

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Хакасский государственный университет им. Н. Ф. Катанова»  
Колледж педагогического образования, информатики и права

## РЕФЕРАТ

на тему:

Изучение программатора микроконтроллера AVR на базе  
микроконтроллера AtM8

Автор реферата: \_\_\_\_\_ Е.С. Девяшина  
(подпись) (инициалы, фамилия)

Специальность: 09.02.01 Компьютерные системы и комплексы

Курс: III

Группа: Т-31

Зачет/незачет: \_\_\_\_\_

Руководитель: \_\_\_\_\_  
(подпись, дата) (инициалы, фамилия)

г. Абакан, 2015 г.

## СОДЕРЖАНИЕ

Введение.....	3
Программаторы- проблема выбора.....	4
Аппаратное устройства программаторов.....	5
Программное обеспечение программатора .....	10
Программатор для AVR.....	13
Схема USB программатора на Atmega8.....	14
Заключение.....	15
Библиографический список.....	16

## **ВВЕДЕНИЕ**

Актуальность исследования: Микроконтроллеры семейства AVR фирмы Atmel пользуются заслуженной популярностью как среди радиолюбителей, так и профессионалов: выгодное соотношение цены и качества влияет на выбор специалистов. Программатор микроконтроллеров AtM8 является наиболее простым в изготовлении и использовании, а также одним из самых бюджетных программаторов для Atmel. Все вышесказанное определяет актуальность выбранной темы.

Цель исследования: Разработка программатора микроконтроллеров AtM8 для микроконтроллеров AVR.

Задачи исследования:

1. Изучить процесс записи и считывания информации в постоянные запоминающие устройства.
2. Описать принцип работы и характеристики программатора микроконтроллеров AtM8 .
3. Спроектировать устройство.
4. Описать процесс разработки программатора микроконтроллеров AtM8.
5. Проанализировать качество разработанного устройства экспериментальным путем.

## **Программаторы — проблемы выбора**

Прочитав название статьи каких-нибудь 10-15 лет назад, мы бы очень удивились: нам бы ваши проблемы. Работая на «ящиках», мы и понятия не имели, что в этом вопросе бывает какая-то неопределенность. Перечень разрешенных к применению микросхем был настолько «широк», что приходилось выбирать практически из одной позиции, да и вопрос с программаторами обстоял элементарно просто: тебе нужен программатор — сделай его сам. Каждая лаборатория с гордостью делала свой, самый лучший и надежный программатор. Каких устройств там только не было: начиная от простых, но очень надежных изделий на тумблерах, до очень сложных, занимавших половину рабочего стола и иногда работавших мощных программаторов. Их разработчики с любовью программировали каждый бит тех сложных устройств (благо устройства эти не были обременены большим количеством информации). Обычно пользоваться таким программатором мог только человек, давший ему жизнь, а весь процесс программирования со стороны воспринимался, как шаманский танец с бубном. Теперь мы, к великому сожалению, лишены всей прелести тех лет.

С падением железного занавеса выяснилось, что перечень программируемых интегральных схем в мире просто гигантский — это микросхемы памяти как с параллельным, так и с последовательным доступом информации (EPROM, EEPROM, FLASH); микроконтроллеры с внутренней памятью команд и данных; микросхемы программируемой логики (PLD). Причем, перечень таких изделий с каждым годом стремительно растет, имея тенденцию к усложнению изделий и к увеличению их гибкости. С другой стороны, как ответ на потребность использования этих микросхем, рынок наполняется большим количеством программаторов. Как не ошибиться и сделать правильный выбор при приобретении программатора? В этой статье мы попытаемся дать представление об устройстве программаторов и ответить на этот вопрос.

## **Аппаратное устройства программаторов**

В первую очередь коснемся той детали программатора, с которой Вам придется взаимодействовать больше всего - это колодка, куда помещается программируемая микросхема. Эта одна из самых важных деталей программатора, от качества и надежности которой зависит способность программатора выполнять свои функции. Как показал наш многолетний опыт, любой программатор вне зависимости от его сложности, стоимости и функциональных возможностей обязательно должен быть снабжен специальной тестовой колодкой, обеспечивающей многократный надежный контакт с программируемой микросхемой. Фирмы, выпускающие такие сокетки, гарантируют надежный контакт при десятках тысяч операций установки в нее микросхем. Наиболее удобными для пользователя являются специальные сокетки с нулевым усилием (ZIF socket). Если программатор не снабжен специальными тестовыми сокетками, предназначенными для многократных установок микросхем, а вместо них стоят дешевые одноразовые колодки, то считайте, что Вы просто зря потратили свои деньги. Вы быстро сможете в этом убедиться, когда безвозвратно испортите микросхемы с однократным программированием из-за отсутствия контакта в колодке. В недорогих программаторах обычно устанавливаются универсальные (рассчитанные как на узкий, так и на широкий тип корпуса) ZIF DIP сокетки. В более дорогих образцах программаторов могут устанавливаться одновременно несколько видов ZIF сокетов, рассчитанных на разные типы корпусов микросхем (LCC, QFP и т.д.). Иногда программаторы снабжаются универсальными сменными головками под различные типы корпусов. Для программирования микросхем с корпусами, отличными от DIP и с большим числом выводов, программаторы снабжаются специальными адаптерами под соответствующий тип корпуса. В связи с тем, что на этих адаптерах также должны устанавливаться высоконадежные тестовые сокетки, стоимость таких адаптеров может оказаться довольно существенной.

