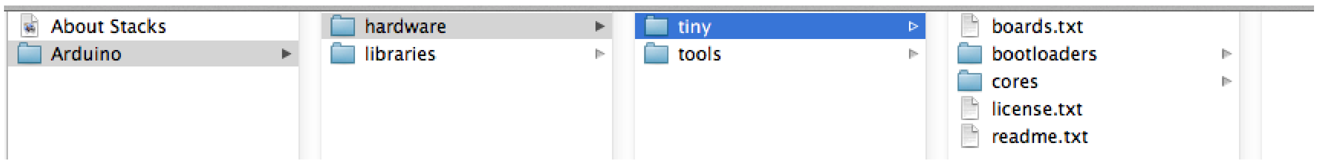
Chipper First Script

Download the up to date version of the Arduino software at the Arduino web page
Donwload the board definitions from arduino-tiny

If you are installing the Arduino software on Windows, please see this guide about installing drivers.

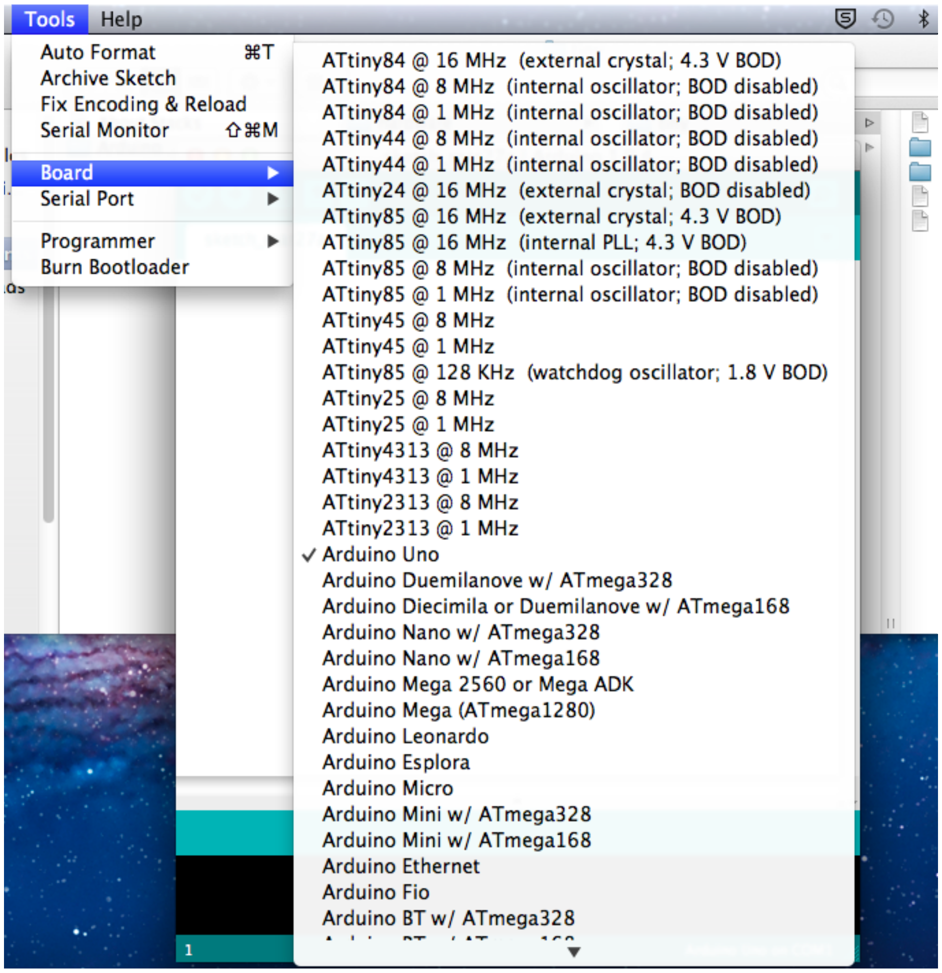
Install the Arduino software. Next we need to install the board definitions for the ATtiny chips. Board definitions are what make the Arduino software compatible with the ATtiny chips.

In your Documents folder, open up the Arduino folder and create a folder called “hardware”. Inside of the “hardware” folder take the 2 folders (“tiny” and “tools”) inside of the folder, downloaded from attiny-tiny and move them into the “hardware” folder.



