Featured in this Guide

FTDI Friend + extras

$14.75 Add to Cart

Arduino Uno R3 (Atmega328 - assembled)

$29.95 Add to Cart

USB cable - A/MiniB

$4.00 Add to Cart
Breadboarding wire bundle

$6.00 Add to Cart
USB FTDI TTL-232 cable

$20.00 Add to Cart
Add all to Cart
×

Out of Stock Notification

Your Name: Your E-mail:
Notify Me

FTDI Friend

FTDI Friend is a friend, not an enemy

Feedback? Corrections?

Programming the Arduino Bootloader Created by Ladyada

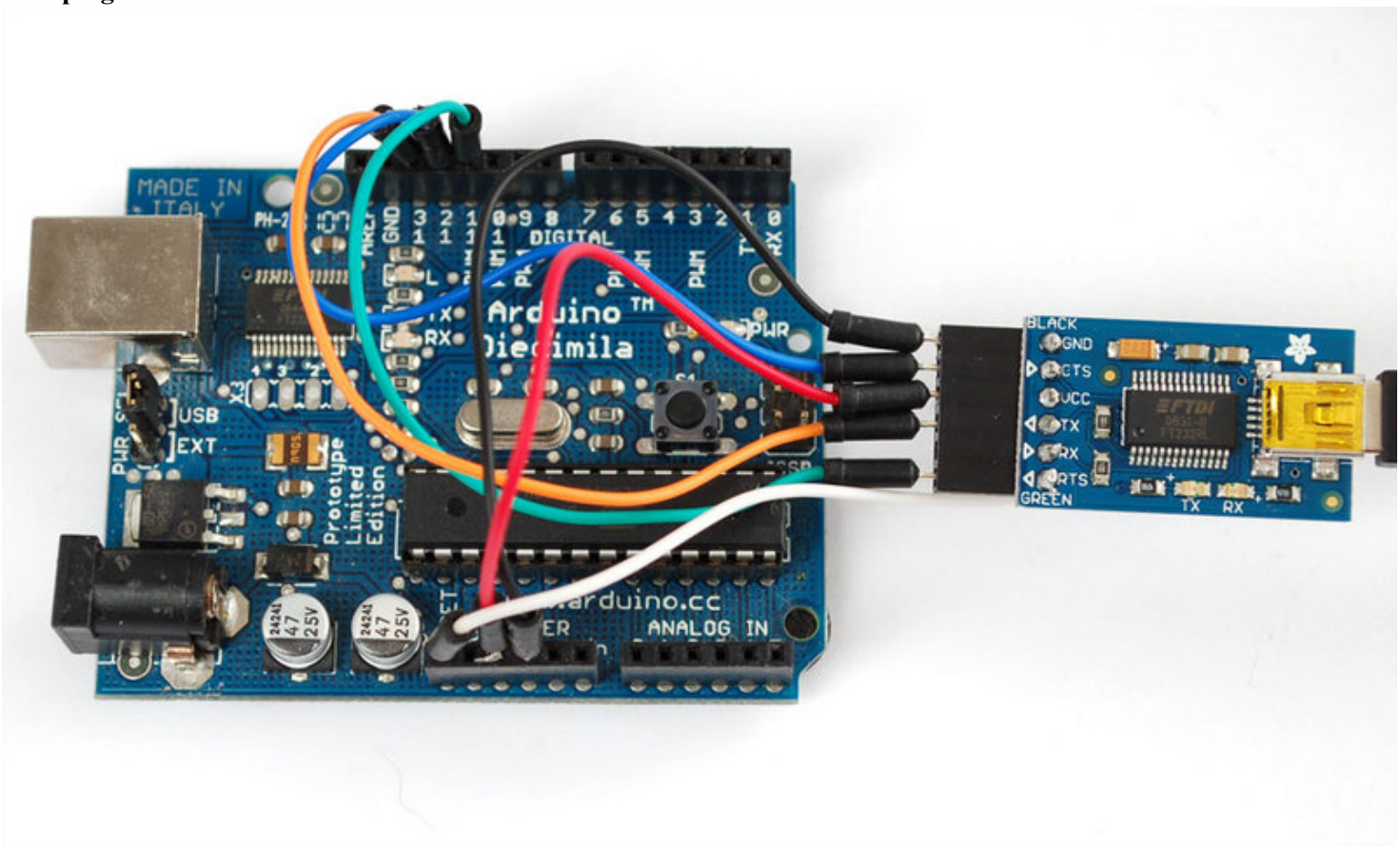
Don't forget, if you have two 'duinos, you can turn one into an ISP programmer, check out this tutorial which runs much faster than the below.