Заглянем внутрь программаторов и в общих чертах попытаемся понять, чем же они отличаются друг от друга. Принципиально существует две концепции построения программаторов. Первая, и наиболее очевидная, заключается в построении программаторов на базе массива универсальных аппаратных драйверов. Универсальные драйверы подводятся к выводам тестовой сокетки и должны удовлетворять ряду специфических аппаратных требований по программированию микросхем. В перечень таких требований входят: способность подавать и считывать логические уровни, способность подавать сложные тактовые последовательности, способность подводить напряжение в диапазоне 0...27 В с точностью 0.1 В и т.д. и т.п. Удовлетворение всем этим требованиям приводит к колоссальным аппаратным затратам и избыточности всего устройства в целом. Количество драйверов универсального программатора должно соответствовать количеству выводов тестовой сокетки, например, 40 драйверов для сокетки DIP-40, или 84 драйвера для сокетки LCC-84. В результате, устройство становится очень сложным и дорогостоящим, но при этом абсолютно универсальным. Имея 40 универсальных драйверов и универсальную тестовую сокетку DIP-40 можно с уверенностью сказать, что удастся поддержать все существующие, а также любые новые, микросхемы в корпусе DIP (с числом выводов до 40) без дополнительных адаптеров. Именно по такой схеме строятся дорогие универсальные программаторы.

Вторая концепция заключается в том, что аппаратура программатора оптимизируется под предполагаемый перечень поддерживаемых микросхем. Программаторы этого класса, как правило, значительно дешевле универсальных программаторов, но такие изделия в известной степени теряют универсальные свойства. Добавление новых типов поддерживаемых программатором микросхем может быть сопряжено со значительными трудностями, а часто и с невозможностью расширения списка программируемых устройств.

Используя многолетний опыт разработки и производства программаторов, нами разработана и внедрена компромиссная концепция построения программаторов -

универсальный драйвер разбивается на два функциональных блока: универсальный логический драйвер и устройство коммутации "высокого" напряжения. Такая архитектура программатора позволила в значительной степени сохранить преимущества универсального драйвера и существенно сократить аппаратные затраты и, как следствие, уменьшить себестоимость и цену конечных изделий.

Рассмотрим способ подключения программаторов к компьютеру. Наиболее распространенными способами подключения являются:

- подключение к принтерному порту,
- подключение к последовательному порту,
- установка специальной платы в компьютер.

Каждый из этих способов имеет свои преимущества и недостатки. Использование специальных плат, устанавливаемых в компьютер, значительно упрощает схемотехнику программатора. В этом случае, как правило, удается отказаться от специального, довольно мощного источника питания, воспользовавшись источником питания компьютера, а также использовать центральный процессор компьютера в качестве управляющего процессора программатора. При способе подключения программатора к компьютеру посредством встраиваемых в компьютер плат удается достигнуть довольно значительных скоростей обмена между компьютером и программатором за счет непосредственного управления последним. Но такая реализация программатора имеет и существенные недостатки. Во-первых, значительно снижается мобильность программатора, т.е. возможность использования одного программатора на разных компьютерах (например, в пределах одной лаборатории), во-вторых, использование таких устройств с портативными компьютерами notebook сопряжено с необходимостью использования специальных карт сопряжения.

Другой вариант сопряжения программатора с компьютером - последовательный канал компьютера. Это вполне допустимый вариант сопряжения, допускающий работу программатора с компьютерами всех типов. К существенным недостаткам такого варианта сопряжения можно отнести невысокую пропускную способность канала. Максимальная скорость последовательного канала RS-232 ограничена значением 115 кБод, что существенно ограничивает обмен между компьютером и программатором, и, следовательно, снижает производительность последнего.

