Лекция 2. Архитектура SCADA-систем.

Преподаватель: **Искакова Айгуль Малдыбековна**
Кафедры «Автоматизация и управление»

iskakova1979@mail.ru

Содержание

1. Архитектура SCADA-систем.
2. Архитектура программного обеспечения. Среда проектирования – компоненты общего назначения и специализированные.
3. Редактор проектов, специализированные библиотеки и средства конвертации.
4. Ведение архивов данных в SCADA системе. Тренды. Алармы.
5. Этапы развития человеко-машинного интерфейса. Этапы разработки АСУТП на основе SCADA.

2.1 Архитектура программного обеспечения. Среда проектирования – компоненты общего назначения и специализированные.

Диспетчерское управление реализуется с помощью SCADA-систем. Задачей таких систем является обеспечение автоматического дистанционного наблюдения и дискретного управления функциями большого количества распределенных устройств (находящихся на большом расстоянии друг от друга и от диспетчерского пункта). Количество возможных устройств, работающих под управлением систем диспетчерского контроля и управления, велико и может достигать нескольких сотен. Для этих систем наиболее характерной задачей является сбор и передача данных, которая реализуется дистанционно расположенными терминальными устройствами. На рисунке представлена архитектура комплекс технических средств многоуровневой системы управления технологическими процессами.

