

*„der elektrische Schaltkreis,
eine Erweiterung des Zentralen Nervensystems“*

Marshall McLuhan





tinker (verb)

he spent hours tinkering with the car | these proposals will do no more than tinker with the existing laws: try to mend/improve, work amateurishly on, fiddle with, play (about/around) with, toy with, trifle with, dally with, dabble with, potter about with, fool about/around with; tamper with, interfere with, meddle with; tinker at/with the edges of, adjust slightly; informal mess about/around with, rearrange the deckchairs on the Titanic; Brit. informal muck about/around with.

Physical Computing

... bedeutet im weitesten Sinne, interaktive, physische Systeme durch die Verwendung von Hardware und Software zu erstellen. Diese Systeme reagieren auf Ereignisse in der realen, analogen Welt und/oder wirken auf sie ein.

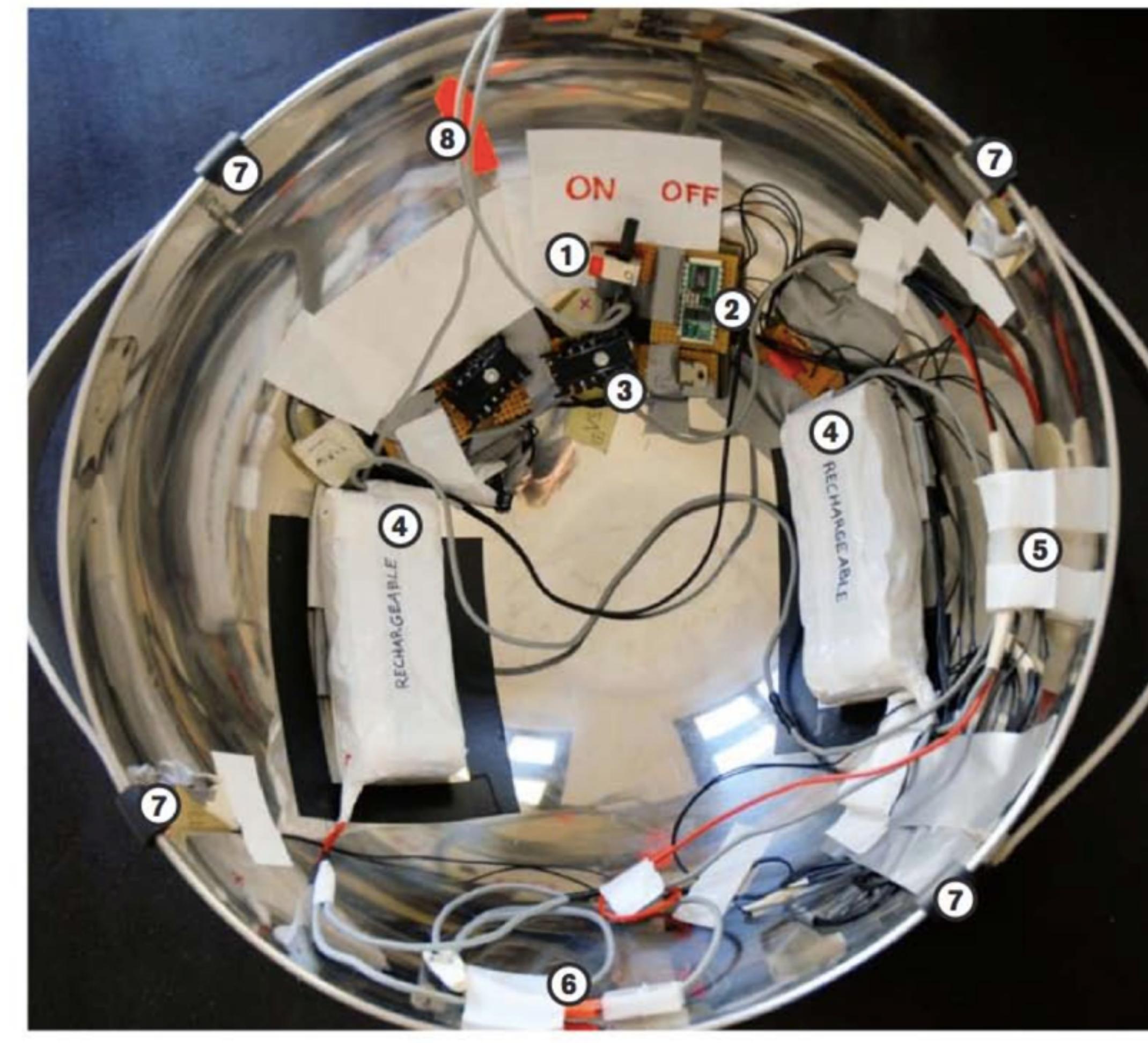
TED: Massimo Banzi

how arduino is open sourcing imagination



Herzfassen



**OVERVIEW / WHAT, WHERE, WHAT FOR**

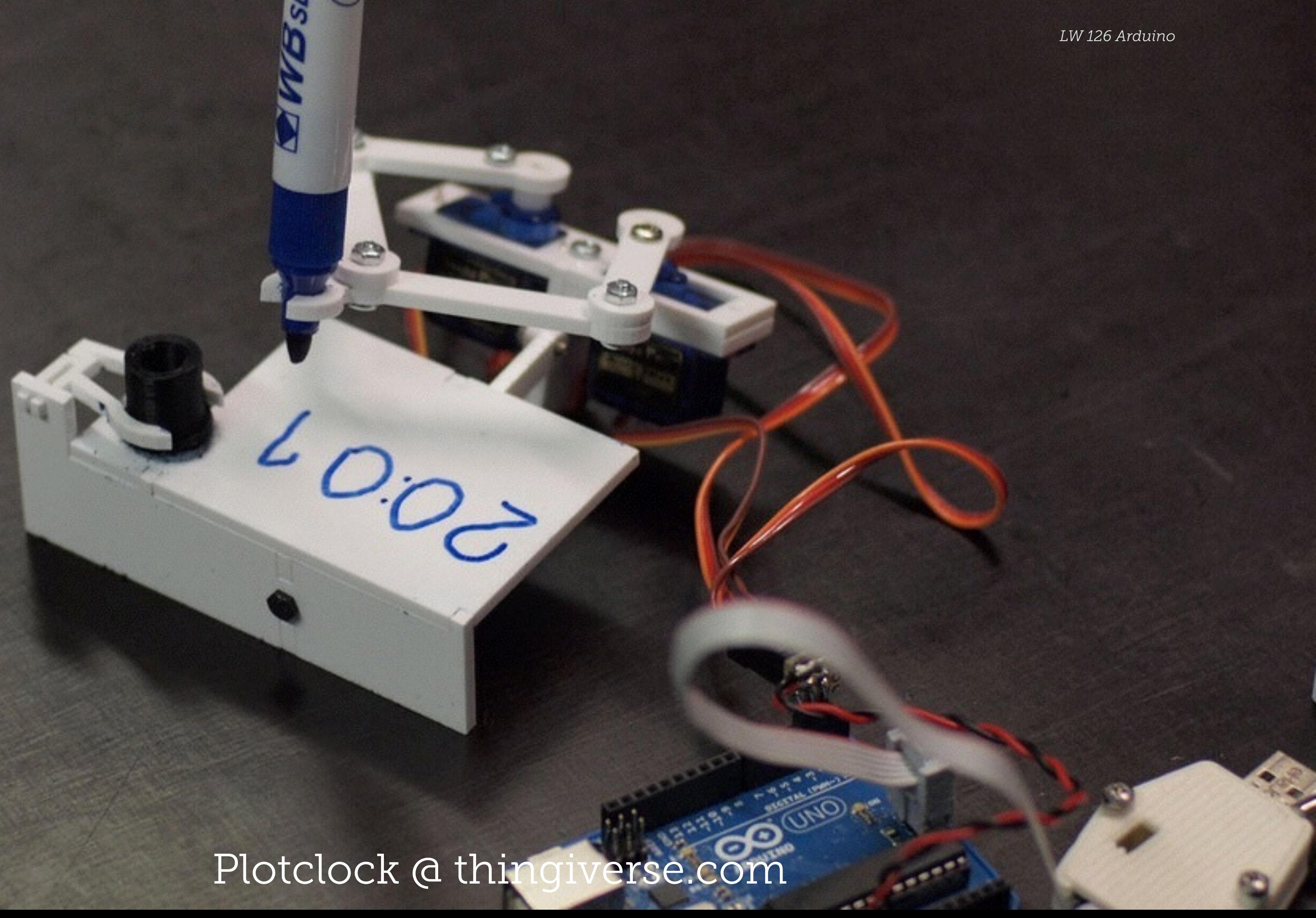
- 1) on/off switch
- 2) microcontroller (basic stamp)
- 3) red control LED
- 4) battery pack
- 5) connecting plug battery to electronics
- 6) charging cable and plug for battery charger
- 7) rubber tube to damp vibration
- 8) neon mark to know which way to place the inner bowl down again. (there is an according neon mark on the wood mounted on the inner bowl)



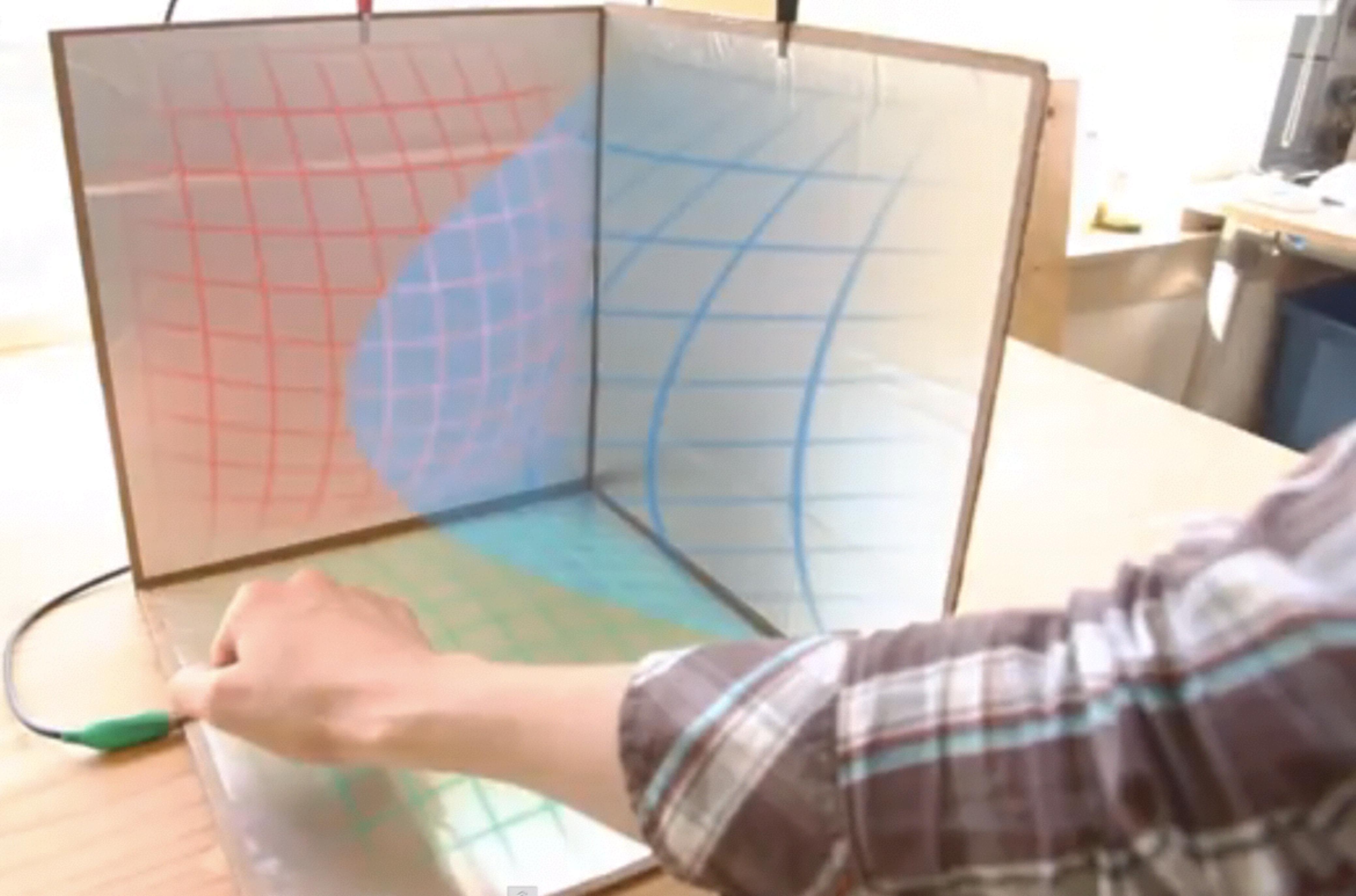




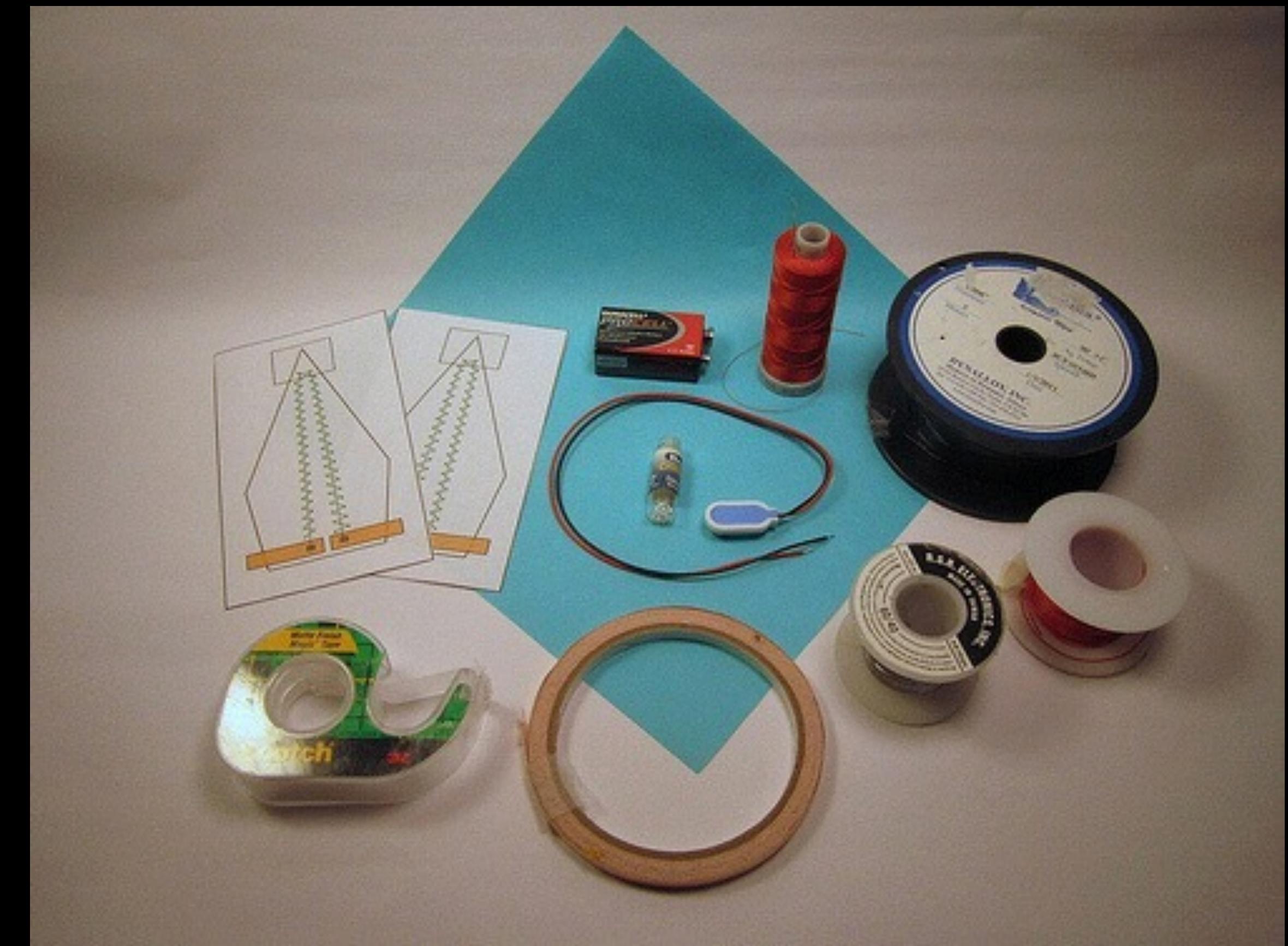
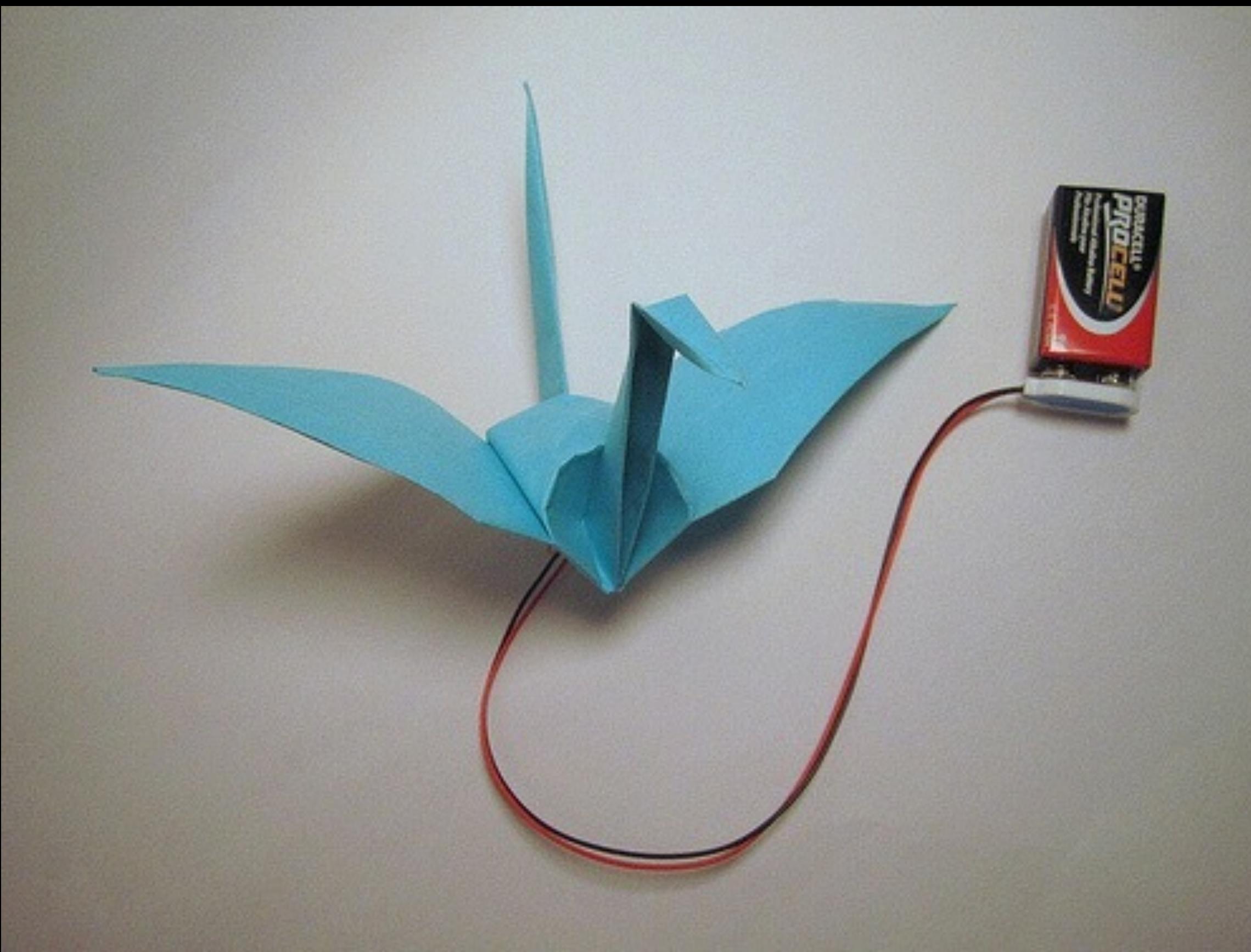




Plotclock @ thingiverse.com



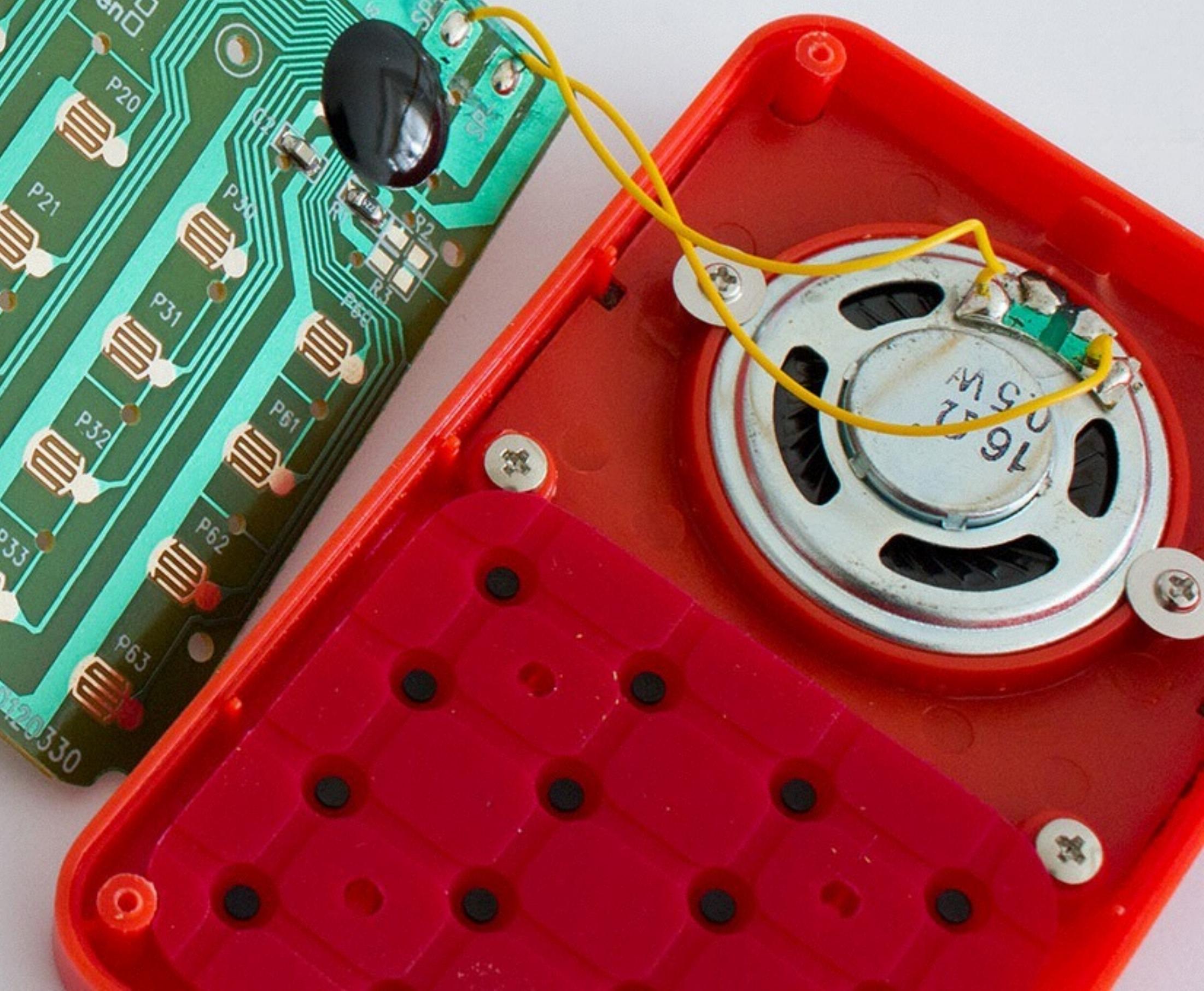
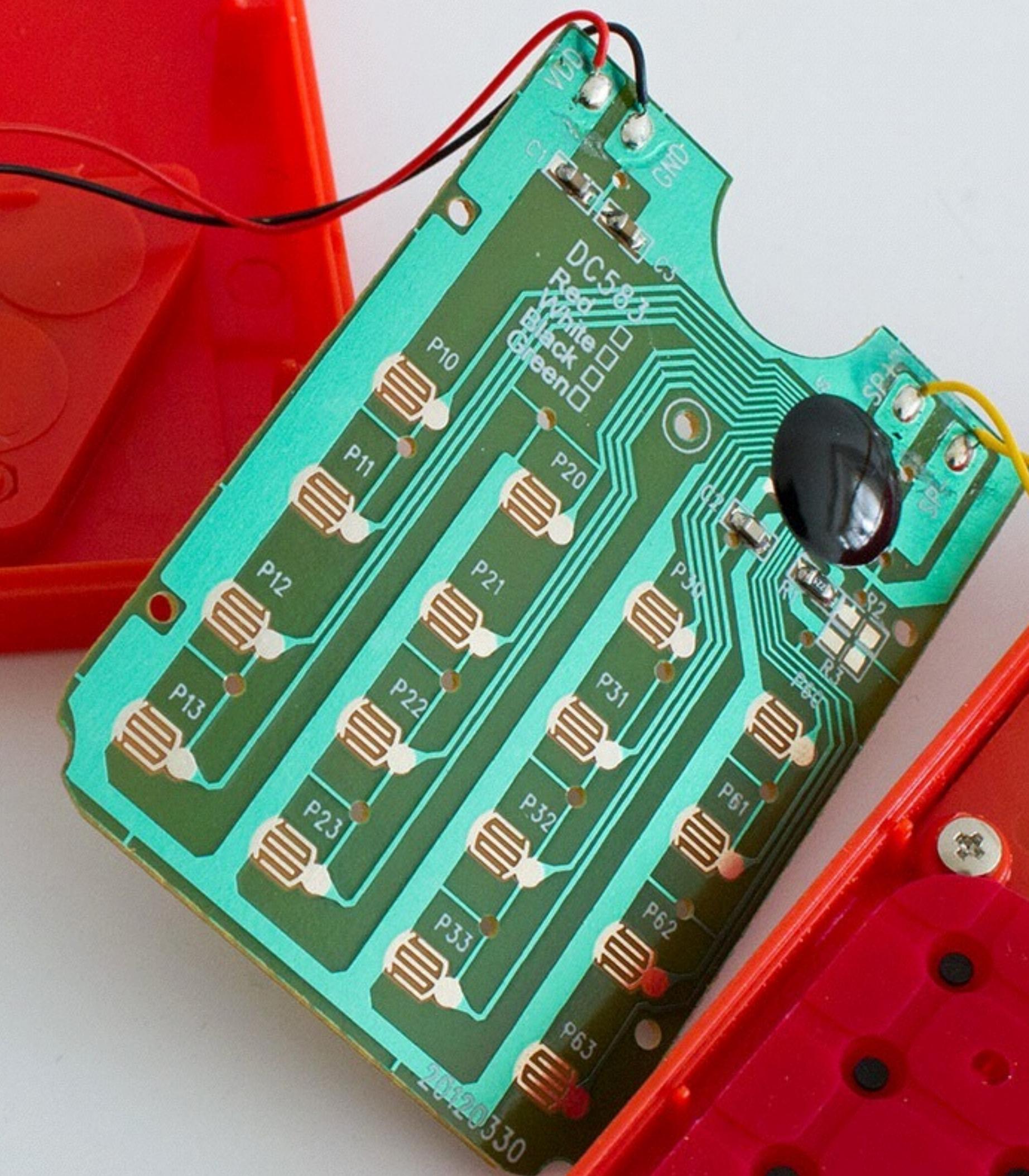
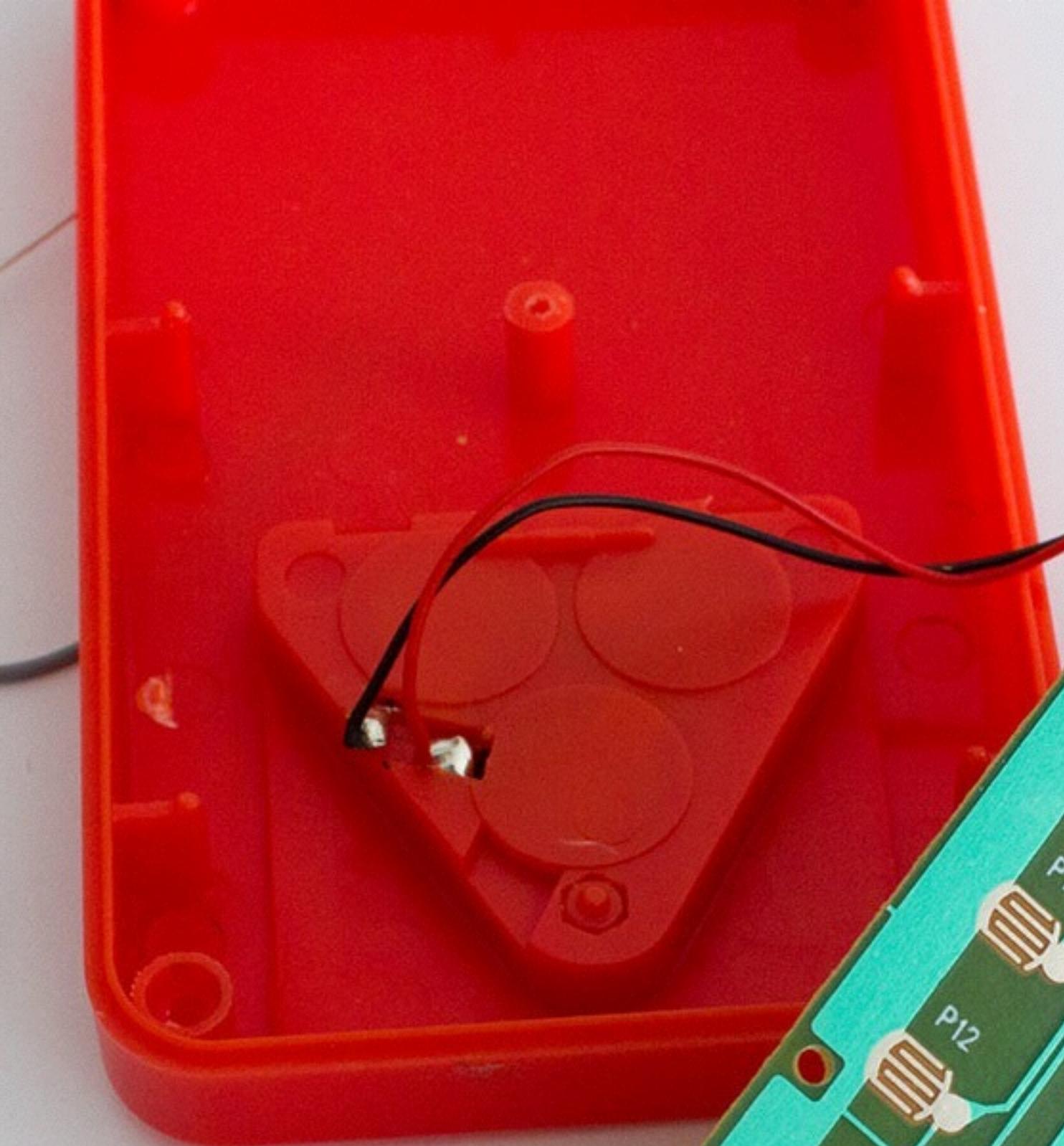
Flexinol

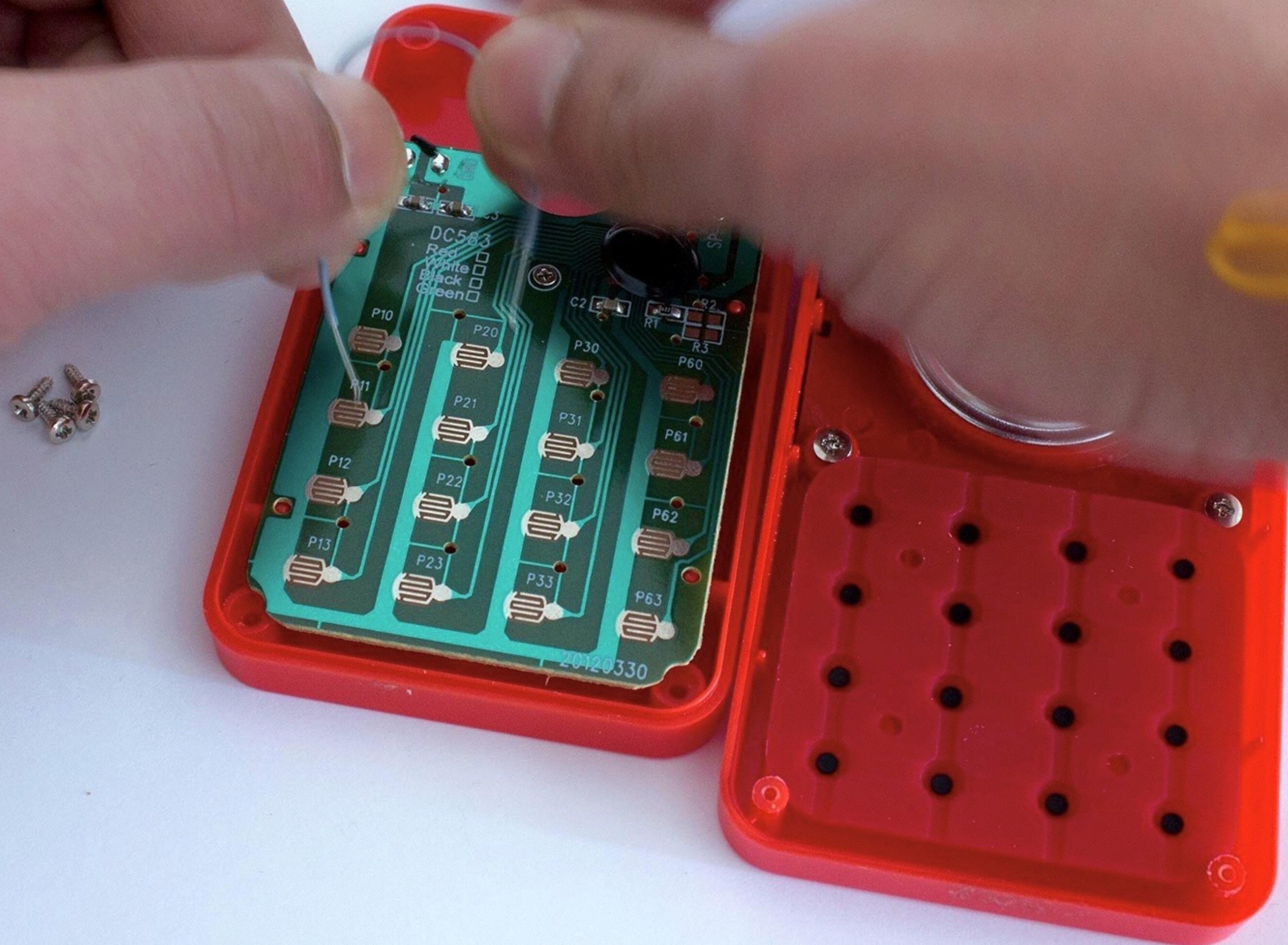


Electronic Origami Flapping Crane

Decontextualisation







Be Bold



Be Patient

Documentation

Documentation

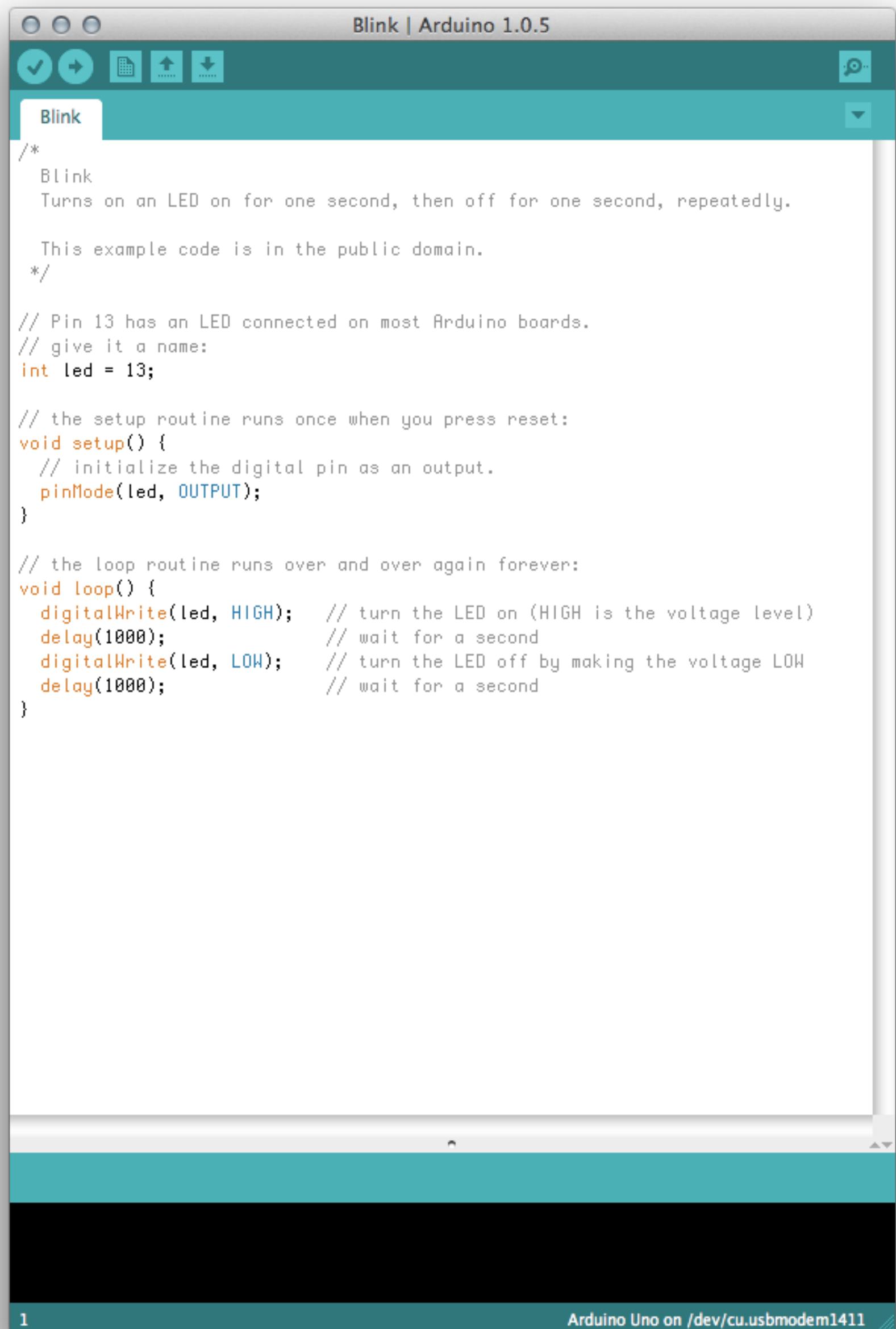
Documentation



fritzing

The logo consists of the word "fritzing" in a bold, white, sans-serif font. The letters are outlined in black and filled with white. There are small black dots at the top and bottom of each letter's vertical stroke. The background is a solid orange rectangle.





