

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Воронежский государственный технический университет»

Кафедра автоматизированных и вычислительных систем

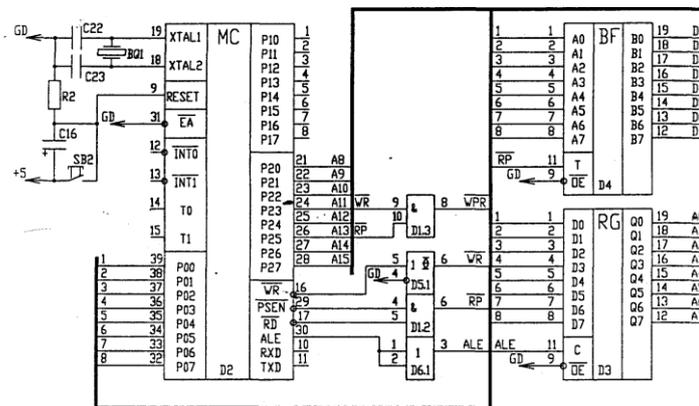
689 - 2021

**ПРОЕКТИРОВАНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ НА ОСНОВЕ
МИКРОКОНТРОЛЛЕРОВ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению курсового проекта
по дисциплине «Микропроцессорные системы»
для студентов направления 09.03.01

«Информатика и вычислительная техника» (профиль
«Вычислительные машины, комплексы, системы и сети»)
очной и заочной форм обучения



Воронеж 2021

УДК 681.32
ББК 30.2

Составитель канд. техн. наук Г.В. Петрухнова

Проектирование микропроцессорных систем на основе микроконтроллеров: методические указания к выполнению курсового проекта по дисциплине «Микропроцессорные системы» для студентов направления 09.03.01 «Информатика и вычислительная техника» (профиль “Вычислительные машины, комплексы, системы и сети”) очной и заочной форм обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. Г.В. Петрухнова. Воронеж, 2021. 26 с.

Методические указания содержат теоретические и практические сведения о проектировании микропроцессорных систем на основе микроконтроллеров и предназначены для студентов четвертого курса.

Методические указания подготовлены в электронном виде и содержатся в файле Проектирование_МПС.doc

Ил. 12. Библиогр.: 18 назв.

УДК 681.32
ББК 30.2

Рецензент – А.В. Бурковский, доц. канд. техн. наук, декан ФЭСУ

*Издается по решению учебно-методического совета
Воронежского государственного технического университета*

1. ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ НА ОСНОВЕ ОДНОКРИСТАЛЬНОГО МИКРОКОНТРОЛЛЕРА

Микропроцессорные системы на основе МК обычно используются для решения задач управления различным оборудованием как встроенные. Важной особенностью данного применения является то, что информация поступает с разных источников, которые необходимо периодически обслуживать в режиме реального времени. Такие устройства чаще всего являются контроллерами.

Технология проектирования МПС на базе МК соответствует принципам неразрывного проектирования и отладки аппаратных и программных средств, принятым в микропроцессорной технике. Независимо от «масштабов» таких систем обобщенный алгоритм их проектирования может быть представлен в соответствии с рис. 1.



Рис. 1. Обобщенный алгоритм проектирования МПС

На разных этапах проектирования микропроцессорную систему рассматривают на соответствующем уровне абстрагирования (т.е. с соответствующей степенью детализации). Возможны такие уровни абстрактного представления как «черный ящик»; структурный (функциональный), программный, логический, схемный.

На уровне «черный ящик» МПС описывается в терминах входных и выходных сигналов управления, информационных потоков.

На структурном (функциональном) уровне МПС представляют взаимосвязанными между собой микропроцессором, запоминающими устройствами, устройствами ввода/вывода, линиями связи и т.п. МПС на этом уровне описывается функциями отдельных устройств и их взаимосвязью.

На программном уровне различают два подуровня: команд процессора и языкового. МПС в этом случае рассматривается как абстрактный автомат и описывается при помощи машинных кодов или последовательностью операторов языка программирования высокого уровня, вызывающих определенное преобразование некоторой структуры данных.

Логический уровень применяется для фрагментов МПС, реализованных на интегральных схемах малой и средней степени интеграции. Для их описания широко используются таблицы истинности, булева алгебра, временные диаграммы.

Схемный уровень характерен для элементов и узлов МПС и позволяет формально описывать их работу на уровне электрических напряжений и токов. Для этой цели используются идеализированные и реальные временные диаграммы, эпюры напряжений, токов и т.п.

В зависимости от сложности проектируемой МПС используют различные уровни ее детализации, а также может меняться объем и содержание блоков обобщенного алгоритма проектирования.

На основе проведенного системного анализа составляется техническое задание, в котором отражается, какие подзадачи необходимо решить, чтобы достичь поставленной цели. В техническом задании также формулируются требования к разрабатываемому контроллеру с точки зрения реализации определенной функции управления, пожеланий пользователя и возможностей разрабатываемой системы. Техническое задание может иметь вид текстового описания.

На основании требований пользователя составляется функциональная спецификация, которая определяет функции, выполняемые системой после завершения проектирования, и тем самым уточняется, насколько система должно соответствовать предъявляемым требованиям. Функциональная спецификация включает в себя описания форматов данных, как на входе, так и на выходе МПС, а также внешние условия, управляющие действиями системы.

Функциональная спецификация и требования пользователя используются для оценки функционирования микропроцессорной системы после завершения проектирования. Требования к типу используемого МК в явном виде формулируются на данном этапе редко.

Разработка алгоритма управления обычно сводится к выбору одного из нескольких возможных вариантов алгоритмов, которые отличаются соотношением объема программного обеспечения и аппаратных средств. Ошибки, допущенные на этом этапе, обнаруживаются чаще всего только при испытаниях законченного

изделия и приводят к необходимости дорогостоящей переработки всего устройства.

Аппаратная часть должна выбираться так, чтобы обеспечить реализацию требуемых алгоритмов управления. При этом необходимо учитывать, что максимальное применение стандартных аппаратных средств упрощает разработку и обеспечивает высокое быстродействие разрабатываемой МПС в целом, но, как правило, приводит к увеличению стоимости и росту энергопотребления, поскольку такой подход предполагает использование сложных МК, специализированных интерфейсных схем.

Увеличение доли программного обеспечения, с одной стороны, позволяет сократить число элементов разрабатываемой микропроцессорной системы и снизить стоимость аппаратных средств, с другой стороны, 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. Это основной критерий, на основе которого осуществляется распределение функций между аппаратными и программными средствами, а также оптимизация состава и структуры аппаратных и программных средств. Часто определяющими требованиями являются возможность защиты информации (программного кода) разрабатываемой МПС, необходимость обеспечения максимальной продолжительности работы в автономном режиме и другие.

При выборе типа МК учитываются следующие основные характеристики:

- разрядность;
- быстродействие;
- набор команд и способов адресации;
- требования к источнику питания и потреблению мощности в различных режимах;
- объем ПЗУ программ и ОЗУ данных;
- возможности подключения внешней памяти программ и данных;
- наличие и возможности встроенных периферийных устройств МК, включая средства поддержки работы в реальном времени (система прерываний, 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.

Перечислить все требования к МК невозможно, поскольку они формируются исходя из специфики проектируемой микропроцессорной системы. Определяю-

щими могут оказаться, например, требования к точности внутреннего компаратора напряжений или наличию определенного интерфейса последовательного ввода/вывода.

Номенклатура выпускаемых в настоящее время МК богата и разнообразна. Современные МК выпускаются, как правило, сериями (семействами). В рамках семейства МК совместимы программно и конструктивно, но различаются по своим возможностям (объем памяти, набор периферийных устройств и т.д.). В основе каждого семейства микроконтроллеров лежит определенное процессорное ядро. Число моделей МК в рамках семейства обычно велико и достигается за счет разнообразия периферии. Все это открывает широкие возможности выбора близкого к оптимальному МК, не имеющего функциональной избыточности.

