

# Упражнение 23. Создание проекта в *Vijeo Designer*

## 1. Программная часть ПЛК

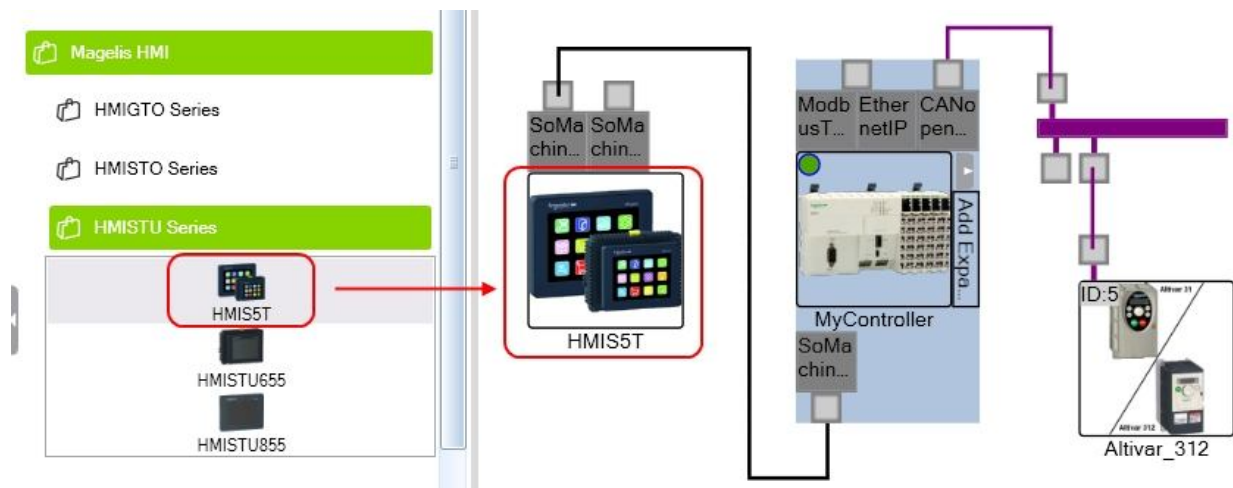
Для ознакомления с настройками связи ПЛК с панелью оператора по протоколу *SoMachine* выберем в качестве объекта управления преобразователь частоты *Altivar 312*, который соединен с ПЛК полевой шиной нижнего уровня *CANopen*, в проекте предусмотрим вычисление положения вала оси вращения, а также скорости вращения с помощью энкодера, который подключен к быстрым входам ПЛК.

Дополнительные задачи, которые необходимо реализовать с помощью панели оператора в упражнении:

- контроль доступа к странице управления приводом ;
- создание компонента тревоги (*Alarm*) при обрыве связи с преобразователем частоты;
- создание страницы тренда скорости движения привода.

1.1 Создайте в *SoMachine* новый проект с ПЛК *M258* и языком программирования *CFC* в программных модулях *POU*, назовите проект "*23 1 Vijeo\_Designer*".

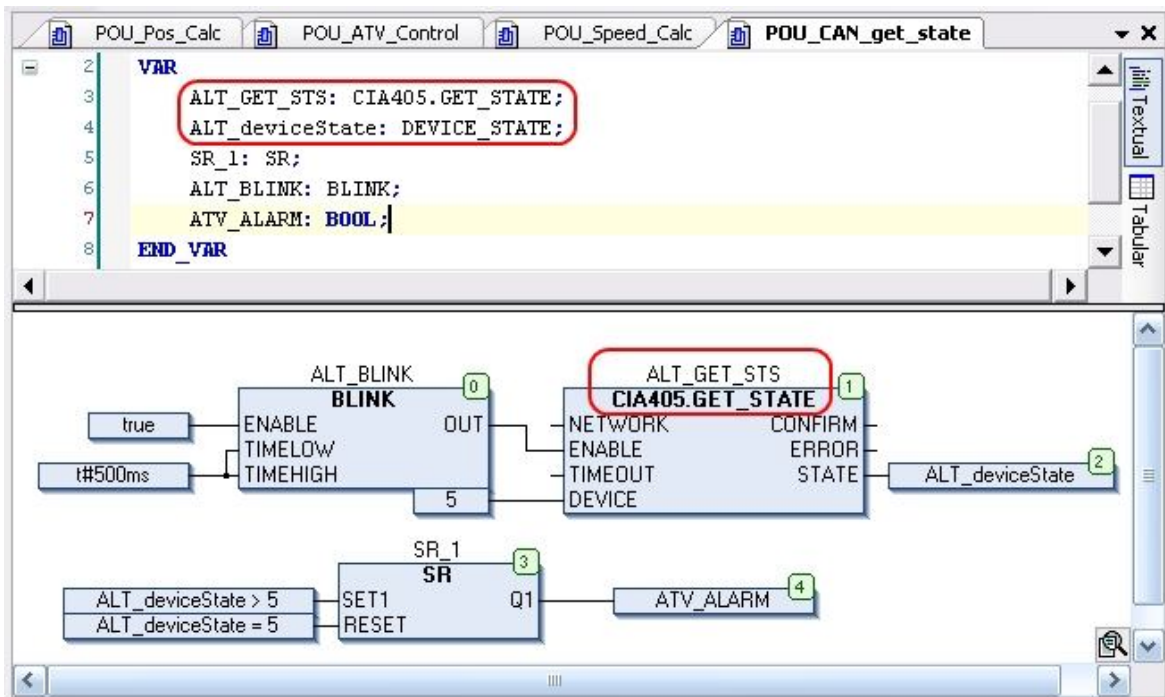
1.2 Добавляйте в графическом конфигураторе шину *CANopen* к порту *CANopen* ПЛК, а затем к шине *CANopen* добавляйте ведомое устройство *Altivar 312*, настроив его сетевой адрес (на рисунке адрес ПЧ *Altivar 312* имеет значение «5», она должна отвечать настройкам ПЧ). Добавьте в проект графический терминал *HMIS5T*, предусмотрев соединение панели *COM1* с последовательным портом ПЛК.



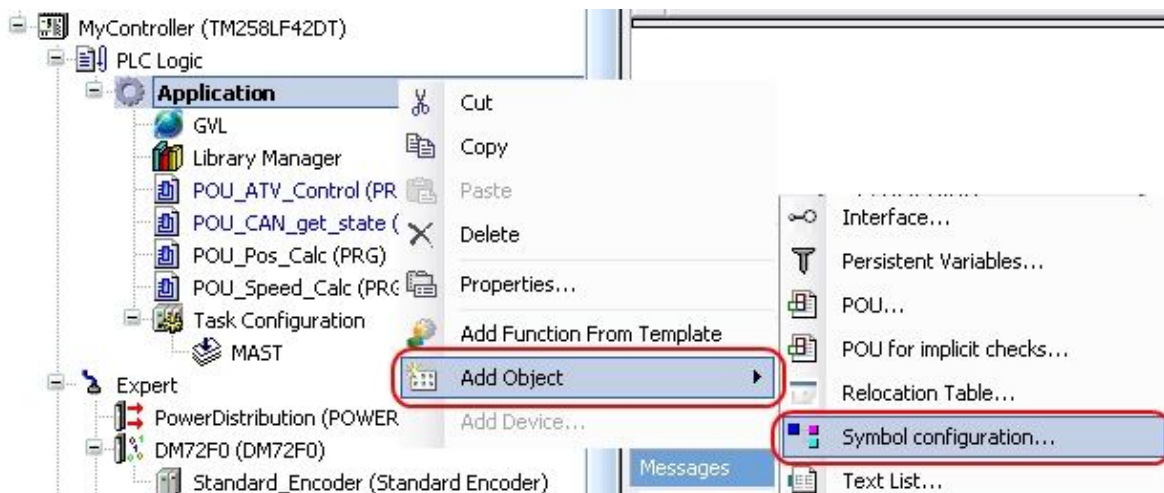
1.3 Создайте в проекте программные модули *POU* для управления преобразователем частоты *POU\_ATV\_control*, вычисления скорости вращения *POU\_Speed\_calc* и положения вала *POU\_Pos\_calc*, используя импорт/экспорт программ из предыдущего упражнения «*22\_1 Encoder*».

1.4 Создайте программу на языке *CFC* для получения сведений о состоянии узла *CANopen* (преобразователя частоты) с именем *POU\_CAN\_get\_state*. Добавьте из библиотеки *CiA405* в поле программы функциональный блок *CIA405.GET\_STATE* и присвойте ему экземпляр, например *ALT\_GET\_STS*. Опрос узла сети *CANopen* под номером «5» (см. рис.) осуществляется по переднему фронту импульса на входе указанного функционального блока *ENABLE*. Создайте в программе пульсирующий блок типа *BLINK*, который обеспечит периодическую проверку состояния узла.

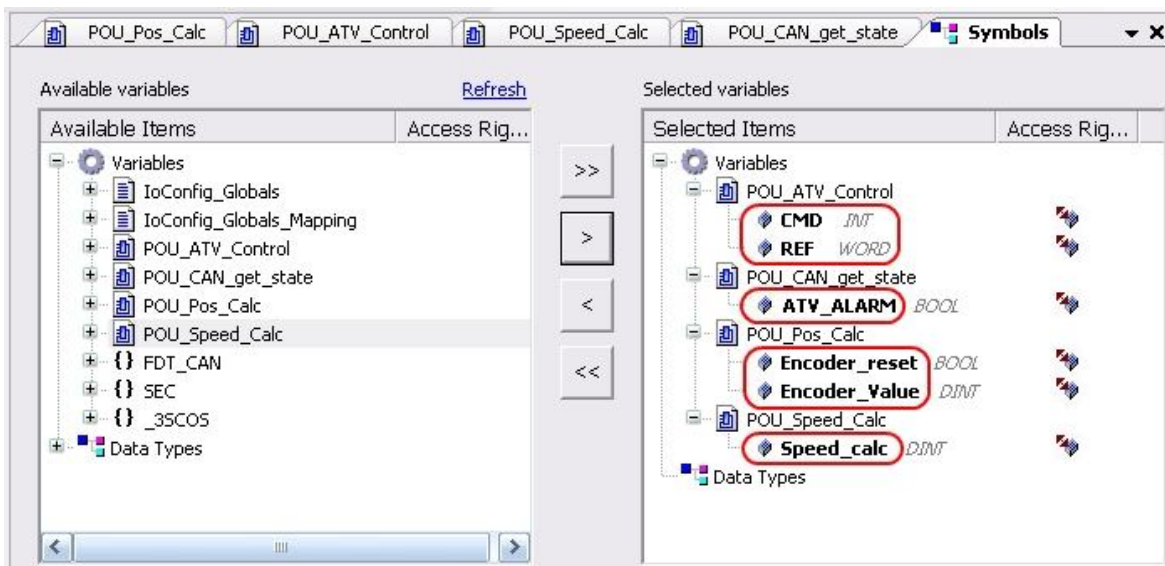
Текущее состояние узла отображается в выходной переменной типа *DEVICE\_STATE*, значения которой в зависимости от состояния узла указаны в соответствующем разделе помощи. В общем случае при отказе узла, потери связи с ним переменная типа *DEVICE\_STATE* принимает значение «>5», а при нормальной работе без ошибок «=5». Поэтому в предложенной программе дискретной переменной *ATV\_ALARM* будет присваиваться значение лог. 1 при выполнении условия *ALT\_deviceState > 5*, а сброс переменной *ATV\_ALARM* – после возобновления нормальной работы узла, т.е. *ALT\_deviceState = 5*. Для объединения логических действий в примере использован функциональный блок триггера *SR*.



