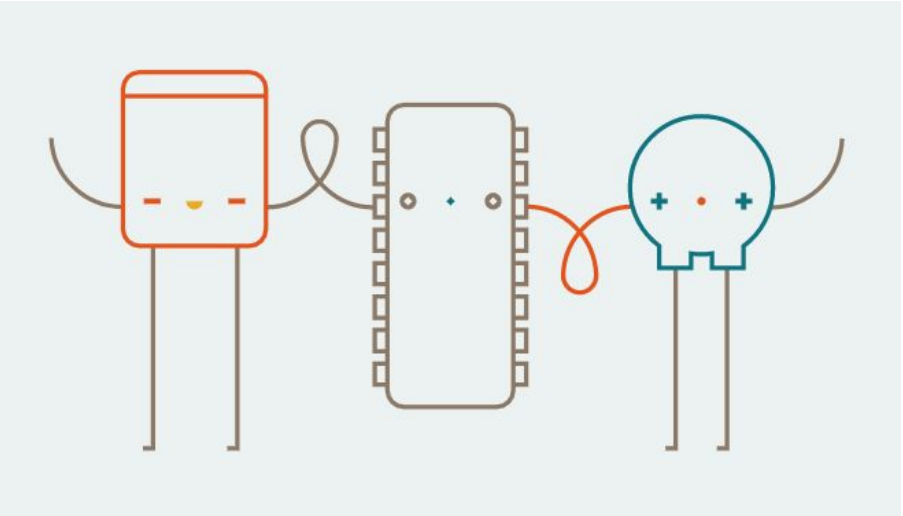
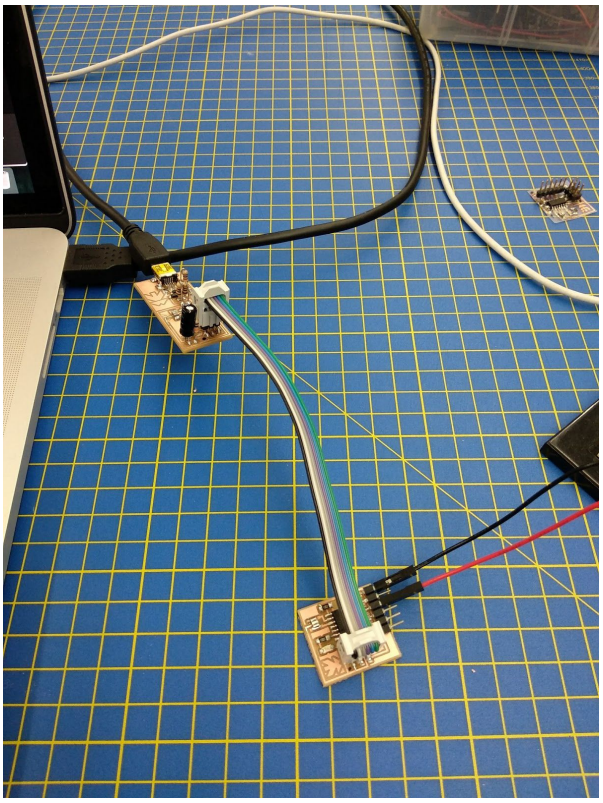
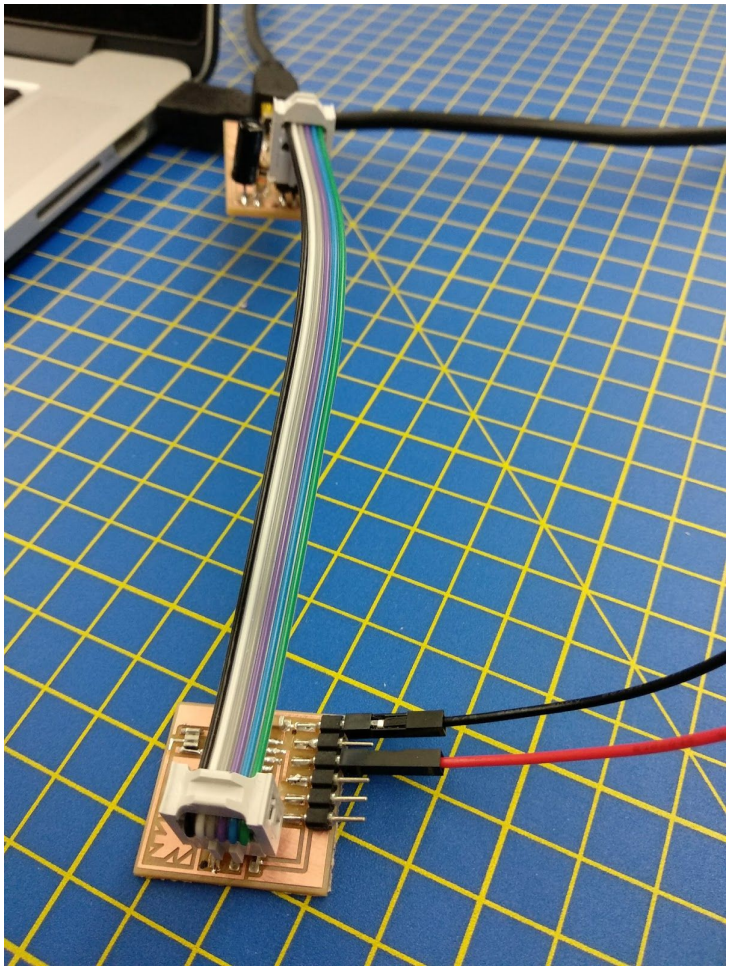
-ATtiny44(A SSU)-



