

Тема 4.1. ЦИФРОВЫЕ СИСТЕМЫ АВТОМАТИЗАЦИИ

1. Структура цифровых систем автоматизации
2. Требования к программному обеспечению систем управления
3. Системы автоматического контроля и сбора информации (SCADA)
4. Основные этапы проектирования микропроцессорных систем
5. Порядок разработки программного обеспечения

1. СТРУКТУРА ЦИФРОВЫХ СИСТЕМ АВТОМАТИЗАЦИИ

Электронные системы управления, создаваемые на базе дискретных элементов и интегральных микросхем, выполняющих какую-либо определенную задачу управления, относятся к системам с жесткой логикой, т. е. алгоритм их функционирования определяется схемотехникой системы. У микропроцессорных систем такое ограничение отсутствует, т. е. при одной и той же структуре данные системы могут реализовывать различные алгоритмы управления вследствие соответствующего изменения записи команд в элементах памяти системы. Благодаря этому микропроцессорные системы образуют особый класс электронных систем управления и обладают рядом уникальных возможностей с точки зрения реализации самых сложных задач управления.

В микропроцессорной системе обработка информации ведется в двоичном цифровом коде. Поэтому все многообразие поступающих в систему сигналов должно быть сведено к единой двоичной кодовой структуре, т. е. структуре вида «логический 0» или «логическая 1». Сигналы, поступающие в систему управления, можно условно разделить на следующие группы:

- сигналы от контактных или других датчиков, имеющие только два возможных состояния — открыт («логическая 1») и закрыт («логический 0»);
- сигналы от терминального устройства, т. е. от элементов системы, на которые воздействует оператор для корректирования действия системы управления (например, датчик положения педали управления подачей топлива либо контроллер управления). К этой группе могут быть отнесены и различные запросы на индикацию состояния тех или иных элементов системы управления;
- информация о режимах работы агрегатов (температура узлов, нагрузочный режим, напряжение, частота вращения двигателя и др.).

Главными критериями применения цифровых систем автоматизации являются надежность, эффективность и безопасность работы. Программное и аппаратное обеспечение систем автоматизации создается тщательно, на базе проверенных годами платформ и технологий. Эти платформы и технологии часто производятся в небольших количествах (иногда только по заказу) и имеют достаточно высокую стоимость. Подобные системы сложны и трудоемки в сопровождении и администрировании.

Непрерывно возрастающая мощность при сравнительной дешевизне делают экономически оправданными попытки широкого использования «персональных» компьютеров (РС-компьютеров) на сложных технологических объектах.

Системы автоматизированного управления на базе ПК имеют определенные достоинства и недостатки.

Достоинства состоят в существенной экономии средств, когда ПК является частью технологического оборудования, открытости программного и аппаратного обеспечения, высокой производительности и низкой цене. Компьютерные средства имеют практически неограниченный объем ОЗУ, позволяют решать задачи визуализации без значительных дополнительных затрат на проектирование, использовать пользовательские функции, написанные на языке «СИ».

К числу **недостатков**, связанных с применением ПК, следует отнести необходимость реализации режима реального времени в основном только за счет расширения системы. Для обычных ПК отсутствует возможность масштабирования аппаратного обеспечения (в семействах программируемых контроллеров имеется значительно большее разнообразие устройств различных классов мощности). Аппаратное обеспечение стандартных ПК (но не ПК промышленного исполнения) намного уступает программируемым контроллерам с точки зрения надежности. То

же самое касается и системного программного обеспечения, так как оно значительно проще в программируемых контроллерах, чем в ПК. В обычных компьютерных средствах нет устройств памяти, буферизуемых батарей. Таким образом, невозможна организация «перманентных» переменных. В этих системах невозможно распознавание ситуации перебоя электропитания.

К настоящему времени выработана четырехуровневая схема, которой придерживаются все производители систем автоматизации технологических процессов:

- 1) Расчет и анализ финансово-экономических показателей системы автоматизации.
- 2) Разработка станции оперативного управления для технологического персонала.
- 3) Согласование программируемых логических контроллеров, управляющих узлами технологического объекта.
- 4) Выбор датчиков и исполняющих устройств.