Quit the Arduino software and open it again.

Click on the “Tools” tab in the Arduino software and then click on “Board”, it should reveal contents that look like this:


Tools
Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Board
Serial Port
Programmer
Burn Bootloader
ATtiny84 @ 16 MHz (external crystal; 4.3 V BOD)
ATtiny84 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny84 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny44 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny44 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny24 @ 16 MHz (external crystal; BOD disabled)
ATtiny85 @ 16 MHz (external crystal; 4.3 V BOD)
ATtiny85 @ 16 MHz (internal PLL; 4.3 V BOD)
ATtiny85 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny85 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny45 @ 8 MHz
ATtiny45 @ 1 MHz
ATtiny85 @ 128 KHz (watchdog oscillator; 1.8 V BOD)
ATtiny25 @ 8 MHz
ATtiny25 @ 1 MHz
ATtiny4313 @ 8 MHz
ATtiny4313 @ 1 MHz
ATtiny2313 @ 8 MHz
ATtiny2313 @ 1 MHz
✓ Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328

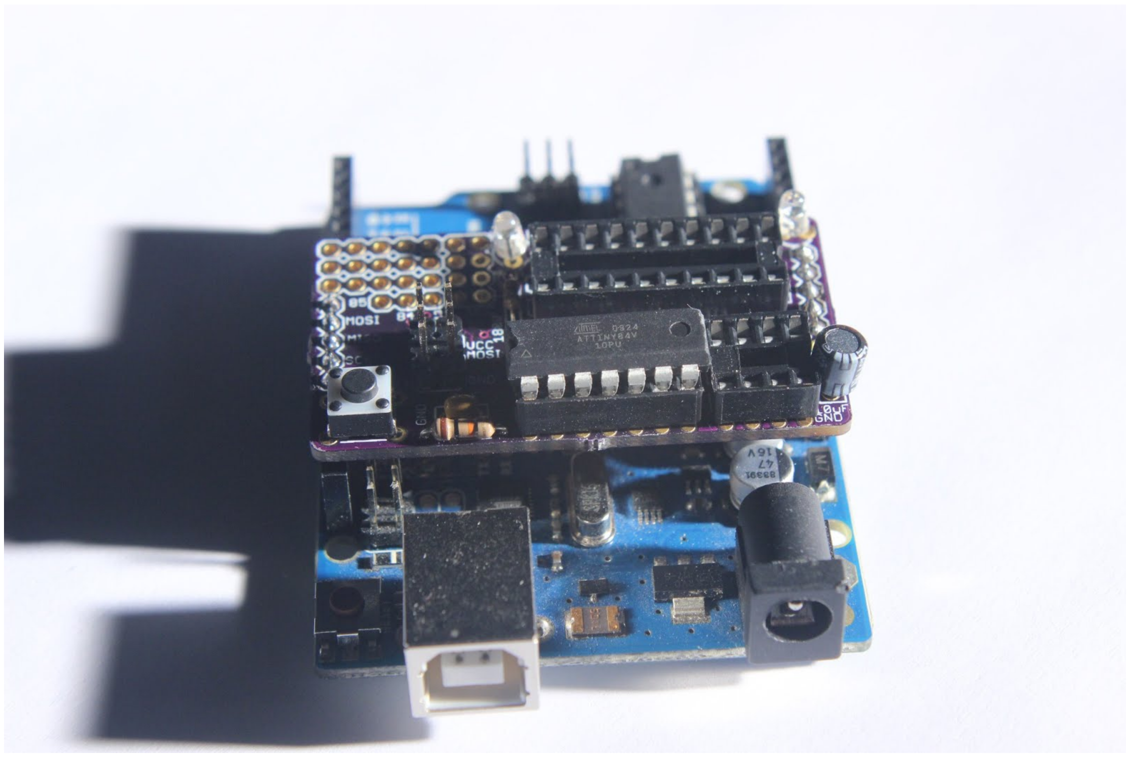
That is everything you need to configure in the Arduino software. At this point, the guide will split if you are programming with an Arduino board or AVR programmer.

