

By The Way

James Lyne:
Everyday cybercrime – and what you can do about it

Emails verschlüsseln

[MAC]: gpgtools.org [WIN]: gpg4win.org

moron-zirfas@fh-potsdam.de

„der elektrische Schaltkreis, eine Erweiterung des Zentralen Nervensystems“

Marshall McLuhan



Interface Werkstatt Blockseminar || Day Three



easy-macro.com

Day Three

Tinker Time

PHYSICAL COMPUTING

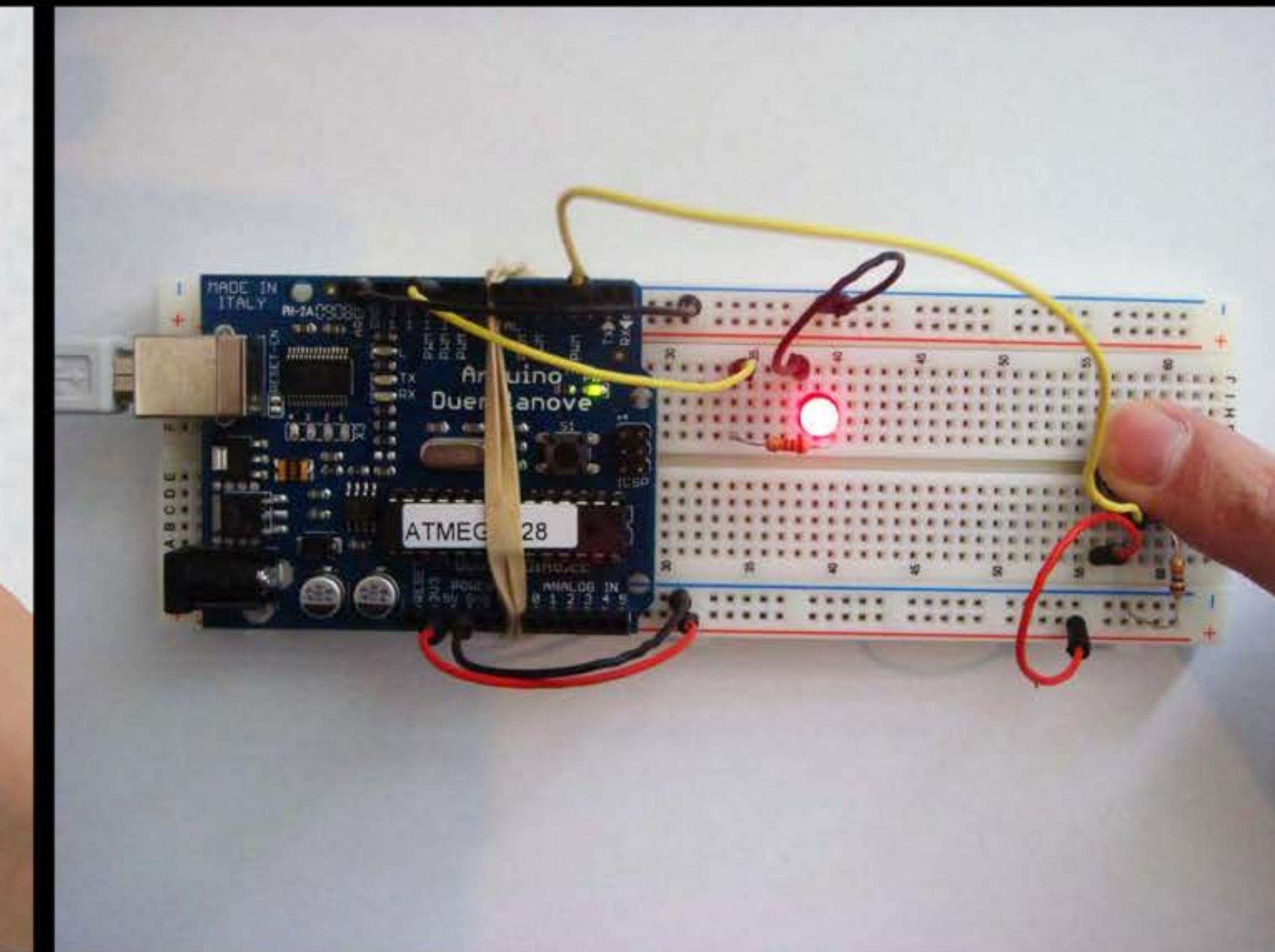
... bedeutet im weitesten Sinne, interaktive, physische Systeme durch die Verwendung von Hardware und Software zu erstellen. Diese Systeme reagieren auf Ereignisse in der realen, analogen Welt und/oder wirken auf sie ein.

TED: Massimo Banzi

how arduino is open sourcing imagination

AUSGABE/ AKTUATOREN	EINGABE/ SENSOREN
PUSHBUTTON	KIPPSCHALTER
DRUCKSENSOR	BIEGESENSOR
PIEZODRAHT	POTENTIOMETER
PIEZODRAHT	LICHTSENSOR
PIEZODRAHT	LICHTSCHRANKE (LED/LASER)
PIEZODRAHT	ABSTANDS- MESSER (ULTRASCHALL)
PIEZODRAHT	TEMPERATUR- SENSOR
PIEZODRAHT	MIKROFON (KONDENSATOR)
PIEZODRAHT	WIDERSTANDS- MESSUNG
LAUTSPRECHER	PRÄSENZSENSOR (KAPAZITÄT)
SERVO	BESCHLEUNIGUNGS- SENSOR (ACCELEROMETER)
ELEKTROMOTOR/ STEPPER (O. VIBRAMOTOR)	GYRO
ELEKTROMAGNET	CO2 SENSOR
FLIPDOT	KOMPLEXERE SENSOREN*
MUSCLEWIRE	KEYBOARD
ELEKTRO- MAGNETISCHER IMPULSZÄHLER	MAUS
LCD/SEGMENT- DISPLAY	KAMERA
THERMODRUCKER	COMPUTER (DATEN / API)
COMPUTER	...
...	

* Z.B. ALKOHOL, GAS, FEUCHTIGKEIT,
LUFTDRUCK, WIND, KOMPASS, GPS, RFID, ...



AUSGABE/ AKTUATOREN	EINGABE/ SENSOREN
LED/ DIODE	PUSHBUTTON
LED/ MATRIX	KIPPSCHALTER
ELEKTRO- LUMINESENZ DRAHT	DRUCKSENSOR
PIEZO	BIEGESENSOR
LAUTSPRECHER	PIEZOMETER
SERVO	POTENTIOMETER
ELEKTROMOTOR/ STEPPER (O. VIBRAMOTOR)	LICHTSENSOR
ELEKTROMAGNET	LICHTSCHRANKE (LED/LASER)
FLIPDOT	ABSTANDS- MESSER (ULTRASCHALL)
MUSCLEWIRE	TEMPERATUR- SENSOR
ELEKTRO- MAGNETISCHER IMPULSZÄHLER	MIKROFON (KONDENSATOR)
LCD/SEGMENT- DISPLAY	WIDERSTANDS- MESSUNG
THERMODRUCKER	PRÄSENZSENSOR (KAPAZITÄT)
COMPUTER	BESCHLEUNIGUNGS- SENSOR (ACCELEROMETER)
...	GYRO
	CO2 SENSOR
	KOMPLEXERE SENSOREN*
	KEYBOARD
	MAUS
	KAMERA
	COMPUTER (DATEN / API)
	...

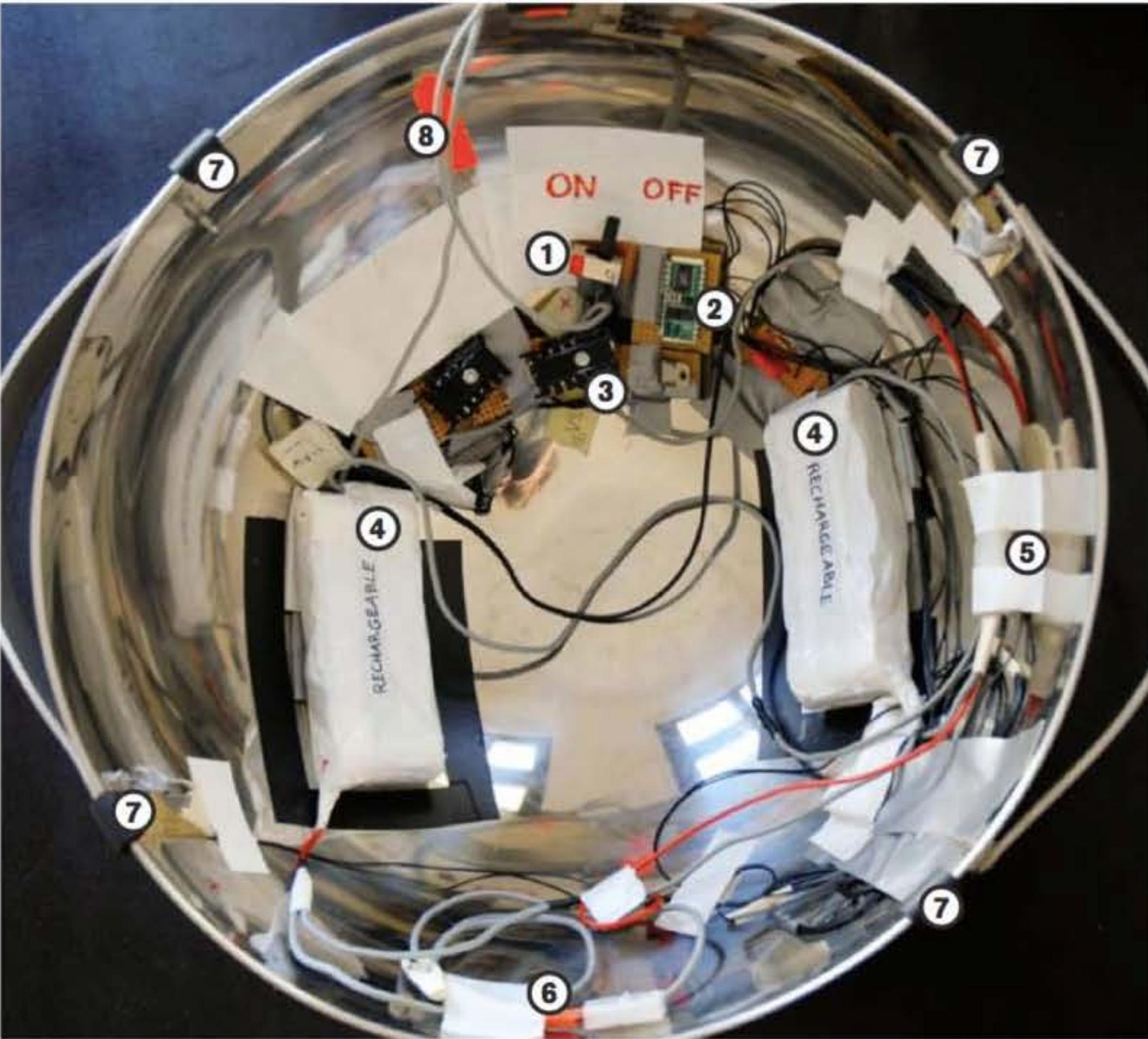
* Z.B. ALKOHOL, GAS, FEUCHTIGKEIT,
LUFTDRUCK, WIND, KOMPASS, GPS, RFID, ...



herzfassen



Interface Werkstatt Blockseminar || Day Three



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)





Interface Werkstatt Blockseminar || Day Three



Interface Werkstatt Blockseminar || Day Three

AUSGABE/ AKTUATOREN	EINGABE/ SENSOREN
PUSHBUTTON	KIPPSCHALTER
DRUCKSENSOR	BIEGESENSOR
PIEZODRAHT	POTENTIOMETER
PIEZOMATRIX	LICHTSENSOR
PIEZOLEUCHTE	LICHTSCHRANKE (LED/LASER)
PIEZOMOTOR	ABSTANDS- MESSER (ULTRASCHALL)
PIEZOKOMMUNIKATION	TEMPERATUR- SENSOR
PIEZOGitarre	MIKROFON (KONDENSATOR)
PIEZOGitarre	WIDERSTANDS- MESSUNG
PIEZOGitarre	PRÄSENZSENSOR (KAPAZITÄT)
PIEZOGitarre	BESCHLEUNIGUNGS- SENSOR (ACCELEROMETER)
PIEZOGitarre	GYRO
PIEZOGitarre	CO2 SENSOR
PIEZOGitarre	KOMPLEXERE SENSOREN*
PIEZOGitarre	KEYBOARD
PIEZOGitarre	MAUS
PIEZOGitarre	KAMERA
PIEZOGitarre	COMPUTER (DATEN / API)
PIEZOGitarre	...

* Z.B. ALKOHOL, GAS, FEUCHTIGKEIT,
LUFTDRUCK, WIND, KOMPASS, GPS, RFID, ...



Maison Hermès x Tokujin Yoshioka: Scarves in Shop Window, Tokyo

AUSGABE/ AKTUATOREN	EINGABE/ SENSOREN
PUSHBUTTON	KIPPSCHALTER
DRUCKSENSOR	BIEGESENSOR
PIEZODRAHT	POTENTIOMETER
PIEZODRAHT	LICHTSENSOR
PIEZODRAHT	LICHTSCHRANKE (LED/LASER)
PIEZODRAHT	ABSTANDS- MESSER (ULTRASCHALL)
PIEZODRAHT	TEMPERATUR- SENSOR
PIEZODRAHT	MIKROFON (KONDENSATOR)
PIEZODRAHT	WIDERSTANDS- MESSUNG
LAUTSPRECHER	PRÄSENZSENSOR (KAPAZITÄT)
SERVO	BESCHLEUNIGUNGS- SENSOR (ACCELEROMETER)
ELEKTROMOTOR/ STEPPER (O. VIBRAMOTOR)	GYRO
(VENTILATOR)	CO2 SENSOR
ELEKTROMAGNET	KOMPLEXERE SENSOREN*
FLIPDOT	KEYBOARD
MUSCLEWIRE	MAUS
ELEKTRO- MAGNETISCHER IMPULSZÄHLER	KAMERA
LCD/SEGMENT- DISPLAY	COMPUTER (DATEN / API)
THERMODRUCKER	...
COMPUTER	
...	

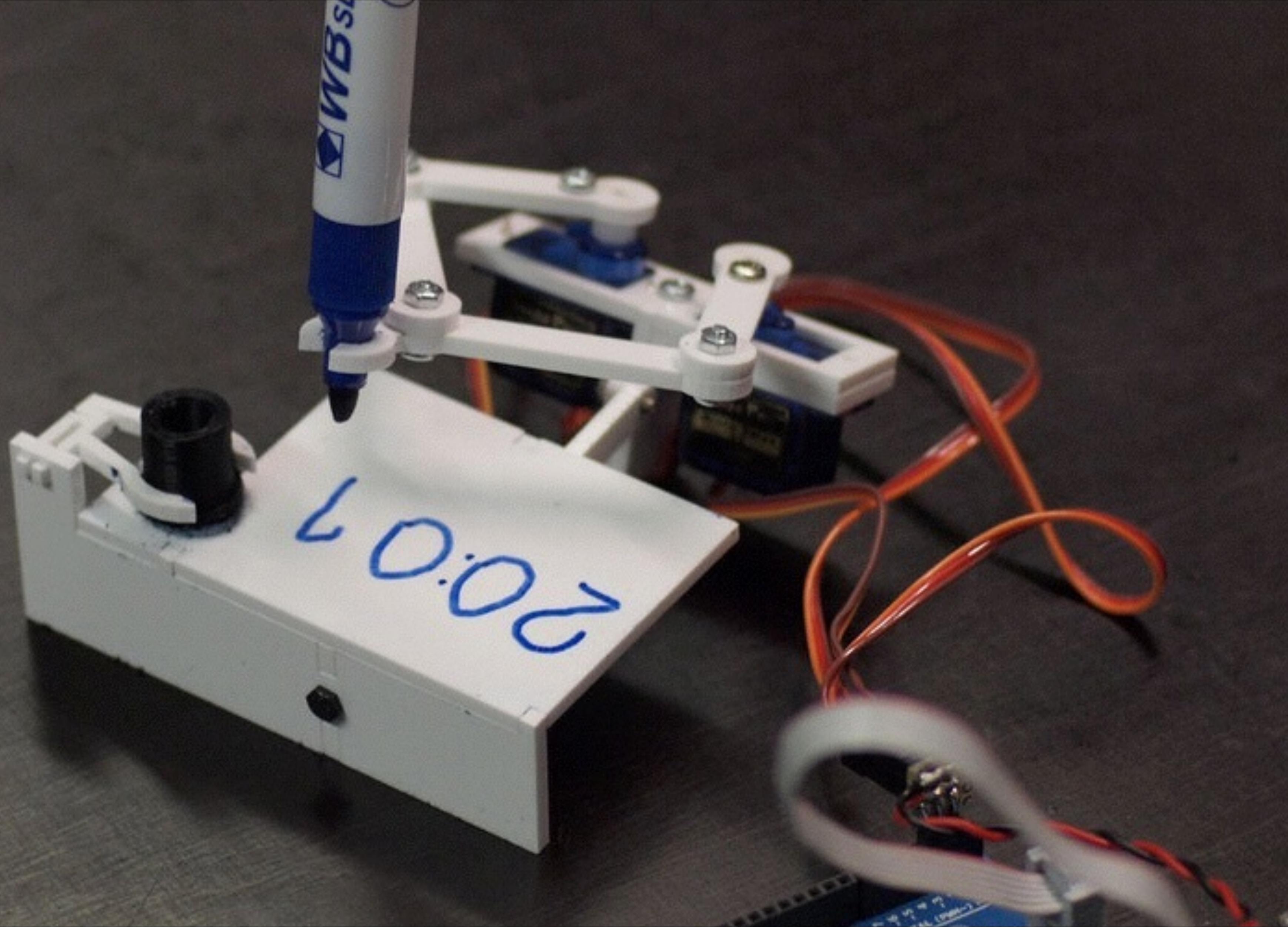
* Z.B. ALKOHOL, GAS, FEUCHTIGKEIT,
LUFTDRUCK, WIND, KOMPASS, GPS, RFID, ...



Universidad de Cartagena



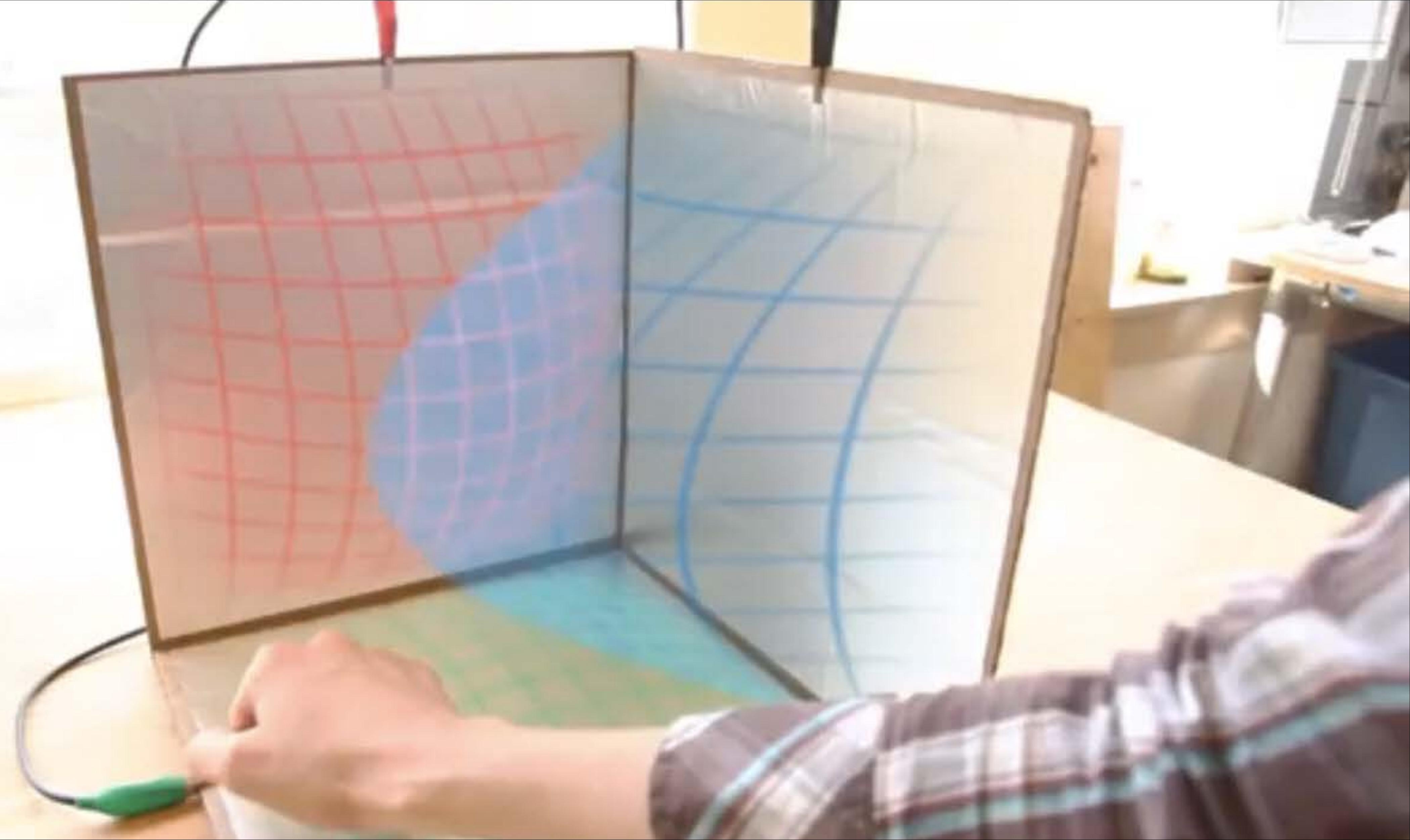
littleprinter.com



Plotclock



Muscle Wire



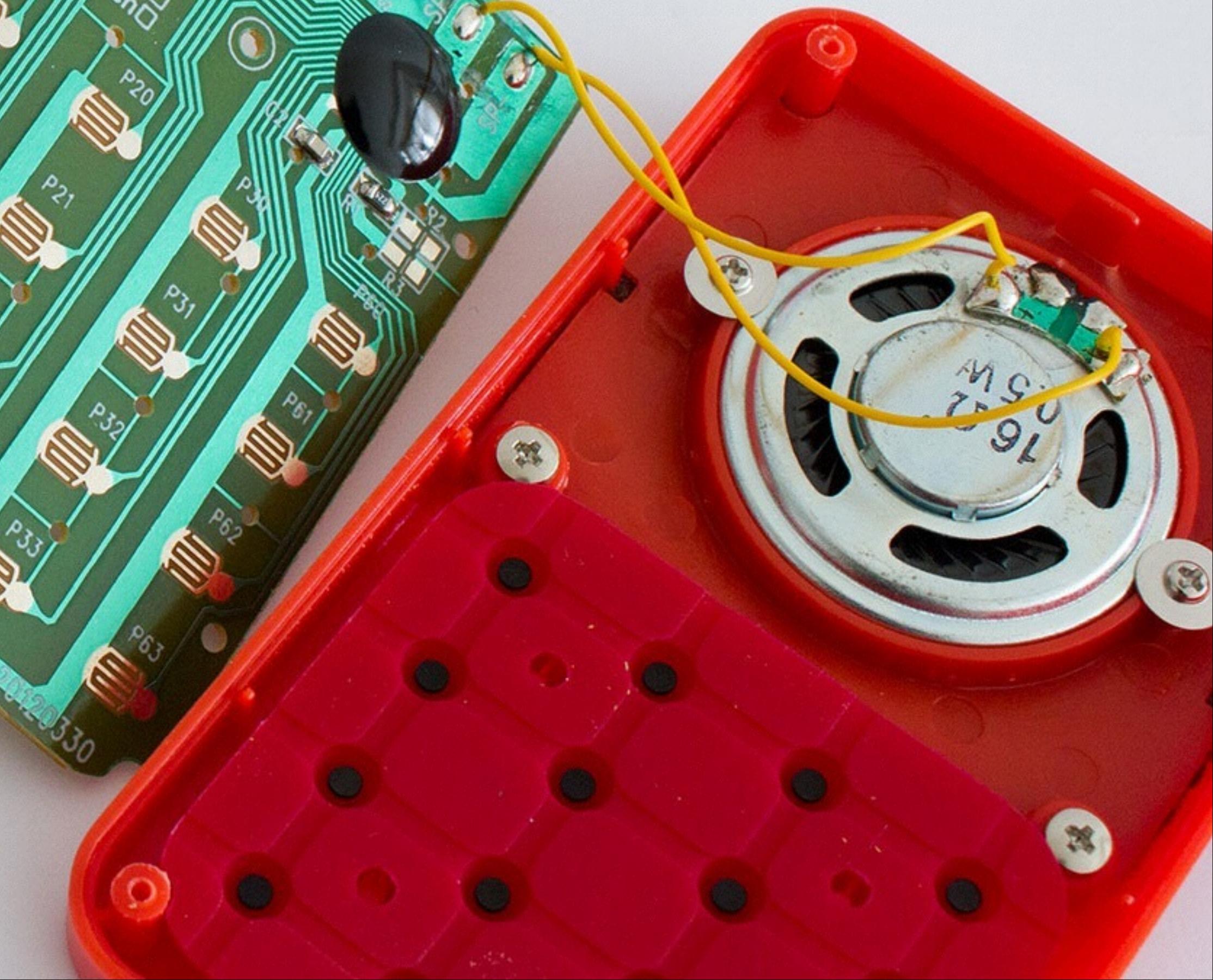
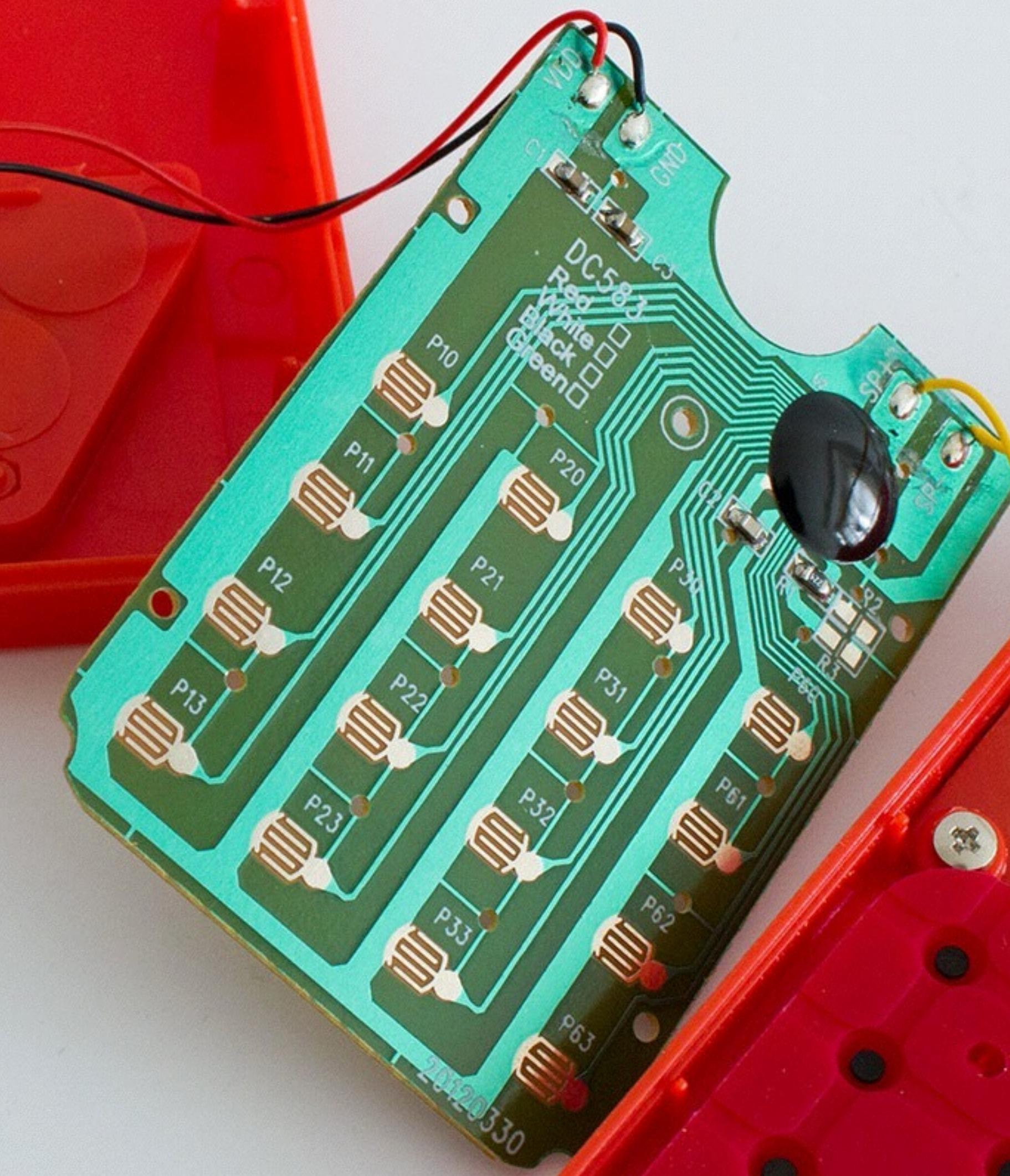
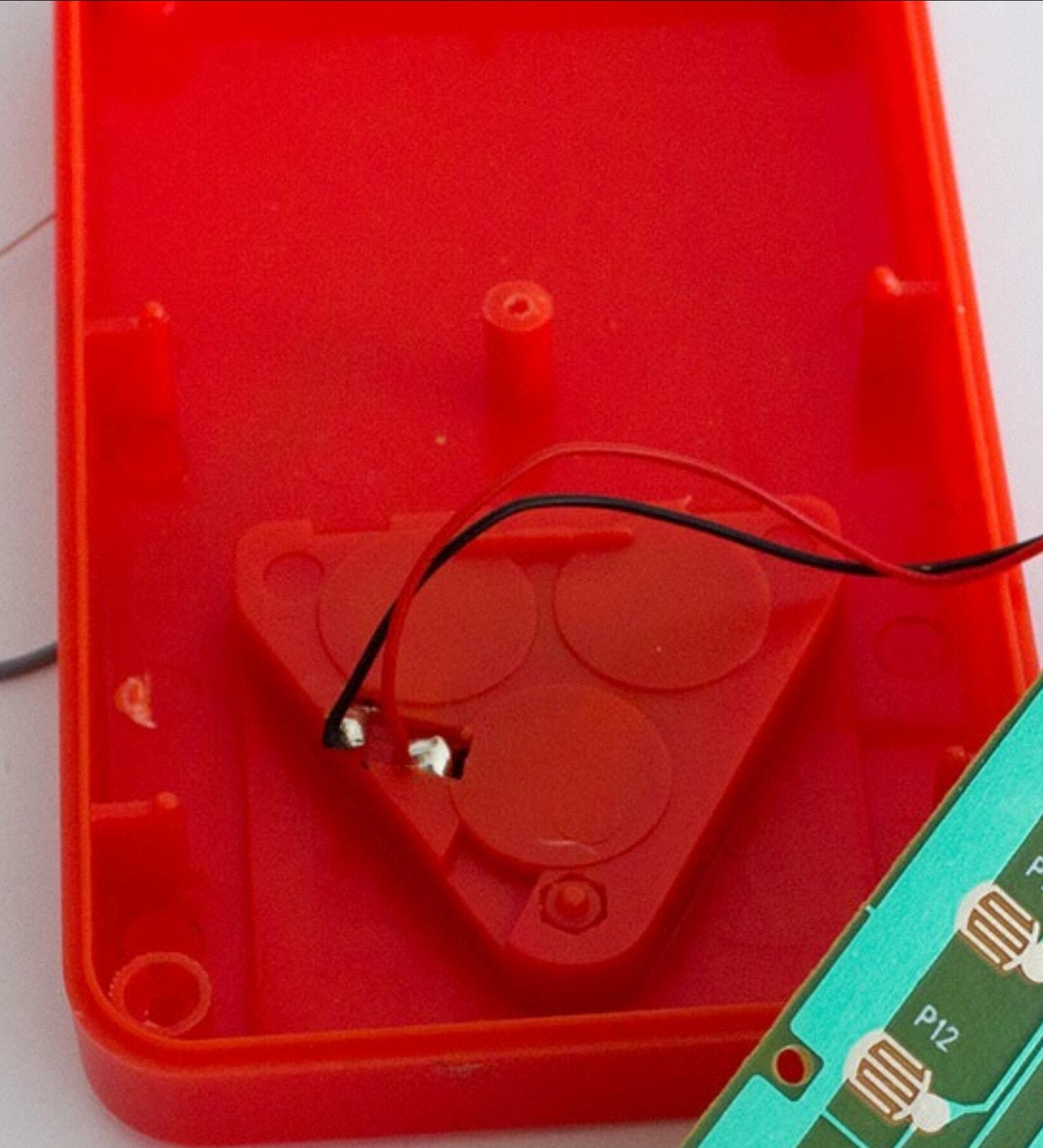
MAKE - Touchless 3D Tracking interface

Decontextualisation





4-40





Be Bold



Be Patient

Documentation

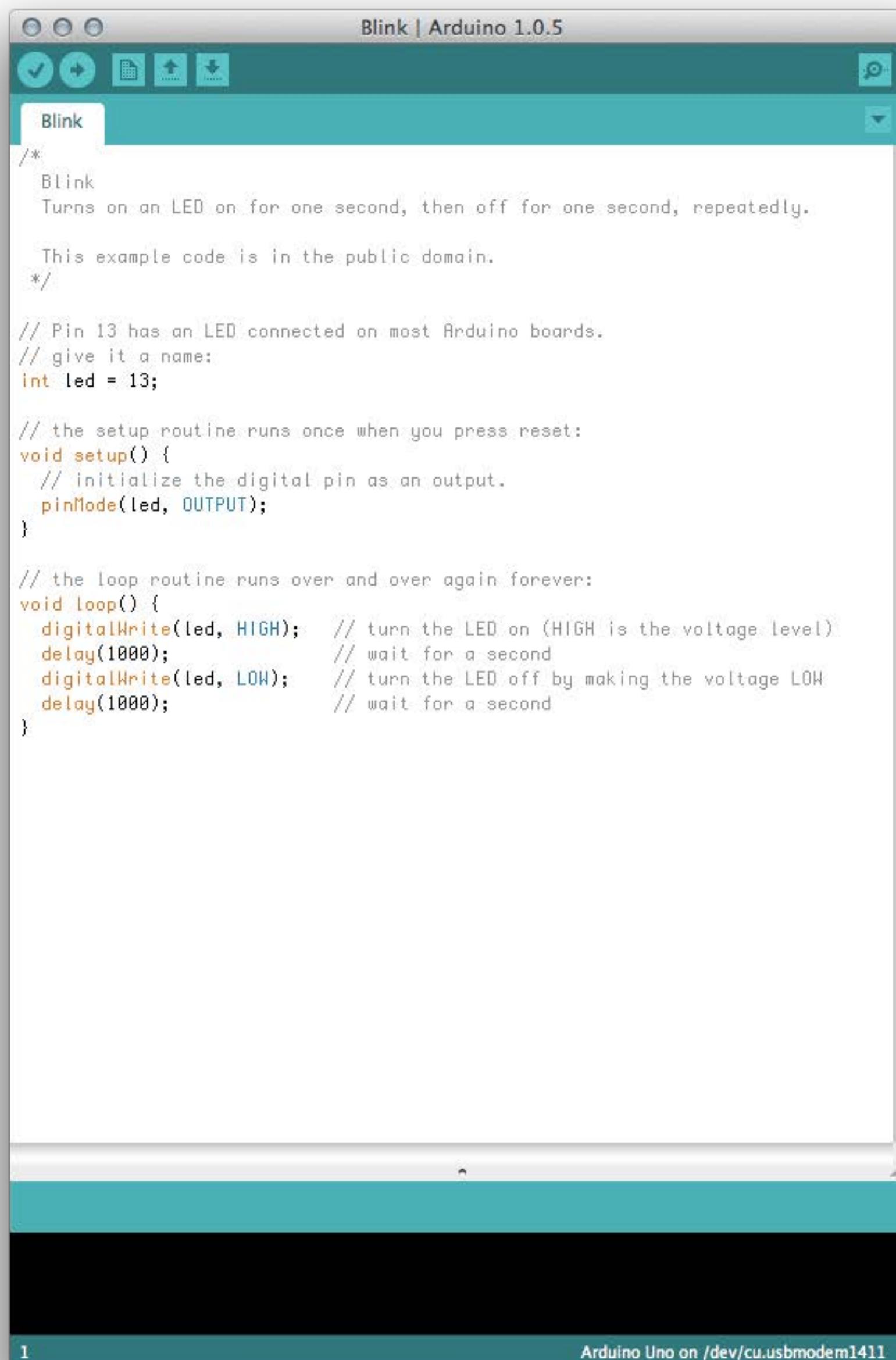
Documentation

Documentation

The logo for fritzing, featuring the word "fritzing" in a white, lowercase, sans-serif font. The letter "o" in each word has a small black circle at its top center. The background is a solid orange rectangle with a thin brown horizontal bar at the bottom.