Version of Arduino I used:
1.8.5 (Latest, should be)

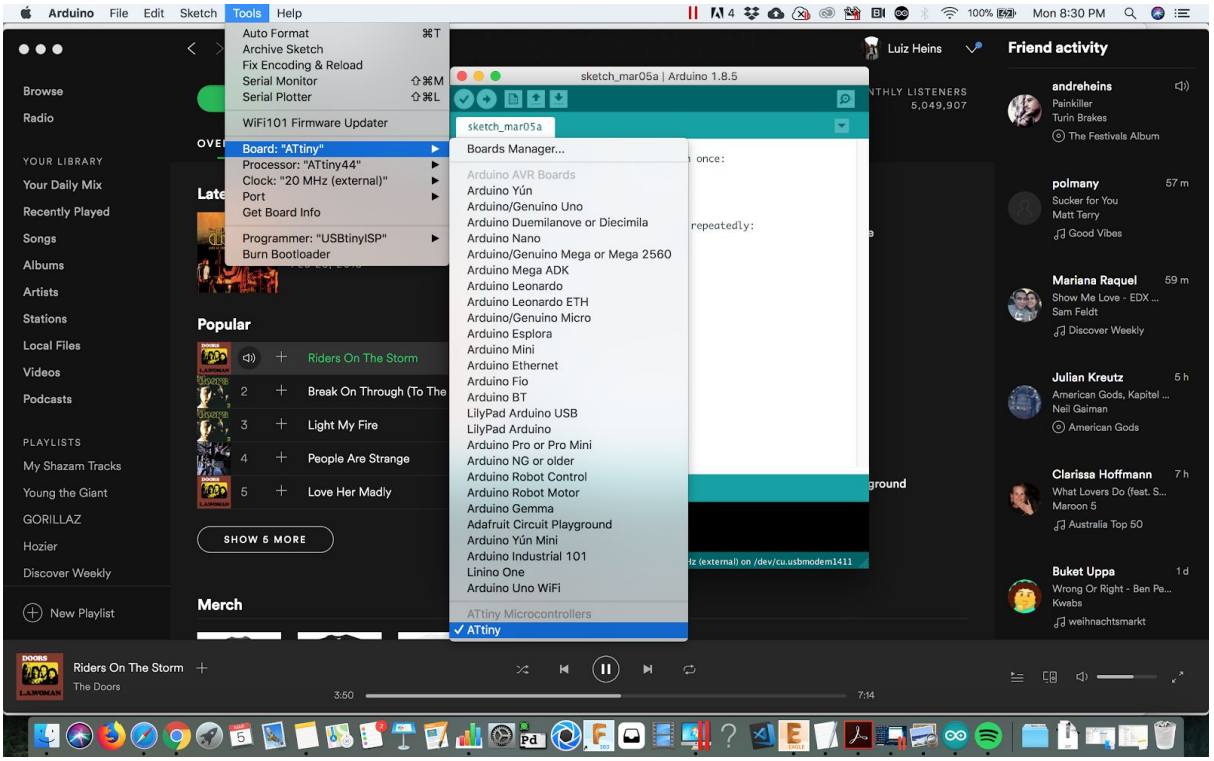
Programming your board is straight forward, if all went well with stuffing. You'll need your ISP-programmer. I left, on the table in front of the Laser-Cutter, the power-supply wired. The black alligator-cable is GND (-), the red is VCC (+).
You'll clip them to the breadboard (The VCC is already clipped, the Gnd not) and use the Female-Male jumper-cables to power your board, plugging them to the FTDI pins of your board. Like these images below: (Note the red and black cables pinned to the FTDI pins of your board)

