

Многофункциональный универсальный программатор - 2

Николай Хлюпин (RA4NAL)
г. Киров



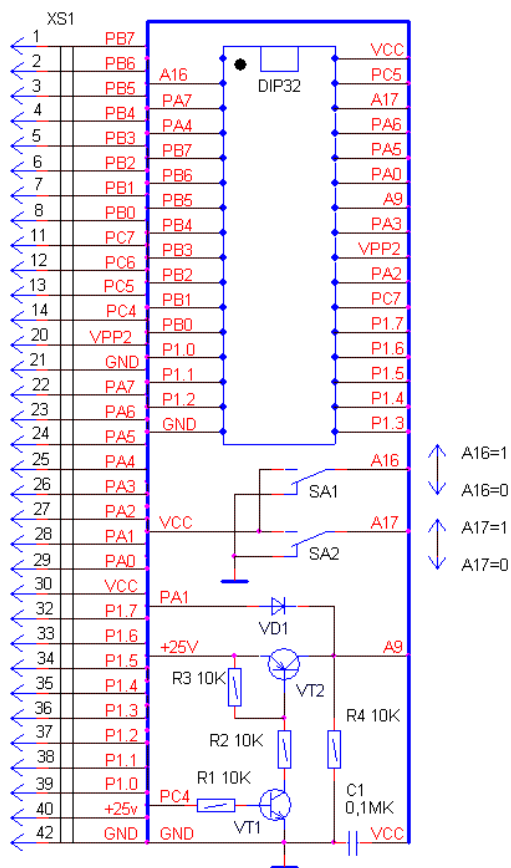
Прошло около 20 лет с момента разработки многофункционального программатора [1]. Контроллеров сейчас выпускается очень много, и я просто не в силах обеспечивать их поддержку. Да и во многих случаях для того, чтобы запрограммировать контроллер, например Arduino или STM32, никакой программатор вообще не нужен. Так что я думал, что тема закрыта.

Но неожиданно, для одной секретной миссии, мне потребовалось запрограммировать ПЗУ W29C020 объемом 256 Кбайт. Зачем? Расскажу, когда истечет срок давности. Но, поверьте, все абсолютно безобидно, никакого криминала, разумеется.

И тут я вспомнил об этом программаторе. Однако проблема в том, что в прошлом веке память измерялась не в гигабайтах и терабайтах, а в килобайтах. Наиболее распространенными были EPROM ПЗУ с УФ стиранием серии K573 объемом 2 и 8 Кбайт. Именно под них и начиналась разработка программатора. Вся его архитектура была рассчитана на программирование памяти объемом не более 64 Кбайт.

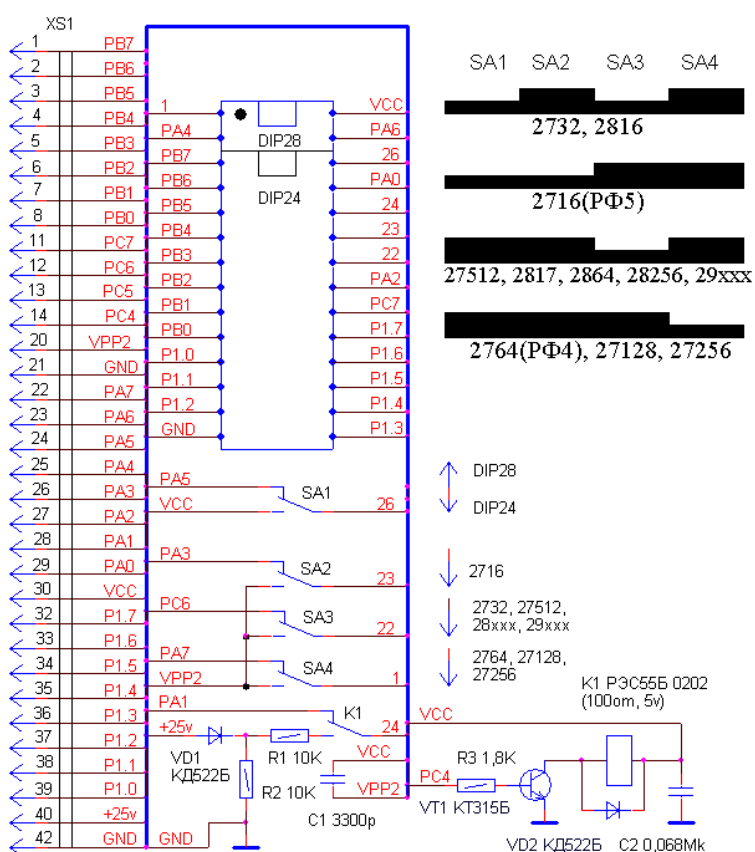
Но ведь можно программировать память страницами. Неудобно и долго, но для разовых задач вполне допустимо. Поэтому я разработал сменный модуль для программирования микросхем памяти EEPROM и FLASH в корпусах DIP32 объемом 128 и 256 Кбайт для этого программатора.

