УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите и законспектируйте новый теоретический материал тезисно по плану лекции.

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: xvsviv@rambler.ru в трехдневный срок с момента получения задания.

При возникновении вопросов по приведенному материалу

обращаться по следующим номерам телефонов:072-138-93-11.

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Лекция. Управление технологическими процессами. Общая схема типовой информационной системы

Часть 1

План

Основные этапы развития и современные принципы построения информационных систем. Распределенные системы. Закрытые системы. Открытые системы. Требования к информационной системе технологических процессов. Особенности программного обеспечения технологических процессов.

Основные этапы развития и современные принципы построения информационных систем

Непрерывную во времени картину развития информационных систем можно разделить на три этапа, обусловленных появлением качественно новых научных идей и технических средств. В ходе истории меняется характер объектов и методов управления, средств автоматизации и других компонентов, составляющих содержание современной системы управления.

- Первый этап отражает внедрение систем автоматического регулирования (САР). Объектами управления на этом этапе являются отдельные параметры, установки, агрегаты; решение задач стабилизации, программного управления, слежения переходит от человека к САР. У человека появляются функции расчета задания и параметры настройки регуляторов.
- Второй этап автоматизация технологических процессов. Объектом управления становится рассредоточенная в пространстве система; с помощью систем автоматического управления (САУ) реализуются все более сложные законы управления, решаются задачи оптимального и адаптивного управления, проводится идентификация объекта и состояний системы. Характерной особенностью этого этапа является внедрение систем телемеханики в управление технологическими процессами. Человек все больше отдаляется от объекта управления, между объектом и диспетчером выстраивается целый ряд измерительных систем, исполнительных механизмов, средств телемеханики, мнемосхем и других средств отображения информации (СОИ).
- Третий этап автоматизированные системы управления технологическими процессами – характеризуется внедрением в управление технологическими процессами вычислительной техники. Вначале применение микропроцессоров, использование на отдельных фазах управления вычислительных систем; затем активное развитие человеко-машинных систем управления, инженерной психологии, методов и моделей исследования операций и, наконец, диспетчерское управление на основе использования автоматических информационных систем сбора данных и современных вычислительных комплексов.

Современная идеология построения сложных автоматизированных систем управления технологическими процессами уверенно развивается в направлении

применения распределенных принципов построения систем в противоположность централизованным.

Распределенную систему управления (PCV, DCS – Distributed Control System) можно определить как систему, состоящую из множества устройств, разнесенных в пространстве, каждое из которых не зависит от остальных, но взаимодействует с ними для выполнения общей задачи.

Распределенная система имеет следующие характеристики, отличающие ее от сосредоточенной:

- большее быстродействие благодаря распределению задач между параллельно работающими процессорами;
- повышенная надежность (отказ одного из контролеров не влияет на работоспособность других);
- большая устойчивость к сбоям;
- более простое наращивание или реконфигурирование системы;
- упрощенная процедура модернизации;
- большая простота проектирования, настройки, диагностики и обслуживания.

Такие АСУ ТП содержат компьютерные сети. Следовательно, распределенные АСУ ТП характеризуются физической, логической и функциональной структурами, единство которых определяет архитектуру АСУ ТП. В результате централизованные архитектуры уступили свои позиции на рынке, в то время как открытые распределенные системы являются основой построения систем. Приведем три основные предпосылки перехода разработчиков систем АСУ ТП к применению распределенных сетевых технологий.

- 1. Изделия из кремния дешевеют, а изделия из меди дорожают. За последние годы эта тенденция стала особенно заметна. Прошли те времена, когда нормой жизни считался огромный шкаф, напичканный автоматикой, с выходящими из него толстыми пучками кабелей, ведущими к датчикам и исполнительным механизмам. Сегодня экономически целесообразно установить в цехе или на участке нескольких локальных контроллеров или УСО, объединенных в единую сеть, чем прокладка разветвленных кабельных систем.
- 2. Стоимость работ по установке, тестированию, вводу в эксплуатацию и сопровождению централизованной системы горазда выше, чем распределенной. Количество проводных соединений в централизованной у распределенной. Системы значительно больше, чем централизованным управлением обычно требуют, чтобы каждый датчик или группа датчиков подключались к центральному компьютеру отдельным и довольно дорогим высококачественным кабелем. В противоположность этому в распределенной системе рядом с каждой группой датчиков располагается один интеллектуальный узел, преобразующий сигналы датчиков в цифровую последовательность и передающий их в этом виде в систему управления. При этом прокладка кабелей и развертывание систем обходится значительно дешевле. Необходимо учитывать многократно возрастающую вероятность ошибки при монтаже проводов в многочисленных кроссовых клеммных колодках и сложность поиска и устранения неисправностей. Особо следует отметить ситуацию, когда в составе объекта управления появляется еще несколько входных или выходных каналов. Добавление новых линий связи к уже проложенной кабельной системе – занятие не из простых.
- 3. Растущая потребность в «распределенном интеллекте». Стремительное удешевление вычислительной техники, позволившее применять автономные вычислители в каждом из узлов АСУ ТП в непосредственной близости от исполнительных устройств и датчиков. Сегодня, когда микропроцессоры и другие специализированные микросхемы дешевеют, становится

целесообразным выделять в общей системе АСУ отдельные локальные задачи, решение которых поручать локальным контроллерам. Контур управления, таким образом, замыкается на нижнем уровне. Сеть же позволяет контроллерам в качестве аргументов для вычисления вектора использовать переменные других контроллеров, обеспечивая связанность системы управления в целом.

Такая архитектура существенно увеличивает производительность, надежность, живучесть и масштабируемость систем. Она также открыла принципиально новые возможности распределения функций в децентрализованных системах. При этом децентрализация предполагает не только рассосредоточение аппаратуры, но и распределение функций обработки информации и управления между автономными компьютерами.

Сегодня никакой производитель не может поставлять всю номенклатуру изделий, используемых в современных информационных системах. От специалистов требуется умение применять для построения информационных систем высокотехнологичные изделия различных компаний, и, естественно, эти изделия должны быть совместимыми. Для гарантии совместимости различных производителей необходимы открытые стандарты аппаратных и программных средств. Помимо аппаратной составляющей для совместимости необходимо использование программных платформ с открытым кодом (на контроллерах применяют ОС на базе Linux, например, BusyBox, также сейчас получает развитие платформа от Microsoft .Net MicroFramework, которая также является открытой).

Системы, являющиеся уникальными (их делает и поддерживает только один производитель), работающие по уникальным протоколам связи, получили название «Закрытые системы» (closed systems) (например, Siemens). Большинство таких систем зародилось во времена, когда проблема интеграции изделий других производителей не была актуальной.

«Отверытые системы» (open systems) приводят специфические требования в соответствие интересам всех. Только при использовании принципов открытых систем интеграция изделий разных производителей в одну сеть может быть решена без особых проблем. Открытая система — это модульная система, которая допускает замену любого модуля на аналогичный модуль другого производителя, имеющийся в свободной продаже по конкурентоспособным ценам, а интеграция систем с другими системами выполняется без преодоления чрезвычайных мер.

Открытость означает:

- отсутствие патентов или авторских прав на спецификацию стандарта и его расширений;
- отсутствие лицензионной платы за использование стандарта;
- широкий доступ к спецификациям стандарта и его расширениям;
- получение спецификаций в результате открытого обсуждения и конкурса между экспертами крупнейших промышленных ведущих мировых фирмпроизводителей и пользователей;
- принадлежность прав собственности некоммерческим профессиональным национальным и международным организациям.

Открытость можно рассматривать на разных уровнях иерархии программного и аппаратного обеспечения системы или ее составных частей. Открытыми, например, могут быть:

- физические интерфейсы, протоколы обмена, методы контроля ошибок, системы адресации, форматы данных, типы организации сети, интерфейсы между программами;
- пользовательские интерфейсы, языки программирования, управляющие команды модулей ввода-вывода, языки управления базами данных, операционные системы и т.д.;
- конструктивные элементы: шкафы, стойки, разъемы и т.п.;

- системы, включающие в себя все перечисленные выше элементы.
- Открытые системы обладают следующими положительными свойствами:
- модульность;
- платформенная независимость;
- взаимозаменяемость с компонентами других производителей;
- интероперабельность (возможность совместной работы с компонентами других производителей);
- масштабируемость.

В отличие от открытых, закрытые системы разрабатываются по внутренним стандартам отдельных предприятий. Части (модули) закрытых систем не могут быть заменены аналогичными изделиями других производителей, а заказчик, однажды применив закрытую систему, оказывается привязанным к ее разработчику систем. Отметим, что закрытые системы тоже могут быть модульными, интероперабельными, масштабируемыми. Отличие открытых систем состоит в том, что все перечисленные свойства должны выполняться для компонентов, изготовленных разными производителями и имеющихся в свободной продаже.

С развитием технических и программных средств на крупных предприятиях изменяются и подходы к разработке систем АСУ ТП. Система АСУ ТП представляется как ядро, отвечающее за сбор, первичную обработку и обмен информации с датчиков, преобразователей и исполнительных механизмов. Она также включает набор сервисных программ, отвечающих за организацию хранения, доступа и отображения информации пользователям системы. Высокая надежность средств вычислительной техники и PLC повысила отказоустойчивость АСУ ТП в целом. Это обстоятельство, а также возможность создания единого информационного пространства на базе компьютерных сетей внутри крупного цеха приводит к необходимости интеграции систем АСУ ТП отдельных агрегатов с системами цехового уровня (системами АСУП). Сегодня расширяется функциональность и место системы АСУП цеха в корпоративных системах управления на крупных предприятиях России. Так, в рамках внедрения корпоративной информационной системы управления предприятием в цехах вводятся в эксплуатацию такие системы, как система управления кадрами, система бухгалтерского учета, система учета затрат на производство, система планирования и учета затрат на ремонты и другие. Таким образом, система АСУП цеха становится основным поставщиком оперативной и достоверной информации о производственной деятельности цеха в системы корпоративного уровня. Развитие сетевой инфраструктуры внутри цеха и интеграция ее в корпоративную сеть предприятия приводит к существенному росту количества пользователей системы, как среди технологического персонала цеха, так и корпоративных клиентов. Это ставит новые задачи по администрированию систем, организации защиты информации от несанкционированного доступа, защиты информации от вирусных атак.

Таким образом архитектура современных информационных систем разрабатывается на принципах:

- стандартности;
- модульности;
- распределенности;
- открытости;
- обеспечения требуемыми метрологическими характеристиками измерительных каналов во всем диапазоне внешних воздействий окружающей среды;
- способности работать в широком диапазоне температур, вибраций, электромагнитных электрических и сетевых помех;
- обеспечения высокой надежности (время наработки на отказ 100000 часов и более);

• наличия набора измерительных средств, сопрягаемых с обширным парком датчиков и исполнительных механизмов.

