



[Example Projects](#)
[Workshops](#)

[Actuators](#)
[Connections](#)
[Power](#)
[Sensors](#)
[Traces](#)

[Circuits and Code](#) [Wireless](#)

[Conductive Materials](#)
[Non-Conductive Materials](#)

[Tools](#)

[Techniques](#)

Circuits and Code

- [Arduino Overview](#)
- [Arduino as Bluetooth HID](#)
- [ATtiny: 7-Segment Display](#)
- [ATtiny: Capacitive Sensing](#)
- [ATtiny: Programming](#)
- [ATtiny Serial & Wireless Boards!](#)
- [ATtiny: Soft Serial](#)
- [ATtiny: Sound](#)
- [ATtiny: Soft Fade](#)
- [Multiplexed Matrix](#)
- [Controlling EL Panel and EL Wire](#)
- [EL panel/wire inverter hack](#)
- [Lasercut Fabric Circuits and Traces](#)
- [Heat Controlling Circuit](#)
- [LED with Light Sensor](#)
- [Lilypad XBee Shield](#)
- [Pull-up Resistors](#)
- [RGB Colour Fade](#)
- [simple heat circuit](#)
- [Solar Powered Motor Circuit](#)
- [Sound Amplifying Circuits](#)
- [Teensy as HID Device](#)
- [Volume Detection](#)
- [Visualization: 2x2 Matrix](#)
- [Visualization: Drawing](#)
- [Visualization: Graph](#)
- [Visualization: Pressure Sensor Matrix](#)
- [Visualization: Touchpad](#)
- [Xbee lilypad shield PCB](#)
- [About](#)
- [Newsletter](#)
- [Shopping Local](#)

SEARCH



Content by [Mika Satomi](#) and [Hannah Perner-Wilson](#)

The following institutions have funded our research and supported our work:

From 2013-2014 Mika is a guest professor at the [eLab](#) at [Kunsthochschule Berlin-Weissensee](#)

Design Research Lab

From July - December 2013 Hannah was a researcher at the [UdK's Design Research Lab](#)

Smart Textiles Design Lab

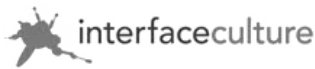
From 2010-2012 Mika was a guest researcher in the [Smart Textiles Design Lab](#) at [The Swedish School of Textiles](#)

high-low Leah

From 2009 - 2011 Hannah was a graduate student in the [MIT Media Lab's High-Low Tech](#) research group led by [Leah Buechley](#)

distance lab

In 2009 Hannah and Mika were both research fellows at the [Distance Lab](#)



Between 2003 - 2009 Hannah and Mika were both students at [Interface Cultures](#)



We support the Open Source Hardware movement. All our own designs published on this website are released under the [Free Cultural Works definition](#)

[Circuits and Code](#)

ATtiny: Programming

This post is a summary that covers how to turn your [arduino](#) board into an ISP programmer and use it to program an ATtiny85 or 45 8-pin microcontroller.

The instructions in this post are based on the following two tutorials written by David Mellis:

Arduino board as ATtiny programmer >> <http://hlt.media.mit.edu/?p=1706>

Programming an ATtiny w/ Arduino 1.0.1 >> <http://hlt.media.mit.edu/?p=1695>

If you get fed up with all the wires, see how to make your own ATtiny programming shield for Arduino:

>> <http://www.kobakant.at/DIY/?p=3996>

Turning your Arduino into an ISP Programmer

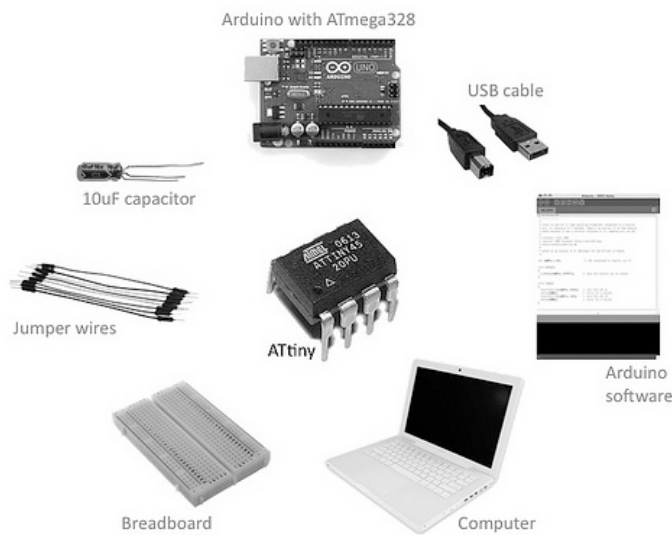
What is an ISP (In System Programmer)???