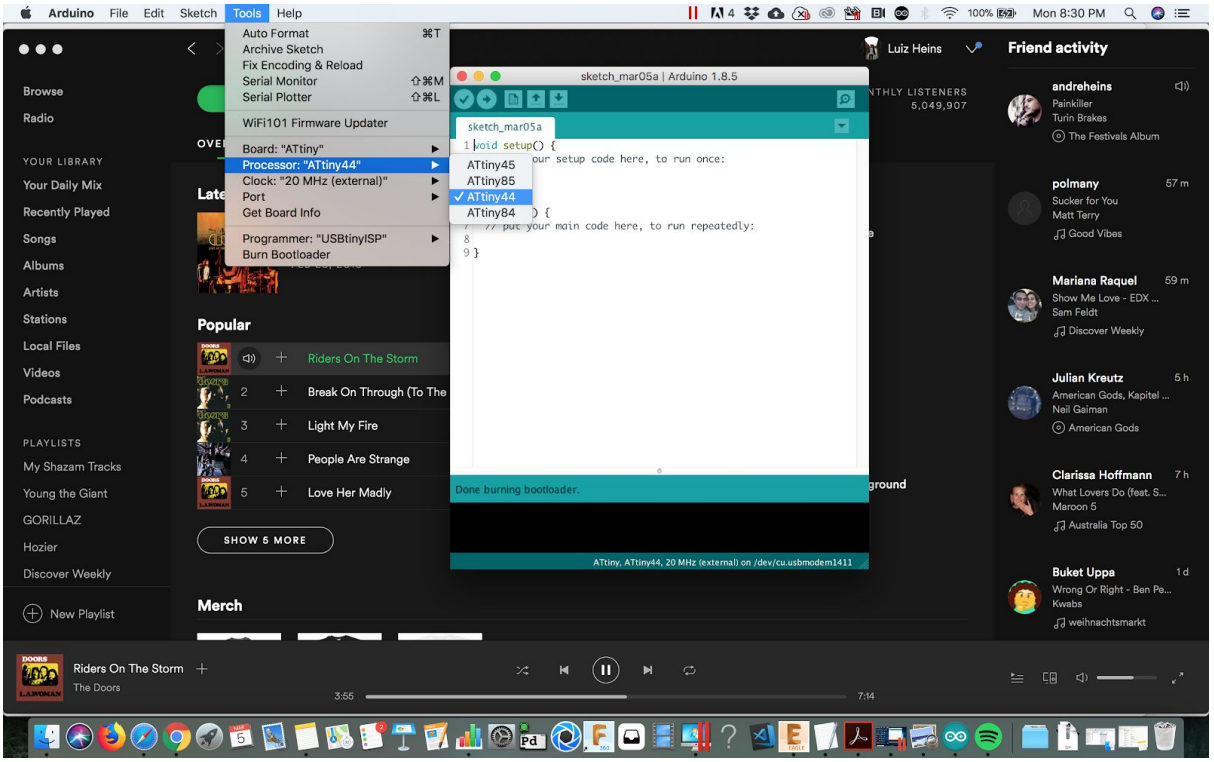


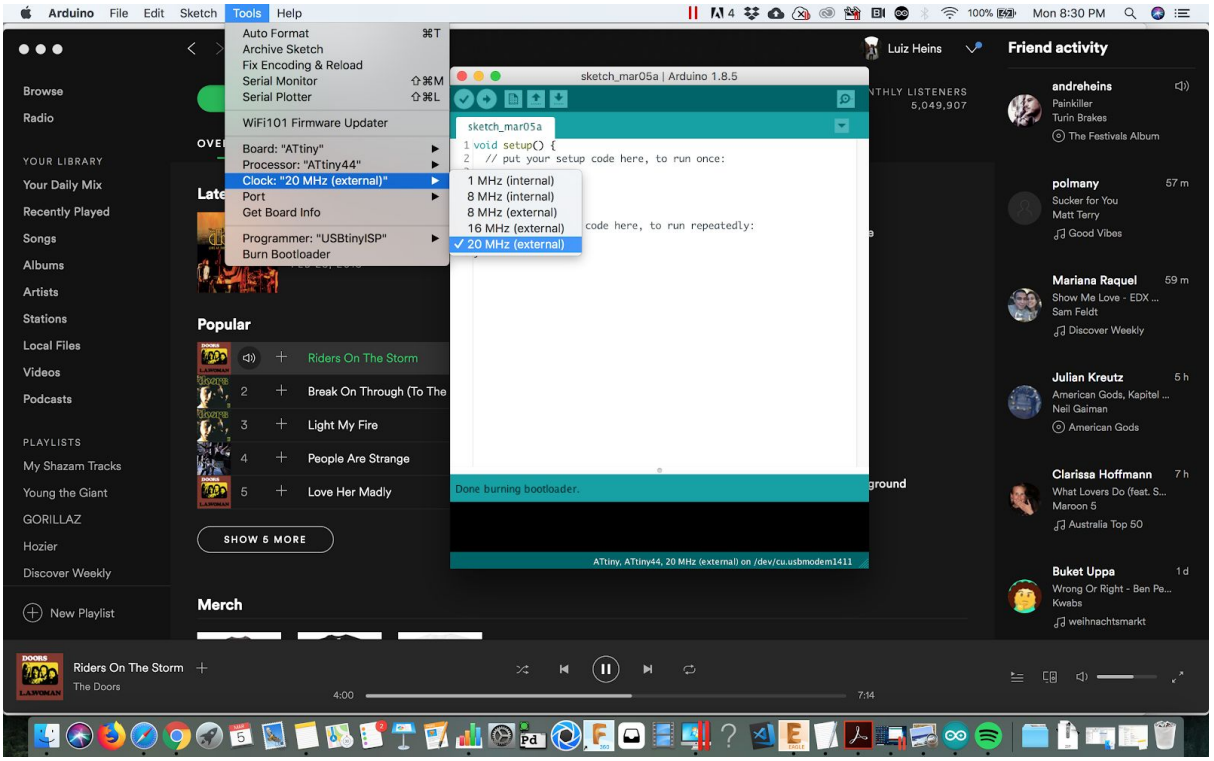


Plug your ISP programmer also as you see in the picture.
To your computer, only the USB connected to the ISP programmer needs to be connected. If someone else is already using the Power-Supply, you can power the Gnd and Vcc of your board in different ways.
You can use the FTDI cable plugged to your board. That will also power your board from another USB port. Make sure to take the 5V FTDI one and not the “3V3” one. It means 3.3V - It is written on the USB portion of the cable.
Another way to power your board is if you have another arduino, 5V and Gnd.
The image above shows the Black cable plugged to the 1st FTDI pin. That is ground/GND/(-). The red on the 3rd one is the VCC/positive/(+)/5V.
Make sure you downloaded the ATtiny board to your arduino.
Here is how:
http://highlowtech.org/?p=1695
And here the library:
https://github.com/damellis/attiny/

