



**APS
Fablab
Verona**

workshop introduttivo
Arduino



Verona, 8 Maggio 2025



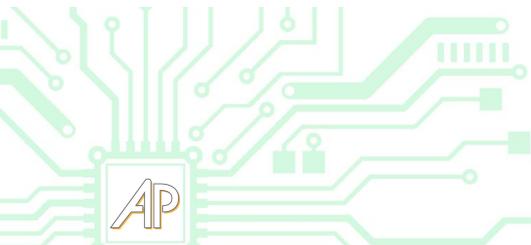
Aldo Pizzeghella
aldopizzeghella@gmail.com



bando del **Fondo Sociale Europeo**
promosso dalla **Regione Veneto**

“attivare la cittadinanza, di qualsiasi tipo
in attività laboratoriali, workshop ed altre
attività pratiche
in cui i cittadini possono
apprendere in modo informale competenze
e sviluppare nuovi interessi”

<https://forms.gle/h8WLn1JC5Wk4tTEZ7>





**APS
Fablab
Verona**



Gruppo Nerd
giovedì sera
20:30 - 23:30



Repair Café
giovedì sera (pari)
18:30 - 20:30



MakerSpace
giovedì sera
20:30 - 23:30



**CoderDojo
Verona**
sabato
15:00 - 17:00



**IS
Fablab
Verona**

**311
VERONA**

AP

workshop: **Arduino Display** - 8 maggio 2025 - Aldo Pizzeghella

Formazione docenti



Formazione Studenti



Progetti



chi sono



Esperto del processo Artigrafiche (stampa), tecnico informatico, consulente su sistemi di color management, calibrazione e misurazione del colore

Collaboro con FabLabVr - CoderDojoVr - corsi STEM

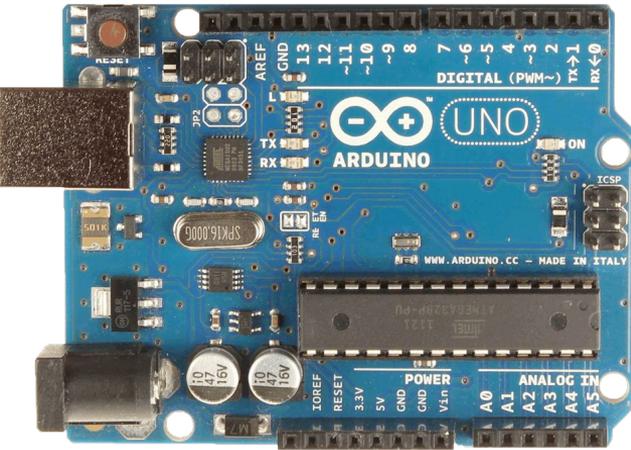
Appassionato di elettronica e programmazione

Seguo Arduino fin dalla nascita partecipo a corsi, sviluppo prototipi di automazione e IOT...



AP

che cosa è?



E' una scheda **elettronica programmabile**

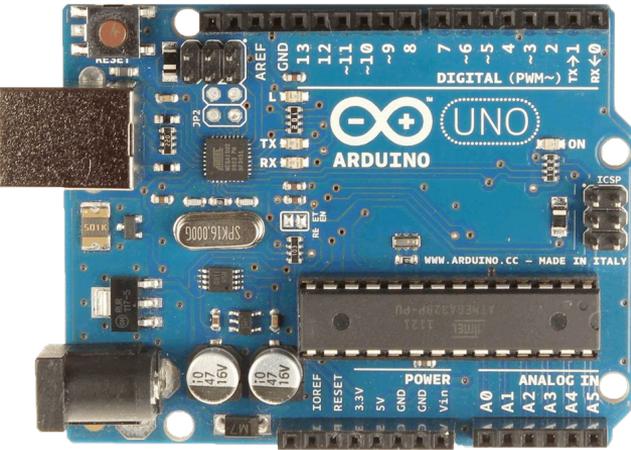
E' un microcontrollore

- facile da programmare
- economica
- flessibile
- espandibile
- open source

E' stata progettata per semplificare l'introduzione della programmazione ad allievi di una scuola di design ad **IVREA - ITALIA**

Grazie alla sua semplicità il progetto si è immediatamente diffuso in tutto il mondo

a chi è rivolto?



Hobbisti – Studenti

semplici ed economici prodotti con i quali qualsiasi persona anche non esperta (a qualsiasi età), può avvicinarsi alla programmazione e alla realizzazione di prototipi.

Scuole

di ogni ordine e grado di sono trovati a disposizione strumenti per introdurre l'insegnamento dell'elettronica ed informatica in modo semplice ma altamente stimolante.

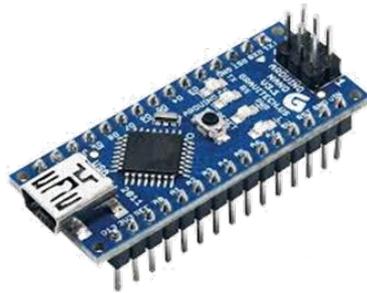
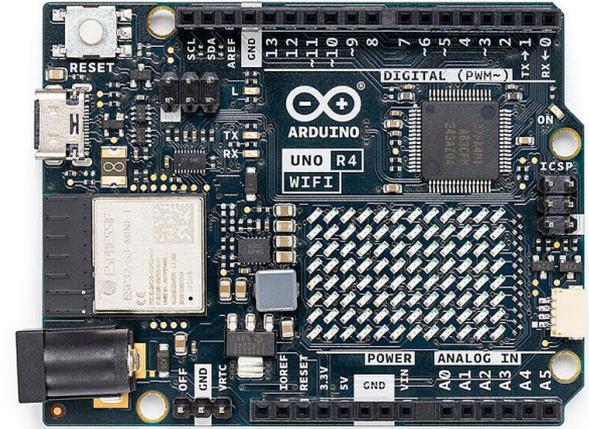
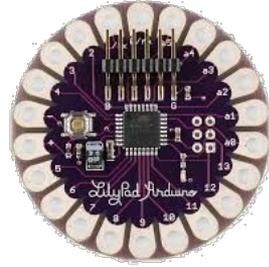
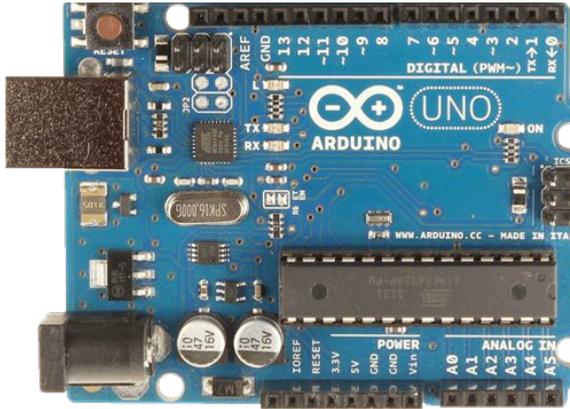
Università

hanno acceso il loro interesse e si sono attivate sviluppando metodologie didattiche innovative.