The answer >> http://en.wikipedia.org/wiki/In-system_programming

1 Materials and Tools

- Laptop with the Arduino software version 1.0.1 or 0022 installed
- Arduino Uno or Duemilanove (*with an ATmega328, not an older board with an ATmega168!*)
- USB cable
- ATtiny45 or ATtiny85 ([Sparkfun](#), [DigiKey](#), [RS](#))

- 10 uF capacitor
- Breadboard
- Jumper wires



2 Download and Save the ATtiny Folder

- Download the “ATtiny” folder from this GitHub repository:
 >> <https://github.com/damellis/attiny/tree/Arduino1>
- In your Arduino sketchbook folder create a new sub-folder called “hardware”
- Put the “ATtiny” folder inside this “hardware” folder
- You should end up with folder structure like this: “Documents > Arduino > hardware > attiny”

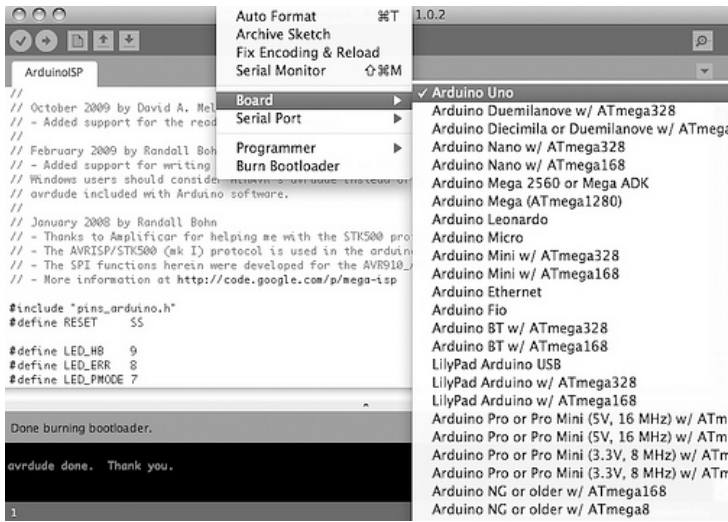


- Quit and restart the Arduino software
- Look inside the “Tools > board” menu and you should see the ATtiny entries

ATtiny45 (internal 1 MHz clock)
 ATtiny45 (internal 8 MHz clock)
 ATtiny45 (external 20 MHz clock)
 ATtiny85 (internal 1 MHz clock)
 ATtiny85 (internal 8 MHz clock)
 ATtiny85 (external 20 MHz clock)
 ATtiny44 (internal 1 MHz clock)
 ATtiny44 (internal 8 MHz clock)
 ATtiny44 (external 20 MHz clock)
 ATtiny84 (internal 1 MHz clock)
 ATtiny84 (internal 8 MHz clock)
 ATtiny84 (external 20 MHz clock)

3 Upload “ArduinoISP” to your Arduino

- Open “ArduinoISP” sketch from “Examples” folder
- Select “Arduino Uno” from the “Tools > Board” menu
- Upload sketch



Using your Arduino ISP to Program an ATtiny Microcontroller

1 Wiring your ISP connection

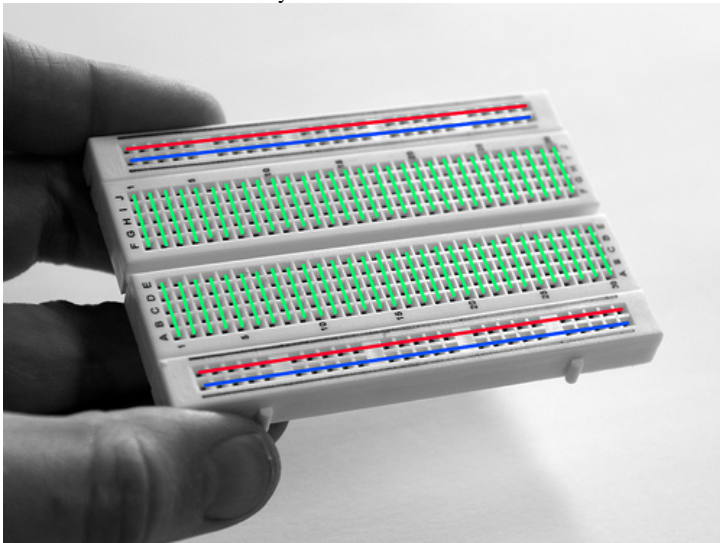
ATtiny --- Arduino

Pin PB2 (SCK) --- Pin 13
 Pin PB1 (MISO) --- Pin 12
 Pin PB0 (MOSI) --- Pin 11
 Pin PB5 (Reset) --- Pin 10
 Plus (VCC) --- +5V
 Minus (GND) --- GND