Наряду с миниатюризацией аппаратуры как черты научно-технического прогресса, имеющей необратимый, эволюционный характер, в последний период наметился скачкообразный переход к микроминиатюризации на базе микроэлектронной технологии. микроминиатюрные твердотельные преобразователи, Создаются датчики, однокристальные вычислители и исполнительные элементы с малым энергопотреблением, высокой надежностью, большим ресурсом, высоким быстродействием, низкой стоимостью. Микроминиатюрная элементная база сделала возможным создание распределенных систем контроля с высокой информативностью и надежностью, сенсоры которой способны сами создавать себе условия, благоприятные для функционирования. Для сложных технологических объектов замена десятков и сотен традиционных ПИ- и ПИД- регуляторов настройками однокристальные микропроцессорные c постоянными на самоорганизующиеся регуляторы поднимет системы управления производством на новый уровень развития.

Задача синтеза конкретной системы сводится к оптимизации выбора оборудования: промышленных контроллеров, управляющих вычислительных комплексов, коммуникационного оборудования, каналов связи, рабочих станций. Цель синтеза — оптимальное решение заданной совокупности задач АСУ ТП в установленное время на известном множестве оборудования с заданными характеристиками и с учетом материальных затрат.

В особо ответственных информационных системах требуется их отказоустойчивость. Отказоустойчивые информационные системы:

- во-первых, допускают проведение ремонта без нарушения непрерывности своей работы;
- во-вторых, сохраняют работоспособность при отказе любого устройства комплекса.

Отказоустойчивость достигается введением избыточности – аппаратурной, программной, временной.

Анализ областей применения промышленных автоматизированных систем управления, требующих высокой оперативности и надежности при круглосуточной эксплуатации, показывает, что управляющие вычислительные комплексы в подобных системах должны:

- иметь функционально-модульную структуру, легко конфигурироваться и резервироваться, а в случае необходимости наращиваться по числу процессоров, объему оперативной и внешней памяти, количеству и номенклатуре средств телекоммуникаций и средств отображения информации;
- обеспечивать высокую производительность: предельно высокую реактивность режимов реального времени на нижнем уровне управления при количестве обрабатываемых каналов ввода-вывода, доходящем, как правило, до нескольких тысяч, а также время реакции на запрос оператора с клавиатуры при оперативном диспетчировании до 0,5 с;
- соответствовать требованиям отказоустойчивости, не допускать при непрерывном круглосуточном режиме эксплуатации снижения производительности при возникновении неисправностей и непрерывной дистанционной диагностике работоспособности комплекса при времени обнаружения неисправности порядка секунды и времени ликвидации не более часа;
- отвечать обобщенным требованиям промышленного исполнения, т.е. обладать устойчивостью в процессе эксплуатации в тяжелых промышленных условиях;
- быть масштабируемыми для построения многоуровневого распределения иерархических систем на единой аппаратно-программной платформе;
- отвечать стандартам и требованиям открытых систем, обеспечивающим их интеграцию с аппаратными и программными средствами различных производителей;
- обеспечивать интеграцию в локальные офисные и промышленные сети;

• иметь базовое программное обеспечение и инструментальные средства проектирования отказоустойчивых АСУ ТП в различных операционных системах.

Особенности программного обеспечения технологических процессов

Естественно, специфика промышленных применений в металлургии наложила свой отпечаток на используемое программное обеспечение. Основные требования к программному обеспечению сводятся к следующему:

- Первым требованием является надежность программного обеспечения. Действительно, одно дело, когда у вас «зависает» редактор текста в офисе, а другое дело, когда неправильно работает программа, управляющая прокатным станом, разливкой стали или загрузкой материалов в доменную печь.
- Вторым требованием является быстрое реагирование на какие-либо внешние события или изменения в параметрах управляемых процессов. Системы, работающие в соответствии с этим требованием, относятся к системам реального времени. Многие процессы требуют высокого быстродействия системы управления. Рассмотрим, например, регулирование скорости прокатного стана. Работу различных двигателей и механизмов прокатного стана необходимо синхронизировать с высокой точностью, в противном случае стальная полоса может либо порваться, либо значительно прогнуться. Идея управления заключается в некотором ослаблении натяжения стальной полосы в течение всего процесса. Высокая скорость движения полосы (10-100 м/с) обуславливает необходимость распознать изменение скорости любого двигателя в пределах нескольких миллисекунд с последующей коррекцией скорости других двигателей.
- Третьим требованием, часто предъявляемым к программному обеспечению систем управления, является многозадачность. Это требование проистекает из-за сложной и многоуровневой природы управляемых процессов в металлургии, когда необходимо одновременно реализовать сложные алгоритмы управления различными подсистемами реального объекта. Каждая задача выполняет свою долю работы по управлению объектом, и все они делят между собой ресурсы вычислительной системы в зависимости от своего приоритета и от внешних и внутренних событий, связанных с конкретной задачей.
- Четвертое требование управление по внешним событиям (прерываниям). Особенность компьютерного управления процессом заключается в том, что ход исполнения программы, порядок выполнения операторов программы нельзя определить заранее. Внешние сигналы могут прерывать или изменять последовательность исполнения операторов программы. Кроме того, существует проблема эффективного использования ресурсов компьютерной системы с учетом временных ограничений. Все это требует специальных методов программирования. Дополнительную проблему представляет собой тестирование систем реального времени из—за отсутствия предсказуемого порядка выполнения.
- Пятое требование параллельное управление процессами. Компьютер, взаимодействующий с такими объектами или управляющий им, должен учитывать эту параллельную природу, а в некоторых ситуациях и работать в соответствии с ней, компьютер должен управлять параллельными процессами.
- Шестое требование открытость возможность модификации и расширения программного обеспечения, а также гибкость возможность быстрой перенастройки программ управления и внесения изменений в технологические базы данных.
- Седьмое требование функциональность обеспечивается благодаря использованию эффективных математических моделей, интеллектуальных систем. В основу программного обеспечения положены современные достижения в областях теории и технологии математического моделирования и оптимального управления техническими системами.

- Восьмое требование интуштивно понятный интерфейс, обеспечивающий легкое и быстрое восприятие информации человеком, за которым остается «последнее слово» в принятии решения, а также быстрый и безошибочный ввод информации. АРМ технологического персонала работают под управлением ОС семейства Windows, поэтому в основу пользовательского интерфейса положены диалоговые элементы: окна, строковые опускающиеся меню, панели инструментов; используются элементы визуализации и структурирования обработанных данных графики, диаграммы, таблицы и т.п.
- Девятое требование расширяемость и открытость систем достигается за счет модульной структуры и использования стандартных интерфейсов. Каждый из модулей имеет четко очерченный круг функций, минимально взаимодействует с другими модулями и имеет возможность расширения независимо от других модулей. Для организации взаимодействия между приложениями используются технологии DDE, OLE, COM, DCOM и другие (см. далее). Для построения модульной структуры используется принцип декомпозиции задач и моделей. Для реализации этого принципа используются средства автоматизированного проектирования информационных систем CASE-средства (см. далее).. Для функционального моделирования Технологических процессов в этих системах используются методологии проектирования IDEF0 и IDEF1X (см. далее).
- Десятое требование отделение и интеграция данных от программ. Одной из характерных особенностей разрабатываемого в настоящее время программного обеспечения для управления сложными металлургическими процессами является использование больших массивов данных для получения и представления новой информации о ходе технологического процесса в реальном времени. Для решения этих задач используются распределенные трехуровневые приложения баз данных Windows DNA, в которых, в отличие от клиент/серверных приложений, клиенты не имеют непосредственного доступа к критическим ресурсам, таким как соединения с базой данных. Поэтому такие ресурсы полностью защищены от непредсказуемых действий пользователя.
- Одиннадцатое требование оценивание информации. Для металлургического производства характерной особенностью использования в приложениях информации из распределенных баз данных является тот факт, что эта информация не является полной. Причиной этому служит зашумленность информации, ее запаздывание, а также отсутствие технических средств контроля информации. Поэтому в разработанных программных продуктах предусмотрена подсистема оценивания информации и соответствующее программное обеспечение.
- Двенадцатое требование безопасность. Защита от несанкционированного доступа осуществляется путем использования подсистемы распределения прав доступа, а защита от непреднамеренного неверного ввода информации обеспечивается функционированием специально разработанной модели, которая оценивает введенную информацию по интегральным показателям.

Названным требованиям должны удовлетворять все уровни программного обеспечения (BIOS, операционные системы, прикладные программы).

Ведущие фирмы-поставщики ориентируются в основном на рынок офисных компьютеров и не применяют специальных мер для обеспечения требований индустриальных приложений. Основная опасность при обращении к BIOS — это возможность запрета прерываний на достаточно долгое время, в результате чего может быть пропущена важная информация от быстродействующих датчиков и телекоммуникационных устройств. В связи с этим имеются фирмы, предлагающие ориентированные на приложения «жесткого» реального времени, а с другой стороны, многие операционные системы минимизируют взаимодействие с BIOS.

Задачи реального времени составляют одну из сложнейших и крайне важных областей применения вычислительной техники. С уверенностью можно сказать, что ссылки на красивое сочетание «реальное время» стали часто использовать в специализированной печати. С не меньшей уверенностью можно сказать, что смысл этого термина трактуется специалистами по-разному.

Говорят, что информационная система работает в реальном времени, если ее быстродействие адекватно скорости протекания физических процессов на объектах контроля или управления.

Система управления реального времени должна собрать данные, произвести их обработку в соответствии с заданными алгоритмами и выдать управляющие воздействия за такой промежуток времени, который обеспечивает успешное решение поставленных перед системой задач.

Для системы управления прокатным станом время реакции системы должно быть в пределах нескольких миллисекунд, а для систем контроля за окружающей средой металлургического предприятия — несколько минут. Но, тем не менее, оба эти примера — из области задач реального времени.

Следовательно, практически все системы промышленной автоматизации являются системами реального времени. Принадлежность системы к классу систем реального времени никак не связана с ее быстродействием и во многом определяется динамическими свойствами самого технологического процесса и требованиями к качеству управления технологическими параметрами. Интуитивно понятно, что быстродействие системы реального времени должно быть тем больше, чем выше скорость протекания процесса на объекте контроля и управления.

Задачи реального времени предъявляют такие требования к аппаратному и программному обеспечению, как надежность, своевременная реакция на внешние события, высокая пропускная способность передающей среды и т.д. Для выполнения этих требований имеются специальные операционные системы реального времени.

Под операционной системой реального времени понимают систему, которая выполняет свои функции, а также отвечает на внешние асинхронные события за предсказуемый промежуток времени.

