

Министерство образования и науки Российской Федерации
Нижекамский химико-технологический институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего профессионального образования
«Казанский национальный исследовательский технологический университет»

А. В. Долганов, Г.Б. Минигалиев, В.В. Елизаров

ИНТЕГРИРОВАННЫЕ СИСТЕМЫ ПРОЕКТИРОВАНИЯ И УПРАВЛЕНИЯ

ПРАКТИКУМ

**Нижекамск
2014**

УДК 681.5

Д 64

Печатается по решению методического совета Нижнекамского химико-технологического института (филиал) ФГБОУ ВПО «КНИТУ»

Рецензенты:

Саримов Н.Н., кандидат физико-математических наук;

Кириллов Д.А., кандидат технических наук,
инженер АСУТП I кат. ОАО «Танеко».

Долганов, А.В.

Д 64 Интегрированные системы проектирования и управления : практикум / А.В. Долганов, Г.Б. Минигалиев, В.В. Елизаров. – Нижнекамск : Нижнекамский химико-технологический институт (филиал) ФГБОУ ВПО «КНИТУ», 2014. – 124с.

Учебное пособие может стать основой для изучения InTouch – самого популярного в мире HMI-пакета мощных и гибких средств разработки операторских интерфейсов для создания АСУТП дискретных и непрерывных производств, распределенных систем управления, диспетчерского управления и сбора данных (SCADA), а также других областей промышленного применения.

Учебное пособие предназначено для специалистов в области автоматизации производства и студентам старших курсов высших технических учебных заведений, обучающихся по направлениям бакалаврской подготовки 220700 «Автоматизация технологических процессов и производств» и 220400 «Управление в технических системах».

УДК 681.5

© Долганов А.В., Минигалиев Г.Б., Елизаров В.В., 2014

© Нижнекамский химико-технологический институт
(филиал) ФГБОУ ВПО «КНИТУ», 2014

Содержание

Введение	4
1. Требования к современным автоматизированным системам управления производством.....	5
2. Типы взаимодействия с контроллерами.....	10
3. Аппаратная реализация связи с устройствами ввода/вывода.....	13
4. Программные средства для операторских станций в системе автоматизации управления производством (SCADA – системы).....	14
5. Характеристики SCADA – систем.....	16
6. Выбор SCADA – программы для конкретной системы автоматизации производства.....	18
7. SCADA система InTouch фирмы Wonderware.....	21
8. Лабораторные работы по InTouch.....	32
Лабораторная работа №1. Окна InTouch.....	32
Лабораторная работа №2. Объекты InTouch.....	35
Лабораторная работа №3. Теги и анимационные связи.....	40
Лабораторная работа №4. Скрипты в InTouch.....	71
Лабораторная работа №5. Алармы и события.....	79
Лабораторная работа №6. Тренды архива и реального времени.....	92
Лабораторная работа №7. Обмен данными с MS Excel.....	103
Лабораторная работа №8. Мастера группы Windows Controls.....	107
Лабораторная работа №9. Средства безопасности.....	113
Лабораторная работа №10. Рецепты и SQL-запросы.....	117
Литература.....	123

Введение

Сегодня трудно кого-нибудь удивить книгой по «компьютерной тематике». И это понятно, так как человечество уже не может существовать без компьютера. Тем не менее, существуют такие сферы человеческой деятельности и такие программные продукты, которые совершенно незаслуженно обделены вниманием и авторов, и книгоиздателей. Речь идет о человеке-операторе, контролирующем и управляющем какими-либо техническими процессами на производстве, в коммунальном хозяйстве, на транспорте и, соответственно, о программном обеспечении, предназначенном для создания интерфейса между оператором и контролируемым процессом.

Существует множество программных продуктов, помогающих реализовать человеко-машинный интерфейс (HMI, Human Machine Interface), но самым популярным является InTouch, разработанный корпорацией Wonderware. Сообщество людей во всем мире, использующих этот продукт в своей повседневной деятельности, насчитывает около миллиона человек. Большинство из них – операторы, которые используют InTouch, даже не подозревая этого. Но для многих людей, тех, кто занимается разработкой HMI, InTouch – повседневный инструмент.

Это учебное пособие предназначено, в первую очередь, для читателя, который имеет возможность и желание изучить один из самых популярных в мире человеко-машинных интерфейсов Wonderware InTouch либо под руководством преподавателя, либо самостоятельно, но обязательно путем кропотливой практической работы.

Необходимо подчеркнуть, что предлагаемое учебное пособие нельзя рассматривать как некоторое дополнение к разнообразной документации по InTouch. Основная цель данного пособия: дать возможность читателю получить представление об InTouch как о некотором целостном механизме для решения задач операторского взаимодействия с технологическим процессом с помощью его графического представления. Поэтому задания составлены таким образом, чтобы в итоге получилось некоторое работоспособное учебное приложение, предназначенное для изучения разнообразных возможностей InTouch на примере конкретного технологического процесса.

1. Требования к современным автоматизированным системам управления производством

В современном промышленном производстве все большее значение приобретает возможность оперативного доступа к достоверной и точной информации из любой точки управления производством, поскольку это определяющим образом влияет на эффективность работы предприятия, включая производительность труда, качество и конкурентоспособность выпускаемой продукции.

Эта задача решается путем создания интегрированной многоуровневой распределенной автоматизированной системы управления (АСУ). Интегрированная система автоматизации предприятия может быть представлена в виде 5-уровневой пирамиды (рис.1).

Первый уровень, уровень ввода-вывода (I/O), включает набор датчиков и исполнительных устройств, встраиваемых в конструктивные узлы технологического оборудования и предназначенных для сбора первичной информации и реализации исполнительных воздействий.

Современные датчики из однофункциональных средств определения текущих значений измеряемых величин постепенно превращаются в многофункциональные средства автоматизации, которые решают еще целый ряд задач по диагностике, преобразованию измерительной информации, выполнению простых алгоритмов управления и т. д. Такая многофункциональность стала возможна после оснащения датчиков встроенным микропроцессором.

Современные интеллектуальные датчики выполняют, кроме процесса измерения, преобразования измеряемых сигналов в типовые аналоговые и цифровые значения, самодиагностику своей работы, дистанционную настройку диапазона измерения, первичную обработку измерительной информации, иногда еще ряд достаточно простых, типовых алгоритмов контроля и управления. Они имеют интерфейсы к стандартным/типовым полевым цифровым сетям, что делает их совместимыми с практически любыми современными средствами автоматизации, и позволяет информационно общаться с этими средствами и получать питание от блоков питания этих средств.

Второй уровень, непосредственное управление, служит для непосредственного автоматического управления технологическими процессами с помощью промышленных контроллеров и характеризуется следующими показателями:

- предельно высокой реактивностью режимов реального времени;
- предельной надежностью (на уровне надежности основного оборудования);
- возможностью встраивания в основное оборудование;
- функциональной полнотой модулей УСО;
- возможностью автономной работы при отказах комплексов управления верхних уровней;
- возможностью функционирования в цеховых условиях.

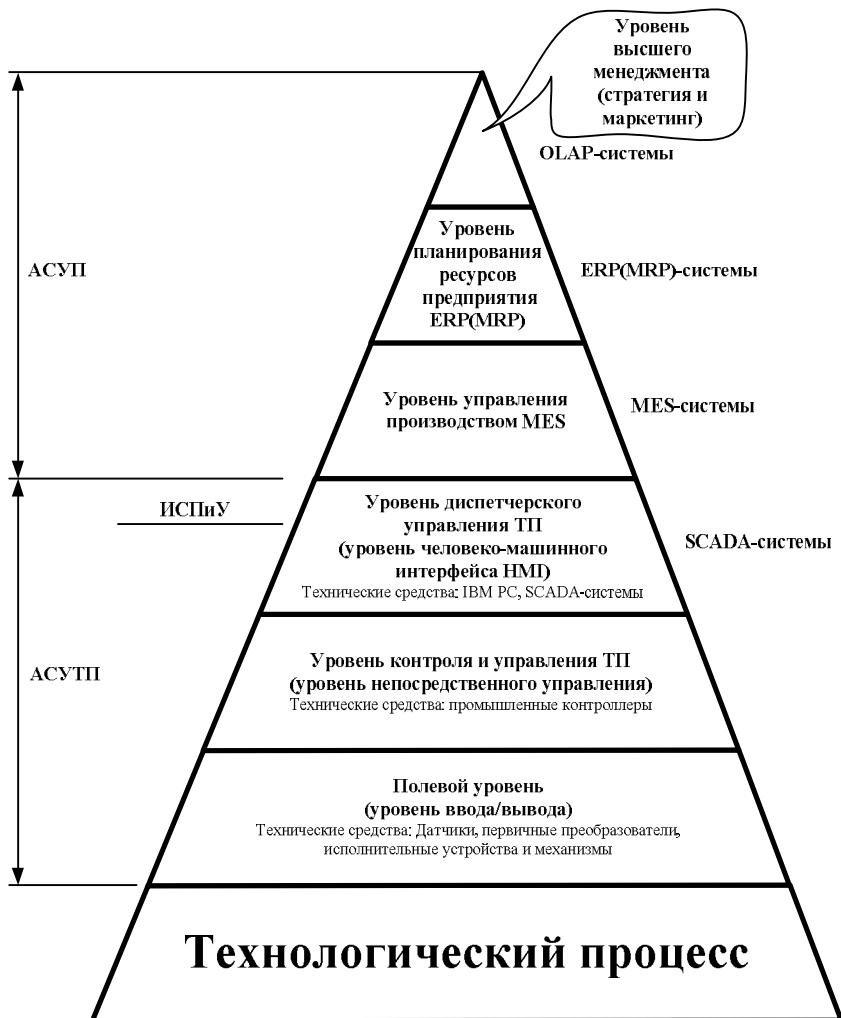


Рис.1. Уровни интегрированной системы автоматизации производства

В промышленные контроллеры загружаются программы и данные из ЭВМ третьего уровня, уставки, обеспечивающие координацию и управление агрегатом по критериям оптимальности управления технологическим процессом в целом, выполняется вывод на третий уровень управления служебной, диагностической и оперативной информации, т. е. данных о состоянии агрегата, технологического процесса.

Этот уровень управления реализуется, например, на промышленных контроллерах Arcaps, DeltaV, Centum, Simatic и др.

Третий уровень, SCADA-уровень (Supervisory Control and Data Acquisition - сбор данных и диспетчерское управление), предназначен для отображения (или визуализации) данных в производственном процессе и оперативного комплексного управления различными агрегатами, в том числе и с участием диспетчерского персонала.

Этот уровень управления должен обеспечивать:

- диспетчерское наблюдение за технологическим процессом по его графическому отображению на экране в реальном масштабе времени;

- расчет и выбор законов управления, настроек и уставок, соответствующих заданным показателям качества управления и текущим (или прогнозным) параметрам объекта управления;

- оперативное сопровождение моделей объектов управления типа «агрегат», «технологический процесс», корректировку моделей по результатам обработки информации от второго уровня;

- синхронизацию и устойчивую работу систем типа «агрегат» для группового управления технологическим оборудованием;

- ведение единой базы данных технологического процесса;

- связь с третьим уровнем.

Отвечая этим требованиям, ЭВМ на третьем уровне управления должны иметь достаточно высокую производительность как при решении задач в реальном масштабе времени, так и при обработке графической информации, обеспечивая работу в реальном времени с базами данных среднего объема и с расширенным набором интеллектуальных видеотерминалов.

Третий уровень управления реализуется на базе специализированных промышленных компьютеров, или в ряде случаев на базе персонального компьютера. Диспетчерский интерфейс реализуется SCADA-системами, например InTouch, iFix, Genesis32, WinCC и др.

Машины третьего уровня должны объединяться в однородную локальную сеть предприятия (типа Ethernet) с выходом на четвертый уровень управления.

Четвертый уровень, уровень управления производством MES (Manufacturing Execution System) - средства управления производством – характеризуется необходимостью решения задач оперативной упорядоченной обработки первичной информации из цеха и передачи этой информации на верхний уровень планирования ресурсов предприятия. Решение этих задач на данном уровне управления обеспечивает оптимизацию управления ресурсами цеха как единого организационно-

технологического объекта по заданиям, поступающим с верхнего уровня, и при оперативном учете текущих параметров, определяющих состояние объекта управления. Решение этих задач возлагается обычно на серверы в локальных сетях предприятия.

Пятый уровень, верхний уровень управления, определяется как MRP (Manufacturing Resource Planning) и ERP (Enterprise Resource Planning) – планирование ресурсов предприятия.

Задачи, решаемые на этом уровне, в аспекте требований, предъявляемых к ЭВМ, отличаются, главным образом, повышенными требованиями к ресурсам (например, для ведения единой интегрированной - централизованной или распределенной, однородной или неоднородной - базы данных, планирования и диспетчирования на уровне предприятия в целом, автоматизации обработки информации в основных и вспомогательных административно-хозяйственных подразделениях предприятия: бухгалтерский учет, материально-техническое снабжение и т.п.). Обычно для решения задач данного уровня выбирают универсальные ЭВМ, а также многопроцессорные системы повышенной производительности.

Наиболее известные системы этого уровня предлагаются компаниями SAP, Oracle, BAAN и др.

Шестой уровень, уровень высшего менеджмента (OLAP-системы – On-Line Analytical Processing – оперативный анализ данных). Информационные системы масштаба предприятия, как правило, содержат приложения, предназначенные для комплексного многомерного анализа данных, их динамики, тенденций и т.п. Такой анализ в конечном итоге призван содействовать принятию решений. Нередко эти системы так и называются — системы поддержки принятия решений.

Системы поддержки принятия решений обычно обладают средствами предоставления пользователю агрегатных данных для различных выборок из исходного набора в удобном для восприятия и анализа виде. Как правило, такие агрегатные функции образуют многомерный (и, следовательно, нереляционный) набор данных (нередко называемый гиперкубом или метакубом), оси которого содержат параметры, а ячейки — зависящие от них агрегатные данные). Вдоль каждой оси данные могут быть организованы в виде иерархии, представляющей различные уровни их детализации. Благодаря такой модели данных пользователи могут формулировать сложные запросы, генерировать отчеты, получать подмножества данных.

Этот уровень управления должен обеспечивать следующие требования к приложениям для многомерного анализа:

- предоставление пользователю результатов анализа за приемлемое время (обычно не более 5 с), пусть даже ценой менее детального анализа;
- возможность осуществления любого логического и статистического анализа, характерного для данного приложения, и его сохранения в доступном для конечного пользователя виде;
- многопользовательский доступ к данным с поддержкой соответствующих механизмов блокировок и средств авторизованного доступа;
- многомерное концептуальное представление данных, включая полную поддержку для иерархий и множественных иерархий (это — ключевое требование OLAP);
- возможность обращаться к любой нужной информации независимо от ее объема и места хранения.

Следует отметить, что OLAP-функциональность может быть реализована различными способами, начиная с простейших средств анализа данных в офисных приложениях и заканчивая распределенными аналитическими системами, основанными на серверных продуктах.

Источником в OLAP-системах является сервер, поставляющий данные для анализа.

Наиболее известные системы этого уровня предлагаются компаниями Oracle, Arbor, MicroStrategy, Hyperion, Comshare и др.

Исторически сложилось так, что верхний уровень (АСУП) и нижние уровни (АСУТП) развивались независимо друг от друга и фактически отсутствовал достаточно интеллектуальный интерфейс, который бы их объединял. Это обстоятельство на современном уровне развития промышленности стало тормозящим фактором. Для эффективной работы производственного предприятия и для принятия на верхнем уровне как стратегических, так и тактических решений требуется интеграция всех систем управления производством.

Возможности систем управления производством во многом определяются составом и функциями комплекса инструментальных программных средств, предназначенного для построения автоматизированных систем управления технологическими процессами и для интеграции их как с системами управления производством верхнего уровня, так и со средствами управления нижнего уровня (датчики, исполнительные механизмы и др.). Использование такого инструментария обеспечивает возможность создания интегрированных сквозных систем управления производством в реальном масштабе времени.

Важной причиной появления на рынке инструментальных систем для решения задач комплексной автоматизации является низкая эф-

фективность традиционного и необходимость структурированного подхода к построению интегрированных систем управления производством.

Недостатки традиционного построения АСУТП:

- множество интерфейсов, сложность и запутанность связей между объектами;
- несовместимость форматов данных и структуры сообщений;
- сложность внесения изменений, что может вызвать переработку большего объема программ.

Преимущества структурированного подхода:

- нормализация данных;
- стандартные формы сообщений;
- гибкие средства интеграции приложений, включая АСУП.

Такой модульный структурированный подход к построению АСУТП обеспечивает возможность эффективной модернизации системы, облегчает внесение в нее изменений, что в совокупности гарантирует защиту ранее вложенных инвестиций и уменьшение стоимости управления.

2. Типы взаимодействия с контроллерами

Центральное звено систем автоматизации - микропроцессорный контроллер - объединяет под этим названием ряд классов и типов универсальных микропроцессорных средств, которые удовлетворяют запросам разных категорий заказчиков.

По мощности, косвенно характеризуемой числом обслуживаемых входов/выходов, контроллеры подразделяются на следующие классы:

- класс самых малых контроллеров (десятки входов/выходов);
- класс малых контроллеров (сотни входов/выходов);
- класс больших контроллеров (тысячи входов/выходов).

По типу взаимодействия контроллеров со следующим уровнем, SCADA-системы, можно выделить несколько разновидностей.

1. *Закрытые системы*, распределенные системы управления (PCY), характеризуется тем, что в состав каждого комплекса технических средств включается специально разработанное программное обеспечение, которое не может быть применено в микропроцессорных системах других фирм. Это программное обеспечение включает в себя средства для программирования контроллера, средства визуализации станций оператора и собственные протоколы взаимодействия системы управления с контроллером (рис.2).

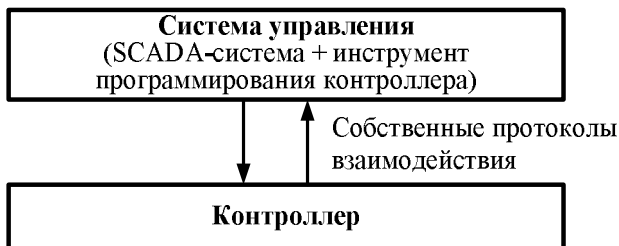


Рис. 2. Схема распределенной системы управления.

Для получения данных в SCADA-системе от контроллера, используются собственные протоколы фирм-производителей систем управления, которые реально обеспечивают скоростной обмен данными.

К этому типу взаимодействия можно отнести следующие распределенные системы управления: Centum CS3000 фирмы Yokogawa, DeltaV фирмы Emerson Process Management, I/A Series фирмы Foxboro и др.

2. *Открытые системы*, сетевые комплексы на основе микропроцессорных контроллеров, характеризуется тем, что применяются открытые пакеты прикладных программ для операторских станций, предназначенных не для конкретной микропроцессорной системы, а приспособленные для применения в разных программно-технических и сетевых средах (рис.3).

Свойство открытости состоит в том, что пакет прикладных программ:

- поддерживает совокупность интерфейсов и драйверов, позволяющих использовать пакет для широкого класса микропроцессорных контроллеров, систем и сетей передачи данных;
- содержит средства разработки новых интерфейсов и драйверов для микропроцессорных приборов и сетей;
- позволяет расширять функциональные возможности систем в соответствии с заданиями на конкретные проекты путём подключения программ пользователя.

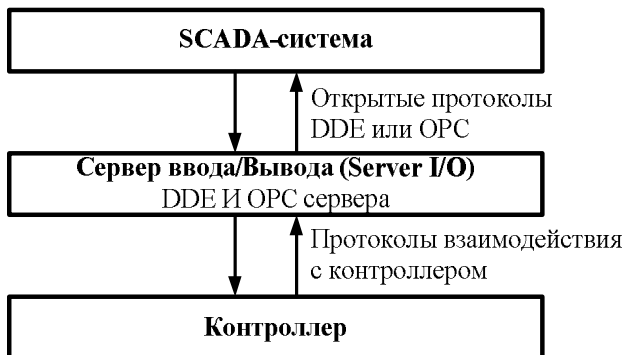


Рис. 3. Схема применения SCADA-систем

Для подсоединения драйверов ввода/вывода к SCADA-системе в настоящее время используются следующие механизмы:

- протокол DDE (Dynamic Data Exchange - динамический обмен данными);
- собственные протоколы фирм-производителей SCADA-систем;
- новый OPC-протокол (OLE for Process Control).

Изначально протокол DDE применялся в первых человеко-машинных интерфейсах в качестве механизма разделения данных между прикладными системами и устройствами типа ПЛК (программируемые логические контроллеры). Протокол DDE представляет собой коммуникационный протокол для обмена данными между различными Windows-приложениями. Этот протокол реализует взаимосвязи типа клиент-сервер между двумя одновременно исполняющимися программами.

Основная цель OPC стандарта (OLE for Process Control) заключается в определении механизма доступа к данным с любого устройства из приложений. OPC позволяет производителям оборудования поставлять программные компоненты, которые стандартным способом обеспечивают клиентов данными с ПЛК.

Стандарт OPC разрабатывался специально для использования в промышленной автоматизации, и он имеет проблемно-ориентированную модель взаимодействия, которая реализована через совокупность COM/DCOM-интерфейсов.

Стандарт состоит из трех основных спецификаций:

- 1) доступ к данным ПВ (Data Access);
- 2) обработка тревог и событий (Alarms & Events);
- 3) доступ к историческим данным (Historical Data Access).

OPC-серверов, соответственно, тоже может быть три вида, хотя не возбраняется совмещать все эти функции в одном. OPC-серверы физических устройств обычно являются только серверами данных (Data Access Servers). Серверы тревог и исторические чаще всего применяются на серверах данных. Сервер тревог формирует определенные логические переменные, называемые состояниями (conditions), имея в качестве исходной информации некую переменную (тег), полученную от сервера данных. Серверы исторических данных получают от серверов данных параметры в реальном времени и архивируют их, а затем предоставляют эти данные другим приложениям (например, для построения графиков трендов).

Центральное место среди спецификаций OPC занимает доступ к данным РВ (Data Access). Базовым понятием этой спецификации является элемент данных (Item). Каждый элемент данных (т. е. фактически - параметр технологического процесса) имеет значение, время последнего обновления (timestamp) и признак качества, определяющий степень достоверности значения. Значение может быть практически любого скалярного типа (булево, целое, с плавающей точкой и т.п.) или строкой (на самом деле это так называемый OLE VARIANT). Время представляется с 100-наносекундной точностью (на самом деле это FILETIME Win32 API). Качество – это код, содержащий в себе грубую оценку достоверности параметра – UNCERTAIN, GOOD и BAD (не определено, хорошее и плохое), а на случай плохой оценки - ещё и расшифровку, например, QUAL_SENSOR_FAILURE – неисправность датчика.

К этому типу взаимодействия можно отнести следующие SCADA-системы: InTouch фирмы Wonderware, iFix фирмы Intellution, Genesis фирмы Iconics и др.

3. Аппаратная реализация связи с устройствами ввода/вывода

Для организации взаимодействия с контроллерами могут быть использованы следующие аппаратные средства:

- COM-порты; в этом случае контроллер или объединенные сетью контроллеры подключаются по протоколам RS-232, RS-422, RS-485;
- сетевые платы; использование такой аппаратной поддержки возможно, если соответствующие контроллеры снабжены интерфейсным выходом на Ethernet;

- вставные платы; в этом случае протокол взаимодействия определяется платой и может быть уникальным; в настоящее время предлагаются реализации в стандартах ISA, PCI, CompactPCI.

4. Программные средства для операторских станций в системе автоматизации управления производством (SCADA-системы)

Программные системы и пакеты прикладных программ, обеспечивающие работу компьютерных операторских станций, в литературе получили наименование «SCADA-программы» (Supervisory Control and Data Acquisition – сбор данных, наблюдение и управление).

SCADA-программы применяют в своей деятельности пользователи-сотрудники организаций следующих групп:

- промышленных предприятий, разрабатывающих и реконструирующих микропроцессорные системы управления производственными процессами;
- проектных и наладочных фирм, создающих и модернизирующих системы контроля и управления;
- системных интеграторов, разрабатывающих эффективные программно-технические комплексы управления, использующие технические и программные средства разных изготовителей;
- фирм-разработчиков микропроцессорных средств автоматизации управления.

SCADA-программы обеспечивают реализацию основных функций операторских станций в реальном времени:

- сбор текущей технологической информации от контроллеров или других приборов и устройств, связанных непосредственно или через сеть с операторской станцией;
- необходимую первичную обработку измерительной информации, а также вычислительную и логическую обработку технологических данных в операторских станциях;
- архивизацию и хранение текущей информации, и ее дальнейшую обработку;
- представление текущей и исторической информации на дисплее (в формах динамизированных мнемосхем, гистограмм, анимационных изображений, таблиц, трендов, аварийных сообщений и т.д.);
- печать отчетов и протоколов в задаваемых формах – по времени или по запросу оператора;
- регистрацию аварийных ситуаций в моменты их возникновения и вывод аварийных сообщений на экран или на внешние устройст-

ва;

- ввод команд и сообщений оператора, их обработку или передачу в контроллеры и другие устройства;
- подключение и организацию взаимодействия прикладных программ пользователя с переменными базы данных реального времени и командами оператора;
- информационные сетевые взаимодействия между узлами системы управления.

Каждая SCADA-программа содержит в своем составе две базовые подсистемы:

- *инструментальная система* (система разработки) – среда разработки программного обеспечения, действующего в составе операторских станций;
- *исполнительная система*, поддерживающая работу программного обеспечения операторских станций в реальном времени.

Инструментальная система применяется при разработке программ операторских станций проектантами и системными интеграторами. Каждый экземпляр инструментальной системы многократно используется для разработки операторских станций в разных проектах; применяется также эксплуатационниками на предприятиях для коррекции и модернизаций программного обеспечения станций.

Исполнительная система поддерживает работу программного обеспечения отдельной станции. Инструментальная система технически может поддерживать также работу операторской станции в реальном времени и может быть применена в качестве исполнительной. В некоторых специализированных SCADA-программах инструментальная и исполнительная системы не разделены и представляют собой единое целое.

Преимущества использования SCADA-программ по сравнению с непосредственным программированием операторских станций состоят в следующем. Практически исключается необходимость в привлечении высококвалифицированных программистов для разработки операторских станций. Для этого достаточно квалификации специалиста по автоматизации производства и программиста средней квалификации; значительно сокращаются затраты труда и времени на разработку операторских станций; поддерживается на высоком уровне качество созданных программ; существенно повышается удобство работы оператора производства.

5. Характеристики SCADA-систем

1. Общие данные SCADA-программ:

- фирма-разработчик; год первого выпуска и общий тираж (характеристика опыта фирмы, отработанности и популярности SCADA-программы); распространители в России и СНГ;

- примеры предприятий в России, эксплуатирующих SCADA-программу (возможность получения отзывов о реальной работе SCADA-программы).

2. Структурные особенности SCADA-программ:

- структурное строение пакета: модульность (возможность формировать функциональный состав операторских станций комбинацией составляющих SCADA-программу программных модулей);

- реализация структуры клиент-сервер, типы реализуемых станций (в т.ч. без непосредственной связи с технологическим процессом через УСО или контроллеры);

- наличие вариантов SCADA-программы, различающихся информационной мощностью (число входов-выходов); специальных станций в номенклатуре SCADA-программы – просмотра технологических данных и архивов (станций руководства) и архивных; специальных программных систем для обеспечения работы смежных уровней управления – непосредственного управления процессами, диспетчерского управления и др.

3. Функциональные характеристики SCADA-программ:

- возможности графического редактора, используемого для построения мнемосхем;

- средства отображения: многообразие динамических изменений любых элементов мнемосхем, использование элементов мультипликации, многооконный экран; наличие библиотек «мастер-объектов» (Wizard) – готовых изображений со средствами отображения и других фрагментов проектов, возможность их пополнения пользователями;

- характеристики трендов реального времени (оперативных) и архивных (исторических);

- поддержка специальных клавиш оператора (горячие клавиши, сенсорные зоны, возможности функциональных клавиатур);

- возможности модификации отображения данных на работающей операторской станции (переключение переменных, выводимых на мнемосхемы и на тренды, изменение масштабов кривых на графиках, коррекция форм отчетов и др.);

- количественные ограничения в SCADA-программе: число архивируемых переменных, отображаемых переменных на одной мнemo-

схеме; число мнемосхем на одной операторской станции и т.п.;

- характеристики встроенных средств обработки данных: мощность библиотеки арифметических, логических, управленческих и других программных модулей и способа программирования алгоритмов относительно несложной обработки данных;

- языки и процедуры создания пользовательских алгоритмов обработки данных;

- возможности построения рапортов, отчетов, протоколов в заданных формах;

- характеристики защиты от несанкционированного доступа к станции оператора: число уровней доступа, число индивидуальных паролей на каждом уровне;

- характеристики средств обеспечения надежности: диагностика связи операторской станции с контроллерами и другими операторскими станциями: возможности горячего резервирования станции оператора и/или ее отдельных функций.

4. Используемая аппаратно-программная платформа:

- минимальные и рекомендуемые требования к техническим характеристикам компьютера для реализации операторских станций (в структурных вариантах: отдельная станция, структура клиент - сервер, станция наблюдения и т. д.);

- циклы ввода-вывода данных; циклы обновления данных на экранах станции (при рекомендуемой аппаратно-программной платформе).

