

Introduzione al coding con Arduino

Gerardo Giardino/Maurizio Simeoni

10 Dicembre 2018

Premessa

**Il 65% dei bambini che iniziano
le elementari farà un lavoro
che oggi non esiste.**

Lo scenario è in continuo cambiamento

- Il mondo reale e quello digitale sono sovrapposti
 - Le regole del primo non sempre valgono nel secondo
 - La velocità del cambiamento mette fuori uso i canali tradizionali
 - Il lavoro si è trasformato e continua a trasformarsi
 - Il Digital Divide è un fattore di esclusione estremamente critico
- 
- A large, solid blue downward-pointing arrow indicating a logical flow or consequence.
- È importante per tutti comprendere come la trasformazione portata dal digitale influenzi il nostro mondo (Sicurezza, dati sensibili, hacker, social ...)

CoderDojo

Quello che facciamo con il CoderDojo, e che suggerisco a voi, è avviare un percorso di avvicinamento al digitale per i nostri ragazzi e ragazze poiché sarà fondamentale per tutti loro la capacità di utilizzare **consapevolmente** la tecnologia.

Elementi di Robotica & Arduino

Vocabolario essenziale

Input = ingresso

Output = uscita

Pin = un connettore di Input o di Output

Analogico = grandezze continue che rappresentano per «analogia» le variabili di un sistema

Digitale = grandezze che assumono solamente due valori o stati (1/0, High/Low, Vero/Falso)

Sketch = il programma che viene trasferito alla scheda Arduino (e qui eseguito)

USB = Universal Serial Bus

IDE = Integrated Development Environment

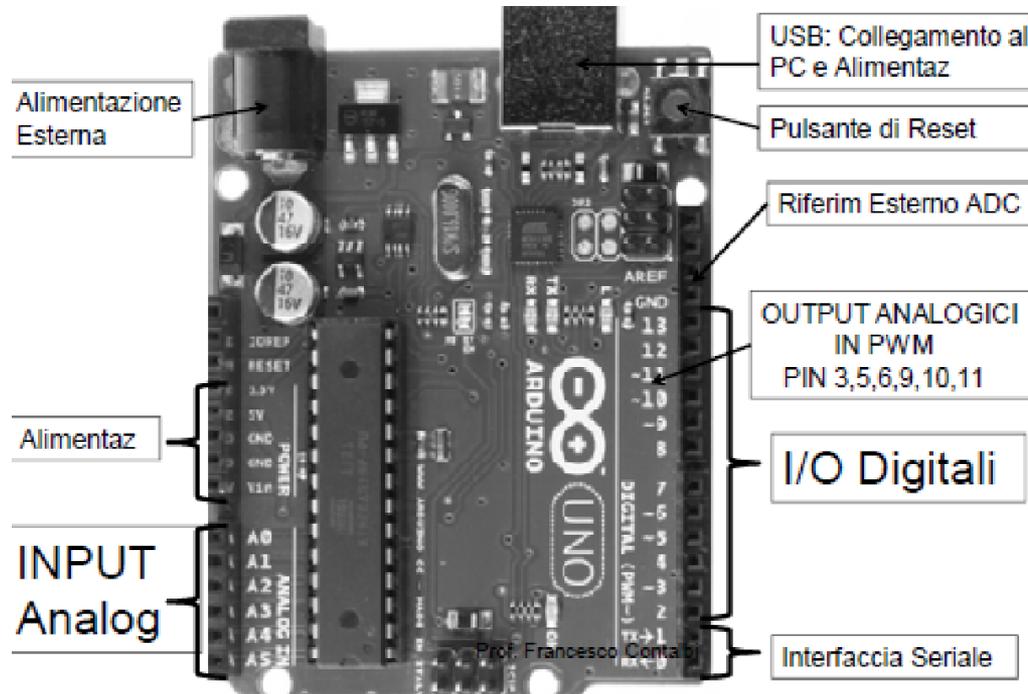
CONOSCIAMO ARDUINO

Arduino è un piccolo computer (in realtà è un microcontrollore)

