

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Якушин Владимир Андреевич
Должность: ректор, д.ю.н., профессор
Дата подписания: 06.10.2025 12:10:00
Уникальный программный ключ:
a5427c2559e1ff4b007ed9b1994671e27053e0dc

Министерство науки и высшего образования РФ
Образовательная автономная некоммерческая организация
высшего образования
«Волжский университет имени В.Н. Татищева» (институт)

УТВЕРЖДАЮ
Ректор Якушин В.А.
от 27.06.2024г.

Методическое указание
по выполнению курсовой работы
по дисциплине «Микропроцессорные системы»

Направление подготовки 09.03.01 Информатика и вычислительная
техника

Квалификация (степень) выпускника – бакалавр

Форма обучения – очная, заочная, очно-заочная

Тольятти, 2024 г.

Методическое указание по выполнению курсовой работы по дисциплине «Микропроцессорные системы» составлено с требованиями ФГОС, ВО, ОПОП по направлению подготовки 09.03.01 Информатика и вычислительная техника (уровень высшего образования: бакалавриат) и учебного плана.

Одобрено Учебно-методическим советом вуза
протокол № 63 от 07.05.2024г
Председатель УМС
к.п.н. И.И. Муртаева

Содержание

1. Разработка микропроцессорной системы на основе микроконтроллера	4
1.1. Основные этапы разработки	4
1.2. Разработка и отладка программного обеспечения	9
1.3. Методы и средства совместной отладки аппаратных и программных средств	11
2. Содержание курсовой работы	14
3. Примерные темы курсовых работ:	154

1. Разработка микропроцессорной системы на основе микроконтроллера

1.1. Основные этапы разработки

МПС на основе МК используются чаще всего в качестве встроенных систем для решения задач управления некоторым объектом. Важной особенностью данного применения является работа в реальном времени, т.е. обеспечение реакции на внешние события в течение определенного временного интервала. Такие устройства получили название контроллеров.

Технология проектирования контроллеров на базе МК полностью соответствует принципу неразрывного проектирования и отладки аппаратных и программных средств, принятому в микропроцессорной технике. Это означает, что перед разработчиком такого рода МПС стоит задача реализации полного цикла проектирования, начиная от разработки алгоритма функционирования и заканчивая комплексными испытаниями в составе изделия, а, возможно, и сопровождением при производстве. Сложившаяся к настоящему времени методология проектирования контроллеров может быть представлена так, как показано на [рис. 1.1](#).

В техническом задании формулируются требования к контроллеру с точки зрения реализации определенной функции управления. Техническое задание включает в себя набор требований, который определяет, что пользователь хочет от контроллера и что разрабатываемый прибор должен делать. Техническое задание может иметь вид текстового описания, не свободного в общем случае от внутренних противоречий.