5. Характеристики открытости SCADA-программы:

- поддерживаемые интерфейсы межпрограммного взаимодействия, в том числе: обмена данными с приложениями, например, DDE; с базами данных - SQL/ODBC; объектные средства взаимодействия OLE, ActiveX; поддержка универсального промышленного интерфейса OPC;

- число драйверов и серверов ввода-вывода для связи с контроллерами и внешними устройствами отечественных и зарубежных фирм, перечень обслуживаемых ими контроллеров и устройств; характеристики средств разработки новых драйверов и серверов ввода-вывода;

- перечень поддерживаемых стандартных сетевых протоколов.

6. Данные о распространении и сопровождении SCADA-программы:

- наличие, состав, полнота документации, в том числе, на русском языке;

- наличие русифицированной версии SCADA-программы, пол-

нота русификации;

- особенности сопровождения SCADA-программы: консультации, горячая линия;
- место, сроки и условия обучения пользователей работе со SCADA-программой.

6. *Стоимостные характеристики SCADA-программы – цены:*

- базового состава SCADA-программы: инструментального комплекса; дополнительного исполнительного комплекса;
- функционально ограниченных вариантов SCADA-программы;
- сервера SCADA-программы (при ее клиент/серверной архитектуре);
- одного драйвера к определенному типу контролеров;
- инструментария для разработки новых драйверов контроллеров;
- обучения пользователей.

6. Выбор SCADA-программы для конкретной системы автоматизации производства

При выборе SCADA-программы для конкретного проекта необходимо вначале четко определить набор требований к характеристикам операторских станций. На их основе определяются требования к SCADA-программе и подвергаются анализу имеющиеся на рынке SCADA-программы на соответствие поставленным требованиям. Из удовлетворяющих требованиям проекта выбирается программа, наилучшая по совокупности технико-экономических показателей.

Остановимся на характеристиках и данных объекта автоматизации и разрабатываемой автоматизированной системы управления, которые следует учитывать при выборе SCADA-программы.

Характеристики объекта и требования к системе автоматизации:

- характер технологического процесса (непрерывный, периодический, дискретный и т.д.), его информационная мощность, динамика процессов, территориальная распределенность объекта управления;
- функции системы управления (контроль и/или учет, или контроль и управление, или диспетчеризация, или сочетание ряда функций);
- необходимое число операторских станций, их взаимное расположение и требуемые информационные взаимосвязи; связи с другими участками и службами предприятия;
- число измеряемых и управляемых величин и функции перера-

ботки текущей технологической информации на каждой операторской станции;

- требования к динамике обновления информации на экране операторской станции, к объему и периодичности записей данных в исторический архив, к формам дисплейных кадров, к удобству и полноте представления измеряемой информации оператору;

- требования к средствам обеспечения надежности основных функций операторских станций (диагностика неисправностей, резервирование станций или их отдельных устройств или функций);

- перспективы повторного использования проекта или его фрагментов.

Характеристики разрабатываемой системы автоматизации:

- характеристики сетевой структуры системы автоматизации, протоколы и формы обмена информацией системы с другими подразделениями предприятия;

- информационная структура разрабатываемой системы – связи между источниками и потребителями данных (в вариантах: локальная структура – только непосредственные связи станция-контроллеры; распределённая структура: несколько станций используют общие контроллеры и обмениваются информацией между собой);

- типы контроллеров и других устройств, передающих и принимающих текущие данные от операторских станций; их число: типы передаваемых сигналов;

- требования к аппаратно-программной платформе операторских станций (в том числе к типам компьютеров, к операционным системам, к ресурсам памяти и т.д.);

- квалификация пользователей SCADA-программы. допустимый объем непосредственного программирования и требования к языкам программирования для реализации пользовательских программ.

В результате анализа перечисленных характеристик, ограничений и требований подготавливается четкое, полное и конкретное техническое задание на SCADA-программу.

В случае крупной закупки целесообразно с помощью консалтинговой фирмы организовать тендерные торги. Поэтапная процедура проведения тендера выглядит следующим образом:

- составляются полные, конкретные и однозначно понимаемые технические требования на исполнительные и соответствующие инструментальные системы;

- на основе данных настоящего сопоставительного обзора, выделяются 3-5 SCADA-программы лучше других удовлетворяющие составленным техническим требованиям с учётом экономических ог-

раничений;

- продавцам SCADA-программ рассылаются приглашения на участие в закрытом тендере;

- заказчиком с помощью консалтинговой фирмы формируются критерии выбора наиболее подходящей («оптимальной») SCADA-программы;

- при получении от продавцов тендерных технико-коммерческих предложений проводится их сопоставление по следующей процедуре:

- 1) анализ технических характеристик по представленной документации на соответствие техническим требованиям; если некоторые из предложений оказываются неполными или не соответствующими заданию, они уточняются в дополнительных переговорах с продавцами;

- 2) организация экспертной группы, в задачу которой входит отбор для покупки SCADA-программы из числа участвующих в тендере по критериям, установленным покупателем;

- 3) личное ознакомление экспертов с технико-коммерческими предложениями и с предлагаемыми SCADA-программами предпочтительно на территории фирм-продавцов: оценка полноты и ясности документации, проведение с помощью продавцов разработки и отладки нескольких существенных для автоматизируемого объекта тестовых примеров;

- 4) ознакомление с опытом практического применения: организация контактов с предприятиями, эксплуатирующими рассматриваемые системы; в переговорах с непосредственными пользователями должны быть выявлены достоинства и недостатки систем, которые проявляются только при практическом использовании;

- 5) на основе всей полученной информации и заданных заказчиком критериев экспертами производится ранжирование участвующих в тендере SCADA-программ по отдельным критериям;

- 6) решается задача многокритериального выбора, на основе которой выявляется наиболее подходящая для данного объекта автоматизации SCADA-программа.

7. SCADA-система InTouch фирмы Wonderware

7.1 Общие сведения

Система InTouch фирмы Wonderware (США) распространяется с конца 80-х годов, продано более 120 тыс. систем. Версия InTouch 7.11 выпущена в 2001 г. Она реализует 32-разрядную арифметику, работает под Windows 9x/NT 4.0/2000/XP/Vista/7.

InTouch является базовым продуктом в составе интегрированного комплекта программных систем The Factory Suite 2000 (FS 2000) фирмы Wonderware. Системы в комплекте FS 2000 обеспечивают организацию рабочих мест операторов, диспетчеров и руководителей производства на основных уровнях оперативного управления производством: непосредственного цифрового управления исполнительными механизмами, управления технологическими процессами, диспетчерского управления всем производством, а также реализуют информационную поддержку работы предприятия средствами Internet/Intranet.

7.2 Программные компоненты InTouch

Программное обеспечение InTouch состоит из трех основных модулей: InTouch Application Manager (Проводник приложений InTouch), WindowMaker и WindowViewer. InTouch содержит также диагностическую программу Wonderware Logger (рис. 4).

Проводник приложений InTouch (InTouch Application Manager), предназначен для создания новых приложений и открытия уже существующих приложений в WindowMaker или в WindowViewer. Он также используется для настройки WindowViewer в качестве службы Windows NT/2000/XP/Vista/7, настройки технологии сетевой разработки приложений (NAD) и т.д.

Среда разработки WindowMaker. В среде разработки с помощью объектно-ориентированной графики создаются анимационные сенсорные окна. Эти окна могут подключаться к данным ввода/вывода промышленных контроллеров и к данным других приложений Microsoft Windows.

Среда исполнения WindowViewer. В среде исполнения отображаются графические окна, созданные с помощью WindowMaker. WindowViewer выполняет сценарии InTouch, отвечает за ведение журналов и подготовку отчетов по архивным данным и алармам и может выступать в роли клиента или сервера для коммуникационных протоколов DDE и SuiteLink.

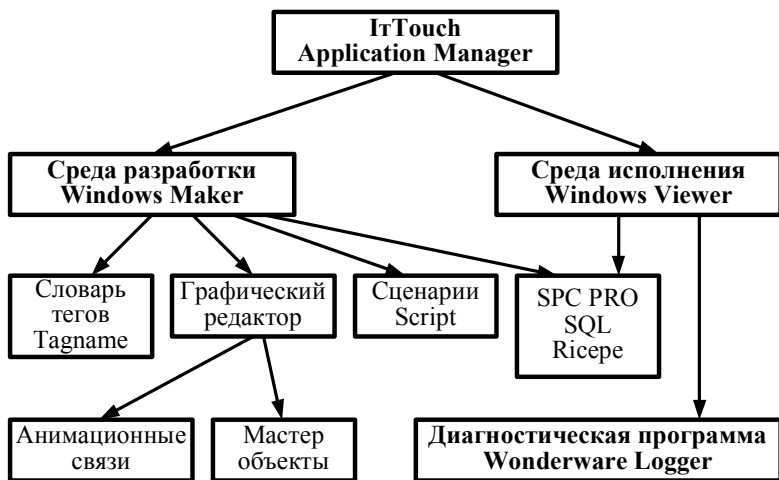


Рис. 4. Структура программного обеспечения InTouch

Словарь переменных (Tagname Dictionary), это база данных реального времени, где хранится вся информация о переменных приложения.

Объектно-ориентированная графика. Мощные средства проектирования облегчают рисование, выравнивание, вращение, дублирование, копирование, вставку, разделение на слои и многие другие операции. InTouch допускает практически неограниченное количество динамических изображений в каждом окне.

Анимационные связи (Animation Link). Эти связи обеспечивают возможность «оживления» любых объектов и их комбинаций для создания практически неограниченного набора «мультипликационных» характеристик.

Мастер-объекты. InTouch включает в себя обширную библиотеку Wizards, т.е. предварительно сконфигурированных вспомогательных средств (переключатели, счетчики, регуляторы и т.п.), позволяющих быстро создавать прикладные программы для конкретных условий производственного предприятия.

Сценарии (Script). Мощный, гибкий и удобный в использовании язык сценариев – одно из преимуществ InTouch по сравнению с теми SCADA, которые для создания сценариев используют VBA.

Менеджер рецептов (Recipe Manager). Данный модуль позволяет легко создавать, модифицировать и загружать рецепты или установочные параметры. Такие переменные, как уставки, ингредиенты, временные

ные и температурные характеристики перемещаются в базу данных и могут быть загружены в контроллер в любой момент.

Модуль статистического контроля технологического процесса (SPC Pro). Этот модуль, используемый в виде одноузловой или распределенной системы, является мощным средством мониторинга и анализа статистических показателей технологического процесса. Анализ рассчитанных в реальном масштабе времени статистических показателей часто позволяет улучшить качество процесса и, в итоге, качество продукции.

Модуль обращения к базам данных с помощью структурированного языка запросов (SQL). Данный модуль предоставляет пользователям доступ к наиболее известным базам данных (MS SQL Server, MS Access, Oracle, dBase и т.д.), для которых поддерживается стандарт связности открытых баз данных (ODBC).

Диагностическая программа Wonderware Logger. Программа Wonderware Logger записывает информацию о всех операциях, выполняемых на компьютере. Например, начало работы с данными, возникновение ошибок, данные сервера ввода/вывода и т.д. Программа состоит из двух компонентов: интерфейс пользователя и регистрация событий.

Интерфейс пользователя (WWLOGVWR.EXE) представляет собой приложение, в котором вы можете просматривать сообщения об ошибках и информативные сообщения, генерируемые программными компонентами Wonderware. В этом приложении можно настроить формат отображения сообщений, указать место расположения файла журнала, определить срок его хранения на диске и т.д.

Регистрация событий (WWLOGSVC.EXE) действует как фоновая задача, которая обрабатывает входящие сообщения и передает их для отображения в соответствующем приложении, а также выполняет функцию форматирования и записи текстовых строк в файл журнала.

7.3 Организация взаимодействия с контроллерами

InTouch для организации взаимодействия, поддерживает стандартный протокол DDE (FastDDE и NetDDE) и специально разработанный протокол SuiteLink. В основе протокола SuiteLink лежит протокол TCP/IP, со стандартным интерфейсом Microsoft Winsock и обладает следующими характеристиками:

- передача данных осуществляется в формате VTQ (Value, Time, Quality - значение, время, качество), в соответствии с которым каждая пересылаемая клиенту единица информации сопровождается метками

времени и качества данных;

- благодаря системному монитору операционной системы Windows NT (Performance Monitor) стал возможным расширенный анализ производительности по передаче данных, степени загрузки сервера, степени потребления ресурсов компьютера и сети, что особенно важно для проектирования и сопровождения больших распределенных промышленных сетей;

- поддержка обмена данными между приложениями происходит независимо от того, исполняются ли эти приложения на одном узле сети или на разных.

Для реализации функций OPC - клиента Wonderware предлагает OPCLink-сервер, преобразующий OPC в SuiteLink-протокол. OPCLink-сервер позволяет подключить все продукты Wonderware к OPC. Он подключается локально или удаленно к любому OPC серверу и передает данные с помощью DDE, FastDDE и SuiteLink любому локальному или удаленному клиенту. Используя OPCLink вы можете:

- просмотреть и отредактировать темы, созданные в файле конфигурации;
- преобразовать запросы DDE/SuiteLink и OPC;
- отобразить значения элементов, меток времени и качества данных;
- редактировать темы;
- загрузить и сохранить файлы конфигурации;
- просмотреть элементы в выбранных OPC серверах;
- отследить статус элементов.

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 (рис.5).

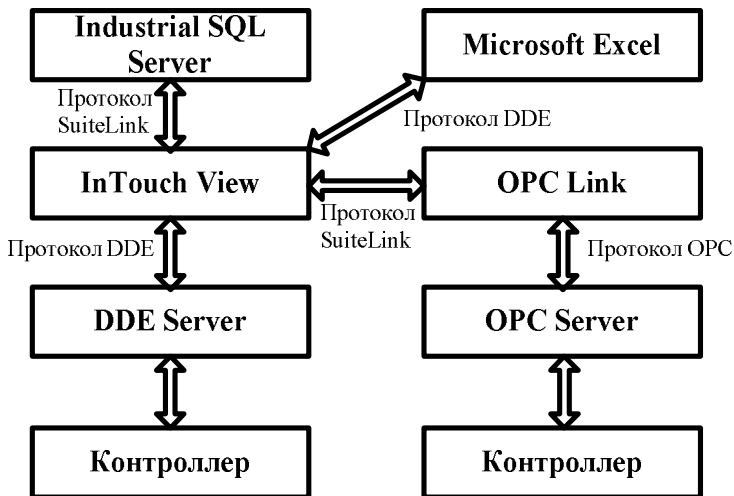



Рис. 5. Организация взаимодействия с контроллерами

7.4 Лицензирование

Пакет Wonderware FactorySuite 2000 состоит из различных программных компонентов (в т.ч. InTouch). Соглашение о лицензии Wonderware предоставляет вам право использовать и отображать данное программное обеспечение на одном компьютере. Кроме того, Wonderware предоставляет несколько механизмов лицензирования и активизации для определенных программ пакета. Система управления лицензиями FactorySuite может работать как при наличии, так и при отсутствии аппаратного ключа (dongle). При успешной проверке лицензионной информации активизируются соответствующие программные компоненты FactorySuite 2000. В пакете FactorySuite 2000 все продукты Wonderware активизируются с помощью одного файла лицензии – WWSUITE.LIC. При покупке любого продукта Wonderware в этот файл включаются данные о пользователе, о функциональных возможностях продукта, серийный номер и другая информация. Файл лицензии WWSUITE.LIC распространяется на дискете. Для установки файла лицензии необходимо использовать утилиту лицензии FactorySuite 2000 (*Start → Programs → Wonderware FactorySuite → Common → License Utility*). Затем выбрать в меню *File* пункт *Install License File* (или кнопку ) и указать файл лицензии WWSUITE.LIC.

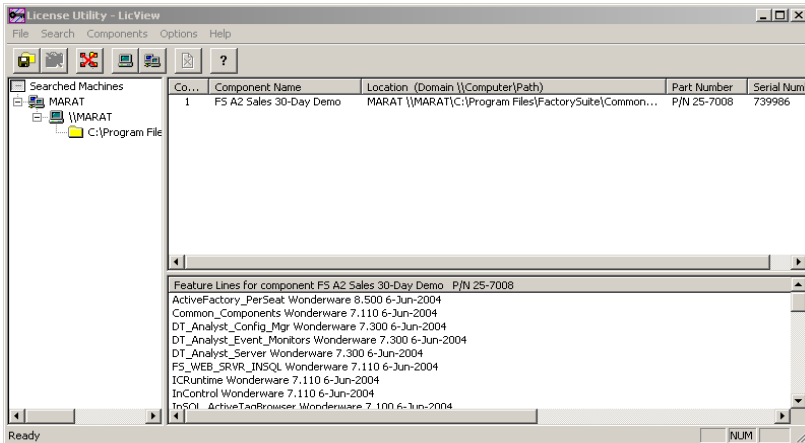


Рис. 6. Окно утилиты лицензирования

Эта утилита позволяет также просматривать информацию о лицензии. Для просмотра или управления файлами лицензий необходимо иметь права администратора в том каталоге, где эти файлы находятся. Например, чтобы установить файл лицензии в каталог на сетевом компьютере, вы должны иметь права администратора в этом каталоге.

Утилита лицензии позволяет добавлять различные компоненты лицензии в файл лицензии для активизации различных продуктов на одном компьютере. Утилита лицензии приравнивает продукт как компонент лицензии к соответствующим функциональным линиям в файле лицензии.

В случае если лицензия отсутствует, то с InTouch можно работать в демонстрационном режиме. Функциональные ограничения на работу InTouch в демонстрационном режиме:

- работа программы завершается через 2 часа;
- пользовательские тэги - не более 32;
- нельзя установить новые элементы ActiveX.

7.5 InTouch Application Manager

Проводник приложений InTouch предназначен для создания новых приложений и открытия уже существующих приложений в WindowMaker или в WindowViewer. Он также используется для настройки WindowViewer в качестве службы Windows NT/2000, настройки технологии сетевой разработки приложений (NAD) и т.д.

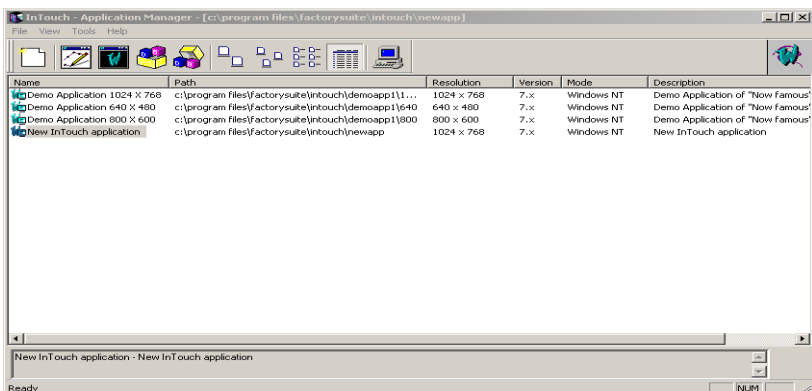


Рис. 7. Окно InTouch Application Manager

Где:

- кнопка создания нового приложения;
- кнопка запуска WindowMaker;



- кнопка запуска WindowViewer;
 - утилита DBDump, используется для экспорта словаря тэгов приложения InTouch в текстовый файл, который можно просмотреть или отредактировать в другой программе, например, в MS Excel;



- утилита DBLoad позволяет загрузить должным образом отформатированный файл словаря тэгов в существующее приложение InTouch;



- кнопки для изменения вида отображения проектов;



- кнопка настройки свойств узла.

Для создания нового приложения проделайте следующие операции. Запустите Проводник приложений (*Start* → *Programs* → *Wonderware FactorySuite* → *InTouch*). В меню *File* выберите команду *New* или



нажмите кнопку на панели инструментов. Появится окно мастера создания нового приложения *Create New Application*. Нажмите *Next*. Появится второе окно данного мастера. В поле ввода введите путь к папке, в которой вы хотите создать новое приложение (нужно ввести новую папку специально для данного приложения). Нажмите *Next*. В следующем окне в поле *Name* введите уникальное имя для нового

приложения, которое появляется в окне InTouch - Application Manager. В поле Description введите его описание (вводить описание необязательно). Нажмите Finish. Снова появится окно проводника приложений, в котором вы увидите строку (значок) с только что введенным именем созданного приложения. Чтобы открыть приложение, выберите его в списке, затем нажмите кнопку WindowMaker на панели инструментов (WindowViewer не запускается для нового приложения). Открыть приложение можно также с помощью двойного нажатия на его строку (значок) или выбора приложения и нажатия ENTER.

Для добавления проекта, созданного на другом компьютере, в список необходимо выполнить следующие действия, зайти в меню *Tools* выбрать пункт *Find Application* (или нажать правую кнопку на пустом месте и выбрать пункт *Find Application*) и указать директорию, где находится проект.

7.6 Среда разработки WindowMaker

WindowMaker поддерживает стандартные графические интерфейсы пользователя (GUI) ОС MS Windows, включая использование правой кнопки мыши, перемещаемых и закрепленных панелей инструментов, выпадающих меню и т. п.

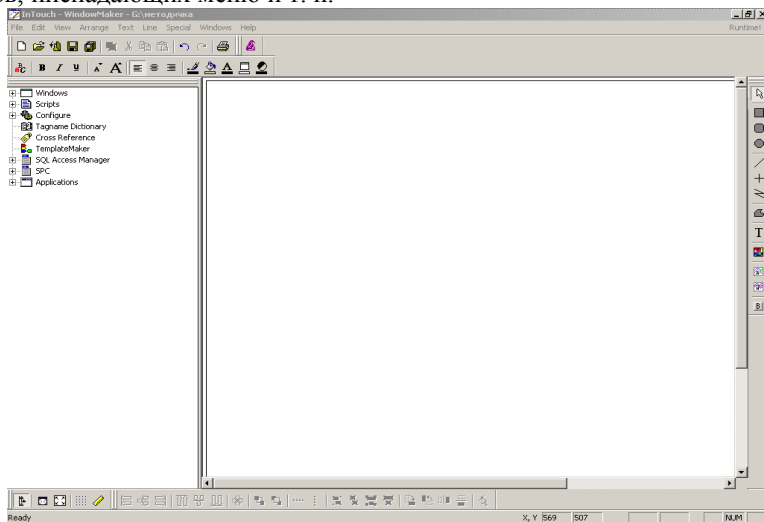


Рис. 8. Среда разработки WindowMaker

В правой части экрана находится Application Explorer, проводник приложений, представляет приложение в иерархическом виде с досту-

пом к любому компоненту приложения и многим, часто используемым командам и функциям WindowMaker.

В WindowMaker включены следующие инструментальные панели:

- General (основная) - содержит инструменты, выполняющие большинство команд управления окнами, которые находятся в меню File, а также инструменты управления буфером обмена из меню Edit (рис. 9);



Рис. 9. Панель инструментов General (основная)

- Wizards/ActiveX - содержит по умолчанию единственный инструмент, используемый для доступа к диалоговому окну Wizard Selection (на эту панель инструментов можно добавить любой установленный мастер-объект или элемент ActiveX);



Рис. 10. Панель инструментов Wizard/ActiveX

- Format (форматирование) - группирует инструменты, выполняющие большинство команд, находящихся в меню Text, а также здесь имеются инструменты цветовой палитры для выбора цветов линии, текста, заливки, фона и прозрачного объекта;



Рис. 11. Панель инструментов Format

- Arrange (монтаж) -- группирует такие инструменты (выполняющие команды меню Arrange), как различные виды выравнивания объектов, переносы объектов на задний и передний план, создание (разборка) символов и ячеек, повороты объектов;



Рис. 12. Панель инструментов Arrange

- Draw Object (рисование объекта) - включает инструменты для создания простых и сложных объектов. Первые восемь инструментов и последний предназначены для создания простых объектов: прямоугольник (квадрат), скругленный прямоугольник, окружность (эллипс), прямая линия под любым углом, горизонтальная и вертикальная

прямая, ломаная линия, многоугольник, текстовый объект и трехмерная кнопка. С помощью остальных трех инструментов панели Drawing могут быть созданы сложные объекты операторских интерфейсов: контейнер для вставки растровых изображений, тренд реального времени и архивный тренд.



Рис. 13. Панель инструментов Drawing

- View (вид) - содержит инструменты, выполняющие многие команды для управления состоянием окна WindowMaker из меню View;



Рис. 14. Панель инструментов View

В панели View (Вид) представлено всего пять кнопок (слева направо):

- кнопка, соответствующая команде отображения/ закрытия окна Application Explorer;
- кнопка, соответствующая команде Hide All (спрятать все), относящейся к панелям инструментов;
- кнопка переключения обычного изображения окна в полноэкранное и обратно;
- кнопка, соответствующая команде Snap to Grid (привязать к координатной сетке);
- кнопка отображения/закрытия линейки окна WindowMaker.

Следует отметить, что многие инструменты недоступны до тех пор, пока окно не будет открыто, а объекты не будут размещены в окне и выбраны. Если инструмент не активен, то это означает, что его функция не применима к текущему состоянию окна или выбранному объекту.

Для запуска WindowViewer необходимо нажать на Runtime в строке меню WindowMaker (или выполнить команду WindowViewer из меню File). Если WindowMaker (Window-Viewer) запустить непосредственно средствами MS Windows, то автоматически открывается последнее приложение из ранее открытых в среде разработки (в среде исполнения).

7.7 Среда исполнения WindowViewer

В среде исполнения отображаются графические окна, созданные с помощью Window-Maker. WindowViewer выполняет сценарии InTouch, отвечает за ведение журналов и подготовку отчетов по архивным данным и алармам и может выступать в роли клиента или сервера для коммуникационных протоколов DDE и SuiteLink.

Многие свойства среды исполнения можно настроить в среде разработки. Например, задать тактовый интервал для проверки внутренних таймеров и частоту обновления системных тэгов времени, ввести режим отладки сценариев, задать параметры автоматического подключения к серверам ввода/вывода (в случае отсутствия соединения), ввести значения системных тэгов отсутствия активности в системе, задать параметры окна программы WindowViewer, определить окна, которые будут открываться при непосредственном запуске WindowViewer и т.д.

Кнопки заголовка окна WindowViewer и команды меню используются, как правило, при отладке приложения или для удобства работы пользователя с правами администратора (у оператора к этим инструментам обычно нет доступа, что обеспечивается с помощью соответствующих настроек в WindowMaker, либо специальных всплывающих окон). В строку меню WindowViewer входят следующие элементы:

- *File* - открытие и закрытие окон в среде исполнения, переключение на программу WindowMaker, закрытие Window-Viewer, просмотр параметров лицензии;
- *Logic* - запуск и прерывание всех сценариев (кроме уже выполняемых асинхронных сценариев);
- *Special* - подключение к серверам ввода/вывода (при отсутствии соединения), восстановление процессов регистрации алармов и архивирования, остановка процесса архивирования, просмотр сообщений диагностической программы Wonderware Logger, работа с функциями безопасности;
- *Development* - переход в среду разработки.


8. Лабораторные задания по InTouch

Лабораторная работа № 1. Окна InTouch.

Цель работы:

Изучение свойств окон в WindowMaker (WM).

Теоретические положения

Свойства каждого окна (наличие заголовка, цвет фона, размеры и т. д.) определяются при его создании. Создание нового окна производится в среде разработки WindowMaker щелчком по иконке  панели инструментов General или командой *File/New Window*. На экране появится диалог Window Properties (Рис. 15)

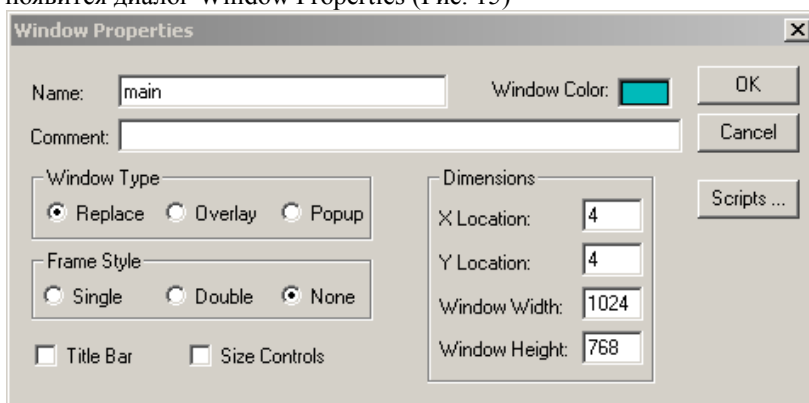


Рис. 15. Свойства окна

Каждое окно должно иметь свое имя для его идентификации в приложении (Name). Цвет фона создаваемого окна выбирается из цветовой палитры, вызываемой на экран щелчком по окошку Window Color.

В поле Comment можно ввести комментарий, связанный с этим окном (необязательно). Эта информация нужна только для документирования и не используется приложением.

InTouch предлагает три типа окон (Window Type):

- Replace (заменяющее) - закрывает все существующие окна, перекрываемые им при появлении на экране, включая окна типа Popup и другие окна типа Replace.

- Overlay (перекрывающее) - появляется поверх всех отображаемых в текущий момент окон. Когда окно типа Overlay закрывается, все скрываемые им окна восстанавливаются. Щелчок мыши по любому видимому участку лежащего ниже окна приводит к переходу его на передний план.

- Popur (всплывающее) - похоже на окно типа Overlay, только оно всегда остается поверх всех других открытых окон. Окно закрывается после соответствующей команды пользователя.

Выбор типа создаваемого окна производится включением соответствующей кнопки в поле Window Type.