Блок-схема цифровой системы сбора данных и управления показана на рис. 1.



Рис. 1. Блок-схема цифровой системы сбора данных и управления

Компьютер в составе распределенного комплекса автоматизации, функционирующий в качестве станции оперативного управления, выполняет следующие задачи:

- обеспечивает сбор данных от контроллеров;
- осуществляет обработку полученных данных;
- отображает результаты обработки данных на устройствах вывода информации;
- обрабатывает действия оперативного персонала и передает соответствующие команды контроллерам;
- накапливает результаты обработки данных и действий персонала.

Преобразование различных сигналов в требуемый их вид (цифровой код) для последующей обработки центральным процессором (ЦПУ) выполняют предварительные устройства, к которым можно отнести аналого-цифровые (АЦП) и цифроаналоговые (ЦАП) преобразователи. АЦП применяют для преобразования непрерывного линейного сигнала датчиков температуры, давления, напряжения в цифровой код, а ЦАП — для обратного преобразования.

Для связи между выходами микропроцессора и исполнительными устройствами системы управления используются усилители сигналов или коммутационные элементы (силовые цепи).

Все сигналы от внешних источников, преобразованные к единому виду, поступают на интерфейс ввода-вывода, который может входить в состав микросхемы процессора или выполняться в виде отдельных элементов. Устройство ввода-вывода обеспечивает совместную работу ЦПУ и всех других устройств системы, являющихся по отношению к ЦПУ внешними.

Работа с внешними устройствами выполняется либо по методу периодического опроса их состояния, либо посредством организации системы прерываний от них. При работе микропроцессора с реализацией системы прерываний в нем осуществляется следующий порядок действий:

- 1) в момент, когда одно из внешних устройств готово выдать или принять очередную информацию или оказать воздействие на функционирование системы управления, оно посылает в ЦПУ сигнал готовности (запрос на прерывание);

2) получив сигнал готовности от внешнего устройства, ЦПУ вначале заканчивает выполнение текущей команды, а затем приостанавливает выполнение действий, предусмотренных основной программой, и выдает сигнал готовности начать работу, связанную с возникшим прерыванием (разрешение прерывания);

3) при наличии обоих указанных сигналов готовности происходит обработка прерывания, т.е. выполнение подпрограммы, предусмотренной запросом данного внешнего устройства;

4) если во время решения ЦПУ текущей задачи сигнал готовности прислали несколько внешних устройств, то первой будет принята для обработки или выдана информация внешнему устройству со старшим приоритетом. Уровень приоритетности внешних устройств задается либо при проектировании системы, либо закладывается в программу.

Далее обрабатывается информация внешних устройств с очередностью, определяемой старшинством их приоритета. Число градаций старшинства приоритетов (так называемая глубина прерываний) зависит от типа микропроцессора. Она колеблется от 2 до 8 и более.

Для обеспечения работы микропроцессорной системы управления в реальном масштабе времени, т.е. с выдачей необходимых команд в определенные периоды времени, в ее состав вводят таймер, который обычно выполняют в виде отдельной интегральной микросхемы. Получив управляющую команду (управляющее слово), таймер формирует определенную последовательность временных сигналов.

2. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Специфика промышленного применения накладывает свой отпечаток на используемое программное обеспечение. Можно сформулировать следующие основные требования к программному обеспечению

Оно должно быть достаточно надежным. Действительно, одно дело — когда зависает редактор текста в офисной программе, совсем иное — когда неправильно работает программа, управляющая производственным агрегатом или транспортным средством.

Программное обеспечение должно быстро реагировать на какие-либо внешние события или изменения в параметрах управляемых процессов. Системы, работающие в соответствии со вторым требованием, часто называют системами реального времени. Типовое время реагирования на внешние воздействия, необходимое современным промышленным системам, составляет десятки микросекунд.

