

Министерство образования Республики Беларусь  
Учреждение образования  
«Гомельский государственный технический университет  
имени П. О. Сухого»

Кафедра «Автоматизированный электропривод»

В. А. Савельев, И. В. Дорощенко

## ПРОГРАММИРУЕМЫЕ КОНТРОЛЛЕРЫ И ПРОМЫШЛЕННЫЕ СЕТИ

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ  
для студентов специальности 1-53 01 05  
«Автоматизированные электроприводы»  
дневной формы обучения

*Учебное электронное издание  
комбинированного распространения*

Гомель 2022

УДК 681.5:62–503.55(075.8)  
ББК 32.971.322.5я73  
С13

Рецензент: доц. каф. «Информационные технологии» ГГТУ им. П. О. Сухого  
канд. техн. наук, доц. *В. С. Захаренко*

**Савельев, В. А.**

С13 Программируемые контроллеры и промышленные сети : учеб.-метод. пособие для студентов специальности 1-53 01 05 «Автоматизированные электроприводы» днев. формы обучения / В. А. Савельев, И. В. Дорощенко. – Гомель : ГГТУ им. П. О. Сухого, 2022. – 63 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; дисковод CD-ROM ; мышь ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

ISBN 978-985-535-504-6.

Содержит методические указания по изучению основ программирования и конфигурирования программируемого логического контроллера, программируемого терминала для работы с преобразователем частоты производства Schneider Electric, а также позволяет рассмотреть основы построения систем управления с использованием современных средств автоматизации.

Для студентов специальности 1-53 01 05 «Автоматизированные электроприводы» дневной формы обучения.

УДК 681.5:62–503.55(075.8)  
ББК 32.971.322.5я73

ISBN 978-985-535-504-6

© Савельев В. А., Дорощенко И. В., 2022  
© Учреждение образования «Гомельский  
государственный технический университет  
имени П. О. Сухого», 2022

## Введение

На сегодняшний день промышленная автоматизация связана с очень широким спектром продукции для автоматизации процессов: программируемые логические контроллеры (ПЛК), преобразователи частоты (ПЧ), программируемые терминалы (ПТ) и т. д.

Программируемые логические контроллеры являются в современном производстве одним из основных программно-технических средств автоматизации технологических объектов управления в различных отраслях промышленности. Разработчики и производители предлагают на рынке средств автоматизации сотни различных моделей ПЛК, различающихся техническими характеристиками, функциональными возможностями, стоимостью, средствами программирования и т. д.

Преобразователь частоты в комплекте с ПЛК может применяться для создания многофункциональных систем управления электроприводами. Применение регулируемого частотного электропривода позволяет экономить электроэнергию устранением неоправданных ее затрат, которые имеют место быть при альтернативных методах регулирования.

Программируемые терминалы используются в качестве панелей оператора для эффективного контроля и управления технологическим оборудованием в режиме реального времени. Эти устройства позволяют отображать информацию как в графическом, так и в символьном виде, а также дают возможность вводить требуемые данные.

В этих условиях инженеру-электрику, занимающемуся проектированием, наладкой и эксплуатацией автоматизированных систем управления технологическими процессами важно знать структурно-функциональную организацию и технические параметры ПЛК, ПЧ и ПТ, владеть навыками их программирования, уметь решать типовые задачи автоматизации.

Целью настоящего учебно-методического пособия является изучение основных приемов работы с аппаратно-программным комплексом оборудования (ПЛК – ПЧ – ПТ) – одного из ведущих производителей *Schneider Electric* – и получение практических навыков работы с системами автоматизации.

## 1. Краткая характеристика аппаратной части

На рис. 1.1 приведено расположение элементов ПЛК *Modicon M221*.

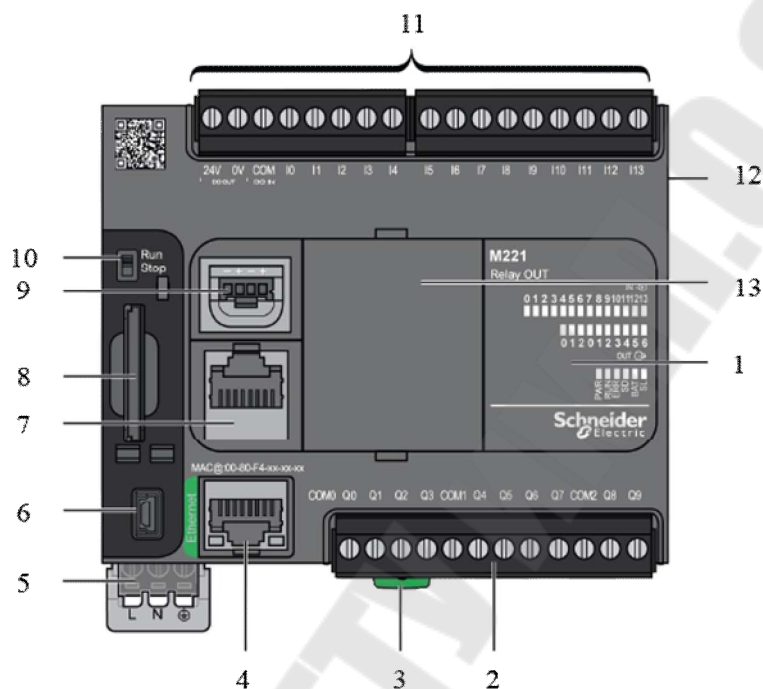


Рис. 1.1. Расположение элементов программируемого логического контроллера:

- 1 – индикаторы состояния; 2 – выходная съемная клеммная колодка;
- 3 – замок для *DIN*-рейки; 4 – порт *Ethernet* (*RJ-45*);
- 5 – источник питания 24 В постоянного тока; 6 – порт *USB* для подключения к ПК; 7 – последовательный порт *RS-232/485* (*RJ-45*);
- 8 – слот для *SD*-карты; 9 – 2 аналоговых входа;
- 10 – переключатель Пуск/Стоп; 11 – входная съемная клеммная колодка;
- 12 – разъем расширения ввода/вывода; 13 – слот для картриджа

Характеристики контроллера:

- 256 Кбайт для внутренних переменных *RAM*;
- 256 Кбайт флэш-памяти для резервного копирования приложений и данных;
- 256 Кбайт для пользовательского применения и данные *RAM* с 10000 инструкций;
- *SD*-карта до 2 Гб опциональна для хранения программ и рецептов;
- скорость исполнения инструкции: 0,2 мкс булевы инструкции, 0,3 мс – событийные и периодические задания, 60 мкс – иные инструкции;

– порт *Ethernet RJ45* (10/100 Мбит/с, интерфейс *MDI/MDIX*) с поддержкой протокола *Modbus TCP* (Клиент/Сервер), *Ethernet IP* (адаптер), протоколы *UDP, TCP*;

– порт *Serial Link 1-RS232\RS485*, порт *Serial Link 2 RS485* (в ПЛК книжного формата);

– 2 аналоговых входа «в базе» с разрешением 10 бит;

– 4 быстрых входа и 2 быстрых выхода до 100 кГц.

Подробнее о контроллере можно узнать в [5].

На рис. 1.2 приведено расположение элементов конструкции ПЧ *Altivar ATV320*.

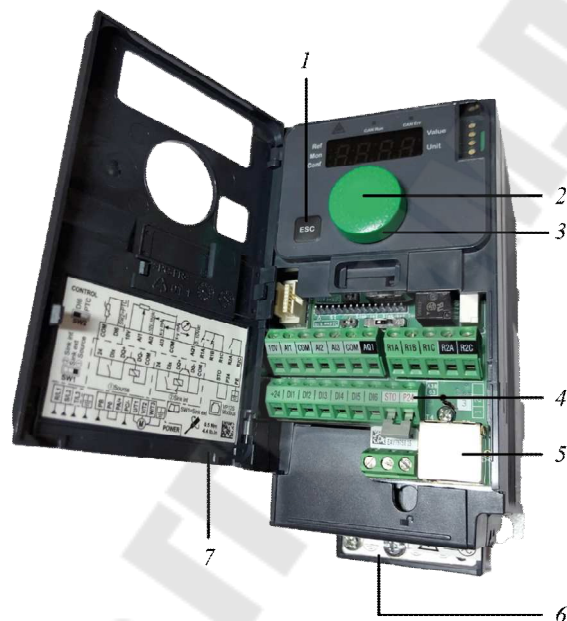


Рис. 1.2. Расположение элементов конструкции преобразователя частоты *Allivar ATV320*:

- 1 – клавиша *ESC* используется для навигации по меню (возврат) и настройки параметров (отмена); 2 – клавиша *ENT* (нажатие в центре кругового навигатора) используется для навигации по меню (следующий) и настройки параметров (подтверждение выбора); 3 – круговая навигационная клавиша используется для навигации по меню (вверх или вниз) и настройки параметров (увеличение или уменьшение значения или выбор элемента).

Она может использоваться в качестве виртуального аналогового входа 1 для задания частоты ПЧ;

4 – клеммы управления для подключения входов/выходов;

5 – порт связи *RJ-45* для доступа ко встроенным интерфейсам (последовательной шине *Modbus* и шине *CANopen*);

6 – силовые клеммы; 7 – защитная крышка для доступа к клеммам управления (включает в себя наклейку с монтажной схемой)

На рис. 1.3 показаны функции дисплея ПЧ *Altivar ATV320*.

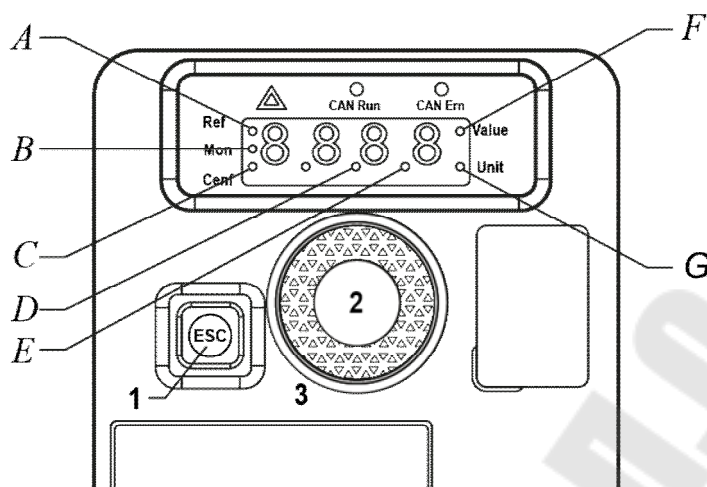


Рис. 1.3. Функции дисплея преобразователя частоты *Allivar ATV320*:

- A* – режим задания скорости; *B* – режим мониторинга;
- C* – режим конфигурации; *D* – десятичная точка используется для отображения значений параметров (1/100 значения);
- E* – десятичная точка используется для отображения значений параметров (1/10 значения); *F* – текущее значение отображаемого параметра;
- G* – единица измерения отображаемого параметра

Преобразователи серии *Altivar Machine ATV320* ориентированы на встраивание в машины механизмы с простыми и повышенными требованиями к управлению двигателем и коммуникационным возможностям.

Данная серия предоставляет расширенные возможности автоматизации и максимальную производительность при использовании в промышленных машинах и механизмах:

- надежное управление асинхронными или синхронными двигателями;
- полноценную интеграцию в коммуникационную систему любой архитектуры (*Ethernet*, *CANopen*, *Profibus*, и т. д.);
- компактное или «книжное» исполнение для встраивания во все типы шкафов;
- встроенные функции безопасности;
- увеличенная устойчивость к агрессивным средам.

В состав преобразователя входят: съемный пульт, ЭМС-фильтр, аналоговый вход, *PID*-регулятор.

На рис. 1.4 показаны элементы конструкции ПТ *Magelis GTO4310*.

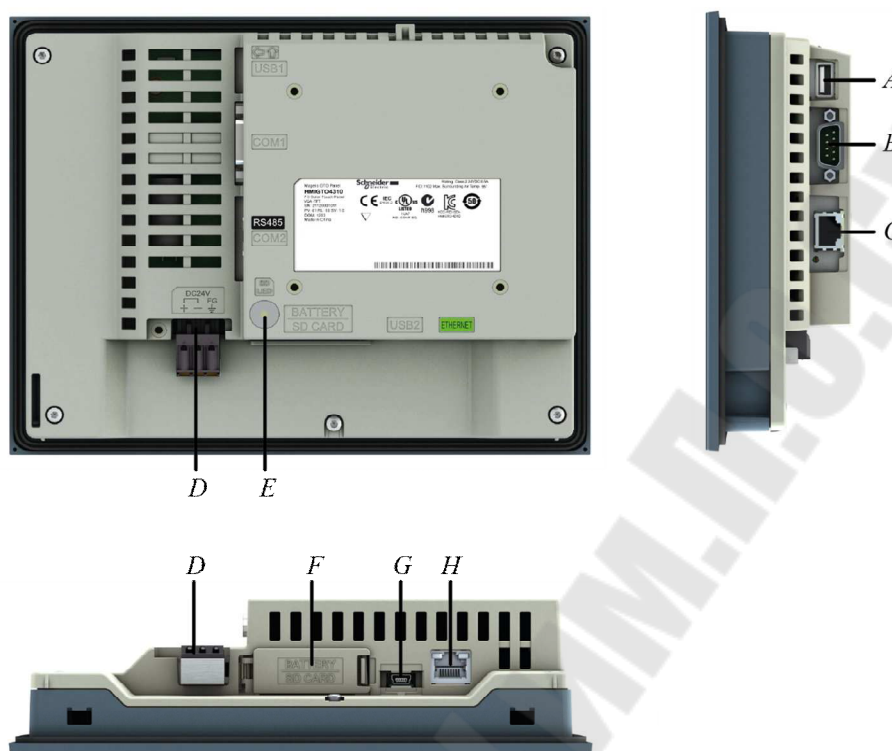


Рис. 1.4. Элементы программируемого терминала *Magelis GTO4310*:  
A – порт *USB*; B – последовательный интерфейс *RS-232C (COM1)*;  
C – последовательный интерфейс *RS-485 (COM2)*; D – разъем питания 24 В;  
E – индикатор доступа к *SD*-карте; F – крышка интерфейса *SD*-карты / Сменная крышка отсека аккумулятора; G – порт *USB*; H – интерфейс *Ethernet*

## 2. Создание нового проекта

Для программирования терминалов серии *Magelis HMI GTO* применяется программное обеспечение *Vijeo Designer*.

Для создания нового проекта в среде программирования *Vijeo Designer* необходимо выполнить действия, описанные ниже.

- Задать имя проекта (*Project Name*) и его описание (*Description or Comment*) при необходимости (рис. 2.1).

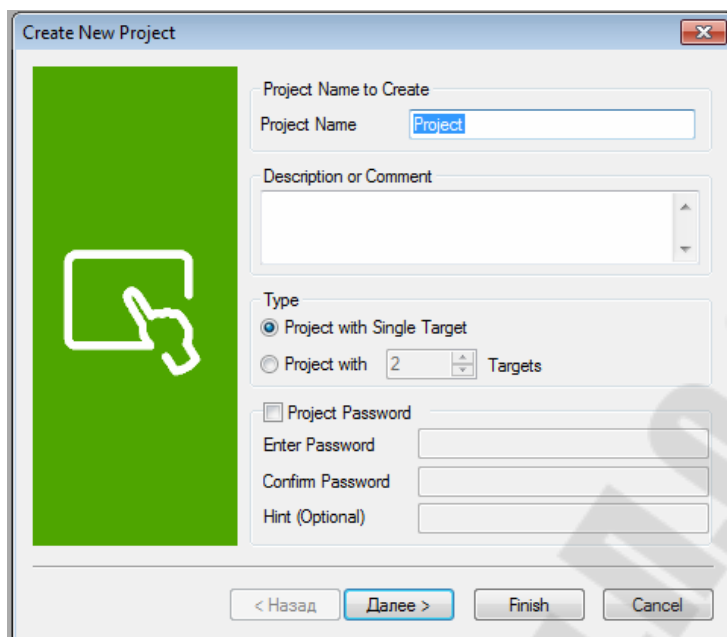


Рис. 2.1. Настройка имени проекта и его описания

- Выбрать количество целей (*target*) (под целью понимается терминал).
- При необходимости задать пароль (*Project Password*).
- Присвоить имена терминалам (*Target Name*) (рис. 2.2).

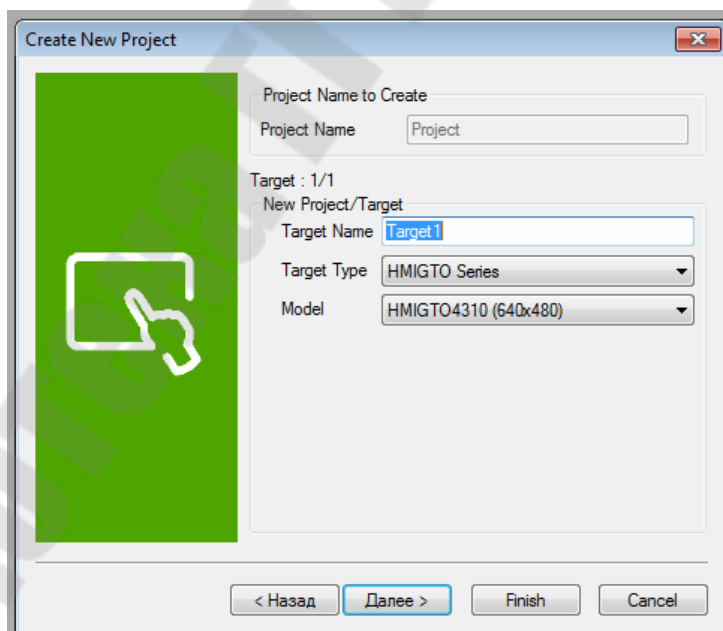


Рис. 2.2. Присвоение имени



- Выбрать тип терминала (*Target Type*) – в нашем случае это терминал серии *GTO*, модель терминала *GTO4310* с разрешением 640x480.
- Назначить *IP*-адрес терминала (например, 192.168.1.100), а так-же маску подсети (*Subnet Mask*) (255.255.255.0) (рис. 2.3).

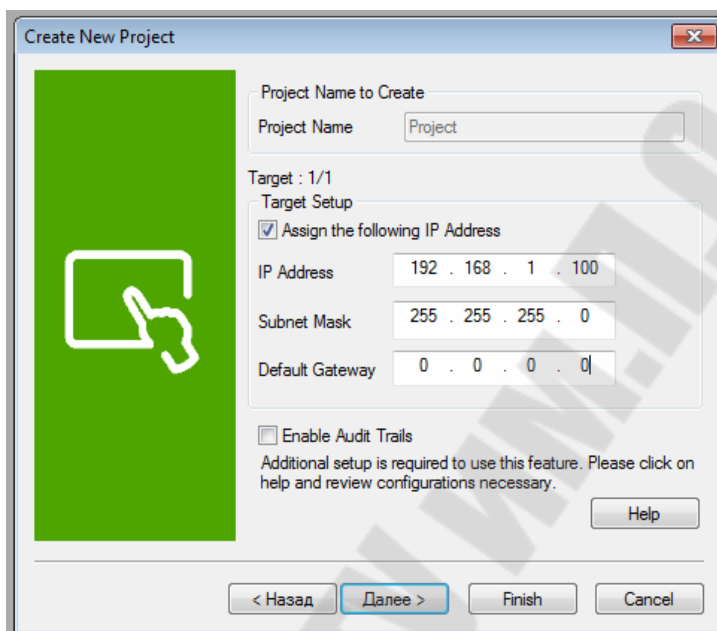


Рис. 2.3. Маска подсети

- Добавить драйверы внешнего оборудования (нажать *Add*) (рис. 2.4).

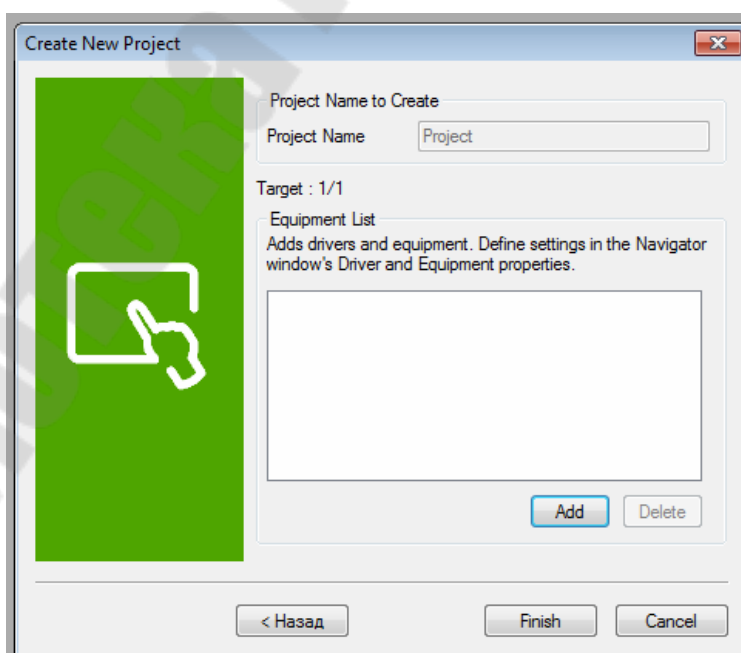


Рис. 2.4. Добавление драйверов внешнего оборудования

- Для связи с внешним оборудованием по локальной сети *Ethernet* выбрать драйвер *Modbus TCP/IP* и тип оборудования *Modbus Equipment* (рис. 2.5).

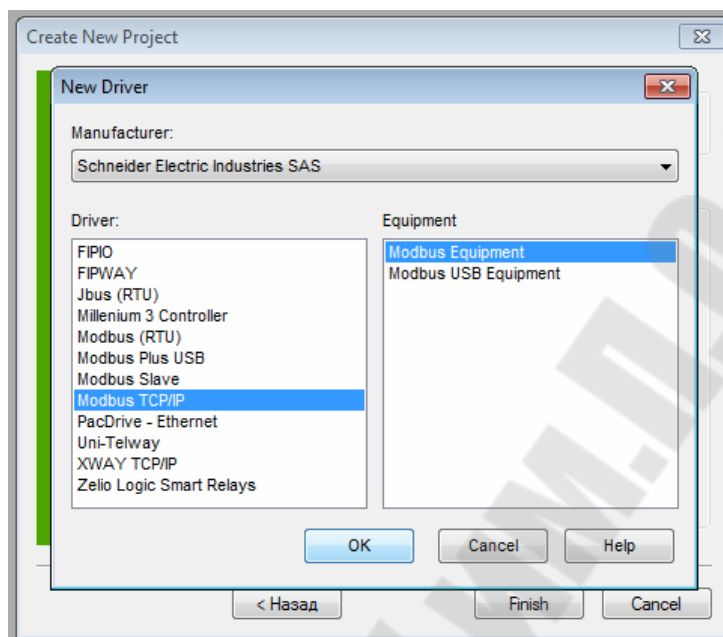


Рис. 2.5. Выбор типа оборудования *Modbus Equipment*

- Для связи с внешним оборудованием по протоколу *RS-485* выбрать драйвер *Modbus (RTU)* и тип оборудования *Modbus Equipment* (рис. 2.6, 2.7).

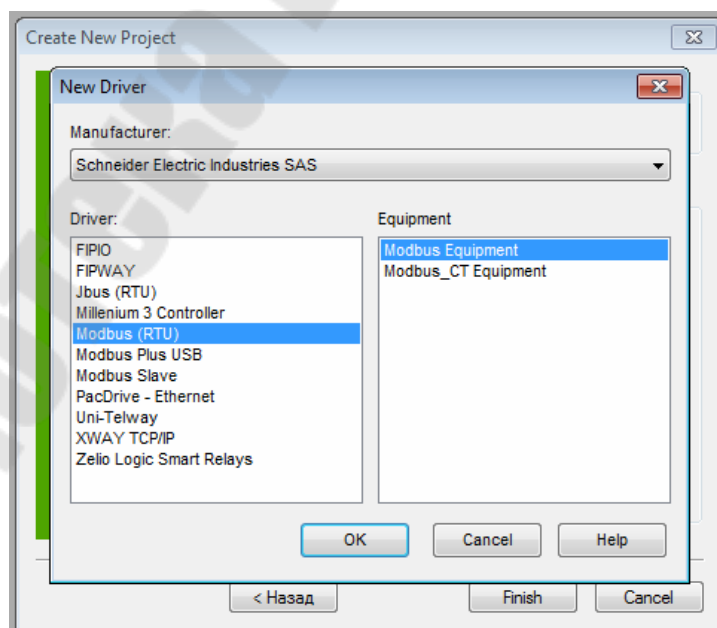


Рис. 2.6. Выбор драйвера *Modbus (RTU)*

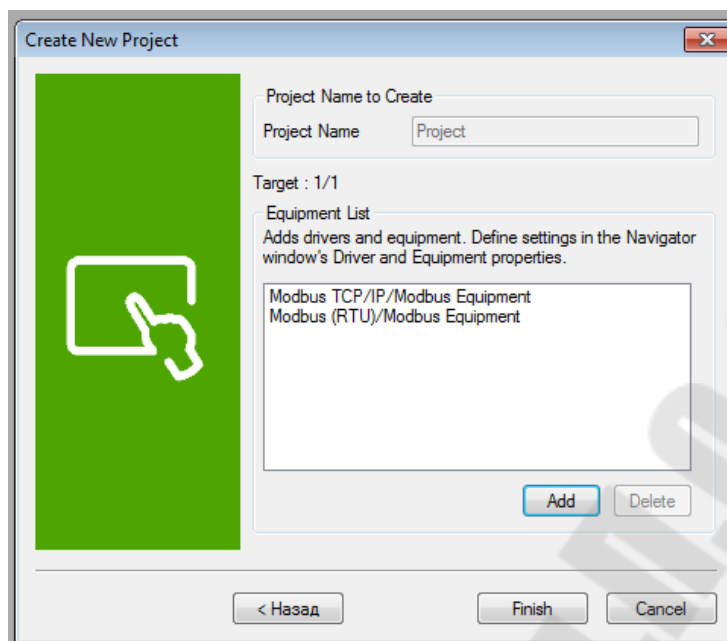


Рис. 2.7. Выбор типа оборудования *Modbus Equipment*

- Для изменения параметров сетевого адреса терминала необходимо:
- кликнуть по имени терминала в дереве проекта (*Target1*) (рис. 2.8);
  - в открывшемся меню выбрать *Network*;
  - нажать на «...» напротив *Network Configuration*;
  - установить необходимый адрес и нажать *OK*;
  - обязательно нажать кнопку *Use for Download*.

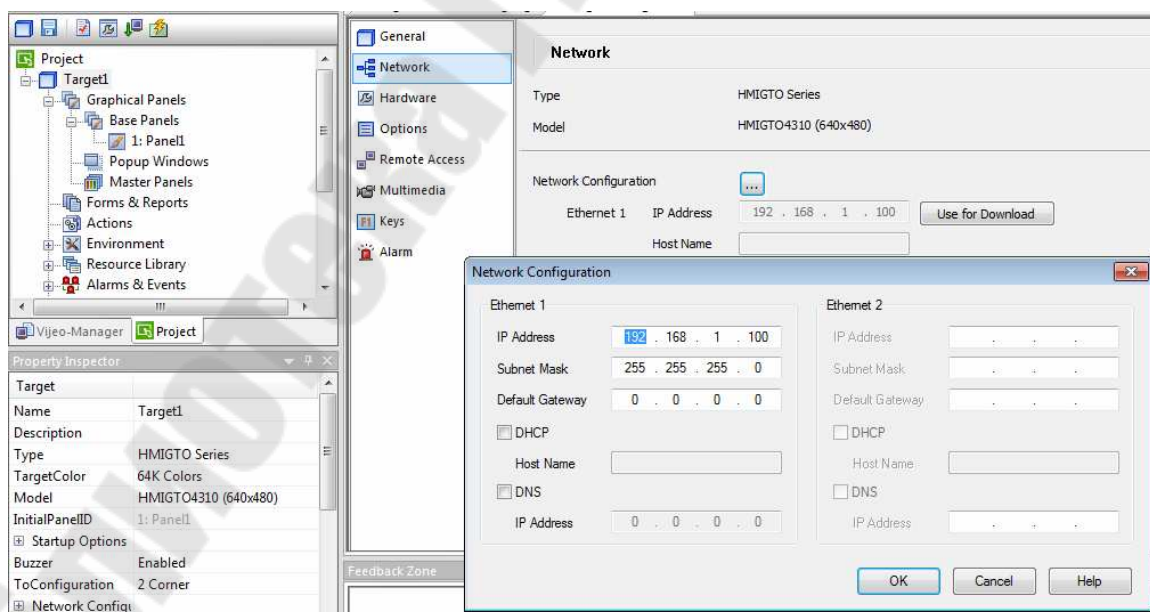


Рис. 2.8. Дерево проекта

Для программирования логического контроллера серии *Modicon M221* применяется программное обеспечение *EcoStruxure Machine Expert – Basic*.