В поле Frame Style (стиль обрамления) выбирается необходимый стиль обрамления окна:

- Single - окно с рамкой, допускается заголовок;
- Double - окно с рамкой без заголовка;
- None - окно без рамки и заголовка.

Чтобы у окна была полоса с заголовком, где выводится имя окна, включают опцию Title Bar. Эта полоса также служит для перемещения окна при захвате ее мышью. При выборе этой опции отключатся опции Double и None для стиля обрамления.

Для возможности изменения размеров окна, когда оно откроется в WindowMaker, следует выбрать опцию Size Controls (управление размером).

В группе полей Dimentions определяются текущие размеры и положение окна на рабочем поле:

- X Location - расстояние в пикселях между левым краем рабочего поля WindowMaker и левым краем описываемого окна;
- Y Location - расстояние в пикселях между верхним краем рабочего поля WindowMaker и верхним краем описываемого окна;
- Window Width - ширина окна в пикселях;
- Window Height - высота окна в пикселях.

По умолчанию при создании нового окна эти параметры примут значения предыдущего (последнего) созданного окна.

Кнопка Scripts (скрипты) дает возможность войти в диалог Window Script для создания оконного сценария.

Содержание работы

1. Создать четыре окна следующих типов: Replace, два Overlay и Popur. Расположив их так, как показано на рис. 16, чтобы они перекрывали остальные окна. Проверить влияние изменения различных

параметров панели Window Properties на внешний вид и поведение окон в WindowViewer.

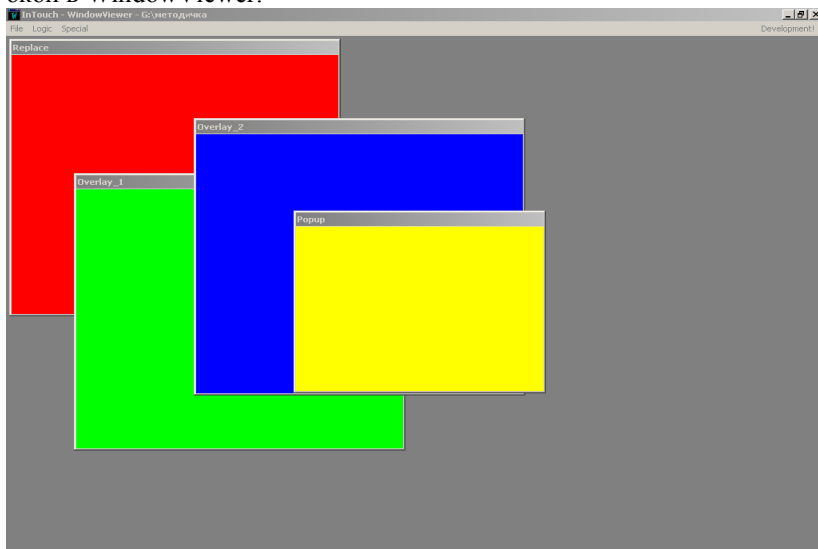


Рис. 16. Расположение заданных окон.

2. Удалите эти окна. Создайте два окна со следующими размерами:

1 окно Menu – тип PopUp, x – позиция = 0; y – позиция = 0; ширина = 1024; высота = 200.

2 окно Main – тип Replace; x – позиция = 0; y – позиция = 200; ширина = 1024; высота = 568.

У обоих окон убрать галочку Title Bar и Size Controls.

Методические указания по работе

1. Для создания проекта необходимо запустить InTouch, выбрать *File/New*, далее указать путь и название проекта и открыть проект в WindowMaker.

2. Для создания окон в WindowMaker необходимо выбрать *File/New Window* и появившемся окне Window Properties задать настройки окна.

Лабораторная работа № 2. Объекты InTouch.

Цель работы:

Работа с графическими и текстовыми объектами (создание, изменение, выравнивание, отличие символов от ячеек и т.п.). Знакомство с библиотекой мастер-объектов InTouch (Wizards). Вставка графических объектов из стандартных Windows-приложений.

Теоретические положения

Объекты и их свойства

Простые объекты. WindowMaker поддерживает четыре базовых типа простых объектов: линии, заполненные контуры, текст и кнопки. Каждый из этих простых объектов имеет свойства, влияющие на его внешний вид. Такими свойствами являются цвет линии, цвет заполнения, высота, ширина, ориентация и т.д., и они могут быть статическими или динамическими.

- Линия - это объект, представляющий собой один или несколько связанных отрезков. Толщина линии и ее стиль являются статическими свойствами линии, присваиваемыми ей во время создания, через меню *Line*, и лишь цвет линии может быть связан с анимационной функцией.

- Заполненный контур (прямоугольник, скругленный прямоугольник, круг, эллипс, многоугольник) представляет собой двухмерный объект. К динамическим свойствам такого объекта относятся цвет контурной линии, цвет заполнения, насыщенность цвета заполнения, высота, ширина, расположение, видимость и ориентация.


- Текст представляет собой последовательность символов. К статическим свойствам текста относятся тип шрифта, его размер, выделение, курсив, подчеркивание, выравнивание. Анимационные свойства шрифта - цвет, видимость и расположение.

- Кнопка - часто используемый объект при создании операторских интерфейсов. С кнопками могут быть связаны функции различных типов. Нажатие кнопки может вызвать выполнение скриптов, кнопкой можно производить ввод аналоговых и дискретных величин и т.д.


Текст редактируется с помощью команды *Special/Substitute Strings...* или при помощи клавиш *Ctrl+L*.

Сложные объекты.


Символ – это некоторая комбинация простых объектов, которые обрабатываются как один объект. Они создаются путем выбора двух и более объектов последующего запуска команды меню *Arrange/Make*

Symbol или нажатием на кнопку . Любое изменение статических или динамических свойств символа влияет на все составляющие символа. Например, если создать символ "насос" из двух кругов и двух прямоугольников и присвоить ему динамическое свойство *Fill Color* (цвет заполнения), то это свойство будет распространяться на все четыре простых объекта.

Ячейка – это совокупность двух или более объектов, символов или других компонентов, образующих единый элемент. Они создаются путем выбора двух и более объектов, символов или ячеек и последующего запуска команды меню *Arrange/Make Cell* или нажатием на кнопку

. Ячейки реализуют пространственную взаимосвязь между составляющими их графическими элементами. Каждая составляющая ячейки может иметь свои собственные динамические свойства. Ячейки используются для таких виртуальных устройств, как панель управления контроллером, движковый регулятор и т. д.

Ячейка не может менять свой размер, ему нельзя присваивать динамические свойства (внутри компонента есть объекты и символы со своими динамическими свойствами). Нельзя изменять и статические свойства (внешний вид). Для изменения статических и динамических свойств компонента его надо разобрать на составные части командой

меню *Arrange/Break Cell* или нажатием на кнопку . Однако компоненты можно дублировать, копировать, вставлять, выравнивать, перемещать и т. д.

Мастер-объект - это предварительно созданный компонент с определенными статическими и динамическими свойствами, находящийся в библиотеке мастер-объектов (Wizards) и доступный для многократного применения. Но, в отличие от компонента, динамические свойства которого настраиваются для каждой составляющей отдельно до объединения в компонент, динамические свойства мастер-объекта быстро настраиваются с помощью специализированного диалога.

Выделение и установка размеров объектов


После того, как вы нарисовали объект и нажали на него, он окружается несколькими маленькими квадратами. Эти квадраты называются «узлами». Узлы используются для изменения размеров и формы

объекта. Понятие «выбранный» является ключевым при редактировании графических объектов в WindowMaker. Наличие «узлов» вокруг объекта означает, что он выбран. Чтобы выбрать объект, нужно нажать прямо на него. При нажатии на пустой участок окна отменяется выбор любого объекта в окне. Как правило, любая выполняемая команда применяется ко всем выбранным объектам, если она предназначена для этих объектов. Чтобы выделить несколько объектов, выберите сначала первый объект, затем, удерживая нажатой клавишу SHIFT, нажимайте на другие объекты, которые нужно выбрать. Чтобы отменить выбор отдельного объекта из группы выбранных, удерживайте нажатой клавишу SHIFT и нажмите на нужный объект. Для выбора группы объектов поместите курсор на пустой участок окна. Нажмите кнопку мыши и перемещайте курсор. Появится пунктирная рамка. Перемещайте курсор до тех пор, пока рамка выделения не охватит полностью все нужные объекты. Отпустите кнопку мыши. Все объекты, которые оказались полностью внутри рамки, будут выбраны. При нажатии левой кнопки мыши вместе с клавишей SHIFT объект, расположенный под курсором, не будет перемещаться. Вместо этого появится рамка выделения.


Работа с мастерами


Для того чтобы вставить мастера в рабочее окно, нужно нажать



кнопку  на панели инструментов Wizards/ActiveX. Появится диалоговое окно Wizard Selection. В списке мастеров следует выбрать нужную категорию мастера, далее на правой панели выбрать конкретного мастера, затем нажать ОК, либо дважды нажать на мастера. После этого курсор примет форму угла с символом «W» при возврате в окно приложения. В окне нужно нажать на то место, куда хотите вставить мастера. Затем дважды нажать на вставленный объект, чтобы его настроить (если нужно). Для того, чтобы иметь возможность использовать объекты библиотеки Symbol Factory после установки InTouch, нужно дополнительно установить Productivity Pack (указанный программный продукт находится на установочном диске InTouch). После того, как установка Productivity Pack будет завершена, в WM следует открыть закладку Wizard Installation окна Wizard/ActiveX Installation (меню *Special - Configure Wizard/ActiveX Installation*), убедиться, что в списке Installed Wizards есть строка Symbol Factory 1.1.13, нажать ОК. После этого библиотека Symbol Factory (и другие дополнительные мастера) появится в списке мастеров в окне Wizard Selection.

Работа с изображениями и растрами

Все графические объекты типа растровых рисунков, снимков экранов изображений в форматах BMP, JPEG, JPG, PCX, TGA, которые создаются в других программах Windows, должны вставляться в растровый контейнер . WM воспринимает рисунок как один простой объект. В WM можно поворачивать растровые рисунки, изображения. Любые связи рисунка поворачиваются вместе с ним. Можно также создать рисунок с прозрачным цветом, чтобы были видны объекты, расположенные за ним, для этого необходимо выбрать прозрачный

цвет, нажав на кнопку  из панели Format. Если создать рисунок с прозрачным цветом, то цвет фона окна и любые объекты за рисунком будут видны везде, где будет использоваться данный прозрачный цвет (в каждом рисунке можно использовать только один прозрачный цвет).

Содержание работы

1. Используя кнопочные панели WindowMaker и библиотеку Wizards, в окне Main создать учебный объект визуализации InTouch по аналогии с представленным на рис.17. Внешний вид объекта и расположение элементов могут варьироваться по усмотрению обучающего.

2. В MS Word Art оформить название своего предприятия и вставить его в окно InTouch. Сделать прозрачный фон данного объекта.

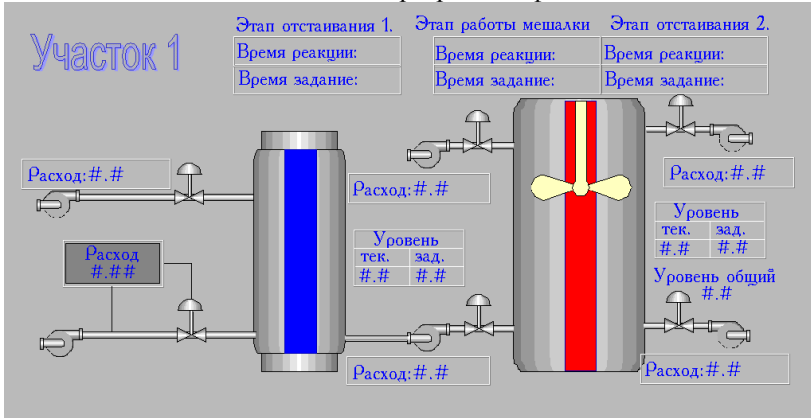




Рис. 17. Внешний вид учебного пособия.

Методические указания по работе

1. Трубы, вентили, бак и стрелки-индикаторы изобразите (максимально упрощенно) самостоятельно с помощью панели инструментов для рисования объекта, либо выберите из библиотеки символов Symbol Factory (кнопка Wizards, мастер Symbol Factory, кнопка ОК, нажатие в окне, выбор элемента в каком-либо из разделов: 3D Pipes, 3D Tanks, Arrows). Кнопки диспетчерского открывания/закрывания вентилей и соответствующие индикаторы, панели для ввода/вывода числовой и текстовой информации также изображаются либо с помощью подходящих мастер-объектов, либо самостоятельно. Индикация степени заполнения бака выполняется с помощью панелей инструментов для рисования объекта и монтажа (при выполнении данного задания не следует использовать раздел Scales из библиотеки символов).

2. Оформленный в MS WordArt текст вставьте в MS Paint. Далее

в рабочем окне WindowMaker с помощью кнопки  начертите рас-
тровый контейнер и вставьте в него свою надпись из MS Paint (формат
BMP), выбирая Paste Bitmap в меню Edit. Фон надписи сделайте про-

зрачным с помощью кнопки Transparent Color  панели форматиро-
вания.

Лабораторная работа № 3. Теги и анимационные связи.

Цель работы:

Работа с библиотекой тэгов (типы тэгов, определение нового тэга, поля тэгов и т.п.). Знакомство с форматами адресов ввода/вывода в InTouch. Создание составных типов тэгов (супертэги). Изучение особенностей ввода/вывода числовой и символьной информации в окнах InTouch. Задание различных анимационных связей для объектов InTouch.

Теоретические положения

Определение имени доступа в словаре переменных InTouch

Словарь тэгов (база данных выполнения) – основа системы InTouch. Во время выполнения база данных содержит текущие значения всех своих записей. Для того, чтобы создать базу данных выполнения, InTouch необходима информация обо всех создаваемых переменных. Каждой переменной должен быть присвоен тэг и тип. InTouch также необходима дополнительная информация о некоторых типах переменных.

В пакете InTouch используется два базовых типа переменных - Memoгу (внутренние) и I/O (переменные ввода/вывода).

Переменные типа Memoгу могут быть использованы для создания различных системных констант, моделирования элементов системы управления и в вычисляемых переменных, доступных другим Windows-программам.

Все тэги, которые считывают или записывают свои значения в другом приложении Windows, относятся к типу внешних тэгов. Сюда относятся все входы и выходы программируемых контроллеров, компьютеров управления процессами, а также данные из узлов сети. Доступ к внешним тэгам осуществляется при помощи коммуникационных протоколов Microsoft Dynamic Data Exchange (DDE) и Wonderware SuiteLink. При изменении значения внешнего тэга чтения/записи оно немедленно записывается в удаленное приложение. Тэг можно также обновить из удаленного приложения путем изменения в этом приложении записи, с которой этот тэг связан.

Определение новой переменной в базе данных InTouch, как и просмотр, и модификация атрибутов уже существующих переменных, производится в диалоге Tagname Dictionary (рис.18). Доступ к этому диалогу осуществляется командой меню *Special/Tagname Dictionary* в

окне среды разработки WindowMaker или двойным щелчком по иконке Tagname Dictionary в окне Application Explorer.

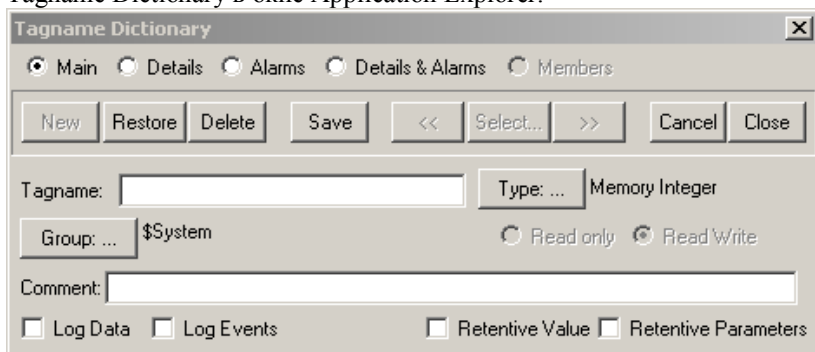


Рис.18. Диалог Tagname Dictionary (Словарь переменных).

Переключатели параметров в верхней части диалогового окна Tagname Dictionary используются для выбора детализации отображения тэгов:

- **Main** - Отображает главное диалоговое окно словаря тэгов. При работе с супертэгами параметр Main отображает только родительский или корневой тэг. Любые изменения в родительском или корневом тэге могут перезаписать информацию тэга-члена. Произведя изменения, нажмите Save. Появится запрос подтверждения перезаписи тэга-члена изменениями корневого тэга.

- **Details** - Отображает детальное диалоговое окно для выбранного типа тэга.

- **Alarms** - Отображает диалоговое окно конфигурации алармов для выбранного типа тэга.

- **Details & Alarms** - Отображает как детальное диалоговое окно, так и конфигурацию алармов для выбранного типа тэга.

- **Members** - Отображает детальное диалоговое окно члена супертэга.

Поля **Tagname** и **Comment** предназначены для ввода имени переменной и соответствующего комментария. Тэги могут включать в себя до 32 символов и должны начинаться с буквенного символа (A-Z или a-z). Остальные символы могут быть с A по Z, a-z, 0-9, !, @, -, ?, #, \$, %, _, \ и &.

По умолчанию включена опция Read/Write (чтение/запись). Можно отметить и опцию Read Only, если в процессе исполнения WindowViewer должен только читать значение переменной.

В любое время в режиме разработки можно открыть список переменных приложения щелчком по кнопке **Select** для выбора соответствующей переменной, просмотра списка или модификации атрибутов. Диалог **Select Tag** (выбор переменной) представлен на рис. 19.

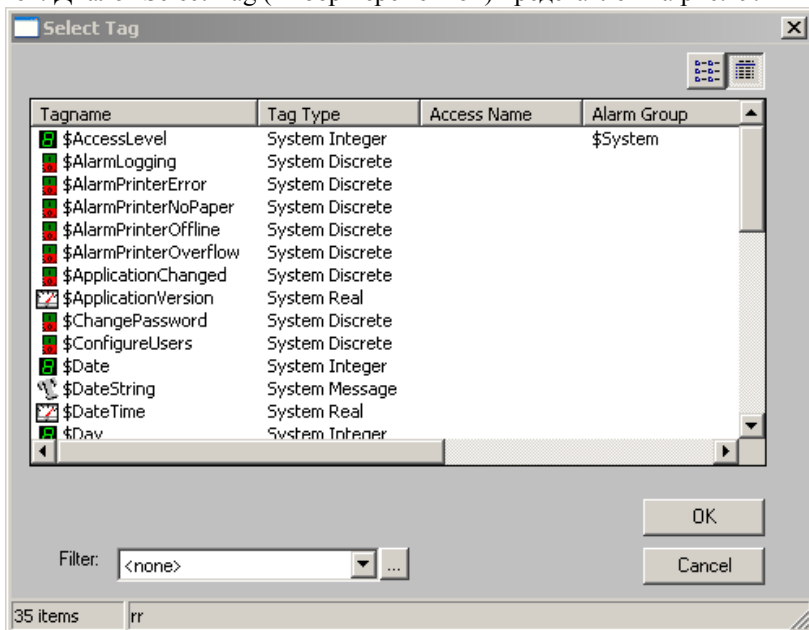


Рис.19. Диалог **Select Tag** (выбор переменной).

Для каждой переменной в этом диалоге приведена следующая информация: имя переменной, ее тип, имя доступа, группа аларма и комментарий.

Выбор **Log Data** нужен для регистрации тэга, во время выполнения приложения, в архивном файле. Регистрация происходит всякий раз, когда значение его единиц измерения превысит значение **Log Deadband** или по умолчанию один раз в час, независимо от изменений значения.

Выбор **Log Events** нужен для регистрации все изменений значений тэга, произведенные оператором, сервером ввода/вывода, Quick-сценарием или системой. При включении параметра **Log Events** активизируется поле **Priority**. Введенное в этом поле значение определяет уровень приоритета событий для тэга. В данное поле можно вводить значения от 1 до 999, где 1- высший приоритет, а 999-низший.

Выбор **Retentive Value** необходим, если нужно сохранить текущее значение тэга при выходе из WindowViewer. Это значение будет использоваться в качестве исходного значения тэга при каждом последующем запуске WindowViewer.

Выбор **Retentive Parameters** необходим, если нужно сохранять все изменения, производимые оператором со значением любых предельных полей тэга. Это значение будет использоваться в качестве исходного для алармов при перезапуске WindowViewer.

Группа алармов (**Alarm group**) для переменной определяется в диалоге, вызываемом нажатием кнопки Group диалога Tagname Dictionary.

Выбор типа переменной осуществляется в диалоге **Tag Types** (тип переменной, рис. 20), вызываемом на экран нажатием кнопки Type диалога Tagname Dictionary.

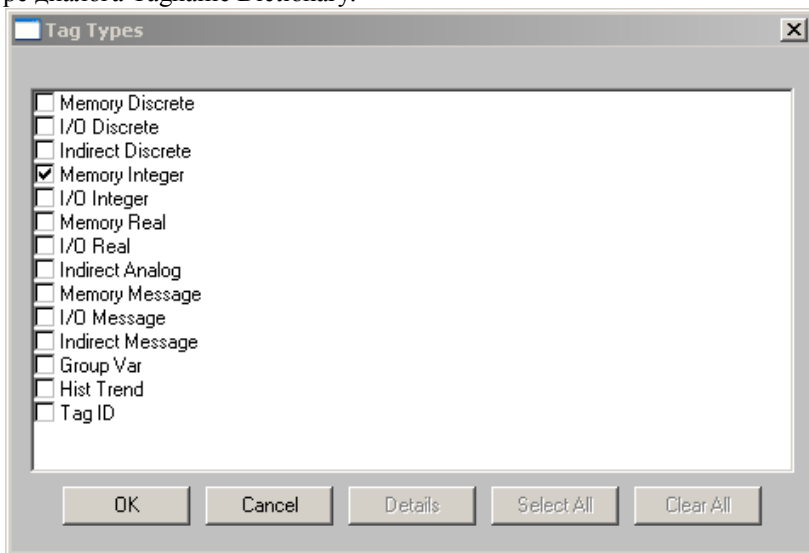


Рис.20. Диалог Tag Types (тип переменной).

В этом диалоге представлен полный список основных типов переменных InTouch. Выбор завершается отметкой соответствующей опции и щелчком по Ok.

После выбора типа переменной программа возвращает пользователя в диалог Tagname Dictionary (Словарь переменных). При этом будет открыт и дополнительный диалог подробного описания переменной, содержание которого зависит от выбранного типа.

На рис.21 представлен диалог подробного описания вещественной переменной типа I/O Integer.

The dialog box contains the following fields and options:

- Initial Value: 0
- Min EU: -32768
- Max EU: 32767
- Deadband: 0
- Min Raw: -32768
- Max Raw: 32767
- Eng Units: (empty text box)
- Access Name: ... (button) Unassigned
- Conversion:
 - Linear
 - Square Root
- Item: (empty text box)
- Use Tagname as Item Name
- Log Deadband: 0

Рис.21. Диалог подробного описания переменной типа I/O.

Значение **Initial Value**, будет присвоено тэгу при первой загрузке базы данных в среду выполнения.

В поле **Deadband** вводится величина, на которое должен измениться тэг, прежде чем будет обновлена база данных.

В поле **Min EU** вводится величина, отображаемое при получении минимального необработанного значения тэга.

В поле **Max EU** введите величина, отображаемое при получении максимального необработанного значения тэга.

В поле **Min Raw** введите минимальный предельный уровень необработанного целого значения внешнего тэга.

В поле **Max Raw** введите максимальный предельный уровень необработанного целого значения внешнего тэга.

Значения **Min EU**, **Min Raw**, **Max EU** и **Max Raw** используются для масштабирования внешних тэгов.

В поле **Eng Units** введите название единиц измерения данного тэга.

Поле **Conversion**, применяется, если будет использоваться база данных для масштабирования необработанных данных при расчете единиц измерения.

Linear – результат рассчитывается при помощи линейной интерполяции между конечными точками.

Алгоритм линейного масштабирования данных ввода выглядит следующим образом:

$$EUValue = (RawValue - MinRaw) * ((MaxEU - MinEU) / (MaxRaw - MinRaw)) + MinEU$$

Алгоритм линейного масштабирования данных вывода выглядит следующим образом:

$RawValue = (EUValue - MinEU) * ((MaxRaw - MinRaw) / (MaxEU - MinEU)) + MinRaw$

Square Root – для интерполяции используются необработанные значения.

Это удобно при масштабировании данных ввода с нелинейных устройств, таких как преобразователи давления.

Алгоритм масштабирования данных ввода по квадратному корню выглядит следующим образом:

$EUValue = \sqrt{RawValue - MinRaw} * ((MaxEU - MinEU) / \sqrt{MaxRaw - MinRaw}) + MinEU$

Алгоритм масштабирования данных вывода по квадратному корню выглядит следующим образом:

$RawValue = \text{square}(EUValue - MinEU) * (\sqrt{MaxRaw - MinRaw} / (MaxEU - MinEU)) + MinRaw$

Кнопка **Access Name** (имя доступа) используется для определения канала обмена (канала доступа) с сервером, с которым будет связана описываемая переменная. Имя доступа Access Name определяется именем узла, именем приложения и именем группы или топика. Имя топика должно совпадать с соответствующим именем, заданным при конфигурировании DDE, SuiteLink-сервера.

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 рис. 22 узлы исполнения (View-узлы), каждый со своей копией одного и того же приложения, ссылаются на свои собственные источники данных ввода/вывода (серверы ввода/вывода).



Рис. 22. Сеть View - узлов с собственными серверами 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.

Щелчок по кнопке Access Name (рис.23) вызывает на экран одноименный диалог.

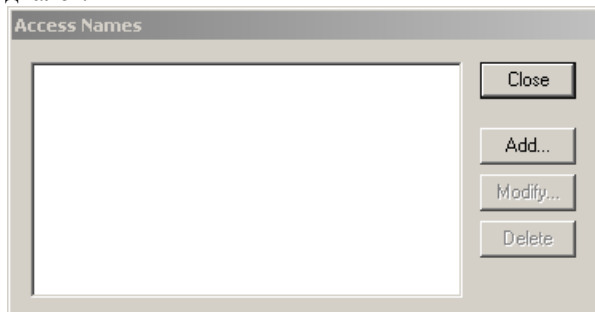


Рис. 23. Диалог Access Names (имена доступа).

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 (рис.24).

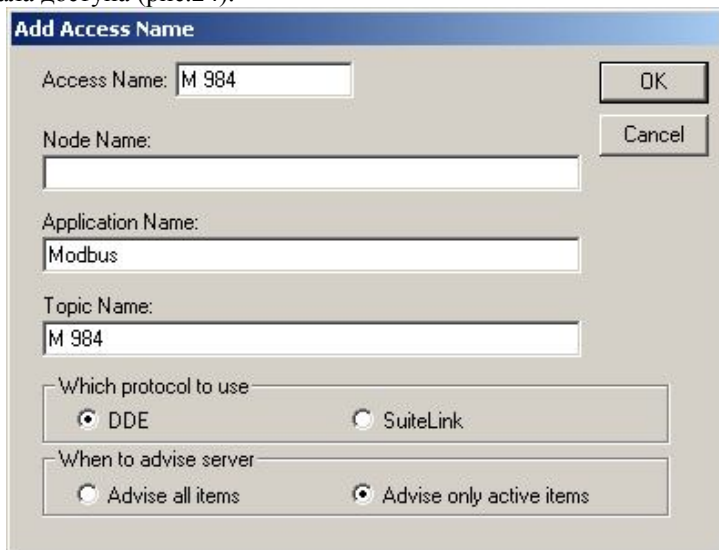


Рис.24. Диалог определения нового канала доступа (локальный адрес).

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 (рис.25).

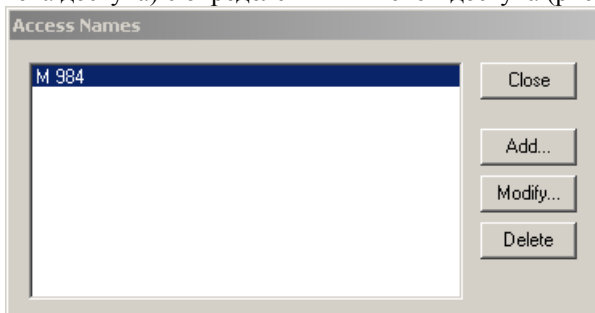


Рис.25. Диалог Access Names с определенным именем доступа.

Глобальные адреса источников данных ввода/вывода позволяют нескольким View-узлам обращаться к одному и тому же серверу ввода/вывода. Такой подход предоставляет возможность отказаться от нескольких серверов ввода/вывода, однако, он менее защищён от отказов (рис.26).

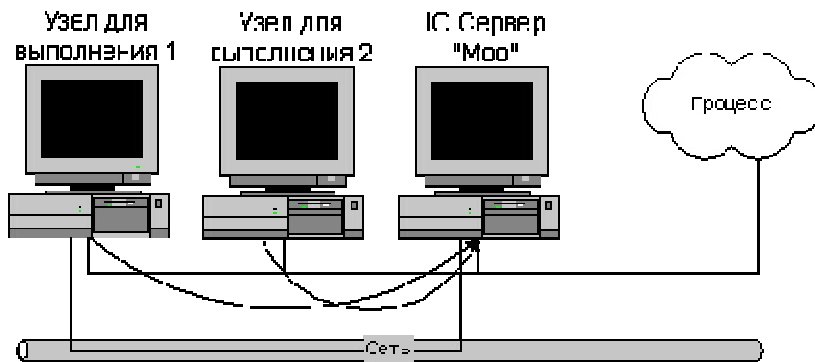


Рис.26. Архитектура с двумя View-узлами и сервером ввода/вывода.

