

Русская документация к AVRDUDE

Часть 1.

Автор перевода Баранов П.В. (KVorb)

<http://kvorb.ru>

Параметры командной строки.

Описание параметров

AVRDUDE это инструмент командной строки, которые используются в следующем виде:

avrdude -p partno options ...

Опции командной строки используются для контроля параметров запуска AVRDUDE.

Используются следующие опции:

-p *partno*

Это единственный обязательный параметр, он указывает AVRDUDE тип микроконтроллера, который подключен к программатору. В *partno* параметром является краткий идентификатор указанный в конфигурационном файле. Укажите *-p ?* чтобы увидеть список возможных идентификаторов перечисленных в конфигурационном файле. Если микроконтроллер неизвестен AVRDUDE, это значит, что его нет в файле конфигурации, но вы можете самостоятельно его туда добавить, если у вас есть *datasheet* на этот микроконтроллер. Сейчас поддерживаются следующие типы микроконтроллеров:

c128	AT90CAN128
pwm2	AT90PWM2
pwm3	AT90PWM3
1200	AT90S1200
2313	AT90S2313
2333	AT90S2333
2343	AT90S2343 (*)
4414	AT90S4414
4433	AT90S4433
4434	AT90S4434
8515	AT90S8515
8535	AT90S8535
m103	ATmega103
m128	ATMega128
m1280	ATmega1280
m1281	ATmega1281
m1284p	ATmega1284P
m128rfa1	ATmega128RFA1
m16	ATmega16
m161	ATmega161
m162	ATmega162
m163	ATmega163
m164	ATmega164
m169	ATmega169
m2560	ATmega2560 (**)
m2561	ATmega2561 (**)
m32	ATmega32
m324	ATmega324
m329	ATmega329
m3290	ATmega3290
m48	ATmega48
m64	ATmega64
m640	ATmega640
m644	ATmega644
m649	ATmega649

m6490	ATmega6490
m8	ATmega8
m8515	ATmega8515
m8535	ATmega8535
m88	ATmega88
t12	ATtiny12
t13	ATtiny13
t15	ATtiny15
t2313	ATtiny2313
t25	ATtiny25
t26	ATtiny26
t45	ATtiny45
t85	ATtiny85
x128a1	ATXmega128A1
x128a1d	ATXmega128A1revD

(*) В AT90S2313 и ATtiny22 используется один и тот же алгоритм.

(**) Flash адреса выше 128 Кбайт не поддерживаются всеми аппаратными программаторами. Для работы с ними используйте jtag2, stk500v2 и bit-bang программаторы.

-b *baudrate*

Переопределяет скорость соединения по RS-232, указанную для программатора в файле конфигурации.

-B *bitclock*

Указывает bit clock period для интерфейса JTAG или ISP clock (только для JTAG ICE). Значением – число микросекунд. По умолчанию в JTAG ICE используется bit clock period равный 1 микросекунде, подходящий для микроконтроллеров работающих на частоте 4 МГц и выше.

-c *programmer-id*

Указывает программатор, который будет использоваться. AVRDUDE известны несколько распространенных программаторов. Используйте эту опцию, чтобы указать, какой из них использовать. Идентификаторы известных программаторов перечислены в конфигурационном файле. Укажите -c ? чтобы увидеть список этих программаторов. Если у вас есть программатор, который неизвестен AVRDUDE и он управляется с помощью компьютера через параллельный порт, есть хороший шанс, что его можно будет легко добавить в файл конфигурации без каких-либо изменений в коде AVRDUDE. Просто скопировать существующий код и изменить номера контактов в соответствии со схемой подключения вашего программатора.

-C *config-file*

Использовать указанный файл конфигурации. По умолчанию AVRDUDE читает конфигурацию из файла /usr/local/etc/avrdude.conf (FreeBSD and Linux).

-D

Отключить автоматическое стирание flash-памяти. Автоматическое стирание flash не используется для микроконтроллеров ATmega.

-e

Стирает микроконтроллер. При этом в flash ROM и EEPROM записывается значение '0xFF', и очищаются биты защиты. Эта опция не распространяется на микроконтроллеры ATmega, в которых используется постраничная запись.