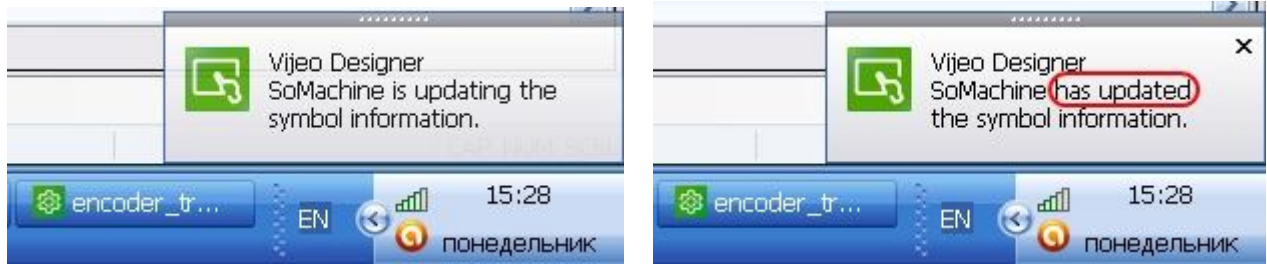
1.5 В дереве проекта для ветви *Application* с помощью команды *Add object...* добавьте узел *Symbol configuration*.



Откройте окно созданного узла проекта и сконфигурируйте его, связав переменные со средой *Vijeo Designer* по примеру:



Перейдите в окно конфигурации приложения панели оператора *Vijeo Designer* и дождитесь обновления созданной таблицы символов программы – процесс обновления отображается в виде всплывающих сообщений в правом нижнем углу экрана.



1.6 Откомпилируйте проект и загрузите его в контроллер, проверьте работу программной части ПЛК.

- преобразователь частоты в сети *CANopen*:

Expression	Type	Value	Prepared value	Comment
ALT_GET_STS	CIA405.GET_STATE			
ALT_deviceS...	DEVICE_STATE	OPERATIONAL		
SR_1	SR			
ALT_BLINK	BLINK			
ATV_ALARM	BOOL	FALSE		

The ladder logic diagram shows the following logic:

- ALT\_BLINK (0)**: A BLINK block with inputs 'true' and '#500ms'. Its output 'OUT' is connected to the 'NETWORK' input of **ALT\_GET\_STS (1)**.
- ALT\_GET\_STS (1)**: A CIA405.GET\_STATE block with inputs 'ENABLE', 'TIMEOUT', and 'DEVICE'. Its output 'STATE' is connected to the 'ALT\_deviceState (2)' variable, which is set to 'OPERATIONAL'.
- SR\_1 (3)**: An SR (Set-Reset) block with inputs 'SET1' and 'RESET'. The 'SET1' input is connected to the expression 'ALT\_deviceState > 5'. The 'RESET' input is connected to the expression 'ALT\_deviceState = 5'. Its output 'Q1' is connected to the 'ATV\_ALARM (4)' variable, which is set to 'FALSE'.

- преобразователь частоты не в сети *CANopen*:

Expression	Type	Value	Prepared value	Comment
ALT_GET_STS	CIA405.GET_STATE			
ALT_deviceState	DEVICE_STATE	NOT_AVAL		
SR_1	SR			
ALT_BLINK	BLINK			
ATV_ALARM	BOOL	TRUE		

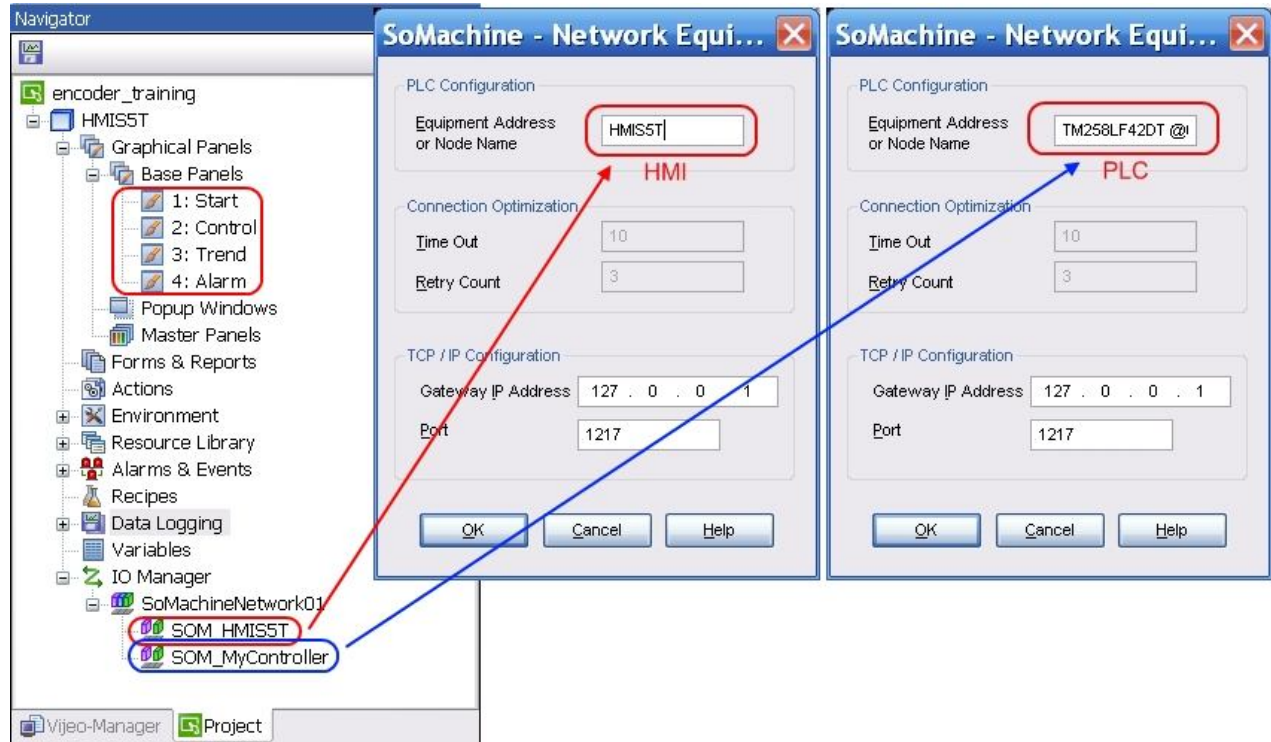
The ladder logic diagram shows the following logic:

- ALT\_BLINK (0)**: A BLINK block with inputs 'true' and '#500ms'. Its output 'OUT' is connected to the 'NETWORK' input of **ALT\_GET\_STS (1)**.
- ALT\_GET\_STS (1)**: A CIA405.GET\_STATE block with inputs 'ENABLE', 'TIMEOUT', and 'DEVICE'. Its output 'STATE' is connected to the 'ALT\_deviceState (2)' variable, which is set to 'NOT\_AVAL'.
- SR\_1 (3)**: An SR (Set-Reset) block with inputs 'SET1' and 'RESET'. The 'SET1' input is connected to the expression 'ALT\_deviceState > 5'. The 'RESET' input is connected to the expression 'ALT\_deviceState = 5'. Its output 'Q1' is connected to the 'ATV\_ALARM (4)' variable, which is set to 'TRUE'.

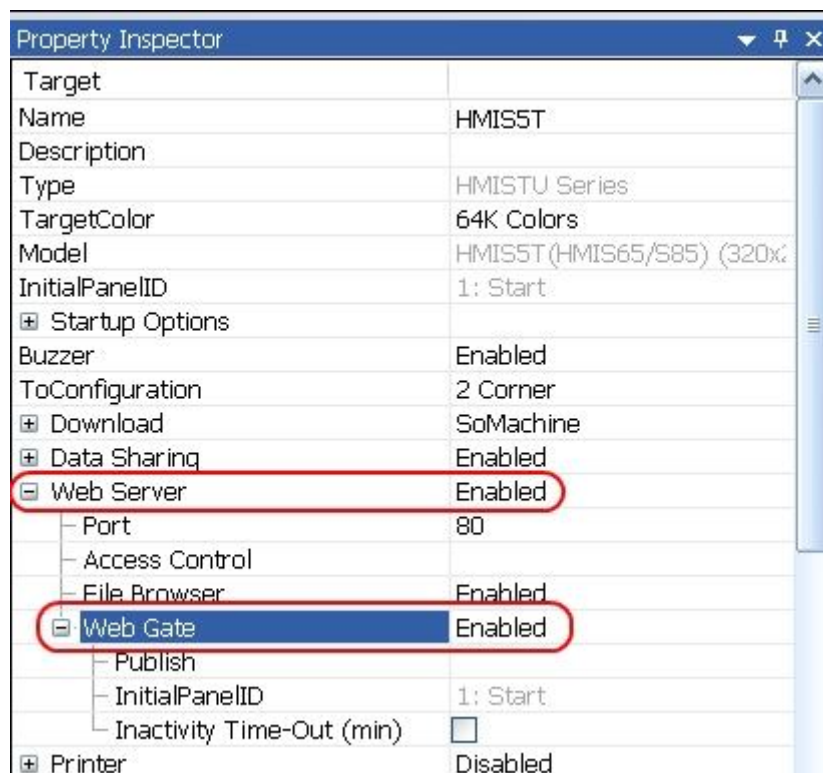
## 2. Настройка интерфейса HMIS5T

2.1 Выполните команду *Logout* и перейдите в среду *Vijeo Designer* для создания операторского интерфейса. С помощью дерева проекта создайте в узле *Base Panels* страницы проекта с именами *Start*, *Control*, *Trend* и *Alarm*. В свойствах проекта выберите начальную страницу *Start*, которая будет использоваться для ввода данных зарегистрированных пользователей и предоставления им доступа к остальным страницам приложения.