Два View-узла исполняют идентичные копии одного и того же приложения и ссылаются на один и тот же источник ввода/вывода (I/O сервер). Поэтому при определении канала доступа к информации вво-

да/вывода необходимо использовать четырехуровневый адрес (Node-узел, Application-приложение, Topic-объект, Item-элемент).

Заполненный диалог при определении имени доступа для такой конфигурации представлен на рис. 27.

The image shows a dialog box titled "Add Access Name". It contains the following fields and options:

- Access Name:** Text box containing "PLC".
- Node Name:** Text box containing "Moo".
- Application Name:** Text box containing "Modbus".
- Topic Name:** Text box containing "PLC1".
- Which protocol to use:** Radio buttons for "DDE" (selected) and "SuiteLink".
- When to advise server:** Radio buttons for "Advise all items" and "Advise only active items" (selected).
- Buttons:** "OK" and "Cancel" buttons on the right side.

Рис.27. Диалог определения нового канала доступа (глобальный адрес).

При выборе имени доступа действует то же правило, что и при локальной адресации: рекомендуется, чтобы это имя совпадало с именем группы данных или топика (Topic Name). Но поле Node Name (имя узла) необходимо заполнить. В качестве этого имени при глобальной адресации выбирают имя узла, на котором установлен сервер ввода/вывода, являющийся источником данных для нескольких приложений.

Для каждой переменной ввода/вывода задается атрибут Access Name. С одним именем доступа, как правило, связано большое количество переменных. Распределение переменных по группам (топикам) произвольное. Но для оптимизации функционирования серверов рекомендуется в одну группу относить переменные с одинаковой частотой обновления. В противном случае частота, задаваемая при конфигурировании топика в сервере, должна соответствовать минимальному временному кванту. Желательно на этапе конфигурирования сервера определить группы (топики) для каждого частотного диапазона и в соответствии с этими группами создать имена доступа (Access Name) в

InTouch (лучше даже, чтобы имена групп совпадали с именами доступа). А далее каждую описываемую в InTouch-приложении переменную типа I/O связывать с подходящим именем доступа для обеспечения рационального пакетирования данных.

Супертэги

InTouch поддерживает структуру супертэгов, позволяющую создавать составные типы тэгов. Все тэги, созданные на основе супертэгов, работают так же, как обычные тэги. Они поддерживают тренды, алармы и все поля тэгов. Для удобства работы InTouch оснащён утилитой TemplateMaker, которую можно использовать для создания супертэгов. TemplateMaker позволяет создавать, редактировать и удалять шаблоны супертэгов и тэги-члены. InTouch сохраняет все шаблоны супертэгов в файле SUPERTAG.DAT в своем установочном каталоге (не в каталоге приложения). Это позволяет использовать шаблоны в любом приложении. Создавать супертэги можно различными способами. Например, можно создать супертэг непосредственно из словаря тэгов, в окне ввода выражений, в Quick-сценариях и т.д.

Поскольку тэги InTouch ограничены 32 символами, каждый из супертэгов **Родительский экземпляр\Дочерний тип\Подтип** также ограничен 32 символами.

Синтаксис ссылки супертэга поддерживается везде, где можно использовать обычные тэги. Например, допустимой ссылкой супертэга может быть: ColdRoom4\ EvapUnit1\ FanMotor2.MaxE.

Вывод числовых значений

Текстовые объекты могут использоваться для отображения статических или динамических числовых значений. Задавая текстовому объекту при помощи окна выбора анимационных связей User Inputs - Analog или Value Display - Analog, можно отображать значение целого или действительного тэга. Для указания формата отображения аналогового значения обычно используется символ «#» - номер или знак фунта и «.» - точка. Символ «#» отображает любой целый номер. Сочетание символа «.» и символов «#» справа от точки позволяет выводить действительное значение с требуемой точностью.

Удаление тэгов

В случае, если тэг используется в приложении, то удалить его нельзя. При помощи утилиты перекрестных ссылок InTouch (Cross Reference) можно определить названия окон и координаты объектов, а также сценарии InTouch, в которых используется данный тэг. Для не-

которых операций счетчик не обновляется автоматически, например, для удаления окна, изменения тэгов в связях или сценариях и т.п. В таких случаях InTouch продолжает считать, что тэг по-прежнему используется в приложении, и не дает его удалить. Для обновления информации о количестве используемых тэгах, используется утилита Update Use Counts из меню Special. Для удаления неиспользуемых тэгов используется утилита Delete Unused Tags из меню Special.

Анимационные связи

Один и тот же объект может иметь набор различных динамических свойств. Комбинации этих свойств предоставляют возможность создавать на экране в режиме исполнения (Runtime) практически любые динамические эффекты. Для установки динамических свойств необходимо вызвать на экран диалог их выбора (рис.28). Это достигается командой Special/Animation Link или двойным щелчком левой кнопки мыши на объекте.

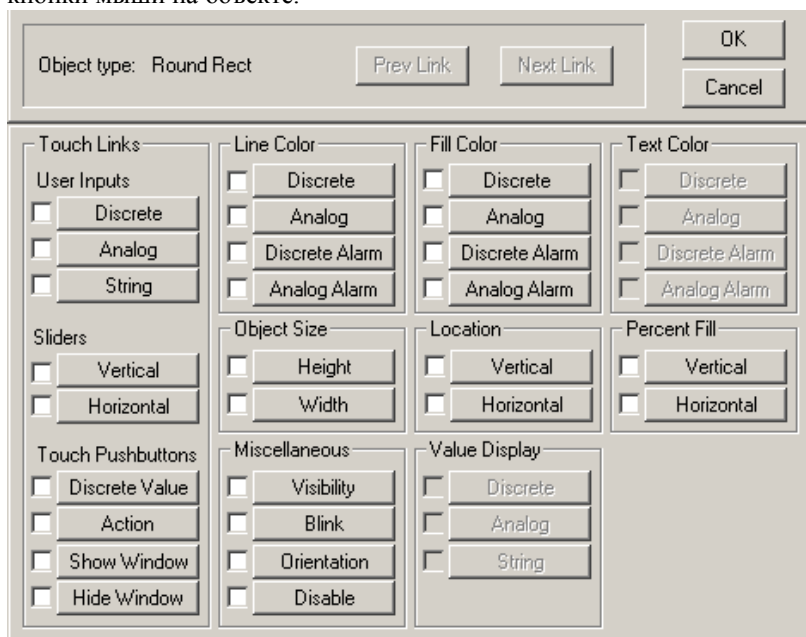


Рис. 28. Диалог выбора динамических свойств объекта.

Все динамические связи можно разделить на две группы: Touch Links (левая колонка) и Display Links (три колонки справа). С помо-

щью свойств Touch Links выполняется ввод данных в систему. Свойства Display Links осуществляют вывод информации на экран дисплея.

Нажатие на любую клавишу диалога (рис.28) вызывает появление нового диалога для определения соответствующего свойства объекта. Количество диалогов соответствует количеству динамических свойств (кнопок) диалога выбора. Все диалоги различны, но большинство из них имеет общие характеристики:

- окно типа объекта;
- одинаковую палитру цветов;
- быстрый вызов словаря переменных;
- быстрый доступ к полям переменных;
- поддержку правой кнопки мыши в полях Tagname (имя переменной) и Expression (выражение).

*Создание связи для ввода данных **User Inputs***

Связи по касанию (Touch Links) типа User Inputs позволяют создать сенсорные объекты для ввода данных оператором в систему (Рис. 29).

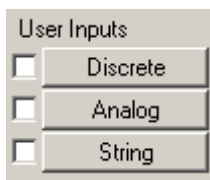


Рис. 29. Анимационные связи ввода данных в систему.

Ввод User Inputs можно поделить на три типа:

- **Discrete** - используется для управления значением дискретного тэга; при активизации связи данного типа открывается диалоговое окно, предлагающее оператору сделать выбор;

- **Analog** - используется для ввода значения аналогового тэга (целого или действительного); при активизации связи данного типа появляется поле, в котором значение можно ввести с обычной клавиатуры или с экранной цифровой;

- **String** - позволяет создать объект, в котором можно ввести строку сообщения; при активизации связи данного типа появляется поле, в которое можно ввести сообщение, используя обычную или экранную клавиатуру.

При выборе дискретного ввода данных появится следующий диалог (рис.30).

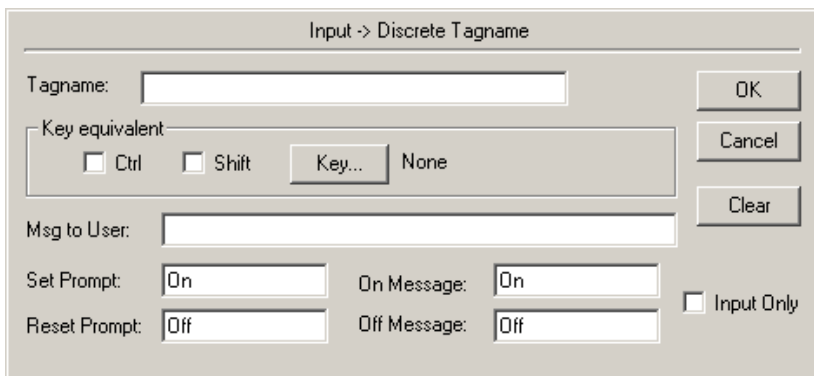


Рис. 30. Диалог настройки вывода дискретного тэга

В поле **Tagname** вводится тэг дискретного типа.

Нажатием кнопки **Key**, можно присвоить данной связи эквивалент вызова с клавиатуры.

В поле **Msg to User** вводится текст, который будет выводиться в диалоговом окне ввода при активизации данной связи.

В полях **Set Prompt** и **Reset Prompt** вводится текст для кнопок, на которые оператор будет нажимать в диалоговом окне ввода для включения и выключения дискретного значения.

В полях **On Message** и **Off Message** вводится текст, который должен появляться в текстовом поле при включении и выключении объекта.

Выбор **Input Only** необходим, если надо запретить отображение ввода в текстовом поле, соответствующем данному объекту.

*Создание связей по касанию ползунков **Sliders***

Связи по касанию ползунков **Sliders** позволяют создать объекты или символы, которые могут перемещаться по окну мышью или другими методами манипуляции, например, касанием сенсорного экрана пальцем (рис. 31).

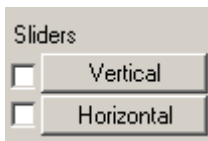


Рис. 31. Анимационные связи создания ползунков.

При перемещении объекта или символа изменяется значение связанного с ним тэга. Объект может иметь связи по касанию ползунка

горизонтального или вертикального типов, или и того и другого. При использовании обоих типов можно создать объект, изменяющий значения сразу двух тэгов.

При выборе создания вертикального ползунка появится следующее окно настроек (рис.32).

The image shows a dialog box titled "Vertical Slider". It has a "Tagname:" field at the top. Below it is a "Properties" section with four input fields: "At Top:" (value 0), "At Bottom:" (value 100), "Up:" (value 0), and "Down:" (value 261). Below the properties is a "Reference Location" section with three radio buttons: "Top", "Middle", and "Bottom" (which is selected). On the right side of the dialog are three buttons: "OK", "Cancel", and "Clear".

Рис. 32. Диалог настройки вертикального ползунка

В поле **Tagname** вводится тэг аналогового типа (целый или действительный).

В поле **At Top** вводится значение тэга для крайнего верхнего положения ползунка.

В поле **At Bottom** вводится значение тэга для крайнего нижнего положения ползунка.

В поле **Up** вводится количество пикселей, на которое ползунок может передвигаться вверх с текущей позиции.

В поле **Down** вводится количество пикселей, на которое ползунок может передвигаться вниз.

Выбор **Reference Location**, позволяет фиксировать позиции (вверх, середина, низ) объекта для перемещения ползунка.

*Создание связей по касанию кнопок **Touch Pushbuttons***

Связи по касанию кнопок (Touch Pushbutton) позволяют создать объекты, немедленно выполняющие определенное действие при нажатии на них мышью или при касании их пальцем на сенсорном экране (рис. 33).

The image shows a dialog box titled "Touch Pushbuttons". It contains four checkboxes, each followed by a button: "Discrete Value", "Action", "Show Window", and "Hide Window". All checkboxes are currently unchecked.

Рис. 33. Анимационные связи по нажатию кнопок.

Существует четыре типа связей по касанию кнопок:

Discrete Value (Дискретное значение) - позволяет превратить объект или символ в кнопку, управляющую состоянием дискретного тэга. Действиями кнопки могут быть установка значения, сброс, переключение, мгновенное включение (Direct) и мгновенное выключение (Reverse).

Action (Действие) - позволяет связать объект, символ или кнопку с тремя различными сценариями действия: **On Down**, **While Down** и **On Up**. Сценарии действия позволяют установить определенное значение тэга, показать и/или скрыть окно, запустить и управлять выполнением других приложений, выполнять функции и т.д.

Show Window (Показать окно) - превращает объект или символ в кнопку, при нажатии или касании которой открывается одно или несколько окон.

Hide Window (Скрыть окно) - превращает объект или символ в кнопку, при нажатии или касании которой закрывается одно или несколько окон.

*Создание связи по касанию кнопки дискретного значения **Discrete Value***

После выбора Discrete Value появится окно с настройками (рис.34).

В поле **Tagname** вводится тэг дискретного типа.

Нажав **Key**, можно присвоить данной связи эквивалент вызова с клавиатуры.

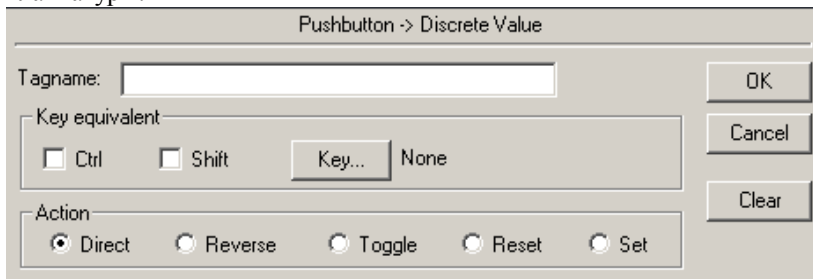


Рис. 34. Диалог настройки ввода дискретного значения.

Возможен следующий выбор **Action** для данной связи:

- **Direct** - выставляет значение равное 1 (True, On, Yes), пока кнопка удерживается нажатой. При отпускании кнопки значение автоматически переходит на 0 (False, Off, No).

- **Reverse** - выставляет значение равное 0 (False, Off, No), пока кнопка удерживается нажатой. При отпускании кнопки значение автоматически переходит на 1 (True, On, Yes).
- **Toggle** - выставляет обратное значение дискретного тэга при нажатии, например, если тэг равен 1, то при нажатии на кнопку он становится равным 0 и наоборот.
- **Reset** - при нажатии на кнопку выставляется значение равное 0 (False, Off, No).
- **Set** - при нажатии на кнопку выставляется значение равное 1 (True, On, Yes).

*Создание связи по нажатию на кнопку действие **Action***

После выбора данной анимационной связи появится окно редактора скриптов (рис.35).

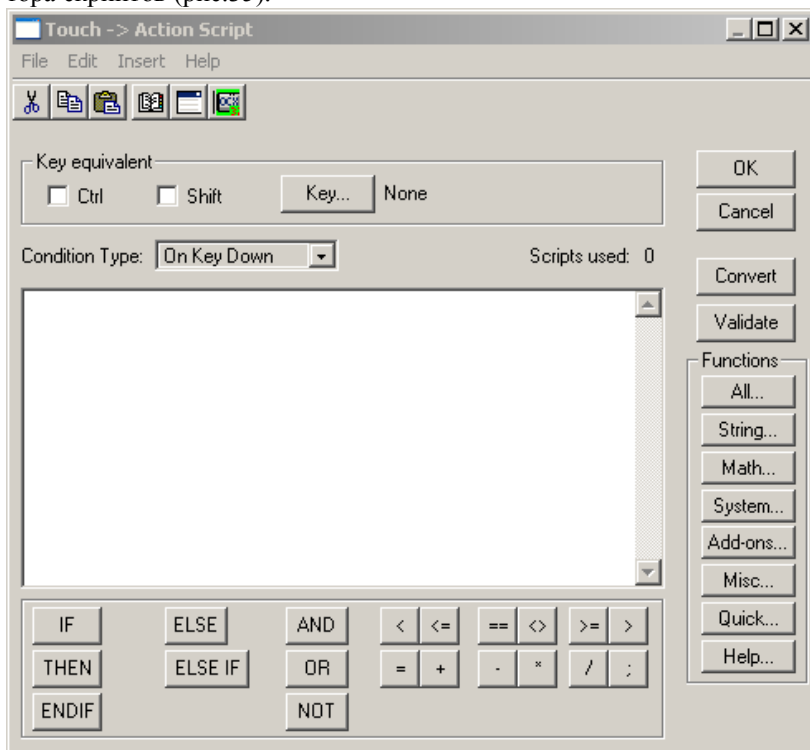


Рис. 35. Окно редактора скриптов.

Нажав стрелку **Condition Type** можно выбрать тип сценария для данного объекта. Для одной клавиши можно выбрать все три типа сценариев:

- **On Key Down** - выполняется сценарий однократно при нажатии на клавишу;
- **While Down** - выполняется сценарий многократно через заданный интервал, пока клавиша удерживается нажатой;
- **On Key Up** - выполняется сценарий однократно при отпускании клавиши.

*Создание связи по нажатию на кнопку открытия и закрытия окна **Show Windows** и **Hide Window***

После выбора данной анимационной связи появится окно выбора окон (рис.36).

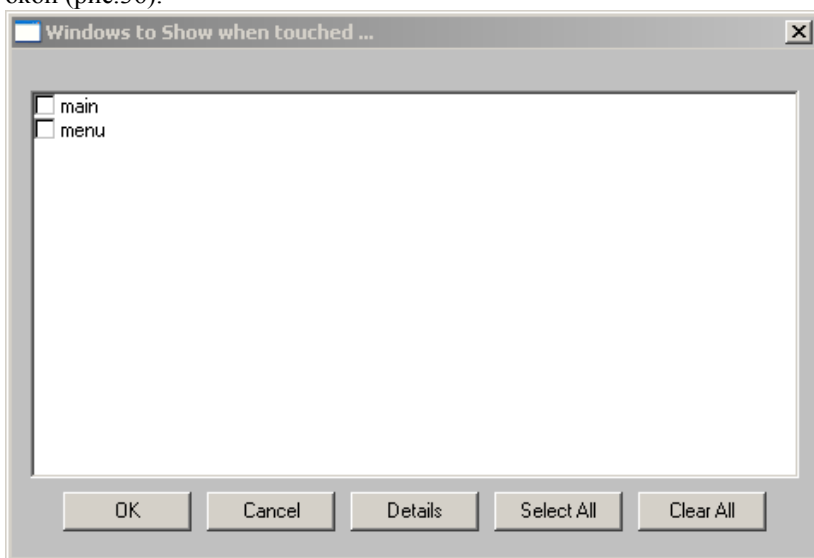


Рис. 36. Диалог выбора окон.

Создание связей отображения

Связи отображения используются для вывода информации оператору. Существует восемь типов связей отображения.

- **Линия, заливка и цвет текста** - дискретный, аналоговый, дискретный аларм.
- **Размер объекта** - высота, ширина.
- **Положение** - горизонтальное, вертикальное.
- **Процентная заливка** - горизонтальная, вертикальная.

- **Прочие** - видимость, ориентация, мерцание, отключение.
- **Отображение значения** - дискретное, аналоговое, текстовое.

*Создание цветовых связей **Line Color, Fill Color, Text Color***

Цветовые связи используются для анимации таких атрибутов объекта, как **цвет линии, цвет заливки и цвет текста** (рис. 37).

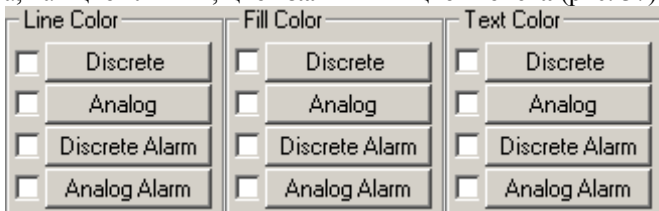


Рис. 37. Анимационные связи изменения цветов линий, заливки и текста.

Существует четыре типа цветовых связей линии, заливки и текста.

Дискретная - позволяет управлять цветовыми атрибутами заливки, линии и текста объекта или символа, связанного со значением дискретного выражения (рис. 38).

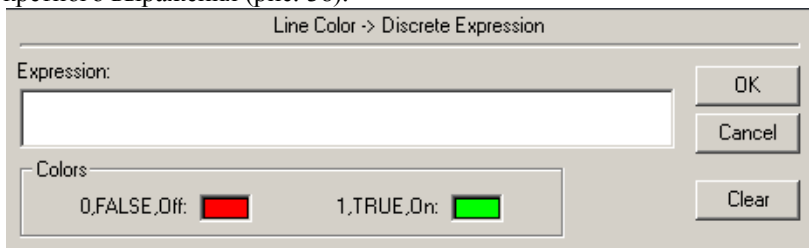


Рис. 38. Диалог изменения цвета по значению дискретного тэга

В поле **Expression** вводится дискретный тэг или выражение, равное true или false.

В группе **Colors** выбирается цвет на соответствующее состояние дискретного тэга.

Аналоговая - позволяет управлять цветовыми атрибутами заливки, линии и текста объекта или символа, связанного со значением аналогового выражения. Можно определить пять диапазонов значения с помощью четырех контрольных точек. Для обозначения пяти диапазонов значений можно выбрать пять цветов (рис.39).



Рис. 39. Диалог изменения цвета по значению аналогового тэга

В поле **Expression** введите аналоговый (**целый или действительный**) тэг или выражение, равное аналоговому значению.

В каждом поле **Break Points** можно задать значения контрольных точек (для тэгов действительного типа допустимы десятичные значения), при которых будет меняться цвет объекта.

В группе **Colors** определяется цвет, который будет использоваться для каждой контрольной точки.

Дискретный аларм - позволяет управлять цветовыми атрибутами заливки, линии и текста объекта или символа, связанного с состоянием аларма тэга, группы алармов или переменной группы. Цветовая связь такого типа позволяет выбрать два цвета: один для нормального состояния тэга, другой для состояния аларма. Эта связь может применяться как для аналоговых, так и для дискретных тэгов. При использовании с аналоговым тэгом она реагирует на любое состояние аларма тэга (рис.40)

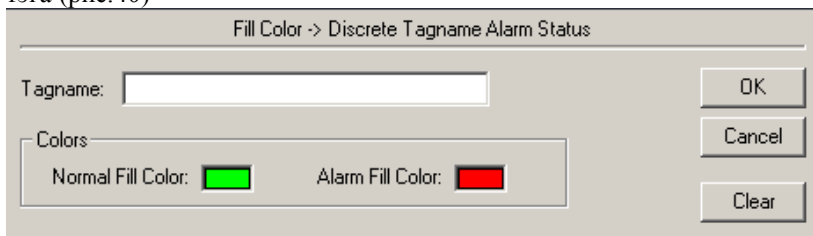


Рис. 40. Диалог изменения цвета по состоянию дискретного аларма тэга

В поле **Tagname** вводится дискретный тэг, чье состояние аларма должно быть связано с объектом.

В группе **Colors** выбирается цвет на соответствующее состояние аларма тэга.

Аналоговый аларм - позволяет управлять цветовыми атрибутами заливки, линии и текста объекта или символа, связанного с состоянием аларма аналогового тэга, группы алармов или переменной группы. Можно назначить один цвет для нормального состояния тэга и разные цвета для каждого состояния аларма (рис. 41).

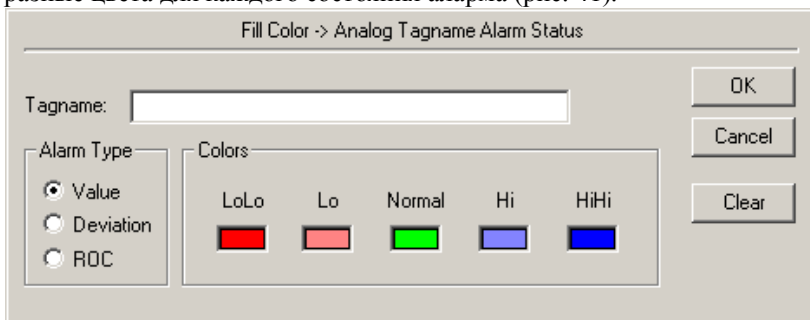


Рис. 41. Диалог изменения цвета по состоянию аналогового аларма тэга.

В поле **Tagname** вводится аналоговый (**целый или действительный**) тэг, чье состояние аларма должно быть связано с объектом.

В группе **Alarm Type** выбирается тип аларма для данного объекта. Существует три взаимоисключающих типа аналоговых цветовых связей:

- **Value Alarm** – можно выбрать до пяти разных цветов для каждого состояния алармов значения, определенных для данного тэга;
- **Deviation** – можно выбрать до трех разных цветов для каждого состояния алармов отклонения, определенных для данного тэга;
- **ROC** (скорость изменения) – можно выбрать два разных цвета для каждого состояния алармов скорости изменения, определенных для тэга.

В группе **Colors** выбирается цвет на соответствующее состояние аларма тэга.

*Создание связей размера объекта **Object Size***

Связи **размера объекта** позволяют изменять высоту и/или ширину объекта в зависимости от значения аналогового (**целого или действительного**) тэга или выражения. С помощью этих связей можно управлять направлением, в котором объект увеличивается по высоте и/или ширине. К объекту можно прикрепить одновременно связь высоты и связь ширины (рис.42).

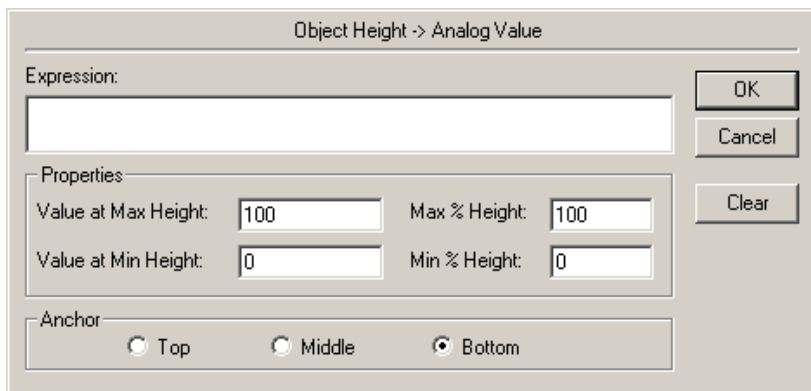


Рис. 42. Диалог настройки изменения размера объекта.

В поле **Expression** вводится аналоговый (**целый или действительный**) тэг или выражение, равное аналоговому значению.

В поле **Value at Max Height** вводится значение, которое будет вызывать изменение размера объекта до максимальной высоты.

В поле **Value at Min Height** вводится значение, которое будет вызывать изменение размера объекта до минимальной высоты.

В поле **Max % Height** вводится процент (0-100) высоты, который объект будет достигать при значении, заданном в поле **Value at Max Height**.

В поле **Min % Height** вводится процент (0-100) высоты, который объект будет достигать при значении, заданном в поле **Value at Min Height**.

В группе **Anchor** выбирается направление, в котором объект будет увеличиваться.

При выборе параметра **Top** объект будет увеличиваться в направлении от верхнего края вниз. При выборе **Middle** объект будет увеличиваться в обоих направлениях от середины. Параметр **Bottom** означает, что объект будет увеличиваться в направлении от нижнего края вверх.

*Создание связей положения **Location***

С помощью **связей положения** можно заставить объект автоматически передвигаться по горизонтали или вертикали, или в обоих направлениях при изменении значения аналогового тэга или выражения (рис. 43).

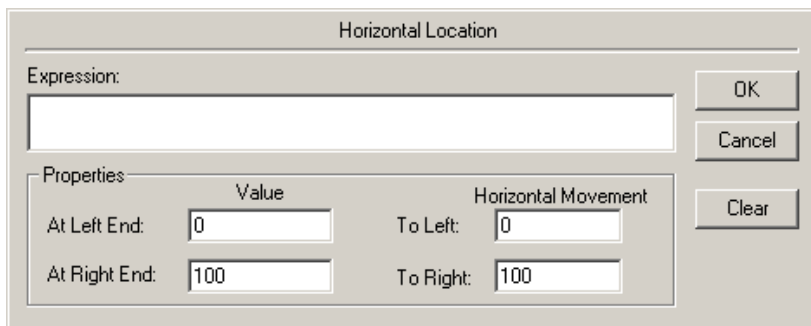


Рис. 43. Диалог настройки изменения положения объекта.

В поле **Expression** вводится аналоговый (**целый или действительный**) тэг или выражение, равное аналоговому значению.

В поле **At Left End** вводится значение тэга, при котором объект должен находиться в крайнем левом положении.

В поле **At Right End** вводится значение тэга, при котором объект должен находиться в крайнем правом положении.

В поле **To Left** вводится количество пикселей, на которое объект может передвигаться влево от исходного положения. В крайней левой позиции значение тэга будет равно значению, введенному в поле **At Left End**.

