



Istituto Comprensivo 3 di Modena

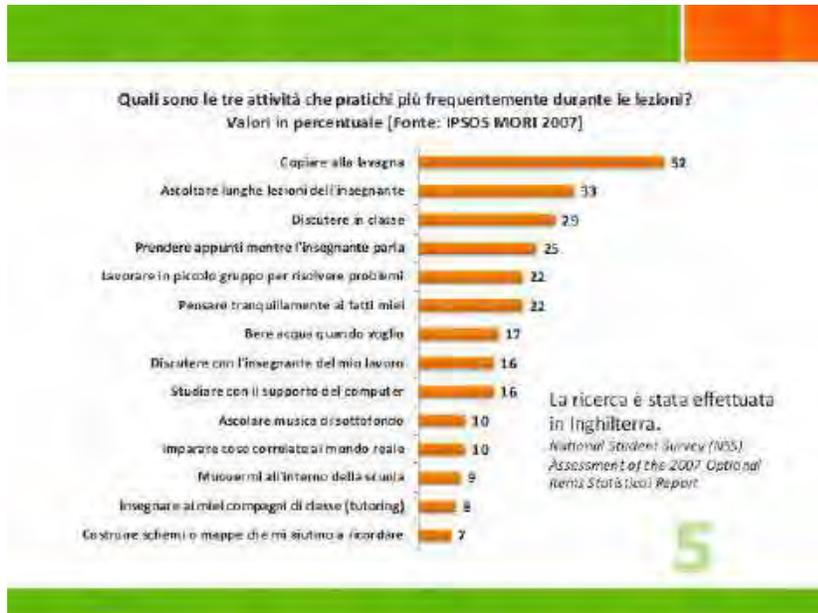
DA SCRATCH A C++: Insegnare a pensare tra Makeblock e Arduino



Docente di tecnologia presso
I.C. Rita Levi Montalcini di Salerno

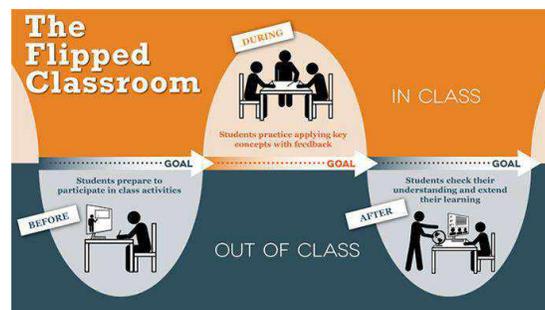


METODOLOGIE



METODOLOGIE

- IBSE (Inquiry Based Science Education)
- 5E (Engagement, Explore, Explain, Elaborate, Evaluate)
- IBL (Inquiry Based Learning)
- PBL (Problem Based Learning)
- Flipped Classroom
- Cooperative Learning





Howard Gardner (1943)

Jerome Bruner (1915-2016)

Seymour Papert (1928-2016)

Jean Piaget (1896-1980)

**Lev Semënovič Vygotskij
(1896-1934)**



Howard Gardner (1943)

Il progetto della mia scuola ideale del futuro si basa su due ipotesi: la prima è che **non tutti hanno gli stessi interessi e le stesse capacità**; non impariamo tutti allo stesso modo. La seconda ipotesi può far male: **è che ai nostri giorni nessuno può imparare tutto ciò che c'è da imparare.**

Nella maggior parte delle nazioni del pianeta, le scuole sono organizzate in modo uniforme. Le stesse materie sono insegnate e valutate in modo molto simile. **Sembra ingiusto essere in grado di trattare tutti gli studenti come se fossero uguali.**

L'intelligenza, ciò che consideriamo azioni intelligenti, viene modificata nel corso della storia. **L'intelligenza non è una sostanza nella testa come l'olio in un serbatoio di petrolio.** È una raccolta di potenzialità che sono state completate.

È molto importante valutare le intelligenze, **ma i test standardizzati non sono la soluzione.**

Ogni essere umano ha una combinazione unica di intelligenza. Questa è la sfida educativa fondamentale: **stimolare ogni studente in modo personalizzato**



Jerome Bruner (1915-2016)

Dove si trova la conoscenza? I bambini di solito cominciano con il **dare per scontato che l'insegnante possieda la conoscenza e la trasmetta alla classe**. Se si creano le condizioni opportune, imparano presto che anche altri componenti della classe potrebbero possedere delle conoscenze, e che queste conoscenze possono essere condivise. In questa seconda fase, la conoscenza esiste nel gruppo – ma in modo inerte. È possibile allora vedere la discussione di gruppo come un modo di creare conoscenza invece che semplicemente come un modo per scoprire chi possiede quali conoscenze?

3.5 Le nuove tecnologie – il cosa e il come dell'apprendimento

La rivoluzione digitale ha cambiato i modi di apprendere e tolto alla scuola il monopolio di come si impara: è una sfida gigantesca.

Ma non è certo solo una questione di mezzi con i quali si impara. Oggi tutte le discipline - sia teoriche che pratiche - sono, infatti, caoticamente parte della rete e sono accessibili in mille forme, rapidamente. Con la possibilità, ulteriore, di essere manipolate, variate, confuse, confrontate, espansive e ricollocate anche in termini produttivi, *on demand*, con un'attenzione a una domanda sempre più differenziata e resa anche singolare, personale

Lo stesso modo di imparare - il funzionamento del cervello umano - viene chiamato in causa: organizzazione della memoria, presenza simultanea di molti codici, compresenza di procedure analogiche e logiche, relazione immediata tra produzione costruita e fruita, etc. Questa è la prima generazione di docenti ed educatori che ha perso il monopolio delle

conoscenze e dei mezzi per trasmetterle e che si misura, al contempo, con l'imparare, il produrre e il comunicare su vasta scala.

Ai docenti viene ora chiesto di insegnare a distinguere, scegliere, confrontare in mezzo a un mare di informazioni complesse e contraddittorie, valutando il sapere e le competenze che i giovani hanno acquisito in moltissimi modi, anche lontano dalla scuola e diversi da come loro hanno imparato. Un tempo-scuola e un tempo di apprendimento più ricco e flessibile è inevitabile e richiede, a sua volta, per i docenti, un tempo di preparazione del lavoro molto diverso dalla vecchia lezione: un modo che sia cognitivo, artigianale, produttivo, culturale insieme e che attraversi continuamente i confini tra scuola e fuori.


Ministero dell'Istruzione, dell'Università e della Ricerca

Una politica nazionale di contrasto del fallimento formativo e della povertà educativa

Cabina di regia per la lotta alla dispersione scolastica e alla povertà educativa



Dove si trova la conoscenza? I bambini di solito cominciano con il **dare per scontato che l'insegnante possieda la conoscenza e la trasmetta alla classe**. Se si creano le condizioni opportune, imparano presto che anche altri componenti della classe potrebbero possedere delle conoscenze, e che queste conoscenze possono essere condivise. In questa seconda fase, la conoscenza esiste nel gruppo – ma in modo inerte. È possibile allora vedere la discussione di gruppo come un modo di creare conoscenza invece che semplicemente come un modo per scoprire chi possiede quali conoscenze?

C'è un ulteriore passo da compiere, che ci porta a toccare uno degli aspetti più profondi della conoscenza umana. **Se nessun membro del gruppo "sa" la risposta, dove si può andare a "scovarla"?** È il balzo che porta a concepire la cultura come un magazzino, come un deposito di attrezzi o qualcosa di simile. Esistono cose note a tutti gli individui; più cose ancora sono conosciute dal gruppo possono essere scoperte tramite una discussione all'interno del gruppo; e molte più ancora sono immagazzinate in qualche altro posto – nella "cultura", per esempio nella testa delle persone più colte, nei manuali, nei libri, nelle mappe e così via.

Per definizione, praticamente nessuno in una cultura sa tutto quello che c'è da sapere su di essa. E allora **cosa dobbiamo fare quando non sappiamo come andare avanti? E quali sono i problemi che incontriamo nel reperire la conoscenza che ci serve?** Se sappiamo rispondere a questa domanda siamo sulla buona strada per capire cos'è una cultura. Non ci vorrà molto perché un bambino cominci a capire che **la conoscenza è potere**, o che è una forma di ricchezza, o che è una rete di sicurezza.“

L'obiettivo principale dell'educazione nelle scuole dovrebbe essere quello di **creare uomini e donne che siano capaci di fare cose nuove**, non soltanto di ripetere semplicemente ciò che le altre generazioni hanno fatto.

I sistemi naturali sono soggetti non solo a evoluzioni ma a rivoluzioni, cioè dipendono non solo da leggi e fattori causali, ma anche **da eventi non prevedibili con elevata probabilità e dalle decisioni individuali e collettive di esseri umani, mai regolate da leggi di tipo deterministico.**

Il problema fondamentale è: qual è l'obiettivo della scuola? È formare menti conformi a quanto già conosciamo – e sappiamo fino a che punto – e che si limitano a riprodurre e ripetere ciò che hanno fatto le generazioni precedenti, o **vogliamo formare menti innovative e capaci di nuove costruzioni in una area o in un'altra?**



Jean Piaget (1896-1980)



**Lev Semënovič Vygotskij
(1896-1934)**

L'insegnante deve assumere il ruolo di facilitatore, non di fornitore di contenuti.

La conoscenza è il prodotto dell'interazione tra la persona e l'ambiente, ma in quanto mezzo inteso come qualcosa di sociale e culturale, non solo fisico.

A mano a mano che ci sviluppiamo, smettiamo di imitare semplicemente il comportamento degli altri o di reagire automaticamente agli stimoli provenienti dall'ambiente.

Per comprendere il linguaggio degli altri, non è sufficiente comprenderne le parole, è necessario capirne il pensiero.

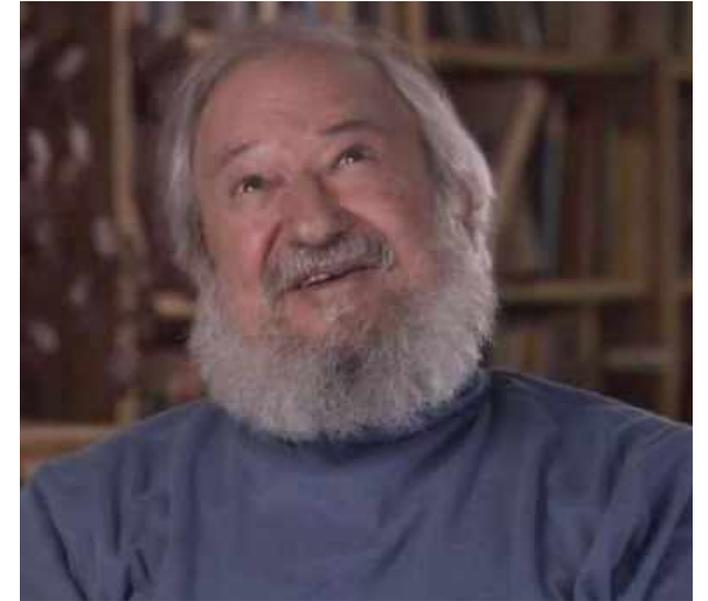
Non si risolve il problema applicando una regola, ma è il riflettere su un problema che stimola l'apprendimento.

E' possibile costruire efficacemente la conoscenza solo se chi apprende è "coinvolto" (engagement) nella costruzione di prodotti che abbiano un significato per sé e per la comunità.

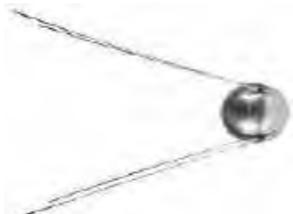
Le idee delle persone prendono forma e si trasformano quando trovano espressione attraverso differenti media, quando sono inserite in particolari contesti e quando si sviluppano al di fuori della mente individuale.

Ciò che rende la matematica ripugnante ai bambini non è la difficoltà, ma il procedere secondo un insensato rituale dettato dai tempi di un programma di studi prefissato.

Apprendimento vede il discente come soggetto, ma l'istruzione vede il discente come oggetto di un'attività esercitata da un maestro.



Seymour Papert (1928-2016)



1 min fa (1957)
Sputnik 1 in orbita

30 s fa (1981)
Shuttle

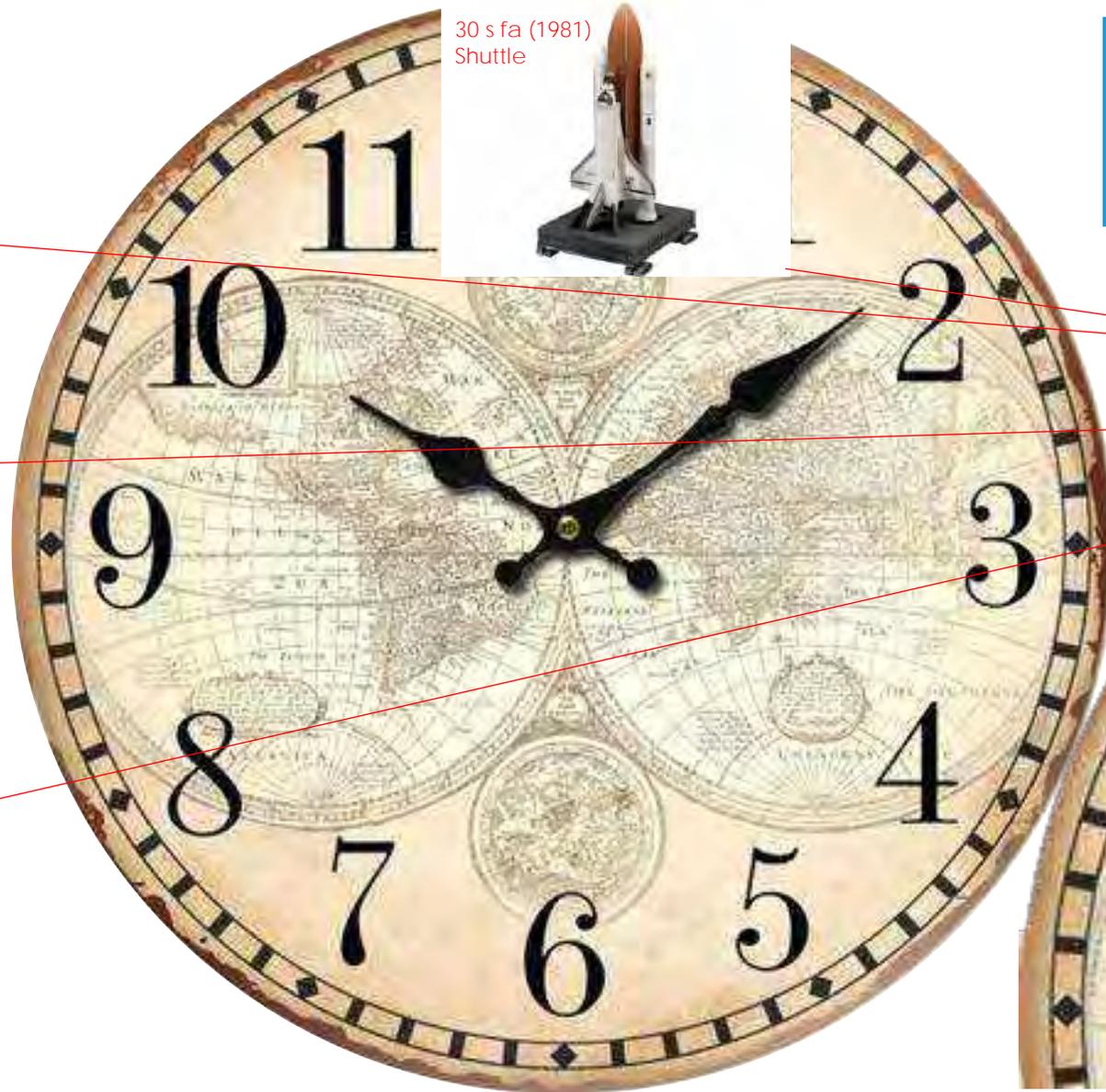
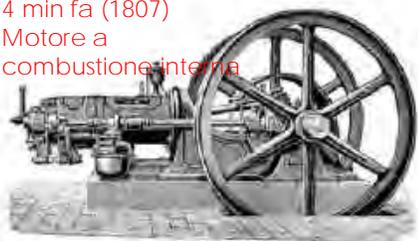


1 s fa (2010)
primo veicolo senza equipaggio

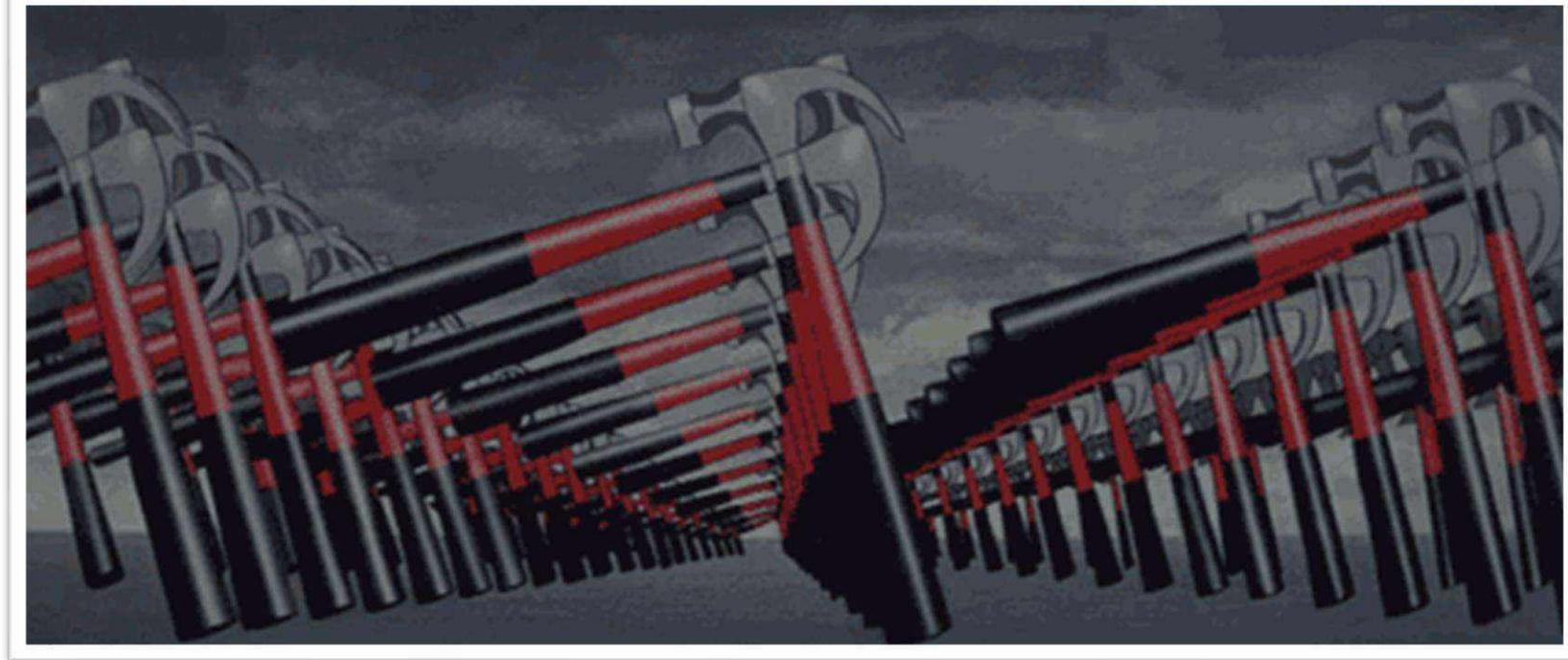


2,5 min fa (1885)
Prima automobile a motore

4 min fa (1807)
Motore a
combustione interna







Pink Floyd – The Wall - 1979

Imbuto di Norimberga
Germania XVI-XVII



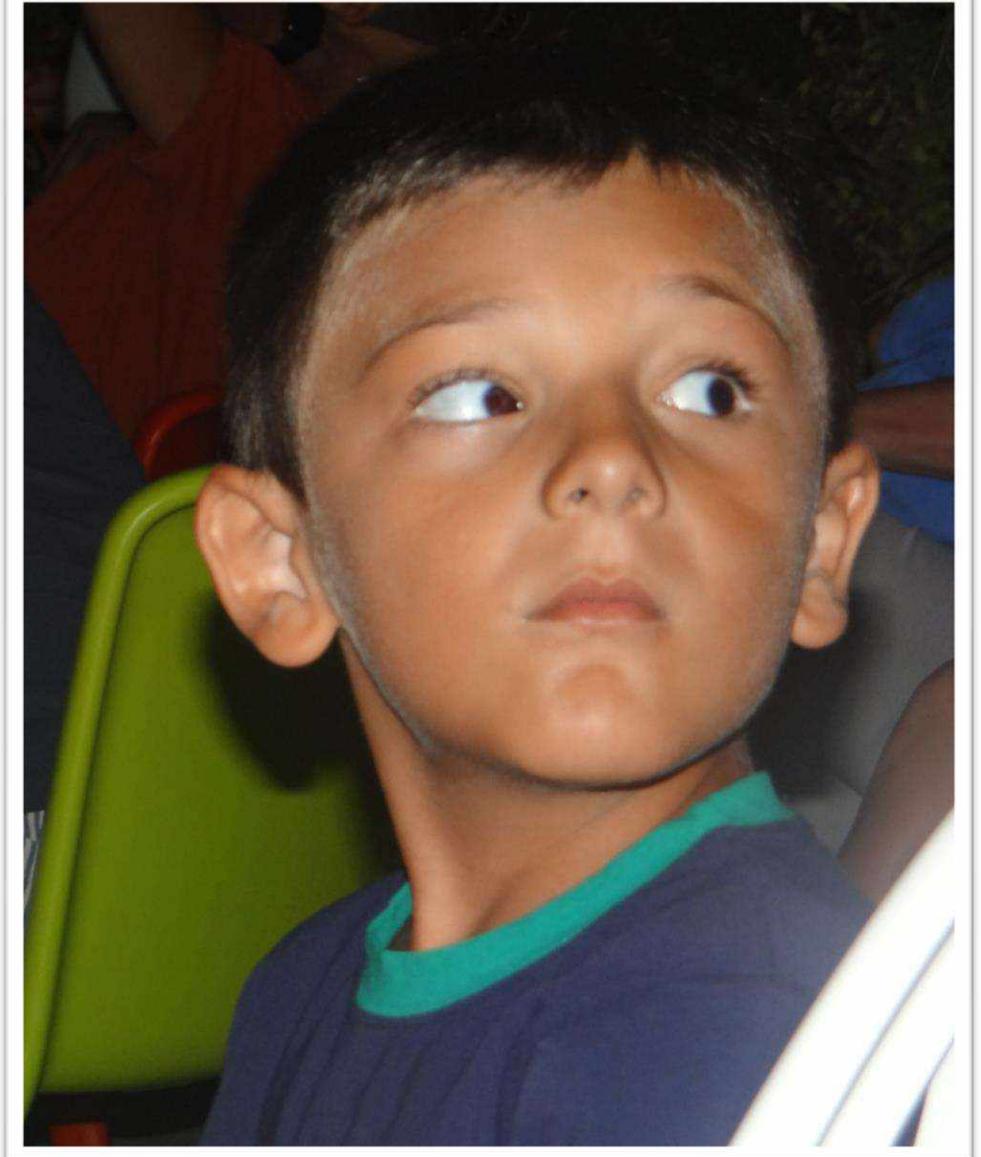
Mark Prensky

Digital Natives, Digital Immigrants,
pubblicato nel 2001

NATIVI DIGITALI



Immigrati digitali



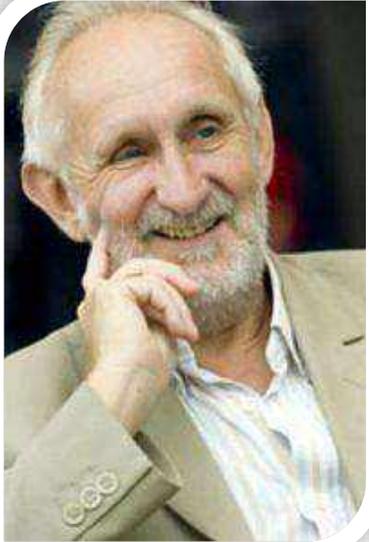
Nativi digitali





PRIMITIVI digitali





“The most important aspect of innovation is not technology, it is value. Innovation is the process of creating value starting from ideas.”



Joe Tidd and John Bessant
Managing Innovation





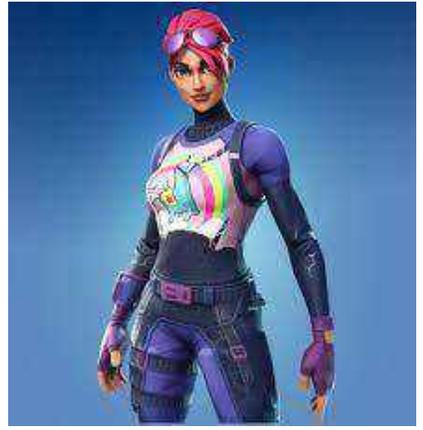


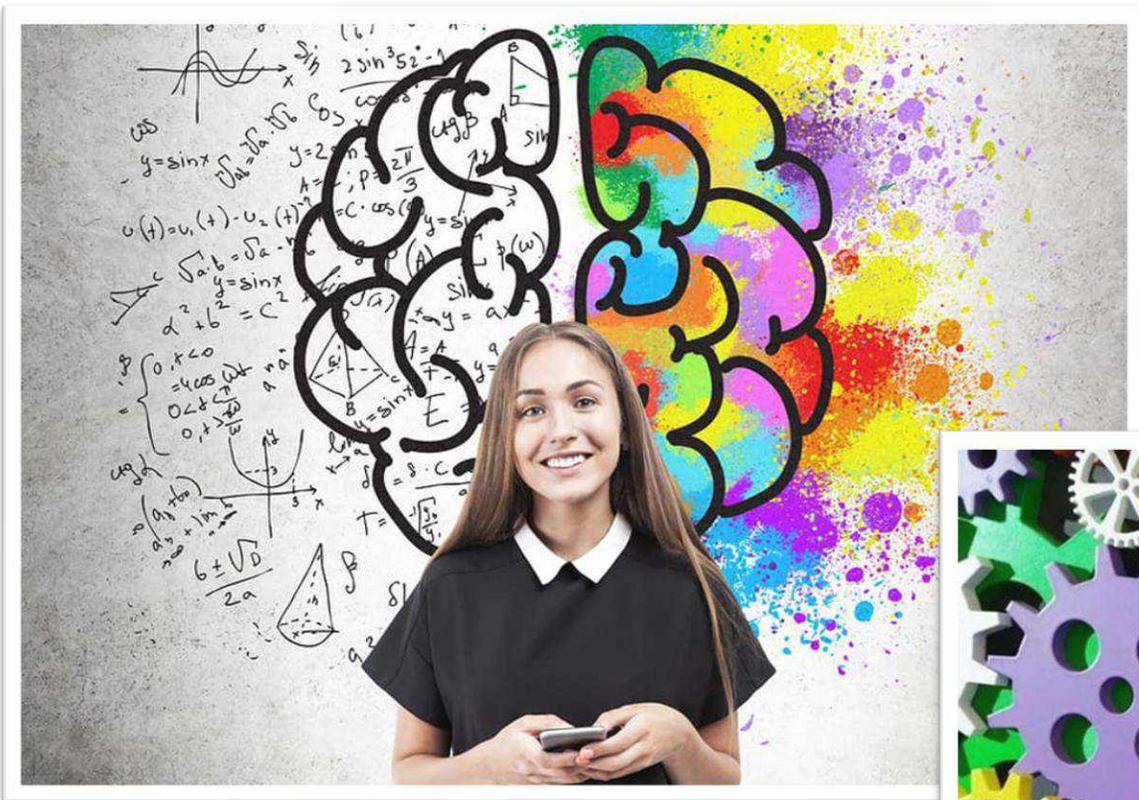
\$ 93.000.000

NFT
Non Fungible
Tokens



EVERDAYS – THE FIRST 5000 DAYS – Beeple (2021)





IMMAGINAZIONE

CREATIVITA'



Jannette Wing :

“Il pensiero computazionale rappresenta un atteggiamento e un complesso di attività che sono universalmente applicabili e che chiunque, non soltanto gli informatici, dovrebbe essere desideroso di apprendere e utilizzare”.