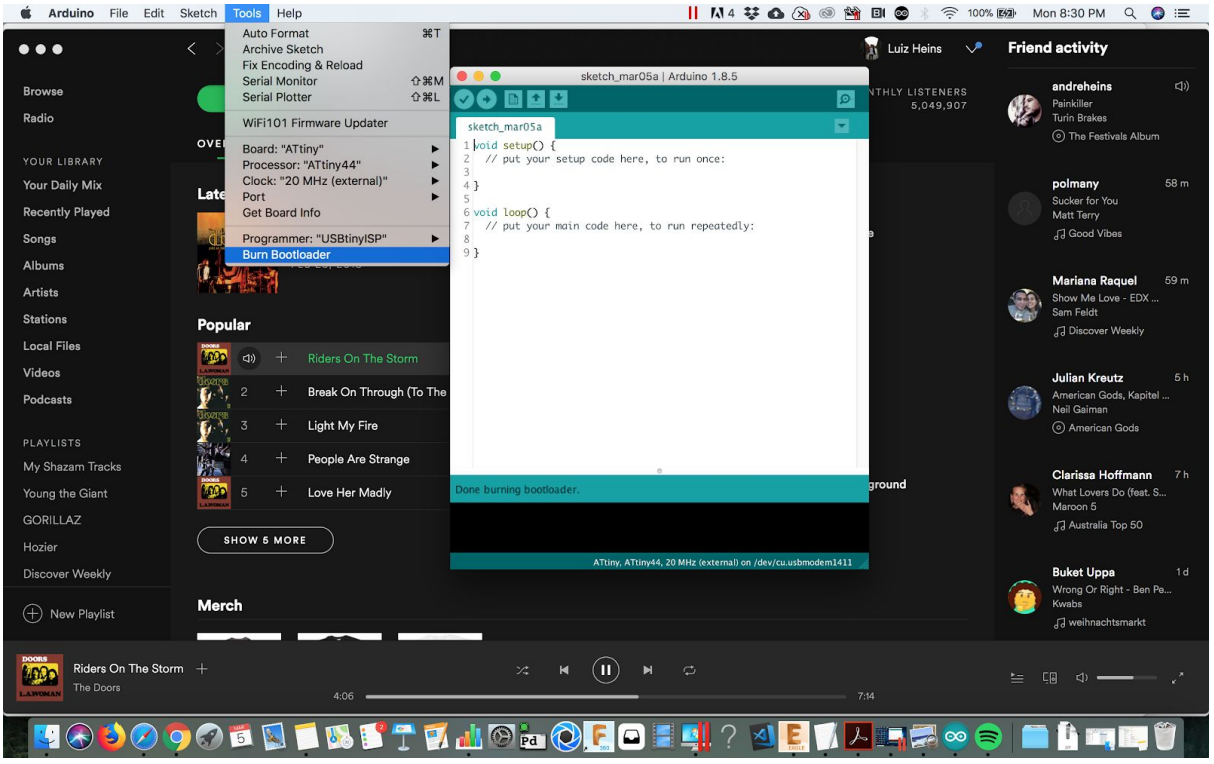
Once you have them,


Arduino File Edit Sketch Tools Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Serial Plotter ⇧⌘L
WiFi101 Firmware Updater
Board: "ATtiny"
Processor: "ATtiny44"
Clock: "20 MHz (external)"
Port
Get Board Info
Programmer: "USBtinyISP"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino/Genuino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground
Arduino Yún Mini
Arduino Industrial 101
Linino One
Arduino Uno WiFi
ATtiny Microcontrollers
ATtiny
sketch_mar05a | Arduino 1.8.5
sketch_mar05a
Luiz Heins
Friend activity
Popular
Riders On The Storm
Break On Through (To The
Light My Fire
People Are Strange
Love Her Madly
SHOW 5 MORE
Merch
Browse
Radio
YOUR LIBRARY
Your Daily Mix
Recently Played
Songs
Albums
Artists
Stations
Local Files
Videos
Podcasts
PLAYLISTS
My Shazam Tracks
Young the Giant
GORILLAZ
Hozier
Discover Weekly
New Playlist
Riders On The Storm
The Doors
3:50
7:14

