



Polymath. Режим Double Click. Описание.

Содержание

| | |
|---|----|
| 1. Общие сведения | 3 |
| 2. Создание проекта | 4 |
| 3. Создание переменных (Tag) | 6 |
| 4. Изменение подключенных устройств и параметров связи | 14 |
| 5. Системные сообщения, функциональные клавиши, языки и шрифты. | 17 |
| 6. Экраны (Pages) | 19 |
| 7. Стандартные объекты | 22 |
| 8. События и Аварии (Alarm) | 28 |
| 9. Рецепты (Recipe) | 36 |
| 10. Уровни доступа и пароли (Users and Passwords) | 39 |
| 11. Скрипты (Script) | 42 |
| 12. Списки (List) | 43 |
| 13. Подключение панели к компьютеру по Ethernet | 46 |
| 14. Загрузки проекта в терминал с помощью USB-накопителя | 49 |

Общие сведения

Существуют три версии ПО для различных типов панелей:

а) Polymath Smart Click – бесплатный для панелей серии **SC (3,5, 7 и 10 дюймов)** – бюджетная серия с ограничениями:

1. максимального числа экранов 100 и 10 всплывающих;
2. кол-во переменных (Тегов) – 800;
3. возможность подключения не более 6 устройств к одному порту Ethernet;
4. Ограниченное число рецептов – 10;
5. Максимум 15 скриптов и не более 2000 символов;
6. Максимум 8 языков в проекте;
7. Макс. Кол-во графиков трендов – 10.
8. Не более 20 пользователей на проект.
9. Нет косвенной адресации переменных.
10. Нет сетевых переменных (для обмена данными между панелями и ПК).
11. Нет объектов Cronotherm и Scheduler (Расписания).
12. Нет отправки SMS и E-mail в случае возникновения аварии или события.

Бесплатную версию софта можно скачать с сайта ESAHMI.COM

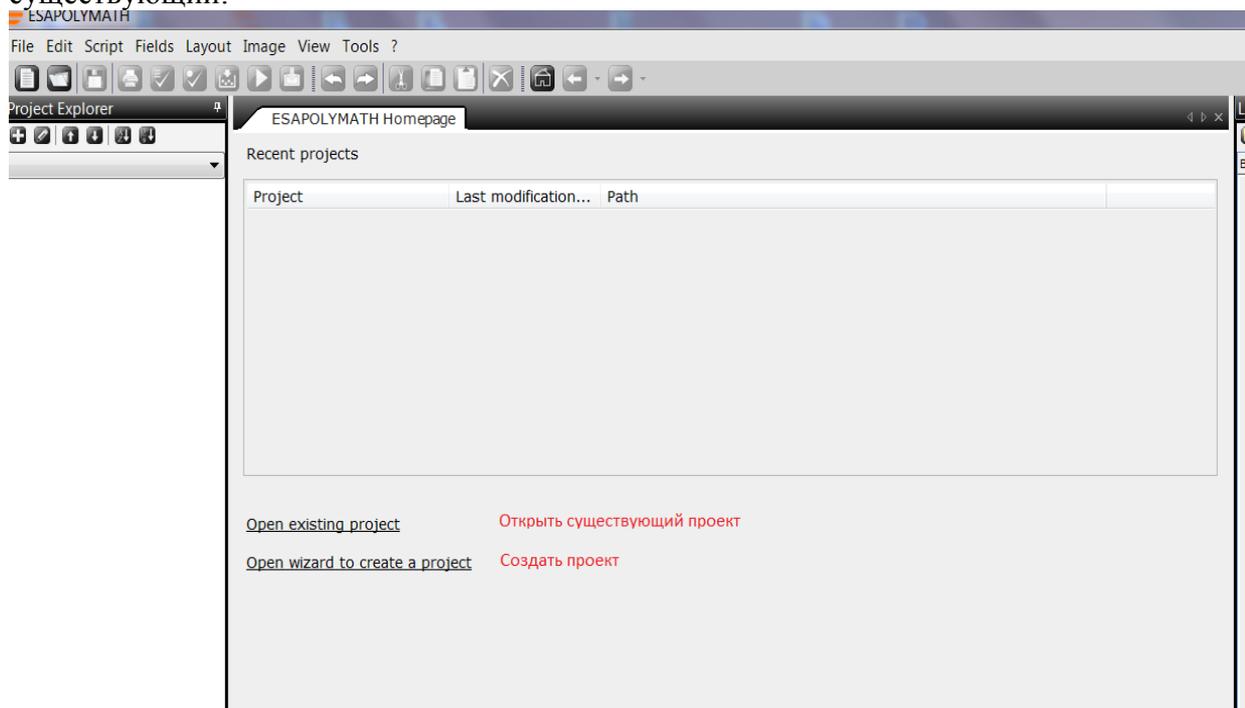
б) Polymath HMI – для создания проектов для панелей SC, VT и IT.

с) Polymath Advanced – для создания проектов для панелей SC, VT, IT и проектов исполняемых на компьютерах.

Две последних версии имеют два режима работы с проектом Double Click (эквивалентен Smart Click) и Extended (свойства объектов отображаются в виде таблиц, а не всплывающих окон).

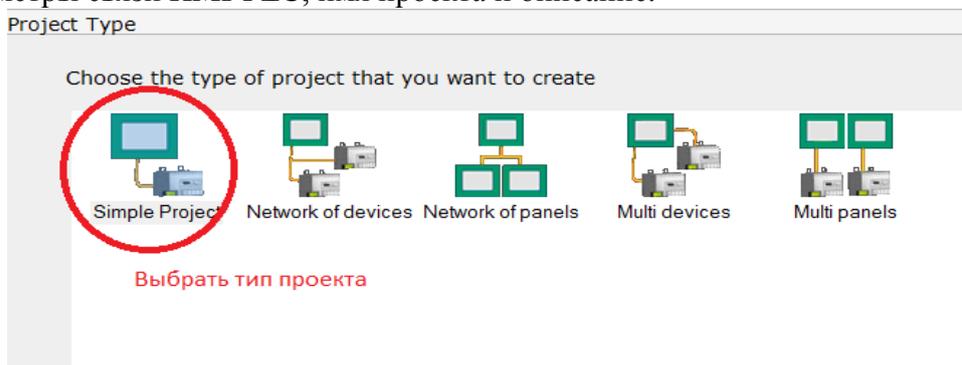
Создание проекта.

При запуске **Polymath** предложит либо создать новый проект, либо загрузить уже существующий:

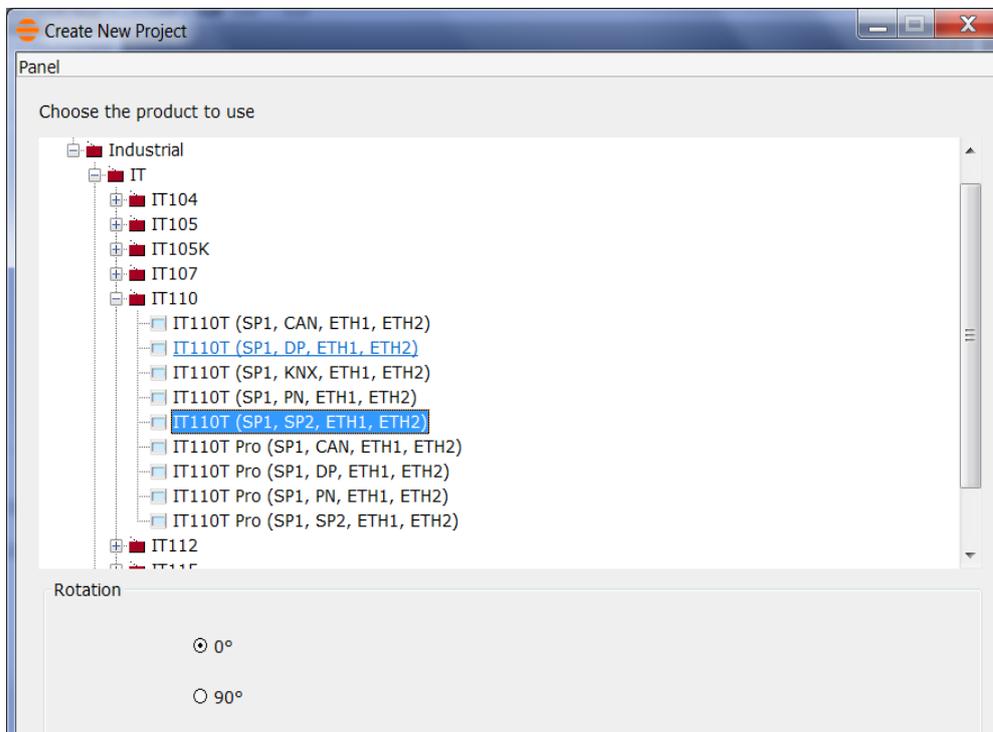


Общие рекомендации: при создании нового проекта обязательно позаботьтесь о том что бы иметь доступ в системное меню терминала. Это достигается путем установки кнопки под паролем на каком-либо экране, осуществляющей переход в системное меню. Подробно об этом в разделе **Create a push button (cmp.46)**

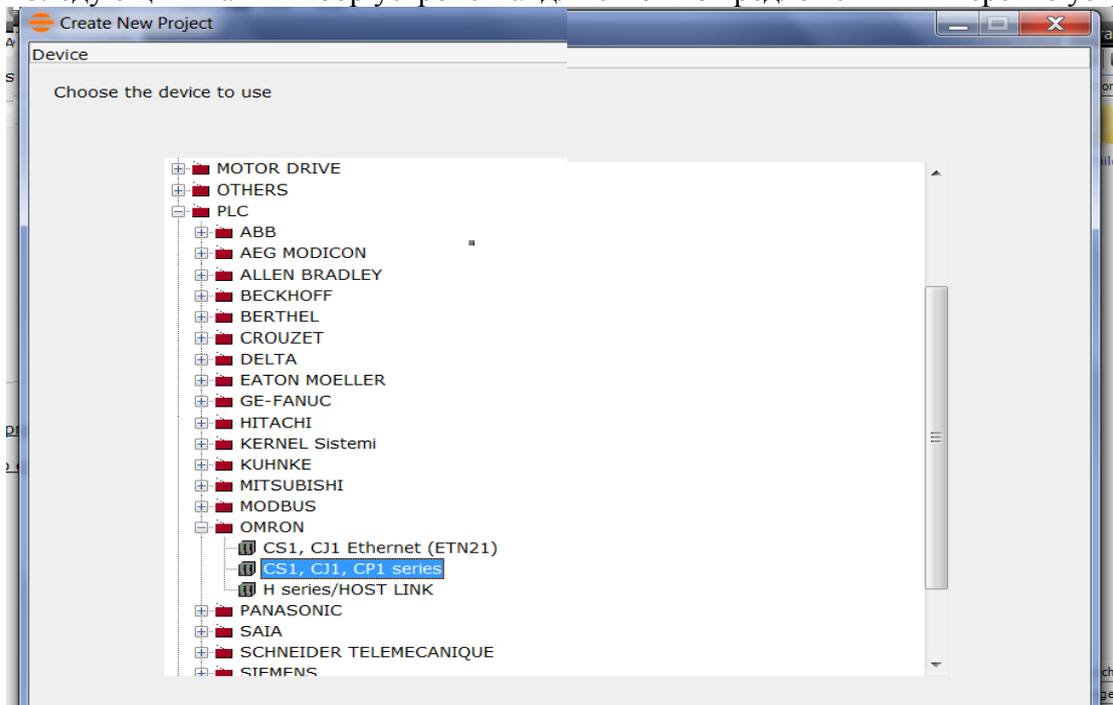
При создании нового проекта появится окно, в котором нужно будет выбрать параметры связи НМІ-PLC, имя проекта и описание.



Далее следует выбрать модель НМІ в зависимости от размеров экрана и конфигурации портов, а также, расположение проекта относительно терминала: большая сторона терминала расположена либо горизонтально, либо вертикально.



Следующий шаг – выбор устройства для связи из предложенных в перечне устройств:



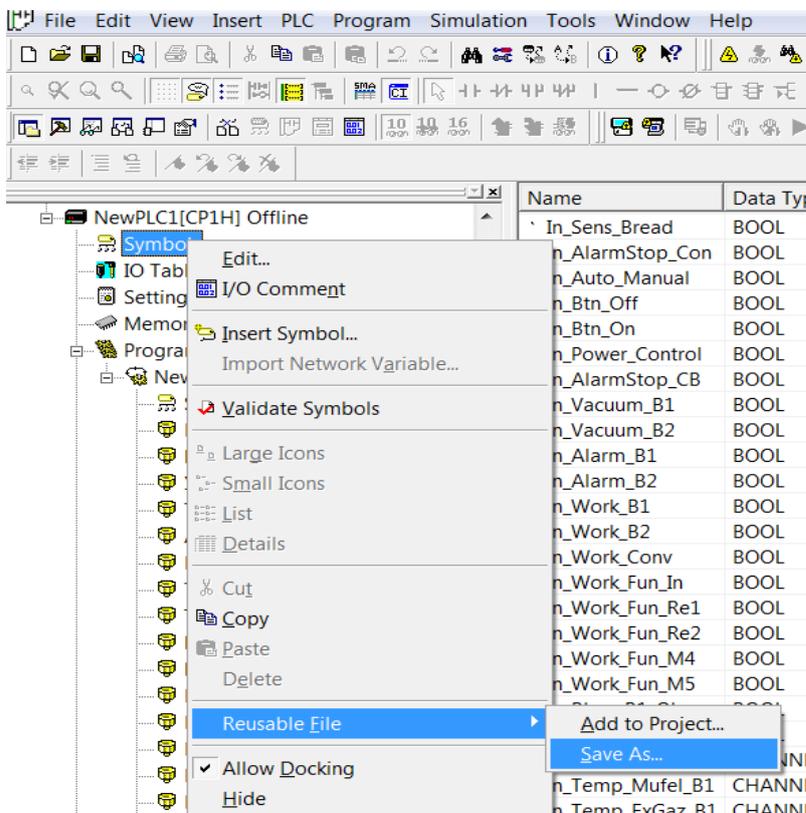
Создание переменных.

При программировании панелей ESA используются символьные переменные связанные с адресами (регистрами) подключаемого устройства. В этом заключается ряд преимуществ: например, легко изменить адрес переменной, возможно присваивать один и тот же адрес разным переменным.

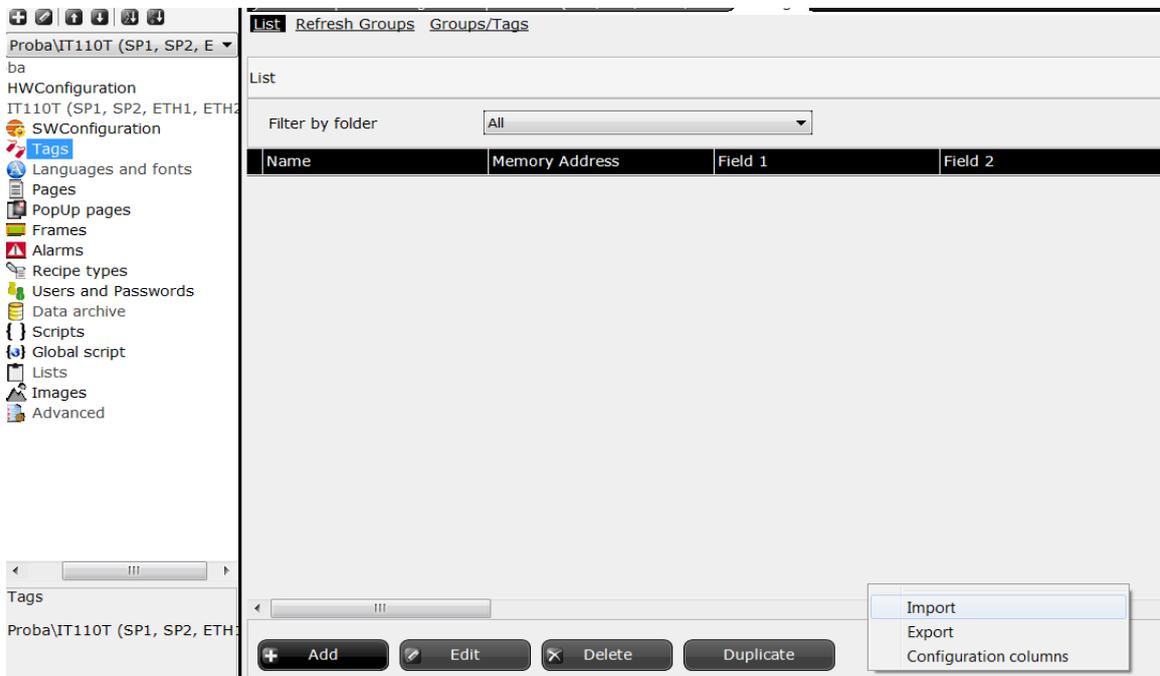
Если при создании проекта вы выбрали устройства Siemens, Vipa, Omron, после этого появится окно, предлагающее загрузить символьную таблицу переменных из проекта для контроллера контроллера, что существенно экономит время на создание проекта HMI:



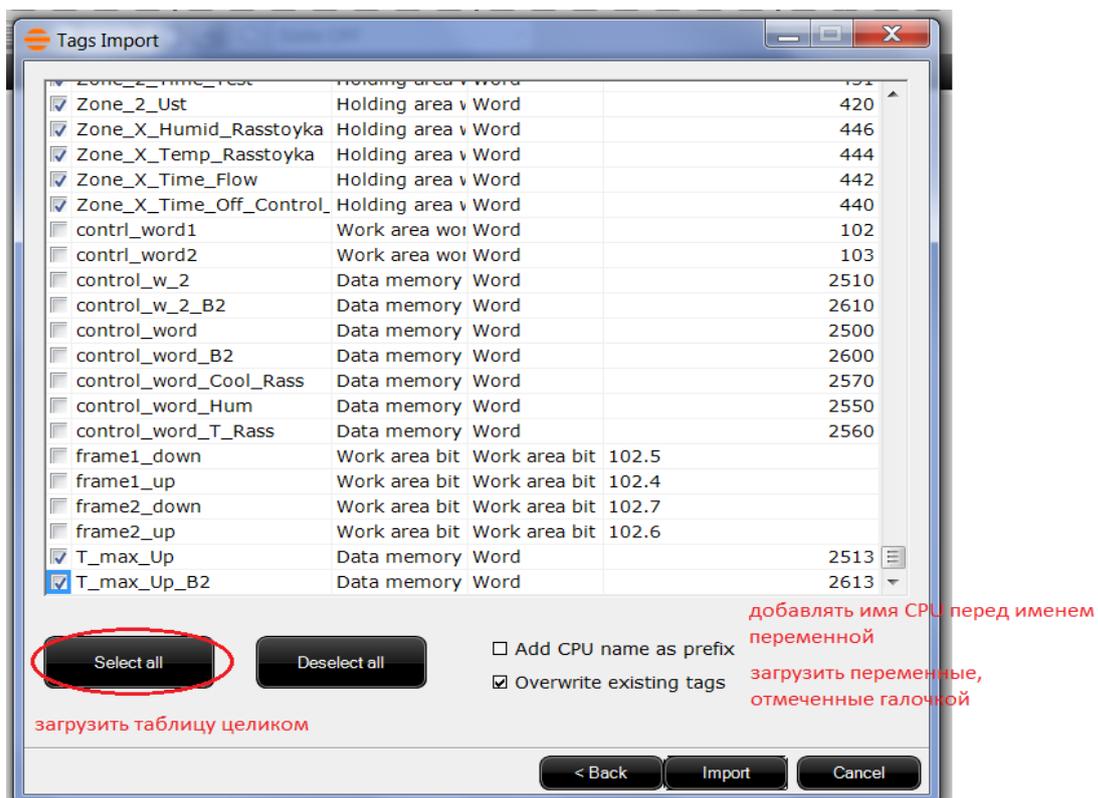
Для создания файла переменных следует в проекте **CX-PROGRAMMER** открыть таблицу символьных переменных и по правой клавише мыши создать файл с символьными переменными, которые возможно импортировать в Polymath.



Символьные переменные можно загрузить так же в уже существующий проект, щелкнув левой клавишей мыши в дереве проекта с левой стороны на опцию «Tags», далее с низу экрана нажав на опцию «Tools» и «Import»:



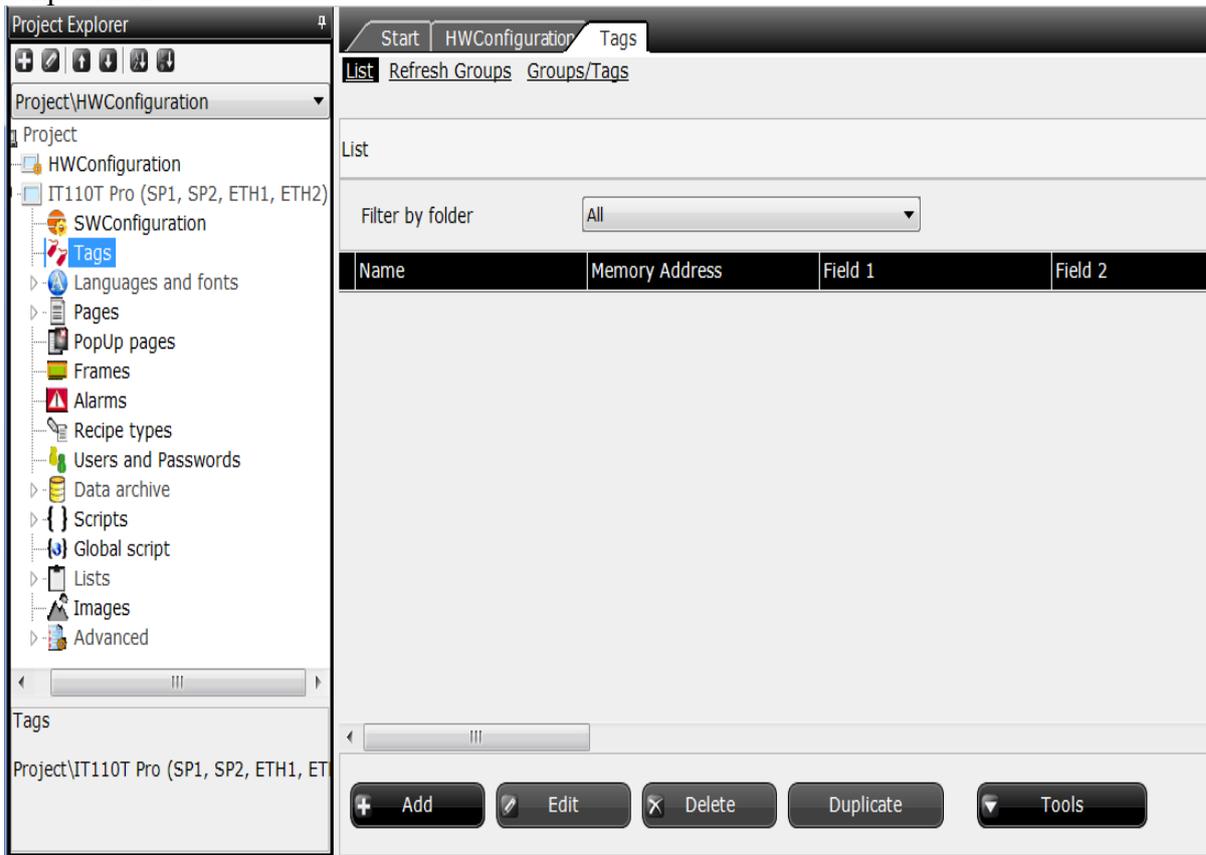
После указания пути к файлу с символьными переменными предлагается загрузить либо всю таблицу целиком, либо с помощью галочек отметить загружаемые переменные:



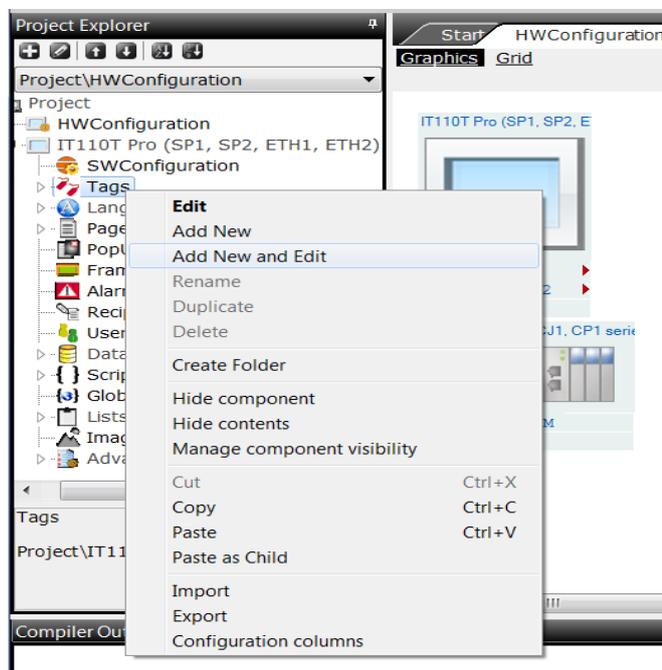
Так же можно загружать переменные из файлов Excel.