В поле **To Right** введите количество пикселей, на которое объект может передвигаться вправо от исходного положения. В крайней правой позиции значение тэга будет равно значению, введенному в поле **At Right End**.

*Создание связей процентной заливки **Percent Fill***

С помощью связей **процентной заливки** можно изменять уровень заливки контурных объектов (или символов, содержащих контурные объекты) в зависимости от значения аналогового тэга или выражения, имеющего аналоговое значение. Например, такая связь позволяет отображать уровень жидкости в емкости. Для объекта или символа можно определить связь горизонтальной заливки, вертикальной или и той, и другой одновременно (рис. 44).

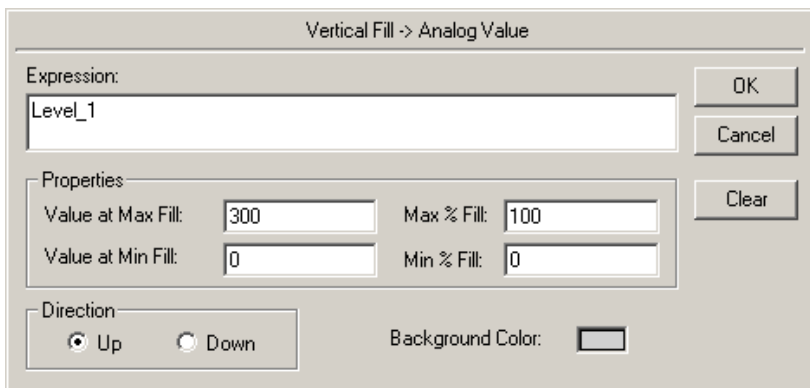


Рис. 44. Диалог настройки процентной заливки объекта.

В поле **Expression** вводится аналоговый (**целый или действительный**) тэг или выражение, равное аналоговому значению.

В поле **Value at Max Fill** вводится значение выражения, при котором заливка объекта должна достигать максимального уровня.

В поле **Value at Min Fill** вводится значение выражения, при котором заливка объекта должна достигать минимального уровня.

В поле **Max % Fill** вводите процент (0-100), которого должна достигать заливка объекта при значении выражения, заданном в поле **Value at Max Fill**.

В поле **Min % Fill** вводится процент (0-100), которого должна достигать заливка объекта при значении выражения, заданном в поле **Value at Min Fill**.

В группе **Direction** выбирается направление, в котором должна происходить заливка:

- **Up** - заливка будет происходить снизу вверх.
- **Down** - заливка будет происходить сверху вниз.

Поле **Background Color** выбирается цвет для **незалитой** части объекта. Цвет заливки выбирается при создании объекта. Если для объекта определить одновременно связи **вертикальной** и **горизонтальной процентной заливки**, то в качестве цвета фона будет использоваться последний цвет, выбранный в диалоговом окне любой из этих связей.

*Создание дополнительных типов связей **Miscellaneous***

Существует четыре дополнительных типа связей (рис. 45).



Рис. 45. Анимационные дополнительные типы связи

- **Видимость (Visibility)** - позволяет управлять степенью видимости объекта в зависимости от значения дискретного тэга или выражения (рис. 46).

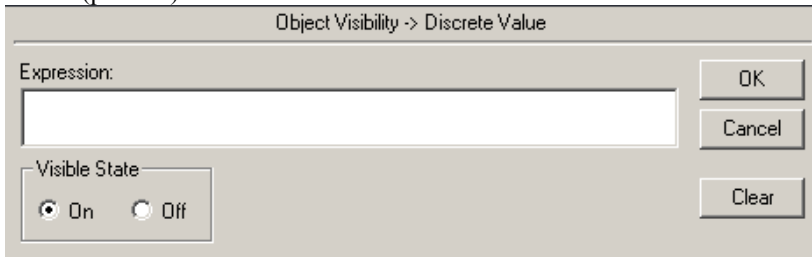


Рис. 46. Диалог настройки видимости объекта.

В поле **Expression** вводится дискретный тэг или выражение, равное дискретному значению.

Группа **Visible State** позволяет настроить видимость объекта. При выборе **On** объект станет невидимым, когда выражение будет иметь значение **true**. При выборе **Off** объект станет видимым, когда выражение будет иметь значение **true**.

- **Мерцание (Blinking)** - вызывает мерцание объекта при определенном значении дискретного тэга или выражения (рис. 47).

В поле **Expression** вводится дискретный тэг или выражение, равное дискретному значению.

Выбор **Blink Invisible** позволяет сделать, что объект или символ исчезали и вновь появляться на экране. Выбор **Blink visible with these attributes** означает, что объект или символ будут оставаться видимым на экране, а его мерцание будет осуществляться путем смены выбранных цветовых атрибутов.

Выбор **Blink Speed** позволяет настроить скорость мерцания.

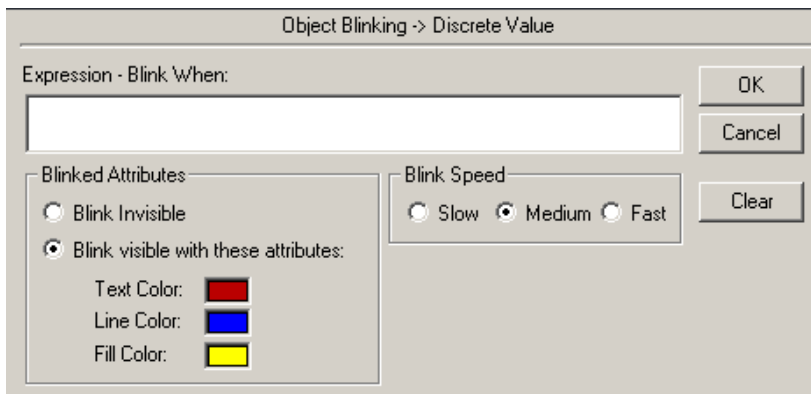


Рис. 47. Диалог настройки мерцания объекта.

- **Ориентация (Orientation)** - вызывает разворот объекта при определенном значении тэга или выражения (рис. 48).

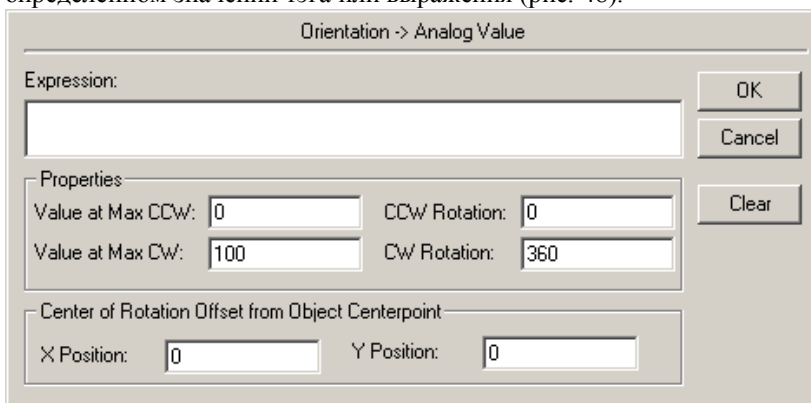


Рис. 48. Диалог настройки вращения объекта.

В поле **Expression** вводится аналоговый (**целый или действительный**) тэг или выражение, равное аналоговому значению.

В поле **Value at Max CCW** вводится значение выражения, при котором объект должен развернуться на максимальную позицию против часовой стрелки.

В поле **Value at Max CW** вводите значение выражения, при котором объект должен развернуться на максимальную позицию по часовой стрелке.

В поле **CCW Rotation** вводятся градусы, на которые объект должен повернуться против часовой стрелки при достижении значения **Value at Max CCW**.

В поле **CW Rotation** вводятся градусы, на которые объект должен повернуться по часовой стрелке при достижении значения **Value at Max CW**.

В поле **X Position** вводится количество пикселей, на которое центр вращения должен переместиться по горизонтали от центра объекта.

В поле **Y Position** вводится количество пикселей, на которое центр вращения должен переместиться по вертикали от центра объекта.

Анимационная связь вращение использует в качестве центра вращения центр объекта или символа.

- **Отключение (Disable)** - позволяет отключить сенсорную функцию объекта при определенном значении тэга или выражения (рис. 49).

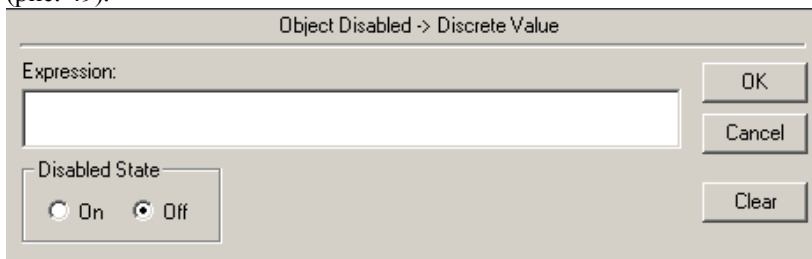


Рис. 49. Диалог настройки отключения объекта.

В поле **Expression** вводится дискретный тэг или выражение, равное дискретному значению.

Группа **Disabled State** позволяет настроить отключение объекта. Состояние **On** означает, что сенсорная функция объекта отключается, пока выражение имеет значение **true**. При выборе **Off** объект станет доступным, когда выражение имеет значение **true**.

*Создание связей отображения значения **Value Display***

Связи отображения значения позволяют использовать текстовые объекты для вывода значений дискретного, аналогового или текстового тэга (рис. 59).

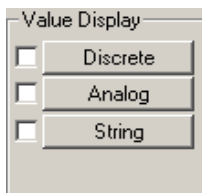


Рис. 50. Анимационные связи вывода значений на экран

Существует три типа таких связей:

- **Дискретное** - использует дискретное значение выражения для вывода определенного пользователем сообщения типа On или Off (рис. 51).

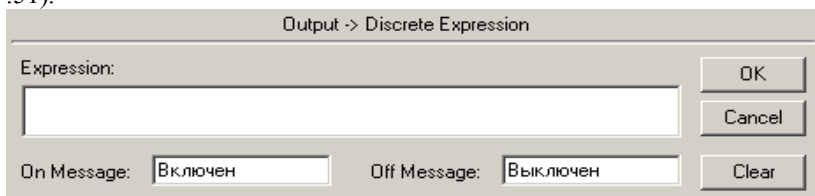


Рис. 51. Диалог настройки вывода на экран дискретного значения.

В поле **Expression** вводится дискретный тэг или выражение, равное дискретному значению.

В поле **On Message** вводится сообщение, которое должно выводиться, когда дискретное выражение будет равно 1 (True, On, Yes).

В поле **Off Message** вводится сообщение, которое должно выводиться, когда дискретное выражение будет равно 0 (False, Off, No).

- **Аналоговое** - выводит аналоговое значение выражения (рис. 52).

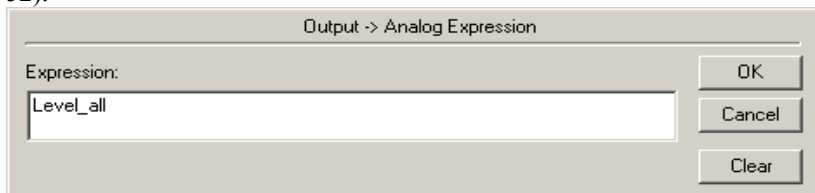


Рис. 52. Диалог настройки вывода на экран аналогового значения.

В поле **Expression** вводится аналоговый (**целый или действительный**) тэг или выражение, равное аналоговому значению. (В этом выражении можно также использовать дискретный тэг, который будет выводить только 1 или 0.)

- **Текстовое** - выводит текстовое значение выражения (рис. 53)

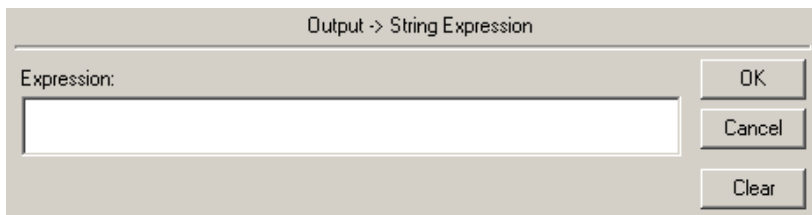


Рис. 53 Диалог настройки вывода на экран текстового значения.

В поле **Expression** вводится текстовый тэг или выражение, равное текстовому тэгу.

Содержание работы

1. В основном окне организовать ввод/вывод внутренних действительных тэгов для расходов, уровня и задания уровня и вывод целого тэга для времени и задания времени отстаивания и перемешивания.

2. Обеспечить вывод значений тэга целого или действительного типа, задаваемого с помощью программы моделирования DDE-сервера.

3. Организовать ввод/вывод символьной информации в поле комментария заданного внутреннего тэга (ввод осуществить через графический объект).

4. В основном окне задать следующие анимационные связи: степень заполнения бака в зависимости от значения тэга уровня жидкости; ползунок увеличения/уменьшения задания уровня для емкостей.

Методические указания по работе

1. Требуемые целые и действительные тэги создать либо с помощью кнопки *New* диалогового окна *Tagname Dictionary* (рис. 18), либо непосредственно при задании анимационных связей в поле *TagNameTagName*.

Необходимо создать следующие тэги для данного учебного задания.

Имя тэга (Tagname)	Тип тэга (Type)	Начальное значение (Initial Value)	Мин. значение (Min Value)	Макс. значение (Max Value)
Flow_1	Memory Real	45	0	300
Flow_2	Memory Real	36	0	300
Flow_3	Memory Real	125	0	300
Flow_4	Memory Real	24	0	300
Flow_5	Memory Real	69	0	300
Flow_6	Memory Real	195	0	300
Level_1	Memory Real	0	0	300
Level_2	Memory Real	0	0	600
Level_set_1	Memory Real	152	0	300
Level_set_2	Memory Real	264	0	600
Volume_all	Memory Real	0	0	32767
Time_1	Memory Int	0	0	32767
Time_2	Memory Int	0	0	32767
Time_3	Memory Int	0	0	32767
Time_set_1	Memory Int	8	0	32767
Time_set_2	Memory Int	12	0	32767
Time_set_3	Memory Int	7	0	32767

Для созданных тэгов организуем вывод на экран при помощи анимационной связи *User Inputs –Analog*.

2. Программа моделирования DDE-сервера находится по следующему пути «**d:\user\intoch\dde_test.exe**». Программа является генератором случайных чисел, внешний вид представлен на рис. 54.

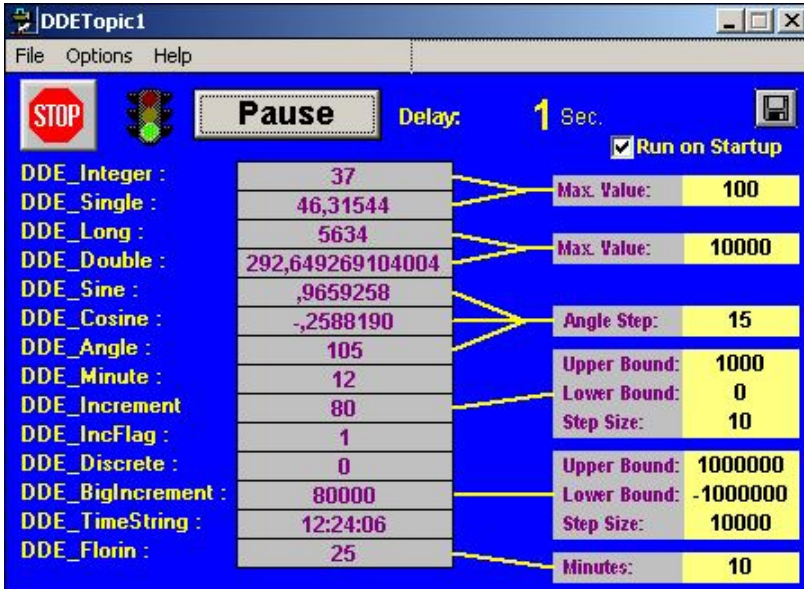


Рис. 54. Внешний вид программы DDE-сервера.

В своем проекте создайте тэг «dde_time», при помощи этого тэга получим значение времени из DDE-сервера. Для тэга зададим тип I/O Message, для типов I/O необходимо задать имя доступа (Access Name). В окне добавления имени канала доступа зададим:

- произвольное имя канала доступа (Access Name);
- сетевое имя (Node Name) оставим пустым;
- именем приложения (Application Name) будет являться имя программы без расширения: **dde_test**;
- именем группы (Topic Name) будет являться следующее:

DDETopic1.

Далее в диалоговом окне свойств тэга в поле Item нужно ввести имя переменной из программы моделирования DDE-сервера: **DDE_TimeString**.

3. Нарисуйте прямоугольник и задайте ему анимационную связь *User Inputs – String* и задайте *Tagname* для расхода: **Flow_1.Comment**. Внутри прямоугольника поместите какой-либо символ (текст) со связью *Value Display – String*.

4. Показатель уровня выполняется при помощи анимационной связи *Percent Fill – Vertical*, где именем тэга для первой емкости будет

тэг Level_1, а для второй емкости тэг Level_2, пример вертикального заполнения показан на рис. 55.

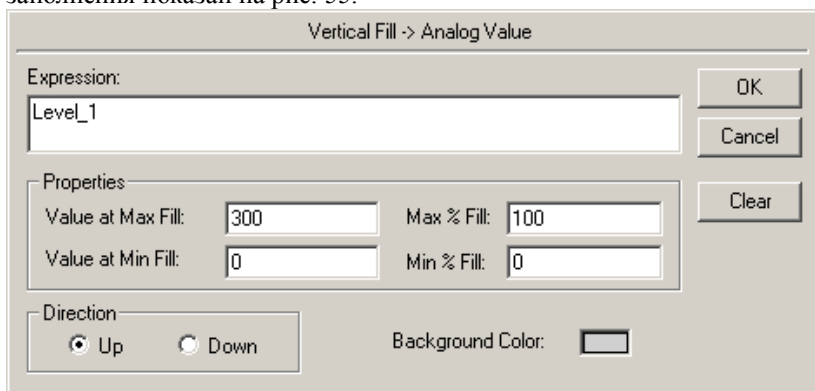


Рис. 55. Пример задания анимационной связи процентной заливки

Для создания ползунка сначала нарисуйте стрелку при помощи библиотеки Symbol Factory и становите ее как показано на рис. 56. Далее выберите анимационную связь Sliders и настройте ее на тэг Level_set_1 (для первой емкости).

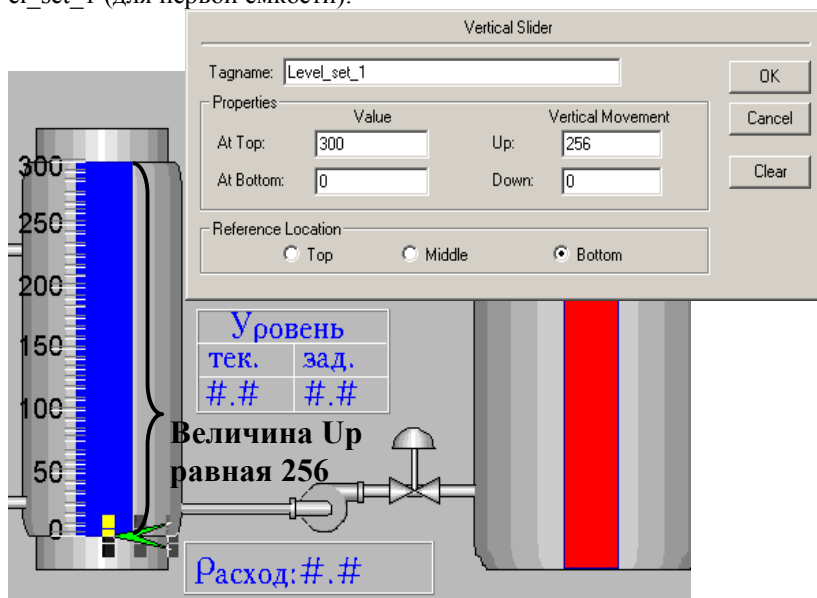


Рис. 56. Настройка анимационной связи Sliders.

Лабораторная работа № 4. Скрипты в InTouch.

Цель работы:

Получение начальных навыков работы с языком сценариев InTouch. Изучение особенностей программирования Quick-Function.

Теоретические положения

Типы скриптов Script.

Скрипты в InTouch – это программные фрагменты, активизируемые по событиям (по нажатию клавиши, кнопки, открытию окна, изменению значения переменной и т. д.). Порядок выполнения скриптов зависит от приложения.

В InTouch различают несколько типов скриптов.

- **Application Scripts** (скрипты уровня приложения) относятся ко всему приложению и используются для запуска других приложений, имитации технологических процессов, вычисления значений переменных и т.д. Существует три типа сценариев приложения:

- *On Startup* - выполняется однократно при запуске приложения;
- *While Running* - выполняется непрерывно с заданной частотой, пока приложение выполняется;
- *On Shutdown* - выполняется однократно при завершении работы приложения.

Когда выбирается сценарий *While Running*, становится активным поле *Every 0 Milliseconds*. В этом поле нужно ввести время в миллисекундах, через которое сценарий будет выполняться повторно.

- **Window Scripts** (скрипты уровня окна) связываются с конкретным окном. Существует три типа таких сценариев:

- *On Show* - выполняется однократно при открытии окна;
- *While Showing* - выполняется непрерывно через заданный интервал, пока окно открыто;
- *On Hide* - выполняется однократно при сворачивании окна.

- **Key Scripts** (клавишные скрипты) 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.). Существует три типа сценариев клавиши:

- *On Key Down* - выполняется один раз при первом нажатии клавиши;

- *While Down* - выполняется непрерывно с заданным интервалом, пока нажата клавиша;
- *On Key Up* - выполняется один раз при отпускании клавиши.
- **Touch Pushbutton Action Scripts** (скрипты, запускаемые кнопками) 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.
- **Condition Scripts** (скрипты по изменению логического выражения) связываются с логической переменной или выражением, которое будет принимать значения либо "истина", либо "ложь". Логические скрипты могут содержать в себе и аналоговые переменные.
Существует четыре типа сценария условия:
 - *On True* - выполняется однократно, когда условие становится истинным;
 - *On False* - выполняется однократно, когда условие становится ложным;
 - *While True* - выполняется непрерывно, пока условие остается истинным;
 - *While False* - выполняется непрерывно, пока условие остается ложным.
- **Data Change Scripts** (скрипты по изменению данных) 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.
- **ActiveX Event** (скрипты событий ActiveX) 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.

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. На рис. 57 приведен диалог Application Scripts (скрипты уровня приложения).

Редактор скриптов InTouch поддерживает два типа скриптов:

простые и сложные. Простые скрипты - это скрипты, содержащие операторы присваивания, сравнения, простые математические функции и т. д. Сложные скрипты позволяют выполнять различные логические операции типа IF - THEN - ELSE, а также могут включать циклы типа FOR - NEXT.

Справа, в поле Functions, размещены клавиши вызова списков различных групп встроенных функций. Доступ к спискам встроенных функций возможен также командой Insert/Functions с последующим выбором группы функций.

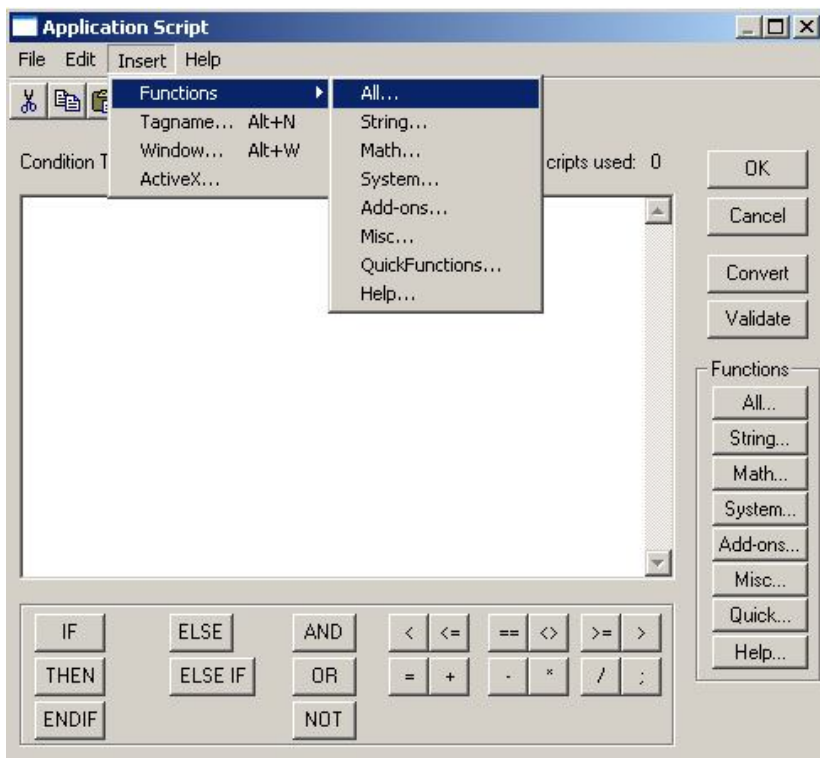


Рис. 57. Редактор скриптов Application Scripts (уровень приложения)

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;
- System... - системные функции;
- Misc... - функции для работы с алармами распределенных систем, трендами, печатью и др.

Вызов списка функций группы осуществляется нажатием соответствующей клавиши.

Использование Quick-функций

QuickFunctions хранятся в том приложении, в котором были созданы. Вызывая Quick-функции из других сценариев или выражений, вы можете создать сценарий один раз и использовать его повторно сколько угодно раз. Повторное использование этих сценариев снижает трудозатраты разработчика, поскольку отпадает необходимость в дублировании кода, не нужно копировать его и вставлять в другие сценарии. Повторяемый код хранится в одном сценарии и в одном месте, позволяя тем самым обновлять все экземпляры сценария за один сеанс редактирования. Quick-функции могут быть определены как асинхронные, т.е. при выполнении приложения они будут работать в фоновом режиме, не мешая основному процессу WindowViewer. Это позволяет WindowViewer выполнять особенно длительные операции, например, запросы к базе данных SQL, отдельно от основного хода программы. В качестве имен аргументов Quick-функций нельзя использовать имена тэгов. Имена тэгов имеют более высокий приоритет, чем такие же имена аргументов, и потому сценарий будет выполняться некорректно. Имя аргумента не влияет на счетчик использования тэгов, поскольку рассматривается системой как локальная переменная. При вызове Quick-функции в поле Expression должен содержаться оператор CALL вызова функций, каждая из которых, в свою очередь, должна иметь в качестве последнего оператора RETURN для возврата результата в вызывающее выражение. Когда встречается RETURN, выполнение Quick-функции немедленно прерывается. Возвращаемые данные могут быть дискретного, целого и действительного типа.

Содержание работы

Обеспечить автоматический режим работы объекта, состоящий из следующих этапов.

1 Заполнение емкости 1 из двух трубопроводов. Заполнение емкости начинается после открытия одного из клапанов.

2 При достижении заданного оператором уровня в емкости 1, начинается режим отстаивания.

3 После окончания заданного времени реакции, начинается опорожнение емкости 1 в емкость 2 при помощи клапана 3, и одновременно открываются клапана подачи компонентов сверху емкости 2.

4 При достижении заданного оператором уровня в емкости 2, начинается режим перемешивания.

5 После окончания заданного времени перемешивания, начинается режим отстаивания в емкости 2.

6 После окончания заданного времени реакции, начинается опорожнение емкости 2 при помощи клапана 6. После опорожнения емкости 2 начинается заново заполнение емкости 1.

7 По формуле зависимости объема заполнения бака от уровня наполнения с помощью *QuickFunction* вычислить объем заполнения емкостей 1 и 2, а также общий объем готовой продукции за все циклы работы объекта.

8 Для клапанов задать открытие и закрытие и цветом показать работу клапана.

Методические указания по работе

1. Для модели работы объекта используйте скрипты типа *Condition*.