Для создания нового проекта в среде программирования *EcoStruxure Machine Expert – Basic* необходимо:

- выбрать пункт меню *New project* (рис. 2.9);

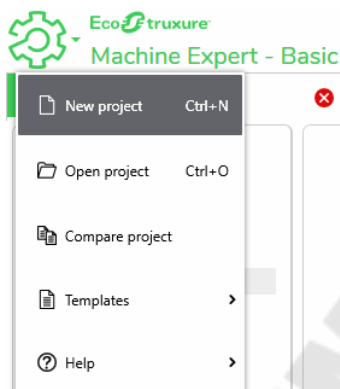


Рис. 2.9. Выбор пункта меню

– на вкладке свойств (*Properties*) (рис. 2.10), в разделе *Application Protect*, в пунктах защита чтения (*Read protection*) и защита записи (*Write protection*) можно установить режим *Active* и записать пароль (например, 0000). Возможно также выбрать опцию *Inactive*. В этом случае парольная защита не будет действовать. Но если ранее был установлен пароль, потребуется его ввести для снятия защиты;

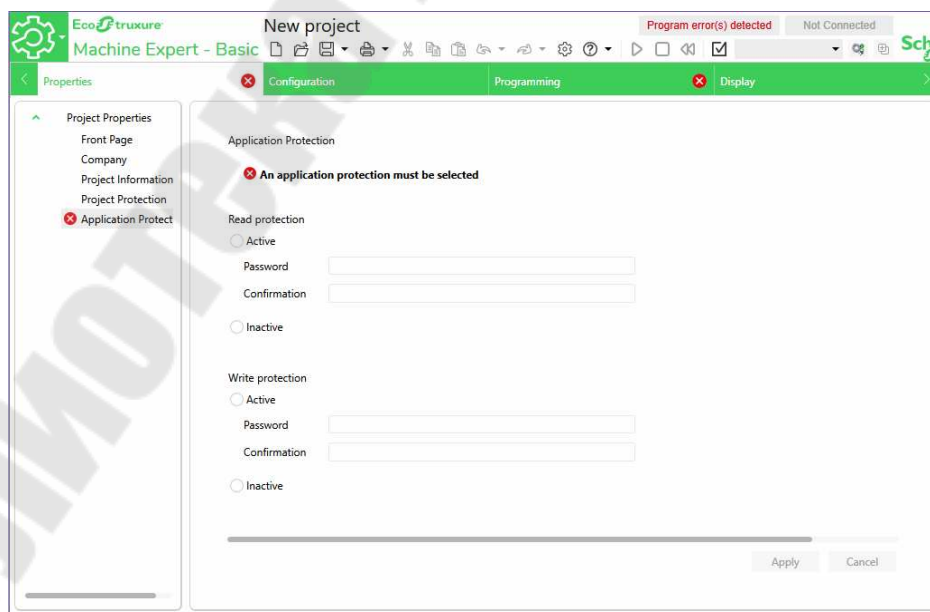


Рис. 2.10. Вкладка свойств

- нажать кнопку *Apply* для подтверждения действий;
- перейти на вкладку *Configuration* и выбрать нужную модель контроллера (рис. 2.11);
- из расположенного в правой части экрана меню *M221 Logic Controllers* выбираем модель *TM221CE24T* и перетаскиваем ее в поле проекта. На вопрос *Do you want to replace... ?* отвечаем *Yes*;

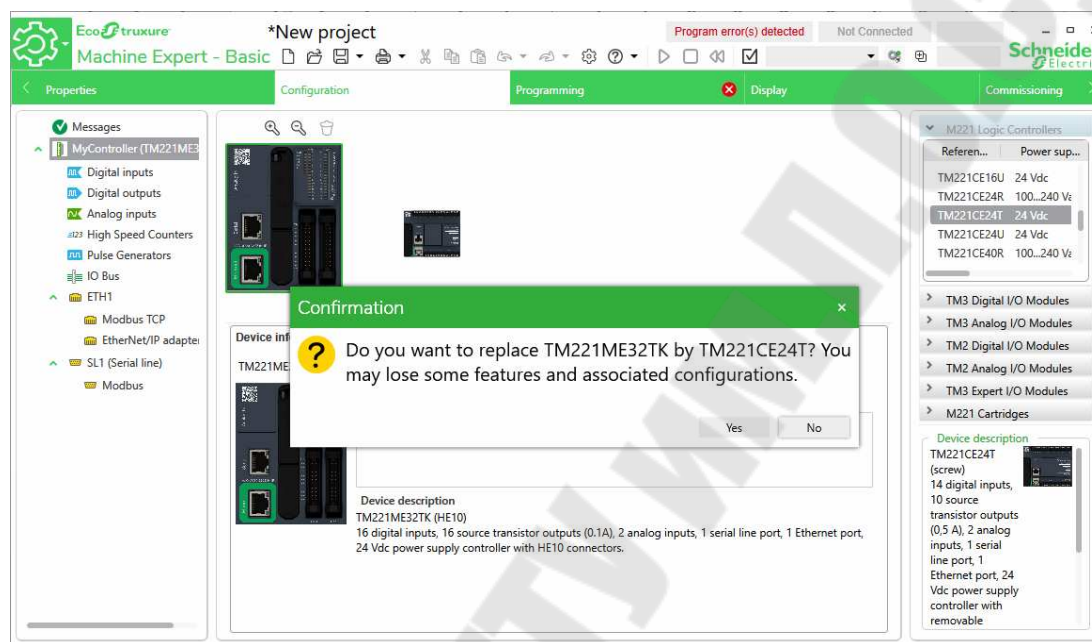


Рис. 2.11. Выбор нужной модели контроллера

– если в дальнейшем не планируется создавать проекты с другими контроллерами, можно установить данную модель по умолчанию. Для этого необходимо в меню *System Settings* выбрать раздел *Configurator* и там выбрать нужную модель контроллера (*Preferred Controller*) по умолчанию (рис. 2.12).

Для связи контроллера с другими элементами системы (преобразователями частоты, программируемыми терминалами и другим) можно воспользоваться последовательными интерфейсами контроллера, работающими по протоколам *Modbus RTU* и *Modbus TCP/IP*. Для этого необходимо настроить эти интерфейсы.

Протокол *Modbus TCP/IP* реализован по шине *Ethernet*, для его конфигурации необходимо выбрать раздел *ETH1* на вкладке *Configuration* (рис. 2.13).

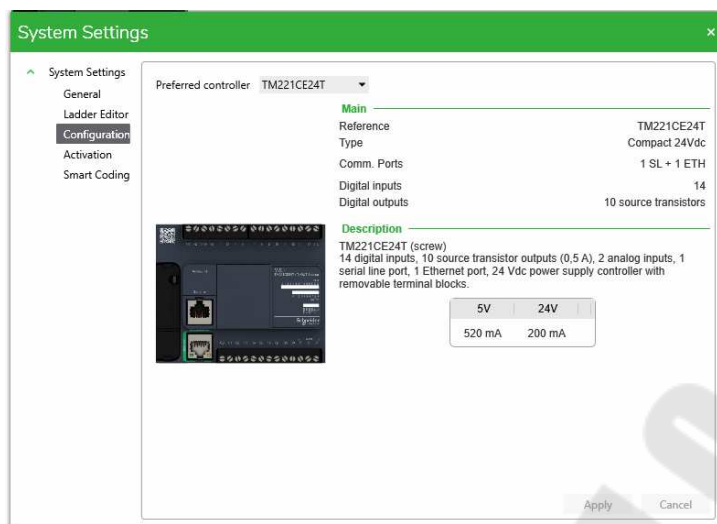


Рис. 2.12. Выбор модели контроллера по умолчанию

В пункте *Fixed IP address* нужно указать адрес контроллера в сети, например, *192.168.1.101*, и маску подсети *255.255.255.0*, а также установить все галочки в разделе *Security Parameters* и нажать *Apply*.

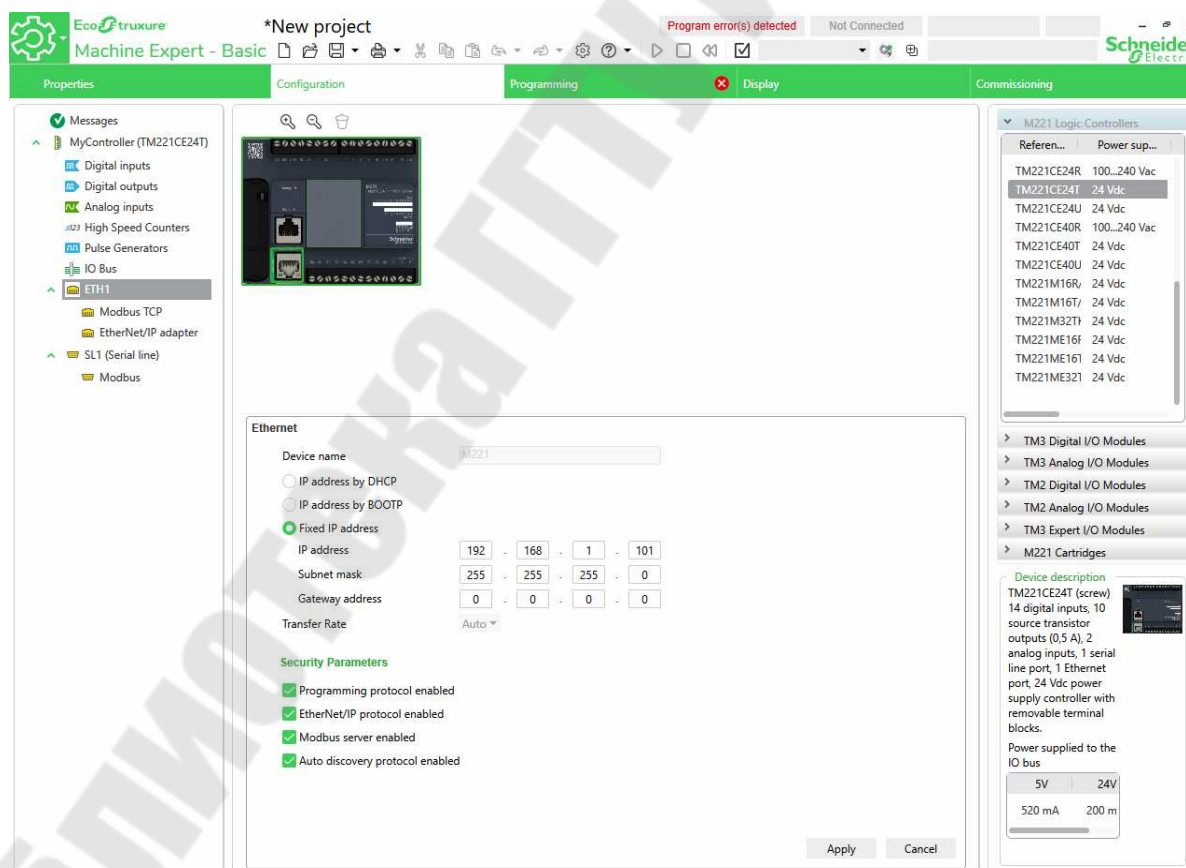


Рис. 2.13. Выбор раздела *ETH1*

### 3. Инициализация прерываний

Прерывания в терминологии *Schneider* – это события (*Events*). Изначально задача *Events* во вкладке *Programming* не активна (рис. 3.1).

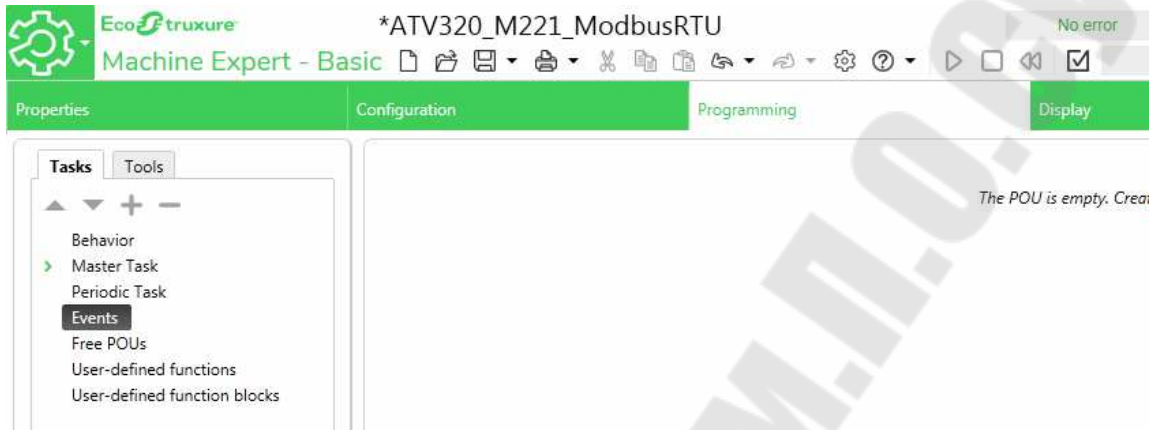


Рис. 3.1. Вкладка *Programming*

Для активации задач *Events* необходимо перейти на вкладку настройки контроллера (*Configuration*) и выбрать цифровые входы (*Digital inputs*) (рис. 3.2).

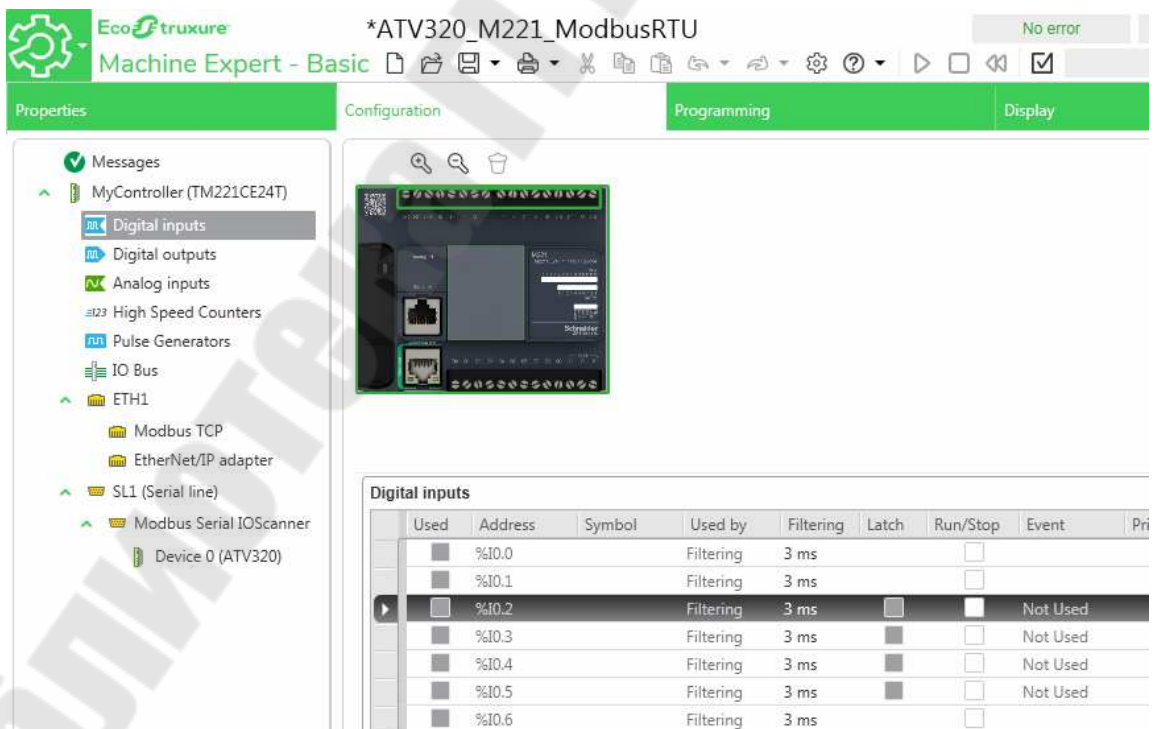


Рис. 3.2. Выбор цифровых входов

За прерывания отвечают входы  $\%I0.2$ ,  $\%I0.3$ ,  $\%I0.4$ ,  $\%I0.5$ . Для этих входов необходимо отключить входные фильтры (3 ms по умолчанию) (рис. 3.3) – в выпадающем меню необходимо выбрать *No Filter*.

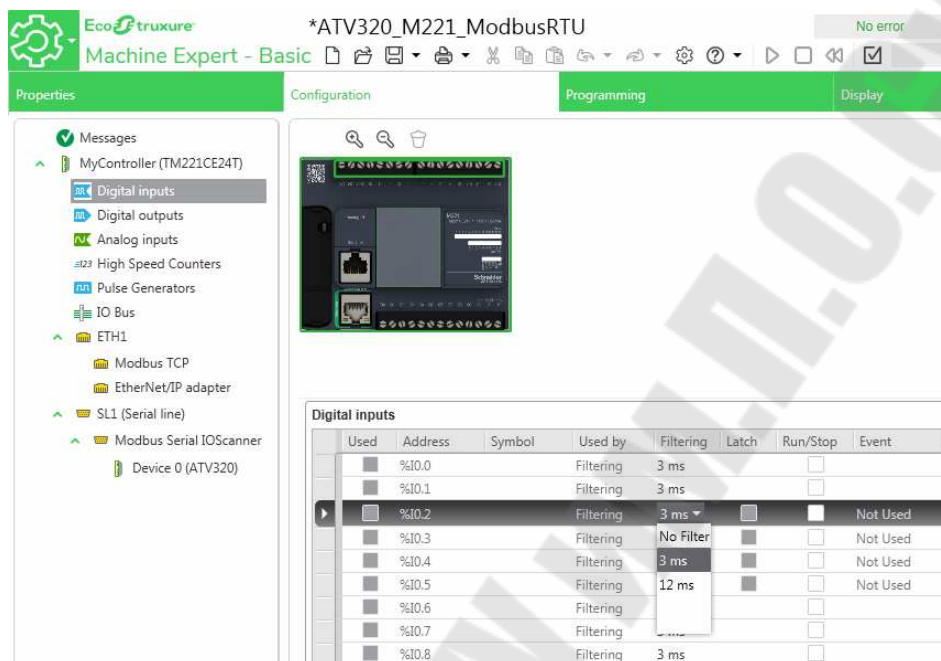


Рис. 3.3. Отключение входных фильтров

Затем в колонке *Events* заменить *Not Used* на нужный критерий срабатывания прерывания (рис. 3.4).

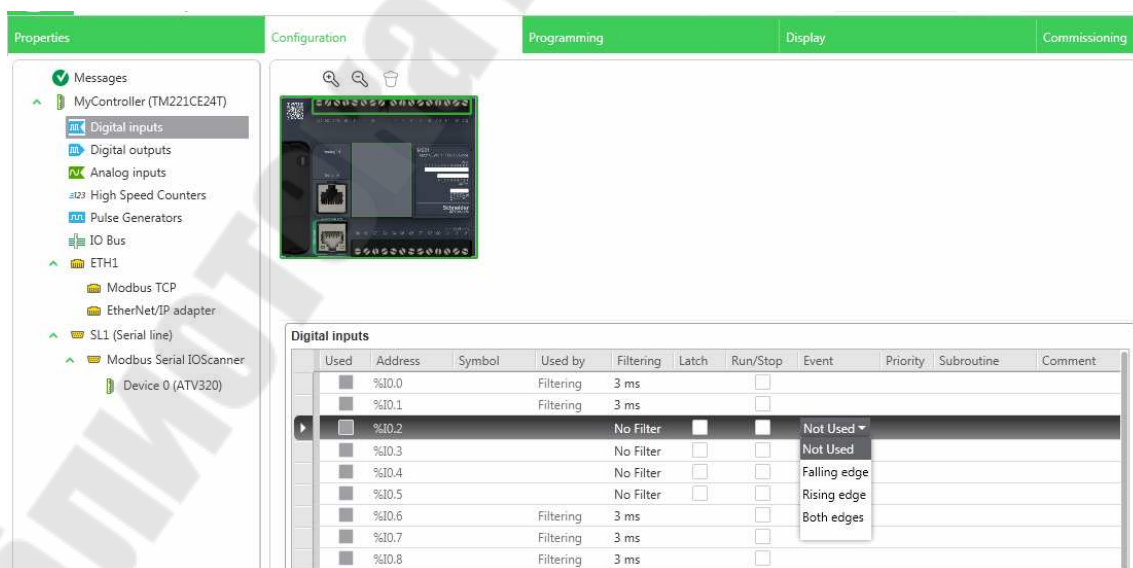


Рис. 3.4. Выбор критерия срабатывания прерывания



Например, выбрать передний фронт (*Rising edge*) (рис. 3.5). После этого в колонке приоритета (*Priority*) появится номер от 0 до 7. Чем меньше номер, тем выше приоритет прерывания. Номер также можно изменить. Кроме того, каждому прерыванию можно присвоить символическое имя, например, *INT\_2*. Не забываем нажать *Apply*!

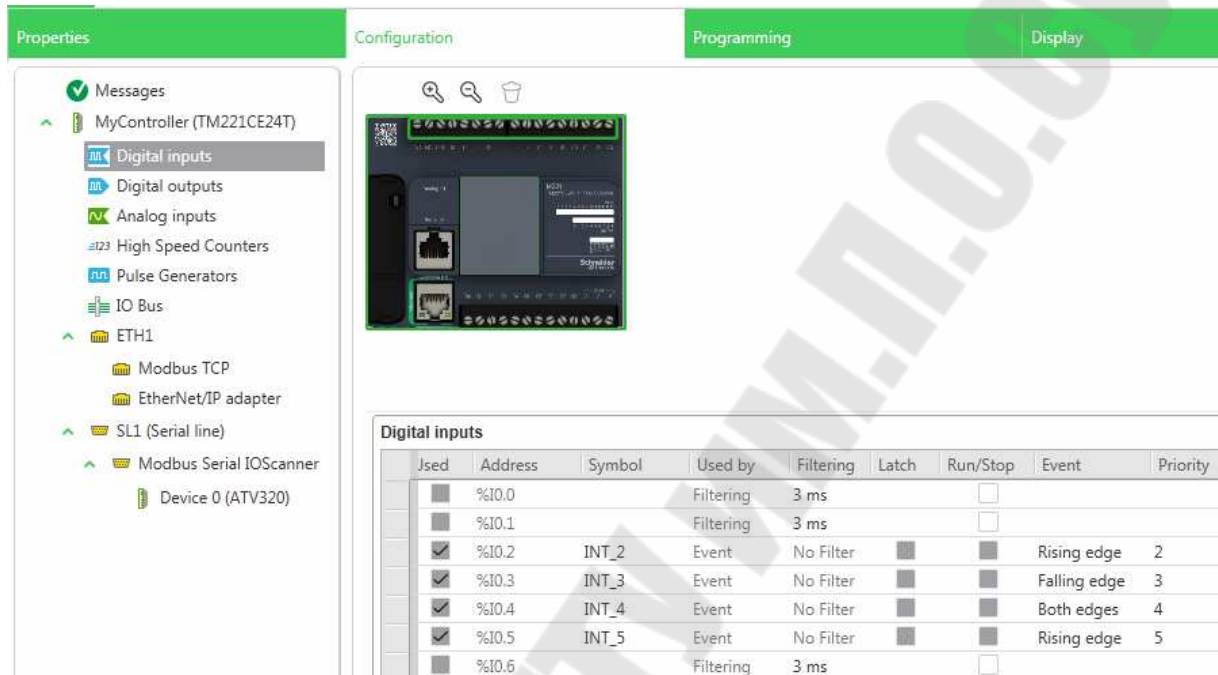


Рис. 3.5. Выбор фронта

После этого во вкладке *Programming* в задаче *Events* появятся четыре обработчика прерываний (рис. 3.6).

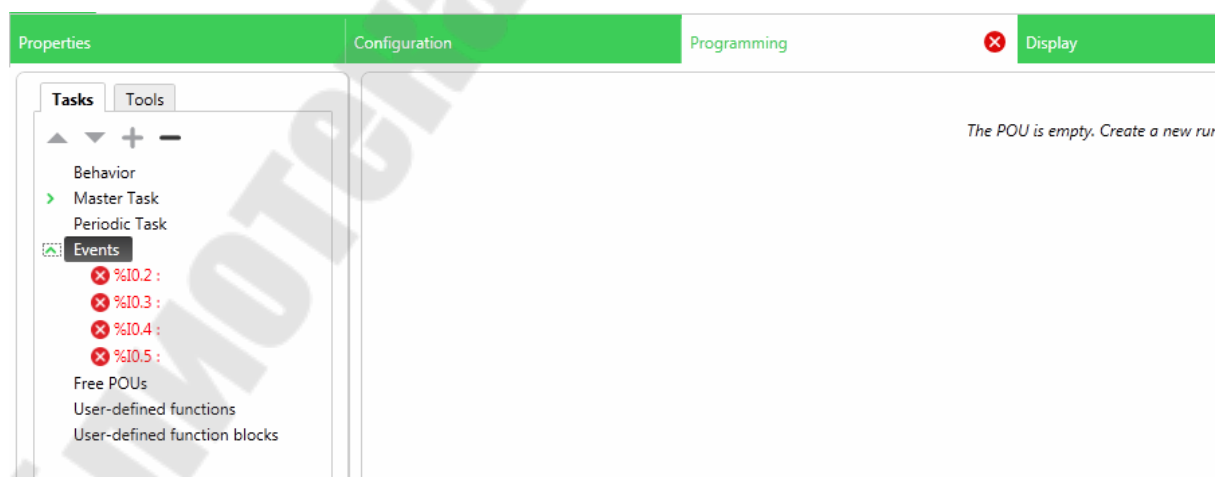


Рис. 3.6. Обработки прерываний

При нажатии правой кнопкой мыши на любое из событий %I0.2, %I0.3, %I0.4, %I0.5 появится меню, в котором можно выбрать *Add Free POU* (рис. 3.7).

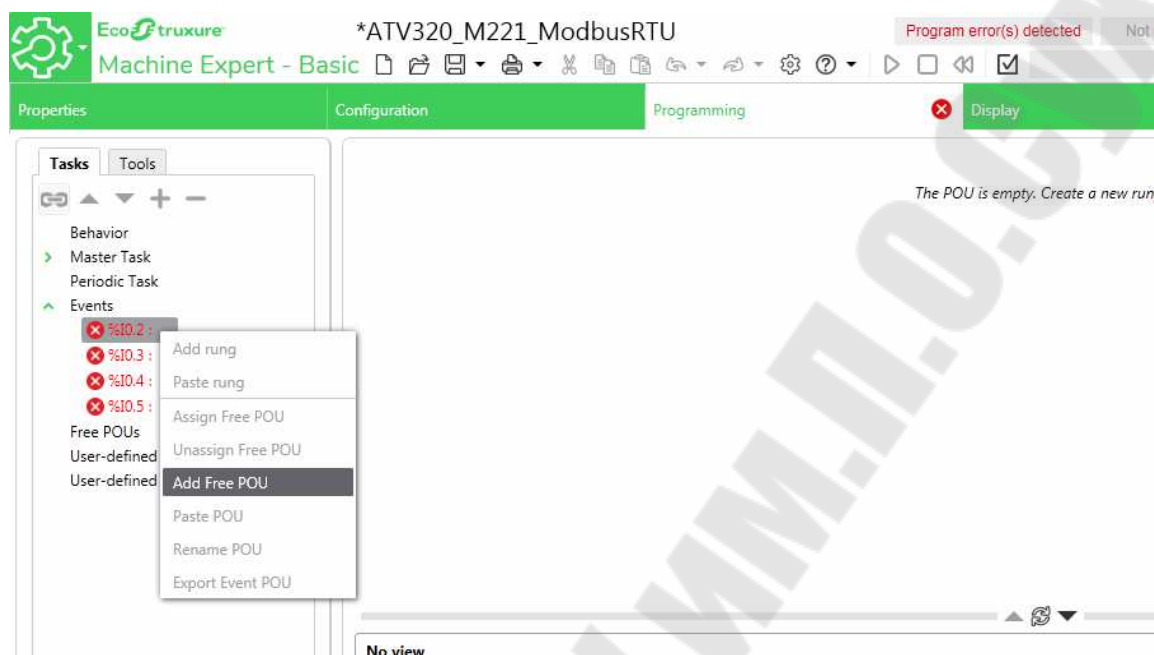


Рис. 3.7. Выбор *Add Free POU*

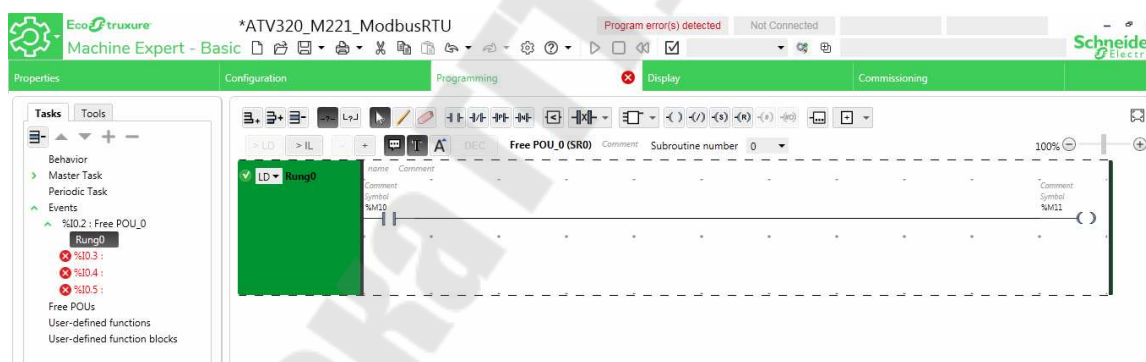


Рис. 3.8. Написание программы обработчика для прерывания

После этого можно написать программу обработчика для каждого прерывания (рис. 3.8).

