

Министерство образования Республики Беларусь

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

Кафедра «Автоматизированный электропривод»

В. А. Савельев

ПРОГРАММИРУЕМЫЕ КОНТРОЛЛЕРЫ И ПРОМЫШЛЕННЫЕ СЕТИ

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

*Рекомендовано учебно-методическим объединением
по образованию в области автоматизации технологических процессов
производств и управления в качестве
учебно-методического пособия
для студентов учреждений высшего образования
по специальности 6-05-0713-04 «Автоматизация
технологических процессов и производств»*

Электронный аналог печатного издания

Гомель 2026

УДК 681.5:62-503.55(075.8)
ББК 32.971.322.5я73
С12

Рецензенты: зав. каф. «Промышленная электроника» Гомельского государственного
технического университета имени П. О. Сухого
канд. техн. наук, доц. *Ю. В. Крышнев*;
заместитель главного конструктора ОАО «СтанкоГомель»
Ю. Л. Аникейчик

- Савельев, В. А.**
С12 Программируемые контроллеры и промышленные сети : учеб.-метод. пособие /
В. А. Савельев ; М-во образования Респ. Беларусь, Гомел. гос. техн. ун-т им. П. О. Су-
хого. – Гомель : ГГТУ им. П. О. Сухого, 2026. – 108 с. – Систем. требования: PC не ни-
же Intel Celeron 300 МГц ; 2 Gb RAM ; свободное место на HDD 16 Mb ; ALT Lunux 10.1 ;
Adobe Acrobat Reader. – URL: <https://elib.gstu.by>. – Загл. с титул. экрана.
ISBN 978-985-535-572-5.

Пособие структурно разделено на два модуля. Первый модуль посвящен программиро-
ванию контроллеров на языке функциональных блоков на примере ПЛК Mitsubishi Alpha 2. Вто-
рой модуль рассматривает вопросы программирования контроллеров на языке лестничных диа-
грамм на примере ПЛК Schneider Modicon M221, а также методику программного решения задач
промышленной автоматизации за счет применения программируемого терминала Schneider Ma-
gelis GTO4310 и преобразователя частоты Schneider Altivar ATV320.

Для студентов специальности 6-05-0713-04 «Автоматизация технологических процессов
и производств».

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

ISBN 978-985-535-572-5

© Савельев В. А., 2026
© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2026

Оглавление

Введение	4
Модуль 1. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ	6
1. Назначение, характеристики и схемы включения программируемого логического контроллера Mitsubishi Alpha 2	6
2. Среда программирования программируемого логического контроллера Mitsubishi Alpha 2	9
Функциональные блоки группы FANC	13
3. Составление, отладка и запись программ в контроллер Alpha 2 в среде SW0D5-ALVLS-EU	20
4. Примеры простейших программ в среде SW0D5-ALVLS-EU	23
5. Методика проведения лабораторных работ по модулю 1	27
5.1. Лабораторный стенд для исследования программируемых логических контроллеров на базе электропривода постоянного тока с механической передачей «винт-гайка»	27
5.2. Последовательность выполнения лабораторных работ	28
5.3. Задания к лабораторным работам	29
6. Контрольные вопросы по модулю 1	30
Модуль 2. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ЛЕСТНИЧНЫХ ДИАГРАММ	32
7. Краткая характеристика аппаратной части	32
8. Создание нового проекта	35
9. Инициализация прерываний	43
10. Создание всплывающих окон	47
11. Панель ввода значений	55
12. Создание журнала тревог	61
13. Отображение на экране программируемых терминалов информации о состоянии связи с программируемыми логическими контроллерами и режиме работы программируемых логических контроллеров	63
14. Конфигурирование преобразователя частоты для работы по протоколу Modbus	73
15. Управление преобразователем частоты по протоколу Modbus RTU	82
16. Настройка функции ATV Logic	88
17. Пример создания приложения	92
18. Методика проведения лабораторных работ по модулю 2	100
18.1. Описание лабораторного стенда	100
18.2. Рекомендуемая последовательность выполнения работ	102
18.3. Примерный перечень заданий для лабораторных работ	103
19. Контрольные вопросы по модулю 2	106
Литература	107

Введение

В современном производстве программируемые логические контроллеры (ПЛК) играют ключевую роль в автоматизации технологических процессов различной сложности. Эти устройства пришли на смену традиционным релейно-контактным схемам, которые обладали значительными недостатками, такими как громоздкость и отсутствие оперативности в изменении структуры управления. Программируемые логические контроллеры, являясь компактными и перепрограммируемыми, обеспечивают гибкость и эффективность управления технологическим оборудованием.

Среди множества производителей, предлагающих решения для автоматизации, следует уделить внимание контроллерам производства Mitsubishi и Schneider Electric. Контроллеры Mitsubishi отличаются высокой надежностью и простотой в использовании, что делает их популярными в малых и средних производственных предприятиях. Они предлагают широкий спектр функций и возможностей, включая интеграцию с различными датчиками и исполнительными механизмами. Удобный интерфейс программирования и возможность расширения системы позволяют инженерам быстро адаптировать контроллеры под конкретные задачи, что является важным преимуществом в условиях динамично меняющегося производства.

Контроллеры Schneider, в свою очередь, представляют собой более мощные решения, предназначенные для автоматизации сложных производственных процессов. Они предлагают широкий выбор моделей с различными функциональными возможностями, что позволяет выбрать оптимальный вариант для специфических задач. Schneider Modicon известны своей высокой производительностью и надежностью, а также поддержкой современных промышленных протоколов связи, что делает их идеальными для интеграции в сложные системы автоматизации.

Важно отметить, что наряду с ПЛК, в сфере промышленной автоматизации активно применяются преобразователи частоты (ПЧ) и программируемые терминалы (ПТ). Эти устройства, работающие в сочетании с контроллерами, образуют многофункциональные системы управления, позволяющие оптимизировать затраты электроэнергии и обеспечивать эффективный контроль и управление в режиме реального времени.

В условиях стремительного развития технологий автоматизации инженерам-электрикам необходимо владеть знаниями о структурно-

функциональной организации и технических параметрах ПЛК, ПЧ и ПТ. Настоящее учебно-методическое пособие направлено на изучение основных приемов работы с аппаратно-программным комплексом оборудования, включая контроллеры Mitsubishi Alpha 2 и Schneider Modicon. Оно призвано помочь студентам получить практические навыки, необходимые для проектирования, наладки и эксплуатации автоматизированных систем управления технологическими процессами.

Книга структурно разделена на два модуля. Первый модуль посвящен программированию контроллеров на языке функциональных блоков на примере ПЛК Mitsubishi Alpha 2. Второй модуль рассматривает вопросы программирования контроллеров на языке лестничных диаграмм на примере ПЛК Schneider Modicon M221. Также во втором модуле рассматривается методика программного решения задач промышленной автоматизации за счет применения программируемого терминала Schneider Magelis GTO4310 и преобразователя частоты Schneider Altivar ATV320.

Оба модуля содержат необходимые теоретические сведения для программного решения задач автоматизации, а также описание лабораторных стендов и методики проведения лабораторных практикумов.

Изучение материала в рамках данного учебно-методического пособия позволит студентам не только освоить теоретические основы автоматизации, но и развить практические навыки, необходимые для эффективной работы в этой области. Наличие современных средств автоматизации в учебном процессе способствует подготовке высококвалифицированных специалистов, способных успешно решать задачи автоматизации на уровне современных требований и стандартов.

Модуль 1. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ

1. Назначение, характеристики и схемы включения программируемого логического контроллера Mitsubishi Alpha 2

Основные элементы контроллера Alpha 2 компании Mitsubishi Electric (рис. 1.1): 1 – монтажное отверстие; 2 – контактные клеммы подключения питания; 3 – контактные клеммы подключения входных цепей; 4 – монтажный винт для подключения модуля расширения; 5 – корпус модуля расширения; 6 – канавка для соединения с DIN рейкой; 7 – монтажный замок для установки на DIN рейке; 8 – выходные контактные клеммы; 9 – операционные клавиши; 10 – дисплей; 11 – крышка порта связи для программирования.

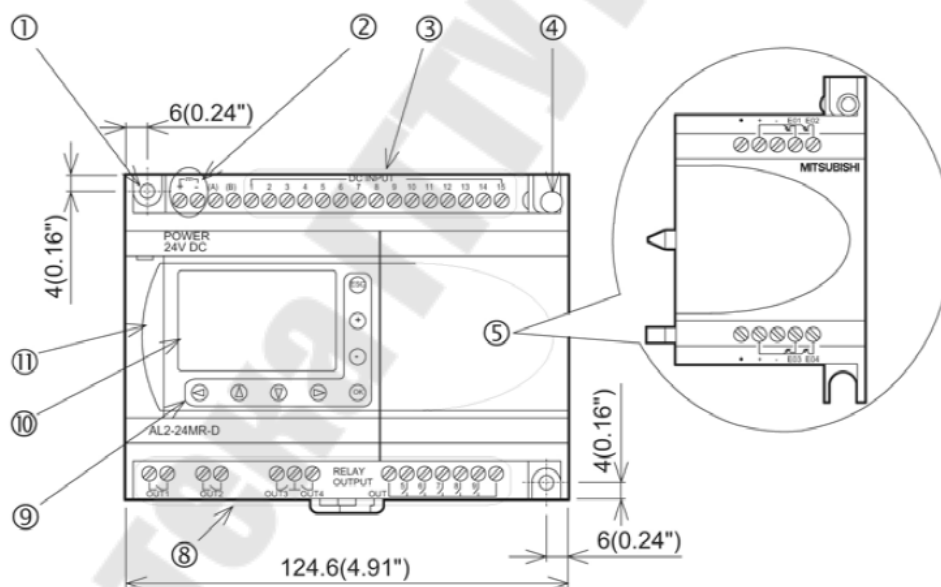


Рис. 1.1. Внешний вид и основные элементы контроллера Alpha 2

В табл. 1.1 приведены основные технические характеристики основного блока контроллера Alpha 2.

При электромонтаже необходимо соблюдать напряжение питания и полярность питания (при питании контроллера 24 VDC) или правильно подключать фазу и нулевой провод (при питании контроллера 100–240 VAC), где *L* – фаза, *N*– нулевой провод.

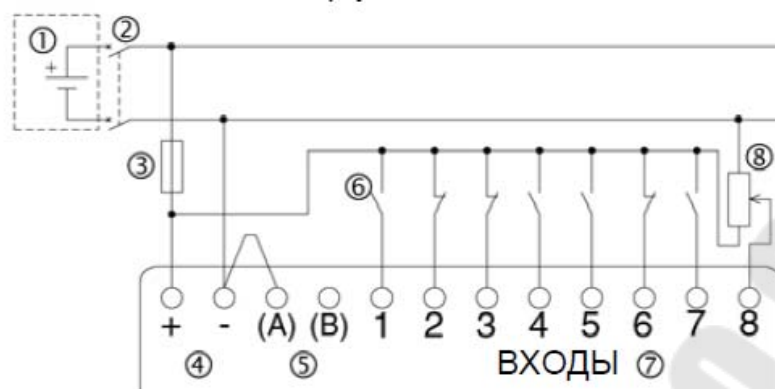
Также необходимо правильно подключать входные и выходные сигналы. На рис. 1.2 показано как правильно подключать питание и входы контроллера при источнике постоянного напряжения – 24 VDC. Здесь обозначено: 1 – источник питания (+24 VDC); 2 – устройство отсоединения цепи питания; 3 – устройство защиты схемы (ограничение до 1А); 4 – контакты подключения питания постоянного тока; 5 – входные контактные клеммы для выбора типа подключения (с общим стоком или истоком); 6 – входные цепи управления (датчики, выключатели); 7 – входные контактные клеммы; 8 – аналоговый вход.

Таблица 1.1

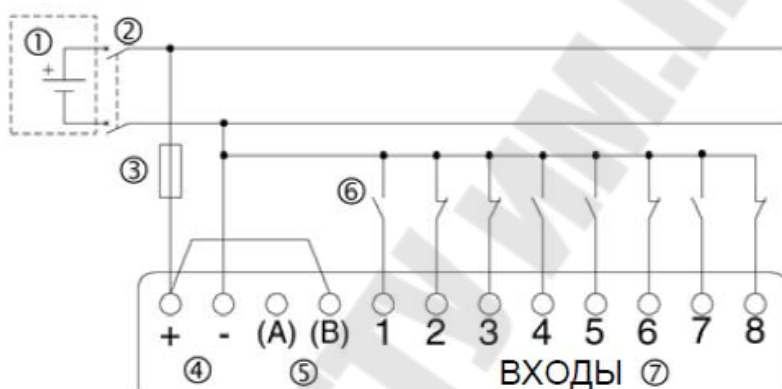
**Основные технические характеристики основного блока
контроллера Alpha 2**

Параметры			AL2-14MR-D
Встроенный ввод-вывод			14
Цифровые входы (24 VDC или 220 VAC)			8
Аналоговые входы (0–10 В)			–
Встроенные выходы (24–220 В)			6
Максимальная потребляемая мощность, Вт			7,5
Типовая потребляемая мощность	Все входы и выходы ВКЛ./ВЫКЛ.	Вт	4/1
Масса, кг			0,3
Размеры (Д × Ш × В), мм			124,6 × 90 × 52

На рис. 1.3 показана схема электрических соединений выходных реле главного блока на переменном и/или постоянном токе. Здесь обозначено: 1 – главный блок контроллера серии «Alpha 2»; 2 – взаимно-исключающие выходы; 3 – выходные устройства; 4 – устройства защиты схемы; 5 – аварийный выключатель; 6 – источник питания постоянного тока; 7 – источник питания переменного тока.



а)



б)

Рис. 1.2. Схемы подключения питания постоянного тока (24 VDC) и входных цепей:

а – с общим «+» (исток); б – с общим «-» (сток)

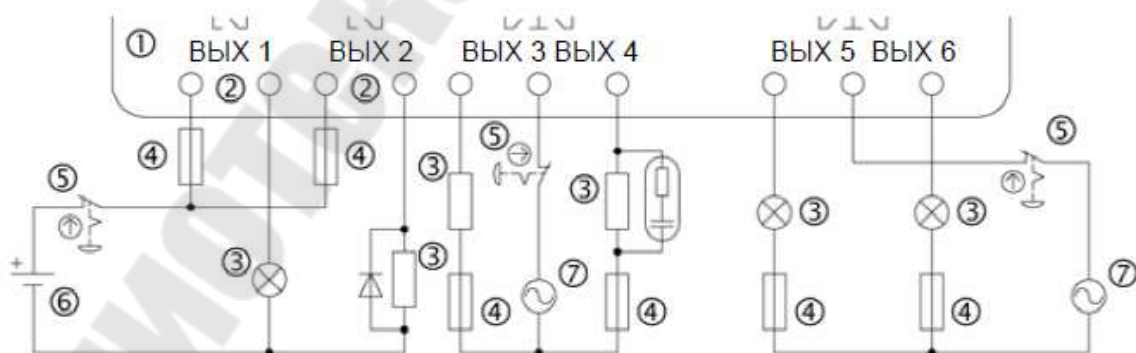


Рис. 1.3. Схема электрических соединений выходных реле главного блока (переменный ток и/или постоянный ток)

2. Среда программирования программируемого логического контроллера Mitsubishi Alpha 2

Контроллеры Alpha 2 программируется при помощи программы SW0D5-ALVLS-EU или при помощи клавиш контроллера. С помощью программы можно также проверять и/или мониторить написанную программу.

Для установки программы SW0D5-ALVLS-EU необходим обычный компьютер (Pentium 133 MHz, RAM 32 MB, Hard Disk 10 MB) с операционной системой Windows 95 или выше. Также необходимо наличие последовательного порта (или USB порта – через преобразователь).

Для программирования контроллера нужно:

Шаг 1. Открыть программу SW0D5-ALVLS-EU:

Пуск/Все программы/Mitsubishi Alpha Controller/Alpha Programming

Шаг 2. Открыть новый проект (рис. 2.1).

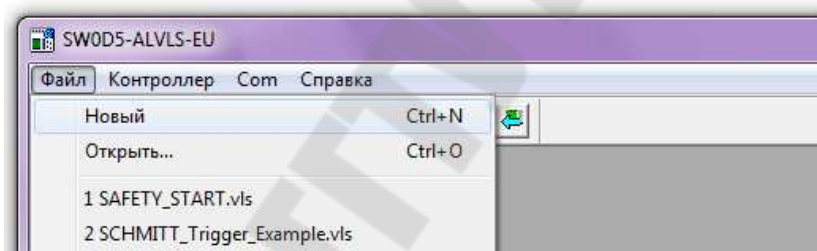


Рис. 2.1. Меню создания нового проекта

Шаг 3. Выбрать модель контроллера (рис. 2.2). Сколько входов/выходов физически существует в контроллере и серию контроллера (Alpha или Alpha 2/Alpha XL). После выбора нажать *OK*.

На рис. 2.3 приведен интерфейс программы SW0D5-ALVLS-EU.

В строке меню находятся опции «Файл», «Правка», «Вид», «Вставить», «Инструментальные средства», «Поиск», «Контроллер», «Com», «Опция», «Окно» и «Справка».

Дальнейшую информацию можно найти в справочной системе пакета SW0D5-ALVLS-EU.

Стандартная панель инструментов содержит кнопки «Новый», «Открыть», «Сохранить», «Вырезать», «Вставить», «Печатать», «О программе», «Контекстная справка», «Изменение размера окна» и «Прочитать информацию контроллера».

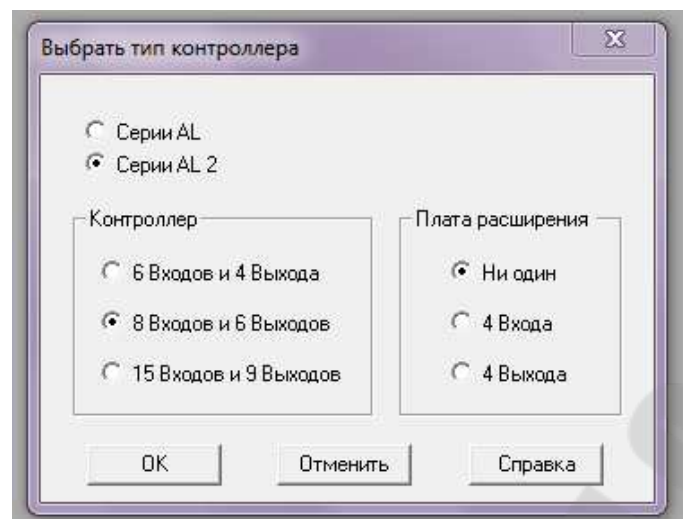


Рис. 2.2. Окно выбора типа ПЛК и числа входов/выходов

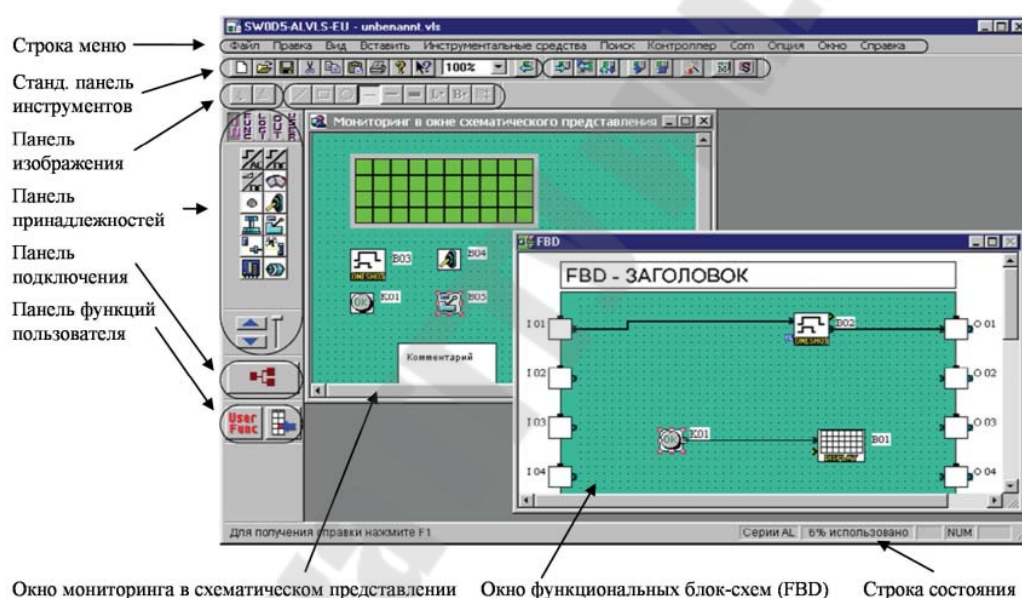


Рис. 2.3. Основные элементы программного интерфейса SW0D5-ALVLS-EU

Панель изображения содержит кнопки «Импортировать» и «Экспортировать» в окне «Мониторинг в окне схематического представления». Кнопка «Импортировать» позволяет вставить предварительно экспортированный файл изображения (*.img) в окно «Мониторинг в окне схематического представления».

Панель инструмента рисования содержит кнопки «Рисование линий», «Рисование прямоугольников», «Рисование овалов», «Тонкая линия», «Средняя линия», «Толстая линия», «Цвет линии», «Цвет кисти» и «Выравнивание объектов».

Инструментальная панель контроллера содержит кнопки «Запись в контроллер», «Сверить данные контроллера», «Диагностика контроллера», «Запуск контроллера», «Остановить контроллер», «Мастер Авто FBD», «Запуск/останов мониторинга» и «Запуск/останов моделирования».

Инструментальная панель подключения представлена единственной кнопкой «Подключение».

Строка состояния содержит текущее состояние «используемой серии», «процентного использования памяти контроллера», «клавиши *Caps Lock*», «клавиши *Num Lock*» и «клавиши *Scroll Lock*».

Существуют четыре основных типа блоков: входы, функциональные блоки, логические блоки и выходы (рис. 2.4). К входам подключаются датчики, к выходам – исполнительные механизмы, а программирование осуществляется при помощи функциональных и логических блоков.

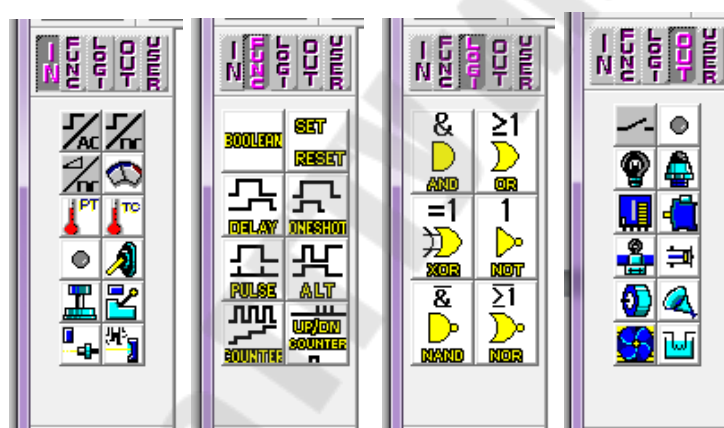
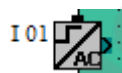


Рис. 2.4. Фрагмент программного интерфейса, отображающий основные типы функциональных блоков

Функциональные блоки группы IN



Двоичный (релейный) вход (Binary Input AC). Вход включен, если на входную клемму приложено переменное напряжение 100–240 VAC (только для контроллеров с питанием 100–240 VAC). Если напряжения нет – на входе 0.



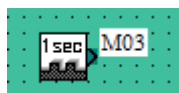
Двоичный (релейный) вход (Binary Input DC). Вход включен, если на входную клемму приложено постоянное напряжение 24 VDC (только для контроллеров с питанием 24 VDC). Если напряжения нет – на входе 0.



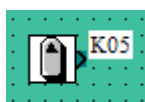
Аналоговый вход (Analog input DC). Вход изменяет свое значение от 0 до 255 в зависимости от напряжения на входе (0–10 VDC) (только для контроллеров с питанием 24 VDC).

Входные блоки ставятся во входные ячейки на рабочем поле программирования.

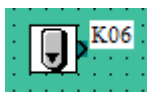
Также к группе входов относятся источник секундных импульсов, 8 клавиш (вверх, вниз, вправо, влево, плюс, минус, ОК и ESC) для ввода и просмотра параметров. Эти элементы не ставятся в стандартное поле входов, – они ставятся непосредственно на поле программирования.



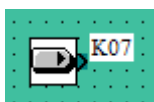
Источник секундных импульсов (*1 second pulse*). Блок выдает каждую секунду один импульс.



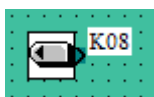
Стрелка вверх (*Up arrow*). На выходе блока появляется сигнал при нажатии на стрелку вверх на клавиатуре контроллера.



Стрелка вниз (*Down arrow*). На выходе блока появляется сигнал при нажатии на стрелку вниз на клавиатуре контроллера.



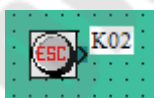
Стрелка вправо (*Right arrow*). На выходе блока появляется сигнал при нажатии на стрелку вправо на клавиатуре контроллера.



Стрелка влево (*Left arrow*). На выходе блока появляется сигнал при нажатии на стрелку влево на клавиатуре контроллера.



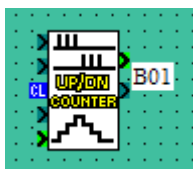
Клавиша *OK*. На выходе блока появляется сигнал при нажатии на клавишу *OK* на клавиатуре контроллера.



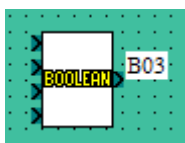
Клавиша *ESC*. На выходе блока появляется сигнал при нажатии на клавишу *ESC* на клавиатуре контроллера.

Функциональные блоки группы FANC

Название блока пишется внутри блока. Структура и принципы работы с блоками одинаковые. Ниже, как пример, описан блок UP/DN COUNTER – реверсивный счетчик импульсов. Нажимая два раза на блок, открывается окно настройки параметров.

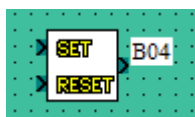


Темно-зеленые клеммы – двоичные сигналы, т. е. которые могут быть либо включены, либо выключены. Светло-зеленые клеммы могут принимать различные значения от -32768 до 32767 , т. е. переменные величины. Они используются для обработки физических величин (температура, давление и т. п.), в счетчиках, элементах сравнения (компараторах), для вывода на экран и т. д.



Блок булевой алгебры (Boolean). Блок позволяет управлять бинарным выходом, в зависимости от логической формулы, записанной в него. Логическая формула – формула булевой алгебры, с использованием элементов логики: И, ИЛИ, НЕ, И-НЕ, ИЛИ-НЕ.

Перед использованием этого блока Boolean желательно посмотреть логические элементы. Двойным щелчком открывается окно ввода параметров. Ниже, на рис. 2.5, приведен пример. Выход O 01 (O – Output) включен, если верно записанное в блок логическое условие: I 01 (I – Input) and (I 02 or I 03). То есть одновременно включены вход 1 и один из входов 2 или 3.



RS-триггер (Set/Reset). При подаче сигнала (даже кратковременного) на вход S (SET) выход включается. При подаче сигнала на вход R (RESET) выход выключается. При подаче сигналов на оба входа выполняется та функция, которая поставлена в приоритете – включение или выключение.

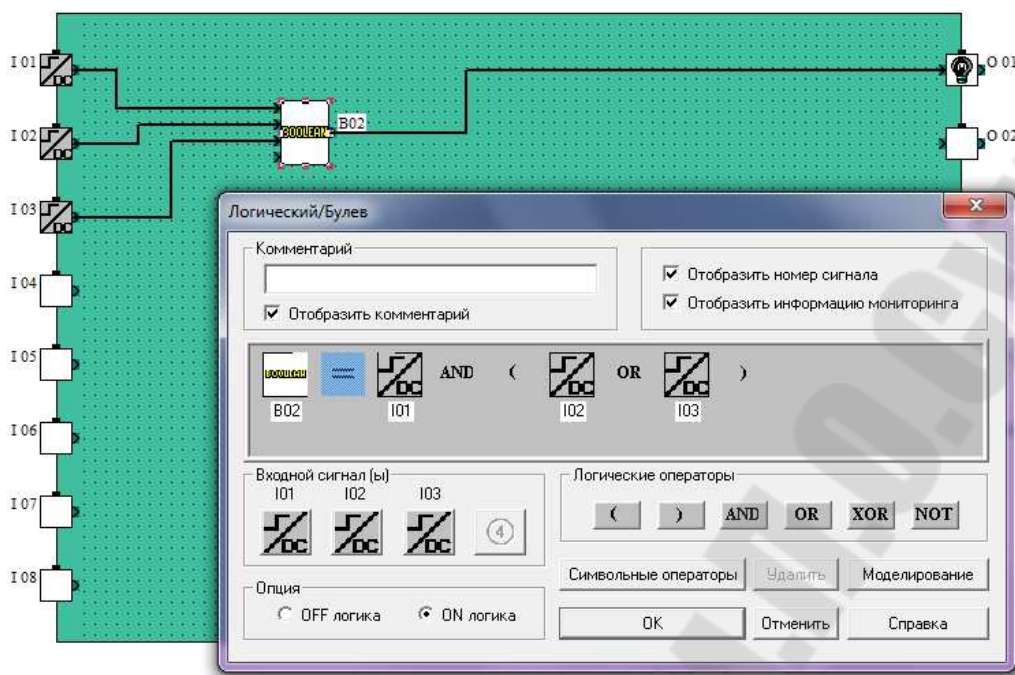
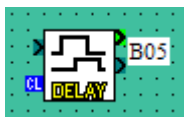
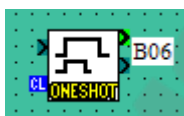


Рис. 2.5. Структура блока Boolean

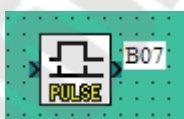
Задержка (Delay). Блок работает с задержкой на включение (ON-Delay), на выключение (OFF-Delay) или на включение и на выключение (ON-Delay, OFF-Delay). При подаче сигнала на вход начинает работать таймер. После установленного времени изменяется выход. Обнуление блока происходит при подаче сигнала на вход CL (CLEAR). Текущее значение блока (светло-зеленая клемма выхода) можно выводить на экран или использовать в проекте. Максимальное время задержки до 32767 секунд.



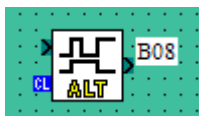
Кратковременный импульс с запуском таймера (One Short). При подаче кратковременного сигнала на вход блока включается выход и начинает работать таймер. По истечении времени выход выключается. Максимальное время задержки до 32767 секунд. Текущее значение блока можно выводить на экран или использовать в проекте.



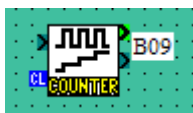
Кратковременный импульс (Pulse). Блок дает кратковременный импульс на выходе, если на вход подается сигнал. Можно выбирать момент подачи импульса:



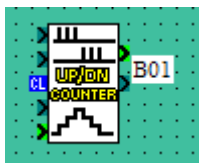
- переход с выключенного состояния на включенное;
- переход с включенного состояния на выключенное;
- переход с выключенного состояния на включенное и с включенного состояния на выключенное.



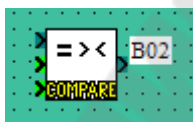
Т-триггер (Alt). При подаче сигнала на вход блока выход включается. При подаче повторного сигнала на вход выход выключается. При подаче сигнала на вход CL выход обнуляется. Если вход CL включен, то выход никогда не включится, несмотря на состояние входного сигнала.



Счетчик импульсов (Counter). При подаче на вход импульсных сигналов счетчик считает их и при достижении указанного числа импульсов включает выход. Обнуление осуществляется при подаче импульса на вход CL – обычно используется для обнуления счетчика.



Реверсивный счетчик (Up/Down counter). Счетчик импульсов, способный прибавлять и вычитать импульсы, в зависимости от того, на какой вход был подан сигнал. Первый вход сверху прибавляет импульсы, второй вход отнимает. Внутри блока необходимо указать число импульсов (Number of Counts), которое необходимо достичь для того, чтобы вход включился. Обнуление осуществляется при подаче импульса на вход CL – обычно используется для обнуления счетчика. Если на входе CL постоянно держится сигнал, то счетчик импульсов не работает. При подаче сигнала на четвертый вход счетчик сразу достигает максимального значения и выход включается.



Блок сравнения (Compare). Используется для сравнения двух величин. Величины могут быть как переменными, так и постоянными.

Если необходимо сравнить величину какого-либо сигнала с числом, то к первому светло-зеленому входу подключается вход сигнала (например, с выхода счетчика), а второй остается свободным. Открыв блок, слева указываем источник сигнала (*SigAnalogVal*), а справа – по-

стоянную величину, с которой происходит сравнение (например, 5). Также необходимо выбрать условие сравнения – больше, больше или равно, равно, не равно, меньше или равно, меньше (рис. 2.6).

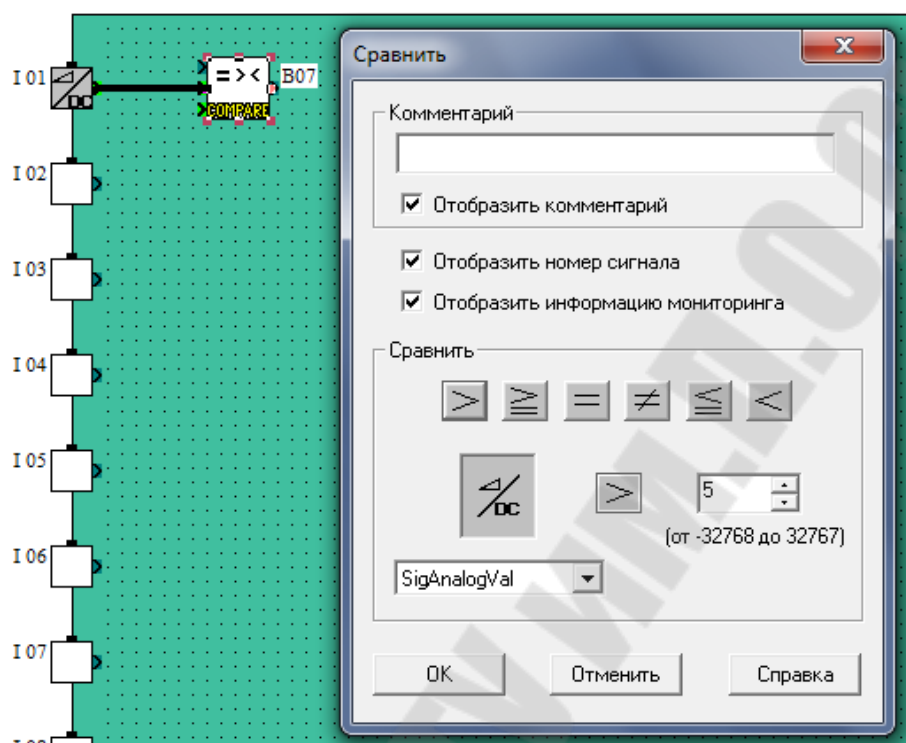


Рис. 2.6. Настройки блока сравнения

Если необходимо сравнивать две переменные величины, то ко второму светло-зеленому входу подключается второй источник сигнала. Двоичный вход может быть не подключен.

Двоичный вход подключается тогда, когда нужно делать сравнения с каким-то интервалом времени, например, каждую минуту. Тогда к первому входу подключается источник двоичного сигнала. Если двоичный вход подключен, то сравнение будет происходить только тогда, когда первый вход включен.

Если условие верно – выход блока включается.

Генератор импульсов (*Flicker*). При включении входа начинает работать генератор импульсов. Время импульса и перерывы между импульсами выбираются произвольно. Импульсы могут следовать бесконечно, либо определенное количество циклов или определенное время. Настройки производятся в меню «Параметры мерцания» (рис. 2.7).



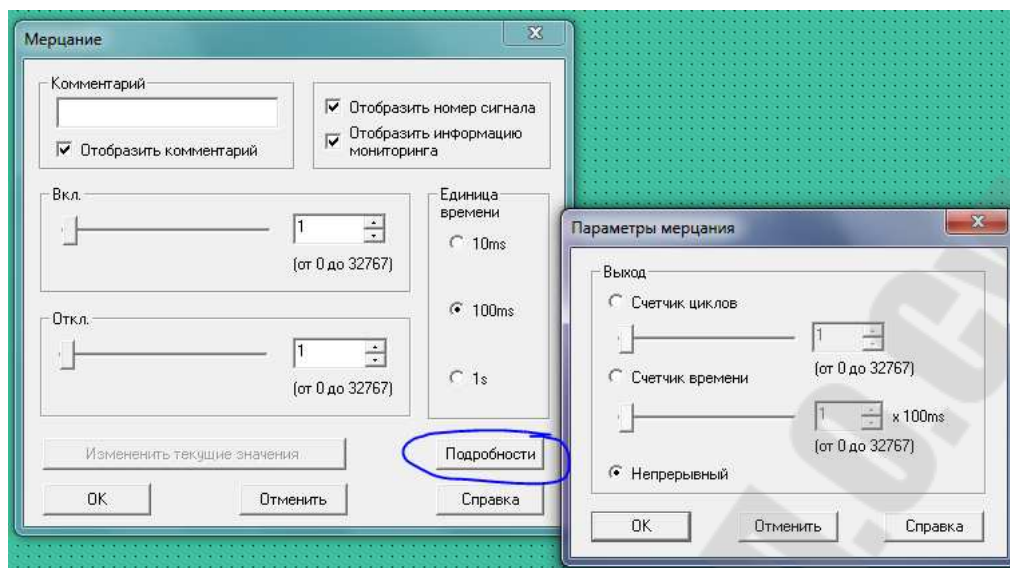
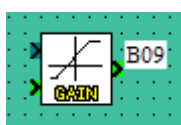


Рис. 2.7. Настройки блока генератора импульсов



(Gain). Данный блок используется для линейных преобразований. Например, преобразование аналогового входа (0–10 VDC/0–255) в проценты или реальное значение параметра в других единицах.

На рис. 2.8 приведен пример, в котором аналоговый вход (0–10 VDC/0–255) преобразуется в 0–100 %, т. е. 0 VDC – 0 %, а 10 VDC – 100 %. На выходе блока уже линейная зависимость в процентах от входного напряжения.

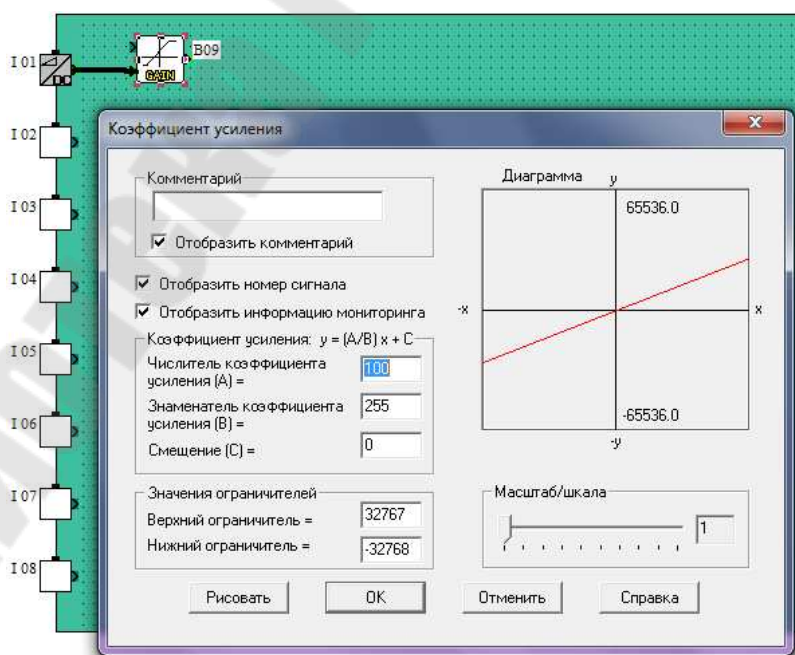
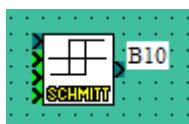


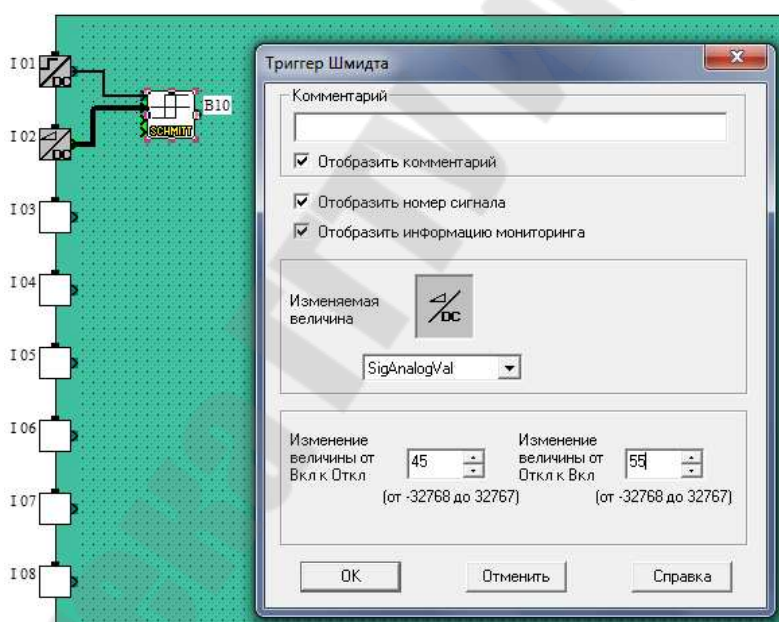
Рис. 2.8. Настройки блока усиления аналогового сигнала

Двоичный вход можно оставить свободным. В этом случае линейаризация будет происходить непрерывно. Если на двоичный вход подключен сигнал, то блок будет делать линейаризацию, только когда двоичный вход включен.