2.2 Для устройств сети *SoMachineNetwork* (между ПЛК и панелью оператора) необходимо задать их сетевые адреса или имена. В поле *Equipment Address or Node Name* для панели оператора вводите ее название (*HMIS5T*), а для ПЛК введите его *ID*, определенный с помощью *Gateway*.



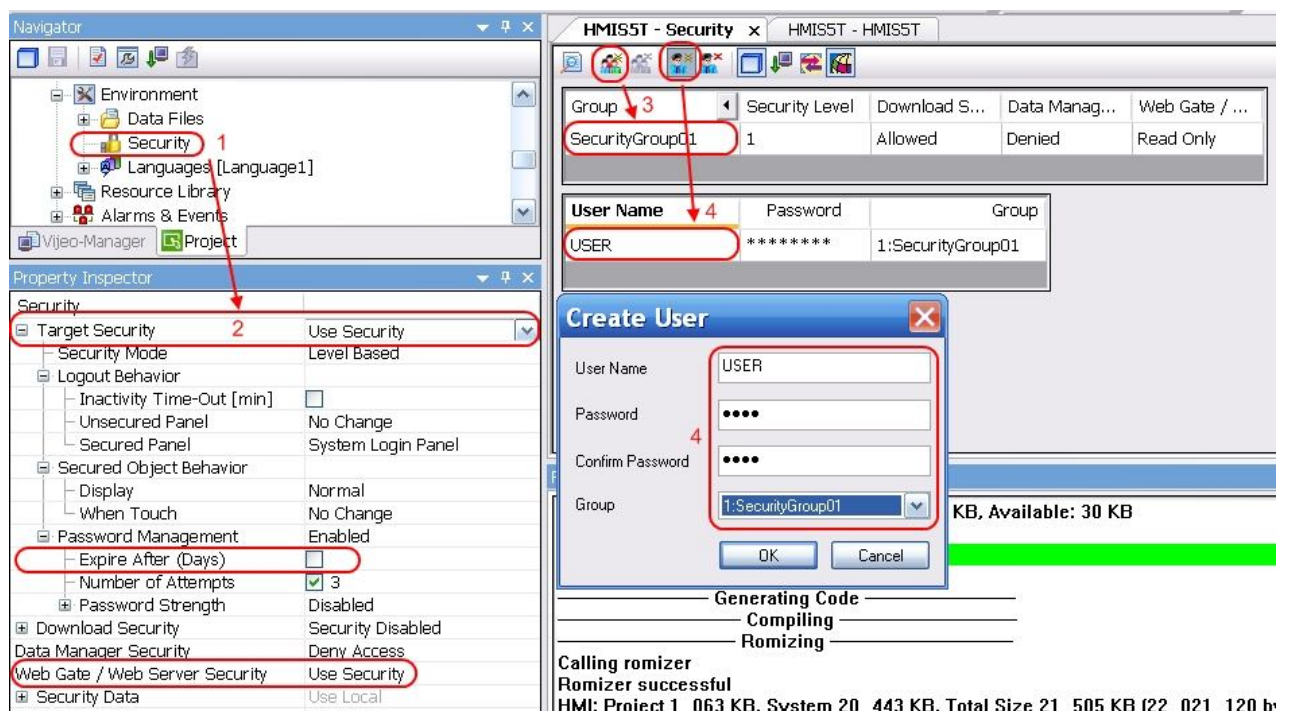
2.3 Для возможности удаленного просмотра приложения панели и управления процессом с помощью технологии *Web Gate* установите в свойствах проекта разрешения для *Web Server (Enabled)* и *Web Gate (Enabled)*.



2.4 В свойствах проекта настройте способ настройки время панели на ручной режим (*Time Management = Enabled; Type = Manual Configuration*). Указанное действие необходимо для коррекции времени панели, если в ПЛК отсутствует программный модуль, передающий автоматически время ПЛК в системные переменные исполнительной среды *Runtime* панели.



2.5 С помощью дерева проекта перейдите в настройки безопасности проекта (группа *Security* в узле *Environment*). Создайте в системе новую группу *SecurityGroup01* с уровнем безопасности 1 (из 8). Добавьте пользователя в созданную группу, присвоив ему имя и пароль, как указано на рис. В свойствах безопасности выберите также использование защиты для удаленного доступа по *Web Gate (Use Security)*, ознакомьтесь с остальными настройками безопасности для приложения, применяя их по желанию (требованию тех. задания).



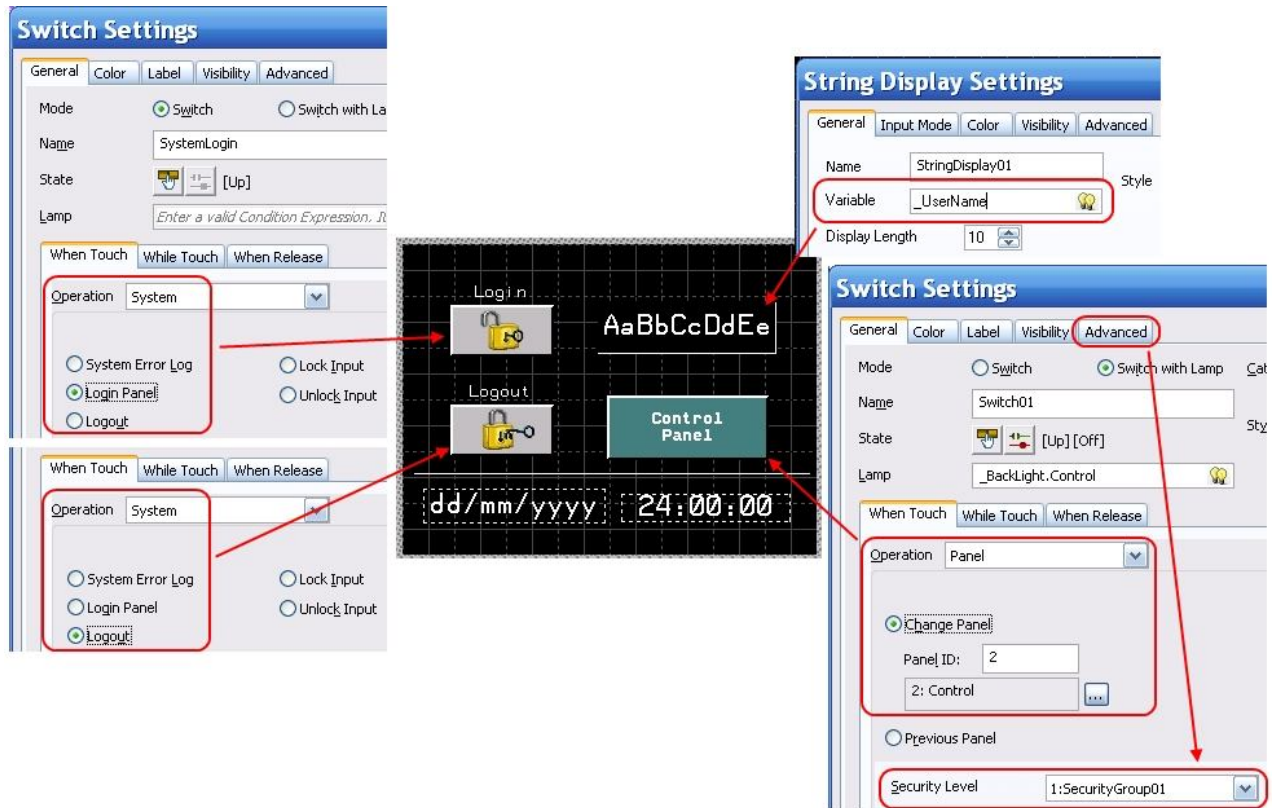
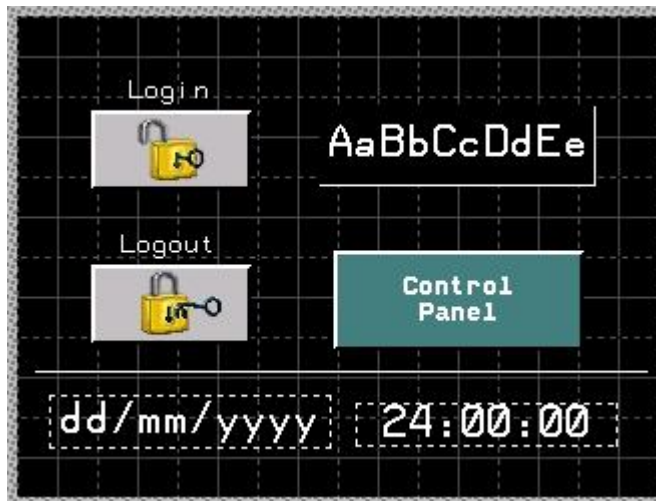
## 2.6 Используя графические объекты

- *String Display* (отображение имени текущего пользователя);
- *Switch* (вход / выход пользователя в систему, защищенная кнопка перехода на страницу управления процессом);
- *Date Display* (отображение текущей даты);
- *Time Display* (отображение текущего времени);

сконфигурируйте первую страницу проекта под названием *Start*. Пример страницы приведен ниже.

Настройте созданные элементы в соответствии с их назначением по указанному примеру.

Для перехода между страницами используется действие переключателя *Switch Operation: Panel*, для входа выхода пользователя используется действие *Operation: System (Login, Logout)*. Настройки защиты от несанкционированного доступа находятся в вкладке *Advanced* свойств графических объектов в поле *Security Level*.