Однако, для реализации на практике оптимального варианта необходимы глубокая проработка алгоритма управления, оценка объема исполняемой программы и числа линий сопряжения с объектом и т.д. Допущенные на данном этапе ошибки могут впоследствии привести к необходимости смены модели МК и разработке новой печатной платы. В таких условиях целесообразно выполнять предварительное моделирование основных элементов прикладной программы с использованием симулятора выбранного МК.

При отсутствии МК, обеспечивающего требуемые характеристики проектируемой МПС, необходимо вернуться к этапу разработки алгоритма управления, пересмотреть соотношение между объемом программного обеспечения и аппаратных средств, рассмотреть возможности использования внешних схем обрaмления МК.

Разработка аппаратных средств состоит из разработки схемы электрической принципиальной, разводки топологии плат, монтажа макета и его автономной отладки.

При автономной отладке аппаратуры на основе МК с открытой архитектурой проводится контроль состояния многоразрядных шин адреса и данных с целью проверки правильности обращения к внешним ресурсам памяти и периферийным устройствам. При закрытой архитектуре МК реализация большинства функций разрабатываемого устройства осуществляется его внутренними средствами. Поэтому разрабатываемая микропроцессорная система будет включать малое число периферийных ИС, а обмен данными с ними будет идти преимущественно по последовательным интерфейсам. В этом случае первоочередными окажутся задачи согласования по нагрузочной способности параллельных портов МК и отладка алгоритмов обмена по последовательным каналам.

Содержание этапов разработки программного обеспечения, его трансляции и отладки на моделях существенно зависит от используемых системных средств. Задачи предварительной обработки данных часто требуют использования вычислений с плавающей точкой, трансцендентных функций. 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.

2. ЦЕЛИ И СОДЕРЖАНИЕ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Целью курсового проектирования является освоение технологии проектных работ, выбор и обоснование технических решений, развитие навыков самостоятельной работы, а также закрепление и расширение знаний, полученных на лекциях, лабораторных и в ходе самостоятельной работы. При этом в качестве аппаратных средств рекомендуется использование микроконтроллеров.

Курсовой проект должен состоять из описательно-расчетной (а при наличии – и экспериментальной) части, оформленной в виде пояснительной записки. Проект также должен содержать и графическую часть в виде обязательного чертежа функциональной электрической схемы. Пояснительная записка, как правило, состоит из следующих элементов, расположенных в указанной ниже последовательности:

- титульный лист,
- задание на проектирование,
- замечания руководителя,
- содержание,
- введение,
- основная часть,
- заключение,
- библиографический список,
- приложение.

Тема курсового проекта – «Разработка схем и программных модулей на базе микроконтроллеров». **Варианты заданий** выдаются преподавателем индивидуально.

3. ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Примерный объем пояснительной записки составляет 20 – 30 страниц текста. Пояснительная записка подписывается студентом на титульном листе с указанием даты окончания проектирования. Содержание каждой части записки пояснено ниже.

3.1. Титульный лист

Титульный лист является началом пояснительной записки и выполняется согласно стандартам, существующим на момент выполнения курсового проекта.

3.2. Задание на проектирование

Задание на курсовой проект составляется по установленной форме (см. Приложение 1), подписывается руководителем проекта, студентом и помещается за титульным листом. После задания на проектирование должен быть раздел «Замечания руководителя».

3.3. Содержание

Содержание помещается в начале пояснительной записки, за листом замечания руководителя, наглядно характеризует последовательность разделов и их элементов в записке с указанием номеров страниц.

3.4. Введение

Введение к пояснительной записке указывает на общее состояние решаемых в проекте задач, целевое назначение проекта, отражает важность и актуальность темы проекта и указывается метод, положенный в основу решения главной задачи проекта. Объем введения 1-2 страницы.

3.5. Основная часть

В основной части курсового проекта необходимо включить такие разделы как состояние предметной области; методы и средства выполнения курсового проекта; постановка задачи; описание микропроцессорной системы на различных уровнях абстрактного представления («черный ящик»; структурный (функциональный), программный, логический); описание ИС, используемых для разработки схемы электрической функциональной; описание разработанных аппаратных и программных средств.

Основной текст пояснительной записки излагается в строгой логической последовательности, разбивается на разделы, подразделы, пункты.

3.6. Заключение

В выводах должны быть описаны основные результаты выполненной работы и рекомендации по их практическому использованию. В отличие от основной части проекта заключение и введение не нумеруются.

3.7. Библиографический список

В библиографическом списке указывается вся литература, которая была использована в процессе работы над проектом и на которую должны иметься ссылки в тексте записки. Источники следует расположить в порядке появления ссылок в тексте записки.

3.8. Приложение

Приложение оформляется как продолжение пояснительной записки на последующих ее листах. В приложении помещаются справочные материалы. Например, перечень элементов для принципиальной электрической схемы, техническое описание микроконтроллеров и системы их команд, краткие сведения об инструментальной среде программирования, листинг программ и др. Каждое приложение должно начинаться с новой страницы с указанием в верхнем правом углу первого листа слова Приложение, например, «Приложение 1» и иметь тематический заголовок (см. Приложение 1).

4. РАЗРАБОТКА СХЕМЫ ЭЛЕКТРИЧЕСКОЙ ФУНКЦИОНАЛЬНОЙ

На схеме функциональной электрической необходимо изобразить все электрические элементы или устройства, необходимые для осуществления и контроля в изделии заданных процессов, все электрические связи между ними, а также электрические элементы, которыми заканчиваются входные и выходные цепи.

После выбора микроконтроллера и уточнения его пригодности для реализации заданных функций, разработка функциональной электрической схемы может осуществляться параллельно с разработкой алгоритма и программного обеспечения проектируемого устройства. Используя техническую документацию и рекомендуемые схемы включения микроконтроллера, разрабатывается функциональная схема проектируемого устройств. Необходимую техническую документацию для выбора микроконтроллера, разработки схемы, программирования и отладки программ можно найти на сайтах фирм производителей микроконтроллеров и производителей инструментальных систем проектирования.

5. РАЗРАБОТКА И ОТЛАДКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Преобразование схемы алгоритма в исходный текст программы для выбранного микроконтроллера осуществляется достаточно просто с использованием инструментальных средств разработки и отладки программного обеспечения. В настоящее время для программирования и отладки программного обеспечения МК используются большое количество разнообразных инструментальных средств.

В состав современных профессиональных средств написания и отладки программного обеспечения для микроконтроллеров обычно входят эмуляторы процессоров или отладочные платы, тестовый редактор, компиляторы языка высокого уровня (чаще СИ) и ассемблера, редактор связей (компоновщик). Все программные средства обычно объединены интегрированной средой разработки проекта. Если используется отладочная плата, то необходим загрузчик программы в память микроконтроллера, который необязательно входит в интегрированную среду. Двоичный код программы записывается в постоянную память в виде HEX-файла, создаваемого после прохождения этапов трансляции и компоновки исходного текста программы.

Управляющая программа микроконтроллера жестко зависит от схемы разрабатываемого устройства. Невозможно разработать алгоритм, не имея перед глазами схему устройства. При этом не исключены как ошибки монтажа устройства, так и самой его схемы. В процессе отладки программы производится поиск ошибок не только в программном коде, но и в аппаратной части разрабатываемого устройства.

Если МК имеет гарвардскую архитектуру, то память программ и данных физически и логически разделены. Память данных обычно имеет ограниченный объем. Это обстоятельство нужно учитывать при разработке программ для МК. Так, при программировании МК константы лучше хранить не как переменные, а заносить в ПЗУ программ. Прикладные программы должны ориентироваться на работу без использования больших массивов данных.