*“Non comprate un videogame:
fatene uno.
Non scaricate l'ultima app:
Progettatela.
Non usate il vostro telefono:
programmategelo”.*







LIVE IN VENICE

Pink Floyd

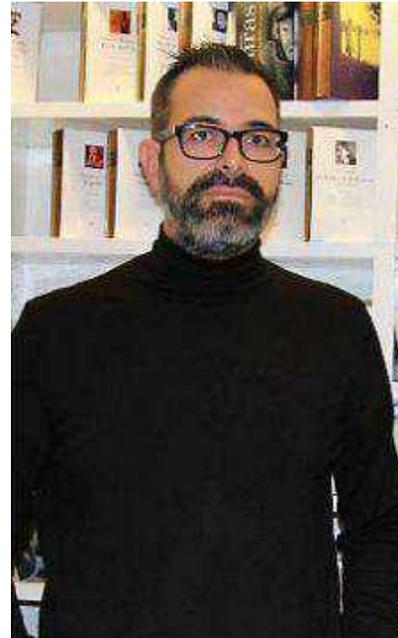
15 luglio 1989



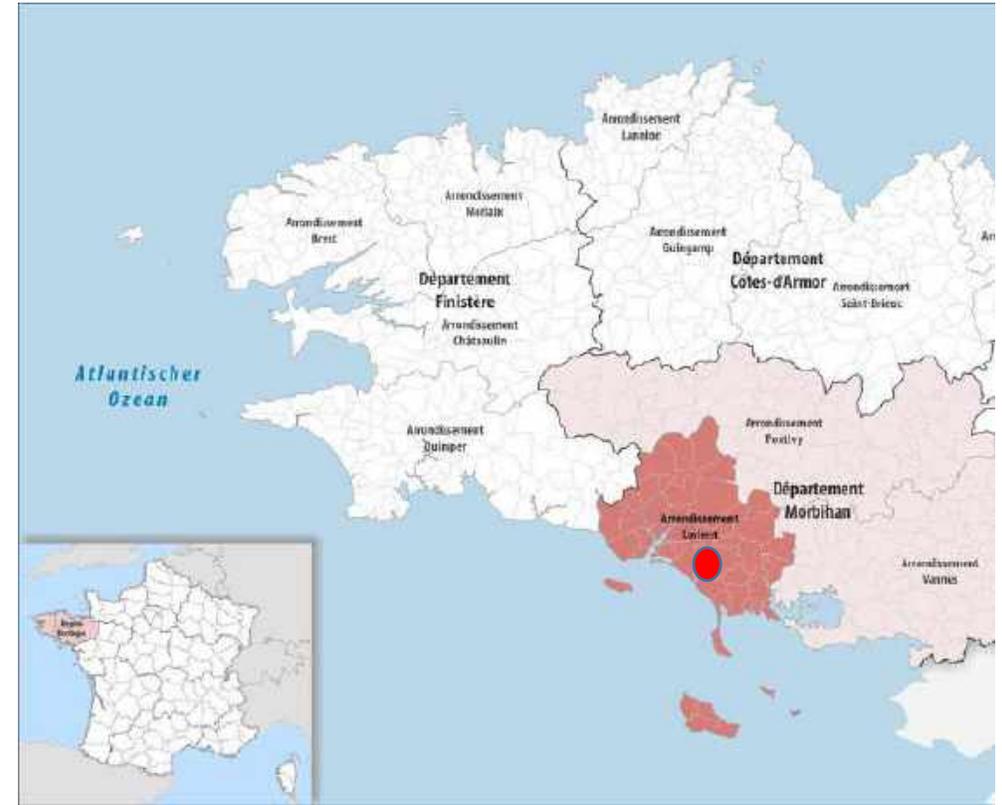
Valerie Trierweiler



Francois Hollande

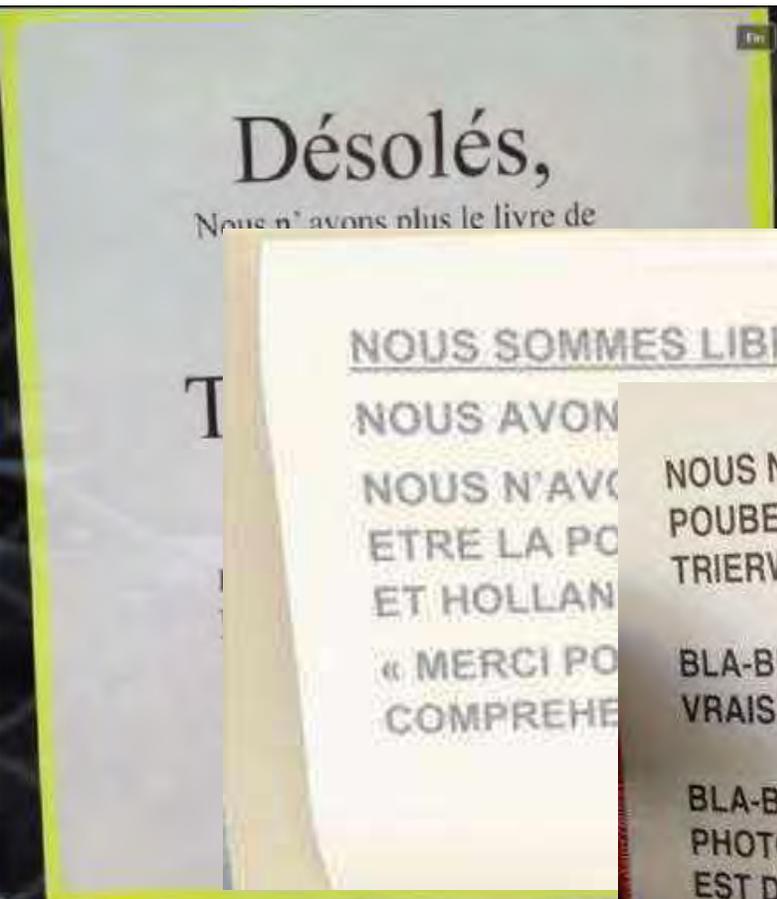


Damjan Petrovic





Un vrai lib



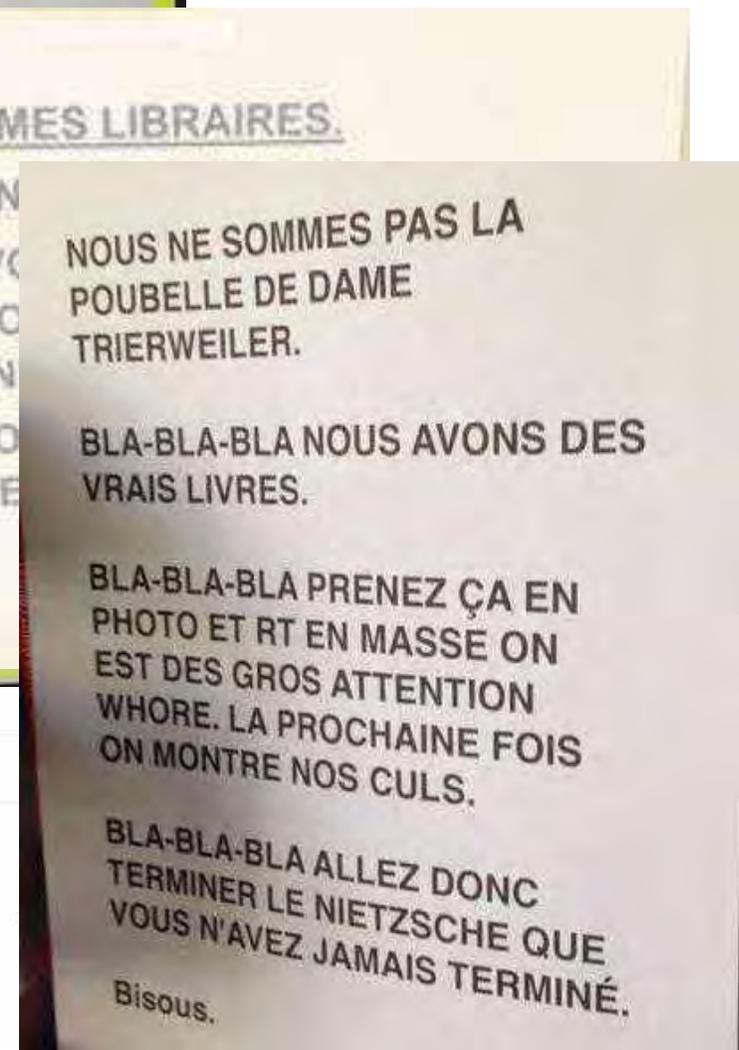
07:08 - 5 set 2014

26 Retweet 15 Mi piace

4 26 15



Sofian @Crohnfighter · 5 set 2014
In risposta a @Phocida
@Phocida: refuser de servir de la merde c'est un honneur

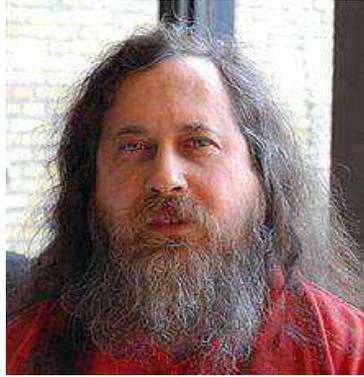




ALAN TURING



JOHN VON NEUMANN



RICHARD STALLMAN



BILL GATES



MARK ZUKEMBERG



STEVE JOBS



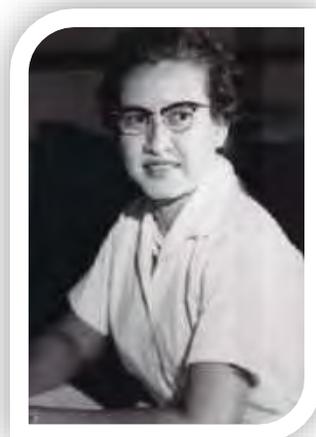
TIM BERNERS LEE



HEDI LAMARR



MARGARETH HAMILTON



KATHERINE JOHNSON



ADA LOVELACE



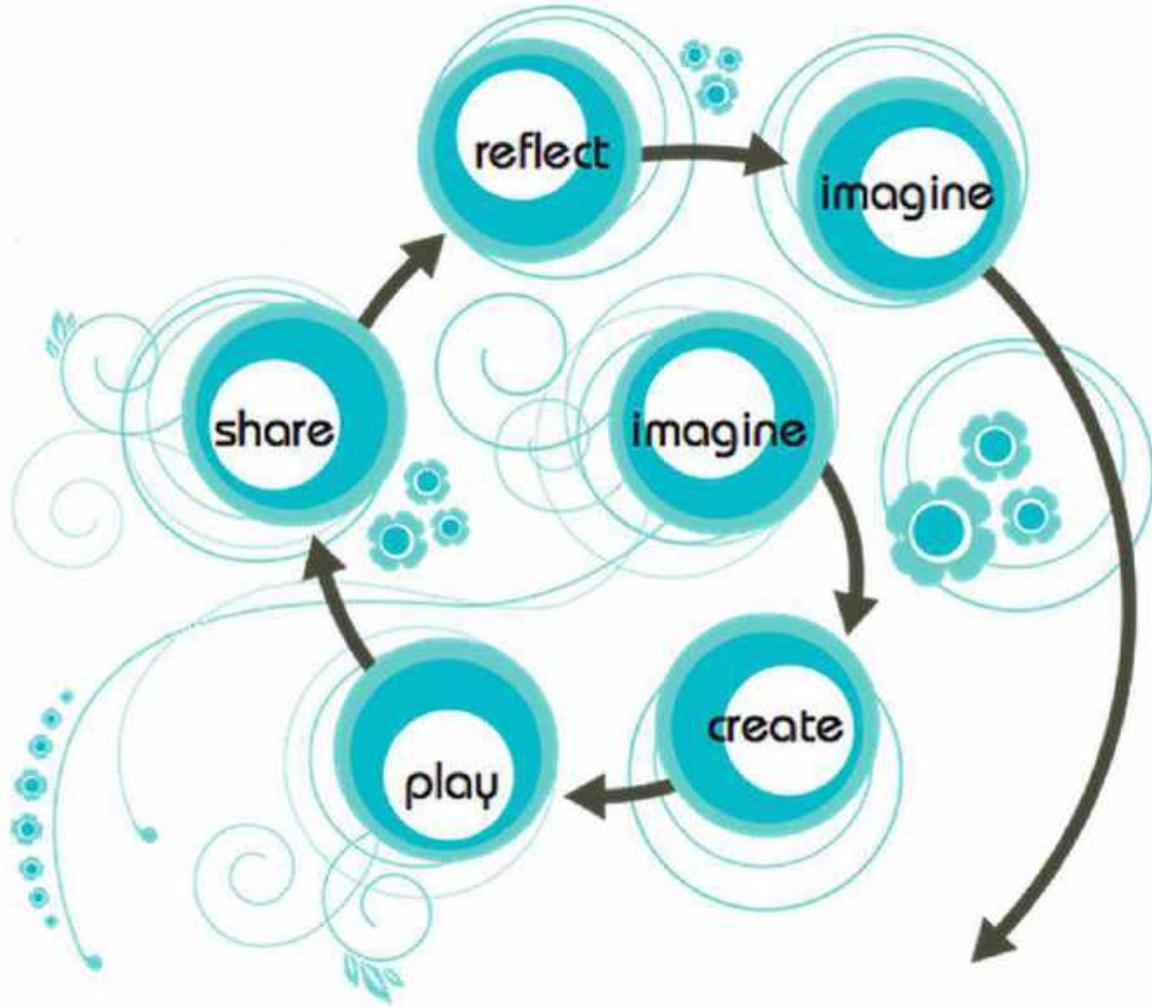
EVIDENZA – riconoscimento dei problemi e delle idee da sviluppare

ANALISI – dividere ogni problema in parti più piccole

SINTESI – Sequenza ordinata di passi fino alla definizione di problemi più complessi

ENUMERAZIONE – verifica scrupolosa di ogni passaggio per evitare errori

Renè Descartes – discorso sul metodo (1637)



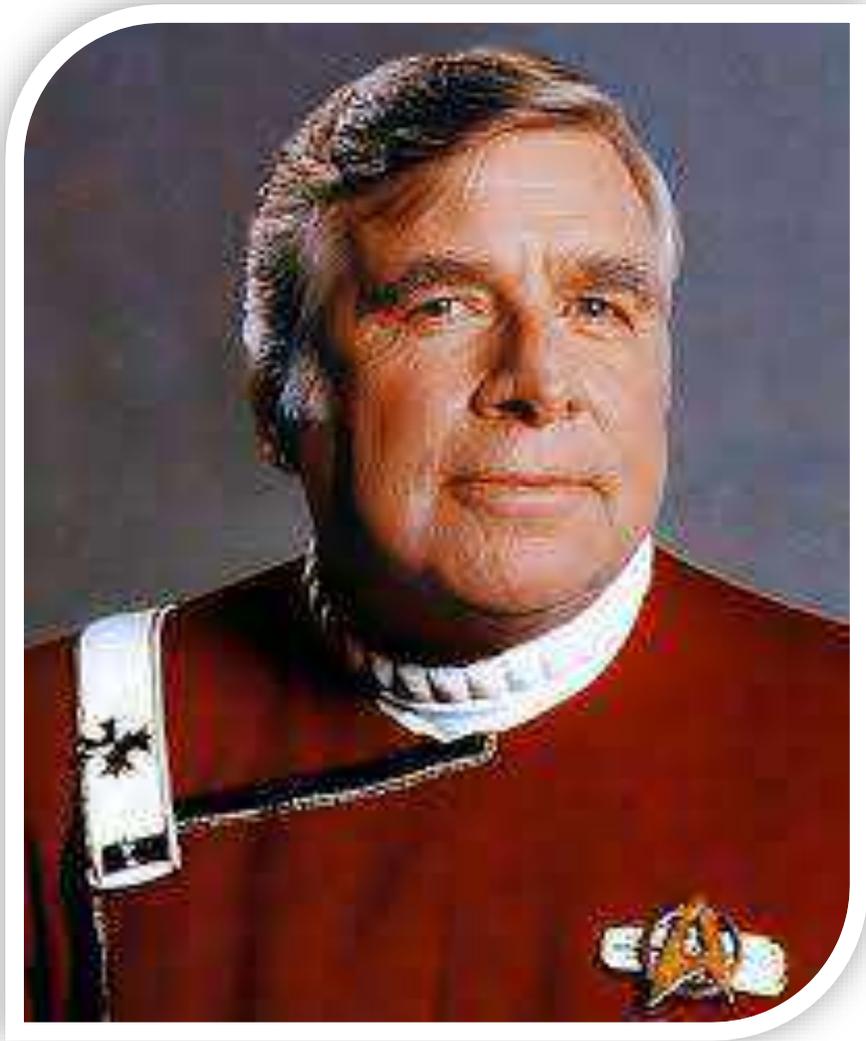
MITCH RESNICK

ESISTE UNA COSTANTE SVALUTAZIONE DELLE OPERAZIONI MANUALI RISPETTO AL LAVORO INTELLETTUALE.

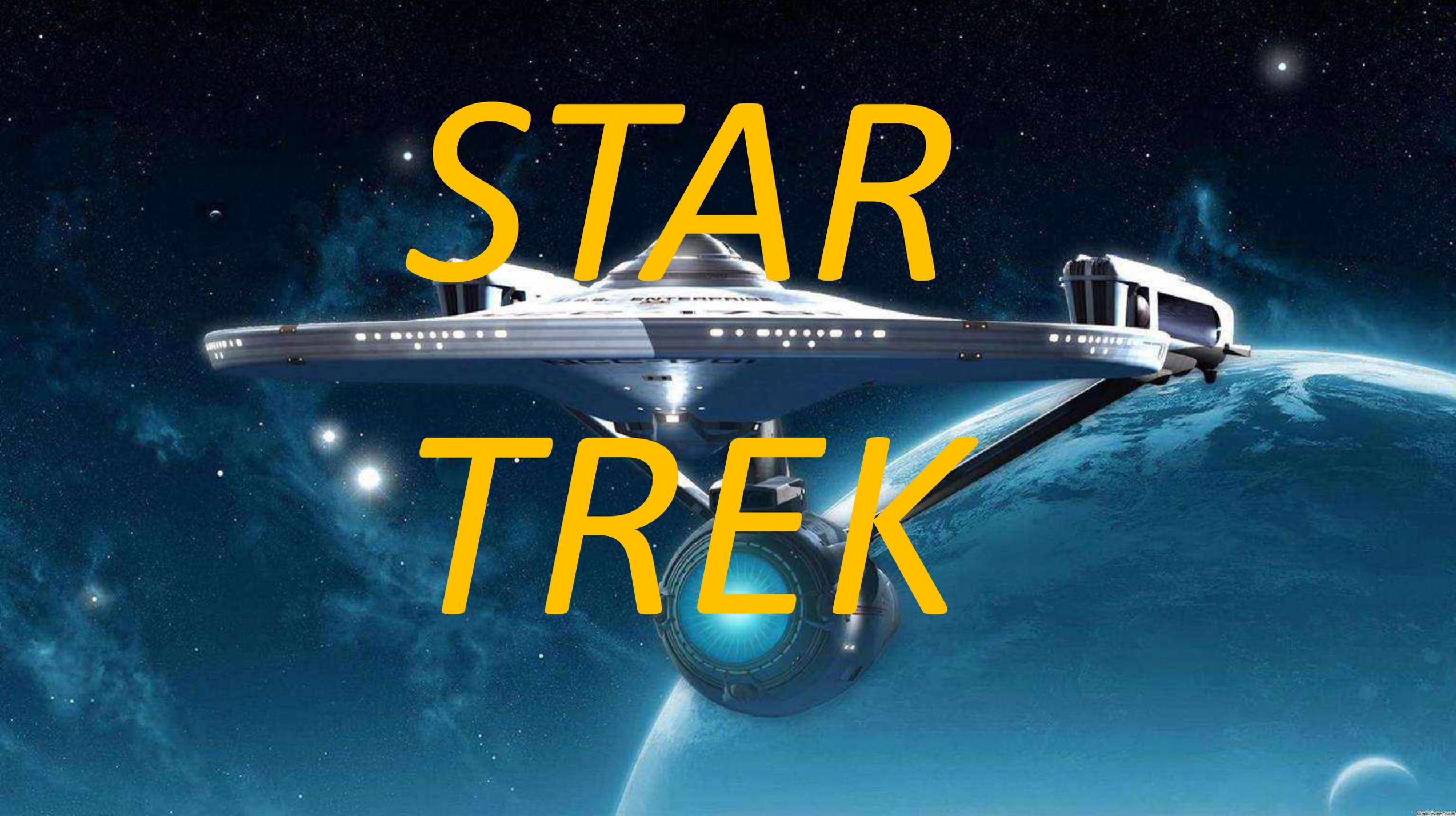
E SE IN UN MONDO DI SCATOLE CHIUSE SFOSSE NECESSARIO PROPRIO UNA CULTURA DEL FARE, UNA IDEA DI MAKING?

LAVORO BEN FATTO



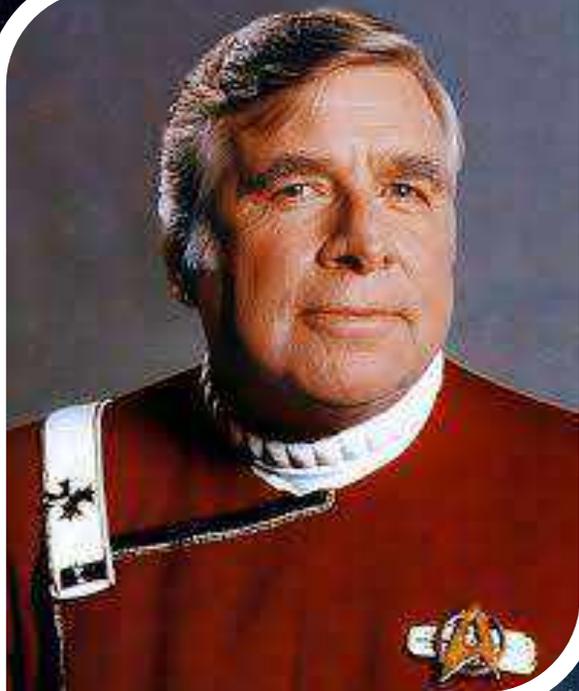


GENE RODDENBERRY

A Star Trek Enterprise ship is shown orbiting the Earth. The ship is a sleek, white, saucer-shaped vessel with a central nacelle and two side nacelles. It is positioned in the upper half of the frame, with the Earth's blue and white horizon visible below it. The background is a deep blue space filled with stars and nebulae. The word "STAR" is written in large, bold, yellow, italicized capital letters across the top of the ship.

STAR

TREK



Eugene Wesley Roddenberry
El Paso 1921 – Santa Monica 1991





Comunicatore

Il «Communicator», il dispositivo usato dal personale della Flotta Stellare che permette la trasmissione vocale. Utilizzato dal capitano Kirk all'interno e all'esterno dell'Enterprise, fece la sua prima comparsa nell'episodio «The Cage» («Lo zoo di Talos»).
8 settembre 1966.

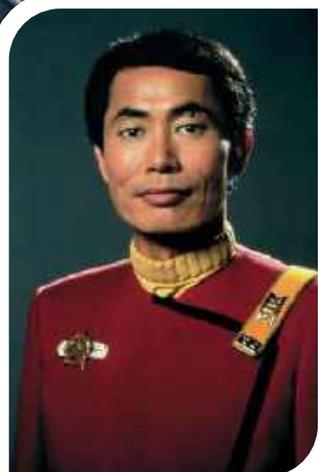
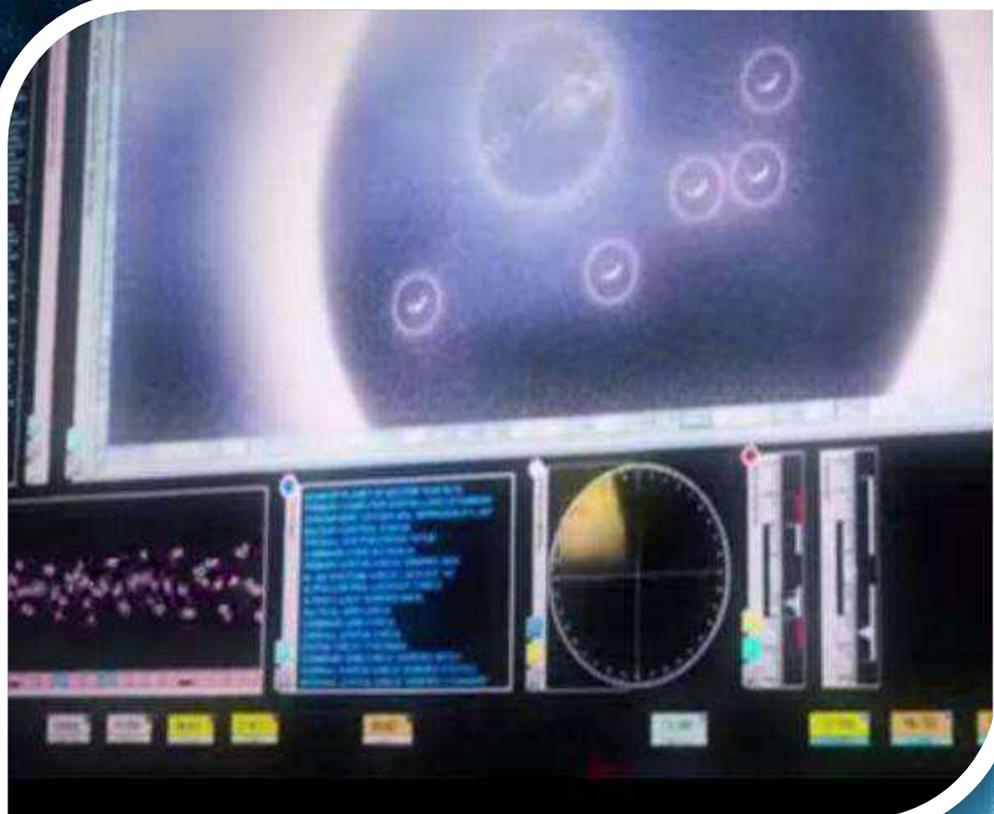
Porte Automatiche

In tempi in cui si girava in economia, il suono dell'apertura e della chiusura delle porte automatiche era doppiato da un semplice soffio di Roddenberry nel microfono.



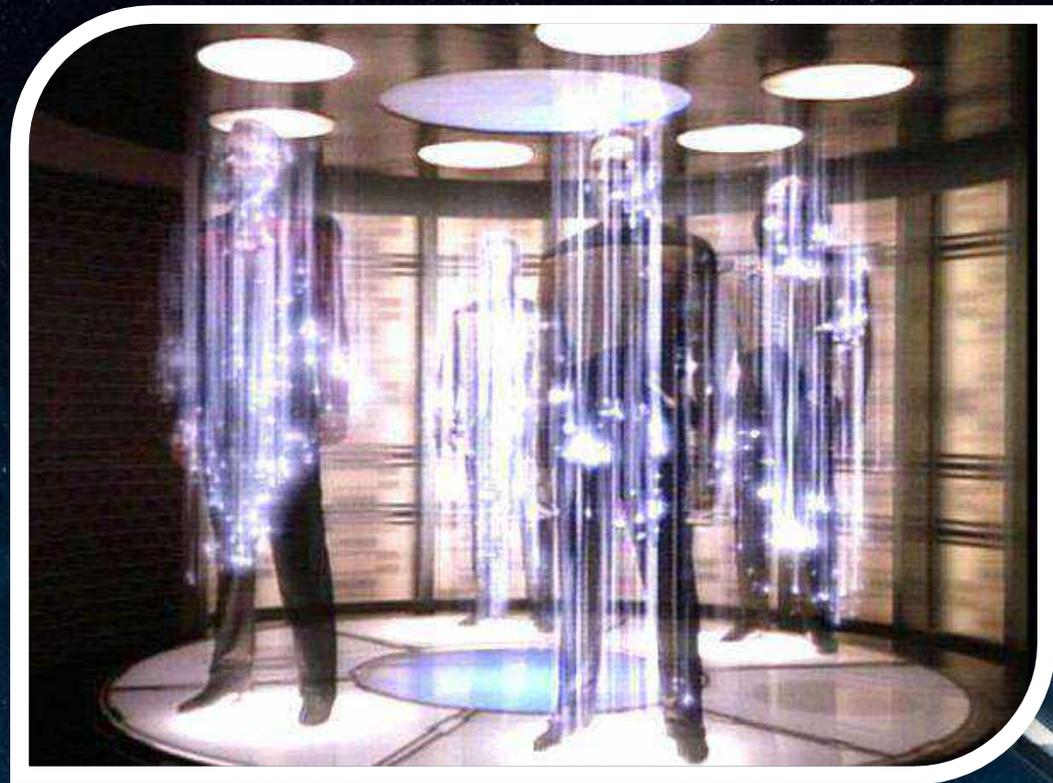
Personal computer

Nel 1966, anno del debutto di Star Trek, i computer erano grandi, costosi e lenti, ma questo non ha impedito a Gene Roddenberry di immaginarli in modo diverso. Cinque anni dopo l'ultima puntata di Star Trek il primo personal computer, l'Altair 8800, ha visto la luce grazie a Ed Roberts.



Lo schermo visore

«*Sulu, sullo schermo!*» è una delle frasi più note pronunciate dal capitano Kirk. Prima di Star Trek le navicelle spaziali erano dotate "solamente" di oblò. Un passaggio epocale, come quello dallo schermo a tubo catodico ai televisori al plasma.

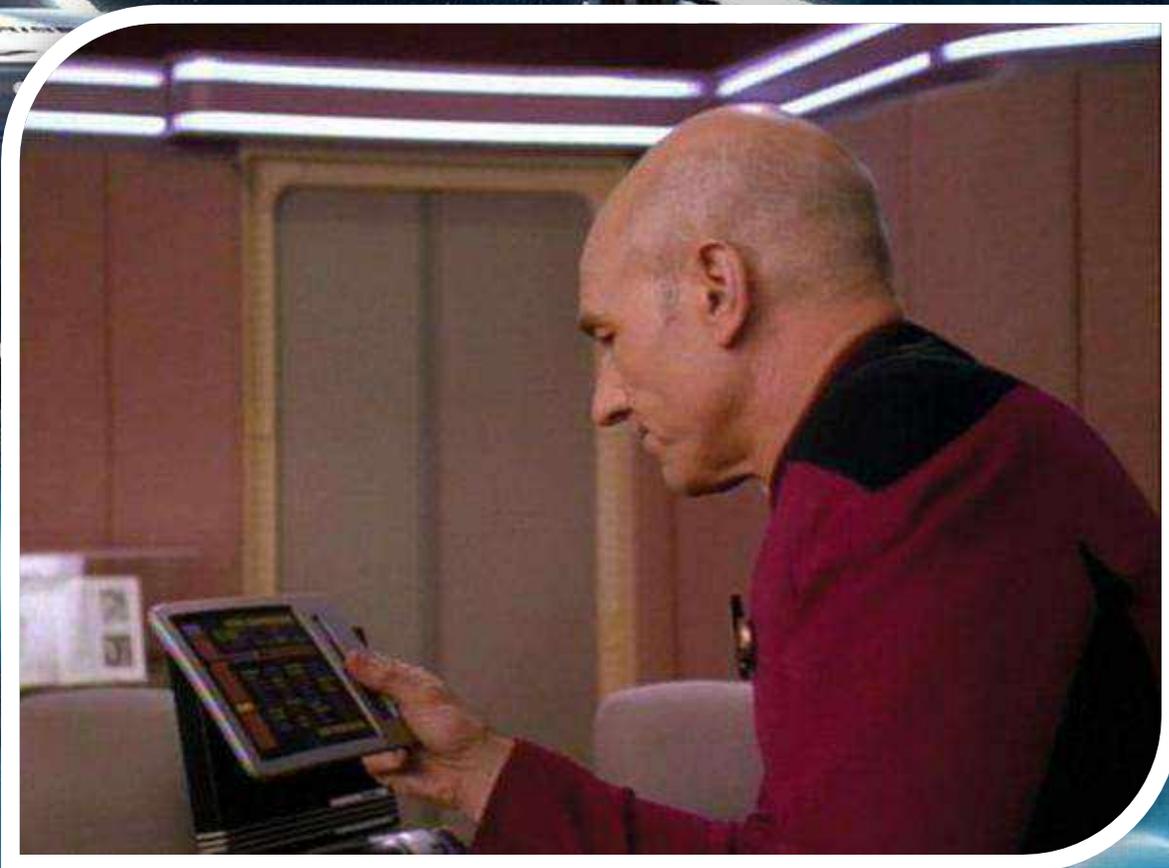


Tablet

«Giornale di bordo del Capitano, data astrale 1312.4...». Inizia così il primo episodio (intitolato "Oltre la galassia") di Star Trek. La voce del capitano James Kirk, che aggiorna il diario di bordo dell'Enterprise, farà da prologo a tutti e 29 gli episodi della prima stagione.

Localizzatore di equipaggio (GPS)

Il localizzatore veniva utilizzato sulle navi stellari per indicare la posizione dei membri dell'equipaggio. In Star Trek si riusciva a localizzare chiunque sulla superficie di un pianeta. Il Dipartimento della Difesa iniziò a sviluppare il sistema GPS circa 4 anni dopo la fine della serie, all'apice della guerra fredda.



Traduttore universale

Grazie al traduttore universale il dottor Spock e il capitano Kirk hanno potuto comprendere le lingue delle creature aliene come i Sulibani, i Tholiani, i Klingon e gli Andoriani e far fronte a diverse situazioni di pericolo. Oggi Google Translate e il traduttore «Skype» di Microsoft permettono agli utenti di superare le barriere linguistiche.



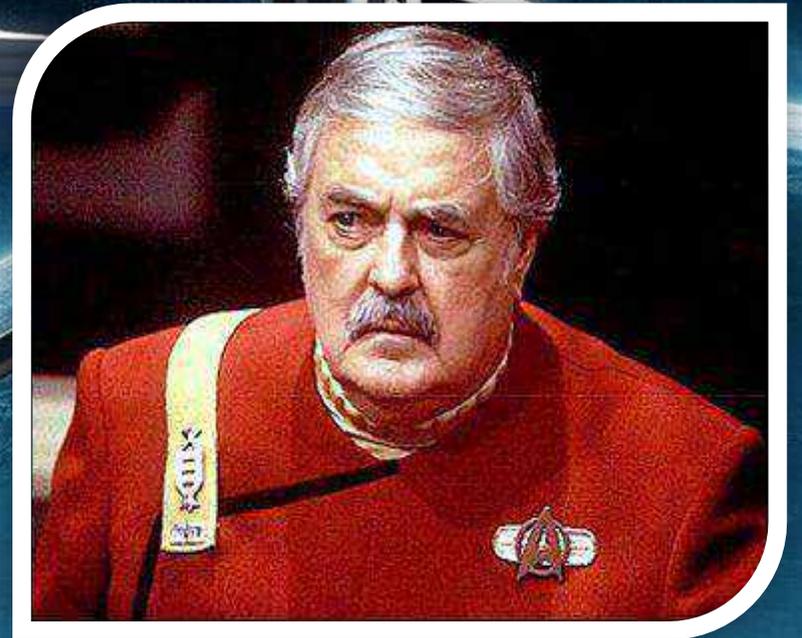
Space travel

Uno dei più grandi lasciti di «Star Trek» è il viaggio spaziale, tentativo decisamente rischioso da compiere negli anni '60. Oggi si programmano viaggi su altri pianeti con discreta facilità.