Программные средства должны быть удобны в использовании, соответствовать решаемой задаче, иметь логичный интерфейс. Данные требования должны предъявляться вообще к любому программному продукту. Для систем сбора данных и управления выработаны определенные правила. Они не стандартизированы, но все же разработчики известных систем их придерживаются. Основная преследуемая цель — минимизация ошибок персонала по вине путаного интерфейса.

Программные средства должны позволять выполнять переконфигурацию «на ходу», иметь возможность изменения настроек системы (форм представления, графиков, мнемосхем) и условий обработки информации о технологическом объекте (шкал, уставок, размерностей и т.д.) в конкретной подсистеме, не прерывая и не мешая другим подсистемам.

Важным свойством программного обеспечения является выборочное резервирование. Это одно из самых ресурсоемких, но обязательных требований для систем повышенной ответственности. Однако на практике не вся управленческая информация требует резервирования, поэтому резервирование должно быть выборочным, настраиваемым.

Программное обеспечение должно обеспечивать свойство масштабируемости системы. Система управления должна обладать способностью работать как с малым, так и с очень большим количеством параметров, эффективно использовать аппаратные ресурсы.

Программное обеспечение должно обеспечивать возможность внешнего доступа к информации о работе системы сбора данных и управления. Для решения многих задач (например, для расчета финансово-экономических показателей) требуется доступ к информации, получаемой системой сбора данных и управления в ходе работы. Этот доступ должен быть оперативным, простым, но с разграничением прав (экономист не должен иметь возможности управлять

объектом). Какой бы ни была интенсивность работы «внешних» пользователей, никакого влияния на работоспособность системы управления они оказывать не должны.

Система управления должна обладать свойством расширяемости. Под расширяемостью понимается возможность дополнительного добавления новых типов контроллеров, новых типов рабочих мест, новых форм хранения информации и т.д., т.е. возможность развития системы в будущем.

Очевидно, что система должна быть модернизируемой. Очень часто ошибки при модернизации одной из подсистем (обычно с целью оптимизации) приводят к появлению ошибок в других подсистемах. Проблема усугубляется тем, что системы подобной сложности создаются не одним человеком, а целой группой разработчиков.

Процесс управления должен быть многозадачным. Это требование часто (но не всегда) предъявляется к программному обеспечению систем управления. Оно проистекает из подчас чрезвычайно сложной и многоуровневой природы управляемых процессов, когда необходимо одновременно реализовать сложные алгоритмы управления различными аспектами или частями реального объекта. По каждой задаче выполняется определенная доля работы по управлению объектом, и все задачи делят ресурсы вычислительной системы в зависимости от своего приоритета и от внешних и внутренних событий, конкретно с ними связанных.

Другим тесно связанным с многозадачностью понятием является многопоточность. Под этим термином, как правило, понимают возможность выполнения в рамках одной задачи нескольких независимых процессов (потоков команд), которые в отличие от задач пользуются общими участками кода и данных. Названным требованиям должны удовлетворять все уровни программного обеспечения системы.

Условно можно разделить программное обеспечение на три уровня.

- 1) Базовая система ввода-вывода (BIOS).
- 2) Операционная система и драйверы (ОС).
- 3) Собственно прикладные программы.

BIOS осуществляет непосредственную связь между аппаратурой и программным обеспечением верхних уровней.

Операционная система (ОС) выполняет базовые функции связи с оператором, запуска программ, распределения памяти, поддержки файловой системы и т.п. Создатель системы автоматизации, как правило, встает перед выбором: или использовать многозадачное ядро реального времени, работающее над DOS, или перейти на другую операционную систему. В первом случае используются самодельные или коммерчески распространяемые библиотеки программ, реализующие функции многозадачной ОС, например RTKernel. Во втором случае открывается богатый выбор операционных систем, разработанных специально для применения в системах «жесткого» реального времени. Эти ОС часто так и называют операционными системами реального времени. На рынке существует несколько десятков операционных систем реального времени. У нас известны такие ОС, как QNX, OS 9000, VxWorks, iRMX, VRTX, Nucleus и др.