В памяти данных МК также находится стек. Стековая память обычно имеет ограниченный объем. При разработке программ для компьютеров наполняемость стека контролируется интегрированной средой программирования. Разработчик микроконтроллерных систем должен следить за тем, чтобы стек не переполнялся.

Также следует следить за правильностью указания типов переменных. Если тип переменной указан некорректно, то можно испортить хранимые в памяти данные. Желательно избегать вычислений с плавающей запятой, упрощать задачи деления, умножения и вычитания. Выражения с условиями лучше записывать с указанием равенства или неравенства нулю, поскольку условные переходы, которые получаются при компилировании условных конструкций языка С, выполняются по нулевому или ненулевому результату в аккумуляторе микроконтроллера. Все это позволяет при компилировании получать максимально «быстрый» код.

В настоящее время существуют **2 способа написания программ**: снизу вверх и сверху вниз. При написании программ снизу вверх нельзя приступить к ее отладке, не завершив полностью написания текста всей программы. При этом ошибки в написании блоков программы или непонимание алгоритма работы приводят к тому, что приходится переделывать или выбрасывать полностью отдель-

ные фрагменты программного кода. При таком подходе ошибки могут привести к неработоспособности всего разрабатываемого устройства.

При разработке программы сверху вниз она может быть оттранслирована и выполнена на уровне фрагментов алгоритма, а также можно воспользоваться подпрограммами.

Следует заметить, что алгоритмы программ для МК отличаются от алгоритмов программ для универсальных компьютеров. При выполнении программы на универсальном компьютере ее запуск, взаимодействие с внешними и внутренними устройствами, и с человеком выполняет операционная система. Программа, написанная для МК, решает эти задачи самостоятельно. В компьютере программа в определенный момент времени запускается и завершается. Программа, управляющая МК, запускается при включении устройства и не завершает свою работу, пока не будет выключено питание. Она как бы «наблюдает» за использованием ресурсов микроконтроллерной системы, поэтому ее иногда называют «монитором». Схема алгоритма *программы-монитора* приведена на рис.2.

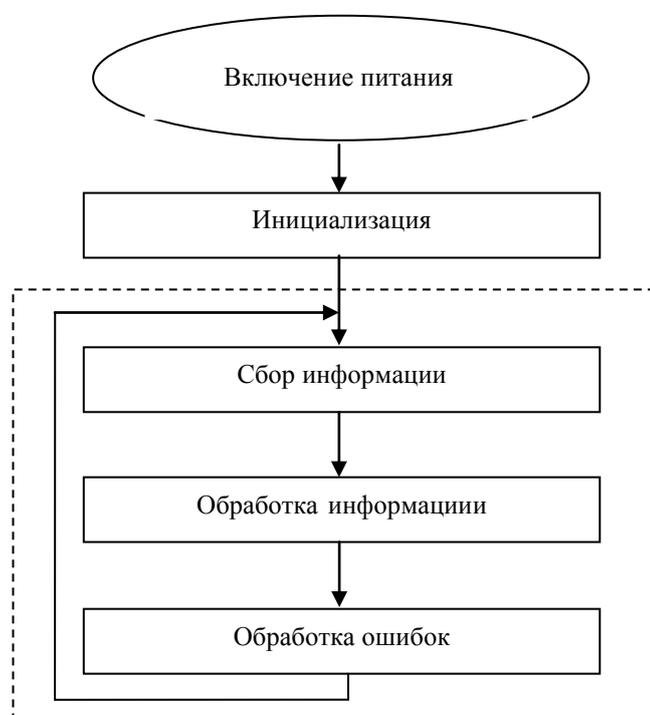


Рис. 2. Схема работа программы–монитора

Основная часть программы МК выполняется после инициализации. При этом важно, чтобы каждая конкретная задача решалась отдельной подпрограммой. Разработка будет эффективной, если вводом информации будет заниматься одна программа, управлять подключенными устройствами - вторая подпрограмма, а общий алгоритм работы отслеживать третья. При этом нельзя допускать ситуации, когда, например, программа ввода данных пытается управлять устройством, или, например, обрабатывать информацию и различные ошибки управления.

Разработанные подпрограммы в соответствии с алгоритмом включают в бесконечный цикл. Обмен данными между подпрограммами осуществляется либо через параметры, либо через глобальные переменные. Блок обработки ошибок

программы предназначен для сообщений оператору о непредвиденных ситуациях, например, некорректный ввод информации с периферийного или управляющего по отношению к МК устройства.

В рассмотренной выше схеме все время процессора принадлежит одной программе-монитору. Однако на практике часто возникают ситуации, когда в микроконтроллер информация поступает с разных источников, которые необходимо периодически обслуживать, исходя из различных заранее заданных интервалов времени. В этом случае время микроконтроллера разбивается на **временные слоты** (интервалы). Наиболее эффективно для этих целей использовать таймеры/счетчики или процессоры событий. Для каждого процесса пишется своя программа-монитор, управление которой осуществляет **диспетчер задач** (см. рис.3).

Использование временных слотов позволяет реализовать устройства, имеющих различное время реакции или опроса. Однако при этом возникают сложности. Часто сигнал на входе МК длится всего несколько микросекунд. В то же время временной интервал между приходами сигналов может быть достаточно большой. Для решения возникших трудностей в МК предусмотрен механизм прерываний основной программы.

Механизм прерываний позволяет обрабатывать короткие сигналы или пакеты сигналов, приходящие в случайные моменты времени. Основное ограничение при использовании прерываний – нужно успеть обработать и запомнить полученную информацию до поступления очередного запроса на прерывание. Поэтому обработку прерываний следует по возможности проводить максимально быстро. Зачастую внутри обработчика прерывания достаточно поднимать флаг о совершении этого прерывания (присвоить некоторой переменной значение логической единицы), а затем внутри основной программы по наличию этого флага выполнить основные действия.

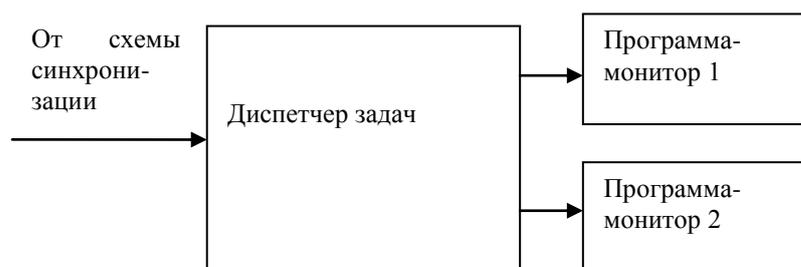


Рис. 3. Схема реализации параллельных программных потоков

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 микроконтроллеров вследствие малого количества ресурсов, страничного распределения памяти, неудобной индексной адресации и некоторых других архитектурных ограничений не обеспечивает компилятору возможность генерировать эффектив-

ный код. Для обхода этих ограничений разработчики ряда компиляторов вынуждены были переложить 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 микроконтроллеров 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 режиме, задавать условные или безусловные точки останова, контролировать и свободно модифицировать содержимое ячеек памяти и регистров симулируемого МК.

Среди основных черт программных средств Keil uVision можно отметить приведенные ниже.

1. Базу данных микроконтроллеров, содержащую подробную информацию обо всех поддерживаемых устройствах. Здесь хранятся их конфигурационные данные и ссылки на источники информации с дополнительными техническими описаниями. При добавлении нового устройства в проект все его уникальные опции устанавливаются автоматически.

2. Менеджер проектов, служащий для объединения отдельных текстов программных модулей и файлов в группы, обрабатываемые по единым правилам. Подобная группировка позволяет лучше ориентироваться среди множества файлов.

