MightyOhm

Join the resistance.


Usage Instructions

To use the HV Rescue Shield 2:

Step 0:

Verify your Arduino is working and your IDE is properly configured. A simple blinking LED circuit is good for this.

Step 1:

Download and load the HVRescue_Shield sketch and install it in your Arduino working folder. Open the Arduino IDE and click the "Upload" button. On older Arduinos, you'll have to manually press the reset button on the Arduino first. Note that if you have the Rescue Shield installed on the Arduino and a target AVR in one of the sockets, the upload will most likely fail. Make sure the sockets are empty before uploading.
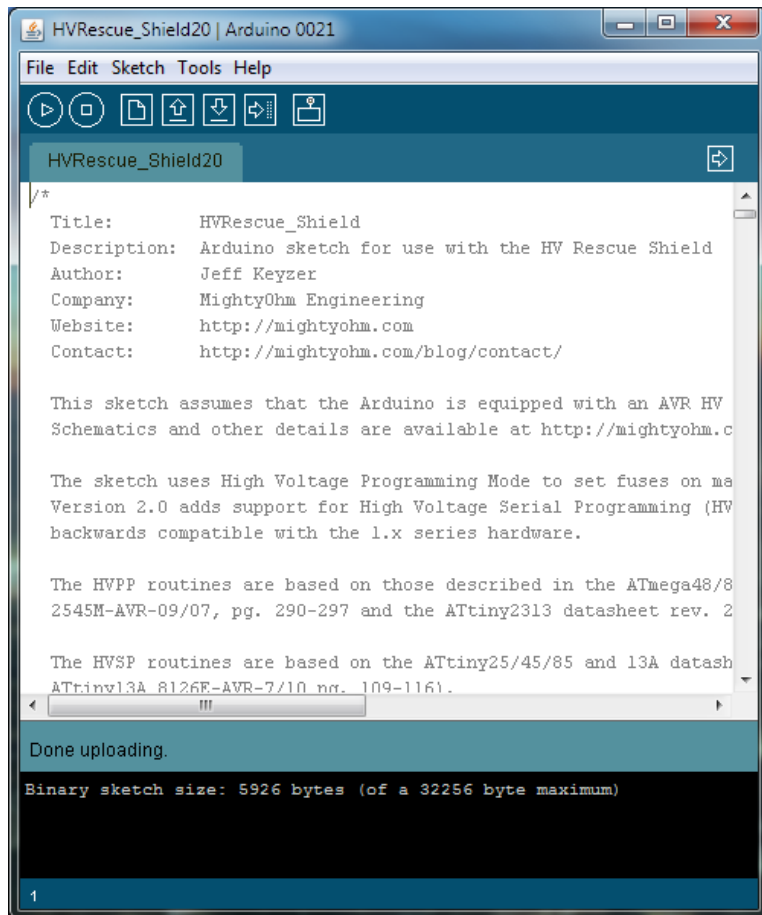

Step 2:

Install the Rescue Shield on the Arduino. **Make sure there is no target AVR installed in either socket at this point.** The red and green LEDs will glow briefly as the Arduino finishes its reset, but eventually the green READY LED should turn on.

Step 3:

Click the "Serial Monitor" button at the top of the Arduino IDE window. It's the rightmost button in the toolbar, a rectangle with an antenna sticking out the top.

If interactive mode is enabled in the sketch, after a moment you should see a prompt to select a programming mode, as shown here:
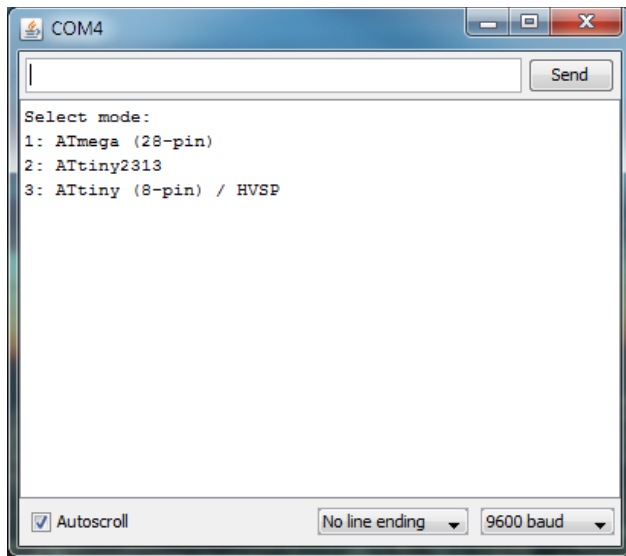
The process shown here is the same for all programming modes. If you want to program the fuses on an ATtiny2313, select mode 2. If you want to program an 8-pin ATtiny, select mode 3 (HVSP mode).