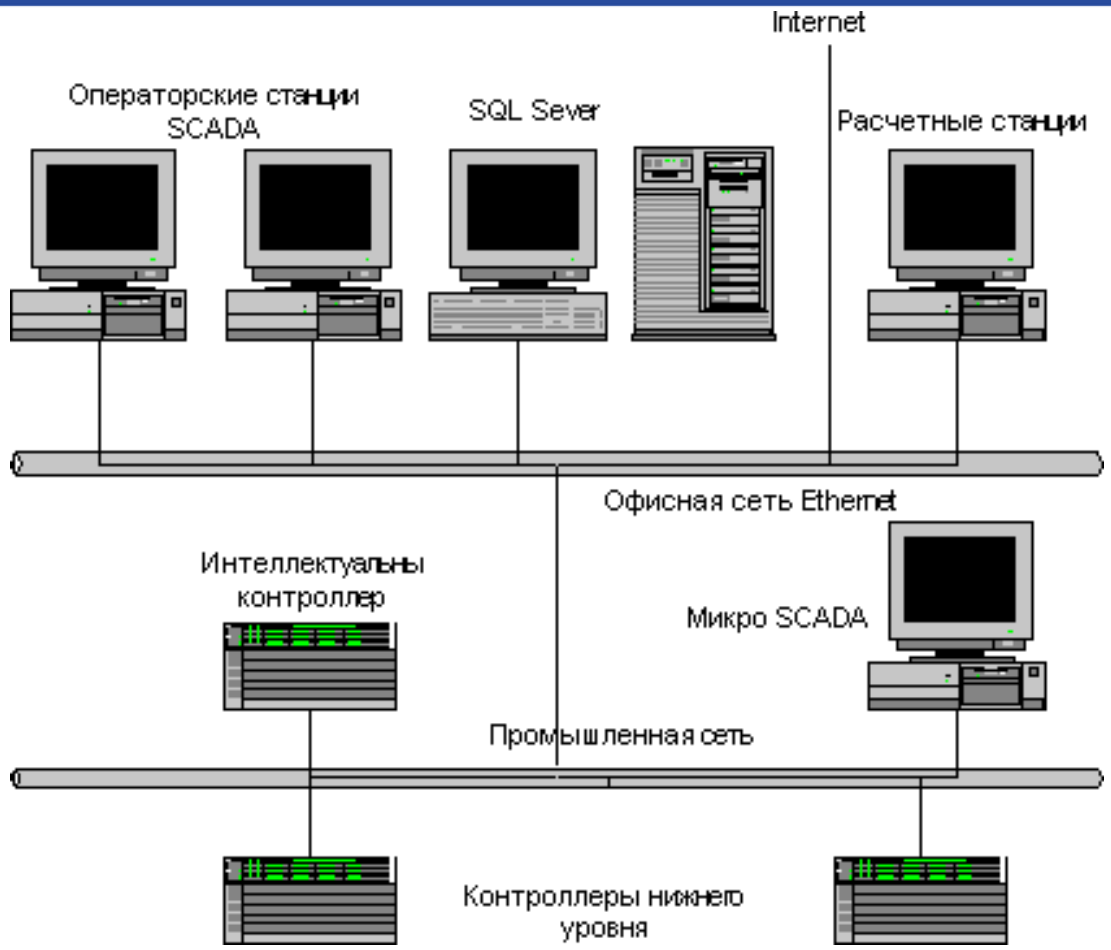


Рисунок 3. Архитектура SCADA-систем

- Как правило, это двух- или трехуровневые системы, так как именно на этих уровнях реализуется непосредственное управление технологическими процессами. Специфика каждой конкретной системы управления определяется используемой на каждом уровне программно - аппаратной платформой.
- Нижний уровень (полевой) – состоит из первичных датчиков, осуществляющих сбор информации о ходе технологического процесса, приводов и исполнительных устройств, реализующих регулирующие и управляющие воздействия, кабельных соединений, и нормирующих преобразователей.
- Средний уровень объекта (контроллерный) – состоит из контроллеров и устройств аналого-цифрового, цифро-аналогово, дискретного, импульсного и т.д. преобразования и устройств для сопряжения с верхним уровнем (шлюзов). Отдельные контроллеры могут быть объединены друг с другом при помощи контроллерных сетей. Контроллерные сети строятся на базе интерфейсов RS-232, RS-485, RS-422 или по сетевым протоколам и стандартам: Ethernet Fault Tolerant Ethernet (FTE), FTE Bridge, Profibus, Modbus и др.

- Датчики поставляют информацию локальным программируемым логическим контроллерам (PLC - Programming Logical Controoller), которые могут выполнять следующие функции:
- - сбор и обработка информации о параметрах технологического процесса;
- - управление электроприводами и другими исполнительными механизмами;
- - решение задач автоматического логического управления и др.
- Так как информация в контроллерах предварительно обрабатывается и частично используется на месте, существенно снижаются требования к пропускной способности каналов связи.
- В качестве локальных PLC в системах контроля и управления различными технологическими процессами. На рынке представлены многие десятки и даже сотни типов контроллеров, способных обрабатывать от нескольких переменных до нескольких сот переменных.
- К аппаратно-программным средствам контроллерного уровня управления предъявляются жесткие требования по надежности, времени реакции на исполнительные устройства, датчики и т.д. Программируемые логические контроллеры должны гарантированно откликаться на внешние события, поступающие от объекта, за время, определенное для каждого события.

- Для критичных с этой точки зрения объектов рекомендуется использовать контроллеры с операционными системами реального времени (ОСРВ). Контроллеры под управлением ОСРВ функционируют в режиме жесткого реального времени.
- Разработка, отладка и исполнение программ управления локальными контроллерами осуществляется с помощью специализированного программного обеспечения, широко представленного на рынке.
- К этому классу инструментального ПО относятся пакеты типа ISaGRAF (CJ International France), InConrol (Wonderware, USA), Paradym 31 (Intellution, USA), имеющие открытую архитектуру.
- Информация с локальных контроллеров может направляться в сеть диспетчерского пункта непосредственно, а также через контроллеры верхнего уровня (см. рис.). В зависимости от поставленной задачи контроллеры верхнего уровня (концентраторы, интеллектуальные или коммуникационные контроллеры) реализуют различные функции.

- Некоторые из них перечислены ниже:
- - сбор данных с локальных контроллеров;
- - обработка данных, включая масштабирование;
- - поддержание единого времени в системе;
- - синхронизация работы подсистем;
- - организация архивов по выбранным параметрам;
- - обмен информацией между локальными контроллерами и верхним уровнем;
- - работа в автономном режиме при нарушениях связи с верхним уровнем;
- - резервирование каналов передачи данных и др.
- Верхний уровень – (информационно-вычислительный) состоит и компьютеров, объединенных в локальную сеть Ethernet с использованием в качестве передающей среды медной витой пары или (при больших расстояниях) оптоволокна. Протокол передачи данных – для удаленных подключений TCP IP. Датчики с нижнего уровня поставляют информацию среднему уровню управления локальным контроллерам (ПЛК), которые могут обеспечить реализацию следующих функции:
- - сбор, первичную обработку и хранение информации о состоянии оборудования и параметрах технологического процесса;
- - автоматическое логическое управление и регулирование;
- - исполнение команд с пункта управления;
- - самодиагностику работы программного обеспечения и состояния самого контроллера;
- - обмена информацией с пунктами управления.

Станции управления предназначены для отображения хода технологического процесса и оперативного управления. Эти задачи и призваны решать SCADA - системы. SCADA - это специализированное программное обеспечение, ориентированное на обеспечение интерфейса между диспетчером и системой управления, а также коммуникацию с внешним миром.

Спектр функциональных возможностей определен самой ролью SCADA в системах управления и реализован практически во всех пакетах:

2.2. Редактор проектов, специализированные библиотеки и средства конвертации.