Главное требование, предъявляемое к операционным системам общего назначения, заключается в том, что они должны обеспечить оптимальное разделение всех ресурсов между всеми процессами. Соответственно, не должно быть высокоприоритетных задач, которые использовали бы какой-либо ресурс системы столько, сколько необходимо. В системах же реального времени требуется обеспечить требуемый уровень сервиса за вполне определенное, ограниченное время, т.е. ОС реального времени должна быть предсказуемой. Правильная, но запоздалая реакция системы на внешние события может быть гибельной.

Операционные системы реального времени, как правило, должны удовлетворять следующим требованиям:

- своевременность реакции системы на любые внешние события за априорно известный интервал времени;
- высокая пропускная способность программных интерфейсов, гарантирующая доставку необходимых объемов данных за детерминированное время;
- развитые механизмы синхронизации процессов, обеспечивающие разделение и совместное использование наиболее критических ресурсов;
- желательно, чтобы система была многозадачной и многопроцессорной, что дает возможность дефрагментировать задачу и запускать фрагменты задач на выполнение одновременно. Последнее обеспечивает максимальное использование всех ресурсов системы;
- диспетчеризация вытесняющих задач, основанная на приоритетах, которые гарантируют, что задача с самым высоким приоритетом будет сразу же вызвана на выполнение, когда она попадет в очередь активных процессов;

• непосредственное управление планированием со стороны пользовательских процессов для обеспечения выполнения соответствующей задачи.

Принято различать системы *«жесткого» и «мягкого» реального времени*.

Системой «жесткого» реального времени называется система, где неспособность обеспечить реакцию на какие-либо события в заданное время является отказом и ведет к невозможности решения поставленной задачи. Последствия таких отказов могут быть весьма печальные, если, например, вовремя не сработала система блокировки аварийного поднятия кислородной фурмы в сталеплавильном конвертере. Фактически в системах жесткого реального времени время реакции на любые внешние события не должно зависеть от числа выполняемых прикладных процессов, и оно заранее определено.

Системы «мягкого» реального времени проработаны теоретически пока далеко не полностью, поэтому дать точное определение для «мягкого» реального времени дать не представляется возможным. Иногда считают, что в системах «мягкого» реального времени реакция на внешние события находится в зависимости от числа загружаемых прикладных процессов. При этом максимально возможное время реакции заранее определено. Для простоты будем считать, что сюда относятся все системы реального времени, не попадающие в категорию «жестких».

Системы реального времени можно разделить в 4 класса.

1-й класс — программирование на уровне микропроцессоров. При этом программы для программируемых микропроцессоров, встраиваемых в различные устройства, очень небольшие и обычно написаны на языке низкого уровня, типа ассемблера.

2-й класс — минимальное ядро системы реального времени. На более высоком уровне находятся системы реального времени, обеспечивающие минимальную среду исполнения. Предусмотрены лишь основные функции, а управление памятью и диспетчер часто недоступны. Ядро представляет собой набор программ, выполняющих типичные, необходимые для встроенных систем низкого уровня функции, такие, как операции с плавающей запятой и минимальный сервис ввода-вывода. Прикладные программы разрабатываются в инструментальной среде, а выполняются, как правило, на встроенных системах.

3-й класс – ядро системы реального времени и инструментальная среда. Этот класс систем обладает многими чертами ОС с полным сервисом. Разработка ведется в инструментальной среде, а исполнение — на целевых системах. Этот тип систем обеспечивает гораздо более высокий уровень сервиса для разработчика прикладной программы.

4-й класс – ОС с полным сервисом. Такие ОС могут быть применимы для любых приложений реального времени. Разработка и исполнение прикладных программ ведутся в рамках одной и той же системы.

Системы 2-го и 3-го классов принято относить к системам «жесткого» реального времени, 4-го класса — «мягкого». Очевидно, это можно объяснить тем, что в первом случае к системе предъявляются более жесткие требования по времени реакции и необходимому объему памяти, чем во втором.

В настоящее время существует достаточно большое число ОС, удовлетворяющих требованиям реального времени. Это широко известные ОС VxWorks, QNX, OS-9, RTKernel и многие другие.

Операционная система ОЅ-9 (фирма Microware Systems Corp., USA). Это многозадачная, многопользовательская, модульная, вытесняемая, Unix-подобная операционная система реального времени.

Основные свойства операционной системы:

- возможность размещения в ПЗУ как операционной системы, так и приложений пользователя;
- переносимость на основные микропроцессорные платформы Motorola, Intel, PowerPC и др.;

- детерминированное, полностью вытесняемое ядро с минимальным гарантированным временем реакции на прерывание 3 мкс;
- развитые сетевые средства (TCP/IP, Profibus, Ethernet и др.).

Основа операционной системы – модули, перемещаемые объекты в память OS-9. Все функциональные компоненты, включая микроядро, реализованы в виде отдельных модулей. OS-9 компактна и не требует больших ресурсов контроллеров.

Модульная структура OS-9 обеспечивает модульную технику программирования, позволяя пользователям легко наращивать и изменять конфигурацию OS-9. Наращивание не требует перекомпиляции всей операционной системы, необходима компиляция только того модуля, который обеспечивает дополнительные функции. Модульность позволяет ОЅ-9 поддерживать общий набор прикладных программ для всего диапазона компьютеров, начиная от встроенных одноплатных компьютеров и заканчивая большими системами с разделением времени. Ядро OS-9 управляет памятью и процессорным временем, а также предлагает большой набор служебных системных функций. Многозадачное ядро OS-9 разрешает одновременное выполнение нескольких независимых программ с помощью механизма переключения задач и межпроцессорной связи. Все программы OS-9 выполняются как процессы, и каждый процесс с соответствующими правами доступа сможет осуществлять доступ к любым системным ресурсам, выполняя соответствующие системные вызовы. В дополнение к операционной системе OS-9 Microware представляет профессиональную среду программирования. Компиляторы языков высокого уровня, отладчики, средства разработки и связи обеспечивают исчерпывающую среду для разработки и поддержки прикладных программ.

Часть 2. Обобщенная схема автоматизированной системы управления технологическим процессом

Обобщенная схема автоматизированной системы управления технологическим процессом. Нижний, средний и верхний уровни. Назначение устройства связи с объектом УСО, индустриальные (промышленные) компьютеры, промышленные контроллеры – программируемые логические контроллеры, открытые магистрально-модульные системы. Общая характеристика прикладного программного обеспечения современных промышленные контроллеры и пакетов прикладных программ операторских станций технологических процессов.

Обобщенная схема автоматизированной системы управления технологическим процессом

Автоматизированный технологический комплекс в металлургии как объект контроля и управления характеризуется следующими признаками:

- большим объемом контролируемых переменных и расчетных признаков;
- низкой прозрачностью процесса, из-за отсутствия возможности непосредственного контроля развития отдельных стадий процесса;
- существенным запаздыванием в получении информации о выходных показателях процесса, в связи с чем результаты непосредственного контроля часто не позволяют определять первопричину расстройства хода технологического процесса;
- глубокой взаимосвязью всех процессов плавки при ограниченности ресурсов на управление;
- наличием случайных измерительных помех различной природы и характера, что снижает качество информации.

Обобщенная схема информационной системы представлена на рис. 4.1.

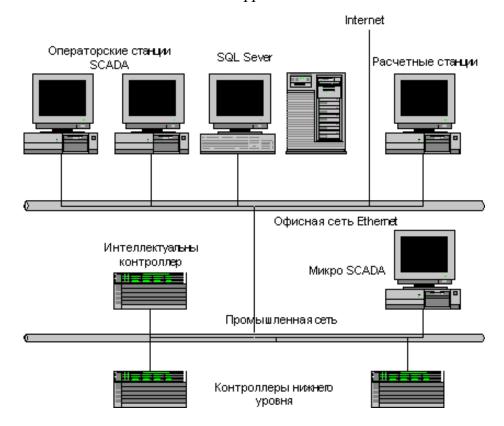


Рис. 4.1. Обобщенная схема системы контроля и управления

Это трехуровневая, иерархическая система управления. Для каждого из уровней предполагает индивидуальный подход к разработке программного обеспечения, зависящий от решаемых задач. Заметим, что при выборе базовых программно-технических компонентов необходимо обеспечить, с одной стороны, простоту реализации отдельных составных частей и комплекса системы в целом, а с другой — технологичность сопровождения, модернизации и развития комплекса в процессе эксплуатации.

<u>Нижний уровень</u>

Основным функциональным назначением этого уровня является ввод электрических сигналов от датчиков, их первичная обработка, программная реализация законов автоматического регулирования, вывод сигналов управления исполнительными механизмами и информационное взаимодействие со средним (промежуточным) уровнем.

На самом нижнем уровне на вход управляющего вычислительного комплекса от датчиков (термопар, индуктивных датчиков, счетчиков продукции и др.) поступает измерительная информация о текущих значениях параметров, характеризующих ход технологического процесса (состояние и параметры заготовок, качество обработанных деталей, их количество и др.). Компьютер обрабатывает эту информацию в соответствии с принятым законом управления (алгоритмом управления), определяет управляющие воздействия, которые необходимо приложить к исполнительным механизмам для изменения управляемых параметров, с тем, чтобы управляемый процесс протекал оптимальным образом.

Неотъемлемой частью автоматизированной системы управления технологическими процессами являются устройства связи с объектом (УСО), назначение которых заключается в сопряжении датчиковой аппаратуры и исполнительных механизмов контролируемого объекта и/или технологического процесс с вычислительными средствами системы. УСО представляет собой комплекс в виде специализированных функциональных блоков, осуществляющий необходимый информационный обмен между технологическим объектом и управляющей информационной системой. Заметим, что использование модулей УСО может потребоваться не только на нижнем уровне системы управления, но и на

верхних уровнях — например, для связи с диспетчерским оборудованием или для скоростного ввода информации от специальных подсистем.

На рис. 4.2 изображена общая структурная схема системы связи компьютера с объектом управления

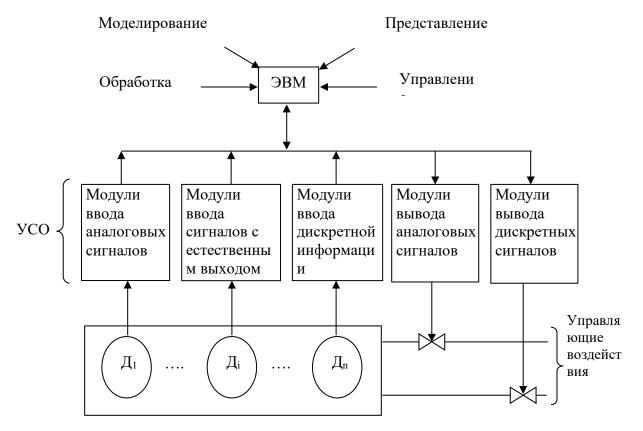


Рис. 4.2. Структурная схема связи компьютера с объектом управления

Как правило, на УСО возлагаются следующие функции.