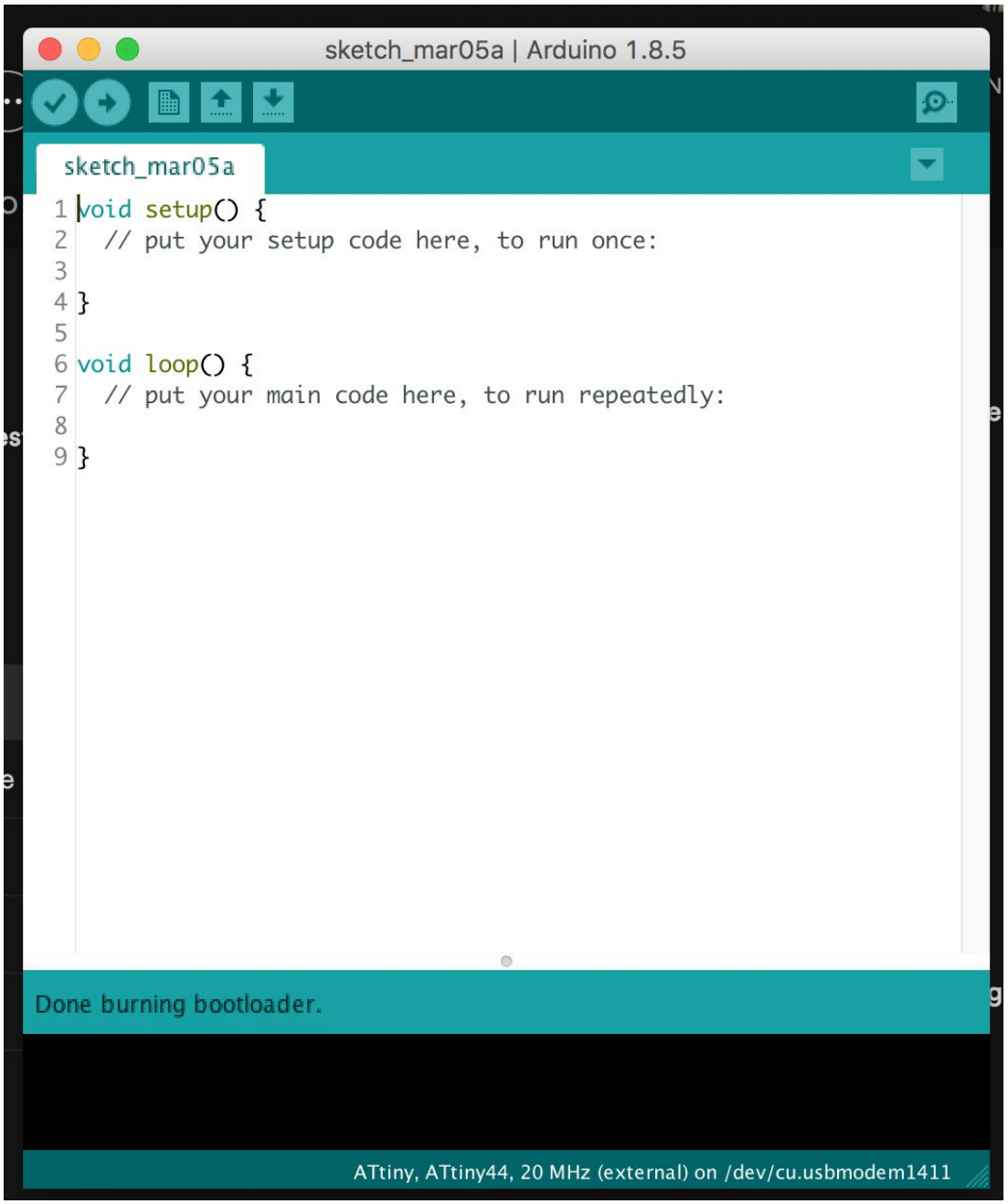

Arduino File Edit Sketch Tools Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Serial Plotter ⇧⌘L
WiFi101 Firmware Updater
Board: "ATtiny"
Processor: "ATtiny44"
Clock: "20 MHz (external)"
Port
Get Board Info
Programmer: "USBtinyISP"
Burn Bootloader
ATtiny45
ATtiny85
ATtiny44
ATtiny84
sketch_mar05a | Arduino 1.8.5
sketch_mar05a
void setup() {
our setup code here, to run once:
// put your main code here, to run repeatedly:
Done burning bootloader.
ATtiny, ATtiny44, 20 MHz (external) on /dev/cu.usbmodem1411
Luiz Heins
Friend activity
Popular
Riders On The Storm
Break On Through (To The
Light My Fire
People Are Strange
Love Her Madly
SHOW 5 MORE
Merch
Riders On The Storm
The Doors
3:55
7:14



MAKE SURE TO HAVE SELECTED THE USBtinyISP (that's your programmer)

