число без знака». Другими словами, только беззнаковое одиночное слово и беззнаковое двойное слово могут использовать эту функцию. Для типа данных — «16-битное число без знака», диапазон значений — 0~65535. Таким образом, параметр этой функции «Разрядный бит {Digit Bit}» может принимать значения от 1 до 5. Это означает, что длина числа 1~5 знаков. «1» означает разряд единиц, «2» — разряд десятков, «3» — разряд сотен, «4» — разряд тысяч, «5» — разряд десятков тысяч. Например, если «Digit Bit» равен 3, а «Режим» {Mode} — «Добавить {Add}», это означает, что для сложения используется разряд сотен. Если разряд сотен в значении числа этого регистра равен 6 (например в числе 34632), то при нажатии этого компонента разряд сотен будет увеличиваться в последовательности 7, затем 8, 9, 0, 1, 2 ... и т.д. В этом случае атрибут «Добавить» аналогичен атрибуту «Зацикливание». Но атрибут «Добавить» или «Вычитать» функции «**Set Up Figures By Bit**» действителен только для одной цифры, а остальные цифры не изменяются.

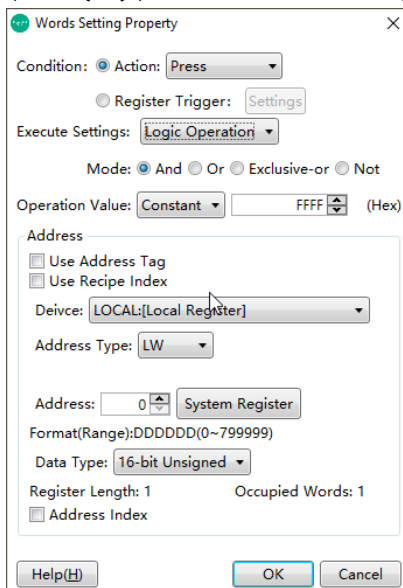
Для примера разместим на создаваемом окне панели оператора пять кнопок и один цифровой дисплей. Все пять зелёных кнопок имеют одинаковые настройки за исключением числа в параметре Digit Bit. Для кнопки Единиц этот параметр равен 1, для кнопки Десятков – 2, для кнопки Сотен – 3 и т.д.

Для всех кнопок (за исключением Digit Bit) настройки выглядят следующим образом.



Адрес регистра для всех кнопок и дисплея, который настроен на отображение числа с пятью знаками выбран один и тот же, например LW100. Нажатием кнопки соответствующего разряда числа, можно увеличивать цифру только в этом разряде числа в регистре LW100. Содержимое числа в регистре LW100 при этом будет отображаться на индикаторе над кнопками.

- **Логическая операция {Logic Operation}** - выполнить в словах побитную логическую операцию. Эта функция имеет четыре режима: «И {AND}», «Или {OR}», «Исключительное ИЛИ {Exclusive-OR}» и «Нет {NOT}».



На рисунке выше, когда компонент работает, значение в регистре LW0 будет иметь логическую операцию «И» с 0xFFFF. Логический результат будет записан в регистр LW0. Операции других трех режимов аналогичны режиму «И», но с соответствующей логической операцией.

Атрибут «Значение операции {Operation Value}» может быть назначен как «Постоянная» - (Константа {Constant}), так и значением «Переменная» {Variable}. По умолчанию установлено значение «Константа», а формат данных — шестнадцатеричный. Это может быть переменная, но тип данных должен соответствовать типу данных регистра.

Атрибут «Тип данных {Data Type}» для логической операции может быть «16-битный без знака», «32-битный без знака», «64-битный без знака», «16-битный двоично-десятичный», и «32-битный двоично-десятичный».

Специальное уведомление:

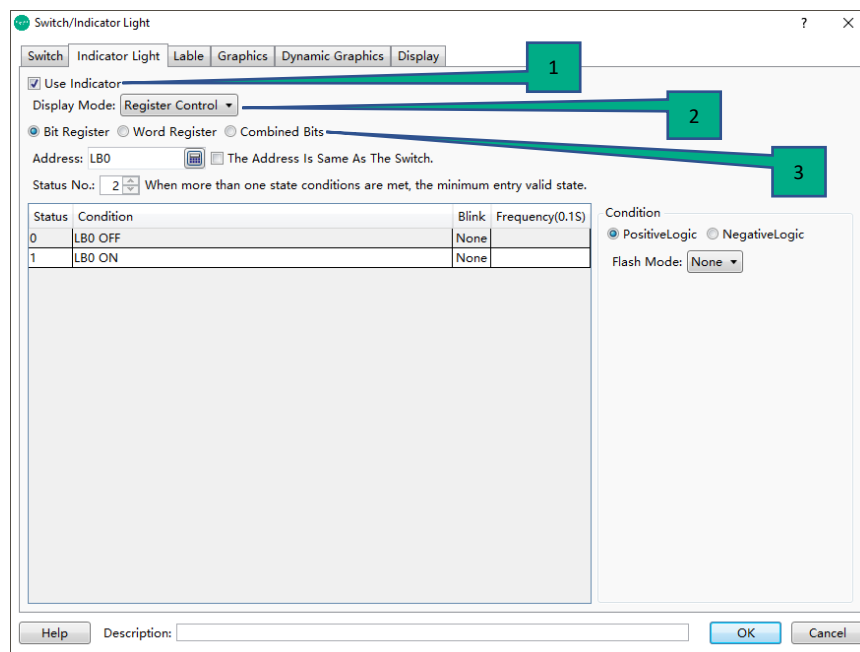
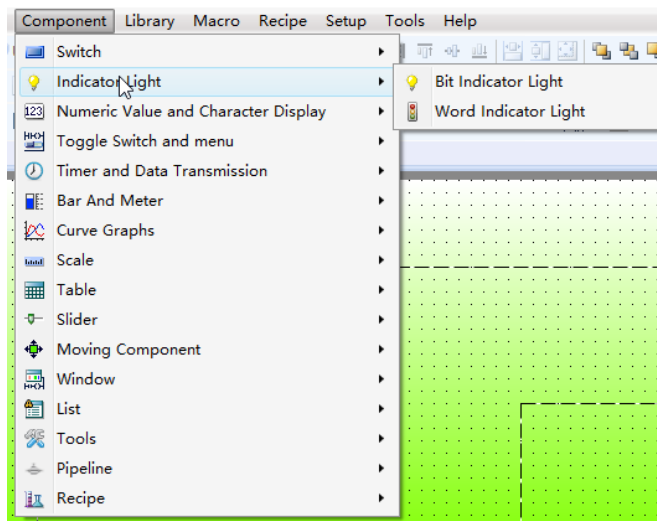
Во всех этих функциях для «Word Set» «Action» по умолчанию — «Нажать {Press}». Это означает, что операция будет выполняться соответственно при нажатии на компонент. «Action» также может быть «Отпустить {Release}», это означает операция будет выполнена соответственно при прекращении нажатия на компонента на экране панели оператора. «Action»- «Отпустить {Release}» не работает с функциями «Увеличение {Increase}» и «Уменьшение {Decrease}».

Индикатор {Indicator Light}

Элемент "Индикатор" сочетает в себе функции визуального индикатора для отображения информации и простейшего

элемента управления. При этом обе функции могут быть реализованы, как по отдельности, так и одновременно в зависимости от настроек

Bit-индикатор {Bit Indicator Light}



1. Опция Индикатор {Use Indicator}. Позволяет задействовать или отключить функцию индикатора для данного элемента.
2. Режим индикации {Display Mode}
Этих режимов может быть два:
 - Состояние регистра {Register Control}
 - Циклическая смена {Automatic looping}
3. Тип индикатора в зависимости от типа данных.
 - Битовый регистр
Когда выбран «Bit регистр» {Bit Register} - компонент является битовым индикатором. Нужно ввести адрес бита.
 - Статус {Status} (перечень состояний).
Статус битового адреса может быть ON (Вкл) или OFF (Выкл). Так что здесь по умолчанию два состояния и это не может быть изменено.
 - Состояние {Condition}
 - Позитивная логика {Positive Logic}
Статус 0 соответствует состоянию выключено (OFF), а статус 1 состоянию включено (ON).
 - Негативная логика {Negative Logic}
Статус 0 соответствует состоянию включено (ON), а статус 1 состоянию выключено (OFF).
 - Мерцание {Flash Mode}

Он включает в себя три режима: «Нет», «Мерцание текста» и «Мерцание фона». Вы можете установить режим мерцания для выбранного статуса. Установленные параметры отобразятся в таблице слева.

- Нет {None} – Означает нет мерцания.
 - Мерцающий текст {Blink Text} – Текст может мерцать с заданной частотой.
 - Мерцающий фон {Blink Picture} – Фон может мерцать с заданной частотой.
 - Частота мерцания {Flash Freq.}
- Задание частоты мерцания фона или текста. Частота задается в десятых долях секунды.

Word – индикатор {Word Indicator Light}

Компонент является Word индикатором, когда выбран «Word - регистр» {Word Register}

Status	Condition	Blink	Frequency(0.1S)
0	LW0 == 0	None	
1	LW0 == 1	None	
2(Error)	Other		

4. Режим индикации {Display Mode}

Этих режимов может быть два:

- Состояние регистра {Register Control}
- Циклическая смена {Automatic looping}

5. Тип индикатора в зависимости от типа данных.

- Word – регистр {Word Register}

Когда выбран «Word - регистр» {Word Register} - компонент является Word индикатором. Для его задания нужно определить и ввести адрес регистра Word.

- Статус {Status No} (перечень состояний).

Количество заданных состояний: от 1 до 256. Каждому состоянию соответствует своё значение адреса слова. Когда несколько статусов одновременно удовлетворяют условию, эффективным будет только статус с наименьшим значением.

- Состояние {Condition}

- Диапазон {Range}

Результирующее значение состояния Word Register будет определяться в соответствии с результатом сравнения и логической операции значения адреса слова. Операция сравнения включает «>», «<», «>=», «<=», «==» и «!=». Логическая операция может быть «И» {AND}, «ИЛИ» {OR} или «Нет логической операции» {None}.

- Bit.

Текст или фон могут отображаться в соответствии с состоянием одного бита адреса слова, например: бит LW0.1 в адресе слова LW0.

- *Состояние {Condition}*
- Позитивная логика {Positive Logic}
Статус 0 соответствует состоянию выключено (OFF), а статус 1 состоянию включено (ON).
- Негативная логика {Negative Logic}
Статус 0 соответствует состоянию включено (ON), а статус 1 состоянию выключено (OFF).

- *Мерцание {Flash Mode}*
Он включает в себя три режима: «Нет», «Мерцание текста» и «Мерцание фона». Вы можете установить режим мерцания для выбранного статуса. Установленные параметры отобразятся в таблице слева.
- Нет {None} – Означает нет мерцания.
- Мерцающий текст {Blink Text} – Текст может мерцать с заданной частотой.
- Мерцающий фон {Blink Picture} – Фон может мерцать с заданной частотой.
- *Частота мерцания {Flash Freq.}*
Задание частоты мерцания фона или текста. Частота задается в десятых долях секунды.

Status	Condition	Tag Content	Blink	Frequen
0	LW0 Bit0 OFF		Text	5
1	LW0 == 1		Picture	5
2(Error)	Other		None	

Illegal Input: ☒ Show Error Status ☐ Keep Current Status

☒ Error Notification LB0

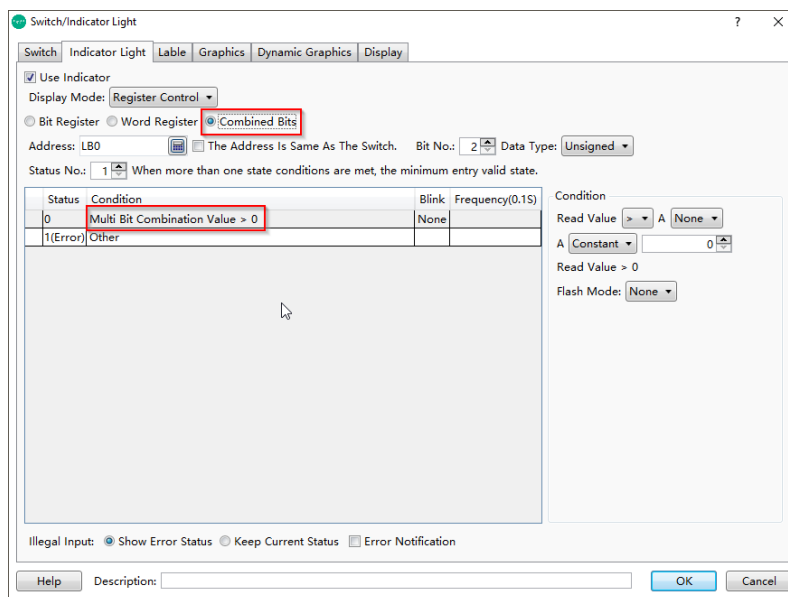
- *Недопустимый ввод {Illegal Input}*
Значение адреса слова не удовлетворяет заданному условию.
 - Показать статус ошибки {Show Error Status}
Индикатор отображает статус «Ошибка», когда произошел неправильный ввод. Это последний статус отображены в таблице.
 - Сохранить текущий статус {Keep Current status}
Индикатор сохраняет текущий статус, когда произошел незаконный ввод.
- *Уведомление об ошибке {Error Notification}*
Здесь вы можете выбрать битовый адрес. Он будет установлен на ON, если значение адреса слова не удовлетворяет условию. Он будет установлен на OFF, когда условие будет выполнено.

Multi Bit Combination Indicator Light

- Режим индикации {Display Mode}
Этих режимов может быть два:
- *Состояние регистра {Register Control}*
- *Циклическая смена {Automatic looping}*

➤ *Последовательность {Combined Bits}*

Отображаемый статус определяется условием комбинации нескольких битов.



➤ *Бит № {Bit No}*

Условие может быть комбинацией 2 до 32 бит. «Бит №» означает количество бит.

➤ *Тип данных {Data Type}*

Без знака

Если количество бит равно n, значение комбинации будет от 0 до (2n-1).

Со знаком

Если количество бит равно n, значение комбинации будет от -(2n-1) до +(2n-1)-1.

➤ *Начальный адрес {Starting Address}*

Начальный адрес тесно связан с количеством битов. Например, если начальный адрес LB0 и количество битов равно 2, будет сформирована комбинация. LB1 будет старшим битом, а LB0 будет младшим битом.

Диапазон 2-битного значения комбинации будет от 0 до 3, если выбран тип данных без знака.

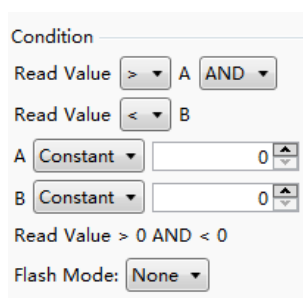
Диапазон 2-битного значения комбинации будет равен от -2 до 1, если выбран тип данных со знаком.

➤ *Статус № {Status No} (перечень состояний).*

Количество заданных условий состояния: от 1 до 256. Каждому условию соответствует своё значение в слове по указанному адресу. Когда несколько статусов одновременно удовлетворяют условию, эффективным будет только статус с наименьшим значением.

➤ *Состояние {Condition}*

Содержимое состояния (значение) будет определяться в соответствии со значением задействованных в логическом условии переменных. Операция сравнения включает «>», «<», «>=», «<=», «==» и «!=». Логическая операция может быть «И» {AND}, «ИЛИ» {OR} или «Нет логической операции» {None}.



➤ *Мерцание {Flash Mode}*

Он включает в себя три режима: «Нет», «Мерцание текста» и «Мерцание фона». Вы можете установить режим мерцания для выбранного статуса. Установленные параметры отобразятся в таблице слева.

