Программатор
USBasp

Руководство пользователя

Илья Данилов

Программатор USBasp. Руководство пользователя
Описание составлено Ильей Даниловым

При составлении использованы материалы:
http://www.fischl.de/usbasp
http://www.nongnu.org/avrdude/

Данная работа лицензирована под лицензией
Creative Commons Attribution-Shared Alike 3.0

Эта лицензия позволяет другим перерабатывать, исправлять и развивать произведение даже в коммерческих целях при условии указания авторства и лицензирования производных работ на аналогичных условиях. Все новые произведения основанные на лицензированном под нею будут иметь аналогичную лицензию, поэтому все производные будет разрешено изменять и использовать в коммерческих целях.

Для получения полного текста лицензии, загрузите:
http://creativecommons.org/licenses/by-sa/3.0/

Или пошлите письмо по адресу:

Creative Commons
171 Second Street, Suite 300
San Francisco, California, 94105, USA

Первая редакция: 11 Окт 2009
Вторая редакция: 13 Фев 2010

Введение

Данное руководство содержит начальные сведения о простейшем программаторе микроконтроллеров ATMEL/AVR USBasp[1], разработанном Томасом Филчем (Thomas Filch), моих модификациях USBasp-id[2] и бесплатной программе avrdude[3].

Для получния исчерпывающей информации советую обратиться к документации avrdude и проекту V-USB[4].

Я буду рад любым вопросам в моем блоге mk90.blogspot.com.

Илья Данилов

Системные требования

Программатор USBasp позволяет осуществлять запись и считывание памяти микроконтроллеров семейства ATMEL/AVR, поддерживающих последовательное программирование без извлечения из схемы, или ICSP[5]. Для использования программатора потребуются:

- IBM-PC совместимый компьютер не хуже Intel Pentium, 256K RAM
- свободный порт USB 1.1
- операционная система Windows 2000 и выше, Linux с ядром 2.2 и выше или MacOS X
- установленное свободное ПО - libusb и avrdude
- схема с поддерживаемым микроконтроллером ATMEL/AVR, совместимая с сигналами TTL +5В и имеющая правильно разведенный разъем ICSP[6]

Комплектность

- программатор USBasp в разборном корпусе
- кабель ICSP 10-контактный
- кабель ICSP 6-контактный
- (опционально) CD-диск со свободным ПО и документацией[7]

Установка программного обеспечения

Windows

1. Установите `avrdude`, если он еще не установлен. Как правило, он входит в состав пакета WinAVR[8];
2. Распакуйте архивный файл `usbasp.2009-02-28.tar.gz` [9] во временный каталог;
3. Выберите драйвер `libusb`, согласно установленной версии WinAVR. Самой свежей версии соответствует подкаталог: `bin/win-driver/libusb0.1.12.1`

[1] http://www.fischl.de/usbasp
[2] http://mk90.blogspot.com/2009/11/usbasp.html
[3] http://www.nongnu.org/avrdude/
[4] http://www.obdev.at/vusb/
[5] In Circuit Serial Progamming - внутрисхемное программирование
[6] Назначение контактов разъемов см. на стр. 7
[7] Все записанные на CD материалы также доступны для скачивания в сети Internet
[8] Скачать со страницы проекта: http://winavr.sourceforge.net/
[9] Скачать со страницы автора: http://www.fischl.de/usbasp/

4. Перейдите в каталог и найдите в нем файл `usbasp.inf`, затем нажмите на нем правой кнопкой мыши, активируя контекстное меню, выберите пункт «Установить» и следуйте инструкциям системы.

После успешной установки драйвера `libusb`, в Диспетчере Задач можно наблюдать появление нового устройства USBasp при подключении программатора к компьютеру.

