**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ**

В.В. Русанов, М.Ю. Шевелёв

МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА И СИСТЕМЫ

Руководство к организации самостоятельной работы

2012

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Кафедра Промышленной электроники

В.В. Русанов, М.Ю. Шевелёв

МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА И СИСТЕМЫ» (МПУиС)

Руководство к организации самостоятельной работы для студентов направления 210100 «Электроника и наноэлектроника»

2012

Русанов В.В., Шевелёв М.Ю.

Микропроцессорные устройства и системы: Руководство к организации самостоятельной работы. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2012. – 90 с.

Пособие предназначено для организации самостоятельной работы студентов специальности «Промышленная электроника» по дисциплине «Микропроцессорные устройства и системы». В пособии приведена рабочая программа курса, варианты индивидуальных, контрольных и творческих заданий, а также большое число примеров выполнения типовых электронных блоков. Значительная часть пособия посвящена выполнению курсового проекта по дисциплине. Приводится пример выполнения типового проекта, а также ссылки на различные ГОСТы, которыми необходимо руководствоваться при работе.

Оглавление

1 Введение

Данное руководство предназначено для помощи студентам в организации самостоятельной работы по курсу «Микропроцессорные устройства и системы». Руководство включает рабочую программу дисциплины, примерные варианты индивидуальных, творческих заданий и контрольных работ.

В процессе работы студентам приходится много заниматься поиском технической литературы для выполнения различных заданий. В настоящее время практически неисчерпаемым источником информации является глобальная сеть Internet. Но найти требуемую информацию зачастую достаточно сложно ввиду того, что часто трудно ручаться за ее достоверность. С целью упрощения поиска достоверной литературы в руководство включен раздел «Примеры выполнения типовых блоков заданий». В данном разделе детально рассмотрены примеры некоторых технических решений реализации блоков заданий, названных типовыми блоками. Такое название они получили вследствие того, что являются неотъемлемыми частями практически любого микропроцессорного устройства. Эта информация авторами взята из различных источников, переработана, проверена на практике и представлена в наиболее простом и понятном виде.

Значительную часть руководства занимает раздел «Методические указания к выполнению курсового проекта» по данной дисциплине. Курсовой проект является завершением курса и предполагает проектирование какого-либо цифрового устройства, содержащего однокристальный микроконтроллер. Для успешного выполнения курсового проекта студенты должны применить на практике все знания, полученные при изучении самой дисциплины. В разделе приведены общие рекомендации по выполнению проекта, ссылки на ГОСТы, которыми студенты должны руководствоваться для технически грамотного оформления графических и текстовых работ. В конце раздела

приведен детальный пример проектирования микропроцессорного устройства.

В свете того, что ТУСУР имеет инновационную направленность, студентам рекомендуется выполнять курсовой проект не только в теории, на бумаге, но и на практике. От того, какие практические навыки получит студент, будут зависеть успехи в его будущей профессиональной деятельности. Для практической работы над курсовыми проектами на кафедре промышленной электроники ТУСУР созданы все условия. Это и научные лаборатории, и СКБ «Импульс» и лаборатории группового проектного обучения, оснащенные самым современным лабораторным и промышленным оборудованием.

2 ПРИМЕРЫ ВЫПОЛНЕНИЯ ТИПОВЫХ БЛОКОВ ЗАДАНИЙ

В данном разделе приводятся частные решения некоторых типовых электронных блоков, которые будут полезны студентам как при выполнении индивидуальных, творческих заданий, так и при выполнении курсового проекта.

2.1. Сопряжение микроконтроллера с семисегментными светодиодными индикаторами.

Семисегментные светодиодные индикаторы представляют собой очень простое и недорогое решение задачи отображения числовой информации в различных промышленных и бытовых устройствах. Достоинства: высокая контрастность, т.е. хорошая различимость информации в любое время суток, стабильная работа в широком температурном диапазоне.

Типичным представителем семисегментного светодиодного индикатора является отечественный индикатор АЛС324 [7]. Он состоит из восьми светодиодов, соединенных по схеме с общим анодом или общим катодом. Каждый светодиод, в зависимости от типа индикатора, потребляет ток от 10 до 20 мА. Если индикатор выполнен по схеме с общим анодом, то необходимо этот общий вывод подключить к цепи «+5В», а остальные выводы индикатора (вторые выводы светодиодов) подключить к управляющему устройству. В качестве такого устройства может выступать специализированный дешифратор, например К514ИД1, который входной двоичный код преобразует в код семисегментного индикатора. Этот дешифратор имеет встроенные резисторы, которые ограничивают ток, протекающий через светодиоды. Кроме того, индикатор можно подключить через токоограничительные резисторы напрямую к портам микроконтроллера. Но в этом случае необходимо учитывать нагрузочную способность портов микроконтроллера.

Например, максимальный ток логического «нуля» каждого вывода порта P0 микроконтроллера AT89S51 фирмы Atmel составляет 17 мА, а суммарная нагрузка на порт P0 не должна превышать 26 мА. Остальные порты имеют более низкую нагрузочную способность [11]. Поэтому, как правило, светодиоды не подключают к портам этого микроконтроллера напрямую.

Рассмотрим другой пример. Микроконтроллер ATMega16 той же фирмы для корпуса PDIP имеет нагрузочную способность каждого отдельного вывода порта около 20 мА, суммарная нагрузка на порт *A* не должна превышать 100 мА, суммарная нагрузка портов *B*, *C* и *D* не должна превышать 100 мА, а суммарная нагрузка на все порты не должна превышать 200 мА [12]. Из этого следует, что к данному микроконтроллеру можно подключать напрямую светодиоды (через токоограничительные резисторы) или другую нагрузку, учитывая вышеуказанные ограничения.

Рассмотрим варианты статической и динамической индикации.

При статической индикации информация выводится одновременно на все индикаторы через дешифраторы (рис. 2.1). Задача программиста состоит в том, чтобы представить каждое число, выводимое на индикаторы, в двоично-десятичном коде, так как каждое число выводится на индикаторы как отдельная тетрада: старшая или младшая.

Такая реализация индикации очень проста с программной точки зрения, но достаточно затратная с аппаратной стороны, так как для каждого индикатора необходим свой дешифратор. Недостатком такой реализации является и то, что число индикаторов зависит от числа портов микроконтроллера. Кроме того, необходимо учитывать, что к портам микроконтроллера могут быть подключены дополнительные периферийные устройства, а не только индикаторы.

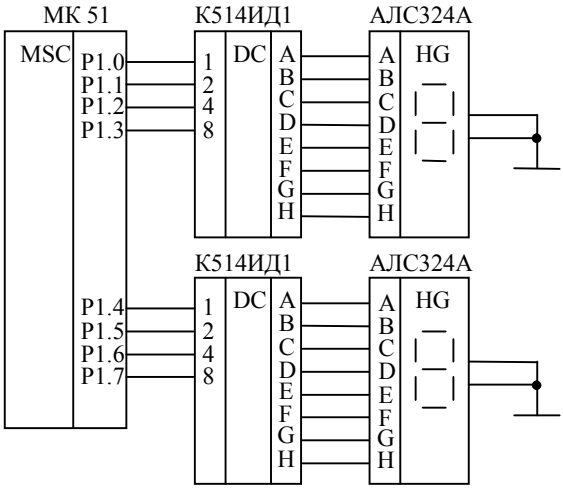


Рисунок 2.1 – Пример реализации статического варианта индикации.

При динамической индикации (рис. 2.2) аппаратная часть проще, но программная часть сложнее по сравнению со статической индикацией. В данном варианте используется только один дешифратор, к которому параллельно подсоединены все индикаторы. Входы разрешения каждого индикатора подключаются к свободным линиям портов микроконтроллера. Под входом разрешения индикатора в данном случае понимается вывод, соединяющий все одноименные выводы светодиодов. Если индикатор выполнен по схеме с общим катодом, то, чтобы «зажечь» хотя бы один сегмент, необходимо на вход разрешения (этот общий вывод) подать уровень логического «нуля», а на требуемый сегмент – логическую «единицу». Так как порт микроконтроллера не всегда может обеспечить втекающий ток в несколько десятков миллиампер (эта ситуация может возникнуть, когда необходимо зажечь несколько сегментов одного индикатора), то необходимо поставить какой-либо токовый усилитель. В схеме на рис. 2.2. эту роль играют инверторы, обладающие повышенной нагрузочной способностью.

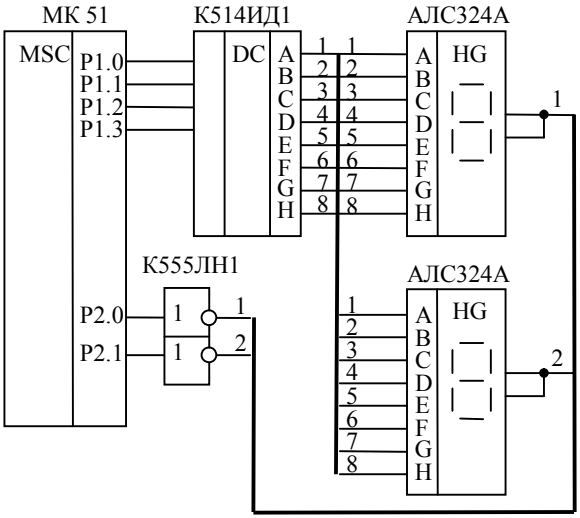


Рисунок 2.2 – Пример реализации динамического варианта индикации.

Вывод информации на индикаторы производится по следующему алгоритму. На входы разрешения всех индикаторов через усилители подается сигнал, запрещающий их работу. Другими словами, все индикаторы «гасятся». Из порта микроконтроллера на дешифратор подается цифра (например, младший разряд числа), которую необходимо вывести на первый индикатор, а на вход разрешения этого индикатора подается разрешающий сигнал. На индикаторе зажигается требуемая цифра. Через несколько миллисекунд на входы разрешения опять подается запрещающий сигнал (который гасит этот индикатор) и на дешифратор выдается другая цифра (следующий разряд числа), а на вход разрешения второго индикатора выдается разрешающий уровень напряжения. Эта цифра загорается на несколько миллисекунд на втором индикаторе. Эта последовательность действий повторяется в цикле. На индикаторах с частотой в несколько кГц (в зависимости от алгоритма работы программы) будет зажигаться

и гаснуть число. Для человеческого глаза такая частота неразличима, поэтому число будет казаться горящим постоянно.

Рассмотрим достоинства и недостатки каждого типа индикации.

Достоинством статического варианта индикации является простота написания программы для вывода информации на индикаторы.

Недостатки статического варианта индикации:

- для индикации выделяется слишком большое число линий порта. Большинство современных МК имеют небольшое число линий порта, а так как каждый индикатор требует по 4 линии порта, то можно подключить очень мало индикаторов;
- увеличиваются геометрические размеры устройства, так как на печатной плате устройства необходимо разместить большее число корпусов микросхем по сравнению с вариантом динамической индикации;
- увеличивается стоимость изделия на дополнительную площадь печатной платы и стоимость дополнительных дешифраторов;
- увеличивается стоимость изделия за счет увеличения числа паек дополнительных дешифраторов;
- увеличивается время монтажа устройства из-за пайки дешифраторов;
- уменьшается коэффициент надежности работы всего устройства, так как общая надежность формируется из надежности отдельных компонентов.

2.2. Сопряжение микроконтроллера с алфавитно-цифровым жидкокристаллическим дисплеем.

Алфавитно-цифровые (или знакогенерирующие) жидкокристаллические индикаторы (ЖКИ) – это сложные цифровые устройства, которые к настоящему времени практически вытеснили семисегментные светодиодные индикаторы почти из всех сфер применения.

Их достоинства:

- низкое потребление (5-15 мА);
- простота подключения к микроконтроллерам;
- небольшие габаритные размеры и вес;
- отображение большого объема информации, включая не только цифры и буквы, но и специальные знаки;
- возможность создания своих собственных символов (до 8);
- унифицированный интерфейс для различных индикаторов.

Это позволяет один раз создать подпрограмму работы с ЖКИ, а затем пользоваться ею для индикаторов различных геометрических размеров с разным количеством строк и знаков в строке с минимальными изменениями исходного кода подпрограммы.

Алгоритм работы с ЖКИ подробно описан в [1], поэтому здесь на нем не будем останавливаться.

Рассмотрим лишь те вопросы, которые касаются конкретной схемотехники и некоторые практические рекомендации при работе с ЖКИ.

На рисунках 2.3 и 2.4 приведены два варианта подключения контроллера индикатора к микроконтроллеру: по 8- и 4-х разрядной линиям.

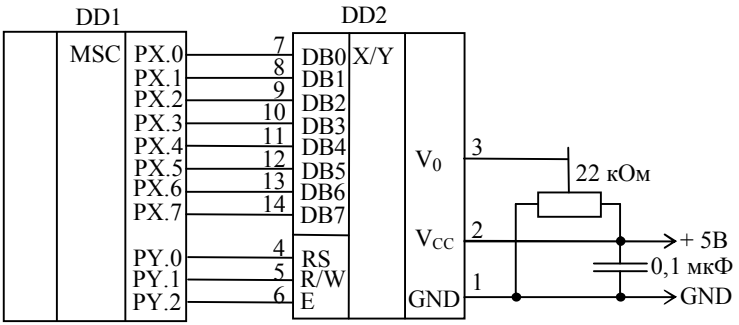


Рисунок 2.3 – Вариант подключения ЖКИ к управляющему микроконтроллеру по 8-ми проводной линии связи.

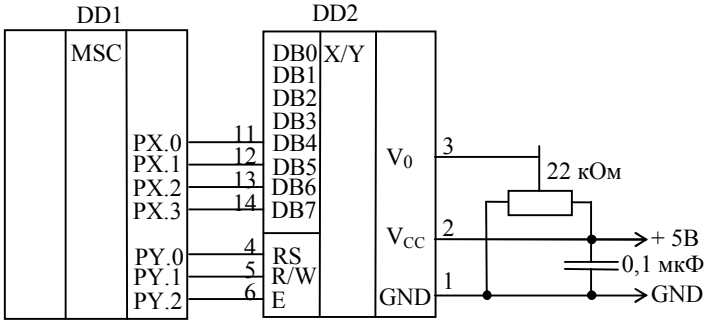


Рисунок 2.4 – Вариант подключения ЖКИ к управляющему микроконтроллеру по 4-х проводной линии связи.

Несмотря на наличие четырехразрядного режима, семь необходимых для связи линий (4 информационные и 3 управляющие) могут оказаться чрезмерным требованием для приборов, которые в современных условиях нередко строятся с применением микро-ЭВМ в корпусах с очень малым числом выводов и имеющих ограниченный ресурс свободных портов. Оказалось, что часто наиболее удобным способом подключения ЖКИ-модуля становится использование отдельного контроллера на базе конкретной микро-ЭВМ с последовательным интерфейсом, осуществляющего посредничество между управляющей системой и ЖКИ-модулем. Одним из таких примеров является контроллер CE110. Этот контроллер подключается к управляющему микроконтроллеру посредством интерфейса I^2C, а к ЖКИ – с использованием стандартного параллельного интерфейса ЖКИ.

Последовательности действий, которые необходимо выполнять управляющей системе при совершении операций записи и чтения для 8-ми и 4-х разрядной шины приведены в [1]. Кроме того, в руководстве по лабораторному циклу приводятся на языке ассемблера типовые процедуры записи данных в регистр данных и регистр команд контроллера ЖКИ.

Несколько слов о практическом использовании алфавитно-цифрового индикатора в системах, где он должен играть роль не просто как средство отображения данных, а для отображения ввода данных с клавиатуры, особенно, если объем вводимой с клавиатуры информации может превышать длину строки дисплея. Т.е. когда индикатор используется для реализации простейшего текстового редактора. Это могут быть электронные карманные словари, электронные записные книжки, какие-либо удаленные терминалы для ввода информации и пр.