Aziende

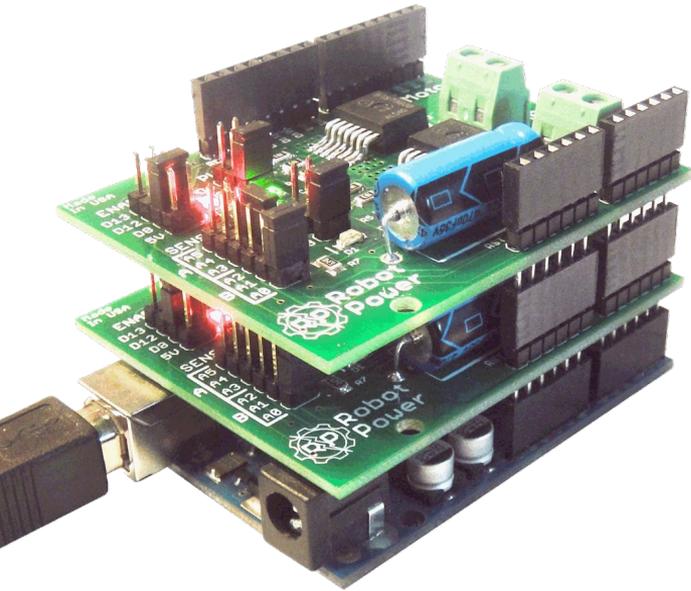
hanno capito le potenzialità e la dimensione di un mercato in espansione e si sono messi ad investire e proporre nuovi componenti e soluzioni

la famiglia di ARDUINO

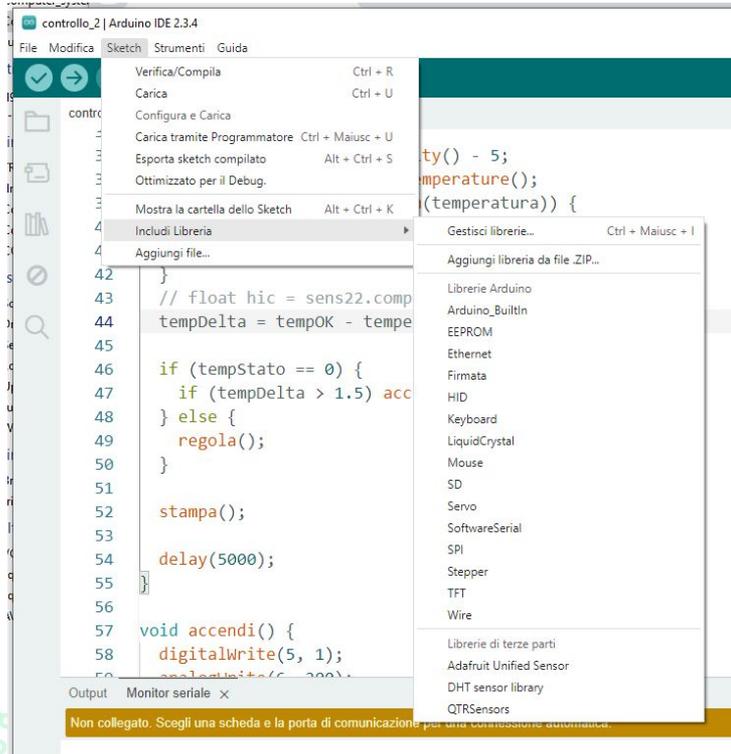


Shields

Arduino UNO lo possiamo espandere senza avere grandi conoscenze di elettronica. Sono state prodotte moltissime schede (SHIELDS) che impilandosi una sopra l'altra aggiungono dispositivi e funzionalità pressoché infinite.



Librerie



Le **Librerie** sono un'altro punto di forza.

Per poter governare i dispositivi elettronici che colleghiamo è necessario scrivere un programma opportuno.

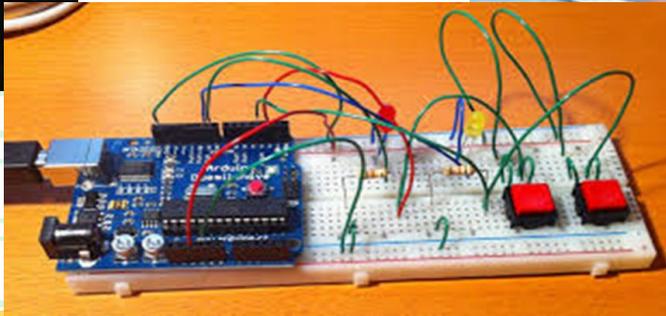
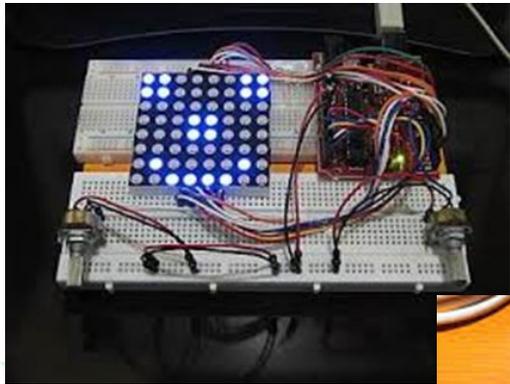
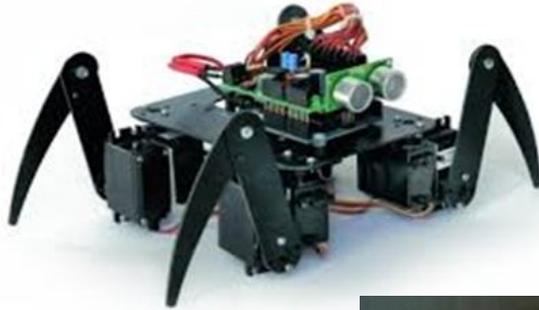
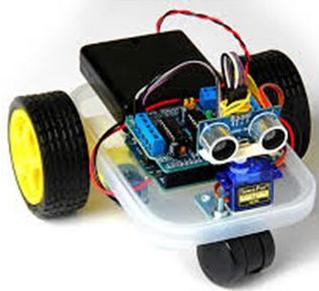
Questo impone la conoscenza profonda del dispositivo impiegato.

In rete troviamo chi ci aiuta.

Troviamo una enorme quantità di software già scritto per qualsiasi dispositivo vogliamo collegare.

Questi software sono organizzati in **Librerie** da includere nel nostro programma.

cosa possiamo fare?