Keyboard Hack





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.
*/

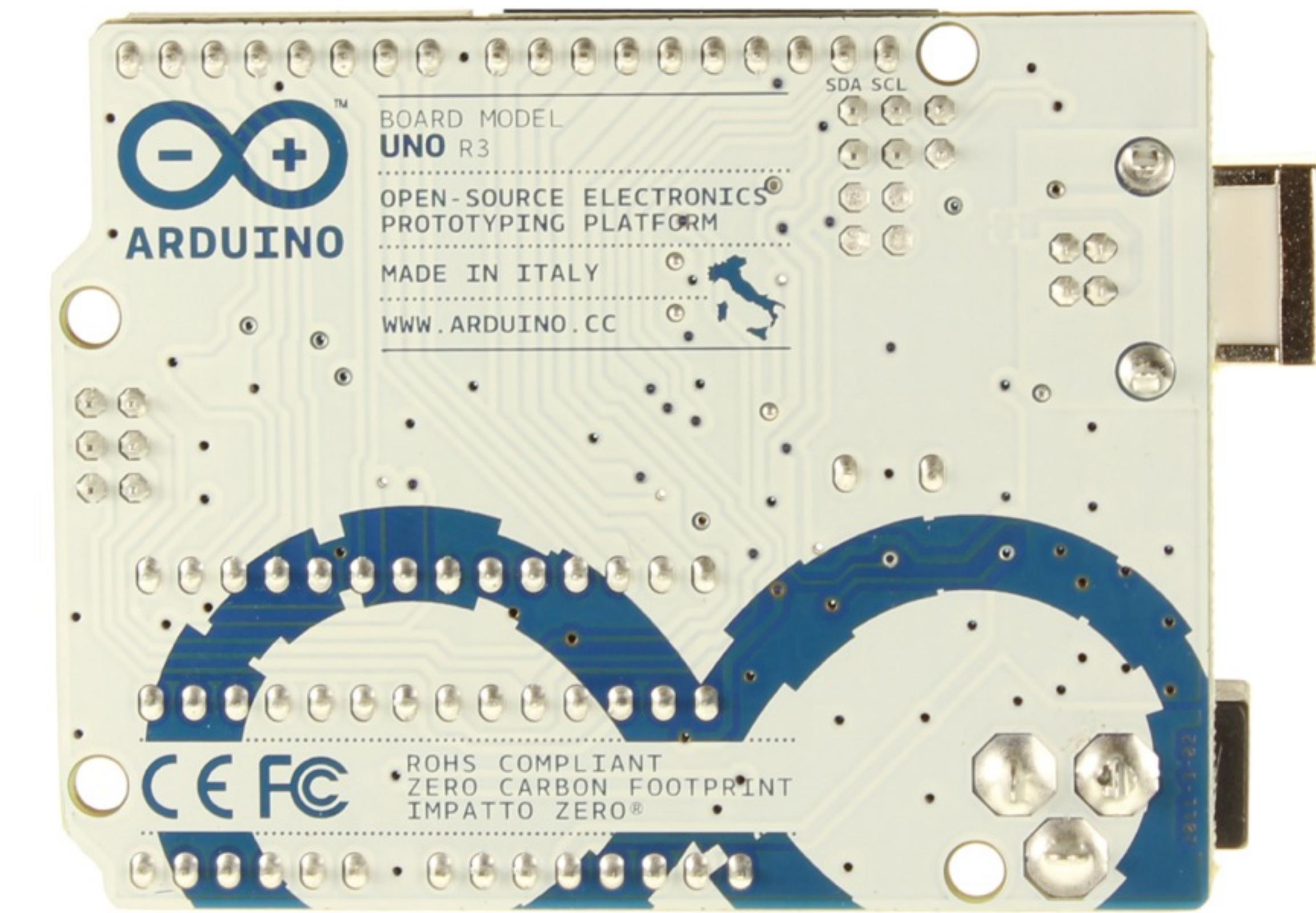
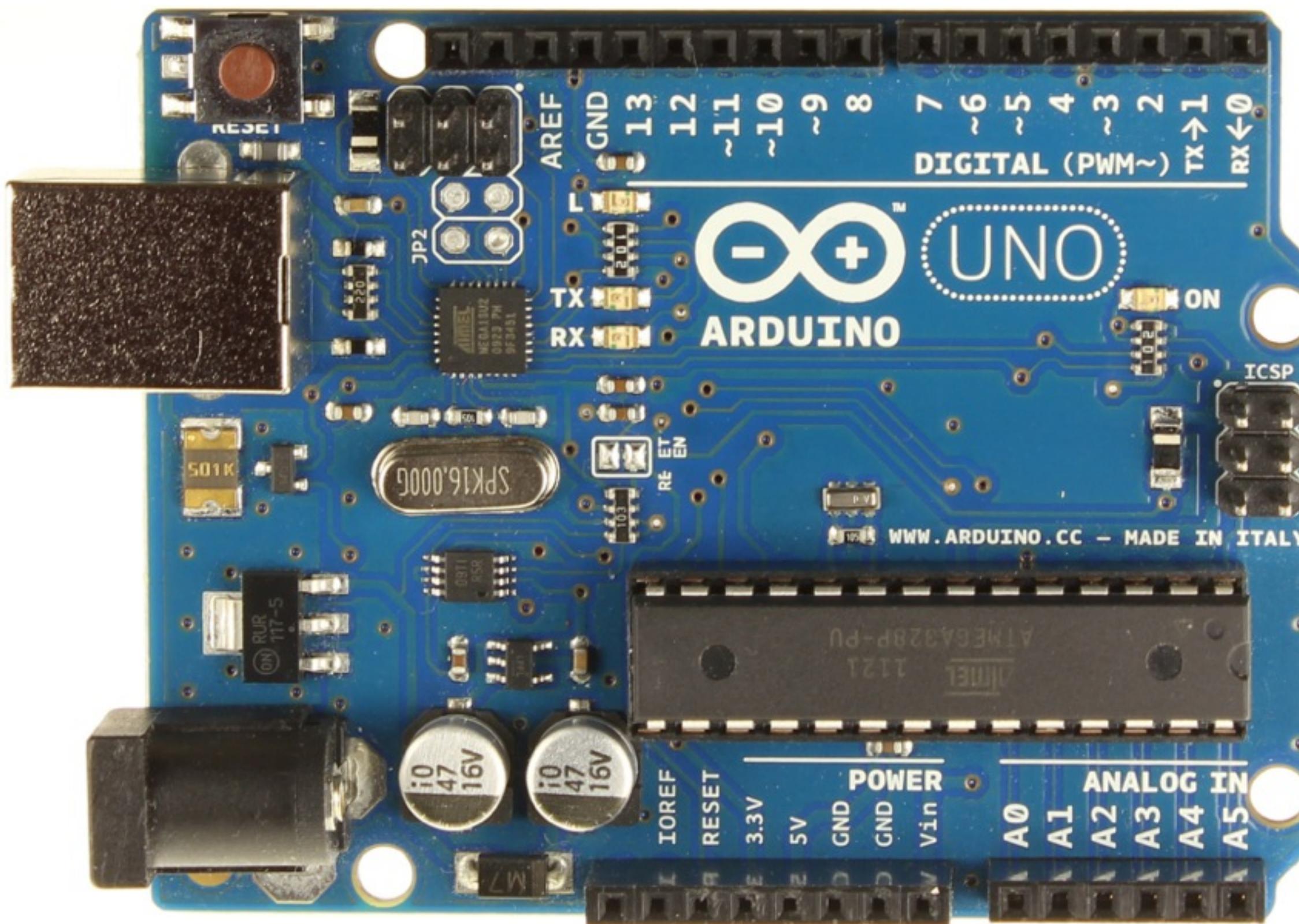
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

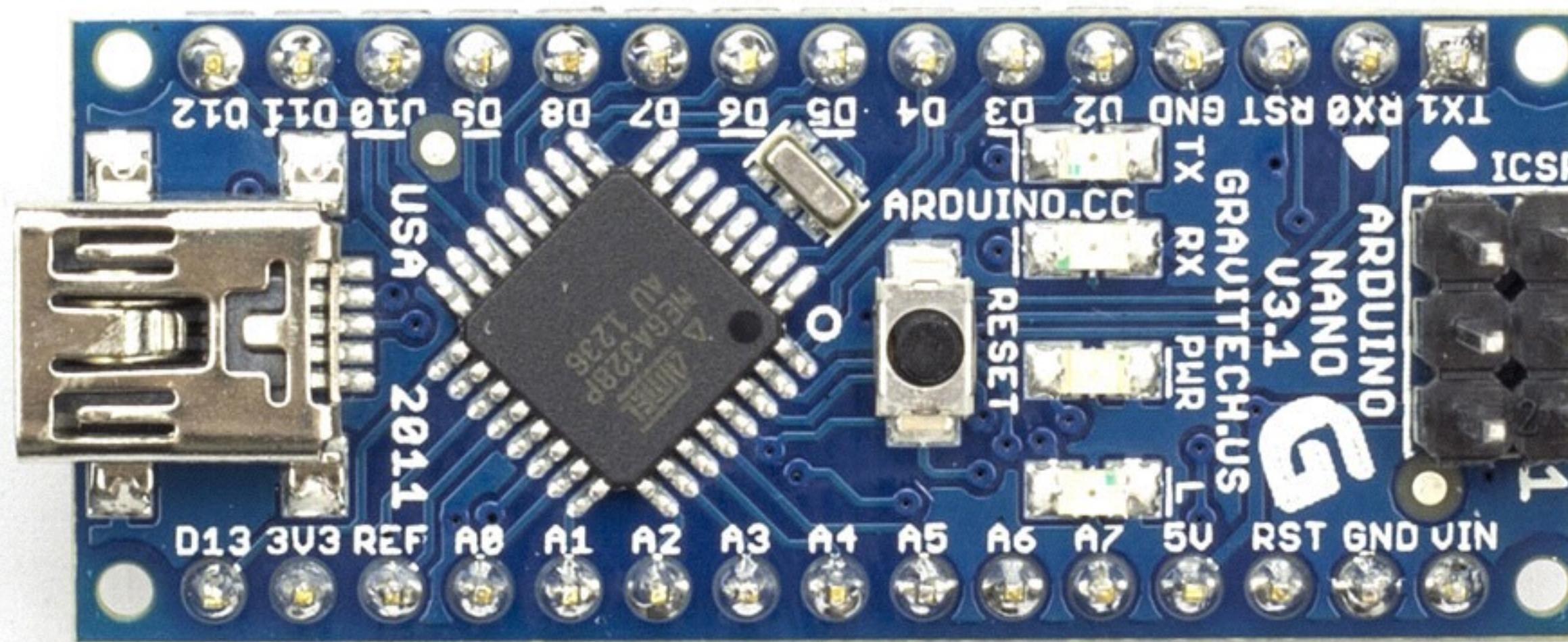
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

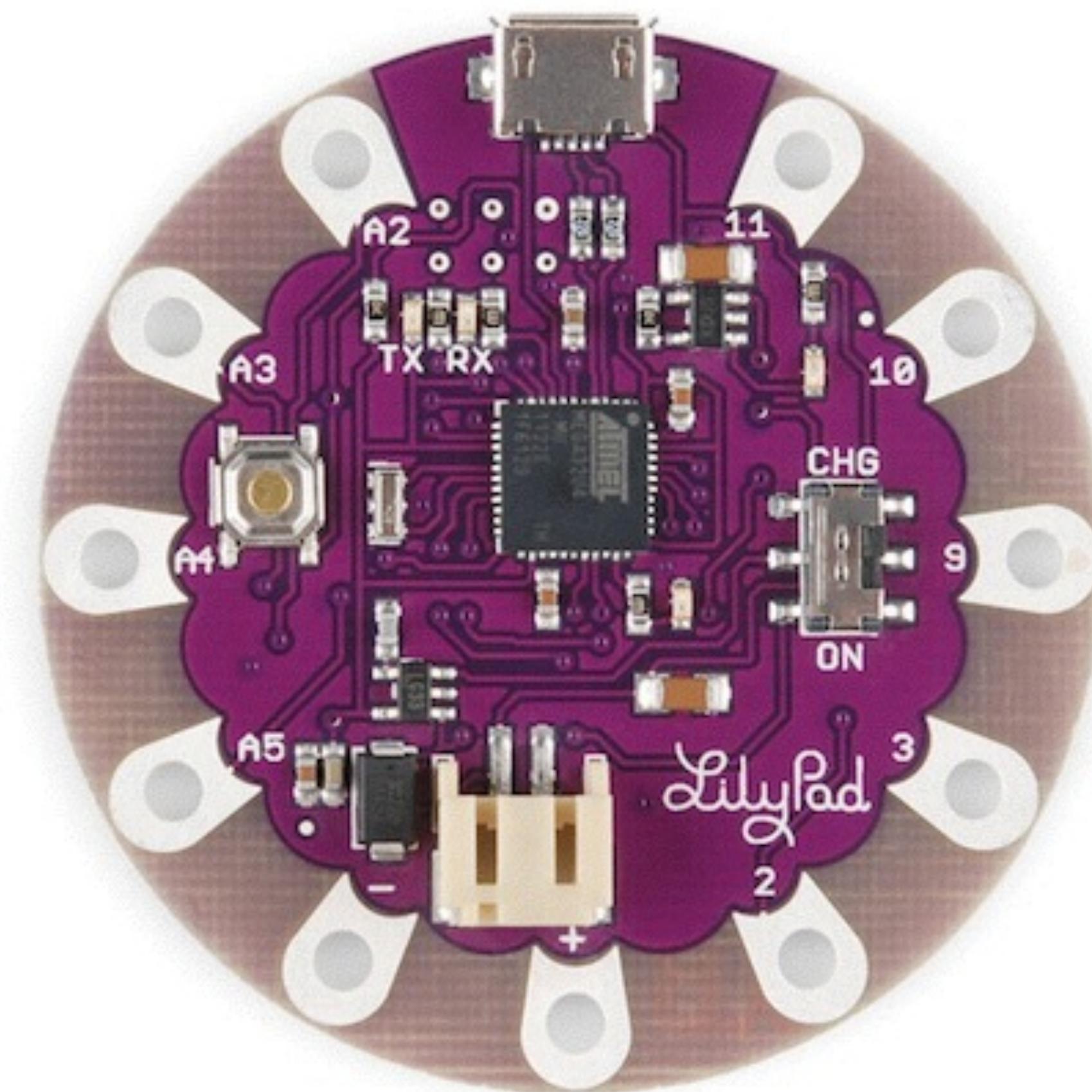
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);      // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(led, LOW);       // turn the LED off by making the voltage LOW
  delay(1000);          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

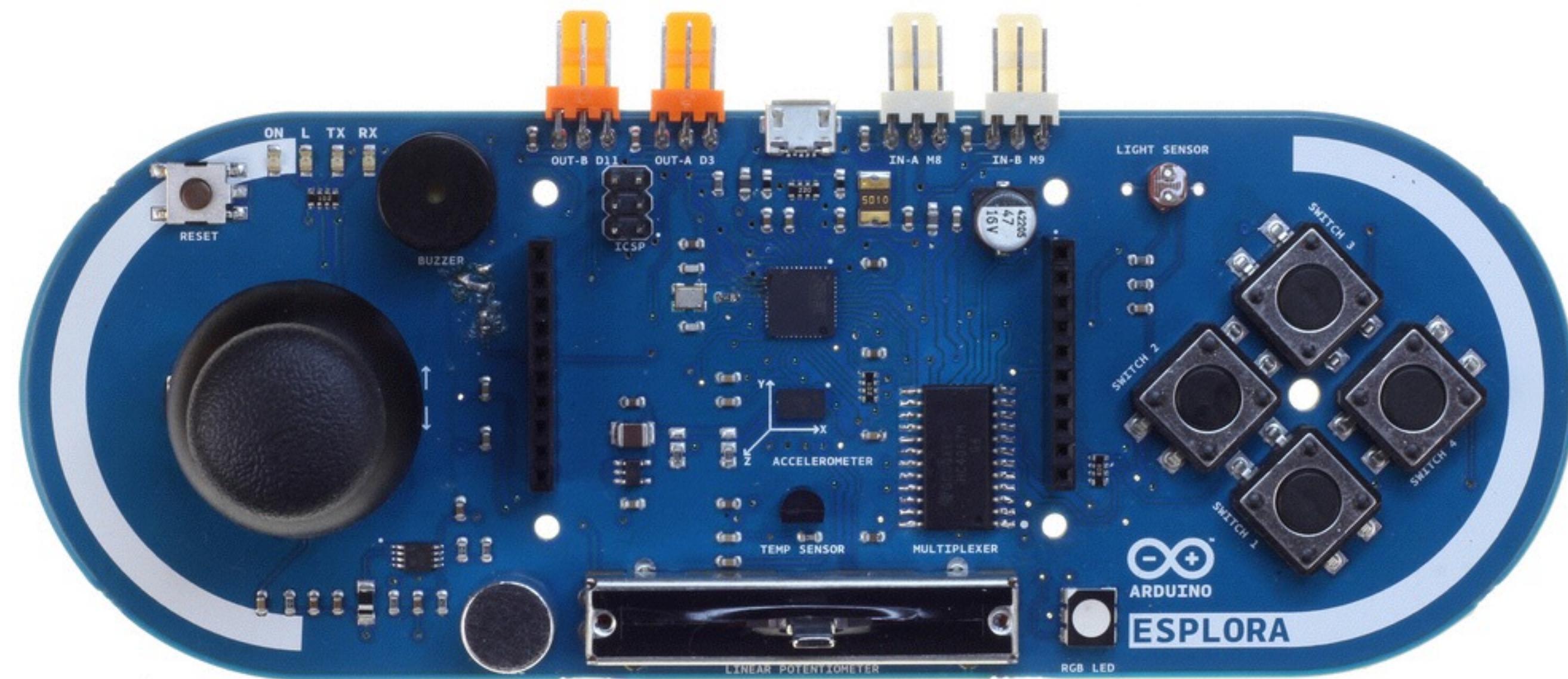
**The Arduino language is
based on C/C++.**

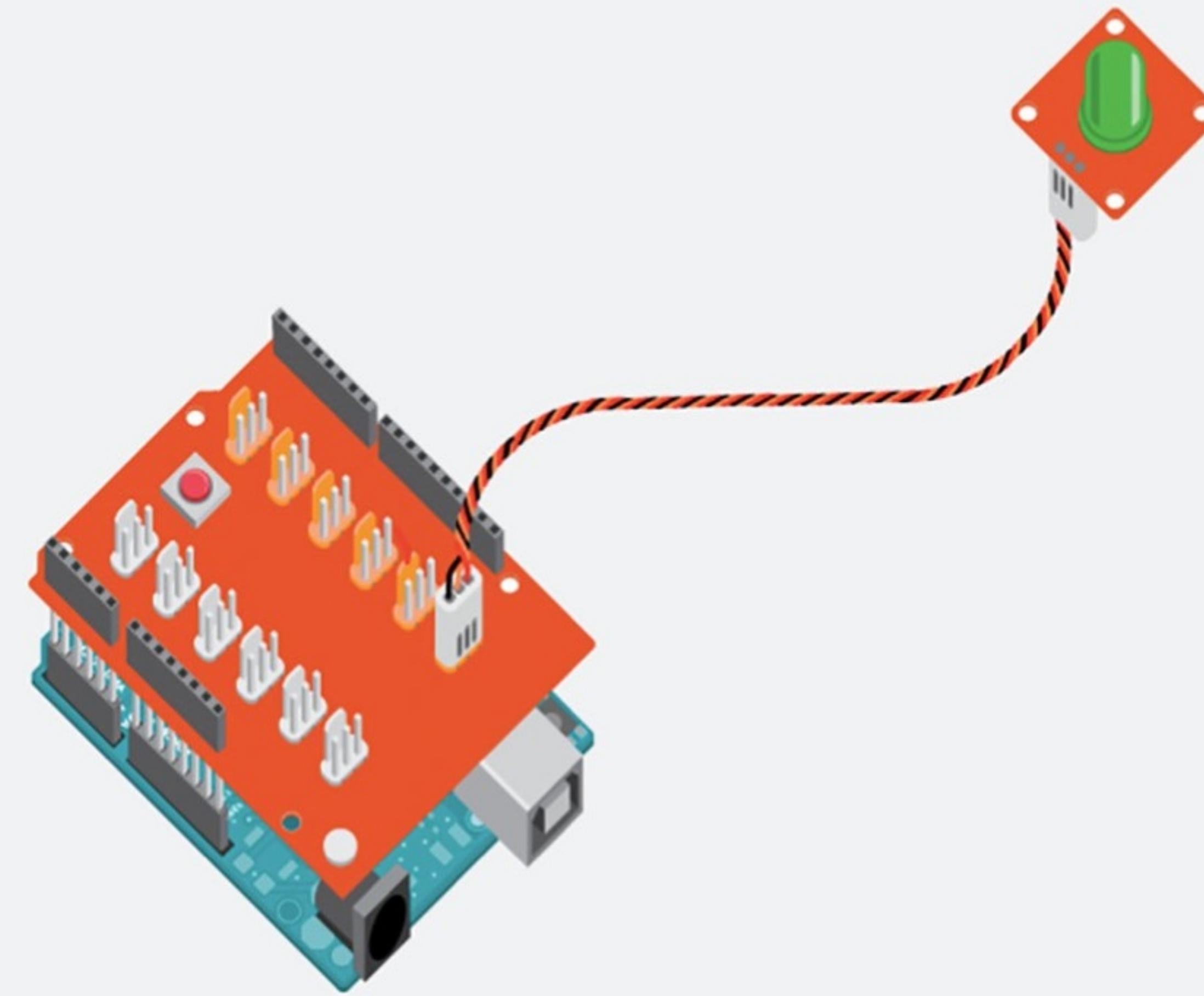
It links against AVR Libc and allows the use of any of its functions.