Программу, разумеется, тоже пришлось дорабатывать. В старом модуле для EPROM, EEPROM, FLASH я исключил возможность программирования микросхем памяти объемом 64 Кбайт в корпусах DIP32, так как они сейчас программируются в новом модуле. Показанные ниже варианты схем (см. рис. 1-3) сменного модуля для программирования ПЗУ до 64 Кбайт полностью аналогичны оригинальным



Сменный блок EEPROM, FLASH 128 и 256 Kbyte

Рис. 1



Сменный блок EPROM, EEPROM, FLASH

(С ручным выбором типа)

Рис. 2

за исключением того, что вместо DIP32 в них используется DIP28. Просто “обрезаны” за ненадобностью 4 вывода. Переделывать, конечно, ничего не надо, но при новом изготовлении это следует учесть.

Выбор страницы памяти в новом модуле (рис. 1) осуществляется вручную, переключателями SA1 и SA2. Можно добавить и SA3 на вывод 1 микросхемы для памяти 512 Кбайт, но такого экземпляра у меня нет. Узел на VT1, VT2, VD1, R1-R4 делать необязательно. Он используется только для чтения ID микросхемы. Эта информация нигде не используется, предназначена только для сведения. В этом режиме на вывод A9 подается +12 В. Можно просто исключить указанные выше элементы и соединить непосредственно линии PA1 и A9.

Работа с программатором не изменилась, она подробно описана в pdf файле, ссылка в конце статьи [2, 3]. Но при программировании микросхем EEPROM объемом

более 64 Кбайт файл прошивки следует предварительно “разрезать” на кусочки по 64 Кбайта. BIN файл можно “разрезать” с помощью, например, программы WinHex. А если прошивка в HEX, достаточно любого текстового редактора. Но тут могут быть сложности, поэтому лучше предварительно преобразовать HEX в BIN.

Алгоритм программирования будет такой. Загружаем первый файл, устанавливаем переключателями A16=0 и A17=0, программируем область памяти 0000-FFFF. Затем загружаем второй файл, устанавливаем A16=1, A17=0 и вновь программируем 0000-FFFF. Реально при этом данные запишутся по адресам 10000-1FFFF. Повторяем процедуру с третьим файлом, установив A16=0 и A17=1. И, наконец, записываем четвертый файл при A16=1, A17=1.

Чтение осуществляется аналогично. Четыре прочитанных файла нужно собрать в один с помощью WinHex. Тут также проще сохранить

файлы в BIN формате, объединить их, а затем, при необходимости, преобразовать в HEX.

