

Е.Б. Андреев

ПРОЕКТИРОВАНИЕ СИСТЕМ УПРАВЛЕНИЯ В **SCADA** - ПАКЕТЕ **In Touch 7.0**



КОМПЬЮТЕРНЫЙ ПРАКТИКУМ

Е.Б. АНДРЕЕВ

**Проектирование систем
управления
в SCADA - пакете InTouch7.0**



Компьютерный практикум

Москва - 2005

УДК 658.52,011.56

Рецензент:

В.Е.Попадьюко, профессор, зав. каф. АТП.

Андреев Е.Б.

Проектирование систем управления в SCADA-пакете InTouch7.

Компьютерный практикум. - М.: РГУ нефти и газа, 2005. – 70 с.

Пакет InTouch является одним из наиболее популярных программных продуктов класса SCADA. В России он нашел свое применение во многих отраслях народного хозяйства, в том числе и в нефтегазовой отрасли. Данный компьютерный практикум предназначен для углубленного ознакомления студентов с проектированием систем управления в пакете InTouch.

Компьютерный практикум предназначен для студентов кафедры АТП, изучающих SCADA-системы в курсе «Автоматизация технологических процессов». Он будет полезен и студентам кафедры АСУ в курсе «Проектирование автоматизированных систем обработки информации и управления», а также слушателям Научного центра по повышению квалификации специалистов.

© Москва, РГУ нефти и газа, 2005

© Андреев Е.Б., 2005

Введение.

Предлагаемый компьютерный практикум дает возможность студентам приобрести практические навыки работы в SCADA - пакете **InTouch**. Практикум адресован студентам, изучающим SCADA - системы в курсе «Автоматизация технологических процессов».

Практикум включает шесть разделов. Каждый из этих разделов состоит из двух частей: краткого теоретического материала по разделу и подробного описания практических действий по выполнению задания раздела. Задания на практическую работу во всех разделах взаимосвязаны между собой, что дает возможность, осваивая различные компоненты пакета **InTouch**, создавать от раздела к разделу различные фрагменты единого проекта. Результат работы с практикумом - шестиоконный интерфейс диспетчера/оператора, позволяющий реализовать базовые функции сбора данных и диспетчерского управления (SCADA).

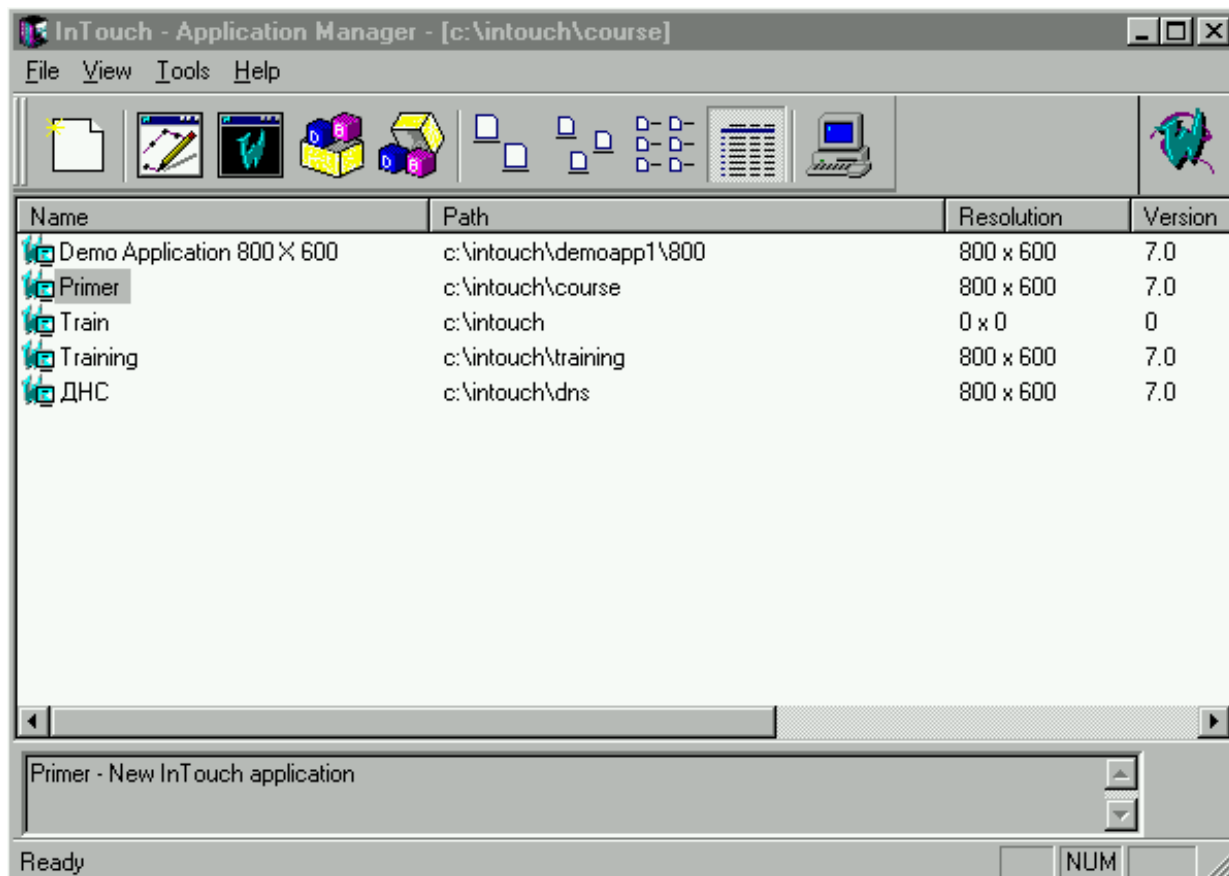
Тестирование практикума показало его работоспособность по всем описанным компонентам и позволило оценить предполагаемое время выполнения заданий - 9 часов.

В конце практикума приведено задание для самостоятельной работы студентов, которое может быть полезным при выполнении курсового проекта.

Перед началом практической работы в пакете **InTouch** желательно получить подробную информацию о пакете в учебном пособии «SCADA-пакет **InTouch** интегрированной системы **FactorySuite**», авторы Андреев Е.Б., Куцевич Н.А.

Раздел 1. Application Manager и WindowMaker.

Программа **Application Manager** предназначена для создания новых приложений, открытия созданных приложений в режимах **WindowMaker** или **WindowViewer**, уничтожения приложений, а также для запуска утилит «Словаря переменных» (**Tagname Dictionary**) **DBDump** и **DBLoad**.



Инструментарий **Application Manager** включает в себя 10 иконок и меню с опциями **File**, **View**, **Tools**, **Help**. Первые пять иконок дублируют команды меню **File** и предназначены (слева направо) для: создания нового приложения, открытия приложения в режиме разработки (**WindowMaker**), в режиме просмотра (**WindowViewer**), запуска утилит загрузки и выгрузки базы данных в виде текстового файла в/из «Словарь переменных». Следующие 4 конки соответствуют командам меню **View** и предназначены для представления приложений в поле **Application Manager**, соответственно: крупными и мелкими иконками по всему полю, мелкими иконками списком, мелкими иконками подробным списком. Последняя иконка предназначена для конфигурирования компьютера в сети.

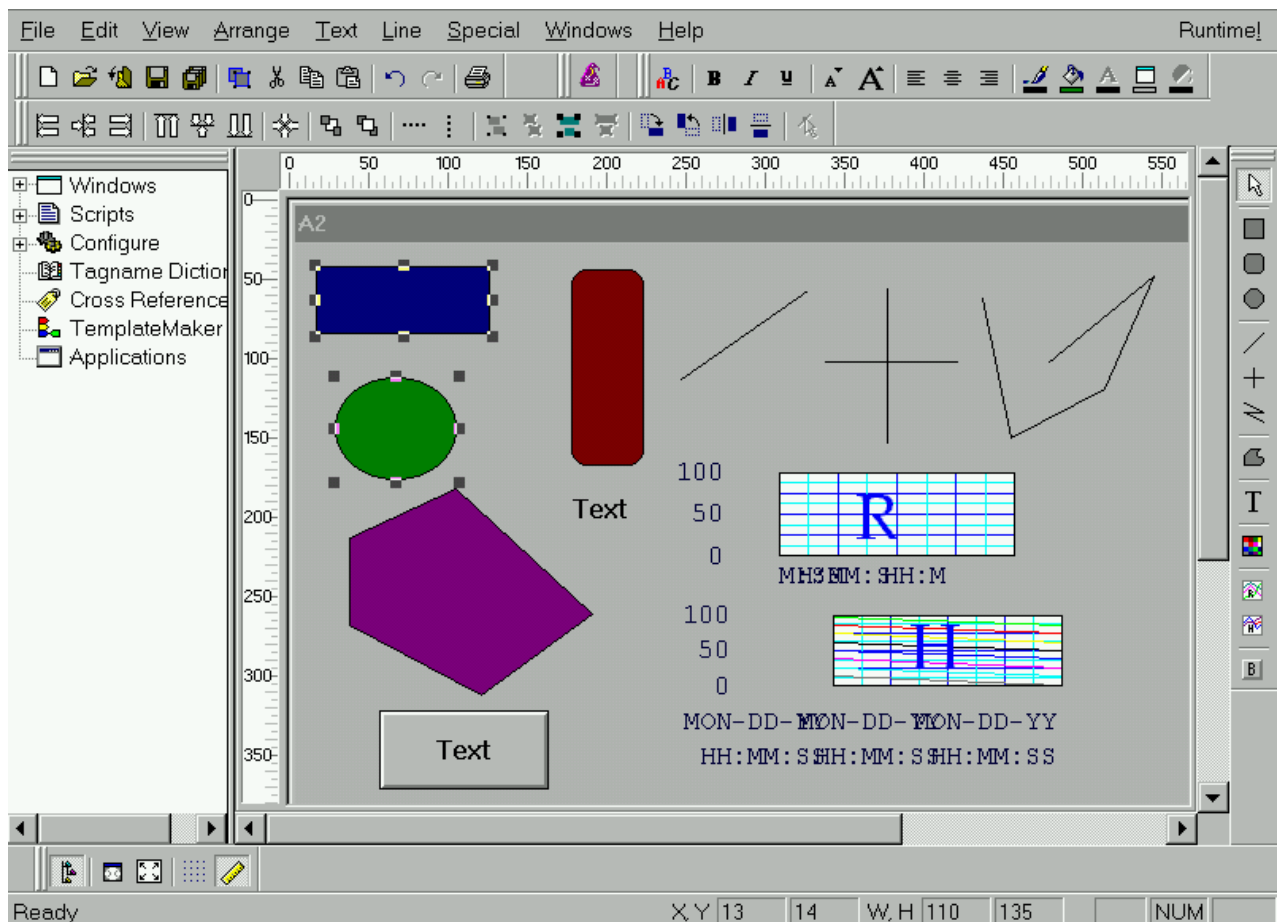
Находясь в **Application Manager**, можно изменить имя приложения (**File/Rename**) и удалить иконку с именем приложения (**File/Delete**).

Удаление каталогов и файлов самого приложения можно осуществить командой меню **Tools/Find Applications** (найти приложение), а затем воспользоваться командами правой кнопки мыши (**Delete**).

Для того чтобы открыть приложение, надо его выделить и щелкнуть по иконке **WindowMaker** (дважды щелкнуть на имени приложения).

Среда разработки **WindowMaker**.

Графический интерфейс **WindowMaker** соответствует стандарту интерфейсов **Windows 95,98/NT**. Он включает в себя меню, инструментальные панели, **Application Explorer**, строку статуса (состояний), линейку и рабочее поле.



Application Explorer. Это графическое представление приложения в иерархическом виде. В нем представлены все составляющие приложения. **Application Explorer** обеспечивает простой и быстрый доступ к ним, а также к наиболее широко используемым командам и функциям **WindowMaker**. В системе **Windows NT** на **Application Explorer** может быть возложена функция запуска других **Windows** - программ и приложений **FactorySuite**.

Компоненты окна **Application Explorer** могут быть свернуты и

развернуты. Если группа развернута (**Windows, Scripts, Configure**), то двойной щелчок на любом элементе группы вызывает его открытие.

Окна. Любое приложение **InTouch** состоит из множества окон с графическими и текстовыми объектами. **InTouch** предлагает три типа окон:

- **Replace** (заменяющее) - перекрывает все имеющиеся окна.
- **Overlay** (перекрывающее) - появляется поверх всех отображенных в текущий момент окон. Нажатие мыши на любом видимом участке лежащего под ним окна приводит к переходу на это окно. Если окно **Overlay** закрыть, то все скрываемые им окна восстанавливаются.

- **Popur** (всплывающее) - тип, аналогичный **Overlay**, но всегда остается поверх остальных окон. Для его закрытия требуются специальные команды.

Панели инструментов. Панели инструментов сгруппированы в соответствии с функциональным назначением инструментов. Всего в интерфейсе представлено 6 панелей. Все панели можно переместить в любое место окна: размещенная вдоль края панель - закрепленная, панель внутри окна - перемещаемая.

Для того чтобы спрятать или показать инструментальную панель, следует в меню **View** выполнить соответствующую команду (6 первых команд меню). Повторная команда вызывает обратное действие. Для перемещаемых панелей при закрытии можно воспользоваться меню правой кнопки или традиционным способом.

Чтобы спрятать или показать все панели сразу, надо щелкнуть на инструменте **Hide/Restore All** панели **View**.

- Панель **General** (общего назначения) содержит большинство команд меню **File** и **Edit**. Глубина вложенности команд **Undo/Redo** по умолчанию - 10. Но ее можно увеличить до 25 в диалоге **WindowMaker Properties** (команда **Special/Configure** либо двойной щелчок на **WindowMaker** в группе **Configure** окна **Application Explorer**).

- Панель **Wizard/ActiveX** по умолчанию содержит только один инструмент - доступ к диалогу **Wizard Selection** для вставки **wizard** - средств в окна. Имеется возможность добавлять инструменты в эту панель.

- Инструменты панели **Format** (форматов) дублируются командами меню **Text**. Кроме того, в эту панель включены средства управления цветом линий, текста, заполнения, фона окна, а также цветом прозрачных объектов.



- Панель **Draw Object** (рисование) включает 8 инструментов для создания простых объектов и четыре инструмента для создания сложных объектов (контейнера для вставки растровых изображений, объекта «тренд реального времени», объекта «исторический тренд», 3-мерных кнопок с текстом).



- Инструменты панели **View** (управление изображением) дублируют команды меню **View** и управляют изображением окна **WindowMaker**.



- Панель **Arrange** (выравнивание) содержит инструменты, эквивалентные командам меню **Arrange**.



Первые 7 инструментов - для выравнивания объектов, 2 - для управления уровнем объектов (передний и задний план), 2 - для равномерной расстановки объектов по горизонтали и вертикали, 2 - для работы с символами, 2 - для работы с компонентами, 4 - для вращения объектов и создания зеркальных отображений. Последний инструмент - для изменения формы многоугольника и ломаной линии.

Цветовая палитра InTouch насчитывает миллионы оттенков (16,7) и ограничивается только возможностями видеокарты компьютера.


Цветовая палитра появляется как при щелчке мыши по окрашенному прямоугольнику любого диалога, так и при щелчке по инструментам управления цветом линии, текста, заполнения, фона окна.

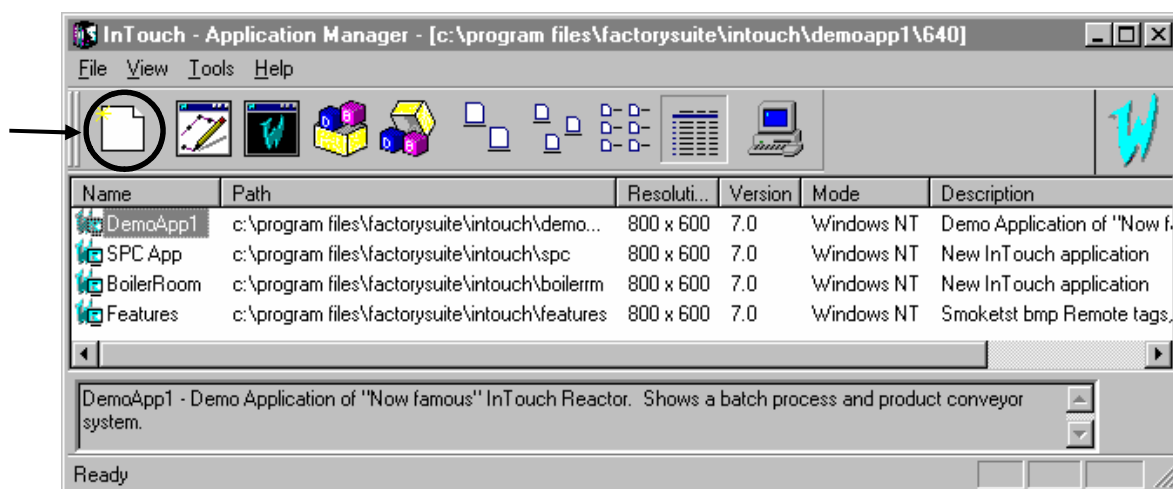
Линейка WindowMaker. Предназначена для более точного размещения объектов в окне. Малое деление соответствует 5 пикселям, среднее - 10, большое - 50 пикселям. Для перемещения выбранного объекта на 10 (50) пикселей надо нажать соответствующую клавишу со стрелкой, удерживая при этом клавишу SHIFT (CTRL).

Работа 1. Окна и графические объекты.

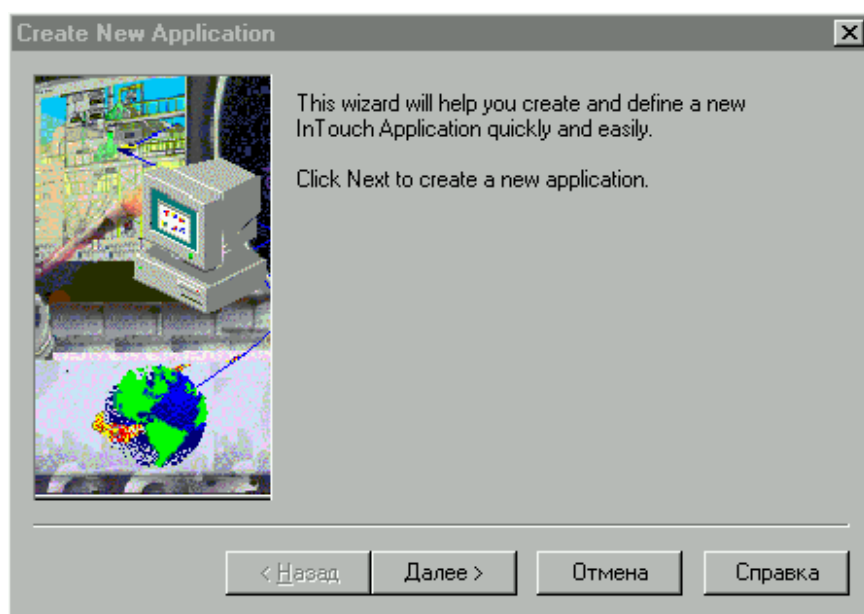
Цель работы:

- создание нового приложения;
- создание окон приложения;
- создание графических объектов в окнах;
- импортирование окон из другого приложения.

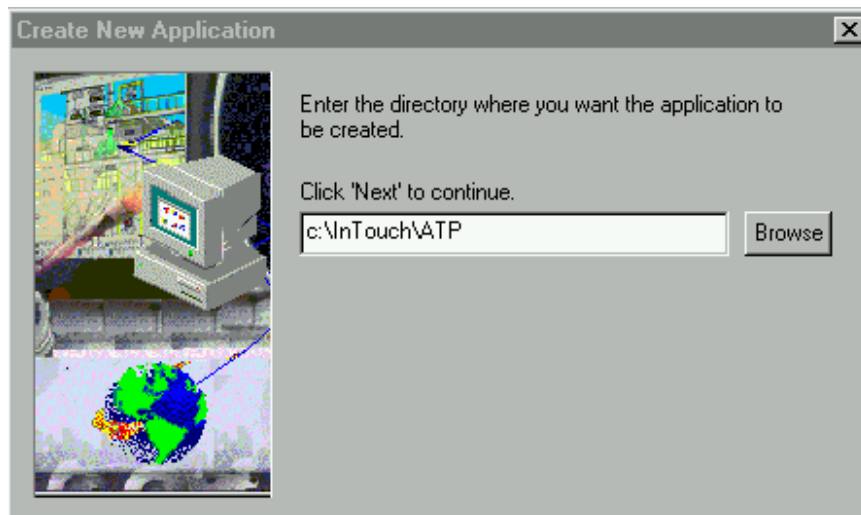
1. Запустите **InTouch Application Manager** .
2. Нажмите кнопку **New** на панели инструментов (или выполните команду **File/New**).



Появится следующее окно:

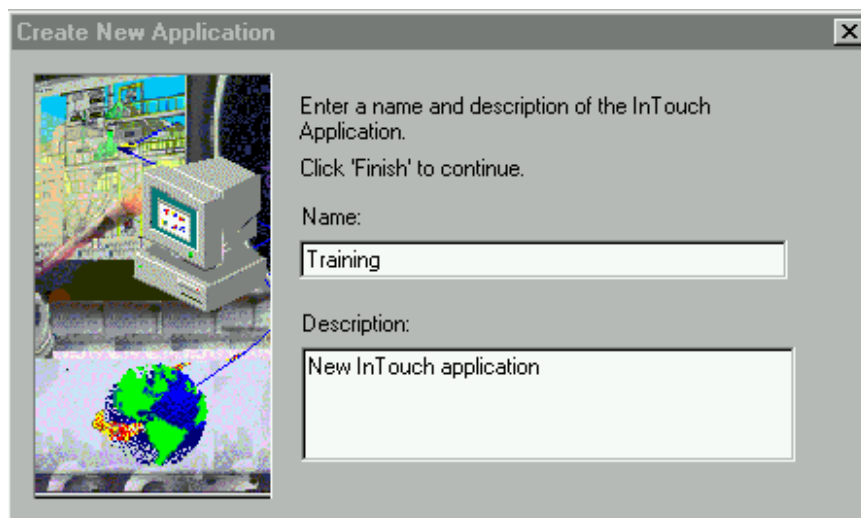


3. Нажмите **Next** (далее). На экран будет выведено следующее диалоговое окно:

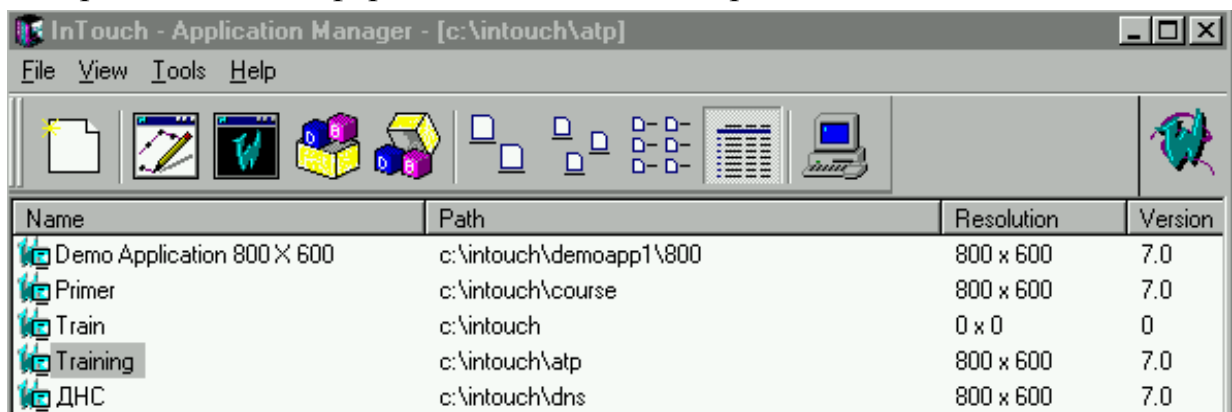


4. Укажите путь к каталогу, в котором будет храниться создаваемое приложение, например, **c:\InTouch\ATP**, и нажмите **Next**. После этого будет создан каталог.

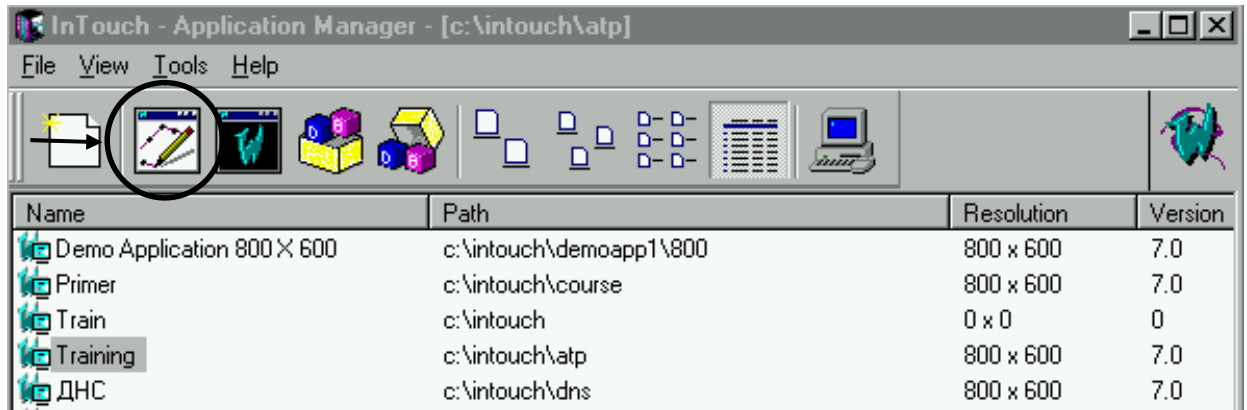
5. В следующем диалоговом окне в поле **Name** введите имя приложения, например, **Training** и в поле **Description** - описание (не обязательно).