AP

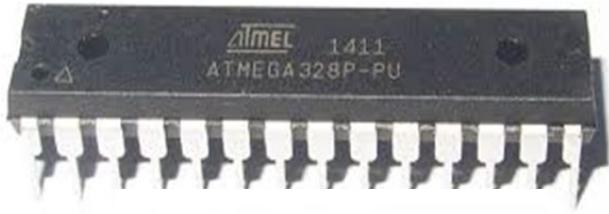
il cuore

ATmega 328P (PU o AU)

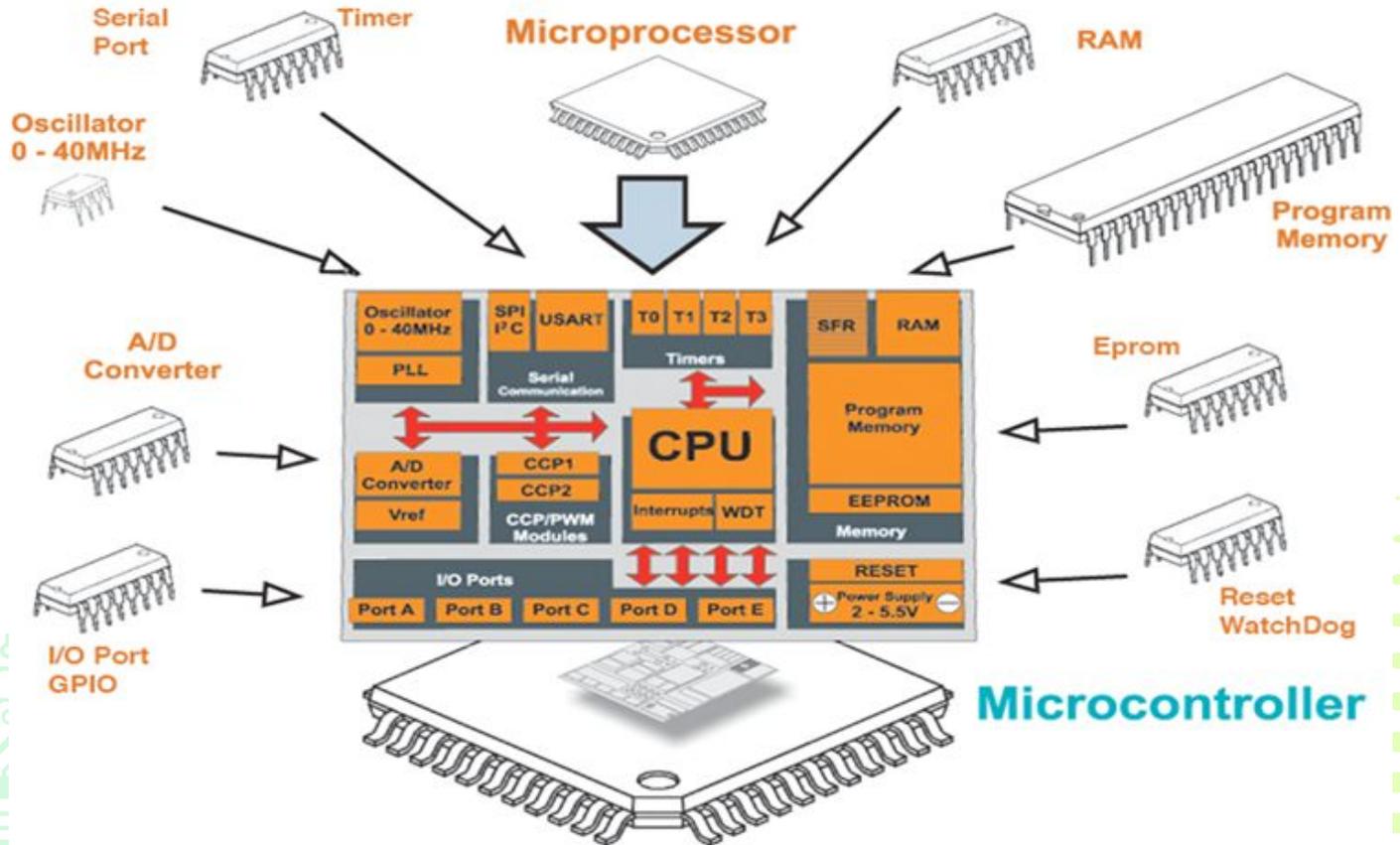
microcontrollore prodotto da **ATMEL** (Microchip Technology)
dispositivo **SINGLE CHIP**
architettura **RISC - HARVAD**

Caratteristiche:

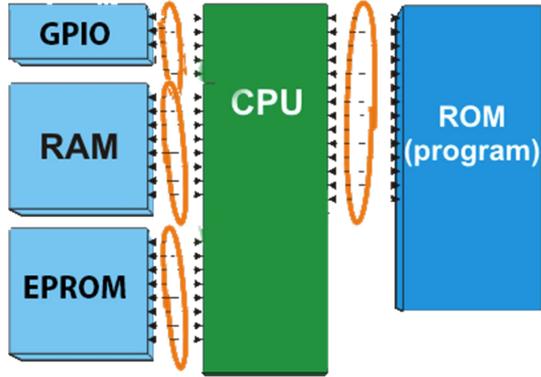
- clock: 20 MHz -16 MHz Arduino
- 8 bit
- ROM: 32 KByte (area per i programmi)
- RAM: 2 KByte (area per i dati)
- EEPROM: 1 KByte (area per dati permanenti)
- 20 porte I/O (PIN)



microcontrollore



Harvard



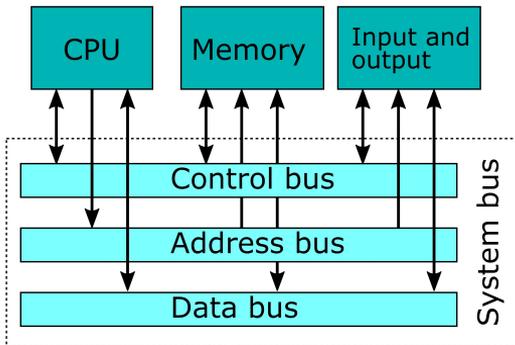
i microprocessori (CPU) sono classificati in due distinti modi:

architettura **Harvard**

vi è una netta separazione tra la memoria ove risiedono i programmi e la memoria dati.

Questo semplifica i circuiti e migliora la velocità con macchine a basse prestazioni.