-E *exitspec*[,...]

По умолчанию, AVRDUDE после программирования возвращает параллельный порт в то же состояние, которое было до начала работы. Эта опция изменяет состояние линий RESET и Vcc параллельного порта в соответствии с аргументами:

reset

На линии RESET будет оставлен низкий уровень, чтобы оставить микроконтроллер в состоянии сброса.

noreset

На линии RESET будет установлен высокий уровень, чтобы запустить микроконтроллер после программирования.

vcc

Эта опция оставляет высокий уровень на линии "Vcc" которая может использоваться для питания микроконтроллера.

novcc

Эта опция устанавливает низкий уровень на линии "Vcc" после программирования.

Можно использовать несколько аргументов разделяя их запятыми.

-F

По умолчанию перед программированием AVRDUDE проверяет сигнатуру микроконтроллера. Иногда сигнатура микроконтроллера повреждается (стирается или перезаписывается), но при этом микроконтроллер продолжает исправно функционировать. Эта опция позволяет отключить проверку сигнатуры.

-i *delay*

Для bitbang программаторов, делает паузу в микросекундах перед каждой отправкой кадого бита. Если компьютер слишком быстрый или микроконтроллер работает на слишком низкой частоте (от кварца 32 кГц или внутреннего RC генератора на 128 кГц), то эта опция позволяет соблюсти условие по которому частота ISP не должна превышать 1/4 частоты процессора.

-n

Запрещает производить запись данных в микроконтроллер (используется для отладки AVRDUDE).

-O

Эта опция калибрует RC генератор в соответствии с апнотой AVR503. Это поддерживается только на программаторах STK500v2, AVRISP mkII, and JTAG ICE mkII. При этом результат будет храниться в ячейке EEPROM по адресу 0.

-P *port*

Указывает к какому порту подключен программатор.

-q

Отключает отображение прогресс бара при чтении или записи в микроконтроллер. Указывается дважды, чтобы выводить на экран еще меньше информации.

-u

Отключает чтение и сравнение ячеек конфигурации до и после программирования. Этот параметр надо указывать если вы хотите изменить ячейки конфигурации, иначе AVRDUDE изменит их на прочитанные до программирования, для вашей безопасности.

-t

Включает в AVRDUDE терминальный режим.

-U memtype:op:filename[:format]

Указывает тип памяти:

calibration Один или несколько байт калибровки RC генератора.

eeprom EEPROM память микроконтроллера.

efuse Дополнительный конфигурационный байт

flash FLASH память микроконтроллера.

fuse Конфигурационный байт для микроконтроллеров с одним fuse-байтом.

hfuse Старший конфигурационный байт.

lfuse Младший конфигурационный байт.

lock Ячейки защиты.

signature Идентификатор микроконтроллера.

Поле *op* определяет какая операция будет выполнена:

r чтение памяти из микроконтроллера и запись в файл

w чтение прошивки из файла и запись в память микроконтроллера

v чтение памяти из микроконтроллера и сравнение её с прошивкой

Поле *filename* указывает в какой файл писать или читать данные. Поле *format* является необязательным и содержит формат файла. Возможные значения:

i Intel Hex

s Motorola S-record

r Raw-формат

m

Фактические значения байтов для записи указываются в командной строке. Байты помещаются в поле *filename* и разделяются запятыми или пробелами. Этот режим применяется за записи конфигурационных ячеек без создания специального файла или переключения в терминальный режим. Если байт указан в формате 0x, то он будет записан в шестнадцатеричном значении. Если перед байтом стоит 0, то будет записано восьмеричное число. По умолчанию байты записываются в десятичной системе.

a Авто определение формата.

d Десятичный формат. Числа разделяются запятыми.

h Шестнадцатеричный формат. Перед числами ставится 0x.

o Восьмеричный формат. Перед числами ставится ноль.

b Двоичный формат. Перед числами ставится 0b.

По умолчанию (если не указан параметр) формат файла для записи в микроконтроллер определяется автоматически, а данные из микроконтроллера читаются в сыром двоичном формате.