#### 4. Создание всплывающих окон

Создадим всплывающее окно, которое будет появляться при возникновении прерывания по входу *I0.2* ПЛК.

Для этого сначала в дереве проекта в окне *Navigator* выберем раздел всплывающих окон (*Popup Windows*) и создадим новое

всплывающее окно (*New Popup Window*) (рис. 4.1). Назовем новое окно *Int\_02*.

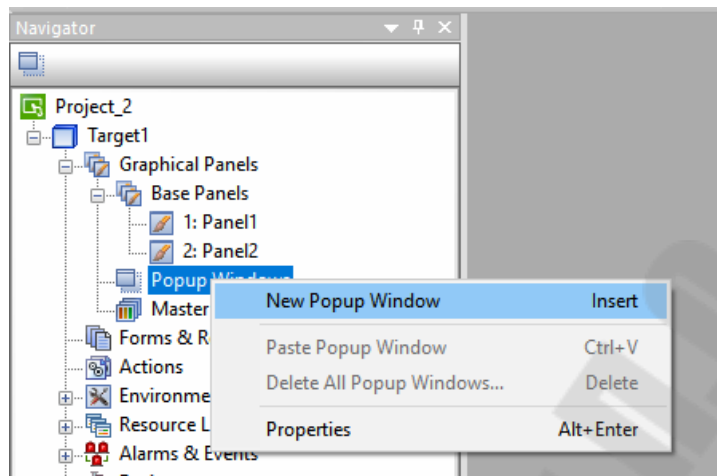


Рис. 4.1. Создание новой панели

В новом всплывающем окне по нажатию правой кнопкой мыши создадим новую панель (*New Panel*) (рис. 4.2).

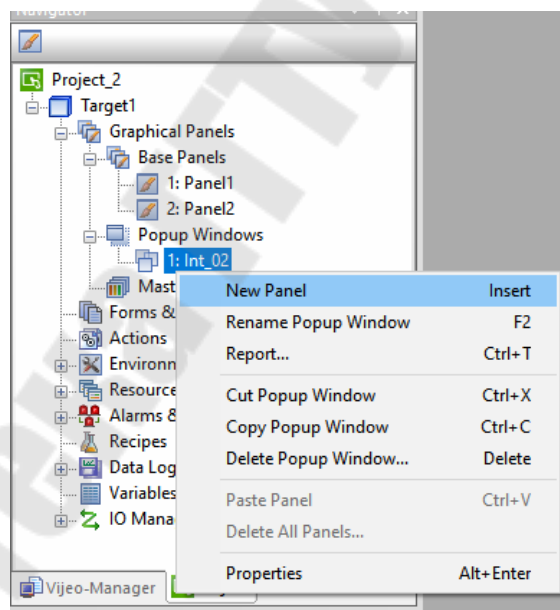


Рис. 4.2. Создание новой панели

Сделаем заливку фона вновь созданной панели красным цветом, и разместим на панели надпись «ВНИМАНИЕ! СРАБОТАЛ ИНДУКТИВНЫЙ ДАТЧИК» (рис. 4.3).

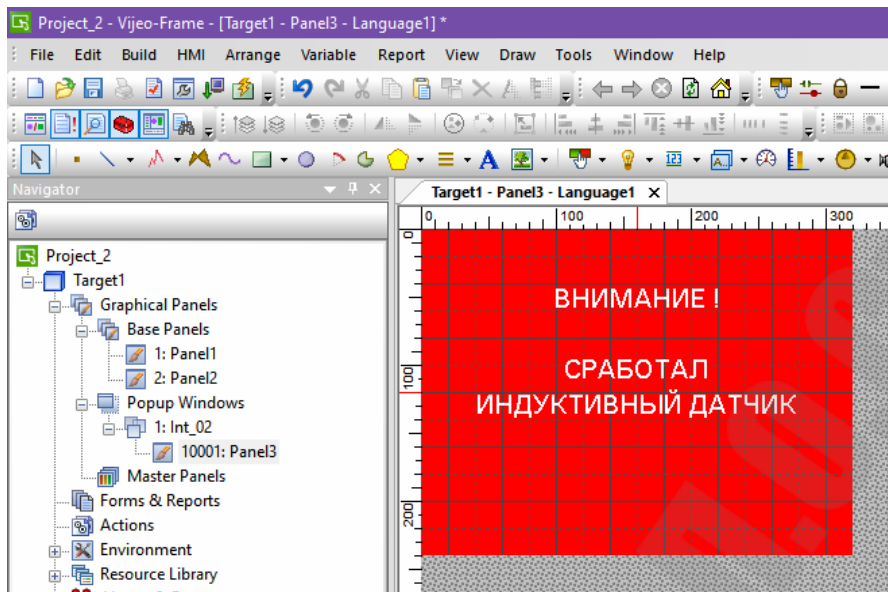


Рис. 4.3. Размещение надписи

Добавляем кнопку, закрывающую окно. Для этого в панели инструментов *Toolchest* выберем библиотеку *Schneider Electric Image Library*, откроем папку *Status Icons* и перетащим элемент *Status OK* на панель (рис. 4.4).

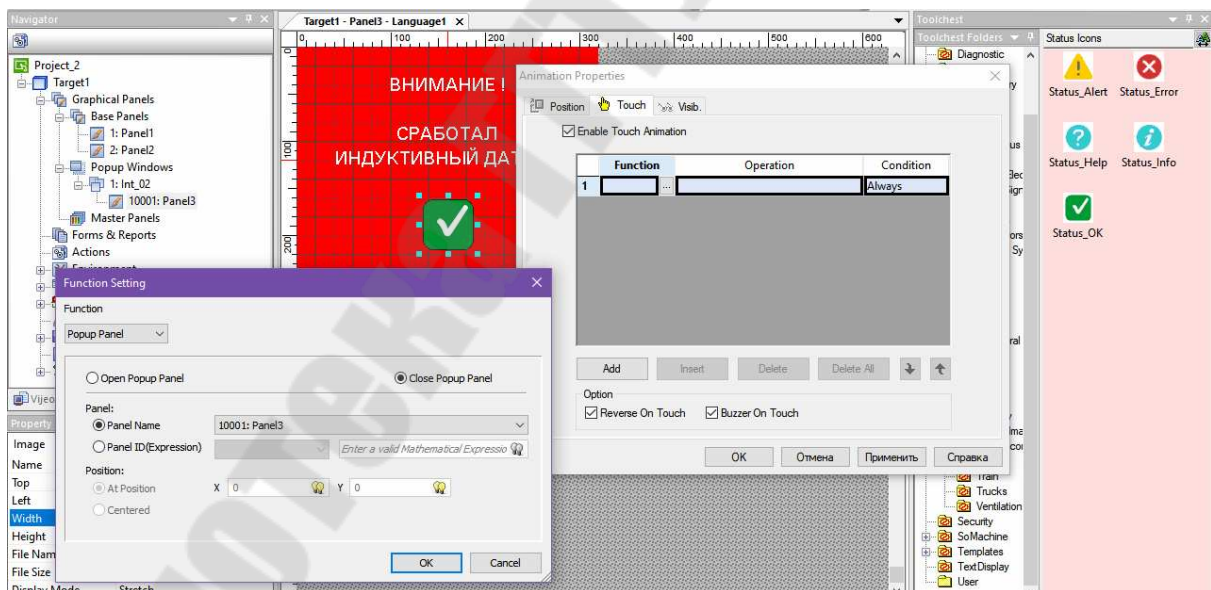


Рис. 4.4. Добавление кнопки, закрывающей окно

Добавим элементу *Status OK* первую функцию, закрывающую панель. Для этого двойным нажатием левой кнопки мыши на элемент *Status OK* вызовем окно *Animation Properties*, где на вкладке *Touch* нажмем кнопку *Add*. В окне свойств функции (*Function Settings*)

укажем тип функции (*Function*) – всплывающее окно (*Popup Panel*), закрыть всплывающее окно (*Close Popup Panel*).

Добавим вторую функцию кнопке – сброс бита прерывания  $\%M0$  (рис. 4.5). Для этого придется создать новую переменную типа *bool* с именем *INT\_02* и адресом  $\%M0$  на устройстве *ModbusEquipment02*.

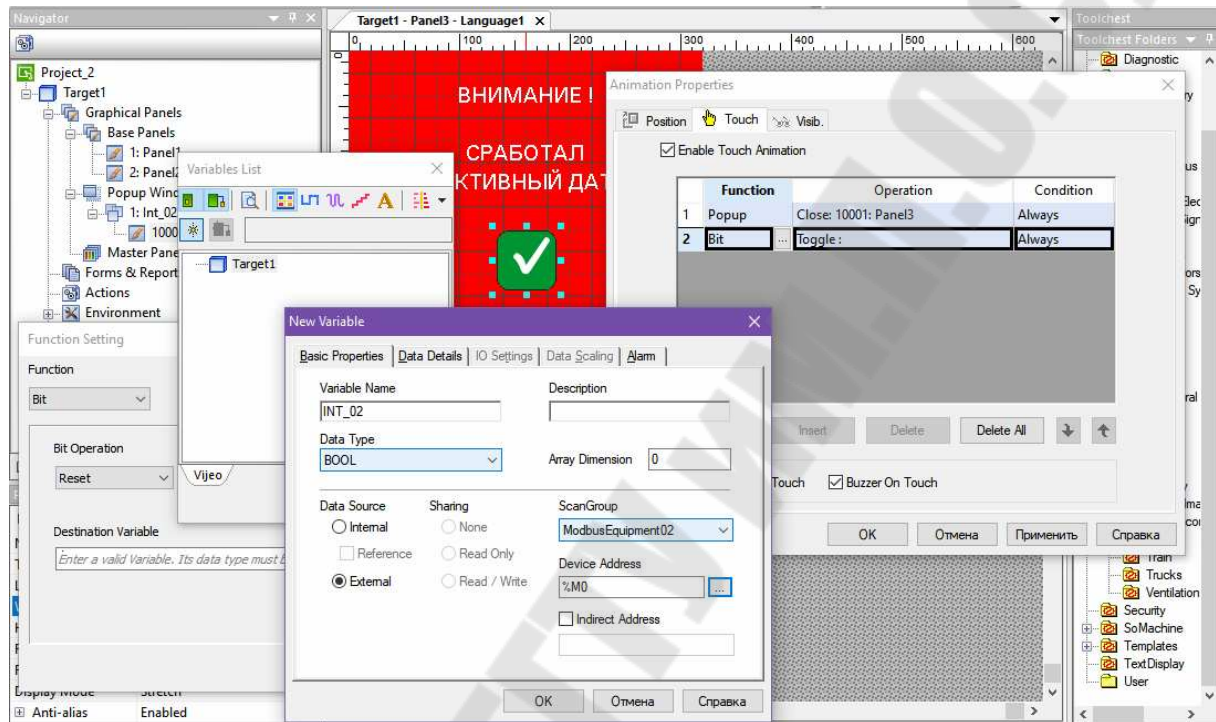


Рис. 4.5. Добавление сброса бита прерывания

Добавим экшен, запускающий всплывающий экран по условию установки бита  $\%M0$  (рис. 4.6). Перейдем в дерево проекта, в окно *Navigator* в раздел *Actions* и создадим новое действие (*Create a new Action*). Выберем тип срабатывания (*Trigger Type*) – по условию (*Conditional*). В пункте *Trigger Variable* выберем переменную  $\%M0$ . Нажмем *Next*.

В появившемся окне укажем тип операции (*Operation*) – всплывающее окно (*Popup*), а в качестве имени панели (*Panel Name*) выберем *Panel3* (рис. 4.7).

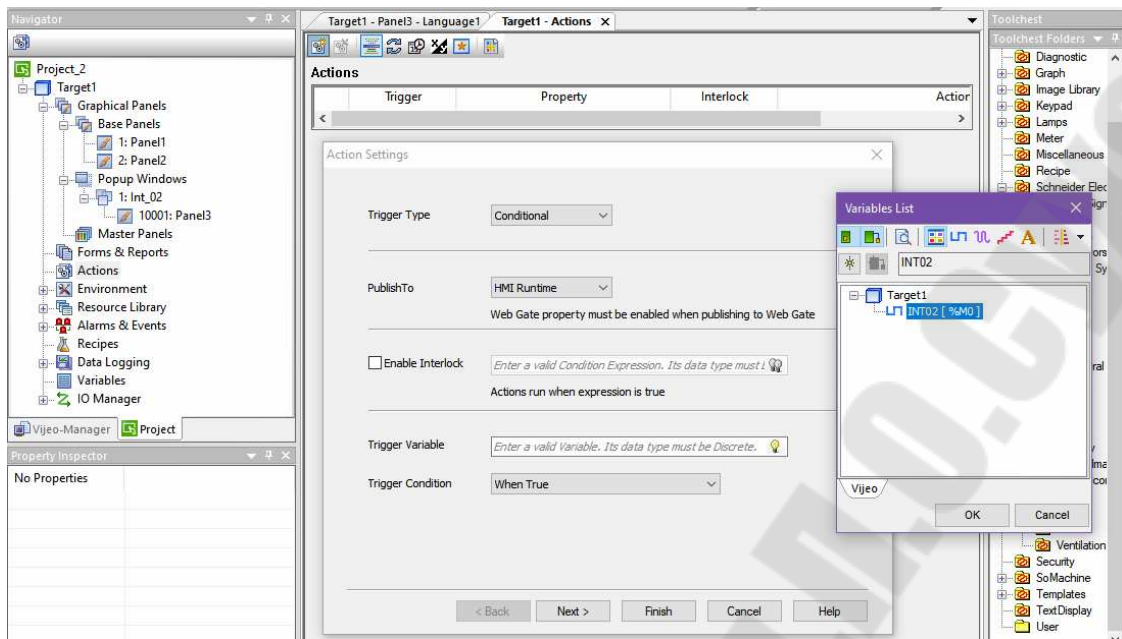


Рис. 4.6. Добавление экшена

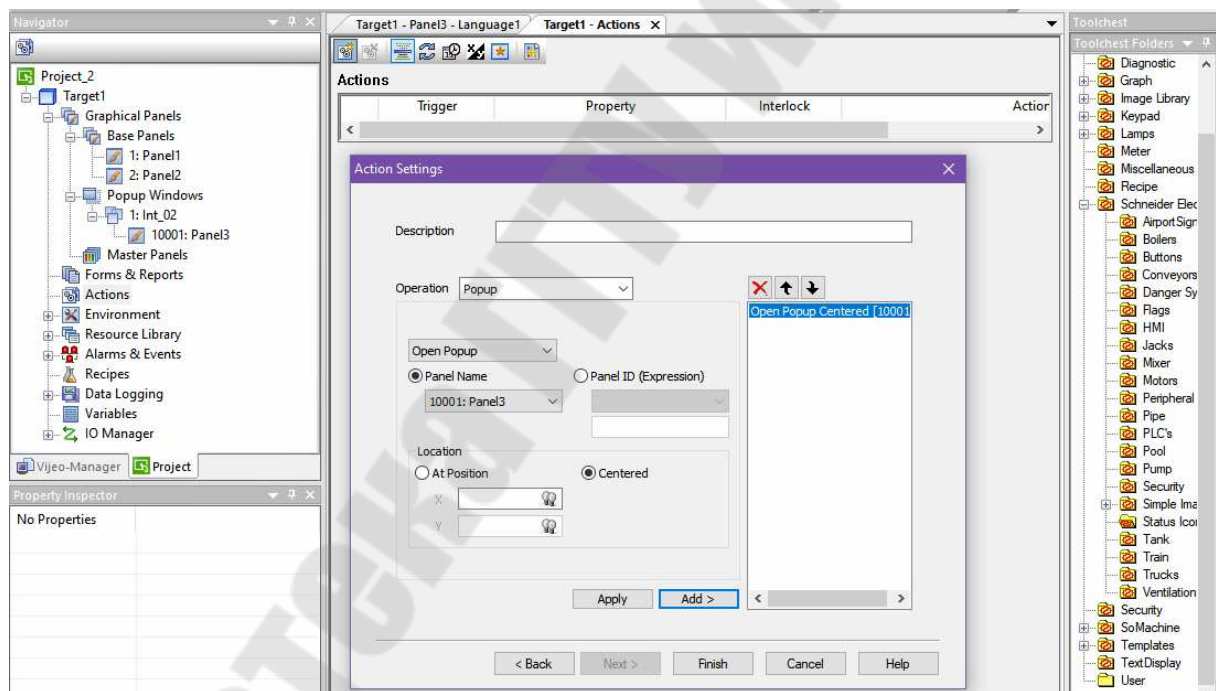


Рис. 4.7. Выбор имени панели

В *Machine Expert – Basic* активируем прерывания по входу  $\%I0.2$  с критерием срабатывания по переднему фронту и приоритетом 7 (рис. 4.8).

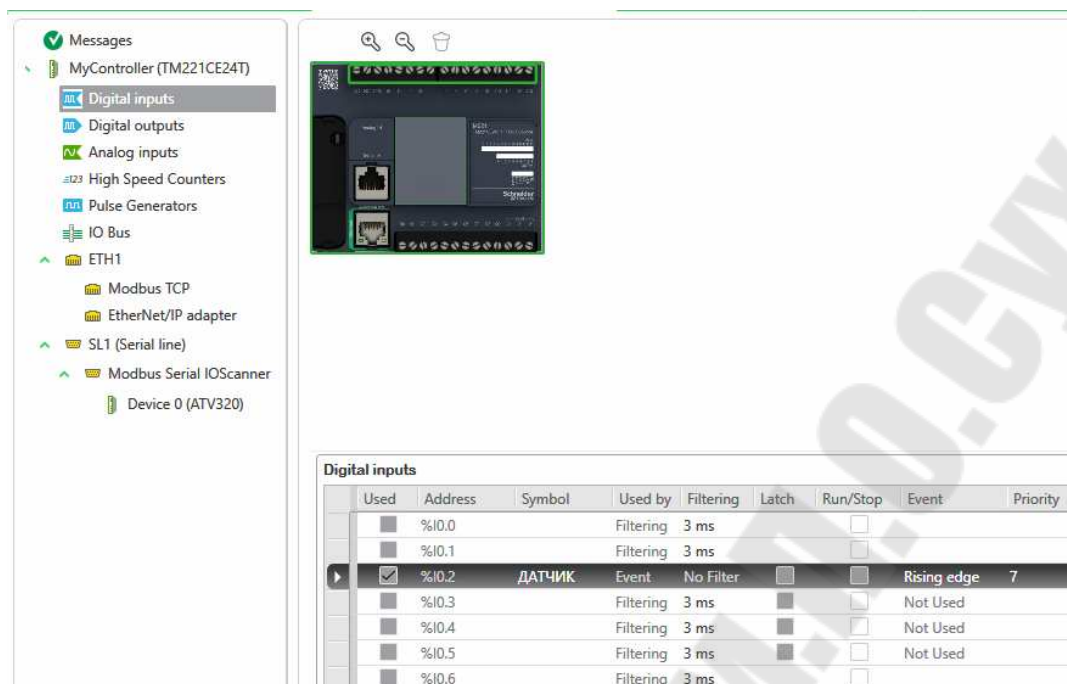


Рис. 4.8. Активизация прерывания по входу

Создадим обработчик прерывания, устанавливающий в 1 бит  $%M0$  (рис. 4.9).

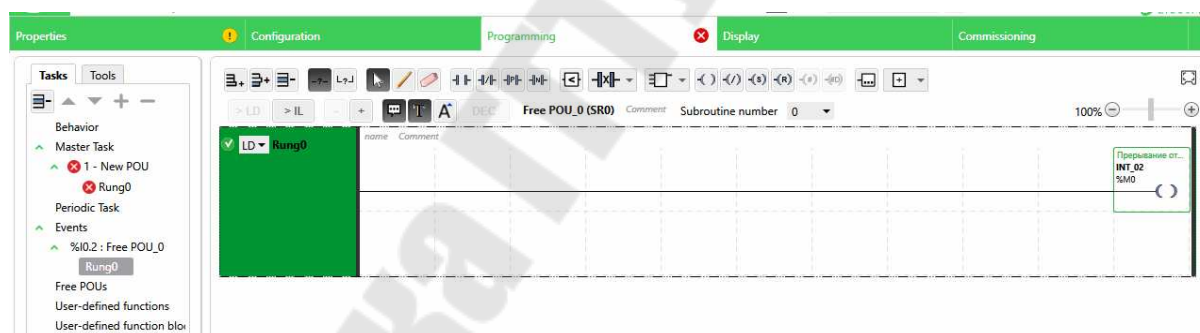


Рис. 4.9. Создание обработчика прерывания

В основной программе добавим контакт  $%I0.2$ , устанавливающий бит  $%M0$  (рис. 4.10). Это необходимо для активации прерывания при срабатывании контакта  $%I0.2$ .

Скомпилируем проект и запустим его симуляцию.

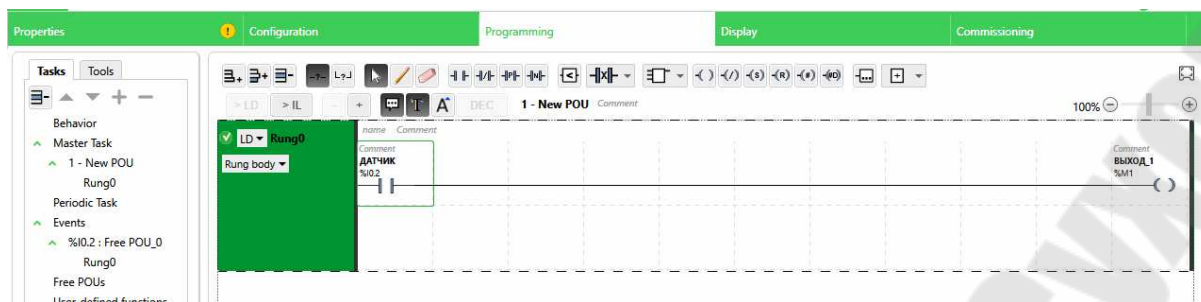


Рис. 4.10. Добавление контакта

Для совместной работы симуляторов *Machine Expert – Basic* и *Vijeo Designer* установим в последнем адрес устройства *ModbusEquipment* 127.0.0.1 (рис. 4.11).

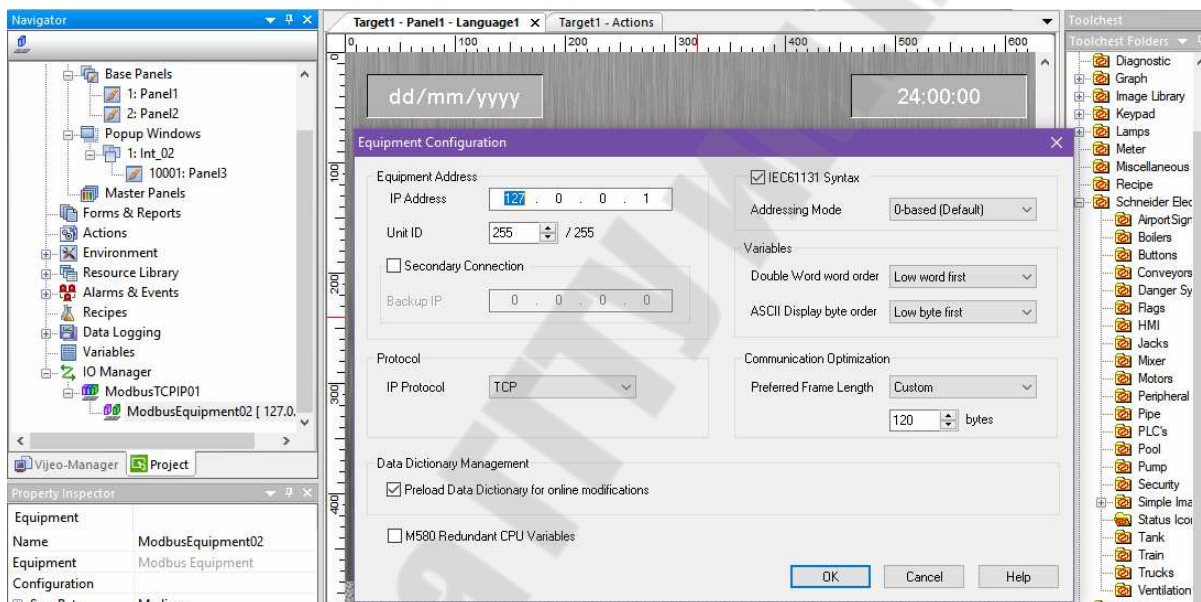


Рис. 4.11. Установка адреса устройства

На рис. 4.12 приведена симуляция проекта до срабатывания датчика – контакт %0.2 разомкнут, всплывающего окна нет.

На рис. 4.13 приведена симуляция проекта после срабатывания датчика – контакт %0.2 замкнут, появилось всплывающее окно.