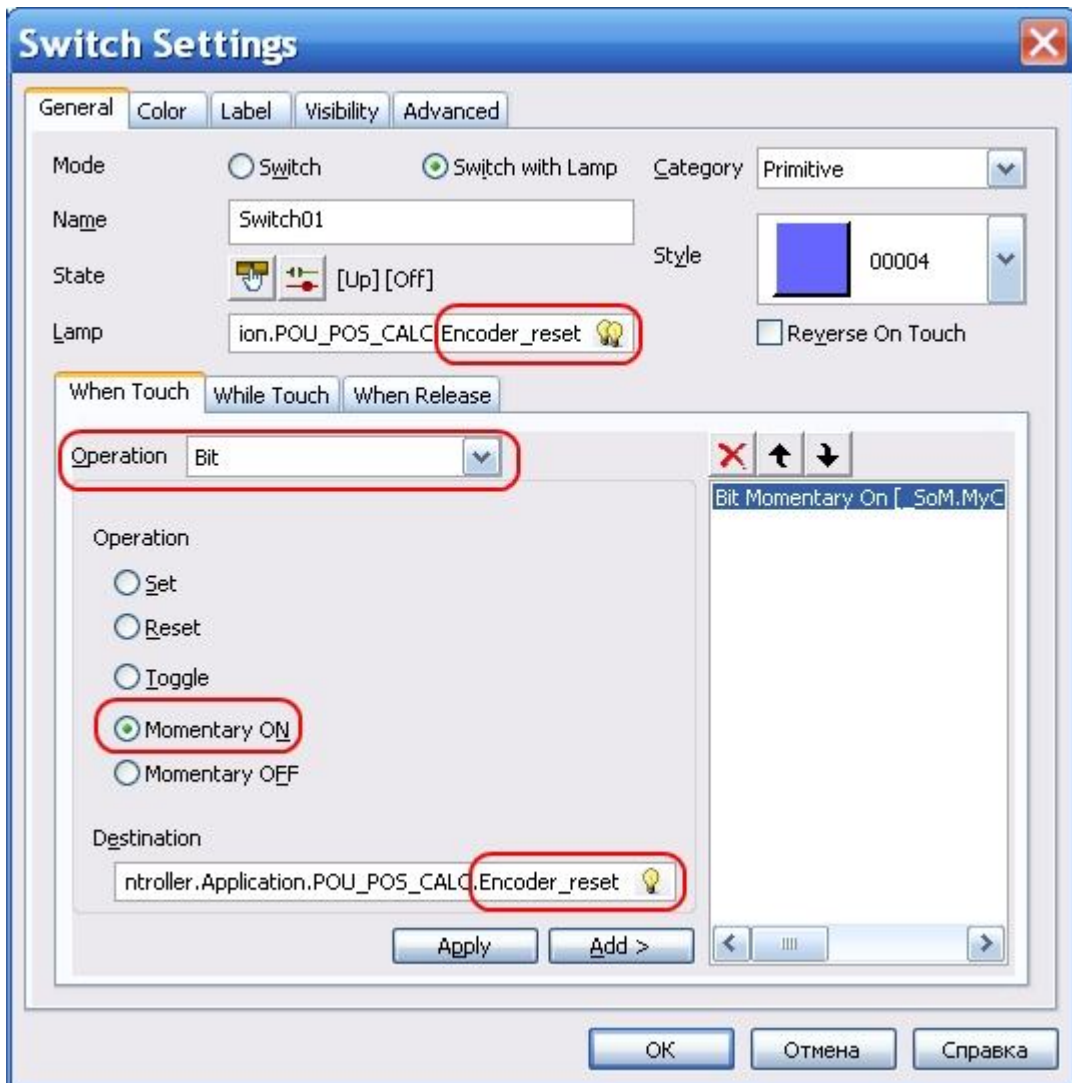
2.7 Перейдите на главную страницу проекта *Control Panel* и сконфигурируйте ее в соответствии с задачами по следующему примеру.



Пример страницы управления содержит следующие графические элементы:

- *Switch* (кнопка «Encoder reset» сброса текущего положения оси, кнопка «ESC» возврата на стартовую страницу, кнопки «Fwd», «Rev» и «Stop» управления режимом работы привода, кнопки «Trend Panel» и «Alarm Panel» для перехода на соответствующие страницы, кнопки «▲» и «▼» для быстрого задания скорости вращения);
- *Bar Graph* (линейное отображение заданной скорости привода);
- *Numeric Display* (цифровое отображение заданной скорости привода с возможностью точного ввода заданной скорости);
- *Meter* (отображение фактической скорости привода с помощью шкалы в об/мин);
- *3-D Box* из библиотеки образов *Vijeo Designer (Toolchest Folders / Containers)* – для отображения текущей позиции механизма, определенного с помощью энкодера.

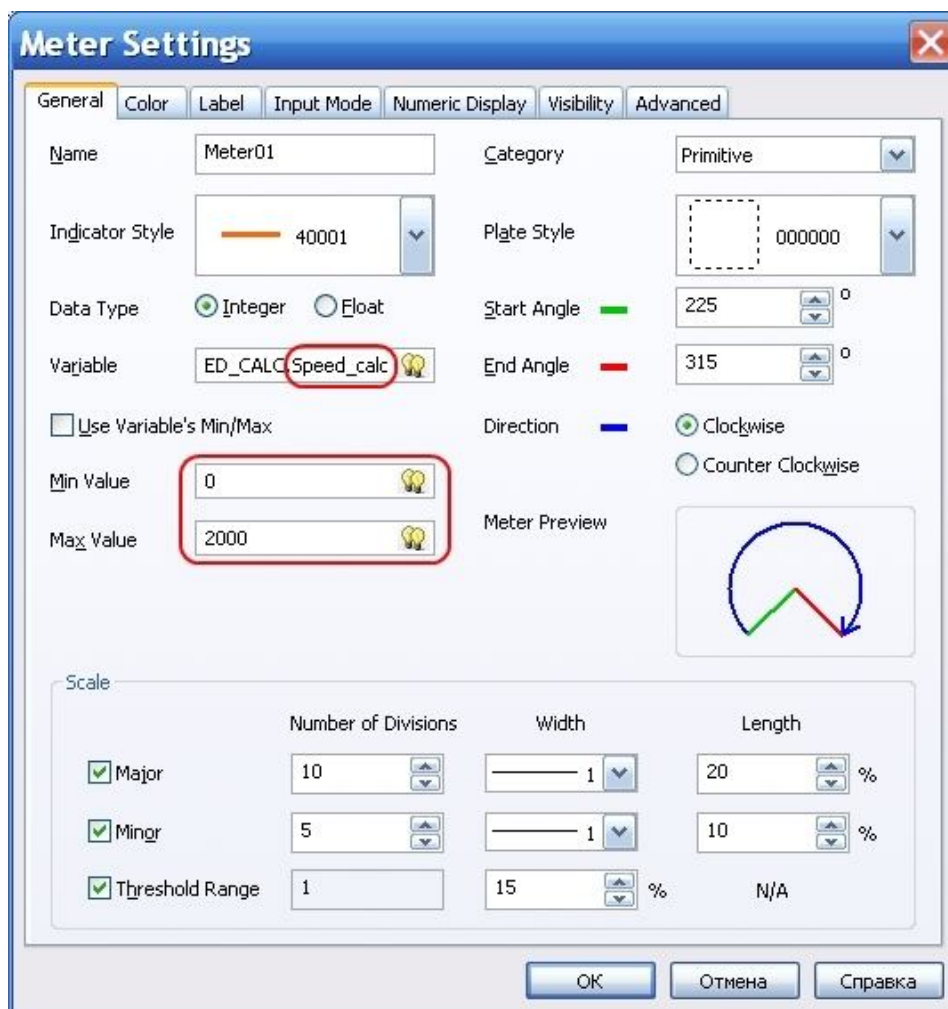
Примеры настройки графических элементов для выполнения поставленных задач приведены ниже.



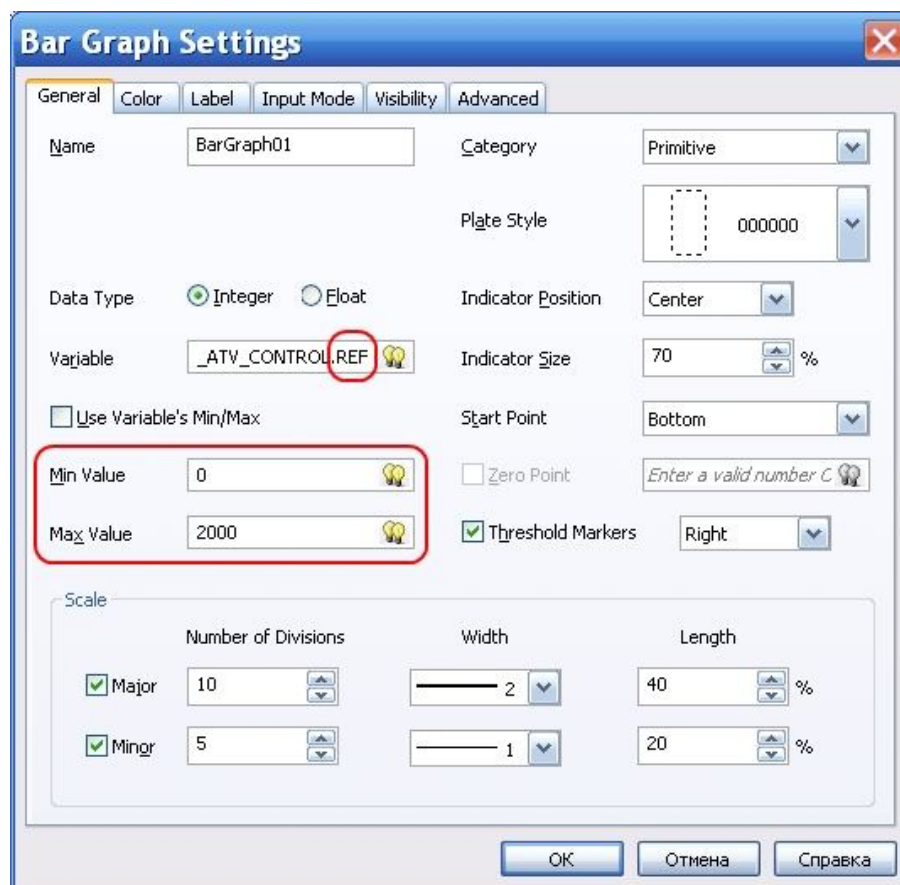
Настройки кнопки сброса текущего положения оси рабочего механизма «Encoder reset»