Что же касается таких операционных систем, как Windows NT и OS/2, то эти новейшие операционные системы оказывают определенное давление на рынок операционных систем реального времени. Тем не менее обе системы ориентированы на офисное применение и не могут обеспечить скорость отклика на внешние воздействия, сравнимую со скоростью специализированных ОС реального времени. Кроме общей неразвитости механизмов управления приоритетами могут возникать проблемы некоторой непредсказуемости поведения этих ОС. Например, совершенно неожиданно система может инициировать массивный свопинг на диск, забирая для этого мероприятия практически все ресурсы. Однако системы на базе Windows NT широко используются как ОС для терминалов в дополнение к автономно действующим контроллерам (например, в системах Honeywell).

Honeywell — американская корпорация, производящая электронные системы управления и автоматизации.

Не так давно Microsoft выпустила на рынок ОС Windows CE для встраиваемых приложений. Немного особняком стоит программное обеспечение, предназначенное для эксплуатации на верхнем уровне систем автоматизации, например в диспетчерских электростанций, пультах управления сложными агрегатами и т.д.

С середины 80-х годов ряд фирм-разработчиков программных систем приступили к разработке и выпуску универсальных и открытых пакетов прикладных программ для компьютерных операторских станций, предназначенных не для конкретной микропроцессорной системы, но приспособленных для применения в разных программно-технических и сетевых средах.

Свойство открытости состоит в том, что пакет прикладных программ поддерживает совокупность интерфейсов и драйверов, позволяющих использовать пакет для широкого класса микропроцессорных контроллеров, систем и сетей передачи данных, содержит средства разработки новых (не предусмотренные разработчиком) интерфейсов и драйверов для микропроцессорных приборов и сетей. Он позволяет расширять функциональные возможности систем в соответствии с заданиями на конкретные проекты — путем подключения программ пользователя.

Программные системы и пакеты прикладных программ, обеспечивающие работу компьютерных операторских станций, в литературе получили наименование SCADA-программы (аббревиатура SCADA означает Supervisory Control And Data Acquisition — сбор данных, наблюдение и управление).

Эти программы обеспечивают визуализацию текущей и исторической информации на дисплеях (в формах динамизированных мнемосхем, гистограмм, анимационных изображений, таблиц, трендов, аварийных сообщений и т. д.). Они реализуют печать отчетов и протоколов в задаваемых формах — по времени или по запросу оператора, осуществляют регистрацию аварийных ситуаций в моменты их возникновения и вывод аварийных сообщений на экран или на внешние устройства, реализуют ввод команд и сообщений оператора, их обработку или передачу в контроллеры и другие устройства.

В последнее время получает распространение программное обеспечение класса MES.

MES (от англ. manufacturing execution system - система управления производственными процессами) - специализированное прикладное программное обеспечение, предназначенное для решения задач синхронизации, координации, анализа и оптимизации выпуска продукции в рамках какого-либо производства. MES-системы относятся к классу систем управления уровня цеха, но могут использоваться и для интегрированного управления производством на предприятии в целом.

Основной задачей программного обеспечения **MES** является оптимизация управления производством в целом, включая планирование запасов комплектующих, расписание технологических процессов и т.д. Системы MES, с одной стороны, взаимодействуют с системами SCADA, образуя единую систему управления производством, с другой стороны, — часто интегрируются в систему планирования и управления предприятием в целом.

3. СИСТЕМЫ АВТОМАТИЧЕСКОГО КОНТРОЛЯ И СБОРА ИНФОРМАЦИИ (SCADA)

SCADA (*supervisory control and data acquisition*, диспетчерское управление и сбор данных) - программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью АСУ ТП, автоматизированных систем учета электроэнергии, системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д. SCADA-системы используются во всех отраслях хозяйства, где требуется обеспечивать автоматическое управление технологическими процессами в режиме реального времени. Данное программное обеспечение устанавливается на компьютеры и, для связи с объектом, использует драйверы ввода-вывода или OPC-серверы.

OPC (OLE for Process Control) - семейство программных технологий, предоставляющих единый интерфейс для управления объектами автоматизации и технологическими процессами. Многие из OPC протоколов базируются на Windows-технологиях.

