

ISP (ICSP) программатор из Arduino/Freduino

Любая версия Freeduino / Arduino может выступать в роли ISP программатора, с помощью которого можно, например, изменить bootloader в другой аналогичной плате, или прошить bootloader в новую микросхему ATmega. В описании указывается Arduino, но фактически может быть использован любой клон, включая все варианты Freeduino.

Здесь рассматривается два варианта ISP программатора на основе Arduino: Mega-ISP, который может быть собран из любой версии Arduino и его клонов, и Bit-Bang программатор, для которого подойдут только те варианты Arduino, которые имеют дополнительно выведенные на специальный разъем контакты микросхемы FT232RL.

Особенности ISP/ICSP

Аббревиатуры ISP и ICSP означают In System Programming и In Circuit Serial Programming соответственно. Это означает программирование чипа уже подключенного в некоторую схему, программирование в готовом устройстве по последовательному протоколу.

Принципиально важным является то, что программируемый микроконтроллер должен успешно "завестись", и только после этого он будет в состоянии принимать данные от программатора. Для нас это означает, что он должен быть подключен к питанию, и иметь соответствующий источник тактовых сигналов. Питание проще всего подавать через сам разъем ICSP, а вот с источником тактовых сигналов есть некоторые особенности.

Источник тактовых сигналов выбирается в микроконтроллерах серии ATmega с помощью так называемых fuse-битов, которые, также как и память программ, и EEPROM доступны для изменения с помощью программатора.

Производитель перед продажей выставляет fuse-биты так, что в качестве источника тактовых сигналов выбран внутренний осциллятор, и такой микроконтроллер можно просто подключить к ISP программатору с учетом расположения его выводов, и начать работу. Однако, если с помощью ISP программатора изменить значения fuse-битов так, что изменится источник тактовых сигналов, то чтобы ISP программатор опять начал работать с микроконтроллером, придется соответствующий источник к нему подключить. Таким образом, нужно быть внимательным при изменении значений fuse-битов.



Mega-ISP

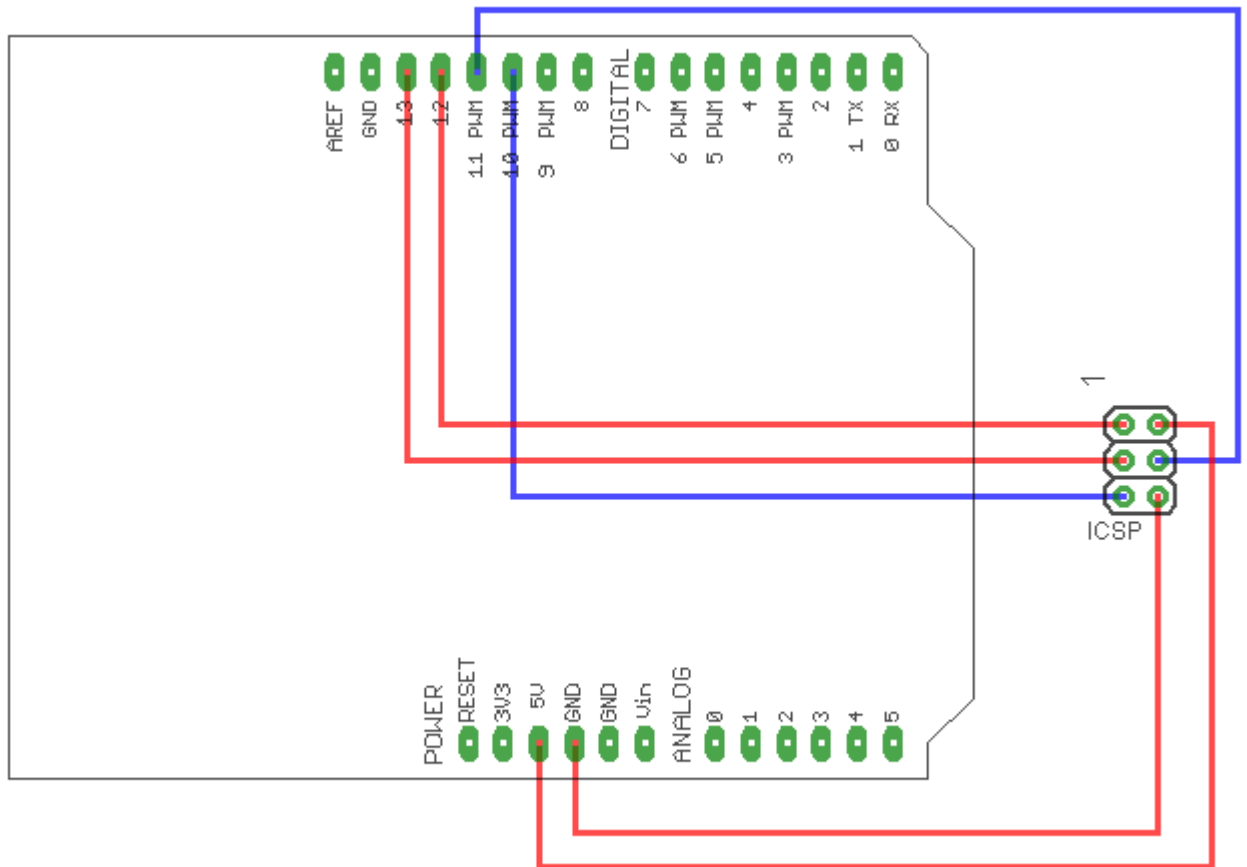
Из известных проектов по созданию ISP программатора на основе Arduino наиболее интересным нам кажется [Mega-ISP](#). К сожалению проект не имеет нормальной документации, поэтому здесь мы опишем его наиболее важные особенности.