Триггер Шмидта (Schmitt Trigger). Блок имеет четыре входа: один двоичный (темно-зеленый) и три аналоговых (светло-зеленых), а также один двоичный выход. Выход включается, если входной аналоговый сигнал (первый светло-зеленый вход) выше верхнего и не ниже нижнего предела. Выход выключается, если входной аналоговый сигнал ниже нижнего и не выше верхнего предела.

Блок может использоваться для задания гистерезиса. На рис. 2.9 приведены графики включения/выключения выхода блока.



Двоичный вход
Максимальный
уровень
Аналоговый вход
Минимальный
уровень
Двоичный выход

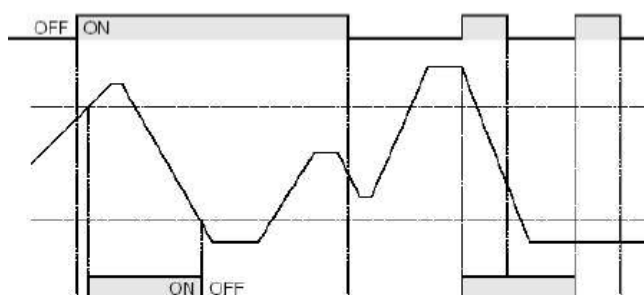
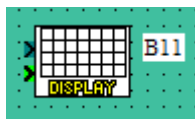


Рис. 2.9. Настройки и диаграммы работы блока триггера Шмидта

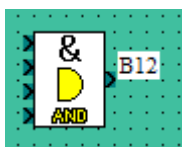
Верхний и нижний уровни срабатывания могут быть постоянными или переменными. Двоичный вход можно оставить свободным. Если двоичный вход подключен, то триггер Шмидта работает только тогда, когда двоичный вход включен.



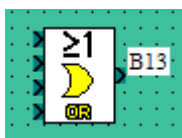
Дисплей (Display). Предназначен для отображения на экране контроллера состояния входов/выходов, текущего значения параметров, таймеров, ввода значений с клавиатуры и т. д. Размер экрана у контроллера Alpha – 4 строки по 10 символов в каждой.

Программирование экранов осуществляется слоями. То есть на одном экране может отображаться постоянный текст, на втором – одна из переменных величин, на третьем – вторая переменная величина и т. д.

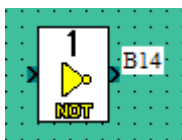
Функциональные блоки группы LOGI



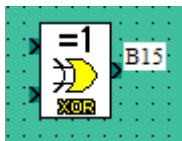
Элемент И (AND). Выход включается, когда все входы включены. Имеет до четырех входов. Неподключенные входы считаются включенными.



Элемент ИЛИ (OR). Выход включается, когда хотя бы один из входов включен. Имеет до четырех входов. Неподключенные входы считаются выключенными.

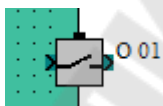


Инвертор (NOT). Выход включается, когда вход выключен и наоборот.



Исключающий ИЛИ (XOR). Имеет два входа. Выход включается, когда входные сигналы разные. Выход выключен, когда оба входных сигнала одинаковые.

Функциональные блоки группы OUT



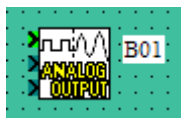
Двоичный (релейный) выход (Binary output). Выход включается, если на вход подается сигнал. Выходные блоки ставятся в выходные ячейки на рабочем поле программирования.

Группа OUT содержит ряд других функциональных блоков, например,



и т. д., но по своему действию

они ничем не отличаются друг от друга, а отличается только анимация этих блоков. То есть нет никакой разницы, какой блок применять в качестве двоичного выхода.



Блок аналогового вывода (Analog Output). Располагается в группе FUNC и предназначен для вывода аналоговых сигналов с использованием модуля расширения AL2-2DA.

3. Составление, отладка и запись программ в контроллер Alpha 2 в среде SW0D5-ALVLS-EU

Шаг 1. Для составления программы необходимо точно знать, что контроллер должен выполнять, т. е. необходимо знать точное техническое задание. Прежде всего, необходимо выяснить сколько и каких входов/выходов используется в проекте. Это необходимо для выбора подходящего контроллера.

Шаг 2. После открытия программы SW0D5-ALVLS-EU необходимо выбрать подходящий контроллер и нажать *OK*. Программа готова к программированию. Написание программы осуществляется при помощи описанных выше элементов (в контроллере Alpha 2 номенклатура функциональных блоков гораздо обширнее – до 38 различных функциональных блоков).

Шаг 3. Выбрать тип блока IN. Выбрать нужный блок кратковременным нажатием левой клавиши мышки. Вторым нажатием поставить его в пустую клетку ВХОДОВ (рис. 3.1). Двойным нажатием открыть блок и дать входу название (например, Вход № 1). Нажать *OK*.

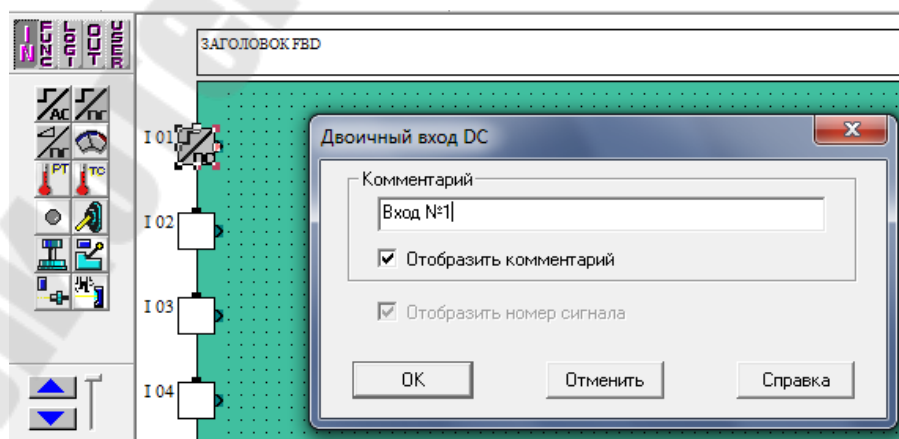


Рис. 3.1. Определение типа входа ПЛК

Шаг 4. Аналогично поставить и назвать все входы.

Шаг 5. Таким же образом поставить и назвать выходы (рис. 3.2). Только в этом случае выбор типа блока OUT. Ставить элементы необходимо в клетки ВЫХОДОВ. Дать название блокам.

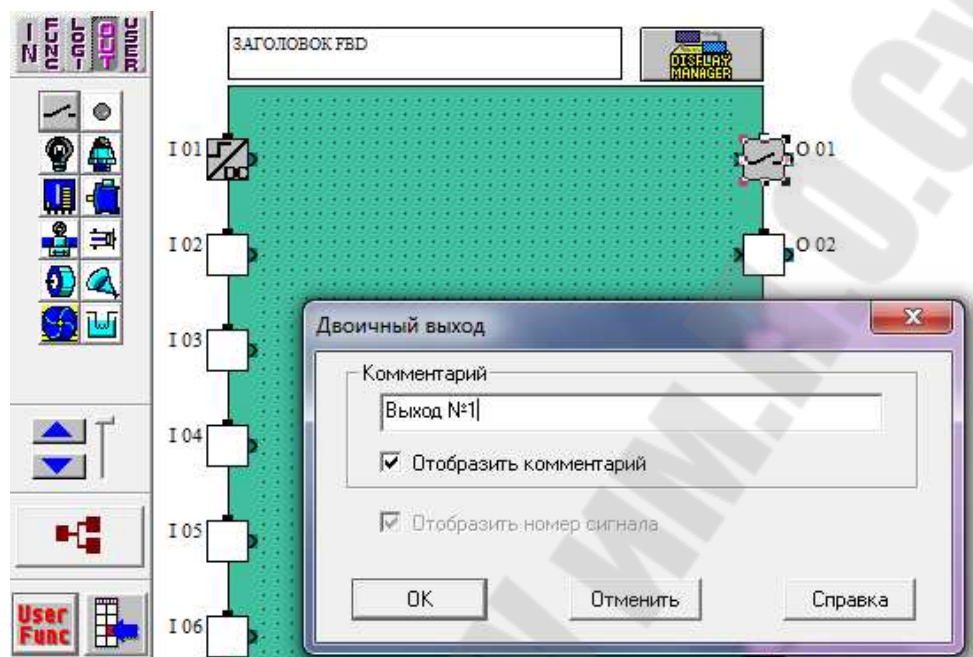


Рис. 3.2. Определение типа выхода ПЛК

Шаг 6. Начать непосредственное программирование контроллера. Для этого выбрать тип блока FUNC (функциональный блок) или LOGI (логический блок). Из перечня элементов выбрать необходимые и поставить их на рабочее поле программирования аналогичным образом, – путем перетягивания (рис. 3.3).

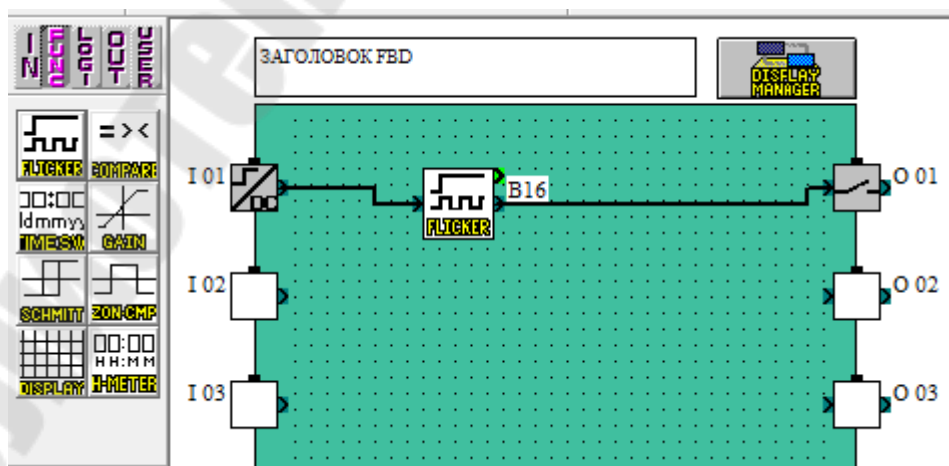


Рис. 3.3. Пример соединения входа и выхода

Соединение блоков при помощи линий. Для этого нажмите на значок СОЕДИНИТЕЛЬНЫЕ ЛИНИИ и соедините блок входа с блоком счетчика. Соединение идет от выхода одного блока с входом следующего, т. е. у линии должны быть начало и конец. При неправильном соединении программа будет подсказывать, как правильно необходимо соединять элементы.

Проверка (симуляция – *simulation*) программы производится без подключения контроллера (рис. 3.4). В этом режиме можно проверять программу и делать корректировки. Для начала режима проверки нажать на иконку  или выбрать из меню *Контроллер/Моделирование/Запуск*.

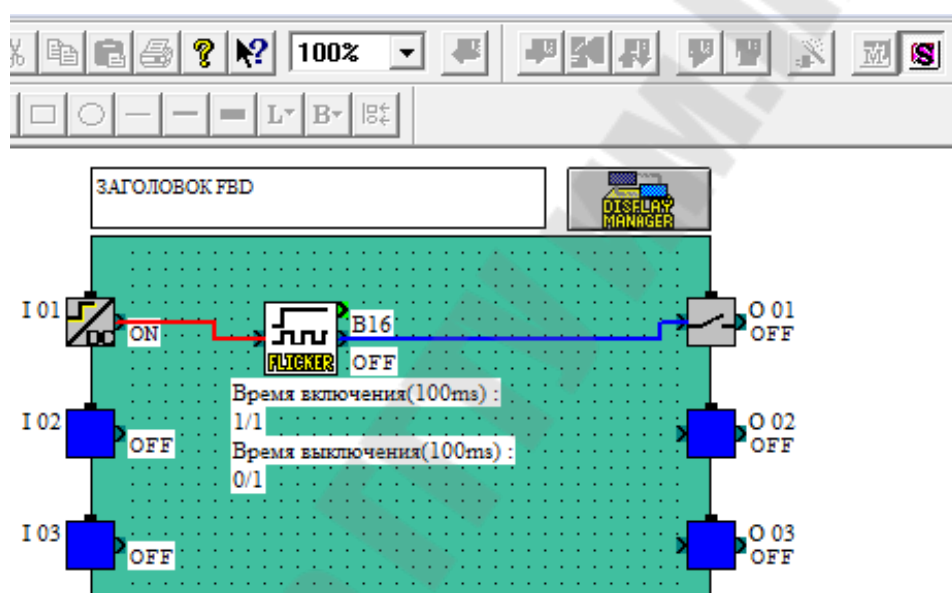


Рис. 3.4. Пример программы в режиме симуляции

Красными линиями показано наличие сигнала (ON) на данном участке, синими – отсутствие (OFF). Как видно на примере, вход I 01 включен – красная линия, а выход O 01 нет – синяя линия.

Отключение режима проверки осуществляется путем повторного нажатия на иконку  или выбрать из меню *Контроллер/Моделирование/Останов*.

После проверки программы необходимо ее записать в контроллер. Для этого необходимо соединить контроллер с компьютером при помощи кабеля AL-232CAB. При помощи отвертки осторожно открыть защитную крышку на контроллере. Подключить кабель к контроллеру, затем подключить кабель к компьютеру. При подключении кабеля контроллер должен быть выключен.

Для записи программы в контроллер необходимо подключить питание контроллера и убедиться в том, что контроллер не выполняет в данный момент другую программу (программа может автоматически начать исполняться сразу после подачи питания на контроллер). В случае выполнения программы в контроллере, перед записью новой программы контроллер нужно остановить. Для этого необходимо нажать иконку  или воспользоваться меню *Контроллер/Управление контроллером/Останов*.

Запись новой программы происходит в выключенном режиме проверки при помощи иконки  или через меню *Контроллер/Запись в контроллер*.

Программу можно также читать из контроллера. Чтение из контроллера в компьютер происходит при помощи иконки  или через меню: *Контроллер/Считывание из контроллера*.

Для запуска записанной программы на исполнение в контроллере необходимо нажать иконку  или воспользоваться меню: *Контроллер/Управление контроллером/Работа*.

После окончания записи отсоединить кабель при выключенном контроллере и поставить защитную крышку назад в контроллер.

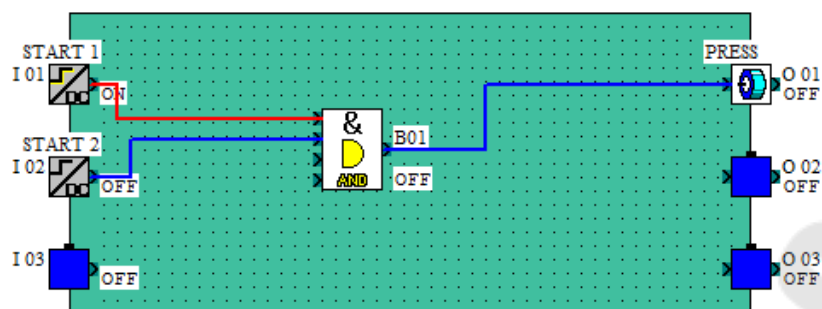
4. Примеры простейших программ в среде SW0D5-ALVLS-EU

В качестве примера рассмотрим программу автоматизации работы на прессе.

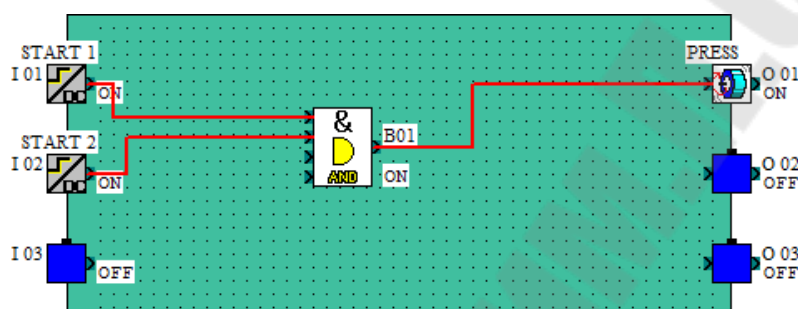
Согласно технике безопасности запуск пресса должен осуществляться одновременным нажатием двух кнопок. То есть обе руки должны быть заняты во избежание попадания одной из рук под пресс.

В примере задание упрощено, хотя на самом деле также ставится защита в виде лучей, которые дают разрешение на запуск системы. Обычно они ставятся по периметру пресса и не дают разрешения на запуск, если в зоне есть какая-либо преграда.

Входы контроллера соединены с кнопками (I 01 – START 1, I 02 – START 2) (рис. 4.1). Выходы контроллера соединены с исполнительными механизмами (O 01 – запуск пресса, O 02 – сигнальная лампа).



а)



б)

Рис. 4.1. Работа элемента И:

а – нажата только одна кнопка – пресс не включен;

б – нажато обе кнопки – пресс включен

Пример 1. Использование логического элемента И (AND).

Запуск прессы (включить выход О 01), если одновременно нажаты обе кнопки START 1 и START 2. Прессование будет осуществляться пока нажаты и удерживаются обе кнопки START (рис. 4.1).

Согласно программе, пресс (выход О 01) включится, если нажаты обе кнопки START 1 и START 2 (включены входы I 01 и I 02). Блок AND (И) включится, если на все входные сигналы, которые подключены, включены.

Пример 2. Использование счетчика импульсов (Counter) и клавиш контроллера.

Для автоматического учета штампованных деталей нужно считать количество включений прессы. Штампованные детали можно считать: по количеству срабатываний прессы или по фактическому выходу продукции. Рассмотрим первый вариант.

Перед началом рабочей смены обнулить счетчик нажатием клавиши ECS на клавиатуре контроллера. Например, данное задание можно решить как показано на рис. 4.2.

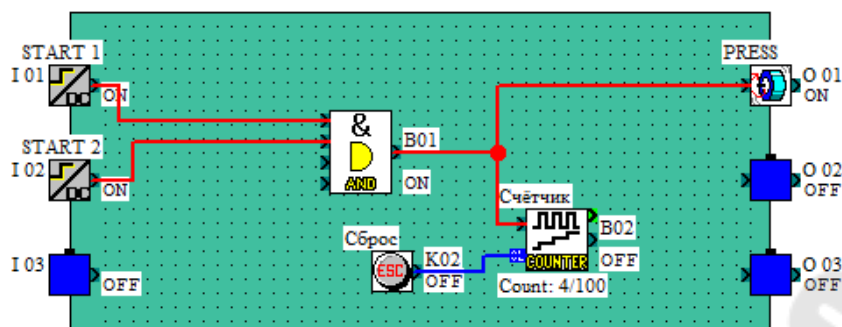


Рис. 4.2. Использование счетчика для подсчета событий

Пример 3. Использование дисплеев.

Для того чтобы человек видел сколько деталей он проштамповал, необходимо отображать на экране контроллера эту информацию.

Согласно программе, на экране отображается текущее количество запрессованных деталей (рис. 4.3). Используются два экрана – B03 и B04. На экране B03 отображается текущее значение счетчика (в настройках блока нужно выбрать текущее значение – CurCount), а на экране B04 текст – «Сделано». Двоичные входы блоков B03 и B04 можно оставить свободными, так как у нас используется только один экран.

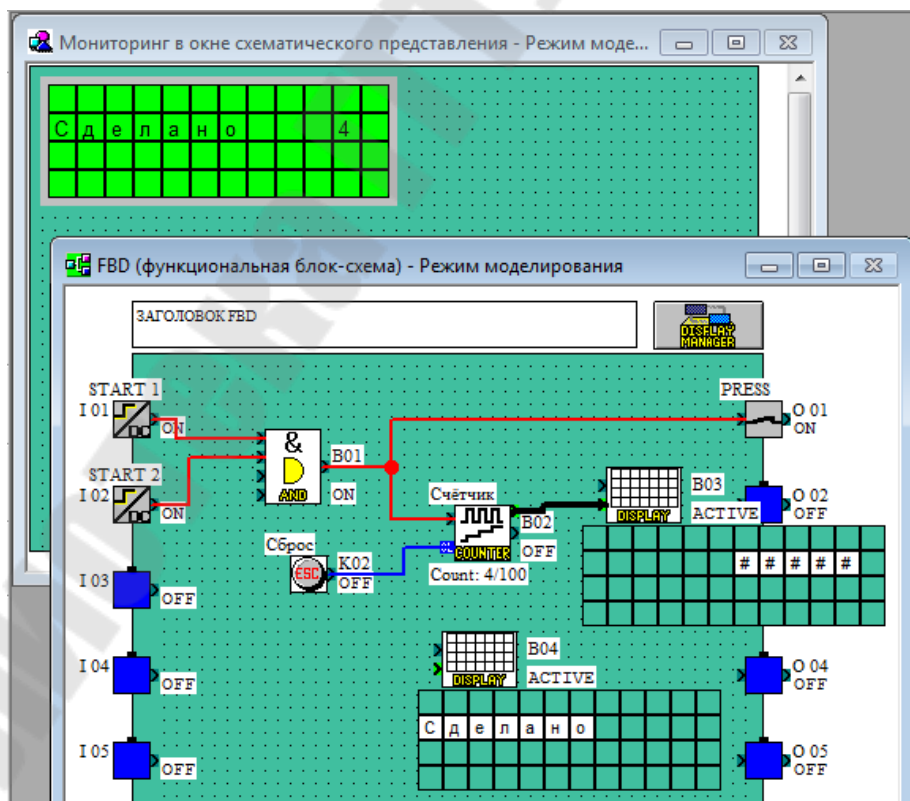


Рис. 4.3. Пример использования дисплеев

Пример 4. Пример использования двухстороннего счетчика, клавиш контроллера для задания величин, блока сравнения.

Для повышения производительности и контроля за скоростью работы работника мастер задает при помощи клавиш необходимое количество запрессованных деталей с клавиатуры контроллера (клавиши «+» и «-»), которое работник должен сделать, например, за смену.

К предыдущей части программы добавилась часть программы, при помощи которой можно вводить количество деталей для прессовки при помощи клавиш контроллера (рис. 4.4). Двухсторонний счетчик считает импульсы вверх или вниз. Задание двухстороннего счетчика осуществляется при помощи клавиш «+» и «-». Блок сравнения обнуляет двухсторонний счетчик, если количество деталей меньше нуля (количество деталей не может быть меньше нуля) – на вход CL подается сигнал.

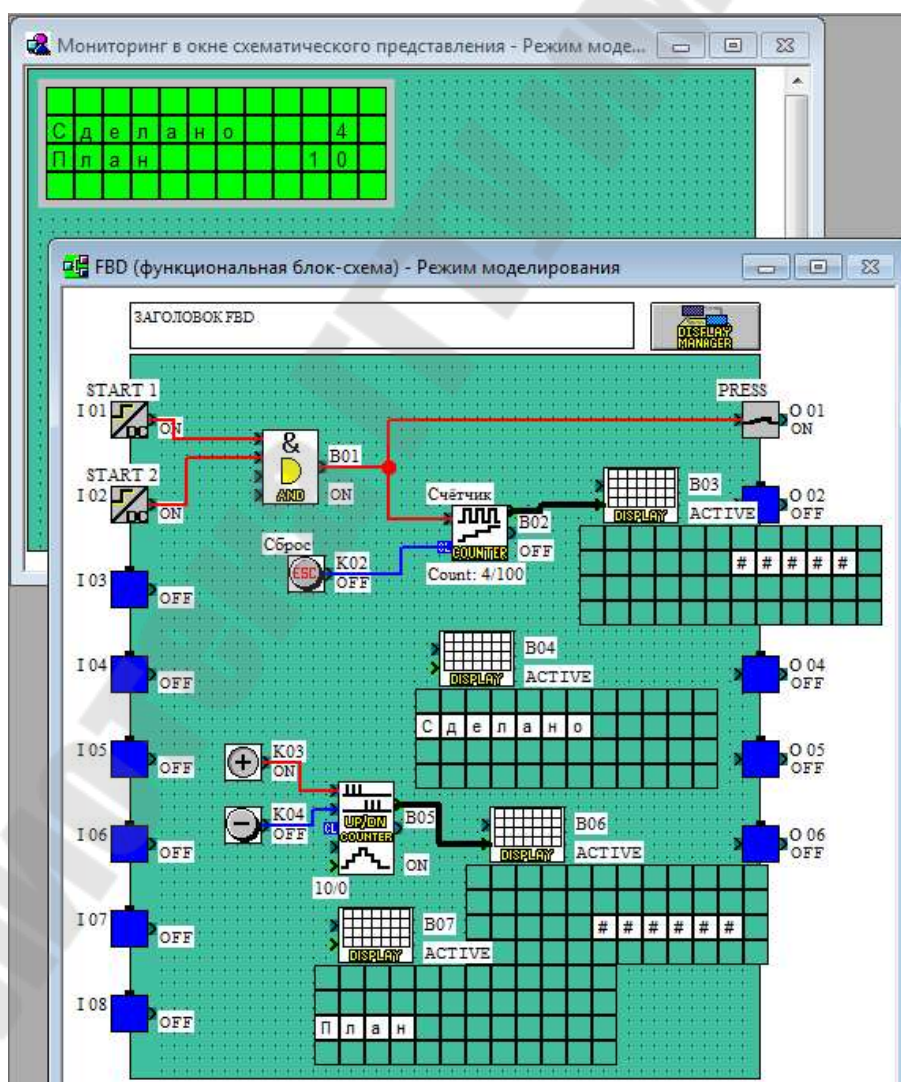


Рис. 4.4. Пример использования двухстороннего счетчика

5. Методика проведения лабораторных работ по модулю 1

5.1. Лабораторный стенд для исследования программируемых логических контроллеров на базе электропривода постоянного тока с механической передачей «винт-гайка»

Внешний вид лабораторного стенда для исследования работы ПЛК Mitsubishi Alpha 2 приведен на рис. 5.1.

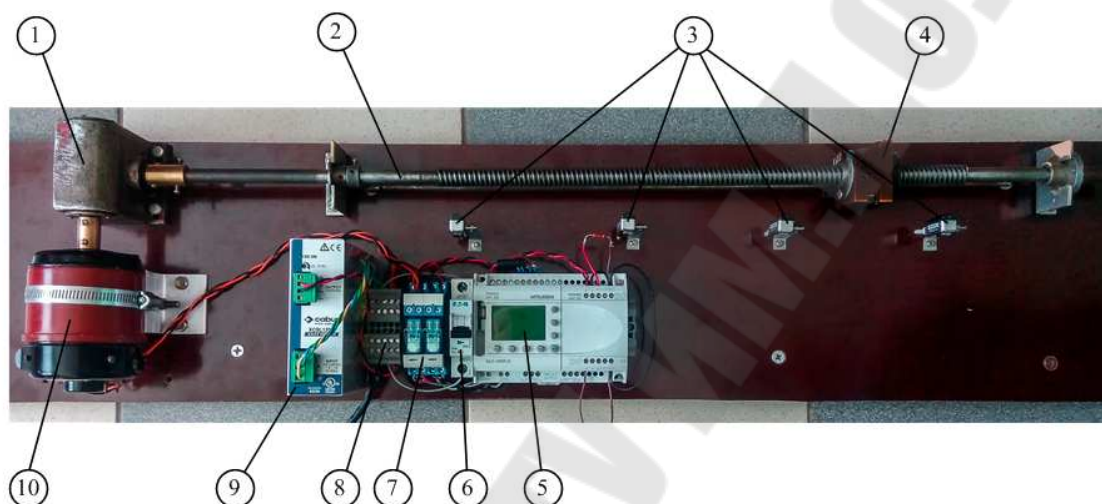


Рис. 5.1. Внешний вид лабораторного стенда для исследования ПЛК на базе электропривода постоянного тока с механической передачей «винт-гайка»

В состав стенда входят: червячный редуктор 1; механическая передача «винт-гайка», состоящая из винта 2 и гайки 4; герконовые датчики 3; ПЛК 5; выключатель 6; внешние реле направления 7; клеммник 8; блок питания 9 и электродвигатель постоянного тока 10.

Червячный редуктор предназначен для понижения скорости вращения двигателя и передачи вращающего момента под углом 90 градусов. Передача «винт-гайка» предназначена для преобразования вращательного движения винта в поступательное движение гайки (исполнительного органа). На гайке закреплен магнит, воздействующий на герконовые датчики. Герконовые датчики выполняют роль путевых датчиков и предназначены для определения положения гайки с магнитом. Блок питания формирует постоянное напряжение +24 В для питания ПЛК, а также обмоток якоря и возбуждения двигателя.

Стенд работает следующим образом. Необходимо включить стенд в сеть 220 В. После этого включить автоматический выключа-

тель 6. Напряжение питания +24 В с выхода блока питания будет подано на ПЛК 5 и на обмотку возбуждения электродвигателя 10. (Подавать напряжение на якорь электродвигателя постоянного тока без подачи напряжения на обмотку возбуждения недопустимо!)

После запуска ПЛК записанная в него программа приводит к замыканию одного из двух реле направления 7, и двигатель начинает вращаться в одном из направлений. При этом гайка с магнитом также движется в выбранном направлении. При попадании магнита в поле действия датчика 3, сигнал с него поступает на соответствующий вход ПЛК таким образом, влияя на работу последнего.

Герконовые датчики нумеруются слева направо и подключены соответственно ко входам ПЛК I 01–I 04. При написании программы этим входам следует назначать релейный (двоичный) тип (см. п. 2).

В качестве управляющих выходов ПЛК использованы релейные выходы O 01 и O 02. При включении выхода O 01 гайка движется влево (от датчика 4 к датчику 1), при включении выхода O 02 – вправо.

5.2. Последовательность выполнения лабораторных работ

1. Ознакомиться с составом лабораторного оборудования.
2. Изучить электрическую схему стенда.
3. Включить питание контроллера (выключатель, установленный на DIN-рейке). Ознакомиться с меню контроллера, отображаемого на дисплее. Для этого необходимо использовать клавиши со стрелками, клавишу подтверждения (OK) и клавишу возврата (Esc).
4. Если язык вывода информации на дисплее отличается от русского, необходимо выбрать русский язык.
5. Отключить питание контроллера.
6. Запустить на компьютере программное обеспечение SW0D5-ALVLS-EU для программирования ПЛК Alpha 2.
7. Изучить основы работы в данной программе, используя настоящее учебно-методическое пособие, руководство пользователя или помощь (Help) к программному обеспечению.
8. Создать новый проект для контроллера Alpha 2 с 8 входами и 6 выходами.
9. Написать программу для ПЛК в соответствии с заданием с помощью программного обеспечения SW0D5-ALVLS-EU и сохранить ее на жестком диске компьютера.
10. Проверить работоспособность программы в режиме симуляции и продемонстрировать ее преподавателю.

11. Подключить к ПЛК интерфейсный кабель для связи ПЛК с компьютером.

12. Включить питание контроллера.

13. С помощью программного обеспечения SW0D5-ALVLS-EU загрузить написанную программу в память ПЛК (в командной строке в меню «Контроллер» выбрать пункт «Запись в контроллер»).

14. Перевести ПЛК в режим RUN (нажать *OK*, выбрать пункт RUN и нажать *OK* повторно).

15. Продемонстрировать учебный стенд макета преподавателю.

Отчет по лабораторной работе должен содержать: название, цель работы, индивидуальное задание, выдаваемое преподавателем, программу, написанную на языке функциональных блоков в программной среде SW0D5-ALVLS-EU, пояснения по логике написания программы.

5.3. Задания к лабораторным работам

Лабораторная работа № 1

Знакомство с ПЛК Mitsubishi Alpha 2 и средой программирования Alpha Software

Ознакомиться с материалами параграфа 5.1 настоящего учебно-методического пособия, а также с техническим описанием ПЛК Mitsubishi Alpha 2 [5–8]. Установить программное обеспечение Alpha Software (если оно не установлено). Создать простейший проект, в котором кнопка «Вправо» (*Right arrow*) ПЛК запустит движение исполнительного органа вправо. А кнопка «Влево» (*Left arrow*) запустит движение влево. При этом между моментами смены направления движения должна быть обеспечена пауза для исключения одновременного срабатывания двух реле направления и возникновения короткого замыкания блока питания стенда.

Лабораторная работа № 2

Исследование возможностей обработки дискретных сигналов на языке FBD

Составить программу, в соответствии с которой после включения стенда исполнительный орган будет парковаться у датчика № 1. После выдержки в 5 с исполнительный орган начнет движение к датчику № 4, а затем вернется к датчику № 2, где и остановится. При этом на экран ПЛК должно выводиться сообщение о текущем режиме работы.

Лабораторная работа № 3

Исследование возможностей по заданию цикла работы производственного механизма на языке FBD

Составить программу, в соответствии с которой после включения станда исполнительный орган будет парковаться у датчика № 3. После чего ПЛК будет ожидать, когда пользователь нажатием на кнопку «+» контроллера укажет номер датчика и нажмет кнопку *OK*. Далее исполнительный орган должен доехать до указанного датчика и после выдержки времени вернуться к датчику № 3. При этом на экран ПЛК должно выводиться сообщение о текущем режиме работы.

Лабораторная работа № 4

Исследование возможностей обработки аналоговых сигналов на языке FBD

Составить программу, в соответствии с которой после включения станда исполнительный орган будет парковаться у датчика №4. После выдержки в 5 с исполнительный орган начнет движение к датчику № 2, затем вернется к датчику № 3, после чего отправится к датчику № 1 и, наконец, вернется к датчику № 4. При этом на экран ПЛК должно выводиться сообщение о текущем режиме работы.

6. Контрольные вопросы по модулю 1

1. Что такое программируемый логический контроллер и в чем его преимущество перед релейно-контакторными схемами?
2. Какие характеристики имеют ПЛК семейства Alpha 2 производства Mitsubishi Electric?
3. Какие типы входов и выходов имеет контроллер Alpha 2 и каковы их электрические характеристики?
4. Что представляет собой среда программирования ПЛК серии Alpha 2?
5. Какие группы функциональных блоков наиболее часто используются при составлении программ для ПЛК семейства Alpha 2? Охарактеризуйте выполняемые ими функции.
6. Как составляется программа в среде SW0D5-ALVLS-EU?
7. Для чего предназначен режим симуляции в программной среде SW0D5-ALVLS-EU и как его произвести?
8. Приведите примеры применения функциональных блоков группы IN.

9. Приведите примеры применения дискретных функциональных блоков группы FUNC.

10. Приведите пример использования аналоговых функциональных блоков группы FUNC.

11. Как записать программу в контроллер?

Модуль 2. ПРОГРАМИРОВАНИЕ НА ЯЗЫКЕ ЛЕСТНИЧНЫХ ДИАГРАММ

7. Краткая характеристика аппаратной части

На рис. 7.1 приведено расположение элементов ПЛК Modicon M221: 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.

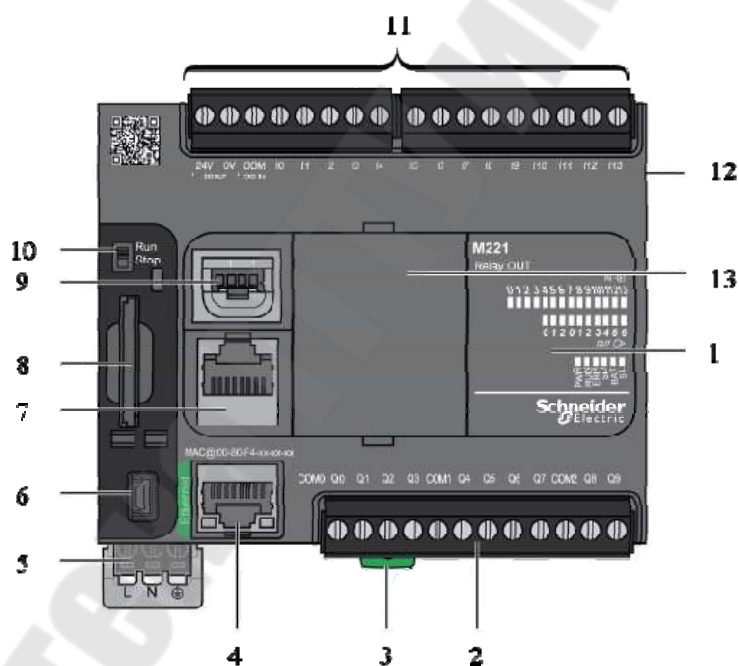


Рис. 7.1. Расположение элементов программируемого логического контроллера

Характеристики контроллера:

- 256 Кбайт для внутренних переменных RAM;
- 256 Кбайт флэш-память для резервного копирования приложений и данных;
- 256 Кбайт для пользовательского применения и данные RAM с 10000 инструкций;

- SD карта до 2 Гб опционально 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. 7.2 приведено расположение элементов конструкции ПЧ *Altivar 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).

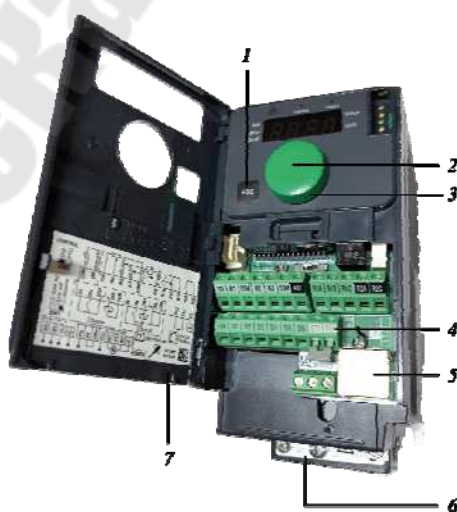


Рис. 7.2. Расположение элементов конструкции преобразователя частоты *Altivar ATV320*

На рис. 7.3 показаны функции дисплея ПЧ *Altivar 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.

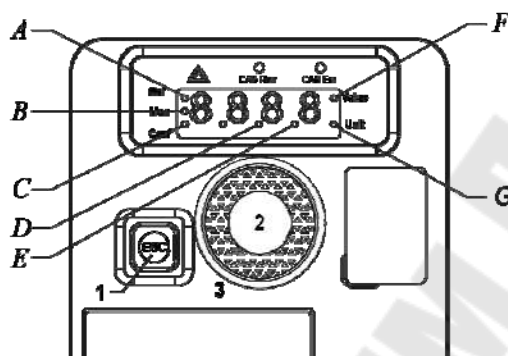


Рис. 7.3. Функции дисплея преобразователя частоты *Altivar ATV320*

Преобразователи серии *Altivar Machine ATV320* ориентированы на встраивание в машины и 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.

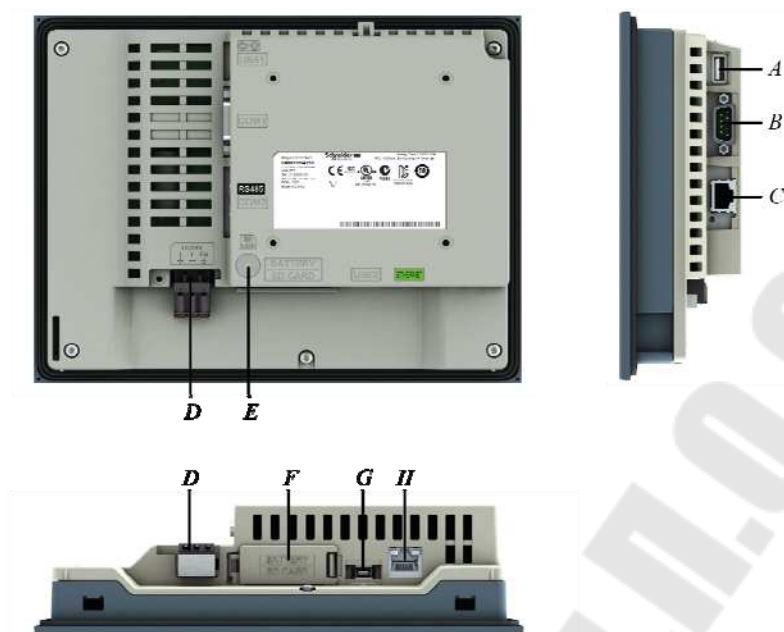


Рис. 7.4. Элементы программируемого терминала *Magelis GTO4310*

На рис. 7.4 показаны элементы конструкции ПТ *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.

