

FONDAMENTI DI ARDUINO

LEZIONE 1: LE BASI

8 novembre 2016

presso POLIFACTORY

il makerspace del Politecnico di Milano

PROGRAMMA DEL CORSO

8/11	Le basi, corrente elettrica, led, resistenze
10/11	Ingressi analogici, PWM, debouncing
15/11	Display + seriale + wifi
17/11	Supporto al progetto

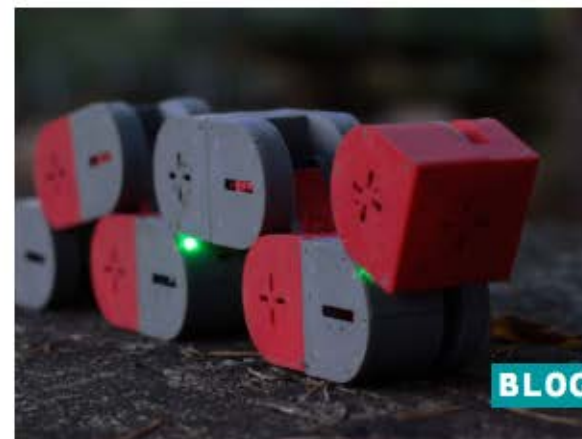
CHE COS'È ARDUINO?



[BUY AN ARDUINO](#) 

[LEARN ARDUINO](#) 

[SUPPORT ARDUINO](#) 



DTTO IS A 3D-PRINTED, SELF-CONFIGURABLE MODULAR ROBOT



[GENUINO SWAGS,](#)
[SHOW OFF YOUR](#)

CHE COS'È ARDUINO?

Arduino è una piattaforma di prototipazione elettronica open-source basata su un di un hardware e un software flessibile, facile da usare.

È stato progettato per artisti, designer, hobbysti e per chiunque sia interessato alla creazione di oggetti e ambienti interattivi.

CHE COS'È ARDUINO?

HARDWARE

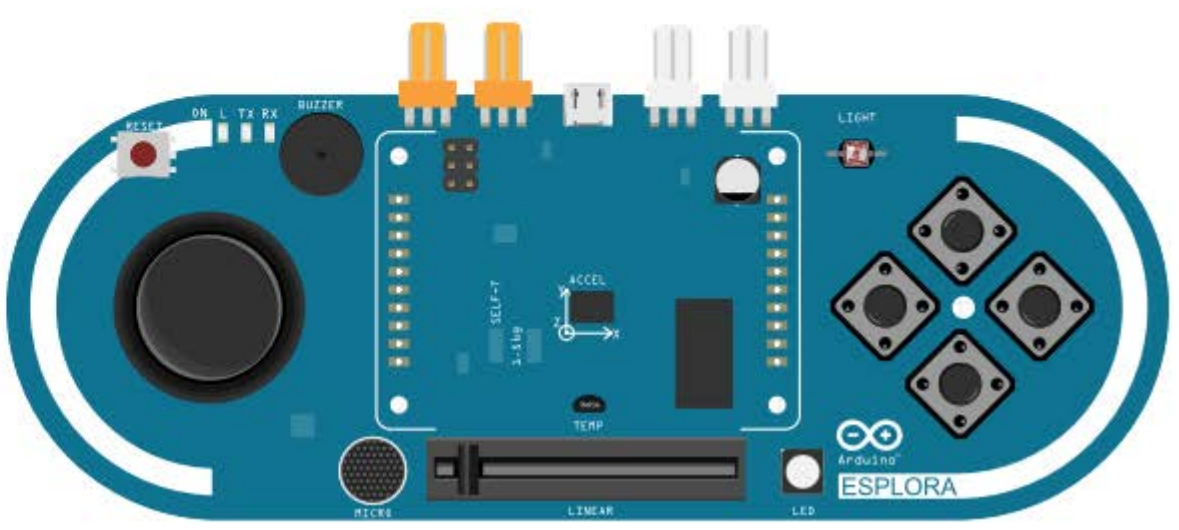
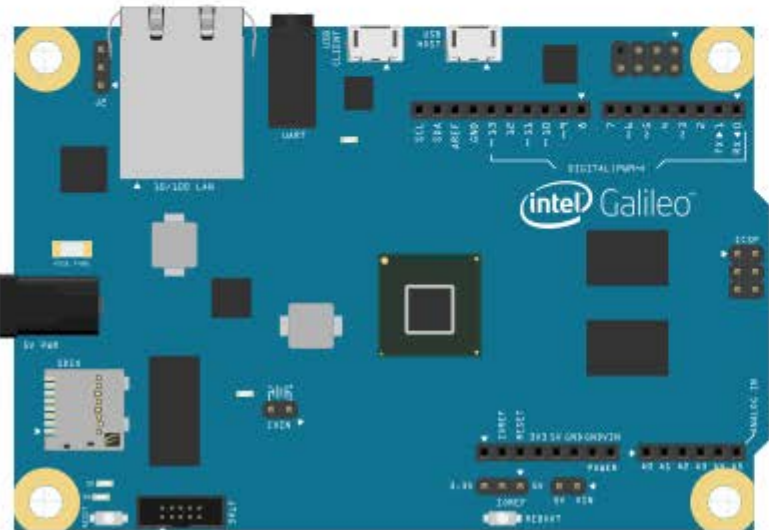
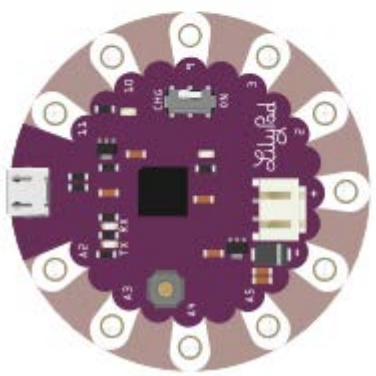
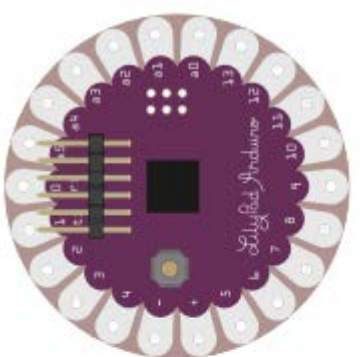
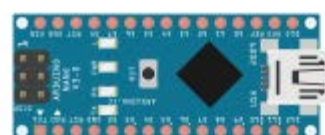
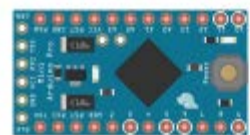
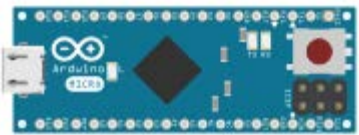
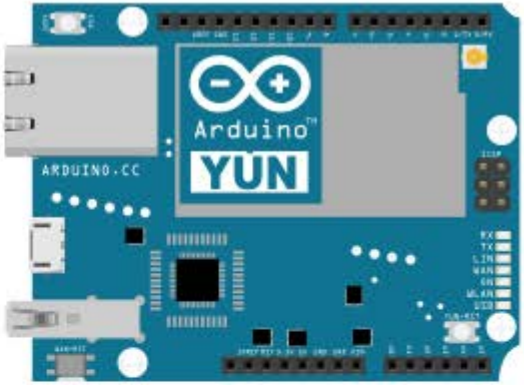
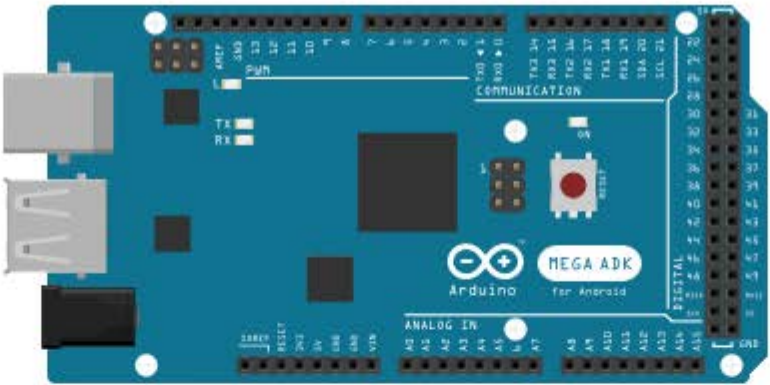
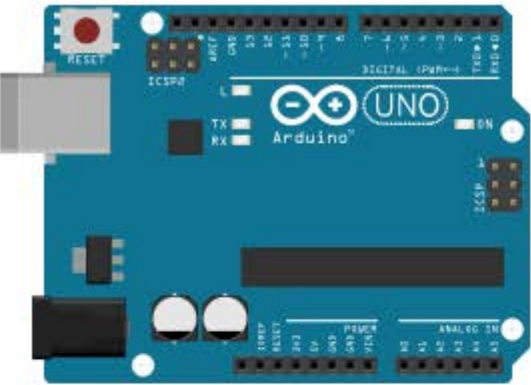
+

COMMUNITY

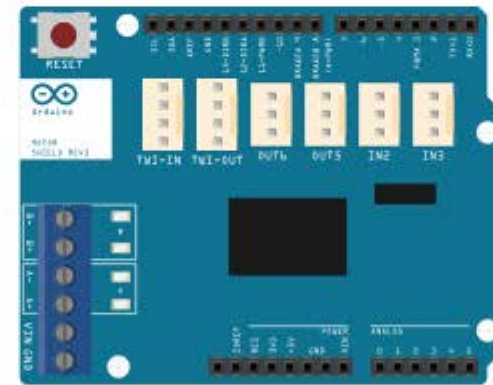
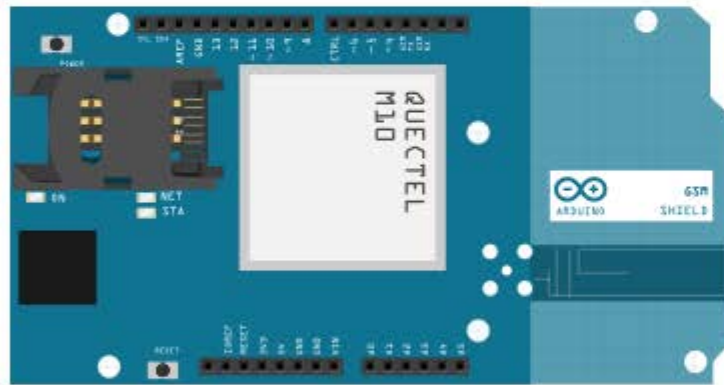
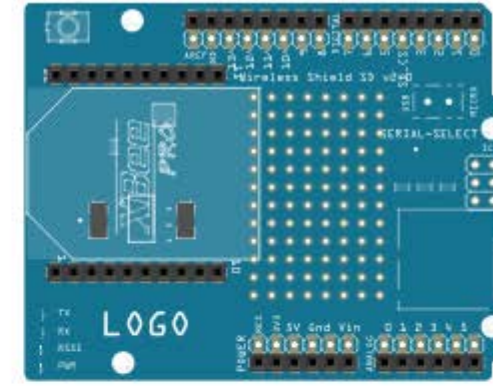
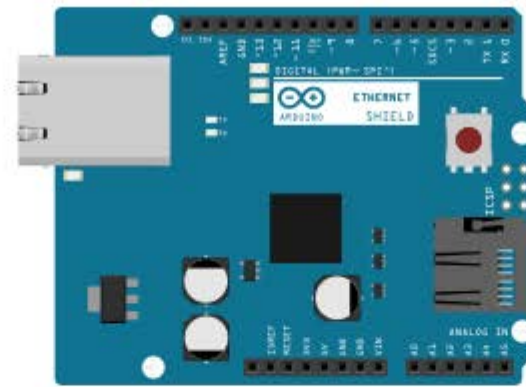
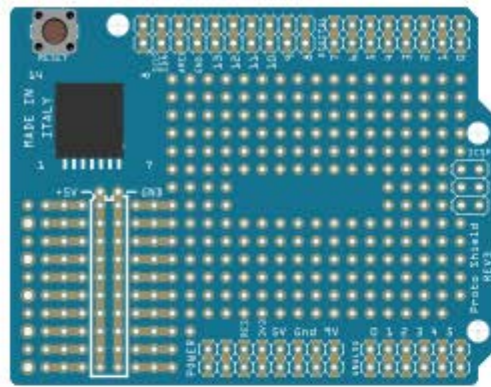
+