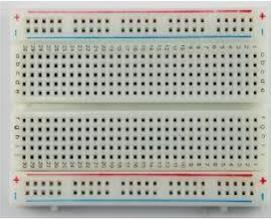
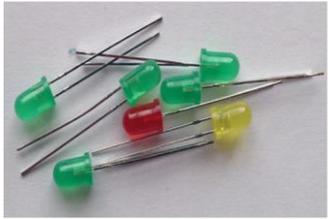
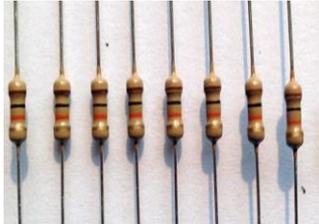
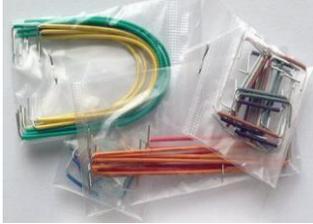
Può eseguire programmi e interagire con il mondo attraverso alcune interfacce:

- **Porta USB** – per collegarsi con un PC
- **Interfaccia Seriale** – per collegarsi con altri dispositivi
- **Pin Digitali**: Sono di tipo I/O (Input/Output). Riconoscono e producono solo 2 stati:
Acceso=1=ON, Spento=0=OFF
- **Pin Analogici**: Sono di tipo I (solo Input). Riconoscono un numero elevato di stati
(1024=10bit)

LA SCHEDA ARDUINO

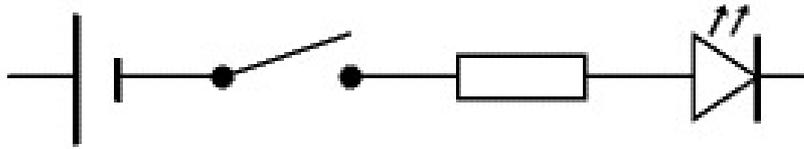


L'ambiente di sviluppo - Hardware

Immagine	Nome	Immagine	Nome
	Breadboard		LED
	Cavo USB		Resistenze
	ARDUINO UNO		Ponticelli

Conosciamo le componenti di un circuito

Esempio di simboli elettronici

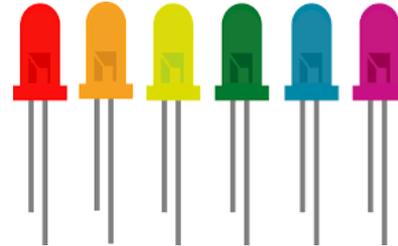


Batteria

Interruttore

Resistore

LED

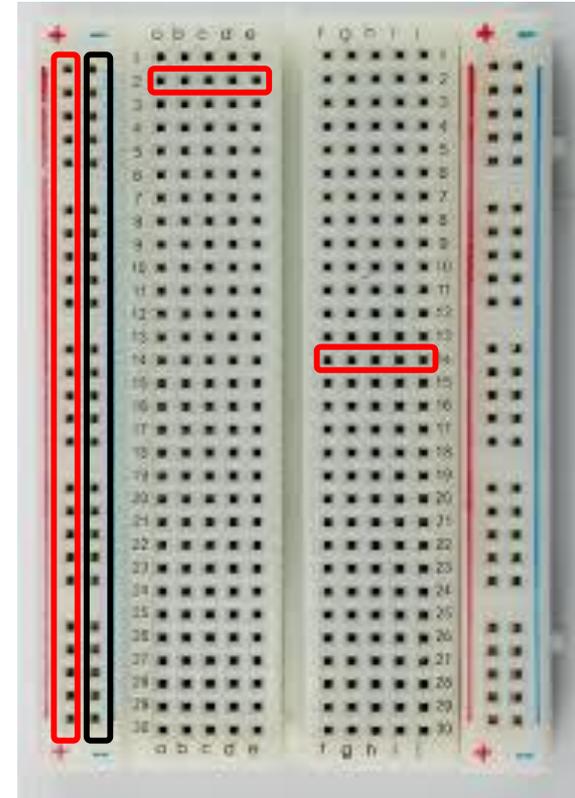


La Breadboard

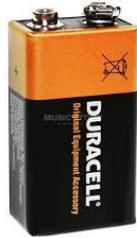
La Breadboard è un componente molto importante
Che consente di creare i circuiti elettronici senza
saldature e di fare modifiche «al volo»

