Одиночная система: Intouch

Возможности:

1. Интерфейсы связи - Более 240 нативных драйверов и OPC
2. Открытая архитектура
3. Подключение к базам данных:  
   Подключение к любой SQL базе данных, например: SQL Server, Oracle, MySQL, Sybase, OSI PI, MS Access, Excel, и т.д., Посредством стандартных технологий: ADO.NET, ADO, OLE-DB, и ODBC.
4. Интеграция Tag-баз:
5. Поддерживаемые системы конфигурирования PLC(список расширяется):
   1. TwinCAT (Beckhoff)
   2. CodeSys (3S)
   3. RSLogix 5000 Family (Rockwell)
   4. Allen-Bradley PLC5, SLC500 (Rockwell)
   5. AutomationDirect Koyo (AD)
   6. AutomationDirect PAC 3000 (AD)
   7. Schneider Unity Modbus (Schneider-Electric)
6. Повышает эффективность на этапах разработки и обслуживания
7. Удаленное развертывание
   1. Подключение к «Рабочим станциям» посредством канала TCP/IP, например Ethernet, Wireless Ethernet (IEC 802.11), Internet, Dial-Up соединения.
   2. Загрузка проекта на Рабочую станции; Импорт проекта с Рабочей станции; Инсталляция и обновление системных файлов; Установка лицензий; Запуск/Останов работы проекта.
   3. Удаленное Конфигурирование проекта в On-line режиме.
8. Удаленная диагностика
   1. Отладка проекта в режиме удаленного подключения к Рабочей станции.
   2. Просмотр тегов, запись в теги, выполнение действий удаленно.
   3. Отслеживание событий и интерфейсов связи удаленно
9. Полноценный инструмент отладки скриптов
10. Лицензирование
    1. Среда разработки:
       1. InTouch Machine Edition Studio
       2. 1000 или 3000 тегов
       3. USB Ключ
       4. Одина клиентская лицензия на Web Thin Client, Secure Viewer Thin Client, Studio Mobile Access (SMA) Thin Client.
    2. Среда исполнения:
       1. Compact / Standard
       2. 100, 500, 1000, 2000, 3000 тегов
       3. От 1 до 5 драйверов IO
       4. Софтверная лицензия

Поддерживаемые архитектуры InTouch

Поддерживаются следующие сетевые архитектуры InTouch:

• однокомпьютерная;

• клиентская;

• серверная;

• система разработки сетевых приложений (NAD).

Однокомпьютерная архитектура

Однокомпьютерное приложение обычно включает один компьютер, не подключенный к сети и выполняющий функции первичного интерфейса оператора. Такой компьютер связывается с технологическим процессом прямым соединением, например, последовательным кабелем. В такой архитектуре разработка приложения InTouch ведется на одном компьютере. Можно скопировать приложение на другой компьютер, внести в него изменения и скопировать обратно.

Клиентская архитектура

При использовании клиентской архитектуры имеется уникальный экземпляр приложения InTouch для каждого компьютера, на котором работает Табло Окон (узел просмотра, View node), или в уникальном месте на сетевом сервере. В примере, приведенном ниже, приложение разрабатывается и тестируется на узле разработки, а затем копируется на каждый из узлов просмотра.

Данная архитектура обладает резервированием, так как каждый узел может быть автономным, и нет ограничений на количество используемых узлов просмотра.

Каждый узел просмотра должен иметь идентичную копию приложения и идентичный доступ ко всем сетевым информационным ресурсам, например, серверам ввода-вывода или IndustrialSQL Server. Однако каждый узел просмотра ведет собственный обмен данными с разделяемым сервером, что может привести к повышенной нагрузке на сеть.

Можно вносить изменения в приложение и тестировать его на узле разработки без влияния на работающие процессы. Однако затем необходимо распространить измененное приложение на узлы просмотра. Необходимо локально прекращать работу каждого узла просмотра, копировать на него новое приложение и затем запускать узел заново.

Серверная архитектура

При использовании серверной архитектуры общее приложение InTouch распространяется на несколько узлов просмотра. На приведенном ниже рисунке два узла просмотра имеют доступ к одному приложению с узла разработки.