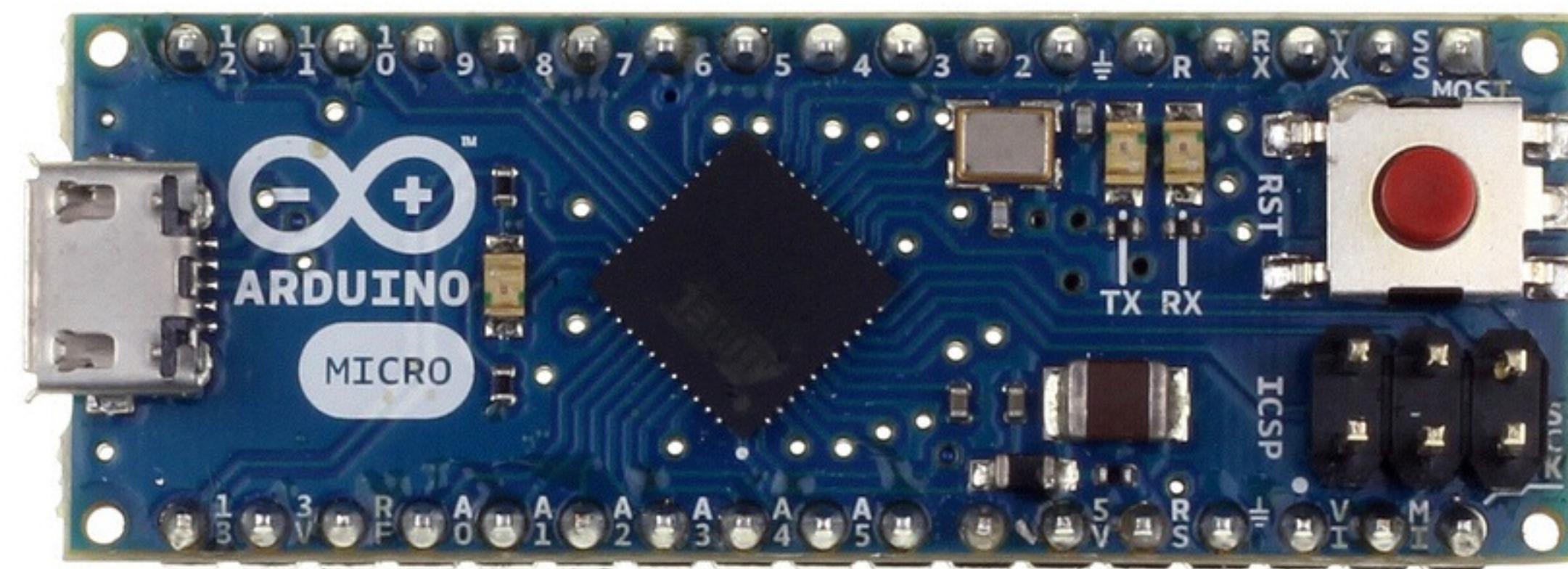


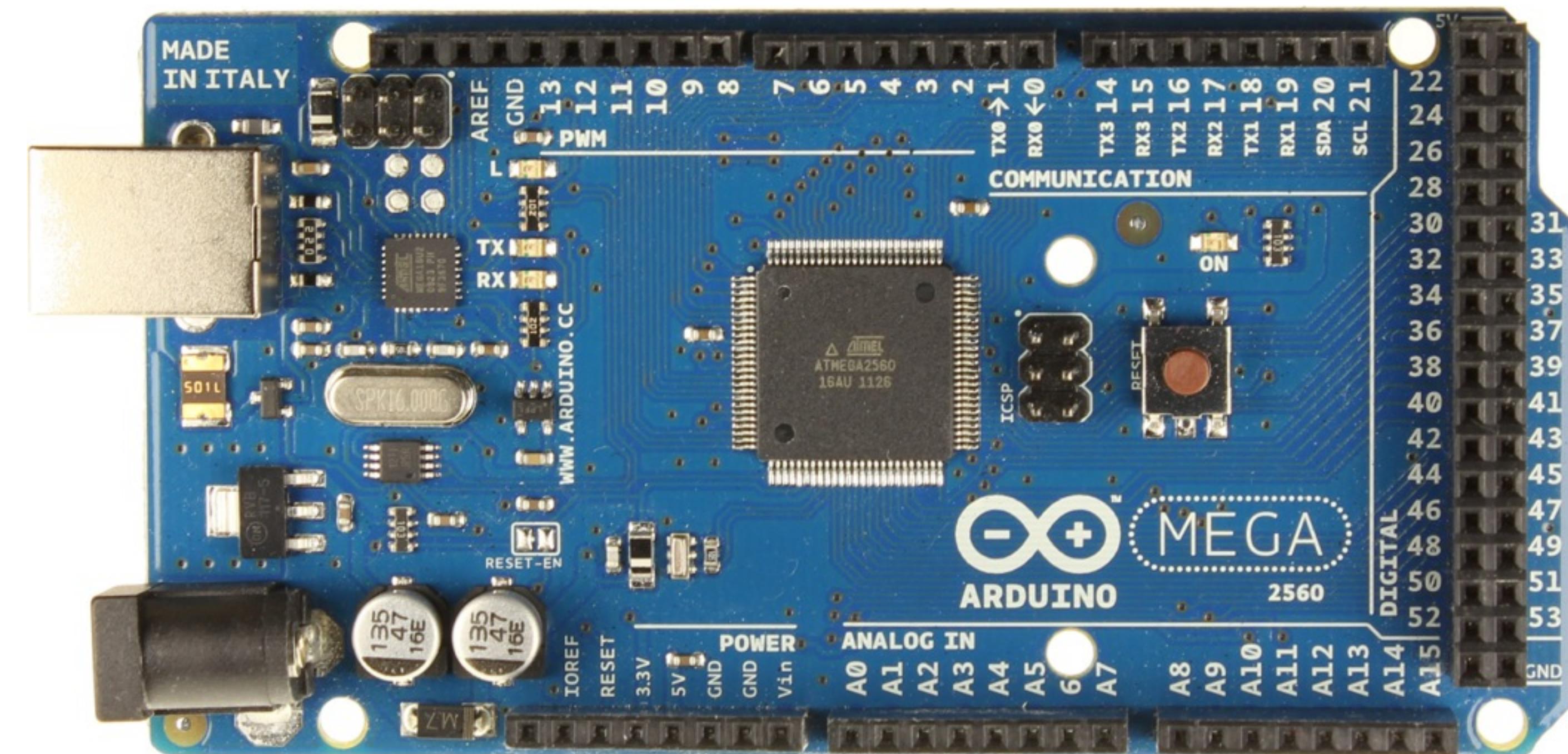


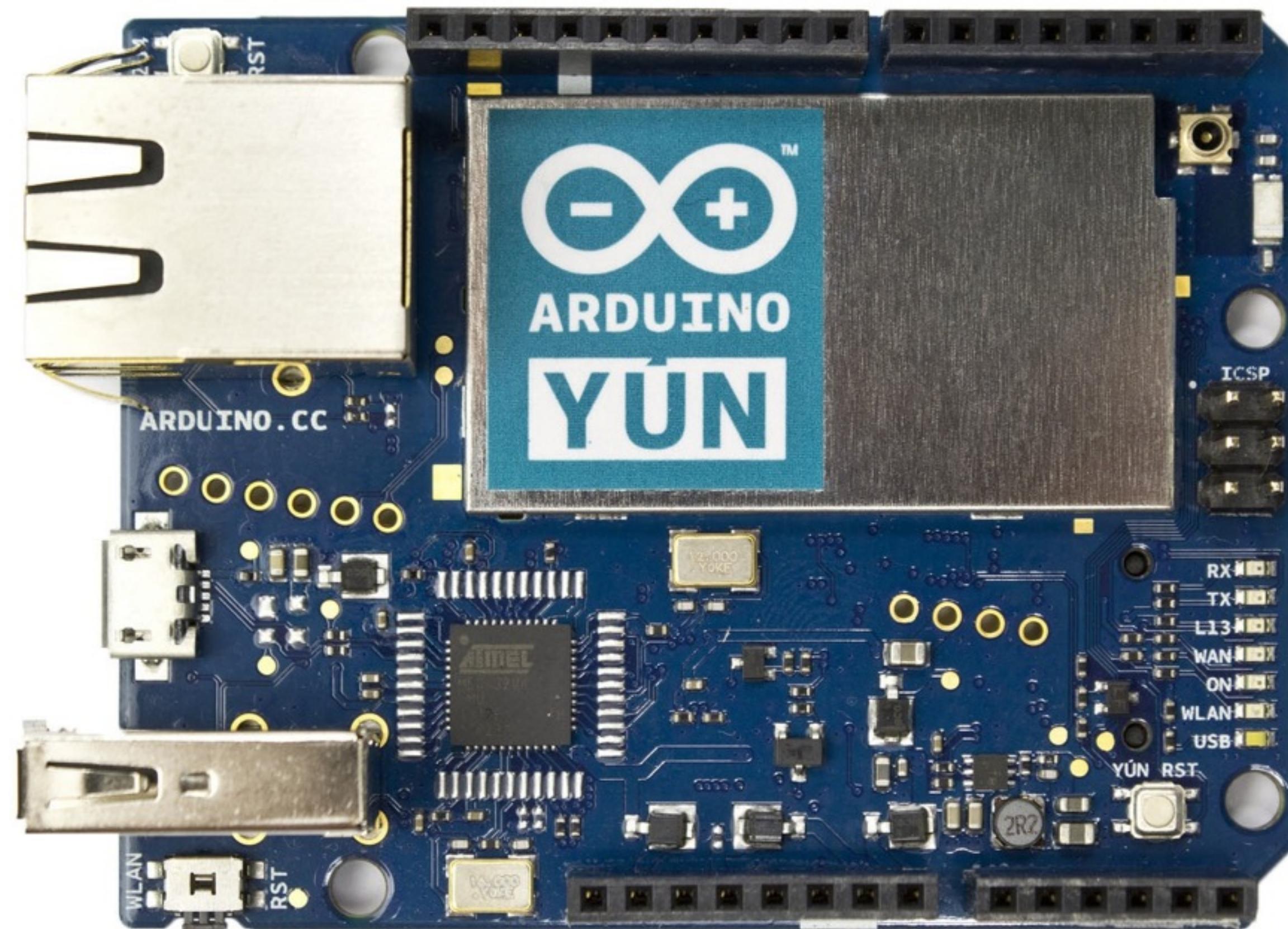


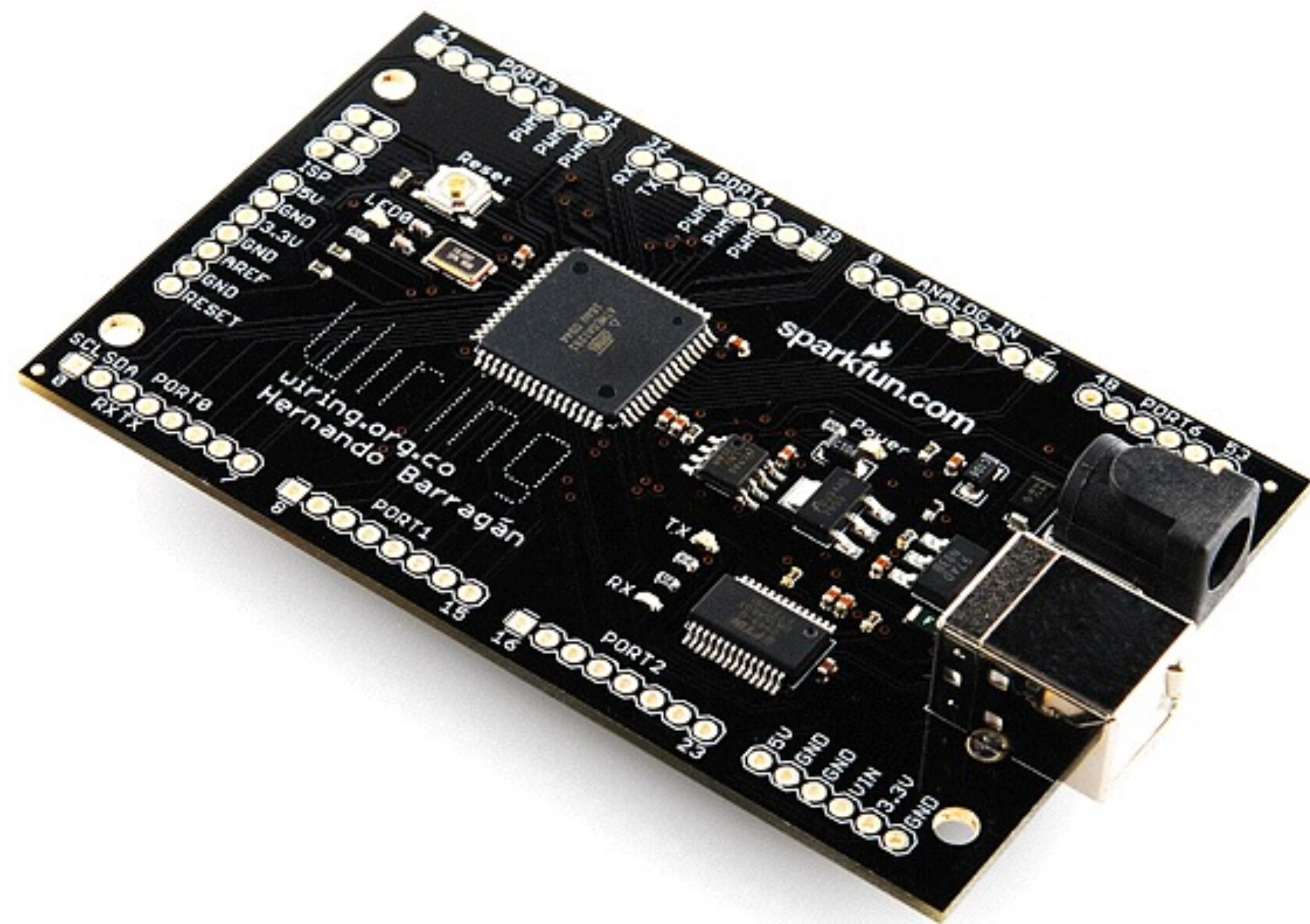


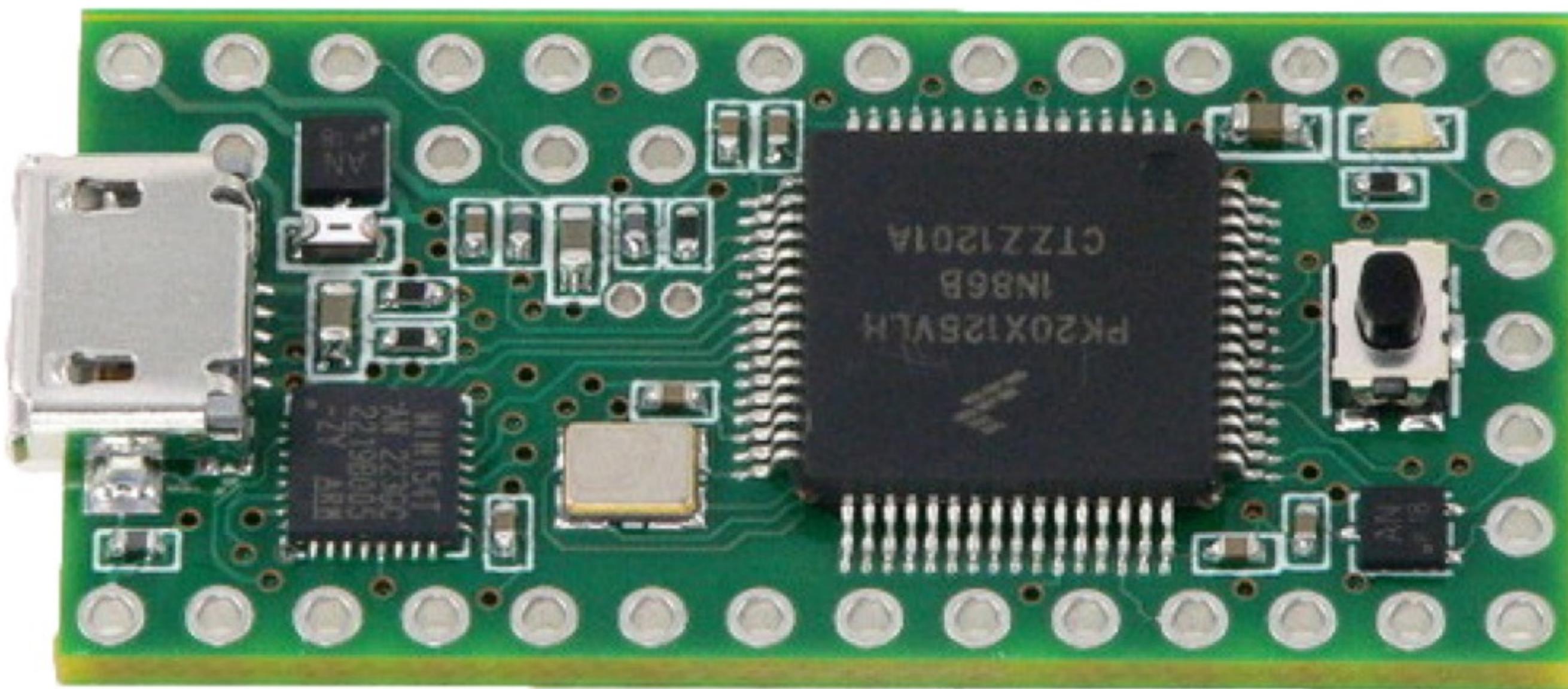


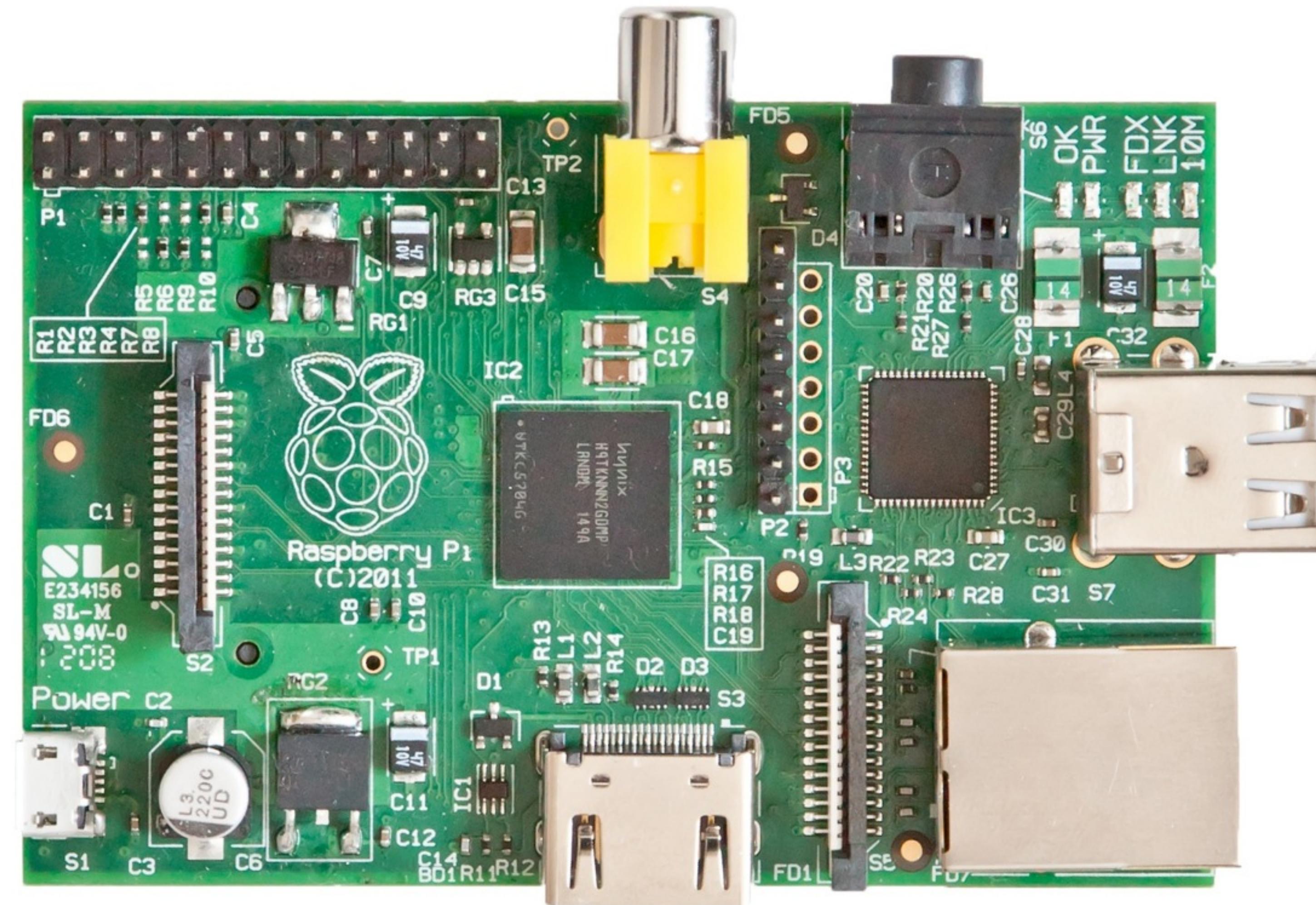






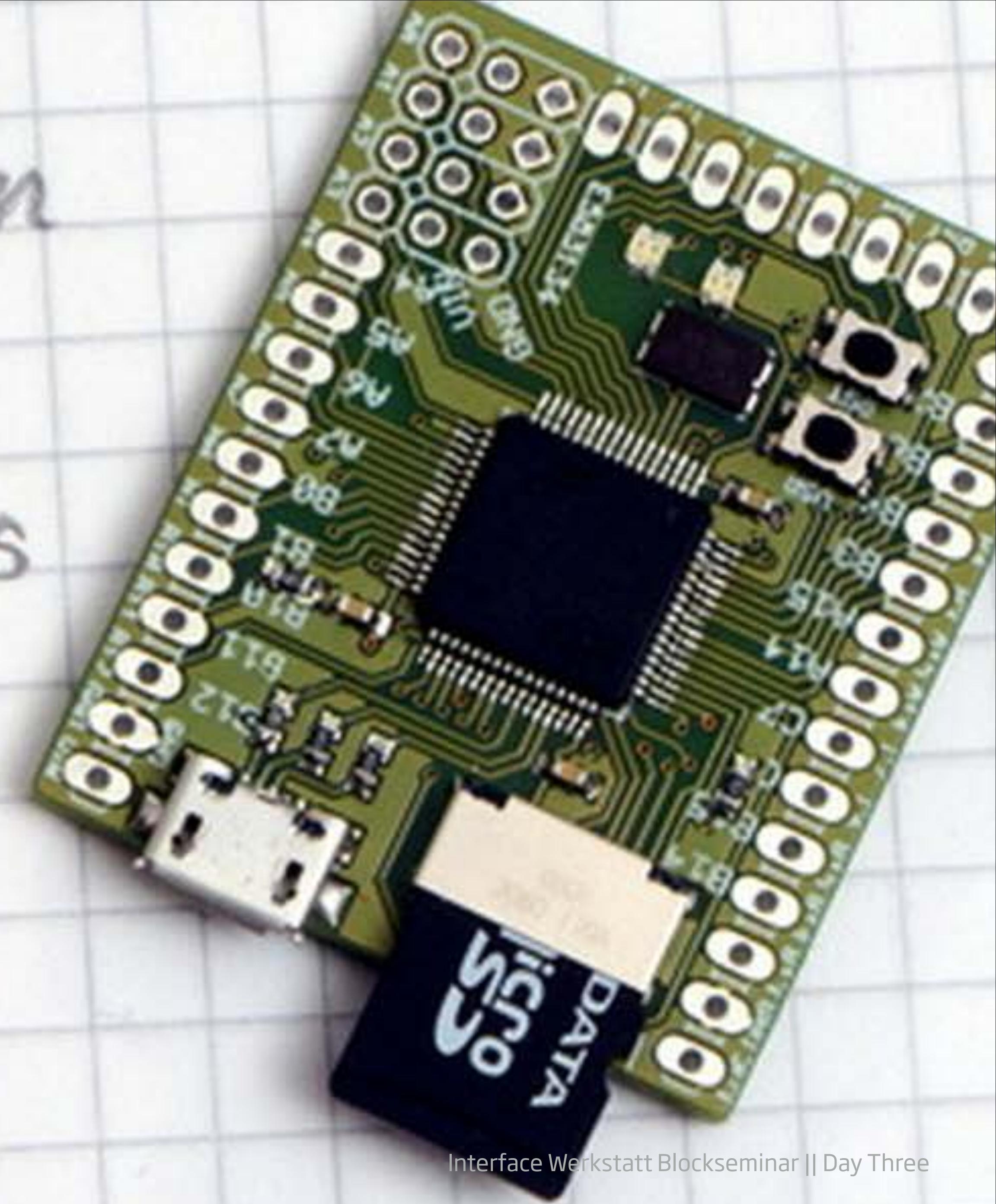


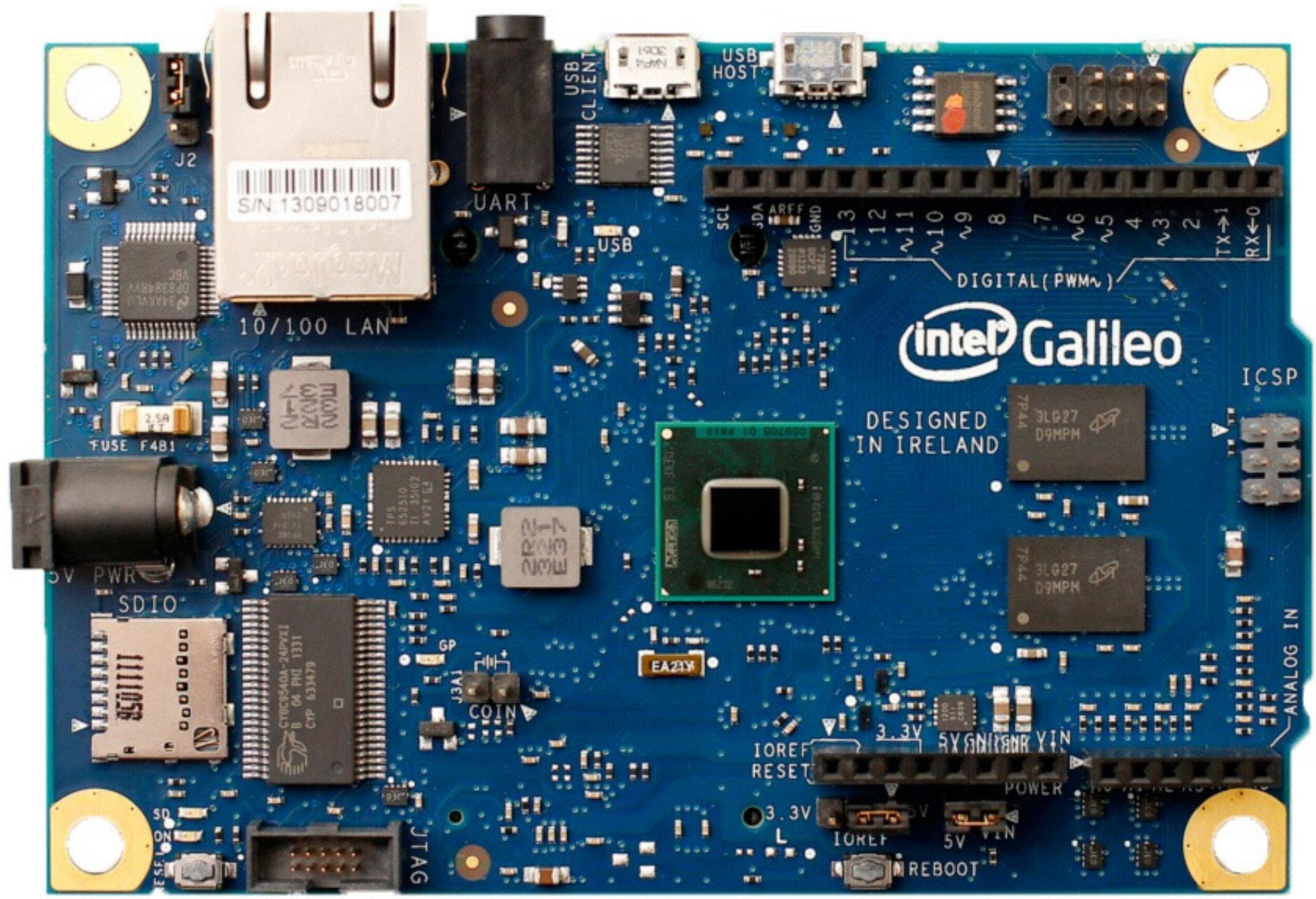


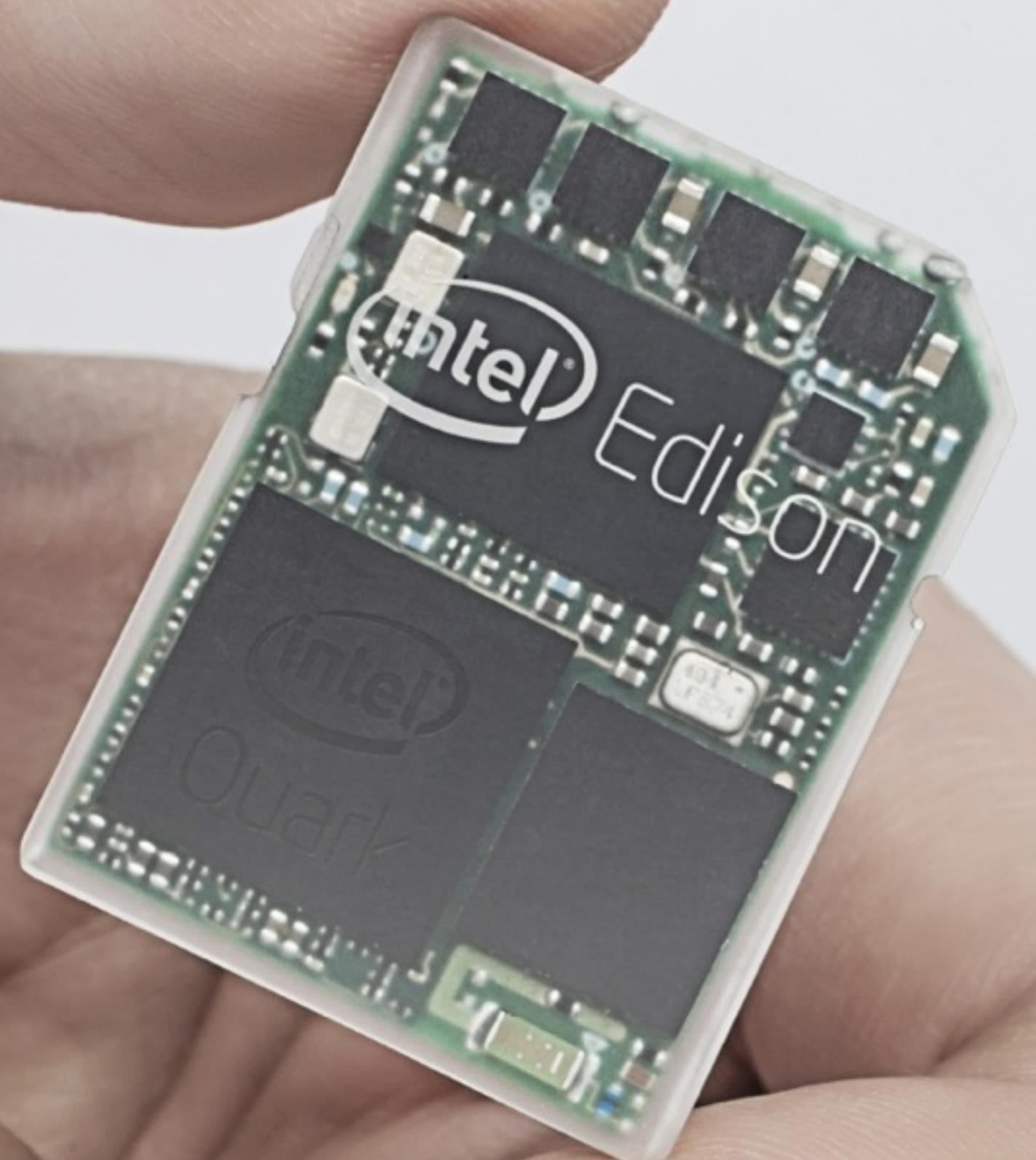


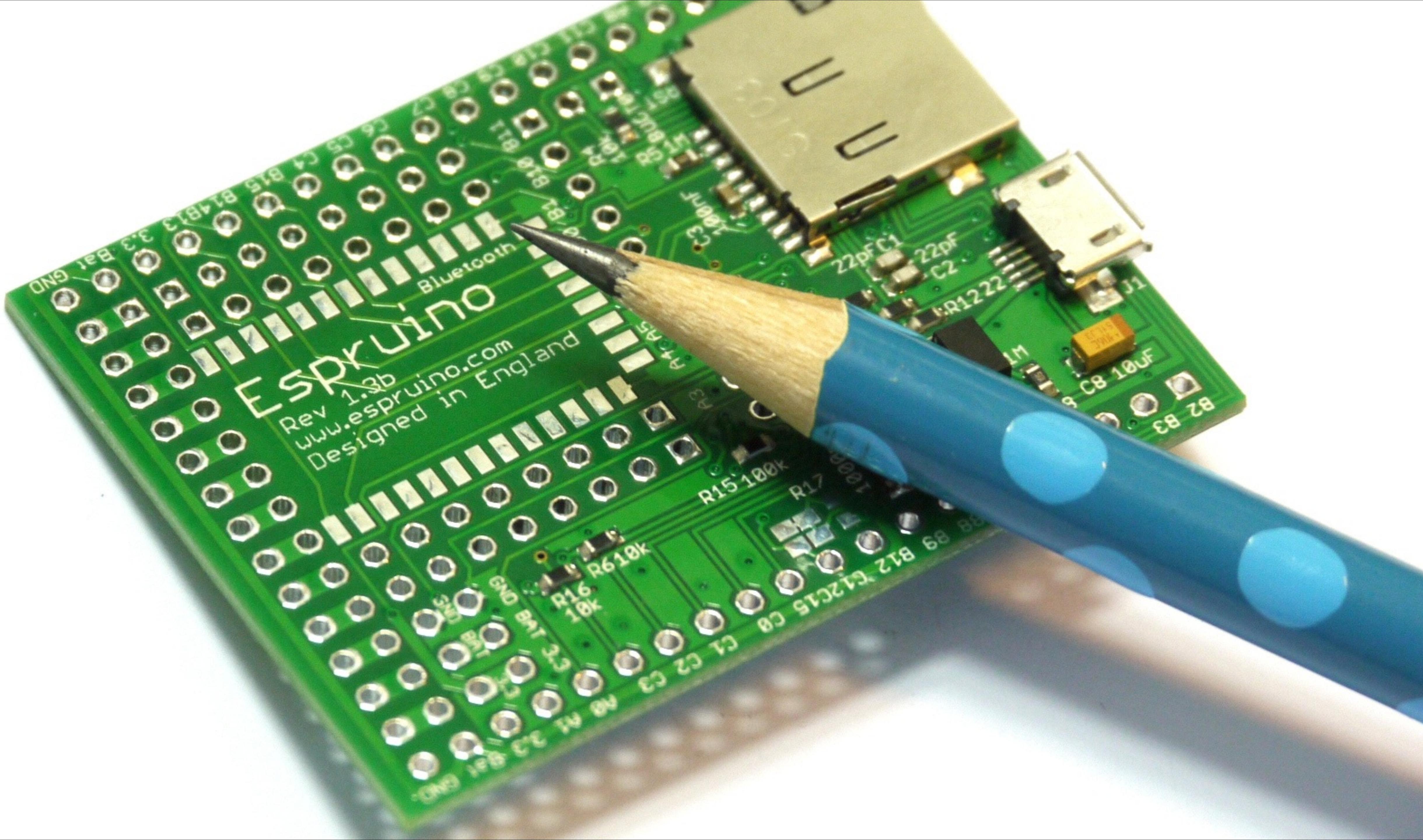


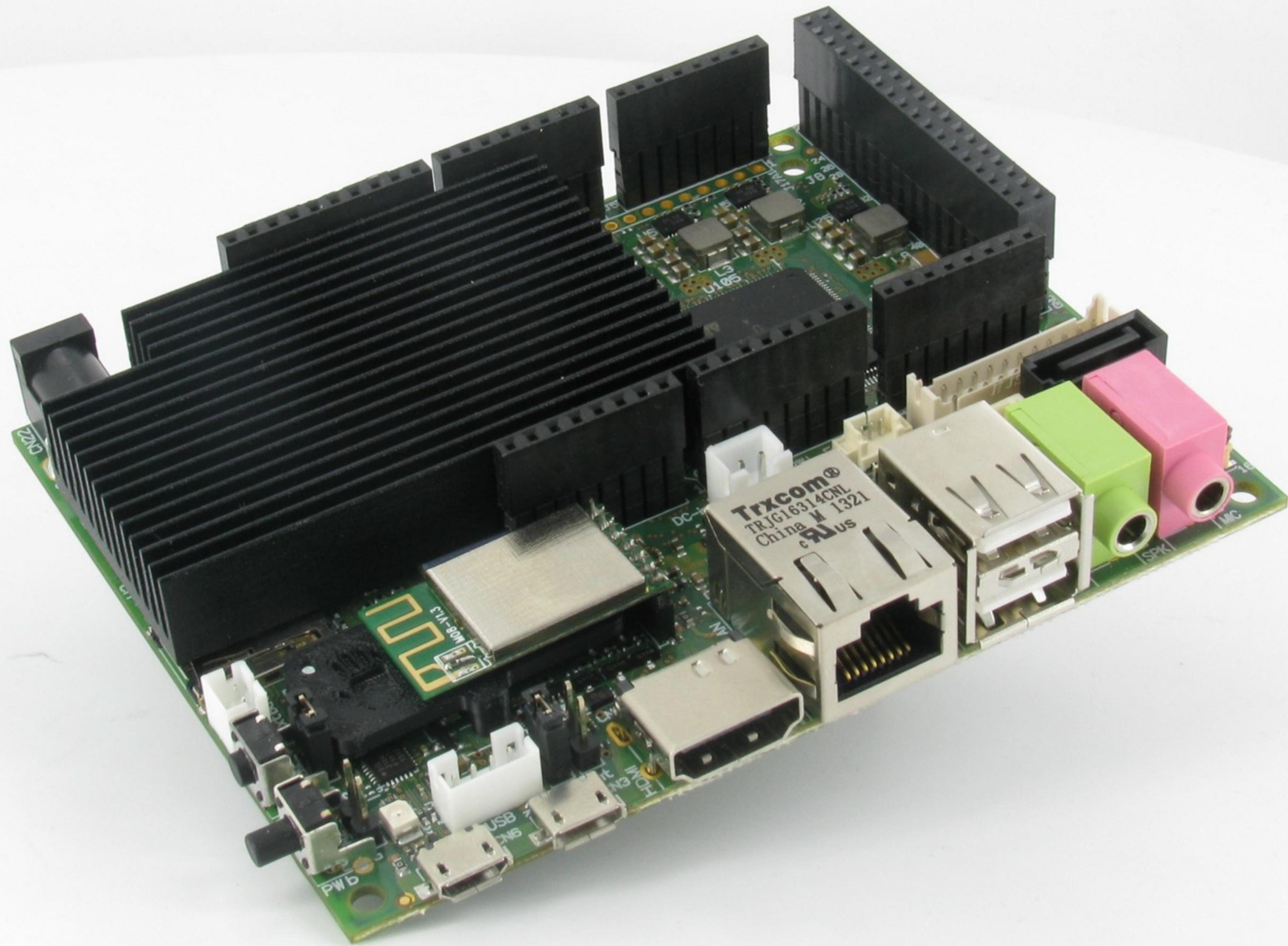
Micro Python
Python for
microcontrollers