I fori sono distanziati di 2,54 mm (1/10 di pollice) che
è la distanza tra i piedini dei circuiti integrati

I fori sono collegati tra loro come in figura →



Vtaggio, Corrente, Resistenza



Ha un estremo da cui esce la corrente (Polo Positivo) ed uno da cui entra (Polo Negativo)

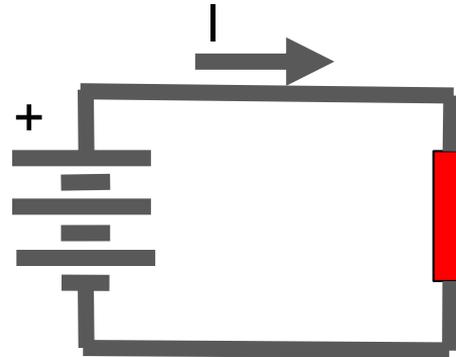


La corrente, la tensione e la resistenza sono collegate:

V

$$V = R \times I$$

Legge di Ohm

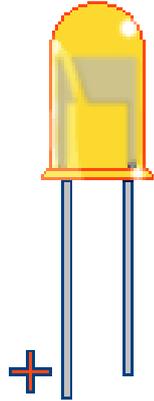


Resistenza



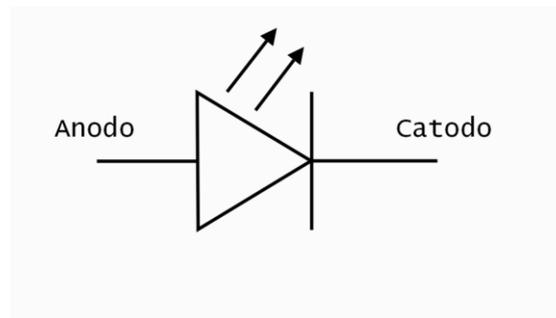
Regola/limita la corrente elettrica come il rubinetto fa per l'acqua

LED – Light Emitting Diode



anodo	+	gambo lungo		resistenza e poi verso il positivo della batteria
catodo	-	gambo corto	il catodo ha una specie di quadratino piatto nel corpo plastico del LED	Massa o terra (ground/GRD)

simbolo elettrico



L'ambiente di sviluppo - Software

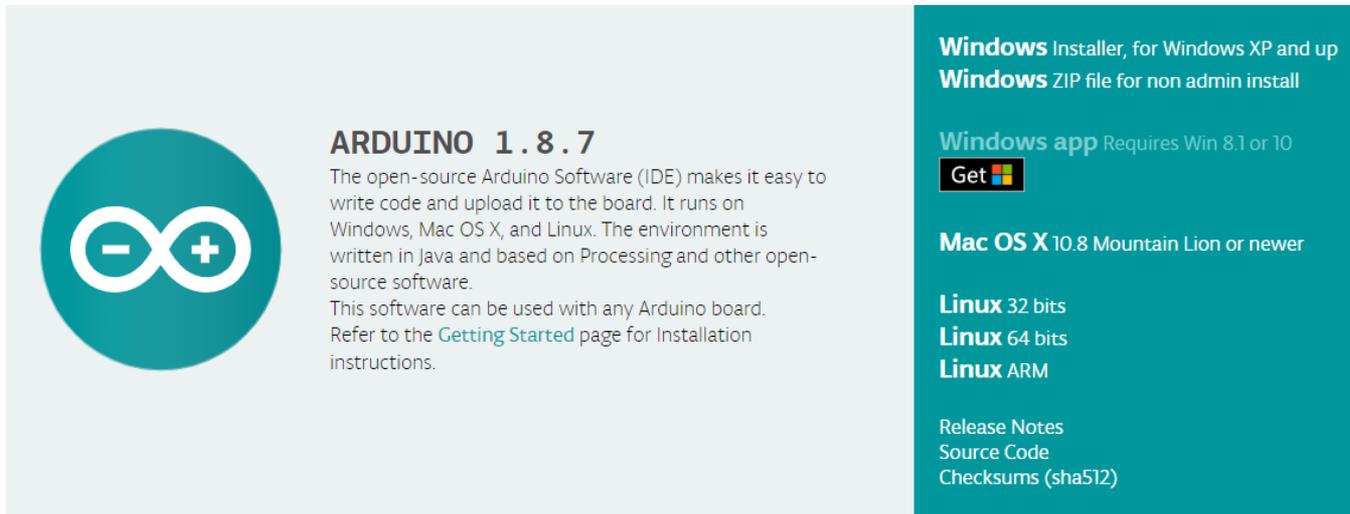
Il programma che viene usato per scrivere i programmi per Arduino (ambiente di sviluppo o IDE) si chiama **arduino.exe** ed è scaricabile dal sito