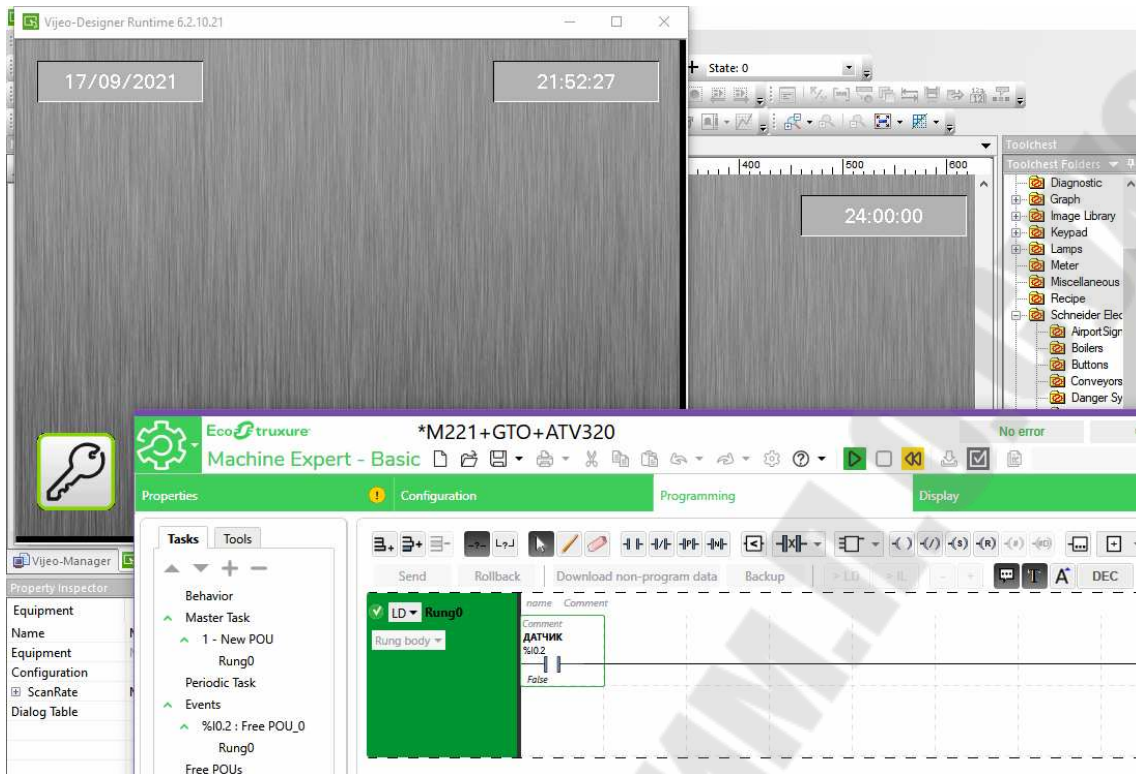


Рис. 4.12. Симуляция проекта до срабатывания датчика

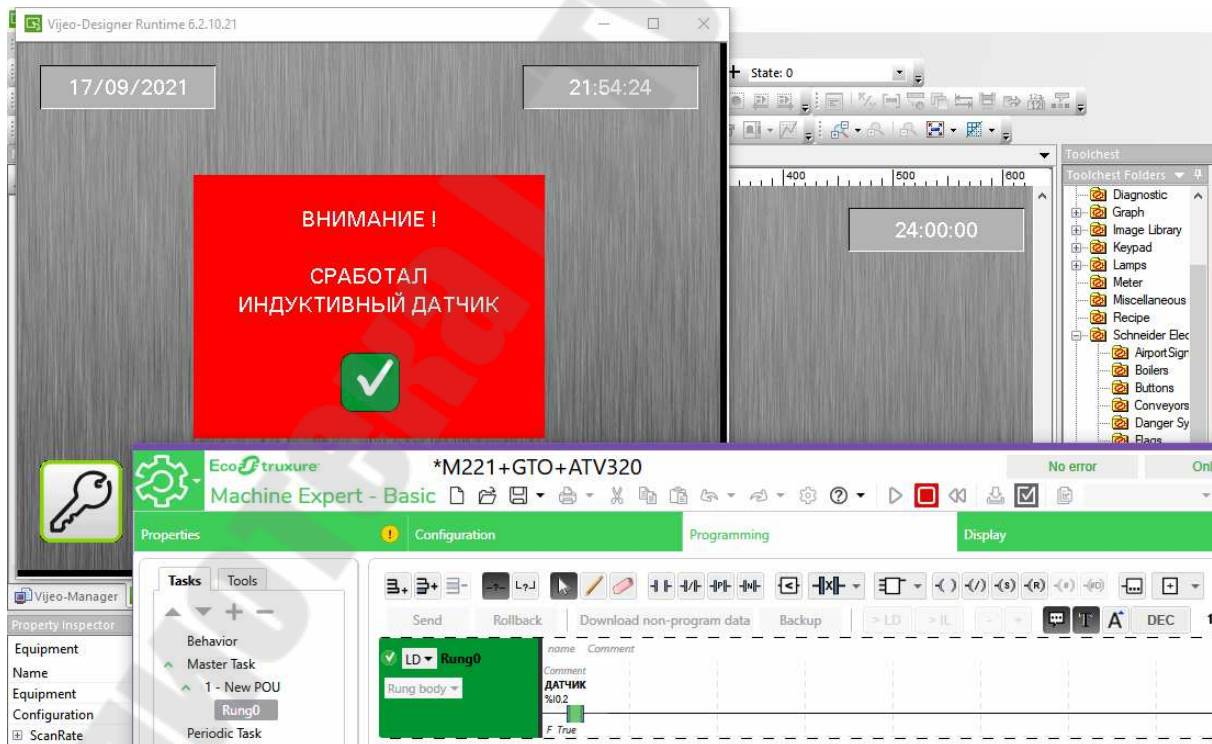


Рис. 4.13. Симуляция проекта после срабатывания датчика

## 5. Панель ввода значений

Создадим панель ввода значений параметров.

Вначале создадим переменные типа *INT* (16 бит со знаком) с именами «*Параметр\_1*», «*Параметр\_2*» и т. д., а также переменные с именами «*Номер\_параметра*» и «*Заданное\_значение*» (рис. 5.1).

Для переменной «*Заданное\_значение*» в открывшемся окне свойств переменной (*Variable Properties*), на вкладке основных свойств (*Basic Properties*) установим галочку в чекбоксе: *Reference*, в качестве отображаемой строки (*Reference String*) укажем «*Параметр\_%d*». Здесь *%d* будет выводить десятичное значение параметра, а в качестве параметра (*Parameter 1*) выберем переменную «*Номер\_параметра*».

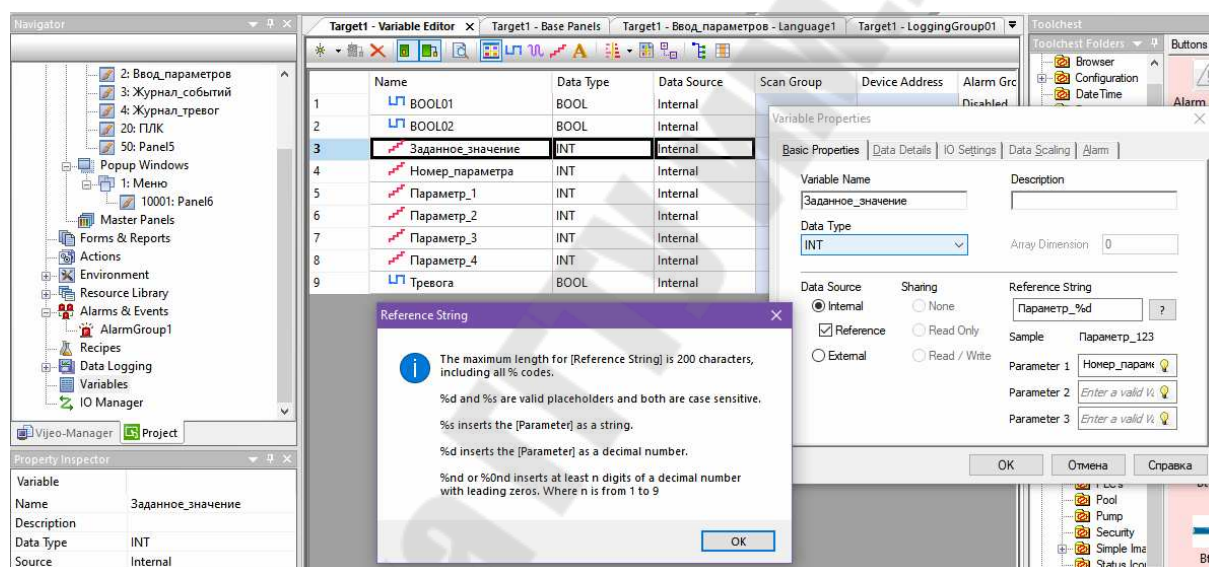


Рис. 5.1. Создание переменного типа *INT*

Для переменной «*Номер\_параметра*» в окне свойств переменной (*Variable Properties*) выберем вкладку *Data Details* и укажем начальное значение переменной (*Initial Value*) 1 и диапазон допустимых значений (*Input Range*) от min 1 до max 8 (рис. 5.2).

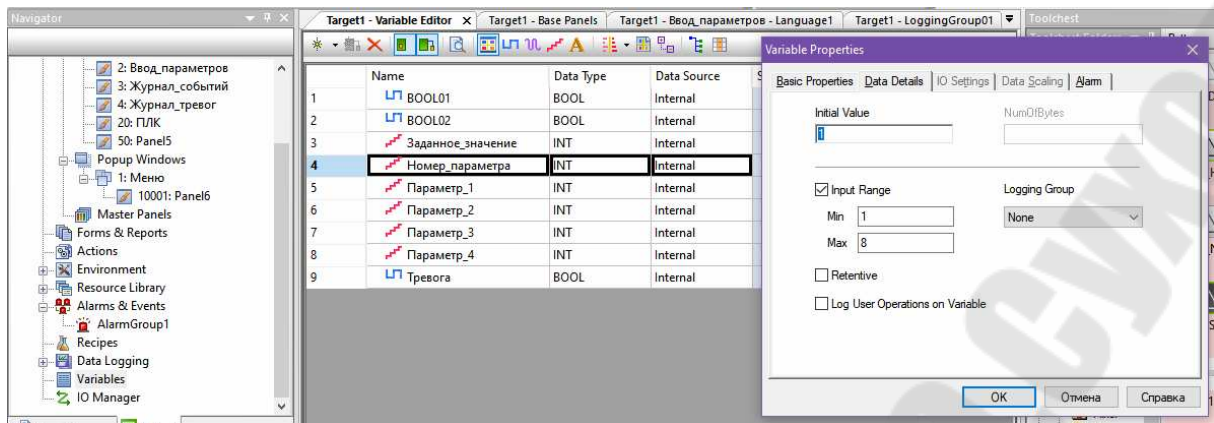


Рис. 5.2. Окно свойств переменных

С помощью встроенного графического редактора создадим панель ввода параметров (рис. 5.3).

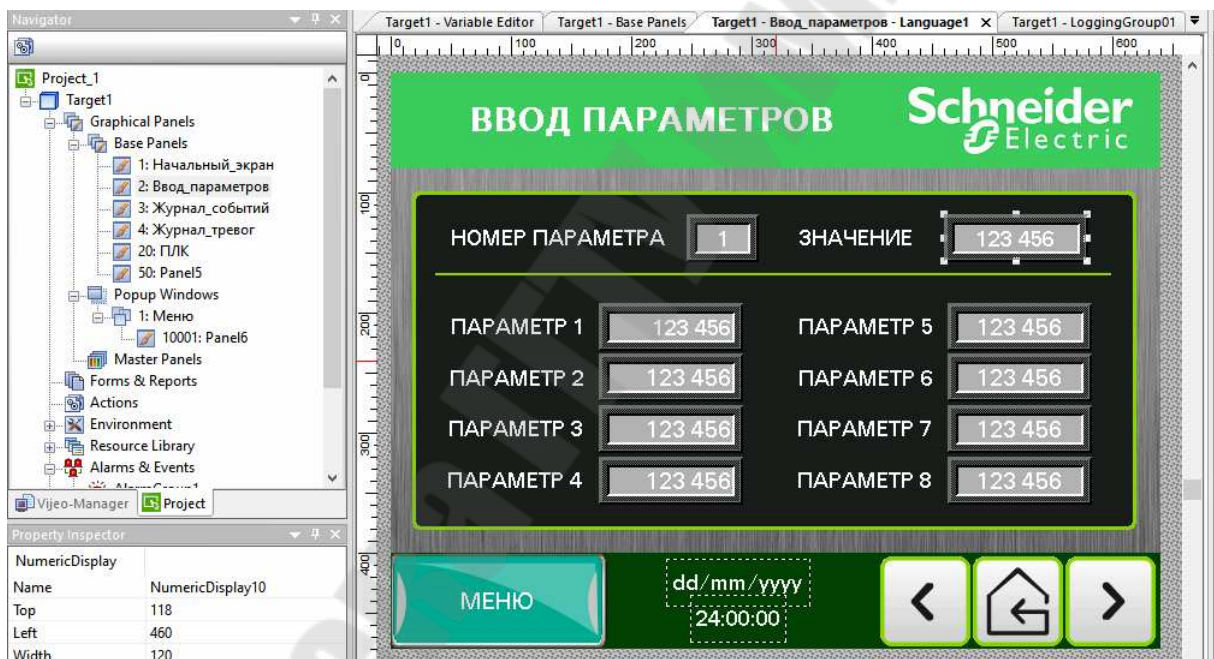


Рис. 5.3. Создание панели ввода параметров

Выведем на экран ПТ сообщение, содержащее значение переменной. Для этого создадим текстовый ресурс, содержащий сообщение для журнала событий (рис. 5.4). В дереве проекта, в разделе *Resource Library* выберем пункт *Text* и на панели инструментов нажмем кнопку создания нового ресурса (*New Resource*). В появившемся окне нажмем *OK* для подтверждения.

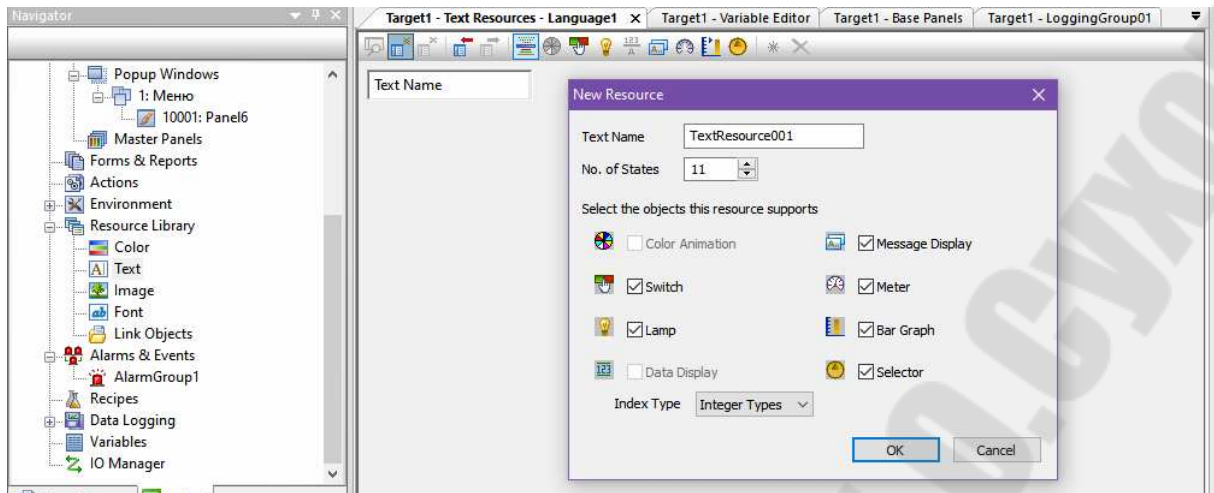


Рис. 5.4. Создание текстового ресурса

Введем текст сообщения: «Установлено значение параметра 1:» (рис. 5.5). Не выходя из режима редактирования текста, нажатием правой кнопкой мыши на строку с текстом вызовем всплывающее меню, где выберем пункт *Insert* → *Variable...* и выберем отображаемую переменную «*Параметр\_1*».

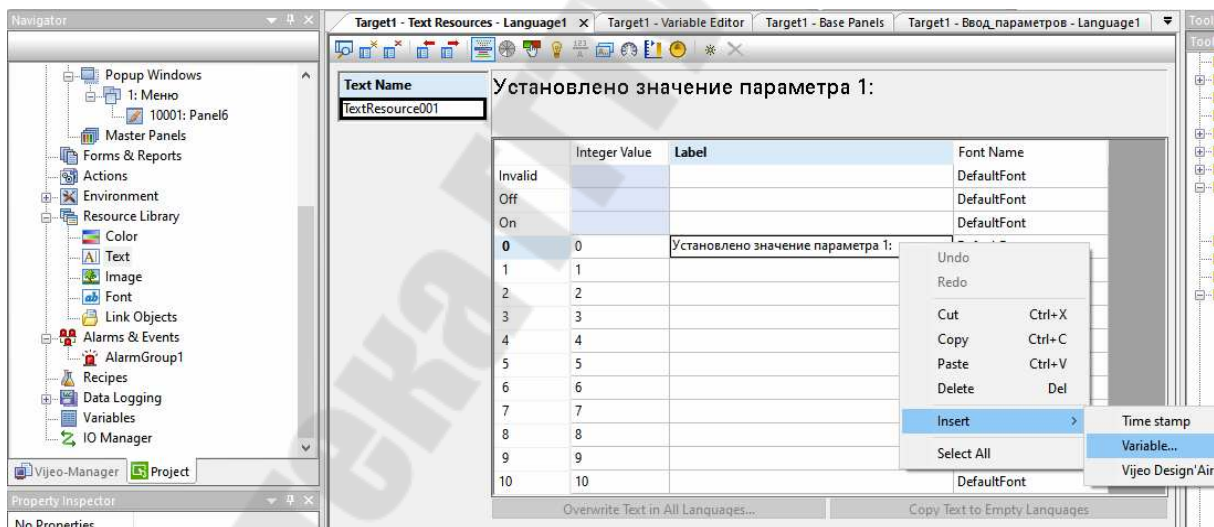


Рис. 5.5. Ввод текста сообщения

Создадим группу событий *EventGroup1* в разделе тревог и событий (*Alarms & Events*) дерева проекта (рис. 5.6). Для этого выберем пункт всплывающего меню *New Event Group* → *Event Group*.

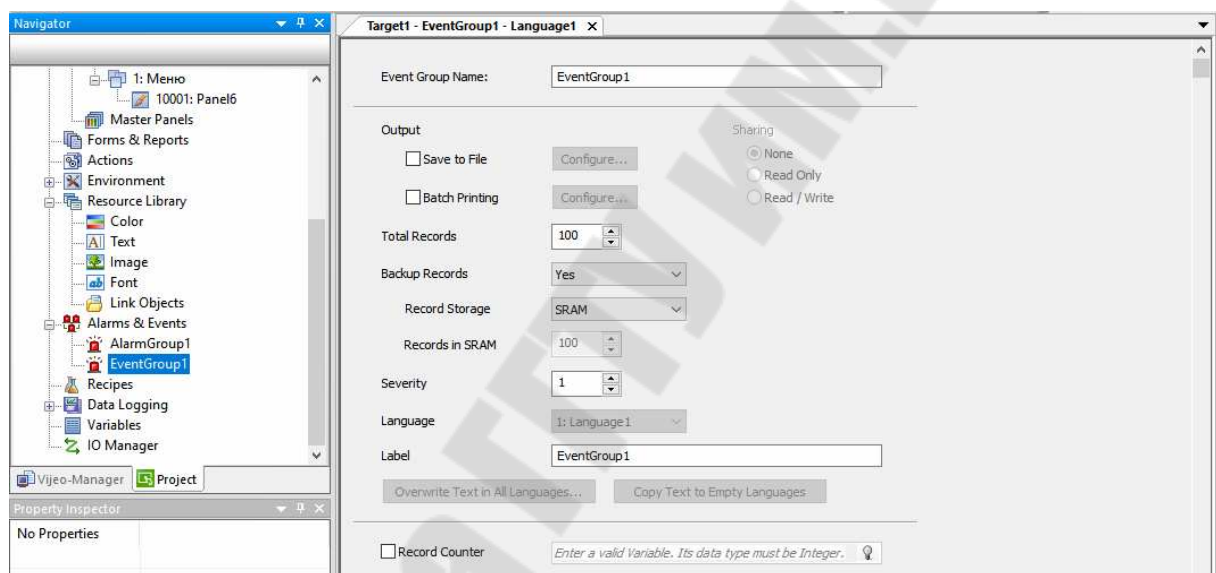
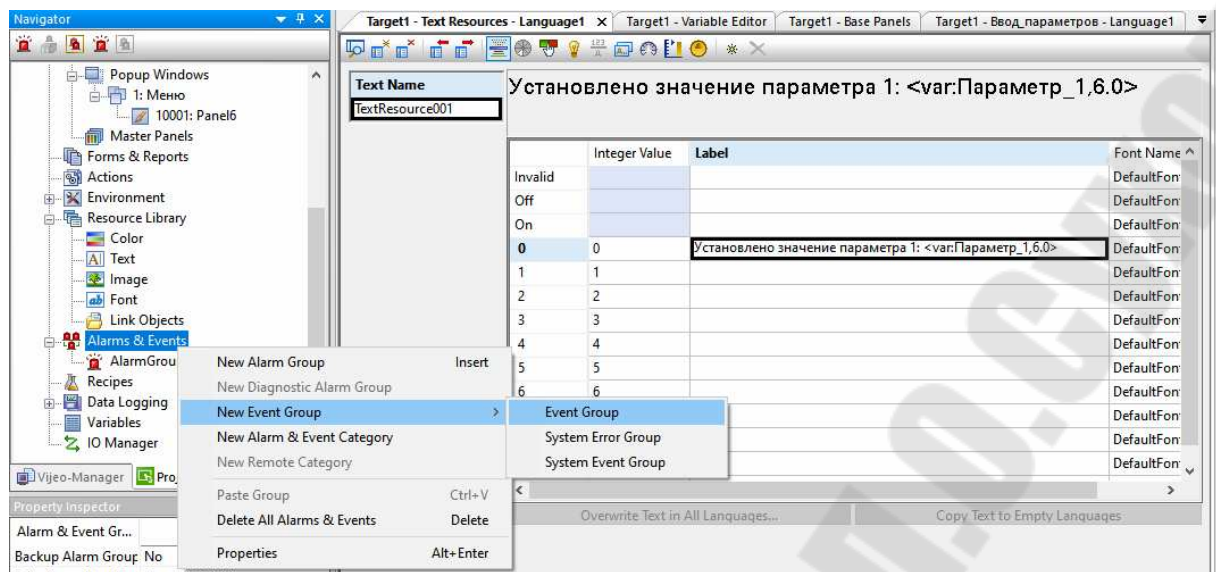


Рис. 5.6. Создание группы событий

Создадим событие, по которому будет отображаться сообщение в журнале событий. Этим событием будет изменение переменной. Для этого в окне *Navigator* в дереве проекта выберем раздел *Actions* (рис. 5.7). На панели задач открывшегося окна *Actions* нажмем значок «Создать новое действие» (*Create a new Action*). В появившемся окне настроек (*Action Settings*) укажем тип срабатывания (*Trigger Type*) по условию (*Conditional*). В качестве переменной для срабатывания (*Trigger Variable*) выберем *Параметр\_1*. Нажмем кнопку *Next*.

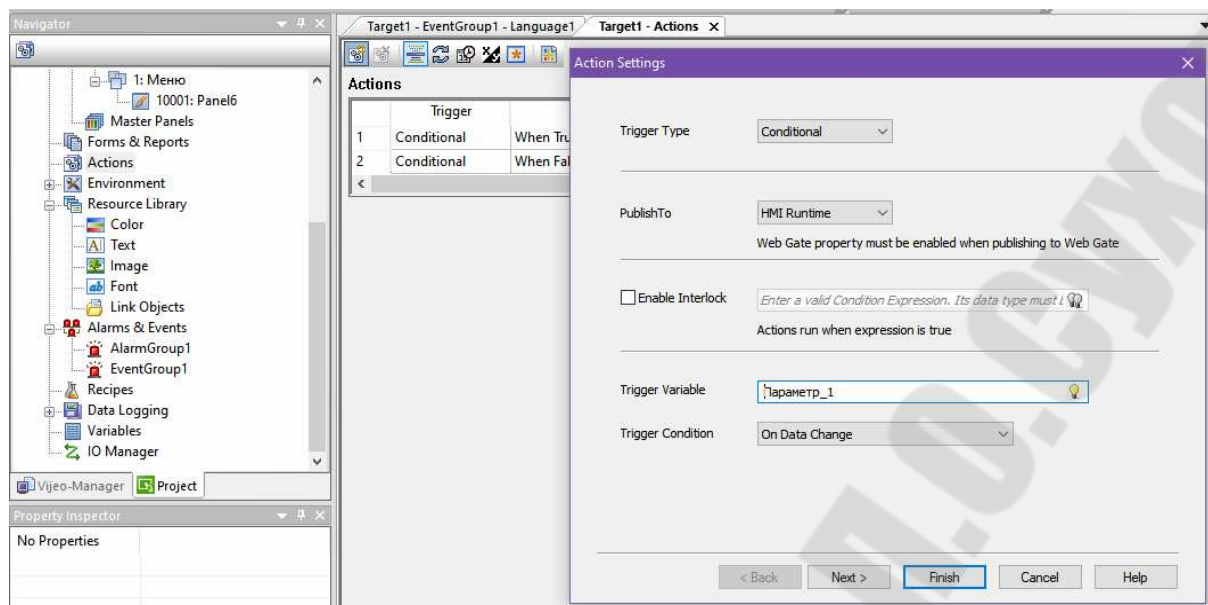


Рис. 5.7. Выбор раздела в дереве проекта

В появившемся окне в качестве операции (*Operation*) выберем «Сообщение о событии» (*Event Message*) (рис. 5.8). В качестве выводимого сообщения (*Text Resource*) выберем *TextResource001*, обязательно нажмем кнопку «Добавить» (*Add*) и «Завершить» (*Finish*).

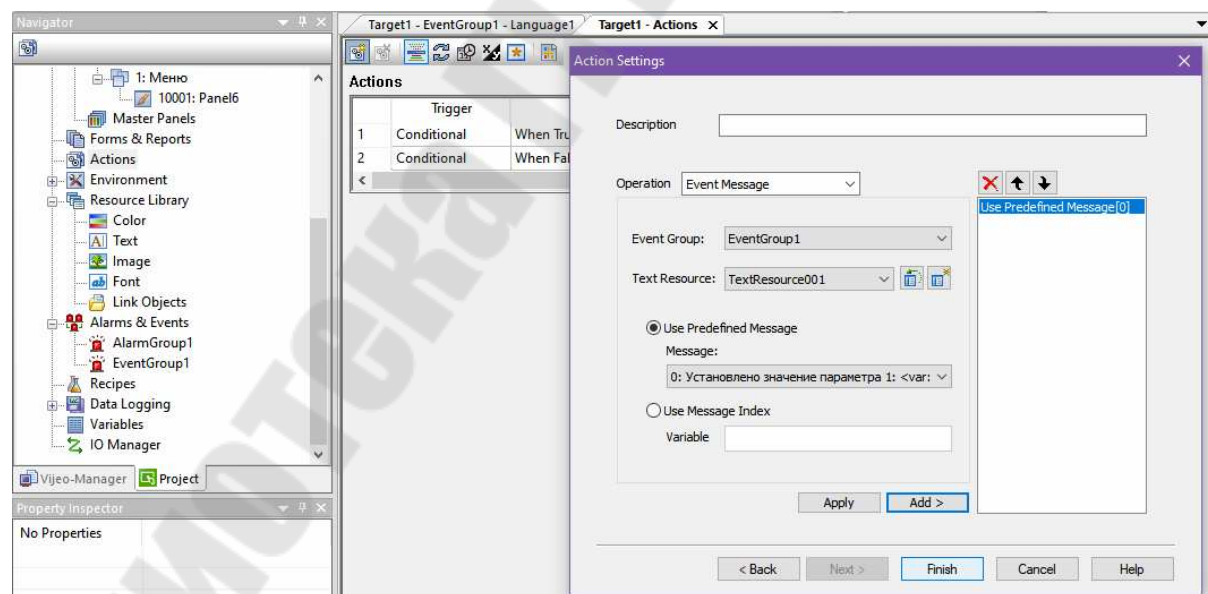


Рис. 5.8. Выбор операции

Создадим панель журнала событий. Для этого в разделе «Основные панели» (*Base Pannels*) дерева проекта создадим новую панель (*New Pannel*) и переименуем ее в «Журнал событий» (рис. 5.9).

В меню *Рисунок (Draw)* → *Журнал тревог (Alarm Summary)* выберем инструмент «*Журнал событий*» (*Event Summary*) (его также можно выбрать на панели инструментов) и разместим изображение журнала на созданной панели.

Двойным нажатием левой кнопкой мыши на созданном журнале откроем его свойства. На вкладке «*Общие*» (*General*) в пункте «*Группы тревог и событий*» (*Alarm & Event Group*) укажем *EventGroup1*, а в пункте «*Список тревог и событий*» (*Alarm & Event List*) выберем «*Журнал*» (*Log*) (рис. 5.9).

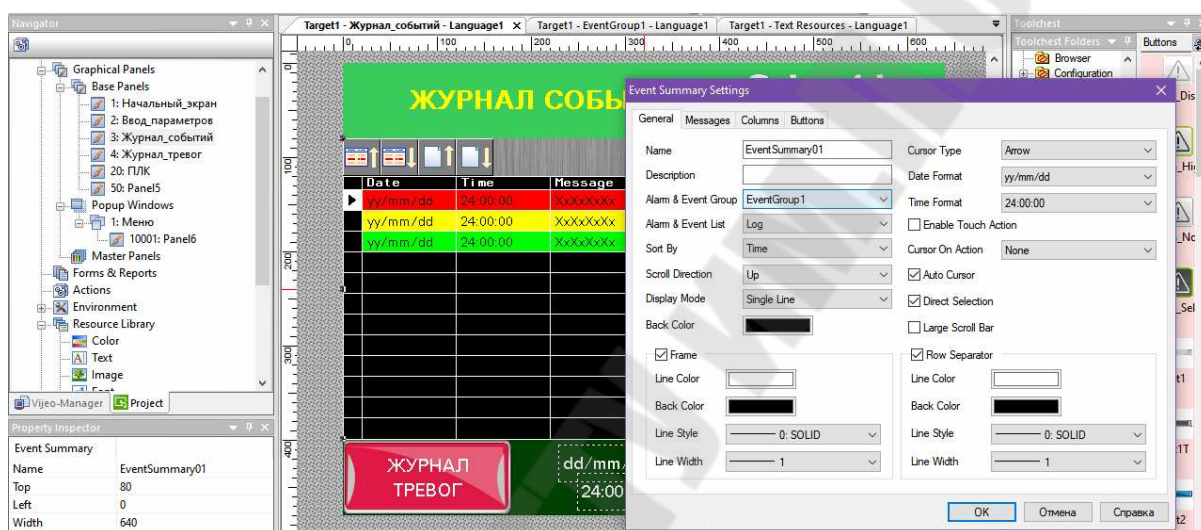


Рис. 5.9. Создание панели раздела дерева

Результат можно посмотреть в симуляторе (рис. 5.10).

