

Аннотация

В данной работе рассматривается процесс разработки учебного модуля с необходимым прикладным программным обеспечением. Целью работы было дополнить и усовершенствовать техническую базу кафедры вычислительной техники устройством разработки и отладки микропроцессорных средств и систем; предоставить возможность студентам получить практические навыки по программированию таких систем. В главе «Постановка задачи» сформулированы требования к изделию и проведен их предварительный анализ. В аналитической части проведен анализ рынка выпускаемых промышленностью 8-разрядных микроконтроллеров, программных и аппаратных средства разработки. В главе «Проектно-конструкторская часть» описана разработка протокола обмена информацией по последовательному порту для прошивки программы в учебный модуль. Особое внимание уделено проектированию программной части. Руководство пользователя и отчет по практическому применению модуля в учебном процессе (в ходе выполнения курсовых работ студентами 3 курса) приведены в главе «Экспериментальная часть». Выполнена экономическая оценка проекта и рассмотрены вопросы безопасности и экологичности проекта.

Summary

Содержание

Аннотация.....	4
Summary.....	5
Содержание.....	6
Введение.....	8
1. Постановка задачи.....	10
1.1. Требования к аппаратной части учебного модуля.....	10
1.2. Требования к прикладному программному обеспечению.....	12
2. Аналитическая часть.....	14
2.1. Обзор рынка 8-разрядных микроконтроллеров.....	14
2.1.1. Семейство MCS-51.....	14
2.1.2. Семейство PIC.....	15
2.1.3. Семейство AVR.....	15
2.1.4. Семейство ADuC.....	17
2.1.5. Семейство PSoC.....	17
2.1.6. Сводная таблица по 8-разрядным микроконтроллерам.....	18
2.2. Обзор современных аппаратных средств разработки для 8-разрядных микроконтроллеров.....	19
2.2.1. STK-AT89C5130 программатор-отладчик.....	19
2.2.2. Отладочная плата STK-500.....	21
2.2.3. ME-UNI-DS3. Универсальная отладочная система на базе МК разных производителей.....	23
2.2.4. Сводная таблица по средствам разработки.....	26
2.3. Обзор современных программных средств разработки для 8-разрядных микроконтроллеров.....	26
2.3.1. Keil μ Vision (Keil C).....	27
2.3.2. AVR-Studio.....	27
2.3.3. PonyProg.....	28
2.3.4. Сводная таблица по программным средствам разработки.....	29
2.4. Выводы по главе.....	29
3. Проектно-конструкторская часть.....	30
3.1. Разработка аппаратной части.....	30
3.1.1. Описание базового модуля.....	30
3.1.2. Описание дополнительных модулей.....	32
3.2. Разработка программной части.....	33
3.2.1. Разработка протокола загрузки программы пользователя для MCS-51.....	34

3.2.2. Разбор файла формата HEX.....	37
3.2.3. Разработка встраиваемого программного обеспечения.....	42
3.2.4. Разработка программы для компьютера.....	47
3.3. Руководство программиста.....	48
3.4. Выводы по главе.....	55
4. Экспериментальная часть.....	57
4.1. Руководство пользователя.....	57
4.1.1. Введение.....	57
4.1.2. Назначение и условия применения.....	58
4.1.3. Подготовка к работе.....	58
4.1.4 Описание операций.....	62
4.1.5. Аварийные ситуации.....	68
4.1.6. Рекомендации по освоению.....	69
4.2. Применение модуля в учебном процессе.....	69
4.3. Улучшение по результатам эксплуатации модуля.....	71
4.4. Выводы по экспериментальной части.....	72
5. Организационно-экономическая часть.....	73
5.1. Цель создания учебного модуля.....	73
5.2. Описание программного изделия.....	74
5.3. Обзор рынка сбыта и аналогичной продукции.....	74
5.4. Организация и планирование процесса разработки ПП.....	75
5.4.1. Формирование состава выполняемых работ.....	75
5.4.2. Расчет трудоемкости выполнения работ.....	77
5.5. Оценка стоимости изделия.....	79
6. Безопасность и экологичность проекта.....	89
6.1. Характеристика персонального компьютера как технического источника электромагнитного поля.....	89
6.2. Воздействие электромагнитного поля на пользователя персонального компьютера.....	91
6.3. Проведение электромагнитной мониторинга на рабочем месте пользователя персонального компьютера.....	96
6.4. Аттестация рабочего места пользователя персонального компьютера по электромагнитной безопасности.....	97
6.5. Расчет искусственного освещения производственного помещения с ПК.....	98
Заключение.....	101
Литература.....	102
Приложение.....	104

Введение

Стремительное развитие вычислительной техники налагает высокие требования к подготовке специалистов университетами. Одной из основных проблем большинства студентов – это отсутствие опыта работы с современными инструментами проектирования и недостаточное понимание этапов и методов разработки. Такое положение дел особенно сильно затрагивает практическую сторону области программирования микроконтроллерных устройств. На кафедре вычислительной техники Чувашского Государственного Университета имени И. Н. Ульянова до настоящего времени отсутствовали технические средства разработки и отладки микропроцессорных систем. Трудно, а иногда и практически невозможно проверить результаты проделанной работы. Только теоретически, исходя из поведения программы в отладчике с пошаговой трассировкой, можно судить о правильности выполнения алгоритма. Для решения этой проблемы и была выбрана тема дипломного проектирования.

Современный рынок средств разработки и отладки микропроцессорных устройств предлагает огромное число отладочных модулей. Их многообразие связано с большим количеством серий микроконтроллеров и периферийных микросхем к ним, решающих задачи широкого круга. Значительная часть предлагаемых устройств направлена на решение узкоспециализированных задач. Такой подход не применим в образовательных целях в связи с разными вариантами заданий. Универсальные продукты содержат решение нашей задачи, но и они не подходят к требованиям учебного процесса, основными из которых являются низкая стоимость и высокая конфигурируемость системы, схожая по своим свойствам к детскому конструктору. Все вышесказанное показывает актуальность этой темы и разработки изделия, удовлетворяющая по своим характеристикам, соответствующим требованиям.

Разработанный нами в ходе дипломного проектирования на результатах, достигнутых при подготовке к конференции 2008 года, учебный модуль дает возможность студентам на практике просмотреть результаты выполнения программы, а в случае необходимости отладить ее. Учебный модуль, состоящий из базового и дополнительных модулей, представляет собой микропроцессорное устройство с возможностью сборки различных электронных устройств. В состав модуля входят устройства индикации (семисегментные индикаторы, разные виды жидкокристаллических дисплеев, линейка светодиодов), ввода информации (клавиатура, дискретные датчики), связи с компьютером по коммуникационному порту (конвертер уровней TTL-RS232), коммутации (перенаправитель портов, выход стандартных разъемов на клеммники и на PC/2) и готовые к работе

функциональные микросхемы (часы реального времени, термометр). Этим модулям хватит на моделировании большинства распространенных бытовых и промышленных устройств.

Разрабатываемое программное обеспечение выполняет решение задач по обслуживанию учебного модуля. Первая решаемая задача – это непосредственное обслуживание базового модуля через интерфейс персонального компьютера. Сюда входит идентификация микросхемы внешней памяти программ, стирание, чтение и запись flash-памяти, а также проверка содержимого памяти. Такая возможность позволяет нам отказаться от использования внешнего дорогостоящего программатора. Вторая задача – проверка работоспособности основных частей учебного модуля перед их использованием.

При проектировании программной и аппаратной части были заложены требования, которые позволят превзойти некоторые существующие продукты на рынке и применять его в учебном процессе по дисциплине «Микропроцессорные средства и системы».

1. Постановка задачи

Учебный модуль состоит из двух связанных между собой частей — аппаратной и программной. В аппаратную часть учебного модуля входит центральная плата, называемая базовым модулем, и дополнительные платы. Базовый модуль имеет в своем составе микроконтроллер с обслуживающими его цепями и дополнительными микросхемами памяти. Назначение этого модуля — выполнять программу пользователя и управлять подключенными к нему внешними устройствами. Дополнительные модули являются как раз этими самыми внешними устройствами (по отношению к микроконтроллеру). Их количество влияет на спектр моделируемых устройств, что дает нам разные варианты задания для курсовых работ. Программное обеспечение необходимо для обслуживания аппаратной части учебного модуля. Использование внешней памяти программ с возможностью пятивольтовой записи (согласованность с уровнями ТТЛ сигналов) позволяет записывать в него данные непосредственно через микроконтроллер. Задача программы — передать данные с персонального компьютера на базовый модуль микроконтроллеру для его записи во внешнюю память программ. Как следует из сказанного — необходимо иметь программу на стороне компьютера и базового модуля. Интерфейс для связи выбран последовательный коммуникационный порт СОМ компьютера. Распространенность в промышленности и легкость программирования сыграли основную роль при его выборе. Проконтролировать визуально работоспособность всего изделия никак невозможно. Для проверки учебного модуля нужен пакет небольших программ, эксплуатирующих дополнительные модули. При усомнении в целостности какого-нибудь модуля можно проверить с помощью этих программ.

Порядок проектирования аппаратного и программного обеспечения проводится совместно. На алгоритм работы программы влияет применяемый тип микросхемы flash-памяти. Выбор типа зависит от наличия на российском рынке и их доступность физическим лицам.

К устройству предъявляются высокие требования по части применимости его в образовательном процессе. В следующих разделах определены требования к учебному модулю — отдельно к программному и аппаратному обеспечению.

1.1. Требования к аппаратной части учебного модуля

Основными требованиями, предъявляемыми к аппаратному обеспечению учебного модуля:

1. Низкая себестоимость. Низкая себестоимость позволяет сделать доступным изделие для самостоятельной сборки и для бюджета университета. В общем случае уменьшение цены достигается за счет отказа от внешнего программатора. Его заменит разрабатываемое к модулю программное обеспечение. При учетывании того факта, что стоимость многих электронных компонент при покупке в розничной сети намного выше, чем для оптовых партий, снижение цены в конечном случае можно достичь при массовой сборке.

2. Электрическая и механическая безопасность. Электрическая безопасность достигается применением в изделии сертифицированного блока питания на стандартное пятивольтовое напряжение. Это исключает самостоятельное его изготовление и упрощает разработку модуля. Для защиты нижней стороны плат базового и дополнительных модулей применено оргстекло прозрачного цвета. Соответствующая сторона прикрывается им. Таким образом, исключается случайное замыкание проводников металлическими предметами или между модулями между собой.

3. Повышенный срок эксплуатации. Большое количество циклов перезаписи увеличивает ресурс изделия (или только flash-памяти программ). Его увеличение достигается путем использования flash-памяти соответствующего типа. Внутренняя память программ микроконтроллера AT89S52 позволяет перезаписать программу, как заявлено в документации, не менее 1000. Этого не достаточно для полноценного экспериментирования с программой. Количество циклов перезаписи для микросхемы SST29EE010 составляет 100000, что уже дает нам увеличение в 100 раз. При предположении, что студент за одно занятие перезапишет память 20 раз при количестве занятий 2 раза в неделю и 18 учебных недель, общее количество перезаписи составит 720. При ресурсе 1000 циклов этого хватило бы только на полтора семестра. В нашем же случае об этом можно и не волноваться.

4. Расширяемость. Независимость дополнительных модулей от базового и друг от друга дает нам возможность дополнять уже готового изделия новыми модулями. Это достигается стандартизацией связей между модулями. При изготовлении очередного модуля следует придерживаться только стандартов разъема, участвующих в связи с базовым модулем. Таким же способом можно дополнить комплект с базовым модулем другого микроконтроллера, например PIC.

5. Модульность. Однородность связей между модулями, а также отдельные платы модулей, позволяют переконфигурировать систему по своему усмотрению. Можно собрать калькулятор (клавиатура + ЖК-индикатор) или часы (микросхема часов реального времени + семисегментный индикатор + клавиатура) и тому подобное.

6. Высокая ремонтпригодность. Микросхемы памяти на плату не запаены. Вместо них туда запаены соответствующие колодки DIP. Это удобно для оперативной замены и

проверки микросхем. Такое решение позволяет оставить плату при выходе из строя микросхем и быстро найти причину возникновения ошибки.

7. Удобство эксплуатации. Переход от программирования к проверке программы должен быть простым и быстрым. Отсутствие возможности перепутать контакты при подключении внешних устройств не только уменьшает вероятность выхода из строя модулей, но и исчезает постоянный контроль своих действий. Для этого следует использовать безопасные разъемы — разъемы с ключами.

Все требования никак не должны отражаться на наглядности и удобстве использования. Компоненты модулей должны быть снабжены функциональными и позиционными обозначениями, указывающие соответствие между принципиальной схемой и реальным изделием. Компонентов, по возможности, необходимо выделять в группы по выполняемым функциям. Такими группами могут быть управляющие, указывающие состояние, коммутационные и основные элементы.

1.2. Требования к прикладному программному обеспечению

Программное обеспечение тесно связано с используемым аппаратным обеспечением. Поэтому его проектирование должно проводиться параллельно, опираясь на ранее принятые решения. Само по себе, без учебного модуля, программа использоваться не может. Так как готовое изделие ориентировано на свободное распространение среди кафедр и образовательных учреждений, ориентироваться только на бесплатные инструменты разработки и библиотеки. Это позволит без нарушения законодательства использовать его в учебном процессе.

Весь программный комплекс состоит из трех частей.

1. Программа на стороне персонального компьютера.
2. Программа на стороне базового модуля.
3. Пакет программ для тестирования изделия.

Первая программа имеет графический интерфейс и является основным элементом при работе с модулем. Она выполняет все функции по прошивке программы пользователя в базовый модуль. Разбор файлов в формате HEX; отправка и прием пакетов команд по работе с памятью flash — идентификация микросхемы, стирание, чтение и запись данных; протоколирование выполненных команд. Это действия программы во время режима программирования. Кроме этого существует режим выполнения. При этом программа находится в постоянном ожидании данных от модуля, прерываясь только на запросы

отправки пользовательской информации. Такая функция удобна для отладки программ, работающих с последовательным портом. При этом должна быть команда пользовательской настройки СОМ порта — скорости, размера данных и стоповых бит, контроля четности и остальных параметров интерфейса последовательного порта.

Программа на микроконтроллер базового модуля выполняет функции ожидания пакетов команд от персонального компьютера, их разбор, выполнением необходимых действий и отправкой результатов действий. Требованиями к этой программе являются быстроедействие и надежность при выполнении команд. Протокол обмена данными между компьютером и модулем должен иметь функции контроля целостности.

Пакет программ на микроконтроллер ориентирован на целевой микроконтроллер. Он должен охватывать как можно больше дополнительных устройств. В его задачи входят проверка базового и дополнительных модулей. При усомнении в работоспособности отдельных частей модуля, есть возможность быстрой проверки и локализации неисправности.

2. Аналитическая часть

В главе рассматриваются существующие готовые решения, проводится анализ их функциональности. Оцениваются методы решения задач, их эффективность и применимость для нашего случая. Целью обзора является разработка окончательного варианта требований к проектируемому продукту.

2.1. Обзор рынка 8-разрядных микроконтроллеров

Из-за широкого распространения разнообразных функциональных приборов и устройств в быту и промышленности, есть огромный спрос на микроконтроллеры. В связи с техническим прогрессом появились и 16/32-разрядные микросхемы с высокими скоростями выполнения программ, большими объемами памяти. Но все же такими же популярными остаются 8-разрядные небольшие микроконтроллеры, применяемые в различных областях техники. Их возможностей хватает на решение многих задач.

В настоящее время рынок предлагает модели микроконтроллеров от различных производителей, конкурирующих между собой. У каждой модели есть свои достоинства и недостатки.

2.1.1. Семейство MCS-51

Семейство однокристальных микроконтроллеров MCS-51 было представлено фирмой Intel на рынке в начале 1980-х годов. Свободное распространение лицензии на ядро позволило получить широкую популярность устройства по всему миру. Дальнейшее развитие пошло по пути наращивания встроенных периферийных возможностей. Большую роль в поддержании популярности сыграло создание коммерчески пригодного к использованию кристалла энергонезависимой flash-памяти фирмой Intel в 1988 году. Ведущими производителями на рынке микроконтроллеров C51-архитектуры являются фирмы Philips, Atmel, Maxim/Dallas, Cygnal и множество других. Аналоги 8051 выпускали также советские предприятия в Минске, Киеве, Воронеже, Новосибирске.

В настоящее время микроконтроллеры семейства MCS-51 сдают позиции микроконтроллерам AVR и PIC.

2.1.2. Семейство PIC

Микроконтроллеры серии PIC отличаются низкой стоимостью и простой системой команд. По этим причинам продукция компании MICROCHIP завоевала основную часть рынка 8-разрядных микроконтроллеров.

Система команд базового семейства PIC165x содержит 33 команды. Все команды, кроме команд перехода, выполняются за один машинный цикл (четыре машинных такта) с перекрытием по времени выборок команд и их исполнения. При этом достигается производительности 5 MIPS при тактовой частоте 20 МГц. Симметричная система команд позволяет выполнять операции с любым регистром, используя любой метод адресации. Пользователь может сохранять результат операции или в аккумуляторе или во втором регистре, используемом для операции.

В настоящее время MICROCHIP выпускает пять основных семейства 8-разрядных RISC-микроконтроллеров, совместимых снизу вверх по программному коду.

1. Базовое семейство PIC15Cx с 12-разрядными командами, простые недорогие микроконтроллеры с минимальной периферией.
2. PIC12Cxxx с 12-разрядными командами со встроенным тактовым генератором, выпускаемые в миниатюрном 8-выводном исполнении.
3. Mid-range PIC16x/7x/8x/9x с 14-разрядными командами. Наиболее многочисленное семейство, объединяющее микроконтроллеры с разнообразными периферийными устройствами.
4. High-end PIC17C4x/5xx высокопроизводительные микроконтроллеры с расширенной системой команд 16-разрядного формата, работающие на частоте до 33 МГц, с объёмом памяти программ до 16 Кслов. Кроме обширной периферии почти все микроконтроллеры этого семейства имеют встроенный аппаратный умножитель 8x8, выполняющий операцию умножения за один машинный цикл.
5. PIC18Cxxx микроконтроллеры имеют расширенное RISC-ядро, оптимизированное под использование Си-компилятора, адресное пространство программ до 2 Мбайт, до 4 Кбайт встроенной памяти данных и производительность 10 MIPS.

2.1.3. Семейство AVR

Настоящая революция в мире микроконтроллеров произошла в 1996 году, когда корпорация Atmel представила свое семейство чипов на новом прогрессивном ядре AVR. Более продуманная архитектура AVR, быстродействие, превосходящее контроллеры

Microchip, привлекательная ценовая политика способствовали популяризации продукции данного семейства.

AVR-архитектура, на основе которой построены микроконтроллеры семейства AT90S, объединяет мощный гарвардский RISC-процессор с отдельным доступом к памяти программ и данных, 32 регистра общего назначения, каждый из которых может работать как регистр-аккумулятор, и развитую систему команд фиксированной 16-бит длины. Большинство команд выполняются за один машинный такт с одновременным исполнением текущей и выборкой следующей команды, что обеспечивает производительность до 1 MIPS на каждый МГц тактовой частоты.

32 регистра общего назначения образуют регистровый файл быстрого доступа, где каждый регистр напрямую связан с АЛУ. За один такт из регистрового файла выбираются два операнда, выполняется операция, и результат возвращается в регистровый файл. АЛУ поддерживает арифметические и логические операции с регистрами, между регистром и константой или непосредственно с регистром. Регистровый файл также доступен как часть памяти данных. 6 из 32-х регистров могут использоваться как три 16-разрядных регистра-указателя для косвенной адресации. Старшие микроконтроллеры семейства AVR имеют в составе АЛУ аппаратный умножитель.

Базовый набор команд AVR содержит 120 инструкций. Инструкции битовых операций включают инструкции установки, очистки и тестирования битов. AVR-архитектура оптимизирована под язык высокого уровня Си. Кроме того, все кристаллы семейства совместимы "снизу вверх".

Все микроконтроллеры AVR имеют встроенную FLASH ROM с возможностью внутрисхемного программирования через последовательный 4-проводной интерфейс. Многие чипы имеют функцию самопрограммирования.

В рамках единой базовой архитектуры AVR-микроконтроллеры подразделяются на три подсемейства.

1. Classic AVR — основная линия микроконтроллеров с производительностью отдельных модификаций до 16 MIPS, FLASH ROM программ 2–8 Кбайт, EEPROM данных 64–512 байт, SRAM 128–512 байт.

2. Mega AVR с производительностью 4–6 MIPS для сложных приложений, требующих большого объема памяти, FLASH ROM программ 64–128 Кбайт, EEPROM данных 64–512 байт, SRAM 2–4 Кбайт, SRAM 4 Кбайт, встроенный 10-разрядный 8-канальный АЦП, аппаратный умножитель 8x8.

3. Tiny AVR — низкостоимостные микроконтроллеры в 8-выводном исполнении имеют встроенную схему контроля напряжения питания, что позволяет обойтись без внешних супервизорных микросхем.

AVR-микроконтроллеры поддерживают спящий режим и режим микропотребления. В спящем режиме останавливается центральное процессорное ядро, в то время как регистры, таймеры-счётчики, сторожевой таймер и система прерываний продолжают функционировать. В режиме микропотребления сохраняется содержимое всех регистров, останавливается тактовый генератор, запрещаются все функции микроконтроллера, пока не поступит сигнал внешнего прерывания или аппаратного сброса.

В зависимости от модели, AVR-микроконтроллеры работают в диапазоне напряжений 2,7–6 В либо 4–6 В (исключение составляет ATtiny12V с напряжением питания 1,2 В).

2.1.4. Семейство ADuC

Микросхема ADuC8xx — интегральная система сбора информации, включающая в себя прецизионный многоканальный АЦП с самокалибровкой, прецизионный ЦАП и программируемое 8-битное микропроцессорное ядро, совместимое с 8051. Отличительной особенностью семейства микросхем серии ADuC8xx является сам их принцип построения. Эти микросхемы не являются микроконтроллером со встроенными АЦП-ЦАП. Они представляют собой удачно скомбинированные АЦП и ЦАП со встроенным в них микроконтроллером и флэш-памятью. Поэтому их основным достоинством является высокая точность аналого-цифрового (до 24 бит, при низких, до 100 Гц, частотах) и цифро-аналогового преобразования, удачно сочетаемая с возможностью непосредственной обработки получаемой информации.

Микроконтроллер представляет собой «стандартное» ядро 8051 с максимальной рабочей частотой 16 МГц (12 МГц — типовая), тремя байтовыми портами ввода/вывода, один из которых, порт 3, обладает повышенной нагрузочной способностью, тремя 16-разрядными таймерами/счетчиками и расширенной периферией.

2.1.5. Семейство PSoC

Микросхемы PSoC фирмы Cypress представляют собой 8-битный микроконтроллер, содержащий микропроцессорное ядро и массив цифровых и аналоговых блоков, позволяющий реализовывать необходимые пользователю периферийные функции, как

вполне стандартные, например ШИМ, АЦП или UART, так и такие необычные для микроконтроллеров, как, например, аналоговые фильтры и инструментальные усилители.

Одной, во всех отношениях примечательной и полезной особенностью PSoC является возможность динамического переконфигурирования. Сразу, во время работы, по какому-нибудь внешнему или внутреннему событию, внутренняя организация прибора изменяется, и он начинает выполнять совершенно иные функции. Для переконфигурирования PSoC достаточно изменить значения специальных динамических регистров, определяющих внутреннюю текущую архитектуру прибора. Биты управления расположены в пространстве ввода-вывода ядра, и поэтому конфигурация аналоговой и цифровой периферии представляет запись соответствующих значений в эти регистры во время выполнения программы.

Цифровые блоки PSoC делятся на два типа: основные (basic) и коммуникационные (communication). Основные блоки включают: таймеры, счетчики, ШИМ, генераторы псевдослучайной последовательности, генератор CRC (Cyclic Redundancy Check), генератор зоны нечувствительности (для ШИМ). В коммуникационные блоки входят: интерфейс SPI (режимы Master/Slave), интерфейс I2C, инфракрасный порт, UART. Аналоговые PSoC блоки могут включать: усилители с программируемым коэффициентом усиления, компараторы, программируемые фильтры (низких частот, высоких частот, полосовые и т.д.), АЦП (накапливающие, сигма-дельта, последовательного приближения), ЦАП. Интересным дополнительным свойством является наличие SMP (Switch Mode Pump) — механизма, позволяющего PSoC работать от напряжения питания 1 В.

Центральной частью системы является 8-разрядное ядро M8C с тактовой частотой до 24 МГц (4 MIPS) при питании напряжением 5 В.

Благодаря тому, что PSoC позволяют сократить количество используемых внешних компонентов, это существенно упрощает процесс разработки, удешевляет устройство и одновременно повышает его гибкость за счет возможности перепрограммирования в системе или реконфигурирования внутренней структуры прямо в процессе работы.

2.1.6. Сводная таблица по 8-разрядным микроконтроллерам

В сводной таблице 1 показан обобщенный результат по разделу 2.1. «Обзор рынка 8-разрядных микроконтроллеров». Энергопотребление очень сильно зависит от режима работы и частоты тактирования. При спящем режиме потребление падает до десятков микроампер и поэтому не рассматривается. Потребляемый ток приводится для активного режима работы и

номинальной частоты микроконтроллера по документации. Строка АЦП/ЦАП указывает на присутствие в моделях преобразователей аналогового сигналов в цифровой и цифрового в аналоговый одновременно. Цены приведены по каталогу интернет-магазина «Чип и Дип» по состоянию на июнь 2009 года.

Таблица 1. Сводная таблица по семействам 8-разрядных микроконтроллеров

<i>Название</i>	MCS-51	PIC	AVR	ADuC	PSoC
<i>Архитектура</i>	CISC	RISC	RISC	CISC	CISC
<i>Быстродействие</i>	1 MIPS при 12 МГц	5 MIPS при 20 МГц	до 16 MIPS при 16 МГц	16 MIPS при 16 МГц	до 4 MIPS при 24 МГц
<i>Энергопотребление</i>	25 мА при 12 МГц		12 мА при 8 МГц	20 мА при 16,78 МГц	27 мА при 3 МГц
<i>АЦП/ЦАП</i>	—	—	—	+	+
<i>Бесплатное средство разработки</i>	ограниченная версия Keil	MPLAB	AVR Studio	ограниченная версия Keil	PSoC Designer
<i>модель/цена</i>	AT89S52 130 р	PIC16F870 260 р	Atmega8515 260 р	ADuC814 560 р	CY8C24894 480 р

2.2. Обзор современных аппаратных средств разработки для 8-разрядных микроконтроллеров

В главе дается обзор аппаратных средств разработки для 8-разрядных микроконтроллеров. В связи с большим количеством таких средств разработок, выполнен обзор только трех наиболее популярных.

