

## Многофункциональный программатор



- Любой компьютер, любая ОС
- PROM, EPROM, EEPROM, FLASH, I2C, MCS-51, PIC, AVR
- Параллельный интерфейс для AVR
- RS-232 или USB
- Русскоязычная кодировка DOS, WIN, KOI-8
- Регулируемое напряжение программирования
- Возможность модернизации
- Доступная элементная база

В настоящее время разработано большое количество различных типов программаторов, как любительских конструкций, так и профессиональных, выпускаемых серийно. Обычно программатор управляется персональным компьютером, к которому он подключается через стандартный порт – параллельный или последовательный.

Но стандартная аппаратно-программная конфигурация даже самых быстрых компьютеров не позволяет точно сформировать интервалы времени меньше нескольких сотен миллисекунд. Для программирования же современных ПЗУ необходимы импульсы длительностью порядка единиц микросекунд и менее.

Наиболее остро эта проблема стоит в многозадачных операционных системах, поэтому перед программированием микросхемы часто приходится перезагружать компьютер в режиме эмуляции MS DOS, что весьма неудобно, особенно, если в процессе отладки придется повторять эту операцию несколько раз. А версии Windows, начиная с Millenium, вообще не поддерживают режим «чистой» MS DOS.

Предлагаемый программатор является попыткой создать простое и дешевое устройство для использования в любительских условиях с минимальными требованиями к компьютеру и с возможностью расширения номенклатуры программируемых микросхем. Поддерживаются наиболее часто применяемые в радиолюбительской практике классы микросхем: PROM, EPROM, EEPROM, FLASH, MCS-51, PIC12, PIC16, PIC18, ATMEL AVR и микросхемы памяти с интерфейсом I2C.

Основное отличие данной конструкции от других – это возможность работы с любым компьютером, под управлением любой операционной системы. Не обязательно Windows или DOS, даже не обязательно IBM - совместимый компьютер. Важно, чтобы имелся COM-порт и стандартная коммуникационная программа для работы с ним. Если нет COM-порта, программатор можно подключить и к USB через адаптер USB-COM.

С целью упрощения и удешевления конструкции, а также для обеспечения максимальной универсальности и возможности модернизации, пришлось отказаться от соблазна программировать все микросхемы, вставляя их в одну универсальную

панельку. Для каждого класса ПЗУ используется простой сменный модуль, который подключается к универсальному разъему программатора. В большинстве случаев это просто переходник между разъемом и панелькой под микросхему. Следует отметить, что в отличие от профессиональных разработок, данная конструкция позволяет программировать уже устаревшие, но еще иногда применяемые в любительской практике ПЗУ с пережигаемыми переключками серии 556PT... и 155PE3. Программировать эти микросхемы в одной универсальной панельке вообще вряд ли возможно.

### **Работа с программатором.**

Пользователь управляет процессом программирования, набирая на клавиатуре компьютера команды и наблюдая на экране результаты их выполнения. Все команды состоят из одной или двух латинских букв, которые можно набирать, как в верхнем, так и в нижнем регистре клавиатуры. За командой могут следовать один, два или три цифровых параметра, разделенных пробелом. Не значащие нули слева можно не вводить, а команду редактировать клавишей "Back Space". Ввод завершается нажатием "Enter". В случае какой-либо ошибки при вводе выдается сообщение "ERROR", а на выполнение команд, последствия которых могут быть необратимы для программируемой микросхемы, запрашивается подтверждение. Полный список всех доступных команд постоянно присутствует на экране, поэтому, даже если пользоваться программатором один раз в год, не придется что-либо вспоминать.

На компьютере должна быть запущена какая-либо коммуникационная программа, например «Telex», входящая в комплект Norton Commander 5.0, «Terminal» из Windows и т.п. Я обычно пользуюсь простой, но очень удобной программой «Tera Term Pro 2.3».

Настраивая коммуникационную программу, необходимо удалить строки инициализации модема, отключить режимы "Локальное эхо" и "Трансляция CR/LF" и выбрать терминал ANSI. Кодовая таблица может быть ASCII, Windows или KOI-8. Режим работы последовательного порта, к которому подключен программатор, должен быть следующим: скорость – 1200, 2400, 4800, 9600, 14400, 19200,

28800 или 57600 Бод, число бит – 8, контроля четности нет, 1 стоп-бит, управление потоком - аппаратное (RTS/CTS) или управления нет.

Включив питание программатора необходимо нажать на клавиатуре компьютера любую буквенно-цифровую клавишу в латинском регистре, т.е. с кодом менее 128. При этом программатор автоматически определит скорость работы порта и будет посылать информацию в компьютер на этой же скорости. Если все сделано правильно, на экране появится предложение выбрать кодовую таблицу, с которой будет работать программатор. Как указывалось выше, поддерживается кодировка DOS, Windows и KOI-8. Все дальнейшие сообщения будут выводиться на экран на русском языке. Если символы не читаемы, значит кодовая таблица выбрана неверно и следует выключить и снова включить программатор, а затем выбрать другую кодировку.

После этого необходимо выбрать из списка класс и тип ПЗУ, с которым предполагается работа, после чего на экране появится меню с указанием формата всех доступных команд. Можно загрузить файл в ОЗУ программатора, просмотреть его DUMP в символьном и 16-ричном виде, изменить содержимое ячеек или переместить блок ОЗУ, сохранить содержимое ОЗУ в файле, ну и, конечно, запрограммировать микросхему. Все команды достаточно простые и понятные, поэтому подробно описывать каждую из них нет смысла.

Необходимо отметить только следующее:

- Объем памяти программатора ограничен, поэтому в списке поддерживаемых микросхем приведены далеко не все возможные типы. Например, выбрав 2764, можно запрограммировать и ее аналог - K573PФ4 или, выбрав PIC16CE625 можно запрограммировать PIC16C622. Поэтому, если в списке нет нужной микросхемы, вполне возможно, что можно выбрать другую, имеющую подобный алгоритм программирования. Более того, т.к. имеется возможность вводить начальный и конечный адреса программирования, фактически можно работать с очень многими ПЗУ, например с большинством PIC16C/F 6,7,8,9..., хотя в списке приведено всего несколько типов. Для облегчения поиска аналога на экран выводятся основные характеристики выбранного типа.

- Программатор позволяет загрузить в свое ОЗУ HEX или BIN файл. Поддерживаются форматы Intel HEX8 и Intel HEX32. HEX файл загружается по протоколу ASCII в соответствии с определенными в нем адресами, а BIN - по стандарту X-Modem с адреса 0000H. Аналогично можно сохранить содержимое ОЗУ в файле. При сохранении HEX файла можно задать паузу между символами, если быстрое действие компьютера недостаточно для работы на выбранной скорости. Сохраняется область ОЗУ, соответствующая объему выбранного типа ПЗУ.