Как говорится в [1], ЖКИ имеет режим сдвига содержимого экрана, причем можно выбрать направление. Особенность заключается в том, что при применении двустрочных индикаторов одновременно будут сдвигаться обе строки. Точнее сами данные, записанные по конкретным адресам, будут оставаться на месте, а будет изменяться адрес, с которого будут отображаться данные на самом дисплее. Например, если на первой строке дисплея с организацией 2x16 (две строки по 16 знаков каждая) с нулевого адрес находится надпись «Введите данные», а с начала второй строки (с адреса 40H) производится ввод данных с клавиатуры, то логичным было бы при вводе 17 знака сдвинуть содержимое дисплея на один знак вправо., чтобы увидеть этот знак. Но в этом случае произойдет сдвиг не только той строки, что была введена, но и верхней строки. Это изображено на рисунке 2.5. Жирной рамкой обведена та область адресов, что является видимой. Эта рамка, условно говоря, может перемещаться вправо и влево, используя внутренний счетчик перемещения, значение которого скрыто от пользователя.

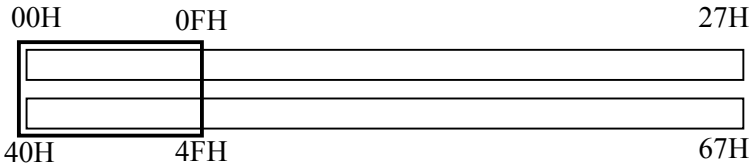


Рисунок 2.5 – Схематичное изображение видимой части ЖКИ

Выходом из такой ситуации может являться отказ от режима смещения содержимого экрана и создание программы, которая в оперативной памяти управляющего микроконтроллера будет эмулировать область памяти контроллера ЖКИ. Эта программа должна автоматически при необходимости сдвигать одну строку в нужном направлении. Достигается это процедурой, которая будет смещать адреса символов в нужную сторону, переписывая их по-очереди в соседние ячейки памяти. И после любого изменения информации в ОЗУ микроконтроллера необходимо просто переписать всю эту область памяти в память ЖКИ. Таким способом можно реализовать любые возможности отображения при текстовом вводе информации.

2.3. Вариант программной реализации матричной клавиатуры 4x4 клавиши.

Матричная клавиатура представляет собой набор горизонтально и вертикально расположенных проводников, на пересечении которых расположены кнопки (рис. 2.6, *а*). Такая структура позволяет значительно сэкономить число линий портов микроконтроллера по сравнению с линейной организацией (рис. 2.6, *б*). Например, в представленном на рисунке варианте для матричной клавиатуры для девяти кнопок необходимо всего 6 линий порта, когда для линейной клавиатуры необходимо 9 линий.

Работа с матричной клавиатурой для любого микроконтроллера производится по следующему алгоритму: порт, отвечающий за строки, настраивается на ввод (для микроконтроллера семейства МК51 в порт необходимо записать все «единицы», а для AVR-микроконтроллера нужно установить в «нуль» биты регистра DDRx, который отвечает за направление). Порт, отвечающий за столбцы, настраивается на вывод. В столбцы по-очереди выдается «бегущий нуль», а затем производится чтение строк. Если в какой-либо строке обнаружен

«нуль», следовательно, была нажата какая-то клавиша. Зная номер столбца, в который был выдан «бегущий нуль» и номер строки, в которой обнаружен «нуль» при чтении, можно однозначно определить, какая именно была нажата клавиша.

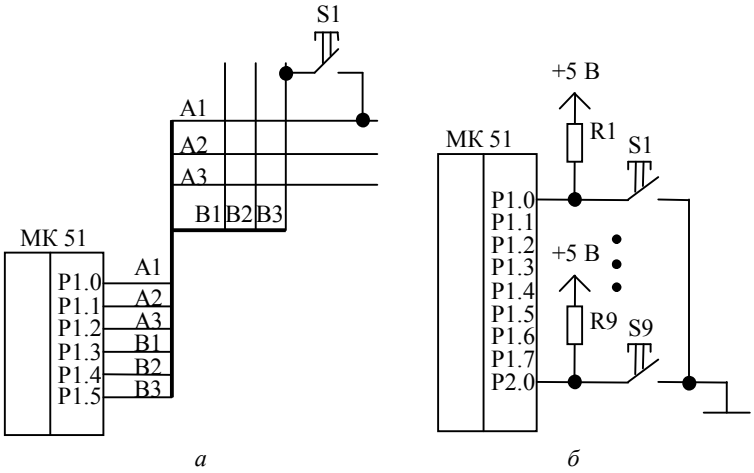


Рисунок 2.6 Варианты реализации клавиатуры.

Работа с линейной клавиатурой производится по тому же принципу, только число строк здесь равно числу кнопок, а столбец один и он всегда подключен к общей шине. При нажатии какой-либо клавиши на соответствующей линии порта появляется логический «нуль».

При реализации любого типа клавиатуры рекомендуется линии портов, работающих на ввод информации, присоединить к шине питания через резисторы номиналом 10 – 100 кОм, как это показано на рис. 2.6 *б*. Если этого не сделать, то при работе рядом источника сильного электромагнитного излучения на входные линии порта могут наводиться помехи и состояние порта может быть неправильно прочитано микроконтроллером.

Очень важной особенностью алгоритма опроса клавиатуры является устранение явления «дребезга контактов». Это явление заключается в том, что процесс замыкания кнопки не

происходит мгновенно, а сопровождается множественными микропробоями между контактами кнопки при ее нажатии и отпускании. Поэтому, если не устранять это явление, то вместо однократного нажатия будет происходить множественное нажатие, если опрос клавиатуры производится циклически.

Алгоритмов программного устранения дребезга контактов существует несколько. Одним из простейших алгоритмов является формирование небольшой временной задержки после первого зафиксированного нажатия. В этом случае, даже если и происходит явление дребезга, то оно не оказывает влияния на программу.

Одним из наиболее эффективных вариантов является алгоритм многократного циклического опроса клавиатуры. Его суть состоит в следующем: процедура сканирования клавиатуры, т.е. проверка на нажатие какой-либо клавиши, производится в бесконечном цикле. Клавиша считается нажатой, если было зафиксировано ее нажатие подряд в течение нескольких циклов опроса клавиатуры. Раскроем алгоритм подробнее. Если было зафиксировано нажатие клавиши, то производится запоминание ее скан-кода, счетчик числа зафиксированных нажатий увеличивается на единицу и производится повторное сканирование клавиатуры. Если при повторном сканировании ни одна клавиша не была нажата, т.е. произошло явление «дребезга», то счетчик числа зафиксированных нажатий обнуляется. Если при следующих сканированиях клавиатуры снова было зафиксировано нажатие клавиши и скан-код текущей нажатой клавиши был равен сохраненному скан-коду предыдущего зафиксированного нажатия, то счетчик числа зафиксированных нажатий увеличивается на единицу. Если же скан-коды не совпали, то, значит, были подряд нажаты две разные клавиши. В этом случае счетчик числа зафиксированных нажатий или сбрасывается, или просто не изменяет своего значения (это зависит от требуемой задачи). Как только счетчик числа зафиксированных нажатий

достигнет требуемого значения, то клавиша считается нажатой. Такую же процедуру необходимо произвести и для фиксации того, что клавиша была отпущена. Для этого при сканировании фиксируется число циклов сканирования, в течение которых данная клавиша не была нажата. Как только это число достигнет требуемого значения, то клавиша будет считаться отпущенной. Эту процедуру фиксации отпускания клавиши необходимо делать потому, что в противном случае будет неясно, что одну и ту же клавишу нажали несколько раз подряд.

Необходимо заметить, что рассматриваемый алгоритм справедлив только для фиксации однократного нажатия клавиши и непригоден в чистом виде для реализации функции автоповтора нажатия при длительном удержании клавиши.

Все рассмотренные выше варианты являются чисто программными. Существует большое количество специализированных микросхем – контроллеров клавиатуры, которые, если нет жесткой экономии каждой копейки при разработке устройства, позволяют значительно сократить время разработки нового устройства. Кроме того, сама клавиатура требует значительного числа линий портов. Специализированные же микросхемы, как правило, имеют последовательный интерфейс для связи с микроконтроллером, что позволяет сэкономить несколько линий портов и использовать их для каких-либо других целей.

2.4. Вариант сопряжения микроконтроллера с персональным компьютером по последовательному порту.

Во многих задачах управление технологическим процессом осуществляется из персонального компьютера, который является «мозгом» системы управления и дает команды исполнительным устройствам через стандартные интерфейсы. В большинстве случаев сопряжение персонального компьютера с исполнительными устройствами осуществляется по последовательному интерфейсу. Как правило, основой

исполнительного устройства является однокристальный микроконтроллер. Рассмотрим сначала принцип и типы передачи, а затем наиболее распространенный вариант сопряжения микроконтроллера и персонального компьютера.

2.4.1 Общие сведения о стандарте RS-232C.

Интерфейс RS-232C предназначен для подключения аппаратуры, передающей или принимающей данные (ООД — оконечное оборудование данных, или АПД — аппаратура передачи данных; DTE — Data Terminal Equipment), к оконечной аппаратуре каналов данных (АКД; DCE — Data Communication Equipment). В роли АПД может выступать компьютер, принтер, плоттер и другое периферийное оборудование. В роли АКД обычно выступает модем. Конечной целью подключения является соединение двух устройств АПД. Полная схема соединения приведена на рис. 2.7; интерфейс позволяет исключить канал удаленной связи вместе с парой устройств АКД, соединив устройства непосредственно с помощью нуль-модемного кабеля.

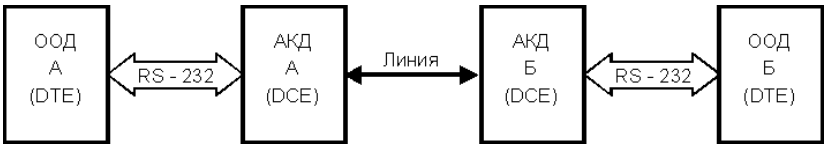


Рисунок 2.7 – Полная схема соединения по RS232.

Стандарт описывает управляющие сигналы интерфейса, пересылку данных, электрический интерфейс и типы разъемов. В стандарте предусмотрены асинхронный и синхронный режимы обмена, но COM-порты поддерживают только асинхронный режим. Функционально RS-232C эквивалентен стандарту МККТТ V.24/V.28 и стыку С2, но они имеют различные названия сигналов.

Стандарт RS-232C описывает несимметричные передатчики и приемники — сигнал передается относительно общего провода — схемной земли (симметричные дифференциальные сигналы используются в других интерфейсах — например, RS-422). Интерфейс не обеспечивает гальванической развязки устройств. Логической единице (состояние MARK) на входе данных (сигнал RxD) соответствует диапазон напряжения от –12 до –3 В; логическому нулю — от +3 до +12 В (состояние SPACE). Для входов управляющих сигналов состоянию ON (“включено”) соответствует диапазон от +3 до +12 В, состоянию OFF (“выключено”) — от –12 до –3 В. Диапазон от –3 до +3 В — зона нечувствительности, обусловливающая гистерезис приемника: состояние линии будет считаться измененным только после пересечения порога (рис. 2.8). Уровни сигналов на выходах передатчиков должны быть в диапазонах от –12 до –5 В и от +5 до +12 В. Разность потенциалов между схемными землями (SG) соединяемых устройств должна быть менее 2 В, при более высокой разности потенциалов возможно неверное восприятие сигналов. Заметим, что сигналы уровней ТТЛ (на входах и выходах микросхем UART) передаются в прямом коде для линий TxD и RxD и в инверсном — для всех остальных.

Интерфейс предполагает наличие защитного заземления для соединяемых устройств, если они оба питаются от сети переменного тока и имеют сетевые фильтры.

Стандарт RS-232C регламентирует типы применяемых разъемов.

На аппаратуре АПД (в том числе на COM-портах) принято устанавливать вилки DB-25P или более компактный вариант — DB-9P. Девятиштырьковые разъемы не имеют контактов для дополнительных сигналов, необходимых для синхронного режима (в большинстве 25-штырьковых разъемах эти контакты не используются).

На аппаратуре АКД (модемах) устанавливают розетки DB-25S или DB-9S.

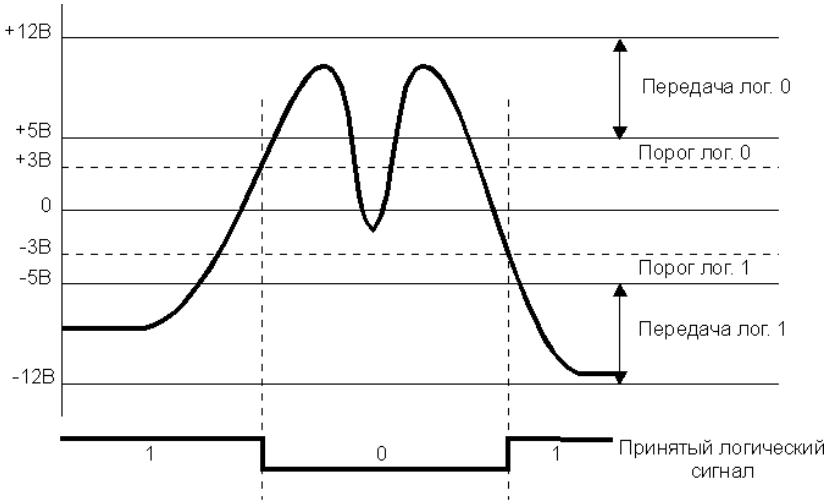


Рисунок 2.8 – Прием сигналов RS232.

Это правило предполагает, что разъемы АКД могут подключаться к разъемам АПД непосредственно или через переходные “прямые” кабели с розеткой и вилкой, у которых контакты соединены “один в один”. Переходные кабели могут являться и переходниками с 9 на 25-штырьковые разъемы (см. рис. 2.9).

Если аппаратура АПД соединяется без модемов, то разъемы устройств (вилки) соединяются между собой нуль-модемным кабелем (Zero-modem, или Z-modem), имеющим на обоих концах розетки, контакты которых соединяются перекрестно по одной из схем, приведенных на рис. 2.10.

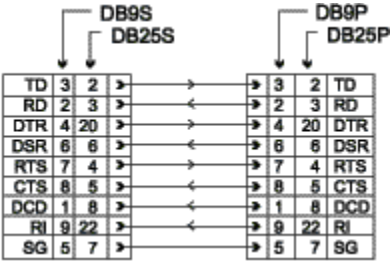


Рисунок 2.9 – Кабели для подключения.

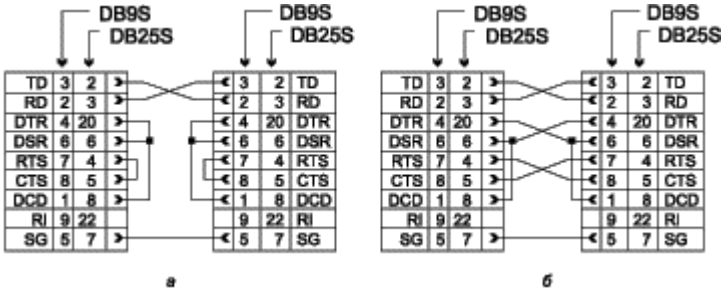


Рисунок 2.10 – Соединения в нуль-модемном кабеле.

Подмножество сигналов RS-232C, относящихся к асинхронному режиму, рассмотрим с точки зрения COM-порта PC. Для удобства будем пользоваться мнемоникой названий, принятой в описаниях COM-портов и большинства устройств (она отличается от безликих обозначений RS-232 и V.24). Напомним, что активному состоянию управляющих сигналов (“включено”) и нулевому значению бита передаваемых данных соответствует положительный потенциал (выше +3 В) сигнала интерфейса, а состоянию “выключено” и единичному биту — отрицательный (ниже –3 В).

2.4.2 Назначение сигналов интерфейса

Назначение сигналов для поддержки полного аппаратного интерфейса приведено ниже. Нормальную последовательность управляющих сигналов для случая подключения модема к COM-порту иллюстрирует рис. 2.11.