➤ Нет {None} – Означает нет мерцания.

➤ Мерцающий текст {Blink Text} – Текст может мерцать с заданной частотой.

➤ Мерцающий фон {Blink Picture} – Фон может мерцать с заданной частотой.

➤ *Частота мерцания {Flash Freq.}*

Задание частоты мерцания фона или текста. Частота задается в десятых долях секунды.

Подробную информацию об атрибутах текущего компонента можно увидеть в таблице слева.

Switch/Indicator Light

Switch Indicator Light Lable Graphics Dynamic Graphics Display

☒ Use Indicator

Display Mode: Register Control

☐ Bit Register ☐ Word Register ☒ Combined Bits

Address: LB0 ☐ The Address Is Same As The Switch. Bit No.: 2 Data Type: Unsigned

Status No.: 3 When more than one state conditions are met, the minimum entry valid state.

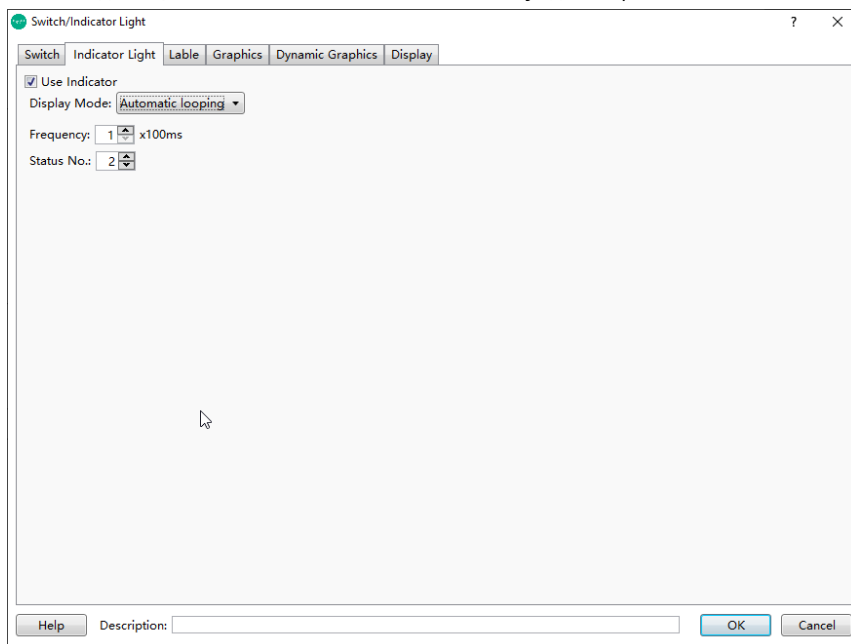
Status	Condition	Blink	Frequency(0.1S)
0	Multi Bit Combination Value == 1	None	
1	Multi Bit Combination Value == 2	None	
2	Multi Bit Combination Value == 3	None	
3(Error)	Other		

Illegal Input: ☒ Show Error Status ☐ Keep Current Status ☒ Error Notification LB0

Help Description: OK Cancel

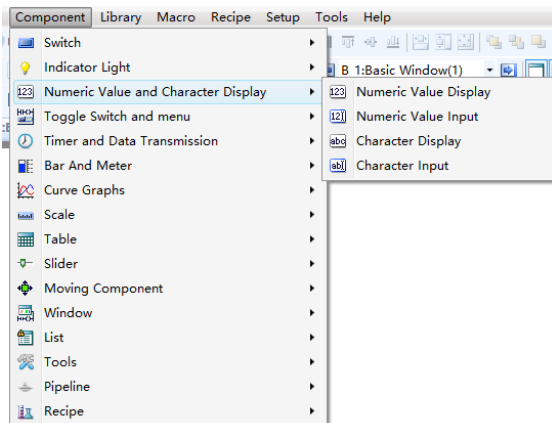
- **Недопустимый ввод {Illegal Input}**
Значение многобитовой комбинации не удовлетворяет предустановленному условию.
- **Показать статус ошибки {Show Error Status}**
Индикатор отображает статус «Ошибка», когда произошел неправильный ввод. Этот статус, отображается в последней строке таблицы.
- **Сохранить текущий статус {Keep Current Status}**
Индикатор сохраняет текущий статус, когда произошел некорректный ввод.
- **Уведомление об ошибке {Error Notification}**
Здесь вы можете выбрать битовый адрес. Он будет установлен на ON, если значение многобитовой комбинации не удовлетворяет условию. Он будет установлен на OFF, когда условие будет выполнено.
- **Display Mode**
Если на вкладке свойств окна «Индикатор» {Indicator Light} Вы отмените «Использовать Индикатор» {Use Indicator}, то первой опцией выбора будет «Режим отображения (Режим индикации) {Display Mode}».
Он включает в себя два варианта:
 - Состояние Регистра {Register Control}
 - Циклическая смена {Automatic looping}
 1. Состояние Регистра {Register Control} режим отображения зависящий от состояния указанного регистра.
В качестве инструмента управления можно выбрать:
 - состояние Bit-регистра,
 - содержимое Word-регистра
 - или результат логической операции комбинации из нескольких бит.
 2. Автоматическое заикливание

Если выбран режим отображения «Циклическая смена», то все состояния будут циклически отображаться в соответствии с указанной частотой. Единицей частоты является 100 миллисекунд. Настройка показана ниже.



Дисплеи {Numeric Value and Character Display}

Цифровой дисплей {Numeric Value Display}. Отображение числового значения.



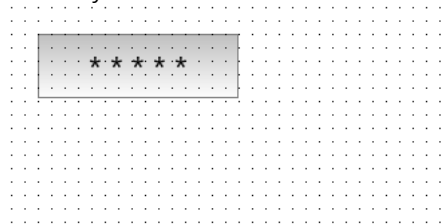
- Общие положения {General}
 - Атрибут операции {Operation Attribute}
- « Атрибут операции » включает четыре типа операций: « Цифровой дисплей », « Цифровой ввод», « Отображение символов » и

«Ввод символов» . Здесь можно изменить тип операции этого текущего компонента. Он очень эффективен для модификации и обслуживания проекта.

Режим отображения экрана {Display Mode}

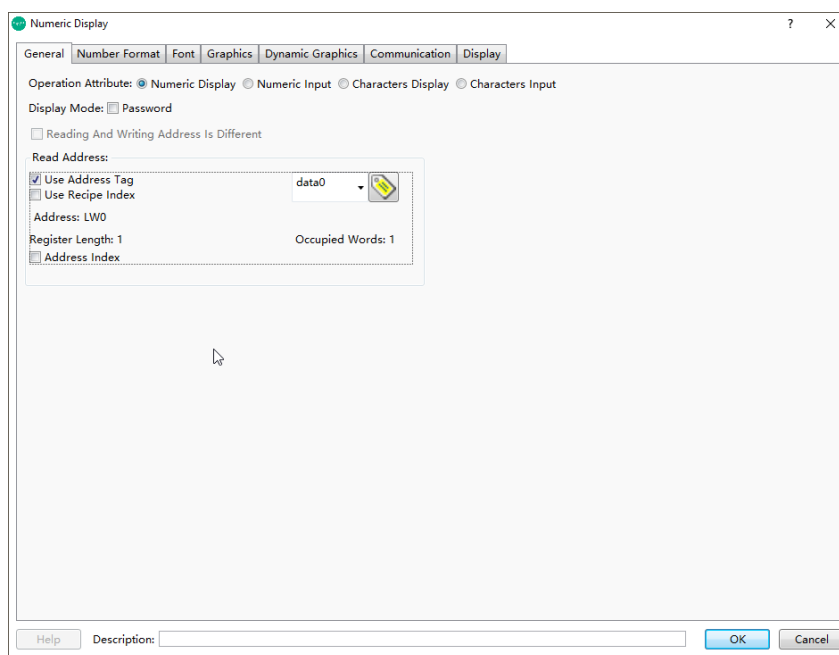
➤ *Пароль {Password}*

Когда установлен флажок «Пароль», в компоненте (отображении значения) будет отображаться «*» (см. рисунок ниже). Он всегда используется для компонента ввода числового значения. Компонент отображения числового значения обычно не используется.



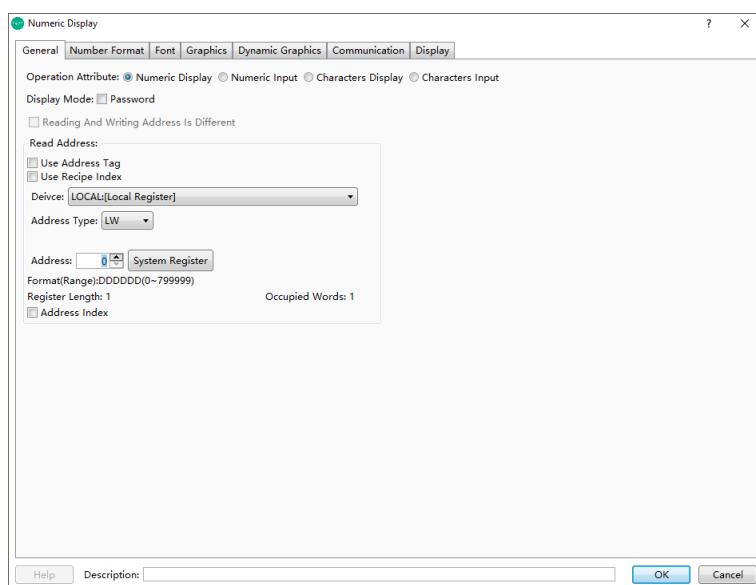
➤ *Адрес чтения {Read Address}*

➤ *Использовать адресную метку {Use Address Tag}*

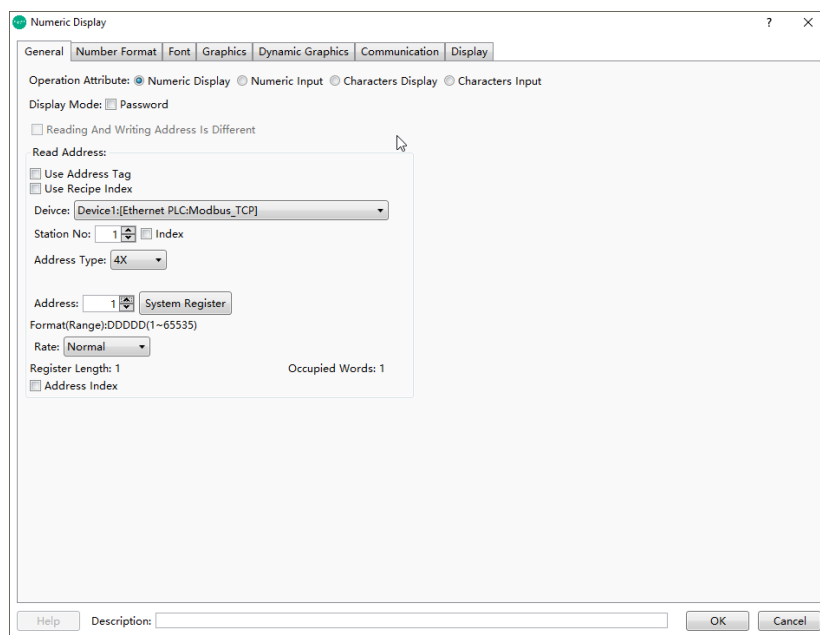


Когда выбрана эта функция, переменные в библиотеке адресных тегов {Address Tag Library} могут быть напрямую сопоставлены требуемым компонентам . Библиотека адресных тегов может быть использована для повышения эффективности работы с проектом. Если многие компоненты в проекте связаны с одной и той же переменной в библиотеке адресных тегов, и этот адрес необходимо изменить, необходимо изменить только адрес переменной, связанный в библиотеке тегов, без изменения каждого компонента.

➤ *Настройка адреса {Device, Station No, Address Type, Address...}*

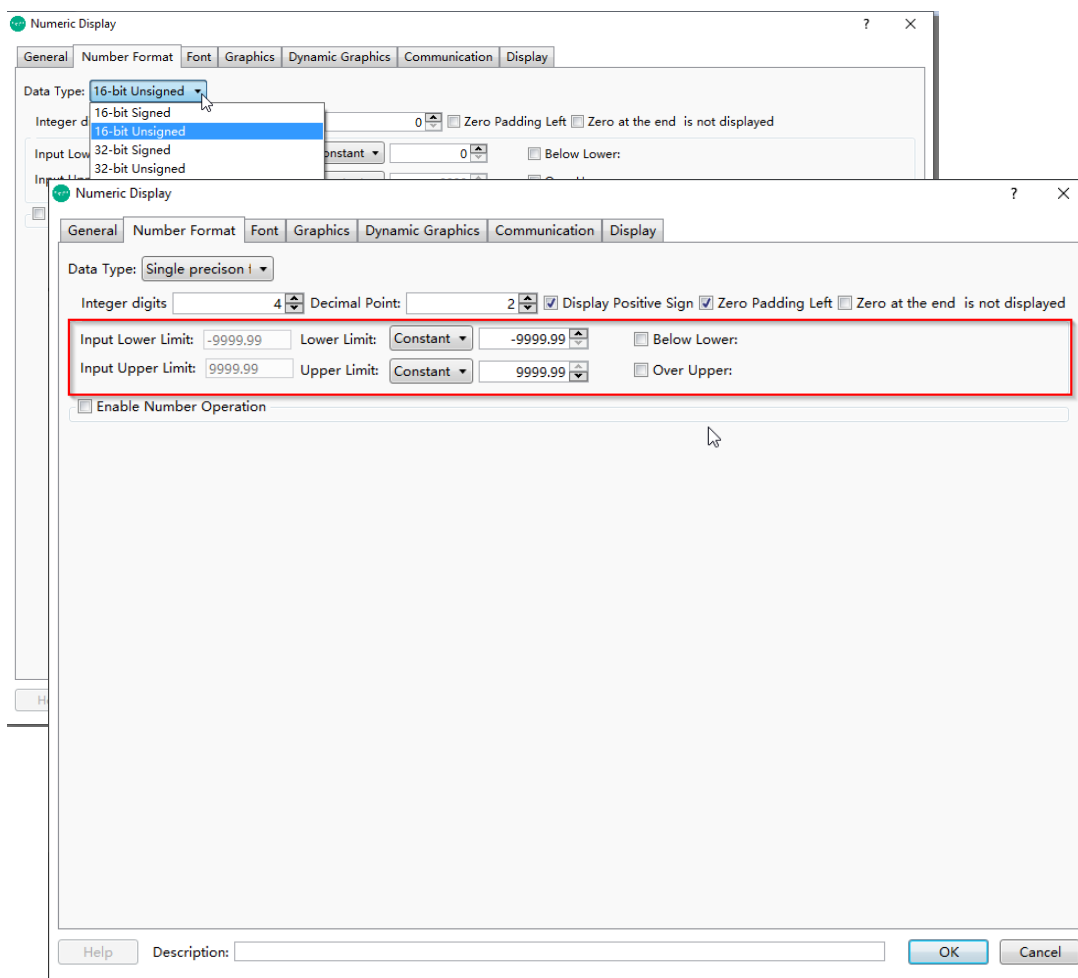


Настройка адреса включает атрибуты: « Устройство {Device} », « Тип адреса {Address Type} », «Адрес {Address}», «Формат (диапазон) {Format (Range)} », « Указатель адреса {Address Index}» и так далее.