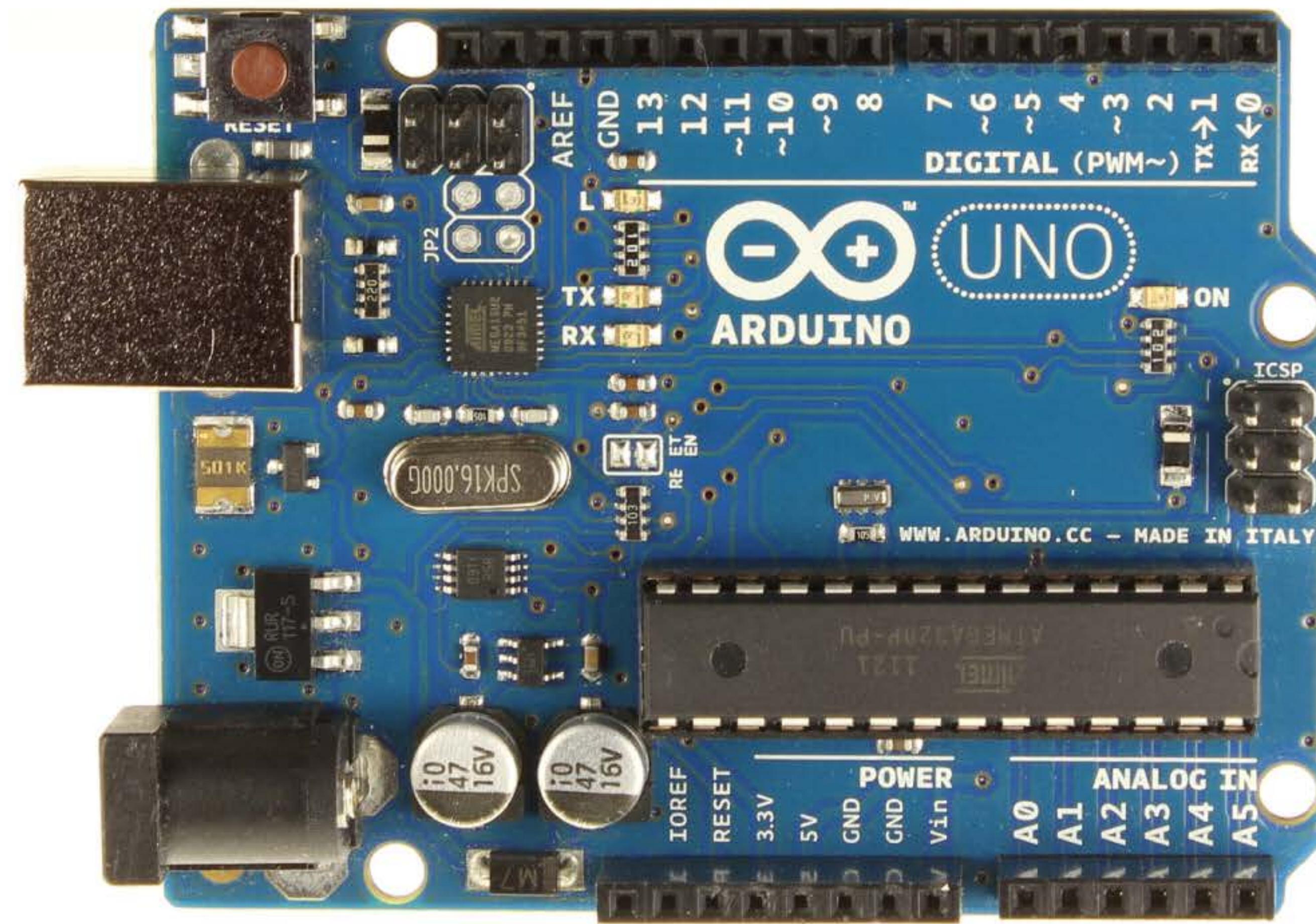


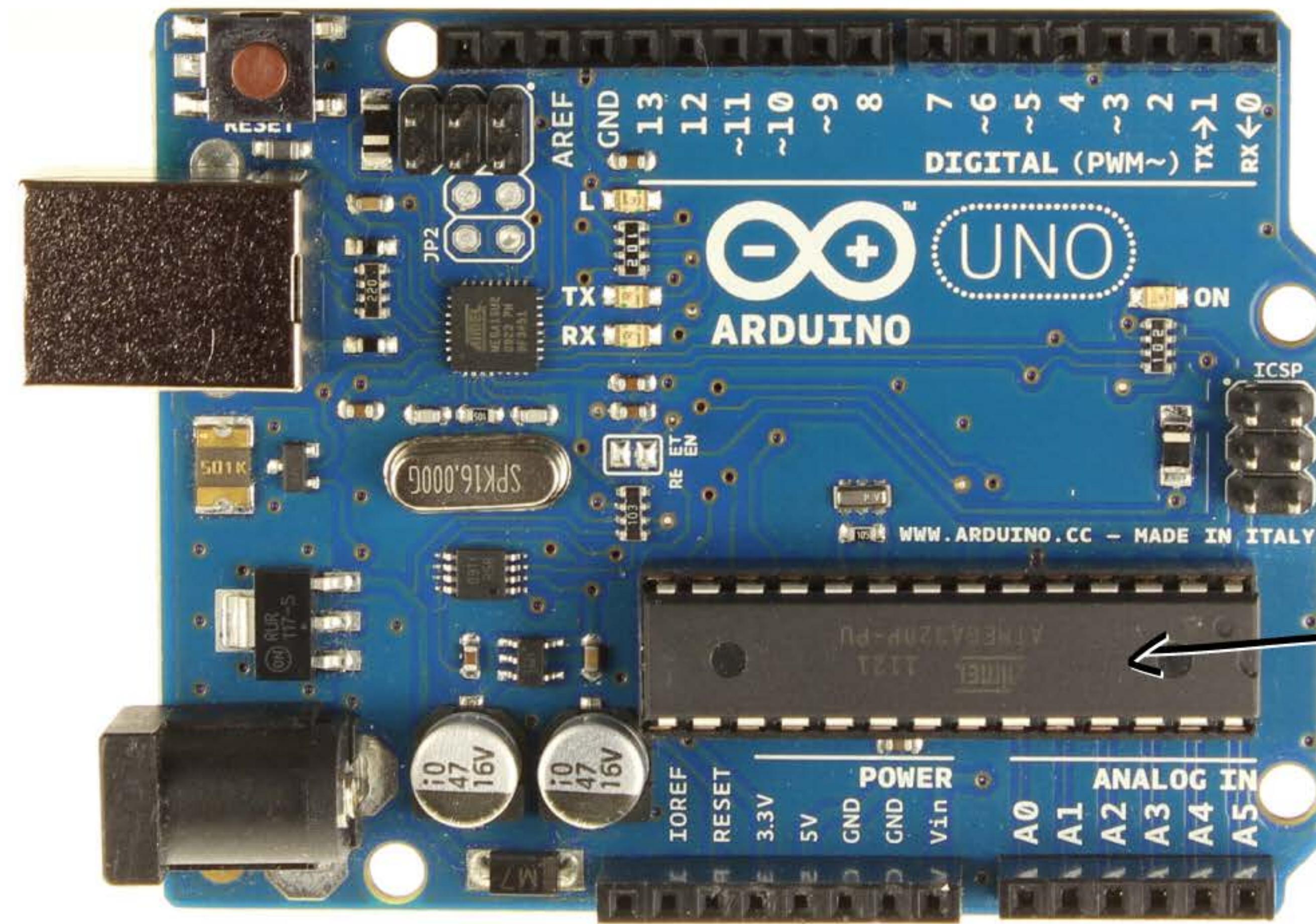




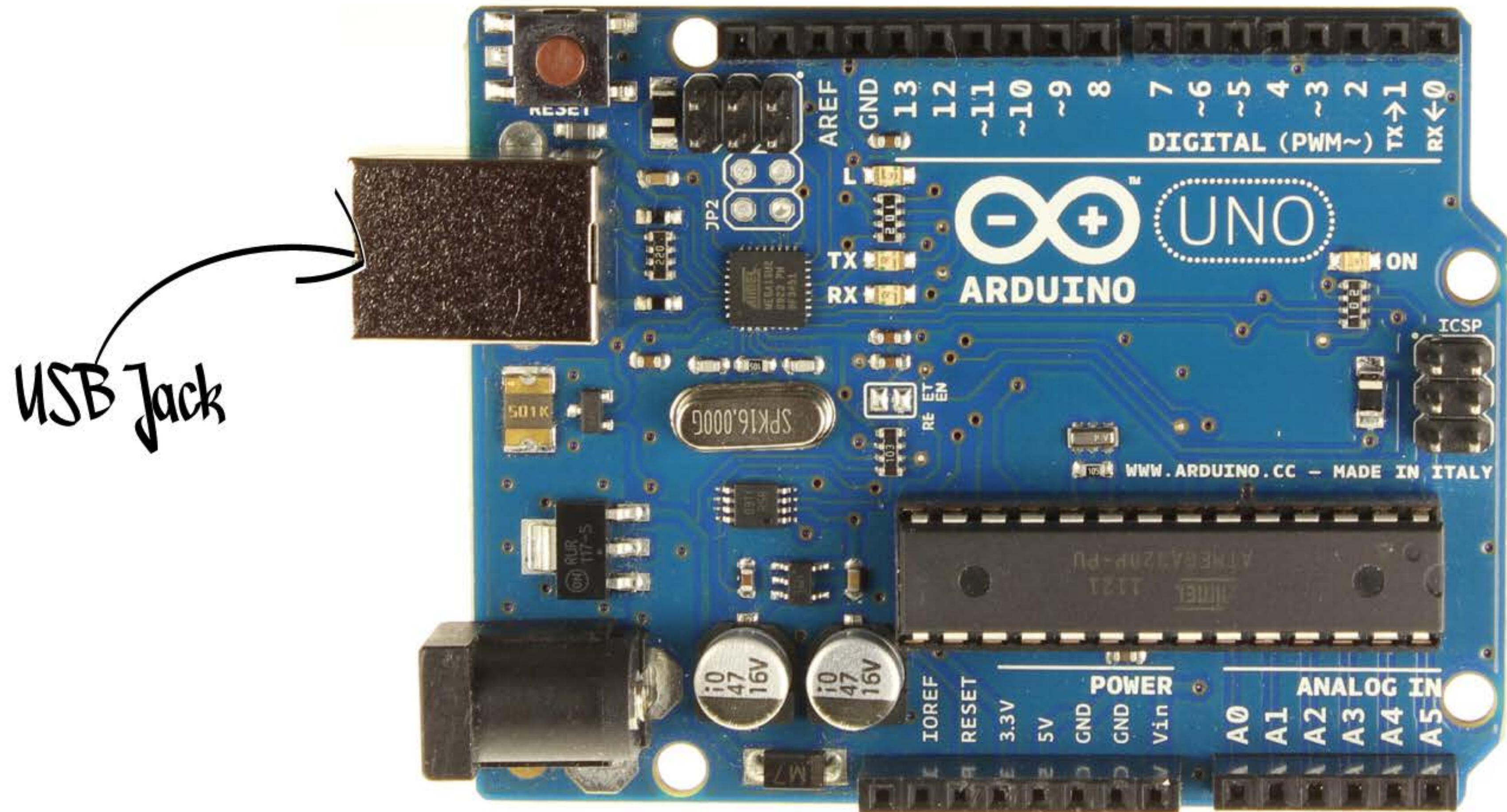


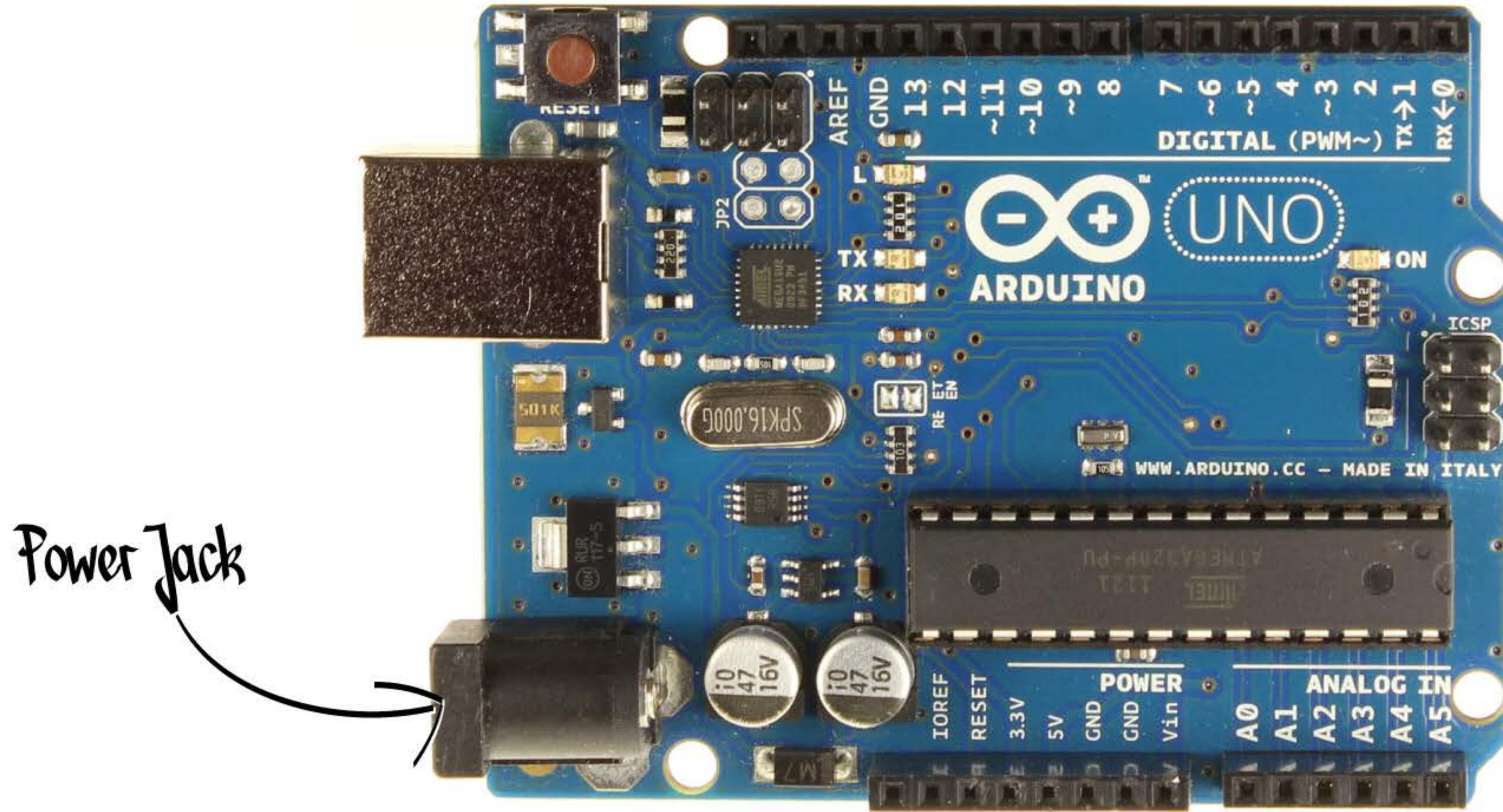




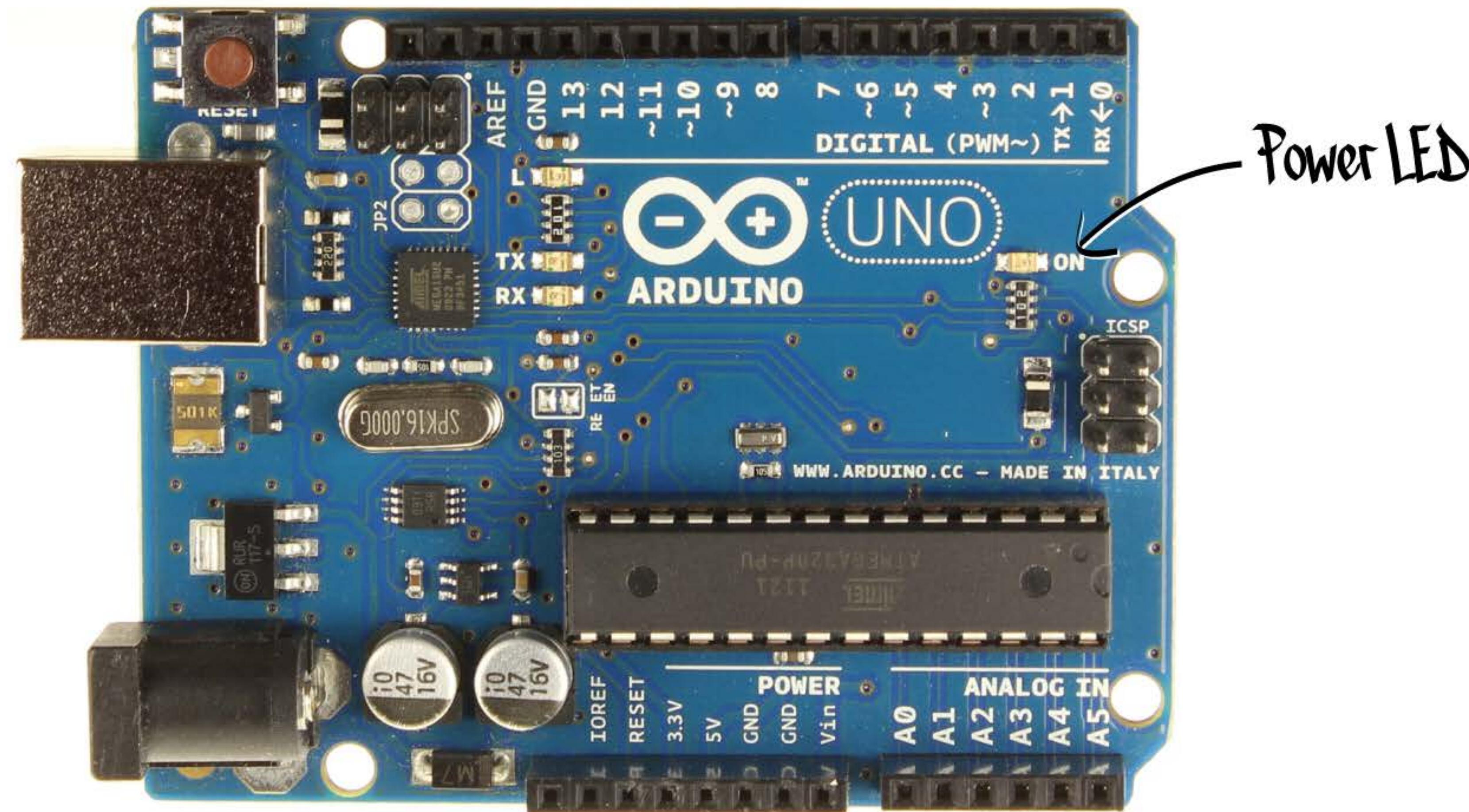


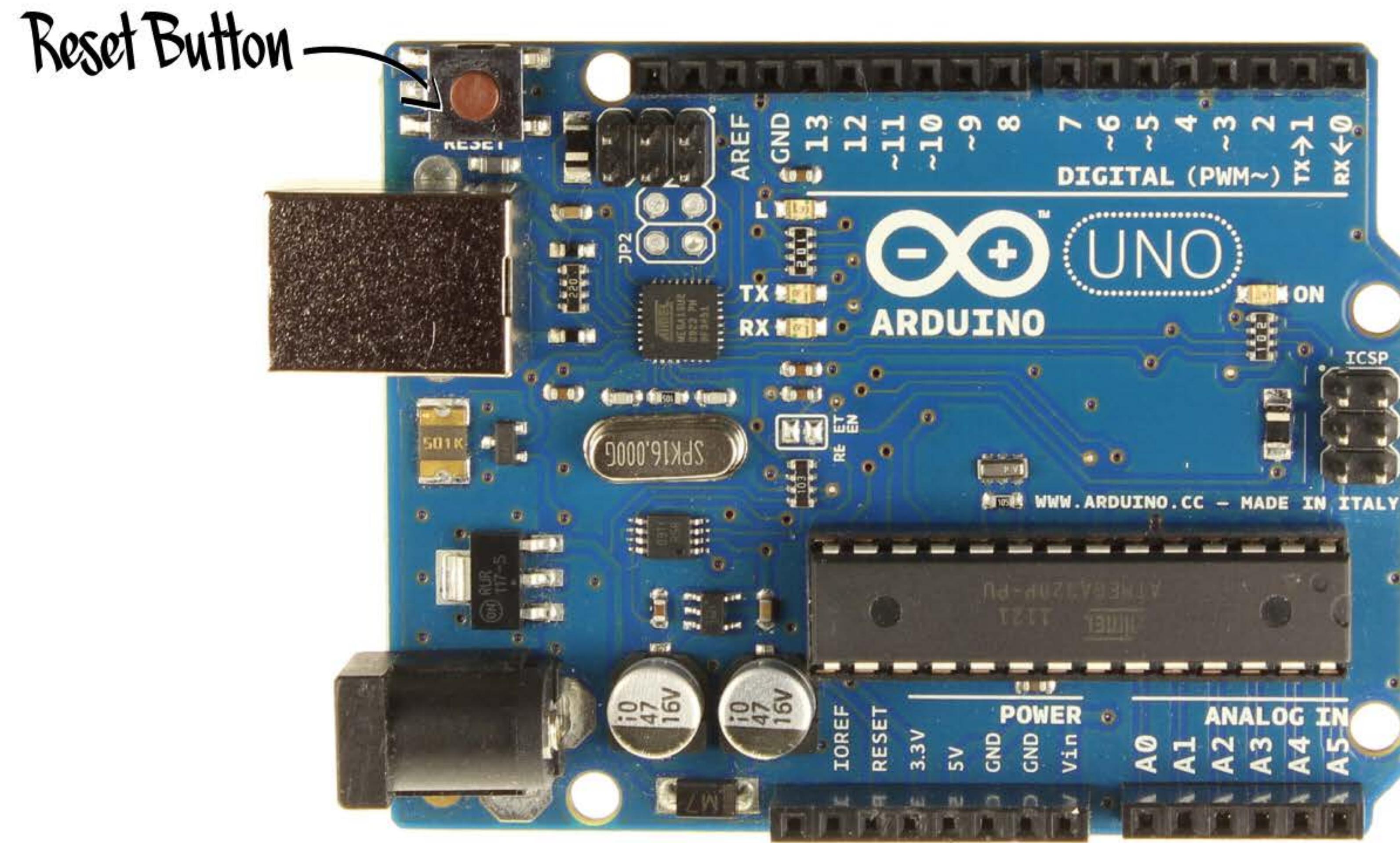
Microcontroller



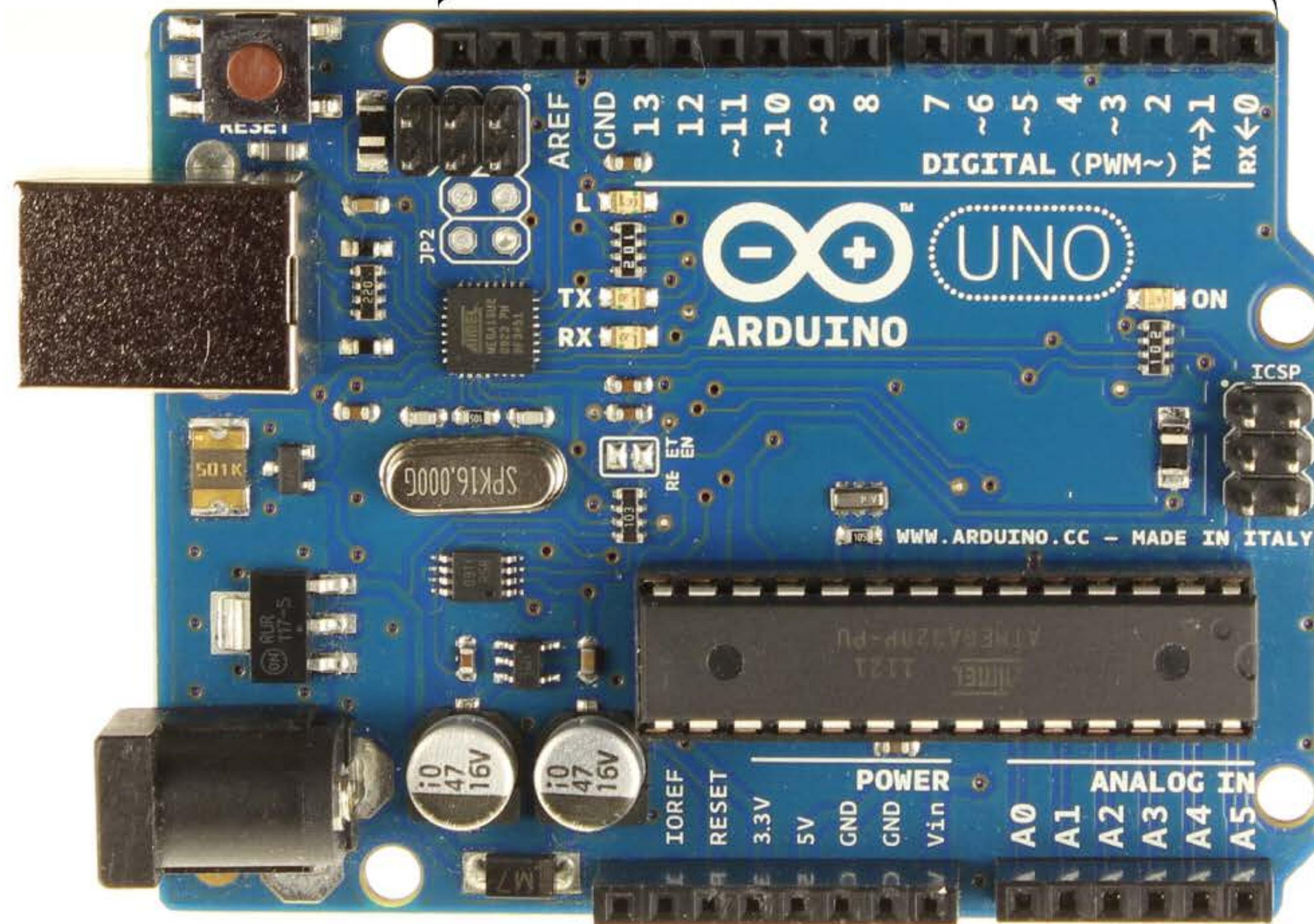




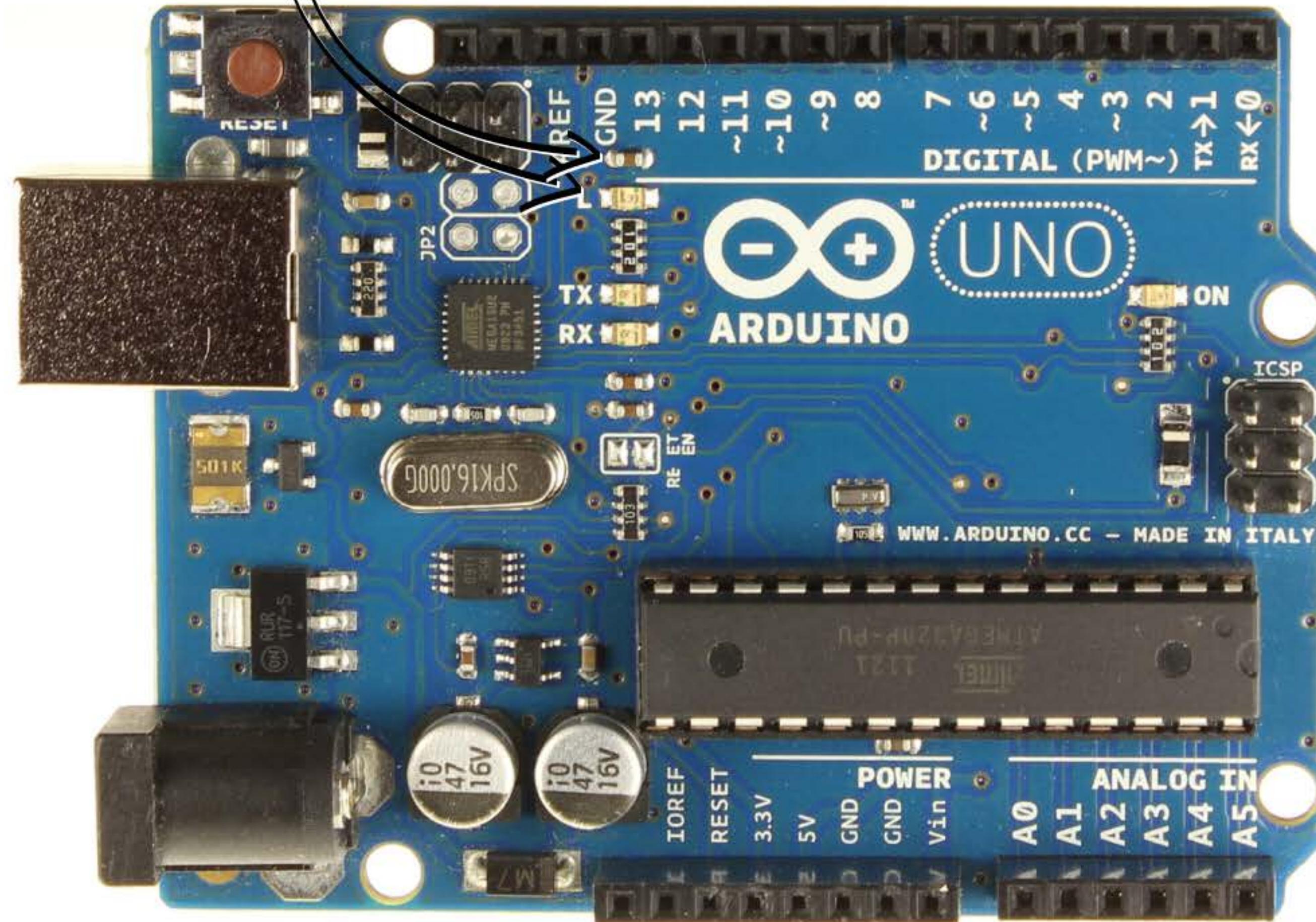


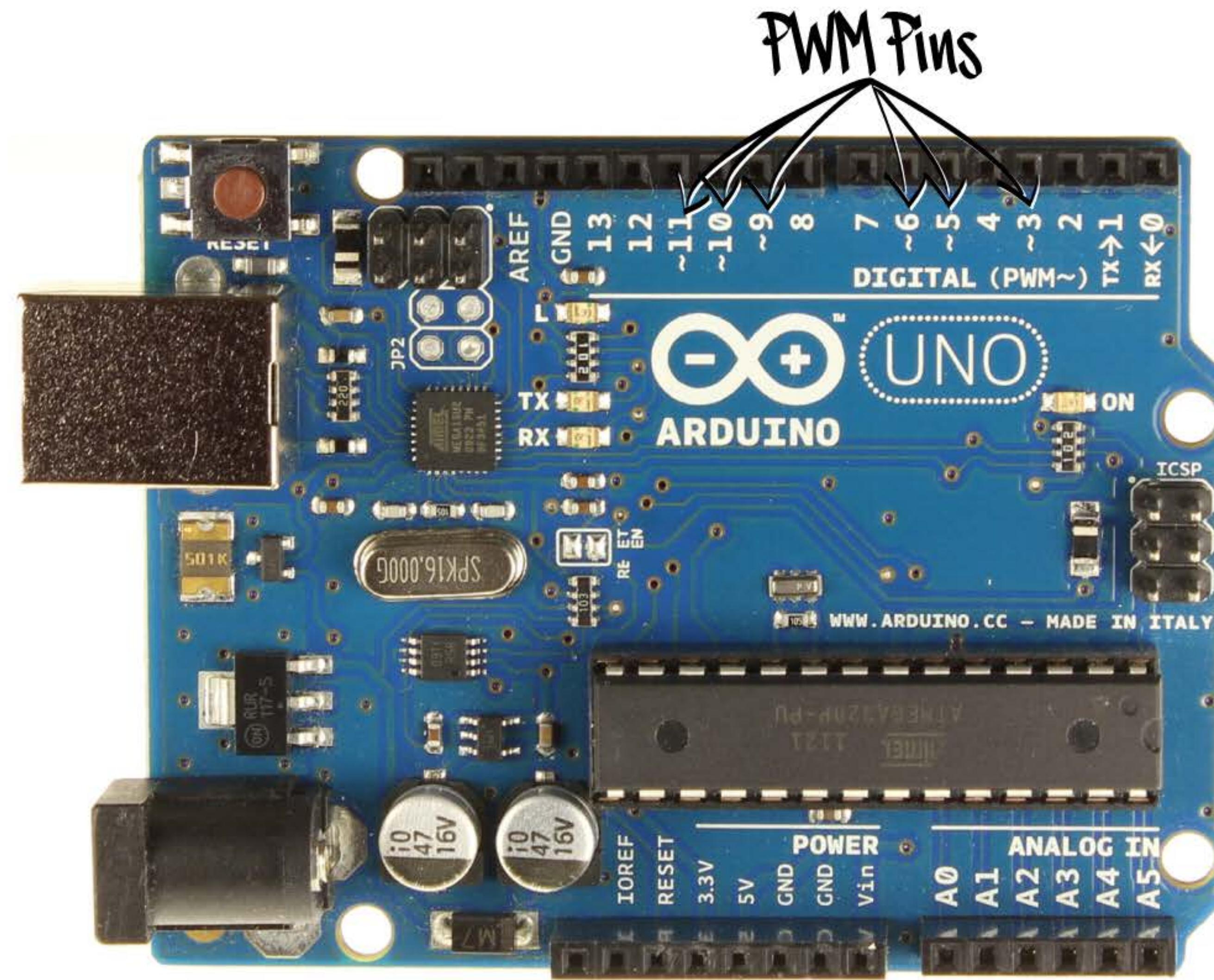


Digital Pins



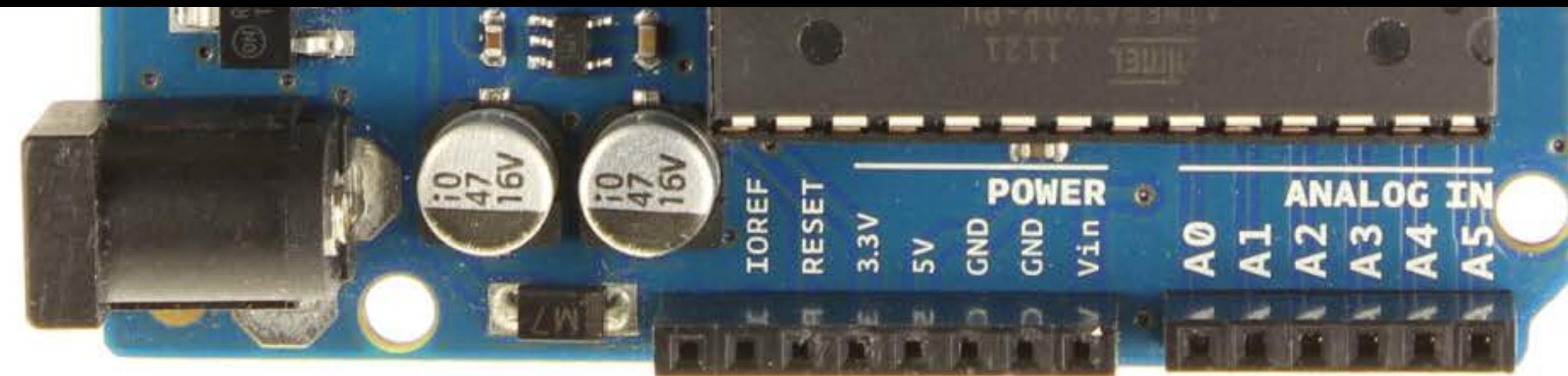
Pin 13(L) LED
Pin 13 Resistor

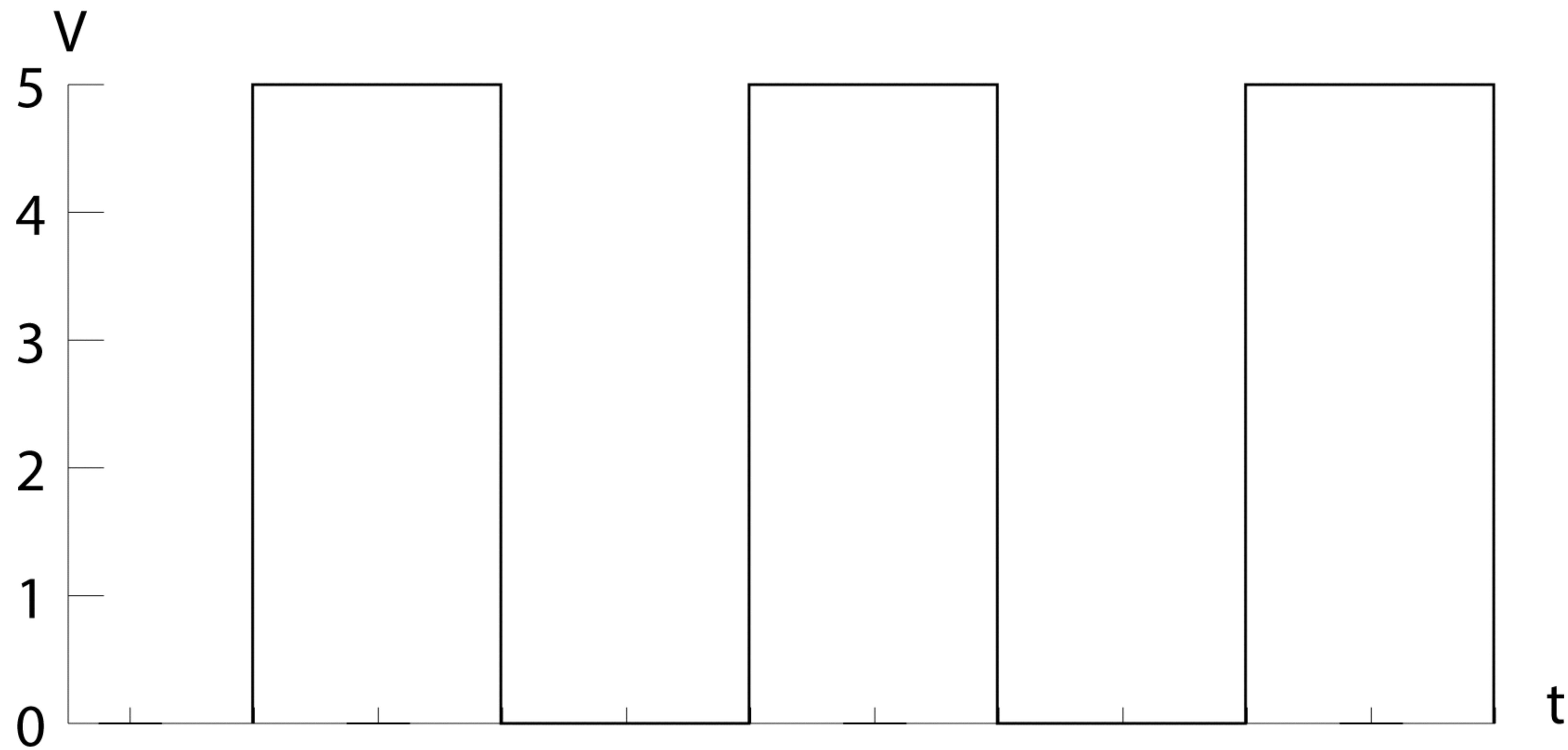


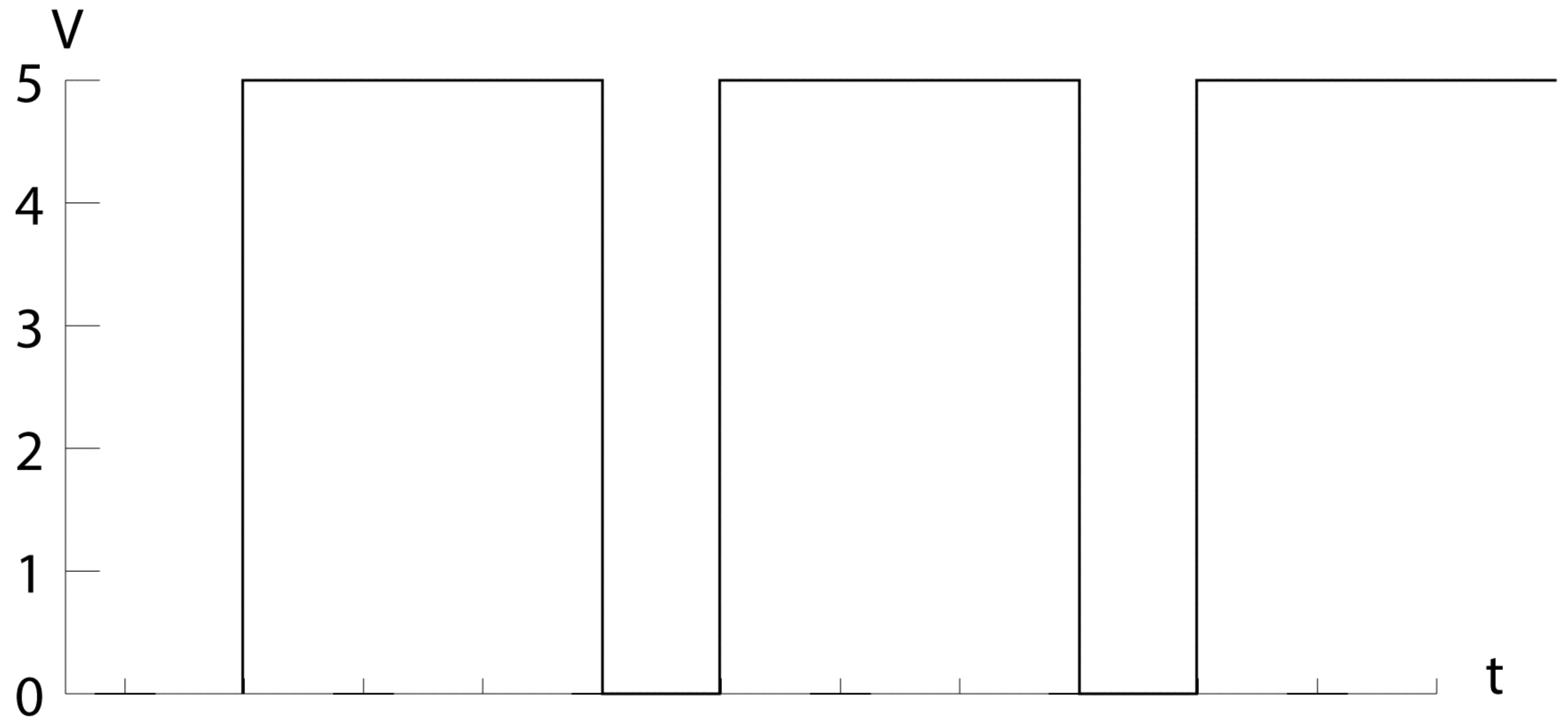


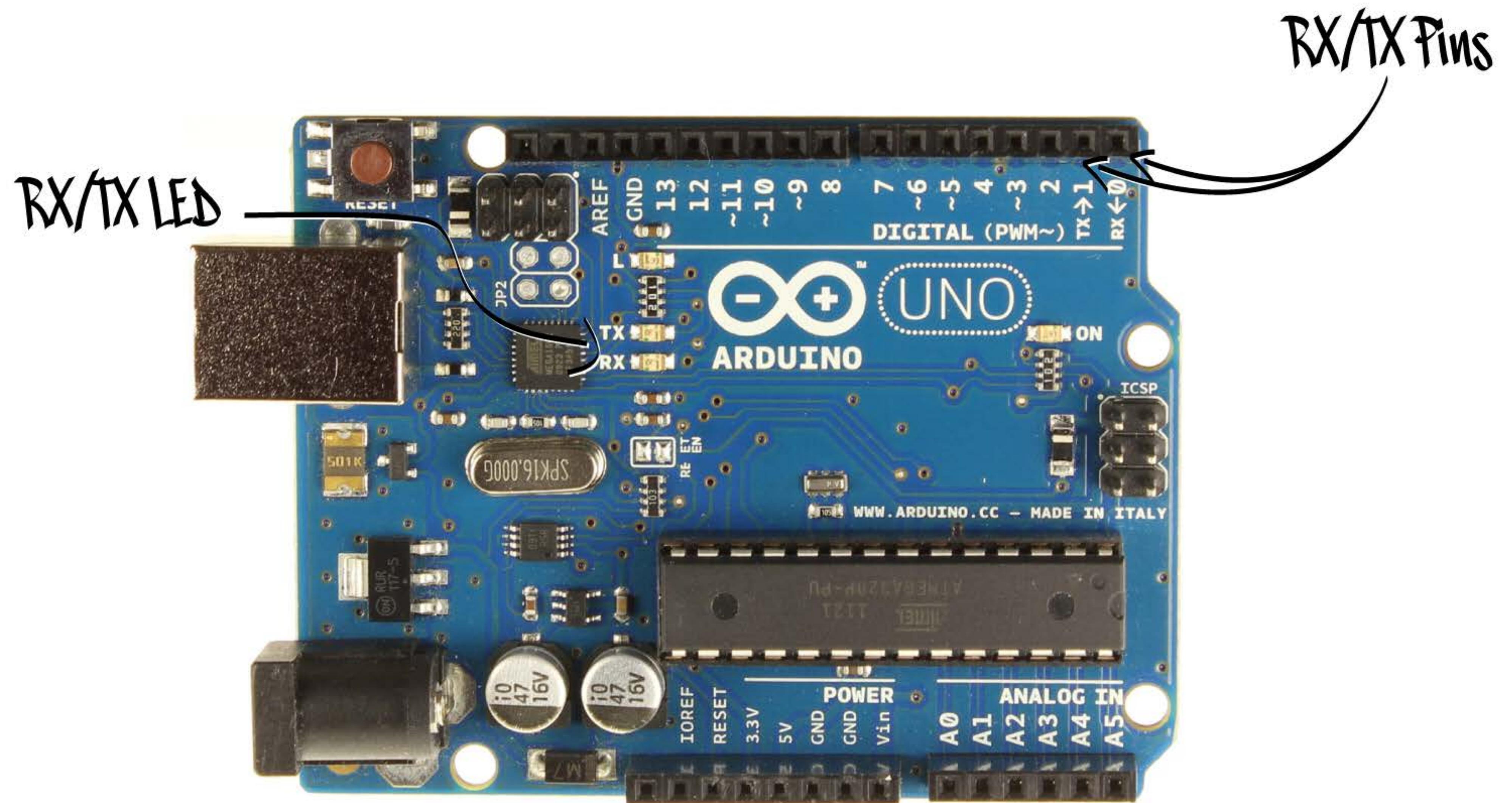


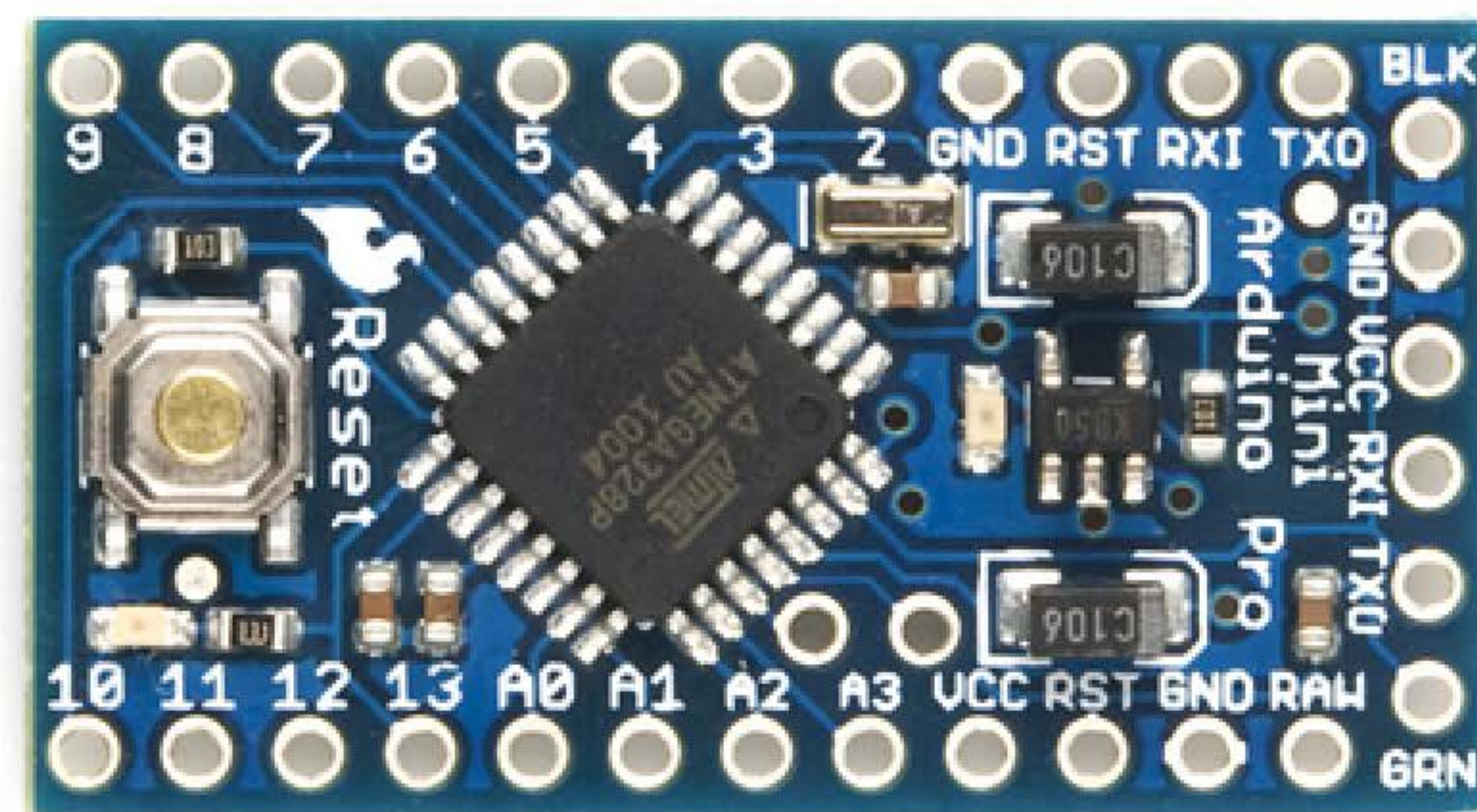
PWM ~ (Pulse Width Modulation)