- Нормализация аналогового сигнала приведение границ шкалы первичного непрерывного сигнала к одному из стандартных диапазонов входного сигнала аналого-цифрового преобразователя измерительного канала. Наиболее распространены следующие диапазоны: 0–20 мA, 4–20 мA, от 0 до 5 В; от –5 до 5В; от 0 до 10 В, а также сигналы датчиков с естественными выходными сигналами (термопары, термометры сопротивления, тензометрические датчики и т.п.).
- Предварительная низкочастотная фильтрация аналогового сигнала ограничение частот первичного сигнала с целью снижения влияния на результаты измеренных помех различного происхождения. На промышленных объектах наиболее распространены помехи с частотой сети переменного тока, а также хаотические импульсные помехи, вызванные влиянием на технические средства измерительного канала переходных процессов и наводок при коммутации исполнительных механизмов повышенной мощности.
- Обеспечение гальванической развязки между источником аналогового или дискретного сигнала и измерительными каналами системы. В равной степени это относится к изоляции между каналами дискретного вывода системы и управляемым силовым оборудованием. Помимо собственно защиты выходных и входных цепей, гальваническая изоляция позволяет снизить влияние на

систему помех по цепям заземления за счет полного разделения вычислительной системы и контролируемого оборудования.

Помимо перечисленных функций ряд устройств связи с объектом может выполнять более сложные задачи за счет наличия в их составе подсистемы аналого-цифрового преобразования и дискретного ввода-вывода, микропроцессора и средств организации одного из интерфейсов последовательной передачи данных.

Основными параметрами УСО в общем случае являются скорость и погрешность преобразования, а также диапазон входных/выходных напряжений и токов. Наиболее оптимальным вариантом структурной организации модуля УСО является разделение его на две функционально-конструктивные части. Одна часть — системная, реализующая функции сопряжения с РLС (РС) и в ряде случаев функции обработки информации, и вторая — собственно модуль УСО, осуществляющий непосредственный прием сигналов от датчиков технологических процессов и выдачу управляющих воздействий на исполнительные механизмы.

На плате модуля УСО могут быть также установлены специализированные процессоры, выполняющие цифровую фильтрацию и предварительную обработку информации с датчиков, эталонный источник напряжения для оценки и коррекции метрологических характеристик АЦП и др. Такие интеллектуальные УСО позволяют существенно сократить затраты времени центрального процессора и повысить производительность системы. С развитием микроэлектроники цифровая обработка сигналов и другие интеллектуальные функции реализуются в УСО уже на уровне первичных преобразователей.

Для промышленных контроллеров характерна магистрально-модульная структура. Модуль процессора и модули УСО имеют единый функциональный, электрический и конструктивный интерфейс.

Для организации управления в настоящее время чаще всего используется синхронный принцип связи компьютера с объектом управления. При этом процесс управления разбивается на циклы равной продолжительности тактирующими сигналами таймера (электронных часов). За время каждого цикла осуществляется:

- последовательный опрос сигналов датчиков, преобразование их в цифровой код и ввод в PLC (PC);
- обработка в PLC (PC) поступившей информации и расчет необходимых управляющих воздействий;
- выдача управляющих воздействий на объект управления и запоминание их на время всего цикла.

В оставшееся время до конца цикла PLC (PC) выполняет так называемые «фоновые» задачи, которые могут прерываться тактирующими сигналами без нарушения программы.

Для некоторых важных параметров объекта применяется асинхронный принцип связи с PLC (PC). При этом вместо тактирующих сигналов с таймера PLC (PC) поступают через устройство прерывания сигналы датчиков прерывания (ДП), связанных с объектом. PLC (PC) реагирует на сигналы прерывания, с учетом уровня их приоритета, и переходит на работу по программе для аварийного режима.

Заметим, что для реализации этих задач в промышленных системах существуют, в принципе, два конкурирующих направления в технике: индустриальные (промышленные) компьютеры РС и программируемые логические контроллеры (промышленные контроллеры) PLC.

Индустриальные (промышленные) компьютеры (PC), как правило, программно совместимы с обычными офисными PC, но адаптированы для жестких условий промышленной эксплуатации. Они должны работать круглосуточно, подвергаясь воздействию окружающей среды, влаги, пыли, вибрации. В качестве устройств сопряжения

с объектом управления данные системы комплектуются дополнительными платами (адаптерами) расширения, которые на рынке представлены в большом разнообразии.

Промышленные контроллеры – программируемые логические контроллеры (PLC – Programmable Logic Controller) представляют собой специализированные вычислительные устройства, предназначенные для управления процессом (объектом) в реальном времени.

На рынке представлены многие десятки и даже сотни типов контроллеров, способных обрабатывать от нескольких переменных до нескольких сот переменных.

Основные базовые функции промышленного контроллера

Основные, базовые функции промышленного контроллера приведены на **рис. 4.3.** Назначение этих функций очевидно.

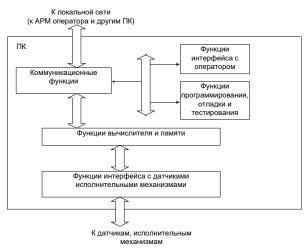


Рис. 4.3. Базовые функции промышленного контроллера

PLC являются ответственными компонентами и, в значительной степени, определяют свойства всей информационной системы: открытость, модульность, наращиваемость, гибкость, надежность функционирования. PLC представляют собой специализированные вычислительные устройства, предназначенные для управления процессом (объектом) в реальном времени.

В роли управляющих машин традиционные универсальные РС не всегда пригодны по следующим причинам:

- недостаточный спектр периферийного оборудования и устройств;
- время решения задач в системах управления должно быть строго ограничено реальным временем, что в настоящее время сложно реализуемо для многих задач на PC;
- работа РС в необслуживаемом круглосуточном режиме невозможна;
- как правило, РС не обеспечивают высокую технологичность ремонта и обслуживания, что необходимо в АСУ ТП;
- гибкость и необходимость модернизации в процессе эксплуатации, наращивание каналов ввода/вывода без потерь быстродействия на PC проблематичны;
- принцип распределенности систем на РС реализовать достаточно сложно;
- в универсальных PC большое количество ресурсов тратится на пользовательский интерфейс, который в системах АСУ ТП не всегда необходим;
- в управляющих электронно-вычислительных машинах необходимо специализированное программное обеспечение: ОС и прикладные программы,

т.к. целью информационной среды является обеспечение заданной реакции на воздействия, то используются ОС реального времени;

• сменяемость универсальных РС быстрая, а управляющие электронновычислительные машины мало меняются, преемственность достигается тем, что в состав магистральных шин закладывается заведомая избыточность.

Промышленные контроллеры имеют вычислительное ядро и модули ввода-вывода, принимающие информацию (сигналы) с датчиков, переключателей, преобразователей, других устройств и контроллеров и осуществляющие управление процессом или объектом, выдачей управляющих сигналов на приводы, клапаны, переключатели и другие исполнительные устройства. Современные PLC часто объединяются в сеть, а программные средства, разрабатываемые для них, позволяют в удобной для оператора форме программировать и управлять ими через Host-компьютер, находящийся на более высоком уровне. В PLC, в отличие от PC, как правило, используются операционные системы реального времени – ОС (см. далее) Эксплуатируются РLС (РС) в основном в производственных условиях. В связи с этим конструктивной особенностью PLC (PC) является пылезащитный и сейсмостойкий корпус, защищающий их при работе в неблагоприятных условиях внешней среды от механических ударов и вибраций, повышенной влажности, запыленности и т.д. Используется модульный конструктивный принцип, обеспечивающий удобство обслуживания и ремонта. Эти комплексы имеют большее, чем у офисных компьютеров, количество слотов расширения, что позволяет обслуживать большое количество датчиков и исполнительных. В порядке объективности все же следует отметить, что в последнее время наметилась тенденция стирания граней между PLC и PC, разрабатываются PC, которые имеют специальное конструктивное исполнение и другие свойства, приближающих их к PLC.

Основная тенденция в создании современных автоматизированных информационных систем заключается в следующем: там, где требуется повышенная надежность и управление в реальном масштабе времени, применяют PLC, которые по сравнению с PC общего назначения более устойчивы к зависаниям, обладают детерминированностью поведения, детерминированными реакциями на события реального времени и практически не подвержены атакам программ-вирусов.

К аппаратно-программным средствам этого уровня обычно предъявляются жесткие требования по надежности, времени реакции на исполнительные устройства, датчики и т.д. Работоспособность указанных устройств в промышленности должна обеспечиваться в относительно жестких условиях эксплуатации (например, повышенные температуры, сильная загрязненность помещений, токопроводящая пыль, наличие агрессивных газов) при непрерывном круглосуточном режиме работы.

Для обеспечения надежности работы аппаратуры этого уровня обычно применяются специальные технологические решения. В частности, в ней должны отсутствовать механические элементы — вращающиеся диски, вентиляторы и т.п., ухудшающие надежность работы аппаратуры. Для охлаждения элементов используются пассивные системы охлаждения — радиаторы естественной конвекции, а ПЗУ реализовано на флэш (Flash)-дисках.

Для построения современных АСУ ТП применяются *открытые магистрально-модульные системы (ОММС)*

Признаки открытой магистрально-модульные системы:

- требования по реализации работы в реальном масштабе времени;
- открытая модульная архитектура;
- стандартизация всех видов программного обеспечения;
- совместимость с протоколами и интерфейсами вычислительных сетей общего назначения и промышленных сетей;
- развитый стандартный инструментарий;
- распределенность и мультипроцессорность.

Отверытая система — это система, построенная по магистрально-модульному принципу на основе совместимых унифицированных компонентов, не имеющих логических ограничений по наращиванию, сопровождаемая исчерпывающей информацией по всем видам обеспечения и отвечающая требованиям работы в реальном масштабе времени.

Магистрально-модульный принцип построения включает в себя:

- замену аппаратных компонентов без выключения системы («hot swapping», горячая замена);
- распределенность построения, большое количество и разнообразие подключаемых периферийных устройств (ПУ), следовательно большое количество каналов прерываний;
- необходимость стандартизации аппаратного и программного обеспечения. Бывают различные группы стандартов: корпоративные (Siemens, Bosh), национальные (ANSI, ГОСТ), международные (ISO, IEC). Чем выше ранг стандарта, тем более открытой является система. К примеру, в универсальных ПК открытость частичная, в результате ограниченная по развитию система. Стандарты ОММС распространяются как на конструктив, так и на программное обеспечение (POSIX).

Обобщенная модель магистрально-модульной системы представлена на рис. 4.4.