- Команда "Принять текстовый файл" имеется далеко не во всех терминальных программах, поэтому сохранить на диске HEX файл с содержимым

ПЗУ можно следующим образом. Вначале нужно набрать на клавиатуре команду **SH** (сохранить HEX файл), затем в терминальной программе включить режим протокола, при котором вся выводимая на экран информация будет сохраняться в файле протокола. После нажатия «Enter» начнется передача содержимого ОЗУ программатора. Когда передача закончится, режим протокола нужно отключить и нажать «Enter». Теперь следует открыть файл протокола в любом текстовом редакторе и удалить из него лишнюю информацию, если она там оказалась из-за неверно выбранного момента включения и отключения режима протокола.

- Если выбрана работа с PIC16 или ATMEL AVR, все данные 16-битные, при этом в HEX файле сначала следует младший байт, а затем старший. Слово конфигурации для PIC16xxxx размещается в ОЗУ по стандартному адресу 2007H, а область памяти данных для FLASH PIC16Fxxx начинается с адреса 2100H. При работе с другими типами ПЗУ все данные 8-битные.

- Информация в ОЗУ программатора должна размещаться по тем же адресам, по которым она будет записываться в ПЗУ, т.е. содержимое 0-й ячейки ОЗУ будет записано в 0-ю ячейку ПЗУ и т.д. Исключением является PIC18. Для этого класса области ID Locations, Configuration words и Data Memory начинаются в ОЗУ с FD00H, FE00H и FF00H соответственно. Но при сохранении в HEX файле они будут иметь стандартные адреса 200000H, 300000H и F00000H. Если сохранить содержимое ОЗУ в BIN файл, то ID Locations, Configuration words и Data Memory не будут сохранены.

Программатор предназначен в первую очередь для радиолюбителей - конструкторов, т.е. для тех, кто знает, что представляет собой программируемая микросхема. Поэтому, с целью обеспечить возможно большую универсальность данного устройства, пришлось пожертвовать некоторыми сервисными функциями. В частности, слово конфигурации для PIC контроллеров нужно вводить вручную, а не ставить привычные флажки типа "защита включена-выключена". Однако, это не является, на мой взгляд, серьезным недостатком, т.к. при ассемблировании слово конфигурации может быть включено в HEX файл.

Программирование ATMEL AVR имеет некоторые особенности. В отличие от фирмы «Microchip», которая для PIC контроллеров четко определила размещение в файле прошивки области памяти программ, данных и слова конфигурации, ATMEL предлагает использовать для памяти данных отдельный файл, а FUSE биты вообще записывать "на бумажке". На мой взгляд это неудобно. Поэтому я решил максимально приблизиться к стандарту "Microchip" и разместить память данных (EEPROM) в файле прошивки непосредственно за памятью программ (FLASH), а после нее LOCK и FUSE биты. И все это считается 16-битным. Т. к. память данных 8-и битная, старший байт каждого слова равен нулю (так делает Microchip для PIC16).

Например, для AT90S1200 память программ занимает адреса в ОЗУ программатора с 0000 до 01FF, а память данных с 0200 до 023F. Затем (для AT90S1200 по адресу 0240) следует слово, старший байт которого это - xxxxx21x, где 2 и 1 - это LOCK биты 2 и 1 соответственно. Младший байт - это FUSE биты в том виде, в каком они определены "ATMEL" документации на соответствующий контроллер.

Такое размещение информации очень легко задать при ассемблировании. После текста собственно программы вместо директивы **.eseg** следует перейти на адрес, следующий непосредственно за памятью FLASH, например для AT90S1200: **.org \$0200** и, как обычно, разместить информацию для записи в память данных EEPROM директивами **.db** ассемблера. Получится как раз то, что нужно: каждый байт запишется в виде слова, у которого старший байт равен нулю. Затем, перейдя на адрес, следующий за EEPROM, (для AT90S1200 - **.org \$0240**) директивой **.dw** можно записать LOCK и FUSE биты - это старший и младший байт соответственно. Конечно, AVR Studio будет ругаться, но сделает все правильно.

Стандартный файл прошивки FLASH полностью подходит для загрузки в память программатора. А вот файл памяти данных с расширением **.eep** нужно предварительно изменить. Это, конечно, недостаток, но, если вы пишете собственную программу, он не имеет значения, т.к. на этапе написания исходного текста можно включить данные EEPROM в файл FLASH, как описано выше. При этом ничто не мешает создать и стандартный **.eep** файл для пользователей других типов программаторов.

Преобразовать стандартный файл **.eep** в файл, пригодный для загрузки в этот программатор можно вручную, ведь его объем обычно не велик. FUSE биты можно ввести непосредственно в программаторе или не программировать их вообще, оставив значения "по умолчанию". Зато в одном файле можно будет иметь полную информацию для прошивки микроконтроллера и не ломать впоследствии голову, какие FUSE надо запрограммировать, а какие нет.

Для ATMEL AVR поддерживаются все режимы программирования: как Low voltage, так и High voltage. Это объясняется тем, что в наиболее простом и широко используемом режиме Low voltage не для всех типов AVR и не все FUSE биты можно запрограммировать. А некоторые типы вообще не поддерживают этот режим программирования. Более того, изменив значения определенных FUSE бит, можно запретить этот режим. А кто может быть уверенным, что не сделает этого по ошибке или незнанию? В этом случае ничего невозможно будет поделать, кроме как высказать все, что вы думаете о разработчиках фирмы ATMEL и выбросить микросхему. Поэтому, приведенная ниже схема сменного блока для программирования AVR, рассчитана только на режим "High voltage". Режим же "Low voltage" предполагается использовать исключительно для внутрисхемного программирова-

ния при отладке. Разумеется, можно сделать и для него сменный блок, который получится намного проще.

Из этих же соображений сменный блок MCS-51 рассчитан на программирование AT89S53 и AT89S8252 в параллельном режиме. Последовательное программирование этих контроллеров предполагается использовать только внутрисхемно. Их подключение аналогично AVR, поэтому интерфейсный разъем ISP на блоке MCS-51 можно не устанавливать, а пользоваться для внутрисхемного программирования разъемом на сменном блоке для AVR. Следует обратить внимание на то, что при последовательном программировании выводы PSEN (29) и ALE/PROG (30) должны висеть в воздухе.