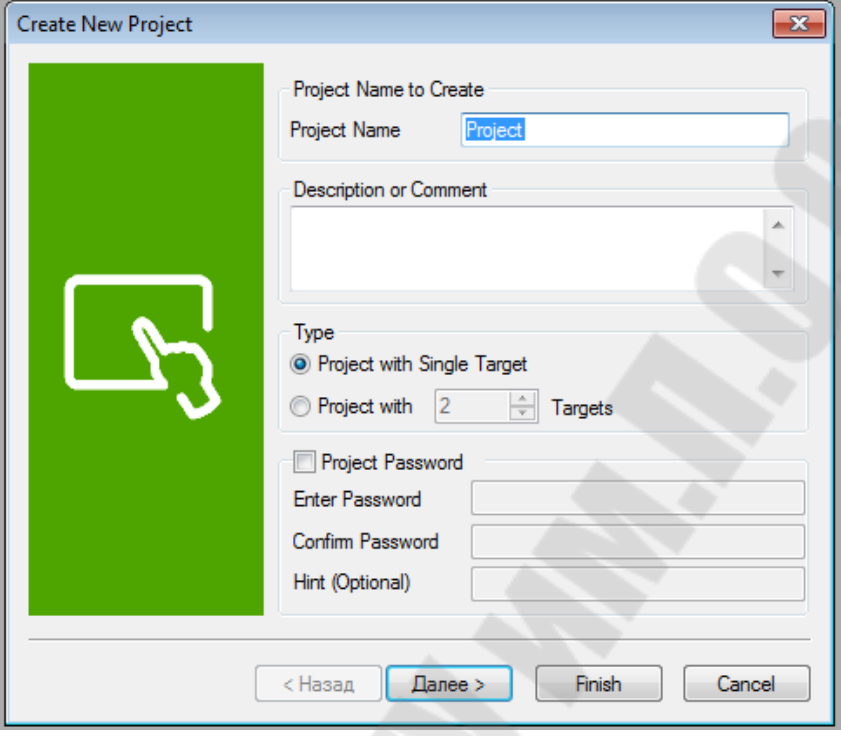
8. Создание нового проекта

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

Для создания нового проекта в среде программирования *Vijeo Designer* необходимо:

1. Задать имя проекта (*Project Name*) и его описание (*Description or Comment*) при необходимости (рис. 8.1).
2. Выбрать количество целей (*target*) (под целью понимается терминал).
3. При необходимости задать пароль (*Project Password*).
4. Присвоить имена терминалам (*Target Name*) (рис. 8.2).
5. Выбрать тип терминала (*Target Type*) – в нашем случае это терминал серии GTO, модель терминала GTO4310 с разрешением 640 × 480.
6. Назначить IP-адрес терминала (например, 192.168.1.100), а также маску подсети (*Subnet Mask*) (255.255.255.0) (рис. 8.3).

7. Добавить драйверы внешнего оборудования (нажать *Add*) (рис. 8.4).



Project Name to Create

Project Name

Description or Comment

Type

☒ Project with Single Target

☐ Project with Targets

☐ Project Password

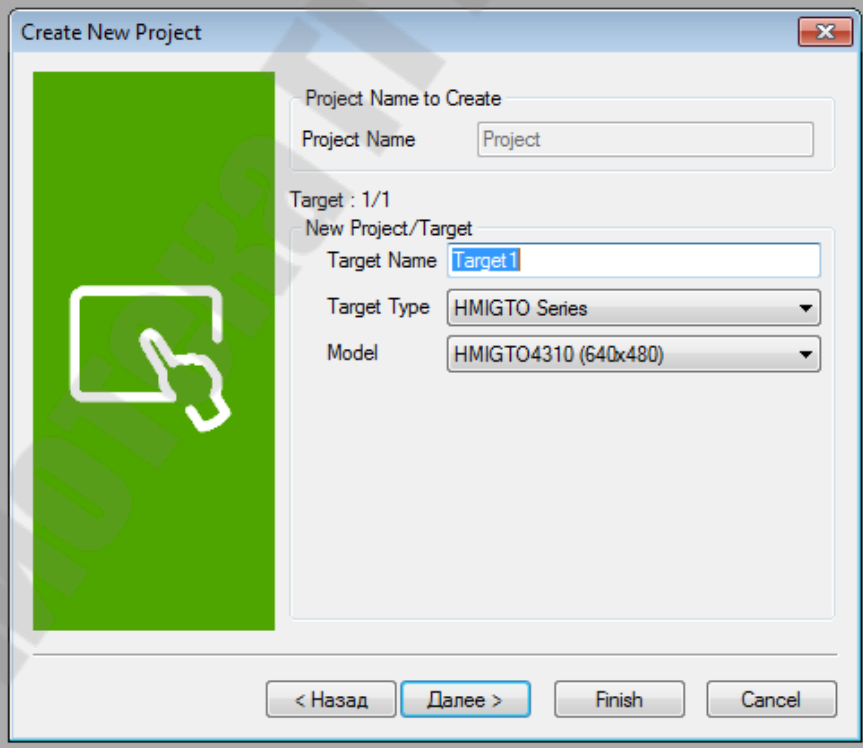
Enter Password

Confirm Password

Hint (Optional)

< Назад Далее > Finish Cancel

Рис. 8.1. Настройка имени проекта и его описания



Project Name to Create

Project Name

Target : 1/1

New Project/Target

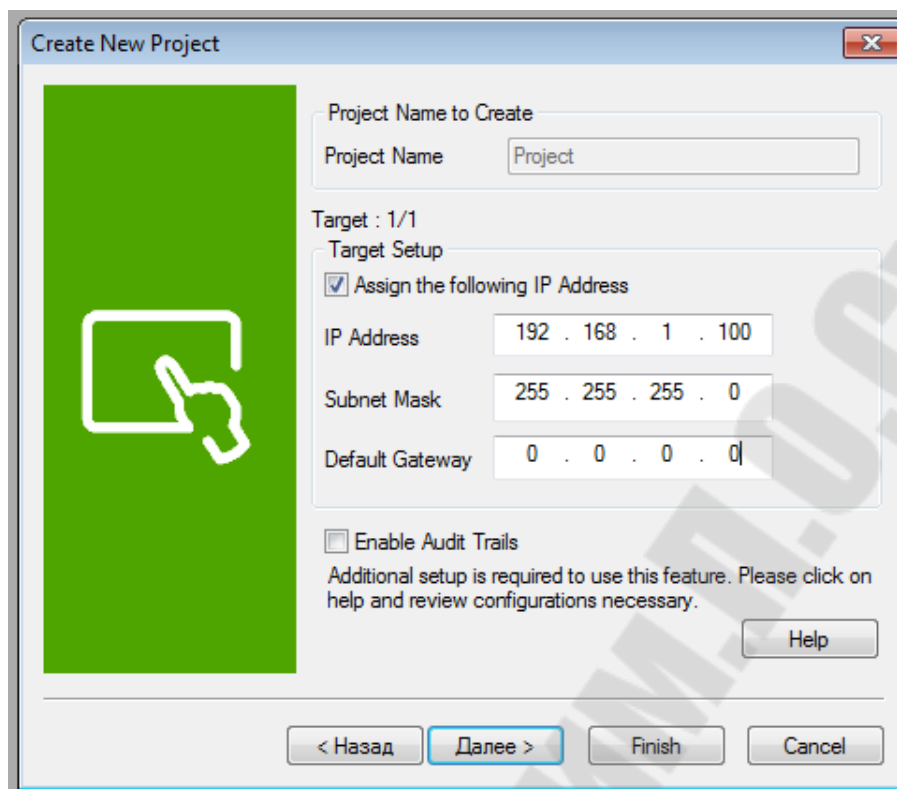
Target Name

Target Type

Model

< Назад Далее > Finish Cancel

Рис. 8.2. Присвоение имени



Create New Project

Project Name to Create

Project Name

Target : 1/1

Target Setup

☒ Assign the following IP Address

IP Address

Subnet Mask

Default Gateway

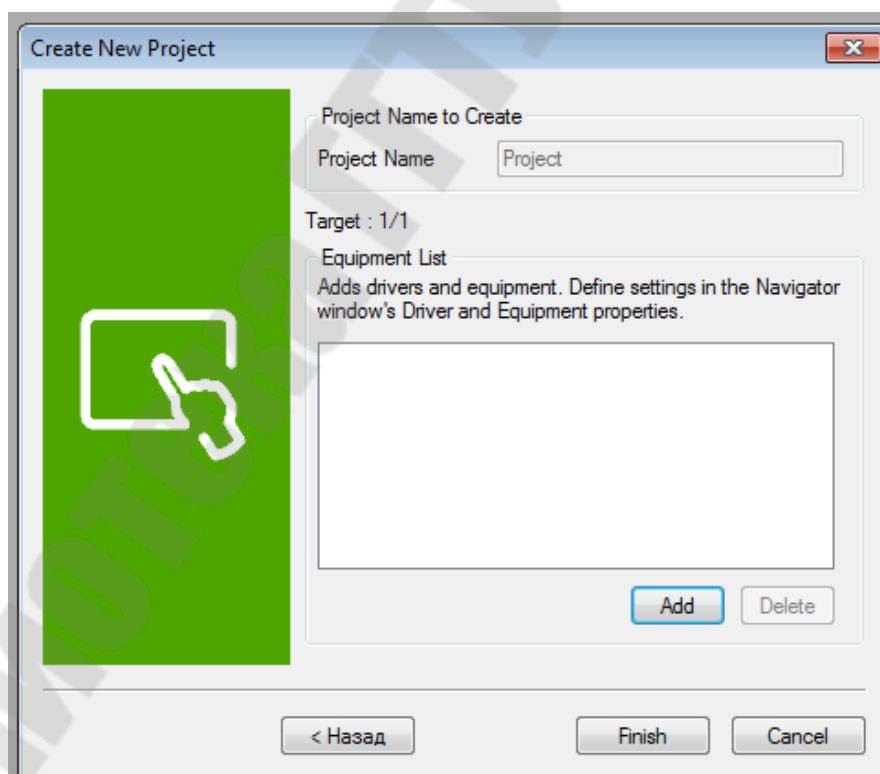
☐ Enable Audit Trails

Additional setup is required to use this feature. Please click on help and review configurations necessary.

[Help](#)

< Назад [Далее >](#) Finish Cancel

Рис. 8.3. Маска подсети



Create New Project

Project Name to Create

Project Name

Target : 1/1

Equipment List

Adds drivers and equipment. Define settings in the Navigator window's Driver and Equipment properties.

[Add](#) [Delete](#)

< Назад Finish Cancel

Рис. 8.4. Добавление драйверов внешнего оборудования

8. Для связи с внешним оборудованием по локальной сети Ethernet выбрать драйвер *Modbus TCP/IP* и тип оборудования *Modbus Equipment* (рис. 8.5).

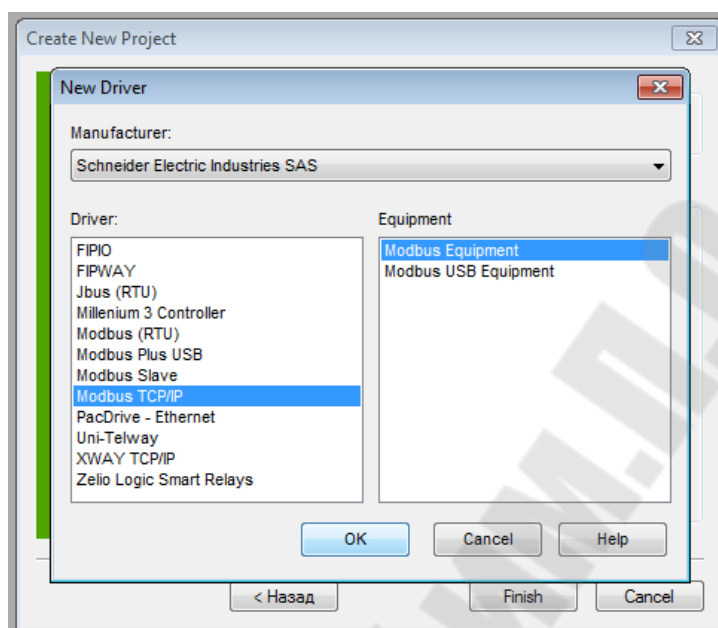


Рис. 8.5. Выбор типа оборудования Modbus Equipment

9. Для связи с внешним оборудованием по протоколу RS-485 выбрать драйвер *Modbus (RTU)* и тип оборудования *Modbus Equipment* (рис. 8.6 и 8.7).

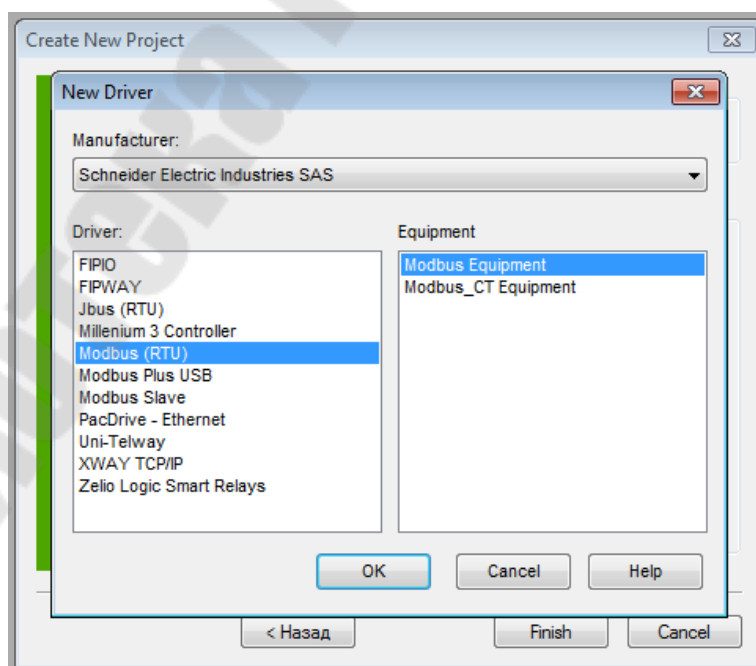


Рис. 8.6. Выбор драйвера Modbus (RTU)

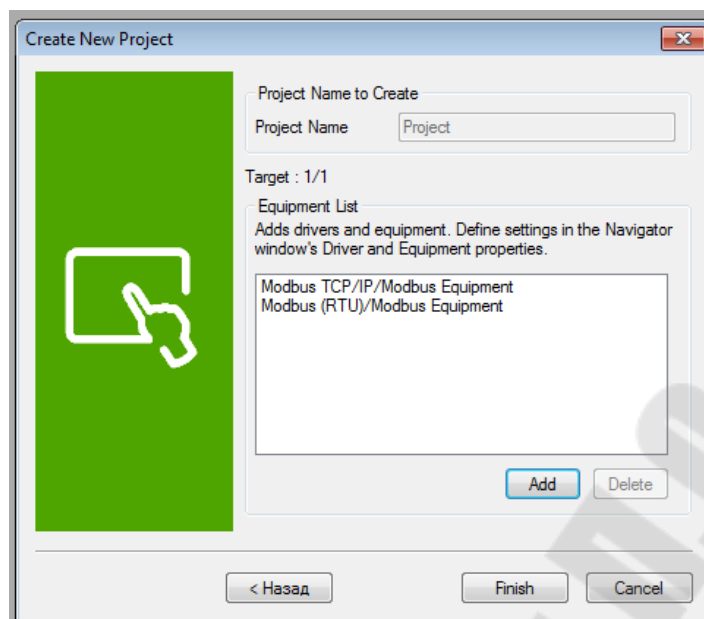


Рис. 8.7. Выбор типа оборудования *Modbus Equipment*

- Для изменения параметров сетевого адреса терминала необходимо:
- кликнуть по имени терминала в дереве проекта (*Target1*) (рис. 8.8);
 - в открывшемся меню выбрать *Network*;
 - нажать на «...» напротив *Network Configuration*;
 - установить необходимый адрес (*IP Address*), например, *192.168.1.2* и маску подсети (*Subnet Mask*) *255.255.255.0*, и нажать *OK*;
 - обязательно нажать кнопку *Use for Download*.

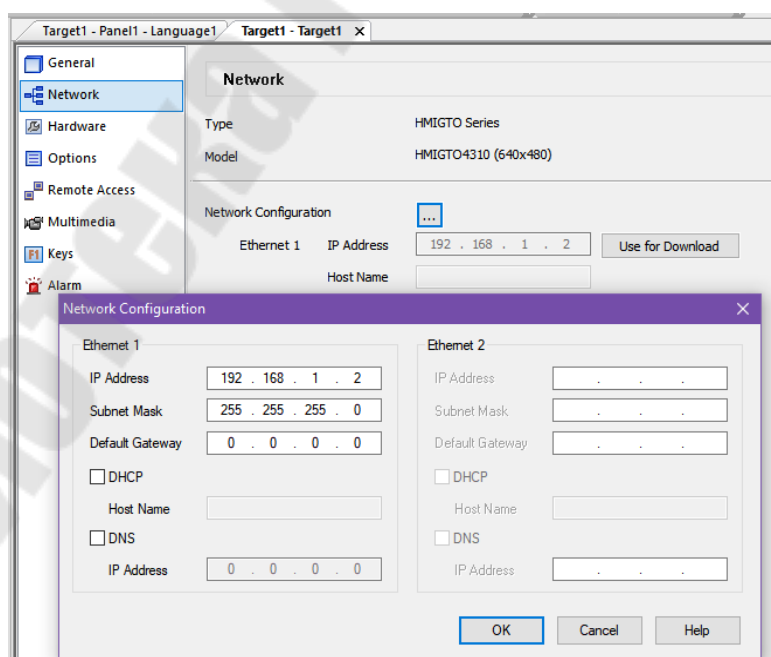


Рис. 8.8. Дерево проекта

В меню *General* необходимо в качестве способа загрузки проекта в ПТ выбрать USB (рис. 8.9).

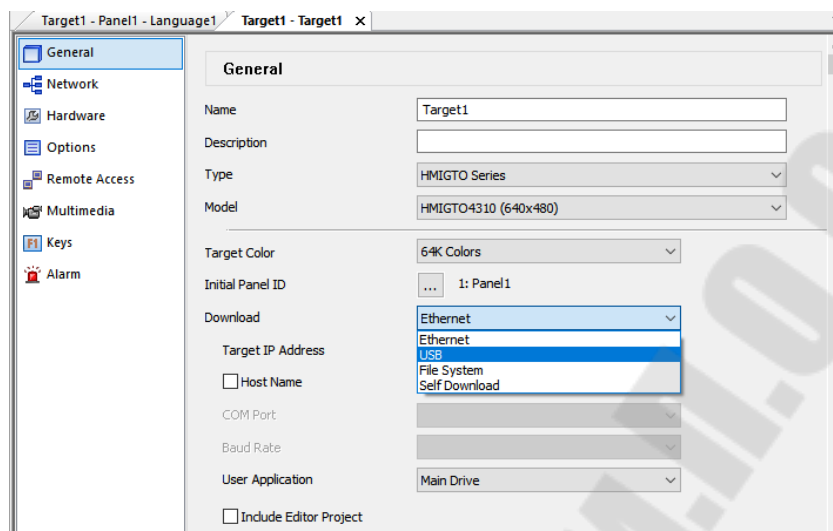


Рис. 8.9. Меню General

Для настройки связи между ПТ и ПЛК необходимо в дереве проекта выбрать пункт *ModbusEquipment01[0.0.0.0]* и указать адрес (*IP Address*) устройства (ПЛК) таким образом, чтобы ПЛК и ПТ находились в одной сети (рис. 8.10). То есть если адрес ПТ *192.168.1.2*, то адрес ПЛК должен отличаться только последней цифрой, например, *192.168.1.3*.

Кроме того, на вкладке *Equipment Configuration* необходимо поставить галочку в чекбоксе *IEC61131 Syntax* для правильного отображения переменных ПЛК в *Vijeo Designer*.

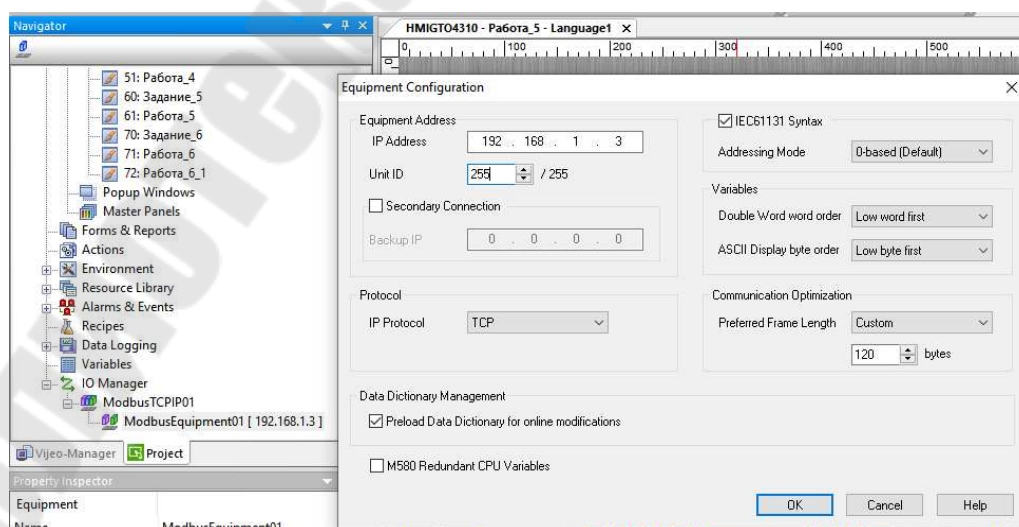


Рис. 8.10. Вкладка *Equipment Configuration*

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:

1. Выбрать пункт меню *New project* (рис. 8.11).

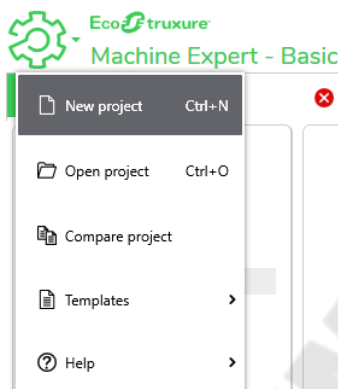


Рис. 8.11. Выбор пункта меню

2. На вкладке свойств (*Properties*) (рис. 8.12), в разделе *Application Protect*, в пунктах защита чтения (*Read protection*) и защита записи (*Write protection*) установить режим *Active* и записать пароль (например, 0000). Также возможно выбрать опцию *Inactive*. В этом случае, парольная защита не будет действовать. Но если ранее был установлен пароль, потребуется его ввести для снятия защиты.

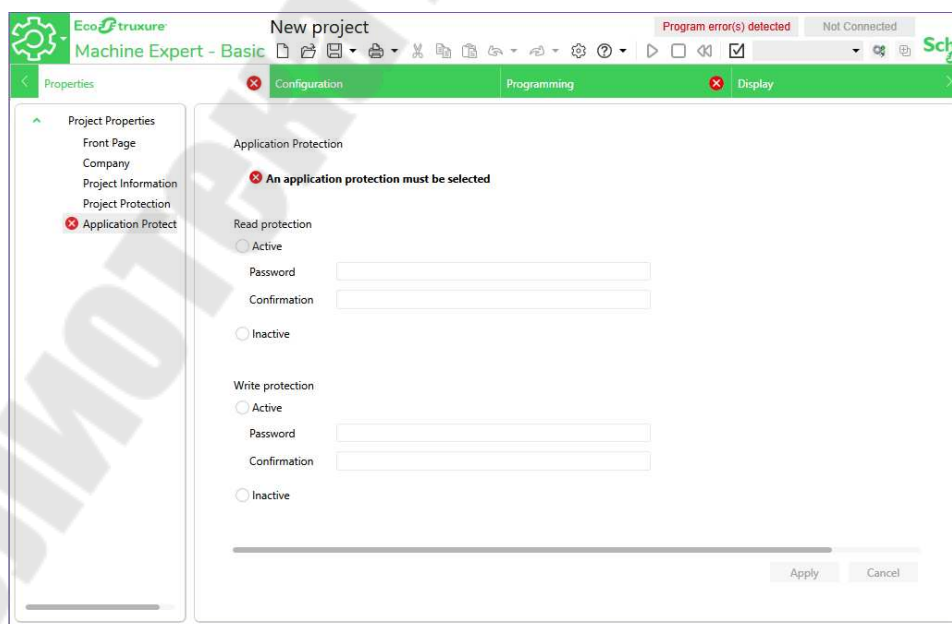


Рис. 8.12. Вкладка свойств

3. Для подтверждения действий нажать кнопку *Apply*.
4. Перейти на вкладку *Configuration* и выбрать нужную модель контроллера (рис. 8.13).

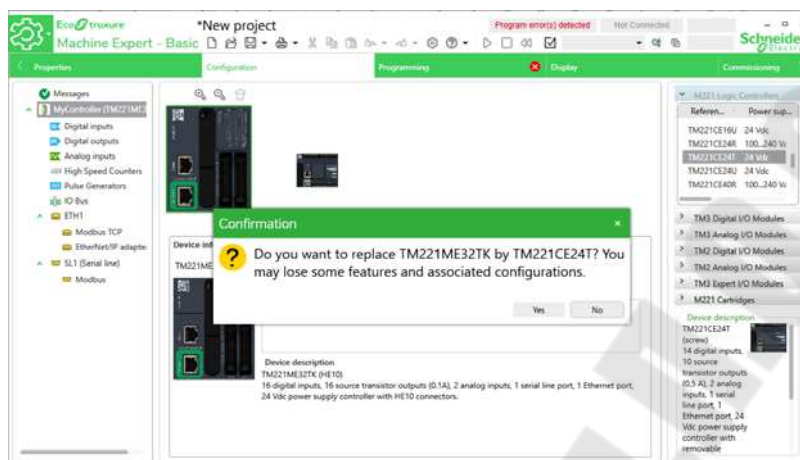


Рис. 8.13. Выбор нужной модели контроллера

5. 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*».

6. 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. 8.14).



Рис. 8.14. Выбор модели контроллера по умолчанию

Для связи контроллера с другими элементами системы (преобразователями частоты, программируемыми терминалами и др.) 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, и для его конфигурации необходимо выбрать раздел *ETH1* на вкладке *Configuration* (рис. 8.15).

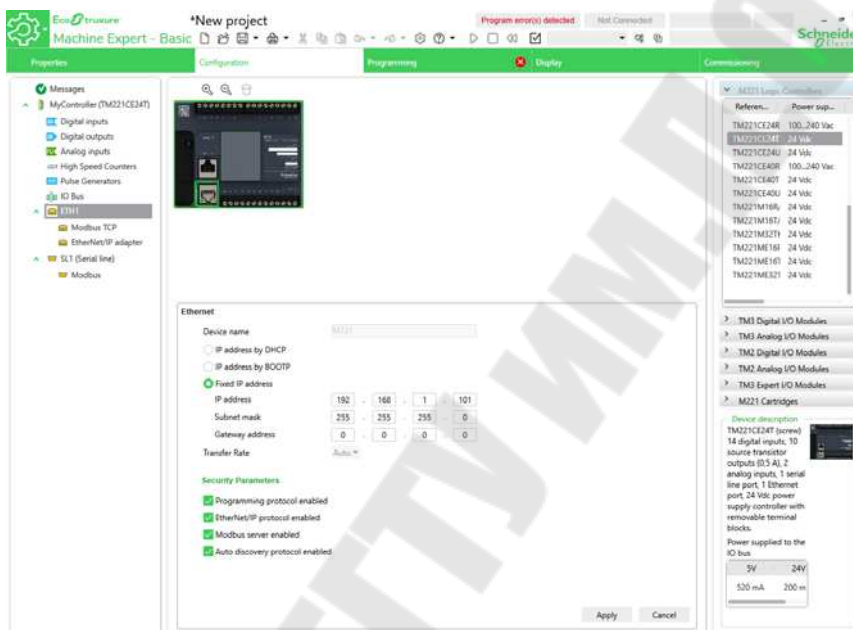


Рис. 8.15. Выбор раздела ETH1

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*.

9. Инициализация прерываний

Прерывания в терминологии Schneider – это события (*Events*).

Изначально задача *Events* во вкладке *Programming* не активна (рис. 9.1).

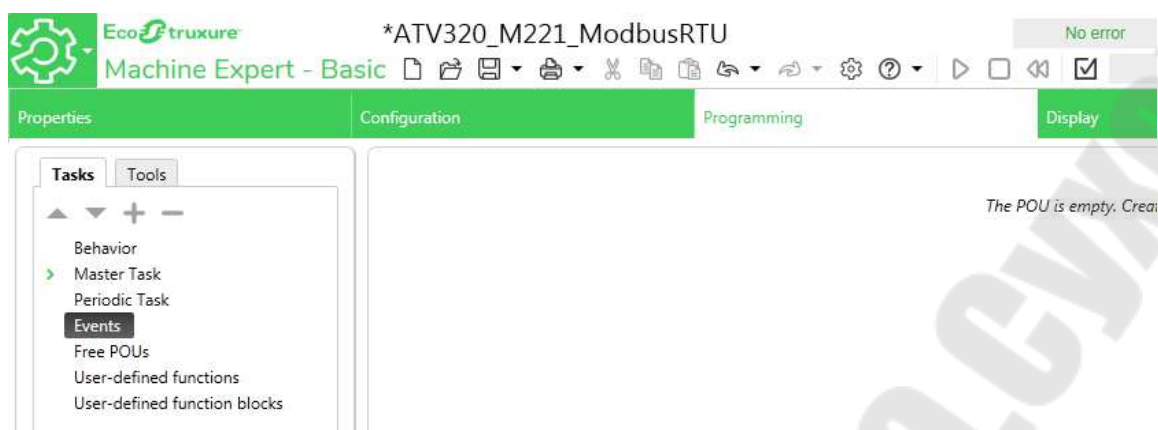


Рис. 9.1. Вкладка Programming

Для активации задач *Events* необходимо перейти на вкладку настройки контроллера (*Configuration*) и выбрать цифровые входы (*Digital inputs*) (рис. 9.2).

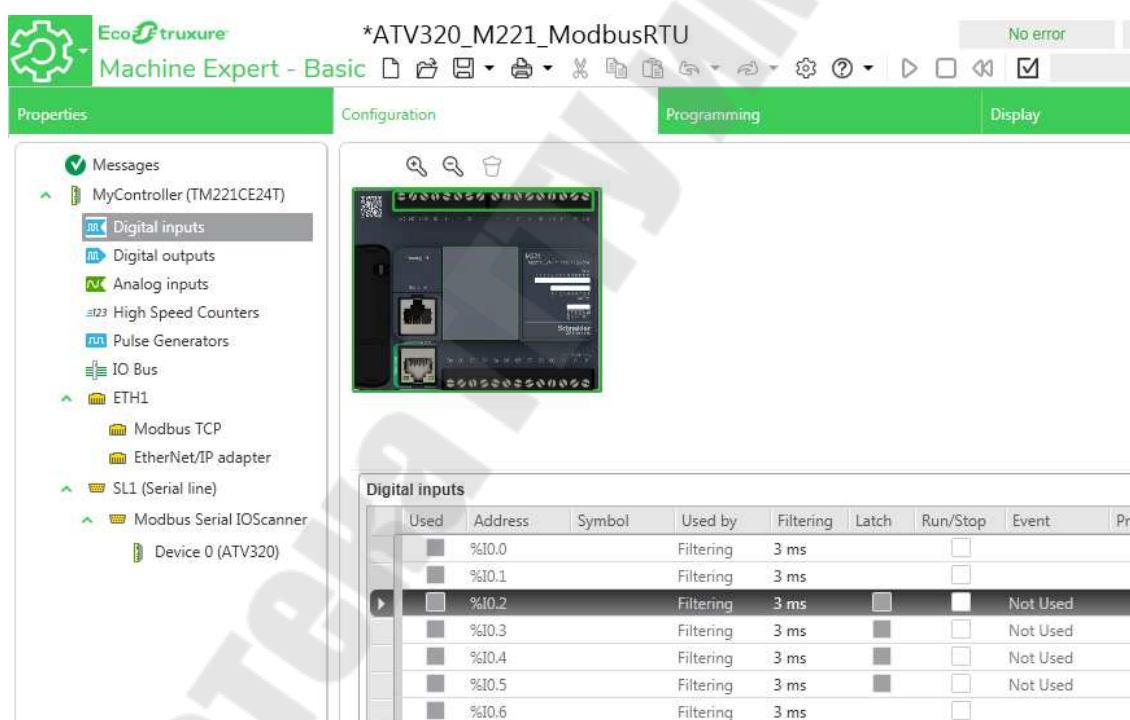


Рис. 9.2. Выбор цифровых входов

За прерывания отвечают входы *%I0.2*, *%I0.3*, *%I0.4*, *%I0.5*. Для этих входов необходимо отключить входные фильтры (3 ms по умолчанию) (рис. 9.3) – в выпадающем меню необходимо выбрать *No Filter*.

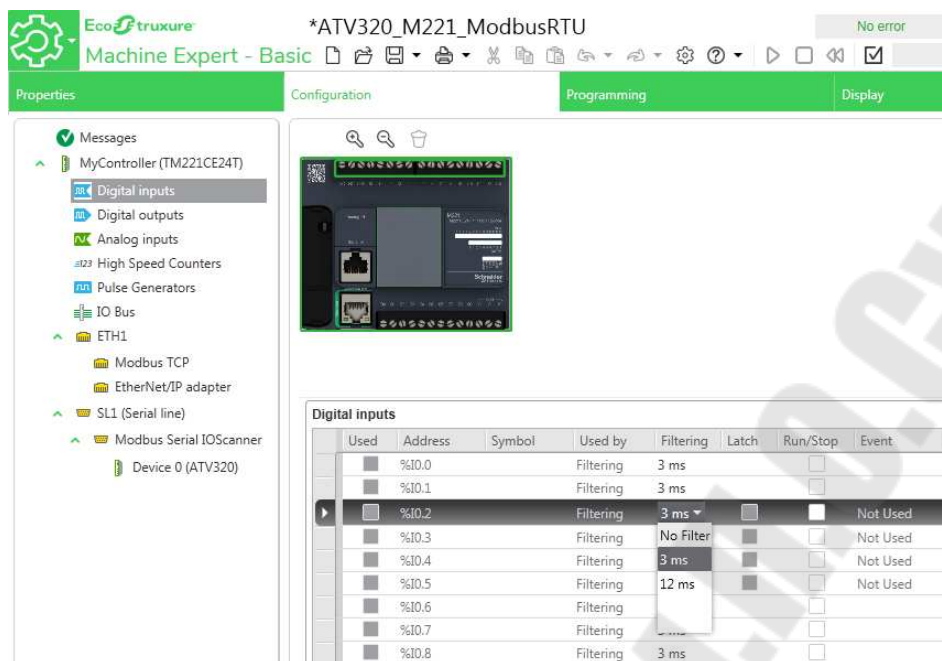


Рис. 9.3. Отключение входных фильтров

Затем в колонке *Events* заменить *Not Used* на нужный критерий срабатывания прерывания (рис. 9.4).

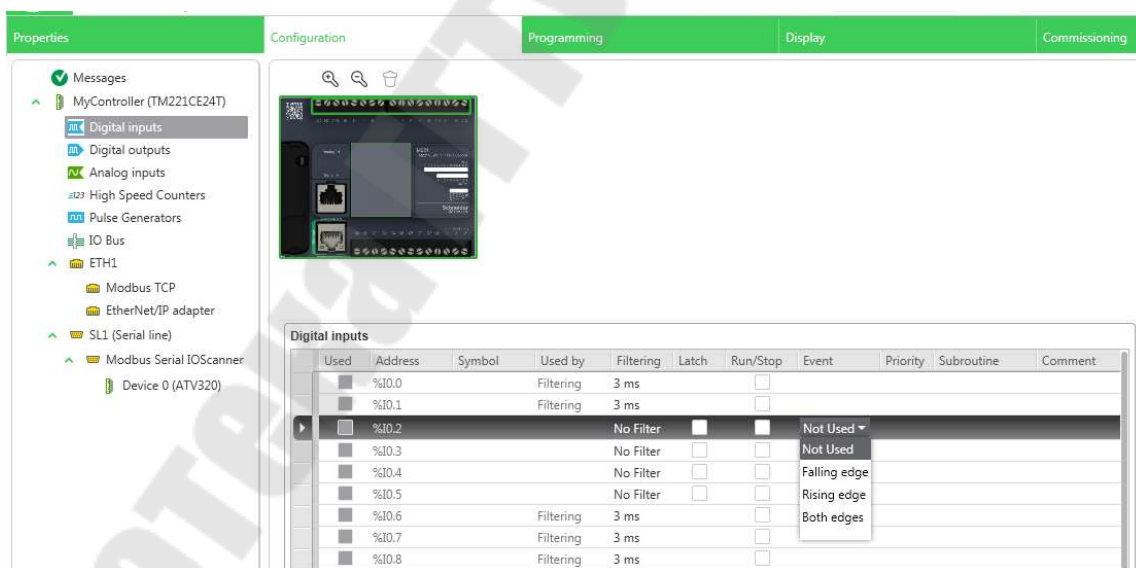


Рис. 9.4. Выбор критерия срабатывания прерывания

Например, выбрать передний фронт (*Rising edge*) (рис. 9.5). 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*!

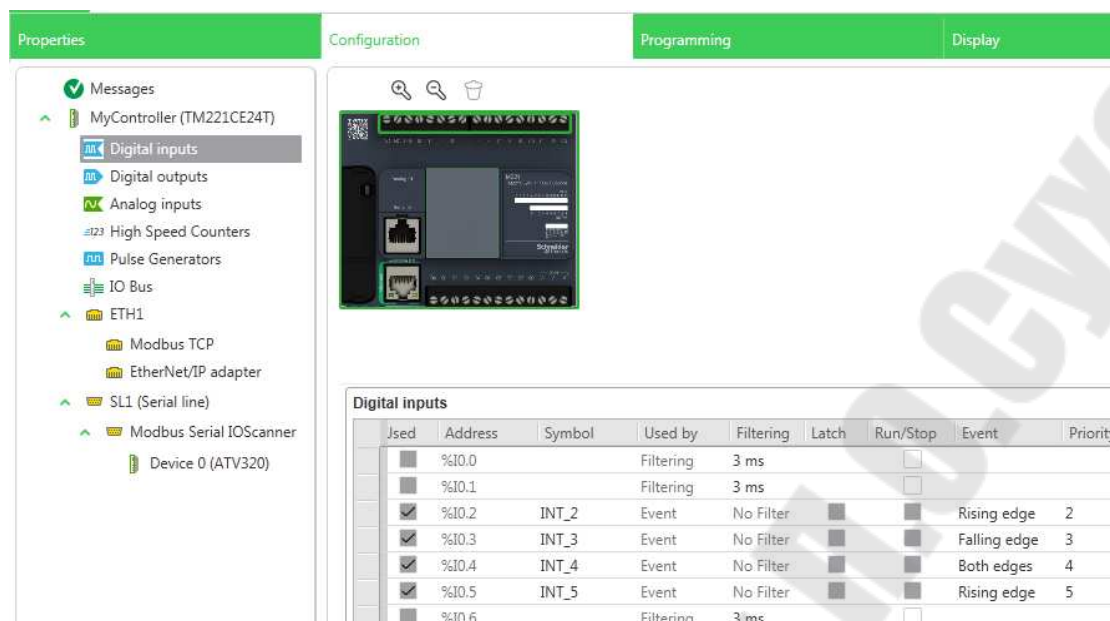


Рис. 9.5. Выбор фронта

После этого во вкладке *Programming* в задаче *Events* появятся четыре обработчика прерываний (рис. 9.6).