PG – Protected Ground — защитная земля, соединяется с корпусом устройства и экраном кабеля

SG – Signal Ground — сигнальная (схемная) земля, относительно которой действуют уровни сигналов

TD – Transmit Data — последовательные данные — выход передатчика

RD – Receive Data — последовательные данные — вход приемника

RTS – Request To Send — выход запроса передачи данных: состояние “включено” уведомляет модем о наличии у терминала данных для передачи. В полудуплексном режиме используется для управления направлением — состояние “включено” служит сигналом модему на переключение в режим передачи

CTS – Clear To Send — вход разрешения терминалу передавать данные. Состояние “выключено” запрещает передачу данных. Сигнал используется для аппаратного управления потоками данных

DSR – Data Set Ready — вход сигнала готовности от аппаратуры передачи данных (модем в рабочем режиме подключен к каналу и закончил действия по согласованию с аппаратурой на противоположном конце канала)

DTR – Data Terminal Ready — выход сигнала готовности терминала к обмену данными. Состояние “включено” поддерживает коммутируемый канал в состоянии соединения

DCD – Data Carrier Detected — вход сигнала обнаружения несущей удаленного модема

RI – Ring Indicator — вход индикатора вызова (звонка). В коммутируемом канале этим сигналом модем сигнализирует о принятии вызова.

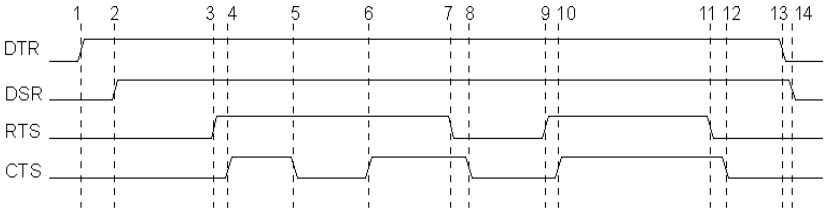


Рисунок 2.11 – Последовательность управляющих сигналов интерфейса.

Установкой DTR компьютер указывает на желание использовать модем.

Установкой DSR модем сигнализирует о своей готовности и установлении соединения.

Сигналом RTS компьютер запрашивает разрешение на передачу и заявляет о своей готовности принимать данные от модема.

Сигналом CTS модем уведомляет о своей готовности к приему данных от компьютера и передаче их в линию.

Снятием CTS модем сигнализирует о невозможности дальнейшего приема (например, буфер заполнен) — компьютер должен приостановить передачу данных.

Сигналом CTS модем разрешает компьютеру продолжить передачу (в буфере появилось место).

Снятие RTS может означать как заполнение буфера компьютера (модем должен приостановить передачу данных в компьютер), так и отсутствие данных для передачи в модем. Обычно в этом случае модем прекращает пересылку данных в компьютер.

Модем подтверждает снятие RTS сбросом CTS.

Компьютер повторно устанавливает RTS для возобновления передачи.

Модем подтверждает готовность к этим действиям.

Компьютер указывает на завершение обмена.

Модем отвечает подтверждением.

Компьютер снимает DTR, что обычно является сигналом на разрыв соединения (“повесить трубку”).

Модем сбросом DSR сигнализирует о разрыве соединения.

Из рассмотрения этой последовательности становятся понятными соединения DTR–DSR и RTS–CTS в нуль-модемных кабелях.

2.4.3 Асинхронный режим передачи

Асинхронный режим передачи является байт-ориентированным (символьно-ориентированным): минимальная пересылаемая единица информации — один байт (один символ). Формат посылки байта иллюстрирует рис. 2.12. Передача

каждого байта начинается со старт-бита, сигнализирующего приемнику о начале посылки, за которым следуют биты данных и, возможно, бит четности (Parity). Завершает посылку стоп-бит, гарантирующий паузу между посылками. Старт-бит следующего байта посылается в любой момент после стоп-бита, то есть между передачами возможны паузы произвольной длительности. Старт-бит, имеющий всегда строго определенное значение (логический 0), обеспечивает простой механизм синхронизации приемника по сигналу от передатчика.

Подразумевается, что приемник и передатчик работают на одной скорости обмена. Внутренний генератор синхронизации приемника использует счетчик-делитель опорной частоты, обнуляемый в момент приема начала старт-бита. Этот счетчик генерирует внутренние стробы, по которым приемник фиксирует последующие принимаемые биты. В идеале стробы располагаются в середине битовых интервалов, что позволяет принимать данные и при незначительном рассогласовании скоростей приемника и передатчика.

Очевидно, что при передаче 8 бит данных, одного контрольного и одного стоп-бита предельно допустимое рассогласование скоростей, при котором данные будут распознаны верно, не может превышать 5 %. С учетом фазовых искажений и дискретности работы внутреннего счетчика синхронизации реально допустимо меньшее отклонение частот. Чем меньше коэффициент деления опорной частоты внутреннего генератора (чем выше частота передачи), тем больше погрешность привязки стробов к середине битового интервала, и требования к согласованности частот становятся более строгими. Чем выше частота передачи, тем больше влияние искажений фронтов на фазу принимаемого сигнала. Взаимодействие этих факторов приводит к повышению требований к согласованности частот приемника и передатчика с ростом частоты обмена.

Рис. 2.12 – Формат асинхронной передачи RS-232C.

Формат асинхронной посылки позволяет выявлять возможные ошибки передачи.

Если принят перепад, сигнализирующий о начале посылки, а по стробу старт-бита зафиксирован уровень логической единицы, старт-бит считается ложным и приемник снова переходит в состояние ожидания. Об этой ошибке приемник может не сообщать.

Если во время, отведенное под стоп-бит, обнаружен уровень логического нуля, фиксируется ошибка стоп-бита.

Если применяется контроль четности, то после посылки бит данных передается контрольный бит. Этот бит дополняет количество единичных бит данных до четного или нечетного в зависимости от принятого соглашения. Прием байта с неверным значением контрольного бита приводит к фиксации ошибки.

Контроль формата позволяет обнаруживать обрыв линии: как правило, при обрыве приемник “видит” логический нуль, который сначала трактуется как старт-бит и нулевые биты данных, но потом срабатывает контроль стоп-бита.

Для асинхронного режима принят ряд стандартных скоростей обмена: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 и 115200 бит/с. Иногда вместо единицы измерения “бит/с” используют “бод” (baud), но при рассмотрении двоичных передаваемых сигналов это некорректно. В бодах принято измерять частоту изменения состояния линии, а при недвоичном способе кодирования (широко применяемом в современных модемах) в канале связи

скорости передачи бит (бит/с) и изменения сигнала (бод) могут отличаться в несколько раз.

Количество бит данных может составлять 5, 6, 7 или 8 (5- и 6-битные форматы распространены незначительно). Количество стоп-бит может быть 1, 1,5 или 2 (“полтора бита” означает только длительность стопового интервала).

2.4.4 Управление потоком данных

Для управления потоком данных (Flow Control) могут использоваться два варианта протокола — аппаратный и программный. Иногда управление потоком путают с квитированием. Квитирование (handshaking) подразумевает посылку уведомления о получении элемента, в то время как управление потоком предполагает посылку уведомления о возможности или невозможности последующего приема данных. Зачастую управление потоком основано на механизме квитирования.

Аппаратный протокол управления потоком RTS/CTS (hardware flow control) использует сигнал CTS, который позволяет остановить передачу данных, если приемник не готов к их приему (рис. 2.13). Передатчик “выпускает” очередной байт только при включенной линии CTS. Байт, который уже начал передаваться, задержать сигналом CTS невозможно (это гарантирует целостность посылки). Аппаратный протокол обеспечивает самую быструю реакцию передатчика на состояние приемника. Микросхемы асинхронных приемопередатчиков имеют не менее двух регистров в приемной части — сдвигающий, для приема очередной посылки, и хранящий, из которого считывается принятый байт. Это позволяет реализовать обмен по аппаратному протоколу без потери данных.

Рисунок 2.13 – Аппаратное управление потоком.

Аппаратный протокол удобно использовать при подключении принтеров и плоттеров, если они его поддерживают. При непосредственном (без модемов) соединении двух компьютеров аппаратный протокол требует перекрестного соединения линий RTS — CTS.

При непосредственном соединении у передающего терминала должно быть обеспечено состояние “включено” на линии CTS (соединением собственных линий RTS — CTS), в противном случае передатчик будет “молчать”.

Применяемые в IBM PC приемопередатчики 8250/16450/16550 сигнал CTS аппаратно не отрабатывают, а только показывают его состояние в регистре MSR. Реализация протокола RTS/CTS возлагается на драйвер BIOS Int 14h, и называть его “аппаратным” не совсем корректно. Если же программа, пользующаяся COM-портом, взаимодействует с UART на уровне регистров (а не через BIOS), то обработкой сигнала CTS для поддержки данного протокола она занимается сама. Ряд коммуникационных программ позволяет игнорировать сигнал CTS (если не используется модем), и для них не требуется соединение входа CTS с выходом даже своего сигнала RTS. Однако существуют и иные приемопередатчики (например, 8251), в которых сигнал CTS отрабатывается аппаратно. Для них, а также для “честных” программ, использование сигнала CTS на разъемах (а то и на кабелях) обязательно.

Программный протокол управления потоком XON/XOFF предполагает наличие двунаправленного канала передачи данных. Работает протокол следующим образом: если

устройство, принимающее данные, обнаруживает причины, по которым оно не может их дальше принимать, оно по обратному последовательному каналу посылает байт-символ XOFF (13h). Противоположное устройство, приняв этот символ, приостанавливает передачу. Когда принимающее устройство снова становится готовым к приему данных, оно посылает символ XON (11h), приняв который противоположное устройство возобновляет передачу. Время реакции передатчика на изменение состояния приемника по сравнению с аппаратным протоколом увеличивается, по крайней мере, на время передачи символа (XON или XOFF) плюс время реакции программы передатчика на прием символа (рис. 2.14). Из этого следует, что данные без потерь могут приниматься только приемником, имеющим дополнительный буфер принимаемых данных и сигнализирующим о неготовности заблаговременно (имея в буфере свободное место).

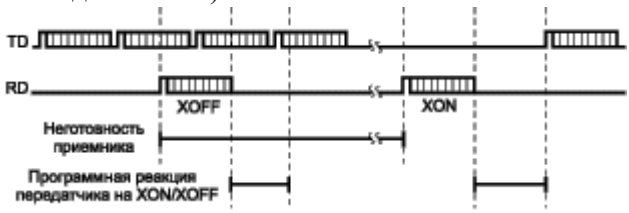


Рисунок 2.14 – Программное управление потоком XON/XOFF.

Преимущество программного протокола заключается в отсутствии необходимости передачи управляющих сигналов интерфейса — минимальный кабель для двустороннего обмена может иметь только 3 провода (RD,TD и GND). Недостатком, помимо обязательного наличия буфера и большего времени реакции (снижающего общую производительность канала из-за ожидания сигнала XON), является сложность реализации полнодуплексного режима обмена. В этом случае из потока принимаемых данных должны выделяться (и обрабатываться)

символы управления потоком, что ограничивает набор передаваемых символов.

Кроме этих двух распространенных стандартных протоколов, поддерживаемых и ПУ, и ОС, существуют и другие, которые не будут здесь рассмотрены.

Зачастую на практике в тех случаях, когда не требуется большой скорости передачи данных, поступают еще проще. Информацию из микроконтроллера в компьютер и обратно передают побайтно, реализуя какой-либо свой уникальный программный протокол. При этом всю ответственность да доставку и достоверность данных несет сам разработчик.

В простейшем режиме алгоритм обмена данными может выглядеть так. Формируется стандарт посылки одного кадра, который включает следующую информацию: первый байт – длина посылки, второй байт – служебный (который будет говорить о типе посылки. Например, это команда или данные), следующие несколько байт – данные, а затем байт контрольной суммы. Эта контрольная сумма служит гарантом достоверности передачи информации. В этом случае ведущее устройство (это может быть как МК, так и компьютер) поступает следующим образом. Например, посылаются данные объемом 16 байт. Посылка тогда будет выглядеть так: Старший байт (число 18) – длина посылки (сам этот байт может не учитываться), затем байт, говорящий о том, что это данные, затем 16 байт данных, а затем байт с контрольной суммой. Алгоритм формирования контрольной суммы может быть любой, например арифметическое сложение всех байт данных, начиная с первого. Ведомое устройство, получив первый байт, считает, что это длина посылки и активирует счетчик числа полученных байтов данных. Затем оно по второму байту определяет тип посылки и начинает принимать в буфер 16 байт. Приняв 17-й байт, ведомый считает, что это контрольная сумма, производит вычисление контрольной суммы принятой посылки и сравнивает принятую и вычисленную контрольные суммы. Если они

совпадают, то ведомый должен выслать ведущему подтверждение об успешном приеме данных посылкой, которая будет содержать три байта: объем передаваемой информации, затем команда, подтверждающая успешность приема, и контрольная сумма посылки. В случае, когда возникают какие-либо ошибки, разработчик должен сам предусматривать алгоритм, который бы обрабатывал ошибки и исправлял их.

2.4.5 Конвертер RS-232-TTL

При разработке различных электронных устройств с использованием микроконтроллеров очень часто оказывается полезной возможность подключения их к персональному компьютеру через последовательный порт. Однако напрямую это сделать невозможно, поскольку по стандарту RS-232, сигнал передается уровнями -3..-15В (логическая «1») и +3..+15В (логический «0»).

Для преобразования уровней RS-232 в стандартные логические уровни TTL обычно используют специальные микросхемы преобразователей. Однако далеко не всегда имеет смысл закладывать преобразователь уровней в схему проектируемого устройства, поскольку часто бывает так, что связь с компьютером нужна только на этапе изготовления и отладки устройства, а для конечного изделия в ней нет никакой необходимости. Логичным выходом в данной ситуации может послужить изготовление отдельного конвертера уровней RS-232 в TTL, схема которого приведена на рисунке 2.15.

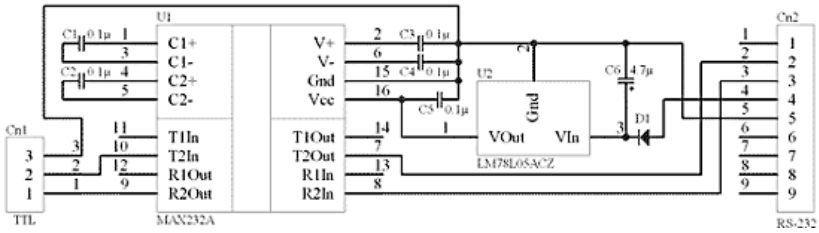


Рисунок 2.15 – Схема конвертера RS-232 – TTL.

Основу предлагаемого конвертера составляет широко распространенная микросхема преобразователя уровней MAX232A фирмы Maxim (U1), которая имеет также множество аналогов других производителей (Analog Devices, LG и др.).

Данная микросхема рассчитана на напряжение питания 5В и имеет встроенные удвоитель и инвертор напряжения на переключаемых конденсаторах для получения напряжений +10В и -10В, необходимых для работы с сигналами стандарта RS-232. Для работы микросхемы требуется 4 внешних конденсатора (C1, C2, C3, C4) емкостью 0.1 мкФ, которые используются в преобразователе напряжения. Кроме того, с целью упрощения использования данного конвертера, в нем предусмотрена схема питания прямо от последовательного порта, что избавляет от необходимости использования внешних источников питания. Напряжение питания 5В создается маломощным линейным стабилизатором напряжения LM78L05 (U2), вход которого подключен к накопительному конденсатору C6. Конденсатор C6 заряжается через диод от сигнала Data Terminal Ready (DTR, 4-й контакт 9-pin разъема RS-232). Диод D1 может быть любого типа.

Для нормальной работы такого преобразователя питания требуется, чтобы большую часть времени сигнал DTR имел значение логического нуля. Это должно обеспечиваться терминальной программой или программой пользователя.

Использование описанного выше конвертера оказывается удобным в тех случаях, когда в процессе эксплуатации устройства не требуется наличие возможности связи с компьютером, но она нужна на этапе отладки или изготовления устройства. Типичным примером этого может служить, например, устройство с Flash или EEPROM памятью, требующей начальной инициализации.

Кроме того, часто бывает очень удобно в процессе разработки выводить в последовательный порт различного рода отладочную информацию, что иногда позволяет обойтись без аппаратных эмуляторов.

2.5. Вариант сопряжения микроконтроллера с микросхемой Flash-памяти по протоколу I^2C.

2.5.1 Общее описание микросхемы Flash-памяти

Общее описание микросхемы Flash-памяти рассмотрим на примере микросхемы AT24C16 фирмы Atmel. Существует много аналогов этой микросхемы, выпускаемых другими производителями. Например, 24AA16 фирмы Microchip.