Для каждого узла просмотра необходимо выполнить следующее:

• логический накопитель должен быть поставлен в соответствие разделяемому сетевому накопителю узла разработки;

• разделяемое приложение должно быть зарегистрировано в программе InTouch;

• компьютер должен иметь идентичный доступ ко всем источникам данных, на которые ссылается приложение.

Имеются также способы указания расположения источников данных с использованием комбинации скриптов для указания имени узла и изменения расположения каждого источника данных на основе этого имени.

При использовании такой архитектуры ведется только одно приложение. При внесении изменений в приложение и перезапуске Табло Окон узлы просмотра автоматически обновляются. Эта архитектура имеет ряд недостатков:

• ограничение возможностей разработки приложения;

• отсутствие резервирования в случае отказа узла разработки;

• все узлы должны иметь одинаковое разрешение экрана.

Архитектура разработки сетевых приложений (NAD)

При использовании этой архитектуры ведется основной экземпляр приложения на центральном сетевом пункте, который обычно представляет собой узел разработки.

Каждый узел просмотра копирует приложение в заданное пользователем место и управляет его работой.

При оповещении клиентов об изменении приложения (с помощью команды Notify Clients (Уведомить клиентов) в меню Special в Построителе Окон) в каталоге приложения устанавливается флаг, который затем считывается узлами просмотра.

Имеется возможность настраивать режим действий узлов просмотра при внесении изменений в приложение.

Возможные действия – от игнорирования изменений до автоматического прекращения работы и перезапуска узла просмотра с перезагрузкой основного приложения.

На приведенном ниже рисунке для двух узлов просмотра основное приложение зарегистрировано с узла разработки, однако фактически узлы просмотра работают с этим приложением локально на своих компьютерах.

*Примечание – Если приложение настроено на запись исторических данных в каталог приложения на узле основного приложения, то все узлы NOD будут пытаться записывать свои исторические данные в основное приложение. Чтобы избежать этого, следует установить запись исторических данных в локальный каталог, а не на узел основного приложения.*

Если выполняется распространение крупного сложного приложения на несколько узлов, то при первоначальной загрузке реакция системы может быть достаточно медленной. Однако при этом обеспечивается оптимизация обновлений. Передача приложения может составить проблему для медленных сетей или для последовательных соединений.

Кроме того, необходимо учитывать другие сетевые ограничения, например, пользователей маршрутизаторов, отфильтровывающих определенные типы сетевого трафика и адреса.

Факторы, учитываемые при планировании сетевых приложений

Независимо от архитектуры, выбранной для построения приложения InTouch, необходимо учитывать следующие факторы:

• доступ к источникам данных ввода-вывода;

• доступ к разделяемым файлам;

• место регистрации данных;

• особые требования, обусловленные сетью.

Доступ к данным ввода-вывода для сетевых приложений

Для доступа к данным ввода-вывода в реальном времени человеко-машинный интерфейс InTouch использует имена доступа (Access Names). Каждое имя доступа назначается адресу ввода-вывода, состоящему из имени узла, приложения и раздела. В распределенных приложениях ссылки ввода-вывода могут задаваться как глобальные адреса сетевого сервера ввода-вывода или как локальные адреса локального сервера ввода-вывода.

*Примечание – Система InTouchView может использовать только одно имя доступа Galaxy. Создать в InTouchView другие имена доступа невозможно. Подробнее об ограничениях, связанных с InTouchView, см. в разделе “Просмотр приложений в реальном времени”.*

Узел просмотра должен иметь такой же доступ к источникам данных, как и узел разработки.

Глобальные адреса ввода-вывода

Использование глобальных адресов данных ввода-вывода позволяет всем узлам просмотра совместно использовать общий сетевой сервер ввода-вывода. За счет этого устраняется необходимость в использовании нескольких серверов ввода-вывода, однако такая система менее устойчива к сбоям, а также обладать меньшей общей производительностью.

На приведенном ниже рисунке показаны два узла просмотра, работающие с экземпляром одного приложения. Оба узла просмотра обращаются к одному источнику данных ввода-вывода. Так как каждое приложение использует для источника данных полностью заданный адрес ввода-вывода, все ссылки относятся к одному и тому же серверу ввода-вывода.