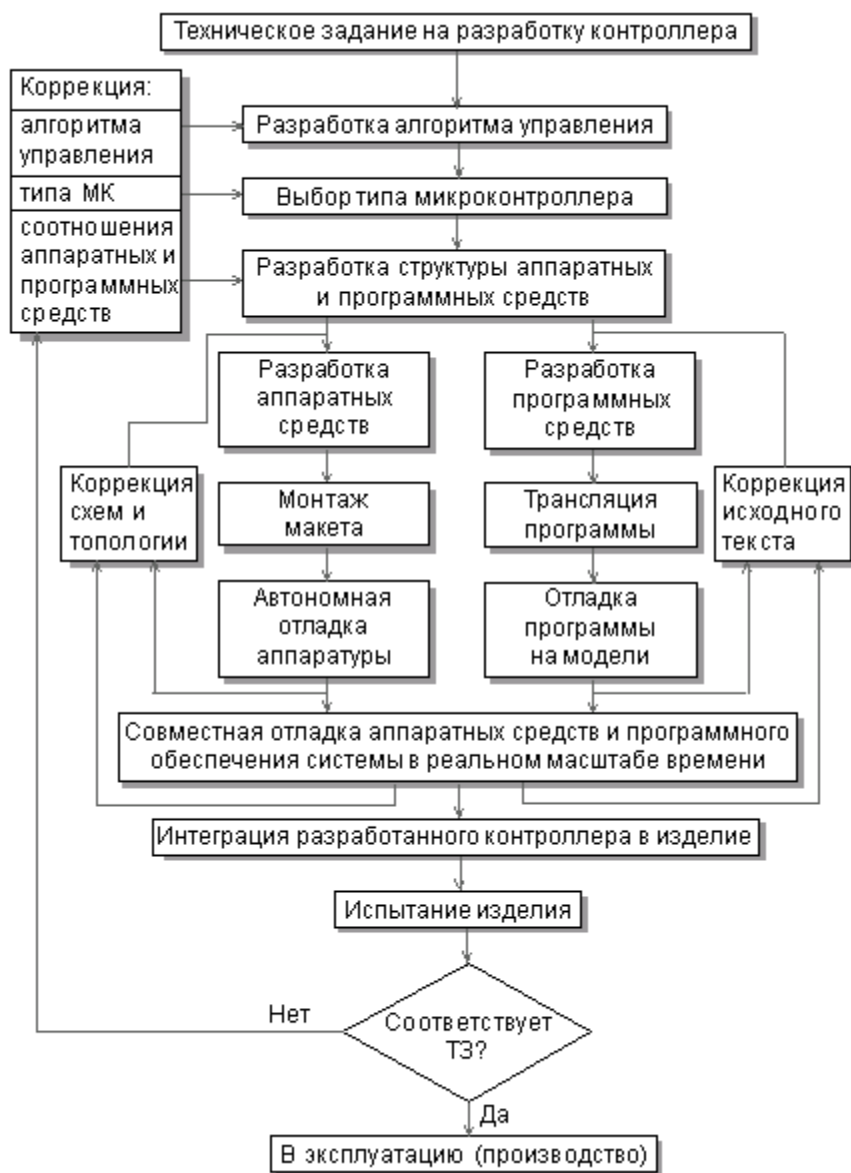


Рис. 1.1. Основные этапы разработки контроллера.

На основании требований пользователя составляется функциональная спецификация, которая определяет функции, выполняемые контроллером для пользователя после завершения проектирования, уточняя тем самым, насколько устройство соответствует предъявляемым требованиям. Она включает в себя описания форматов данных, как на входе, так и на выходе, а также внешние условия, управляющие действиями контроллера.

Функциональная спецификация и требования пользователя являются критериями оценки функционирования контроллера после завершения проектирования. Может потребоваться проведение нескольких итераций, включающих обсуждение требований и функциональной спецификации с

потенциальными пользователями контроллера, и соответствующую коррекцию требований и спецификации. Требования к типу используемого МК формулируются на данном этапе чаще всего в неявном виде.

Этап разработки алгоритма управления является наиболее ответственным, поскольку ошибки данного этапа обычно обнаруживаются только при испытаниях законченного изделия и приводят к необходимости дорогостоящей переработки всего устройства. Разработка алгоритма обычно сводится к выбору одного из нескольких возможных вариантов алгоритмов, отличающихся соотношением объема программного обеспечения и аппаратных средств.

При этом необходимо исходить из того, что максимальное использование аппаратных средств упрощает разработку и обеспечивает высокое быстродействие контроллера в целом, но сопровождается, как правило, увеличением стоимости и потребляемой мощности. Связано это с тем, что увеличение доли аппаратных средств достигается либо путем выбора более сложного МК, либо путем использования специализированных интерфейсных схем. И то, и другое приводит к росту стоимости и энергопотребления. Увеличение удельного веса программного обеспечения позволяет сократить число элементов контроллера и стоимость аппаратных средств, но это приводит к снижению быстродействия, увеличению необходимого объема внутренней памяти МК, увеличению сроков разработки и отладки программного обеспечения. Критерием выбора здесь и далее является возможность максимальной реализации заданных функций программными средствами при минимальных аппаратных затратах и при условии обеспечения заданных показателей быстродействия и надежности в полном диапазоне условий эксплуатации. Часто определяющими требованиями являются возможность защиты информации (программного кода) контроллера, необходимость обеспечения максимальной продолжительности работы в автономном режиме и другие. В результате

выполнения этого этапа окончательно формулируются требования к параметрам используемого МК.