2.2.1. STK-AT89C5130 программатор-отладчик

Плата (рис. 1) предназначена для освоения программирования микроконтроллера фирмы ATMEL AT89C5131 и разработки на его основе собственных устройств и систем. Позволяет, не вдаваясь в тонкости устройства и схемотехники микроконтроллера, быстро и легко подключиться к внешнему интерфейсу компьютера USB и разрабатывать собственные устройства на основе данного микроконтроллера.

Комплект поставки содержит программное обеспечение, в том числе драйверы и примеры программирования, демонстрирующие работу устройства пользователю. Данное программное обеспечение упрощает программирование микроконтроллера и позволяет

быстро и легко создавать собственное программное обеспечение на базе микроконтроллера ATMEL AT89C5131.



Рис. 1. Отладочная плата STK-AT89C5131

Плата состоит из микроконтроллера AT89C5131, необходимого набора деталей, перемычек и разъёмов, обеспечивающих наглядную работу микроконтроллера и подключение его к USB и RS-232 (опционально). Небольшие габаритные размеры платы 115x94 мм позволяют устанавливать её во множество корпусов. Несмотря на малые габариты, плата имеет достаточно большое макетное поле, которое позволяет разместить на нём собственную схему. Макетное поле сформировано рядами металлизированных отверстий с шагом 2,54 и имеет размер 94x50 мм.

Питание схемы может осуществляться как от USB, так и от внешнего источника напряжением 5В. Для удобства отладки программного обеспечения разработчика на плату установлен двухстрочный жидкокристаллический индикатор (с контроллером аналогичным HD44780 фирмы HITACHI и KS0066 фирмы SAMSUNG) со светодиодной подсветкой. Устанавливается индикатор посредством 16-ти контактного разъёма и при необходимости может быть легко отсоединён. Под индикатором сохраняется возможность монтировать свои электронные компоненты высотой не более 8 мм. Для демонстрации возможностей мощных портов микроконтроллера управление подсветкой индикатора осуществляется посредством подключенных через перемычки от портов P3.3, P3.5, P3.6, P3.7. При

необходимости переключки могут быть отключены, и пользователь может использовать данные порты в собственных целях при макетировании.

Для удобства работы с индикатором предусмотрена регулировка его контрастности с помощью переменного резистора. Все порты микроконтроллера AT89C5131 выведены к макетному полю и сгруппированы в три группы металлизированных отверстий по 16 в каждой из групп.

Программа для внутрисистемного программирования FLIP работает под управлением операционной системы Windows 98/2000, NT или XP непосредственно через USB порт и не требует наличия специальных программаторов.

2.2.2. Отладочная плата STK-500

STK500 (рис. 2) это отладочный модуль с интегрированным программатором, предназначенный для разработки и отладки устройств на МК AVR. STK500 имеет ряд преимуществ выгодно отличающих его от других программаторов:

- возможность высоковольтного (HV) программирования;
- внутрисхемное программирование (ISP);
- возможность установки частоты ISP и HV интерфейса;
- возможность регулировки питающего VTG и опорного AREF напряжения;
- возможность регулировки ISP уровней;
- возможность обновления прошивки основного управляющего контроллера,

благодаря чему программатор поддерживает все новинки от Atmel.

Недостатками STK500, на наш взгляд, являются:

- COM интерфейс, отсутствующий на современных PC;
- большие габариты;
- отсутствие панелей с нулевым усилием сводит на нет программирование

микросхем непосредственно в программаторе т.к. что бы достать микросхему из панели необходимо приложить немалое усилие, а в случае применения отвертки велика опасность повредить плату.

- наличие дополнительного источника питания, что еще больше усложняет применение программатора в "полевых" условиях.

Для работы с программатором необходимо установить программу AVR Studio, являющуюся бесплатным средством разработки и отладки от ATMEL.

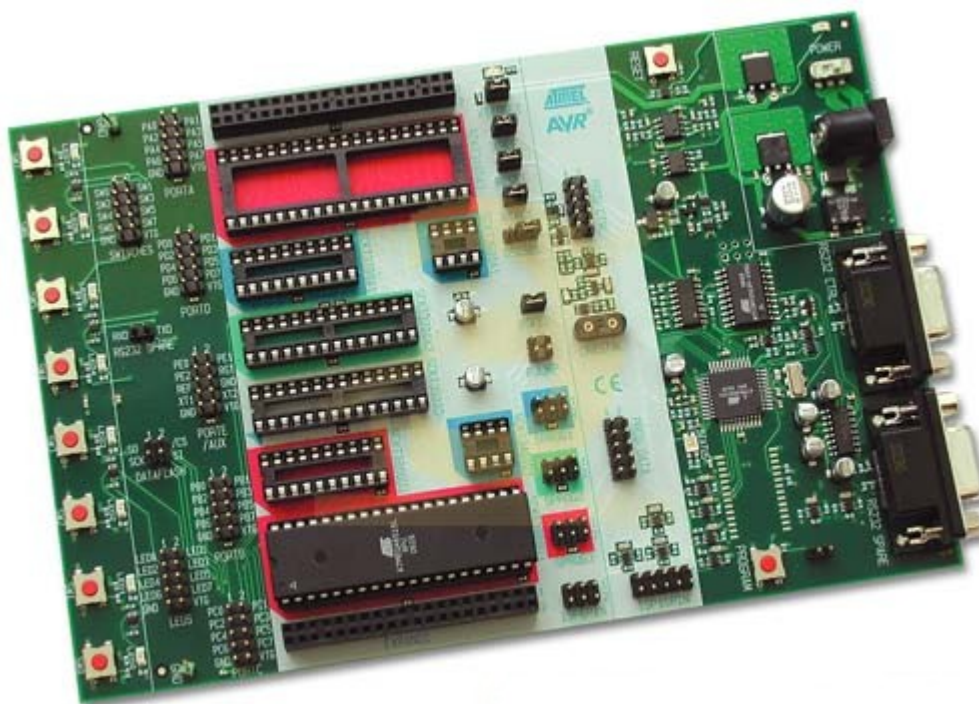


Рис. 2. Отладочная плата STK-500

Отличительные особенности:

- совместимость с программой AVR Studio;
- связь с ПК через интерфейс RS-232 для программирования и управления;
- стабилизированный источник питания с входом 10 – 15В;
- 8-выв., 20-выв., 28-выв., 40-выв. панели для установки DIP-корпусов AVR-микроконтроллеров;
- поддержка параллельного и последовательного программирования повышенным напряжением всех AVR-микроконтроллеров;
- последовательное внутрисистемное программирование всех AVR-устройств;
- внутрисистемный программатор для программирования микроконтроллера непосредственно в целевом приложении;
- перепрограммирование AVR-микроконтроллеров;
- 8 кнопок общего назначения;
- 8 светодиодов общего назначения;
- все порты ввода-вывода выведены на штырьки разъема;
- дополнительный порт RS-232 общего назначения;
- разъемы расширения для подключения внешних модулей и областей для макетирования;
- встроенная флэш-память DataFlash емкостью 2 Мбит для энергонезависимого хранения данных.

2.2.3. ME-UNI-DS3. Универсальная отладочная система на базе МК разных производителей

Универсальная система разработки для PIC, PSOC, AVR, 8051, ARM и dsPIC (рис. 3). Система поддерживает PIC, AVR, 8051, ARM, PSoc и dsPIC микроконтроллеры с большим количеством периферии. Этого достаточно, чтобы переключиться и продолжить работать в той же среде разработки, но с другим чипом.



Рис. 3. Отладочная плата ME-UNI-DS3

Отладочная плата имеет множество особенностей. Можно выбирать между питанием от USB или внешним источником. Также поддерживаются символьные и графические ЖК-дисплеи.

- На плате имеется разъем DB9 для связи по RS-232;
- все выводы микроконтроллера доступны через штыревые разъемы IDC10;
- на плате размещен USB-программатор;
- кнопки через джампер подсоединяются к земле или питанию;
- подключение к портам резисторов с подтяжкой на землю или питание, по выбору пользователя;

- 72 светодиода, 72 кнопок;
- подключение LCD-дисплея через коннектор к плате;
- выбор опорного источника питания для АЦП/ЦАП между питанием (5 В) или 4,096 В;
- часы реального времени на основе PFC8583 с поддержкой литиевой батарейки;
- ЦАП;
- каждый элемент на плате может быть отключен через джампер;
- графический LCD на 128х64 точек;
- последовательный 12-битный АЦП напряжения для приложений с измерением;
- выбор питания – от USB или внешнего источника питания;
- автономный Ethernet-контроллер через SPI-интерфейс;
- связь по RS-485 для соединения в сеть нескольких устройств;
- CAN-модуль для сообщения с другими микроконтроллерами.

Система разработки UNI-DS3 имеет полный набор функций для микроконтроллеров PIC, AVR, 8051, ARM, PSoC и dsPIC. Она предназначена для исследования возможностей различных микроконтроллеров студентами и инженерами. Позволяет сопрягать микроконтроллер с внешними схемами широкого спектра периферийных устройств, что позволяет пользователю сосредоточиться только на программном обеспечении. На плате обозначены все компоненты, режимы работы, а также некоторые другие полезные примечания.

В комплекте с отладочной системой поставляется программное обеспечение на CD-носителе. В нем содержится

- программа для внутрисхемного программирования микроконтроллера;
- демо-версии компиляторов;
- схемы платы;
- примеры программ;
- руководство по установке драйвера USB;
- драйвер для установки USB-программатора;
- руководство по эксплуатации в PDF формате;

Стоимости комплекта разработки и дополнительных комплектующих показаны в таблице 2. Цены приведены по каталогу интернет-магазина «Чип и Дип» по состоянию на май 2009.

Таблица 2. Стоимость дополнительных комплектующих к плате

<i>Название комплекта</i>	<i>Цена, руб</i>
----------------------------------	-------------------------

Отладочная плата UNI-DS3	6300
ME-UNI-DS3 40 PIN PIC CARD Дочерний модуль с установленным МК PIC16F877	2780
ME-UNI-DS3 80 PIN PIC CARD Дочерний модуль с установленным МК PIC18F8520	3300
ME-UNI-DS3 64 PIN AVR CARD Дочерний модуль с установленным МК ATmega128	2780
ME-UNI-DS3 48 PIN PSOC CARD Дочерний модуль с установленным МК Cypress CY8C27643	нет данных
ME-UNI-DS3 40 PIN 8051 CARD Дочерний модуль с установленным МК AT89S8253	3300
ME-UNI-DS3 64 PIN ARM CARD Дочерний модуль с установленным МК NXP LPC2148	3300
ME-UNI-DS3 80 PIN DSPIC CARD Дочерний модуль с установленным МК dsPIC30F6014A	нет данных
ME-LCD2x16, матричный ЖКИ с подсветкой для наборов фирмы MIKROELEKTRONIKA, формат 2x16	890
ME-GLCD 128x64, графический ЖКИ с подсветкой для наборов фирмы MIKROELEKTRONIKA, формат 128x64	1970

Данные по цене на дочерние модули с установленным микроконтроллером dsPIC30F6014A и CY8C27643 отсутствуют. На момент составления

Для сравнения выберем комплект из отладочной платы, дочерних модулей для микроконтроллеров AVR и 8051 и двухстрочного ЖК-дисплея. Общая сумма такого комплекта составит 13270,00 рублей.

2.2.4. Сводная таблица по средствам разработки

В сводной таблице 3 показан обобщенный результат по подразделу 2.2. «Обзор современных аппаратных средств разработки для 8-разрядных микроконтроллеров». Цены приведены по каталогу интернет-магазина «Чип и Дип» по состоянию на май 2009. Данный магазин выбран по причине его готовности к работе с физическими лицами.

Таблица 3. Сводная таблица по аппаратным средствам разработки

<i>Название</i>	<i>Семейство процессоров</i>	<i>Возможность программирования</i>	<i>Дополнительные модулей</i>	<i>Отлич. особенн.</i>	<i>Цена, руб.</i>
STKAT89C5131	AT89C5131	Внутрисхемное	USB RS-232 LCD (2x16)	Без возм. замены МК	3550
STK500	8-разрядные AVR	Внутрисхемное, высоковольтное последовательное, параллельное	Светодиоды кнопки RS-232 Serial EEPROM	Универс.	3542
ME-UNI-DS3	PIC, PSOC, AVR, 8051, ARM и dsPIC	Внутрисхемное	RS-232 RS-485 светодиоды кнопки RTC Ethernet CAN АЦП/ЦАП	Универс.	13270

2.3. Обзор современных программных средств разработки для 8-разрядных микроконтроллеров

Удобные средства разработки программ на ассемблере со встроенным отладчиком значительно уменьшают время разработки. В этой главе представлен обзор интегрированной среды разработки Keil μ Vision от Keil и AVR Studio от Atmel. А также приведено описание универсального программатора PonyProg для микросхем памяти и микроконтроллеров. Все рассматриваемые продукты можно достать и пользоваться свободно. Свободная версия Keil ограничена объемом компилируемого кода до 2 Кб.

2.3.1. Keil μ Vision (Keil C)

Среда μ Vision от компании Keil объединяет систему управления проектами, систему разработки и редактор исходных кодов, интерактивный отладчик и симулятор в одном мощном инструменте. Система разработки программного обеспечения μ Vision является крайне легкой в использовании и помогает быстро создать программы. μ Vision редактор и отладчик интегрированы в одно приложение, которое представляет взаимосвязанные инструменты для разработки программного обеспечения.

Отладчик μ Vision от компании Keil поддерживает режим симуляции, используя только персональный компьютер или ноутбук, а также режим отладки, взаимодействуя с целевой системой (однокристальным микроконтроллером) с помощью соответствующего интерфейса. μ Vision включает такие традиционные свойства, как простые и условные точки останова, окна наблюдения состояния переменных и контроля исполнения программы, а также такие специфические свойства, как логический анализатор, профайлер исполнения программы и анализ покрытия кода.

Данная среда программирования кроме графического интерфейса и отладчика включает компиляторы для однокристальных микропроцессоров серий: C51, C166/167, C251, ARM более 70 производителей, в том числе таких как Analog Devices, Atmel, Daewoo, Dallas Semiconductor, Honeywell, Intel, Maxim, OKI, Philips, Samsung, Sanyo, Sharp, Siemens, TDK и многих других. Компилятор для серии AVR отсутствует.

Следует заметить, что данный продукт – коммерческий. На сегодня имеется стабильная версия μ Vision3 и бета версия μ Vision4. Демо-версия для серии C51 доступна для скачивания на официальном сайте разработчика (www.keil.com). Ограничения налагаются на объем кода. Он не должен превышать 2 Кб. Этого достаточно для написания многих программ на языке программирования ассемблер.

2.3.2. AVR-Studio

AVR Studio 4 - новая профессиональная интегрированная среда разработки (IDE), предназначенная для написания и отладки прикладных программ для AVR микропроцессоров в среде Windows 9x/NT/2000. AVR Studio 4 содержит ассемблер и симулятор. Также IDE поддерживает такие средства разработки для AVR как: ICE50, ICE40, JTAGICE, ICE200, STK500/501/502 и AVRISP. В ближайшие месяцы будет расширен список поддерживаемых AVR Studio 4 микроконтроллеров и средств разработки. Обо всех обновлениях можно будет узнать на сайте компании Atmel (www.atmel.com).

AVR Studio поддерживает COFF как формат выходных данных для символьной отладки. Другие программные средства третьих фирм также могут быть сконфигурированы для работы с AVR Studio. Ключевое окно в AVR Studio это окно исходного текста программы. Когда объектный файл открыт, автоматически создается окно исходного текста программ. В окне отображается код, который выполняется в отладочном окружении (эмуляторе или программном симуляторе) а текстовый маркер всегда находится на строке, которая будет выполнена в следующем цикле. Пользователь может выполнять программу

полностью в пошаговом режиме, трассируя блоки функций, или выполняя программу до места, где стоит курсор. В дополнение можно определять неограниченное число точек останова, каждая из которых может быть включена или выключена. Точки останова сохраняются между сессиями работы. В окне исходного текста программы выводится информация о процессе выполнения программы. В дополнение, AVR Studio имеет много других окон, которые позволяют управлять и отображать информацию о любом элементе микроконтроллера. Настройки рабочего окружения сохраняются при выходе. При первом запуске требуется настроить окна для управления и вывода необходимой информации. Во время следующей загрузки настройки автоматически восстанавливаются.

2.3.3. PonyProg

PonyProg — это программа на ПК к простейшему универсальному программатору. Она позволяет работать со множеством видов микросхем. На сегодняшний день доступны программированию многие микроконтроллеры серии AVR. Программа PonyProg — это открытый проект. Для распространения этой программы и еще нескольких проектов в Интернете создан специальный сайт по адресу www.lancos.com. Программа распространяется с открытой лицензией (GNU), то есть вместе с текстом программы, который разрешается изменять по своему усмотрению. Однако в пакет программы входит специальная библиотека, которая содержит текст всех основных функций, обеспечивающих процесс программирования микросхем. Его изменять не рекомендуется.

Простая схема подключения через СОМ или параллельный порты, который не составит труда собрать, бесплатный дистрибутив делают его популярным при программировании микросхем памяти и микроконтроллеров.

2.3.4. Сводная таблица по программным средствам разработки

В сводной таблице 4 показан обобщенный результат по подразделу 2.3. «Обзор современных программных средств разработки для 8-разрядных микроконтроллеров» для каждого рассмотренного программного средства.

Таблица 4. Сводная таблица по программным средствам разработки

Название	Keil uVision	AVR Studio	PonyProg
----------	--------------	------------	----------

Функциональное назначение	интегрированная среда разработки (IDE)	интегрированная среда разработки (IDE)	Программатор
Ограничение	до 2 Кб конечного кода	без ограничений	без ограничений
Поддерживаемые семейства МК	MCS-51 (в том числе и ADuC)	AVR	AVR, MCS-51, PIC
Сайт разработчика	www.keil.com	www.atmel.com	www.lancos.com

2.4. Выводы по главе

В настоящей главе рассмотрены наиболее популярные семейства микроконтроллеров на ядрах C51 (от Intel), PIC (от Microchip), AVR (от Atmel). Также дается общий обзор интересных свойств PSoC (от Cypress) и ADuC (от Analog Devices). Для практических применений были выбраны микроконтроллеры серии MCS-51 и AVR. Данный выбор продиктован распространенностью данных микроконтроллеров (в том числе и на различных предприятиях г. Чебоксары). Приобретение навыков при работе с ними будет востребовано в дальнейшей профессиональной деятельности студентов. К тому же, по ценовым показателям, данные микроконтроллеры способствуют достижению низкой себестоимости, сформулированной в требованиях.

Анализ существующих аппаратных средств разработки показал достоинства и недостатки каждого из них. Решения, заложенные в них, будут учитываться при проектировании учебного модуля.

При написании программ для микроконтроллера MCS-51 можно воспользоваться интегрированной средой разработки Keil μ Vision. Ограничения свободной версии достаточно для написания многих учебных программ на ассемблере. Для семейства 8-разрядных AVR удобно использовать AVR Studio от самого производителя. Программирование же осуществляется через PonyProg с помощью несложной схемы.

3. Проектно-конструкторская часть

В этой главе подробно рассматриваются проектирование отдельно аппаратной и программной части. Процесс проектирования введется с учетом требований, предъявленных в главе «Постановка задачи».

3.1. Разработка аппаратной части

Аппаратная часть разрабатывалась в период подготовки к научной конференции 2008 года. Разработка печатных плат ввелась на САПР P-CAD 2006.

Учебный модуль состоит из базового модуля и множества дополнительных модулей. Таким образом, реализован принцип модульности, с возможностью дополнения другими комплектующими.

3.1.1. Описание базового модуля

Базовый модуль содержит вычислительное устройство (микроконтроллер) с дополнительными микросхемами памяти (внешняя память программ и данных), с цепью сброса, питания и управляющих цепей режимами работы. Микроконтроллер выбран из семейства MCS-51. Модель AT89S52 от компании Atmel соответствует этой серии с небольшими изменениями (увеличена оперативная память, добавлены новые периферийные устройства). К микроконтроллеру подключены по рекомендуемой схеме внешние памяти программ и данных. Модуль имеет два режима работы – режим программирования и режим выполнения. При режиме программирования выполняется программа, зашитая во встроенной памяти программ микроконтроллера. Эта программа выполняет прием данных из последовательного порта и прошивает их во внешнюю память программ. Таким образом, происходит программирование базового модуля. При режиме выполнения действует программа, записанная во внешней памяти программ. Распределение памяти при различных режимах показана на рис. 4. Текущий режим указывается свечением определенного светодиода.

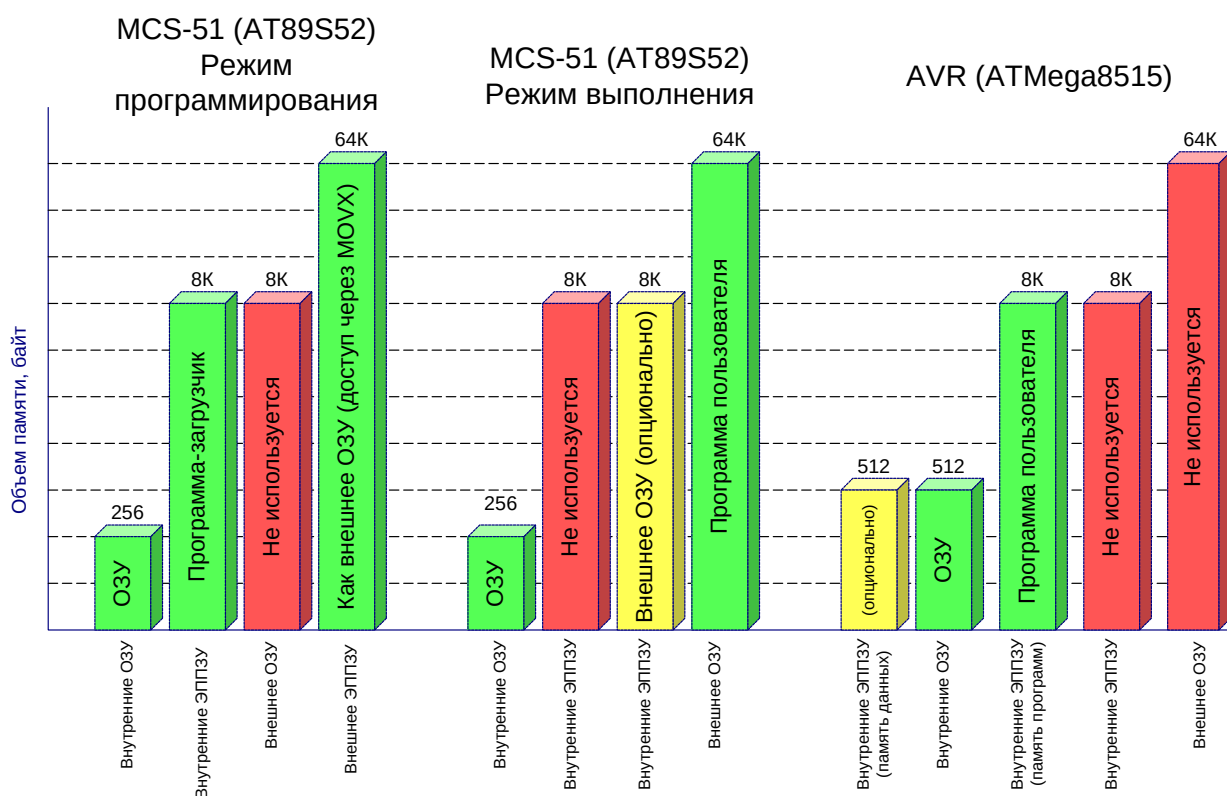


Рис. 4. Распределение памяти при различных режимах

Переключение режимов реализовано на основе возможности выполнения микроконтроллером программы, как из внутренней памяти, так и из внешней памяти программ. Выбор зависит от сигнала #EA AT89S52 — разрешение внешней памяти программ. Низкий уровень сигнала разрешает доступ к внешней памяти, а при высоком уровне выполняется внутренняя память flash.

По краям платы на разъемы вынесены неиспользуемые порты микроконтроллера (порт 1 и порт 3). При анализе последовательности выводов микроконтроллера AT89S52 было замечено, что он совпадает с микроконтроллером серии AVR mega — ATmega8515. Таким образом, в базовом модуле реализована возможность замены микроконтроллеров и использования его как базовый модуль с Atmega8515. Для этого на плате модуля были добавлены разъемы на другие порты (порт 0 и порт 2 для AT89S52) и возможность выбора цепи сброса. При работе с AT89S52 эти порты не используются, так как они участвуют в циклах записи и чтения при обращении к микросхемам внешней памяти. Также необходимо учесть логику сброса двух разных микроконтроллеров. У семейства AVR цепь сброса инверсная, а у семейства MCS-51 — прямая. Переключение цепей сброса выполняется джампером.

3.1.2. Описание дополнительных модулей

Дополнительные модули реализуют средства ввода-вывода информации, интерфейсные модули, выполняют коммутационные задачи и подключают дополнительные микросхемы. На сегодня к ним относятся:

1. Конвертер сигналов RS232 – TTL.
2. Дискретные датчики.
3. Часы реального времени на DS1306
4. Перенаправитель портов.
5. Линейка светодиодов.
6. Выход на клеммники.
7. Семисегментный индикатор.
8. Клавиатура.
9. ЖК индикатор MT-10T7.
10. ЖК индикатор WH1604.
11. Термодатчик DS18B20

Конвертер сигналов RS232 – TTL реализован на микросхеме ADM202 от MAXIM. Модуль применяется для согласования уровней сигналов при связи компьютера и базового модуля по последовательному порту COM.

Дискретные датчики используются для подачи на вход микроконтроллера логических ноль или единица. В схеме применены навесные резисторы с переключателем DIP. Также его можно применять как внешние навесные резисторы для портов, настроенных на вход. Микроконтроллеры серии MCS-51 внутренних подключаемых резисторов не имеют.

Часы реального времени позволяет определять и следить за текущим временем независимо от сбоев в питании. Интерфейс связи с базовым модулем выбирается переключателем — стандартный SPI или фирменный от Motorola. На выходе подключен светодиод. Его программное включение выдаст мигание с частотой 1 Гц.

Модуль перенаправителя портов служит для изменения последовательности или разрыва разрядов порта между двумя стандартными разъемами ВН10.

Линейка светодиодов используется для вывода двоичной информации. Его удобно использовать при слежении за состоянием порта. Например, подключенный параллельно к последовательному порту, светодиод своим миганием показывает, что состояние линий RxD и TxD изменяются и, следовательно, происходит обмен.