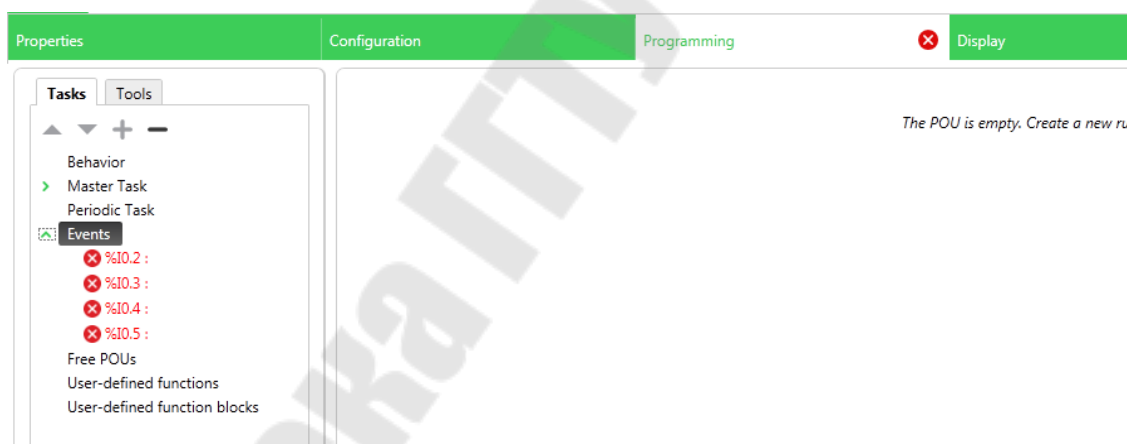


Рис. 9.6. Обработки прерываний

При нажатии правой кнопкой мыши на любом из событий %I0.2, %I0.3, %I0.4, %I0.5 появится меню, в котором можно выбрать *Add Free POU* (рис. 9.7).

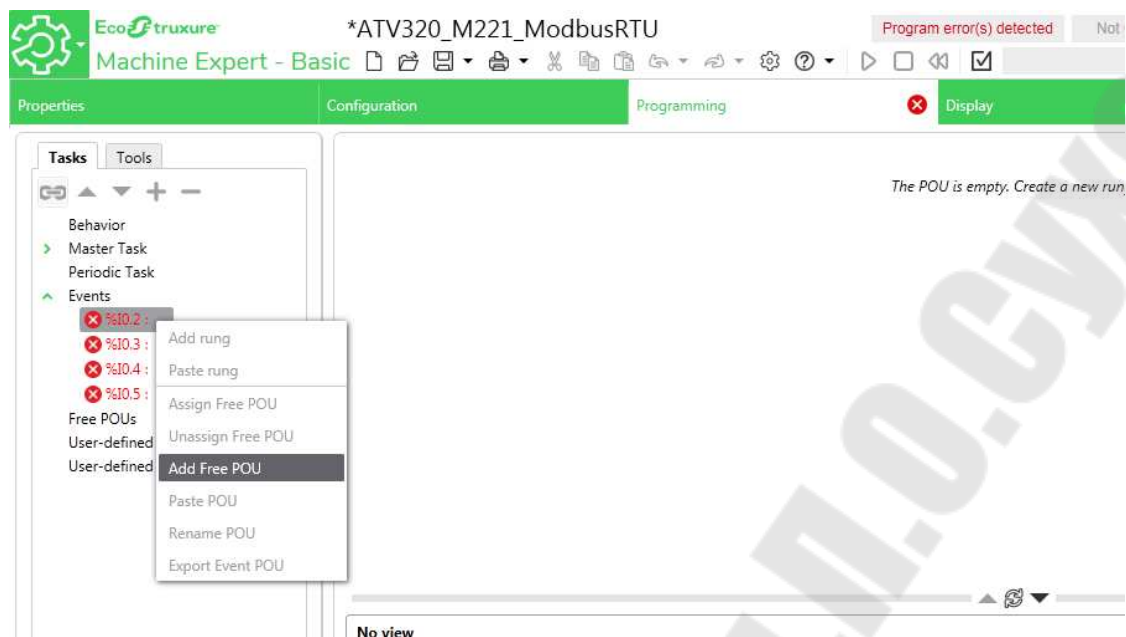


Рис. 9.7. Выбор Add Free POU

После этого можно написать программу обработчика для каждого прерывания (рис. 9.8).



Рис. 9.8. Написание программы обработчика для прерывания

10. 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. 10.1). Назовем новое окно *Int_02*.

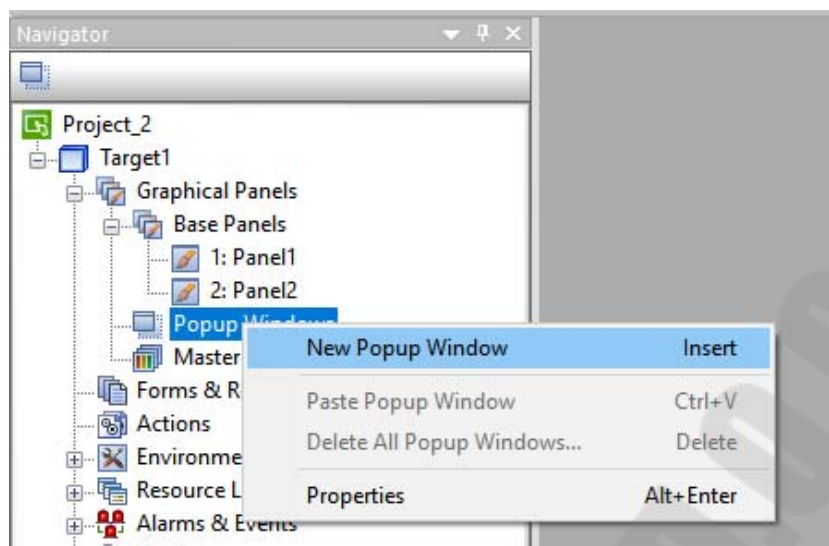


Рис. 10.1. Создание новой панели

В новом всплывающем окне по нажатию правой кнопкой мыши создадим новую панель (*New Panel*) (рис. 10.2).

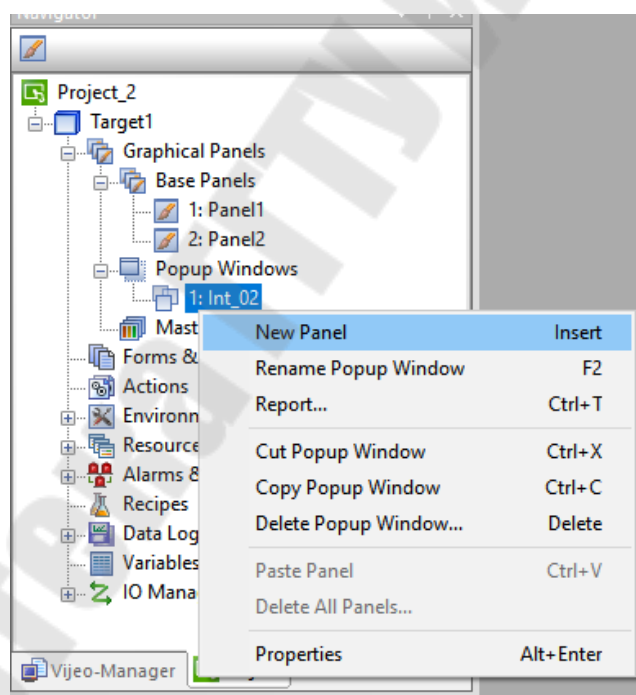


Рис. 10.2. Создание новой панели

Сделаем заливку фона вновь созданной панели красным цветом, и разместим на панели надпись «ВНИМАНИЕ! СРАБОТАЛ ИНДУКТИВНЫЙ ДАТЧИК» (рис. 10.3).

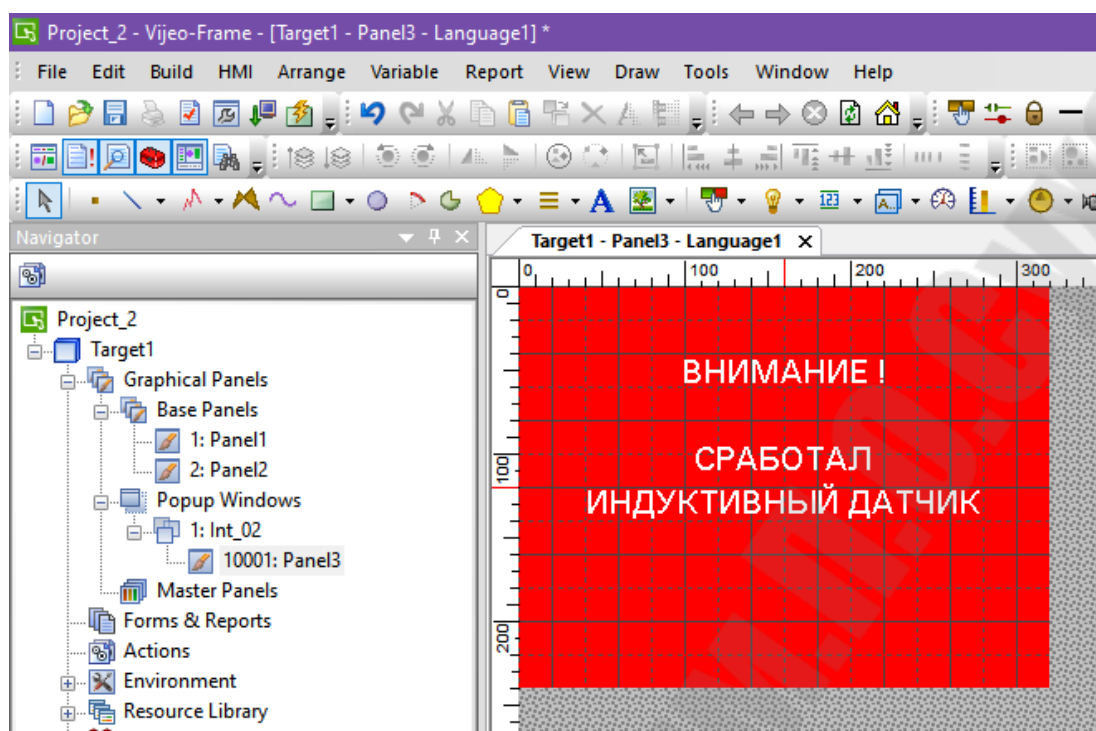


Рис. 10.3. Размещение надписи

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. 10.4).

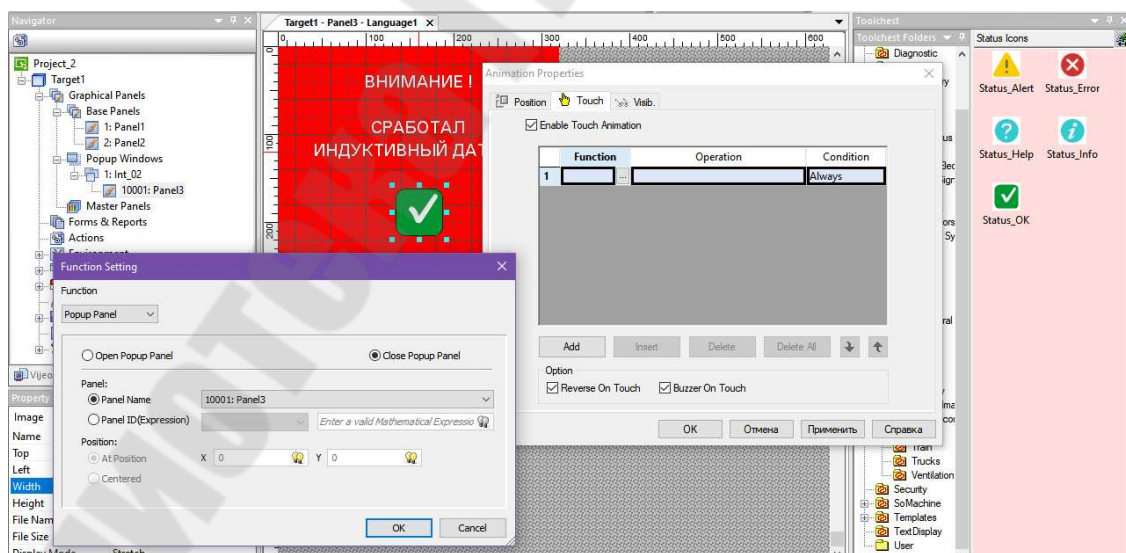


Рис. 10.4. Добавление кнопки, закрывающей окно

Добавим элементу *Status OK* первую функцию, закрывающую панель. Для этого двойным нажатием левой кнопки мыши на

элементе *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. 10.5). Для этого придется создать новую переменную типа *bool* с именем *INT_02* и адресом *%M0* на устройстве *ModbusEquipment02*.

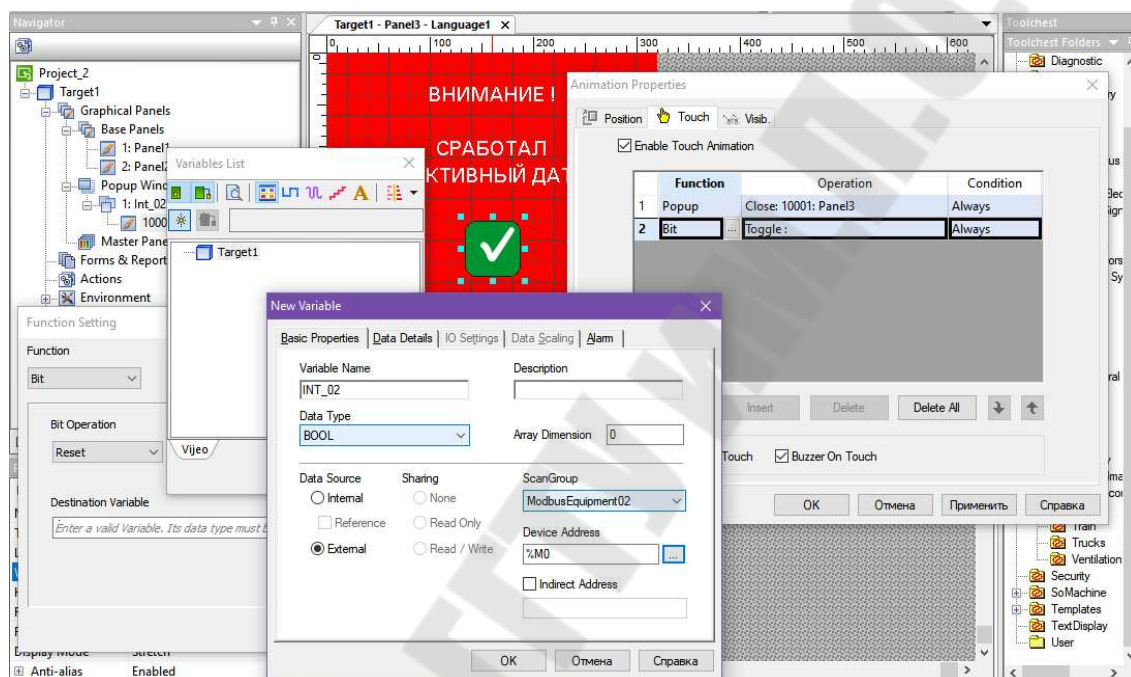


Рис. 10.5. Добавление сброса бита прерывания

Добавим экшен, запускающий всплывающий экран по условию установки бита *%M0* (рис. 10.6). Перейдем в дерево проекта в окне *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*.

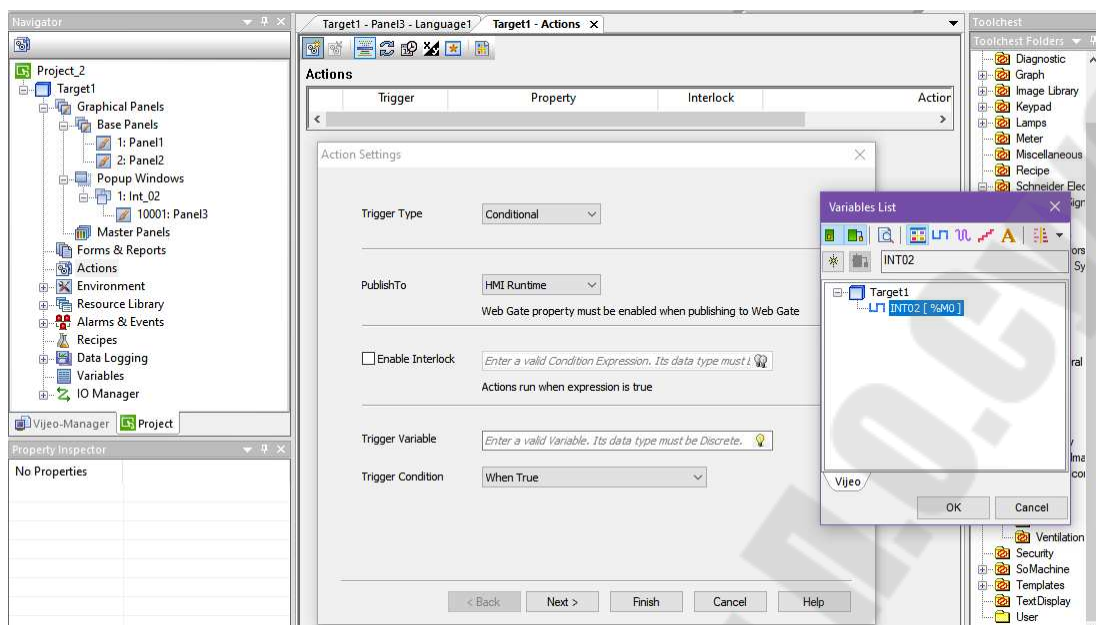


Рис. 10.6. Добавление экшена

В появившемся окне укажем тип операции (*Operation*) – всплывающее окно (*Popup*), а в качестве имени панели (*Panel Name*) выберем *Panel3* (рис. 10.7).

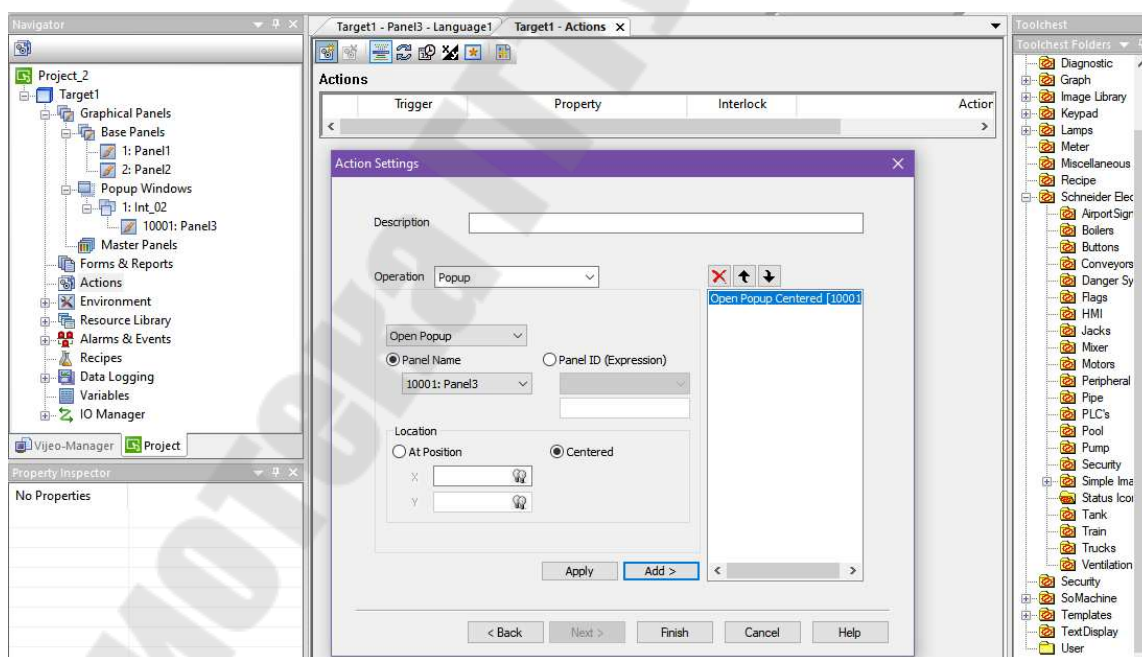


Рис. 10.7. Выбор имени панели

В Machine Expert – Basic активируем прерывания по входу *%I0.2* с критерием срабатывания по переднему фронту и приоритетом 7 (рис. 10.8).

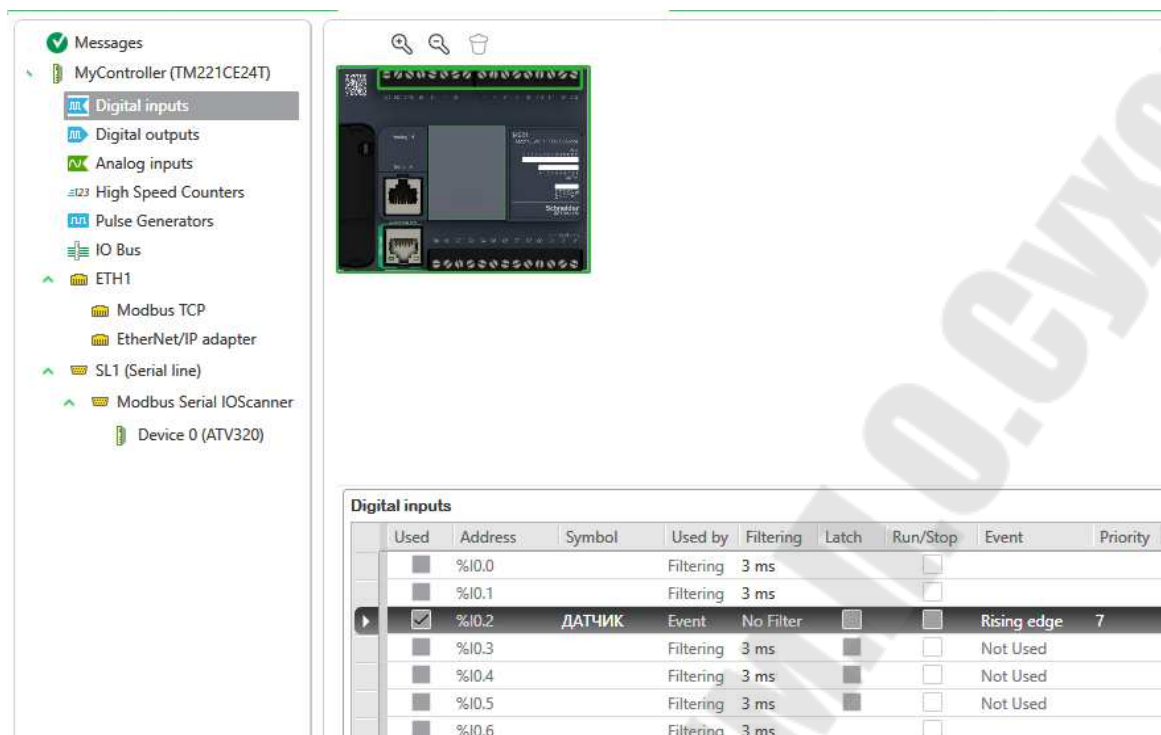


Рис. 10.8. Активизация прерывания по входу

Создадим обработчик прерывания, устанавливающий в «1» бит $\%M0$ (рис. 10.9).

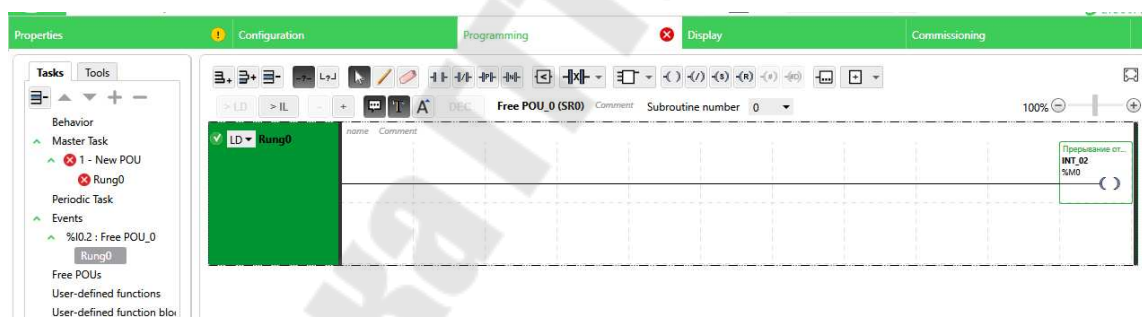


Рис. 10.9. Создание обработчика прерывания

В основной программе добавим контакт $\%I0.2$, устанавливающий бит $\%M0$ (рис. 10.10). Это необходимо для активации прерывания при срабатывании контакта $\%I0.2$.

Скомпилируем проект и запустим его симуляцию.

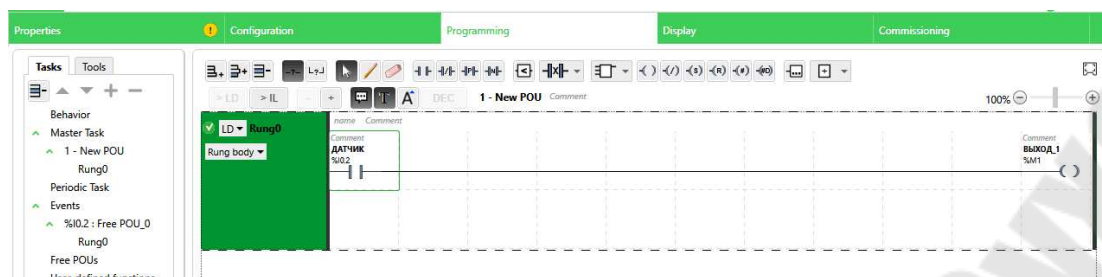


Рис. 10.10. Добавление контакта

Для совместной работы симуляторов Machine Expert – Basic и Vijeo Designer установим в последнем адрес устройства *ModbusEquipment* 127.0.0.1 (рис. 10.11).

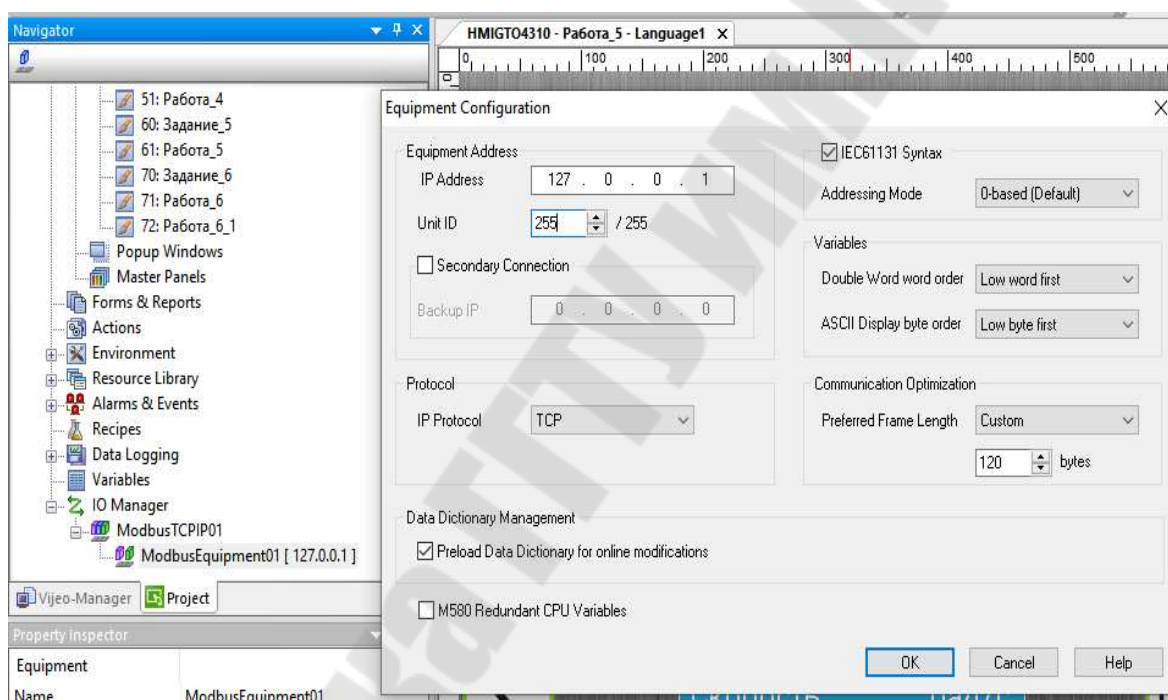


Рис. 10.11. Установка адреса устройства

На рис. 10.12 приведена симуляция проекта до срабатывания датчика – контакт %0.2 разомкнут, всплывающего окна нет.

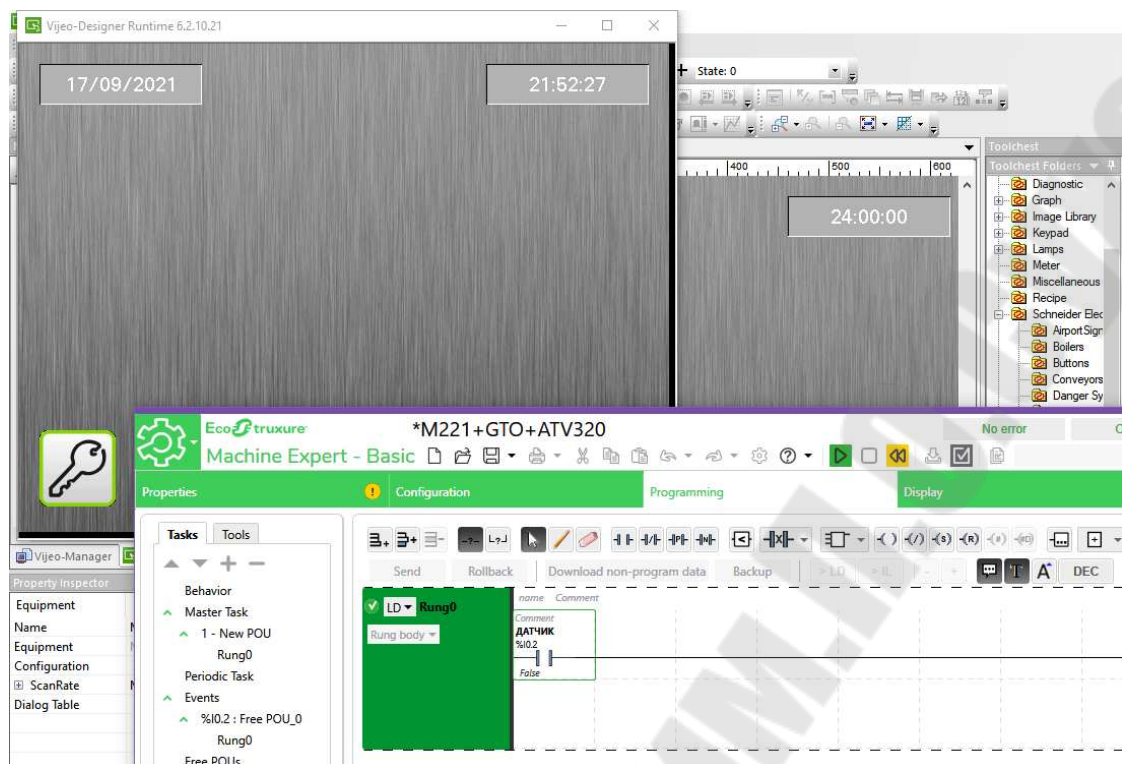


Рис. 10.12. Симуляция проекта до срабатывания датчика

На рис. 10.13 приведена симуляция проекта после срабатывания датчика – контакт %0.2 замкнут, появилось всплывающее окно.

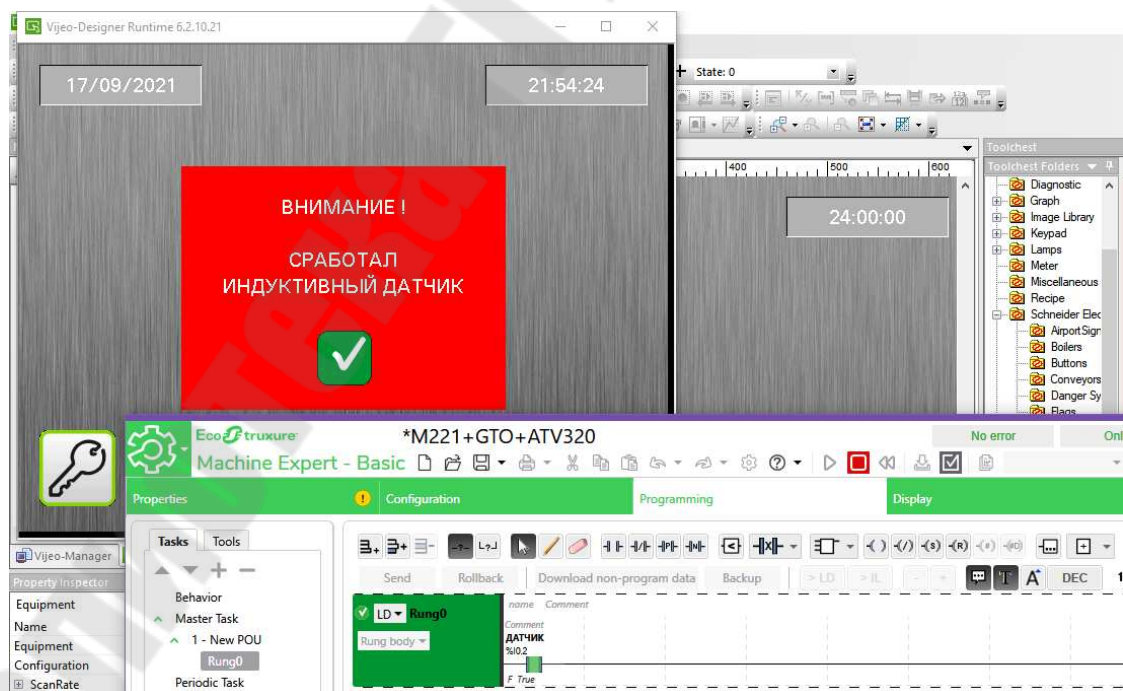


Рис. 10.13. Симуляция проекта после срабатывания датчика

11. Панель ввода значений

Создадим панель ввода значений параметров.

Вначале создадим внешние переменные типа INT (16 бит со знаком) с именами «*Параметр_1*», «*Параметр_2*» и т. д., а также внутренние переменные терминала с именами «*Номер_параметра*» и «*Заданное_значение*» (рис. 11.1).

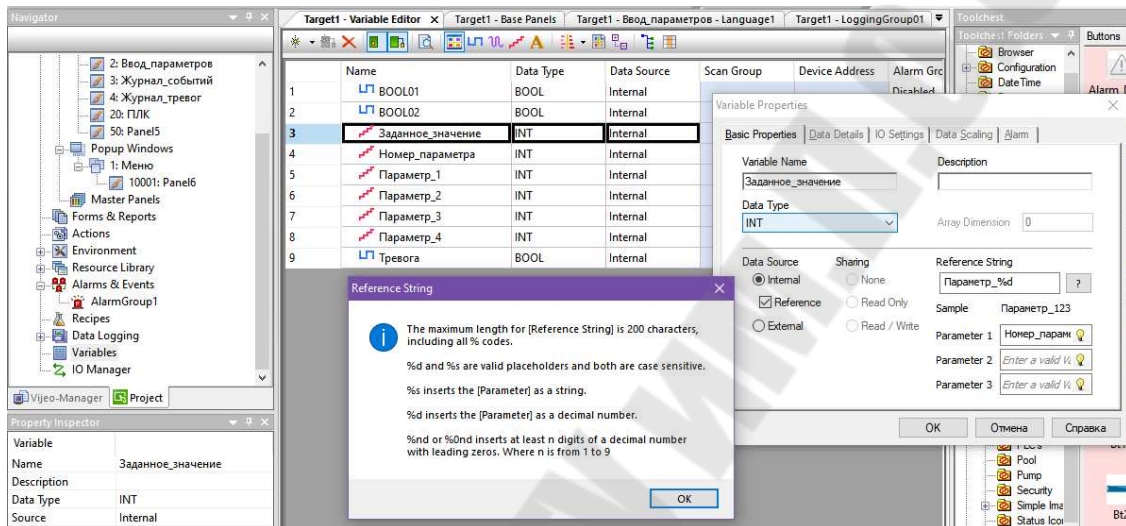


Рис. 11.1. Создание переменной типа INT

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*».

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. 11.2).

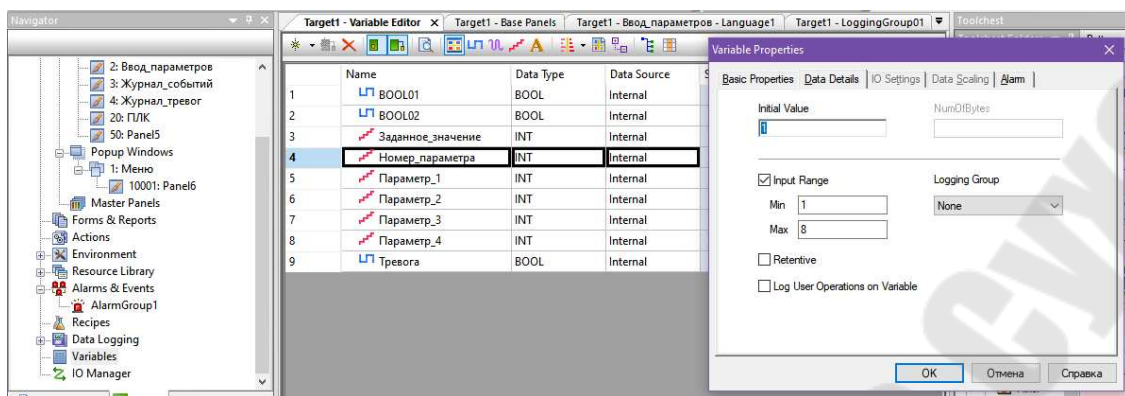


Рис. 11.2. Окно свойств переменных

С помощью встроенного графического редактора создадим панель ввода параметров (рис. 11.3).

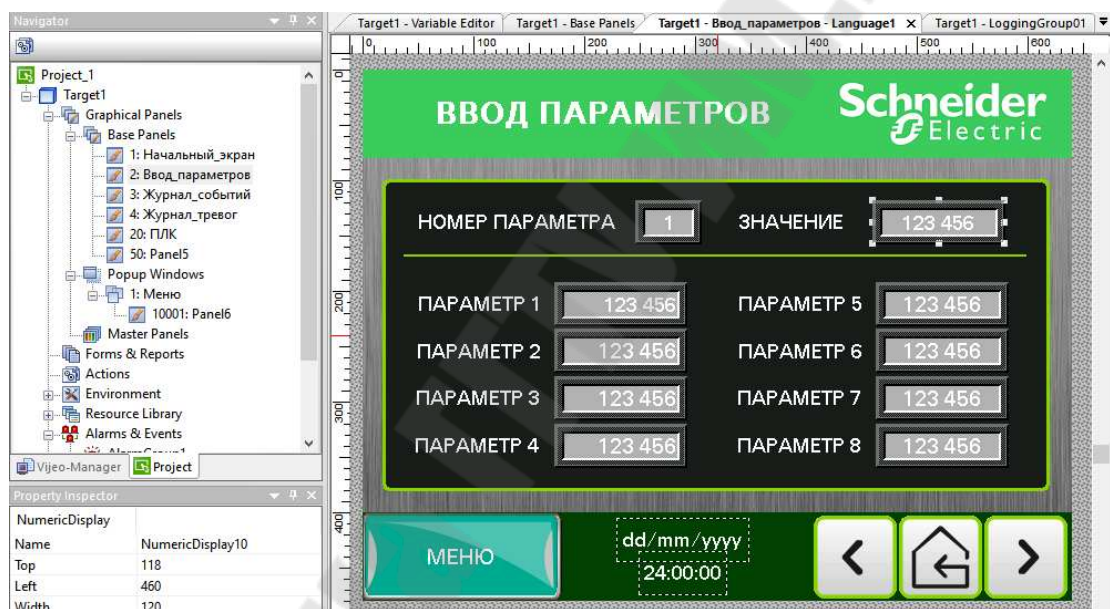


Рис. 11.3. Создание панели ввода параметров

Выведем на экран ПТ сообщение, содержащее значение переменной. Для этого создадим текстовый ресурс, содержащий сообщение для журнала событий (рис. 11.4). 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.