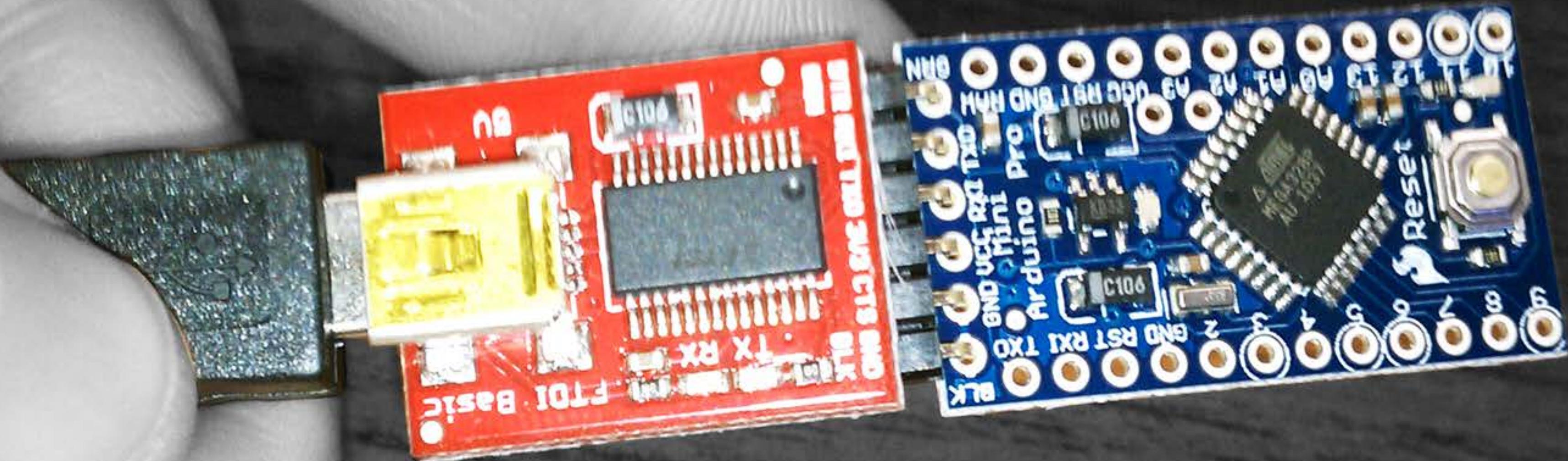


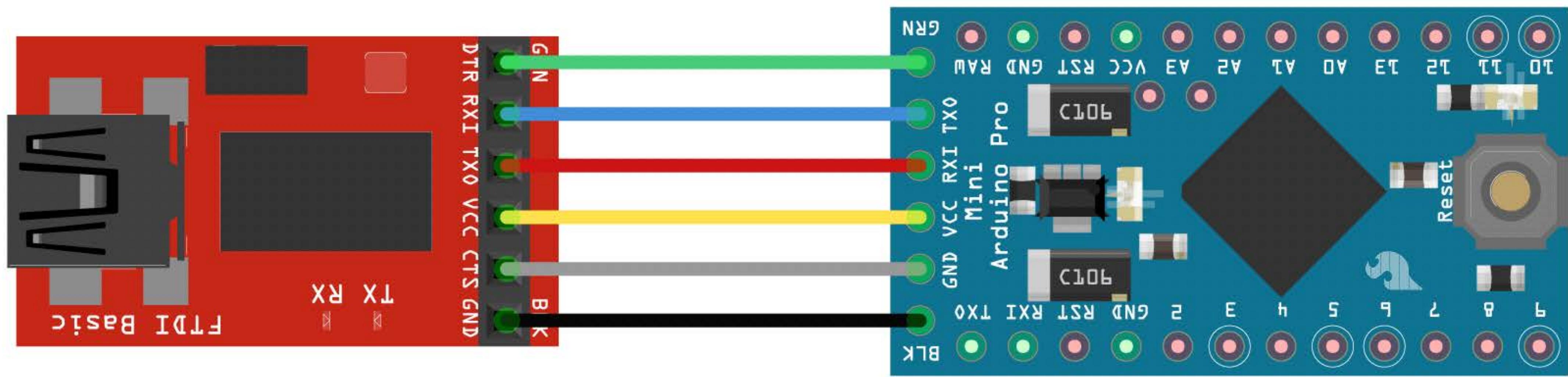


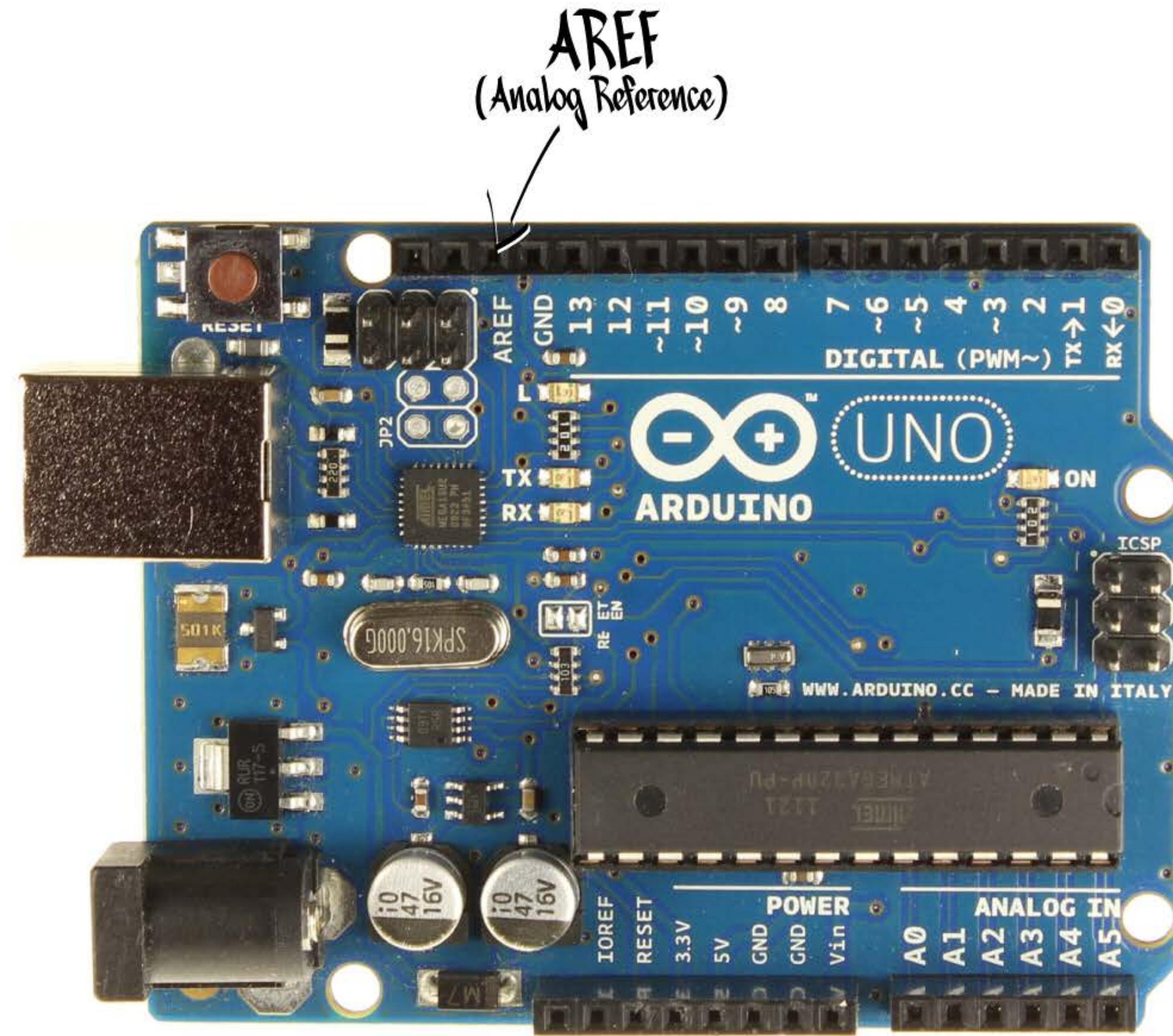


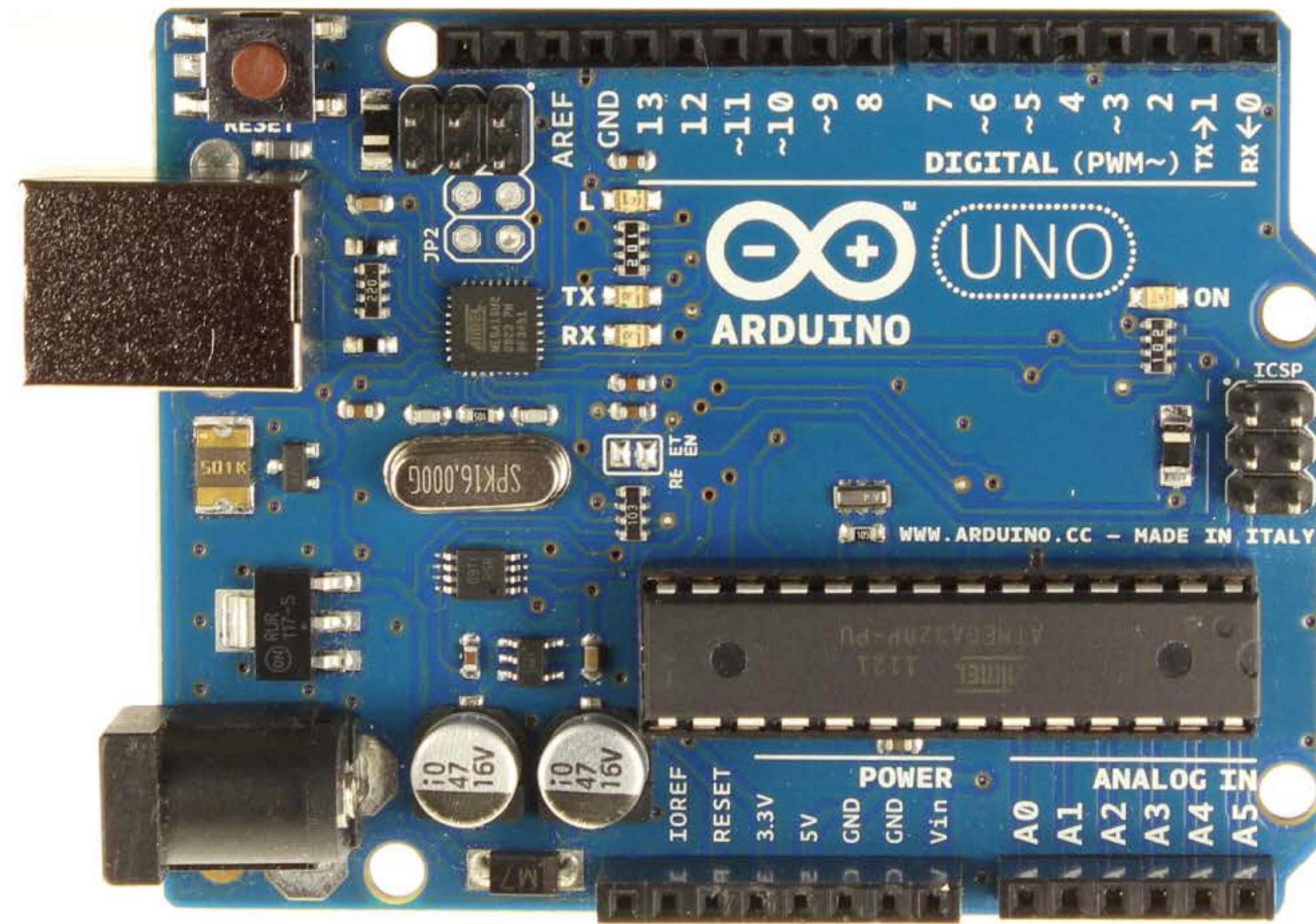




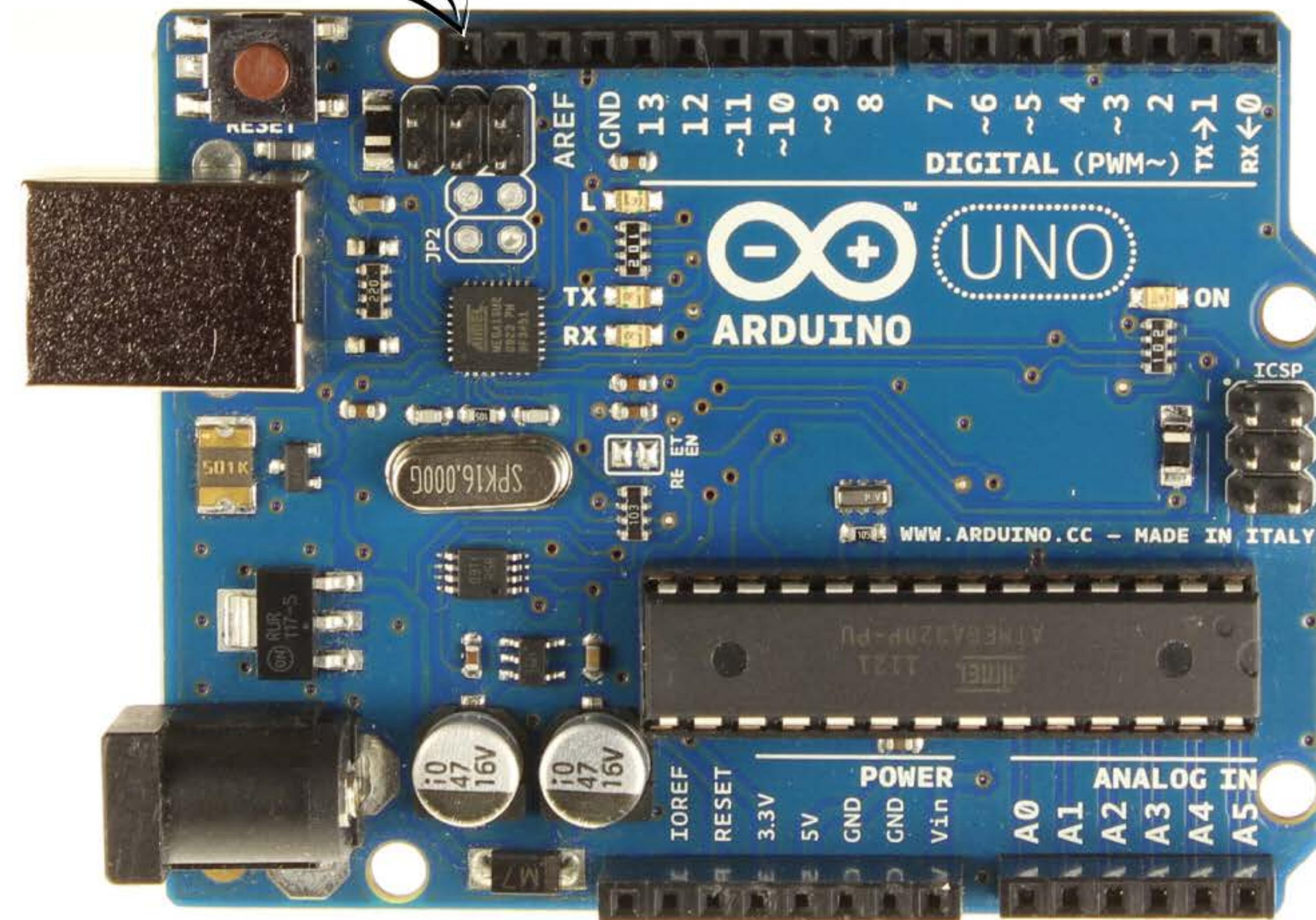




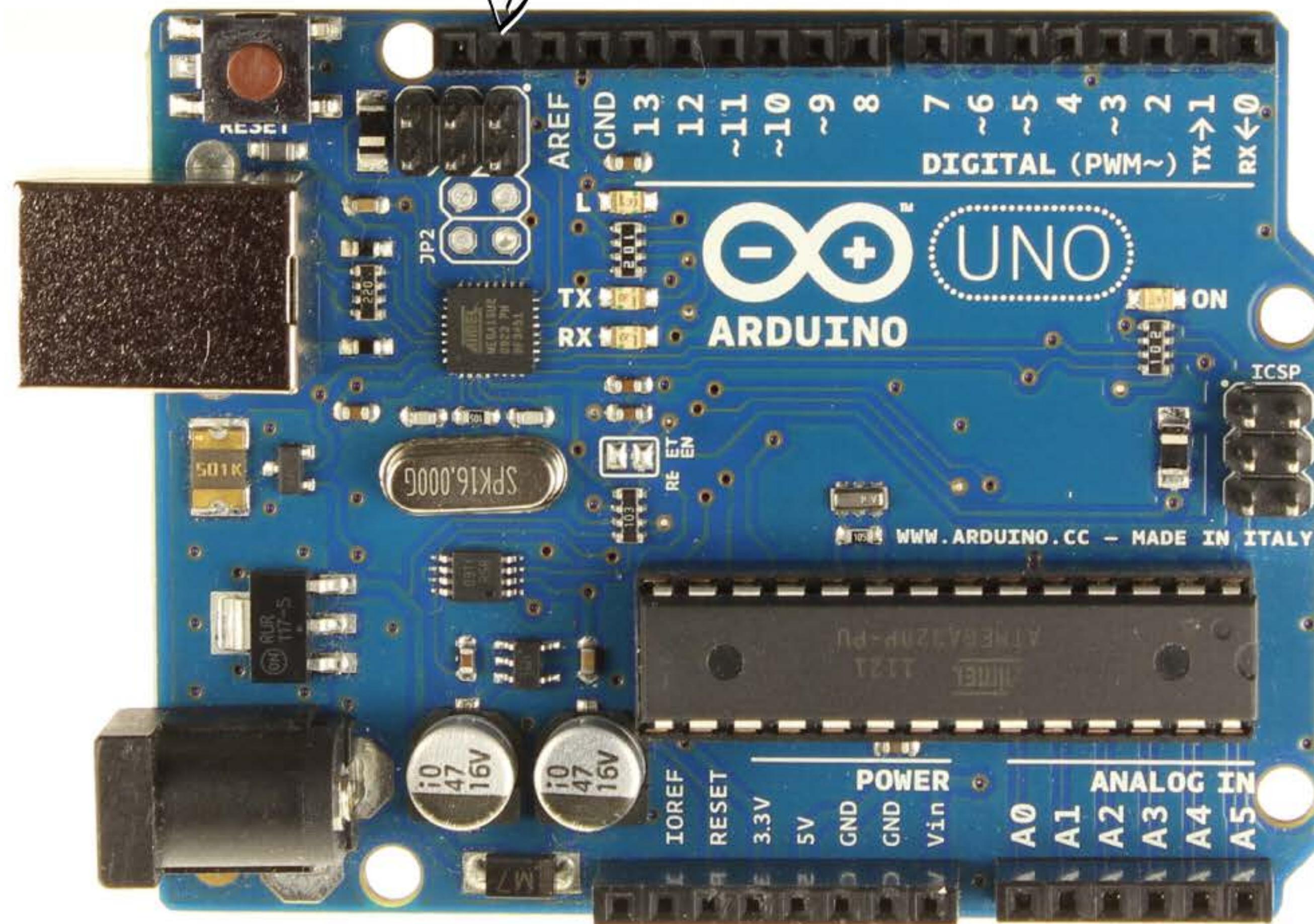


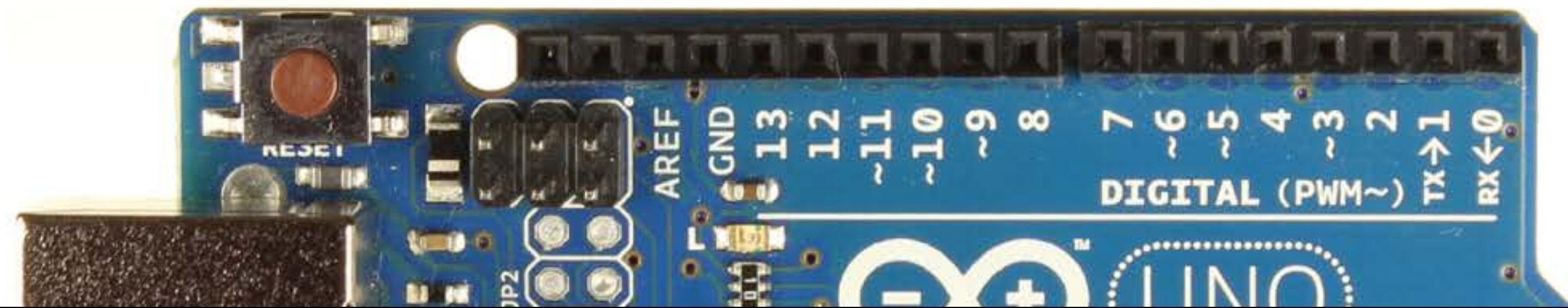


SCL(Serial Clock)

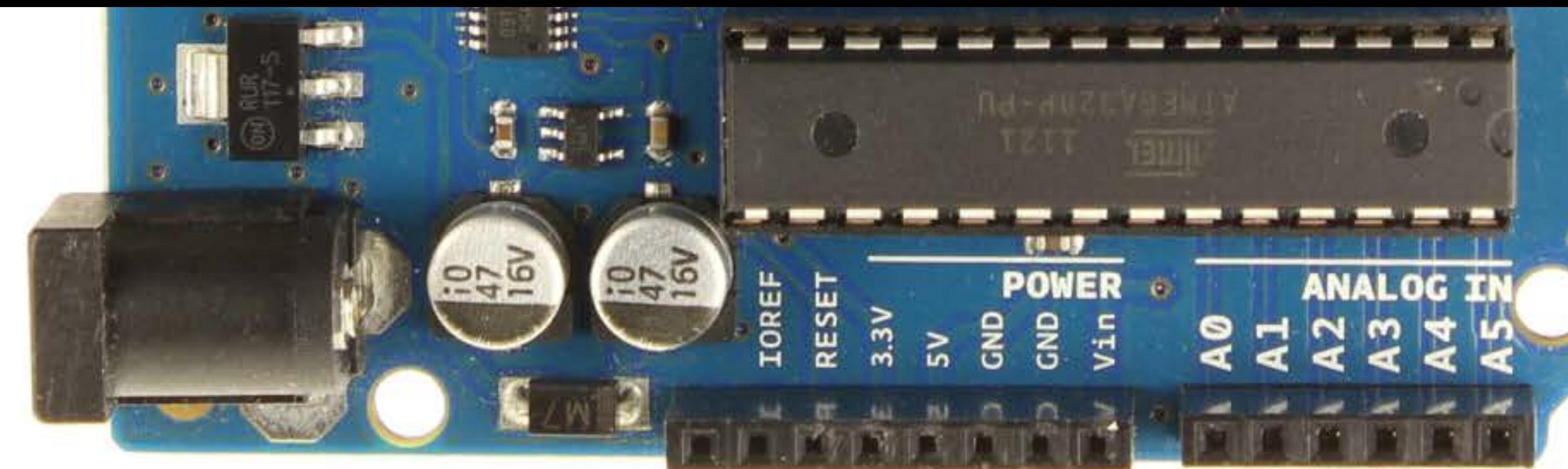


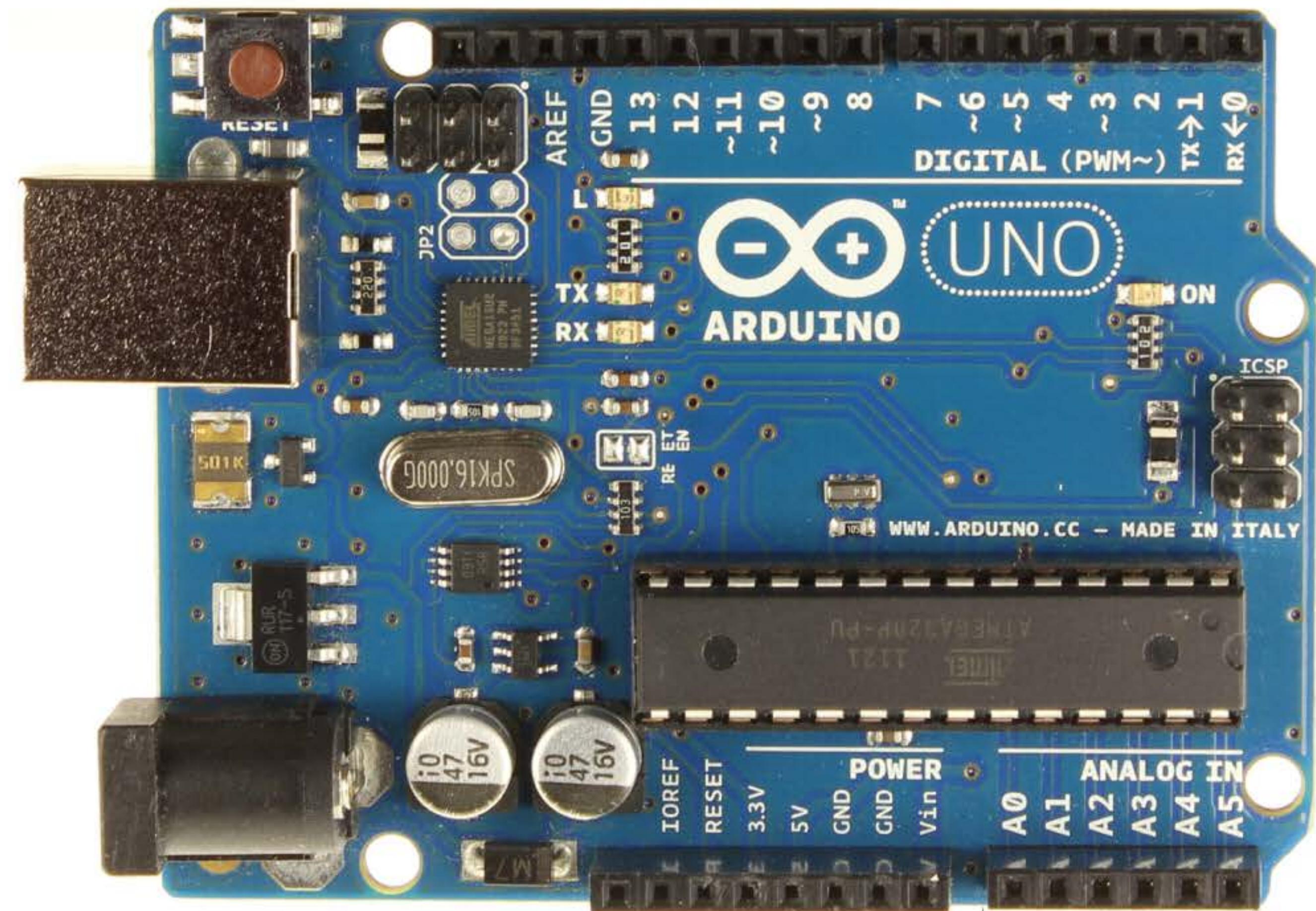
SDA (Serial Data)



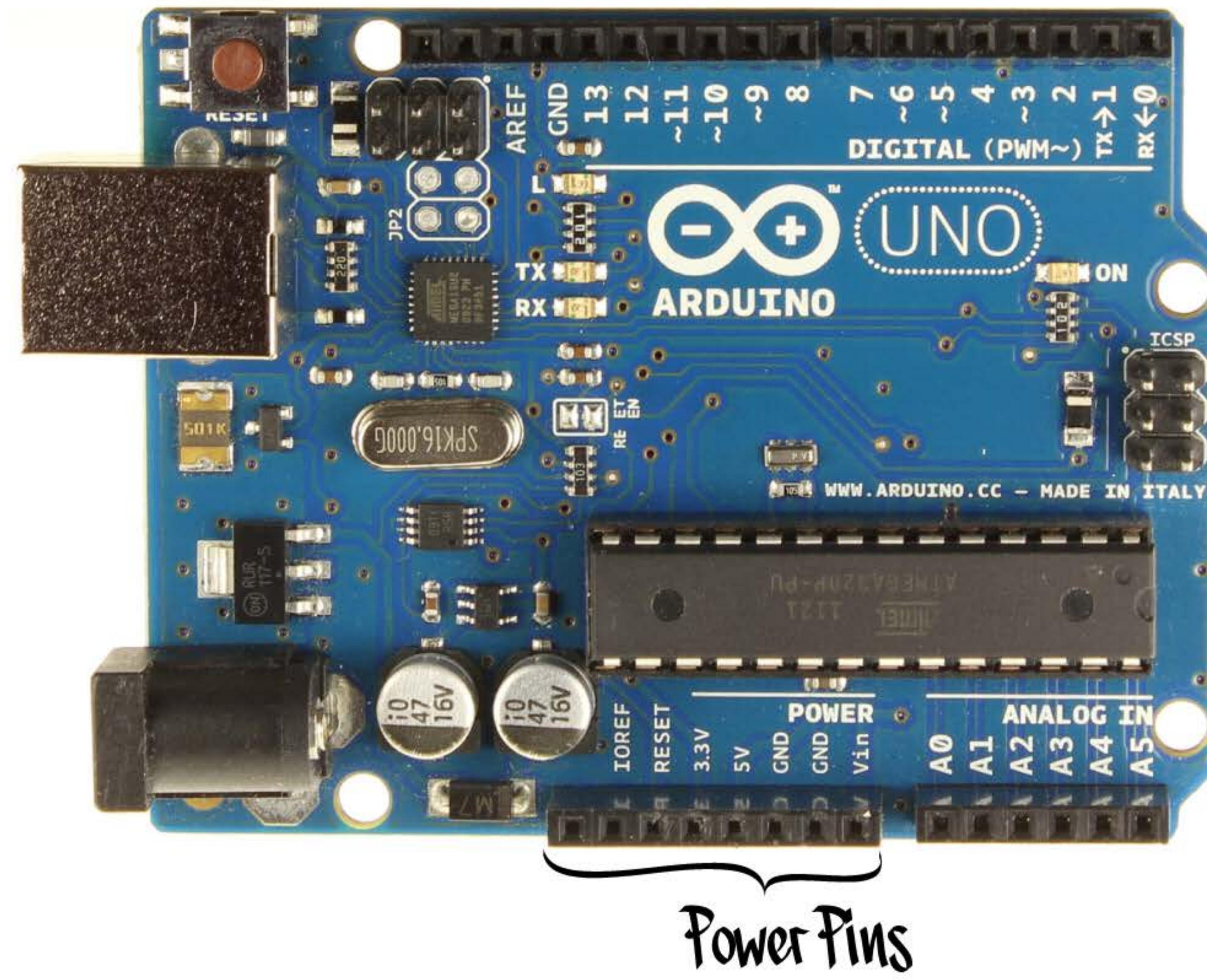


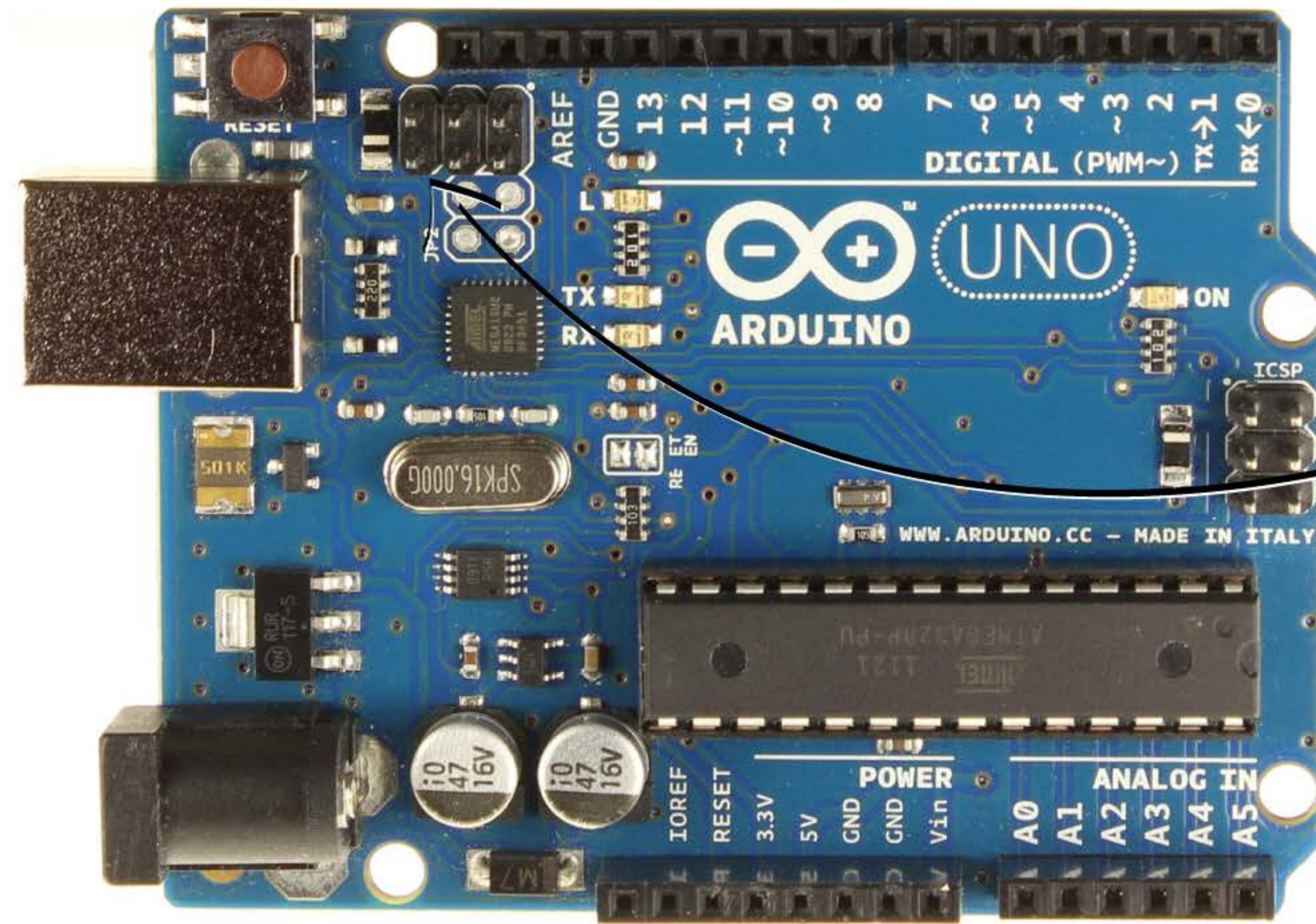
GND / Ground / Masse / Minus Pol



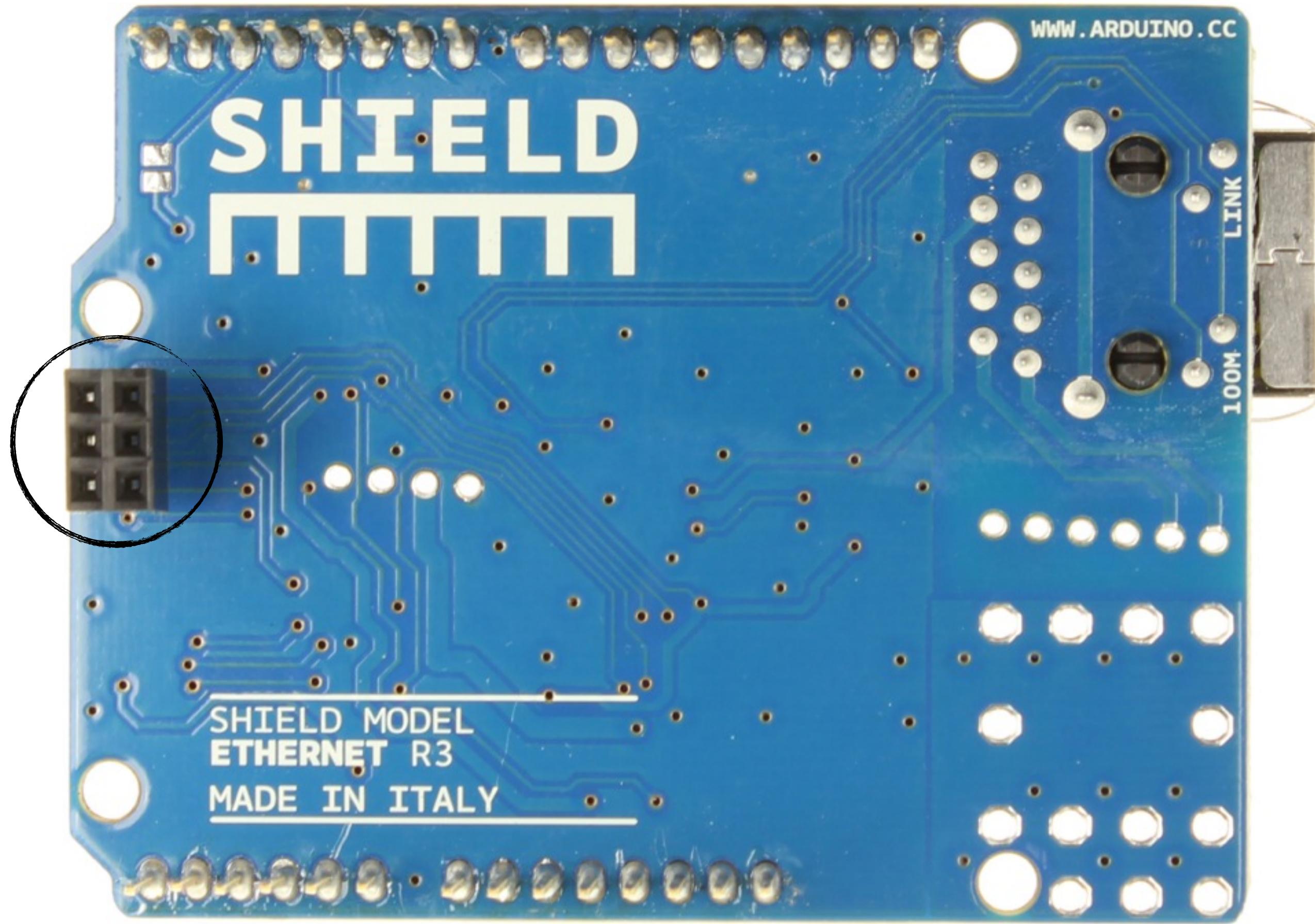


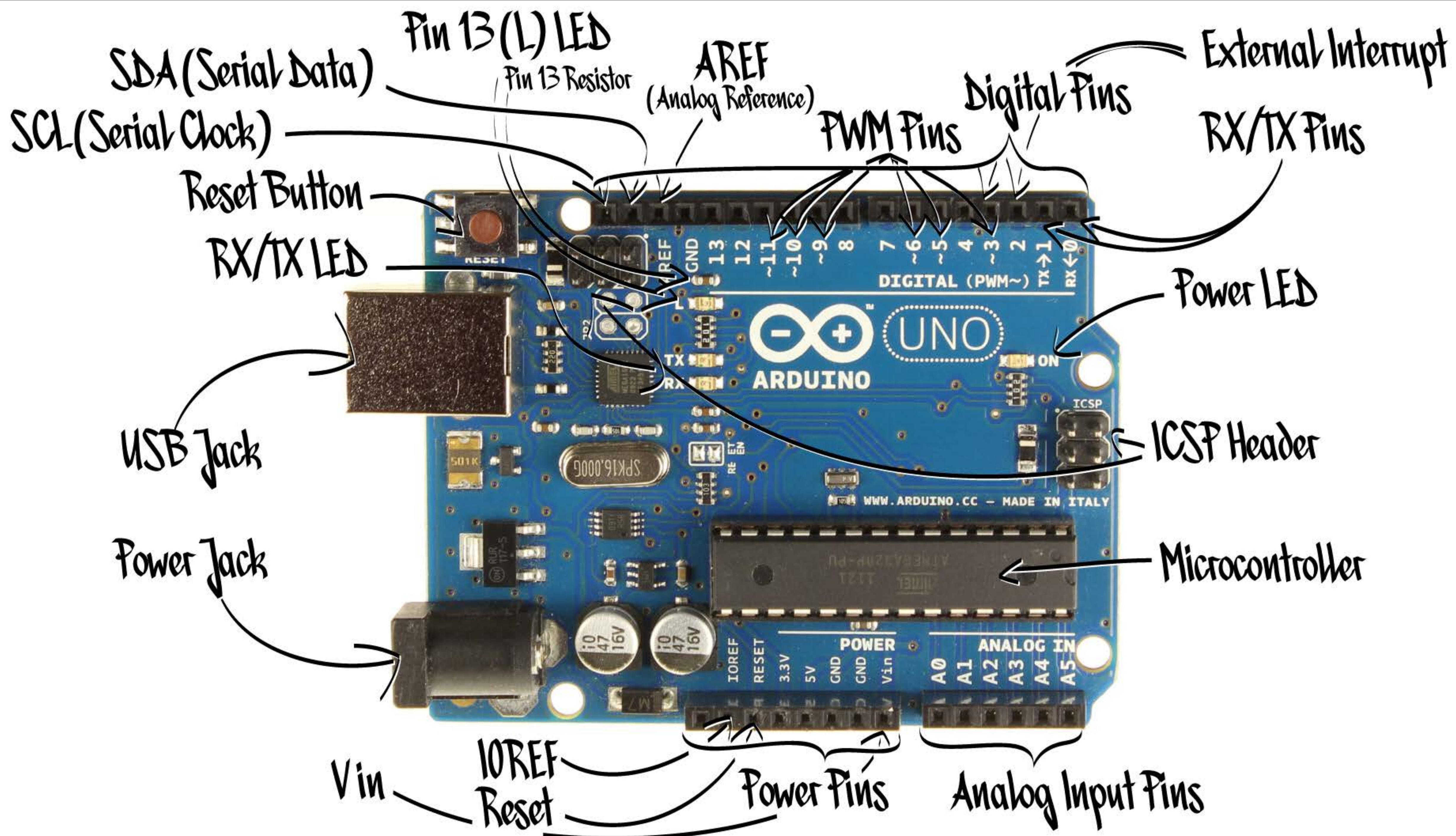
Analog Input Pins





ICSP Header





?