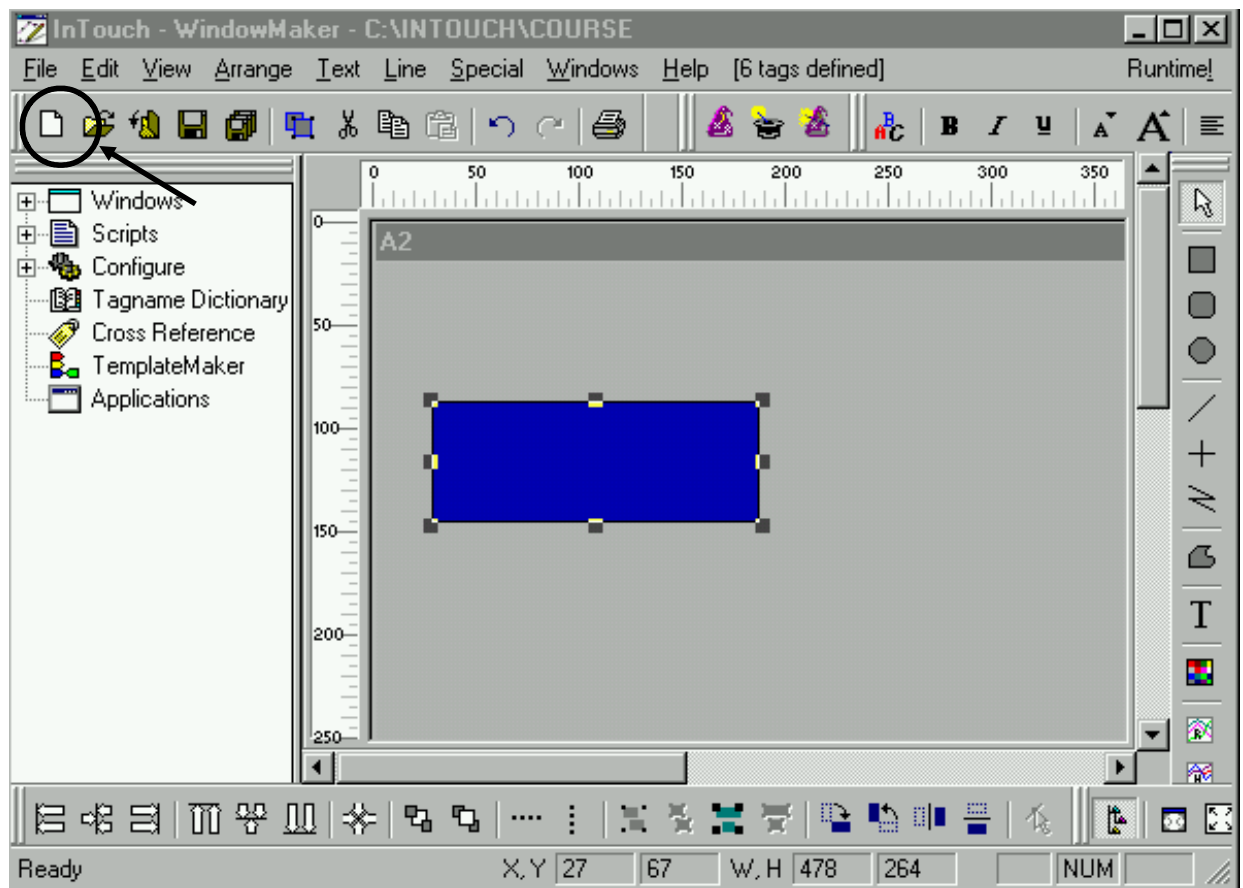
6. Нажмите кнопку **Finish** (готово). **Application Manager** немедленно отобразит в окне информацию о созданном приложении.



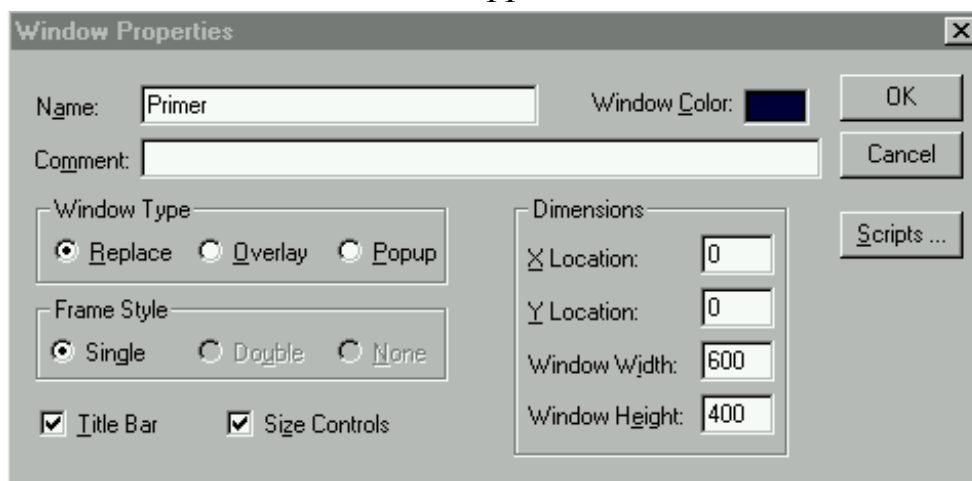
7. Дважды щелкните на строке с приложением **Training** или выделите его, а затем нажмите кнопку **WindowMaker**.



В результате будет запущена среда разработки **WindowMaker**.

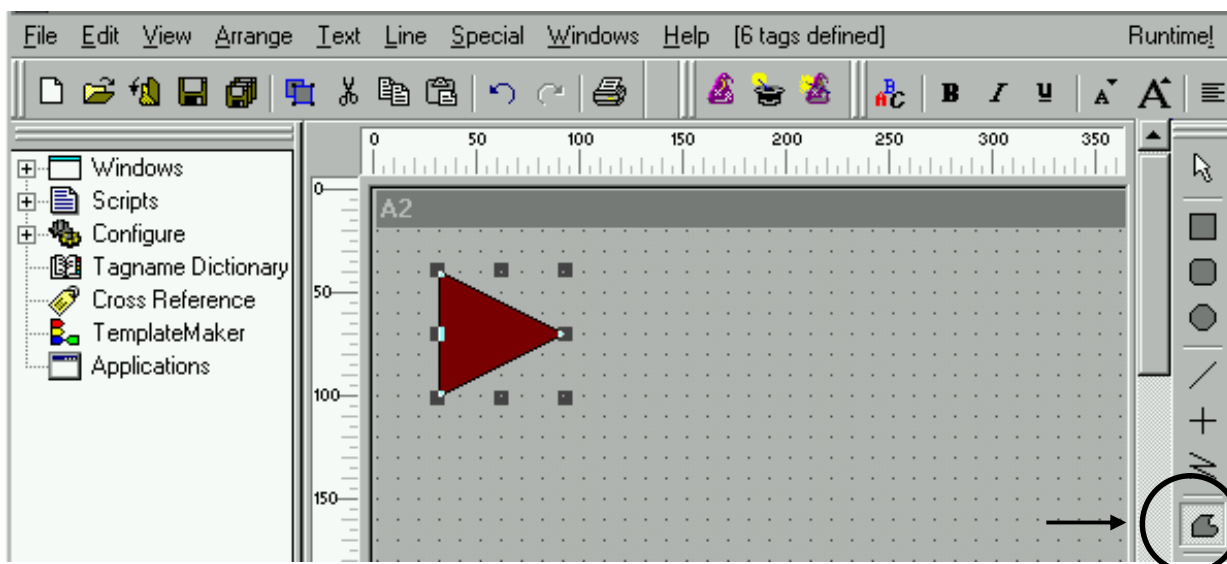


8. Нажмите кнопку **New** (см. рис. выше) на главной панели инструментов (или выполните команду **File/New Window**). Появится диалог **Window Properties**.

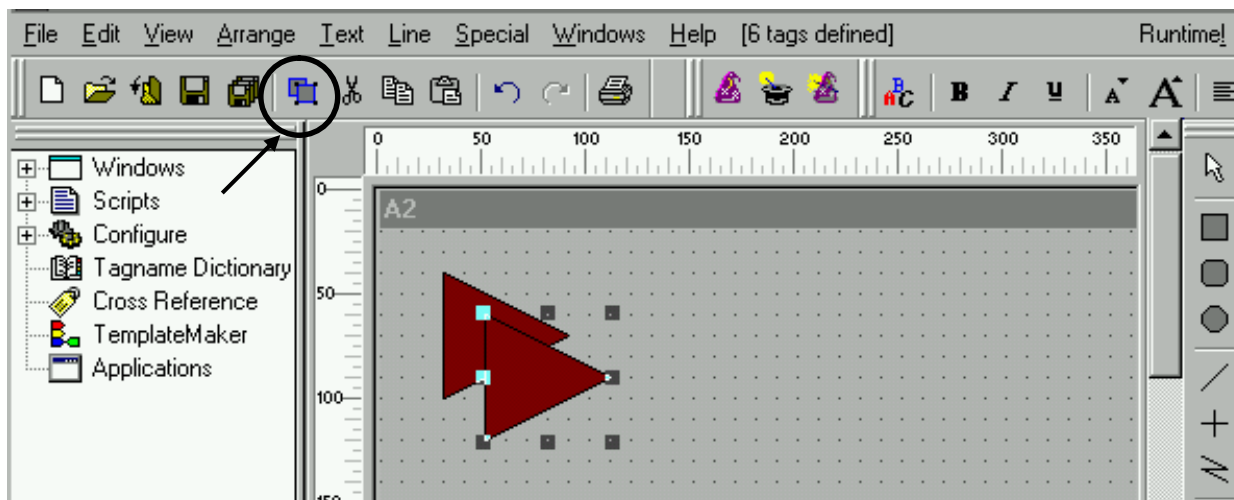


9. Введите имя окна (**Scratch** - черновик) в поле **Name** и нажмите **Ok**.

10. Нарисуйте клапан в этом окне. Для этого, используя средство **Polygon** (многоугольник - панель **Draw**, справа), нарисуйте треугольник.



11. Если треугольник не выделен, выделите его и нажмите кнопку **Duplicate** (дублировать - панель инструментов **General**, вверху).



12. Выделите второй треугольник и нажмите кнопку **Flip Horizontal** (отобразить зеркально по вертикали - панель **Arrange**, внизу справа).

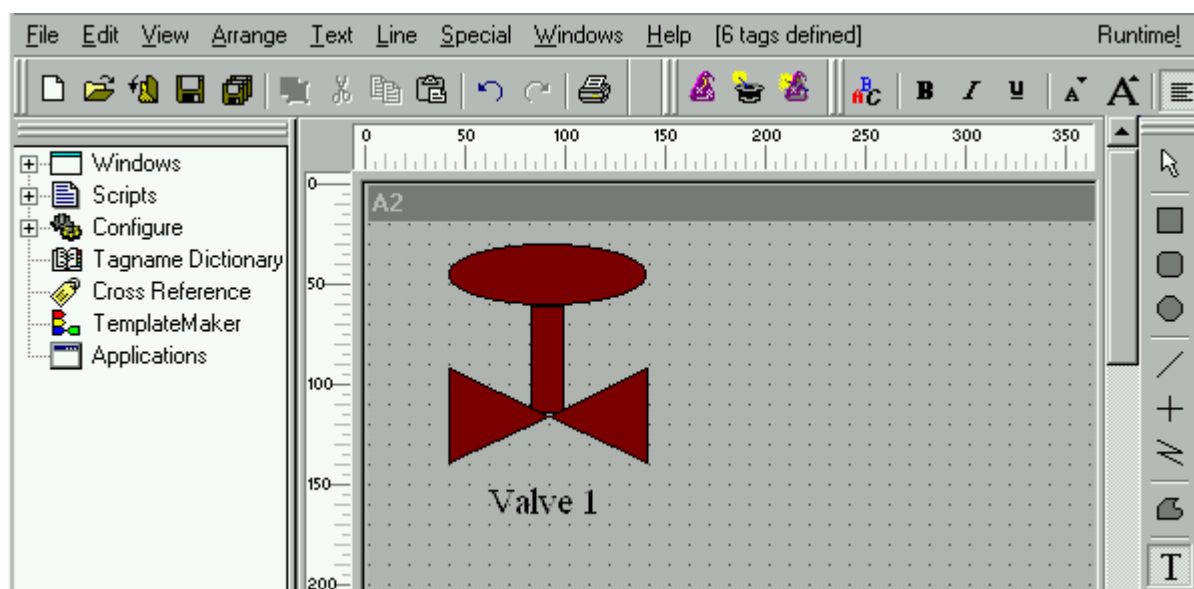
13. Соедините два треугольника, используя клавиши стрелок и мышь. Для перемещения объекта предварительно выделите его.

14. Добавьте шток клапана с помощью инструмента **Rectangle** (прямоугольник - панель **Draw**).

15. Расположите шток под клапаном. Для этого выделите шток и нажмите кнопку **Send to Back** (поместить под - панель **Arrange**).

16. Нарисуйте привод клапана с помощью инструмента **Ellipse**.

17. Используя инструмент **T (Text)**, добавьте под клапаном текст **Клапан1 (Valve1)**. В результате рабочее поле **WindowMaker** будет выглядеть следующим образом:



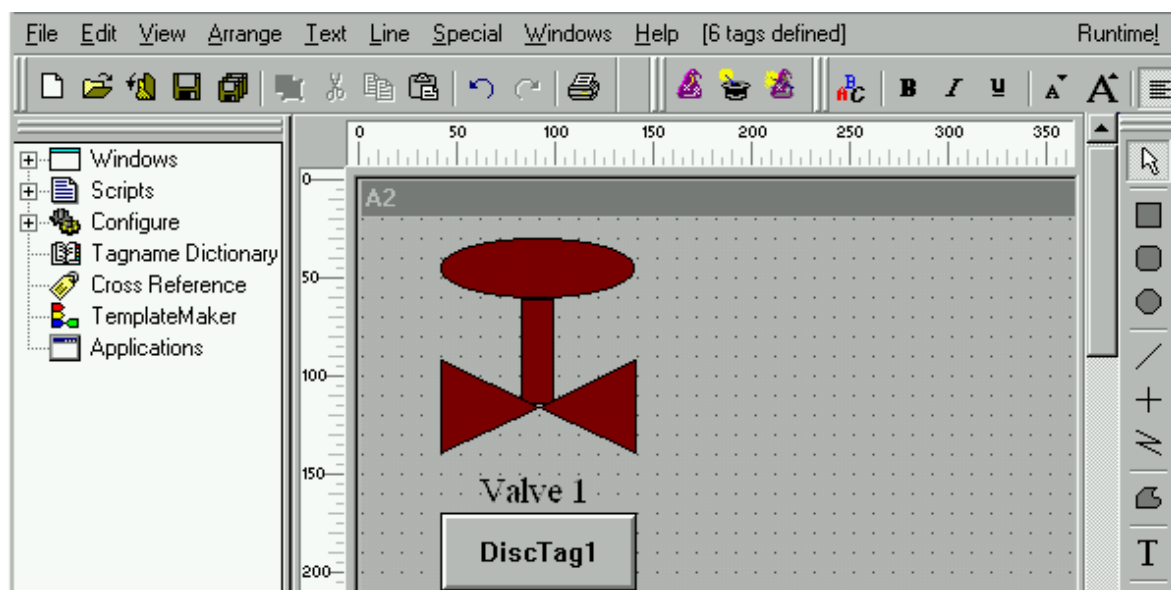
Созданный клапан состоит из пяти отдельных элементов (включая текст). Чтобы сделать из них единый объект, нажмите клавишу **F2** для выделения всех объектов в окне, а затем нажмите кнопку **Make Symbol** (объединить в символ - панель **Arrange**). Теперь все составляющие клапан элементы объединены в *символ* и могут обрабатываться как *единый объект*. Можно изменять размер клапана, выделив его и потянув за один из захватов. Можно переместить клапан в новое место. У созданного объекта можно изменить цвет заполнения, линий, цвет рамки, а также цвет и тип шрифта надписи.

18. Добавьте кнопку под клапаном с помощью инструмента **Button** (кнопка - панель **Draw**).

19. Для замены текста кнопки выделите ее и выполните команду **Special /Substitute Strings** (или **CTRL+L**). В поле **Text** диалога **Substitute Strings** введите текст - **DiscTag1**.

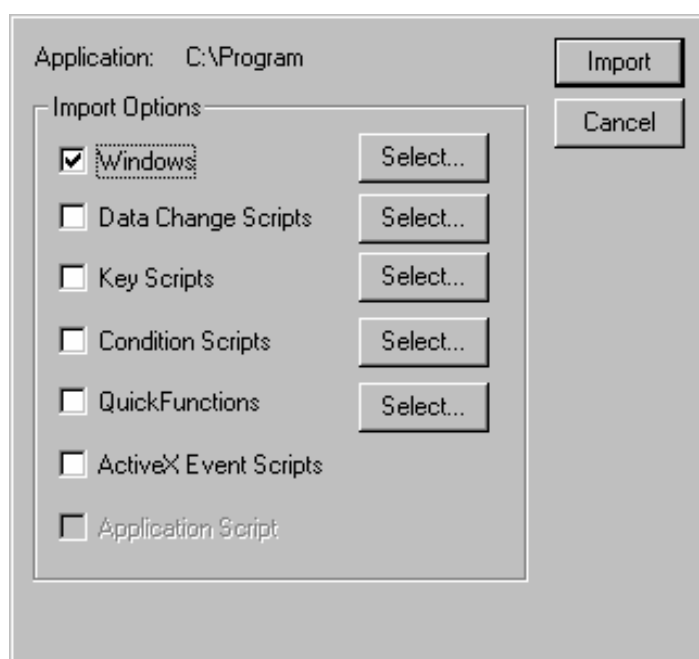


20. Нажмите **Ok**. Созданный рисунок будет иметь следующий вид:

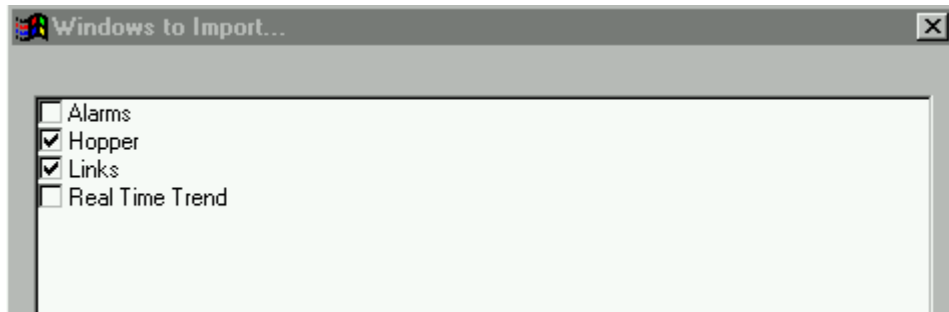


21. Соответствующими командами меню **File** сохраните окно и закройте все окна **WindowMaker**.

22. **Импорт окон**. Выполните команду меню **File/Import**. Укажите путь к подкаталогу **WWimp70** (расположен в каталоге **FactorySuite** на диске SCADA) и щелкните **Ok**. Появится следующее диалоговое окно:

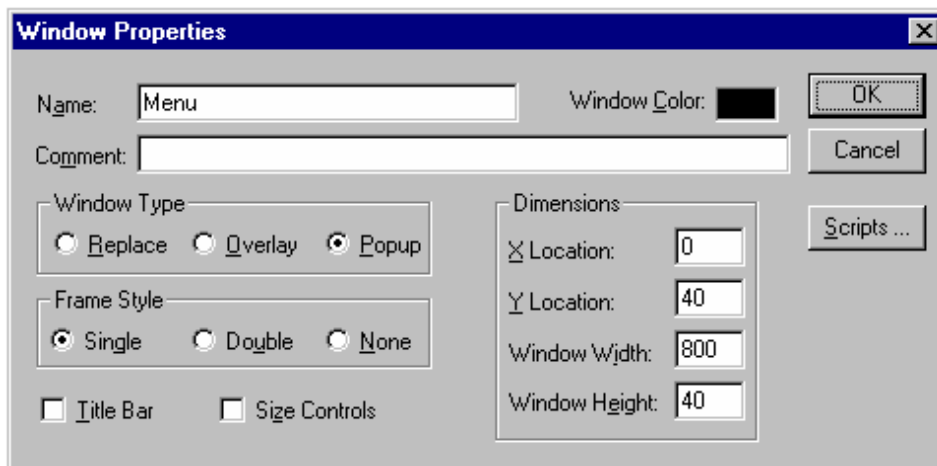


23. Выбрав **Windows** (окна) для импорта, щелкните по кнопке **Select**. В диалоге **Windows to Import** выберите окна **Hopper** и **Links** и нажмите **Ok**.



24. Вновь появится диалог из пункта 22. Нажмите кнопку **Import** для импорта окон. Эти окна автоматически не откроются, но они уже импортированы.

25. Создайте новое окно **Menu** со следующими свойствами:



26. В этом окне создайте шесть кнопок. Разместите их в окне **Menu** в соответствии с рисунком.



27. Добавьте текст между кнопками **Disk Space Free** и **#MB** (для изменения цвета текста воспользуйтесь инструментом **A** панели **Format**, вверху справа).



28. Нажмите иконку **Wizards...** (вверху в середине, справа от панели **General**) и в появившемся диалоге **Wizard Selection** выберите группу мастеров **Clocks**. Выделите мастер-объект **Digital Time/Date** и нажмите **Ok** (или дважды щелкните по мастер-объекту). Расположите его слева от кнопки **Scratch**, создайте его копию и разместите ее справа от кнопки **HTrends**.

29. Двойной щелчок на мастер-объекте вызывает на экран диалог **Digital Clock Wizard**.



Сконфигурируйте мастер-объекты таким образом, чтобы в левой части окна **Menu** выводить время (опция **Show Time**), а в правой - дату (опция **Show Date**).



30. Сохраните и закройте окно **Menu**.

Раздел 2. Анимационные связи.

После создания графических объектов их можно «оживить», если прикрепить к ним **анимационные связи**. Анимационные связи вызывают изменение внешнего вида объекта в зависимости от изменения значений переменной или выражения. Например, символ насоса может быть красным, когда насос выключен, и зеленым, когда он включен. Символ насоса можно также превратить в кнопку, чувствительную к нажатию.

Для присваивания объекту какого-либо динамического свойства предварительно он должен быть создан и выбран. А затем надо ввести команду **Special/Animation Links** (или дважды щелкнуть на объекте). На экране появится диалоговое окно выбора анимационных связей (см. на следующей странице). Всего в диалоге представлено 34 кнопки, которые размещены в четырех столбцах на 11 полях.

- С помощью динамических связей, представленных в левом столбце диалога (**Touch Links** - связи по нажатию), осуществляется ввод информации в систему.

- Все остальные динамические связи предназначены для вывода информации на дисплей (**Display Links** - дисплейные связи).

Все переменные, имена которых будут использованы при конфигурировании динамических связей, должны быть определены в словаре переменных. Если какая - то переменная не определена в словаре, то на экране появится диалог с предложением ее определить. При подтверждении будет открыт словарь переменных.

The image shows a software configuration window titled 'Touch Links'. It is organized into several sections, each with a title and a list of options, each preceded by a small square checkbox:

- Touch Links**
 - User Inputs**
 - ☐ Discrete
 - ☐ Analog
 - ☐ String
 - Sliders**
 - ☐ Vertical
 - ☐ Horizontal
 - Touch Pushbuttons**
 - ☐ Discrete Value
 - ☐ Action
 - ☐ Show Window
 - ☐ Hide Window
- Line Color**
 - ☐ Discrete
 - ☐ Analog
 - ☐ Discrete Alarm
 - ☐ Analog Alarm
- Fill Color**
 - ☐ Discrete
 - ☐ Analog
 - ☐ Discrete Alarm
 - ☐ Analog Alarm
- Text Color**
 - ☐ Discrete
 - ☐ Analog
 - ☐ Discrete Alarm
 - ☐ Analog Alarm
- Object Size**
 - ☐ Height
 - ☐ Width
- Location**
 - ☐ Vertical
 - ☐ Horizontal
- Percent Fill**
 - ☐ Vertical
 - ☐ Horizontal
- Miscellaneous**
 - ☐ Visibility
 - ☐ Blink
 - ☐ Orientation
 - ☐ Disable
- Value Display**
 - ☐ Discrete
 - ☐ Analog
 - ☐ String

• Свойства **Touch Links** предоставляют оператору возможность открыть/закрыть клапан, запустить на выполнение скрипт, ввести новое значение переменной (задание), запустить распечатку отчета, перейти в другое окно и т. д.