- *Формат числа {Number Format}*
- *Тип данных {Data Type}*
 - « Тип данных » может быть:
 - «16-битный со знаком {16-bit Signed}»,
 - «16-битный без знака {16-bit Unsigned}»,
 - «32-битный со знаком {32-bit Signed}»,
 - «32-битный без знака {32-bit Unsigned}»,
 - «16-битный двоично-десятичный код {16-bit BCD (Binary Coded Decimal)}»,
 - «32-битный BCD {32-bit BCD}»,
 - «16-битное шестнадцатеричное {16-bit Hexadecimal}»,
 - «32-битное шестнадцатеричное {32-bit Hexadecimal}»,
 - «16-битное двоичное {16-bit Binary}»,
 - «32-битное двоичное {32-bit Binary}»,
 - «число с плавающей запятой одинарной точности {Single precision floating point number}»,
 - «число с плавающей запятой двойной точности {Double precision floating point number}»,
 - «64-битный со знаком {64-bit Signed}»,
 - «64-битный без знака {64-bit Unsigned}»,

«64-битный без знака {16-bit Signed}»



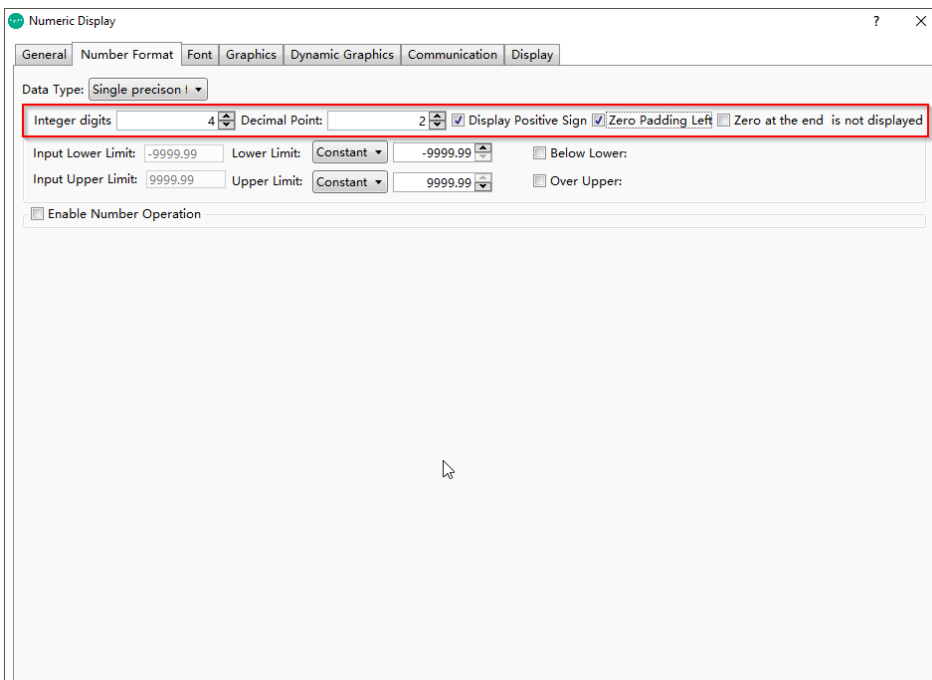
- **Формат отображения числа {Number Format}. Формат дисплея.**

«Формат дисплея» включает «Количество целых знаков числа, (Разрядов) {Integer digits} », « Количество знаков после запятой, (Разделитель){Decimal Point}», «Отображение положительного знака {Display Positive Sign}» и « Заполнение нулями слева {Zero Padding Left} » .

«Integer digits» означает целое число цифр данных. «Decimal Point» означает число десятичных цифр данных.

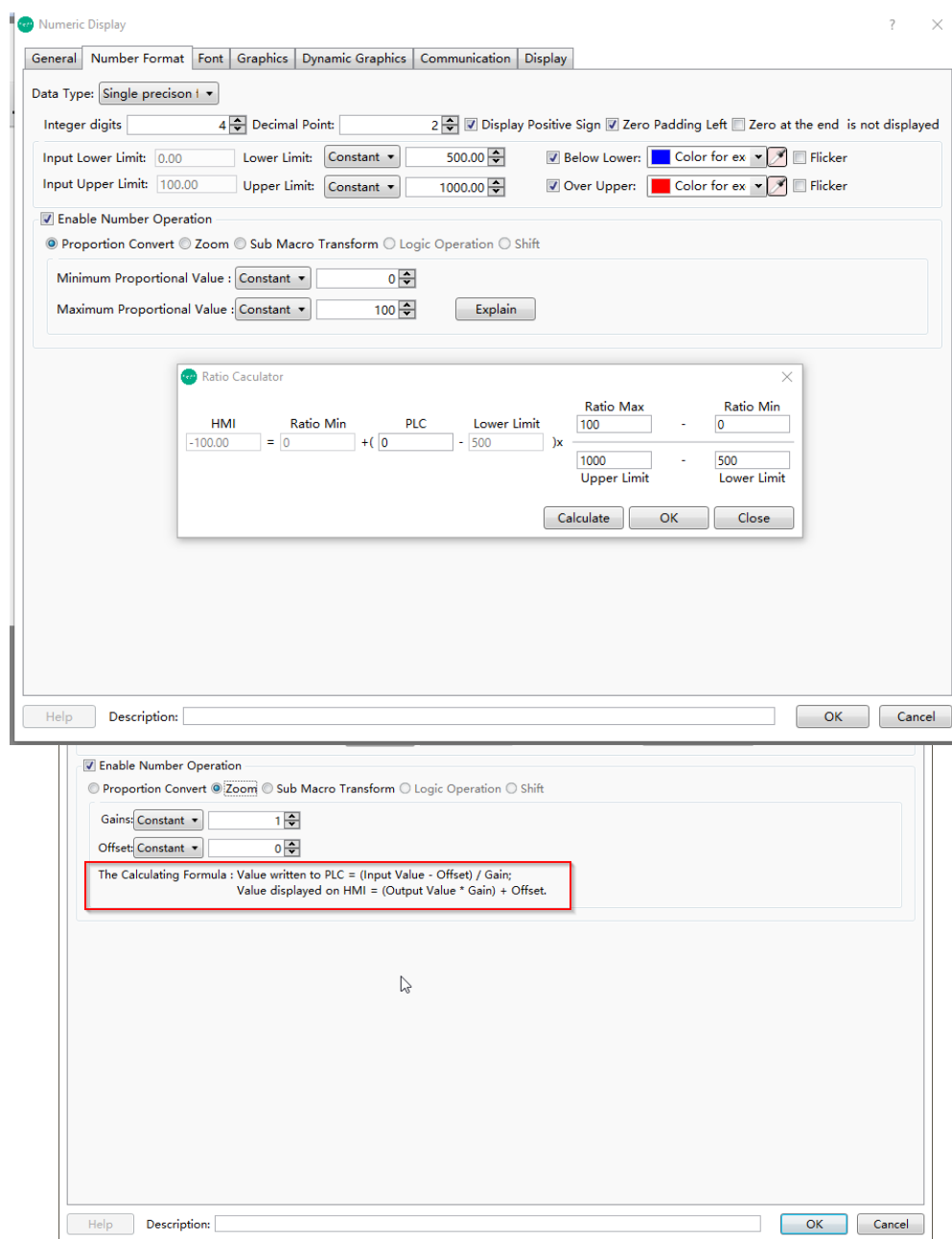
Примечание: когда тип данных является целым числом, а количество десятичных знаков не равно нулю, отображаемое значение будет уменьшено, чтобы соответствовать заданному десятичному числу цифр. Например, если данные представляют собой целое число «55» и установлены два десятичных бита, отображаемое значение будет «0,55» . Фактически, эта функция может изменить только значение и тип отображаемых данных. Фактическое значение и фактический тип не изменяются. Это по-прежнему целое число «55» .

- **Верхний/нижний предел числа**

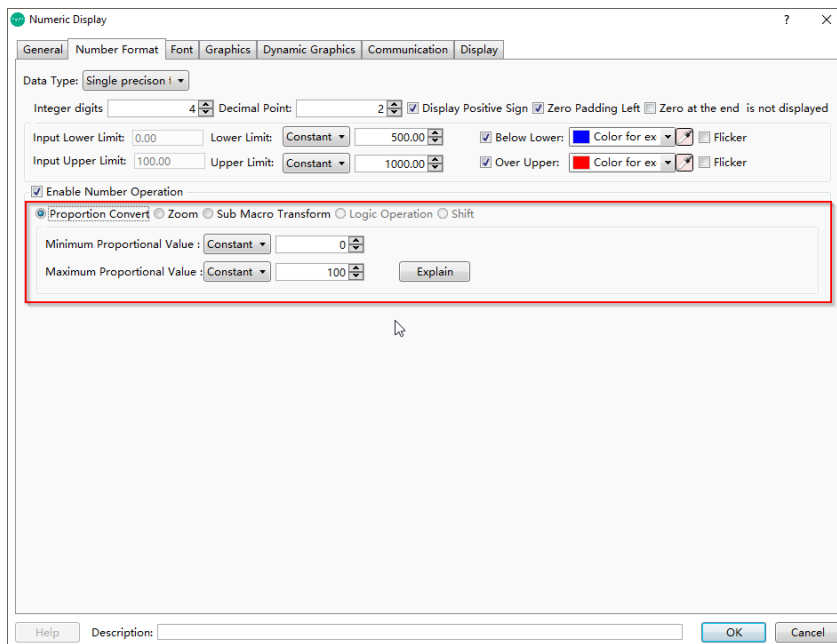


Вы можете задать число выше предела установленного в графе «Нижний предел {Lower Limit}» и ниже предела, установленного в графе «Верхний предел {Upper Limit}». Также Вы можете установить цвет шрифта данных за пределами. Когда установлен флажок «Мерцание {Flicker}», текст будет мигать, если данные меньше нижнего предела или больше верхнего предела.

- Разрешить числовые операции {Enable Number Operation}



- Пропорциональное изменение {Proportional Convert},
Когда активирована функция «Пропорциональное изменение {Proportional Convert}», отображаемое значение будет результатом пропорционального преобразования. Формула пропорционального преобразования отображается при нажатии кнопки «Развернуть (Пояснить) {Explain}».



The Calculating Formula of Using Zooming Function Is As Below:

Value written to PLC = (Input Value - Offset) + Gain

Value displayed on HMI = (Output Value * Gain) + Offset

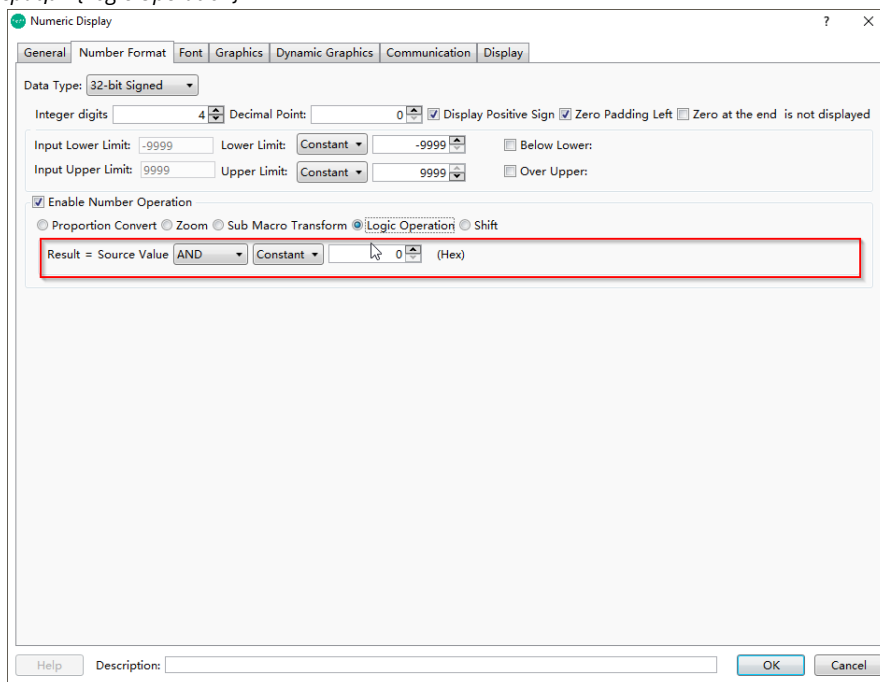
$$\text{Output Value} - \text{Minimum Input Value} / (\text{Maximum Ratio Value} - \text{Minimum Ratio Value})$$

$$\text{HMI Indicating Value} = \text{Minimum Ratio Value} + (\text{Output Value} - \text{Minimum Input Value}) * (\text{Maximum Ratio Value} - \text{Minimum Ratio Value}) / (\text{Maximum Input Value} - \text{Minimum Input Value})$$

➤ *Масштаб {Zoom}*

Когда выбрана функция «Масштаб {Zoom}», отображаемое значение будет результатом преобразования масштабирования. Формула преобразования масштабирования {The Calculating Formula} отображается текстом в этом же окне ».

➤ **Логическая операция {Logic Operation}**



Данные могут логически обрабатываться компонентом, логическая операция которого может быть «И {AND}», «ИЛИ {OR}» или «Исключающее ИЛИ {XOR}». В итоге будет отображаться результат логической операции.

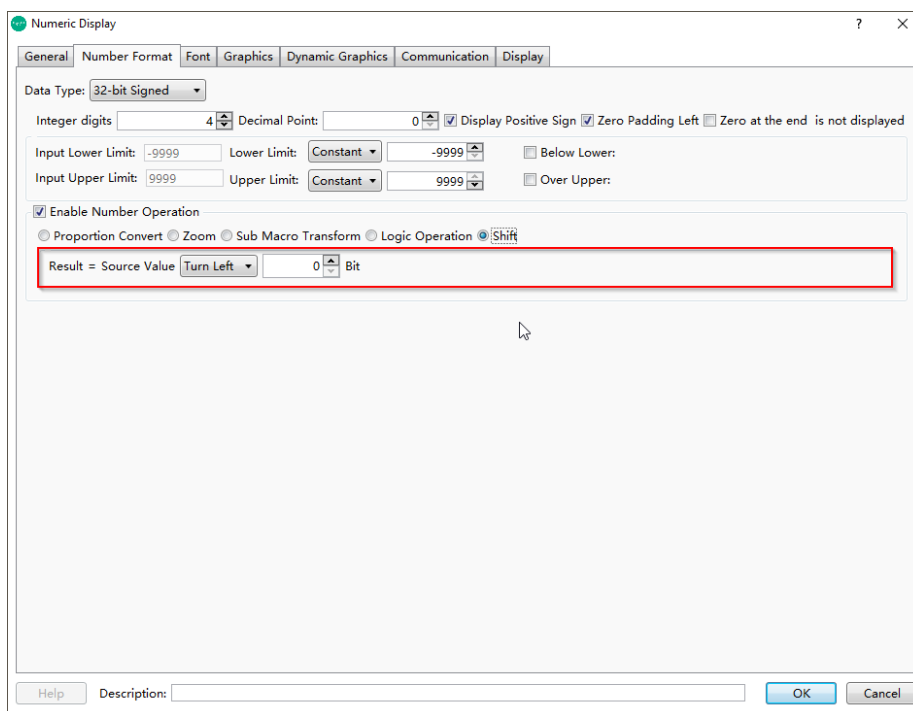
➤ **Сдвиг {Shift}**

Данные могут быть смещены влево или вправо компонентом. И результат операции переключения будет отображаться.

Элементы закладок *Шрифт {Font}*», «Оформление (Графика) {Graphics}», «Анимация (Динамическая графика) {Dynamic Graphics}», «Обмен {Communication}», «Отображение {Display}» описаны в разделе «Рисование {Drawing}» т.к. имеют одинаковую структуру для подобных закладок, где они используются.

Ввод числового значения {Numeric Input}. Цифровой ввод.

