FABRICADEMY
textile and technology academy

#FABRICADEMY2021 TUTORIALS

SENSING & OUTPUTS
ATTINY TUTORIAL - PART 1

15-12-2021 | EMMA PARESCHI

Fabricademy | textile & technology academy 2021

Today:

- Intro to Attiny
- How to program an Attiny

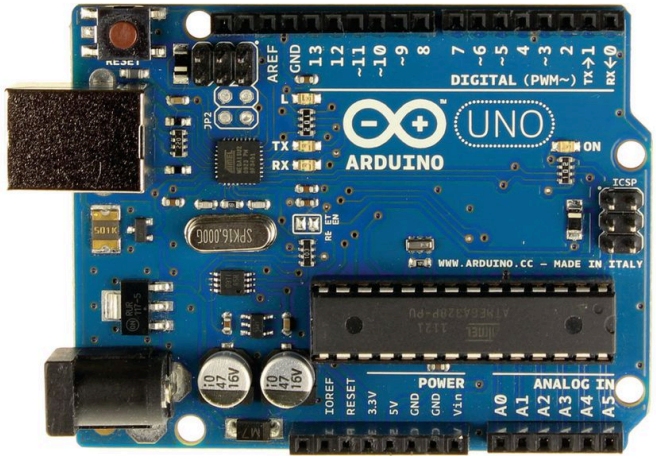
Circuits:

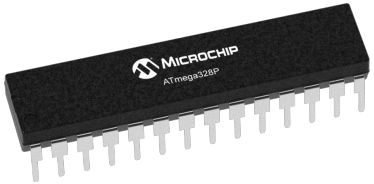
- Attiny, Led
- Attiny, Led and Switch
- Attiny, Neopixel and Switch
- (Attiny, Power load and Switch)

How to integrate the Attiny in your project

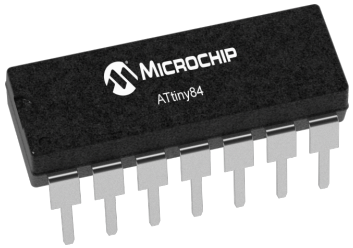
Intro to Attiny

AVR Microcontroller





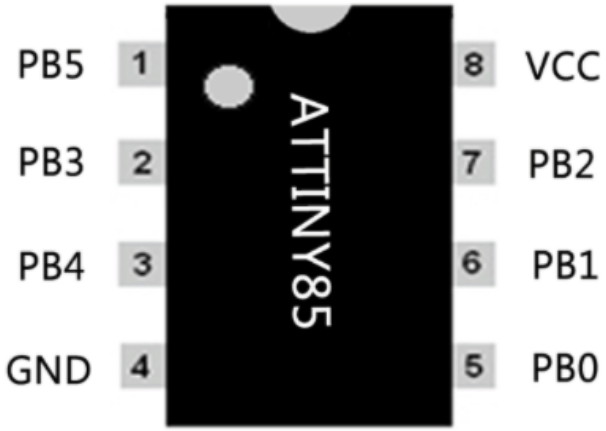
(Arduino Uno)
ATmega328



Attiny44
Attiny84

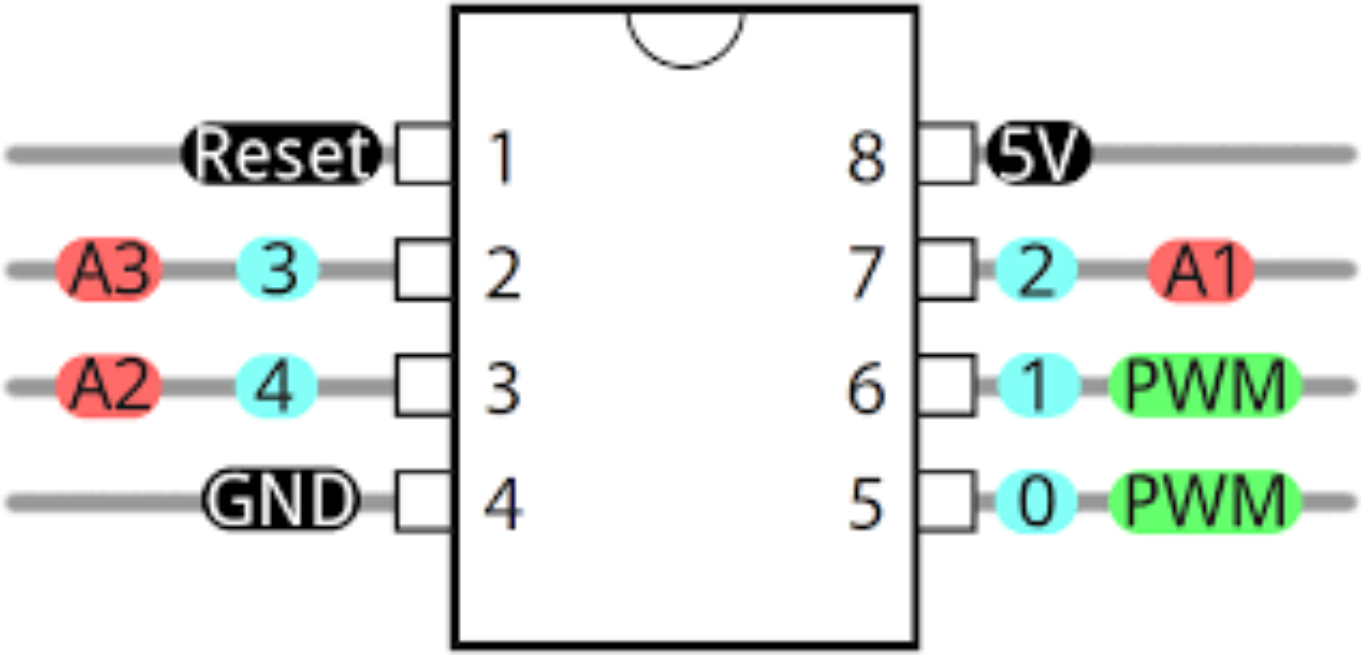
Attiny45
Attiny85

Attiny 45/85 - Pinout

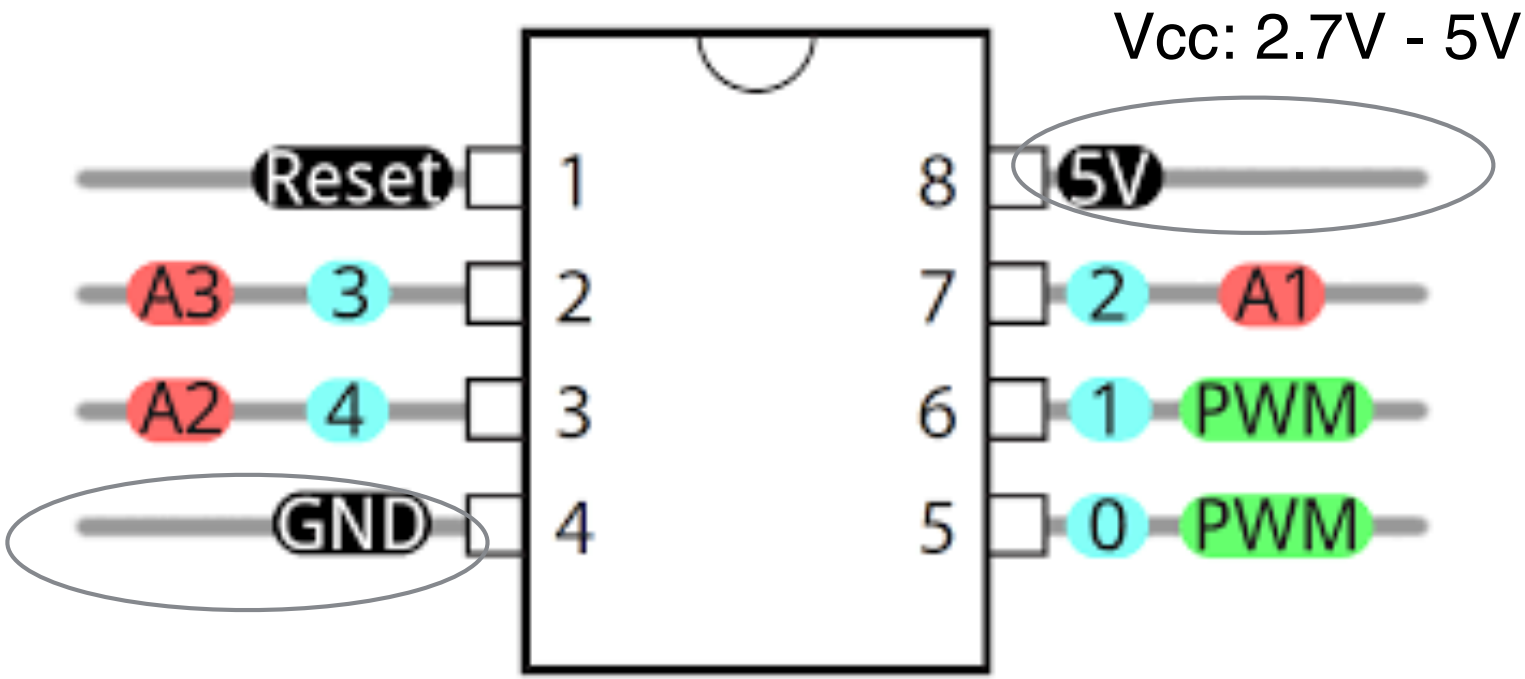


datasheet: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2586-AVR-8-bit-Microcontroller-ATtiny25-ATtiny45-ATtiny85_Datasheet.pdf

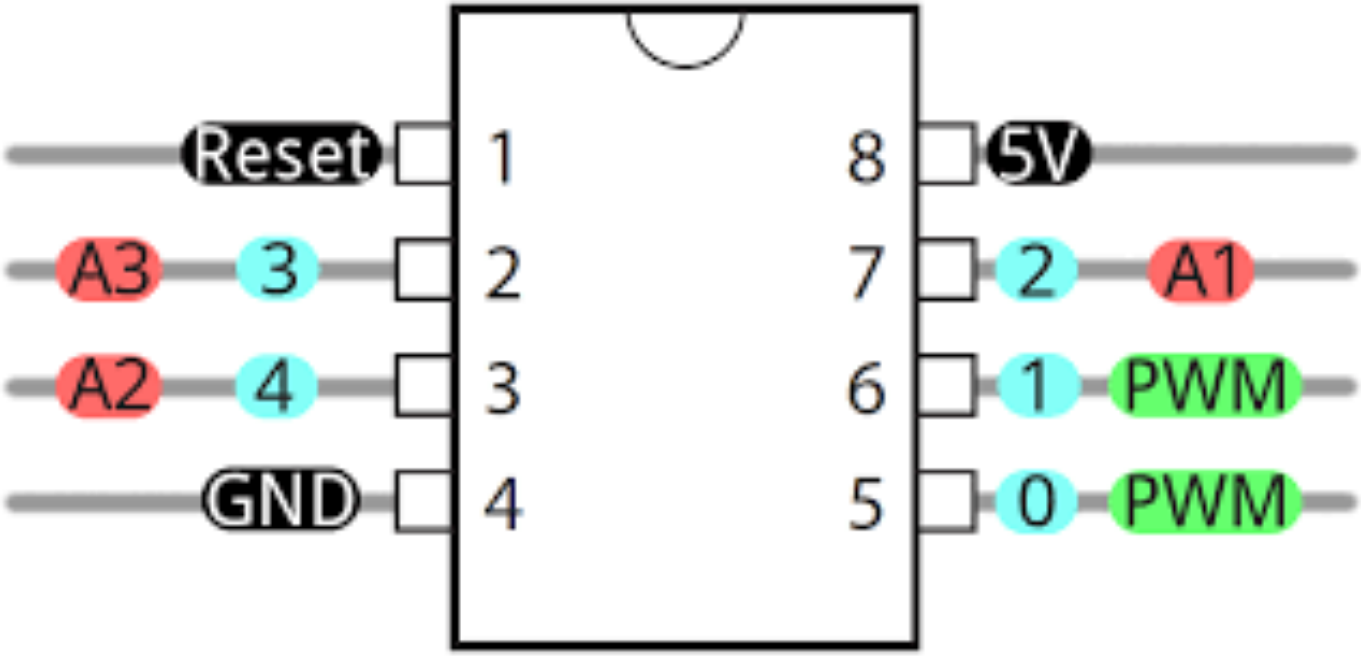
The number in this pinout are the official pin numbers.
Be careful! We won't use this numbering!!!!!!



Attiny 45/85 - Power pins



Attiny 45/85 - pinout



Digital pins → Read digital voltage: digital sensors; digitalRead(pin).
Set a digital voltage: On/OFF output devices, digitalWrite(pin, HIGH/LOW)

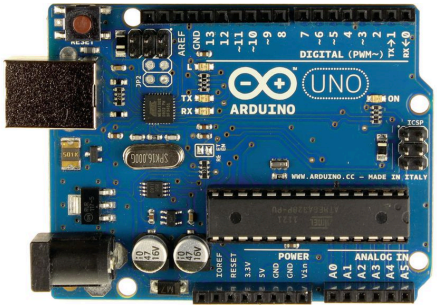
PWM pins → Set a kind of analog voltage: control the intensity of the output device, analogWrite(pin, 0/255)

Analog pins → Read an volatge: analog sensor; analogRead(pin)

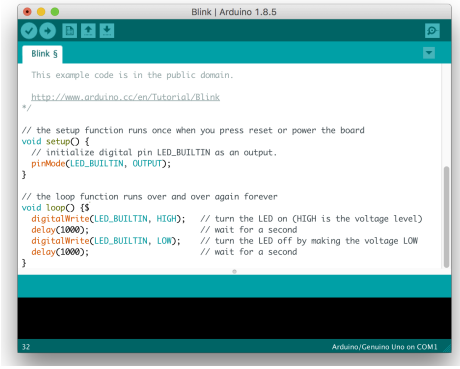
Blue and Red pin names-> numbers you have to use in Arduino IDE. We call this numbering Arduino numbers.