Следует остановиться еще на одной особенности ATMEL AVR. Не все типы программируются корректно при напряжении питания 5 В. Поэтому, в схему сменного блока (рис.7) введен тумблер SA1, который позволяет понизить напряжение питания до 3,8 В. В некоторых случаях это оказывает неоценимую помощь и позволяет сберечь нервы и время. Диоды VD1 и VD2 должны выдерживать ток до 1 А на случай короткого замыкания.

Для микросхем памяти с интерфейсом I2C можно задать Slave адрес и время записи страницы. Это может потребоваться, если придется программировать какую-либо нестандартную память. Старшая тетрада Slave адреса - это номер группы. Младшая тетрада - адрес устройства, задаваемый соединением входов A0, A1 и A2 с общим проводом или +Уп. Стандартный Slave адрес для микросхем памяти - A0H, а время записи страницы - 0AH (10ms). Также предусмотрена возможность соединения 7-й ножки программируемой микросхемы с общим проводом или Vcc (а вдруг потребуется).

Последняя версия прошивки программатора поддерживает следующие классы и типы микросхем:

**PROM:** 155PE3, 556PTxx  
**EPROM:** 2716...27512, 27C64...27C512  
**EEPROM:** W27C512, 28C16A, 28C17A, 28C64A, AT28C64B, AT28C256, AT29C256/257/512  
**MCS-51:** 8751, 87C51, 87C51FA, 87C51FB, 87C51FC, 87C51RD+, AT89C51, AT89C52, AT89C55, AT89C1051, AT89C2051, AT89C4051, AT89S53, AT89S8252  
**PIC16:** PIC12F629/675, PIC16F84/C84, PIC16F627/A, PIC16F628/A, PIC16F648A, PIC16F73/74, PIC16F76/77, PIC16F873/874, PIC16F876/877, PIC16F876A/877A, PIC16C5xx, PIC16C6x, PIC16CE6xx, PIC16C7x  
**AVR:** AT90S1200, AT90S2313, AT90S2323/2343, AT90S2333, AT90S4414/4434, AT90S4433, AT90S8515/8535, ATtiny10/11, ATtiny12, ATtiny15, ATtiny28  
**I2C:** 24LC01...24LC512, PCA8581, PCF8582, ST24E16  
**PIC18:** PIC18Fxx2/xx8  
**PIC12:** PIC12C508/509, PIC12F508/509, PIC16F505

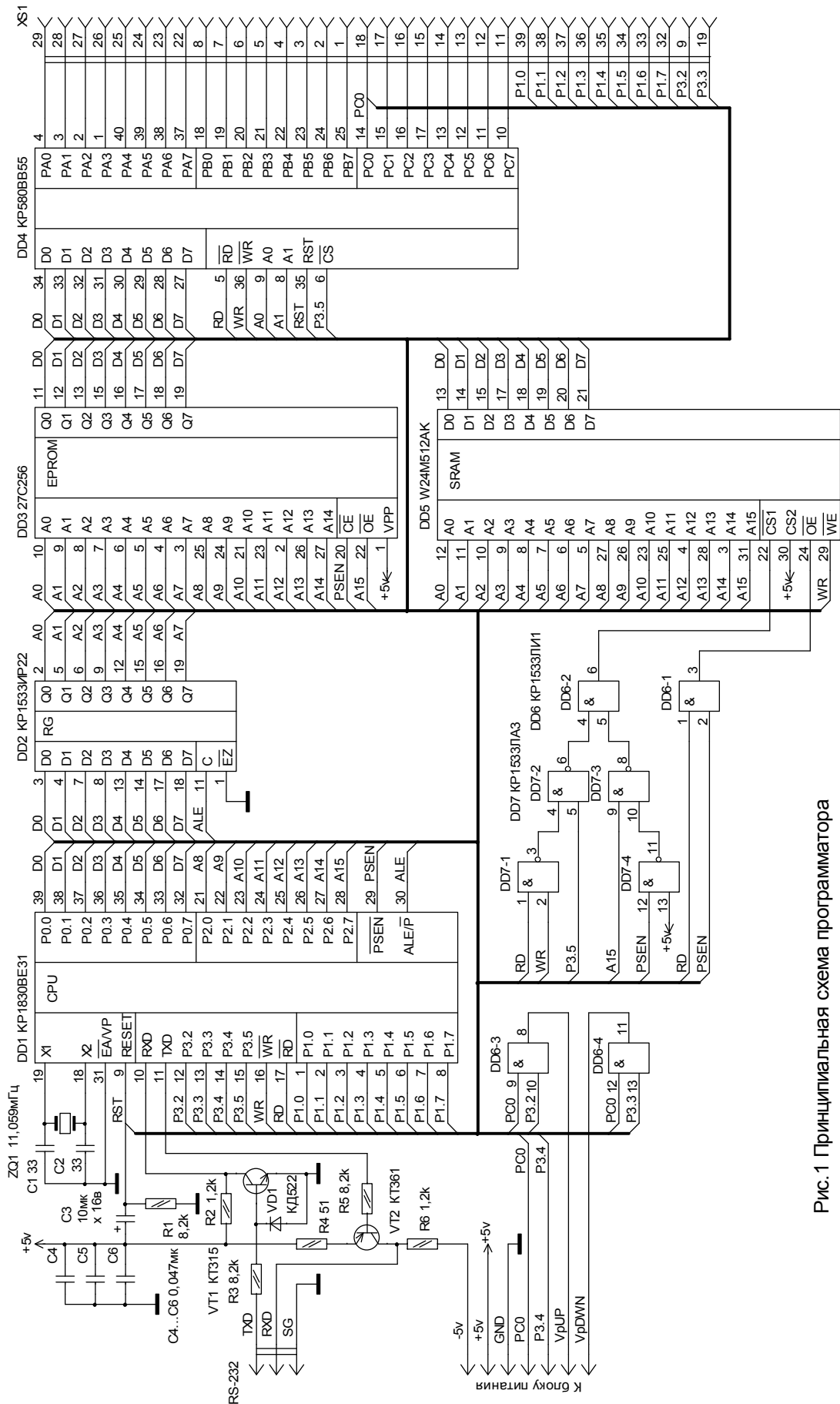


Рис.1 Принципиальная схема программатора

Для расширения номенклатуры программируемых микросхем есть несколько путей. Во-первых, по мере появления у автора возможности, необходимости и желания, программа модернизируется. Во-вторых, архитектура программатора открытая, поэтому любой желающий, конечно, при наличии достаточных знаний и опыта, может написать свою собственную программу. Наконец, в-третьих, можно разработать программный модуль для нужного типа микросхемы и непосредственно перед программированием загружать его в верхнюю половину ОЗУ в виде HEX файла, а затем запускать на выполнение командой G <adress> из меню. Для корректного возврата в основную программу этот модуль должен заканчиваться командой L JMP 0080H. Необходимо также сохранять неизменным содержимое четырех ячеек внутреннего ОЗУ микроЭВМ - 7CH, 7DH, 7EH и 7FH. Все остальные ресурсы - в распоряжении программиста.