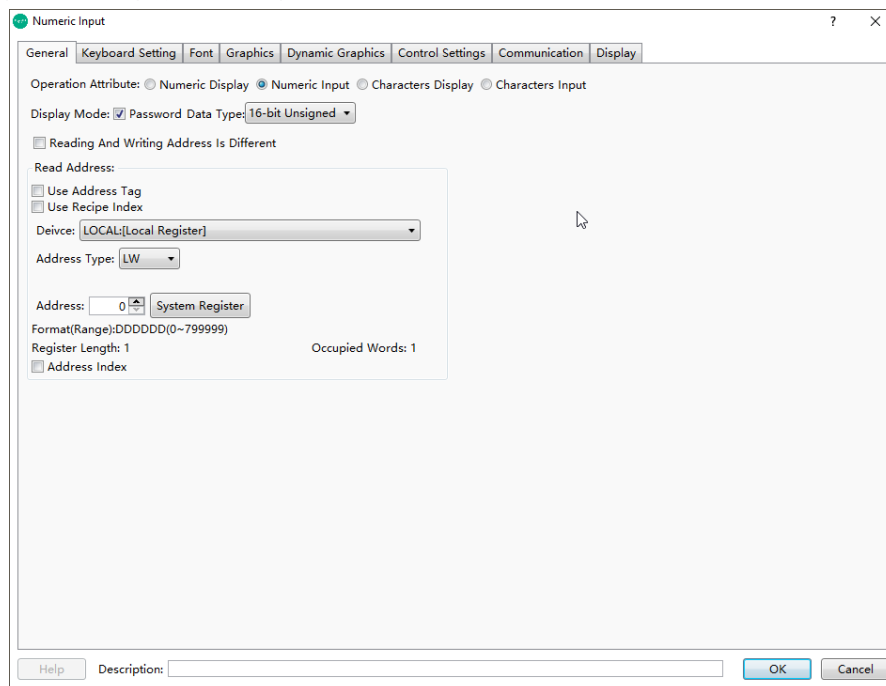
Настройка атрибутов компонента «Цифровой ввод» обычно такая же, как и у компонента «Цифровой дисплей». Различия



подробно описаны ниже.

Общие положения {General}

- Режим отображения {Display Mode}. Пароль {Password}

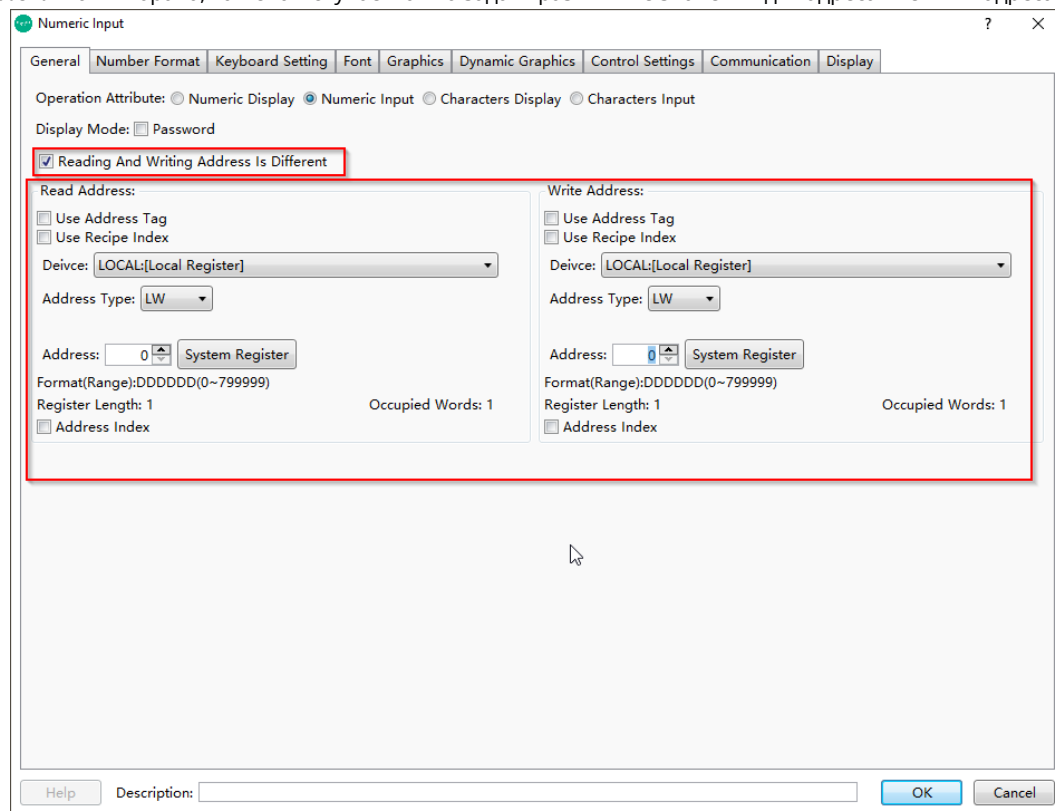


Когда установлен флажок «Пароль», в компоненте будет отображаться «*» (см. рисунок ниже). Эта опция может быть выбрана, если текущий компонент используется для ввода пароля.



- Адреса Чтения и Записи различны {Reading and Writing Address is Different}

Компонент «Ввод числового значения {Numeric Input}» интегрирован с функциями Чтения данных из адреса и Записи данных в адрес. Когда опция «Адреса Чтения и Записи различны» не отмечена, метод задания адреса для Numeric Input использования такой же, как и для компонента «Цифровой Дисплей {Numeric Value Display}». Если опция «Адреса Чтения и Записи различны» выбрана, то в этом случае можно задать различные значения для адреса Чтения и адреса Записи.

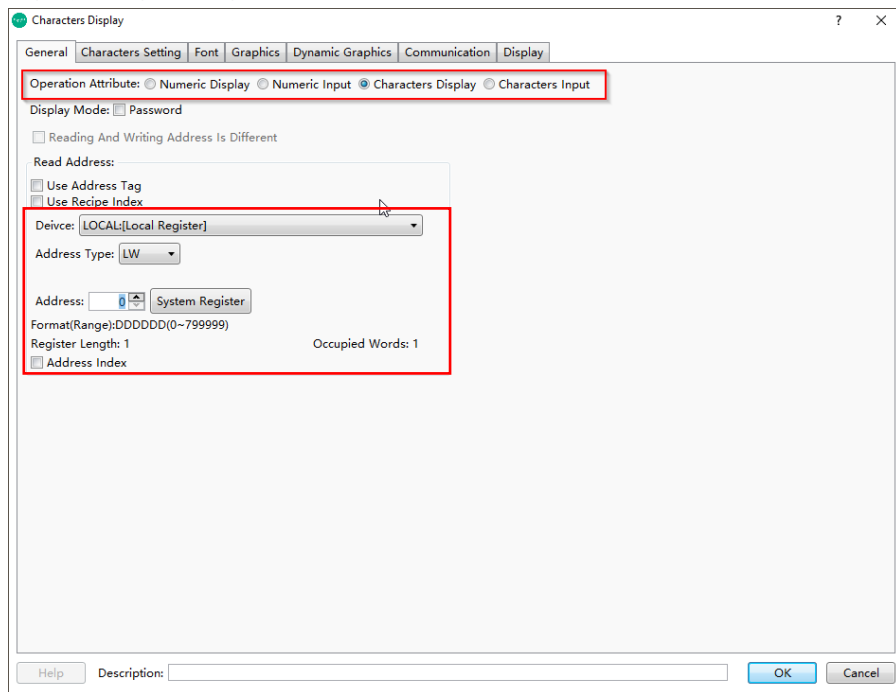


Элементы остальных закладок «Формат дисплея {Number Format}», «Настройка клавиатуры {Keyboard Setting}», «Шрифт {Font}», «Оформление (Графика) {Graphics}», «Анимация (Динамическая графика) {Dynamic Graphics}», «Обмен {Communication}», «Отображение {Display}» описаны в разделе «Рисование {Drawing}» т.к. имеют одинаковую структуру для всех закладок, где они используются.

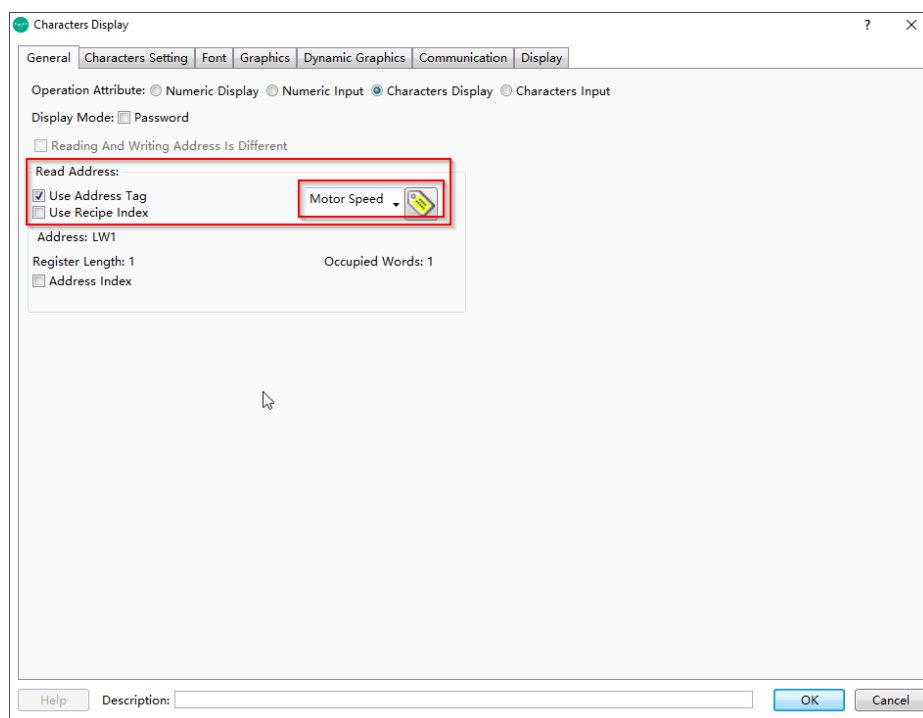
Символьный дисплей {Characters Display}

Основные положения {General}

- Атрибут операции {Operation Attribute}

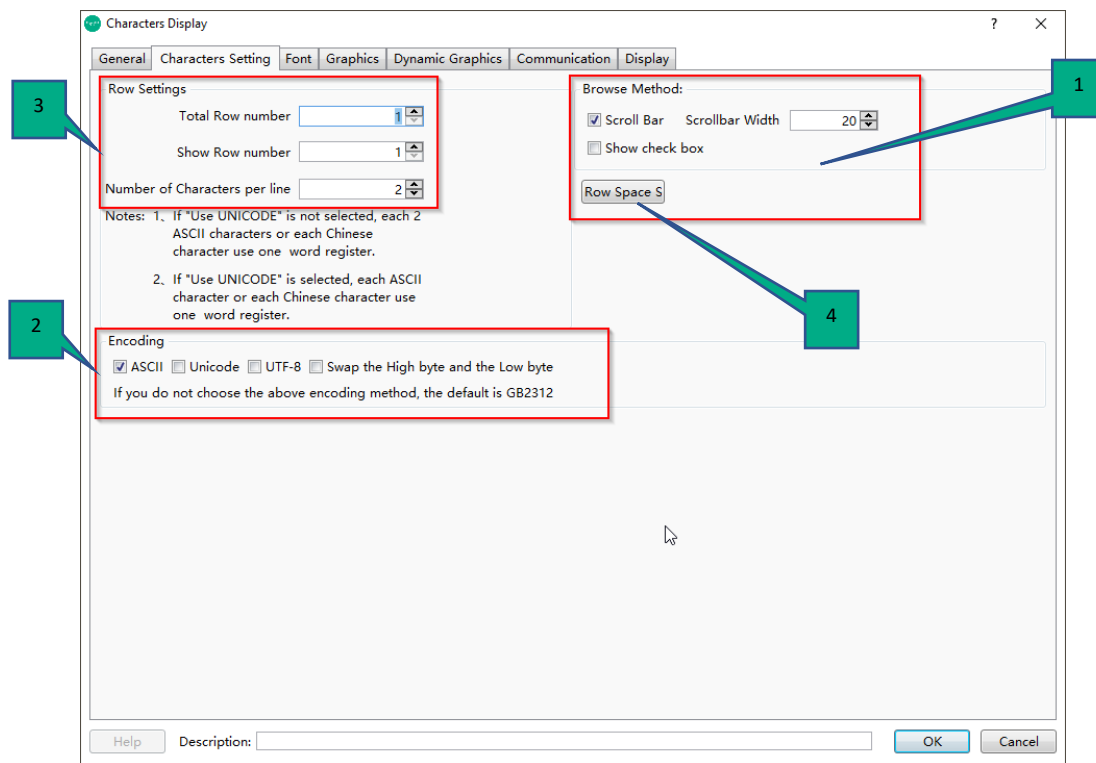


- «Атрибут операции {Operation Attribute}» включает четыре типа операций: «Цифровой дисплей {Numeric Display}», «Цифровой ввод {Numeric Input}», «Символьный дисплей {Characters Display}» и «Ввод символьных значений {Characters Input}» . Если вы хотите, чтобы текущий компонент, например «Символьный дисплей», был изменен на компонент «Ввод символьных значений» или один из «Числовых» компонентов, вы можете изменить его здесь. Это очень удобно для модификации и обслуживания проекта. Настройка адреса включает атрибуты: «Устройство {Device}», «Тип адреса {Address Type}», «Адрес {Address}», «Формат (диапазон) {Format (Range)}», «Указатель адреса {Address Index}» и так далее. См.: Раздел «General functions/ Address editor/ Standard Byte Address Input.»
- Чтение адреса {Read Address}. Использовать адресную метку {Use Address Tag}.



Когда эта функция отмечена, переменные в Библиотеке адресных тегов {Address Tag Library} могут быть напрямую связаны компонентом. Библиотека адресных тегов может быть использована для повышения эффективности обслуживания проекта.

Если в Библиотеке адресных тегов проекта несколько компонентов связаны с одной и той же переменной, и этот адрес нужно изменить, то необходимо изменить только адрес этой переменной, связанный в библиотеке тегов с компонентами, без изменения каждого компонента.



Формат дисплея {Characters Setting}

1. Метод просмотра {Browse Method}

Вы можете использовать «Полосу прокрутки {Scroll Bar}» для просмотра всего содержимого символьного дисплея. Ширина полосы прокрутки {Scrollbar Width} может быть установлена произвольно.

2. Декодирование {Encoding}

➤ ASCII

Если этот флажок установлен, символы будут отображаться в формате ASCII. В этом формате будут отображаться только символы ASCII, включая цифры, символы и английские буквы.

➤ Юникод {Unicode}

Если этот флажок установлен, символы будут отображаться в формате UNICODE. В режиме UNICODE каждый символ или буква будут занимать два байта.