*Не используемые в программе переменные впоследствии можно будет удалить через **Tools/Utilities/Remove unused tags** (Инструменты/утилиты/удалить не используемые теги) в главном меню.*

Для создания переменных вручную либо щелкните два раза ЛКМ по закладке **TAGS** в дереве проектов:



а затем **Add** и для редактирования переменной **Edit**.
Или наведя курсором на закладку **Tags**, по ПКМ выбрать **Add New and Edit**:



Окно редактирования переменной:

Start HWConfiguration Имя_переменной

General Value Device Limits Conversion Thresholds Indirect Address Events

Identification

Name Имя_переменной

Comment

Address

Type Device

- Device
- Internal
- System
- Indexed
- Network

Закладка **General** (общий)

Name – имя переменной – должно быть уникальным. Не должно содержать пробелов, может быть написано и на кириллице, но если в дальнейшем планируется использовать переменную в скриптах, то необходимо использовать латиницу.

Type – тип переменной по принадлежности к подключаемому устройству (ПЛК, регулятору и т.д), внутренняя переменная панели, системная, с косвенной адресацией и сетевая. Два последних типа переменных не доступны в панелях серии SC.

Системные переменные содержат полезные данные: часы и календарь, состояние коммуникационных портов, страниц и объектов используемых в проекте.

Если адрес переменной – контроллер или иное устройство, отличное от панели, то выбираем **Device**.

Закладка **Value** тип переменной

Start HWConfiguration Имя_переменной

General Value Device Limits Conversion Thresholds Indirect Address Events

Tag Type

Type Integer

- Char
- Byte
- Boolean
- Integer
- Unsigned Integer
- Long
- Unsigned Long
- Real
- Double
- String

Array

Char (Символ) – символьная переменная;

Byte – байт (8бит);

Boolean (Булево значение) – бит;

Integer (Целое) – целое число со знаком 16 бит;

Unsigned Integer (Беззнаковое целое) – целое число 16 бит без знака;

Long (Длинное целое) – целое число со знаком 64 бита;

Unsigned Long (Беззнаковое длинное целое)– целое число без знака 64 бита;

Real (Вещественное) – число с плавающей запятой 32 бита;

Double (Двойной точности) – целое число со знаком 32 бита;

String (Строка) – символьная строка.

Закладка **Device**

Start HWConfiguration Имя_переменной

General Value **Device** Limits Conversion Thresholds Indirect Address Events

Memory Address

Memory Address CS1, C11, CP1 series

Refresh group Class_0_5: 500 msec

Update Device enabled

Update always, even when a tag isn't used by any field

Read only

Data Area

Work area word (W) Type Word

Signed

Address

W 0

- **Refresh group (обновить группу)** – переменные в ESA обновляются с определенной периодичностью. Для того оптимизации загрузки процессора переменные ранжируют по частоте опроса.
Из опыта работы максимальная скорость обмена по Rs-232/485 – 0,2-0,5 сек;
- **Update always, even when a tag isn't used by any field** – производить обновление переменной даже в случае, если эта переменная не используется в текущем окне;
- **Read only** – только чтение переменной (запрет изменение значения с терминала);
- **Data aria (область данных):** в первом поле выбирается область памяти переменной, присвоенной в контроллере (или ином устройстве) в соответствии с типом (бит или слово), во втором поле **Type** подтверждается тип этой переменной в соответствии с типом переменной в контроллере;
- **Signed-** галочка означает, что переменная в контроллере - со знаком;
- **Address-** в этом поле устанавливается адрес области памяти (слово и номер бита для битовых переменных).

Закладка **Limits** вводит ограничение на максимальное и минимальное значение при вводе с терминала **Input limit on Tag Value (входные ограничения на значения тега, либо на значение, поступающее из контроллера** **Input limits on device value (входные ограничения на значения устройства).**

HWConfiguration Tags Имя_переменной

General Value Device **Limits** Conversion Thresholds Indirect Address Events

Input limits on Tag value

Enable

Min 0

Max 100

Advice in case of wrong input

Type Simple Page for both limits

Complete Page for both limits

Simple Page for both limits

Complete Page only for the max limit

Complete Page only for the min limit

Advice in case of wrong input (совет в случае неверного ввода)- в случае ввода числа, выходящего за рамки диапазона, терминал либо предупредит о неверном вводе (**Simpl**-простая страница), либо предложит установить одно из значений лимита (**Complete** -в случае полной страницы);

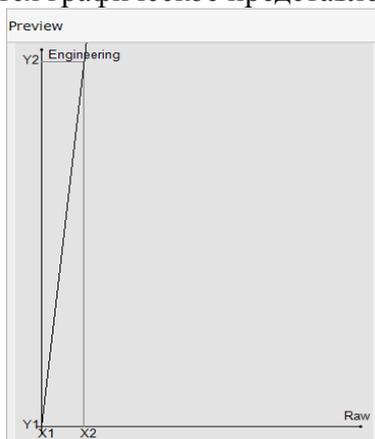
Conversion- масштабирование – либо по линейному, либо по квадратному корню - величины, вводимой с экрана или выводимой на экран относительно величины, находящейся в по соответствующему адресу в устройстве;

В случае выбора линейного преобразования, потребуется ввести координаты двух точек, по которым будет рассчитан коэффициент пропорциональности.

где координата **X (Row)**– данные в устройстве, **Y(Engineering)** – данные на терминале в поле отображения.

Ниже находится поле, в котором можно по подставленным значениям X или Y просчитать значения Y или X.

Правее находится графическое представление преобразования.



Закладка **Events** – привязывает определенные события, связанные с переменной, к функциям или скриптам (подпрограммам), выполняемым над этой переменной.

| General | Value | Device | Limits | Conversion | Thresholds | Indirect Address | Events | |
|---------|-------|--------|--------|------------|------------|------------------|------------------|-----------------------------------|
| List | | | | | | | | |
| | | | | | | | OnInitialization | |
| | | | | | | | OnOffLine | |
| | | | | | | | OnOffScan | |
| | | | | | | | OnOnLine | |
| | | | | | | | OnOnScan | |
| | | | | | | | OnRawValueChange | ShowPageNumber(Page=Переменная_1) |
| | | | | | | | OnValSent | |
| | | | | | | | OnValueChange | ShowPageNumber(Page=Переменная_1) |

Наиболее важные из этих действий:

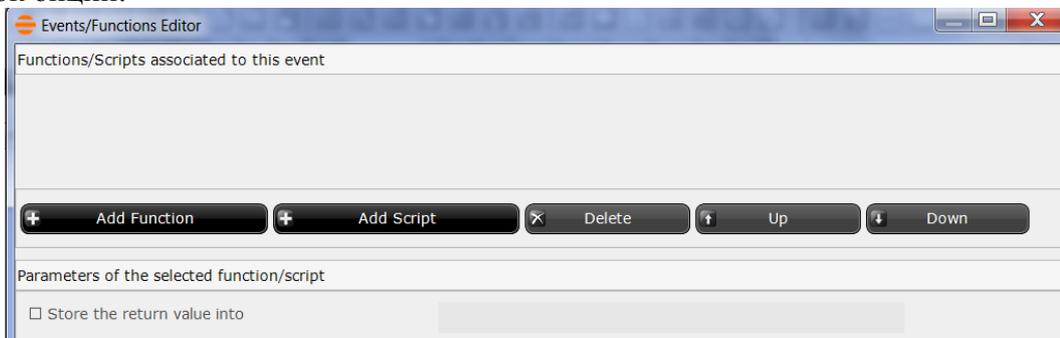
- **OnInitializing** – первое обращение к переменной при скане первого цикла программы;
- **OnRawValueChange** – изменение значения переменной в устройстве (по переднему фронту сигнала);
- **OnValSend** – посылка значения переменной в устройство (по переднему фронту);
- **OnValueChange**- изменение значения переменной в поле ввода (по переднему фронту);

Далее следует описать, что должно произойти при наступлении выше описанных событий.

Рассмотрим на примере задачи изменения текущего экрана при изменении значения переменной в контроллере:

- изменение значения переменной в контроле – это событие **OnRawValueChange**.

При нажатии ЛКМ на поле левее колонки событий появится значок , который откроет список опций:

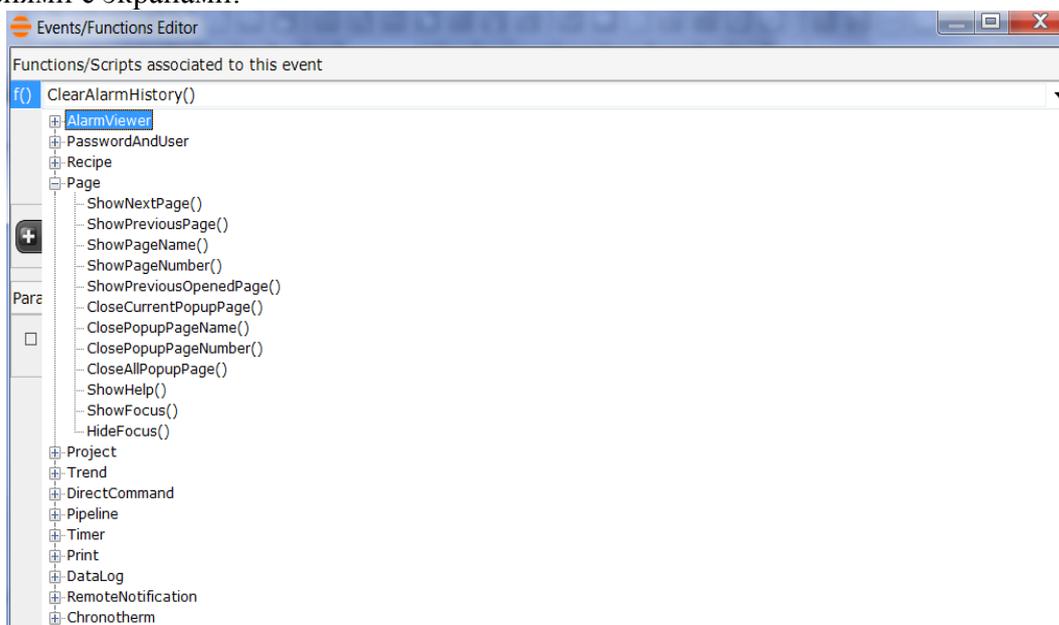


- **Add Function**- добавить функцию, которая будет выполняться при наступлении события;
- **Add Script**- добавить скрипт (макрос), который будет выполняться при наступлении события;
- **Delete** – удалить выделенную курсором функцию или макрос;

Для нашей задачи следует выбрать «добавить функцию». Выше появится поле выбора функций.

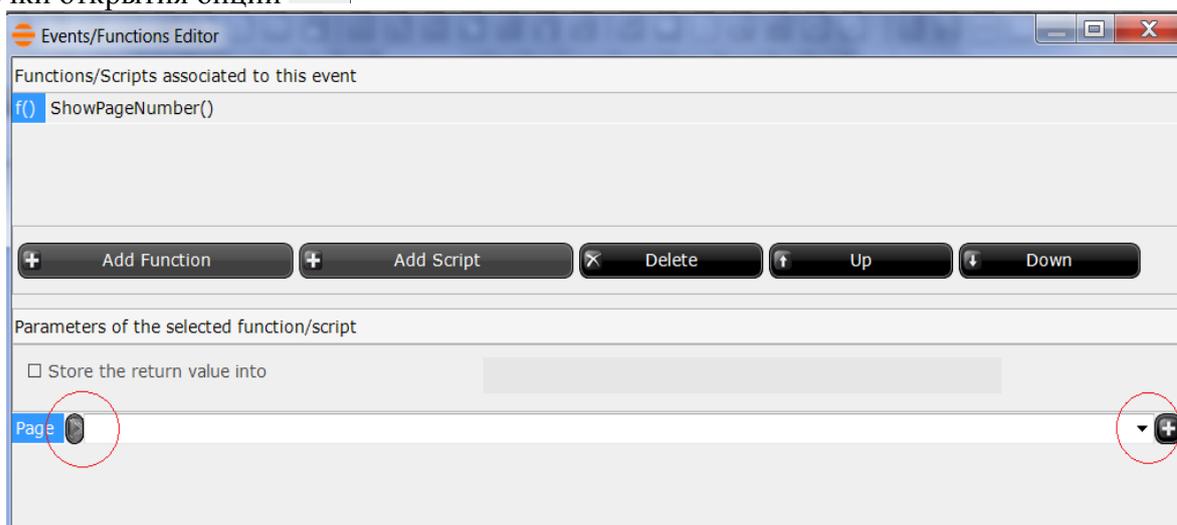


По стрелке перейти в список функций и выбрать группу **Page**, связанную с действиями с экранами:



Для открытия страницы с номером, соответствующим значению переменной, следует выбрать функцию **ShowPageNumber**

В открывшемся снизу окошке **Page** нажать на значок  до появления справа стрелочки открытия опций 



Далее из списка тегов следует выбрать тег, который должен будет содержать номер окна. (В нашем случае эта переменная называется «Переменная_1»)

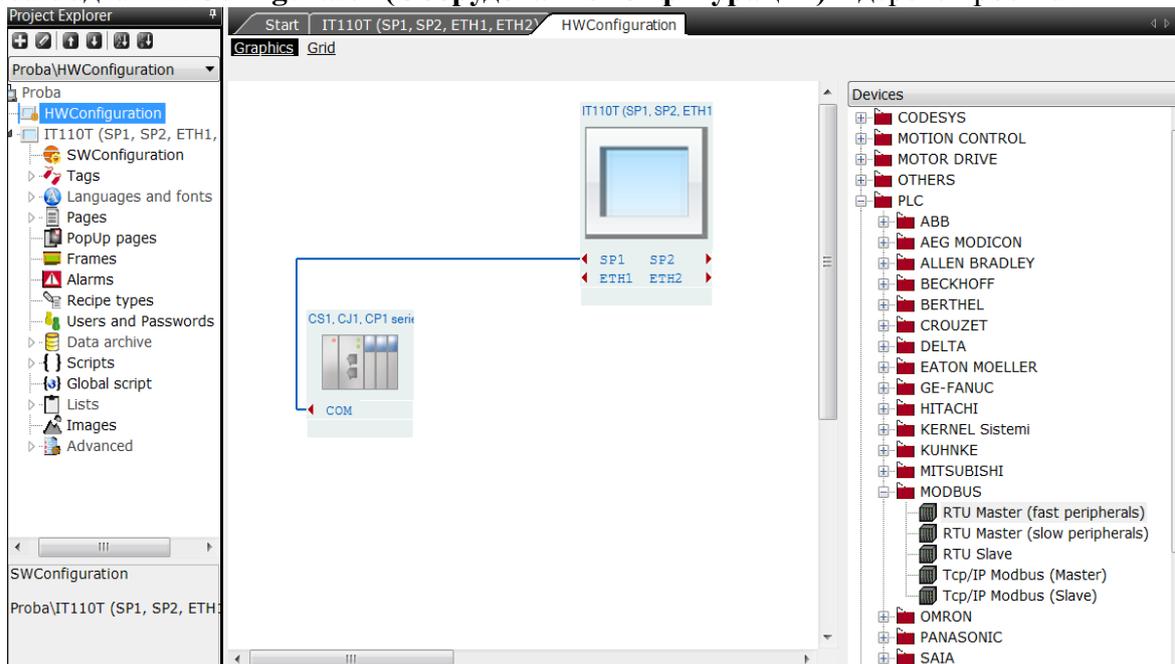
В результате на экране событий в поле события **OnRawValueChanged** появится соответствующая функция, где номеру страницы поставлено в соответствие значение переменной:

| | |
|-------------------|-----------------------------------|
| OnInitialization | |
| OnOffLine | |
| OnOffScan | |
| OnOnLine | |
| OnOnScan | |
| OnRawValueChanged | ShowPageNumber(Page=Переменная_1) |
| OnValSent | |
| OnValueChanged | |

!!! Не забудьте установить галочку для постоянного опроса этой переменной в поле Update always, even when a tag isn't used by any field

Изменение подключенных устройств и параметров связи.

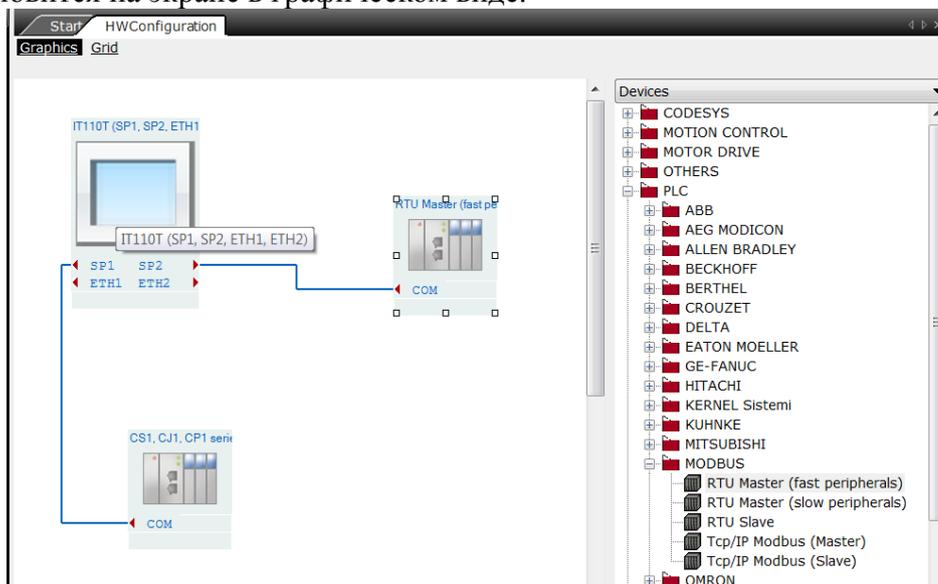
Закладка **HWConfigurator (ОборудованиеКонфигурация)** в дереве проекта:



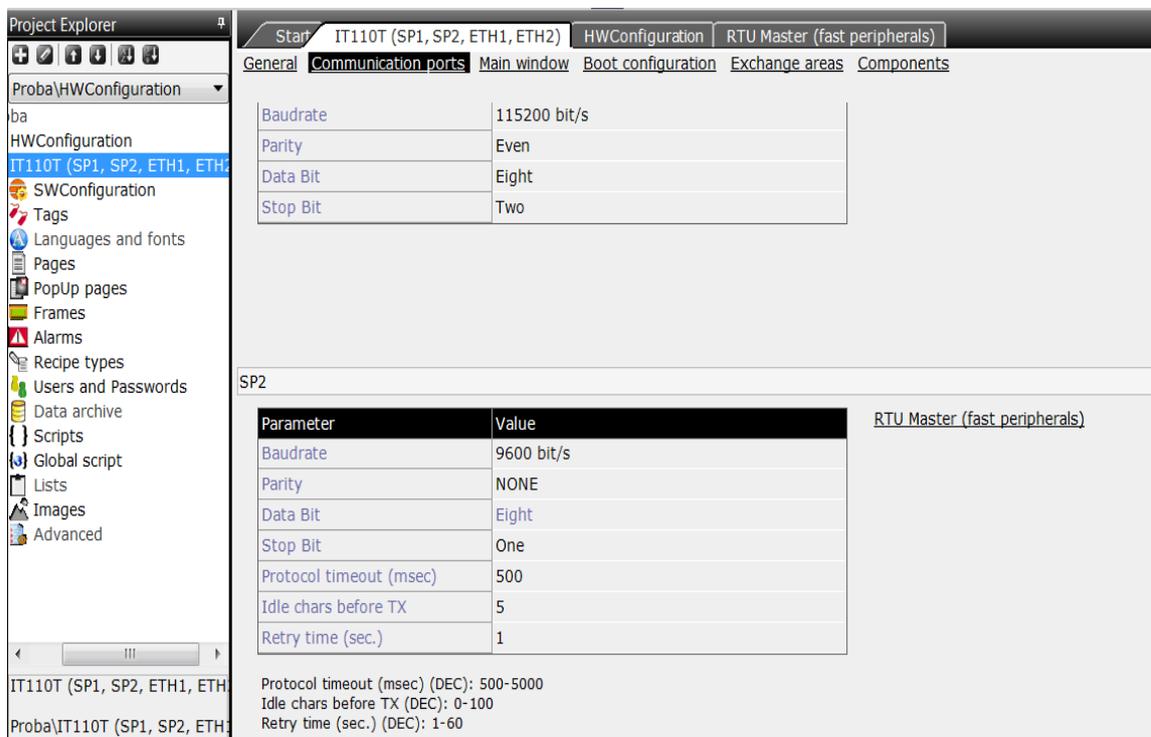
Здесь можно менять и добавлять подключаемые устройства, настраивать параметры связи.

В правой части экрана в закладке Device приведен список возможных подключаемых устройств.

Например, для создания подключения по ModBus в закладке **PLC** нужно выбрать закладку ModBus (Панель может выступать как ведущим, так и ведомым устройством) и курсором перетащить необходимую опцию на экран. В случае, если такая связь допустима, связь установится на экране в графическом виде.



Для настройки параметров связи два раза щелкните на изображении панели оператора:



Для серийных портов:

Baudrate – скорость обмена между панелью и подключенным к порту устройством . Контроллер CP1 позволяет проводить обмен на скорости 115200 бит/сек. Остальные параметры связи выбираются исходя из рекомендаций на устройство или из соображений надежности или скорости связи:

Parity - проверка четности или нечетности (**Even** или **Odd**).

Data Bit – количество передаваемых бит информации: семь (**Seven**) или восемь (**Eight**).

Stop Bit – стоповые биты.