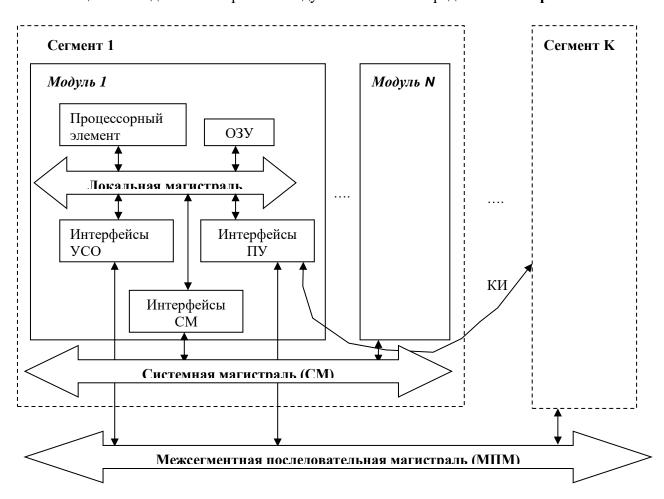


Рис. 4.4. Обобщенная схема ММС: ПУ – периферийное устройство; КИ – коммуникационный

В рамках магистрально-модульной системы (ММС) может быть подключено множество модулей, в каждом из них может отсутствовать ряд элементов (например, ОЗУ).

Модуль объединяют в сегменты. В рамках одного сегмента обмен данными между модулями осуществляются через системную магистраль (СМ).

Множество сегментов объединяются в вычислительную систему с помощью межсегментной последовательной магистрали (МПМ). С помощью коммуникационного интерфейса (КИ) можно напрямую объединять различные модули из разных сегментов. Прикладное программное обеспечение современных РLС, имеющих встроенную операционную систему, может быть разработано как с использованием традиционных инструментальных средств (компиляторы языков С, С++, Паскаль и т.д.), так и на основе специализированных языковых средств. Традиционная технология требует от разработчика знаний не только в области использования языков программирования, но и особенностей операционной системы, а также аппаратных возможностей данного контроллера и организации системы ввода-вывода. При этом разработанное программное обеспечение будет привязано только к данному типу контроллера и не может быть перенесено на другую аппаратно-программную платформу.

Потребность в специальных платформо-независимых языках программирования возникла давно. Она послужила причиной создания стандарта на такие языки программирования PLC. В результате появился стандарт МЭК 61131-3, специфицирующий синтаксис и мнемонику языков программирования, которые могут применяться для любого логического контроллера.

Стандартом определяется пять языков: три графических (SFC, FBD, LD) и два текстовых (ST, IL).

SFC (Sequential Function Charts) — графический язык последовательных функциональных схем. Язык SFC предназначен для использования на этапе проектирования программного обеспечения и позволяет описать "скелет" программы — логику ее работы на уровне последовательных шагов и условий переходов (транзакций). Это, по своей сути, аналог блок-схем программ.

FBD (Function Block Diagram) — графический язык функциональных блоковых диаграмм. Язык FBD применяется для построения комплексных процедур, состоящих из различных функциональных библиотечных блоков — арифметических, регуляторов, тригонометрических и т.д.). Аналог — принципиальные схемы электронной аппаратуры.

LD (*Ladder Diagrams*) – графический язык релейной логики. Язык LD применяется для описания логических выражений различного уровня сложности. Позволяет моделировать различные схемы релейной логики.

ST (Structured Text) – язык структурированного текста. Это язык высокого уровня, по мнемонике похож на Паскаль и применяется для разработки процедур со сложной логикой, обработки данных.

IL (*Instruction List*) – *язык инструкций*. Это язык низкого уровня класса ассемблера и применяется для программирования эффективных, оптимизированных процедур.

Программное обеспечение систем контроля и управления технологическими процессами по применению разделяется на встраиваемое для системы исполнения и настольное – для системы разработки (**рис. 4.5**).

Система исполнения загружается в ПЗУ РLС при изготовлении контроллера или загружается пользователем. Чаще всего в качестве встраиваемого ПО используют операционные системы реального времени. Начинаются они обычно с маленького ядра и загрузчика программ, к которому как строительные блоки присоединяются системные службы и утилиты. Требования, предъявляемые к *среде исполнения* систем реального времени, следующие:

- небольшой объем памяти необходим для возможности ее встраивания;
- система должна быть полностью резидентна в памяти для того, чтобы избежать замещения страниц памяти или подкачки;
- система должна быть многозадачной для обеспечения максимально эффективного использования всех ресурсов системы;

- микроядро ОС должно быть с приоритетом на обслуживание прерываний. Приоритет на прерывание означает, что готовый к запуску процесс, обладающий некоторым приоритетом, обязательно имеет преимущество в очереди по отношению к процессу с более низким приоритетом, быстро заменяет последний и поступает на выполнение. Ядро заканчивает любую сервисную работу, как только поступает задача с высшим приоритетом, что гарантирует предсказуемость системы;
- диспетчер с приоритетом дает возможность разработчику прикладных программ присвоить каждому загрузочному модулю приоритет. Присвоение приоритета используется для определения очередности запуска программ, готовых к исполнению. Диспетчеризация, в основу которой положен принцип присвоения приоритета, и наличие ядра с приоритетом на прерывание позволяют разработчику прикладной программы полностью контролировать систему. Если поступает событие с высшим приоритетом, то система прекращает обработку задачи с низшим приоритетом и отвечает на вновь поступивший запрос.

Основой любой среды исполнения в реальном времени является ядро или диспетчер. Ядро управляет аппаратными средствами целевого компьютера: центральным процессором, памятью и устройствами ввода/вывода, контролирует работу всех других систем и программных средств прикладного характера. В системе реального времени диспетчер занимает место между ядром операционной системы и прикладными программами, обеспечивает специальный сервис, необходимый для работы приложений реального времени.

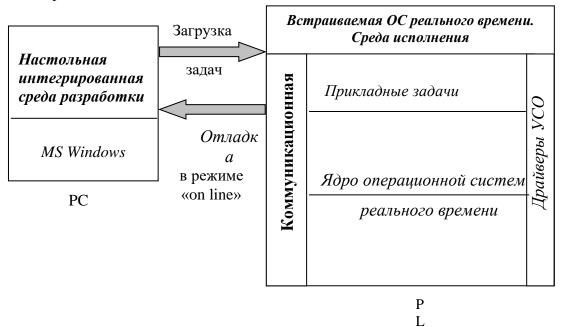


Рис. 4.5. Взаимодействие сред разработки и исполнения

В ОС могут использоваться универсальные языки программирования. Они позволяют решать задачи программирования PLC, но их применение требует квалификации в теории и технологии программирования, знания тонкостей аппаратного обеспечения, поэтому использование традиционных языков программирования — не всегда лучшее решение. В связи с этим фирмы, производящие оборудование для построения информационных систем и систем автоматизации, стремятся сопровождать свою продукцию набором программных инструментов, упрощающих и ускоряющих процесс программирования, с помощью которых пользователь по определенным правилам и соглашениям мог бы описать логику работы PLC. Эти программные инструменты

специально ориентированы на разработку программ управления внешними устройствами и контроллерами .

В качестве технических средств среды разработки чаще всего используются PC, а в качестве операционной системы – ОС общего назначения, в частности MS Windows.

К этому классу инструментального ПО относятся пакеты типа ISaGRAF (CJ International France), InConrol (Wonderware, USA), Paradym 31 (Intellution, USA).

Примером реализации этого стандарта является инструментальная система ISaGRAF разработки фирмы CJ International (Франция), позволяющая разрабатывать программное обеспечение для интеллектуальных PLC. ISaGRAF предоставляет возможность разрабатывать процедуры с использованием языка C/C++, причем эти процедуры можно вызывать из любого, описанного выше языка.

Система ISaGRAF состоит из двух частей: системы разработки ISaGRAF Workbench и системы исполнения ISaGRAF Target (рис. 4.6).

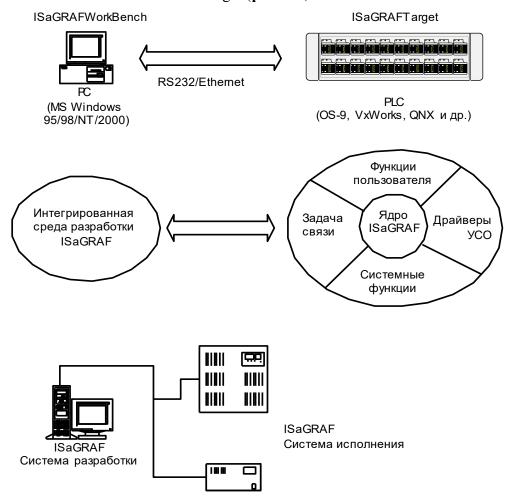


Рис. 4.6. Структура системы ISaGRAF

Система разработки (ISaGRAF Workbench) представляет собой набор Windows-приложений, интегрированных в единую инструментальную среду и работающих под управлением ОС Windows 2000.

Основу системы исполнения (ISaGRAF Target) составляет набор программных модулей, выполняющих самостоятельные задачи под управлением ядра ISaGRAF. Ядро ISaGRAF реализует поддержку стандартных языков программирования, типового набора функций и функциональных блоков и драйверов ввода-вывода. Система исполнения ISaGRAF Target загружается в ПЗУ при изготовлении контроллера или загружается пользователем. Задача связи обеспечивает поддержку процедуры загрузки пользовательского ISaGRAF-приложения со стороны программируемого контроллера, а

также доступ к рабочим переменным этого приложения со стороны отладчика системы разработки ядра ISaGRAF.

Общий алгоритм работы с пакетом ISaGRAF можно представить следующим образом:

- создание проекта в рамках ISaGRAF Workbench;
- написание программ на языках ISaGRAF в соответствующих редакторах;
- описание переменных в редакторе переменных;
- описание соединений переменных с физическими и логическими устройствами вводавывода;
- компилирование проекта в исполняемый код для ISaGRAF Target;
- отладка программы в ISaGRAF Workbench;
- отладка программы в ISaGRAF Target.

Средний уровень

Основная задача этого уровня заключается в сборе информации с различных подсистем и/или контроллеров, их обработка и передача на верхний уровень. Средний уровень чаще всего представляет собой станции контроля и регистрации технологических параметров и предназначен для замены регистрирующих приборов. Современный подход к этому уровню — это SCADA-система. Кроме того, средний уровень является связующим звеном нижнего и верхнего уровней системы.

Аппаратная база этого уровня должна быть более мощной, предусматривающей возможность обмена через промышленные (Fieldbus) и офисные сети (например, Ethernet) с нижним и верхним уровнем. Предусмотрена организация логических и горизонтальных соединений с базами данных реального времени. Обычно используют промышленные компьютеры РС совместимой архитектуры, а чаще всего — интеллектуальные контроллеры.