Встроенные технологические языки программирования это инструмент, который предназначен для решения новых задач на базе уже существующей системы контроля и управления технологическими процессами. Такими задачами являются:

- программно-логическое управление технологическим оборудованием;
- алгоритмы оптимального (рационального) управления;
- расчет значений переменных по формулам;
- визуализация значений трендов целевой обработки (т.е. построение каких-либо - расчетных трендов на базе полученных в результате измерений); в качестве исходных -трендов могут быть использованы тренды реального времени и/или исторические тренды;
-архивирование дат и времени определенных событий;
-создание сценариев динамики экрана;
-интегрирование мгновенных значений расхода в задачах дозирования;
-создание альтернативных фильтров входных переменных.

- В каждой SCADA -системе имеется встроенный набор стандартных алгоритмов, однако для - решения уникальных задач приходится прибегать к созданию собственных алгоритмов на встроенных языках. Большинство SCADA -систем имеют встроенные языки высокого уровня, подобные языку Visual Basic. Эти языки позволяют задать адекватную реакцию приложения на события, связанные с изменением значений переменных, с выполнением некоторого логического условия, с нажатием комбинации клавиш. Также возможно создание программных фрагментов, циклически выполняемых с заданной частотой.
- Встроенные языки программирования - мощное средство SCADA -систем, предоставляющее разработчику гибкий инструмент для разработки сложных приложений. Первые версии SCADA -систем либо не имели подобных языков, либо эти языки реализовывали небогатый набор функций. В современных версиях SCADA -систем функциональные возможности языков становятся существенно богаче.

- Можно выделить два подхода, используемых разработчиками встроенных в SCADA -системы языков программирования:
- ориентация языка программирования на потребности и задачи технолога;
- ориентация языка программирования на потребности и задачи системного интегратора.
- Ориентация встроенного языка на потребности и задачи технолога
- Функции в таких языках являются высокоуровневыми, не требующими профессиональных навыков программирования при их использовании. Количество таких функций в базовых поставках исчисляется сотнями, и существуют свободно распространяемые библиотеки дополнительных функций. Операции в подобных языках часто направлены на выполнение конкретных технологических операций, т.е. такие языки являются узкоспециализированными проблемно-ориентированными языками.

- **Ориентация встроенного языка на потребности и задачи системного интегратора**
- В этом случае в качестве языков чаще всего используются Visual Basic - подобные языки. В некоторых системах также используются языки на основе C++. Такой подход является наиболее универсальным, однако разработка программ для SCADA -систем на таких языках подчас весьма трудоемка.
- В каждом языке допускается расширение набора функций. В языках, ориентированных на технологов, это расширение достигается с помощью дополнительных инструментальных средств (Too1kits). Разработка дополнительных функций выполняется обычно программистами профессионалами.
- Разработка новых функций при втором подходе выполняется обычно разработчиками приложений (как и в традиционных языках программирования).
- Полнота использования возможностей встроенных языков (особенно при втором подходе) требует соответствующего уровня квалификации разработчика. Однако, большинство встроенных языков достаточно просты, и, поскольку они специально ориентированы на работу со SCADA - системой, освоение их элементарных возможностей обычно не представляет особых затруднений.

- Во всех языках функции разделяются на группы, часть из которых являются общеупотребительными: математические функции, функции работы со строками, обмен по SQL, DDE-обмен и т. д. Наряду с этими группами во многих SCADA -системах есть собственные уникальные наборы функций.
- В разрабатываемом приложении создаются программные фрагменты, состоящие из операторов и функций языка, которые выполняют некоторую последовательность действий. Эти программные фрагменты связываются с разнообразными событиями в приложении, такими как нажатие кнопки, открытие окна, выполнение некоторого логического условия. Каждое из событий ассоциируется с графическим объектом, окном, таймером, открытием/ закрытием приложения. Когда приложение содержит сотни окон, тысячи различных графических объектов, а с каждым из них связано несколько событий, в приложении может «работать» огромное количество отдельных программных фрагментов. Велика вероятность их одновременной активизации.
- Каждая из функций во встроенном языке выполняется в синхронном или асинхронном режиме. В синхронном режиме выполнение следующей функции не начинается до тех пор, пока не завершилось исполнение предыдущей. При запуске асинхронной функции управление переходит к ней, не дожидаясь завершения исполнения предыдущей функции.

- В связи с этим возникает несколько вопросов. С каким приоритетом исполняется каждый из фрагментов, допускается ли рекурсия при обработке событий и если да, то каков уровень вложенности? В SCADA -системах уровень вложенности пока не стандартизован, но оговаривается особо в рамках каждой из них.
- Разработанные программные фрагменты (часто их называют «скрипты») могут выполняться в режиме интерпретации, могут быть скомпилированы в исполняемый код, и выполняться, например, как вызовы функций из библиотеки DLL.
- Необходимо отметить, что в том и другом случае скрипты могут выполняться в том же потоке, что и основная часть программы, либо могут выполняться в отдельном потоке. Это зависит, в частности, от тина используемой SCADA - системы.