You can burn chips right from the Arduin IDE with one more edit, open up the **programmers.txt** file (its in **hardware\arduino** in the IDE installation) and add the following text to the bottom.

Copy Code

```
ftdifriend.name=FTDI Friend bitbang
ftdifriend.communication=serial
ftdifriend.protocol=ftdifriend
```

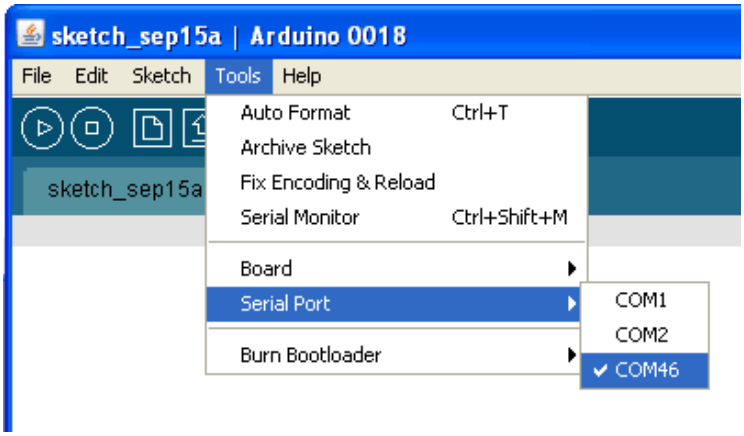
Save **programmers.txt**



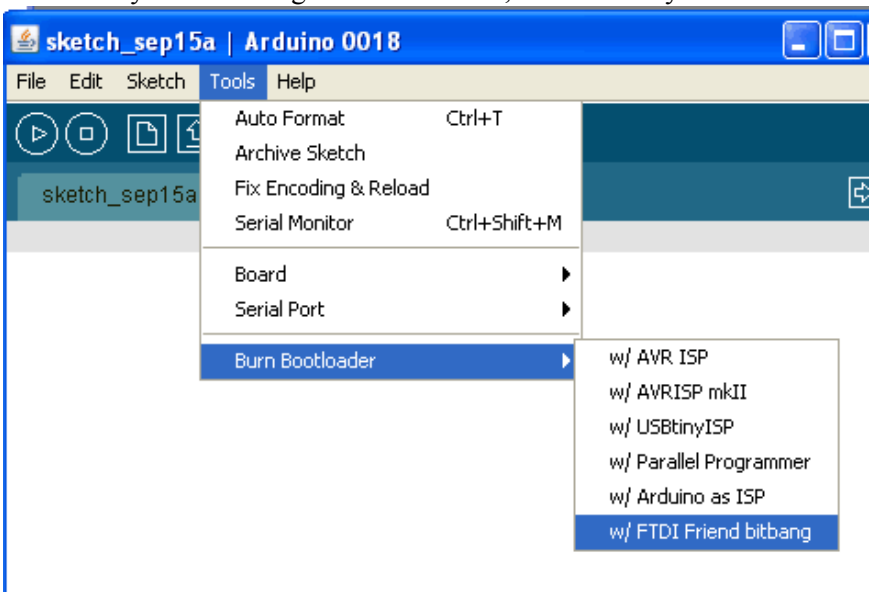
Its time to wire it up!

- Connect the Black (Ground) wire to the ground of your chip
- Connect the Red (VCC) wire to the power/VCC/5V pin of your chip
- Connect the White (DTR) wire to the Reset pin
- Connect the Orange (TX) wire to SCK pin (Arduino pin 13)
- Connect the Green (RTS) wire to MOSI pin (Arduino pin 12)
- Connect the Blue (CTS) wire to the MISO pin (Arduino pin 11)

Start up the IDE and select the COM port of the FTDI friend



Make sure you have the right **Board** selected, for whatever you want to burn. Then select **Burn Bootloader→FTDI Friend**