The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.0.5". The main window displays the "Blink" sketch. The code is as follows:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
*/

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

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

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

The status bar at the bottom indicates "1" and "Arduino Uno on /dev/cu.usbmodem1411".

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
*/

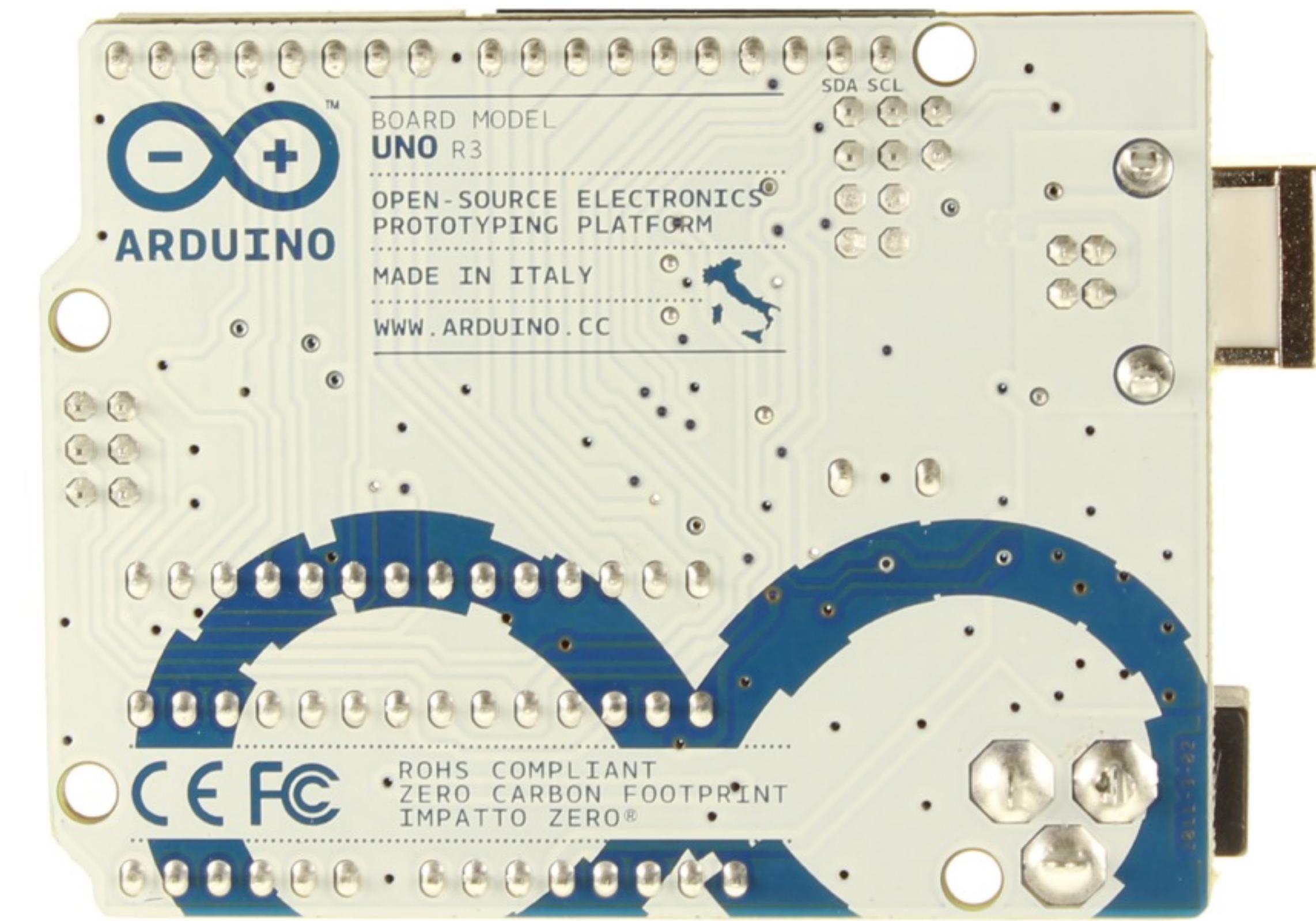
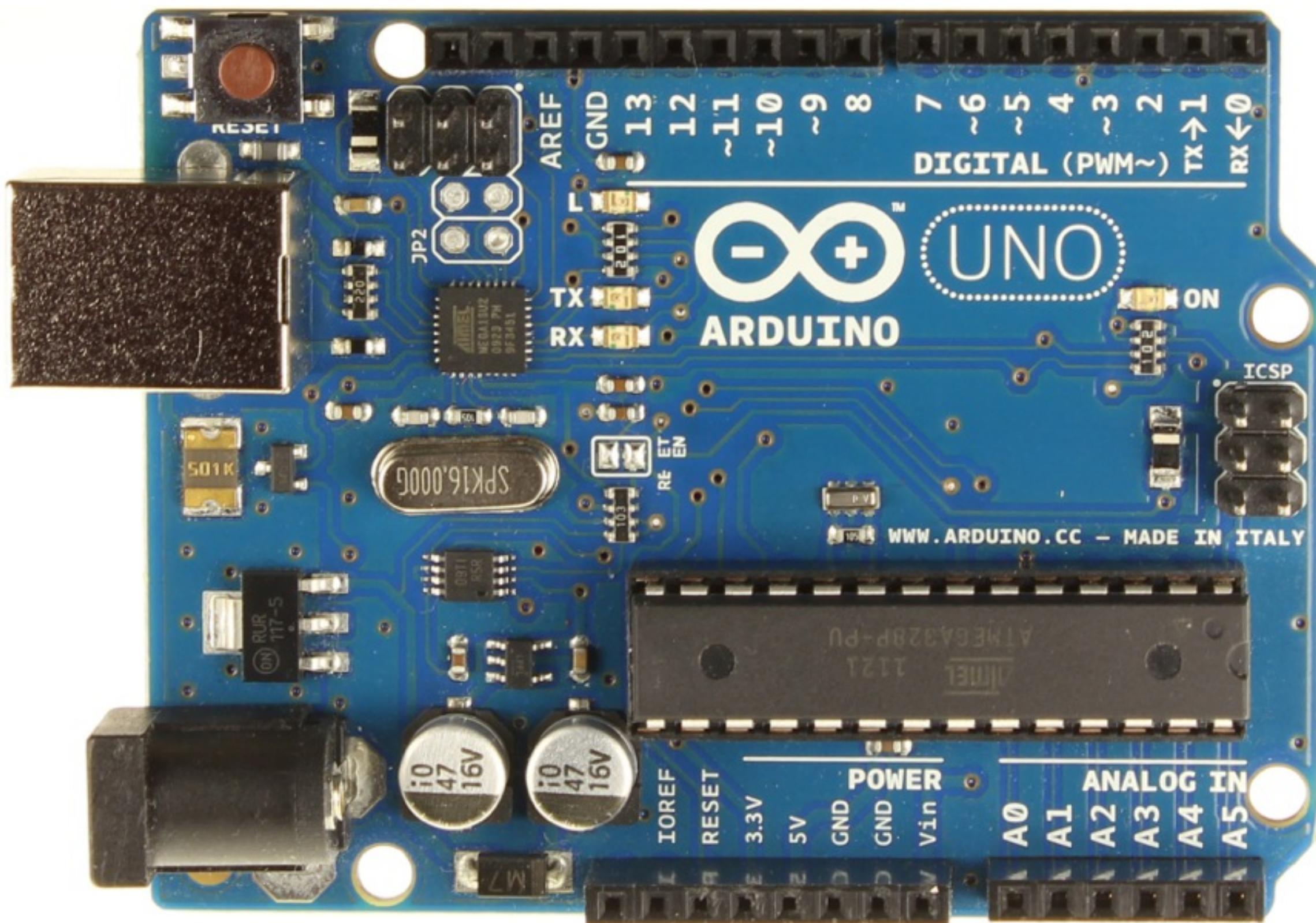
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

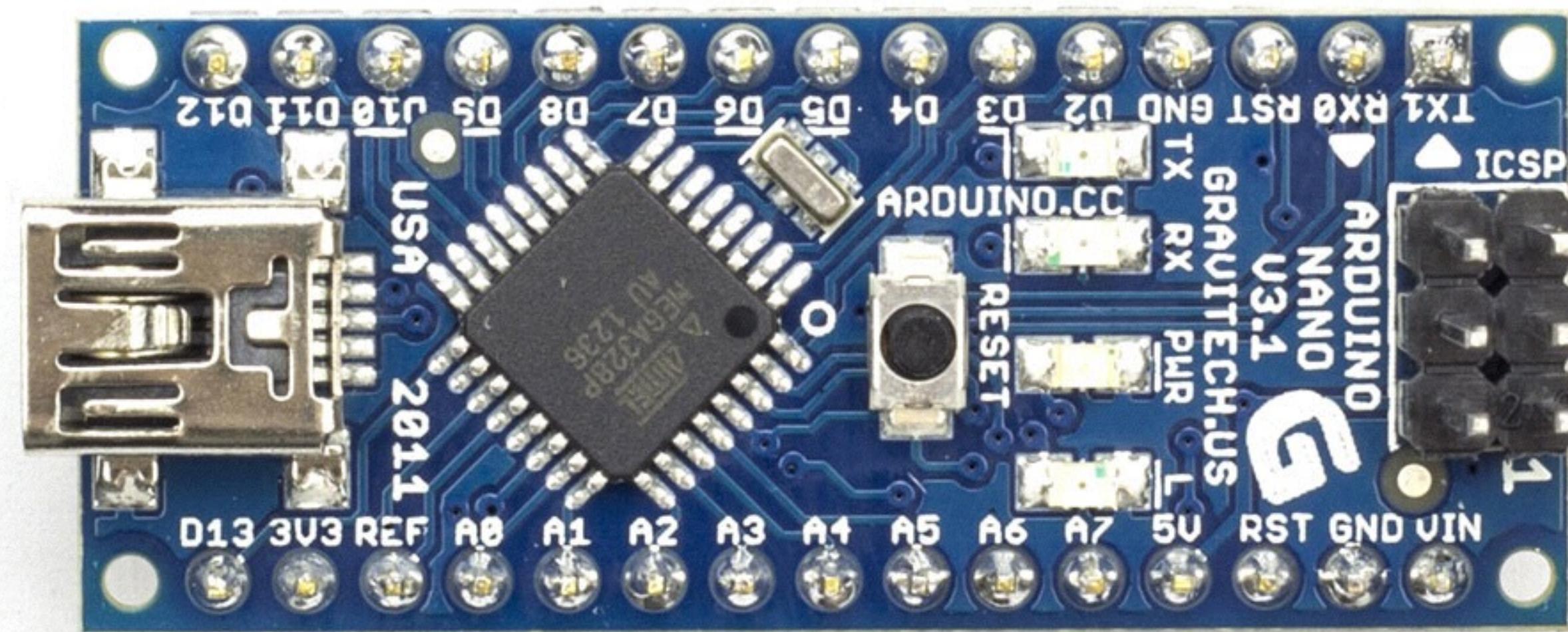
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