SOFTWARE

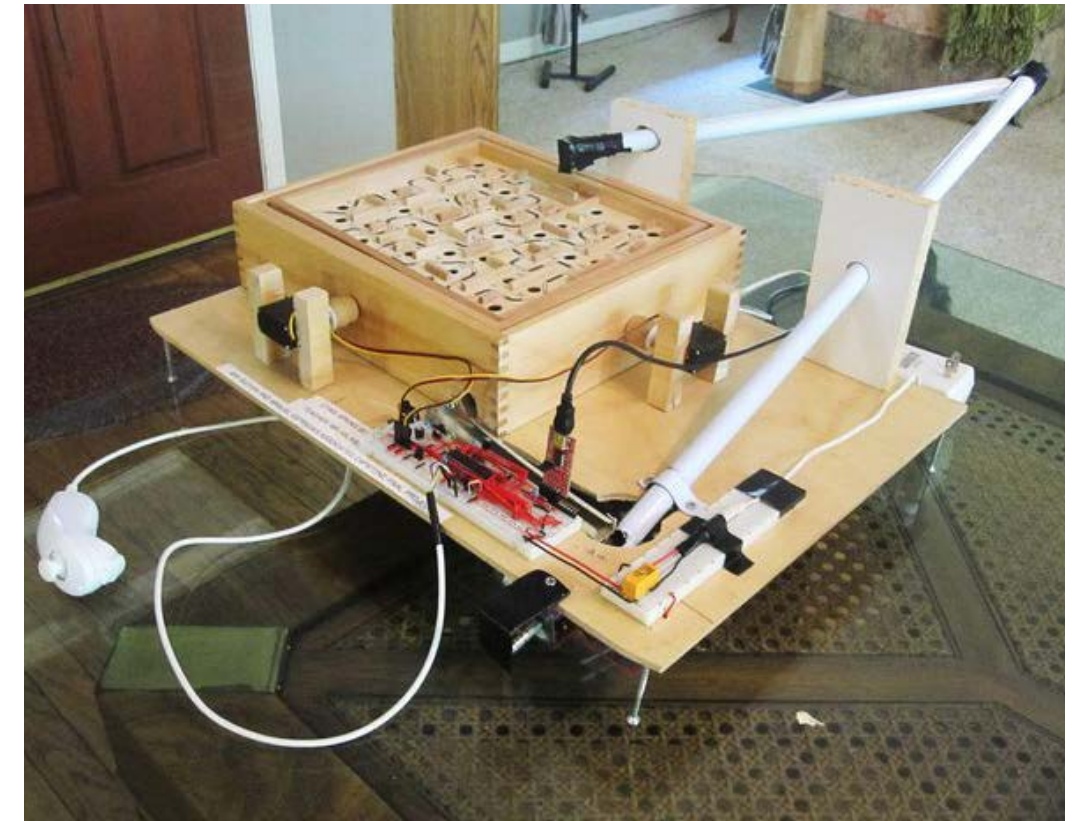
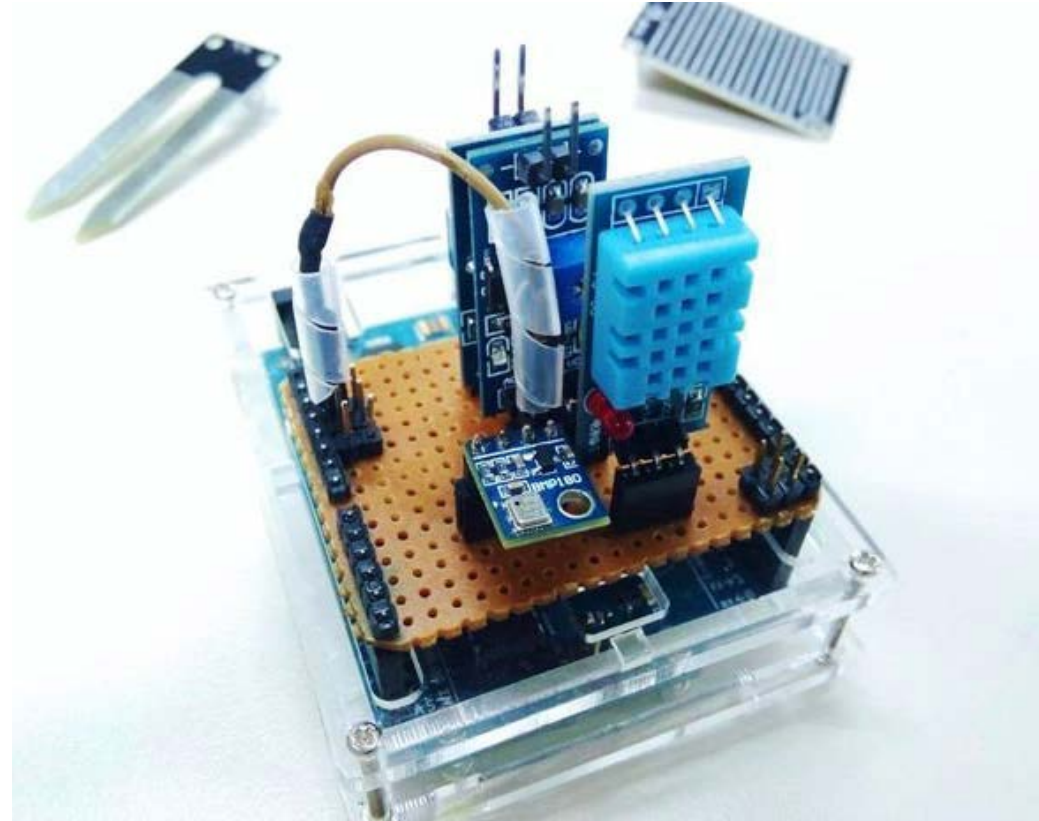
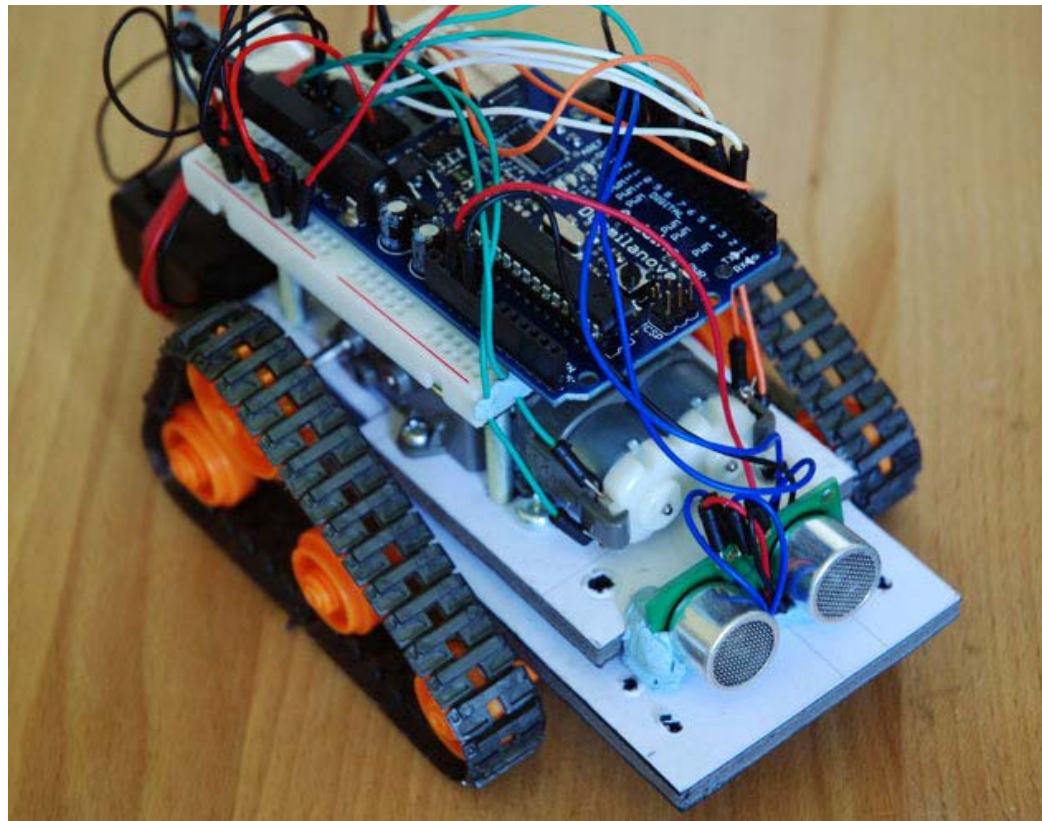
HARDWARE: BOARD