Дополнительные параметры зависят от выбранного протокола связи, для Modbus это:

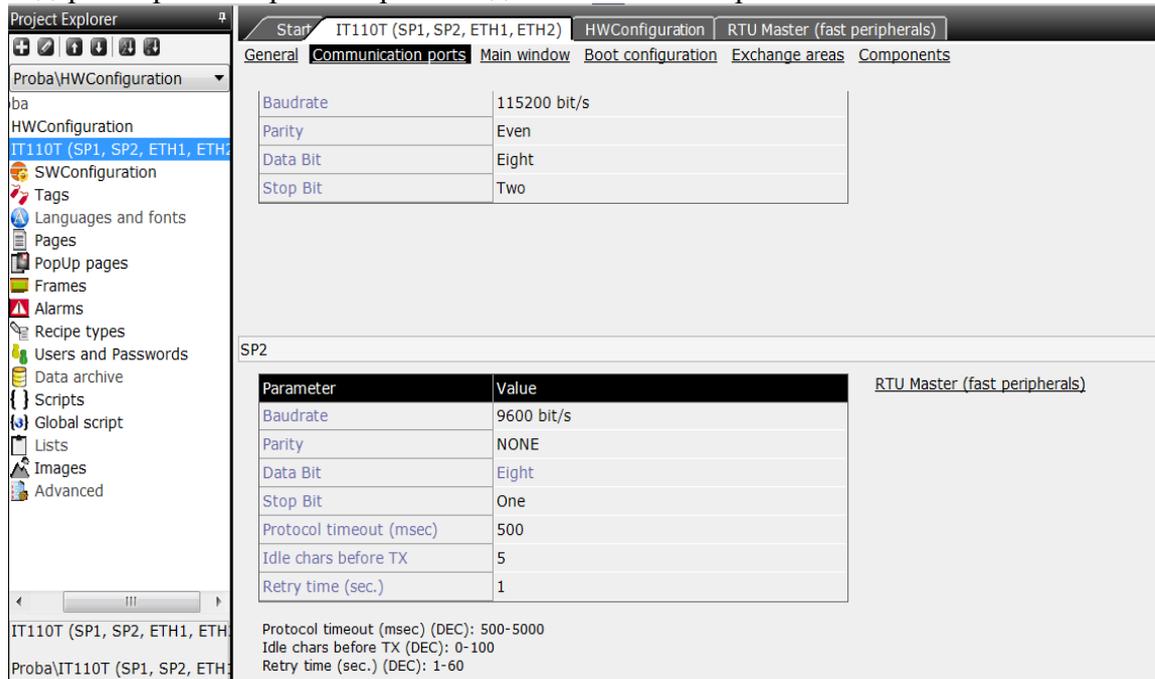
Protocol timeout – период молчания между сообщениями

Idle chars before TX -

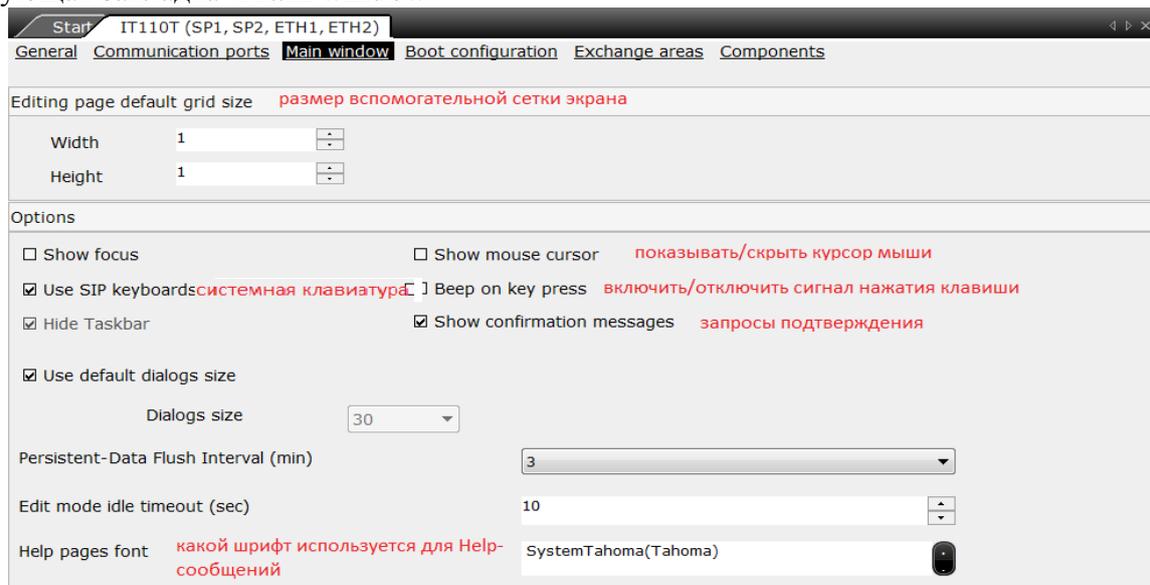
Retry time – период попытки восстановления связи при сбое.

Настройка функций интерфейса.

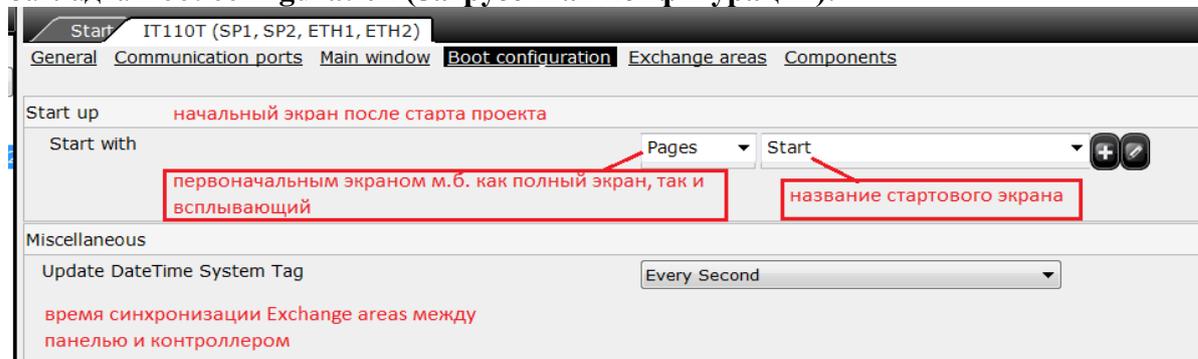
В дереве проекта при выборе закладки с именем выбранной панели



Следующая закладка- **Main window**:



Закладка **Boot configuration (Загрузочная конфигурация)**:



Закладка **Exchange areas (Зоны обмена)**:

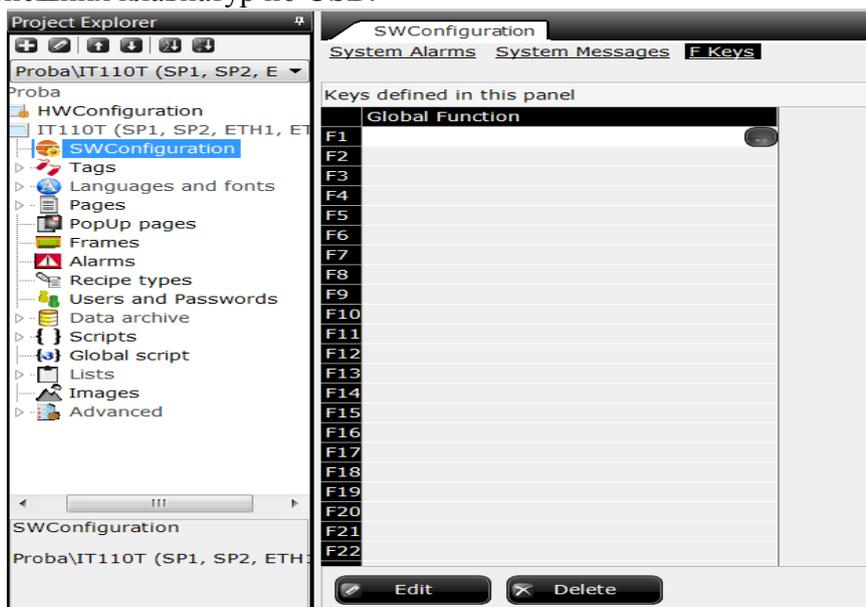
Пересылка таблиц данных либо между несколькими устройствами, подключенными к панели, либо между системными переменными панели и подключенными устройствами.

Закладка **Components (компоненты)**– список активный опций.

Системные сообщения, функциональные клавиши, языки и шрифты.

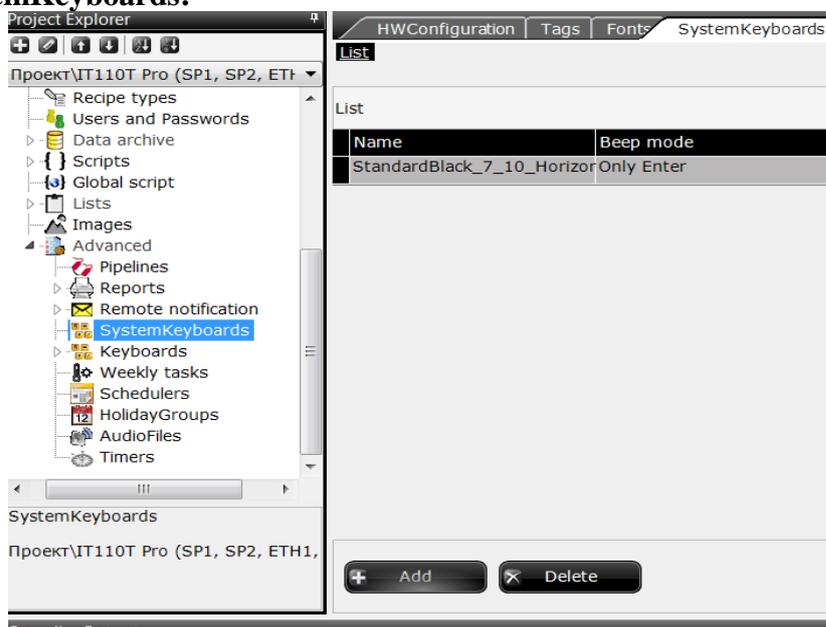
Закладка **SWConfiguration** (конфигурация ПО) в дереве проекта: **System Alarms** - системные сообщения, которые выдаёт устройство при сбое или ошибке в проекте или среде исполнения в время работы. Эти сообщения можно отредактировать. При выборе языка можно выбрать в качестве языка системных сообщений русский язык. (Будет рассказано ниже).

F Keys (F-клавиши) позволяет присваивать определенные функции функциональным клавишам для панели IT105TK (с встроенной клавиатурой) и для остальных панелей в случае подключения внешних клавиатур по USB.

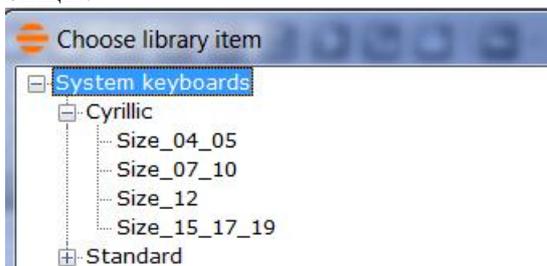


Закладка **Languages and Fonts** в дереве проекта — позволяет добавлять языки проект, что позволяет создавать многоязыковые сообщения переключаемые по команде из проекта. В данной закладке так же можно добавить дополнительные шрифты в проект и использовать их для каждого из языков. Так же можно изменить вид и тип системных клавиатур ввода для каждого языка.

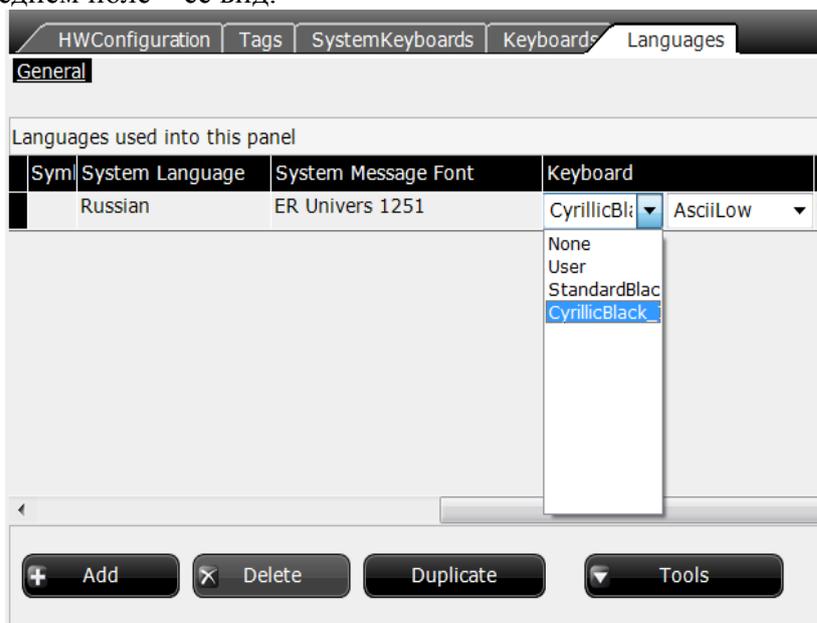
Для того, что бы при вводе информации с экрана использовать кириллицу, необходимо ее загрузить через опцию **Advanced** в дереве проекта. В опции **Advanced** выберите опцию **SystemKeyboards**:



Нажмите кнопку **Add** и в открывшемся окне выберите подходящие размеры клавиатуры кириллицы:



Далее, открываем опцию **Languages and Fonts** в дереве проекта и выбираем опцию **Languages** и в столбце **Keyboard** и **Runtime Keyboard** выбираем клавиатуру с кириллицей, а в соседнем поле – ее вид.



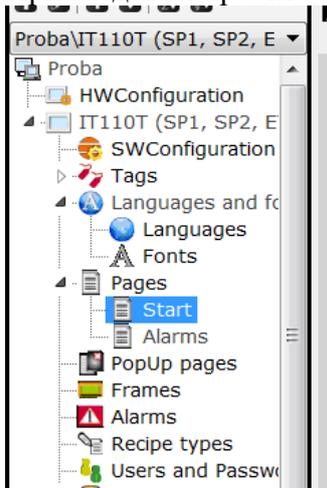
Если в столбце **SystemKeyboards** выбрать **ER Univers 1251**, то и в поле ввода и редактирования рецептов можно будет вводить и просматривать информацию на русском языке. И вообще, весь диалог будет проходить на русском языке.

Внимание!!! При использовании нескольких языков используйте цифровые пароли, чтобы исключить проблему доступа к объекту под паролем.

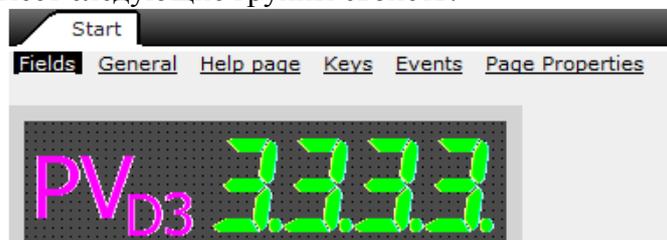
В этой закладке мы добавляем/изменяем языки пользователя в проекте, а в закладке **Fonts** – используемые шрифты. Используется шесть системных шрифтов и можно подключить до двадцати четырех шрифтов Windows. При выборе шрифта Windows необходимо установить галочку в поле **Active**, в противном случае шрифт не будет загружен в проект.

Страницы (Экраны).

При создании проекта автоматически создаются два окна: **Start** и **Alarm**



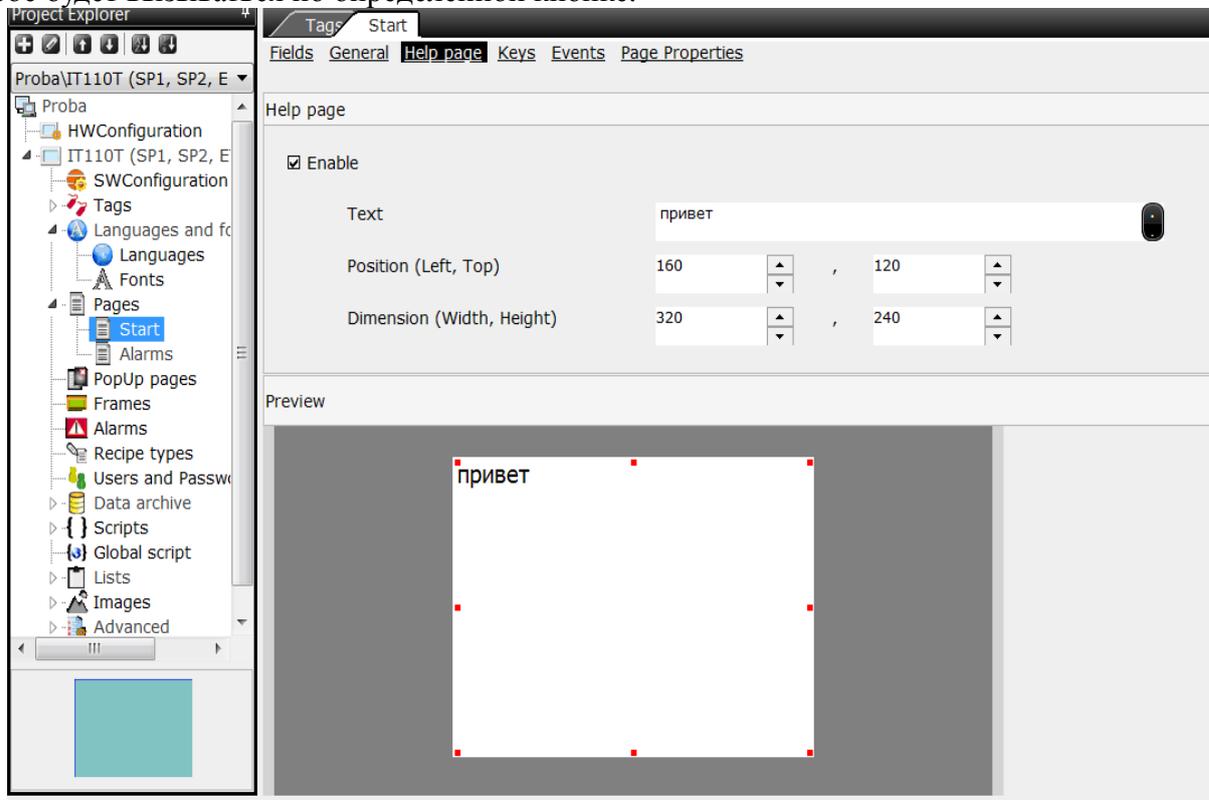
Каждый экран имеет следующие группы свойств:



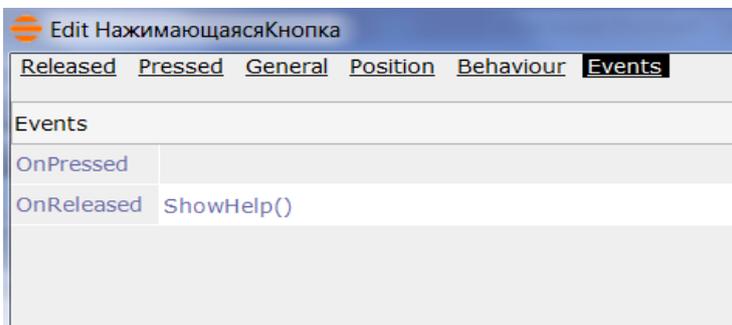
– **Fields** – отображаемая область на которой размещаются объекты;

– **General** – имя экрана, комментарий, номер экрана, описание, размер ячеек сетки привязки объектов

– **Help page:** на каждом экране возможно ввести сообщение помощи (экран Help), которое будет вызываться по определенной кнопке.



Вызвать экран **Help** можно с помощью функции **Create push button** на панели инструментов. Для этого следует использовать закладку **Events(События)** в свойствах кнопки.

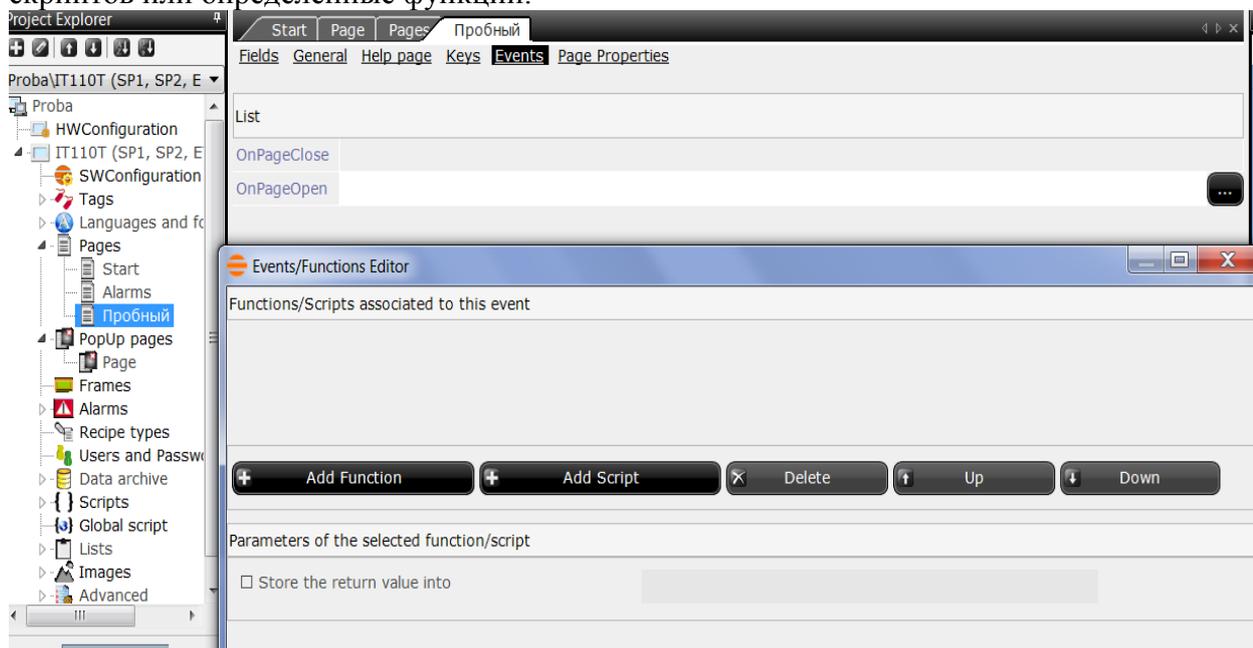


Использовать некоторые функции возможно только по отжатию кнопки , в том числе и **ShowHelp**.

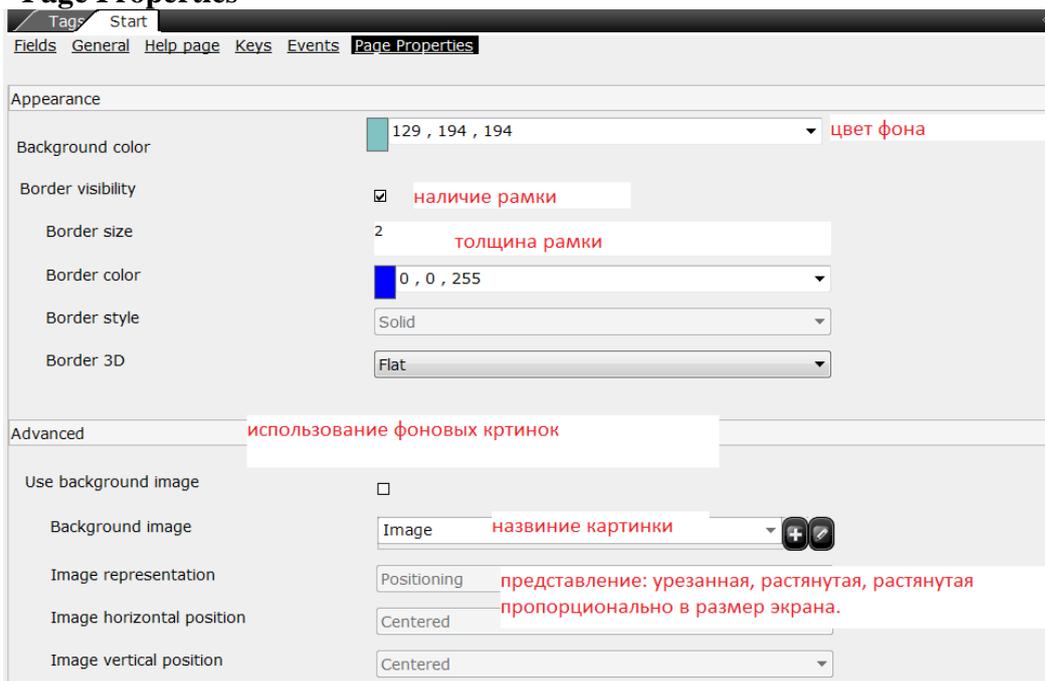
- **Keys** - позволяет присваивать определенные функции функциональным клавишам для панели IT105TK (с встроенной клавиатурой) и для остальных панелей в случае подключения внешних клавиатур по USB.