<http://arduino.cc/en/Main/Software>. Sito ufficiale della piattaforma Arduino.

I programmi (chiamati **Sketch**) vengono compilati con l'IDE (Integrated Development Environment) di Arduino caricati sulla piattaforma (tramite cavo USB) e registrati nel microcontrollore. Ciò ne permette l'esecuzione.

Download dell'ambiente di sviluppo - IDE

1. Scarichiamo l'ambiente di sviluppo da qui: <https://www.arduino.cc/en/Main/Software>



The screenshot shows the Arduino 1.8.7 download page. On the left, there is a teal circle containing the Arduino logo (an infinity symbol with a minus sign on the left and a plus sign on the right). To the right of the logo, the text reads: **ARDUINO 1.8.7**
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.

On the right side of the page, there are several download options:

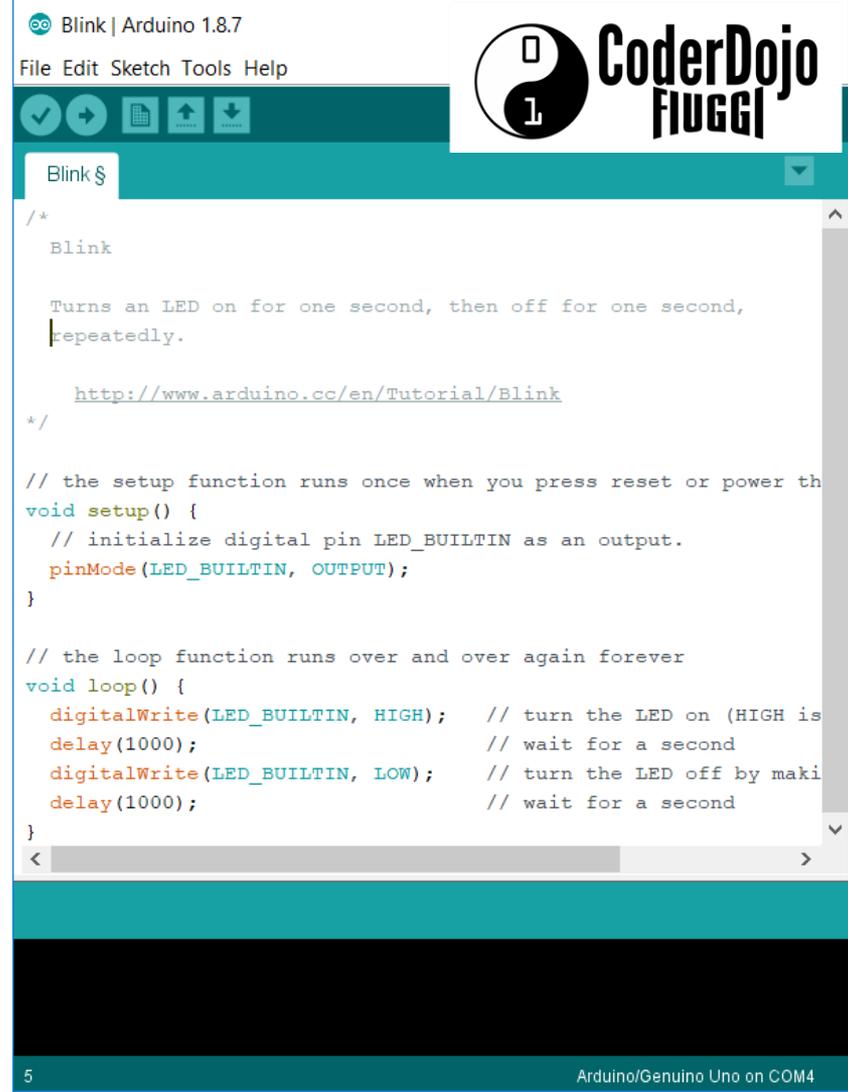
- Windows** Installer, for Windows XP and up
- Windows** ZIP file for non admin install
- Windows app** Requires Win 8.1 or 10
- Get** (with the Windows logo)
- Mac OS X** 10.8 Mountain Lion or newer
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM

At the bottom of the right column, there are links for [Release Notes](#), [Source Code](#), and [Checksums \(sha512\)](#).