Типы связей **Touch Links**:

- **User Inputs** (ввод значений) - логический, аналоговый, символьный;
- **Sliders** (движковые регуляторы) - вертикальный, горизонтальный;
- **Touch Pushbuttons** (кнопки) - для логического ввода, исполнительные, для открытия/закрытия окон.

• Вывод информации оператору осуществляется с помощью дисплейных связей **Display Links** 8 типов (подчеркнуты):

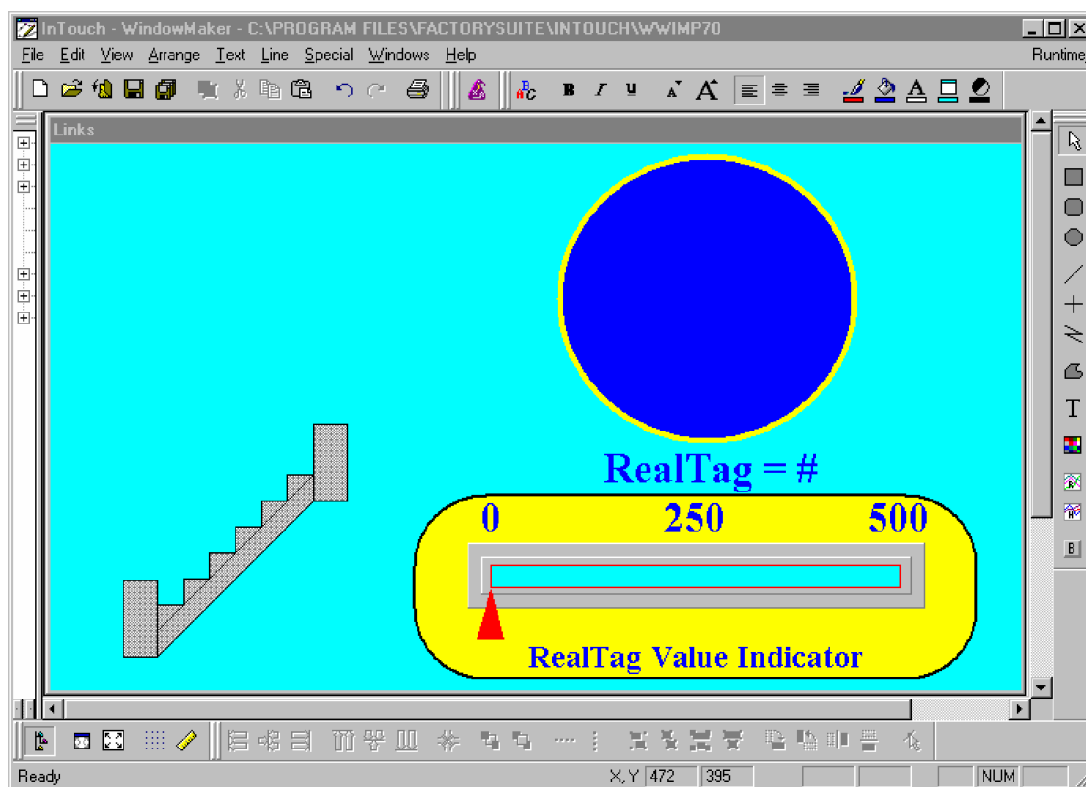
- цвет линии, контура и текста - логический, аналоговый, аварийное состояние логического и аналогового типа;
- размер объекта - высота, ширина;
- перемещение - горизонтальное, вертикальное;
- степень заполнения объекта (вертикальное, горизонтальное);
- смешанный - видимость, вращение, мерцание, блокировка;
- вывод на экран - логический, аналоговый, символьный.

Работа 2. Создание анимационных связей.

Цель работы:

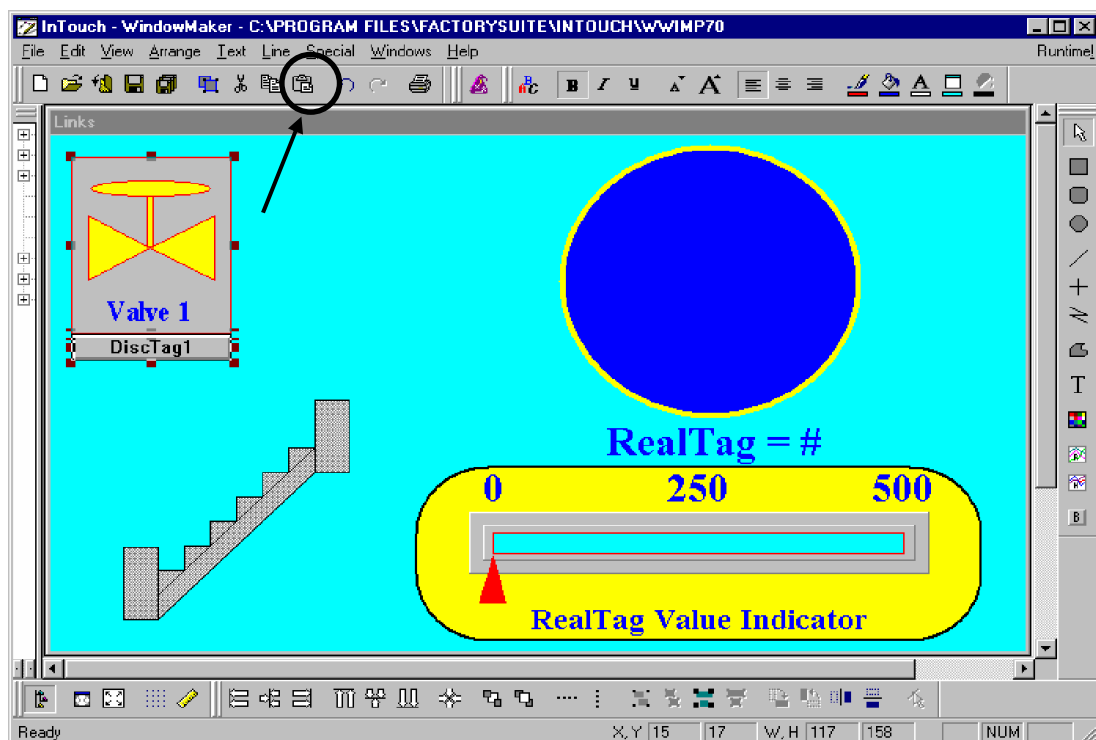
- копирование и вставка объектов;
- присваивание объектам динамических свойств.

1. Откройте окно **Links**.

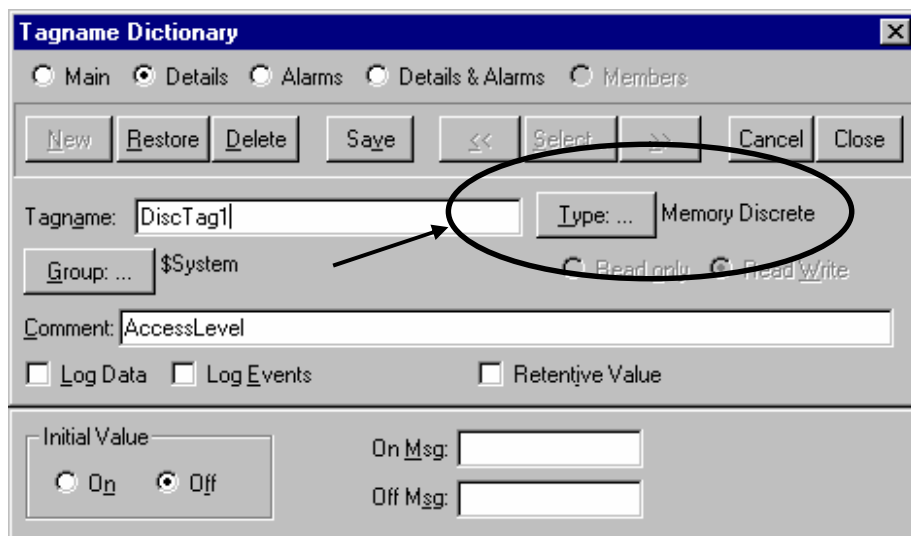


2. Скопируйте клапан и кнопку из ранее созданного окна. Для этого откройте его, выделите объекты и нажмите кнопку **Copy** на главной панели инструментов.

3. Перейдите в окно **Links** с помощью команд меню **Windows** (выбор окон) и нажмите кнопку **Paste** (вставить).



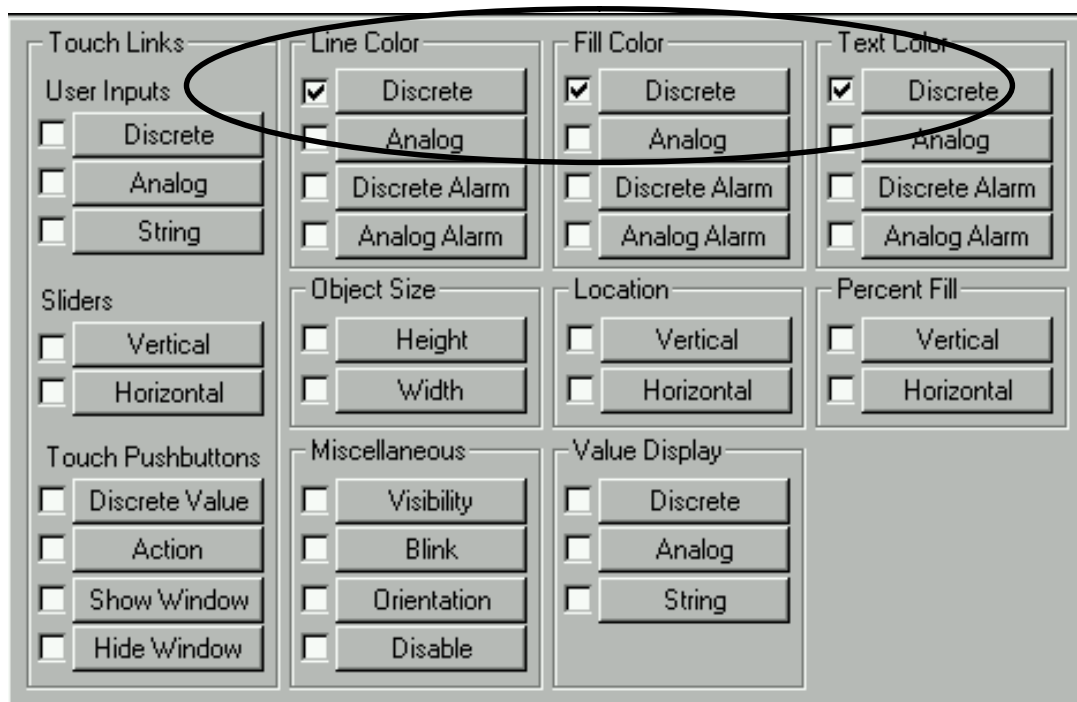
4. Командой **Special/Tagname Dictionary** откройте словарь переменных и создайте две переменные **DiscTag1** и **DiscTag2** типа **Memory Discrete**.



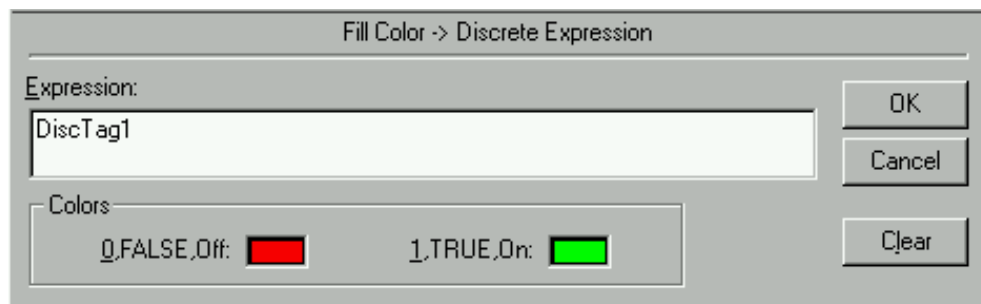
Для этого в словаре переменных (**Tagname Dictionary**) нажмите кнопку **New** и введите имя первой переменной в поле **Tagname**. Затем откройте диалог выбора типа переменной **Tag Types** нажатием кнопки **Type** (см. рис.). Выберите тип переменной **Memory Discrete** и нажмите **Ok**. В словаре переменных сохраните только что определенную переменную (**Save**).

Проделайте эту же последовательность операций для определения второй переменной.

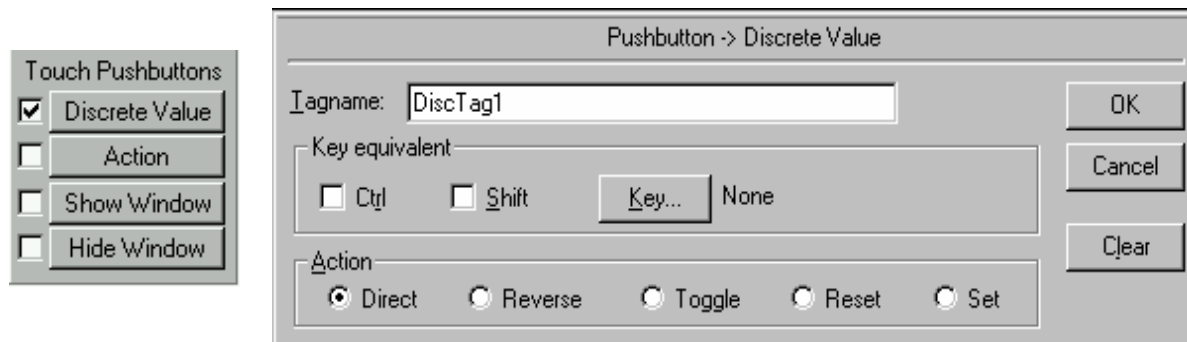
5. Дважды щелкните на клапане для вызова меню анимационных связей. Поочередно установите динамические связи **Line Color**, **Fill Color** и **Text Color** (все - типа **Discrete**, см. рис.) для переменной с именем **DiscTag1**.



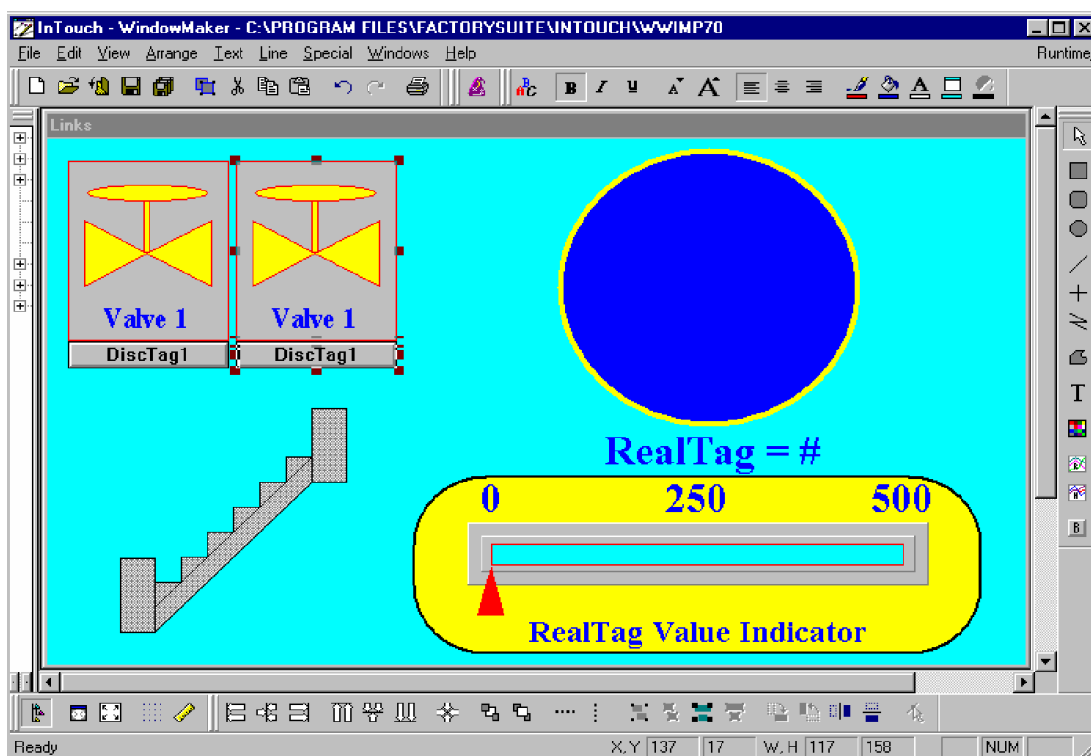
6. В выводимых диалогах для **0, FALSE, Off** выберите красный цвет, а для **1, TRUE, On** - светло-зеленый.



7. Для анимирования кнопки **DiscTag1** дважды щелкните на ней, затем определите её как **Touch Pushbutton** со свойством **Discrete Value** и с действием **Direct**. В поле **Tagname** введите имя переменной **DiscTag1**.



8. После динамизации клапана и кнопки создайте их копию: выделите клапан с кнопкой и щелкните по кнопке **Duplicate** общей панели инструментов. Расположите клапаны рядом, как показано на рисунке.

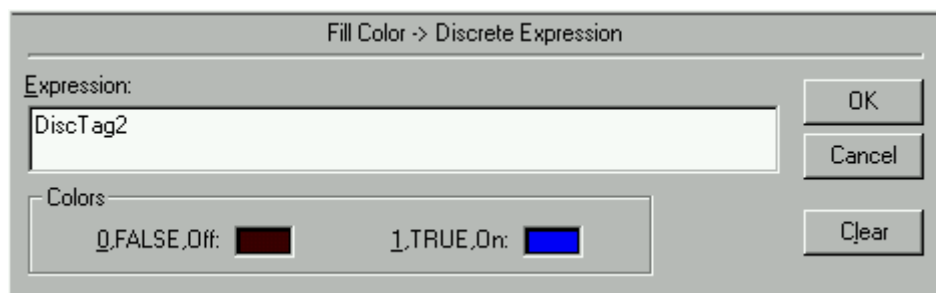


9. Выделите второй клапан и выполните команду **Special/Substitute Strings** (или CTRL+L). Замените текст под вторым клапаном на **Valve 2** (предварительно «разбейте» символ на составные части). Затем такой же командой замените текст кнопки под вторым клапаном на **DiscTag2**.

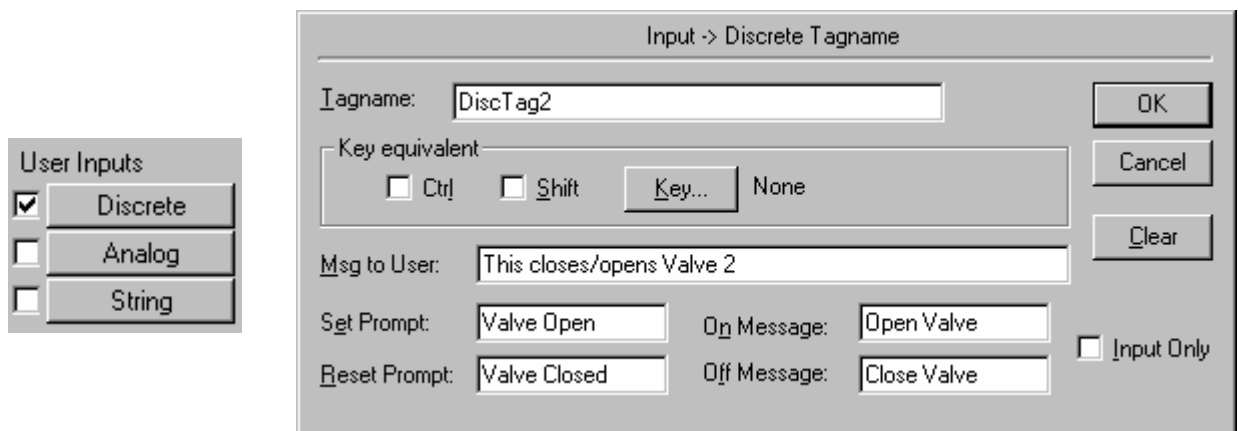


10. Дважды щелкните на втором клапане для вызова меню анимационных связей. Поочередно установите динамические связи **Line Color**, **Fill Color** и **Text Color** (все - типа **Discrete**) для переменной с именем **DiscTag2** (см. п.5).

11. В этих диалогах для **0, FALSE, Off** выберите черный цвет, а для **1, TRUE, On** - синий.

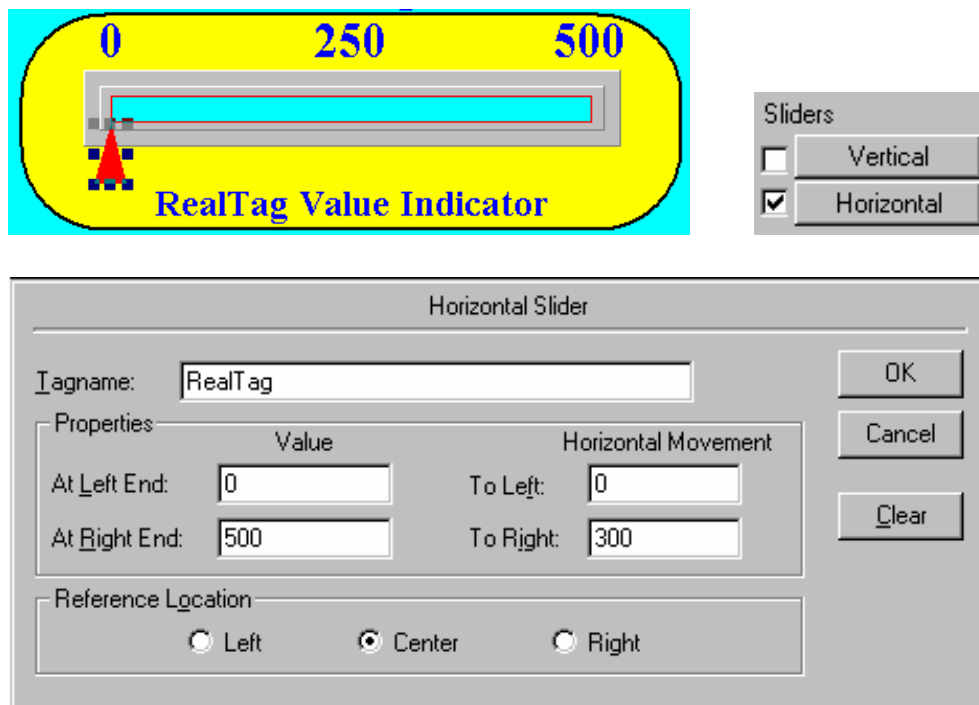


12. Определите кнопку **DiscTag2** как **User Input** со свойством **Discrete** и «привяжите» к переменной **DiscTag2**. Задайте текст приглашения и сообщений о состоянии клапана.



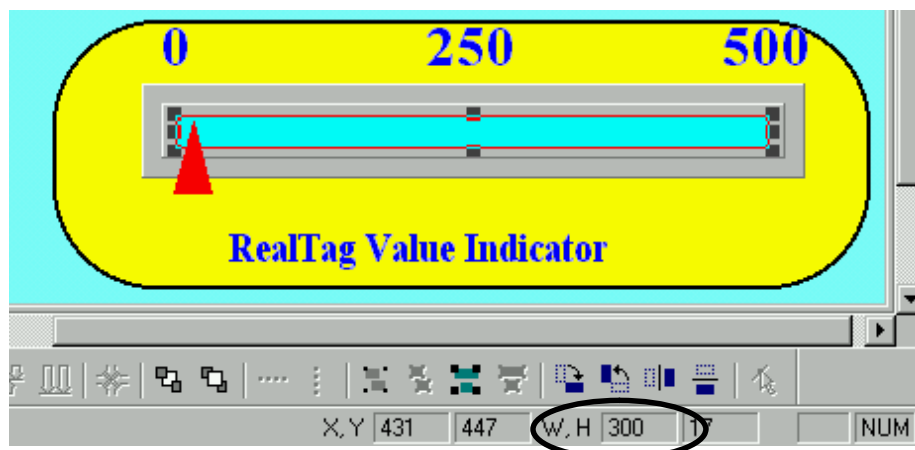
13. Перейдите в среду исполнения **WindowViewer** (щелчком по тексту **Runtime** в правом верхнем углу окна **WindowMaker**). Клапаны должны изменять цвет в зависимости от состояния.

14. Вернитесь в среду **WindowMaker** (щелчком по тексту **Development** в правом верхнем углу окна **WindowViewer** или командой меню **File**). Установите динамические свойства для указателя движкового регулятора **RealTag Value Indicator**. Дважды щелкните на указателе. Определите его как **Slider/Horizontal** и «привяжите» к нему переменную **RealTag** типа **Memory Real**.



15. В поле **At Left End Value** (значения переменной, соответствующее крайнему левому положению движка) введите **0**, в поле **At Right End Value** (значения переменной, соответствующее крайнему правому положению движка) - **500**, в поле **Horizontal Movement To Left** - **0** и в поле **Horizontal Movement To Right** (перемещение движка вправо в пикселях) – **300 (L)**.

Для определения величины **L** выделите шкалу индикатора. Ширина шкалы в пикселях (**L**) будет отображена в строке состояния (300, см. рис.).



16. Если переменная **RealTag** не определена, то система предложит определить ее. Создайте переменную **RealTag** типа **Memory Real** с **Initial Value - 250**, **Min Value - 0** и **Max Value - 500**.