- Events

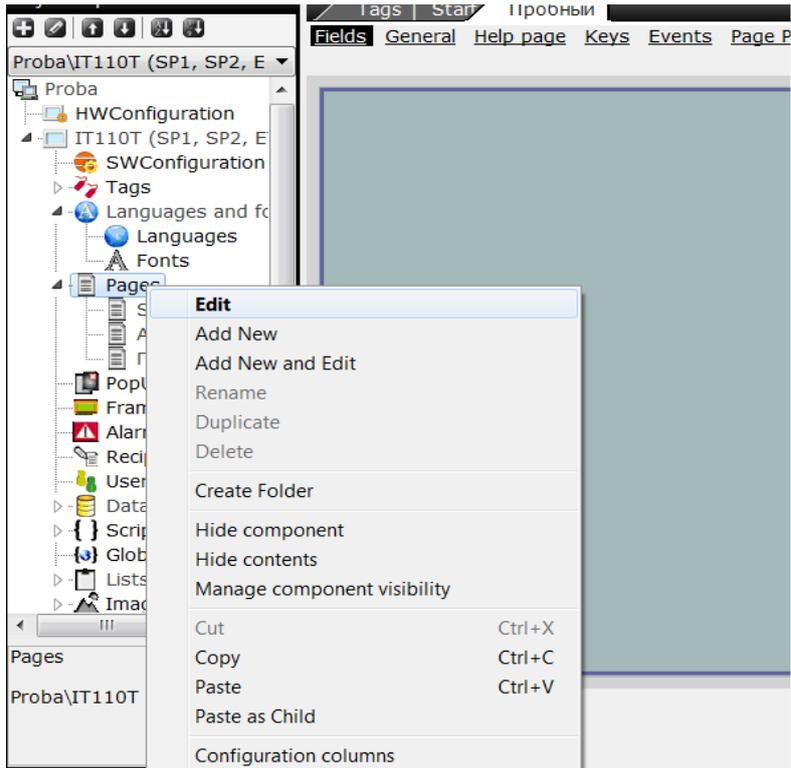
К открытию и закрытию экрана привязываются определенные события с помощью скриптов или определенных функции.



- Page Properties



Для создания нового экрана установите курсор на поле Pages в дереве проектов и по правой клавише мыши в появившемся меню выберите либо **Add New**, либо **Add New and Edit** (создать и редактировать):



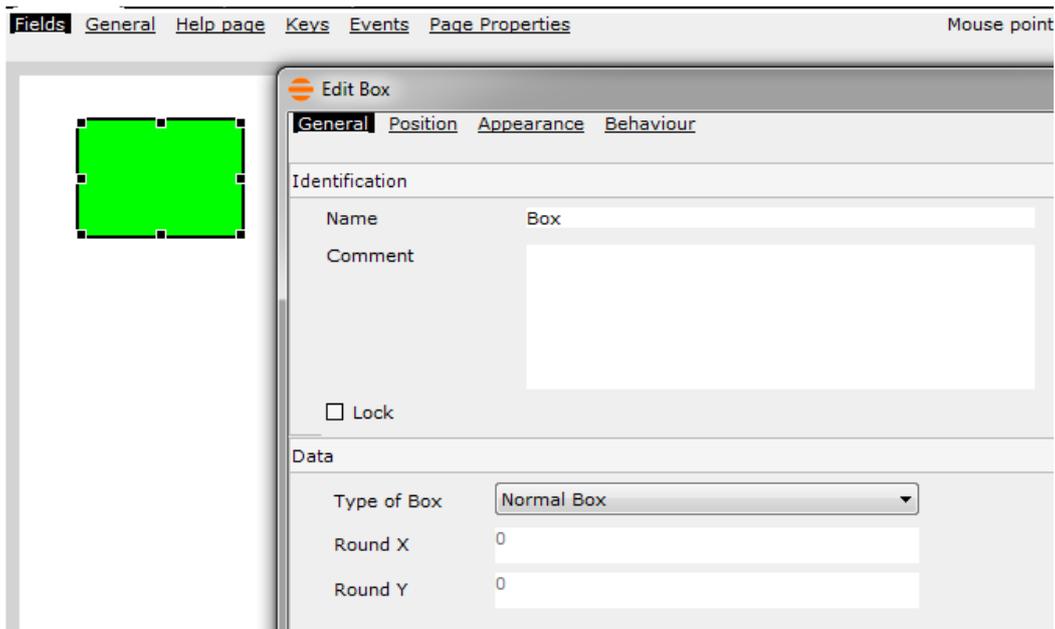
Стандартные объекты.

Большинство свойств объектов, размещаемых на экране может быть привязано к состоянию переменной: цвет заливки, цвет рамки, ее наличие, толщина, отображается объект на экране или нет, цвет шрифта, мигание. Это означает что практически любой объект может выполнять роль индикатора/лампы.

1) Простые фигуры:

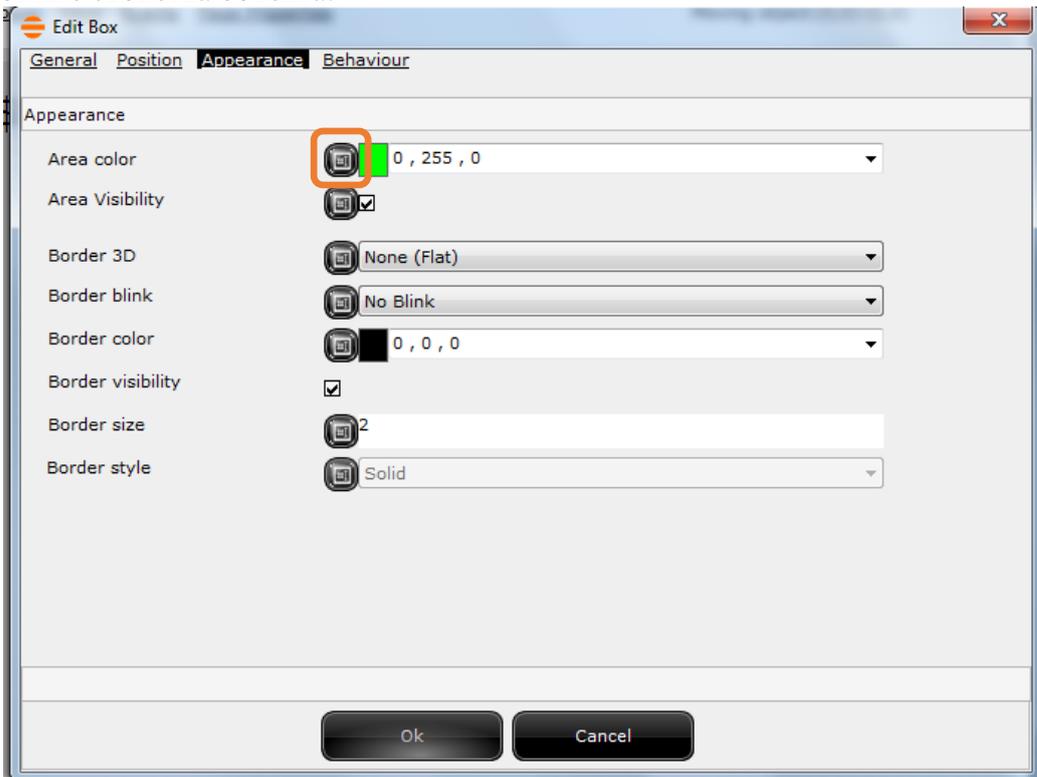


прямоугольник, эллипс, дуга, сектор, линия, многоугольник, составная линия, 6 угольник, надпись и картинка.

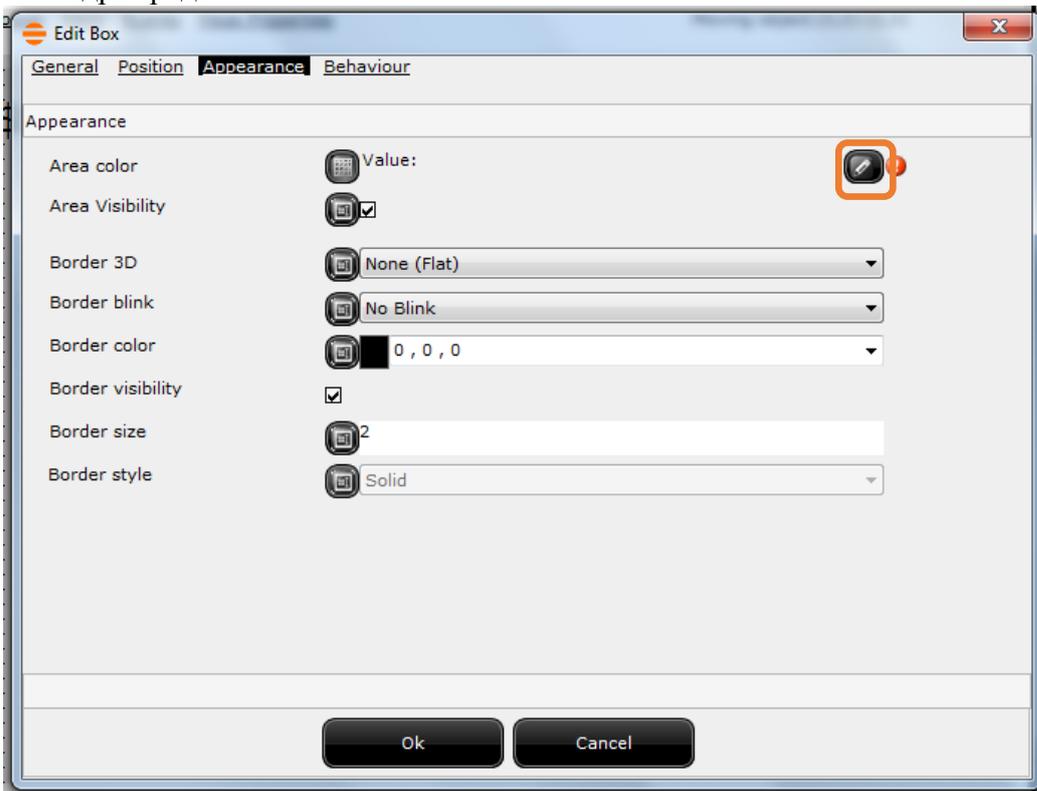


Рассмотрим использование стандартных объектов на примере прямоугольника:

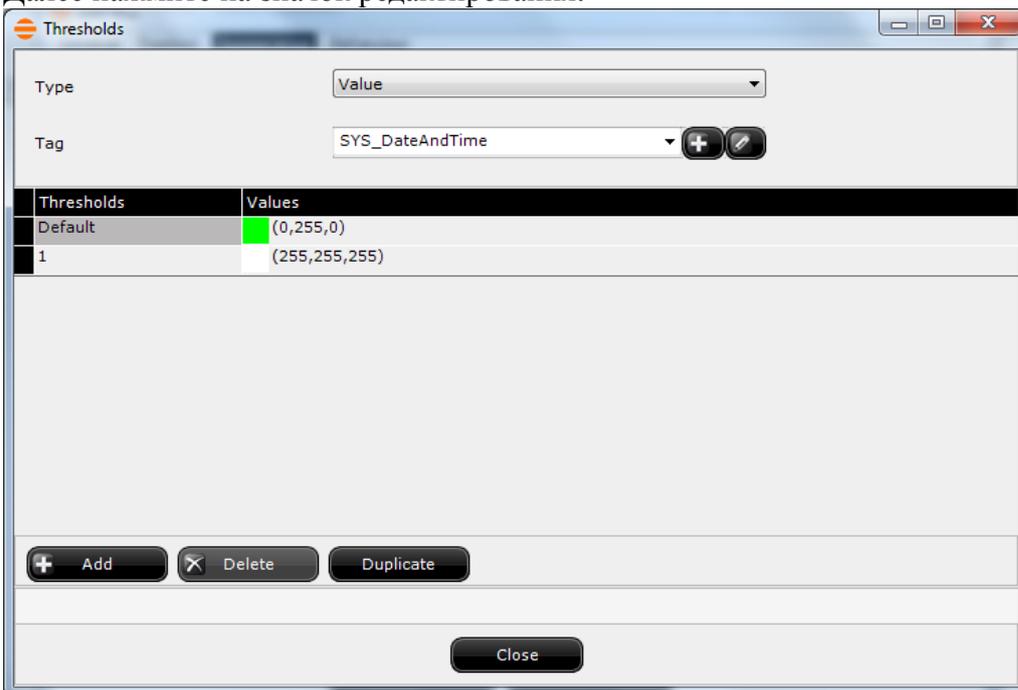
General – Имя объекта и комментарий, Lock фиксирует объект на данном месте. Data – варианты формы объекта. Position – координаты положения объекта и его размеры. Appearance – основные свойства объекта:



Для привязки изменения цвета к переменной вам следует 2 раза нажать на отмеченный черный квадрат рядом с Area color:

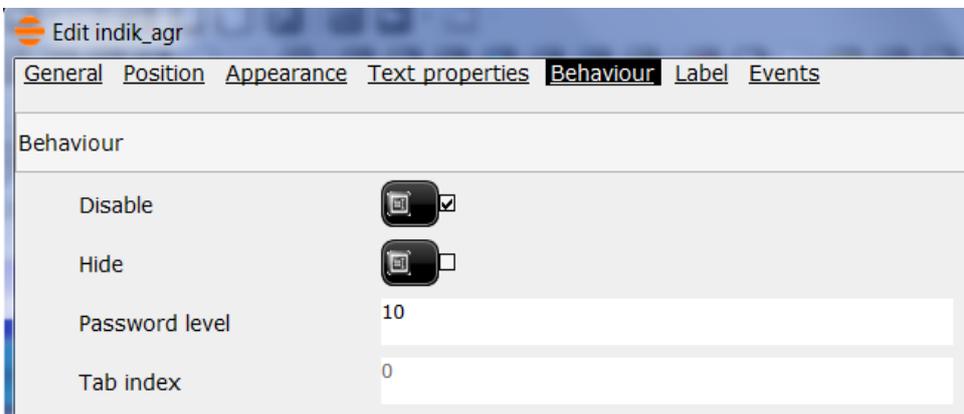


Далее нажмите на значок редактирования:



В отрывшемся окне можно выбрать переменную, ее тип и установить для каждого ее значения цвет. Аналогично можно менять отображение, цвет, толщину и другие свойства объекта.

- **Border 3D** – форма границы объекта. Либо **None(Flat)** плоская, либо различные формы 3D.
 - **Border blink** – моргание границы объекта.
 - **Border color** – цвет границы объекта. Здесь так же, как и выше цвет можно задавать напрямую или с помощью тега.
 - **Border visibility** – видимость границы объекта.
 - **Border size** – толщина границы объекта.
- Закладка **Behaviour** – определяет поведение объекта:



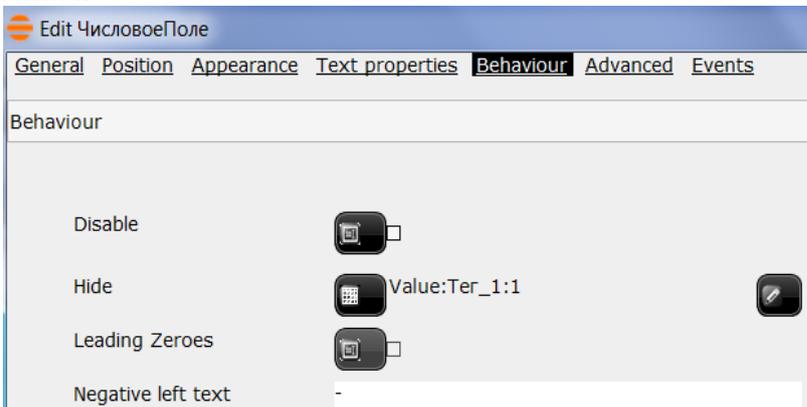
Если галочка установлена в поле **Disable**, то у объекта блокируются свойства кнопки и остаются только свойство индикации информации. Кроме того, можно воспользоваться свойствами поля  для управления Tach- свойствами объекта.

Hide – скрыть объект.

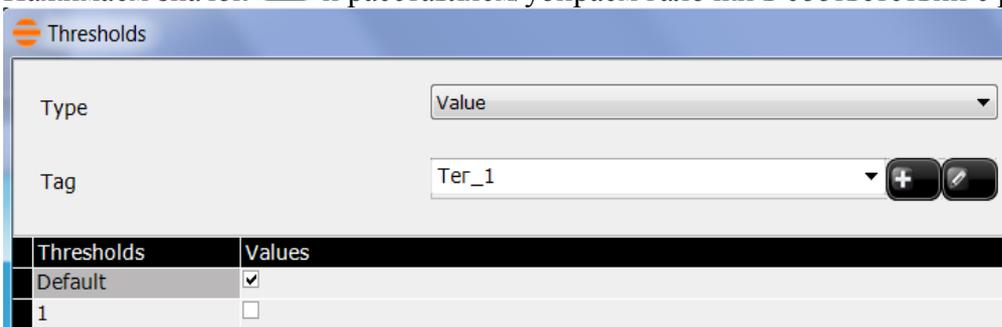
Скрыть объект можно используя значения переменной (бит или слово).

Иногда возникает необходимость инвертировать бит: при значении переменной=0 скрывать, при значении переменной, равной 1 – показывать.

Для этого щелкаем по полю  ЛКМ до для перехода в режим «по значению переменной»:



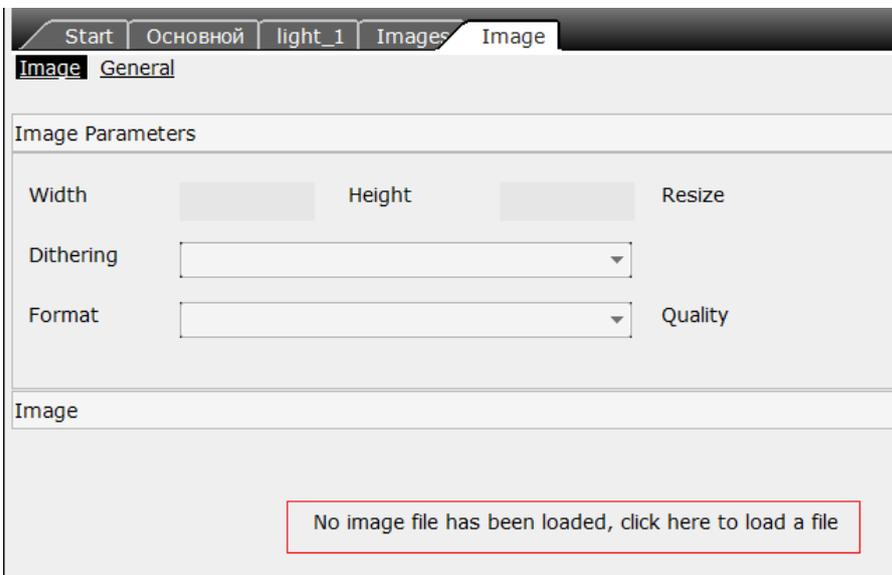
Нажимаем значок  и расставляем/убираем галочки в соответствии с рисунком ниже:



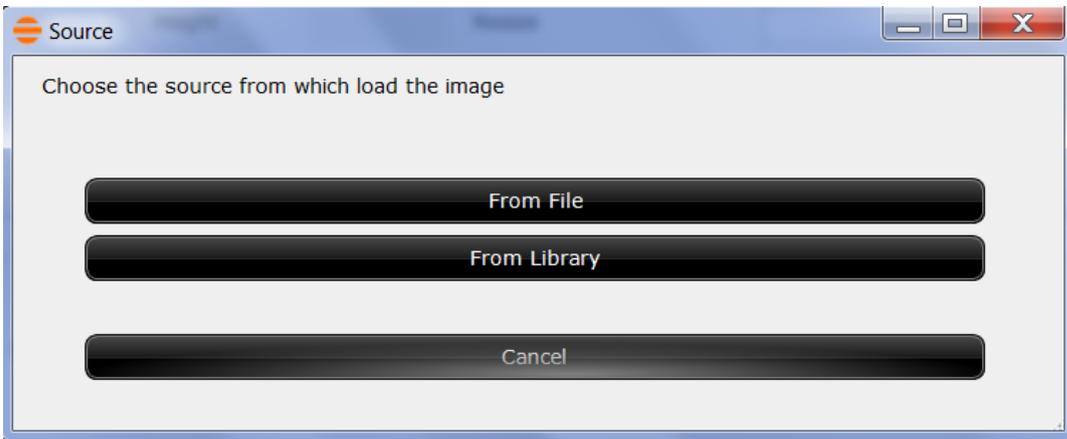
Кнопка добавления изображения **Images**:



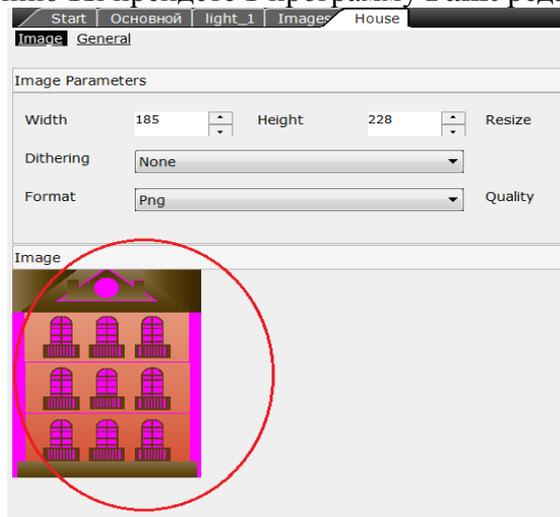
Для добавления картинки нажмите на кнопку, после чего наведите курсор на поле экрана нажав левую кнопку мыши задайте размеры картинки, после чего щёлкнув два раза на появившемся поле откроется окно выбора изображения:



Далее нужно будет выбрать, откуда будет загружено изображение: из файла или библиотеки:



После добавления изображения в проект вы можете его отредактировать: после 2-го щелчка по изображению вы перейдете в программу **Paint** редактирования картинки:



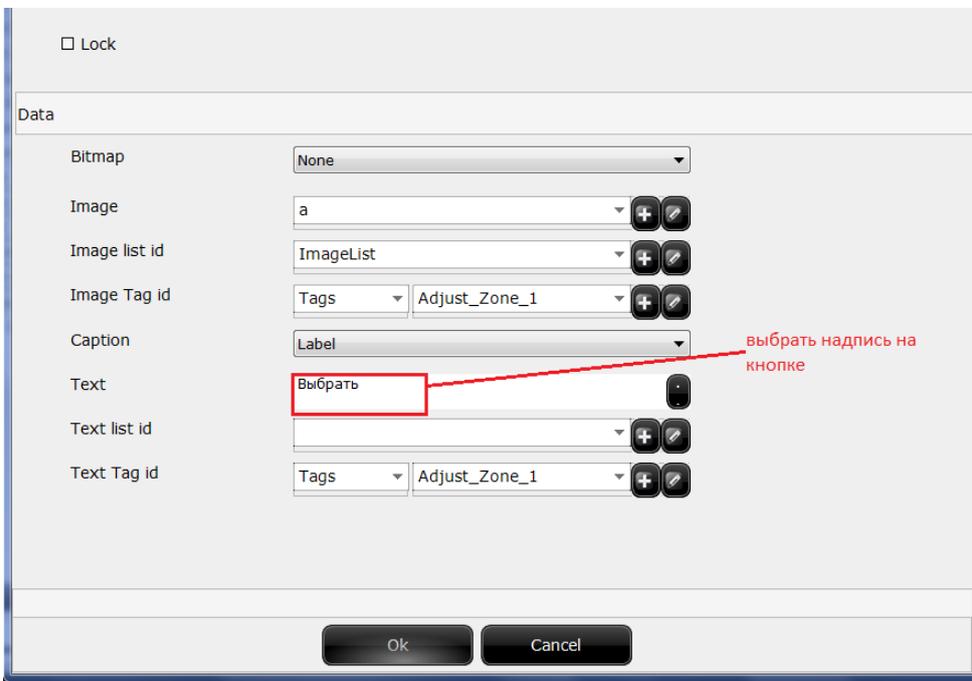
Далее эти картинки можно размещать на экранах с помощью инструмента **Create an Image**  или **Create a symbolic field**.