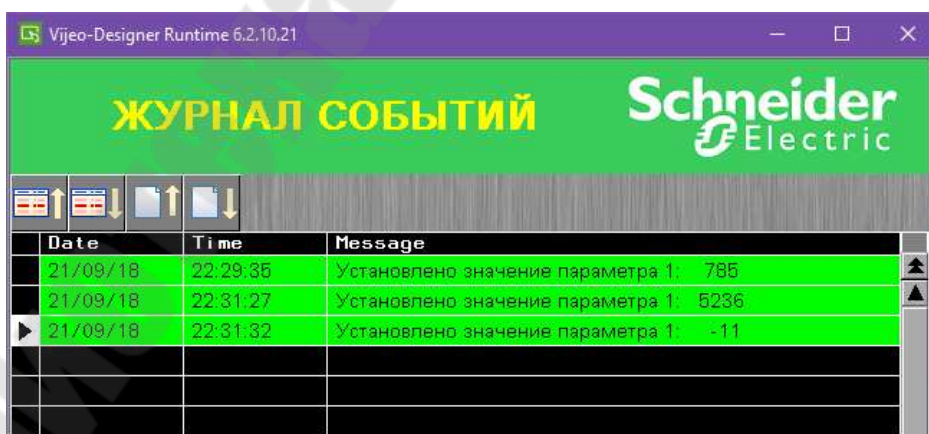


Рис. 5.10. Симулятор

## 6. Создание журнала тревог

В окне навигатора проекта (*Navigator*) нажатием правой кнопки мыши по элементу *Base Panels* в дереве проекта вызовем всплывающее меню, в котором выберем *New Panel* (рис. 6.1). Назовем новую панель «Журнал тревог».

Для создания самого журнала тревог выбираем на панели инструментов инструмент *Alarm Summary* и размещаем изображение журнала на панели.

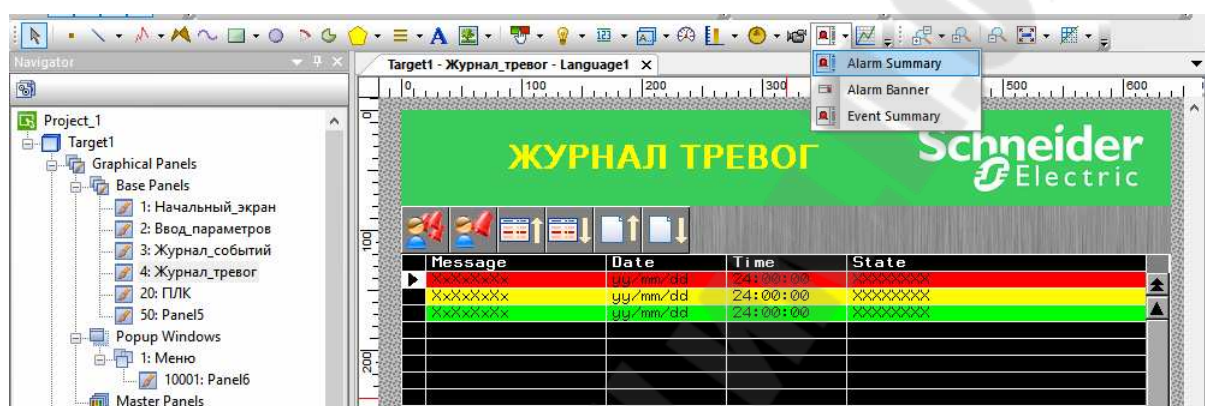


Рис. 6.1. Вызов всплывающего меню

Выберем в дереве проекта раздел переменных (*Variables*) и создадим переменную «Тревога» типа *bool* (рис. 6.2). При создании переменной во вкладке *Alarm* запишем текст сообщения о тревоге. Для этого в разделе *Configuration* нажимаем кнопку *Alarm Message* и во всплывающем окне пишем «Сработала защита».

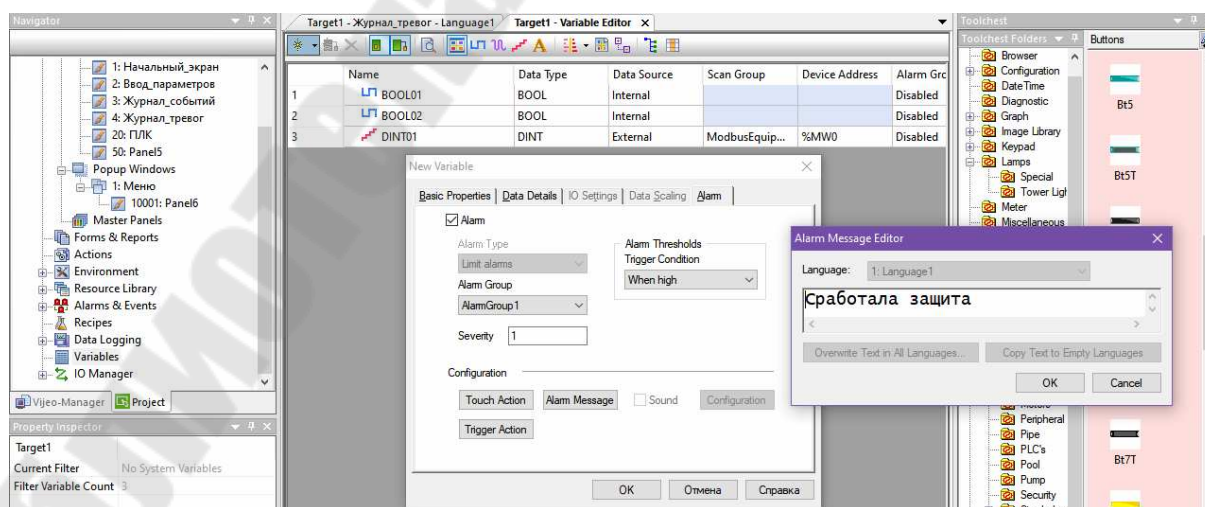


Рис. 6.2. Создание переменной типа *bool*



В дереве проекта выберем раздел *Alarm & Events* и добавим группу *AlarmGroup1*. В группе *AlarmGroup1* создадим внутреннюю переменную типа *bool* с именем «Тревога» (рис. 6.3).

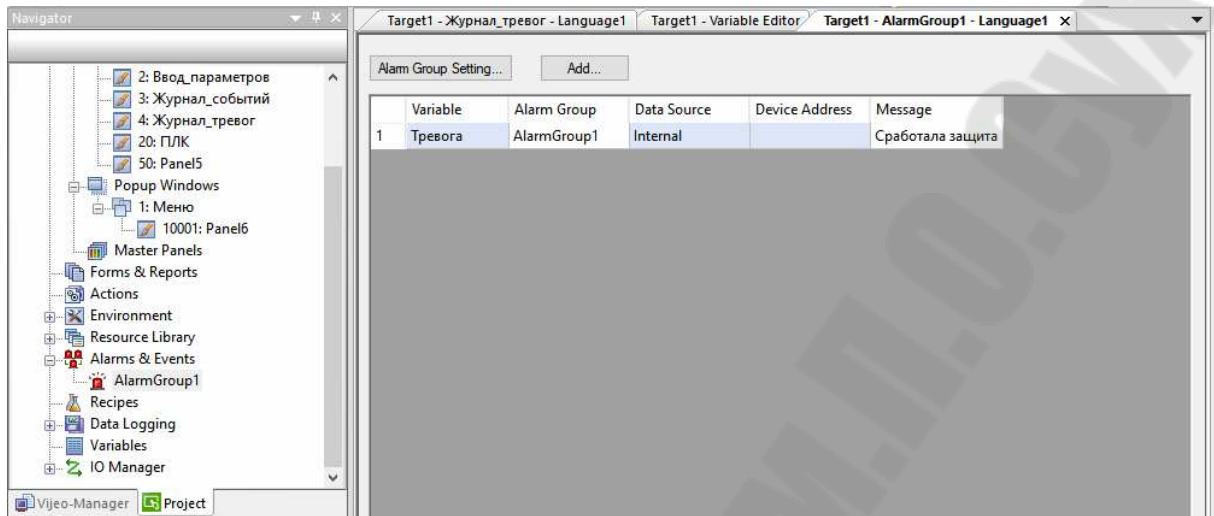


Рис. 6.3. Создание внутренней переменной

Двойным кликом по таблице тревог вызываем всплывающее окно настроек *Alarm Summary Settings*, в котором устанавливаем параметр *Alarm & Events List* в значение *Log* (журнал) (рис. 6.4).

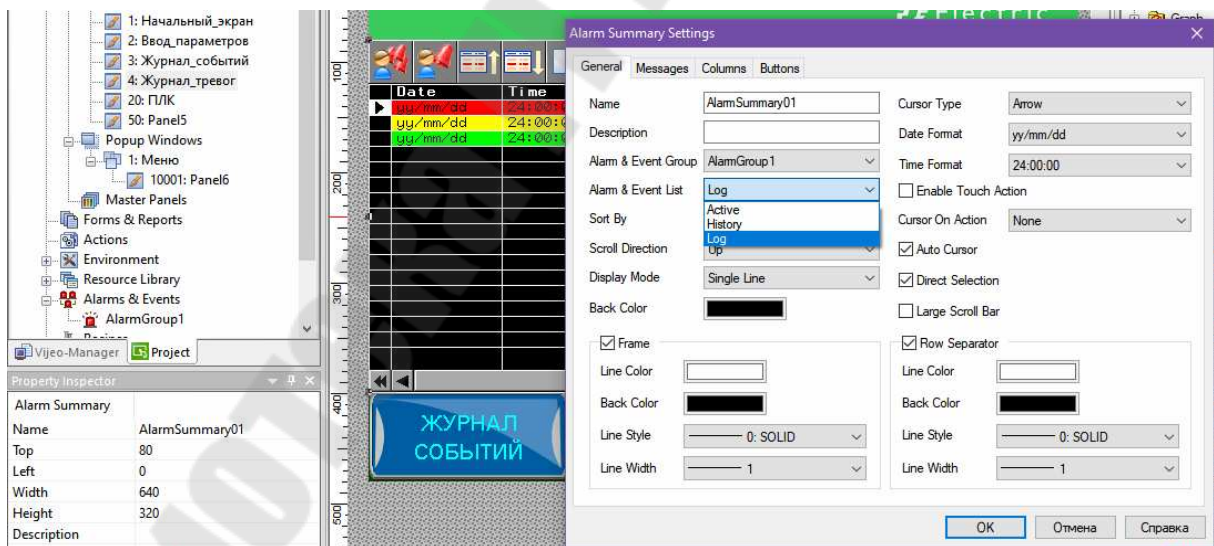


Рис. 6.4. Установка параметра

## 7. Отображение на экране программируемых терминалов информации о состоянии связи с программируемыми логическими контроллерами и режиме работы программируемых логических контроллеров

Для совместной симуляции программ *Machine Expert – Basic* и *Vijeo Designer* установим адрес ПЛК. Для этого в дереве проекта выберем раздел *IO Manager* и перейдем в подраздел оборудования шины *Modbus (ModbusEquipment01)* (рис. 7.1). По двойному нажатию левой кнопки мыши на названии *ModbusEquipment01* появится окно настройки *Equipment Configuration*. Здесь укажем IP-адрес ПЛК (*IP Address*) 127.0.0.1, необходимый для совместной симуляции.

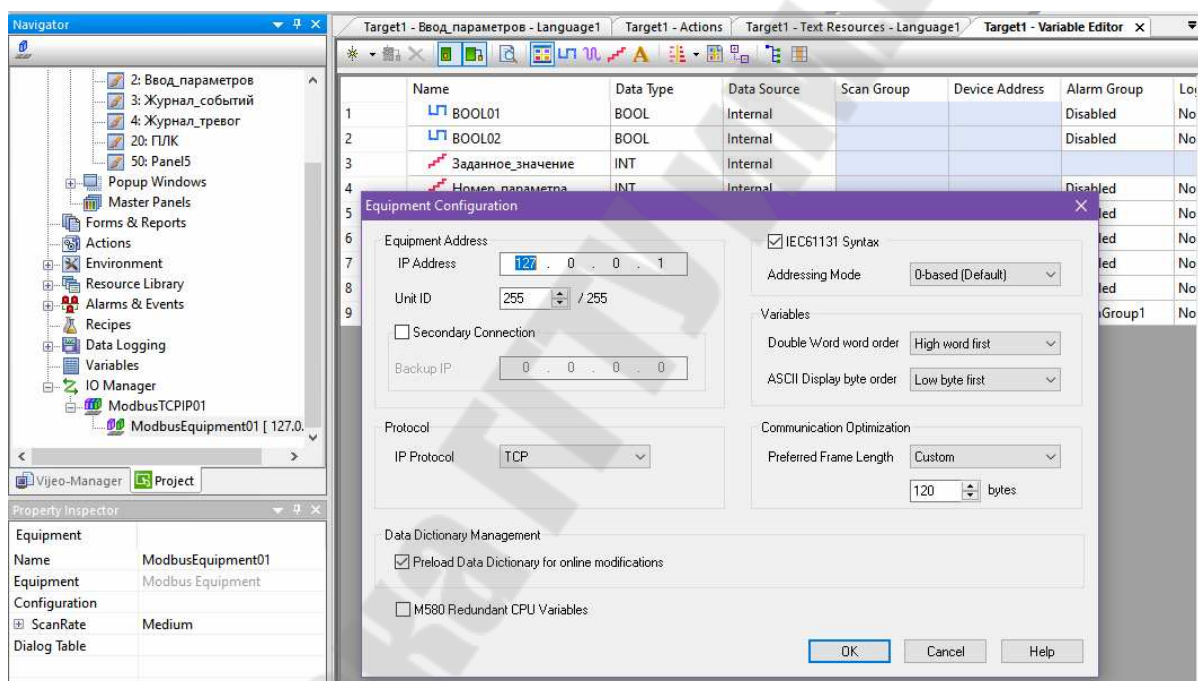


Рис. 7.1. Выбор раздела и подраздела

Создадим внешнюю переменную типа *bool* с именем *PLC\_RUN\_STOP* с параметром *%S12*.

Создадим также внутреннюю переменную типа *bool* с именем *Connection*.

Еще создадим две внешние переменные типа *INT* с именами *GOOD* и *BAD* с адресами *@stat:0* и *@stat:1*, соответственно (рис. 7.2).

Содержание    Указатель    Поиск

- 5.2.6 Changing Equipment I
- 5.2.7 Setting Up Redundant
- 5.3 Collecting Driver Statistics**
  - 5.3.1 Creating Variables to C
  - 5.3.2 Displaying Driver Stati
  - 5.3.3 Rolling Over Driver Sti
- 5.4 Working with Driver Properti
  - 5.4.1 Modbus Plus
  - 5.4.2 Modbus RTU and TCI
  - 5.4.3 SoMachine - Network.
- 5.5 Working with Alarms on the
- 5.6 Working with the Dialog Tab
- 5.7 Working with the Reflex Tab
- 5.8 Equipment Driver Troublesh
- 6. Creating Variables
  - 6.1 About Variables and Device
    - 6.1.1 Variables
    - 6.1.2 Variables and Data Ty
    - 6.1.3 Variables and Device
    - 6.1.4 Source: Internal Versu
  - 6.2 Setting Up Variables
  - 6.3 Connecting to Equipment
  - 6.4 Creating Internal Variables

### Current Statistics (@stat) and Previous Statistics (@prevstat)

The driver collects the current and previous statistics listed in the following table. In the variable's Device Address to view the current statistics or previous statistics, then type a colon and the index number.

Examples:

```
@stat:2 = Total Count
@prevstat:8 = Initialization Failed
```

For details on how to set up variables so you can display driver statistics, see Section 5.3.1, [Creating Variables t](#)

Index	Name	Description
0	Total Good	Received without error.
1	Total Bad	The total number of errors of all types.
2	Total Count	Total Good + Total Bad.
3	No Reply	No reply was received in response to the request.
4	Checksum Failed	Received a request or response *1 with an incorrect checksum.
5	Wrong Address	Received a request or response *1 with an incorrect address.

The screenshot shows the Vjeco-Manager software interface. On the left is a 'Navigator' pane with a tree view of graphical panels and variables. The main window is titled 'Target1 - Variable Editor' and contains a table of variables:

Name	Data Type	Data Source	Scan Group	Device Address	Alarm Group	Location
1	BOOL01	BOOL	Internal		Disabled	No
2	BOOL02	BOOL	Internal		Disabled	No
3	Connection	BOOL	Internal		Disabled	No
4	GOOD	INT	External	ModbusEquip...	@stat:0	Disabled
5	INT01	INT	External	ModbusEquip...		Disabled
6	PLC_RUN_STA				Disabled	No
7	Заданное_зн				Disabled	No
8	Номер_пара				Disabled	No
9	Параметр_1				Disabled	No
10	Параметр_2				Disabled	No
11	Параметр_3				Disabled	No
12	Параметр_4				Disabled	No
13	Тревога				AlarmGroup1	No

The 'Variable Properties' dialog box for 'INT01' shows the following settings:

- Variable Name: INT01
- Description: (empty)
- Data Type: INT
- Array Dimension: 0
- Data Source: External
- ScanGroup: ModbusEquipment01
- Device Address: @stat:1
- Indirect Address: (unchecked)

Рис. 7.2. Создание переменных

Создадим текстовый ресурс *TextResource002*, содержащий сообщения «Панель оператора включена», «ПЛК перешел в режим RUN», «ПЛК перешел в режим STOP», «Обрыв связи с ПЛК», «Связь с ПЛК восстановлена» (рис. 7.3).

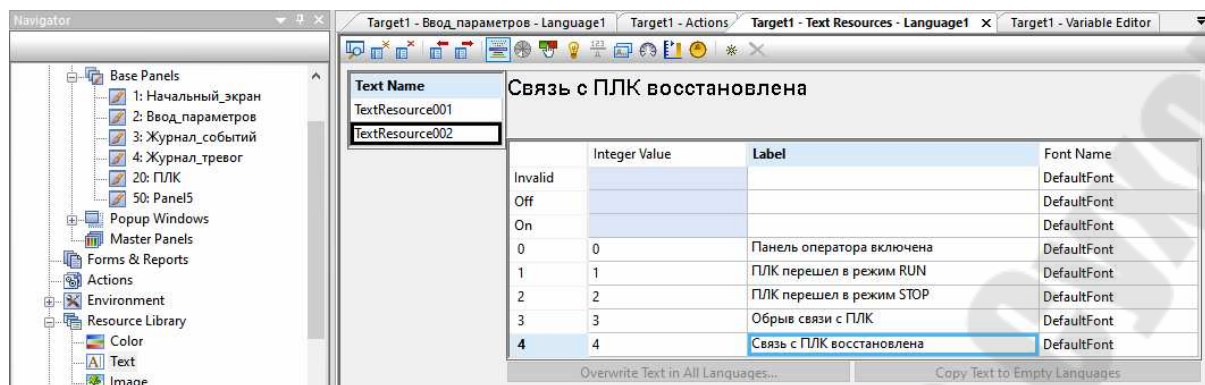


Рис. 7.3. Создание текстового ресурса

Создадим группу событий *EventGroup1*. Создадим несколько экшенов.

Первый экшен выводит сообщение о запуске панели оператора. Для этого в окне навигатора (*Navigator*), в дереве проекта выберем раздел *Actions* (рис. 7.4). На панели задач открывшегося окна *Actions* нажмем значок «Создать новое действие» (*Create a new Action*). В появившемся окне настроек (*Action Settings*) укажем тип срабатывания (*Trigger Type*) по событию (*Event*). Нажмем кнопку *Next*.

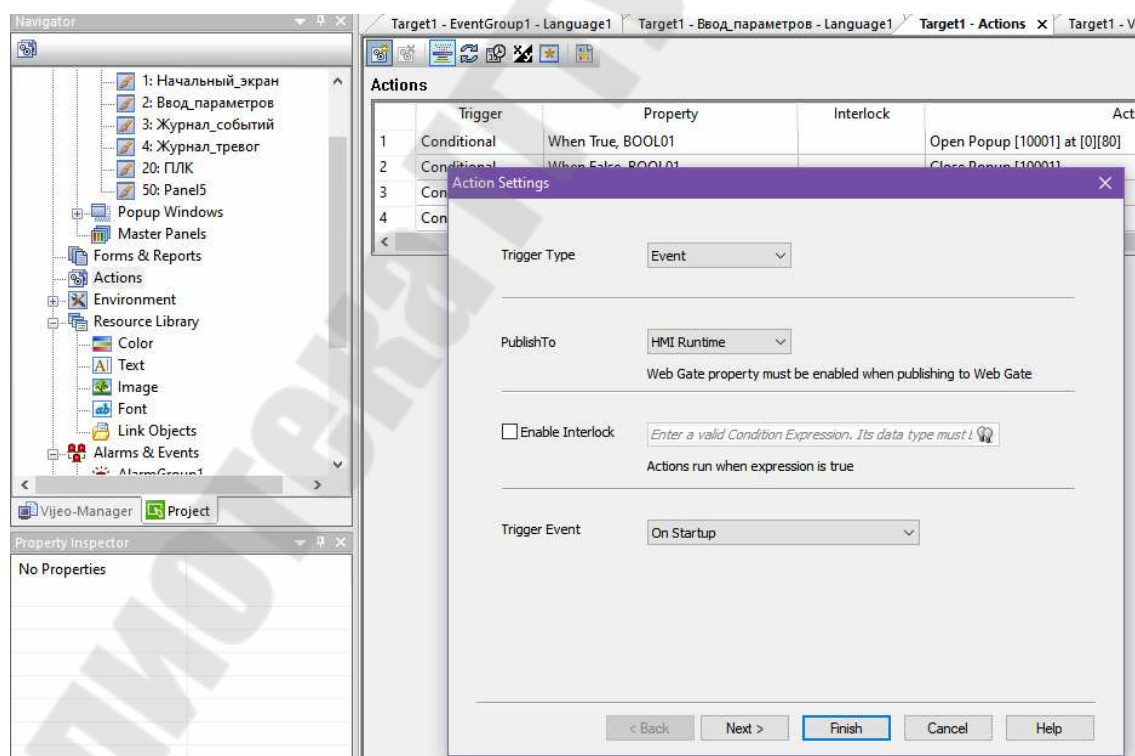


Рис. 7.4. Выбор раздела

В появившемся окне в качестве операции (*Operation*) выберем «Сообщение о событии» (*Event Message*) (рис. 7.5). В качестве выводимого сообщения (*Text Resource*) выберем *TextResource\_2*, выберем ранее созданное сообщение (*Use Predefinite Message*) «0: Панель оператора включена», обязательно нажмем кнопку «Добавить» (*Add*) и «Завершить» (*Finish*).

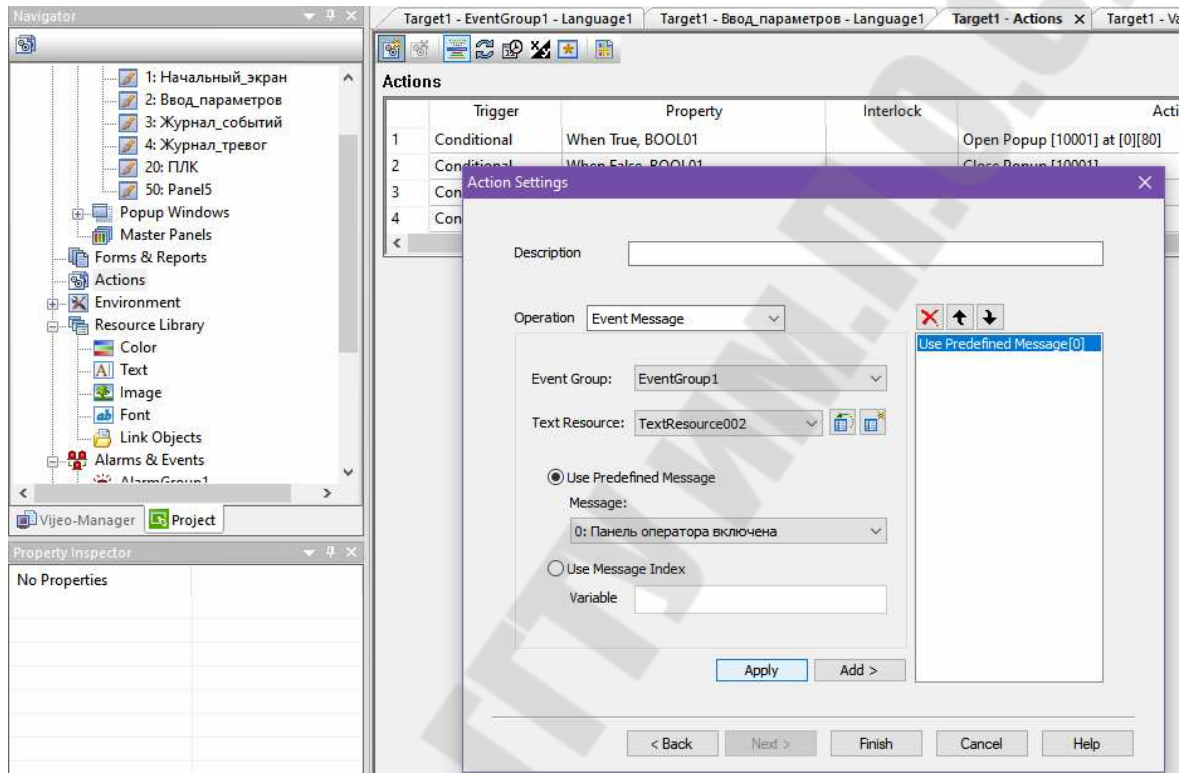


Рис. 7.5. Выбор операции

Второй и третий экшны выводят сообщения о переходе ПЛК в режим *RUN* и *STOP* соответственно (рис. 7.6, 7.7).