Спектр задач интеллектуальных контроллеров и промышленных компьютеров более разнообразен и включает:

- сбор данных с контроллерного уровня;
- обработку данных, включая масштабирование;
- синхронизацию работы подсистем;
- организацию архивов по выбранным параметрам;
- обмен информации между нижним и верхним уровнями;
- резервирование каналов передачи данных.

База данных реального времени. До появления IndustrialSQL Server (Wonderware, США) задача регистрации информации в реальном времени могла быть решена либо на уровне программного обеспечения интеллектуального контроллера, либо на уровне SCADA-системы. С возникновением IndustrialSQL Server появилась дополнительная возможность обеспечить высокоскоростное хранение информации в базах данных, архивировать данные с целью экономии места на диске, обеспечивать стандартный доступ к данным. IndustrialSQL Server представляет собой расширение Microsoft SQL Server и обеспечивает сбор данных в сотни раз быстрее, чем любые другие реляционные базы данных. Он может использоваться как в небольших цехах с сотней регистрируемых технологических параметров, так и на крупных промышленных предприятиях с сотнями тысяч параметров.

Системы Micro-SCADA. Рассматривая обобщенную структуру систем управления, следует ввести и еще одно понятие - Micro-SCADA. Micro-SCADA - это системы, реализующие стандартные (базовые) функции, присущие SCADA - системам верхнего уровня, но ориентированные на решение задач автоматизации в определенной отрасли (узкоспециализированные). В противоположность им SCADA - системы верхнего уровня

являются универсальными. Выбор ориентации определяет спектр драйверов или серверов ввода-вывода для подключаемого специфичного контроллерного оборудования, набор графических объектов.

Верхний уровень

Основным средством коммуникации на этом уровне обычно является компьютерная сеть Ethernet

Компьютеры этого уровня решают следующие задачи:

- диспетчерский сбор, сигнализация управления и отображение всего технологического процесса или его отдельных подсистем (SCADA-системы), реализованные на операторских станциях;
- хранение технологической информации в базе данных;
- компьютерная поддержка принятия решений (комплекс математических моделей технологического процесса, интеллектуальная поддержка принятия решений, системы распознавания образов), реализованная на рабочих станциях;
- интеграция локальной сети в корпоративную сеть всего предприятия (коммуникационный сервер).

Выбор базовых компонентов этого уровня, предназначенных для организации комфортного человеко-машинного интерфейса, во многом определяется сложившейся привычной ориентацией на персональные компьютеры и имеющийся подготовленный персонал для работы на них. В качестве программного обеспечения этого уровня можно использовать достаточно широкий спектр традиционных SCADA-систем (Genesis, InTouch, WinCC и др.), СУБД (SQL Server, SQL Base, Oracle Server, Informix, SyBase и др.), коммуникационных программ Internet Information Server (IIS) и др.

Пакеты прикладных программ операторских станций технологических процессов (SCADA)

Главными функциями таких SCADA-программ являются отображение технологического процесса в виде мнемосхем, сигнализация об аварийных ситуациях, обеспечение общего управления процессом со стороны оператора-технолога, хранение истории процессов, выполнение функций безопасности, общесистемных функций и т.п. Несмотря на множество указанных функций, выполняемых SCADA, основными ее отличительными признаками является наличие интерфейса с пользователем.

Диспетчерское управление и сбор данных (SCADA) является основным и в настоящее время остается одним из наиболее перспективных методов автоматизированного управления сложными динамическими системами. Именно на принципах диспетчерского управления строятся крупные автоматизированные системы в металлургии.

Конечно, многие отделы АСУ крупных металлургических предприятий, как правило, имеют свои наработки в этой области. В то же время есть фирмы, специализирующиеся на разработке универсальных SCADA-программ, таких как, InTouch, SIMATIC WinCC, и многие другие.

SCADA — процесс сбора информации реального времени с удаленных точек (объектов) для обработки, анализа и возможного управления удаленными объектами. Требование обработки реального времени обусловлено необходимостью доставки (выдачи) всех необходимых сообщений и данных на центральный интерфейс диспетчера.

Основные требования к SCADA-системам:

- создание богатых возможностей для реализации графического интерфейса;
- обеспечение открытости как с точки зрения подключения различного контроллерного оборудования, так и коммуникации с другими программами;
- обеспечение простоты разработки приложений;

• использование новых технологий.

Все современные SCADA-системы включают три основных структурных компонента (**puc. 4.7**).

Удаленный терминал (Remote Terminal Unit, RTU) — осуществляющий обработку задачи (управление) в режиме реального времени. Спектр его воплощений широк — от простейших датчиков, осуществляющих съем информации с объекта, до специализированных многопроцессорных отказоустойчивых вычислительных комплексов, осуществляющих обработку информации и управление в режиме жесткого реального времени. Использование устройств низкоуровневой обработки информации позволяет снизить требования к пропускной способности каналов связи с центральным диспетчерским пунктом. Заметим, что число контролируемых удаленных данных в современной SCADA-системе может достигать десятки тысяч.

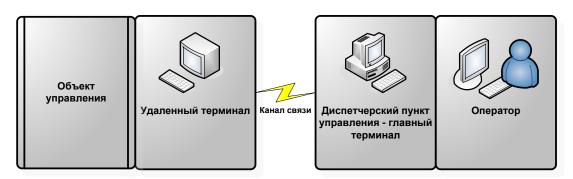


Рис. 4.7. Основные структурные компоненты SCADA-систем

Диспетичерский пункт управления – главный терминал (Master Terminal Unit, MTU) – осуществляет обработку данных и управление высокого уровня, как правило, в режиме мягкого реального времени. Одна из основных функций – обеспечение интерфейса между человеком-оператором и системой. В зависимости от конкретной системы диспетичерский пункт управления может быть реализован в самом разнообразном виде – от единичного компьютера с дополнительными устройствами подключения к каналам связи до больших вычислительных систем или объединенных в локальную сеть рабочих станций и серверов.

Коммуникационная система — каналы связи (Communication System, CS) необходима для передачи данных с удаленных точек (объектов, терминалов) на центральный интерфейс оператора-диспетчера и передачи сигналов управления на удаленный терминал (объект) в зависимости от конкретного исполнения.

Можно выделить четыре основных функциональных компонента систем диспетчерского управления и сбора данных (**рис. 4.8**):

- человек-оператор;
- компьютер взаимодействия с человеком;
- компьютер взаимодействия с задачей (объектом);
- задача (объект управления).

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

- автоматизированная разработка, дающая возможность создания ПО системы автоматизации без реального программирования;
- средства исполнения прикладных программ;
- сбор первичной информации от устройств нижнего уровня;
- обработка первичной информации;
- регистрация алармов и исторических данных;
- хранение информации с возможностью ее пост-обработки (как правило, реализуется через интерфейсы к наиболее популярным базам данных);

- визуализация информации в виде мнемосхем, графиков и т.п.;
- возможность работы прикладной системы с наборами параметров, рассматриваемых как "единое целое" ("recipe" или "установки");
- язык для реализации алгоритмов управления, математических и логических вычислений;
- средства документирования как самого алгоритма, так и технологического процесса;
- драйверы к обслуживанию нижнего уровня АСУ ТП;
- сетевые функции;
- средства защиты от несанкционированного доступа в систему;
- многооконный графический интерфейс и другие функции.

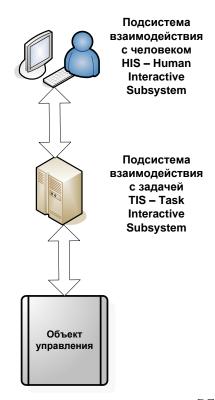


Рис. 4.8. Функциональные компоненты SCADA-систем

Программные продукты класса SCADA широко представлены на мировом рынке. Это несколько десятков SCADA - систем, многие из которых нашли свое применение и в России. Наиболее популярные из них приведены ниже:

- InTouch (Wonderware) CIIIA;
- Citect (CI Technology) Австралия;
- FIX (Intellution) CШA;
- Genesis (Iconics Co) CIIIA;
- Factory Link (United States Data Co) CIIIA;
- RealFlex (BJ Software Systems) CIIIA;
- Sitex (Jade Software) Великобритания;
- TraceMode (AdAstrA) Россия;
- Cimplicity (GE Fanuc) CIIIA;
- САРГОН (НВТ Автоматика) Россия.

Часть 3. Интеграция в информационных системах технологических процессов

Общая характеристика промышленных сетей. Механизмы обмена данными *OPC-серверы*. Использование Internet-технологии в задачах управления.

Общая характеристика промышленных сетей

Каналы связи контроллеров с интеллектуальными контроллерами и промышленными компьютерами отличаются большим разнообразием. Выбор конкретного решения зависит от архитектуры системы, расстояния, числа контролируемых точек, требований к пропускной способности и надежности канала, наличия допустимых линий связи. Задача промышленных сетей (Fieldbus) состоит в организации физической и логической связи различных компонентов автоматизации таким образом, чтобы информация с этого уровня была доступна общезаводской информационной системе.

Таким образом, **промышленные сети Fieldbus** — это специализированные быстродействующие каналы связи, позволяющие эффективно решать задачи надежных и помехоустойчивых соединений с компонентами автоматизации различных разновидностей, в частности программируемых логических контроллеров, промышленных персональных компьютеров, удаленных модулей УСО и др.

Сформулируем лишь некоторые основные требования, которые можно предъявить к "идеальной" промышленной сети:

- 1) предсказуемость времени доставки информации;
- 2) помехоустойчивость;
- 3) доступность и простота организации физического канала передачи данных;
- 4) максимальный сервис для приложений верхнего уровня;
- 5) минимальная стоимость устройств аппаратной реализации, особенно на уровне контроллеров;
- 6) возможность получения «распределенного интеллекта», путем представления максимального доступа к каналу нескольким ведущим узлам.
- 7) управляемость и самовосстановление в случае возникновения нештатных ситуаций.

Следовательно, промышленные сети должны полностью удовлетворять запросам потребителей по модульности, надежности, защите от внешних помех, простоте в построении, монтаже и программировании.

Нетрудно заметить, что требования противоречивы, так второе требование противоречит третьему. В целом следует отметить, что промышленная сеть, по своей сути, – один большой компромисс. От того, как расставлены акценты в этом компромиссе, зависит успешность решения задач, стоящих перед данной сетевой архитектурой.

Сегодня говорить о некой универсальной промышленной сети еще рано. Однако известны требования к ней и понятны прикладные задачи, которые необходимо решать с ее помощью:

• автоматизация на уровне периферийного оборудования.

Здесь необходимы следующие качества: высокая скорость передачи, короткое время реакции на события, длина линий связи обычно до 300 метров;

• автоматизация на уровне управления технологическими процессами (уровень производства).

