

high-low tech

- [PROJECTS](#)
- [PEOPLE](#)
- [PUBLICATIONS](#)
- [TUTORIALS](#)

[LilyTiny Sparkle Bracelet](#)  
[Paper Lamps with Codeable Objects](#)  
[Paper circuits w/ copper tape](#)  
[Using the Codeable Objects Processing Library](#)  
[LilyTiny Monster](#)  
[Simple Arduino audio samples](#)  
[Using the TinyProgrammer](#)  
[Arduino board as ATtiny programmer](#)  
[Programming an ATtiny w/ Arduino 1.0](#)  
[Microcontroller circuit with copper tape](#)  
[Simple circuit with copper tape](#)  
[Electronic Origami Flapping Crane](#)  
[LED Dragon Kite](#)  
[Light-up Postcard](#)  
[Paper Speakers](#)  
[Salt and Vinegar Etching](#)  
[Electronics and Textiles \(e-textiles\)](#)  
[Programming an ATtiny w/ Arduino 0022](#)  
[Painted Circuits](#)  
[Data Logging with uLog Module](#)

- [MATERIALS](#)
  - [NEWS & EVENTS](#)
  - [WORKSHOPS](#)
  - [MIT CLASSES](#)
  - [ADMISSIONS](#)
- [NEWSLETTER](#)

## [Programming an ATtiny w/ Arduino 0022](#)

This tutorial shows you how to program an ATtiny45 or ATtiny85 microcontroller using the Arduino software and hardware. The ATtiny45 and ATtiny85 are small (8-leg), cheap (\$2-3) microcontrollers that are convenient for running simple programs.

**This tutorial is obsolete! See the [updated version of this tutorial](#) (with support for Arduino 1.0).**

They are almost identical, except that the ATtiny85 has twice the memory of the ATtiny45 and can therefore hold more complex programs. We like to use both of them with paper circuits and other craft electronics. To program them, we'll use a port of the Arduino core libraries created by [Alessandro Saporetti](#) and slightly modified by HLT.

## Materials and Tools

For this tutorial, you'll need:

- [Arduino Uno or Duemilanove](#) (w/ an ATmega328, not an older board with an ATmega168)
- [ATtiny45](#) or [ATtiny85](#) (8-pin DIP package)
- a 10 uF capacitor (e.g. [from Sparkfun](#) or [from Digi-Key](#))
- a breadboard
- jumper wires

For more information, see our [list of materials and parts](#) and our [list of prototyping supplies](#).

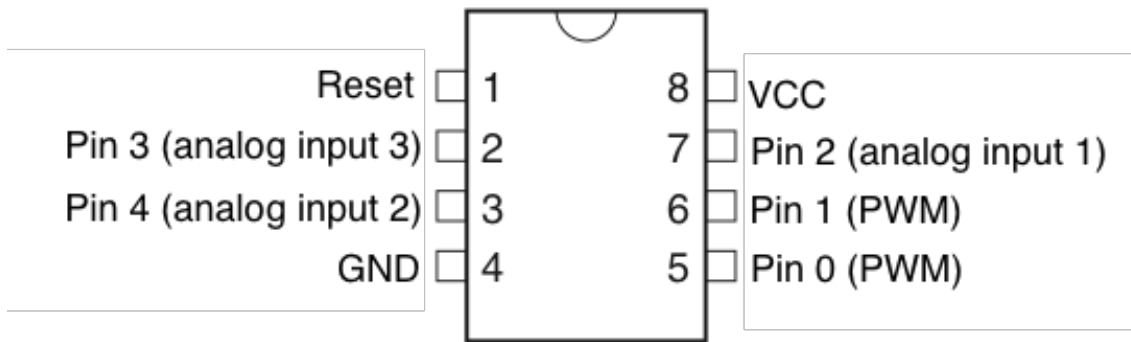
## Software

You'll need the Arduino software (version 0022) and a plugin to add support for the ATtiny45/85. Installation instructions for the Arduino software are available [for Windows](#) and [for Mac OS X](#).