3. Встроенный редактор, облегчающий работу с исходным текстом за счет использования многооконного интерфейса, выделения синтаксических элементов шрифтом и цветом. Некоторые среды имеют опции настройки в соответствии со вкусами разработчика. Редактирование остается доступным во время отладки кода, что позволяет сразу исправлять ошибки и отмечать проблемные участки.

4. Средства автоматической компиляции, ассемблирования и компоновки проекта, которые предназначены для создания исполняемого (загрузочного) модуля программы. При этом между файлами автоматически генерируются новые ассемблерные и компиляторные связи, которые в дальнейшем позволяют обрабатывать только те файлы, в которых произошли изменения, или файлы, находящиеся в зависимости от изменённых. Функция глобальной оптимизации проекта позволяет достичь наилучшего использования регистров микроконтроллера путем неоднократной компиляции исходного кода.

5. Отладчик-симулятор, отлаживающий работу скомпилированной программы на виртуальной модели МП. Обычно довольно достоверно моделируется работа ядра МК и его периферийных устройств: портов ввода-вывода, таймеров, системы прерываний. Для облегчения комплексной отладки разрабатываемого ПО возможно подключение программных моделей нестандартного оборудования.

6. Дополнительные утилиты, облегчающие выполнение наиболее распространенных задач. Число и набор меняется от версии к версии.

ОФОРМЛЕНИЕ ЗАДАНИЯ НА КУРСОВОЙ ПРОЕКТ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГТУ»)
Кафедра автоматизированных и вычислительных систем

ЗАДАНИЕ

на курсовой проект

по дисциплине «Микропроцессорные системы»

Тема проекта: «Разработка схем и программ для однокристальных микроконтроллеров»

Студент группы ВМ-121: Торев Игорь Иванович_____.

Вариант: «Разработка схемы электрической принципиальной подключения LCD-дисплея к микроконтроллеру Atmega328p»

Технические условия: LCD-дисплей WH0802A-YGH-CT, среда разработки CodeVisionAVR

Содержание и объем работы: разработка схемы электрической принципиальной подключения к МК Atmega328p LCD-дисплея WH0802A-YGH-CT; написание программы инициализации дисплея; составление расчетно-пояснительной записки.

Сроки выполнения этапов: 1й этап – _____, 2й этап – _____, 3й этап – _____ Срок защиты курсового проекта: _____

Руководитель

Г.В. Петрухнова

Подпись, дата

Инициалы, фамилия

Задание принял студент

И.И. Торев

Подпись, дата

Инициалы, фамилия

ПРИМЕРЫ СХЕМ ПОДКЛЮЧЕНИЯ К КОРПУСУ МИКРОКОНТРОЛЛЕРА
ВНЕШНИХ УСТРОЙСТВ

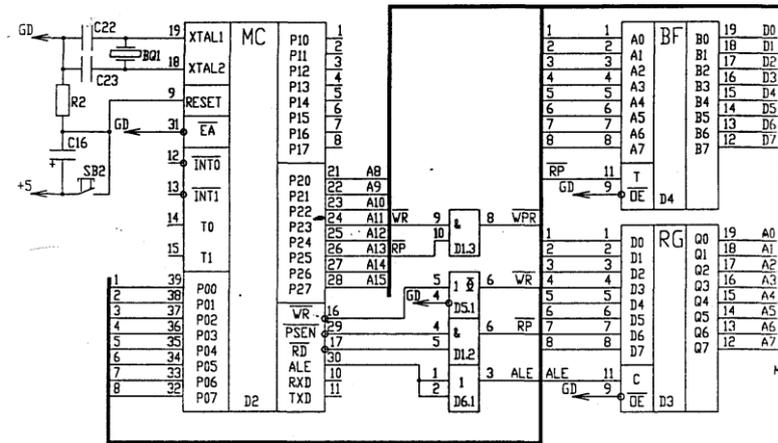


Рис. П2.1. Ядро МПС на базе ОМК ВЕ51

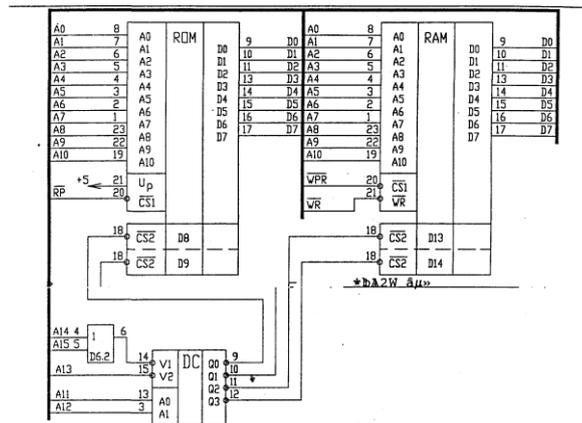
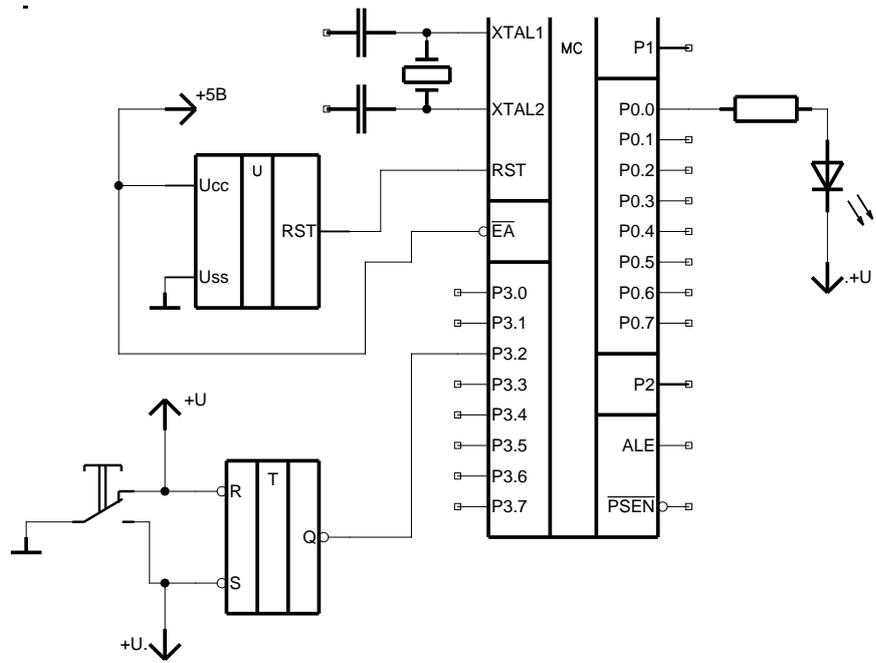