Реально я протестировал программирование W29C020 и W29C011. Можно с высокой долей вероятности утверждать, что программатор прочитает любую память EPROM, EEPROM, FLASH серий 28xxxx и 29xxxx в корпусе DIP32 на 64, 128 и 256 Кбайт. Если установить 3-й переключатель, то и 512 Кбайт. К сожалению, про запись я так сказать не могу. Очень уж разная организация памяти и алгоритмы программирования у разных производителей. Для того, чтобы отладить работу с конкретным типом памяти, нужно его иметь в наличии. Я проверил программирование того, что у меня было.

Ну и раз уж дело дошло до этого программатора, я решил сделать еще одну модернизацию. Более 10 лет у меня ждал своей участи контроллер Atmel 89C51RD2. Это 8051 совместимый контроллер с FLASH памятью программ 64 Кбайта. Кто-то (не помню уже, кто) много лет назад подарил его мне. Как оказалось, это был Данайский Дар.

Фирма Atmel выпускала AT89C51RD2 и T89C51RD2. А вот просто 89C51RD2 с эмблемой Atmel в природе существовать не должен. Пришлось потратить немало времени, чтобы разобраться с этим. AT и T версии имеют существенные отличия. Главное в данном случае, это то, что встроенный UART bootloader, который позволяет записать прошивку в память без программатора, в T89C51RD2 зашит в последний Кбайт FLASH памяти. При стирании памяти он уничтожается. В AT89C51RD2 загрузчик размещается в отдельной области памяти и не может быть уничтожен.

Оказалось, что в моем экземпляре загрузчика нет вообще. Очевидно, это все-таки T89C51RD2, в котором был стерт загрузчик. Т.е. его программирование можно осуществить только аппаратным программатором. Многофункциональный

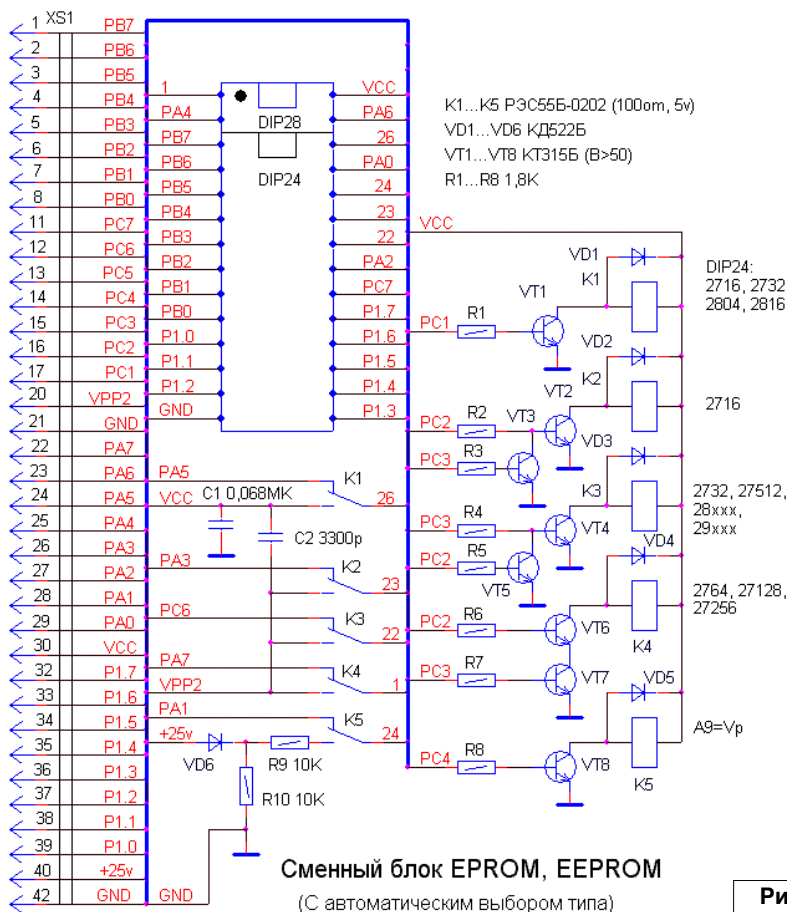


Рис. 3

программатор вполне справится с этой задачей, но нужно разбираться с алгоритмом и дорабатывать прошивку. А это не входило в мои планы.