➤ Поменять местами старший байт и младший байт

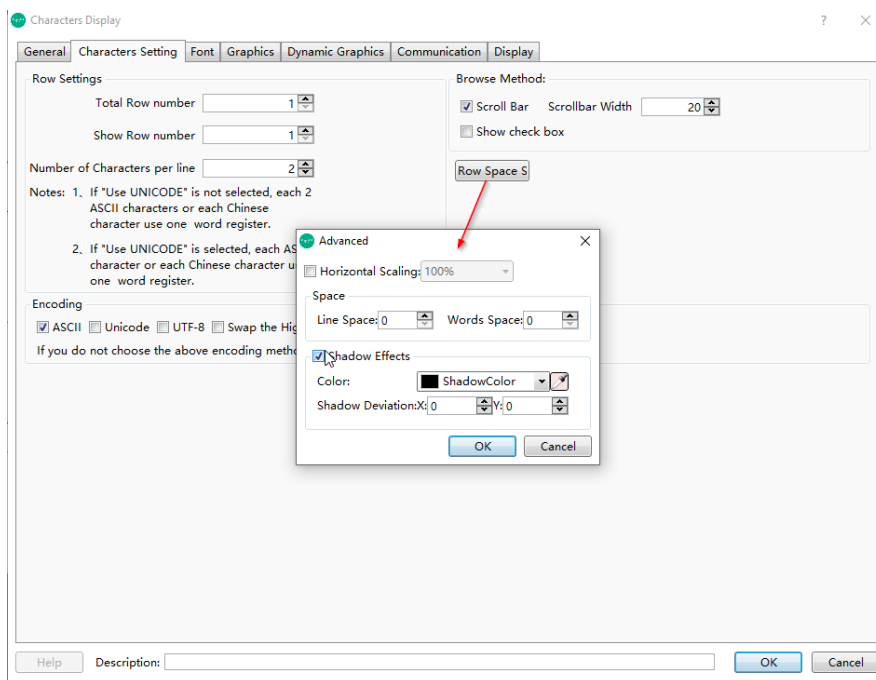
После проверки кода «ASCII» вы можете выбрать функцию «Поменять местами старший и младший байты».

3. Настройки строки {Row Settings}

«Настройки строки {Row Setting}» включают «Общее количество строк {Total Row number}», «Показать номер строки (это количество строк для отображения) {Show Row number}» и «Количество символов в строке {Number of Characters per line}». «Общее количество строк» означает количество строк, содержащихся в текущем символьном дисплее. «Показать номер строки» устанавливает количество строк, отображаемых в текущем дисплее. Если количество отображаемых строк меньше, чем общее количество строк, полоса прокрутки появится автоматически.

4. Настройки пространства между строками

Дополнительное диалоговое окно появится, если нажать кнопку «Настройка... {Row Space Setting}». Вы можете установить такие атрибуты, как «Горизонтальное масштабирование {Horizontal Scaling}», «Межстрочный интервал {Line Space}», «Область (интервал, отступ) между символами {Word Space}», «Теневые эффекты {Shadow Effect}».

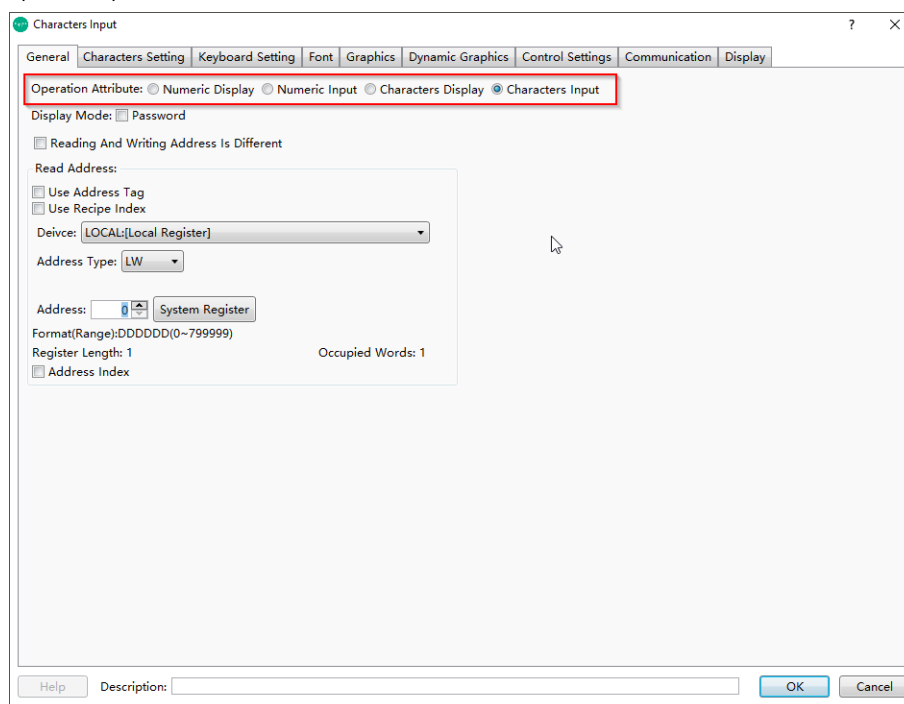


Элементы остальных закладок «Шрифт {Font}», «Оформление (Графика) {Graphics}», «Анимация (Динамическая графика) {Dynamic Graphics}», «Обмен {Communication}», «Отображение {Display}» описаны в разделе «Рисование {Drawing}» т.к. имеют одинаковую структуру для всех закладок, где они используются.

Ввод символов {Characters Input}

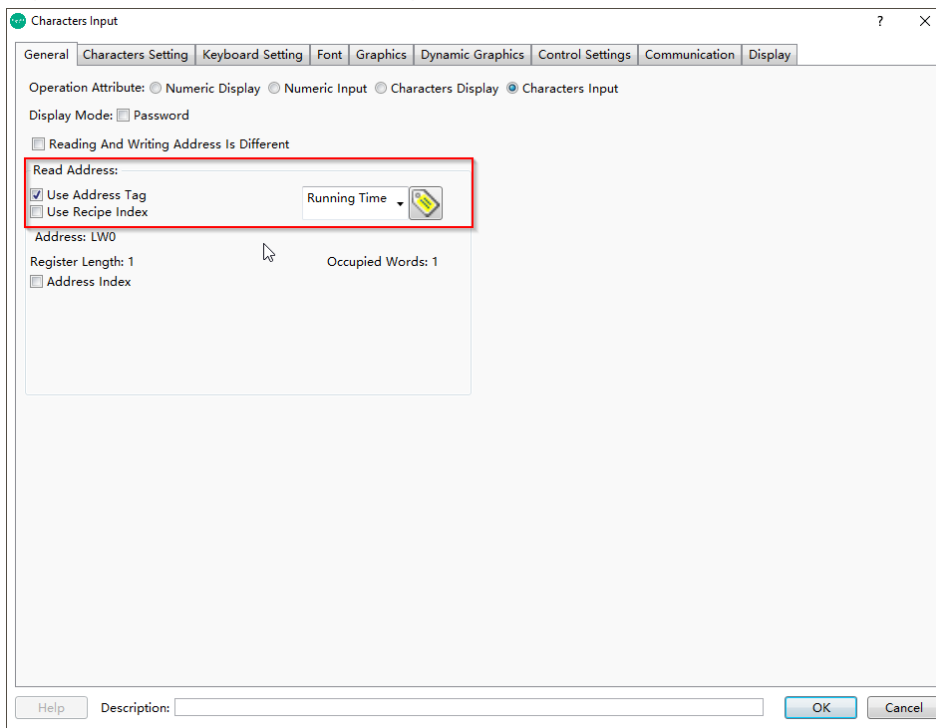
Основные положения {General}

- Атрибут операции {Operation Attribute}



- «Атрибут операции {Operation Attribute}» включает четыре типа операций: «Цифровой дисплей {Numeric Display}», «Цифровой ввод {Numeric Input}», «Символьный дисплей {Characters Display}» и «Ввод символьных значений {Characters Input}». Если вы хотите, чтобы текущий компонент, например «Символьный дисплей», был изменен на компонент «Ввод символьных значений» или один из «Числовых» компонентов, вы можете изменить его здесь. Это очень удобно для модификации и обслуживания проекта.
- «Настройка адреса {Read Address}» включает атрибуты: «Устройство {Device}», «Тип адреса {Address Type}», «Адрес {Address}», «Формат (диапазон) {Format (Range)}», «Указатель адреса {Address Index}» и так далее. См.: Раздел «General functions/ Address editor/ Standard Byte Address Input.»

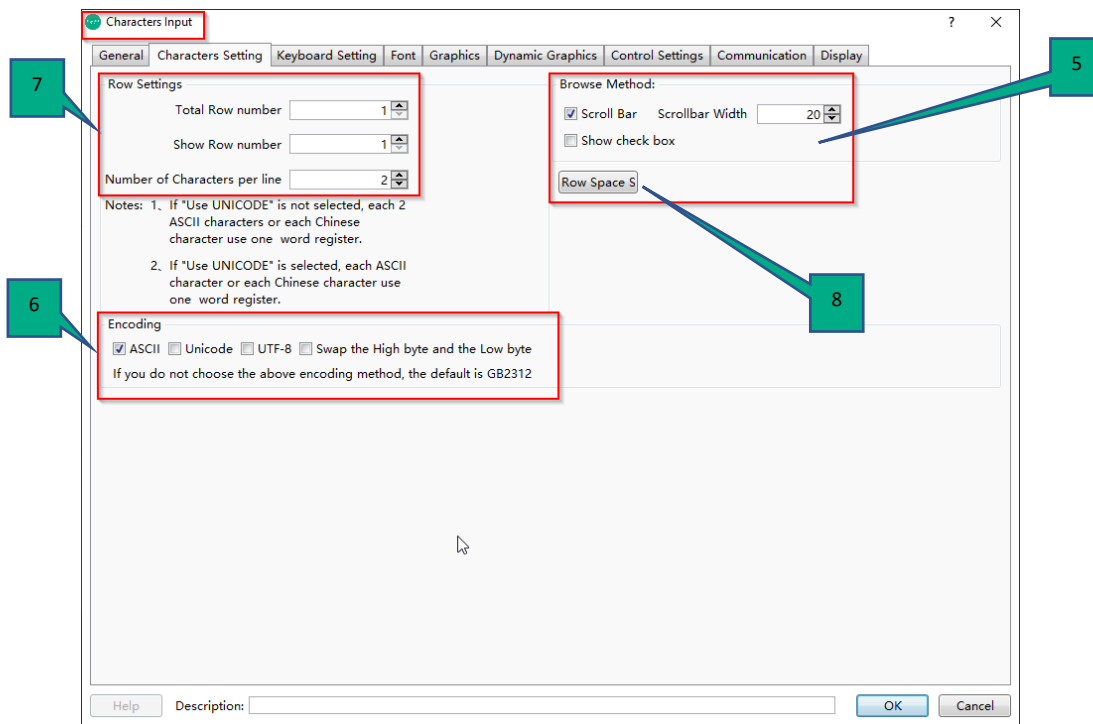
- Чтение адреса {Read Address}. Использовать адресную метку {Use Address Tag}.



Когда эта функция отмечена, переменные в Библиотеке адресных тегов {Address Tag Library} могут быть напрямую связаны компонентом. Библиотека адресных тегов может быть использована для повышения эффективности обслуживания проекта.

Если в Библиотеке адресных тегов проекта несколько компонентов связаны с одной и той же переменной, и этот адрес нужно изменить, то необходимо изменить только адрес этой переменной, связанный в библиотеке тегов с компонентами, без изменения каждого компонента.

- Формат дисплея {Characters Setting}



5. Метод просмотра {Browse Method}

Вы можете использовать «Полосу прокрутки {Scroll Bar}» для просмотра всего содержимого символического дисплея. Ширина полосы прокрутки {Scrollbar Width} может быть установлена произвольно.

6. Декодирование {Encoding}

- ASCII

Если этот флажок установлен, символы будут отображаться в формате ASCII. В этом формате будут отображаться только символы ASCII, включая цифры, символы и английские буквы.

➤ **Юникод {Unicode}**

Если этот флажок установлен, символы будут отображаться в формате UNICODE. В режиме UNICODE каждый символ или буква будут занимать два байта.

➤ **Поменять местами старший байт и младший байт**

После проверки кода «ASCII» вы можете выбрать функцию «Поменять местами старший и младший байты».

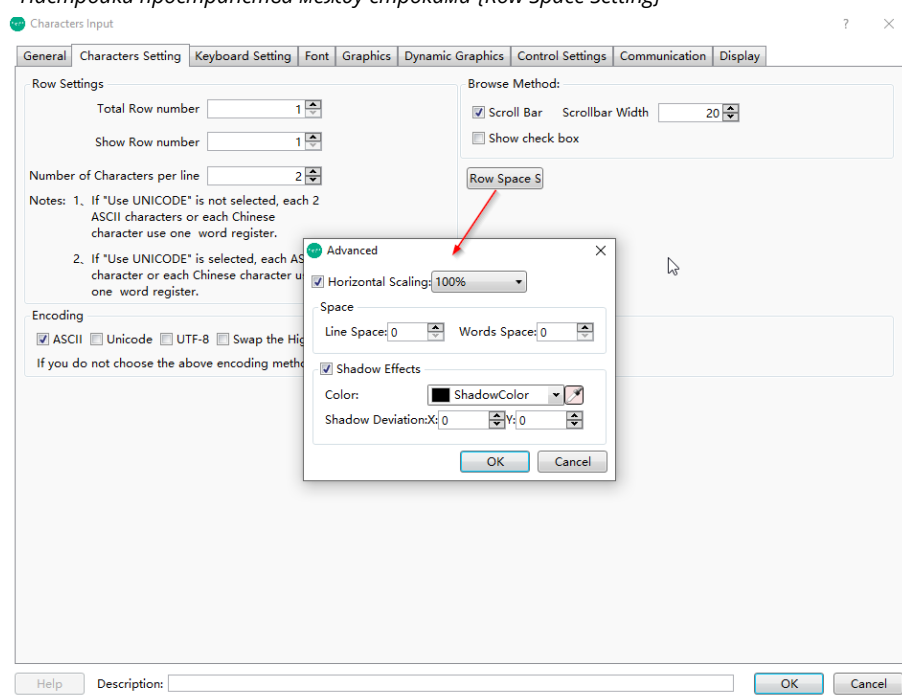
7. **Настройки строки {Row Settings}**

«Настройки строки {Row Setting}» включают «Общее количество строк {Total Row number}», «Показать номер строки (это количество строк для отображения) {Show Row number}» и «Количество символов в строке {Number of Characters per line}».

«Общее количество строк» означает количество строк, содержащихся в текущем символьном дисплее.

«Показать номер строки» устанавливает количество строк, отображаемых в текущем дисплее. Если количество отображаемых строк меньше, чем общее количество строк, полоса прокрутки появится автоматически.

8. **Настройки пространства между строками {Row Space Setting}**



Дополнительное диалоговое окно появится, если нажать кнопку «Настройка... {Row Space Setting}». Вы можете установить такие атрибуты, как «Горизонтальное масштабирование {Horizontal Scaling}», «Межстрочный интервал {Line Space}», «Область (интервал, отступ) между символами {Word Space}», «Теневые эффекты {Shadow Effect}».

Элементы остальных закладок «Шрифт {Font}», «Оформление (Графика) {Graphics}», «Анимация (Динамическая графика) {Dynamic Graphics}», «Обмен {Communication}», «Отображение {Display}» описаны в разделе «Рисование {Drawing}» т.к. имеют одинаковую структуру для всех закладок, где они используются.

Какие типы данных поддерживаются?