Настройки графического элемента *Numeric Display* для точного ввода заданной скорости

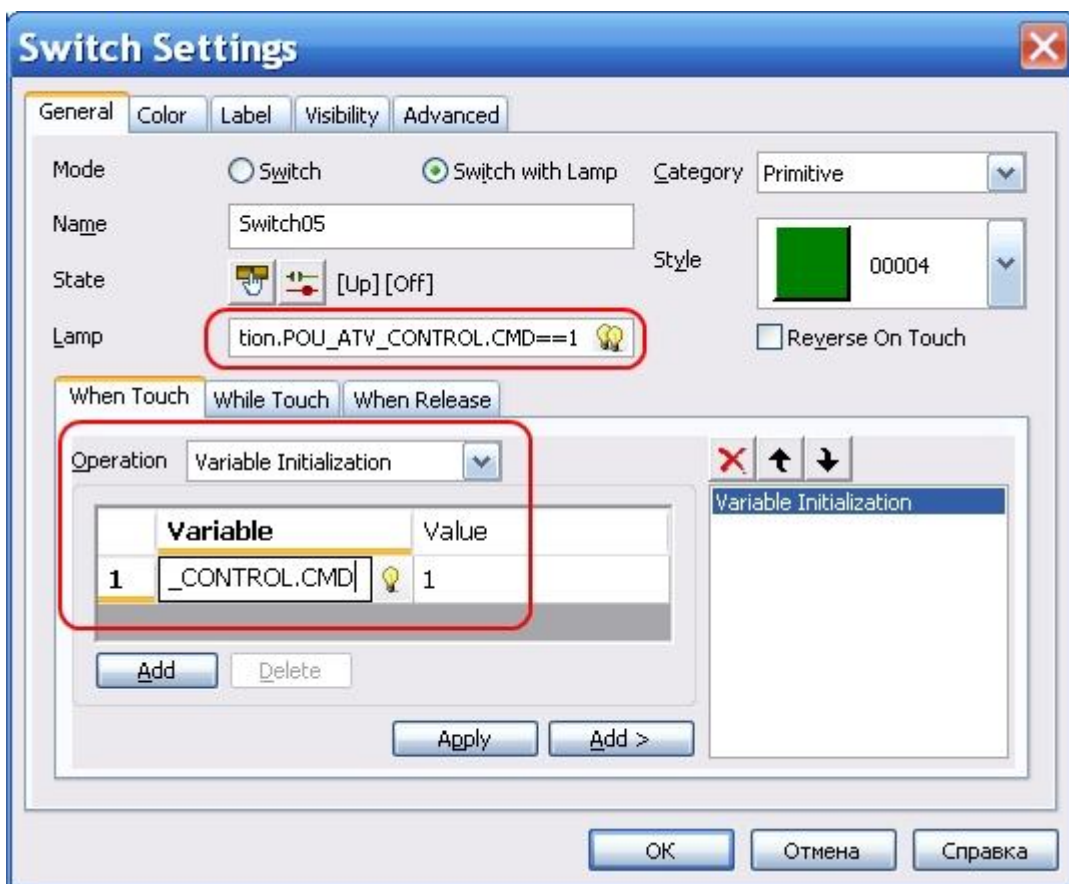


Настройки графического элемента *Meter* для отображения текущей скорости

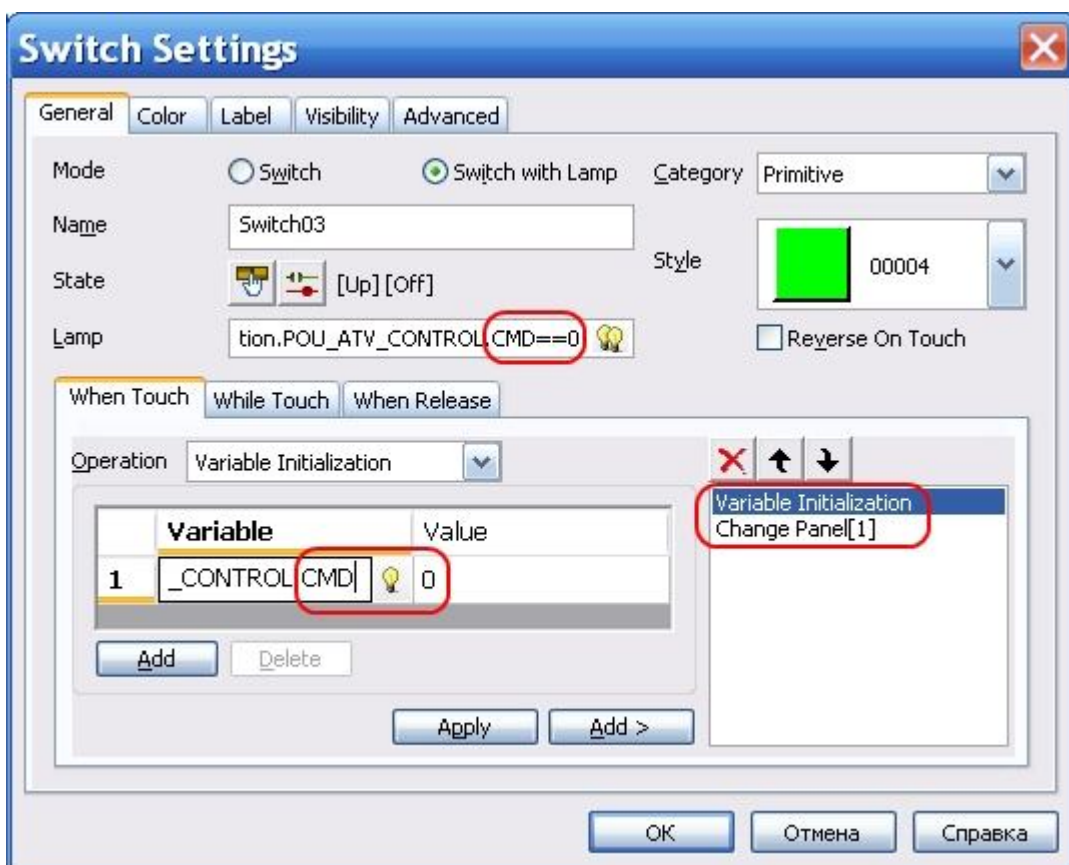


Настройки графического элемента *Bar Graph* для линейного отображения заданной скорости