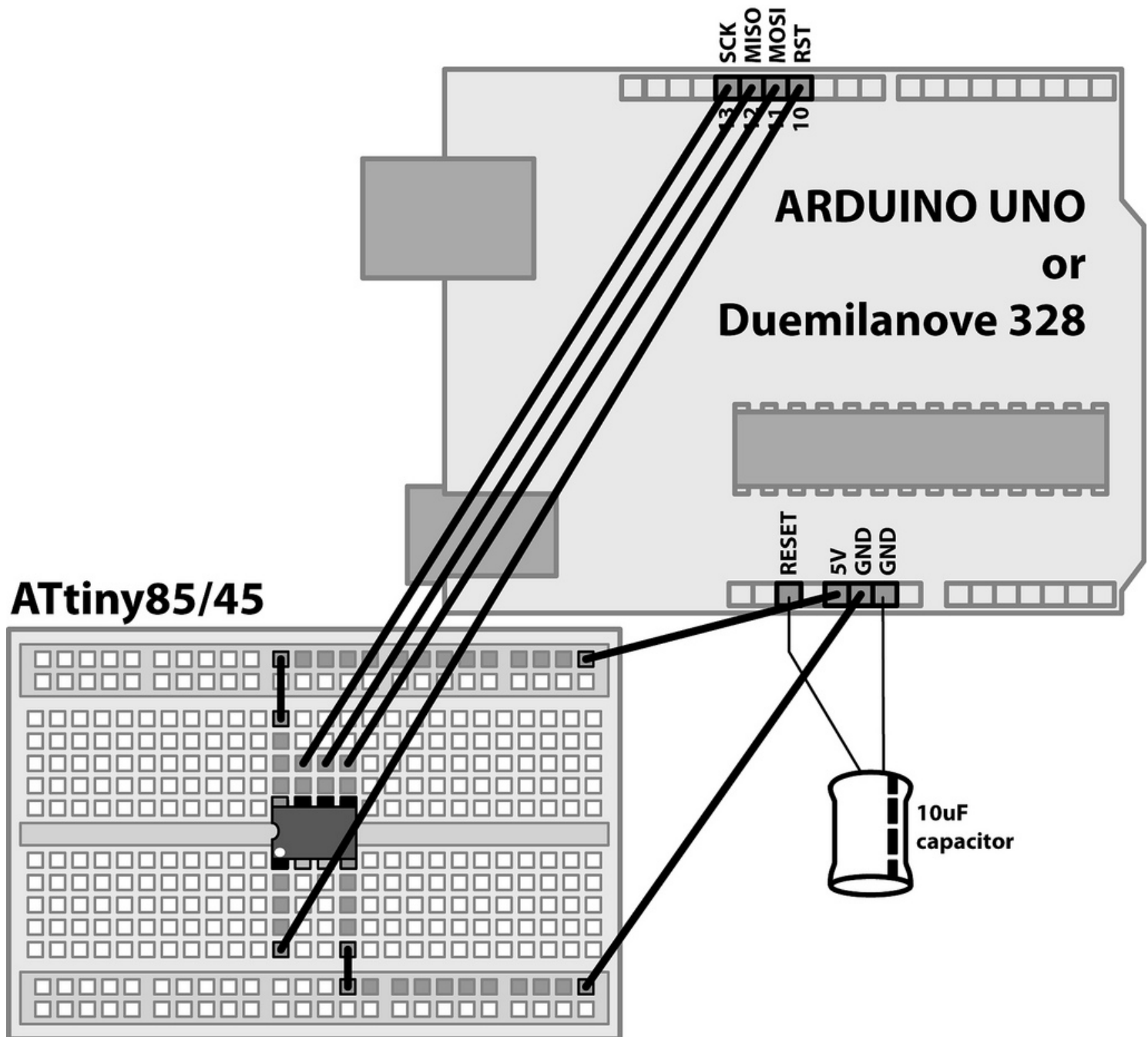
10uF Capacitor:

Arduino pins: RESET ---||--- GND

How a breadboard is internally connected:



How to connect the Arduino and the ATtiny using the breadboard, jumper wires and one 10uF capacitor:

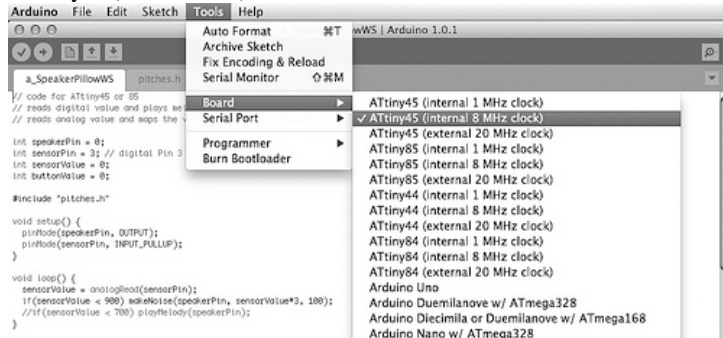


2 Burn Bootloader

To set the clock speed of your ATtiny to be faster (8Mhz) than the default 1 MHz.