Chipper Board Pinouts

Chip	LED(s) Pin	Sensor Pin
ATtiny85	4	5
ATtiny84	0,2	9
ATtiny4313	8,9	14

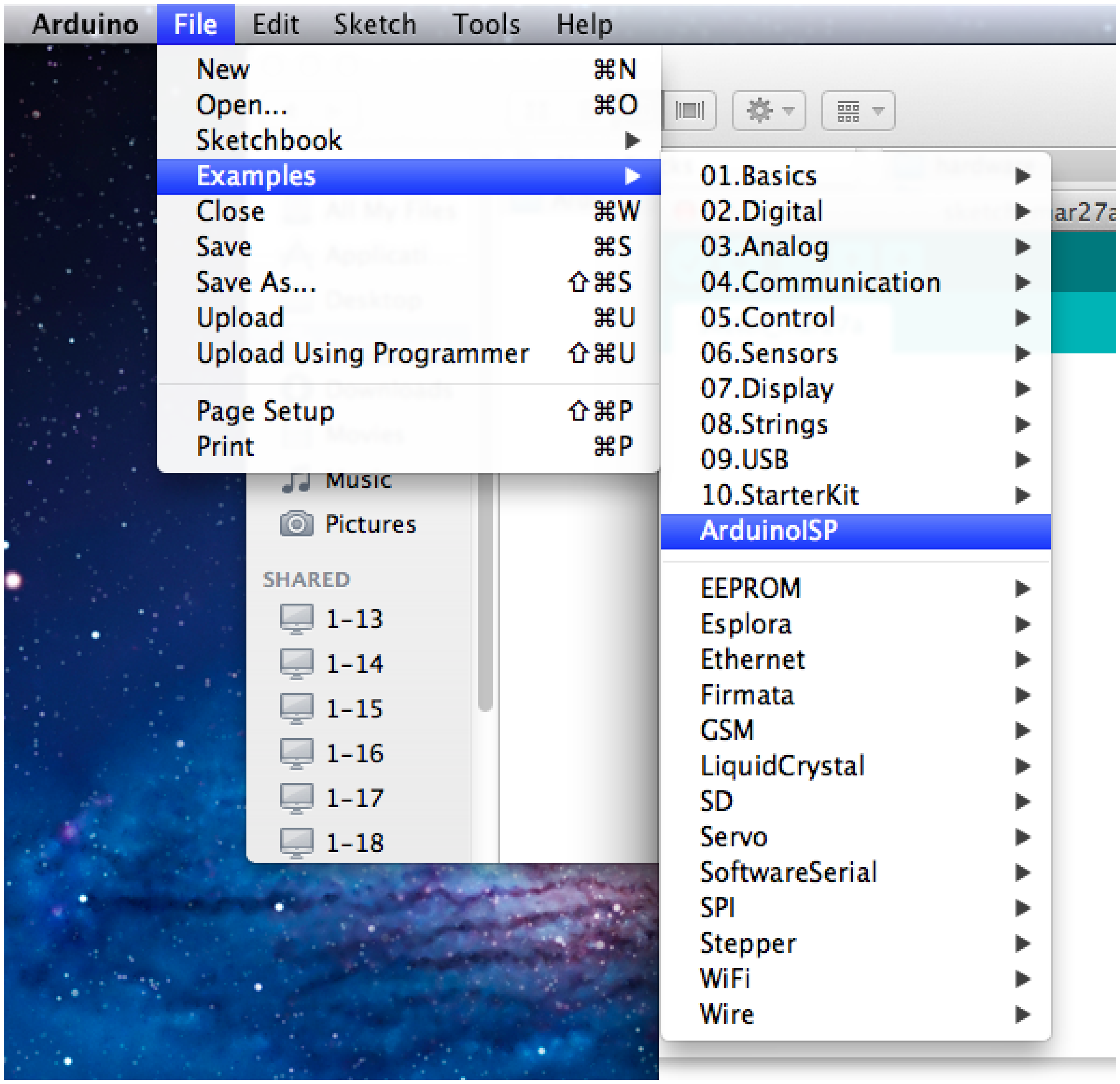
Programming with Arduino Board

If you have an Arduino Uno or previous Uno version (ex. Duemilanove the Leonardo will not work) you can plug the shield directly into the board.



First, however, you need to upload a script to the Arduino Uno.

Open the ArduinoISP sketch and remove anything including the Chipper shield that is attached to the Arduino Uno. Whenever you are programming the Arduino, in order to avoid any upload errors try to remove all peripherals. Remember to select the Arduino Uno under the board section. The majority of uploading issues to the ATtiny chips are caused by selecting the wrong board.