Выход на клеммники со стандартного разъема предназначен для подсоединения проводов с другими внешними устройствами, для которых делать специальный модуль не

целесообразно. Примером является термодатчик с тремя выводами — два вывода питания и один вывод для интерфейса; кнопка с или без фиксации (во многих случаях дискретные датчики не удобны при частых переключениях); герконы, реле, шаговые двигатели.

Семисегментный индикатор с четырьмя цифрами идеально подходит для вывода значений времени, температуры и других измеренных параметров. Индикатор специальным образом спроектирован для совмещения с клавиатурой 4x4.

Клавиатура из 16 кнопок, размещенных на пересечении линий портов 4 по 4. На каждой линии есть резистор на 2,7 кОм. Модуль спроектирован по рекомендациям Atmel.

Цифровой ЖК-индикатор МТ-10Т7 и алфавитно-цифровой WH1604 применяются для вывода сложной информации.

3.2. Разработка программной части

Программное обеспечение для разрабатываемого модуля состоит из трех частей.

1. Программа на стороне персонального компьютера (Programmer51).
2. Программа на стороне базового модуля (Embedded51).
3. Пакет программ для тестирования изделия.

В главе рассматриваются разработка протокола для обмена информацией между базовым модулем и компьютером. Он является связывающей частью между двумя программами (Programmer51 и Embedded51). Его подробное описание приводится в главе «3.2.1. Разработка протокола загрузки программы пользователя для MCS-51». Проектирование Programmer51 рассматривается в главе 3.2.4 «Разработка программы для компьютера». Разработка модуля по работе с HEX форматом рассматривается отдельно от проектирования Programmer51 в главе «3.2.2. Разбор файла формата HEX». При написании программы использовался язык программирования C++. На его выбор повлияла возможность объектно-ориентированного подхода при написании программ. Такой подход эффективен при разработке сложных программ с множеством связей между модулями. К тому же есть опыт использования C++ в ходе обучения на данной специальности.

В главе «3.2.3. Разработка встраиваемого программного обеспечения» приводятся блок-схемы алгоритмов подпрограмм Embedded51. В функцию программы входит прием пакета команды через последовательный порт с последующим их разбором, выполнением и отправкой ответа. При написании программы была использована интегрированная среда разработки Keil uVision, с ограничением компилируемого кода до 2 Кб. Выбор языка

программирования ассемблер позволит нам не превысить критического объема и выиграть в быстродействии выполнения.

Программы разрабатывались по требованиям, изложенным в постановке задачи.

3.2.1. Разработка протокола загрузки программы пользователя для MCS-51

Чтобы принимающее устройство умело правильно интерпретировать полученные данные, необходима договоренность между приемником и передатчиком – протокол обмена. Обмен данными происходит по последовательному порту COM компьютера и UART микроконтроллера. На рис. 5. изображена обобщенная граф-схема автомата, описывающего последовательность приема данных, их анализ, обработку и отправку ответа.

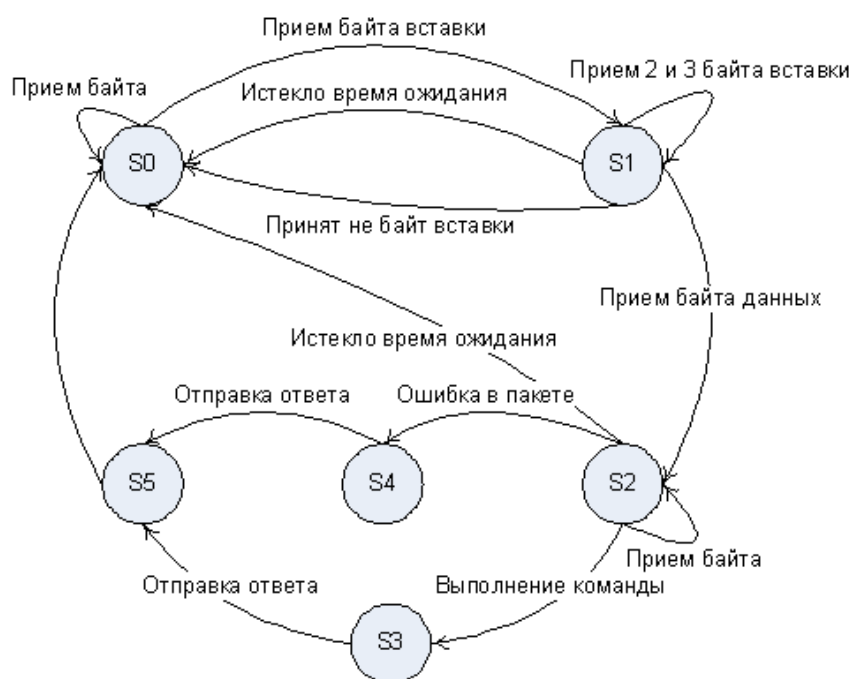


Рис. 5. Граф-схема автомата приема данных

В разрабатываемом протоколе необходимо предусмотреть формат пакетов отправляемых команд со стороны персонального компьютера (далее по тексту ПК) и возможных ответов от базового модуля. Перечисленные выше возможности, которые можно проделать над памятью FLASH, составляют перечень команд, отправляемых с ПК:

- программирование памяти;
- чтение из памяти;

- идентификация устройства;
- стирание.

Инициатором обмена является программа на стороне ПК. Это облегчает алгоритмы обмена данными и восстановление связи после его обрыва. ПК пересылает программе базового модуля (далее по тексту МК, от «микроконтроллер») пакет данных, называемых командой, для выполнения. Пакет данных содержит в себе код и параметры команды. Учитывая то, что есть вероятность ошибки при передаче данных по каналу связи, правильным будет, если в формат передаваемых команд и ответов будет добавлена контрольная сумма, гарантирующая целостность всего пакета. Значение этого поля равно дополнению по модулю 256 суммы по модулю 256 всех байтов, начиная с поля кода команды (вставочные байты не учитываются). Рассмотрим каждую команду и ответ на него по отдельности.

Идентификация устройства FLASH памяти позволяет определить производителя и код продукта. Выявляет ошибки типа отсутствия микросхемы в колодке, несоответствие памяти определенному типу (SST29EE010) или повреждение и выход из строя. Ответом МК будет код производителя и устройства, считанное по правилам, описанные в документации к микросхеме памяти. На рис. 6а. показан формат команды на получение идентификатора flash-памяти. Первый байт (0x01) — это код команды. Параметров команды пакет не имеет, поэтому сразу за кодом следует контрольная сумма. Его значение 0xFF получено по вышеописанному правилу. Ответом (рис. 6б) будет 4 байтная последовательность. Код ответа 0x05 дает информацию о том, что данный пакет является ответом на запрос получения ID. Поле manID содержит код производителя, а devID — код устройства. Контрольная сумма CS вычисляется исходя из остальных полей, как сказано выше.

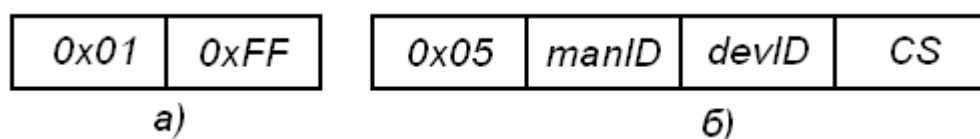


Рис. 6. Формат команды (а) и ответа (б) на получение идентификатора flash-памяти

Стирание FLASH памяти позволяет быстро заполнить память значениями FFh. Это удобнее той же операции записи по времени выполнения. На рис. 7а и 7б показаны форматы команды и ответа на запрос стирания flash-памяти соответственно. Код команды 0x02 соответствует коду команды стирания памяти, а 0x06 — коду ответа на эту команду. Параметров команда и ответ не имеют. Контрольные суммы уже вычислены.

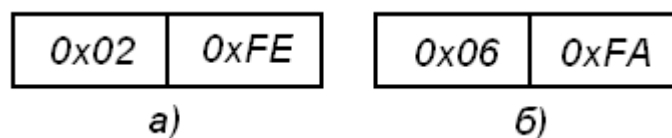


Рис. 7. Формат команды и ответа на запрос стирания flash-памяти

Рис. 8а. описывает формат команды запроса на чтение. Код команды 0x03. В поле AddrL и AddrH заносятся младший и старший адреса страницы, которую необходимо прочесть. Таким образом можно адресовать до 64 Кб памяти. Контрольная сумма CS вычисляется вышеописанным способом по трем первым полям. Формат ответа имеет вид, представленный на рис. 8б. Код ответа 0x07. В AddrL и AddrH заносятся младший и старший байты адреса страницы. В полях Data0 ... Data127 содержатся прочитанные данные. Контрольная сумма CS вычисляется по первым 131 полям.

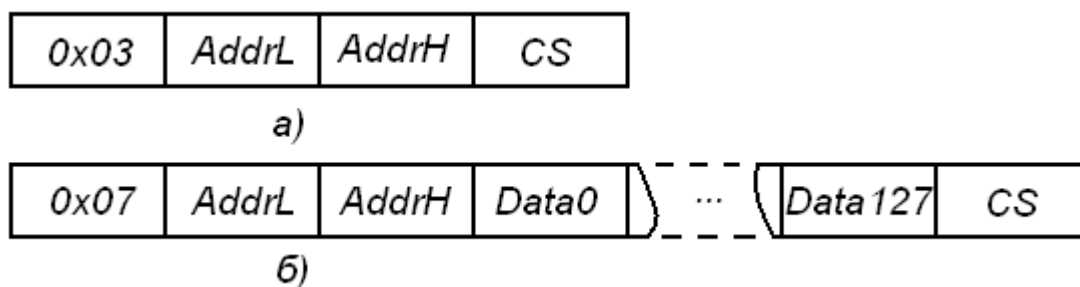


Рис. 8. Формат команды и ответа на запрос чтения flash-памяти

Чтение FLASH памяти происходит постранично. ПК отправляет запрос на чтение определенной страницы и ожидает ответ. Это позволяет получить значения только определенных областей памяти быстрее. К тому же уменьшается вероятность ошибки и появляется возможность его локализации. Но чтение всей памяти займет большее время по сравнению с передачей данных для всего устройства.

Запись во FLASH память также происходит постранично. Параметрами команды являются номер страницы и записываемые данные в объеме 128 байт. После записи во FLASH память от МК приходит ответ с номером записанной страницы. На рис. 9 показаны формат команды (рис. 9а) и ответа (рис. 9б) на запрос записи во flash-память. Значения полей аналогичны вышеописанной команде и ответу на запрос записи. В этом случае код команды для записи 0x04, а ответа на него 0x08.

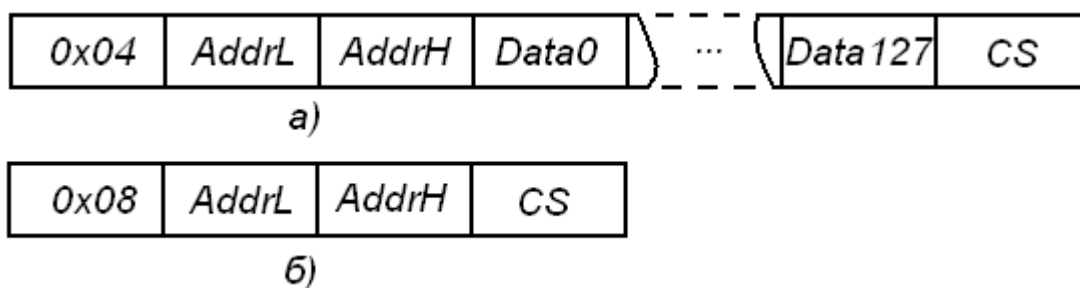


Рис. 9. Формат команды и ответа на запрос записи во flash-память

Каждый пакет данных, принимаемый и передаваемый, содержит последовательность трех вставочных байт и заканчивается контрольной суммой. Вставочные байты в описаниях форматов посылок не указаны.

Во время принятия и выполнения команд возможны разного вида ошибки. Неверный формат команды (неопределенный код команды, недостаточное количество параметров, неверно высчитанная контрольная сумма) или невозможность или ошибочность выполнения команд. На такие ситуации будет пакет-ответ в следующем формате (рис. 10):

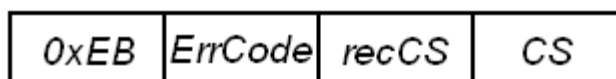


Рис. 10. Формат пакета ошибки

0xEB — код, указывающий, что пакет является описанием произошедшей ошибки. ErrCode содержит код ошибки. Возможные коды ошибки:

- 0x01 — ошибка в контрольной сумме;
- 0x02 — ошибка в коде команды.
- 0x03 — ошибка при выполнении команды.

В поле recCS заносится высчитанная контрольная сумма последнего принятого пакета. Оно пригодится в случае ошибки с кодом 0x01 — ошибка в контрольной сумме.

3.2.2. Разбор файла формата HEX

В этом разделе описывается формат объектного шестнадцатичного файла Intel-HEX для 8-, 16- и 32-битных микроконтроллеров. Он может быть использован в качестве исходного для программаторов или аппаратных эмуляторов. Приведены описания алгоритмов разбора и получения данных для прошивки.

Шестнадцатеричный объектный формат файлов Intel-HEX (далее просто HEX-формат) – это способ представить двоичные данные в виде кодов ASCII. Поскольку файл состоит из символов ASCII, а не двоичных кодов, появляется возможность хранить данные на бумаге, перфоленте или перфокартах, выводить их на терминал, принтер и т.д. Восьмибитовый HEX-формат файлов предусматривает размещение данных и кода в 16-разрядном линейном адресном пространстве для 8-разрядных процессоров Intel. 16-разрядный HEX-формат файлов дополнительно позволяет использовать 20-разрядное сегментное пространство адресов 16-разрядных процессоров Intel. И, наконец, 32-разрядный формат позволяет оперировать линейным 32-разрядным адресным пространством 32-разрядных процессоров.

Шестнадцатеричное представление двоичных данных в виде ASCII требует использование двух символов для записи одного байта, при этом первый символ всегда соответствует старшей тетраде битов одного байта. Такой подход увеличивает количество символов в двое по сравнению с количеством двоичных данных.

Формат файла организован в виде набора записей, содержащих сведения о типе, количестве данных, адресе их загрузки в память и дополнительные сведения. В настоящее время определены 6 различных типов записей, однако не все их комбинации определены для разных форматов данных. Записи могут быть следующих типов:

- Данные (определена для всех форматов данных).
- Маркер конца файла (определена для всех форматов файла).
- Сегментный адрес (определена для 16- и 32-битных форматов).
- Сегментный адрес старта (определена для 16- и 32-битных форматов).
- Линейный адрес (определена только для 32-битного формата).
- Линейный адрес старта (определена только для 32-битного формата).

На рис. 11. представлен общий формат записи строки HEX файла.

МАРКЕР ЗАПИСИ	КОЛ-ВО ДАННЫХ	СМЕЩЕНИЕ	ТИП ЗАПИСИ	ДАННЫЕ ИЛИ ИНФОРМАЦИЯ	КОНТРОЛЬНАЯ СУММА
	RECLen	OFFSET	TYPEREc	INFO/DATA	CHECKSUM
«:»	1 байт	2 байта	1 байт	RECLen байт	1 байт

Рис. 11. Общий формат записи

Каждая запись представляет собой ASCII-строку файла. В одной строке – одна запись.

Каждая запись начинается с МАРКЕРА ЗАПИСИ, который обозначается ASCII-символом двоеточие («:»).

Каждая запись содержит поле RECLen, определяющее количество байтов данных или информационных байтов, назначение которых определяется типом записи. Максимальное значение этого поля – 255 (0FF в шестнадцатиричном формате).

Каждая запись содержит поле OFFSET, определяющее 16-битное смещение в адресном пространстве байтов данных. Это поле используется только в записях данных, а в остальных случаях оно должно быть равно нулю.

Каждая запись содержит поле TYPEREC, определяющее тип текущей записи (из ранее упомянутых шести). Это поле используется для интерпретации всех остальных полей записи.

Типы записей кодируются следующими значениями поля TYPEREC (в ASCII):

- «00» – данные.
- «01» – маркер конца файла.
- «02» – адрес сегмента.
- «03» – сегментный адрес старта.
- «04» – линейный адрес.
- «05» – линейный адрес старта.

Каждая запись содержит поле INFO/DATA переменной длины, которое содержит ноль или более байтов, закодированных символами ASCII. Назначение этих байтов определяется типом записи.

Наконец, каждая запись завершается полем CHECKSUM, гарантирующим целостность всех данных записи. Значение этого поля равно дополнению по модулю 256 до нуля суммы по модулю 256 всех байтов, начиная с поля RECLen и заканчивая последним байтом поля INFO/DATA. При считывании записи следует суммировать по модулю 256 все байты записи, включая поле CHECKSUM. Если в конце концов сумма равна нулю, это означает, что данные считаны без искажений, в противном случае данные недостоверны.

Формат записи линейного адреса следующий показан на рис. 12.

МАРКЕР ЗАПИСИ	КОЛ-ВО ДАННЫХ	СМЕЩЕНИЕ	ТИП ЗАПИСИ	ДАННЫЕ ИЛИ ИНФОРМАЦИЯ	КОНТРОЛЬНАЯ СУММА
	RECLen	OFFSET	TYPEREC	ULBA	CHECKSUM
«:»	«02»	«0000»	«04»	2 байта	1 байт

Рис. 12. Формат записи линейного адреса

Эта запись служит для задания значения битов 16-31 в линейном базовом адресе (LBA, Linear Base Address), причем биты 0-15 LBA равны нулю. Биты 16-31 LBA определяются верхним линейным базовым адресом (ULBA, Upper Linear Base Address). Абсолютное значение адреса байта данных в памяти определяется как сумма значения LBA

и значения поля OFFSET в последующих записях данных, плюс индекс байта данных внутри поля DATA. Эта сумма выполняется без учета переполнения результата (т.е. не может превышать 0FFFFFFF, 4Гб). Фактический линейный адрес байта данных вычисляется в итоге по формуле:

$$\text{ByteAddr} = (\text{LBA} + \text{DRLO} + \text{DRI}) \bmod 4\text{G},$$

где DRLO – значение поля OFFSET записи данных, DRI – индекс байта в поле DATA записи данных, mod 4G – операция «сложение по модулю 32».

Когда запись «Линейный адрес» встречается в файле, вычисляется значение LBA, которое действует для всех последующих записей данных, пока не встретится снова запись «Линейный адрес». По умолчанию LBA=0. Значение остальных полей рассмотрено ранее.

Формат записи адреса сегмента показан на рис. 13.

МАРКЕР ЗАПИСИ	КОЛ-ВО ДАННЫХ	СМЕЩЕНИЕ	ТИП ЗАПИСИ	ДАННЫЕ ИЛИ ИНФОРМАЦИЯ	КОНТРОЛЬНАЯ СУММА
	RECLEN	OFFSET	TYPERECL	USBA	CHECKSUM
«:»	«02»	«0000»	«02»	2 байта	1 байт

Рис. 13. Формат записи адреса сегмента

Эта запись служит для задания значения битов 4-19 сегментного базового адреса (SBA, Segment Base Address), где биты 0-3 SBA равны нулю. Биты 4-19 SBA определяются верхним базовым адресом сегмента (USBA, Upper Segment Base Address). Абсолютный адрес байта в записи данных вычисляется путем прибавления к SBA значения поля OFFSET записи данных и индекса байта относительно начала поля DATA/INFO. Прибавление смещения (OFFSET) осуществляется по модулю 65536 (64K), без учета переполнения. Таким образом, адрес конкретного байта вычисляется по формуле:

$$\text{ByteAddr} = \text{SBA} + ([\text{DRLO} + \text{DRI}] \bmod 64\text{K}),$$

где DRLO – значение поля OFFSET записи данных, DRI – индекс байта в поле DATA записи данных, mod 64K – операция «сложение по модулю 65536».

Когда запись «Адрес сегмента» встречается в файле, вычисляется значение SBA, которое действует для всех последующих записей данных, пока не встретится снова запись «Адрес сегмента». По умолчанию SBA=0. Значение остальных полей рассмотрено ранее.

Формат записи с данными показан на рис. 14.

МАРКЕР ЗАПИСИ	КОЛ-ВО ДАННЫХ	СМЕЩЕНИЕ	ТИП ЗАПИСИ	ДАННЫЕ ИЛИ ИНФОРМАЦИЯ	КОНТРОЛЬНАЯ СУММА
	RECLen	OFFSET	TYPEREc	ULBA	CHECKSUM
«:»	1 байт	2 байта	«00»	RECLen байтов	1 байт

Рис. 14. Формат записи с данными

Эта запись собственно и содержит данные. Метод вычисления фактического (абсолютного) адреса каждого байта данных в памяти определяется по вышеприведенным формулам и зависит от формата данных. Назначение всех полей этой записи рассмотрено ранее.

Формат записи линейного адреса старта показан на рис. 15.

МАРКЕР ЗАПИСИ	КОЛ-ВО ДАННЫХ	СМЕЩЕНИЕ	ТИП ЗАПИСИ	ДАННЫЕ ИЛИ ИНФОРМАЦИЯ	КОНТРОЛЬНАЯ СУММА
	RECLen	OFFSET	TYPEREc	EIP	CHECKSUM
«:»	«04»	«0000»	«05»	4 байта	1 байт

Рис. 15. Формат записи линейного адреса старта

Запись «Линейный адрес старта» используется для указания адреса, с которого начинается исполнение объектного файла. Это значение заносится в регистр EIP процессора. Обратите внимание, что эта запись определяет только точку входа сегмента кода для защищенного режима процессоров 80386. В обычном режиме точка старта определяется записью «Сегментный адрес старта», которая определяет значения пары регистров CS:IP.

Запись «Линейный адрес старта» может находиться в любом месте файла. Если ее нет, загрузчик использует адрес старта по умолчанию.

Значение регистра EIP процессора содержится в соответствующем поле записи, для него требуется всегда 4 байта. Назначение остальных полей записи рассмотрено ранее.

Формат записи сегментного адреса старта показан на рис. 16.

МАРКЕР ЗАПИСИ	КОЛ-ВО ДАННЫХ	СМЕЩЕНИЕ	ТИП ЗАПИСИ	ДАННЫЕ ИЛИ ИНФОРМАЦИЯ	КОНТРОЛЬНАЯ СУММА
	RECLen	OFFSET	TYPEREc	CS:IP	CHECKSUM
«:»	«04»	«0000»	«03»	4 байта	1 байт

Рис. 16. Формат записи сегментного адреса старта

Запись «Сегментный адрес старта» используется для указания адреса, с которого начинается исполнение объектного файла. Это значение определяет 20-битный адрес,

заносимый в регистры CS:IP процессора. Обратите внимание, что эта запись определяет только точку входа в 20-битном адресном пространстве процессоров 8086/80186.

Запись «Сегментный адрес старта» может находиться в любом месте файла. Если ее нет, загрузчик использует значение по умолчанию.

Значение регистров CS:IP процессора содержится в соответствующем поле записи, для него требуется всегда 4 байта. Значение хранится в порядке «от старшего к младшему», т.е. младший байт значения регистра IP хранится в четвертом байте поля CS:IP, старший – в третьем, затем во втором хранится младший байт значения регистра CS, и в первом – старший байт регистра CS. Назначение остальных полей записи рассмотрено ранее.

Формат записи маркера конца файла или терминатора показан на рис. 17.

МАРКЕР ЗАПИСИ	КОЛ-ВО ДАННЫХ	СМЕЩЕНИЕ	ТИП ЗАПИСИ	КОНТРОЛЬНАЯ СУММА
	RECLEN	OFFSET	TYPERECD	CHECKSUM
«C»	«00»	«0000»	«01»	FF

Рис. 17. Формат записи маркера конца файла (терминатора)

Эта запись не содержит полей с изменяющимися данными, поэтому выглядит всегда совершенно одинаково: «C:00000001FF». Запись обозначает конец данных в файле и должна быть. Все последующие строки, если они есть в файле, игнорируются.

3.2.3. Разработка встраиваемого программного обеспечения

Разработка программы на микроконтроллер ввелась в интегрированной среде разработки Keil uVision. Использовался язык программирования ассемблер.

Программа на стороне микроконтроллера предназначена для приема от ПК команд, выполнения и формирования ответа на них. Основная программа представляется как бесконечный цикл с ожиданием данных, декодированием и выполнением действий. Блок-схема программы МК представляется на рис. 18.

Прерывание от последовательного порта по приему символа формирует последовательность байт, составляющих команду. По окончании формирования, устанавливается флаг приема очередной команды. Принятая команда декодируется на код команды и соответствующие к нему параметры. Исходя из кода команды, выполняются необходимые действия.

Для практической реализации конечного автомата применили команду тестирования бита с последующим его сбросом — **jbc**. Состояния автомата — это переменные-биты. При инициализации нулевое состояние устанавливается.

```
jbc      state_0, jump_0
jbc      state_1, jump_1
jbc      state_2, jump_2
jbc      state_3, jump_3
jbc      state_4, jump_4
.....
```

В части обработки состояния следующее состояние устанавливается командой **setb**.



Рис. 18. Блок-схема алгоритма программы для МК

Блок-схема подпрограммы приема пакета данных показана на рис. 19. При составлении алгоритм применялся теория конечного автомата. Это позволило сделать программу на ассемблере понятной и дополняемой.

Блок-схемы алгоритмов подпрограмм «Выполнение команды» и «Посылка ответа» представлены на рис. 20. Операции стирания, чтения и записи во flash-память выполняются по требованиям технической документации [5]. Временные диаграммы и блок-схемы алгоритмов действий в отчете не приведены. Они рассматривались в ходе преддипломной практики.

Управляющие сигналы flash-памяти #WE, #OE, #CE и управляющий сигнал #ALE загрузкой в регистр-защелку во время режима программирования подключаются к портам P1 микроконтроллера. Таким образом, временные диаграммы записи-чтения flash-памяти соблюдаются программой. Это дает гибкость при работе с памятью и страхует от возможной несогласованности во временных диаграммах flash и внешнего цикла чтения-записи.

Оперативная память AT89S52 имеет нерегулярное свободное пространство. Специальные регистры ввода-вывода разбросаны по оперативной памяти микроконтроллера. Для сохранения страницы flash (128 байт данных) в памяти программ определена таблица индексов свободных ячеек ОЗУ. Для получения i-й ячейки памяти, обращаемся к таблице с памяти программ. Значение, стоящее по этому адресу определяет адрес i-ой свободной ячейки ОЗУ. Приведем часть программы записи:

```

mov          DataCount, #0
mov          DPTR, #PN
read_cicle:
mov          R7, AddrPage + 1
mov          A, AddrPage + 0
add          A, DataCount
mov          R6, A
call         ReadByte ; R6 <- MEM[R7:R6]
mov          A, DataCount
movc         A, @A + DPTR
mov          R0, A
mov          A, R6
mov          @R0, A
inc          DataCount
mov          A, DataCount
cjne         A, #128, read_cicle

```

#PN — адрес таблицы индексов в памяти программ.