Во-первых, после загрузки .pde проекта в вашу плату Arduino, скорее всего потребуется тем или иным образом отключить программный сброс. Владельцы Freeduino 2009 могут просто снять перемычку JRS, владельцы Arduino Duemilanove могут перерезать ножом ту же перемычку. Для Freeduino Through-Hole, Arduino Diecimila, Freeduino MaxSerial (и Arduino Duemilanove) можно выпаять конденсатор CRS, но мы советуем менее радикальный способ - соединить выводы Reset и +5V резистором номиналом порядка 120 Ом ([оригинал здесь](#)). Если все сделано правильно, при открытии COM-порта не будет происходить сброс микроконтроллера (это видно по светодиоду 13), и как побочный эффект станет невозможно обновить его собственную прошивку из среды Arduino.

Во-вторых, данный ISP программатор при общении с ПК использует протокол AVRISP/STK500v1 только со скоростью 19200 бит/сек, и нормально работает не со всеми вариантами ПО для ПК. Поэтому не удивляйтесь, что не удастся что-либо прошить из среды Arduino. Авторы рекомендуют использовать программу avrdude из состава Win-AVR.

Сборка программатора Mega-ISP

Если Вы с помощью одной платы Arduino собираетесь прошить другую, то Вам потребуется разъем 2x3 и 6 проводков. Схема сборки:

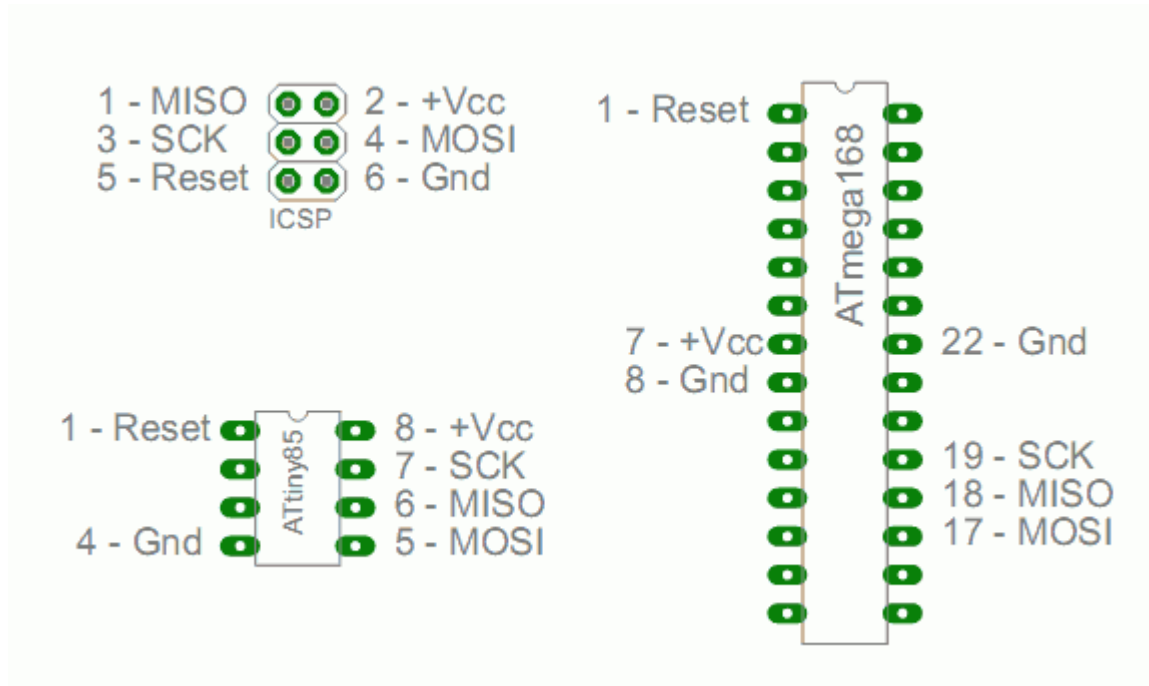


Если нужно прошить отдельный микроконтроллер (не забываем, что его fuse-биты должны быть выставлены так, что он тактируется от внутреннего осцилятора), то нужно узнать согласно [документации Atmel](#) расположение выводов Gnd, Vcc, MOSI, MISO, SCK.

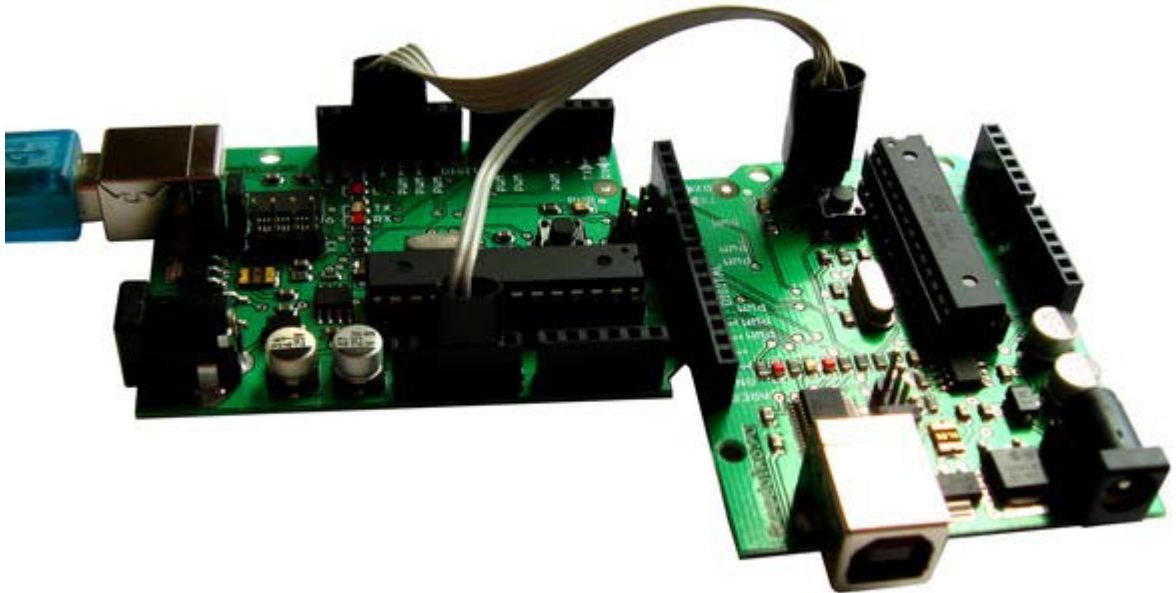
Соединение с Arduino такое: 10 - Reset, 11 - MOSI, 12 - MISO, 13 - SCK.

Не забываем подключить "землю" и питание.

Для примера приводим разводку для разъема ICSP и микроконтроллеров ATmega168 и ATtiny85.



На фото представлен один из вариантов реализации:



Работа с программатором Mega-ISP

Рассмотрим например, как прошить новый boot-loader.

Скачайте [архив с .pde проектом и подходящей версией avrdude](#). Распакуйте архив в какую-нибудь папку.

Откройте .pde проект в среде Arduino и загрузите его в вашу плату. После этого можно отключать программный сброс согласно описания выше.

Подсоедините вторую плату (которую будем "прошивать") к первой.

