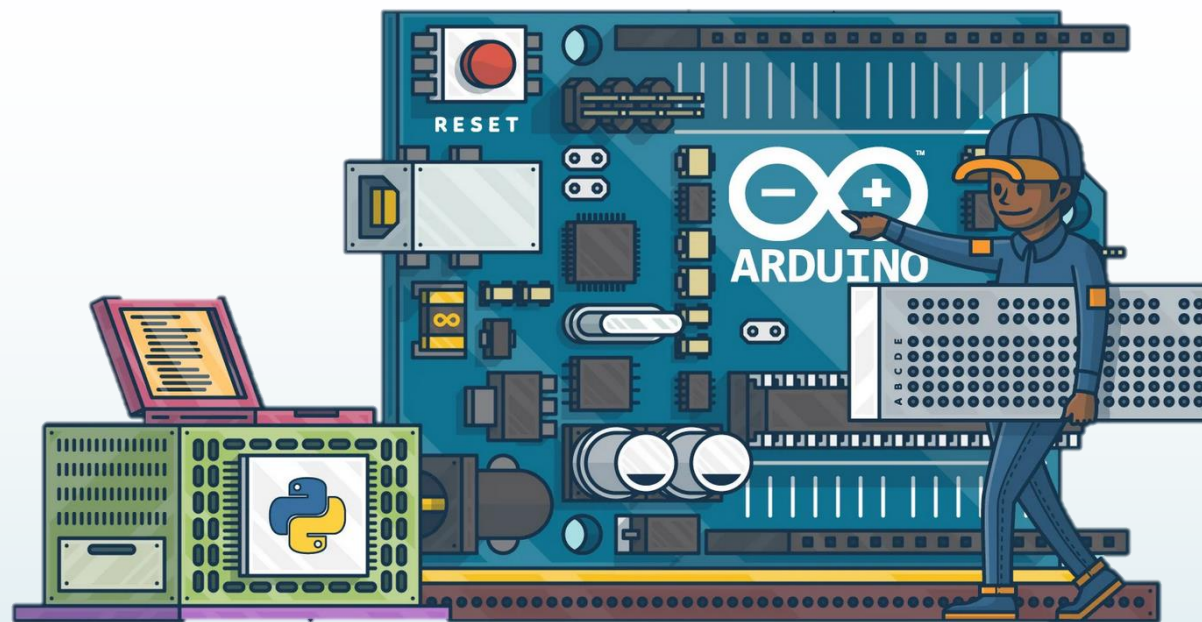




Lezione 1

Il mondo dell'automazione con ARDUINO

Liceo scientifico "Nicolò Palmeri" - A.S. 2024-2025

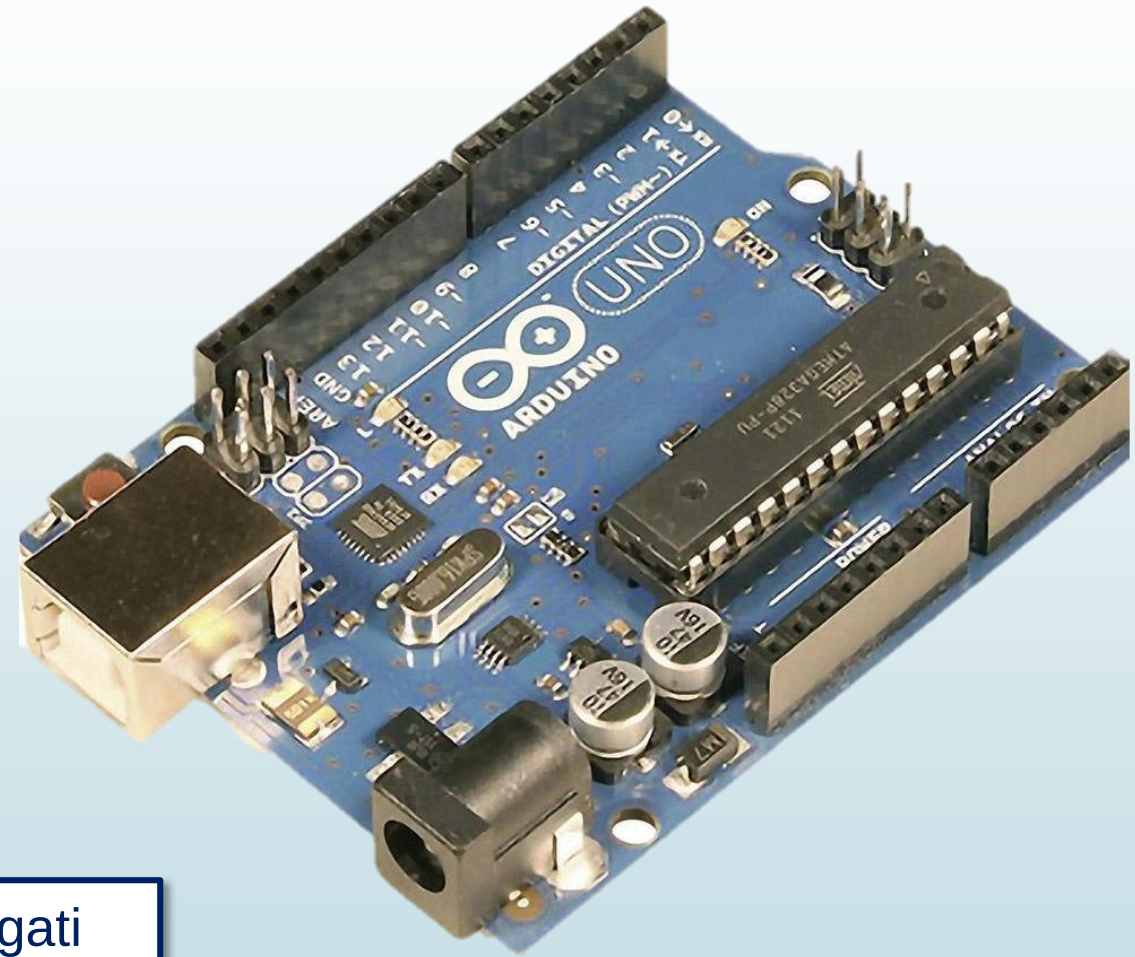
Prof. G. Vesco

Tutor: Prof.ssa E. La Tona

Cos'è ARDUINO

Arduino è una scheda elettronica dotata di **microcontrollore**. In aggiunta, Arduino include un opportuno software di sviluppo (**Arduino IDE** Integrated Development Enviroment), all'interno del quale è possibile scrivere i programmi (denominati **sketch**) che verranno eseguiti dal microcontrollore.

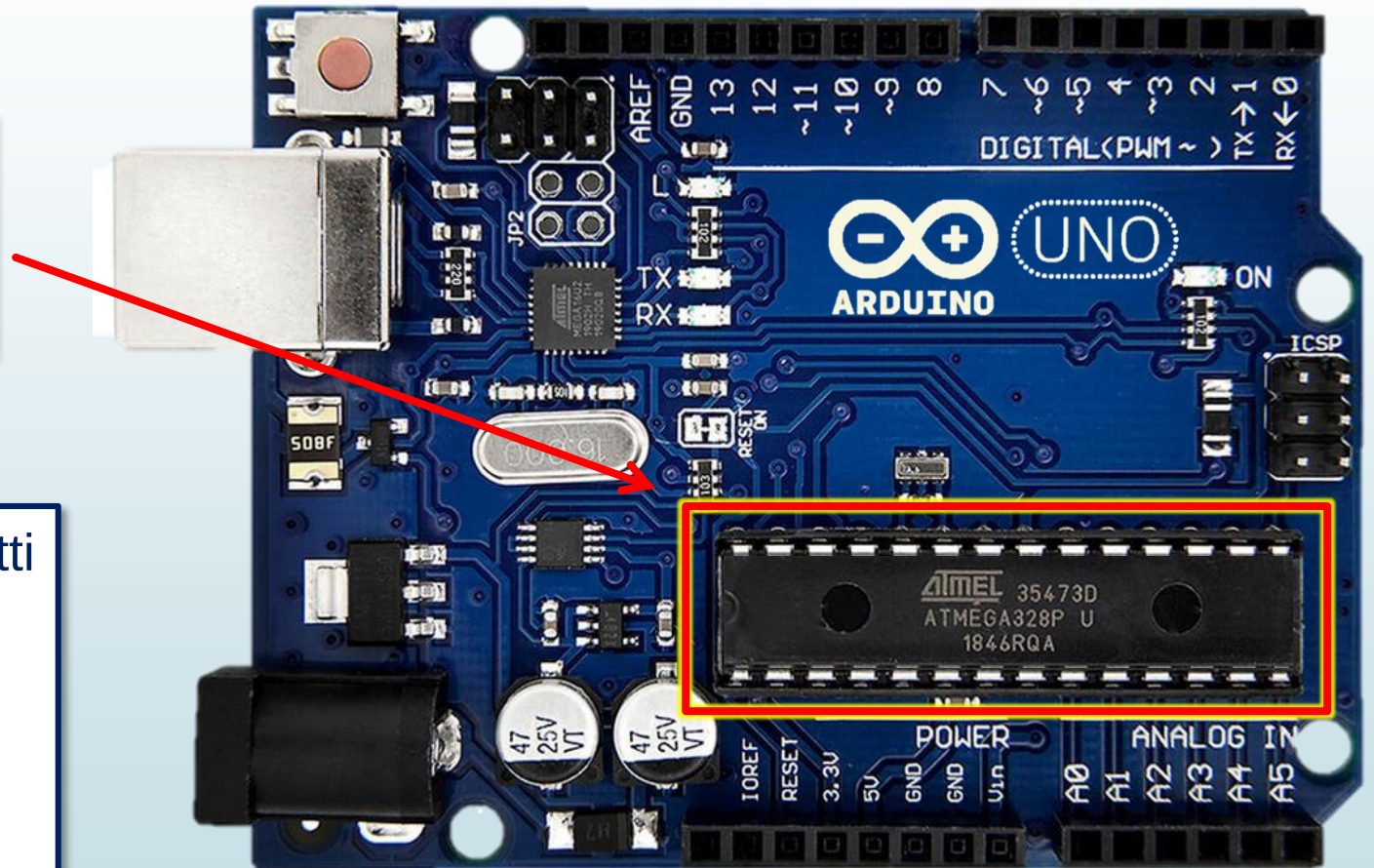
Attraverso **Sensori** e **Attuatori** collegati alla scheda Arduino è possibile realizzare un'infinità di progetti di **automazione**.