How to program the Attiny

How to program - AVR

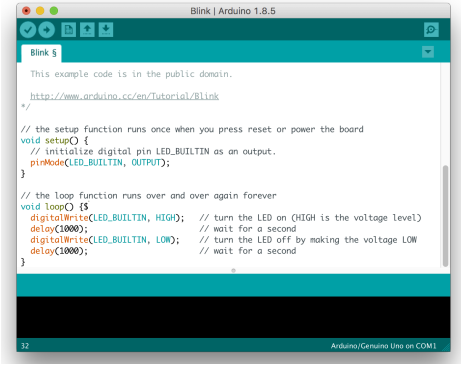


(Arduino Uno)
ATmega328



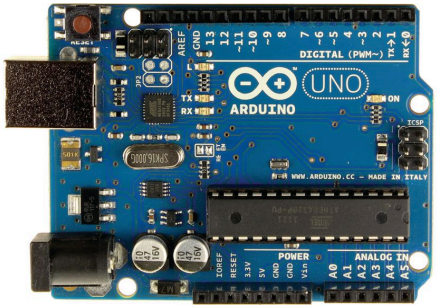
Arduino IDE

Attiny45
Attiny85



Arduino IDE

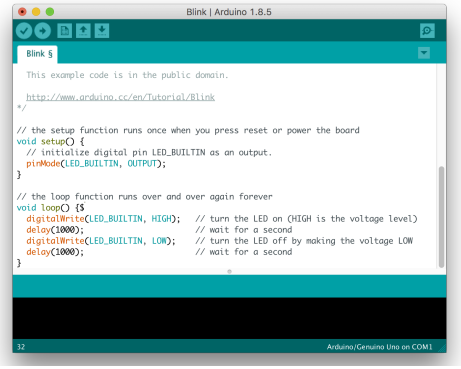
How to program - AVR



(Arduino Uno)
ATmega328

USB cable

Laptop

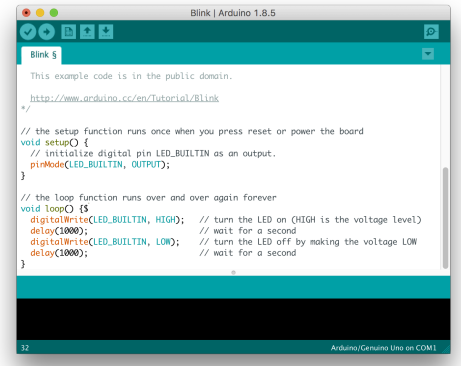


Arduino Uno

Attiny45
Attiny85

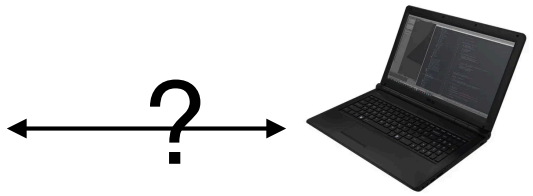
?

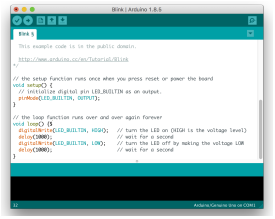
Laptop



Arduino Uno

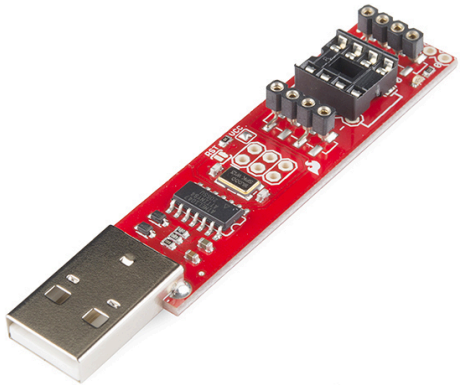
Hot to program - Attiny





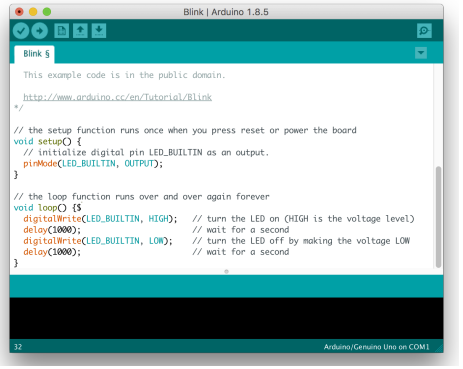
Option 1

Attiny45



Tiny AVR Programmer

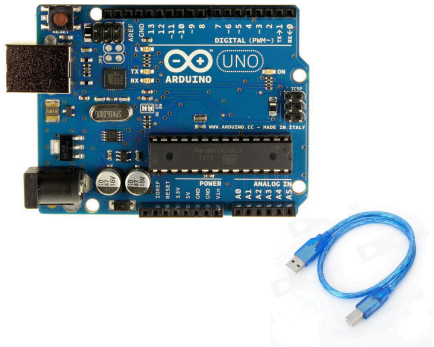
Laptop



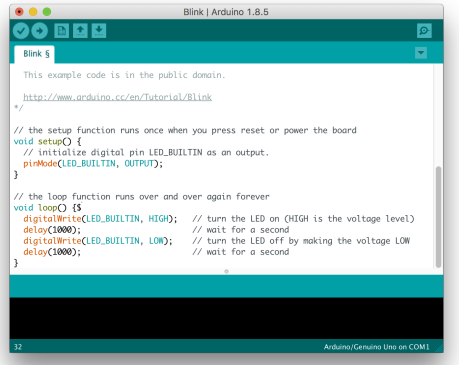
Arduino Uno

Option 2

Attiny45

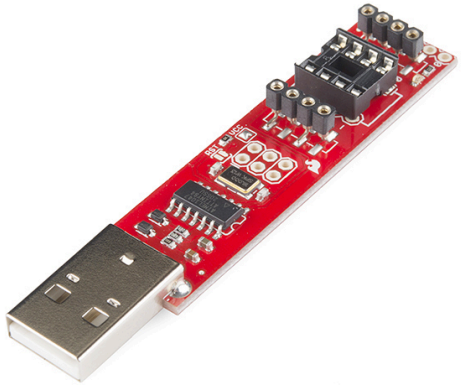


Laptop

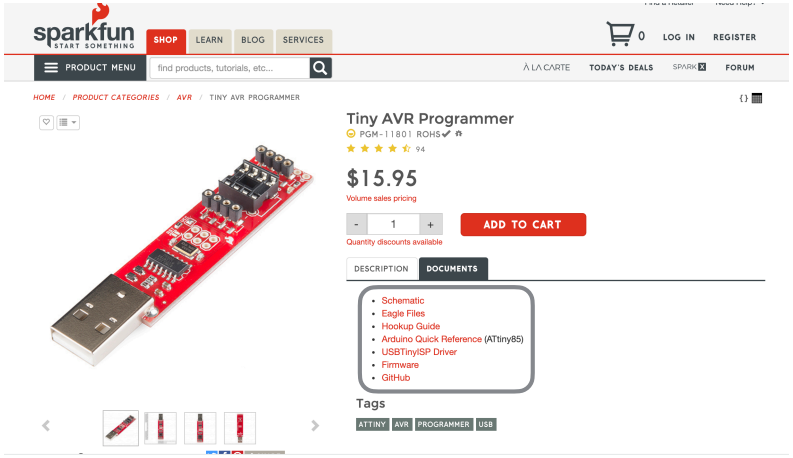


Arduino Uno

Tiny AVR Programmer



https://www.sparkfun.com/products/11801



Driver Installation

Before you can start using the Tiny AVR Programmer, you may need to set it up on your computer. If you're using a **Mac or Linux** machine, you **don't need to install drivers**. Just plug the board in, and skip to the Programming in Arduino page.

TINY AVR PROGRAMMER HOOKUP GUIDE - PROGRAMMING IN ARDUINO

If you're using any version of **Windows**, you've got a few steps to follow before you can join your Mac/Linux comrades. There are two sets of instruction for driver installation on this page. The first is the easiest, quickest method, and should work for most everyone. The second installation process is only required if the first one fails -- it takes a more manual approach to the driver installation.

Automatically Install the Drivers with Zadig

To begin, **plug the Tiny AVR Programmer into your computer**. Upon initially connecting the board, Windows will try to automatically install the drivers. Some computers may be lucky, but most will turn up with a message notifying you that the driver install failed.

Click the link below to download the Zadig software and drivers:

DOWNLOAD THE ZADIG USBTINY DRIVERS (ZIP)

Use your favorite unzipper to extract the ZIP file. Don't forget where you put the extracted folder!

Comments 21
View Paginated
Print

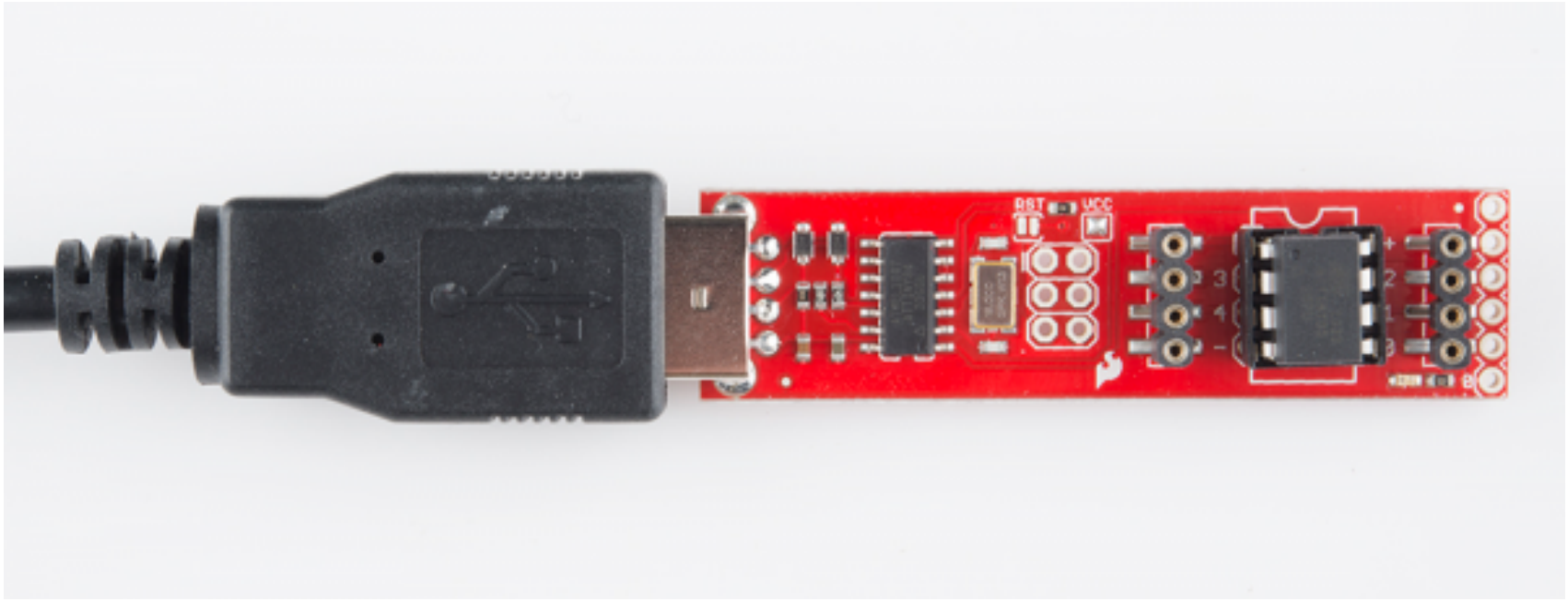
Tags

ARDUINO ATTINY HOOKUP PROGRAMMING

If you have a Windows the installation of the drivers could take some steps. Follow the instruction on this page.

https://learn.sparkfun.com/tutorials/tiny-avr-programmer-hookup-guide/?_ga=2.223970844.604127917.1608028904-1712614024.1598002523#automatic-install