Linux

Установите через менеджер пакетов вашего дистрибутива `avrdude` и `libusb`. Например, для дистрибутивов Debian или Ubuntu, можно набрать:

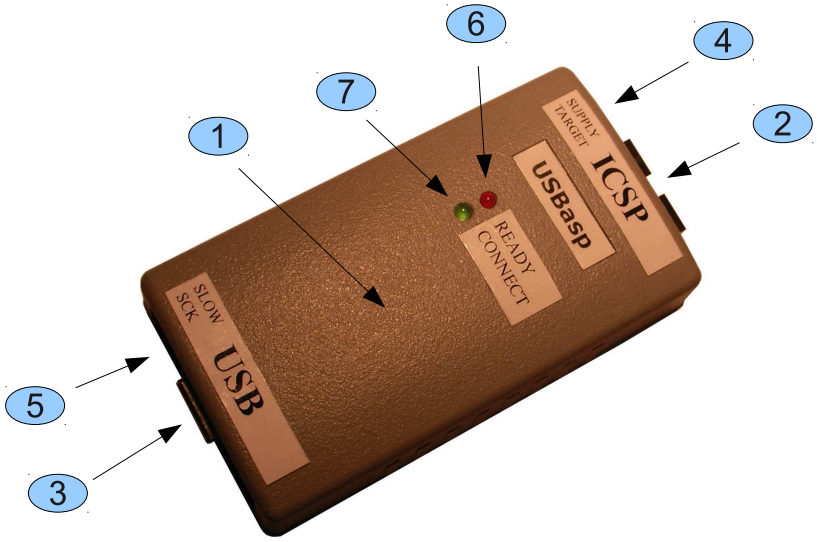
```
apt-get install avrdude avrdude-doc
```

Если источники пакетов правильно настроены, произойдет получение и установка необходимых бинарных пакетов ПО.

Для самостоятельной компиляции `avrdude` и `libusb`, следуйте инструкциям в архивах с исходным кодом.

Подключение программатора

Внешний вид



1. **Разборный корпус**. Корпус необходимо разбирать только для установки 6-ти контактного кабеля ICSP или обновления внтуренней прошивки программатора (см. стр. 5);

2. **Разъем 10-контактного кабеля ICSP** для подключения программатора к программируемой схеме;

3. **Разъем USB**, для подключения к компьютеру;

4. **Джампер подачи питания** на программируемую схему, при его замыкании в разъемах ICSP на линии VCC появляется напряжение +5В;

5. **Джампер понижения скорости** программирования. Установите его перед программированием схем, у которых тактовая частота ниже 1,5 МГц;

6. **Индикатор готовности**. Сообщает, что программатор готов к работе - внутренняя микропрограмма успешно стартовала;

7. **Индикатор входа в режим программирования**. Указывает на то, что программатор установил связь с программируемым микроконтроллером и перевел его в режим программирования. Отключение схемы или программатора в этот момент может привести к отрицательным последствиям (например, не будут восстановлены фьюзбиты или будет прерван цикл программирования).

Порядок подключения

Выберите джампером «SUPPLY TARGET» режим питания программируемой схемы - программирование выполняется надежнее, если схема питается от программатора. При этом надо учитывать предельный потребляемый ток - он *не должен превышать 500 мА*. Если программируемая схема имеет собственное питание, удалите джампер.

Подключите разъем ICSP к программируемой схеме при помощи входящего в комплект кабеля, соблюдая положение первого контакта разъема (провод красного цвета, метка на разъеме IDC).

Внимание! Внутреннее питание прогармматора составляет +5В, поэтому удостоверьтесь, что программируемая схема совместима с сигналами +5В.

Подключите программатор к компьютеру с помощью любого кабеля USB A-B (с квадратным разъемом B-типа). Обмен данными между программатором и компьютером происходит в режиме low-speed bulk transfer, поэтому никаких особенных требований к кабелю по помехозащищенности не предъявляется.

Светящийся красный светодиод «READY» указывает на готовность программатора к работе. При отсутствии сигнала готовности, попробуйте сначала удалить джампер «SUPPLY TARGET», затем - отключить разъем ICSP. Если светодиод после этого загорается, скорее всего, проблема в слишком большом токе потребления программируемой схемы или коротком замыкании в цепи питания.

Использование avrdude

Для программирования микроконтроллеров через USBasp можно пользоваться свободной программой avrdude, через интерфейс командной строки. Существуют также и графические пользовательские интерфейсы[10], но все они в конечном итоге вызывают avrdude с теми или иными параметрами.