STEM



Dr. Spok approach



Dr. Scott approach

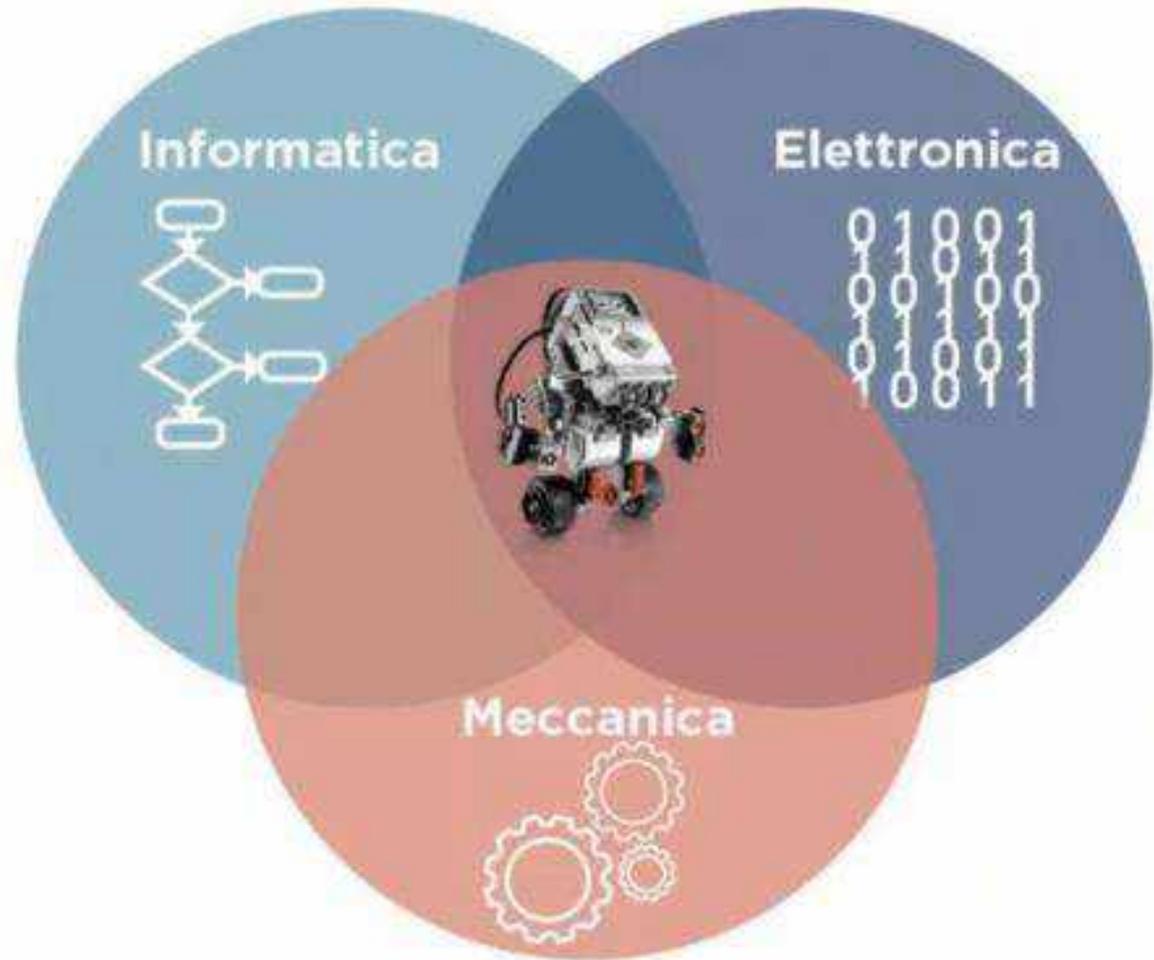
ROBOTICA

4 aprile 1978



ROBOTICA

INFORMATICA



MBLOCK



The screenshots illustrate the mBlock environment's capabilities:

- Top-left:** Shows the main workspace with a panda sprite and various sensor blocks like 'show image' and 'turn off'.
- Middle-left:** Shows a 'mBlock robot code' sprite with a circular sensor array.
- Bottom-right:** Shows a 'trigonometric function' data chart window with a sine wave graph and a corresponding code block for plotting data.

MBLOCK

Incontro 2 e Incontro 3

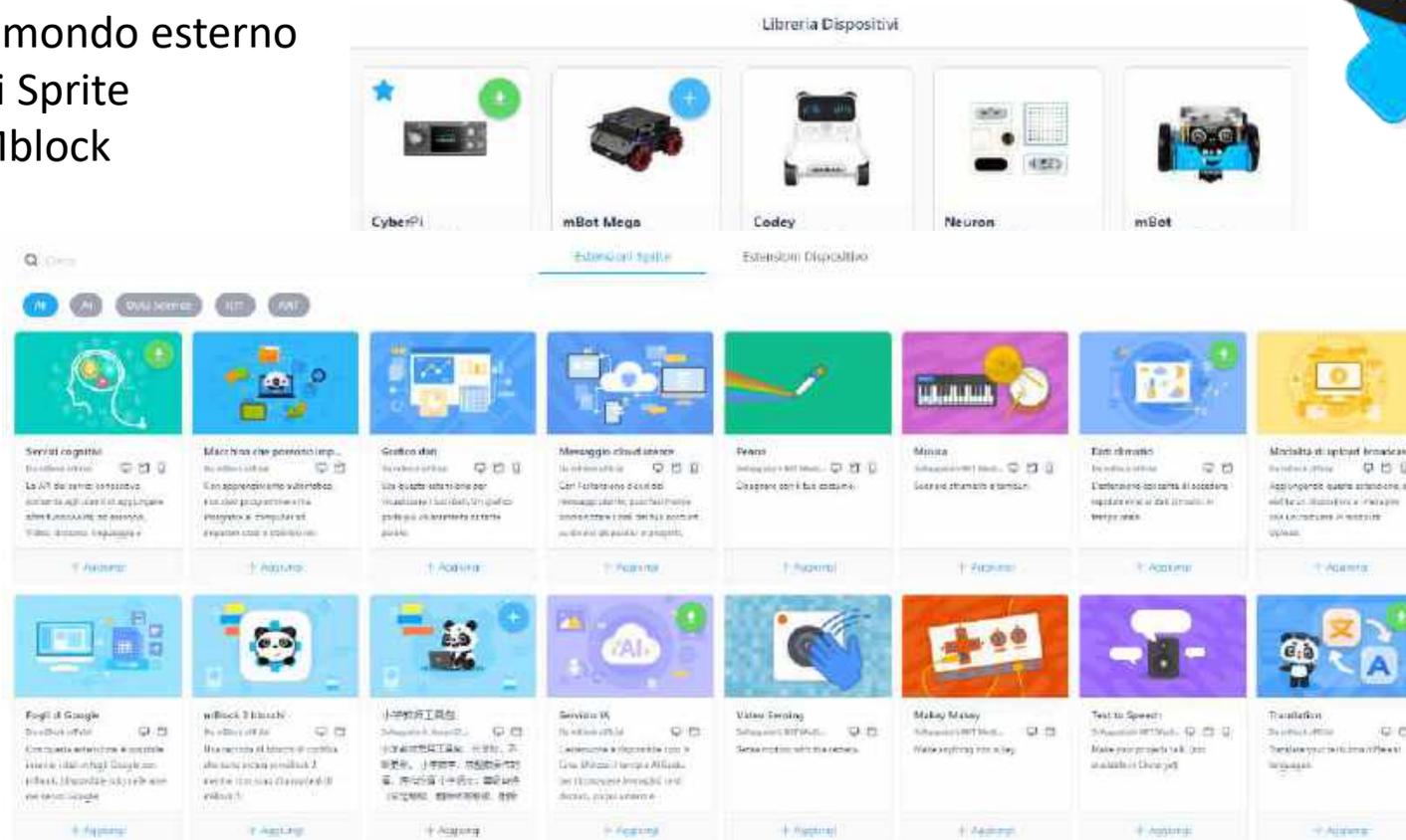
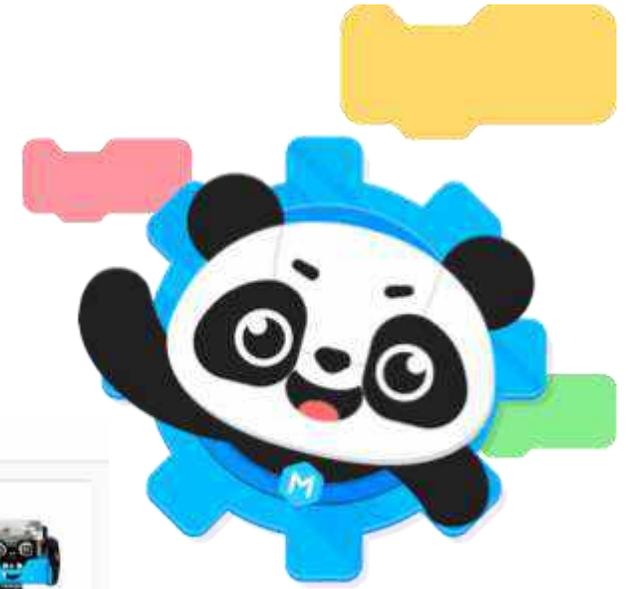
Muoviamoci sullo stage

Interazione tra Sprite

Interazione col mondo esterno

Estensioni degli Sprite

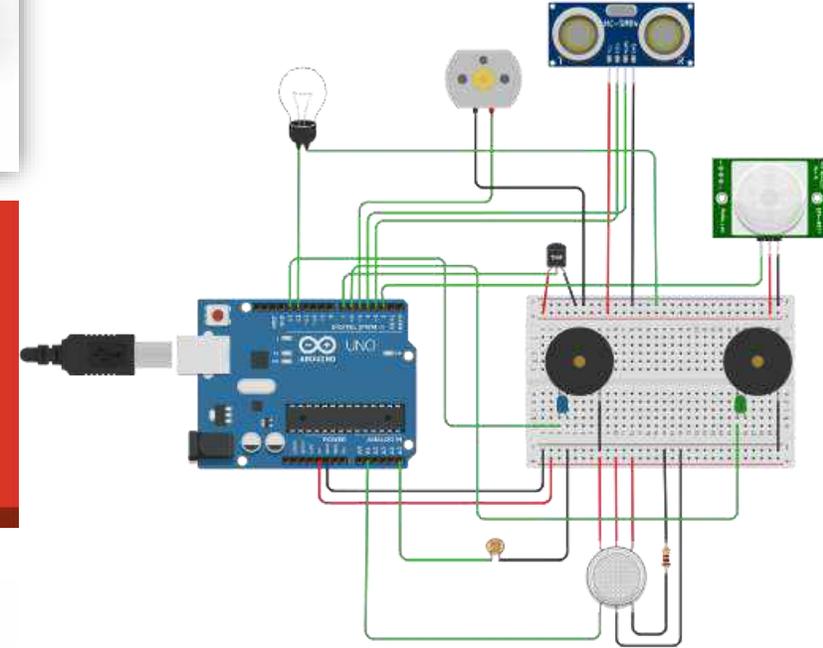
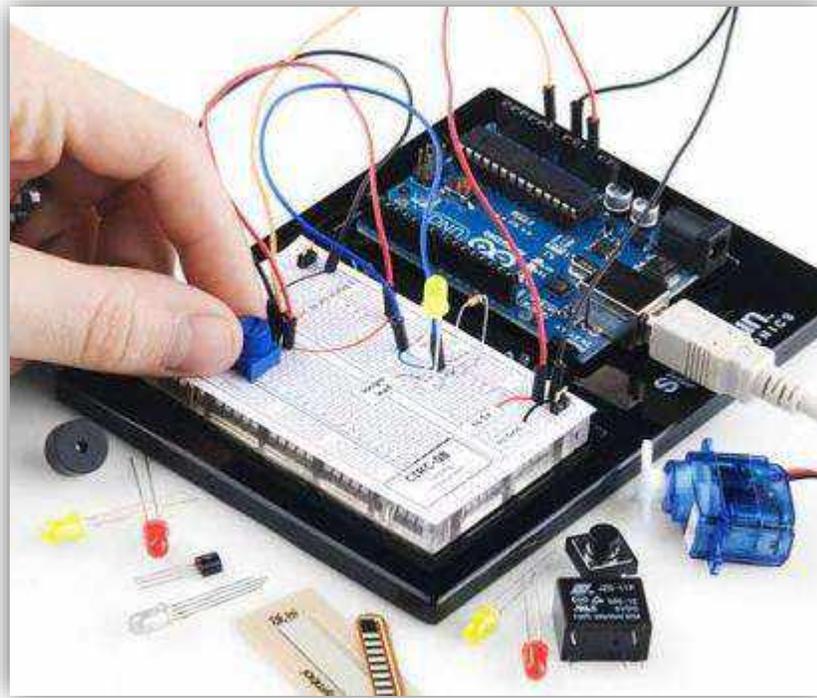
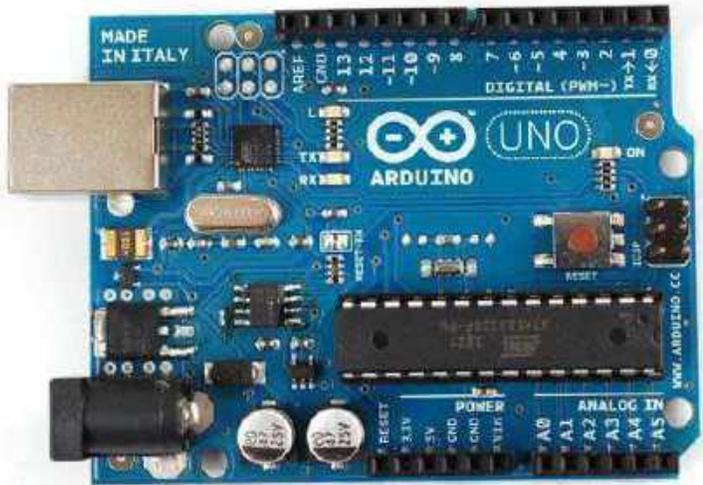
Robotica con Mblock



ARDUINO

Incontro 4

- Introduzione ad Arduino
- Concetto di segnale
- Circuiti elettrici
- Fritzing
- Tinkercad
- Presentare un progetto



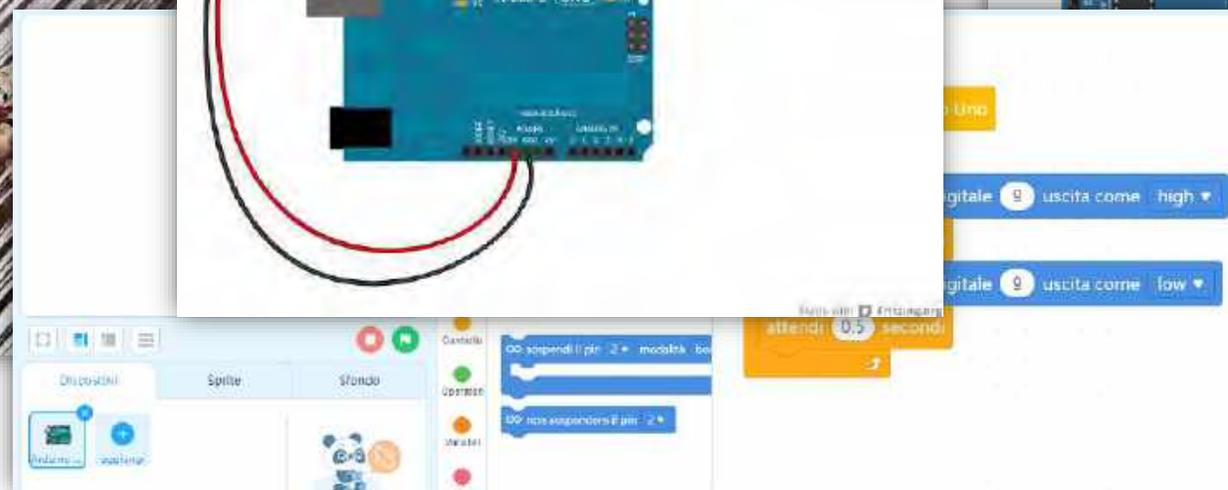
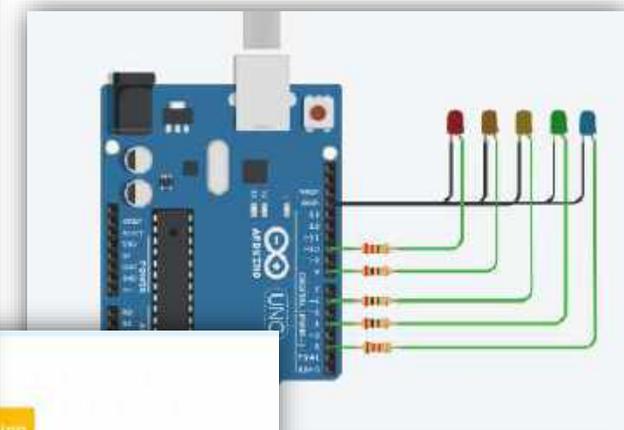
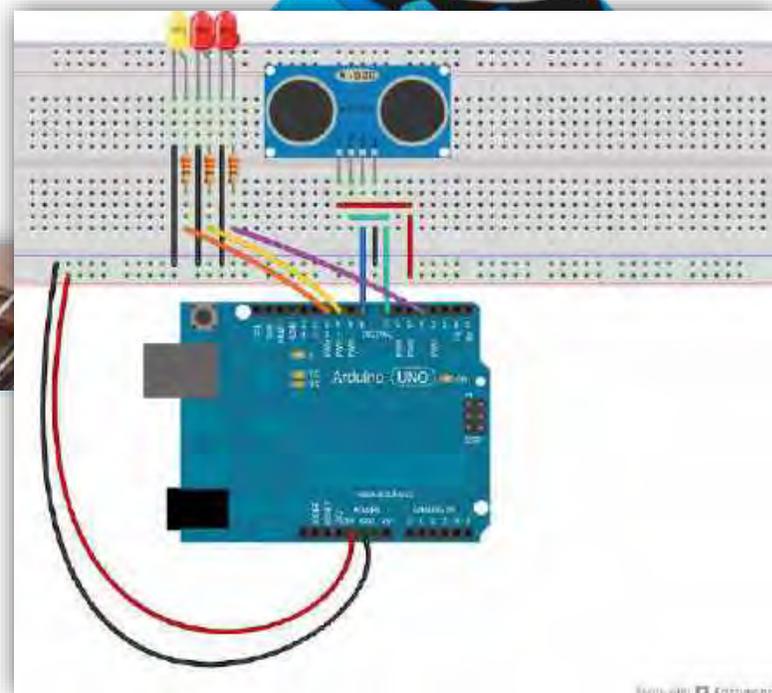
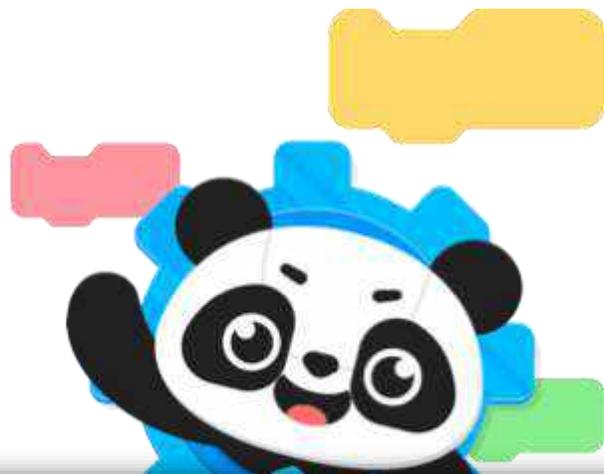
AUTODESK®
TINKERCAD®

*Innovation
for education*

MBLOCK E ARDUINO

Incontro 5

- Leggi di Ohm
- LED e Resistenze
- Semaforo
- Sensore ad ultrasuoni
- Sensore di parcheggio



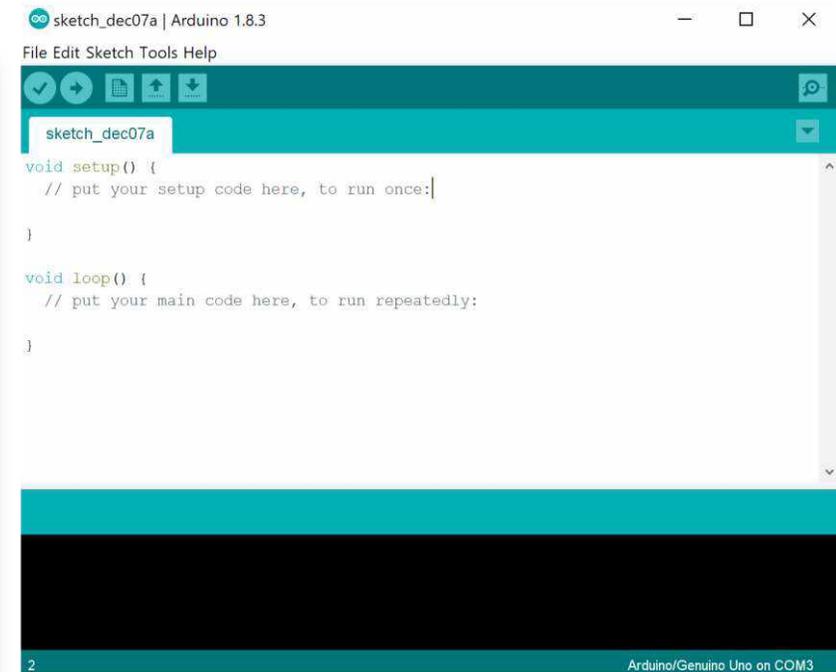
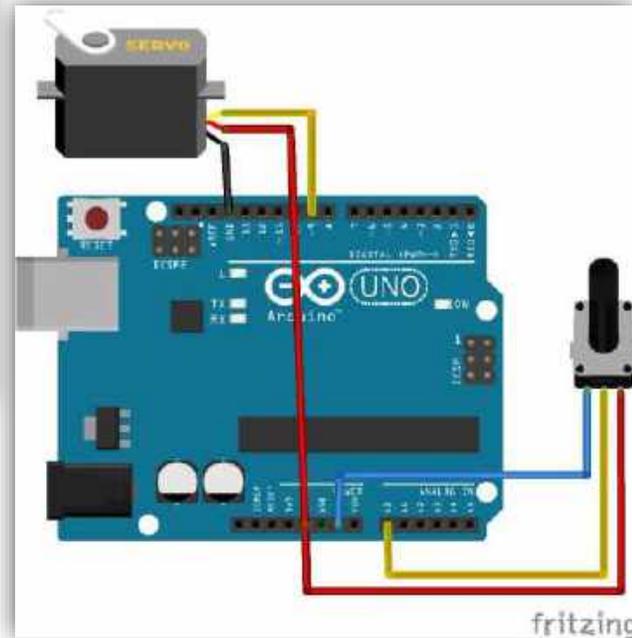
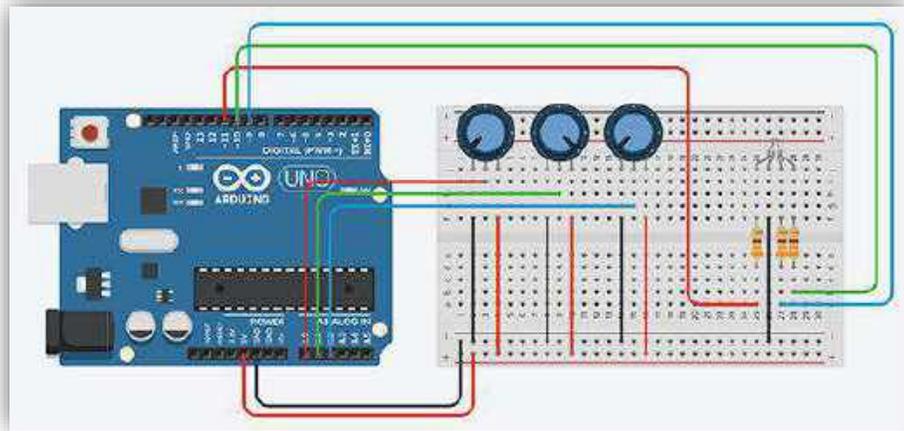
MBLOCK E ARDUINO

Incontro 6

Potenzimetri

Servomotori

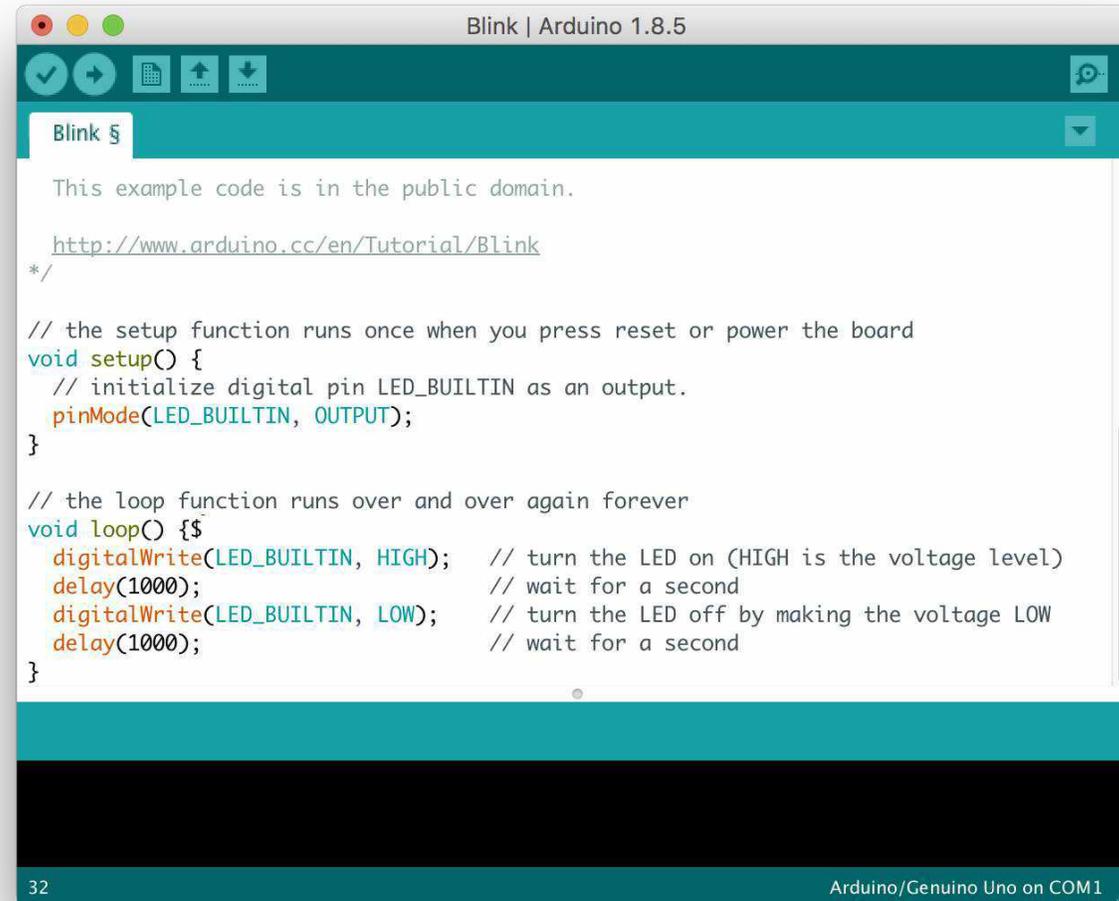
Introduzione all'IDE di Arduino



ARDUINO

Incontro 7

- IDE Arduino
- LED
- Servomotori
- Potenzimetri
- Librerie
- Musica e array (??)



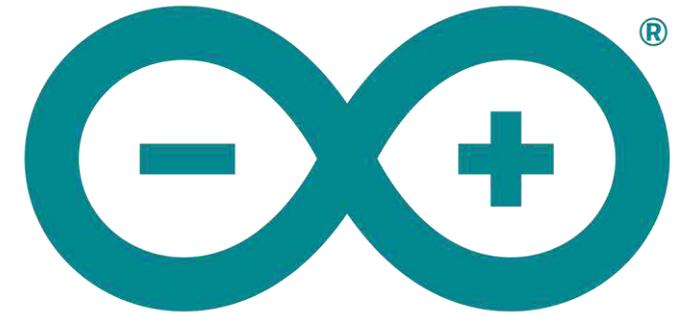
The screenshot shows the Arduino IDE interface with the 'Blink' example code loaded. The code is as follows:

```
Blink | Arduino 1.8.5  
Blink §  
This example code is in the public domain.  
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  
}
```

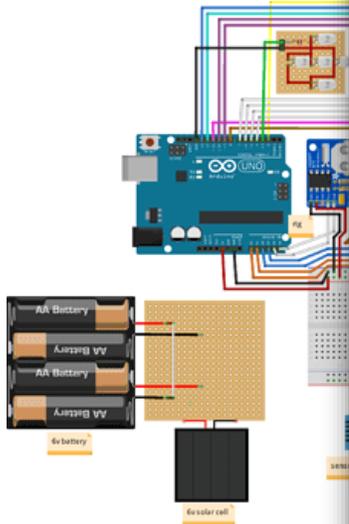
At the bottom of the IDE window, it shows '32' on the left and 'Arduino/Genuino Uno on COM1' on the right.



PROGETTI ARDUINO



ARDUINO

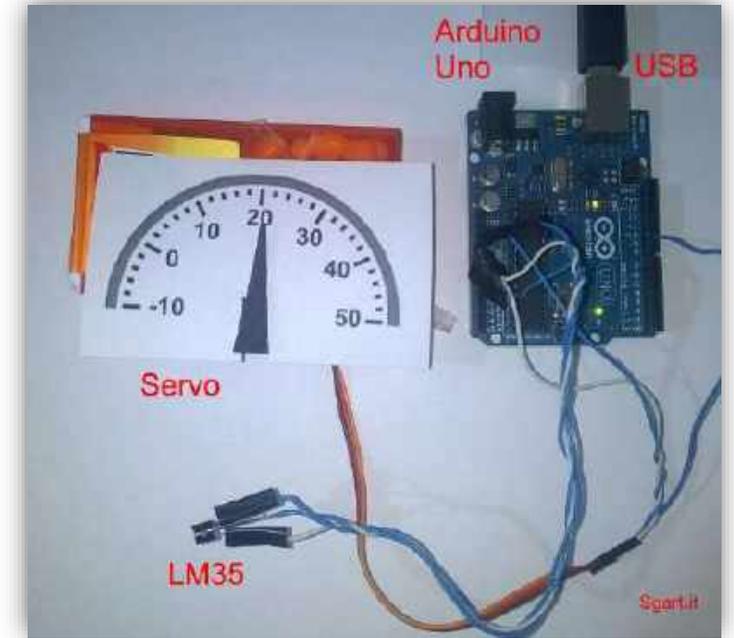


PROJECT WORKS

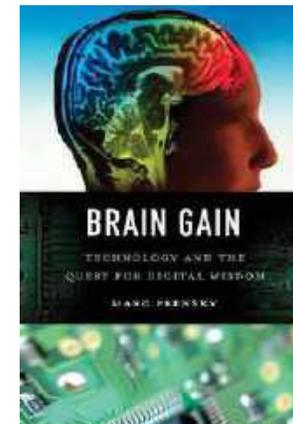
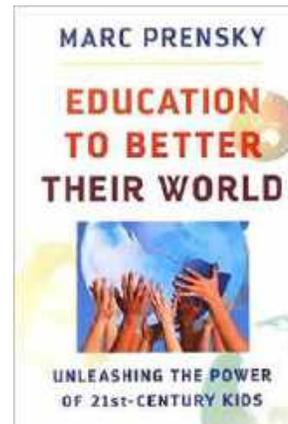
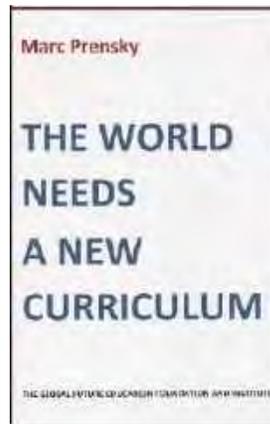
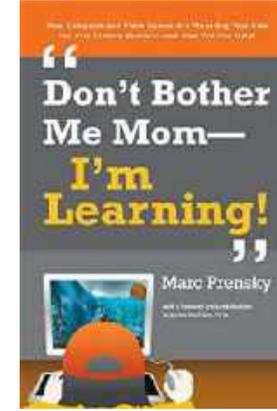
- Descrizione dell'idea di progetto;
- Schema dei collegamenti;
- Codici o pseudocodici;
- Video e foto della realizzazione.



- Stazione Meteo;
- Lampada da Comodino;
- Indicatore di temperature;
- Semaforo per ciechi;
- Presepe;
- Crepuscolare;
- Tutto quello che vi viene in mente.