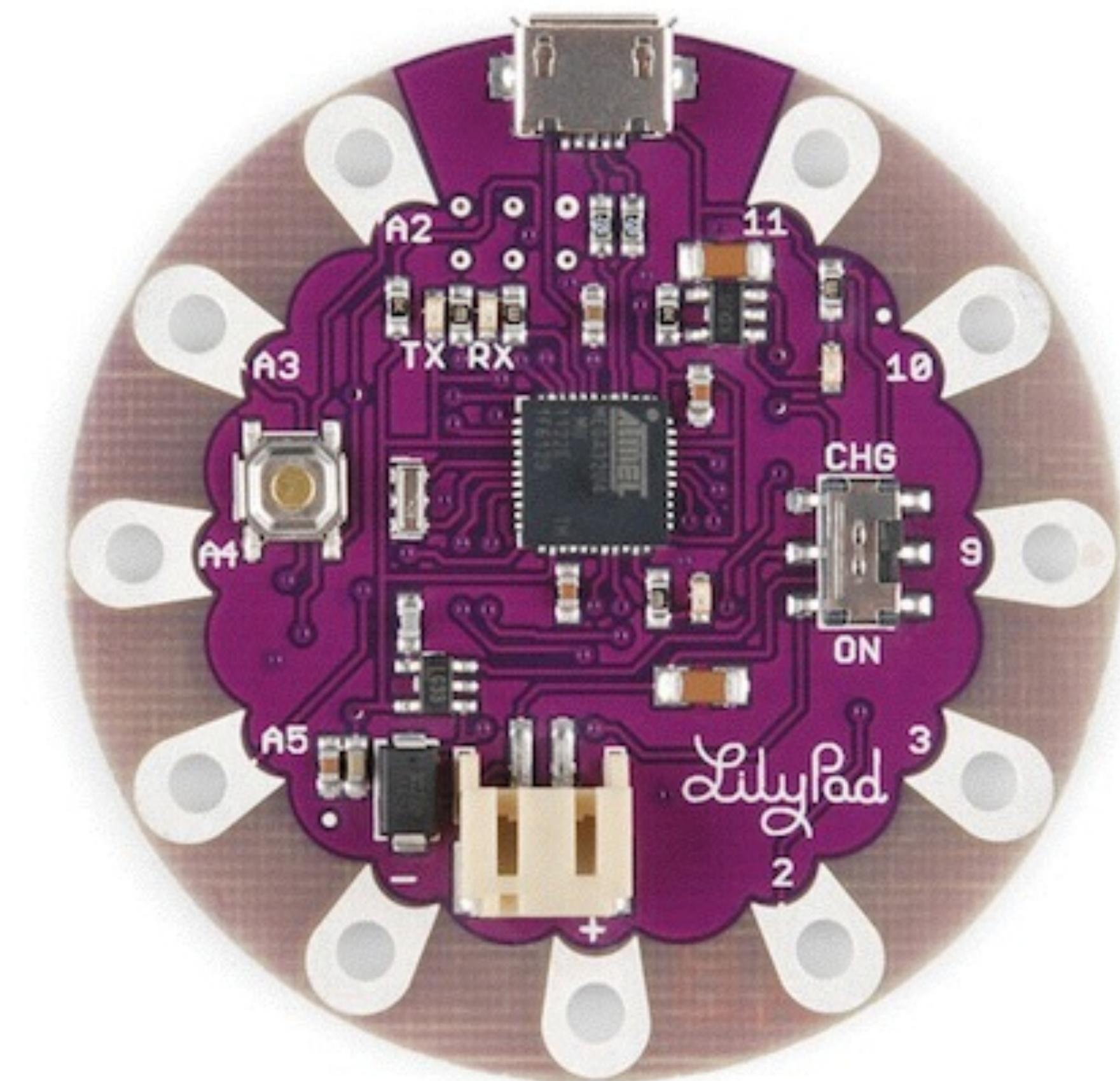
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
  delay(1000);                // wait for a second
}
```

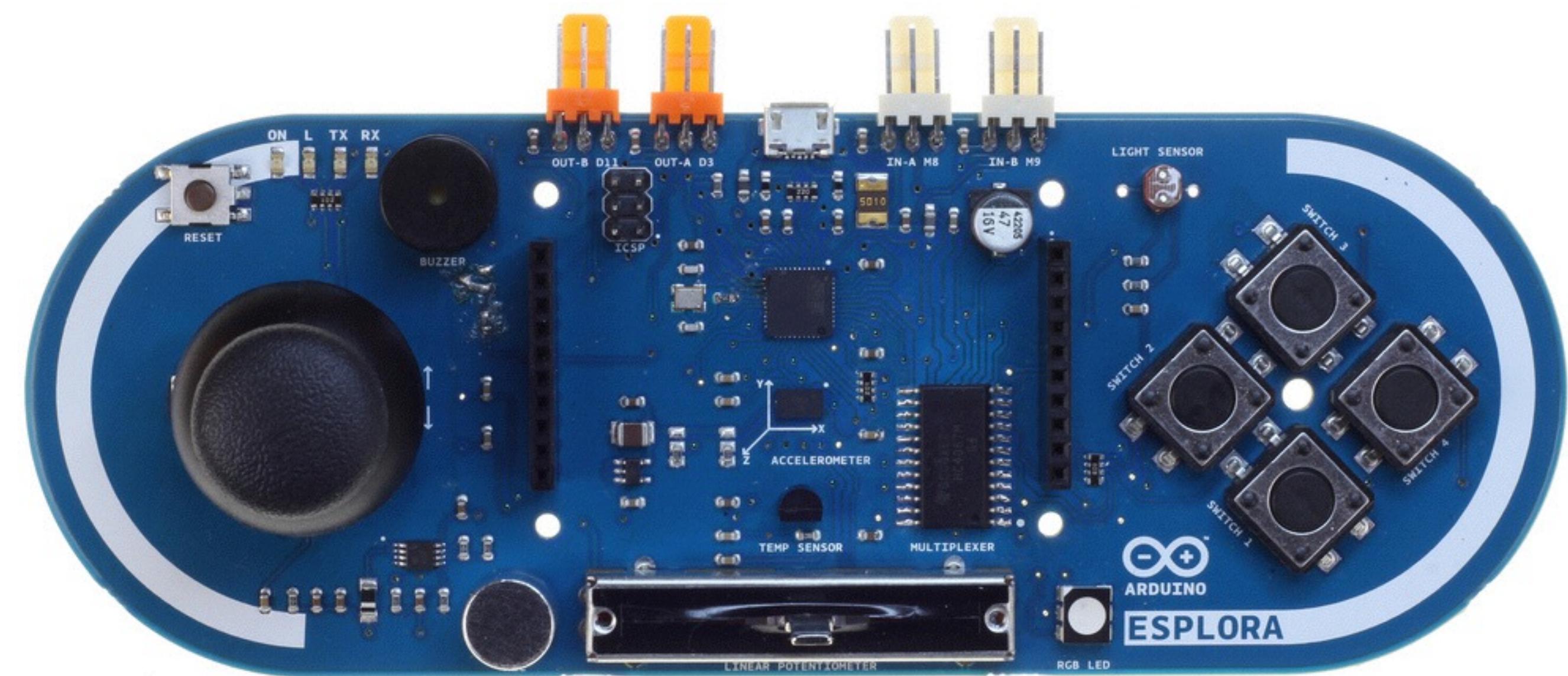
The Arduino language is based on C/C++.

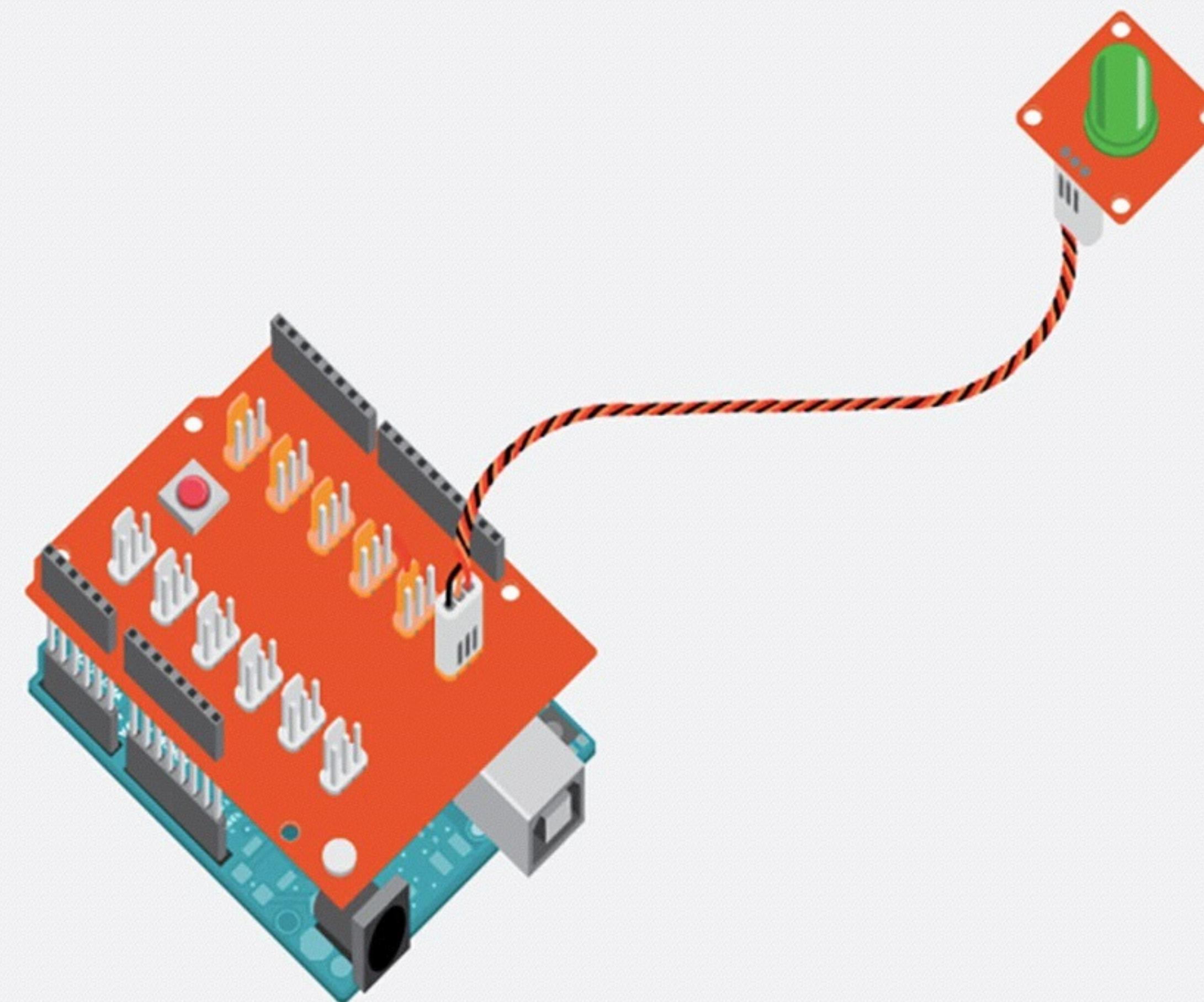
It links against AVR Libc and allows the use of any of its functions.

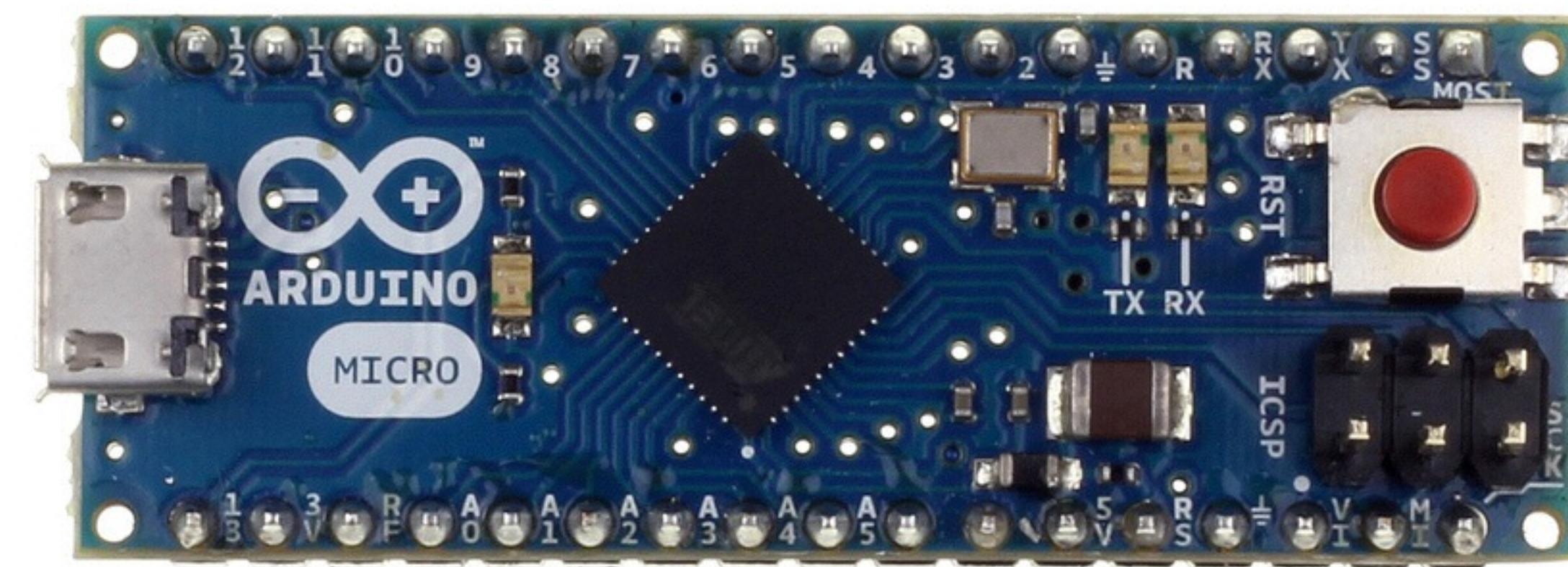


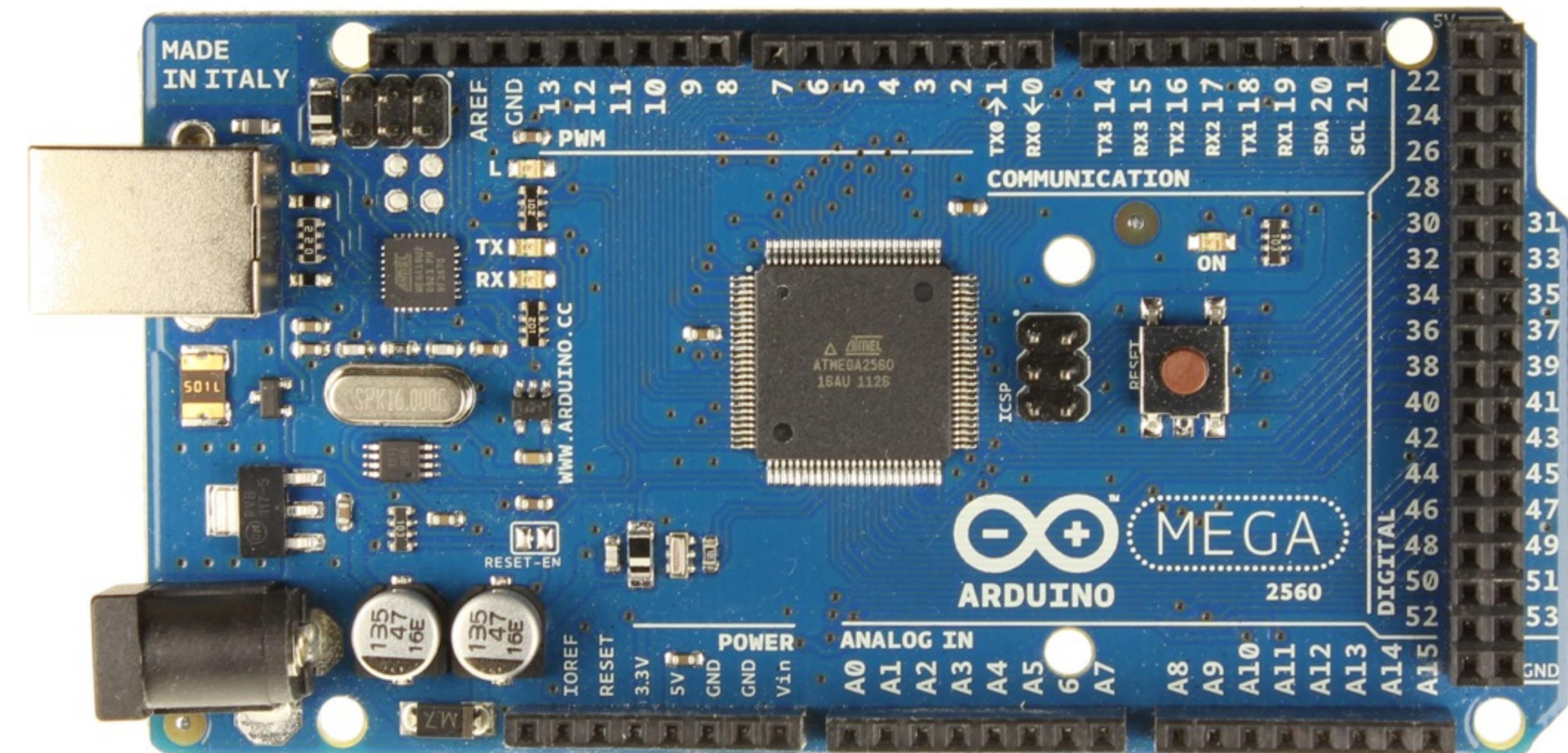


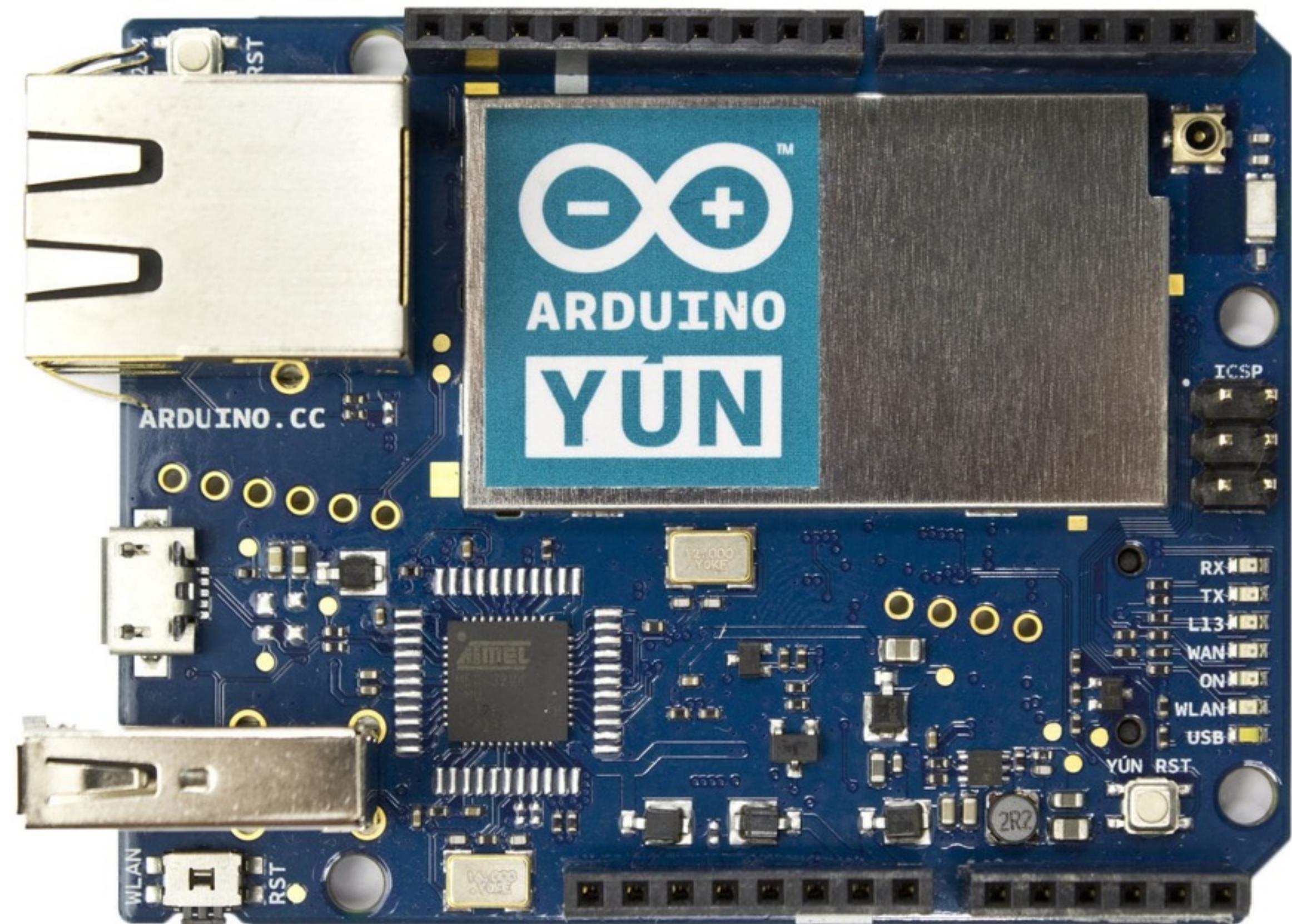


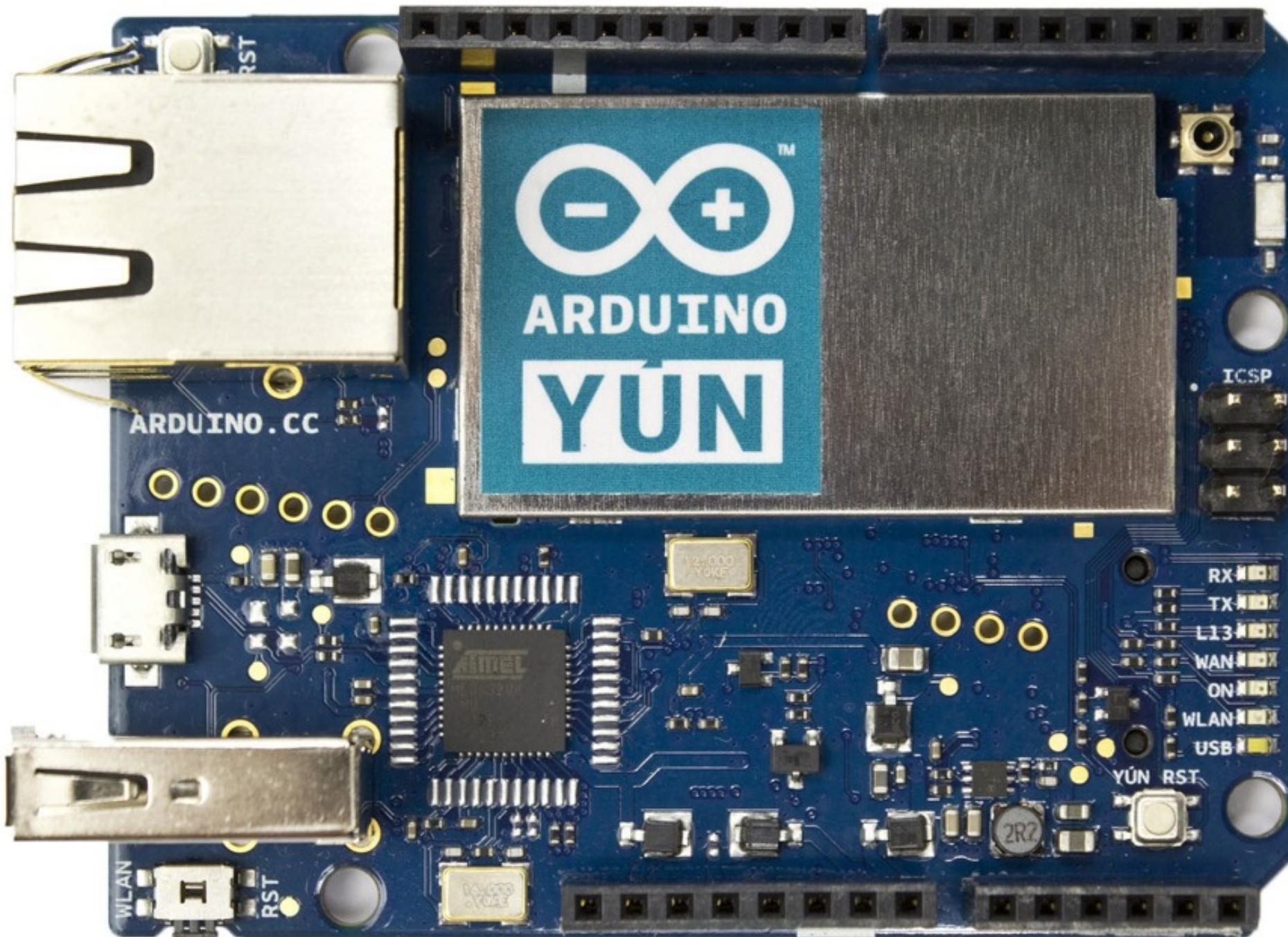












Arduino :: Temboo

We're Hiring

LIBRARY DOWNLOAD SUPPORT SIGN UP LOG IN

TEMBOO

ARDUINO + TEMBOO

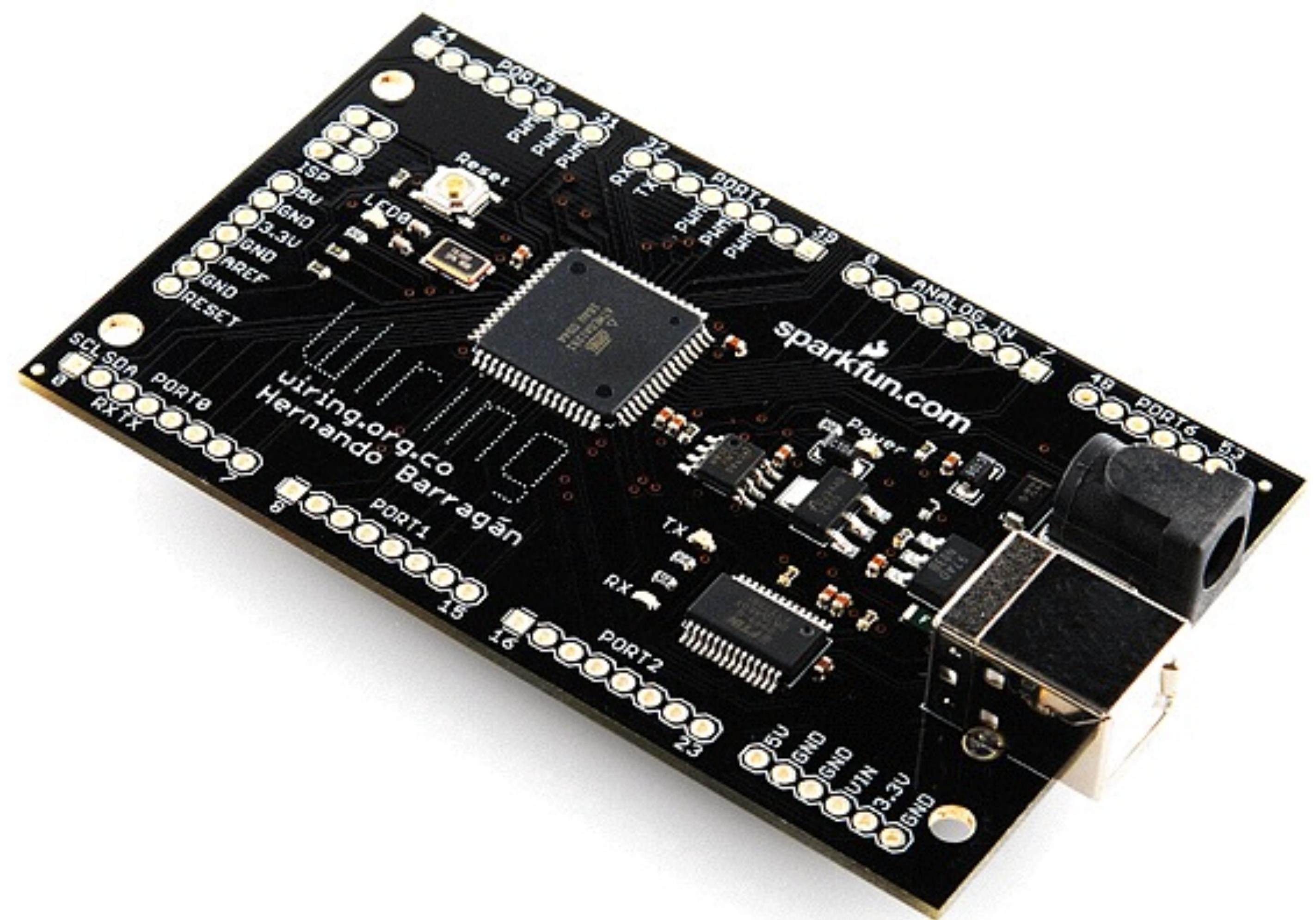
The Arduino Yún comes loaded with the power of Temboo, making it simple to connect with a vast array of web-based resources and services.

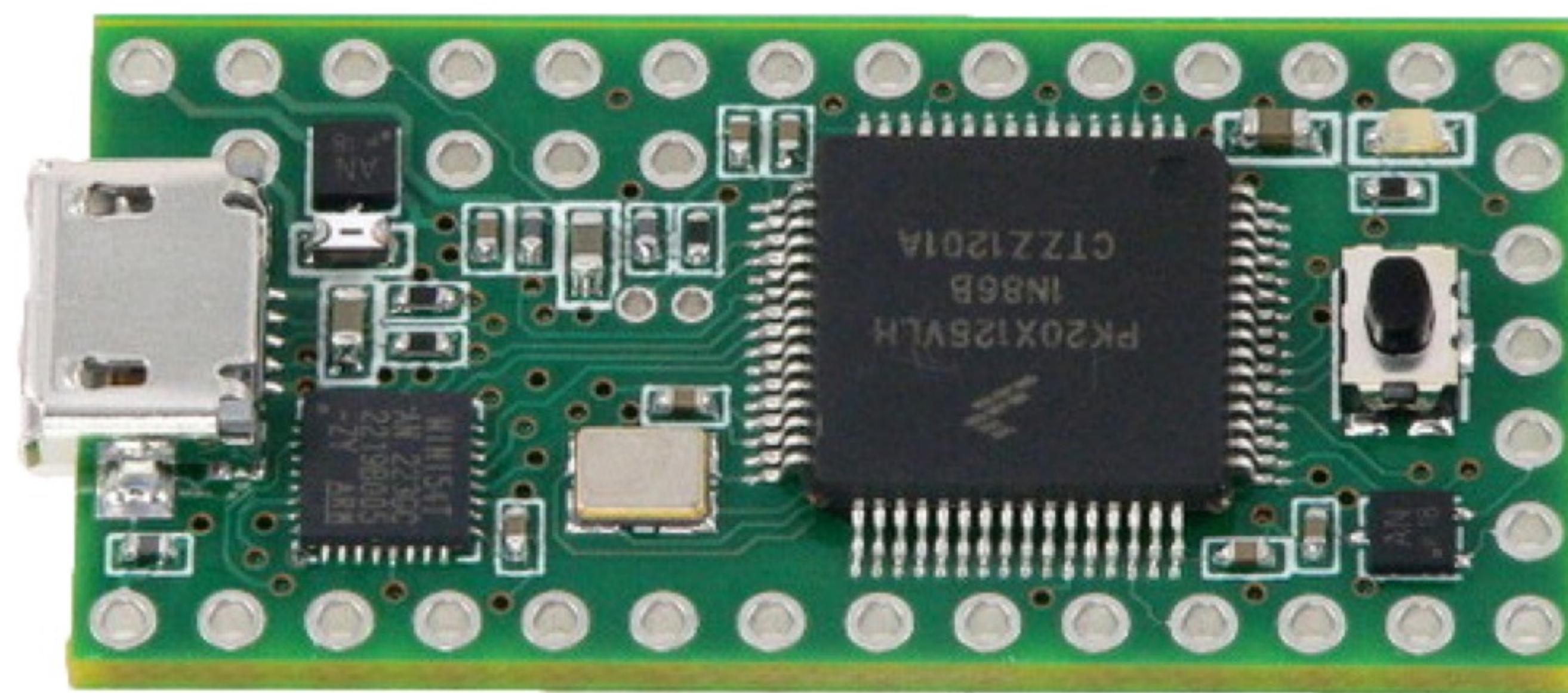
Get Started: How to Temboo with your Yún

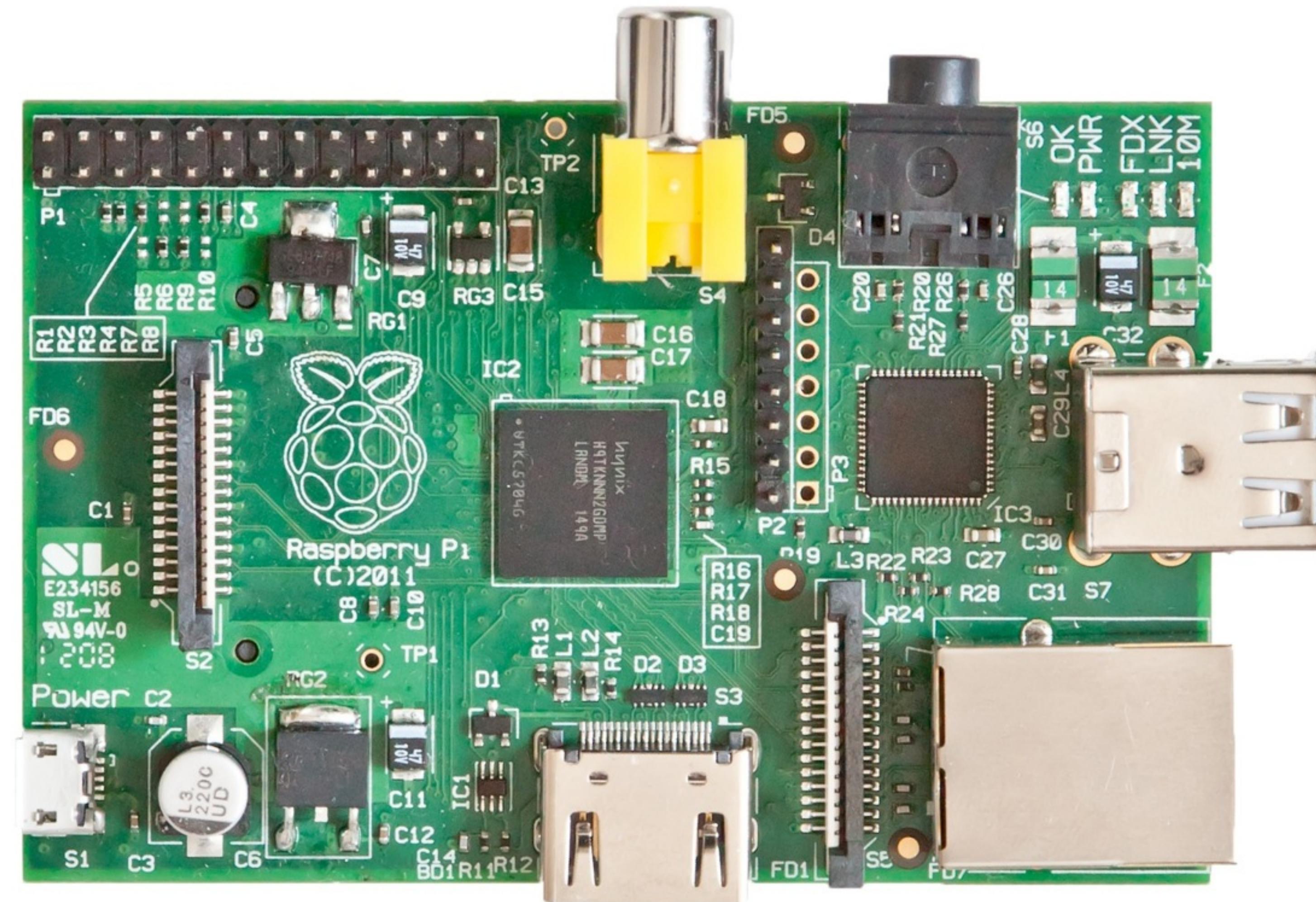
See what your Yún can do in just a few minutes.

Post data to a Google Spreadsheet

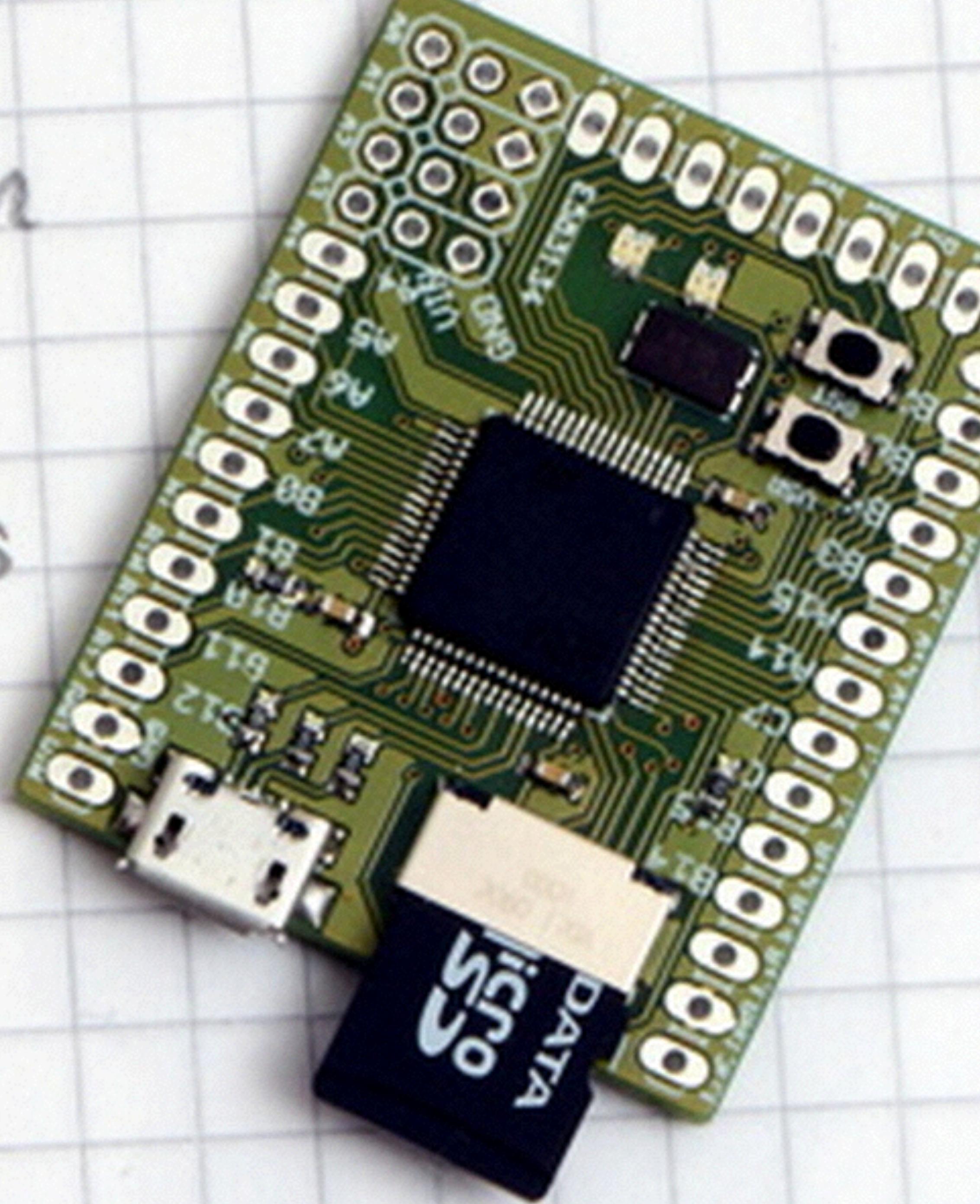
Read your Twitter timeline

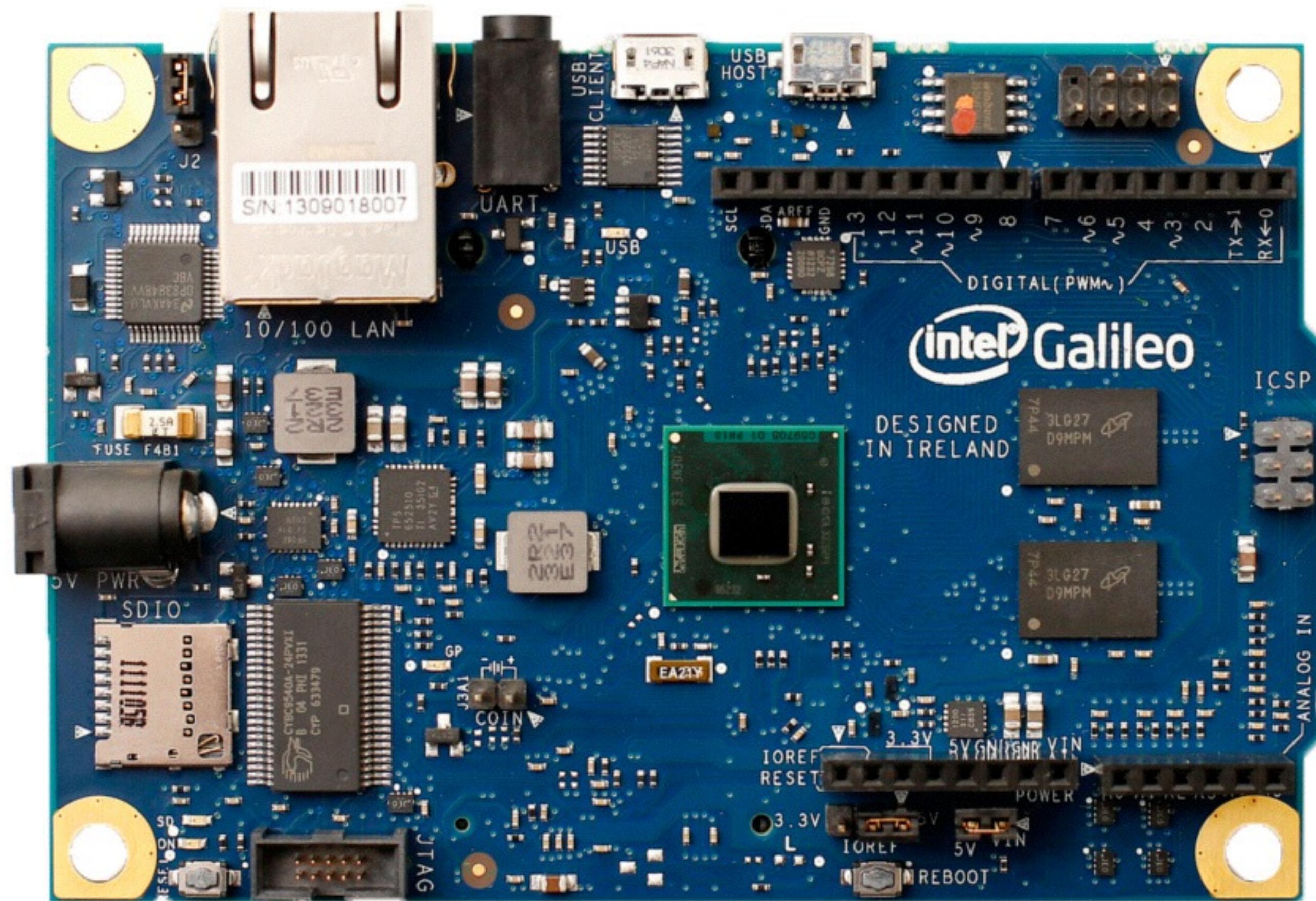


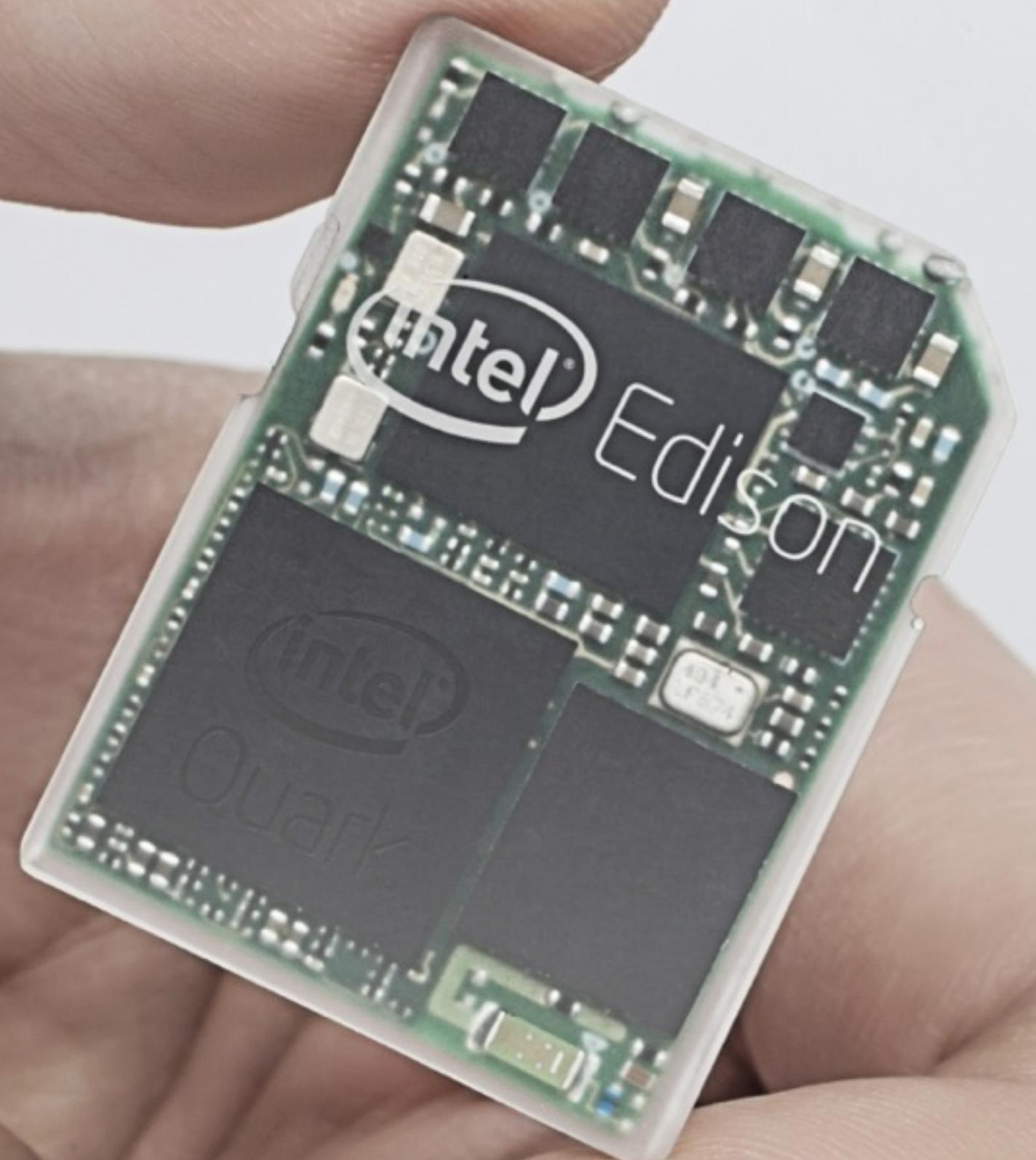


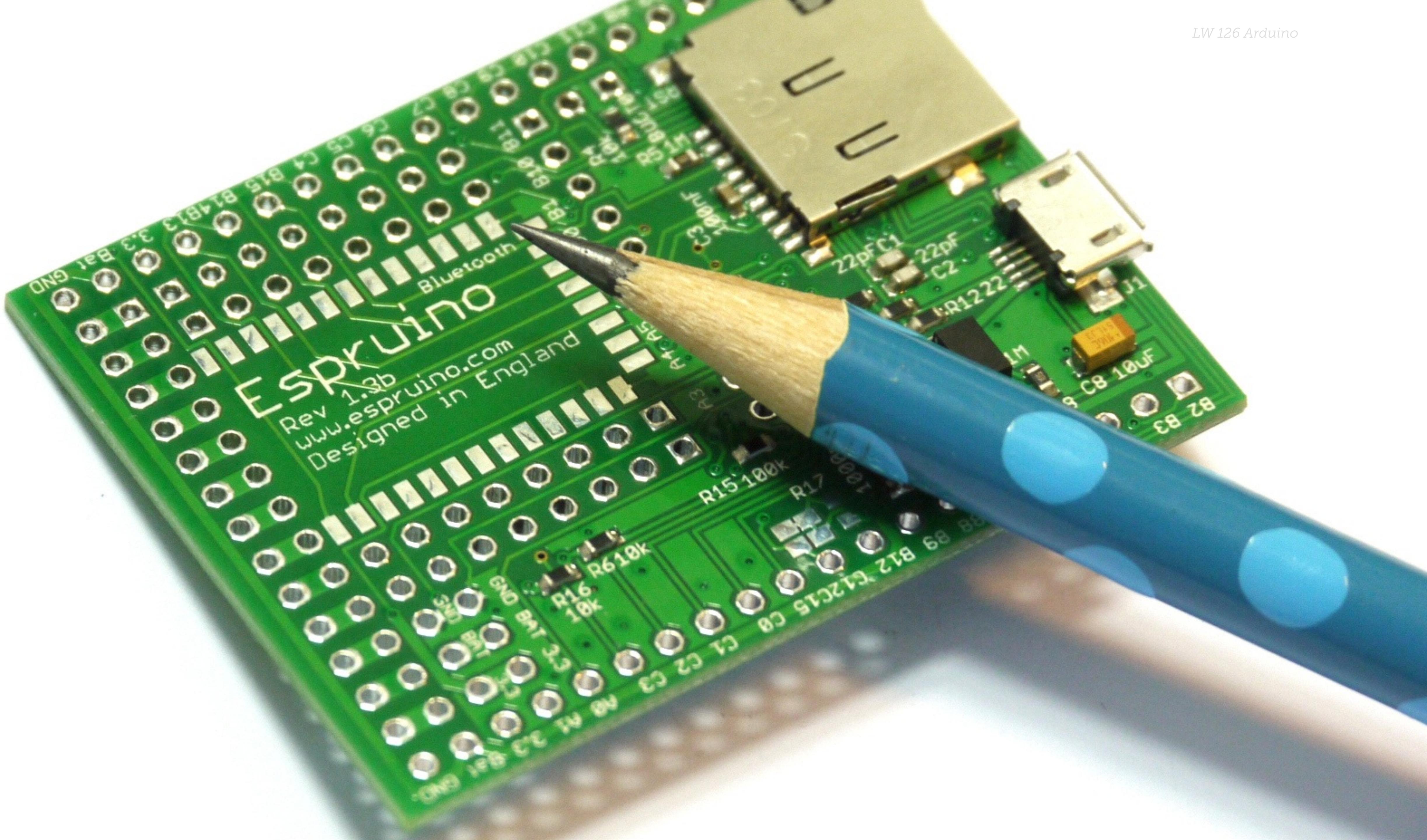


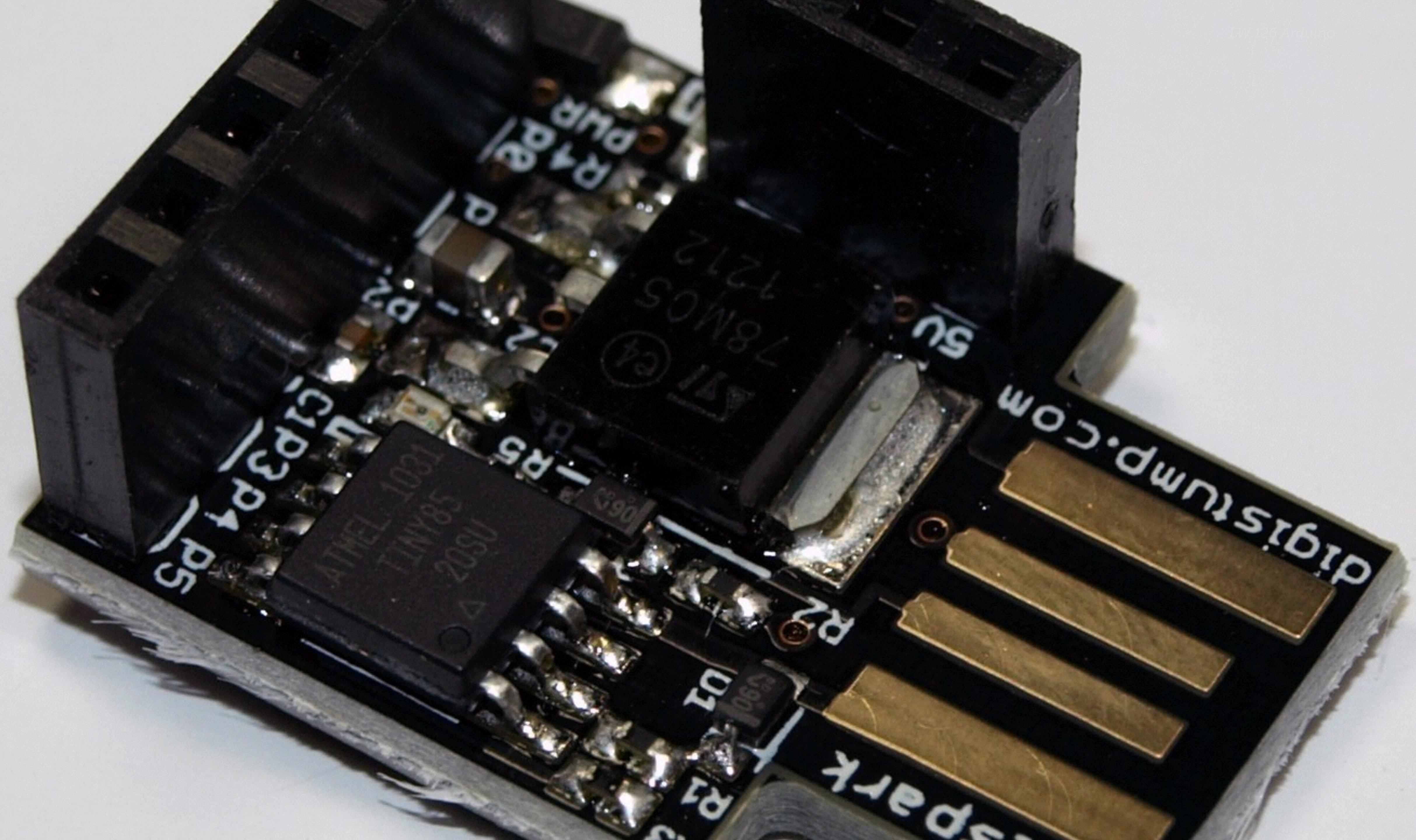
Micro Python
Python for
microcontrollers

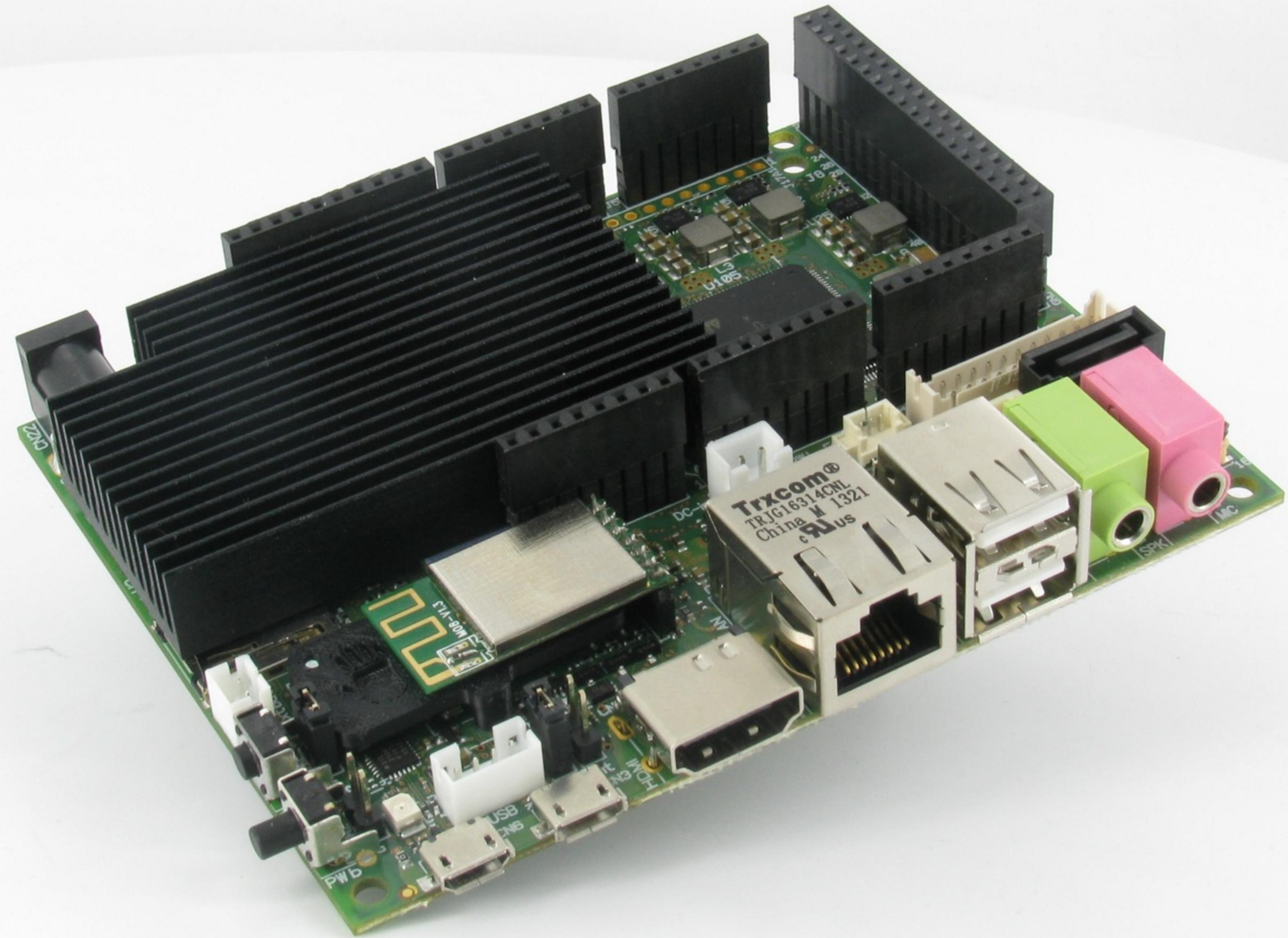




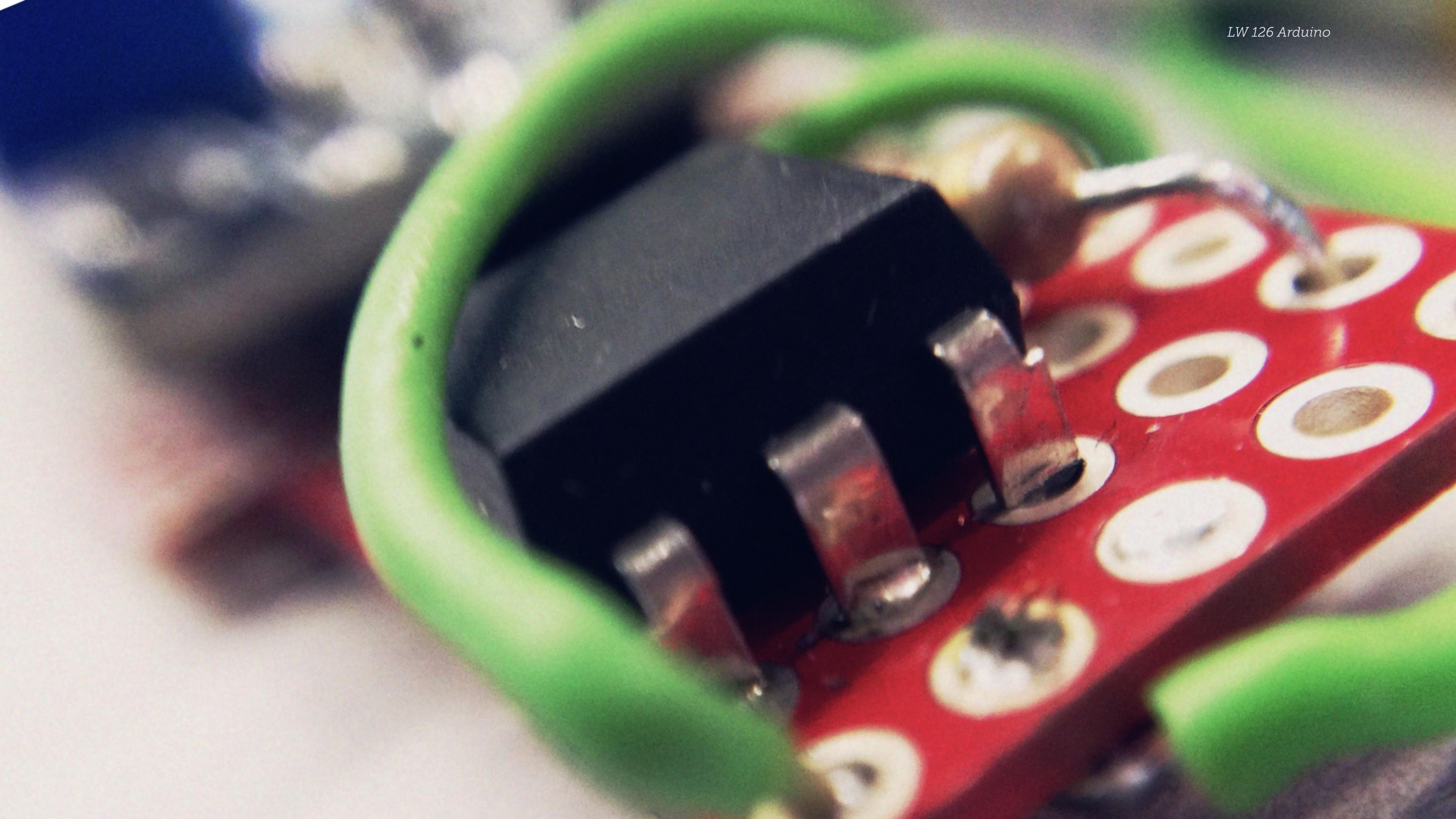


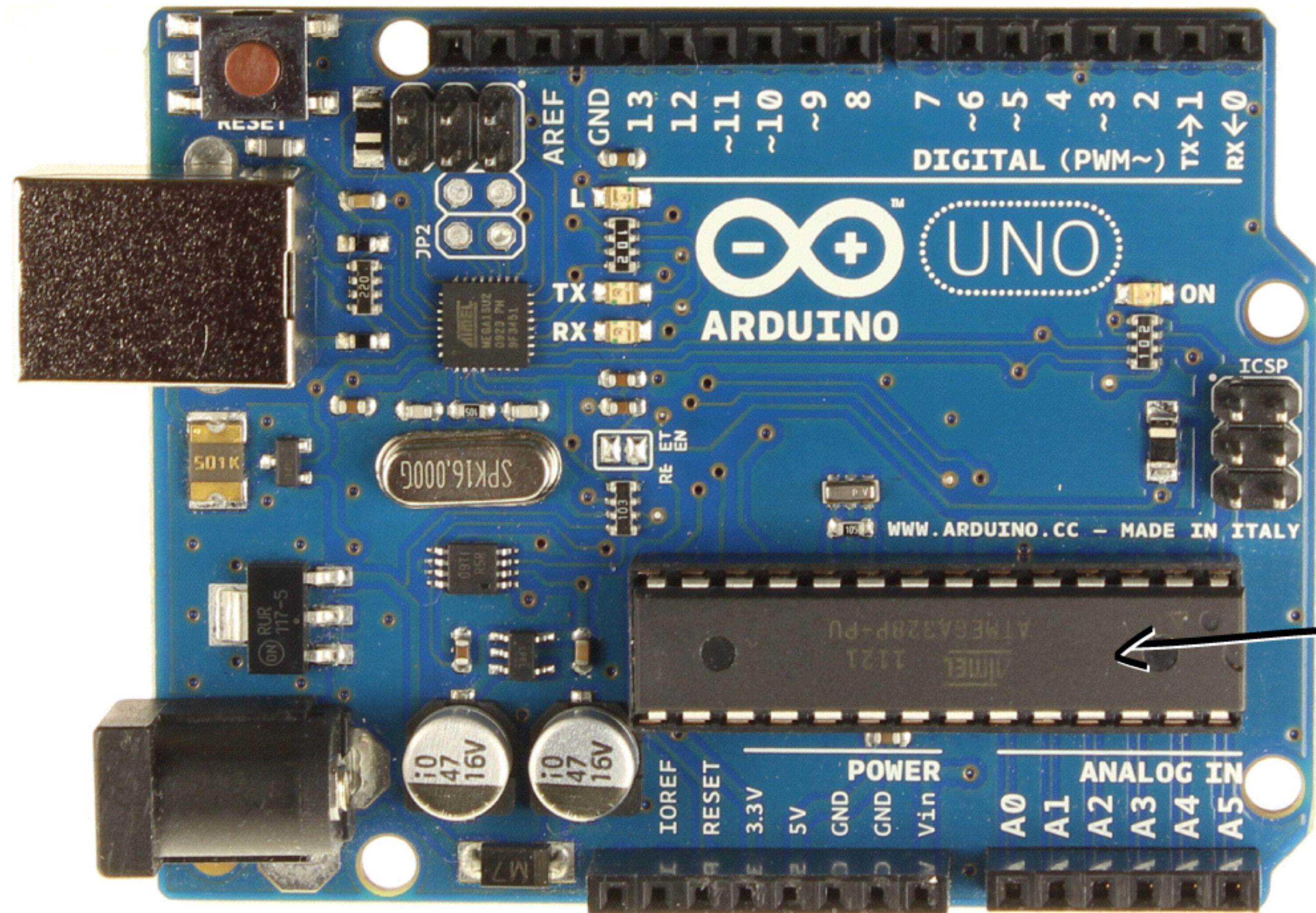




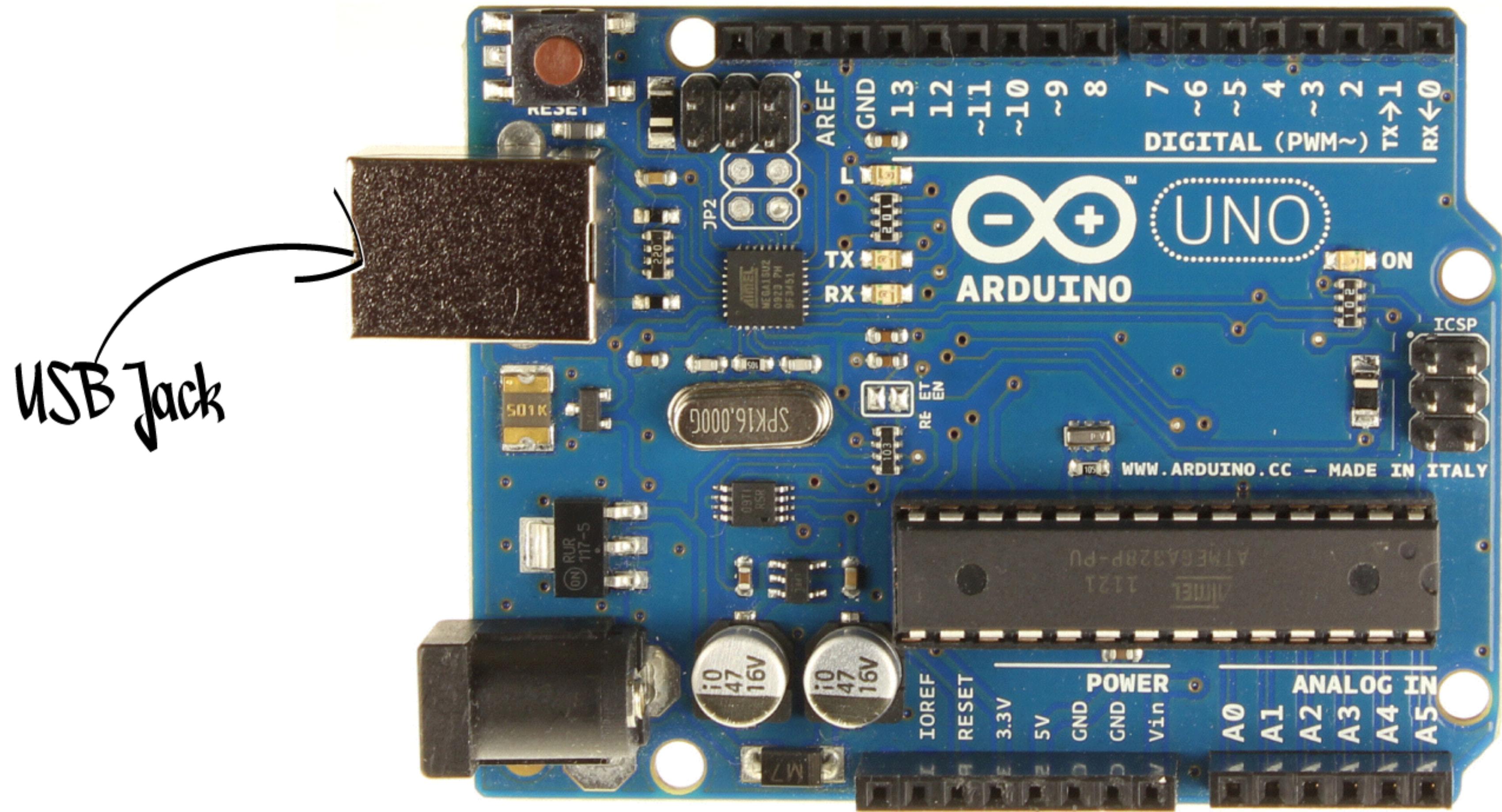


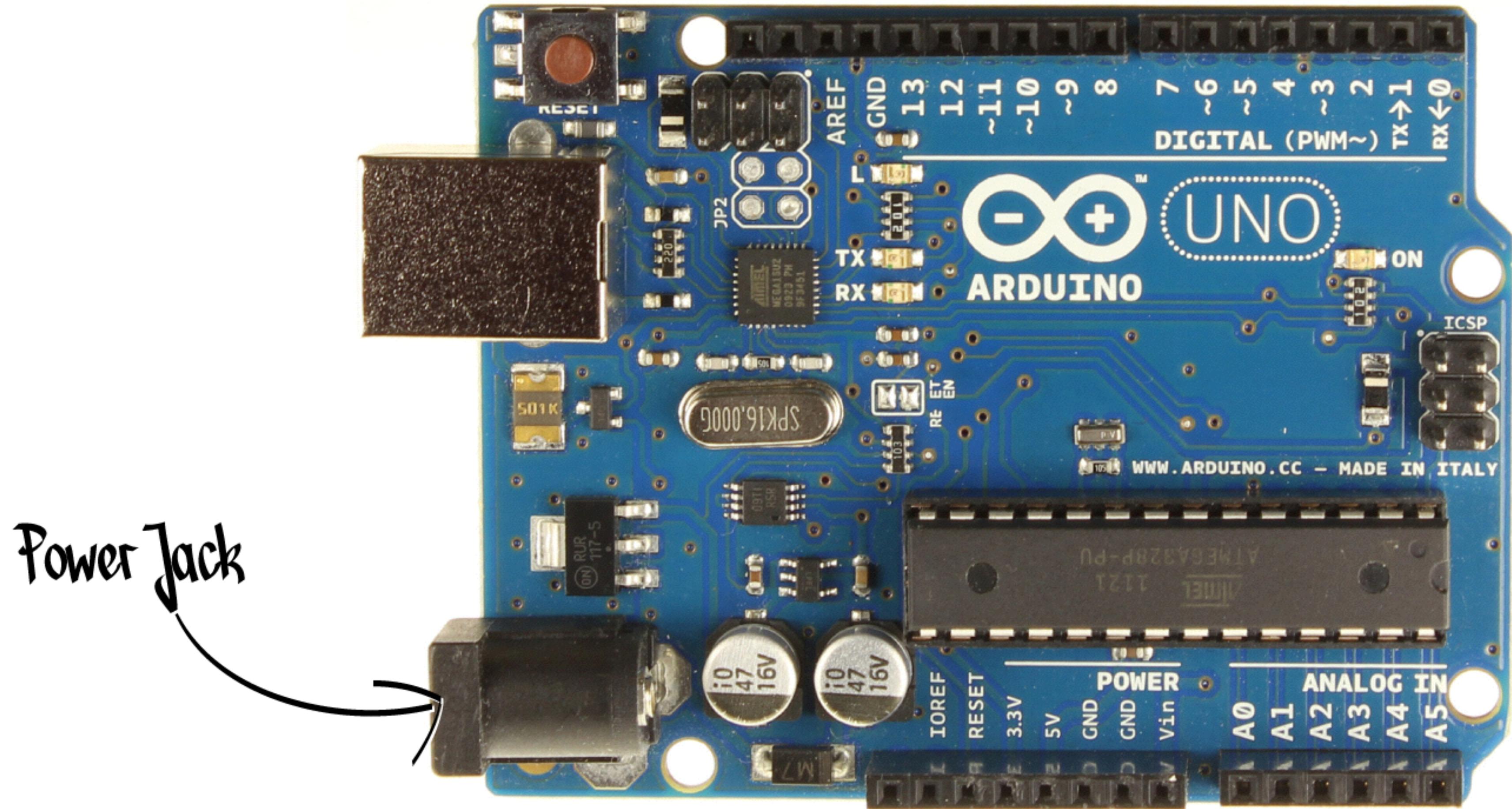


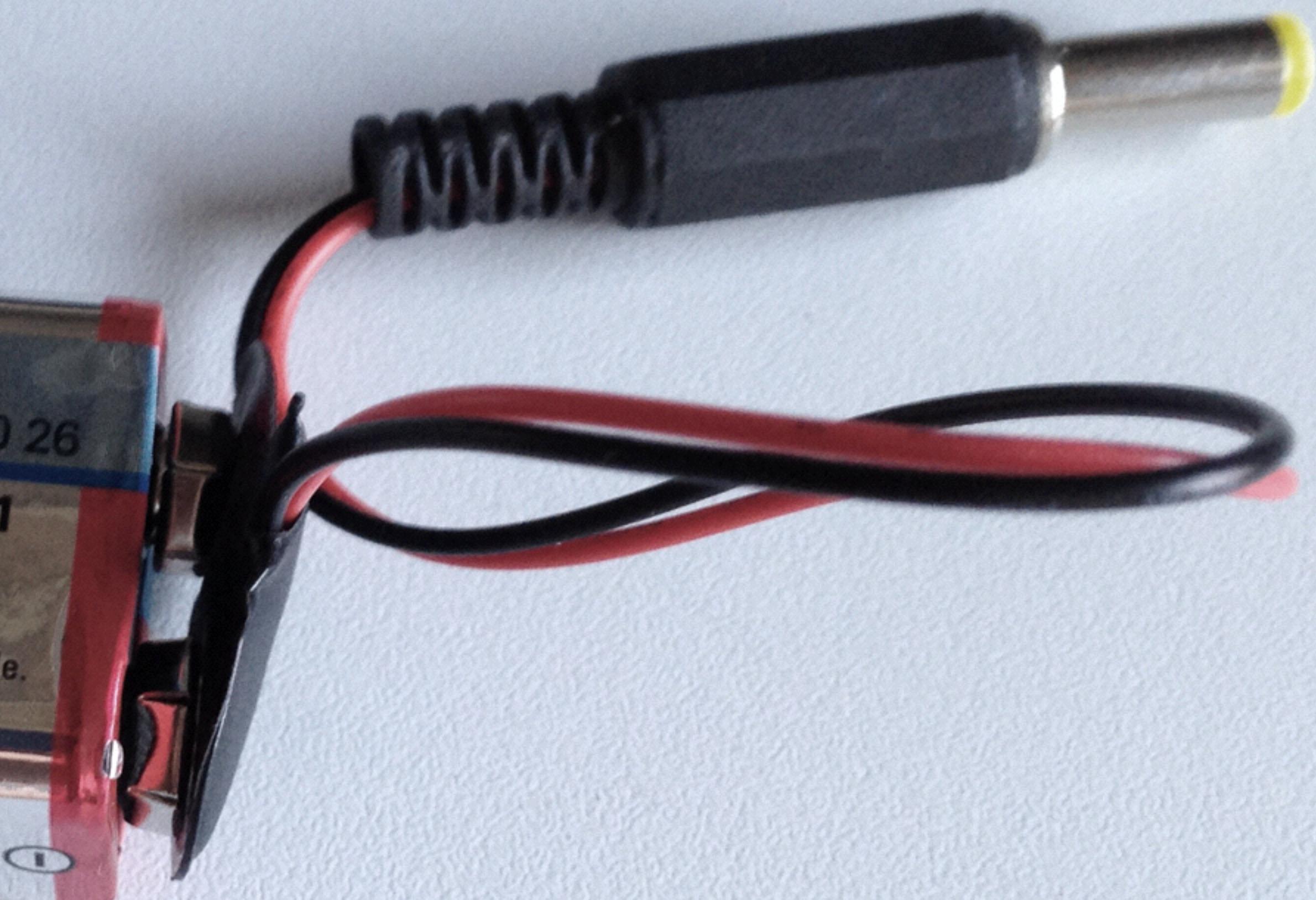


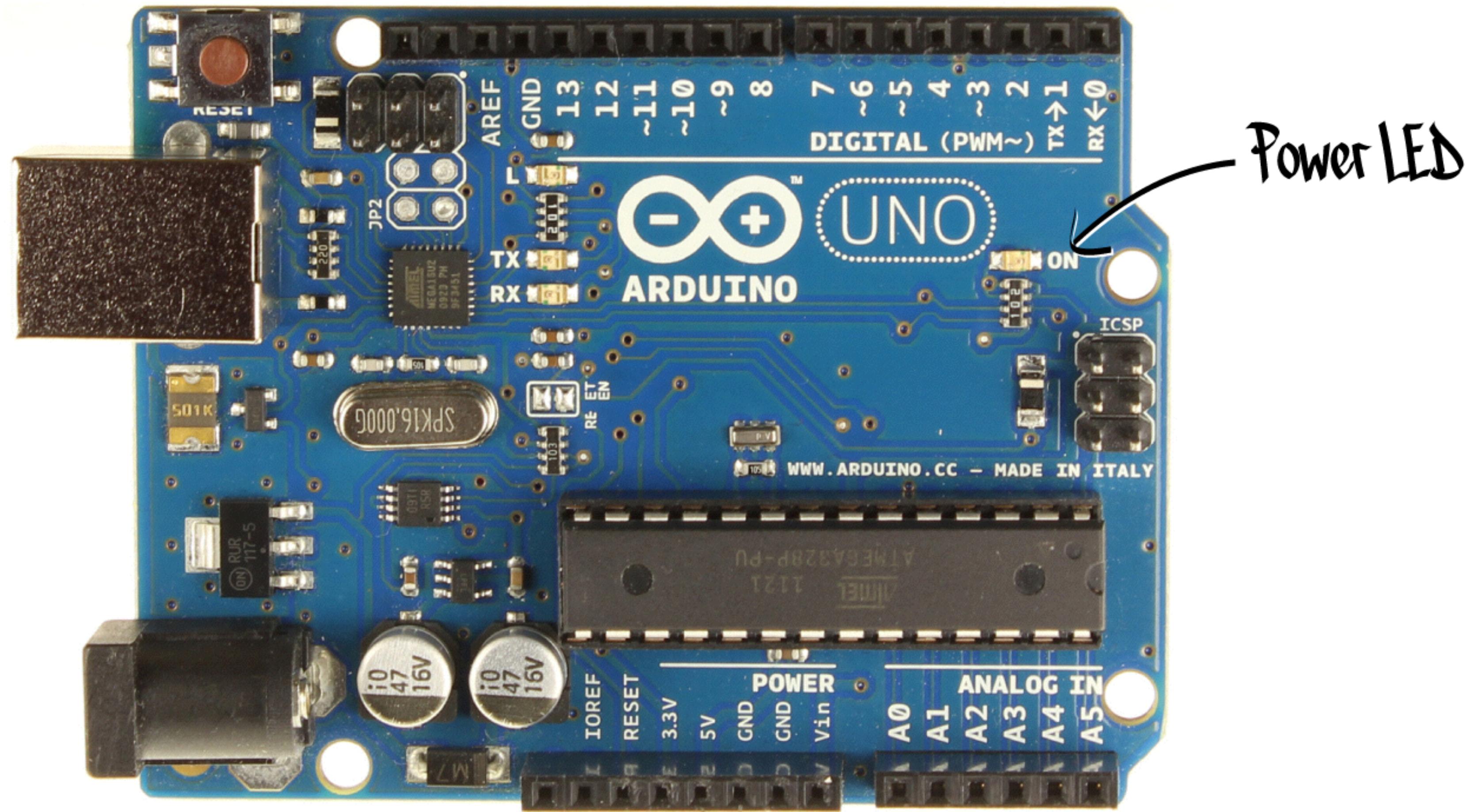


Microcontroller