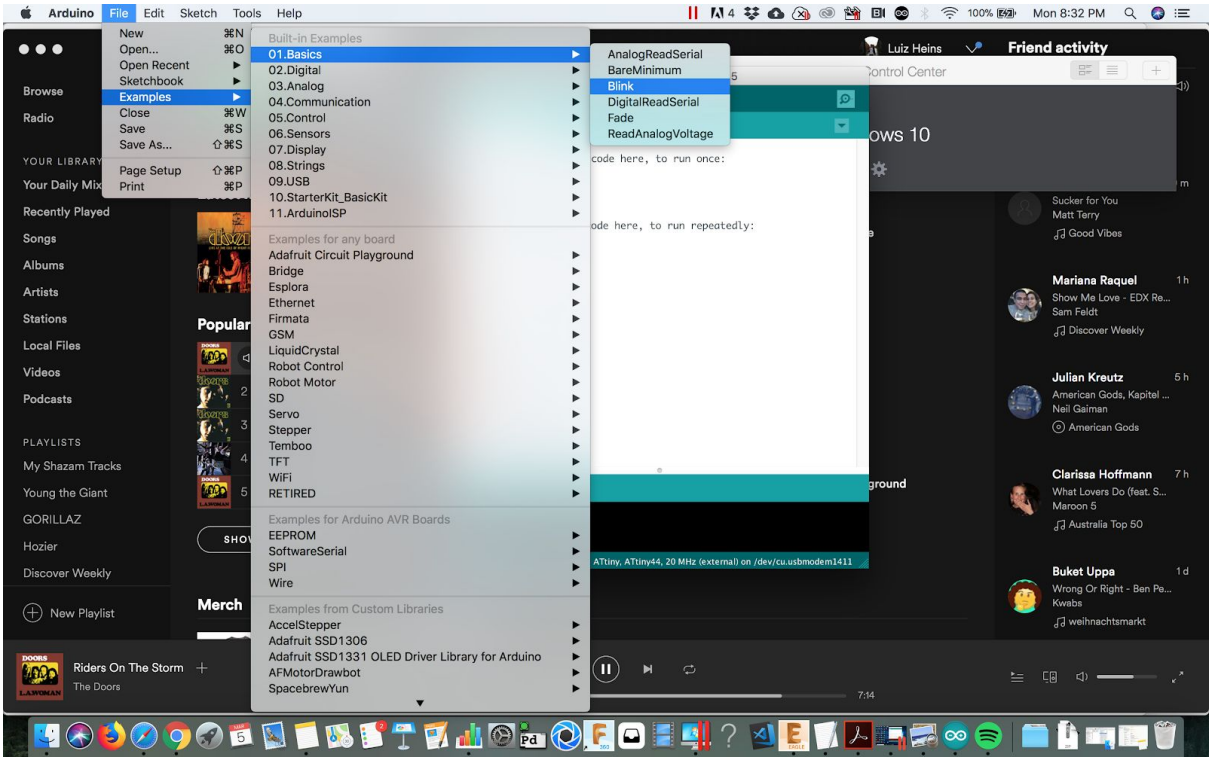


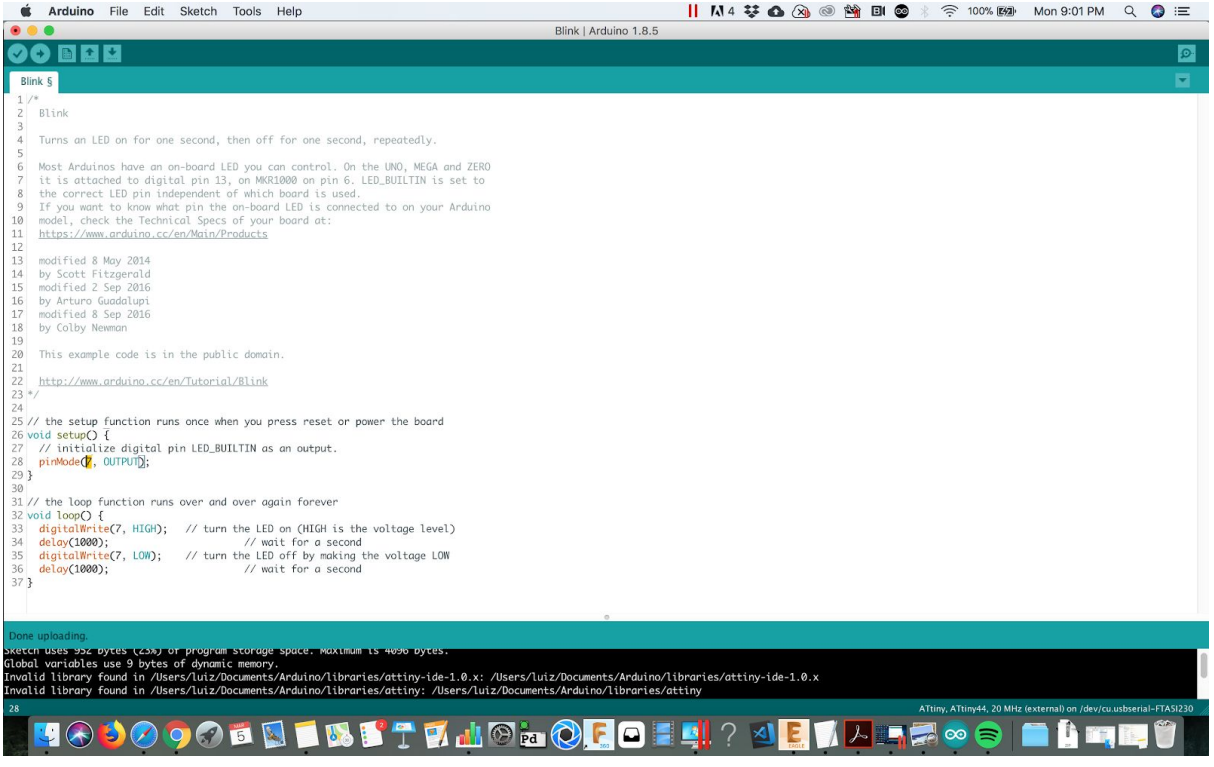
Click on "Burn Bootloader". Cross your fingers.



No? Check your soldering with the Multimeter. Find the problem, try again.