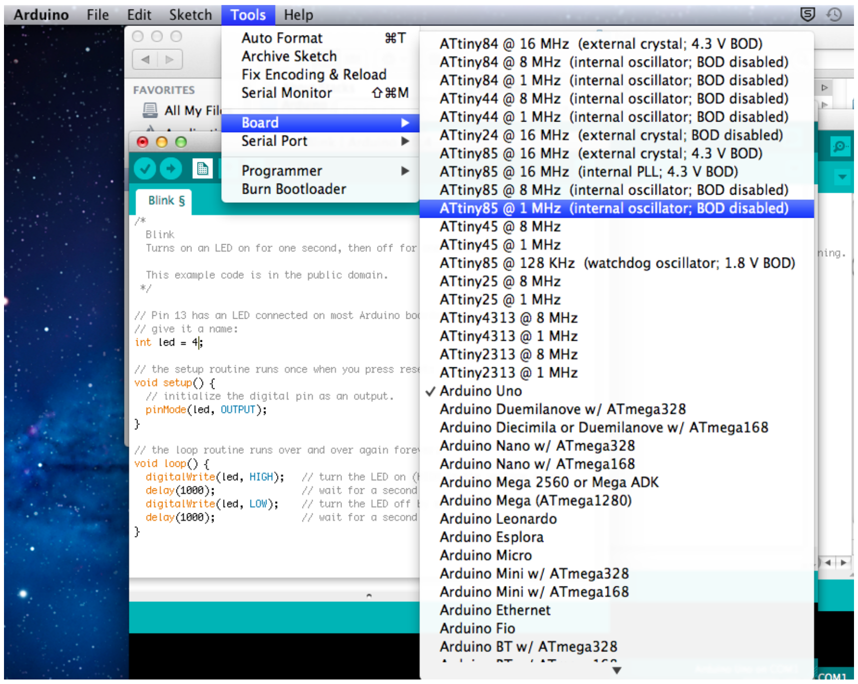
If you are using an Arduino Uno, change the delay at the end of the heartbeat function to 40 ms from 20 ms before uploading. Older boards do not need this change. Before uploading make sure that nothing is plugged into the Arduino. Anytime you want to program an ATtiny with an Arduino, the ArduinoISP sketch has to be on the Arduino.

```
void heartbeat() {
  if (hbval > 192) hbdelta = -hbdelta;
  if (hbval < 32) hbdelta = -hbdelta;
  hbval += hbdelta;
  analogWrite(LED_HB, hbval);
  delay(20);
}
```

After uploading the script it is time to program the ATtiny. For this example I will be using the ATtiny85.

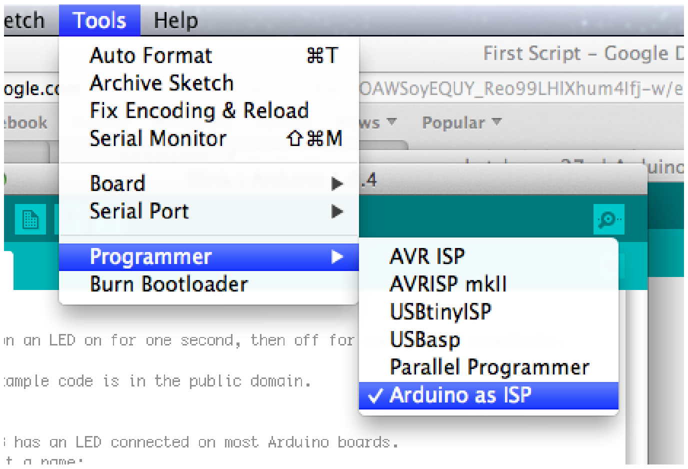
Plug the ATtiny into the shield. The direction of the chips can be noted by the indent in the body or little hole in ATtiny85 which corresponds to the top of the chip. The top of the chip must always point to the left (the outline of the board on the shield shows a corresponding divot for the ATtiny84 and 2313).

Next, open up the blink sketch from examples>basics>blink. Change the led at the top of the script from 13 to 4. Change the board that you are programming from the Arduino Uno to ATtiny85 1MHz. All of the ATtiny chips are by default set to 1 MHz. We will talk about how to change the clock speed with bootloaders in a few moments.