17. Если переменная **IntTag** не определена, определите ее как переменную типа **Memory Integer** с **Initial Value = 250**, **Min Value = 0** и **Max Value=500** (переменная понадобится позже).

18. Присвойте текстовому объекту **RealTag #** динамическое свойство **User Input/Analog** и «привяжите» его к переменной **RealTag**. Задайте **Min Value=0** и **Max Value=480**.

19. Дважды щелкните на прямоугольнике шкалы **Value Indicator** и добавьте свойство **Percent Fill** (степень заполнения).



20. Определите свойство **Percent Fill/Horizontal**, «привязав» его к переменной **RealTag**. Укажите 500 для **Value at Max Fill** (значение переменной, соответствующее максимальному заполнению) и выберите цвет фона. Направление заполнения (**Direction**) - вправо (**Right**).

21. Перейдите в среду исполнения. При перемещении указателя длина закрашенной области индикатора должна соответствовать отображаемому значению переменной. Введите новое значение переменной, щелкнув на поле отображения значения переменной. Индикатор значения должен переместиться.

Задание для самостоятельного выполнения.

1. Задайте свойство **Object Size/Height** (высота) большому кругу в этом окне так, чтобы его высота была **100%**, когда **RealTag = 500**, и **0%**, когда **RealTag=0**. Увеличение от центра.

Задайте свойство **Object Size/Width** (ширина) этому же объекту, чтобы его ширина была **100%**, когда **RealTag = 500**, и **0%**, когда **RealTag = 0**. Увеличение из центра.

Задайте кругу свойство **Fill Color /Analog**, указав **100, 200, 300** и **400** в качестве **BreakPoints**. Выберите 5 цветов. «Привяжите» это свойство к переменной **RealTag**. Перейдите в среду исполнения и проверьте работу приложения.

2. Выделите объект **Транспортер** и «разбейте» его на составляющие. Обратите внимание, что он состоит из нескольких частей: верхней ступеньки, набора ступенек, направляющих и др.

Набору ступенек задайте свойство **Location/Vertical**. В поле **Expression** введите **\$Second MOD 10** (с пробелами). В поле **At Top** введите **10**, в поле **Up - 20**. Значения полей **At Bottom** и **Down** задайте равными **0**.

Этому же набору ступенек задайте свойство **Location/Horizontal**. В поле **Expression** также введите **\$Second MOD 10**. Оставьте значения полей **At Left End** и **To Left** равными **0**. В поле **At Right End** введите значение **10**, а в поле **To Right** - **20**.

Перейдите в среду исполнения. Ступени транспортера должны перемещаться.

Раздел 3. Скрипты.

С помощью скриптов можно значительно расширить возможности разрабатываемых приложений. Они позволяют исполнять определенные последовательности команд и логических операций после выполнения некоторого условия (раскрытия окна, нажатия на клавишу, изменения значения переменной или выражения и т. д.).

Типы скриптов.

- Скрипты уровня приложения (**Application Scripts**) относятся ко всему приложению и используются для запуска других приложений, имитации технологических процессов, вычисления значений переменных и т.д. Могут исполняться один раз в момент запуска или завершения приложения или периодически через заданные интервалы времени.

- Скрипты уровня окна (**Window Scripts**) связываются с конкретным окном. Могут исполняться один раз в момент открытия или закрытия окна или периодически через заданные интервалы времени пока окно отображается на экране.

- Клавишные скрипты (**Key Scripts**) привязываются к какой-либо клавише или комбинации клавиш клавиатуры. Это может быть полезным при создании каких-либо глобальных для всего приложения функций (возврат в главное окно, окончание сеанса работы с приложением и т. д.). Могут исполняться один раз в момент нажатия на клавишу или ее отпускания, а также периодически пока нажата клавиша.

Для определения комбинации клавиш можно отметить опции **SHIFT** и **CTRL**, и нажать кнопку **Key**. Откроется окно со списком допустимых клавиш. Для выбора клавиши в диалоге следует щелкнуть на ее имени.

- Скрипты, запускаемые кнопками (**Touch Pushbutton Action Scripts**) очень похожи на клавишные скрипты и связываются с объектами, которые будут использоваться в качестве исполнительных кнопок. Эти скрипты запускаются при каждом нажатии на объект-кнопку.

- Скрипты по изменению логического выражения (**Condition Scripts**) связываются с логической переменной или выражением, которое будет принимать значения либо «истина», либо «ложь». Логические скрипты могут содержать в себе и аналоговые переменные. Могут исполняться один раз в момент перехода переменной в состояние «истина» («ложь») или непрерывно пока значение переменной «истина» («ложь»).

- Скрипты по изменению данных (**Data Change Scripts**) связываются либо с переменной, либо с полем переменной. Эти скрипты исполняются только один раз, когда значение переменной либо поля меняется на величину, превышающую значение допуска, заданного в словаре переменных.

- Скрипты событий **ActiveX** (**ActiveX Event**) предназначены для поддержки механизма реакции на события в **ActiveX** - объектах. С каждым событием может быть связан один скрипт типа **ActiveX Event**, запускающийся в **WindowViewer** во время исполнения приложения.

- **Quick Function** - скрипты, которые могут вызываться из других скриптов и использоваться в выражениях при определении динамических свойств объектов.

Редактор скриптов.

Диалоги редактора, открываемые при создании скриптов различных типов, имеют небольшие отличия. Вызов диалога редактора скриптов в окне **WindowMaker** осуществляется командой **Special/Scripts** с последующим выбором типа создаваемого или редактируемого скрипта. Для этого можно также воспользоваться окном **Application Explorer**, выбрав папку **Scripts** (открыть редактор уровня окна).

В нижней части редактора (см. рис. на следующей странице) – кнопки ввода ключевых слов и математических операторов. После нажатия кнопки они будут вставлены в скрипт.

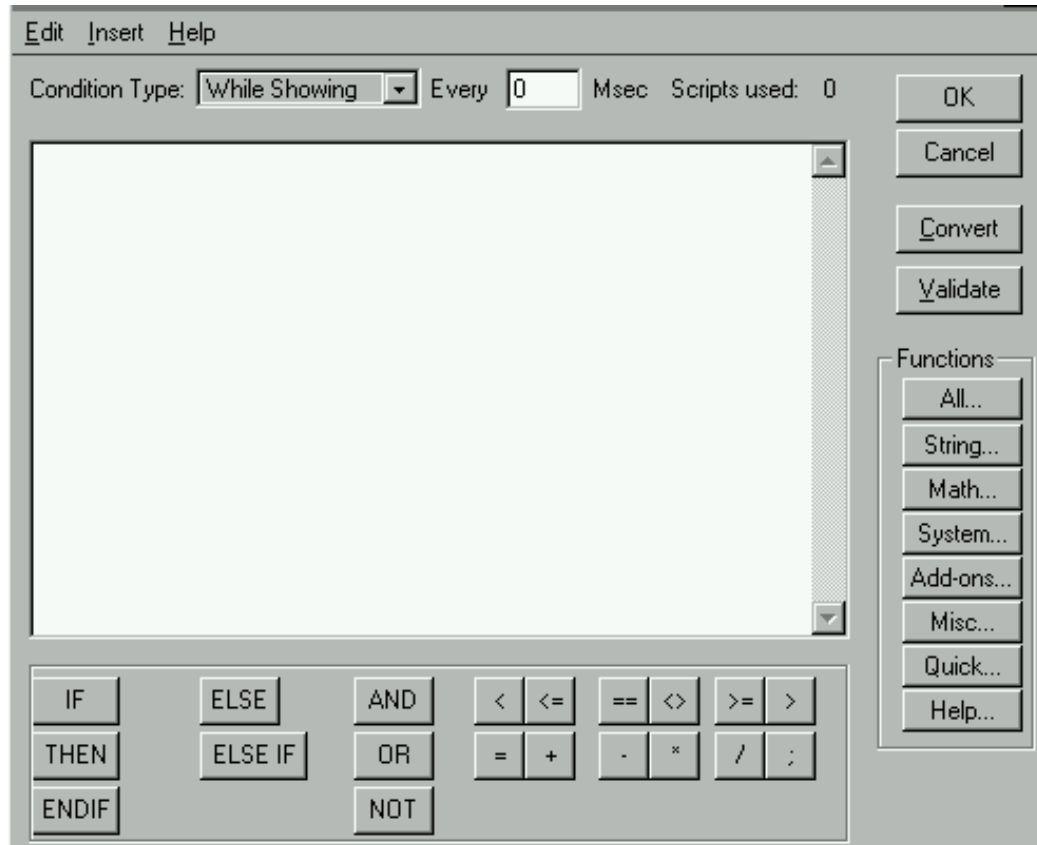
Справа, в поле **Functions**, размещены кнопки выбора встроенных функций. Сверху расположено меню и несколько иконок, дублирующих команды меню.

⇒ Создание нового скрипта – команда **Script/New**.

⇒ Выделить весь скрипт, вырезать, копировать, вставить, уничтожить весь скрипт – команды меню **Edit**.

Для вставки в скрипт **имени переменной** или имени окна – команды меню **Insert (Tagname, Window)** или соответствующие иконки. В первом

случае появится **браузер** переменных. Двойной щелчок на выбранной переменной вставляет ее в скрипт. Двойной щелчок по имени переменной выводит на экран диалог определения переменной. Во втором случае появится список всех окон приложения.



⇒ Для вставки в скрипт **поля переменной** следует в **браузере** переменных выбрать переменную и нажать кнопку со стрелкой **Dot Field**. Появится список полей этой переменной. После выбора поля щелкнуть **Ok**. Поле переменной будет вставлено в скрипт. Быстро вставить поле переменной можно, напечатав имя переменной с точкой, и дважды щелкнуть справа. Появится диалог **Choose field name**. Выбрать поле, которое сразу же будет вставлено в скрипт.

⇒ Для поиска или замены переменной ввести команду **Edit/Find**. На экране появится диалог **Replace**. Ввести исходное имя переменной (верхняя строка) и нажать **Find Next** (следующее). Имя этой переменной будет высвечено в поле скрипта. Нажать кнопку **Replace** (заменить) и нажать **Find Next** для поиска следующего «старого» имени переменной. Для замены переменной по всему тексту скрипта нажать **Replace All**.

⇒ Для проверки синтаксиса скрипта нажать **Validate** (проверка). Проверка выполняется автоматически при нажатии **Ok** или **Save**. При обнаружении ошибки на экран будет выведено сообщение.

⇒ При создании нескольких скриптов для сохранения одного нажать **Save**, а затем создать новый скрипт кнопкой **New**. Сохранение скриптов уровня приложения и окна производится нажатием **Ok**.

⇒ Для выхода из редактора скрипта ввести команду **Scripts/Exit**.

Встроенные функции.

В пакете **InTouch** имеется набор встроенных функций, которые могут быть связаны с командами или использованы в скриптах для выполнения самых различных задач.

Все встроенные функции разбиты на четыре группы:

- **String...** - для обработки различных символьных строк и переменных;
- **Math...** - математические функции (работают с целыми и вещественными аргументами);
- **System...** - системные функции делятся на две категории: файловые (**File**) и для работы с **Windows** - приложениями (**Info**). Применяются для выполнения таких действий, как запуск другого **Windows** – приложения, получение информации о приложении, копирование, уничтожение или перемещение файлов и т. д.;
- **Misc...** - функции для работы с алармами распределенных систем (**Ack**), архивными трендами (**HT**), печатью и др.

Вызов списка функций группы осуществляется нажатием соответствующей клавиши. Назначение функций и описание их аргументов приведено в документации.

Вставка встроенных функций в скрипт производится щелчком по выбранной функции в списке функций. Она вместе со своими аргументами будет автоматически вставлена в текст скрипта в точку, указанную курсором. После этого можно отредактировать список аргументов.

По окончании редактирования скрипта следует нажать кнопку **Ok**. При обнаружении в скрипте каких-либо ошибок на экран будет выведено соответствующее сообщение. В большинстве случаев курсор установится в ту позицию, которая привела к появлению ошибки. Прежде чем скрипт будет сохранен, все ошибки должны быть исправлены.

Работа 3. Использование скриптов в InTouch.

Цель работы:

- замена фиктивных переменных;
- создание скриптов, запускаемых кнопками;
- создание скриптов по условию.

1. Откройте окно **Hopper**.

В этом окне представлен бункер высотой 200 пикселей. Синий указатель – курсор “привязан” к переменной **HopperSetpoint** (задание уровня). Красный указатель служит для отображения текущего значения переменной **HopperLevel** (уровень в бункере).

Запуск конвейера в дальнейшем следует «привязать» к дискретной переменной **Conveyor_on**. Конвейер будет включен/выключен при **Conveyor_on = 1/0**. Управление значением переменной **Conveyor_on** должно осуществляться кнопкой **Conveyor_on Toggle**.

Нижняя крышка бункера “привязана” к переменной **GateCount** типа **Integer**. Полное открытие (перемещение) крышки - 30 пикселей.

2. При импортировании окна **Hopper** все относящиеся к этому окну переменные преобразовались в фиктивные, которые следует заменить на обычные. Нажмите клавишу **F2** для выделения всех объектов в окне и выполните команду **Special/Substitute Tags** (замена переменных).

Current Name:	Required Type	New Name:
?d:Conveyor_on	Discrete	?d:Conveyor_on
?d:Gate	Discrete	?d:Gate
?i:\$Second	Analog	?i:\$Second
?i:GateCount	Analog	?i:GateCount
?r:HopperLevel	Analog	?r:HopperLevel
?r:HopperSetpoint	Analog	?r:HopperSetpoint

Buttons: OK, Cancel, Index, Convert, Replace

3. Нажмите **Convert**, затем **Local** для преобразования переменных.

Current Name:	Required Type	New Name:
?d:Conveyor_on	Discrete	Conveyor_on
?d:Gate	Discrete	Gate
?i:\$Second	Analog	\$Second
?i:GateCount	Analog	GateCount
?r:HopperLevel	Analog	HopperLevel
?r:HopperSetpoint	Analog	HopperSetpoint

Buttons: OK, Cancel, Index, Convert, Replace

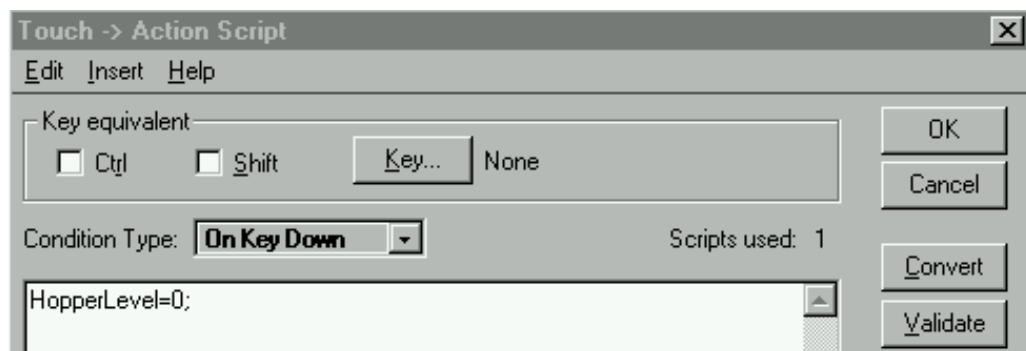
4. Определите в словаре переменных следующие переменные:

- **Conveyor_on**, Memory Discrete;
- **Gate**, Memory Discrete;
- **GateCount**, Memory Integer, Min Value – 0, Max Value - 100;
- **HopperLevel**, Memory Real, Min Value – 0, Max Value - 100;
- **HopperSetpoint**, Memory Real, Min Value – 0, Max Value – 100.

Анимирование кнопок HopperLevel.

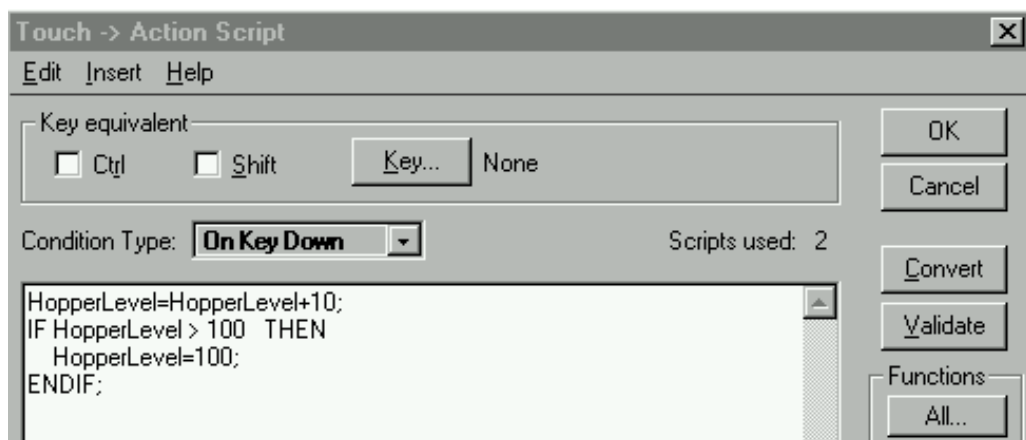
5. Создайте скрипт, запускаемый кнопкой «**HopperLevel=0**» (**Touch Pushbutton Action**). Тип скрипта - **On Key Down** (исполняется один раз в момент нажатия на кнопку) - следует выбрать из списка.

Нажатие кнопки «**HopperLevel=0**» должно обнулять уровень в бункере.



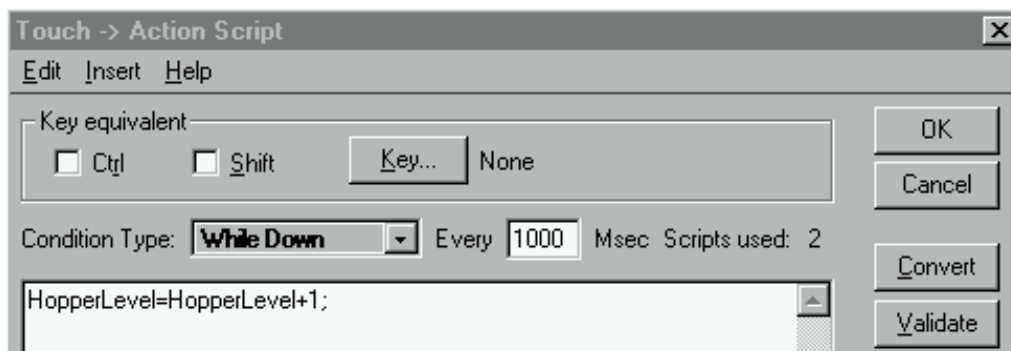
6. Создайте два скрипта **Touch Pushbutton Action**, запускаемых кнопкой «**HopperLevel+10**».

Тип первого скрипта - **On Key Down** (исполняется один раз в момент нажатия на кнопку). Щелчок по кнопке «**HopperLevel+10**» должно вызывать увеличение уровня на 10 единиц.



Тип второго скрипта - **While Down** (выполняется периодически через заданные интервалы времени пока нажата кнопка) – также выбирается из выпадающего списка.

При удержании кнопки «**HopperLevel+10**» уровень должен увеличиваться на единицу каждые 1000 мсек (1сек).

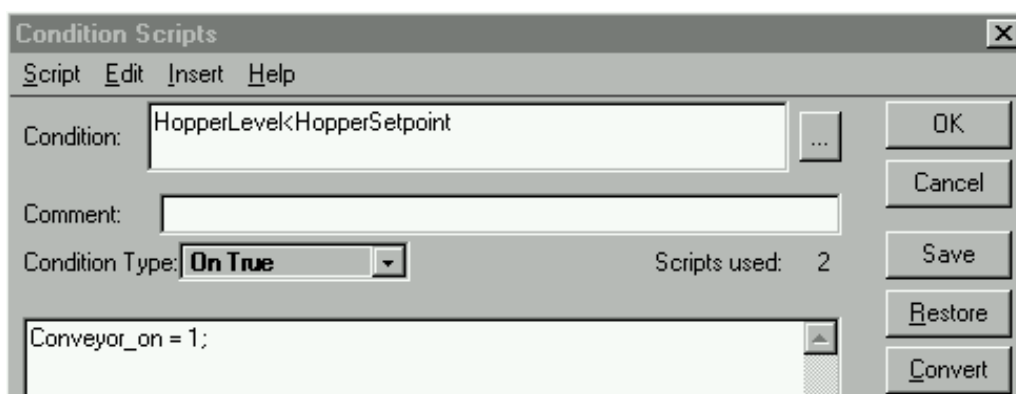


7. Перейдите в среду исполнения и проверьте работу кнопок.

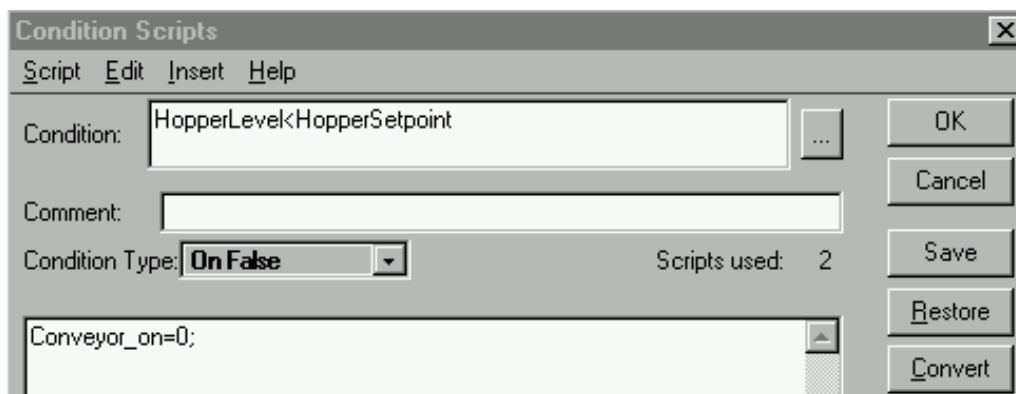
8. Для анимирования заполнения бункера создайте два скрипта по условию (**Condition Scripts**).

Первый скрипт устанавливает переменную «**Conveyor_on**» в **1** (конвейер включен), когда значение переменной **HopperLevel** меньше, чем заданное **HopperSetpoint** (Condition – условие, см. рис. внизу), и присваивает переменной «**Conveyor_on**» значение **0** (конвейер выключен) при других условиях.

Щелкните на **Special/Scripts/Condition Scripts** и создайте скрипт по условию типа **On True** (исполняется один раз при переходе условия в состояние **True**). Скрипт этого типа включает конвейер, если текущий уровень меньше заданного.

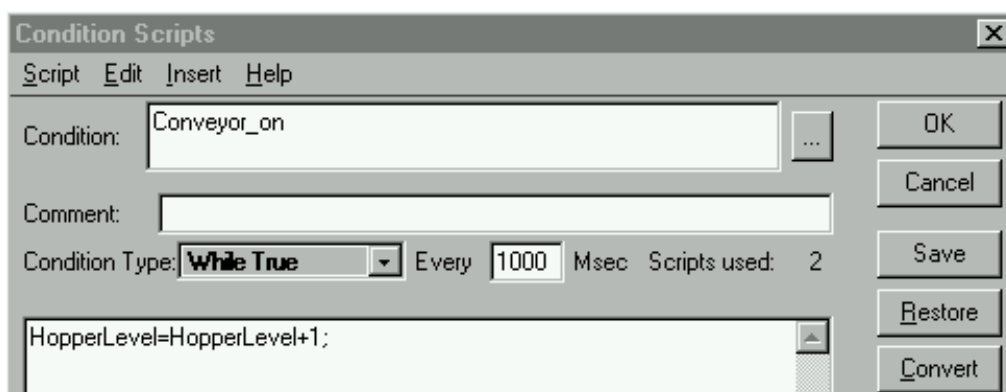


Но конвейер надо остановить, если уровень будет равен или больше заданного (переход условия в состояние **False**). Для этого состояния скрипт выглядит следующим образом:



9. Перейдите в среду исполнения для проверки работы скриптов.

10. Второй скрипт по условию будет увеличивать значение переменной **HopperLevel** на **1** каждые 1000 мсек (1 сек.), если переменная **Conveyor_on** = **TRUE** (конвейер включен). Щелкните **Script/New**, затем **Edit/Clear** для очистки окна редактора. Создайте скрипт типа **While True** (исполняется непрерывно, пока значением условия является **True**):

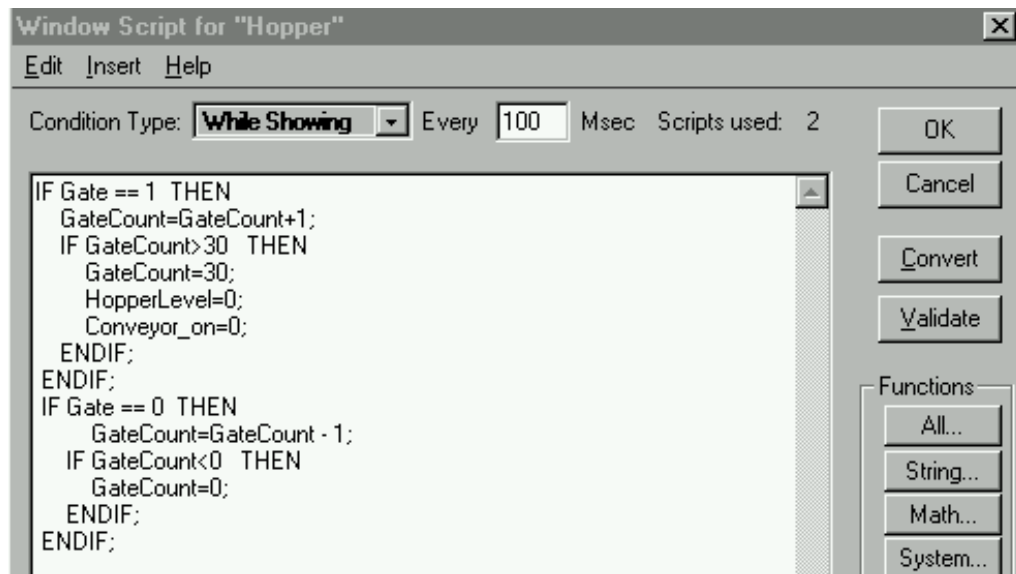


11. Перейдите в среду исполнения для проверки работы скриптов.