DataCount — количество обработанных байт (от 0 до 127)

ReadByte — функция считывания байта из flash. Адрес передается через R6 и R7. Считанная данная записывается в R6.

AddrPage — адрес страницы flash, которую обрабатываем.

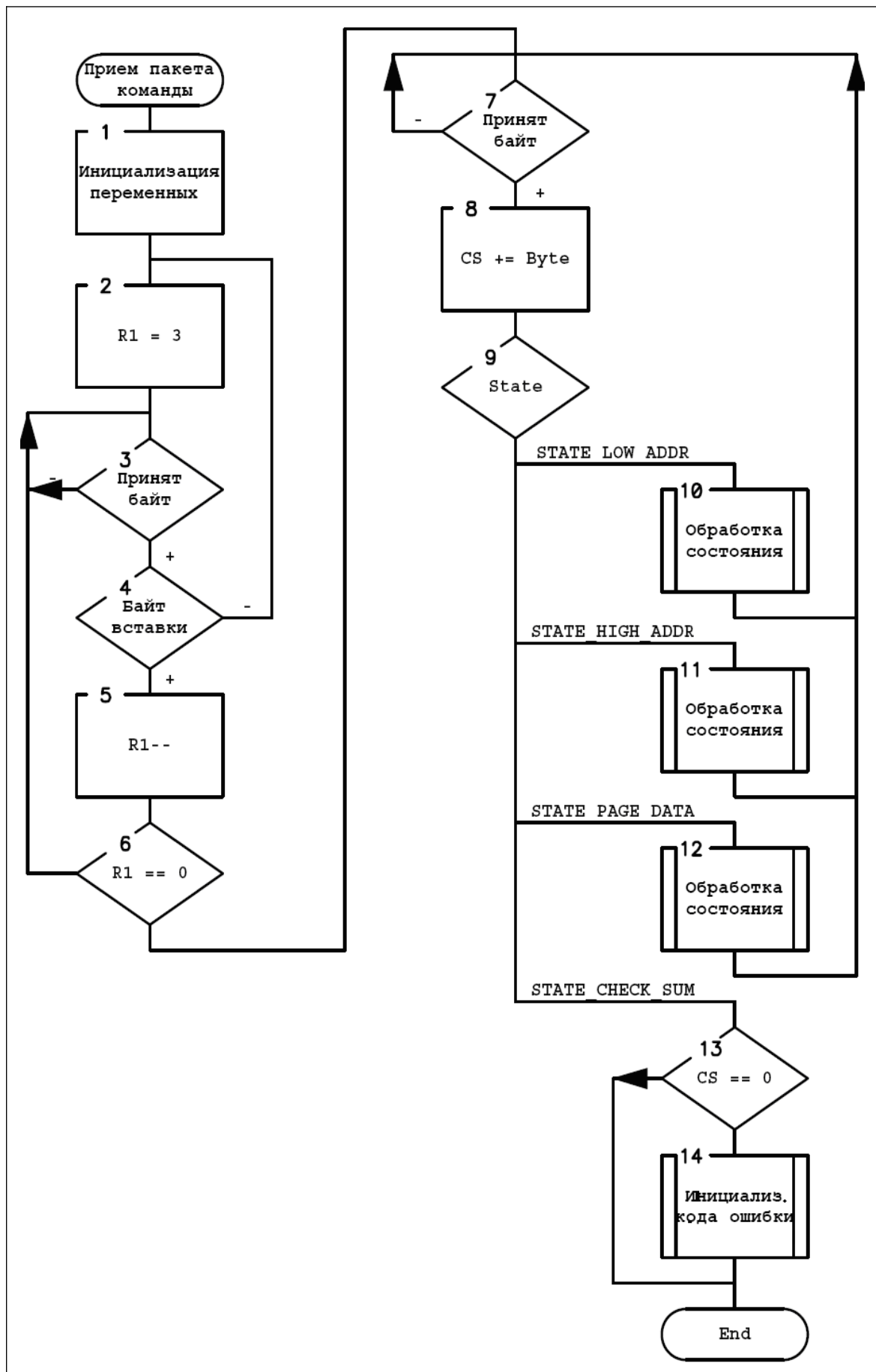


Рис. 19. Блок-схема алгоритма приема пакета команды

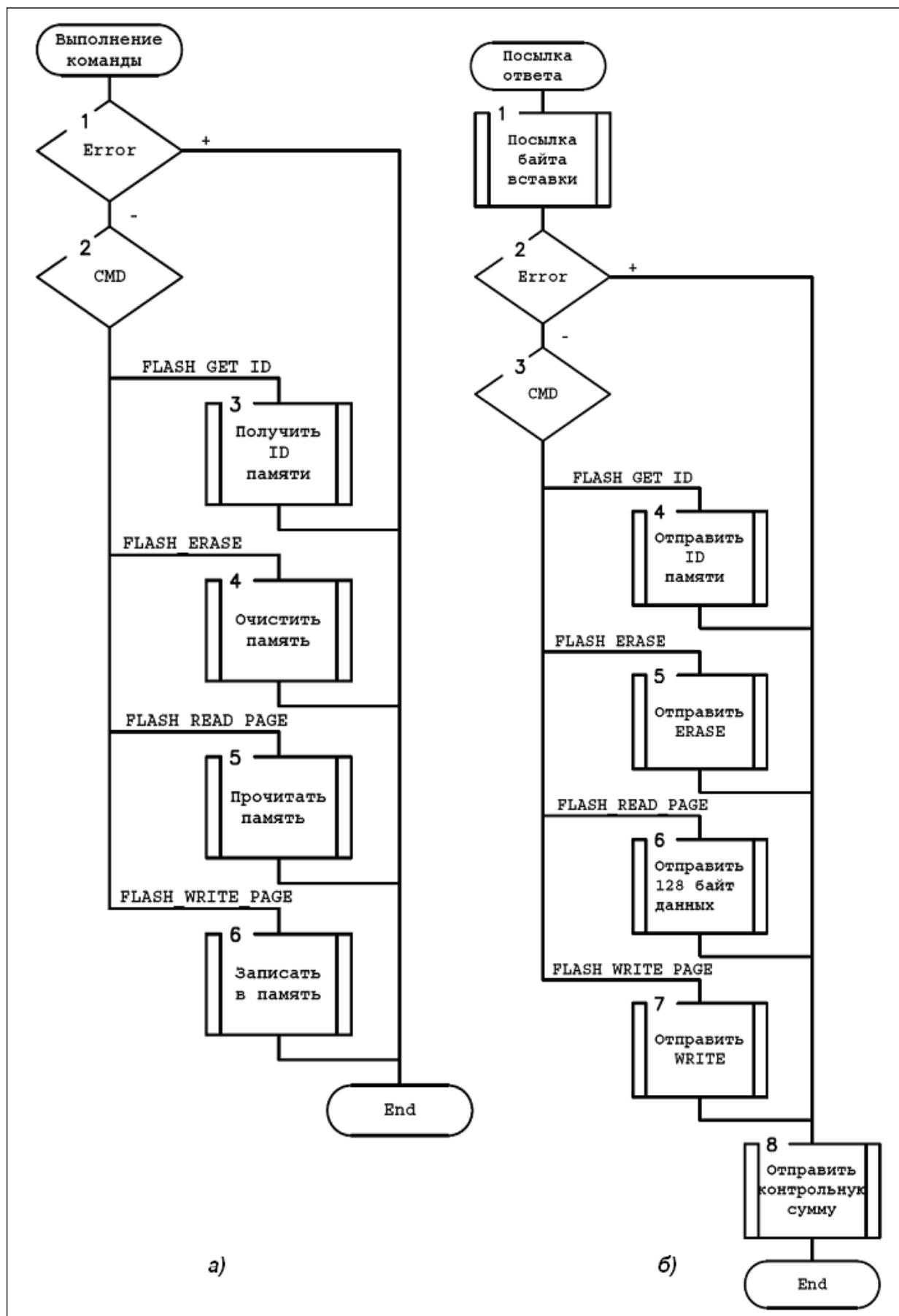


Рис. 20. Блок схема подпрограмм: а) «Выполнение команд» и б) «Посылка ответа»

3.2.4. Разработка программы для компьютера

Программа разрабатывалась в среде Microsoft Visual Studio Express Edition 2008, распространяемая бесплатно. При написании программы Programmer51 придерживались соглашения по оформлению кода группы RSDN.

Пользовательский интерфейс составлен на основе библиотеки wxWidgets. wxWidgets — это библиотека для написания приложений с графическим пользовательским интерфейсом (graphical user interface, GUI). Одной из особенностей, выгодно отличающих wxWidgets от других библиотек, является ее переносимость. Для выполнения на другой платформе необходима только перекомпиляция исходных кодов под целевую платформу (если в исходных кодах нет платформозависимого кода). Это обстоятельство дает огромное преимущество по сравнению с полным переписыванием приложения под каждую платформу и в том числе означает, что не нужно изучать различные API для каждой платформы.

Работа с коммуникационным портом, к сожалению в библиотеке не разработана и является специфичной для платформы. Поэтому, чтобы облегчить в будущем перенос программы под Linux все платформозависимые функции были вынесены в отдельный модуль.

Программа на стороне компьютера исполняет роль сервера. Ее основная функция – отправка команды пользователя и ожидание ответа от базового модуля. Базовый модуль имеет два режима работы. Это режим программирования и выполнения. При режиме программирования микроконтроллер выполняет работу по обслуживанию микросхемы flash-памяти. Для этого режима выделены следующие команды (все команды относятся к микросхеме памяти):

- Идентификация устройства и производителя.
- Стирание.
- Чтение содержимого устройства.
- Запись (программирование).
- Проверка содержимого памяти.

При режиме выполнения исполняется программа пользователя, записанная во внешней памяти программ (во время предыдущего режима). В этом случае наша программа используется для передачи по последовательному произвольных данных и приема информации от модуля. Форматы пакетов передачи задает уже сам пользователь, и их содержание зависит от решаемой задачи.

3.3. Руководство программиста

В главе приводятся описания классов и методов, используемых в Programmer51.

Программа написана на языке высокого уровня C++ с применением объектно-ориентированного подхода. В архитектуре приложения части представления и обработки данных разделены на слабо связанные компоненты. Интерфейсная часть представляется классом главного фрейма MainFrame:

class MainFrame : public wxFrame

Класс главного фрейма программы. Получается наследованием класса wxFrame библиотеки wxWidgets. Содержит методы обработки событий и выполняет основную логику Programmer51.

MainFrame(const wxString& title)

Конструктор класса. Создает элементы графического элемента и инициализирует внутренние переменные.

Параметры:

&title — название заголовка главного окна.

void OnOpenFile(wxCommandEvent&)

Обработчик команды «Открыть файл». Выводит стандартное окно выбора файла.

void OnFontConfig(wxCommandEvent&)

Обработчик команды «Настройки шрифта». Выводит стандартное окно настройки шрифтов.

void OnCOMConfig(wxCommandEvent&)

Обработчик команды «Настройки COM порта». Выводит стандартное окно настройки COM порта.

void OnClearLog(wxCommandEvent&)

Обработчик команды «Очистить лог».

void OnFlashRead(wxCommandEvent&)

Обработчик команды «Чтение flash». Выполняется чтение всей памяти flash путем передачи команд чтения и анализа приходящих ответов.

void OnFlashProg(wxCommandEvent&)

Обработчик команды «Программирование flash». Выполняется запись в память flash открытого hex файла.

void OnFlashCheck(wxCommandEvent&)

Обработчик команды «Сравнение flash». Выполняется чтение содержимого памяти flash и его сравнения с открытым hex файлом.

void OnFlashErase(wxCommandEvent&)

Обработчик команды «Стирание flash». Выполняется стирание памяти flash путем передачи соответствующей команды.

void OnFlashGetID(wxCommandEvent&)

Обработчик команды «Получение идентификатора flash». Выполняется передача команды для получения идентификатора flash-памяти. Полученное значение выводится на экран.

void OnQuit(wxCommandEvent&)

Обработчик выхода из приложения.

void OnAbout(wxCommandEvent&)

Обработчик команды «О программе». Выводит диалоговое окно с информацией о версии программы и его создателях.

void OnHelp(wxCommandEvent&)

Обработчик команды «Помощь». Вызывает справку о программе.

void OnButtonSaveLog(wxCommandEvent&)

Обработчик нажатия кнопки «Сохранить лог». Выводит стандартное диалоговое окно сохранения результатов. Содержимое лога сохраняется в файле. По умолчанию название файла log.log.

void OnButtonSend(wxCommandEvent&)

Обработчик нажатия кнопки «Отправить». Отправляет на последовательный порт введенных пользователем данных.

void OnCheckBoxCR(wxCommandEvent&)

Обработчик нажатия на флажок cbxCR.

void OnCheckBoxLF(wxCommandEvent&)

Обработчик нажатия на флажок cbxLF.

void OnNotebook(wxNotebookEvent&)

Обработчик переключения вкладок «Программирование» и «Интерфейс». Перенастраивает COM порт для каждого режима.

void OnResize(wxSizeEvent&)

Обработчик изменения размеров окна. Происходит перерисовка положений и размеров элементов графического интерфейса и перерасчет столбцов при отображении содержимого hex файла.

void OnTimer(wxTimerEvent&)

Обработчик прерывания от таймера. Происходит с периодом в одну секунду. Считывается содержимое буфера COM порта.

void OnUpdateUI(wxUpdateUIEvent& evt)

Обновление элементов главного меню и панели инструментов. В зависимости от режимов работы, некоторые команды отключаются.

void SetLogText(wxString text)

Записывает текст в лог файл с добавлением в начало времени и даты отправки его в лог.

Параметры:

text — строка для вывода в лог.

bool LoadHexFile(wxString filename, int column)

Загружает в окно отображения данных для прошивки из файла.

Параметры:

`filename` — название файла для загрузки.

`column` — количество столбцов при выводе.

При удачной загрузке возвращается `true`, в противном случае `false`.

`bool SetCOMPortInfo()`

Устанавливает в строку состояния информацию об открытом порте и его основные настройки.

`void Pause(int ms)`

Выдержка паузы.

Параметры:

`ms` — количество миллисекунд для задержки.

`bool ResponseParser(const unsigned char *response, int len, unsigned char command)`

Разбор принятого от базового модуля ответа.

Параметры:

`*response` — указатель на строку обрабатываемых символов.

`len` — длина `response`.

`command` — код команды, на которую пришел ответ.

При ответе, указывающем, что команда выполнена правильно, возвращается `true`.
В противном случае возвращается `false`.

Множество функций обработки сообщений от элементов управления со своей логикой составляют модель поведения приложения. Описания методов класса `MainFrame` не приведено по причине ограниченного объема расчетно-пояснительной записки. К тому же назначение метода легко угадывается по его названию.

Класс для работы с COM портом компьютера `SCOMM` выполняет всю работу по его обслуживанию. Работа с портом происходит в синхронном режиме с настроенными таймаутами операций чтения и записи.

class cCOMM

Класс для работы с последовательным коммуникационным портом COM персонального компьютера. Содержит методы для инициализации, настройке порта, приема и передачи данных.

cCOMM()

Конструктор класса. Открывает свободный COM порт и иницирует его стандартными настройками.

~cCOMM()

Закрывает COM порт.

bool isError()

Возвращает факт ошибки.

В случае ошибки при открытии порта или настройке, а также остальных видов произошедших ошибок, возвращается true. В противном случае возвращается false.

bool SetComConfig()

Выводит стандартное диалоговое окно настройки COM порта.

Возвращает true при успешной настройке, в противном случае возвращается false.

void SaveCurrentComState()

Сохраняет текущие настройки COM порта.

void RestoreComState()

Восстанавливает сохраненные настройки COM порта.

void SetStandardComState()

Устанавливает стандартные настройки COM порта.

int GetPortNumber()

Возвращает номер открытого COM порта (от 1 до 255).

int GetBaudRate()

Возвращает скорость обмена открытого COM порта, биты в секунду.

int GetByteSize()

Возвращает размер байта передаваемых символов.

int GetParity()

Возвращает признак метода контроля целостности данных.

int GetStopBit()

Возвращает количество стоповых бит, участвующих в передаче символа. При одном стоповом бите — 10, при полтора стоповых битах — 15, при двух стоповых битах — 20.

bool ResetComPort()

Сброс последовательного COM порта. Очистка очередей буферов приема и передачи данных.

При успешном завершении операции возвращает `true`. В противном случае возвращается `false`.

int WriteComPort(unsigned char *buffer, int bytes)

Передача по последовательному порту.

Параметры:

`*buffer` — указатель на последовательность символов для передачи по последовательному порту.

`bytes` — количество передаваемых символов.

Возвращает количество переданных данных.

int ReadComPort(unsigned char *buffer, int bytes)

Прием символов из последовательного порта.

Параметры:

`*buffer` — указатель на входной буфер, куда помещаются принятые данные.

`bytes` — размер буфера.

Возвращает количество прочитанных байт.

Класс `cHexParser` манипулирует с файлами формата HEX. Разбор файла, его обработка и последующее сохранение – важные действия в программе. Используемая память программ имеет объем 64 кБ. Большая его часть не используется. Для экономии памяти

внутренняя структура хранения данных (внутренний буфер) представляет собой список. Каждый элемент списка содержит данные по 256 байт. Многие методы класса работают с этим списком.

class cHexParser

Класс для работы с данными в формате hex. Предоставляет методы по загрузке hex-файлов и бинарных данных, форматированного вывода данных, сравнения содержимого и другие методы.

cHexParser()

Конструктор класса. Инициализирует внутренние переменные класса.

~cHexParser()

Деструктор класса. Высвобождает память от переменной, где хранится содержимое последнего загруженного файла HEX или бинарных данных.

int LoadHexFile(const wchar_t *FileName)

Загружает во внутренний буфер бинарные данные с файла формата HEX. Параметры:

*FileName — указатель на строку в формате UNICODE открываемого файла.

При удачном завершении возвращает количество данных, загруженных из HEX файла. В противном случае возвращает -1.

int LoadHexData(const char *hex, int len)

Загружает во внутренний буфер бинарные данные со входного буфера.

Параметры:

*hex — указатель на строку необходимых для загрузки бинарных данных.

len — длина входного буфера hex.

При удачном завершении возвращает количество данных, загруженных из HEX файла. В противном случае возвращает -1.

bool CompareToInternalBuffer(const char *hex, int len)

Сравнивает содержимое внутреннего буфера с заданным буфером.

Параметры:

*hex — указатель на строку сравниваемых бинарных данных.

len — длина входного буфера hex.

При совпадении входного буфера с внутренним, возвращается true. В противном случае возвращается false.

void GetErrorString(wchar_t *ErrorString, size_t size)

Формирует строку с содержанием об ошибке при обработке данных в классе.

ErrorString — указатель на строку с содержанием об ошибке.

size — размер передаваемой в функцию строки.

wchar_t *GetFormatMemoryTable(int cByteInLine, int &cSymbol)

Возвращает указатель на строку в формате UNICODE с отформатированными hex-данными.

Параметры:

cByteInLine — количество байтов в символьном hex представлении в строке.

cSymbol — возвращается размер отформатированных данных.

unsigned char *GetNextPageMemory(int page, int &addr)

Возвращает указатель на буфер, содержащей 256 байтов бинарных данных (одну страницу).

Параметры:

page — номер страницы для запроса данных.

&addr — базовый адрес выданной страницы.

bool SaveToFile()

Сохраняет содержимое внутреннего буфера в файл с именем “flash.bin”.

Возвращает true при успешном сохранении внутреннего буфера в файл. В противном случае возвращается false.

3.4. Выводы по главе

При разработке использовались современные средства и подходы к разработке ПО. При этом для Programmer51 учитывалась возможность переноса на платформу Linux. Это получается благодаря использованию библиотеки wxWidgets и выделении

платформозависимого кода в отдельные модули. Решена проблема совмещения в одной главной программе интерфейса для двух режимов работы базового модуля.

Модули разбора файлов формата hex написаны с учетом всевозможных типов записей по документации [7]. Применением односвязного списка при хранении бинарных данных, извлеченных из hex-файла или flash-памяти, решило проблему эффективного использования памяти.

Разработанный протокол позволяет добавлять дополнительные команды с небольшими изменениями в тексте программы Embedded51. Правила, введенные для контроля целостности и повторные вызовы неудачно выполненных команд без разрешения пользователя, добавляют помехозащищенность и удобство при работе с программой. Протокол отвечает требованиям по надежности и прост в реализации.

Программа Embedded51 занимает память 997 байт (при доступных 2048 байт). Прием, обработка и передача ответа разделены по времени (выполняются последовательно). Это исключает случайные длительные задержки при приеме-передаче данных по последовательному порту.

Базовый модуль имеет размер 120 мм на 150 мм. На площади 1,8 дм² получилось разместить все разъемы, микросхемы, элементы индикации и управления. Получилась логическая группировка одинаковых функциональных элементов по группам соединение, управление/индикация и микроконтроллер с памятью.

4. Экспериментальная часть

В экспериментальной части приведены результаты эксперимента внедрения учебного модуля в учебный процесс кафедры вычислительной техники. Приведено руководство пользователя по РД 50–34.698–90 для автоматизированных систем.

4.1. Руководство пользователя

4.1.1. Введение

4.1.1.1. Область применения

Учебный модуль предназначен для применения в образовательном процессе для разработки и отладки программ на 8-разрядных микроконтроллерах. Рекомендуется использовать в качестве макета при проведении лабораторных работ и курсового проектирования студентами высших учебных заведений при изучении дисциплины «Микропроцессорные средства и системы» или аналогичной ей, а также при проведении научно-исследовательской работ студентами высших учебных заведений.

4.1.1.2. Краткое описание возможностей

Учебный модуль позволяет разрабатывать программы под микроконтроллеры ATmega8515 и AT89S52. Модуль имеет внешнюю память ОЗУ объемом 64 Кбайта и ППЗУ объемом 64 Кбайта. Программирование устройства осуществляется без извлечения микросхемы.

4.1.1.3. Уровень подготовки пользователя

Пользователь должен быть знаком с основами электротехники и цифровой электроники (прослушать курс лекций по дисциплине «Основы электротехники» или аналогичной ей).

Перед использованием модуля рекомендуется изучить данное руководство.

К работе с модулем допускается персонал, прошедший инструктаж по технике противопожарной и электробезопасности.

Для успешного использования учебного модуля и понимания его принципов работы предлагается ознакомиться с принципиальной схемой базового и дополнительных модулей, а также ознакомиться с технической документацией на применяемые компоненты.

Также пользователь должен иметь навыки работы с ОС Windows на уровне опытного пользователя.

4.1.1.4. Перечень эксплуатационной документации

Все эксплуатационные документы включены в комплект учебного модуля. Сюда входят:

1. Принципиальные схемы на каждый модуль.
2. Сборочные чертежи на каждый модуль.
3. Данное «Руководство пользователя».

4.1.2. Назначение и условия применения

4.1.2.1. Виды деятельности, функции

Данное изделие применяется как отладочная плата с целевым микроконтроллером ATmega8515 или AT89S52 с возможностями внутрисхемного программирования.

4.1.2.2. Программные и аппаратные требования к системе

Для прошивки программы в учебный модуль необходим IBM PC совместимый компьютер с установленной операционной Windows 9x/NT/2000/XP/Vista (64-разрядные версии операционных систем не поддерживаются). Минимальные и рекомендуемые требования к аппаратной части определяются установленной операционной системой. Помимо этого на аппаратную часть накладываются следующие дополнительные требования:

- наличие COM порта с интерфейсом RS-232;
- наличие LPT порта.

Дополнительно на компьютере должно быть установлено следующее программное обеспечение:

- VC 2008 Runtime Redistributable Package (поставляется в комплекте);
- PonyProg v2.07 (поставляется в комплекте).

4.1.3. Подготовка к работе

4.1.3.1. Состав дистрибутива

В таблице 5 представлена комплектация учебного модуля.

Дополнительно на диске прилагаются:

- установочный файл ПО Programmer;

- VC 2008 Runtime Redistributable Package x86;
- установочный файл ПО PonyProg;
- тестовые программы с примерами подключения модулей;
- принципиальные схемы на модули;
- сборочные чертежи на модули;
- видеоролик с примером работы с модулем;
- данное руководство в электронном виде;

Таблица 5. Состав учебного модуля

№ п/п	Комплектуемое	Количество
1	Базовый модуль	1
2	Конвертер сигналов RS232 – TTL.	1
3	Дискретные датчики.	1
4	Часы реального времени на DS1306	1
5	Перенаправитель портов.	1
6	Линейка светодиодов.	1
7	Выход на клеммники.	1
8	Семисегментный индикатор.	1
9	Клавиатура.	1
10	ЖК индикатор МТ-10Т7.	1
11	ЖК индикатор WH1604.	1*
12	Термодатчик DS18B20	2*
13	Кварц, 12 МГц	1
14	Кварц, 16 МГц	1
15	Кварц, 11,0592 МГц	1
16	Шлейф соединительный	5
17	Нуль-модемный кабель DB-9М 1,8 м	1
18	Кабель удлинительный LPT 1,8 м	1*
19	Программатор AVR	1*
20	Микросхема АТmega8515	1
21	Микросхема АТ89S52	1
22	Микросхема К555КП11	1
23	Микросхема SST29EE010	1

24	Микросхема НМ6264LP	1
25	Микросхема SN74HC573AN	1
26	Коробка	1
27	Диск с ПО и документацией	1*
28	Книжка «Руководство пользователя»	1*

* - количество указано на партию

4.1.3.2. Запуск системы

Базовый модуль представляет собой вычислительное устройство на основе микроконтроллера AT89S52 (ATmega8515), дополненного внешней памятью программ и данных. Микросхемы имеют все внешние цепи обвязки, необходимые для его нормальной работы. Рассмотрим состав элементов и их значение в базовом модуле подробнее.

На рис. 21 представлена монтажная схема платы базового модуля. Каждый элемент снабжен обозначениями, указывающие его функциональное назначение и порядковый номер.