Версия ПО

Версия SystemeHMI Studio: 3.0.12512.0

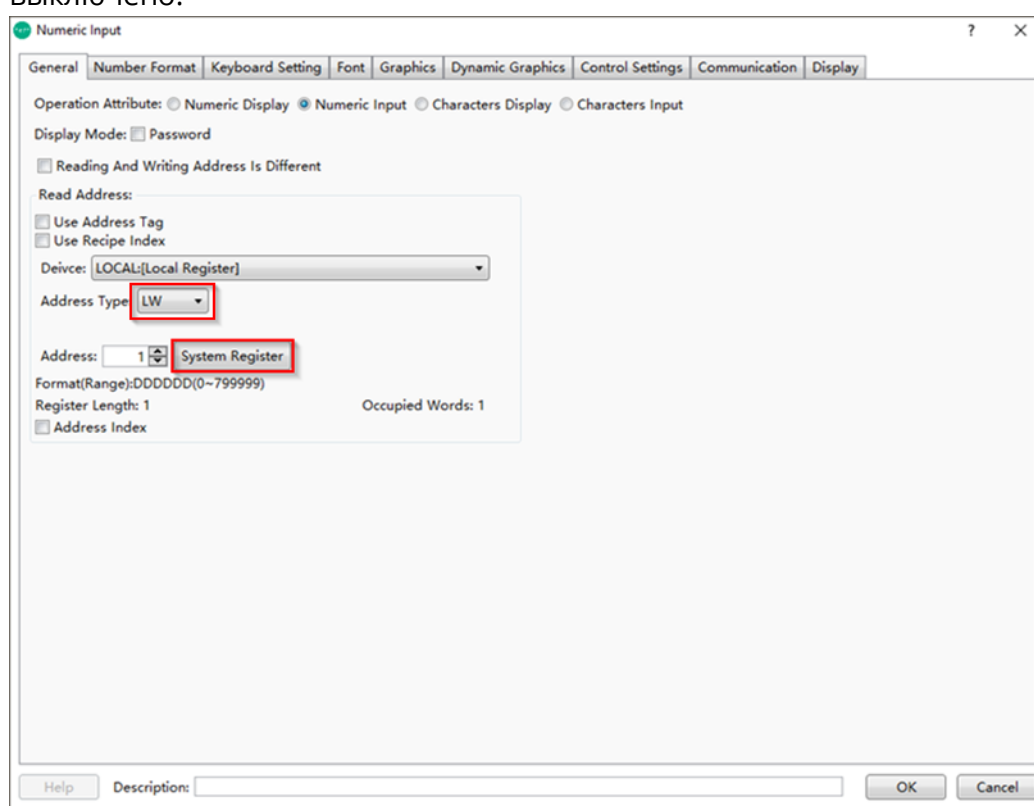
Версия ОС Windows: Windows 10 Pro, 21H2, сборка 19044.2604

Тип регистра HMI подразделяется на «Регистр слов» и «Битовый регистр».

Регистры слов:

LW: внутренний «Регистр слов» в HMI. *Данные теряются* при отключении питания. Диапазон адресов регистров: 0 - 799 999.

RW: Внутренний «Регистр слов» в HMI. *Данные могут быть сохранены*, если питание выключено.



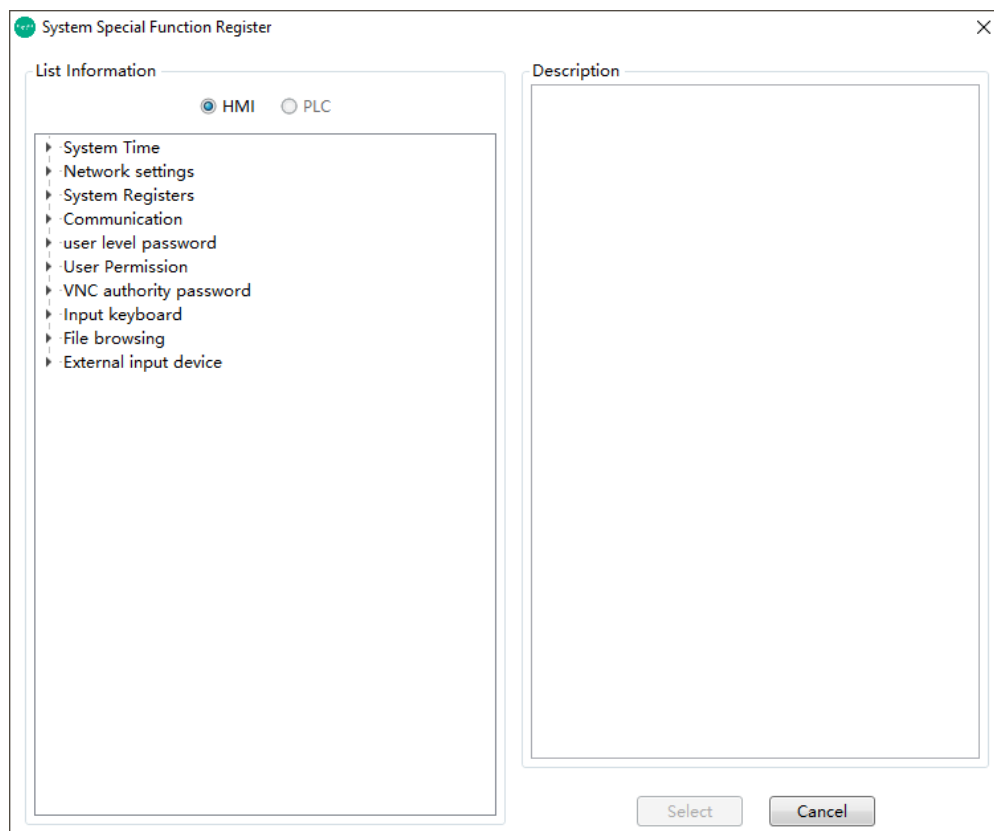
Диапазон

адресов регистров: 0–524288.

SRW: специальный внутренний «Регистр слов» в HMI. Диапазон адресов регистров: 0 - 11023.

Кнопка **System Register** (Системный регистр) открывает список **System Special Function Register** (Системный специальный Регистр функций), для выбора конкретной функции, закреплённой за этим регистром SRW. При работе с таким компонентом как например «Числовой дисплей», можно использовать системные регистры SRW0~7 , которые содержат системное время.

Поле **Description** (Описание) знакомит с функцией выбранного регистра.



Битовый регистр.

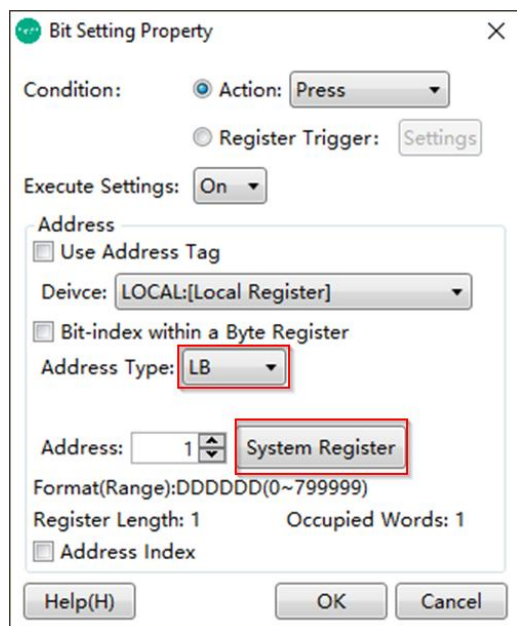
LB: внутренний «Битовый регистр» в HMI. *Данные теряются* при отключении питания. Диапазон адресов регистров 0 - 799,999.

SRB: специальный внутренний «Битовый регистр» в HMI.

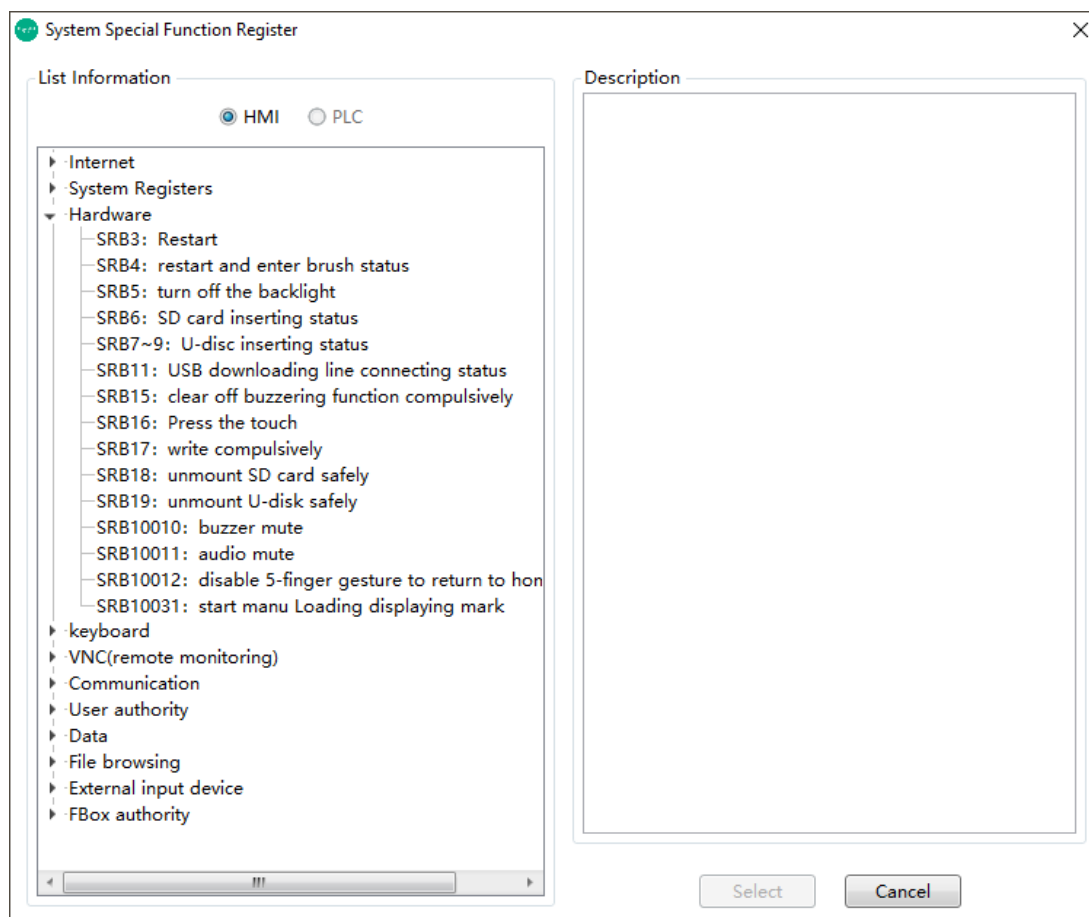
Диапазон адресов регистров: 0 -11023.

Кнопка **System Register** (Системный регистр) открывает список **System Special Function Register** (Системный Специальный Регистр функций), для выбора конкретной функции, закреплённой за этим регистром SRB. При работе с таким компонентом как например «Набор бит», можно использовать системный регистр SRW16 который изменяет своё состояние на «1» при нажатии на сенсорный экран.

Поле **Description** (Описание) знакомит с функцией выбранного регистра.



The 'Bit Setting Property' dialog box is shown. It has a title bar with a green icon and a close button. The 'Condition' section has two radio buttons: 'Action' (selected) and 'Register Trigger'. The 'Action' button has a dropdown menu showing 'Press'. The 'Register Trigger' button has a 'Settings' button next to it. The 'Execute Settings' section has a dropdown menu showing 'On'. The 'Address' section has a checkbox for 'Use Address Tag', a dropdown menu for 'Device' showing 'LOCAL:[Local Register]', a checkbox for 'Bit-index within a Byte Register', and a dropdown menu for 'Address Type' showing 'LB'. The 'Address' field has a value of '1' and a 'System Register' button next to it. The 'Format(Range):' field shows 'DDDDDD(0~799999)'. The 'Register Length' is '1' and 'Occupied Words' is '1'. There is an 'Address Index' checkbox. At the bottom are 'Help(H)', 'OK', and 'Cancel' buttons.



The 'System Special Function Register' dialog box is shown. It has a title bar with a green icon and a close button. The 'List Information' section has two radio buttons: 'HMI' (selected) and 'PLC'. Below them is a tree view showing a list of system registers. The 'Description' section is a large text area on the right. At the bottom are 'Select' and 'Cancel' buttons.

List Information

☒ HMI ☐ PLC

- Internet
- System Registers
- Hardware
 - SRB3: Restart
 - SRB4: restart and enter brush status
 - SRB5: turn off the backlight
 - SRB6: SD card inserting status
 - SRB7~9: U-disc inserting status
 - SRB11: USB downloading line connecting status
 - SRB15: clear off buzzing function compulsively
 - SRB16: Press the touch
 - SRB17: write compulsively
 - SRB18: unmount SD card safely
 - SRB19: unmount U-disk safely
 - SRB10010: buzzer mute
 - SRB10011: audio mute
 - SRB10012: disable 5-finger gesture to return to hon
 - SRB10031: start manu Loading displaying mark
- keyboard
- VNC(remote monitoring)
- Communication
- User authority
- Data
- File browsing
- External input device
- FBox authority

Description

Select Cancel

Примечание. «Регистр слов» и «Битовый регистр» в HMI — это две разные области, поэтому адреса не пересекается. Например, LW0 и LB0 — это два регистра, которые не являются

связанный. SRW0 и SRB0 также являются двумя разными специальными системными регистрами. Но каждый «Регистр слов» можно разделить на 16 «Битовых регистры». Например, LW0 можно разделить на 16-битовых регистров : LW0.0 ~ 0,15.

Как настроить связь между панелью HMISGU*** и ПЛК SystemePLC S250

Версия ПО

Версия Systeme HMI Studio: 3.0.12512.0

Версия ОС Windows: Windows 10 PRO v1903

Панели Systeme HMI подразделяются на следующие типы

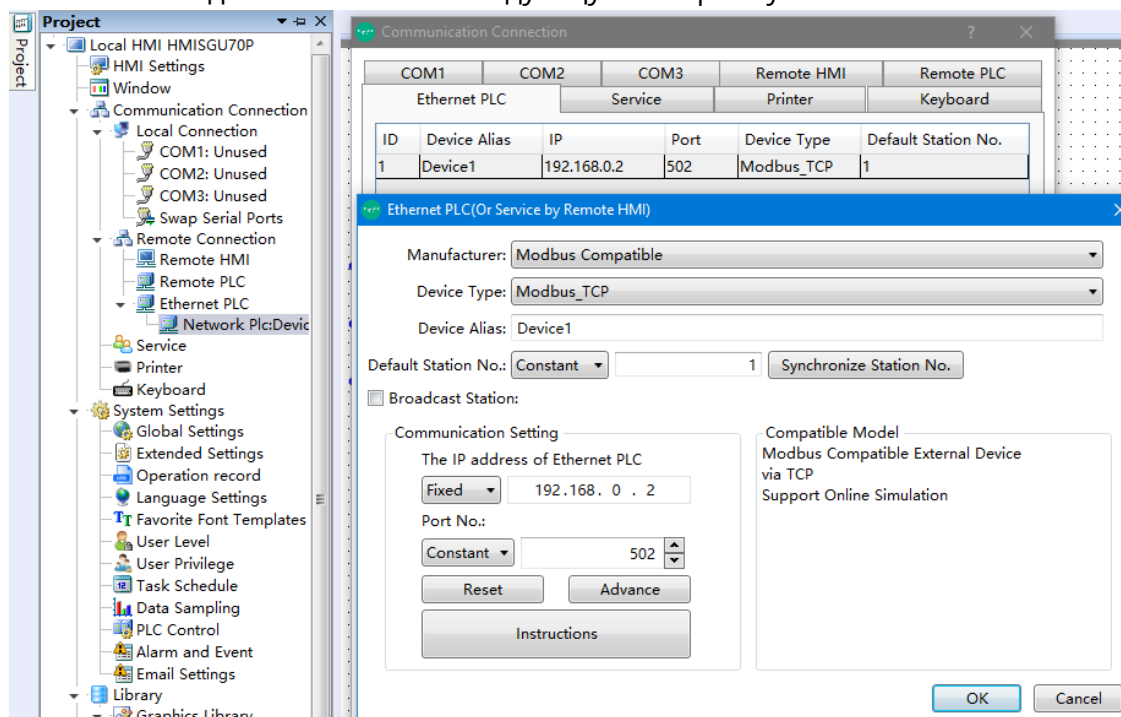
Common HMI: HMISGU43P HMISGU70P, HMISGU70PE, HMISGU101P, HMISGU101PE, HMISGU101ME