Рис. П2.2. Подключение модулей памяти к ядру МПС на базе ОМК ВЕ51



. Рис. П2.3. Функциональная схема подключения к BE51 кнопки и светодиода

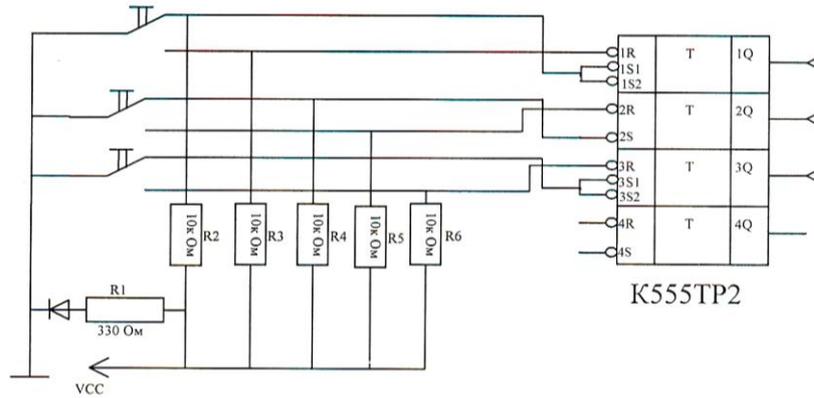


Рис. П2.4. Функциональная схема подключения кнопок с подавлением дребезга

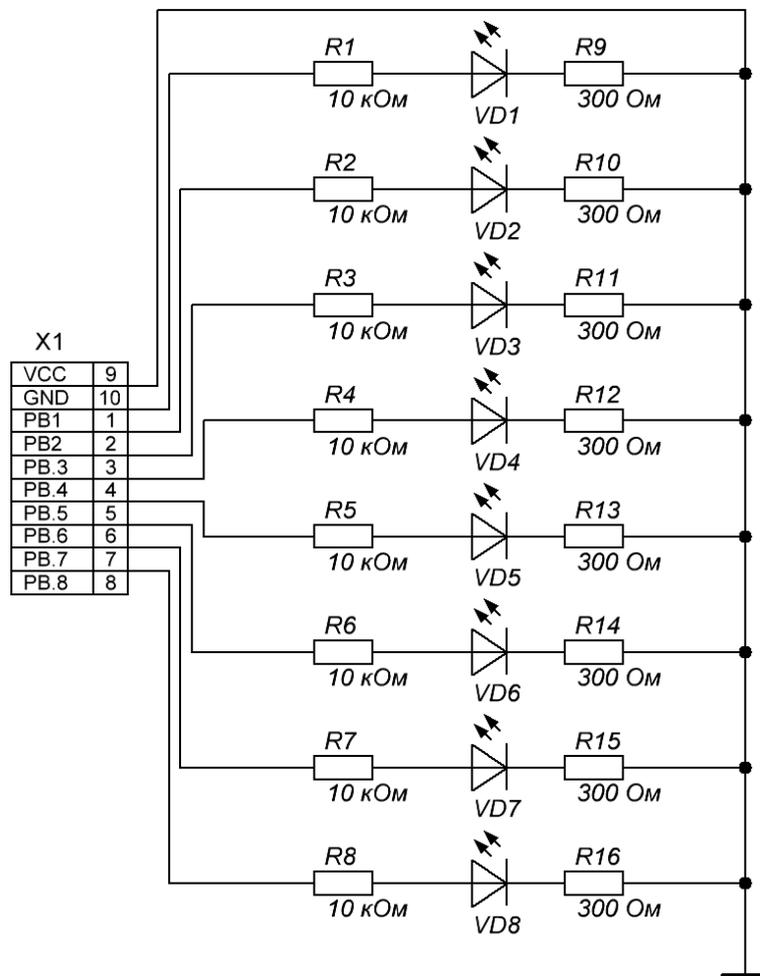


Рис. П2.5. Схема подключения светодиодов к порту МК



Рис. П2.6. Схема подключения тумблера к порту МК

ПОДКЛЮЧЕНИЕ СЕМИСЕГМЕНТНОГО ИНДИКАТОРА К МИКРОКОНТРОЛЛЕРУ

Существуют схемы индикаторов с общим анодом и общим катодом. В светодиодном индикаторе обычно девять выводов. Если один идёт к катодам всех сегментов, а остальные восемь – к аноду каждого из сегментов, то имеем схему с общим катодом. Если все наоборот – это схема с общим анодом. Часто делают не один, а два общих вывода на разных концах цоколя – это упрощает разводку, не увеличивая габаритов.



Рис. ПЗ.1. Схемы питания светодиодов в индикаторе

В том случае, если светодиоды в индикаторе имеют соединённые вместе аноды (схема с общим анодом), общий анод подключается к источнику напряжения $+V_{DD}$, а катоды светодиодов – сегментов подключаются к схеме управления (например, микроконтроллеру), которая отвечает за формирование изображения на индикаторе. Зажигаются сегменты низким уровнем (логический 0) на выводе схемы управления. По отношению к схеме управления ток светодиодов является вытекающим, так что могут использоваться интегральные схемы, которые имеют выходы с открытым стоком. Изменяя величину питающего индикатор напряжения V_{DD} , можно регулировать яркость свечения.

Если в индикаторе соединены вместе катоды (схема с общим катодом), то общий катод подключается к общему проводу схемы, а аноды светодиодов подключаются к схеме управления. В этом случае сегмент зажигается высоким уровнем на выходе схемы управления, для которой ток светодиода является вытекающим, что не позволяет использовать выходы с открытым стоком, необходим выход, выполненный по двухтактной схеме.

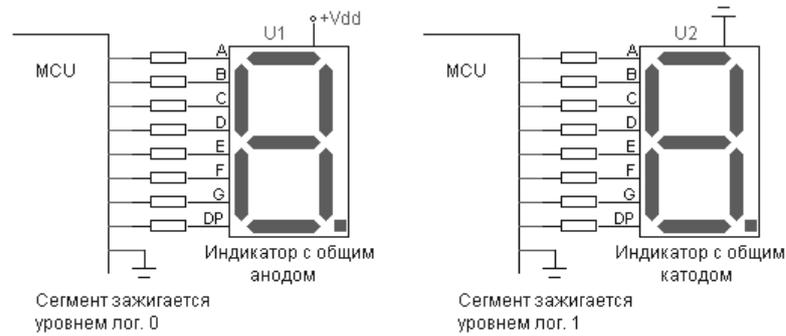


Рис. ПЗ.2. Схемы подключения индикатора к микроконтроллеру

Регулировать яркость можно, подключив общий вывод индикатора к источнику смещающего напряжения $0..V_{DD}$, рассчитанного на втекающий ток, например к эмиттерному повторителю на транзисторе структуры р-п-р. Увеличивая смещение, будем уменьшать яркость свечения.

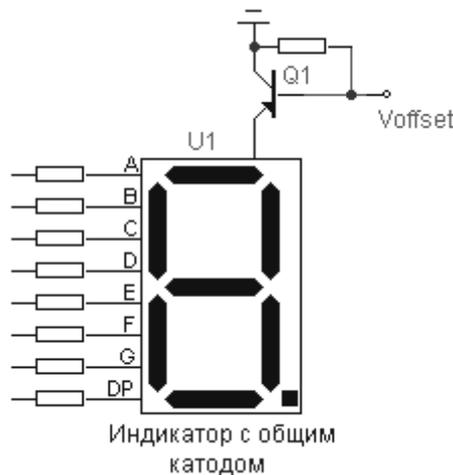


Рис. ПЗ.3. Схемы подключения индикатора к микроконтроллеру с возможностью регулировки яркости свечения

Существует несколько типов семисегментных знакосинтезаторов, для управления которыми используются различные микросхемы преобразователи семисегментного кода.

Для сопряжения сегментных индикаторов с микросхемами ТТЛ-логики используются микросхемы К514ИД1, К514ИД2, К514ИД4.

Они представляют собой преобразователи двоично десятичного кода в код управления семисегментным индикатором. Микросхемы К514ИД1 и К514ИД4 имеют открытые эмиттерные выходы и резисторы в коллекторных цепях и используются для управления индикаторами с общим катодом без внешних резисторов. Микросхема К514ИД2 имеет открытые коллекторные выходы и используется с индикаторами, имеющими общий анод (рис. ПЗ.4). При этом требуется применение внешних резисторов. Допустимый ток микросхемы по каждому выходу – 22 мА.