- **2.3. Механизм OLE for Process Control (OPC) как основной способ взаимодействия SCADA – системы с внешним миром.**

- Рассмотрим основные принципы взаимодействия SCADA – системы с внешним миром.
- Современные SCADA – системы не ограничивают выбор аппатратуры нижнего уровня, так как предоставляют большой выбор драйверов или серверов ввода- вывода и имеют хорошо развитые средства создания собственных программных модулей для драйверов нижнего уровня. Сами драйверы разрабатываются с использованием стандартных языков программирования.
- Для подсоединения драйверов к системе в настоящее время используются следующие механизмы:
- Ставший стандартом де-факто динамический обмен данными (DDE). Однако в современных SCADA – системах DDE применяется редко.
- Собственные протоколы, разработанные фирмами-производителями SCADA – систем. Достоинтсвом таких протоколов является самая высокая скорость обмена данными.
- Протокол OPC (OLE for Process Control). Данный протокол является стандартным и поддерживается большинством SCADA – систем.
- Новый стандарт обмен, ориентированный на задачи промышленной автоматизации - OPC (OLE for Process Control) – был разработан на базе механизма OLE. Стандарт OPC обладает следующими преимуществами:
- Позволяет объединить на уровне объектов различные системы управления и контроля, функционирующие в распределенной гетерогенной среде;
- Устраняет необходимость использования нестандартных протоколов обмена данными между устройством и SCADA – системой.

- Основная цель стандарта OPC заключается в создании универсального механизма доступа к любому аппаратному устройству из прикладной программы. OPC позволяет производителям оборудования поставить программные компоненты, которые стандартным способом обеспечивают связь ПО с технологическим контроллером.
- Таким образом, с точки зрения SCADA – систем, появление OPC – серверов означает разработку программных стандартов обмена с технологическими устройствами.
- OPC – интерфейс допускает различные варианты обмена:
- - получение данных с физических устройств;
- - обмен между частями распределенного приложения;
- - обмен между различными приложениями.
- В первую очередь в качестве серверов OPC выступают драйверы, написанные в соответствии со стандартом OPC и осуществляющие обмен данными с компонентами систем автоматизированного управления через соответствующие коммуникационное оборудование. Кроме того, любая программа, снабженная стандартным OPC – интерфейсом, может выступать в качестве OPC – сервера.

- Применительно к SCADA – системам, OPC-интерфейсом, расположенные на всех компютерах системы управления, стандартным образом могут поставлять данные в программу визуализации, базу данных и т.д.
- При обмене данными с OPC – сервером возможно два режима:
- Периодический режим, с заданной частотой данные запрашиваются OPC – клиентом;
- Режим обмена по изменению значения, когда обмен происходит при изменении значения, когда обмен происходит при изменении значения переменной на заранее заданную величину.
- Предпочтительным является второй тип обмена. При обмене через OPC-интерфейс данные передаются в виде особых структур, называемых пакетами, которые содержат следующие поля:
- Value (значение);
- Quality (качество);
- Timestamp (отметка времени).
- Такой пакет в терминологии называют «элемент данных». Поле Quality позволяет определить, не произошла ли ошибка в момент измерения величины или во время передачи данных.
- Во всех современных SCADA – системах при обмене данными осуществляется проверка поля Quality. Причем в различных системах реакция на «неудовлетворительное» значение качества получаемых данных может быть реализована по-разному. Обрабатывать поле Quality может либо приложение пользователя, либо сама SCADA – система.
- В рамках стандарта OPC все элементы данных объединяются в группы. Каждый элемент данных и группа имеют свое уникальное имя. Элементы данных и группы могут быть организованы в иерархическую структуру. Все элементы в каждой группе обновляются периодически, через равные промежутки времени, причем обновление элементов происходит синхронно.