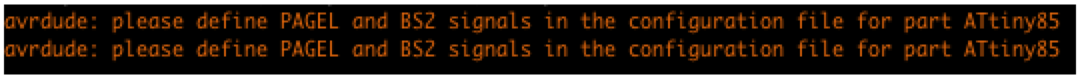

Arduino File Edit Sketch Tools Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Board
Serial Port
Programmer
Burn Bootloader
FAVORITES
All My Fil
Blink §
ATtiny84 @ 16 MHz (external crystal; 4.3 V BOD)
ATtiny84 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny84 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny44 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny44 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny24 @ 16 MHz (external crystal; BOD disabled)
ATtiny85 @ 16 MHz (external crystal; 4.3 V BOD)
ATtiny85 @ 16 MHz (internal PLL; 4.3 V BOD)
ATtiny85 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny85 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny45 @ 8 MHz
ATtiny45 @ 1 MHz
ATtiny85 @ 128 KHz (watchdog oscillator; 1.8 V BOD)
ATtiny25 @ 8 MHz
ATtiny25 @ 1 MHz
ATtiny4313 @ 8 MHz
ATtiny4313 @ 1 MHz
ATtiny2313 @ 8 MHz
ATtiny2313 @ 1 MHz
✓ Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
COM1

```
/*
  Blink
  Turns on an LED on for one second, then off for
 
  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino bo
// give it a name:
int led = 4;

// the setup routine runs once when you press rese
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forev
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off b
  delay(1000);               // wait for a second
}
```

Then select programmer “Arduino as ISP”



Now you can upload the script. The ATtiny should blink on and off. You will get an error message that looks like this, but do not worry everything is fine:



```
avrdude: please define PAGEL and BS2 signals in the configuration file for part ATtiny85
avrdude: please define PAGEL and BS2 signals in the configuration file for part ATtiny85
```

Using an Arduino Mega

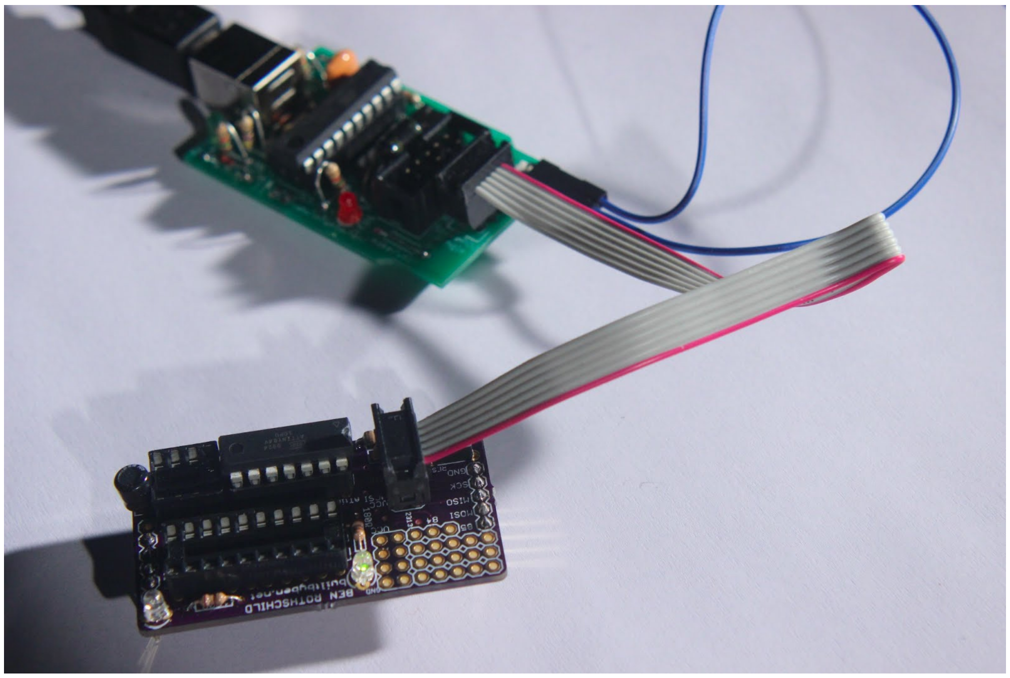
If you are using a board other than an Arduino Uno such as an Arduino Mega you cannot plug the Chipper Shield directly into the Arduino. Instead run jumper wires from MISO,MOSI,RESET,SCK,GND,VCC on the 6 pin header (each pin is labeled) on the Chipper Shield to the appropriate pin on the Arduino board.