Основные характеристики микросхемы:

- напряжение низкого уровня и напряжение, соответствующее стандартным операциям переключения питания:

5.0 (Vcc = 4.5V - 5.5V), 2.7 (Vcc = 2.7V - 5.5V)

2.5 (Vcc = 2.5V - 5.5V), 1.8 (Vcc = 1.8V - 5.5V);

- организация внутреннего ПЗУ: 2048 x 8 бит (16 Кбит или 2 кБайта);
- 2-проводной последовательный интерфейс;
- двунаправленный протокол передачи данных;
- совместимость по частоте 100 кГц (1.8V, 2.5V, 2.7V) и 400 кГц (5V);
- вывод защиты записи, обеспечивающий аппаратную защиту данных;
- поддержка страничной записи в 16-байтовом режиме;
- поддержка неполной страничной записи;
- самосинхронизирующийся цикл записи (макс. - 10 мс);
- высокая надежность;
- гарантированное число циклов перезаписи: 1 миллион;
- сохранение данных в памяти: в течение 100 лет;
- возможность работы в широком диапазоне температур;
- 8-штырьковый или 14-штырьковый модуль JEDEC SOIC, 8-штырьковый модуль PDIP.

AT24C16 содержит 16384 бит последовательной памяти EEPROM (2048-битных слов), которая может быть перезаписана с помощью электрических сигналов и считана программным образом. Данной устройство разработано для применения в

промышленных и коммерческих областях, где важным условием являются невысокие значения мощности и напряжения.

AT24C16 включает 8-штырьковый модуль PDIP, 8- и 14-штырьковый модуль SOIC, доступ осуществляется через 2-проводной последовательный интерфейс. Кроме того, разработано несколько вариантов микросхем данного семейства: 5.0V (4.5V - 5.5V), 2.7V (2.7V - 5.5V), 1.8V(1.8V-5.5V).

Назначение ножек микросхемы:

Вывод	Функция
A0-A2	Адресные входы
SDA	Линия приема/передачи данных
SCL	Линия синхронизации
WP	Защита от записи
NC	Не подключен

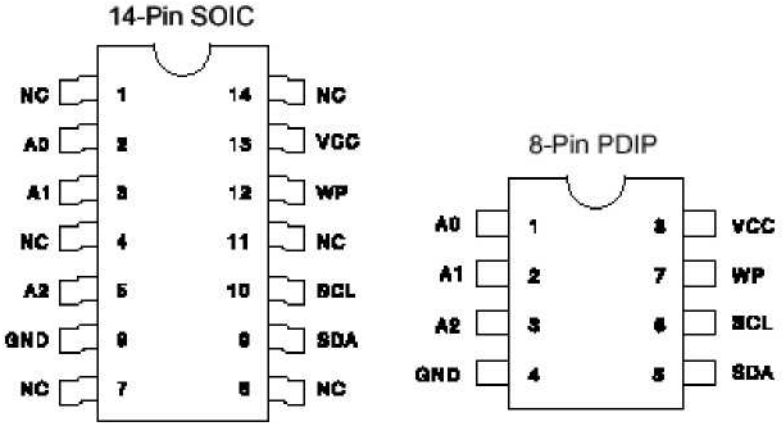


Рисунок 2.16 – Внешний вид микросхемы.

Описание выводов.

SERIAL CLOCK (SCL) - Линия синхронизации. Вход SCL используется при передаче в E^2PROM (положительный фронт) и отправке данных на внешнее устройство (отрицательный фронт).

SERIAL DATA (SDA). Это вывод для двунаправленной последовательной передачи данных. Вывод со свободным

стоком, к нему можно подключать любое количество открытых коллекторов или стоков.

Адреса страниц/устройства (A2, A1, AO). Выводы A2, A1 и A0 - это адресные входы устройств, разработанные для микросхем AT24C01A и AT24C02. На одинарную систему шин может быть адресовано до 8 1К/2К – устройств.

Микросхема AT24C16 не использует выводы A0, A1, A2.

Защита от записи (WP – Write Protection): Схемы семейства AT24Cxx имеют вывод защиты от записи, с помощью которых можно защитить аппаратные данные. Этот вывод используется для обычных операций чтения/записи в случае, если он подключен к «земле» (GND). Когда на этот вывод подается напряжение, свойство защиты от записи проявляется так, как показано в таблице ниже.

Организация памяти AT24C16: внутренняя память объемом в 16 Кбит состоит из 8 блоков. Каждый блок содержит 256 8-байтных страниц. Для произвольного доступа к данным необходима 11-битная адресация.

2.5.2 Работа с устройствами

Синхронизация и передача данных.

Вывод SDA обычно соединяется с внешним устройством. Данные могут быть переданы по SDA только тогда, когда на SCL подан сигнал низкого уровня (см. таблицу). Если на линии SCL – сигнал высокого уровня, то изменение уровня на SDA будет означать выдачу сигналов старт-стопных состояний, как описано ниже.

Состояние "старт": Изменение уровня сигнала с высокого на низкий на SDA при условии сигнала высокого уровня на линии SCL означает, что линия SDA находится в состоянии "старт", что должно предшествовать выполнению любой другой команды.

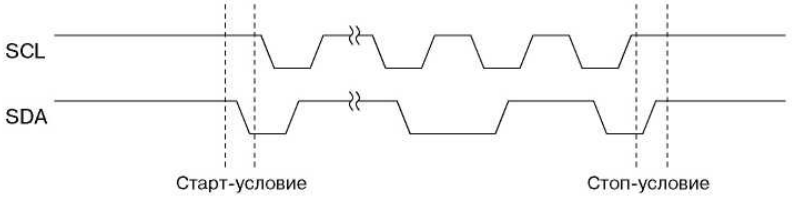


Рисунок 2.17 – Реализация условий «старт» и «стоп».

Состояние "стоп": Изменение уровня сигнала с низкого на высокий на линии SDA при условии наличия на линии SCL сигнала высокого уровня свидетельствует о том, что линия SDA находится в состоянии "стоп". После чтения данных и получения команды "стоп" E^2PROM перейдет в режим резервного питания.

Подтверждение приема: Все адреса и данные последовательно передаются с E^2PROM и на него в виде 8-битовых слов. После получения каждого слова E^2PROM выдает "0". Это происходит в процессе передачи 9-го импульса синхронизации.

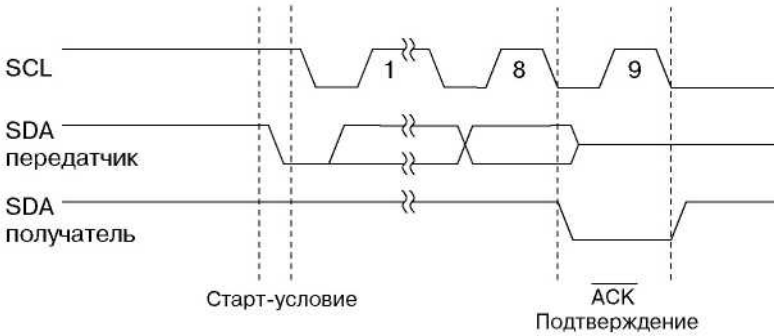


Рисунок 2.18 – Диаграмма подтверждения приёма байта по шине 1^2C.

Режим ожидания. Режим ожидания для схем семейства AT24C16 доступен после включения питания и получения стопового бита, а также после завершения любых внутренних операций.

2.5.3 Адресация устройств

Устройства E^2PROM с объемом памяти 16К после перехода в состояние "старт" должны получать слово (8 бит) с адресом устройства - только тогда микросхема сможет произвести операцию чтения или записи. Первые четыре бита слова адреса представляют собой обязательную последовательность "10". Данная последовательность идентична для всех устройств E^2PROM. Следующие 3 бита представляют собой адреса устройств A2, A1 и A0 - для 1К/2К E^2PROM. Эти биты соответствуют входам с аналогичными названиями. E^2PROM с объемом памяти 4К использует только биты адресов A2 и A1, а следующий за ними бит представляет собой адрес страницы памяти. Оба бита адресов устройств соответствуют выходам на микросхеме с аналогичными названиями. Вывод A0 не подключен.

Адресный байт для E^2PROM с объемом памяти 8К имеет только бит устройства A2, а биты A1 и A0 используются для адресации страницы памяти. Бит A2 соответствует выводу A2 на микросхеме. Выводы A1 и A0 не подключены. E^2PROM с 16К памяти не использует никаких устройств, и следующие 3 бита представляют собой адрес страницы памяти.

Следует обратить внимание на эти биты, используемые для адресации страниц памяти в устройствах 4К, 8К и 16К.

Младший бит адреса устройств используется для выбора режима чтения/записи. Если бит равен 1, происходит чтение, иначе запись. После сравнения адресов устройств E^2PROM выдает 0. Если сравнение не было произведено, микросхема возвращается в режим ожидания.

2.5.4 Операция записи

Запись байта: После того, как E^2PROM получит адресный байт и подтвердит возможность приема, должна происходить операция записи. Получив адрес и ответив выдачей "0",

устройство примет первые 8 бит данных. Затем E^{2}PROM выдает "0" и микроконтроллер должен остановить процесс записи путем выдачи стоп-сигнала. В этот момент E^{2}PROM начинает цикл записи в постоянную память. До тех пор, пока запись не будет завершена, отключаются все входы и E^{2}PROM не реагирует ни на какие сигналы.

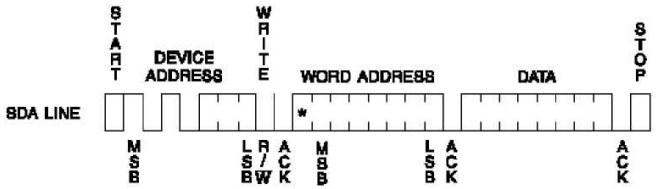
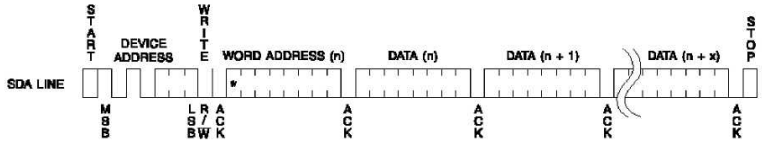


Рисунок 2.19 – Операция записи байта

Страничная запись: Устройства E^{2}PROM с объемом памяти в 16К производят 16-байтную запись. Процесс страничной записи инициируется также, как запись одного байта, отличие в том, что микроконтроллер после передачи первого слова не выдает стоп-сигнал. Вместо этого, как только E^{2}PROM подтвердит получение первого слова данных, микроконтроллер может передать ему еще до 15 слов данных. После получения каждого слова E^{2}PROM будет выдавать на линии "0". Микроконтроллер прекращает страничную запись, выдавая стоп-сигнал. Каждый раз, получив слово данных, E^{2}PROM инкрементирует младшие 4 адресных бита. Старшие адресные биты не инкрементируются. Если в E^{2}PROM передается больше 16 слов данных, адрес слова данных вернется на начало и предыдущие данные будут перезаписаны.



* Этот бит для устройства 1К может быть любым.

Рисунок 2.20 – Операция записи страницы

Опрос устройства: Как только E^2PROM начнет внутренне тактируемый цикл записи и отключит свои входы, можно инициировать запрос на подтверждение получения данных. Этот процесс включает отправку слова с адресом устройства, а затем выдачу стоп-сигнала. Бит чтения/записи устанавливается в зависимости от требуемой операции. E^2PROM выставит "0", позволяющий продолжить запись или чтение, только после завершения своего внутреннего цикла.

2.5.5 Операция чтения

Операция чтения инициируется точно так же, как и операция записи, за тем исключением, что бит чтения/записи в слове адреса устройства устанавливается равным 1. Существует 3 вида операции чтения: чтение текущего адреса, произвольная выборка адреса и последовательное чтение.

Чтение текущего адреса: Внутренний счетчик адреса содержит последний адрес, к которому производилось обращение во время операции чтения или записи, увеличенный на 1. Этот адрес остается корректным в промежутке между операциями до тех пор, пока к микросхеме подключено питание. Во время чтения адреса "перепрыгивают" с последнего байта последней страницы памяти на первый байт первой страницы. Во время записи адреса "перепрыгивают" с последнего байта текущей страницы на первый байт той же самой страницы.

Как только байт адреса устройства с битом чтения/записи, установленным 1, будет выдан микроконтроллером и принят E^2PROM, микроконтроллер принимает бит подтверждения, а затем принимает байт данных. После этого E^2PROM увеличивает внутренний счетчик адреса. Микроконтроллер после приема байта данных выдаст на вход не "0", а стоп-сигнал.

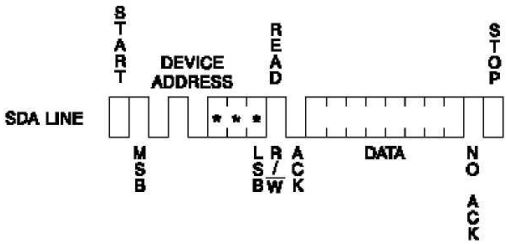


Рисунок 2.21 – Чтение байта по текущему адресу.

Чтение в режиме произвольного доступа: Как только слово адреса устройства и адрес данных будут приняты E^{2}PROM, микроконтроллер должен сгенерировать еще один старт-сигнал. Он инициирует чтение текущего адреса путем отправки адреса устройства с битом чтения/записи, установленным в «1». E^{2}PROM подтверждает получение адреса устройства выдачей ACK и микроконтроллер последовательно считывает слово данных. Затем он отвечает, но не выдачей "0", а генерацией стоп-сигнала.

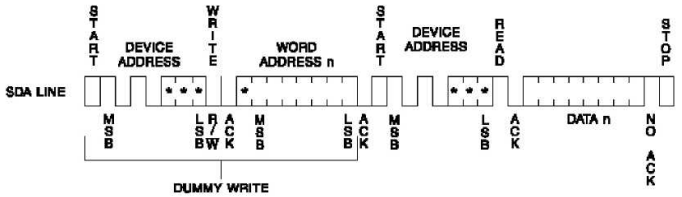


Рисунок 2.22 – Чтение байта по произвольному адресу.

Чтение в режиме последовательного доступа: Последовательное чтение данных инициируется в процессе либо чтения текущего адреса, либо чтения произвольного адреса.

После того, как микроконтроллер получит слово данных, он подтверждает их получение. Пока E^{2}PROM получает сигнал о подтверждении, он будет продолжать наращивать адрес слова данных и последовательно считывать слова данных. Когда счетчик достигнет верхнего адреса памяти, он "перепрыгнет" на начало и последовательное чтение будет продолжено.

3 ВОПРОСЫ К КОНТРОЛЬНОЙ РАБОТЕ № 1

Контрольная работа № 1 выполняется письменно.

1. Микропроцессорная техника. Определение.
2. Микропроцессорная система. Определение.
3. Микропроцессорное устройство. Определение.
4. Микропроцессор. Определение.
5. Микроконтроллер. Определение.
6. Области использования МК (привести сравнительную таблицу «Характеристика задач/разрядность-производительность»).
7. Основные направления развития микропроцессоров и микроконтроллеров.
8. Архитектура микроконтроллера. Определение.
9. Структура микроконтроллера. Определение.
10. Архитектура: CISC. Определение.
11. Архитектура:RISC. Определение.
12. Архитектура: VLIW. Определение.
13. Архитектура Фон-Неймана. Определение. Достоинства и недостатки.
14. Гарвардская архитектура. Определение. Достоинства и недостатки.
18. Классификация современных микропроцессоров по функциональному признаку.
19. Микропроцессоры общего назначения. Назначение, исходя из классификации по функциональному признаку.
20. Микроконтроллеры. Назначение, исходя из классификации по функциональному признаку.
21. Назначение и область применения 8-разрядных МК.
22. Назначение и область применения 16-разрядных МК.
23. Назначение и область применения 32-разрядных МК.
24. Цифровые процессоры сигналов. Определение, назначение и область применения.

26. Типовая структура микропроцессорной системы. Привести рисунок.

27. Основные режимы работы микропроцессорной системы.

28. Прерывание. Определение.

29. Исключение. Определение.

30. Классификация прерываний и исключений. Привести рисунок.

31. Маскируемые и немаскируемые прерывания. Определение.