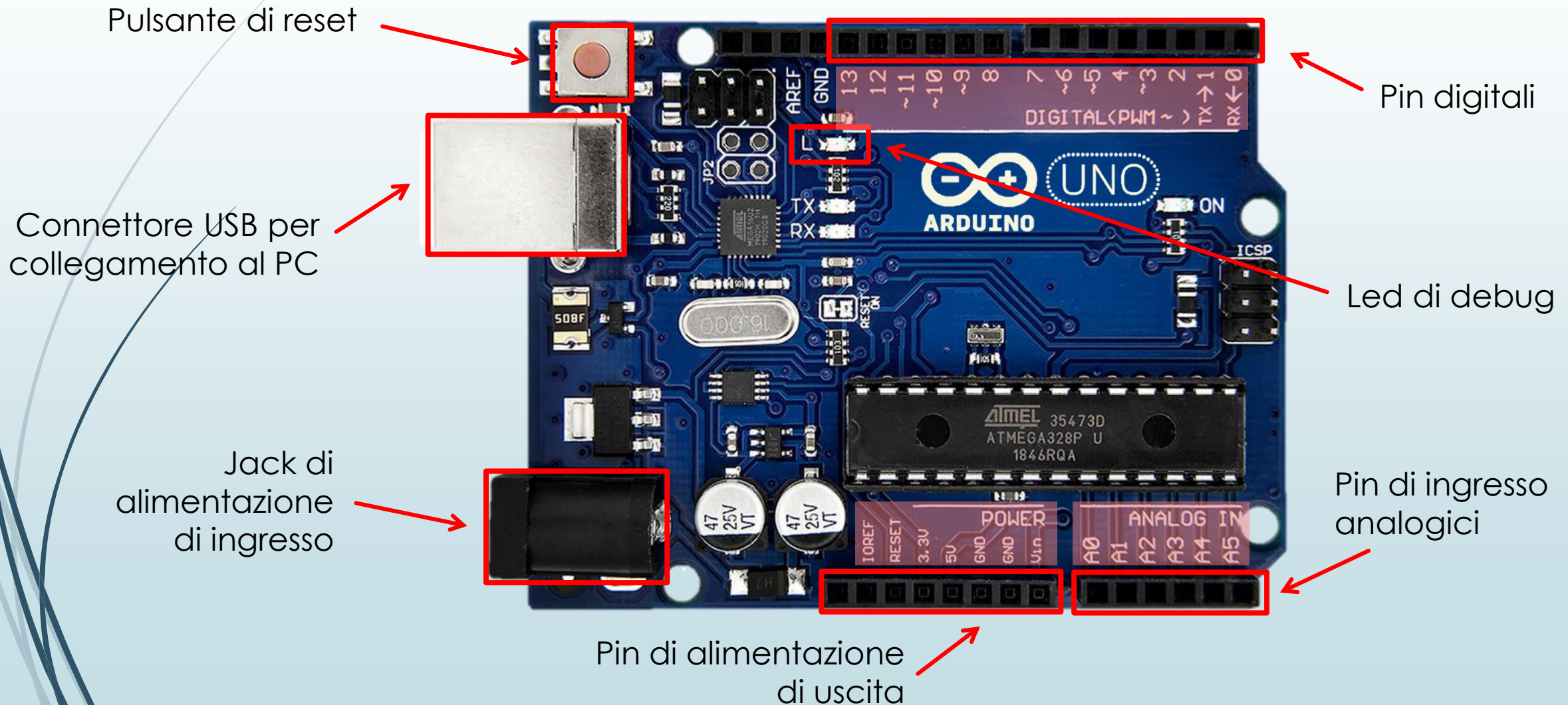
Cos'è ARDUINO

Il cuore della scheda Arduino è appunto il **microcontrollore ATMEL ATmega328P**

Un microcontrollore è a tutti gli effetti un **microcomputer** dotato di **CPU**, **RAM** e **memoria permanente**; ma anche *porte di ingresso e uscita*, *contatori*, *timer*, *convertitori A/D*.



Conosciamo la scheda ARDUINO UNO

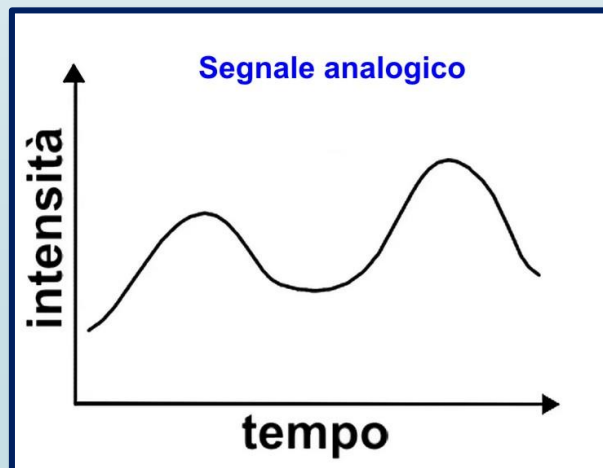


Grandezza analogiche e digitali

Le **grandezze analogiche** sono quelle grandezze che variano con continuità, ovvero i valori assunti non hanno “**salti**”.

La grandezza analogica assume un numero infinito di valori.

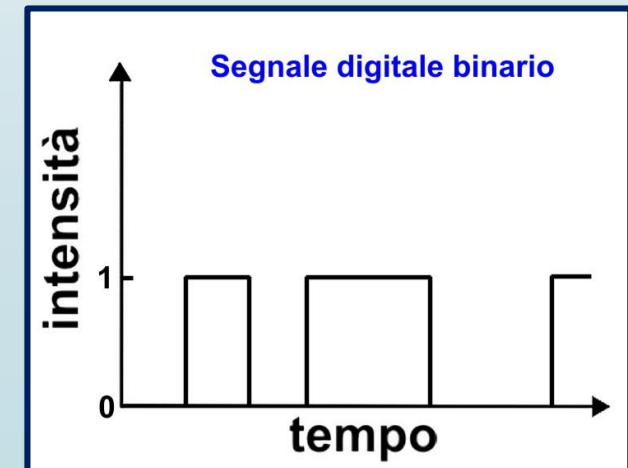
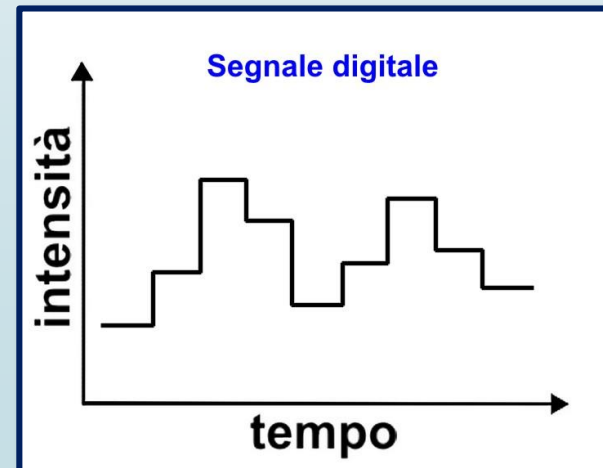
Esempio: **temperatura**



Le **grandezze digitali** sono quelle grandezze a valori discreti, ovvero che variano “**a salti**”.