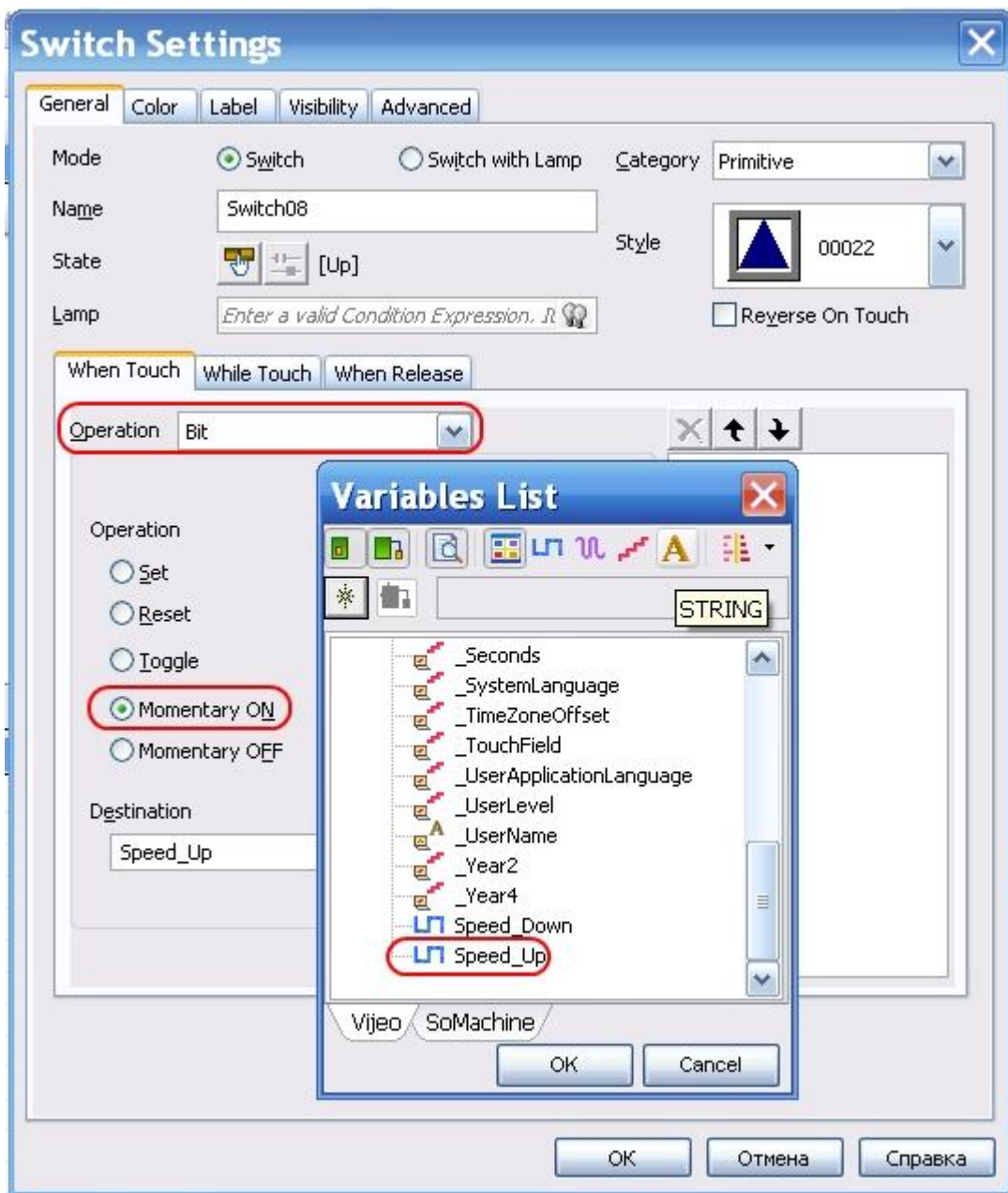




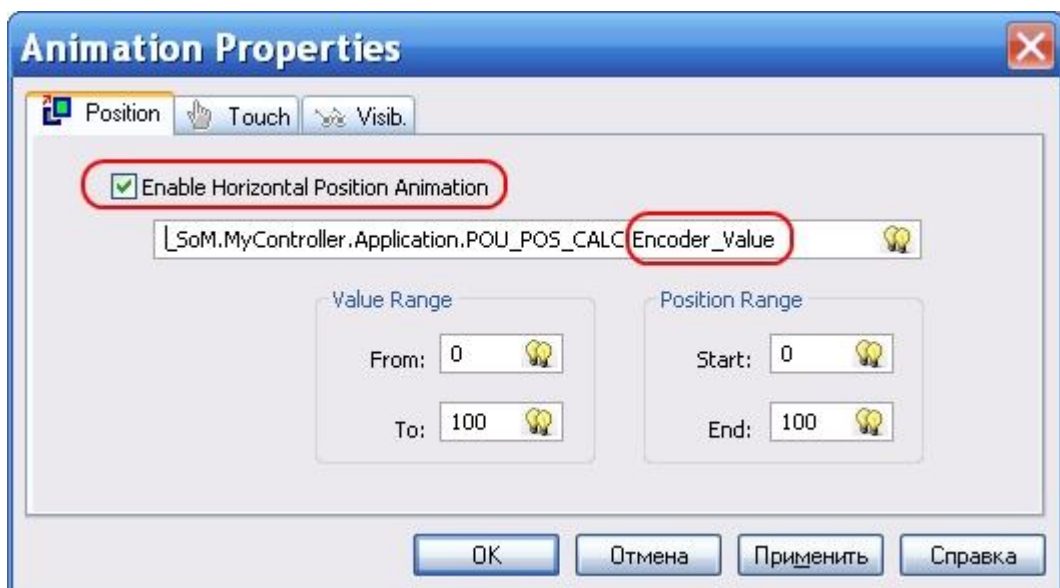
Настройки кнопки «Fwd» для задания движения привода вперед



Настройки кнопки «ESC» для возврата на стартовую страницу и одновременной остановки привода



Настройки кнопки «▲» для быстрого изменения задания скорости вращения



Настройки графического элемента из библиотеки образов для динамического горизонтального перемещения в соответствии с переменной «Encoder\_Value»

В проекте предусмотрено быстрое изменение задания путем нажатия кнопок «▲» или «▼». Для решения задачи можно использовать действие со скриптом положительного или отрицательного приращения заданной скорости с шагом, например, 50 об/мин. Для создания такого сценария необходимо выполнить несколько последовательных процедур:

а) добавить новое действие (*Actions*) с помощью дерева проекта, назначить выполнение действия на внутреннюю логическую переменную *Speed\_Up* (связанную с кнопкой «▲») и периодичность выполнения действия, например 0,5 сек., т.е., при нажатой кнопке «▲» (*Speed\_Up = true*) задание скорости будет увеличиваться на 50 об/мин через каждые 0,5 сек.



Добавление нового действия в проект

б) связать действие с выполнением нового скрипта;



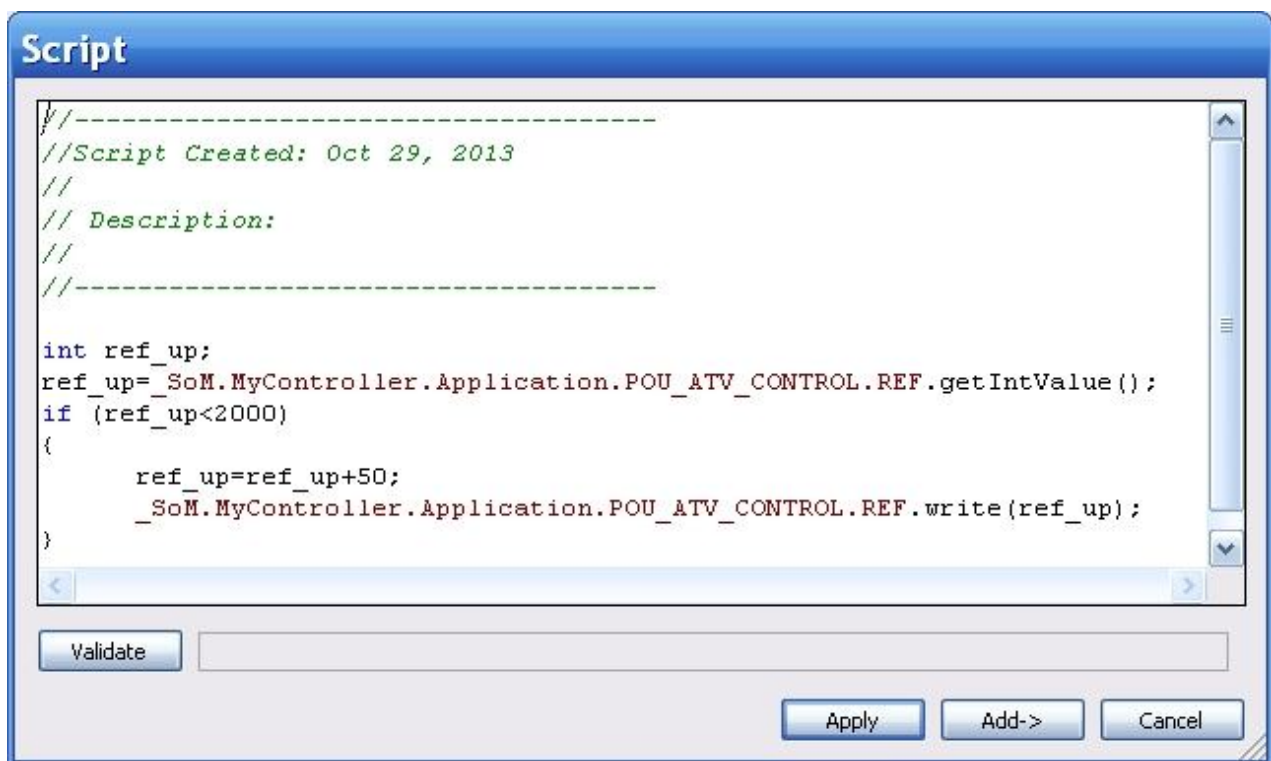
Привязка действия к скрипту

в) используя синтаксис языка программирования *Java* создать скрипт для приращения скорости.

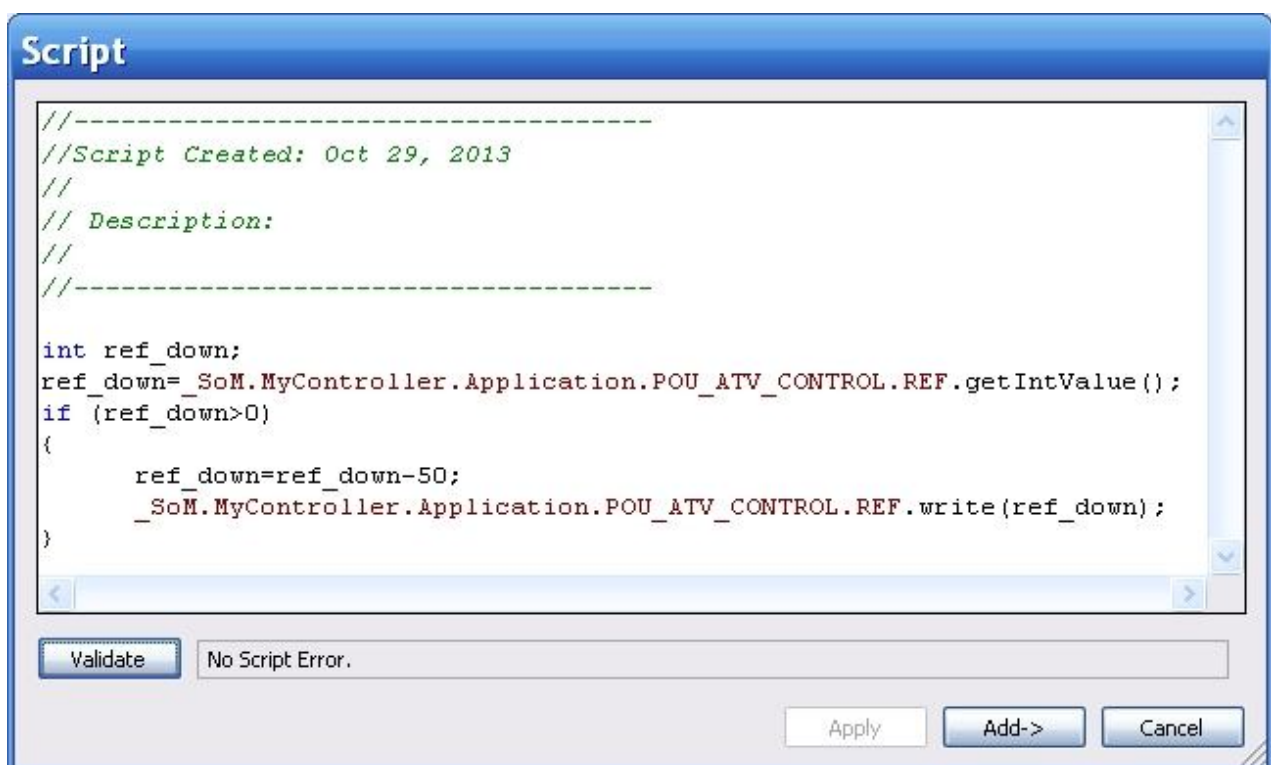
Небольшой скрипт можно условно разделить на четыре части:

- объявление внутренних переменных скрипта;
- установка исходных значений переменным скрипта, если исходные значения берутся из переменных *Vijeo Designer* или *SoMachine* – используется метод *getIntValue()*;
- выражение скрипта (логические инструкции, математические вычисления, работа с массивами и структурами, условные и безусловные переходы и т.п.);
- запись значений переменных скрипта в переменные *Vijeo Designer* или *SoMachine* – используется метод «Переменная *Vijeo Designer*». *write*(«переменная скрипта»).

Для контроля соответствия созданного скрипта синтаксису *Java* используйте кнопку «*Validate*».