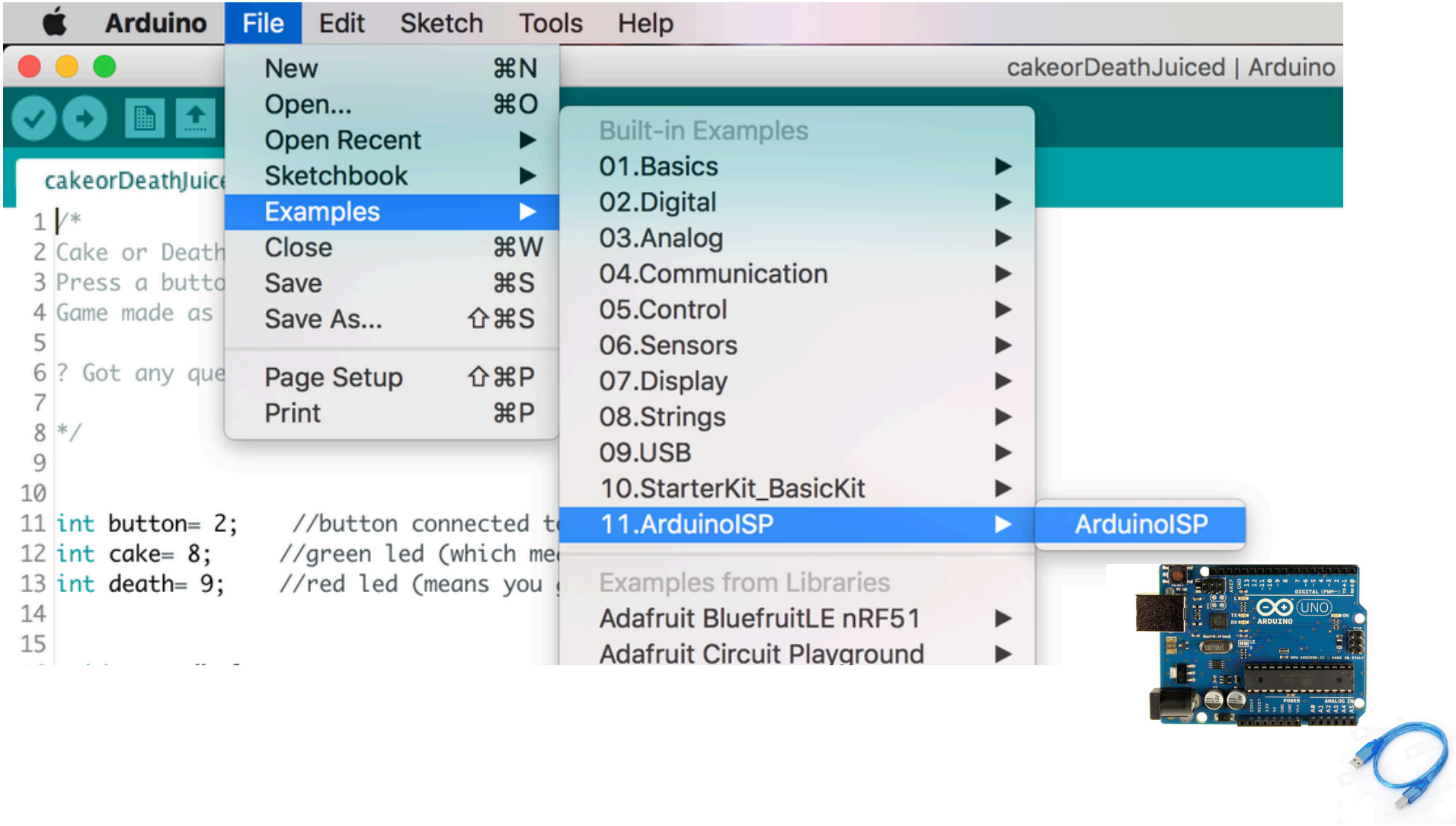
Tiny AVR Programmer - option 1



Insert the Attiny in the socket.
Be sure about the orientation

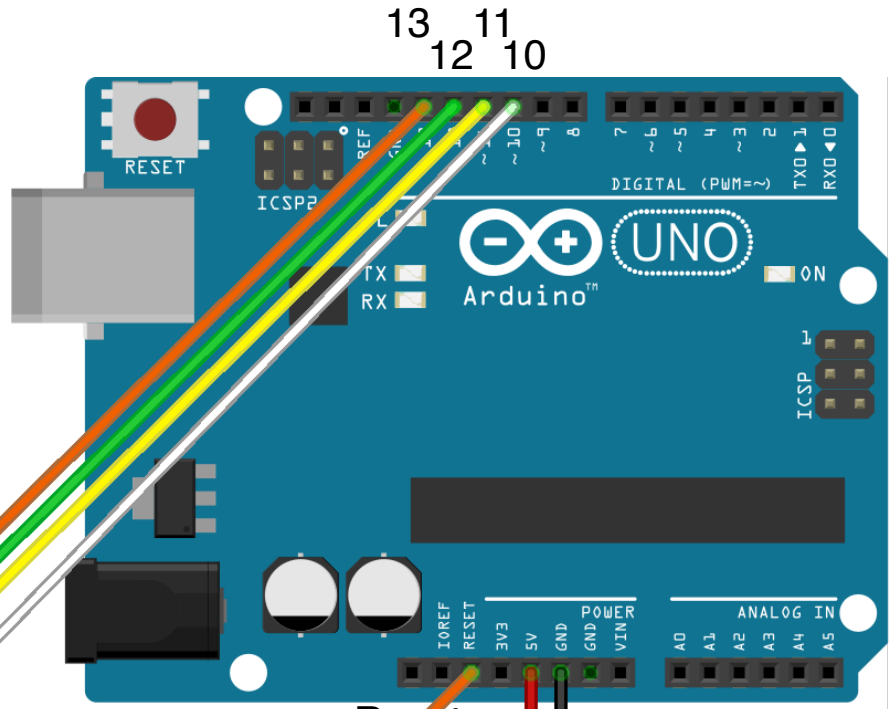
Arduino as Programmer - option 2

Transform Arduino UNO in programmer:
upload the right code (ArduinoISP) in your Arduino Uno

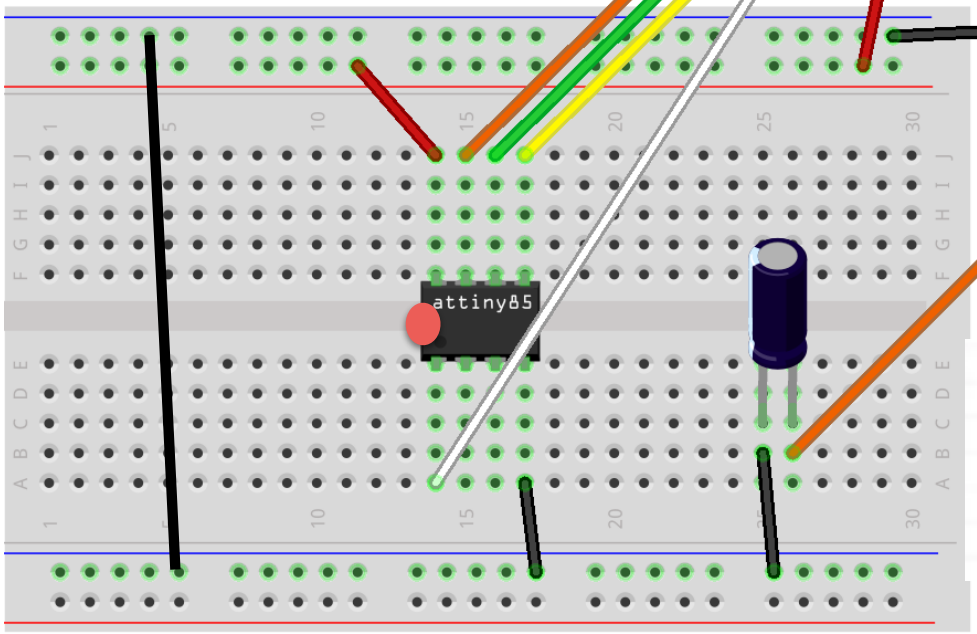


Arduino as Programmer

Properly connect the Attiny to the Arduino UNO Programmer.



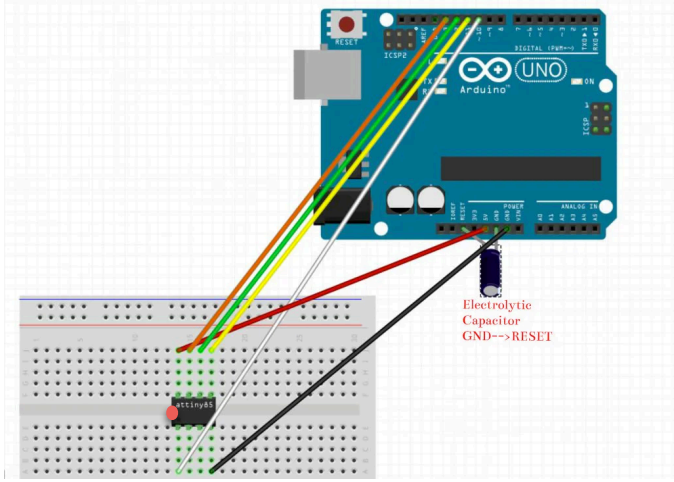
There is a little mark on the package, it indicates the orientation.

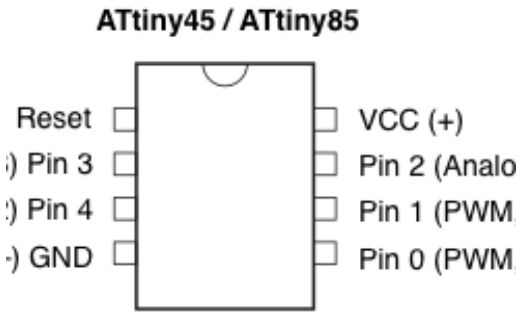


Maybe

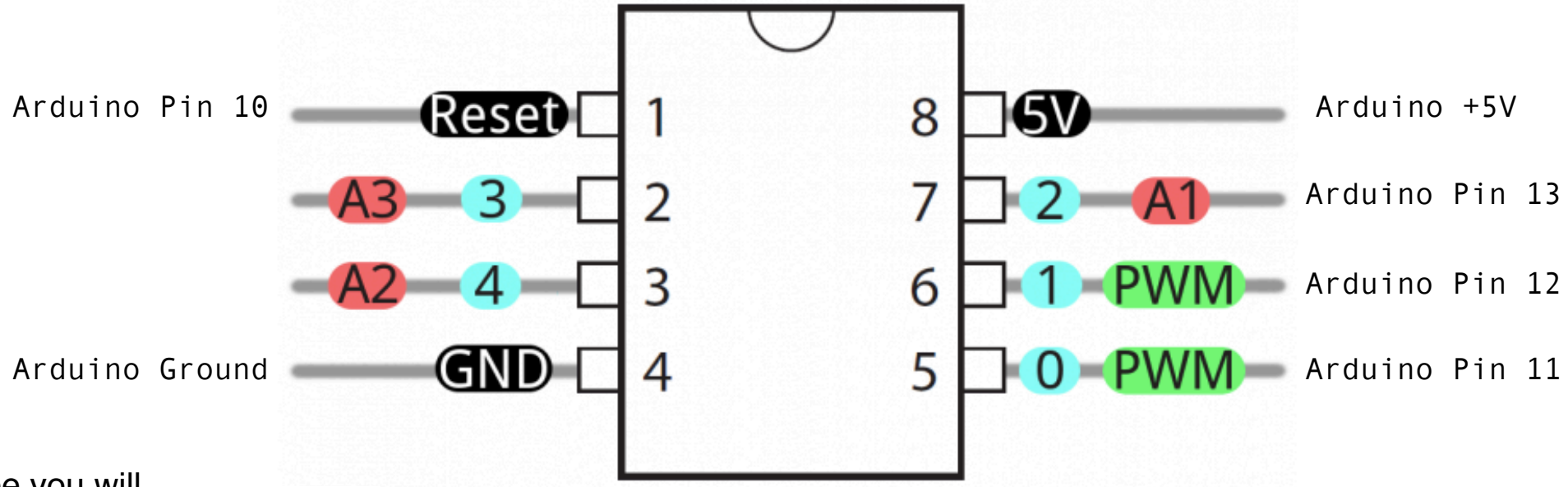
Electrolytic Capacitor
GND-->RESET
Short leg to GND!!!!!!!!!!

Arduino as Programmer





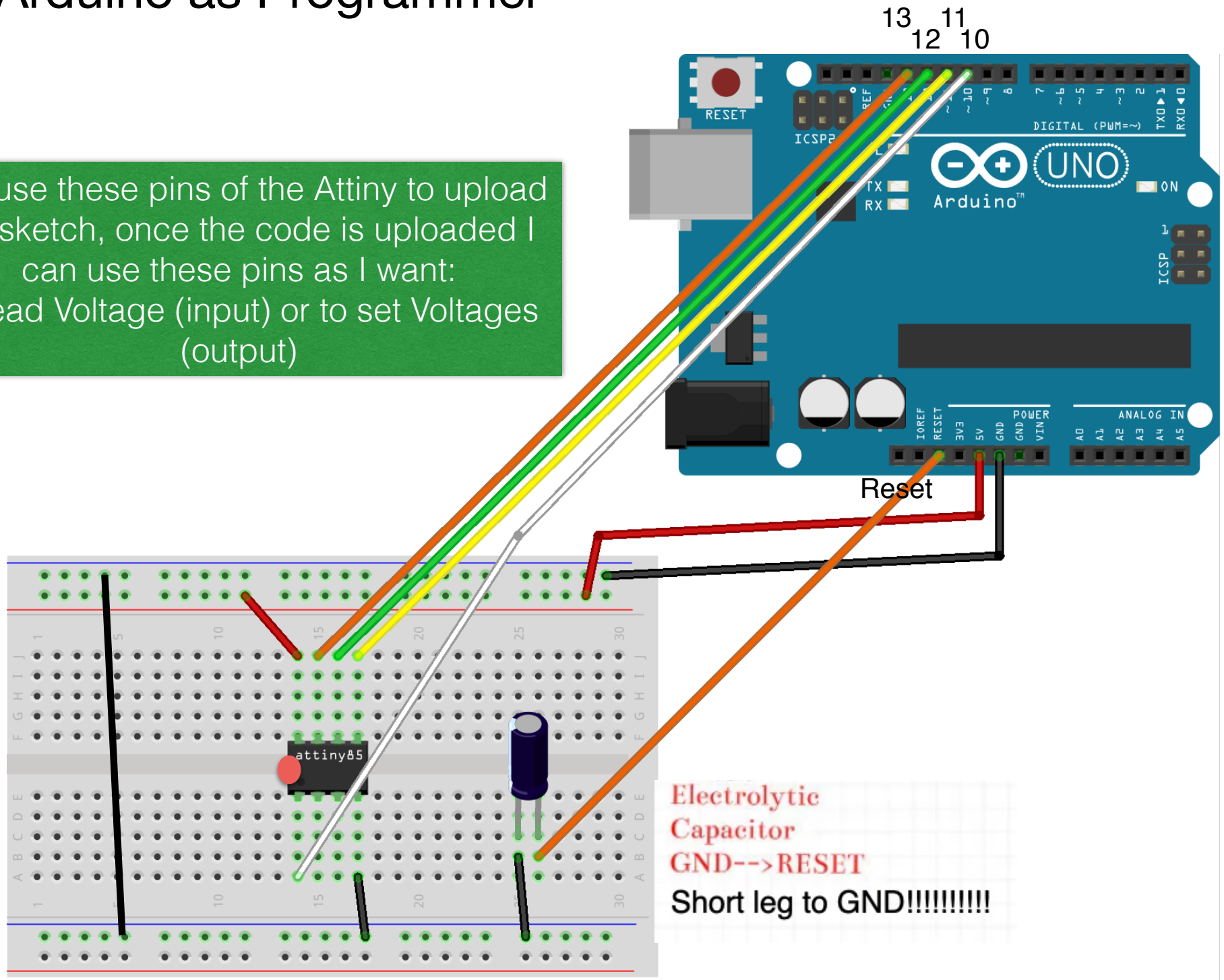
Arduino-->	ATtinv85
5V	Vcc
GND	GND
Pin 13	Pin 2
Pin 12	Pin 1
Pin 11	Pin 0
Pin 10	Reset



Maybe you will need the capacitor

Arduino as Programmer

You use these pins of the Attiny to upload the sketch, once the code is uploaded I can use these pins as I want:
to read Voltage (input) or to set Voltages (output)