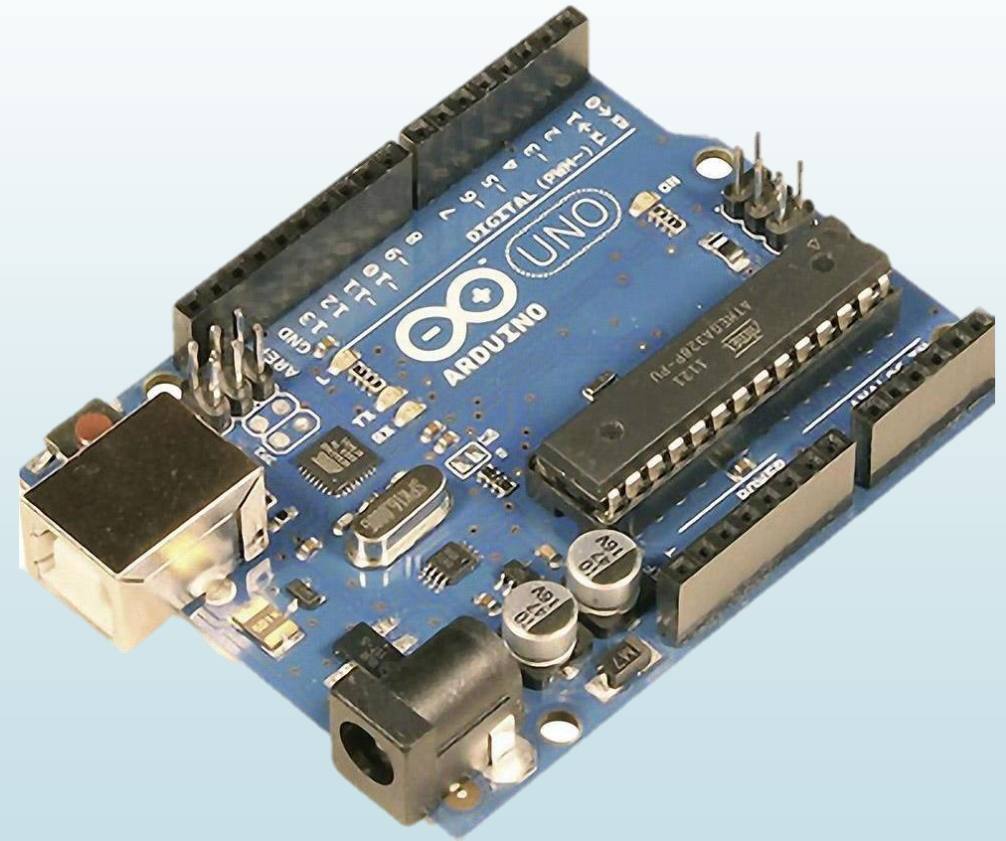
La grandezza digitale assume solo un numero finito di valori.

Se la grandezza digitale è anche **binaria** significa che assumerà solo 2 valori che solitamente sono **0** e **1**.



Caratteristiche di ARDUINO UNO

Caratteristica	Valore
Memoria Flash di programma	32 KB
SRAM	2 KB
Tensione di alimentazione	7-12V
Tensione di lavoro	5V
Pin digitali (I/O)	14
Pin Analogici (I)	6
Corrente continua per Pin	40 mA



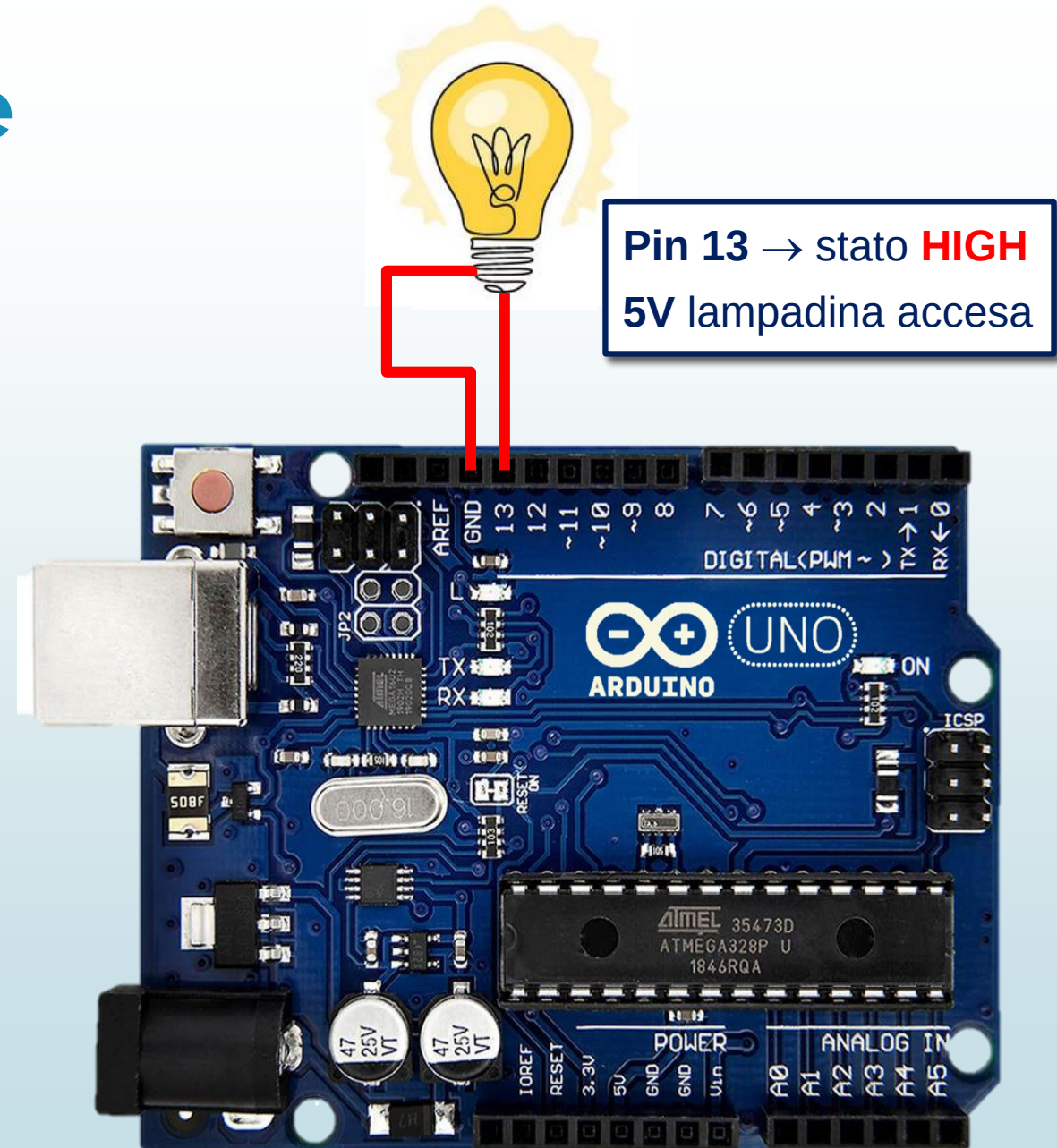
Livelli di tensione

I microcontrollori sono **dispositivi digitali**, dove l'informazione è codificata in 2 stati discreti:

- **HIGH** (stato logico 1)
- **LOW** (stato logico 0)

Tali stati corrispondono a 2 livelli di tensione:

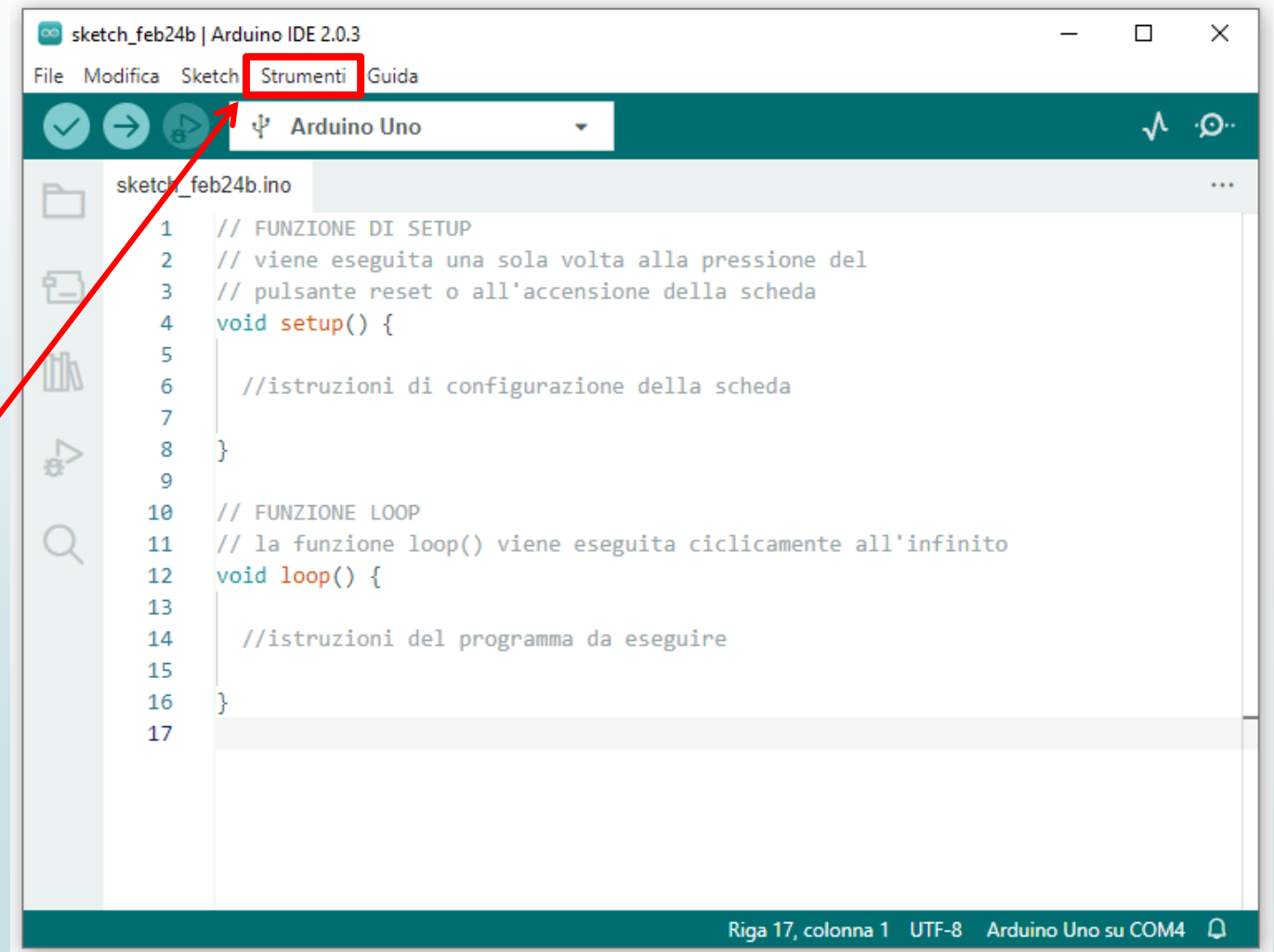
- **5V** → stato **HIGH**
- **0V** → stato **LOW**



Software IDE

Per poter scrivere un programma per Arduino si utilizza il software **Arduino IDE**. Il linguaggio utilizzato è molto simile al C/C++.

Dal menù **Strumenti** è possibile selezionare la scheda in uso (**Arduino UNO**) e la porta **COM** di comunicazione tra computer e scheda.



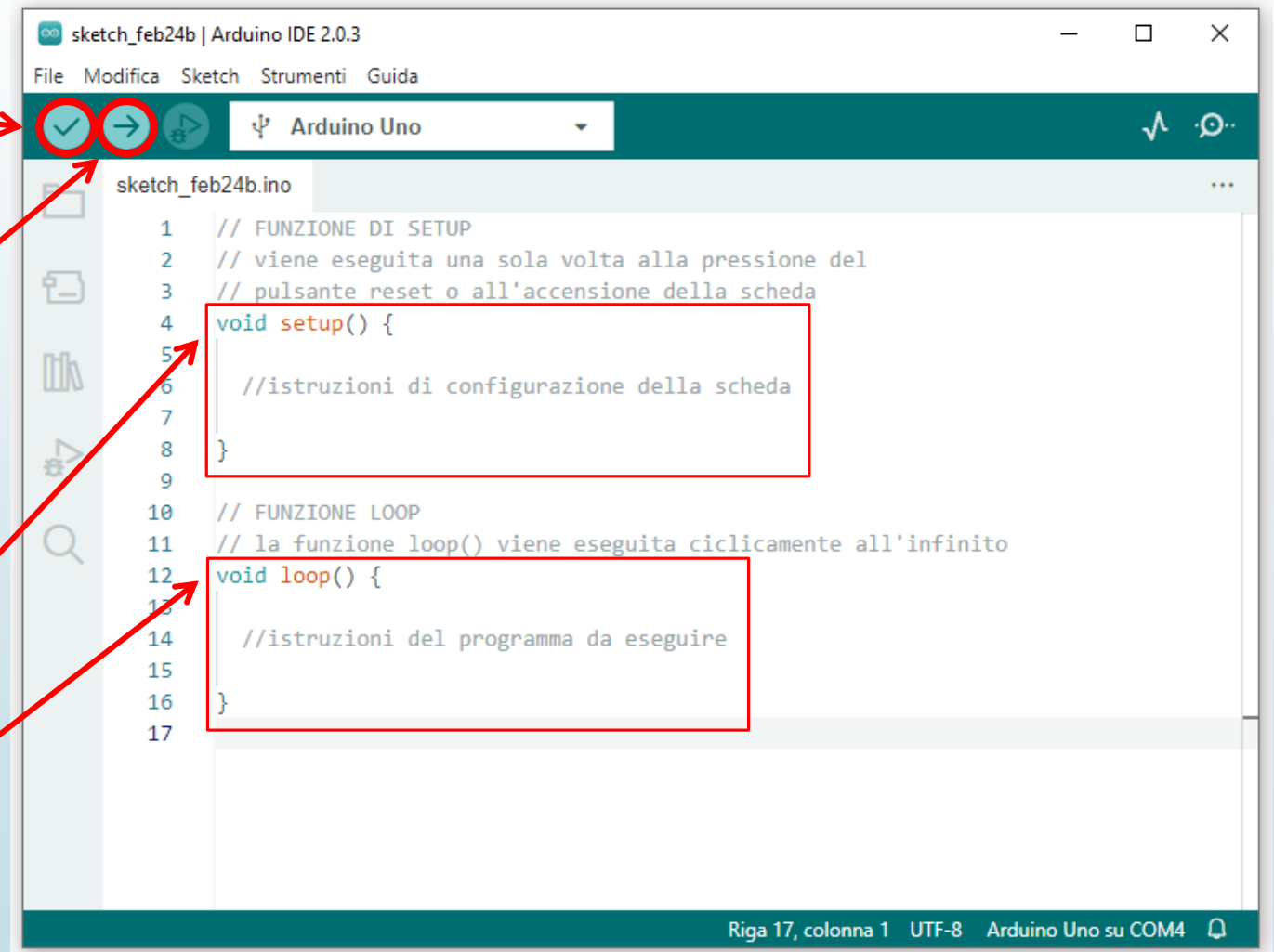
Software IDE

Il pulsante **Verifica**: controlla la presenza di errori nel codice scritto

Il pulsante **Carica**: compila il programma e lo carica sulla scheda Arduino

Il programma di Arduino è formato da **2 parti**:

- funzione di **setup()**, che contiene i settaggi della scheda
- funzione di **loop()**, che contiene le istruzioni da eseguire



Far lampeggiare un LED - software



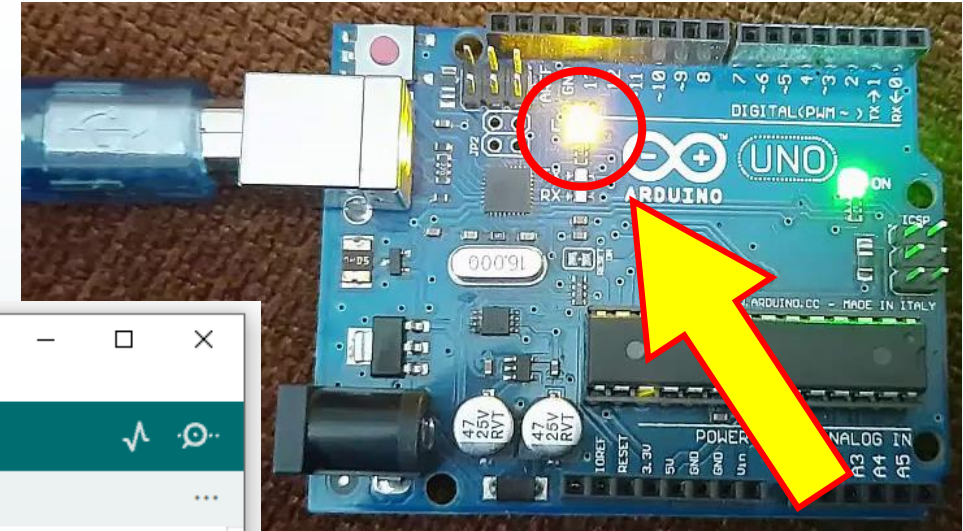
```
Blink.ino
1 // FUNZIONE DI SETUP
2 // viene eseguita una sola volta alla pressione del pulsante reset o all'accensione della scheda
3 void setup() {
4
5   // configurazione del pin digitale 13 come uscita
6   pinMode(13, OUTPUT);
7
8 }
9
10 // FUNZIONE LOOP
11 // la funzione loop() viene eseguita ciclicamente all'infinito
12 void loop() {
13
14   digitalWrite(13, HIGH); // ACCENDE IL LED (pin 13 a livello alto di tensione)
15   delay(1000);             // attendi 1 secondo (1000 millisecondi)
16   digitalWrite(13, LOW);  // SPEGNE IL LED (pin 13 a livello basso di tensione)
17   delay(1000);            // attendi 1 secondo (1000 millisecondi)
18
19 }
20
```