Download: [Arduino software](#), [attiny45\\_85.zip](#)

## The ATtiny45 / ATtiny85 Microcontroller

Here's the pinout of the ATtiny45 and ATtiny85:



## Installing ATtiny45 / ATtiny85 support in Arduino

- Download: [attiny45\\_85.zip](#)
- Locate your Arduino sketchbook folder (you can find its location in the preferences dialog in the Arduino software)
- Create a new sub-folder called “hardware” in the sketchbook folder.
- Copy the attiny45\_85 folder from the attiny45\_85.zip to the hardware folder.
- Restart the Arduino development environment.

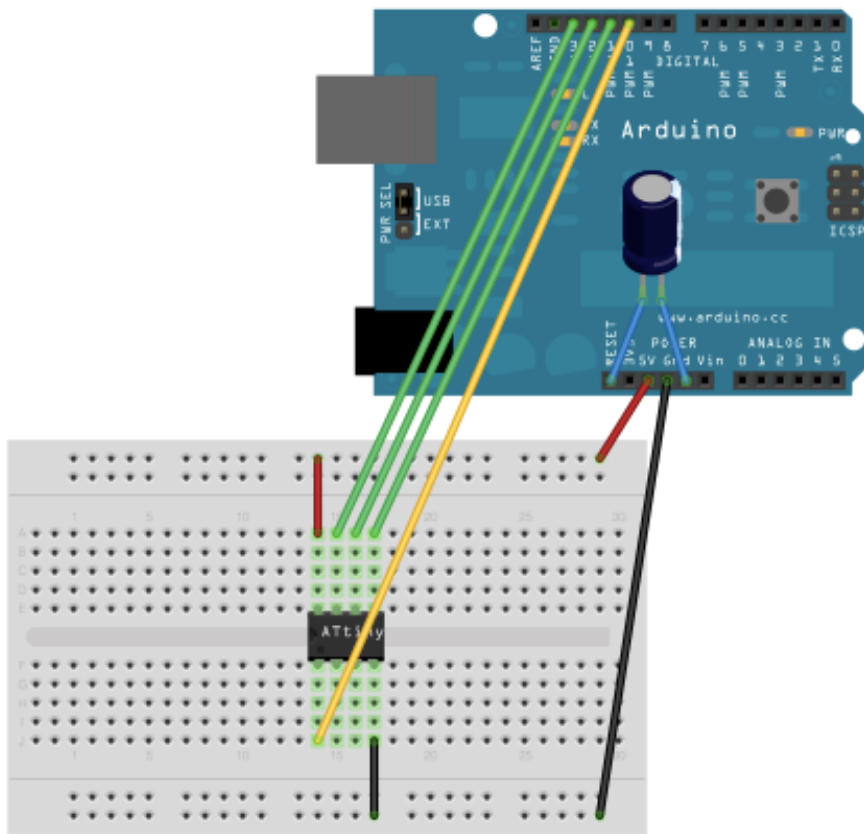
## Turning the Arduino board into a programmer

We’re going to use the Arduino board to program the ATtiny45 or ATtiny85. First, we’ll need to turn the Arduino board into an “in-system programmer” (ISP). To this:

- Run the Arduino development environment.
- Open the ArduinoISP sketch from the examples menu.
- Select the board and serial port that correspond to your Arduino board.
- Upload the ArduinoISP sketch.

## Connecting the Arduino board and the ATtiny

We’ll connect the Arduino board to the ATtiny as shown in the following diagram. Use the dot in the corner of the ATtiny to orient it properly. We’ll also connect a 10 uF capacitor between reset and ground on the Arduino board as shown in the diagram (the stripe on the capacitor that’s marked with a negative sign (-) should go to ground). The capacitor prevents the Arduino board from resetting (which starts the bootloader), thus ensuring that the Arduino IDE talks to the ArduinoISP (not the bootloader) during the upload of sketches. (The capacitor is needed if you’re using an Arduino Uno, and might also be necessary for an Arduino Duemilanove.)