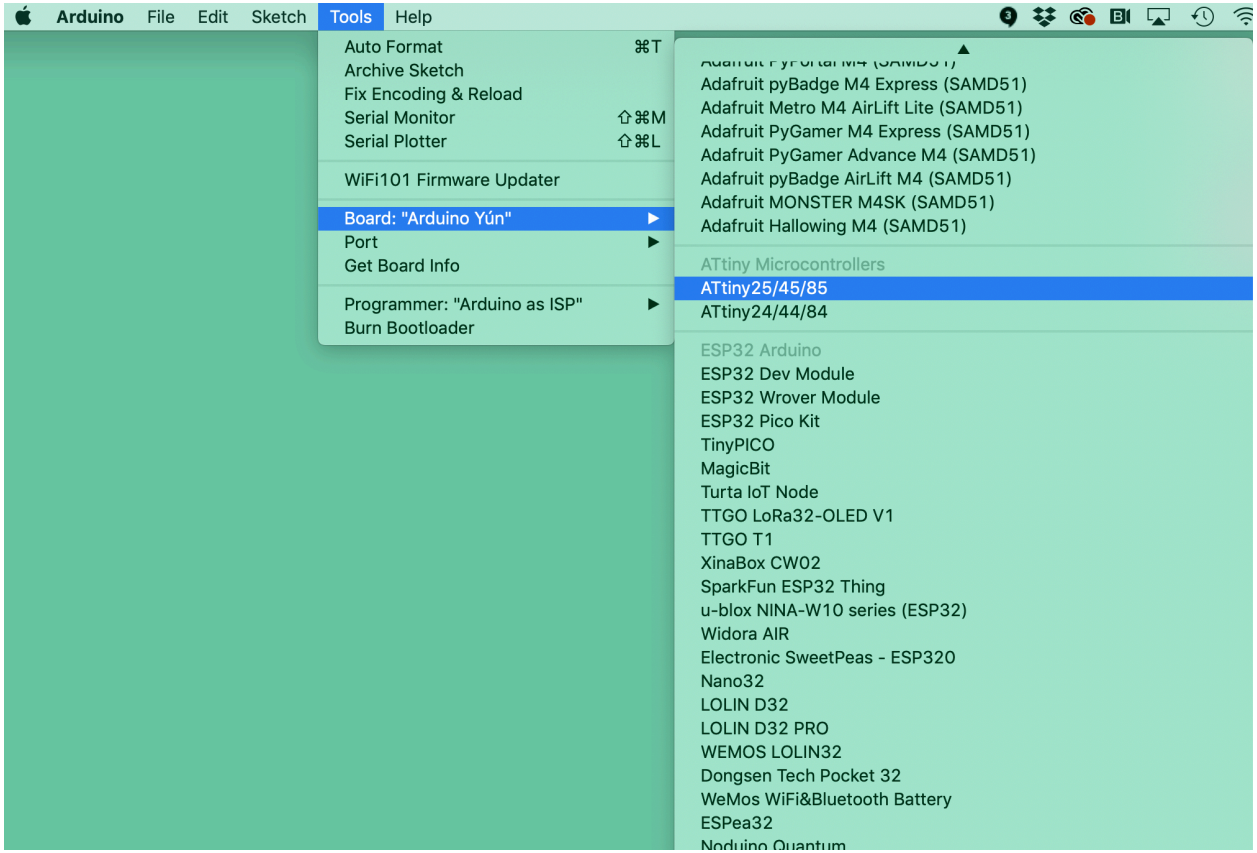


Arduino IDE - setup

Install the Attiny Board in Arduino IDE:
https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json

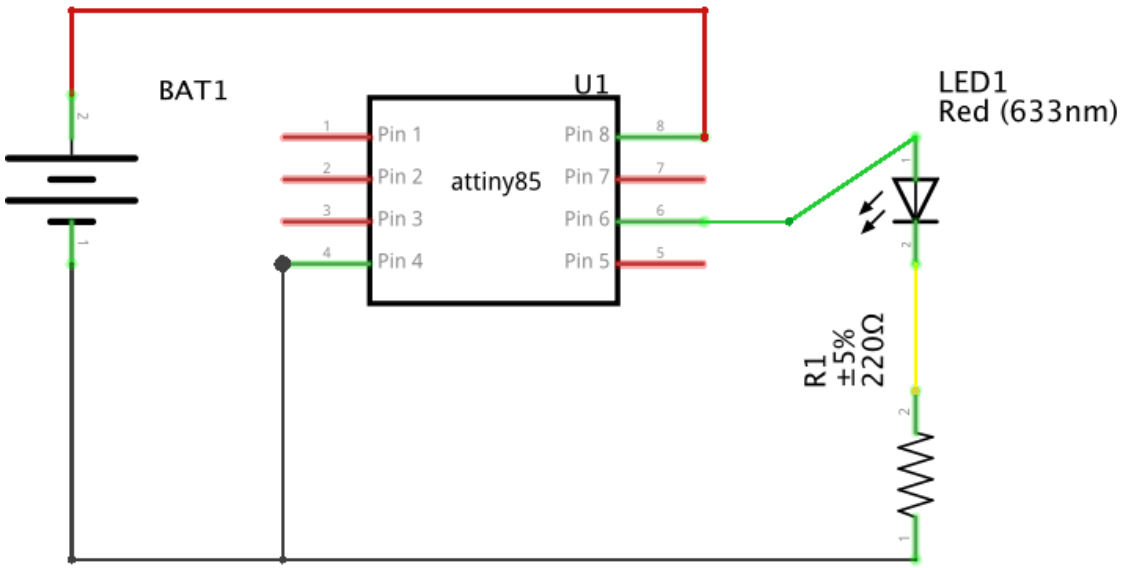
Resources:

- How install Attiny board in Arduino IDE:
  https://create.arduino.cc/projecthub/arjun/programming-attiny85-with-arduino-uno-afb829

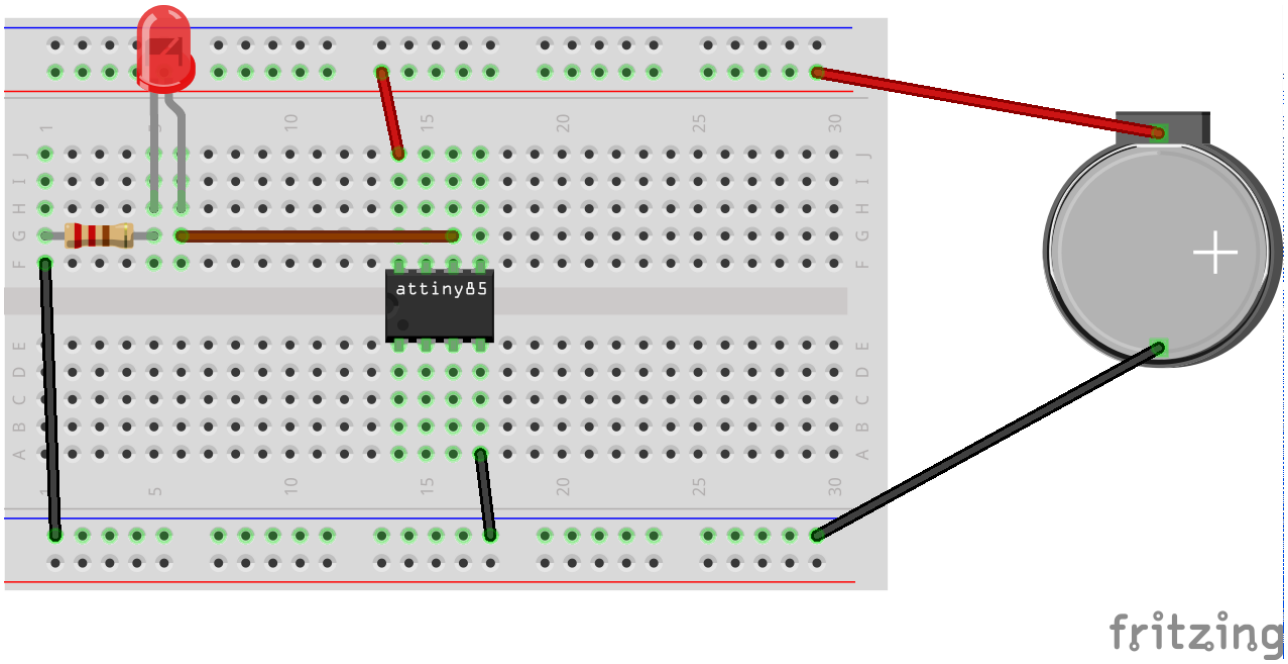


How to program the Attiny

Goal: Attiny 45/85 - Led



Don’t make this circuit yet.



Steps

1. Connect the Attiny to your Programmer:
   - AVR programmer
   - Arduino UNO (make sure that the Arduino is uploaded with the ARduinoISP sketch)

2. In Arduino IDE: Select board (attiny85 (internal 8MHz)) and port

3. In Arduino IDE: Select programmer, Arduino as ISP or FabISP

4. In Arduino IDE: If it is the first time YOU use the Attiny => Burn the bootloader

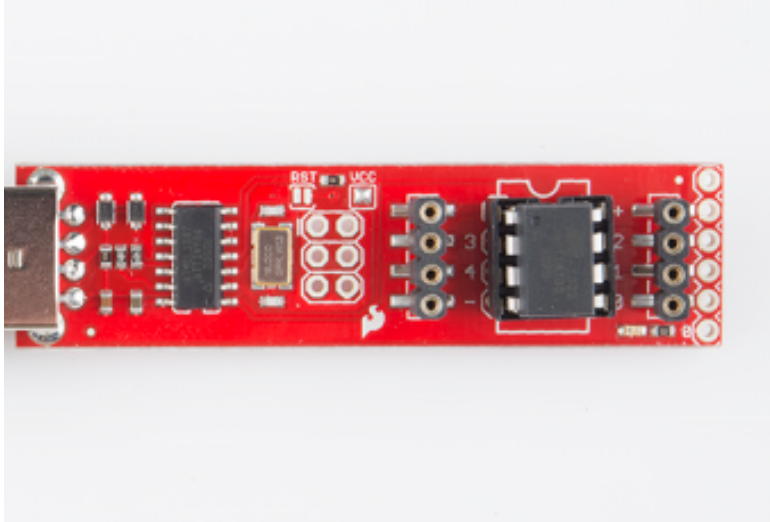
5. On Breadboard: connect the Attiny to your input/output devices

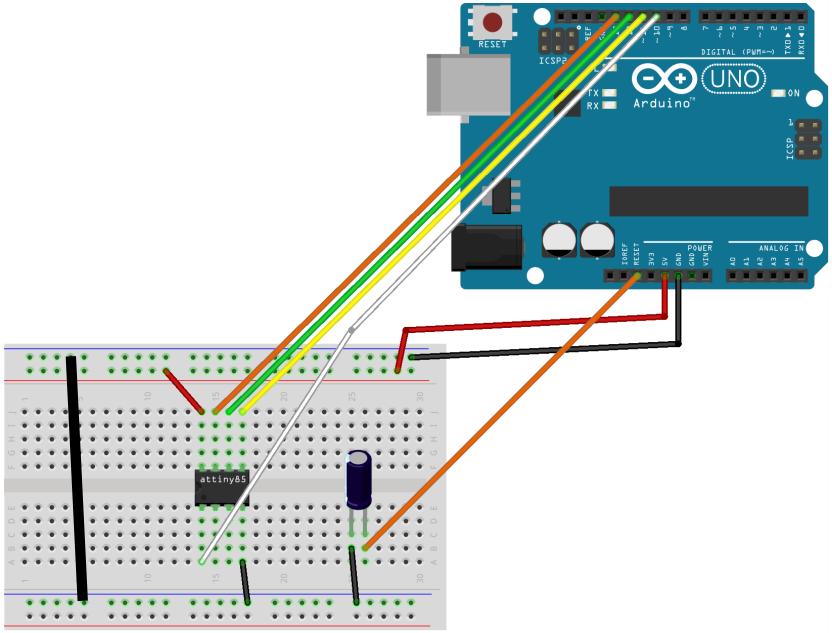
6. In Arduino IDE: Upload the code to the Attiny

How to program Attiny with Arduino Uno:
https://www.instructables.com/id/Program-an-ATtiny-with-Arduino/

Step 1

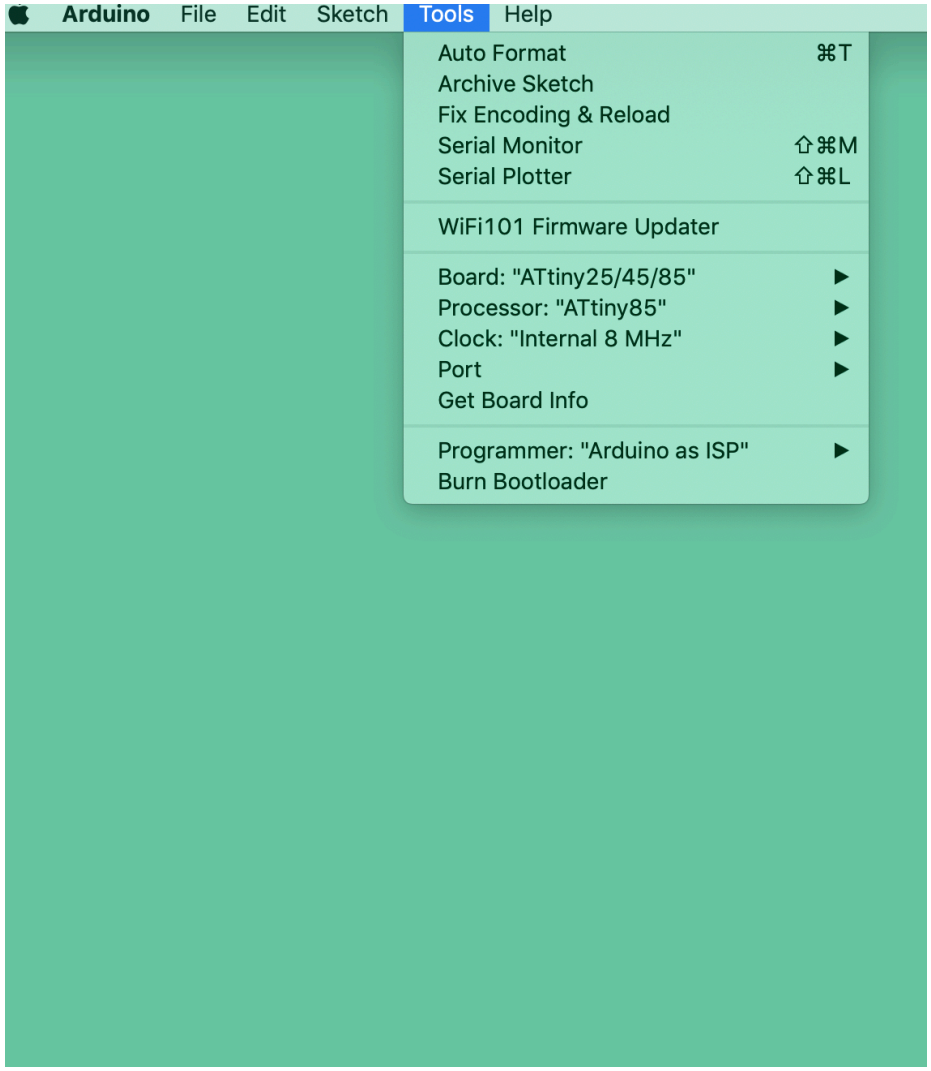
- Connect the Attiny to your Programmer:
  - AVR programmer
  - Arduino UNO (make sure that the Arduino is uploaded with the ARduinoISP sketch)