### Особенности принципиальной схемы

Основной блок программатора состоит из двух функциональных узлов, размещенных в одном корпусе - собственно программатора и блока питания.

Программатор (рис. 1) представляет собой универсальный микропроцессорный контроллер на базе однокристалльной микроЭВМ КР1830ВЕ31 (80С31). В его состав входят следующие узлы: ПЗУ DD3 для хранения управляющей программы, ОЗУ DD5 для хранения данных, регистр для демультимплексирования адреса и данных DD2, параллельный порт для связи с программируемой микросхемой DD4, а также узлы приемника и передатчика RS-232C, осуществляющие преобразование уровней - VT1 и VT2. Все эти элементы включены по стандартной схеме. Следует отметить только одну важную особенность. На элементах DD6.1, DD6.2 и

DD7 выполнен дешифратор адреса, который формирует совместную область памяти программ и данных по адресам 8000H...FFFFH. Таким образом, первую половину памяти программ по адресам 0000H...7FFFH занимает ПЗУ DD3, а вторую половину - ОЗУ DD5. Вся память данных по адресам 0000H...FFFFH занимает ОЗУ.

Эта особенность позволяет упростить разработку и отладку новых версий программного обеспечения программатора. Нет необходимости применять для этого дорогостоящие аппаратные эмуляторы, достаточно просто оттранслировать программу с адреса 8000H, загрузить HEX файл в ОЗУ и запустить его на выполнение командой G8000. Если ожидаемый результат не достигнут, следует внести в программу необходимые изменения и повторить загрузку и запуск. При этом в текст программы можно вставить точки останова, вывод служебных сообщений и т.д. Полностью отлаженная программа транслируется с адреса 0000H и записывается в ПЗУ, которое затем устанавливается на место DD3.

Таким образом, данную конструкцию можно использовать не только по прямому назначению, но и для каких-либо иных целей, ведь в распоряжении пользователя 4 двунаправленных 8-и разрядных порта ввода-вывода и последовательный интерфейс связи с компьютером.

Блок питания (рис. 2) вырабатывает все необходимые для работы программатора напряжения. Напряжение питания программируемой микросхемы стабилизируется микросхемой DA1. Ключ на транзисторах VT3 и VT5 позволяет отключать питание ПЗУ, когда к нему не производится обращение. Напряжение на коллекторе VT3 может принимать два фиксированных значения - 5 или 6,5 вольт, последнее необходимо для программирования ПЗУ с пе-

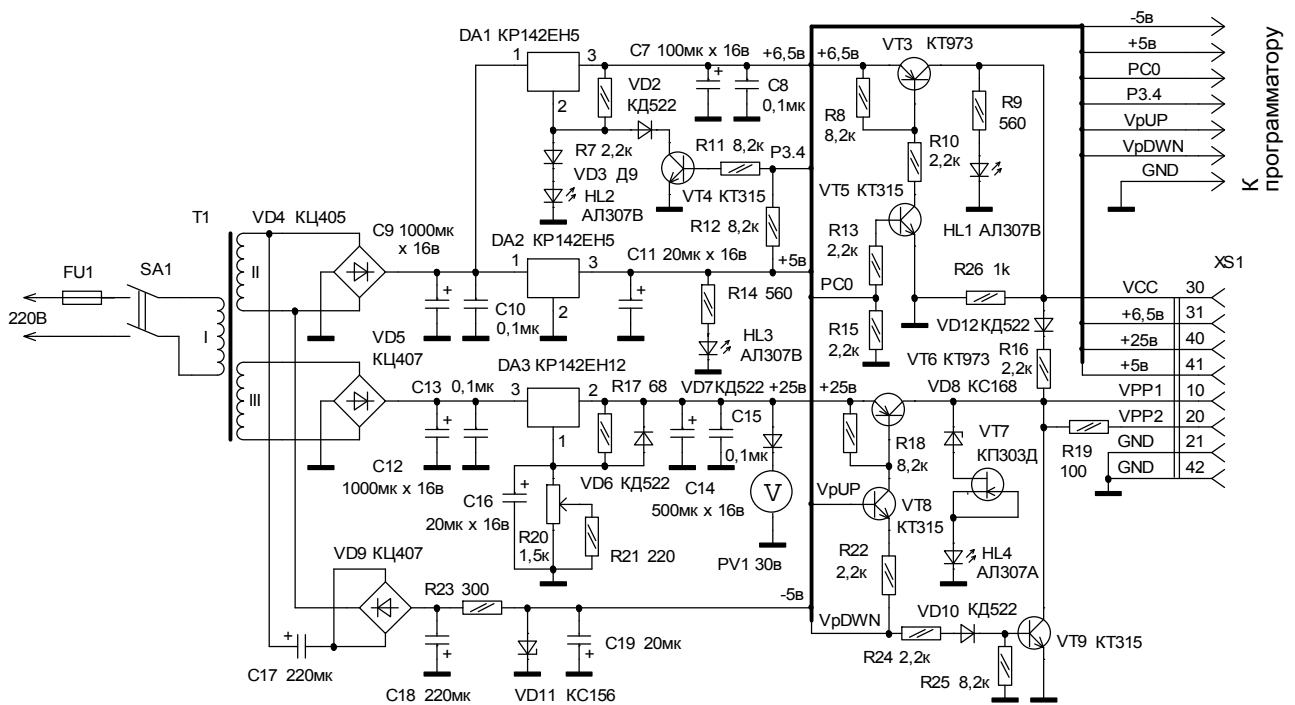


Рис.2 Принципиальная схема блока питания

режигаемыми переключками 556PT..., а также с УФ стиранием серии 27С... Коммутация осуществляется транзистором VT4.

На микросхеме DA2 собран стабилизатор на 5 вольт, который необходим для питания самого программатора. Напряжение программирования вырабатывает стабилизатор DA3, с помощью резистора R20 оно может регулироваться от 5 до 25 вольт. Контроль осуществляется стрелочным вольтметром PV1. Не эстетично? Зато дешево, надежно и практично! Благодаря ключу на транзисторах VT6, VT8, VT9 это напряжение может принимать нулевое значение, равняться напряжению питания (Vcc), или напряжению программирования (Vpp). Стабилизатор на VD11 необходим для работы преобразователя уровня интерфейса RS-232C.