Istruzione **pinMode()**:
serve per configurare i
piedini di ARDUINO
come ingressi o uscite

Istruzione **digitalWrite()**:
serve per settare un pin
digitale a **HIGH** o **LOW**

Istruzione **delay()**:
serve per introdurre una
pausa dei *ms* indicati

Far lampeggiare un LED



```
Blink | Arduino IDE 2.0.3
File Modifica Sketch Strumenti Guida
Arduino Uno
Blink.ino
1 // FUNZIONE DI SETUP
2 // viene eseguita una sola volta alla pressione del pulsante reset o all'accensione della scheda
3 void setup() {
4
5     // configurazione del pin digitale 13 come uscita
6     pinMode(13, OUTPUT);
7
8 }
9
10 // FUNZIONE LOOP
11 // la funzione loop() viene eseguita ciclicamente all'infinito
12 void loop() {
13
14     digitalWrite(13, HIGH); // ACCENDE IL LED (pin 13 a livello alto di tensione)
15     delay(1000);           // attendi 1 secondo (1000 millisecondi)
16     digitalWrite(13, LOW);  // SPEGNE IL LED (pin 13 a livello basso di tensione)
17     delay(1000);           // attendi 1 secondo (1000 millisecondi)
18
19 }
20
```

Riga 20, colonna 1 UTF-8 Arduino Uno su COM4 2



Il LED presente sulla scheda Arduino e collegato al PIN 13, lampeggerà in maniera ciclica restando acceso per 1 secondo e spento per 1 altro secondo.

Far lampeggiare un LED esterno

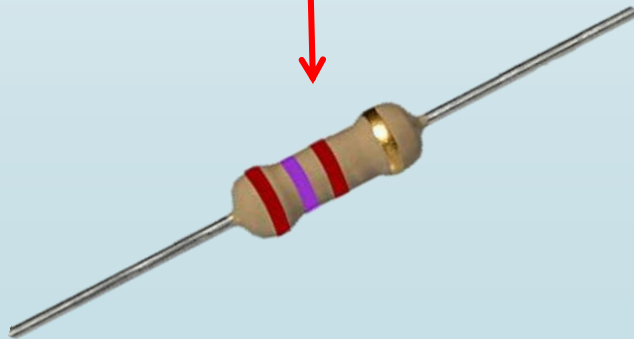
Procediamo adesso a **far lampeggiare un LED esterno** attraverso la scheda Arduino.

Abbiamo bisogno di collegare il LED alla scheda;
per far ciò ci serviranno 3 oggetti:

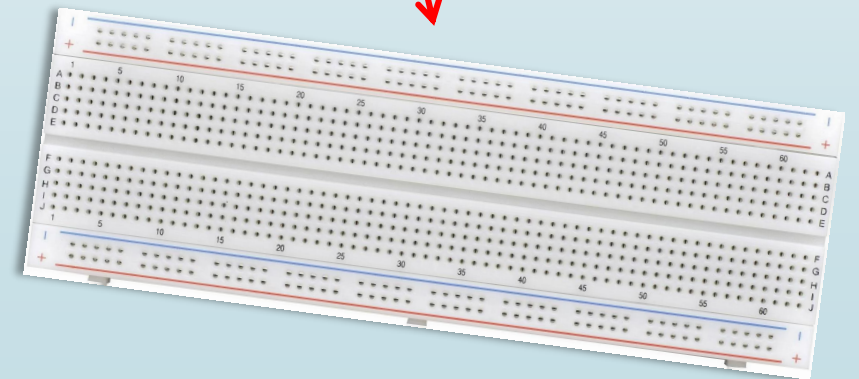
un **LED**



una **Resistenza**

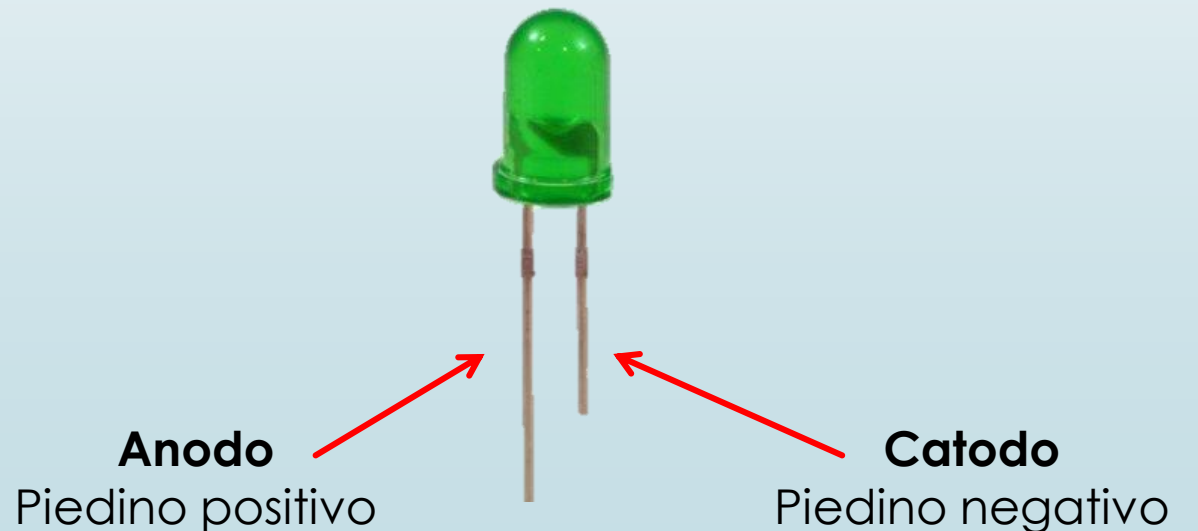
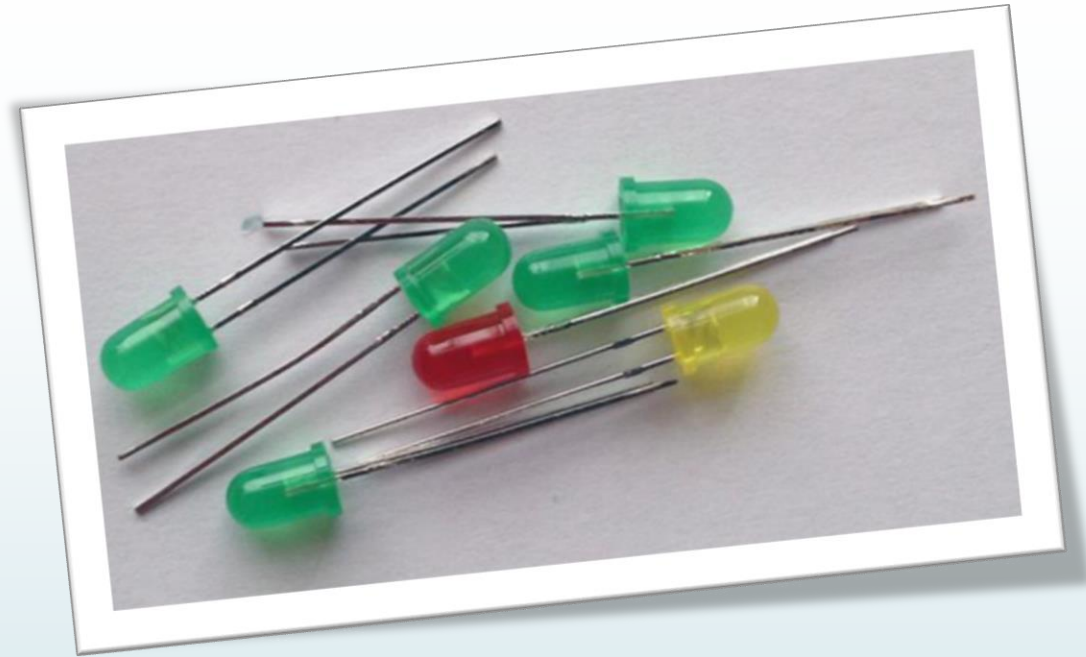


una **Breadboard**



LED

L'acronimo **LED** deriva dalle parole inglesi **L**ight **E**mitting **D**iode, sono diodi che **emettono luce** quando vengono attraversati dalla corrente elettrica. Poiché sono dei diodi essi emetteranno luce solo quando sono attraversati nel giusto senso dalla corrente elettrica, nel senso opposto della corrente il diodo non si illumina.