32. Причины исключений.

33. Прямой доступ к памяти. Определение, назначение.

34. Виды структур микропроцессорной системы. Централизованная МПС. Определение.

35. Виды структур микропроцессорной системы. Децентрализованная МПС. Определение. Структура. Преимущества и недостатки.

36. Виды структур микропроцессорной системы. Иерархическая МПС. Определение. Структура. Преимущества и недостатки.

37. Основные этапы проектирования МПС. Блок-схема алгоритма.

4 ВОПРОСЫ К КОНТРОЛЬНОЙ РАБОТЕ № 2

Контрольная работа № 2 выполняется письменно.

1. Интерфейс I^2C. Определение.
2. Программируемая логика. Общее назначение и области использования.
3. Классификация ИС ПЛ по уровню интеграции и архитектуре.
4. Классификация ИС ПЛ по типу памяти конфигурации.
5. Классификация ИС ПЛ по зависимости задержек сигналов от путей распространения.
6. Схематическое представление ЛИЗМОП-транзистора с двойным затвором (привести рисунок и дать пояснения).
7. Программируемая логика. Ключевой транзистор, управляемый триггером памяти конфигурации (привести рисунок и дать к нему пояснения).
8. Схематическое представление программируемой перемычки «antifuse» (привести рисунок и дать пояснения)
9. FPGA – определение.
10. CPLD – определение.
11. Области применения микросхем с программируемой логикой.
12. Программируемая матричная логика. Определение.
13. Программируемая матричная логика. Привести схему основных элементов ПМЛ.
14. Базовые матричные кристаллы. Определение.
15. Схемотехника ИС ПЛ. Привести схему организации двунаправленных выводов.
16. Структура FPGA с межсоединениями. Рисунок.
17. Полные ресурсы межсоединений в микросхемах FPGA.
18. Упрощенная архитектура классической CPLD.

19. Структура модуля классического таймера-счетчика МК 51. Дополнительно привести рисунок, поясняющий измерение временного интервала. Достоинства и недостатки.

20. Принцип действия канала входного захвата таймера. Привести рисунок, поясняющий измерение временного интервала средствами канала входного захвата.

21. Принцип действия канала выходного сравнения таймера. Привести рисунок, поясняющий формирование временного интервала средствами канала выходного сравнения.

22. Что такое процессор событий?

23. Принцип реализации ШИМ.

24. Типовая структура модуля АЦП.

25. ЦАП на основе ШИМ-генератора.

5 БИЛЕТЫ К КОНТРОЛЬНОЙ РАБОТЕ № 3

Третья контрольная работа выполняется в конце семестра и включает вопросы по всему курсу. Каждый билет состоит из четырех вопросов: трех теоретических и одного практического.

Положительная оценка ставится в том случае, если студент ответил на практический вопрос и один из теоретических вопросов.

Билет 1.

1. Средства и методы проектирования и автономной отладки АС МП системы. Тестовая процедура. Определение.
2. Микропроцессорная техника. Определение.
3. Шина USB. Определение. Назначение.
4. Привести блок-схему алгоритма или листинг программы инициализации знакогенерирующего ЖКИ.

Билет 2.

5. Средства и методы проектирования и автономной отладки АС МП системы. Аппаратные средства отладки.
6. Микропроцессорная система. Определение.
7. Интерфейс I2C. Определение.
8. Привести блок-схему алгоритма или листинг программы опроса занятости ЖКИ по биту BF.

Билет 3.

9. Средства и методы отладки программных средств МП системы. Симулятор. Определение.
10. Микропроцессорное устройство. Определение.
11. Программируемая логика. Общее назначение и области использования.
12. Привести блок-схему алгоритма или листинг программы записи данных в регистр команд ЖКИ.

Билет 4.

13. Средства и методы отладки программных средств МП системы. Отладчик. Определение.

14. Микропроцессор. Определение.

15. Классификация ИС ПЛ по уровню интеграции и архитектуре.

16. Привести блок-схему алгоритма или листинг программы записи данных в регистр данных ЖКИ

Билет 5.

17. Прототипные платы. Системные комплекты. Определение.

18. Микроконтроллер. Определение.

19. Классификация ИС ПЛ по типу памяти конфигурации

20. Привести блок-схему алгоритма или листинг программы выборки адреса устройства с интерфейсом I2C.

Билет 6.

21. Прототипные платы. Отладочные платы и системы. Определение.

22. Основные направления развития микропроцессоров и микроконтроллеров.

23. Классификация ИС ПЛ по зависимости задержек сигналов от путей распространения

24. Привести блок-схему алгоритма или листинг программы сканирования матричной клавиатуры со структурой 4 х 4 клавиши.

Билет 7.

25. Прототипные платы. Одноплатные компьютеры и контроллеры. Определение.

26. Архитектура CISC. Определение.

27. Схематическое представление ЛИЗМОП-транзистора с двойным затвором. (привести рисунок и дать пояснения)

28. Привести блок-схему алгоритма или листинг программы устранения дребезга контактов при сканировании клавиатуры.

Билет 8.

29. Направления применений прототипных плат.

30. Архитектура RISC. Определение.

31. Программируемая логика. Ключевой транзистор, управляемый триггером памяти конфигурации (привести рисунок и дать к нему пояснения).

32. Привести способы устранения дребезга контактов при сканировании клавиатуры (программный, аппаратный, их виды).

Билет 9.

33. Интегрированные среды разработки (оболочки). Определение, назначение.

34. Архитектура VLIW. Определение.

35. Схематическое представление программируемой перемычки «antifuse» (привести рисунок и дать пояснения).

36. Привести блок-схему алгоритма или листинг программы установки адреса для вывода данных по адресу 41H ЖКИ.

Билет 10.

37. Эмулятор ПЗУ. Определение. Область применения.

38. Архитектура Фон-Неймана. Определение. Достоинства и недостатки.

39. FPGA – определение.

40. Привести блок-схему алгоритма или листинг программы пересылки 1 байта по UART с использованием прерываний для фиксации окончания передачи.

Билет 11.

41. Внутрисхемные эмуляторы. Определение. Область применения.

42. Гарвардская архитектура. Определение. Достоинства и недостатки.

43. CPLD – определение.

44. Привести блок-схему алгоритма или листинг программы приема 1 байта данных по UART с использованием прерывания.

Билет 12.

45. Внутрисхемные эмуляторы. Блоки, входящие в структуру.

46. Классификация современных микропроцессоров по функциональному признаку.

47. Области применения микросхем с программируемой логикой.

48. Привести блок-схему алгоритма или листинг программы приема 1 байта данных по UART без использования прерываний.

Билет 13.

49. Программаторы. Определение.

50. Назначение и область применения 8-разрядных МК.

51. Программируемая матричная логика. Определение.

52. Привести блок-схему алгоритма или листинг программы установки ЖКИ в режим развертки одной строки с отображением курсора в виде подчерка.

Билет 14.

53. Программаторы. Виды, назначение.

54. Назначение и область применения 16-разрядных МК.

55. Базовые матричные кристаллы. Определение.

56. Привести блок-схему алгоритма или листинг программы установки ЖКИ в режим развертки одной строки с отображением курсора в виде мерцающего знакоместа.

Билет 15.

57. Программаторы. Область применения.

58. Назначение и область применения 32-разрядных МК.

59. Схемотехника ИС ПЛ. Привести схему организации двунаправленных выводов.

60. Привести блок-схему алгоритма или листинг программы чтения байта данных из Flash-памяти (емкость 1k) по протоколу I2C.

Билет 16.

61. Логические анализаторы. Определение.

62. Цифровые процессоры сигналов. Определение, назначение и область применения.

63. Схемотехника ИС ПЛ. Привести схему организации двунаправленных выводов.

64. Привести блок-схему алгоритма или листинг программы записи байта данных по текущему адресу в Flash-память (емкость 1k) по протоколу I2C.

Билет 17.

65. Логические анализаторы. Область применения.

66. Прерывание. Определение. Типы прерываний.

67. Основные этапы процедуры проектирования.

68. Привести блок-схему алгоритма или листинг программы настройки UART на режим 3 (TH1=0FDH) с использованием прерываний.

Билет 18.

69. Распределение аппаратных и программных средств в процессе проектирования МП устройства.

70. Исключение. Определение. Типы исключений.

71. Сопряженное проектирование. Определение.

72. Привести блок-схему алгоритма или листинг программы настройки UART на режим 1 (TH1=0FDH) с использованием прерываний.

6 ВАРИАНТЫ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ № 1

Разработать алгоритм приема (передачи) данных по UART, используя прерывания от последовательно порта, и реализовать программу на ассемблере МК51 по вариантам:

№	Направление передачи данных	Режим UART	Скорость передачи, кбит/с
1	Передача	0	500
2	Передача	0	1000
3	Передача	0	2000
4	Прием	0	500
5	Прием	0	1000
6	Прием	0	2000
7	Передача	1	1200
8	Передача	1	2400
9	Передача	1	4800
10	Прием	1	1200
11	Прием	1	2400
12	Прием	1	4800
13	Передача	2	187,5
14	Передача	2	375
15	Передача	2	750
16	Прием	2	187,5
17	Прием	2	375
18	Прием	2	750
19	Передача	3	1200
20	Передача	3	2400
21	Передача	3	4800
22	Прием	3	1200
23	Прием	3	2400
24	Прием	3	4800

Дополнительные условия:

1. Начальный адрес массива для приема или передачи данных вычисляется по формуле: $№_{группы} + №_{вар.}$

Например: гр. 361-**1**, вар. № **2**. Начальный адрес массива: 3=1 + 2

2. Количество передаваемых (принимаемых) байт = № варианта * 2.

Отчет должен содержать следующее:

1) титульный лист;

2) лист с заданием;

3) блок-схема алгоритма программы;

4) листинг программы с комментариями.

7 ВАРИАНТЫ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ № 2

Первый тип задания № 2.

Задание: реализовать протокол обмена данными микроконтроллера с Flash-памятью по шине I^2C в соответствии с заданным вариантом.

Таблица 7.1 – Числовые данные.

№	Операция		Объем Flash, кБит	Объем данных, байт	Тип операции	Нач. адрес Flash
	Запись	Чтение				
1	+		1	5	Побайтная	0
2		+	1	10	Побайтная	10H
3	+		2	15	Побайтная	20H
4		+	2	20	Побайтная	30H
5	+		4	25	Побайтная	40H
6		+	4	30	Побайтная	50H
7	+		8	35	Побайтная	60H
8		+	8	40	Побайтная	70H
9	+		16	45	Побайтная	15H
10		+	16	50	Побайтная	25H
11	+		1	5	Страничная запись	0
12		+	1	10	Чтение послед-ти	10H
13	+		2	15	Страничная запись	20H
14		+	2	20	Чтение послед-ти	30H
15	+		4	25	Страничная запись	40H
16		+	4	30	Чтение послед-ти	50H
17	+		8	35	Страничная запись	60H
18		+	8	40	Чтение послед-ти	70H

Продолжение таблицы 7.1.

№	Операция		Объем Flash, кБит	Объем данных, байт	Тип операции	Нач. адрес Flash
	Запись	Чтение				
19	+		16	45	Страничная запись	15H
20		+	16	50	Чтение послед-ти	25H

Дополнительные условия:

1. Начальный адрес массива для приема или передачи данных вычисляется по формуле: $№_{группы} + №_{вар}$.

Например: гр. 361-**1**, вар. № **2**. Начальный адрес массива: 3=1 + 2

2. Число передаваемых (принимаемых) байт = № вар. * 2.

Отчет должен содержать титульный лист, лист с заданием, блок-схему алгоритма программы, листинг программы с комментариями.

Второй тип задания № 2.

Разработать алгоритм формирования периодического сигнала Uвых. на порту P1.0, на основе таймера, если на вход прерывания(int0,int1) поступает периодический сигнал Uвх. в соответствии с приведенным ниже рисунком, и реализовать программу на ассемблере МК51 по вариантам:

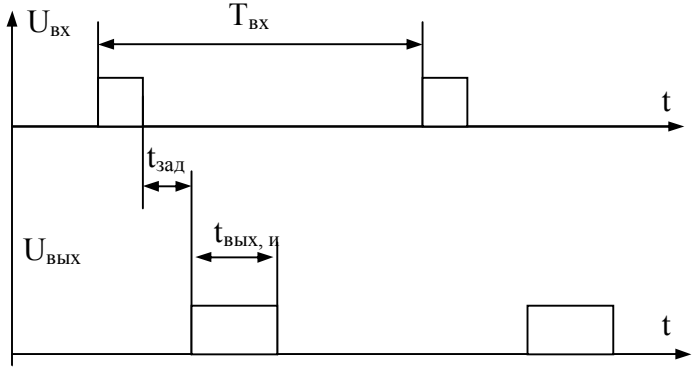


Рисунок 7.1 – Временные диаграммы к заданию № 2.

Таблица 7.2 – Числовые данные

№	Fкв, МГц	$T_{вх}$, Мс	$t_{вых.и.}$, мкс	$N=t_{зад}/T_{вх}$	
1	12	40	480	3/16	
2	8	20	720	4/16	
3	6	10	240	5/16	
4	4	40	480	6/16	
5	12	20	720	7/16	
6	8	10	240	3/16	
7	6	40	480	4/16	
8	4	20	720	5/16	
9	12	10	240	6/16	
10	8	40	480	7/16	
11	6	20	720	3/16	
12	4	10	240	4/16	
13	12	40	480	5/16	
14	8	20	720	6/16	
15	6	10	240	7/16	
16	4	40	480	3/16	
17	12	20	720	4/16	
18	8	10	240	5/16	
19	6	40	480	6/16	

8 ТВОРЧЕСКОЕ ЗАДАНИЕ

Творческое задание выдается студенту при достижении им рейтингового порога в 80 баллов для повышения своей рейтинговой оценки.

Данное задание предлагается к выполнению творчески активным студентам и предполагает детальную проработку блока какого-либо устройства или написание достаточно сложной управляющей программы для микроконтроллера. Выполнение творческого задания поможет студентам при выполнении курсового проектирования в следующем семестре, так как во многих случаях предполагает не только теоретическую разработку блока устройства, но и создание реального макета блока и отладку управляющей программы с его помощью или с помощью программно-аппаратного лабораторного комплекса, который позволяет реализовать типовые конфигурации микропроцессорной системы.

Примеры вариантов творческого задания:

1. Написать программу обмена данными между двумя микроконтроллерами: ведущим и ведомым с проверкой контроля на четность.

2. Написать программу, реализующую протокол RC-5 для микроконтроллера семейства MSC-51 или AVR;

3. Разработать схему электрическую функциональную и блок-схему алгоритма работы устройства – электронный школьный звонок.

Для примера конкретизируем задание для варианта № 3.

Устройство управления школьным звонком должно реализовать полный годовой календарь, включающий минуты, часы, день недели, число, месяц, год и работать от внешнего источника питания постоянного напряжения +5 В и реализовать реальную сетку расписания школьных уроков. Для этого

необходимо обеспечить функцию программирования сетки расписания пользователем.

Сеток расписания должно быть две: для работы с понедельника по пятницу и отдельно - для субботы, так как в субботу занятия проводятся по сокращенному расписанию.

Состав устройства:

1. Микроконтроллер;

2. Жидкокристаллический алфавитно-цифровой индикатор – 4 строки по 20 знаков каждая;

3. Микросхема энергонезависимой памяти для хранения сеток расписания;

4. Микросхема часов для реализации времени;

5. Клавиши управления прибором: выбор режима, настройка даты/времени, запуск, тревога.

6. Силовой ключ, управляющий схемой включения звонка на 220 В.

7. Блок питания от сети 220 В. Выходное напряжение блока питания +5 В.

8. Пластиковый корпус.

9 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

9.1 Общие положения

Курсовой проект логически завершает цикл «Цифровая и микропроцессорная техника» и предполагает разработку студентом законченного устройства с применением микропроцессорных средств.