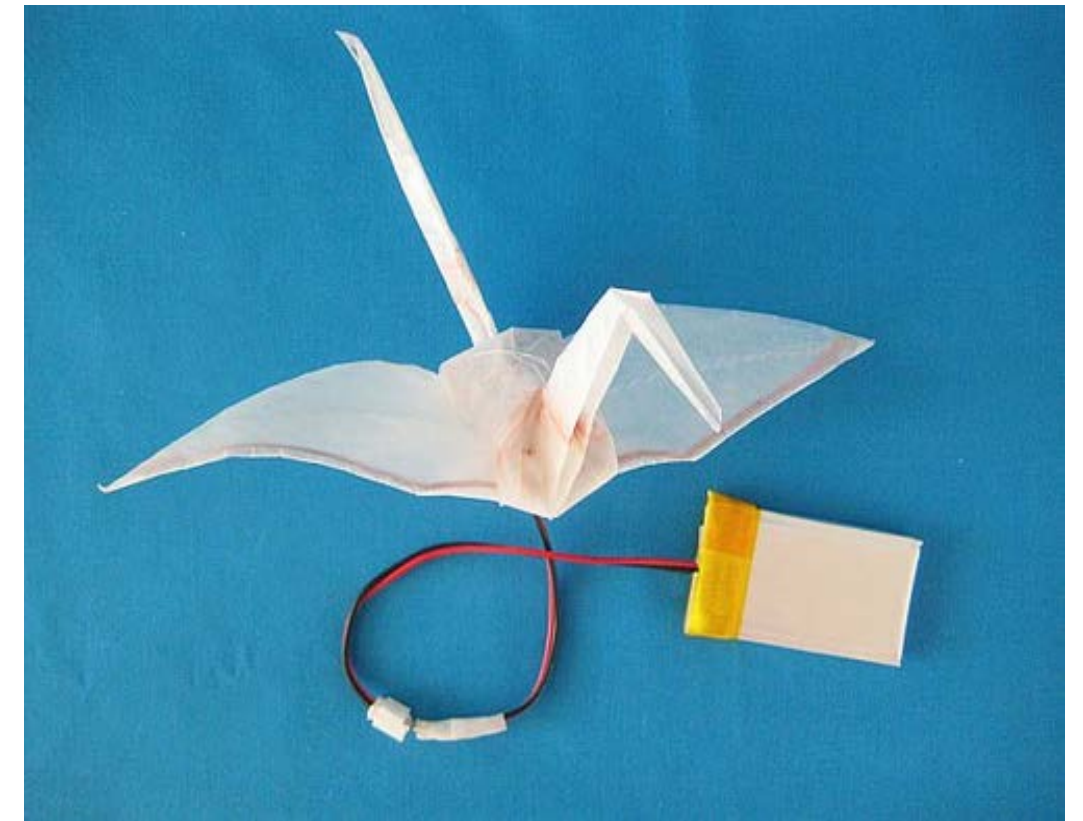
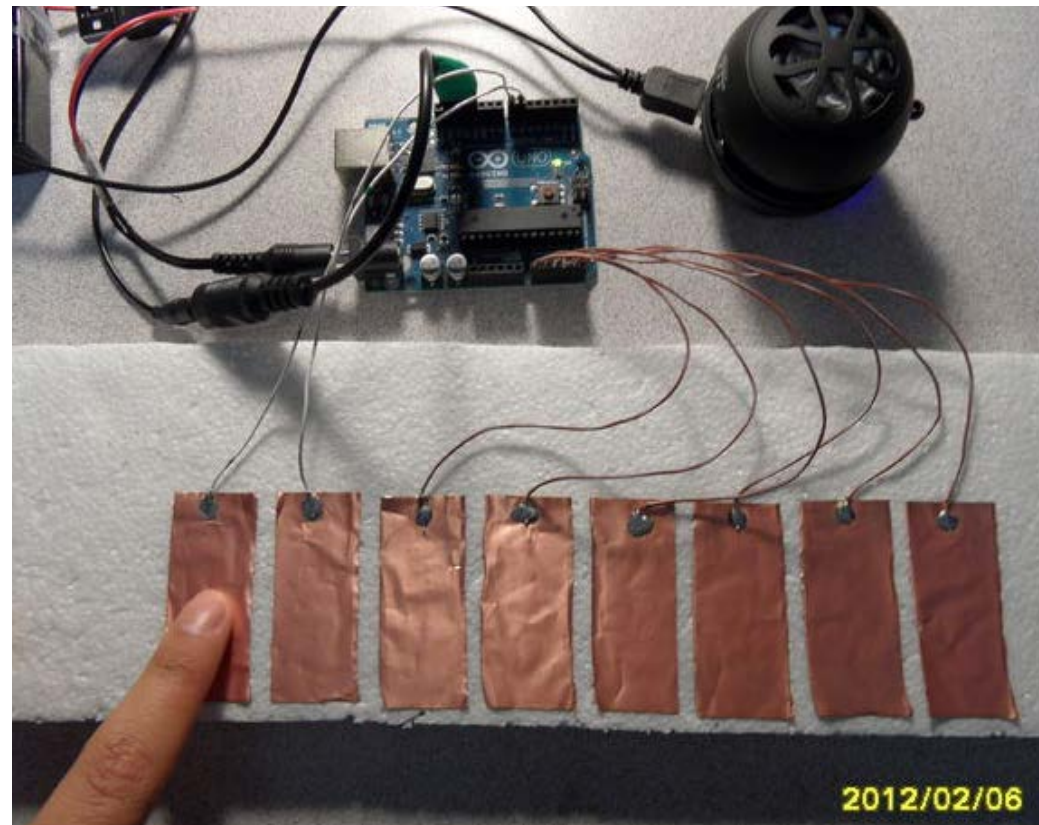
HARDWARE: SHIELD



COMMUNITY



COMMUNITY



SOFTWARE



ARDUINO
USA ONLY

Genuino
OUTSIDE USA

Home Buy **Download** Products ▾ Learning ▾ Forum Support ▾ Blog

LOG IN SIGN UP

DOWNLOAD

ENGLISH ▾

Download the Arduino Software



ARDUINO 1.6.12

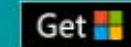
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app (Microsoft Store)



Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

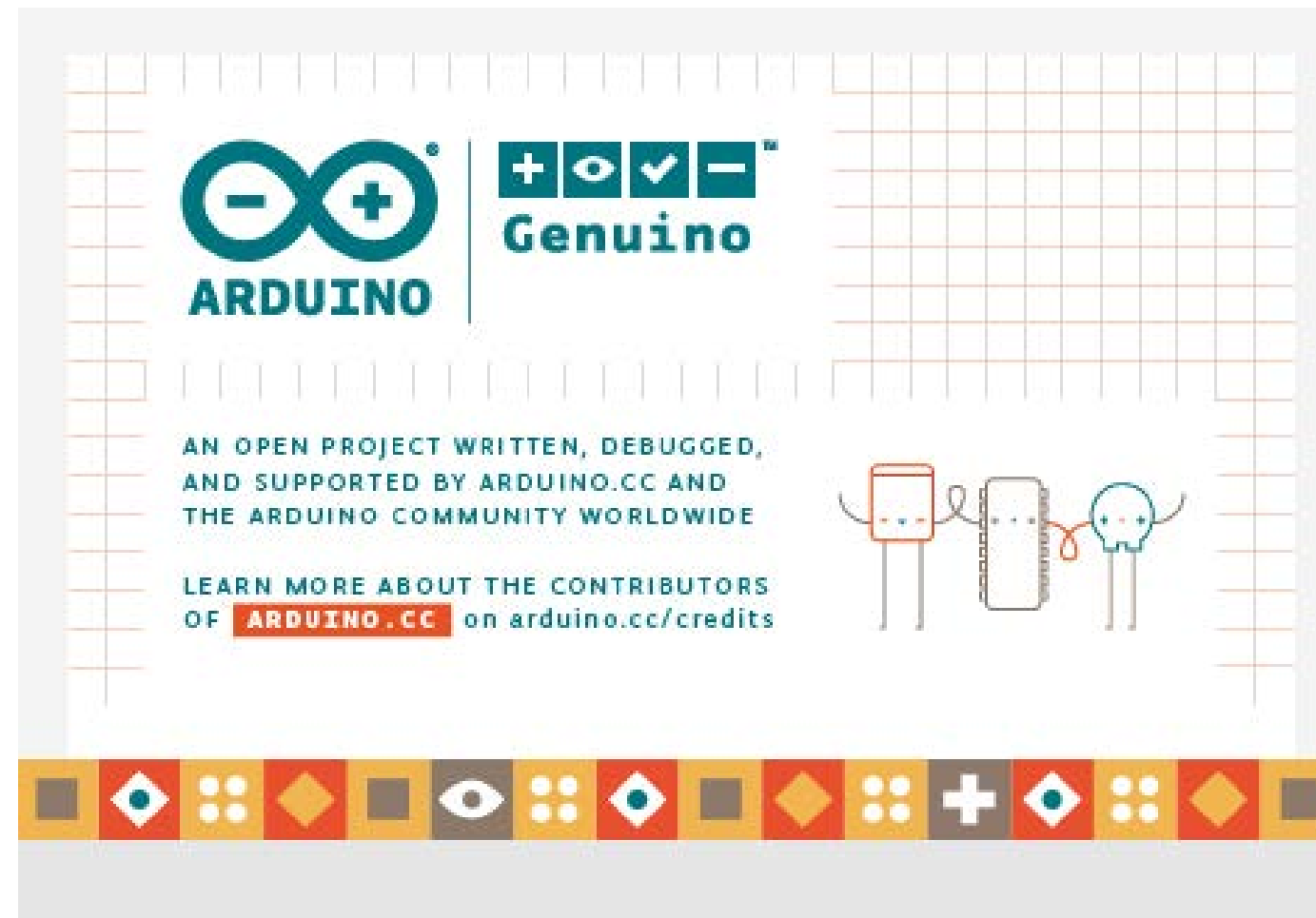
Linux ARM (experimental)

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

SOFTWARE

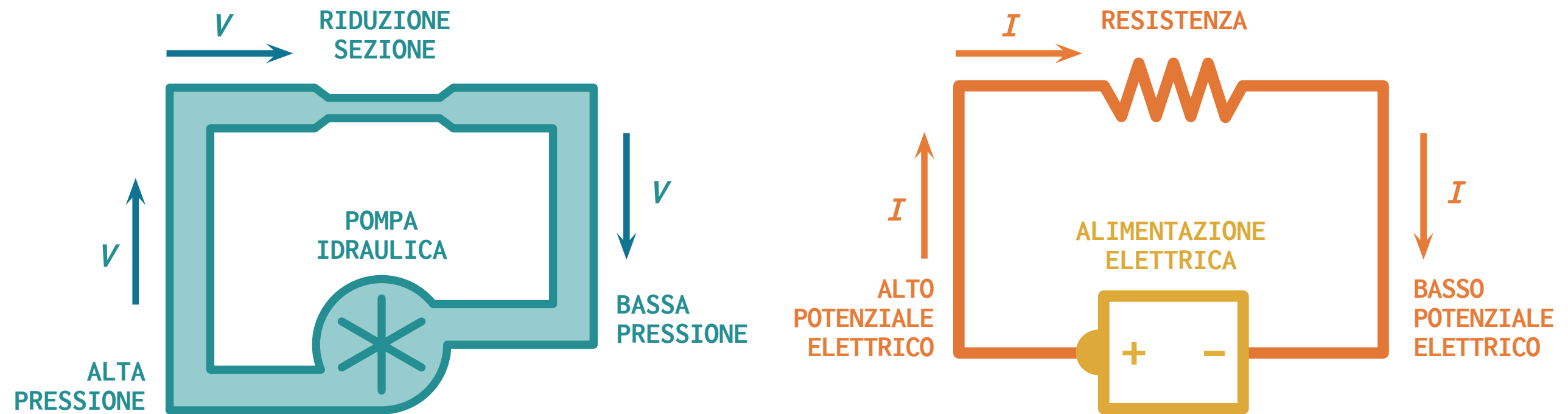


**COME CONQUISTARE
IL MONDO...**

...0 FARLO ESPLODERE!