Сценарий по условию <i>Condition</i>
Условие: <i>Klapan_1</i>
<pre>On False: Flow_1_old = Flow_1; Flow_1 = 0; On True: Flow_1 = Flow_1_old; While True 100 Msec: Level_1 = Level_1 + Flow_1/600; IF Level_1 >= Level_set_1 THEN Klapan_1 = 0; Klapan_2 = 0; Start_ots_1 = 1; ENDIF;</pre>

Условие: Клапан_2

On False:
Flow_2_old = Flow_2;
Flow_2 = 0;
On True:
Flow_2 = Flow_2_old;
While True 100 Msec:
Level_1 = Level_1 + Flow_2/600;
IF Level_1 >= Level_set_1 THEN
 Klapan_1 = 0;
 Klapan_2 = 0;
 Start_ots_1 = 1;
ENDIF;

Условие: Start_ots_1

On True:
Time_1 = 0;
While True 1000 Msec:
Time_1 = Time_1 + 1;
IF Time_1 >= Time_set_1 THEN
 Start_ots_1 = 0;
 Klapan_3 = 1;
 Klapan_4 = 1;
 Klapan_5 = 1;
ENDIF;

Условие: Клапан_3

On False:
Flow_3_old = Flow_3;
Flow_3 = 0;
On True:
Flow_3 = Flow_3_old;
While True 100 Msec:
Level_2 = Level_2 + Flow_3/600;
Level_1 = Level_1 - Flow_3/600;
IF Level_1 <= 1 THEN
 Klapan_3 = 0;
ENDIF;

Условие: Клапан_4
On False: Flow_4_old = Flow_4; Flow_4 = 0; On True: Flow_4 = Flow_4_old; While True 100 Msec: Level_2 = Level_2 + Flow_4/600;
Условие: Клапан_5
On False: Flow_5_old = Flow_5; Flow_5 = 0; On True: Flow_5 = Flow_5_old; While True 100 Msec: Level_2 = Level_2 + Flow_5/600;
Условие: Level_1 >= Level_set_1
On True: Start_ots_1 = 1; While True 100 Msec: Klapan_1 = 0; Klapan_2 = 0;
Условие: Level_2 >= Level_set_2
On True: Start_ots_2 = 1; While True 100 Msec: Klapan_3 = 0; Klapan_4 = 0; Klapan_5 = 0;
Условие: Start_ots_2
On True: Time_2 = 0; While True 1000 Msec: Time_2 = Time_2 + 1; IF Time_2 >= Time_set_2 THEN Start_ots_2 = 0; Start_ots_3 = 1; ENDIF;

Условие: Start_ots_3
On True: Time_3 = 0; While True 1000 Msec: Time_3 = Time_3 + 1; IF Time_3 >= Time_set_3 THEN Start_ots_3 = 0; Klapan_6 = 1; ENDIF;
Условие: Klapan_6
On False: Flow_6_old = Flow_6; Flow_6 = 0; On True: Flow_6 = Flow_6_old; While True 100 Msec: Level_2 = Level_2 - Flow_6/600; Level_all = Level_all + Flow_6/600; IF Level_2 <= 1 THEN Klapan_6 = 0; Klapan_1 = 1; Klapan_2 = 1; ENDIF;

2. Создайте новую *Quick*-функцию с одним действительным аргументом (имя аргумента не должно совпадать с именем какого-либо тэга). Для возврата в вызывающий сценарий используйте оператор вида: «RETURN Аналоговое выражение для расчёта объёма;». В вызывающем сценарии используйте оператор следующего вида: «Тэг объема = CALL ИмяФункции(Тэг уровня);».

Пример:

Quick Function	
Функция – Volume	Аргумент – TankLevel (Real)
RETURN PI()*0.5*0.5* TankLevel	

3. На клапан задайте анимационную связь *Touch Pushbuttons – Discrete* и в поле *Tagname* введите тэг работы клапана: **Klapan_1**, с условием *Action - Toggle*. Для цветовой визуализации используйте анимационную связь *Fill Color– Discrete* и задайте *Tagname: Klapan_1*.

Лабораторная работа № 5. Алармы и события.

Цель работы:

Изучение методов поддержки алармов и событий в InTouch (типы, группы, приоритеты, задание условий для тэгов, поля, квитирование, библиотечные функции). Настройка InTouch с целью создания регистрационных файлов (журналов). Настройка стандартных экранов алармов или экранов распределенных алармов для локальных оповещений.

Теоретические положения

Алармы и события в InTouch

В системе InTouch действуют два вида оповещений, информирующих оператора о протекании процесса: алармы и события. Алармы представляют собой предупреждения о тревожных состояниях процесса, которые требуют отклика оператора. Типичный случай генерации аларма – превышение каким-то параметром процесса предела, определенного пользователем, например, когда значение выходит за верхний пороговый уровень. Это вызывает состояние неподтвержденного (неквитированного) аларма, который используется для уведомления оператора о проблеме. InTouch может также зарегистрировать этот аларм в файле, который хранится на диске, или/и отправить его на принтер. Если оператор подтверждает аларм, то система переводит аларм в квитированное состояние. События – это сообщения о нормальном состоянии системы, которые не требуют отклика оператора. Типичный случай генерации события – вход оператора в систему InTouch. Такое событие также может быть зарегистрировано в файле на диске или (и) отправлено на принтер.

Типовые алармы

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). По умолчанию дискретный аларм может срабатывать на ON или OFF.

Аналоговые алармы базируются на анализе выхода значений переменной за указанные верхние и нижние пределы. Аналоговые алармы могут быть заданы в нескольких комбинациях:

- High и High High (верхний и выше верхнего);

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

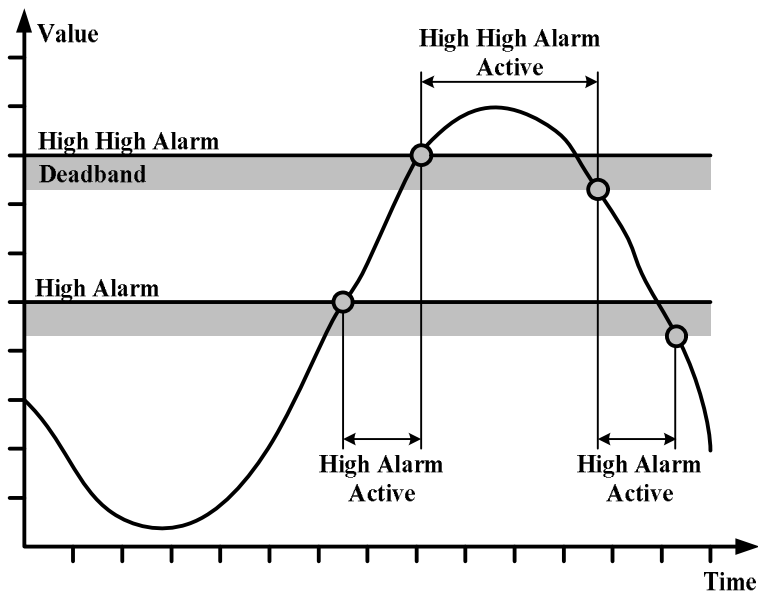


Рис. 58. Графическая интерпретация алармов типа Hi и HiHi.

Из рис. 58 видно, что алармы Hi и HiHi срабатывают при достижении переменной заданных для каждого аларма пределов (High Alarm, High High Alarm). Для выхода переменной из состояния аларма (HiHi или Hi) необходимо, чтобы ее значение стало меньше порогового на величину, называемую зоной нечувствительности (Deadband). Аналогично можно интерпретировать алармы типа Lo и LoLo.

Для аларма типа Deviation, срабатывание аларма будет происходить при отклонении значения переменной от заданного значения (Setpoint), причем это заданное значение в ходе технологического процесса может изменяться либо оператором, либо программно (автоматически). Аларм срабатывает при выходе значения переменной за границу предельно допустимого отклонения.

Алармы типа ROC срабатывают, когда скорость изменения параметра становится больше предельно допустимой. Понятие зоны нечувствительности (Deadband) к алармам этого типа не применяется.

Типы алармов и событий

В зависимости от своих характеристик алармы подразделяются на несколько категорий по типу (Type) и классу (Class).

Алармы	Стандартный тип	Распределенный класс	Распределенный тип
Discrete	DISC	DSC	DSC
Deviation - Major	LDEV	DEV	MAJDEV
Deviation - Minor	SDEV	DEV	MINDEV
Rate - of - Change	ROC	ROC	ROC
SPC	SPC	SPC	SPC
Value - LoLo	LOLO	VALUE	LOLO
Value - Lo	LO	VALUE	LO
Value - High	HI	VALUE	HI
Value - HiHi	HIHI	VALUE	HIHI

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

События в InTouch также делятся в зависимости от их характеристик на несколько общих категорий (Event Types).

Тип	Событие
ACK	Аларм был подтвержден
ALM	Возникла аварийная ситуация
EVT	Возникло аварийное событие
RTN	Переменная перешла из аварийного состояния в обычное
SYS	Возникло системное событие
USER	Изменение значения переменной \$Operator
DDE	Получено значение переменной от DDE - клиента
LGC	Скрипт изменил значение переменной
OPR	Оператор ввел новое значение переменной

Первые шесть событий выбираются автоматически при разрешении регистрации событий. Для остальных трех событий разрешение регистрации устанавливается при определении переменной в словаре переменной.

Каждый сконфигурированный в InTouch аларм имеет определенное для него значение приоритета. Приоритет обозначает степень критичности (важности) аларма и может варьироваться в пределах от 1 до 999, причем 1 - максимальная важность. На основе диапазона приоритетов можно создать связи анимации, сценарии квитиования, а также использовать фильтры для выборочного просмотра и печати алармов.

Например, если для производственного процесса необходимы четыре степени критичности, то можно выделить следующие диапазоны приоритетов:

- критический (1 ... 249),
- значительный (250... 499),
- незначительный (500 ... 749),
- рекомендательный (750 ... 999).

Группы алармов

Каждый аларм связан 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. Группы алармов являются полезным средством фильтрации вывода информации об алармах на экран дисплея или принтер.

Каждая переменная связывается с какой-либо группой алармов. 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.

Древовидная структура может иметь до восьми уровней, при этом каждая входящая в дерево группа может иметь до 16 подгрупп (рис. 59).

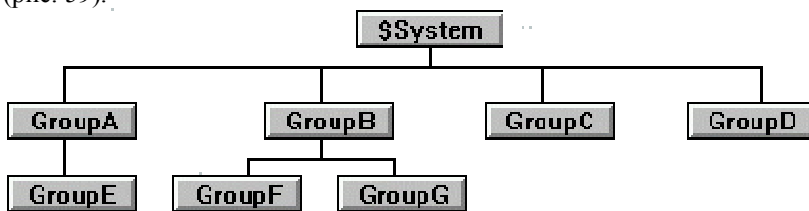


Рис. 59. Иерархическая древовидная структура групп алармов.

Для создания таких групп в меню окна WindowMaker предусмотрена команда `Special/Alarm Groups` (группы алармов), вызывающая появление диалога `Alarm Groups` (рис.60). При определении переменных в словаре `Tagname Dictionary` нажатие кнопки `Group` (см. рис.18) также выводит на экран этот диалог.



Рис. 60. Диалог Alarm Group (группы алармов).

Воспользовавшись кнопкой Add, можно добавить группу алармов, а также формировать древовидную структуру системы алармов, определяя родительские группы и группы-потомки. При этом открывается диалог (рис.61) Add Alarm Group (добавить группу алармов). Кнопка Parent Group (родительская группа) предназначена для выбора родительской группы в древовидной структуре. В диалоге предусмотрено поле Comment (комментарий) для ввода необязательного текста, комментирующего данную группу.

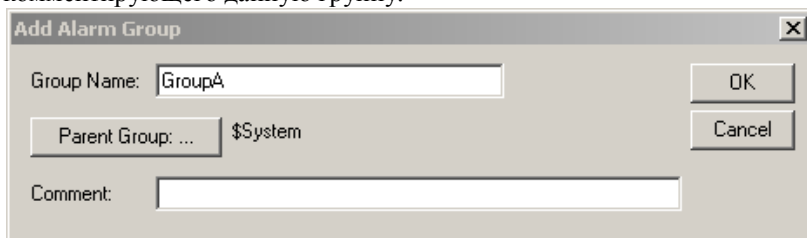


Рис. 61. Диалог Add Alarm Group (добавить группу алармов).

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).

С помощью объекта "Текущие алармы" на экран дисплея выводится информация только о текущих подтвержденных или неподтвержденных аварийных ситуациях. В случае возврата ситуации в нормальное состояние запись о ней исчезает из текущей аварийной сводки.

С помощью объекта "Архивная сводка алармов" на дисплей выводятся данные об аварийных ситуациях или событиях, включая количество уже произошедших аварийных ситуаций данного типа, время подтверждения, время возврата в нормальное состояние.

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).

Для создания объекта вывода алармов следует сначала вывести на экран диалоговое окно Wizard Selection (Выбор мастера). Далее производится выбор категории Alarm Displays (окна вывода алармов) в списке мастеров, в категории выбирается стандартная система алармов (Standard Alarm Displays).

MM/DD	HH:MM:SS	EVT	Type	Pri	Name	Group
MM/DD	HH:MM:SS	EVT	Type	Pri	Name	Group
MM/DD	HH:MM:SS	EVT	Type	Pri	Name	Group
MM/DD	HH:MM:SS	EVT	Type	Pri	Name	Group
MM/DD	HH:MM:SS	EVT	Type	Pri	Name	Group

Рис. 62. Стандартный объект вывода аварийной информации.

Конфигурирование окна вывода аварийной информации производится в диалоге Alarm Configuration (параметры окна вывода аварийной информации). Вызов этого диалога производится командой Special/Animation Links меню WindowMaker (рис. 63). Быстрый доступ к этому диалогу можно получить, воспользовавшись меню правой кнопки мыши с последующим щелчком на строке Properties.

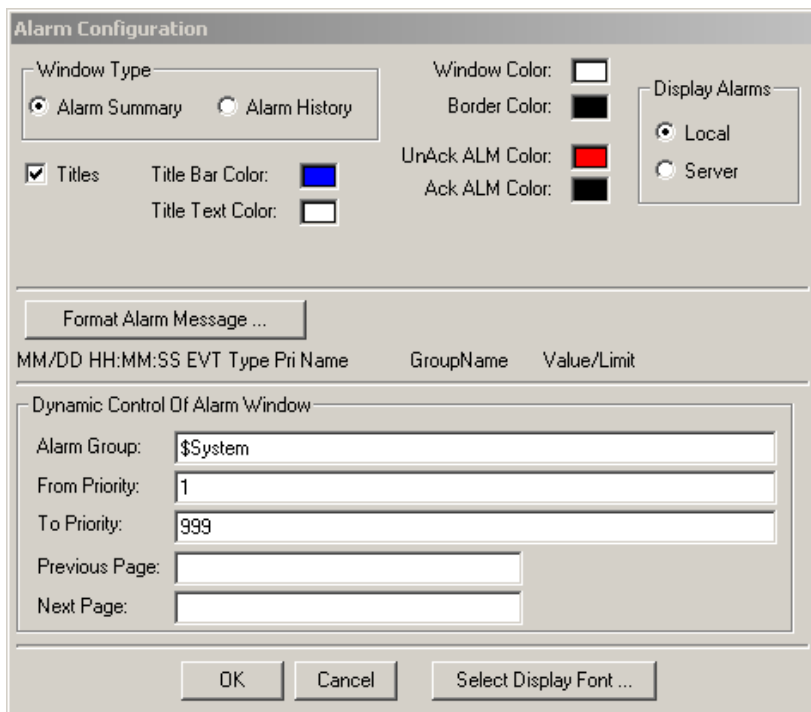


Рис. 63. Диалог Alarm Configuration.

В этом диалоге определяется тип окна вывода аварийной информации ("Текущие алармы" или "Архивная сводка алармов"), группа алармов (Alarm Group), границы диапазона приоритетов окна вывода алармов (From/To Priority), дискретные переменные для перехода на предыдущую (Previous Page) и следующую (Next Page) страницу списка алармов. Для выбора шрифтов следует воспользоваться кнопкой Select Display Font.

Нажатие кнопки Format Alarm Message (форматирование аварийного сообщения) выводит на экран одноименный диалог (рис.64), где определяется информация, включаемая в аварийное сообщение.

Format Alarm Message

Date MM/DD MMM DD MM/DD/YY MMM DD YYYY
 DD/MM DD MMM DD/MM/YY DD MMM YYYY

Time 24 Hour AM/PM HH MM SS MSec

Event (ACK, RTN, ALM, EVT)

Alarm Type (HIHI, SDEV, OPR, etc.)

Operator Length:

Priority

Comment Length:

Tagname Length:

Group Name Length:

Value Length:

Limit Length:

Alarm State (UNACK_ALM, ACK_ALM, etc.)

MM/DD HH:MM:SS EVT Type Pri Name GroupName Value/Limit

Рис. 64. Диалог Format Alarm Message.

В строку аварийного сообщения можно включить текущую дату (Date), текущее время (Time), тип аларма (Alarm Type), приоритет (Priority), имя переменной (Tagname), ее текущее значение (Value), а также группу алармов (Group Name) и статус аларма (Alarm State).

Конфигурирование стандартной системы алармов

В соответствии с алгоритмом настройки системы алармов InTouch следующий этап предполагает настройку системы алармов в целом, т. е. определение общих свойств системы, а также свойств регистрации и печати алармов. Для входа в диалог конфигурирования стандартной системы алармов следует воспользоваться командой Special/Configure/Alarms либо в группе Configure окна Application Explorer дважды щелкнуть на строке Alarms.

Здесь возможно определить различные параметры стандартной системы алармов:

- количество записей об аварийных ситуациях, которые одновременно будут находиться в буфере алармов;

- размер буфера печати подключенного к параллельному порту принтера;
- период времени в миллисекундах, через который WindowViewer будет периодически обращаться к принтеру;
- поведение окна при добавлении нового аварийного сообщения к списку;
- разрешение регистрации событий, связанных с изменением данных в результате операций ввода/вывода, действий оператора, скрипта или системы и т. д.

Параметры регистрации алармов/событий

Кроме возможности отображения информации об аварийных ситуациях на экране дисплея InTouch позволяет сохранять ее на жестком диске компьютера. Регистрационный файл является обычным ASCII-файлом и может впоследствии обрабатываться любым текстовым редактором. Генерация файлов, определение максимальной длины, вид регистрируемой информации, срок хранения регистрационных файлов на диске и другие параметры задаются пользователем.

Для определения параметров регистрации в файле надо щелкнуть на закладке Logging (Регистрация) диалога Alarm Properties.

Параметры печати

В дополнение к выводу информации об аварийных ситуациях на экран дисплея и в регистрационный файл на диск возможен вывод ее и на печать. Содержание выводимой информации определяется пользователем на закладке Printing (Печать) диалога Alarm Properties.

Во время печати информации об аварийных ситуациях порт принтера находится под полным контролем InTouch, поэтому для этой цели необходимо использование отдельного принтера. Никакое другое приложение, включая операционную систему, не сможет пользоваться этим принтером в обычном режиме, пока не будет запрещен вывод аварийной информации.

На этой закладке можно определить следующие параметры печати:

- порт, к которому подключен принтер;
- параметры этого порта (скорость передачи, вид контроля четности, разрядность данных, количество стоповых битов);
- формат аварийных сообщений (все отображаемые, записываемые на диск и печатаемые сообщения форматируются одинаковым образом);
- группу аварийных ситуаций;
- значение приоритета регистрируемой аварийной ситуации.

Распределённая система алармов

Стандартную систему алармов рекомендуется использовать для идентичных InTouch-приложений. Распределенная система расширяет возможности стандартной и позволяет подтверждать аварийные ситуации, генерируемые системами алармов других включенных в сеть InTouch-приложений.

Распределенная система имеет следующие характеристики:

- возможность отображения и подтверждения алармов любого InTouch-узла сети;
- новый объект вывода с линейками прокрутки, настраиваемой шириной столбцов, возможностью выбора группы алармов, панелью состояния, динамическими типами вывода и различными цветами для разных приоритетов алармов;
- функции QuickScript, реализующие динамическое управление отображением и подтверждением алармов;
- механизм группирования, обеспечивающий одновременное обращение к нескольким контрольным группам разных приложений по одному имени;
- возможность добавления комментариев к аварийной информации при подтверждении алармов.

Содержание работы

1. Для тэга уровня жидкости в первой емкости задать аларм типа *Rate of Change*, а для соответствующего тэга второй емкости – алармы типа *Hi* и *HiHi*. Тэги уровня отнести к специальной группе алармов. С тэгами расходов жидкости связать аларм типа *Hi*. Ввести аларм работы клапанов.

2. При возникновении нового аларма в системе в основном окне вывести предупреждающее сообщение и обеспечить возможность квитирования аларма (рис. 65).



Рис. 65. Окно приложения с возможностью квитирования тэгов.

3. В окне «Меню» создать экран вывода текущих сообщений (рис. 66).

	Процесс	MM/DD HH:MM:SS EVT Type Pri Name	GroupName	Value/Limit
	Истор. алармы	MM/DD HH:MM:SS EVT Type Pri Name	GroupName	Value/Limit
	Тренды реал.	MM/DD HH:MM:SS EVT Type Pri Name	GroupName	Value/Limit
	Тренды истор.	MM/DD HH:MM:SS EVT Type Pri Name	GroupName	Value/Limit

Рис. 66. Окно вывода на экран текущих алармов.

В новом окне «Alarm» создать экран просмотра архива алармов и обеспечить просмотр оператором всех сообщений и сортировку выводимых сообщений (рис. 67).

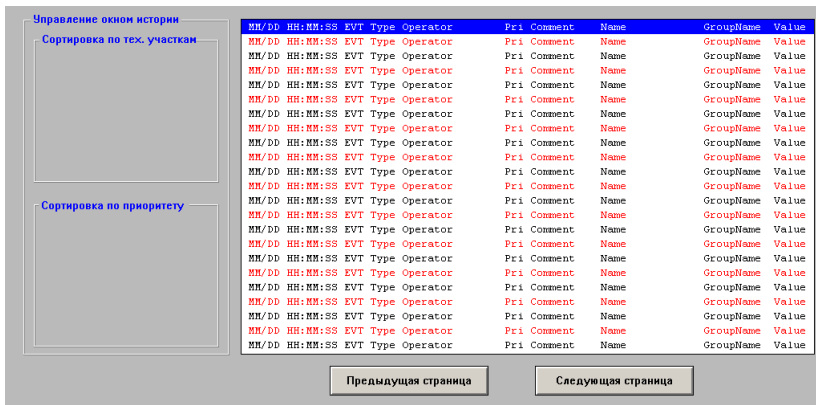


Рис. 67. Окно вывода на экран исторических алармов.

4. Обеспечить запись алармов в регистрационные файлы InTouch.

Методические указания по работе

1. Отрегулируйте состояние алармов для указанных тэгов так, чтобы данные состояния достигались в автоматическом режиме работы объекта. В словаре тэгов задайте условие возникновения аларма:

Level_1 – условие **Rate of Change** = 50% в минуту.

Level_2 – условие **Hi** = 550 и **HiHi** = 580.

Klapan_1 ... 6 – условие **Alarm State** = On.

Для тэгов, относящихся с первой емкости алармы отнести к группе «*Group_1*», а тэги для второй емкости отнести к группе «*Group_2*»

2. Предупреждающее сообщение можно выводить в текстовом виде или посредством какого-либо соответствующего графического изображения. Данное сообщение свяжите с системным тэгом *\$NewAlarm*, используя анимационные связи *Visibility* и *Blink*. Создайте кнопку с анимационными связями *Action* и *Visibility*. В *Action* необходимо обнулить значение тэга *\$NewAlarm* и с помощью функции *Ack* реализовать квитирование аларма.

Анимационная связь Action
Ack \$System; \$NewAlarm = 0;

3. Экран текущих сообщений создается при помощи мастера *Standard Alarm Display* (кнопка *Wizards*, раздел *Alarm Display*). Для

настройки вывода алармов необходимо правой кнопкой нажать на окно алармов и выбрать *Properties*. Появится окно *Alarm Configuration*. Вывод текущих значений алармов определяется выбором *Alarm Summary*. С помощью кнопки *Format Alarm Message* возможно отформатировать сообщения об аларме. При выводе текущих значений выберите: *Time, Event, Alarm Type, Operator, Comment, Tagname, Value*.

Необходимо создать окно «Alarm» с аналогичными параметрами окна «Main».

Экран архивных сообщений создается при помощи мастера *Standard Alarm Display* (кнопка *Wizards*, раздел *Alarm Display*). Вывод архивных значений алармов определяется выбором *Alarm Summary*. С помощью кнопки *Format Alarm Message* отформатируйте сообщения об аларме: *Date, Time, Event, Alarm Type, Operator, Priority, Comment, Tagname, Group Name, Value*. Необходимо задать два тэга типа *Memory Discrete* для пролистывания дневника алармов (поля *Previous Page* и *Next Page*). Создайте кнопки «Предыдущая страница», «Следующая страница» с анимационной связью *Touch Pushbuttons – Discrete Value*. В поле *Tagname* необходимо ввести имя дискретного тэга, которое было ранее введено в поле *Previous Page* и *Next Page*.

В рабочем окне организуйте операторский ввод необходимых параметров для этой функции, которые обеспечивает следующее: выбор группы алармов; сортировку алармов по приоритету.

Необходимо задать анимационные связи для кнопок «Процесс» и «Истор. алармы». Для кнопки «Процесс» зададим свойство открытия окна «main», при помощи анимационной связи *Show Window*, а для кнопки «Истор. алармы» зададим открытие окна «alarm».

4. В InTouch для регистрации алармов в файле на диске в меню *Special* или в менеджере приложения нажмите *Configure – Alarms*. В закладке *Logging* необходимо отметить *Logging Enabled* и указать путь к специальной папке для регистрации алармов (данная папка должна быть предварительно создана средствами MS Windows).

Лабораторная работа № 6. Тренды архива и реального времени.

Цель работы:

Организация архивирования данных в InTouch. Настройка мастер-объектов для работы с архивными (историческими) трендами. Создание, конфигурирование и отображение трендов реального времени. Использование косвенных типов тэгов в трендах реального времени.

Теоретические положения

Тренды в SCADA-системах

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

Тренды реального времени (Real Time) отображают динамические изменения параметра 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) после того, как данные будут записаны на диск, и можно будет использовать режим прокрутки предыдущих значений назад с целью посмотреть прошлые значения. Отображаемые данные тренда в таком режиме будут неподвижны и будут отображаться только за определенный период.

Архивирование данных в InTouch

При работе системы в режиме WindowViewer (среда исполнения) InTouch может производить запись значений переменных в регистрационный файл. Для того, чтобы архивирование переменной выполнялось, необходимо включить опцию *Log Data* (регистрация данных) при определении переменной в диалоге Tagname Dictionary (рис. 68).

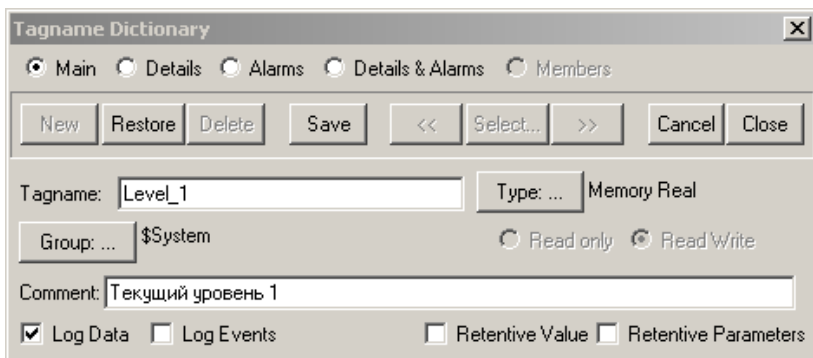


Рис.68. Диалог Tagname Dictionary с отмеченной опцией *Log Data*.

Запись в регистрационный файл производится всякий раз при изменении переменной на величину, превышающую порог для архивирования (*Log Deadband*), и по умолчанию один раз в час, если значение переменной за это время не изменилось. Поле *Log Deadband* находится в диалоге детального описания целой или вещественной переменной (рис.69).

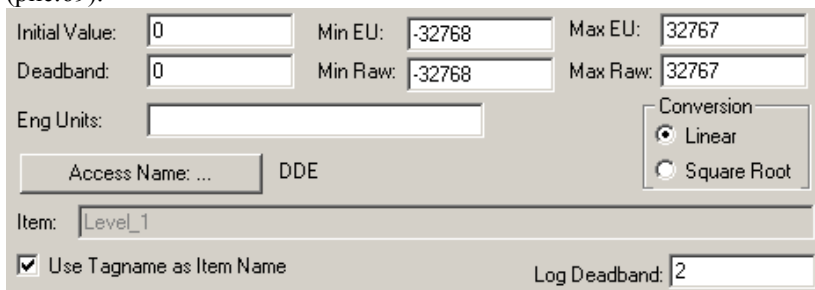


Рис.69 Диалог детального описания вещественной переменной.

Чтобы значения переменных, для которых опция *Log Data* разрешена, записывались в регистрационные файлы, необходимо общее разрешение глобальной функции регистрации. Его задают в диалоге *Historical Logging Properties* (параметры архивирования, рис. 70), который вызывается на экран командой *Special/Configure/Historical Logging*. В этот диалог можно также войти из окна *Application Explorer*.

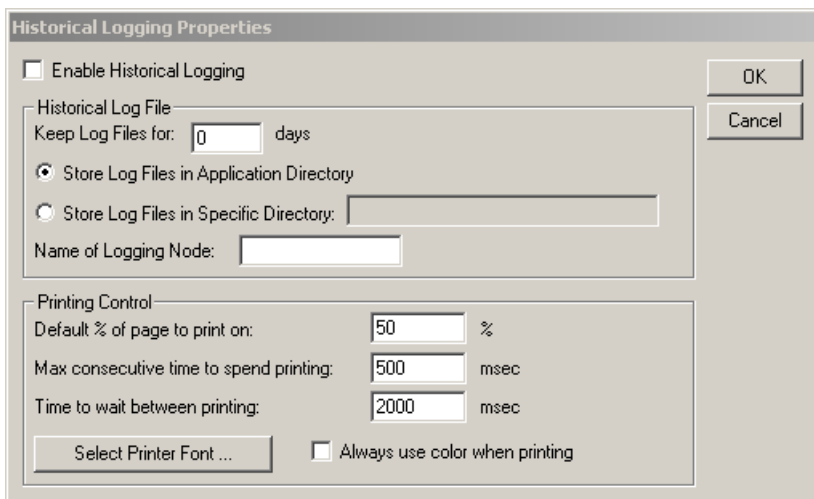


Рис.70. Диалог Historical Logging Properties.