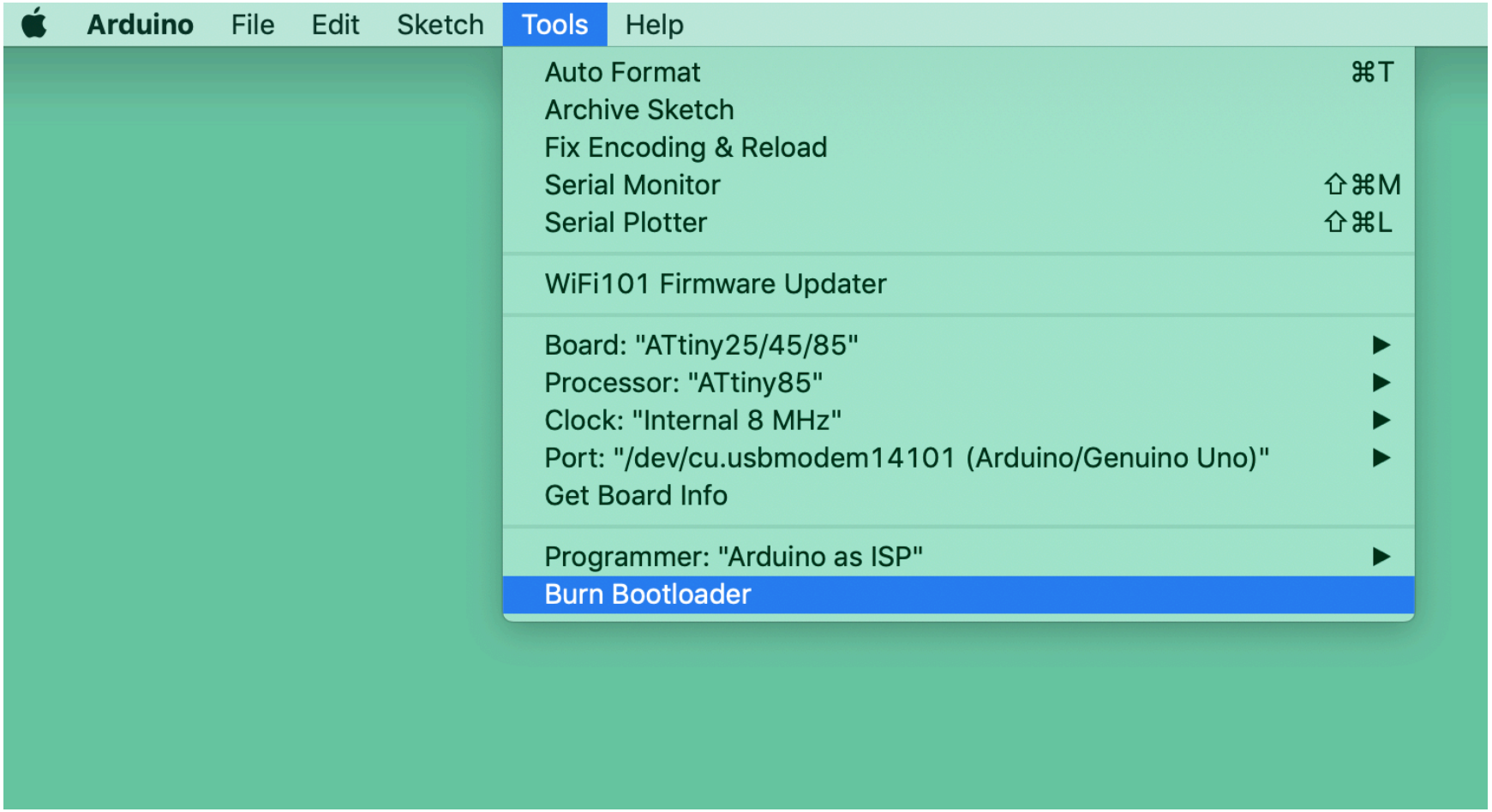
Step 2/3

- In Arduino IDE: Select board (attiny85 (internal 8MHz)) and port
- In Arduino IDE: Select programmer, Arduino as ISP or FabISP



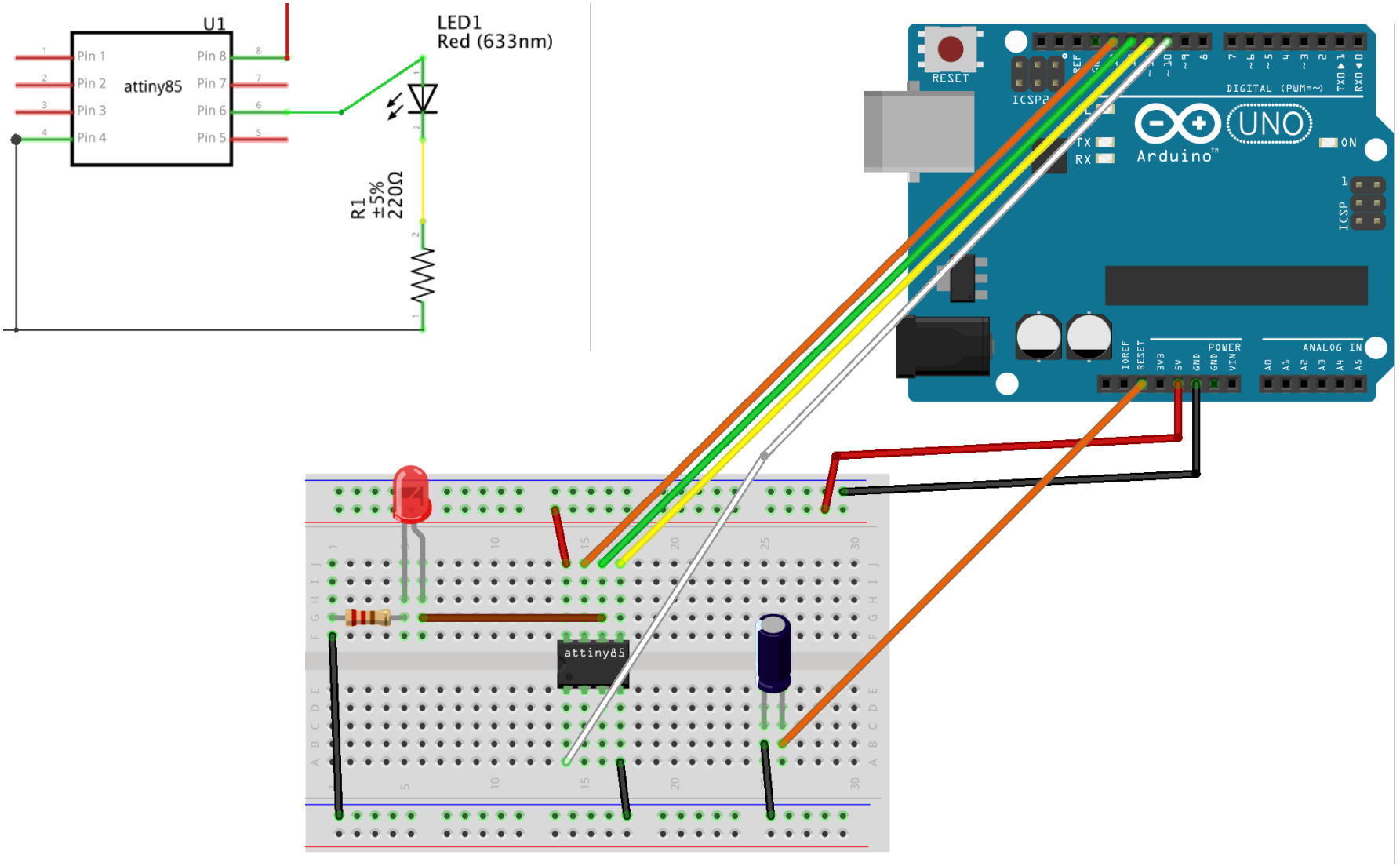
Step 4

- In Arduino IDE: **If it is the first time YOU use the Attiny** => Burn the bootloader



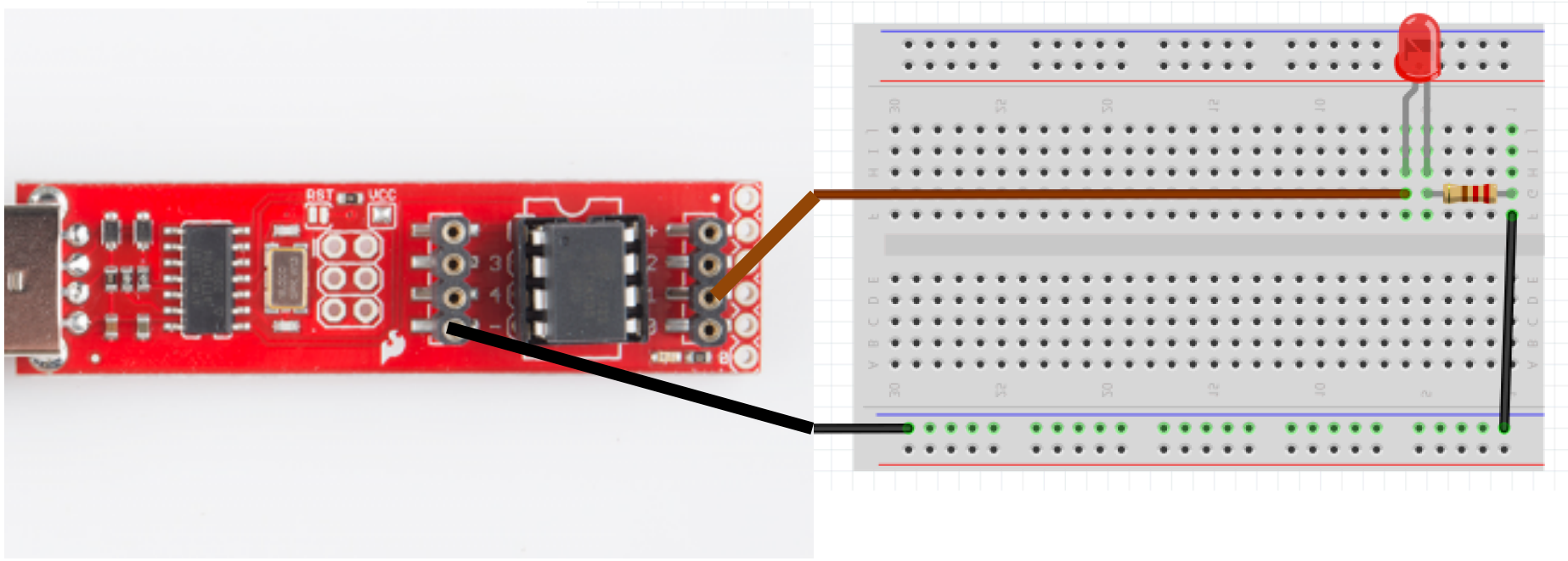
Step 5

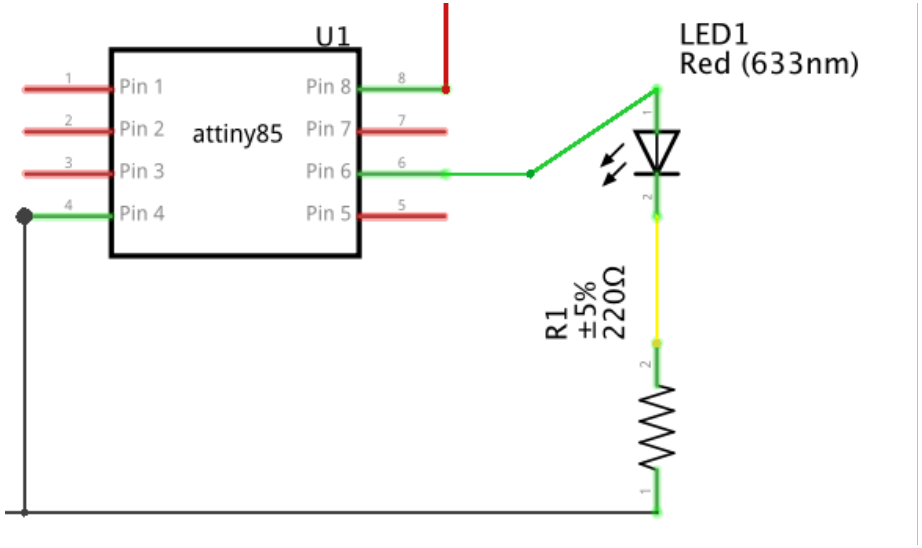
- On Breadboard: connect the Attiny to your input/output devices



Step 5

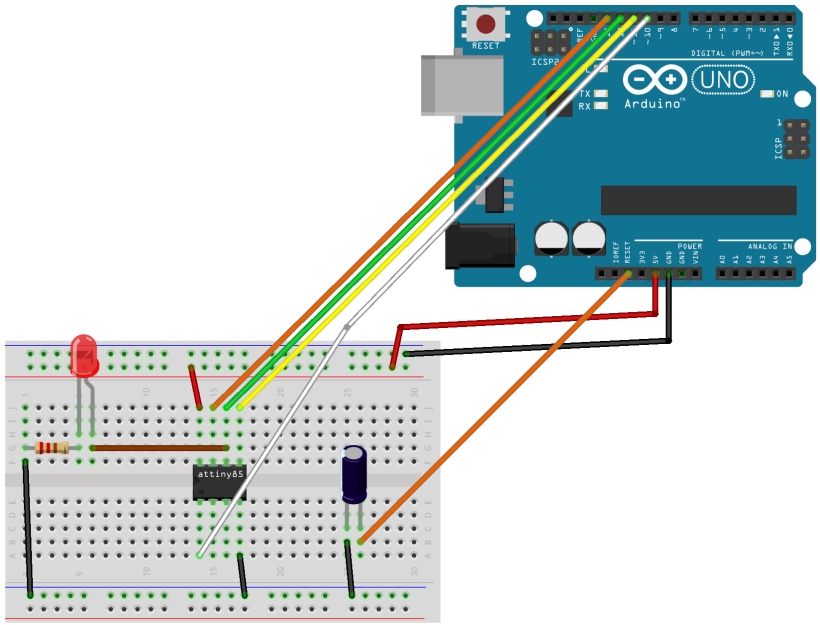
- On Breadboard: connect the Attiny to your input/output devices

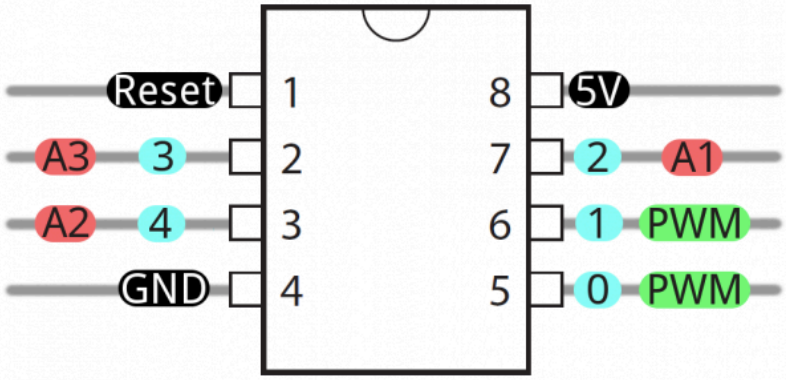




Step 6

- In Arduino IDE: Upload the code to the Attiny





attiny_led | Arduino 1.8.5

Open

attiny_led

```
/* Emma Pareschi, Nov 2019
 *
 *  the led connected to pin 1. it blinks
*/

int led_pin = 1;

void setup() {
  // put your setup code here, to run once:
  pinMode(led_pin, OUTPUT);   // set the pin 3 as OUTPUT

}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(led_pin, HIGH);      //turn the Led ON
  delay(1000);                  //wait
  digitalWrite(led_pin, LOW);       //turn the Led OFF
  delay(1000);                  //wait
}
```