Иногда SCADA-системы комплектуются дополнительным ПО для программирования промышленных контроллеров. Такие SCADA-системы называются интегрированными и к ним добавляют термин **SoftLogic**.

Термин “SCADA” имеет двойное толкование. Наиболее широко распространено понимание SCADA как приложения, то есть программного комплекса, обеспечивающего выполнение указанных функций, а также инструментальных средств для разработки этого программного обеспечения. Однако часто под SCADA-системой подразумевают программно-аппаратный комплекс. Подобное понимание термина SCADA более характерно для раздела телеметрия (телеизмерение).

Телеметрия, телеизмерение — информация о значениях измеряемых параметров контролируемых и управляемых объектов методами и средствами телемеханики

Значение термина SCADA претерпело изменения вместе с развитием технологий автоматизации и управления технологическими процессами. В 80-е годы под SCADA-системами чаще понимали программно-аппаратные комплексы сбора данных реального времени. С 90-х годов в связи с тем, что всё большая часть функций автоматического управления решается не аппаратными, а программными средствами, термин SCADA больше используется для обозначения только программной части человеко-машинного интерфейса АСУ ТП.

Основные задачи, решаемые SCADA-системами

SCADA-системы решают следующие задачи:

- Обмен данными с “устройствами связи с объектом”, (то есть с промышленными контроллерами и платами ввода/вывода) в реальном времени через драйверы.
- Обработка информации в реальном времени.
- Логическое управление.
- Отображение информации на экране монитора в удобной и понятной для человека форме.
- Ведение базы данных реального времени с технологической информацией.
- Аварийная сигнализация и управление тревожными сообщениями.
- Подготовка и генерирование отчетов о ходе технологического процесса.
- Осуществление сетевого взаимодействия между SCADA-станциями (компьютерами).
- Обеспечение связи с внешними приложениями (СУБД, электронные таблицы, текстовые процессоры и т. д.). В системе управления предприятием такими приложениями чаще всего являются приложения, относимые к уровню «MES».

Главными функциями SCADA-программ являются отображение технологического процесса в виде мнемосхем, сигнализация об аварийных ситуациях, ведение системного журнала, обеспечение общего управления процессом со стороны оператора и т.п. Таким образом, SCADA-программы обеспечивают реализацию основных функций операторских станций в реальном времени: собирают текущую технологическую информацию от контроллеров или других приборов и устройств, связанных непосредственно или через сеть с операторской станцией; проводят необходимую первичную обработку измерительной информации, а также вычислительную и логическую обработку технологических данных в операторских станциях; осуществляют архивизацию и хранение текущей информации и ее дальнейшую обработку.

Понятие SCADA обычно относится к централизованным системам контроля и управления всей системой, или комплексами систем, осуществляемого с участием человека. Большинство управляющих воздействий выполняется автоматически контроллерами, находящимися на объекте контроля и управления или программируемыми логическими контроллерами (ПЛК).

Непосредственное управление процессом обычно обеспечивается контроллерами оборудования, а SCADA управляет режимами работы. Например, программируемый контроллер может управлять потоком охлаждающей воды внутри части производственного процесса, а SCADA система может позволить операторам изменять параметры потока, менять маршруты движения жидкости, заполнять те или иные ёмкости, а также следить за сигналами сигнализации, такими как — потеря потока и высокая температура, которые должны быть отображены, записаны, и на которые оператор должен своевременно реагировать. Цикл управления с обратной связью происходит через контроллеры оборудования или ПЛК, в то время как SCADA система контролирует полное выполнение цикла.