Elektrotechnik

Grundlagen

U (Spannung / Voltage)



U

$\frac{I \times R}{}$

I (Strom / Current)



R (Widerstand / Resistance)



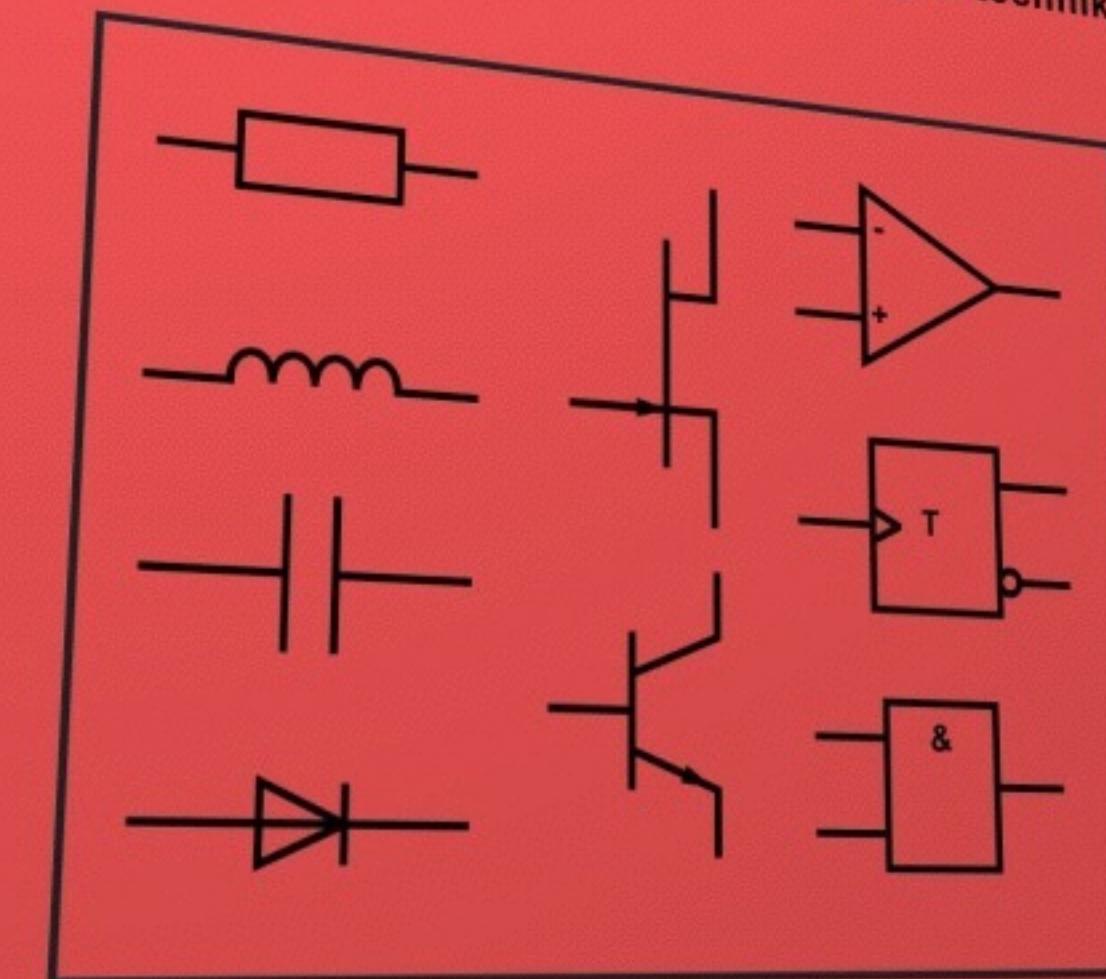
Das Magische Dreieck

adafruit.com/all-about-leds

Patrick Schnabel

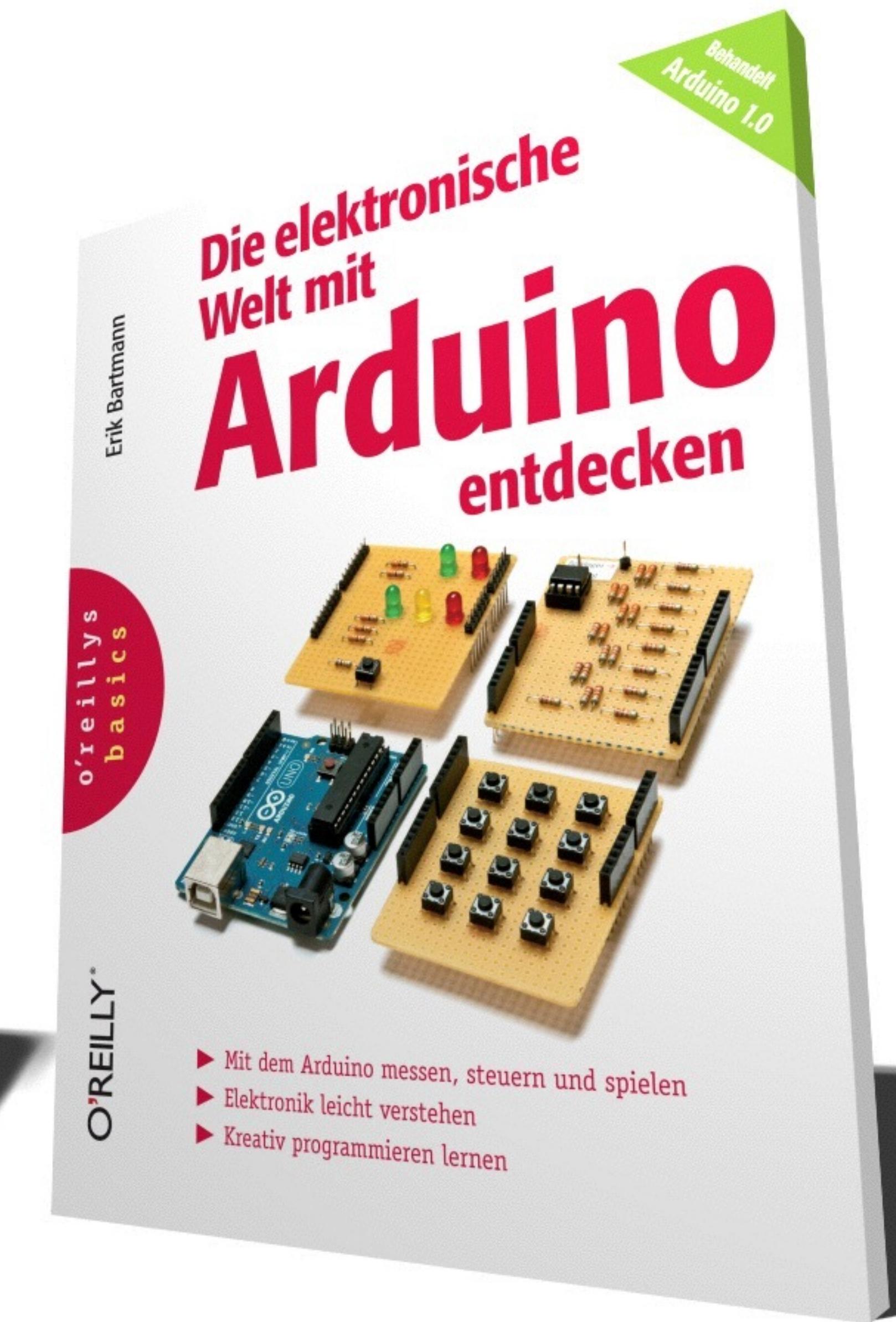
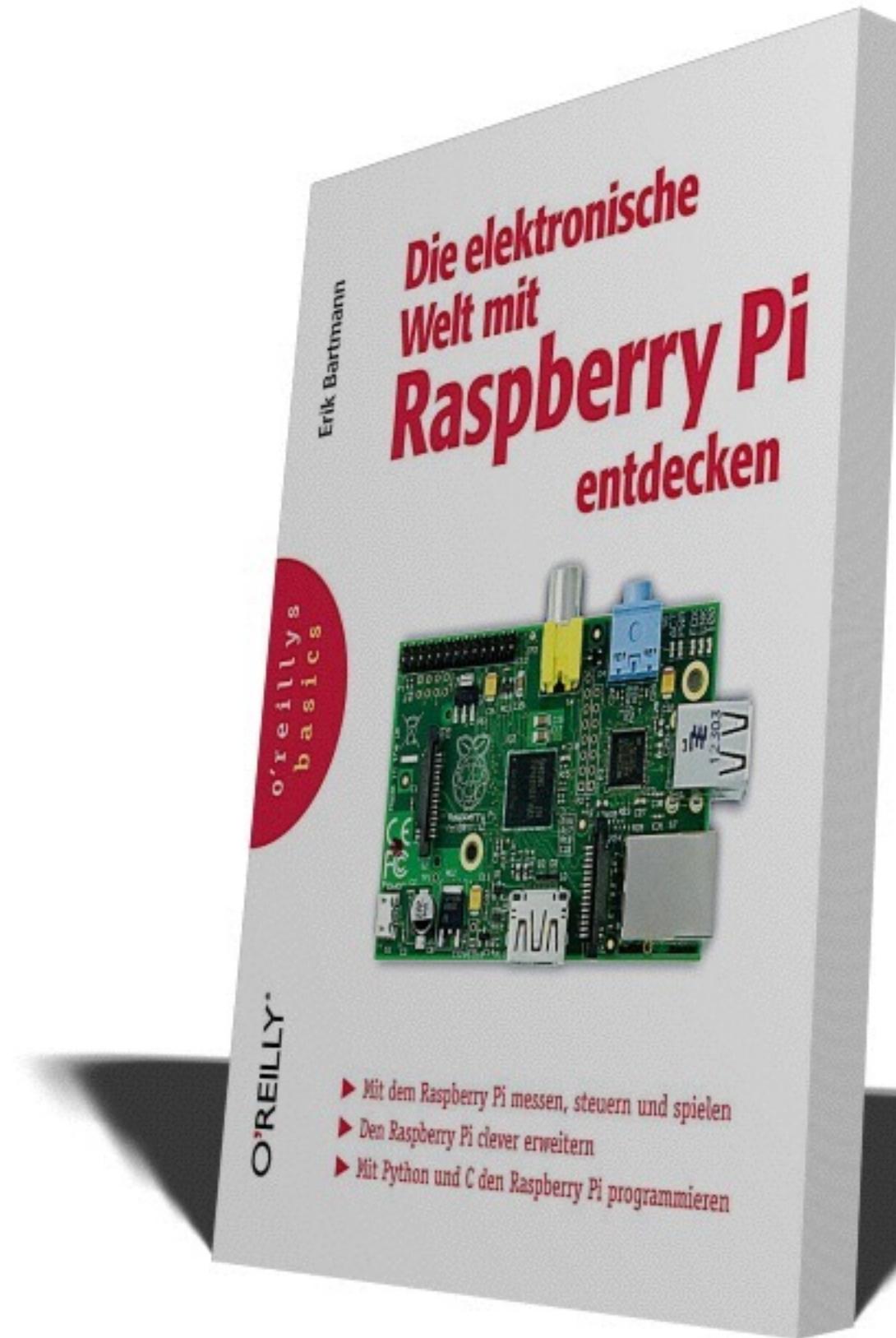
Elektronik-Fibel

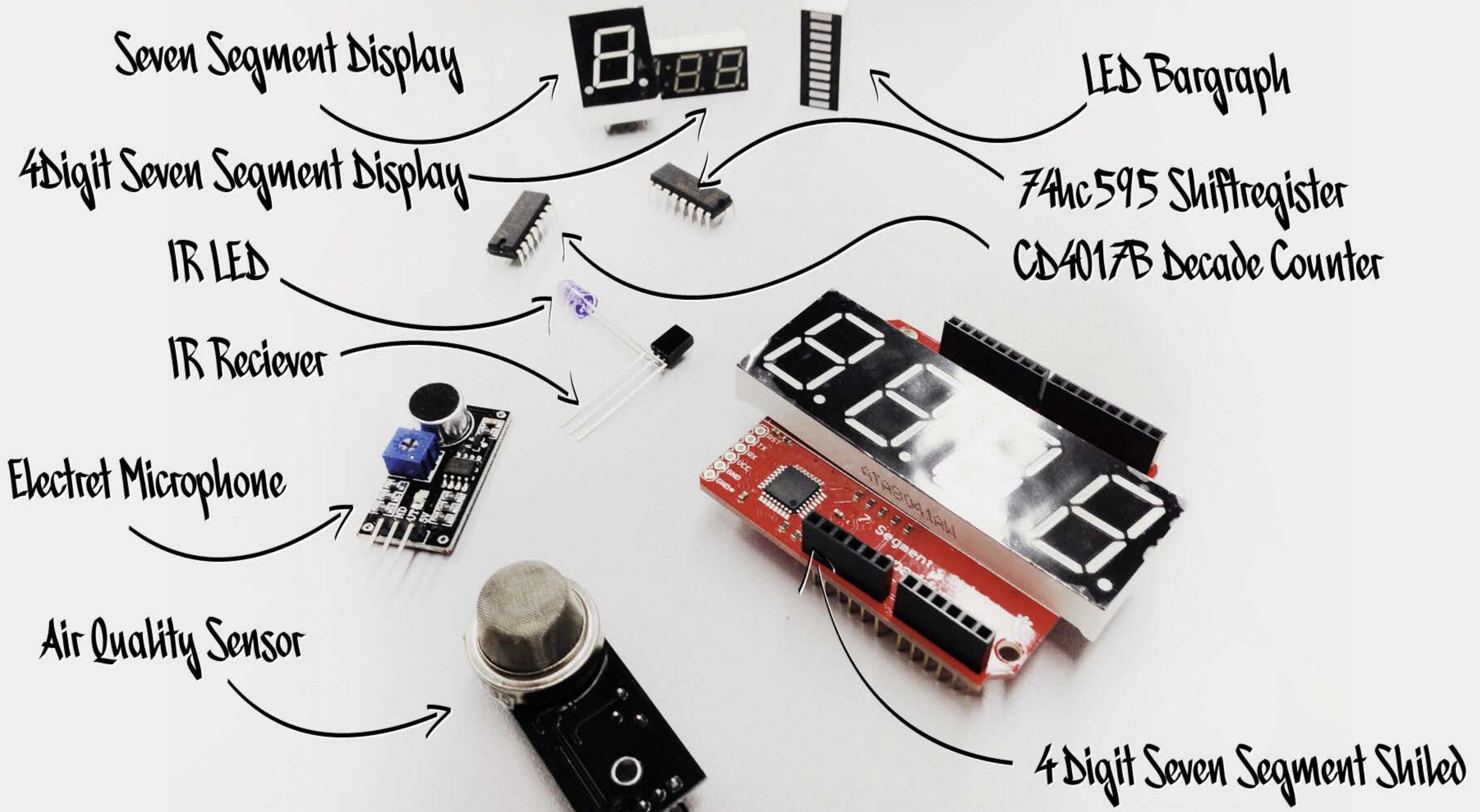
Grundlagen
Bauelemente
Schaltungstechnik
Digitaltechnik



<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 die Lötspitze sauber.

Halten Sie den Schwamm sauber.

Überprüfen Sie den Lötkolben auf Mängel.

Zweckentfremden Sie den Lötkolben nicht.

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.

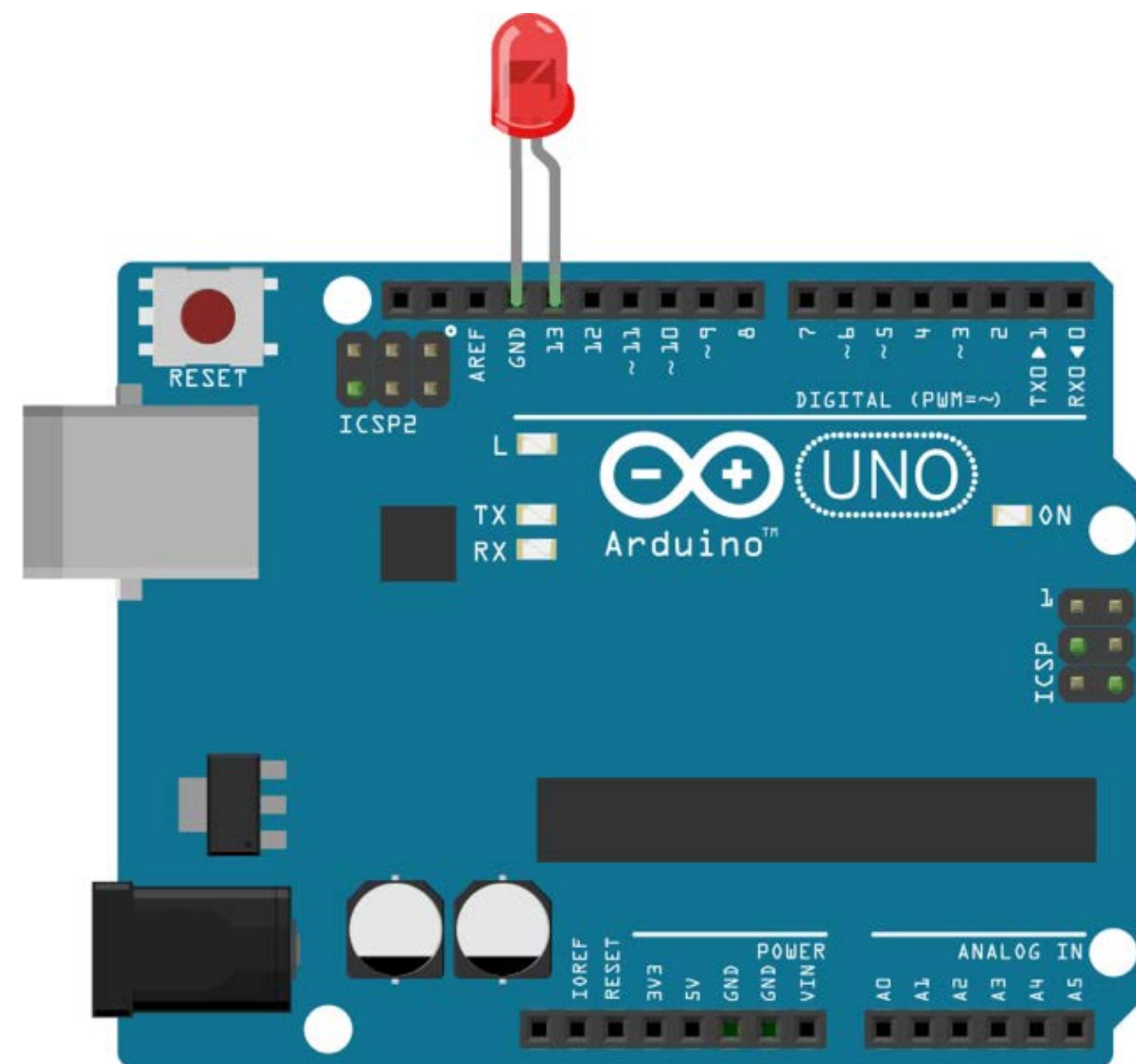
Halten sie den ausgesteckten, aber noch heißen Lötkolben von brennbaren Stoffen fern.

Lötstück erhitzen

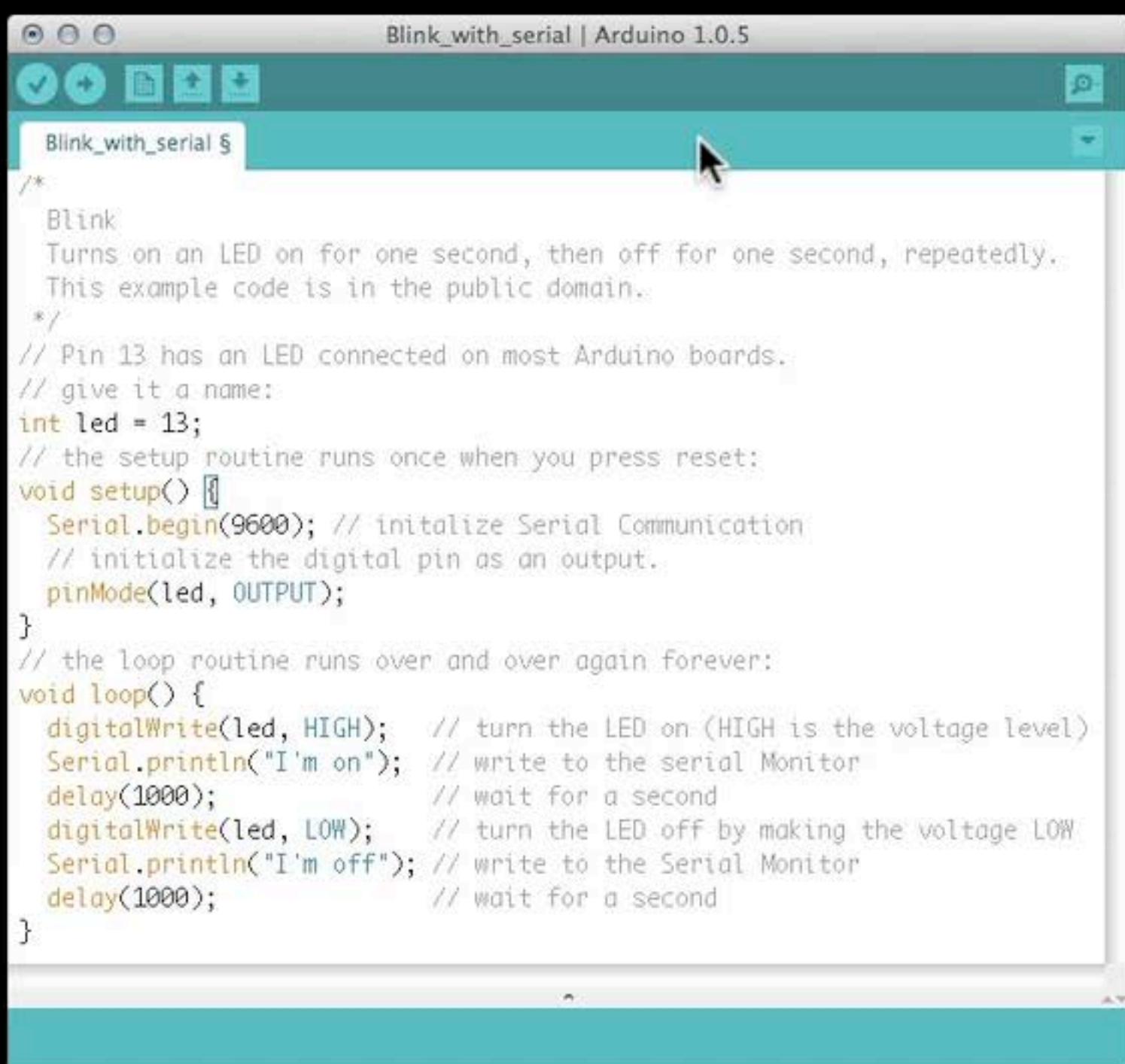
hands on Löten

hands on digital

github.com/fabiantheblind/digitalio



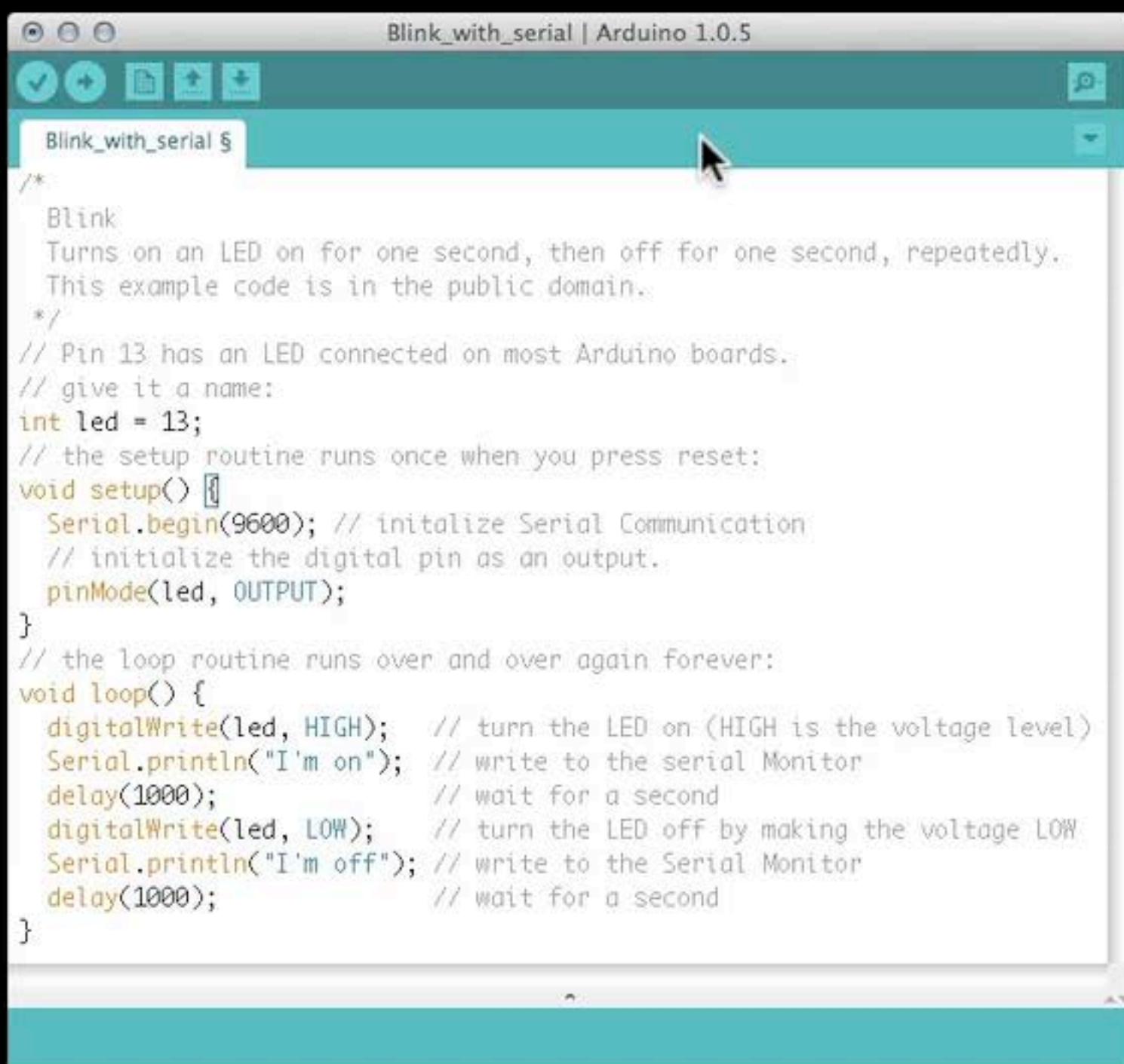
blink



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

```
/*
  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
}
```

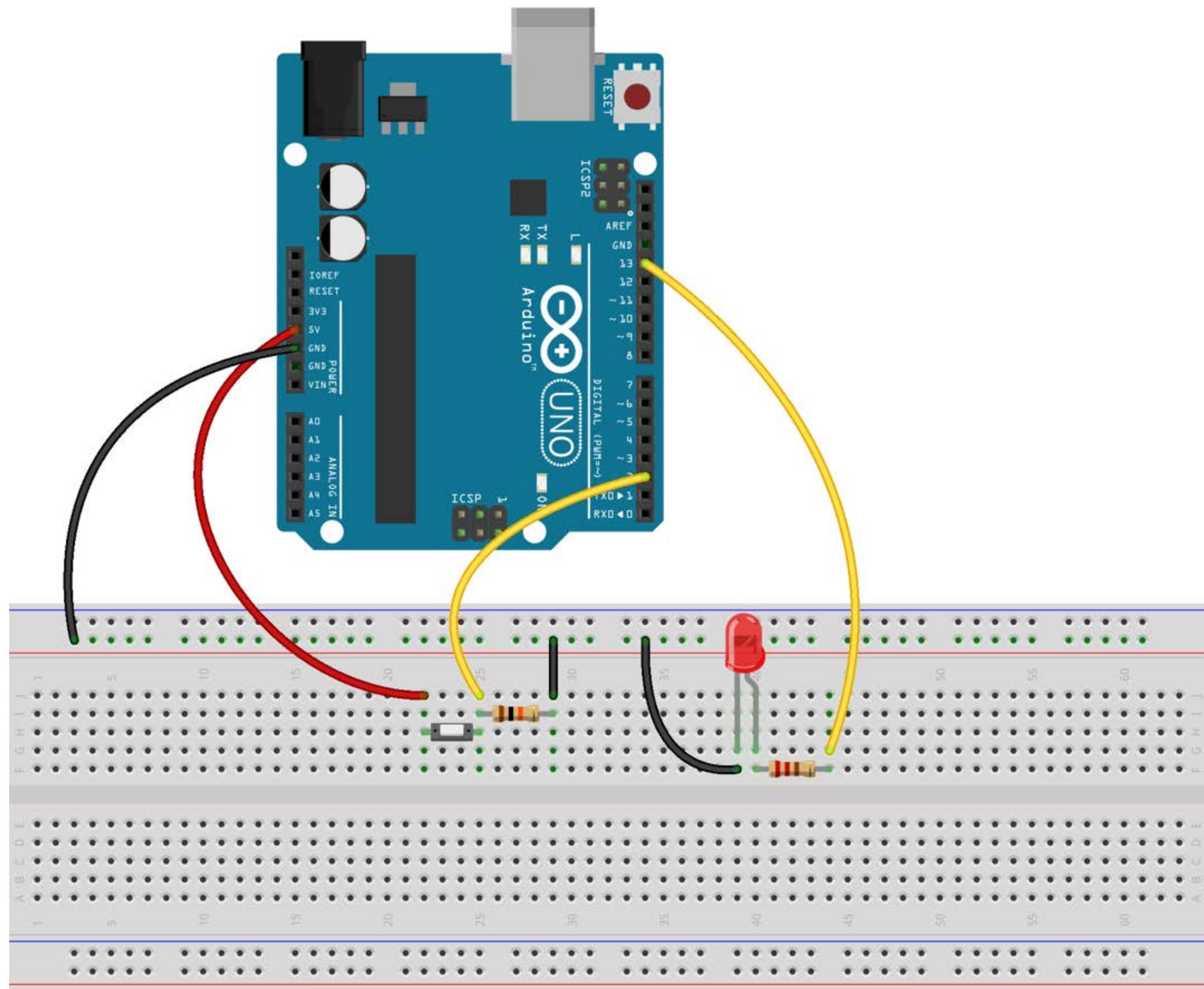
Blink with Serial Sketch



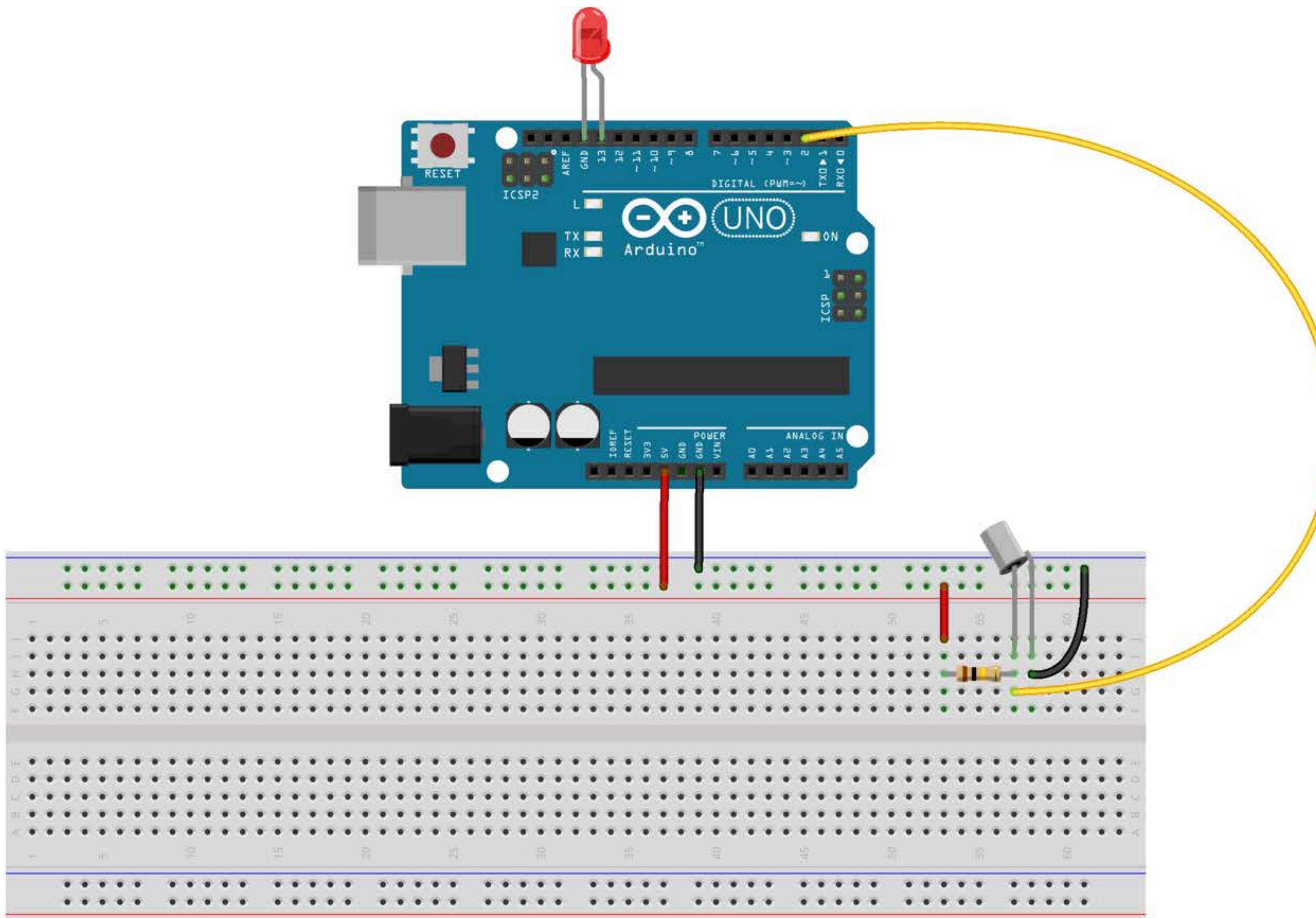
The screenshot shows the Arduino IDE interface with the title bar "Blink_with_serial | Arduino 1.0.5". The main window displays the following C++ code:

```
/*
  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");   // write to the serial Monitor
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW
  Serial.println("I'm off");  // write to the Serial Monitor
  delay(1000);               // wait for a second
}
```

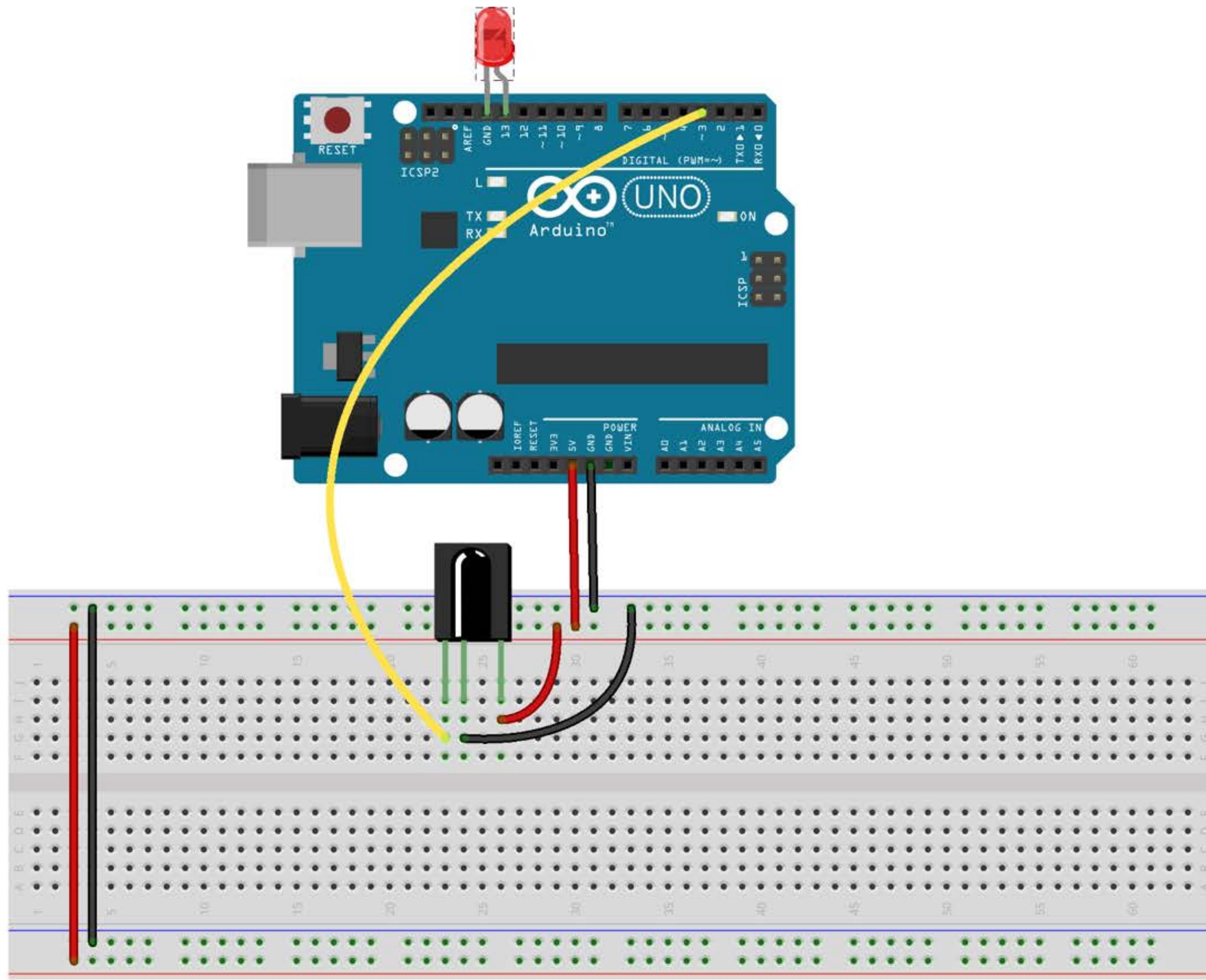
Blink with Serial Sketch



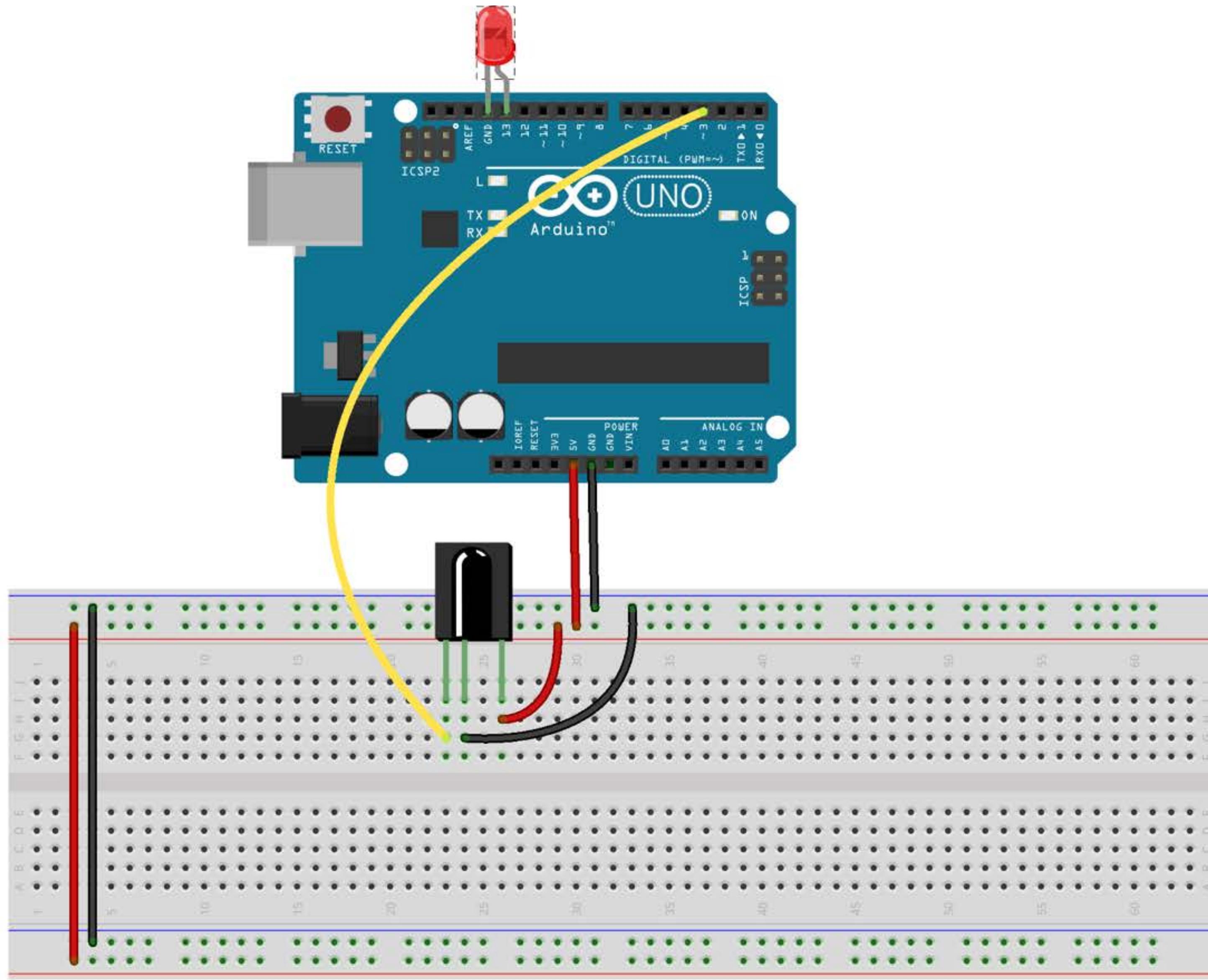
pushbutton



tiltswitch



infrared



IR Library github.com/shirriff/Arduino-IRremote

```

/*
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;

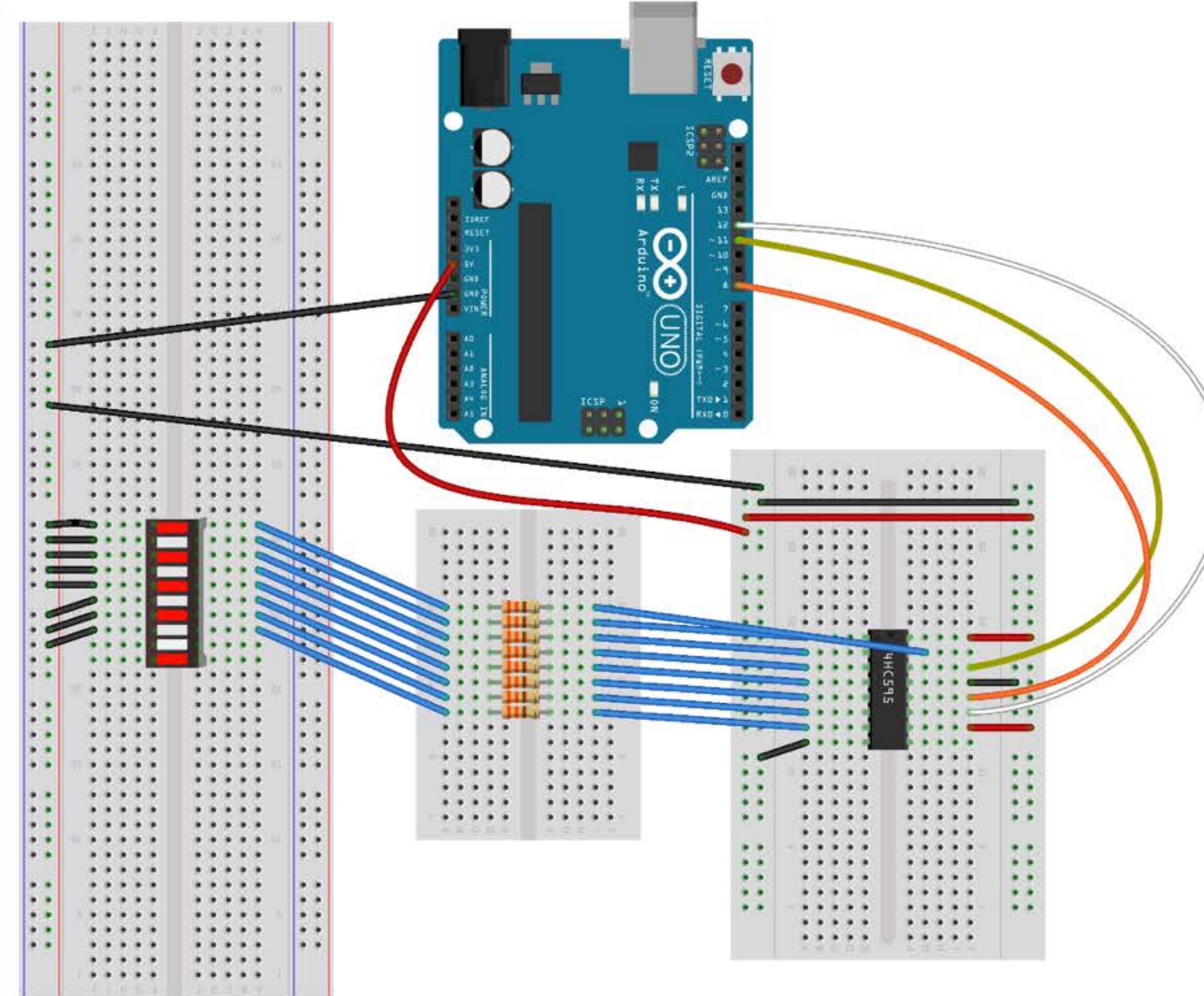
byte leds = 0;

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

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 setBarGraphOff(){
    leds = 0;
    digitalWrite(latchPin, LOW);
}