2. Doppio click su `arduino-xxx-.exe` per installare

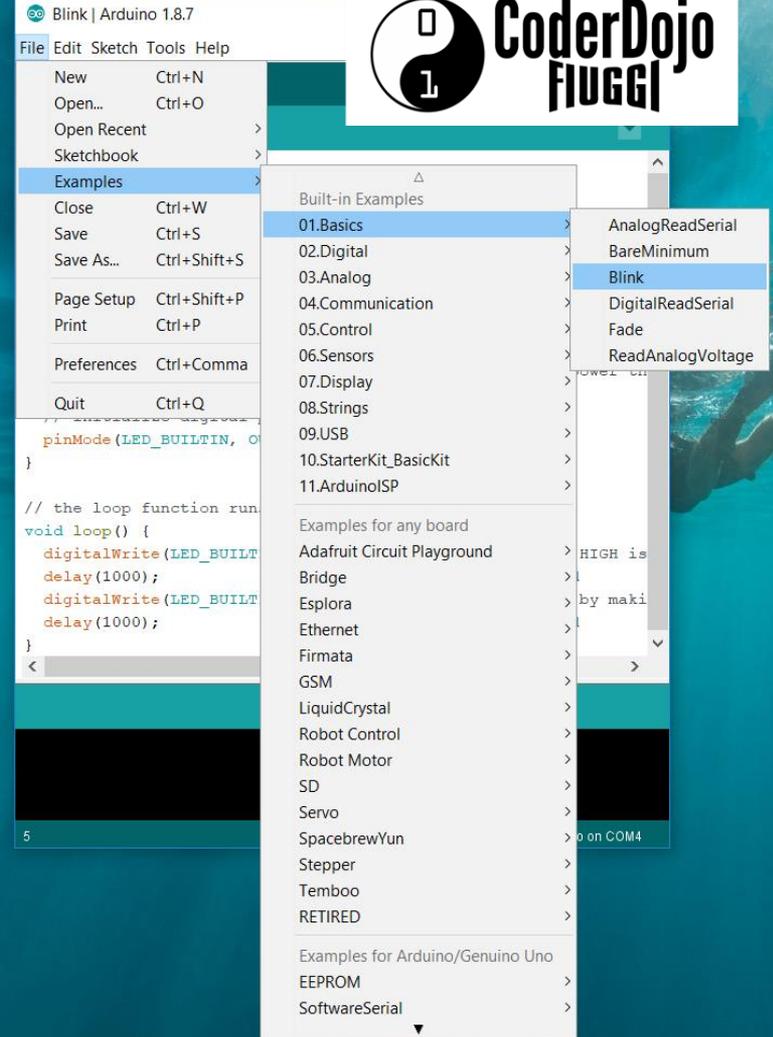
Come si presenta IDE

- E' un text editor → per scrivere e modificare il programma
- Possiamo verificare la sintassi
- Possiamo trasferire (upload) il programma sulla scheda (Lo sketch verrà tradotto automaticamente in linguaggio macchina)