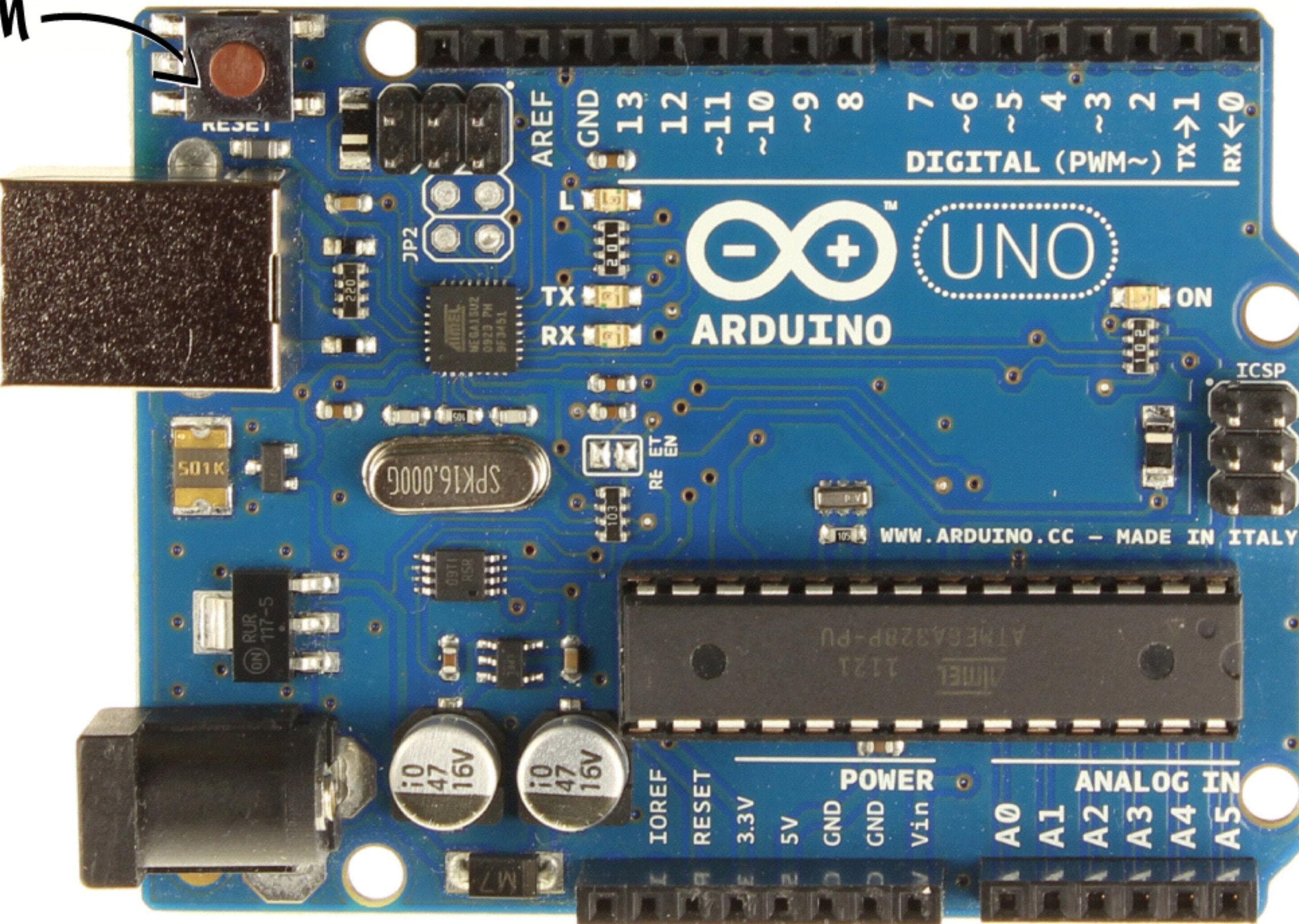




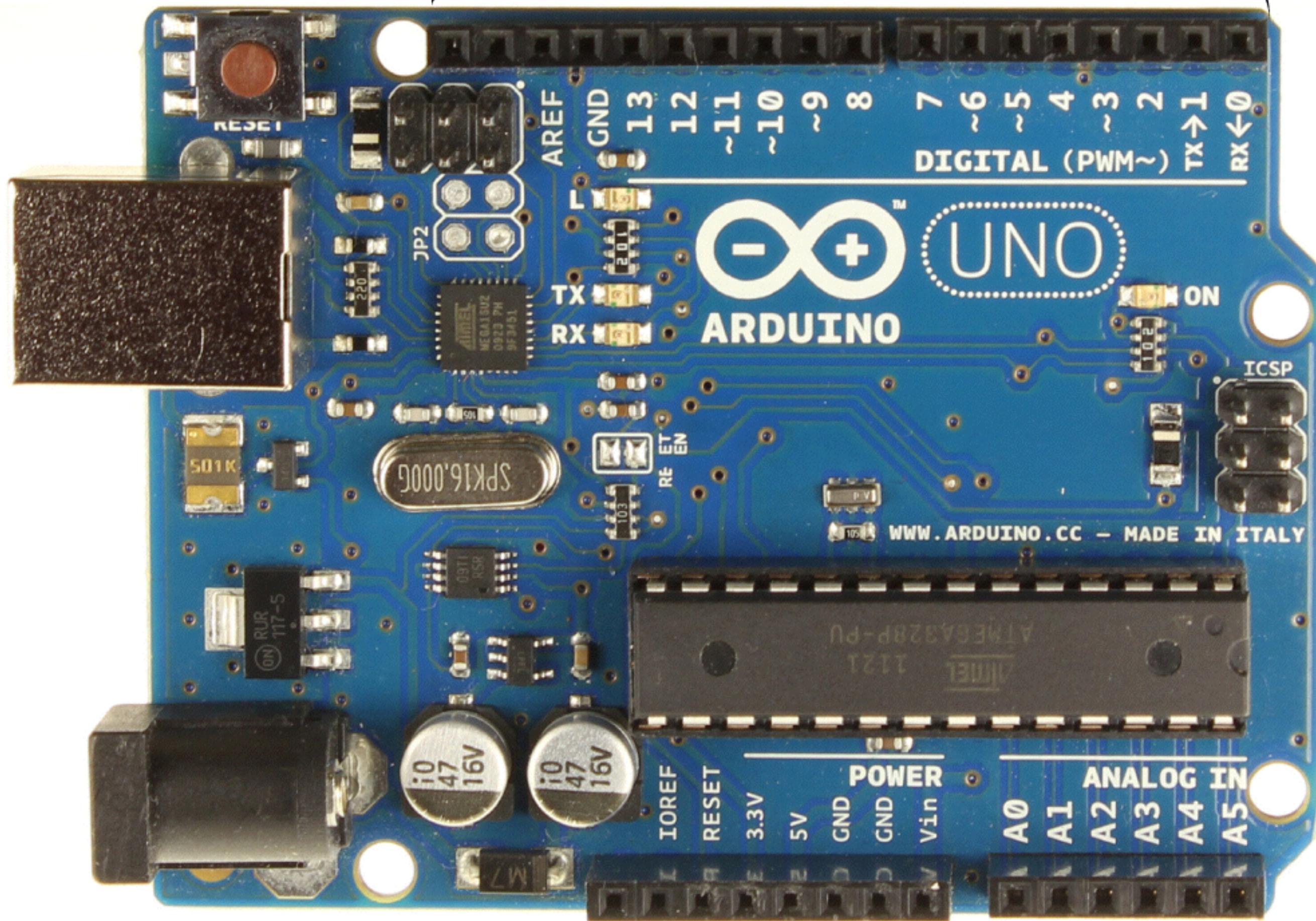




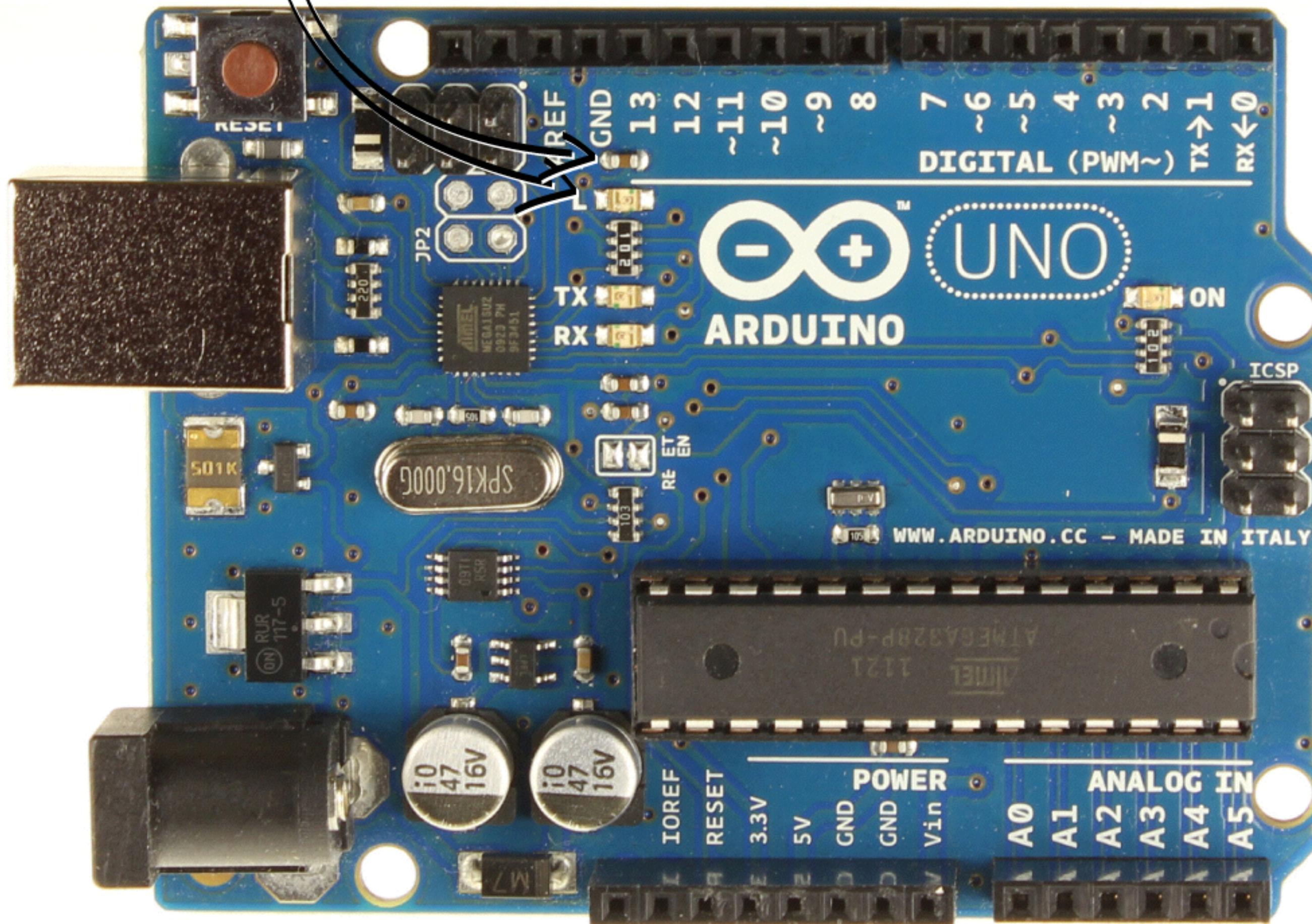
Reset Button

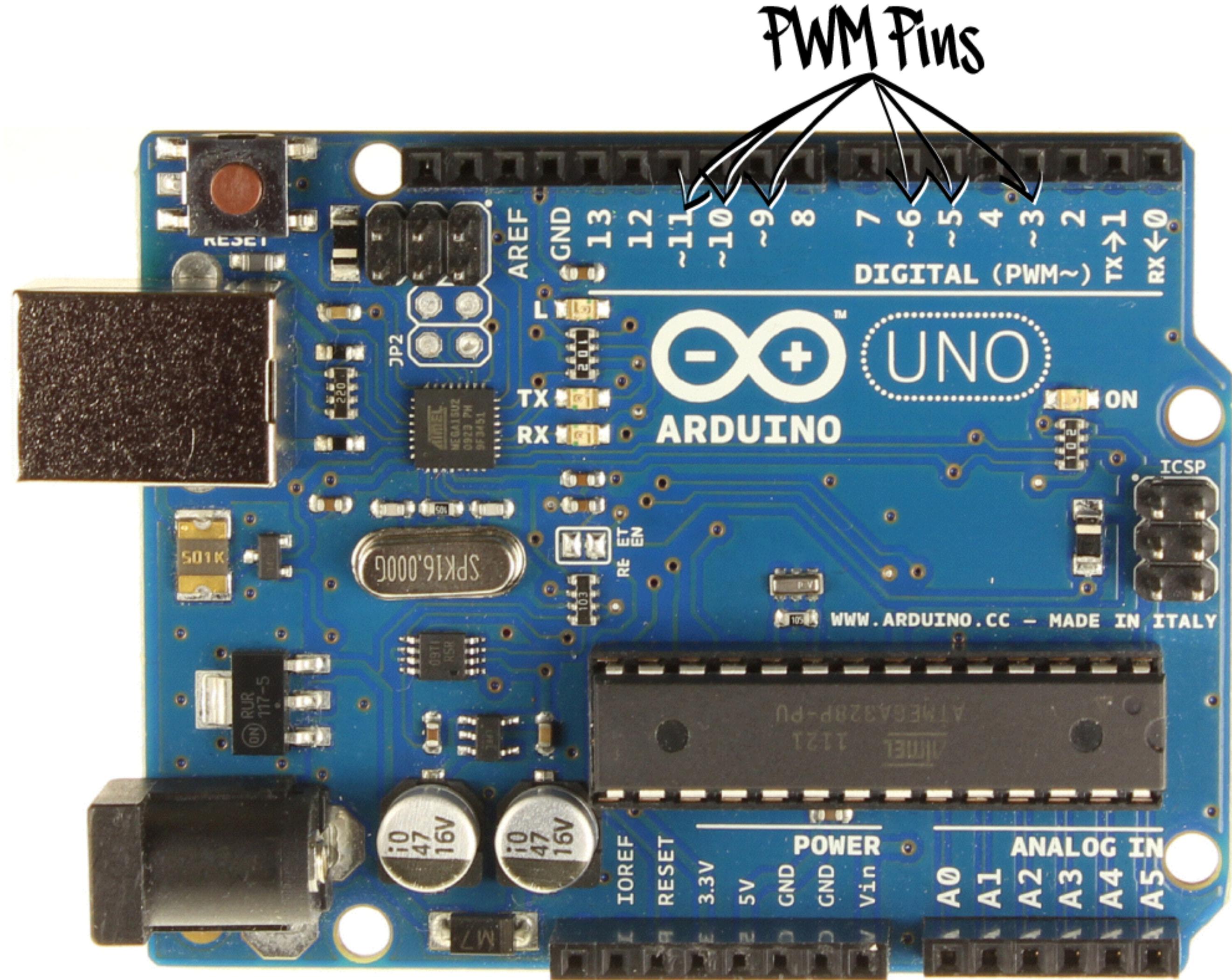


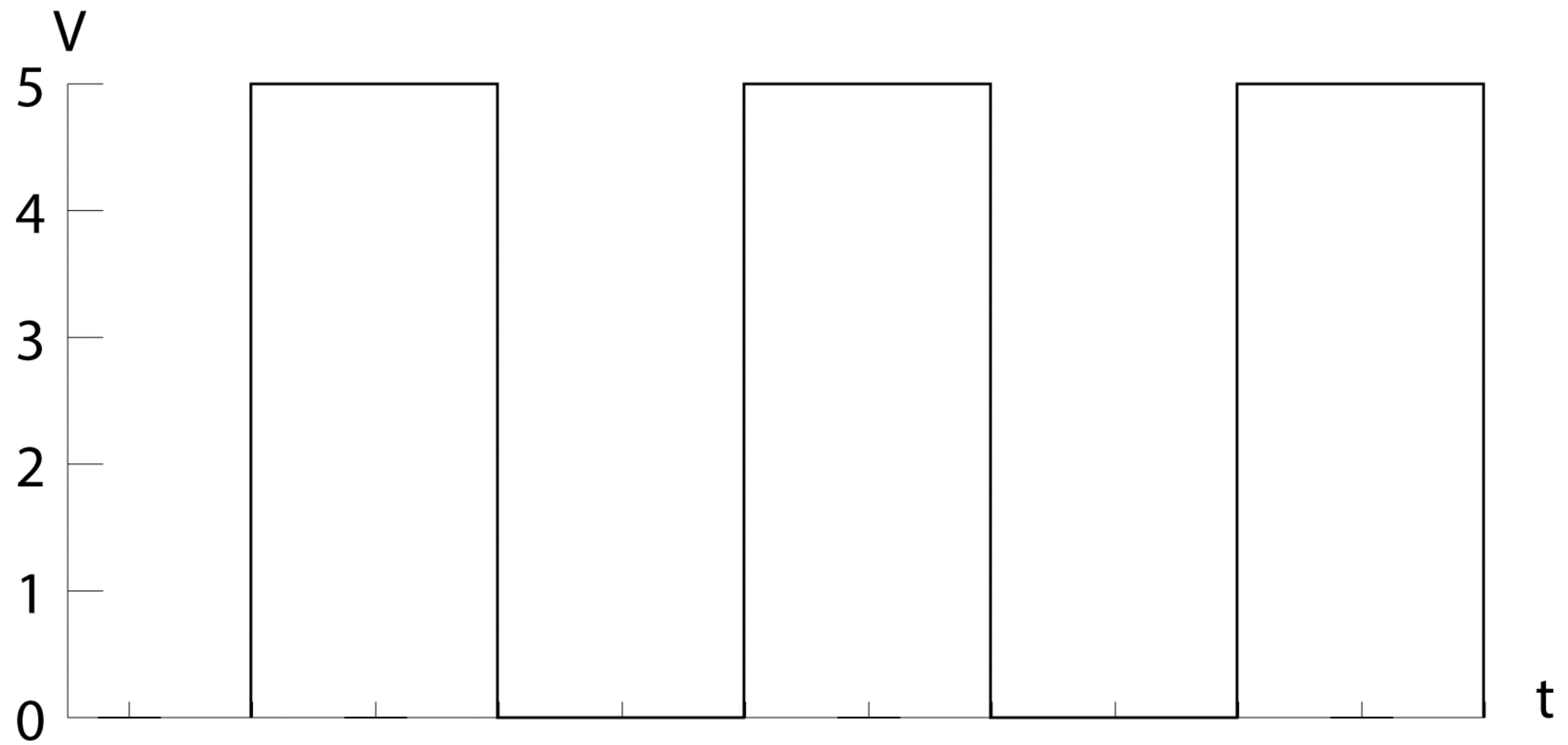
Digital Pins

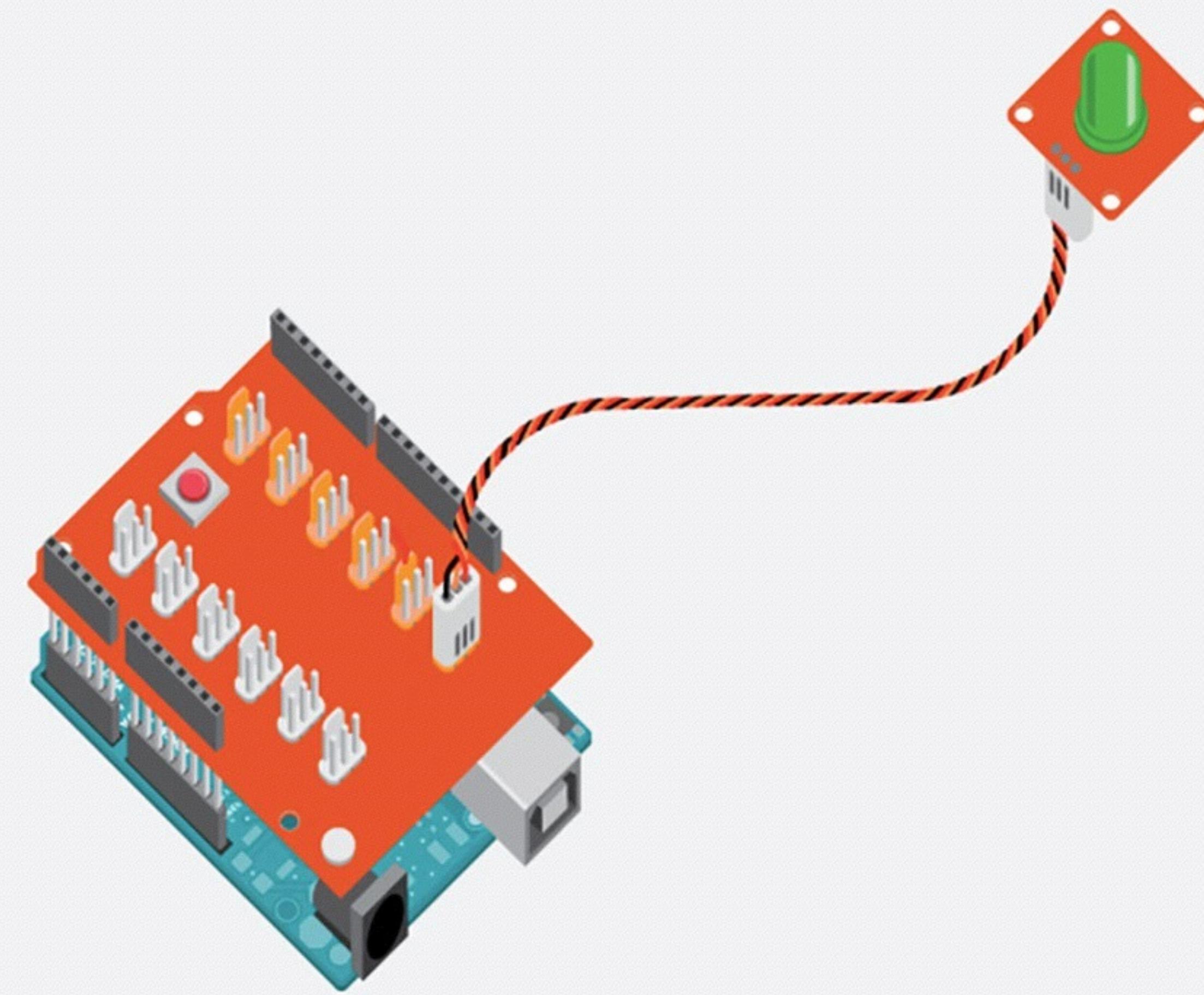


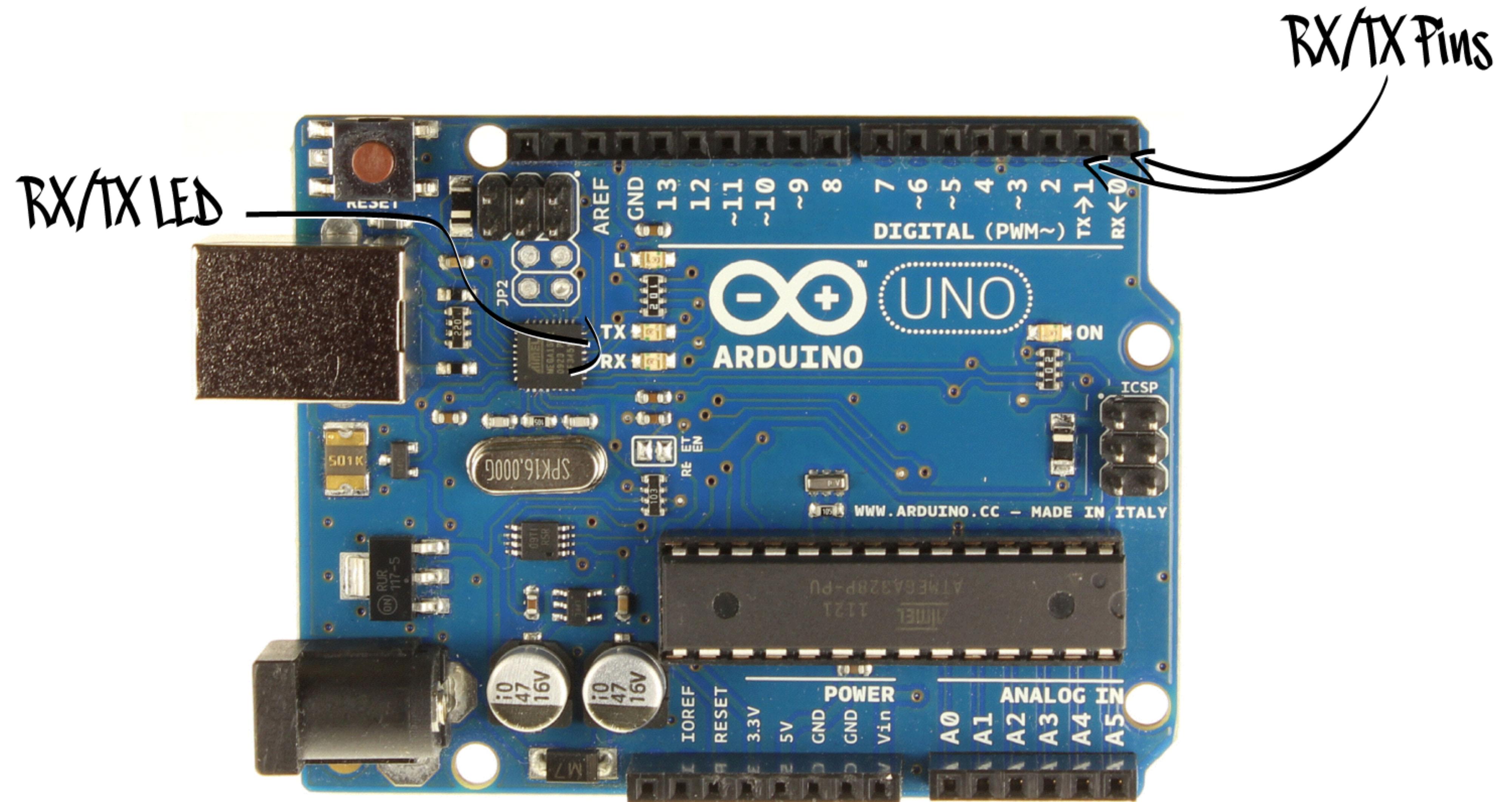
Pin 13 (L) LED
Pin 13 Resistor

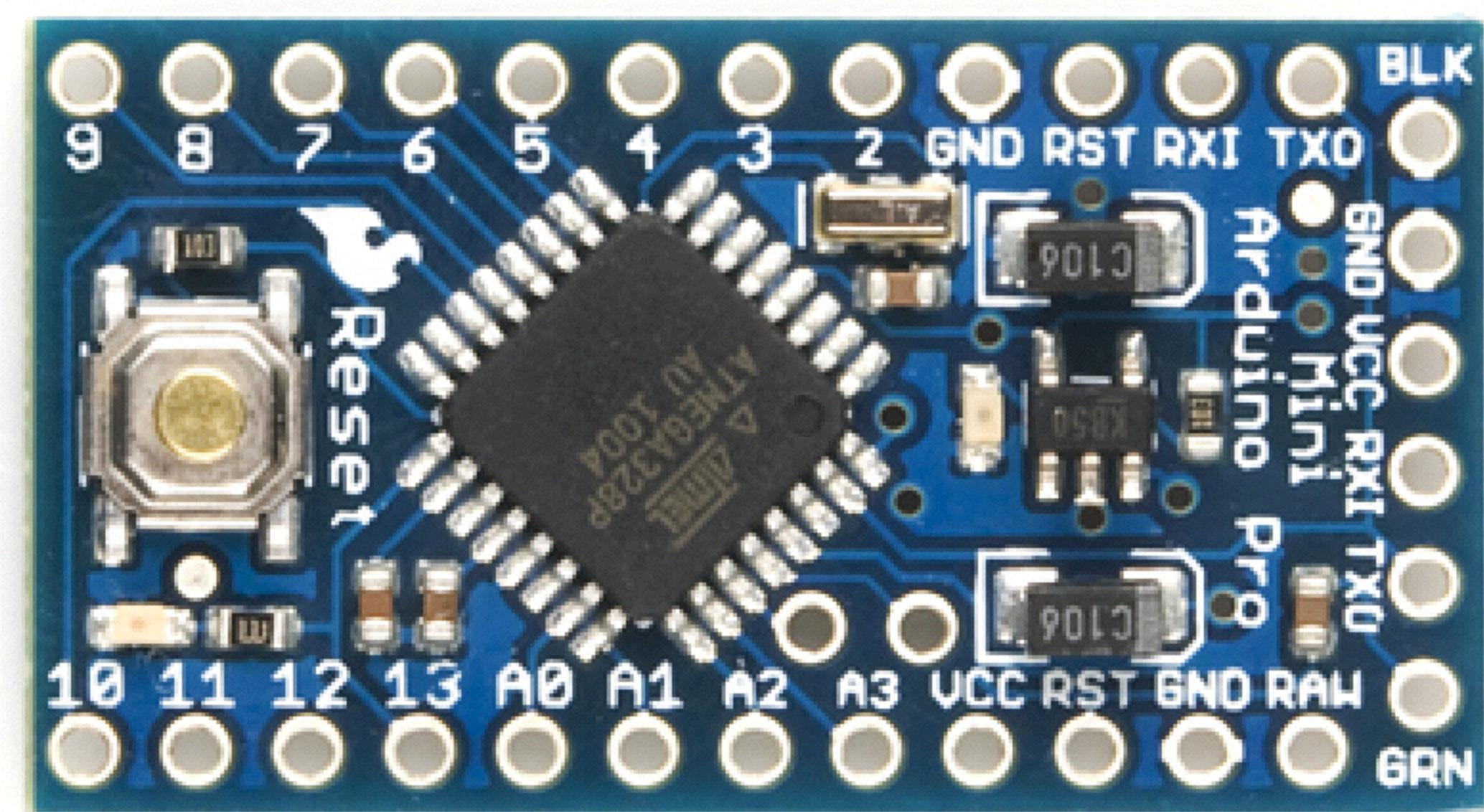


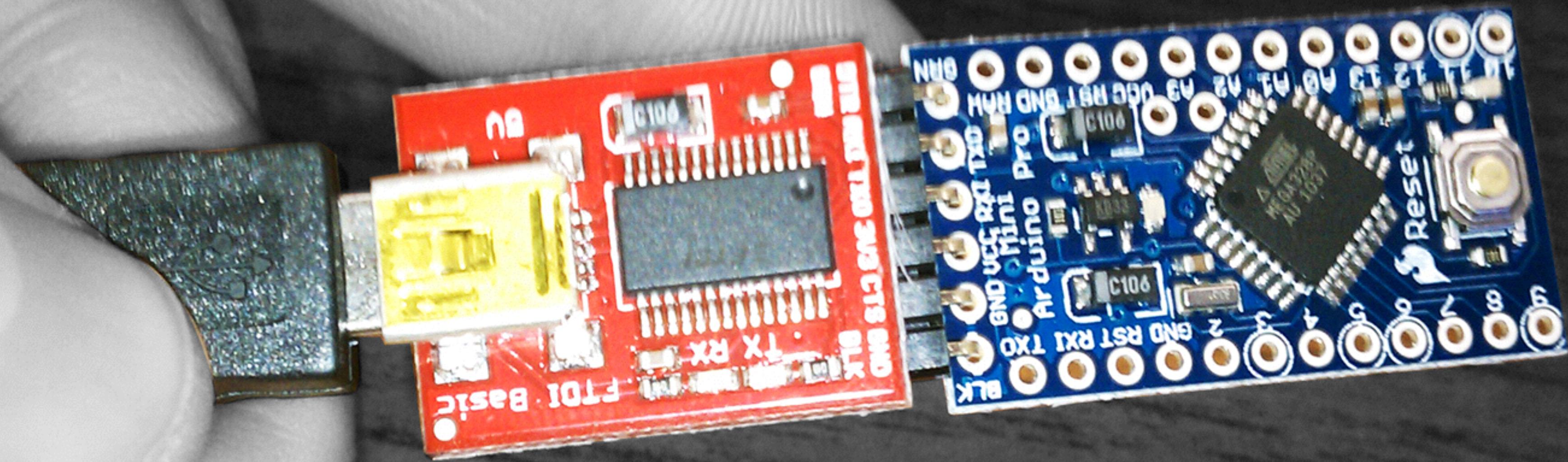


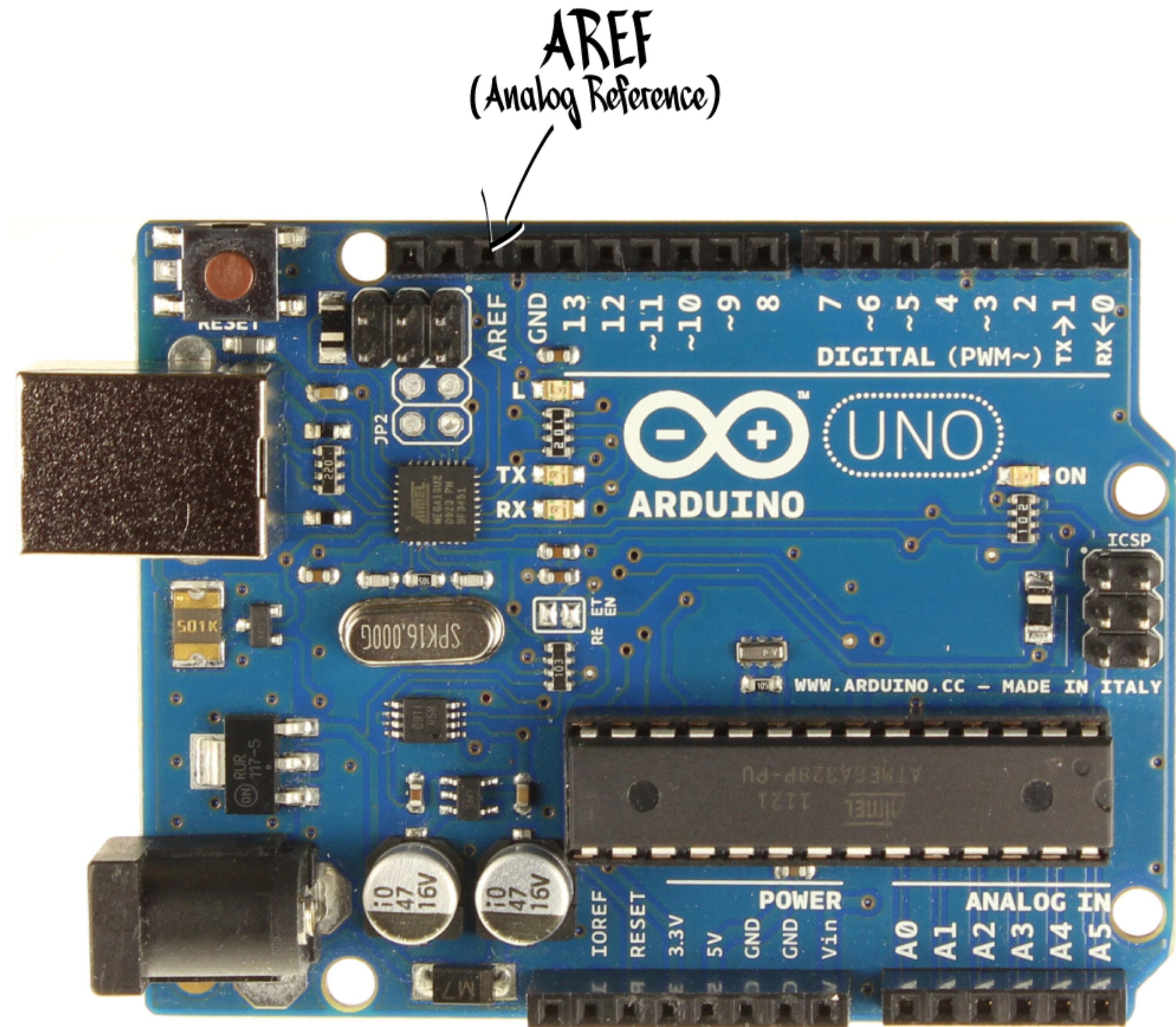


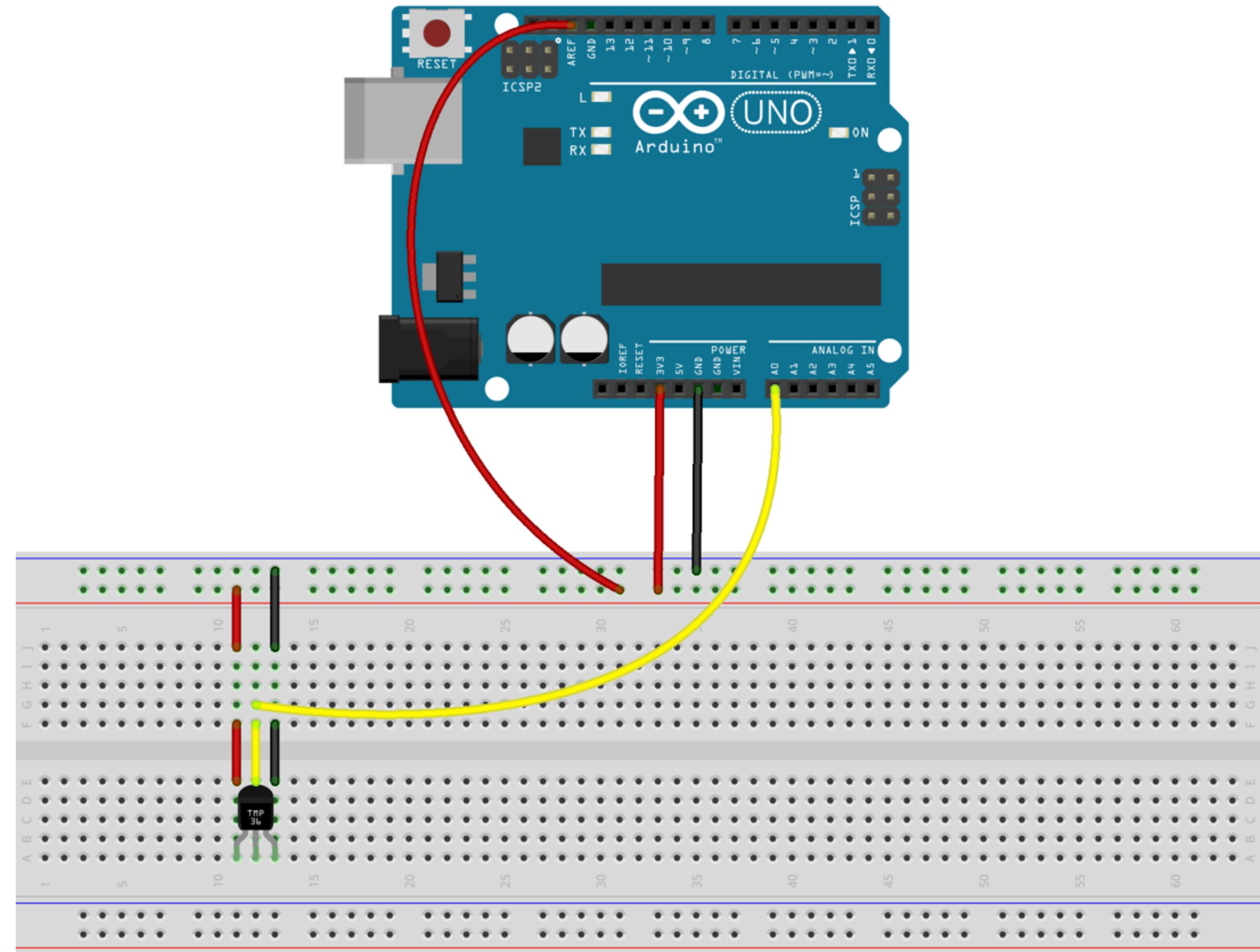






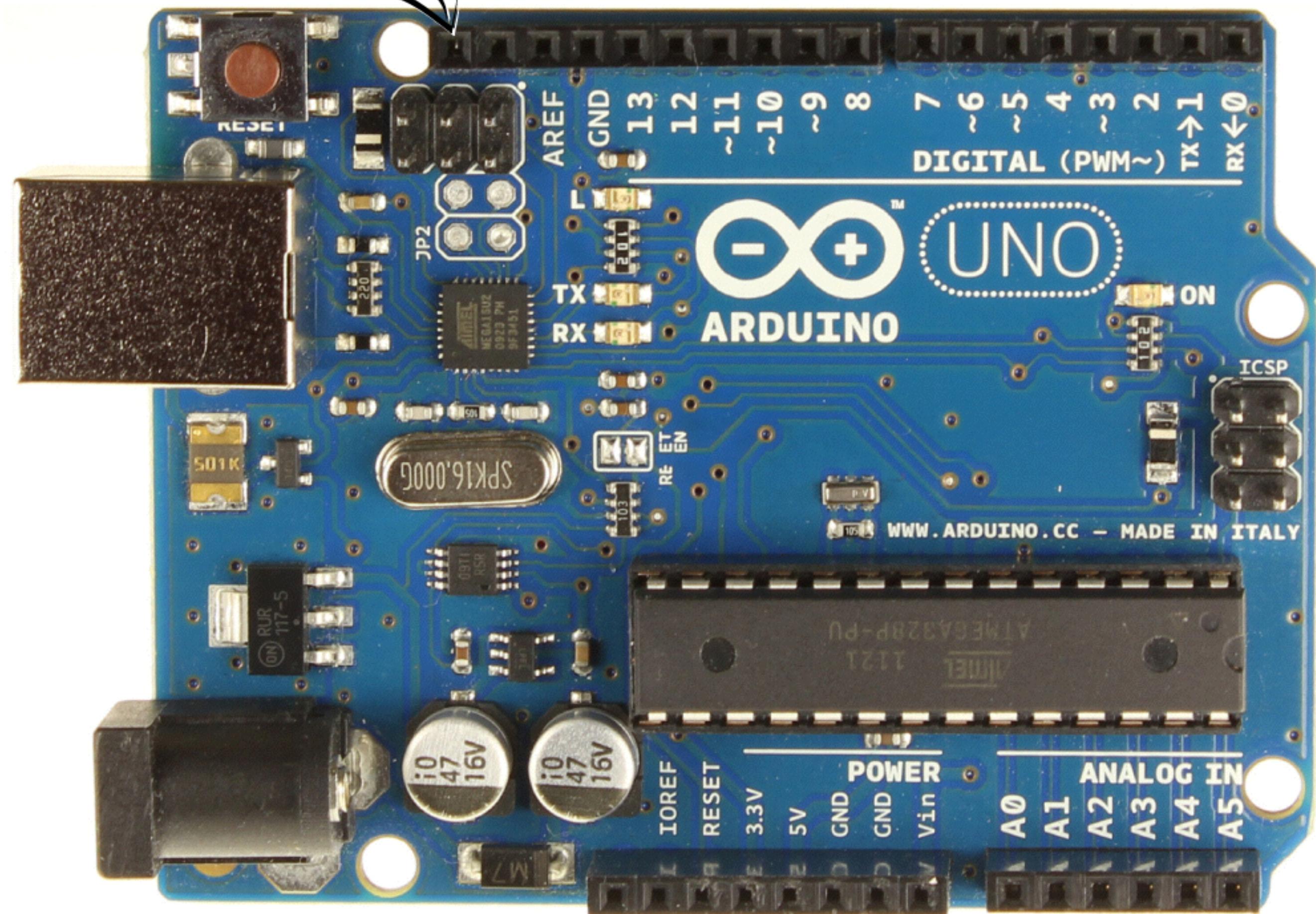




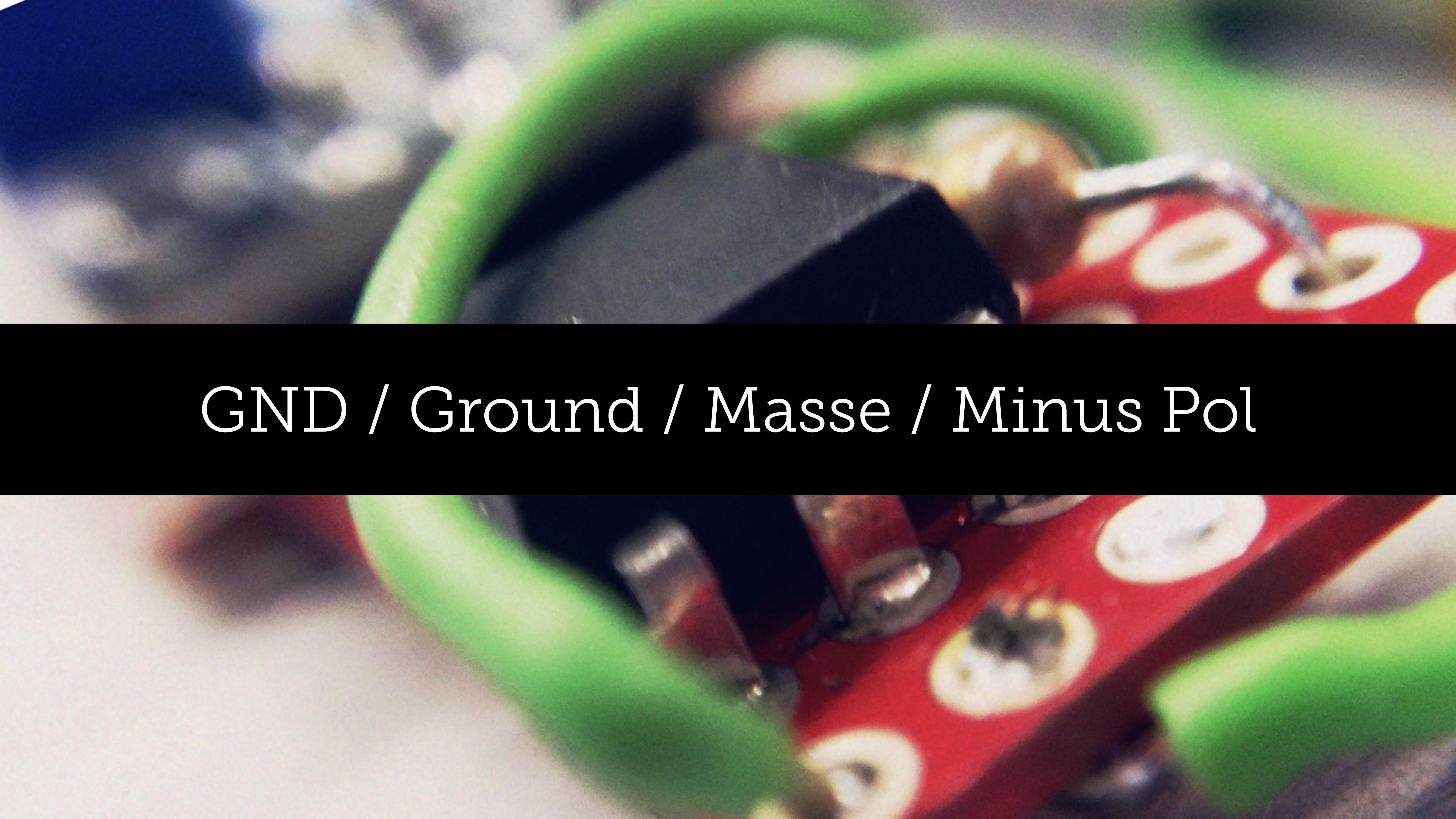


Unnamed Pins

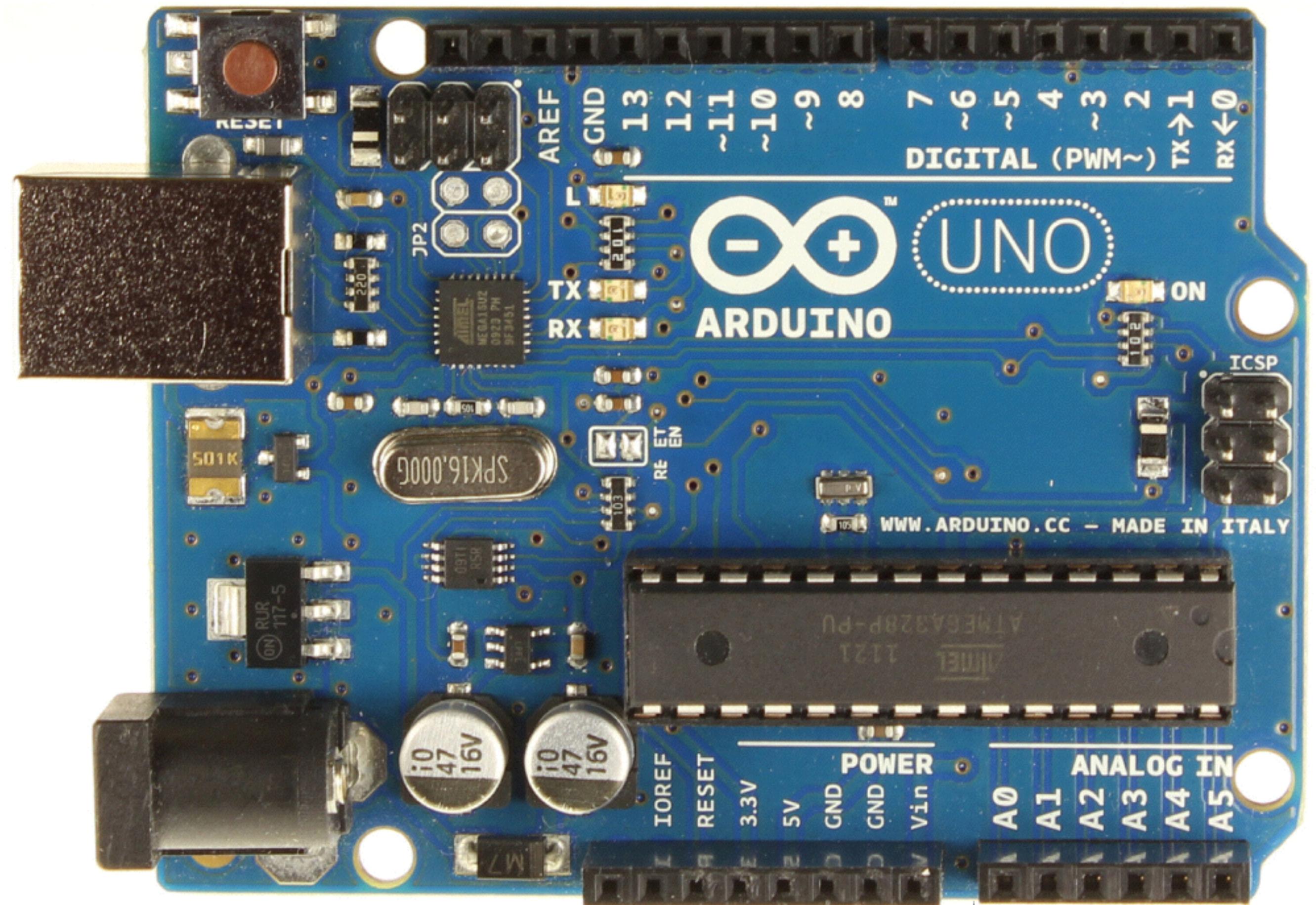
SCL(Serial Clock)



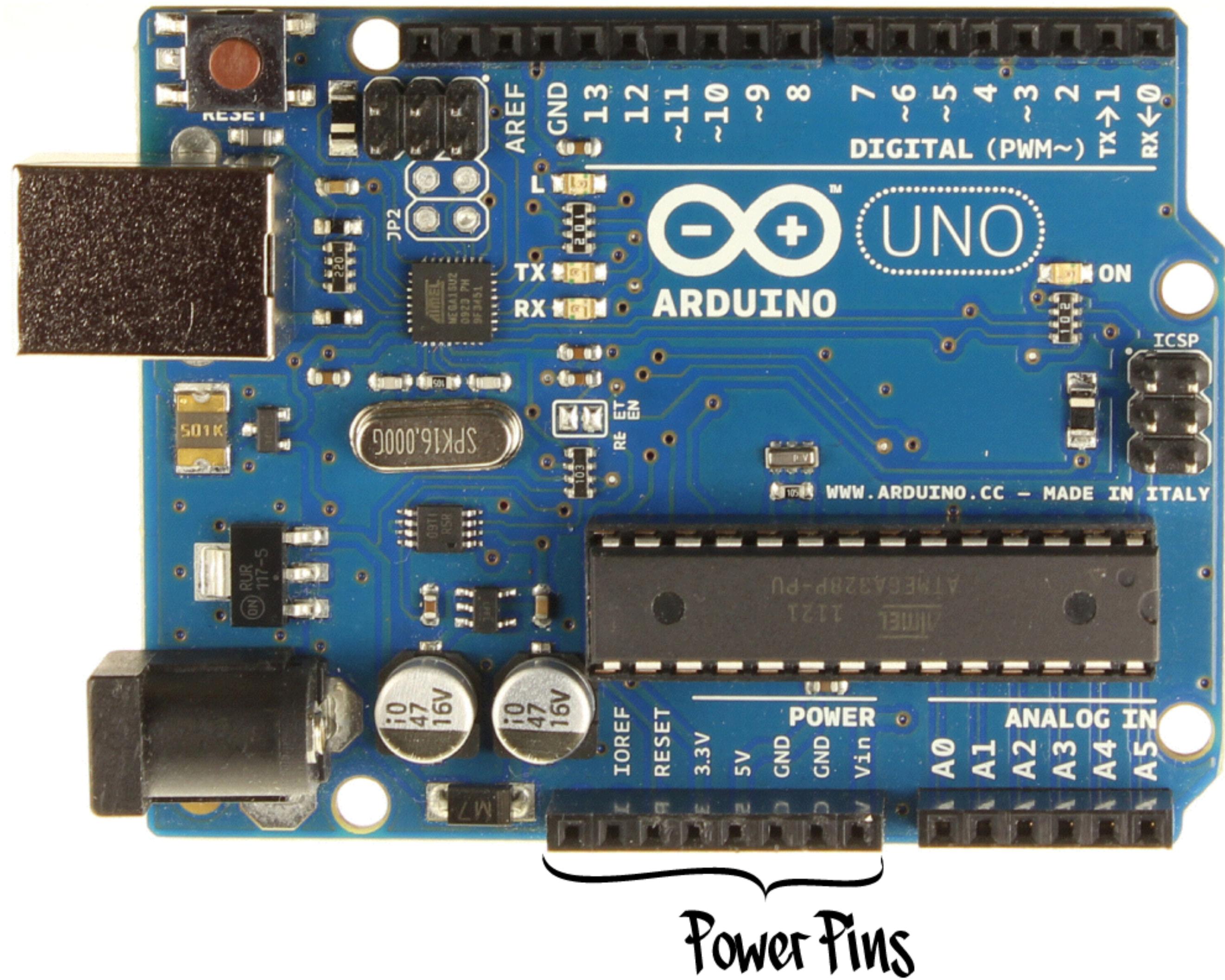


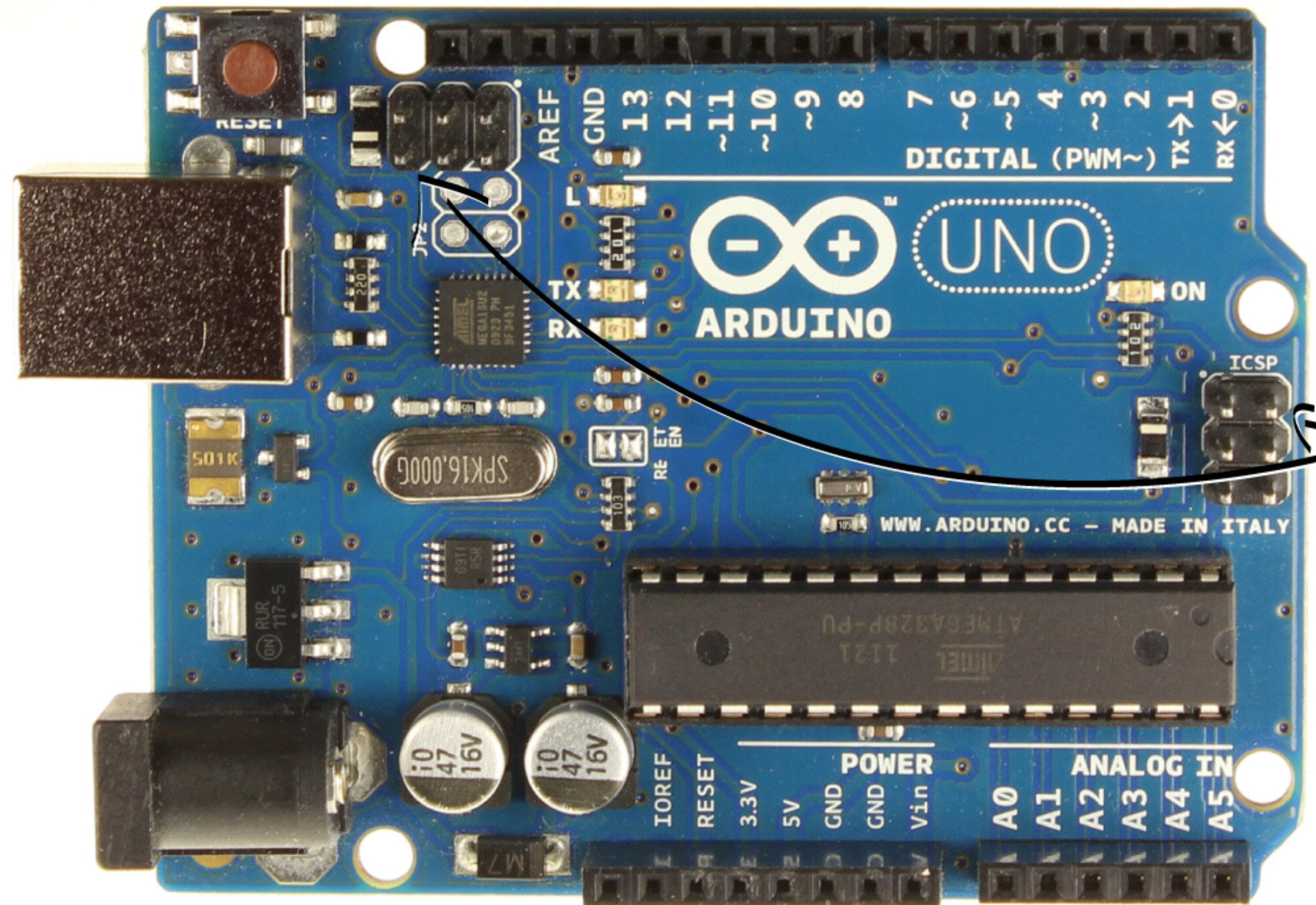


GND / Ground / Masse / Minus Pol

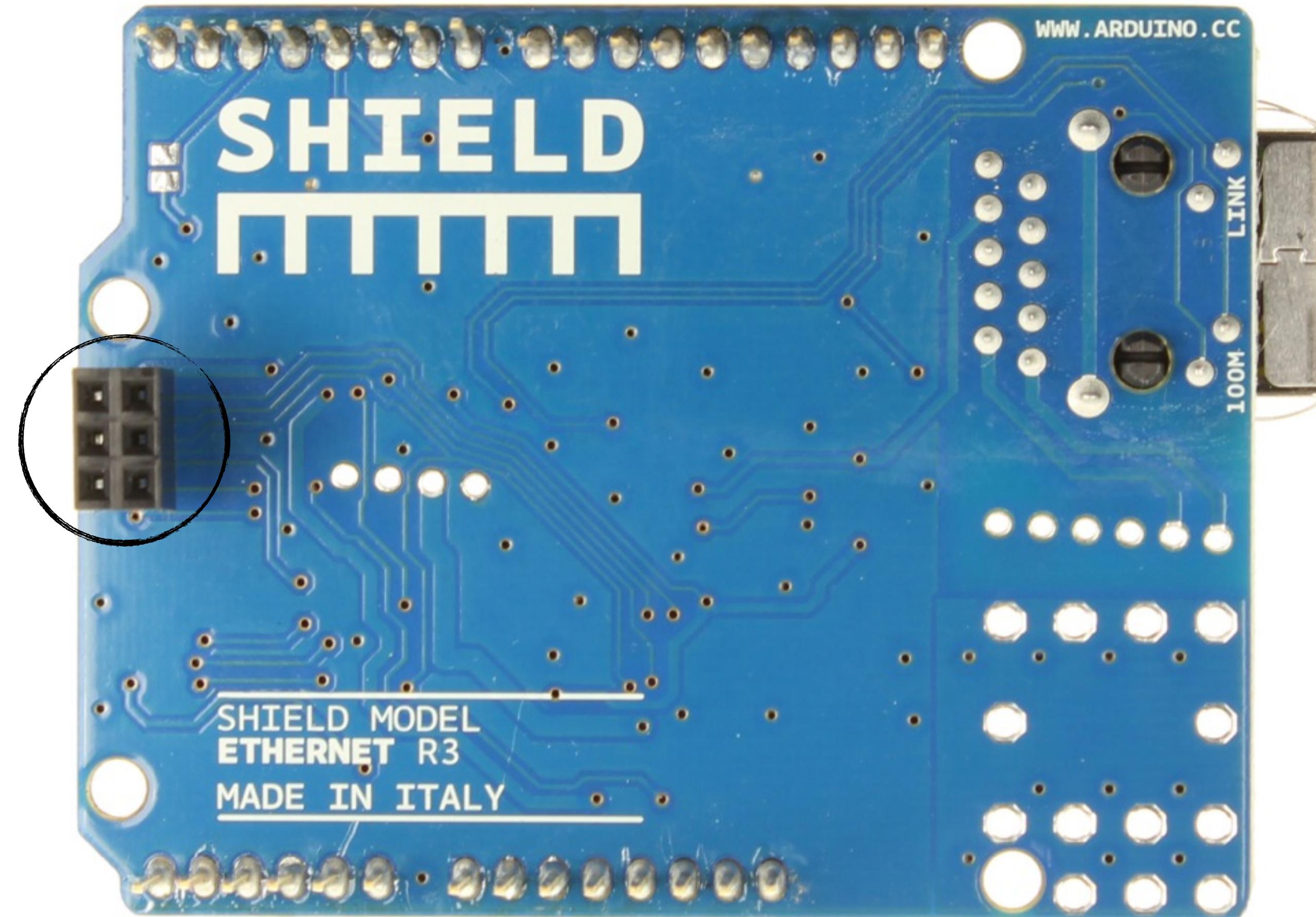


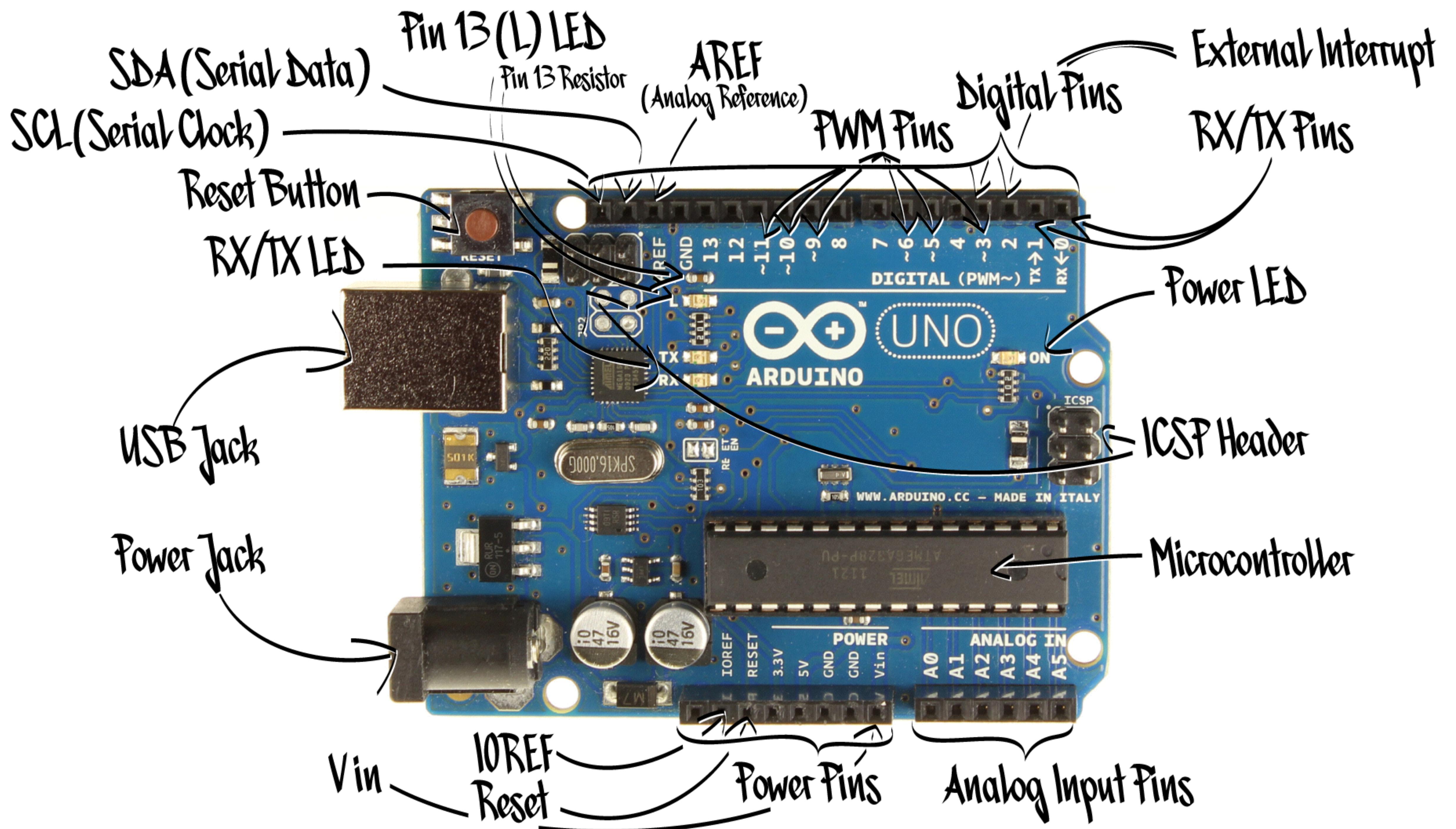
Analog Input Pins





ICSP Header

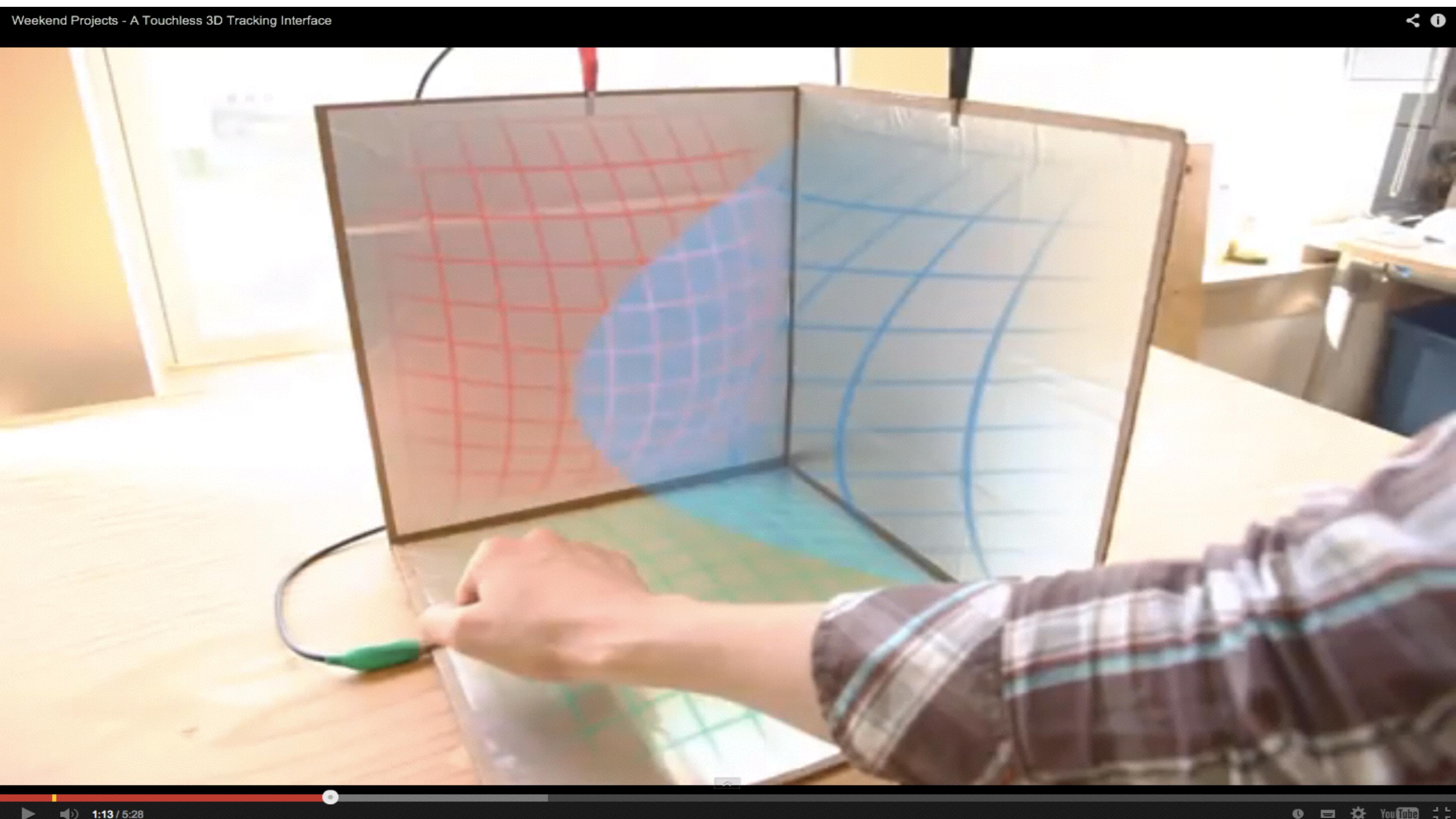




?

Elektrotechnik

Grundlagen

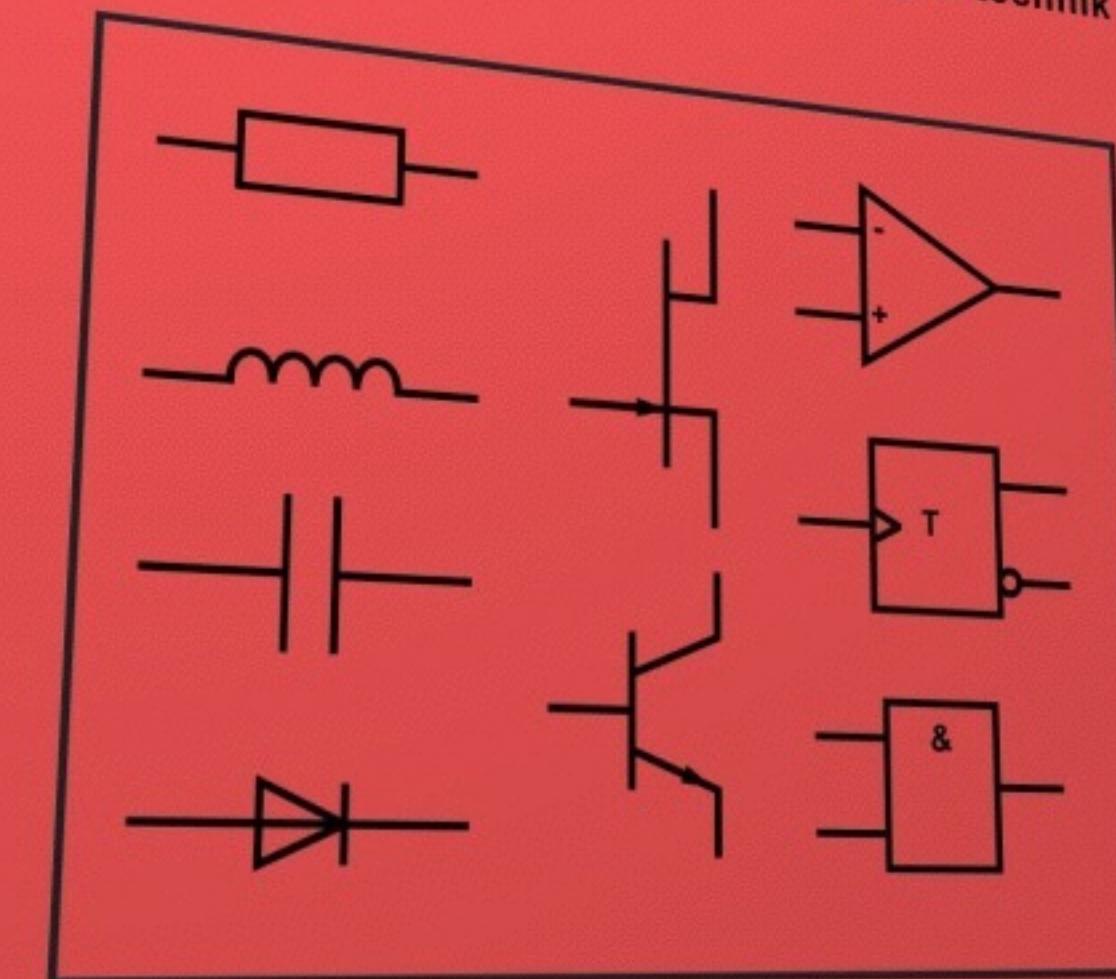


adafruit.com/all-about-leds

Patrick Schnabel

Elektronik-Fibel

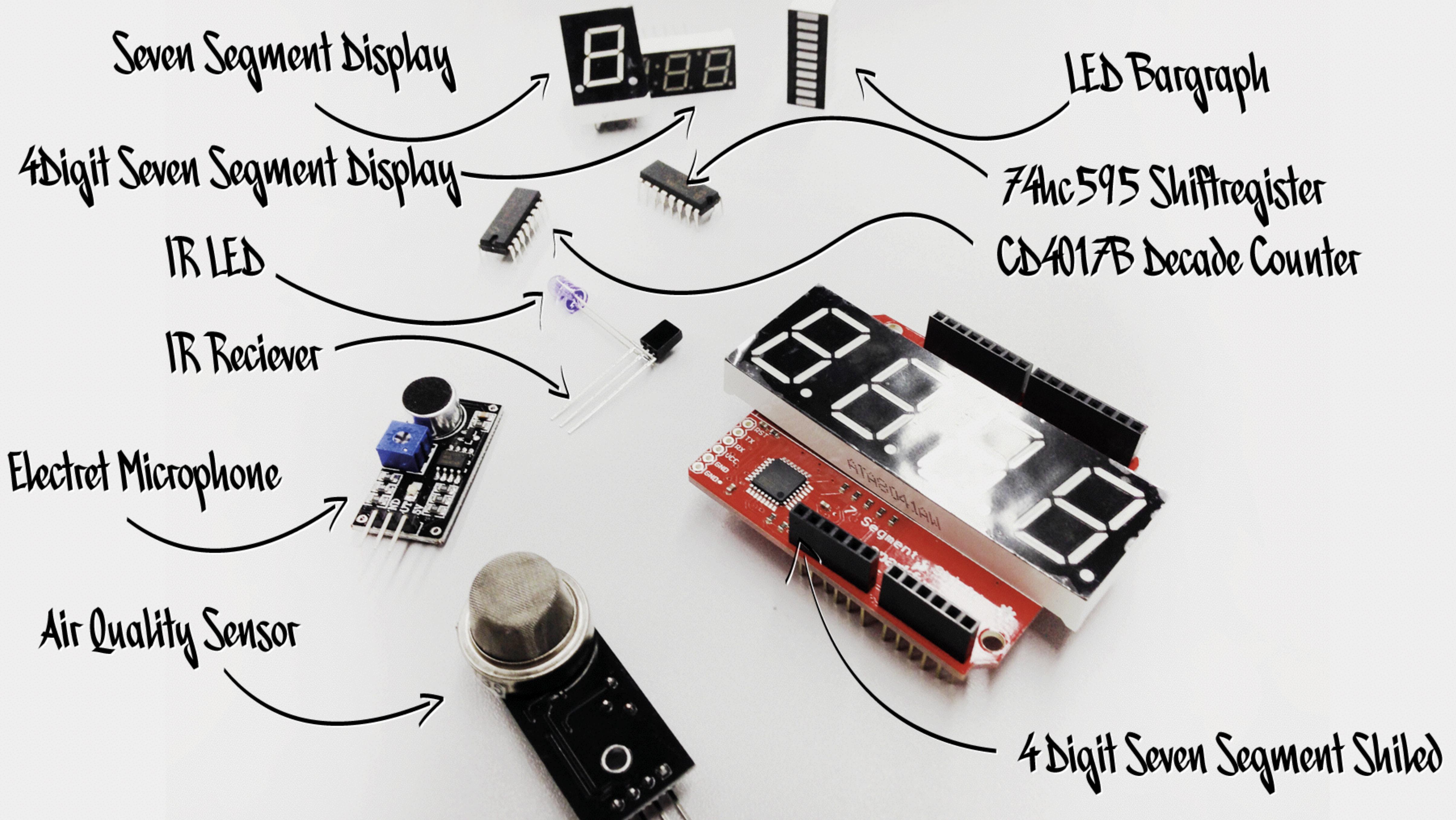
Grundlagen
Bauelemente
Schaltungstechnik
Digitaltechnik



<http://www.elektronik-fibel.de/>

<http://www.elektronik-kompendium.de/>





Löten

**Lassen Sie den Lötkolben
nie unbeaufsichtigt.**

Achten sie darauf,
dass der Lötkolben nicht
das eigene Kabel verletzt.

**Trennen Sie den Lötkolben bei
Verlassen des Arbeitsplatzes
immer vom Netz.**

Halten Sie den Schwamm sauber.

Halten Sie die Lötpitze sauber.

Zweckentfremden Sie den Lötkolben nicht.