Mark Prensky



Alessandro Baricco



<https://youtu.be/T3Kzsii1z-8> – Sulla narrazione

https://youtu.be/BieC_ib_l1o - Sulla felicità

(Non in rete) - Sulla verità



<https://youtu.be/SK-0VbHxBbo>
The Game





Stefano Laffi

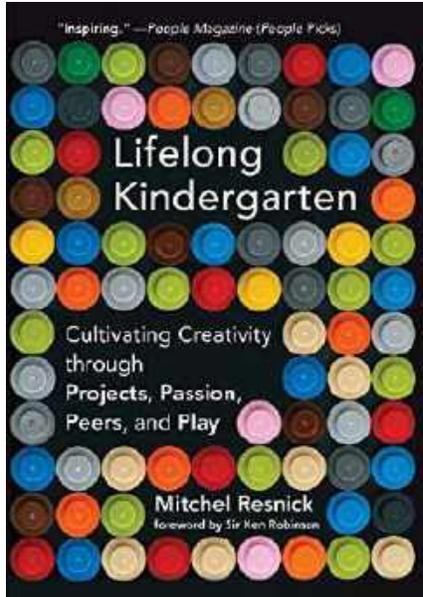


<https://youtu.be/Ctvuh3e70Kw> - La congiura contro i giovani

<https://youtu.be/UZIFAoVfWpQ> - What are young people stand for?



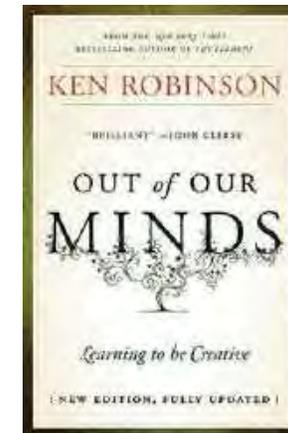
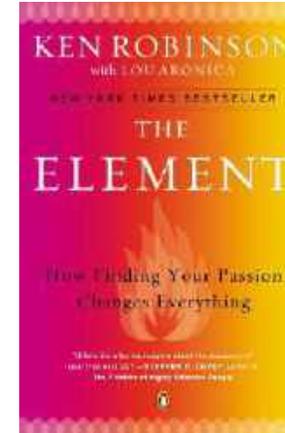
Mitch Resnick



<https://youtu.be/IfvgVpQI56I> - Kindergarten For Our Whole Lives



Ken Robinson



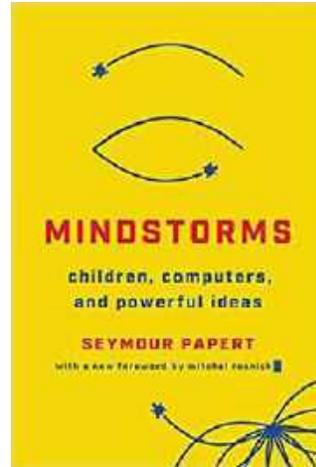
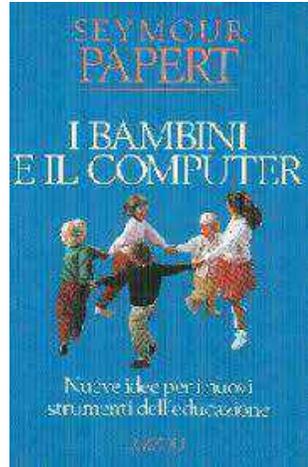
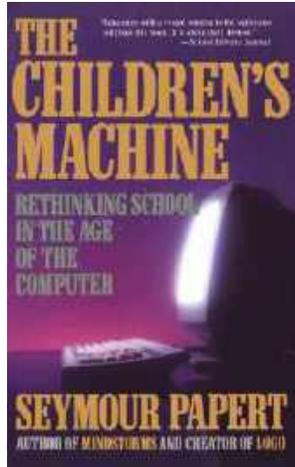
<https://youtu.be/K3uXSYQWAwA> - la scuola uccide la creatività

<https://youtu.be/SVeNeN4MoNU> - Cambiare i paradigmi dell'educazione

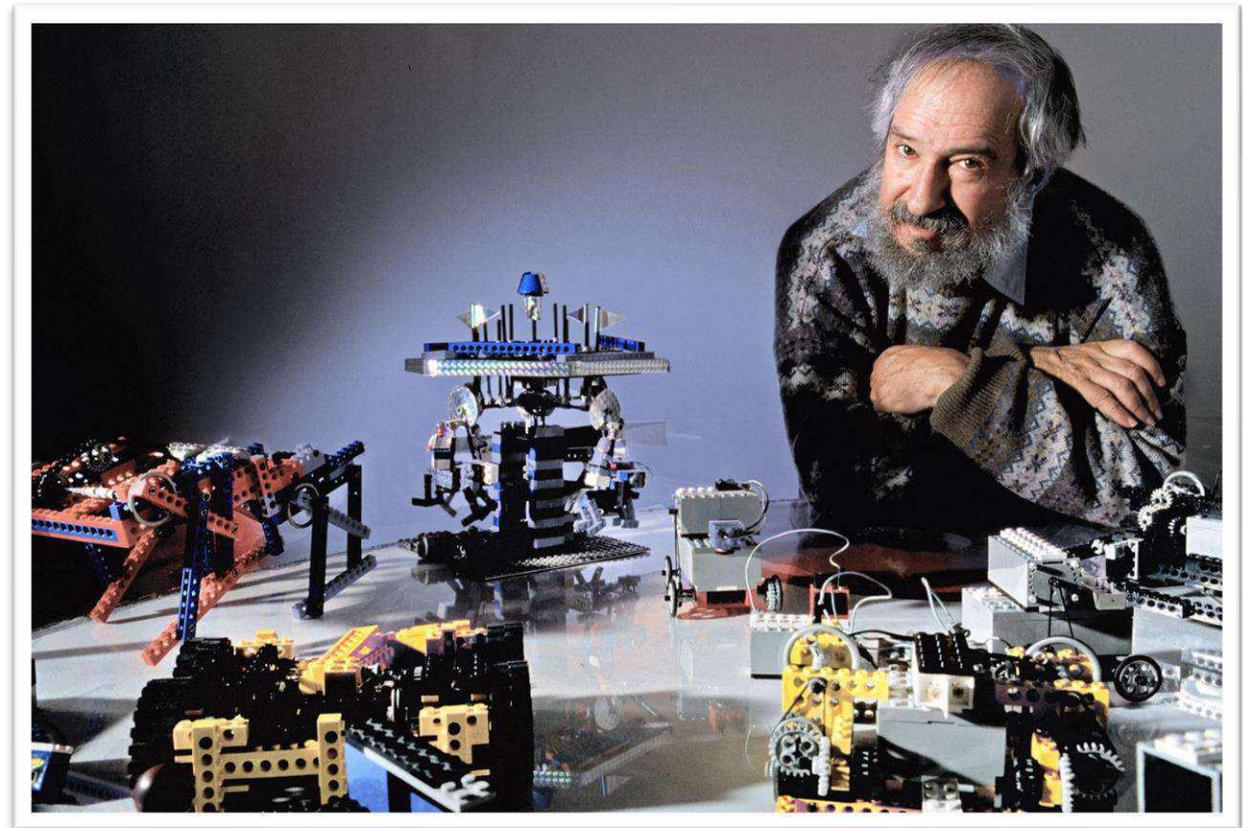
<https://youtu.be/17fbxRQgMIU> - Trovare il tuo elemento

<https://youtu.be/ALlcNmqC-KI> - La scuola sotto processo

Seymour Papert



<https://youtu.be/ZG9cYhekB8A> - Logo: Teaching





ARDUINO



ROBOTICA - 4 APRILE 1978



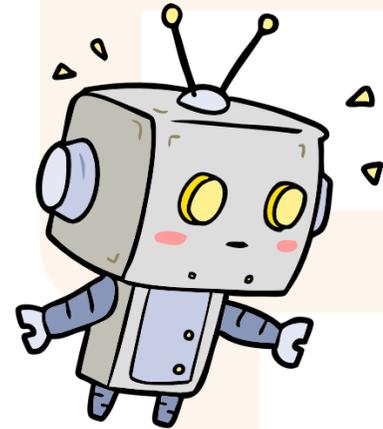
CONCETTI:

MODELLO
ANTROPOMORFO

SENSORI ED
ATTUATORI

MICROCONTROLLORE

FORZA DELLE IDEE





CONCETTI:

MODELLO
ANTROPOMORFO

SENSORI ED
ATTUATORI

MICROCONTROLLORE

FORZA DELLE IDEE

ROBOTICA - 4 APRILE 1978





CONCETTI:

MICROCONTROLLORE

ALIMENTAZIONE

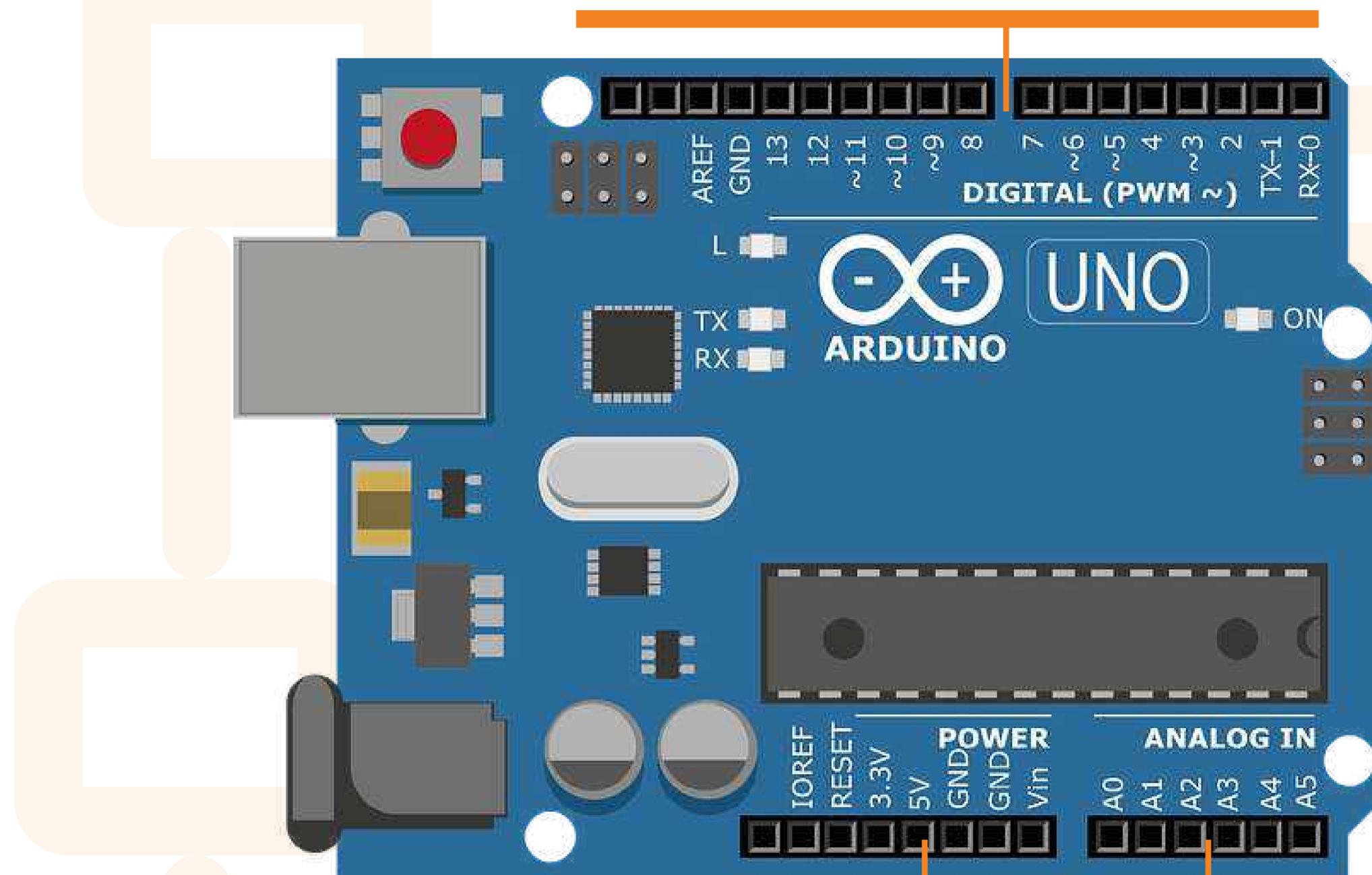
PIN DI
ALIMENTAZIONE

SEGNALI DIGITALI

SEGNALI ANALOGICI

MICROCONTROLLORE ARDUINO UNO

PIN DIGITALI (IN/OUT)



PIN ALIMENTAZIONE

PIN ANALOGICI (IN/OUT)



CONCETTI:

SCIENCE JOURNAL

SIMULATORE

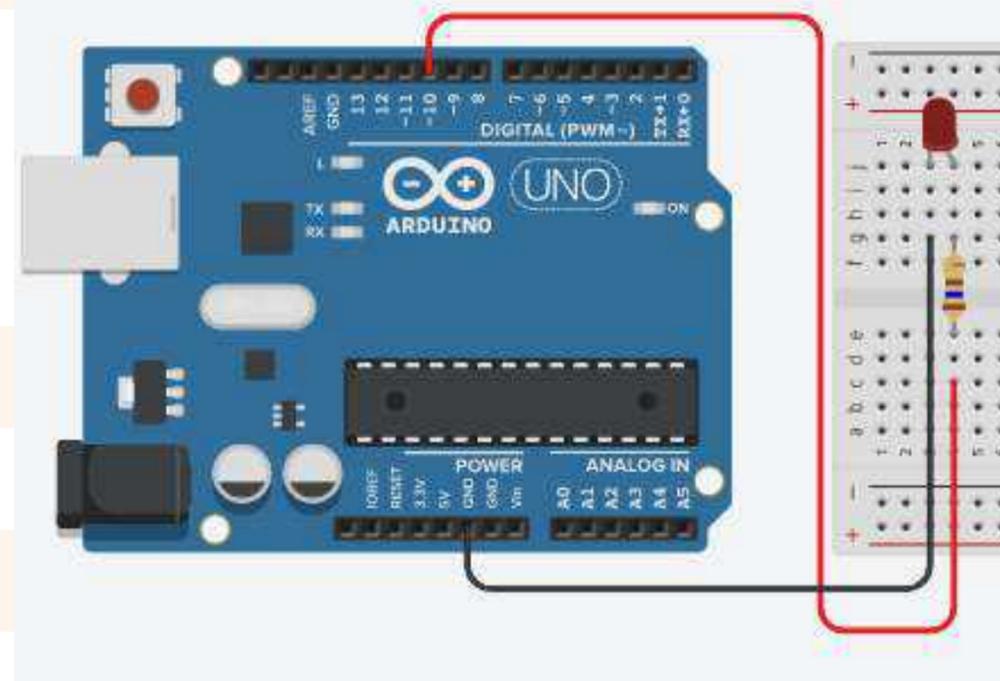
CIRCUITI

PROGRAMMAZIONE
A BLOCCHI

PROGRAMMAZIONE
TESTUALE



AUTODESK
Tinkercad



Blocchi + testo

1 (Arduino Uno R3)

- Uscita
- Ingresso
- Annotazione
- Controlla
- Matematica
- Variabili

al'avvio

sempre

```
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(10, OUTPUT);
6 }
7
8 void loop()
9 {
10  digitalWrite(10, HIGH);
11  delay(500); // Wait for 500 millisecond(s)
12  digitalWrite(10, LOW);
13  delay(500); // Wait for 500 millisecond(s)
14 }
```



CONCETTI:

FILOSOFIA MBLOCK

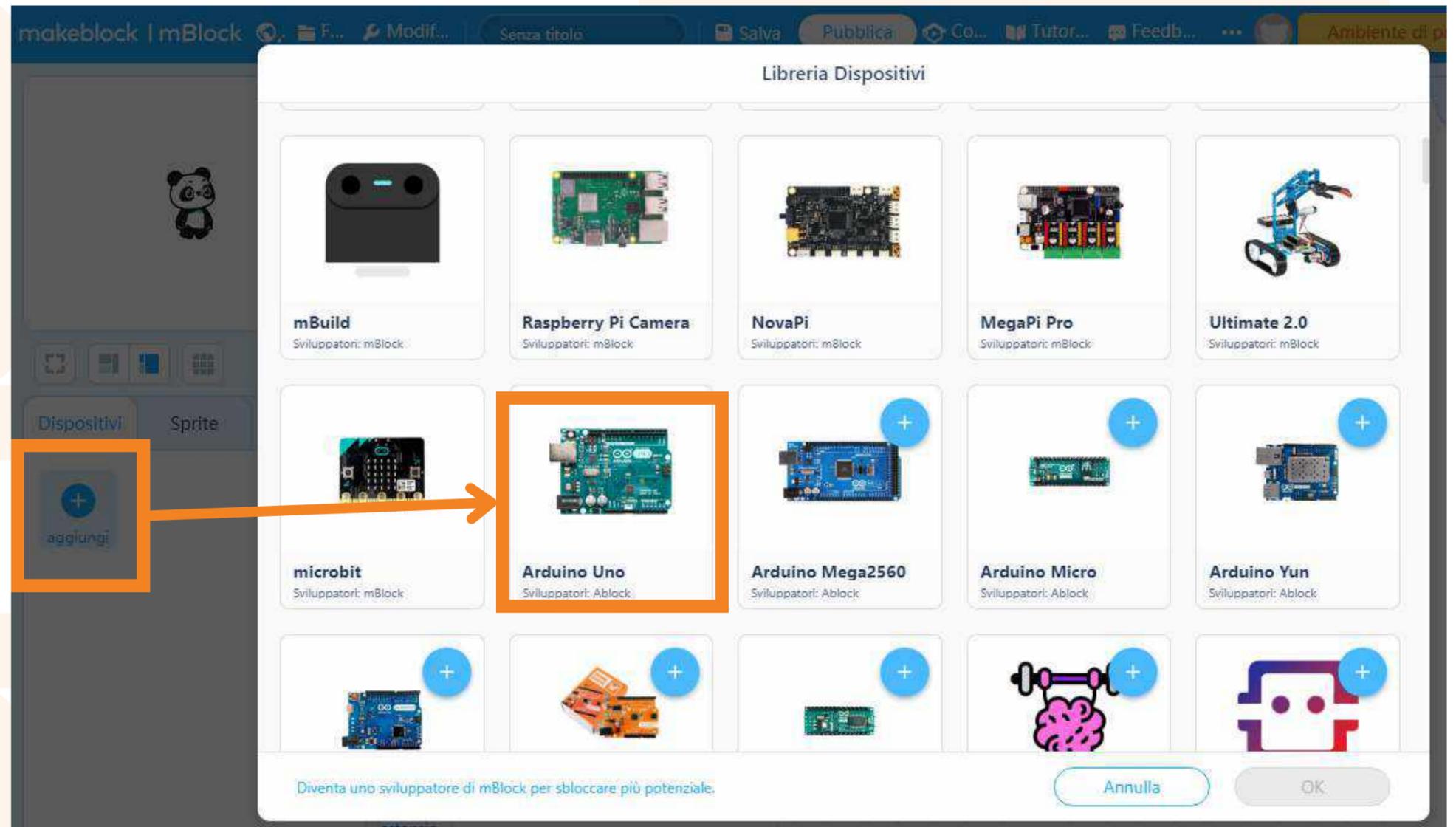
DISPOSITIVI

CARICAMENTI

ESTENSIONI



Mblock





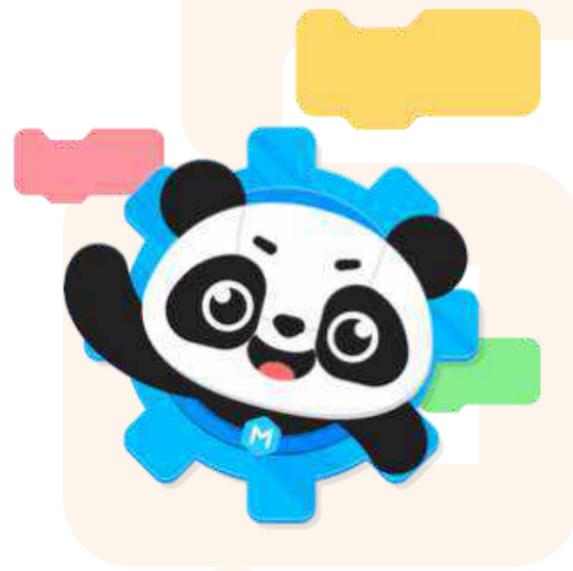
CONCETTI:

FILOSOFIA MBLOCK

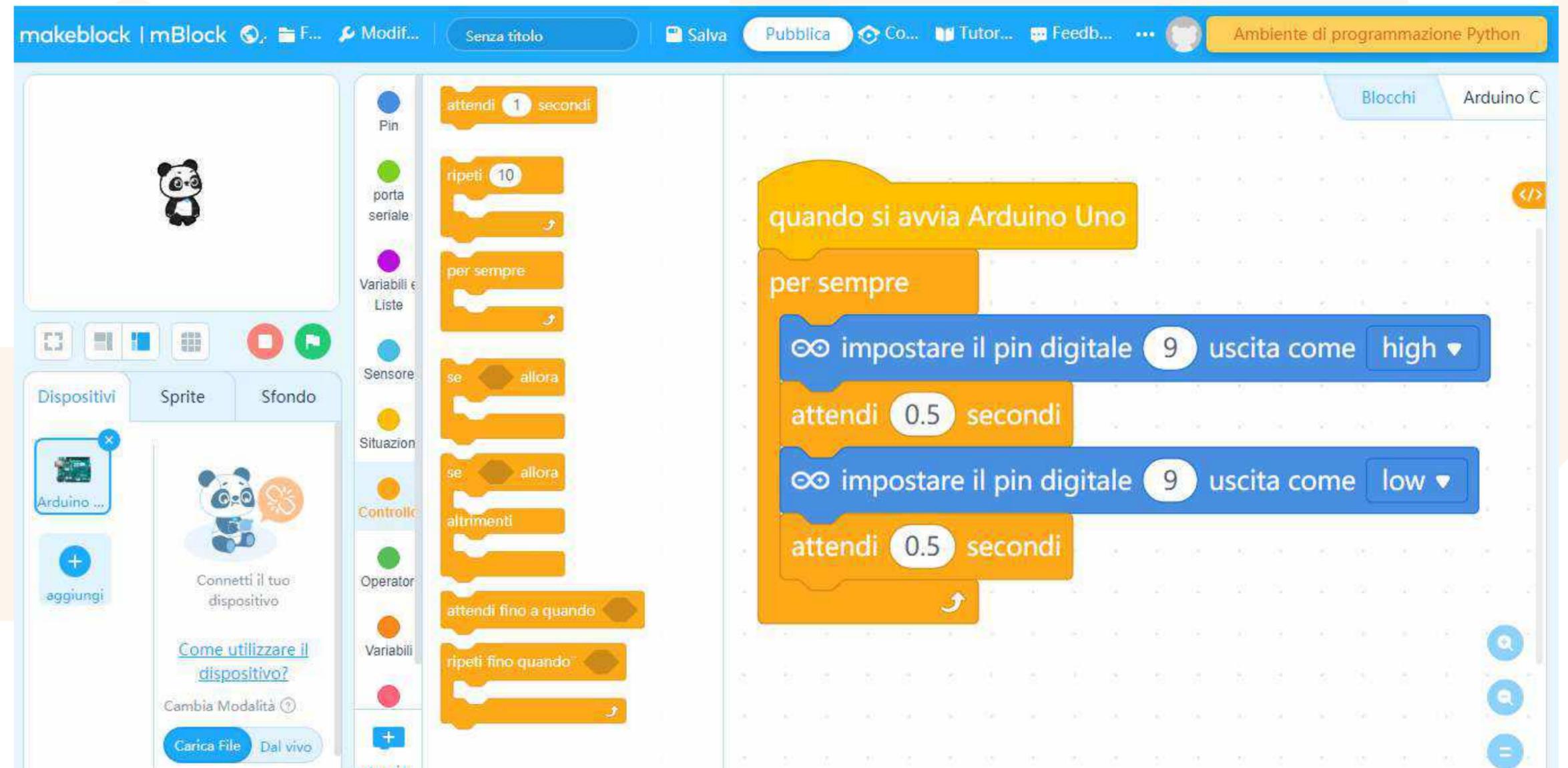
DISPOSITIVI

CARICAMENTI

ESTENSIONI



Mblock





CONCETTI:

IDE ARDUINO

SINTASSI

FUNZIONE SETUP

FUNZIONE LOOP

VERIFICA

STRUMENTI

```
Blink | Arduino 1.8.5  
✓ → [Icons] 🔍  
Blink §  
This example code is in the public domain.  
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  
}  
  
32 Arduino/Genuino Uno on COM1
```



CONCETTI:

IDE ARDUINO

SINTASSI

FUNZIONE SETUP

FUNZIONE LOOP

VERIFICA

STRUMENTI



```
sketch_jun13a | Arduino 1.8.19  
File Modifica Sketch Strumenti Aiuto  
sketch_jun13a  
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```



CONCETTI:

CIRCUITI ELETTRICI

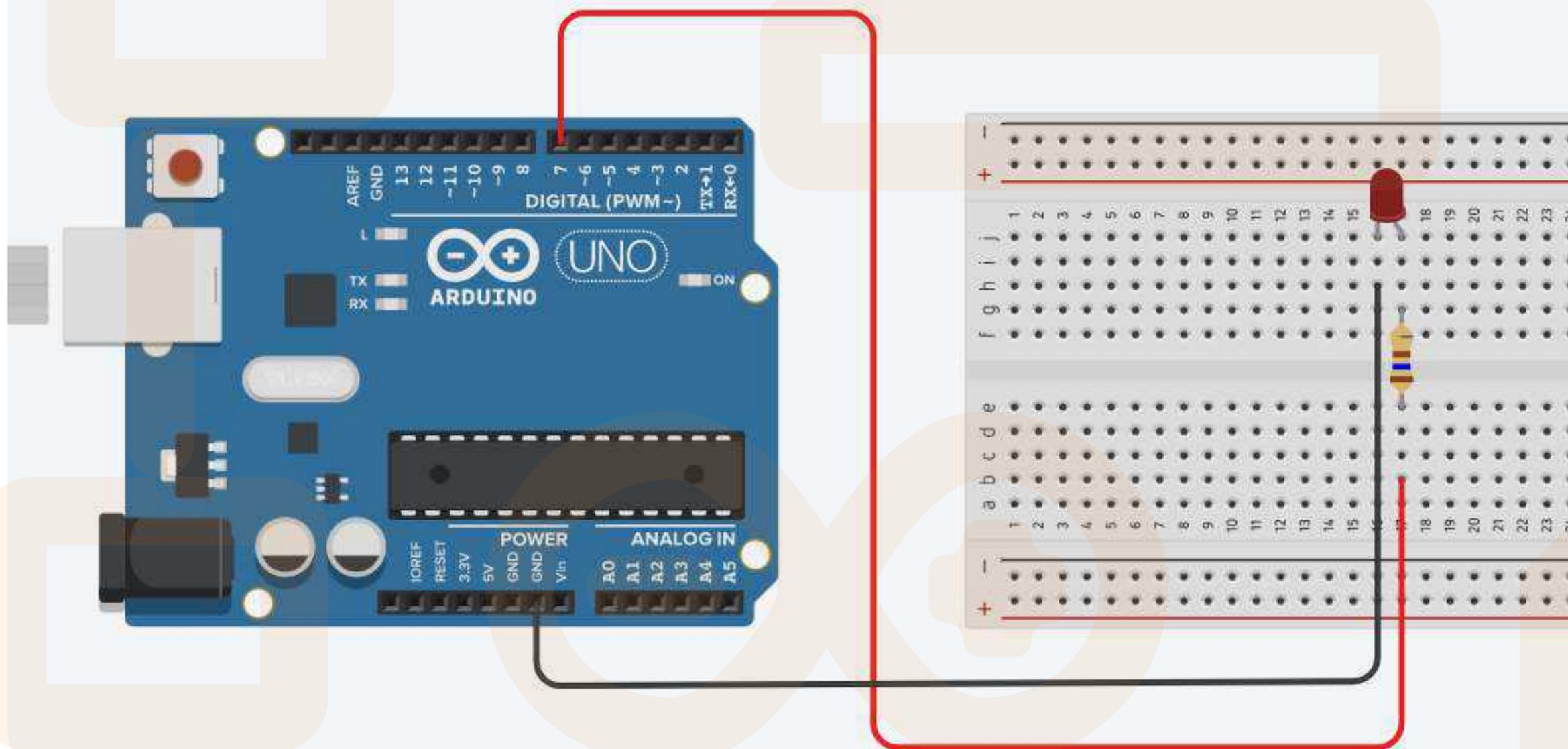
LEGGE DI OHM

SEGNALI DIGITALI

BREADBOARD

PROGRAMMAZIONE
A BLOCCHI

LED BLINKING - SCHEMA





CONCETTI:

CIRCUITI ELETTRICI

LEGGE DI OHM

SEGNALI DIGITALI

BREADBOARD

PROGRAMMAZIONE
A BLOCCHI

LED BLINKING

quando si avvia Arduino Uno

per sempre

impostare il pin digitale 7 uscita come high

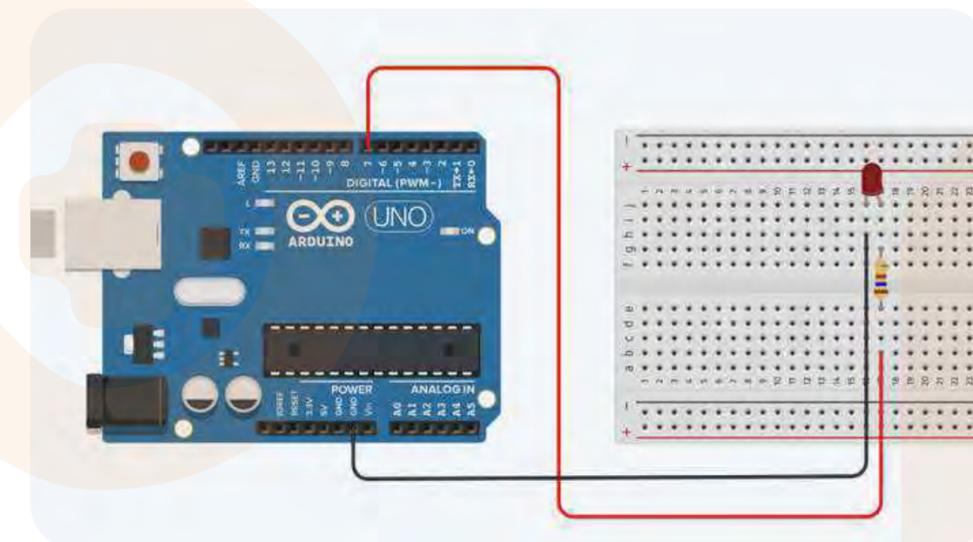
attendi 0.5 secondi

impostare il pin digitale 7 uscita come low

attendi 0.5 secondi

MATERIALI:

- LED ROSSO
- RESISTENZA DA 220 OHM
- SCHEDA ARDUINO UNO
- BREADBOARD





CONCETTI:

CIRCUITI ELETTRICI

LEGGE DI OHM

SEGNALI DIGITALI

BREADBOARD

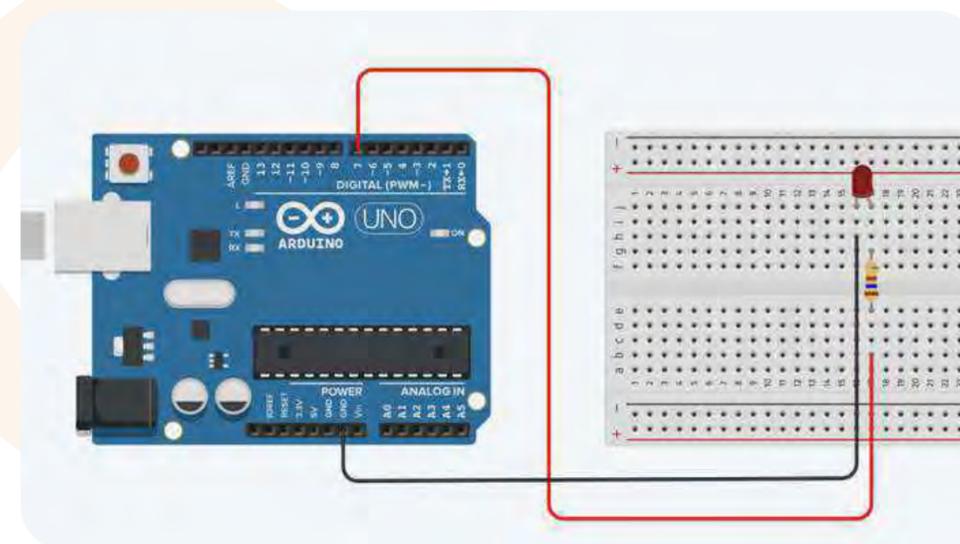
PROGRAMMAZIONE
TESTUALE

LED BLINKING