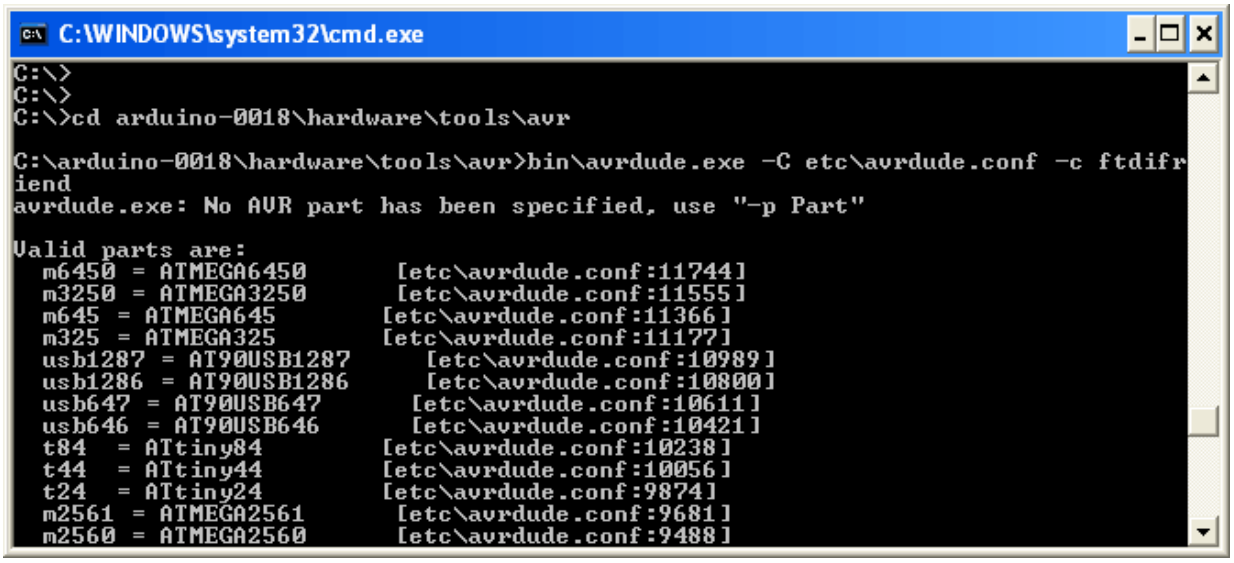
It will take a really really long time to program the chip, about two hours!

Its taking forever because its programming very inefficiently. It takes 4 bytes to program one byte of the AVR, and data is sent as a single bit in two USB packet, each packet takes 3 milliseconds and an AVR has 32768 bytes = 262144 bits. 262144 bits * 2 packet/bit * 3 ms/packet * 4 bytes/byte = 6291456 ms = 6300 seconds = 104 minutes! If you used a real AVR programmer, it would take maybe 15 seconds so thats why its nice to have one.

Do this before you go to bed or watch a movie! The **L** (pin 13) LED will be on 'solid' while its programming. When it goes out you're done. Its very slow but if you're in a pinch, it may come in handy!

Using the Command Line AVRdude

Now if you open up a terminal and try running **avrdude -c ftdifriend** If you are using the avrdude installation thats inside the Arduino IDE, you can open up a **cmd** terminal and **cd** to the directory where you have the IDE installed and then go to **hardware\tools\avr**. Then you can run **bin\avrdude.exe -C etc\avrdude.conf -c ftdifriend** so for example, my installation is in **C:\arduino-0018** I **cd** to **C:\arduino-0018\hardware\tools\avr.**