Überprüfen Sie den Lötkolben auf Mängel.

Lassen Sie den Lötkolben nie unbeaufsichtigt.

Seien Sie vorsichtig beim Umgang mit dem Lötkolben.

Tragen Sie eine Schutzbrille. (Lötzinn kann spritzen!)

Stecken sie den Lötkolben immer in die Haltevorrichtung.

Achten sie darauf, dass der Lötkolben nicht das eigene Kabel verletzt.

Trennen Sie den Lötkolben bei Verlassen des Arbeitsplatzes immer vom Netz.

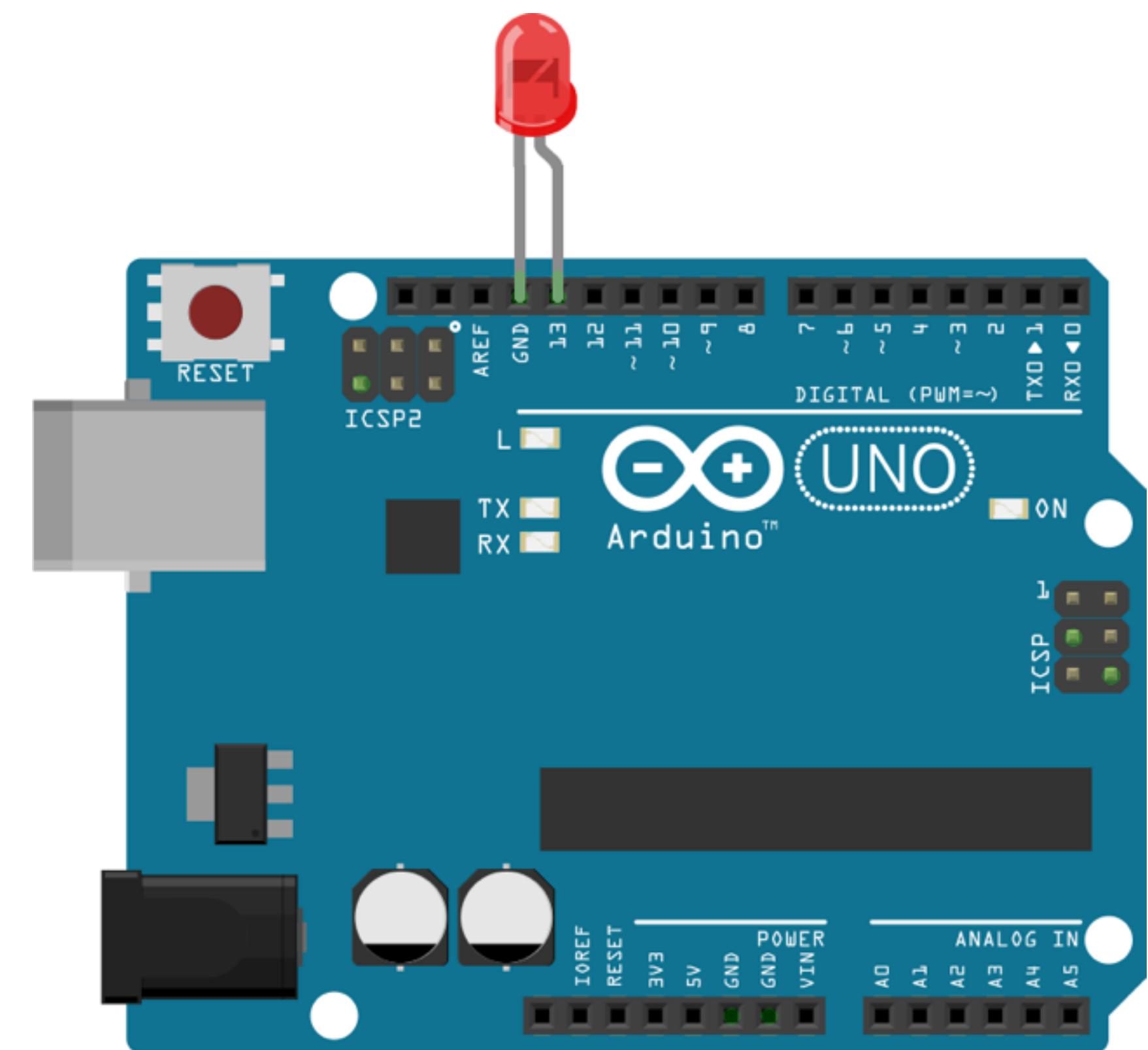
Halten sie den ausgesteckten, aber noch heißen Lötkolben von brennbaren Stoffen fern.

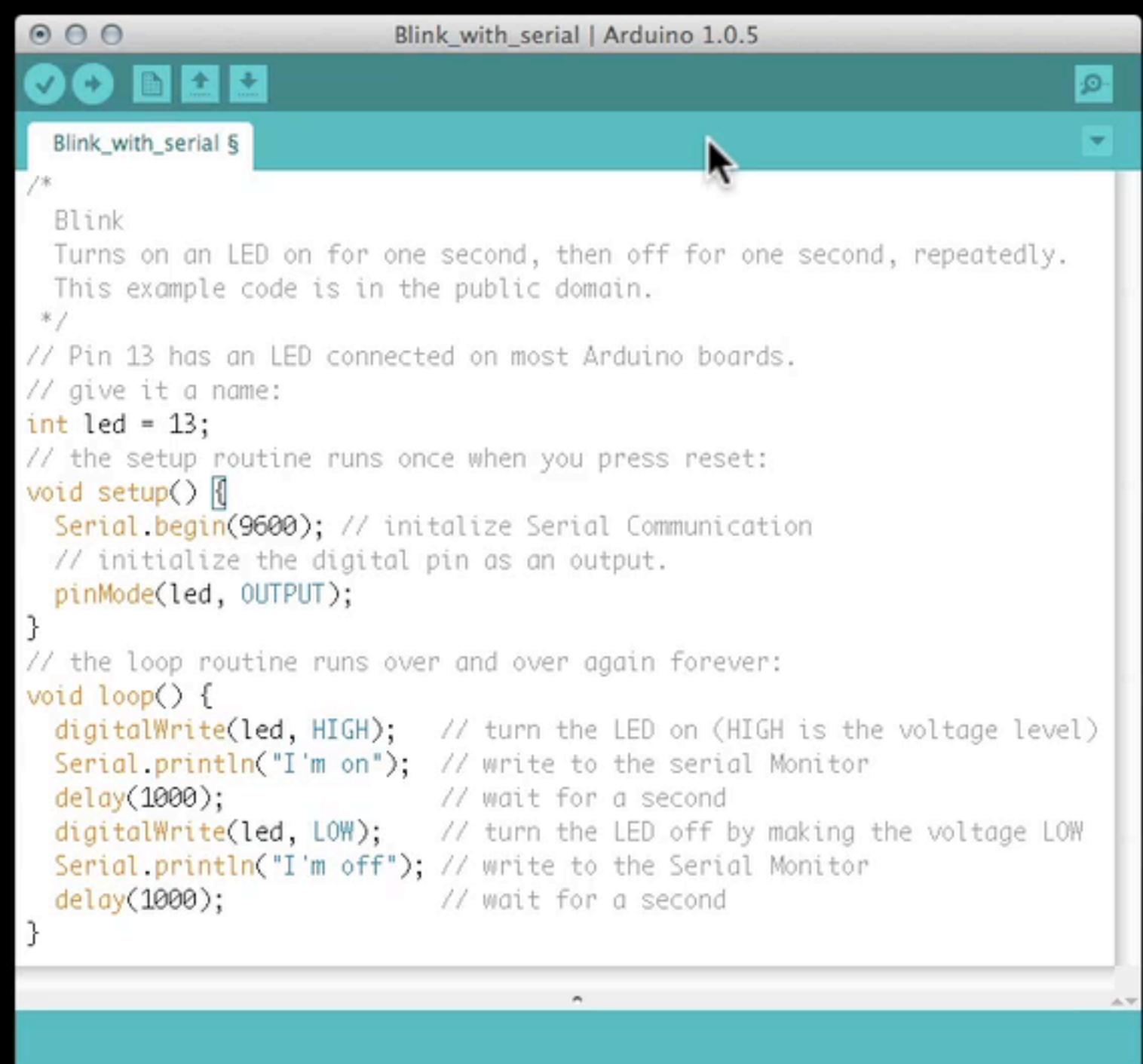
Lötstück erhitzen

hands on löten

hands on digital

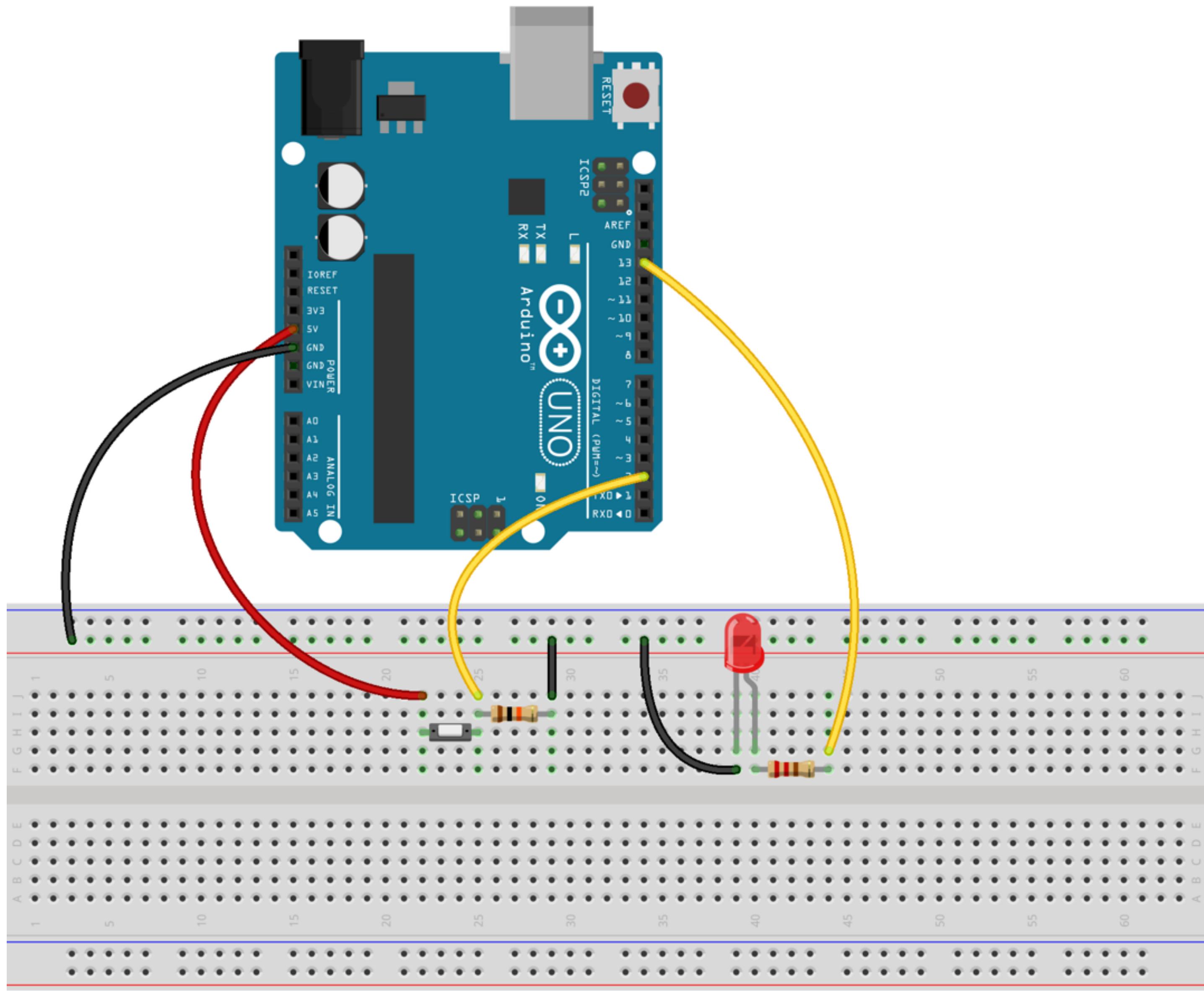
github.com/fabiantheblind/digitalio

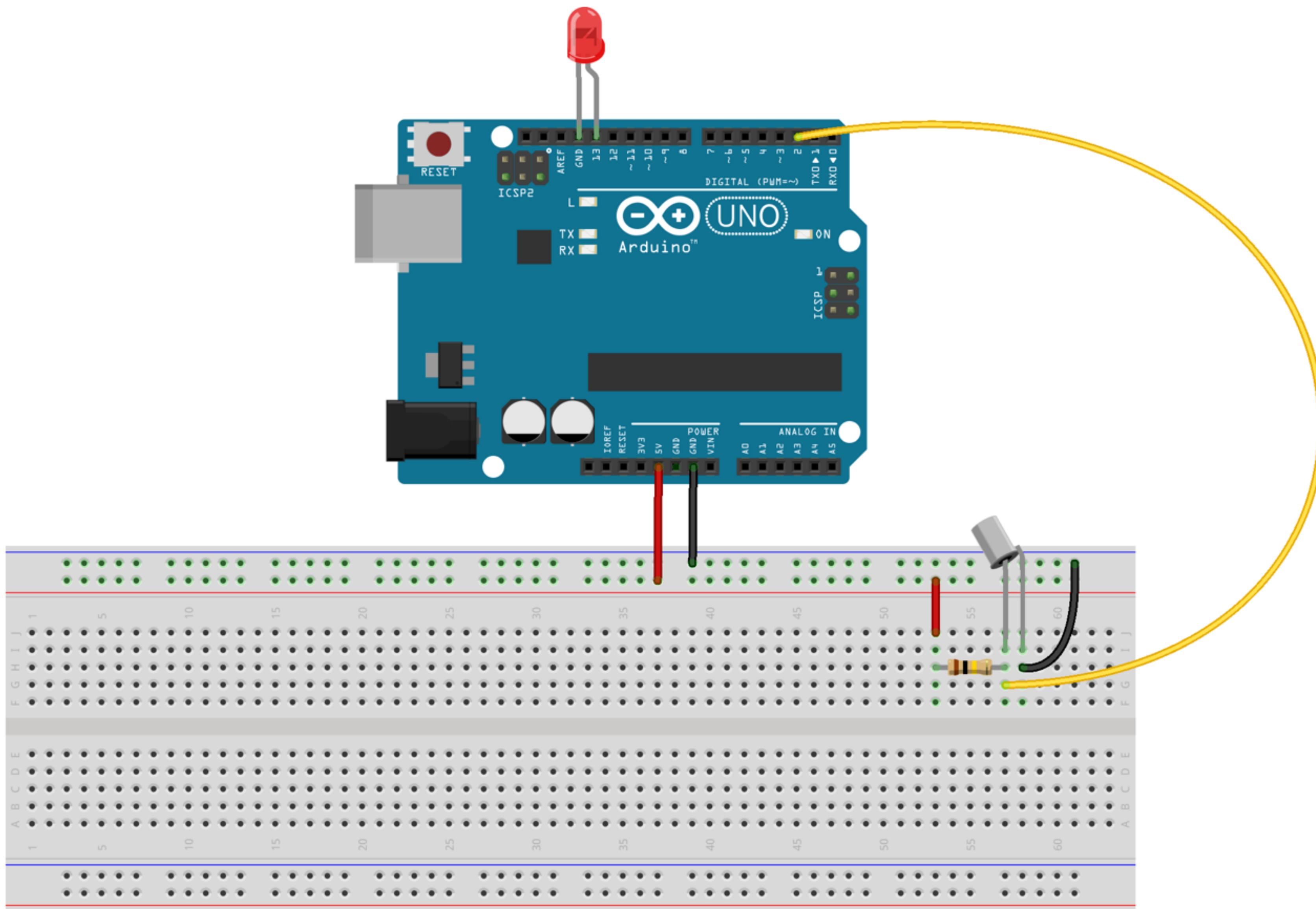


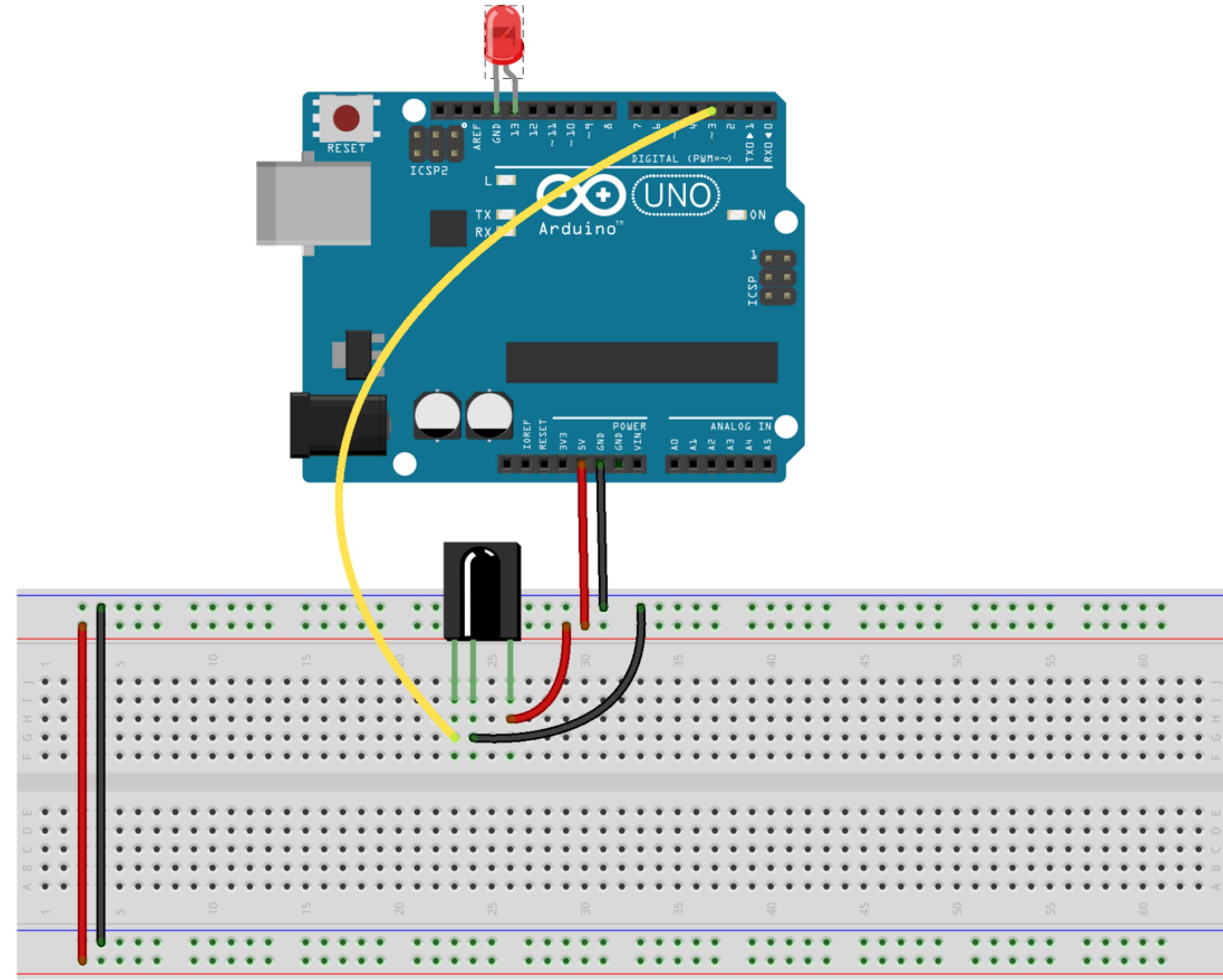


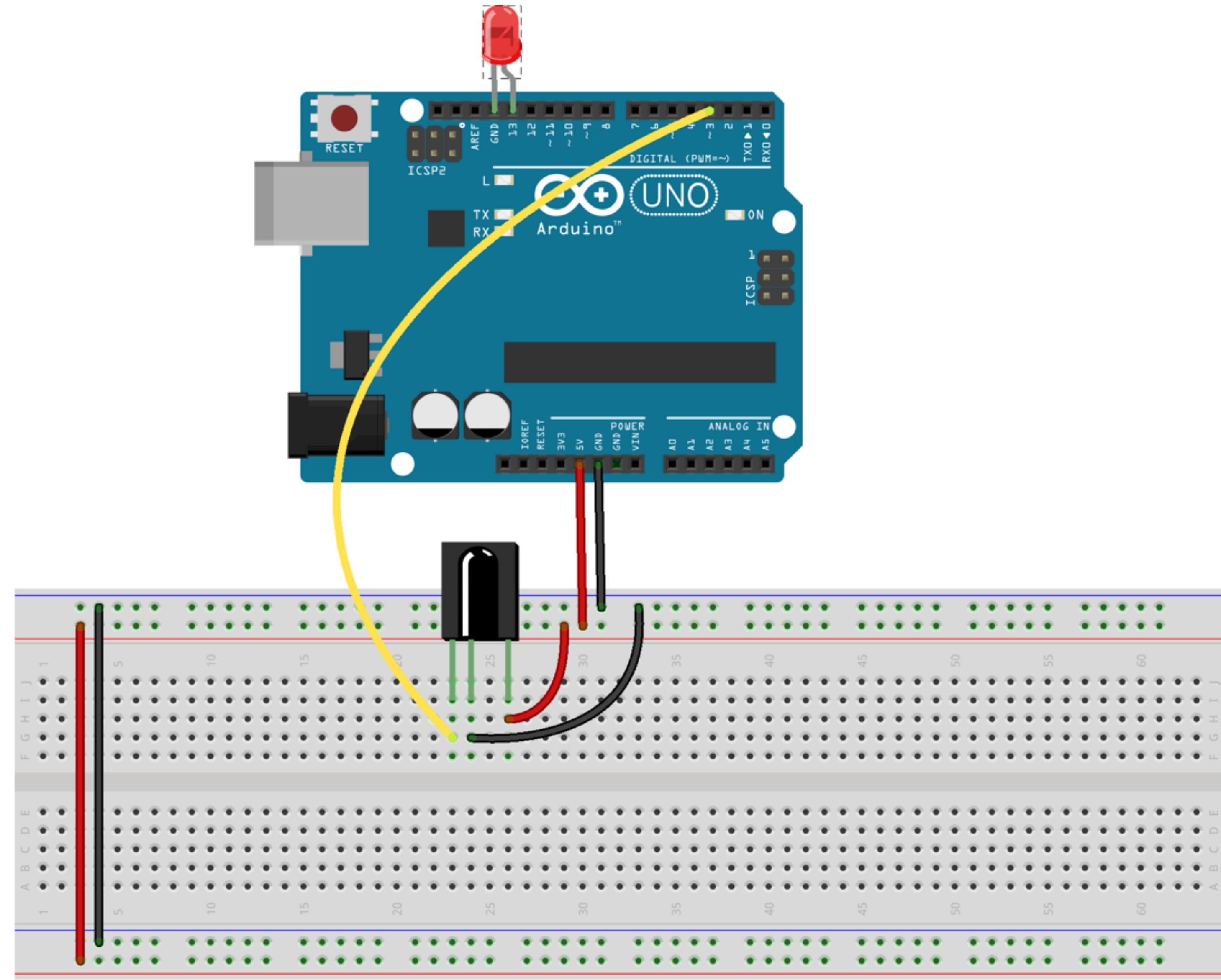
The screenshot shows the Arduino IDE interface with the title bar "Blink_with_serial | Arduino 1.0.5". The main window displays the C++ code for the "Blink_with_serial" sketch. The code is as follows:

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
  This example code is in the public domain.
*/
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;
// the setup routine runs once when you press reset:
void setup() {
  Serial.begin(9600); // initialize Serial Communication
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
  Serial.println("I'm on");
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW
  Serial.println("I'm off");
  delay(1000);               // wait for a second
}
```









```

/*
Adafruit Arduino - Lesson 4. 8 LEDs and a Shift Register