Для программатора разработаны 6 сменных модулей:

- EPROM, EEPROM (рис. 3-1 и 3-2). Позволяет программировать ПЗУ с УФ и электрическим стиранием объемом до 64 Кбайт: 27xxx; 27Сxxx; W27С512; 28Сxxx; 29Сxxx; 573РФ2, 4, 5, 6, 8. Разработано два варианта этого модуля. В первом из них (рис. 3-1) тип ПЗУ выбирается с помощью переключателей SA1...SA4, которые коммутируют напряжение Vcc и Vpp на нужные выводы панельки. Во втором (рис. 3-2) эта коммутация осуществ-

ляется автоматически с помощью герконовых реле. Вариант, конечно не самый лучший, но зато самый простой. Для опытных радиолюбителей не составит труда разработать и третий вариант модуля – с электронной коммутацией. Для работы с этим модулем необходима прошивка версии не ранее декабря 2003г.

- MCS-51 (рис. 4). Для программирования микроконтроллеров семейства MCS-51: 87С51, АТ89С51/52/55, АТ89С2051, АТ89С53, АТ89С8252 и др., которые устанавливаются в панельки DIP-40 и DIP-20.

- PIC (рис. 5). Программируются FLASH микроконтроллеры PIC12Fxxx, PIC16Fxxx, PIC18Fxxx, а так же с УФ стиранием и однократно программируемые (PIC12С5xx, PIC16Сxxx и др). Возможно внутрисхемное программирование с запуском записанной программы на выполнение.

- PROM (рис. 6). Программирование ПЗУ с пережигаемыми переключками 556PT... и 155PE3. Самый сложный модуль, содержит три панельки, два переключателя и восемь транзисторных ключевых каскадов. В XXI веке реально может потребоваться только для программирования 556PT4 и 155PE3, поэтому из схемы вполне можно исключить не используемые элементы или не делать этот модуль вообще.

- AVR (рис. 7). Программируются все ATMEL AVR микроконтроллеры семейств Classic и Tiny, производимые на начало 2002 года в DIP корпусах. Блок предназначен для программирования по алгоритму «High voltage programming», но допускает и внутрисхемное программирование непосредственно в устройстве по алгоритму «Serial Low voltage» с запуском загруженной программы на выполнение.

- I2C (рис. 8). Предназначен для программирования ПЗУ с интерфейсом I2C объемом от 128 байт до 64 кбайт.

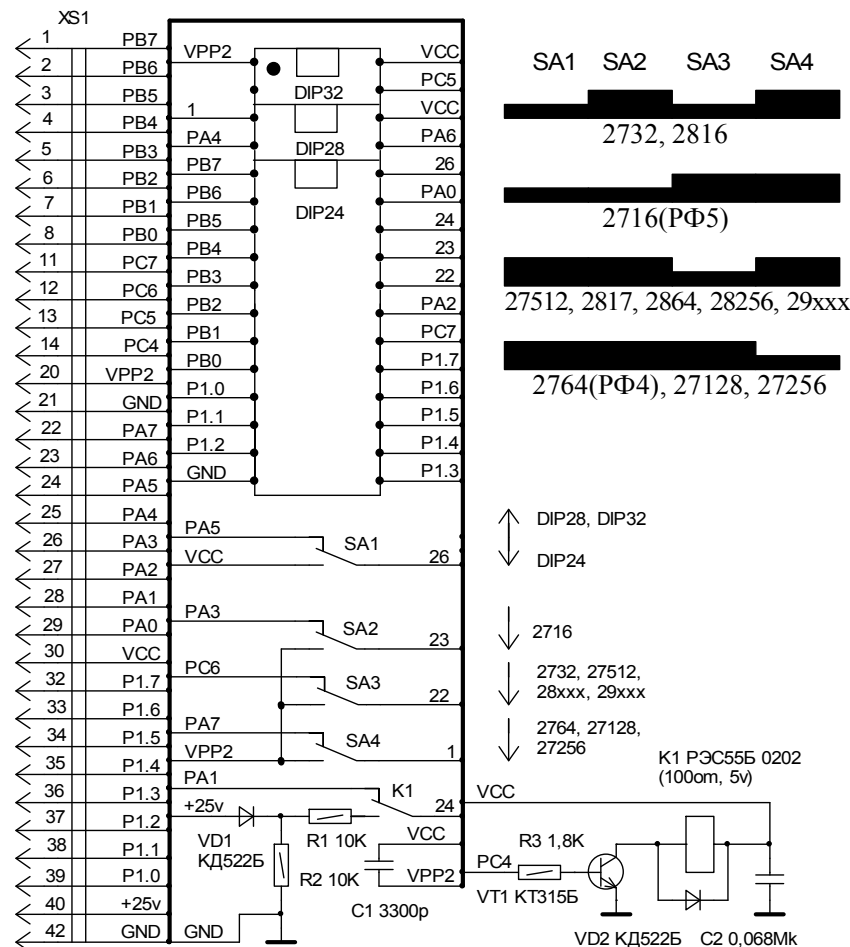


Рис.3-1 Сменный блок EPROM, EEPROM, FLASH  
(С ручным выбором типа)

### Конструкция и детали.

Как указывалось выше, программатор состоит из основного блока и сменных модулей. На лицевой панели основного блока размещаются тумблер включения питания, универсальный разъем для подключения сменных модулей, стрелочный вольтметр, регулятор программирующего напряжения и три светодиода, индицирующие включение питания, Vcc и Vpp.

Все детали конструкции размещаются на двух печатных платах, которые в процессе отладки претерпели многочисленные изменения, поэтому их чертежи не приводятся. Наиболее дефицитным

является ОЗУ DD5 – микросхема W24M512AK или ее аналог UM61512AK. Это статическое ОЗУ на 64 Кбайт, широко применявшееся в материнских платах 286-386-486 компьютеров. Можно использовать микросхемы не только этих типов, но и любые другие статические ОЗУ, в названии которых есть цифры 512 в корпусе на 32 ноги.

Объем ОЗУ определяет только максимальный объем программируемой микросхемы. Т.е. внешнее ОЗУ используется только в качестве буфера для программирования. Поэтому, можно использовать статическое ОЗУ и меньшей емкости, например, 62256 или аналогичное, с цифрами в названии 256 в корпусе на 28 ног. Цоколевка совпадает, 1-ю ногу нужно вставлять в панельку на место 3-й и т.д., 28-ю на место 30-й. Аналогично 573РФ5 и 573РФ4. Подобные ОЗУ более широко распространены. Разумеется, программировать можно будет только микросхемы до 32кб.