Основные компоненты SCADA

SCADA—система обычно содержит следующие подсистемы:

- Драйверы или серверы ввода-вывода — программы, обеспечивающие связь SCADA с промышленными контроллерами, счётчиками, АЦП и другими устройствами ввода-вывода информации.
- Система реального времени — программа, обеспечивающая обработку данных в пределах заданного временного цикла с учетом приоритетов.
- Человеко-машинный интерфейс (HMI, англ. *Human Machine Interface* — инструмент, который представляет данные о ходе процесса человеку оператору, что позволяет оператору контролировать процесс и управлять им. Программа-редактор для разработки человеко-машинного интерфейса.
- Система логического управления — программа, обеспечивающая исполнение пользовательских программ (скриптов) логического управления в SCADA-системе. Набор редакторов для их разработки.
- База данных реального времени — программа, обеспечивающая сохранение истории процесса в режиме реального времени.
- Система управления тревогами — программа, обеспечивающая автоматический контроль технологических событий, отнесение их к категории нормальных, предупреждающих или аварийных, а также обработку событий оператором или компьютером.
- Генератор отчетов — программа, обеспечивающая создание пользовательских отчетов о технологических событиях. Набор редакторов для их разработки.
- Внешние интерфейсы — стандартные интерфейсы обмена данными между SCADA и другими приложениями.

SCADA-программы реализуют подключение и организацию взаимодействия прикладных программ пользователя с переменными базы данных реального времени и командами оператора, осуществляют информационные сетевые взаимодействия между узлами системы управления.

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

- 1) инструментальную подсистему разработки (Development Kit),
- 2) исполнительную подсистему (Runtime), поддерживающую работу программного обеспечения операторских станций в реальном времени.

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

Исполнительная подсистема поддерживает, как правило, работу программного обеспечения отдельной станции. Инструментальная система технически тоже может поддерживать в реальном времени работу операторской станции. Очевидно, что она же может быть использована и в качестве исполнительной системы. В некоторых специализированных SCADA-программах инструментальная и исполнительная системы не разделены и представляют собой единое целое. В тексте обзора возможность использования инструментальной системы в качестве исполнительной по умолчанию не оговаривается.

Многие системные интеграторы или подразделения АСУ крупных предприятий, как правило, имеют свои наработки в этой области. В то же время есть фирмы, специализирующиеся на разработке универсальных SCADA-программ, таких, как Genesis (Iconics), FixDmacs (Intellution), InTouch (Wonderware). Из отечественных разработок хотелось бы отметить программу TraceMode (Adastra Research Group). Граница между программными обеспечениями классов SCADA и HMI весьма условная, поэтому многие системы SCADA могут работать в качестве интерфейса HMI для систем нижнего уровня.

4. ОСНОВНЫЕ ЭТАПЫ ПРОЕКТИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Общая процедура проектирования микропроцессорных систем включает следующие этапы:

1) Составление технического задания. Формулировка требований к системе. На основании требований составляется функциональная спецификация (функции выполняемые контроллером для пользователя). Описание форматов данных.

2) Разработка алгоритма управления. Выбор алгоритма на основе соотношения объема программных и аппаратных средств.

3) Выбор типа микропроцессора или микроконтроллера. Учитываются разрядность, быстродействие, набор команд и способов адресации, объем ПЗУ и ОЗУ, наличие документации (или ее полноты).

4) Разработка структуры аппаратных и программных средств.

Определяются следующие элементы:

а) Состав имеющихся и подлежащих разработке аппаратных модулей, протоколы обмена между модулями, типы разъемов.

б) Разработка конструкции контроллерного устройства.

в) Выбирается объем и тип памяти для хранения данных, номенклатура периферийных устройств, протоколы обмена информацией и состав требуемых сигналов управления системой.

г) Определяется состав и связи программных модулей, язык программирования.

д) Осуществляется выбор средств проектирования и отладки.

5) Разработка и отладка аппаратных средств.

а) Разработка общей принципиальной схемы.

б) Разводка топологии плат.

в) Монтаж макета и его автономная отладка.

Традиционная разработка включает в себя проектирование электрической схемы, ее моделирование, разработку печатной платы или комплекта плат.

Наиболее современной является мезонин-технология. Данная технология унифицирует размеры базовой платы. В состав базовой платы входят базовый микропроцессор или микроконтроллер, память, периферийные схемы. Если этого недостаточно, то дополнительные устройства (многоканальные системы ввода/вывода, дополнительная память, таймеры, счетчики, сетевые адаптеры и др.) размещаются на мезонине (типичный размер 45x99 мм) над базовой платой.