http://learn.adafruit.com/adafruit-arduino-lesson-4-eight-leds/arduino-code

Shiftout tutorial

http://arduino.cc/en/tutorial/ShiftOut

PINS 1-7, 15      Q0 " Q7      Output Pins
PIN 8             GND        Ground, Vss
PIN 9             Q7"       Serial Out
PIN 10            MR         Master Reclear, active low
PIN 11            SH_CP     Shift register clock pin
PIN 12            ST_CP     Storage register clock pin (latch pin)
PIN 13            OE         Output enable, active low
PIN 14            DS         Serial data input
PIN 16            Vcc        Positive supply voltage

*/
//Pin connected to ST_CP of 74HC595
int latchPin = 8;
//Pin connected to SH_CP of 74HC595
int clockPin = 12;
///Pin connected to DS of 74HC595
int dataPin = 11;

byte leds = 0;

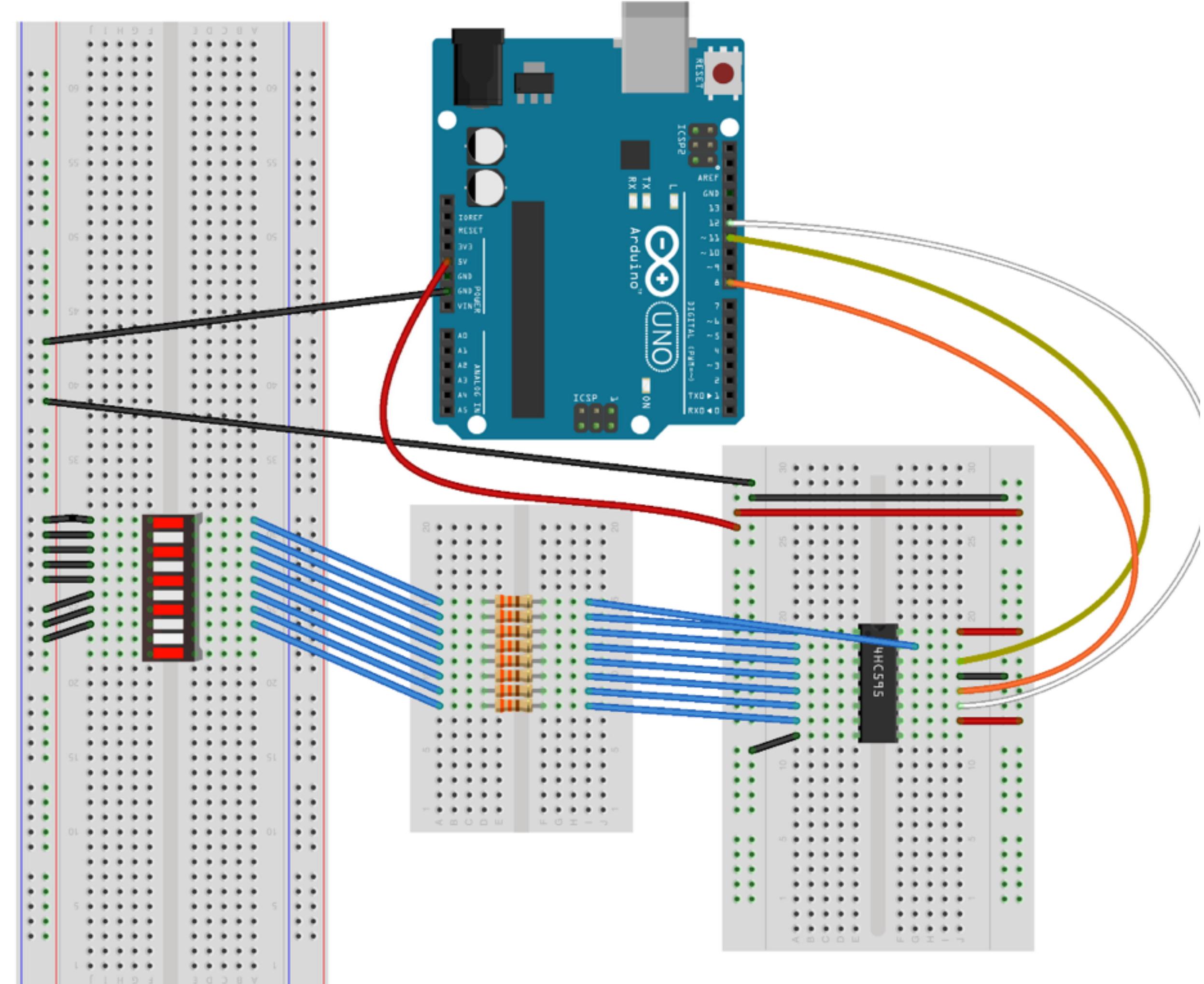
void setup() {
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    setGraph(B10000111);
    // setBarGraph0();
    // setBarGraphOf();
}

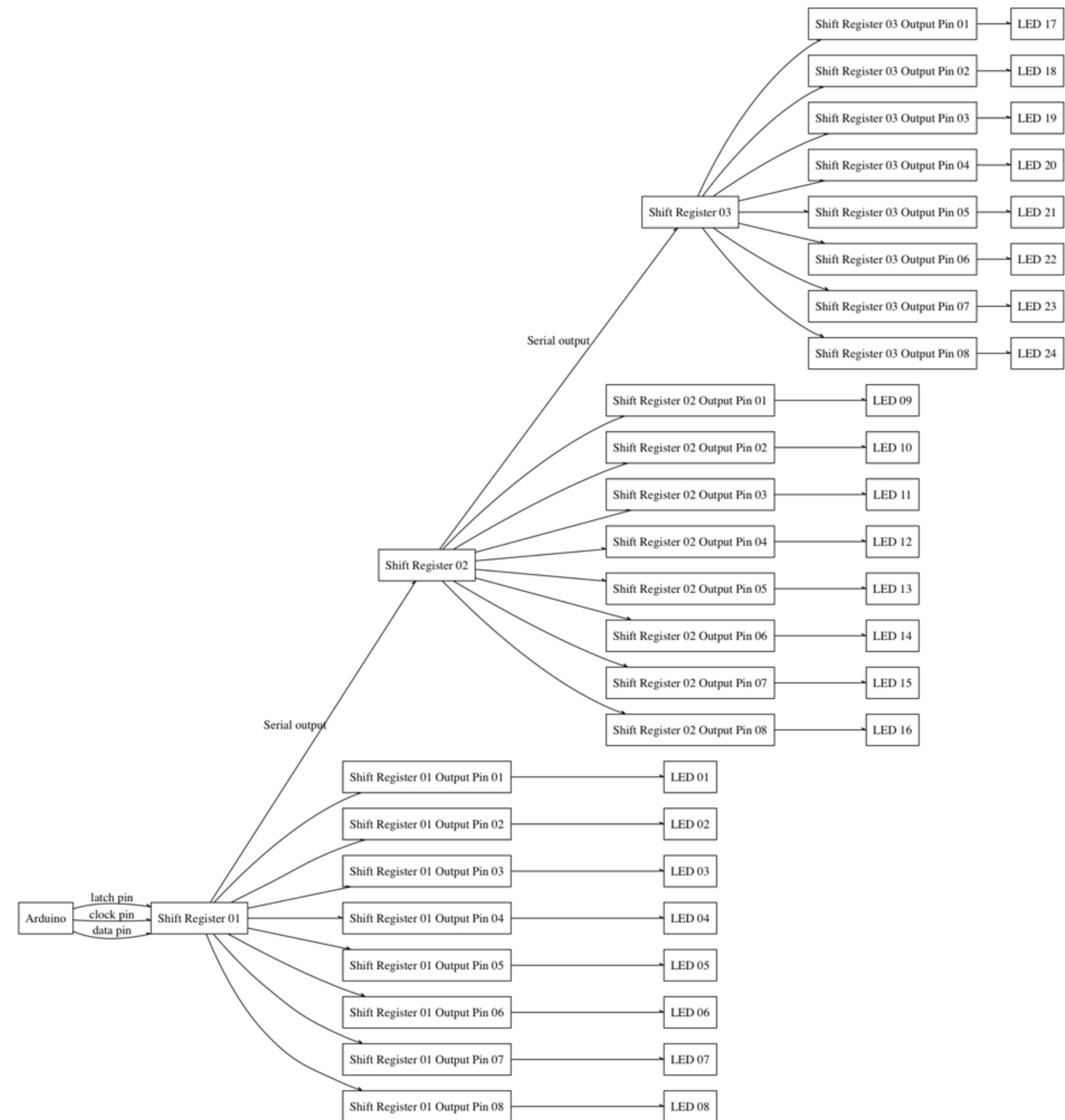
void loop(){
    //loopGraph();
}
void loopGraph(){
    leds = 0;
    updateShiftRegister();
    delay(500);
    for (int i = 0; i < 8; i++){
        bitSet(leds, i);
        updateShiftRegister();
        delay(500);
    }
}
void setGraph(byte val){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, val);
    digitalWrite(latchPin, HIGH);
}

void setBarGraphOf(){
    leds = 0;
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, leds);
    digitalWrite(latchPin, HIGH);
}

void setBarGraph(int i){
    //   leds = 0;
    bitSet(leds, i);
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, leds);
    digitalWrite(latchPin, HIGH);
}