Можно настроить приложение InTouch таким образом, чтобы оно идентифицировало элемент данных, хранящийся на другом узле, используя для этого трехкомпонентную систему адресации в имени доступа. Система адресации для имени доступа включает имя узла, имя приложения, а также имя раздела, в котором находятся удаленные данные. Приложение InTouch получает удаленные данные, используя имя доступа в сочетании с именем элемента данных. Подробнее о задании имени доступа для удаленного сервера ввода-вывода см. в документе “Человекомашинный интерфейс InTouch®. Руководство по управлению данными” (глава 5).

*Примечание – Если при создании имен доступа в Построителе Окон этими именами используется протокол SuiteLink, то программное обеспечение не позволяет именам доступа обращаться к одному и тому же узлу, приложению и разделу. Не разрешается использовать функцию IOSetAccessName() для перенаправления имен доступа с целью их дублирования во время работы: такое использование приведет к тому, что перенаправленные имена доступа не будут работать.*

Локальные адреса ввода-вывода

Локальные адреса данных ввода-вывода используются, если каждый узел просмотра имеет собственный сервер ввода-вывода. Такая архитектура обеспечивает высокую устойчивость к сбоям, так как каждый узел просмотра может продолжать работать независимо в случае отказа сети.

На рисунке, приведенном ниже, два узла просмотра работают с экземплярами одного приложения, ссылающимися на собственные источники данных ввода-вывода. Так как каждое приложение использует для источника данных локальный адрес ввода-вывода, каждая ссылка относится к локальному серверу ввода-вывода.

Использование локального сервера ввода-вывода значительно повышает нагрузку на сеть, связывающую процессы. Например, использование трех узлов утраивает трафик, создаваемый одним узлом, так как запросы каждого узла требуется обрабатывать отдельно.  
Коммуникационный протокол Wonderware SuiteLink

Коммуникационный протокол Wonderware SuiteLink построен на основе протокола TCP/IP. Этот протокол следует использовать для высокоскоростных промышленных приложений благодаря его следующим свойствам:

• система VTQ (Value Time Quality, “Значение – время - качество”), в которой со всеми значениями данных, передаваемыми VTQ-клиентам (VTQ-aware clients), связывается отметка времени и индикатор качества. Человекомашинный интерфейс InTouch является VTQ-клиентом, данные тегов которого передаются с индикатором VTQ;

• широкомасштабная диагностика потока данных, нагрузки на сервер, потребления ресурсов компьютера и сетевого транспорта доступна через монитор производительности операционной системы Microsoft Windows;

• между приложениями могут поддерживаться стабильно высокие объемы данных, независимо от того, работают ли эти приложения на одном узле или распределены по большом количеству узлов.

Протокол SuiteLink не является заменой для DDE, FastDDE или NetDDE. Каждое соединение между клиентом и сервером зависит от условий конкретной сети.

Доступ к разделяемым файлам

В распределенных приложениях ссылки на файлы могут быть настроены в следующих формах:

• глобальные адреса сетевого файл-сервера;

• локальные адреса локальных файлов.

Узел просмотра должен иметь такой же доступ к источникам данных, как и узел разработки.

Глобальные адреса файлов данных

Имеется возможность настроить глобальные адреса файлов данных таким образом, чтобы все узлы просмотра совместно использовали (разделяли) общий сетевой набор файлов.

Таким образом, обеспечивается ведение файлов с одним источником, однако такая система менее устойчива к сбоям, чем использование локальных экземпляров.

На рисунке, приведенном ниже, два узла просмотра работают с экземпляром одного приложения, однако ссылаются на один и тот же файл программы технологического процесса.

Так как каждое приложение использует буквенное обозначение накопителя, поставленное в соответствие полностью заданному сетевому пути к файлу, все ссылки указывают на один и тот же файл.

Локальные адреса файлов данных

Имеется возможность использовать локальные адреса файлов данных, когда каждый узел просмотра имеет собственный экземпляр файла. На приведенном ниже рисунке каждый из трех узлов просмотра работает с экземпляром одного и того же приложения и ссылается на локальный экземпляр файла программы технологического процесса.