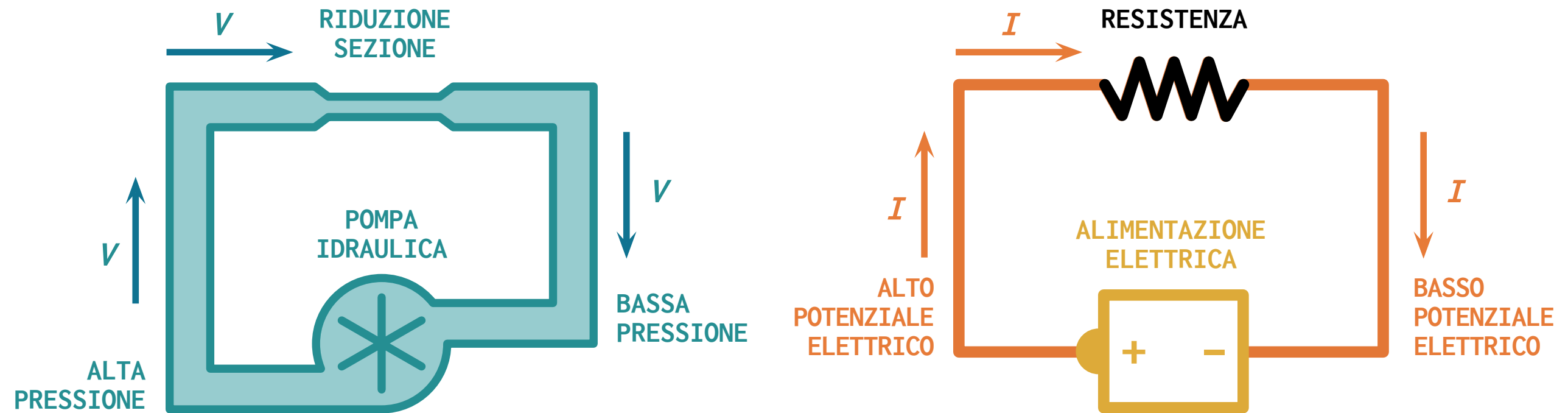


VOLTAGGIO. CORRENTE. RESISTENZA.



LA METAFORA ACQUATICA:
se immaginiamo di avere un circuito chiuso
d'acqua, quest'ultima rappresenta gli elettroni.

VOLTAGGIO. CORRENTE. RESISTENZA.



- (I)** La velocità del flusso d'acqua rappresenta la **corrente**
- (V)** La differenza di pressione fra due punti rappresenta la **tensione** (o differenza di potenziale) tra due punti
- (R)** La variazione di sezione è la **resistenza**

LA LEGGE DI OHM

$$V = R * I \qquad I = \frac{V}{R}$$

Moltiplicando la resistenza per l'intensità di corrente elettrica, si ottiene la differenza di potenziale elettrico ai capi di un conduttore.

Dividendo la differenza di potenziale con l'intensità di corrente, si ottiene il valore della resistenza elettrica.

LA LEGGE DI OHM

$$V = R * I \qquad I = \frac{V}{R}$$

A **parità di differenza di potenziale (V)**:

- * se aumento la resistenza (R) diminuirà la corrente (I)
- * se diminuisco la resistenza (R) aumenterà la corrente (I)

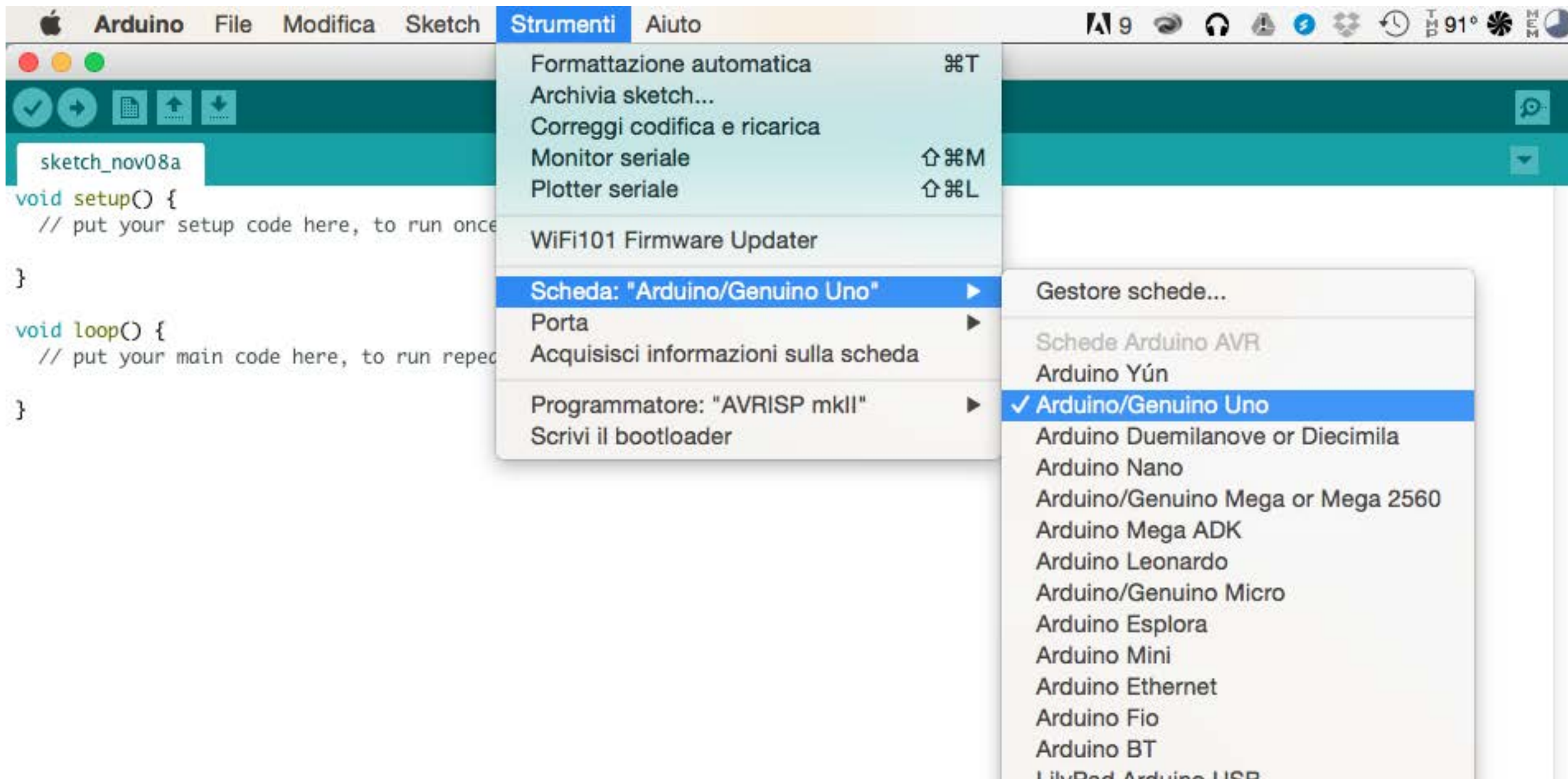
Casi limite:

- * in un CIRCUITO APERTO la resistenza è infinita, quindi non passa corrente
- * se la resistenza è zero si verifica un CORTO CIRCUITO, poichè passa infinita corrente

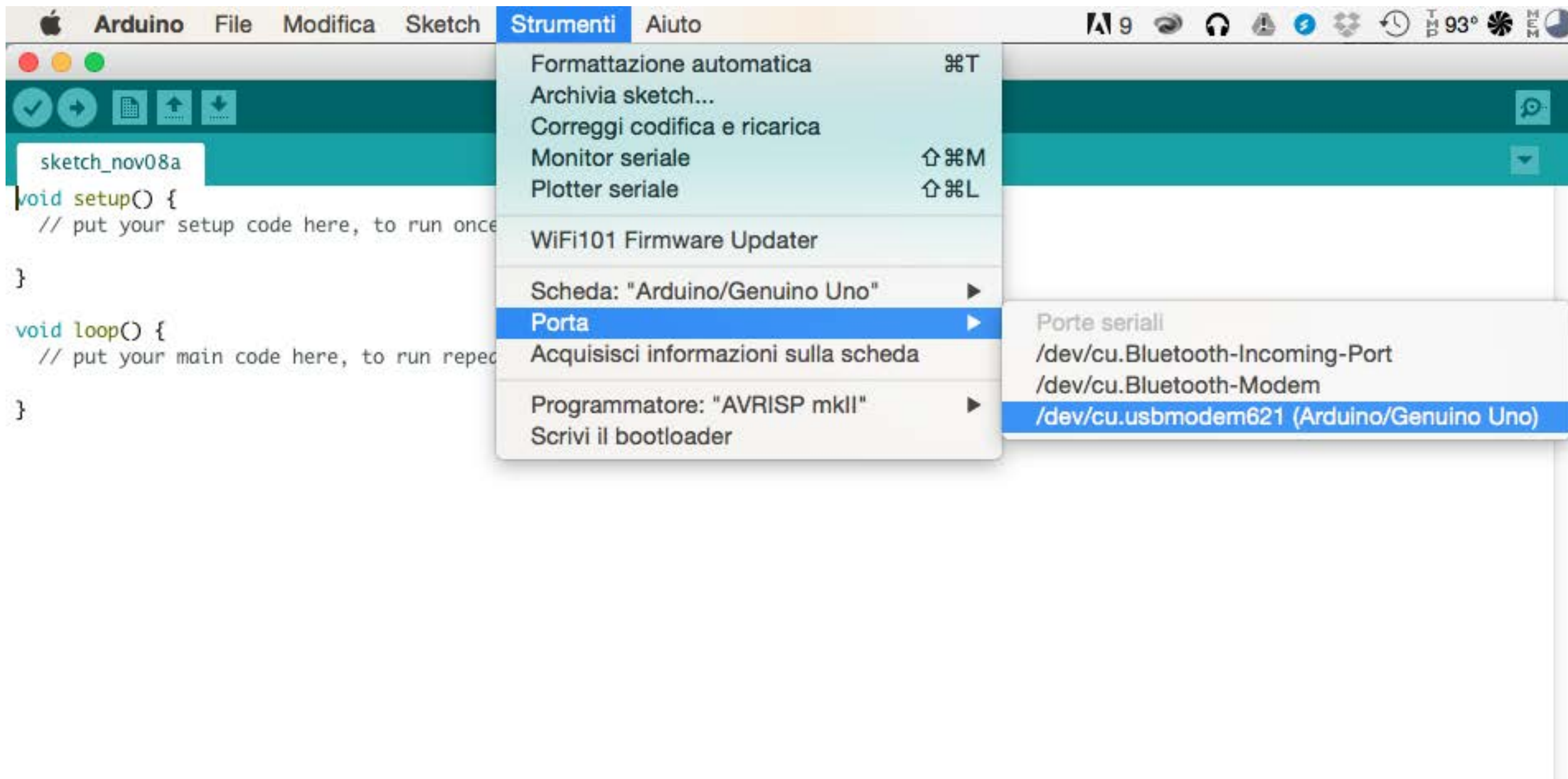
The image features a vibrant, abstract background composed of overlapping, curved bands in shades of teal, orange, and yellow. In the center, a teal rectangular box with rounded corners contains the text "SI PARTE!" in white, bold, uppercase letters. The overall design is modern and energetic, with a focus on geometric shapes and a rich color palette.