Включение опции *Enable Historical Logging* дает общее разрешение на регистрацию значений переменных. Срок хранения регистрационных файлов на диске (исключая текущий день) определяется в поле *Keep Log Files for* в днях. Если в это поле введено значение 0, файлы будут храниться бесконечно долго. Регистрационные файлы могут быть размещены в каталоге приложения (опция по умолчанию *Store Log Files in Application Directory*). В противном случае следует отметить опцию *Store Log Files in Specific Directory* (хранить файлы в ином каталоге) и ввести полный путь до каталога, в котором будут храниться регистрационные файлы (при работе с распределенными архивами – полный сетевой путь).

Версия InTouch 7.0 (7.1) создает регистрационные файлы с расширением .LGH и .IDX. По умолчанию имена этих файлов имеют следующий формат:

YYMMDD00.LGH и YYMMDD00:IDX,

где:

- YY, MM, DD - соответственно, год, месяц и день создания файла;

- 00 - всегда нули.

Кроме того, в этом же диалоге определяются параметры печати графиков.

Тренды реального времени в InTouch

Тренды реального времени являются динамическими. Во время выполнения приложения они постоянно обновляются, позволяя вычерчивать изменения четырех тэгов или выражений. Способность присваивать выражение перу тренда полезна при создании разнообразных отображений, показывающих тэги с широко изменяющимся диапазоном значений. Первоначально, при вставке объекта тренда реального времени используются системные параметры настройки по умолчанию. После настройки указанного объекта следующий тренд реального времени будет исходно настроен с ранее заданными вами параметрами.

Свойства тэгов *muna Indirect*

Косвенные типы тэгов (*Indirect*) позволяют создать, к примеру, один объект тренда реального времени и динамически переназначать тэги этого объекта для нескольких источников. Когда вы приравниваете косвенный тэг к тэгу другого источника, косвенный тэг и тэг источник становятся точными копиями друг друга во всех отношениях, включая поля, сценарии, и т.д. Изменение значения тэга источника отражается на косвенном тэге. При изменении значения косвенного тэга соответственно изменяется и тэг источник. Значения косвенного тэга можно определить в базе данных как сохраняемые и перенастроить их так, чтобы при запуске они принимали свое последнее значение. Косвенные тэги назначаются при помощи поля *Name*.

Отображение трендов

Создание тренда реального времени.

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 (рис. 71).

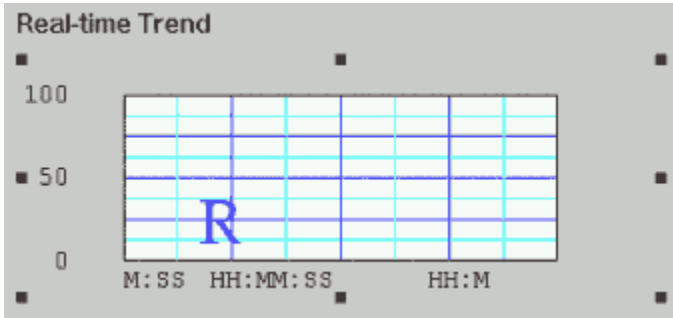


Рис.71. Объект "тренд реального времени".

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).

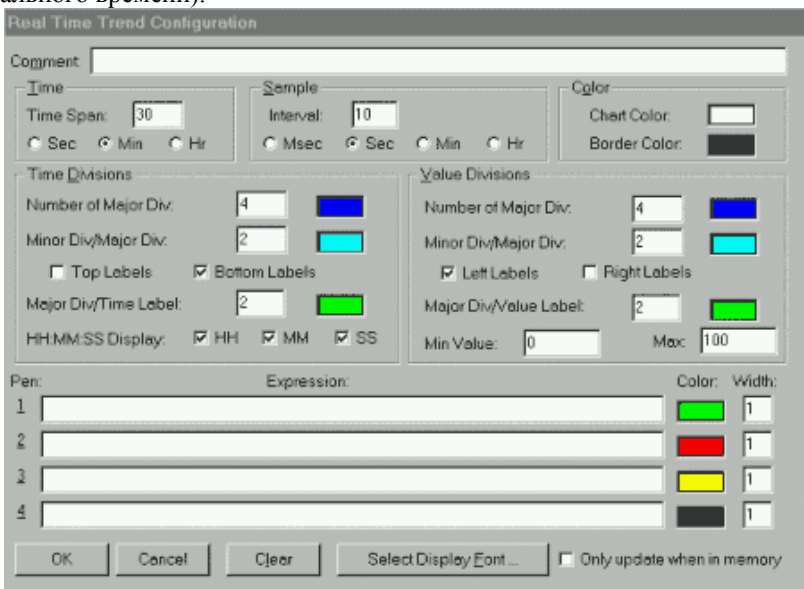


Рис.72. Диалог Real Time Trend Configuration.

Среди настроек этого диалога можно отметить диапазон времени, охватываемый трендом (*Time Span*), частоту вывода значение переменной (*Interval*), разрешение сетки по большим и малым делением горизонтальной и вертикальной осей (*Time Division, Value Division*), цвета фона и рамки графика (*Color*). Конфигурирование перьев тренда включает выбор имени переменной или выражения, цвета и толщины линии для каждого пера (поле *Expression*).

Для повышения производительности системы следует отметить опцию *Only update when in memory* (обновлять, когда в памяти). В этом случае обновление данных тренда будет производиться только в моменты, когда окно с трендом отображается на дисплее (находится в RAM).

Существуют следующие способы повышения производительности при работе с трендами реального времени: уменьшение толщины линии графика, уменьшение частоты вывода значений переменной. Например, если установлен диапазон времени (*Time Span*) в 30 минут, а частота вывода - 2 секунды, то число измерений, которые нужно провести за каждые 30 минут, будет равно 900 ($30 * 60/2 = 900$). При частоте выводе в 5 секунд число измерений существенно уменьшается: $30 * 60/5 = 360$.

Создание архивного тренда.

При конфигурировании архивного тренда можно создать "визуры" (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. 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).

Для создания архивного тренда необходимо: запустить мастер создания объектов Wizards; выбрать в списке категорию Trends; этот диалог будет иметь следующий вид (рис. 73).

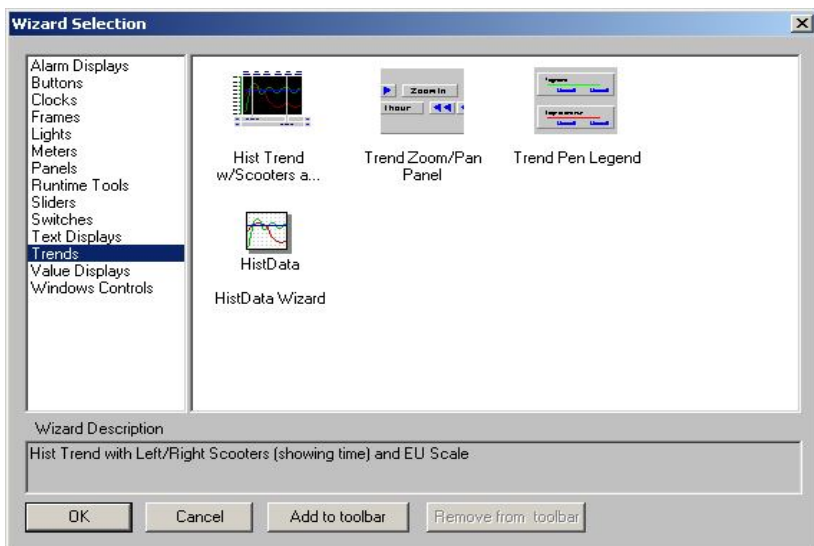


Рис.73. Диалог Wizard Selection (выбор мастер-средств).

Далее из предложенного набора мастер-средств необходимо выбрать *Hist Trend w/Scooters and Scale* (архивный тренд с бегунками) и установить в поле проекта (рис.74). Объекты этого типа ведут себя аналогично любым другим объектам, то есть их можно перемещать, масштабировать и т. д.

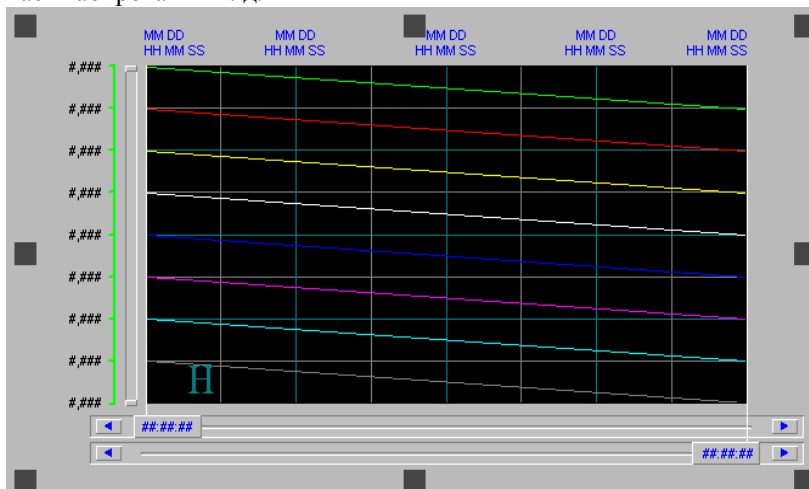


Рис.74. Объект "архивный тренд".

Двойной щелчок на объекте приводит к появлению на экране диалога конфигурирования архивного тренда (Historical Trend Char Wizard).

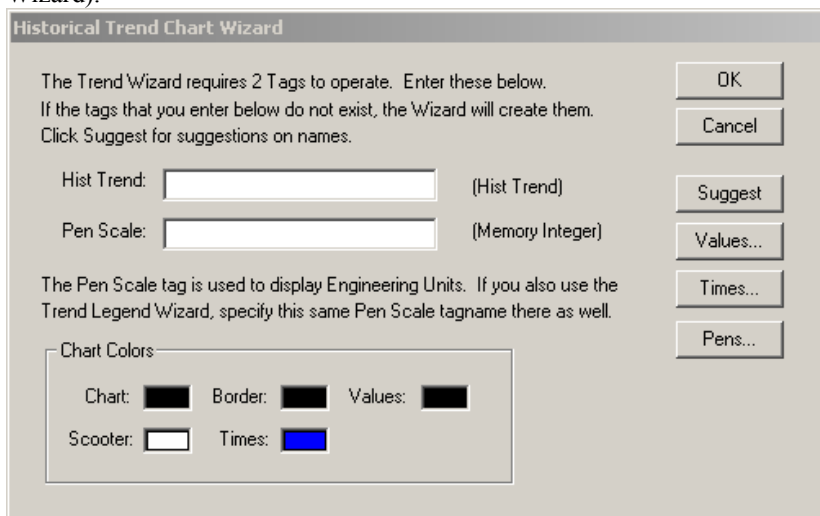


Рис.75. Диалог конфигурирования архивного тренда.

Для конфигурирования тренда с параметрами по умолчанию следует нажать кнопку Suggest (вариант). Нажатие кнопок Times и Values выводит на экран окна конфигурирования разрешения сетки по большим и малым делением горизонтальной и вертикальной осей, цвета фона и рамки графика, временного диапазона и т. д. Кнопка Pens (перья) предназначена для настройки перьев архивного тренда.

Чтобы добавить в тренд функции масштабирования и перемещения или элементы управления перьями, следует использовать панели Zoom/Pan и Trend Pen Legend (рис.73), соответственно. Для того, чтобы эти компоненты работали совместно, они должны иметь одинаковые имена (Hist Trend).

Изменение параметров архивных трендов в режиме исполнения

При управлении в режиме реального времени оператор анализирует архивную информацию. Объем информации, ее временные диапазоны, объем статистических данных, необходимые для принятия решения по управлению технологическим процессом, заранее не известны. Поэтому оператор должен иметь возможность менять на-

стройки архивных трендов, не выходя из режима Runtime. В InTouch такая возможность существует.

Для этого следует включить опцию Allow runtime changes (разрешить изменения во время исполнения) в диалоге конфигурирования архивного тренда, для стандартной компоненты Historical Trend из панели инструментов Drawing.

Теперь в режиме WindowViewer щелчок на архивном тренде будет вызывать на экран диалог изменения параметров архивного тренда (Historical Trend Setup, рис.76). В этом диалоге можно определить дату и время начала архивного тренда (поле Chart Start), его временной диапазон (Chart Length), присвоить перьям цвет и имена переменных, выбирая их из словаря.

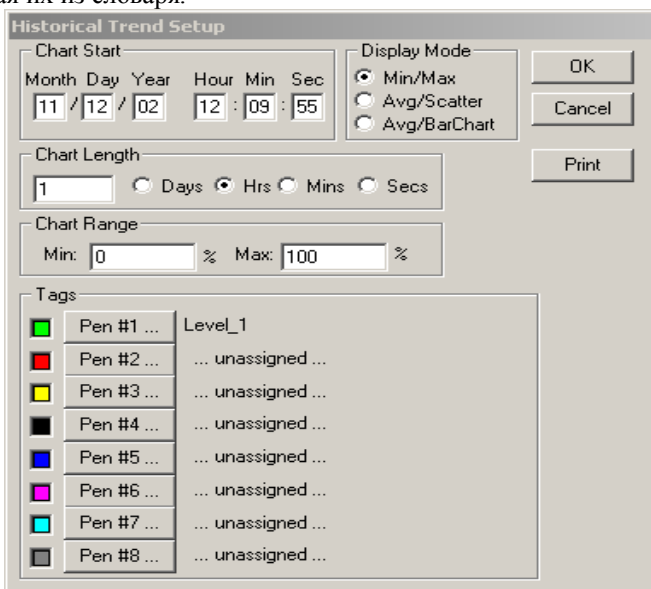


Рис.76. Диалог изменения параметров архивного тренда.

Архивный тренд может выводиться в одном из трех возможных режимах:

- *Min/Max* - график изменения значений переменной в виде вертикальных линий в процентах от всего диапазона, позволяющий оценить скорость изменения переменной;
- *Average/Scatter* - график среднего значения переменной;
- *Average/Bar Chart* - график среднего значения переменной в виде гистограммы.

Выбор режима производится в поле Display Mode.

Содержание работы

1. Обеспечить архивирование значений тэгов уровня емкостей 1 и 2. Проверить наличие архивных файлов.
2. С помощью мастер-объектов Hist Trend w/Scooter and Scale и Trend Pen Legend организовать вывод графической и числовой информации из архивных файлов. Ввести кнопку обновления исторического тренда (рис.77).

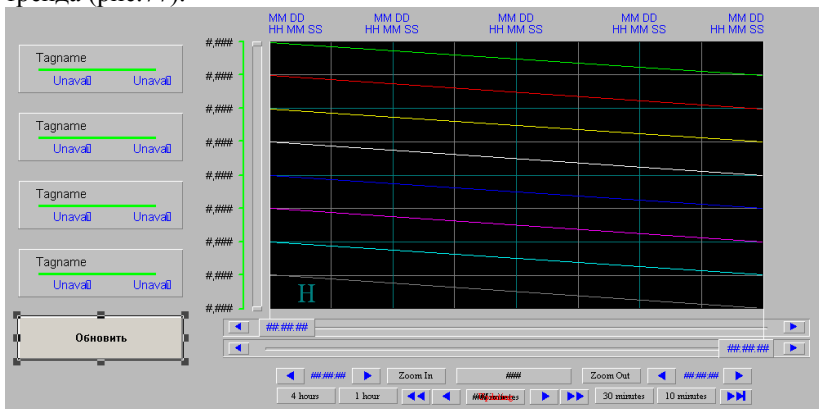


Рис.77. Окно вывода на экран исторических трендов.

3. Сконфигурировать тренд реального времени для рассматриваемых тэгов; используя масштабирование, сделать его наглядным.

Методические указания по работе

1. В *Tagname Dictionary* найдите требуемый тэг и выберите для него параметр *Log Data*. Чтобы обеспечить запись значений заданных тэгов в архивные файлы на диске, воспользуйтесь меню *Special* или в менеджере приложения нажмите *Configure – Historical Logging*. Далее в появившемся окне *Historical Logging Properties* отметьте *Enable Historical Logging* и укажите путь к специальной папке для записи архивов (данная папка должна быть предварительно создана средствами MS Windows). Запустите *WindowViewer* и спустя некоторое время убедитесь, что запись в архивные файлы производится.

2. Необходимо создать окно «trend_hist» с аналогичными параметрами окна «Main».

Необходимо задать анимационные связи для кнопки «Тренды истор.». Для кнопки «Тренды истор.» зададим свойство открытия окна

«trend_hist», при помощи анимационной связи *Show Window*.

Для вывода графической информации из архивных файлов рекомендуется использовать мастер *Hist Trend w/Scooter and Scale* (кнопка *Wizards*, раздел *Trends*). Вставьте выбранный объект в рабочее окно. Далее нажмите дважды на мастере тренда, чтобы открыть диалоговое окно настройки *Historical Trend Chart Wizard*. С помощью кнопки *Suggest* задайте имена тэгов тренда архива и шкалы пера, а затем посредством кнопки *Pens* привяжите тэги уровня наполнения к любым перьям (привязать тэги к перьям можно только после отключения *WindowViewer*).


Создайте кнопку с анимационной связью *Action*, в которой с помощью функции *HTUpdateToCurrentTime* реализуйте обновление рассматриваемого объекта с учетом последних записанных на диск данных. Пример анимационной связи *Action*:

Анимационная связь Action
<code>HTUpdateToCurrentTime("HistTrend");</code>

Для вывода числовой информации из архивных файлов используйте мастер *Trend Pen Legend* (кнопка *Wizards*, раздел *Trends*). Проследите, чтобы цветовая индикация для тэгов в данных объектах соответствовала цветам соответствующих графиков в мастере тренда.

3. Необходимо создать окно «trend_real» с аналогичными параметрами окна «Main».

Необходимо задать анимационные связи для кнопки «Тренды реал.». Для кнопки «Тренды реал.» зададим свойство открытия окна «trend_real», при помощи анимационной связи *Show Window*.

Для создания графика реального времени необходимо выбрать инструмент тренд реального времени  (кнопка *Real-time Trend* панели *Drawing*).

Для дискретных тэгов увеличьте разброс значений и разнесите соответствующие графики по вертикальной оси с помощью привязки к перьям тренда арифметических выражений следующего вида: «*ДискретныйТэг * a + b*», где *a*, *b* - числовые коэффициенты, рассчитанные на основе выбранного масштаба вывода данных трендом реального времени.

Лабораторная работа № 7. Обмен данными с MS Excel.

Цель работы:

Работа с системными функциями InTouch. Организация обмена данными с MS Excel: «непрерывная» связь, обмен по запросу.

Теоретические положения

Системные функции

InTouch предоставляет пользователю множество встроенных функций, которые могут привязываться к объектам или кнопкам, а также использоваться в сценариях для выполнения различных задач. Доступны следующие типы функций: строковый (*String*), математический (*Math*), системный (*System*), дополнительный (*Add-ons*), смешанный (*Misc*), пользовательский (*Quick*). Системные функции используются для выполнения таких операций над системой, как запуск или активизация другого приложения Windows, копирование, удаление или перемещение файлов, чтение и запись данных в файлы, поиск информации для текущего приложения. В ряде системных функций используется параметр *Trigger*. InTouch выполняет подобную функцию каждый раз, когда меняется значение *Trigger*. *Trigger* может быть любым аналоговым тэгом (не обязательно системным тэгом). Этот параметр используется только в выражениях связей анимации, а при использовании данной функции в сценарии любой аналоговый тэг может использоваться как заглушка, поскольку данный параметр не влияет на выполнение сценария.

Обмен данными MS Excel и InTouch

Когда другое приложение MS Windows запрашивает значение данных из InTouch, оно также должно знать три компонента адреса ввода/вывода:

а) VIEW (**имя приложения – application name**) указывает выполняемую программу InTouch, содержащую требуемый элемент данных;

б) TAGNAME (**имя темы – topic name**) используется всегда при чтении или записи тэга в базе данных InTouch;

в) ActualTagname (**имя элемента – item name**) указывает собственно тэг, определенный в словаре тэгов InTouch.

Например, чтобы обратиться к значению данных InTouch из Excel, необходимо в той ячейке таблицы, куда будет вставлено это

значение, ввести формулу удаленной ссылки DDE: «=VIEW|TAGNAME!'ИмяТэга'». Если в качестве сетевого протокола используется Wonderware NetDDE, то перед именем приложения в адресе ввода/вывода должно стоять имя удаленного узла. Например: «\\ИмяУзла\VIEW|TAGNAME!'ИмяТэга'».

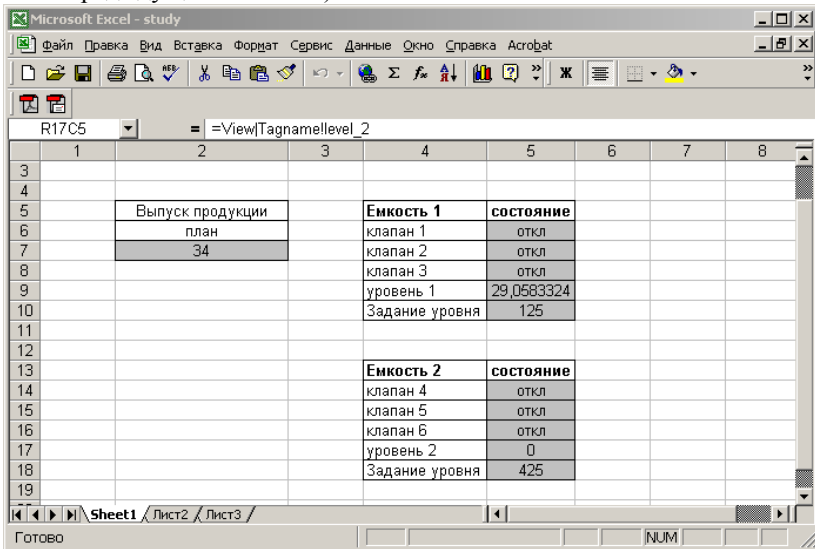
Для связи с каким-либо приложением в выполняемой программе InTouch должна быть задана следующая информация: имя приложения (Application Name), имя темы внутри приложения (Topic Name); конкретный элемент внутри темы (Item), где хранится нужное значение данных. Например, в случае обращения к MS Excel, именем приложения будет «Excel», имя темы будет скомпоновано из имен книги и листа «[ИмяКниги.xls]ИмяЛиста», а именем элемента будет указатель табличной ячейки, в которой будет происходить чтение или запись требуемых данных (например, R5C3).

WVExecute (передает команду в заданное приложение и тему), *WV Poke* (передает значение в заданное приложение, тему и элемент), *WV Request* (делает однократный запрос значения из заданного приложения, темы и элемента) используют такие же функции MS Windows, что и MS Visual Basic (DDEML). Одна функция обычно выполняет несколько задач. Например, *WV Poke* выполняет сразу три операции: DDE Initiate, DDE Poke, DDE Terminate. Благодаря этому, эти функции более защищены от ошибок и, вместе с тем, они менее эффективны при обработке множества сообщений DDE. Следует отметить, что когда команда DDE выполняет длительную процедуру в другом приложении, то это может привести к замедлению связи, но потери данных не произойдет.

Содержание работы

1. Создать кнопку вызова заданной книги MS Excel. При этом, если приложение MS Excel уже находится в оперативной памяти, то не загружать его вторично, а просто активизировать. С помощью данной кнопки обеспечить также запись в специальный файл служебной информации о рабочей книге MS Excel: дату и время последнего обновления, размер файла.

2. В книге MS Excel создать таблицу, в которую из InTouch поступают текущие значения тэгов работы клапанов, уровня наполнения, задания для уровней (рис.78). По окончании каждого технологического цикла записывать в таблицу MS Excel дату и время окончания цикла, а также объем полученного продукта (запись проводить с сохранением предыдущих значений).



The screenshot shows the Microsoft Excel interface with a table containing data for two tanks. The table has 8 columns and 19 rows. The data is organized into two sections: 'Емкость 1' (Tank 1) and 'Емкость 2' (Tank 2). Each section includes a 'Выпуск продукции' (Product Output) row, a 'Задание уровня' (Level Setpoint) row, and a 'состояние' (Status) row for each of the six valves.

	1	2	3	4	5	6	7	8
3								
4								
5		Выпуск продукции		Емкость 1	состояние			
6		план		клапан 1	откл			
7		34		клапан 2	откл			
8				клапан 3	откл			
9				уровень 1	29,0583324			
10				Задание уровня	125			
11								
12								
13				Емкость 2	состояние			
14				клапан 4	откл			
15				клапан 5	откл			
16				клапан 6	откл			
17				уровень 2	0			
18				Задание уровня	425			
19								

Рис.78. Окно MS Excel, для организации обмена данных с InTouch.

3. Из MS Excel передавать в InTouch плановый объем порции выпускаемого продукта следующим способом:

- создать аналоговый тэг типа I/O для чтения из некоторой ячейки таблицы MS Excel;

Методические указания по работе

1. Создайте кнопку с анимационной связью *Action*. В *Action* для запуска (активизации) заданной книги MS Excel используйте функции: *InfoAppActive*, *ActivateApp* (эти функции можно использовать в сочетании с функцией *InfoAppTitle*), *StartApp*. Для *StartApp* следует указать не только путь к файлу EXCEL.EXE, но также и путь к рабочей книге. Для организации записи служебной информации в специальный файл используйте функции: *InfoFile*, *StringFromTime*, *FileWriteMessage*, *Text* (или *StringFromIntg*).

Анимационная связь Action для кнопки MS Excel

```
{активизация (запуск) рабочей книги - Study.xls}
IF InfoAppActive( InfoAppTitle( "Excel" )) = 1 THEN
  ActivateApp InfoAppTitle( "Excel");
ELSE
  StartApp "c:\progra~1\microso~2\office\excel d:\user\intouch\study.xls";
ENDIF;
{запись служебной информации а infostudy.txt}
DIM Str AS MESSAGE;
Str = StringFromTime(InfoFile("d:\user\intouch \study.xls", 3, $Year ), 3);
FileWriteMessage("d:\user\intouch\infostudy.txt ", -1, Str + ", ", 0 );
Str = Text(InfoFile("d:\user\intouch\study.xls ", 2, $Year), "#" );
FileWriteMessage("d:\user\intouch\infostudy.txt ", -1, Str, 1 );
```

2. Откройте рабочую книгу MS Excel. В любых выбранных вами ячейках введите формулы для связи с заданными тэгами учебного приложения InTouch. Например: «=View|Tagname!"Level_1"». Проверьте соответствие полученных результатов значениям тэгов в InTouch.

Для организации записи данных в таблицу MS Excel по запросу используйте функцию *WWPoke*. При этом, после ввода строки адрес ввода следующей строки должен увеличиваться на единицу.

3. Отведите в рабочей книге MS Excel ячейку под плановый объем выпускаемой продукции (например, R7C2) и запишите в нее какое-либо число. В учебном приложении InTouch для указанного показателя введите аналоговый тэг типа I/O. Адреса ввода для данного тэга в InTouch: *Application Name* - Excel, *Topic Name* - [Study.xls]Sheet1, *Item* - R7C2 (здесь предполагается, что данные находятся на листе Sheet1 в книге Study).

Лабораторная работа № 8 . Мастера группы Windows Controls.

Цель работы:

Изучение объектов типа *RadioButtonGroup*, *ListBox*, *ComboBox* группы *Windows Controls*. Работа с библиотекой функций *InTouch*. Использование циклов FOR-NEXT в сценариях *InTouch*.

Теоретические положения

Сложные мастера группы Windows Controls

Управляющие элементы группы *Windows Controls* применяются в приложениях *InTouch* для отображения текста, сбора вводимой пользователем информации или вывода вариантов выбора для оператора при выполнении приложения. Представленные в *InTouch* объекты типа *ListBox*, *ComboBox* являются сложными мастерами. В отличие от обычных мастеров, для указанных объектов разработаны специальные системные функции, которые могут использоваться в *Quick*-сценариях *InTouch*. Мастер управляющего элемента окна должен иметь имя (можно использовать имя, задаваемое по умолчанию) для идентификации данного объекта *Windows Controls*. *InTouch* использует имя элемента для выполнения какого-либо действия с данным объектом в соответствующих функциях сценария. Имя элемента не увеличивает число используемых приложением тэгов и должно быть уникальным для каждого управляющего элемента. Использование тэгов, хотя и не обязательно, является ключевым для эффективной работы управляющего элемента. Например, выбор пункта из списка не имеет никакого смысла, если этот пункт не привязан к определенному тэгу и поэтому не передается автоматически в *InTouch*.

Организация циклов в Quick-сценариях InTouch

Цикл FOR-NEXT используется для выполнения определенных операторов сценария многократно за один прогон сценария. Формат цикла FOR-NEXT следующий:

FOR АналоговыйТэг = НачальноеВыражение TO КонечноеВыражение

[STEP ВыражениеИзменения]

... операторы ...

[IF (ЛогическоеВыражение) THEN EXIT FOR; ENDIF;]

... операторы ...

NEXT;

Следует обратить внимание на оператор EXIT FOR. Этот оператор используется для принудительного выхода из цикла. Применять оператор EXIT FOR имеет смысл только в логической конструкции типа:

IF (ЛогическоеВыражение) THEN (Операторы) ENDIF;».

При этом «ЛогическоеВыражение» принимает значение «истина» в том случае, когда выполнение операторов внутри цикла больше не требуется. Циклы FOR-NEXT прерывают все другие операции InTouch. При выполнении приложения не происходит никакого перемещения данных в программу WindowViewer или из неё, не обновляются никакие связи анимации и не выполняются никакие другие сценарии, включая асинхронные Quick-функции. Однако, если циклы FOR-NEXT используются внутри асинхронных Quick-функций, они не останавливают выполнение других операций. По умолчанию циклы FOR-NEXT должны завершить работу в течение 5 секунд. Это ограничение встроено в подсистему циклов FOR-NEXT ради безопасности. Однако данное ограничение можно изменить, добавив, например, следующий ключ в файл INTOUCH.INI, находящийся в каталоге приложения: LoopTimeout = 20, где: 20 - число секунд до преждевременного завершения цикла.