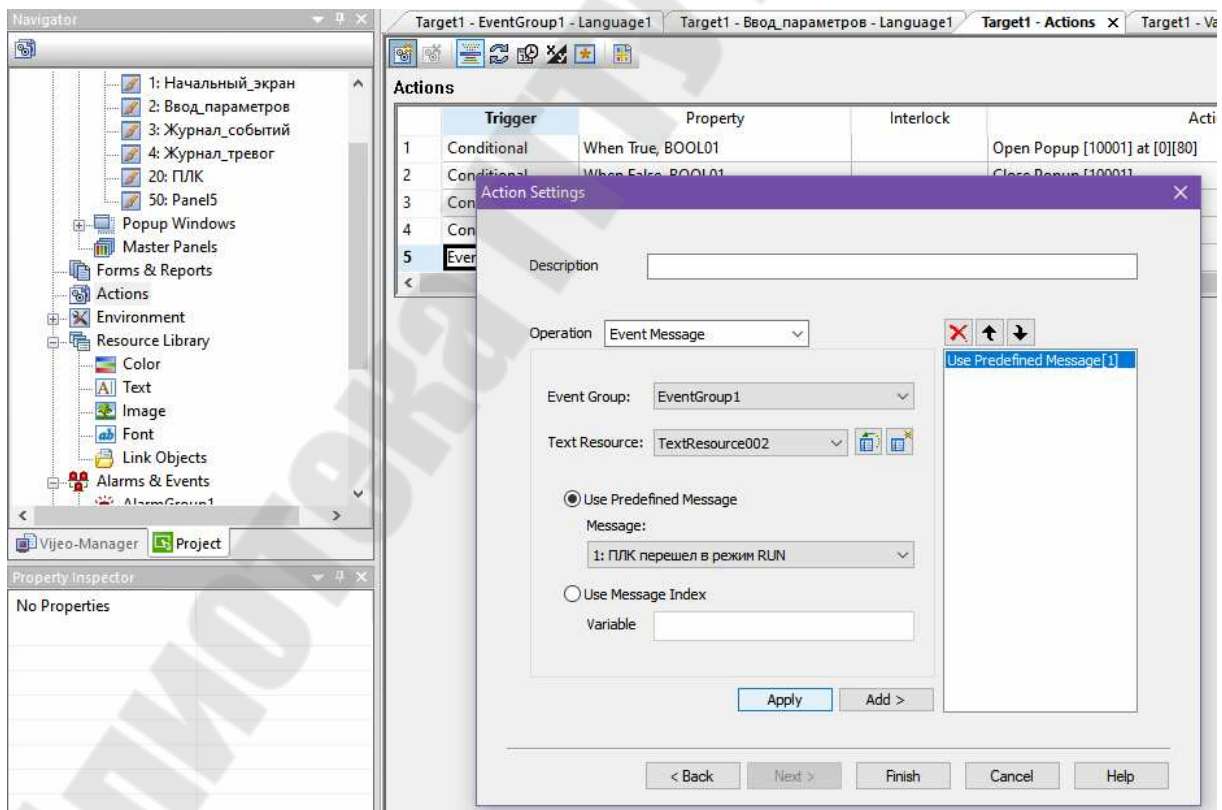
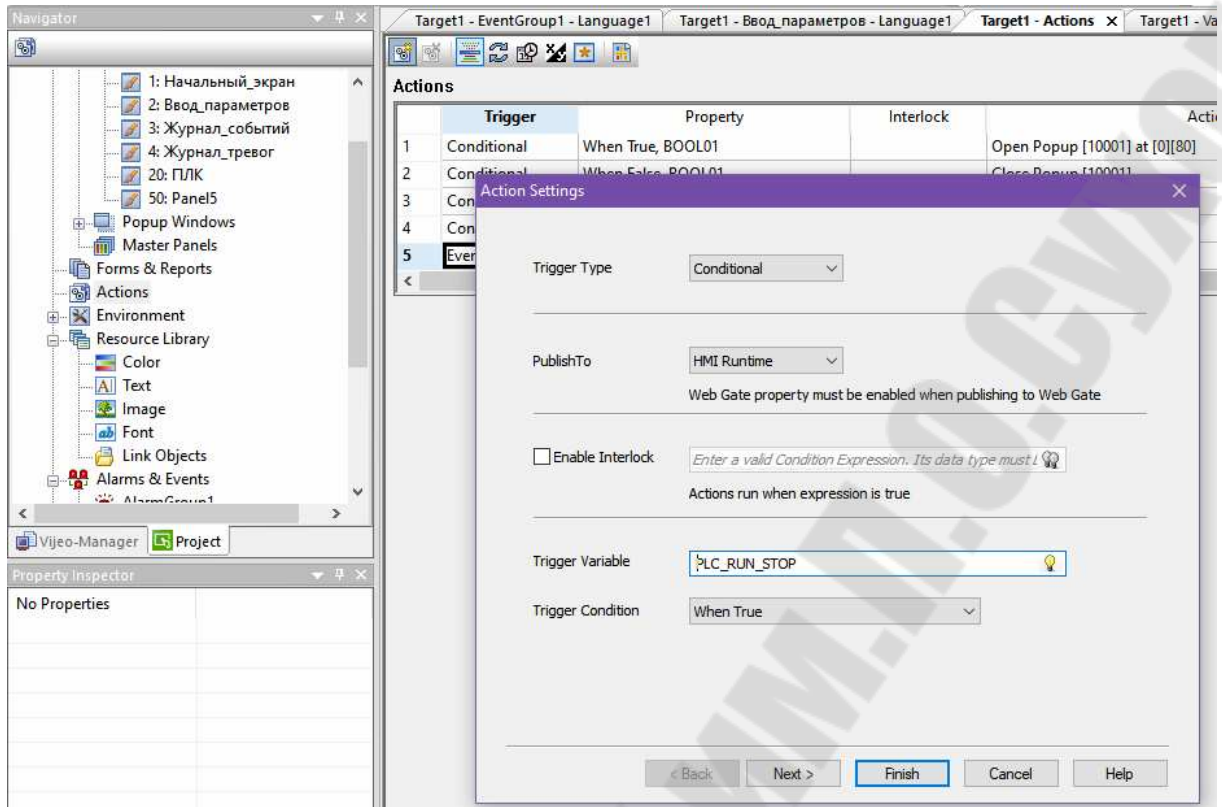


Рис. 7.6. Переход программирующего логического контроллера в режиме RUN

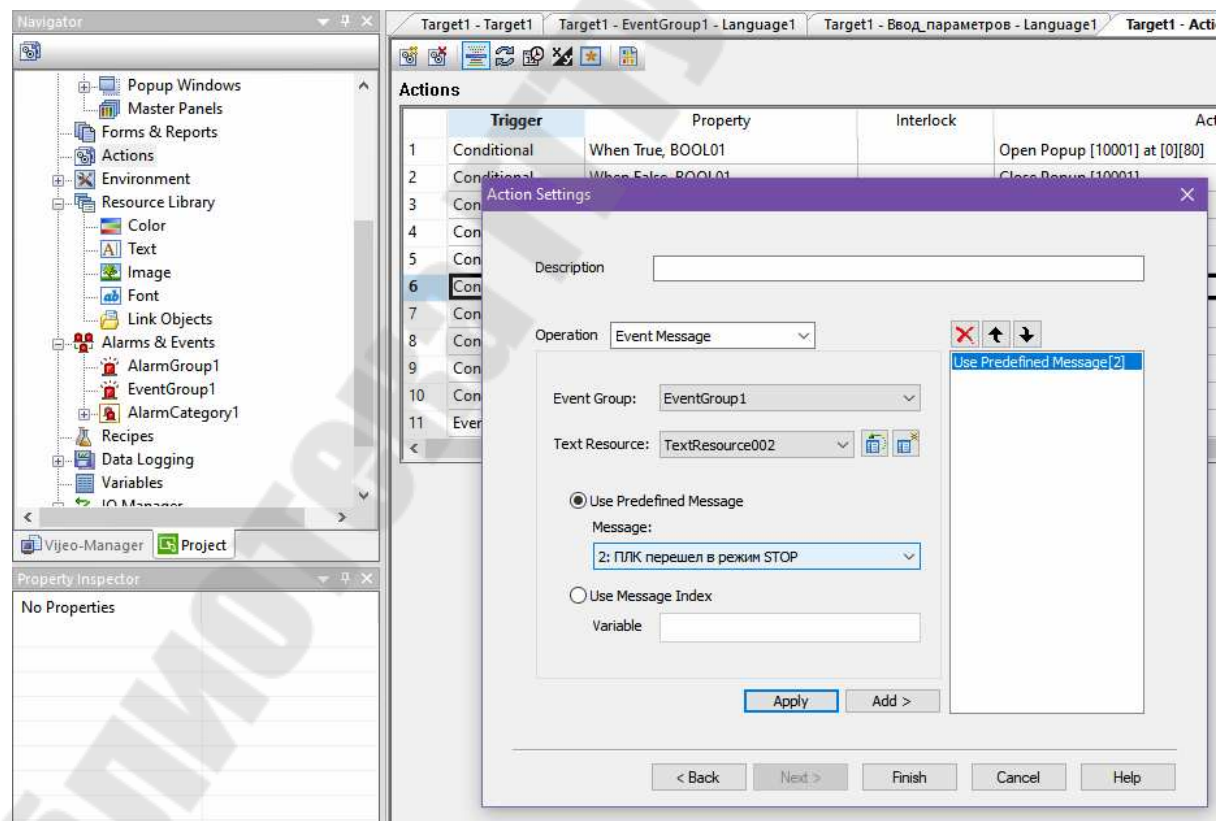
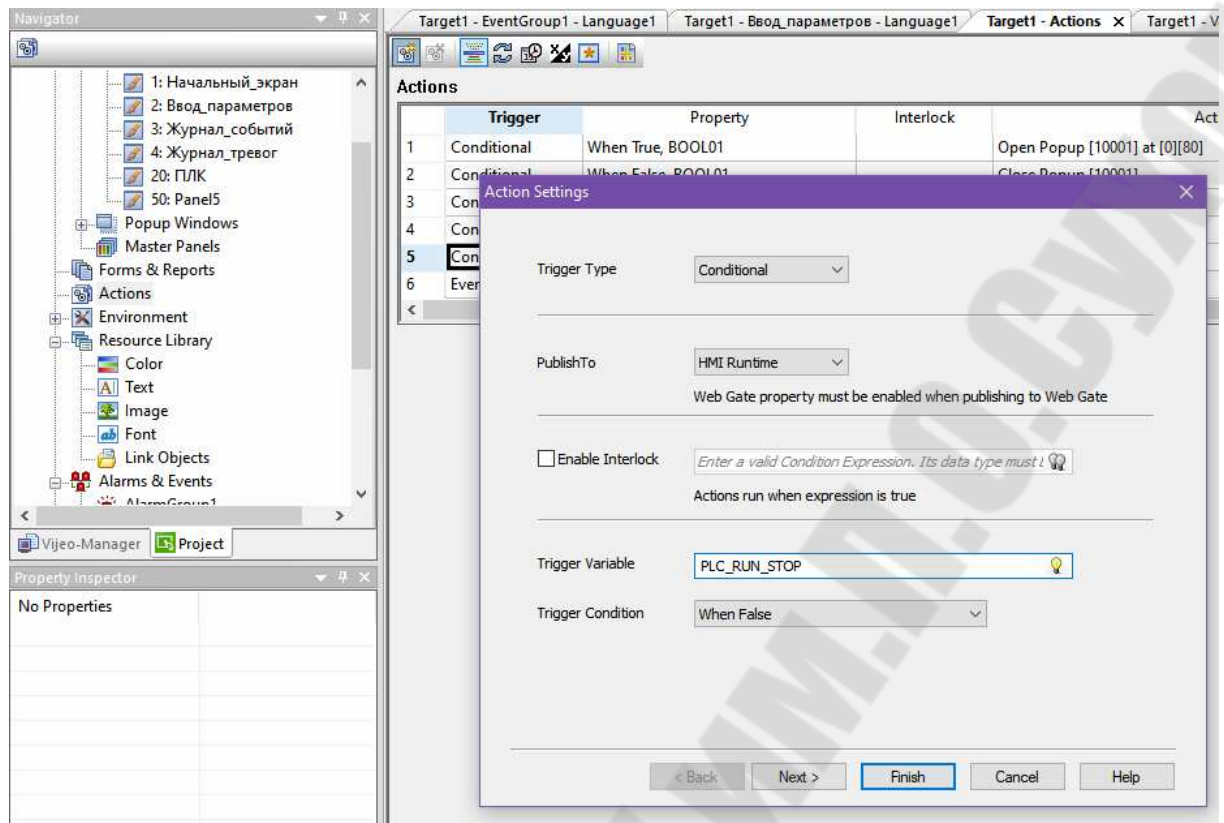


Рис. 7.7. Переход программируемого логического контроллера в режиме *STOP*

Аналогичным способом создадим еще два экшена для управления битовой переменной *Connection*. Системная переменная *GOOD* устанавливает переменную *Connection* в 0, а системная переменная *BAD* – в 1 (рис. 7.8–7.11).

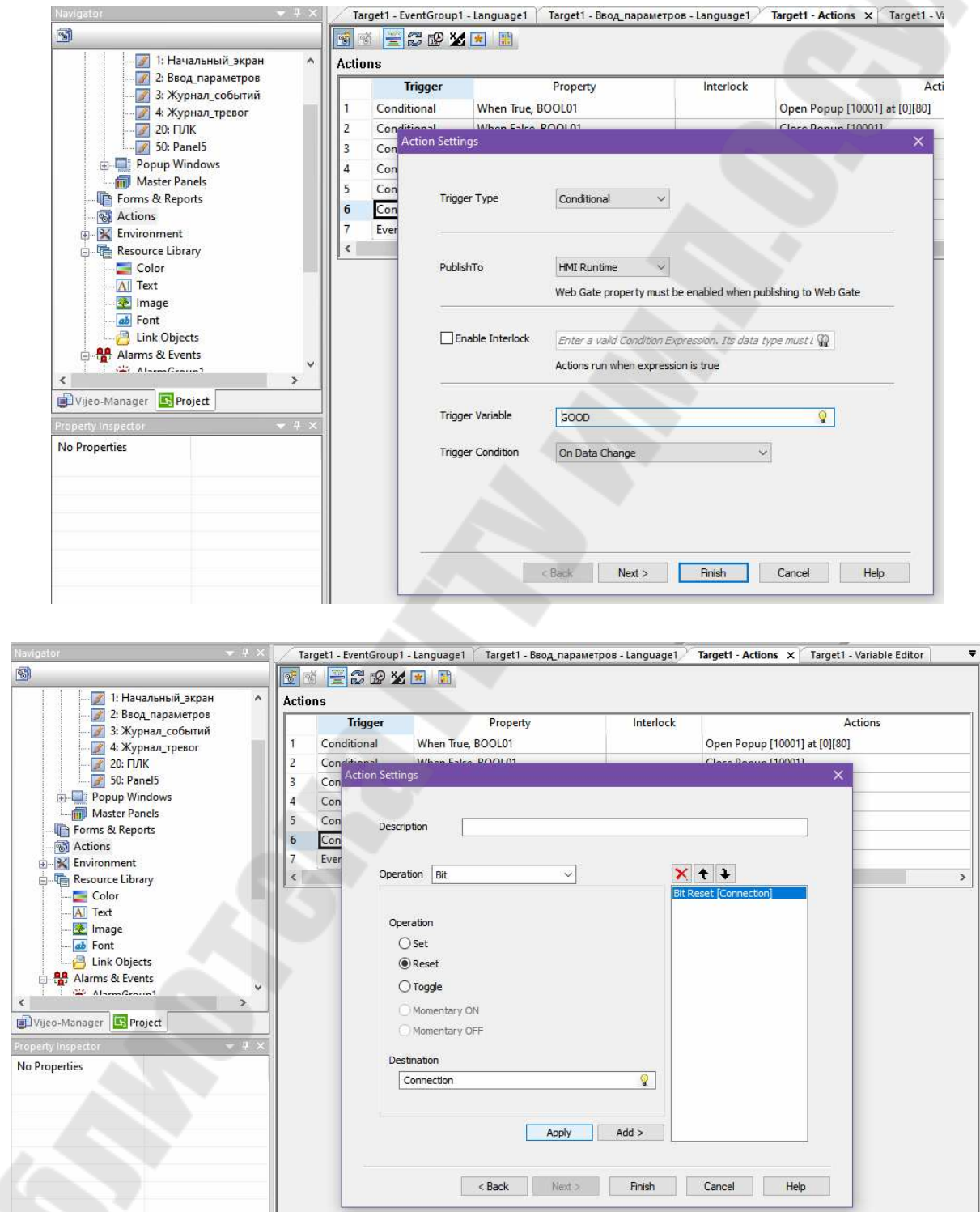


Рис. 7.8. Создание переменной переменной *GOOD*



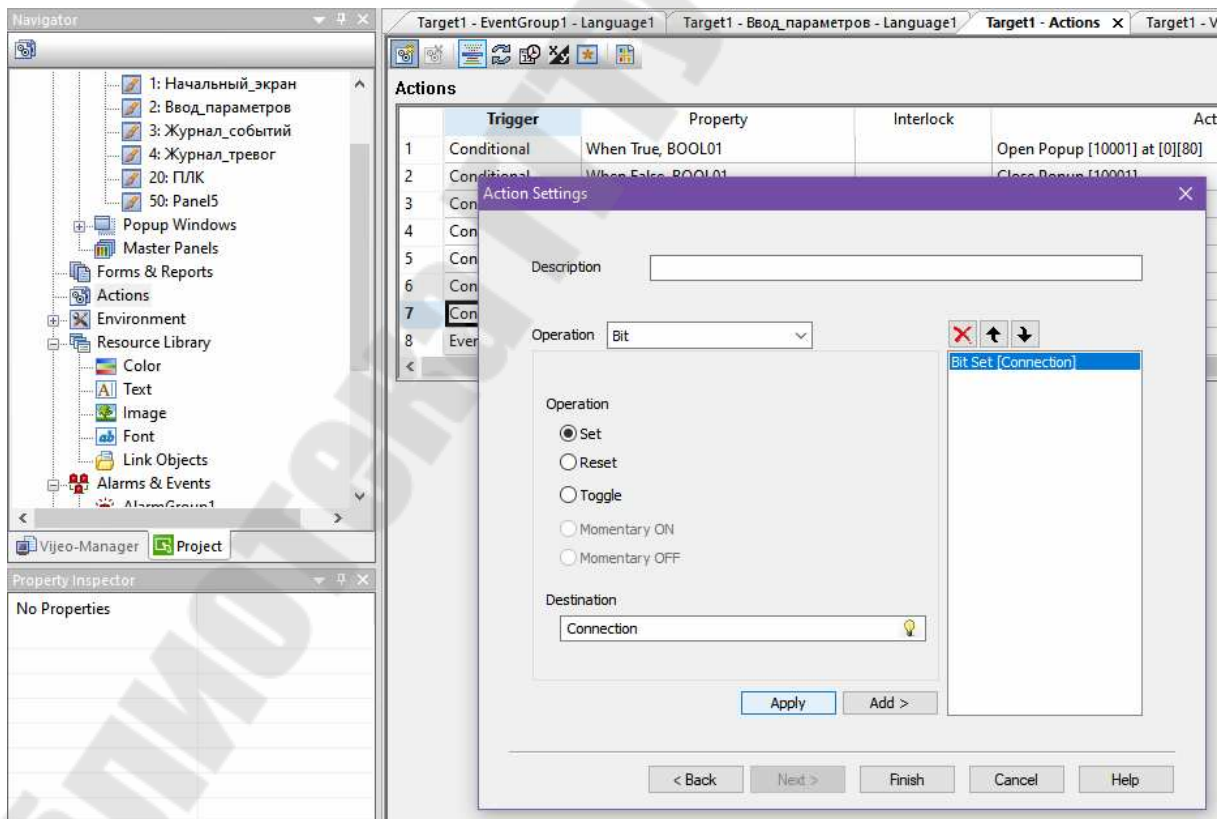
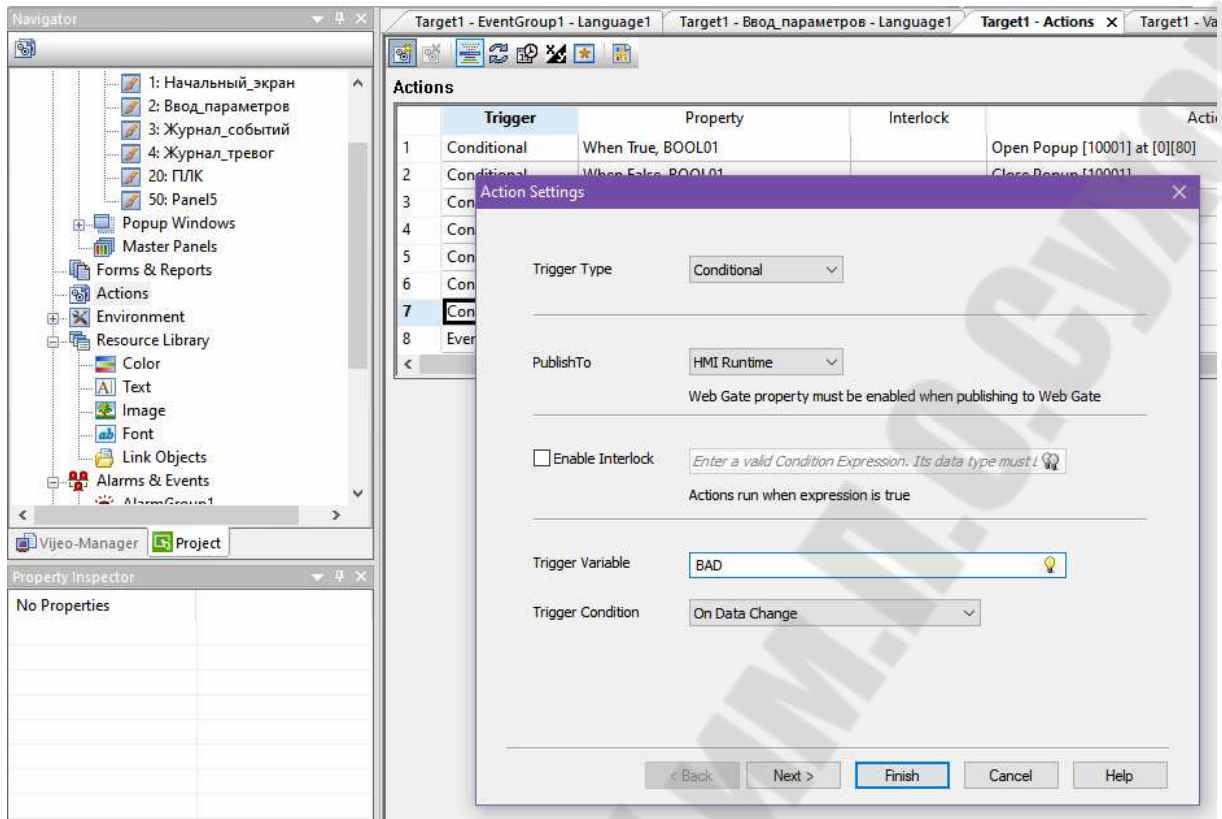


Рис. 7.9. Установка переменной BAD

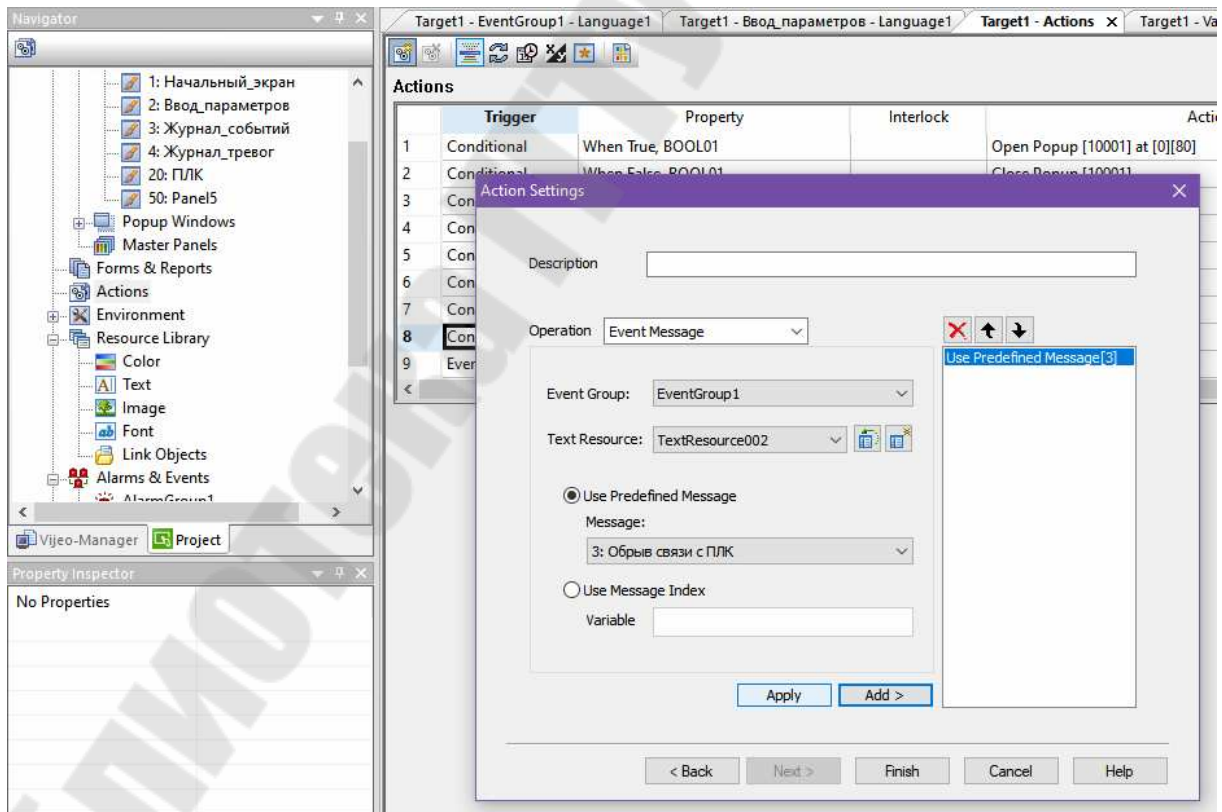
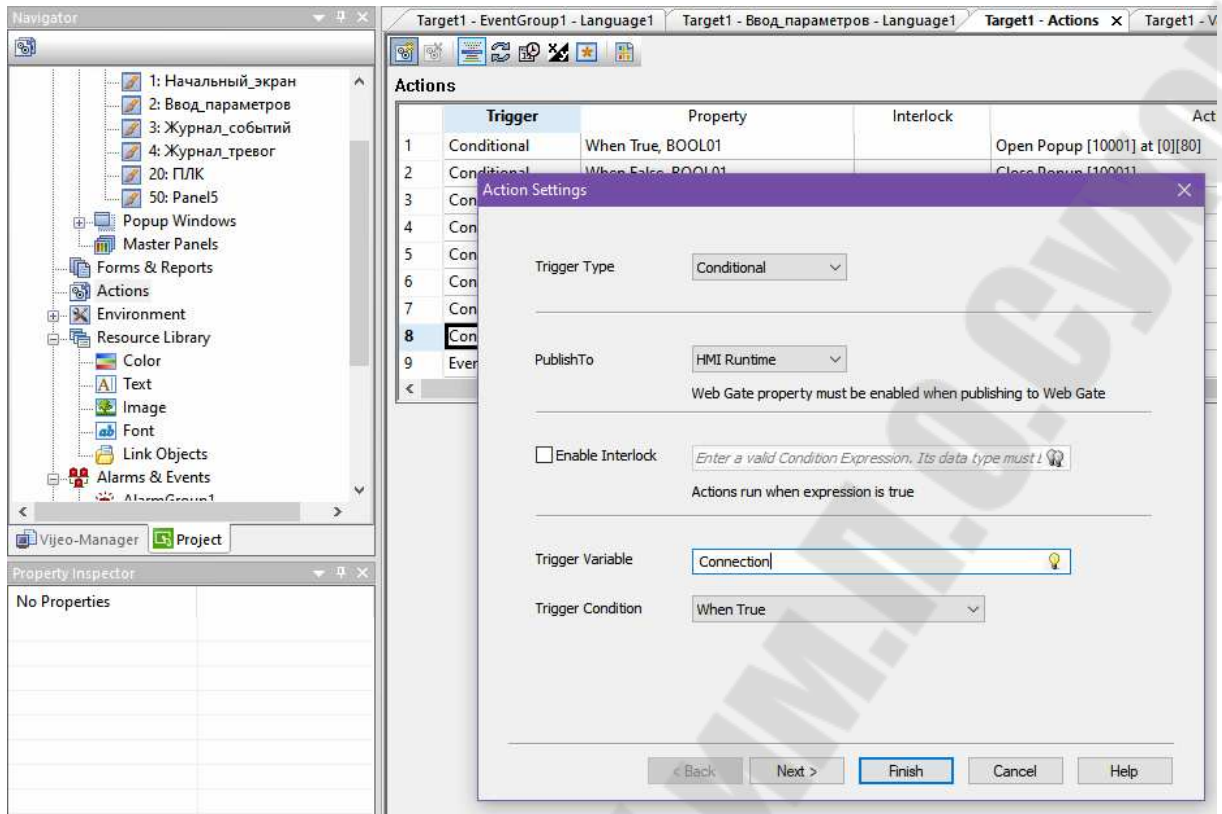