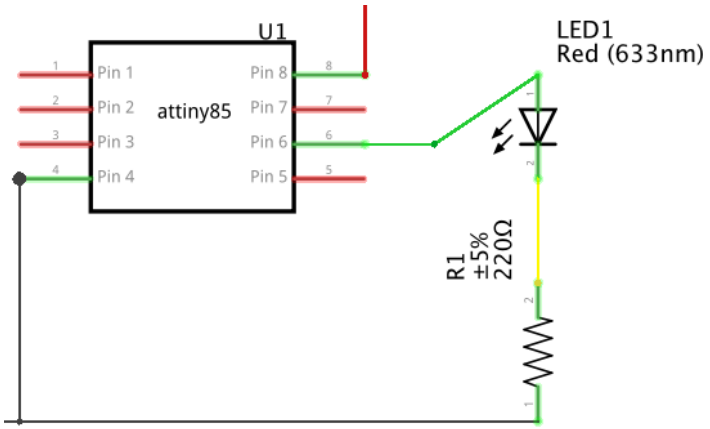
```
Done Saving.

avrdude: verifying ...
avrdude: 692 bytes of flash verified

avrdude done.  Thank you.
```

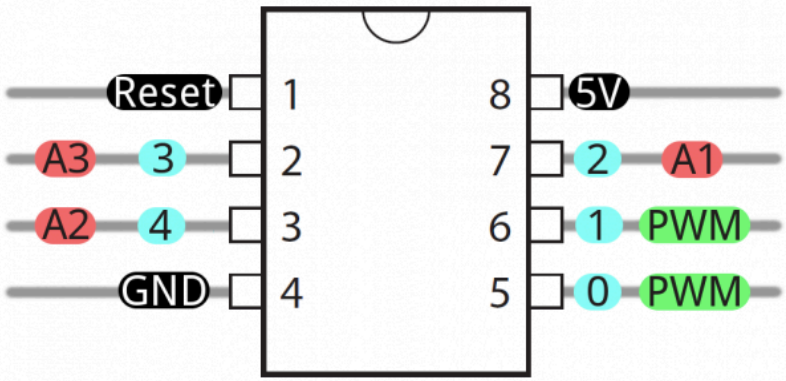
6 ATtiny25/45/85, ATtiny85

attiny_led

Step 6

- In Arduino IDE: Upload the code to the Attiny





attiny_led | Arduino 1.8.5

Open

attiny_led

```
/* Emma Pareschi, Nov 2019
 *
 *  the led connected to pin 1. it blinks
*/

int led_pin = 1;

void setup() {
  // put your setup code here, to run once:
  pinMode(led_pin, OUTPUT);   // set the pin 3 as OUTPUT

}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(led_pin, HIGH);      //turn the Led ON
  delay(1000);                  //wait
  digitalWrite(led_pin, LOW);       //turn the Led OFF
  delay(1000);                  //wait
}
```

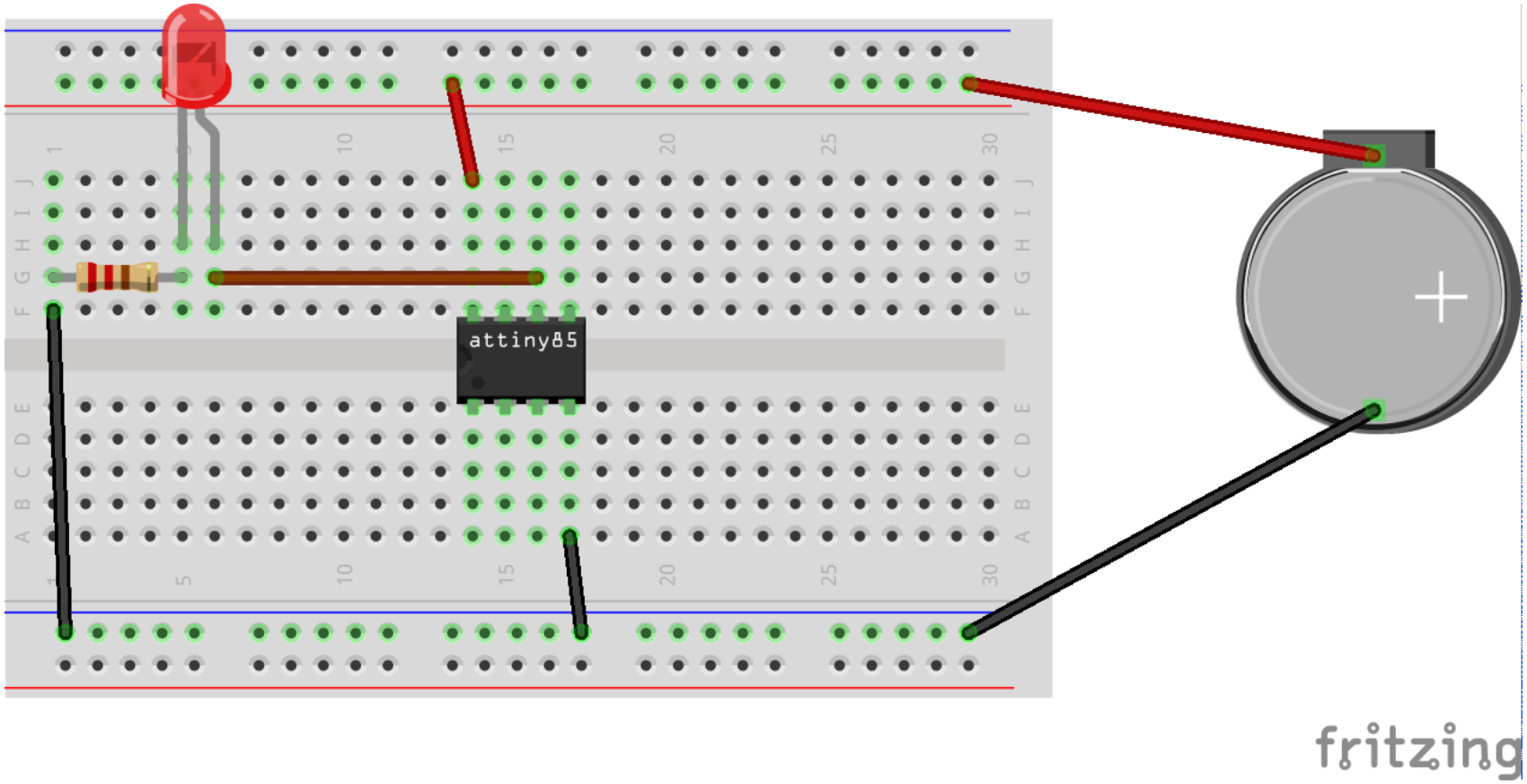
Done Saving.

```
avrdude: verifying ...
avrdude: 692 bytes of flash verified

avrdude done.  Thank you.
```

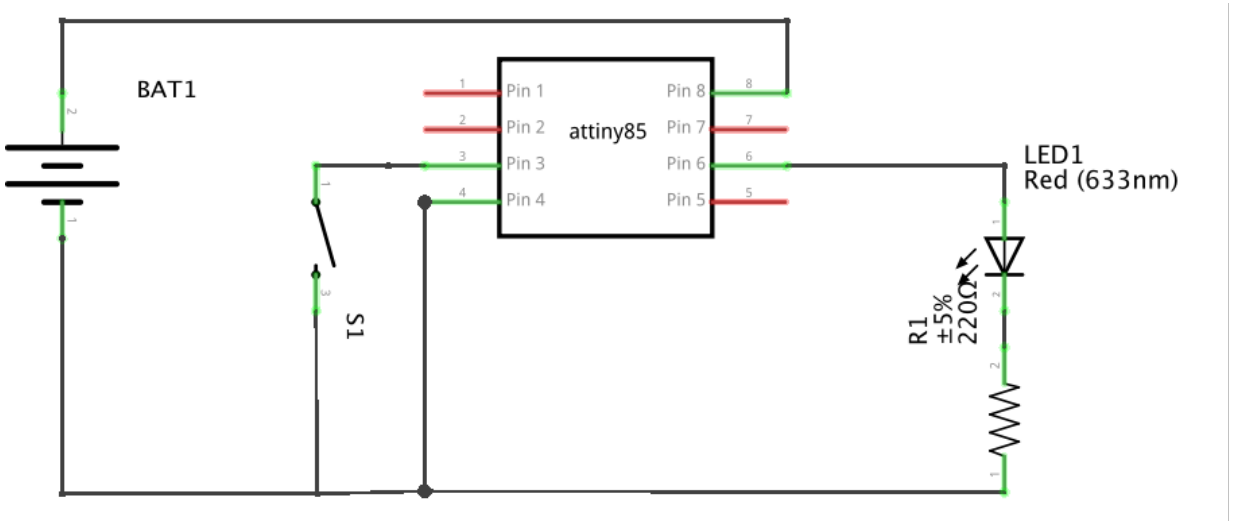
6 ATtiny25/45/85, ATtiny85

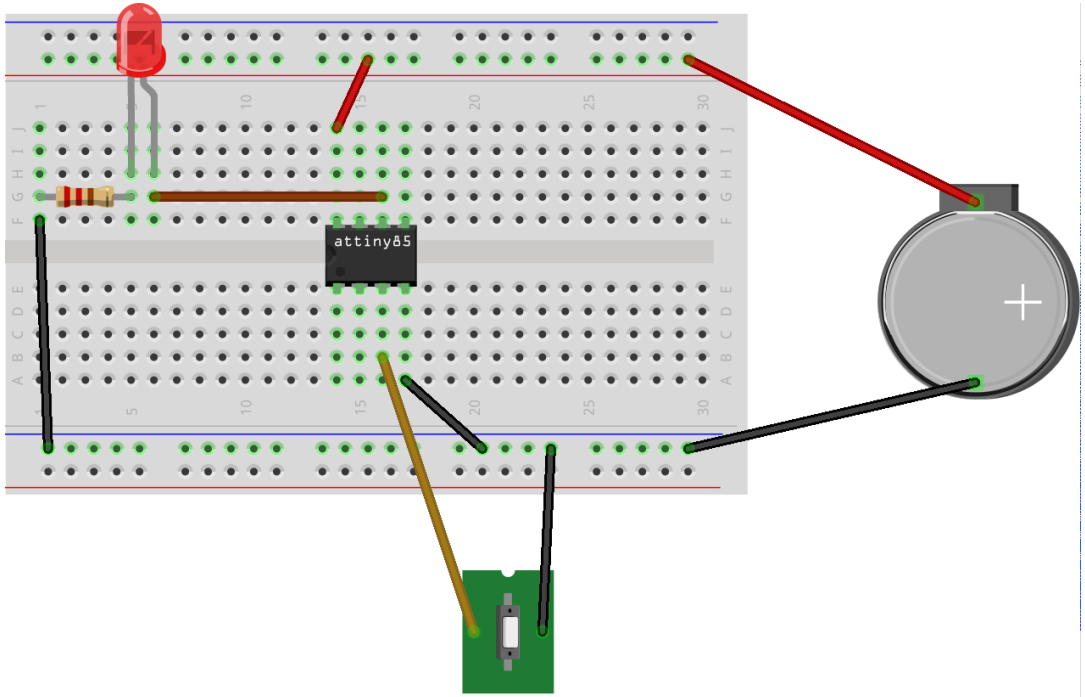
attiny_led

Attiny45 Led - Stand Alone

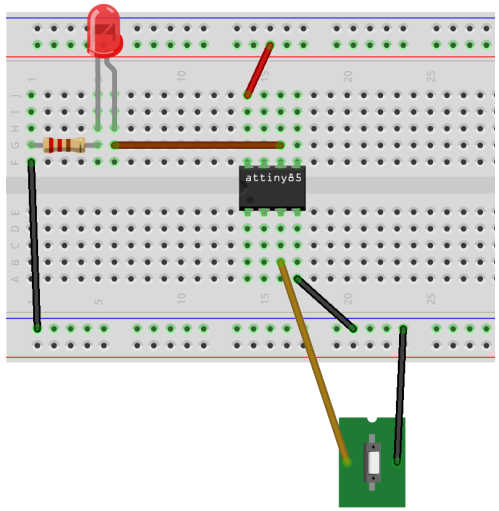


Goal: Attiny45 Led - Sw

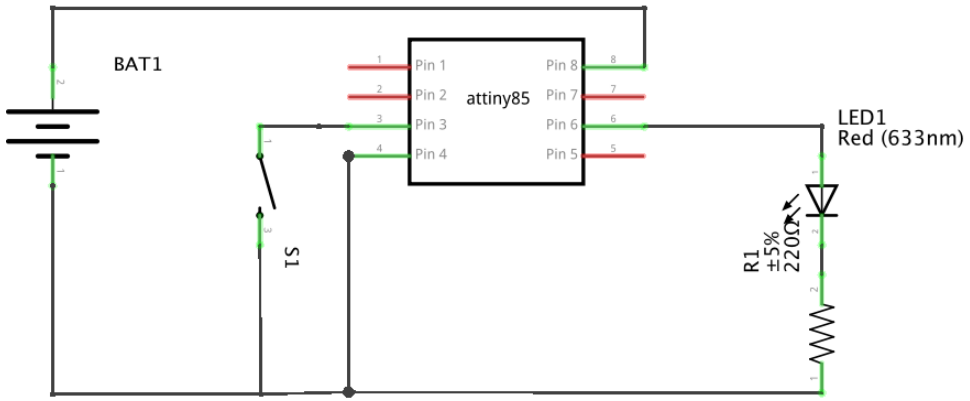




Attiny45 Led - Sw - Code



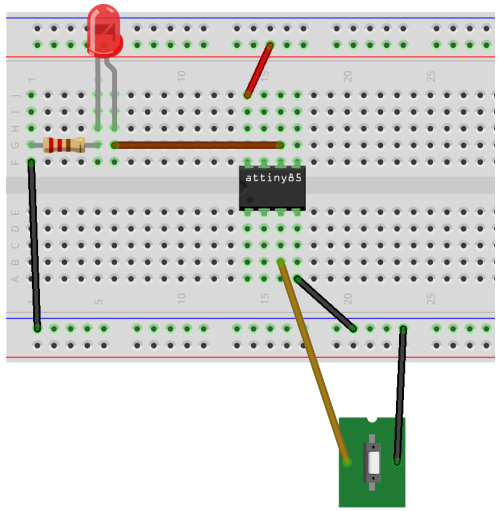
In this image the Programmer is missing!!!!!