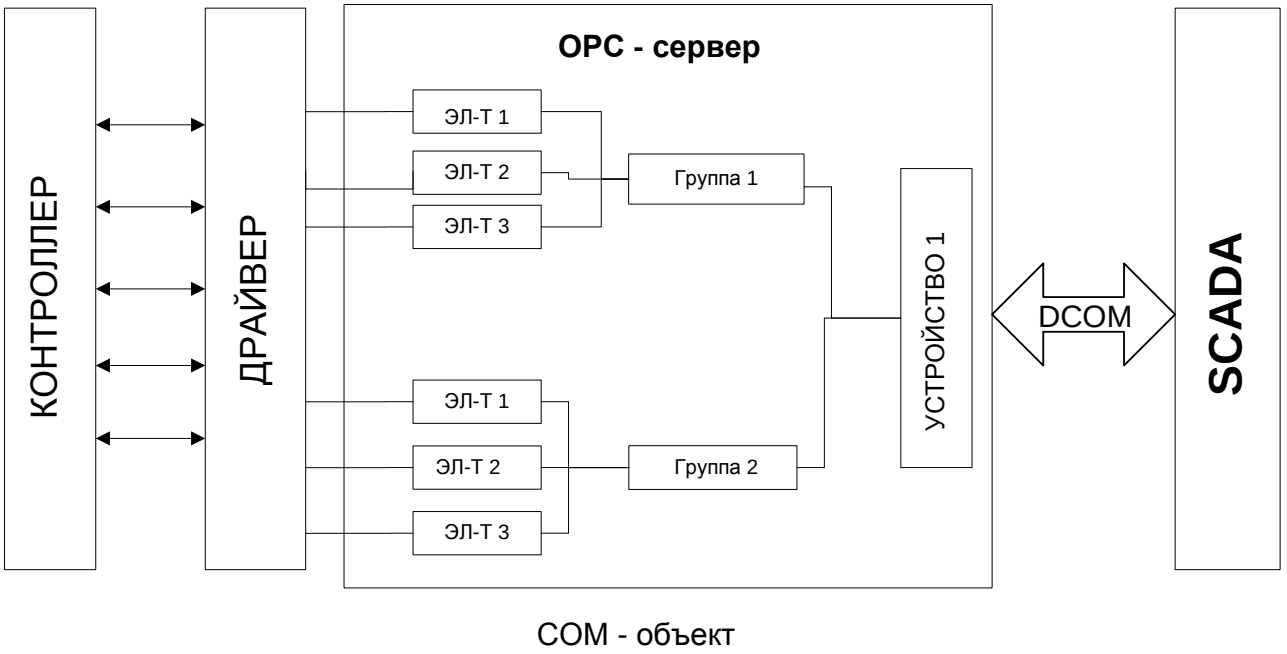


Рисунок. 4. Схема OPC-взаимодействия.

- Элементы данных часто называют тегами (ТАС). Именно эти тэги и являются технологическими переменными в SCADA -системе.
- OPC-сервер должен осуществлять буферизацию данных, запрашиваемых различными клиентскими приложениями, и оптимизировать их передачу так, чтобы коммуникация с физическими устройствами была наиболее эффективной. Буферизация данных необходима для того, чтобы исключить их потерю, и чтобы была _возможность их многократного считывания.
- Важным преимуществом OPC является возможность превращения системы управления в своего рода «конструктор», разнотипные элементы которого могут быть подключены к системе стандартным образом - через OPC-интерфейс.
- Использование технологии OPC в качестве процедуры обеспечения целостного доступа к производственным данным дает следующие преимущества:
- - производители устройств имеют возможность создания универсальных «переходников» от своего устройства к стандартизованному интерфейсу;
- - производители программного обеспечения SCADA могут ориентировать свои программные продукты на работу со стандартным интерфейсом, не зависящим от типа устройства; - потребители (заказчики) комплектуют свои системы такими устройствами и программным обеспечением, которое наиболее подходит для решения поставленных задач.
- Технологические устройства представляются управляющему ПО в виде серверов OPC и в общем случае являются «черными ящиками».

- Конечно, не бывает средства, решающего сразу все проблемы. OPC может использоваться только в тех операционных системах, где поддерживается механизм Мюгозой OCOM. В настоящее время к таким ОС относятся Windows NT/95/98, а также некоторые системы семейства UNIX.
- OPC не обеспечивает работу в жестком реальном времени, поскольку в DCOM отсутствуют понятия качества обслуживания, крайних сроков и т.д. В то же время контроль за «устареванием» данных имеется - каждое передаваемое значение сопровождается меткой времени. Несмотря на то, что требования жесткого реального времени, строго говоря, не выполняются, реальное время передачи данных порядка 50 миллисекунд достигается без всяких специальных мер.
- Не следует думать, что любое устройство можно просто так «через OPC» подключить к любой SCADA -системе - для этого надо иметь OPC-сервер для данного устройства. Сервер можно получить либо вместе с устройством, либо купить, либо написать самостоятельно. Для написания OPC-серверов в составе некоторых SCADA поставляется специальное ПО.