For example, at the top of the ArduinoISP sketch it lists the corresponding Arduino Mega pins. Then, while the Arduino has the ArduinoISP sketch on it, upload your program like you would on the Arduino Uno. Select the ATtiny board and change the pin numbers.

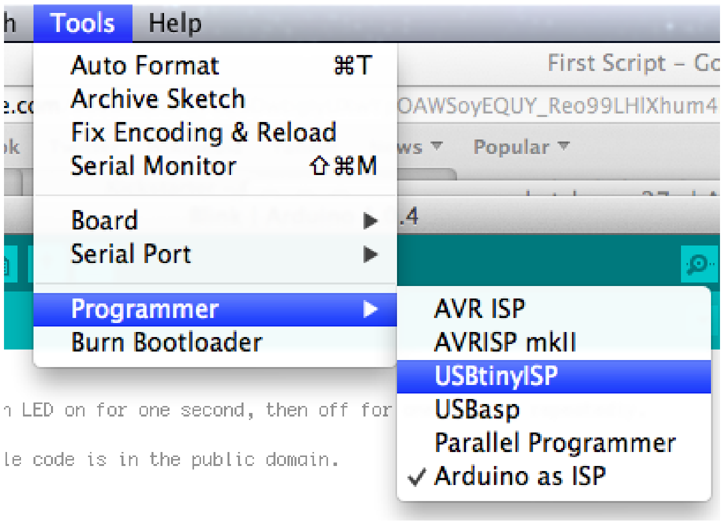
```
// This sketch turns the Arduino into a AVRISP
// using the following arduino pins:
//
// pin name:    not-mega:         mega(1280 and 2560)
// slave reset: 10:               53
// MOSI:        11:               51
// MISO:        12:               50
// SCK:         13:               52
//
```

AVR PROGRAMMER

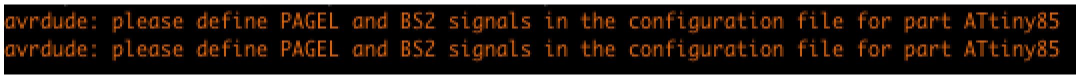
Here I will be using an USBtinyISP. Plug the 6 pin connector into the 6 pin header on the Chipper Shield so that the ribbon cable is pointing to the left of the Chipper (away from the chips).



Then select USBtinyISP under tools>programmer. Next open up the examples>basics>digital and change the led from 13 to 4. Plug the ATtiny85 into the shield so that the divot on the ATtiny is facing toward the 6 pin connector on the board and upload the script.



If you get this error message, you can ignore it:

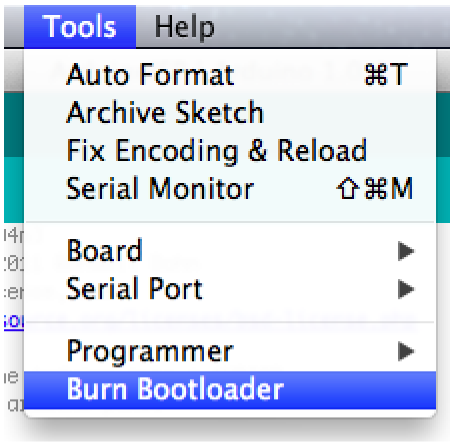


```
avrdude: please define PAGEL and BS2 signals in the configuration file for part ATtiny85
avrdude: please define PAGEL and BS2 signals in the configuration file for part ATtiny85
```