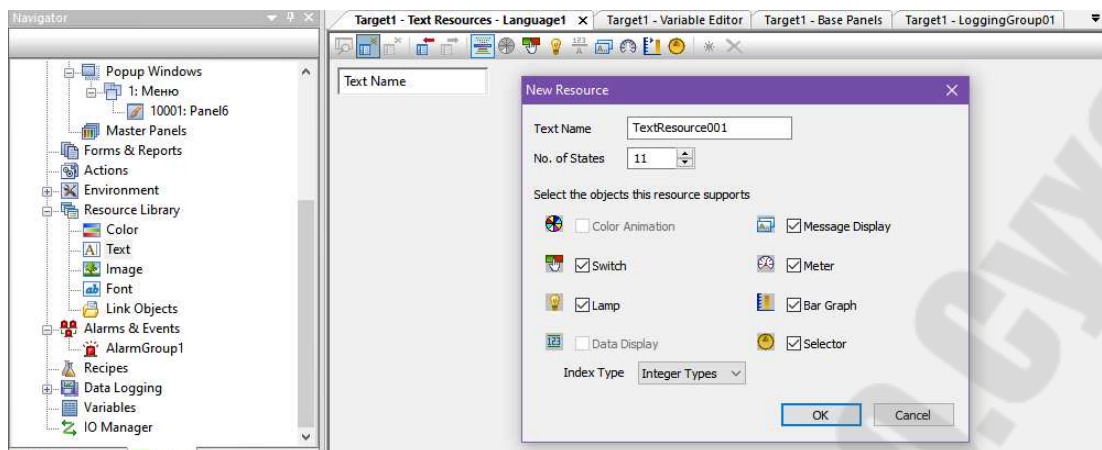


Рис. 11.4. Создание текстового ресурса

Введем текст сообщения: «Установлено значение параметра 1:» (рис. 11.5). Не выходя из режима редактирования текста, нажатием правой кнопкой мыши на строке с текстом вызовем всплывающее меню, где выберем пункт *Insert -> Variable...* и выберем отображаемую переменную «*Параметр_1*».

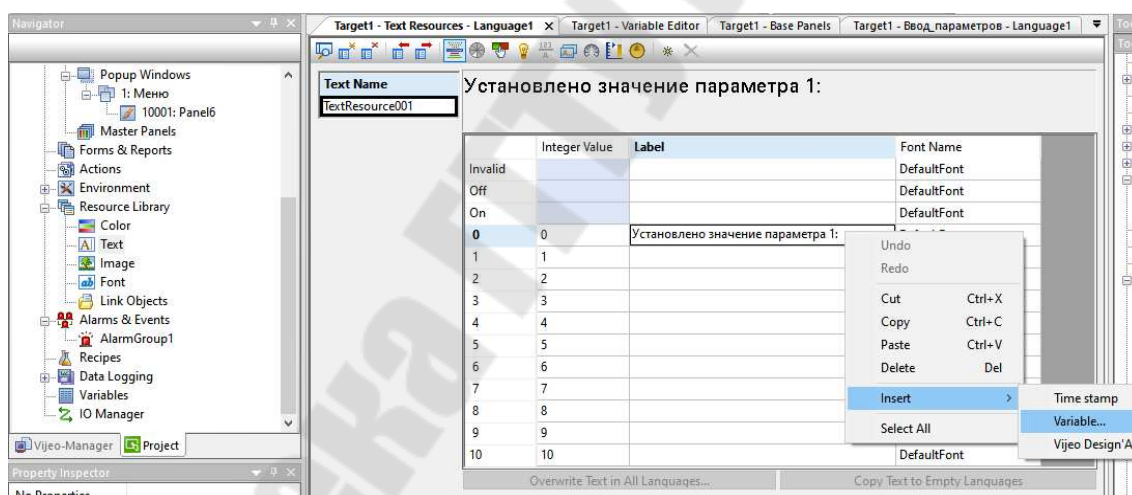


Рис. 11.5. Ввод текста сообщения

Создадим группу событий «*EventGroup1*» в разделе тревог и событий (*Alarms & Events*) дерева проекта (рис. 11.6). Для этого выберем пункт всплывающего меню *New Event Group -> Event Group*.

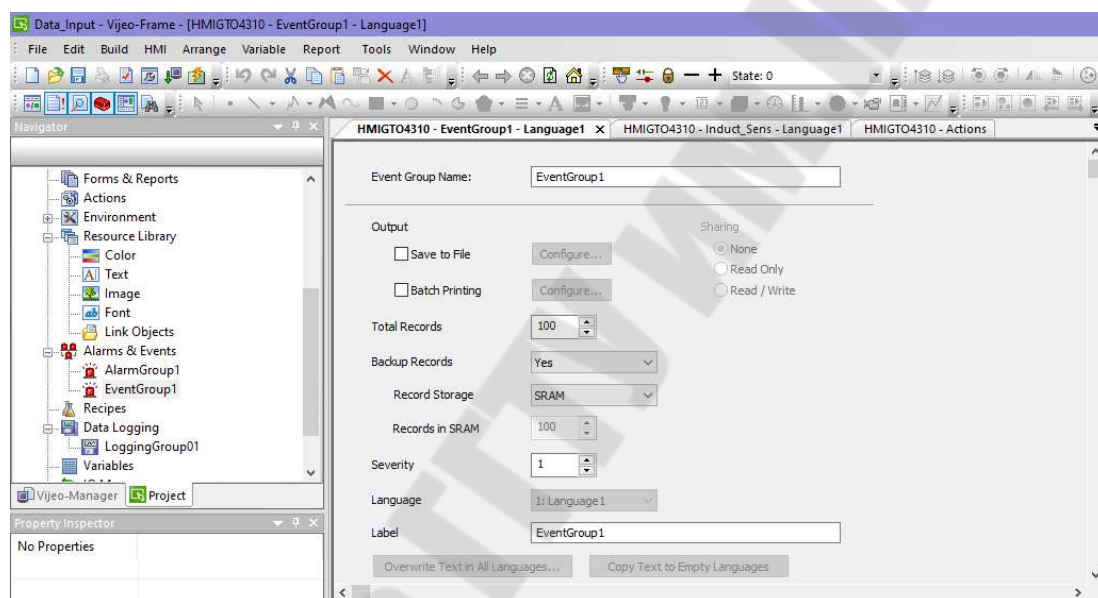
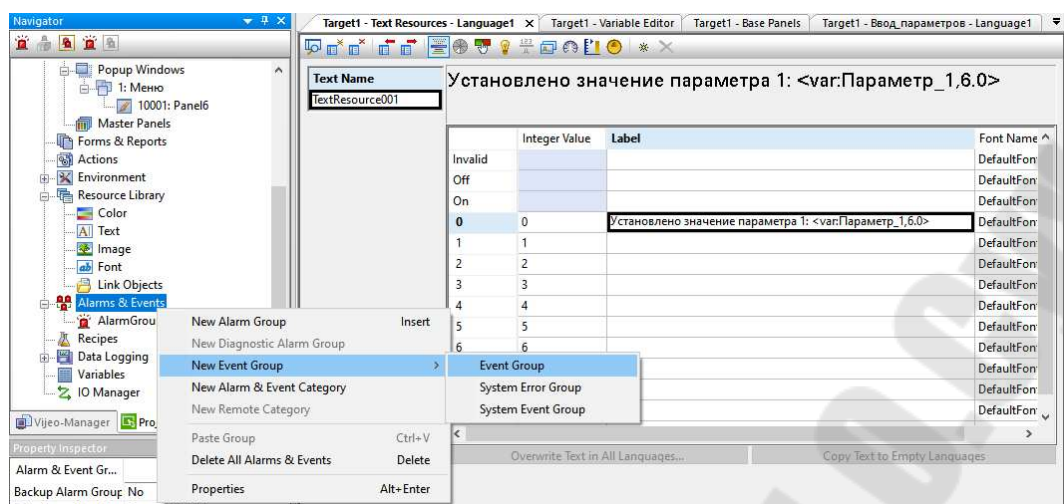


Рис. 11.6. Создание группы событий

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. 11.7). 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*.

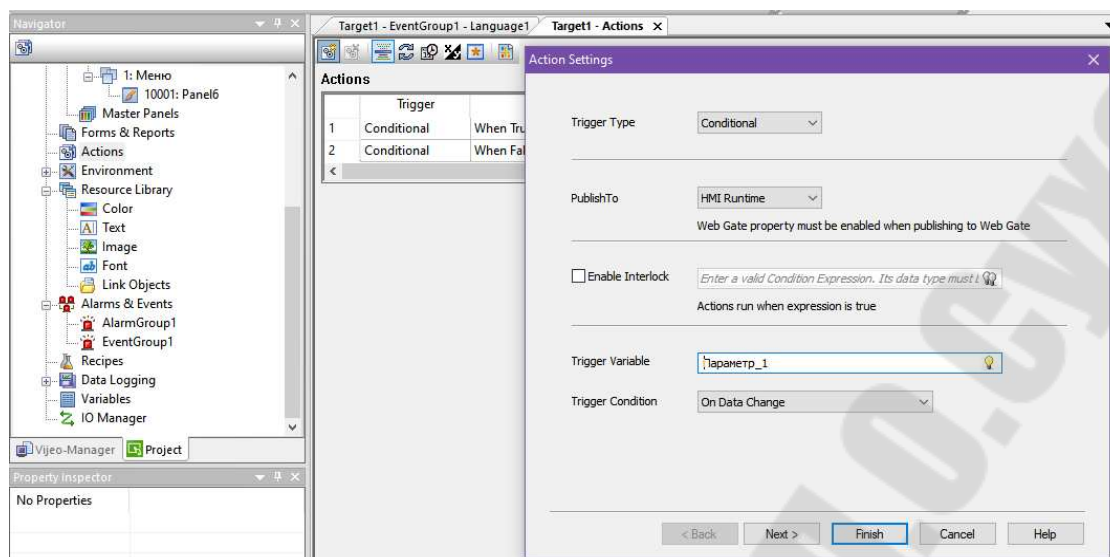


Рис. 11.7. Выбор раздела в дереве проекта

В появившемся окне в качестве операции (*Operation*) выберем «Сообщение о событии» (*Event Message*) (рис. 11.8). В качестве выводимого сообщения (*Text Resource*) выберем *TextResource001*, обязательно нажмем кнопку «Добавить» (*Add*) и «Завершить» (*Finish*).

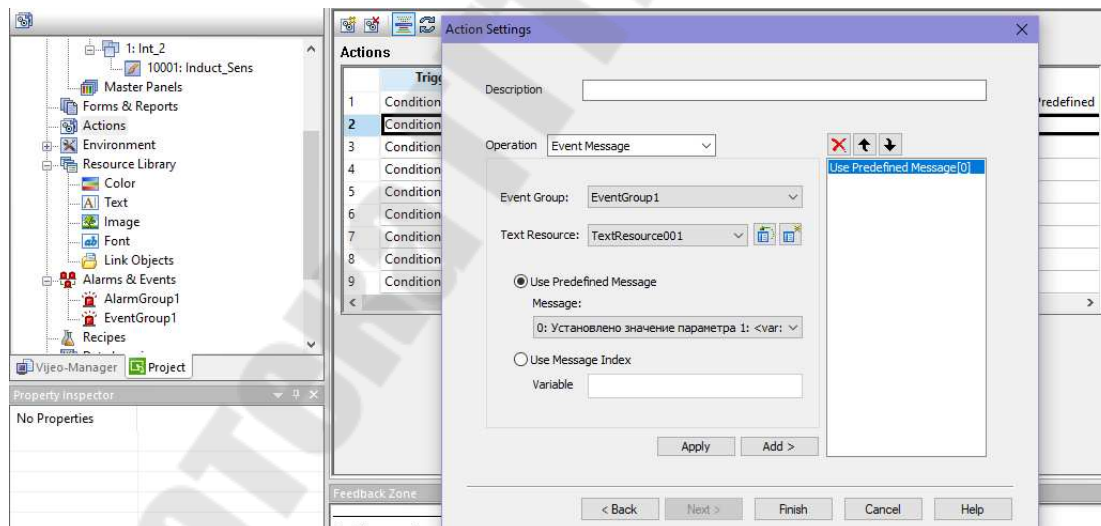


Рис. 11.8. Выбор операции

Аналогичным образом создадим экшны для оставшихся переменных (рис. 11.9).

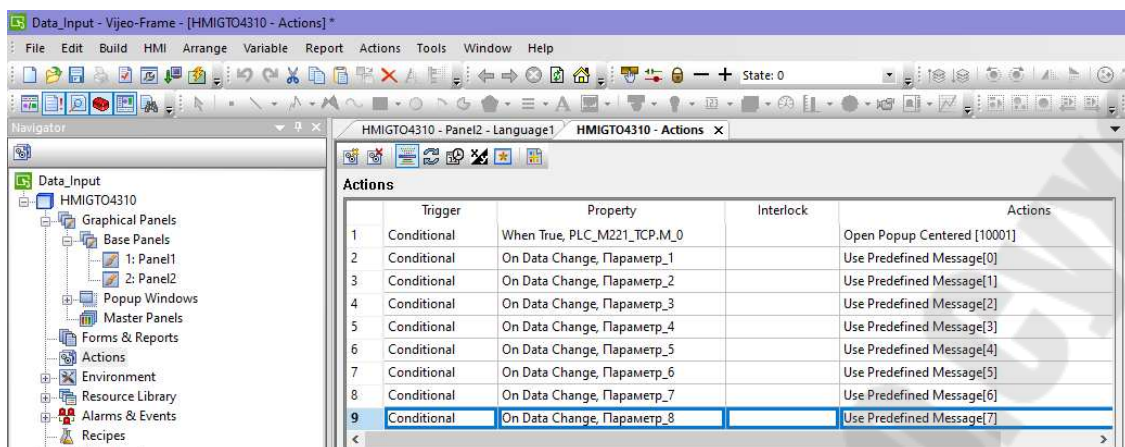


Рис. 11.9. Результат создания экшенов

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. 11.10). 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*).

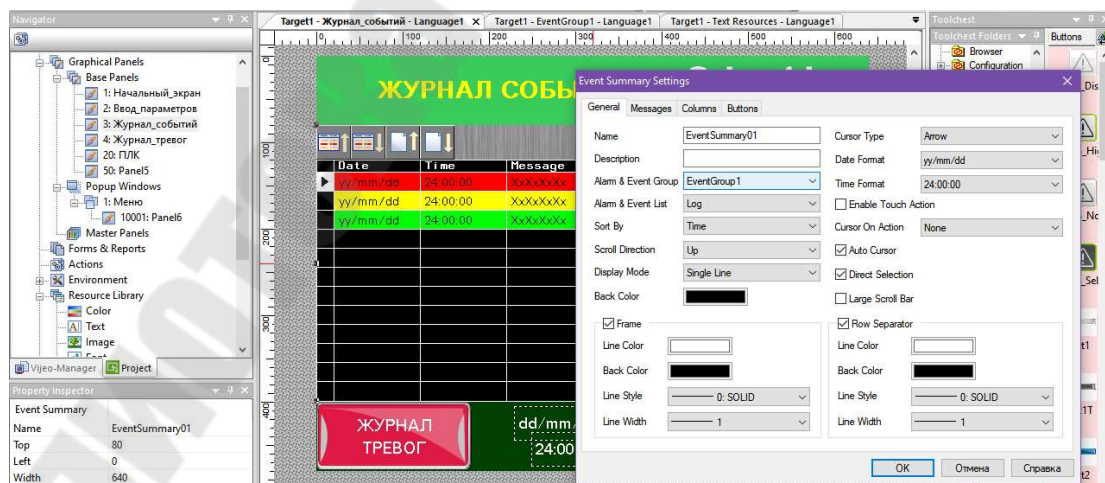


Рис. 11.10. Создание панели раздела дерева

Результат можно посмотреть в симуляторе (рис. 11.11).

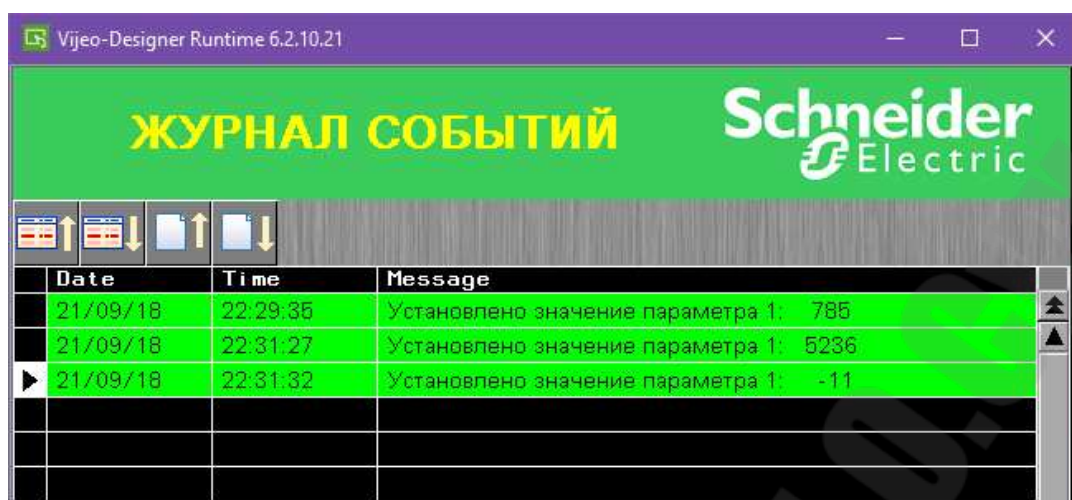


Рис. 11.11. Журнал событий в симуляторе

12. Создание журнала тревог

В окне навигатора проекта (*Navigator*) нажатием правой кнопки мыши на элементе *Base Panels* в дереве проекта вызовем всплывающее меню, в котором выберем *New Panel* (рис. 12.1). Назовем новую панель «Журнал тревог».

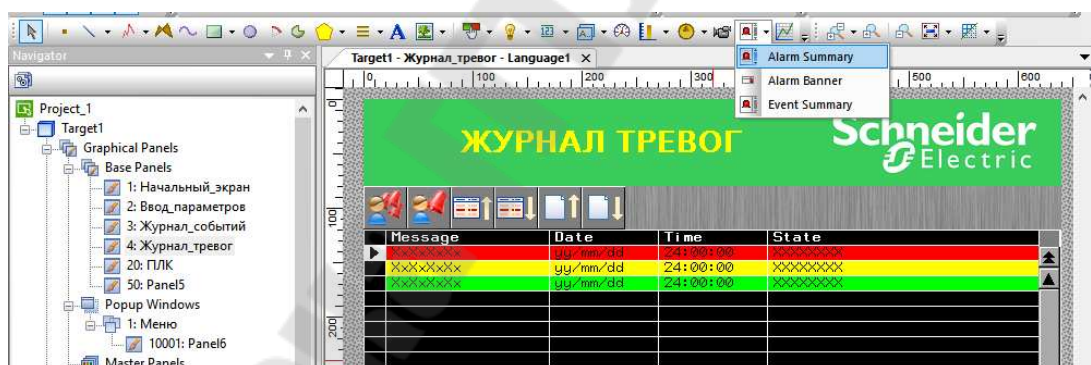


Рис. 12.1. Вызов всплывающего меню

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

Выберем в дереве проекта раздел переменных (*Variables*) и создадим переменную «Тревога» типа *bool* (рис. 12.2). 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».

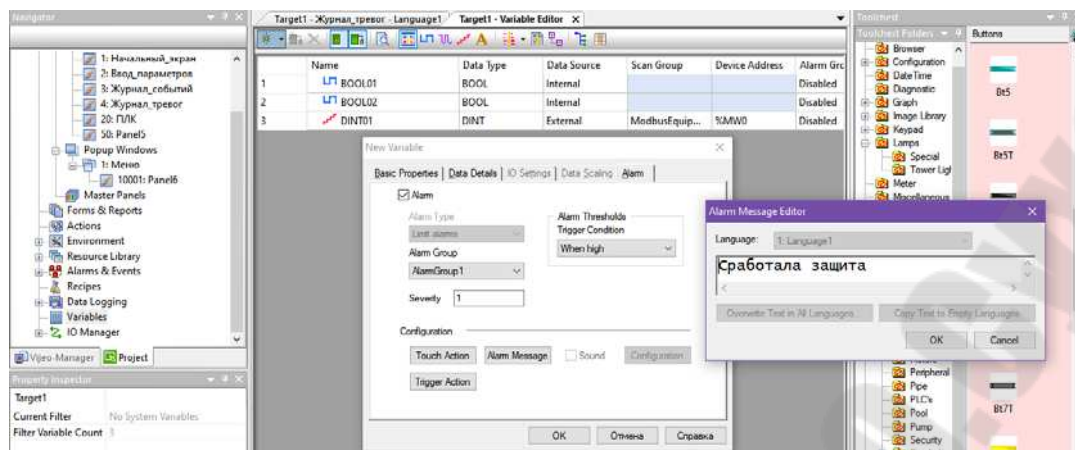


Рис. 12.2. Создание переменной типа *bool*

В дереве проекта выберем раздел *Alarm & Events* и добавим группу *AlarmGroup1*. В группе *AlarmGroup1* создадим внутреннюю переменную типа *bool* с именем «Тревога» (рис. 12.3).

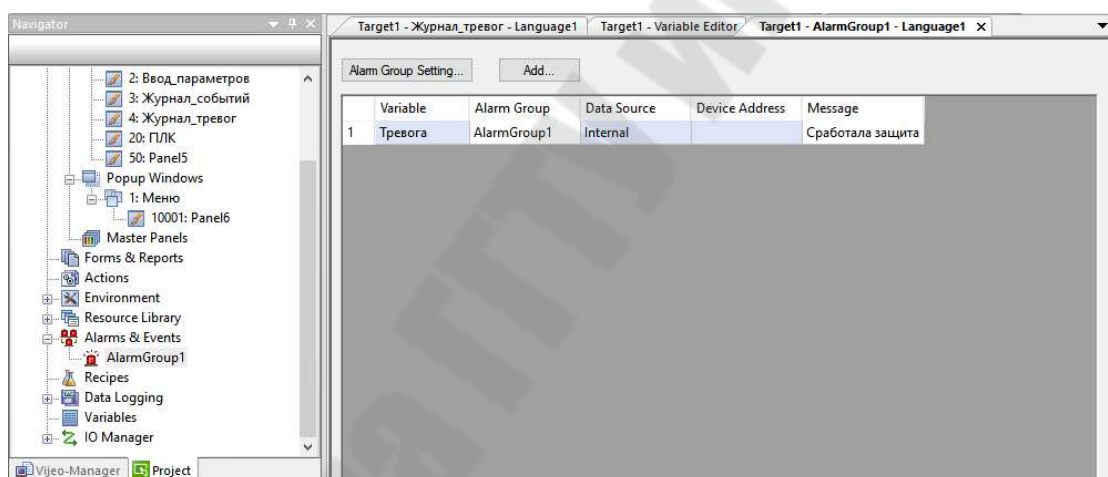


Рис. 12.3. Создание внутренней переменной

Двойным кликом по таблице тревог вызываем всплывающее окно настроек *Alarm Summary Settings*, в котором устанавливаем параметр *Alarm & Events List* в значение *Log* (журнал) (рис. 12.4).

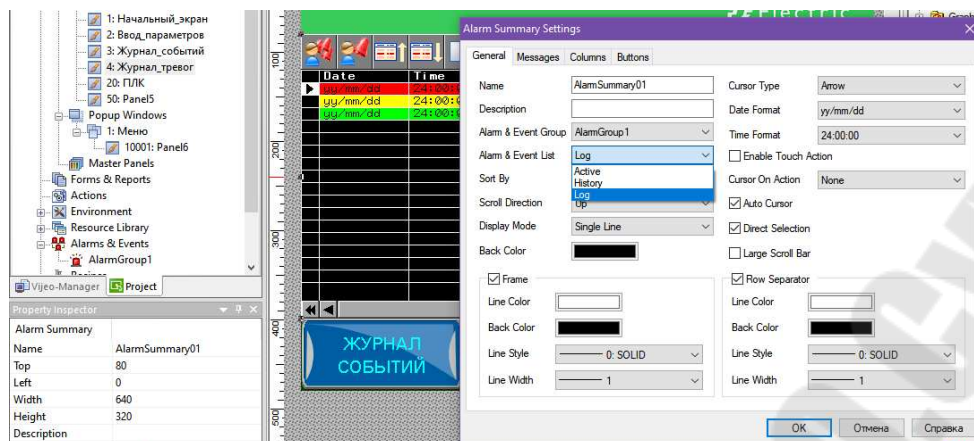


Рис. 12.4. Установка параметра

13. 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. 13.1). 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.

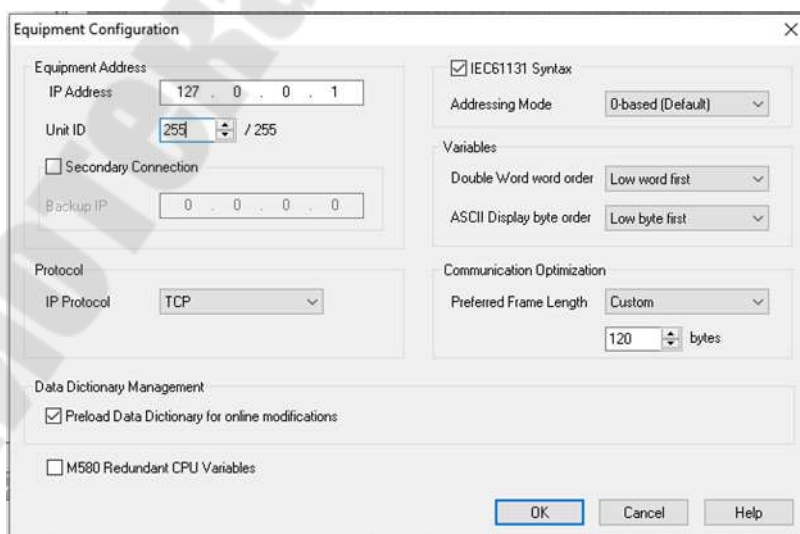


Рис. 13.1. Выбор раздела и подраздела

Создадим внешнюю переменную типа *bool* с именем «*PLC_RUN_STOP*» с параметром *%S12*.

Также создадим 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. 13.2).

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 1](#)

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 bottom part of the image shows the Variable Editor window with a list of variables. A dialog box is open for creating a new variable named *BAD* of type *INT*, with data source *External* and device address *@stat:1*.

Рис. 13.2. 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. 13.3).

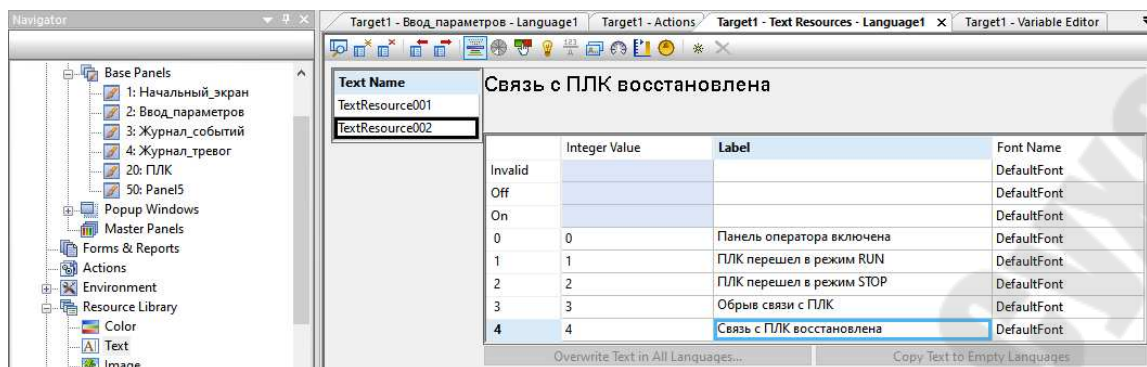


Рис. 13.3. Создание текстового ресурса

Создадим группу событий *EventGroup1*. Создадим несколько экшонов.

Первый экшен выводит сообщение о запуске панели оператора. Для этого в окне навигатора (*Navigator*) в дереве проекта выберем раздел *Actions* (рис. 13.4). 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*.

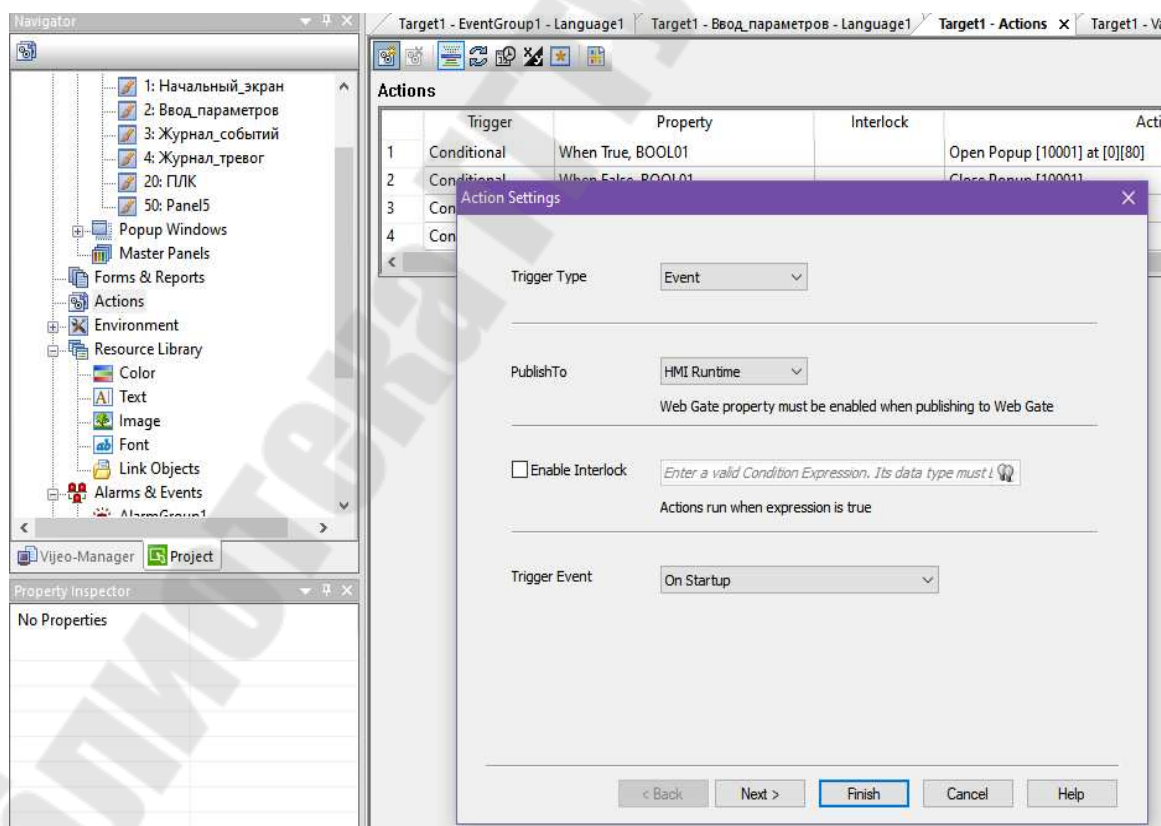


Рис. 13.4. Выбор раздела

В появившемся окне в качестве операции (*Operation*) выберем «Сообщение о событии» (*Event Message*) (рис. 13.5). 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*).

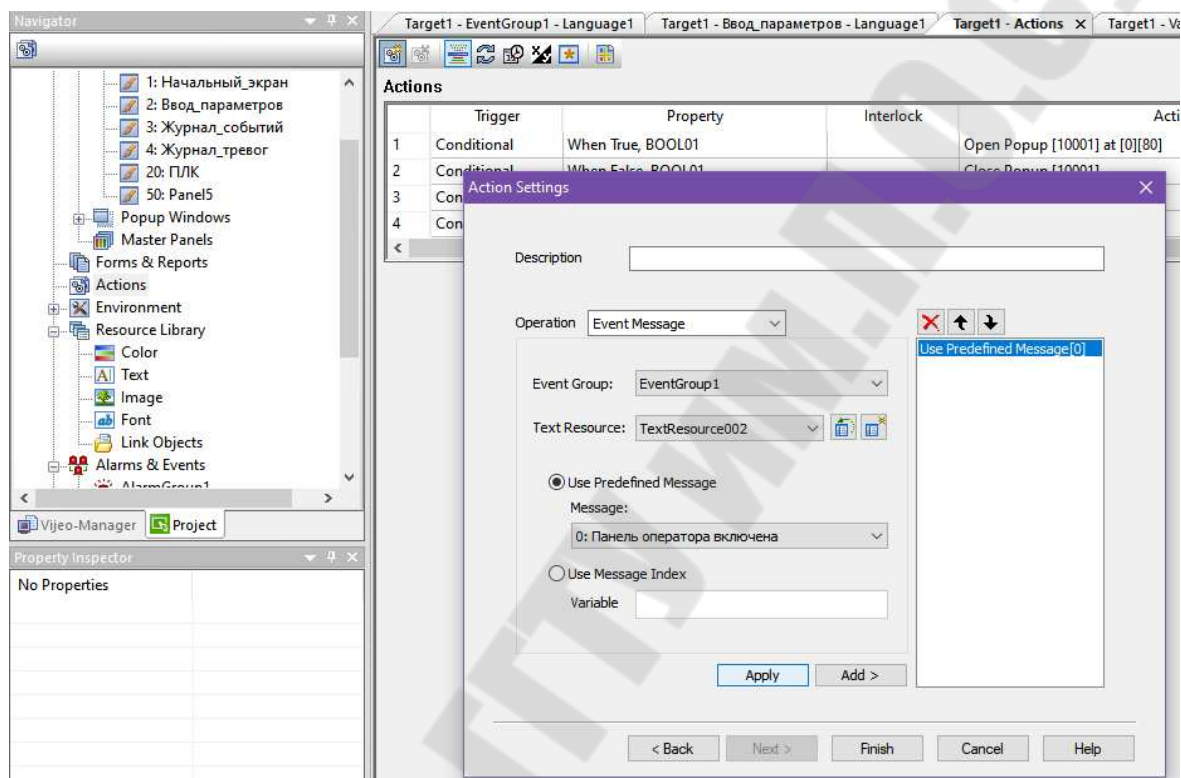


Рис. 13.5. Выбор операции

Второй и третий экшны выводят сообщения о переходе ПЛК в режим *RUN* и *STOP* соответственно (рис. 13.6 и 13.7).

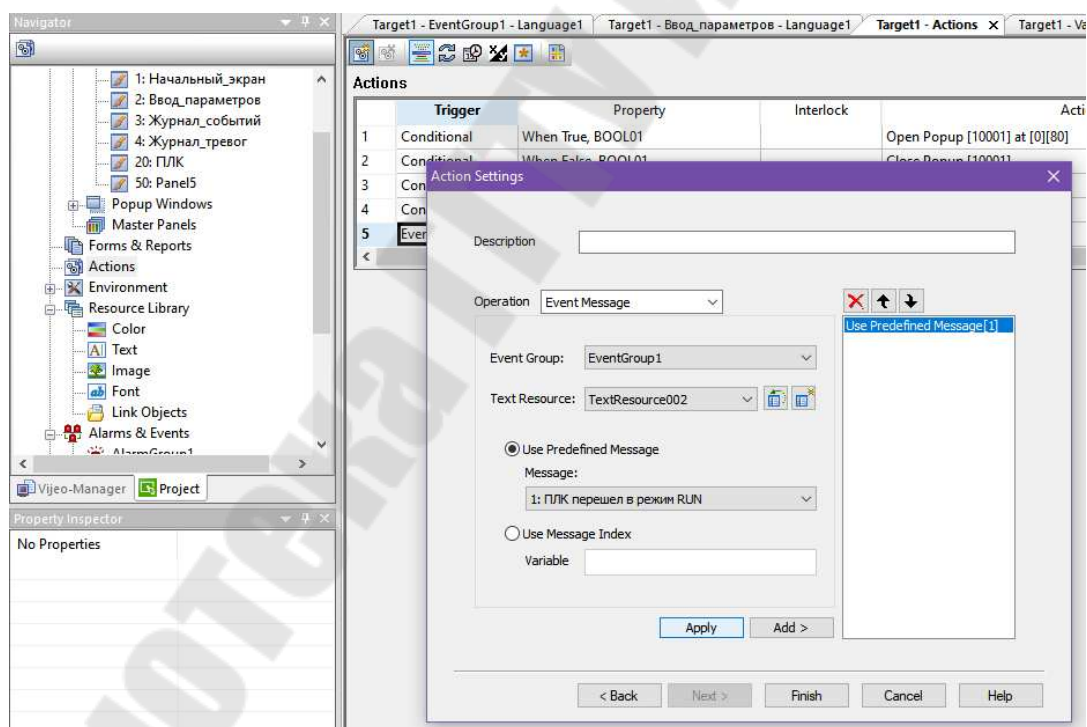
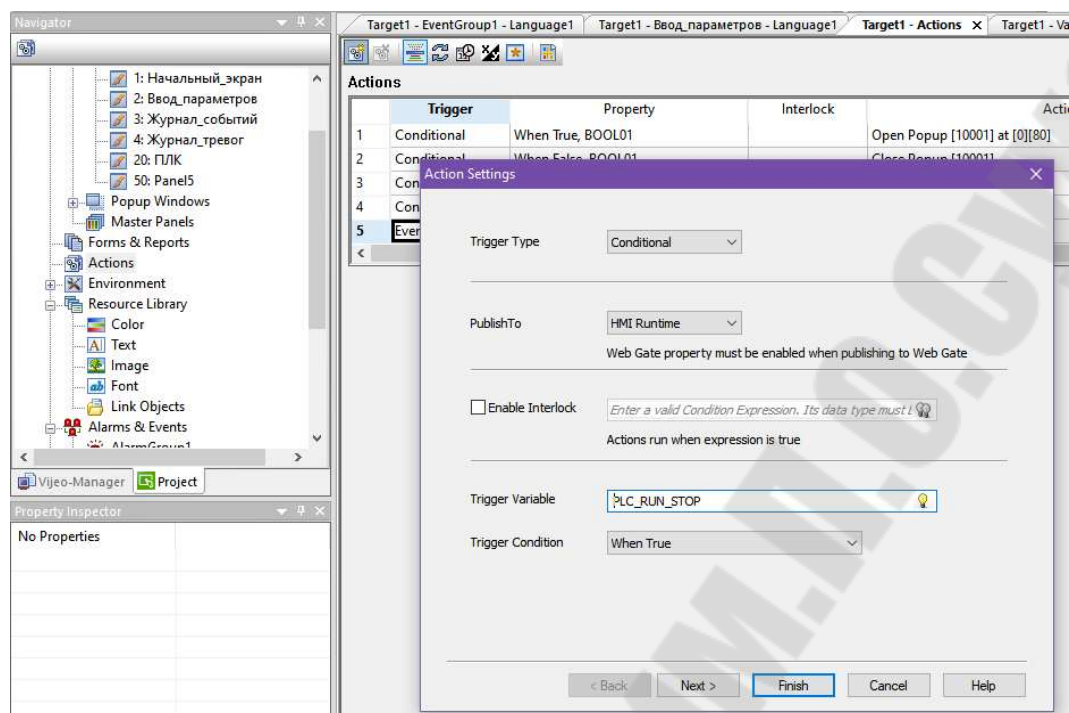


Рис.13.6. Переход программирующего логического контроллера в режим RUN

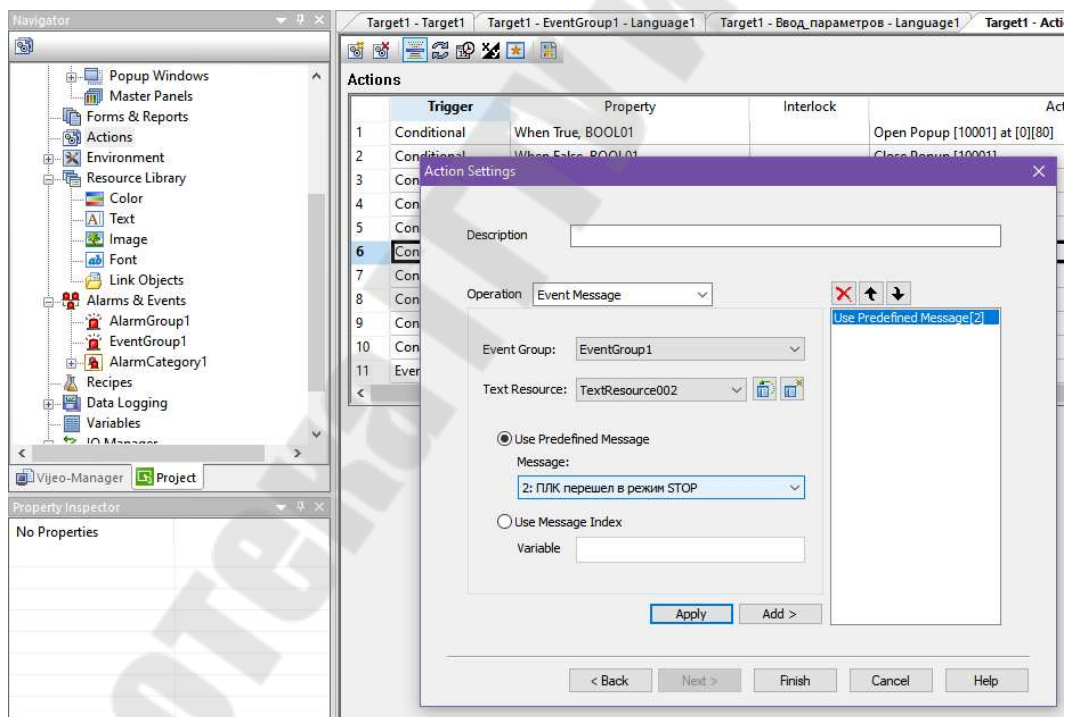
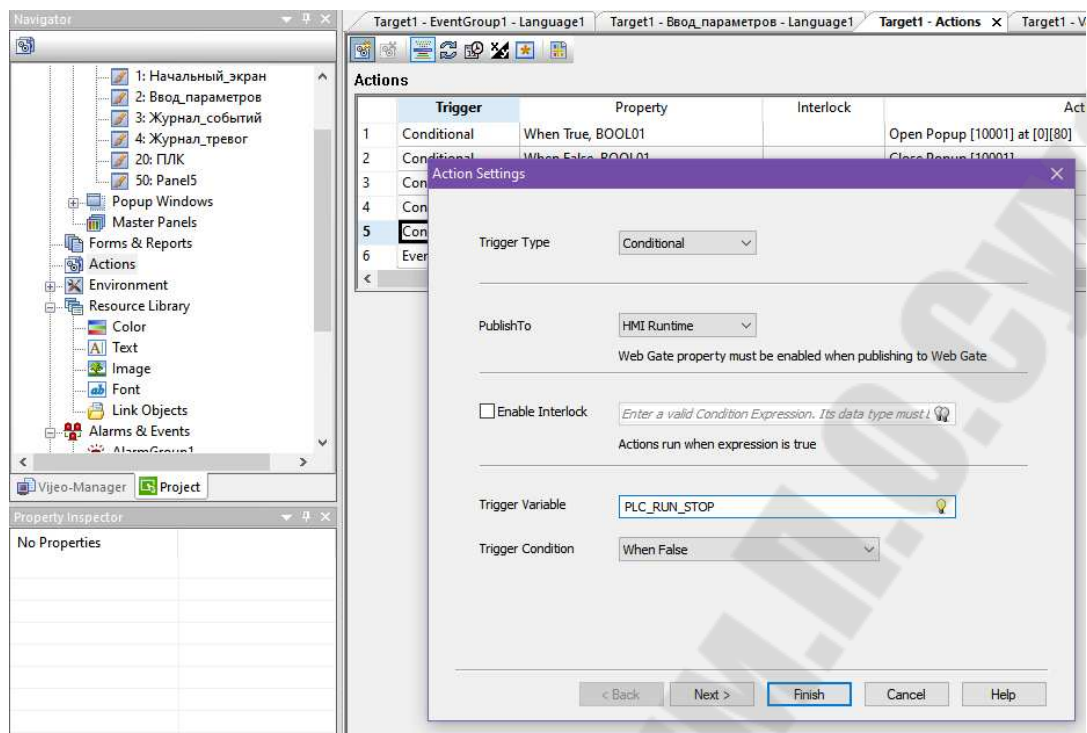


Рис. 13.7. Переход программируемого логического контроллера в режим STOP

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. 13.8–13.11).