Рассмотрим основные этапы разработки:

- введение;
- конкретизация технического задания, включающая, кроме технических параметров, четкую формулировку функций, выполняемых устройством;
- разбиение устройства на отдельные функциональные блоки;
- выбор реализации блоков: программно или аппаратно;
- разработка функциональных схем блоков, реализуемых аппаратным способом;
- разработка блок-схемы алгоритмов блоков, реализуемых программным способом;
- разработка схемы электрической принципиальной и расчет элементов схемы. Составление перечня элементов;
- практическое изготовление блоков прибора, включающих пайку схемы на макетной печатной плате и ее настройку (этот этап выполняется по согласованию с преподавателем);
- написание прикладной программы для микроконтроллера;
- отладка отдельных модулей прикладной программы на программном симуляторе микроконтроллера на ПК;
- программирование реального микроконтроллера и комплексная отладка прибора в целом (выполнение этого этапа производится по согласованию с преподавателем);
- составление пояснительной записки;
- сдача пояснительной записки на проверку преподавателю;
- защита курсового проекта.

Необходимо отметить, что данное разделение этапов разработки чисто условное, и при проектировании реального прибора несколько этапов могут быть объединены в один и наоборот, один этап может быть разложен на более мелкие составляющие.

9.2 Структура пояснительной записки

Пояснительная записка является основным документом, в котором описываются все этапы разработки, приводятся все расчеты, схемы, диаграммы, листинги и на основании которой студенту выставляется оценка.

При составлении пояснительной записки желательно придерживаться следующей структуры.

9.2.1 Титульный лист

Титульный лист выполняется в соответствии с положениями ГОСТ. Пример оформления титульного листа приведен в приложении 1.

9.2.2 Аннотация

Аннотация составляется на русском языке на отдельном листе в соответствии с ГОСТ и содержит краткое описание проектируемого устройства.

9.2.3 Задание на проектирование

Задание на проектирование выполняется в соответствии с ГОСТ. Пример оформления задания приведен в приложении 2.

9.2.4 Содержание

Содержание выполняется на отдельном листе (листах) в соответствии с ГОСТ. Пример оформления содержания приведен в приложении 3.

9.2.5 Введение

Введение (и прочие текстовые блоки) выполняется в соответствии с общими правилами оформления текстовых документов ГОСТ и должно кратко описывать область, к которой относится данное устройство, текущее состояние развития данного класса устройств, обоснование необходимости разработки нового устройства.

9.2.6 Конкретизация технического задания

Задание на курсовой проект выдается в очень краткой форме. Оно может содержать лишь название и область применения проектируемого устройства. В данном разделе необходимо конкретизировать условия работы устройства и перечень выполняемых им функций, задать ограничения на эти функции и обосновать разрядность обрабатываемых данных. Определяющим должно быть удобство практической эксплуатации и новый набор качественных характеристик (более высокое быстродействие, точность, новые функциональные возможности) по сравнению с вариантом реализации подобного устройства на дискретных элементах или функционального аналога.

Здесь определяется диапазон рабочих температур, конструктивное исполнение (плата, блок или отдельное изделие), область применения. Во многих случаях проектируемое устройство наряду с цифровыми может содержать и аналоговые узлы (измерительные усилители, ЦАП, АЦП). И хотя детальной разработке подлежат чисто цифровые

узлы, аналоговые блоки и выполняемые ими функции должны быть описаны при разработке принципиальной схемы проектируемого устройства.

9.2.7 Разработка функциональной схемы устройства

Схема электрическая функциональная выполняется в соответствии с ГОСТ. Она (код схемы Э2) разъясняет процессы, происходящие в отдельных функциональных частях устройства и в устройстве в целом и строится для оптимального варианта проектируемого устройства, который необходимо выбрать из нескольких возможных.

На этом этапе однозначно определяется алгоритм работы устройства в целом и оптимальный состав аппаратных и программных средств. Для этого необходимо определить, какие функции устройства будут реализованы программно, а какие – аппаратно, причем аппаратные блоки делятся, в свою очередь, на цифровые и аналоговые. На основании этого выбирается тип микроконтроллера, объем и тип его памяти, а также набор дополнительных цифровых элементов, необходимых для реализации цифровой части устройства. Степень детализации проработки должна быть достаточной для того, чтобы на последующих стадиях можно было производить независимую разработку аппаратных и программных средств микропроцессорной системы.

9.2.8 Разработка блок-схемы алгоритма программы

Разработке исходного текста прикладной программы микроконтроллера предшествует разработка блок-схемы алгоритма этой программы. Используется метод декомпозиции, при котором вся задача последовательно разделяется на меньшие функциональные модули (подпрограммы), каждый из которых можно разрабатывать отдельно от других. Разделение задачи на модули и операторы выполняется последовательно до

такого уровня, когда просматривается возможность реализации модуля с помощью нескольких или даже одной команды микропроцессора.

Блок-схема представляет весь алгоритм программы в строгой логической последовательности и позволяет разработчику программы определить формат внутреннего представления переменных и предварительно распределить внутренние программно доступные ресурсы микроконтроллера для их реализации (регистры общего назначения, ячейки резидентной памяти данных, флаги пользователя и т.п.).

Важным свойством блок-схемы является то, что она позволяет абстрагироваться от конкретного микроконтроллера, так как описывает сам алгоритм, т.е. последовательность действий программы, а не сами команды. А разработчик может выбрать тот тип микроконтроллера, который будет наиболее оптимальным по соотношению цена/функциональные возможности.

Пример оформления блок-схемы алгоритма приведен в приложении 4.

Собственно при выполнении курсового проекта можно как приводить блок схемы непосредственно в тексте пояснительной записки, так и выносить их в приложение.

9.2.9 Разработка схемы электрической принципиальной

Схема электрическая принципиальная выполняется в соответствии с ГОСТ и определяет полный состав элементов (микросхем, резисторов, конденсаторов и т.д.) и связей между ними. Она служит исходным документом для разработки других конструкторских документов (печатных плат, сборочных чертежей). Проектируемое устройство, как правило, содержит печатную плату с микросхемами, источники вторичного электропитания и базовый блок, на лицевой панели которого располагаются кнопки, тумблеры, программные переключатели,

светодиоды, цифровые индикаторы. Принципиальная схема разрабатывается отдельно для каждого блока (кроме источников питания) и для всего устройства в целом (связи между разъемами блоков).

На принципиальных схемах цифровых интегральных микросхем обычно не изображаются выводы для подключения источников питания. Эти соединения приводятся в текстовой или табличной информации. Условные графические обозначения и линии связей выполняют линиями одной и той же толщины. Утолщенные линии используются для обозначения линий групповой связи.

Латинский алфавит определяет последовательность расположения обозначений в перечне элементов: конденсаторы (C1, C2, C3...C5), аналоговые микросхемы (DA1, DA2...DA8), цифровые микросхемы (DD1, DD2...DD6), резисторы (R1, R2...R5, R6), полупроводниковые приборы (VD1, VT1...VT3), разъем (XP1) и т.д.

На этом этапе необходимо провести расчет всех элементов схемы и обосновать выбор конкретных комплектующих.

9.2.10 Разработка прикладной программы

Микроконтроллер решает стоящие перед ним задачи управления объектами в реальном масштабе времени. У разработчика есть две возможности написать программу: на языке ассемблера и на языке высокого уровня (например, на Си). С целью обеспечения максимального быстродействия прикладные программы разрабатываются на языке ассемблера. По сравнению с программами, подготовленными на алгоритмических языках высокого уровня, они после трансляции требуют меньшего объема памяти программ.

Но к настоящему времени большинство разработчиков пользуются языками высокого уровня, так как они позволяют писать достаточно сложные программы за очень короткий срок.

В этом случае разработчику нет необходимости долго думать над отдельными операциями, как это приходится делать на ассемблере. Например, необходимо выполнить операцию деления 4-х байтового числа на однобайтовое число. Разработка этой процедуры на ассемблере у студента займет не один час, а на языке Си эта операция записывается в одну строку. Кроме того, на языке высокого уровня легче прослеживать логику работы программы и очень легко модифицировать, не беспокоясь о таких вещах, как, например, максимальное расстояние (в байтах) от текущей команды до метки при выполнении условного или безусловного перехода.

Но для лучшего понимания архитектуры, программной модели микроконтроллера и с целью получения опыта написания простых программ рекомендуется разрабатывать программы сначала на ассемблере, а уже потом переходить на языки высокого уровня.

Широко распространен модульный принцип построения прикладной программы. Такая программа содержит основной модуль и ряд подпрограмм, к которым он обращается по мере необходимости. Подпрограмма должна выполнять законченную процедуру обработки информации, иметь один вход и один выход. Любая подпрограмма допускает автономную отладку.

Для трансляции исходных программ, предварительно записанных с помощью любого текстового редактора на ассемблере(*.asm) или языке высокого уровня, например С (*.c), используют программу ассемблер или компилятор С. В процессе преобразования программы в объектный модуль ассемблер может выявить синтаксические ошибки, связанные с несоблюдением правил записи команд. После исправления ошибок вызывают редактор связей (линковщик), позволяющий компоновать из отдельных объектных модулей единый загрузочный модуль формата *.hex фирмы Intel, который непосредственно используется программатором для записи программы в память программ или для отладки программы с

помощью программного симулятора или аппаратного эмулятора. Отладка позволяет обнаружить смысловые ошибки, не позволяющие программе выполнить функции, заложенные разработчиком.

Не всегда на симуляторе можно проверить всю программу. Это касается тех случаев, когда, например, необходимо произвести опрос матричной клавиатуры или получить данные с какого-либо датчика в последовательном цифровом коде определенной длительности. В этом случае проверяются отдельные подпрограммы или фрагменты программы. Необходимо помнить, что скорость реализации команд на симуляторе значительно меньше, чем в реальном масштабе времени (иногда в 1000 раз). Поэтому подпрограммы временной задержки нужно модифицировать или вообще исключить из программы, тестируемой с помощью симулятора.

С помощью аппаратного эмулятора можно проверять программу полностью, но для этого необходимо будет подключать к эмулятору реальные внешние блоки (клавиатуру, датчики), чтобы проверить работу программы.

Конечным продуктом разработки прикладной программы является ее листинг (файл с расширением *.lst) и карта прошивки ПЗУ (файл с расширением *.hex).

9.2.11 Заключение

В этом разделе необходимо подвести итог своей работе, критически рассмотрев ее со всех точек зрения. Необходимо сделать выводы о том, насколько актуальна и как раскрыта тема задания, на каком уровне произведена разработка (высоком, среднем, низком), насколько конкретными получались результаты и насколько они соответствуют результатам, полученным при изучении аналогов.

9.2.12 Оформление и защита проекта

Выполненный курсовой проект оформляется в виде пояснительной записки объемом 15-25 страниц и чертежей принципиальных схем, которые при небольшом формате могут быть вшиты в пояснительную записку. При оформлении необходимо соблюдать требования и правила, оговоренные в стандарте вуза по оформлению курсовых и дипломных проектов. Подчеркнем, что введение и заключение являются необходимыми разделами пояснительной записки.

Защита включает пятиминутный доклад и ответы на контрольные вопросы. Доклад должен отражать: тему проекта, цель и назначение разработки, актуальность темы, основные задачи, решаемые в проекте, основное содержание (с привлечением графического материала пояснительной записки и чертежей), выводы и рекомендации по результатам курсового проекта. Доклад ведется от третьего лица и носит характер сообщения о проделанной работе перед коллегами.

10 Пример проектирования устройства «Электронный школьный звонок».

Этапы курсового проекта рассмотрим на примере проектирования устройства «Электронный школьный звонок». Сразу сделаем оговорку, что пример будет приведен в сжатом исполнении, отражающем только основные этапы проектирования.

Пример технического задания приведен в приложении 2.

10.1. Введение

В настоящее время автоматизация применяется практически во всех сферах человеческой деятельности. Так как звонки в школах подчиняются строгому расписанию, автоматизация подачи звонков представляется вполне естественным шагом.

Проведя поиск литературы по данной теме, в том числе и в сети Интернет, можно сделать вывод, что на рынке представлено большое число предложений подобного рода. Проанализировав их, рассмотрев все их достоинства и недостатки, можно сформулировать требования к разрабатываемому устройству.

10.2 Конкретизация технического задания.

Конкретизируем требования к заданию на проектирование. Устройство должно:

- включать и выключать звонок на уроки автоматически в течение всего учебного года.
- сохранять работоспособность часов в случае отключения внешнего питания. В противном случае после включения электричества нужно будет заново устанавливать текущее время и дату;
- реализовать реальную сетку расписаний на уроки с учетом того, что в некоторые дни (например, в субботу) расписание может отличаться от обычного.
- иметь индикатор для отображения даты, текущего времени, номера урока и смены.
- иметь возможность изменения сетки расписания конечным пользователем, т.е. работниками школы. Для этого нужно предусмотреть режим редактирования сетки или с помощью самого устройства или с помощью специального программного обеспечения при подключении устройства к компьютеру.
- комплектоваться клавиатурой, с помощью которой можно будет производить настройку даты и времени;
- иметь простую возможность принудительного отключения звонка с помощью одной кнопки или выключателя;
- иметь кнопку подачи тревожного звонка и по возможности быть подключено к линии «Тревога» пожарной сигнализации.

Технические требования:

- напряжение питания устройства: 5 В:
- потребляемая мощность: не более 1 Вт;
- коммутируемое напряжение для звонка: 220 В , 50 Гц, 1 А;
- время работоспособности часов реального времени от резервного источника питания: не менее 24 часов;
- тип памяти для хранения сеток расписания: энергонезависимая.

10.3 Разработка структуры устройства

Исходя из детализированных требований к заданию, можно предложить следующую структуру устройства.

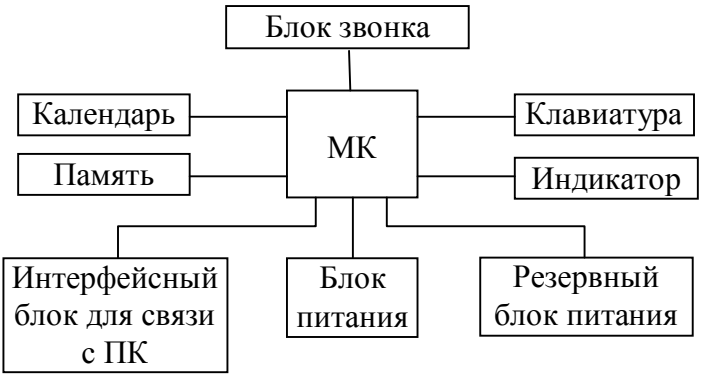


Рисунок 10.1 – Структурная схема устройства.

На этой схеме представлены все блоки, которые входят в структуру проектируемого устройства.

Календарь – это блок, который отвечает за реализацию функции полного годового календаря.

Память – это блок, в котором хранятся сетки расписаний.

Интерфейсный блок – это блок связи устройства с компьютером.

Блок звонка – это блок, осуществляющий замыкание и размыкание силовой цепи для подачи звонка.

10.4 Разработка функциональной схемы устройства

Перед разработкой собственно функциональной схемы проанализируем структуру устройства и произведем разделение на программные и аппаратные блоки. Рассматривать будем только те блоки, функции которых можно реализовать двояко: или в виде отдельного устройства или программно с помощью микроконтроллера.

Одним из таких блоков является блок памяти, в которой будут храниться сетки расписаний. Из задания следует, что содержимое памяти должно сохраняться при выключении питания и пользователь должен иметь возможность самостоятельно изменять содержимое памяти или с клавиатуры устройства или из компьютера через один из стандартных портов ввода-вывода. Следовательно, необходимо использовать тип памяти EEPROM или FLASH. Можно применить отдельную микросхему памяти и подключить ее к микроконтроллеру или использовать однокристальный микроконтроллер со встроенной EEPROM. Если использовать микроконтроллер AT89S51 семейства MSC51, то у него нет встроенной памяти данных EEPROM. Если рассмотреть дальше модельный ряд микроконтроллеров компании ATMEL, то можно выбрать или модель AT89S8252 (2 Кб EEPROM) семейства MSC51 или микроконтроллер ATMega8 (512 байт EEPROM) или ATMega16 (1 кбайт EEPROM) семейства AVR. В этом случае необходимо выяснить, какой вариант лучше. Отдельная микросхема памяти увеличит стоимость изделия на свою себестоимость, стоимость ее монтажа, увеличит размеры печатной платы и ухудшит коэффициент надежности изделия в целом. С другой стороны, микроконтроллер со встроенной EEPROM памятью может оказаться дороже, чем первый вариант.