Подключение программатора к принтерному порту компьютера нам видится наиболее предпочтительным вариантом. Этот способ сочетает в себе достаточно высокую пропускную способность канала и не требует серьезных аппаратных затрат. При использовании этого способа удастся воспользоваться центральным процессором компьютера в качестве управляющего процессора программатора.

Способ обновления версии - это довольно существенный вопрос эксплуатации программаторов. Необходимость обновления версии может возникнуть по ряду причин, во-первых, при выявлении ошибки работы программатора (увы, такое тоже встречается, все мы грешны), либо при расширении списка поддерживаемых программатором микросхем. Способ обновления версии программатора зависит от его аппаратного устройства. В одних изделиях алгоритмы программирования жестко "зашиты" в аппаратуру, в других - они являются загружаемыми. В первом случае для модификации версии требуется модификация самого устройства программатора (например, перепрограммирование ПЗУ самого программатора), а это сопряжено с рядом дополнительных неудобств по доставке изделия производителю или в региональный сервисный центр. Другое дело, если обновление версии осуществляется только обновлением программного обеспечения программатора. Именно по такой схеме построены программаторы с загружаемыми алгоритмами программирования. В таких программаторах Вы обновляете только программное обеспечение и работаете уже с новой версией. Второй способ видится нам наиболее удобным в эксплуатации, и именно по такой



схеме строится обновление версий всех выпускаемых фирмой "Фитон" программаторов

Теперь обратим внимание на такую "мелочь", как способность программатора определять правильность установки микросхемы в тестовую колодку. При кажущейся незначительности этой опции, мы начинаем понимать всю ее важность только после выхода из строя микросхемы при неверной установке ее в колодку (а такое рано или поздно случается). И винить в этом случае некого - сами виноваты. Именно для предотвращения таких ситуаций и служит эта опция. Здесь необходимо указать, что полноценная реализация такой возможности требует от разработчика больших усилий и, порой, изобретательности. Дело в том, что необходимо протестировать микросхему в колодке в самом щадящем для нее режиме, при этом ни в коем случае не допуская выхода микросхемы из строя.

И в заключение обсуждения аппаратного устройства программаторов, на наш взгляд, необходимо упомянуть о настоятельной необходимости проведения программаторами процедуры самотестирования. Обычно эта процедура проводится после инициализации аппаратуры программатора. Цель этой процедуры - встроенными средствами провести проверку работоспособности всего оборудования устройства и принять решение о возможности полноценной работы программатора. К сожалению, встроенными средствами не всегда удастся однозначно убедиться в работоспособности всех узлов устройства, но тем не менее эта процедура обязательно должна проводиться с целью минимизации вероятности эксплуатации неработоспособного оборудования.

## **Программное обеспечение программатора**

Дадим общее представление о возможных способах реализации программного обеспечения (ПО) программатора. Первое, на что нужно обратить внимание, - это под управлением какой операционной системы работает программатор. Большинство программного обеспечения программаторов реализовано под управлением DOS и не предъявляет к компьютеру специфических требований. Обычно, это программы, разработанные довольно давно. В последние несколько лет стали появляться программаторы, работающие под управлением операционной системы Windows. Это современные продукты, которые, как правило, выглядят гораздо элегантнее, имеют более дружелюбный интерфейс и обладают большим числом сервисных возможностей.

Рассмотрим архитектуру программного обеспечения программаторов. Наиболее распространенной является архитектура, в которой в качестве ядра программатора выступает промежуточный буфер данных. Все операции в программаторе выполняются с этим буфером. Для программирования микросхемы необходимо загрузить файл в буфер, запрограммировать данные из буфера в микросхему, сравнить содержимое микросхемы и буфера. При чтении данные из микросхемы записываются в буфер. Размер промежуточного буфера данных, обычно, коррелирован с размером текущего типа микросхемы. Многолетний опыт разработки и производства программаторов позволил выработать концепцию оригинальной многобуферной архитектуры ПО программаторов с неограниченным количеством буферов. Такая архитектура позволяет работать с неограниченным количеством независимых наборов данных, проводить их анализ и редактирование. Например, Вы можете воспользоваться двумя буферами для считывания в них двух разных микросхем, проведения анализа этих наборов

данных и, на их базе, создания нового массива данных в третьем буфере для последующего программирования его в микросхемы памяти или сохранения на диске.

Стандартным набором функций программаторов обычно являются следующие функции: чтение, запись, сравнение, контроль на чистоту, стирание (только для электрически перепрограммируемых устройств) микросхем. Некоторые программаторы имеют функцию автоматического программирования. Эта функция позволяет осуществить часто используемую комбинацию действий для конкретного типа микросхемы. Обычно, такая комбинация состоит из такого набора: стереть микросхему, проконтролировать стертость, запрограммировать, сравнить запрограммированные данные с оригиналом, установить защиту. Удобство этой функции заключается в том, что весь "джентльменский" набор активизируется одним нажатием.