- **Ведение архивов данных в SCADA системе. Тренды. Алармы.**
-
- **Тренды.**
- Графическое представление изменения значений технологических параметров во времени способствует лучшему пониманию динамики технологического процесса предприятия.
- Подсистема создания трендов и хранения информации о параметрах с целью ее дальнейшего анализа и использования для управления является неотъемлемой частью любой SCADA - системы.
- Тренд - это упорядоченная совокупность значений технологической переменной, каждое из которых записывается в память компьютера через определенный интервал времени.
- Тренды реального времени (Real Time) отображают динамические изменения параметра в текущем времени. При появлении нового значения параметра в окне тренда происходит смещение графика.
- Тренд становится историческим (Historical) после того, как данные будут записаны на диск, и можно будет использовать режим прокрутки для просмотра предыдущих значений. Отображаемые данные тренда в таком режиме будут неподвижны, и будут отображаться только за определенный период. Различают часовые, сменные и суточные тренды, которые используются для печати отчетных документов за соответствующий период. Значения исторических трендов берутся из базы данных технологических параметров.
- Тренды реального времени являются динамическими объектами. Они позволяют выводить значения переменных по мере их поступления. Тренды реального времени могут создаваться как для конкретной переменной, так и для выражения, которое содержит одну или несколько переменных. Данные будут появляться в окне тренда и двигаться справа налево.

- Исторические (архивные) тренды не являются динамическими. Они обеспечивают «снимок» состояния данных за прошедшее время на основе архивных данных. В отличие от трендов реального времени исторические тренды обновляются только по команде - при запуске скрипта, изменении значения выражения или нажатии оператором соответствующей кнопки.
- При конфигурировании архивного тренда можно создать «визиры» (типа ползунков, бегунков), с помощью которых удобно получать значения всех отображаемых на одном графике переменных в один и тот же момент времени. Бегунки архивного тренда представляют собой позиционные индикаторы на временной оси, положение которых определяет объем извлекаемых данных. Связав объект «движковый регулятор» с полем бегунка, можно осуществлять перемещение вдоль архивного тренда. Кроме того, имеются функции вычисления среднего, минимального и максимального значений в определенном бегунком положении. Можно создать правый и левый бегунки и проводить обработку данных кривой, расположенной между бегунками.
- Благодаря системе распределенных архивов на один и тот же график можно выводить информацию из нескольких баз данных. Необходимо отметить, что на один и тот же график могут быть выведены несколько трендов реального времени.

- ПРИМЕР
- TrendWordX32 - открытое решение по высокопроизводительному построению графических зависимостей контролируемых технологических параметров от времени либо от других параметров. Поддерживает спецификацию OPC для доступа к историческим данным (OPC HDA), устанавливающую требования к подсистеме извлечения и представления исторических данных из баз данных технологических архивов.
- Пакет TrendWordX32 обеспечивает накопление и представление текущих данных в виде графических зависимостей от времени. Кроме того, TrendWordX32 является мощным средством архивации накапливаемой информации в базах данных с возможностью последующего извлечения и просмотра на графиках. Полностью соответствует спецификациям OPC доступа к текущим и историческим данным.
- Основные функциональные возможности TrendWordX32:
- представление значений контролируемых параметров, получаемых от серверов OPC на графиках различных типов в реальном масштабе времени;
- архивирование значений контролируемых параметров в базах данных MS SQL Server, MS Access, Orac1e, MSDE;
- генерация отчетов на основе данных архива и публикация отчетов в Интернет;
- вычисление статистических характеристик выборок значений контролируемых параметров;
- извлечение значений контролируемых параметров из архивов и представление в виде графиков различных типов;
- вывод графиков на печатающее устройство;
- разработка и исполнение сценарных процедур VDA;
- возможность вставки элементов просмотра графиков TrendWordX32 AtiveX в различные контейнеры AtiveX;
- встроенные средства генерации отчетов в базах данных и MS Exce1.

- **Алармы.**
- Состояние тревоги, в дальнейшем аларм (A1arm) - это некоторое сообщение, предупреждающее оператора о возникновении определенной ситуации, которая может привести к серьезным последствиям, и потому требующее его внимания и вмешательства.
- В системах управления принято различать неподтвержденные и подтвержденные алармы. Аларм называется подтвержденным после того, как оператор отреагировал на сообщение об аларме. До этого аларм оставался в состоянии неподтвержденного.
- Наряду с алармами, в SCADA -системах существует понятие событий. События представляют собой обычные статусные сообщения системы и не требуют реакции оператора. Обычно событие генерируется при возникновении в системе определенных условий (типа регистрации оператора в системе).