Рис. 7.10. Установка переменной Connection

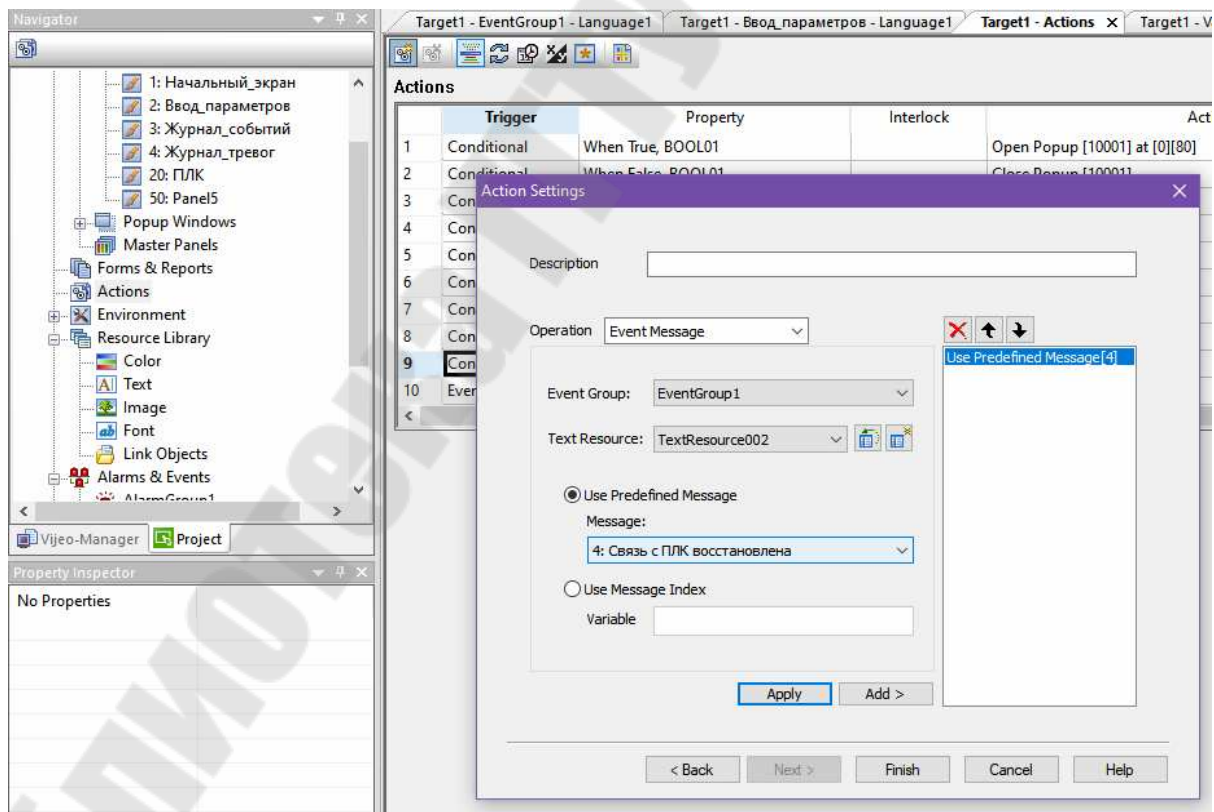
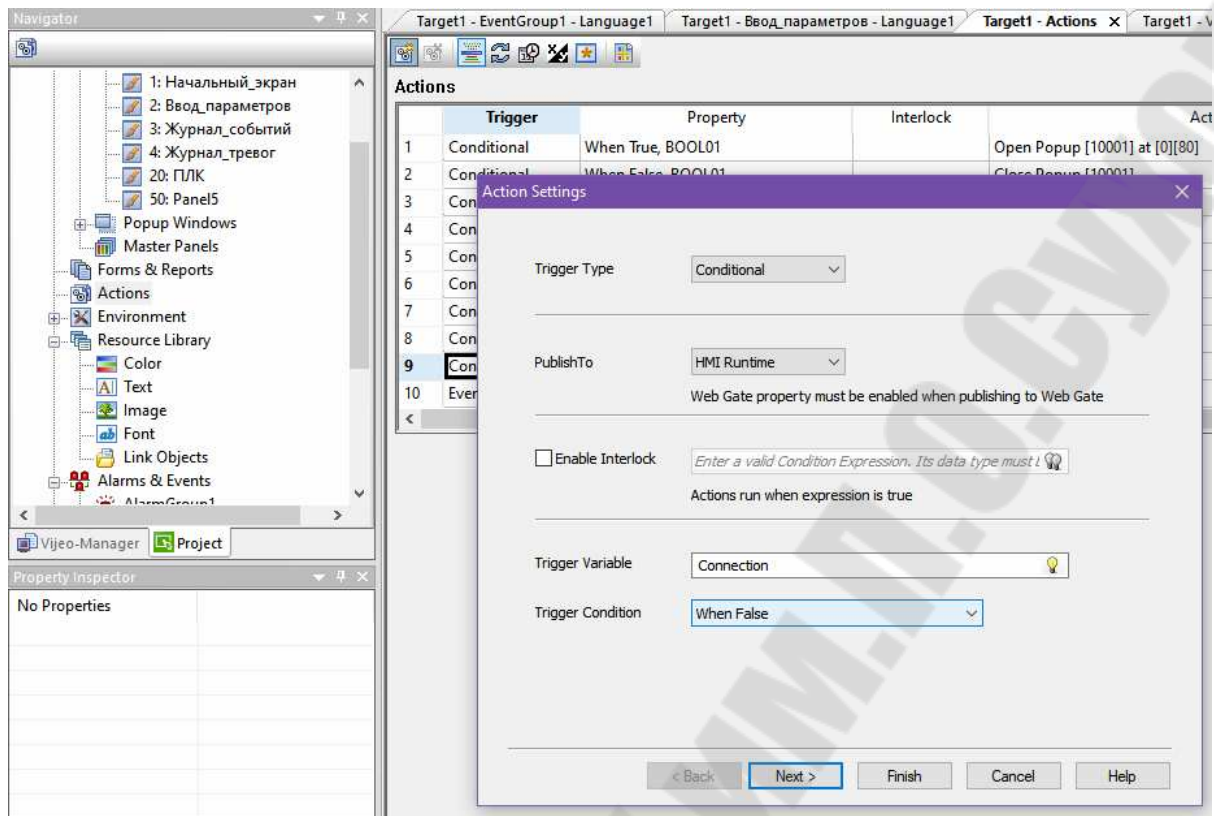


Рис. 7.11. Установка переменной Connection

Таким образом, в симуляторе можно наблюдать работу программы, представленную на рис. 7.12.

Date	Time	Message
21/09/20	11:34:09	Панель оператора включена
21/09/20	11:34:09	ПЛК перешел в режим RUN
21/09/20	11:34:39	ПЛК перешел в режим STOP
21/09/20	11:34:55	Обрыв связи с ПЛК
21/09/20	11:35:12	Связь с ПЛК восстановлена
21/09/20	11:35:18	ПЛК перешел в режим RUN
21/09/20	11:35:42	Сработала защита
▶ 21/09/20	11:35:42	Сработала защита

Рис. 7.12. Работа программы

## 8. Конфигурирование преобразователя частоты для работы по протоколу *Modbus*

Прежде чем приступить к конфигурированию преобразователя частоты, настоятельно рекомендуется вернуться к заводским настройкам. Для этого необходимо с помощью энкодера и дисплея на передней панели преобразователя частоты установить следующие значения параметров:

COнF->FCS- ->FCSI = In1 – заводская конфигурация;

COнF->FCS- ->FrY- = ALL – все параметры;

COнF->FCS- ->GFS = YES – возврат к заводским настройкам.

Для обеспечения оптимальной эффективности ПЧ в части точности и времени отклика необходимо выполнить некоторые действия.

Ввести значения параметров, указанных на паспортной табличке двигателя, в меню [ПРИВОД]:

COнF->FULL->drC- ->

bFr = 50 (Гц) – частота стандартного двигателя;

tFr = 60 ( $\leq 10 \times bFr$ ) (Гц) – максимальная частота;

Ctt = Std – скалярное управление;

COнF->FULL->drC->ASU- ->

nPr = 0.75 (кВт) – номинальная мощность;

$UnS = 380$  (В) – номинальное напряжение двигателя;  
 $nCr = 1.9$  (А) – номинальный ток статора;  
 $FrS = 50$  (Гц) – номинальная частота тока статора;  
 $nSP = 2820$  (об/мин) – номинальная частота вращения вала;  
 $tUn = YES$  (2 с) – автоподстройка параметров двигателя;  
 $tUS = dOnE$ ;  
 $Stun = PEAS$  – используется результат автоподстройки;  
 $tUnU = tП$  – оценка теплового состояния по потребляемому току;

$AUt = nO$  – настройка деактивирована;  
 $PPC = nPr$  – мощность двигателя;  
 $rSA = 10.00$  мОм;  
 $LFA = 36.31$  мГн;  
 $IdA = 1.4$  А;  
 $trA = 77$  мс;  
 $CO nF \rightarrow FULL \rightarrow drC - \rightarrow$   
 $SFt = HF2$  – оптимизация шума ЭД;  
 $SFr = 16.0$  – частота модуляции;  
 $nrd = YES$  – уменьшение шума;  
 $CO nF \rightarrow FULL \rightarrow I\_O \rightarrow$   
 $tCC = 2C$  – двухпроводная схема;  
 $tCt = trn$  – изменение состояния;  
 $run = no$  – не задан @no (только для 3-проводной схемы);  
 $Frd = LI1$  – дискретный вход 1 – вперед @Li1;  
 $rrS = LI2$  – дискретный вход 2 – назад @Li2;  
 $CO nF \rightarrow FULL \rightarrow CtL - \rightarrow$   
 $Frl = AI1$  – задание частоты через аналоговый вход 1 @AI1;  
( $Frl = Пdb$  – задание частоты через Modbus);  
 $CO nF \rightarrow FULL \rightarrow SIП - \rightarrow$   
 $tCC = 2C$  – двухпроводная схема.

Перечисленные выше параметры преобразователя частоты можно установить с помощью программного обеспечения *SoMove* (рис. 8.1).

Чтобы производить управление преобразователем частоты с помощью станции *Master Modbus* необходимо выбрать шину *Modbus* в качестве активного канала команд.

Code	Long Label	Conf0	Default Value	Min Value
FAB	Frequency boost	0 Hz	0 Hz	0 Hz
SVL	Motor surge limitation	No	No	
SOP	Attenuation time	10 μs	10 μs	
VBR	Braking level	820 V	820 V	820 V
LBA	Load sharing	No	No	
LBC	Load correction	0 Hz	0 Hz	0 Hz
LBC1	Correction min speed	0 Hz	0 Hz	0 Hz
LBC2	Correction max speed	0.1 Hz	0.1 Hz	0.1 Hz
LBC3	Torque offset	0 %	0 %	0 %
LBF	Sharing filter	100 ms	100 ms	0 ms
▼ Asynchronous motor				
NPR	Nominal motor power	0.75 kW	0.75 kW	0.09 kW
COS	Motor 1 Cosinus Phi	0.77	0.77	0.5
UNS	Nominal motor voltage	380 V	400 V	200 V
NCR	Nominal motor current	1.9 A	2 A	0.5 A
FRS	Nominal motor frequency	50 Hz	50 Hz	10 Hz
NSP	Nominal motor speed	2820 rpm	1400 rpm	0 rpm
STUN	Tune selection	Measure	Default	
TUNU	Autotuning usage	Use the motor thermal evolution	Use the motor thermal evolution	
AUT	Automatic autotune	No	No	
FLU	Motor fluxing configure	No	No	
RSA	AsyncMotor Stator resistance	10078 mOhm	0 mOhm	0 mOhm
LFA	AsyncMotor Leakage inductance	36.31 mH	0 mH	0 mH
IDA	Magnetizing current	1.4 A	0 A	0 A
TRA	Rotor time constant	77 ms	0 ms	0 ms
MPC	Motor parameter choice	Nominal motor power	Nominal motor power	
► Synchronous motor				
► Inputs / Outputs				
► Command				
► Function Blocks				
► Application function				
► Monitoring				
► Communication				

Code	Long Label	Conf0	Default Value	Min Value
► Settings				
▼ Motor control				
BFR	Motor Standard	50Hz motor frequency	50Hz motor frequency	
TFR	Max frequency	60 Hz	60 Hz	10 Hz
CTT	Motor control type	U/F VC Standard motor law	U/F VC Standard motor law	
SPG	Speed proportional gain	40 %	40 %	0 %
SPGU	Inertia factor	40 %	40 %	0 %
SIT	Speed time integral	63 ms	63 ms	1 ms
SFC	K speed loop filter	65	65	0
FFH	Filter time of the estimated speed	6.4 ms	6.4 ms	0 ms
CRTF	Filter time of the current	3.2 ms	3.2 ms	0 ms
UFR	IR compensation	100 %	100 %	0 %
SLP	Slip Compensation	100 %	100 %	0 %
U1	Volt point 1 on 5pt V/F	0 V	0 V	0 V
F1	Freq point 1 on 5pt V/F	0 Hz	0 Hz	0 Hz
U2	Volt point 2 on 5pt V/F	0 V	0 V	0 V
F2	Freq point 2 on 5pt V/F	0 Hz	0 Hz	0 Hz
U3	Volt point 3 on 5pt V/F	0 V	0 V	0 V
F3	Freq point 3 on 5pt V/F	0 Hz	0 Hz	0 Hz
U4	Volt point 4 on 5pt V/F	0 V	0 V	0 V
F4	Freq point 4 on 5pt V/F	0 Hz	0 Hz	0 Hz
U5	Volt point 5 on 5pt V/F	0 V	0 V	0 V
F5	Freq point 5 on 5pt V/F	0 Hz	0 Hz	0 Hz
CLJ	Current limitation	3.4 A	3.4 A	0 A
SFT	Switching frequency type	Switch.frequency type 2	Switch.frequency type 1	
SFR	Switching frequency	16 kHz	4 kHz	2 kHz
NRD	Motor Noise Reduction	Yes	No	
BOA	Boost activation	Dynamic	Dynamic	
BOO	Boost	0 %	0 %	-100 %
FAB	Frequency boost	0 Hz	0 Hz	0 Hz
SVL	Motor surge limitation	No	No	
SOP	Attenuation time	10 μs	10 μs	
VBR	Braking level	820 V	820 V	820 V

Рис. 8.1. Установка параметров преобразователя частоты

Настройка связи:

COнF->FULL->COП- ->Пd1- ->

Add = 1 – Modbus Address @OFF;

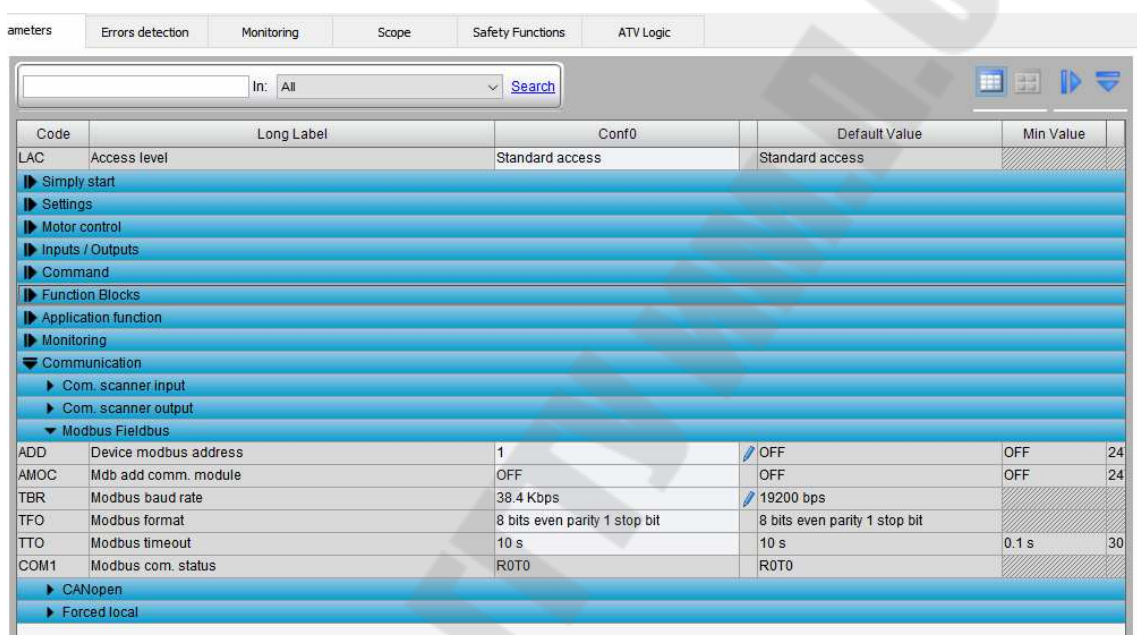
tbr = 38 4 – Modbus baud rate @19 2;

tFO = 8E1 – Modbus format @8E1;

ttO = 10.0 – Modbus Timeout @10;

COП1 = rOtO – состояние связи.

На рис. 8.2 показано, как перечисленные параметры можно изменить в программе *SoMove*.



The screenshot shows the 'Modbus Fieldbus' configuration section in the SoMove software. The table below lists the parameters and their current values.

Code	Long Label	Conf0	Default Value	Min Value
LAC	Access level	Standard access	Standard access	
ADD	Device modbus address	1	OFF	24
AMOC	Mdb add comm. module	OFF	OFF	24
TBR	Modbus baud rate	38.4 Kbps	19200 bps	
TFO	Modbus format	8 bits even parity 1 stop bit	8 bits even parity 1 stop bit	
TTO	Modbus timeout	10 s	10 s	0.1 s 30
COM1	Modbus com. status	ROT0	ROT0	

Рис. 8.2. Изменение параметров в программе *SoMove*

После конфигурации необходимо произвести отключение и затем повторное включение преобразователя частоты для того, чтобы настройки вступили в силу.

Связь по *Modbus* можно установить одним из следующих методов:

- настроить один из последовательных интерфейсов в режим *Modbus Serial IOScanner* с использованием протокола *Modbus RTU*;
- настроить порт *Ethernet* в режим *Modbus TCP IOScanner*.

Для управления преобразователем частоты с помощью программируемого логического контроллера (протокол *RS-485*) необходимо выполнить действия, описанные ниже.

- В ПО *Machine Expert-Basic* перейти во вкладку *Configuration*, затем выбрать подраздел *SL1 (Serial line)*. Необходимо выбрать протокол *Modbus Serial IOScanner* (рис. 8.3).

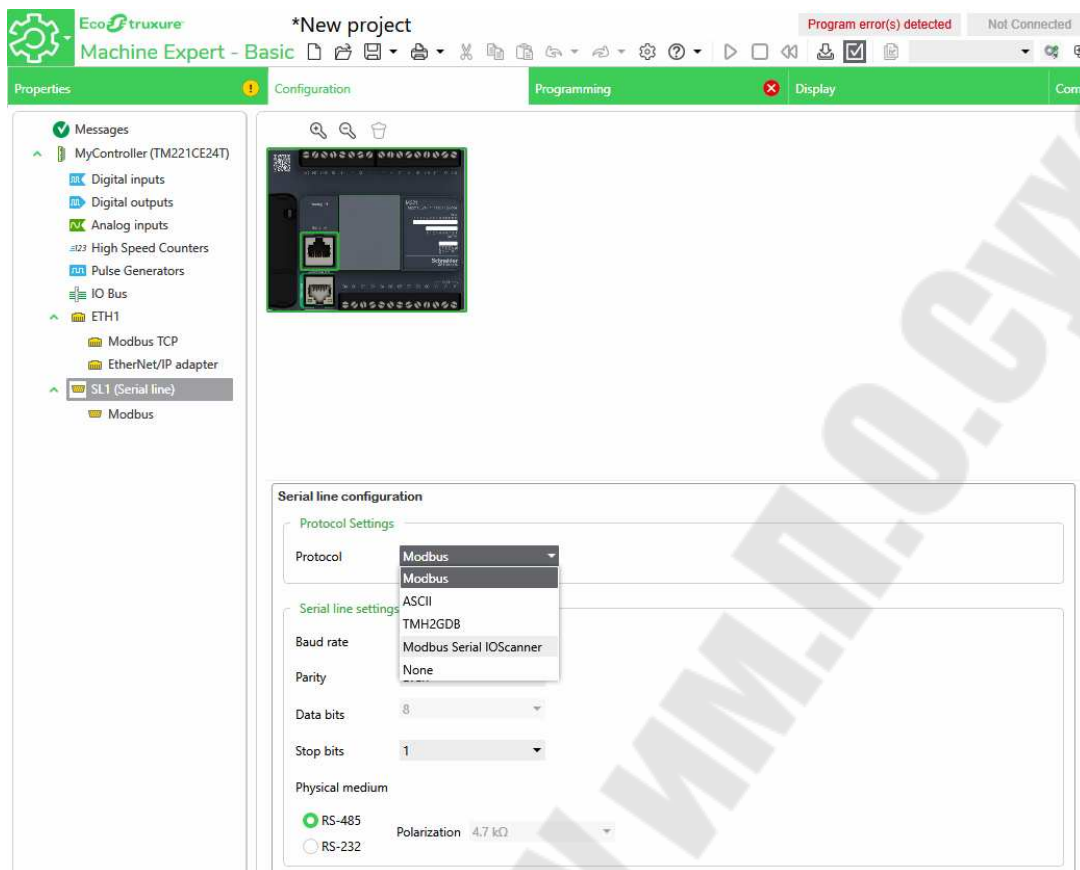


Рис. 8.3. Выбор протокола

• В подразделе *SL1 (Serial line)* необходимо указать параметры соединения (рис. 8.4), которые должны совпадать с параметрами ПЧ (см. рис. 8.2).

Далее необходимо перейти в *Modbus Serial IOScanner*, в появившемся окне нажать *Drive* и выбирать необходимый ПЧ, например, *ATV320*. После выбора нажать кнопку *Add*, а затем *Apply* (рис. 8.5).

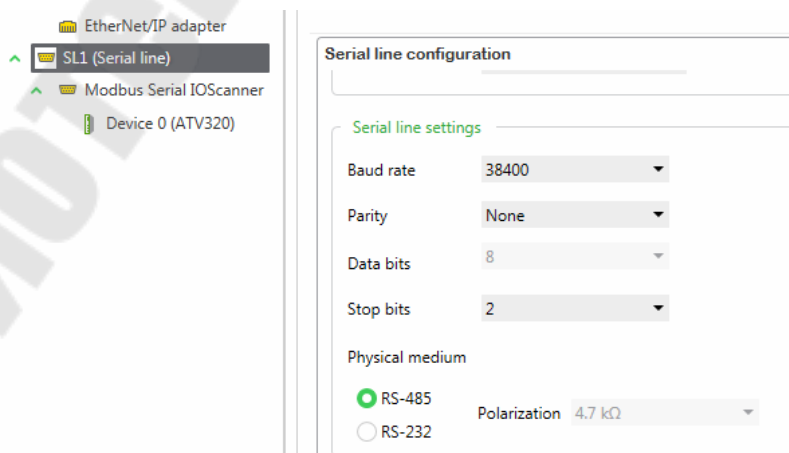


Рис. 8.4. Указание параметров соединения



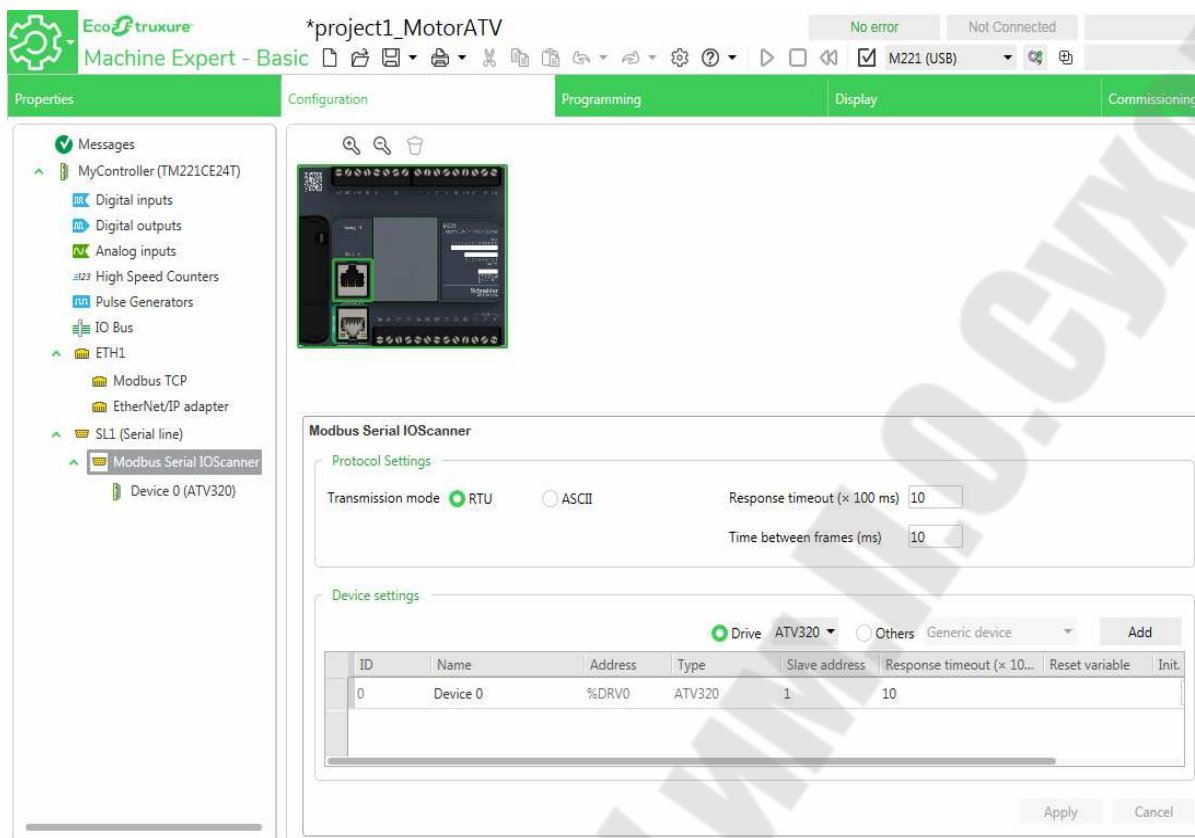


Рис. 8.5. Добавление параметров

Для задания фиксированных скоростей вращения двигателя с помощью преобразователя частоты необходимо сконфигурировать следующие параметры:

COнF->FULL->FUн- ->PSS- ->

PS2 = LI3 – 2 скорости;

PS4 = LI4 – 4 скорости;

PS8 = LI5 – 8 скоростей;

(PS16 = LI6 – 16 скоростей;)

SP2...SP16 = 0...599 (Гц) – заданные скорости со 2-й по 16;

JPF = 27.8 (Гц) – пропускаемая частота 1;

JF2 = 0.0 (Гц) – пропускаемая частота 2;

JF3 = 0.0 (Гц) – пропускаемая частота 3;

JFH = 5.0 (Гц) – девиация частот.

Приведенные выше настройки также можно выполнить в программе *SoMove* (рис. 8.6).

Code	Long Label	Conf0	Default Value	Min Value
▼ Application function				
▶ Ref Freq switch				
▶ Ref. operations				
▶ Ramp switching				
▶ Stop configuration				
▶ Auto DC injection				
▶ Jog				
▼ Preset speeds				
PS2	2 Preset Freq assignment	LI3	NO	
PS4	4 Preset Freq assignment	LI4	NO	
PS8	8 Preset Freq assignment	LI5	NO	
PS16	16 Preset Freq assignment	NO	NO	
SP2	Preset speed 2	10 Hz	10 Hz	0 Hz
SP3	Preset speed 3	15 Hz	15 Hz	0 Hz
SP4	Preset speed 4	20 Hz	20 Hz	0 Hz
SP5	Preset speed 5	25 Hz	25 Hz	0 Hz
SP6	Preset speed 6	30 Hz	30 Hz	0 Hz
SP7	Preset speed 7	35 Hz	35 Hz	0 Hz
SP8	Preset speed 8	40 Hz	40 Hz	0 Hz
SP9	Preset speed 9	45 Hz	45 Hz	0 Hz
SP10	Preset speed 10	50 Hz	50 Hz	0 Hz
SP11	Preset speed 11	55 Hz	55 Hz	0 Hz
SP12	Preset speed 12	60 Hz	60 Hz	0 Hz
SP13	Preset speed 13	70 Hz	70 Hz	0 Hz
SP14	Preset speed 14	80 Hz	80 Hz	0 Hz
SP15	Preset speed 15	90 Hz	90 Hz	0 Hz
SP16	Preset speed 16	100 Hz	100 Hz	0 Hz
JPF	Skip frequency	27.8 Hz	0 Hz	0 Hz
JF2	Skip frequency 2	0 Hz	0 Hz	0 Hz
JF3	3rd Skip frequency	0 Hz	0 Hz	0 Hz
JFH	Skip Freq. hysteresis	5 Hz	1 Hz	0.1 Hz
▶ +/- speed				
▶ +/- speed around ref				

Рис. 8.6. Выполнение настроек

Для управления от дискретных входов преобразователя частоты необходимо выполнить следующие настройки:

COnt->FULL->CtL- ->

Frl = PI – дискретные входы;

если Frl = All заданная скорость 1 определяется аналоговым входом 1.

Можно также воспользоваться программой *SoMove* (рис. 8.7).

Code	Long Label	Conf0	Default Value	Min Value
LAC	Access level	Standard access	Standard access	
▶ Simply start				
▶ Settings				
▶ Motor control				
▶ Inputs / Outputs				
▼ Command				
FR1	Configuration reference frequency 1	AI1 Analog input	AI1 Analog input	
RIN	Reverse direction disable	AI1 Analog input	^	
PST	Stop key enable	AI2 Analog input	es	
CHCF	Control mode configuration	AI3 Analog input	combined channel mode	
CCS	Command switching	Reference frequency via remote terminal	D1	
CD1	Command channel 1 assign	Reference frequency via Modbus	terminal block	
CD2	Command channel 2 assign	Reference frequency via CANopen		
RFC	Freq switching assignment	Modbus communication	Modbus communication	
FR2	Configuration reference frequency 2	FR1	FR1	
COP	Copy Ch.1-Ch.2	Not configured	Not configured	
FN1	F1 key assignment	No copy	No copy	
FN2	F2 key assignment	NO	NO	
FN3	F3 key assignment	NO	NO	
FN4	F4 key assignment	NO	NO	
BMP	HMI command	Cmd/ref clear on c/over	Cmd/ref clear on c/over	
▶ Function Blocks				
▶ Application function				
▶ Monitoring				
▶ Communication				

Рис. 8.7. Настройка параметров в программе SoMove

Для включения/отключения функциональных блоков (ФБ) необходимо изменить следующий параметр:

COнF->FULL->FbП- ->

FbrП = nO – запуск ФБ вручную.