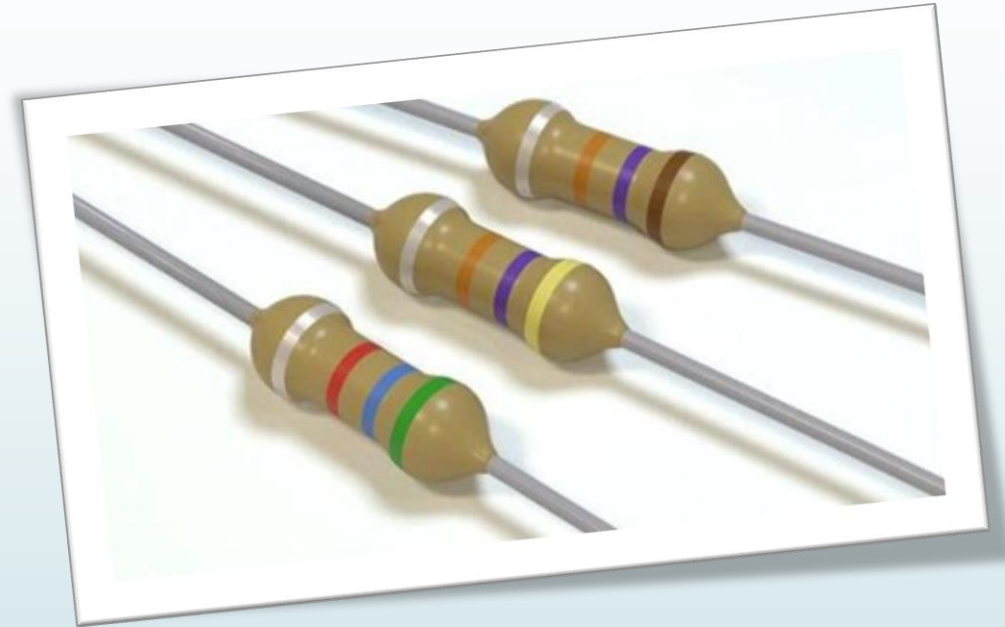
Resistenza

Una **resistenza** o resistore è un componente che si oppone al passaggio della corrente elettrica e converte parte dell'energia elettrica in calore (**effetto Joule**).

Se si mette una resistenza in serie con un componente come un led, la resistenza consuma parte dell'energia elettrica e il led riceve meno energia. Ciò consente di alimentare componenti con la quantità di energia di cui hanno bisogno.

Si usa una resistenza in serie con il led per evitare che riceva troppa corrente.

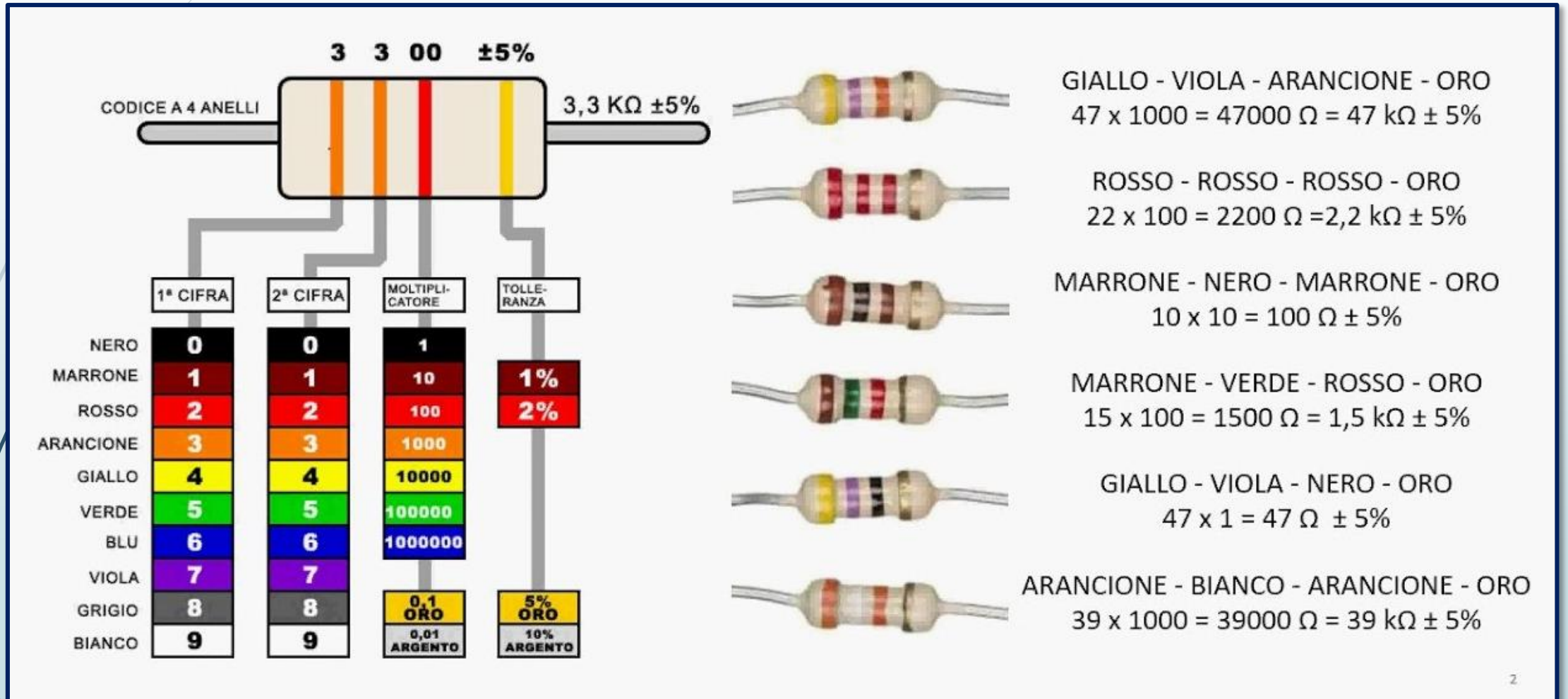
**Senza la resistenza, il led
si brucerebbe rapidamente.**



L'unità di misura
della resistenza è
l'Ohm $\rightarrow \Omega$



Codici colori delle resistenze

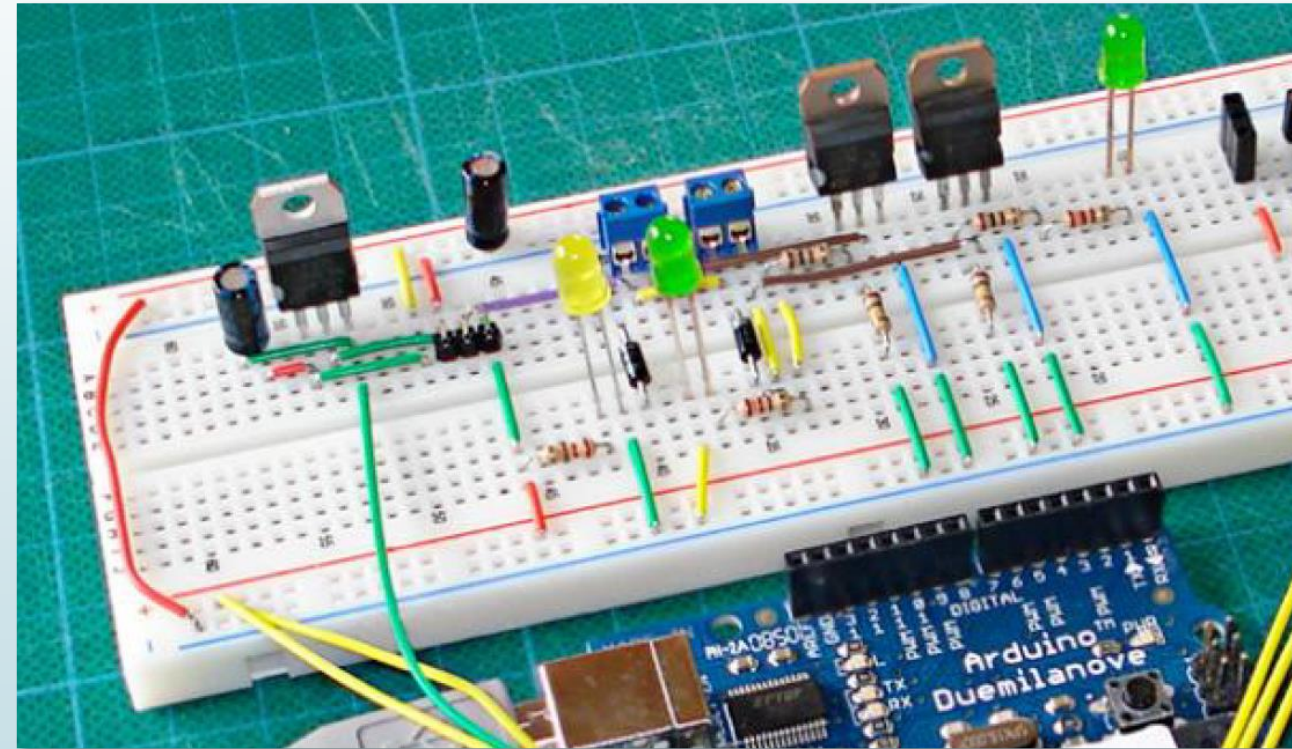


Breadboard

La **breadboard** è una basetta che serve a creare **prototipi di circuiti elettronici**. Ha dei fori su cui possono essere inseriti dei cavi di collegamento.