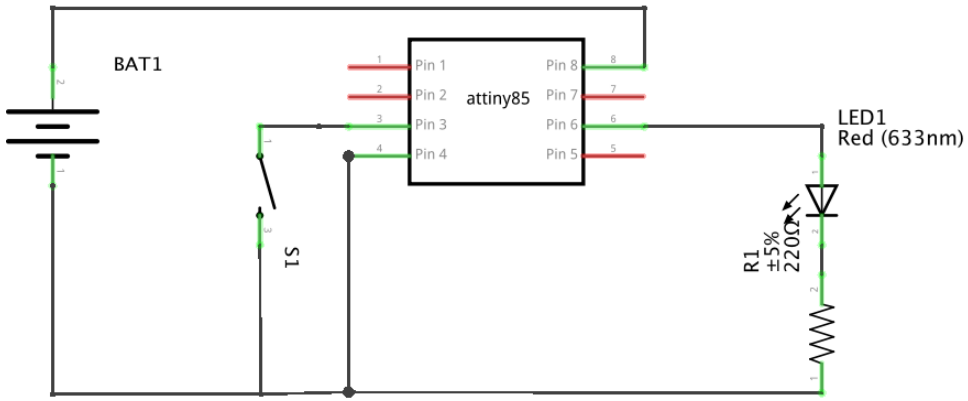
attiny_led_digitalsensor_1

```
/*
Emma Pareschi 2020
the Led turns on when the digital sensor is triggered
 */

// this constant won't change:
int  sw_pin = 4;    // the pin that the pushbutton is attached to

int led_pin = 1; //pin of the led

// Variables will change:
int sw_status = 0;          // current state of the button

void setup() {
  // initialize input and output pins:
  pinMode(sw_pin, INPUT_PULLUP);
  pinMode(led_pin, OUTPUT);

}

void loop() {

  // read the pushbutton input pin:
  sw_status = digitalRead(sw_pin);

  // compare the switch status to its previous state
  if (sw_status == LOW) {
    digitalWrite(led_pin, HIGH);
  } else {
    digitalWrite(led_pin, LOW);
  }

}
```

Attiny45 Led - Sw - Code



In this image the Programmer is missing!!!!!



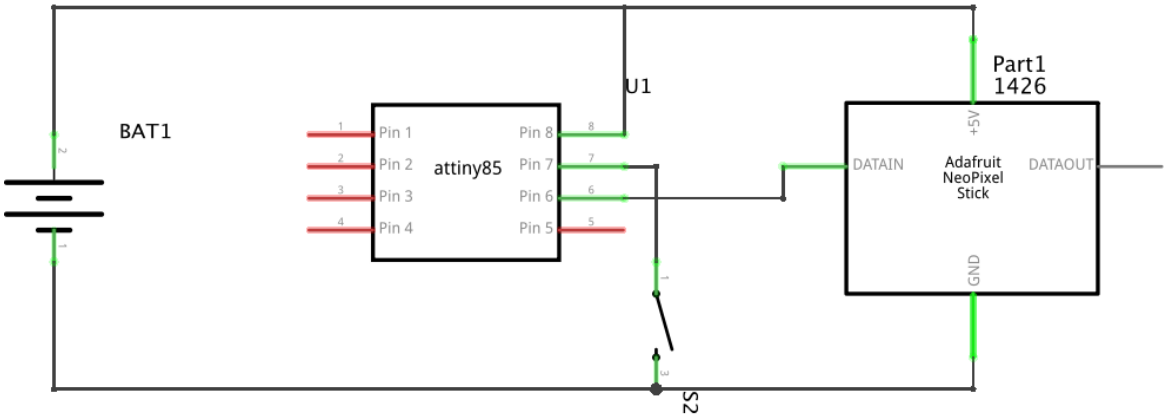
attiny_led_on_off

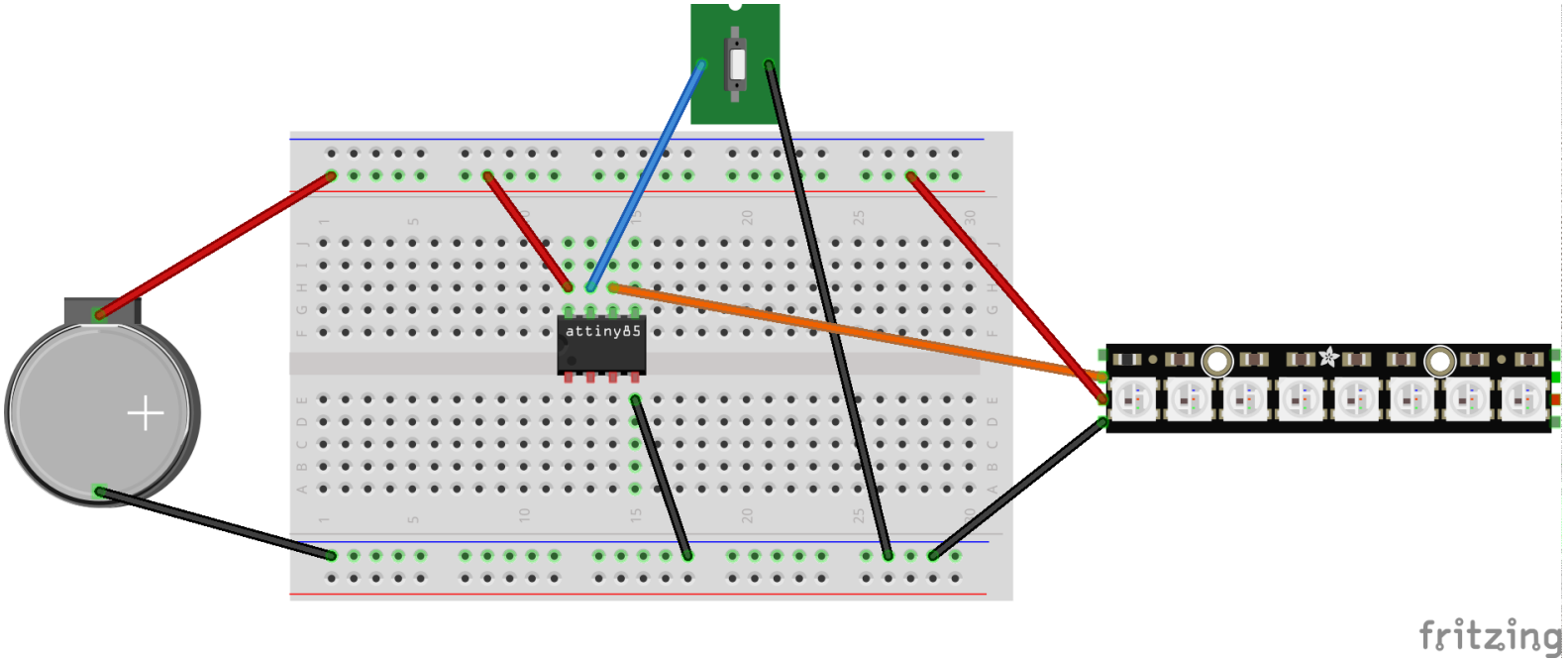
```
/*
Emma Pareschi 2020
I count the times the push button is pressed and I turn on/off a led
 */

// this constant won't change:
int  sw_pin = 4;    // the pin that the pushbutton is attached to
int counter_reset = 2; //variable to reset the counter

int led_pin = 1; //pin of the led

// Variables will change:
int counter = 0;   // counter for the number of button presses
int sw_status = 0;         // current state of the button
int last_sw_status = 1;     // previous state of the button

void setup() {
  // initialize input and output pins:
  pinMode(sw_pin, INPUT_PULLUP);
  pinMode(led_pin, OUTPUT);

}

void loop() {
```

```
Done Saving.

avrdude: verifying ...
avrdude: 944 bytes of flash verified

18                                                    ATtiny25/45/85, ATtiny85
```

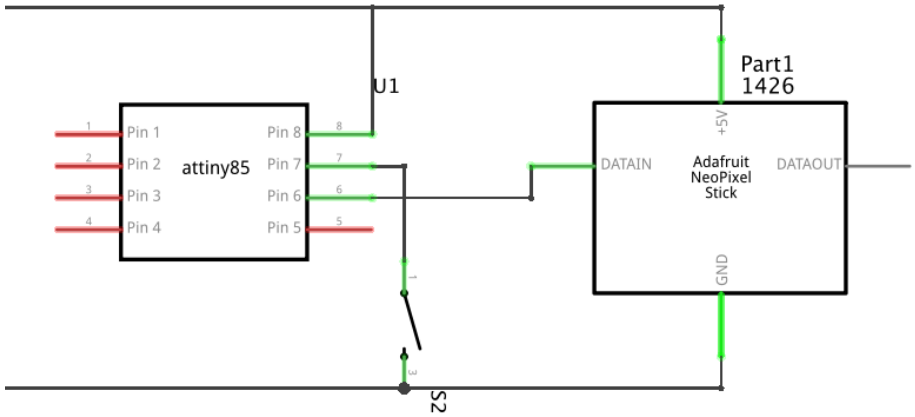
attiny_led_digitalsensor_2

Goal: Attiny45 Neopixel - Sw





Attiny45 Neopixel - Sw - Code



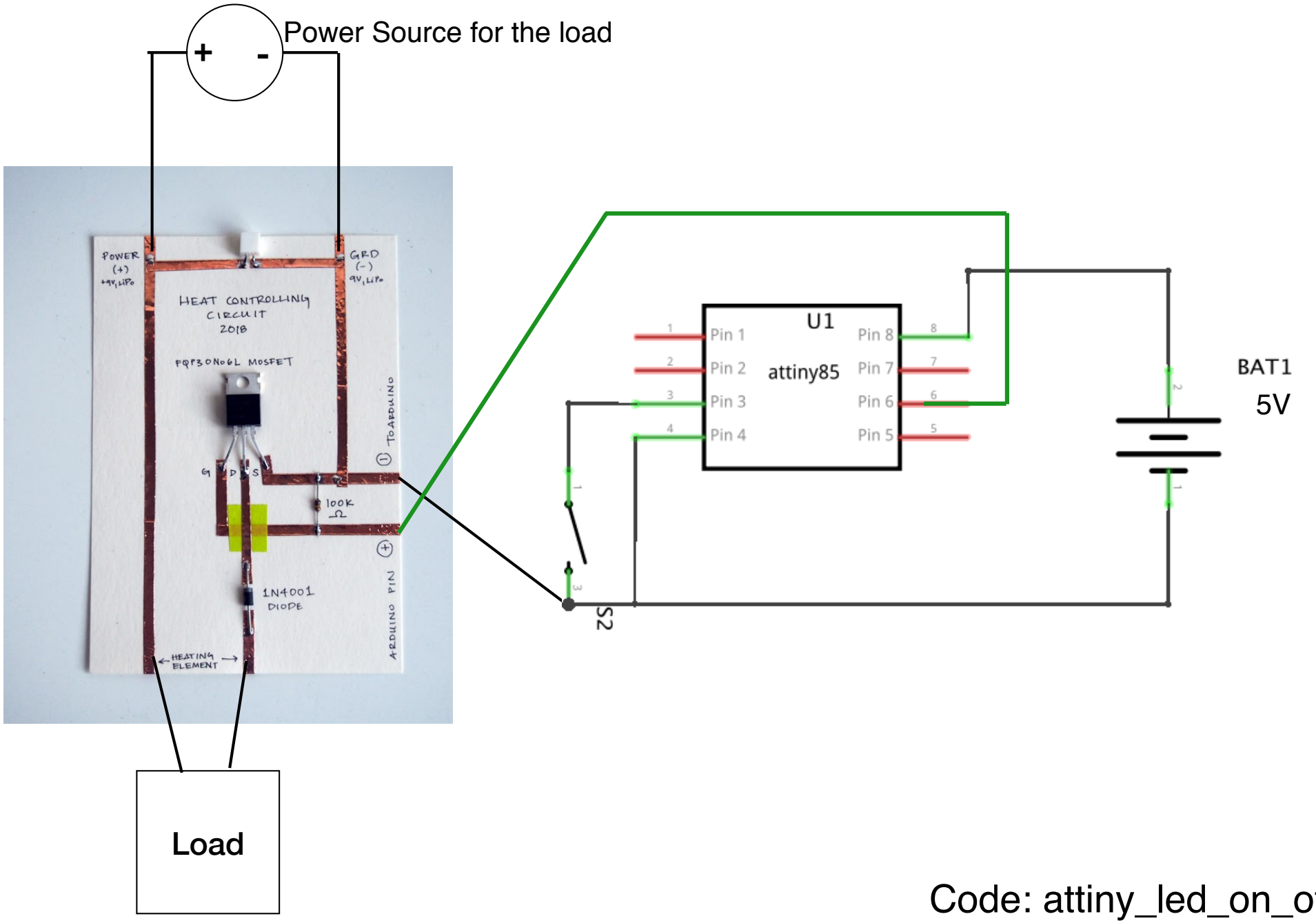
In this image the Programmer is missing!!!!!

```
attiny_neopixel_Wipe_sw
#include <Adafruit_NeoPixel.h>
// A basic everyday NeoPixel strip test program.

// NEOPIXEL BEST PRACTICES for most reliable operation:
// - Add 1000 uF CAPACITOR between NeoPixel strip's + and - connections.
// - MINIMIZE WIRING LENGTH between microcontroller board and first pixel.
// - NeoPixel strip's DATA-IN should pass through a 300-500 OHM RESISTOR.
// - AVOID connecting NeoPixels on a LIVE CIRCUIT. If you must, ALWAYS
//   connect GROUND (-) first, then +, then data.
// - When using a 3.3V microcontroller with a 5V-powered NeoPixel strip,
//   a LOGIC-LEVEL CONVERTER on the data line is STRONGLY RECOMMENDED.
// (Skipping these may work OK on your workbench but can fail in the field)

// Which pin on the Arduino is connected to the NeoPixels?
// On a Trinket or Gemma we suggest changing this to 1:
#define LED_PIN    1

// How many NeoPixels are attached to the Arduino?
#define LED_COUNT 10

// Declare our NeoPixel strip object:
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRB + NEO_KHZ800);
// Argument 1 = Number of pixels in NeoPixel strip
// Argument 2 = Arduino pin number (most are valid)
```

```
Done Saving.

avrdude: verifying ...
avrdude: 2670 bytes of flash verified

19                                   ATtiny25/45/85, ATtiny85, Internal 8 MHz on /dev/
```