```



IC 74HC595 (Shift Register 8 Bit) + Graph

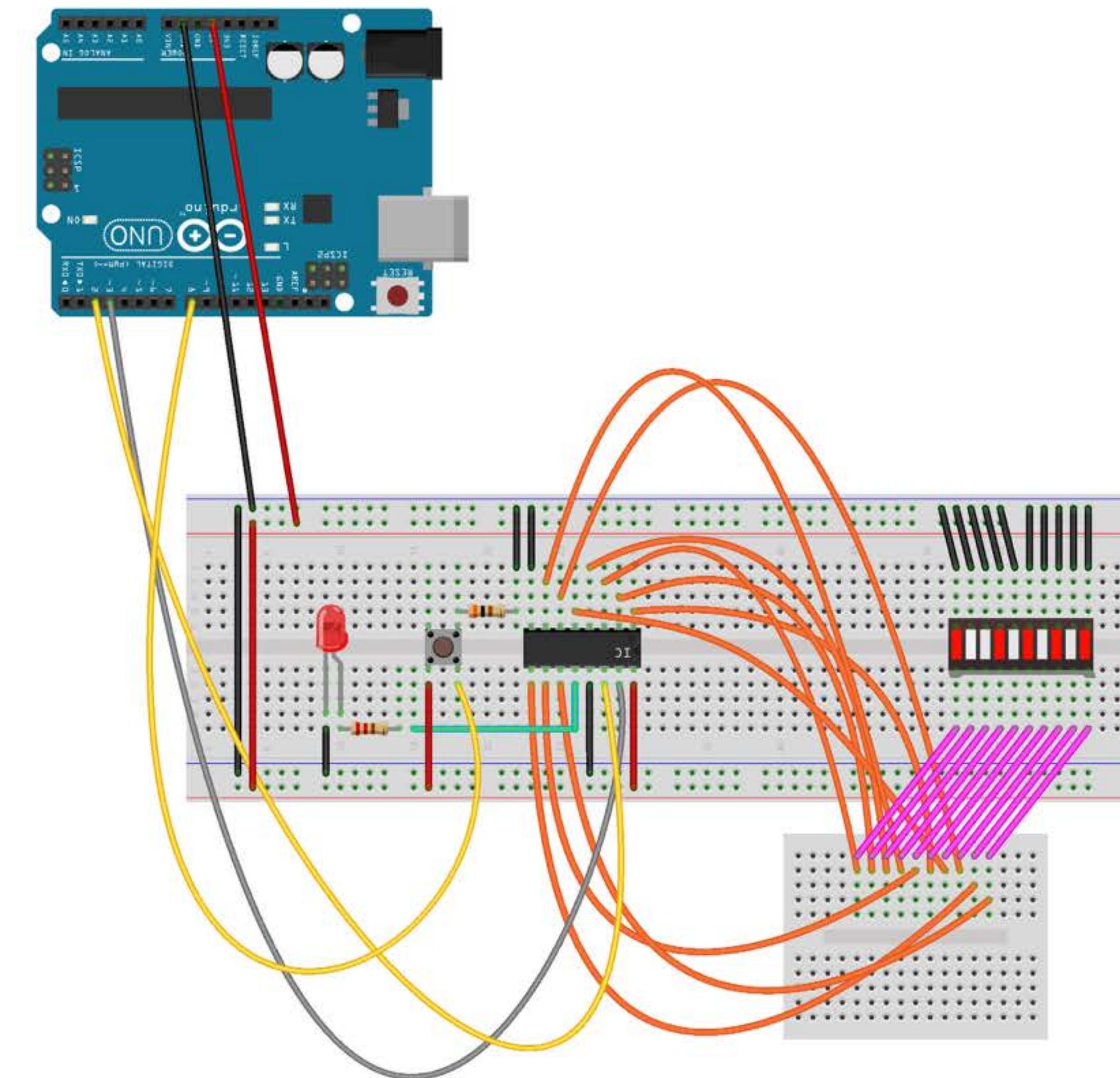
```

/*
 * 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);
  }
}

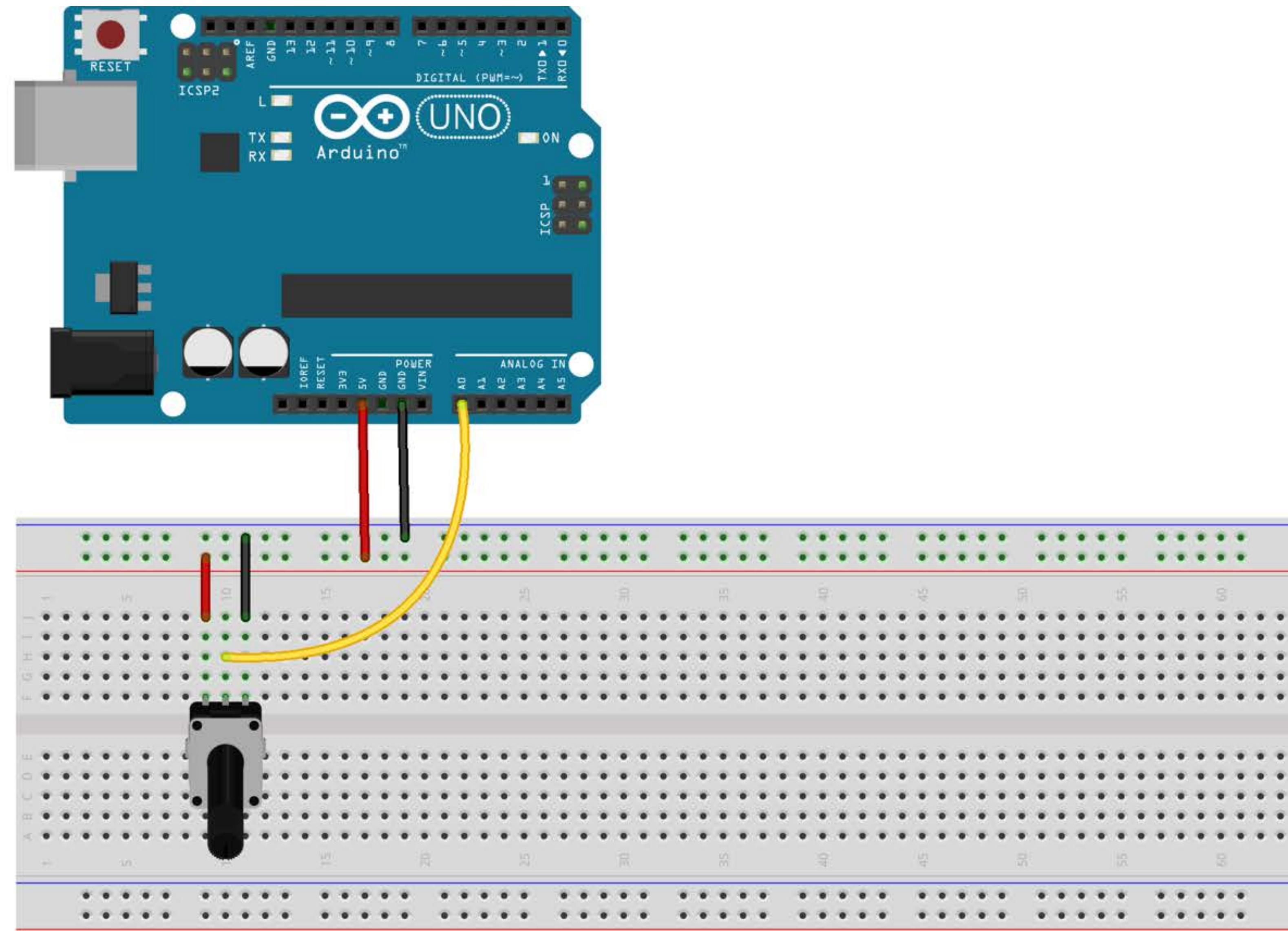
```



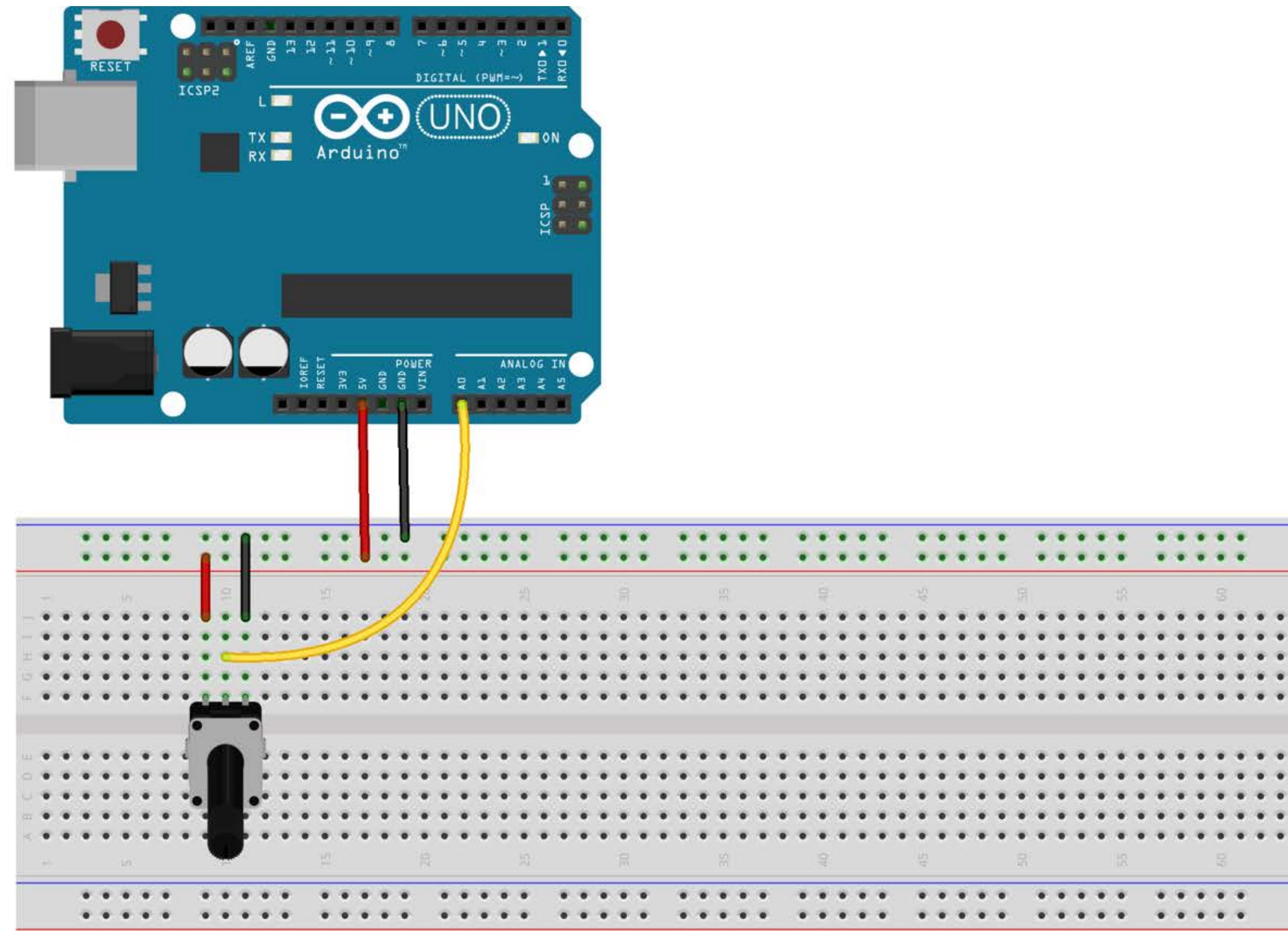
IC 4017B (Decade Counter) + LED-Graph + Pushbutton

hands on analog

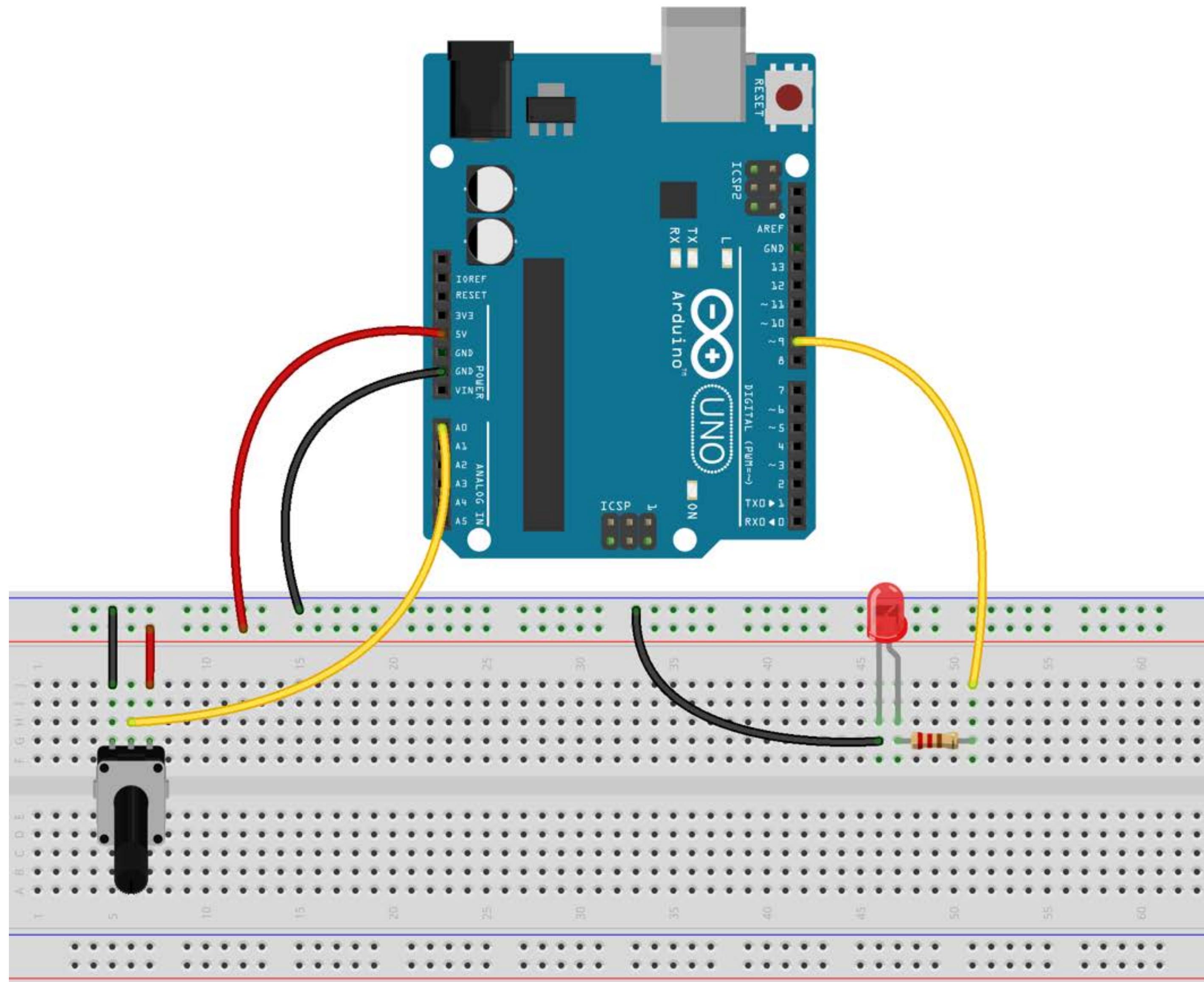
github.com/fabiantheblind/analogio



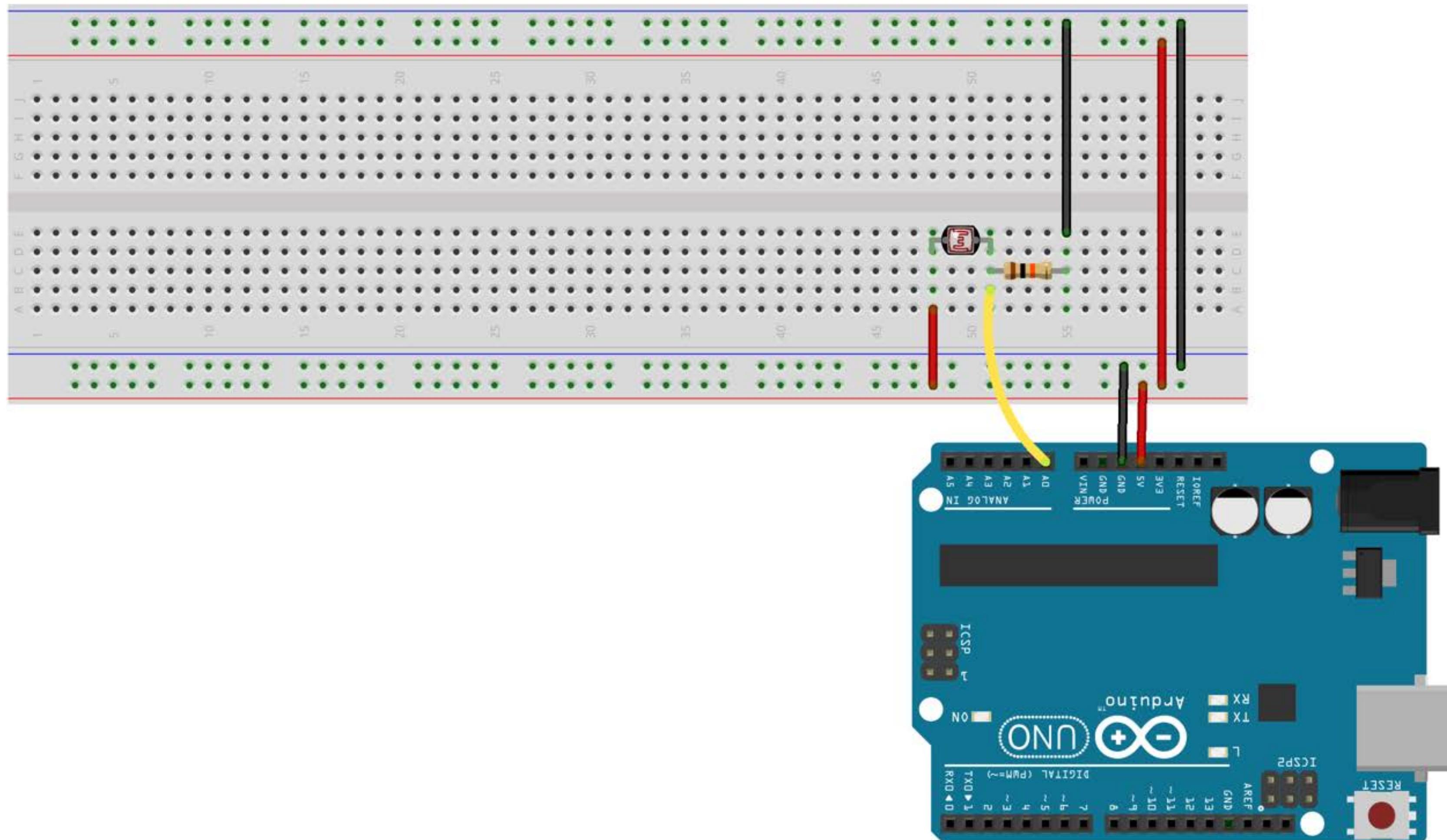
potentiometer



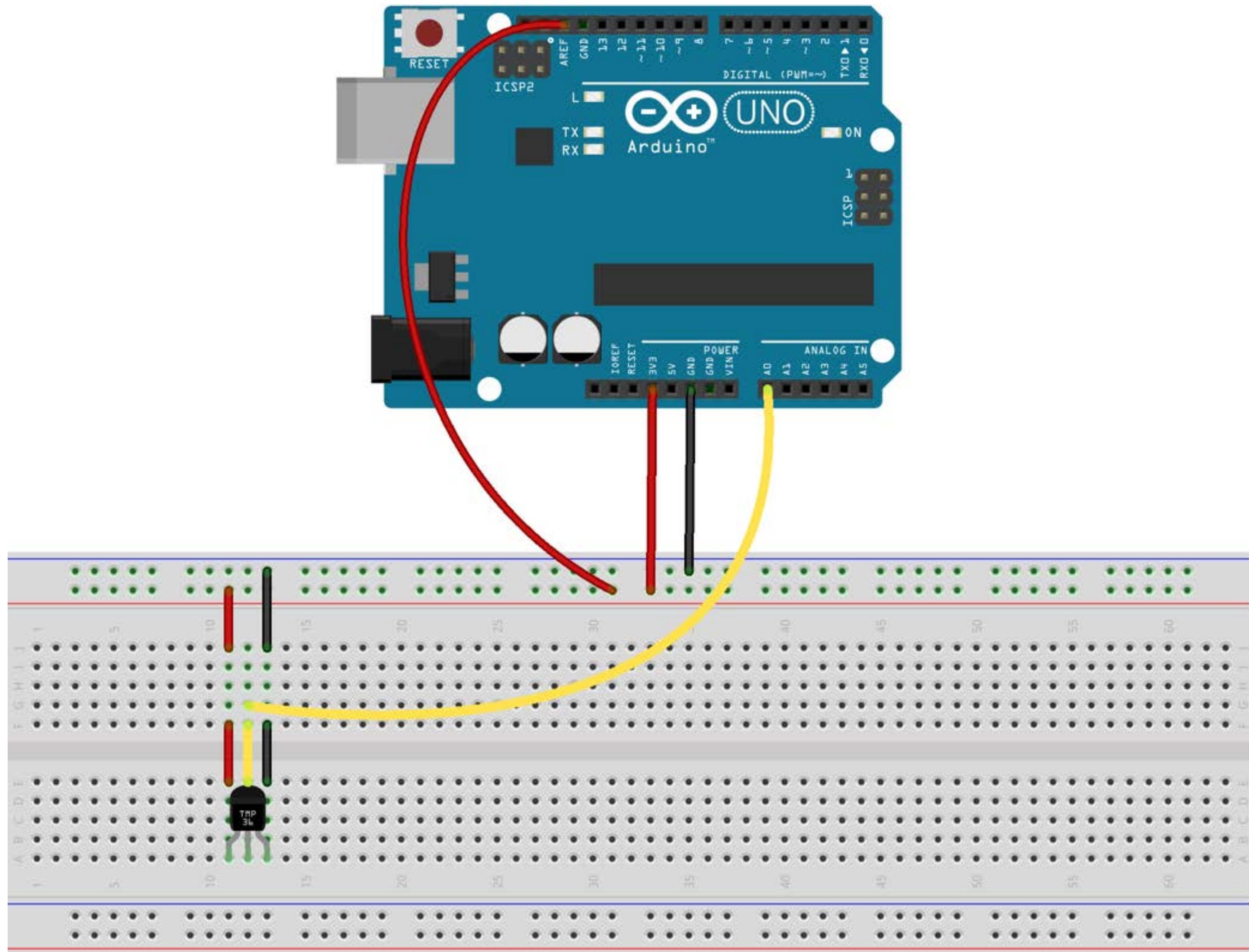
potentiometer



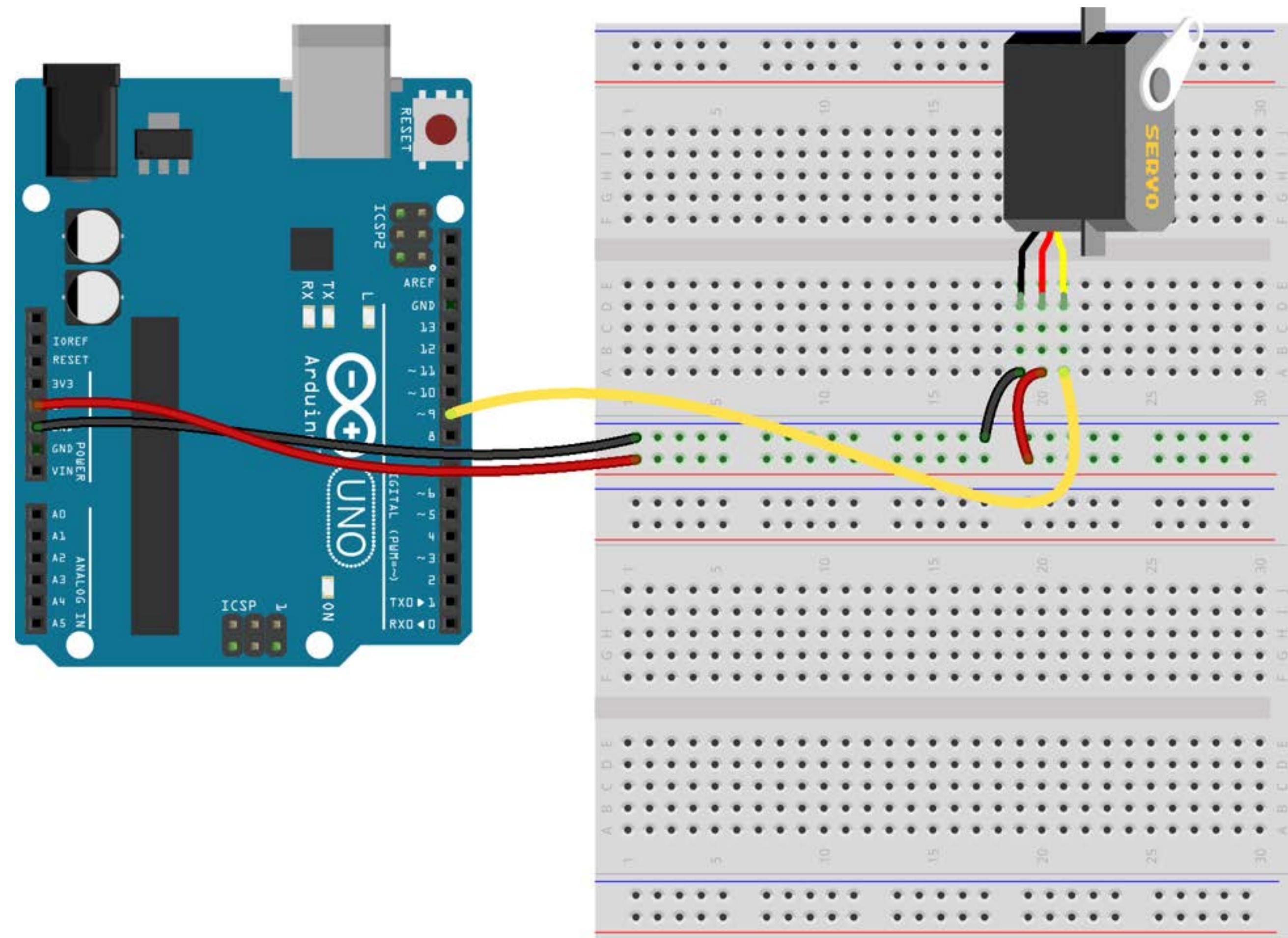
potentiometer & PWM (Pulse Width Modulation)



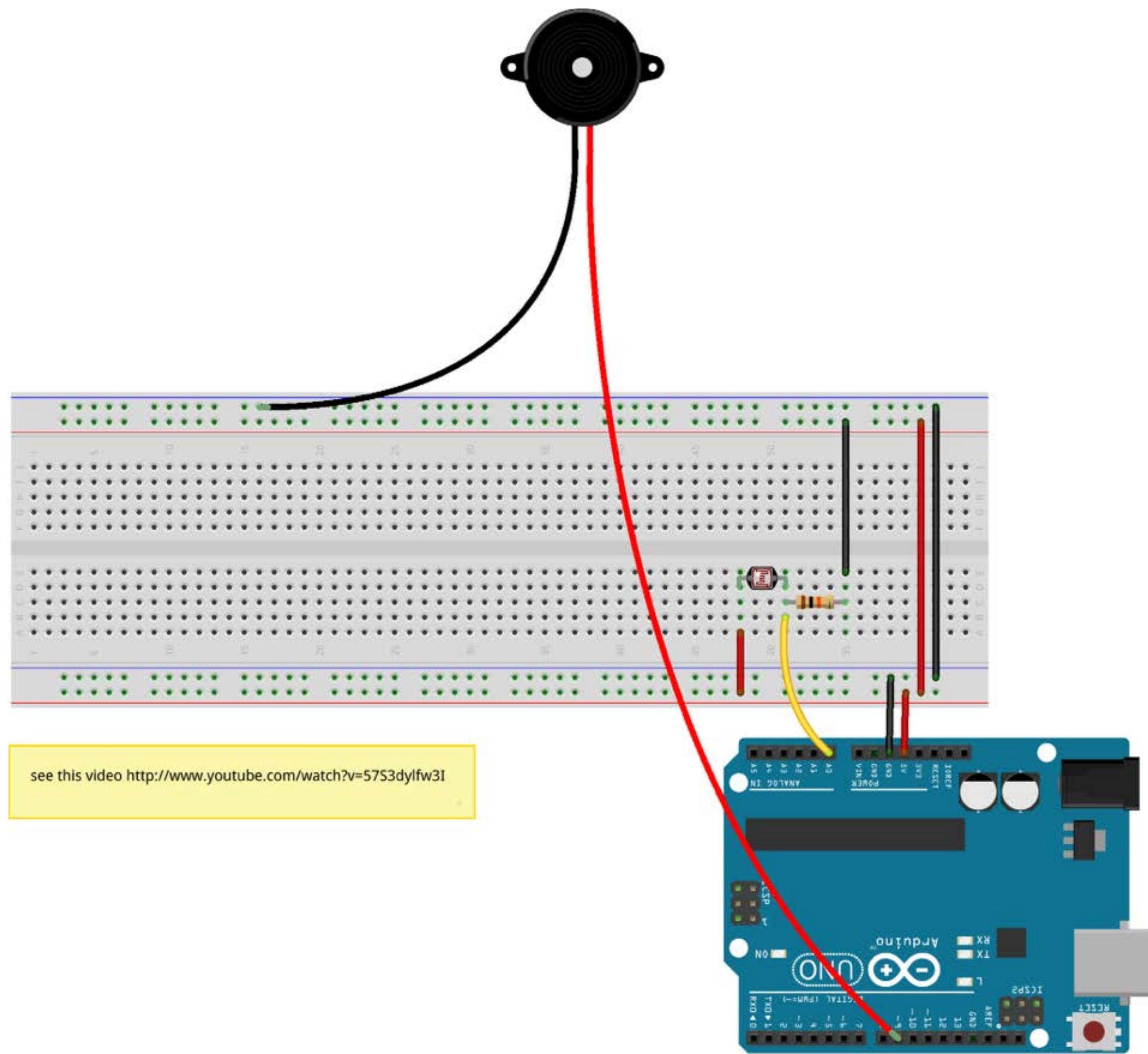
LDR (light dependent resistor), Photoresistor, Photocell



temperature AREF (Analog Reference)