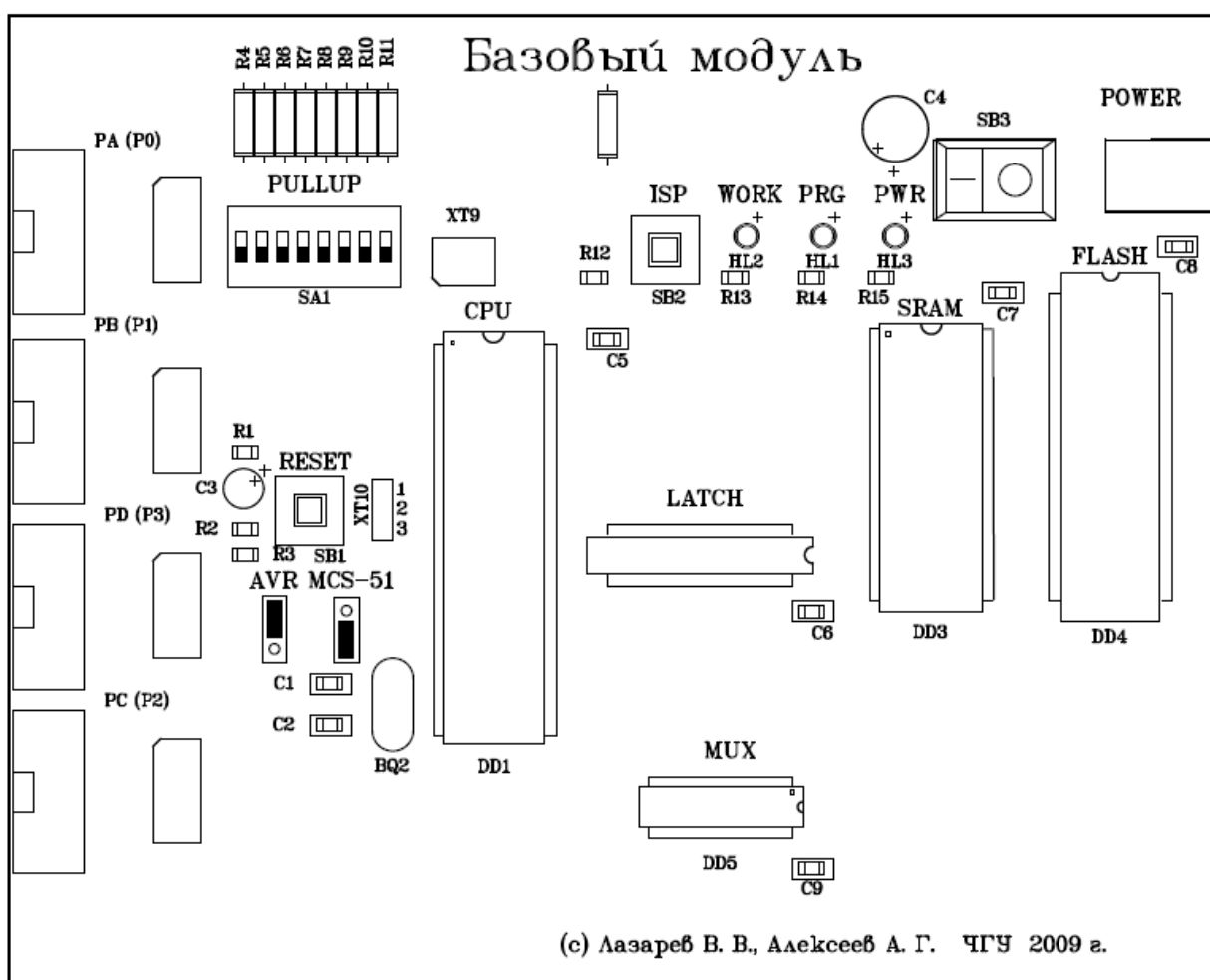


Рис. 21. Монтажная схема базового модуля

Питание от внешнего источника подается через разъем POWER. Выключатель SB3 управляет подачей питания на всю плату. Присутствие питания на плате обнаруживается свечением светодиода HL3 (PWR).

Центральной частью на плате является микросхема DD1 (CPU). Это 40-выводной микроконтроллер AT89S52 (ATmega8515). Внешняя память программ DD4 (FLASH) на основе микросхемы flash-памяти SST29EE010 и память данных DD3 (SRAM) на основе микросхемы статической памяти HY6264LP подключены к микроконтроллеру через регистр-защелку DD2 (LATCH) SN74HC573AN. Мультиплексор DD5 (MUX) 555КП11 конфигурирует управляющие сигналы при различных режимах работы. При использовании ATmega8515 эти микросхемы на базовый модуль не устанавливаются. Все четыре порта микроконтроллера выведены на левый край платы (по два разъема на каждый порт – угловой и прямой). Это позволяет во многих случаях обойтись без разветвителей и подключать к порту сразу два дополнительных устройств. Порт PB(P1) дополнен возможностью подключения внешних подтягивающих резисторов R4 – R11 через переключатель SA1. В разъем BQ2 устанавливается кварцевый резонатор необходимой частоты из числа поставляемых в комплекте.

Цепь сброса должна быть сконфигурирована под выбранный микроконтроллер с помощью перемычки XT10. Для ATmega8515 (AVR) замыкаются контакты 1 и 2 разъема XT10, а для AT89S52 (MCS-51) — 2 и 3. Кнопка SB1 (RESET) сбрасывает модуль в начальное состояние.

Переключение режимов выполняется через переключатель SB2 (ISP). Текущий режим указывается светодиодами HL1 (WORK) и HL2 (PRG). Свечение светодиода HL1 (WORK) указывает то, что модуль находится в режиме выполнения, а свечение светодиода HL2 (PRG) — в режим программирования. Светодиоды HL1 и HL2 горят попеременно, если ни один из них не горит или горят оба вместе, значит модуль неисправен.

При использовании в модуле микроконтроллера ATmega8515 прошивка внутренней памяти программ осуществляется через разъем внутрисхемного программирования XT9.

Подготовка учебного модуля к использованию

Перед использованием учебный модуль необходимо осмотреть визуально на обнаружение дефектов, способных вызвать выход из строя модуль. Только при удостоверении в целостности используемых модулей можно начать их использование. В противном случае попытаться исправить выявленный недостаток.

1. Осмотр дорожек печатных плат на отсутствие нежелательных замыканий и целостность проводников, отсутствие следов короткого замыкания и т.п.

2. Правильность установки микросхем в колодки по ключу и монтажной схеме.
3. Положение джампера на разъеме XT10 соответствует используемому микроконтроллеру.
4. Кнопка выключения питания SB3 в выключенном состоянии.

Подобным образом осматриваются используемые дополнительные модули.

В зависимости от использования в схеме микроконтроллера ATmega8515 или AT89S52 в колодки DD1 – DD5 устанавливаются микросхемы в соответствии с таблицей 6.

Таблица 6. Соответствие микросхем

Условное обозначение	Название микросхем при использовании AT89S52	Название микросхем при использовании ATmega8515
DD1	AT89S52	ATmega8515
DD2	SN74HC573AN	—
DD3	HY6264LP	—
DD4	SST29EE010	—
DD5	555КП11	—

При установке необходимо следить за положением ключа микросхемы и колодки. При неправильной установке есть вероятность выхода из строя устройства при включении питания.

Блок питания вставляется в розетку, разъем питания подключается к разъему POWER базового модуля. При включении выключателя SB3 должен загореться светодиод HL3 (PWR). Отключаем питание по выключателю SB3. Переходим к программированию устройства.

Проверка работоспособности системы

Проверка работоспособности всего изделия проводится пакетом специально подготовленных программ и схем подключения модулей. Программы охватывают все дополнительные модули и микроконтроллеры базового модуля. Программы находятся в каталоге samples с установленной программой.

4.1.4 Описание операций

4.1.4.1. Программирования для ATmega8515

На компьютер предварительно должна быть установлена программа PonyProg, входящая в состав прилагаемого диска. При отсутствии на компьютере разъема LPT

параллельного порта запрограммировать устройство не удастся. Требование присутствие в персональном компьютере разъема такого типа при использовании базового модуля с микроконтроллером серии AVR обязательно. Кабель с LPT-разъемами одним концом подключается к компьютеру, а в другой конец подключается сам программатор. Разъем IDC10 программатора вставляется в разъем XT9 базового модуля. Необходимо удостовериться в том, что навесные резисторы отключены от порта PB через SA1 (состояние OFF переключателей). Разъем порта PB(P1) должен быть свободным. Это связано тем, что при программировании задействованы порты PB микроконтроллера. На модуль подается питание через выключатель SB3.

Через меню **ПУСК** → **Все программы** → **PonyProg** запускаем программу, с помощью которого и происходит программирование базового модуля (микроконтроллера ATmega8515).

Настройка PonyProg

На рис. 22. приведено изображение главного окна программатора PonyProg.

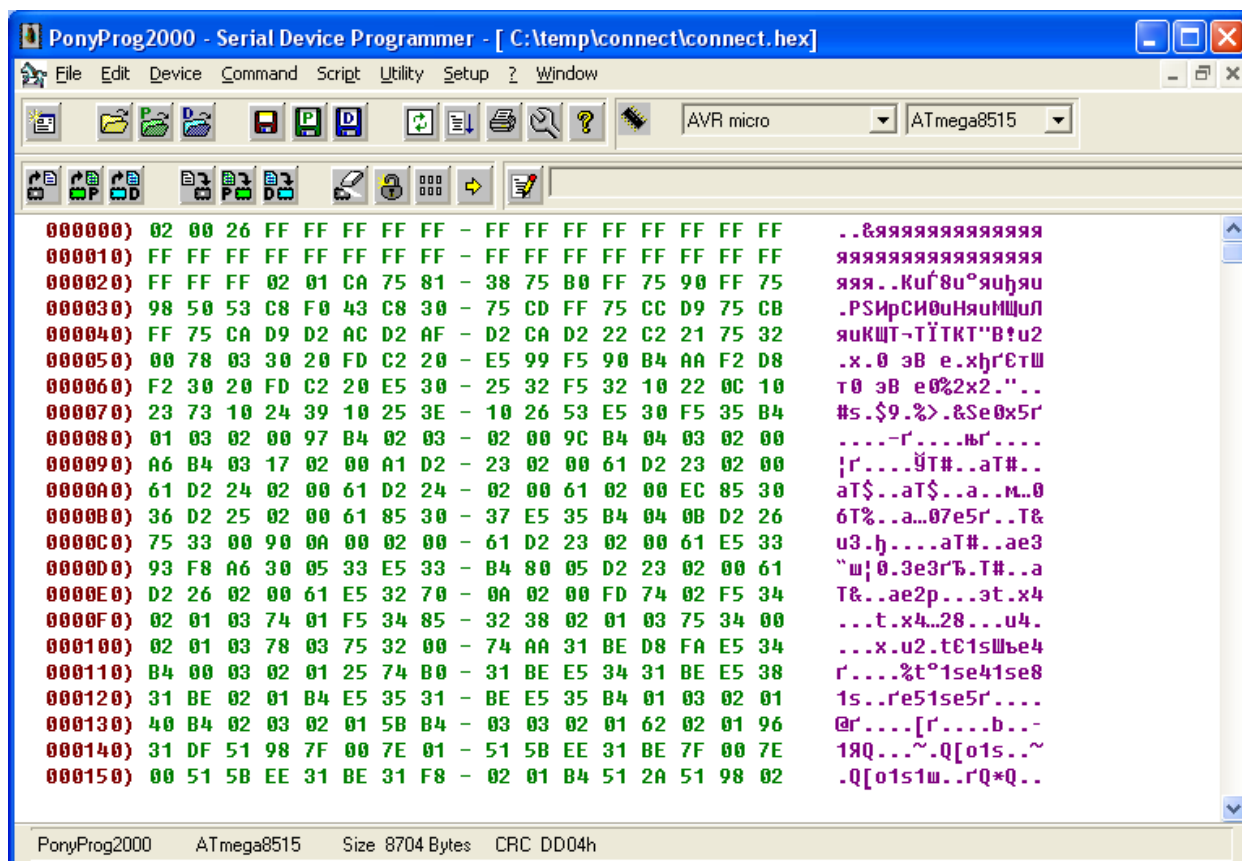


Рис. 22. Главное окно программы PonyProg

В меню **Setup** выбираем **Interface Setup...** для настройки интерфейса параллельного порта. Появляется диалоговое окно настройки **I/O port setup** (см. рис 23.).

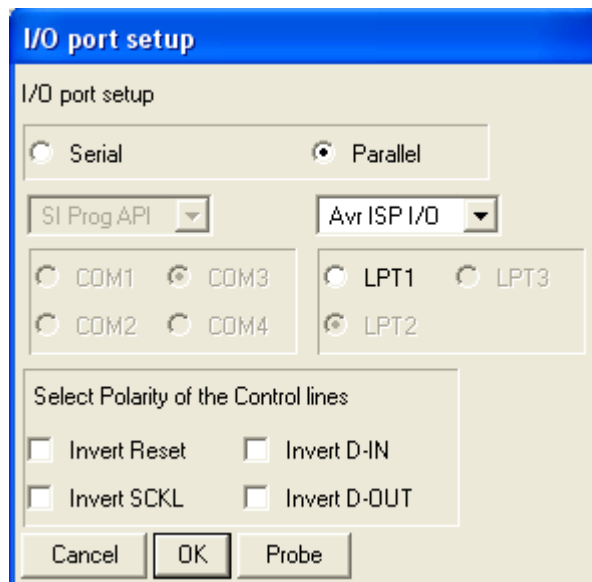


Рис. 23. Диалоговое окно «I/O port setup»

Из группы переключателей **I/O port setup** выбрать **Parallel**. Из выпадающего списка выбрать **Avr ISP I/O** и выбрать активизировавшийся **LPT1**. Если переключатель LPT1 не активизируется, то имеются проблемы с LPT портом. Сохраняем настройки нажатием кнопки OK. Калибровка временных интервалов для настройки порта выполняется через меню Setup → Calibration. Появляется диалоговое окно **Yes or No** (см. рис. 24).

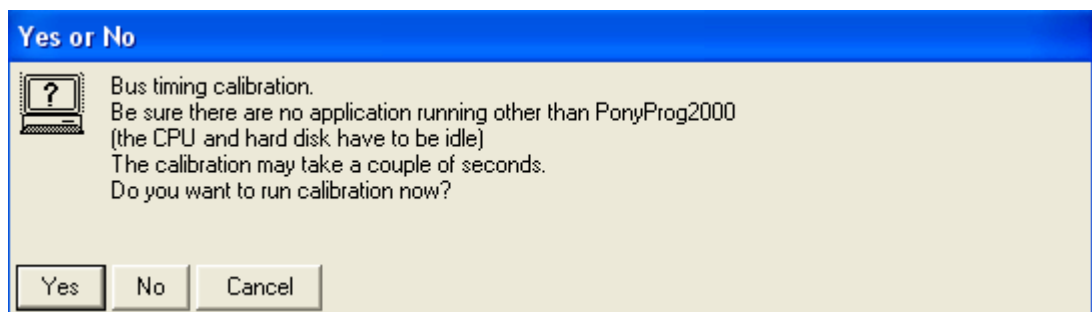


Рис. 24. Диалоговое окно «Yes or NO»

Соглашаемся с выбранным вариантом: **OK**. Завершение калибровки указывается выводом диалогового окна **Notice (Calibration OK)**. Выбор устройства для программирования проводится через меню **Device → AVR micro → ATmega8515**.

Команда через меню **File → Open Program (FLASH) File** выбирается файл для прошивки в программную флеш-память. Основные команды для работы с программатором находятся в меню **Command**. Рассмотрим их.

Read All — считывание данных из памяти программ и EEPROM.

Read Program (FLASH) — считывание данных из памяти программ.

Read Data (EEPROM) — считывание данных из памяти EEPROM.

Write All — запись данных в память программ и EEPROM

Write Program (FLASH) — запись данных в память программ.

Write Data (EEPROM) — запись данных в память EEPROM.

Verify All — сравнение данных из памяти программ и EEPROM с внутренним буфером

Verify Program (FLASH) — сравнение данных из памяти программ с внутренним буфером.

Verify Data (EEPROM) — сравнение данных из памяти EEPROM с внутренним буфером.

Security and Configuration Bits... — управление с битами конфигурации и битов защиты.

4.1.4.2. Программирование для AT89S52

На компьютер предварительно должна быть установлена программа Programmer51, входящая в состав прилагаемого диска. Программирование базового модуля с микроконтроллером AT89S52 происходит через последовательный порт COM. Нуль-модемный кабель подключается к COM порту компьютера и конвертеру уровней RS232/TTL. Конвертер через шлейф подключается к разъему P3. При программировании базового модуля используются все порты микроконтроллера. Поэтому необходимо удостовериться в том, что навесные резисторы отключены от порта PB через SA1 (состояние OFF переключателей) и разъемы портов P0, P1, P2 и P3 свободны. Включается питание модуля через выключатель SB3. Переключателем SB2 (ISP) переходим в режим программирования, при этом должен гореть светодиод HL1 (PRG).

Через меню **ПУСК** → **Все программы** → **Programmer51** запускаем программу, с помощью которого и происходит общение с базовым модулем (флеш-памяти SST29EE010). Для программирования устройства выбираем вкладку «Программирование» (рис. 25).

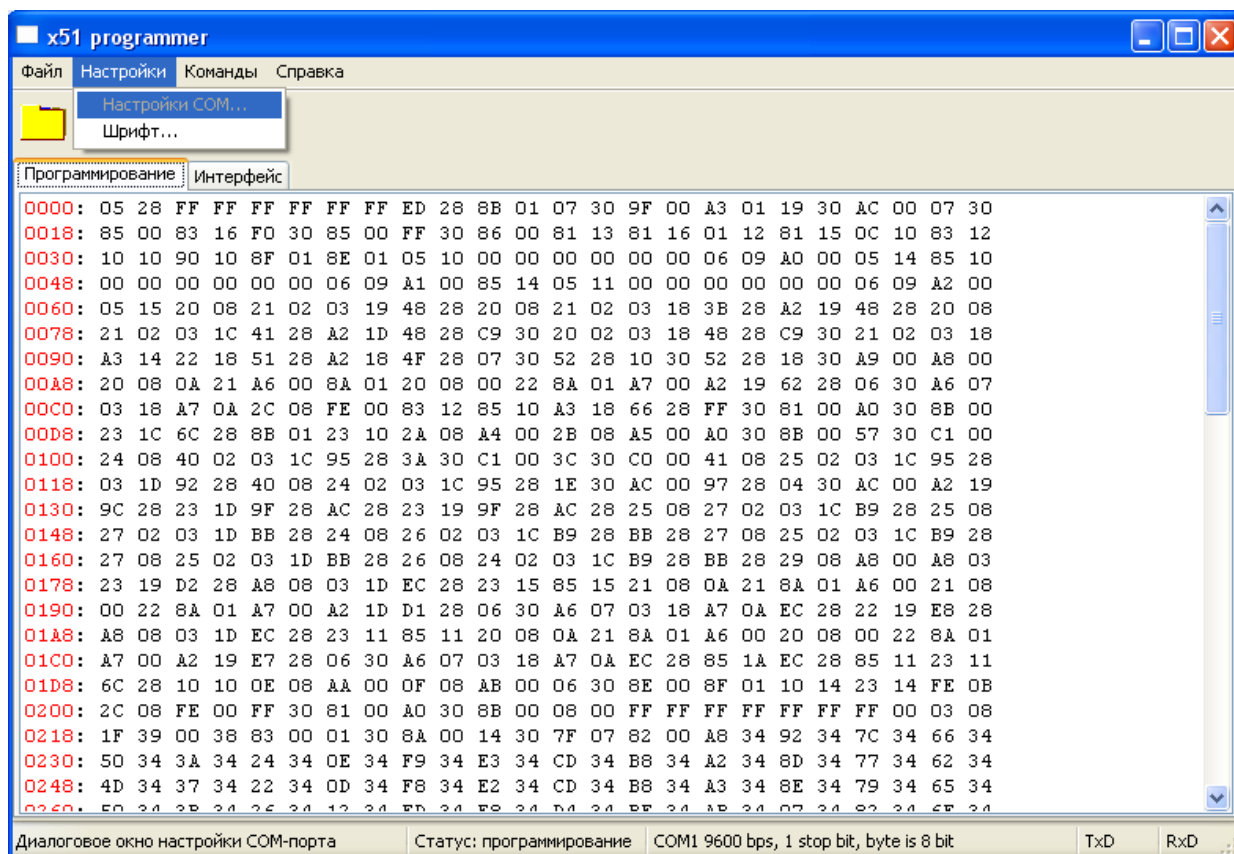


Рис. 25. Внешний вид окна режима программирования

При первом запуске программы она выбирается по умолчанию. Для определения успешности соединения компьютера с базовым модулем выберем команду из меню **Команды** → **Get Flash ID**. В случае обнаружения внешней памяти программ SST29EE010 появляется сообщение с информацией о типе микросхемы и производителе. В ином случае сообщение **Error «Ошибка при приеме данных»**. Возможные причины возникновения ошибки:

- не подключен нуль-модемный кабель;
- не подано питание;
- неправильно сконфигурирован или вышел из строя базовый модуль;

Если внешнее устройство идентифицировалось правильно, можно выполнить программирование модуля. Рассмотрим команды.

Erase — быстрое стирание памяти программ

Read — чтение содержимого памяти программ

Program — программирование памяти программ выбранным HEX файлом.

Check — сравнение содержимого памяти программ с внутренним буфером программы.

Все действия пользователя фиксируются в логге. Лог можно просмотреть на вкладке «Интерфейс». Питание выключается через выключатель SB3. Переходим к сборке модели и его запуску.

4.1.4.3. Сборка модели и его запуск

Сразу после программирования базового модуля рекомендуется переключить SB2 (ISP) для выбора режима выполнения.

Подключение дополнительных модулей к базовому осуществляется по принципиальной схеме пользователя при выключенном питании. Перед включением питания необходимо тщательно проверить собранное устройство на отсутствие в нем нежелательных связей, приводящих к выходу из строя всего модуля или его отдельных частей. Подключения навесных резисторов PULLUP должно также соответствовать принципиальной схеме. Перед подачей питания на плату необходимо показать собранный модуль преподавателю. Питание на плату подается по кнопке SB3. Должен загореться зеленый PWR (HL3) и синий HL2 (WORK) светодиоды. Происходит выполнение программы пользователя. При возникновении неполадок, способных вывести модуль из строя, рекомендуется незамедлительно выключить питание через выключатель SB3.

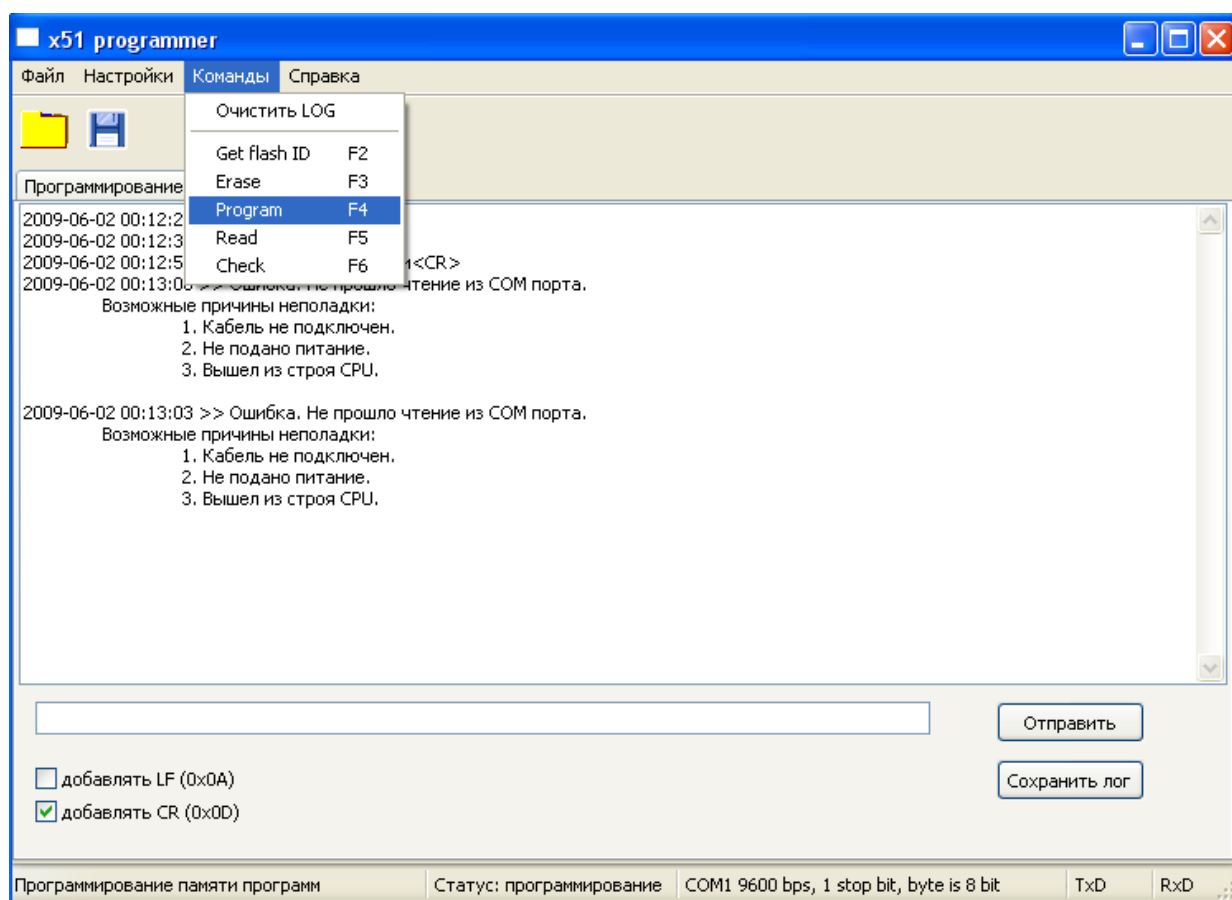


Рис. 26. Внешний вид окна режима выполнения

Интерфейс логически разделен на две части с помощью элемента управления вкладка. На первой вкладке размещаются элементы управления, связанные с режимом программирования. Рабочую область окна полностью занимает отображение памяти. Открытые файлы формата HEX, прочитанные из памяти данные заполняют эту область. Каждая строка начинается с четырехбайтного числа - адреса данных, следующих за символом «:». Адреса данных и сами данные выделяются различным цветом. Описанные выше четыре команды размещены на панели задач и продублированы в меню (в пункте «Команды»). В строке состояния указываются текущие настройки COM-порта, помощь о выбранных пунктах меню, элементах управления. Отдельно показывается статус – режим работы базового модуля. Для слежения за состоянием последовательного порта в строке состояния имеется синий и красный кружки. Их мигание показывает что делает в данное время порт – передает или принимает.

