SparkFun Electronics Reinstalling your Arduino bootloader

by M-Short | January 18, 2011 | *27 comments* Skill Level: ☆ Beginner

How to reprogram the bootloader on your Arduino Uno

What you need (Option 1):

- Your Arduino Uno (to program)
- A Working Arduino
- Some Jumper Cables

What you need (Option 2):

- Your Arduino Uno (to program)
- An AVR Programmer such as the AVR Pocket Programmer
- An AVR Programming Cable (the pocket programmer comes with one)

What you need (software)

- The Arduino Optiboot Bootloader (Pro Mega and Pro Mega Mini 3.3V Bootloader)
- WinAVR (Windows) or Crosspack for AVR Development (Mac) - This software contains avrdude, which is needed! (Note: You could get around downloading this and using the copy of avrdude that comes with Arduino; however installing the version with WinAVR will add the path to the Windows Environment Variables which makes it easier to use in the command prompt)
- Arduino 0018 (or newer) if you are using option 1

Programming the Arduino to act as a Programmer (for Option 1)

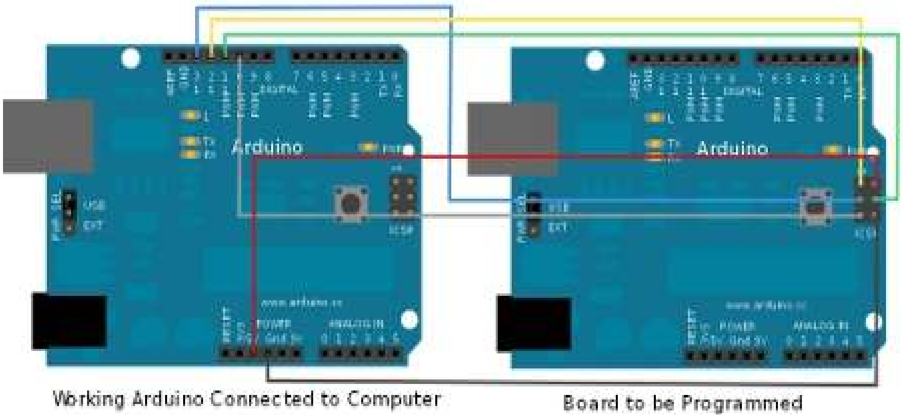
First we have to get the Arduino programmed to work as a programmer! This is very simple; with the Arduino IDE open, open the ArduinoISP sketch by going to File->Examples->ArduinoISP. This sketch will make the Arduino act as an ISP programmer. Select your board in the Tools menu, and make sure the correct COM port is selected (also, copy down or make a mental note of the COM port. It will be needed in the next step). Then just upload the sketch to the Arduino.

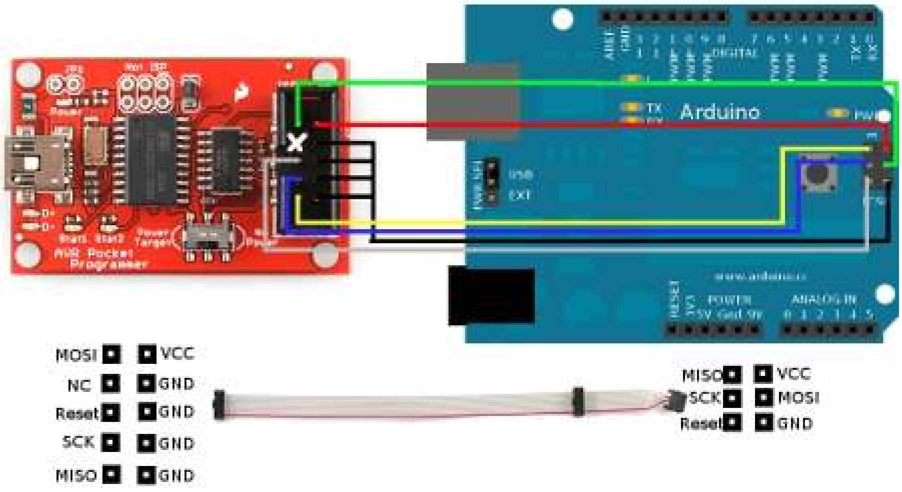
By now, the Arduino is programmed to act as an ISP programmer, and plugged into a USB cable from the computer, next we need the Arduino connected to the target board and then we can upload hex file to the target board.