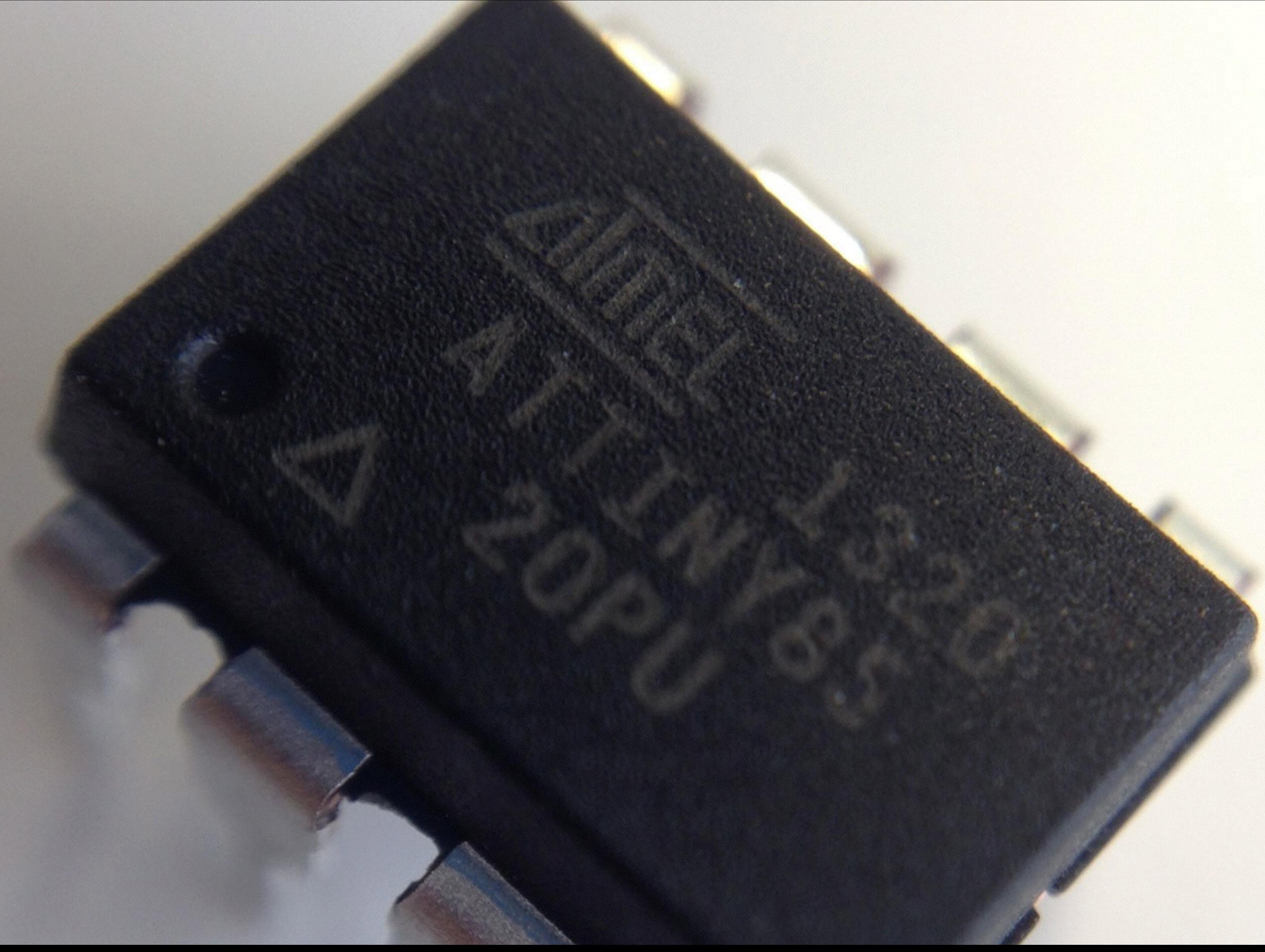
servomotor



light theremin

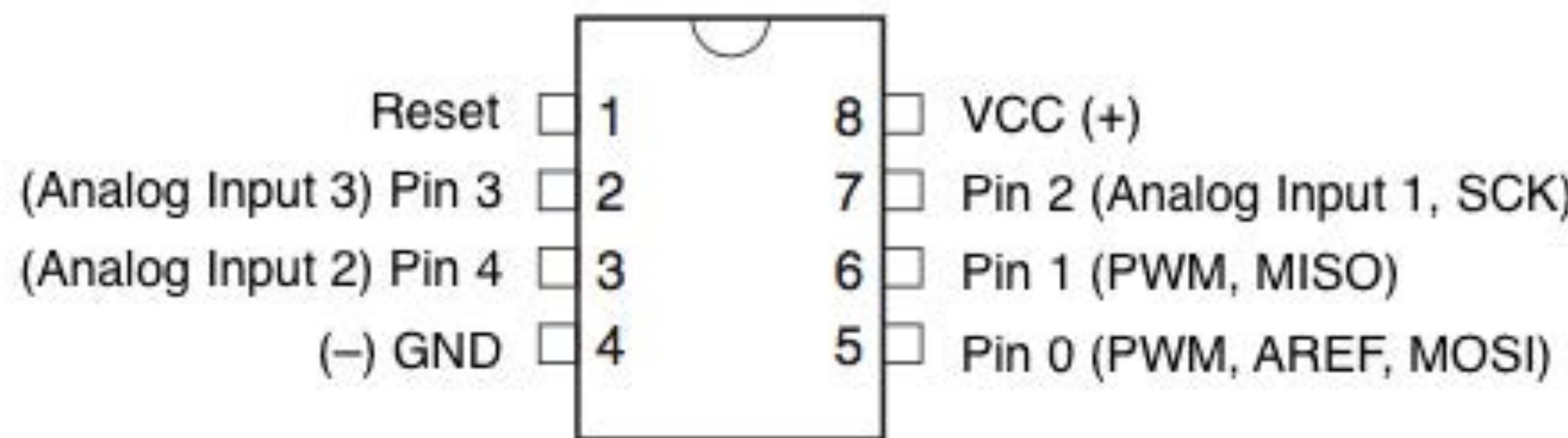
Shrink It

ATtiny 45 || ATtiny 85



ATTiny

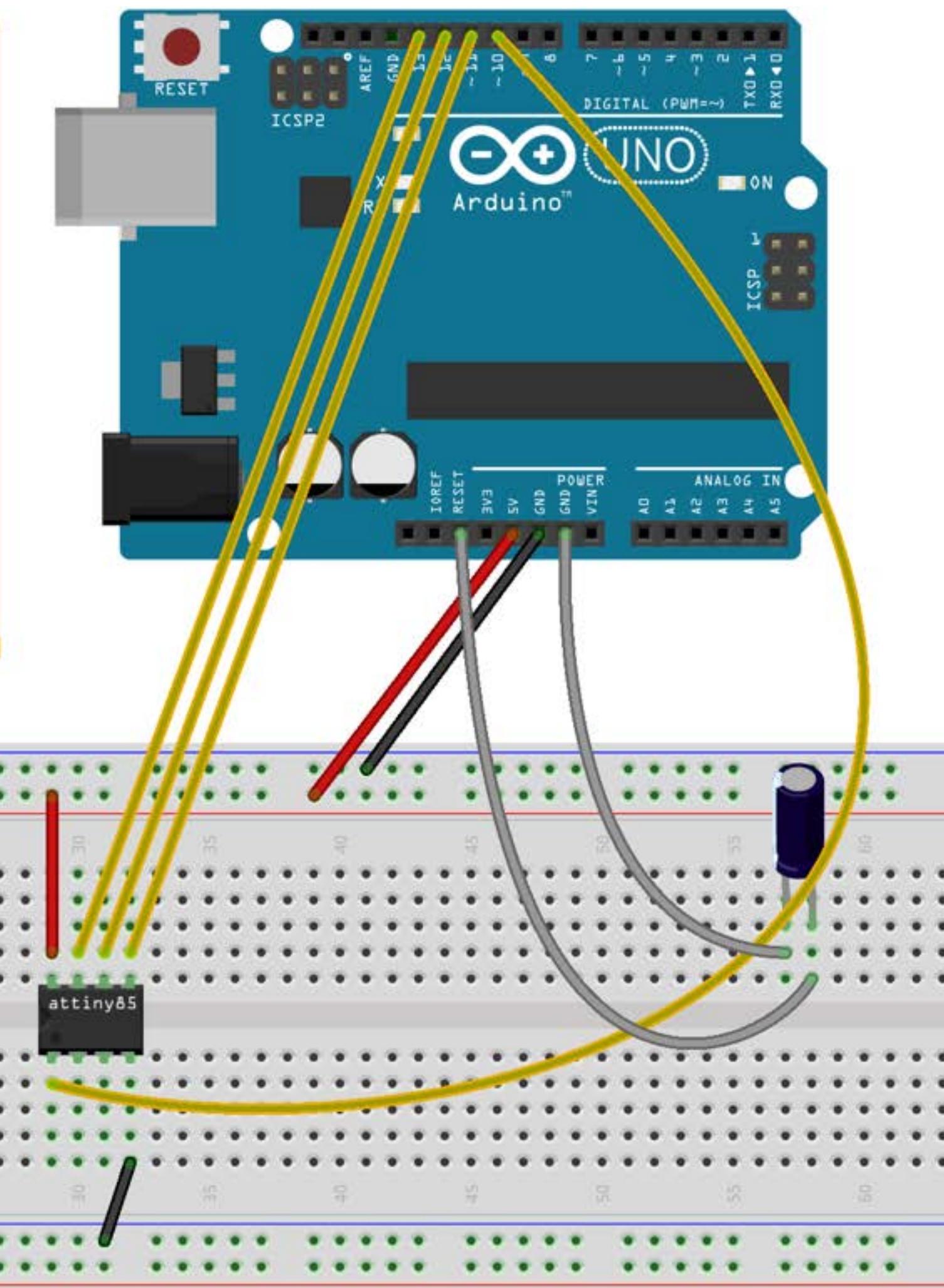
ATtiny45 / ATtiny85



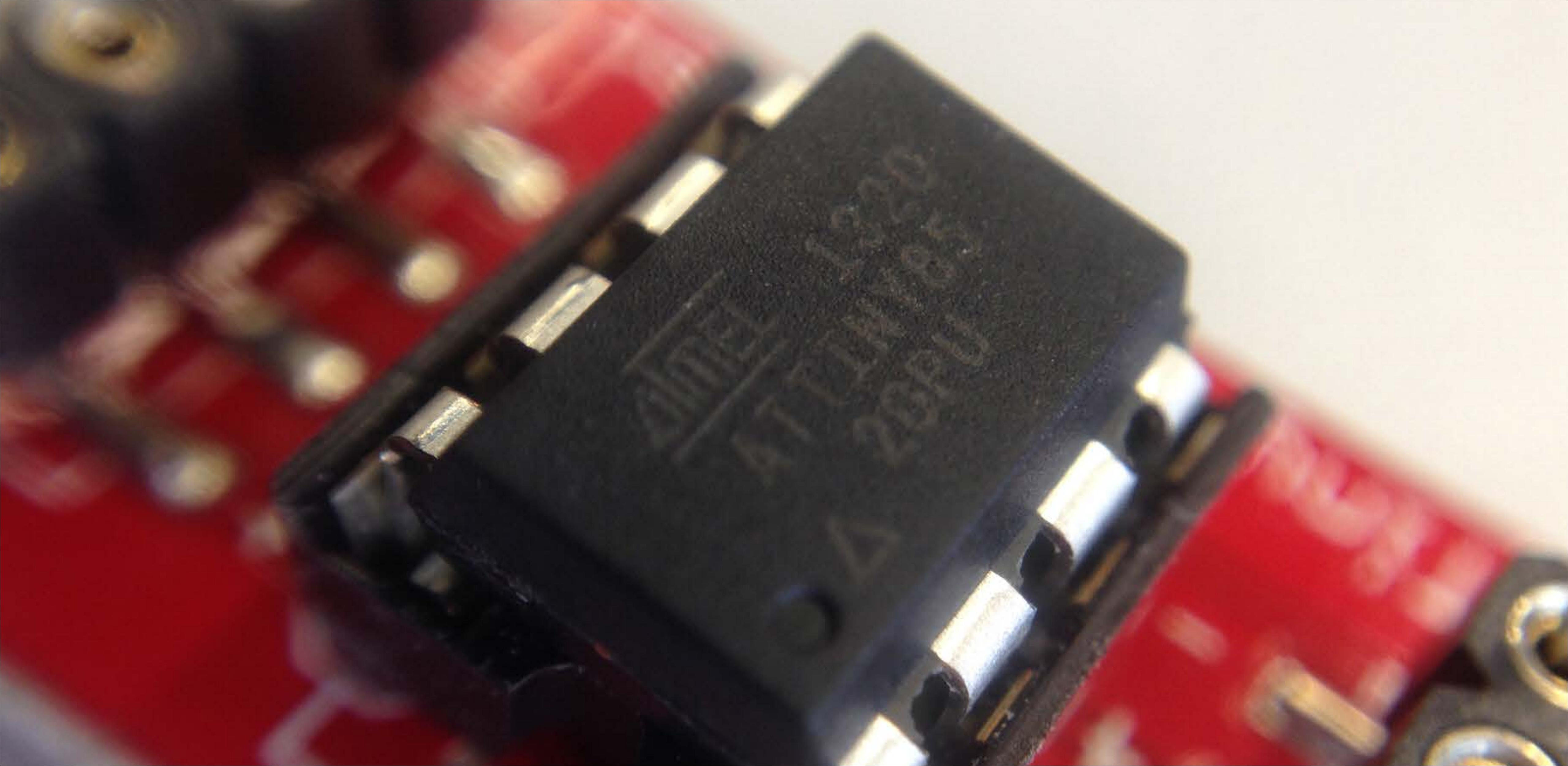
ATtiny Pin Layout

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

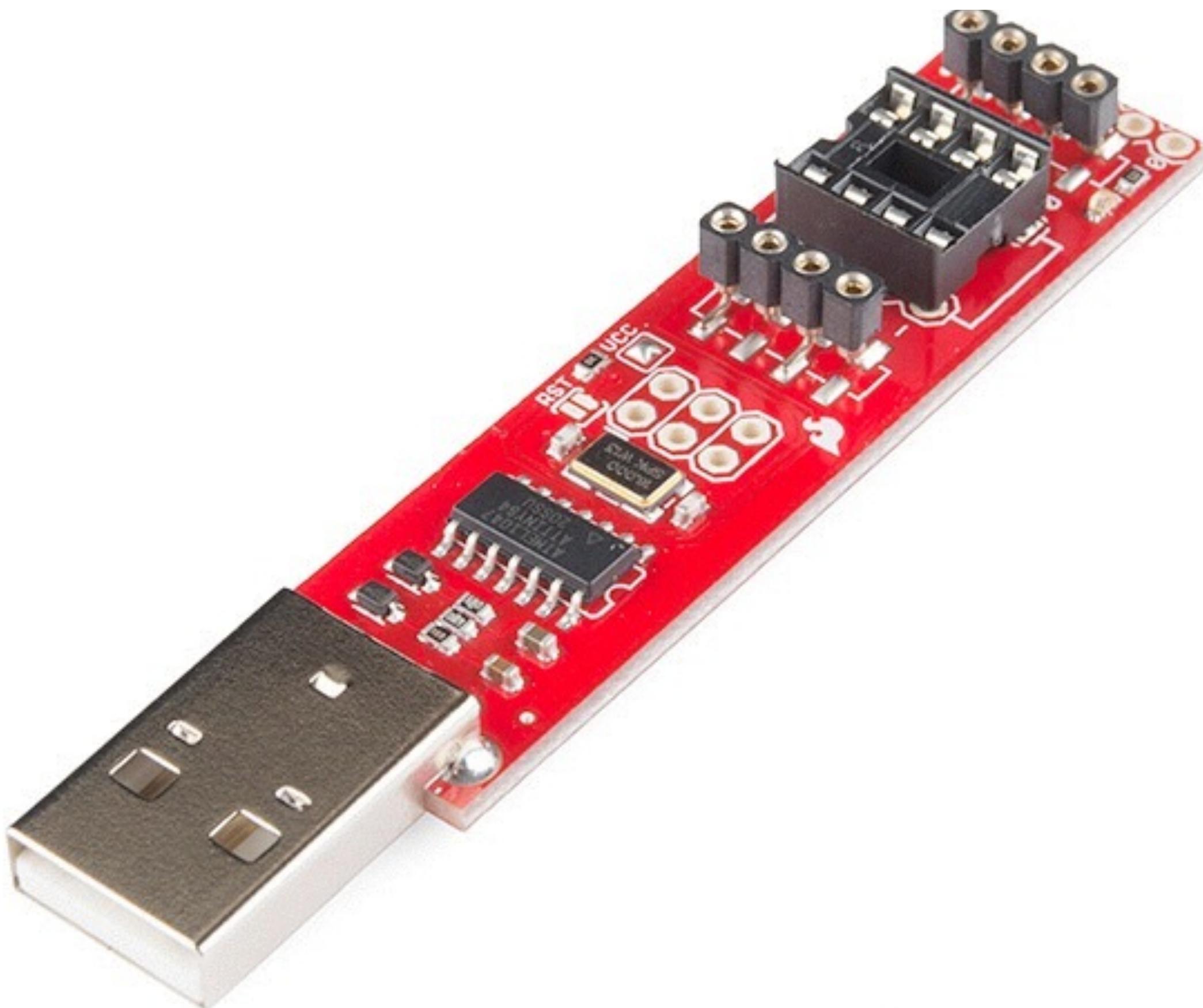
See the tutorials on how to set it up @
<http://highlowtech.org/?p=1229>



program attiny85 with arduino isp



Tiny AVR Programmer



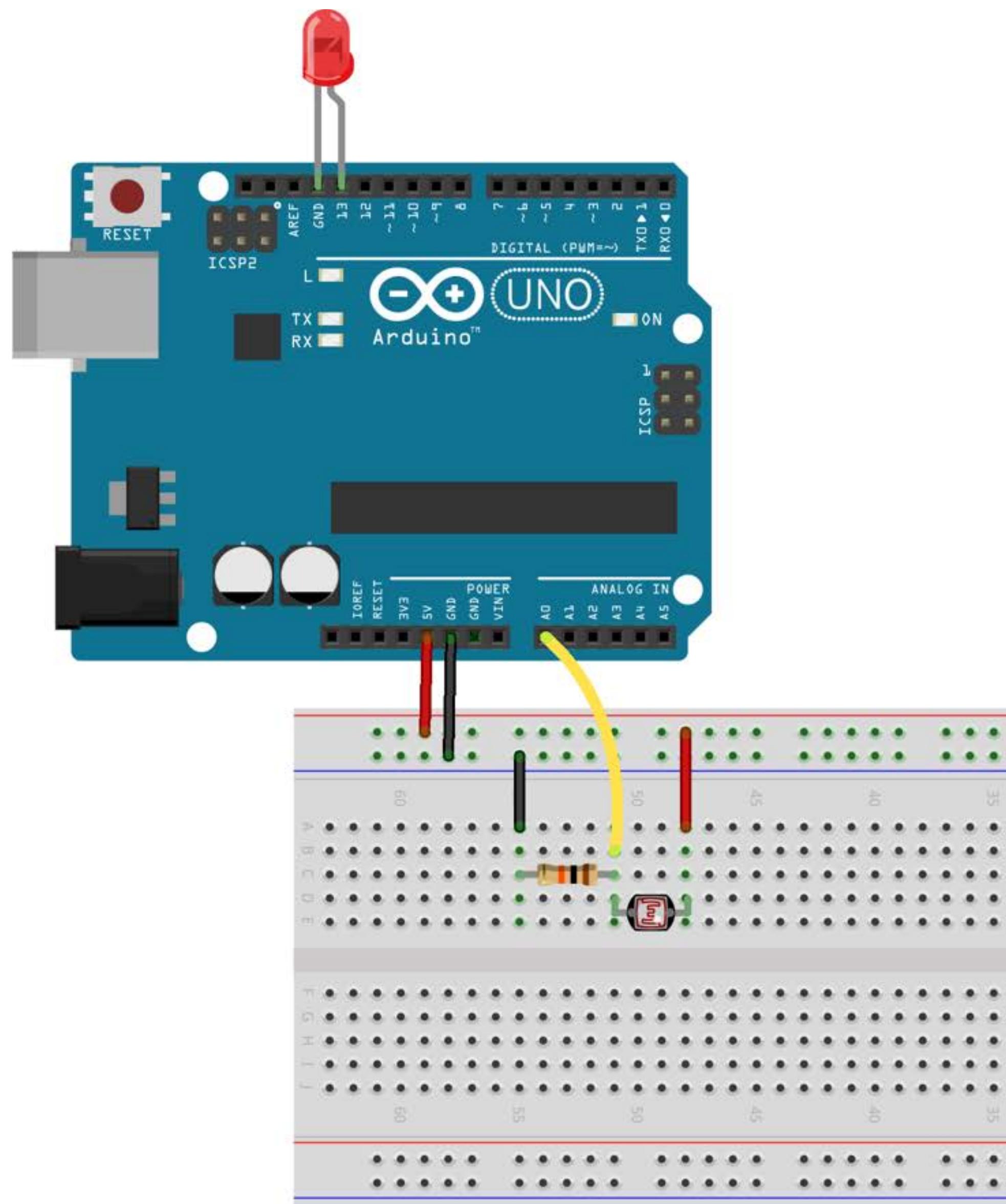
Tiny AVR Programmer

?

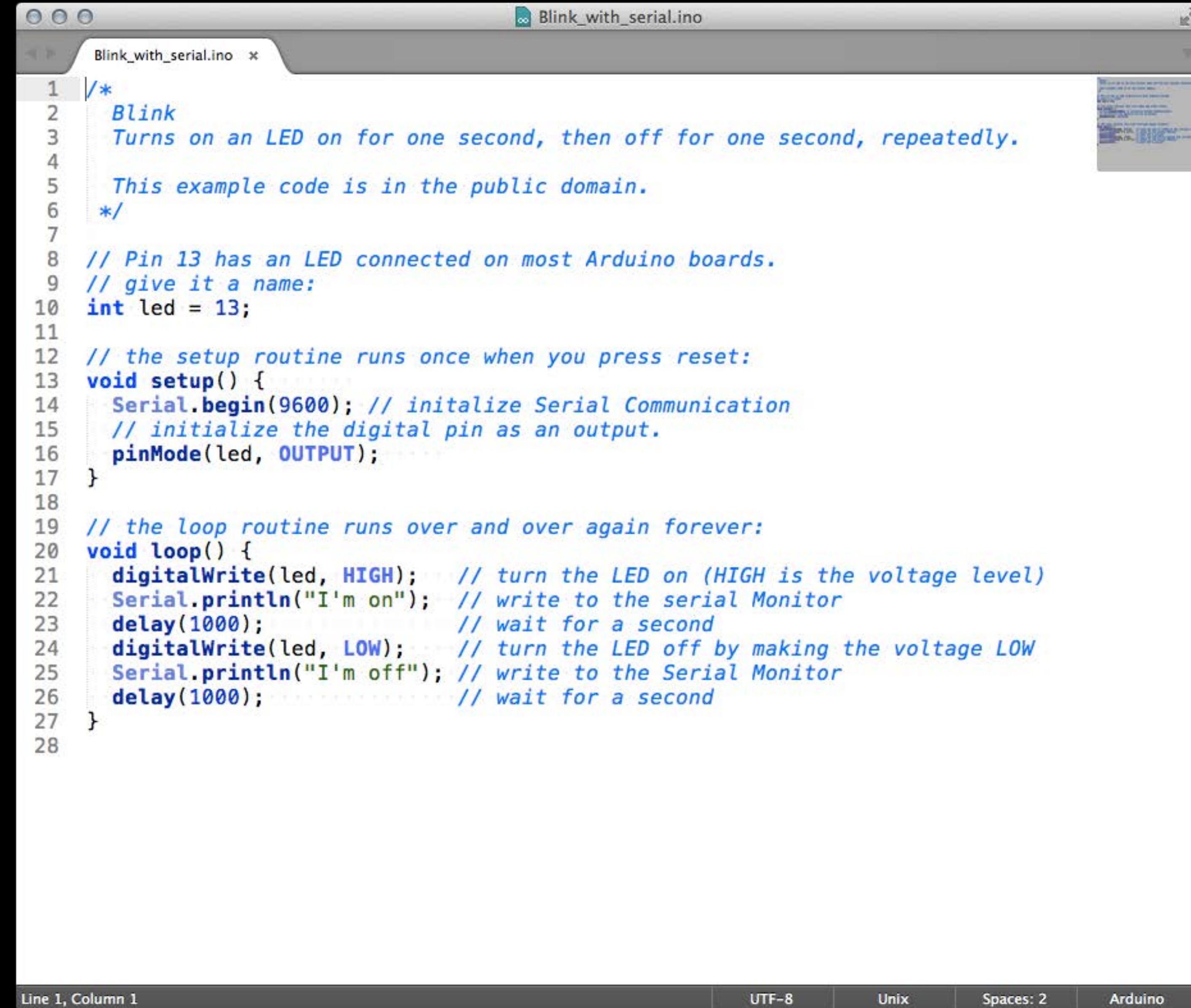
hands on

computer 2 computer communication

github.com/fabiantheblind/c2c



fritzing layout



The screenshot shows the Arduino IDE interface with the file "Blink_with_serial.ino" open. The code is a modified version of the classic Blink example, which now includes serial communication. The code is as follows:

```
1  /*
2   * Blink
3   * Turns on an LED on for one second, then off for one second, repeatedly.
4   *
5   * This example code is in the public domain.
6   */
7
8 // Pin 13 has an LED connected on most Arduino boards.
9 // give it a name:
10 int led = 13;
11
12 // the setup routine runs once when you press reset:
13 void setup() {
14     Serial.begin(9600); // initialize Serial Communication
15     // initialize the digital pin as an output.
16     pinMode(led, OUTPUT);
17 }
18
19 // the loop routine runs over and over again forever:
20 void loop() {
21     digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
22     Serial.println("I'm on");  // write to the serial Monitor
23     delay(1000);              // wait for a second
24     digitalWrite(led, LOW);   // turn the LED off by making the voltage LOW
25     Serial.println("I'm off"); // write to the Serial Monitor
26     delay(1000);              // wait for a second
27 }
28
```

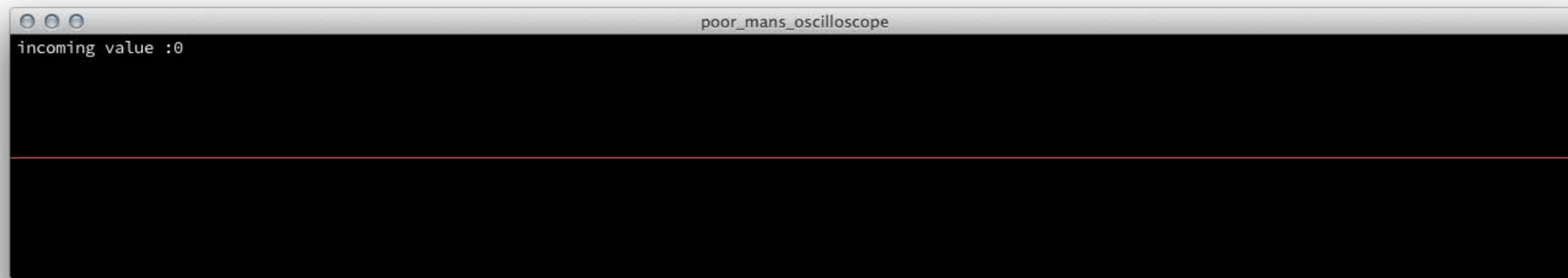
The status bar at the bottom of the IDE shows "Line 1, Column 1" and other settings like "UTF-8", "Unix", "Spaces: 2", and "Arduino".

Processing

list_serial_ports.pde

Arduino → Processing
c2c_outgoing.ino → read_from_serial.pde

Arduino ➔ Processing
c2c_incoming.ino ➔ write_to_serial.pde



poor mans oscilloscope

Arduino → Processing
[poor_mans_oscilloscope.ino](#) → [poor_mans_oscilloscope.pde](#)

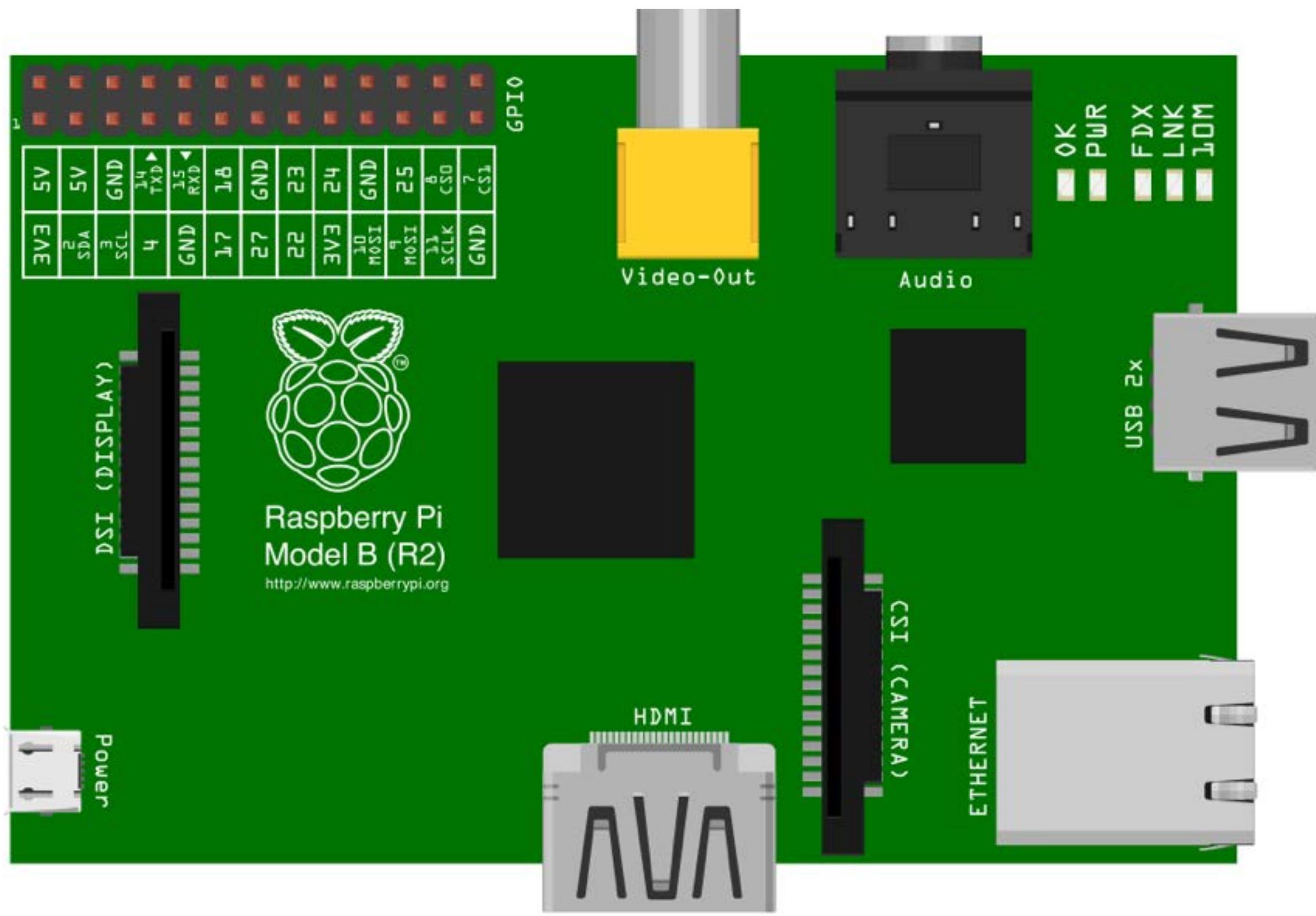
SSH

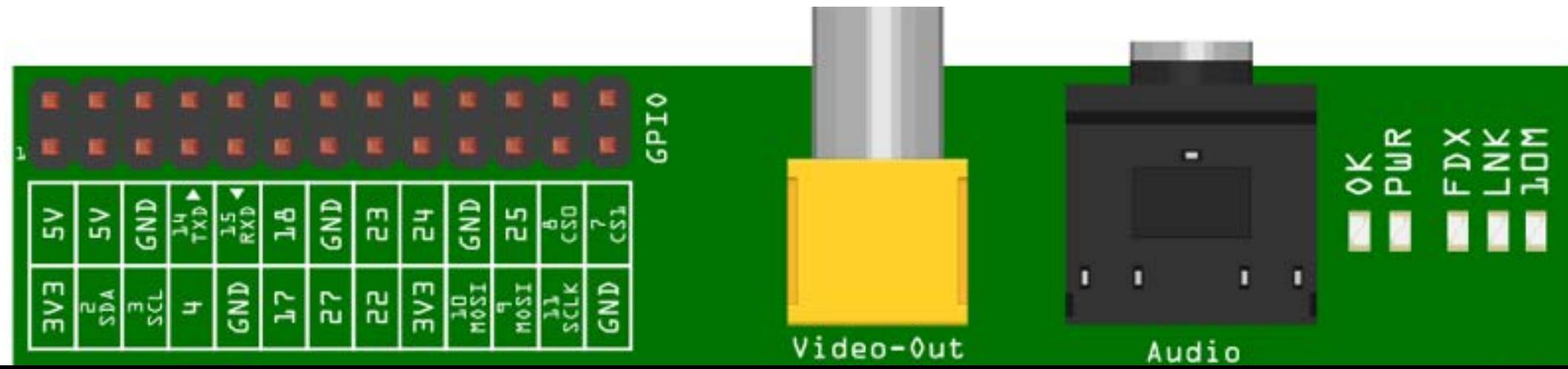
(Secure Shell)

3. Default (bash)

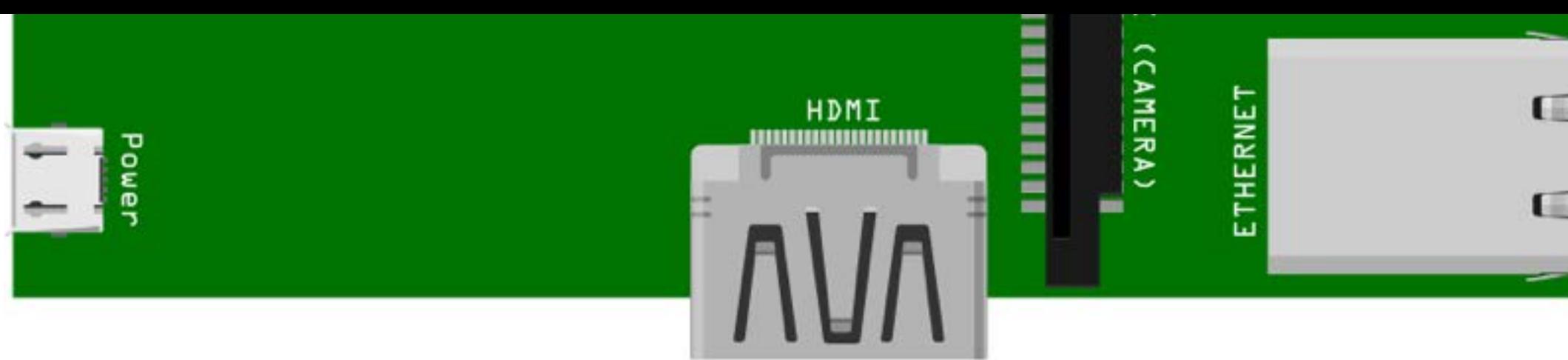
```
474 git commit -a -m "initial local commit"
475 git pull origin master
476 git pull origin master
477 git pull origin master
478 subl README.md
479 git commit -a -m "fix conflict"
480 git push origin master
481 ssh fhp
482 ssh fhp
483 cd /Users/icke/Documents/add-to-pinboard-with-nodejs; clear; pwd
484 ls
485 node index.js
486 ifconfig | grep broadcast
487 arp -a
488 for ip in $(seq 1 254); do ping -c 1 10.0.1.$ip -o ConnectTimeout=5; [ $? -eq 0 ] && echo "10.0.1.$ip UP" || : ; done
489 for ip in $(seq 1 254); do ping -c 1 10.0.1.$ip -o -t 5; [ $? -eq 0 ] && echo "10.0.1.$ip UP" || : ; done
490 clear
491 for ip in $(seq 1 254); do ping -c 1 10.0.1.$ip -o -t 5; [ $? -eq 0 ] && echo "10.0.1.$ip UP" || : ; done
492 brew install nmap
493 nmap -h
494 ssh pi@10.0.1.8
495 ssh pi@10.0.1.2
496 ssh pi@10.0.1.3
497 nmap -sP 10.0.1.1/255
498 nmap -sP 10.0.1.1/24
499 ssh pi@10.0.1.23
500 ssh pi@10.0.1.1
501 history
Fabians-MacBook-Air:~ icke$
```

[WIN & MAC]: arp -a





[MAC]: ssh bah@10.0.1.9



username: bah password: raspberry

[WIN & MAC]: cd www

[MAC]: echo "I wuz here" > [FILENAME].txt

[MAC]: echo "Me 2" >> [FILENAME].txt

[http://10.0.1.9/\[FILENAME\].txt](http://10.0.1.9/[FILENAME].txt)

[MAC]: nano [FILENAME].txt

[MAC]: vi [FILENAME].txt

[MAC]: vim [FILENAME].txt

[MAC]: vimtutor

Hallo, wir sind Uberspace – Uberspace.de

DU BIST SCHON UBERNAUT? EINLOGGEN
Steig ein.

HOME * TECHNIK * PREISE * SUPPORT * HAUSORDNUNG * UBERNAUT WERDEN

UBERSPACE
HOSTING ON ASTEROIDS

★★★

HALLO, WIR SIND UBERSPACE

★★★

Uberspace.de ist deine Plattform für den Betrieb von Websites und Mail. Sie bietet ungewöhnlich viele Möglichkeiten, engagierte Unterstützung durch erfahrene Linux-Admins – und das Beste: den Preis bestimmt du!

DU BESTIMMST DEN PREIS.

Bei Uberspace.de verfolgen wir ein für viele vielleicht erstmal ungewöhnliches Konzept: Du suchst dir deinen Preis selbst aus. Wir legen lediglich einen Mindestpreis von 1 Euro pro Monat fest. Unter dem geht's einfach nicht. Bei anderen Providern wirst du dafür kaum mehr als eine Art Web-Visitenkarte bekommen. Nicht so bei uns!

★ ÜBERZEUGE DICH SELBST

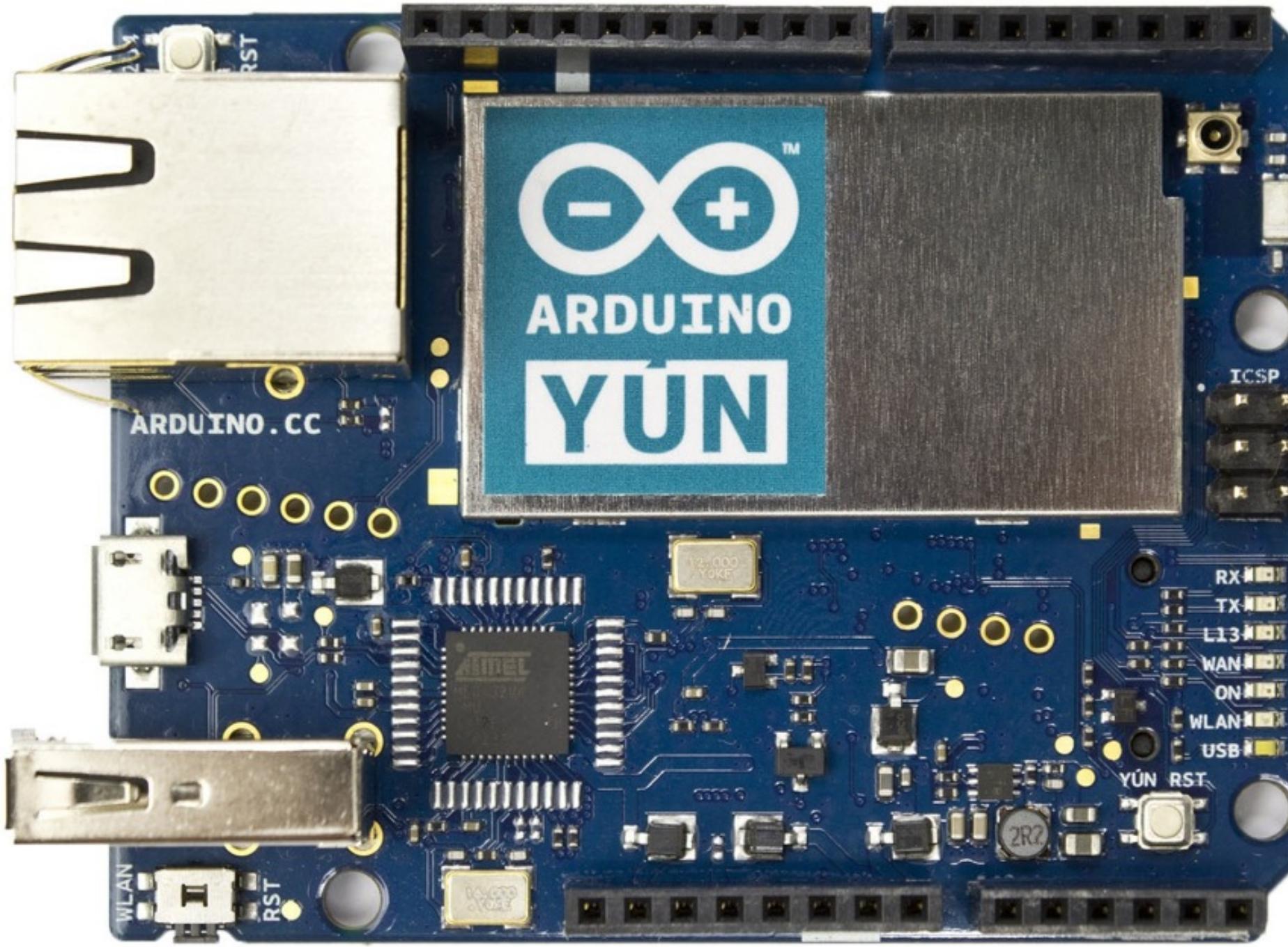
UNGEWOHNT VIELFÄLTIG.

Machen wir's doch erstmal kurz: SSH. Perl. PHP. Python. Ruby. node.js. Erlang. Lua. Compiler. FastCGI. MySQL. CouchDB. MongoDB. Cronjobs. HTTPS. IMAP. SMTP. Webmail. qmail. vmailmgr. maildrop. SpamAssassin. ezmlm-idx. DSPAM. ~/service. runwhen. Eigene Logs. Backups. 10 GB Plattenplatz. Und das ist nur der Anfang.

★ ERFAHRE JEDES DETAIL



uberspace.de



Arduino :: Temboo

We're Hiring

LIBRARY DOWNLOAD SUPPORT SIGN UP LOG IN

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

Arduino Yun + temboo.com

That's all Folks!