Одна плата может нести десятки мезонинов. Каждый мезонин соединяется с носителем разъемом. Используя серийно выпускаемые платы носители и набор мезонинов, разработчик может быстро реализовать сложные системы. Лидерами в этой области являются фирмы «Green Spring Computers» (США) и «PEP Modular Computer» (Германия). Серийно выпускается более 300 типов мезонинов. В большинстве случаев разработка АС ведется путем использования готовых изделий, выпускаемых рядом производителей.

б) Разработка и отладка программы. Использование интегральной среды разработки, высокоуровневое программирование, низкоуровневое программирование. Оптимизация кода. Использование программных симуляторов.

7) Интеграция разработанного контроллерного устройства в систему.

8) Испытание и Введение в эксплуатацию.

5. ПОРЯДОК РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ПО)

Разработка программного обеспечения включает выбор операционной системы, ее тип и характеристика, номенклатура необходимых программных модулей, характер их взаимодействия, используемый тип языка; трансляция и автономная отладка ПО.

Автономная отладка ПО выполняется с помощью симулятора - программной модели используемого микропроцессора или микроконтроллера. На этом этапе разработчики используют широкий набор средств программирования - компиляторы, ассемблеры, дисассемблеры, отладчики, редакторы связей (компоновщики, загрузчики) и другие.

Исходная программа пишется на языке высокого уровня (например, языке СИ). Программа на языке ассемблер является промежуточным этапом для получения выходной исполняемой программы.

Преобразование ассемблерной программы в исполняемые двоичные коды машинных команд осуществляется в два этапа:

1 этап - Получение объектной программы (объектного модуля) с помощью программы транслятора. В процессе трансляции ассемблер может обнаружить синтаксические ошибки, нарушение правил языка. Ассемблер выдает листинг трансляции, содержащий исходную программу, коды команд, диагностическую информацию и сообщения об ошибках. После трансляции может возникнуть необходимость корректировки исходной программы.

2 этап - Компоновка исполняемой программы. При этом возможно объединение нескольких модулей в одну исполняемую программу. В процессе компоновки между различными программами устанавливаются связи. Объектные программы, полученные в процессе трансляции, не привязаны к определенным адресам ячеек памяти, т.е. они могут перемещаться. Компоновщик привязывает программу к определенным адресам. Компоновщик, помимо основной исполняемой программы, выдает файл с расширением «.map» (таблицу компоновки), содержащий порядок следования и схему распределения исходных файлов по конкретным адресам, а также диагностическую информацию.

Язык ассемблера, как правило, для различных микропроцессоров одинаков, а трансляторы и компоновщики отличаются.

Симуляторы иногда поставляются в виде отдельных средств поддержания программирования. Обычно они входят в состав отладчиков. Отладчики являются основным инструментом разработчика программного обеспечения. Отладчик реализует режимы различные выполнения программы - пошаговый или с остановом в контрольных точках, позволяет производить просмотр и коррекцию содержимого регистров и ячеек памяти, обеспечивает в точке останова контроль выполнения предыдущих шагов программы (просмотр трассы), дисассемблирование команд.

Компилятор (compiler) разбивает программу на ряд последовательных этапов, минимизирует программу, контролирует соединения. Результатом компиляции является создание загрузочного (конфигурационного) файла.

3 этап - Комплексная отладка АС и ПО

Этот этап является наиболее сложным и ответственным. На этом этапе используются специальные средства комплексной отладки - **схемные эмуляторы**, включаемые вместо микропроцессора или микроконтроллера и обеспечивающие возможность контроля ее работы с помощью персонального компьютера. Схемные эмуляторы являются наиболее эффективным средством комплексной отладки систем.

В простейшем случае эмулятор позволяет в процессе отладки многократно изменять программу. Более сложные эмуляторы позволяют останавливать работу системы в заданных контрольных точках аналогично схемному эмулятору.

При этом на дисплее базового компьютера может быть представлено содержимое эмулирующей памяти.

Для универсальных микропроцессорных систем после комплексной отладки производится оценка их производительности путем прогона специального набора тестовых программ (benchmarks).

4 этап - Испытание системы в рабочих условиях.