- От эффективности подсистемы алармов зависит скорость идентификации неисправности, возникшей в системе, или определения технологического параметра, вышедшего за установленные регламентом границы. Быстродействие и надежность этой подсистемы могут существенно сократить время простоя технологического оборудования. Например, если оператор не получит вовремя информацию о том, что двигатель насоса перегрелся, это может привести в лучшем случае к выходу насоса из строя, а иногда и к крупной аварии.
- Причины, вызывающие состояние аларма, могут быть самыми разными. Неисправность может возникнуть в самой SCADA -системе, в контроллерах, каналах связи, в технологическом оборудовании. Может выйти из строя датчик или нарушатся его метрологические характеристики. Параметры технологического процесса могут выйти за границы, установленные регламентом и т. д.
- Подсистема алармов - это обязательный компонент любой SCADA -системы. Но возможности подсистем алармов различных SCADA -систем различны. С другой стороны, когда речь идет о типах алармов, то все SCADA -системы поддерживают такие типы алармов, как дискретные и аналоговые. Такие алармы называют типовыми.

- **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, в зависимости от конкретной SCADA - системы.
- **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).

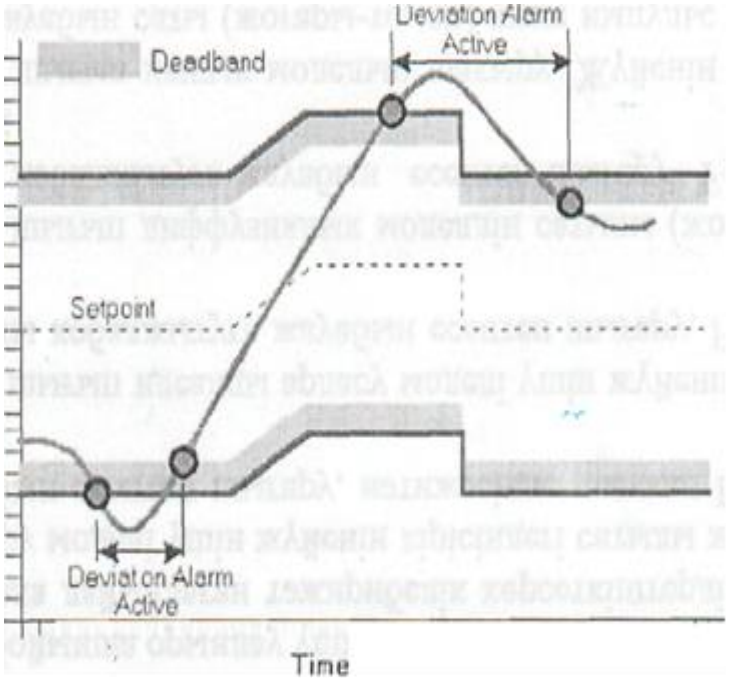


Рисунок. 5 - Графическая интерпретация алармов типа Deviaition.

- Состояние тревоги, в дальнейшем аларм (Alarm) - это некоторое сообщение, предупреждающее оператора о возникновении определенной ситуации, которая может привести к серьезным последствиям, и потому требующее его внимания, а часто и вмешательства. А принял - ли оператор сообщение об аларме? Чтобы снять эти сомнения, в системах управления принято различать неподтвержденные и подтвержденные алармы. Аларм называется подтвержденным после того, как оператор отреагировал на сообщение об аларме. До этого аларм оставался в состоянии неподтвержденного. Наряду с алармами в SCADA - системах существует понятие событий.

- Алармы типа «Rate of Change» возникают, когда скорость изменения параметра становится больше предельно допустимой. Понятие «зона нечувствительности» (Deviaition) к алармам этого типа не применяется,
- Обычно применяется стандартная и распределенная системы алармов.
- **Стандартная система алармов** используется для отображения информации о состоянии тревоги и подтверждения реакции на все аварийные ситуации и события, регистрируемые в данном локальном приложении (локальном АРМ).
- **Распределенная система алармов** расширяет возможности стандартной и позволяет отслеживать и подтверждать аварийные ситуации, генерируемые системами алармов других включенных в сеть приложений (соседних АРМов).

- SCADA - системы поддерживают возможность отображения, регистрации и печати информации как об алармах, так и о системных событиях.
- События, также как и алармы, делятся в зависимости от их характеристик на несколько общих категорий, называемых типами событий (Event Types). Различают следующие типы событий:
- АСК - 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 оператором;
- DDE - получено новое значение переменной от OOE-клиента;
- LGC - скрипт изменил значение переменной;
- ОРК - оператор ввел новое значение переменной.
- Создавая обработчики (т.е. задавая реакции) для данных событий, можно гибко влиять на поведение системы.