При выборе типа МК учитываются следующие основные характеристики:

- разрядность;
- быстродействие;
- набор команд и способов адресации;
- требования к источнику питания и потребляемая мощность в различных режимах;
- объем ПЗУ программ и ОЗУ данных;
- возможности расширения памяти программ и данных;
- наличие и возможности периферийных устройств, включая средства поддержки работы в реальном времени (таймеры, процессоры событий и т.п.);
- возможность перепрограммирования в составе устройства;
- наличие и надежность средств защиты внутренней информации;
- возможность поставки в различных вариантах конструктивного исполнения;
- стоимость в различных вариантах исполнения;
- наличие полной документации;
- наличие и доступность эффективных средств программирования и отладки МК;
- количество и доступность каналов поставки, возможность замены изделиями других фирм.

Список этот не является исчерпывающим, поскольку специфика проектируемого устройства может перенести акцент требований на другие параметры МК. Определяющими могут оказаться, например, требования к точности внутреннего компаратора напряжений или наличие большого числа выходных каналов ШИМ.

Номенклатура выпускаемых в настоящее время МК исчисляется тысячами типов изделий различных фирм. Современная стратегия модульного проектирования обеспечивает потребителя разнообразием моделей МК с одним и тем же процессорным ядром. Такое структурное разнообразие открывает перед разработчиком возможность выбора оптимального МК, не имеющего функциональной избыточности, что минимизирует стоимость комплектующих элементов.

Однако для реализации на практике возможности выбора оптимального МК необходима достаточно глубокая проработка алгоритма управления, оценка объема исполняемой программы и числа линий сопряжения с объектом на этапе выбора МК. Допущенные на данном этапе просчеты могут впоследствии привести к необходимости смены модели МК и повторной разводки печатной платы макета контроллера. В таких условиях целесообразно выполнять предварительное моделирование основных элементов прикладной программы с использованием программно-логической модели выбранного МК.

При отсутствии МК, обеспечивающего требуемые по ТЗ характеристики проектируемого контроллера, необходим возврат к этапу разработки алгоритма управления и пересмотр выбранного соотношения между объемом программного обеспечения и аппаратных средств. Отсутствие подходящего МК чаще всего означает, что для реализации необходимого объема вычислений (алгоритмов управления) за отведенное время нужна дополнительная аппаратная поддержка. Отрицательный результат поиска МК с требуемыми характеристиками может быть связан также с необходимостью обслуживания большого числа объектов управления. В этом случае возможно использование внешних схем обрaмления МК.

На этапе разработки структуры контроллера окончательно определяется состав имеющихся и подлежащих разработке аппаратных модулей, протоколы обмена между модулями, типы разъемов. Выполняется

предварительная проработка конструкции контроллера. В части программного обеспечения определяются состав и связи программных модулей, язык программирования. На этом же этапе осуществляется выбор средств проектирования и отладки.

Возможность перераспределения функций между аппаратными и программными средствами на данном этапе существует, но она ограничена характеристиками уже выбранного МК. При этом необходимо иметь в виду, что современные МК выпускаются, как правило, сериями (семействами) контроллеров, совместимых программно и конструктивно, но различающихся по своим возможностям (объем памяти, набор периферийных устройств и т.д.). Это дает возможность выбора структуры контроллера с целью поиска наиболее оптимального варианта реализации.

Нельзя не упомянуть здесь о новой идеологии разработки устройств на базе МК, предложенной фирмой "Scenix". Она основана на использовании высокоскоростных RISC-микроконтроллеров серии SX с тактовой частотой до 100 МГц. Эти МК имеют минимальный набор встроенной периферии, а все более сложные периферийные модули эмулируются программными средствами. Такие модули программного обеспечения называются "виртуальными периферийными устройствами", они обеспечивают уменьшение числа элементов контроллера, времени разработки, увеличивают гибкость исполнения. К настоящему времени разработаны целые библиотеки виртуальных устройств, содержащие отлаженные программные модули таких устройств как модули ШИМ и ФАПЧ, последовательные интерфейсы, генераторы и измерители частоты, контроллеры прерываний и многие другие.