Hooking up your board

Working Arduino (Option 1)	AVR Programmer (Option 2)	Arduino to be Programmed
Vcc/5V	5V	Vcc/ Pin 2
Gnd	Gnd	Gnd/ Pin 6
MOSI/D11	MOSI	MOSI/ Pin 4
MISO/D12	MISO	MISO/ Pin 1

SCK/D13	SCK	SCK/ Pin 3
D10	Reset	Reset/ Pin 5





Note: You can use the cable that comes with the programmer. The side that goes into the pocket programmer only goes in one way. The end that plugs into the Arduino needs to attach with the notch facing the reset button

Programming the Target Board

The only thing left to do is send the hex file from the computer to the working Arduino so that it can send the program to the target Arduino. Avrdude is a software program that comes with WinAVR (or Crosspack) that can send a hex file to an ISP programmer. Once you've got avrdude installed (via WinAVR or Crosspack) you're only a command away from getting the hex file to the target board. Installing these programs is simple, just follow all the prompts and install the programs with their default settings.

Avrdude is a command line program, so you'll have open a command prompt in order to use it (Start->Run->"cmd" for Windows). Once the command line is opened, type avrdude and press enter; if it's installed correctly then a list of available options will be listed on the screen. Now we're ready to send a command that will program the board. Start by navigating to the directory that the hex file was placed in.

First we are going to set fuse bits. What is a fuse bit you ask, well basically its very low level configuration of your ATMega328 chip. It controls things like whether you are using an external clock, the speed of the internal clock, external reset enable and other fun stuff. Since we don't know where your chip has been and what unsavory people have done to it, we are going to set all the fusebits so that your Arduino will work.

Option 1: avrdude -P *comport* -b 19200 -c avrisp -p m328p -v -e -U efuse:w:0x05:m -U hfuse:w:0xD6:m -U lfuse:w:0xFF:m
Option 2: avrdude -b 19200 -c usbtiny -p m328p -v -e -U efuse:w:0x05:m -U hfuse:w:0xD6:m -U lfuse:w:0xFF:m
Note: These are fuse bits designed for the ATMega328 using the Uno bootloader (at 5V/16MHz), don't use these fuse bits for other configurations without double checking they will work.

Now enter the command, replacing *comport* (com5 for example) with the COM port that your Arduino is plugged into (for option 1), and *hexfilename* for the name of your file. Don't forget the command is case-sensitive, so be careful typing it or an error may occur. The command uses seven parameters: -P is the comport used, -b is the baudrate, -c is the programmer type, -p is the microprocessor being programmed, -v tells avrdude to use a verbose output, -e tells avrdude to erase the chip before programming it, and -U tells avrdude where to put the hex file. We also wrote the lockbits on your chip. The lock bits are similar to fuse bits but they control security and stuff on your chip. In this case they are important because they prevent you from accidently writing over your bootloader by installing a program that is too big.

Option 1: avrdude -P *comport* -b 19200 -c avrisp -p m328p -v -e -U flash:w:*hexfilename.hex* -U lock:w:0x0F:m
Option 2: avrdude -b19200 -c usbtiny -p m328p -v -e -U flash:w:*hexfilename.hex* -U lock:w:0x0F:m

Note: Remember that 'Arduinos' are really just AVR development boards running the Arduino bootloader. You can pretty much use this method to load any .hex file to your ATMega328 chip. For more information check out our tutorial on using your Arduino to program an AVR (http://www.sparkfun.com/tutorials/200)

Note: You can also skip the command line options and use the Arduino IDE to reprogram the bootloader. Make sure you have the correct board and port selected, then go to Tools->Burn Bootloader->(select your ISP) and select "w/ USBtinyISP" for the pocket programmer. The disadvantage to this method is you don't have any control over which bootloader you are uploading to your board

SparkFun Electronics ® | Boulder, Colorado | Customer Service