2) Простое управление:



- **Create a touch button** (Создать сенсорную кнопку).

Основные свойства:



Create a touch aria – инструмент для создания сенсорной области, не имеющий изображения. В закладке **Events** событиям (нажата/отжата) присваиваются определенные функции и скрипты.

В проекте на экране этот объект для визуализации отображается в виде заштрихованного прямоугольника. Для того что бы сделать невидимым/видимым этот объект на стадии разработки проекта, на панели инструментов существует объект **Show/Hide touch**

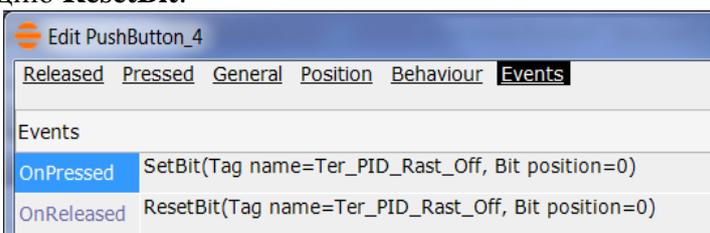


arias

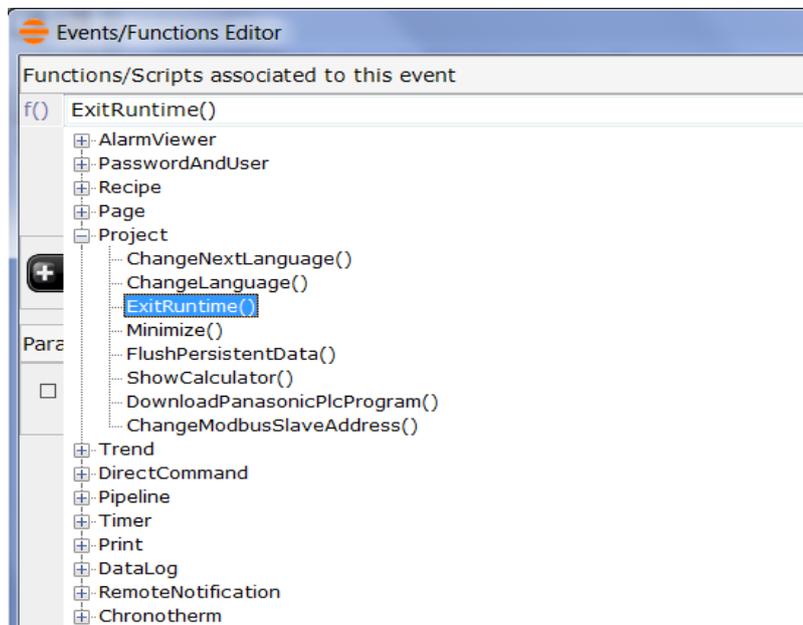


Для создания кнопки без фиксации используется инструмент **Create a push button**

Для создания кнопки без фиксации воспользуемся свойством закладки **Events** и присвоим событию **OnPressed** функцию **SetBit**, а в момент отпускания кнопки - **OnReleased** функцию **ResetBit**:



По функции **OnReleased** можно осуществить переход в системное меню панели. Для этого нужно выбрать функцию **ExitRuntime**.



Для создания переключателя следует воспользоваться инструментом **Create a switch button**. В этом объекте тег назначается в закладке **General**



Объект **Create numeric field (Создать числовое поле)**. С помощью этого объекта создается поле как ввода чисел, так и отображения.

Закладка **General** :

Decimal digits (десятичные цифры) – количество цифр после десятичной точки

Digits (цифры)– общее кол-во цифр, включая разделитель между порядками в случае, если установлен значок присутствия разделителя **Thousep** в закладке **Behaviour**.

Вид разделителя между разрядами устанавливается в папке **Languages and fonts/Languages/General** в столбце **Grouping Symbol**

- **Keyboard Id (идентификатор клавиатуры)**– тип клавиатуры для ввода

- **Representation (представление)** – тип представления переменной.

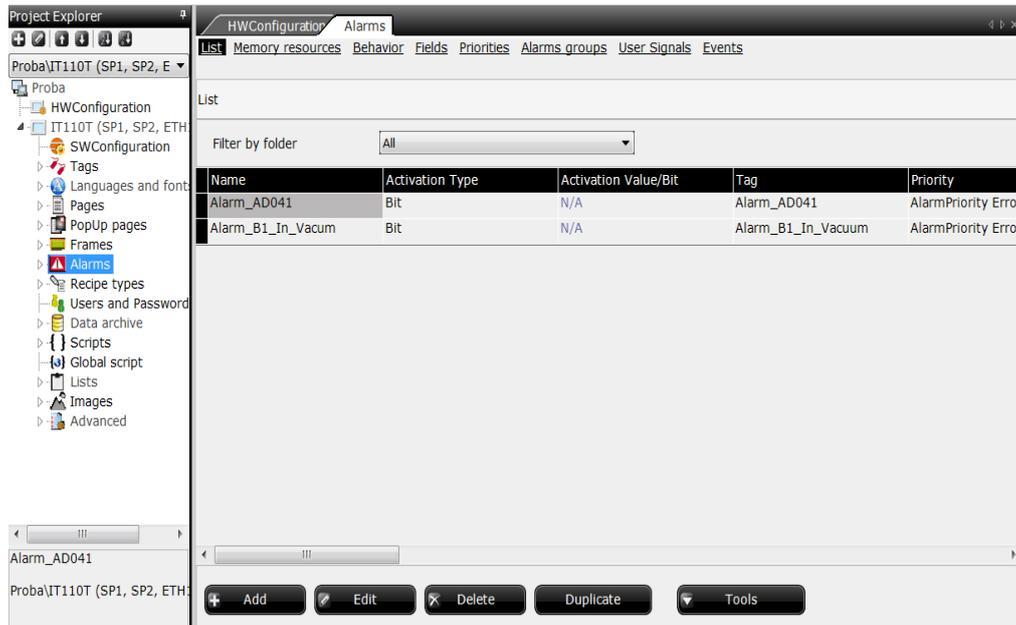
Где **Hexadecimal** – представление в шестнадцатиричном формате. **Binary** – в двоичном формате. **Fixed point** – с фиксированной десятичной точкой. Последний формат полезен тогда, когда в переменной содержатся данные, физическое значение которых кратно порядку. Это представление аналогично линейному преобразованию переменной с коэффициентом, кратным 10.

Tag Id (идентификатор тега) – имя тега.

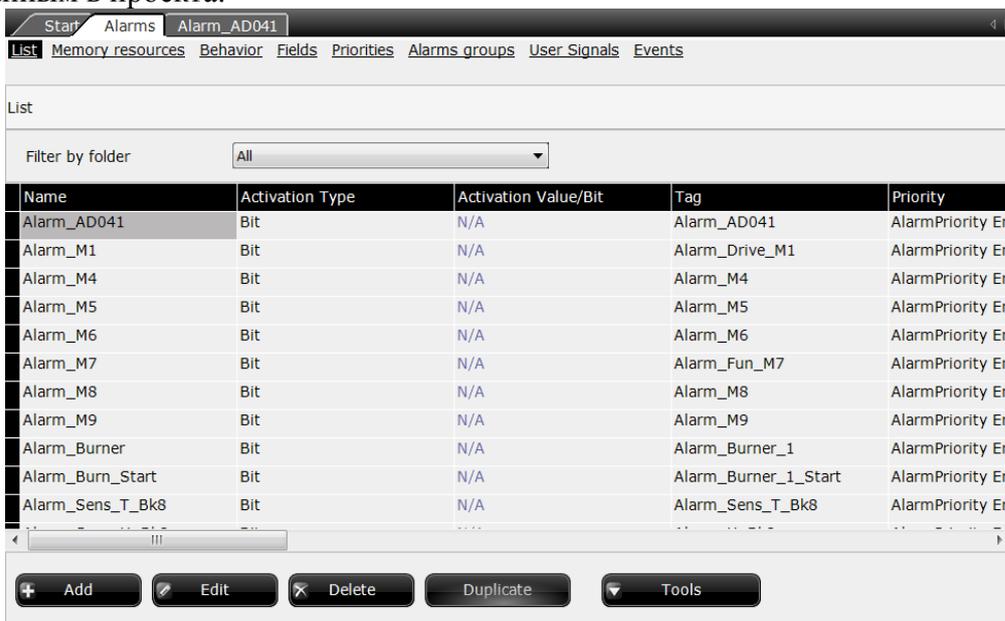
В случае, когда требуется объект отображения числовой информации, функция ввода закрывается с помощью свойства **Disable** в закладке **Behaviour**.

События и Аварии (Alarms)

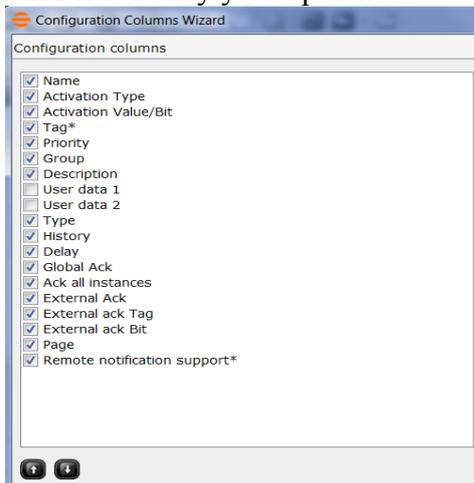
В папку **Alarm** переносятся теги из уже внесенных в проект переменных, относящиеся к авариям.



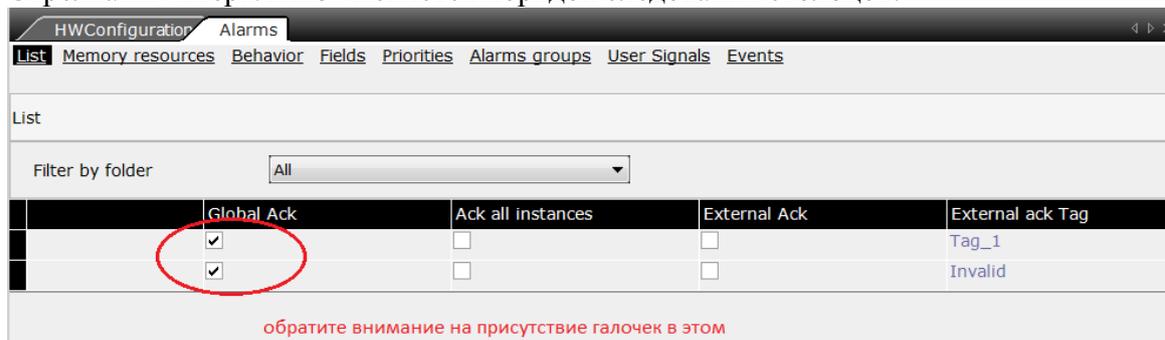
Закладка **List** – это сводная информация по всем авариям и предупреждениям, созданным в проекте.



С помощью кнопки «**Tools/Configuration columns**» отображаемую информацию можно настроить по своему усмотрению:

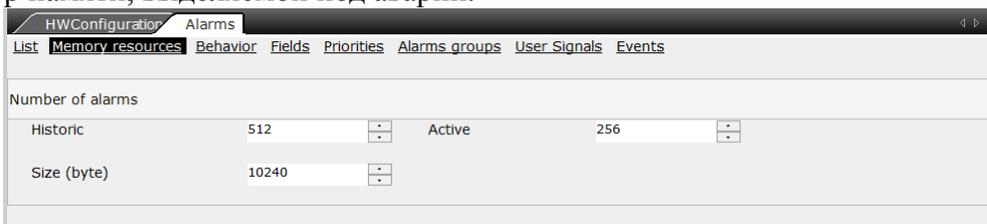


Стрелками «вверх/вниз» меняется порядок следования столбцов.

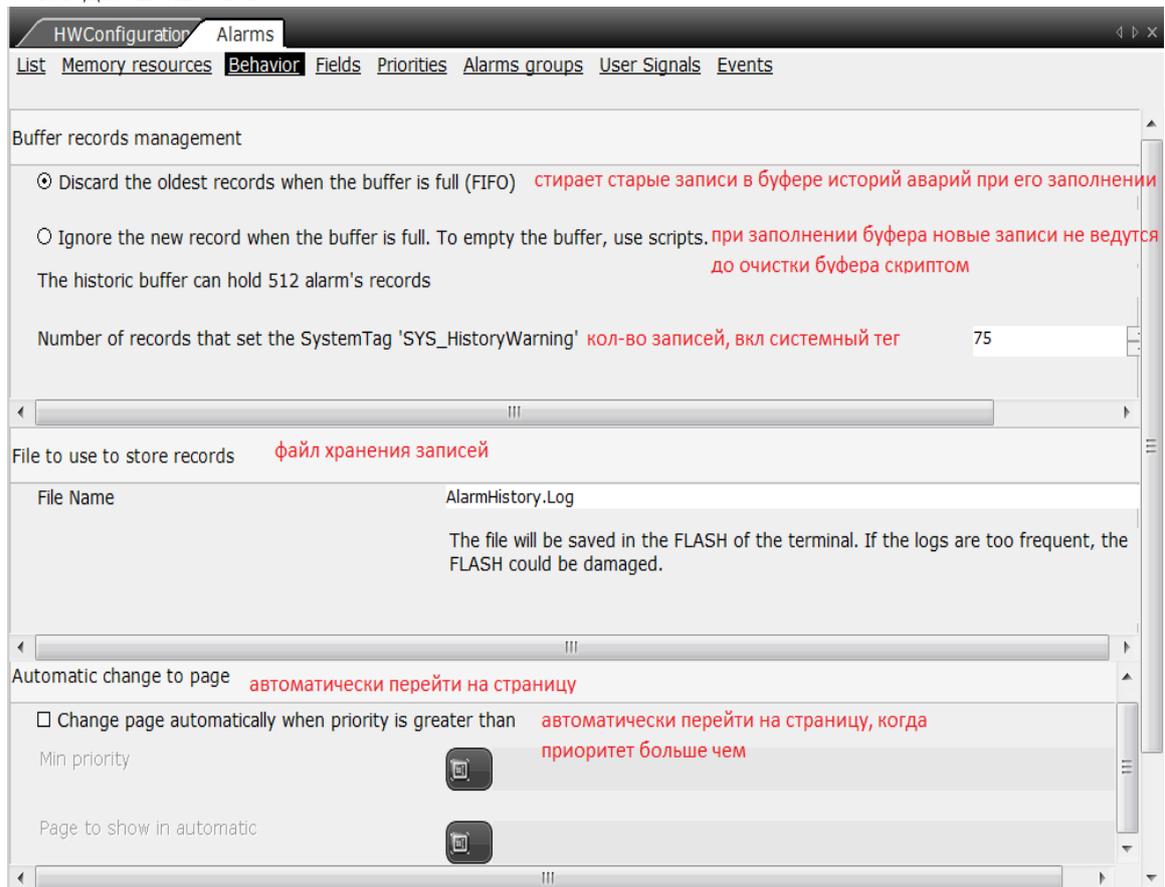


обратите внимание на присутствие галочек в этом столбце - квитирование с помощью одной кнопки

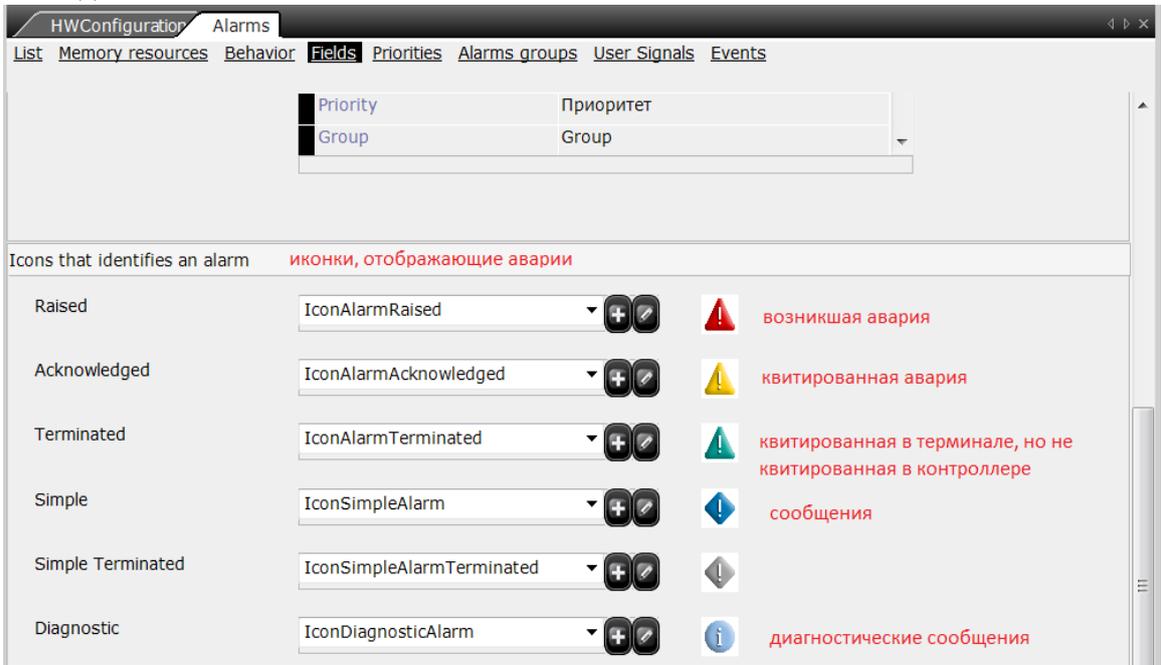
Закладка **Memory resources** позволяет оптимизировать память, выделяемую под Alarm, и определяет кол-во аварий, хранящихся в буфере историй, количество активных аварий и размер памяти, выделяемой под аварии.



Закладка Behavior

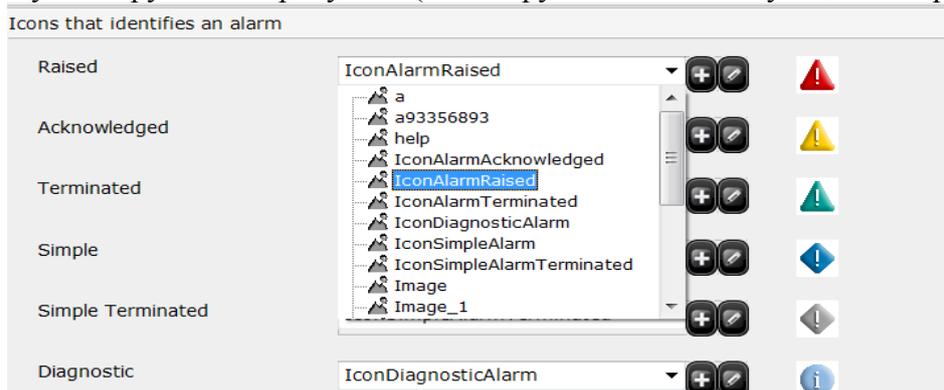


Закладка Fields

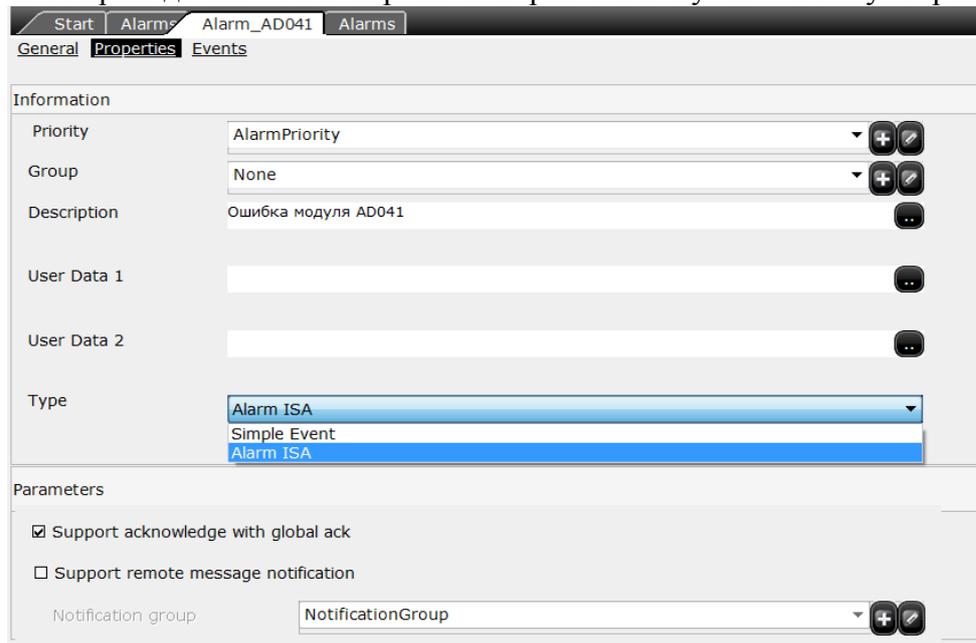


Значки, отображающие аварии, при возникновении аварии (события) будут появляться в установленном вами месте на всех экранах.

Вид значков, обозначающих соответствующие события можно изменить, выбрав из списка уже загруженных рисунков (или загрузить соответствующие изображения в **Images**):



Но отображение аварии (появление значка на экране, отображающего эту аварию) зависит от принадлежности конкретной аварии к какому-либо классу аварий.



На картинке выше отображено меню настройки аварии под названием Alarm_AD041. Если мы выбираем какие-либо Alarm из списка **Priority**, то они принадлежат к Type **Alarm ISA**, а если выбираем Priority **Message**, то в закладке **Type** следует выбрать **Simple Event**.

Закладка **Priorities** позволяет добавлять новые аварии/сообщения с различными приоритетами.