SI PARTE!

COLLEGAMENTO DI ARDUINO: SELEZIONE SCHEDA



COLLEGAMENTO DI ARDUINO: SELEZIONE PORTA



INTERFACCIA DI ARDUINO



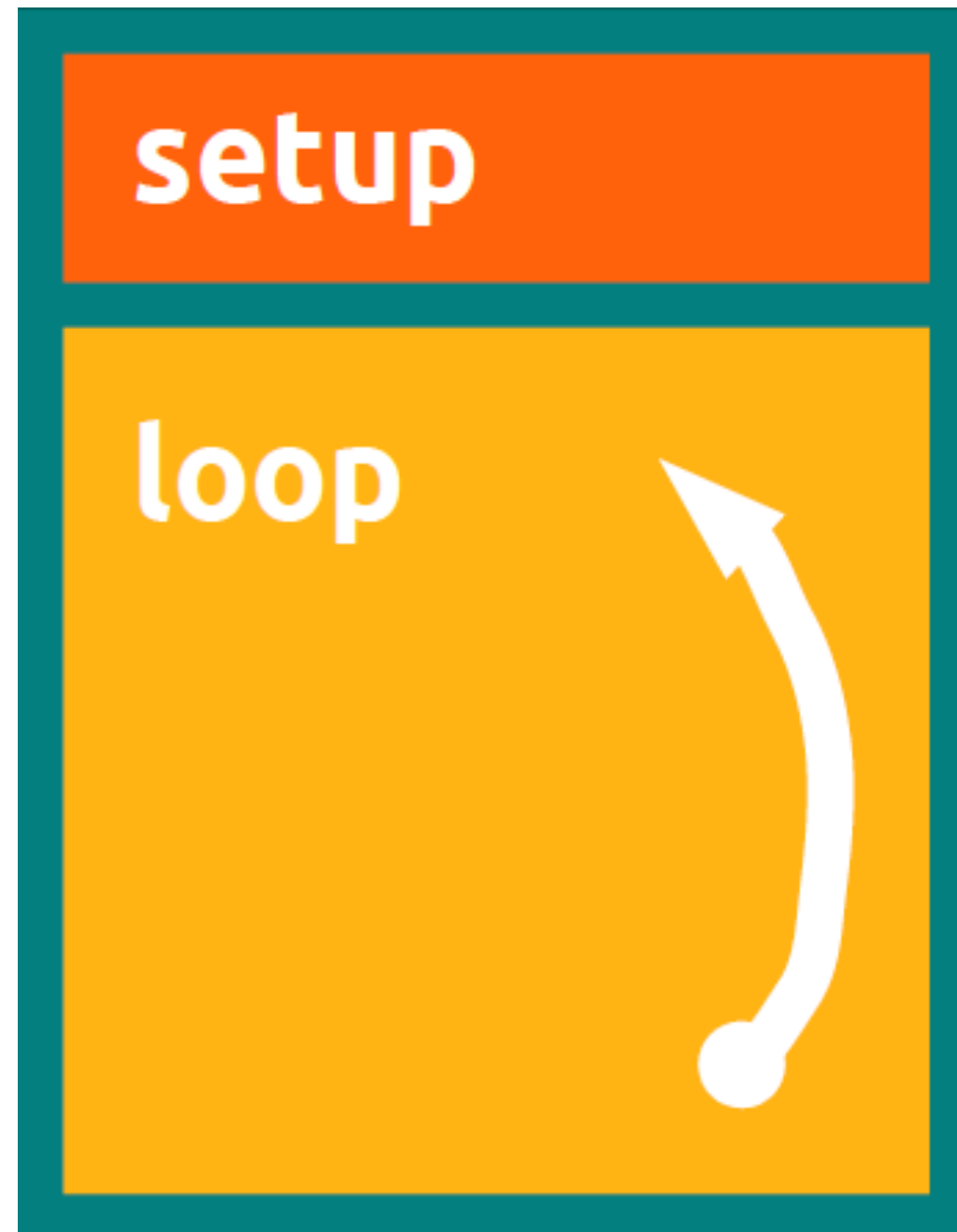
The image shows a screenshot of the Arduino IDE interface. The window title is "sketch_nov08a | Arduino 1.6.12". The interface includes a toolbar with icons for check, run, file, upload, and download. The main editor area contains the following code:

```
sketch_nov08a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

At the bottom of the window, the status bar displays "Arduino/Genuino Uno su /dev/cu.usbmodem621".

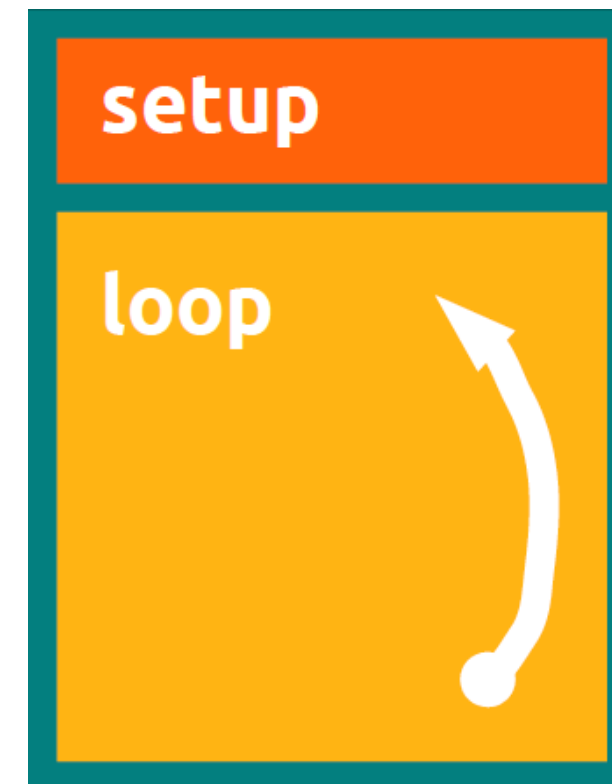
SETUP + LOOP



SETUP + LOOP

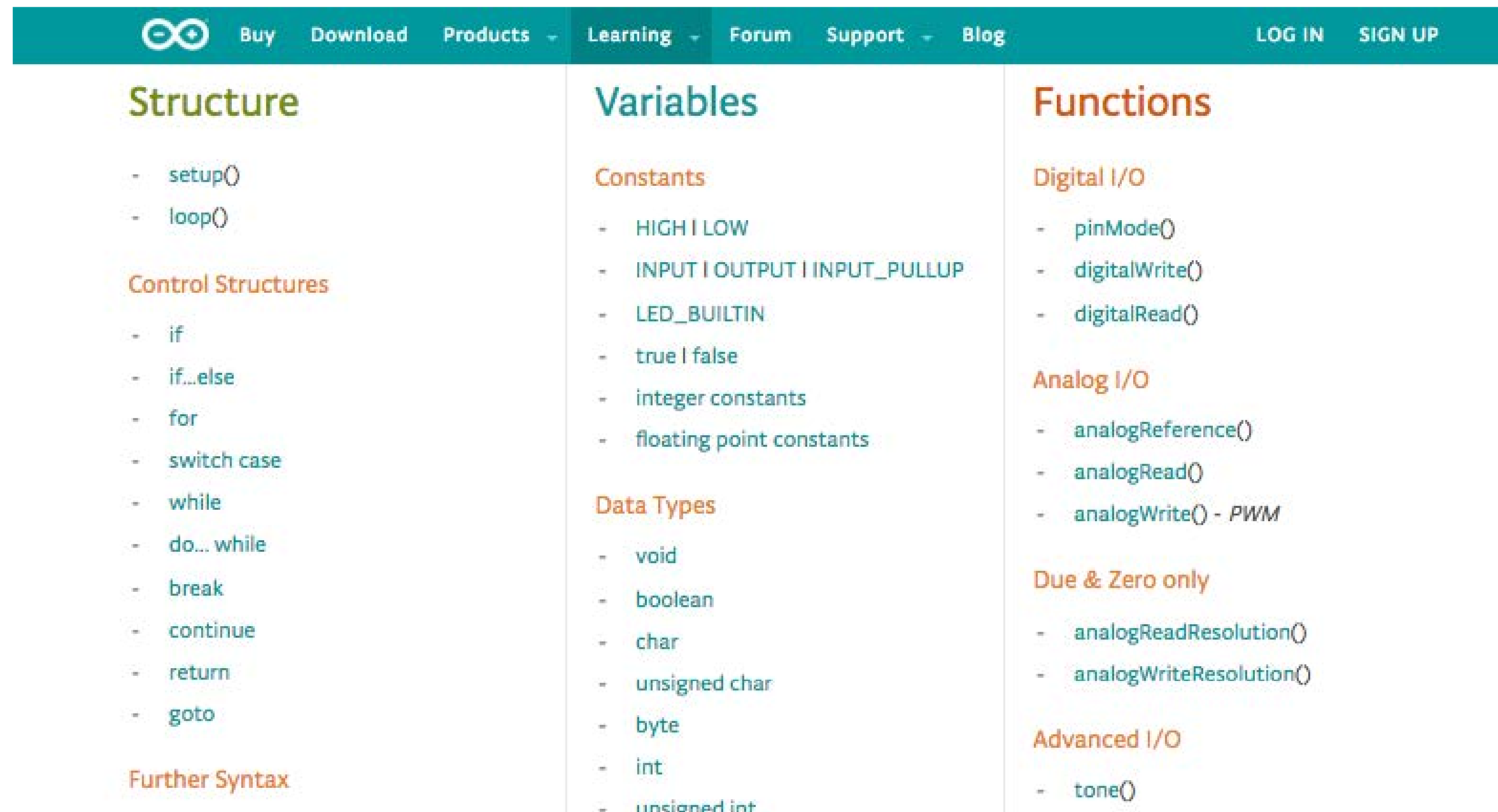
```
sketch_nov08a | Arduino 1.6.12  
sketch_nov08a  
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Arduino/Genuino Uno su /dev/cu.usbmodem621