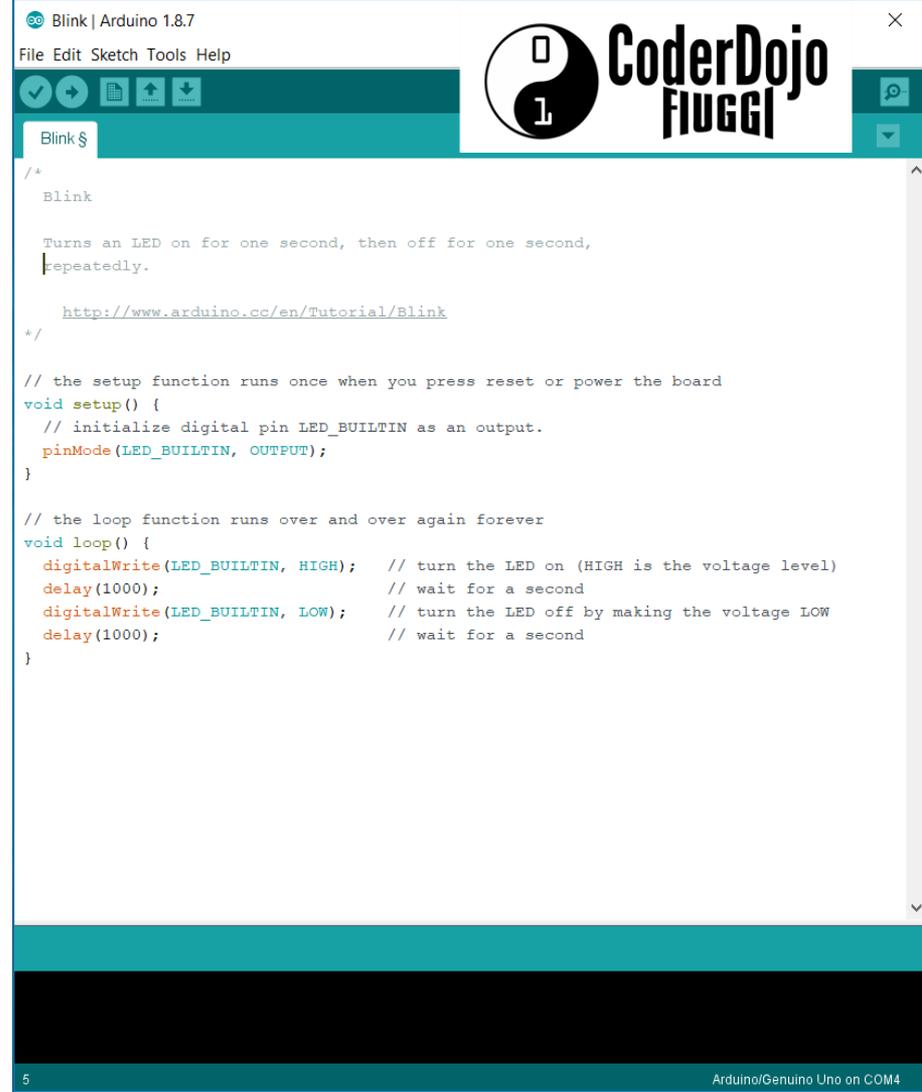
Primo esercizio

1. *Collegiamo PC e Arduino (USB)*
2. *Apriamo IDE*
3. *File*
4. *Esempi*
5. *Basic*
6. *Blink – doppio click*
7. *Verifichiamo il software (✓)*
8. *Lo carichiamo sulla scheda (👉)*
9. *Cosa succede?*



Analizziamo il programma

- Frasi di commento → `/* ...*/` oppure `//`
- I programmi sono composti da 2 funzioni:
void **setup()**: è la prima ad essere chiamata quando parte uno sketch. Viene utilizzata per inizializzare variabili, per impostare lo stato dei pin, per far partire le librerie da usare, per l'impostazione delle comunicazioni seriali.
- void **loop()**: esegue ciclicamente il programma definito al suo interno.



The screenshot shows the Arduino IDE interface. At the top, the title bar reads "Blink | Arduino 1.8.7". Below it is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". A toolbar contains icons for saving, undo, redo, and other functions. On the right side, there is a logo for "CoderDojo Fluggi" featuring a yin-yang symbol. The main area displays the code for the "Blink" sketch. The code includes a multi-line comment describing the program's purpose and a link to the Arduino tutorial. It defines two functions: `void setup()` and `void loop()`. The `setup()` function initializes the LED pin. The `loop()` function turns the LED on and off in a cycle, with 1000ms delays between each state change.

```
Blink | Arduino 1.8.7
File Edit Sketch Tools Help
Blink $
/*
 * Blink
 *
 * Turns an LED on for one second, then off for one second,
 * repeatedly.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Analizziamo il programma - SETUP