Вторая вкладка представляет нам возможность передавать произвольные данные и принимать их от модуля. Весь сеанс связи протоколируется. Указывается дата и время отправки данных и содержимое отправленного пакета. Передаваемые данные набираются в текстовой строке. Для включения к данным символов конца строки и возврата каретки предусмотрены флажки «добавлять CR (0x0D)» и «добавлять LF (0x0A)». Кнопка «Отправить» передаст сформированные данные по последовательному порту. В остальных случаях программа находится в режиме ожидания данных от модуля. Кнопка «Сохранить лог» сохраняет протокол приема и передачи в файл для последующего анализа его студента и преподавателя. На рис. 25 и 26 представлены внешний вид главного окна программы при режимах программирования и выполнения. Они наглядно иллюстрируют вышесказанное.

4.1.5. Аварийные ситуации

При работе с Programmer51 могут возникнуть ошибки. Такие ошибки в большем случае связаны с работой COM портом. Это нарушение целостности пакетов при приеме/передаче данных. Вот список возможных ошибок, выдаваемых Programmer51.

- Ответ слишком короткий.
- Байты-вставки не верны.
- Ответ не на тот запрос.
- Ответ не соответствует длине.
- Отправленный код команды не определен.
- Отправленная контрольная сумма не верна.

- Код ошибки не определен.
- Код ответа не определен.
- Принятая контрольная сумма не верна.
- Не прошло чтение из СОМ порта.
- Код производителя %X (не определен).
- Ошибка при приеме данных.
- Не прошла запись в СОМ порт.
- Для прошивки необходимо открыть файл.
- Ошибка сохранения в файл.
- Ошибка при открытии файла.
- Пустая строка для отправки по СОМ порту.

4.1.6. Рекомендации по освоению

Для успешного освоения учебного модуля необходимо иметь навыки работы с ПК на уровне опытного пользователя ОС Windows и изучить следующее:

- настоящее «Руководство пользователя»;
- основы цифровой электроники;
- принципиальные схемы на модули.

При написании программ на микроконтроллер AT89S52 или ATmega8515 рекомендуется ознакомиться с технической документацией на соответствующее устройство. Прочитать подробнее о настройке и работе с применяемыми периферийными устройствами. На официальном сайте компании Atmel (www.atmel.com) имеются примеры программ на микроконтроллеры.

4.2. Применение модуля в учебном процессе

В 2009 году было отобрана экспериментальная группа из 6 студентов 3-го курса для выполнения курсового проектирования по дисциплине «Микропроцессорные средства и системы» с использованием разработанного изделия. По результатам работы с ними получены экспериментальные данные, которые были внесены в таблицу 7.

Таблица 7. Данные об экспериментальной группе

<i>Студенты</i>	<i>А</i>	<i>Б</i>	<i>В</i>	<i>Г</i>	<i>Д</i>	<i>Е</i>
<i>Задание</i>	Подключение РС клавиатуры к МК	Велосипедный компьютер	Цифровой термометр	Будильник	Терминал с интерфейсом RS232	
<i>Используемые модули</i>						
<i>Конвертер сигналов RS232 – TTL.</i>					+	
<i>Дискретные датчики.</i>		+	+			
<i>RTC DS1306</i>				+		
<i>Перенаправитель портов.</i>	+	+				
<i>Линейка светодиодов.</i>				+		
<i>Выход на клеммники.</i>	+	+	+			
<i>Семисегментный индикатор.</i>	+			+		
<i>Клавиатура.</i>				+	+	
<i>ЖК индикатор MT-10T7.</i>			+			
<i>ЖК индикатор WH1604.</i>		+			+	
<i>Термодатчик DS18B20</i>			+			
<i>тип МК</i>	8052	8052	AVR	AVR	AVR	8052
<i>число часов работы с модулем</i>	7	5	7	7	10	
<i>успешность выполнения работы</i>	+	+	+	+	+	-

4.3. Улучшение по результатам эксплуатации модуля

В ходе практического применения были замечены следующие недостатки, которые в следующей версии изделия будут исправлены:

1. Отсутствие подтягивающих резисторов на порту 0. Проявляется только в режиме программирования — устраняется программно или путем подключения к базовому модулю дискретных датчиков, которые служат подтягивающими резисторами по 4,7 кОм при состоянии датчиков OFF.

2. Невозможность установки на базовый модуль колодки с нулевым усилием. обусловлен неудобством замены микросхем при переконфигурировании модуля AVR/MCS-51. При неаккуратной замене существует вероятность механического повреждения модуля и микросхем.

3. Возможность неправильного включения термодатчика. При неправильном включении термодатчик выходит из строя, базовый модуль при этом не повреждается. Решением было введение организационных мер при работе с модулем.

4. Отсутствие светодиодов на плате конвертера для индикации активности обмена (прием/передача).

5. Плата конвертера разведена для устройства типа slave, что требует разных типов кабелей для подключения модуля к компьютеру и подключения модулей между собой. Для корректной работы с данной версией разводка кабелей была изменена.

6. Нельзя менять кварцы. Применением специальным образом обработанной цанговой колодки микросхемы проблему удалось решить.

7. Неправильная разводка базового модуля. Ошибка с сигналом ALE и переключателем в цепи сброса.

8. Разъем для программирования AVR допускал неправильное включение. Использование разъема с ключом, исключает неправильное подключение.

9. Недостаток розеток на кафедре для блоков питания.

Недостатки под номерами 1, 2, 4, 5, 7, 8 следует принять во внимание при разработке следующей версии учебного модуля. Проблема 9 решается путем установки дополнительного сетевого фильтра с несколькими розетками или при использовании удлинителей. Следует обратить внимание на то, что корпус блока питания модуля имеет большой объем. Поэтому сразу два блока питания в рядом находящиеся розетки не влезают.

4.4. Выводы по экспериментальной части

Модуль успешно опробован в учебном процессе. На момент написания отчета из шести студентов выполнили свою работу пятеро. В ходе применения модуля в учебной практике выявлены недостатки. Большинство улучшений были проведены в рамках текущей версии модуля. Полученные результаты проанализированы, и по ним будут приняты решения по улучшению качества изделия в следующей версии.

В следующем учебном году предполагается включить в цикл лабораторных работ по дисциплине «Микропроцессорные средства и системы» задания, в которых используется учебный модуль. Для этого предполагается написать методическое пособие по изготовленному изделию. Составить варианты заданий и, если это потребуется, создать новые дополнительные модули. Например: АЦП/ЦАП, интерфейс Ethernet или шаговые двигатели. Они увеличат количество заданий до необходимого количества.

5. Организационно-экономическая часть

Современный мир все более зависит от систем, построенных на основе вычислительной техники. Все больше технических устройств включают в себя элементы вычислительной техники и соответствующего управляющего программного обеспечения (ПО) в той или иной форме. В таких системах стоимость ПО порой составляет большую часть общей стоимости. Важным этапом в разработке любого программного изделия является планирование графика работы и стоимости разработки. Правильная оценка в начале проекта ресурсов, необходимых для завершения дела, позволит выполнить работы по плану и уложиться в предоставленный срок и бюджет.

Цель организационно-экономической части состоит в оценке трудоемкости дипломного проекта, что даст построить график выполнения работ, окончание которого необходимо запланировать к сдаче проекта на проверку и утверждение к концу мая – началу июня. В этой части также предоставлен анализ конкурентоспособности конечного продукта по ценовым и качественным характеристикам.

5.1. Цель создания учебного модуля

Основным требованием современного рынка труда остается опыт работы и ориентирование работника в современных технологиях и методологиях. Развитие науки и техники сегодня имеет все большие темпы чем когда-нибудь. Сроки от начала разработки до выхода новой продукции сжимаются, все большие темпы развития дают конкуренция на рынках. Таким образом, университетам поставлена цель подготовить специалистов, отвечающим всем требованиям современного мира техники. Теоретические знания, предоставляемые студентам необходимо подкреплять практическими занятиями. Техническая база университета должна приближаться к базе выпускаемым сегодня технических средств.

Цель дипломной работы — проектирование учебного модуля на базе популярных микроконтроллеров и программного обеспечения для него для ознакомления студентов элементами сложных технических средств. Учебный модуль предназначается для сдачи курсовых работ и применении в научно-исследовательских работах. При его проектировании одним из основных требований было минимальная себестоимость, позволяющая сделать продукт доступным другим кафедрам и учебным заведениям.

5.2. Описание программного изделия

Изделие состоит из аппаратной и программной части. Программная часть применяется для организации связи с аппаратной частью.

Учебный модуль на базе микроконтроллера серии MCS-51 имеет все свойства отладочных модулей. Отладочный модуль применяется в ознакомлении и обучении программированию микроконтроллеров. Также он широко применяется в промышленных разработках для быстрой отладки разрабатываемой продукции. Такое применение эффективно экономически и по времени. Это дает избежать неправильных схемотехнических решений и сэкономить время на ожиданиях заказанных изделий (платы, радиоэлементов).

Разрабатываемое изделие имеет свои характерные особенности, которые не присутствуют в изделиях, продаваемых на рынке. Конфигурируемость, наращиваемость и безопасность – основные требования, которые соблюдались в самом начале проектирования. Разделение учебного модуля на базовую и дополнительную дает возможность применять их в разных сочетаниях. Замена микроконтроллера новыми моделями не коснется дополнительных. Также к микроконтроллеру могут поставляться сторонними разработчиками другие устройства периферии. Для совместимости частей стандартизован только интерфейс соединения.

Опишем преимущества изделия:

1. Электробезопасность.
2. Низкая себестоимость.
3. Конфигурируемость.
4. Наращиваемость.
5. Замена в базовом модуле микроконтроллера другой серией (ATmega8515)

Характерной особенностью базового модуля с микроконтроллером AT89S52 является внешняя память программ и данных. На данный момент нами такой продукции не обнаружено. Это объясняется его специфическим применением.

5.3. Обзор рынка сбыта и аналогичной продукции

Большое количество встроенных периферийных устройств в микроконтроллерах и разнообразные внешние устройства не позволяют учесть все возможные комбинации «микроконтроллер + внешние устройства». Поэтому ассортимент такой продукции

большой и привязан к конкретным сферам разработки. Точно аналогичного продукта найти невозможно, поэтому в данном разделе рассматриваются только приближающиеся по своим характеристикам устройства.

На рынке поддержки разработок можно выделить две группы компаний. В первую входят производители микроконтроллеров. Отладочные платы, привязанные к определенной модели или серии, предназначены для популяризации выпускаемой продукции и повышению уровня продаж за счет их применения в новых серийных разработках. К таким относятся ATSTK-500 от Atmel, PICDEM от Microchip и другие.

Ко второй относятся сторонние производители, занятые разработкой отладочных плат и программ к ним для большого числа устройств. В таблице предоставлен список компаний и адреса их сетевых ресурсов.

Таблица 8. Производители отладочных плат

<i>Название</i>	<i>Сетевой ресурс</i>
mikroElektronika	www.mikroe.com
ProPox	www.propox.com
Olimex	www.olimex.com
Chip45	www.chip45.com
BlackHawk	www.blackhawk-dsp.com
Elprotronic	www.elprotronic.com
Embedded Artists	www.embeddedartists.com

Отдельно рассмотрим продукцию mikroElektronika и Atmel, которые пользуются популярностью и имеют конкурирующие цены.

В технической части отчета в разделе «Обзор современных средств разработки для 8-разрядных микроконтроллеров» были рассмотрены технические характеристики STK-AT89C5130 программатора-отладчика, и универсальных плат ATSTK500 от Atmel для серии МК AVR и ME-UNI-DS3 от mikroElektronika. Список цен на отладочные платы для нас является ориентировочной, которые мы не должны превышать, при этом функциональность должна оставаться примерной такой же.

5.4. Организация и планирование процесса разработки ПП

5.4.1. Формирование состава выполняемых работ

Стадии разработки, этапы и содержание работ соответствуют ГОСТ 19.102-77 (ЕСПД, стадии разработки).

Техническое задание

1. Постановка задачи
2. Сбор исходных материалов
3. Выбор и обоснование критериев эффективности и качества разрабатываемой программы.
4. Определение структуры входных и выходных данных.
5. Предварительный выбор методов решения задач.
6. Определение требований к техническим средствам.
7. Разработка технико-экономического обоснования разработки программы.
8. Определение стадий, этапов и сроков разработки программы и документации на неё.
9. Выбор языков программирования.

Эскизный проект

10. Предварительная разработка структуры входных и выходных данных.
11. Уточнение методов решения задачи.
12. Разработка общего описания алгоритма решения задачи
13. Разработка технико-экономического обоснования.

Технический проект

14. Уточнение структур входных и выходных данных.
15. Разработка алгоритма решения задачи.
16. Разработка структуры программы.

Рабочий проект

17. Программирование и отладка программы.
18. Разработка программных документов.
19. Испытания программы.
20. Корректировка программы и программной документации по результатам испытаний.
21. Внедрение
22. Подготовка и передача программы и программной документации для сопровождения.
23. Передача программы и учебного модуля университету.

5.4.2. Расчет трудоемкости выполнения работ

По степени новизны разрабатываемое программное изделие относится к группе В — разработка программной продукции, имеющей аналоги. По степени сложности алгоритма функционирования программа относится к группе сложности 3 — реализующая алгоритмы стандартных методов решения задачи. Количество переменной информации — 3, нормативно-справочной информации — 2, баз данных — 0. В зависимости от вида представления исходной информации, способа ее контроля и структуры выходных документов относится к группе 1.2 и 2.2.

Оценим трудоемкость разработки программного изделия. Она определяется как сумма трудоемкости каждого этапа разработки:

$$t_{ПП} = t_{ТЗ} + t_{ЭП} + t_{ТП} + t_{РП} + t_{В}.$$

$t_{ТЗ}$ — трудоемкость разработки технического задания на создание ПП, $t_{ЭП}$ — трудоемкость разработки эскизного проекта, $t_{ТП}$ — трудоемкость разработки технического проекта, $t_{РП}$ — трудоемкость разработки рабочего проекта ПП, $t_{В}$ — трудоемкость внедрения разрабатываемого ПП.

Трудоемкость разработки технического задания рассчитывается по формуле

$$t_{ТЗ} = T_{РЗ}^3 + T_{РП}^3 = t_3 K_{РЗ}^3 + t_3 K_{РП}^3 = t_3 (K_{РЗ}^3 + K_{РП}^3).$$

t_3 — норма времени на разработку ТЗ на программный продукт в зависимости от функционального назначения и степени новизны разрабатываемого ПП, чел.дн. (47). $K_{РЗ}^3$ — коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком постановки задач на стадии ТЗ (в случае совместной с разработчиком ПО разработки ТЗ $K_{РЗ}^3 = 0.65$). $K_{РП}^3$ — коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии ТЗ (в случае совместной с разработчиком постановки задач $K_{РП}^3 = 0.35$).

$$t_{ТЗ} = t_3 (K_{РЗ}^3 + K_{РП}^3) = 47(0.65 + 0.35) = 47$$

Трудоемкость разработки эскизного проекта рассчитывается по формуле

$$t_{ЭП} = T_{РЗ}^Э + T_{РП}^Э = t_Э K_{РЗ}^Э + t_Э K_{РП}^Э = t_Э (K_{РЗ}^Э + K_{РП}^Э).$$

$t_Э$ — норма времени на разработку эскизного проекта программного продукта в зависимости от его функционального назначения и степени новизны, чел.дн. (67). $K_{РЗ}^Э$ — коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых

разработчиком программного обеспечения на стадии ЭП (в случае совместной с разработчиком ПО разработки ЭП $K_{P3}^{\partial}=0.7$). K_{PP}^{∂} — коэффициент, учитывающий удельный вес трудоемкости работ, выполняемых разработчиком программного обеспечения на стадии ЭП (в случае совместной с разработчиком постановки задач разработки ЭП $K_{PP}^{\partial}=0.3$).

$$t_{\partial\Pi} = t_{\partial} (K_{P3}^{\partial} + K_{PP}^{\partial}) = 67(0.7 + 0.3) = 67$$

Трудоемкость разработки технического проекта t_{TP} зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации и определяется как сумма времени, затраченного разработчиком постановки задач и разработчиком программного обеспечения, т.е.

$$t_{TP} = (t_{P3}^T + t_{PP}^T) K_B K_P = (t_{P3}^T + t_{PP}^T) K_B \frac{K_{\Pi} n_{\Pi} + K_{HC} n_{HC} + K_B n_B}{n_{\Pi} + n_{HC} + n_B}$$

t_{P3}^T, t_{PP}^T — норма времени, затрачиваемого на разработку технического проекта разработчиком постановки задач и разработчиком программного обеспечения соответственно, чел. дн. ($t_{P3}^T=14, t_{PP}^T=12$). K_B — коэффициент учета вида используемой информации; K_P — коэффициент учета режима обработки информации ($K_P=1.26$). K_{Π}, K_{HC}, K_B — значения коэффициентов учета вида используемой информации для переменной, нормативно-справочной информации и баз данных ($K_{\Pi}=1.00; K_{HC}=0.72; K_B=2.08$). n_{Π}, n_{HC}, n_B — количество наборов данных переменной, нормативно-справочной информации и баз данных.

$$t_{TP} = (t_{P3}^T + t_{PP}^T) K_P \frac{K_{\Pi} n_{\Pi} + K_{HC} n_{HC} + K_B n_B}{n_{\Pi} + n_{HC} + n_B} = (14 + 12) 1.26 \frac{1.00 * 3 + 0.72 * 2 + 2.08 * 0}{3 + 2 + 0} = 29.1$$

Трудоемкость разработки рабочего проекта t_{RP} зависит от функционального назначения ПП, количества разновидностей форм входной и выходной информации, сложности алгоритма функционирования, сложности контроля информации, степени использования готовых программных модулей, уровня алгоритмического языка программирования и определится по формуле

$$t_{RP} = K_K K_P K_{\Pi} K_{\Pi A} (t_{P3}^P + t_{PP}^P) = K_K K_P K_{\Pi} K_{\Pi A} (t_{P3}^P + t_{PP}^P) \frac{K_{\Pi}^{\cdot} n_{\Pi} + K_{HC}^{\cdot} n_{HC} + K_B^{\cdot} n_B}{n_{\Pi} + n_{HC} + n_B}.$$

K_K — коэффициент учета сложности контроля информации ($K_K=1.00$). $K_{\Pi A}$ — коэффициент учета уровня используемого алгоритмического языка программирования (

$K_{\text{я}} = 1.15$). $K_{\text{з}}$ — коэффициент учета степени использования готовых программных модулей ($K_{\text{з}} = 0.8$). $t_{\text{пз}}^{\text{п}}, t_{\text{пп}}^{\text{п}}$ — норма времени, затраченного на разработку рабочего проекта на алгоритмическом языке высокого уровня разработчиком программного обеспечения, чел. дн. ($t_{\text{пз}}^{\text{п}} = 0, t_{\text{пп}}^{\text{п}} = 100$). $K_{\text{п}}', K_{\text{нс}}', K_{\text{б}}'$ — коэффициенты учета сложности алгоритма ПП и вида используемой информации для переменной, нормативно-справочной информации и баз данных $K_{\text{п}}' = 1.00, K_{\text{нс}}' = 0.48, K_{\text{б}}' = 0.40$.

$$t_{\text{пп}} = K_{\text{к}} K_{\text{р}} K_{\text{я}} K_{\text{з}} (t_{\text{пз}}^{\text{п}} + t_{\text{пп}}^{\text{п}}) \frac{K_{\text{п}}' n_{\text{п}} + K_{\text{нс}}' n_{\text{нс}} + K_{\text{б}}' n_{\text{б}}}{n_{\text{п}} + n_{\text{нс}} + n_{\text{б}}} =$$

$$= 1,00 * 1,26 * 1,15 * 0,8 * (100 + 0) \frac{1,00 * 3 + 0,48 * 2 + 0,4 * 0}{3 + 2 + 0} = 91,8$$

Трудоемкость выполнения стадии внедрения $t_{\text{в}}$ может быть рассчитана по формуле

$$t_{\text{в}} = (t_{\text{пз}}^{\text{б}} + t_{\text{пп}}^{\text{б}}) K_{\text{к}} K_{\text{р}} K_{\text{з}}$$

$t_{\text{пз}}^{\text{б}}, t_{\text{пп}}^{\text{б}}$ — норма времени, затрачиваемого разработчиком постановки задач и разработчиком программного обеспечения на выполнение процедур внедрения программного продукта, чел. дн. $t_{\text{пз}}^{\text{б}} = 8, t_{\text{пп}}^{\text{б}} = 24$.

$$t_{\text{в}} = (t_{\text{пз}}^{\text{б}} + t_{\text{пп}}^{\text{б}}) K_{\text{к}} K_{\text{р}} K_{\text{з}} = (8 + 24) * 1,00 * 1,26 * 0,8 = 32,26$$

Таблица 9. Трудоемкость выполнения работ

Этапы разработки	Трудоемкость, чел. дн.
Техническое задание	47
Эскизный проект	67
Технический проект	30
Рабочий проект	92
Внедрение	33
Итого	269

5.5. Оценка стоимости изделия

Цена программного изделия ($\Pi_{\text{п}}$) определяется по формуле

$$\Pi_{\text{п}} = kC + \Pi_{\text{р}}$$

C — затраты на разработку программной продукции (сметная себестоимость). k — коэффициент учета затрат на изготовление опытного образца ПП как продукции

производственно-технического назначения ($k = 1, 1, \dots, 1, 2$). Π_P — нормативная прибыль, рассчитываемая по формуле

$$\Pi_P = \frac{(C - C_M)}{100} P_H$$

P_H — норматив рентабельности, 25%. C_M — материальные затраты, р/изд.

Материальные затраты, необходимые для разработки одного комплекта учебного модуля представлены в таблице. В статье указаны суммарные затраты на материалы, приобретенные для разработки базового и дополнительных модулей.

Таблица 10. Смета затрат на материалы

№ п/п	Наименование материала	Единица	Кол-во	Цена за единицу, р	Сумма, р	
Базовый модуль						
1	Плата печатная, изготовление	дм ²	1,8(1,5x1,2)	—	300,00	
2	Разъем угловой, на плату ВН10R	шт.	4	7,00	28,00	
3	Разъем прямой, на плату ВН10	шт.	5	7,00	35,00	
4	Переключатель DIP8, прямой	шт.	1	14,00	14,00	
5	Кнопка миниатюрная с фикс. 8мм В170G	шт.	1	10,00	10,00	
6	Кнопка миниатюрная без фикс. 8мм В170G	шт.	1	10,00	10,00	
7	Светодиод, 3мм, красный	шт.	1	4,00	4,00	
8	Светодиод, 3мм, синий	шт.	1	4,00	4,00	
9	Светодиод, 3мм, зеленый	шт.	1	4,00	4,00	
10	Однорядая штыревая линейка PLS3	шт.	1	0,20	0,20	
11	Конденсатор электролитический, 1 мкФ	шт.	1	1,00	1,00	
12	Кондженсатор электролитический, 220 мкФ	шт.	1	2,00	2,00	
13	Конденсатор керамический, 0.1 мкФ	шт.	5	0,20	1,00	

14	Конденсатор керамический, 180 пФ	шт.	2	0,10	0,20	
15	Резистор С1-04, 4.7К	шт.	9	0,10	0,90	
16	Резистор ЧИП 1206, 300	шт.	4	0,09	0,36	
17	Резистор ЧИП 1206, 10К	шт.	1	0,09	0,09	
18	Резистор ЧИП 1206, 1.1К	шт.	1	0,09	0,09	
19	Кварц, 12МГц	шт.	1	5,00	5,00	
20	Кварц, 16МГц	шт.	1	5,00	5,00	
21	Кварц, 11.0592МГц	шт.	1	6,00	6,00	
22	Колодка для микросхемы DIP40	шт.	1	4,00	4,00	
23	Колодка для микросхемы DIP10	шт.	1	4,00	4,00	
24	Колодка для микросхемы DIP28	шт.	1	4,00	4,00	
25	Колодка для микросхемы DIP32	шт.	1	4,00	4,00	
26	Колодка для микросхемы DIP16	шт.	1	4,00	4,00	
27	Выключатель питания, KSD-117-S	шт.	1	15,00	15,00	ChipDip
28	Гнездо пит.на панель DS-210B	шт.	1	5,00	5,00	
29	Микроконтроллер AT89S52-24PU	шт.	1	65,00	65,00	
30	Микроконтроллер Atmega8515-16PI	шт.	1	90,00	90,00	
31	Регистр-защелка SN74HC573AN	шт.	1	11,00	11,00	
32	Флеш-память SST29EE010	шт.	1	90,00	90,00	ChipDip
33	Статическая память произвольного доступа HM6264LP	шт.	1	140,00	140,00	ChipDip
34	Мультиплексор K555КП11	шт.	1	5,00	5,00	
35	Джампер	шт.	1	1,00	1,00	
36	Сборка	пайка	330	0,15	49,50	
	Итого	922.34 руб				
Конвертер сигналов TTL – RS-232						
1	Плата печатная	дм²	0,315(0,45x0,70)	60,00	19,00	
2	Разъем на плату, папа DB9	шт.	1	19,00	19,00	
3	Разъем угловой, на плату VH10R	шт.	2	7,00	14,00	

4	Колодка для микросхемы DIP16	шт.	1	4,00	4,00	
5	Конденсатор керамический, 0.1 мкФ	шт.	5	0,20	1,00	
6	Преобразователь уровней ADM202	шт.	1	40,00	40,00	
7	Сборка	пайка	57	0,15	8,55	
	Итого			105,55 руб		
Дискретные датчики						
1	Плата печатная	дм ²	0,18(0,45x0,40)	60,00	10,8	
2	Разъем угловой, на плату ВН10R	шт.	1	7,00	7,00	
3	Переключатель DIP8, прямой	шт.	1	14,00	14,00	
4	Резистор С1-04, 4.7K	шт.	8	0,10	0,80	
5	Сборка	пайка	46	0,15	6,90	
	Итого			39,50 руб		
Часы реального времени на DS1306						
1	Плата печатная	дм ²	0,25(0,55x0,45)	60,00	15,00	
2	Разъем угловой, на плату ВН10R	шт.	1	7,00	7,00	
3	Батарейный отсек CR2032	шт.	1	15,00	15,00	
4	Колодка для микросхемы DIP16	шт.	1	4,00	4,00	
5	Микросхема часов реального времени DS1306	шт.	1	99,00	99,00	
6	Переключатель DIP2, прямой	шт.	1	10,00	10,00	
7	Резистор С1-04, 4.7K		1	0,10	0,10	
8	Резистор С1-04, 300	шт.	1	0,10	0,10	
9	Светодиод, 3мм, зеленый	шт.	1	4,00	4,00	
10	Кварц, часовой 32768 Гц	шт.	1	11,00	11,00	
11	Конденсатор керамический, 6.8 пФ	шт.	2	0,09	0,18	
12	Литиевая батарейка	шт.	1	9,60	9,60	
13	Сборка	пайка	44	0,15	6,60	
	Итого			181,58 руб		
Перенаправитель портов						
1	Плата печатная	дм ²	0,42(0,70x0,60)	60,00	25,20	
2	Разъем угловой, на плату ВН10R	шт.	2	7,00	14,00	
3	Линейка	шт.	1,6	10,00	16,00	