[10] Например, http://khazama.com/project/programmer/

Например, с точки зрения `avrdude`, память ATmega168 разделена на несколько типов:

Название	Значение	Размер, байт
flash	Память программ	16384
eeprom	Память EEPROM	512
hfuse	Старший байт HFUSE	1
lfuse	Младший байт LFUSE	1
efuse	Расширенный EFUSE	1
lock	Маска разрешений	1
calibration	Байт калибровки	1
signature	Сигнатура	3

Типичный запуск `avrdude` имеет следующий синтаксис:

```
avrdude -C avrdude.conf -c usbasp -p <тип МК> -U <память:команда:файл:[формат]>
```

где:

`-C avrdude.conf`	путь к файлу конфигурации avrdude
`-c usbasp`	задание типа программатора, в данном случае usbasp
`-p <тип МК>`	тип программируемого МК, например для ATmega168 надо писать `-p m168`
`-U <память:команда:файл:[формат]>`	операция с указанием типа памяти, команды (чтение,запись или верификация) и имени файла. Можно указывать несколько таких ключей, тогда операции будут выполнены последовательно за один запуск `avrdude`.

Внимание! Неправильные значения фьюз-битов могут значительно влиять на работу микроконтроллера. Пожалуйста, внимательно изучите описания производителя для *прошиваемого* микроконтроллера перед попыткой изменения любого fuse-бита!

Текстовый файл конфигурации `avrdude.conf` содержит описания всех поддерживаемых программаторов и микроконтроллеров и может быть легко отредактирован вручную. Вопрос поддержки нового микроконтроллера решается обновлением `avrdude` до последней версии; иногда допустимо просто обновить `avrdude.conf`.

Примеры

Программирование ATmega168

```
avrdude -C avrdude.conf -c usbasp -p m168 -U flash:w:filename.hex
```

Формат файла с прошивкой распознается автоматически, но его можно задать принудительно, через двоеточие после имени файла.

Установка фьюз-битов для ATmega8

```
avrdude -C avrdude.conf -c usbasp -p m8 -U hfuse:w:0xca:m \
        -U lfuse:w:0xdf:m
```

Для записи значений прямо из командной строки надо указывать опцию формата «m».

После установки этих значений, микроконтроллер будет использовать внешний кварц, после старта передавать управление бутлоадеру размером 1K[11], ICSP будет по-прежнему разрешен, встроенный BOD выключен.

Считывание прошивки из ATmega1280

```
avrdude -C avrdude.conf -c usbasp -p m1280 -U flash:r:file.hex
```

Формат нужно задать в командной строке явно, либо использовать имя *существующего* файла: в последнем случае avrdude сначала попробует определить его формат, а затем перепишет содержимое файла считнными из микроконтроллера данными.

Установка битов защиты ATmega328P

```
avrdude -C avrdude.conf -c usbasp -p m328p -U lock:w:0x0f:m
```

Данной командой секция bootloader-а защищается от чтения и записи из секции основной программы микроконтроллера. Это предотвращает случайную модификацию bootloader-а из основной программы микроконтроллера в результате ошибок или сбоев по питанию.

Самопрограммирование

Для обновления прошивки программатора USBasp предусмотрен режим самопрограммирования. Для осуществления этой операции потребуется другой ICSP-программатор.

Разберите корпус программматора, аккуратно вывинтив шурупы, найдите на печатной плате вертикальную перемычку «SELF PGM». Замкните её *до начала программирования.*

Следуйте инструкциям по использованию второго программатора. Обратите внимание, что перемычка «POWER TARGET» в режиме самопрограммирования управляет подачей питания на разъем ICSP, замкните её, если:

[11] Производитель в документации пишет про «512 слов», однако речь идет про 16-битные слова

- второй программатор питается от программируемой схемы (потребуется также запитать сам USBasp, подключив его к USB);
- на время программирования необходимо питать USBasp от второго программатора (в этом случае надо отключить USBasp от USB).

После успешного программирования, удалите перемычку и соберите корпус USBasp.

Отличия от оригинальной схемы

Оригинальная схема, разработанная Томасом Филчем, была изменена следующим образом:

- добавлена 6-контактная вилка ICSP;
- добавлен самовосстанавливающийся предохранитель 0,5А для защиты цепи питания от короткого замыкания.

Для доступа к 6-контактной вилке программирования необходимо вскрыть корпус программатора. Помните, что разъемы ICSP-10 и ICSP-6 соединены параллельно, одновременное подключение к ним двух программируемых схем невозможно.

Внимание! Следите за ориентацией первого контакта кабеля программирования (метка «1» на печатной плате программатора).

Сообщения об ошибках

```
could not find USB device "USBasp" with vid=0x16c0 pid=0x5dc
```

Причина: устройство USBasp не обнаружено на USB-шине компьютера.