LINGUAGGIO DI PROGRAMMAZIONE

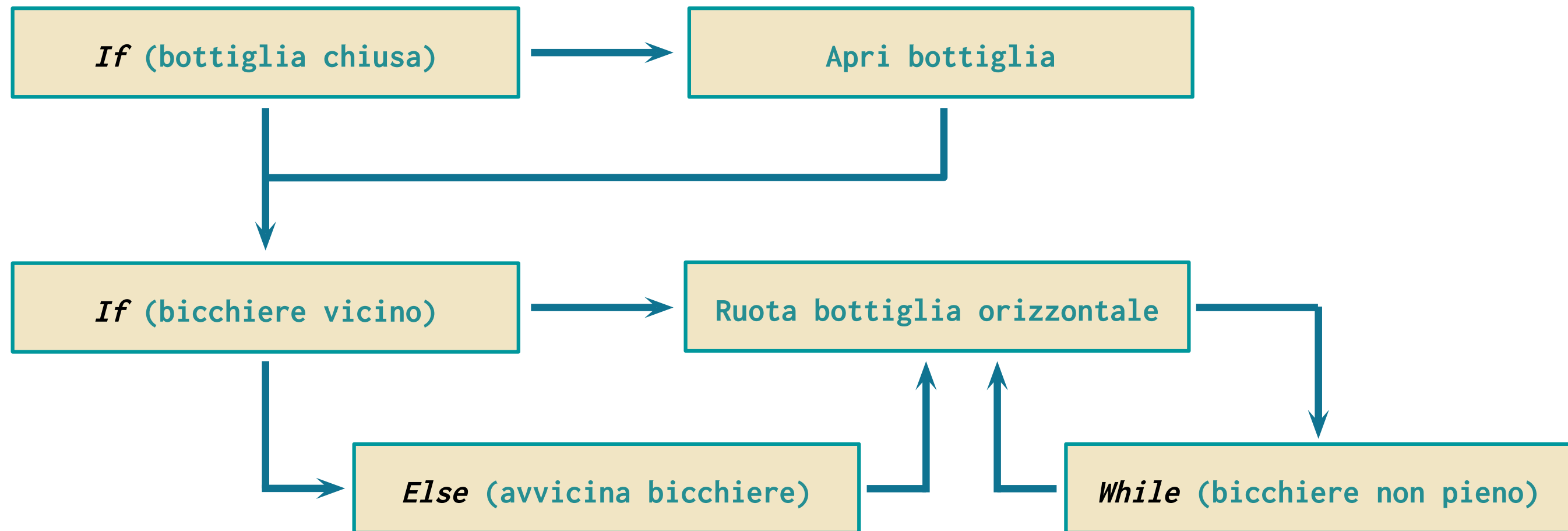
<https://www.arduino.cc/en/Reference/HomePage>



The image shows a screenshot of the Arduino Reference website. At the top is a teal navigation bar with the Arduino logo on the left and links for Buy, Download, Products, Learning, Forum, Support, and Blog. On the right side of the bar are links for LOG IN and SIGN UP. Below the navigation bar, the page is divided into three main columns of content categories:

- Structure**
 - `setup()`
 - `loop()`
- Control Structures**
 - `if`
 - `if...else`
 - `for`
 - `switch case`
 - `while`
 - `do... while`
 - `break`
 - `continue`
 - `return`
 - `goto`
- Further Syntax**
- Variables**
 - Constants**
 - `HIGH` | `LOW`
 - `INPUT` | `OUTPUT` | `INPUT_PULLUP`
 - `LED_BUILTIN`
 - `true` | `false`
 - integer constants
 - floating point constants
 - Data Types**
 - `void`
 - `boolean`
 - `char`
 - `unsigned char`
 - `byte`
 - `int`
 - `unsigned int`
- Functions**
 - Digital I/O**
 - `pinMode()`
 - `digitalWrite()`
 - `digitalRead()`
 - Analog I/O**
 - `analogReference()`
 - `analogRead()`
 - `analogWrite()` - *PWM*
 - Due & Zero only**
 - `analogReadResolution()`
 - `analogWriteResolution()`
 - Advanced I/O**
 - `tone()`

ESEMPIO: COME VERSARSI DA BERE CON ARDUINO



**ESERCIZIO 1:
BLINKG LED**

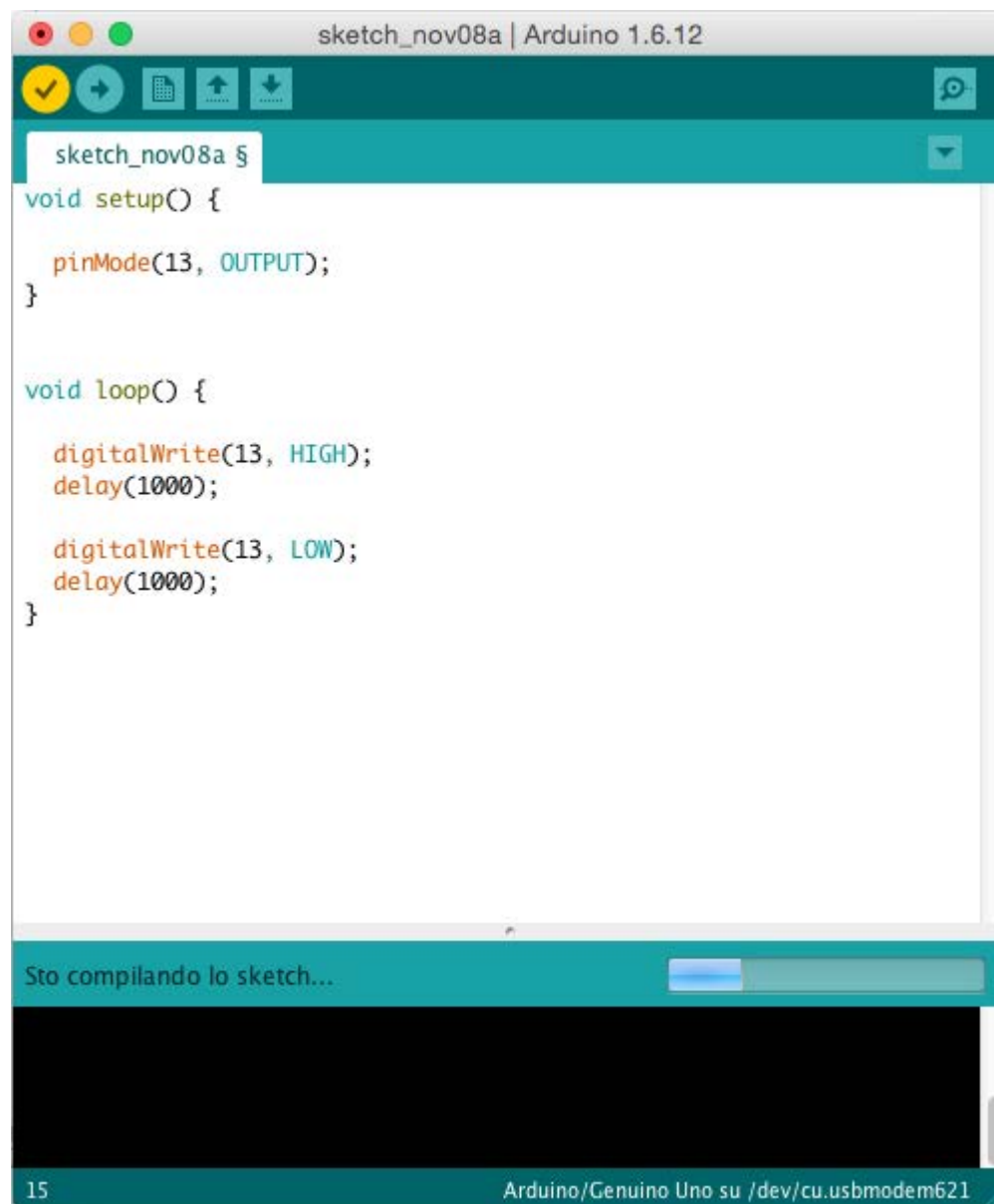
BLINKING LED: SETUP

```
void setup() {  
    pinMode(13, OUTPUT);  
}
```