	двухрядная, PLD80					
4	Сборка	пайка	148	0,15	22,20	
	Итого	77,40 руб				
Линейка светодиодов						
1	Плата печатная	дм ²	0,24(0,40x0,60)	60,00	14,40	
2	Разъем угловой, на плату ВН10R	шт.	1	7,00	7,00	
3	Резистор С1-04, 300	шт.	1	0,10	0,10	
4	Светодиод, 3мм, зеленый	шт.	8	4,00	32,00	
5	Сборка	пайка	44	0,15	6,60	
	Итого	60,10 руб				
Выход на клеммники						
1	Плата печатная	дм ²	0,33(0,55x0,60)	60,00	20,00	
2	Разъем угловой, на плату ВН10R	шт.	1	7,00	7,00	
3	Клеммник DG142V-03P	шт.	2	4,65	9,30	
4	Клеммник DG142V-02	шт.	2	7,00	14,00	
5	Сборка	пайка	40	0,15	6,00	
	Итого	56,30 руб				
Семисегментный индикатор						
1	Плата печатная	дм ²	0,49(0,70x0,70)	60,00	29,40	
2	Разъем угловой, на плату ВН10R	шт.	3	7,00	21,00	
3	Цифровой сегмент. индикатор СА56-21GWA	шт.	1	60,00	60,00	
4	Резистор С1-04, 300	шт.	8	0,10	0,80	
5	Резистор С1-04, 4,7К	шт.	4	0,10	0,40	
6	Транзистор КТ3102	шт.	4	1,60	6,40	
7	Сборка	пайка	80	0,15	12,00	
	Итого	130,00 руб				
Клавиатура						
1	Плата печатная	дм ²	0,74(0,70x1,05)	60,00	44,40	
2	Разъем угловой, на плату ВН10R	шт.	1	7,00	7,00	
3	Резистор С1-04, 2,7К	шт.	8	0,10	0,80	
4	Однорядая штыревая линейка PLS8	шт.	1	1,00	1,00	
5	Клавиатура АК-1604-N-BBW	шт.	1	123,00	123,00	
6	Сборка	пайка	26	0,15	3,90	
	Итого	180,10 руб				
ЖК индикатор МТ-10Т7						

1	Плата печатная	дм ²	0,43(0,50x0,85)	60,00	25,80	
2	Разъем угловой, на плату ВН10R	шт.	1	7,00	7,00	
3	Резистор переменный СА9V	шт.	1	9,00	9,00	
4	Транзистор КТ3102	шт.	1	1,60	1,60	
5	Резистор С1-04, 4,7K	шт.	2	0,10	0,20	
6	LCD МТ-10Т7	шт.	1	114,00	114,00	
7	Сборка	пайка	32	0,15	4,80	
	Итого			162,40 руб		
ЖК индикатор WH1604						
1	Плата печатная	дм ²	1,06(1,25x0,85)	60,00	63,60	
2	Разъем угловой, на плату ВН10R	шт.	2	7,00	14,00	
3	LCD WH1604A	шт.	1	340,00	340,00	
4	Транзистор КТ3102	шт.	1	1,60	1,60	
5	Резистор С1-04, 4,7K	шт.	2	0,10	0,20	
6	Резистор переменный СА9V	шт.	1	9,00	9,00	
7	Сборка	пайка	56	0,15	8,40	
	Итого			436,80 руб		
Прочее						
1	Коробка	шт.	1	30,00	30,00	
2	Нуль-модемный кабель	шт.	1	40,00	40,00	
3	Блок питания	шт.	1	670,00	670,00	
4	IDC-10F на шлейф	шт.	10	7,00	70,00	
5	Шлейф	м	1	1,00	1,00	
6	DS18B20 термометр	шт.	1	75,00	75,00	
	Итого			886,00 руб.		

Таблица 11. Себестоимость комплекта учебного модуля

№ п/п	Наименование модуля	Себестоимость, руб.
1	Базовый модуль	922,34
2	Конвертер сигналов RS232 – TTL	105,55
3	Дискретные датчики	39,50
4	Часы реального времени на DS1306	181,58
5	Перенаправитель портов	77,40
6	Линейка светодиодов	60,10
7	Выход на клеммники	56,30
8	Семисегментный индикатор	130,00
9	Клавиатура	180,10
10	ЖК индикатор МТ-10Т7	162,40
11	ЖК индикатор WH1604	436,80
12	Прочее	886,00
	Итого	3238,07

Сборка одного комплекта учебного модуля со всеми комплектующими составляет 3238,07 рублей. По сравнению с ценой аналогичной продукции такая цена оптимальна. Надо учесть то, что в цену закладывали розничные цены комплектующих.

Действительный (расчетный) годовой фонд времени — это номинальный фонд времени за вычетом неизбежных потерь. Действительный (расчетный) годовой фонд времени работы оборудования при 41-часовой рабочей недели и восьми праздничных днях в году составляет 2070 ч.

К специальному оборудованию внесен универсальный программатор для прошивания программы в микросхему. Его использовали в рабочем проекте. Таким образом, время использования составило 736 часов (92 дня по 8 часов). Макетная плата использовалась для проверки схемотехнических решений, заложенных в базовый модуль и применялась в эскизном и техническом проекте. Время использования составило 776 часов.

Таблица 12. Смета затрат на специальное оборудование

№ п/п	Наименование оборудования	Балансовая цена	Норма годовых амортизац. отчислений	Действительный годовой фонд времени	Время использования оборудования	
1	Универсальный программатор	14844,00	12,0	2070	736	
2	Макетная плата WB-108	1130,00	20,0	2070	776	

$$C_{об} = \frac{\sum_i C_{би} \alpha_i}{100 F_d} t_i,$$

где $C_{би}$ — балансовая цена i -го вида оборудования, в рублях. α_i — норма годовых амортизационных отчислений для оборудования i -го вида, в процентах. F_d — действительный годовой фонд времени, в часах. t_i — время использования i -го вида оборудования при выполнении данной разработки, в часах.

$$C_{об} = \frac{14844 * 12 * 736 + 1130 * 20 * 776}{2070 * 100} = 718,00$$

Затраты на специальное оборудование составили 718,00 рублей.

Затраты, связанные с использованием вычислительной техники, определяются по формуле

$$C_{эвм} = t_{эвм} K_{и} C_{эвм} K_{э}.$$

$t_{ЭВМ}$ — время использования ЭВМ для разработки программы. $Ц_{ЭВМ}$ — цена 1-го часа работы ЭВМ, в рублях (5 руб/час). $K_{Э}$ — коэффициент учета быстродействия ЭВМ ($K_{Э} = 1,0$) $K_{И}$ — поправочный коэффициент учета времени использования ЭВМ ($K_{И} = 1,15$).

$$C_{ЭВМ} = 269 * 8 * 1,15 * 5 = 12374,00$$

Основная заработная плата

$$C_{з.о.} = \sum_i \frac{З_i}{\alpha_i} t_i$$

$З_i$ — среднемесячный оклад i-го исполнителя, в рублях (16000,00). t_i — трудоемкость работ, чел. дн. (определяется из календарного плана). α_i — среднее количество рабочих дней в месяце (22 дня).

$$C_{з.о.} = \frac{16000,00}{22} * 269 = 195637,00$$

На заработную оплату за весь проект идет 195637,00 рублей

Отчисления на социальное страхование

$$C_{СС} = \alpha_{СС} (C_{з.о.} + C_{з.д.})$$

$\alpha_{СС}$ — коэффициент отчислений на социальные нужды (26%).

$$C_{СС} = 0.26 * 195637 = 50865,62$$

Производственная командировка в Москву за необходимым материалом, которого невозможно было достать в Чебоксарах — 1500 рублей.

Контрагентские расходы:

1. Подготовка платы базового модуля (1500,00 рублей)
2. Подготовка платы дополнительных модулей (900,00 рублей)

Общая сумма 2400,00 рублей.

Таблица 13. Смета затрат на разработку программной продукции

№ п/п	Наименование статьи	Сметная себестоимость, р.	Удельный вес, %	Примечание
1	Материалы	3238,07	1,27	
2	Специальное оборудование	718,00	0,28	
3	Основная заработная плата	195637,00	76,91	
4	Дополнительная заработная плата	0,00	0,00	
5	Отчисление на социальные нужды	50865,62	20,00	

6	Контрагентские расходы	2400,00	0,94	
7	Производственная командировка	1500,00	0,59	
	Итого	254358,70		

$$P_p = \frac{(C - C_M)}{100} p_H = \frac{254358,70 - 3238,07}{100} * 25 = 62780,16$$

$$Ц_{\Pi} = kC + P_p = 1,1 * 254358,70 + 62780,16 = 342574,73$$

Цена программного продукта составило 342,575 тыс. рублей.

Дополнительные капитальные вложения K_{Δ} (р/оного потребителя программы), связанные с внедрением разработанного изделия, определяются как

$$K_{\Delta} = \frac{T_{M.B.} K_{ЭВМ}}{T_{\text{пол}}} + Ц_{\Pi}$$

$K_{ЭВМ}$ — капитальные вложения в ПЭВМ, для которой предназначена программа.

$T_{\text{пол}}$ — полезный годовой фонд работы этой ПЭВМ (за вычетом простоев в ремонте), ч/год. $T_{M.B.}$ — машинное время ПЭВМ, нужное данному потребителю для тех задач, которые он решает с помощью разработанной программы, машино-ч/год. Дополнительного программного обеспечения кроме как самой программы на ЭВМ не требуется. Таким образом, капитальные вложения в ЭВМ не нужны. Дополнительное капиталовложение состоит только из цены только программного изделия

$$K_{\Delta} = Ц_{\Pi} = 342575$$

Расходы, связанные с эксплуатацией программного изделия E (р/год на потребителя изделия), определяются как

$$E = T_{M.B.} Ц_{ЭВМ} + \frac{Ц_{\Pi}}{T_c}$$

$T_{M.B.}$ — продолжительность машинного времени ПЭВМ, используемой в течение года для решения задач с помощью данной программы, машино-ч/год/потребитель изделия. $Ц_{ЭВМ}$ — эксплуатационные расходы, приходящие на один час машинного времени этой ПЭВМ, (5 р/машино-ч). T_c — срок службы программы (5 лет).

$$E = 54 * 5 + \frac{342575,00}{5} = 68785$$

68785,00 руб/год на одного потребителя.

$$T_{OK} = \frac{\frac{T_{M.B.2} K_{ЭВМ}}{T_{ПОЛ}} + Ц_{П}}{(1 + \alpha_{д})(1 + \alpha_{CC}) \sum_i 3_i - \left(T_{M.B.} Ц_{ЭВМ} + \frac{Ц_{П}}{T_{C}} \right)} = \frac{342574}{1,26 * 12 * 16000 - (5 * 54 + 342574 / 5)} = 2,32$$

Годовой экономический эффект

$$\begin{aligned} \mathcal{E}_{Г} &= E_{Э} - e_{H} K_{д} = (1 + \alpha_{д})(1 + \alpha_{CC}) \sum_i 3_i - \left(T_{M.B.} Ц_{ЭВМ} + \frac{Ц_{П}}{T_{C}} \right) - e_{H} K_{д} = \\ &= 1,26 * 12 * 16000 - (5 * 54 + 342574 / 5) = 173135,2 \end{aligned}$$

Таблица 14. Сводные экономические показатели

Показатель	Размерность	Система программ
Затраты на разработку программы	р.	254 359
Капитальные вложения	р.	342 575
Эксплуатационные расходы	р./год	68 785
Годовой экономический эффект	р./год	173 135
Срок окупаемости	лет	2,32

6. Безопасность и экологичность проекта

6.1. Характеристика персонального компьютера как технического источника электромагнитного поля

Основными составляющими частями персонального компьютера (далее по тексту ПК) являются: системный блок (процессор) и разнообразные устройства ввода/вывода информации: клавиатура, дисковые накопители, принтер, сканер, и т. п. Каждый персональный компьютер включает средство визуального отображения информации — монитор. Как правило, в его основе - устройство на основе электронно-лучевой трубки. ПК часто оснащают сетевыми фильтрами, источниками бесперебойного питания и другим вспомогательным электрооборудованием. Все эти элементы при работе ПК формируют сложную электромагнитную обстановку на рабочем месте пользователя (таблица 15)[24].

Таблица 15. Элементы компьютера и соответствующие им диапазоны частот

<i>Источник</i>	<i>Диапазон частот, первая гармоника</i>
Монитор	50 Гц
Сетевой трансформатор блока питания	50 Гц
Статический преобразователь напряжения в импульсном блоке питания	20 – 100 кГц
Блок кадровой развертки и синхронизации	48 – 160 Гц
Блок строчной развертки и синхронизации	15 – 110 кГц
Ускоряющее анодное напряжение монитора (только для мониторов с электронно-лучевой трубкой)	0 Гц (электростатика)
Системный блок (процессор)	50 Гц – 1 ГГц
Устройства ввода/вывода информации	0 Гц, 50 Гц
Источники бесперебойного питания	50 Гц, 20 – 100 кГц

Электромагнитное поле, создаваемое персональным компьютером, имеет сложный спектральный состав в диапазоне частот от 0 Гц до 1000 МГц. Электромагнитное поле (далее по тексту ЭМП) имеет электрическую (Е) и магнитную (Н) составляющие, причем взаимосвязь их достаточно сложна, поэтому оценка Е и Н производится отдельно. Максимальные зафиксированные на рабочем месте значения ЭМП представлены в таблице 16[24].

Таблица 16. Максимально значения ЭМП на рабочем месте пользователя

Тип поля	Частота	Напряженность
Электрическое поле	100 кГц – 300 МГц	17,0 – 24,0 В/м
Электрическое поле	0,02 – 2 кГц	150,0 – 155,0 В/м
Электрическое поле	2 – 400 кГц	14,0 – 16,0 В/м
Магнитное поле	100 кГц – 300 МГц	—
Магнитное поле	0,02 – 2 кГц	550,0 – 600,0 мА/м
Магнитное поле	2 – 400 кГц	35,0 мА/м
Электростатическое поле	0 Гц	22,0 кВ/м

Работа современных вычислительных комплексов состоящих из скоростных ПК и программного обеспечения связана с пересылкой больших объемов информации на высоких скоростях между устройствами самого ПК, а так же во внешние устройства и получения информации от них.

Все это сопровождается электромагнитным излучением от каналов передачи информации, и устройств ими соединяемых. Их сумма и определяет ЭМП создаваемые ПК. Внутри ПК в создании ЭМП участвуют[24]:

1. Процессор, шина данных процессора и цепи питания.
2. Контроллеры и северный мост чипсета.
3. Модули память и шина данных.
4. Инверторы питания перечисленных выше устройств.
5. HDD и шины IDE (ATA) и SATA.
6. CD и шина IDE (ATA).
7. Видеокарта и шина AGP или E-PCI.

И внешние линии связи:

1. COM порт и внешние подключения по нему.
2. LTP порт и внешние подключения по нему.
3. USB и подобные.
4. VGA и другие виды предназначенные для подключения мониторов.
5. Беспроводные сетевые адаптеры IEEE 802 для локальных сетей.

Уровень ЭМП за пределами корпуса ПК определяется множеством факторов от конструкции системной платы до качества корпуса и внешних линий связи. Если говорить о воздействии ЭМП на узлы ПК, то эта мощность нагревает элементы схемы ПК, снижает помехоустойчивость его узлов.

6.2. Воздействие электромагнитного поля на пользователя персонального компьютера

При работе монитора на экране кинескопа накапливается электростатический заряд, создающий электростатическое поле (ЭСтП). В разных исследованиях, при разных условиях измерения значения ЭСтП колебались от 8 до 75 кВ/м. При этом люди, работающие с монитором, приобретают электростатический потенциал. Разброс электростатических потенциалов пользователей колеблется в диапазоне от -3 до +5 кВ. Когда ЭСтП субъективно ощущается, потенциал пользователя служит решающим фактором при возникновении неприятных субъективных ощущений. Заметный вклад в общее электростатическое поле вносят электризующиеся от трения поверхности клавиатуры и мыши. Эксперименты показывают, что даже после работы с клавиатурой, электростатическое поле быстро возрастает с 2 до 12 кВ/м. На отдельных рабочих местах в области рук регистрировались напряженности статических электрических полей более 20 кВ/м. По обобщенным данным, у работающих за монитором от 2 до 6 часов в сутки функциональные нарушения центральной нервной системы происходят в среднем в 4,6 раза чаще, чем в контрольных группах, болезни сердечно-сосудистой системы - в 2 раза чаще, болезни верхних дыхательных путей - в 1,9 раза чаще, болезни опорно-двигательного аппарата - в 3,1 раза чаще. С увеличением продолжительности работы на компьютере соотношения здоровых и больных среди пользователей резко возрастает[24].

Исследования функционального состояния пользователя компьютера, проведенные в 1996 году в Центром электромагнитной безопасности, показали, что даже при кратковременной работе (45 минут) в организме пользователя под влиянием электромагнитного излучения монитора происходят значительные изменения гормонального состояния и специфические изменения биотоков мозга. Особенно ярко и устойчиво эти эффекты проявляются у женщин. Замечено, что у групп лиц (в данном случае это составило 20%) отрицательная реакция функционального состояния организма не проявляется при работе с ПК менее 1 часа. Исходя из анализа полученных результатов сделан вывод о возможности формирования специальных критериев профессионального отбора для персонала, использующего компьютер в процессе работы[24].

Влияние аэроионного состава воздуха. Зонами, воспринимающими аэроионы в организме человека, являются дыхательные пути и кожа. Единого мнения относительно механизма воздействия аэроионов на состояние здоровья человека нет[24].

Влияние на зрение. К зрительному утомлению пользователя относят целый комплекс симптомов: появление "пелены" перед глазами, глаза устают, делаются

болезненными, появляются головные боли, нарушается сон, изменяется психофизическое состояние организма. Необходимо отметить, что жалобы на зрение могут быть связаны как с упомянутыми выше факторами, так с условиями освещения, состоянием зрения оператора и др[24].

Синдром длительной статической нагрузки (СДСН). У пользователей дисплеев развивается мышечная слабость, изменения формы позвоночника. В США признано, что СДСН - профессиональное заболевание 1990-1991 годов с самой высокой скоростью распространения. При вынужденной рабочей позе, при статической мышечной нагрузке мышц ног, плеч, шеи и рук длительно пребывают в состоянии сокращения. Поскольку мышцы не расслабляются, в них ухудшается кровоснабжение; нарушается обмен веществ, накапливаются биопродукты распада и, в частности, молочная кислота. У 29 женщин с синдромом длительной статической нагрузки бралась биопсия мышечной ткани, в которых было обнаружено резкое отклонение биохимических показателей от нормы[24].

Стресс. Пользователи дисплеев часто находятся в состоянии стресса. По данным Национального Института охраны труда и профилактики профзаболеваний США (1990 г.) пользователи в большей степени, чем другие профессиональные группы, включая авиадиспетчеров, подвержены развитию стрессорных состояний. При этом у большинства пользователей работа на сопровождается значительном умственным напряжением. Показано, что источниками стресса могут быть: вид деятельности, характерные особенности компьютера, используемое программное обеспечение, организация работы, социальные аспекты. Работа имеет специфические стрессорные факторы, такие как время задержки ответа (реакции) компьютера при выполнении команд человека, "обучаемость командам управления" (простота запоминания, похожесть, простота использования и т.н.), способ визуализации информации и т.д. Пребывание человека в состоянии стресса может привести к изменениям настроения человека, повышению агрессивности, депрессии, раздражительности. Зарегистрированы случаи психосоматических расстройств, нарушения функции желудочно-кишечного тракта, нарушение сна, изменение частоты пульса, менструального цикла. Пребывание человека в условиях длительно действующего стресс-фактора может привести к развитию сердечно-сосудистых заболеваний[24].

Жалобы пользователей персонального компьютера возможные причины их происхождения показаны в таблице 17[24].

Таблица 17. Жалобы пользователей и возможные причины происхождения

Субъективные жалобы пользователей	Возможные причины происхождения
Резь в глазах	Визуальные эргономические параметры монитора, освещение на рабочем месте и в помещении
Головная боль	Аэроионный состав воздуха в рабочей зоне, режим работы
Повышенная нервозность	Электромагнитное поле, цветовая гамма помещения, режим работы
Повышенная утомляемость	Электромагнитное поле, режим работы
Расстройство памяти	Электромагнитное поле, режим работы
Нарушение сна	Режим работы, электромагнитное поле
Выпадение волос	Электростатические поля, режим работы
Прыщи и покраснение кожи	Электростатические поле, аэроионный и пылевой состав воздуха в рабочей зоне
Боли в животе	Неправильная посадка, вызванная неправильным устройством рабочего места
Боль в пояснице	Неправильная посадка пользователя вызванная устройством рабочего места, режим работы
Боль в запястьях и пальцах	Неправильная конфигурация рабочего места, в том числе высота стола не соответствует росту и высоте кресла; неудобная клавиатура; режим работы

Биологическое действие электромагнитных полей

Экспериментальные данные как отечественных, так и зарубежных исследователей свидетельствуют о высокой биологической активности ЭМП во всех частотных диапазонах. При относительно высоких уровнях облучающего ЭМП современная теория признает тепловой механизм воздействия. При относительно низком уровне ЭМП (к примеру, для радиочастот выше 300 МГц это менее 1 мВт/см²) принято говорить о нетепловом или информационном характере воздействия на организм. Механизмы действия ЭМП в этом случае еще мало изучены. Многочисленные исследования в области биологического действия ЭМП позволяют определить наиболее чувствительные системы организма человека: нервная, иммунная, эндокринная и половая. Эти системы организма являются критическими. Реакции этих систем должны обязательно учитываться при оценке риска воздействия ЭМП на население[24].

Биологический эффект ЭМП в условиях длительного многолетнего воздействия накапливается, в результате возможно развитие отдаленных последствий, включая дегенеративные процессы центральной нервной системы, рак крови (лейкозы), опухоли мозга, гормональные заболевания. Особо опасны ЭМП могут быть для детей, беременных (эмбрион), людей с заболеваниями центральной нервной, гормональной, сердечно-сосудистой системы, аллергиков, людей с ослабленным иммунитетом[24].

Влияние на нервную систему.

Большое число исследований, выполненных в России, и сделанные монографические обобщения, дают основание отнести нервную систему к одной из наиболее чувствительных систем в организме человека к воздействию ЭМП. На уровне нервной клетки, структурных образований по передачи нервных импульсов (синапсе), на уровне изолированных нервных структур возникают существенные отклонения при воздействии ЭМП малой интенсивности. Изменяется высшая нервная деятельность, память у людей, имеющих контакт с ЭМП. Эти лица могут иметь склонность к развитию стрессорных реакций. Определенные структуры головного мозга имеют повышенную чувствительность к ЭМП. Изменения проницаемости гемато-энцефалического барьера может привести к неожиданным неблагоприятным эффектам. Особую высокую чувствительность к ЭМП проявляет нервная система эмбриона[24].

Влияние на иммунную систему.

В настоящее время накоплено достаточно данных, указывающих на отрицательное влияние ЭМП на иммунологическую реактивность организма. Результаты исследований ученых России дают основание считать, что при воздействии ЭМП нарушаются процессы иммуногенеза, чаще в сторону их угнетения. Установлено также, что у животных, облученных ЭМП, изменяется характер инфекционного процесса - течение инфекционного процесса отягощается. Возникновение аутоиммунитета связывают не столько с изменением антигенной структуры тканей, сколько с патологией иммунной системы, в результате чего она реагирует против нормальных тканевых антигенов. В соответствии с этой концепцией, основу всех аутоиммунных состояний составляет в первую очередь иммунодефицит по тимус-зависимой клеточной популяции лимфоцитов[24].

Влияние на эндокринную систему и нейрогуморальную реакцию.

При воздействии ЭМП ведущее место отводилось изменениям в гипофиз-надпочечниковой системе. Исследования показали, что при действии ЭМП, как правило, происходила стимуляция гипофизарно-адреналиновой системы, что сопровождалось увеличением содержания адреналина в крови, активацией процессов свертывания крови. Было признано, что одной из систем, рано и закономерно вовлекающей в ответную реакцию организма на воздействие различных факторов внешней среды, является система гипоталамус-гипофиз-кора надпочечников. Результаты исследований подтвердили это положение[24].

Влияние на половую функцию.

Нарушения половой функции обычно связаны с изменением ее регуляции со стороны нервной и нейроэндокринной систем. С этим связаны результаты работы по изучению состояния гонадотропной активности гипофиза при воздействии ЭМП. Многократное облучение ЭМП вызывает понижение активности гипофиза[24]

Любой фактор окружающей среды, воздействующий на женский организм во время беременности и оказывающий влияние на эмбриональное развитие, считается тератогенным. Многие ученые относят ЭМП к этой группе факторов[24].

Первостепенное значение в исследованиях тератогенеза имеет стадия беременности, во время которой воздействует ЭМП. Принято считать, что ЭМП могут, например, вызывать уродства, воздействуя в различные стадии беременности. Хотя периоды максимальной чувствительности к ЭМП имеются. Наиболее уязвимыми периодами являются обычно ранние стадии развития зародыша, соответствующие периодам имплантации и раннего органогенеза[24].

Другие медико-биологические эффекты.

Результаты клинических исследований показали, что длительный контакт с ЭМП в сверхвысокочастотном диапазоне может привести к развитию заболеваний, клиническую картину которого определяют, прежде всего, изменения функционального состояния нервной и сердечно-сосудистой систем. Было предложено выделить самостоятельное заболевание - радиоволновая болезнь. Это заболевание, по мнению авторов, может иметь три синдрома по мере усиления тяжести заболевания[24]:

1. астенический синдром;
2. астено-вегетативный синдром;
3. гипоталамический синдром.