В примере ниже был создан приоритет **Message**. Далее этот приоритет будет присвоен одному из тегов, и наступление события Message на экране будет отображаться своим знаком, как показано ниже.

| Name | Value | Foreground Color | Background Color |
|---------------------------|-------|------------------|------------------|
| AlarmPriority Fatal Error | 0 | (0,0,0) | (255,255,255) |
| AlarmPriority Error | 100 | (0,0,0) | (255,255,255) |
| AlarmPriority Warning | 200 | (0,0,0) | (255,255,255) |
| Message | 30 | (0,0,0) | (255,255,255) |

Закладка **User Signals** позволяет привязать сигнализацию события со значком события и экраном:

Signals used to inform the operator

Enable a signal and move it to the desired position

- Raised Alarms *типы сообщений, выводящихся на экран*
- Message
- Diagnostic Alarm
- Banner

Raised Alarms *поле настройки значка*

Left: 0

Top: 0

Priority: AlarmPriority Error

Page to show:

Buzzer settings

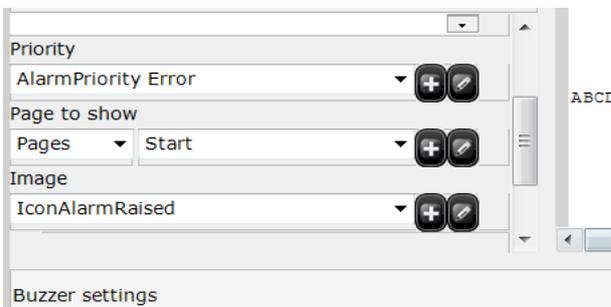
- enable buzzer
- Minimum priority that will trigger the buzzer: AlarmPriority Error

Annotation: значки и бегущую строку можно перемещать по всему полю экрана, задавая позицию

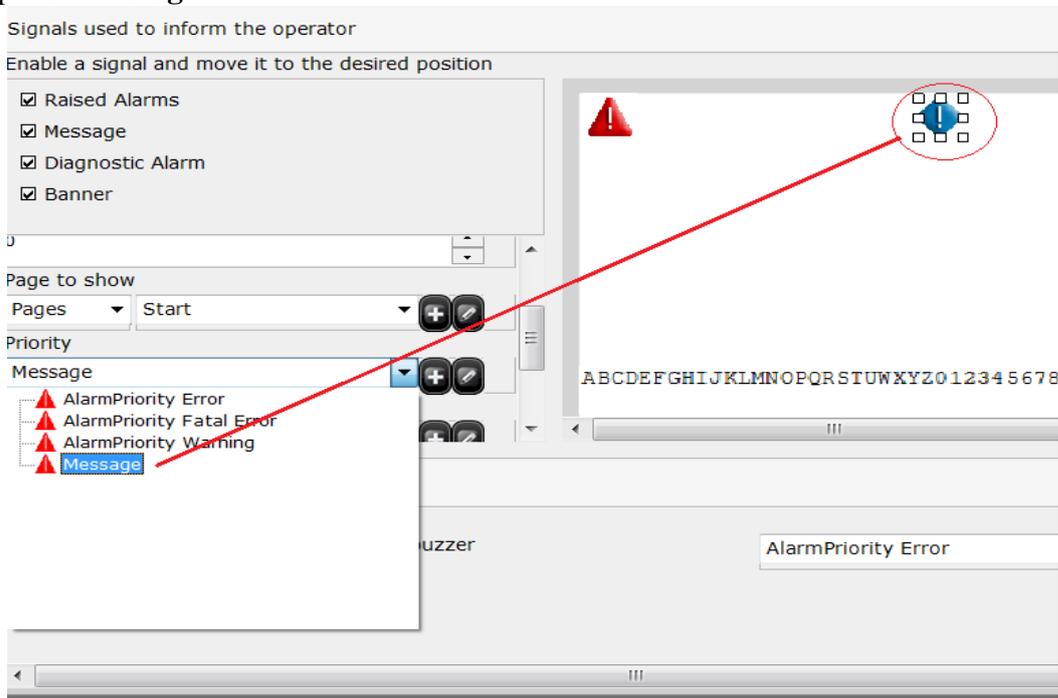
Annotation: по щелчке осущ. переход в поле настройки значка

Annotation: Banner - бегущая строка

Поле настройки значков определяет позицию значка, приоритет и привязку значка к определенному экрану, на который будет осуществлен переход по нажатию на значок.



На рисунке ниже значку **Message** ставится в соответствие ранее созданному нами приоритет **Message**



При наступлении события **Message** на всех экранах высветится этот значок, а при наличии галочки и установки уровня приоритета в опции **Enable Buzzer** в нижней части закладки **User signals** это событие еще сопровождается и звуковым сигналом.

Ознакомимся с логикой работы системы сообщений об авариях/событиях: возникающие аварии/события мы описываем в опции **Alarm** дерева проекта (присваиваем конкретным авариям имя, соответствующий адрес и приоритет), а просматриваем и квитируем возникающие аварии/события с помощью другого объекта – **Create an active alarm viewer** «активные аварии» .

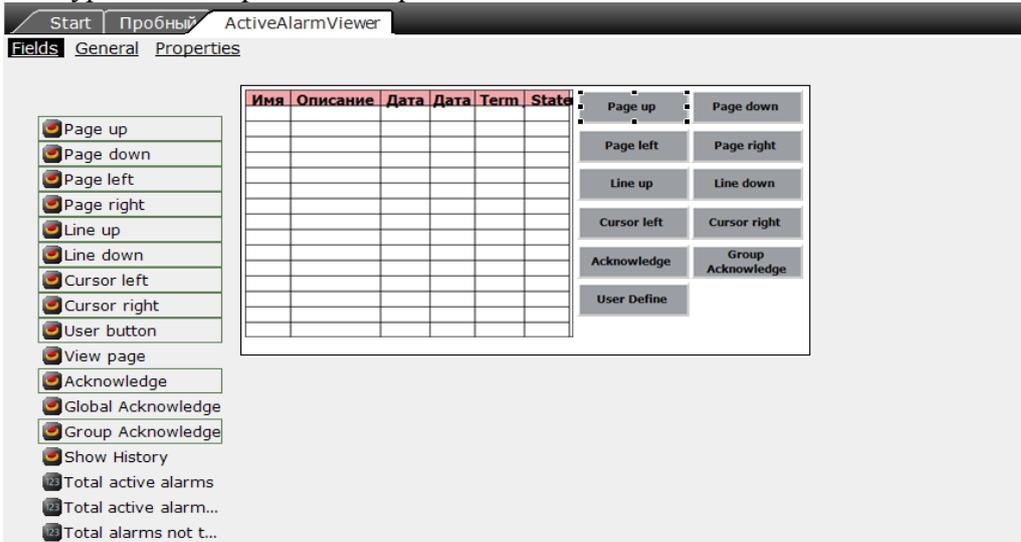
При возникновении аварии/события, отнесенных к определенной категории, на экранах возникает соответствующий значок. При нажатии на значок осуществляется переход на описанный в свойствах значка (**Page to show**) экран (это может быть созданный по умолчанию аварийный экран **Alarm** или любой другой).

Просмотреть же активные события/аварии возможно с помощью объекта «активные аварии».

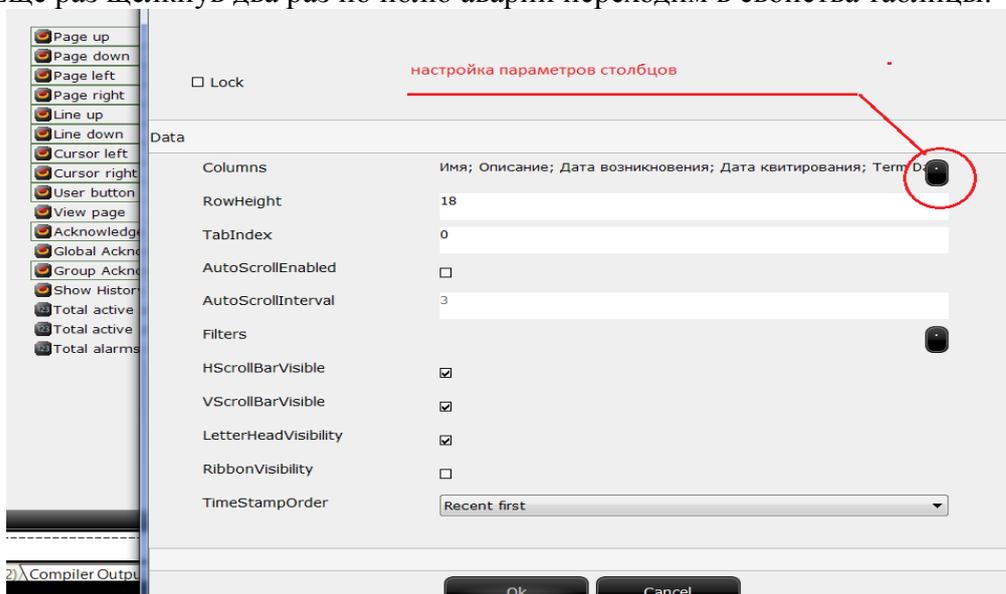
В правой части меню пиктограмм есть три объекта, с помощью которых можно создавать различные журналы: «история событий/аварий», «активные аварии», «журнал данных» (для просмотра данных в опции **Data archive** дерева проекта).

Для этого щелкните два раза по правому полю обозначений, нажмите на появившуюся справа точку и в поле **Text** возникшего поля введите свой вариант обозначения.

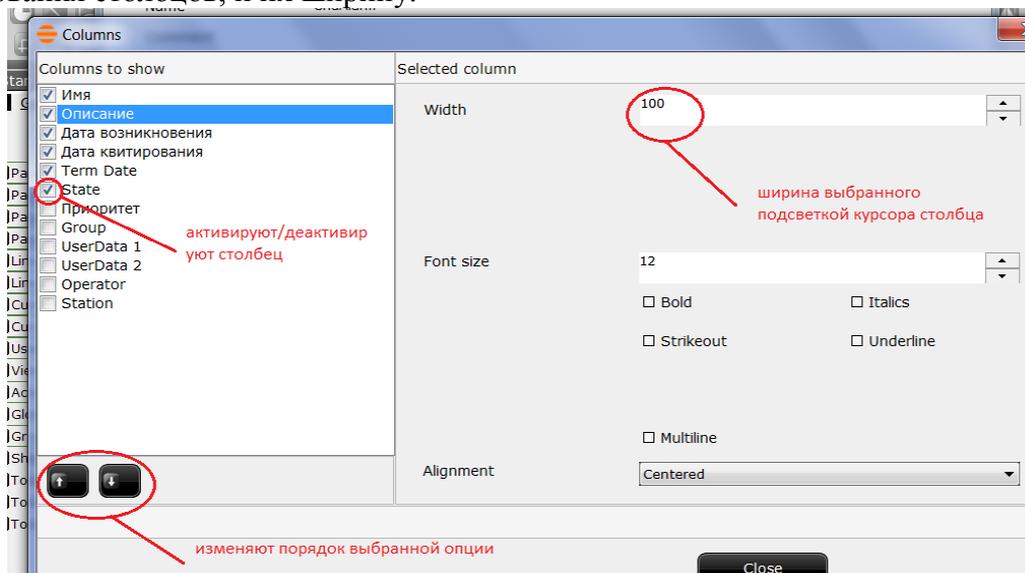
Для редактирования полей журнала(цвета, шрифта, кол-ва столбцов) щелкните два раза ЛКМ по журнала на экране для перехода в свойства объекта **ActiveAlarmViewer**.



Еще раз щелкнув два раза по полю аварий переходим в свойства таблицы:



В настройках параметров столбцов можно установить количество, порядок следования столбцов, и их ширину:



Опция **Alarm**, файл **Alarm**: в этом файле находится вся информация по данной аварии, начиная с адреса, отвечающего за эту аварию, и заканчивая различными настройками. Таких файлов ровно столько, сколько мы добавили в опции **Alarm** в закладке **List**

Закладка **Properties**:

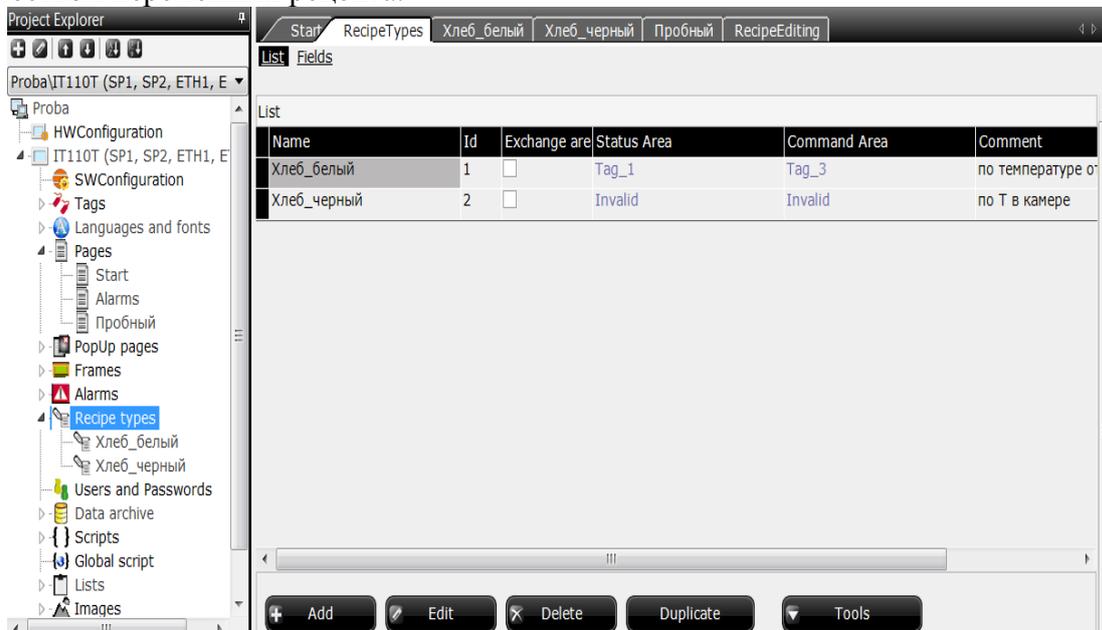
Далее на странице, листая прокрутку страницы:

Рецепты (Recipe Types)

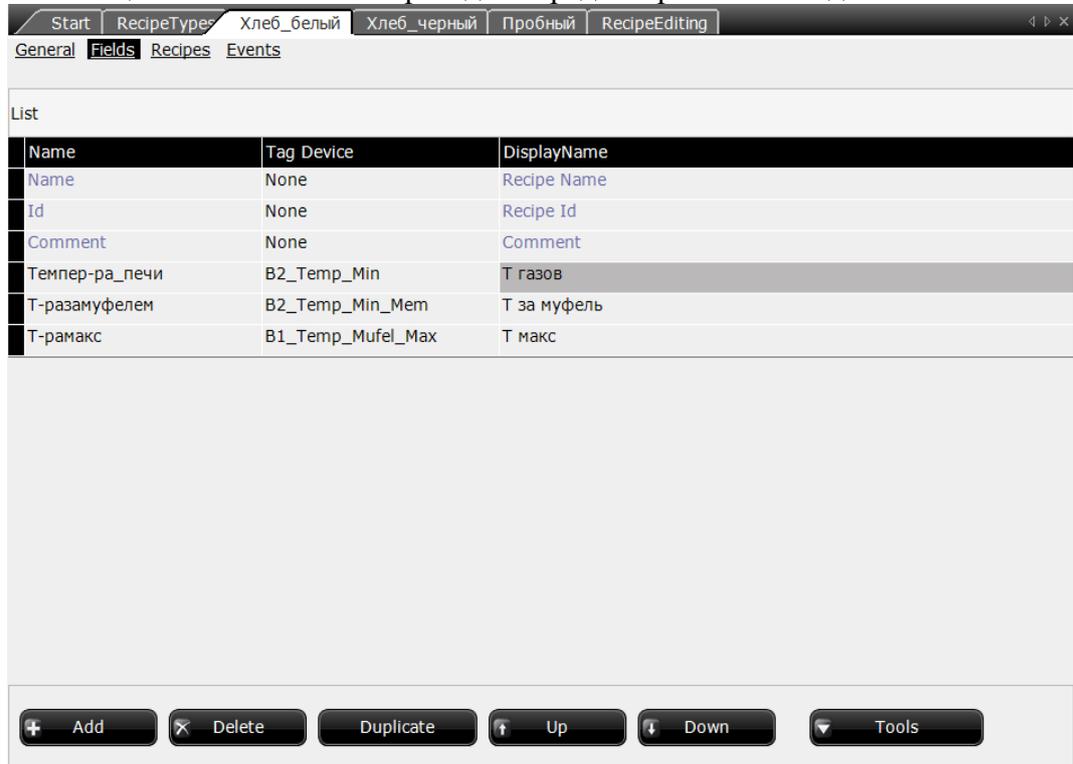
Порядок создания рецептов:

Создаются шаблоны рецептов, каждый со своим количеством переменных, отвечающих параметрам процесса и их названием. Далее под каждый шаблон на отдельном экране можно создать объект редактирования рецепта и объект просмотра и выбора рецепта из ранее введенных. Это разделение позволяет организовать различные уровни доступа для создания и редактирования рецептов (для технолога) и для использования рецептов (оператора).

Создание шаблона- опция **Recipe Type**, закладка **List** с помощью клавиши **Add** позволяет создавать необходимое количество различных шаблонов, каждый со своим количеством переменных рецепта.

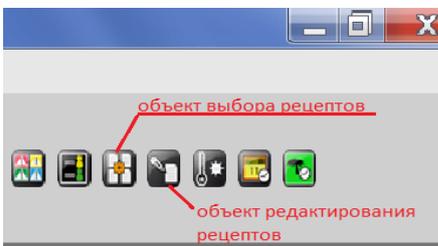


С помощью клавиши **Edit** переходим к редактированию созданных шаблонов:

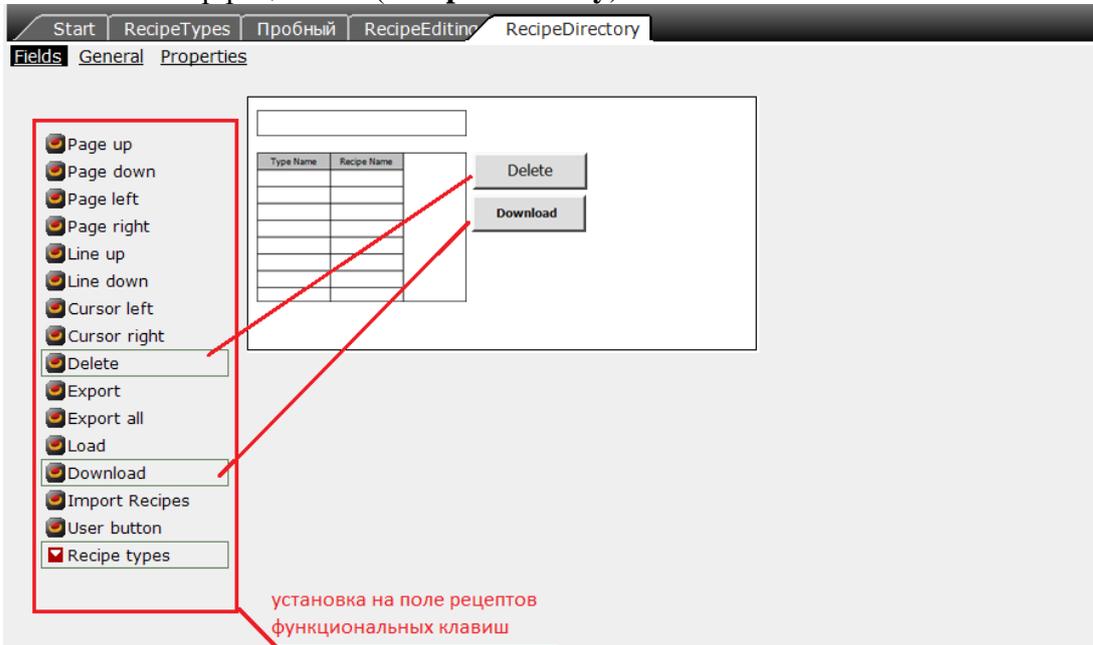


С помощью клавиши **Add** добавляем количество переменных рецепта и присваиваем переменным теги, а так же изменяем имена переменных, отображаемых на экране.

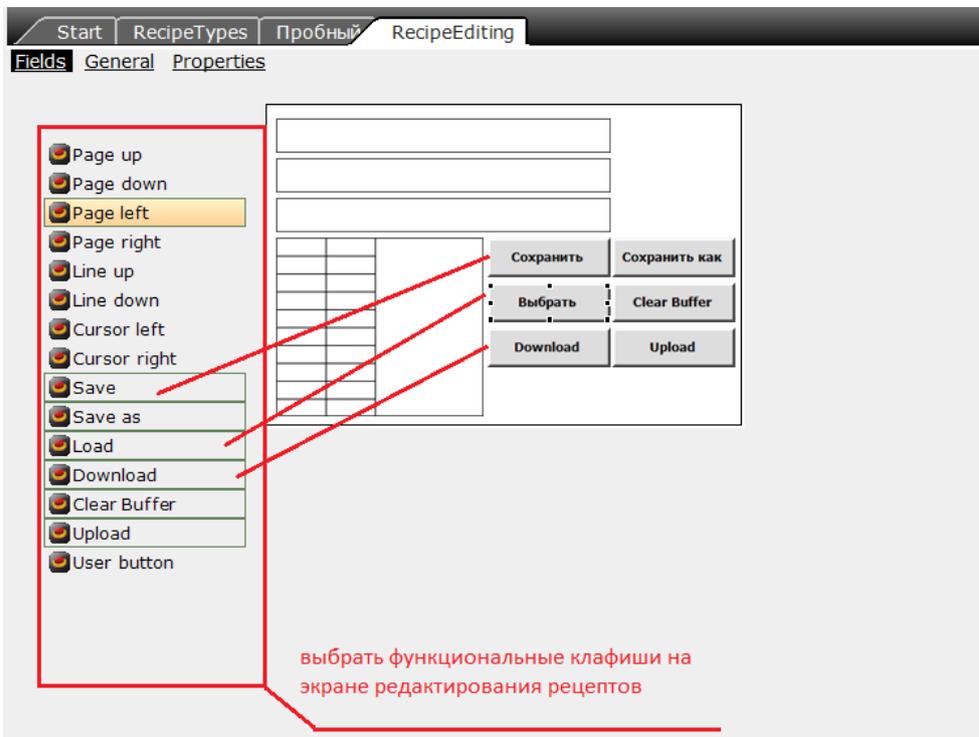
Далее с помощью пиктограмм на разных экранах расставляются объекты выбора рецептов и редактирования:



Объект «Выбор рецептов» (**RecipeDirectory**)



Объект «Редактирование рецептов» (**RecipeEdition**)



Щелкнув по функциональной кнопке в поле рецептов с правой стороны, попадаете в закладку редактирования свойств кнопки. Здесь можно отредактировать надпись на кнопке:

Lock

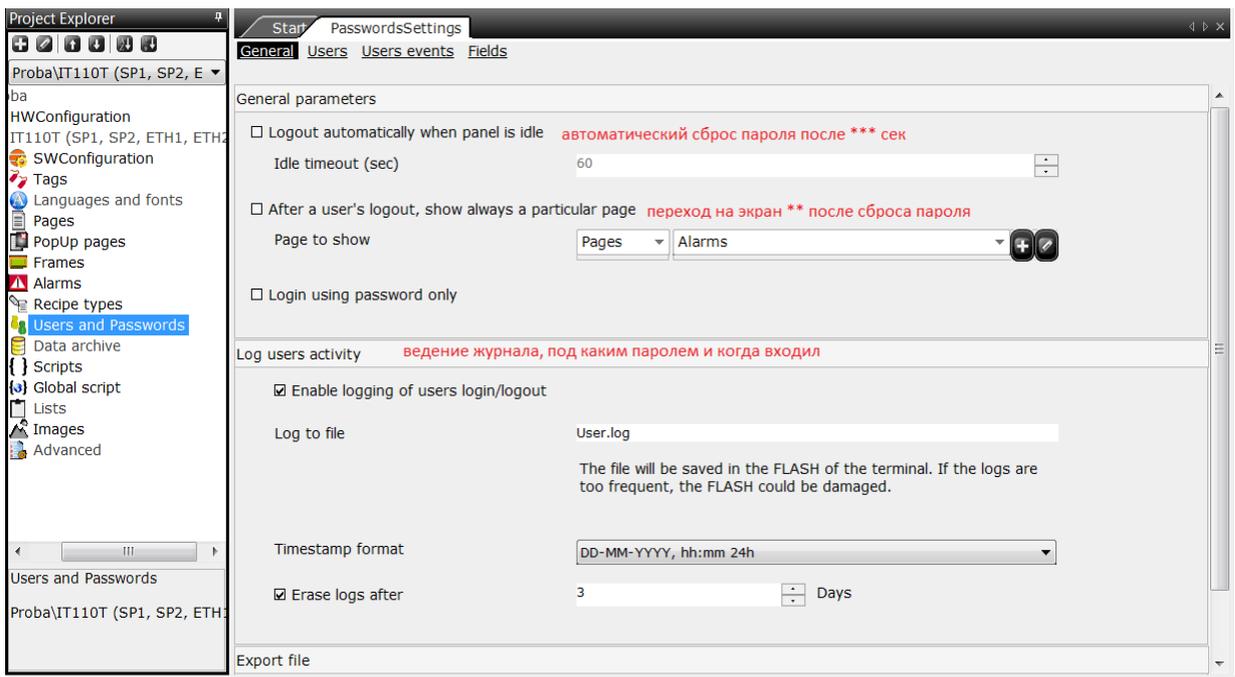
Data

| | | |
|---------------|-----------|-------------------|
| Bitmap | None | |
| Image | a | + ✎ |
| Image list id | ImageList | + ✎ |
| Image Tag id | Tags | Adjust_Zone_1 + ✎ |
| Caption | Label | |
| Text | Выбрать | ⌵ |
| Text list id | | + ✎ |
| Text Tag id | Tags | Adjust_Zone_1 + ✎ |

Выбрать надпись на кнопке

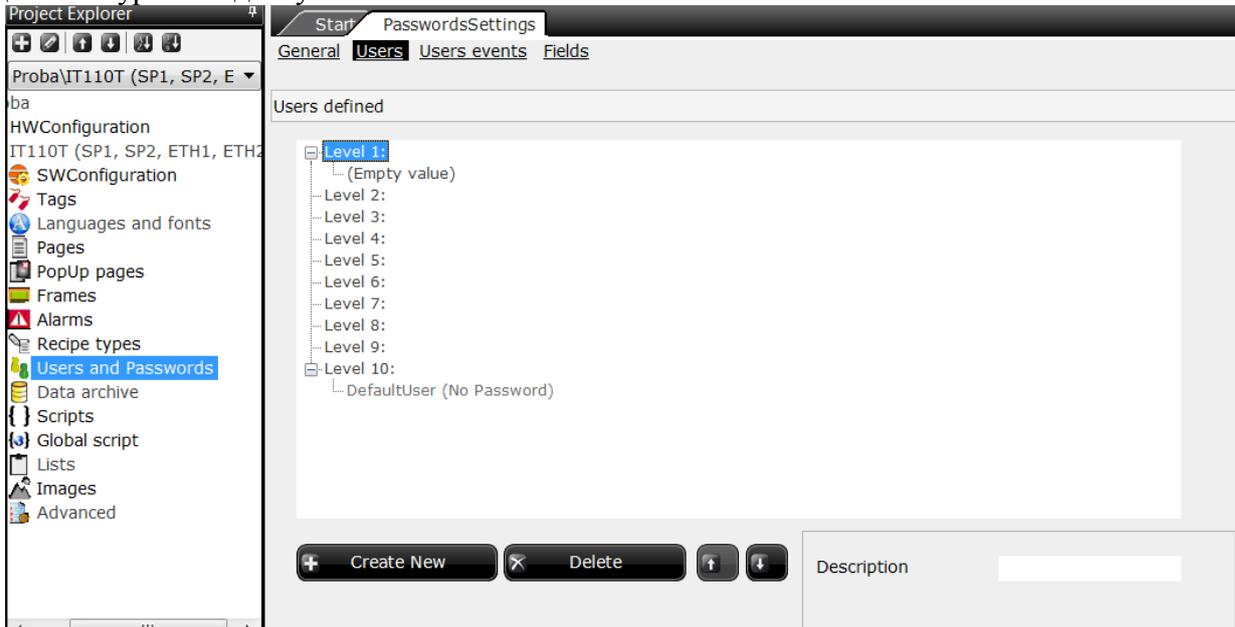
Ok Cancel

Уровни доступа и пароли (Users and Passwords)



Закладка Users

В терминалах ESA происходит ограничение доступа к объектам, имеющим определённый уровень доступа.