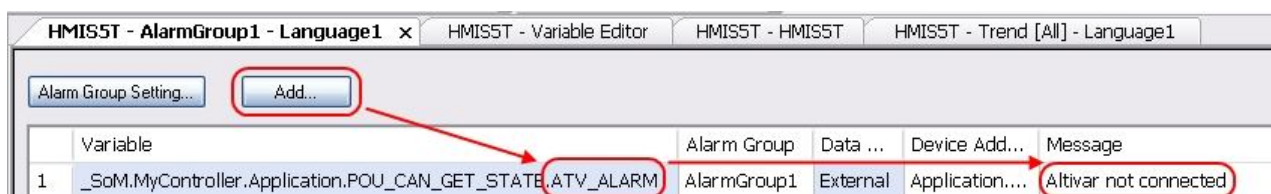



Пример скрипта для положительного приращения задания скорости (кнопка «▲»).

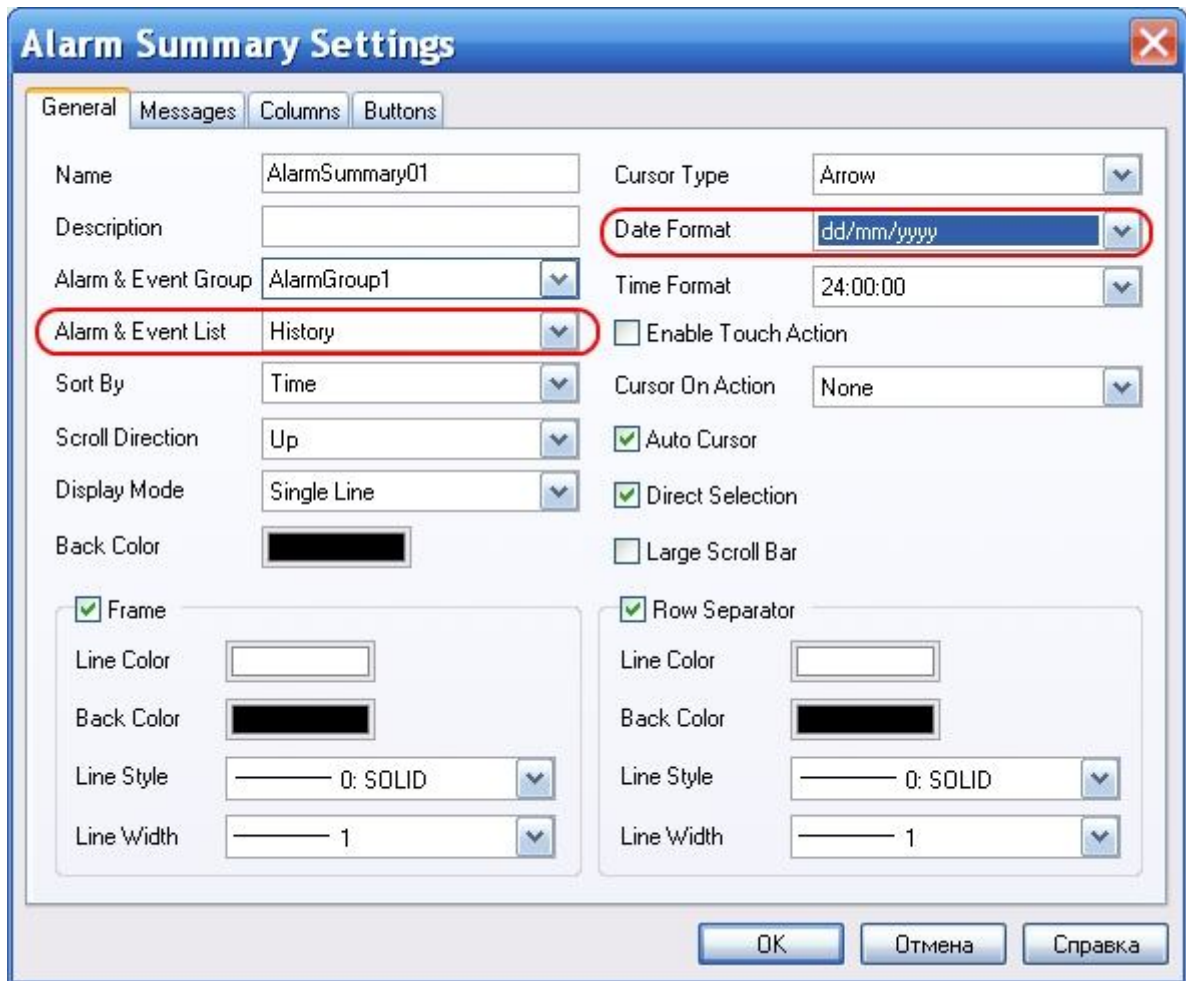


Пример скрипта для отрицательного приращения задания скорости (кнопка «▼»).

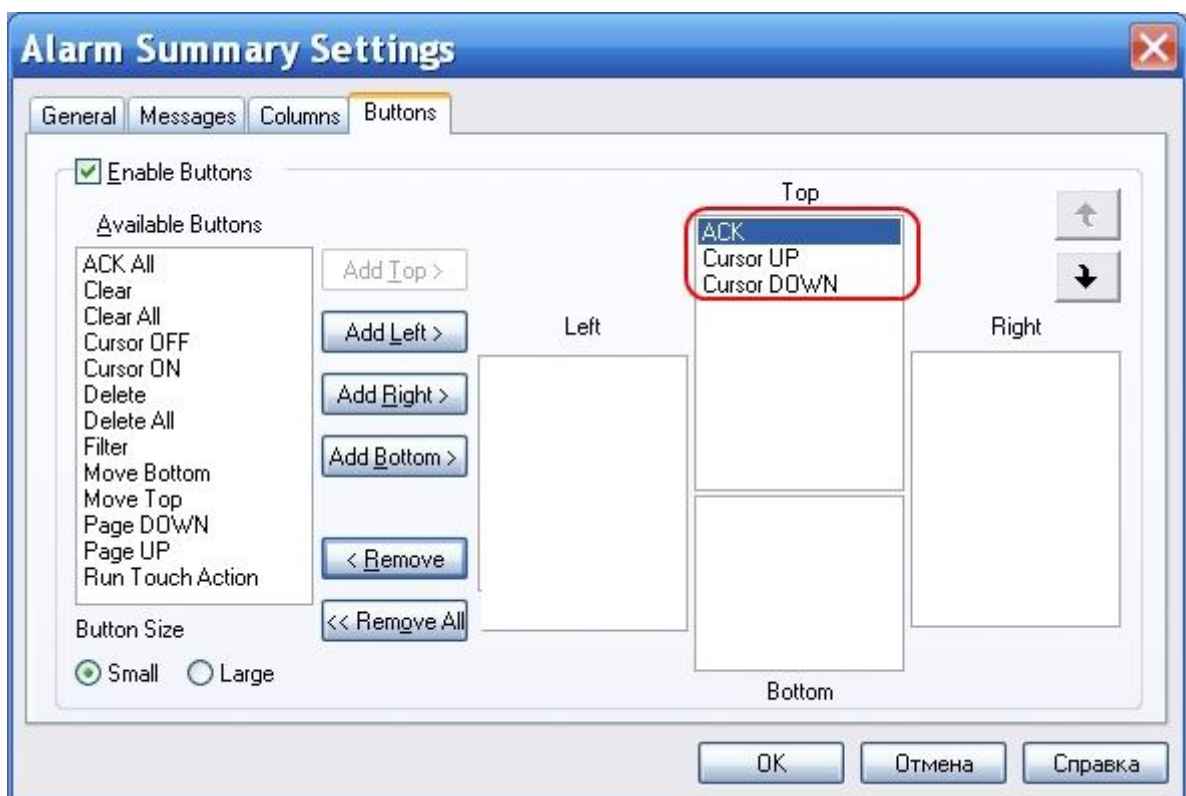
2.8 Конфигурацию страницы тревог *Alarm Panel* следует начать с настройки группы тревог *Alarm Group* из дерева проекта и добавления в группу новой переменной *ATV\_ALARM*, а также формирования сообщения тревоги. Пример указанных действий приведен ниже.



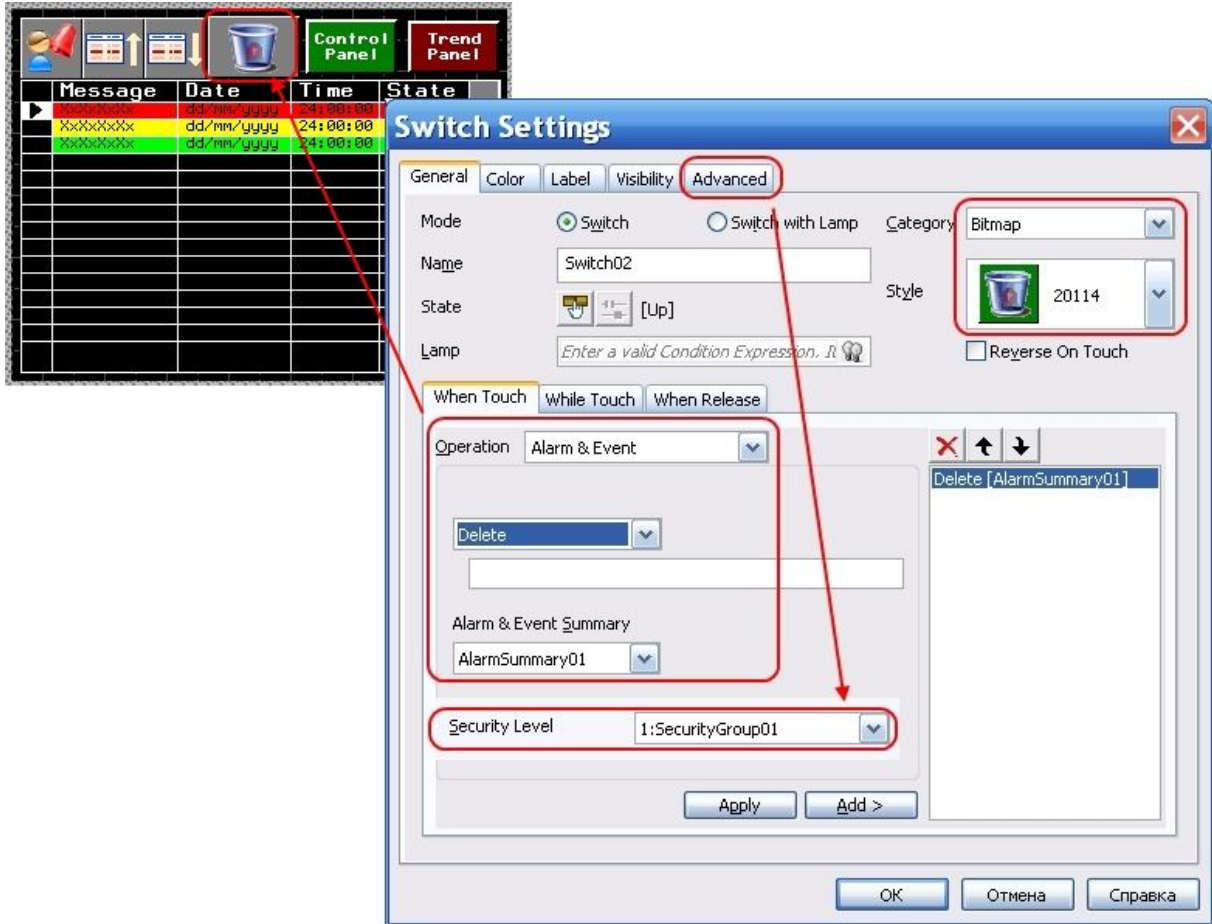
На пустой странице *Alarm Panel* разместите таблицу *Alarm Summary*, используя соответствующую кнопку  на панели инструментов *Vijeo Designer*. Настройте таблицу истории тревог следующим образом.