attiny_neopixel_Wipe_sw

Attiny45 Power load - sw



Code: attiny_led_on_off

Codes Wearable Weeks

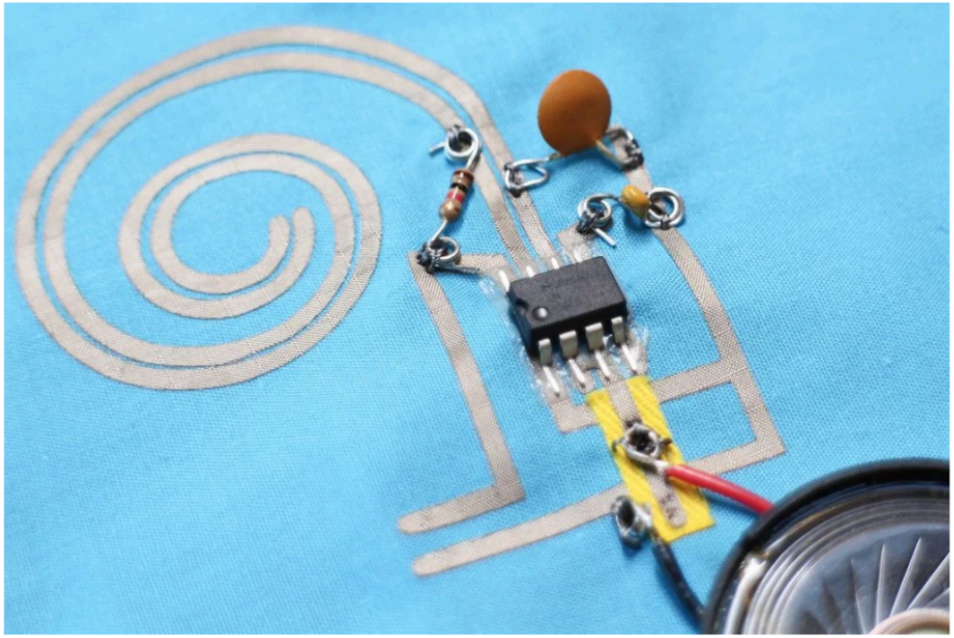
You can use all the codes (and circuits) I shared during wearable weeks but:

- Change the pin numbers!

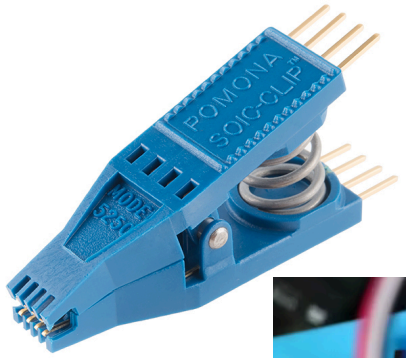
- Remove all Serial commands

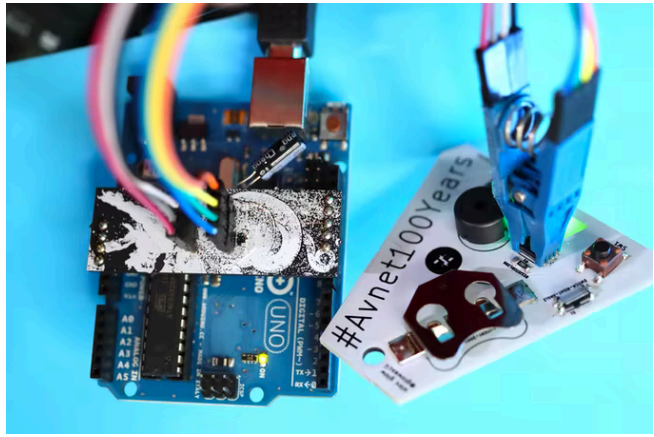
How to integrate the Attiny in your project

ATtiny POV by Mika Satomi

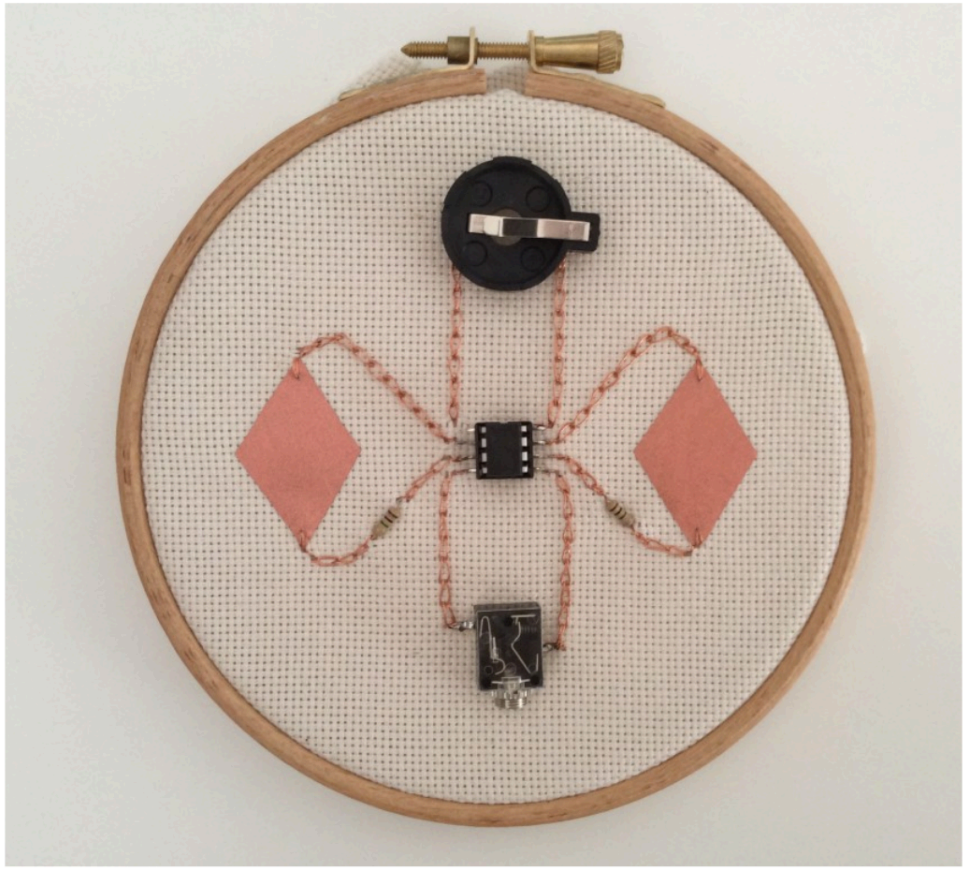


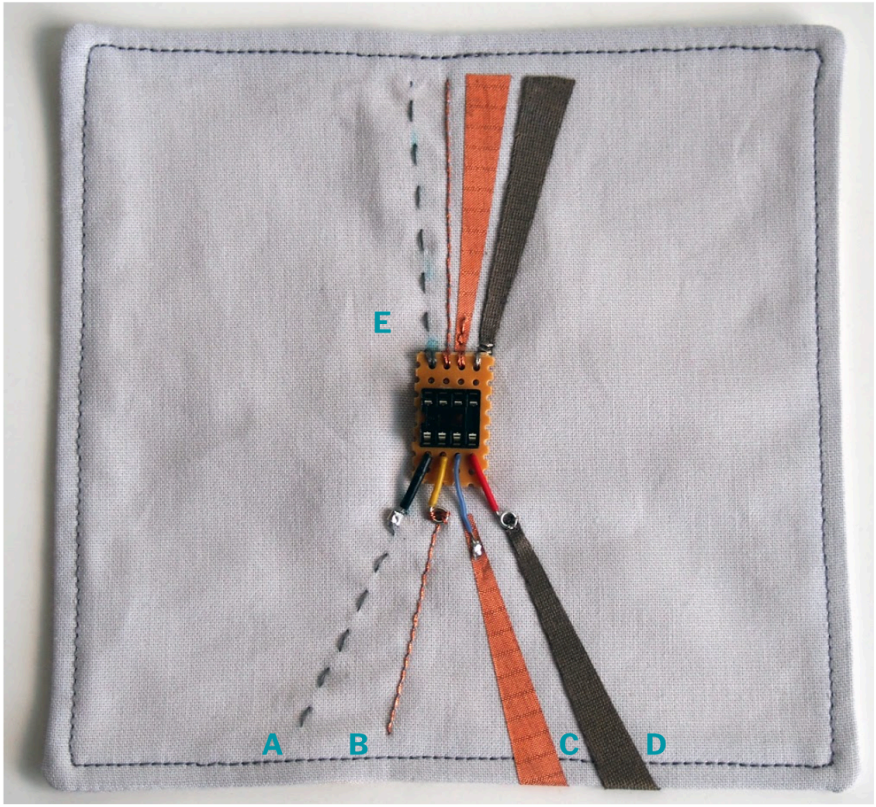
How to Work with Conductive Fabric by Lara Grant



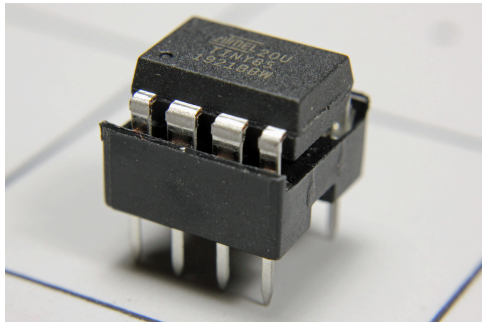


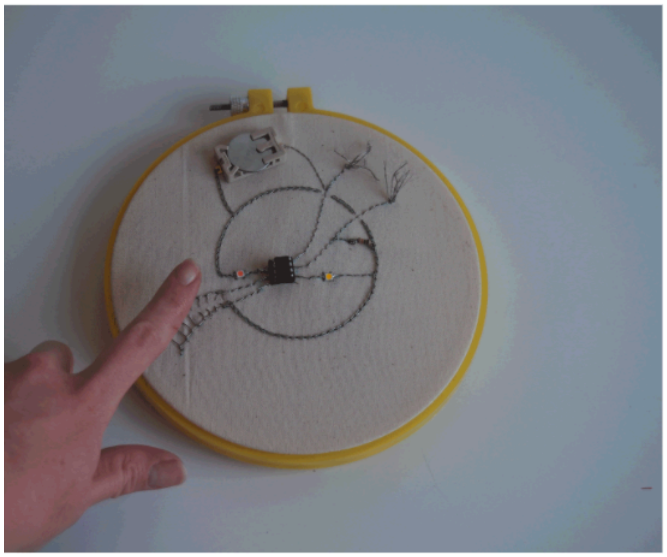
Stulpe circuit by Hannah Perner-Wilson



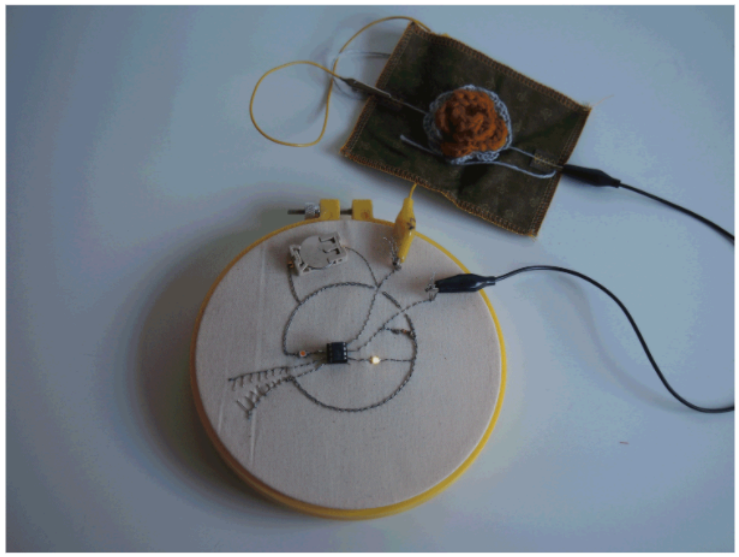


chainStitch Noise by Afroditi Psarra

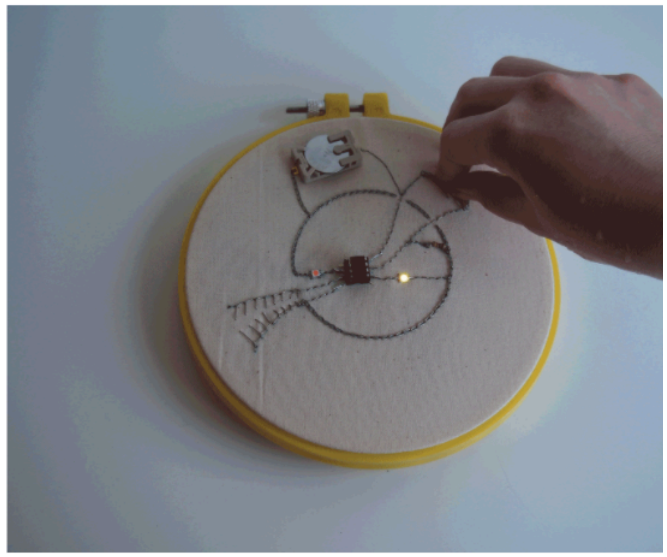




Touch



Variable resistor

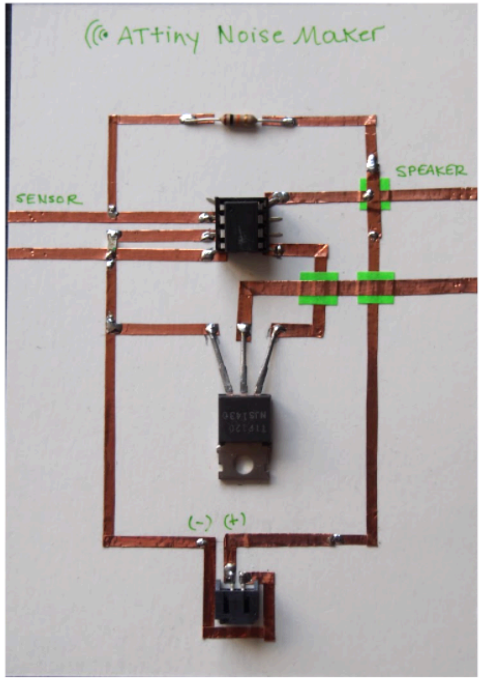


Switch

TOUCH + SENSOR

ATTINY CIRCUIT

+



+

SOUND

ATTINY CIRCUIT

Stretch some 8 bit noise using the ATtiny!