Advanced HMI: HMISGU156ME HMISGU70PEA, HMISGU101MEA

В зависимости от типа применяемой панели подключение к ПЛК S250 может выполняться из этих панелей двумя возможными способами

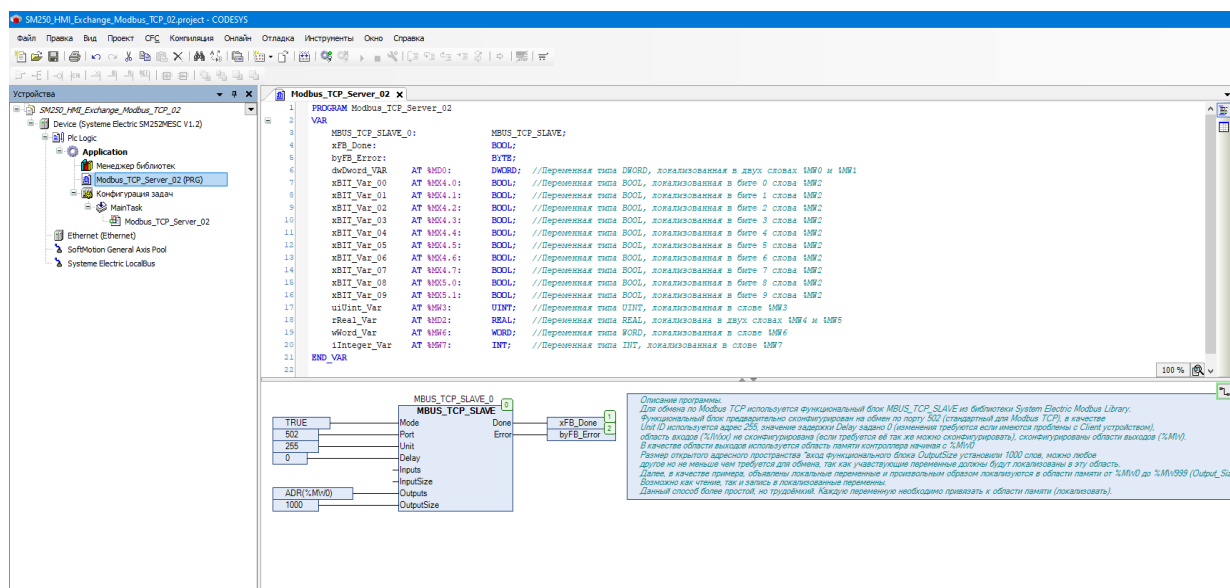
- Связь по открытым коммуникационным протоколам Modbus (последовательный порт) или Modbus TCP (Ethernet);
- Связь по именам переменным проекта ПЛК.

Так для передачи данных первого типа а. в режиме связи по протоколу Modbus в HMI Studio необходимо выполнить следующую настройку:

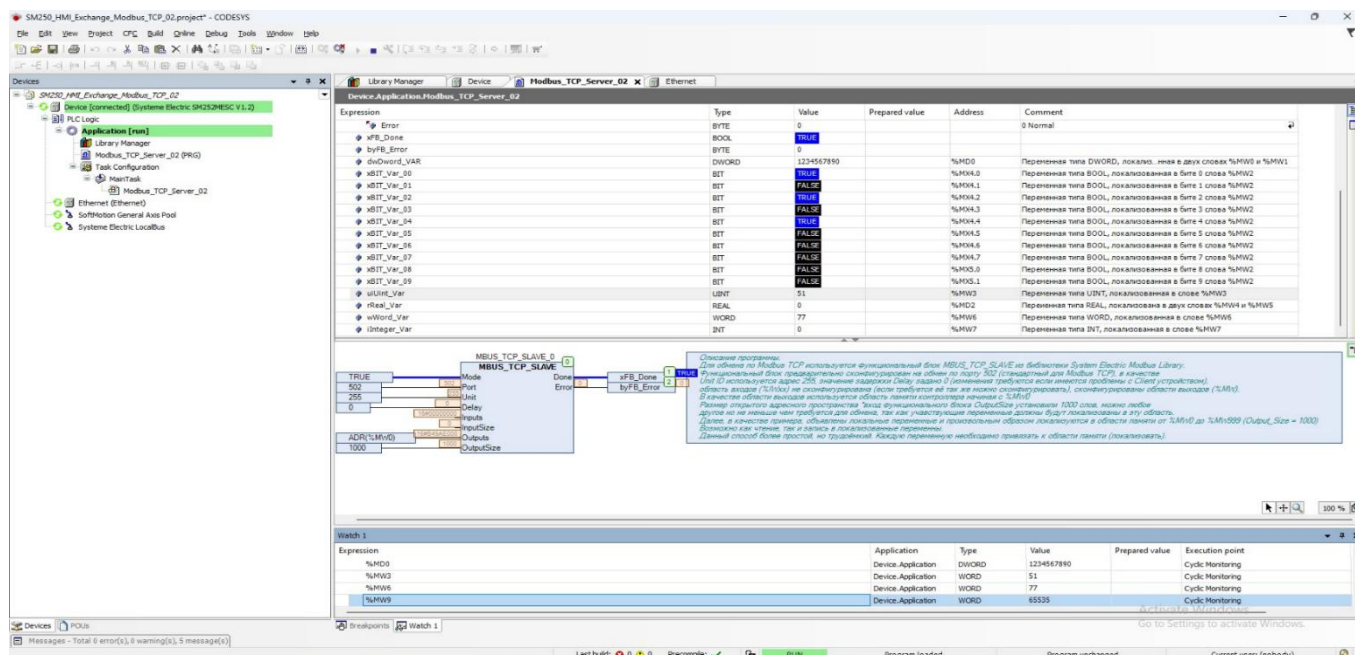


Если применяются панели типа Common HMI или по каким-то причинам требуется обеспечить передачу данных через протокол Modbus TCP. В этом случае придется делать переменные в проекте, занося каждую переменную в проект контроллера и панели. В этом случае в проекте панели это делается автоматически при создании переменной, но для контроллера нужно выполнять настройку функционального блока, обеспечивающего работу в режиме протокола Modbus TCP. Для протокола Modbus TCP контроллер S250 может работать как в режиме Client (Master), так и Server (Slave). В качестве клиента его можно настроить используя или стандартный драйвер CODESYS

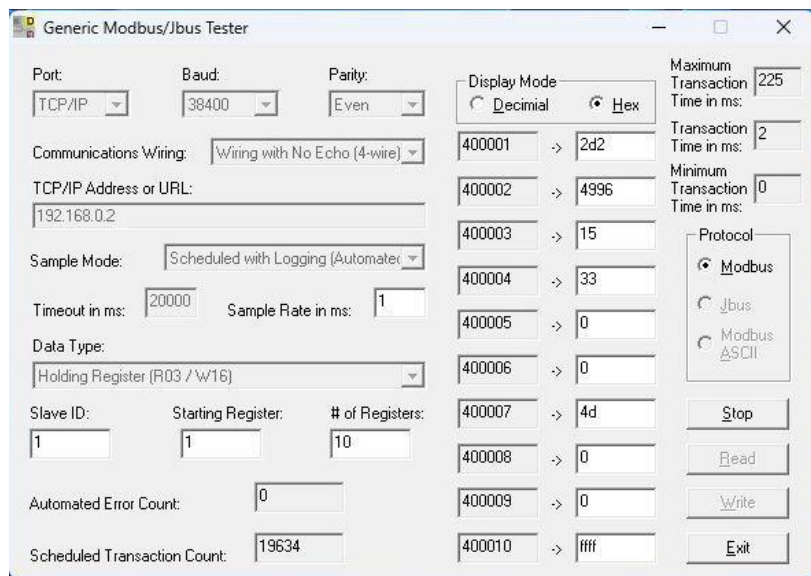
(аналог IOScanner в Schneider Electric), так и используя функциональные блоки из библиотеки Sysytem Electric Modbus. Что касается настройки контроллера в качестве Server, то в настоящее время доступно использовать только функциональные блоки из вышеуказанной библиотеки. В проекте можно использовать только один экземпляр блока. Ниже копия экрана с примером настройки такого блока:



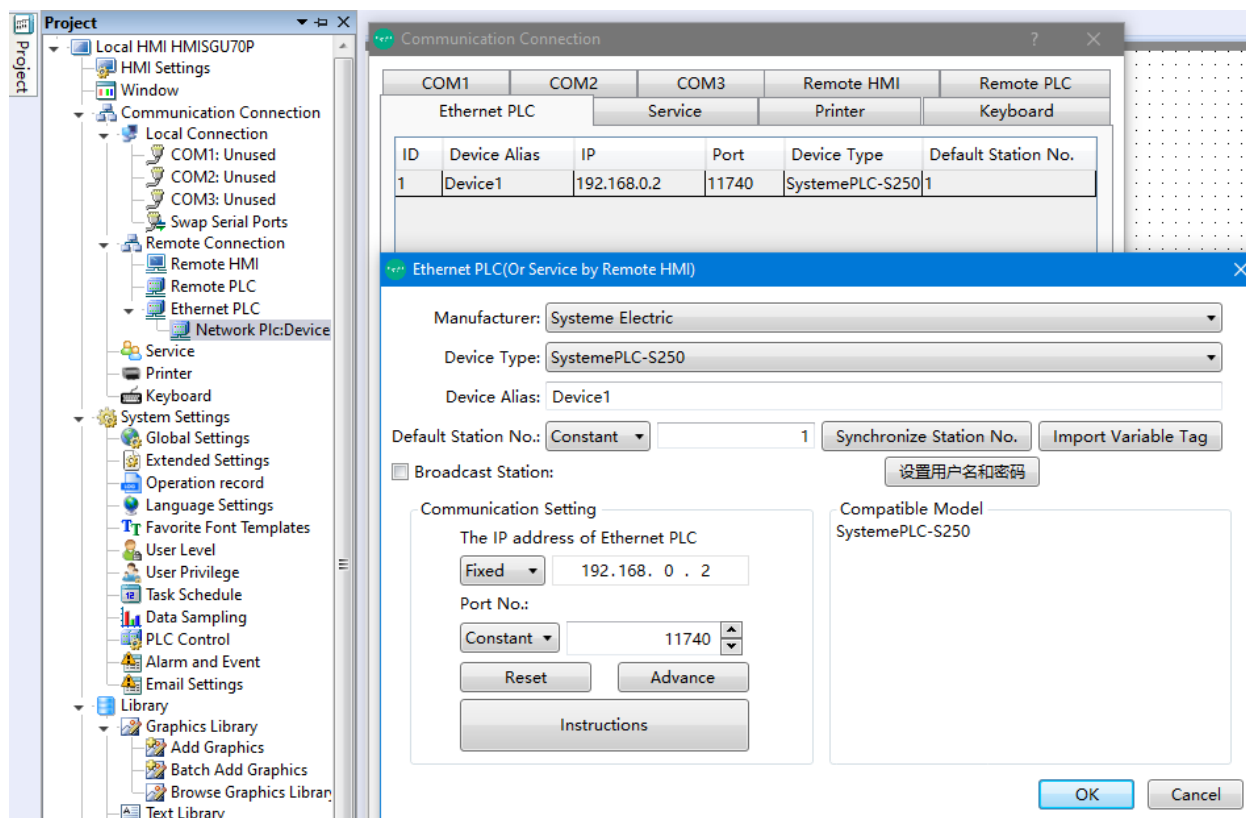
Ниже копия экрана работы такого блока с подключением к ПЛК с отображением текущих данных из памяти контроллера для описанных в проекте переменных %MW0-%MW8 и неописанной в проекте ячейки памяти %mw9:



Ниже вид программы сканнера данных протокола MODBUS для 10 слов памяти контроллера

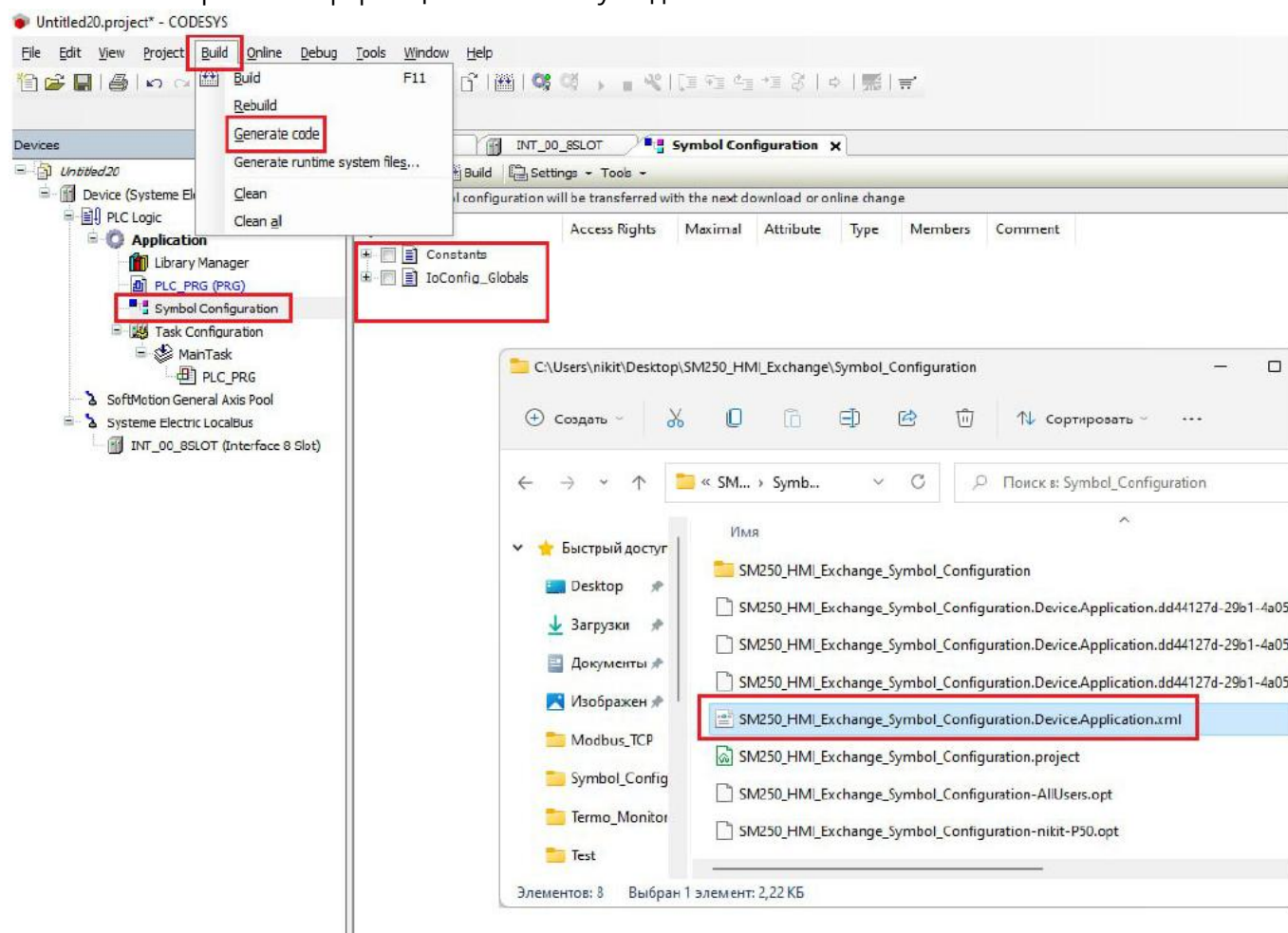


В случае, если используется панель типа b. Advanced HMI, то в этом случае можно настроить передачу данных через переменные проекта. При этом в проекте можно выбрать Device Type: SystemePLC-S250 или протокол CODESYS_V3_TCP.