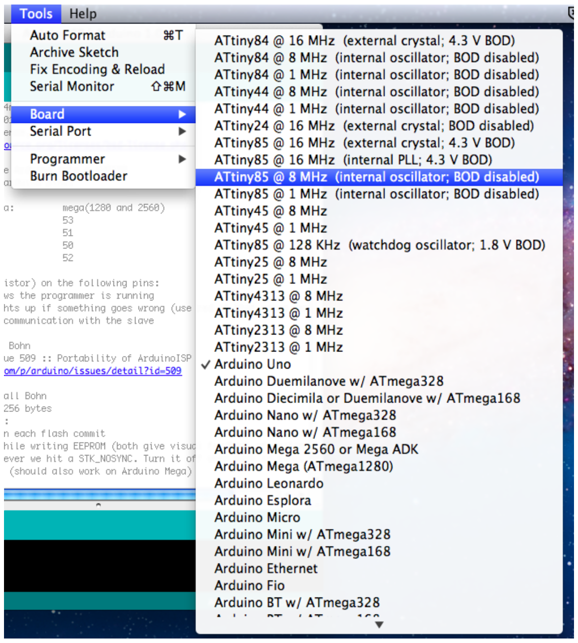
Burning Bootloaders

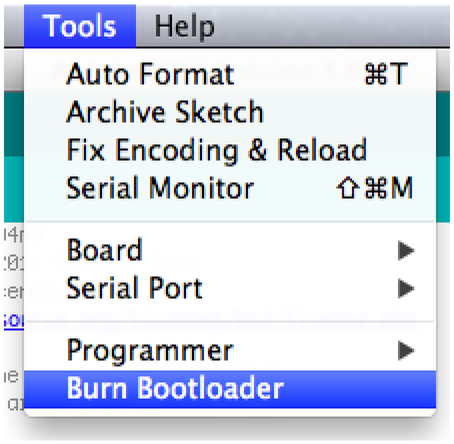
This can be done either with an Arduino board or an AVR programmer. If you are using an Arduino, configure the Arduino with the ArduinoISP sketch and then select Arduino as ISP as the programmer.

With the AVR programmer plug it into the 6 pins on the Chipper shield and select the appropriate programmer under tools>program. In this case we are using USBtinyISP.



Next, select the board that you wish to configure. In this example we are using an ATtiny85 which by default runs at 1 MHZ. We want to configure it to run on its internal 8MHz clock. You can also run it at 16 MHz with an external clock.

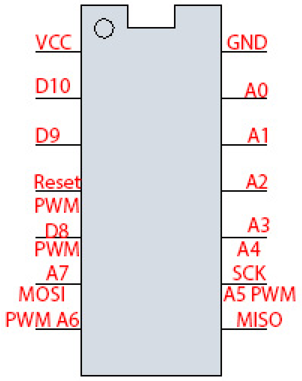

Tools
Help
Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Serial Monitor ⇧⌘M
Board
Serial Port
Programmer
Burn Bootloader
ATtiny84 @ 16 MHz (external crystal; 4.3 V BOD)
ATtiny84 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny84 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny44 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny44 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny24 @ 16 MHz (external crystal; BOD disabled)
ATtiny85 @ 16 MHz (external crystal; 4.3 V BOD)
ATtiny85 @ 16 MHz (internal PLL; 4.3 V BOD)
ATtiny85 @ 8 MHz (internal oscillator; BOD disabled)
ATtiny85 @ 1 MHz (internal oscillator; BOD disabled)
ATtiny45 @ 8 MHz
ATtiny45 @ 1 MHz
ATtiny85 @ 128 KHz (watchdog oscillator; 1.8 V BOD)
ATtiny25 @ 8 MHz
ATtiny25 @ 1 MHz
ATtiny4313 @ 8 MHz
ATtiny4313 @ 1 MHz
ATtiny2313 @ 8 MHz
ATtiny2313 @ 1 MHz
✓ Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328

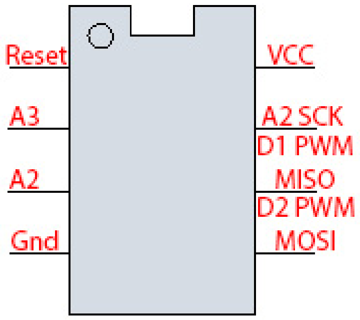


Select ATTiny 8MHz from the list of boards and then select tools>Burn Bootloader. Everything else is taken care of and in a few moments your ATtiny85 will be configured to run at 8MHZ. This is necessary when timing is a crucial part of your program. For example, if you want to run servos off of the ATtiny.

Wondering what pins correspond to what?

Here are the pinouts for the ATtiny85 and 84


VCC
D10
D9
Reset PWM D8
PWM A7 MOSI
PWM A6
GND
A0
A1
A2
A3 A4 SCK
A5 PWM MISO


Reset
A3
A2
Gnd
VCC
A2 SCK
D1 PWM MISO
D2 PWM MOSI