- Select "ATtiny45 (8 MHz)" from the "Tools > Board" menu
- Select "Arduino as ISP" from the "Tools > Programmer" menu
- Select "Burn Bootloader" from the "Tools" menu

"ATtiny45 (8 MHz clock)":



"Arduino as ISP":

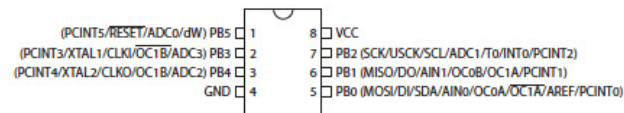


“Burn Bootloader”:



3 Writing a Sketch for the ATtiny

Download the ATtiny25/45/85 datasheet >> <http://at.rs-online.com/web/p/microcontroller/6962339/>



The following Arduino commands are supported for the ATtiny:

```
pinMode()
digitalWrite()
digitalRead()
analogRead()
analogWrite()
shiftOut()
pulseIn()
millis()
micros()
delay()
delayMicroseconds()
SoftwareSerial
```

4 Uploading a Sketch to the ATtiny

- Open the sketch you want to upload to ATtiny
- Select “ATtiny45 (8 MHz)” from the “Tools > Board” menu”
- Select “Arduino as ISP“ from the “Tools > Programmer” menu
- Upload sketch
- The following error message is **okay**:

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 ATtiny8

Notes

- Once you have added parts to your circuit that connect to the programming pins of the ATtiny, you may need to disconnect these parts before uploading a new program.

- Once you have programmed your ATtiny, you may need to remove the programming connections in order for your circuit to function correctly.
- Make sure you have a compatible version of the Arduino software installed. At the time of writing this post version 1.0.1 worked, but version 1.0 had bugs and version 1.0.2 was not working at all.
- Remember to remove the capacitor when uploading the Arduino ISP sketch to the Arduino Uno or Duemilanove and to put the capacitor back for burning the bootloader and uploading any sketches to your ATtiny.
- Declare IN- and OUTPUTS referring to the pin's port number: *pinMode(PB#, INPUT);*
- When reading an analog input, refer to the pin's ADC number: *analogRead(ADC#);*
- To set the internal [pull-up resistors](#) write the following into the setup() function:

```
pinMode(4, INPUT);
digitalWrite(4, HIGH);
```

Because we have declared pin 4 to be an INPUT, writing HIGH to the pin will set the internal pull-up.

Checklist:

Turning your Arduino into an ISP Programmer

No capacitor!

1. Download the Arduino software and install it
2. Download the ATtiny folder and save it in a "hardware" folder in your Arduino sketch folder
3. Restart Arduino
4. Open "ArduinoISP" sketch from "Examples" folder
5. Select "Arduino Uno" from the "Tools > Board" menu
6. Select "/dev/tty/usbserial###" from the "Tools > Serial Port" menu
7. Upload sketch

Using your Arduino ISP to Programming an ATtiny Microcontroller

Add capacitor and programming connections!

1. Select "ATtiny45 (8 MHz)" from the "Tools > Board" menu
2. Select "Arduino as ISP" from the "Tools > Programmer" menu
3. Select "Burn Bootloader" from the "Tools" menu (yes capacitor)
4. Open the sketch you want to upload to the ATtiny
5. Upload sketch (yes capacitor)

The following error message is **okay**:

```
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 ATtiny8
```

3 Comments so far

1. [Usando arduino one como programadora ISP para chips ATtiny | jos.mx](#) on March 27th, 2013

[...] <http://www.kobakant.at/DIY/?p=3742> [...]

2. [Blowing Bootloaders into Atmel Chips | Scargill's Tech Weblog](#) on September 14th, 2013

[...] <http://www.kobakant.at/DIY/?p=3742> [...]

3. [HOW TO GET WHAT YOU WANT](#) on November 27th, 2013

[...] <http://www.kobakant.at/DIY/?p=3742> >> <http://www.kobakant.at/DIY/?p=3996> >> <http://hlt.media.mit.edu/?p=1706> >> [...]

Leave a comment

 name (required)

 email (will not be shown) (required)

 website