Отдельно остановимся на редакторских функциях программаторов. Наиболее распространенным перечнем редакторских функций являются: редактирование данных в шестнадцатеричном формате, возможность заполнения буфера данных константой и подсчет контрольной суммы. Этого простого набора редакторских функций вполне достаточно для простых приложений. Для профессионального использования программаторов необходимы расширенные возможности редактирования. К ним можно отнести возможность редактирования данных не только в шестнадцатеричном формате, но и в двоичном, восьмеричном и десятичном представлении; заполнение массива строкой данных; поиск и замена строки данных; инвертирование данных; копирование массива данных как внутри одного буфера, так и между разными буферами; подсчет контрольной суммы; конвертирование шин адреса и данных.

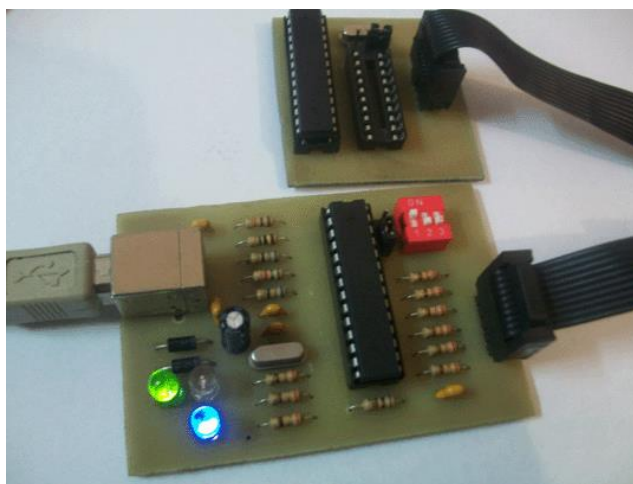
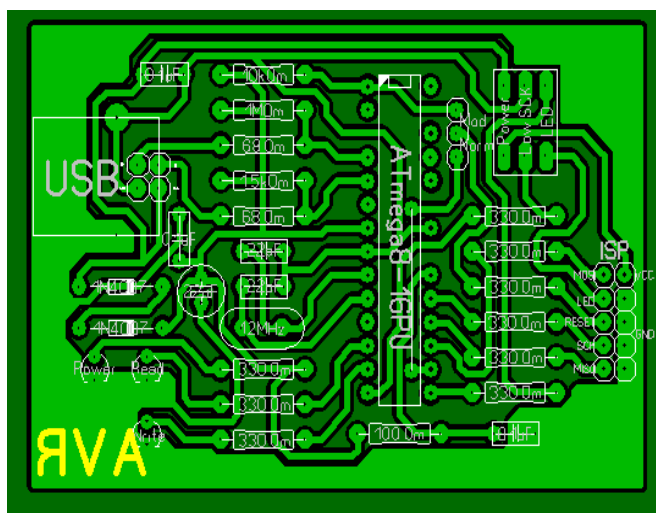
Еще одна особенность ПО программаторов, на которой стоит остановиться отдельно, - это пакетный режим работы. Очень в немногих программаторах такой режим реализован. А преимущество такого режима просто очевидно - это

автоматизация работы. Использую пакетный режим работы, можно создавать сценарии работы с программатором, автоматизируя всю рутинную работу. Наиболее интересны устройства, где пакетный режим работы практически не имеет ограничений, в нем доступны все ресурсы программатора. В пакетном режиме можно загружать файлы, запускать программирование, манипулировать параметрами программирования, окнами на экране, выводить графические данные и т.д., и т.п. В качестве иллюстрации использования пакетного режима работы программатора можно привести задачу программирования партии микросхем, в каждой из которых должен быть запрограммирован серийный номер. На специальном языке создается сценарий работы программатора, который заключается в следующем: оператор указывает начальное значение серийного номера партии микросхем и запускает процедуру программирования, программатор программирует микросхему с текущим серийным номером и вычисляет серийный номер следующей микросхемы, помещая его в соответствующий раздел памяти, далее процедура циклически повторяется. В приведенном примере пакетный режим работы значительно облегчает работу оператора и исключает свойственные оператору ошибки.