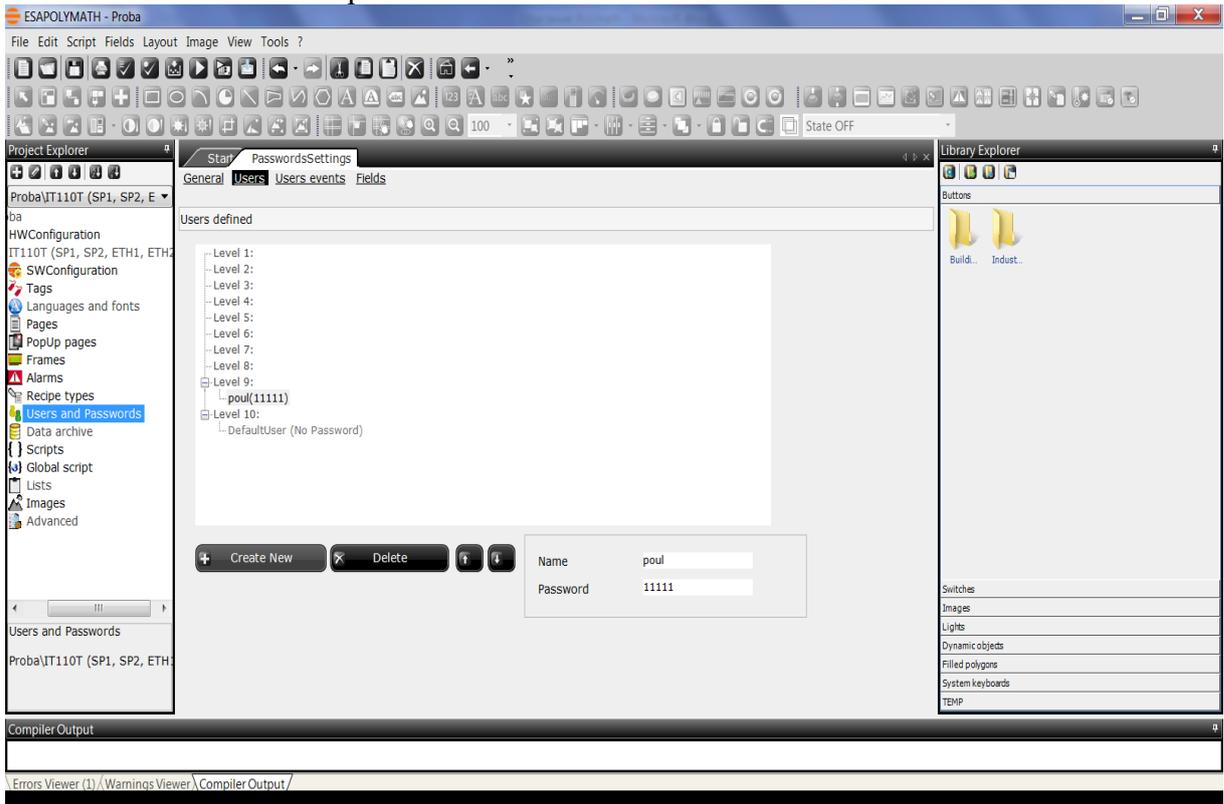


Уровни доступа – иерархические:

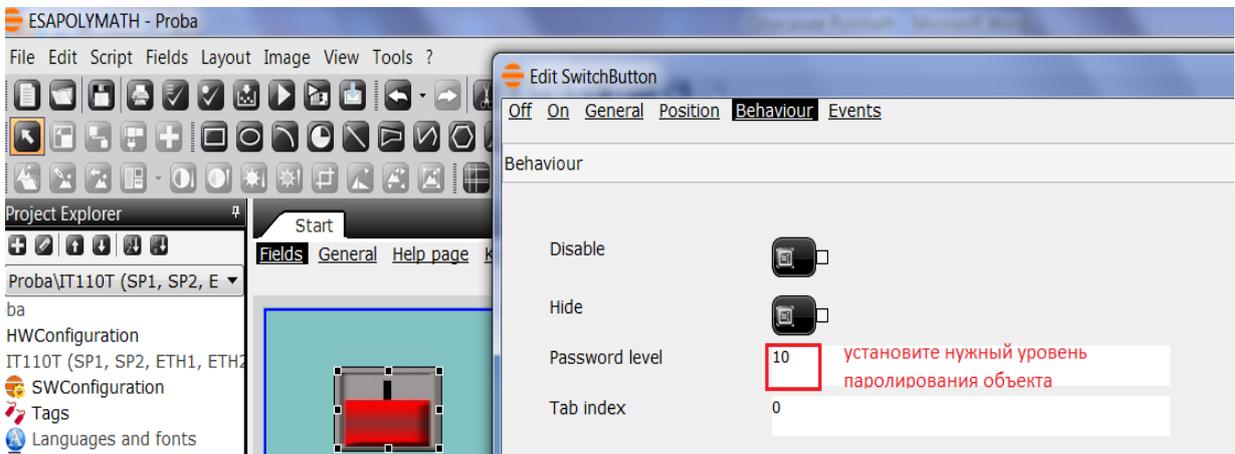
Уровень 10 – без паролирования даёт доступ всем

Уровень 1 – даёт возможность доступа к объектам всех уровней паролирования.

Для активации пароля выберите уровень, нажмите **Create New** и в появившихся окнах введите имя пользователя и пароль:

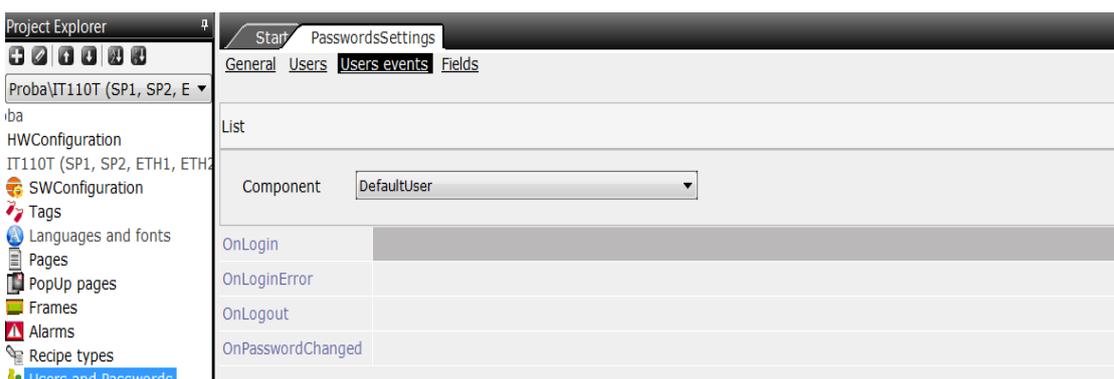


Далее в проекте объектам на экране присваивается уровень пароля. Например, для переключателя:

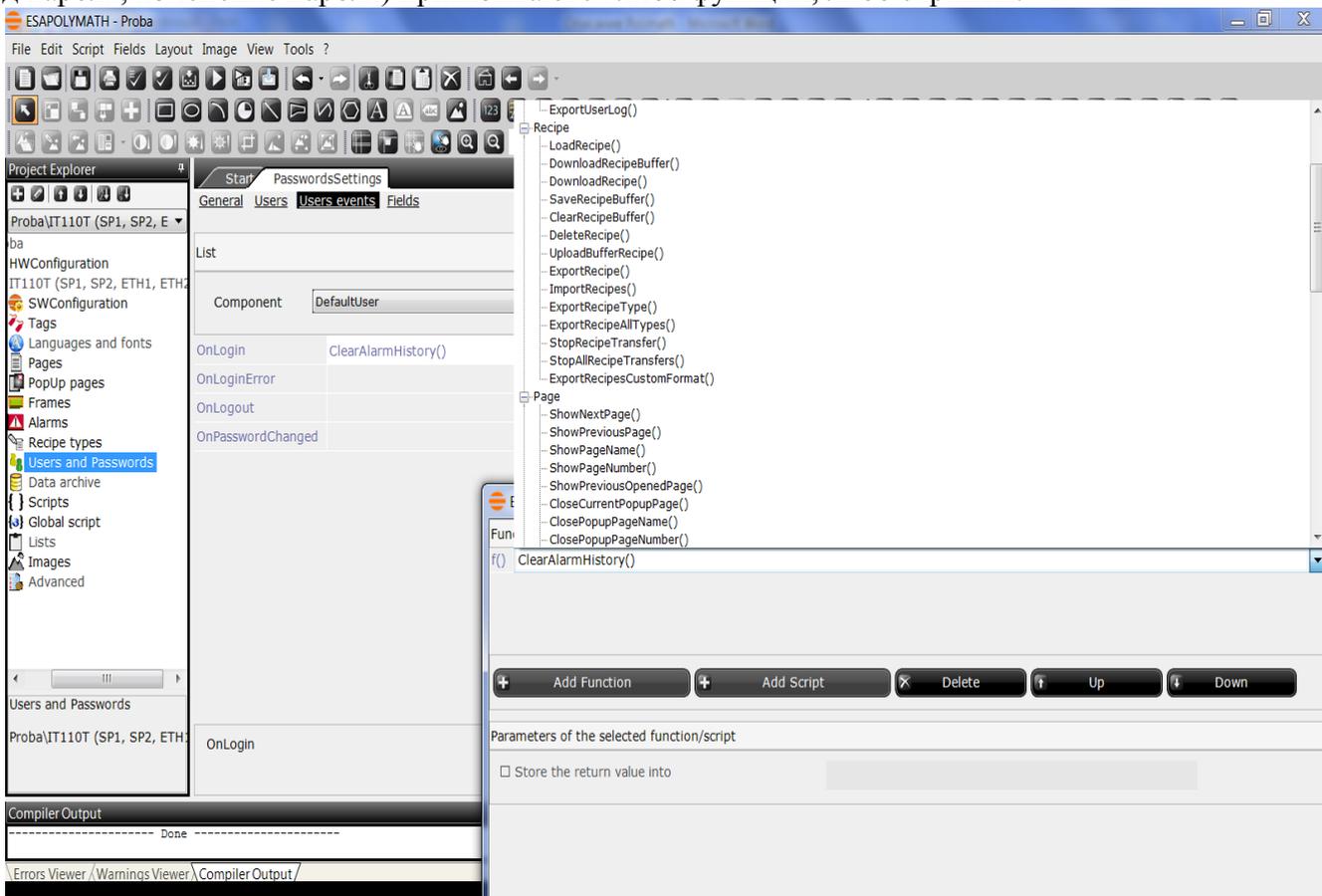


Рекомендация: в процессе работы над проектом иногда приходится изменять язык ввода и тип клавиатуры. Для предотвращения невозможности доступа к объекту, имя и пароль рекомендуется вводить с помощью цифр.

Закладка **Users events**



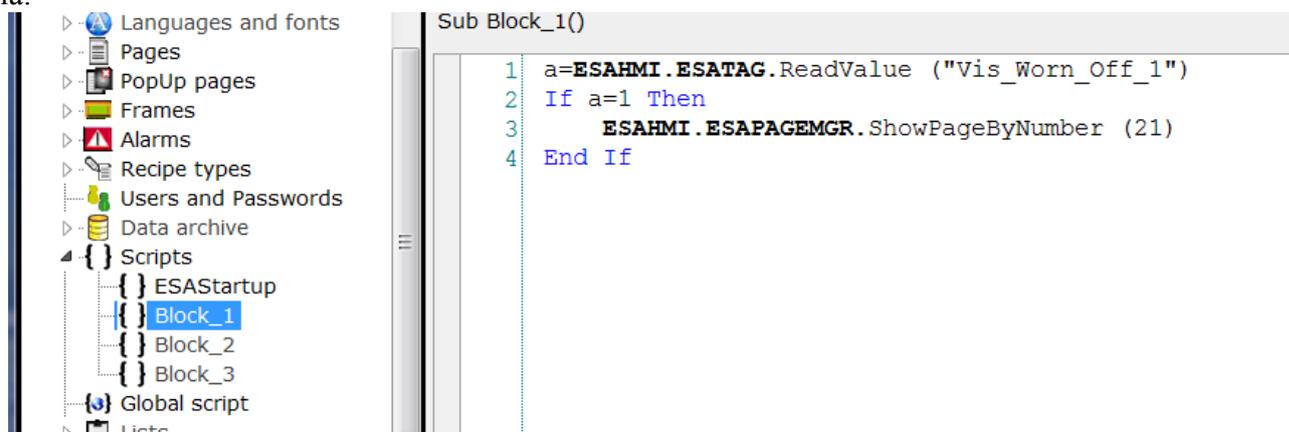
В этой закладке определённым событиям (вход под паролем, неверный пароль, выход из-под пароля, изменение пароля) привязываются либо функции, либо скрипты.



Скрипты (Scripts)

В этой папки описываются подпрограммы (скрипты), которые далее будут использоваться в различных объектах при выполнении условий их запуска.

На картинке ниже приведен пример скрипта, который используется для открытия экрана:



Основой является Visual Basic дополненный специальными командами по управлению элементами проекта, которые начинаются с ESAHMI.

В дереве проекта в папке **Scripts** созданы три скрипта с названиями **Block_1**, **Block_2**, **Block_3**.

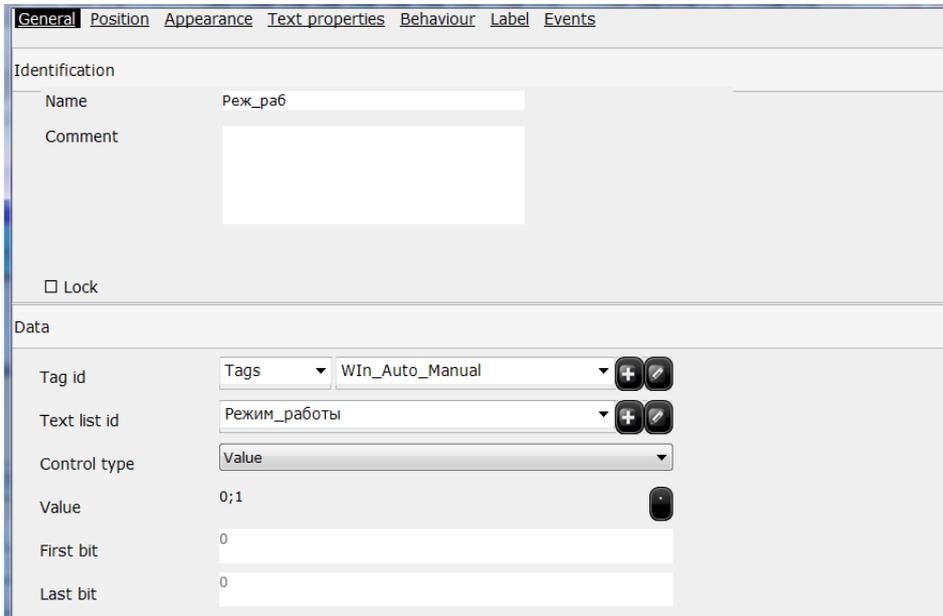
В поле слева приведен пример синтаксиса скрипта **Block_1**, где внутренней переменной **a** присваивается значение тега с именем **Vis_Worn_Off**. Далее при равенстве переменной, **a** единице (значение тега равно единице) открывается страница с номером 21.

Списки (List)

С помощью этой опции можно по значению переменной менять в одном и том же поле либо надписи, либо картинки. В папке **Text** создаем списки надписей, а в папке **Image** – списки картинок, а в самих списке добавляем необходимые надписи или картинки.

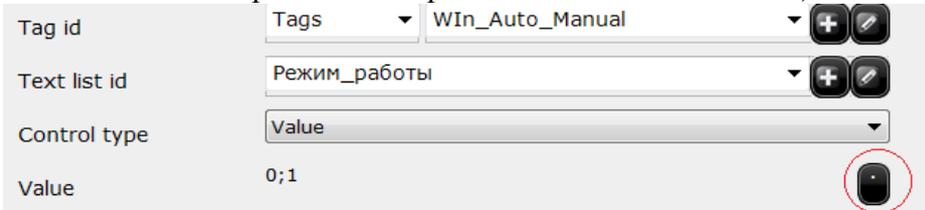
На экраны текстовые объекты устанавливаются с помощью функции **Create a dynamic**

text . Закладка **General**



- **Tag ID** – переменная, отвечающая за изменение текста в окне;
- **Text list id**- название текста, заранее созданного в опции **List/Text** дерева проектов;
- **Control type** – переменная **Tag ID** может быть как битом (две сменяющие друг друга надписи), так и словом;
- **Value** – значения, которое может принимать переменная **Tag ID** в зависимости от количества меняющихся текстов.

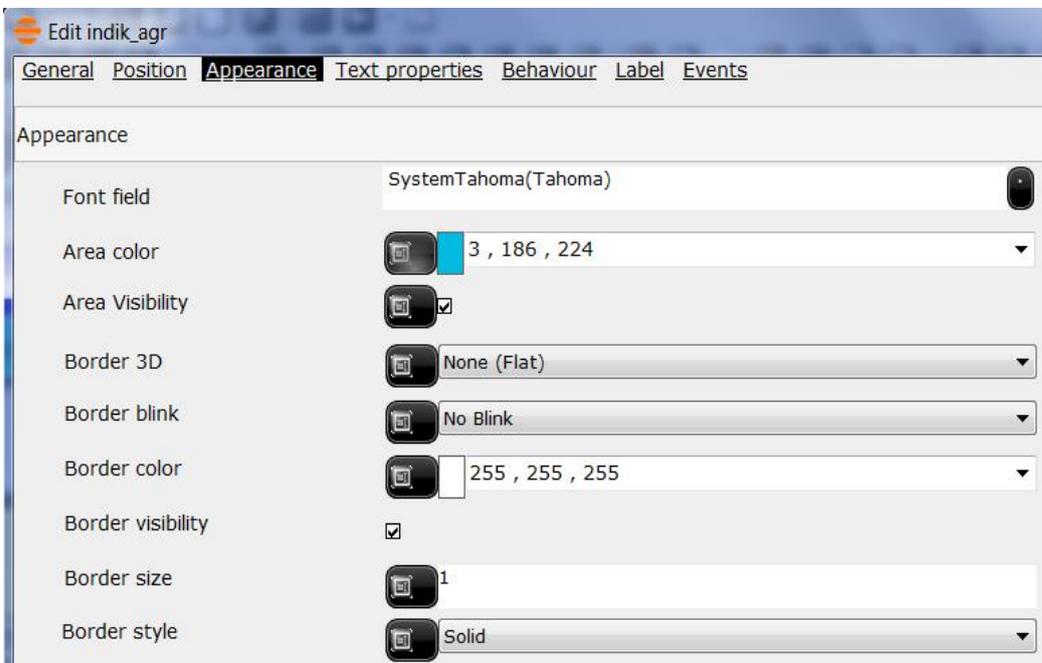
Если нажать на черное поле справа значений в поле **Value**,



то мы перейдём к редактированию соответствия текстов и значения переменной:

| Text | Value |
|----------------------------------|-------|
| Инициализация ситемы управ-я | 0 |
| Не собраны цепи управления | 10 |
| Цепи управления собраны | 2 |
| Готов к работе в ручном режиме | 3 |
| Готов к работе в автомат. режиме | 12 |
| Идёт продувка газоходов | 5 |
| Продувка завершена | 6 |
| Нормальная работа агрегата | 7 |
| Авария систем агрегата | 8 |
| Блокировка агрегата по паролю | 1 |

Закладка **Appearance** описывает свойства объекта, установленного на экране:



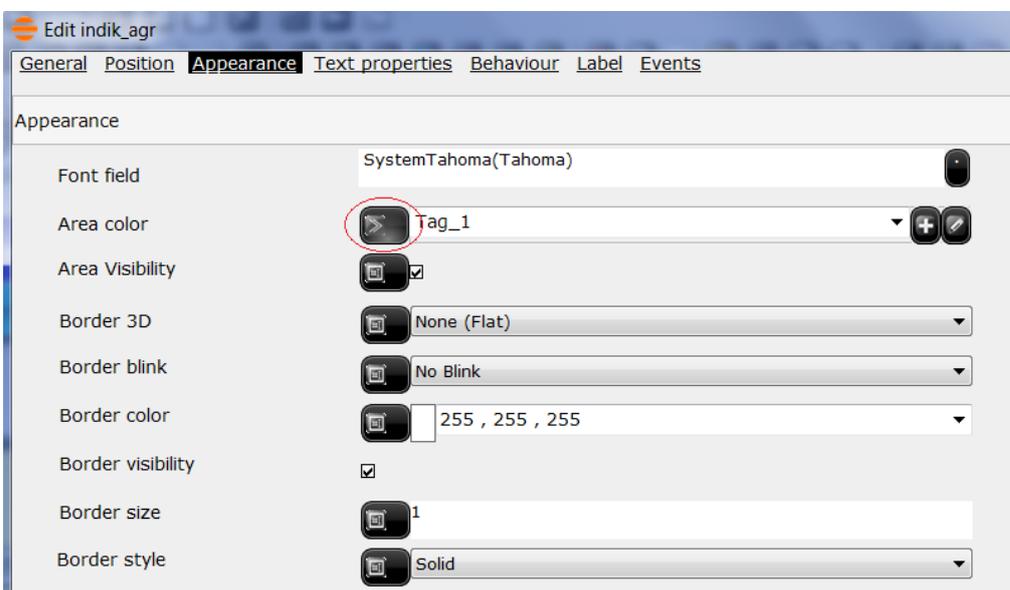
- **Font field**- при нажатии на черное поле правее, переходим в экран редактирования свойств шрифта текста:



где можно изменять тип шрифта, размер шрифта, и его свойства.