Приведем краткие цифры розничной цены по результатам поиска с помощью сайта www.efind.ru.

Стоимость МК AT89S51 около 50 руб.; стоимость микросхемы памяти AT24C16 (2 кБайта EEPROM) около 15 руб.; стоимость МК ATMega8 – около 45 руб. Анализ стоимости обоих вариантов показал, что применение однокристального микроконтроллера ATMega8 выгоднее, чем применение отдельной микросхемы памяти.

Второй блок, подлежащий рассмотрению – это блок реализации полного календаря. Можно воспользоваться отдельной микросхемой, например PCF8583, которая реализует полный календарь и использует шину I^2C для связи с микроконтроллером. Другой вариант – программная реализация функции часов при помощи таймеров микроконтроллера. Рассмотрим оба варианта.

Вариант программной реализации на первый взгляд дешевле, так как не увеличивает стоимость готового изделия на стоимость отдельной микросхемы часов. Но необходимо учесть еще один фактор. Так как по условию задания часы должны сохранять работоспособность при отключении внешнего питания не менее чем 24 часа, то при программной реализации функции календаря в случае отключения питания нужно будет микроконтроллер переводить на питание от резервной батареи, причем таймер, на котором будет основана работа календаря, должен продолжать работать. Следовательно, нужно разработать блок, который будет следить за уровнем внешнего питания и при его понижении подключать резервную батарею. Для этих целей можно использовать специальную микросхему – супервизор питания. Кроме того, необходимо предусмотреть блок, который будет поддерживать резервную батарею в заряженном состоянии (если эти функции не выполняет супервизор).

Но рассмотренный вариант имеет следующие недостатки:

- батарея питания имеет ограниченный ресурс, поэтому придется ее менять раз в 2-3 года. Для этого необходимо будет предусмотреть батарейный отсек с отдельной крышкой для простой замены аккумуляторной батареи;

- супервизор питания – достаточно дорогая микросхема. Ее стоимость по меньшей мере в два раза превышает стоимость;

- разработка дополнительного блока слежения за зарядом батареи тоже вызовет удорожание прибора в целом.

Второй вариант – это применение отдельной микросхемы для реализации календаря, например PCF8583. Данная микросхема имеет связь с микроконтроллером по последовательной шине стандарта I^2C. Она легко программируется с помощью программных средств микроконтроллера. Питание этой микросхемы можно осуществлять от ионистора емкостью, например 0,1 Ф. Он включается параллельно цепи питания. В этом случае никакие дополнительные затраты не нужны. При отключении внешнего питания функционирование всего устройства прекращается, но микросхема часов продолжается работать от ионистора. Чтобы электроэнергия ионистора не расходовалась на питание всей схемы, необходимо поставить обратный диод. При восстановлении питания устройства в микроконтроллер из микросхемы часов загружается текущее время, и работа прибора продолжается в нормальном режиме.

В результате проведенного анализа принято решение об аппаратной реализации функции календаря, так как стоимость микросхемы часов и ионистора гораздо ниже стоимости резервного источника с супервизором питания.

Следующий блок, который необходимо рассмотреть – это блок включения и выключения звонка.

Один способ реализации – с помощью электромагнитного реле. В этом случае управляющий сигнал с микроконтроллера подается на ключевой транзистор, который, в свою очередь, будет управлять обмоткой реле.

Второй вариант – применение симисторной схемы. Сигнал с микроконтроллера подается на оптосимистор, выход которого подключается к управляющему электроду мощного симистора.

Проведем анализ обоих вариантов: обе схемы обеспечивают гальваническую развязку силовой части от цифровой, обе

обеспечивают простое и надежное решение. Но ориентировочный расчет стоимости обоих вариантов показал, что вариант с симисторной схемой дешевле. Кроме того, в отличие от реле, схема будет работать бесшумно. Поэтому выбираем вариант с симисторной схемой.

Следующий блок, подлежащий рассмотрению – блок клавиатуры. Для выбора типа организации клавиатуры (линейной или матричной) нужно определить, сколько кнопок нужно и какие функции они должны выполнять.

Функции, выполняемые клавиатурой:

- корректировка времени, даты;
- принудительная подача звонка, в том числе сигнал «Тревога»;
- принудительное отключение звонка при необходимости.

С учетом этого можно предложить следующий вариант:

- кнопка «Коррекция». При нажатии кнопки устройство переходит в режим редактирования времени и даты. При повторном нажатии кнопки устройство переходит в обычный режим работы;
- кнопка «Выбор». Последовательное нажатие этой кнопки в режиме коррекции времени позволяет по-очереди редактировать все значения времени и даты;
- кнопки «Вперед» и «Назад». Эти кнопки в режиме коррекции позволяет изменять величину (на единицу вперед и назад соответственно), выбранную кнопкой «Выбор».
- кнопка принудительной (ручной) подачи звонка. При ее нажатии напрямую замыкается силовая цепь и подается звонок. Длительность звонка определяется длительностью удержания нажатой кнопки;
- выключатель с подсветкой для принудительного (ручного) отключения силовой цепи звонка.

Из вышесказанного следует, что первые четыре кнопки можно подключить напрямую к портам микроконтроллера.

Так как число кнопок небольшое, то можно реализовать линейный тип клавиатуры.

Следующий блок, подлежащий рассмотрению, это блок индикации.

Объем информации, подлежащий отображению в обычном режиме и режиме коррекции: текущее время и дата, номер текущего урока и номер текущей смены.

Наиболее оптимальным вариантом будет являться жидкокристаллический алфавитно-цифровой индикатор (ЖКИ) с организацией 2 строки на 16 знаков каждая. Он имеет встроенную подсветку, высокую контрастность выводимой информации и небольшое число линий для связи напрямую с микроконтроллером. Можно применить такого же типа индикатор, но не жидкокристаллический, а с матрицей на органических светодиодах (OLED), но его стоимость при прочих одинаковых параметрах будет в 2 раза выше ЖКИ.

Последний блок – это блок связи микроконтроллера с персональным компьютером. Он предназначен для загрузки сетки расписания в EEPROM память микроконтроллера. Здесь тоже можно рассмотреть несколько вариантов. Наиболее распространенными коммуникационными портами компьютера являются:

- параллельный порт LPT;
- последовательный порт RS-232;
- последовательный высокоскоростной порт USB.

Так как высокая скорость приемопередачи не требуется, то необходимо ориентироваться на простоту реализации протокола и его стоимость. Самым недорогим вариантом будет реализация протокола связи через LPT-порт, так как уровни этого порта совместимы с TTL-уровнями и позволяют организовать двунаправленный обмен данными в параллельном коде. Для соединения необходимо будет напрямую соединить один 8-битный порт микроконтроллера с портом LPT. Другой вариант – немного дороже – это соединение по последовательному порту

стандарта RS-232. Эля этого необходимо применить микросхему – преобразователь уровней (например MAX232 или аналоги). Эта микросхема соединяет стандартный порт USART микроконтроллера с COM-портом компьютера. Последний вариант – это реализация обмена данными через порт USB. В этом случае можно выбрать микроконтроллер, который содержит внутри контроллер USB или воспользоваться внешней микросхемой, реализующей протокол USB. Продолжая двигаться по линейке микроконтроллеров фирмы ATMEL, можно выбрать микроконтроллер AT89C5131 (семейства MSC-51) или AT90USB647 (семейство AVR). Розничная цена каждого микроконтроллера превышает 300 руб. Если выбирать отдельный контроллер USB, то можно после анализа рынка остановиться на микросхеме CP2101. Эта микросхема предлагает простое, недорогое (ее стоимость не превышает 100 руб.) и быстрое решение, которое не требует изучения стандарта USB и написание собственного драйвера USB-устройства для персонального компьютера. Микросхема подключается к стандартному USART-порту микроконтроллера и USB-порту компьютера. На компьютере устанавливается драйвер виртуального COM-устройства. Программное обеспечение с внешним устройством будет взаимодействовать через этот виртуальный порт как с обычным COM-устройством.

В результате проведенного анализа в качестве канала связи выберем порт USB, так как на современных портативных компьютерах, зачастую, кроме USB, никаких портов нет. А сетка расписаний устройства должна программироваться с любого компьютера. Так как написать программу для USART проще, чем для USB, то выберем тип контроллера USB – внешний (CP2101).

В итоге подсчитаем общее число занятых линий портов микроконтроллера.

4 линии порта – клавиатура из 4-х кнопок;

3 линии порта – служебные линии интерфейса с ЖКИ;

4 линии порта – информационные линии интерфейса с ЖКИ;

1 линия порта – выход на звонок;

2 линии порта – связь с микросхемой часов по шине I^2C;

1 линия порта – выход запроса прерывания

2 линии порта – интерфейс связи с компьютером;

1 линия порта – прием данных от пожарной сигнализации.

Итого: 18 линий

В результате можно выбрать микроконтроллер ATMega8. Он содержит RISC-ядро, у него 23 линии порта и низкая стоимость (около 50 руб.).

С учетом сказанного выше функциональная схема устройства примет следующий вид.

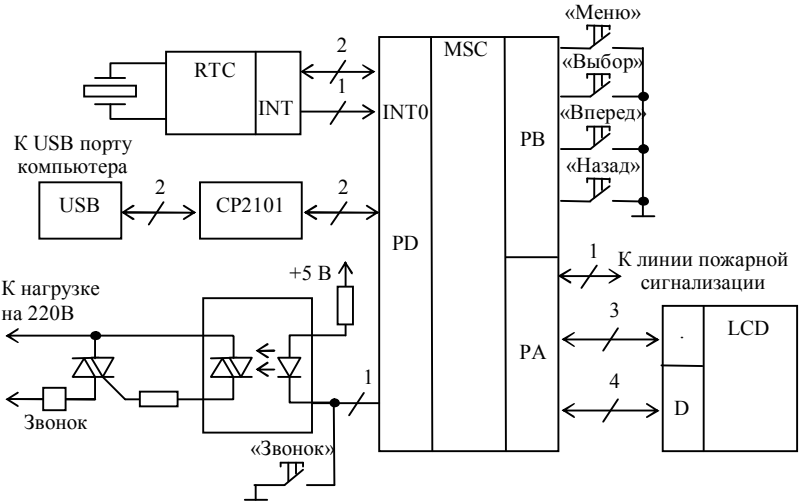


Рисунок 10.2. Функциональная схема устройства

10.5 Разработка блок-схемы алгоритма работы микроконтроллера

Анализируя задание на проектирование, можно утверждать, что сеток расписания должно быть как минимум две: обычная,

т.е. с понедельника по пятницу, и сокращенная, которая будет использоваться по субботам и в предпраздничные дни.

Основные функции, которые будет выполнять микроконтроллер:

- ежесекундно считывать текущее время и дату из микросхемы часов реального времени и выводить на индикатор;
- после вывода времени производить вычисление номера урока и номера смены и также выводить их на индикатор;
- ежеминутно проверять сетку расписания на совпадение времени и при совпадении давать звонок;
- непрерывно производить сканирование клавиатуры, т.е. производить проверку на нажатие какой-либо клавиши.

Дополнительные функции:

- установка связи с компьютером по коммуникационному порту;
- загрузка сетки расписания из компьютера в память микроконтроллера;
- проверка сигнальной линии, подключенной к пожарной сигнализации;
- принудительная подача звонка произвольной длительностью, соответствующей длительности нажатия клавиши «Звонок» (например «Тревога» или просто внеочередной звонок.

Блок-схемы алгоритмов приведены в приложении 4.

Приведем детализированное описание алгоритма работы устройства по представленным блок-схемам.

Основная программа начинает выполняться после аппаратного сброса микроконтроллера. Производится инициализация микроконтроллера: начальная загрузка регистров, настройка последовательного порта, прерываний, таймера, настройка регистров микросхемы часов, настройка направления портов, инициализация ЖКИ.

Основной цикл программы состоит из сканирования клавиатуры и обработки нажатия клавиш. При нажатии клавиши

«Меню» МК переходит в режим редактирования даты и времени. Вся эта информация отображается на ЖКИ. Нажимая кнопку «Выбор», можно перемещаться по дате и времени для редактирования конкретного числа. Редактирование чисел производится путем нажатия кнопок «Вверх» и «Вниз». При повторном нажатии клавиши «Меню» МК переходит в рабочий режим.

Все остальные действия выполняются только по прерываниям.

При нажатии кнопки «Звонок» замыкается непосредственно силовая цепь, и звонок звенит, пока нажата кнопка.

Микросхема часов настраивается таким образом, что каждую секунду выдает запрос прерывания на микроконтроллер (на внешний вывод INT0). Соответствующая подпрограмма обработки прерывания считывает текущее значение даты и времени из микросхемы часов и выводит эти значения на ЖКИ. Затем сравнивает это значение с временной сеткой в EEPROM. Если значения совпадают, то на соответствующий вывод порта подается активный уровень для включения звонка и затем запускается таймер T0 для формирования временной задержки в 3 секунды (длительность звонка). Таймер T0 переполняется определенное число раз и через 3 секунды подается команда на выключение звонка и остановку таймера.

Загрузку сетки расписания производит компьютерная программа. Используя виртуальный COM-порт, блоками по 64 байта данные загружаются в буфер, который формируется в ОЗУ МК. На каждый блок из 64 байт формируется контрольная сумма. После загрузки очередных 64 байт эти данные переписываются в EEPROM. Сразу напрямую переписывать данные в EEPROM не получается, так как данные поступают с большой скоростью, а каждый байт записывается в EEPROM за время, равное 8,5 мс (данные из документации на микроконтроллер). После того, как очередные 64 байта переписываются в EEPROM, МК выдает в компьютер

подтверждение завершения операции в виде контрольной суммы. Эта контрольная сумма сравнивается с контрольной суммой, сформированной компьютерной программой и при их совпадении в МК передаются следующие 64 байта сетки расписания.

10.6 Разработка принципиальной схемы

Сама схема электрическая принципиальная не будет приведена из-за маленького формата пособия. Однако ее можно будет получить у руководителя курса в электронном виде в формате Adobe Acrobat.

.

10.7 Разработка прикладной программы

Раздел не будет приведен по причинам многовариантности практической реализации программы и большого объема исходного кода.

10.8 Заключение

В данной работе было спроектировано микропроцессорное устройство управления звонком на занятия. Оно разработано с учетом опыта, накопленного при изучении аналогов и ориентировано на последующую коммерциализацию. Основными критериями выбора тех или иных компонентов были цена, простота монтажа, функциональность устройства. При продолжении работы над устройством особое внимание необходимо будет уделить надежности устройства, так как оно должно будет непрерывно функционировать в течение нескольких лет.

К недостаткам спроектированного устройства можно отнести отсутствие реальной связи с пожарной сигнализацией и невозможность изменения длительности подаваемого звонка. Т.е. длительность звонка на урок и с урока будут равны.

В целом же можно считать, что задача на проектирование выполнена полностью.

10.9 Список использованных источников

1. Русанов В.В., Шевелев М.Ю. Микропроцессорные устройства и системы. Учебное пособие. Томск: ТУСУР, 2007 г.

2. Шарапов А.В. Цифровая и микропроцессорная техника. Томск: Изд.ТГУ

3. Шарапов А.В. Микропроцессорные устройства и системы. Методическое указания к выполнению курсового проекта. Томск: Изд.ТГУ

4. Дж. Уитсон. 500 практических схем на ИС. Издательство МИР. 1992г.

5. А.А. Мячев. Персональные ЭВМ и микроЭВМ. Основы организации. Москва “Радио и связь” 1991г.

11 Примеры заданий на курсовое проектирование

1. Спроектировать устройство управления бетоносмесительным узлом. Компоненты бетонной смеси поочередно загружаются на весы и затем в смеситель, в котором перемешиваются 20±1 мин. Компоненты загружаются в диапазоне: гравий (0÷1000 кг ± 1%), песок (0÷300 кг ± 1%), цемент (0÷200 кг ± 1%), вода (0÷200 кг ± 1%). Пропорция компонентов задается предварительно. Загрузка компонентов управляется электромагнитными заслонками. В конце смены (8 часов) выдается для контроля следующая информация: номер замеса, текущее время замеса, вес и процентный состав замеса, количество замесов в смену и общий расход компонентов в смену.