К счастью, программатор, который у меня есть на работе, поддерживает этот тип контроллеров. Но где взять код фирменного загрузчика, чтобы восстановить его в памяти и не мучиться в дальнейшем? Во всех документах на контроллер указано, что bootloader свободно доступен на сайте Atmel. Но ни конкретного адреса, ни имени файла нигде не указывается. А найти нужную информацию после поглощения Atmel фирмой Microchip стало очень сложно. Мои поиски на сайте Microchip успехом не увенчались.

Пришлось воспользоваться “машиной времени” и переместиться в прошлое. С большим трудом я нашел код загрузчика на архивной копии сайта Atmel за 2010 год. После его прошивки в контроллер все заработало. Обновление ПО программатора стало возможным без использования дополнительных аппаратных средств.

Вся эта история в очередной раз подтвердила аксиому про бесплатный сыр. Лучше было приобрести AT89C51RD2 или AT89C51ED2 на Aliexpress и не заморачиваться. Обошлось бы это примерно в 3-4 USD.

Но вернемся к доработке программатора для его работы с контроллером, в котором есть FLASH память 64 Кбайта. Изменения в схеме минимальные. Потребуется вывод EA (pin 31) контроллера подключить к “земле” не непосредственно, а через резистор 1-3 кОм (рис. 4). И между этим выводом и

+5 В установить джампер. При разомкнутом состоянии джампера контроллер будет работать с программой, записанной во внешнее ПЗУ 27C256, а при замкнутом – с программой из внутренней FLASH памяти. В последнем случае ПЗУ 27C256 не нужно и его можно не устанавливать в панельку.

Кнопка между выводом PSEN (pin 29) и “землей” через резистор 1 кОм (см. рис. 4) нужна для запуска программы загрузчика при обновлении прошивки во FLASH памяти контроллера. При разомкнутом состоянии кнопки запускается программа пользователя – это обычная работа с программатором. Если включить питание программатора при нажатой кнопке, запускается bootloader. После старта загрузчика кнопку можно отпустить. Эту кнопку можно и не устанавливать, а для запуска загрузчика просто втыкать резистор 1 кОм в панельку для ПЗУ DD3 между выводами 14 и 20. Напрямую соединять вывод PSEN с землей НЕЛЬЗЯ! Может сгореть контроллер...

Дешифратор на элементах D6, D7 (схема в pdf файле с описанием программатора, ссылка в конце статьи) также можно упростить, так как сигнал PSEN не будет использоваться. FLASH память контроллера занимает все адресное пространство памяти программ, поэтому отладочный режим работы с программой, загруженной в ОЗУ, будет не доступен. Но все эти схемные упрощения имеет смысл делать только при новом изготовлении программатора. В уже собранной конструкции ничего менять не нужно, кроме установки джампера.

Для контроллера AT89C51RD2 в корпусе PLCC44 можно приобрести переходник PLCC44 - DIP40 (рис. 5)

на Aliexpress за символическую цену. Вместо AT89C51RD2 можно использовать AT89C51ED2 или T89C51RD2. В последнем случае нужно проявлять осторожность при обновлении прошивки – не следует делать стирание памяти, так как это уничтожит загрузчик. AT89C51ED2 отличается от AT89C51RD2 наличием дополнительной EEPROM памяти, которая в данном случае не используется.

Загрузка прошивки в контроллер осуществляется по интерфейсу RS-232 с помощью специальной программы FLIP (рис. 6). Процедура очень простая. Подключаем программатор с установленным незапрограммированным контроллером к реальному или виртуальному COM порту и включаем питание при нажатой кнопке “Upgrade”, запускаем FLIP и программируем FLASH.