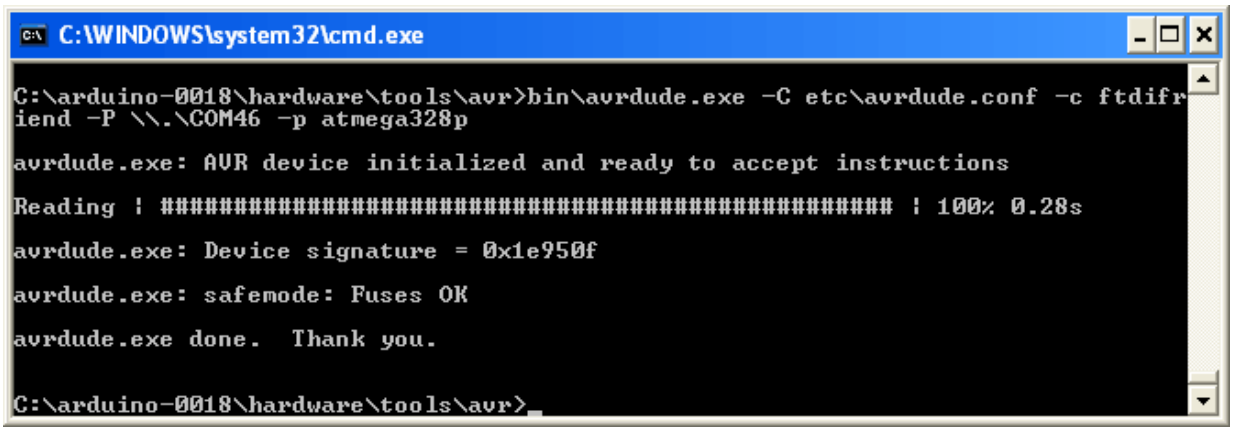


You should get a note that **No AVR part has been specified** (not that it **Can't find programmer id "ftdifriend"**)

Great, now its time to program!

- Connect the Black (Ground) wire to the ground of your chip
- Connect the Red (VCC) wire to the power/VCC/5V pin of your chip
- Connect the White (DTR) wire to the Reset pin
- Connect the Orange (TX) wire to SCK pin
- Connect the Green (RTS) wire to MOSI pin
- Connect the Blue (CTS) wire to the MISO pin We suggest verifying the wiring! We'll verify the connection by running **avrdude -c ftdifriend -P \\.\COMxx -p atmega328p** Replace the **\\.\COMxx** with the COM port you found in the earlier part of this tutorial via the Device Manager. If you're using linux or mac, the COM port should be **/dev/cu.usbserialXXX** or **/dev/ttyUSBx** to match the name. For the device, we're testing with an **Atmega328p** chip which is found in the latest Arduinos. If you're using some other chip, substitute the name right after the **-p**

If you are using the avrdude in the Arduino IDE, you'll need to use **bin\avrdude -C etc\avrdude.conf -c ftdifriend -P \\.\COMxx -p atmega328p** etc.



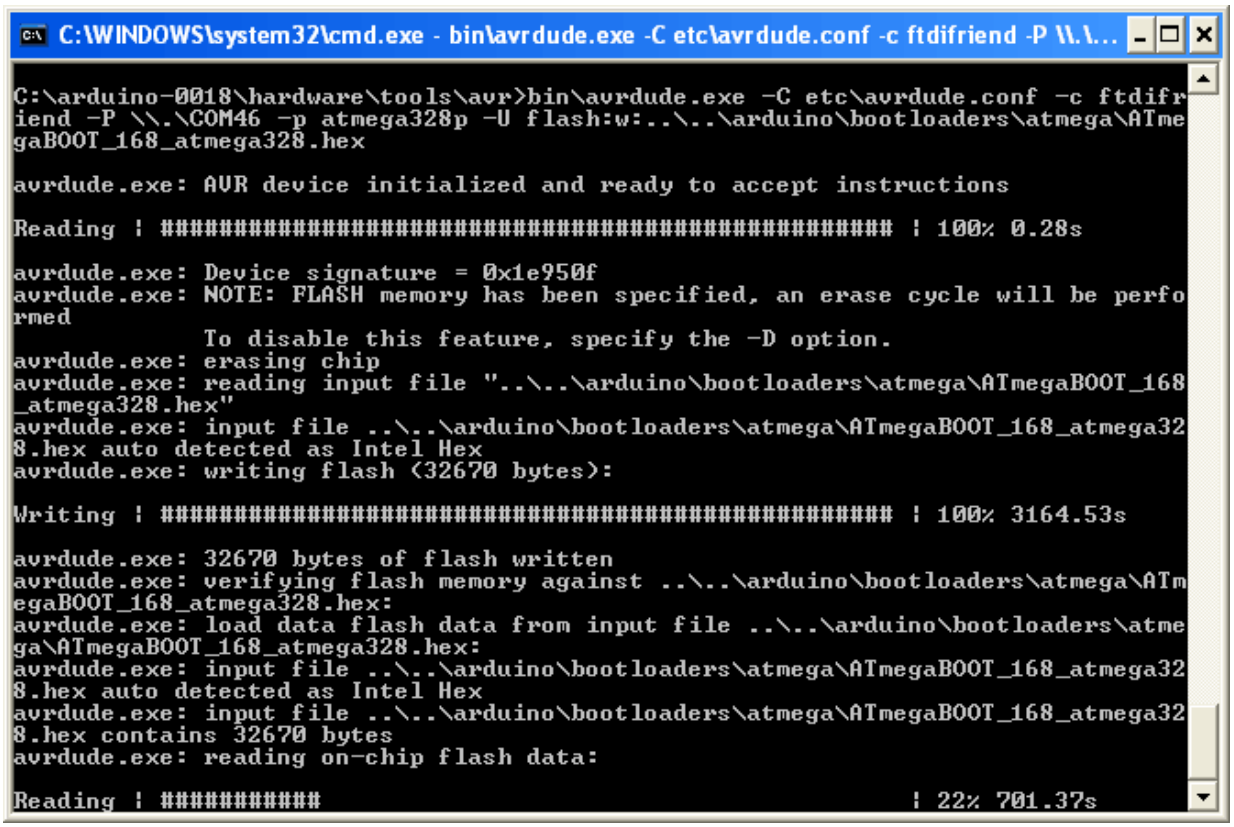
Once you have the hashbar show up nicely, that means you are talking to the chip all right. Yay! You can now program the chip using the **-U flash:w:** command.