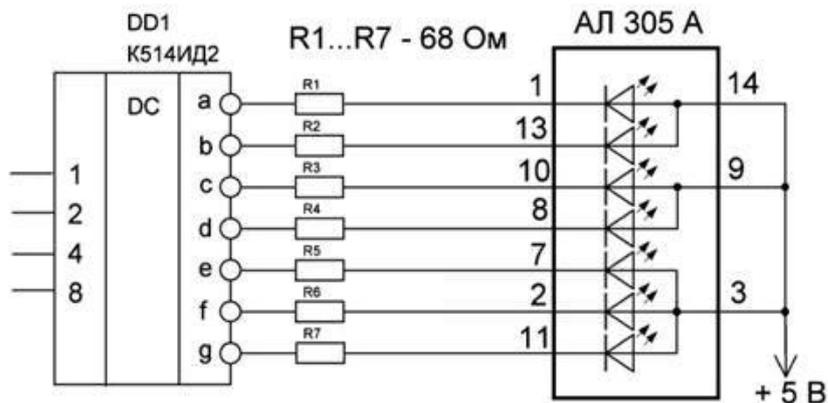


Рис. ПЗ.4. Схемы подключения индикатора к K514ИД2

В КМОП сериях также существуют преобразователи двоично-десятичного кода в код управления семисегментными индикаторами. Микросхема K176ИД2 позволяет непосредственно подключать такие светодиодные индикаторы, как АЛ305А, АЛС321, АЛС324 и др. (рис. ПЗ.5).

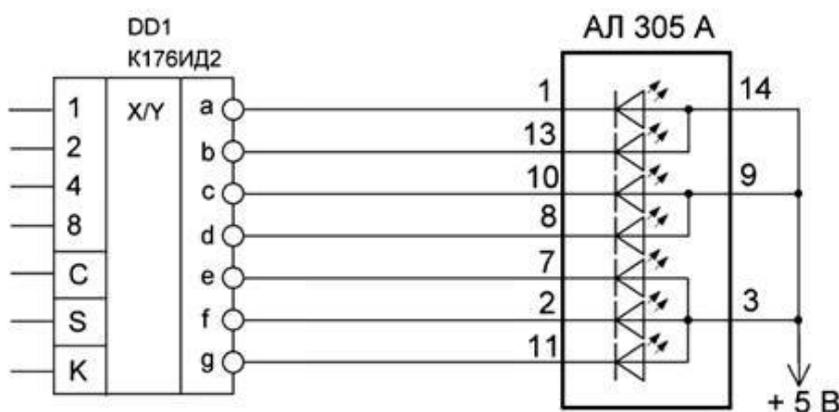


Рис. ПЗ.5. Схемы подключения индикатора к K176ИД2

Управляющий код подается на входы 1-2-4-8. Через вход S можно управлять инверсией выходных сигналов. Вход K используется для разрешения индикации: при подаче напряжения высокого уровня гасится индицируемый знак, при напряжении низкого уровня разрешается индикация. Вход C управляет работой триггеров памяти на входе микросхемы: при подаче напряжения высокого уровня триггеры превращаются в повторители и не влияют на работу, при подаче напряжения низкого уровня сигналы на входах запоминаются, и далее на изменение входных сигналов микросхема не реагирует.

Для управления полупроводниковыми индикаторами в аппаратуре на КМОП интегральных микросхемах могут использоваться также дешифраторы двоичного кода в код управления семисегментным индикатором K564ИД4, K564ИД5 (рис. 4.15). Эти микросхемы предназначены в основном для управления жидкокристаллическими индикаторами, однако могут применяться и с другими типами индикаторов. Микросхема K564ИД5 имеет, в отличие от K564ИД4, входной регистр.

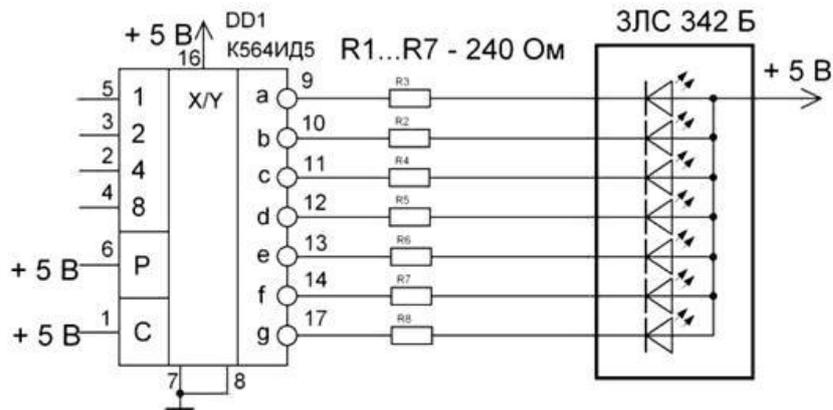


Рис. ПЗ.6. Схемы подключения индикатора к К564ИД5

Ниже приведена схема подключения восьми индикаторов с использованием семи линий порта. Восемь линий порта позволяют подключить до 16 индикаторов (при этом необходимо использовать дешифратор 4/16).

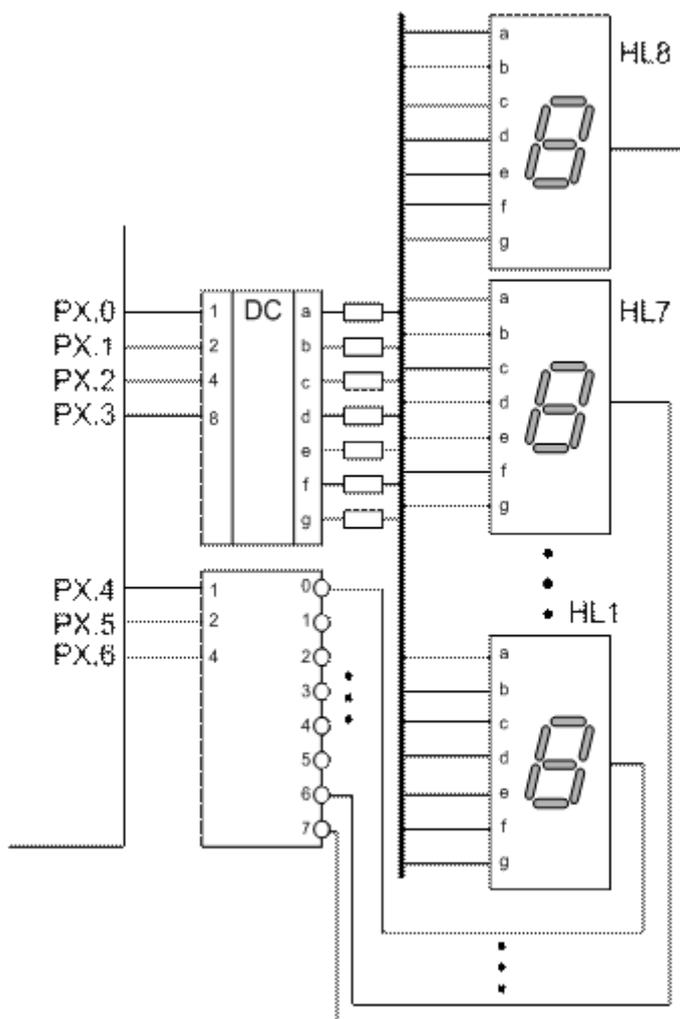


Рис. ПЗ.7. Схемы подключения 8 индикаторов к микроконтроллеру

Стоит отметить, что индикаторы могут подключаться к МК и через другие схемы: буферные регистры, сдвиговые регистры, периферийные адаптеры и др. Ниже приведена схема подключения индикаторов через буферные регистры.