Step 4:

Select a mode by entering the corresponding number. For example, here I have selected mode 1, ATmega mode.
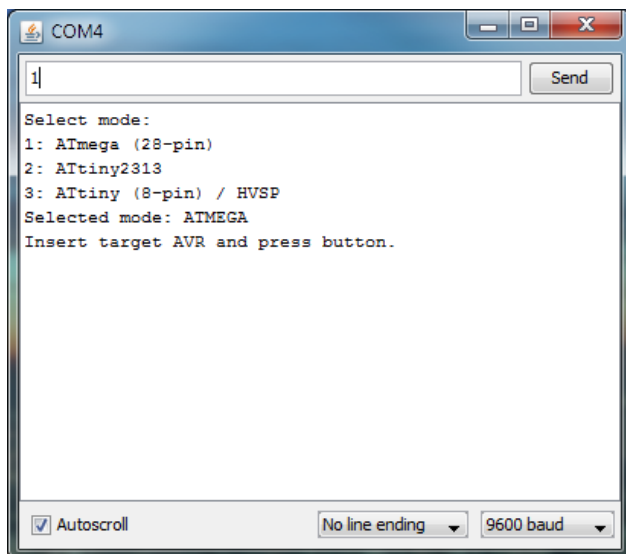

Step 5:

Insert the target AVR into the corresponding socket. IC1 is for the 28-pin ATmega parts, IC2 is for the ATtiny2313, IC4 is for 8-pin ATtiny parts.

Step 6:

Push the button on the Rescue Shield, it's labeled GO!

The red BURN LED should turn on and the Rescue Shield will report the fuse settings currently stored in the target AVR. Then you will be prompted for new fuse values to send to the target.
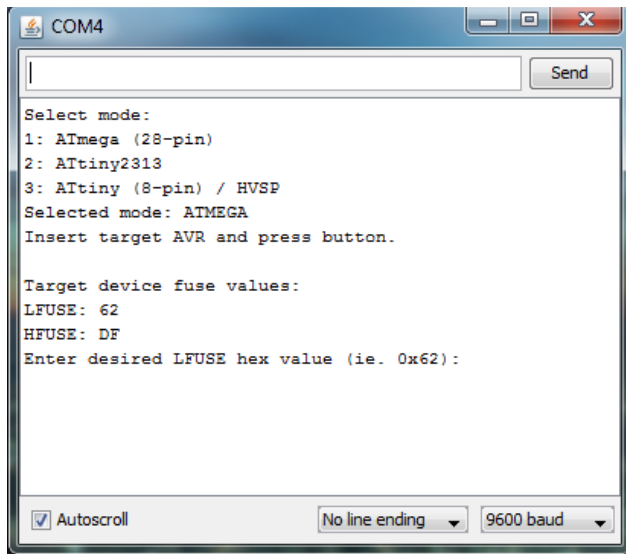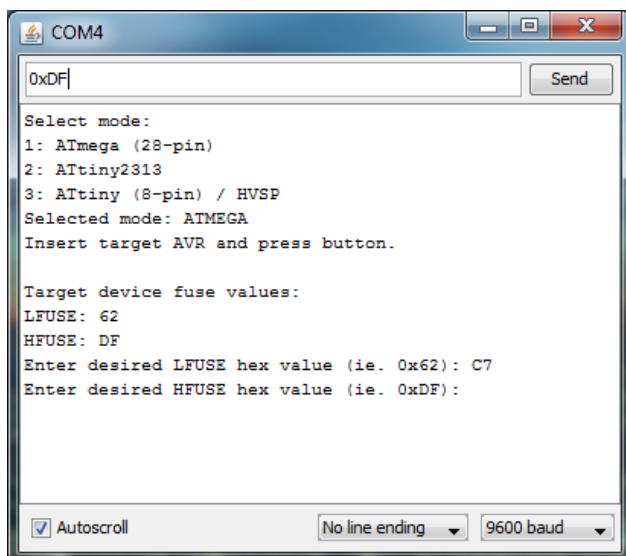
Step 7:

Enter the fuse values, in the form "0xFF". The leading "0x" is important (the fuse value parsing code is very simple). The default values for an ATmega168 are given as a suggestion.