Этот уровень требует следующих качеств: среднее время опроса датчиков до 100 мс, длина линий связи до 1500 м, с возможностью работы устройств во взрывоопасных средах.

На каком бы уровне сеть не работала, она должна быть *стандартной*. Соблюдение всех семи уровней эталонной модели OSI/ISO в системах управления обычно не требуется. Кроме того, наличие дополнительных изолирующих межуровневых интерфейсов привело бы к потере производительности системы в режиме реального времени и сделало бы существенно менее предсказуемыми задержки прохождения сообщений в сети. Большинство промышленных сетей имеют сжатую трехуровневую архитектуру и поддерживают 1-, 2- и 7-й уровни OSI модели.

Основными достоинствами промышленных сетей являются недорогие линии и надежность передачи данных. Данные передаются последовательно бит за битом, как правило, по одному физическому каналу (одному проводнику). Такой режим передачи не только экономит кабельное оборудование, но и позволяет решать задачи по надежной передаче данных на большие расстояния.

Наиболее простым и наиболее распространенным в мире и России сетевым протоколом промышленных сетей является Modbus, популярность которого объясняется простотой как для разработчиков, так и потребителей, а также относительно низкой стоимость реализации, широко применяются сети Profibus, CAN, промышленный Ethernet и др.

В зависимости от области применения весь спектр промышленных сетей можно разделить на два уровня:

- sensor/actuator level (датчиковый уровень) задачи сетей этого уровня сводятся к опросу датчиков и управлению работой разнообразных исполнительных механизмов. Здесь основным параметром является время и частота опроса каналов датчиков. Типичные представители этого уровня Profibus-DP/PA, Interbus-S;
- field level (системный уровень) промышленные сети этого уровня решают задачи по управлению процессом производства, сбором и обработкой данных на уровне промышленных контроллеров. На этом уровне на первый план выдвигается задача передачи больших объемов данных, при этом схема потоков данных может быть достаточно сложной.

Другими словами, необходимо различать промышленные сети для системного уровня (Field busses) и датчикового уровня (Sensor/actuator busses).

Сравнение этих двух классов в самом общем виде можно получить по критериям, представленным в **табл. 4.1.**.

На **рис. 4.9** представлена обобщенная сетевая структура, показывающая в общем виде возможное использование того или иного вида сетей на отдельных уровнях условного промышленного предприятия.

Из всего многообразия промышленных сетей рассмотрим кратко *Profibus (Process Field Bus)*. Под общим названием Profibus понимается совокупность трех различных вариантов этой сети: Profibus-FMS, Profibus-DP, Profibus-PA, каждый из которых ориентирован на свою прикладную область. Архитектура составляющих частей протокола Profibus приведена на **рис. 4.10**.

Таблица 4.1. Сравнительные критерии промышленных сетей типов Fieldbus и Sensorbus

| Основные критерии | Fieldbus | Sensorbus |
|-----------------------------------|-------------------------|----------------|
| Расширение сети | От 100 м до 1500 м | До 100 м |
| Время цикла | От 10 мс до 10 с | От 1 мс до 1 с |
| Объем передаваемых данных за цикл | От 8 байт до сотен байт | От 1 до 8 байт |
| Цена среды передачи | Низкая | Очень низкая |

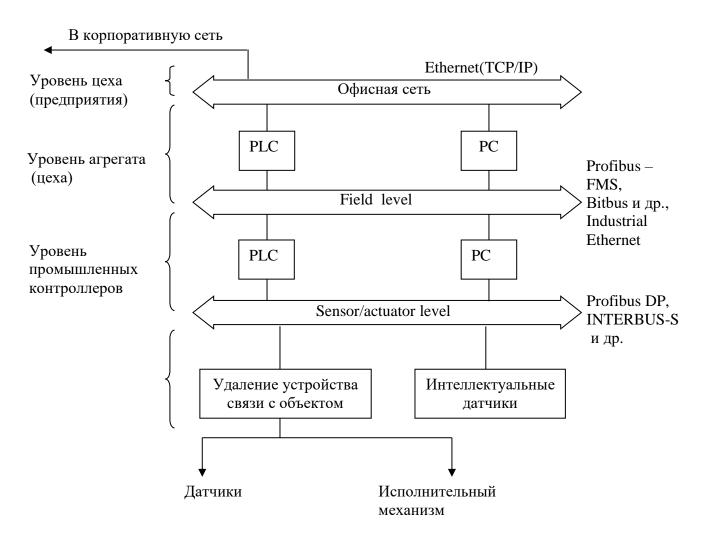


Рис. 4.9. Уровни связи и типовые компьютерные сети в информационных системах технологических процессов

| | PROFIBUS | | |
|--------------|---|-------------|-------------|
| Profibus-FMS | | Profibus-DP | PROFIBUS-PA |
| | Функции пользовательского уровня | | |
| 7 | Fieldbus Message Specification (FMS) | Нет | Нет |
| 36 | Отсутствуют | | |
| 2 | Fieldbus Data Link (FDL) –промышленный канал данных | | |
| 1 | RS-485/ | Fibre Optic | IEC1158-2 |

Рис. 4.10. Архитектура протокола PROFIBUS

Profibus FMS (FMS -- Fieldbus Message Specification, Спецификация сообщений шины полевого уровня). Это универсальное решение для задач взаимодействия на верхнем уровне иерархии промышленных связей и предназначено для взаимодействия между контроллерами. На этом уровне требуется высокая степень функциональности, и этот критерий более важен, чем критерий скорости. Основное назначение — передача больших объемов данных. Следовательно, если требуется объединить в детерминированную сеть несколько контроллеров, оптимальным вариантом будет PROFIBUS-FMS. Протокол FMS по стандарту определяет уровни 1, 2 и 7.

Profibus-DP (**DP** – **Decentral Periphery**, **Децентрализованные периферийные устройства**). Это оптимизированная по производительности версия Profibus, предназначенная специально для критичных по времени взаимодействий между системами автоматизации и распределенным периферийным оборудованием, где на первое место выходит такой параметр, как продолжительность цикла шины. Profibus-DP был разработан с целью упрощения подключения простых и недорогих устройств ввода-вывода в системах с повышенными временными требованиями. Profibus-DP использует уровни 1 и 2 и реализует набор функций, обеспечивающих интерфейс к прикладной задаче пользователя. Отсутствие уровня 7 обеспечивает быструю и эффективную технологию передачи за счет прямого доступа к функциям канального уровня. Обычно такие системы состоят из одного главного устройства, периодически обращающегося к подчиненным устройствам вводавывода. Таким образом, Profibus-DP используется для организации скоростного обмена небольшими порциями данных с распределенными входами и выходами системы.

Profibus-PA (PA – Process Automation, Автоматизация процессов). Проект Profibus-PA применяется в системах автоматизации взрывоопасных производств, где для обеспечения требований безопасности необходима низковольтная аппаратура. Фактически это аналог Profibus-DP, но в иной физической реализации. При этом на физическом уровне (уровень 1 OSI-модели) технология передачи реализована не с помощью RS-485, а согласно стандарту IEC61158-2 для организации технологии передачи во взрывоопасных средах с внутренней защитой данных. По сигнальным проводам, в соответствии со стандартом, по витой паре идет не только информационный обмен с датчиками и исполнительными механизмами, но и происходит подача питания к ним. Причем энергия в кабеле недостаточна, чтобы образовать искру, что соответствует искробезопасной цепи.

Передающая среда в Profibus, как правило, строится на основе RS-485 интерфейса и "медных" средах передачи. Существует возможность использования и оптического кабеля в качестве физической среды передачи информации. Эта технология находит применение в случае присутствия сильного электромагнитного влияния и при необходимости в высокой скорости передачи информации на большие расстояния по сравнению с RS-485. Переход с медного кабеля на оптоволоконный производится с помощью преобразователей RS-485/Fiber Optic. Для увеличения длины сети линейной конфигурации можно ставить до четырех повторителей. В этом случае общая длина сети может достигать шести километров. Для формирования различных структур сети (древообразной, звездообразной и т.п.) используются активные повторители – разветвители сети.

Для взаимодействия между ведущими — контроллерами (компьютерами) — протокол включает метод передачи маркера, а для взаимодействия между ведомыми — периферийными устройствами — метод MASTER-SLAVE (ведущий-ведомый). Таким образом, в Profibus применяется комбинированный метод, называемый гибридным доступом к передающей среде.

Метод передачи маркера обеспечивает присвоение права доступа к шине в пределах точно определенного временного интервала и используется в Profibus только для

нескольких ведущих устройств, при этом маркер циркулирует между всеми ведущими устройствами в пределах максимального времени оборота маркера.

Метод ведущий-ведомый позволяет ведущему устройству (активной станции), которому в настоящий момент принадлежит право передачи информации, взаимодействовать с ведомыми устройствами (пассивными станциями). Каждое ведущее устройство имеет возможность получать данные от ведомых устройств и передавать данные ведомым устройствам.

Ведомые устройства — это обычные периферийные устройства. Типичными ведомыми устройствами являются датчики, исполнительные устройства и передаточные механизмы. У них нет прав доступа к шине — то есть они могут только подтверждать принимаемые сообщения или передавать сообщения ведущему устройству по его запросу. Ведомые устройства называются также "пассивными станциями".

На рис.4.11 показана структура Profibus с тремя активными станциями (ведущими

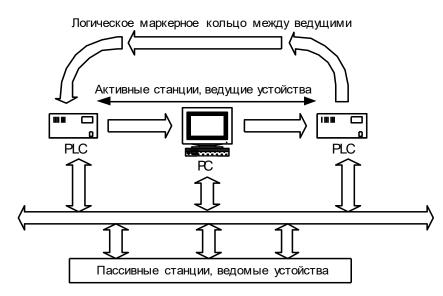


Рис. 4.11. Гибридный метод управления доступом

устройствами) и с несколькими пассивными станциями (ведомыми устройствами).

Три ведущих устройства образуют логическое маркерное кольцо. После того как активная станция получает маркер, ей на определенное время дается разрешение выполнять на шине функции ведущего устройства. Она может взаимодействовать со всеми ведомыми станциями по методу MASTER-SLAVE и со всеми ведущими станциями, участвующими во взаимоотношениях по методу передачи маркера.

На уровне цеха чаще всего используются стандартные сетевые технологи — это комбинация стандартов, топологий, протоколов и реализующих их программно-аппаратных средств, необходимых для создания работоспособной сети. Стандартные сети выпускаются многими формами, по ним имеется подробная документация, накоплен большой опыт их эксплуатации. Наибольшее распространение среди стандартных сетей получила сеть Ethernet. Сеть Ethernet сейчас наиболее популяра в мире и по сути признана в качестве основного сетевого решения на уровне III архитектуры информационной системы.