```
sketch_jun13a $  
  
void setup() {  
  // inserisci qui le istruzioni che saranno eseguite solo una volta  
  pinMode (7, OUTPUT); //individuiamo il Pin 7 come Pin di OUTPUT  
  
}  
  
void loop() {  
  // Qui inseriamo le istruzioni che vengono eseguite in loop  
  digitalWrite (7, HIGH); //inviamo corrente al pin digitale 7  
  delay(500); //ritardo di mezzosecondo  
  digitalWrite (7, LOW); //togliamo corrente al pin digitale 7  
  delay(500); //ritardo di mezzosecondo|
```

MATERIALI:

- LED ROSSO
- RESISTENZA DA 220 OHM
- SCHEDA ARDUINO UNO
- BREADBOARD





CONCETTI:

CIRCUITI ELETTRICI

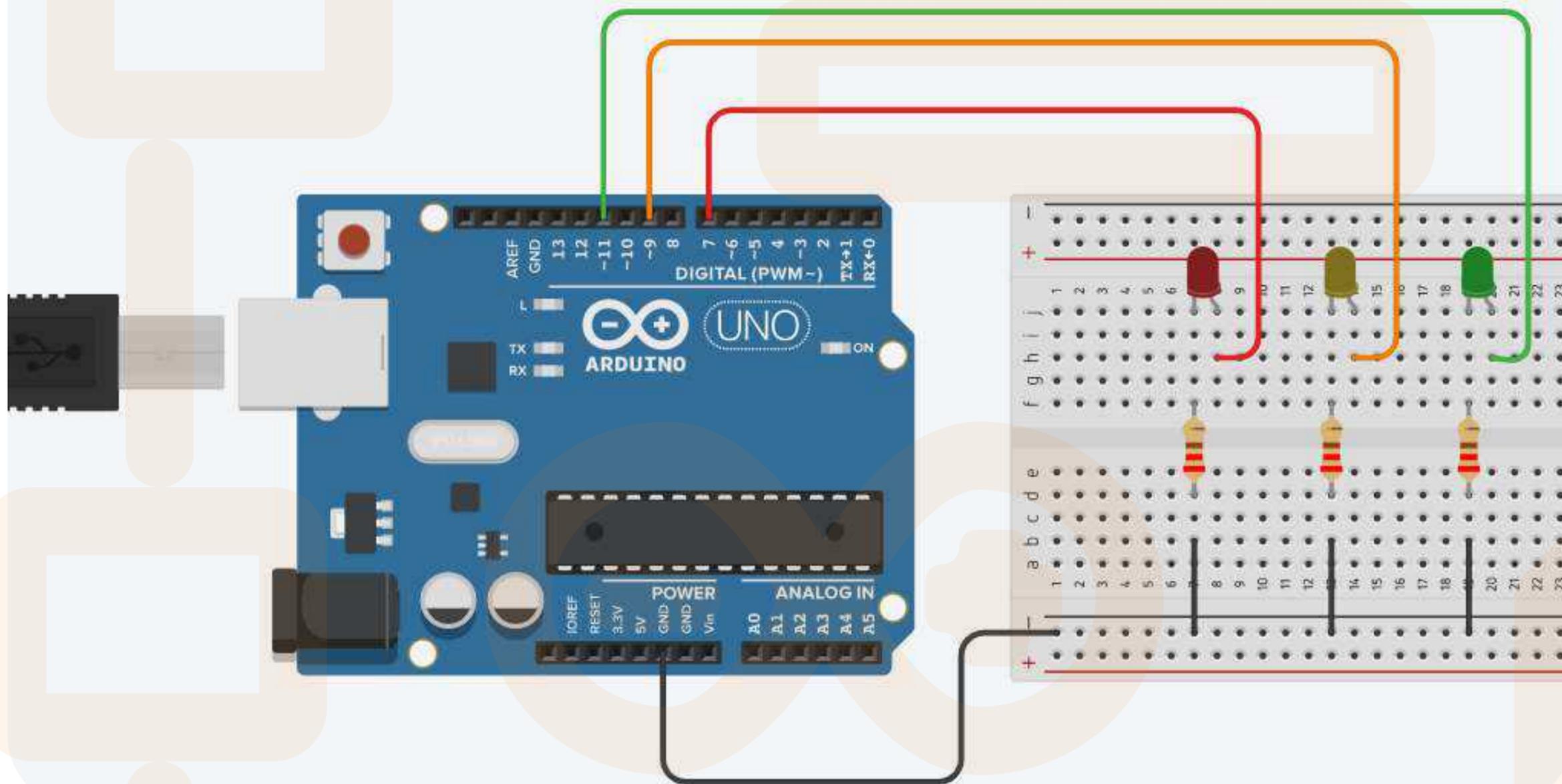
LEGGE DI OHM

SEGNALI DIGITALI

BREADBOARD

PROGRAMMAZIONE
A BLOCCHI

SEMAFORO - SCHEMA





CONCETTI:

CIRCUITI ELETTRICI

LEGGE DI OHM

SEGNALI DIGITALI

BREADBOARD

PROGRAMMAZIONE
A BLOCCHI

SEMAFORO

quando si avvia Arduino Uno

per sempre

impostare il pin digitale 7 uscita come high ▼

impostare il pin digitale 9 uscita come low ▼

impostare il pin digitale 11 uscita come low ▼

attendi 1 secondi

impostare il pin digitale 7 uscita come low ▼

impostare il pin digitale 9 uscita come low ▼

impostare il pin digitale 11 uscita come high ▼

attendi 1 secondi

impostare il pin digitale 7 uscita come low ▼

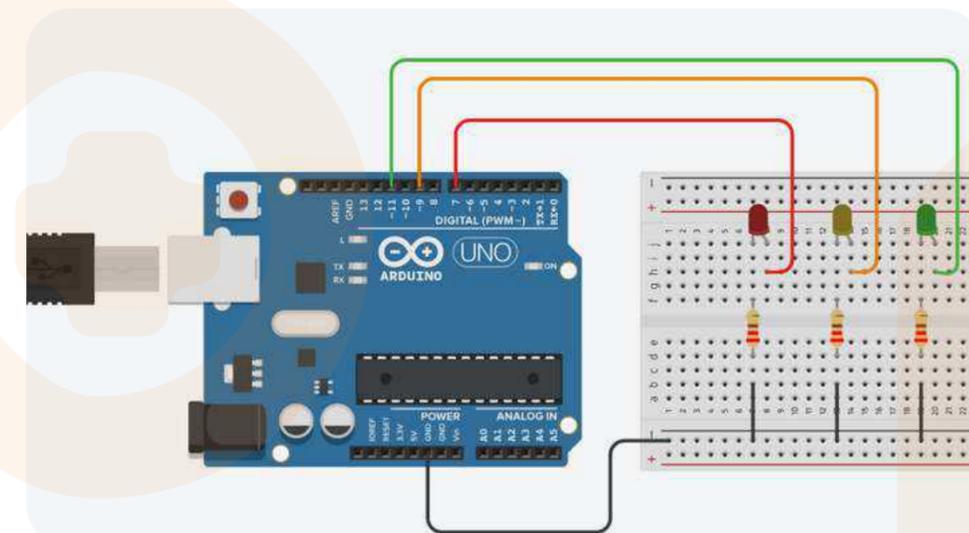
impostare il pin digitale 9 uscita come high ▼

impostare il pin digitale 11 uscita come low ▼

attendi 1 secondi

MATERIALI:

- LED ROSSO/VERDE/GIALLO
- RESISTENZE DA 220 OHM
- SCHEDA ARDUINO UNO
- BREADBOARD





CONCETTI:

CIRCUITI ELETTRICI

LEGGE DI OHM

SEGNALI DIGITALI

BREADBOARD

PROGRAMMAZIONE
A BLOCCHI

SEMAFORO

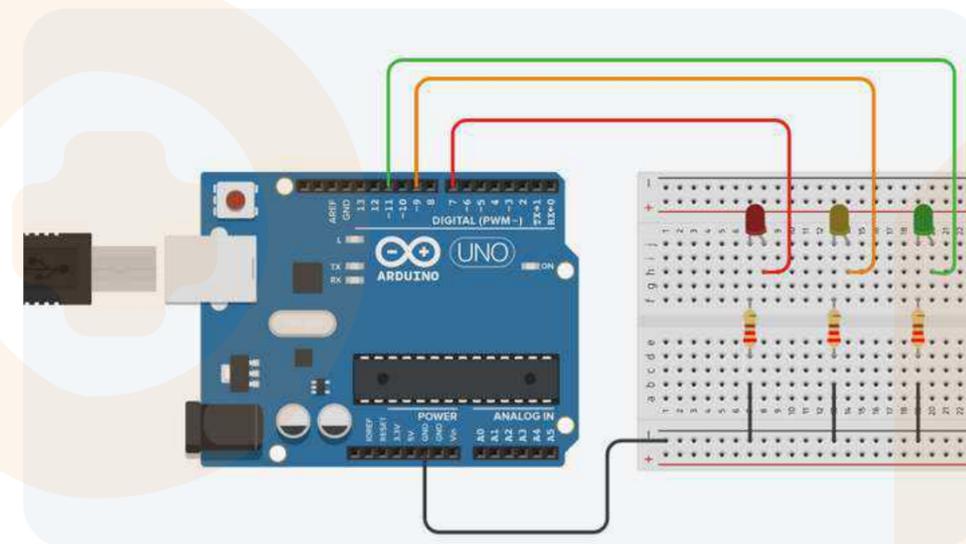
semaforo | Arduino 1.8.13
File Modifica Sketch Strumenti Aiuto

semaforo

```
int rossol = 2;
int giall01 = 3;
int verdel = 4;
int rosso2 = 5;
int giallo2 = 6;
int verde2 = 7;
void setup ()
{
  pinMode(rossol, OUTPUT);
  pinMode(giall01, OUTPUT);
  pinMode(verdel, OUTPUT);
  pinMode(rosso2, OUTPUT);
  pinMode(giallo2, OUTPUT);
  pinMode(verde2, OUTPUT);
}
void loop ()
{
  digitalWrite (rossol, HIGH);
  digitalWrite (verde2, HIGH);
  digitalWrite (rosso2, LOW);
  delay (5000);
  digitalWrite (giallo2, HIGH);
  delay (5000);
  digitalWrite (rossol, LOW);
  digitalWrite (verdel, HIGH);
  digitalWrite (verde2, LOW);
  digitalWrite (giallo2, LOW);
  digitalWrite (rosso2, HIGH);
  delay (5000);
  digitalWrite (giall01, HIGH);
  delay (5000);
  digitalWrite (verdel, LOW);
  digitalWrite (giall01, LOW);
}
```

MATERIALI:

- LED ROSSO/VERDE/GIALLO
- RESISTENZE DA 220 OHM
- SCHEDA ARDUINO UNO
- BREADBOARD





CONCETTI:

CIRCUITI ELETTRICI

LEGGE DI OHM

SEGNALI DIGITALI

BREADBOARD

PROGRAMMAZIONE
A BLOCCHI

IDEE



POSSIAMO FAR
LAMPEGGIARE IL
GIALLO?

SICURI DI
CONOSCERE LA
SEQUENZA?

POSSIAMO FARE
UN DOPPIO
SEMAFORO?

E IL PRESEPE?

E LE COSTELLAZIONI?



CONCETTI:

CIRCUITI ELETTRICI

PIN ANALOGICO

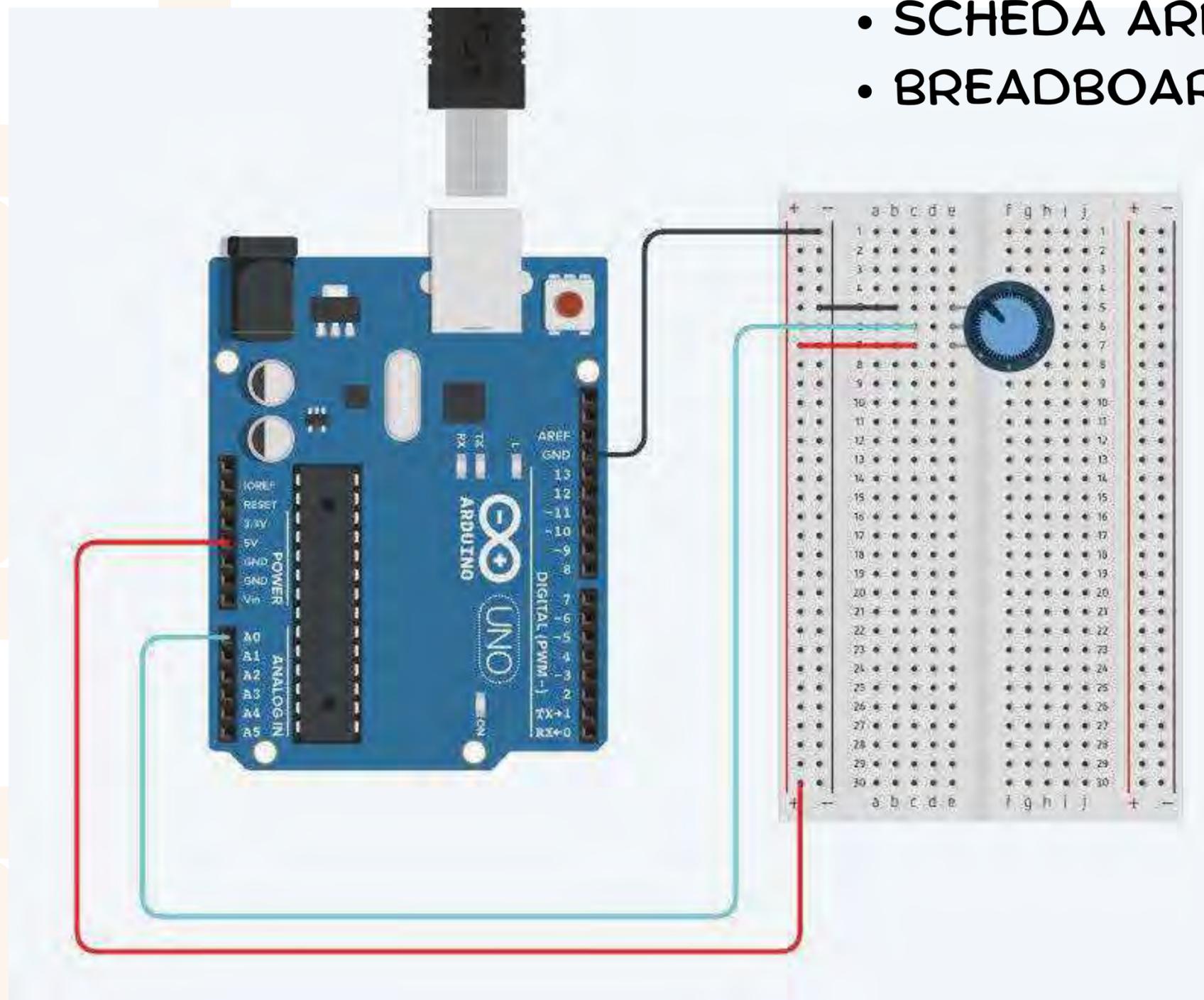
RESISTENZA
VARIABILE

PROGRAMMAZIONE
A BLOCCHI

POTENZIOMETRO

MATERIALI:

- POTENZIOMETRO
- SCHEDA ARDUINO UNO
- BREADBOARD





CONCETTI:

CIRCUITI ELETTRICI

PIN ANALOGICO

RESISTENZA
VARIABILE

PROGRAMMAZIONE
A BLOCCHI

POTENZIOMETRO



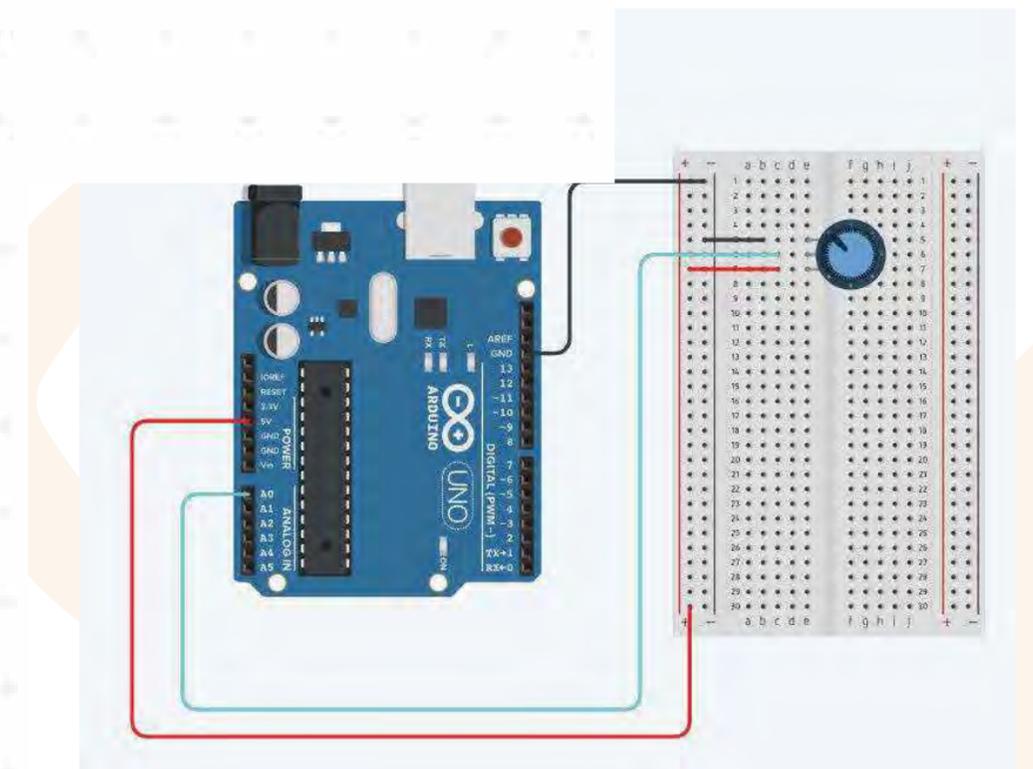
ARDUINO



SPRITE

MATERIALI:

- POTENZIOMETRO
- SCHEDA ARDUINO UNO
- BREADBOARD





CONCETTI:

CIRCUITI ELETTRICI

PIN ANALOGICO

RESISTENZA
VARIABILE

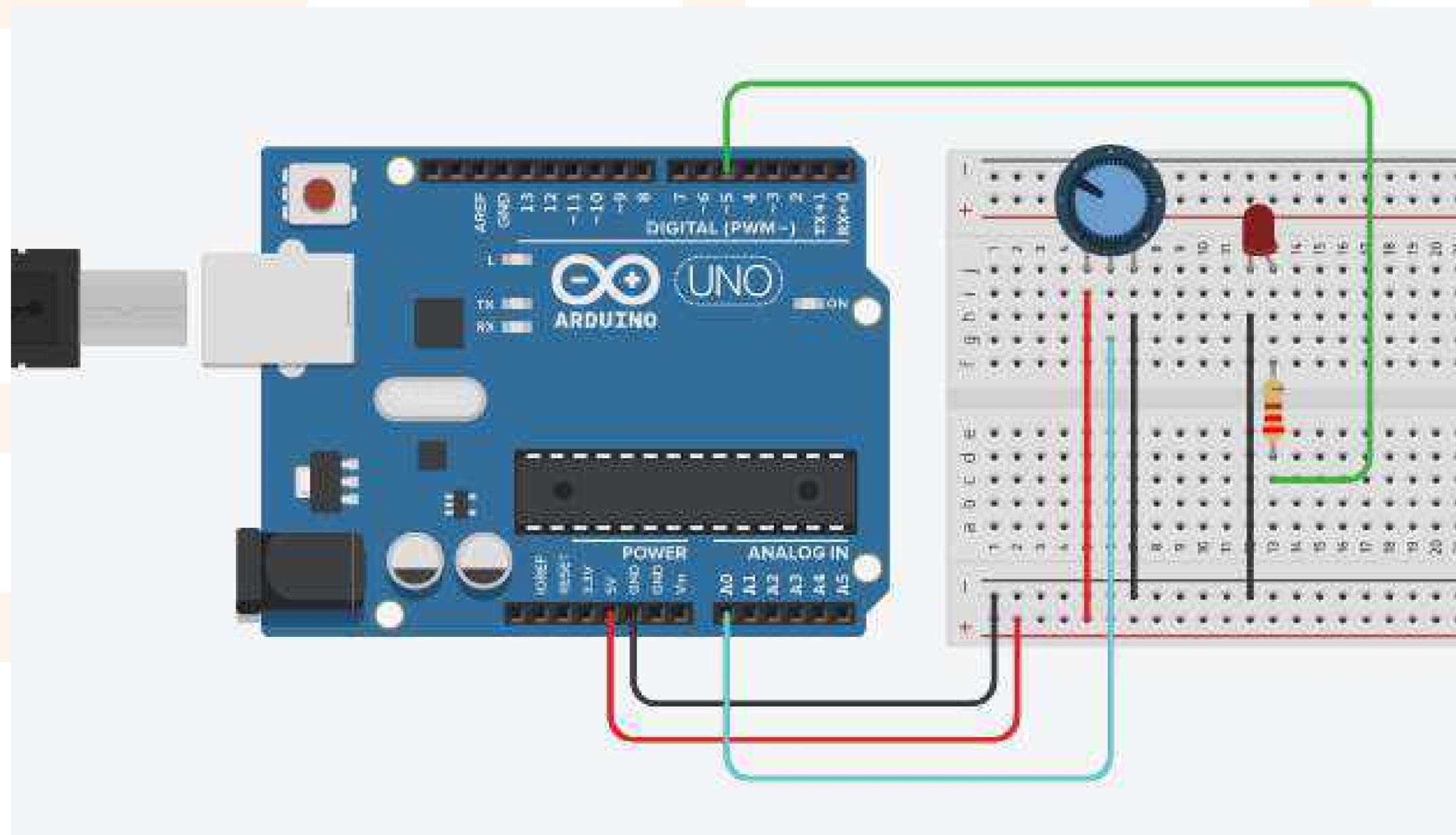
PWM

PROGRAMMAZIONE
A BLOCCHI

POTENZIOMETRO E LED

MATERIALI:

- POTENZIOMETRO
- LED
- SCHEDA ARDUINO UNO
- LED 220 OHM
- BREADBOARD





CONCETTI:

CIRCUITI ELETTRICI

PIN ANALOGICO

RESISTENZA
VARIABILE

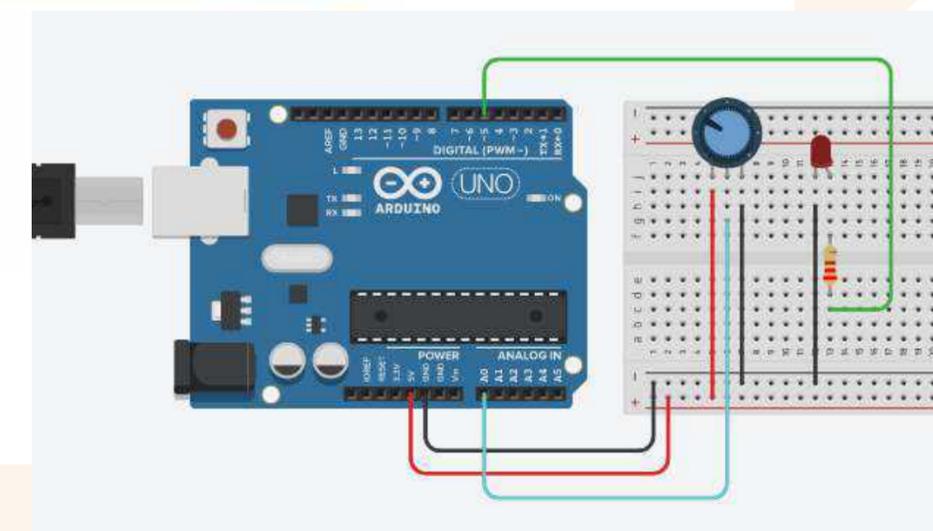
PWM

PROGRAMMAZIONE
A BLOCCHI

POTENZIOMETRO E LED

MATERIALI:

- POTENZIOMETRO
- LED
- SCHEDA ARDUINO UNO
- LED 220 OHM
- BREADBOARD





CONCETTI:

CIRCUITI ELETTRICI

PIN ANALOGICO

RESISTENZA
VARIABILE

PWM

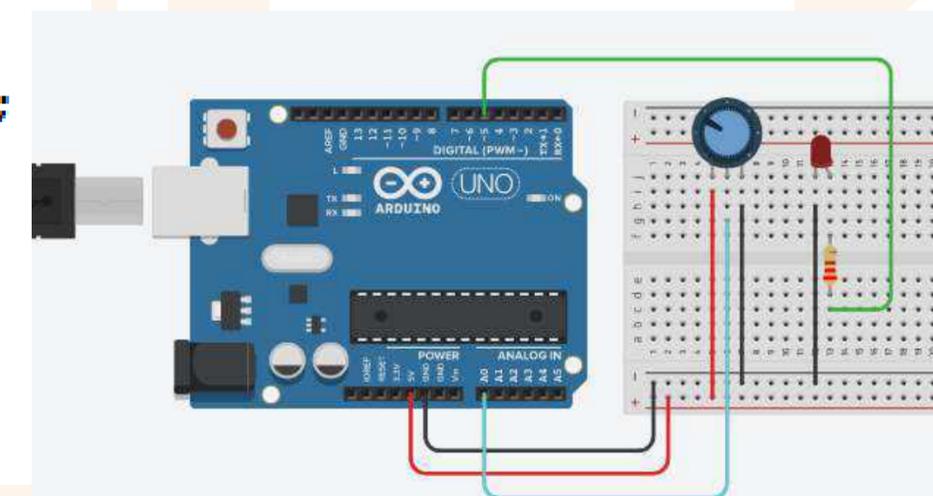
PROGRAMMAZIONE
TESTUALE

POTENZIOMETRO E LED

led_e_potenziometro | Arduino 1.8.19

File Modifica Sketch Strumenti Aiuto

```
led_e_potenziometro §  
  
int pinPotenziometro = A0;  
int pinLed = 5;  
  