```





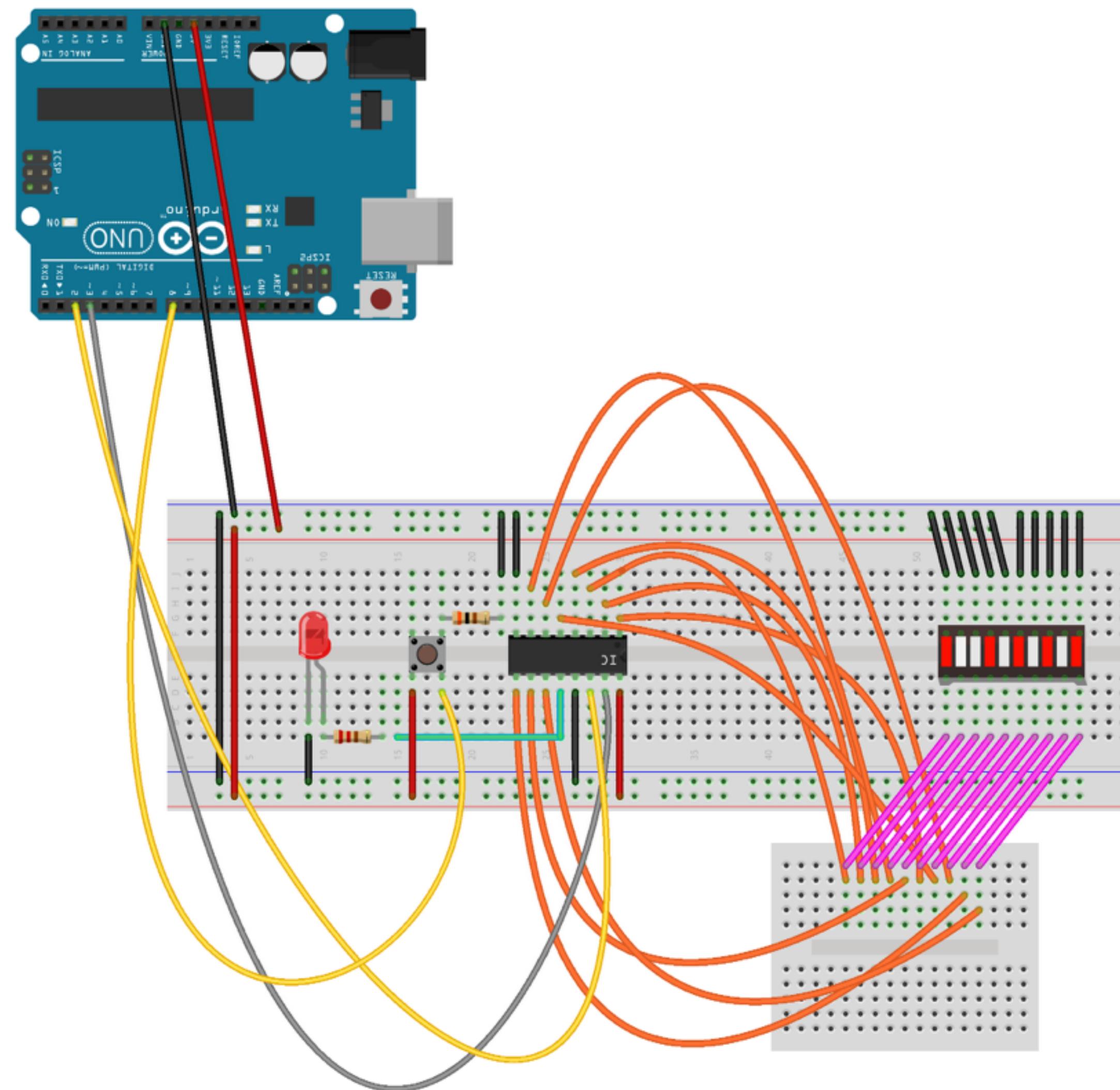
```

/*
 * Simple LED Bar graph control using a 4017 counter and a pushbutton
 * The 4017 is used to save pins on the arduino.
 * written by Fabian Moro 'n Zirfas
 * based on work by Leonel Machava
 * http://codentronix.com
 * http://codentronix.com/2011/06/05/arduino-led-bar-graph-driven-by-a-4017-counter/
 * This code is release under the "MIT License" available at
 * http://www.opensource.org/licenses/mit-license.php
 *
 * and the Arduino Button Example
 * http://arduino.cc/en/Tutorial/Button
 * created 2005
 * by DojoDave <http://www.0j0.org>
 * modified 30 Aug 2011
 * by Tom Igoe
 *
 * CD4017B Datasheet
 * http://www.ti.com/lit/ds/symlink/cd4017b.pdf
 *
 * Pin Functions CD4017B
 *
 * PIN 16 - V DC
 * PIN 15 - RESET
 * PIN 14 - CLOCK
 * PIN 13 - CLOCK INHIBIT (STOP COUNTING) not used need to be GND
 * PIN 12 - CARRY OUT (INDICATE DECIMAL) not used need to be GND
 * PIN 11 - 9
 * PIN 10 - 4
 * PIN 09 - 8
 * PIN 08 - GND
 * PIN 07 - 3
 * PIN 06 - 7
 * PIN 05 - 6
 * PIN 04 - 2
 * PIN 03 - 0
 * PIN 02 - 1
 * PIN 01 - 5
 */
int clockPin = 2; // this pulses the clock
int buttonPin = 8; // this is for the pushbutton
int buttonState = 0;// variable for reading the pushbutton status

void setup() {
  pinMode(clockPin,OUTPUT);
  pinMode(buttonPin, INPUT);
}

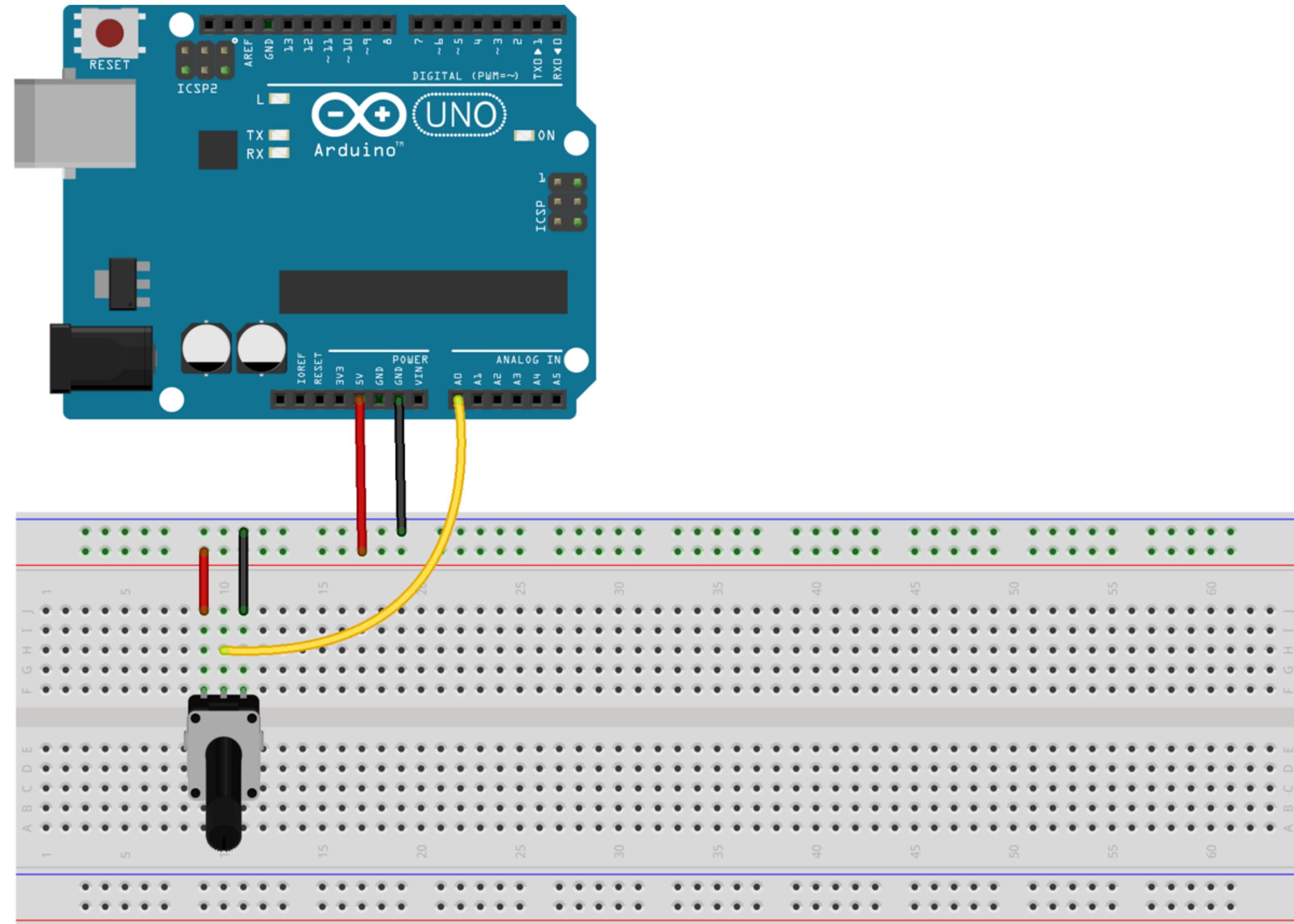
void loop() {
  /**
   * Read the state of the button
   */
  buttonState = digitalRead(buttonPin);
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if(buttonState == HIGH) {
    digitalWrite(clockPin,LOW);
  }
  else {
    // button is pressed
    digitalWrite(clockPin,HIGH);
  }
}

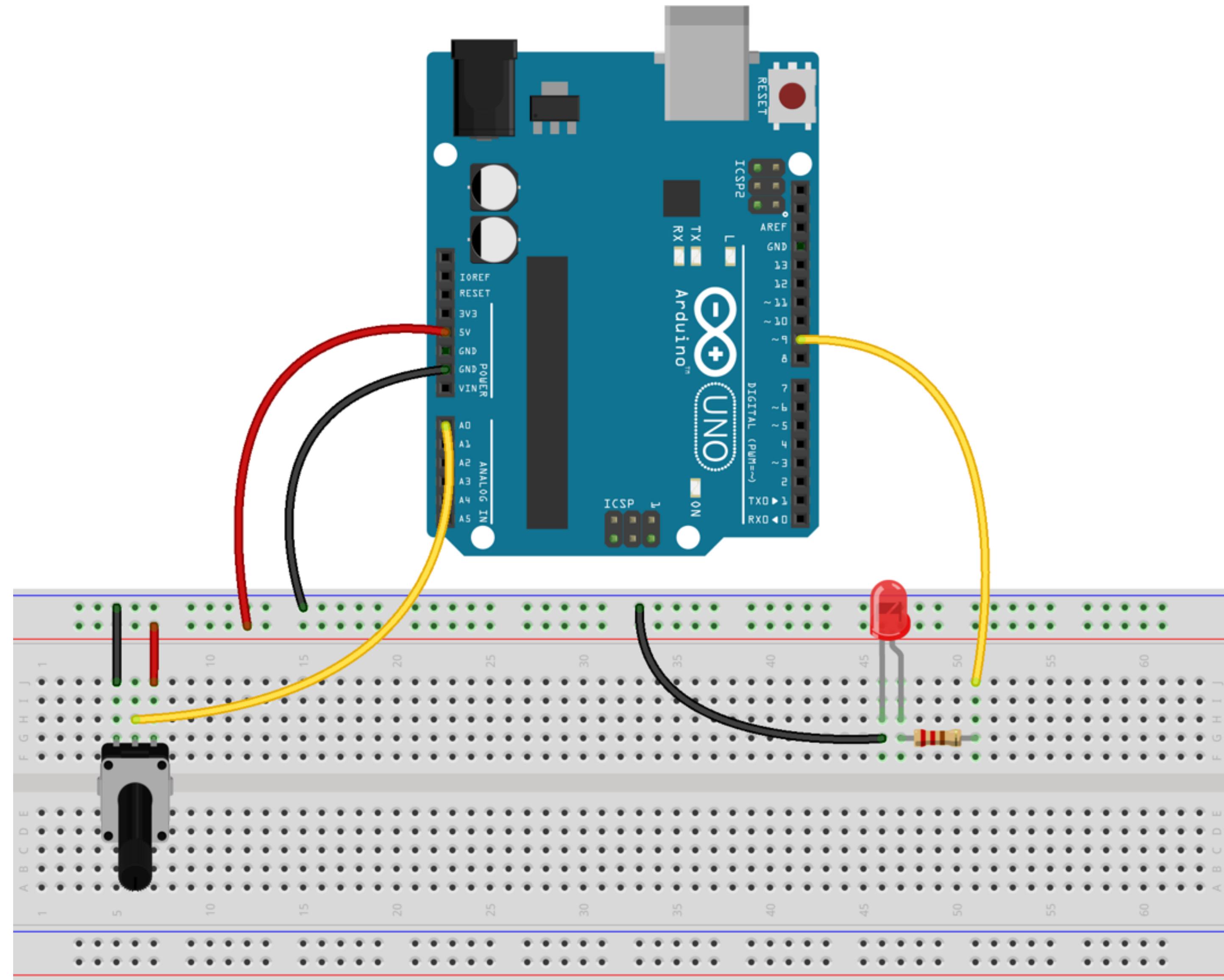
```