12. Нижней крышке бункера присвоено свойство **Location/Horizontal** (горизонтальное перемещение) и «привязано» оно к переменной **GateCount** типа **Integer**. Крышка закрыта при значении **GateCount=0**. Когда **GateCount=30**, крышка полностью открыта. Кнопка **Gate Control** управляет значением дискретной переменной **Gate**.

Создайте скрипт, который будет открывать крышку бункера (увеличивать **GateCount** до **30**), когда **Gate == TRUE**, и закрывать крышку (уменьшать **GateCount** до **0**), когда **Gate == FALSE**.

Наблюдать за работой конвейера и бункера можно только при открытом окне **Hopper**. Поэтому скрипт, анимирующий крышку бункера, должен быть связан с этим окном (скрипт уровня окна - **Window Scripts**). Его тип - **While Showing** (исполняется периодически через заданные промежутки времени пока окно отображается на экране).

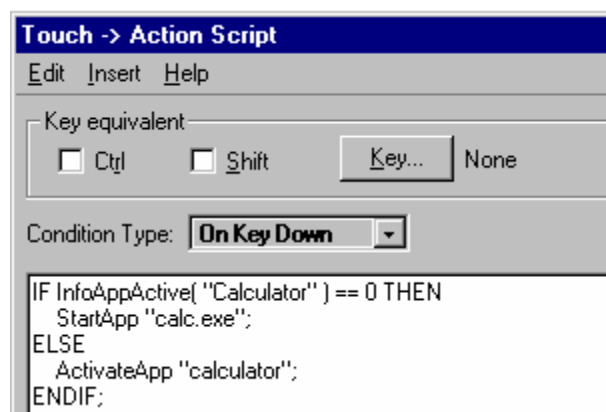


14. Добавьте скрипт типа **On Show** для этого окна (исполняется один раз в момент открытия окна) для того, чтобы все переменные и кнопки находились в начальном состоянии при открытии окна **Hopper**.



15. В **InTouch** имеется набор предопределенных функций. Функцию **InfoAppActive** можно использовать в скриптах для активизации программы **Calculator** (калькулятор). Если калькулятора нет на экране, то скрипт запустит его. Иначе, скрипт будет работать в фоновом режиме.

Создайте скрипт **Touch Pushbutton Action** типа **On Key Down** для кнопки «**Calculator**».



Функции также могут использоваться в поле **Expression** диалогов установки анимационных связей. Откройте окно **Menu** и анимируйте текст **# MB** для отображения размера свободного пространства на диске. Задайте текстовому объекту **#** свойство **Value Display/Analog**, введя в поле **Expression** следующее выражение:

InfoDisk («C»,2,\$Second) /1000000

Раздел 4. Алармы и события в InTouch.

InTouch поддерживает возможность отображения, регистрации и печати информации как об алармах, связанных с аналоговыми или логическими переменными, так и о системных событиях.

Типы алармов.

Стандартные (бывают и распределенные) алармы подразделяются по типу (**Type**) на несколько категорий.

Дискретные алармы срабатывают при изменении состояния дискретной переменной. При этом для срабатывания аларма можно использовать любое из двух состояний: **TRUE/ON (1)** или **FALSE/OFF (0)**.

Алармы	Стандартный тип
Discrete	DISC
Deviation - Major	LDEV
Deviation - Minor	SDEV
Rate - of - Change	ROC
Value - LoLo	LOLO
Value - Lo	LO
Value - High	HI
Value - HiHi	HIHI

Аналоговые алармы базируются на анализе выхода значений переменной за указанные верхние и нижние пределы. Как видно из таблицы, аналоговые алармы могут быть заданы в нескольких комбинациях:

- **Deviation** (отклонение от нормы);
- **Rate of Change - ROC** (скорость изменения).
- **Low** и **Low Low** (нижний и ниже нижнего);
- **High** и **High High** (верхний и выше верхнего);

С **InTouch** - переменной можно связывать алармы любого типа. В зависимости от типа переменной для нее можно определять один или более классов и типов алармов.

Приоритеты алармов.

Каждому аларму в **InTouch** соответствует некоторая величина, называемая приоритетом аларма. Этот приоритет характеризует важность данного аларма и принимает значения от 1 до 999 (наиболее серьезные алармы имеют приоритет 1). Организовав несколько диапазонов значений и связав алармы с каждым диапазоном, можно достаточно легко отфильтровать критические алармы от некритических. Выполнение анимационных функций, скриптов подтверждения, печать и просмотр информации также могут зависеть от приоритетов.

Группы алармов.

Каждая переменная связана с определенной логической группой алармов. Все эти группы определяются пользователем и могут быть организованы в иерархическую структуру до восьми уровней иерархии. Это позволяет сгруппировать алармы в зависимости от их организации, схемы размещения оборудования, приоритетов и любых других признаков.

Если пользователь не определил такую группу для конкретной переменной, то она автоматически связывается с корневой группой алармов **\$System**. С любой группой алармов можно связать как переменную, так и другую группу алармов. Взаимосвязи всех групп алармов представляются древовидной структурой, у которой в качестве корневой является группа **\$System**. Все определяемые группы алармов автоматически становятся потомками этой группы.

Для создания групп алармов в меню окна **WindowMaker** предусмотрена команда **Special/Alarm Groups** (группы алармов), вызывающая появление диалога **Alarm Groups**. При определении переменных в словаре **Tagname Dictionary** нажатие кнопки **Group** также выводит на экран этот диалог.



Определение условий аларма для переменной.

Условия возникновения аварийных ситуаций определяются в словаре переменных (**Tagname Dictionary**). Включение опции **Alarms** (или **Details & Alarms**) вызывает на экран диалог определения условий алармов, соответствующий типу переменной.

Диалог определения условий аларма для аналоговых переменных приведен ниже.

В поля **Alarm Value** вводятся соответствующие граничные значения переменной.

Поле **Value Deaddand** предназначено для ввода величины, на которую должно измениться значение переменной после возврата в диапазон допустимых значений, чтобы прекратилось состояние аварийной ситуации.

В поля **% Deviation** вводят отклонения в % от заданного значения (**Target**), на которые должна измениться переменная для возникновения аварийной ситуации.

Для конфигурирования аларма по скорости отклонения в поле **% per** следует ввести максимально возможное изменение переменной в % за секунду, минуту, час (Sec, Min, Hr).

Вывод информации об алармах.

Для отображения информации об аварийных ситуациях или событиях в **InTouch** предусмотрены два типа объектов (окон): **Alarm Summary** (текущие алармы) и **Alarm History** (архивная сводка алармов).

Создание системы алармов производится в несколько этапов:

- создание объекта (окна) вывода аварийной информации;
- конфигурирование окна вывода аварийной информации;
- форматирование сообщений;
- конфигурирование системы алармов (определение общих свойств алармов, свойств регистрации и печати).

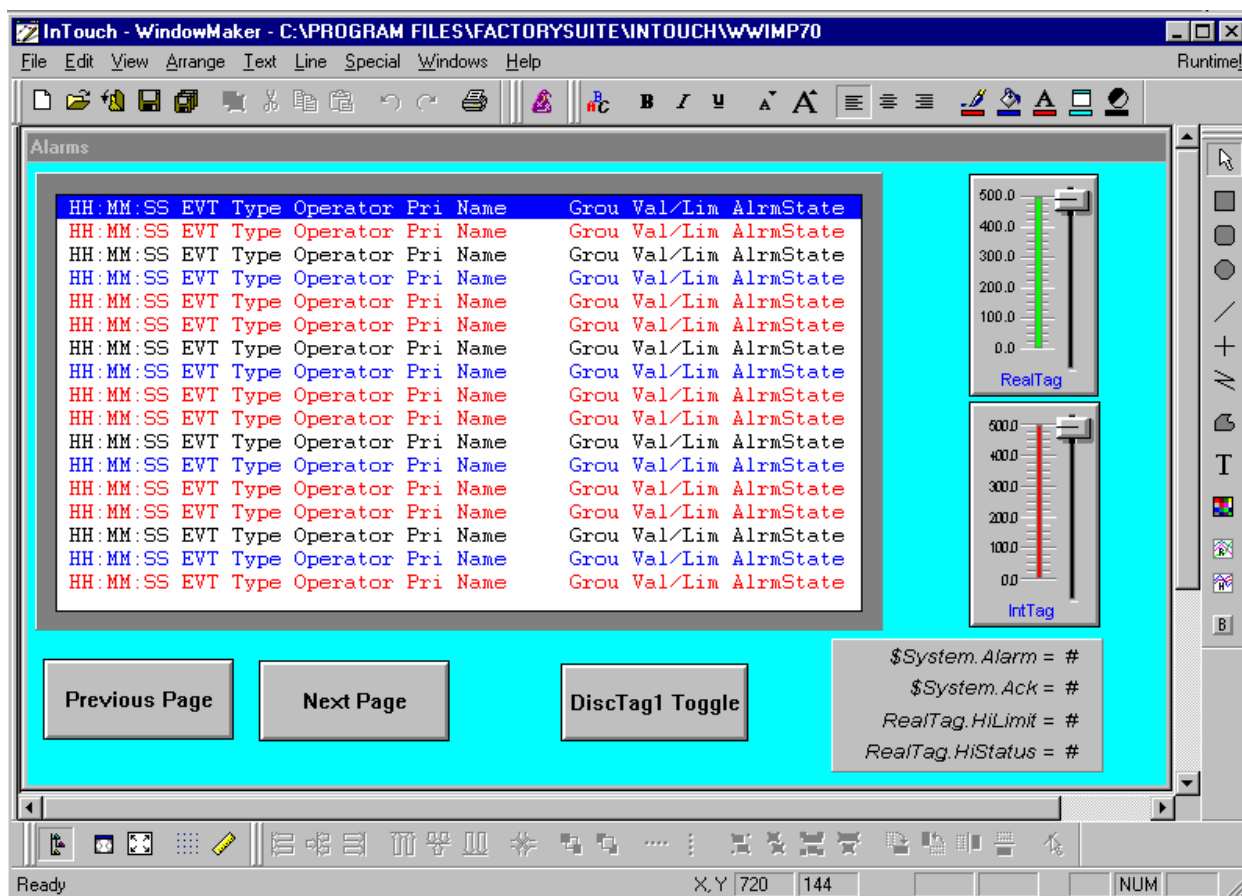
Работа 4. Алармы и события.

Цель работы:

- импортирование окон **Alarms** и **Real Time Trend** из другого приложения;
- конфигурирование окна **Alarms** для отображения алармов;
- создание окна для подтверждения алармов.

1. Импортируйте указанные окна из каталога **WWimp70**. Обратите внимание на то, что можно импортировать скрипты в их исходном виде.

2. Откройте окно **Alarms**:



3. Нажмите **F2**, воспользуйтесь командой **Special/Substitute Tags** и преобразуйте фиктивные переменные (см. работу № 3).

4. Так как среди переменных этого окна есть новые, еще не определенные в словаре переменные, программа выдаст запрос на их определение. Определите переменные **PgUp** и **PgDn** как **Memory Discrete**.

5. В словаре переменных (команда **Special/Tagname Dictionary**) для переменной **DiscTag1** выберите опцию **Details & Alarms** и в поле **Alarm State** (состояние, переход в которое приводит к аларму) установите **None**.

Tagname Dictionary

☐ Main
 ☐ Details
 ☐ Alarms
 ☒ Details & Alarms
 ☐ Members

[New] [Restore] [Delete] [Save] [Select...] [Cancel] [Close]

Tagname: Type: ... Memory Discrete

Group: ... \$System ☐ Read only ☒ Read Write

Comment:

☐ Log Data ☐ Log Events ☐ Retentive Value

Initial Value: ☐ On ☒ Off

On Msg: Off Msg:

Alarm State: ☐ On ☐ Off ☒ None

6. Для переменной **IntTag** выберите опцию **Details & Alarms** (см. рис.) и задайте нижний предел **Low = 100** и верхний **High = 400**. Установите приоритет, равный **1** (самый высокий).

Tagname Dictionary

☐ Main
 ☐ Details
 ☐ Alarms
 ☒ Details & Alarms
 ☐ Members

[New] [Restore] [Delete] [Save] [Select...] [Cancel] [Close]

Tagname: Type: ... Memory Integer

Group: ... \$System ☐ Read only ☒ Read Write

Comment:

☐ Log Data ☐ Log Events ☐ Retentive Value ☐ Retentive Parameters

Initial Value: Eng Units:

Min Value: Deadband:

Max Value: Log Deadband:

Alarm Value	Pri
<input type="checkbox"/> LoLo	
<input checked="" type="checkbox"/> Low 100	1
<input checked="" type="checkbox"/> High 400	1
<input type="checkbox"/> HiHi	

Value Deadband:

☐ Minor Deviation

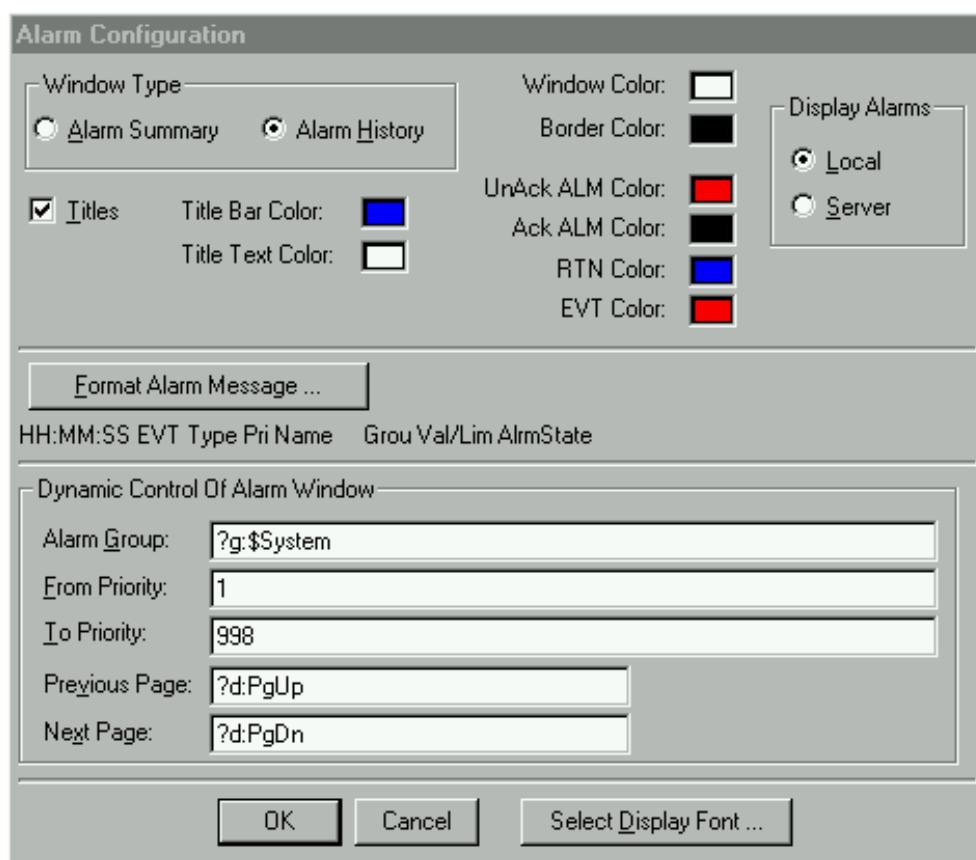
☐ Major Deviation

Deviation Deadband %:

☐ Rate of Change

7. Для переменной **RealTag** выберите опцию **Details & Alarms** и установите самый нижний предел **LoLo = 30**, **Low = 50**, **High = 400** и самый высокий - **HiHi = 450**. Установите приоритет (**Pri**) = 1.

8. Дважды щелкните на объекте **Alarm** для его конфигурирования. Появится диалог конфигурирования стандартного окна алармов **Alarm Configuration**:



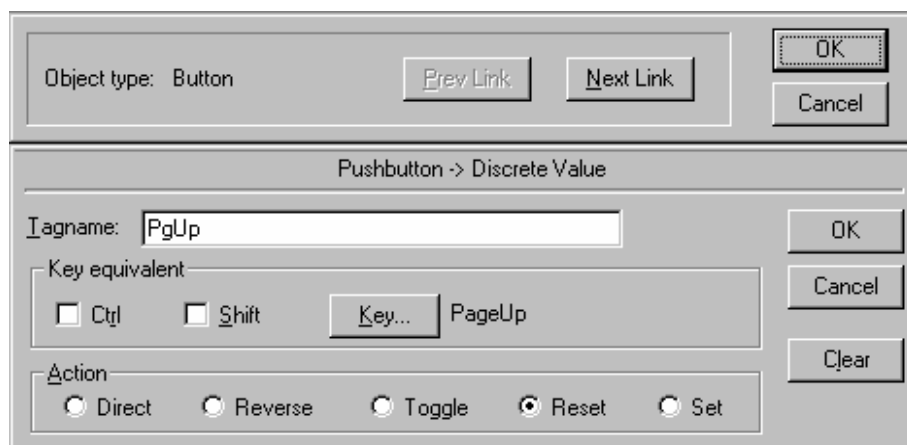
The **Alarm Configuration** dialog box is used to configure the alarm window. It includes the following sections:

- Window Type:** Radio buttons for **Alarm Summary** and **Alarm History**. **Alarm History** is selected.
- Window Color:** A color selection box.
- Border Color:** A color selection box.
- Display Alarms:** Radio buttons for **Local** and **Server**. **Local** is selected.
- Titles:** A checked checkbox. Below it are **Title Bar Color:** (blue) and **Title Text Color:** (white).
- UnAck ALM Color:** (red)
- Ack ALM Color:** (black)
- RTN Color:** (blue)
- EVT Color:** (red)
- Format Alarm Message ...** button.
- Dynamic Control Of Alarm Window:**
 - Alarm Group:** ?g:\$System
 - From Priority:** 1
 - To Priority:** 998
 - Previous Page:** ?d:PgUp
 - Next Page:** ?d:PgDn
- Buttons:** **OK**, **Cancel**, and **Select Display Font ...**.

9. Щелкните на кнопке **Format Alarm Message** для просмотра формата сообщений об алармах.

10. Дважды щелкните на кнопках **Previous Page** и **Next Page**, чтобы просмотреть свойства **Touch Pushbutton Discrete Value** и **Visibility**. Диалоги конфигурирования этих кнопок должны выглядеть следующим образом:

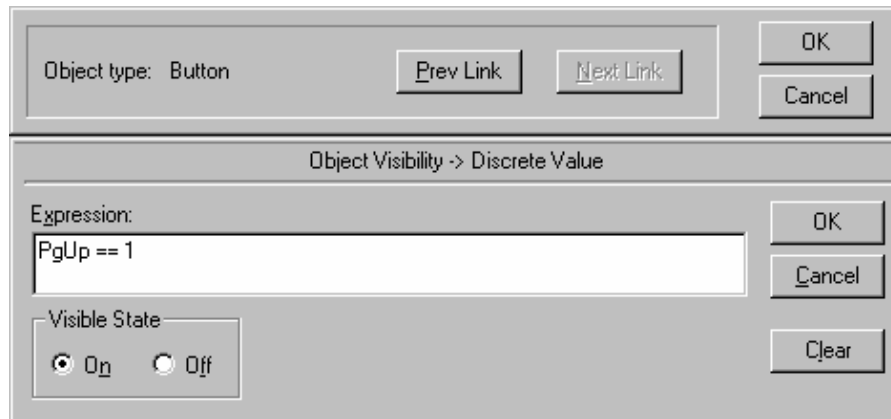
- для свойства **Touch Pushbutton Discrete Value**:



The **Touch Pushbutton Discrete Value** configuration dialog box is used to configure the discrete value of a touch pushbutton. It includes the following sections:

- Object type:** Button
- Prev Link** and **Next Link** buttons.
- Buttons:** **OK** and **Cancel**.
- Pushbutton -> Discrete Value** section:
 - Tagname:** PgUp
 - Key equivalent:**
 - ☐ Ctl
 - ☐ Shift
 - Key...** button
 - PageUp
 - Action:**
 - ☐ Direct
 - ☐ Reverse
 - ☐ Toggle
 - ☒ Reset
 - ☐ Set
- Buttons:** **OK**, **Cancel**, and **Clear**.

- для свойства **Visibility**:



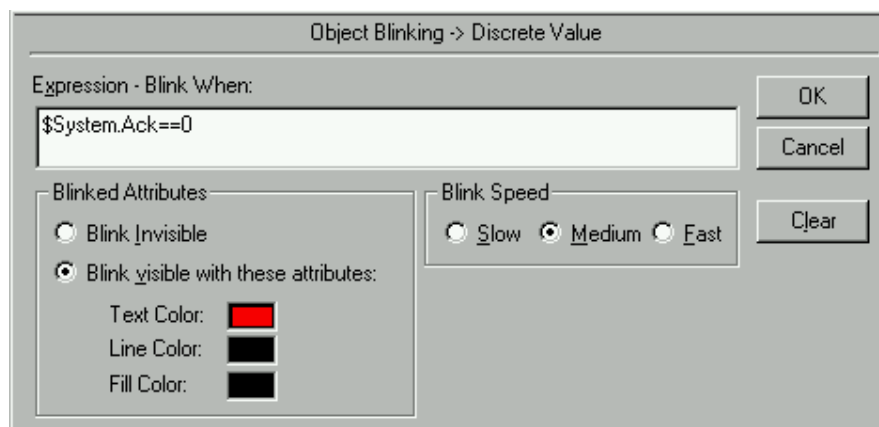
11. Перейдите в **WindowViewer** и, перемещая курсор движкового регулятора, проверьте отображение алармов в окне. Обратите внимание, что при этом изменяются значения полей переменных на панели в правом нижнем углу окна. Введите новое значение предела для **RealTag**.

12. Создайте окно с именем **Alarm Acknowledge** (подтверждение алармов). Это окно должно быть типа **Popup**. Оно будет появляться на экране при выходе переменных **RealTag** и **IntTag** за установленные границы. Установите требуемый размер окна, отмените в его свойствах опции **Title Bar** и **Size Controls**.

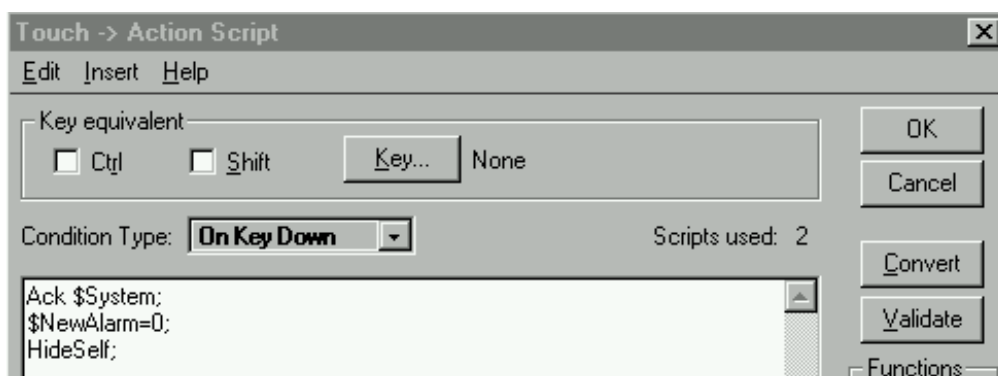


13. В окно введите текст «**Внимание!!!**» и под текстом добавьте кнопку **Acknowledge Alarms** (подтвердить алармы).

14. Для текста «**Внимание!!!**» задайте свойство мерцания (**Blink**). Не забудьте поставить точку после имени переменной **\$System**.