Содержание работы

1. Для архивных алармов организуем сортировку по группам и приоритетам, для этого необходимо воспользоваться объектом типа *RadioButtonGroup*, объекты расположить, так как показано на рис. 79.

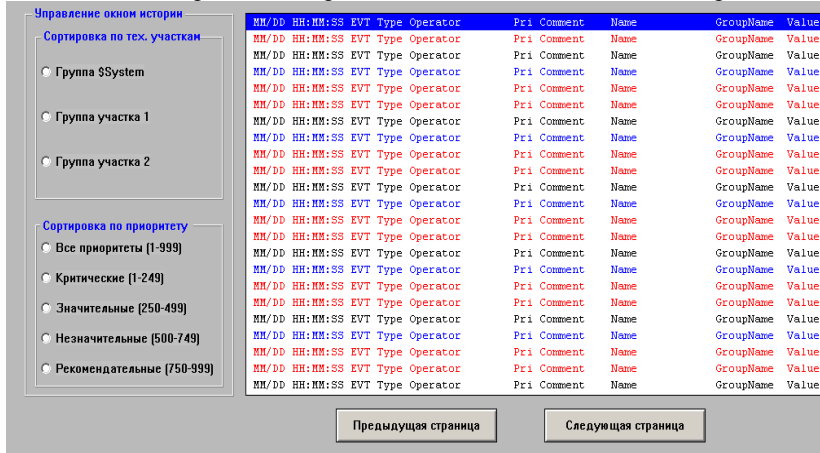


Рис.79. Расположение объектов в окне исторических алармов.

2. Создать объект типа *ListBox* или *ComboBox*. Используя любой стандартный текстовый редактор, создать файл SPS1.TXT и занести в него несколько каких-либо фамилий.

С помощью кнопочного или оконного (*On Show*) сценария загрузить данный список в созданный объект (рис.80). Обеспечить вывод информации из списка в символьный тэг.

The screenshot shows a user registration interface. On the left, there are input fields and buttons for operator name, password, and user configuration. On the right, there is a list box containing several names, with 'Комаров' selected. Below the list box are buttons for removing an item and adding a new name. The bottom of the window features 'Закреть' and 'Выход из InTouch' buttons.

Рис.80. Окно регистрации пользователя

3. Создать объект типа *ListBox* или *ComboBox*. Используя любой стандартный текстовый редактор, создать файл SPS1.TXT и занести в него несколько каких-либо фамилий.

С помощью кнопочного или оконного (*On Show*) сценария загрузить данный список в созданный объект (рис.80). Обеспечить вывод информации из списка в символьный тэг.

4. С помощью кнопки «Добавить новое имя» и поля ввода символьной информации дать возможность пользователю заносить в список новые имена (после каждого добавления список должен сохраняться в файле на диске).

5. Кнопка «Удалить из списка» должна обеспечить удаление выделенного имени из списка, а также из соответствующего файла.

Методические указания по работе

1. Создайте в окне «alarm» объект типа *RadioButtonGroup* (кнопка *Wizards*, раздел *Windows Controls*) с помощью соответствующего мастера. Нажмите дважды на данный объект, появится окно *RadioBut-*

ton Group Control. При этом InTouch сам присвоит этому объекту имя (поле *Control Name*). Необходимо ввести в поле *Tagname* имя какого-либо тэга типа *Memory Integer*, в который будет выводиться номер соответствующего переключателя из списка (если такой тэг не существует, то после нажатия ОК последует запрос на его ввод), и сконфигурировать объект, как показано на рис.81.

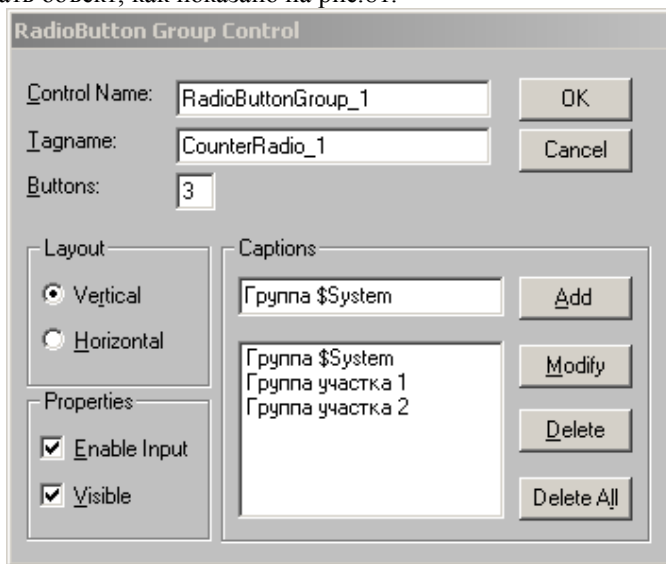


Рис.81. Конфигурация переключателя *RadioButtonGroup*.

Затем сконфигурируем объект исторических алармов, в поле *Alarm Group* вместо значения *\$System* укажем свою переменную «ChangeGroup», типа *Group Var*, которой будем присваивать названия групп «*\$System*», «*Group_1*», «*Group_2*», в зависимости от значения переключателя *RadioButton*.

Для присваивания названий групп в зависимости от значения переключателей *RadioButton*, надо написать скрипт на изменение переменной *CounterRadio_1*.

<p>Сценарий изменения данных (Data Change). Tagname – CounterRadio_1</p>
<pre>IF CounterRadio_1 == 1 THEN ChangeGroup = \$System; ENDIF; IF CounterRadio_1 == 2 THEN ChangeGroup = Group_1; ENDIF; IF CounterRadio_1 == 3 THEN ChangeGroup = Group_2; ENDIF;</pre>
<p>Сценарий уровня приложения (Application). Тип – On Startup</p>
<pre>{При старте присваиваем вывода алармов группы \$System} ChangeGroup = \$System;</pre>

Для сортировки алармов по диапазону приоритетов, создайте аналогичным способом, используя объект типа *RadioButtonGroup*

2. Создайте в рабочем окне объект типа *ListBox* (здесь и далее в учебном приложении предполагается использование только данного типа элементов *Windows Controls*, так как с элементами типа *ComboBox* можно работать по аналогичной схеме) с помощью соответствующего мастера списка (кнопка *Wizards*, раздел *Windows Controls*). Нажмите дважды на данный объект. Появится окно *ListBox Control*. При этом *InTouch* сам присвоит этому объекту имя (поле *Control Name*). Вам следует ввести в поле *Tagname* имя какого-либо тэга типа *Memory Message*, в который будет выводиться информация из списка (если такой тэг не существует, то после нажатия ОК последует запрос на его ввод). Далее в группе свойств отметьте *Sort Items* для автоматической сортировки по алфавиту выводимой информации. В заданной папке создайте файл *SPS1.TXT* следующего вида:

```
LISTBOX,4
Петров,0
Иванов,0
Сидоров,0
Комаров,0
```

Первая строка - служебная. В ней указывается тип объекта, в который может загружаться данный список (не следует путать тип с именем), далее через запятую указывается число строк в списке. Остальные строки должны заканчиваться комбинацией служебных символов «,0». Для загрузки списка в созданный объект используйте

функцию *wcLoadList*. В качестве аргументов этой функции используйте имя созданного объекта (то, что записано в поле *Control Name*) и полный путь к файлу SPS.TXT.

Сценарий окна Registration.	Тип – On Show
wcLoadList ("ListBox_1", "sps.txt");	

3. Для автоматического занесения новых имен в список воспользуйтесь функцией *wcAddItem*, далее используйте функцию *wcSaveList* для обновления файла SPS.TXT.

Кнопочные сценарии.	Тип – On Key Down
Кнопка «Удалить из списка»	
wcDeleteSelection ("ListBox_1"); wcSaveList ("ListBox_1", "sps.txt");	

4. Для выполнения данного пункта задания используйте функции *wcDeleteSelection* и *wcSaveList*.

Кнопочные сценарии.	Тип – On Key Down
Кнопка «Добавить новое имя»	
<pre>IF NewName<>"" THEN wcAddItem ("ListBox_1", NewName); wcSaveList ("ListBox_1", "sps.txt"); NewName=""; ENDIF;</pre>	

Лабораторная работа № 9. Средства безопасности.

Цель работы:

Использование средств безопасности InTouch (внутренние тэги безопасности; диалоговые окна: «вход в систему», «настройка пользователей», «смена паролей»). Организация парольной защиты объектов InTouch.

Теоретические положения

Концепция безопасности в InTouch

Применение функций безопасности к приложению не является обязательным. Однако включение средств обеспечения безопасности приложения позволяет контролировать выполнение оператором определенных функций путем привязки к этим функциям внутренних тэгов. Кроме того, если приложение снабжено функциями безопасности, то можно привязать имя оператора ко всем алармам, происходящим во время работы оператора с системой. Действие функций безопасности основано на концепции входа оператора в приложение путем ввода своего имени и пароля. Администратор определяет имя пользователя, пароль и уровень доступа для каждого оператора (между средствами безопасности ОС Microsoft и InTouch нет никакой связи). Оператору может быть дано право самостоятельно изменять свой пароль, но, естественно, не свой уровень доступа. При создании нового приложения по умолчанию создается имя пользователя *Administrator* с паролем *Wonderware* и уровнем доступа 9999 (доступ ко всем командам безопасности). При добавлении нового имени пользователя в список безопасности и после перезапуска WindowViewer заданное по умолчанию имя пользователя автоматически меняется на *None* с уровнем доступа 0 (запрет доступа к командам безопасности). Поэтому необходимо настроить имя пользователя для администратора системы с уровнем доступа равным или больше 9000, чтобы затем иметь доступ к списку пользователей. Как только вы настроили имена пользователей для приложения, необходимо изменить стандартный пароль пользователя *Administrator*, так как обычно он становится известным большинству пользователей системы (имена *Administrator*, *None* зарезервированы в системе, поэтому для них можно менять только пароли). После входа оператора в приложение доступ к любой защищенной функции будет ему предоставляться путем сравнения уровня доступа оператора со значением, определенным для внутреннего тэга безопасности, связанного с данной функцией.

Организация доступа к средствам обеспечения безопасности

Доступ к средствам обеспечения безопасности можно получить, используя стандартные окна WV с помощью меню *Special - Security*. Однако, на практике, в среде исполнения оператору часто нет доступа к меню *Special*. Поэтому, как правило, разработчиком приложения создается специальное окно идентификации пользователя, в котором оператор задает значения системных тэгов *\$OperatorEntered* и *\$PasswordEntered*. Кроме того, в этом окне можно ввести кнопки смены пароля и настройки пользователей. При нажатии этих кнопок устанавливаются в единицы значения тэгов *\$ChangePassword* и *\$ConfigureUsers* (значение данного тэга менять можно только уполномоченному пользователю). В целях предотвращения несанкционированного доступа к системе во время отсутствия оператора можно использовать системный тэг *\$InactivityTimeout*. Этот тэг принимает значение «истина» по истечении срока, заданного в конфигурации WV. Когда тэг *\$InactivityTimeout* принимает значение «истина», система приравнивает имя работающего пользователя к резервному имени «None» и выставляет тэг безопасности *\$AccessLevel* на 0. Если оператор нажмет кнопку мыши, клавишу клавиатуры или коснется сенсорного экрана до истечения заданного времени ожидания, выход из системы не произойдет.

Содержание работы

1. Создать окно, в котором оператор должен ввести имя и пароль. Ввод имени оператора осуществить с помощью активации имени в списке (рис.80) или непосредственно через поле ввода. После скрытого ввода личного пароля на экране должно появиться соответствующее значение уровня доступа для данного пользователя (рис.80).

2. Написать сценарий, по которому при неверном задании пароля автоматически устанавливается имя пользователя None с уровнем доступа 0.

3. Организовать ввод имен операторов из имеющегося списка.

4. Ввести кнопку «Изменить личный пароль» (рис. 82), с помощью которой оператор может заменить свой пароль (без изменения имени пользователя и уровня доступа), а также кнопку «Конфигурирование пользователей» (рис. 83), которая позволяет пользователям с высоким уровнем доступа задавать имена, пароли и уровни доступа в системе.



Рис.82. Окно изменения личного пароля



Рис.83. Окно конфигурирования пользователей

Методические указания по работе

1. Для ввода имени пользователя будем использовать тэг *operator* (Memory Message), а для ввода пароля – тэг *password* (Memory Message), которые в дальнейшем необходимо присвоит тэгам *\$OperatorEntered* и *\$PasswordEntered*. Для ввода будем использовать анимационную связь *User Inputs – String*, связанную с прямоугольным графическим объектом. При задании анимационной связи для пароля в группе *Echo Characters?* необходимо указать *No*.

Для вывода имени текущего оператора используйте тэг *\$Operator* (*Value Display – String*), а для показания его уровня доступа тэг *\$AccessLevel* (*Value Display – Analog*).

2. Создайте сценарий на изменение данных (*Data Change*) для тэга пароля *password*, при этом в сценарии нужно присвоить тэгам *\$OperatorEntered* и *\$PasswordEntered* значения соответствующих тэгов имен пользователя и пароля. В этом сценарии необходимо проверить

на равенство значения тэгов *\$OperatorEntered* и *\$Operator*. Если эти значения различаются между собой, то, следовательно, пароль был задан неверно. В данном случае тэгу *\$OperatorEntered* нужно принудительно присвоить значение «None» (тэг *\$PasswordEntered* в этой ситуации не используется), а оператору выдать сообщение об ошибке.

Пример скрипта.

Сценарий изменения данных (Data Change). Tagname – password
<pre>\$OperatorEntered = operator; \$PasswordEntered = password; IF \$OperatorEntered <> \$Operator THEN \$OperatorEntered = "none"; ENDIF;</pre>

3. Для ввода имен оператора из списка необходимо использовать тэг *message_listbox* элемента *ListBox* в скрипте на изменения данных и присвоить его тэгу ввода имени оператора.

Сценарий изменения данных (Data Change). Tagname – message_listbox
<pre>operator = message_listbox;</pre>

4. Для кнопок «Изменить личный пароль» и «Конфигурирование пользователей» задайте анимационные связи *Touch Pushbuttons - Discrete Value - Set* для тэгов *\$ChangePassword* и *\$ConfigureUsers*, либо используйте связи *Action*, в которых введите операторы «*\$ChangePassword = 1;*» и «*\$ConfigureUsers = 1;*».

Лабораторная работа № 10. Рецепты и SQL-запросы.

Цель работы:

Знакомство с подсистемой управления рецептами InTouch. Организация доступа к программам управления базами данных с помощью SQL-запросов.

Теоретические положения

Работа с подсистемой управления рецептами

В производственной сфере понятие рецепта используют для описания различных вещей. Например, в хлебобулочном производстве это может быть основной рецепт теста для изготовления печенья, где перечислены все ингредиенты, необходимые для выпечки обычного печенья, а также все дополнительные ингредиенты, например, орехи, фрукты, кусочки шоколада и т. п., которые можно добавлять к основному рецепту для выпечки различных сортов печенья. На металлообрабатывающем предприятии рецептом может быть ряд установочных параметров оборудования. При пакетной обработке данных рецепт может использоваться для описания различных стадий процесса обработки. Recipe Manager системы InTouch в сочетании с базовыми функциями Quick-сценария InTouch может работать в любой из ситуаций, подобной приведенным выше. Подсистема управления рецептами состоит из двух компонентов: Recipe Manager и набор функций для рецептов. Recipe Manager - автономная программа, которая используется для создания шаблонов рецептов. Шаблон рецепта состоит из трех таблиц: *Template Definition* — для определения ингредиентов рецепта (произвольные имена) и соответствующих типов данных (допустимые типы - *Analog, Discrete, Message*); *Unit Definition* - для привязки тэгов InTouch к ингредиентам рецепта (тэги могут быть типа *Memory*, либо типа *I/O* для загрузки рецепта непосредственно в PLC); *Recipe Definition* - для задания имени рецепта и количества каждого ингредиента. Шаблон каждого рецепта сохраняется в формате .CSV (Comma Separated Variable). Экземпляры рецепта можно изменять, создавать или удалять в процессе работы в среде исполнения InTouch с помощью функций для рецептов. Функции для рецептов используются в Quick-сценариях системы InTouch для доступа к шаблонам. С помощью этих функций в InTouch можно выбирать, загружать, изменять, создавать и удалять рецепты, содержащиеся в шаблонах.

Работа с InTouch SQL Access

Продукт InTouch SQL Access состоит из утилиты SQL Access Manager и SQL функций. Данный продукт использует библиотеку Intersolv/Q+E Software Database Library QELIB, которая работает с множеством различных СУБД и включает в себя все коды, необходимые для взаимодействия с драйверами ODBC. Драйверы ODBC необходимы для непосредственного «соединения» с базой данных. Утилита SQL Access Manager создает списки привязок (*Bind Lists*) и шаблоны таблиц (*Table Templates*). Список привязок ассоциирует колонки базы данных с тэгами из словаря данных InTouch. Шаблон таблиц определяет структуру и формат новой таблицы в базе данных. Процесс привязки колонок базы данных к тэгам называется «связыванием». «Связывание» позволяет программе SQL Access Manager манипулировать данными непосредственно в базе данных. Программа SQL Access Manager сохраняет информацию для списков привязки шаблонов таблиц в файле SQL.DEF в формате разделенных запятыми переменных (.CSV). Этот файл можно просмотреть или изменить с помощью SQL Access Manager или любого текстового редактора, например, Notepad.

Содержание работы

1. С помощью программы Recipe Manager ввести данные в таблицы *Template Definition*, *Unit Definition*, *Recipe Definition* (рис.84).

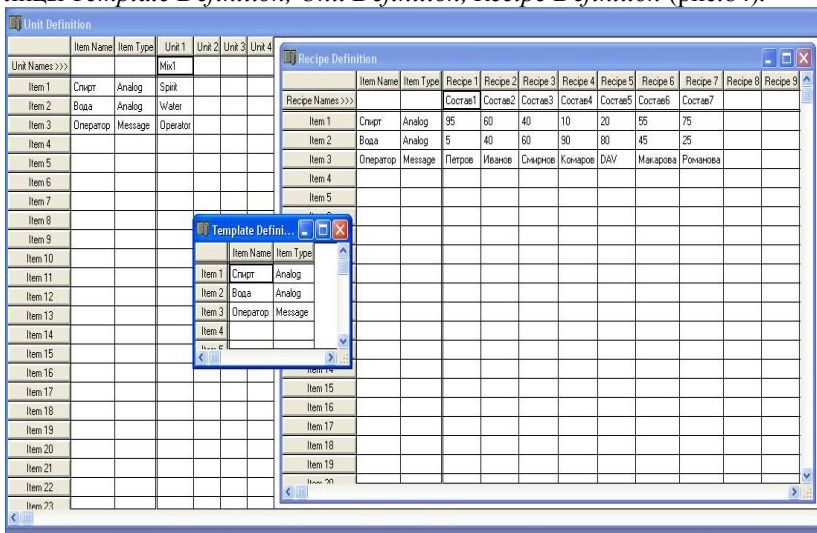


Рис.84. Таблицы программы Recipe Manager.

2. Создать окно в InTouch для загрузки имеющихся рецептов, их корректировки и ввода новых рецептов (рис.85).

№ смеси	9
Спирт, %	40
Вода, %	60
Исполнитель	Иванов

Выбор рецепта

Наименование рецепта

9

Запись рецепта

Рис.85. Панель оператора для взаимодействия с подсистемой управления рецептами.

3. Используя системный источник данных MS Windows ODBC задать системный DSN для базы данных TEST.MDB.

4. Заполнить Bind List в SQL Access Manager (рис.86).

Bind List Configuration

Add Item Delete Item Modify Item Cancel OK

Bind List Name: BindList_my Save

Tagname.FieldName Column Name

Number_my Номер

Tagname FieldName Move Up Move Down

Tagname.FieldName	Column Name
Number_my	Номер
Name_my	Название
Cost_my	Цена

Рис.86. Окно конфигурации Bind List в SQL Access Manager.

5. С помощью кнопки «Установить связь» подключиться к таблице «Таблица1» базы данных TEST.MDB (рис.87, рис.88). Создать кнопки для просмотра записей в указанной таблице и их корректировки (рис.87).

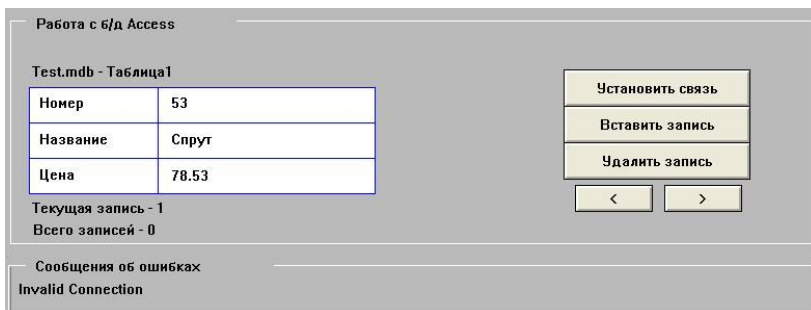


Рис.87. Панель оператора для взаимодействия с таблицей базы данных MS Access.

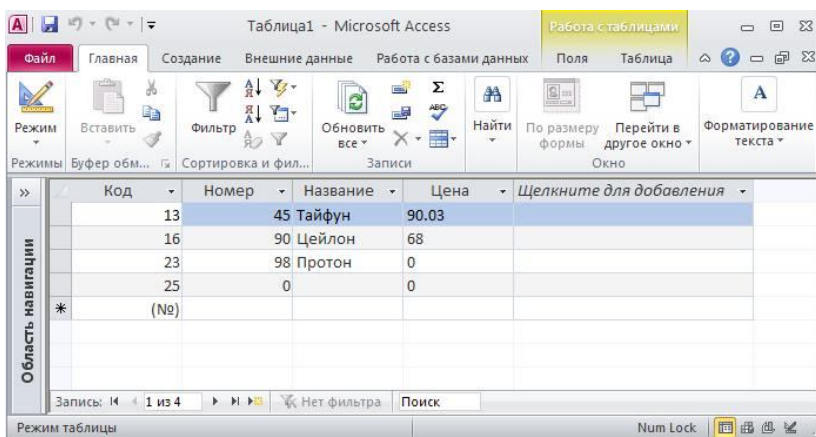


Рис.88. Учебная панель базы данных MS Access.

Методические указания по работе

1. В менеджере приложения WindowMaker нажмите *Applications*, затем дважды нажмите *Recipe*. Далее в окне *Recipe Manager* в меню *File* нажмите *New*. После этого в данном окне появятся три окна с таблицами: *Template Definition*, *Unit Definition*, *Recipe Definition*. В *Template Definition* в поле *Item Type* введите имена двух ингредиентов рецепта (после каждого ввода не забывайте нажимать *Enter*), а в поле *Item Type* соответственно их типы. Например, «% влажности» типа *Analog* (вместо наименования типа достаточно ввести только его первую букву, т.е. в данном случае «A», затем *Enter*), «Анализ выполнил» типа *Message*. При этом указанные наименования автоматически появятся в левой части таблиц *Recipe Definition* и *Unit Definition*. В табли-

це *Recipe Definition* в колонках *Recipe 1*, *Recipe 2* введите заголовки наименования рецептов (*Recipe Names*), затем введите какие-нибудь конкретные числовые и текстовые (если присутствует *Item* типа *Message*) значения. В таблице *Unit Definition* в колонке *Unit 1* задайте какое-нибудь имя для набора тэгов (*Unit Name*), например, *Mix1*. Далее каждый ингредиент свяжите с тэгом соответствующего типа (имя тэга можно получить из списка тэгов с помощью двойного нажатия на пустом поле). Наконец, сохраните введенные данные в .CSV файле (*File - Save*).

2. В кнопочном сценарии выбора рецепта используйте функции *RecipeSelectRecipe* и *RecipeLoad*. В параметрах *RecipeSelectRecipe* следует указать путь к файлу с рецептами и имя тэга типа *Memory Message*, в который запишется имя выбранного рецепта. Для *RecipeLoad* нужен еще один параметр - имя набора тэгов из *Unit Definition* (для приведенного выше примера – *Mix1*). В кнопочном сценарии записи рецепта используйте функцию *RecipeSave*. До запуска указанного сценария в среде исполнения InTouch введите новый рецепт в соответствующие тэги. Затем в режиме записи нового (коррекции уже существующего) рецепта для функции *RecipeSave* задайте новое (существующее) значение тэга имени рецепта.

3. В *Control Panel* MS Windows NT откройте *Data Sources (ODBC)* или *Administrative Tools*, затем *Data Sources (ODBC)* (при использовании MS Windows 2000). В окне *ODBC Data Source Administrator* выберите *System DSN -Add - Microsoft Access Driver (*.mdb)*. Затем задайте любое имя в поле *Data Source Name* и с помощью кнопки *Select* укажите путь к требуемой базе данных.

4. В менеджере приложения WindowMaker нажмите *SQL Access Manager*, затем дважды нажмите *Bind List* и войдите в окно *Bind List Configuration*. В поле *Bind List Name* задайте какое-нибудь имя для нового листа, потом введите названия тэгов и соответствующие названия полей в таблице базы данных (порядок следования полей в *Bind List* должен соответствовать порядку следования этих полей в таблице базы данных).

5. Для того чтобы подключиться к заданной таблице, в сценарии кнопки «Установить связь» используйте функции *SQLConnect*, *SQLSelect*. В параметрах функции *SQLConnect* следует указать ранее введенное имя DSN и имя какого-либо тэга типа *Memory Integer* для хранения номера (ID), присвоенного функцией соединения SQL для каждой связи в базе данных. Для функции *SQLSelect* требуется дополнительно задать имя используемой таблицы базы данных и ранее заданное имя листа в окне *Bind List Configuration*. Для диагностики оши-

бок полезно использовать функцию *SQLErrorMsg*, которая возвращает сообщение об ошибке, соответствующее коду, полученному в результате выполнения предыдущей функции SQL. Для получения данных из рабочей таблицы MS Access используйте следующие функции: *SQLNumRows*, *SQLFirst*, *SQLNext*, *SQLGetRecord*, *SQLPrev*. Корректировку записей в таблице осуществляйте с помощью функций *SQLInsert* и *SQLDelete*. Во избежание потери данных для функции *SQLInsert* используйте другой набор тэгов (отличный от набора тэгов, используемого для чтения данных из указанной таблицы).

Литература

1. Павлеченко А. Практикум по Wonderware InTouch. Базовый и дополнительный курсы. – М.: Научтехлитиздат, 2002. – 140 с.: ил.
2. Андреев Е.Б., Мезенцева С.А., Пелипец С.В. «Проектирование систем управления в SCADA-пакете InTouch». Компьютерный практикум. – М.: РГУ нефти и газа, 2009. – 83 с.
3. Харазов В.Г. Интегрированные системы управления технологическими процессам. – СПб.: Профессия, 2009. – 550 с.: ил.
4. Схиртладзе А.Г. Интегрированные системы проектирования и управления: учебник для студ. высш. учеб. заведений / А.Г. Схиртладзе, Т.Я. Лазарева, Ю.Ф. Мартемьянов. – М.: Академия, 2010. – 352 с.
5. Бойков В.И. Интегрированные системы проектирования и управления. Учебное пособие / В.И. Бойков, Г.И. Болтунов, О.К. Мансурова – СПб.: СПбГУ ИТМО, 2010. – 162 с.
6. Федоров Ю.Н. Основы построения АСУТП взрывоопасных производств. В двух томах. Т.2 «Проектирование». – М.: Синтег, 2006. – 632 с.: ил.
7. Нестеров А.Л. Проектирование АСУТП. Методическое пособие. Книга 1. – СПб.: Деан, 2006. – 552 с.: ил.
8. Нестеров А.Л. Проектирование АСУТП. Методическое пособие. Книга 2. – СПб.: Деан, 2009. – 944 с.: ил.
9. Управляющие вычислительные комплексы: Учеб. Пособие / Под ред. Н. Л. Прохорова. 3-е изд. перераб. и доп. – М.: Финансы и статистика, 2003. – 352с.: ил.
10. Гук М.Ю. Аппаратные средства IBM PC. Энциклопедия. 3-е изд. – СПб.: Питер, 2006. – 1072 с.: ил.
11. Общие правила взрывобезопасности для взрывопожароопасных химических, нефтехимических и нефтехимических производств. ПБ 09-540-03

Учебное пособие

Долганов Андрей Викторович
кандидат технических наук, доцент

Минигалиев Григорий Барыевич
старший преподаватель

Елизаров Виталий Викторович
доктор технических наук, доцент

ИНТЕГРИРОВАННЫЕ СИСТЕМЫ ПРОЕКТИРОВАНИЯ И УПРАВЛЕНИЯ

Корректор Белова И.М.
Худ . редактор Федорова Л.Г.

Сдано в набор 24.11.14
Подписано в печать 26.11.14
Бумага писчая. Гарнитура Таймс.
Усл.печ.л. 7,75. Тираж 50.
Заказ № 12

НХТИ (филиал) ФГБОУ ВПО «КНИТУ»,
г. Нижнекамск, 423570, ул 30 лет Победы, д. 5а.