Mexанизмы обмена данными (OPC - OLE for Process Control)

Существует множество информационных систем, для которых важной составляющей является возможность взаимодействия с устройствами или другими

системами. Для организации информационного взаимодействия должен быть предоставлен некий интерфейс обмена данными, его сложность зависит от множества параметров — физической реализации, потоков и объемов данных и др. Стандарт ОРС, разрабатывался с целью сократить затраты на создание и сопровождение приложений промышленной автоматизации. Основное его предназначение — обеспечение универсального механизма обмена данными между датчиками, исполнительными механизмами, контроллерами, устройствами связи с объектами и системами представления технологической информации, оперативного диспетчерского управления, а также системами управления базами данных.

Особый класс ОРС-приложений представляют собой *ОРС-серверы конкретных* аппаратных устройствв. Они создают своего рода абстракцию аппаратуры, позволяя любому ОРС-клиенту записывать и считывать данные с устройства. Устройство, для которого есть ОРС-сервер, может использоваться вместе с любой современной SCADA-системой.

ОРС-взаимодействие основано на схеме клиент-сервер. ОРС-клиент, например SCADA, вызывая определенные функции объекта ОРС-сервера, подписывается на получение определенных данных с определенной частотой. В свою очередь, ОРС-сервер, опросив физическое устройство, вызывает известные функции клиента, уведомляя его о получении данных и вручая сами данные. Таким образом, при ОРС-взаимодействии используются как прямые (от клиента к серверу), так и обратные (от сервера к клиенту) вызовы. В настоящее время насчитывается более 1000 серверов ОРС для аппаратуры ведущих мировых производителей.

ОРС-серверы могут располагаться на нескольких компьютерах или контроллерах, и доступ к любому из них может осуществляться с любого компьютера сети. Стандарт ОРС относится только к интерфейсам, которые ОРС-сервер предоставляет клиентским программам. Применение ОРС-сервера при разработке заказных программ позволяет скрыть от разработчика всю сложность общения с аппаратурой, предоставляя простой и удобный метод доступа к аппаратуре через интерфейсы СОМ-объекта.

Как и в любой другой клиент-серверной технологии, основой ОРС являются понятия клиента и сервера. Сервер — это программа, которая непосредственно взаимодействует с устройством или информационной системой. Клиент — это программа, которой нужен доступ к устройству или системе. У каждого сервера может быть несколько клиентов, а каждый клиент может обмениваться данными с несколькими серверами. В свою очередь ОРС-сервер (в зависимости от реализации) сам может выступать в роли ОРС-клиента по отношению к другому ОРС-серверу (рис. 4.12).

Клиентская программа и OPC-сервер могут быть установлены на одном и том же компьютере, или на разных компьютерах сети Ethernet. При наличии нескольких компьютеров каждый из них может содержать OPC-серверы и подключенные к нему физические устройства. В такой системе любой OPC-клиент с любого компьютера может обращаться к любому OPC-серверу, в том числе к расположенному на другом компьютере сети.

С ОРС - решениями интеграция в гетерогенные (неоднородные) системы становится достаточно простой. Применительно к SCADA-системам ОРС серверы, расположенные на всех компьютерах системы управления производственного предприятия, стандартным способом могут поставлять данные в программу визуализации, базы данных и т. п., уничтожая, в некотором смысле, само понятие неоднородной системы.

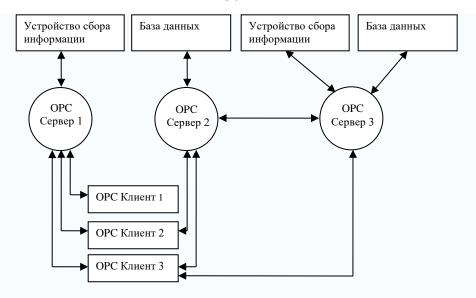


Рис.4.12. Использование ОРС-сервера в системах интеграции

Например, SCADA **на рис. 4.13** может обратиться за данными к модулю вводавывода по пути, указанному на этом рисунке штриховой линией, при этом компьютеры и контроллеры в данной архитектуре могут работать с различными промышленными сетями.

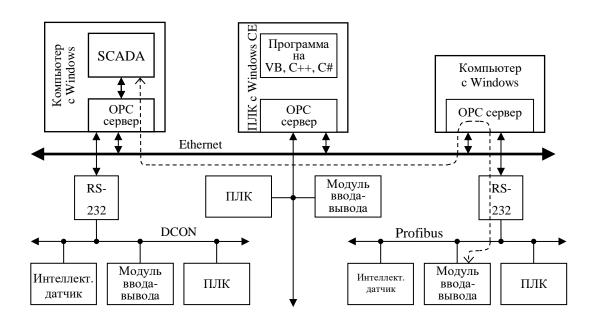


Рис. 4.13. Пример применения *OPC-технологии* для сетевого доступа к данным в системах автоматизации

При использовании и оборудования разных производителей на компьютере (контроллере) может быть установлено несколько ОРС-серверов разных производителей.

Эти приложения могут располагаться как на одном компьютере, так и быть распределенными по сети. Реализация ОРС основана на объектной модели СОМ/DCОМ фирмы Microsoft. COM (Component Object Model) – модель многокомпонентных объектов, позволяющая вызывать те или иные функции (методы) этих объектов так, как будто объекты находятся «рядом». Объект может находиться и в самом деле рядом в адресном пространстве приложения – тогда это просто СОМ. Если же объект находится в другой программе на том же компьютере или на другом узле сети, то это DCOM – Distributed

(распределенная) СОМ. По сути DCОМ – универсальная распределенная модель компонентного объекта.

Фактически ОРС сегодня признан в качестве стандартного механизма взаимодействия программного обеспечения АСУ ТП.

Использование Internet-mexнологии в задачах управления

В последние годы большой интерес вызвало использование Internet-технологии в задачах управления, причем не только на уровне корпоративного управления, но даже нам уровне технологического оборудования. Для этих целей используются web—серверы, которые могут быть расположены как на обычных персональных компьютерах, так непосредственно в контроллерах или модулях ввода-вывода (так называемы микро-web-серверы). С помощью любого web—браузера, например Internet Explorer, можно управлять технологическим процессом, открыв на микро -web-сервере Internet -страничку, с мнемосхемой технологического процесса.

Существует два различных способа управления через Internet, на основе которых построен ряд коммерческих продуктов:

- способ удаленного терминала;
- способ деления SCADA пакетам на серверную и клиентскую части.

Удаленный терминал, можно представить как удлинение проводов мыши, клавиатуры и монитора с помощью Интернета, при этом визуализация выполняется в окне web-браузера. Управляемая программа исполняется на удаленном от пользователя компьютере, на котором устанавливается специализированный web-сервер. Прямо из web-браузера можно работать с запущенными на web-сервере программами, с файловой системой, запускать программы, контролировать выполняющиеся приложения, устанавливать права доступа к системе, получать информацию о загрузке процессора и оперативной памяти. Удаленный терминал можно использовать для управления любой программой, в том числе и SCADA. Его недостатком является требование большой пропускной способности Internet-канала, поскольку через Internet передается вся информация, отображаемая на экране удаленного компьютера.

Способ деления — SCADA пакетам на серверную и клиентскую части предусматривает, что клиентская часть представляет собой web-браузер, который просматривает специализированную web-страницу, находящуюся на web-сервере. На этой странице создается специализированный интерфейс с графикой и анимацией. Поскольку основная часть визуальной динамики пользовательского интерфейса исполняется на клиентском компьютере, а с сервера передаются только данные об объекте автоматизации, существенно снижаются требования к пропускной способности Internet-канала. Данные между клиентом и сервером могут пересылаться в виде сообщений SOAP (Simple Object Access Protocol), которые используют транспортный протокол HTTP. SOAP активизирует два процесса, которые могут находиться на разных компьютерах, но взаимодействуют друг с другом независимо от программных и аппаратных платформ, на которых они реализованы.

Взаимодействие между клиентской и серверной частью (рис.4.14) выглядит следующим образом. Когда пользователь хочет получить данные от контроллера (PLC) через Internet, он нажимает некоторую командную кнопку в окне web-браузера. Этот запрос посылается серверу через Internet. Когда web-сервер получает сообщение, запрос направляется обработчику скрипта, который также располагается на web-сервере. Web-сервис (Web service) создает запрашиваемые данные или передает управление программе,

которая общается с контроллером через OPC-сервер. Полученные таким образом данные выкладываются на web-страницу, которую видит пользователь с помощью web-браузера. При входе на web-сервер операционная система производит идентификацию пользователя и предоставляет доступ к информации в соответствии с его правами.

Web-серверы для Internet обычно располагаются на мощных компьютерах и содержат жесткие диски большой емкости. Однако для удаленного управления в АСУ ТП часто достаточно иметь на сервере всего одну несложную web-страницу. Для этого используются микро-web-серверы (встраиваемые web-серверы, Embedded Web Server), выполненные в виде микросхемы, которая располагается на печатной плате PLС или интеллектуального датчика, что позволяет применять web-серверы даже внутри датчика. Каждый микро-web-сервер доступен по своему Internet-адресу. Технологию применения микро-web-серверов называют «встроенным Интернетом» (Embedded Internet). В заключении отметим, что в настоящее время большинство коммерческих SCADA—пакетов имеют средства построения web—интерфейса.

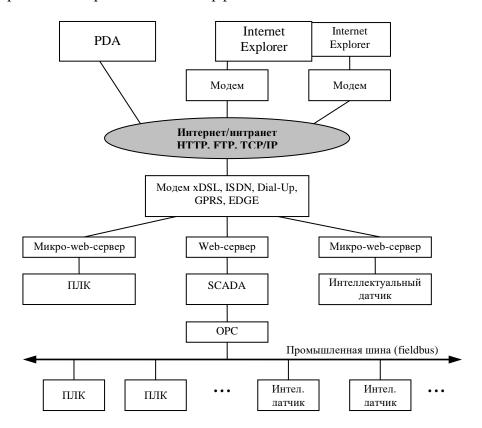


Рис. 4.14. Вариант архитектуры автоматизированной системы, использующей Internet

Вопросы для самоконтроля

- На каких принципах строится архитектура современных информационных систем технологических процессов?
- Какие аппаратно-программные средства используют для обеспечения нижнего уровня автоматизации?
- Чем отличаются промышленные компьютеры (PC) от промышленных программируемых контроллеров (PLC)? Чем вызвана необходимость использования PC в информационных системах технологических процессов?
- Какие коммуникационные средства используются на верхнем уровне автоматизации? Для обеспечения решения каких задач их используют? Чем обусловлен выбор тех или иных средств?
- Для каких целей используют промышленные сети? В чем заключаются их преимущества перед традиционными средствами передачи данных? На какие уровни они подразделяются?
- Какие функции выполняют физические интерфейсы соединений? Приведите примеры стандартных физических интерфейсов и дайте их краткую характеристику.