2. Спроектировать устройство коррекции угла опережения зажигания ДВС в зависимости от частоты вращения коленвала. Датчик оборотов выдает 256 импульсов за 1 оборот коленвала. Датчик ВМТ выдает 1 импульс за 1 оборот.

3. Спроектировать устройство управления ЛПМ кассетного магнитофона. Устройство должно обеспечивать:

а) включение режима: стоп, воспроизведение, запись, перемотка влево (вправо), ускоренная перемотка влево (вправо);

б) индикацию режима;

в) ускоренная перемотка включается при нажатии соответствующих кнопок дольше 2 секунд;

г) управление электромагнитом каретки и тремя ДПТ приемного, падающего и ведущего узлов;

д) в режиме перемотки на двигатель поступает $U_{упр.} = 9$ В, ускоренной перемотки – 15 В, подтормаживания – выводы двигателя подключены к резистору R = 1 кОМ.

4. Спроектировать устройство управления домофоном для подъезда дома на 15 квартир.

Устройство должно обеспечивать тональный вызов и подключение громкоговорящей связи в квартире, номер которой

набран на кнопочной клавиатуре. Тональный вызов должен действовать в течение 1 мин., после чего на 5 мин. действует запрет на вызов. Через 5 минут после последнего нажатия любой клавиши, устройство переходит в дежурный режим. Устройство должно соединяться с квартирными пультами по двухпроводной линии.

5. Спроектировать устройство управления установкой индукционного нагрева сварных соединений стальных трубопроводов.

Устройство должно обеспечивать:

а) ввод скорости нагрева в диапазоне 0,5 ÷ 10 $^{\text{град.}}/_{\text{мин.}}$ с дискретностью 0,5 $^{\text{град.}}/_{\text{мин.}}$;

б) ввод скорости охлаждения с теми же параметрами;

в) ввод максимальной температуры до 995°С, с дискретностью 5°С;

г) ввод времени стабилизации максимальной температуры 10 мин ÷ 150 мин с дискретностью 10 мин.;

д) стабилизация температуры релейная с величиной гистерезиса 3°С;

е) ввод конечной температуры охлаждения, при которой установка выключается до 300°С с дискретностью 5°С;

ж) измерение температуры термопарой с величиной термо-ЭДС Е = 100 мкВ/°С;

з) после выключения установки выдается звуковой сигнал в течение 30 сек.

6. Спроектировать устройство формирования 3-х фазной последовательности синусоидального напряжения со следующими параметрами:

а) установка частоты в диапазоне f = 10÷ 500 Гц с дискретностью 5 Гц;

б) установка амплитуды выходного сигнала каждой фазы отдельно в диапазоне $U_{вых.}$ = 1÷10 В с дискретностью 1 В;

в) установка фазового сдвига отдельно для φ_{AB} и φ_{BC} в диапазоне φ = 110 ÷130 эл.град. с дискретностью 1 град.

7. Спроектировать устройство управления и охранной сигнализации салона автомобиля.

Устройство должно обеспечивать:

а) блокировку дверей;

б) управление стеклоподъемниками;

в) сигнализацию открытой двери;

г) сигнализацию открытого капота или багажника;

д) блокировку включения двигателя;

е) срабатывание световой и звуковой сигнализации при открывании любой двери, багажника, капота, при качании автомобиля в течение 3 секунд.

8. Спроектировать устройство вывода на индикаторную панель 10*100 элементов сообщения в виде “бегущей строки”, набираемого на клавиатуре. Объем сообщения – 128 байт. Периодичность вывода – 5 минут.

9. Спроектировать устройство управления мини-АТС на 4 номера с одной городской линии. Выбор осуществляется путем добавления к номеру линии одной из четырех цифр 0÷3 после соединения с линией.

10. Спроектировать устройство, вычисляющее активную, реактивную и полную мощность, потребляемую из однофазной сети переменного тока в диапазоне до 1кВ·А±1%.

11. Спроектировать экзаменатор для ГАИ, контролирующий ответы на 10 вопросов в билете. Каждый вопрос выбирается случайным образом из банка на 30 вопросов.

При ответе на 8 и более вопросов выдается надпись “сдал”, “не сдал”.

12. Спроектировать устройство управления процессом химического производства продукта A из компонентов B, C и D.

Устройство должно обеспечивать:

а) загрузку при помощи э/м заслонки и взвешивание $M_в = 0÷250$ кг ± 1% компоненты B;

б) загрузку компоненты C ($M_в = 0÷50$ кг±1%) со скоростью $V_c = 1÷10$ кг/мин. ±10% при непрерывном перемешивании смеси;

в) нагревание смеси B + C до T = 50÷250 °C±5% со скоростью $V_т$ = 1÷10 °C/мин.±10%;

г) загрузку компоненты D (M_D = 0÷50 кг±1%);

д) стабилизацию давления P = 10÷20 кг $^{с}/_{см^2}$ с дискретностью ΔP = 0,5 кг $^{с}/_{см^2}$ и температуры по пункту (в) в течении t = 20÷60 мин.±10%.

13. Спроектировать устройство управления светофором на перекрестке главной и второстепенной дороги.

Устройство обеспечивает постоянный разрешающий сигнал на главной дороге, а на второстепенной дороге разрешается проезд, если скопилось более 5 автомобилей или через 3 минуты после поступления первого автомобиля. С $21^{оо}$ до $6^{оо}$ светофор находится в режиме мигающего желтого сигнала.

14. Спроектировать задающий генератор градиентных импульсов ЯМР (ядерно-магнитный резонанс) – томографа, обеспечивающий формирование трапециидальных импульсов:

а) период T = 1÷100 Гц±0,1Гц;

б) фронт и срез импульса линейные и длительность их устанавливается раздельно t = 0,5÷1,5 мС ±1%;

в) вершина импульса плоская;

г) амплитуда изменяется в пределах U_m = – 10 В ÷ +10 В с дискретностью dU = 0,1 В;

15. Спроектировать устройство управления э/печью для обжига керамической плитки. Электропечь имеет 8 зон нагрева. В каждой зоне контроль температуры осуществляется термопарой с термо–ЭДС Ет = 100 мкВ/ºС в диапазоне T = 200÷1200 ºС с точностью dT = 1%. Нагрев в каждой зоне регулируется тиристорным регулятором, регулирующим ток в резистивной нагрузке.

Устройство выдает на самописец значение температуры в каждой зоне каждые 10 минут в течение суток.

16. Спроектировать систему дистанционного управления моделью автомобиля.

Автомобиль должен выполнять следующие действия:

а) движение вперед;

б) движение назад;

в) режим “стоп”;

г) включение головного освещения;

д) включение звукового сигнала;

е) пропорциональное управление поворотом управляющих колес.

17. Спроектировать устройство управления рабочим циклом литьевой машины. Рабочий цикл включает смыкание форм, подвод механизма впрыска, впрыск (Т1), формование (Т2), отвод механизма впрыска, загрузку (Т3), охлаждение (Т4), размыкание форм и выталкивание изделия. Пауза между циклами - Т5. Временные интервалы Т1, Т2, Т3, Т5 - до 99 с, Т4 - до 9999 с.

18. Спроектировать счетчик потребляемой тепловой энергии. Контролируется объем потребляемой горячей воды и разность температур в трубах горячей и холодной воды.

19. Разработать электронное устройство управления инкубатором. Точность задания и стабилизации температуры - 0,1 градуса. Через каждый час обеспечить изменение положения яиц путем поворота на 45 град. Предусмотреть цифровую индикацию температуры. Для аналого-цифрового преобразования использовать метод последовательных приближений.

20. Спроектировать измеритель частоты вращения ротора двигателя. Диапазон измерения (100-1000 об/мин). Импульсный датчик вырабатывает 16 импульсов за каждый оборот. Время измерения - не более трех оборотов ротора.

Литература

Основная литература.

1. Русанов В.В., Шевелев М.Ю. Микропроцессорные устройства и системы. Учебное пособие для вузов. – Томск: ТУСУР – 2006. – 200 с.: ил.

2. Микропроцессорные системы: Учебное пособие для вузов/ Е.К. Александров и др.; Под общей ред. Д.В. Пузанкова. – СПб.: Политехника, 2002. – 935 с.: ил.

3. Шарапов А.В. Микропроцессорные устройства и системы. Методические указания к выполнению курсового проектирования. – Томск: ТУСУР – 2004.

4. Рождественский Д.А. Микропроцессорные устройства в системах управления: Учебное пособие. – Томск: Томский межвузовский центр дистанционного образования, 2003. – 130 с.

5. Белов А.В. Конструирование устройств на микроконтроллерах – СПб.: Наука и Техника, 2005. – 256 с.: ил.

Дополнительная литература.

6. Учебный стенд SDK 1.1. Руководство пользователя. Москва, 2001.

7. Проектирование цифровых устройств на однокристальных микроконтроллерах\ В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева.- М.: Энергоатомиздат, 1990.

8. Белов А. М., Иванов Е. А., Муренко Л. Л. Средства автоматизации программирования микропроцессорных устройств. / Под ред. Домрачева В. Г. – М.: Энергоатомиздат, 1988.

9. Домнин С. Б., Иванов Е. А., Муренко Л. Л. Средства комплексной отладки микропроцессорных устройств. / Под ред. Домрачева В. Г. – М.: Энергоатомиздат, 1988.

10. Уильямс Г. Б. Отладка микропроцессорных систем. / Под ред. Сташина В. В. – М.: Энергоатомиздат, 1988.

11. AT89S51 Datasheet. Atmel CO LTD. USA, 1998.

12. ATMega16. Datasheet. Atmel CO LTD. USA, 2001.

13. Www.atmel.ru/doc

14. http://www.telesys.ru/electronics/projects.php

Приложение 1

Пример оформления титульного листа

Федеральное агентство по образованию

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра промышленной электроники (ПрЭ)

МИКРОПРОЦЕССОРНОЕ УСТРОЙСТВО УПРАВЛЕНИЯ ЗВОНКОМ НА ЗАНЯТИЯ

Пояснительная записка к курсовому проекту по дисциплине “Микропроцессорные устройства и системы”

ФЭТ КП.XXXXXX.006 ПЗ

Студент группы 364-1
Камаев В.Н.
Дата

Руководитель проекта
профессор кафедры ПрЭ
_____ _____

2012 г.

Приложение 2

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра промышленной электроники (ПрЭ)

ЗАДАНИЕ
на курсовое проектирование по дисциплине
“Микропроцессорные устройства и системы”

студенту **Камаеву В.Н.**

группа **364-1,** факультет **электронной техники**

Тема проекта: **Микропроцессорное устройство управления звонком на занятия**

Исходные данные к проекту:

1) организация подача звонков на занятия в институте
2) возможность коррекции программы под данную ситуацию

Содержание пояснительной записки (перечень подлежащих разработке вопросов): выбор микроконтроллера, обоснование функциональной схемы, разработка полной функциональной схемы устройства с перечнем элементов и листинга управляющей программы

Перечень графического материала (с точным указанием обязательных чертежей и схем):

схема электрическая принципиальная - 1 л.

Дата выдачи задания:

_____

Руководитель

_____

Профессор кафедры ПрЭ _____ _____(подпись)

Задание принял к исполнению _____

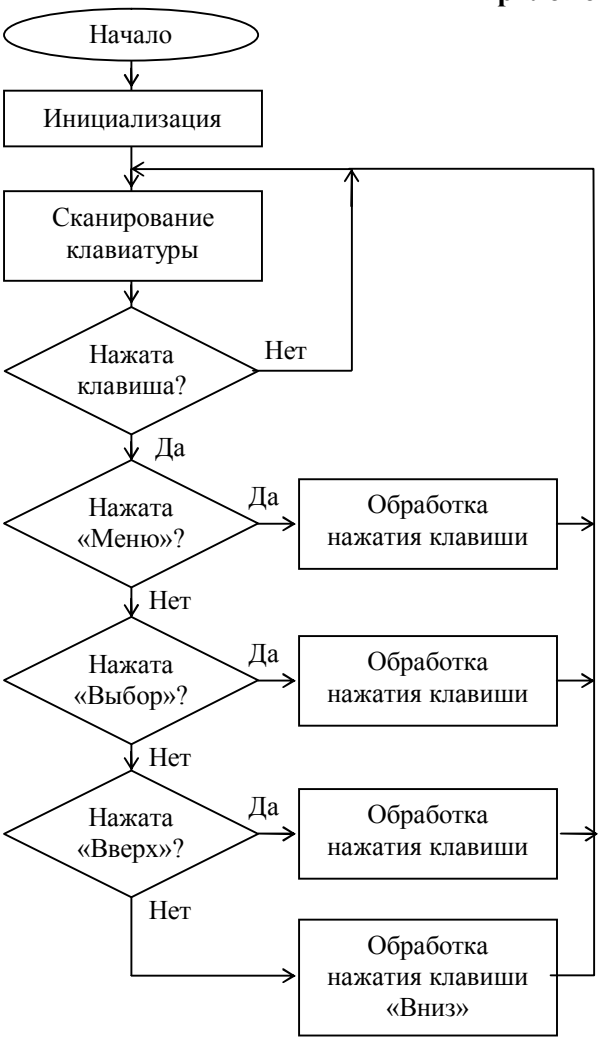
(дата) _____ (подпись студента)

Приложение 3

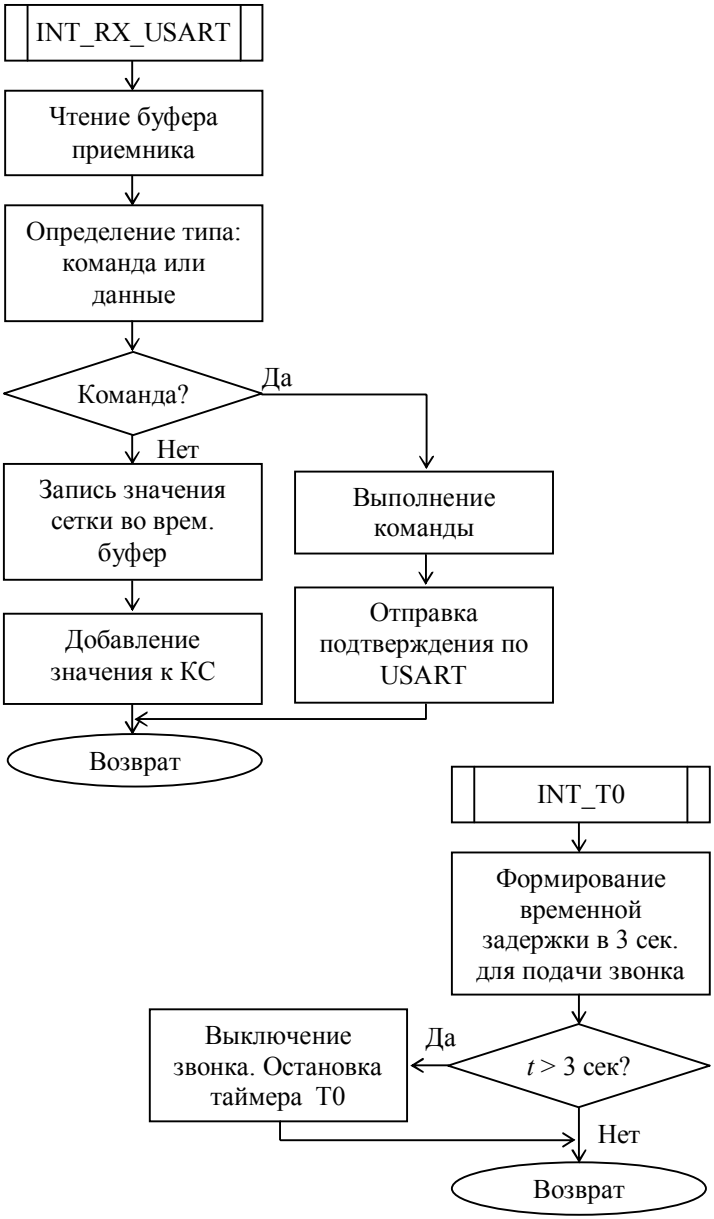
Пример оформления оглавления

Оглавление

Приложение 4



Продолжение приложения 4.



Окончание приложения 4.