В режиме командной строки перейдите в папку с avrdude. Теперь можно запускать avrdude с соответствующими ключами:

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\User>cd C:\Arduino-Mega-ISP

C:\Arduino-Mega-ISP>avrdude -q -C avrdude.conf -p m168 -c avrisp -P COM3 -b
0 -e -U flash:w:ATmegaBOOT_168_diecimila.hex_
```

Разберем ключи:

- q - уменьшает объем выводимой на экран отладочной информации
- C avrdude.conf - указывает расположение .conf файла (в нашем случае он в той же папке)
- p m168 - выбор типа микроконтроллера ATmega168
- c avrisp - выбор протокола
- P COM3 - выбор COM порта. Здесь нужно поставить свой
- b 19200 - выбор скорости порта
- e - Команда на предварительную очистку памяти программ
- U flash:w:ATmegaBOOT_168_diecimila.hex - указывает на .hex файл с прошивкой

Прочие замечания

В процессе работы с программатором было замечено, что иногда процесс прошивки не начинается, и avrdude завершается с ошибкой. В таких случаях нам помогал сброс программатора кнопкой Reset и повторная прошивка.

Архив с ПО также содержит файл sample.bat, в котором кроме bootloader'a программируются fuse и lock биты.

В Сети есть страничка с разведенной платой такого программатора, включающей дополнительные отладочные светодиоды: <http://drug123.org.ua/mega-isp-shield/>



Bit-Bang программатор

"Сердцем" Bit-Bang программатора является микросхема FD232RL (та самая, которая реализует COM-порт на шине USB), а собственно микроконтроллер ATmega никакого участия в процессе

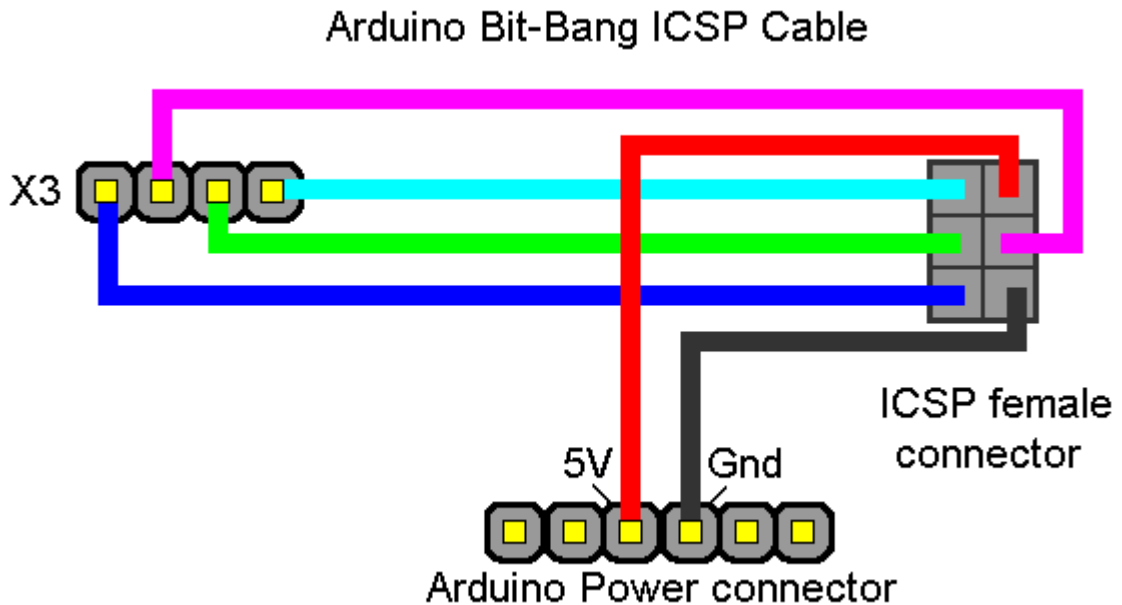
программирования не принимает. Это позволяет программировать микроконтроллер, установленный в "панельке" Вашей Arduino – вторая плата Arduino не потребуется.

Соответственно, для данного программатора подойдут только USB версии Arduino, да и то не все, т.к. не на всех вариантах разведен разъем X3 с дополнительными выводами FT232RL.

Сборка программатора Arduino Bit-Bang

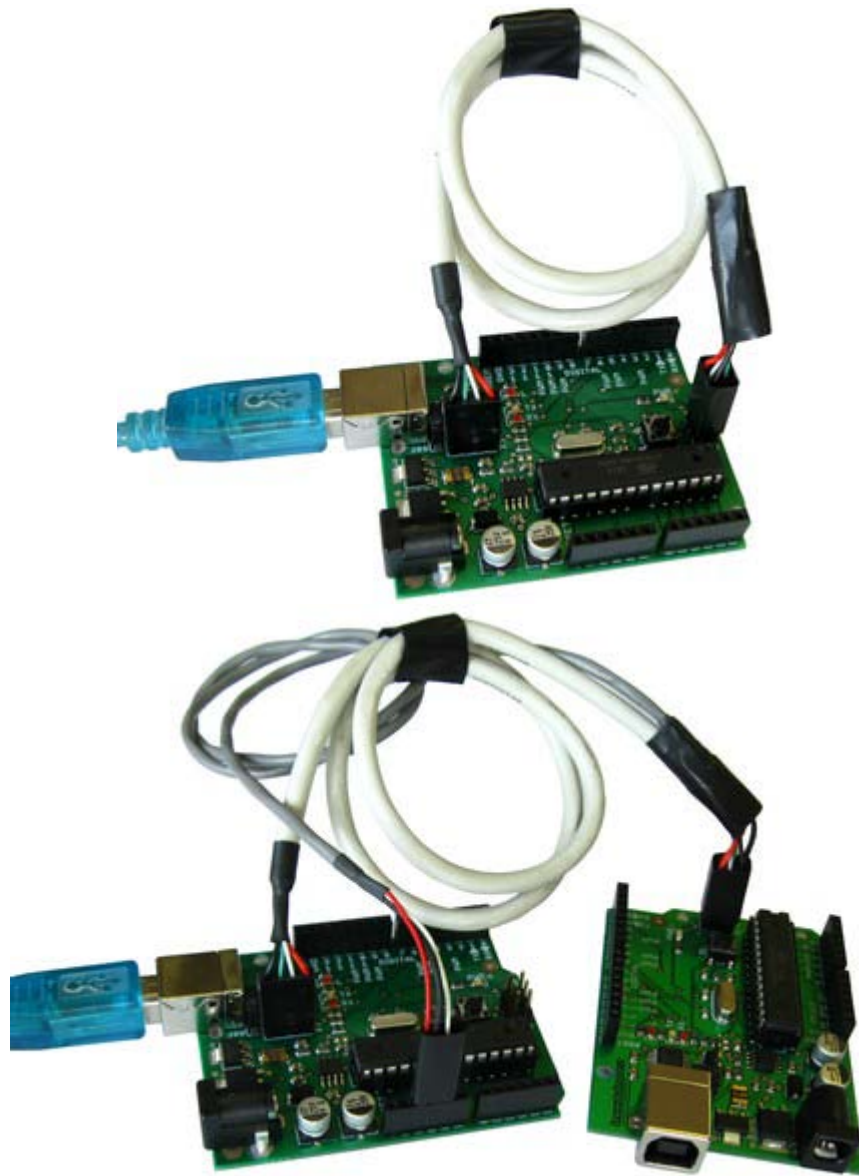