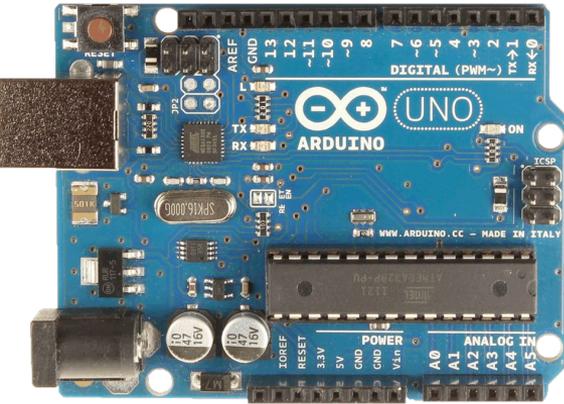
von Neumann



architettura **von Neumann**

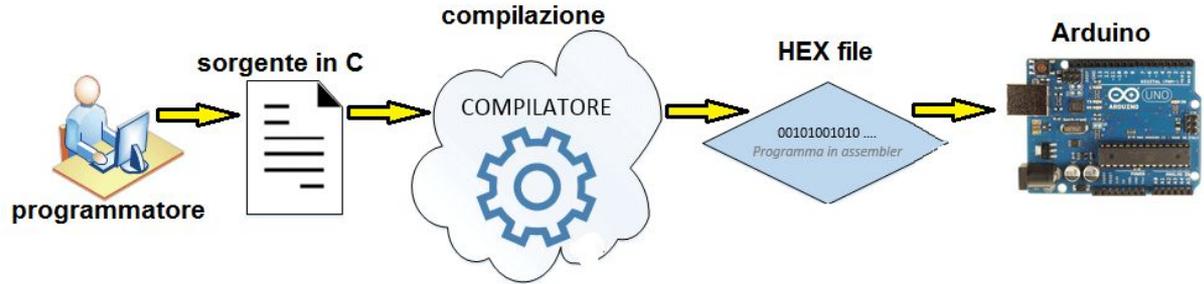
non vi è distinzione tra le aree di memoria, esse sono condivise dagli stessi circuiti logici.

che cosa è



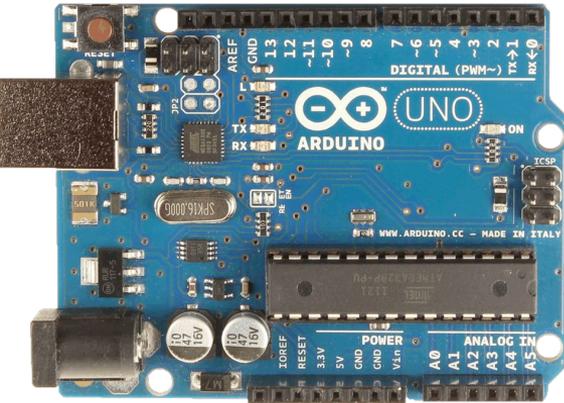
è un **SISTEMA DI SVILUPPO**

- linguaggio di programmazione semplificato
- editor di codice
- compilatore
- programmatore
- boot loader
- scheda elettronica (ATmega328, esp8266, esp32...)
- simulatore



è modulare ciascun componente può essere sostituito, implementato, ampliato, migliorato

Linguaggio



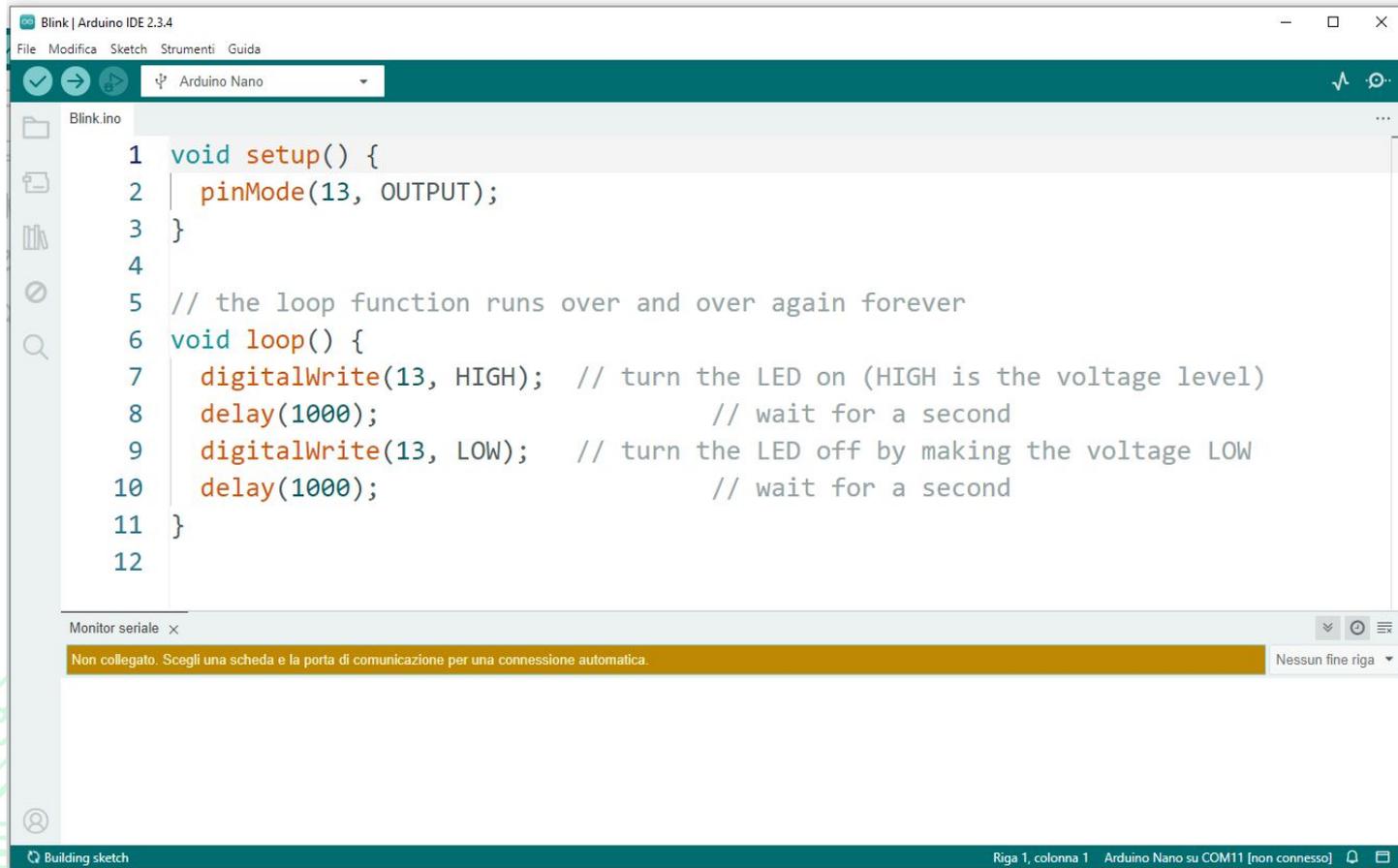
codice: **Arduino IDE** (C++ easy)
Arduino Cloud (C++ easy)
ATmel Studio (C++, Assembly...)