Наиболее ранними клиническими проявлениями последствий воздействия ЭМП на человека являются функциональные нарушения со стороны нервной системы, проявляющиеся прежде всего в виде вегетативных дисфункций неврастенического и астенического синдрома. Лица, длительное время находившиеся в зоне ЭМП, предъявляют жалобы на слабость, раздражительность, быструю утомляемость, ослабление памяти, нарушение сна. Нередко к этим симптомам присоединяются расстройства вегетативных функций. Нарушения со стороны сердечно-сосудистой системы проявляются, как правило, нейроциркуляторной дистонией: лабильность пульса и артериального давления, склонность к гипотонии, боли в области сердца и др. Отмечаются также фазовые изменения состава периферической крови (лабильность показателей) с последующим развитием умеренной лейкопении, нейropении, эритроцитопении. Изменения костного мозга носят характер реактивного

компенсаторного напряжения регенерации. Обычно эти изменения возникают у лиц по роду своей работы постоянно находившихся под действием ЭМП с достаточно большой интенсивностью. Работающие с магнитными полями и ЭМП, а также население, живущее в зоне действия ЭМП жалуются на раздражительность, нетерпеливость. Через 1-3 года у некоторых появляется чувство внутренней напряженности, суетливость. Нарушаются внимание и память. Возникают жалобы на малую эффективность сна и на утомляемость. Учитывая важную роль коры больших полушарий и гипоталамуса в осуществлении психических функций человека, можно ожидать, что длительное повторное воздействие предельно допустимых ЭМП (особенно в дециметровом диапазоне волн) может повести к психическим расстройствам[24].

6.3. Проведение электромагнитной мониторинга на рабочем месте пользователя персонального компьютера

Инструментальный контроль электромагнитной обстановки на рабочих местах пользователей ПК производится:

- при вводе ПК в эксплуатацию и организации новых и реорганизации рабочих мест;
- после проведения организационно-технических мероприятий, направленных на нормализацию электромагнитной обстановки при аттестации рабочих мест по условиям труда;
- по заявкам предприятий и организаций

Инструментальный контроль уровней ЭМП должен осуществляться приборами с допускаемой основной относительной погрешностью измерений $\pm 20\%$. Для измерения уровней напряженности на экране ПК необходимо установить типичное для данного вида работы изображение (текст программы, графики и др.). При проведении измерений должна быть включена вся вычислительная техника, монитор и другое используемое для работы электрооборудование, размешенное в данном помещении. Измерения параметров электростатического поля проводить не ранее, чем через 20 минут после включения ПК.

Измеритель ИП-50 предназначен для измерения напряженности электрического и магнитного поля промышленной частоты при контроле за соблюдением ПДУ, касающихся гигиены труда и коммунально-бытовой гигиены, установленных в стандартах Российской Федерации.

Измеритель ИП-50 имеет следующие основные характеристики:

рабочая частота.....50 \pm 1 Гц.

диапазон измерения среднеквадратического значения напряженности магнитного поля.....0,01 - 100 А/м

первый поддиапазон.....0,01 - 1 А/м

второй поддиапазон.....1 - 100 А/м

диапазон измерения среднеквадратического значения напряженности электрического поля.....1 В/м - 100 кВ/м

первый поддиапазон с ПИП N1.....0,01 - 1 кВ/м

первый поддиапазон с ПИП N2.....0,001 - 0,1 кВ/м

второй поддиапазон с ПИП N1.....1 - 100 кВ/м

второй поддиапазон с ПИП N2.....0,1-10 кВ/м

погрешность измерения, не более..... $\pm 10\%$

Таблица 18. Измерения напряженностей э/м поля монитора ЖК экрана

<i>Расстояние, м</i>	<i>Напряженность магнитного поля, А/м</i>	<i>Напряженность электрического поля, В/м</i>
0,5	0,017	0,015
1,0	0,015	0,010
1,5	0,014	0,010

6.4. Аттестация рабочего места пользователя персонального компьютера по электромагнитной безопасности

В соответствие с таблицей об аттестации рабочего места по электромагнитной безопасности и предельно допустимых уровней воздействия на человека электромагнитных полей аттестуем рабочее место.

Таблица 19. Классы условий труда при действии ЭМП

<i>Показатель</i>	<i>Класс условий труда</i>						
	<i>оптимальный</i>	<i>допустимый</i>	<i>вредный</i>				<i>опасный</i>
	1	2	3.1	3.2	3.3	3.4	4
Электромагнитные поля на рабочем месте пользователя ПЭВМ	-	\leq ВДУ	>ВДУ	-	-	-	-

Таблица 20. Временные допустимые уровни воздействия на человека ЭМП

<i>Виды поля</i>	<i>ВДУ</i>		
	<i>СанПиН</i>	<i>MPR II</i>	<i>ТСО-03</i>
Электрическое	25 В/м	25 В/м	10 В/м
Магнитное	0,200 А/м	0,160 А/м	0,160 А/м

Анализ результатов измерения ЭМП на рабочем месте пользователя ПК (таблица 18), таблицы классов условий труда при действии электромагнитных полей (таблица 19) [12] и таблицы ВДУ наиболее известных стандартов (таблица 20)[24], сделан следующий вывод. Рабочее место пользователя персонального компьютера относится к допустимому классу условий труда по электромагнитной обстановке.

Допустимые условия труда (2 класс) характеризуются такими уровнями факторов среды и трудового процесса, которые не превышают установленных гигиенических нормативов для рабочих мест, а возможные изменения функционального состояния организма восстанавливаются во время регламентированного отдыха или к началу следующей смены и не оказывают неблагоприятного действия в ближайшем и отдаленном периоде на состояние здоровья работников и их потомство. Допустимые условия труда условно относят к безопасным условиям.

6.5. Расчет искусственного освещения производственного помещения с ПК

Метод коэффициента использования светового потока применяется при равномерном расположении светильников и при нормированной горизонтальной освещенности. С помощью этого метода рассчитывают среднюю освещенность поверхности. При этом наиболее целесообразно рассчитывать освещение для помещений со светлым потолком и стенами, особенно при рассеянном и отраженном свете[19].

Ширина помещения $A = 10$ м., длина $B = 8$ м., высота $H = 3$ м. Коэффициент отражения потолка — 70%, стен — 50%, пола — 30%, лампа ЛБ - 36. Необходимо рассчитать такое количество светильников в помещении.

Количество светильников рассчитывается по формуле:

$$N = \frac{E \cdot S \cdot z \cdot K_3}{\eta \cdot \Phi_{\text{л}}} \quad (6.1)$$

где E – минимальная нормируемая освещенность, лк. Для III класса зрительных работ $E = 200$ лк (таблица 1, [11])

S - площадь помещения, m^2 ;

z, K_z – соответственно коэффициенты неравномерности освещения и запаса. Для помещения общественных и жилых зданий с нормальными условиями среды для искусственного освещения коэффициент запаса равен 1,4 (таблица 3, [11]). Коэффициент неравномерности освещения (отношение средней освещенности к минимальной) ;

η – коэффициент использования светового потока, определяется по коэффициентам отражения потолка, стен и индекса помещения. Индекс помещения рассчитывается по формуле:

$$j = \frac{AB}{H(A+B)} \quad (6.2)$$

$$j = \frac{AB}{H(A+B)} = \frac{80}{3 \cdot (8+10)} = 1.4$$

Для освещения помещения рекомендуются газоразрядные лампы типа ЛБ. Поэтому для расчетов будут использоваться показатели для светильника «Циклон ДеЛюкс», который рассчитан на две газоразрядные лампы.

Таблица 21. Характеристики светильников

Светильник	«Глубоко излучатель»			«Универсаль»			«Люцетта»			«Молочный шар»			«Циклон ДеЛюкс» (ЭЛПО)			«Леванто» (ЛПО)			«Муссон» (ЛПО)		
Коэффициент отражения потолка ρ_n , %	30	50	70	30	50	70	30	50	70	30	50	70	70	50	30	70	50	30	70	50	30
Коэффициент отражения стен ρ_c , %	10	30	50	10	30	50	10	30	50	10	30	50	50	30	10	50	30	10	50	30	10
Индекс помещения i	Коэффициент использования η , %																				
0.6	24	27	31	27	30	34	19	25	33	19	22	26	30	24	20	22	17	15	41	34	31
0.8	32	34	37	35	38	41	25	31	41	28	28	32	37	29	26	21	22	19	48	41	38
1.25	39	41	43	44	46	48	31	38	48	31	35	37	47	38	35	28	26	58	50	47	

2	44	46	49	50	52	55	38	45	55	39	40	43	57	47	43	41	34	31	68	58	yy
3	49	51	53	55	57	60	44	51	60	43	45	47	64	51	50	45	37	36	73	61	60
5	52	54	57	58	60	63	48	56	65	46	48	51	70	58	56	49	41	39	78	66	64

При $\rho_n = 0,7$ $\rho_c = 0,5$ и $i = 1,4$ коэффициент использования светового потока $\eta = 0,47$ (таблица 6.6);

$\Phi_{\text{л}}$ - световой поток одной лампы, лм. Для лампы ЛБ-36 световой потока равен 2800 лм.

Так как светильник «Циклон ДеЛюкс» имеет две лампы, то $\Phi_{\text{л}} = 2800 \cdot 2 = 5600 \text{ лм}$

Таким образом, получаем количество светильников $N = \frac{200 \cdot 10 \cdot 8 \cdot 1,1 \cdot 1,4}{0,47 \cdot 5600} = 9,31 \approx 9$

Практическое значение светового потока, которое должно отличаться на -10 + 20%

вычисляется как $\Phi_{\text{пр}} = \frac{E S z K_z}{\eta N} = \frac{200 \cdot 10 \cdot 8 \cdot 1,1 \cdot 1,4}{0,47 \cdot 9} = 5825$

Отклонение теоретического значения светового потока от практического значения:

$$D = 100\% \frac{\Phi_{\text{пр}} - \Phi_{\text{л}}}{\Phi_{\text{пр}}} \quad (6.3)$$

$$D = 100\% \frac{\Phi_{\text{пр}} - \Phi_{\text{л}}}{\Phi_{\text{пр}}} = 100 \frac{5825 - 5600}{5825} = 3,8\%, \text{ что является допустимым отклонением.}$$

Суммарная мощность вычисляется по формуле:

$$P = P_{\text{л}} N n \quad (6.4)$$

где $P_{\text{л}}$ - мощность одной лампы, Вт; для ЛБ-36 мощность равна 36 Вт.

N – количество светильников

n – количество ламп в одном светильнике

$$P = 36 \cdot 9 \cdot 2 = 648 \text{ Вт} = 0,65 \text{ кВт}$$

Таким образом, в офисном помещении площадью 80 м² и высотой 3 м, должно быть установлено 9 светильников «Циклон ДеЛюкс» с лампами ЛБ-36 суммарной мощностью 0,65 кВт.

Заключение

В ходе проведенного анализа в качестве микроконтроллеров для учебного модуля выбраны AVR и MCS-51. На их основе спроектирован и собран учебный модуль. Было разработан программный комплекс, включающий в себя Programmer51, Embedded51, тестовая программа для AVR и 51. Все разработанные части отвечают предъявленным требованиям (не только ПО но и разработанный модуль). Небольшие недочеты, выявленные в ходе эксплуатации, были успешно исправлены. Все используемое сторонне ПО является бесплатным и позволяет применять его в учебном процессе без нарушения лицензионных соглашений.

Проект экологически безопасен и не требует специальной сертификации при его использовании в учебном процессе. Модуль по себестоимости намного дешевле предлагаемой рынком продукции, хотя несколько уступает ему в функциональных характеристиках.

Всего было изготовлено 5 модулей, 4 из них переданы кафедре вычислительной техники ЧГУ и успешно применяются на практических занятиях.

Литература

Книги

1. Агуров П. В. Последовательные интерфейсы ПК. Практика программирования. — СПб.: БХВ-Петербург, 2004. — 496 с.: ил.
2. Ан Пей, Сопряжение ПК с внешними устройствами: Пер. с англ. — М.: ДМК Пресс, 2001. — 320 с. ил.
3. Москатов Е. А. Электронная техника. — Таганрог, 2004. — 121с.
4. Пухальский Г. И., Новосельцева Т. Я. Проектирование дискретных устройств на интегральных микросхемах: Справочник. — М.: Радио и связь, 1990. — 340 с.: ил.
5. Угрюмов Е. П. Цифровая схемотехника. — СПб.: БХВ-Петербург, 2001. — 528 с.: ил.

Нормативные документы

6. ГОСТ 19.701—90 (ИСО 5807 — 85) ЕСПД. Схемы алгоритмов, программ, данных и систем: Условные обозначения и правила выполнения. М.: Изд-во стандартов, 1991. 26 с.
7. ГОСТ 19.102—77 ЕСПД. Стадии разработки.
8. ГОСТ 19.404—79 ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
9. РД 50-34.698—90. Автоматизированные системы. Требования к содержанию документов.
10. Санитарно-эпидемиологические правила и нормативы. Гигиенические требования к персональным ЭВМ и организации работы. СанПиН 2.2.2/2.4.1340-03.—М.: «Книга сервис», 2003.—16 с.
11. СНиП 23-05 – 95 Строительные нормы и правила Российской Федерации. Естественное и искусственное освещение. М.: Изд-во стандартов, 1995.—26 с.
12. Руководство Р 2.2.2006 – 05. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификации условий труда.
13. ГОСТ 6825 – 91 (МЭК 81-84) Лампы люминесцентные трубчатые для общего освещения. М.: Изд-во стандартов, 1991.—26 с.

Учебные пособия и методические указания

14. Программное обеспечение вычислительной техники и автоматизированных систем: Метод. указания к дипломному проектированию / Сост. С. Н. Стоменский, Б. М. Калмыков; Чуваш. ун-т. Чебоксары, 2005. 60 с.
15. Андреева А. А, Симаков А. Л. Системное программирование. Последовательный порт: Текст лекций; Чуваш. ун-т. Чебоксары, 2002. — 60 с.
16. Пухальский Г. И., Новосельцева Т. Я. Цифровые устройства: Учебное пособие для втузов. — СПб.: Политехника, 1996. — 885 с.: ил.
17. Лопаткин А. В. Проектирование печатных плат в системе P-CAD 2001. Учебное пособие для практических занятий. — Нижний Новгород, НГТУ, 2002. — 178 с.
18. Емельянов В. М. Защита населения и территории в чрезвычайных ситуациях: учеб. пособие для высшей школы / В. М. Емельянов, П. А. Некрасов; под ред. В. В. Тарасова. — 4-е изд., доп. и испр.—М.: Академический проект, 2007.—496 с.
19. Безопасность жизнедеятельности: Учебник для вузов / С. В. Белов, А. В. Ильницкая, А. Ф. Козьяков и др.; под общей ред. С. В. Белова. 3-е изд., испр. и доп.—М.: Высш. шк., 2008.—616 с.: ил.

Ресурсы Internet

20. Материалы сайта <http://www.chipdip.ru/>
21. Материалы сайта <http://www.compitech.ru/>
22. Материалы сайта <http://www.emikro.ru/>
23. Материалы сайта <http://www.atmel.ru/>
24. Материалы сайта Европейской Академии Естественных наук
www.eanw.psychology.ru

Техническая документация

25. 1 Mbit (128K x 8) Page-Write EEPROM SST29EE010 / SST29VE010 Data Sheet (Silicon Storage Technology, Inc).
26. 8 bit AVR Microcontroller with 8K Bytes In-System Programmable Flash ATmega8515 (Atmel).
27. Atmel 8051 Microcontrollers Hardware Manual (Atmel).
28. Hexadecimal Object File Format Specification (Intel).

Приложение

```
#include "myInc.inc"
```

```
BSEG AT 0020h
```

```
fReceive:          DBIT  1
fTransmit:         DBIT  1
s__get_command_code:DBIT  1
s__get_check_sum:  DBIT  1
s__get_low_addr:   DBIT  1
s__get_high_addr:  DBIT  1
s__get_page_data:  DBIT  1
```

```
DSEG AT 0030h
```

```
ReceiveByte:      DS      1
State:            DS      1
Checksum:         DS      1
DataCount:        DS      1
ErrorCode:        DS      1
CmdCode:          DS      1
AddrPage:         DS      2
OldCS:            DS      1
```

```
CSEG AT 0A00h
```

```
PN:              DB  \
                 0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5D, 0x5E, 0x5F, \
                 0x60, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, \
                 0x68, 0x69, 0x6A, 0x6B, 0x6C, 0x6D, 0x6E, 0x6F, \
                 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, 0x77, \
                 0x78, 0x79, 0x7A, 0x7B, 0x7C, 0x7D, 0x7E, 0x7F
PN_001:          DB  \
                 0x84, 0x85, 0x86, \
                 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97, \
                 0x9A, 0x9B, 0x9C, 0x9D, 0x9E, 0x9F, \
                 0xA1, 0xA2, 0xA3, 0xA4, 0xA5, 0xA6, 0xA7, \
                 0xA9, 0xAA, 0xAB, 0xAC, 0xAD, 0xAE, 0xAF, \
                 0xB1, 0xB2, 0xB3, 0xB4, 0xB5, 0xB6, 0xB7, \
                 0xB9, 0xBA, 0xBB, 0xBC, 0xBD, 0xBE, 0xBF, \
                 0xC0, 0xC1, 0xC2, 0xC3, 0xC4, 0xC5, 0xC6, 0xC7
PN_002:          DB  \
                 0xC9, \
                 0xD1, 0xD2, 0xD3, 0xD4, 0xD5, 0xD6, 0xD7, \
                 0xD8, 0xD9, 0xDA, 0xDB, 0xDC, 0xDD, 0xDE, 0xDF, \
                 0xE1, 0xE2, 0xE3, 0xE4, 0xE5, 0xE6, 0xE7, \
                 0xE8, 0xE9, 0xEA, 0xEB, 0xEC, 0xED, 0xEE, 0xEF, \
                 0xF1, 0xF2, 0xF3, 0xF4, 0xF5, 0xF6, 0xF7, 0xF8, \
                 0xF9, 0xFA, 0xFB, 0xFC, 0xFD, 0xFE, 0xFF
```

```
CSEG AT 0000h
```

```
org      0
ljmp     INIT
org      23h
ljmp     INT_serial
```

```
INIT:
```

```
;=====
==
;               ИНИЦИАЛИЗАЦИЯ МИКРОКОНТРОЛЛЕРА
;=====
==
mov       SP, #038h
mov       P3, #0FFh
mov       P1, #0FFh
```

```

mov     SCON, #050h    ; /* uart in mode 1 (8 bit), REN=1 */
ANL     T2CON, #0F0h   ; /* EXEN2=0; TR2=0; C/T2#=0; CP/RL2#=0; */
ORL     T2CON, #30h    ; /* RCLK = 1; TCLK=1; */
MOV     TH2, #0FFh     ; /* init value */
MOV     TL2, #0D9h     ; /* init value */
MOV     RCAP2H, #0FFh   ; /* reload value, 9600 Bds at 12MHz */
MOV     RCAP2L, #0D9h   ; /* reload value, 9600 Bds at 12MHz */
SETB    ES             ; /* Enable serial interrupt */
SETB    EA             ; /* Enable global interrupt */
SETB    TR2            ; /* Timer 2 run */

setb    s__get_command_code
clr     fTransmit

; mov     R0, #38
;LLL:
; mov     A, R0
; call    TransmitByte
; inc     R0
; jmp     LLL
;=====
==
;                               ЦИКЛ ПРИЕМА КОМАНДНОГО ПАКЕТА ИЗ ПК
;=====
==
RECEIVE_CICLE:
    mov     CheckSum, #0

    ; Ожидание от ПК трех вставок-байт
begin_some:
    mov     R0, #3
transmit_some:
    jnb     fReceive, $
    clr     fReceive     ; Сброс признака принятого байта
    mov     A, SBUF
    mov     P1, A
    cjne    A, #ByteStaffing, begin_some
    djnz    R0, transmit_some

RECEIVE_NEXT_BYTE:
    jnb     fReceive, $
    clr     fReceive     ; Сброс признака принятого байта

    ; Вычисление по ходу контрольной суммы
    mov     A, ReceiveByte
    add     A, CheckSum
    mov     CheckSum, A

    ; Переход на обработку соответствующего состояния
    jbc     s__get_command_code, GET_COMMAND_CODE
    jbc     s__get_check_sum, GET_CHECK_SUM
    jbc     s__get_low_addr, GET_LOW_ADDR
    jbc     s__get_high_addr, GET_HIGH_ADDR
    jbc     s__get_page_data, GET_PAGE_DATA

    ; Приняли код команды пакета
GET_COMMAND_CODE:
    mov     A, ReceiveByte
    mov     CmdCode, A
    cjne    A, #cmd__flash_id, not_flash_id
    ljmp    get_flash_id
not_flash_id:
    cjne    A, #cmd__flash_erase, not_flash_erase
    ljmp    get_flash_erase

```

```

not_flash_erase:
    cjne A, #cmd__flash_write, not_flash_write
    ljmp get_flash_write
not_flash_write:
    cjne A, #cmd__flash_read, get_undefined_command
    ljmp get_flash_read
get_flash_id:
    setb s__get_check_sum
    ljmp RECEIVE_NEXT_BYTE
get_flash_erase:
    setb s__get_check_sum
    ljmp RECEIVE_NEXT_BYTE
get_flash_read:
    setb s__get_low_addr
    ljmp RECEIVE_NEXT_BYTE
get_flash_write:
    setb s__get_low_addr
    ljmp RECEIVE_NEXT_BYTE
get_undefined_command:
    ljmp ERROR_COMMAND_CODE

    ; Приняли младшую часть адреса
GET_LOW_ADDR:
    mov     AddrPage + 0, ReceiveByte
    setb s__get_high_addr
    ljmp RECEIVE_NEXT_BYTE

    ; Приняли старшую часть адреса
GET_HIGH_ADDR:
    mov     AddrPage + 1, ReceiveByte
    mov     A, CmdCode
    cjne A, #cmd__flash_write, cmd_is_read_flash
    setb s__get_page_data
    mov     DataCount, #0
    mov     DPTR, #PN
    ljmp RECEIVE_NEXT_BYTE
cmd_is_read_flash:
    setb s__get_check_sum
    ljmp RECEIVE_NEXT_BYTE

    ; Прием данных для записи
GET_PAGE_DATA:
    mov     A, DataCount
    movc    A, @A + DPTR
    mov     R0, A
    mov     @R0, ReceiveByte
    inc     DataCount
    mov     A, DataCount
    cjne A, #128, get_next_page_data
    setb s__get_check_sum
    ljmp RECEIVE_NEXT_BYTE
get_next_page_data:
    setb s__get_page_data
    ljmp RECEIVE_NEXT_BYTE

    ; Приняли контрольную сумму
GET_CHECK_SUM:
    mov     A, CheckSum
    jnz     ERROR_CHECK_SUM
    ljmp NOT_ERROR

ERROR_COMMAND_CODE:
    mov     A, #err_cmd_code
    mov     ErrorCode, A

```



```

        ljmp  TRANSMIT_RESPONSE
ERROR_CHECK_SUM:
        mov     A, #err_check_sum
        mov     ErrorCode, A
        mov     OldCS, CheckSum
        ljmp  TRANSMIT_RESPONSE
NOT_ERROR:
        mov     ErrorCode, #err_not_error
        ljmp  TRANSMIT_RESPONSE

;=====
==
;                               ВЫПОЛНЕНИЕ КОМАНДЫ И ПОСЫЛКА ОТВЕТА
;=====
==
TRANSMIT_RESPONSE:
        ; Посылка трех байт-вставок
        mov     R0, #3
        mov     CheckSum, #0
byteStaffingTransmit:
        mov     A, #ByteStaffing
        call  TransmitByte
        djnz  R0, byteStaffingTransmit

        ; В случае ошибки - посылка признака ошибки
        ; и кода ошибки
        mov     A, ErrorCode
        cjne  A, #0, error_pakage_transmit
        ljmp  command_run
error_pakage_transmit:
        mov     A, #Error
        call  TransmitByte
        mov     A, ErrorCode
        call  TransmitByte
        mov     A, OldCS
        call  TransmitByte
        ljmp  TRANSMIT_CHECKSUM ; На прием следующего пакета

command_run:
        ; Посылка кода принятой команды
        mov     A, CmdCode
        call  TransmitByte

        mov     A, CmdCode
        cjne  A, #cmd__flash_id, not_cmd__flash_id
        ljmp  FLASH_ID
not_cmd__flash_id:
        cjne  A, #cmd__flash_erase, not_cmd__flash_erase
        ljmp  FLASH_ERASE
not_cmd__flash_erase:
        cjne  A, #cmd__flash_read, not_cmd__flash_read
        ljmp  FLASH_READ
not_cmd__flash_read:
        ljmp  FLASH_WRITE

FLASH_ID:
        call  Software_ID_Entry ; SOFTWARE ID ENTRY
        call  DELAY_300ms
        mov     R7, #0
        mov     R6, #1
        call  ReadByte ; DEVICE ID READ (MEM[0001H])
        mov     A, R6
        call  TransmitByte

```

```

    mov     R7, #0
    mov     R6, #0
    call    ReadByte    ; MANUFACTURER's ID READ (MEM[0000H])
    mov     A, R6
    call    TransmitByte

    call    Software_ID_Exit
    ljmp    TRANSMIT_CHECKSUM

FLASH_ERASE:
    call    Software_Chip_Erase
    call    Delay_300ms
    ljmp    TRANSMIT_CHECKSUM
;=====
; Чтение из флеш-памяти
;=====
FLASH_READ:
    call    Software_Chip_Reset
    call    Delay_300ms
    mov     DataCount, #0
    mov     DPTR, #PN
read_cicle:
    mov     R7, AddrPage + 1
    mov     A, AddrPage + 0
    add     A, DataCount
    mov     R6, A
    call    ReadByte
    mov     A, DataCount
    movc    A, @A + DPTR
    mov     R0, A
    mov     A, R6
    mov     @R0, A
    inc     DataCount
    mov     A, DataCount
    cjne    A, #128, read_cicle

    ; Передача всего того, что прочли
    mov     DataCount, #0
transmit_page_data:
    mov     A, DataCount
    movc    A, @A + DPTR
    mov     R0, A

    mov     A, @R0
;    cpl     A
;    mov     P1, A
;    call    Delay_1s
    call    TransmitByte

    inc     DataCount
    mov     A, DataCount
    cjne    A, #128, transmit_page_data
    ljmp    TRANSMIT_CHECKSUM

FLASH_WRITE:
    call    Software_ID_Exit
    call    Software_DataProtect_Enable
    mov     DataCount, #0
    mov     DPTR, #PN
write_cicle:
    mov     R7, AddrPage + 1
    mov     A, AddrPage + 0
    add     A, DataCount

```

```

    mov     R6, A
    movc    A, @A + DPTR
    mov     R0, A
    mov     A, @R0
    mov     R5, A
    call    WriteByte
    inc     DataCount
    mov     A, DataCount
    cjne    A, #128, write_cicle

TRANSMIT_CHECKSUM:
    mov     A, #0
    clr     C
    subb    A, CheckSum
    call    TransmitByte
    ljmp    RECEIVE_CICLE

#include "proc.inc"
end

```