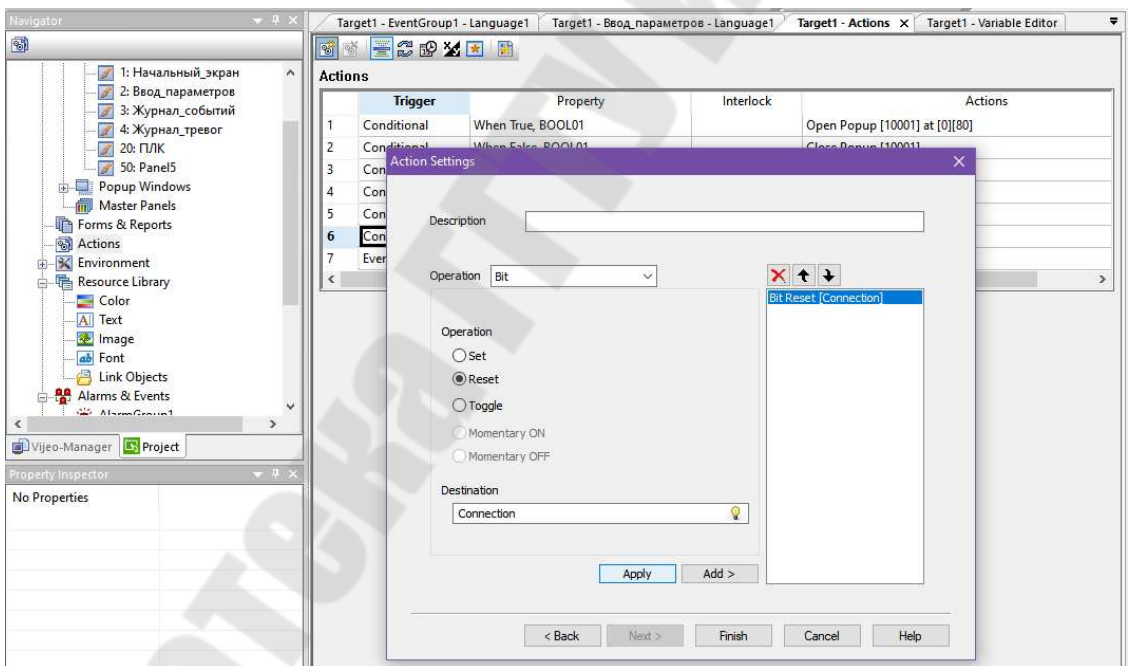
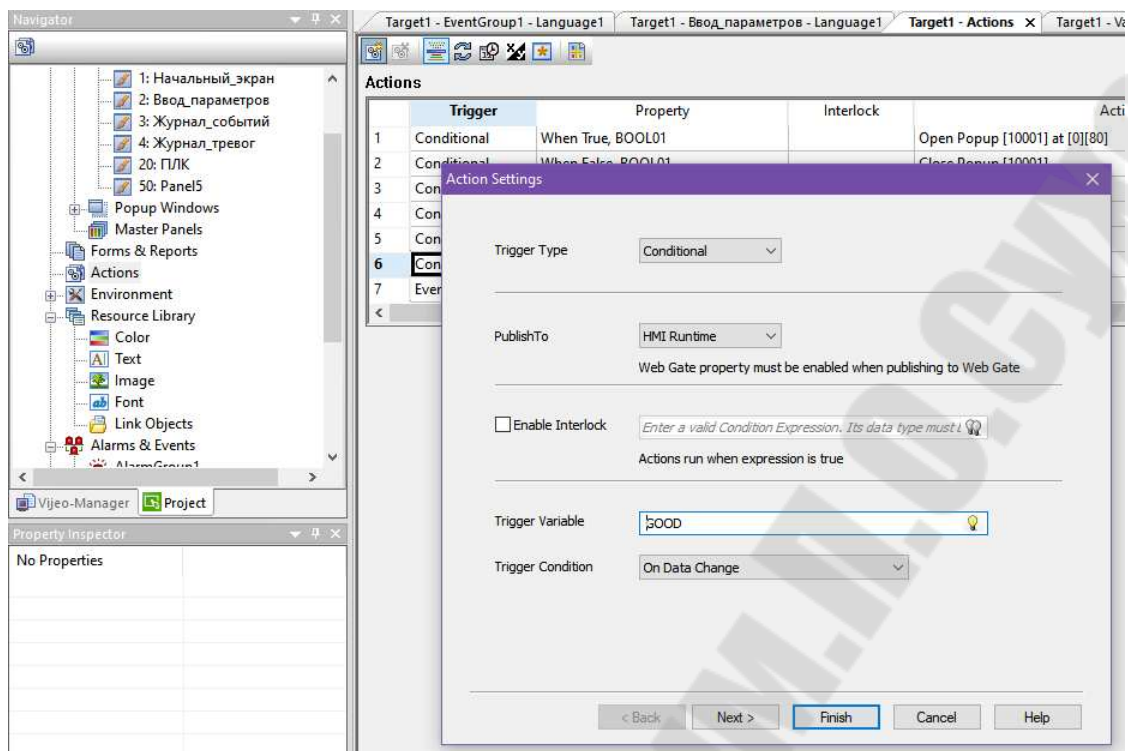


Рис. 13.8. Создание переменной GOOD

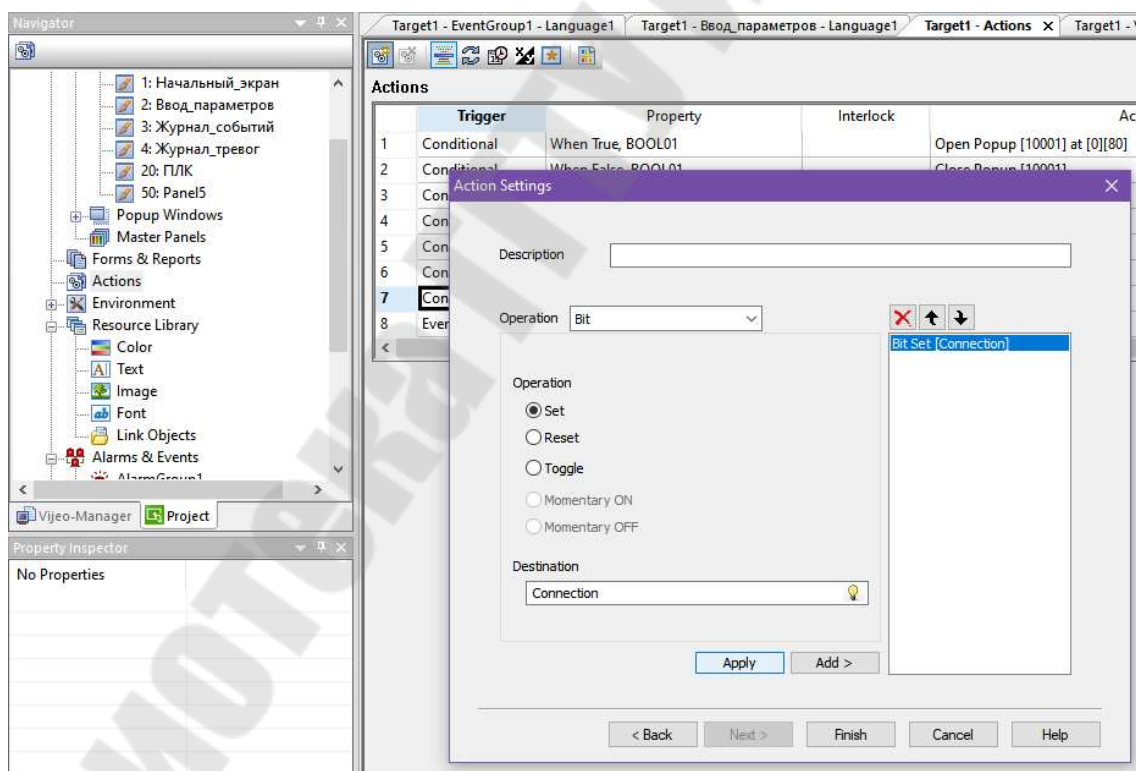
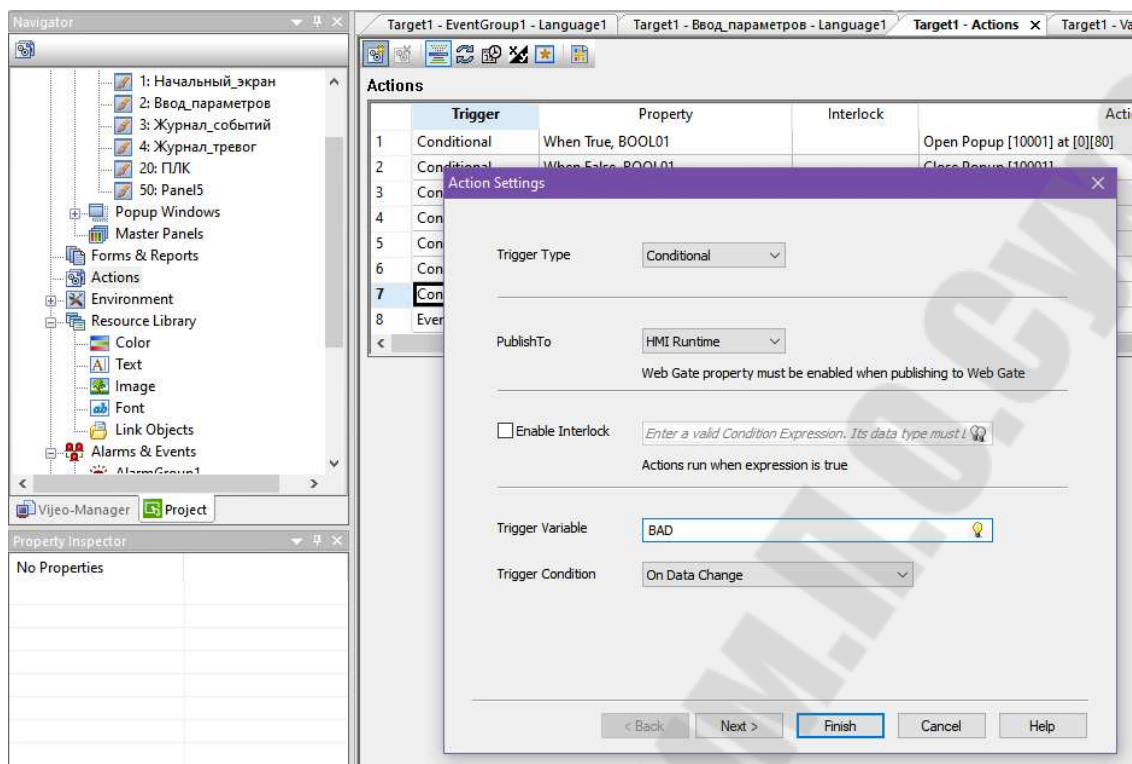


Рис. 13.9. Установка переменной BAD

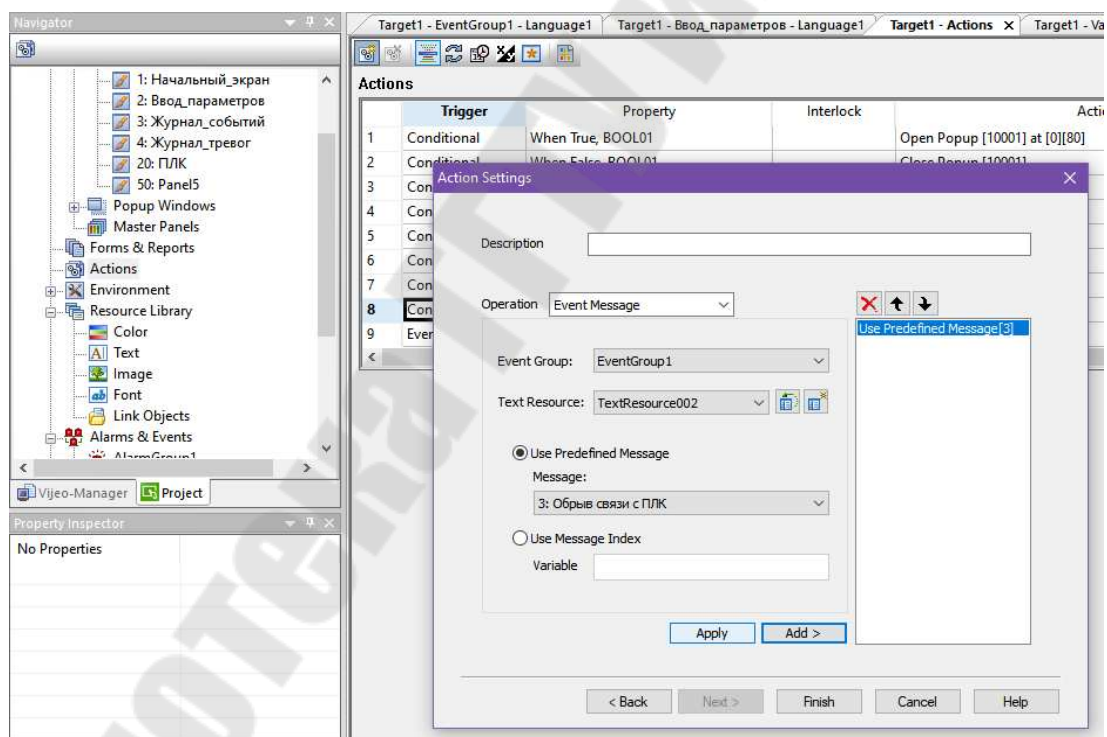
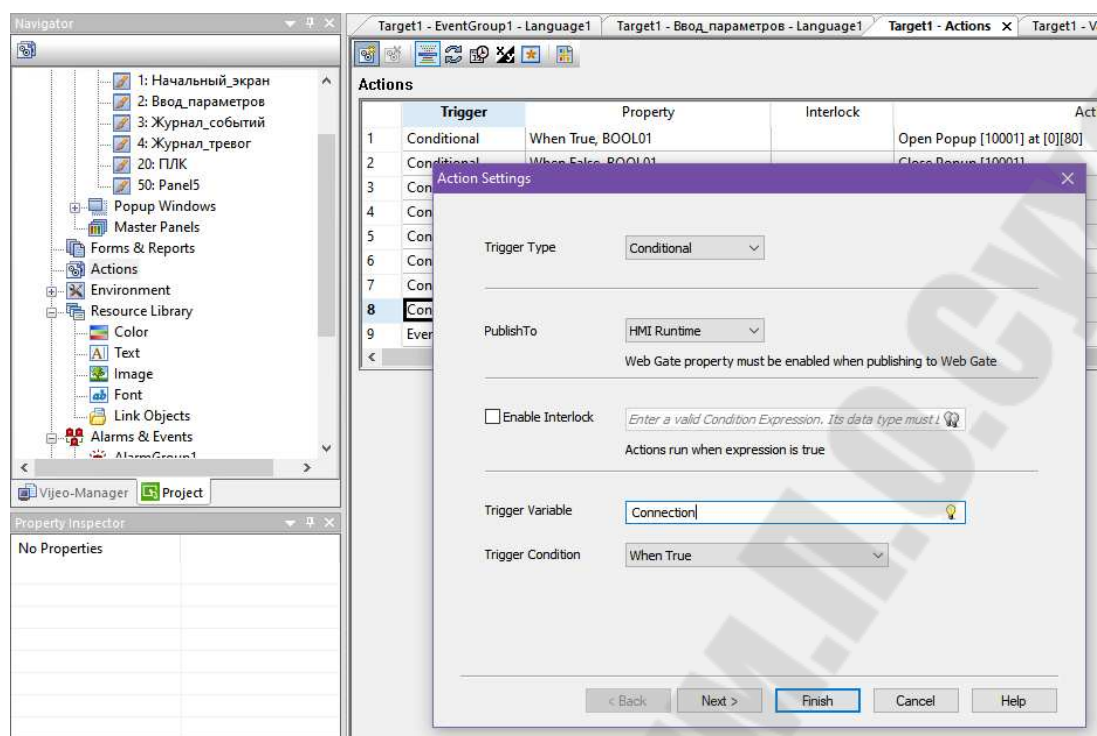


Рис. 13.10. Установка переменной Connection

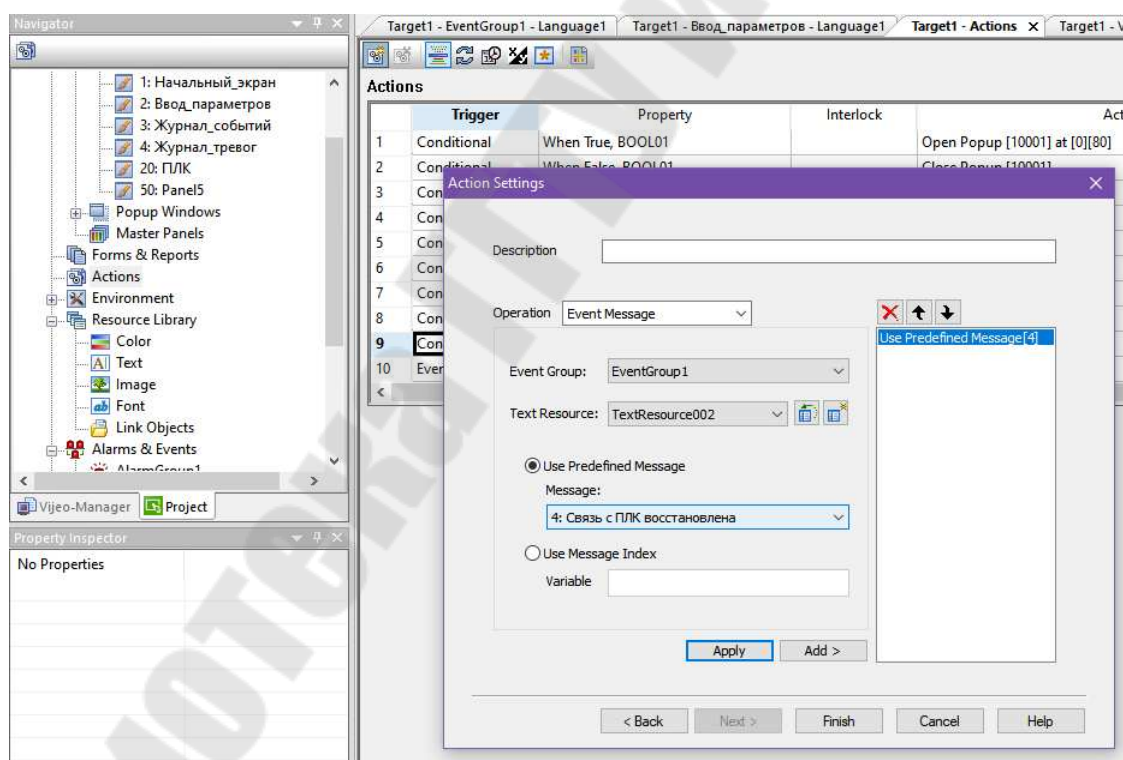
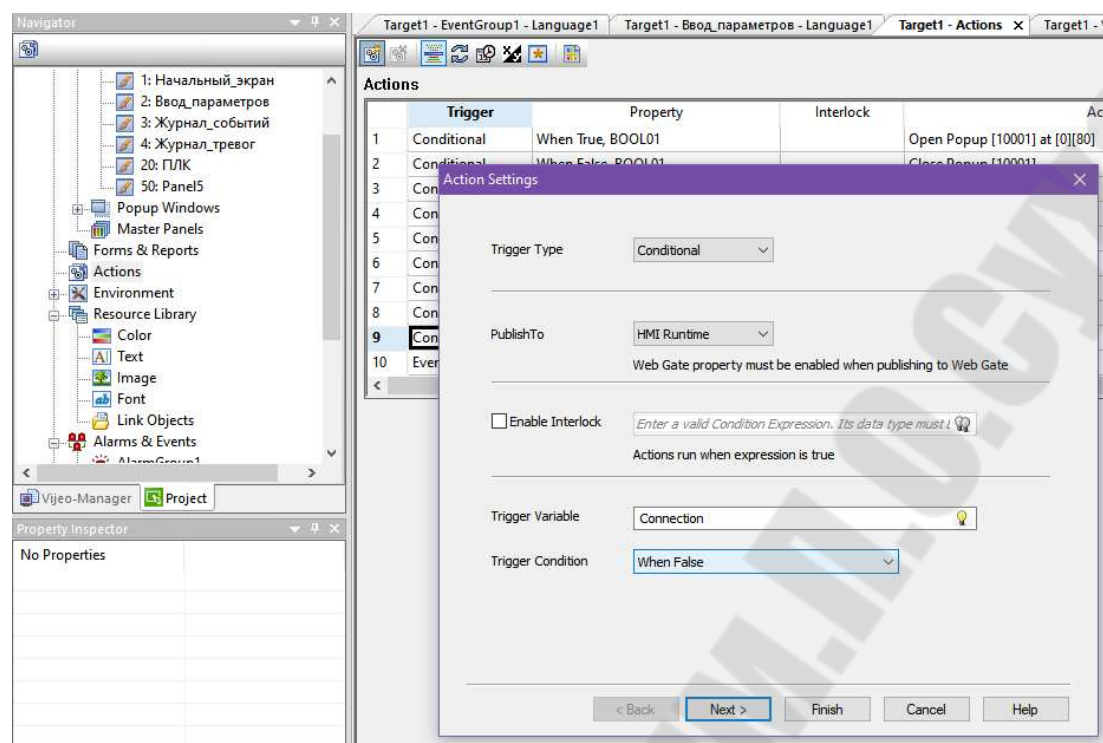


Рис. 13.11. Установка переменной Collection

В результате в симуляторе можно наблюдать следующую работу программы (рис. 13.12).

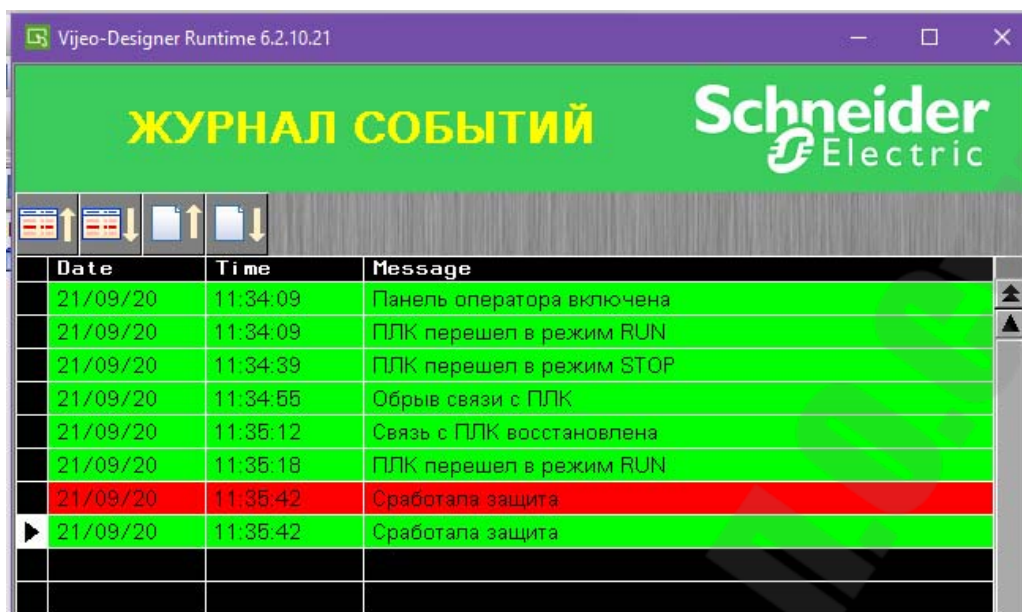


Рис. 13.12. Работа программы

14. Конфигурирование преобразователя частоты для работы по протоколу Modbus

Прежде чем приступить к конфигурированию преобразователя частоты (ПЧ), 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 = InI – заводская конфигурация;

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

COнF->FCS- ->GFS = YES (2 с) – возврат к заводским настройкам.

Настройку ПЧ можно выполнить как с помощью энкодера и семисегментного индикатора на передней панели ПЧ, так и с помощью специализированного программного обеспечения *SoMove*.

Для установки связи ПЧ с ПК необходимо использовать последовательный интерфейс RS-485. Для этого предварительно необходимо выполнить настройку связи на стороне ПЧ:

COнF->FULL->COП- ->Pd1- ->

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.

В данном примере ПЧ присвоен адрес 1, установлена наибольшая скорость передачи данных 38400 бод, параметры передаваемого пакета: 8 бит данных, 1 стоповый бит, проверка на четность.

Такие же параметры необходимо установить в программном обеспечении *SoMove* (рис. 14.1).

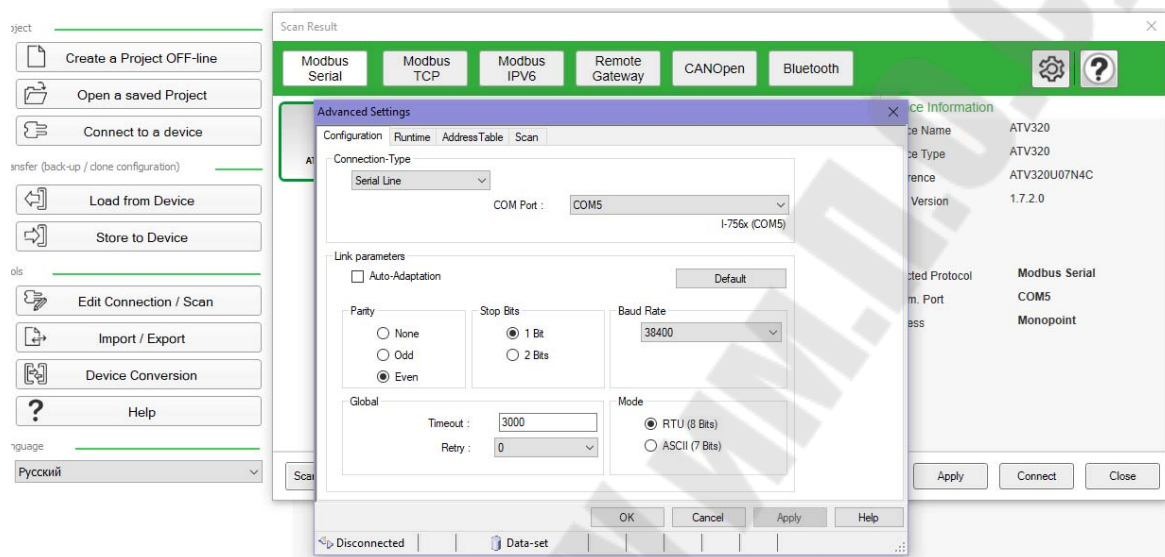


Рис. 14.1. Установка параметров *Modbus* в ПО *SoMove*

Для подключения ПК к линии RS-485 необходимо воспользоваться преобразователем интерфейса RS-485-> USB типа *i7561*. Для правильной работы преобразователя интерфейса необходимо на ПК установить драйвер *i-756x_1223_driverinstaller.exe*.

В диспетчере устройств ПК в разделе *com* и *lpt* появится виртуальный *com*-порт с именем *I-756x* (рис. 14.2). В настройках этого порта необходимо установить такие же параметры связи, как и на ПЧ.

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

- > Мониторы
- > Мыши и иные указывающие устройства
- > Очереди печати
- ✓ Порты (COM и LPT)
 - I-756x (COM5)
- > Программные устройства
- > Процессоры
- ✓ Сетевые адаптеры

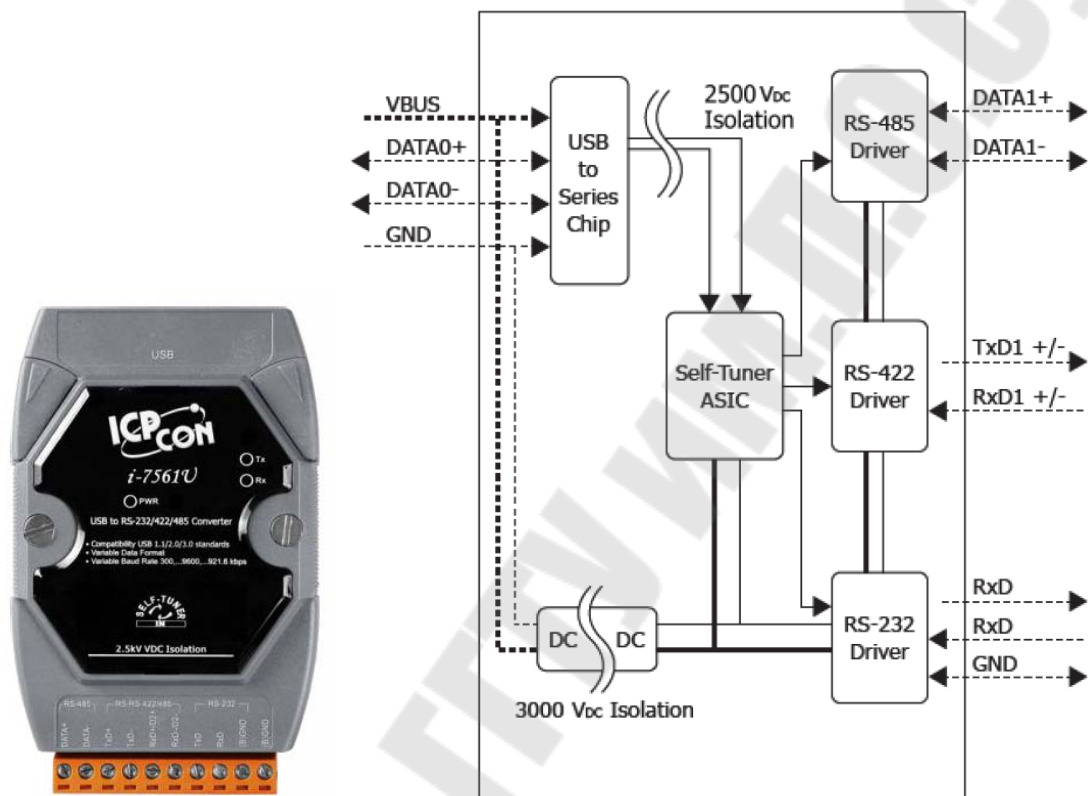


Рис. 14.2. Внешний вид и функциональная схема преобразователя интерфейса типа *i7561*

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

CO_NF->FULL->drC- ->

bFr = 50 (Гц) – частота стандартного двигателя;

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

Ctt = Std – скалярное управление;

CO_NF->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 = ПЕАС – используется результат автоподстройки;
tUnU = tП – оценка теплового состояния по потребляемому току;

AUt = nO – настройка деактивирована;
ПРС = nPr – мощность двигателя;
rSA = 10.00 мОм;
LFA = 36.31 мГн;
IdA = 1.4 А;
trA = 77 мс;
COnF->FULL->drC- ->
SFt = HF2 – оптимизация шума ЭД;
SFr = 16.0 – частота модуляции;
nrd = YES – уменьшение шума;
COnF->FULL->I_O->
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;
COnF->FULL->CtL- ->
Frl = AI1 – задание частоты через аналоговый вход 1 @AI1;
(Frl = Пdb – задание частоты через *Modbus*);
COnF->FULL->SIP- ->
tCC = 2C – двухпроводная схема;
tFr = 70.0 – максимальное допустимое значение частоты;
COnF->FULL->SEt- ->
LSP = 0.0 – минимальное значение частоты;
HSP = 70.0 – максимальное значение частоты (<=tFr).

Перечисленные выше параметры преобразователя частоты можно установить с помощью программного обеспечения *SoMove* (рис. 14.3).

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 μ s	10 μ s	
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
CLI	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 μ s	10 μ s	
VBR	Braking level	820 V	820 V	820 V

Рис. 14.3. Установка параметров преобразователя частоты

Настройка связи:

CO_nF->FULL->СОП- ->Pd1- ->

Add = 1 – Modbus Address @OFF;

tbr = 38 4 – Modbus baud rate @19 2;

tFO = 8E1 – Modbus format @8E1;

ttO = 10.0 – Modbus Timeout @10;

СОП1 = rOtO – состояние связи.

На рис. 14.4 показано, как перечисленные параметры изменить в программе *SoMove*.

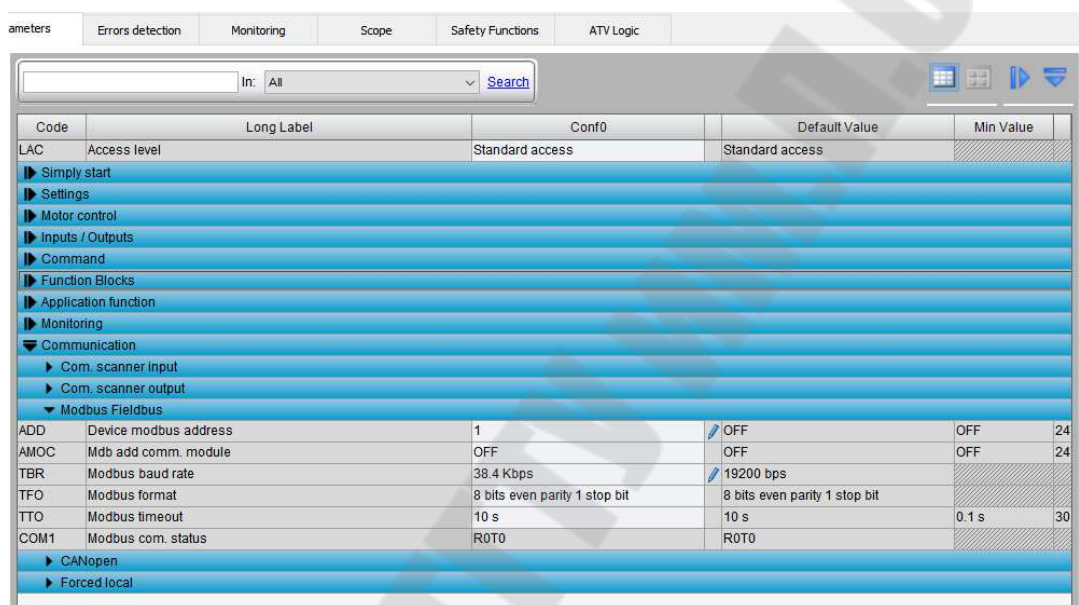


Рис. 14.4. Изменение параметров в программе *SoMove*

(!) 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:

1. В программном обеспечении *Machine Expert – Basic* перейти во вкладку «*Configura-tion*», затем выбрать подраздел «*SL1 (Serial line)*». Необходимо выбрать протокол *Modbus Serial IOScanner* (рис. 14.5).

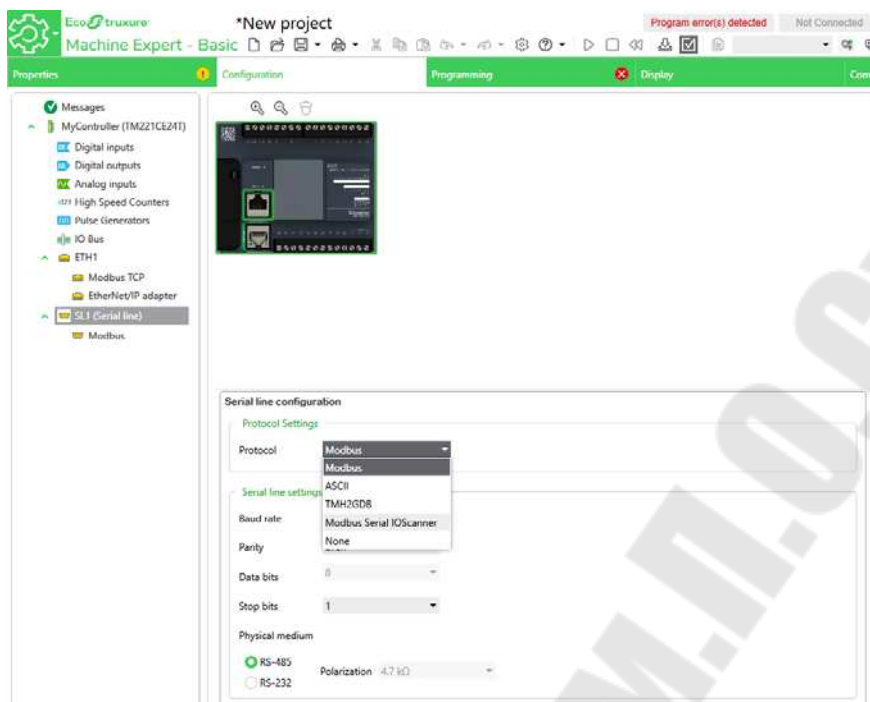


Рис. 14.5. Выбор протокола

2. В подразделе *SL1 Serial line* необходимо указать параметры соединения (рис. 14.6), которые должны совпадать с параметрами ПЧ (см. рис. 14.3).

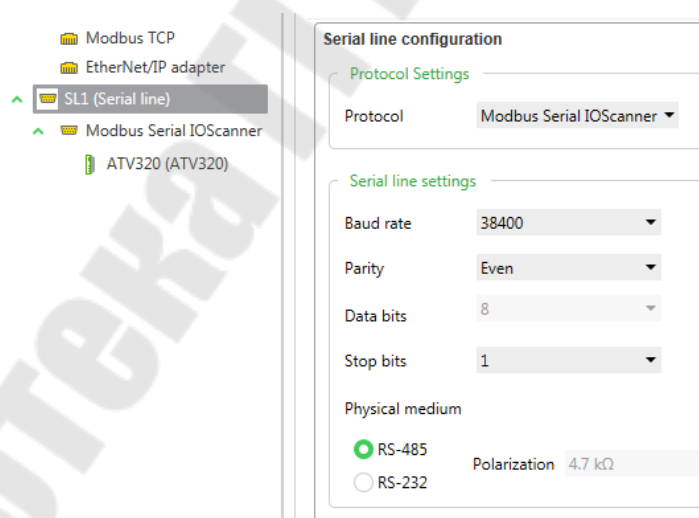


Рис. 14.6. Указание параметров соединения

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. 14.7).