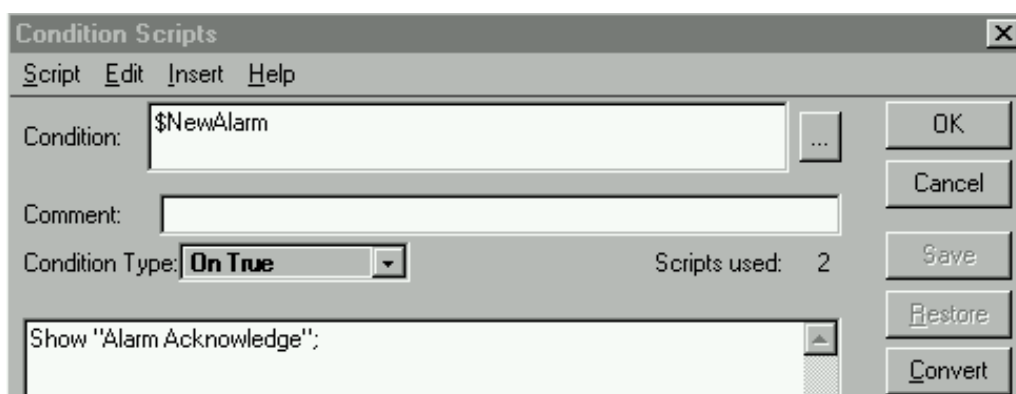


15. Двойным щелчком по кнопке **Acknowledge Alarms** вызовите диалог динамических свойств и создайте скрипт **Touch Pushbutton/Action**, который будет исполняться один раз в момент нажатия кнопки.



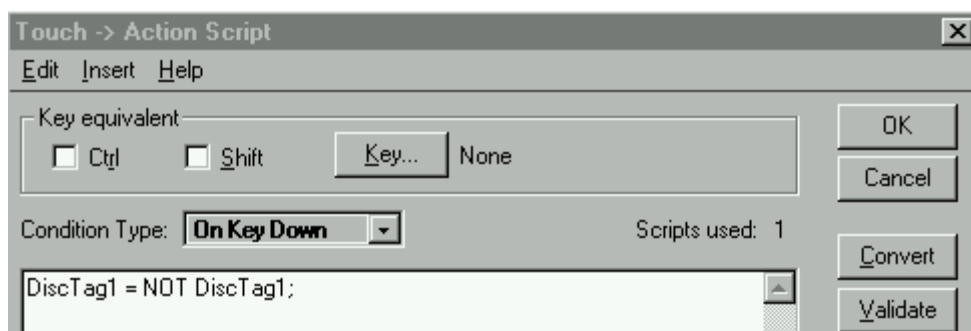
Скрипт предназначен для подтверждения всех алармов (**Ack \$System**). При появлении нового аларма системная переменная **\$NewAlarm** по умолчанию равна 1, после подтверждения (нажатия кнопки) ее надо сбросить в ноль (**\$NewAlarm=0**) и закрыть выпадающее окно (**HideSelf**). Сохраните окно **Alarm Acknowledge** и спрячьте его щелчком по свободному месту окна **Alarms**.

16. А чтобы окно **Alarm Acknowledge** появлялось при возникновении алармов, командой **Spetcial/Scripts/Condition Scripts** создайте скрипт по условию типа **On True**. Нажмите **Script/New** для создания нового скрипта и заполните диалог в соответствии с рисунком.

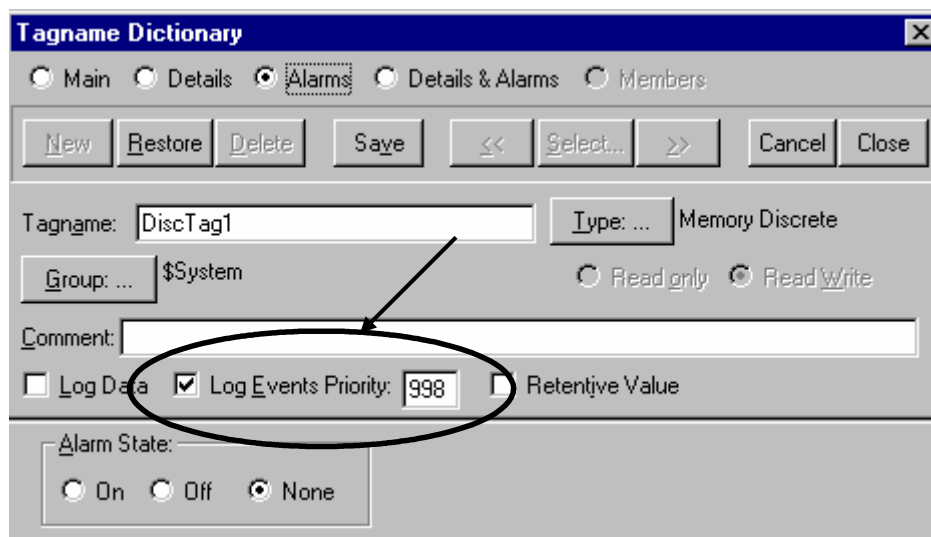


17. Переключитесь в среду исполнения и переместите несколько раз курсоры движковых регуляторов переменных **RealTag** и **IntTag** в крайние положения. Окно **Alarm Acknowledge** должно появляться при возникновении алармов, а при их подтверждении - исчезать. После подтверждения аларма в окне **Alarms** должна появляться новая строка с соответствующей информацией.

19. Возвратясь в среду разработки, дважды щелкните на кнопке **DiscTag1 Toggle** для просмотра скрипта **Touch Pushbutton/Action**:



20. Убедитесь, что в словаре переменных для переменной **DiscTag1** включена опция **Log Events** (регистрация событий) с приоритетом **Priority=998**.



21. Перейдите в среду исполнения и нажмите несколько раз кнопку **DiscTag1 Toggle**. События (нажатие кнопки оператором) должны отображаться строкой в окне алармов.

Раздел 5. Тренды в InTouch.

Графическое представление значений технологических параметров во времени способствует лучшему пониманию динамики параметров технологического процесса. Поэтому подсистема создания трендов и хранения информации о параметрах с целью ее дальнейшего анализа является неотъемлемой частью любой **SCADA** - системы.

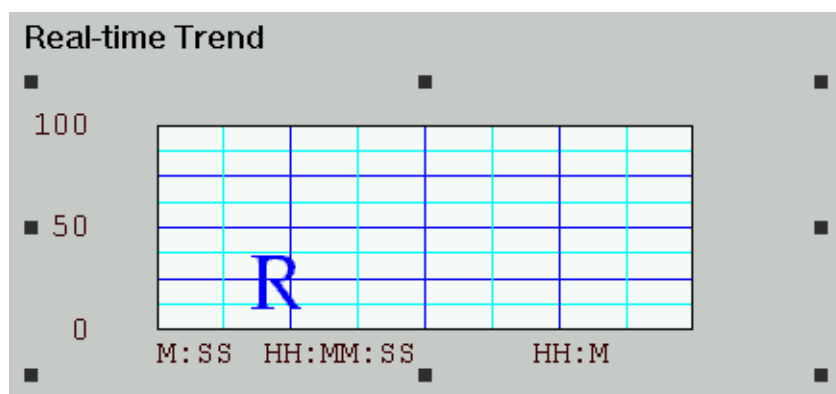
Тренды реального времени (Real Time Trend) отображают динамические изменения параметров в текущем времени. При появлении нового значения параметра в окне тренда происходит прокрутка графика справа налево. Таким образом текущее значение параметра выводится всегда в правой части окна.

Тренды становятся историческими (Historical Trend) после того, как данные будут записаны на диск и можно будет использовать режим прокрутки значений переменных назад с целью посмотреть предыдущие значения. Отображаемые значения переменных в таком режиме будут неподвижны и будут отображаться только за определенный период.

• **Тренды реального времени** являются динамическими объектами. Они позволяют выводить изменения значений переменных, как только они происходят для любой конкретной переменной или для выражения, которое содержит одну или несколько переменных.

Чтобы создать тренд реального времени, необходимо:

- выбрать инструмент *тренд реального времени* в панели инструментов **WindowMaker**;
- щелкнуть в окне, затем переместить мышь по диагонали и сформировать прямоугольник необходимого размера;
- отпустить кнопку мыши, что вызовет появление тренда реального времени в окне.



При создании тренда реального времени настройки его конфигурации устанавливаются по умолчанию (настройки предыдущего тренда).

Для конфигурирования тренда реального времени следует либо дважды щелкнуть на созданном объекте, либо, предварительно выбрав объект, запустить команду **Special/Animation Links**. На экране появится диалог **Real Time Trend Configuration** (конфигурирование тренда реального времени).

В поле **Time** следует ввести диапазон времени, охватываемый трендом на экране (**Time Span**), который будет представлен горизонтальной осью тренда. Размерность времени определяется опциями **Sec** - секунды, **Min** - минуты, **Hr** - часы. Например, если введено **30** и выбрана опция **Min**, то диапазон времени будет равен 30 минутам.

Real Time Trend Configuration

Comment:

Time
 Time Span:
☐ Sec ☒ Min ☐ Hr

Sample
 Interval:
☐ Msec ☒ Sec ☐ Min ☐ Hr

Color
 Chart Color:
 Border Color:

Time Divisions
 Number of Major Div:
 Minor Div/Major Div:
☐ Top Labels ☒ Bottom Labels
 Major Div/Time Label:
 HH:MM:SS Display: ☒ HH ☒ MM ☒ SS

Value Divisions
 Number of Major Div:
 Minor Div/Major Div:
☒ Left Labels ☐ Right Labels
 Major Div/Value Label:
 Min Value: Max:

Pen:

Pen:	Expression:	Color:	Width:
1	<input type="text"/>	<input type="text"/>	<input type="text" value="1"/>
2	<input type="text"/>	<input type="text"/>	<input type="text" value="1"/>
3	<input type="text"/>	<input type="text"/>	<input type="text" value="1"/>
4	<input type="text"/>	<input type="text"/>	<input type="text" value="1"/>

OK Cancel Clear Select Display Font... ☐ Only update when in memory

Поле **Sample** предназначено для определения частоты вывода на график значений переменной.

В поле **Color** производится выбор цвета (**Chart Color** , **Border Color**) для фона графика и рамки тренда, соответственно.

В полях **Number of Major Div** определяется число больших делений сетки графика по оси времени и оси значений переменной. Здесь же следует определить цвет сетки. Число больших делений сетки должно быть **четным** и **кратным** количеству малых делений.

В полях **Minor Div/Major Div** определяется число маленьких делений, выводимых внутри каждого большого деления сетки. Здесь же следует определить их цвет.

Для задания места нахождения временных отметок (сверху - **Top Labels**, снизу - **Bottom Labels**, отсутствуют) следует отметить соответствующую опцию. Если отметки времени конфигурируются, то надо определить число больших делений сетки, которое приходится на отметку времени (поле **Major Div/Time Label**), а также формат вывода (**HH:MM:SS Display**) и цвет.

Диапазон значений, задаваемый в полях **Min Value** и **Max**, относится ко всем отображаемым на графике переменным. Очевидно, все переменные должны иметь одни и те же размерные единицы (**EU**).

Конфигурирование перьев тренда включает выбор имени переменной или выражения, цвета и толщины линии для каждого пера (поле **Expression**).

- **Исторические (архивные) тренды** не являются динамическими. Они обеспечивают «снимок» состояния данных за прошедшее время, то есть по архивным данным. В отличие от трендов реального времени исторические тренды обновляются только по команде - при запуске скрипта, изменении значения выражения или нажатии оператором соответствующей кнопки.

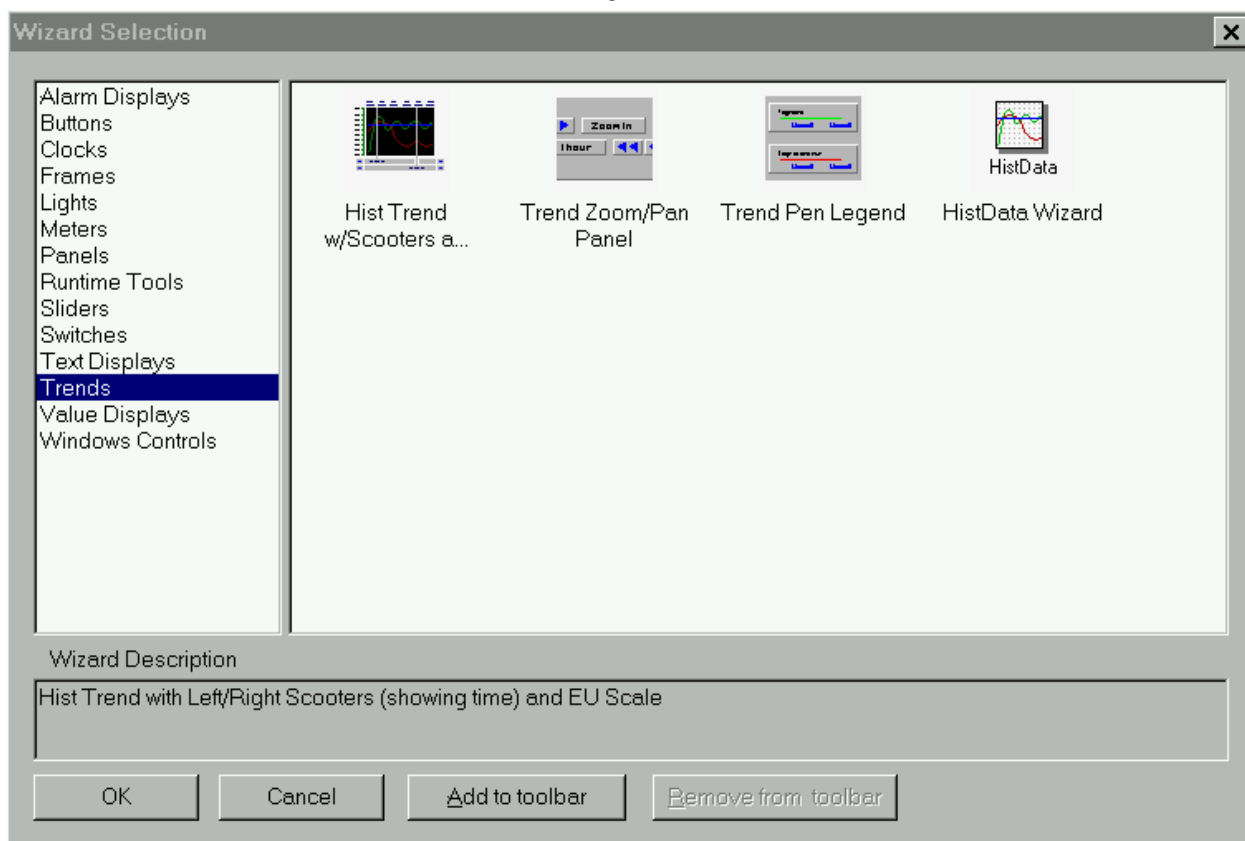
При конфигурировании архивного тренда можно создать «визирь» (ползунки, бегунки), с помощью которых удобно получить значения всех отображаемых переменных на один и тот же момент времени. Бегунки архивного тренда представляют собой позиционные индикаторы на временной оси, положение которых определяет объем извлекаемых данных. Связав объект «движковый регулятор» с полем бегунка, можно осуществлять перемещение вдоль архивного тренда. Кроме того, имеются функции вычисления среднего, минимального и максимального значений в определенном бегунком положении. Можно создать правый и левый бегунки и производить обработку данных кривой, расположенной между бегунками. Вычисляются следующие величины: среднее, минимальное, максимальное, отношение мин/макс и стандартное отклонение. В зависимости от положения бегунков на оси можно реализовать и другие функции (увеличение и уменьшение заключенной между бегунками области графика).

Все сказанное выше о механизме создания тренда реального времени инструментом **Real Time Trend** в среде разработки **WindowMaker** и о его последующем конфигурировании можно отнести и к архивному тренду, создаваемому инструментом **Historical Trend** среды разработки.

Предлагаемый ниже способ создания и конфигурирования архивного тренда предполагает использование мастер-средств библиотеки **Wizard**.

Нажатие кнопки выбора мастер-средств в панели инструментов вызывает появление на экране диалога **Wizard Selection**. После выбора в списке категории **Trends** этот диалог будет выглядеть так, как показано на рисунке ниже.

Следует отметить, что любой мастер-объект для быстрого доступа к нему может быть вставлен в инструментальную панель **Wizard** окна **WindowMaker**. Для этого надо выбрать мастер-объект в правой части диалога и нажать кнопку **Add to Toolbar**. Иконка с выбранным мастер-объектом появится в панели **Wizard**.



После выбора из предложенного набора мастер-средств **Hist Trend with Scooters** (архивный тренд с бегунками) и щелчка по **Ok** программа возвращает пользователя в среду разработки. Курсор мыши при этом примет форму вставки. Осталось выбрать место для вставки и щелкнуть мышью.

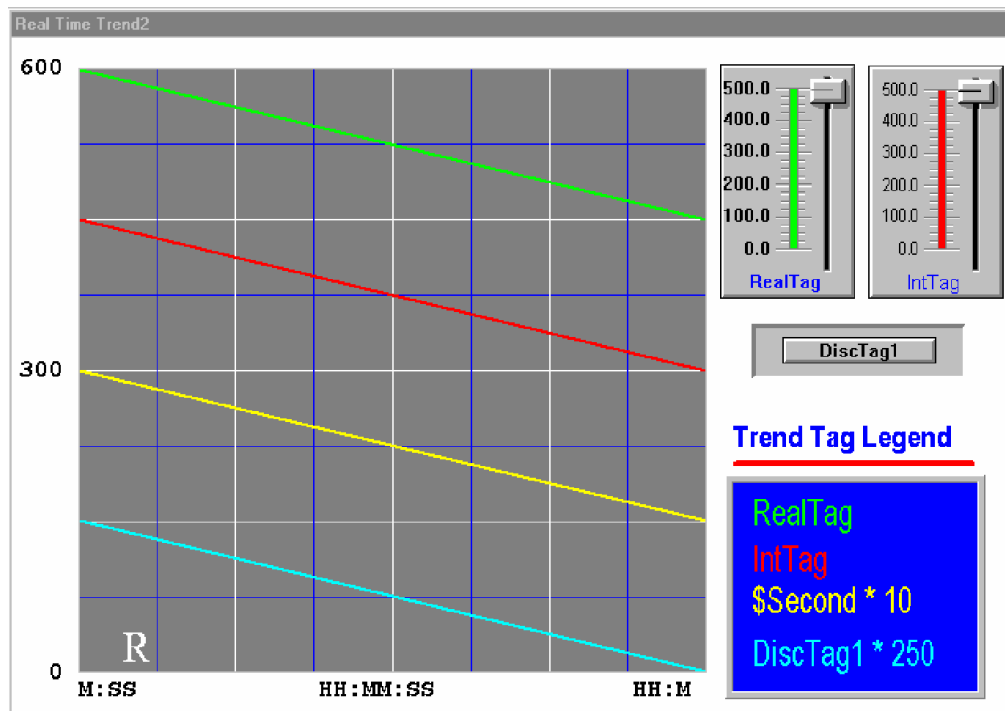
Чтобы добавить в тренд функции масштабирования и перемещения или элементы управления перьями, следует использовать объекты **Zoom/Pan** и **Trend Pen Legend**, соответственно (см. рис.). Объект **HistData** предназначен для сохранения значений переменных в файле истории. Для того, чтобы все перечисленные компоненты работали совместно, они должны иметь одинаковые имена (**Hist Trend**).

Работа 5. Тренды реального времени и архивные тренды.

Цель работы:

- конфигурирование перьев в окне **Real Time Trend**;
- создание окна **Historical Trend** (архивный тренд);
- создание объекта «**архивный тренд**» с использованием библиотеки **Wizard** для отображения переменных **RealTag** и **IntTag**.

1. Откройте окно **Real Time Trend** и произведите в нем замену фиктивных переменных.



2. Дважды щелкните на объекте **Trend** для вывода диалога **Real Time Trend Configuration**.

The screenshot shows the 'Real Time Trend Configuration' dialog box. It has a 'Comment:' field at the top. Below it are three sections: 'Time', 'Sample', and 'Color'. The 'Time' section has 'Time Span' set to 30 and radio buttons for 'Sec', 'Min', and 'Hr'. The 'Sample' section has 'Interval' set to 1 and radio buttons for 'Msec', 'Sec', 'Min', and 'Hr'. The 'Color' section has 'Chart Color' and 'Border Color' dropdowns. Below these are 'Time Divisions' and 'Value Divisions' sections. 'Time Divisions' includes 'Number of Major Div' (4), 'Minor Div/Major Div' (2), checkboxes for 'Top Labels' and 'Bottom Labels', 'Major Div/Time Label' (2), and 'HH:MM:SS Display' with checkboxes for 'HH', 'MM', and 'SS'. 'Value Divisions' includes 'Number of Major Div' (4), 'Minor Div/Major Div' (2), checkboxes for 'Left Labels' and 'Right Labels', 'Major Div/Value Label' (2), 'Min Value' (0), and 'Max' (600). At the bottom, there is a table for 'Pen' settings with columns for 'Pen', 'Expression', 'Color', and 'Width'. The first row is filled with '1', 'RealTag', a green color swatch, and '2'. The other three rows are empty. At the very bottom are buttons for 'OK', 'Cancel', 'Clear', 'Select Display Font ...', and a checkbox for 'Only update when in memory'.

3. Заполните поля **Expression** для перьев 2, 3, и 4. Для этого в соответствующие строки введите имена переменных, которые будут отображаться этими перьями:

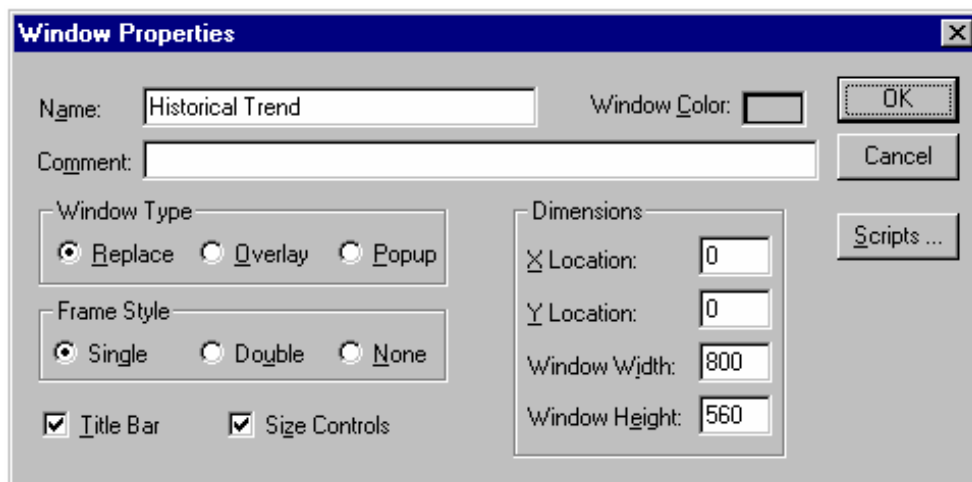
Pen 2: **IntTag** ;

Pen 3: **\$Second*10** ;

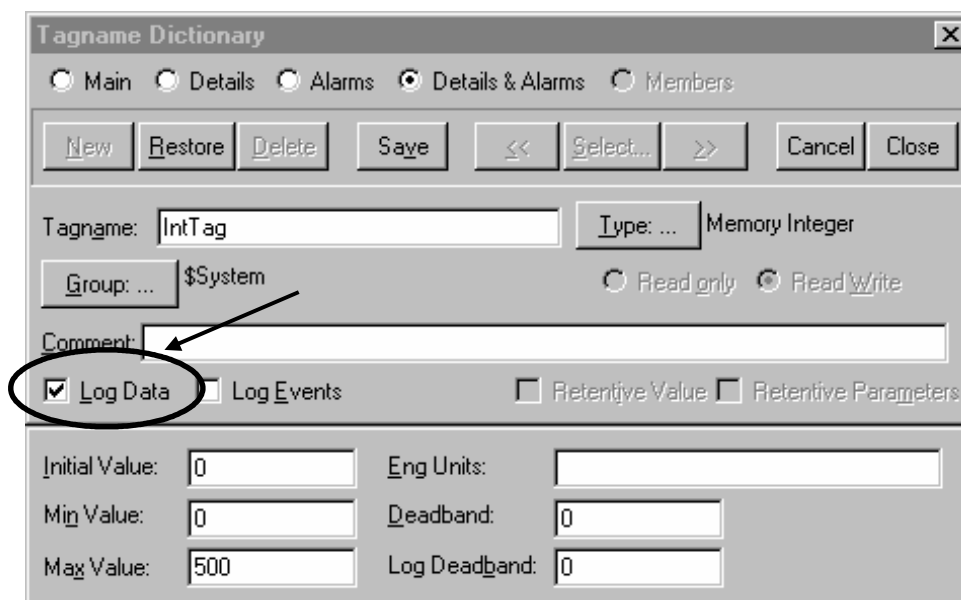
Pen 4: **DiscTag1*250** .

4. Нажмите **Ok** и перейдите в среду исполнения. Несколько раз измените положение движков регуляторов и нажмите на кнопку **DiscTag1**. Убедитесь в правильности отображения информации на тренде.