-v

Расширенный вывод информации.

-V

Отключает автоматическую проверку записанной информации.

-x *extended_param*

Дополнительные параметры.

-y

Указывает AVRDUDE использовать последние четыре байта EEPROM памяти чтобы отслеживать сколько раз этот микроконтроллер был стерт. Если этот параметр указывается совместно с -e, то перед стиранием микроконтроллера счетчик считывается, увеличивается и записывается обратно после стирания.

-Y *cycles*

Указанное в параметре число будет записано в EEPROM как значение счетчика циклов стирания-записи.

Примеры использования AVRDUDE.

Запись файла `diag.hex` в микроконтроллер ATmega128 используя программатор STK500:

```
% avrdude -p m128 -c stk500 -e -U flash:w:diag.hex
```

```
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ##### | 100% 0.03s
```

```
avrdude: Device signature = 0x1e9702
```

```
avrdude: erasing chip
```

```
avrdude: done.
```

```
avrdude: performing op: 1, flash, 0, diag.hex
```

```
avrdude: reading input file "diag.hex"
```

```
avrdude: input file diag.hex auto detected as Intel Hex
```

```
avrdude: writing flash (19278 bytes):
```

```
Writing | ##### | 100% 7.60s
```

```
avrdude: 19456 bytes of flash written
```

```
avrdude: verifying flash memory against diag.hex:
```

```
avrdude: load data flash data from input file diag.hex:
```

```
avrdude: input file diag.hex auto detected as Intel Hex
```

```
avrdude: input file diag.hex contains 19278 bytes
```

```
avrdude: reading on-chip flash data:
```

```
Reading | ##### | 100% 6.83s
```

```
avrdude: verifying ...
```

```
avrdude: 19278 bytes of flash verified
```

```
avrdude: safemode: Fuses OK
```

```
avrdude done. Thank you.
```

Считывание flash памяти из микроконтроллера ATmega128 с помощью программатора STK500 и сохранение данных в бинарный файл c:/diagflash.bin:

```
% avrdude -p m128 -c stk500 -U flash:r:"c:/diag flash.bin":r
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.03s

avrdude: Device signature = 0x1e9702
avrdude: reading flash memory:

Reading | ##### | 100% 46.10s

avrdude: writing output file "c:/diag flash.bin"

avrdude: safemode: Fuses OK

avrdude done. Thank you.
```

Подключение JTAG ICE mkII программатора к USB и включение терминального режима:

```
% avrdude -c jtag2 -p m649 -P usb:1c:37 -t
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.03s

avrdude: Device signature = 0x1e9603

[ ... terminal mode output skipped for brevity ... ]

avrdude done. Thank you.
```

Используя программатор по умолчанию, записать файл diag.hex в flash память, файл eeprom.hex в EEPROM память. Программирование расширенной, старшей и младшей конфигурационных ячеек в 0xFF, 0x89 и 0x2E соответственно:

```
% avrdude -p m128 -u -U flash:w:diag.hex \  
> -U eeprom:w:eeprom.hex \  
> -U efuse:w:0xff:m \  
> -U hfuse:w:0x89:m \  
> -U lfuse:w:0x2e:m
```

```
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ##### | 100% 0.03s
```

```
avrdude: Device signature = 0x1e9702  
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed  
To disable this feature, specify the -D option.  
avrdude: erasing chip  
avrdude: reading input file "diag.hex"  
avrdude: input file diag.hex auto detected as Intel Hex  
avrdude: writing flash (19278 bytes):
```

```
Writing | ##### | 100% 7.60s
```

```
avrdude: 19456 bytes of flash written  
avrdude: verifying flash memory against diag.hex:  
avrdude: load data flash data from input file diag.hex:  
avrdude: input file diag.hex auto detected as Intel Hex  
avrdude: input file diag.hex contains 19278 bytes  
avrdude: reading on-chip flash data:
```

```
Reading | ##### | 100% 6.84s
```

```
avrdude: verifying ...  
avrdude: 19278 bytes of flash verified
```

```
[ ... other memory status output skipped for brevity ... ]
```

```
avrdude done. Thank you.
```