Доступ к разделяемым файлам с использованием UNC-адресов

Адреса, соответствующие Универсальному соглашению об именовании (Universal Naming Convention, UNC-адреса), можно использовать во всех случаях вместо путей к файлам, например, при указании каталогов приложений, в элементах конфигурации, в распределенных алармах.

Если используются UNC-имена, то создавать накопители с установленным соответствием не требуется.

UNC-адрес имеет вид: \\Узел\Разделяемый\_ресурс\Путь, где:

• Узел – имя компьютера, на котором располагается разделяемый файловый ресурс;

• Разделяемый\_ресурс – логическое имя, назначенное разделяемой папке на вышеуказанном компьютере;

• Путь – обычный путь к файлу относительно разделяемого ресурса.

Регистрация данных в распределенной среде

Имеется возможность использовать систему распределенной истории InTouch для получения исторических данных от любого приложения InTouch в сети. Эта система предусматривает также дистанционное извлечение данных из нескольких исторических баз данных одновременно. Эти базы данных называются источниками исторических данных (history providers).

Регистрацию данных в распределенном файле истории может выполнять только один узел InTouch, однако просматривать содержимое этого файла может неограниченное количество узлов InTouch.

Удаленный узел, извлекающий данные из файла истории, может не видеть данные за последний час (по времени узла регистратора). Удаленные тренды могут просматривать только данные, записанные на диск узла, выполняющего регистрацию.

Данные для каждого тега, проверенные на “регистрацию данных” (“Log Data”), автоматически записываются на диск после того, как для данного тега собрано 22 выборочных значения. Если вызывается функция HTUpdateToCurrentTime(), то данные записываются на диск независимо от количества собранных выборочных значений. По умолчанию данные записываются на диск каждый час.

Использование Сервисов Терминалов

Сервисы Терминалов (Terminal Services) – настраиваемый сервис, входящий в состав операционных систем Microsoft Windows Server, управляющих работой Windows-приложений в централизованном режиме с сервера. В Сервисах Терминалов компьютеры-клиенты имеют доступ к узлу-серверу, где одновременно работает несколько экземпляров приложений InTouch.

Среда Сервисов Терминалов включает три основных компонента:

• Terminal Services Server (Сервер Сервисов Терминалов) – управляет вычислительными ресурсами для каждого клиентского сеанса и предоставляет пользователям-клиентам их собственную уникальную среду. Сервер принимает и обрабатывает все нажатия клавиш и операции с мышью, выполняемые на удаленном клиенте, и направляет соответствующему клиенту все данные для вывода на дисплей, относящиеся как к операционной системе, так и к прикладным программам. Вся работа Сервисов Терминалов с приложениями осуществляется на сервере;

• Remote Desktop Protocol (RDP, протокол удаленных компьютеров) – протокол, обеспечивающий связь между сервером и клиентами. Установленное клиентское приложение, работающее с протоколом RDP, передает на сервер входные данные (нажатия клавиш, движения мыши и т.д.);

• Client (клиент). Клиент Сервисов Терминалов не выполняет каких-либо операций с локальным приложением; он только отображает выходные данные приложения. Доступ к Сервисам Терминалов с клиента осуществляется с помощью команды Terminal Services Client (Клиент Сервисов Терминалов) из меню Windows Программы. При подключении к серверу терминалов среда компьютера-клиента выглядит так же, как и сервер Windows. Совершенно очевидно, что приложение не работает локально.