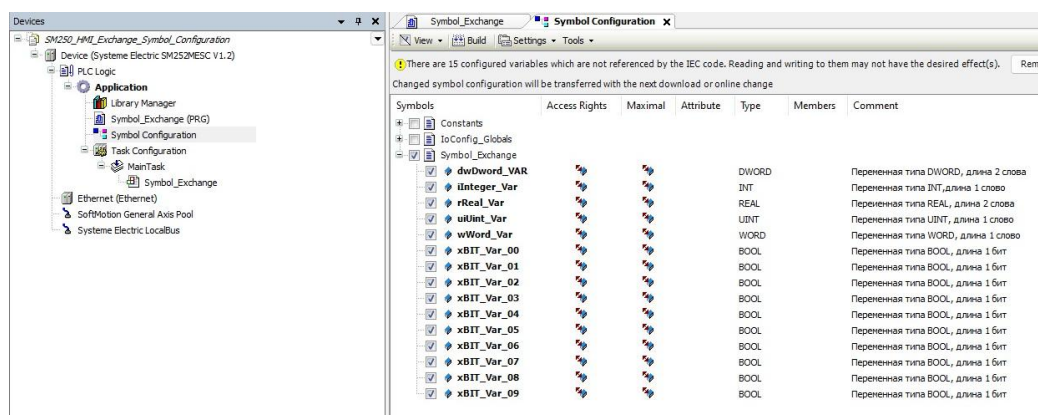


Немного по обмену данными через переменные проекта режим SystemePLC-S250 или режиму CODESYS_V3_TCP. Для организации обмена в проекте контроллера необходимо создать в дереве проекта в разделе Symbol Configuration таблицу переменных доступ, к которым необходим по именам. Туда нужно добавить из проекта все необходимые переменные для обмена и сохранить этот список в виде xml файла. Этот файл всегда будет создаваться при загрузке проекта или его можно сформировать командой из меню CODESYS: Build -> Generate code.

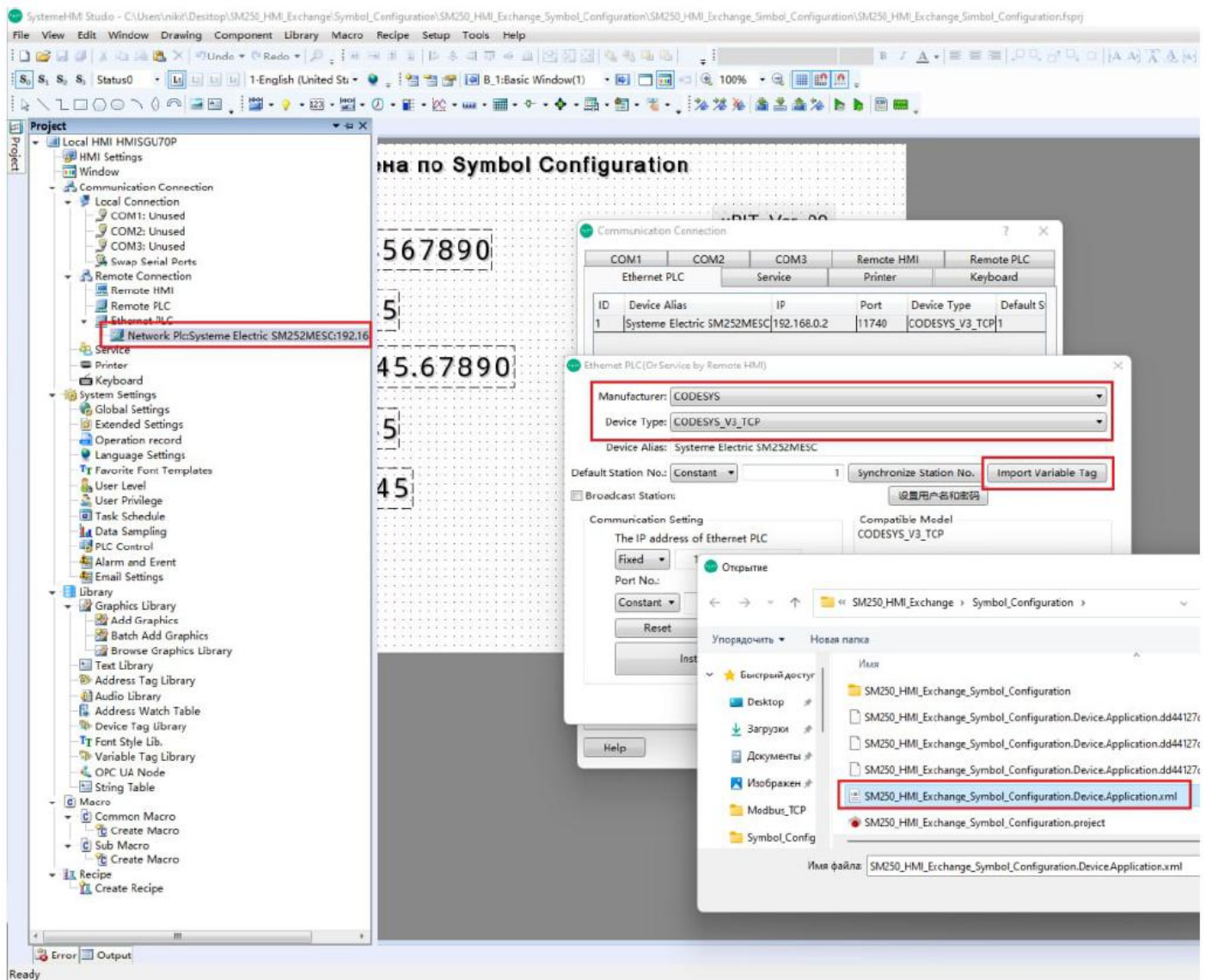
Ниже копии экрана с информацией по такому подключению:



Далее, этот файл будет необходим в HMI Studio для импорта выбранных селектором переменных в проект.



Это выполняется как показано ниже:



Далее, в проекте контроллера будут доступны все переменные из раздела Symbol Configuration проекта контроллера настраиваемого в CODESYS.

Как настроить связь между панелью HMISGU*** и ПЛК SystemePLC S250 по Modbus

Версия ПО

Версия Systeme HMI Studio: 3.0.12512.0 Версия ОС Windows: Windows 10 PRO v1903

Панели Systeme HMI подразделяются на следующие типы

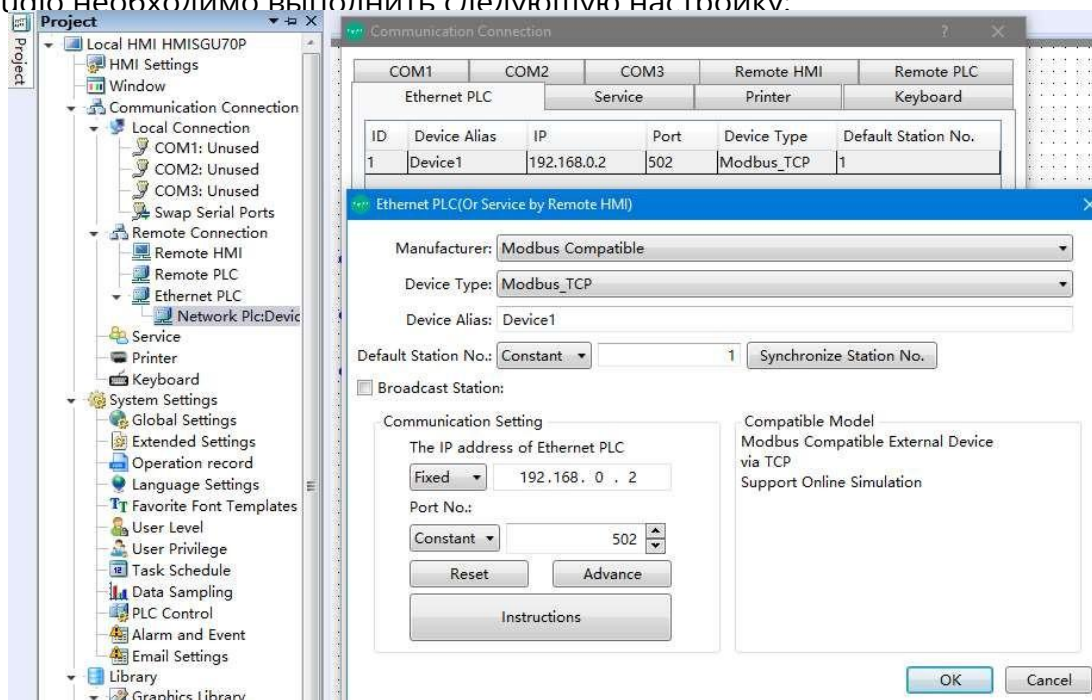
Common HMI: HMISGU43P HMISGU70P, HMISGU70PE, HMISGU101P, HMISGU101PE, HMISGU101ME

Advanced HMI: HMISGU156ME HMISGU70PEA, HMISGU101MEA

В зависимости от типа применяемой панели подключение к ПЛК S250 может выполняться из этих панелей двумя возможными способами

- Связь по открытым коммуникационным протоколам Modbus (последовательный порт) или Modbus TCP (Ethernet);
- Связь по именам переменным проекта ПЛК.

Так для передачи данных первого типа а. в режиме связи по протоколу Modbus в HMI Studio необходимо выполнить следующую настройку:



Если применяются панели типа Common HMI или по каким-то причинам требуется обеспечить передачу данных через протокол Modbus TCP. В этом случае придется делать переменные в проекте, занося каждую переменную в проект контроллера и панели. В этом случае в проекте панели это делается автоматически при создании переменной, но для контроллера нужно выполнять настройку функционального блока, обеспечивающего работу в режиме протокола Modbus TCP. Для протокола Modbus TCP контроллер S250 может работать как в режиме Client (Master), так и Server (Slave). В качестве клиента его можно настроить используя или стандартный драйвер CODESYS (аналог IOScanner в Schneider Electric), так и используя функциональные блоки из библиотеки Sysytem Electric Modbus. Что касается настройки контроллера в качестве Server, то в настоящее время доступно использовать только функциональные блоки из вышеуказанной библиотеки. В проекте можно использовать только один

Modbus_TCP_Server_02

```

PROGRAM Modbus_TCP_Server_02
VAR
    MBUS_TCP_SLAVE_0:      MBUS_TCP_SLAVE;
    xFB_Done:               BOOL;
    byFB_Error:             BYTE;
    dwDword_Var:            AT %MD0: DWORD; //Переменная типа DWORD, локализованная в двух словах %MW0 и %MW1
    xBIT_Var_00:            AT %MX4.0: BOOL; //Переменная типа BOOL, локализованная в бите 0 слова %MW2
    xBIT_Var_01:            AT %MX4.1: BOOL; //Переменная типа BOOL, локализованная в бите 1 слова %MW2
    xBIT_Var_02:            AT %MX4.2: BOOL; //Переменная типа BOOL, локализованная в бите 2 слова %MW2
    xBIT_Var_03:            AT %MX4.3: BOOL; //Переменная типа BOOL, локализованная в бите 3 слова %MW2
    xBIT_Var_04:            AT %MX4.4: BOOL; //Переменная типа BOOL, локализованная в бите 4 слова %MW2
    xBIT_Var_05:            AT %MX4.5: BOOL; //Переменная типа BOOL, локализованная в бите 5 слова %MW2
    xBIT_Var_06:            AT %MX4.6: BOOL; //Переменная типа BOOL, локализованная в бите 6 слова %MW2
    xBIT_Var_07:            AT %MX4.7: BOOL; //Переменная типа BOOL, локализованная в бите 7 слова %MW2
    xBIT_Var_08:            AT %MX5.0: BOOL; //Переменная типа BOOL, локализованная в бите 8 слова %MW2
    xBIT_Var_09:            AT %MX5.1: BOOL; //Переменная типа BOOL, локализованная в бите 9 слова %MW2
    uiUnit_Var:             AT %MD3: UINT; //Переменная типа UINT, локализованная в слове %MW3
    rReal_Var:              AT %MD2: REAL; //Переменная типа REAL, локализована в двух словах %MW4 и %MW5
    wWord_Var:              AT %MW6: WORD; //Переменная типа WORD, локализованная в слове %MW6
    iInteger_Var:           AT %MW7: INT; //Переменная типа INT, локализованная в слове %MW7
END_VAR

```

MBUS_TCP_SLAVE_0

Mode: TRUE
Port: 502
Unit: 255
Delay: 0
Inputs: ADR(%MW0)
OutputSize: 1000

Done: xFB_Done
Error: byFB_Error

Описание программы:
Для обмена по Modbus TCP используется функциональный блок MBUS_TCP_SLAVE из библиотеки System Electric Modbus Library. Функциональный блок предварительно сконфигурирован на обмен по порту 502 (стандартный для Modbus TCP), в качестве Unit (ID) используется адрес 255, значение задержки Delay задано 0 (изменения требуются если имеются проблемы с Client устройством). Область входов (%MW0) не сконфигурирована (если требуется её так же можно сконфигурировать), сконфигурированы области выходов (%MW). В качестве области выходов используется область памяти контроллера начиная с %MW0. Размер открытого адресного пространства "вход функционального блока OutputSize установили 1000 слов, можно любое другое но не меньше чем требуется для обмена, так как участвующие переменные должны будут локализованы в эту область. Далее, в качестве примера, объявлены локальные переменные и произвольным образом локализованы в области памяти от %MW0 до %MW999 (Output_Size). Возможно как чтение, так и запись в локализованные переменные. Данный способ более простой, но трудоемкий. Каждую переменную необходимо привязать к области памяти (локализовать).

Ниже копия экрана работы такого блока с подключением к ПЛК с отображением текущих данных из памяти контроллера для описанных в проекте переменных %MW0-%MW8 и неописанной в проекте ячейки памяти %mw9:

The screenshot displays the SIMATIC Manager interface for a Modbus TCP Slave configuration. The top window shows the 'Device: Application: Modbus_TCP_Server_02' with a list of variables and their values. Below this, a ladder logic diagram shows the connection to the Modbus TCP Slave. The bottom window shows the 'Watch 1' table with the following data:

Expression	Application	Type	Value	Prepared value	Execution point
%MD0	Device: Application	DWORD	1234567890		Cyclic Monitoring
%MW0	Device: Application	WORD	51		Cyclic Monitoring
%MW6	Device: Application	WORD	77		Cyclic Monitoring
%MW9	Device: Application	WORD	65535		Cyclic Monitoring

Ниже вид программы сканнера данных протокола MODBUS для 10 слов памяти контроллера

The screenshot shows the 'Generic Modbus/Jbus Tester' application. The configuration settings are as follows:

- Port: TCP/IP
- Baud: 38400
- Parity: Even
- Display Mode: Hex
- Communications Wiring: Wiring with No Echo (4-wire)
- TCP/IP Address or URL: 192.168.0.2
- Sample Mode: Scheduled with Logging (Automated)
- Timeout in ms: 20000
- Sample Rate in ms: 1
- Data Type: Holding Register (R03 / W16)
- Slave ID: 1
- Starting Register: 1
- # of Registers: 10
- Automated Error Count: 0
- Scheduled Transaction Count: 19634

The right side of the window shows the transaction history and the current transaction details:

- Maximum Transaction Time in ms: 225
- Transaction Time in ms: 2
- Minimum Transaction Time in ms: 0
- Protocol: Modbus
- Buttons: Stop, Read, Write, Exit