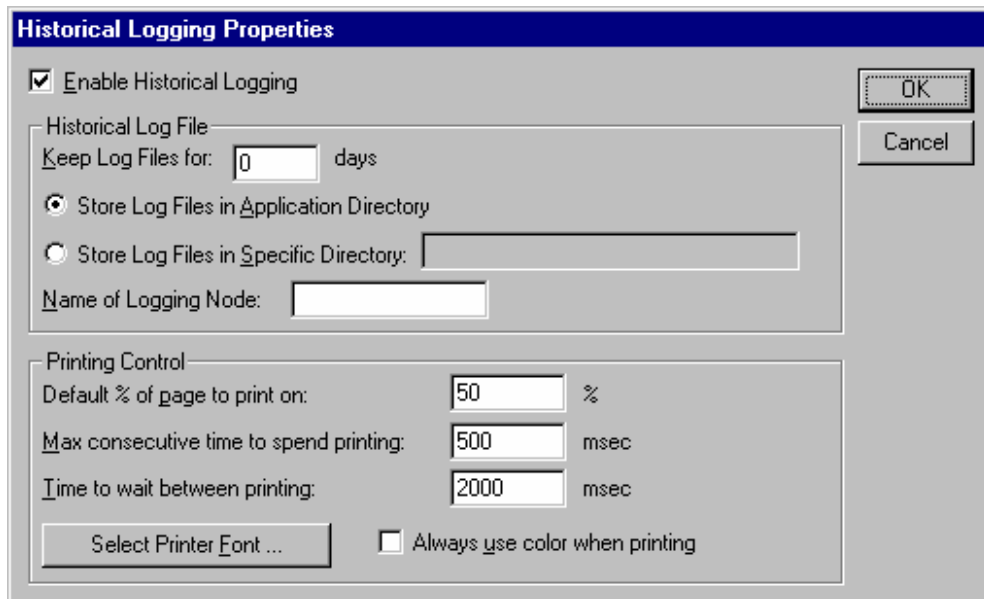
5. Создайте новое окно **Historical Trend** (см. рисунок ниже).



6. Для переменных **IntTag** и **RealTag** установите в словаре переменных опцию **Log Data** (регистрация данных).

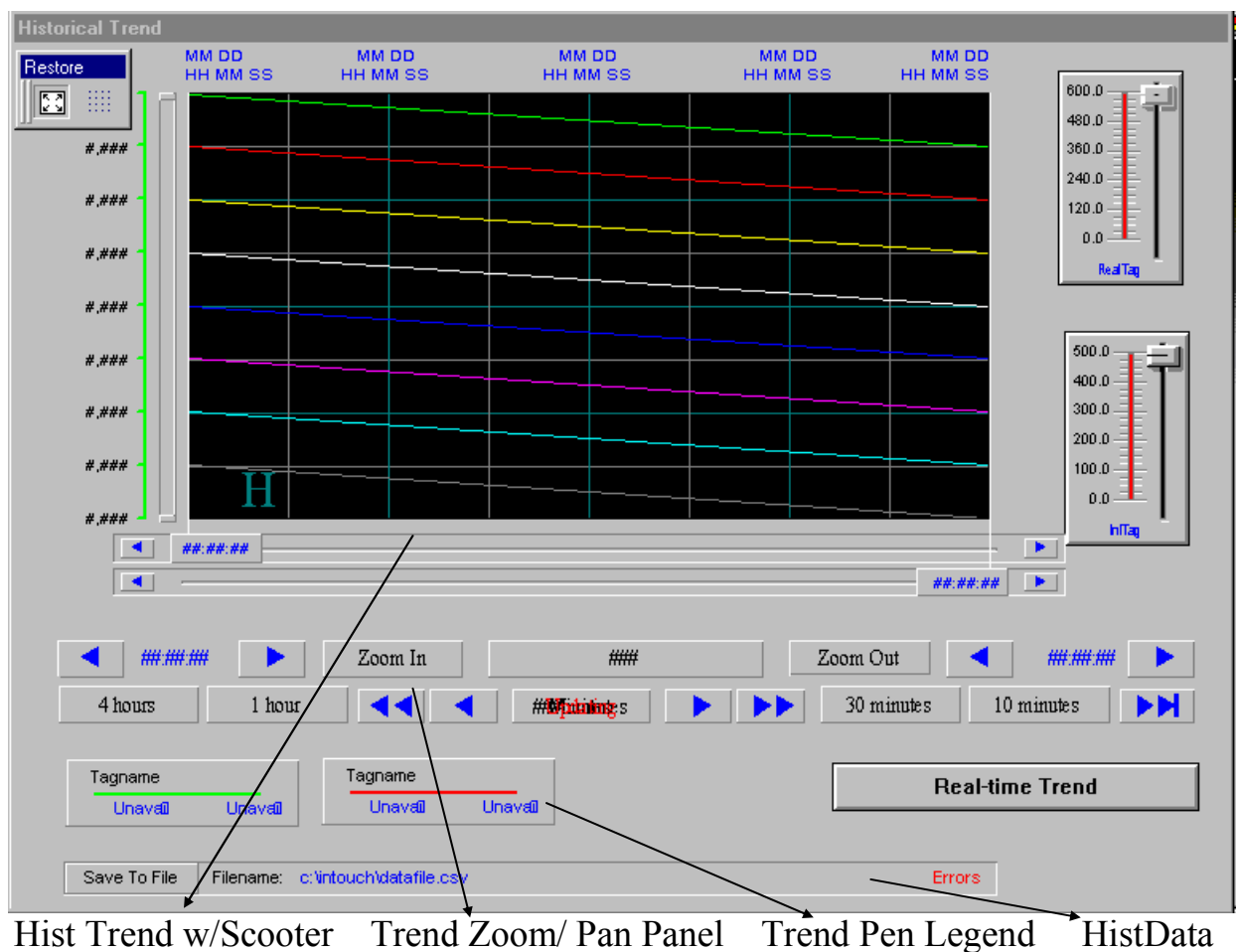


7. Выполните команду **Special/Configure/Historical Logging** и включите опцию **Enable Historical Logging** (разрешить архивную регистрацию).



8. Закройте **WindowViewer**, если он запущен.

9. Создайте в окне **Historical Trend** объекты с использованием мастер-объектов библиотеки **Wizard** (**Hist Trend w/Scooter and Scales**, **Trend Zoom/Pan Panel**, **Trend Pen Legend** и **HistData**) и разместите их в соответствии с рисунком. Дублируйте объект **Trend Pen Legend**.



Добавьте в окно два движковых регулятора и кнопку с текстом **Real Time Trend** (эта кнопка позже будет привязана к окну **Real Time Trend**).

10. Дважды щелкните на тренде. Появится диалог **Historical Trend Chart Wizard**.

11. Нажмите кнопку **Suggest** для заполнения полей. Предложенные имена переменных автоматически появятся в полях диалога. Запомните эти имена, так как все мастер-объекты, используемые в этом окне, будут их использовать.

12. Нажмите кнопку **Pens...**, чтобы выбрать перья для отображения переменных **RealTag** и **IntTag**. Появится диалог **Trend Pens**.

13. Дважды щелкните в поле переменных для вызова браузера переменных. После выбора переменной она будет вставлена в соответствующую строку диалога. Затем нажмите **Ok** для возврата к предыдущему диалогу и еще раз **Ok** для выхода в окно **Historical Trend**.

14. Дважды щелкните на панели **Trend Zoom/Pan Panel**. Появится диалог **Trend Panel Wizard**.

Нажмите кнопку **Suggest** для заполнения полей диалога. Убедитесь, что используется переменная с именем **HistTrend**, и нажмите **Ok**.

15. Дважды щелкните на левом объекте панели **Trend Pen Legend** для переменной **RealTag**.

Tagname
Unavail
Unavail

Trend Legend Wizard

The Trend Legend requires 2 Tags. Enter these below.
If the tags that you enter below do not exist, the Wizard will create them. Click Suggest for suggestions on names.

Hist Trend: (Hist Trend)

Pen Scale: (Memory Integer)

The Pen Scale tag is used to display Engineering Units. If you also use the Trend Chart Wizard, specify this same Pen Scale tagname there as well.

Colors

Tagname Label:

Values:

Trend Pen:

Pen Number:

Value Format: ▼

For Invalid Values Display:

Нажмите кнопку **Suggest** для ввода переменных. В группе **Color** выберите зеленый цвет в поле **Trend Pen**. Убедитесь, что в поле **Pen Number** введена 1, остальные поля оставьте без изменений. Нажмите **Ok** и дважды щелкните мышью по второму объекту панели **Trend Pen Legend** для переменной **IntTag**. Выберите красный цвет в поле **Trend Pen**. В поле **Pen Number** введите 2.

16. Дважды щелкните на объекте **HistData** и в появившемся диалоге нажмите кнопку **Suggest** для ввода имени переменной в поле **Hist Trend**.

HistData Panel Wizard

This Wizard is designed to work in conjunction with a particular Hist Trend. Enter the tag for this trend below:

Hist Trend: (Hist Trend)

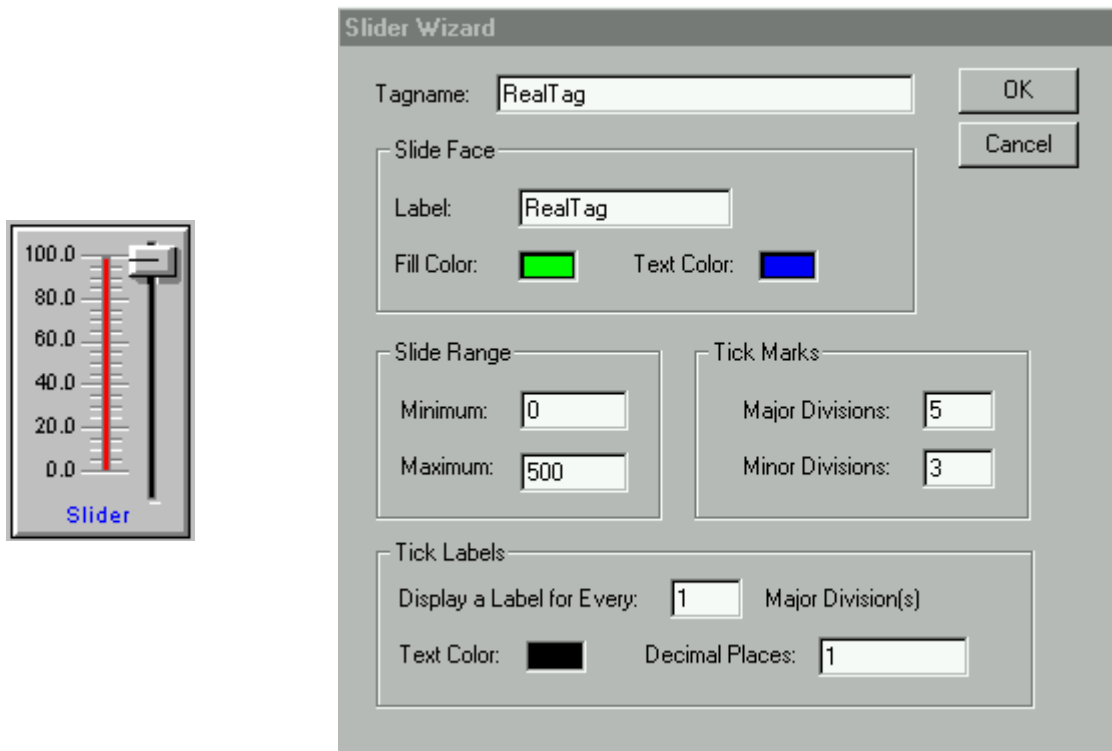
If the tag that you enter above does not exist, the Wizard

Number of Records to Write per CSV File:

At runtime, pressing the button on this Panel will save to a CSV file all the data data between the Scooters on the Trend. To do this, the Panel will write a fixed number of records to the file (specified above), no matter what the actual time duration of the data is. These records will be evenly spaced in time.

Введите **50** в поле **Number of Records to Write per CSV File** (количество записей в CSV - файле) и нажмите **Ok**.

17. Дважды щелкните на первом движковом регуляторе для вызова диалога **Slider Wizard**. В поле **Tagname** введите имя переменной **RealTag**.



В поле **Slide Face/Label** введите **RealTag** и установите диапазон изменения **0 - 500** (поле **Slide Range**). Измените цвет заполнения (**Fill Color**) на зеленый, остальные поля оставьте без изменений. Нажмите **Ok**.

18. Повторите эту процедуру для второго движкового регулятора и переменной **IntTag**, выбрав в поле **Fill Color** красный цвет.

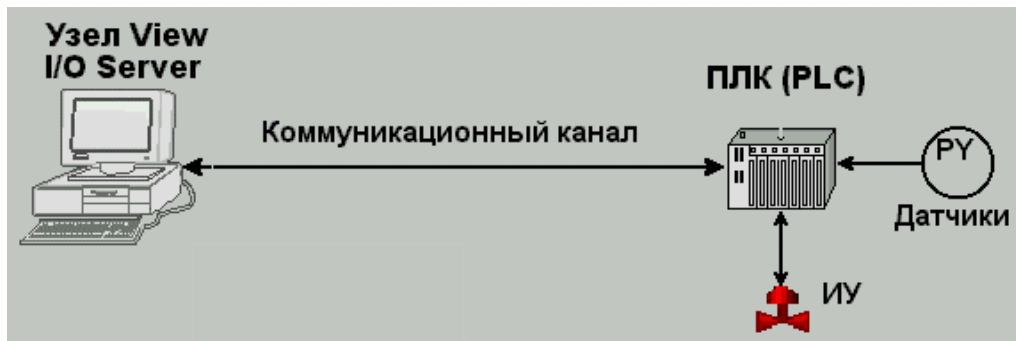
19. Перейдите в **WindowViewer** и переместите несколько раз движки регуляторов для сохранения значений в файле истории. (В появившемся приглашении запустить **HistData** нажмите **Yes**.)

20. С помощью клавиш «больше/меньше» панели **Trend Zoom/Pan Panel** выберите интервал времени для анализа поведения переменных. Выделите левым и правым маркерами (scooters) фрагмент тренда и нажмите кнопку **Zoom In** (**Zoom Out**) для масштабирования. Каждый щелчок по кнопке **Zoom In** (**Zoom Out**) увеличивает (уменьшает) масштаб по оси времени в два раза.

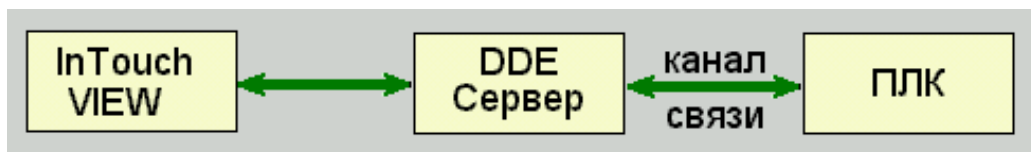
21. Окно **Historical Trend** - последнее, седьмое окно созданного приложения. Для обеспечения навигации по окнам приложения вызовите окно **Menu** и поочередно присвойте всем кнопкам окна свойство **Touch Pushbutton/Show Window** - показать окно). В диалоге выбора окон для отображения на экране не забывайте каждый раз отмечать два окна (окно **Menu** и окно, которое должно вызываться конфигурируемой кнопкой).

Раздел 6. Особенности адресация в InTouch.

Так как **InTouch** предназначен для разработки и поддержания интерфейса сбора данных и диспетчерского управления, среда исполнения **WindowViewer** при взаимодействии с контроллерным уровнем выступает, как правило, в роли **приложения - клиента** (узел **View**), запрашивающего данные у **приложения - сервера** (**I/O Server**).



Через сервер ввода/вывода **InTouch** - приложение имеет возможность читать данные из контроллера или писать данные в него. Процесс обмена информацией **InTouch** - приложения с контроллером можно представить следующей схемой.



Здесь и встает один из главных вопросов организации обмена с серверами ввода/вывода: каким образом обеспечить клиенту доступ к запрашиваемой им информации?

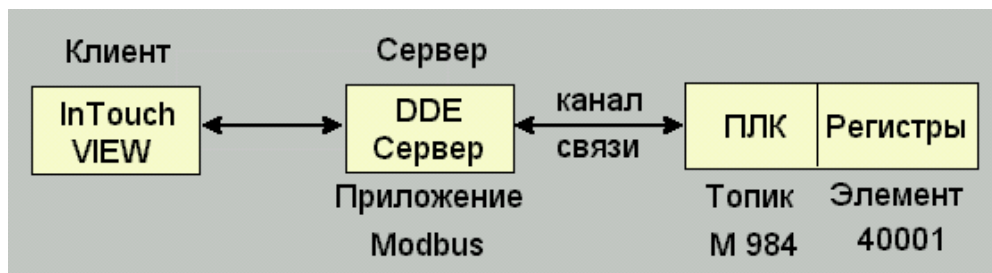
Для организации обмена с приложением определяются каналы обмена или каналы доступа, характеризующиеся следующими параметрами:

- имя узла (Node Name);
- имя приложения (Application Name);
- имя группы данных или топик (Topic Name);
- имя элемента (Item Name).

Имя приложения - это имя программы Windows, которая выполняет функции **DDE**, **FastDDE**, **SuiteLink** - серверов. Имя группы данных (топика) определяется при конфигурировании сервера на прием или передачу группы данных, которыми сервер будет обмениваться с контроллером или объединенными в сеть контроллерами. Определенные параметры группы (топика) зависят от конкретного сервера (поэтому рекомендуется изучать

документацию и справочную систему выбранного сервера). Например, при использовании **Modbus** - сервера, позволяющего обеспечить взаимодействие с контроллером **Modicon Micro 984 PLC**, в качестве имени приложения (Application Name) должен быть **Modbus**, в качестве имени группы или топика (Topic Name) вводится любое имя (текстовая строка), но среди необходимых параметров группы из списка выбирается имя контроллера **Modicon 984 PLC**. А в качестве имени элемента (Item Name) следует выбирать название конкретного регистра контроллера (например, **40001** для контроллера Modicon Micro 984). Чтобы узнать правильный синтаксис имени элемента, необходимый для конкретных **PLC**, нужно обратиться к руководству по соответствующему серверу.

Определены все компоненты коммуникационного канала. С учетом введенных понятий схема обмена информацией для рассмотренного выше примера будет выглядеть следующим образом.



Фирма Wonderware предлагает DDE и SuiteLink - серверы, которые поддерживают более 800 типов контроллеров основных производителей и различные протоколы.

Если нужного драйвера все-таки нет, можно воспользоваться пакетом разработки драйверов **FactorySuite Toolkit**.

Схемы, приведенные выше, интерпретируют стандартный обмен информацией между узлом (приложением) **View** и контроллером (ПЛК) в режиме сбора данных и управления. В этом режиме, как уже было сказано выше, приложение **View** - клиент по определению.

Обмен данными с другими приложениями.

Но приложения **InTouch** могут взаимодействовать не только между собой, но и с другими **Windows** - приложениями. Одним из известных примеров такого приложения является **Microsoft Excel**. **InTouch** - приложение может считывать и записывать какие - либо значения в любую клетку открытой в **Excel** электронной таблицы. Аналогично и программа **Excel** может читать и записывать информацию в базу данных **InTouch** - приложения. Данный механизм обеспечивает одновременное обновление

данных в одном приложении при изменении их значений в другом.

Если клиентом (приложением, запрашивающим информацию) по - прежнему является узел **View**, то **Excel** - это приложение, поставляющее информацию (сервер). В качестве группы или топика (**Topic**) тогда будет выступать имя таблицы **Excel**, а элемент обмена информацией - ячейка в таблице **Excel** (табл.1, вариант 1).

Когда клиентом является приложение **Excel**, а сервером - приложение **View**, группой в этом случае всегда является словарь переменных **InTouch** (база данных) с именем **Tagname**. Элементом обмена будет элемент базы данных - имя переменной (табл.1, вариант 2).

Таблица 1.

Приложение-клиент	Приложение-сервер	Группа	Элемент
1. View	Excel	Sheet1.XLS	R1C1
2. Excel	View	Tagname	R_Level

В случае обмена данными по сети с использованием пакета Wonderware **NetDDE** необходимо к **трехуровневой** структуре адреса добавить четвертый уровень - имя удаленного узла сети (Node Name).

Подводя итог вышесказанному, следует подчеркнуть, что информация по доступу к данным устройств ввода/вывода или других приложений должна храниться в приложении (в словаре переменных). И разработчику в **InTouch**-приложении важно подключиться к вышеописанному каналу доступа. Для этого в **InTouch** необходимо определить имя доступа **Access Name** и связать его с переменной приложения.

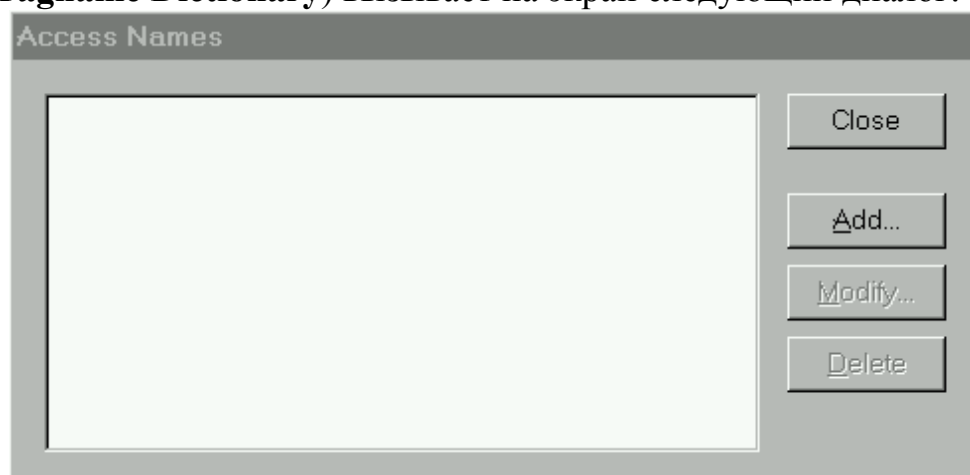
В распределенных системах **InTouch** имя доступа может быть определено либо как локальный адрес, либо как глобальный.

Локальные адреса используются в том случае, когда **View** - узлы имеют свои серверы ввода/вывода. На рисунке узлы исполнения (**View** - узлы), каждый со своей копией одного и того же приложения, ссылаются на свои собственные источники данных ввода/вывода (серверы ввода/вывода).

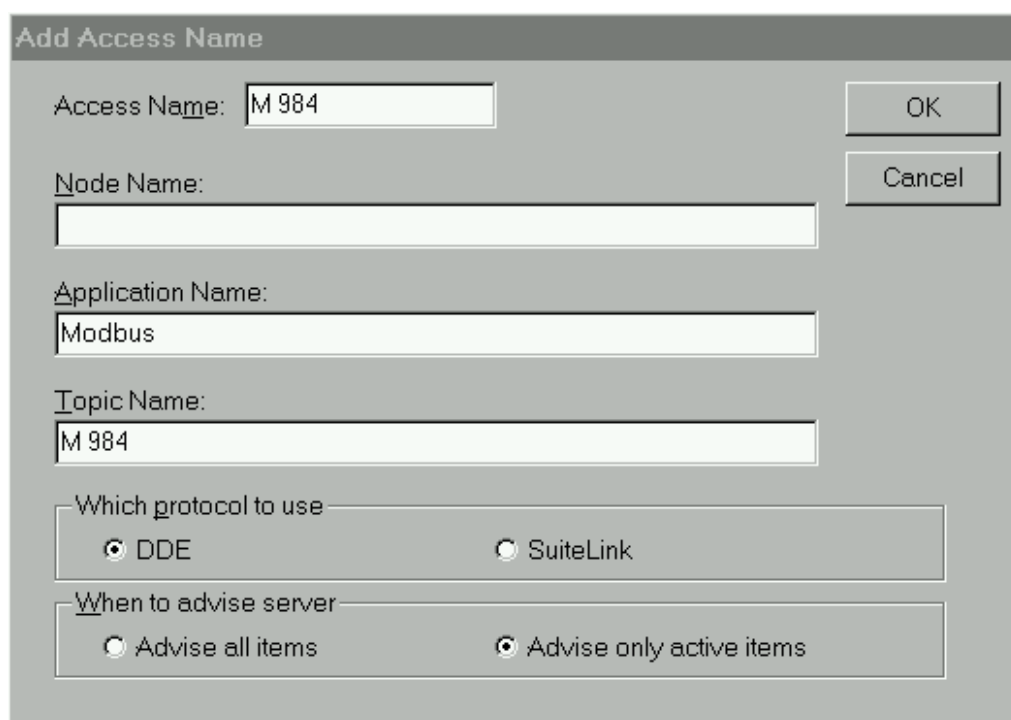


Поэтому при определении канала доступа к информации ввода/вывода достаточно трехуровневого адреса (**Application** - приложение, **Topic** - объект, **Item** - элемент). Имя узла (Node) в этом случае опускается.

Команда **Special/Access Names** (или щелчок по кнопке **Access Name** диалога **Tagname Dictionary**) вызывает на экран следующий диалог:



Этот диалог предназначен для определения нового канала доступа (кнопка **Add**), модификации существующего (**Modify**) или удаления (**Delete**). Щелчок по кнопке **Add** вызывает диалог определения нового канала доступа.



Диалог определения канала доступа заполнен в соответствии с примером, рассмотренным выше. В качестве имени (канала) доступа (**Access Names**) рекомендуется выбирать имя группы или топика (**Topic Name**).

Следует подчеркнуть, что поле **Node Name** (имя узла) оставлено пустым.

Щелчок по кнопке **Ok** возвращает пользователя в диалог **Access Names** (имена доступа) с определенным именем доступа.



Работа 6. DDE - обмен.

Цель работы: организация связи между двумя **Windows** - программами (**Excel** и **InTouch**) по протоколу **DDE**.