1.2. Разработка и отладка программного обеспечения

Содержание этапов разработки программного обеспечения, его трансляции и отладки на моделях существенно зависит от используемых системных средств. В настоящее время ресурсы 8-разрядных МК достаточны для поддержки программирования на языках высокого уровня. Это позволяет

использовать все преимущества структурного программирования, разрабатывать программное обеспечение с использованием отдельно транслируемых модулей. Одновременно продолжают широко использоваться языки низкого уровня типа ассемблера, особенно при необходимости обеспечения контролируемых интервалов времени. Задачи предварительной обработки данных часто требуют использования вычислений с плавающей точкой, трансцендентных функций.

В настоящее время самым мощным средством разработки программного обеспечения для МК являются интегрированные среды разработки, имеющие в своем составе менеджер проектов, текстовый редактор и симулятор, а также допускающие подключение компиляторов языков высокого уровня типа Паскаль или Си. При этом необходимо иметь в виду, что архитектура многих 8-разрядных МК вследствие малого количества ресурсов, страничного распределения памяти, неудобной индексной адресации и некоторых других архитектурных ограничений не обеспечивает компилятору возможности генерировать эффективный код. Для обхода этих ограничений разработчики ряда компиляторов вынуждены были перекладывать на пользователя заботу об оптимизации кода программы.

Для проверки и отладки программного обеспечения используются так называемые программные симуляторы, предоставляющие пользователю возможность выполнять разработанную программу на программно-логической модели МК. Программные симуляторы распространяются, как правило, бесплатно и сконфигурированы сразу на несколько МК одного семейства. Выбор конкретного типа МК среди моделей семейства обеспечивает соответствующая опция меню конфигурации симулятора. При этом моделируется работа ЦП, всех портов ввода/вывода, прерываний и другой периферии. Карта памяти моделируемого МК загружается в симулятор автоматически, отладка ведется в символьных обозначениях регистров.

Загрузив программу в симулятор, пользователь имеет возможность запускать ее в пошаговом или непрерывном режимах, задавать условные или безусловные точки останова, контролировать и свободно модифицировать содержимое ячеек памяти и регистров симулируемого МК.

1.3. Методы и средства совместной отладки аппаратных и программных средств

Этап совместной отладки аппаратных и программных средств в реальном масштабе времени является самым трудоемким и требует использования инструментальных средств отладки. К числу основных инструментальных средств отладки относятся:

- внутрисхемные эмуляторы;
- платы развития (оценочные платы);
- мониторы отладки ;
- эмуляторы ПЗУ.

Внутрисхемный эмулятор – программно-аппаратное средство, способное заменить эмулируемый МК в реальной схеме. Стыковка внутрисхемного эмулятора с отлаживаемой системой производится при помощи кабеля со специальной эмуляционной головкой, которая вставляется вместо МК в отлаживаемую систему. Если МК нельзя удалить из отлаживаемой системы, то использование эмулятора возможно, только если этот микроконтроллер имеет отладочный режим, при котором все его выводы находятся в третьем состоянии. В этом случае для подключения эмулятора используют специальный адаптер-клипсу, который подключается непосредственно к выводам эмулируемого МК.

Внутрисхемный эмулятор – это наиболее мощное и универсальное отладочное средство, которое делает процесс функционирования отлаживаемого контроллера прозрачным, т.е. легко контролируемым, произвольно управляемым и модифицируемым.

Платы развития, или, как принято их называть в зарубежной литературе, оценочные платы (Evaluation Boards), являются своего рода

конструкторами для макетирования электронных устройств. Обычно это печатная плата с установленным на ней МК и всей необходимой ему стандартной периферией. На этой плате также устанавливают схемы связи с внешним компьютером. Как правило, там же имеется свободное поле для монтажа прикладных схем пользователя. Иногда предусмотрена уже готовая разводка для установки дополнительных устройств, рекомендуемых фирмой. Например, ПЗУ, ОЗУ, ЖКИ-дисплей, клавиатура, АЦП и др. Кроме учебных или макетных целей, такие доработанные пользователем платы можно использовать в качестве одноплатных контроллеров, встраиваемых в малосерийную продукцию.

Для большего удобства платы развития комплектуются еще и простейшим средством отладки на базе **монитора** отладки. Используются два типа мониторов отладки: один для МК, имеющих внешнюю шину, а второй – для МК, не имеющих внешней шины.