Настройте сервисные кнопки сводной таблицы истории тревог следующим образом:



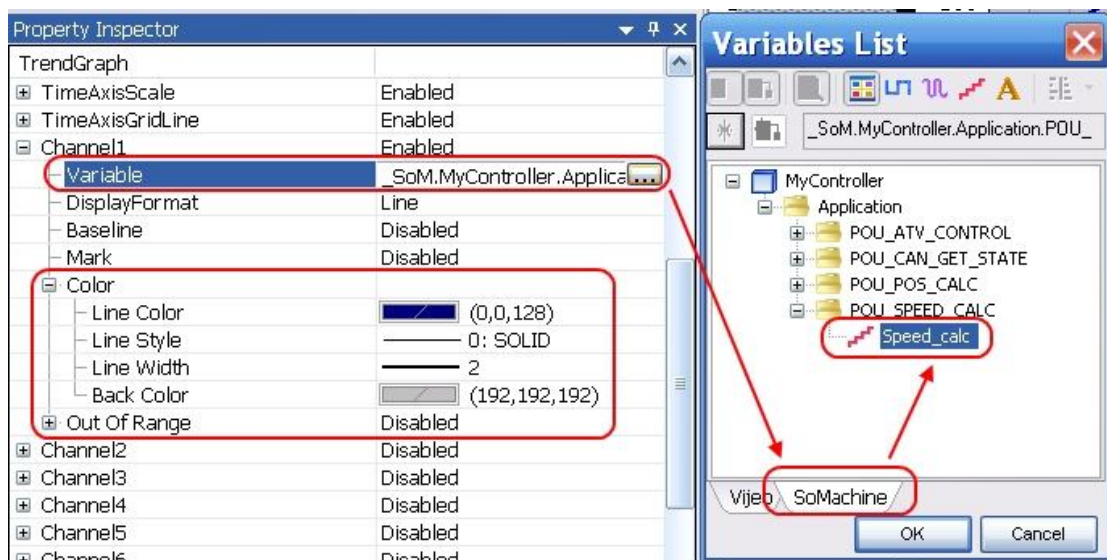
Отдельно создайте на странице *Alarm Panel* кнопку удаления записи из списка тревог, назначив ее на выполнение действия *Delete* из категории *Alarm & Event*. Сделайте действие доступным только для разрешенных пользователей группы безопасности *Security Group*. В рассматриваемом примере вход на главную страницу *Control Panel* и переход из нее на другие страницы доступен только зарегистрированному пользователю, поэтому кнопка удаления будет работать всегда. Для ограничения доступа к удалению записей из списка тревог в других случаях нужно создать новую группу безопасности с более высоким уровнем доступа.



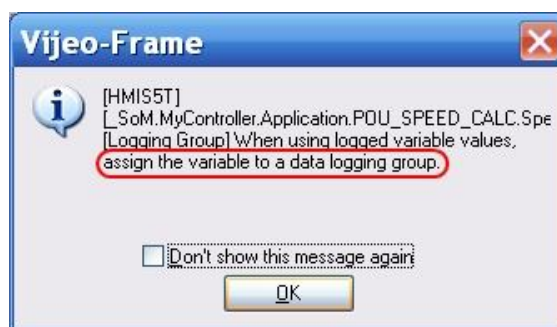
Создайте на странице *Alarm Panel* кнопки возврата на страницу *Control Panel* и перехода на страницу трендов *Trend Panel* с соответствующими надписями.

2.9 Перейдите на страницу *Trend Panel* и разместите на свободном поле графический элемент типа *Trend* с помощью соответствующей кнопки на панели инструментов .

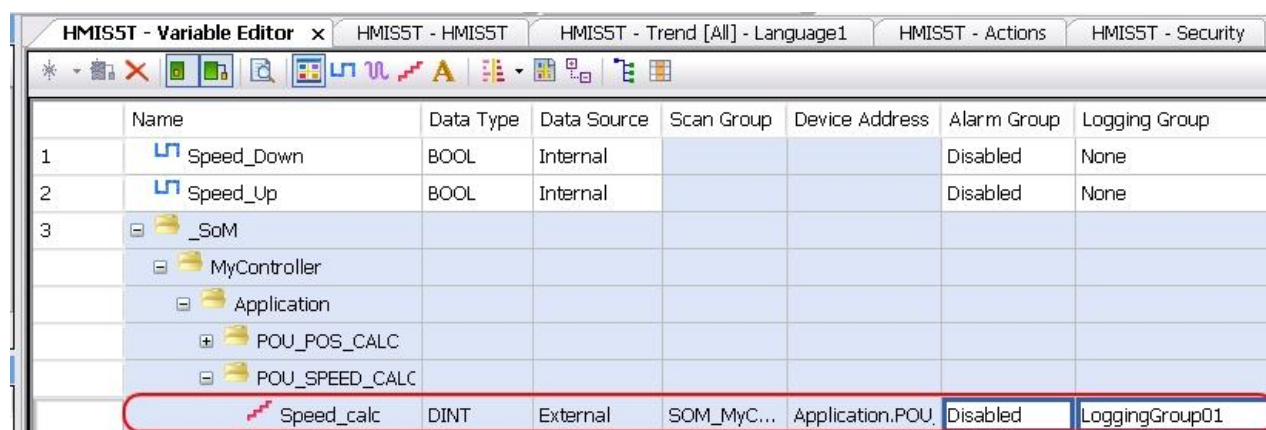
Привяжите канал тренда к переменной фактической скорости привода *Speed\_calc*. Настройте цвет и толщину линий тренда.



Выбранную переменную канала тренда следует внести в группу переменных, значения которых следует сохранять в памяти панели за предыдущие промежутки времени. Эта группа автоматически присутствует в дереве проекта *Vijeo Designer* под названием *Logging Group*, но переменную *Speed\_calc* нужно внести в список переменных группы отдельно. *Vijeo Designer* формирует об этом сообщение отдельно.



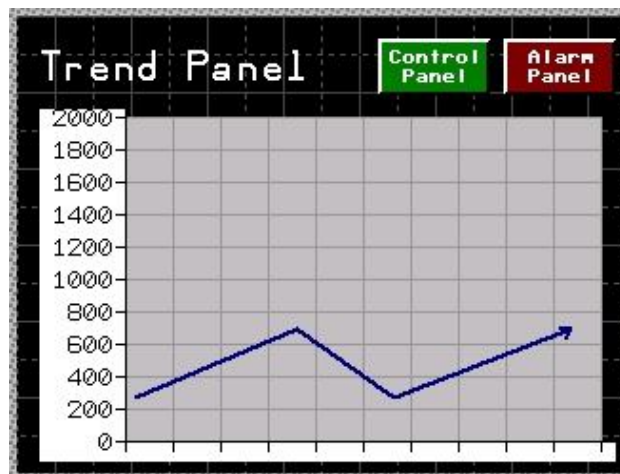
В редакторе переменных *Speed\_calc* вносится в список регистрируемых переменных в колонке *Logging Group*.



Из дерева проекта откройте окно настроек группы регистрируемых переменных *Logging Group* и настройте ее по указанному ниже примеру.



Пример страницы трендов приведен ниже. Предусмотрите кнопки перехода из страницы трендов на страницы *Control Panel* и *Alarm Panel* с помощью графических элементов *Switch*. Заглавие страницы создается с помощью графического элемента типа *Text*.



Пример страницы *Trend Panel*

2.10 Сохраните проект из среды *SoMachine* и выполните симуляцию проекта для отладки. Вызов симуляции осуществляется по команде *Start Device Simulation* из дерева проекта для узла *HMIS5T*.



Проверьте в среде симуляции работу всех элементов динамизации, выполнение скриптов, переходы между страницами и ведение списка тревог.

Variable	Type	Value
<input type="checkbox"/> _SoM.MyController.Application.POU_ATV_CONTROL_CMD	A. INT	1
<input type="checkbox"/> _SoM.MyController.Application.POU_ATV_CONTROLREF	A. UINT	600
<input type="checkbox"/> _SoM.MyController.Application.POU_CAN_GET_STATE.ATV_A...	A. BOOL	0
<input type="checkbox"/> _SoM.MyController.Application.POU_POS_CALC.Encoder_Value	A. DINT	75
<input type="checkbox"/> _SoM.MyController.Application.POU_POS_CALC.Encoder_reset	A. BOOL	0
<input type="checkbox"/> _SoM.MyController.Application.POU_SPEED_CALC.Speed_calc	A. DINT	580

Message	Date	Time	State
Altivar not	29/10/2013	01:19:53	RTN
Altivar not	29/10/2013	01:20:18	ACTIVE

2.11 Перешлите приложение в панель и проверьте его работу. Сохраните проект, создав его резервную копию. Сделайте выводы по работе.