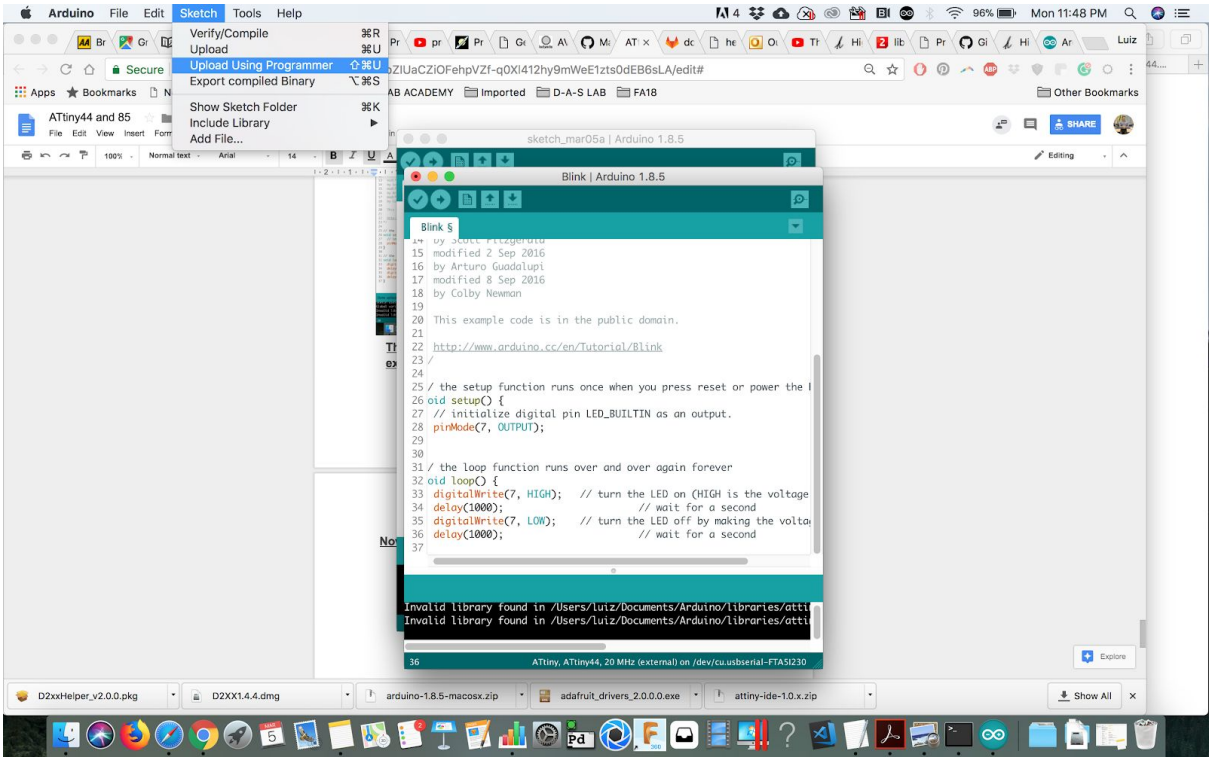
Then:



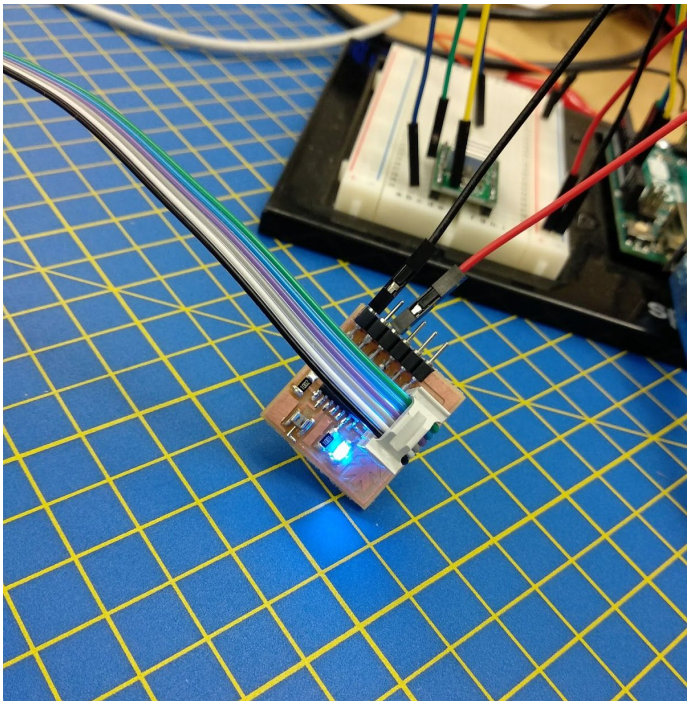


The pin PA7 of your helloWorld is the Arduino pin 7. In the blink example, replace the “LED_BUILTIN” for the number 7. Like this picture above. That’s the only thing you’ll do to the code.

Now, your 2x3 ribbon cable is still connected between the ISP and the HelloWorld, and you're still powering the board.