В программаторе следует использовать ПЗУ 27256 или 27С256 с временем доступа не более 150...200 ns. В большинстве случаев это требование выполняется, однако, если попадется экземпляр начала 80-х годов, да к тому же многократно перепрограммированный, программа может вести себя совершенно непредсказуемо. Поэтому, рекомендую использовать микросхему памяти, выпущенную не

ранее 1990г. и запрограммированную не более 10 раз. Еще лучше использовать FLASH память 29С256 или W27С512. В первом случае цоколевка полностью совпадает с 27С256, а во втором – следует вывод 1 W27С512 подсоединить к "земле", а не к +5 В (рис. 1) или записать управляющую программу в ее верхнюю половину с адреса 8000Н. Для гарантии можно записать программу дважды - с адреса 0000Н и с адреса 8000Н. В саму прошивку никаких изменений вносить в этом случае не нужно.

Кварцевый резонатор ZQ1 должен быть на частоту 11,059МГц. В качестве преобразователей уровня интерфейса RS-232 можно применить специализированную микросхему MAX232, тогда отпадет необходимость в стабилизаторе -5 В. Можно, при необходимости, обеспечить и гальваническую развязку с компьютером используя стандартные схемные решения.

Напряжение на обмотке II трансформатора Т1 должно быть около 10 вольт при токе до 0,5 А, а на обмотке III - 25 вольт при токе до 0,1...0,2 А. Микросхемы стабилизаторов следует разместить на радиаторе.

Универсальный разъем должен быть механически прочным и обеспечивать в течение длительного времени надежный электрический контакт, я использовал приборный разъем типа РП10-42 на 42 контакта. Следует стремиться к минимальной длине соединительных проводов от платы до этого разъема. Переключатели в сменных модулях типа П2К, реле лучше использовать герконовые, например РЭС55Б с сопротивлением обмотки 100 Ом, рабочим напряжением 5 В и током срабатывания около 50 МА (паспорт 0202).

**Налаживание.**

При безошибочном монтаже и отсутствии ошибок в программе цифровая часть наладки не требует. Следует только иметь в виду, что некоторые малоактивные кварцы плохо возбуждаются в этой схеме. В блоке питания необходимо подобрать элементы VD2, VD3 и HL2, чтобы напряжение на коллекторе VT3 могло принимать значения точно 5 и 6,5 вольт. Следует также уточнить номиналы резисторов R17 и R21 так, чтобы пределы регулирования напряжения программирования Vpp резистором R20 были 5...25 вольт по вольтметру PV1.

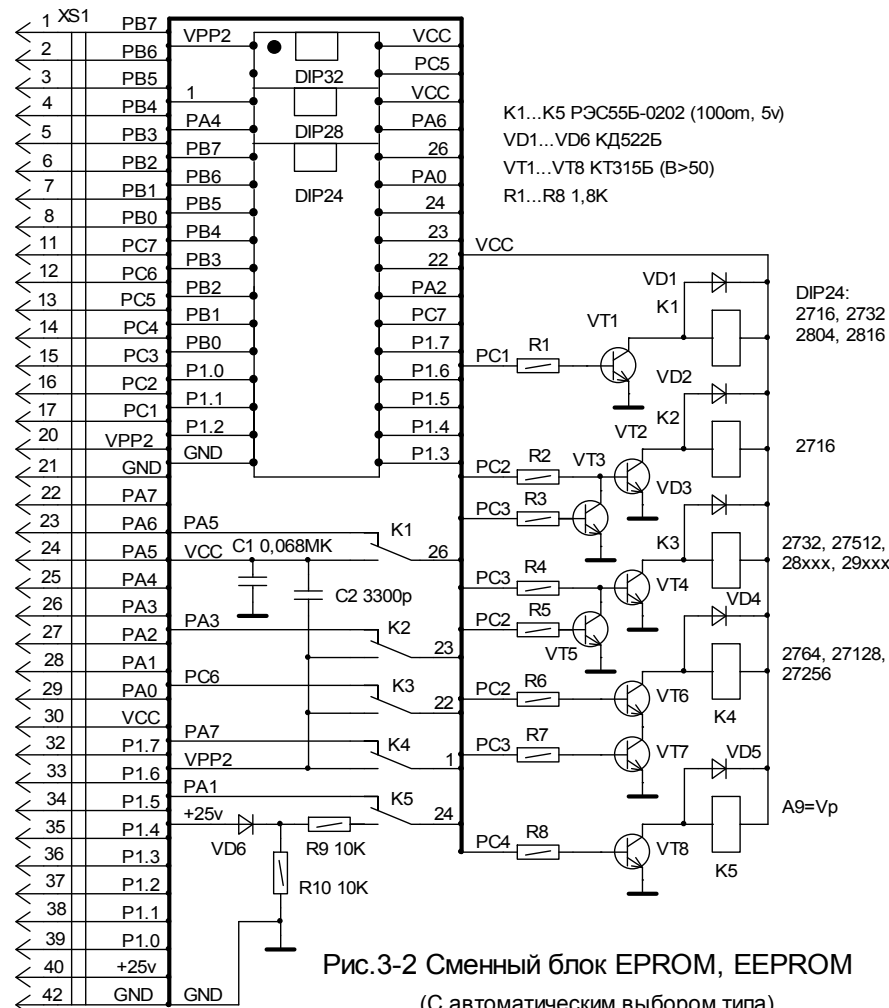


Рис.3-2 Сменный блок EPROM, EEPROM  
(С автоматическим выбором типа)