void setup() {  
  pinMode(pinLed, OUTPUT);  
}  
  
void loop() {  
  
  // Legge il valore dal potenziometro  
  int valPotenziometro = analogRead(pinPotenziometro);  
  // Comanda il LED data la posizione assunta dal  
  // potenziometro  
  analogWrite(pinLed, valPotenziometro/4);  
}
```





CONCETTI:

PIN DIGITALE

PWM

DEEPEN
UNDERSTANDING



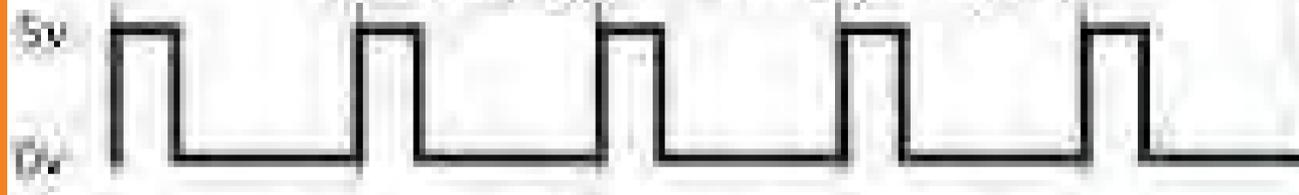
POTENZIOMETRO E LED

Pulse Width Modulation

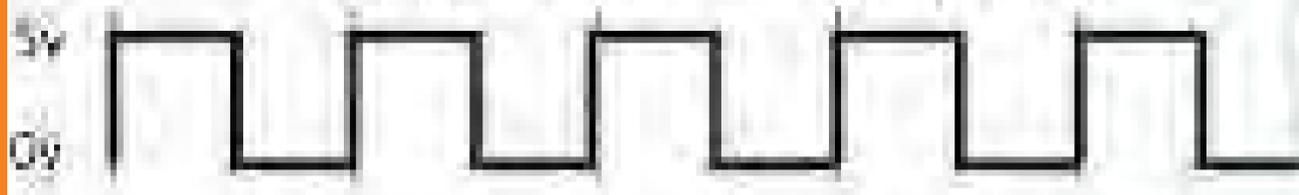
0% Duty Cycle - analogWrite(0)



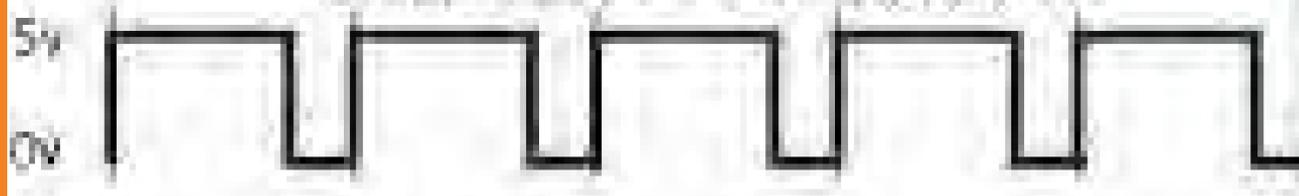
25% Duty Cycle - analogWrite(64)



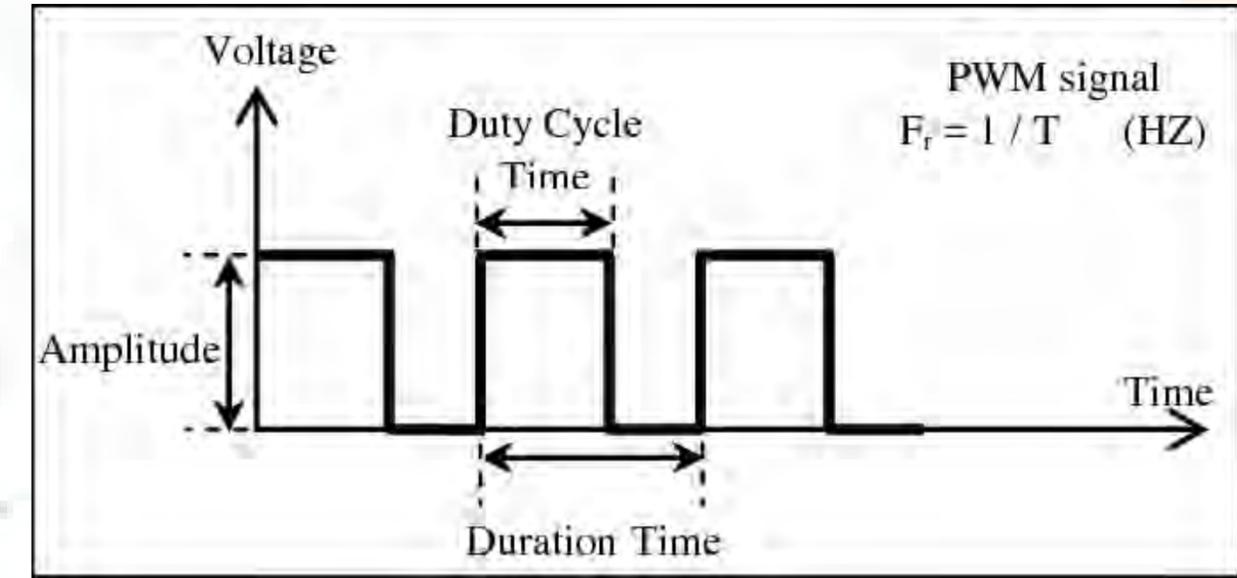
50% Duty Cycle - analogWrite(127)



75% Duty Cycle - analogWrite(191)



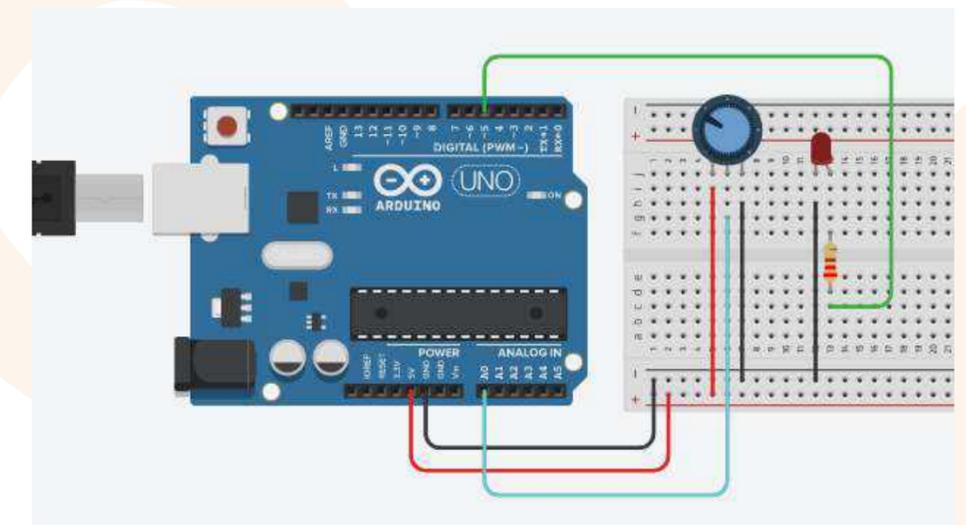
100% Duty Cycle - analogWrite(255)



CONCETTI:

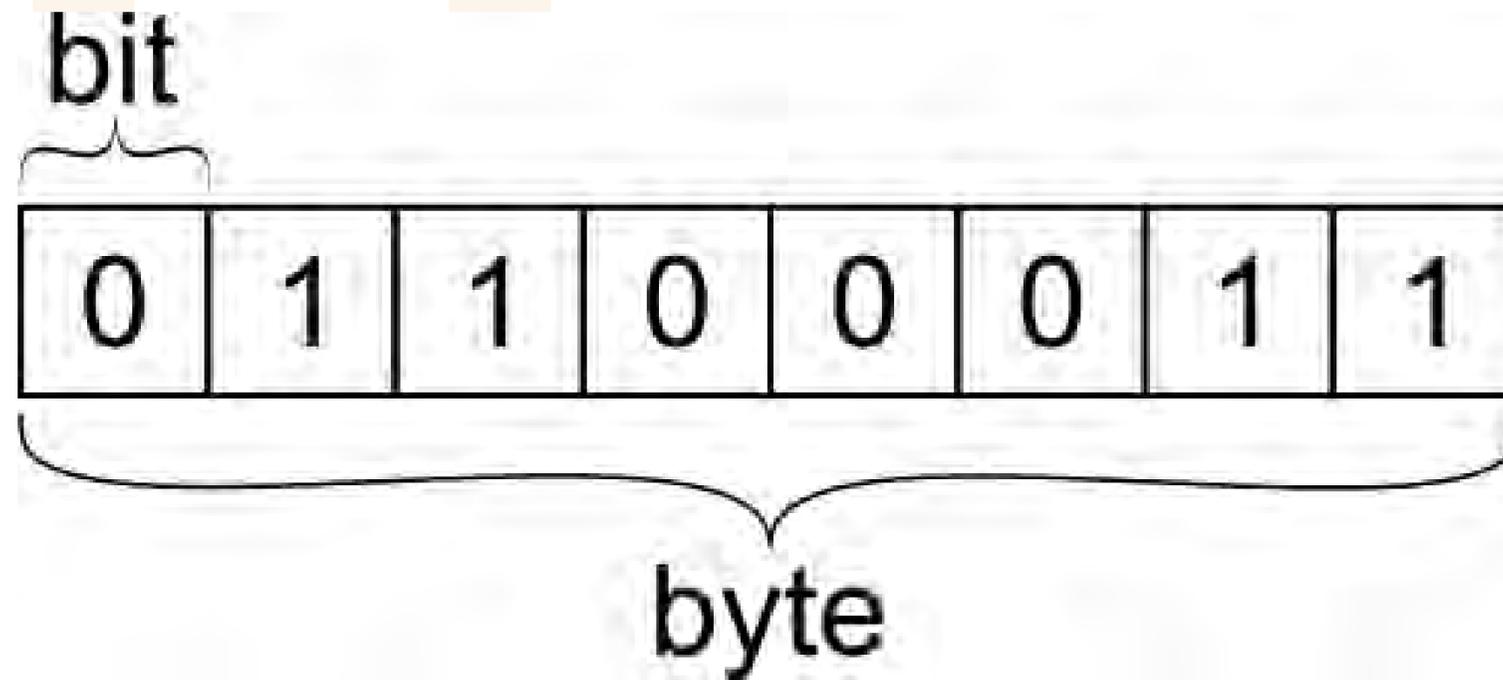
PIN DIGITALE

PWM





CODICE BINARIO E SEGNALI



n	2^n
0	2^0 1
1	2^1 2
2	2^2 4
3	2^3 8
4	2^4 16
5	2^5 32
6	2^6 64
7	2^7 128
8	2^8 256
9	2^9 512
10	2^{10} 1024

CONCETTI:

SEGNALI DIGITALI

SEGNALI ANALOGICI

NUMERI A 8 BIT

NUMERI A 10 BIT

$C \times B^P$

C = CIFRA

B = BASE NUMERAZIONE

P = POSIZIONE



IDEE

CONCETTI:

CIRCUITI ELETTRICI

PIN ANALOGICO

RESISTENZA
VARIABILE

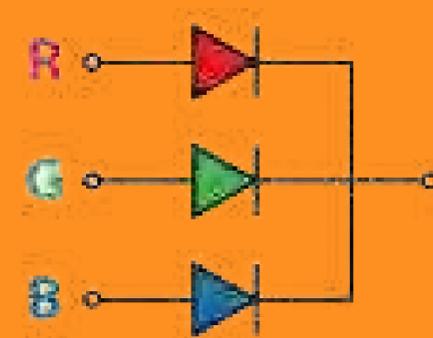
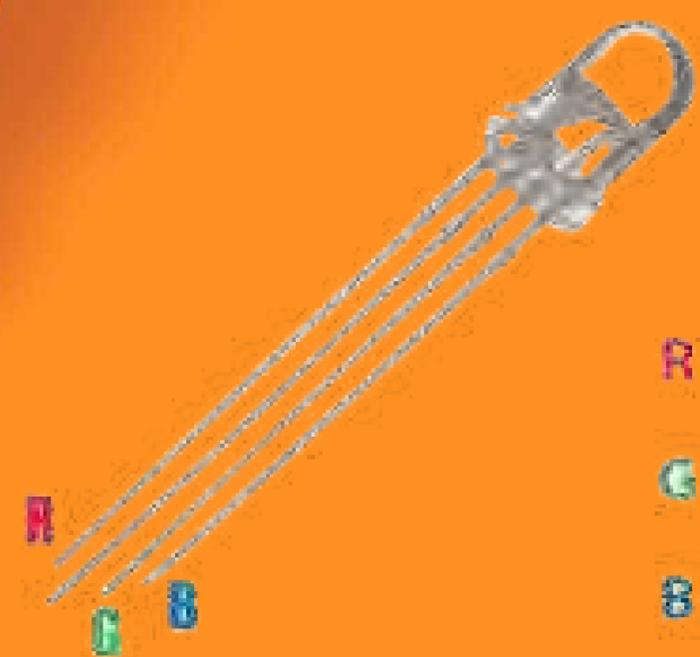
PWM

LED RGB

E SE USASSIMO
UN LED RGB?

QUANTE
RESISTENZE PER
UN LED RGB?

QUANTI
POTENZIOMETRI
PER UN LED RGB?





CONCETTI:

CIRCUITI ELETTRICI

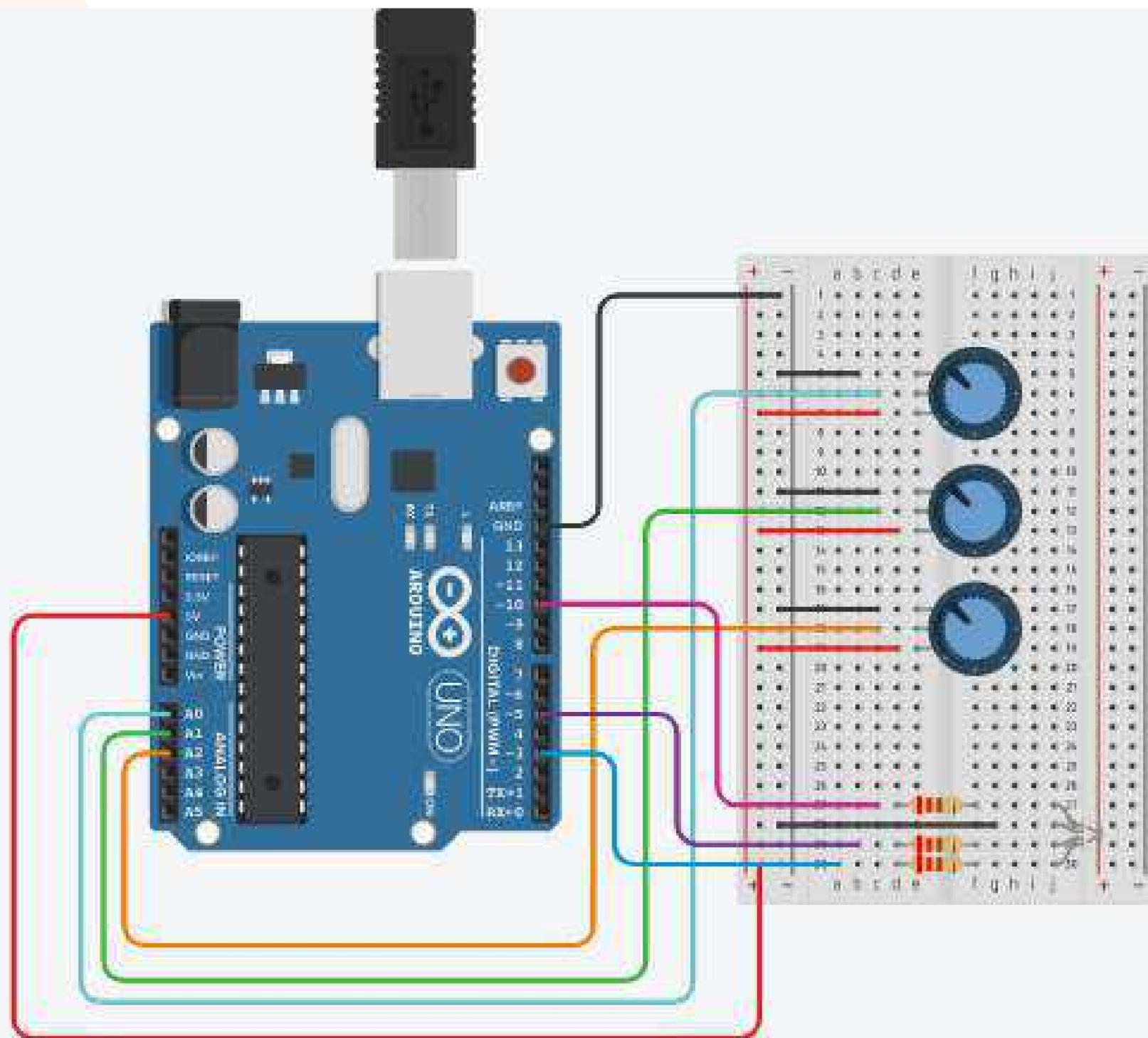
PIN ANALOGICO

RESISTENZA
VARIABILE

PWM

LED RGB

POTENZIOMETRO E LED RGB





IDEE

CONCETTI:

CIRCUITI ELETTRICI

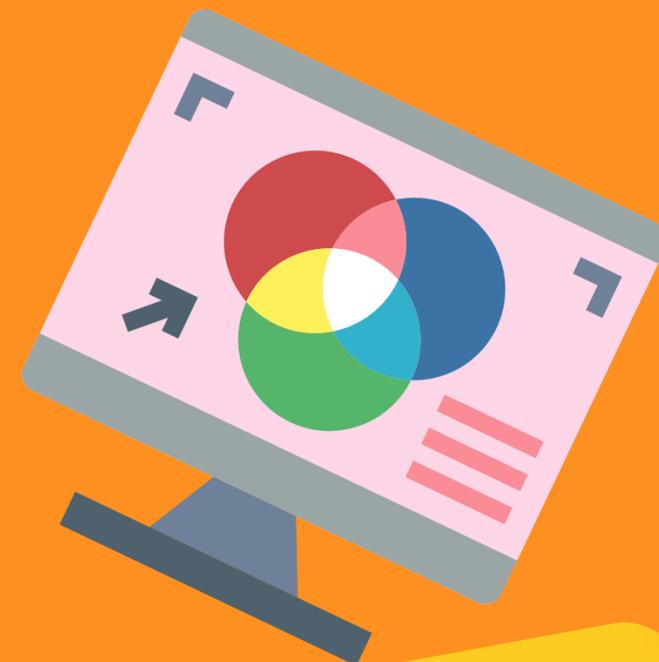
PIN ANALOGICO

RESISTENZA
VARIABILE

PWM

LED RGB

CHI PROVA A
PROGRAMMARLO?



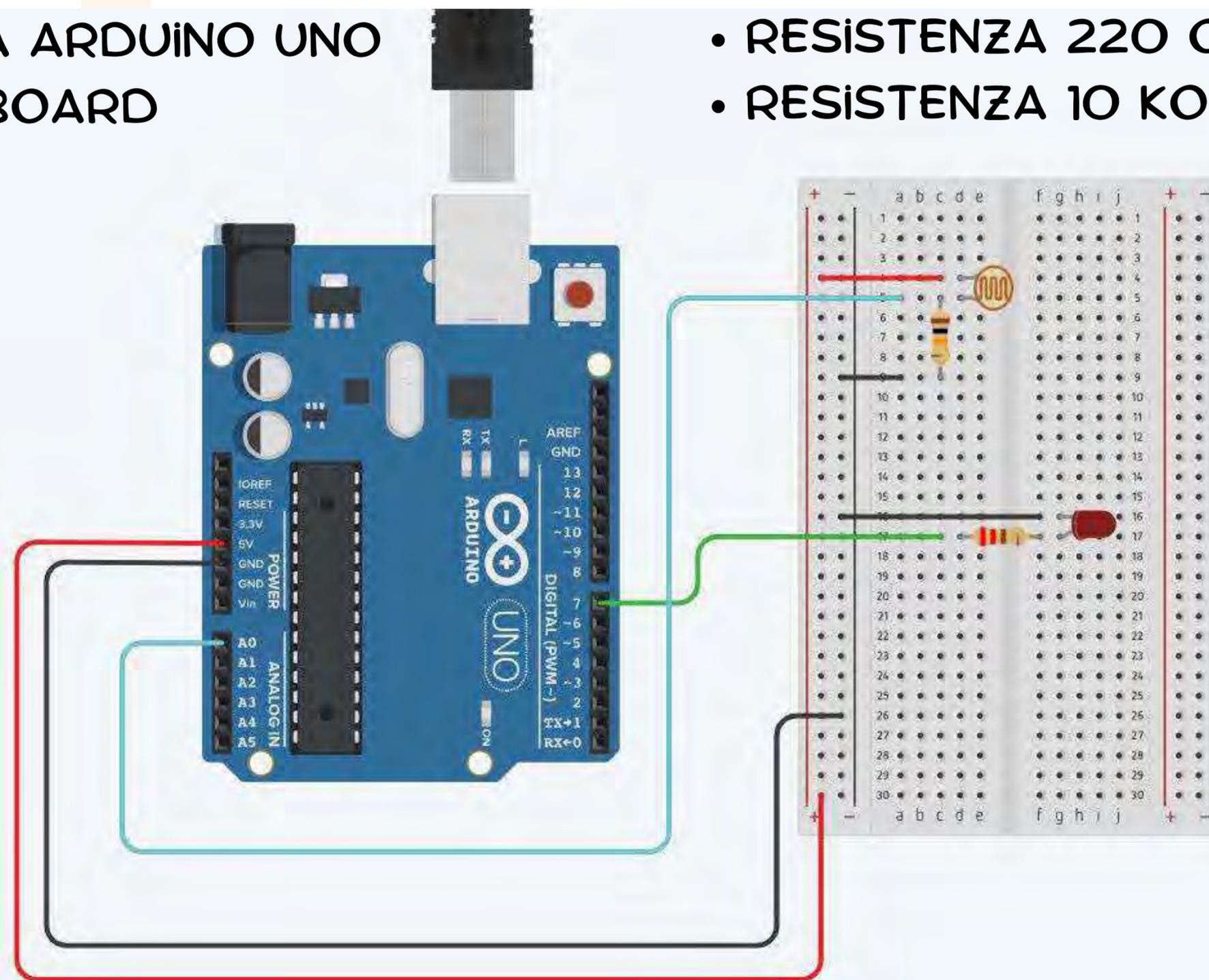


CREPUSCOLARE

MATERIALI:

- FOTORESISTENZA
- SCHEDA ARDUINO UNO
- BREADBOARD

- LED
- RESISTENZA 220 OHM
- RESISTENZA 10 KOHM



CONCETTI:

CIRCUITI ELETTRICI

RESISTENZA
VARIABILE

PARTITORE DI
TENSIONE



CONCETTI:

CIRCUITI ELETTRICI

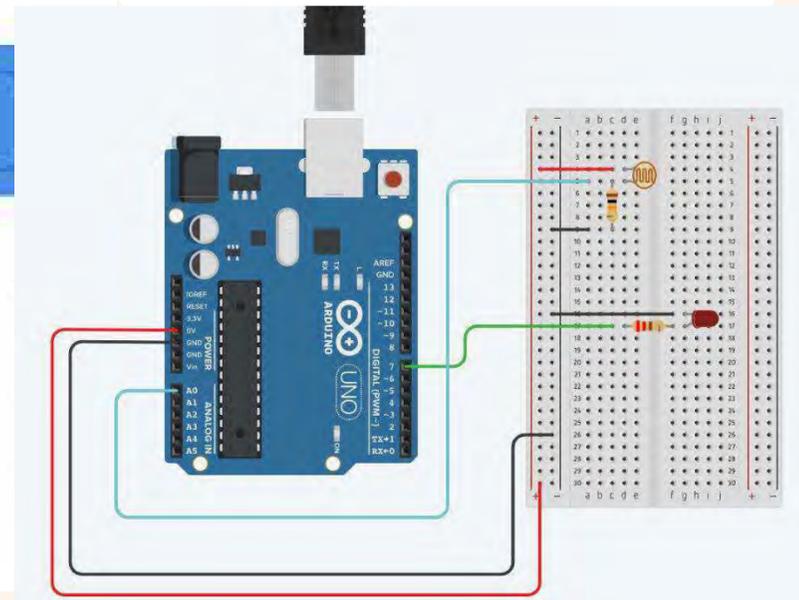
RESISTENZA
VARIABILE

PARTITORE DI
TENSIONE

PROGRAMMAZIONE
A BLOCCHI

CREPUSCOLARE

```
quando si avvia Arduino Uno
per sempre
  imposta luce a leggi pin analogico (A) 0
  se (luce / 4 < 75) allora
    imposta il pin digitale 7 uscita come high
  altrimenti
    imposta il pin digitale 7 uscita come low
  attendi 1 secondi
```





CONCETTI:

CIRCUITI ELETTRICI

RESISTENZA
VARIABILE

PARTITORE DI
TENSIONE

PROGRAMMAZIONE
TESTUALE

CREPUSCOLARE

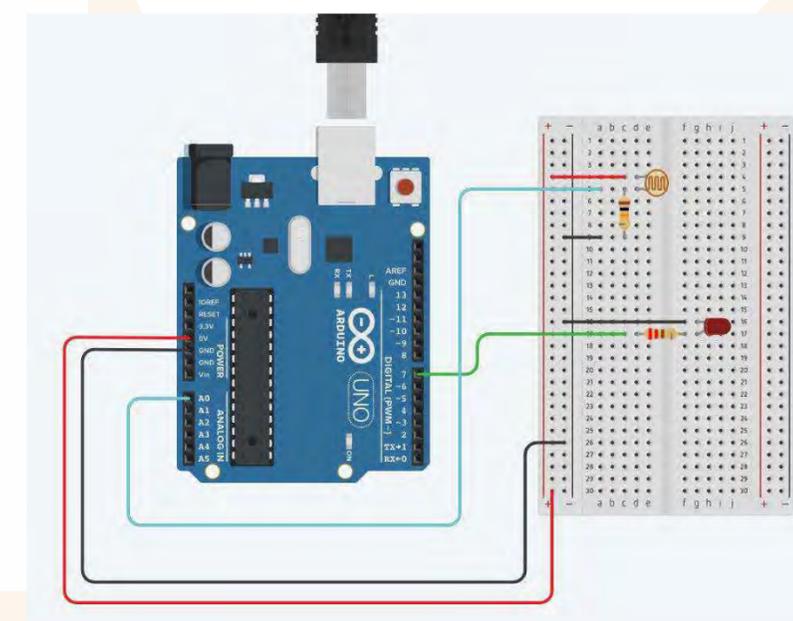
led_e_fotoresistenza | Arduino 1.8.19

File Modifica Sketch Strumenti Aiuto

```
led_e_fotoresistenza $
int luce = 0;