## Программатор для AVR

Решил собрать программатор для AVR. Что может быть проще самодельного программатора для AVR? Не угадаете, проще может быть только покупной программатор для AVR. Есть множество типов программаторов: параллельные, высоковольтные, битбэнг... Лично я попользовался разными программаторами и пришел к выводу, что просто программатор — это скучно, нужно делать маленький комплекс по восстановлению МК. Практически единственное, с чем я сталкивался, так это с неправильно выставленным источником тактирования. Я предпочитаю работать с микроконтроллерами фирмы Atmel. Если быть точным, то я работаю только с ATMEGA16, ATMEGA8, ATTINY2313 и ATTINY13. Скоро начну эксперименты с ATMEGA128. Остальное было баловство, либо не привлекло моего внимания. Среди всего многообразия я остановил свой взор на двух программаторах. На USBasp и AVR910. Но AVR910 был лидером по двум причинам. Во-первых, он совместим с программой идущей в комплекте с AVR Studio; во-вторых, он генерирует меандр, что позволило сэкономить на одной микросхемке.

## Схема USB программатора на Atmega8



Список необходимых деталей:

- Atmega8 — 1 шт.
- Кварц 12МГц — 1 шт.
- Диод 1N4007 – 2 шт.
- Светодиод — 3 шт.
- Резисторы: 68 Ом — 2 шт., 330 Ом — 8 шт., 1,5 Ом — 1 шт., 100 Ом -1 шт., 1,5 кОм -1 шт., 10 кОм -1 шт., 1 мОм -1 шт.
- Конденсаторы: 0,1мк — 3 шт., 22мк x 10В — 1 шт., 22р — 2 шт.

## **Заключение.**

Рассмотренная выше практическая задача дает четкое представление о значимости представленного устройства.

Изучена предметная область, описаны микроконтроллеры и их виды, AVR корпуса.

Рассмотрено назначение программаторов, их классификация и особенности.

На основе изученного материала были выбраны AVR микроконтроллеры, как основа устройства.

Рассмотрены различные принципиальные схемы программаторов, а также рассмотрены их плюсы и минусы, в результате чего была выбрана схема USBASP программатора AVR на микроконтроллере Atmega 8-р. Предоставлено описание программатора AVR USBASP и его характеристик.

## Список литературы.

1. Белов А.В. Микроконтроллеры AVR в радиолюбительской практике – СПб, Наука и техника, 2007 – 352с.
2. Проектирование цифровых устройств на однокристальных микроконтроллерах / В.В. Сташин [ и др.]. – М.: Энергоатомиздат, 1990. – 224 с.
3. Евстифеев А.В. Микроконтроллеры Microchip: практическое руководство/А.В.Евстифеев. – М.: Горячая линия – Телеком, 2002. – 296 с.
4. Кравченко А.В. 10 практических устройств на AVR-микроконтроллерах. Книга 1 – М., Додэка –ХХ1, МК-Пресс, 2008 – 224с.
- 5.Трамперт В. Измерение, управление и регулирование с помощью AVR-микроконтроллеров: Пер. с нем – К., МК-Пресс, 2006 – 208с.
6. Предко М. Руководство по микроконтроллерам. Том 1. / Пер. с англ. под ред.И. И. Шагурина и С.Б. Лужанского - М.: Постмаркет, 2001. - 416 с.
7. Предко М. Руководство по микроконтроллерам. Том 2. / Пер. с англ. под ред.И. И. Шагурина и С.Б. Лужанского - М.: Постмаркет, 2001. - 488 с.



8. Вуд А. Микропроцессоры в вопросах и ответах. / Пер. с англ. под ред. Д.А. Поспелова. - М.: Энергоатомиздат. 1985. - 184 с.
9. Уильямс Г.Б. Отладка микропроцессорных систем: / Пер. с. англ. - М.: Энергоатомиздат, 1988. - 253с.
10. Бродин Б.В., Шагурин И.И. Микроконтроллеры: Справочник. - М.: ЭКОМ, 1999. - 395 с.
11. Программируемые логические ИМС на КМОП-структурах и их применение. / П.П. Мальцев, Н.И. Гарбузов, А.П. Шарапов, А.А. Кнышев. - М.: Энергоатомиздат, 1998. - 158 с.
12. Соловьев В.В., Васильев А.Г. Программируемые логические интегральные схемы и их применение. - Мн.: Беларуская наука, 1998. - 270 с.
13. Лаптев В. Цифровой измеритель температуры на базе AVRмикроконтроллера и RC-цепочки. - Электронные от компоненты, 2001. №2, с.46 – 49
14. Зобин Ю., Тамазов А. Микроконтроллеры популярных семейств//Радио. - 2000. - №6-8.
15. Голубцов М.С., Кириченко А.В. Микроконтроллеры AVR: от простого к сложному. - 2-е изд. - М: Салон-Пресс, 2004