1. Откройте окно **Alarms** и программу **Excel**. Расположите приложения так, что бы был виден движковый регулятор в окне **Alarms**.

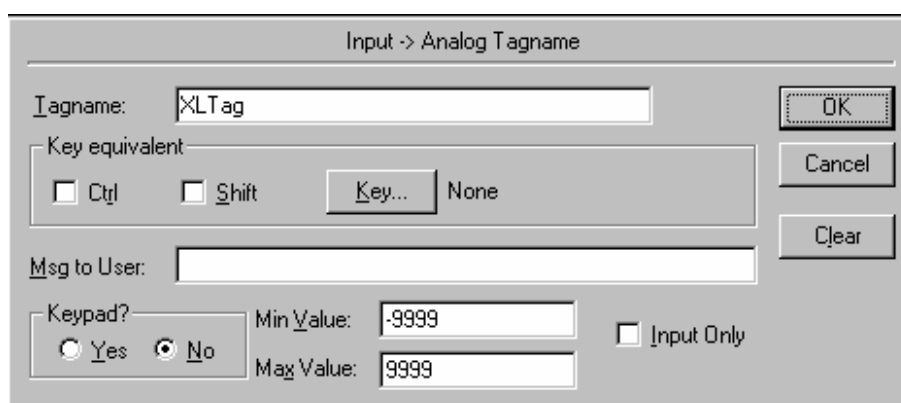
2. Выделите ячейку в таблице **Excel** и введите следующее выражение:
=view|tagname!IntTag

Перед нажатием клавиши **Enter** в **Excel** запустите **WindowViewer**.

3. Вернитесь в **Excel**, нажмите **Enter**. В выделенной ячейке появится значение переменной. Оно должно изменяться при перемещении движка регулятора в окне **WindowViewer**.

4. Теперь сконфигурируйте оба приложения как клиент и сервер. Закройте окно **Alarms** и откройте окно **Scratch** (черновик).

5. Поместите в этом окне текстовый объект **#** и задайте ему свойство **User Input/Analog**. Укажите новую переменную **XLTag** со значениями **Min Value = - 9999** и **Max Value = 9999**.



6. Нажмите **Ok**. Определите переменную **XLTag** как переменную типа **I/O Real** с **Min Eu** и **Min Raw** = – 9999, **Max EU** и **Max Raw** = 9999.

7. Нажмите кнопку **Access Name** в словаре переменных и щелкните на кнопке **Add**. Введите имя канала доступа в поле **Access Name** (для русской версии **Microsoft Office** в поле **Topic Name** следует ввести **[Книга1]Лист1**):

8. Нажмите **Ok**. Имя созданного канала доступа появится в списке доступных.

9. Нажмите **Close**. Имя канала доступа появится рядом с кнопкой **Access Name** в словаре переменных.

10. В поле **Item** укажите строку и столбец ячейки **Excel**, например **r2c2**.

11. Нажмите кнопку **Close**, затем **Ok**.
12. Перейдите в **WindowViewer** и проверьте функционирование связи.
13. Теперь сделайте это с помощью специального мастера. Для этого выберите новую ячейку и введите любое число.
14. Скопируйте ячейку в буфер (CTRL+C), перейдите в **WindowMaker** и вставьте мастер **Value Displays - I/O Analog Display** в окно **Scratch**.
15. Нажмите кнопку **Paste Link**. Поля автоматически будут заполнены информацией о выбранной ячейке **Excel**.
16. Убедитесь, что опция **Allow Input** выбрана.

I/O Analog Value Wizard

Application: OK

Topic: Cancel

Item: Paste Link

Protocol:

If possible, this wizard will use an existing I/O Real Tag.
If a Tag does not exist, however, one will be created.

☒ When creating a tag, Use Item as Tagname ☒ Allow Input

Tagname:

Value Display Format

Number Format:

Data Integrity Checking: Options...

Eng Units:

17. Нажмите **Ok** и переключитесь в **WindowViewer**.

Раздел 7. Пример разработки проекта.

Задание на проектирование.

Требуется создать операторский интерфейс для управления технологическим процессом обезвоживания нефтяной эмульсии (рис.1).

Автоматизация процесса обезвоживания предполагает:

- контроль уровня раздела фаз в отстойнике;
- контроль расхода обезвоженной нефти;
- сигнализацию предельных отклонений верхнего и нижнего уровней раздела фаз в отстойнике;
- автоматическое регулирование уровня раздела фаз в отстойнике.

Кроме этого, операторский интерфейс должен обеспечить оператору возможность перехода с автоматического режима работы контура регулирования уровня на ручной режим (дистанционное управление клапаном).

Второе окно интерфейса - тренды для отображения в реальном времени уровня раздела фаз и расхода обезвоженной нефти.

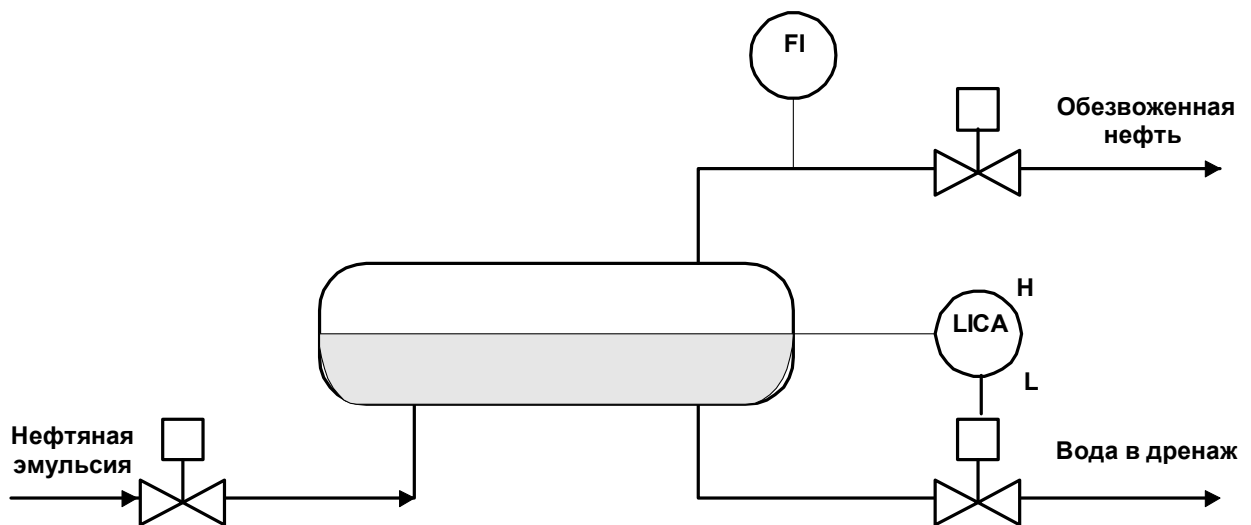


Рис.2. Функциональная схема автоматизации.

Проект (анимация) должен запускаться и останавливаться с помощью кнопки, действие которой связано со сценарием проекта.

Разработка проекта.

1. Запустите **InTouch Application Manager**. Создайте новое приложение и запустите среду разработки приложений **WindowMaker**. Создайте первое окно приложения. Для этого вызовите команду **File/ New Window...** В появившееся диалоговое окно **Window Properties** введите имя окна (до 32 символов и пробелов). Выберите цвет для фона окна, щелкнув по окошечку **Window Color**. При появлении цветовой палитры выберите цвет, щелкнув по нему. Включите опции **Title Bar**, чтобы при загрузке окна его имя появлялось на титульной строке, и **Size Controls** для возможности изменения размеров окна. Оставшиеся опции можно оставить со значениями по умолчанию. Закончив выбор, нажмите **Ok**. На экране появится окно с заданными параметрами.

Выберите на панели инструментов **Draw** инструмент **Rounded Rectangle** (скругленный прямоугольник), щелкнув по нему. На свободном месте окна, нажав и удерживая левую кнопку мыши, протащите ее по диагонали, чтобы сформировать горизонтальную емкость. Если теперь отпустить левую кнопку, объект будет готов и выбран. Теперь можно изменить форму объекта, если потащить за любой из маленьких квадратиков вдоль границ объекта. Для увеличения/уменьшения радиуса скругления углов емкости отключите опцию клавиатуры **Num Lock** и, удерживая клавишу **Shift**, нажмите на клавишу **+/-**. Чтобы переместить объект, просто потащите его, зацепившись где-нибудь внутри объекта. Для выбора цветовых атрибутов объекта выделите его и щелкните по соответствующему инструменту управления цветом.

Теперь нужно создать клапан. Для удобства рисования установите сетку (команда **Arrange/Snap to Grid**). Затем, используя средство **Polygon** (многоугольник - панель **Draw**), нарисуйте равносторонний треугольник, одна из сторон которого параллельна вертикальным линиям сетки. Дублируйте нарисованный треугольник (нажмите кнопку **Duplicate** панели инструментов **General**). Выделите второй треугольник и нажмите кнопку **Flip Horizontal** (отобразить зеркально по вертикали - панель **Arrange**). Соедините два треугольника, используя клавиши стрелок и мышь. Для перемещения объекта предварительно выделите его. Добавьте шток клапана с помощью инструмента **Rectangle** (прямоугольник - панель **Draw**). Расположите шток под клапаном. Для этого выделите шток и нажмите кнопку **Send to Back** (поместить под - панель **Arrange**). Нарисуйте привод клапана с помощью инструмента **Ellipse**.

Клапан готов. Но созданный клапан состоит из четырех отдельных элементов. Чтобы сделать из них единый объект, нажмите клавишу **F2** для выделения всех объектов в окне, а затем нажмите кнопку **Make Symbol** (объединить в символ - панель **Arrange**). Теперь все составляющие клапан элементы объединены в *символ* и могут обрабатываться как *единый объект*. Можно изменять размер клапана, выделив его и потянув за один из захватов. Можно переместить клапан в новое место. У созданного объекта можно изменить цвет заполнения, линий, цвет рамки.

Для мнемосхемы технологического процесса требуется два клапана. Чтобы не рисовать следующий клапан, снова воспользуйтесь инструментом **Duplicate** и создайте необходимое количество клапанов.

Подпишите к емкости участки трубопровода для подачи в нее эмульсии и вывода обезвоженной нефти и дренажной воды. Для этого воспользуйтесь библиотекой мастер-объектов **Wizard** (группа **Symbol Factory**) или простыми объектами (**H/V Line - гор./верт. линия**). Толщину линии можно выбрать, воспользовавшись меню **Line**. В разрывы трубопровода вставьте предварительно нарисованные клапаны.

Если элементы трубопровода и клапаны нарисованы отдельно и не соединены с емкостью, соедините их. Отдельные объекты можно либо протащить к новому месту, либо выбрать и стрелками точно позиционировать на место (с точностью до пикселя). Удерживая клавишу **Shift** во время использования стрелок для перемещения, можно сдвигать объект на 10 пикселей одним нажатием на клавишу, а удерживая **Ctrl** - на 50 пикселей.

Далее в соответствии с заданием нужно создать числовые поля, на которые будут выводиться текущий уровень раздела фаз в отстойнике и расход обезвоженной нефти. Удобнее было бы эти поля поместить в рамки, подкрашенные соответствующим цветом. Воспользовавшись инструментом **Rectangle** (прямоугольник - панель **Draw**), нарисуйте рамку и присвойте ей желаемые цветовые атрибуты. А так как речь идет о двух параметрах, произведите дублирование рамки. Переместите созданные рамки на места, удобные для наблюдения за процессом (возможно, удобнее их поместить в верхней части окна).

Теперь выберите инструмент **T** (текст), которым нужно набрать названия параметров и знаки числового поля для вывода значений переменных на экран. Предварительно подберите шрифт, его стиль и размер, выбрав команду **Text/Fonts**. На свободном месте окна напечатайте один знак

решетки #, нажмите **Enter**. Нажатие этой клавиши завершает ввод первой текстовой строки и позволяет создать новую текстовую строку. Введите размерность параметра и нажмите **Enter**. Наконец, введите название параметра. Теперь нужно сдвинуть эти строки на свои места в одну из рамок. Повторите все проделанное выше для второго параметра.

Для сигнализации предельных отклонений верхнего и нижнего уровней раздела фаз предлагается создать в окне еще две рамки с соответствующими текстовыми полями (только текстовыми полями, т. к. выводить значение уровня в эти рамки нет необходимости). В дальнейшем, при анимации объектов окна, можно будет “заставить” эти рамки мерцать красным цветом при выходе уровня раздела фаз за установленные границы.

Предполагается, что уровень раздела фаз стабилизируется автоматическим регулятором, регулирующее воздействие которого направлено на клапан, установленный на линии сброса дренажной воды. Для перехода с автоматического управления этим клапаном на ручное (дистанционное) введите в схему переключатель **руч./авт.** Щелкните по иконке **Wizard** и в открывшемся меню выберите группу **Switches** (переключатели). Нажмите мышью на переключателе (**Knob Switch**) и щелкните по кнопке **Select** (или дважды щелкните по переключателю). Диалоговая панель автоматически закроется и вернется текущее окно. Курсор изменится на уголок при возвращении в текущее окно. Щелкните мышью для вставки объекта в окно. Дважды щелкните по мастер-объекту, чтобы произвести его конфигурирование (в поле **Tagname** введите имя объекта, например, **Status**, в позицию **Switch Label** - **управление** (control), в позицию **ON Label** - **руч.** (hand), в позицию **OFF Label** - **авт.** (avt)). Так как переменная **Status** еще не определена в словаре переменных, система выдаст запрос на ее определение. Подтвердите запрос и определите переменную **Status** (тип переменной - **Memory Discrete**).

Теперь нужно предоставить оператору возможность дистанционно управлять клапаном. Для этого вновь выберите инструмент **Wizard** и в открывшемся меню щелкните по строке **Sliders** (движковые регуляторы). Вставьте выбранный мастер-объект в окно и при его конфигурировании привяжите к переменной **valve** типа **Memory Integer**.

Но эти два «рычага» управления будут постоянно находиться в окне мнемосхемы, тогда как пользоваться ими оператору приходится редко. Если же в окне представлено несколько контуров регулирования, то экран будет «перегружен» объектами и оператору будет сложно в них ориентироваться.

Создайте новое окно типа **Popup** (всплывающее) без заголовка и без возможности изменения размеров окна. Размеры и положение окна подберите таким образом, чтобы в нем разместились переключатель, движковый регулятор и кнопка для закрытия окна (при вызове окна оно не должно закрывать другие элементы интерфейса). Вырежьте поочередно переключатель и движковый регулятор из первого окна и вставьте в новое (всплывающее) окно (команда **Paste** меню правой кнопки). При этом параметры конфигурирования этих объектов сохраняются. Подпишите в новом окне кнопку с текстом **Заккрыть** (для замены текста на кнопке выполните команду **Special/Substitute String**).

Для вызова на экран всплывающего окна в основном окне также создайте кнопку (**Управление**).

Наконец, нужно создать кнопку, нажатие на которую будет запускать и останавливать процесс (с текстом **Пуск**).

Графическое проектирование операторского интерфейса закончено. Но нужно решить еще три задачи, чтобы завершить проект: анимировать объекты, создать базу данных и написать сценарий проекта.

2. Анимацию объектов и создание базы данных переменных проекта можно выполнять раздельно, хотя удобнее их проводить одновременно для выбранного объекта.

Начните с отстойника. Щелчком выберите его, а затем щелкните по инструменту **Fill** (цвет заполнения) и выберите цвет для заполнения отстойника (для отстоявшейся воды). Далее выберите отстойник и нажмите клавишу **Enter** (или дважды щелкните по объекту). В появившемся диалоговом окне выбора динамических связей отметьте опцию **Percent Fill/Vertical** (процентное заполнение по вертикали) и щелкните по кнопке **Vertical**. В поле **Expression** введите имя описываемой переменной (например, **ots_level**). В графу **Value at Max Fill** (максимум заполнения) введите 3000 (диаметр отстойника в мм), минимум заполнения будет равен 0. В графы **Max%** и **Min%** введите, соответственно, 100 и 0.

Направление заполнения - вверх, отметьте опцию **Up**, после чего выберите цвет фона (**Background Color**). Напомним, что цвет заполнения был выбран ранее, здесь же речь идет о цвете верхней части отстойника (в данном примере в верхней части отстойника находится нефть). Чтобы закончить с отстойником, щелкните по **Ok**. Если имя переменной используется впервые, появится диалоговое окно для определения переменной (**Tagname Dictionary**).

Обратите внимание, что поле для имени переменной уже заполнено. Щелчком по кнопке **Type** выберите тип переменной - **Memory Real** (проект разрабатывается в отсутствие реальных контроллеров, его анимация будет осуществлена с помощью скрипта). В поле **Max Value** введите 3000, в поле **Eng Units** (единицы измерения) - мм.

Если опция **Alarms** не включена, включите ее. Отметьте опцию **Low** и в появившиеся поля внесите значение нижнего уровня раздела фаз (например, 300), при котором будет включаться сигнализация, и уровень приоритета. То же самое сделайте для опции **High** (значение верхнего уровня для сигнализации - 2000). Щелкните по клавише **Close**. Переменная (уровень раздела фаз) анимирована и ее имя занесено в словарь переменных.

Чтобы анимировать объект # (решетка), выберите его и нажмите **Enter** (дважды щелкните по нему). Отметьте опцию **Value Display/Analog** (показать значение / аналоговое) и щелкните по клавише **Analog** для выбора связи. Дважды щелкните по пустому полю **Expression**. В появившемся списке переменных выберите требуемую переменную (уровень раздела фаз **ots_level** - имя этой переменной уже находится в базе данных) и дважды щелкните по ней. Ее имя появится в поле **Expression**. Далее щелчок по **Ok**, затем еще раз **Ok**.

Проделайте вышеописанную операцию для анимации всех числовых полей проекта (переменной, отображающей расход нефти, присвойте имя **debitn**, тип - **Memory Real**).

Если имя переменной используется впервые (его еще нет в базе данных), введите его в поле **Expression** с клавиатуры.

Для анимации табло сигнализации отклонения уровня раздела фаз за верхний предел дважды щелкните по ранее созданному табло. В диалоговом окне описания связей включите опцию **Miscellaneous/Blink** и щелкните по клавише **Blink**. В поле **Expression** введите имя переменной **ots_level** (либо наберите на клавиатуре, либо воспользуйтесь приемом, описанным выше), поставьте точку и без пробела введите **HiStatus** (верхний предел). Отметьте опцию **Blink visible** (видимое мерцание) и выберите цвет мерцания табло.

Те же операции сделайте для анимации табло сигнализации отклонения уровня раздела фаз за нижний предел. В поле **Expression** после точки за именем переменной введите **LoStatus** (нижний предел).

Кнопке **Управление** присвойте динамическое свойство **Touch Pushbutton/Show Window** - показать окно. В появившемся диалоге отметьте имя окна, которое будет открываться при нажатии кнопки **Управление**.

Во втором окне операторского интерфейса - переключатель, предоставляющий оператору возможность перевести контур на ручное управление. Ранее, при конфигурировании этого объекта, его имя было занесено в базу данных (словарь переменных). Объекты, созданные инструментом **Wizard** и сконфигурированные с помощью специального диалогового окна, не требуют дополнительной установки анимационных связей.

То же самое можно сказать и о другом элементе операторского интерфейса - движковом регуляторе. Нужно добавить лишь, что с помощью этого элемента оператор может воздействовать на исполнительное устройство, изменяя положение его штока от 0 до 100%.

Чтобы закончить с всплывающим окном, присвойте кнопке **Заккрыть** динамическое свойство **Touch Pushbutton/Hide Window** - спрятать окно. В появившемся диалоге отметьте имя окна, которое будет закрываться при нажатии кнопки **Заккрыть**.

Прежде чем описывать анимацию кнопки **Пуск** основного окна, следует пояснить ее назначение в этом проекте. Данный проект разрабатывается в условиях отсутствия реального технологического процесса, на объектах которого размещены датчики, исполнительные устройства. Нет реального контроллера с регистрами, устройствами ввода/ вывода, с помощью которых информация считывается с датчиков или передается на исполнительные устройства, нет, наконец, **DDE** драйвера - промежуточного звена между пакетом **InTouch** и контроллером. Поэтому тип всех переменных в данном проекте - **Memory** (для реальных переменных тип **I/O**). С другой стороны, интерфейс оператора должен быть “живым”, т.е. уровень раздела фаз в отстойнике должен изменяться, а для этого должен изменяться расход дренажной воды через клапан и т. д.

Практически любые по сложности операторские интерфейсы могут быть созданы с помощью скриптов.

Существует несколько возможностей анимировать проект с помощью скриптов. Анимация проекта, разработанного в данном примере, базируется на сценарии действий при нажатии кнопки. И этой кнопкой является кнопка **Пуск**.

Дважды щелкните по кнопке **Пуск**. Присвойте ей динамическое свойство **Touch Pushbutton/Discrete Value**. В поле **Tagname** введите имя объекта (например, **Start**), включите опцию **Toggle** (переключать). Щелкните по **Ok**. При определении переменной **Start** выберите тип **Memory Discrete**).

Щелчок по кнопке **Close**, затем **Ok** и еще раз **Ok**. Анимационная связь для кнопки установлена.

Предполагается, что кнопка предназначена не только для запуска проекта, но и для его остановки. Первоначальный текст на кнопке - **Пуск**, но после запуска проекта он должен измениться на **Стоп**. Для задания кнопке этого свойства дважды щелкните по ней и установите динамическое свойство **Value Display/Discrete**. В поле **Expression** введите имя переменной - **Start**, в поле **On Message** - **Стоп**, а в поле **Off Message** - **Пуск**.

3. Для создания скрипта выполните команду **Special/Scripts/ Application Scripts**. В поле **Condition Type** выберите тип скрипта - **While Running** (исполняется непрерывно пока запущено приложение), в поле **Every** - 50 мсек. На белом поле диалогового окна наберите следующую ниже программу (интерпретация программы предлагается читателю):

IF Start THEN

IF NOT Status THEN

ots_level = 1000 + 100 * Cos(\$Second * 10);

valve = ots_level / 30;

ELSE

ots_level = valve * 30 + 50 * Cos(\$Second);

ENDIF;

debitn = 300 - ots_level / 10;

ENDIF;

Щелчок по **Ok**.

Проверьте работу проекта в режиме просмотра **WindowViewer**.

В правом верхнем углу экрана, непосредственно за командами меню, виден текст **Runtime** (выполнение). Чтобы быстро переключиться в этот режим (**WindowViewer**), щелкните по этому тексту. (В режим выполнения можно войти, выбрав команду **File/ WindowViewer**. Для возвращения в режим разработки следует ввести команду **File/WindowMaker**). Из предложенного списка окон приложения выберите окна для просмотра в режиме **Runtime** и нажмите **Ok**).

Чтобы запустить проект, щелкните по кнопке **Пуск**.

Если какие-либо компоненты интерфейса «не работают», исправьте ошибки и добейтесь работы проекта в полном объеме.

Создайте третье окно проекта с именем **Тренды**. Воспользовавшись инструментом **Real-time Trend**, разместите в нем тренд реального времени. Двойным щелчком по тренду вызовите диалог для его конфигурирования.

Диапазон времени, охватываемый трендом на экране (**Time Span**), установите 10 мин. Подберите частоту вывода на график значений переменных, например, 1 секунда (**Sample**). В поле **Max Value** введите 2000 (такой масштаб позволит просматривать все переменные).

Подберите цвета сетки, цифр по осям тренда таким образом, чтобы они были хорошо видны на экране. Произведите конфигурирование перьев для отображения уровня раздела фаз в отстойнике (**ots_level**), расхода обезвоженной нефти (**debitn**), положения штока регулирующего клапана (**valve**).

Для обеспечения навигации по окнам проекта в окне мнемосхемы и окне трендов создайте кнопки, нажатие на которые позволит открывать соответствующие окна (со свойством **Touch Pushbutton/Show Window** - показать окно).

Сохраните проект и перейдите в режим просмотра. Убедитесь, что все компоненты интерфейса работают в соответствии с заданием.

СОДЕРЖАНИЕ

Введение.	3
Раздел 1. Application Manager и WindowMaker.	4
<u>Работа 1.</u> Окна и графические объекты.	8
Раздел 2. Анимационные связи.	15
<u>Работа 2.</u> Создание анимационных связей.	16
Раздел 3. Скрипты.	24
<u>Работа 3.</u> Использование скриптов в InTouch .	27
Раздел 4. Алармы и события в InTouch .	33
<u>Работа 4.</u> Алармы и события.	36
Раздел 5. Тренды в InTouch .	41
<u>Работа 5.</u> Тренды реального времени и архивные тренды.	45
Раздел 6. Особенности адресация в InTouch .	53
<u>Работа 6.</u> DDE - обмен.	57
Раздел 7. Пример разработки проекта.	60

Андреев Евгений Борисович, доцент кафедры АТП
РГУ нефти и газа им. И.М.Губкина.

**Проектирование систем управления в SCADA-пакете
InTouch 7.0**

Подписано в печать 10.09.2005 г.

Формат 60х90/16

Объем 4,0 уч.-изд. л.

Тираж 100 экз.

Заказ №

Отдел оперативной полиграфии РГУ нефти и газа им. И.М.Губкина
117917, Москва, Ленинский просп., 65.