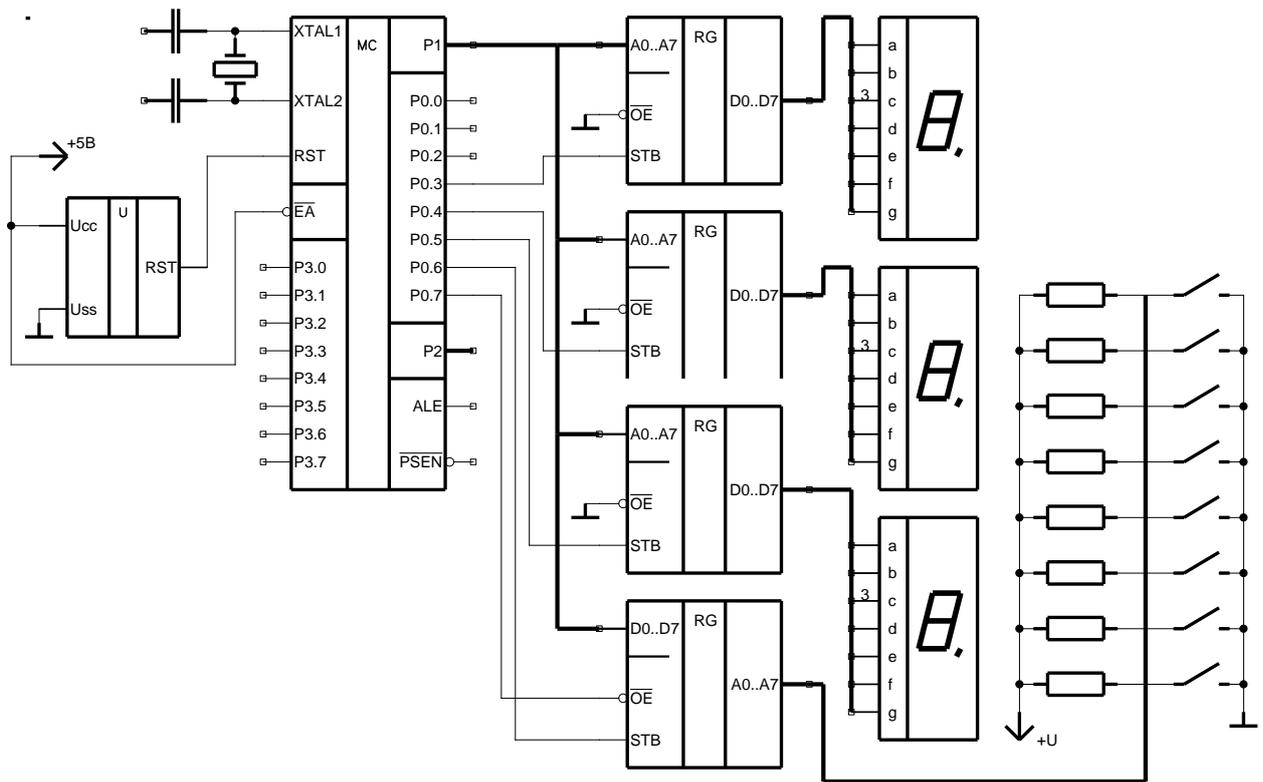


Рис. ПЗ.8. Функциональная схема подключения индикаторов через буферные регистры

Существуют индикаторы со встроенным сдвиговым регистром, благодаря чему намного уменьшается количество задействованных выводов портов микроконтроллера.

Многоразрядные светодиодные индикаторы часто работают по динамическому принципу: выводы одноимённых сегментов всех разрядов соединены вместе. Чтобы вывести информацию на такой индикатор, управляющая микросхема должна циклически подавать ток на общие выводы всех разрядов, в то время как на выводы сегментов ток подаётся в зависимости от того, зажжён ли данный сегмент в данном разряде.



Рис. ПЗ.9. Схема многоразрядного светодиодного индикатора

Динамическая индикация позволяет сократить требуемое для подключения количество выводов. Идея динамической индикации состоит в том, что информация отображается не во всех разрядах индикатора сразу, а поочередно, в каждый момент времени только в одном разряде. В связи с тем, что зрение инерционно, необязательно чтобы все элементы изображения светились непрерывно и одновременно. Если с достаточно высокой частотой последовательно переключаться от отображения одного разряда к следующему, а когда будет достигнут последний разряд индикатора, снова переходить к отображению первого и т. д., то глазом это будет восприниматься так, как если бы каждый разряд отображал информацию статично. Этот метод похож на использовавшийся в кинескопах способ формирования изображения с помощью развёртки.

Так как требуется, чтобы в каждый момент времени работал только один разряд индикатора, то количество выводов можно существенно уменьшить: выводы одноимённых сегментов всех разрядов соединяются вместе, образуя общую шину для управления сегментами. Включение нужного разряда производится с помощью вывода общего анода (или катода, в зависимости от варианта исполнения индикатора) этого разряда.

Как правило, индикаторы, содержащие несколько разрядов, выпускаются именно в расчёте на динамическую индикацию и все необходимые соединения выполнены внутри устройства. N-разрядный индикатор в этом случае имеет 8 выводов для управления сегментами и N выводов для управления включением разрядов (общий анод или катод разряда). Всего требуется $8 + N$ выводов, что намного лучше, чем $8 * N + 1$ при статической индикации.

Для этого в цикле выполняются следующие действия.

1. Гасятся все разряды индикатора – для предотвращения появления артефактов на выводимом изображении при смене состояния шины управления сегментами; если используется схема с общими катодами, для этого на общие катоды всех разрядов подаётся высокий уровень (лог. 1); в схеме с общими анодами, на аноды подаётся лог. 0.

2. На шину управления сегментами выдаются сигналы для отображения символа в очередном разряде.

3. Зажигается очередной разряд.

Либо можно погасить сегменты с помощью шины управления сегментами, переключиться на очередной разряд и выставить на шине сегментов индикатора сигналы для формирования символа. Возможны различные варианты, так что можно выбрать тот, который в данной ситуации проще реализовать.

Затем делается пауза, в течение которой происходит отображение информации на текущем разряде, после чего процесс повторяется. В результате происходит последовательное отображение от первого разряда до последнего, после чего вновь возвращаемся к первому и т. д.

Если разряды переключаются с частотой f , то время отображения одного разряда составит максимум $1 / f$. Максимум – потому что время горения разряда может быть и меньше периода переключения. Можно изменять время горения от 0 до $1 / f$ и тем самым регулировать яркость разряда за счёт эффекта от широтно-импульсной модуляции.

При количестве разрядов N , полное время регенерации изображения на индикаторе в целом составит $N * 1 / f$, соответственно частота регенерации $F = 1 / T = f / N$. Для того чтобы не было заметно мерцания изображения, частота регенерации F должна быть не менее 50 Гц, а лучше не менее 100 Гц.

Следует обратить внимание, что не каждый микроконтроллер сможет обеспечить достаточный ток для управления сегментами, и тем более не каждый сможет непосредственно управлять включением разрядов – ток через общий вывод разряда может превышать ток сегмента в 8 раз, если горят все элементы разряда. Но это не большая проблема – подключить индикатор к микроконтроллеру можно через микросхему-драйвер с мощными выходами или можно использовать ключи на транзисторах.

Вместо обычных повторителей или инверторов для подключения выводов управления разрядами индикатора может использоваться дешифратор $n \times N$ или демультиплексор. Помимо увеличения нагрузочной способности, это даёт возможность ещё уменьшить количество занятых управлением индикатором выводов микроконтроллера. На входы дешифратора подаётся двоичный код, и только на одном выходе, определяемом этим двоичным кодом, будет лог. 1, а на всех остальных будет лог. 0 (или, если выходы инверсные, то наоборот). Дешифратор с трёхбитовым входом имеет до $2^{*3} = 8$ выходов и может использоваться до 8-разрядных индикаторов включительно, а с 4-битовым входом может переключать до 16 разрядов. Демультиплексор осуществляет коммутирование входного сигнала E на один из выходов, задаваемых адресными входами и полностью эквивалентен дешифратору при $E = 1$, а при $E = 0$ на всех выходах будет лог. 0 (или 1, если выходы инверсные).