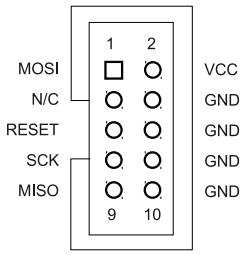
Проврьте, подключен ли USBasp к компьютеру, горит ли светодиод «READY». Попробуйте выключить и включить программатор.

```
avrdude: error: programm enable: target doesn't answer. 1
avrdude: initialization failed, rc=-1
         Double check connections and try again, or use -F to override
         this check.
```

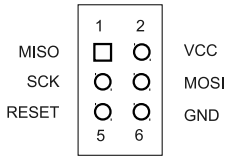
Причина: программируемый микроконтроллер не отвечает на последовательность инициализации режима ICSP.

Проверьте корректность подключения программируемой схемы. Проверьте наличие питания на программируемом микроконтроллере, а также тактовый сигнал. В последнем случае, при подключении ко входу XTAL1 светодиода с нагрузкой, он должен слабо светиться.

Распиновка разъемов ICSP



ICSP-10



ICSP-6

- **MOSI** выход данных, MasterOutSlaveIn
- **VCC** линия питания +5В
- **N/C** свободный
- **GND** земля
- **RESET** линия сброса
- **SCK** синхронизация
- **MISO** вход данных, MasterInSlaveOut

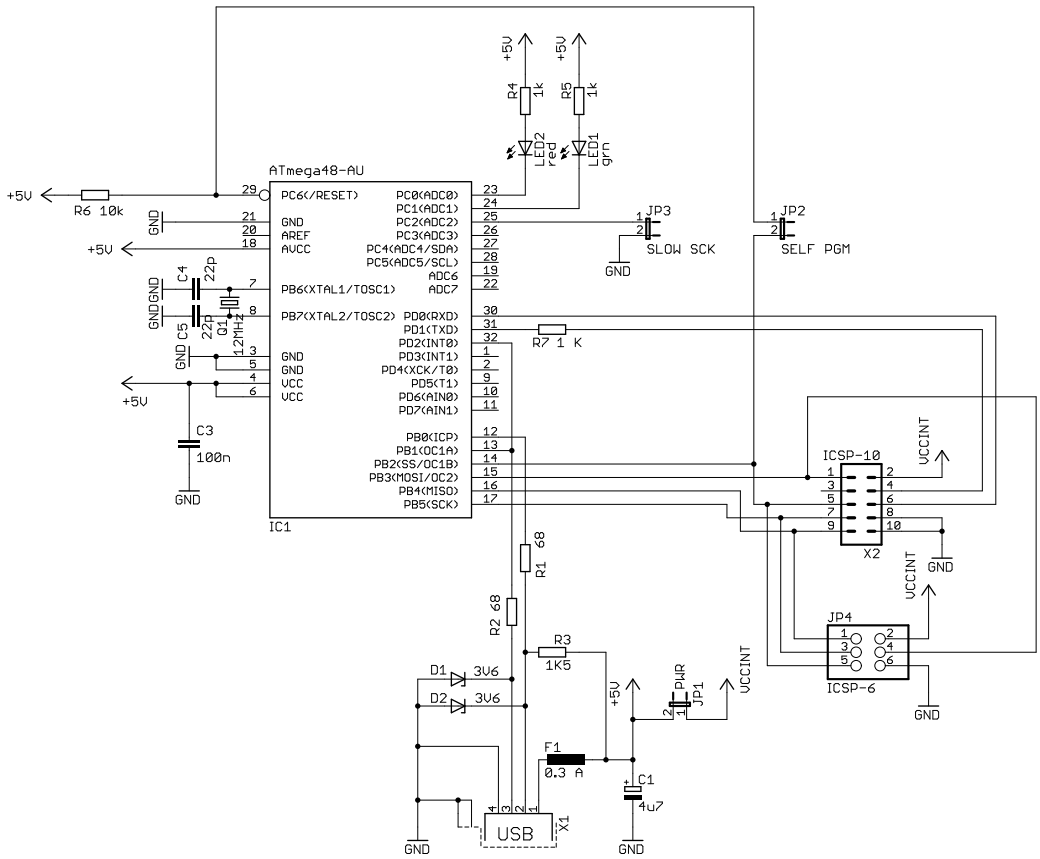



Схема USBasp-id