void setup()
{
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  luce = analogRead(A0);
  if (luce / 10 < 75) {
    digitalWrite(3, HIGH);
  } else {
    digitalWrite(3, LOW);
  }
  Serial.println((luce / 10));
  delay(10); // Delay a little bit to improve simulation performance
}
```





CONCETTI:

FOTORESISTENZA

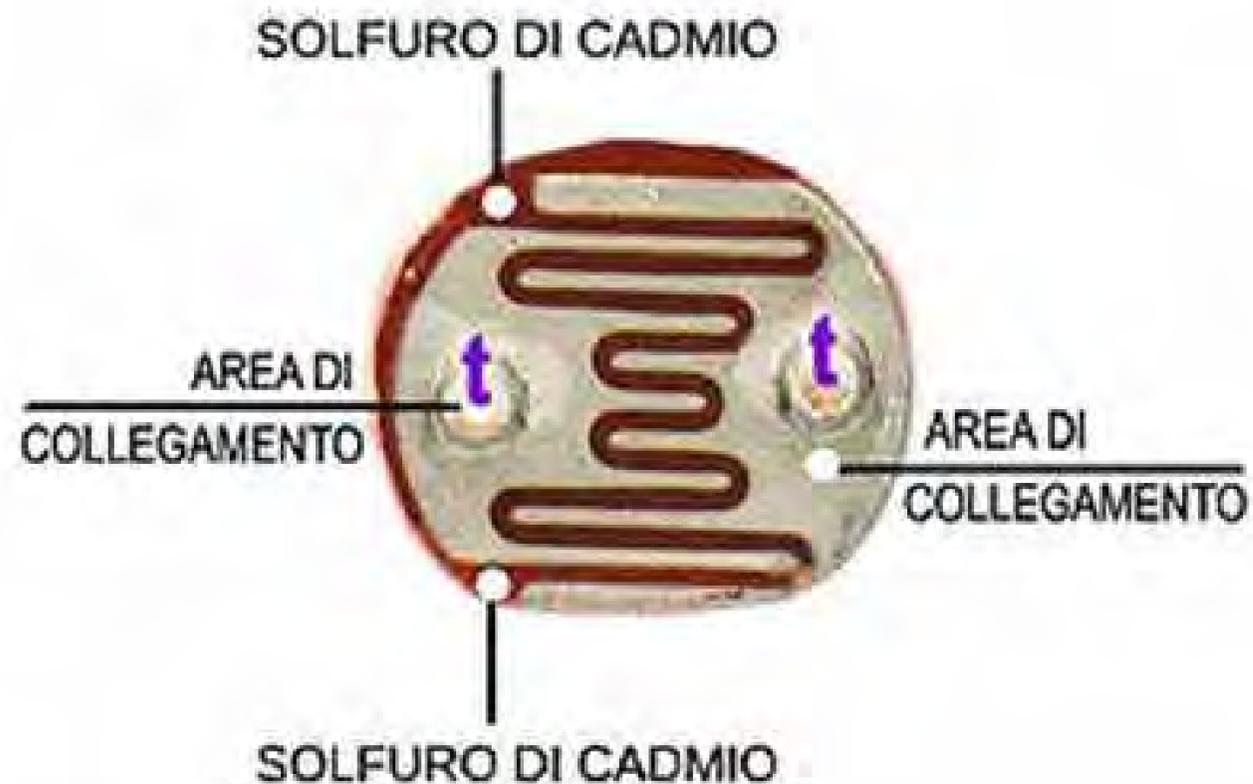
DEEPEN
UNDERSTANDING



FOTORESISTENZA

CONCETTI:

FOTORESISTENZA



LA FOTORESISTENZA È UN COMPONENTE ELETTRONICO LA CUI RESISTENZA È INVERSAMENTE PROPORZIONALE ALLA QUANTITÀ DI LUCE CHE LO COLPISCE.

SI COMPORTA COME UN NORMALE RESISTORE, MA IL SUO VALORE IN OHM DIMINUISCE A MANO A MANO CHE AUMENTA L'INTENSITÀ DELLA LUCE CHE LA COLPISCE.

[HTTPS://WWW.WETURTLE.ORG/DETTAGLIO-TUTORIAL/85/TUTORIAL-LA-FOTORESISTENZA-E-IL-LED-CON-ARDUINO.HTML](https://www.weturtle.org/dettaglio-tutorial/85/tutorial-la-fotoresistenza-e-il-led-con-arduino.html)

[HTTPS://WWW.PHYSICS-AND-RADIO-ELECTRONICS.COM/ELECTRONIC-DEVICES-AND-CIRCUITS/PASSIVE-COMPONENTS/RESISTORS/PHOTORESISTOR.HTML](https://www.physics-and-radio-electronics.com/electronic-devices-and-circuits/passive-components/resistors/photoresistor.html)

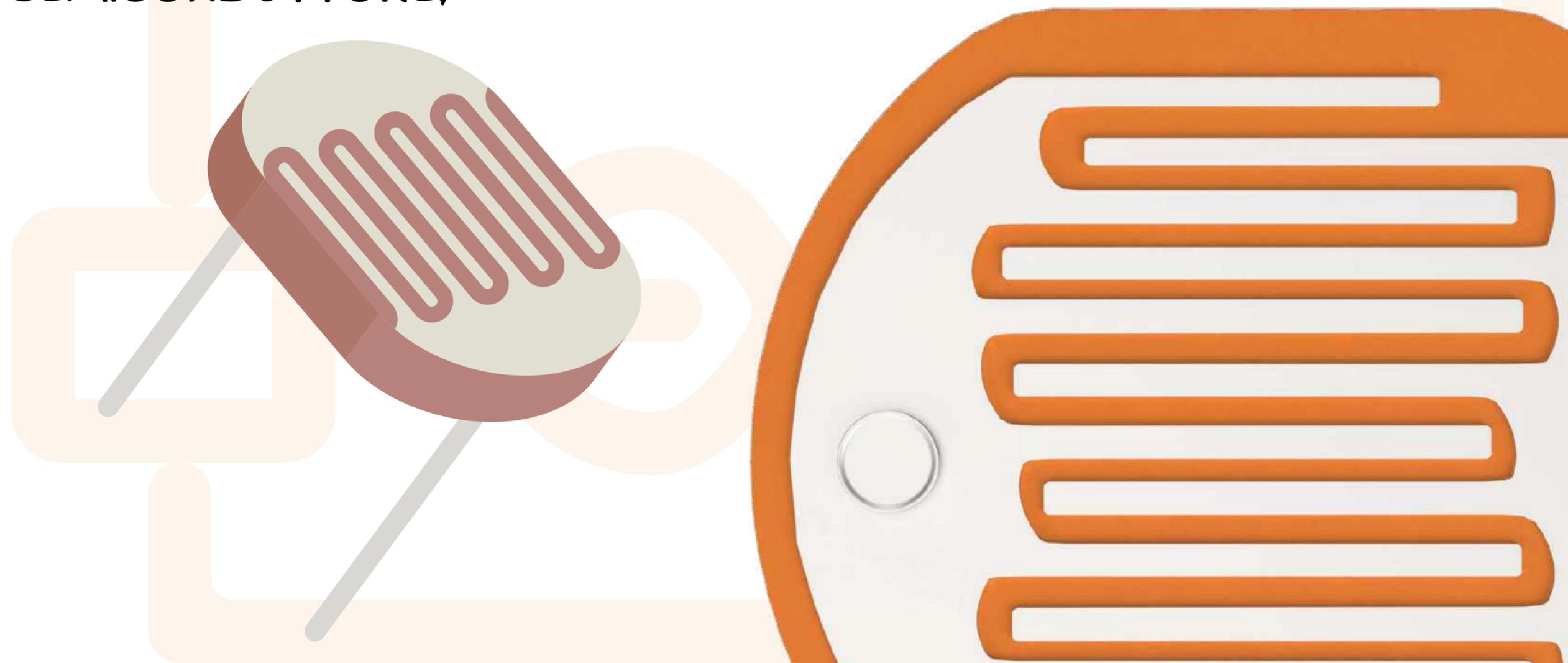


FOTORESISTENZA

IL FUNZIONAMENTO DELLA FOTORESISTENZA SI BASA SULL'EFFETTO CONDUTTIVO DEI MATERIALI SEMICONDUCTORI; LA CONDUCEBILITÀ DI UN SEMICONDUCTORE AUMENTA CON L'AUMENTARE DELLE LACUNE E DEGLI ELETTRONI LIBERI: QUANDO UNA RADIAZIONE LUMINOSA COLPISCE IL SEMICONDUCTORE,

CONCETTI:

FOTORESISTENZA



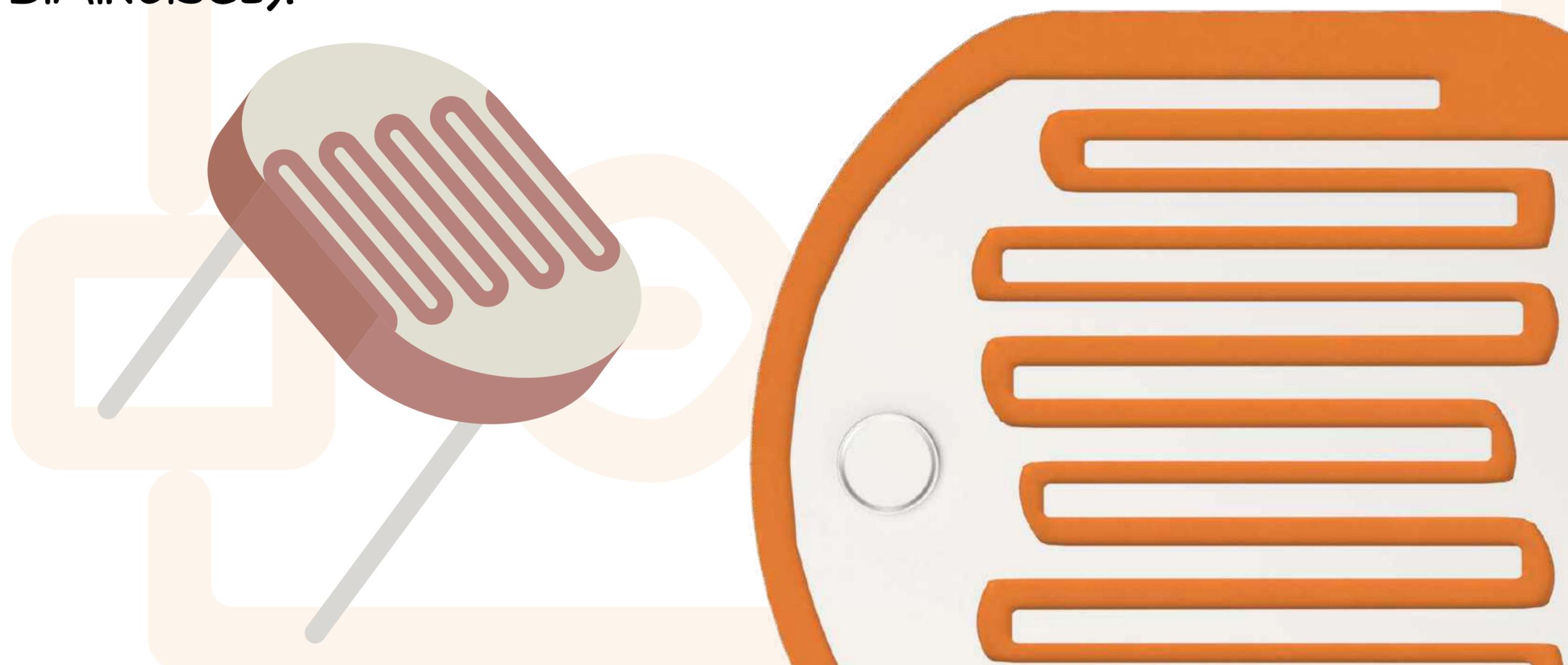


FOTORESISTENZA

I FOTONI POSSONO ROMPERE IL LEGAME COVALENTE (ELETTRONE-LACUNA) GENERANDO UNA COPPIA ELETTRONE LIBERO-LACUNA, CONTRIBUENDO COSÌ AD AUMENTARE I PORTATORI DI CARICA E DI CONSEGUENZA LA CONDUCEBILITÀ DEL MATERIALE (QUINDI LA RESISTENZA DEL MATERIALE DIMINUISCE).

CONCETTI:

FOTORESISTENZA





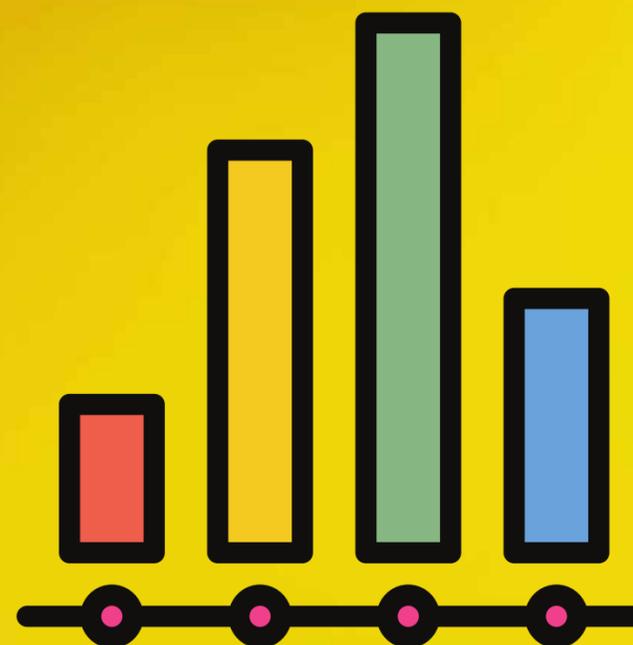
CONCETTI:

SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

LEVEL UP





CONCETTI:

SENSORE DI
TEMPERATURA
LM35

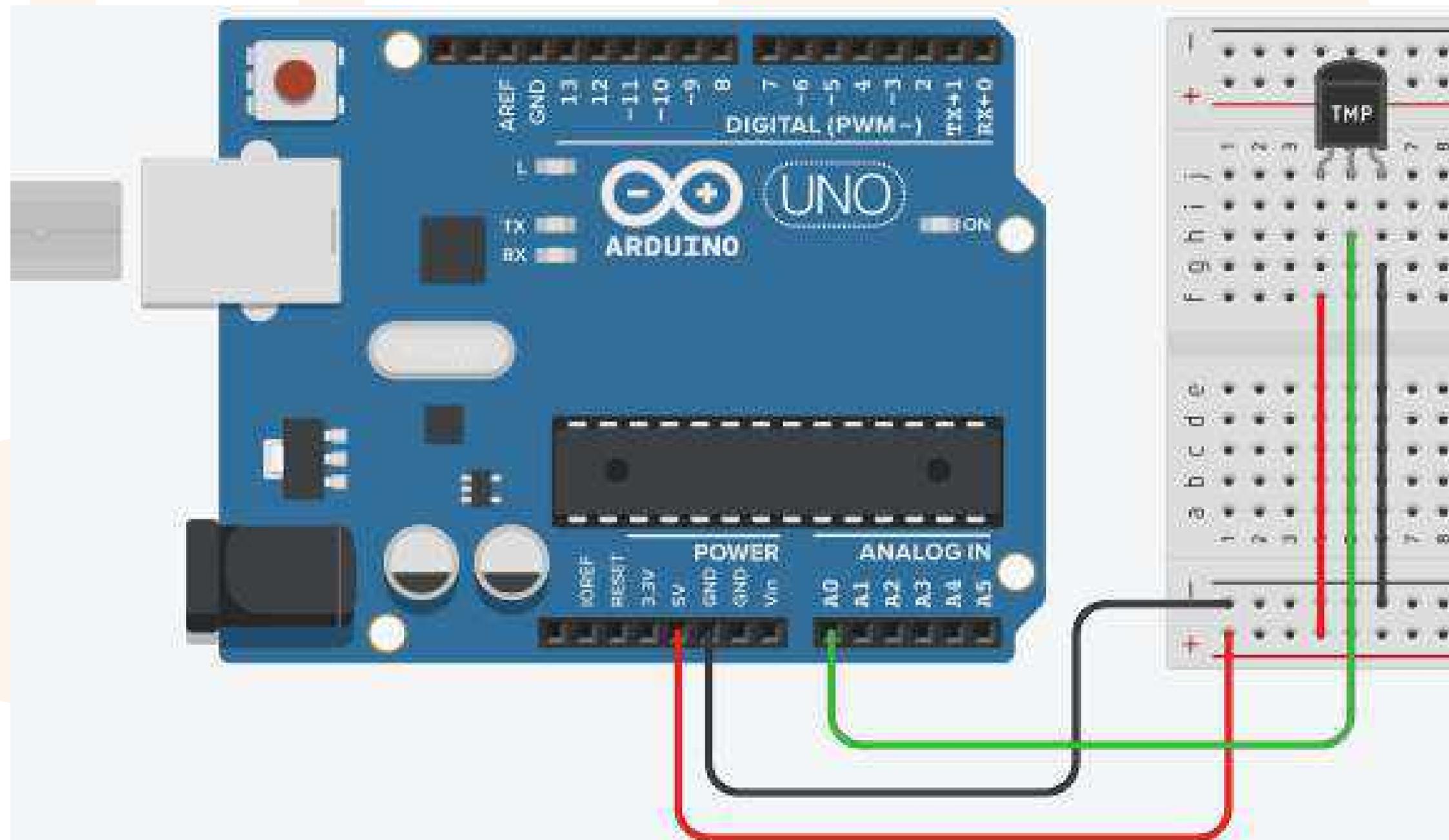
PIN ANALOGICO

EQUAZIONE DI
TARATURA

TEMPERATURA

MATERIALI:

- SENSORE LM35
- SCHEDA ARDUINO UNO
- BREADBOARD





CONCETTI:

SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

DEEPEN
UNDERSTANDING



CONCETTI:

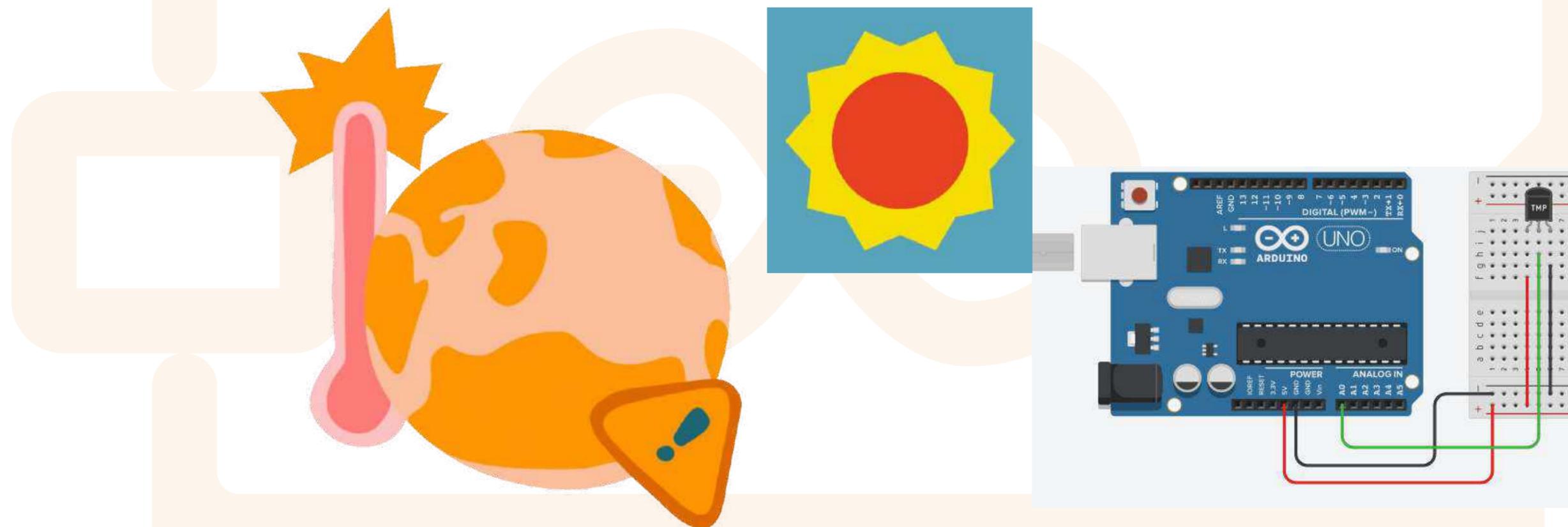
SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

TEMPERATURA

PER CONVERTIRE IL VALORE IN TEMPERATURA UTILIZZIAMO UNA FORMULA. QUESTA SI BASA SULLE INFORMAZIONI FORNITE NELLA SCHEDA TECNICA DEL SENSORE DI TEMPERATURA (UN MANUALE CHE DESCRIVE IL SENSORE E IL SUO FUNZIONAMENTO).





TEMPERATURA

QUESTA FORMULA CONVERTE IL CAMPO ANALOGICO DA 0 A 1.023 NEL CAMPO DEL SENSORE DI TEMPERATURA, CHE È DA -40°C A +125°C.

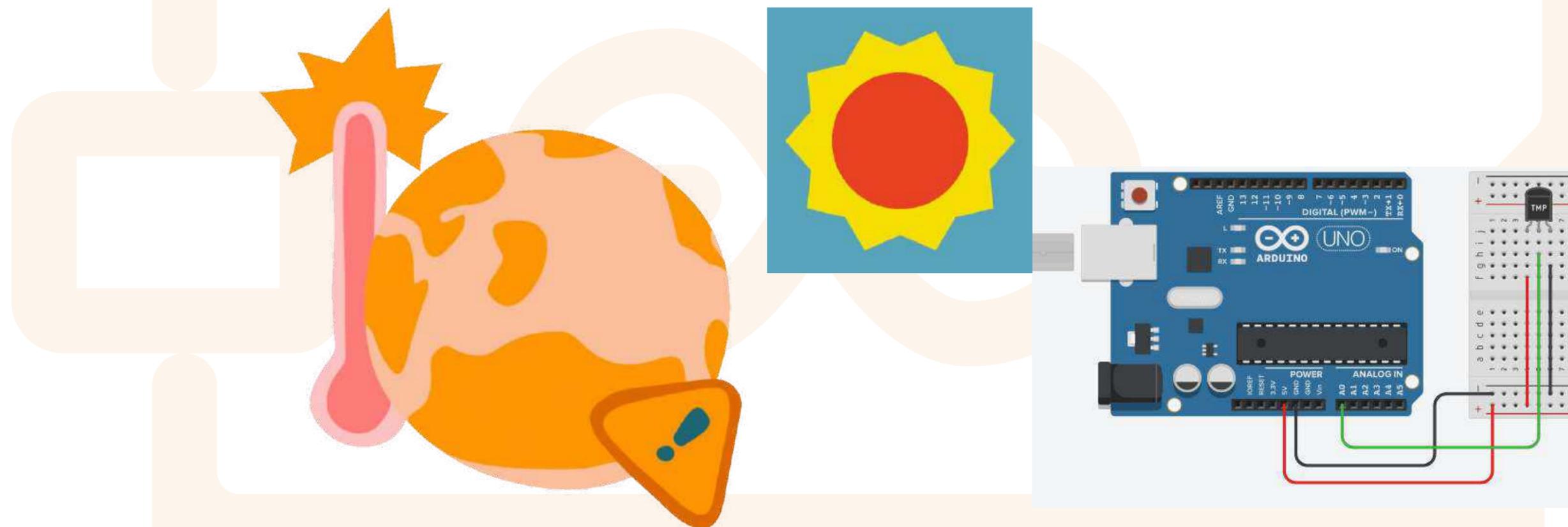
$$\text{TEMPC} = (((\text{SENSORVAL}/1024) * 5) - .5) * 100;$$

CONCETTI:

SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA





CONCETTI:

SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

TEMPERATURA

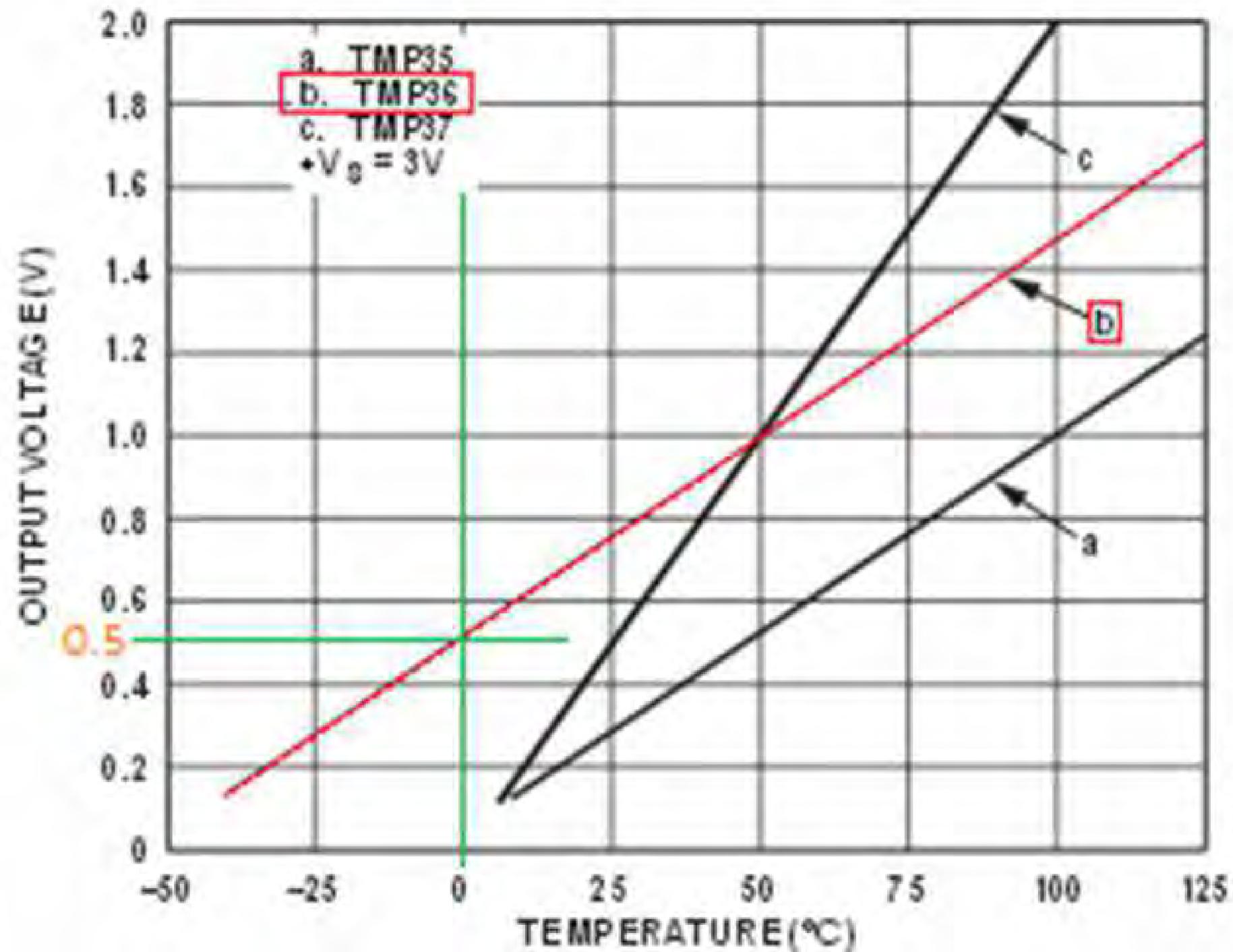


Figure 6. Output Voltage vs. Temperature



CONCETTI:

SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

TEMPERATURA

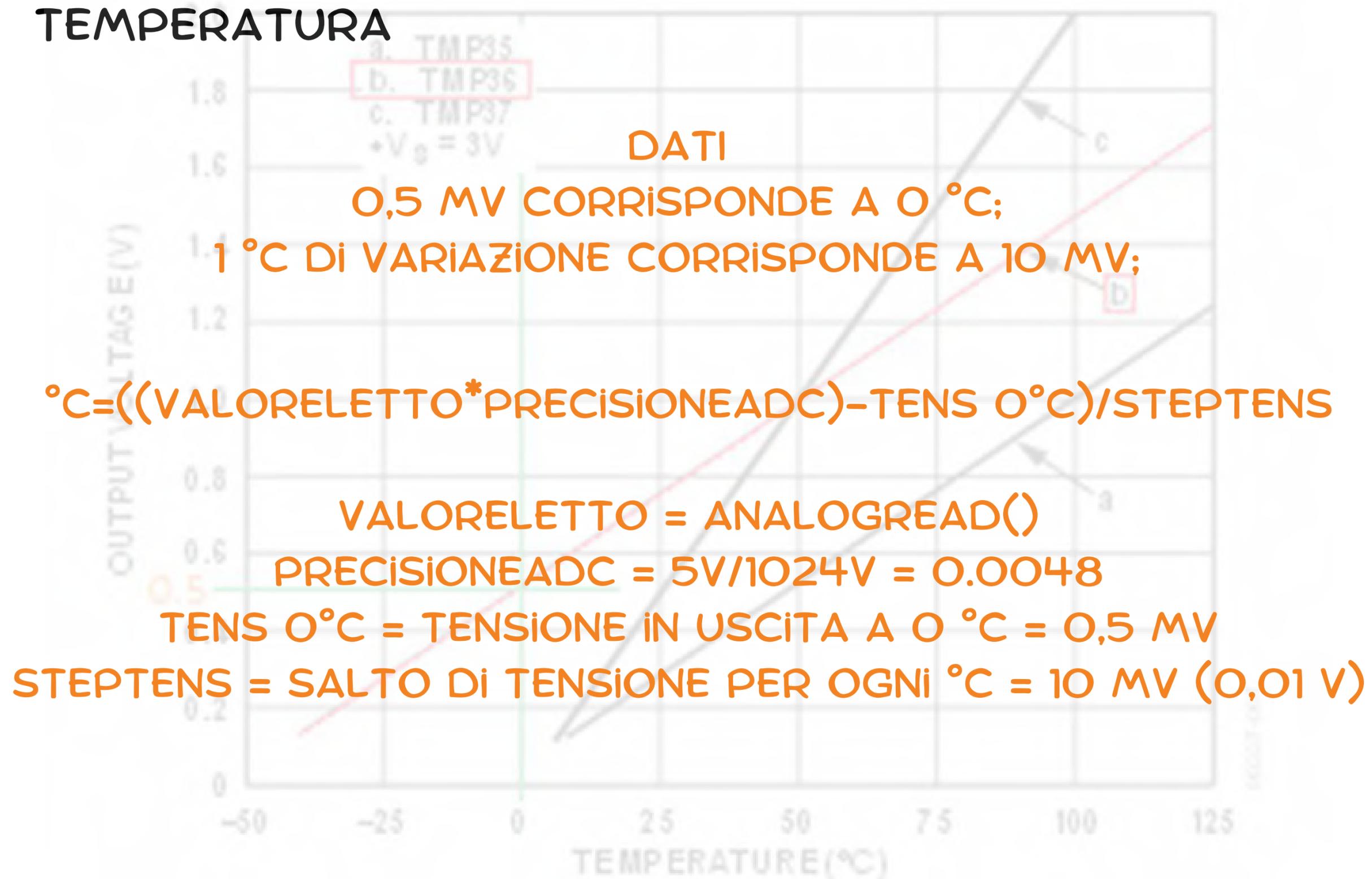


Figure 6. Output Voltage vs. Temperature



TEMPERATURA

CONCETTI:

SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

PROGRAMMAZIONE
A BLOCCHI

ARDUINO

```
quando si avvia Arduino Uno
per sempre
  imposta sensorVal a leggi pin analogico (A) 0
  imposta tempC a (sensorVal * 0.048 - 0.5) / 0.01
  imposta tempF a (tempC * 9 / 5 + 32)
  attendi 5 secondi
```

SPRITE

```
quando si preme il tasto c
  dire unione di La temperatura è di e unione di tempC e °C per 2 secondi
```

```
quando si preme il tasto f
  dire unione di La temperatura è di e unione di tempF e °F per 2 secondi
```



CONCETTI:

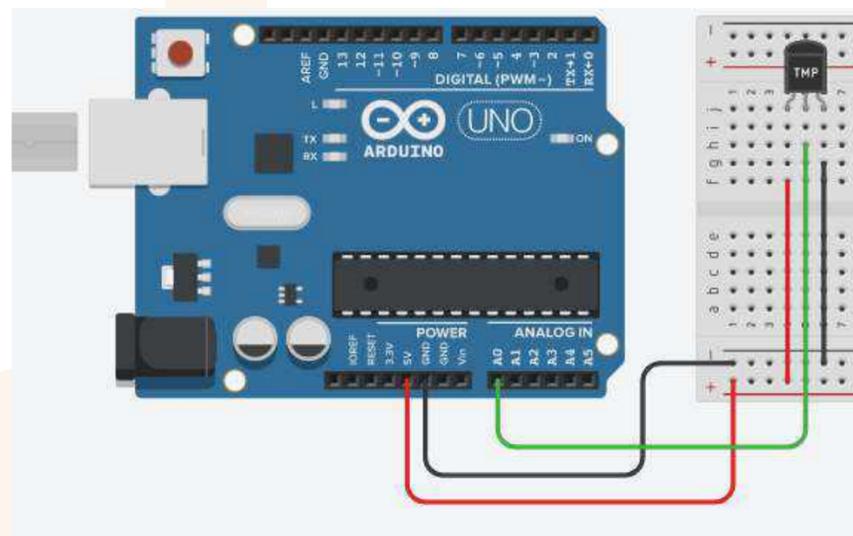
SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

PROGRAMMAZIONE
TESTUALE

TEMPERATURA



sketch_jul03a | Arduino 1.8.19

File Modifica Sketch Strumenti Aiuto



sketch_jul03a \$

```
//pin analogico per il sensore di temperatura
const int tempSensorPin = A0;
//dichiarazione delle variabili
float tempC = 0;
float tempF = 0;
float sensorVal = 0;
```

```
void setup() {
  // apriamo la comunicazione seriale
  Serial.begin(9600);
}
void loop() {
  //salviamo nella variabile il valore letto
  sensorVal = analogRead(tempSensorPin);
  //calcoliamo la temperatura in Celsius
  tempC = (((sensorVal*0.0048)-0.5)/0.01);
  //calcoliamo la temperatura in Fahrenheit
  tempF = (tempC*9)/5+32;
  //stampiamo sul monitor seriale
  Serial.print("Sensor Value: ");
  Serial.print(sensorVal);
  Serial.print(", degrees C: ");
  Serial.print(tempC);
  Serial.print(", degrees F: ");
  Serial.println(tempF);
```

```
//ritardo
delay(500);
```



CONCETTI:

SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

BUZZER

IDEE



POSSIAMO
INSERIRE UN
BUZZER COME
ALLARME?

POSSIAMO INSERIRE UN
ALLARME LUMINOSO?

POSSIAMO PROGRAMMARE
UNA FUNZIONE PER
L'ALLARME?

POSSIAMO IMPOSTARE
UN VALORE DI SOGLIA?



SERVOMOTORI

CONCETTI:

CIRCUITI ELETTRICI

MOVIMENTO
MECCANICO

FUNZIONE DI
MAPPATURA

PROPORZIONI



ANGOLO DI ROTAZIONE DI 180°



SERVOMOTORI

MATERIALI:

- SERVOMOTORE
- SCHEDA ARDUINO UNO
- BREADBOARD

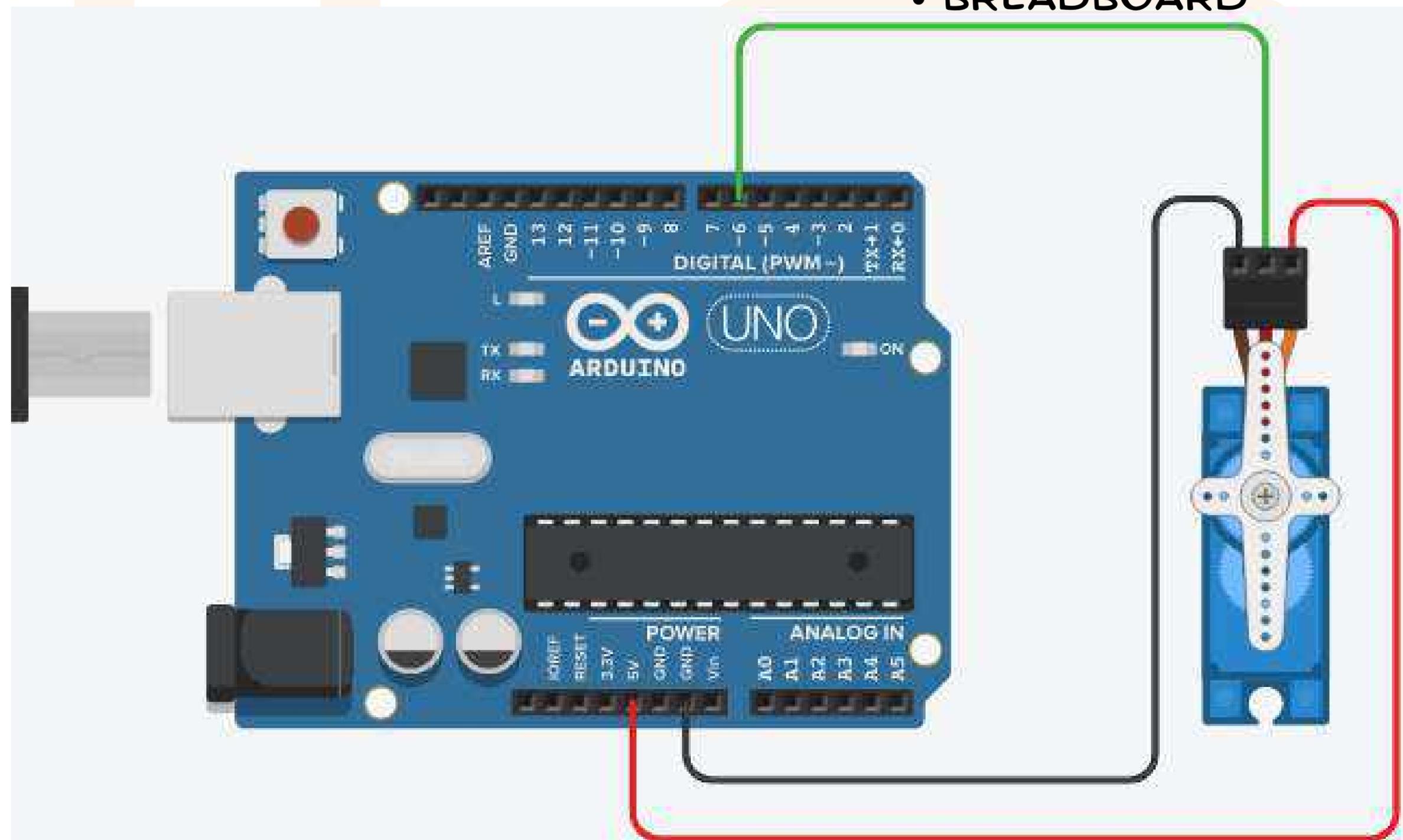
CONCETTI:

CIRCUITI ELETTRICI

MOVIMENTO
MECCANICO

FUNZIONE DI
MAPPATURA

PROPORZIONI





CONCETTI:

CIRCUITI ELETTRICI

MOVIMENTO
MECCANICO

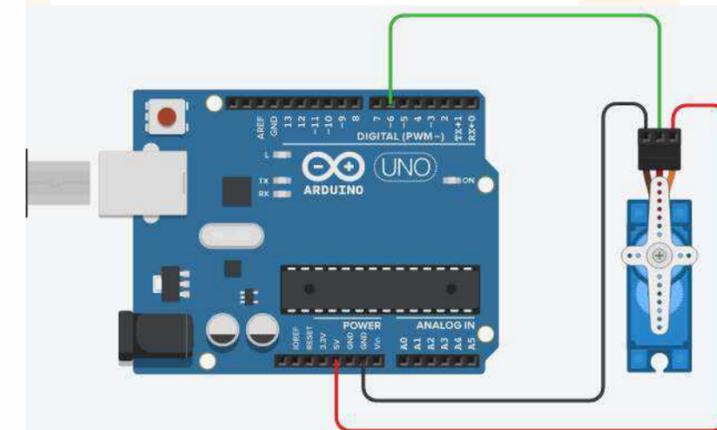
FUNZIONE DI
MAPPATURA

LIBRERIE

PROGRAMMAZIONE
A BLOCCHI

SERVOMOTORI

```
quando si avvia Arduino Uno
imposta angolo servo a 0
impostare l'angolo di servo pin 6 come angolo servo
per sempre
  ripeti 3
    cambia angolo servo di 60
    impostare l'angolo di servo pin 6 come angolo servo
  ripeti 3
    cambia angolo servo di -60
    impostare l'angolo di servo pin 6 come angolo servo
```





CONCETTI:

CIRCUITI ELETTRICI

MOVIMENTO
MECCANICO

FUNZIONE DI
MAPPATURA

LIBRERIE

PROGRAMMAZIONE
TESTUALE

SERVOMOTORI

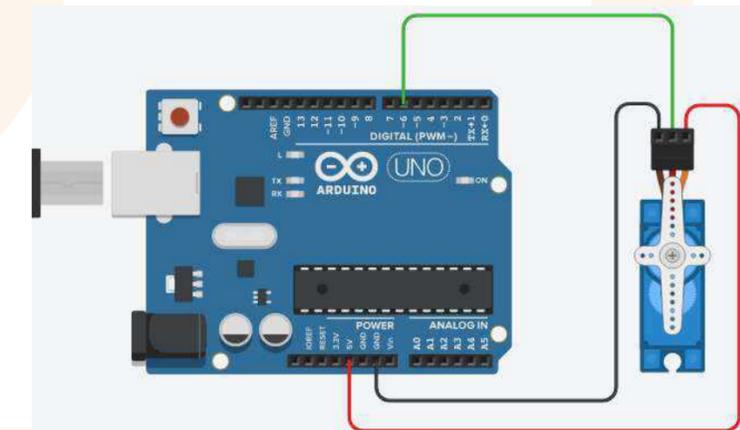
servo | Arduino 1.8.19

File Modifica Sketch Strumenti Aiuto



servo §