- **Приоритеты алармов**.
- Каждому аларму, как правило, соответствует некоторая величина, называемая приоритетом аларма. Этот приоритет характеризует важность данного аларма и может принимать значения из некоторого диапазона (например, от 1 до 999). Самый важный аларм имеет приоритет 1. Если одновременно возникает несколько алармов, то они будут выполняться последовательно, в порядке возрастания значения приоритета. Организовав несколько диапазонов значений и связав алармы с каждым диапазоном, можно достаточно легко отфильтровать критические алармы от некритических. Выполнение анимационных функций, скриптов, печать и просмотр информации также могут зависеть от приоритетов.
- **Группы алармов.**
- Каждый аларм может быть связан с определенной логической группой алармов. Все эти группы определяются пользователем и могут быть организованы в иерархическую структуру. 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 дисплей 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. С любой группой алармов можно связать как переменную, так и другую группу алармов. Взаимосвязи всех групп алармов представляются древовидной структурой.

- *ПРИМЕР*
- A1armWorX32 - развитая система обнаружения, идентификации, фильтрации и сортировки аварийных и других событий, связанных с контролируемым технологическим процессом и состоянием технических средств АСУТП.
- A1armWorX32 является набором программных компонентов, предназначенных для обнаружения аварийных событий, оповещения оперативного персонала, приема подтверждений восприятия информации об аварийных событиях и регистрации информации об авариях в базе данных.
- Основные функциональные возможности A1armWorX32:
- - обнаружение аварийных событий по множеству признаков и критериев, настраиваемых пользователем;
- - передача информации об обнаруженных авариях клиентским приложениям, расположенным на разных узлах локальной или глобальной сети;
- - анализ аварийных событий и действий ответственного персонала;
- - объединение всех аварийных событий и подтверждений восприятия системных сообщений ответственным персоналом в сводки аварийных событий;
- - отображение вспомогательной информации для аварийных событий, позволяющей локализовать и устранить причины аварии;
- - связь с аппаратными средствами системы через интерфейсы OPC;
- - возможность запуска сервера обнаружения аварий в качестве сервисного процесса (службы) Windows NT;
- - имеется мощное средство конфигурирования условий аварийных событий;
- - встроенная среда редактирования сценарных процедур VBA..
-

Контрольные вопросы:

1. Общая структура АСУТП на основе SCADA.?
2. Функции SCADA на уровне контроллеров и датчиков.
3. Разработка и отладка управляющих программ на уровне контроллеров и датчиков.
4. Техническое обеспечение уровня цеха.
5. Функциональные возможности SCADA на уровне цеха.

Основная литература

1. Андреев Е. Б., Куцевич Н. А. Синенко О. В. SCADA-системы: взгляд изнутри/ Е. Б. Андреев, Н.А.Куцевич, О.В.Синенко.—М.: издательство РТСофт, 2004.
2. Букреев В. Г., Цхе А. В. Основы инструментальной системы разработки АСУ/ В. Г. Букреев, А. В. Цхе.— Томск: издательство ТПУ, 2003.
2. Локотов А. Что должна уметь система SCADA/ А. Локотов// Современные технологии автоматизации.— 1998.
3. Анизимиров Л., Айзин В., Фридлянд А. Новая версия Trace Mode для Windows NT/ Л. Анизимиров, В. Айзин, А. Фридлянд// Современные технологии автоматизации.— 1998.
4. Анизимиров Л. Windows- компоненты Trace Mode 4.20/ Л. Анизимиров// Современные технологии автоматизации.— 1996.
5.Волобуев Ю. АСУ ТП в металлургии: проблемы и решения/ Ю.Волобуев Ю.// Современные технологии автоматизации.— 2000.
6. Кузнецов А. SACADA- системы: программистом можешь ты не быть/А. Волобуев// Современные технологии автоматизации.— 1996.

4.2 Дополнительная литература

1.Андреев Е.Б., Куцевич Н.А. SCADA-системы: взгляд изнутри. /Электронная публикация на сайте "SCADA.ru“

2.Открытые системы: концепция или реальность? //Открытые системы. 1993. №4.

3. Куцевич Н.А. Новые технологии и MMI-системы. //Мир компьютерной автоматизации. 2000

4. Crater K.C. An Argument for Open Source Software — Sensors, 2002, Vol. 19, No. 1.

5.Цифровая обработка сигналов./Сергиенко А.Б. -Спб.: Питер, 2002.

6.Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации алгоритмов. - Спб.: Наука, 2000.

7.Степоненко О. С. Компьютер внутри. Киев «Логос» 1998.

8.Web – ресурсы http://www.scada.ru

9. http://itcobe.web.cern.ch/itcobe/welcome.html

10http://web.ni.com

11http://public.web.cern.ch/Public/Welcome.html

12. http://www.vitec.ru