visuale: **mBlock** (simil Scratch)
Scratch S4a
Visualino (simil Scratch)
Ardublock
MiniBloq
Snap 4 A

simulatore: **Tinkercad** (in cloud, simil Scratch, C++)

(**IDE**: integrated development environment)

Arduno IDE - C++ easy



```
Blink | Arduino IDE 2.3.4
File Modifica Sketch Strumenti Guida
Arduino Nano
Blink.ino
1 void setup() {
2   pinMode(13, OUTPUT);
3 }
4
5 // the loop function runs over and over again forever
6 void loop() {
7   digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
8   delay(1000);           // wait for a second
9   digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
10  delay(1000);           // wait for a second
11 }
12

Monitor seriale x
Non collegato. Scegli una scheda e la porta di comunicazione per una connessione automatica. Nessun fine riga
```

Building sketch Riga 1, colonna 1 Arduino Nano su COM11 [non connesso]



mBlock

mBlock v5.4.0

makeblock | mBlock



File Modifica

Senza titolo

Salva

Pubblica



Dispositivi

Sprite

Sfondo



mBot



Arduino ...



aggiungi



Connetti il tuo dispositivo

[Come utilizzare il dispositivo?](#)

- Pin
- porta seriale
- Variabili e Liste
- Sensore
- Situazioni
- Controllo
- Operatori
- Variabili
- Miei blocchi