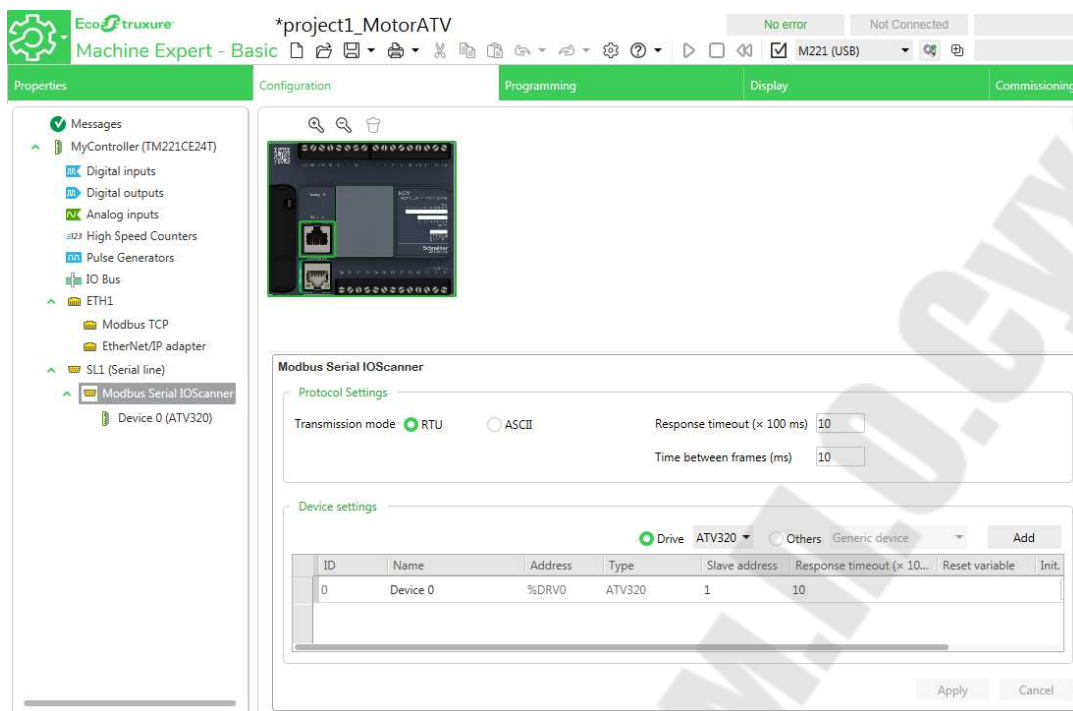


Рис. 14.7. Добавление параметров

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

CO_NF->FULL->FU_n- ->PSS- ->

PS2 = LI3 – 2 скорости;

PS4 = LI4 – 4 скорости;

PS8 = LI5 – 8 скоростей;

(PS16 = LI6 – 16 скоростей;)

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

JPF = 25.0 (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. 14.8).

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. 14.8. Выполнение настроек

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

CONF->FULL->CtL- ->

Frl = Pl – дискретные входы;

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

Также можно воспользоваться программой *SoMove* (рис. 14.9).

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	AI1 Analog input	
PST	Stop key enable	AI2 Analog input	AI2 Analog input	
CHCF	Control mode configuration	AI3 Analog input	AI3 Analog input	
CCS	Command switching	Reference frequency via remote terminal	Reference frequency via remote terminal	
CD1	Command channel 1 assign	Reference frequency via Modbus	Reference frequency via Modbus	
CD2	Command channel 2 assign	Reference frequency via CANopen	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. 14.9. Настройка параметров в программе *SoMove*

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

COnF->FULL->FbП- ->

FbrП = nO – запуск ФБ вручную.

Также можно воспользоваться возможностями программы *SoMove* (рис. 14.10).

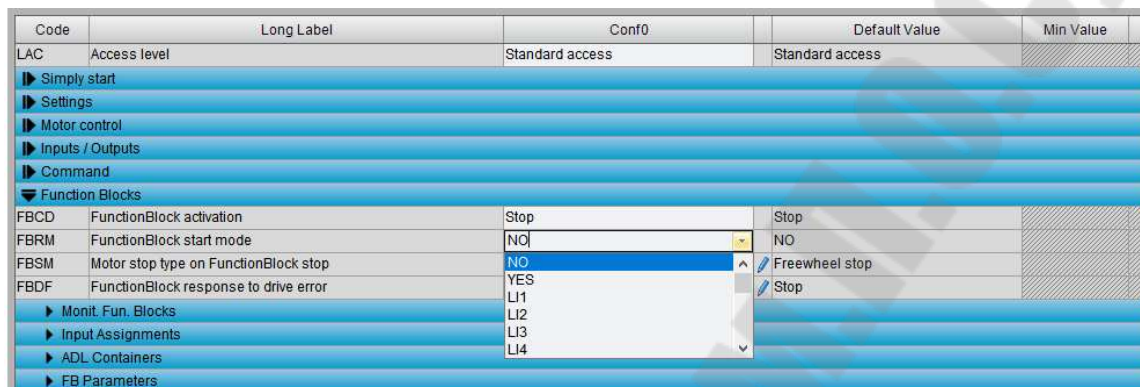


Рис. 14.10. Изменение параметров в программе *SoMove*

15. Управление преобразователем частоты по протоколу Modbus RTU

Создадим проект в среде *Machine Expert – Basic* с настройками последовательного интерфейса *SL1* в режим *Modbus Serial IOScanner*, как показано на рис. 14.3–14.7.

Для управления ПЧ по протоколу *Modbus* используем встроенные функциональные блоки программной среды *Machine Expert – Basic*.

Добавим программную ступень, в которую поместим блок *MC_Power_ATV*: включить/отключить силовую часть (рис. 15.1).

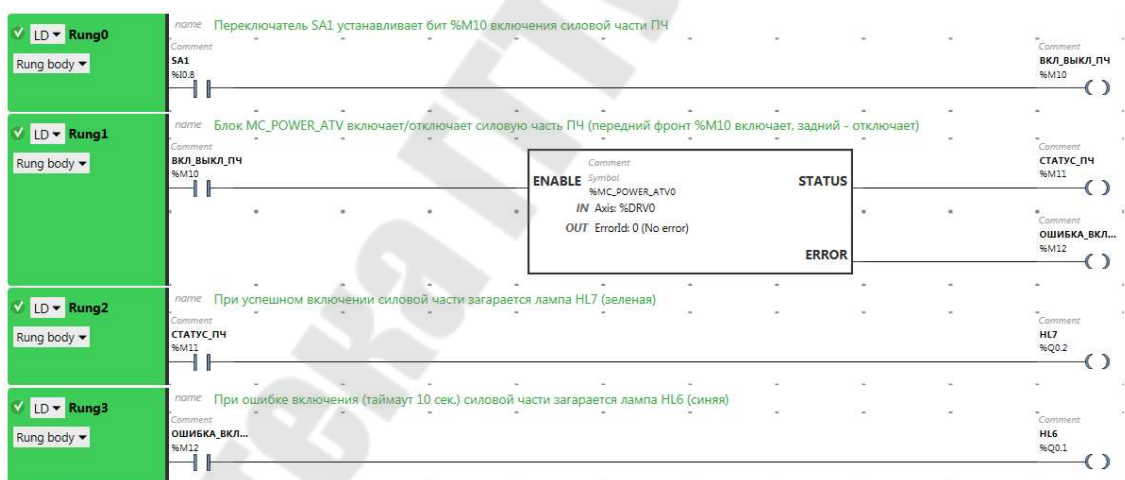
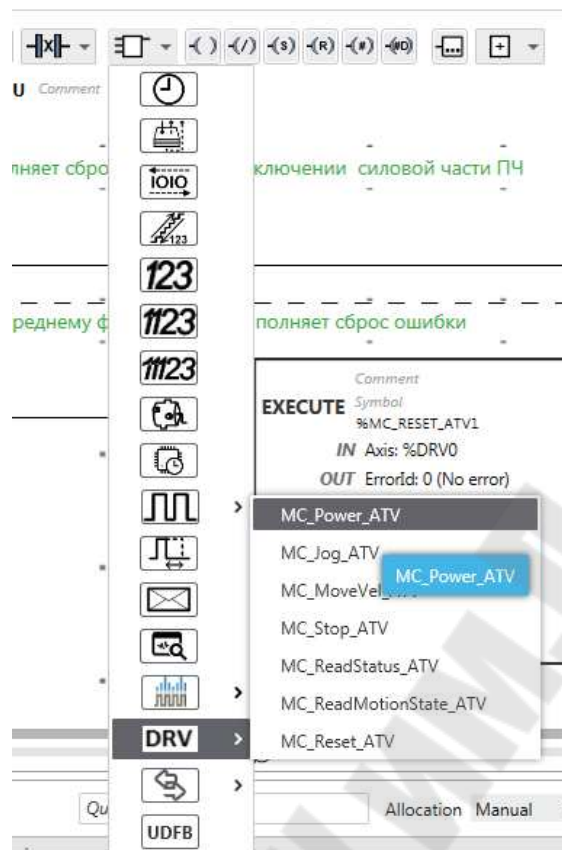


Рис. 15.1. Добавление в проект блока *MC_Power_ATV*

Этот функциональный блок включает или отключает силовую часть привода. **Передний (нарастающий) фронт** на входе *Enable* включает силовую часть. Когда силовая часть включена, сигнал состояния *Status* установлен в «1». **Задний фронт** на входе *Enable* отключает силовую часть. Когда силовая ступень отключена, сигнал состояния *Status* сбрасывается в «0».

Если внутренний регистр состояния привода ПЧ не достиг рабочего состояния до истечения значения таймута, то генерируется ошибка таймута. Если во время выполнения функционального блока обнаружены ошибки, выход *Error* устанавливается в «1». В случае возникновения ошибки только выполнение функционального блока *MC_Reset_ATV* может восстановить состояние силовой части (рис. 15.2).

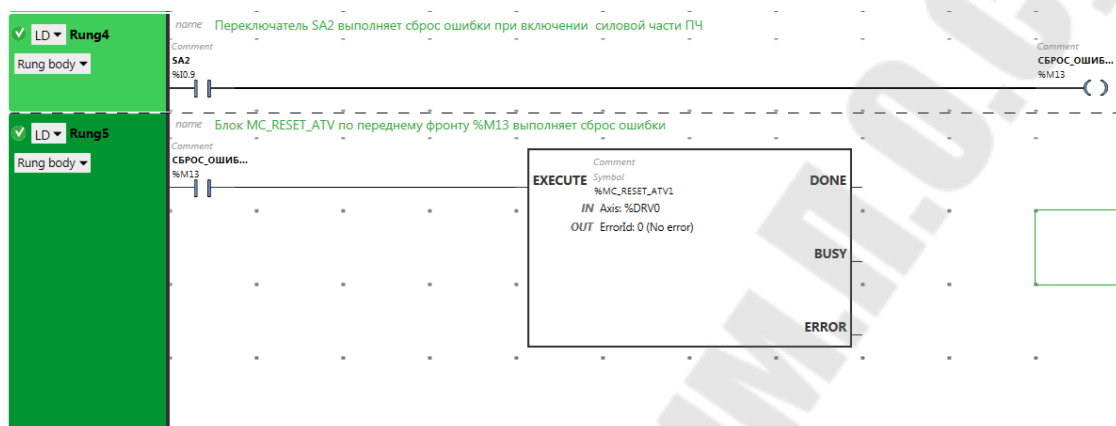


Рис. 15.2. Добавление в проект блока *MC_Reset_ATV*

Для задания направления и частоты вращения вала двигателя используем блок *MC_JOG_ATV* (рис. 15.3). Когда выполняется данная операция, изменение значения скорости (*Vel*) применяется только при обнаружении спадающего/нарастающего фронта входов прямого или обратного направления.

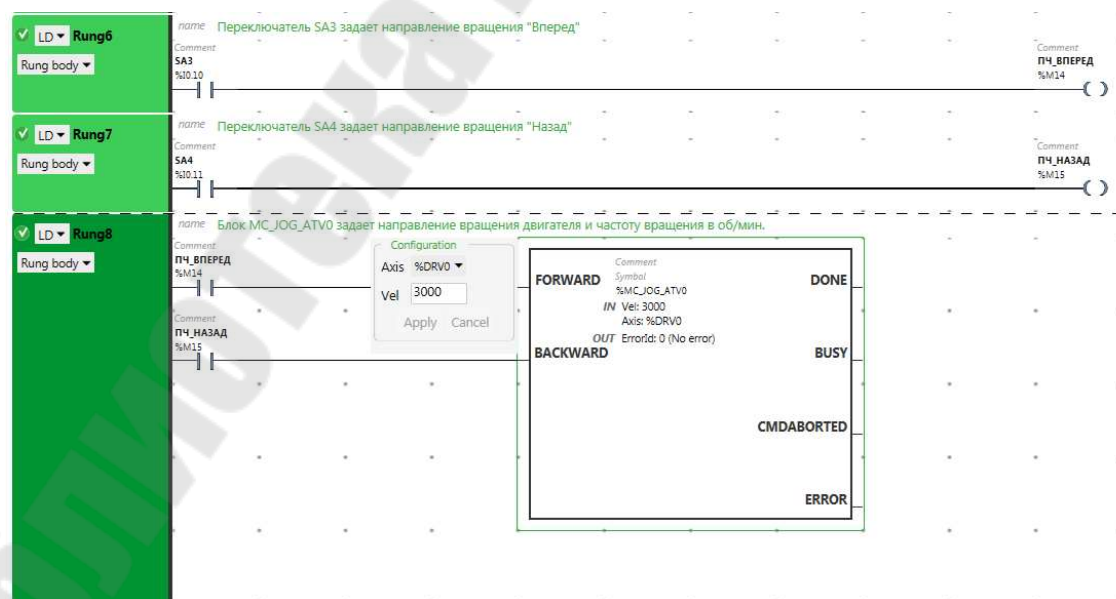


Рис. 15.3. Добавление в проект блока *MC_JOG_ATV*

Если любой из выходов *Error* или *CmdAborted* установлен в «1», входы *Forward* и *Backward* должны сначала быть сброшены в «0», а затем новый нарастающий фронт на входах прямого и/или обратного направления возобновит движение.

Переменная *Vel* задает значение частоты вращения поля статора в [об/мин] таким образом, что, например, 3000, означает 50 Гц при одной паре полюсов. Во время работы блока изменение значения скорости *Vel* применяется только при обнаружении спадающего/нарастающего фронта на входах «Вперед» или «Назад». Переменная *Vel* может принимать значение в диапазоне: –32768–32767.

Для остановки ПЧ применим блок *MC_Stop_ATV*: *остановить движение* (рис. 15.4).

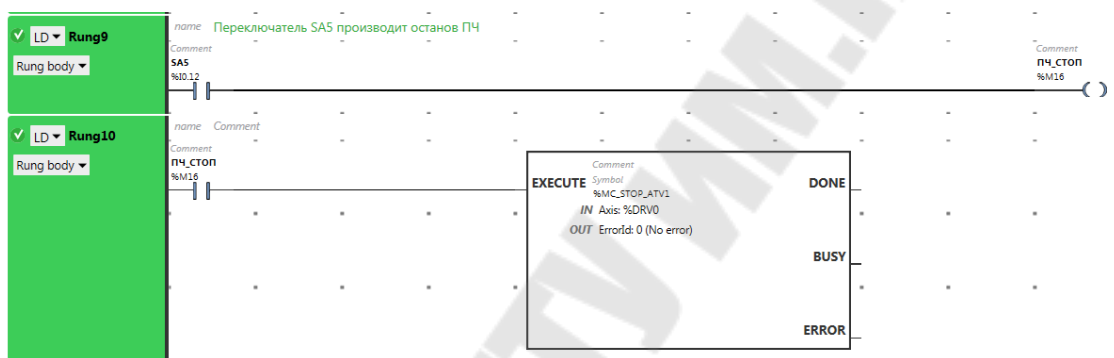


Рис. 15.4. Добавление в проект блока *MC_Stop_ATV*

Этот функциональный блок останавливает текущее движение привода. Параметры остановки, например, замедление, определяются конфигурацией ПЧ.

После прихода переднего фронта на вход *Execute* любое дальнейшее действие на входе *Execute* прекращается до тех пор, пока выход *Done* не будет установлен в значение «1».

Выполнение другого функционального блока группы *DRV* не возможно, если для выхода *Busy* установлено значение «1». В этом случае другой функциональный блок возвращает ошибку. Процедура остановки может быть прервана только при отключении силовой части или при возникновении ошибки.

Для задания скорости ПЧ используем блок *MC_MoveVel_ATV*: *вращение с указанной скоростью* (рис. 15.5).

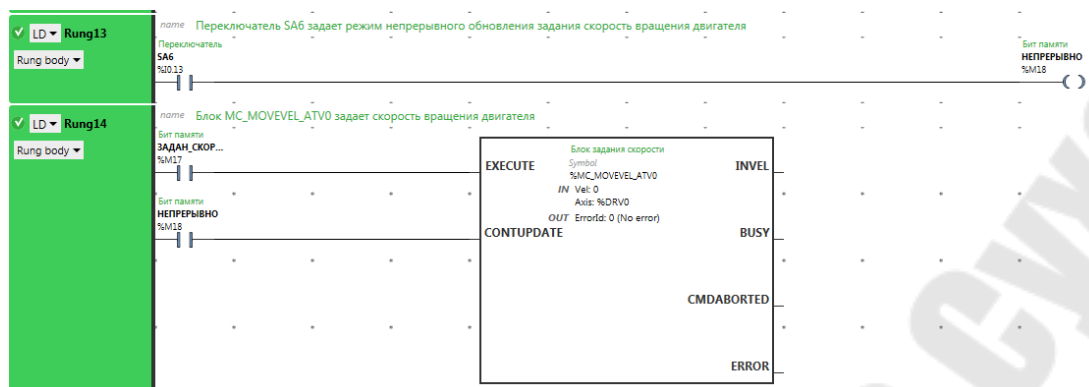


Рис. 15.5. Добавление в проект блока *MC_MoveVel_ATV*

Этот функциональный блок запускает рабочий режим *Profile Velocity* с заданной скоростью. Когда достигнута заданная скорость, выход *InVel* устанавливается в «1».

Если функциональные блоки *MC_Jog_ATV* или *MC_Stop_ATV* включены пока выполняется функциональный блок *MC_MoveVel_ATV* (выход *Busy* установлен в «1»), *MC_MoveVel_ATV* подает команду движение.

Входные значения *ContUpdate* и *Vel* применяются по нарастающему фронту входного сигнала *Execute*.

Если любой из выходов *Error* или *CmdAborted* *MC_MoveVel_ATV* установлен в «1», для возобновления движения необходим новый нарастающий фронт *Execute*.

Для непрерывного задания частоты ПЧ от потенциометра *RP2* можно выполнить, например, последовательность действий, показанную на рис. 15.6.

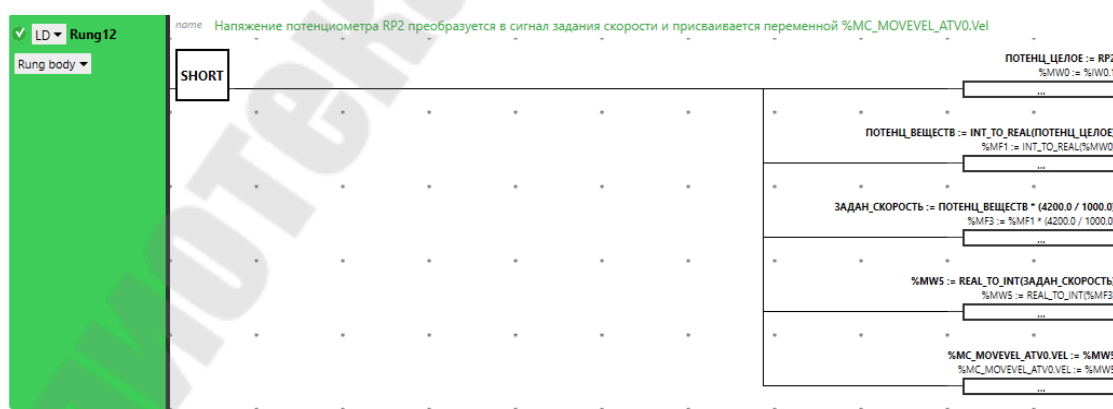


Рис. 15.6. Последовательность действий для задания частоты ПЧ

Здесь значение регистра АЦП (аналогового входа) сохраняется в целочисленном виде в регистре $\%MW0$. Затем это значение преобразуется к виду с плавающей точкой в регистре $\%MF1$, после чего оно домножается на коэффициент пропорциональности, устанавливающий связь между максимальным значением АЦП и максимальным заданием частоты ПЧ. В контроллере M221 АЦП имеют разрядность $n = 10$ бит, а максимальное значение в регистре $\%IW$ составит 1000. Частоте ПЧ 50 Гц соответствует значение задания 3000 (частота вращения поля двигателя). Далее полученный результат преобразуется назад к целочисленному виду и сохраняется в переменной Vel функционального блока $MC_MoveVel_ATV$. Данные вычисления выполняются непрерывно в каждом цикле программы.

Для активации режима непрерывного обновления значения задания частоты ПЧ на вход $ContUpdate$ блока $MC_MoveVel_ATV$ должна быть подана «1» до подачи переднего фронта на вход $Execute$.

Обратите внимание (!) на размещение переменных в памяти (рис. 15.7). Различные типы данных, такие как $\%MW$ (целочисленное слово), $\%MD$ (целочисленное двойное слово) и $\%MF$ (вещественное число с плавающей точкой), занимают одно и то же пространство адресов в памяти. При этом занятые ячейки памяти отмечены в чекбоксах граф «Used» (для выбранного типа переменных) и «Equ Used» (для остальных типов). Поэтому нельзя создавать новую переменную там, где уже стоит хотя бы одна отметка. При этом нужно помнить, что одна переменная типа $\%MW$ занимает 16 бит, а переменная типа $\%MF$ или $\%MD$ занимает 32 бита, т. е. равна по объему занимаемой памяти двум переменным типа $\%MW$.

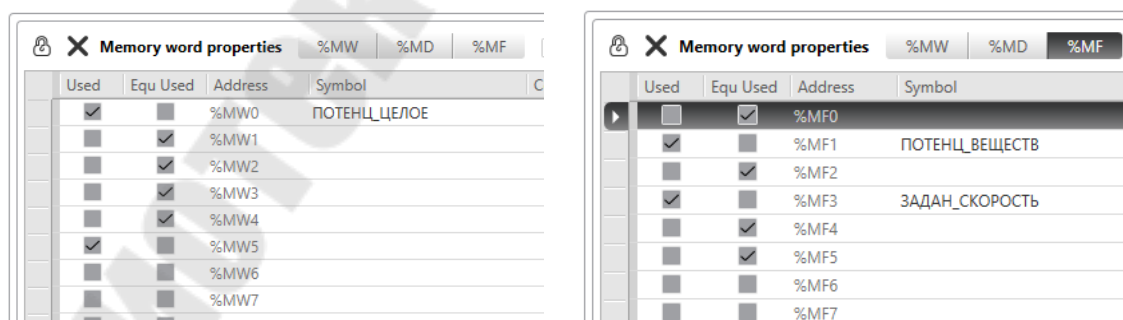


Рис. 15.7. Размещение переменных в памяти

Параметры АЦП ПЛК М221 приведены в табл. 15.1.

Таблица 15.1

Параметры АЦП ПЛК М221

Characteristic		Voltage Input
Number of maximum inputs		2 inputs
Input type		Single-ended
Rated input range		0...+10 Vdc
Digital resolution		10 bits
Input value of LSB		10 mV
Input impedance		100 kΩ
Input delay time		12 ms
Sample duration time		1 ms per channel + 1 scan time
Accuracy		± 1 % of the full scale
Noise resistance - maximum temporary deviation during perturbations		± 5 % maximum of the full scale when EMC perturbation is applied to the power and I/O wiring
Isolation	Between input and internal logic	Not isolated
Connection type		Specific connector and cable (supplied)
Connector insertion/removal durability		Over 100 times
Cable	Type	Proprietary (supplied)
	Length	1 m (3.3 ft)

16. Настройка функции ATV Logic

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

Для этого создадим входные и выходные переменные (рис. 16.1 и 16.2), 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.

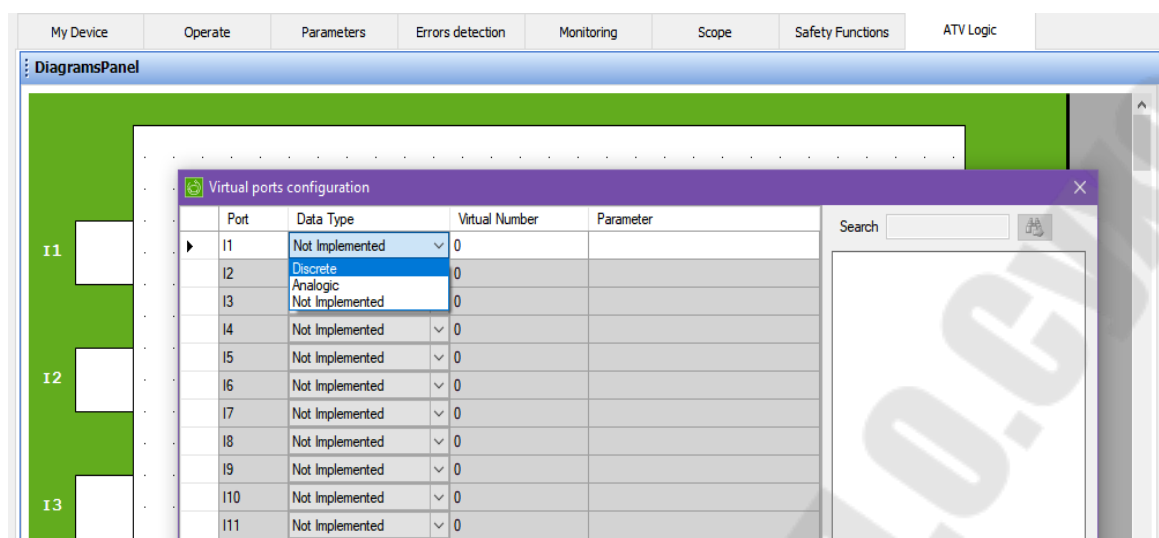


Рис. 16.1. Создание входных параметров

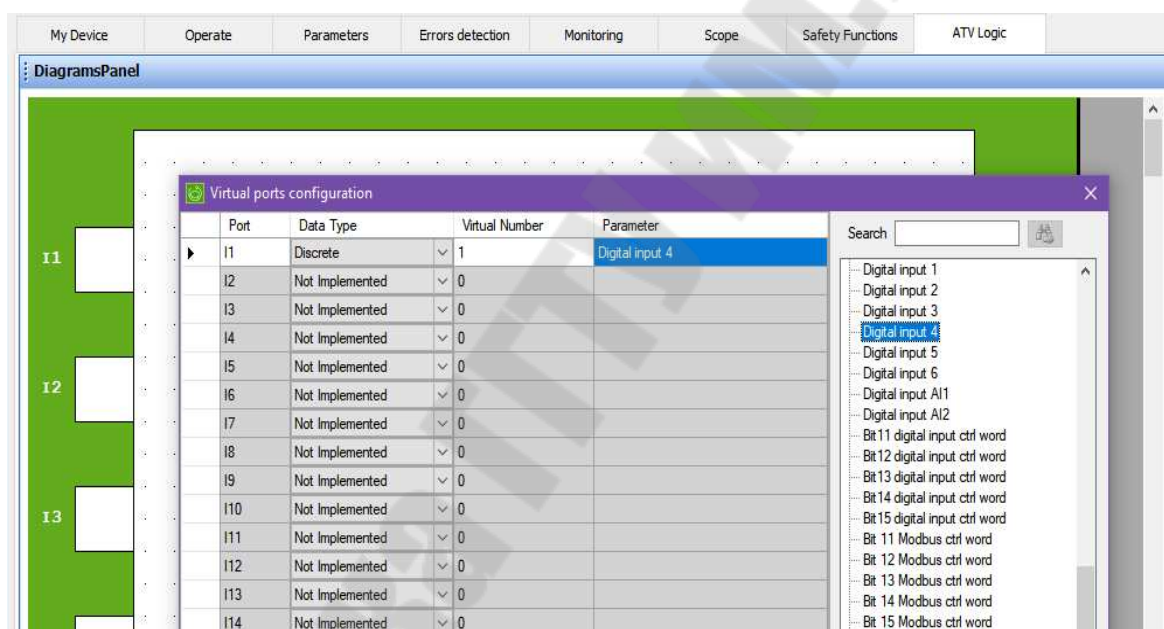


Рис. 16.2. Создание выходных переменных

Создадим логическую функцию «И» между входами и выходом и компилируем проект (рис. 16.3).

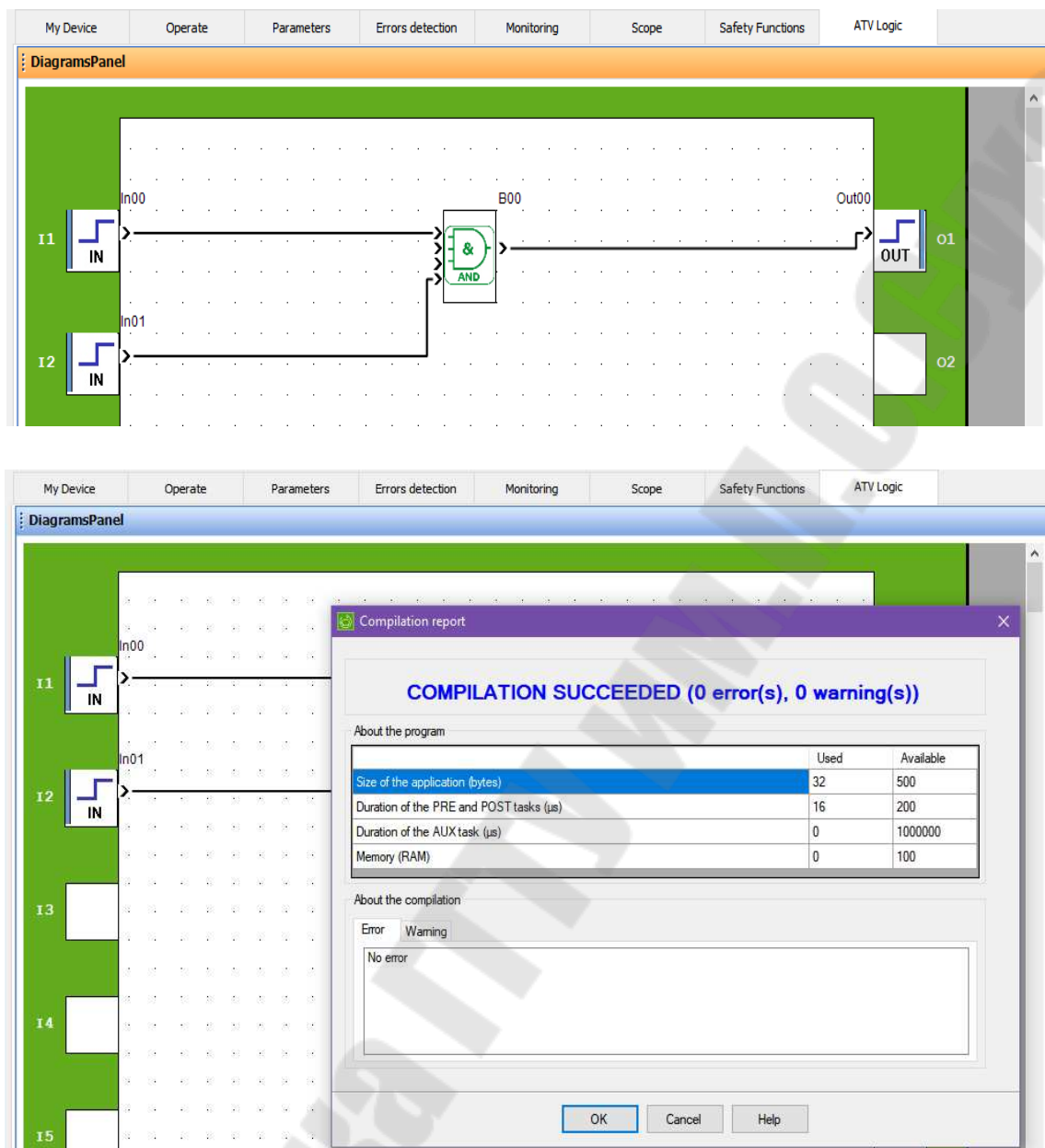


Рис. 16.3. Создание логической функции

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. 16.4).

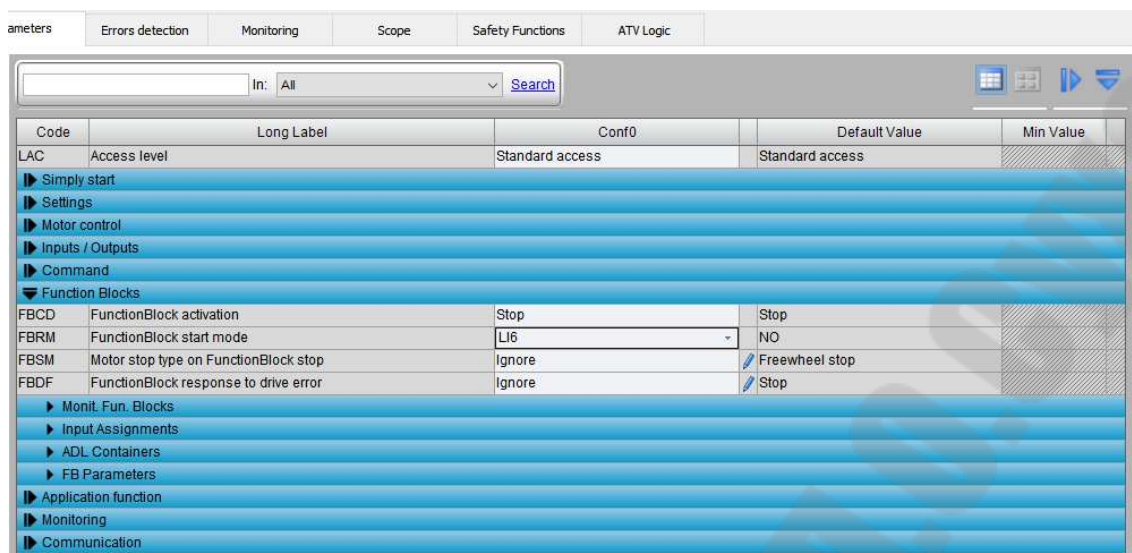


Рис. 16.4. Назначение события с указанием реакций

На рис.16.5 приведена программа *ATV Logic*, позволяющая ПЧ работать с тремя предустановленными частотами.

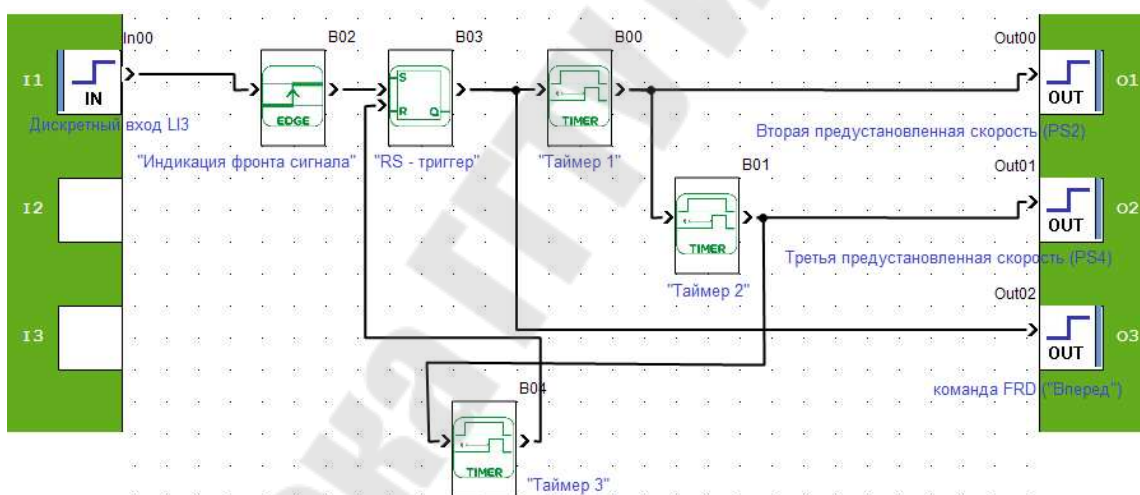


Рис. 16.5. Программа *ATV Logic*, позволяющая ПЧ работать с тремя предустановленными частотами

На рис. 16.6 приведена программа, позволяющая осуществлять подсчет импульсов энкодера.

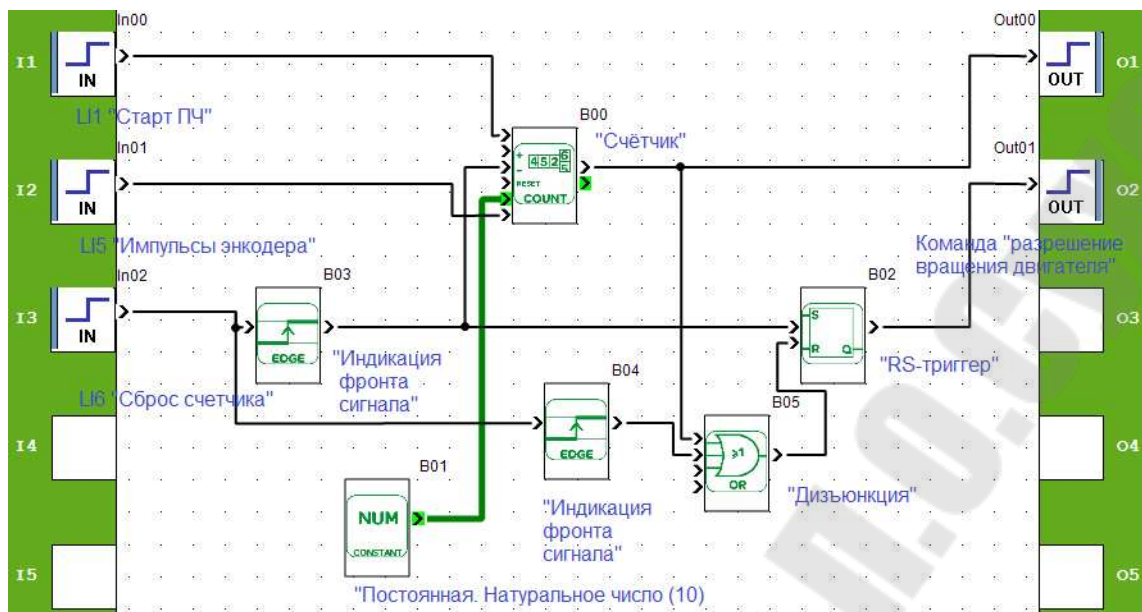


Рис. 16.6. Программа, позволяющая осуществлять подсчет импульсов энкодера

17. Пример создания приложения

Приложение должно удовлетворять следующим критериям:

- управлять заполнением резервуара в соответствии с уставкой заполнения и уровнем тревоги;
- позволять пользователю задавать уставку заполнения и уровень тревоги из ряда предустановок. Для выбора предустановок использовать функцию рецептов;
- выполнять опорожнение бака путем открытия/закрытия донного клапана при нажатии кнопки;
- давать представление об изменении уровня жидкости в резервуаре во времени путем просмотра график тренда;
- информировать пользователя о превышении порогового значения с помощью сигнальной лампы и страницы тревоги.

1. Создадим три идентичные панели (рис. 17.1). На каждой из панелей разместим по три элемента *Switch* с функцией *смена панели* и с надписью «Резервуар», «График», «Тревоги».