Также можно воспользоваться возможностями программы SoMove (рис. 8.8).

Code	Long Label	Conf0	Default Value	Min Value
LAC	Access level	Standard access	Standard access	
▶ Simply start				
▶ Settings				
▶ Motor control				
▶ Inputs / Outputs				
▶ Command				
▼ Function Blocks				
FBCD	FunctionBlock activation	Stop	Stop	
FBRM	FunctionBlock start mode	NO	NO	
FBSM	Motor stop type on FunctionBlock stop	NO	Freewheel stop	
FBDf	FunctionBlock response to drive error	YES	Stop	
▶ Monit. Fun. Blocks				
▶ Input Assignments				
▶ ADL Containers				
▶ FB Parameters				

Рис. 8.8. Изменение параметров в программе SoMove

## 9. Настройка функции *ATV Logic*

Преобразователь частоты *Altivar ATV320* содержит простой программируемый логический контролер *ATV Logic*, функции которого можно запрограммировать в программе *SoMove*.

Для этого создадим входные и выходные переменные (рис. 9.1, 9.2), укажем их тип и привяжем их к физическим переменным преобразователя:

- входы *I1*, *I2* – дискретные, связаны с дискретными входами *DI4* и *DI5* ПЧ соответственно;
- выход *O1* – дискретный, связан в релейным выходом *R2* ПЧ.

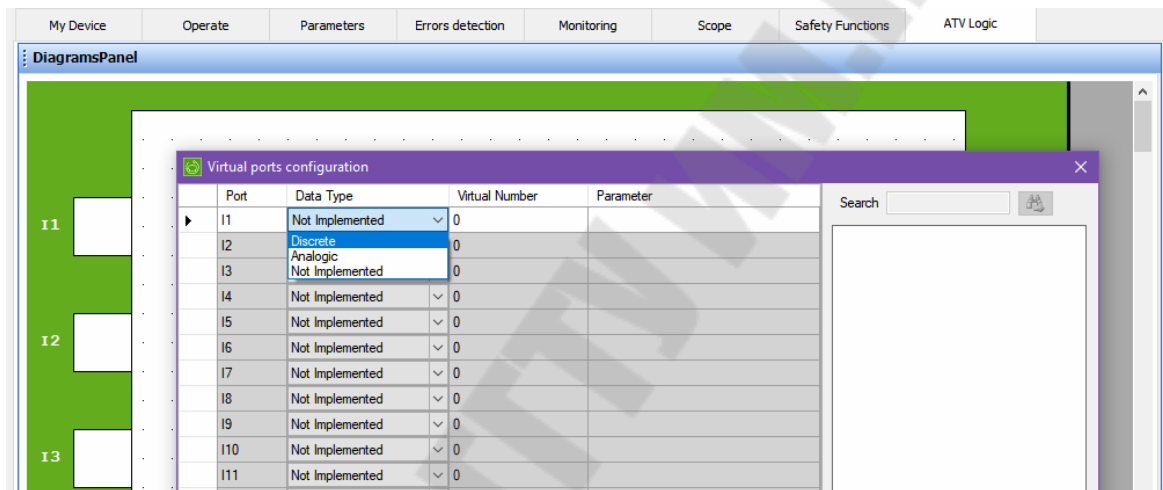


Рис. 9.1. Создание входных параметров

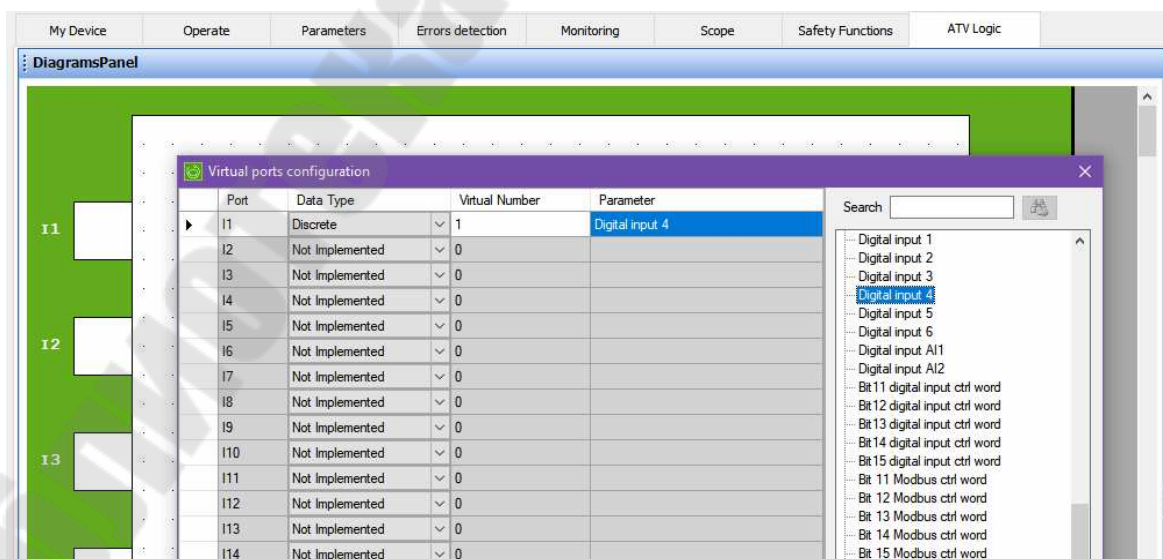


Рис. 9.2. Создание выходных переменных

Создадим логическую функцию «И» между входами и выходом и компилируем проект (рис. 9.3).

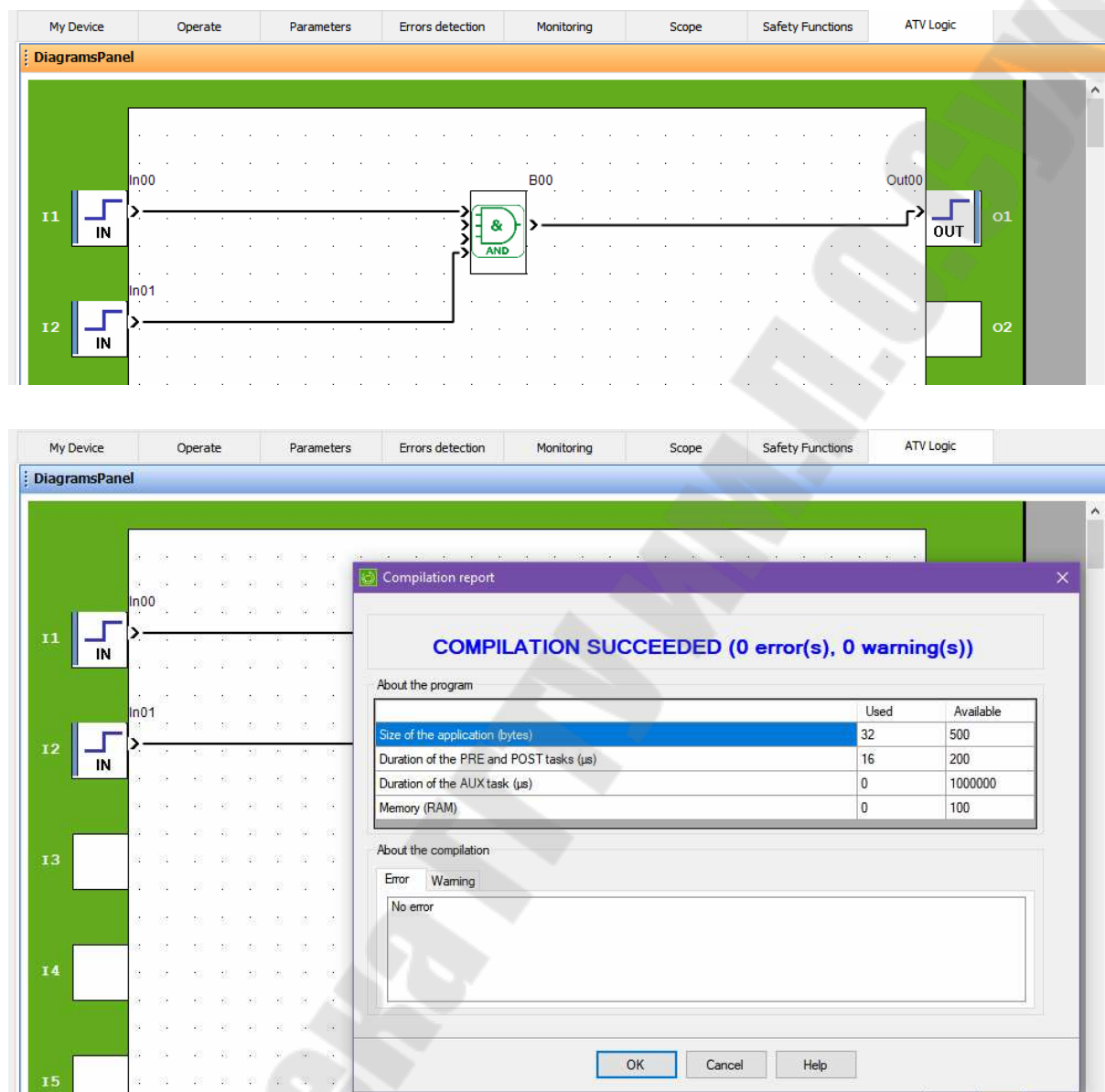


Рис. 9.3. Создание логической функции

Code	Long Label	Conf0	Default Value	Min Value
LAC	Access level	Standard access	Standard access	
▶ Simply start				
▶ Settings				
▶ Motor control				
▶ Inputs / Outputs				
▶ Command				
▼ Function Blocks				
FBCD	FunctionBlock activation	Stop	Stop	
FBRM	FunctionBlock start mode	Li6	NO	
FBSM	Motor stop type on FunctionBlock stop	Ignore	Freewheel stop	
FBDF	FunctionBlock response to drive error	Ignore	Stop	
▶ Monit. Fun. Blocks				
▶ Input Assignments				
▶ ADL Containers				
▶ FB Parameters				
▶ Application function				
▶ Monitoring				
▶ Communication				

Рис. 9.4. Назначение события с указанием реакций

Назначим событие для запуска *ATV Logic*, например, дискретный вход *Li3* ПЧ, также указываем *Ignore* в качестве реакции мотора на остановку *ATV Logic*, и в качестве реакции *ATV Logic* на остановку мотора (рис. 9.4).

## 10. Методика проведения лабораторных работ

### 10.1. Лабораторный стенд

Внешний вид лабораторного стенда приведен на рис. 10.1.

Принципиальная электрическая схема стенда приведена на рис. 10.2.

Чтобы обеспечить работоспособность стенда, необходимо подключить его к сети переменного напряжения 380 В с помощью вилки *XP1*. Питание компонентов сети постоянного тока напряжением 24 В осуществляется с помощью блока питания *A1*, подключенного к сети 220 В через автоматический выключатель *Qf1*.

Силовая часть стенда представляет собой асинхронный электродвигатель *M1*, обмотка статора которого соединена по схеме «звезда» и подключена к силовым клеммам *U, V, W* преобразователя частоты *UZ1*. В свою очередь подача трехфазного напряжения на входные клеммы *L1, L2, L3* преобразователя частоты *UZ1* осуществляется включением автоматического выключателя *QF1* и пускателя *KM1*.

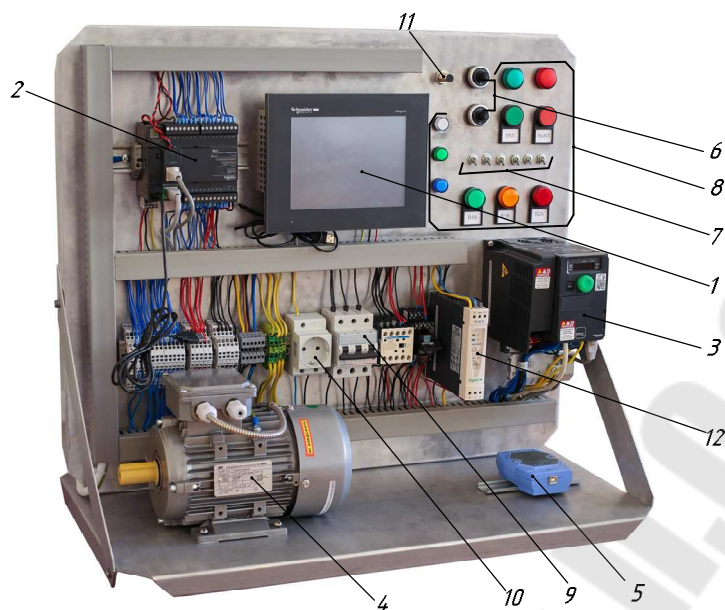


Рис. 10.1. Внешний вид лабораторного стенда:  
 панель оператора – 1; программируемый логический контроллер (ПЛК) – 2;  
 преобразователь частоты (ПЧ) – 3; электродвигатель – 4; преобразователь  
 интерфейса – 5; блок аналогового задания – 6; блок дискретного задания – 7;  
 элементы индикации – 8; автоматический выключатель – 9; розетка – 10;  
 индуктивный датчик – 11; блок питания – 12

Для замыкания силовых контактов пускателя  $KM1$  на катушку  $KM1$  нажатием кнопки  $SB1$  «ВКЛ» необходимо подать напряжение питания 24 В. При этом нормально разомкнутый контакт  $KM1$  осуществляет питание напряжением 24 В катушки реле  $K1$ , а нормально разомкнутые контакты реле  $K1$  замыкаются, включая индикатор  $HL1$  кнопки  $SB1$  и шунтируя ее. Размыкается также нормально замкнутый контакт реле  $K1$ , отключая индикатор  $HL2$  кнопки  $SB2$ . Отключение ПЧ и ЭД от сети переменного тока происходит нажатием кнопки  $SB2$  «ВЫКЛ». Для гашения ЭДС самоиндукции при отключении пускателя  $KM1$  и реле  $K1$  параллельно к катушкам подключены диоды  $VD2$  и  $VD3$  соответственно.

К входам  $I8...I13$  ПЛК подключены тумблеры  $SA1...SA6$ , соответственно. На вход  $I2$  ПЛК подключен индуктивный датчик  $S11$ . На вход  $I3$  ПЛК подключен сигнал с дискретного выхода  $DQ$  преобразователя  $UZ1$ . К выходам  $Q1, Q2, Q3$  ПЛК подключены соответственно, синяя  $HL6$ , зеленая  $HL7$  и белая  $HL8$  индикаторные лампы. К выходам  $Q4...Q9$  ПЛК подключены клеммы управления  $D11...D16$  преобразователя частоты  $UZ1$ .

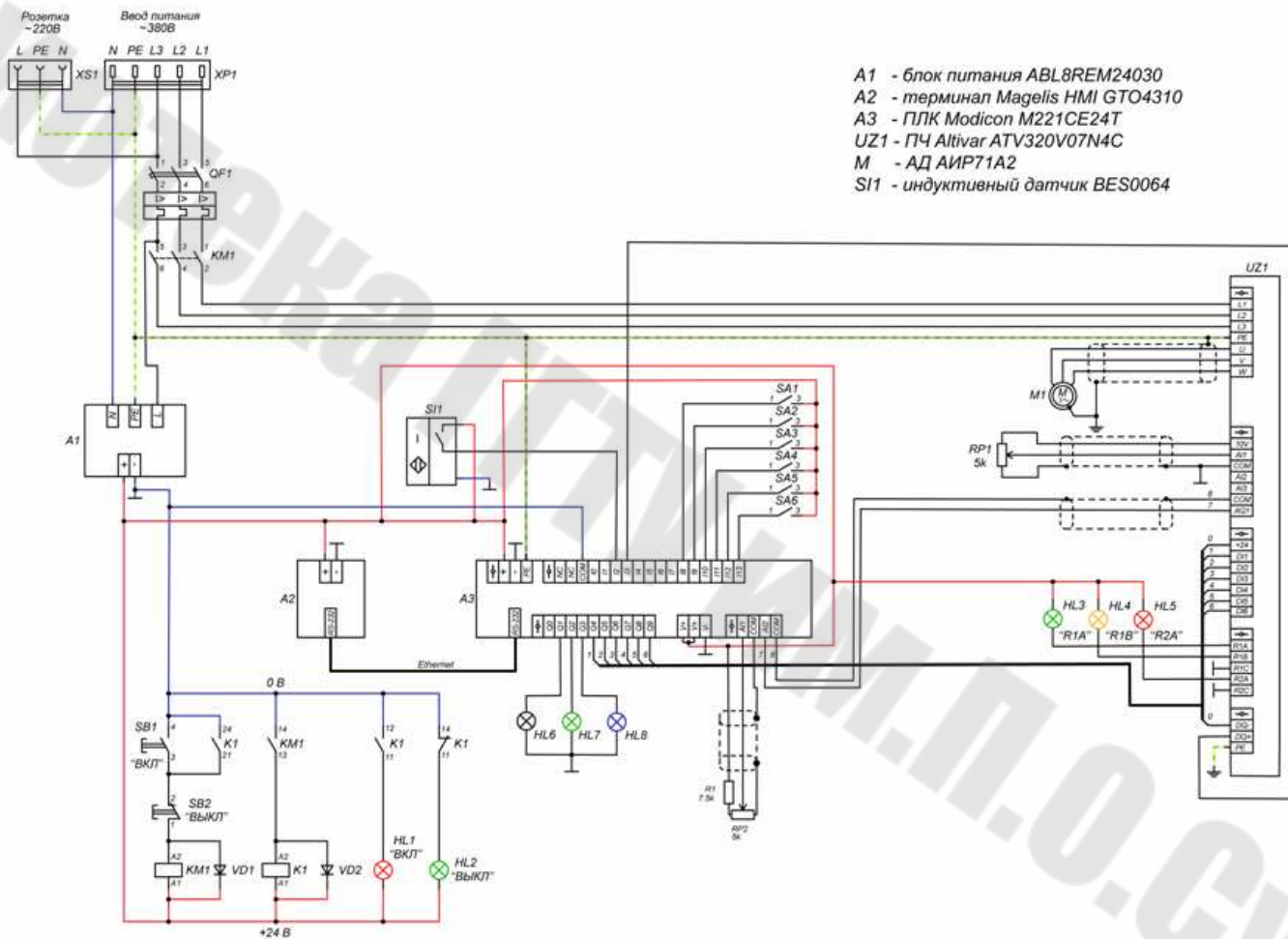


Рис. 10.2. Принципиальная электрическая схема стенда



К аналоговым входам *AI1* и *AI2* ПЛК подключены соответственно потенциометр *RP2* и аналоговый выход *AQ1* преобразователя *UZ1*.

К релейным выходам *RIA*, *R1B* и *R2A* преобразователя частоты *UZ1* подключены соответственно зеленый *HL3*, желтый *HL4* и красный *HL5* индикаторы. На вход *AI1* аналогового задания частоты преобразователя *UZ1* подключен потенциометр *RP1*.

Функции и логика работы входов и выходов задаются управляющей программой ПЛК и ПТ, а также настройками ПЧ.

### **10.2. Рекомендуемая последовательность выполнения работ**

1. Необходимо ознакомиться с устройством лабораторного стенда и особенностями применяемого оборудования. Изучить электрическую схему стенда.

2. С помощью программного обеспечения *Machine Expert – Basic* написать программу для ПЛК *Modicon M221* в соответствии с заданием. Проверить работоспособность программы в режиме симуляции и продемонстрировать ее преподавателю.

3. С помощью программного обеспечения *Vijeo Designer* написать программу для терминала *Magelis HMI GTO* в соответствии с заданием. Проверить работоспособность программы в режиме симуляции и продемонстрировать ее преподавателю.

4. При необходимости (в соответствии с заданием) произвести настройку режима работы преобразователя частоты *Altivar ATV320*.

5. Загрузить написанные программы в контроллер и терминал и продемонстрировать выполнение задания.

6. Составить отчет, содержащий, название, цель работы, индивидуальное задание, выдаваемое преподавателем, программы для контроллера и терминала, пояснения по логике написания программ, анализ результатов работы, выводы по работе.

### **10.3. Примерный перечень заданий для лабораторных работ**

**Задание 1.** Ознакомьтесь с устройством ПЛК семейства *Schneider Modicon M221* и средой программирования *Machine Expert – Basic*. Создайте простейшую программу, позволяющую включать индикаторные лампы *HL6*, *HL7*, *HL8* (см. рис. 10.2) при изменении положения переключателей *SA1*, *SA2*, *SA3* соответственно. Запишите программу в ПЛК и убедитесь в ее работоспособности.

**Задание 2.** Ознакомьтесь с устройством ПТ *Magelis HMI GTO* и средой программирования *Vijeo Designer*. Создайте проект, позво-

ляющий связать ПТ и ПЛК по протоколам *Modbus TCP* и *Modbus RTU*. С помощью виртуального кнопочного переключателя на экране ПТ запустите процесс поочередного мигания ламп *HL6*, *HL7* (см. рис. 10.2), подключенных к ПЛК с одновременным их дублированием виртуальными лампами на экране ПТ. Загрузите части проекта в ПТ и ПЛК и убедитесь в его работоспособности.

**Задание 3.** Создайте проект на основе системы прерываний ПЛК, в котором при срабатывании индуктивного датчика *S11* (см. рис. 10.2) на экране ПТ будет появляться всплывающее окно с предупреждением о срабатывании защиты, и одновременно с этим начнут попеременно мигать лампы *HL6*, *HL7*. Данный режим должен отключаться через виртуальную кнопку сброса на экране ПТ. Загрузите части проекта в ПТ и ПЛК, убедитесь в его работоспособности.

**Задание 4.** Установите связь контроллера *Modicon M221* с преобразователем частоты *Altivar ATV320* по протоколу *SR-485*. Создайте проект, в котором управление преобразователем частоты будет выполняться от контроллера посредством функциональных блоков программной среды *Machine Expert – Basic*. В качестве органов управления используйте переключатели *SA1...SA6* (см. рис. 10.2), а в качестве элементов индикации – лампы *HL6...HL8*. Загрузите проект в ПЛК и убедитесь в возможности пуска и останова двигателя *M1*, подключенного к ПЧ.

**Задание 5.** Измените проект, созданный по заданию 3, таким образом, чтобы всякий раз при срабатывании индуктивного датчика *S11* (см. рис.10.2), на экране терминала появлялся журнал тревог с соответствующими записями о срабатываниях датчика.

**Задание 6.** Создайте проект на основе контроллера *Modicon M221* и терминала *Magelis HMI GTO*, в котором все системные события (включение/выключение контроллера, наличие связи между ПЛК и ПТ, изменение вводимых параметров) заносятся бы в журнал событий. Загрузите части проекта в ПТ и ПЛК, и убедитесь в его работоспособности.

**Задание 7.** Создайте проект, связывающий терминал *Magelis HMI GTO*, контроллер *Modicon M221* и преобразователь частоты *Altivar ATV320* в единую систему управления двигателем *M1* (см. рис. 10.2). Создайте на экране терминала виртуальные органы управления и элементы индикации, позволяющие производить пуск, останов двигателя *M1*, а также производить выбор одной из 8 фиксированных скоростей вращения. Текущий режим работы системы должен отображаться на терминале. При срабатывании индуктивного

датчика двигатель должен останавливаться, а на экран терминала выводится сообщение об аварийном режиме.

**Задание 8.** Измените проект, созданный по заданию 7, таким образом, чтобы изменение скоростей вращения двигателя *M1* происходило по циклу автоматически. Параметры цикла (число участков, частоты вращения и направления вращения на каждом участке, продолжительность работы на каждом участке) должны задаваться через экранное меню терминала. Настройте преобразователь частоты, загрузите части проекта в ПТ и ПЛК и убедитесь в его работоспособности.

**Задание 9.** Задайте циклический режим работы двигателя *M1* путем программирования контроллера *ATV Logic*, встроенного в преобразователь частоты *Altivar ATV320*. В качестве органов управления используйте переключатели *SA1...SA6* (см. рис. 10.2), а в качестве элементов индикации – лампы *HL3...HL5*. Убедитесь в работоспособности программы.

#### **10.4. Контрольные вопросы**

1. Что такое программируемый логический контроллер и в чем его преимущество перед релейно-контакторными схемами?
2. Опишите процесс создания проекта в среде программирования контроллеров *Machine Expert – Basic*.
3. Опишите процесс создания проекта в среде программирования терминалов *Vijeo Designer*.
4. Как в контроллере *Modicon M221* установить связь с преобразователем частоты *Altivar ATV320* по протоколу *RS-485*.
5. Опишите процесс создания прерываний в контроллере *Modicon M221*.
6. Как связать программируемый терминал *Magelis HMIGTO* и контроллер *Modicon M221* по протоколам *Modbus TCP* и *Modbus RTU*.
7. Опишите процесс создания программы с помощью программного обеспечения *SoMove* во встроенном контроллере *ATV Logic* преобразователя частоты *Altivar ATV320*.
8. Опишите процесс настройки преобразователя частоты *Altivar ATV320* для работы с контроллером *Modicon M221* по протоколу *Modbus (RS-485)*.
9. Как в среде *Vijeo Designer* создать журнал событий/тревог и выводить туда информацию о различных системных событиях?

## Литература

1. Кангин, В. В. Промышленные контроллеры в системах автоматизации технологических процессов : учеб. пособие для вузов / В. В. Кангин. – Старый Оскол : ТНТ, 2018. – 407 с.

2. Герасимов, А. В. Программируемые логические контроллеры : учеб. пособие / А. В. Герасимов, И. Н. Терюшов, А. С. Титовцев ; Федер. агентство по образованию, Казан. гос. технол. ун-т. – Казань : КНИТУ, 2008. – 169 с.

3. Сергеев, А. И. Программирование контроллеров систем автоматизации: учебное пособие / А. И. Сергеев, А. М. Черноусова, А. С. Русяев ; Оренбург. гос. ун-т. – Оренбург : Оренбург. гос. ун-т, 2017. – 126 с.

4. Третьяков, А. А. Средства автоматизации управления: системы программирования контроллеров / А. А. Третьяков, И. А. Елизаров, В. Н. Назаров ; Тамб. гос. техн. ун-т. – Тамбов : ТГТУ, 2017. – 82 с.

5. Логический контроллер *Modicon M221*. – Режим доступа: <https://www.se.com/ru/ru/product/TM221CE24T/компактный-базовый-блок-m22124io-транзист-источник-ethernet/?range=62128-логический-контроллер-modicon-m221&selected-node-id=12692210272>. – Дата доступа: 11.12.2020.

6. EcoStruxure Machine Expert – Basic. Generic Functions Library Guide. – Режим доступа: <https://www.se.com/ww/en/download/document/EIO0000003289/>. – Дата доступа: 11.12.2020.

7. SoMachine Programming Guide. – Режим доступа: <https://www.se.com/ww/en/download/document/EIO0000000067/>. – Дата доступа: 11.12.2020.

8. Горячий FAQ о ПЛК Schneider Modicon M221. – Режим доступа: <https://plcontroller.ru/post/6371/>. – Дата доступа: 11.12.2021.

## Содержание

Введение .....	3
1. Краткая характеристика аппаратной части .....	4
2. Создание нового проекта .....	7
3. Инициализация прерываний.....	15
4. Создание всплывающих окон.....	18
5. Панель ввода значений .....	26
6. Создание журнала тревог.....	32
7. Отображение на экране ПТ информации о состоянии связи с ПЛК и режиме работы ПЛК.....	34
8. Конфигурирование преобразователя частоты для работы по протоколу Modbus.....	44
9. Настройка функции ATV Logic.....	52
10. Методика проведения лабораторных работ.....	54
10.1. Лабораторный стенд .....	54
10.2. Рекомендуемая последовательность выполнения работ .....	57
10.3. Примерный перечень заданий для лабораторных работ .....	57
10.4. Контрольные вопросы .....	59
Литература.....	60

Учебное электронное издание комбинированного распространения

**Савельев Вадим Алексеевич  
Дорощенко Игорь Васильевич**

## **ПРОГРАММИРУЕМЫЕ КОНТРОЛЛЕРЫ И ПРОМЫШЛЕННЫЕ СЕТИ**

**Учебно-методическое пособие  
для студентов специальности 1-53 01 05  
«Автоматизированные электроприводы»  
дневной формы обучения**

*Редактор  
Компьютерная верстка*

О. С. Ковалёва  
И. П. Минина

Свидетельство о гос. регистрации в качестве издателя  
печатных изданий за № 1/273 от 04.04.2014 г.  
пр. Октября, 48, 246746, г. Гомель