Pin connections:

- ATtiny Pin 2 to Arduino Pin 13 (or SCK of another programmer)
- ATtiny Pin 1 to Arduino Pin 12 (or MISO of another programmer)
- ATtiny Pin 0 to Arduino Pin 11 (or MOSI of another programmer)
- ATtiny Reset Pin to Arduino Pin 10 (or RESET of another programmer)

## Programming the ATtiny45 or ATtiny85

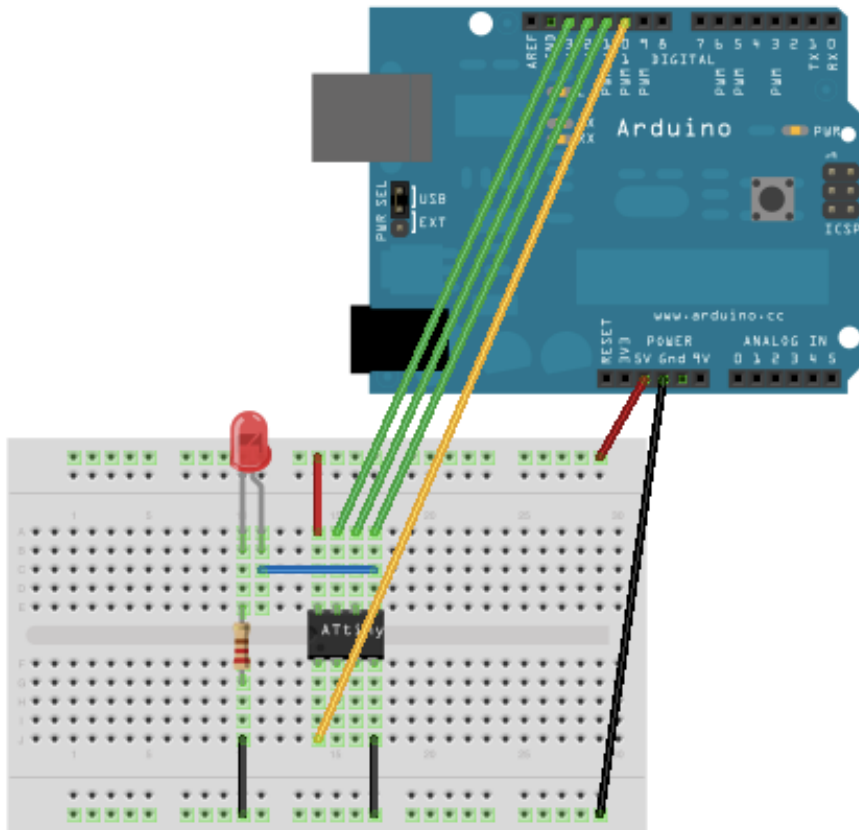
Next, we can use the Arduino as an ISP to upload a program to the ATtiny45/85:

- Open the Blink sketch from the examples menu.
- Change the pin numbers from 13 to 0.
- Select “ATtiny45 (w/ Arduino as ISP)” or “ATtiny85 (w/ Arduino as ISP)” from the Tools > Board menu (leave the serial port set to that of your Arduino board).
- Upload the sketch.

You’ll probably get the following message, but don’t worry, the upload should have worked anyway:

```
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
```

Now, connect an LED and see if it blinks.



## Reference

The following Arduino commands should be supported:

- [pinMode\(\)](#)
- [digitalWrite\(\)](#)
- [digitalRead\(\)](#)
- [analogRead\(\)](#)
- [analogWrite\(\)](#)
- [shiftOut\(\)](#)
- [pulseIn\(\)](#)
- [millis\(\)](#)
- [micros\(\)](#)
- [delay\(\)](#)
- [delayMicroseconds\(\)](#)

[High-Low Tech Group :: MIT Media Lab](#)