Enter the fuse values into the box above the serial window, and click "Send". In this example, I am entering an LFUSE value of 0xC7.
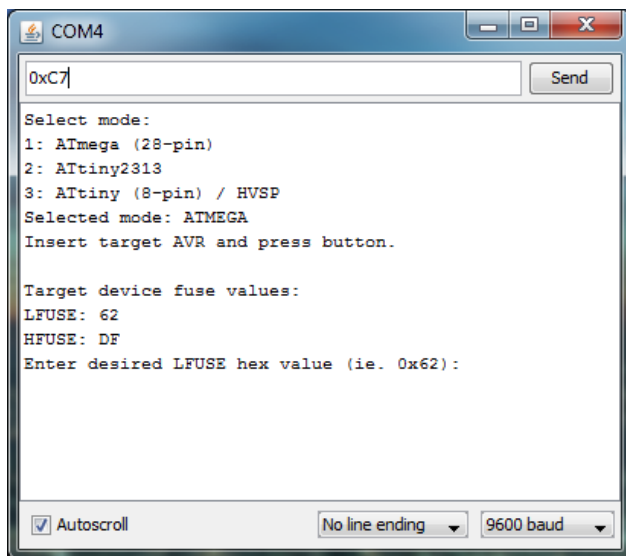
Using the same procedure, enter the HFUSE value:

In this example, I entered 0xDF, the default HFUSE value for the ATmega168.

If EFUSE support is enabled, you will need to enter a desired EFUSE value as well (not shown).

Step 7:

The sketch will burn the desired fuses to the part, and then read the fuses back again to verify. When the process is done, you will see the message "Burn complete." as shown below:
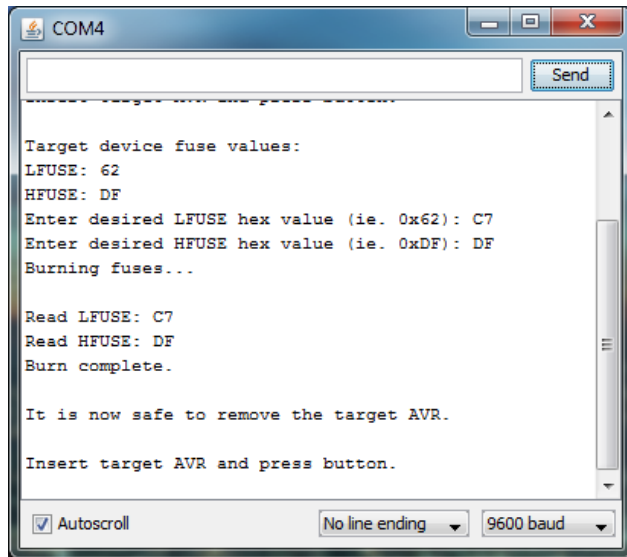
If the fuses read back match the ones you entered, you're done!

The red BURN LED will turn off when the burn process is finished, and the green READY LED will come back on.

Once the BURN LED is out, remove the target AVR. You're done!

Notes:

- If you want to change modes, remove the target AVR, close the serial monitor window, and click the Serial Monitor button to open it again. The Arduino will reset and and the mode selection prompt should appear within a couple moments.
- If you see garbled serial data during the mode selection, it's probably because there is an AVR in one of the Rescue Shield sockets. Remove the AVR and try again.

Options:

Options are listed immediately after the title block of the sketch. If you change options, you must recompile the sketch and upload it to the Arduino. Be sure to remove the Rescue Shield before uploading sketches to the Arduino.

```
// User defined settings
#define  DEFAULTMODE  HVSP // If running in non-interactive mode, you need to set this to ATMEGA, TINY2313, or HVSP.
#define  ASKMODE      1       // Set this to 1 to enable mode question at startup
#define  INTERACTIVE  1       // Set this to 0 to disable interactive (serial) mode
#define  BURN_EFUSE   0       // Set this to 1 to enable burning extended fuse byte
#define  BAUD         9600    // Serial port rate at which to talk to PC

// If interactive mode is off, these fuse settings are used instead of user prompted values
#define  LFUSE        0x62  // default for ATmega168 = 0x62
#define  HFUSE        0xDF  // default for ATmega168 = 0xDF
#define  EFUSE        0xF9  // default for ATmega168 = 0xF9
```

- To disable interactive mode, set INTERACTIVE to 0. You will need to set the LFUSE, HFUSE, and EFUSE (if desired) values below to the desired fuse values.
- If you want to burn the EFUSE byte in addition to LFUSE and HFUSE, set the BURN_EFUSE option to 1.