Для сборки Вам потребуются разъемы 1x4, 2x6, штекер 1x2 (для питания) и несколько проводов. Можно использовать, например пару Audio-кабелей от CD-ROM – у них удобные разборные разъемы, и в таком случае можно обойтись даже без паяльника.

Схема соединений следующая:



Провода питания и "земли" потребуются, если Вы собираетесь программировать внешнее устройство. Для программирования микроконтроллера, находящегося в панели той же самой Arduino они не нужны.

На фотографиях представлены оба варианта соединения – программирование микроконтроллера ATmega168, находящегося в панельке Arduino (провода питания и "земли" не нужны), и программирование микроконтроллера во второй плате.



Работа с программатором Arduino Bit-Bang

Аналогично рассмотрим прошивку нового boot-loader'a.

Скачайте [архив с соответствующей версией avrdude](#). Распакуйте архив в какую-нибудь папку.

Подсоедините программатор к USB порту.

В режиме командной строки перейдите в папку с avrdude. Теперь можно запускать avrdude с соответствующими ключами:

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\User>cd c:\Arduino-Bit-Bang-ISP
C:\Arduino-Bit-Bang-ISP>avrdude -q -C avrdude.conf -p m168 -c diecimila -P
e -U flash:w:ATmegaBOOT_168_diecimila.hex
```

Разберем ключи:

- q - уменьшает объем выводимой на экран отладочной информации
- C avrdude.conf - указывает расположение .conf файла (в нашем случае он в той же папке)
- p m168 - выбор типа микроконтроллера ATmega168
- c diecimila - выбор программатора (он описан под таким именем в .conf файле)
- P ft0 - выбор порта. Выбрано первое устройство с чипом FTDI
- e - Команда на предварительную очистку памяти программ

-U flash:w:ATmegaBOOT_168_diecimila.hex - указывает на .hex файл с прошивкой

Прочие замечания

Архив с ПО также содержит файл sample.bat, в котором кроме bootloader'a программируются fuse и lock биты.

Более полную информацию можно найти по ссылкам:

http://www.geocities.jp/arduino_diecimila/bootloader/index_en.html

http://www.geocities.jp/arduino_diecimila/bootloader/index_old_en.html