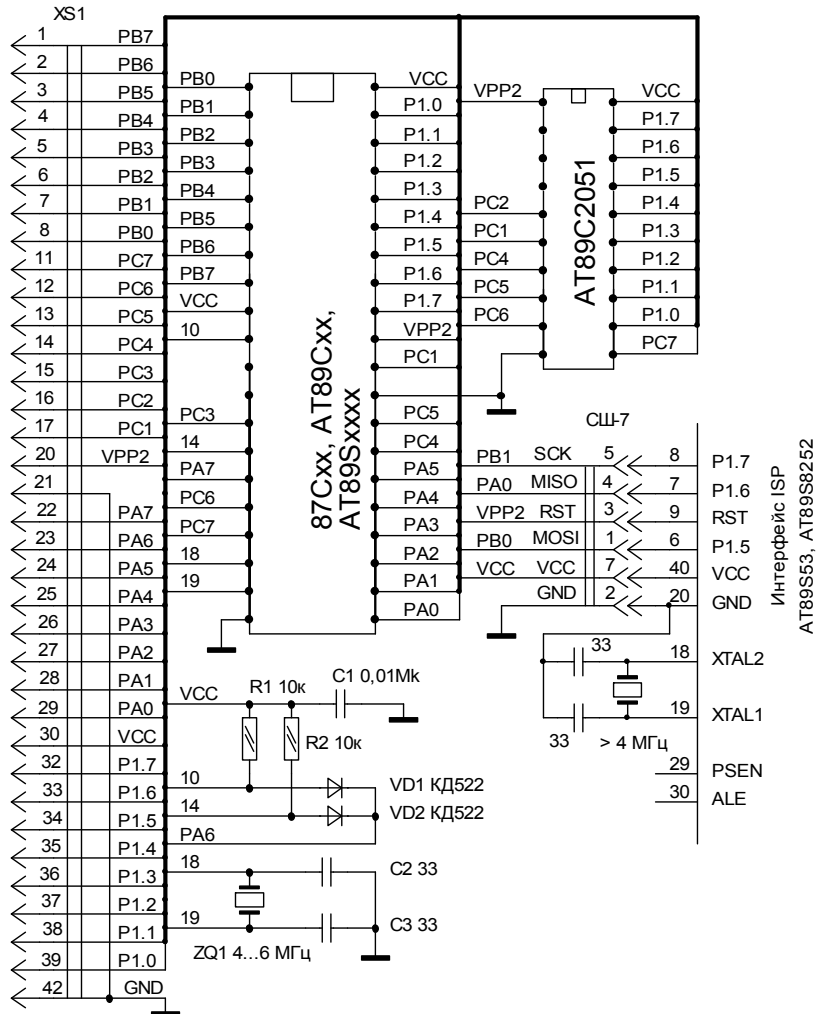


Рис.4 Сменный блок для MCS-51

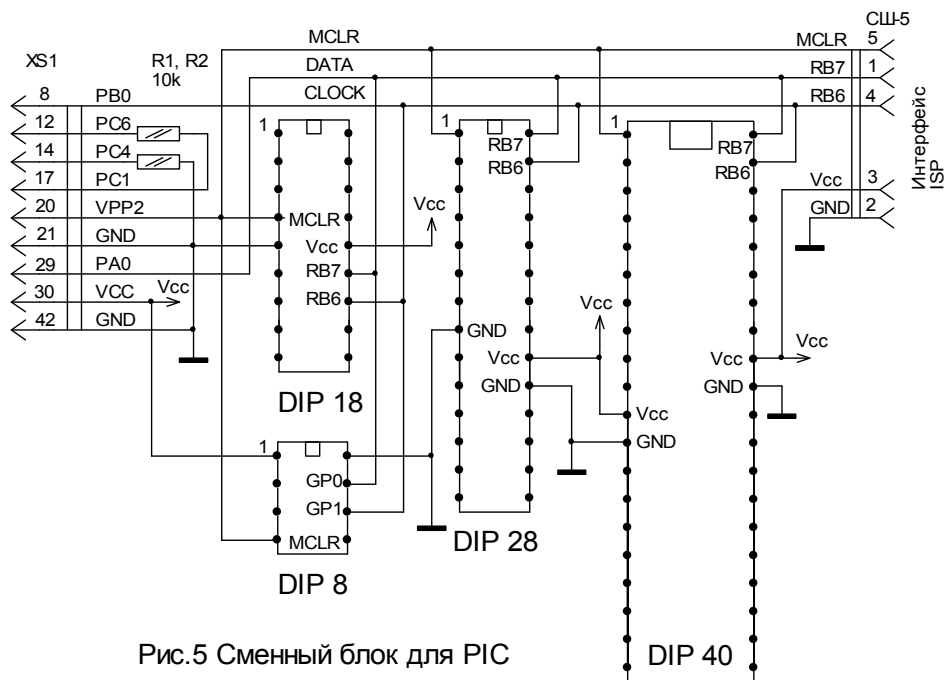


Рис.5 Сменный блок для PIC



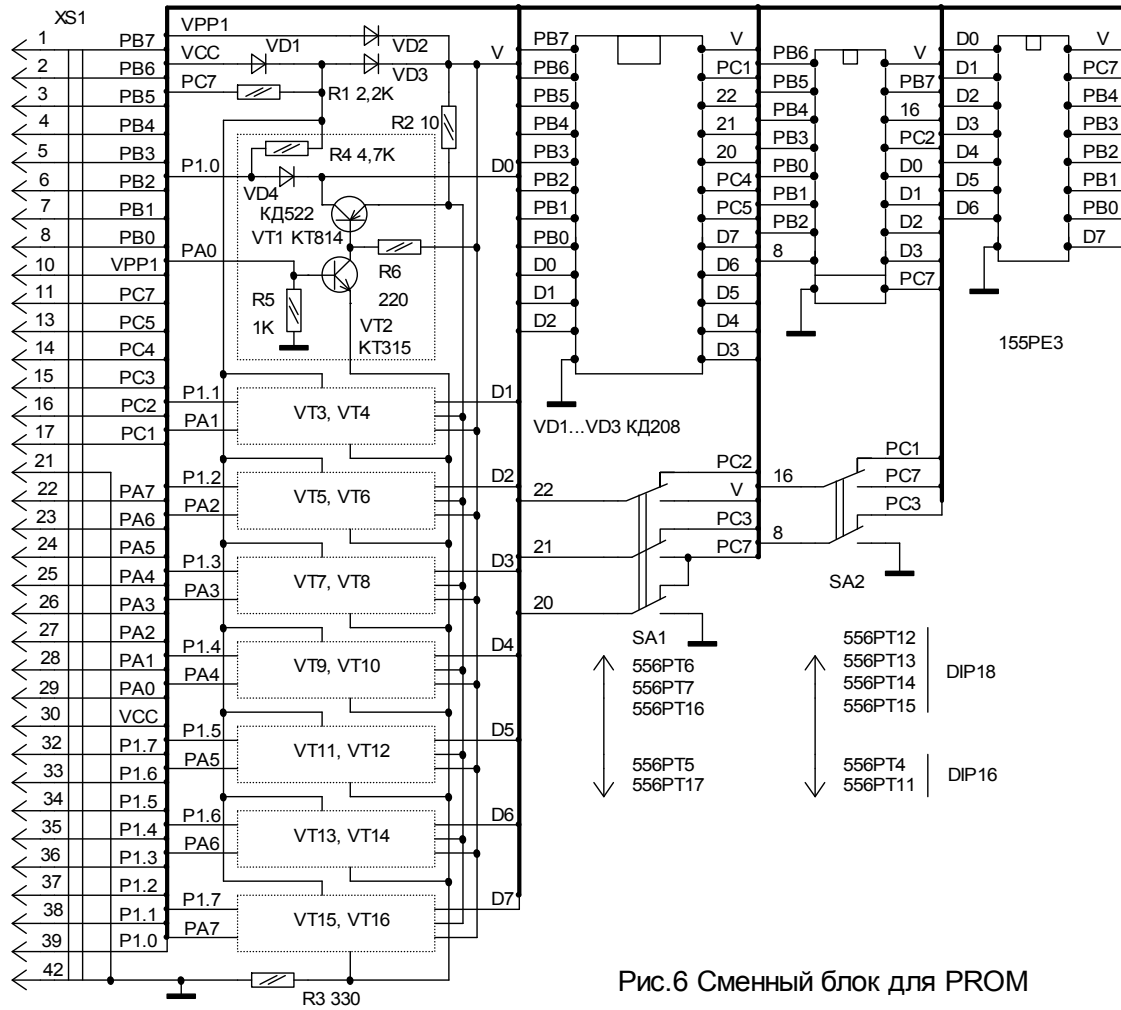


Рис.6 Сменный блок для PROM

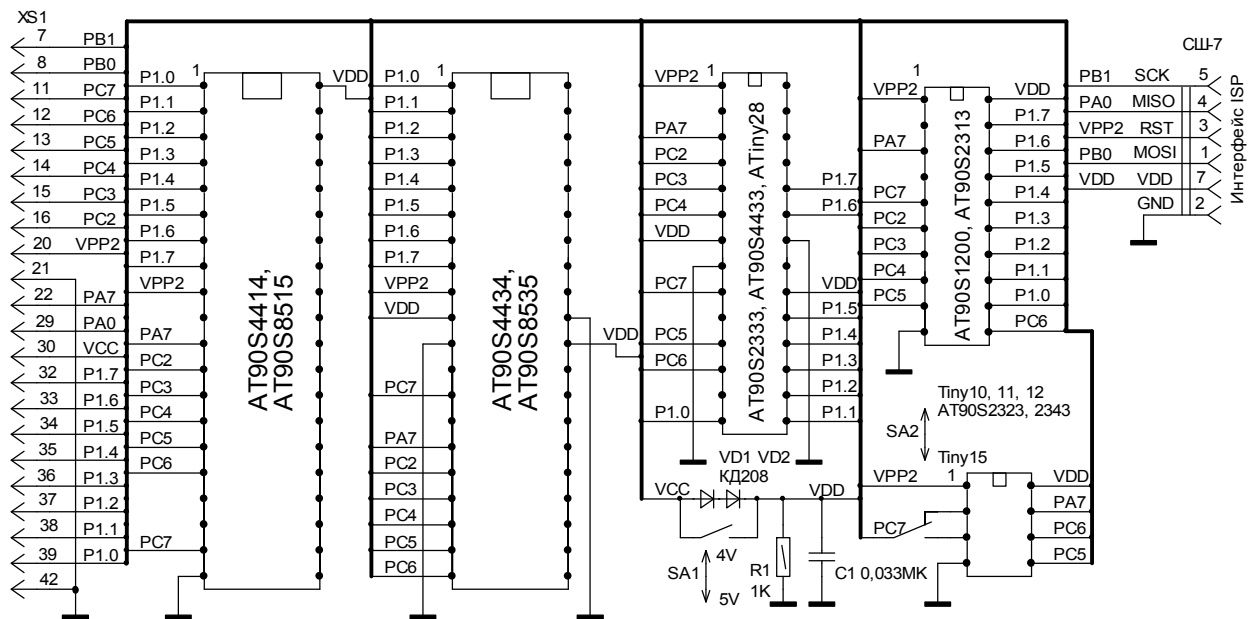


Рис.7 Сменный блок для AVR Classic и Tiny (High voltage programming).

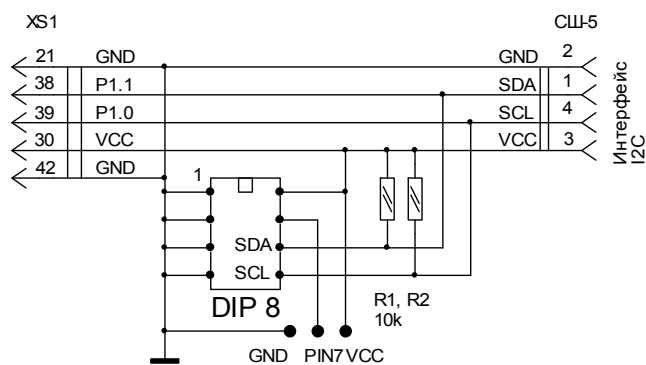


Рис.8 Сменный блок для I2C

С целью экономии средств пользователей и упрощения процедуры Upgrade, наряду с базовым программным обеспечением для программатора разработана своего рода "аварийная дискета" – сокращенная версия управляющей программы, которая помещается в ПЗУ типа 2764. Она поддерживает программирование только серий 27xxx, 28xxx, 29xxx. Таким образом, для того, чтобы использовать обновленную версию прошивки нужно заменить в панельке DD3 штатную ПЗУ на аварийную (Rescue), запрограммировать с ее помощью новую версию ПО и вернуть на место перепрограммиро-

ванную микросхему, вдохнув в программатор новую жизнь.

Следует учесть только один момент. На выводе 27 ПЗУ типа 2764 должен присутствовать уровень логической единицы. В 27256 этот вывод является адресной шиной A14. Проще всего обеспечить их совместимость, если непосредственно между ножками 27 и 28 микросхемы 2764 припаять резистор 10...15ком и слегка отогнуть ножку 27 так, чтобы при установке в панельку она ни с чем не соединялась. На программирование самой 2764 этот резистор не будет оказывать никакого влияния, но ножка 27 должна, разумеется, во время программирования подсоединяться куда следует.

**Прошивку контроллера и другие дополнительные материалы к этой конструкции можно загрузить с сайта автора по адресам:**

<http://ra4nal.qrz.ru>  
<http://ra4nal.lanstek.ru>  
<http://ra4nalr.tut.ru>

**Разработка 1999г.**

**Коммерческое использование с согласия автора. Перепечатка со ссылкой на первоисточник.**