Non necessita di saldature e può essere riutilizzata.

Grazie a questo prezioso strumento, è possibile eseguire velocemente i collegamenti tra la scheda Arduino e i vari componenti.



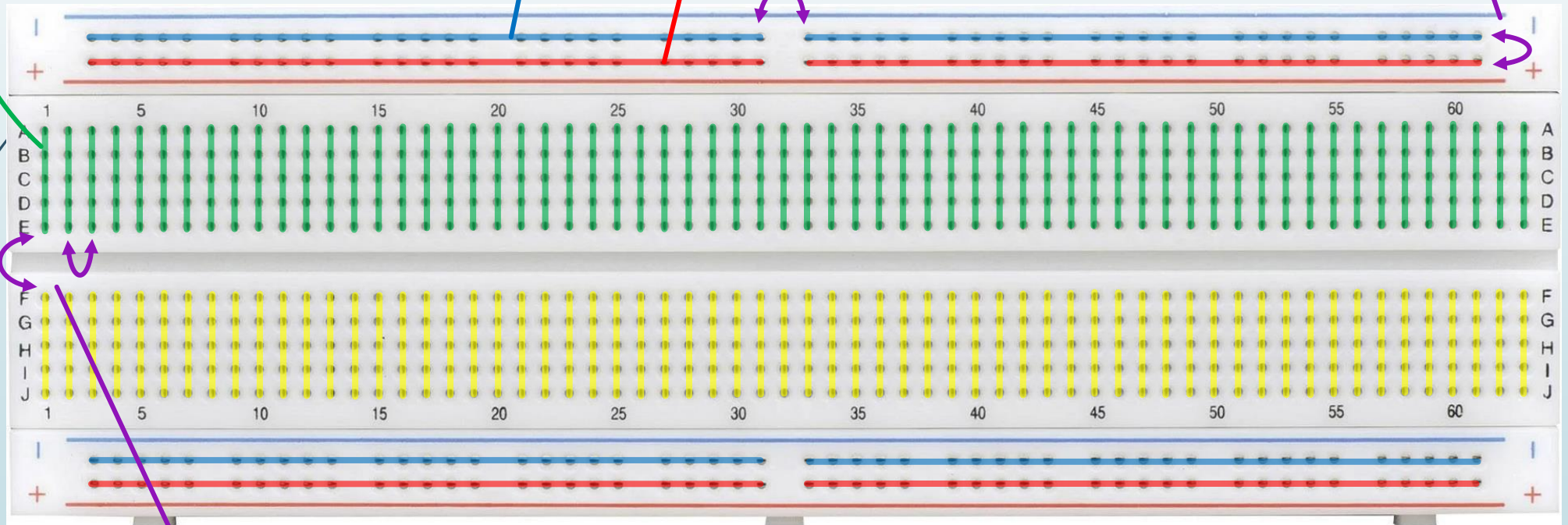
Breadboard

Tutti i fori della colonna
sono connessi insieme

Tutti i fori della riga
sono connessi insieme

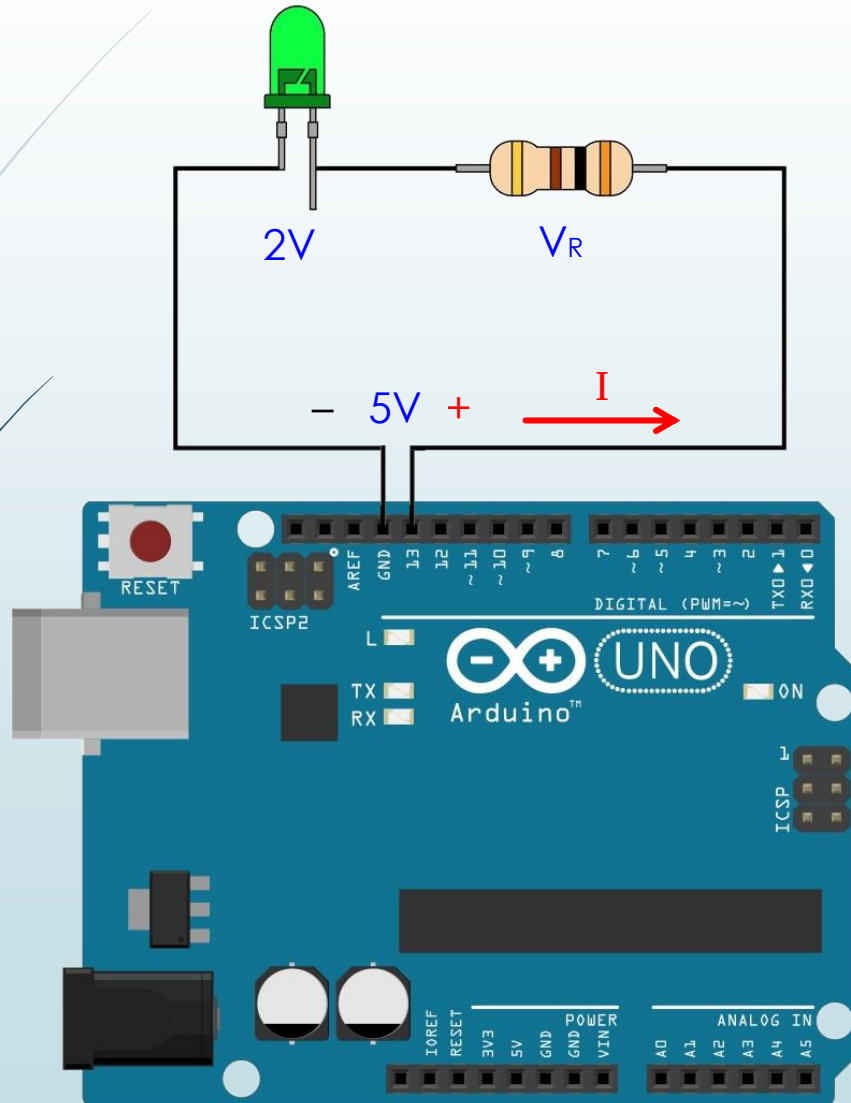
Tutti i fori della riga
sono connessi insieme

Tra le 2 righe non vi è
nessun collegamento



Non vi è nessun collegamento tra le colonne

Circuito elettrico



Calcolare la **resistenza** applicando la **legge di Ohm**:

$$V_R = R \cdot I_R$$

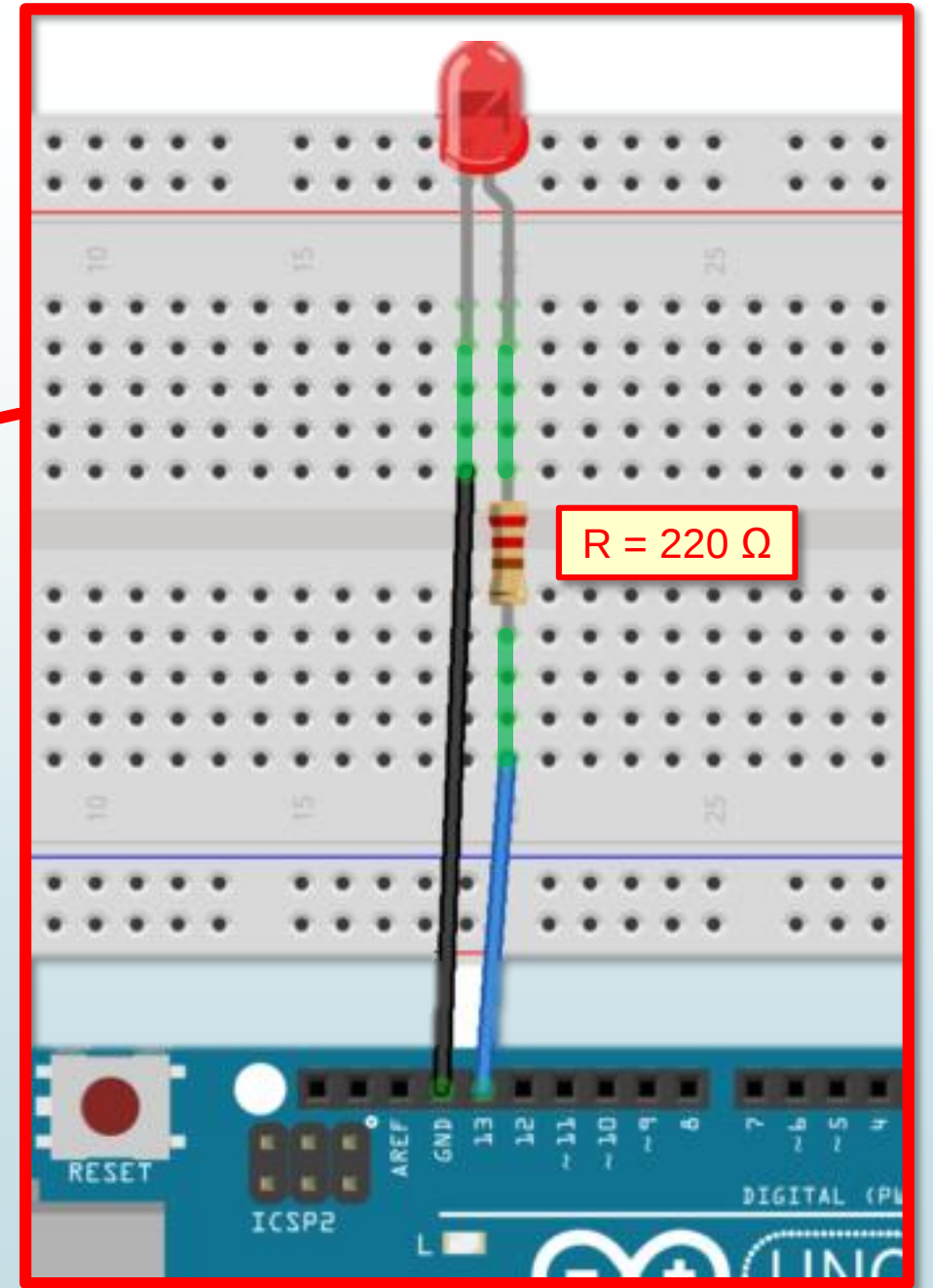
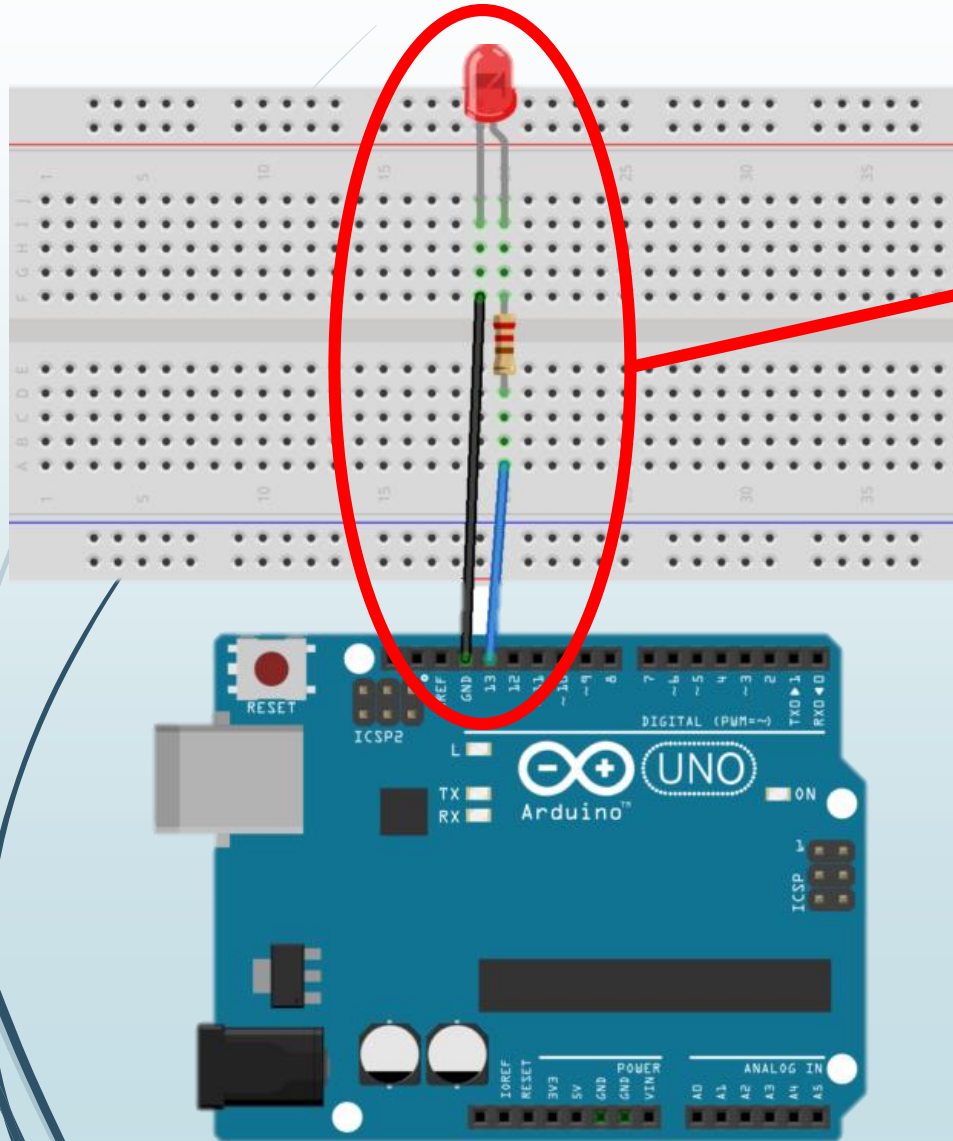
Sapendo che:

- la tensione fornita da Arduino è **5V**
- la tensione sul LED è di **2V**
- la corrente che circola è di **15mA**

Per formula inversa si ricava che:

$$R = \frac{V_R}{I_R} \cong 220 \, \Omega$$

Collegamento del LED

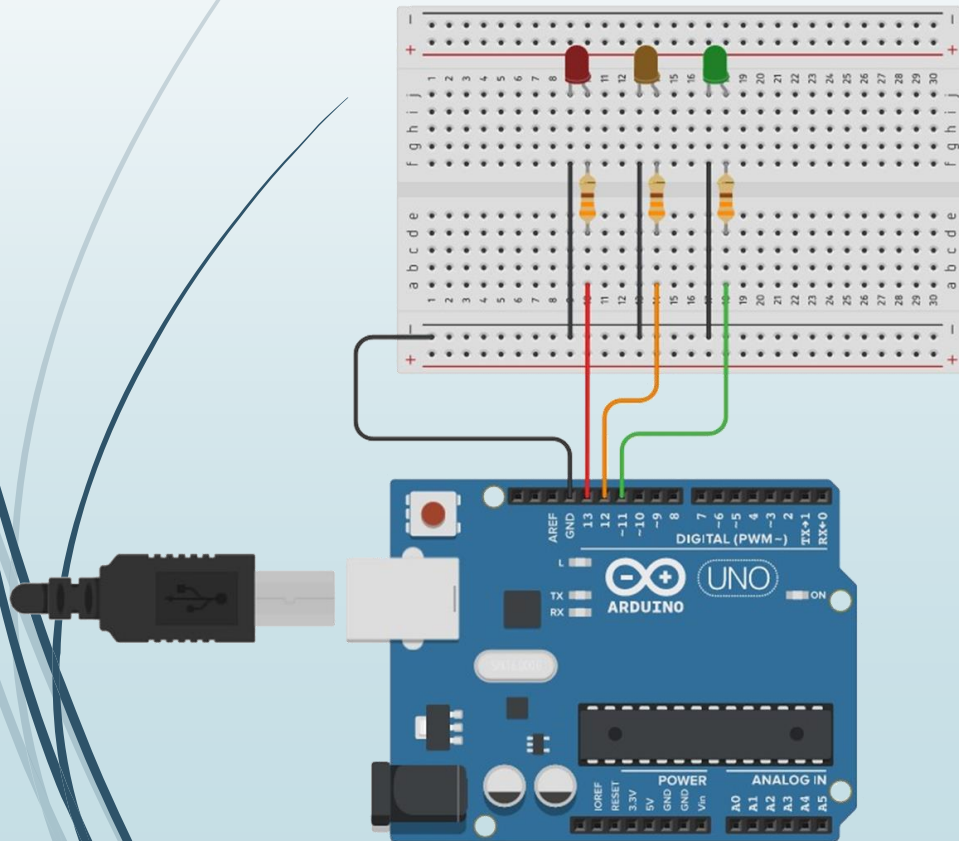


Esercizio 1: ciclo femaforico

Attraverso 3 led simulare un semaforo.

Il ciclo semaforico dovrà far restare acceso:

- il **verde** per **5 secondi**
- l'**arancione** per **2 secondi**
- il **rosso** per altri **7 secondi**



Soluzione

esempio.ino

```
1  #define LEDr 13
2  #define LEDg 12
3  #define LEDv 11
4
5  void setup() {
6      pinMode(LEDr, OUTPUT);
7      pinMode(LEDg, OUTPUT);
8      pinMode(LEDv, OUTPUT);
9  }
10
11 void loop() {
12     digitalWrite(LEDr, LOW);
13     digitalWrite(LEDv, HIGH);
14     delay(5000);
15     digitalWrite(LEDv, LOW);
16     digitalWrite(LEDg, HIGH);
17     delay(2000);
18     digitalWrite(LEDg, LOW);
19     digitalWrite(LEDr, HIGH);
20     delay(7000);
21 }
```

Esercizio 2: lampeggio sfalsato

Far lampeggiare i 2 led con frequenze diverse, ovvero:

LEDv acceso per 1s e spento per 1s

LEDr acceso per 550ms e spento per 550ms