GE HMI SСADA iFix

1. Распределенная база данных, позволяющая с любого рабочего места оператора получить доступ к информации на любом сервере, в том числе одновременно с нескольких серверов. БД поддерживает более 30 типов первичных и вторичных тегов, среди которых: аналоговые и дискретные входы и выходы, теги вычислений, сумматоры, таймеры, непрерывные и статистические функции управления, команды SQL, теги массива и т.д.
2. База данных выполняется в своем потоке, что позволяет редактировать экраны без остановки процессов сбора и обработки данных. Редактирование отдельного тега также не останавливает обработку других тегов в базе.
3. С помощью наборов инструментов можно разработать собственные типы тегов или внешние приложения, использующие данные из базы пакета.
4. Обширный каталог высокопроизводительных прямых драйверов ввода/вывода и ОРС-серверов как для широко распространенных, так и для специализированных ПЛК.
5. Универсальный ОРС-клиент с возможностями конфигурации, мониторинга и оптимизации обмена с серверами ввода/вывода.
6. Утилита «Discover and Auto Configure», позволяющая просматривать проекты контроллеров и автоматически создавать теги в базе данных.
7. Многоуровневая система защиты с ограничением доступа отдельных пользователей или узлов GE iFIX. Возможность синхронизации с системой безопасности Windows. Возможность работы в качестве службы Windows.
8. Использование технологии электронных подписей и записей.
9. Гибкий и мощный механизм распределенной сетевой службы тревог для маршрутизации сообщений по зонам, журналы тревожных сообщений системы.
10. Горячее резервирование и автоматическое переключение серверов, обеспечивающее в случае отказа основного узла оперативное переключение на резервный сервер или резервную локальную сеть. Использование выделенного канала синхронизации серверов с протоколированием событий. Наличие функции резервного копирования и восстановления в случае критических отказов оборудования.
11. Мощные и удобные средства для быстрой разработки экранов:

– переход из режима редактирования в режим разработки проекта без компиляции и остановки управления по нажатию одной кнопки;  
– высокопроизводительная и качественная объектно-ориентированная графика, возможность использования внешних объектов ActiveX и .NET;  
– технология безопасного внедрения ActiveX элементов третьих фирм;  
– интегрированная среда разработки и выполнения Workspace с дружественным интуитивно понятным интерфейсом пользователя и удобным доступом ко всем компонентам проекта через иерархическое дерево;  
– мастера и эксперты для выполнения команд и анимаций, не требующие программирования, с интуитивно понятным графическим интерфейсом для быстрой разработки проекта;  
– встроенная полнофункциональная среда разработки Microsoft Visual Basic for Applications;  
– доступ и контроль свойств внутренних и внешних объектов, взаимодействие с методами и событиями внедряемых элементов через мастера анимации или напрямую в VBA;  
– ActiveX копмоненты для работы с реляционными БД;  
– теговые группы для динамической перенастройки одной экранной формы для отображения типовых технологических объектов;  
– библиотека готовых к применению анимированных объектов и мастер для их создания;  
– возможность создавать и использовать глобальные переменные, глобальные таблицы цветов и значений, а также глобальные VBA-скрипты;  
– встроенный и выполняемый в отдельном потоке «Планировщик», который решает задачи по времени или по событию;  
– механизм кэширования мнемосхем в оперативной памяти компьютера для гарантированно высокой производительности отображения экранов;  
– возможность копирования готовых решений из экранов встронной демосистемы;  
– импорт/экспорт базы данных GE iFIX в офисные приложения для выполнения массовых изменений;  
– встроенные средства диагностики и отладки, интеграция с приложением для контроля изменений GE Change Management. Разработка приложений в режиме онлайн без остановки процесса управления и перезагрузки GE iFIX или компьютера.

1. Поддержка ОPC DA, ОРС Alarm & Events, SQL ODBC, COM/DCOM, OLE DB, ActiveX, .NET.
2. Используя ODBC-драйверы, можно получить доступ к базе данных GE iFIX из MS Access, MS SQL Server, Oracle, генератора отчетов Crystal Reports и др.
3. Сервер локальной истории на базе GE Historian for SCADA, который использует временные ряды с точностью временной метки 1 мкс и снабжен OLE DB-провайдером.
4. Контекстная интерактивная справочная система, возможность создания аналогичной справочной системы для приложения.
5. Поддержка технологии Plug and Solve™ для интеграции с приложениями третьих фирм. Возможность генерации отчетов любой сложности с помощью таких распространенных средств, как MS Excel, Crystal Reports, Dream Report, SyTech и др.
6. Развитые средства отладки и тестирования проектов: например, утилита «Управление выполнением», которая представляет собой графический интерфейс просмотра статистики работы сервисов и служб SCADA.
7. Полная SCADA-функциональность через технологию «тонких» веб-клиентов GE WebSpace. Без необходимости конвертировать экраны поддерживаются вся расширенная графика, анимация и VBA-скрипты для браузеров и мобильных устройств.
8. Специализированные пакеты для энергетики, водоподготовки и водоотведения, управления машинами и станками.