- **Aria color** – задаётся цвет поля текста. Этот цвет можно ввести непосредственно, либо задать двумя способами:

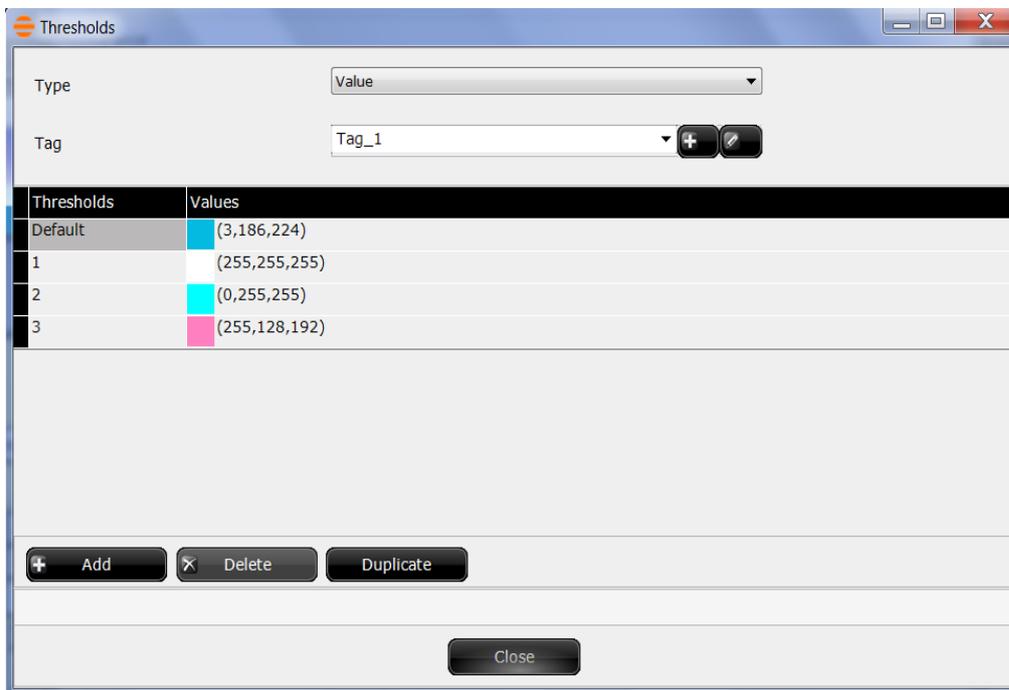
Первый: нажимайте на поле, обведенное красным цветом на рисунке ниже, пока на поле не появится стрелка и в поле левее не появится название тега:



Следует выбрать тег, который будет содержать номер цвета. В зависимости от значения, содержащегося в этой переменной, будет меняться цвет фона.

Второй способ: нажмите на черное поле до появления значка  и надписи **Value**

При нажатии на значек  правее, переходим в поле редактирования соответствия цвета значению переменной:



В поле **Type** выбирается тип переменной: бит или слово.

В поле **Tag** выбирается переменная, от значения которой будет зависеть цвет

В столбце **Thresholds** устанавливаются значения переменной, а в столбце **Values** подбираются соответствующие цвета.

- **Aria Visibility** – при наличии галочки фон становится видимым, при отсутствии – прозрачным.

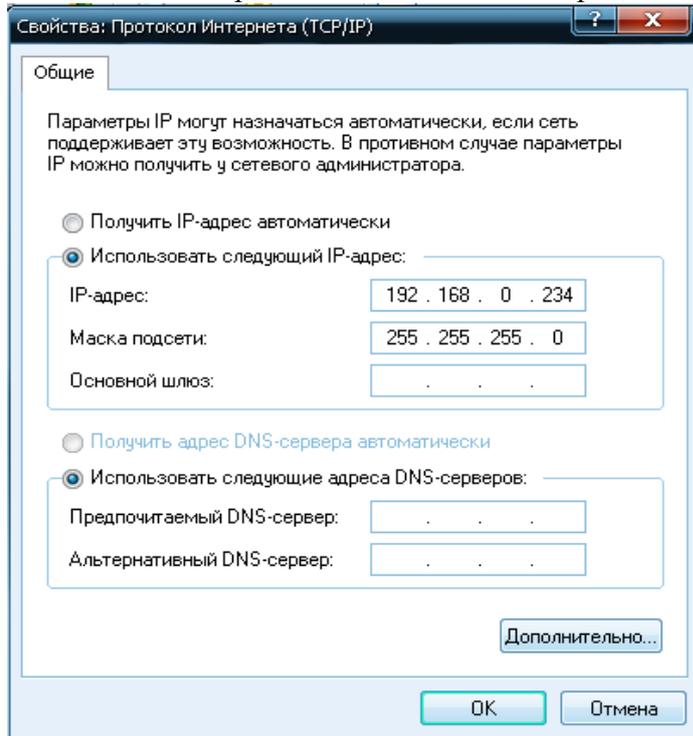
Password level – установление уровня доступа по паролю к объекту.

Динамические рисунки выводятся на экран с помощью объекта **Create a symbolic field**  на панели инструментов. Остальные закладки аналогичны функции **Create a dynamic text**.

Приложение.

Подключение панели к компьютеру по Ethernet

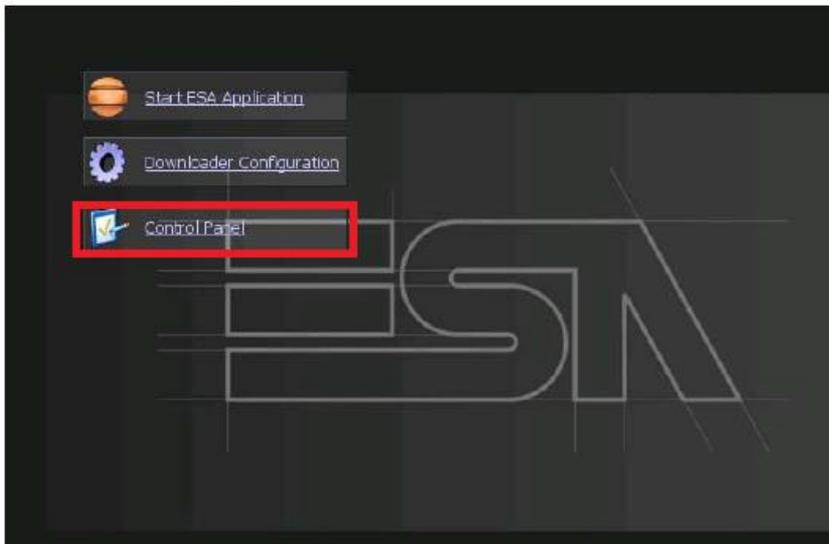
1. Настраиваем сеть в компьютере

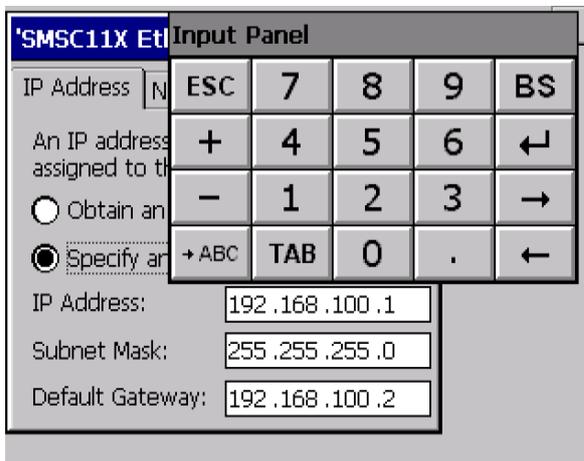


IP- адрес в компьютере и панели должен совпадать по трем первым полям (поле – три цифры между точками) и отличаться по последнему. (в панели тогда должен быть 192.168.0.235)

Маска подсети должна совпадать в компьютере и панели.

2. Настраиваем сеть в панели





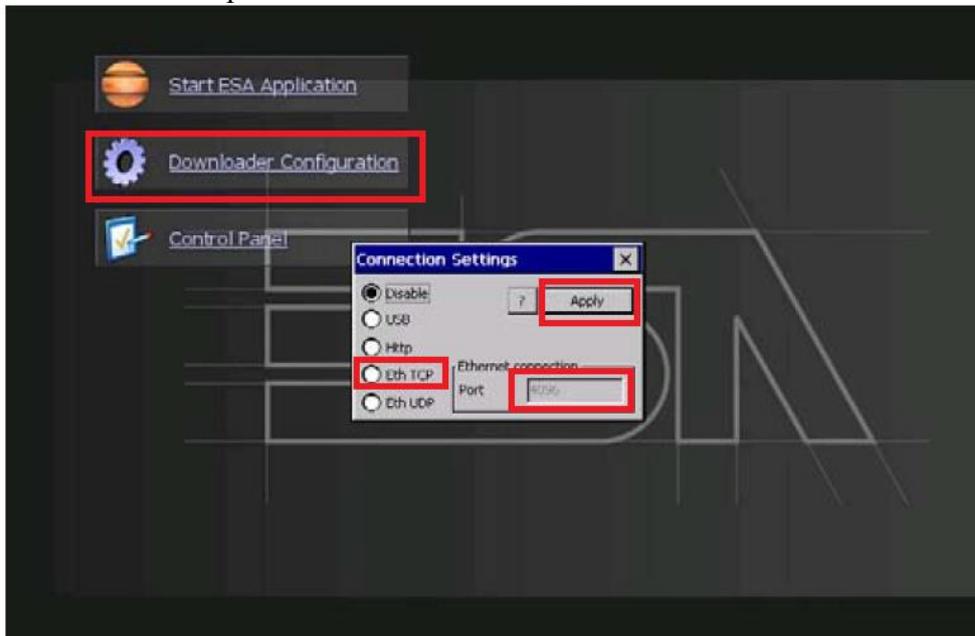
(На цифры, указанные на картинке выше, не обращаем внимания!)

IP-адрес в панели ставим 192.168.0.235

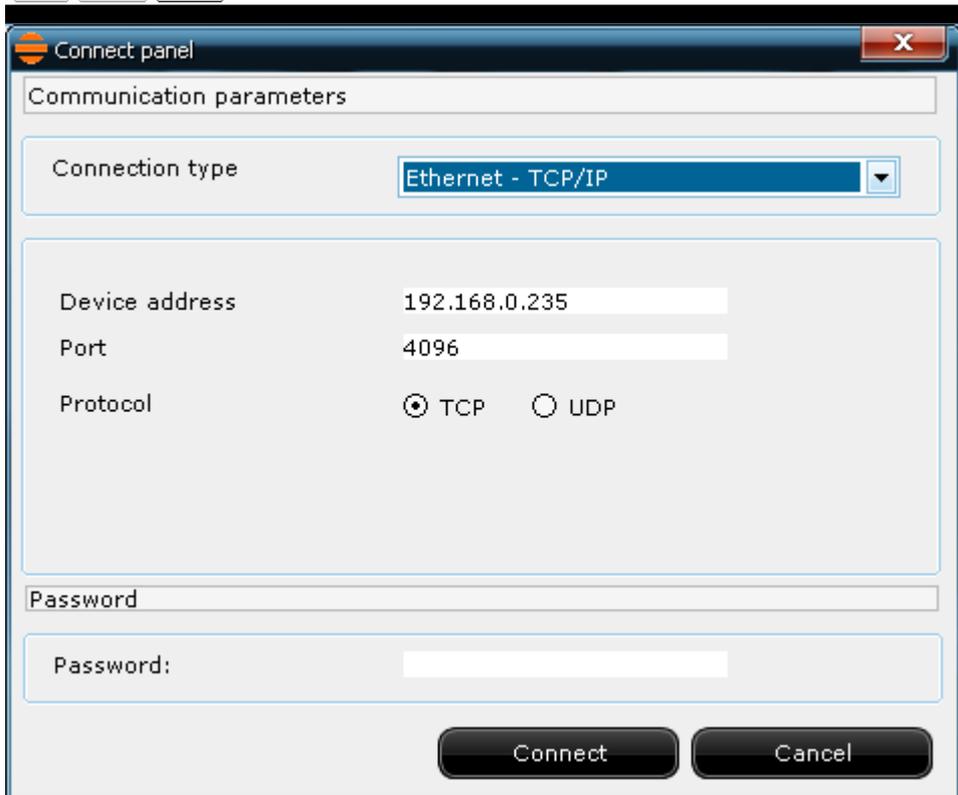
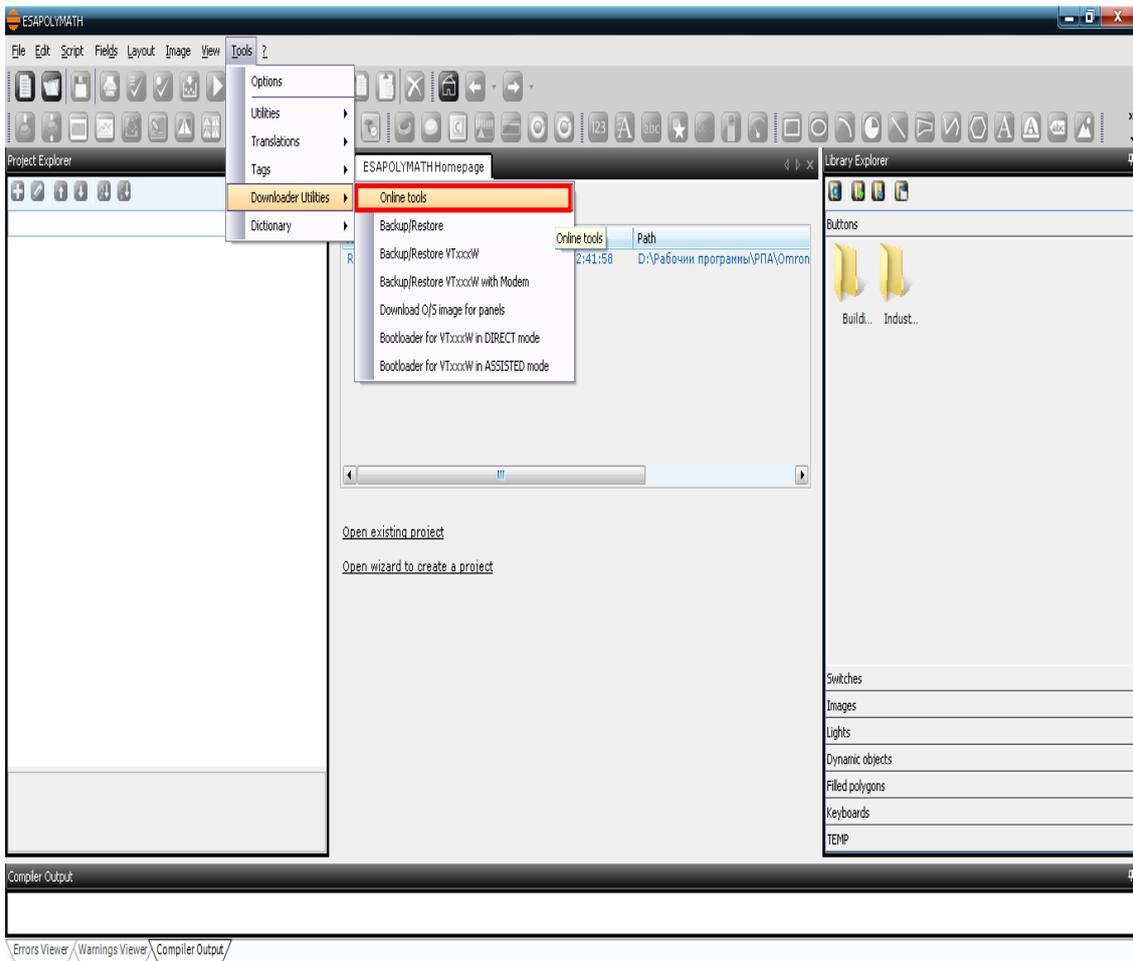
Маска подсети: 255.255.255.0 везде одинакова

Default Gateway: 192.168.0.236 – три первых поля совпадают, последнее отличается от панели и компьютера.

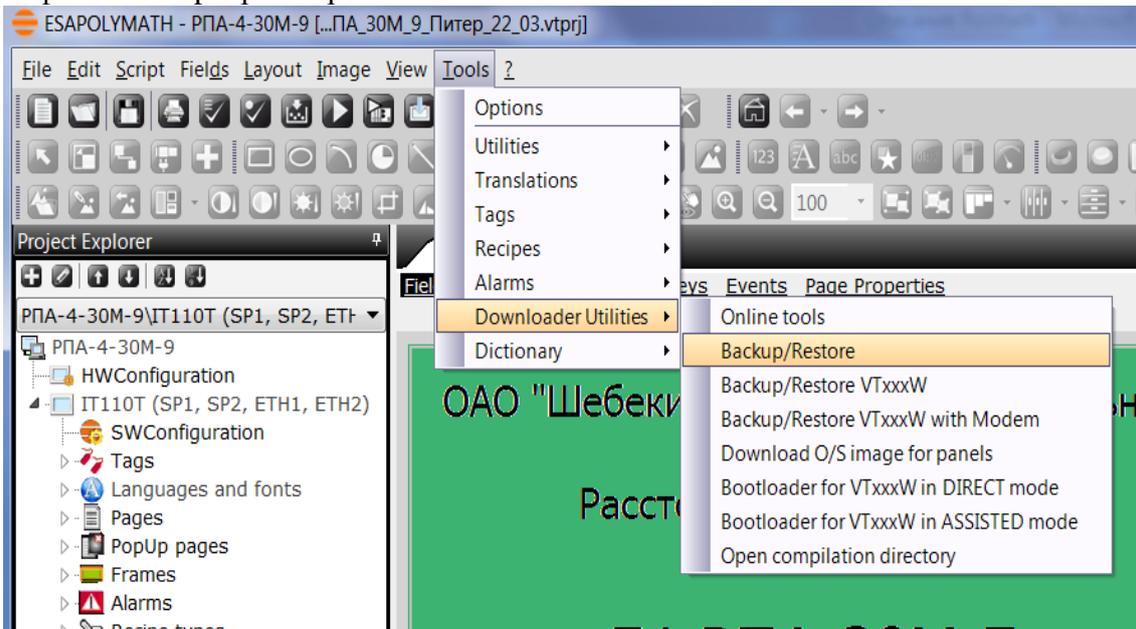
3. Настраиваем в панели тип подключения – Ethernet



4. Настраиваем тип подключения в Polymath



Для загрузки проекта в терминал с помощью USB-накопителя следует сначала создать файл Backup с расширением .ebk

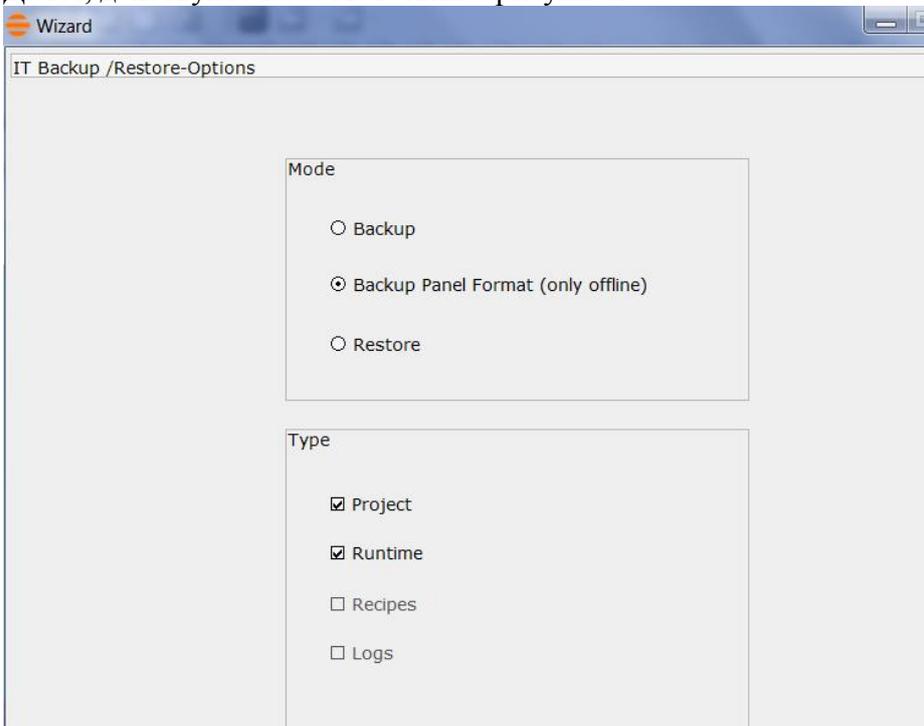


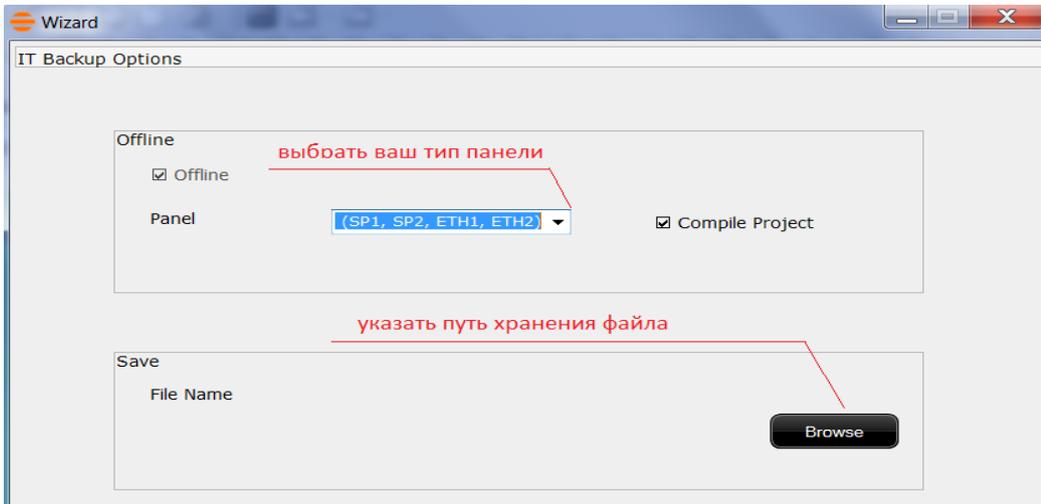
Для этого выбрать опции, изображенные на картинке выше.

Далее, в зависимости от типа вашего оборудования в появившемся окне следует выбрать:



Далее, действуйте в соответствии с рисунком:





Дождитесь компиляции проекта.

Для загрузки в панель перепишите файл с расширением **.ebk** в корень USB-накопителя, вставьте в порт USB-A панели, войдите в системное меню **Control panel**, выберите функцию **Backup** далее **Backup-Restore**. Для записи проекта в панель выберите **Restore**. Далее, нажмите слева поля на «...», выберите **Hard Disk 2** и нажмите **OK**, найдите и выберите курсором проект, нажмите «**OK**» и ожидайте около минуты, пока проект не распакуется и не загрузится.

После выбора типа панели и типа соответствующего контроллера с левой стороны будет создано дерево проекта и будет предоставлена возможность изменить язык интерфейса. Для этого воспользуйтесь опцией в главном меню **Tools/Options/Language**