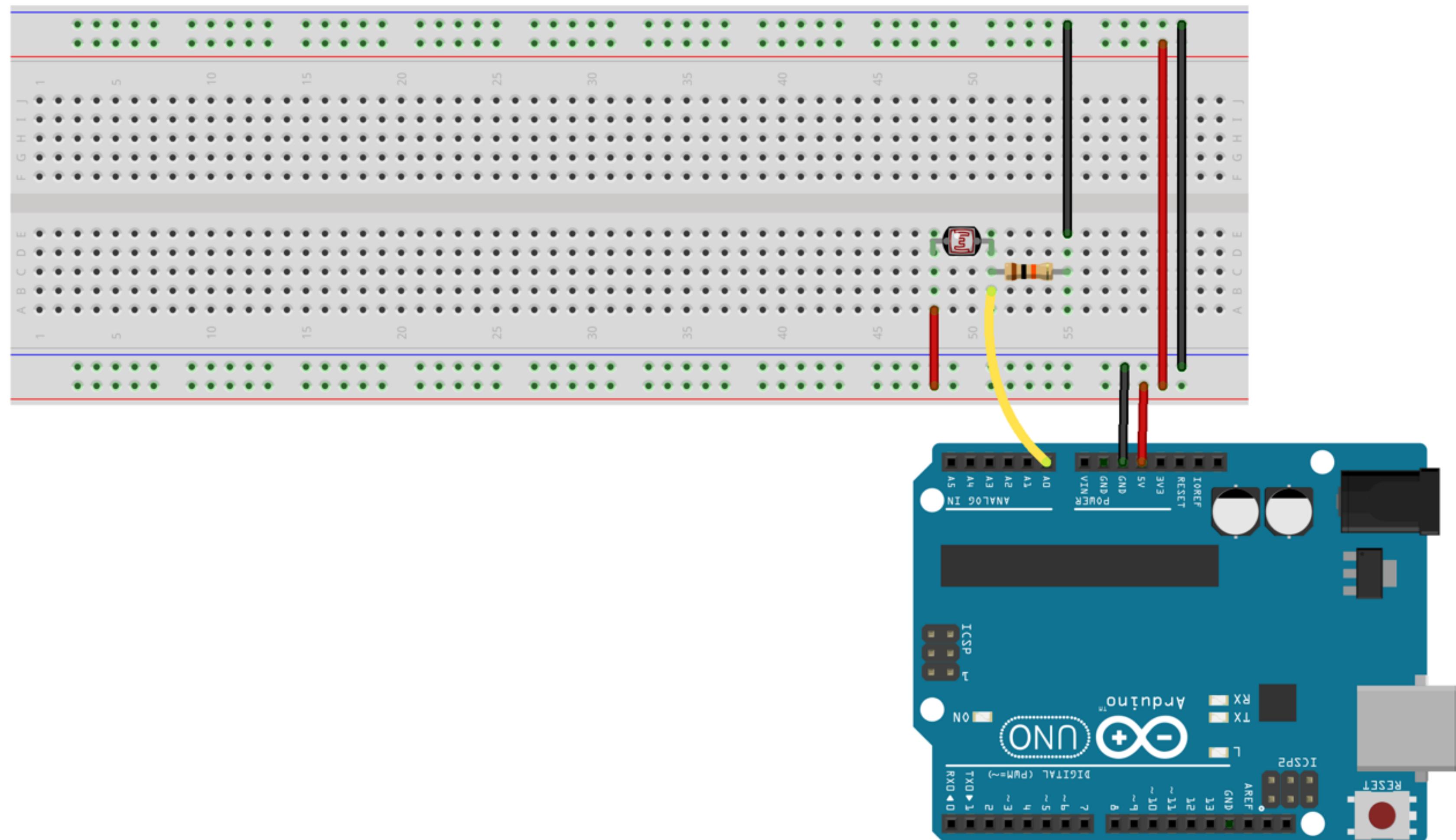


hands on analog

github.com/fabiantheblind/analogio



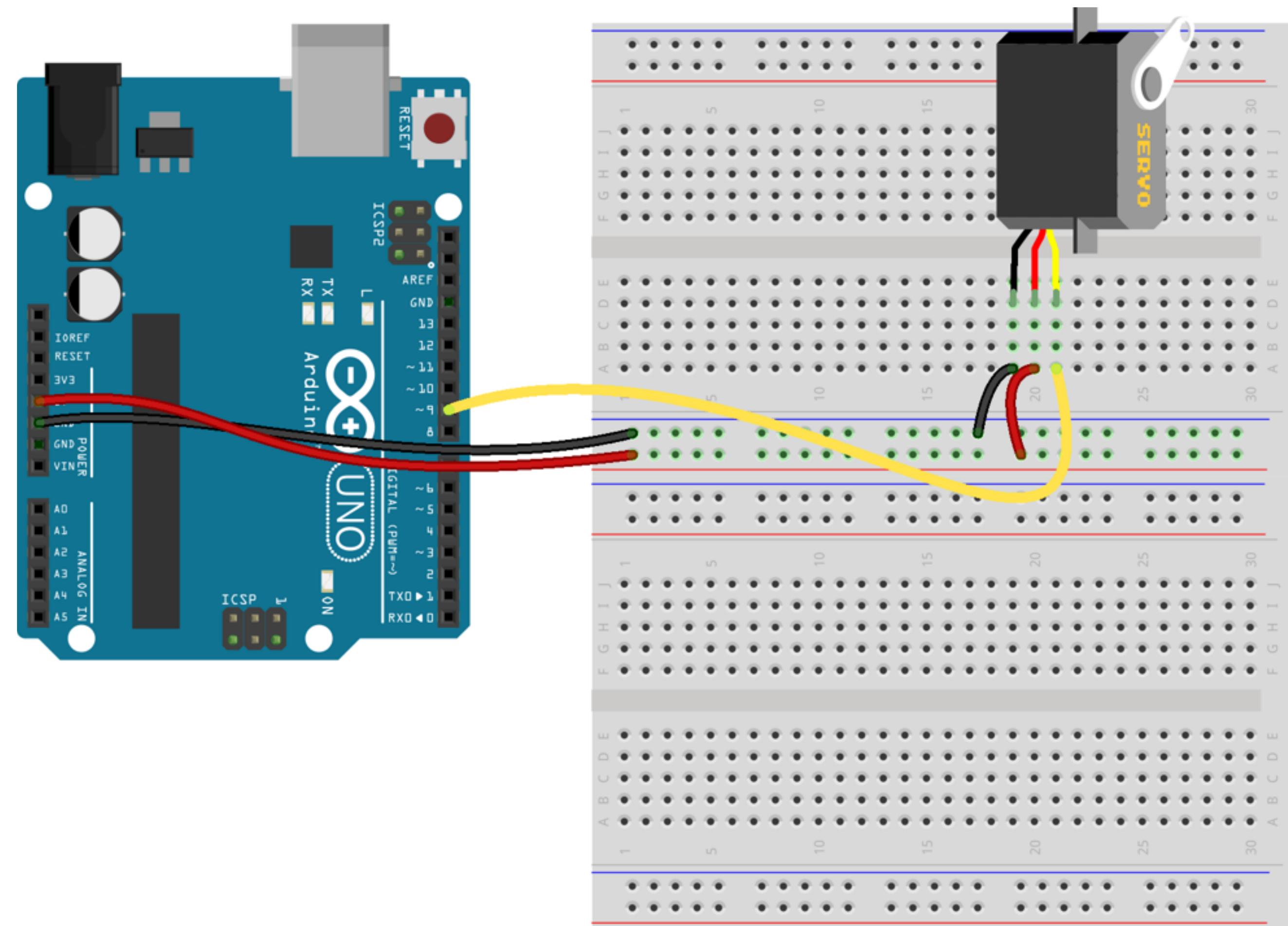


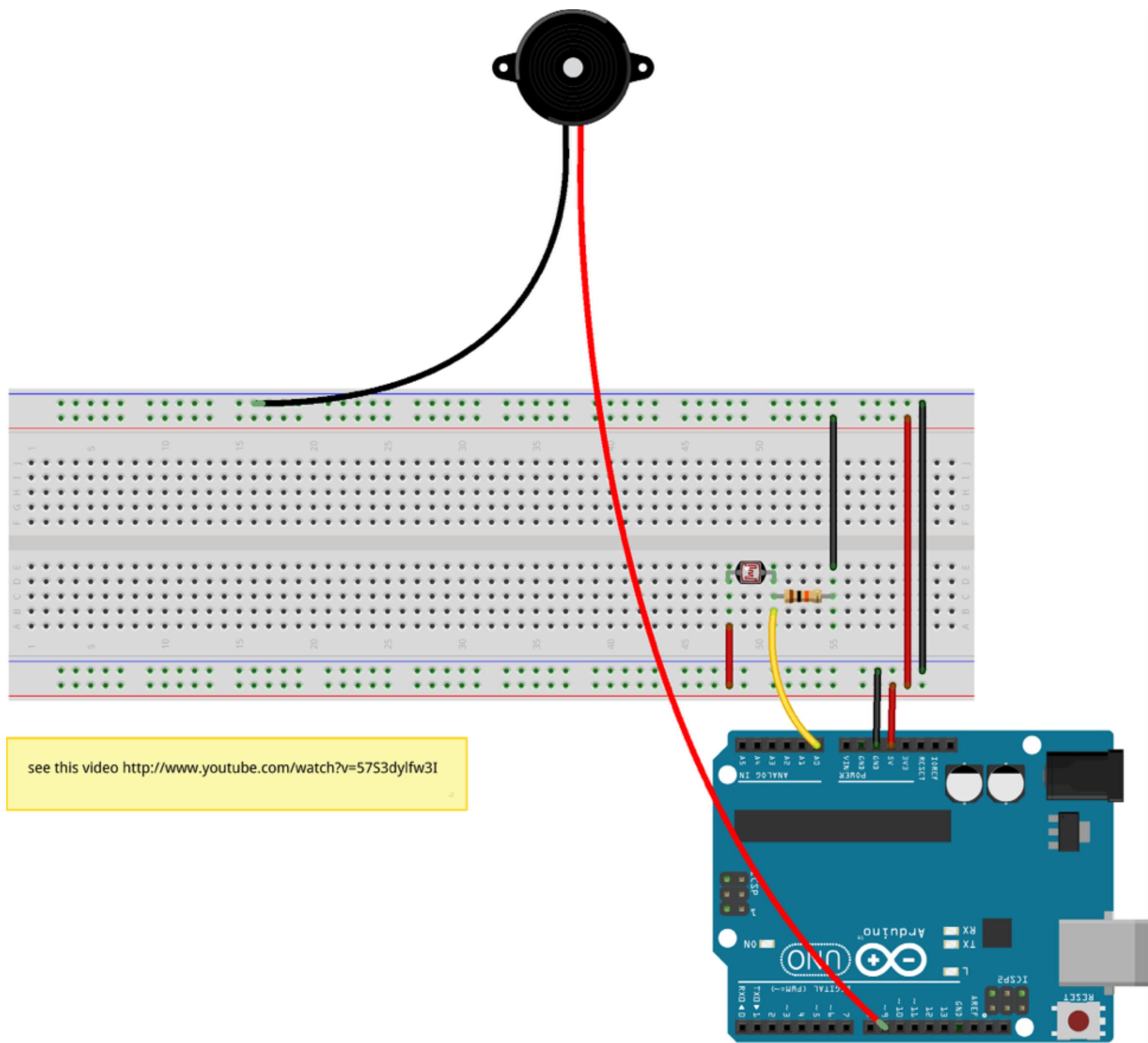


HERZFASSEN (2004)

<http://herzfassen.org/>





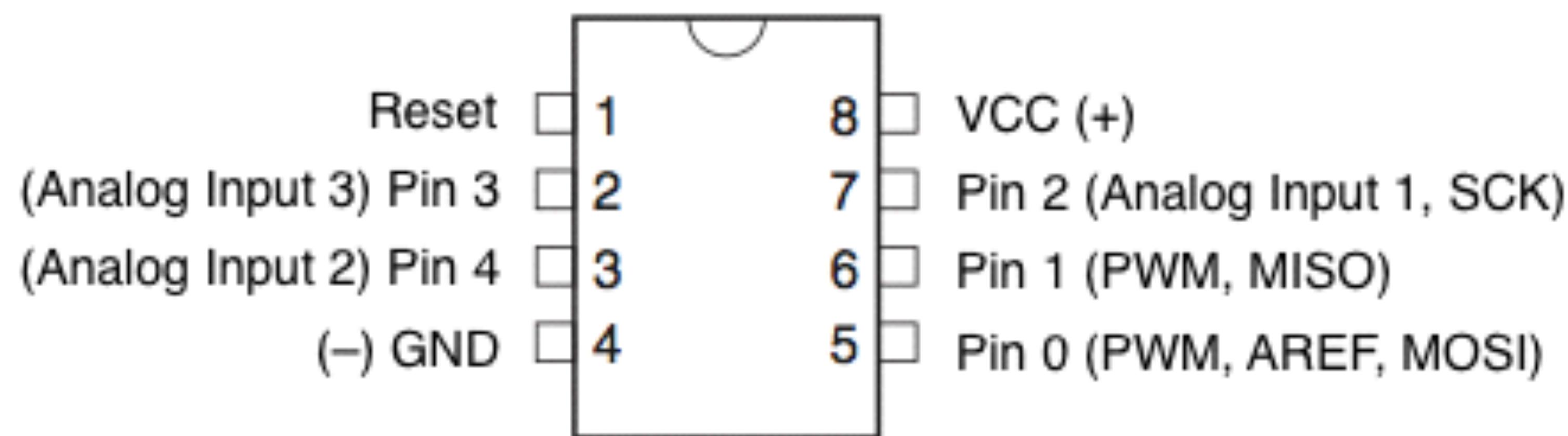


Shrink It

ATtiny 45 || ATtiny 85

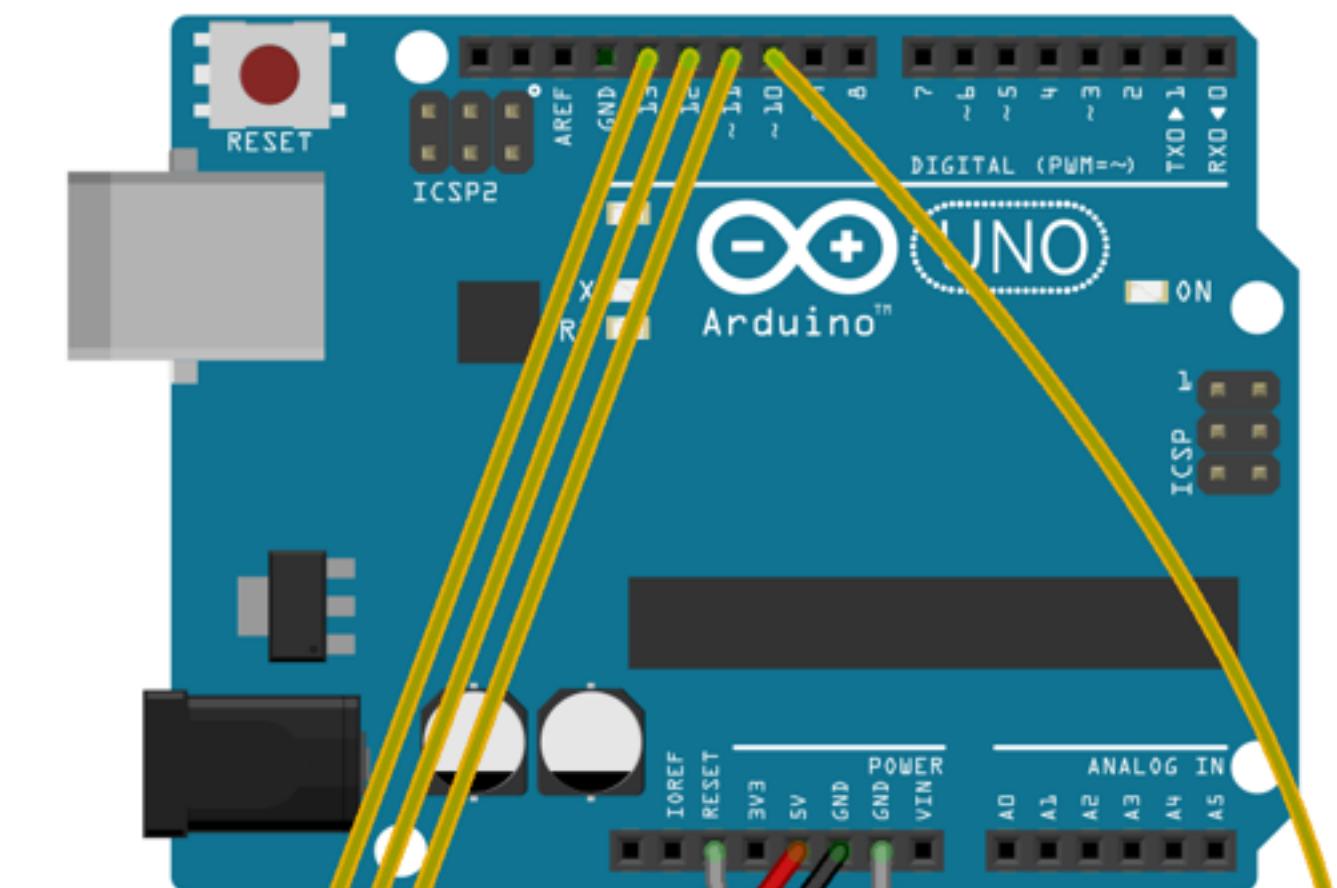
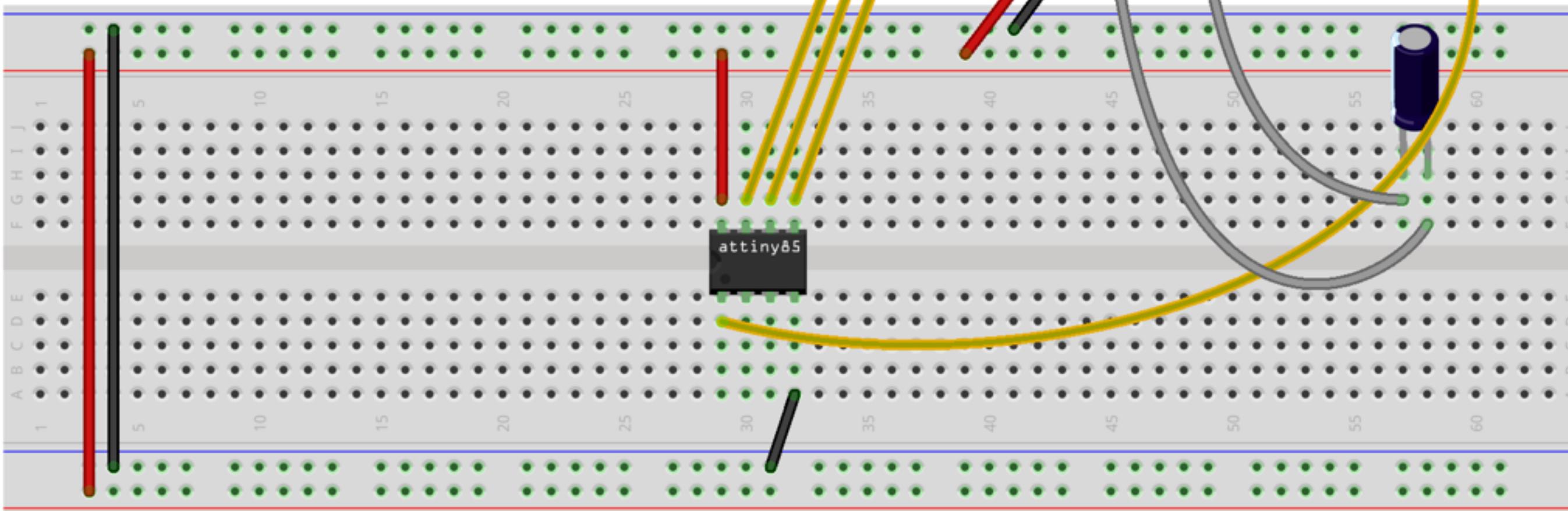


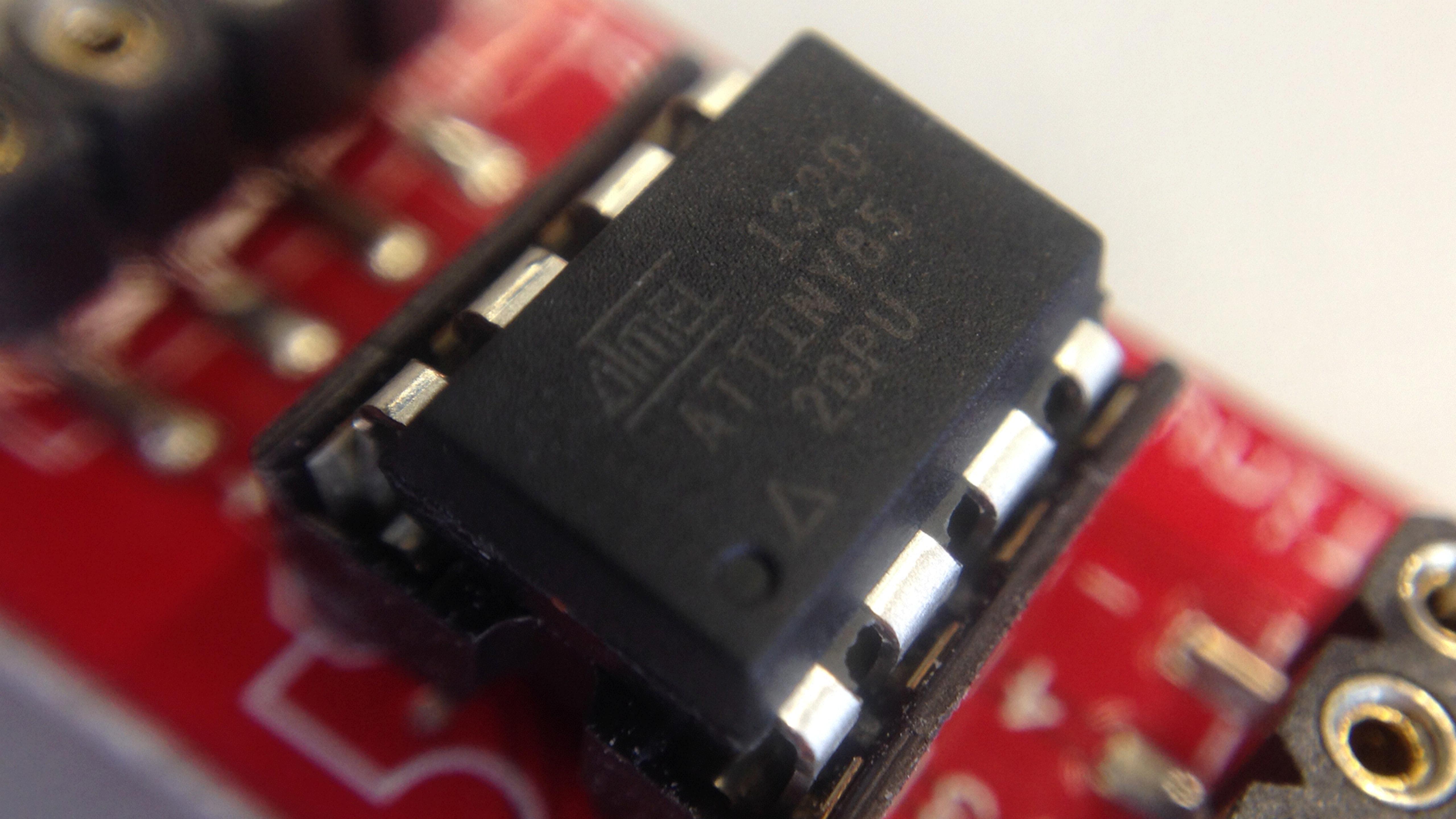
ATtiny 85

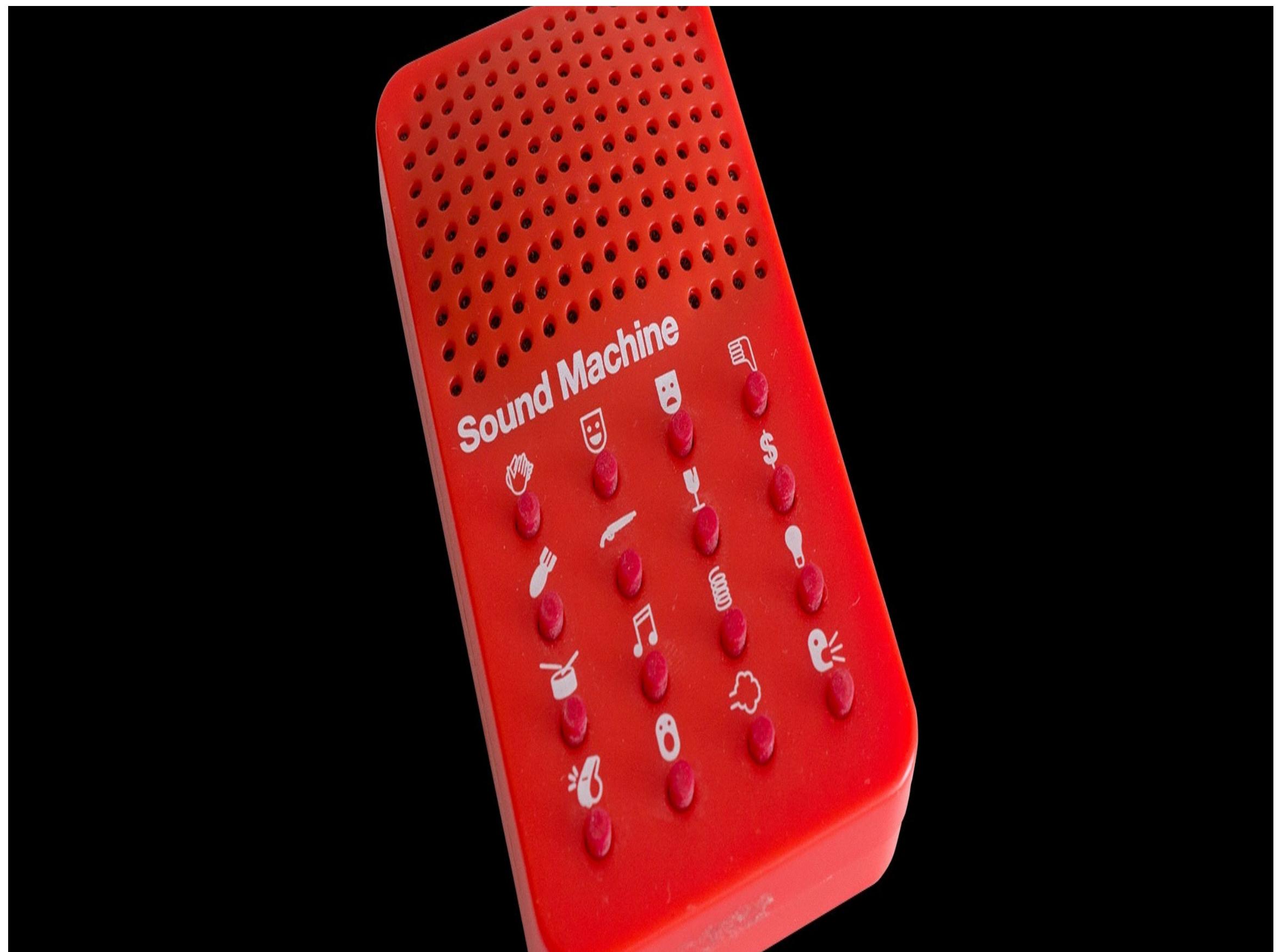
ATtiny45 / ATtiny85

To program a Attiny 45 85 you need:
- Arduino IDE 0.22 (current 1.0.5 does not work)
- this hardware library <http://highlowtech.org/?p=1229>
(also in the repo. Thanks highlowtech group)
- a 10uf capacitor for preventing reset on programming

See the tutorials on how to set it up @
<http://highlowtech.org/?p=1229>







?

hands on c2c

(computer 2 computer communication)

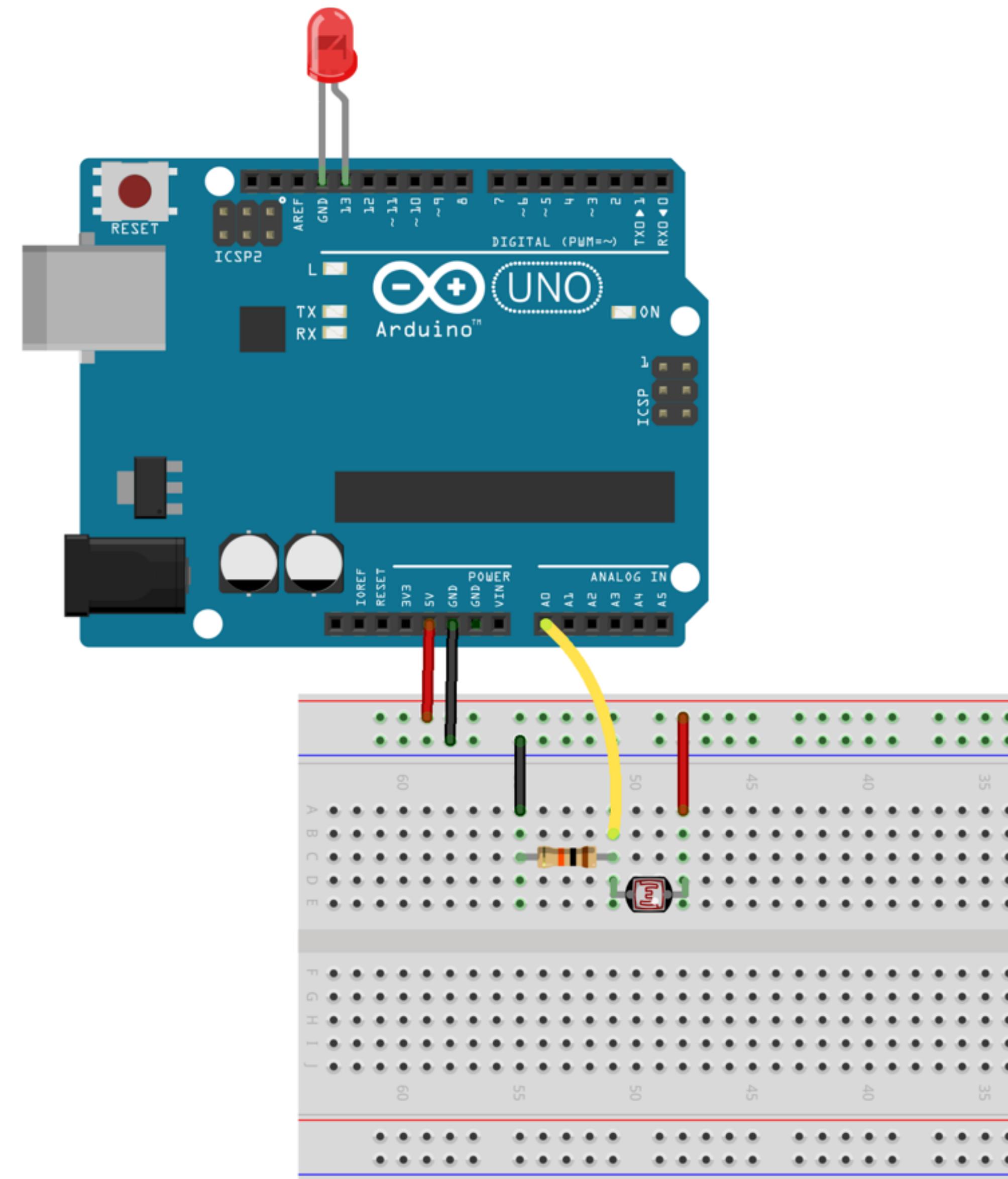
github.com/fabiantheblind/c2c

Serial Ports

00_Serial-Ports-And-Fritzing/list_serial_ports/list_serial_ports.pde

Fritzing

00_Serial-Ports-And-Fritzing/fritzing/c2c.fzz



Arduino → Processing

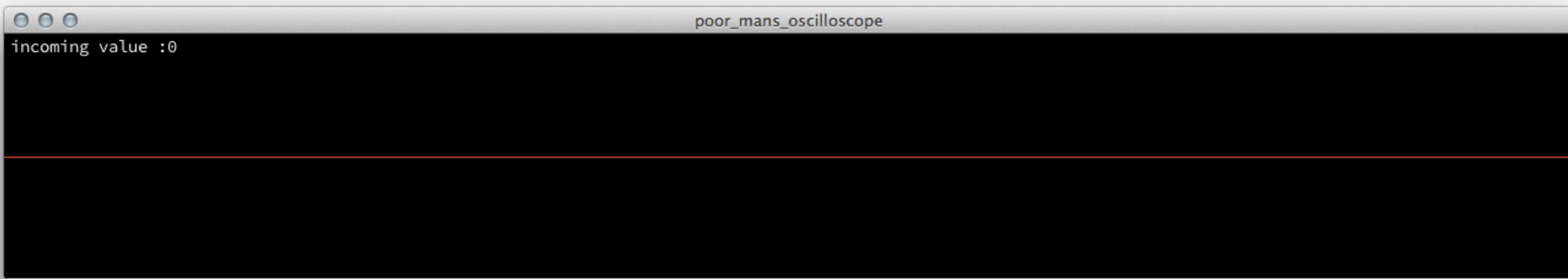
01_Arduino-2-Processing

Arduino → Processing

02_Processing-2-Arduino

Arduino → Processing

03_Poor-Mans-Oscilloscope

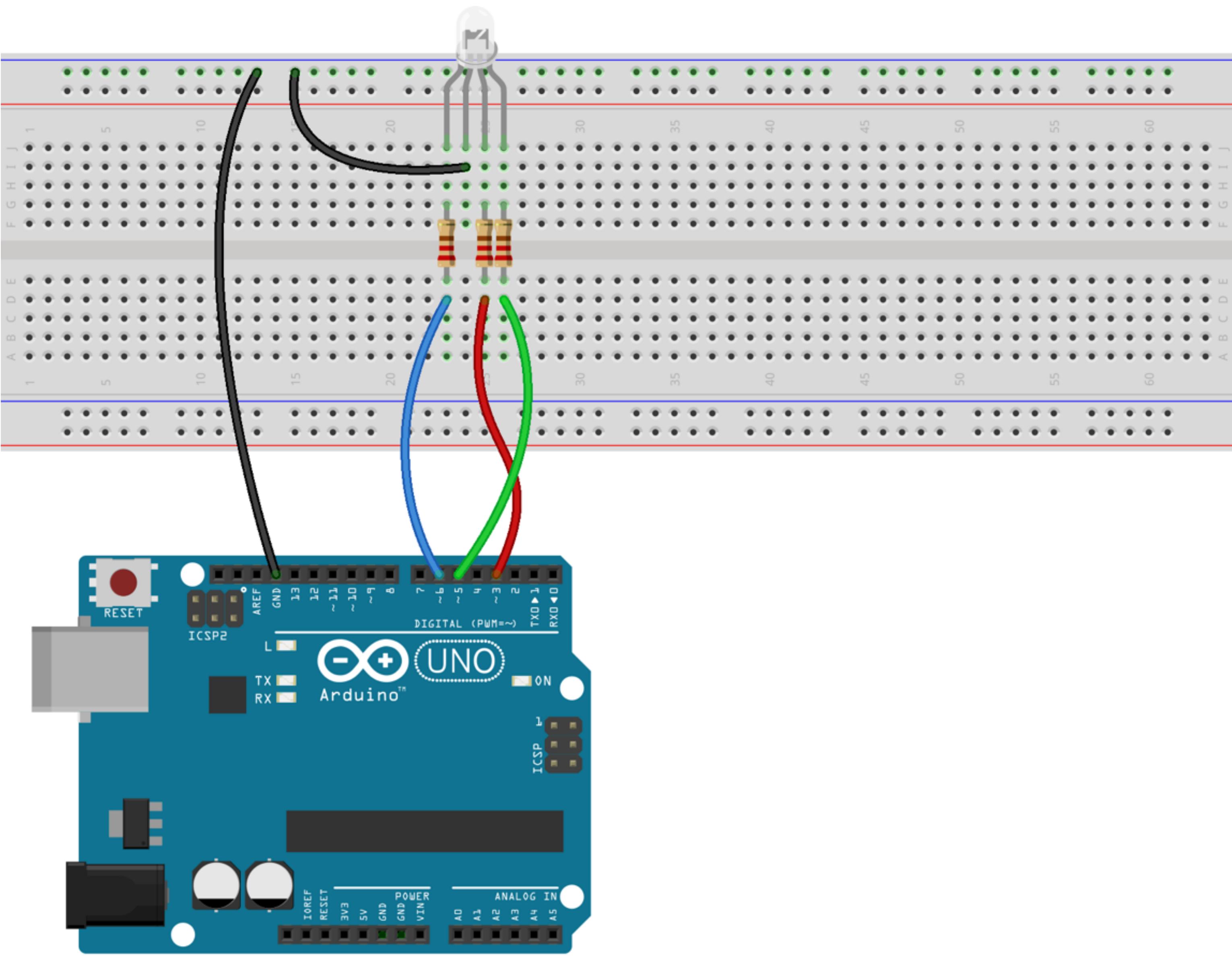


Processing ↘ ↙ Arduino

04_Round-Trip

Arduino → Processing

05_Processing-2-Arduino-Motion-Detection



Arduino → Processing

06_Processing-2-Arduino-Multiple-Values

That's all Folks!