Все интуитивно понятно, поэтому подробно останавливаться на процессе не буду. Если возникнут вопросы, есть встроенный HELP, а в конце статьи есть ссылки на pdf файл с описанием процедуры программирования и саму программу FLIP.

При использовании контроллера T89C51RD2 кнопку “Upgrade” на вывод PSEN можно не устанавливать. Загрузчик запускается командой “G FC00” из меню программатора. После этой команды закрываем терминал и запускаем FLIP. При использовании AT89C51RD2 или AT89C51ED2 такой номер не проходит, так как bootloader в них размещается в отдельной области памяти.

Новая прошивка программатора работает как в 27C256, так и в AT89C51RD2, AT89C51ED2, T89C51RD2. Ссылка на новую прошивку и ее исходный текст приведены в конце странички [2, 3].

В заключение несколько слов о том, как внести изменения в прошивку. На случай, если кто-то решит попрактиковаться в программировании на ассемблере для 8051. Программа написана в кодировке DOS. Ну что вы хотите – это

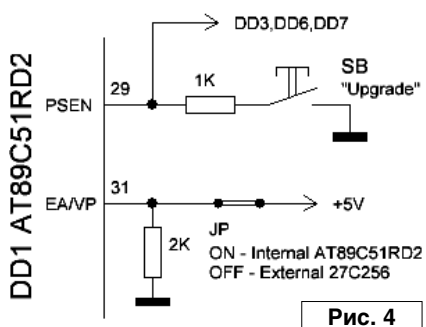


Рис. 4

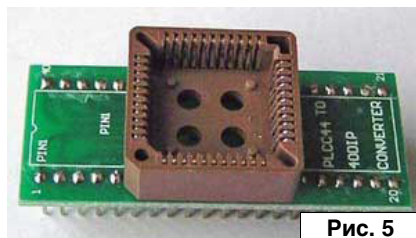


Рис. 5

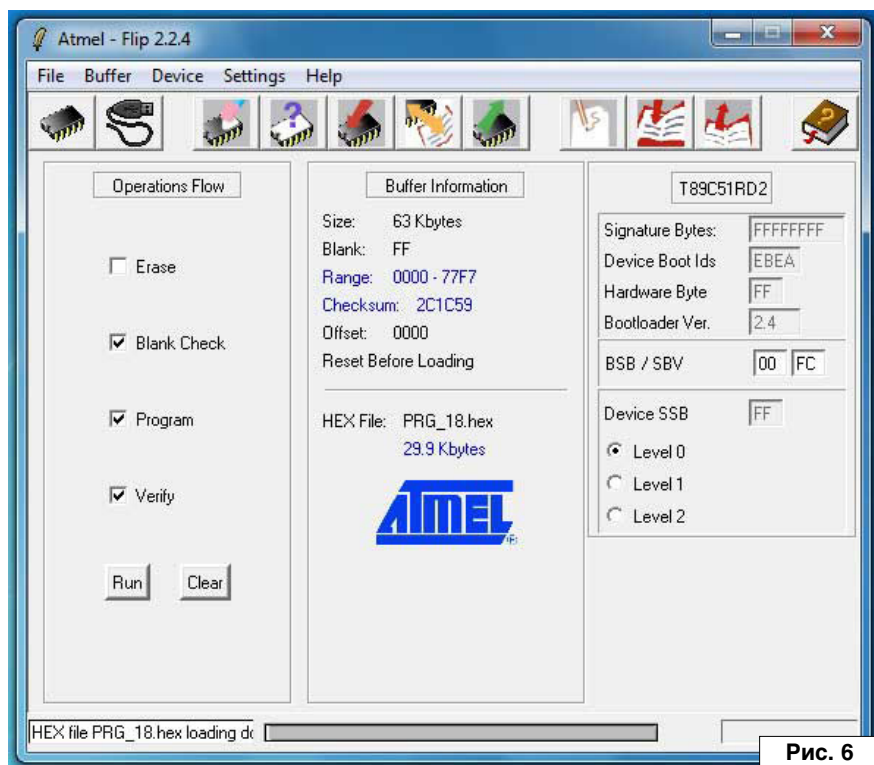


Рис. 6

же было еще в прошлом веке... Автоматически перекодировать ее в WIN нельзя – это полностью нарушит работу программы. Поэтому для просмотра и редактирования выбирайте в текстовом редакторе шрифт SYSTEM или TERMINAL.