attendi 1 secondi

ripeti 10

per sempre

se allora

se allora

altrimenti

attendi fino a quando

ripeti fino a quando

quando si avvia Arduino Uno

per sempre

∞ impostare il pin digitale 9 uscita come high

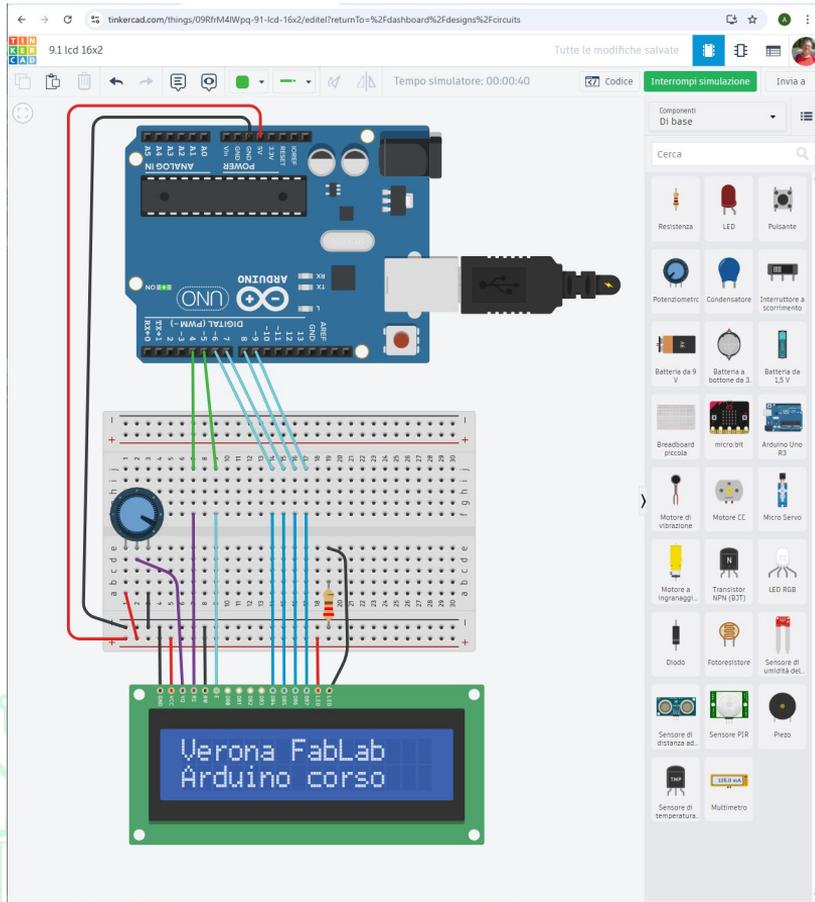
attendi 1 secondi

∞ impostare il pin digitale 9 uscita come low

attendi 1 secondi

AP

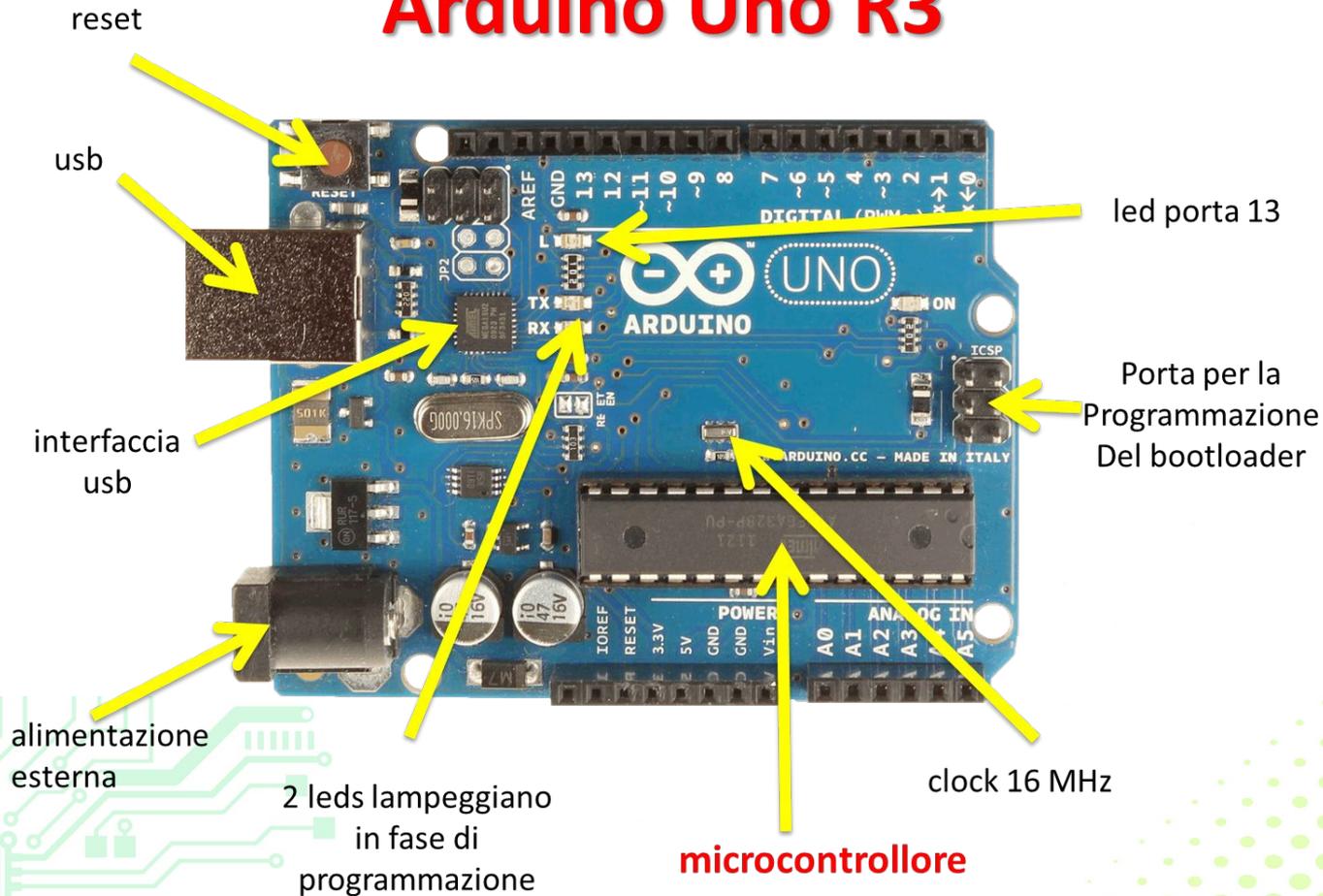
Tinkercad



The screenshot shows the Tinkercad code editor with a block-based program and its corresponding C++ code. The block-based code includes blocks for setting pin modes, writing to the LED, and a loop with delays. The C++ code is a standard Arduino sketch for an LED.

```
1 // C++ code
2
3 void setup ()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6
7
8 void loop ()
9 {
10  digitalWrite(LED_BUILTIN, HIGH);
11  delay(1000); // Wait for 1000 millisecond(s)
12  digitalWrite(LED_BUILTIN, LOW);
13  delay(1000); // Wait for 1000 millisecond(s)
14 }
```

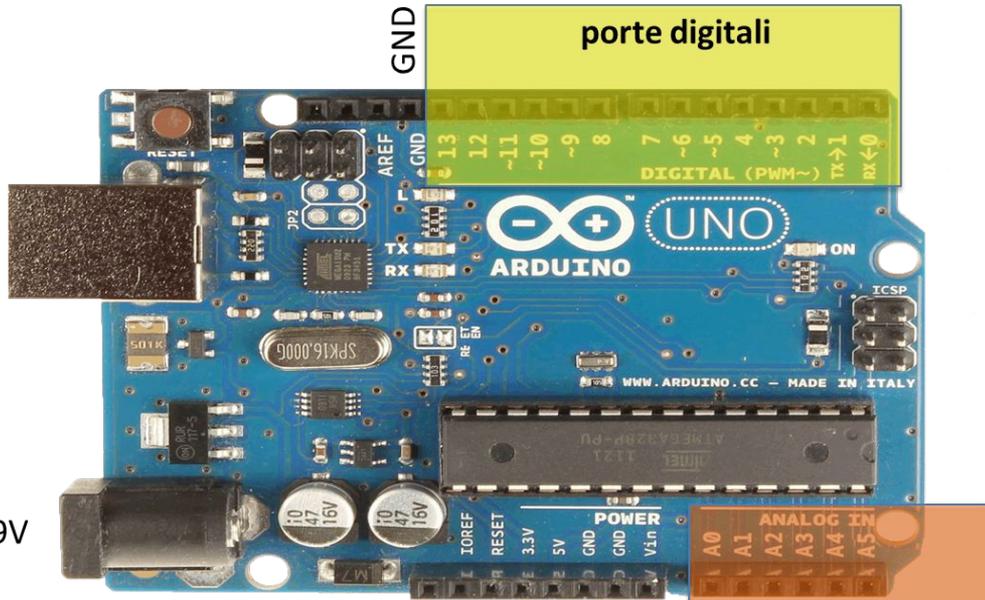
Arduino Uno R3



Arduino Uno R3

USB
Programmazione
dal PC
alimentazione 5V

alimentazione 9V



reset
uscita 3.3v 50mA
alimentazione 5V
GND
alimentazione 9V

porte analogiche
o digitali

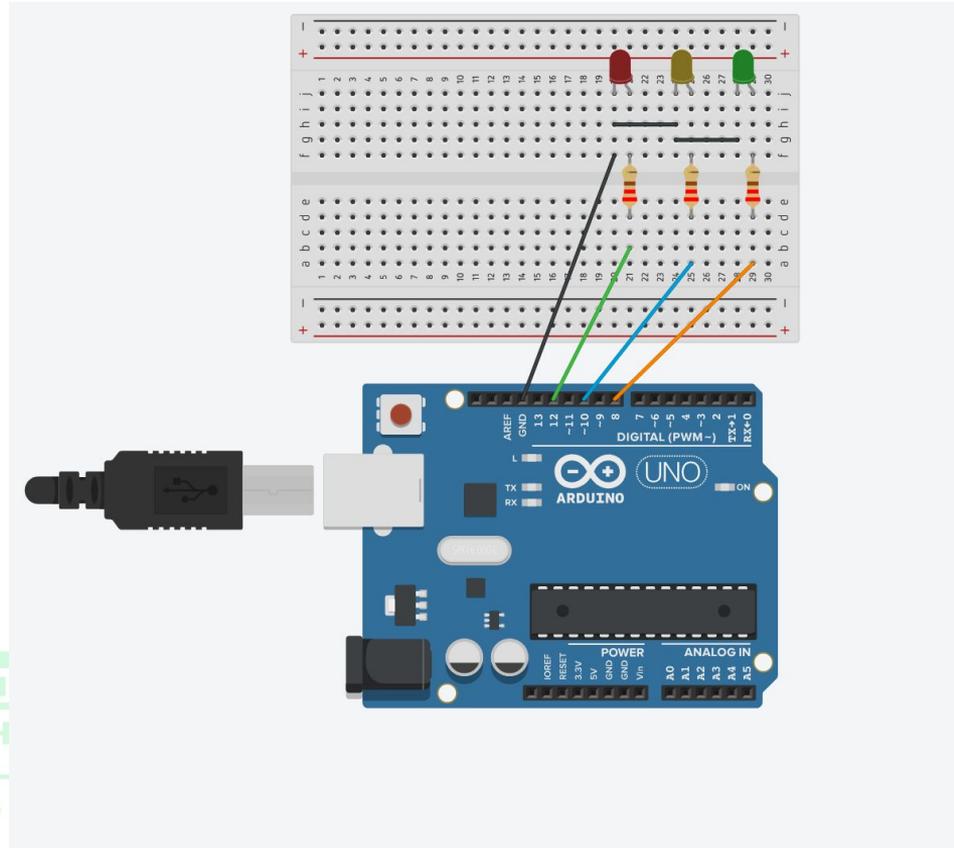


come guastare i PIN

- alimentare il chip di Arduino con più di 5 Volt
- invertire polarità dell'alimentazione
- fornire ai PIN (INPUT) tensioni maggiori di 5V
- prelevare da ciascun PIN (OUTPUT) più di 40 mA
- prelevare più di 200 mA su tutti i PIN
- collegare un singolo PIN direttamente a massa
- collegare due PIN tra loro (uno HIGH - l'altro LOW)