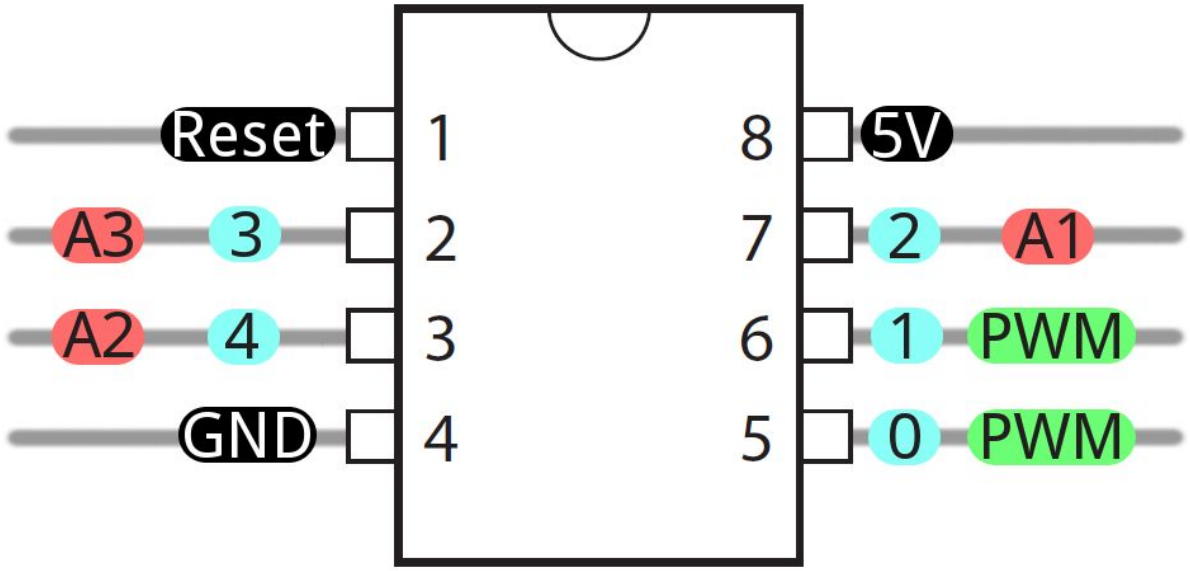
Upload using the Programmer (Or Shift + Command + U, on Mac). If all goes well it should blink! HelloWorld!



- ATtiny85 -

Arduino Pinout: (NOT FINISHED. Ignore from here on, for now)

(NOT FINISHED. Ignore from here on, for now)



5 I/O Pins available
Beyond that, some pins have special functionality.

Analog Input and Output
There are two analog outputs and three analog inputs. Use them just as you would with any Arduino board. Use analogWrite([pin], [0-255]) to do PWM output. This functionality is available on pins ***0 and 1***. For example:

```
int pwmPin = 0;

pinMode(pwmPin, OUTPUT);

for (int i=0; i<=255; i+=5)
{
  analogWrite(pwmPin, i);
  delay(5);
}
```

And use analogRead([pin]) to read an analog voltage between 0 and 5V, and turn it into a 10-bit representation of that voltage. Pins **2, 3, and 4** are capable of analog input, but, when using them as such, they should be referenced as A1, A3, or A2 respectively. For example:

```
int pwmPin = 0;
int analogInPin = A1;

pinMode(pwmPin, OUTPUT);
```

```
pinMode(analogInPin, INPUT);

int analogIn = analogRead(analogInPin); // Read analog voltage on pin 2 (A1)

analogWrite(pwmPin, analogIn / 4); // Output analog reading to dimmable LED
```

Advanced:
http://www.technoblogy.com/show?LE0
(Get up to 4 PWM channels at ATtiny85)

No Serial (UART). Yes SPI and I2C.

You may notice, on the listing of special pin functions there are no UART RX's or TX's. That's because the ATtiny85 doesn't have a built in hardware UART. If you try to compile any Arduino code with `Serial.begin(9600)`'s or `Serial.print()`'s you'll get an error.

So you're out one of the more useful Arduino debugging tools. You can't print to the Serial Monitor. But the ATtiny85 does still have I2C and SPI, which are much more commonly used for sensor communication these days. Unfortunately, the Arduino libraries for these interfaces haven't yet been written for the ATtiny85, but there are some user contributed libraries around the web. USIi2c is an Arduino library which enables I2C on the ATtiny85.

There are other ATtiny85-focused libraries out there too. Like a Servo8Bit, a servo library.

Source:
https://learn.sparkfun.com/tutorials/tiny-avr-programmer-hookup-guide/attiny85-use-hints

PROGRAMMING ATtiny85 in C language, using a Makefile:

program the hello board through the command line using the fabISP:

Install libusb on Mac OSX

About the App

- **App name: libusb**
- **App description: Library for USB device access**
- **App website: http://libusb.info**

Install the App

1. **Press `Command+Space` and type *Terminal* and press *enter/return* key.**

2. **Run in Terminal app:**
3. `ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" < /dev/null 2> /dev/null`
4. **and press *enter/return* key. Wait for the command to finish.**
5. **Run:**
6. `brew install libusb`

Done! You can now use `libusb`

Should look like this:

http://macappstore.org/libusb/

V-USB tester
https://forums.obdev.at/viewtopic.php?f=8&t=7719

Source:
http://maxembedded.com/2015/06/setting-up-avr-gcc-toolchain-on-linux-and-mac-os-x/

http://shallowsky.com/blog/hardware/attiny85-c.html

http://electronut.in/getting-started-with-attiny85-avr-programming/

Makefile:
https://gist.github.com/electronut/8a4c297213620958ebef

AVR tutorial: (For all boards, not only the ATtiny85)
http://www.ladyada.net/learn/avr/avrdude.html

Burning fuses

Fuse memory is a seperate chunk of flash that is not written to when you update the firmware. Instead, the 3fuses tend to be set once (altho they can be set as many times as you'd like). The fuses define things like the clock speed, crystal type, whether JTAG is enabled, what the brownout (minimum voltage) level is, etc. For more information on fuses you can read about 'em here.

First you'll want to calculate fuses using the very convenient AVR Fuse Calculator