WinCC OA: основные функциональные возможности

Система WinCC OA предназначена для решения задач сбора и обработки данных, управления и визуализации – от уровня станка до уровня предприятия

1. Сбор, обработка и архивирование данных
2. Визуализация
3. Обработка событий
4. Формирование отчётов
5. Управление
6. Анализ данных
7. Техническое обслуживание
8. Администрирование
9. Возможность реализации собственных решений, ориентированных на государственные / отраслевые / корпоративные требования
10. Адаптация к меняющимся требованиям рынка, стандартов и бизнес-задач
11. Защита ноу-хау благодаря собственной разработке
12. Возможность внутренней стандартизации и обеспечения преемственности в эксплуатации / сопровождении корпоративных систем
13. Встроенный набор драйверов, протоколов и иных интеграционных средств позволяет WinCC OA выступать единым агрегатором, средой хранения и обработки данных, поступающих от различных информационных систем различных вендоров.
14. Возможность бесшовной интеграции различных географически распределённых систем управления технологическими процессами, систем управления производством, диспетчеризации, энергоменеджмента и др. систем в единое информационное пространство.
15. Стыковка с корпоративными информационными системами (MES, ERP, BI, PIMS, EMI…).
16. Стандартные и специализированные протоколы для организации межуровневого транспорта в т.ч. для каналов связи с низкой пропускной способностью.
17. Построение пользовательских интерфейсов в соответствии с современными стандартами и тенденциями в сфере HMI (в т.ч. поддержка видео, ГИС, мобильных платформ) для снижения нагрузки на человека-оператора и уменьшения вероятности ошибок при принятии решений.
18. Могут быть реализованы следующие задачи
    * отображение текущего состояния схемы электроснабжения;
    * измерение, контроль, отображение и регистрация параметров;
    * дистанционное управление переключением выключателей;
    * диагностика защит и автоматики с аварийной сигнализацией;
    * диагностика и контроль оборудования;
    * формирование базы данных, хранение и документирование;
    * технический учет электроэнергии и контроль энергопотребления;
    * передача информации о состоянии системы электроснабжения в технологическую АСУ, на ЦДП и в другие службы предприятия.
19. Могут быть реализованы следующие задачи
    * Сбор и хранение технологической информации
    * Отображение состояния производств в реальном времени
    * Интерфейс для управления производственными процессами
    * Система тревожной сигнализации
    * Формирование отчетности согласно регламентам предприятия
    * Анализ производственных процессов для повышения эффективности
    * Планирование производственной деятельности на основе анализа текущей и исторической технологической информации
    * Расчет KPI на основе реальных данных
    * Предоставление информации руководству в лаконичной и понятной форме
    * Передача управляющего воздействия от руководства непосредственно производству
    * Использование мобильных устройств
20. Ключевые особенности SIMATIC WinCC Open Architecture
    * Объектно-ориентированный подход к инжинирингу обеспечивает эффективность процесса разработки и возможность гибкого расширения систем
    * Возможность создания распределённых систем – до 2 048 серверов WinCC OA
    * Масштабируемость – от простой одиночной системы до распределённых резервированных высокодоступных систем с более чем 10 млн. сигналов ввода-вывода
    * Поддержка ОС Windows, Linux, iOS и Android
    * Горячее резервирование и резервирование центра управления (резервирование 2x2) позволяют обеспечить соответствие требуемому уровню надёжности и доступности системы
    * Платформа для создания специфических систем и приложений
    * Широкий спектр драйверов и возможностей обмена данными – OPC, OPC UA, S7, Modbus, Rockwell EtherNet/IP, IEC 61850, IEC 60870-5-101/104, DNP3, XML, TCP/IP, …