facciamo lampeggiare dei LED

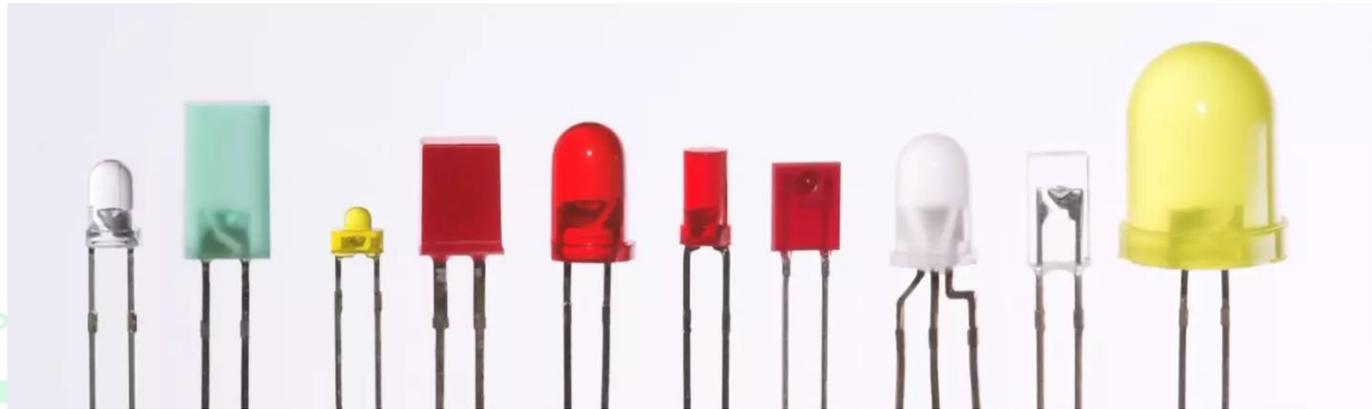


Led



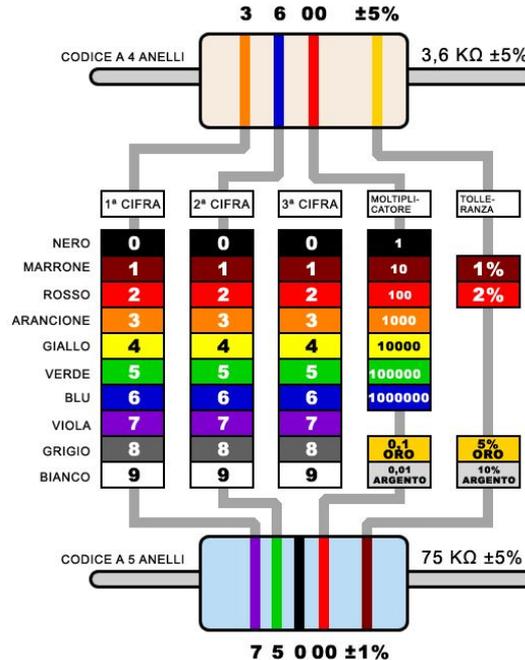
- è un semiconduttore, ha due poli (+ anodo, - catodo)
- è un diodo che emette luce se alimentato
- supporta una tensione massima di **1.8V** (dipende dal colore)
- supporta una corrente massima di **20 mA**
- i piedini di Arduino forniscono 5 Volt (troppi!)
- deve essere inserita una resistenza per ridurre la tensione e proteggere il led
- i valori di resistenza vanno da 220 ohm a 470 ohm

Caduta di tensione		
Infrarosso		1,3
Rosso		1,8
Arancio		2,0
Giallo		1,9
Verde		2,0
Blu		3,0
Ultravioletto		3,0
Bianco		3,0



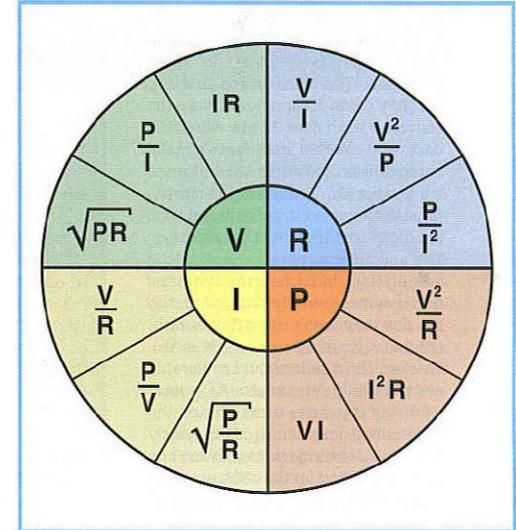
Resistenze

- una resistenza è un componente elettronico **passivo**
- pone un **freno** al passaggio della corrente
- il suo valore si misura in ohm



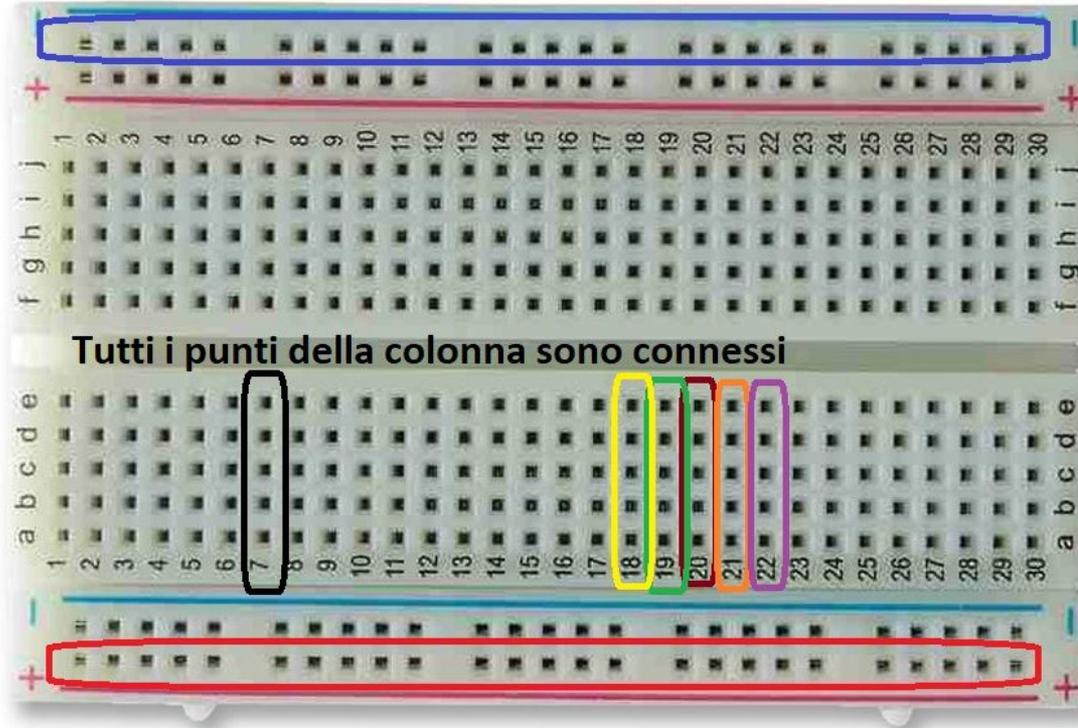
$$I = \frac{V}{R}$$

I = Corrente in Ampere (A)
V = Tensione in Volt (V)
R = Resistenza in Ohm (Ω)