В первом случае отладочный монитор поставляется в виде микросхемы ПЗУ, которая вставляется в специальную розетку на плате развития. Плата также имеет ОЗУ для программ пользователя и канал связи с внешним компьютером или терминалом. Во втором случае плата развития имеет встроенные схемы программирования внутреннего ПЗУ МК, которые управляются от внешнего компьютера. При этом программа монитора просто заносится в ПЗУ МК совместно с прикладными кодами пользователя. Прикладная программа должна быть специально подготовлена: в нужные места необходимо вставить вызовы отладочных подпрограмм монитора. Затем осуществляется пробный прогон. Чтобы внести в программу исправления, пользователю надо стереть ПЗУ и произвести повторную запись. Готовую прикладную программу получают из отлаженной путем удаления всех вызовов мониторных функций и самого монитора отладки. Возможности отладки, предоставляемые комплектом "плата развития плюс монитор", не столь универсальны, как возможности внутрисхемного эмулятора, да и некоторая часть ресурсов МК в процессе отладки отбирается

для работы монитора. Тем не менее, наличие набора готовых программно-аппаратных средств, позволяющих без потери времени приступить к монтажу и отладке проектируемой системы, во многих случаях является решающим фактором. Особенно если учесть, что стоимость такого комплекта несколько меньше, чем стоимость более универсального эмулятора.

Эмулятор ПЗУ – программно-аппаратное средство, позволяющее замещать ПЗУ на отлаживаемой плате, и подставляющее вместо него ОЗУ, в которое может быть загружена программа с компьютера через один из стандартных каналов связи. Это устройство позволяет пользователю избежать многократных циклов перепрограммирования ПЗУ. Эмулятор ПЗУ нужен только для МК, которые могут обращаться к внешней памяти программ. Это устройство сравнимо по сложности и по стоимости с платами развития и имеет одно большое достоинство: универсальность. Эмулятор ПЗУ может работать с любыми типами МК.

Эмулируемая память доступна для просмотра и модификации, но контроль над внутренними управляющими регистрами МК был до недавнего времени невозможен.

В последнее время появились модели интеллектуальных эмуляторов ПЗУ, которые позволяют "заглядывать" внутрь МК на плате пользователя. Интеллектуальные эмуляторы представляют собой гибрид из обычного эмулятора ПЗУ, монитора отладки и схем быстрого переключения шины с одного на другой. Это создает эффект, как если бы монитор отладки был установлен на плате пользователя и при этом он не занимает у МК никаких аппаратных ресурсов, кроме небольшой зоны программных шагов, примерно 4К.

Этап совместной отладки аппаратных и программных средств в реальном масштабе времени завершается, когда аппаратура и программное обеспечение совместно обеспечивают выполнение всех шагов алгоритма работы системы. В конце этапа отлаженная программа заносится с помощью

программатора в энергонезависимую память МК, и проверяется работа контроллера без эмулятора. При этом используются лабораторные источники питания. Часть внешних источников сигналов может моделироваться.

Этап интеграции разработанного контроллера в изделие заключается в повторении работ по совместной отладке аппаратуры и управляющей программы, но при работе в составе изделия, питании от штатного источника и с информацией от штатных источников сигналов и датчиков.

Состав и объем испытаний разработанного и изготовленного контроллера зависит от условий его эксплуатации и определяется соответствующими нормативными документами. Проведение испытаний таких функционально сложных изделий, как современные контроллеры, может потребовать разработки специализированных средств контроля состояния изделия во время испытаний.

2. Содержание курсовой работы

1. Разработка функциональной схемы системы (или отдельных блоков) и описание их работы.
2. Выбор типа датчиков, исполнительных устройств. Выбор и расчет основных параметров каналов связи.
3. Обоснование выбора элементарной базы. Выбор типа микропроцессора, микроконтроллера, применяемых в системе. Краткая характеристика используемых интегральных схем и обоснование их выбора.
4. Разработка алгоритма работы системы (модулей системы).
5. Разработка принципиальной схемы системы (модулей системы).

3. Примерные темы курсовых работ:

1. Микропроцессорная система управления серводвигателем.
2. Микропроцессорная система управления трехфазным инвертором напряжения.
3. Микропроцессорное устройство управления радиомодулем.
4. Микропроцессорная система управления шаговым двигателем.
5. Микропроцессорная система управления лифтом.
6. Микропроцессорный конвертер интерфейсов USB-RS232.