This entry was posted on Monday, December 13th, 2010 at 1:48 pm. You can follow any responses to this entry through the RSS 2.0 feed. Both comments and pings are currently closed.

Comments are closed.

www.**flickr**.com

- Search for:
  Search
-

Pages

- [Home](#)
- [About](#)
- [Archives](#)
- [Contact](#)
- [My Book](#)
- [Products](#)
  - [AVR HV Rescue Shield (Obsolete)](#)
  - [Geiger Counter](#)
  - [HV Rescue Shield 2](#)
    - [Assembly Instructions](#)
    - [Design Files](#)
    - [Source Code](#)
    - [Support](#)
    - [Usage Instructions](#)
  - [ICSPOV](#)
- [Support](#)
- [Tutorials](#)

Jeff says…

Have you seen the [forums?](#)

Categories

- [Amateur Radio](#)
- [Announcements](#)
- [Books and Resources](#)
- [DIY](#)
- [Electronics](#)
- [Embedded Linux](#)
- [Events](#)
- [Microcontrollers](#)
- [Uncategorized](#)

Recent Posts

- [Jootmine On Imelihtne (Soldering is Easy: Estonian Translation)](#)
- [And we're back!](#)
- [Thinking about buying a Geiger Kit? Order now or wait 2 weeks!](#)
- [Geiger Counter Kits are (finally) in stock!](#)
- [See you at the Open Hardware Summit & NY Maker Faire!](#)

Recent Comments

- [Alexander Shabarshin](#) on [DIY Integrated Circuit Design with MOSIS](#)
- Rafi on [I was not prepared for this.](#)
- [Jeff](#) on [Geiger Kit Demo](#)
- Nick on [Geiger Kit Demo](#)
- Kevin Groce on [Video: Staver Electromechanical Vane Display](#)

-

    - [Atmega 168 only?](#) *petr farnik*
    - [Re: Looking for feedback...](#) *jox*
    - [Re: alternatives to the WL-520gU](#) *gerben*
    - [Re: Serial processing code? Patchube?](#) *peterloron*
    - [Re: Why the preference of the SBM-20?](#) *mightyohm*

- ## Electronics Blogs

    - [Adafruit Industries](#)
    - [Chris Gammell's Analog Life](#)
    - [Circuits@Home](#)
    - [DadHacker](#)
    - [EEVblog](#)
    - [EEWeb](#)
    - [etotheipiplusone](#)
    - [Evil Mad Scientist Labs](#)
    - [Hack a Day](#)
    - [Interactive Matter](#)
    - [JeeLabs](#)
    - [Keith's Electronics Blog](#)
    - [Make:](#)
    - [My 2μF](#)
    - [The Amp Hour](#)
    - [The Smell of Molten Projects in the Morning](#)
    - [Tinkerlog](#)

- ## Project Index

    - [Arduino-based AVR High Voltage Programmer](#)
    - [Bluetooth Handset](#)
    - [DIY Light Tent](#)
    - [DIY PID Controlled Soldering Hotplate](#)
    - [PIC RGB Video Generation](#)
    - [PID Controlled Solder Paste Fridge](#)
    - [Wifi Radio Project](#)

- ## The MightyOhm Universe

    - [Discussion Forums](#)
    - [Facebook](#)
    - [Flickr Photostream](#)
    - [MightyOhm Wiki](#)
    - [PCB Flickr Pool](#)
    - [Twitter](#)

- ## Vendors

    - [Mouser Electronics](#)
    - [Sovtube.com](#)

-

    I support the [Open Source Hardware Definition v1.0](#)

- ## [@mightyohm's Tweets](#)

    - Sporting my new Solder:Time watch kit from SpikenzieLabs: http://t.co/C8CA63xM [2 hours ago](#)
    - Pic. http://t.co/C3xS32bP [2 hours ago](#)
    - So, what is this green stretchy fabric thing I got from @[seeedstudio](#) at @[ohsummit](#) supposed to be? A laptop cozy? [2 hours ago](#)
    - Argh, stupid TweetDeck! Anyway, my mini ITX Atom server is still going strong. Pics of the build last year: http://t.co/DUQ01Ebe [3 hours ago](#)
    - I can't believe it's been a year since I built my mini-ITX Intel Atom server, it's still the perfect machin… (cont) http://t.co/bl0yzpTf

- # Meta

  - Log in
  - Entries RSS
  - Comments RSS
  - WordPress.org

© 2008-2011 **Jeff Keyzer / MightyOhm Engineering**
Entries (RSS) and Comments (RSS).

☺