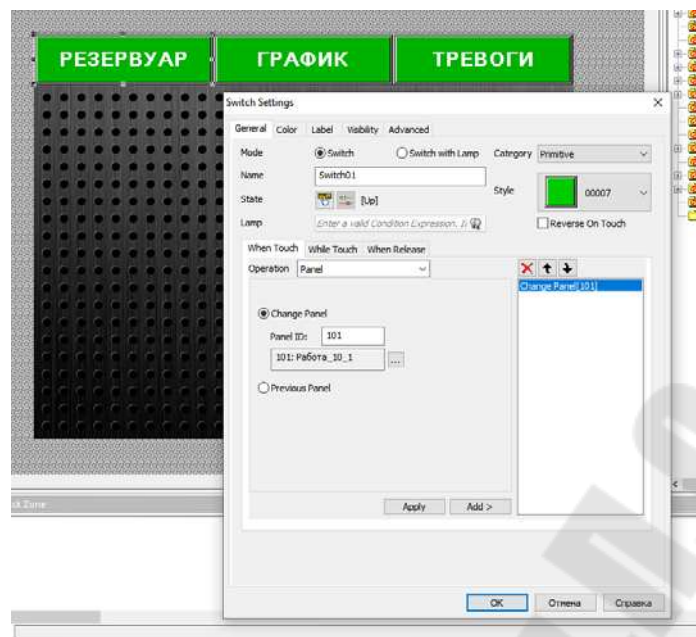


Рис. 17.1. Создание панелей

2. Создадим четыре переменные. Переменную типа *bool* с именем *HIGH_LEVEL* (рис. 17.2), расположенную в памяти ПЛК. Эту переменную на вкладке *Alarm* отнесем к группе *AlarmGroup 1*.

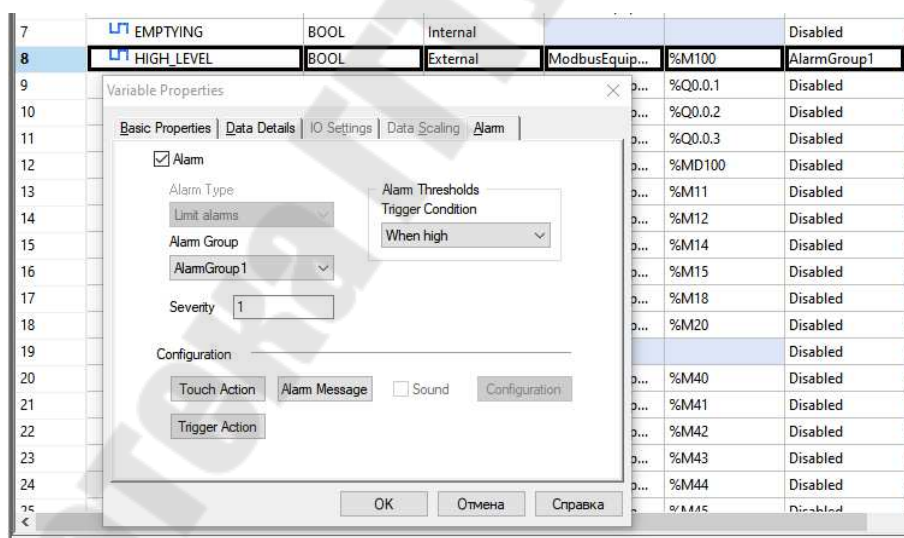


Рис. 17.2. Создание переменных

Также создадим вторую переменную типа *bool* с именем *EMPTYING*, расположенную во внутренней памяти терминала.

3. Еще создадим две переменные типа *UINT* с именами *LEVEL* и *SETUP*. Первая располагается в памяти ПЛК, вторая – в памяти

терминала. Обе переменные отнесем к группе регистрации *LoggingGroup01* на вкладке *Data Details* (рис. 17.3).

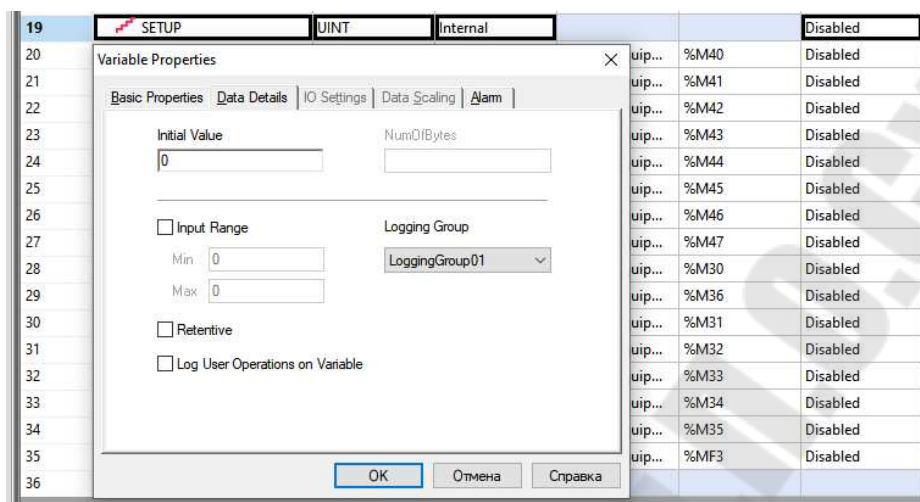


Рис. 17.3. Создание переменных *LEVEL* и *SETUP*

4. На экране «Резервуар» создадим две текстовые надписи «УСТАВКА ЗАПОЛНЕНИЯ:» и «УРОВЕНЬ ТРЕВОГИ:».

Рядом с надписями разместим два элемента типа *Numeric Display*. Для обоих элементов укажем тип отображаемого числа *Integer* (рис. 17.4). Первый элемент свяжем с ранее созданной переменной *LEVEL*, второй – с переменной *SETUP*. Зададим различный цвет фона для обоих элементов. Для переменной *SETUP* разрешим режим ввода значений *Enable Input Mode* на вкладке *Input Mode*.

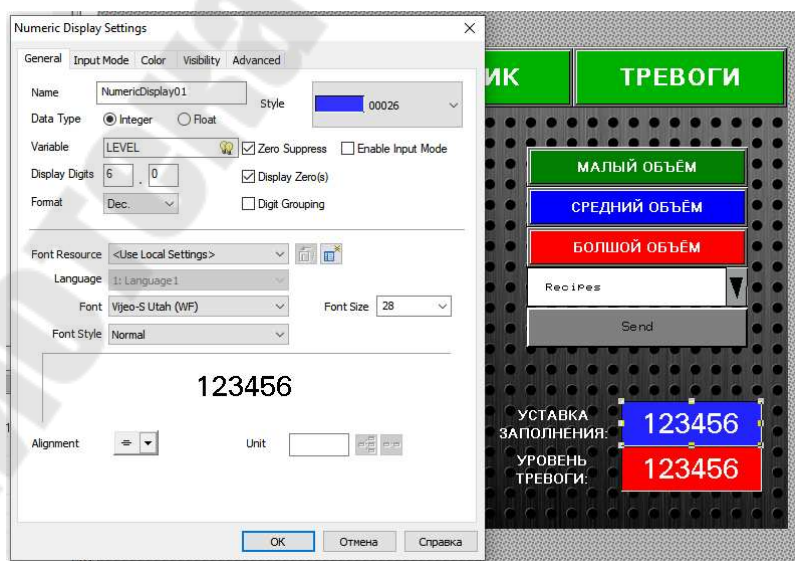


Рис. 17.4. Окно *Numeric Display*

5. Из панели инструментов *Toolchest Folders* перенесем на панель «Резервуар» элемент *TankGrahp_0001_01* (рис. 17.5). Свяжем изображение резервуара с переменной *LEVEL*.

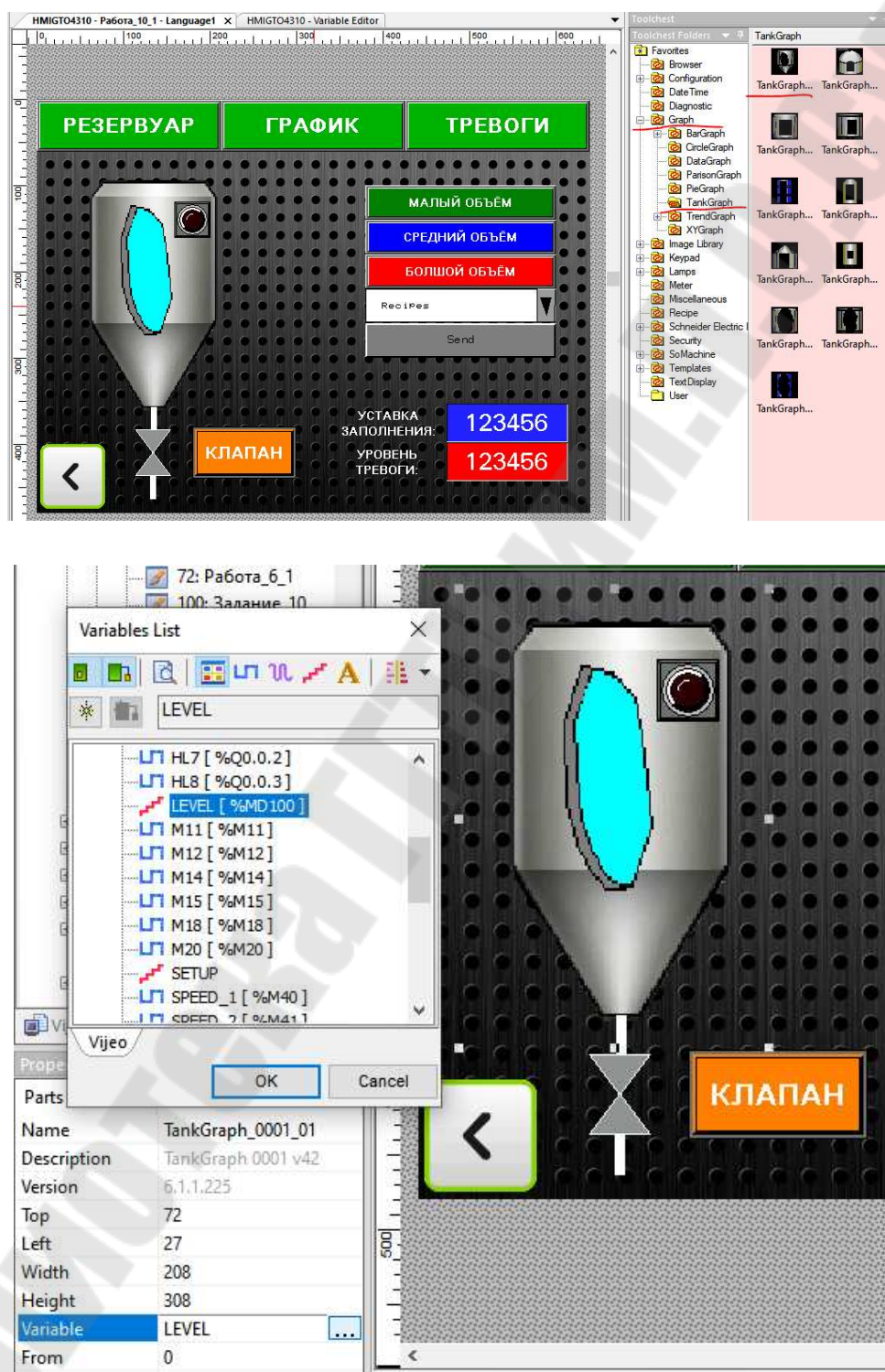


Рис. 17.5. Панель «Резервуар»

6. Для изображения донного клапана воспользуемся инструментами *Normal Line* и *Polygon* панели инструментов *Graphical Tools*. Создадим анимацию клапана, связав его с переменной *EMPTYING*, и выбрав различные цвета для открытого и закрытого состояния (рис. 17.6).

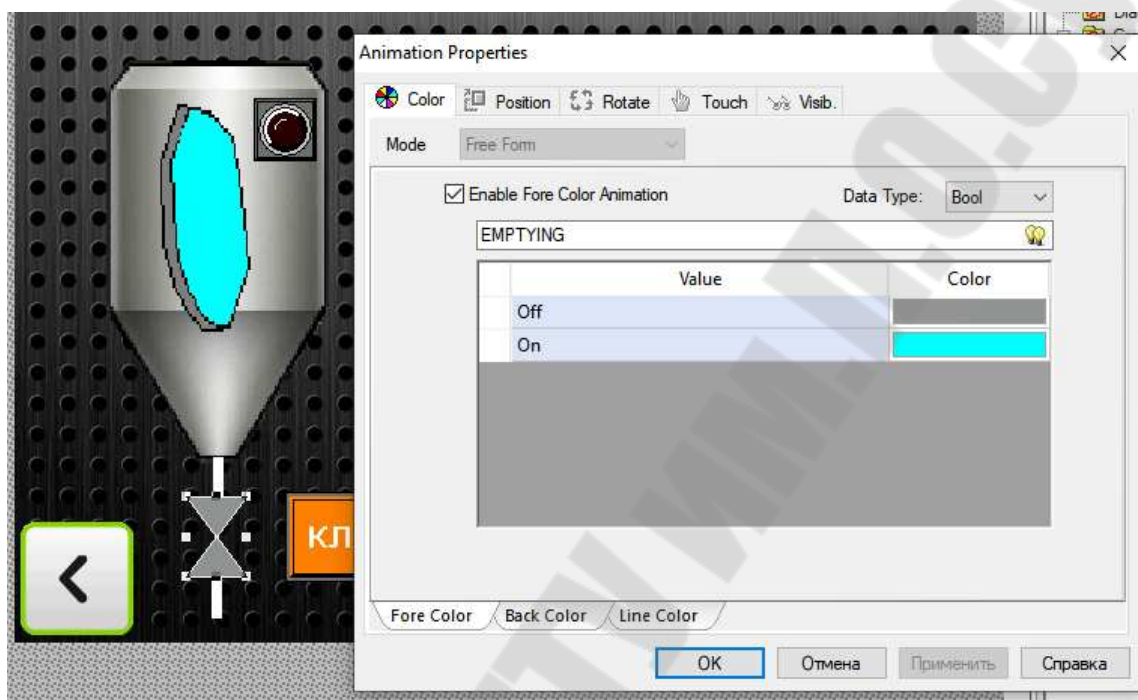


Рис. 17.6. Создание анимации клапана

7. На корпусе резервуара разместим лампу, связав ее с переменной *HIGH_LEVEL* и указав значение *Fast* (частое мигание) для параметра *Speed* на вкладке *Visibility*.

8. Создадим **рецепт** (структуру данных). Для этого в дереве проекта правой кнопкой мыши выберем раздел *Recipes* и в меню нажмем *New Recipe Groupe*. В проект будет добавлена новая структура данных – рецепт (рис. 17.7). На основе данного рецепта создадим три структуры параметров «*Small*», «*Medium*» и «*Large*». В графе «*Language*» присвоим структурам символьные имена «МАЛЫЙ ОБЪЕМ», «СРЕДНИЙ ОБЪЕМ», «БОЛЬШОЙ ОБЪЕМ» (рис. 17.8).

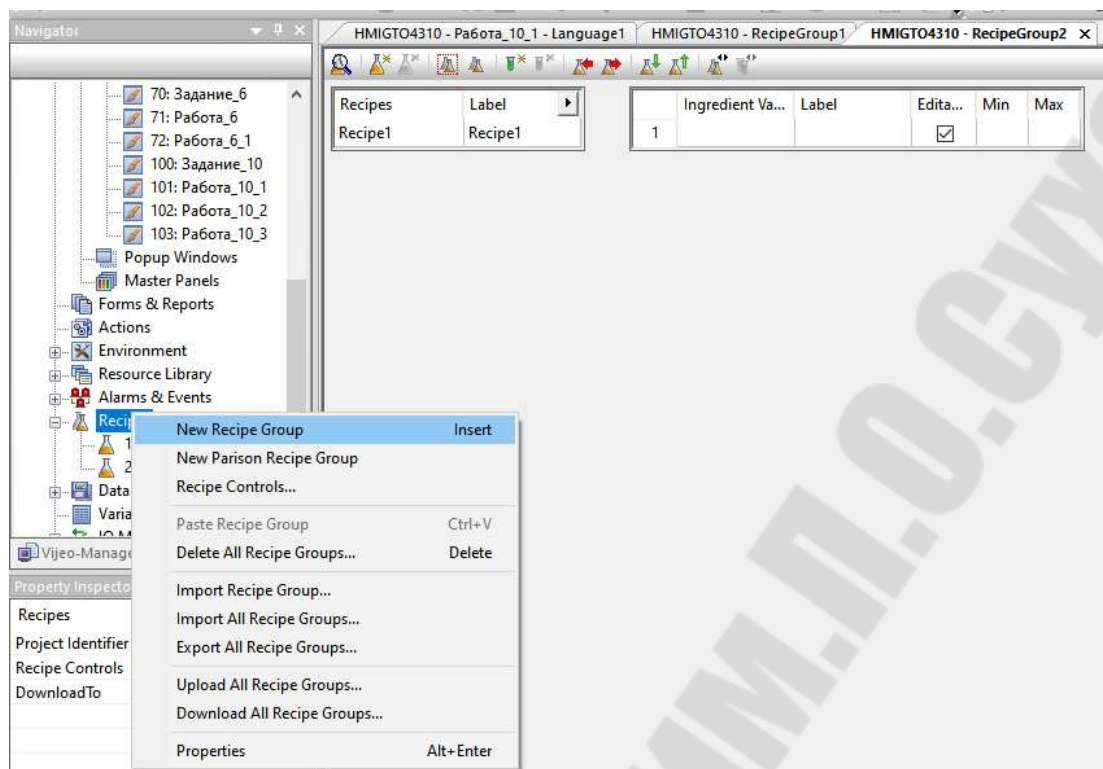


Рис. 17.7. Добавление в проект рецепта

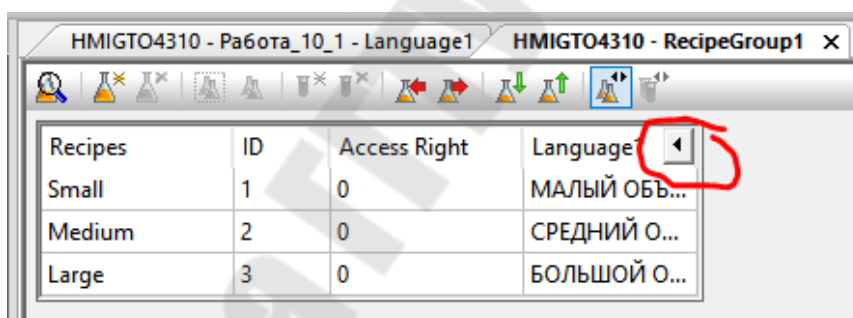


Рис. 17.8. Присвоение имен структурам

Добавим в рецепт две переменные (поля структуры) – *LEVEL* и *SETUP*. Зададим минимальное (0) и максимальное (100) значения переменных. Для каждой структуры зададим уникальные значения переменных (рис. 17.9).

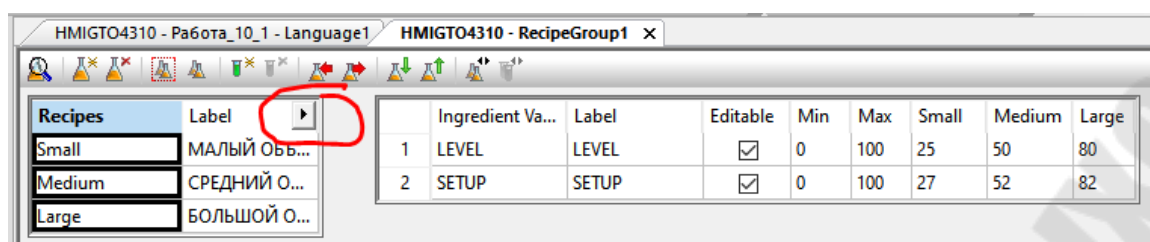


Рис. 17.9. Задание значений переменных структур

9. На панели «РЕЗЕРВУАР» разместим инструменты для выбора рецептов. На панели *Toolchest Folders* выберем инструменты *RecipeGroupeSelect*, *RecipeGroupeSelect* и *Send* (рис. 17.10).

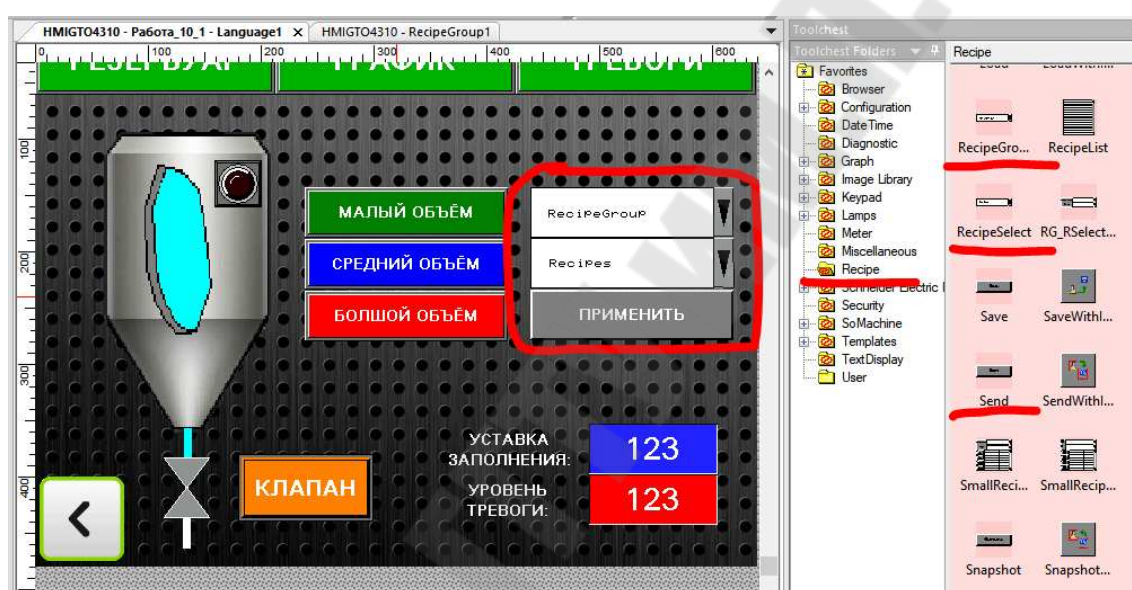


Рис. 17.10. Выбор инструментов

Изменим шрифт инструментов выбора группы рецептов и рецепта, как показано на рис. 17.11.

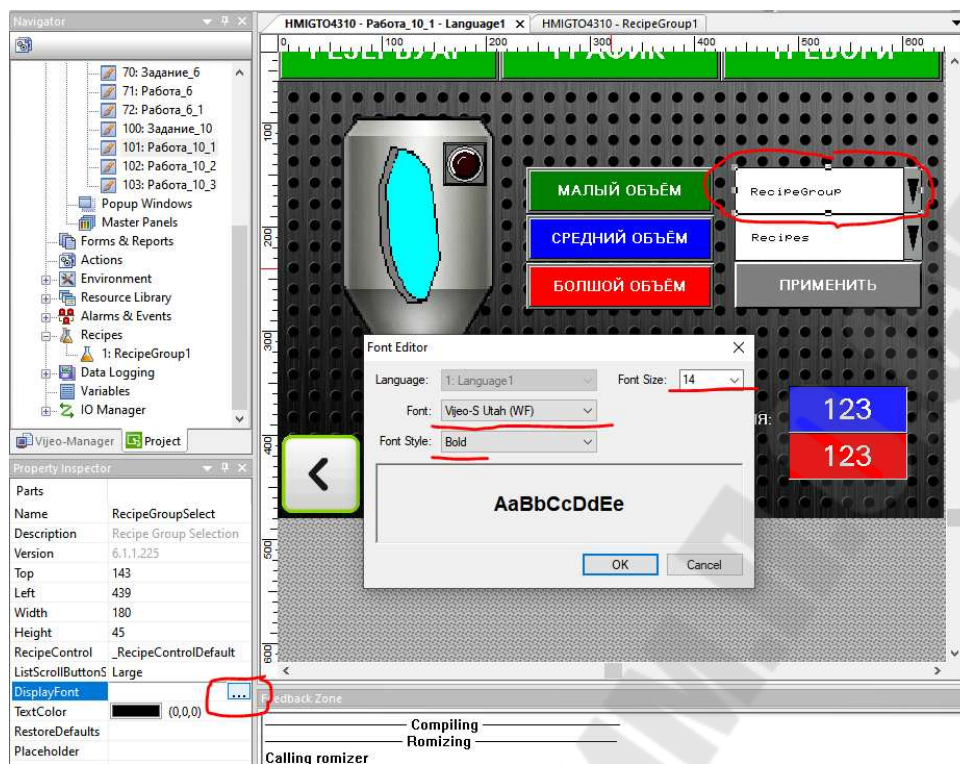


Рис. 17.11. Настройка отображения шрифта рецепта

Также изменим надпись и тип шрифта инструмента SEND (рис. 17.12)

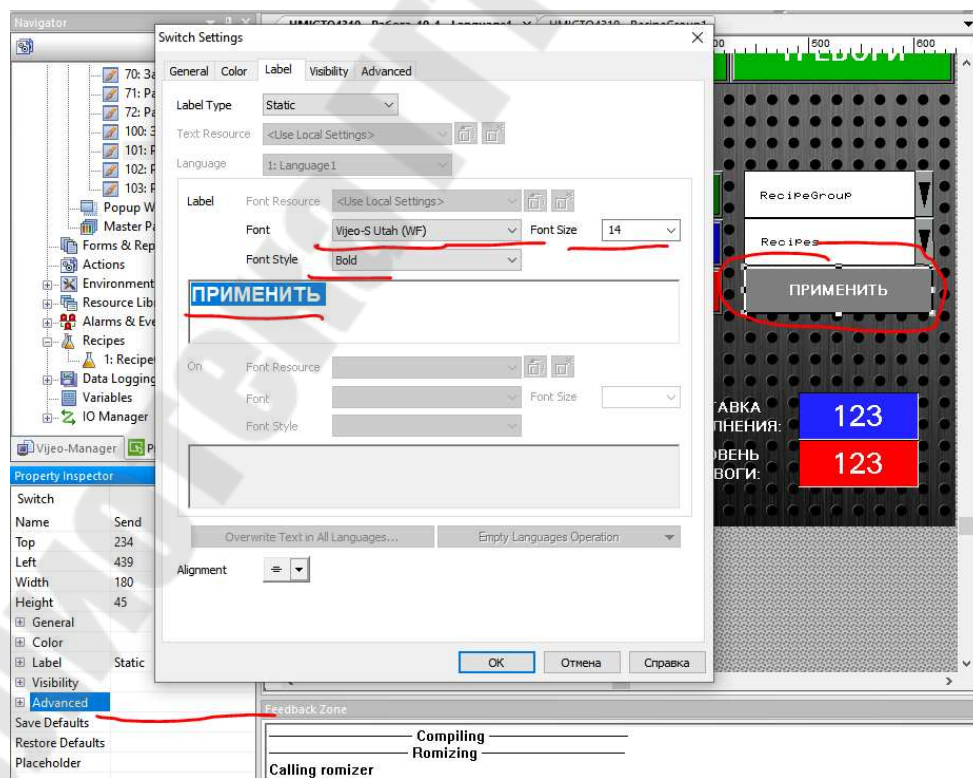


Рис. 17.12. Изменение настроек переключателя

18. Методика проведения лабораторных работ по модулю 2

18.1. Описание лабораторного стенда

Внешний вид лабораторного стенда приведен на рис. 18.1.

В состав стенда входят: 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.

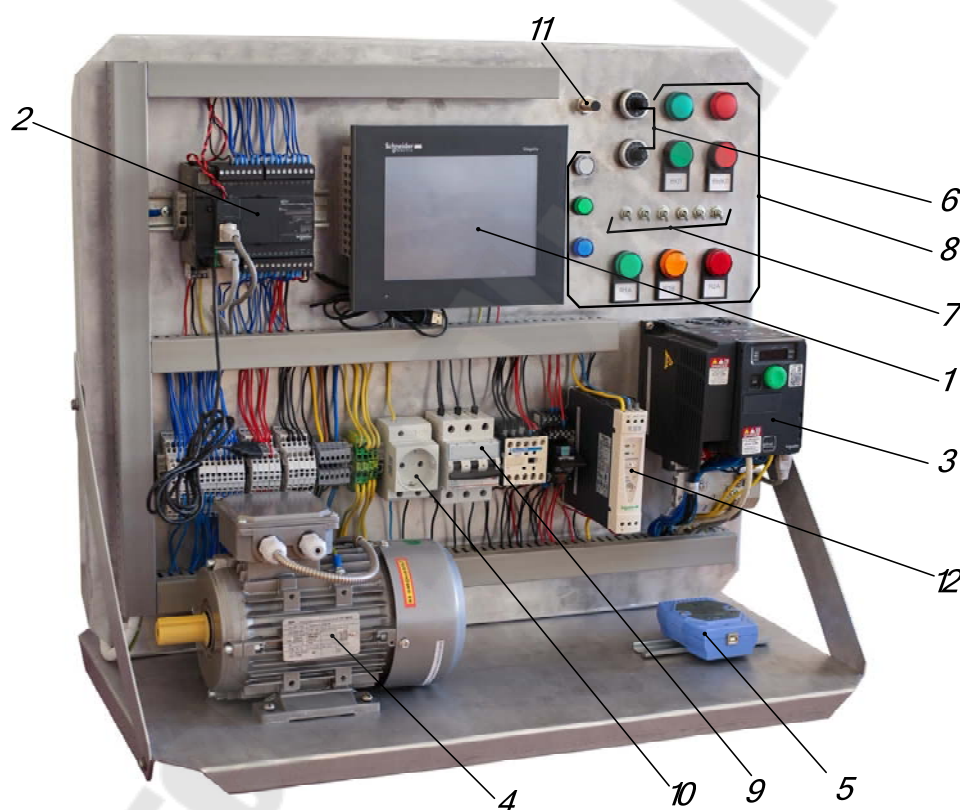


Рис. 18.1. Внешний вид лабораторного стенда

Принципиальная электрическая схема стенда приведена на рис. 18.2.

Чтобы обеспечить работоспособность стенда, необходимо подключить его к сети переменного напряжения 380 В с помощью вилки ХР1. Питание компонентов сети постоянного тока напряжением 24В осуществляется с помощью блока питания А1, подключенного к сети 220 В через автоматический выключатель QF1.

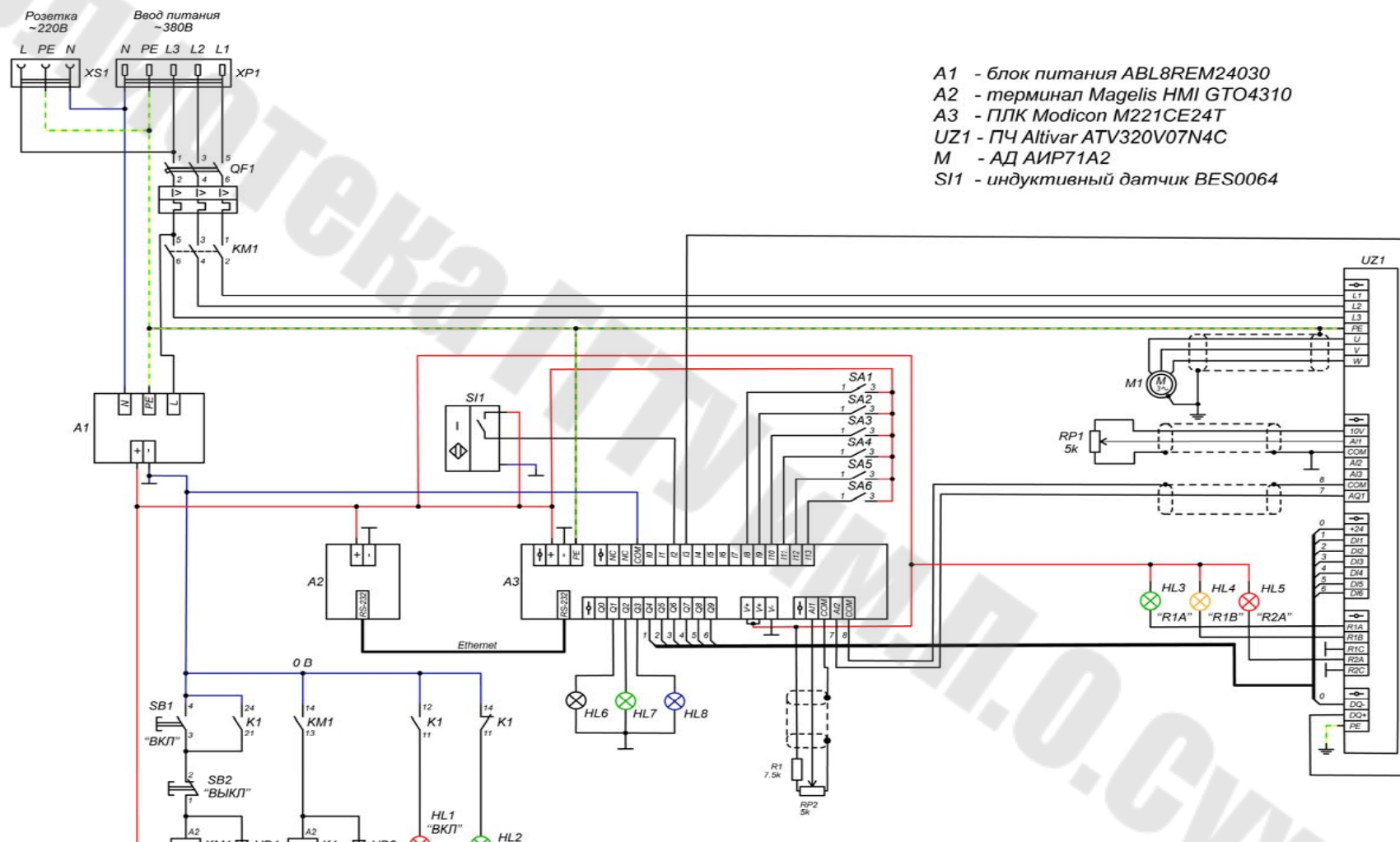


Рис. 18.2. Принципиальная электрическая схема стенда

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.

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. Также размыкается 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 катушкам подключены диоды VD2 и VD3 соответственно.

К входам I8–I13 ПЛК подключены тублеры SA1–SA6 соответственно. На вход I2 ПЛК подключен индуктивный датчик SI1. 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 DI1–DI6 преобразователя частоты UZ1. К аналоговым входам AI1 и AI2 ПЛК подключены соответственно потенциометр RP2 и аналоговый выход AQ1 преобразователя UZ1.

К релейным выходам R1A, 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.

18.2. Рекомендуемая последовательность выполнения работ

1. 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.

18.3. Примерный перечень заданий для лабораторных работ

Лабораторная работа № 1

Исследование работы дискретных входов и выходов ПЛК

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. 18.2) при изменении положения переключателей SA1, SA2, SA3 соответственно. Запишите программу в ПЛК и убедитесь в ее работоспособности.

Лабораторная работа № 2

Исследование взаимодействия ПЛК и ПТ по протоколу MODBUS TCP

Ознакомьтесь с устройством ПТ *Magelis HMI GTO* и средой программирования *Vijeo Designer*. Создайте проект, позволяющий 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 (рис. 18.2), подключенных к ПЛК с одновременным их дублированием виртуальными лампами на экране ПТ. Загрузите части проекта в ПТ и ПЛК и убедитесь в его работоспособности.

Лабораторная работа № 3

Исследование системы прерываний ПЛК

Создайте проект на основе системы прерываний ПЛК, в котором при срабатывании индуктивного датчика SI1 (рис. 18.2) 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 и убедитесь в его работоспособности.

Лабораторная работа № 4

Исследование способа управления ПЧ через цифровые входы

Ознакомьтесь с программной средой *SoMove* для конфигурации и управления ПЧ *Altivar ATV320*. Произведите начальную конфигурацию ПЧ для работы с двигателем *AIP71A2*. Настройте режим работы ПЧ для получения 16-ти фиксированных скоростей при управлении через цифровые (дискретные) входы. Создайте проект в среде *Machine Expert – Basic*, в котором посредством переключателей SA1–SA6 (рис. 18.2), подключенных ко входам %I0.8–%I0.13 ПЛК, осуществляется управление частотой ПЧ по цифровым входам DI1–DI6, подключенным к выходам %Q0.4–%Q0.9 ПЛК.

Лабораторная работа № 5

Исследование способа управления ПЧ по протоколу Modbus RTU (RS-485)

Задание. Установите связь контроллера *Modicon M221* с преобразователем частоты *Altivar ATV320* по протоколу RS-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 (рис. 18.2), а в качестве элементов индикации – лампы HL6–HL8. Задание частоты ПЧ должно осуществляться от потенциометра RP2. Загрузите проект в ПЛК и убедитесь в возможности пуска, реверса, останова двигателя M1, а также задания необходимой частоты вращения.

Лабораторная работа № 6

Создание простой системы автоматизации

Создайте проект, связывающий терминал *Magelis HMI GTO*, контроллер *Modicon M221* и преобразователь частоты *Altivar ATV320*

в единую систему управления двигателем M1 (рис. 18.2). 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 выводиться сообщение об аварийном режиме.

Лабораторная работа № 7

Создание цикла работы производственного механизма

Измените проект, созданный по заданию 6 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.

Лабораторная работа № 8

Исследование возможностей встроенного ПЛК ATV Logic по созданию производственного цикла

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 (рис. 18.2), а в качестве элементов индикации – лампы HL3–HL5. Убедитесь в работоспособности программы.

Лабораторная работа № 9

Исследование работы с журналом тревог ПТ

Измените проект, созданный в лабораторной работе № 3, таким образом, чтобы всякий раз при срабатывании индуктивного датчика SI1 (рис. 18.2), на экране терминала появлялся журнал тревог с соответствующими записями о срабатываниях датчика.

Лабораторная работа № 10

Исследование работы с журналом событий ПТ

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, изме-

нение вводимых параметров) заносились бы в журнал событий. Загрузите части проекта в ПТ и ПЛК и убедитесь в его работоспособности.

19. Контрольные вопросы по модулю 2

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: учебное пособие / А. В. Герасимов, И. Н. Терюшов, А. С. Титовцев ; Федер. агентство по образованию, Казан. гос. технол. ун-т. – Казань : КНИТУ, 2008. – 169 с. : ил., табл., схемы. – URL: <https://biblioclub.ru/index.php?page=book&id=258921> (дата обращения: 11.12.2021).

3. Сергеев, А. И. Программирование контроллеров систем автоматизации : учеб. пособие / А. И. Сергеев, А. М. Черноусова, А. С. Русяев ; Оренбург. гос. ун-т. – Оренбург : Оренбург. гос. ун-т, 2017. – 126 с. : схемы, табл. – URL: <https://biblioclub.ru/index.php?page=book&id=481806> (дата обращения: 11.12.2021).

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. : ил. – URL: <https://biblioclub.ru/index.php?page=book&id=499053> (дата обращения: 11.12.2021).

5. Mitsubishi Alpha 2. Руководство по аппаратной части. Изд-во компании Mitsubishi Electric, 2005. – 131 с.

6. Mitsubishi Alpha 2. Руководство по программному обеспечению. Изд-во компании Mitsubishi Electric, 2005. – 142 с.

7. Mitsubishi Alpha 2. Руководство по программированию. Изд-во компании Mitsubishi Electric, 2003. – 188 с.

8. Mitsubishi Alpha 2. Руководство по средствам связи. Изд-во компании Mitsubishi Electric, 2005. – 142 с.

9. Логический контроллер Modicon M221. – URL: <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 обращения: 10.11.2024).

10. EcoStruxure Machine Expert – Basic. Generic Functions Library Guide. – URL: <https://www.se.com/ww/en/download/document/EIO0000003289/> (дата обращения: 10.11.2024).

11. SoMachine Programming Guide. – URL: <https://www.se.com/ww/en/download/document/EIO0000000067/> (дата обращения: 10.11.2024).

12. Горячий FAQ о ПЛК Schneider Modicon M221. – URL: <https://plcontroller.ru/post/6371/> (дата обращения: 10.11.2024).

Учебное электронное издание комбинированного распространения

Учебное издание

Савельев Вадим Алексеевич

ПРОГРАММИРУЕМЫЕ КОНТРОЛЛЕРЫ И ПРОМЫШЛЕННЫЕ СЕТИ

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Электронный аналог печатного издания

Редактор
Компьютерная верстка

Н. Г. Мансурова
И. П. Минина

Подписано в печать 06.04.26.

Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».

Ризография. Усл. печ. л. 6,28. Уч.-изд. л. 6,67.

Изд. № 41.

<http://www.gstu.by>

Издатель и полиграфическое исполнение
Гомельский государственный
технический университет имени П. О. Сухого.
Свидетельство о гос. регистрации в качестве издателя
печатных изданий за № 1/273 от 04.04.2014 г.
пр. Октября, 48, 246746, г. Гомель