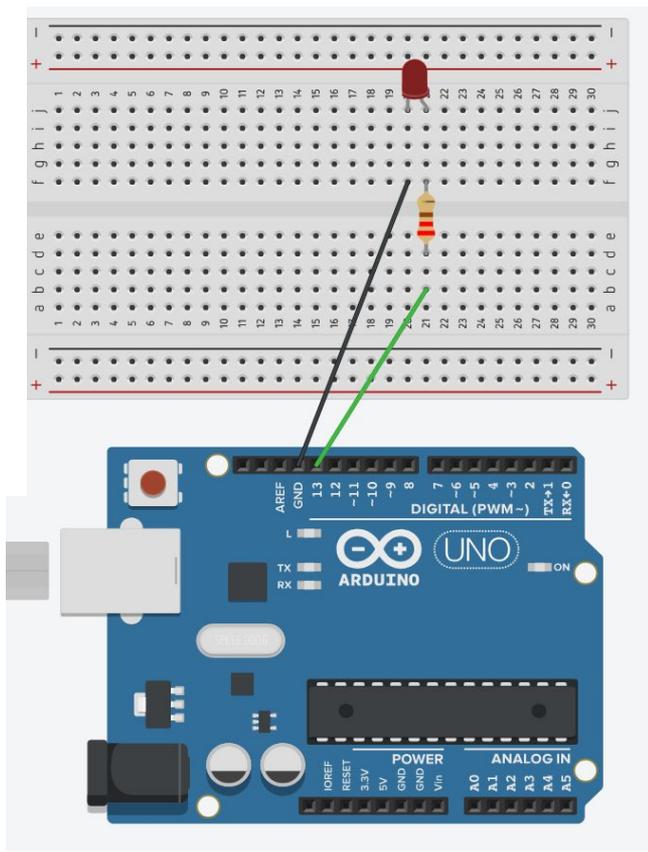
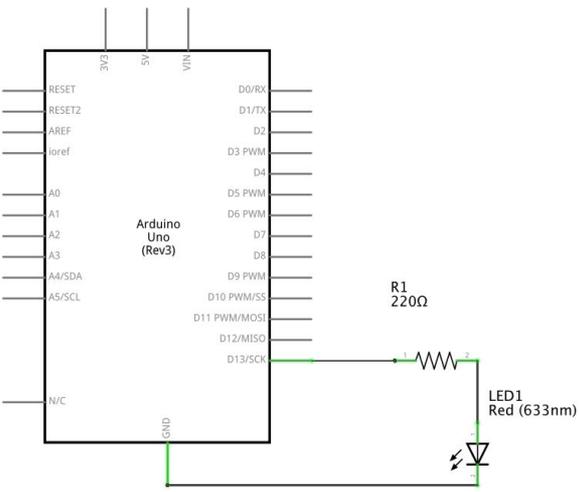


Breadboard

Tutti i punti della linea sono connessi insieme

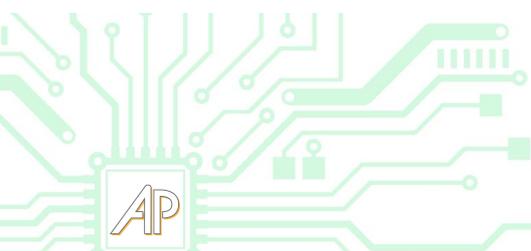


Tutti i punti della colonna sono connessi

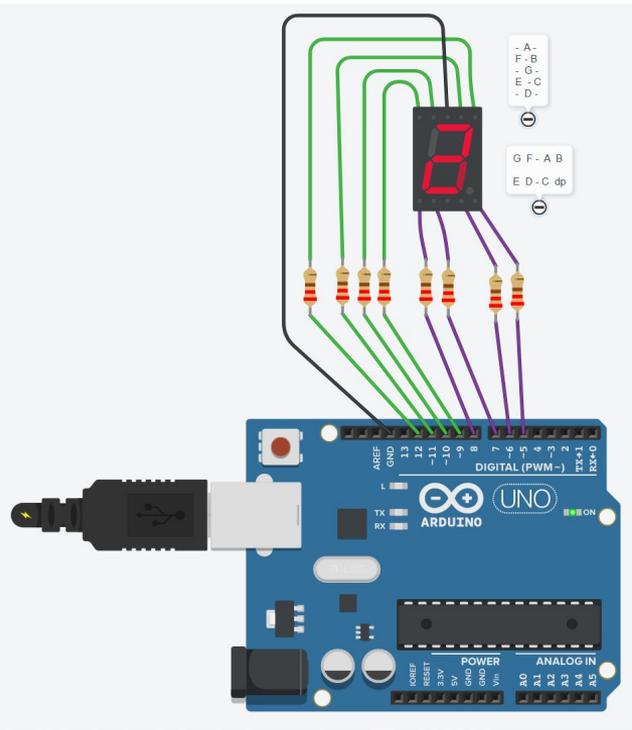


```
void setup() {
  pinMode(12, OUTPUT);
}
```

```
void loop() {
  digitalWrite(12, HIGH);
  delay(1000);
  digitalWrite(12, LOW);
  delay(1000);
}
```



Display 7 segmenti



- prendiamo 8 led (8 resistenze 220)
- li disponiamo in forma opportuna, 1 per il punto
- utilizziamo 8 PIN in modalità OUTPUT
- accendendo i led visualizziamo dei numeri

			A	B	C	D	E	F	G
F	A	0	1	1	1	1	1	1	0
	B <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td>	1	0	1	1	0	0	0	0
E	G	2	1	1	0	1	1	0	1
	C <td>3</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td>	3	1	1	1	1	0	0	1
	D <td>4</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td>	4	0	1	1	0	0	1	1
		5	1	0	1	1	0	1	1
		6	0	0	1	1	1	1	1
		7	1	1	1	0	0	0	0
		8	1	1	1	1	1	1	1
		9	1	1	1	0	0	1	1