BLINKING LED: LOOP

```
void loop() {  
  
    digitalWrite(13, HIGH);  
    delay(1000);  
  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

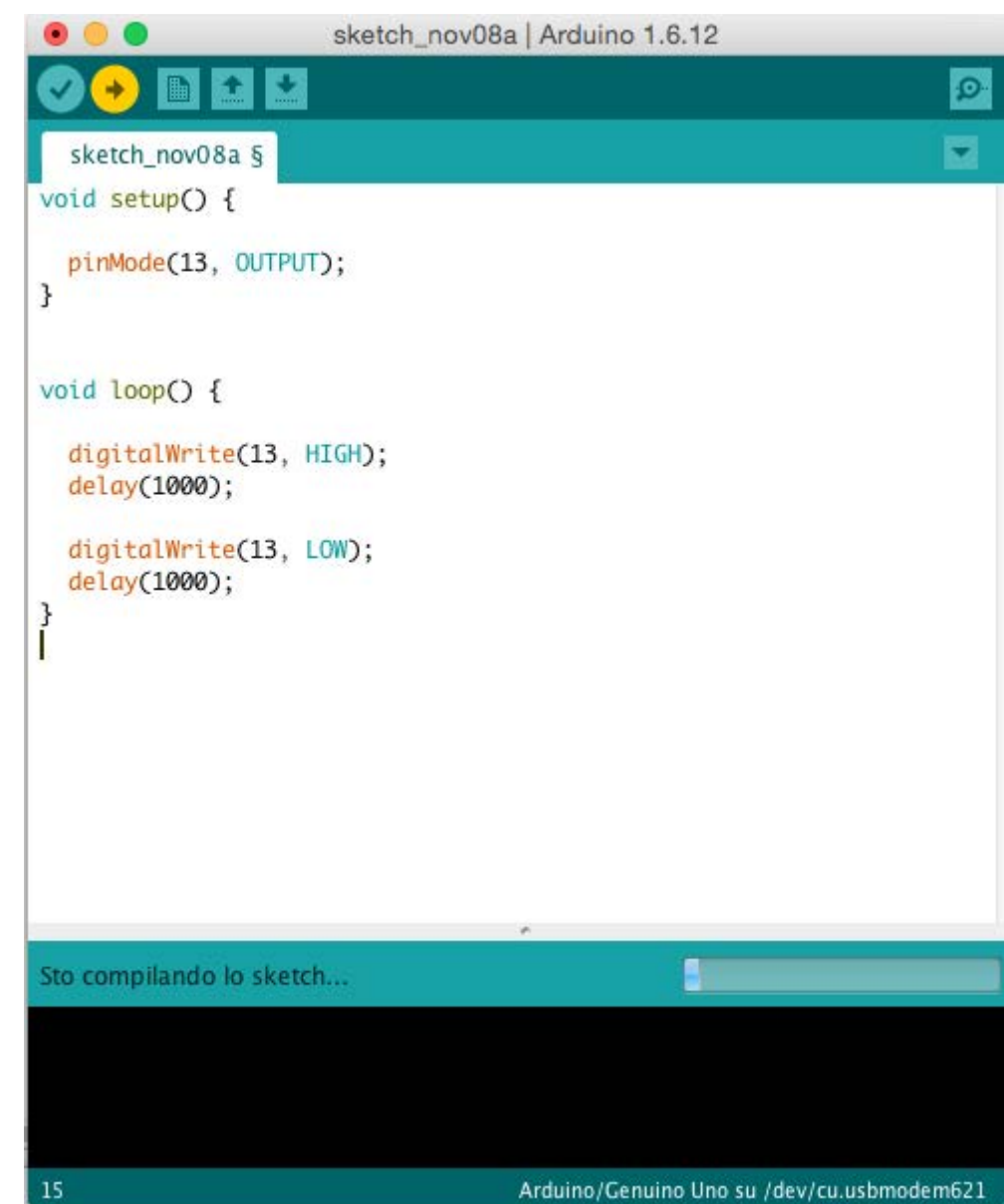
VERIFICA + CARICA



```
sketch_nov08a | Arduino 1.6.12  
sketch_nov08a §  
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

Sto compilando lo sketch...