Press the up arrow to get back the command you wrote and tack onto the end a **-U flash:w:firmware.hex** (where **firmware.hex** is the hex file you want to burn in) For example, if you want to burn a bootloader on, make it easy on yourself by copying the file from use:

bin\avrdude.exe -C etc\avrdude.conf -c ftdifriend -P \\.\COM46 -p atmega328p -U

flash:w:..\..\arduino\bootloaders\atmega\ATmegaBOOT_168_atmega328.hex

(you can also copy your file to C:\ so its not so long to type. Hit return and grab a beer, it'll take about 5 minutes per byte of data you want to write&verify)



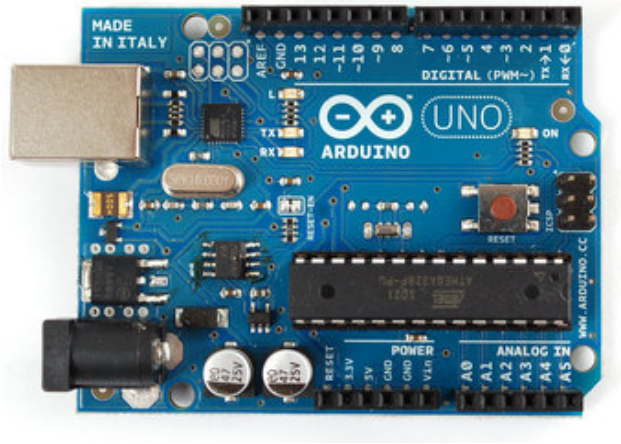
< Programming Blank AVRs Downloads >

Last updated on 2013-01-28 at 05.17.05 PM

Related Guides

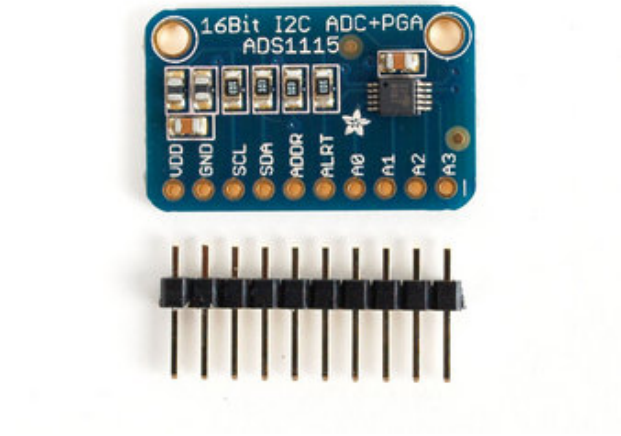
- **rePaper eInk Development Board**

  Daylight Readable Graphical eInk Displays



- **Arduino Tips, Tricks, and Techniques**

  Get the most out of your Arduino



- **Adafruit 4-Channel ADC Breakouts**

  A guide to the ADS1115 and ADS1015 analog converters

About Adafruit

Adafruit was founded in 2005 by MIT engineer, Limor "Ladyada" Fried. Her goal was to create the best place online for learning electronics and making the best designed products for makers of all ages and skill levels.
Learn More.

The Adafruit Learning System

In 2012, after years of coding tutorials by hand, Limor put together a small team to build a custom tutorial management system from the ground up. The Adafruit Learning System allows us to make a wide range of awesome tutorials fast and efficiently. We hope these tutorials will help you learn something new, and inspire you to make something great!

Useful Stuff

Support Forums Adafruit Shop RSS

327 tutorials and counting

Powered by Adafruit Learning Technologies