```
void setup() {  
  
    // initialize digital pin LED_BUILTIN as an output.  
  
    pinMode(LED_BUILTIN, OUTPUT);  
  
}
```

Analizziamo il programma - LOOP

```
// the loop function runs over and over again forever
```

```
void loop() {
```

```
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
```

```
  delay(1000); // wait for a second
```

```
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
```

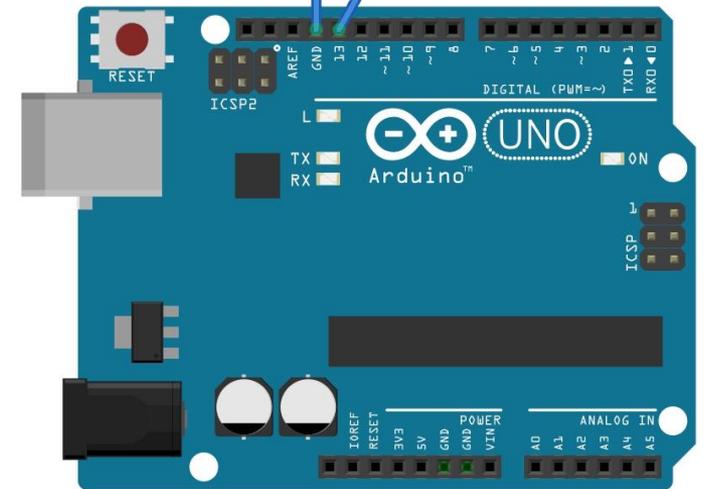
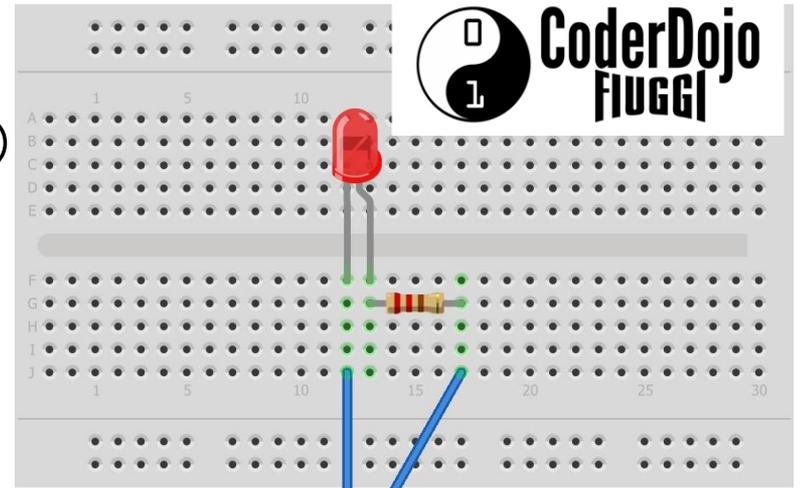
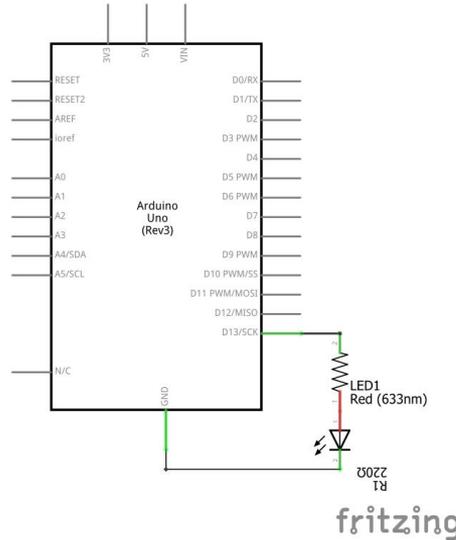
```
  delay(1000); // wait for a second
```