15 Arduino/Genuino Uno su /dev/cu.usbmodem621

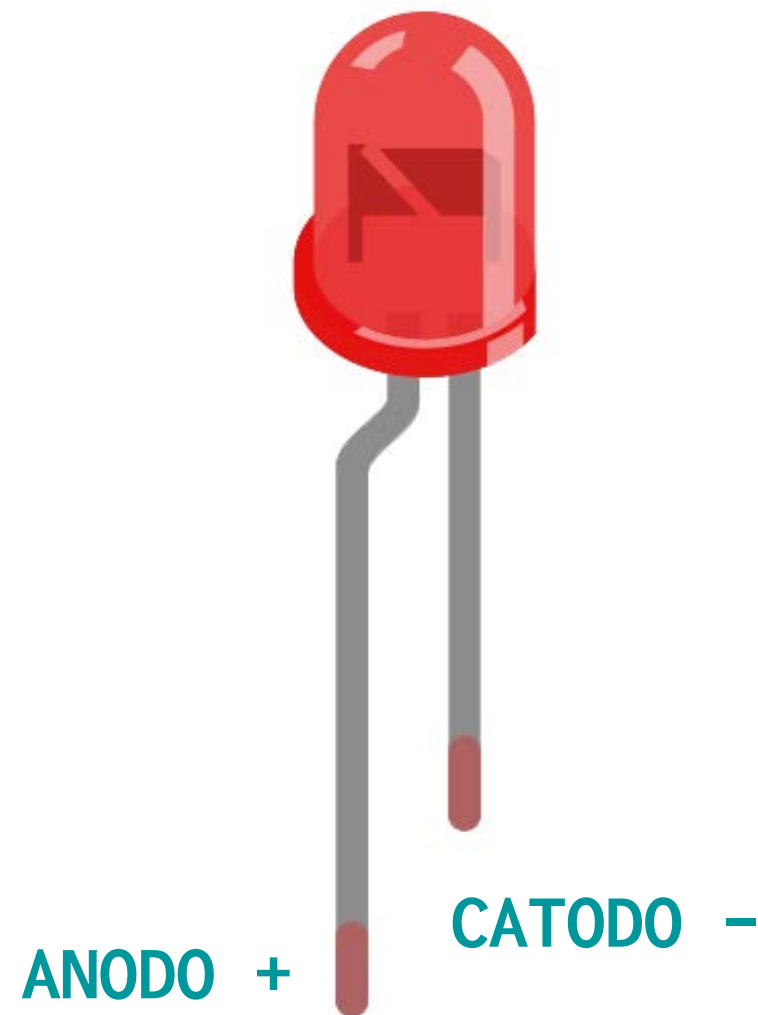


```
sketch_nov08a | Arduino 1.6.12  
sketch_nov08a §  
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

Sto compilando lo sketch...

15 Arduino/Genuino Uno su /dev/cu.usbmodem621

LED



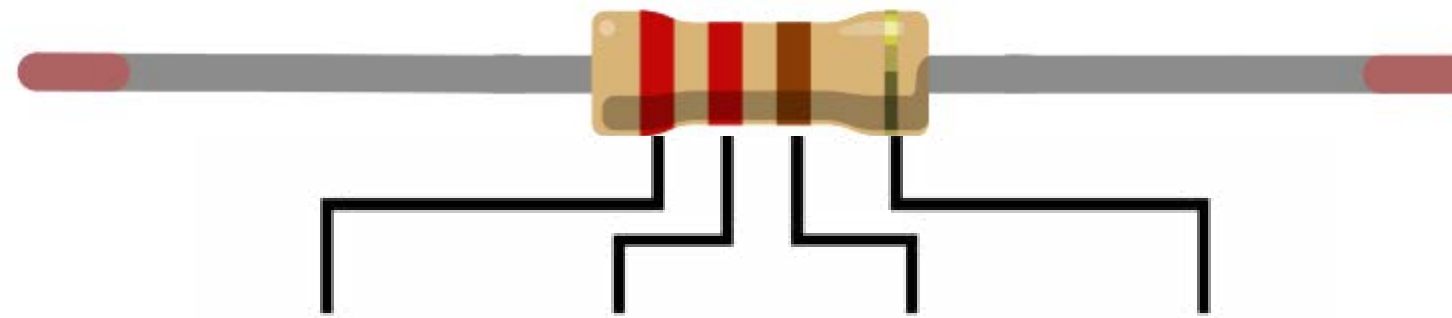
Il led si accende quando l'anodo si trova almeno a ~ 2.0 V in più rispetto al catodo

Il **polo -** è collegato a **GND**

Il **polo +** è collegato al **PIN** dopo il collegamento in serie ad una **resistenza**

NB: NON COLLEGARE MAI IL LED SENZA UNA RESISTENZA IN SERIE

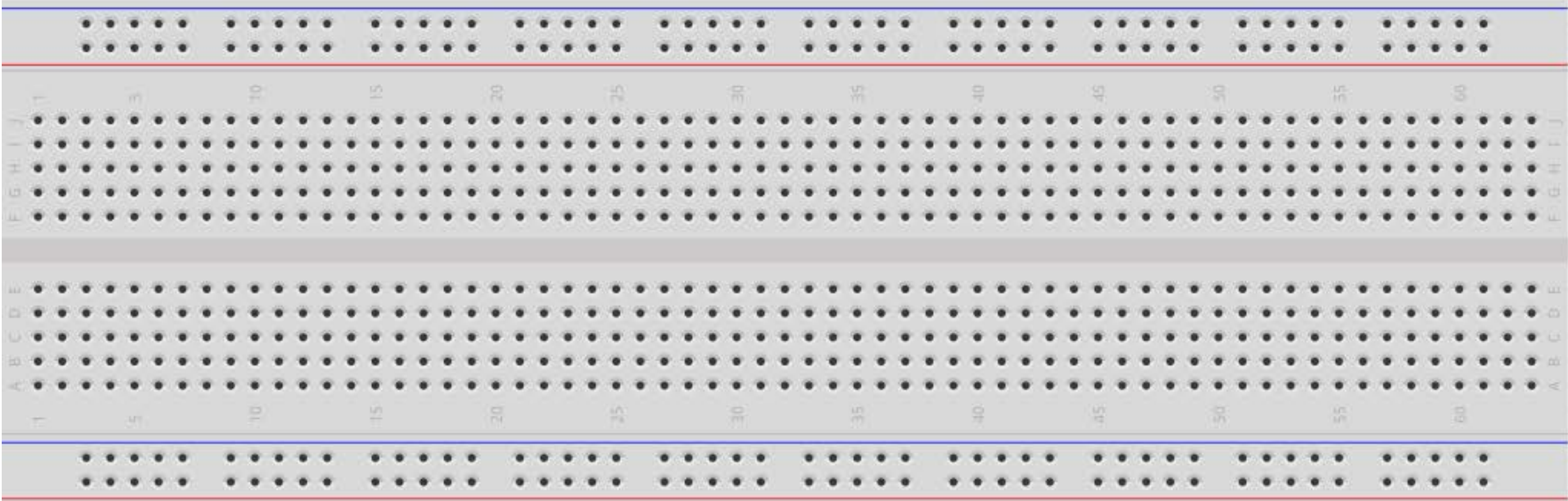
RESISTENZA



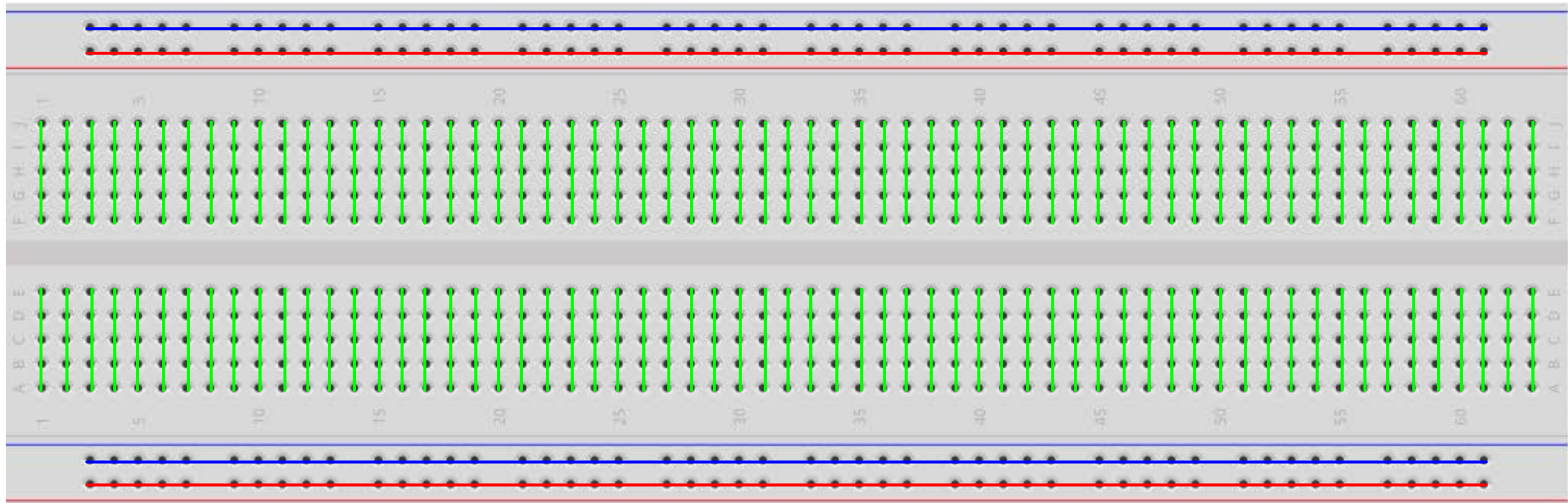
1st digit 2nd digit Multiplier Tolerance

0	0	x 1	
1	1	x 10	±1%
2	2	x 100	±2%
3	3	x 1K	
4	4	x 10K	
5	5	x 100K	
6	6	x 1M	
7	7		
8	8	x 0.1	±5%
9	9	x 0.01	±10%

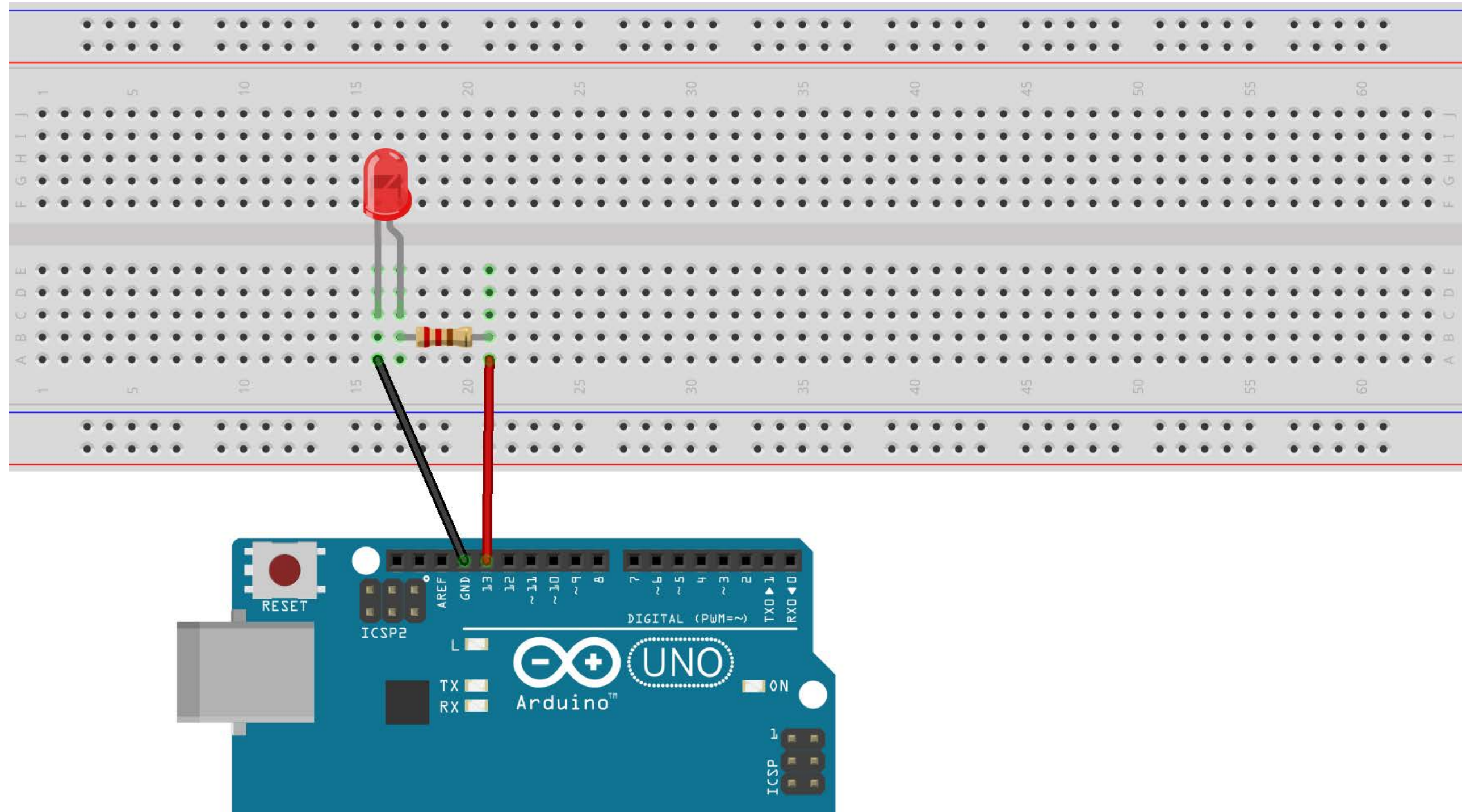
BREADBOARD



BREADBOARD



BLINKING LED: VCOLLEGAMENTI



ESERCIZIO 1B:

• • • — — — • • •

S

0

S

**ESERCIZIO 2:
PULSANTE**

PULSANTE



A e C sono sempre connessi,

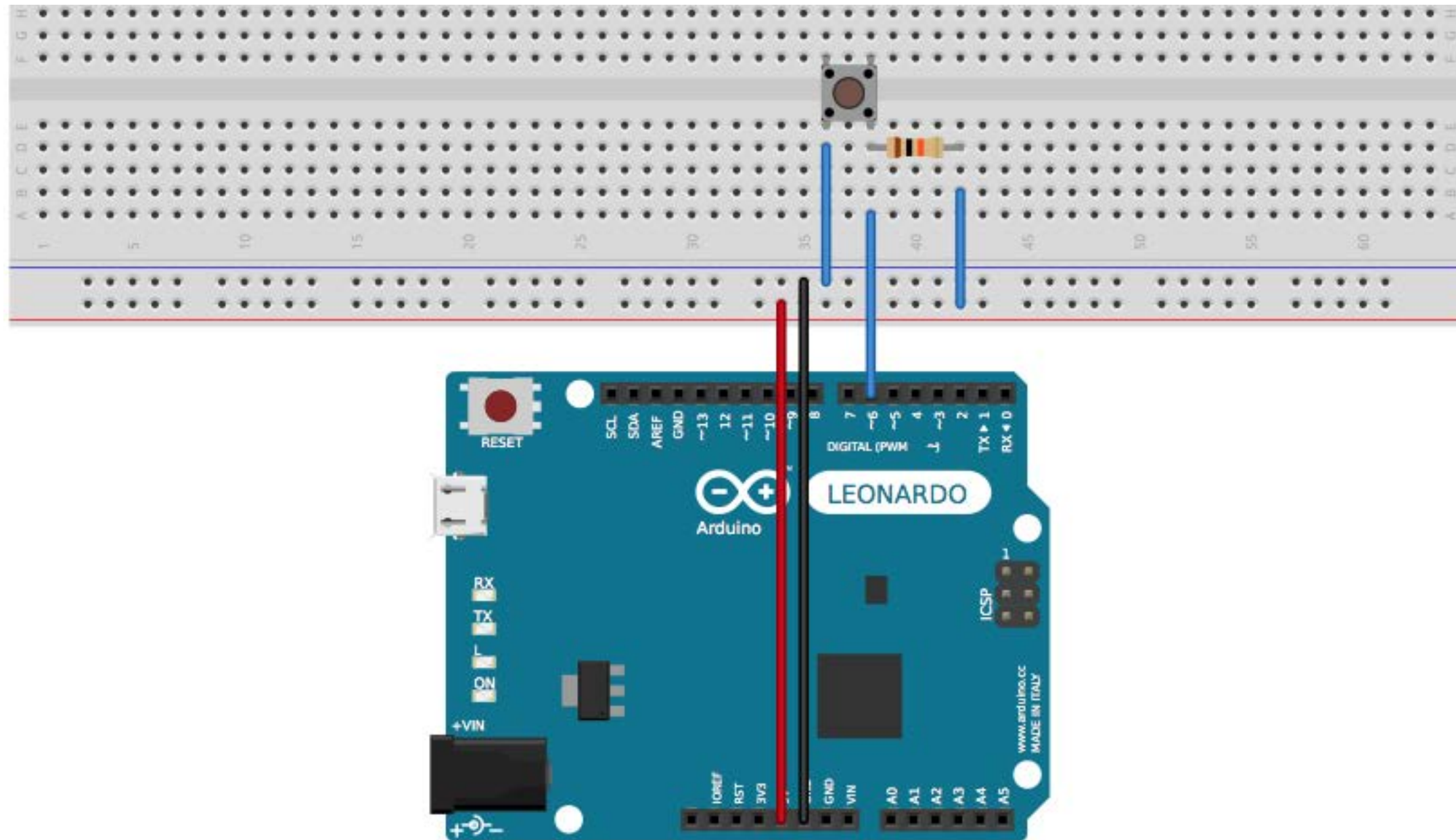
B e D sono sempre connessi.

Quando si schiaccia il bottone sono tutti connessi

PULSANTE: COMPONENTI



PULSANTE: COLLEGAMENTI



PULSANTE: SETUP

```
int ledPin = 13;
int buttonPin = 6;
int interval = 1000;

void setup(){

    pinMode(buttonPin, INPUT);
    pinMode(ledPin, OUTPUT);

}
```

PULSANTE: LOOP

```
void loop(){  
    if (digitalRead(buttonPin) == LOW){  
        digitalWrite(ledPin, HIGH);  
        delay(interval);  
        digitalWrite(ledPin, LOW);  
        delay(interval);  
    } else {  
  
        digitalWrite(ledPin, LOW);  
    }  
}
```

A GIOVEDÌ

CON LA SECONDA LEZIONE