Ассемблер asm51.exe, “довесок” для него, позволяющий работать с русскими буквами iasm51.exe и еще один ассемблер x8051.exe, нужные для компиляции, находятся в архиве с исходным текстом. Там же лежит и файл MOD51 с описанием регистров контроллера. Еще в архиве есть три командных файла – 1.bat, 2.bat и 3.bat. Это обычные текстовые файлы, которые можно посмотреть в любом редакторе.

Следует остановиться еще на одной проблеме. Упомянутые выше

ассемблеры – это DOS файлы, они не запускаются в 64-битных системах. Устанавливать ради этого виртуальную машину мне не захотелось. Компилировать в какой-то другой среде – может и не получиться из-за особенностей синтаксиса. К счастью, решение есть, это программа msdos.exe. Она позволяет запускать досовские программы в Windows 7 и выше. Предупреждаю любителей ретро игрушек – программа не поддерживает графику, звук и т.п. Но для наших целей она подходит идеально. Эту замечательную программу я также включил в архив.

Распаковываем содержимое архива в любое удобное место на диске, все указанные выше файлы, а также файл PRG_18.ASM должны находиться в одном каталоге

(папке). Щелкаем по файлу 1.bat. Если в появившемся через несколько секунд окне есть сообщения об ошибках, находим и устраняем их. Чтобы найти строчки с ошибками, придется просматривать листинг. Упростить поиск ошибок можно, если запустить файл 2.bat. Он запускает ассемблер x8051.exe, который выводит ошибки на экран.

Одна деталь. Ассемблер X8051 не признает регистры, которых нет в классическом 8051. Если в исходном тексте есть обращение к SFR по несуществующим в 8051 адресам, появляется сообщение об ошибке. В программе есть обращение к регистрам AUXR и AUXR1, которых нет в адресном поле классического контроллера 8051, поэтому ассемблер X8051 выдает 4 сообщения об ошибках. Как решить эту проблему, я не стал разбираться, просто игнорируем эти сообщения.

В ассемблере ASM51 такой проблемы нет, но там труднее искать строки с ошибками в тексте программы. Поэтому я использую X8051 только для поиска ошибок, а HEX файл прошивки создает ASM51.

После того, как ошибок будет 0, не считая 4-х сообщений о несуществующих регистрах, запускаем 3.bat. На этот раз будет сообщение о более, чем 50 ошибках, но не обращаем на него внимания. Просто компилятору не нравятся русские буквы в тексте программы. В папке появился файл PRG_18.hex – это готовая к использованию прошивка.

Если ничего менять в программе не собирается, то ничего этого делать не нужно. Файл прошивки PRG_18.hex также есть в архиве [2, 3].

Подробное описание первого варианта программатора (2005 г.); прошивку AT89C51RD2 (AT89C51ED2, T89C51RD2); исходный текст программы для контроллера; утилиту Flip для программирования AT89C51RD2, AT89C51ED2; UART bootloader T89C51RD2 by Atmel (файл prgrd2.zip) вы можете загрузить с сайта нашего журнала: <http://www.radioliga.com> (раздел “Программы”), а также со странички автора по ссылкам [2, 3].



Ссылки

1. Многофункциональный универсальный программатор // <http://ra4nal.qrz.ru/prg2000.shtml>
2. Многофункциональный универсальный программатор - 2 // <http://ra4nal.qrz.ru/prgrd2.shtml>
3. Многофункциональный универсальный программатор - 2 // <http://ra4nal.lanstek.ru/prgrd2.shtml>