В некоторых случаях, особенно если количество разрядов у индикатора большое, при подключении удастся обойтись без дополнительных микросхем.

В спецификации на индикатор указывается потребляемый одним сегментом ток. Обычно это величина порядка нескольких мА, и нагрузочной способности выводов большинства микроконтроллеров достаточно для управления индикатором. Если используется микроконтроллер с малым выходным током выходов или если используется индикатор с большим током (например, большого размера или рассчитанный на работу при ярком внешнем освещении), то подключение осуществляется через драйвер – интегральную микросхему, содержащую набор повторителей или инверторов с мощными выходами. Также можно использовать транзисторы в качестве ключей для управления индикатором.

Как и любой светодиод (также это относится и к обычным диодам), светодиоды сегментов имеют очень резкую зависимость тока от напряжения на светодиоде. Поэтому требуется стабилизация тока через эти светодиоды для обеспечения работы в номинальном режиме. Обычно используется простейший способ – последовательное включение задающих ток резисторов. При выборе номинала резисторов следует учитывать падение напряжения на светодиоде в выбранном режиме работы. Эту величину можно уточнить в спецификации на индикатор. Падение напряжения на светодиоде существенно больше, чем на обычном диоде.

Например, для индикаторов FYQ-3641Ax/Bx падение напряжения на светодиоде, в зависимости от материала, цвета свечения, составляет от 1.6 до 2 В при токе 5 мА и от 1.8 до 2.4 В при токе 30 мА (30 мА – максимально допустимый ток

через светодиод для данного индикатора в непрерывном режиме). Возможен разброс значений для разных устройств даже одного типа (в меньшей степени, но есть разброс между характеристиками светодиодов и в пределах одного индикатора). Кроме того, падение напряжения зависит от температуры. Параметры схемы должны обеспечивать достаточную стабильность тока при изменении падения напряжения на светодиоде.

Для случая, когда ток задаётся с помощью резистора, это означает, что падение напряжения на резисторе должно быть намного больше возможных отклонений напряжения на светодиоде от среднего значения. Предположим, что требуется обеспечить ток через светодиод 5 мА, при этом напряжение на светодиоде составляет в среднем 1.8 В. При напряжении источника 3.3 В падение напряжения на резисторе составит $3.3 - 1.8 = 1.5$ В; значит сопротивление резистора $R1 = 1.5 \text{ В} / 5\text{мА} = 300 \text{ Ом}$. Если в результате разброса параметров или в результате изменения температуры, или по иным причинам, возможно отклонение напряжения на светодиоде в пределах 1.6..2.0 В (± 0.2 В от расчётного значения 1.8 В), это вызовет отклонение тока от расчётного значения не более ± 0.7 мА или не более 14 %. В большинстве практических случаев это достаточная точность для питания цепей светодиодных индикаторов, хотя ещё следует учесть нестабильность питающего напряжения, неидеальность цифровых ключей, допуск резистора.

При напряжении источника 5 В падение напряжения на резисторе составит $5 - 1.8 = 3.2$ В; значит сопротивление резистора $R2 = 3.2 \text{ В} / 5 \text{ мА} = 640 \text{ Ом}$, выбираем 620 Ом – ближайшее значение из ряда E24. В этом случае отклонение напряжения на светодиоде ± 0.2 В вызовет отклонение тока от расчётной величины порядка ± 0.3 мА или не более чем ± 7 %. Получили точность заданного тока лучшую, чем в первом случае. Это вполне ожидаемый результат – увеличение напряжения источника и его сопротивление позволяет сделать его более близким к идеальному источнику тока.

Если задаться предельно допустимой точностью тока ± 20 %, можем получить, что минимальное питающее напряжение составляет 2.8 В, при этом сопротивление ограничивающего ток резистора равно 200 Ом.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Микропроцессорные системы / Е.К. Александров, Р.И. Грушвицкий, М.С. Куприянов, О.Е. Мартынов и др.; под общ. ред. Д.В. Пузанкова. - СПб.: Политехника, 2012.
2. Магда Ю.С. Микроконтроллеры серии 8051: практический подход / Ю.С. Магда. - М.: ДМК Пресс, 2008.
3. Микушин А.В. Занимательно о микроконтроллерах / А.В. Микушин. - СПб.: БХВ-Петербург, 2006.
4. Яцуков В.С. Микроконтроллеры MicroCHIP. Практическое руководство / В.С. Яцуков. - М.: Горячая линия, Телеком, 2002.
5. Редькин П.П. 32/16-битные микроконтроллеры семейства ARM7AT91SAM7 фирмы Atmel. Руководство пользователя / П.П. Редькин. - М.: Додека-XXI, 2008.
6. Тюрин, С.В. Элементы проектирования микропроцессорных устройств и систем / С.В. Тюрин, В.А. Кондусов. - Воронеж: ВГТУ, 2006.
7. Петрухнова Г.В. Архитектура и эволюция микропроцессоров / Г.В. Петрухнова. - Воронеж, ВГТУ, 2011.
8. Петрухнова Г.В. Однокристальные микроконтроллеры семейства VE51 / Г.В. Петрухнова. - Воронеж, ВГТУ, 2010.
9. Баранов В.Н. Применение микроконтроллеров AVR: схемы, алгоритмы, программы / В.Н. Баранов - М.: «Додэка -XXI», 2010.
10. Естифеев А.В. Микроконтроллеры семейств Tiny и Mega фирмы Atmel / А.В. Естифеев. - М.: «Додэка -XXI», 2010.
11. Белов А.В. Самоучитель разработчика устройств на микроконтроллерах AVR / А.В. Белов. - СПб: Наука и техника, 2008.
12. Сажнев А.М. Цифровые устройства и микропроцессоры / А.М. Сажнев, А.В. Микушин. – СПб.: БХВ-Петербург, 2010.
13. Новиков Ю. Основы микропроцессорной техники [Электронный ресурс] / Ю. Новиков, П. Скоробогатов. – Режим доступа: <http://www.intuit.ru/studies/courses/3/3/info>

СОДЕРЖАНИЕ

1. ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ МИКРОПРОЦЕССОРНОЙ СИСТЕМЫ НА ОСНОВЕ ОДНОКРИСТАЛЬНОГО МИКРОКОНТРОЛЛЕРА	1
2. ЦЕЛИ И СОДЕРЖАНИЕ КУРСОВОГО ПРОЕКТИРОВАНИЯ	5
3. ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.....	6
4. РАЗРАБОТКА СХЕМЫ ЭЛЕКТРИЧЕСКОЙ ФУНКЦИОНАЛЬНОЙ..	8
5. . РАЗРАБОТКА И ОТЛАДКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ...	9
ПРИЛОЖЕНИЕ 1.....	13
ПРИЛОЖЕНИЕ 2.....	14
ПРИЛОЖЕНИЕ 3.....	18
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	26

ПРОЕКТИРОВАНИЕ МИКРОПРОЦЕССОРНЫХ СИСТЕМ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРОВ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению курсового проекта
по дисциплине «Микропроцессорные системы»
для студентов направления 09.03.01
«Информатика и вычислительная техника» (профиль
«Вычислительные машины, комплексы, системы и сети»)
очной и заочной форм обучения

Составитель

Галина Викторовна Петрухнова

В авторской редакции

Подписано к изданию .11.2021
Уч.-изд. л. 1,6

ФГБОУ ВО «Воронежский государственный технический
университет»
394026 Воронеж, Московский просп., 14