```
}
```

Cambiamo la frequenza del lampeggio

1. Raddoppiamo la frequenza del ciclo acceso spento
2. Dimezziamo la frequenza di acceso spento
3. 3 sec. Acceso, 10 sec spento, 1 sec acceso, 5 sec spento

Cominciamo a usare le mani 😊



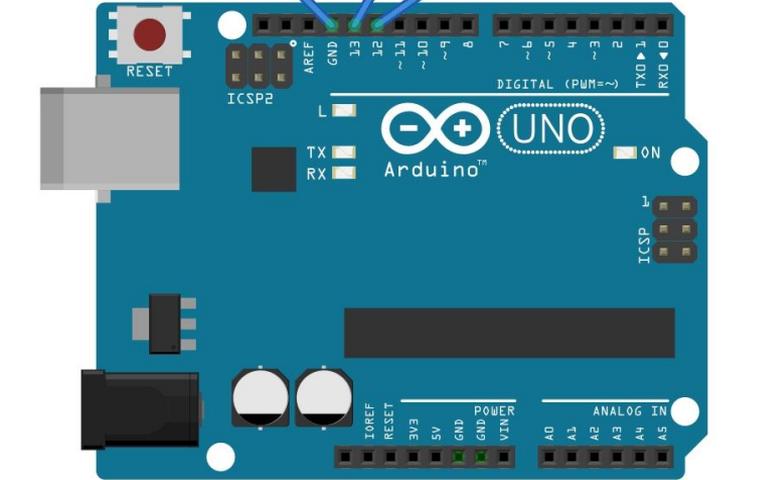
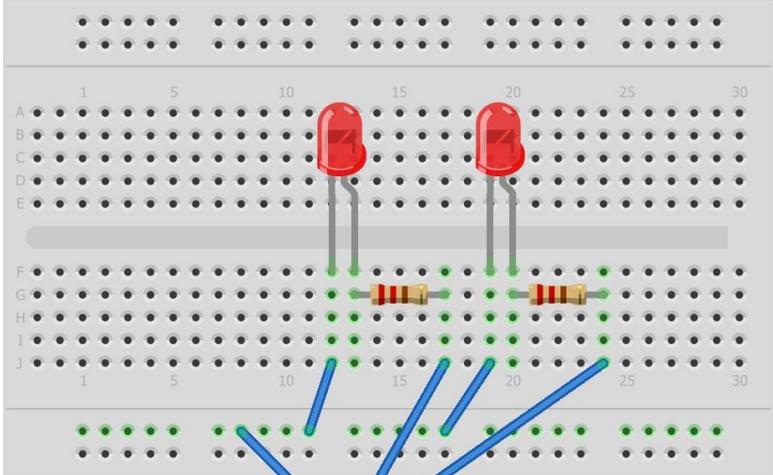
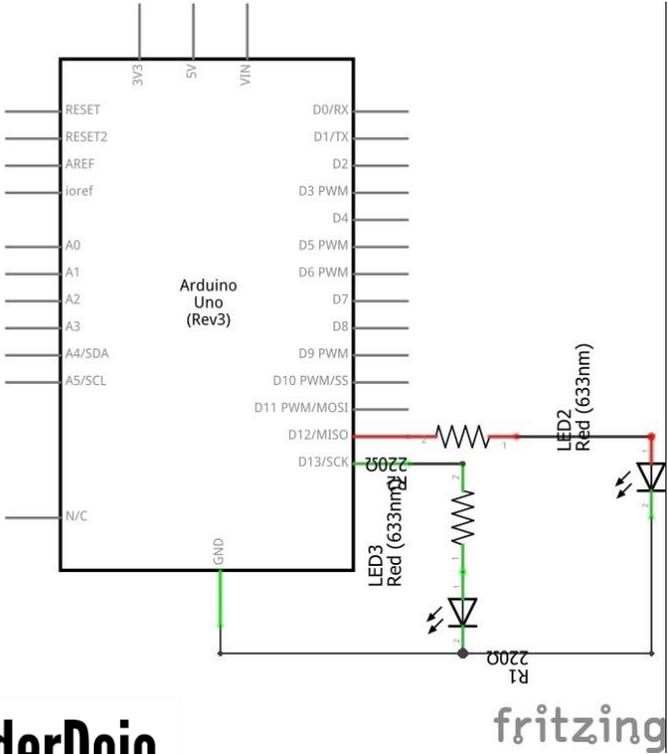
1. Mettiamo un LED esterno sul pin 13
2. poi sul pin 12

We Will Rock You



- Disegnare un circuito con 2 LED
- Scrivere un programma che accenda i LED seguendo la base ritmica del pezzo

Schema fisico



GRAZIE