```
#include <Servo.h> //Inserire la libreria Servo
Servo Servol; //Il nome del servo è Servol
void setup() {
  Servol.attach (6); //Il Servol è collegato al pin digitale 6
}
void loop() {
  //Imposto la posizione a cui si deve spostare il Servol
  Servol.write (-180);
  delay (500);
  Servol.write (-120);
  delay (500);
  Servol.write (-60);
  delay (500);
  Servol.write (0);
  delay (500);
  Servol.write (60);
  delay (500);
  Servol.write (120);
  delay (500);
  Servol.write (180);
  delay (500);
}
```





SERVOMOTORE E POTENZIOMETRO

MATERIALI:

- SERVOMOTORE
- POTENZIOMETRO
- SCHEDA ARDUINO UNO
- BREADBOARD

CONCETTI:

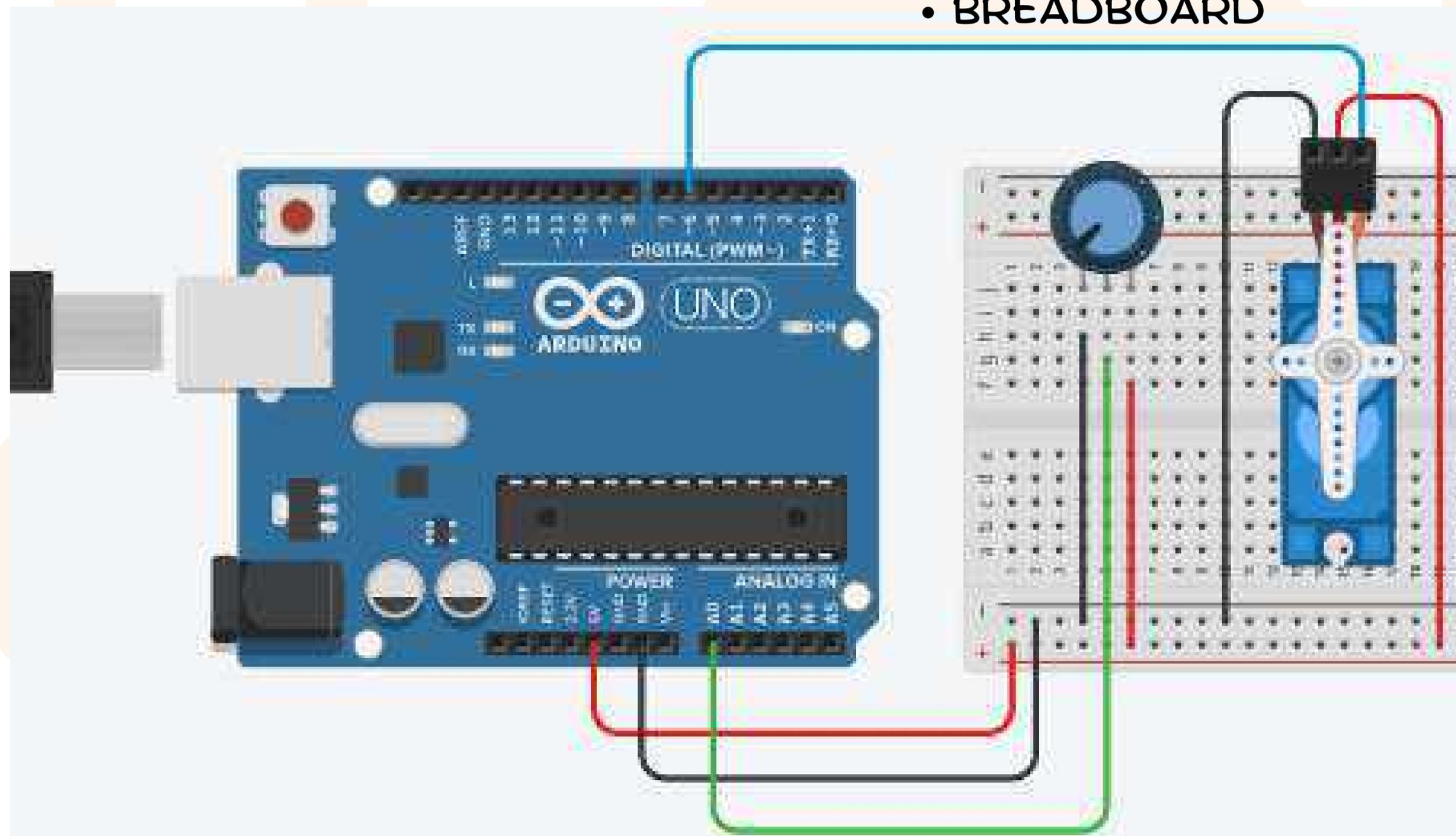
CIRCUITI ELETTRICI

MOVIMENTO
MECCANICO

FUNZIONE DI
MAPPATURA

PROPORZIONI

POTENZIOMETRO





SERVOMOTORE E POTENZIOMETRO

CONCETTI:

MOVIMENTO
MECCANICO

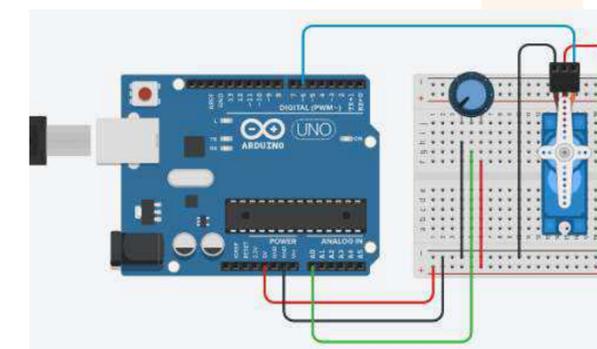
FUNZIONE DI
MAPPATURA

PROPORZIONI

POTENZIOMETRO

PROGRAMMAZIONE
A BLOCCHI

```
quando si avvia Arduino Uno
per sempre:
  imposta Potenzimetro a leggi pin analogico (A) 0
  imposta angolo servo a mappa Potenzimetro da ( 0 , 1023 ) to ( 0 , 179 )
  imposta l'angolo di servo pin 5 come angolo servo
```





CONCETTI:

MOVIMENTO
MECCANICO

FUNZIONE DI
MAPPATURA

PROPORZIONI

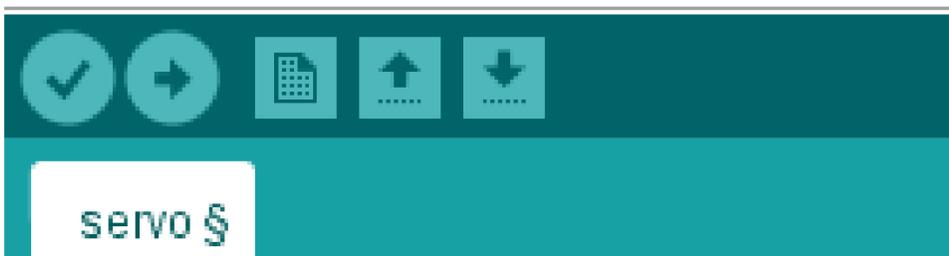
POTENZIOMETRO

PROGRAMMAZIONE
TESTUALE

SERVOMOTORE E POTENZIOMETRO

servo | Arduino 1.8.19

File Modifica Sketch Strumenti Aiuto



```
servo $
```

```
#include <Servo.h>
```

```
Servo myservo;
```

```
int potpin = 0;
```

```
int val;
```

```
void setup()
```

```
{
```

```
  myservo.attach(9);
```

```
}
```

```
void loop()
```

```
{
```

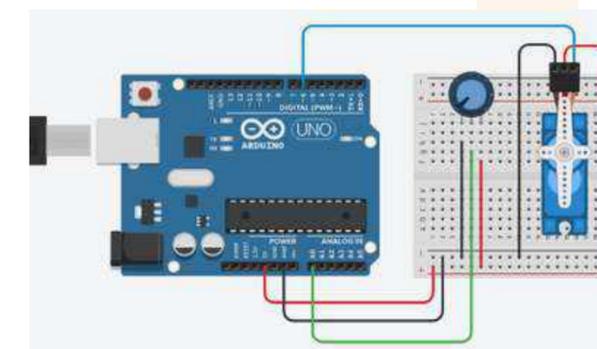
```
  val = analogRead(potpin);
```

```
  val = map(val, 0, 1023, 0, 179);
```

```
  myservo.write(val);
```

```
  delay(15);
```

```
}
```





CONCETTI:

SENSORE DI
TEMPERATURA
LM35

PIN ANALOGICO

EQUAZIONE DI
TARATURA

BUZZER

IDEE



POSSIAMO
USARE PIÙ
SERVOMOTORI?

POSSIAMO CREARE UN
BRACCIO MECCANICO?

E UNA GAUGE PER LA
TEMPERATURA?

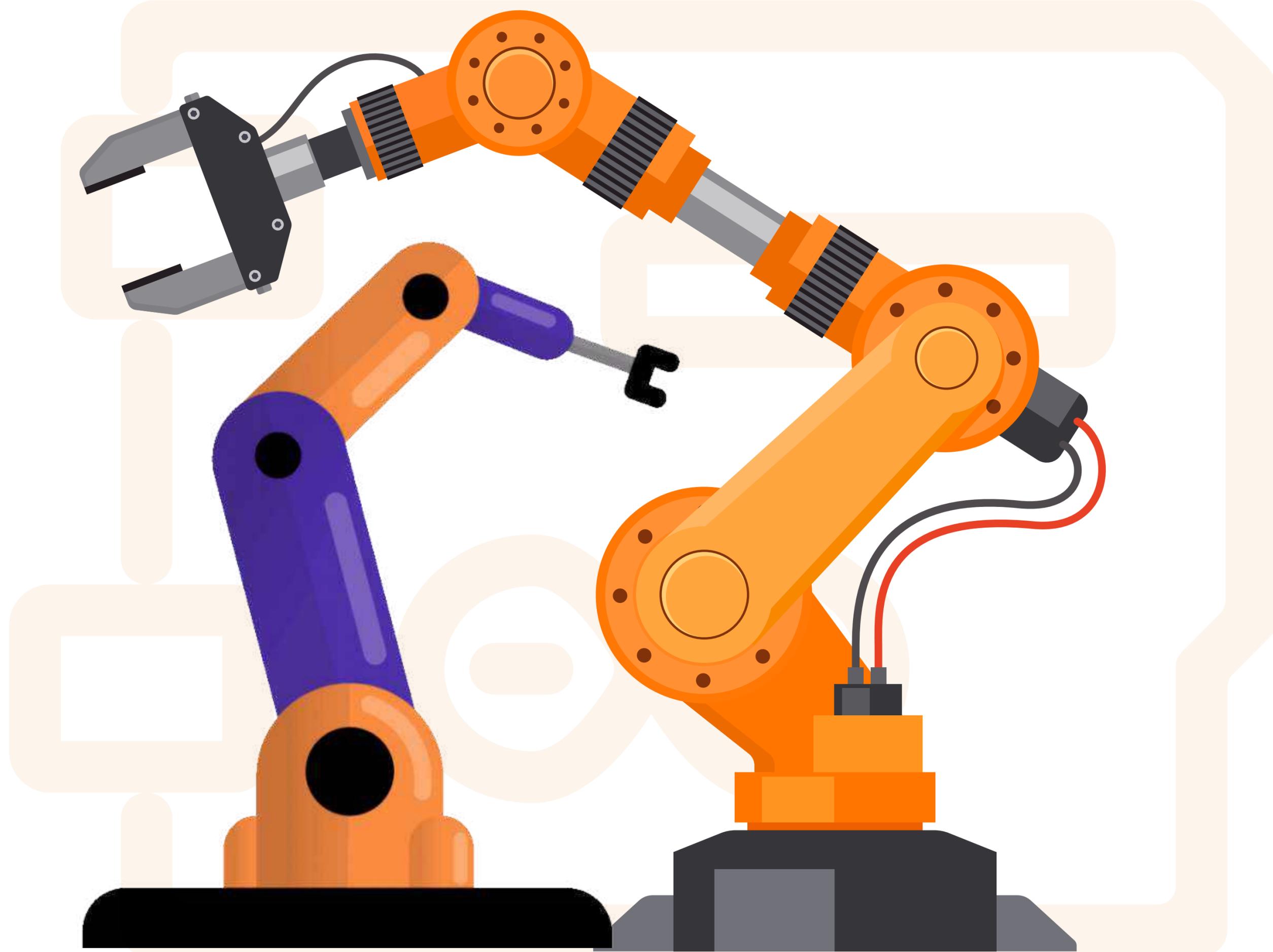
E ANCORA PER LA LUCE
AMBIENTALE?



CONCETTI:

USARE PIÙ
SERVOMOTORI

ALIMENTAZIONE
ESTERNA

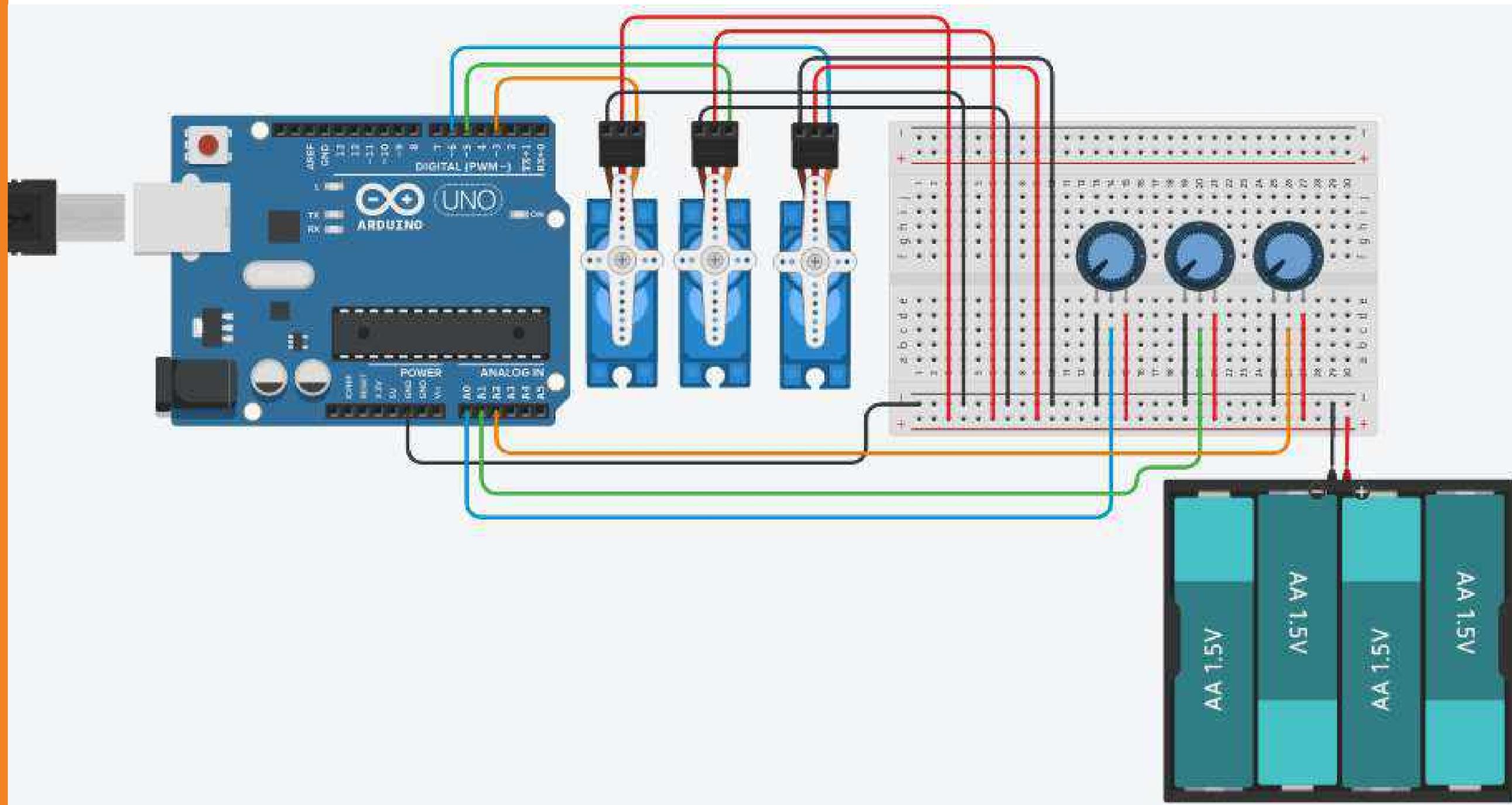




CONCETTI:

USARE PIÙ
SERVOMOTORI

ALIMENTAZIONE
ESTERNA



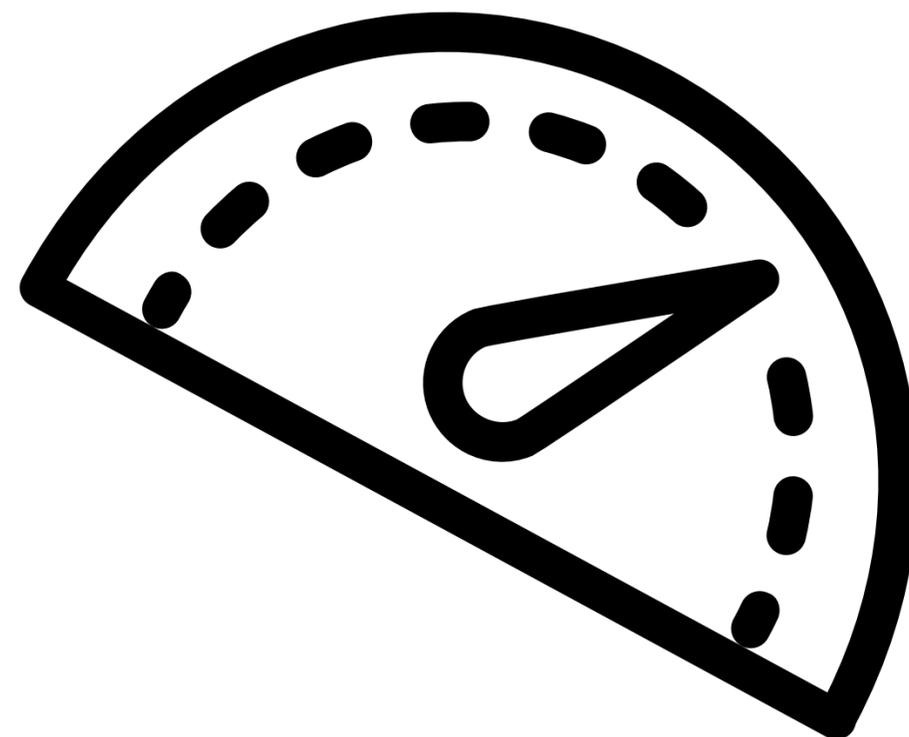
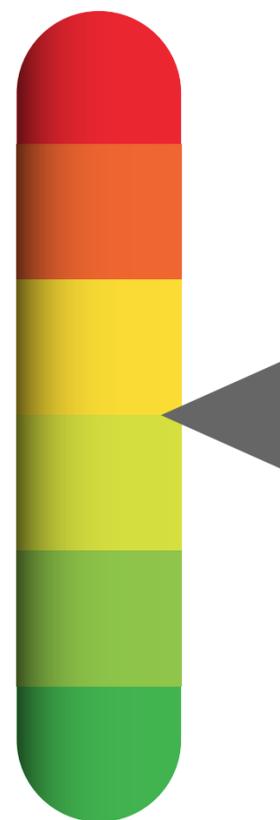
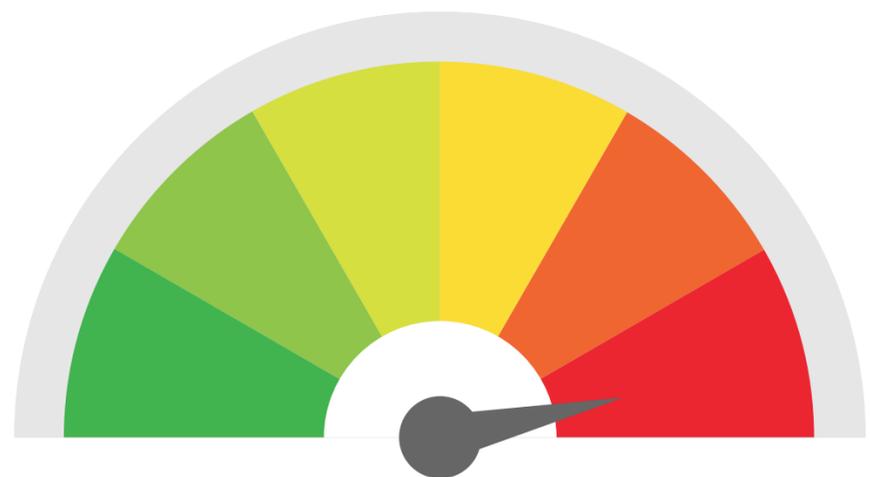


CONCETTI:

MISURARE IL VALORE
DI UN SENSORE

STABILIRE UN RANGE
MIN/MAX

MAPPARE IL VALORE
IN UNA SCALA 0/180



IRRIGAZIONE SERRA



CONCETTI:

SERRA

SISTEMA DI
IRRIGAZIONE

ASTROFISICA

PROGRAMMAZIONE
A BLOCCHI



IRRIGAZIONE SERRA



CONCETTI:

SERRA

SISTEMA DI
IRRIGAZIONE

ASTROFISICA

PROGRAMMAZIONE
A BLOCCHI





CONCETTI:

SVUOTAMENTO DI UN
SERBATORIO

BILANCIO DI ENERGIA

EQUAZIONE DI
BERNOULLI

DEEPEN
UNDERSTANDING



IRRIGAZIONE SERRA

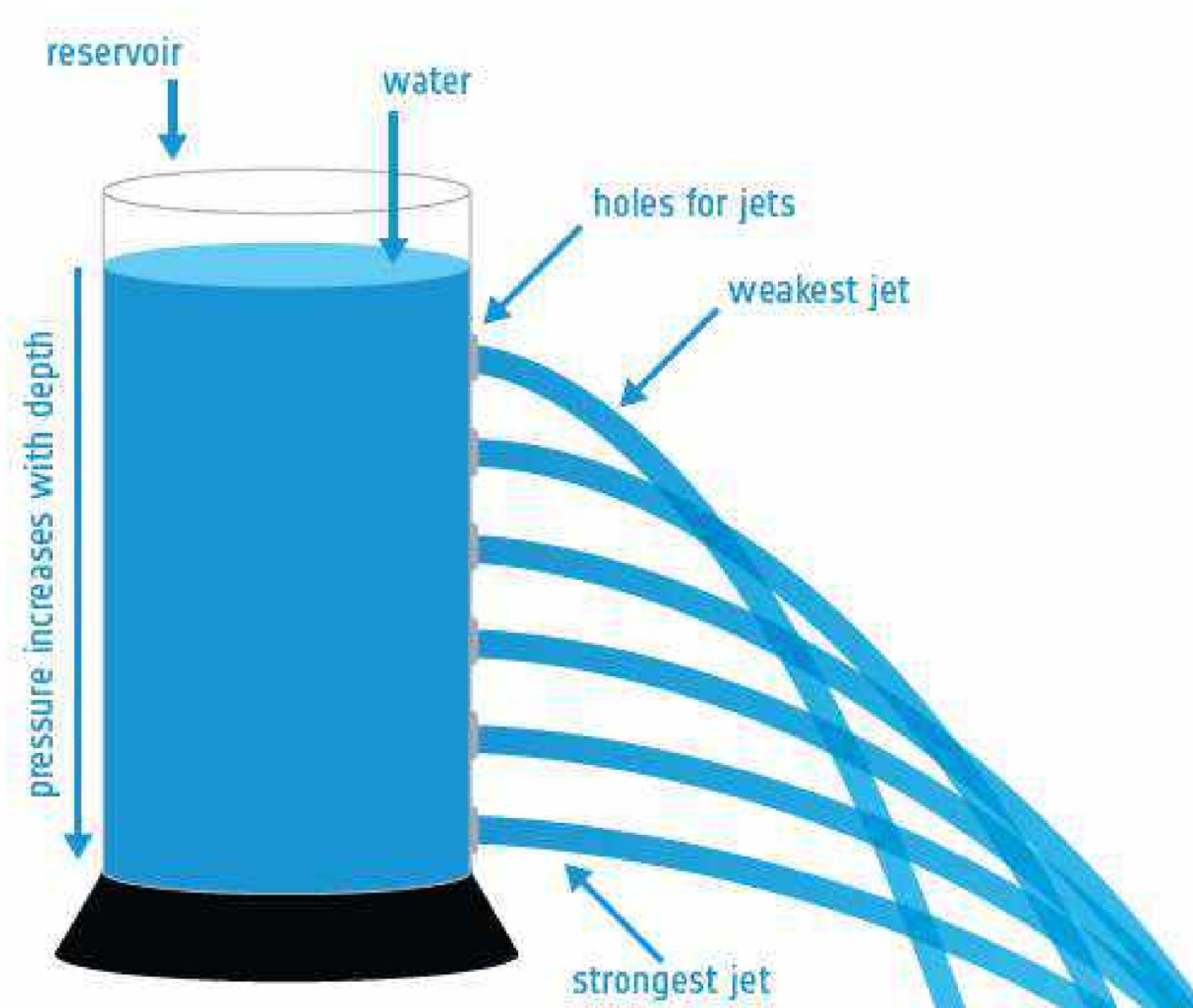
$$p_1 + \frac{1}{2} \rho v_1^2 + \rho g h_1 = p_2 + \frac{1}{2} \rho v_2^2 + \rho g h_2$$

CONCETTI:

SVUOTAMENTO DI UN
SERBATORIO

BILANCIO DI ENERGIA

EQUAZIONE DI
BERNOULLI



IRRIGAZIONE SERRA

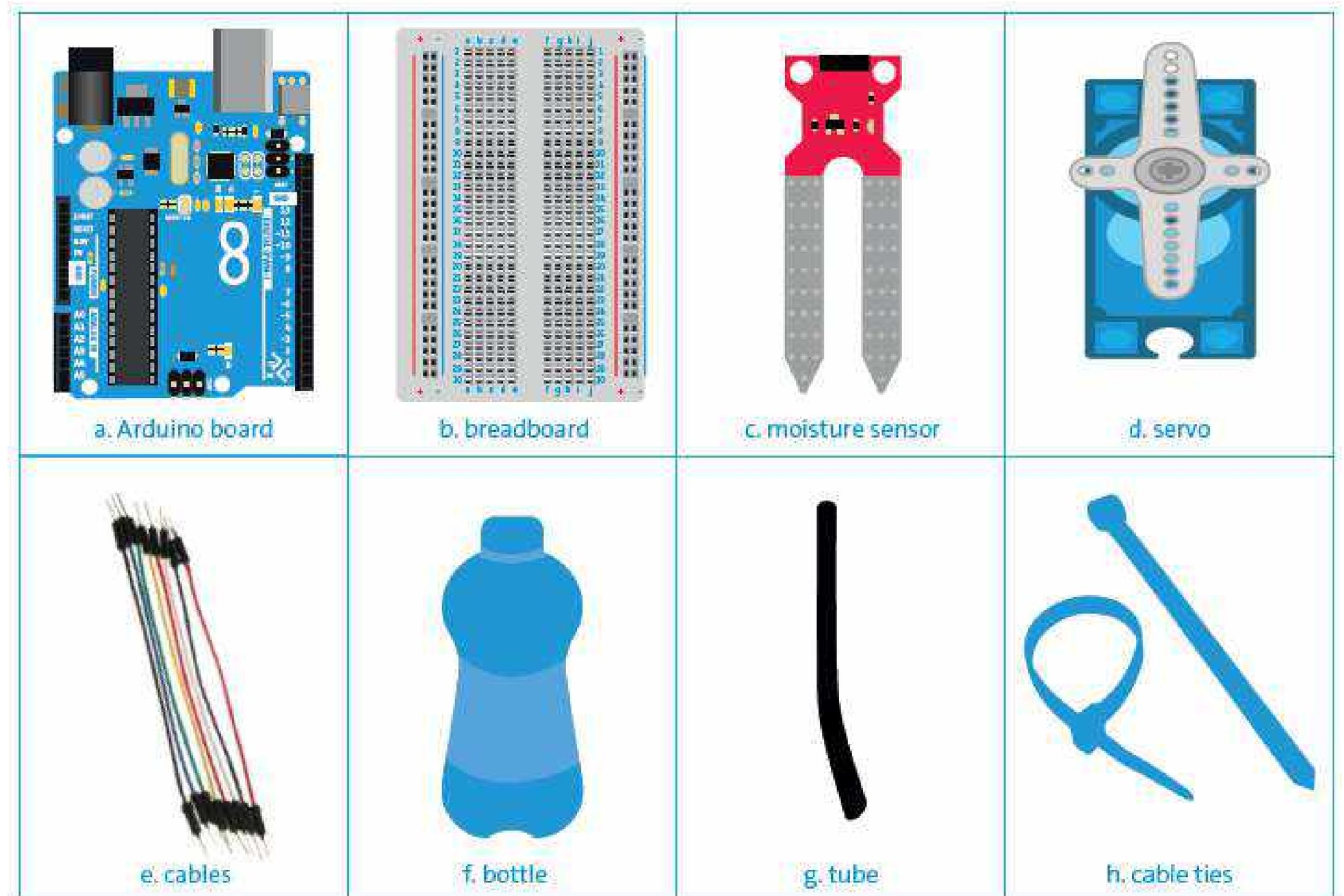


CONCETTI:

SVUOTAMENTO DI UN
SERBATORIO

BILANCIO DI ENERGIA

EQUAZIONE DI
BERNOULLI





IRRIGAZIONE SERRA

CONCETTI:

SVUOTAMENTO DI UN
SERBATORIO

BILANCIO DI ENERGIA

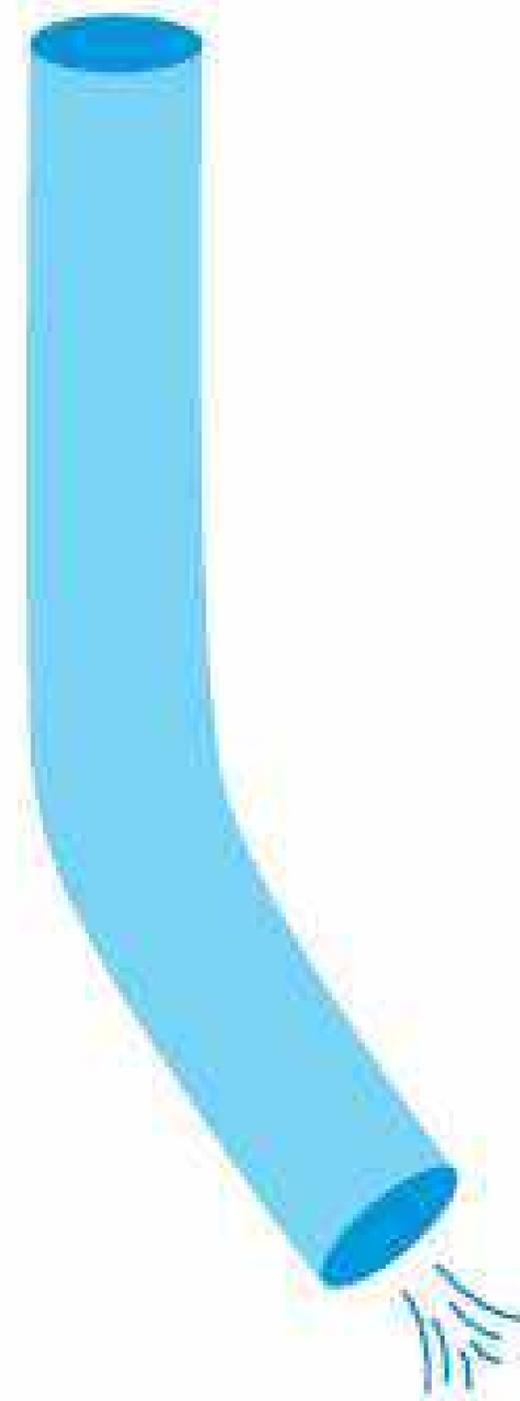
EQUAZIONE DI
BERNOULLI



off



on



IRRIGAZIONE SERRA



CONCETTI:

SVUOTAMENTO DI UN
SERBATORIO

BILANCIO DI ENERGIA

EQUAZIONE DI
BERNOULLI



IRRIGAZIONE SERRA



CONCETTI:

SVUOTAMENTO DI UN
SERBATORIO

BILANCIO DI ENERGIA

EQUAZIONE DI
BERNOULLI



IRRIGAZIONE SERRA



MITICI



IRRIGAZIONE SERRA



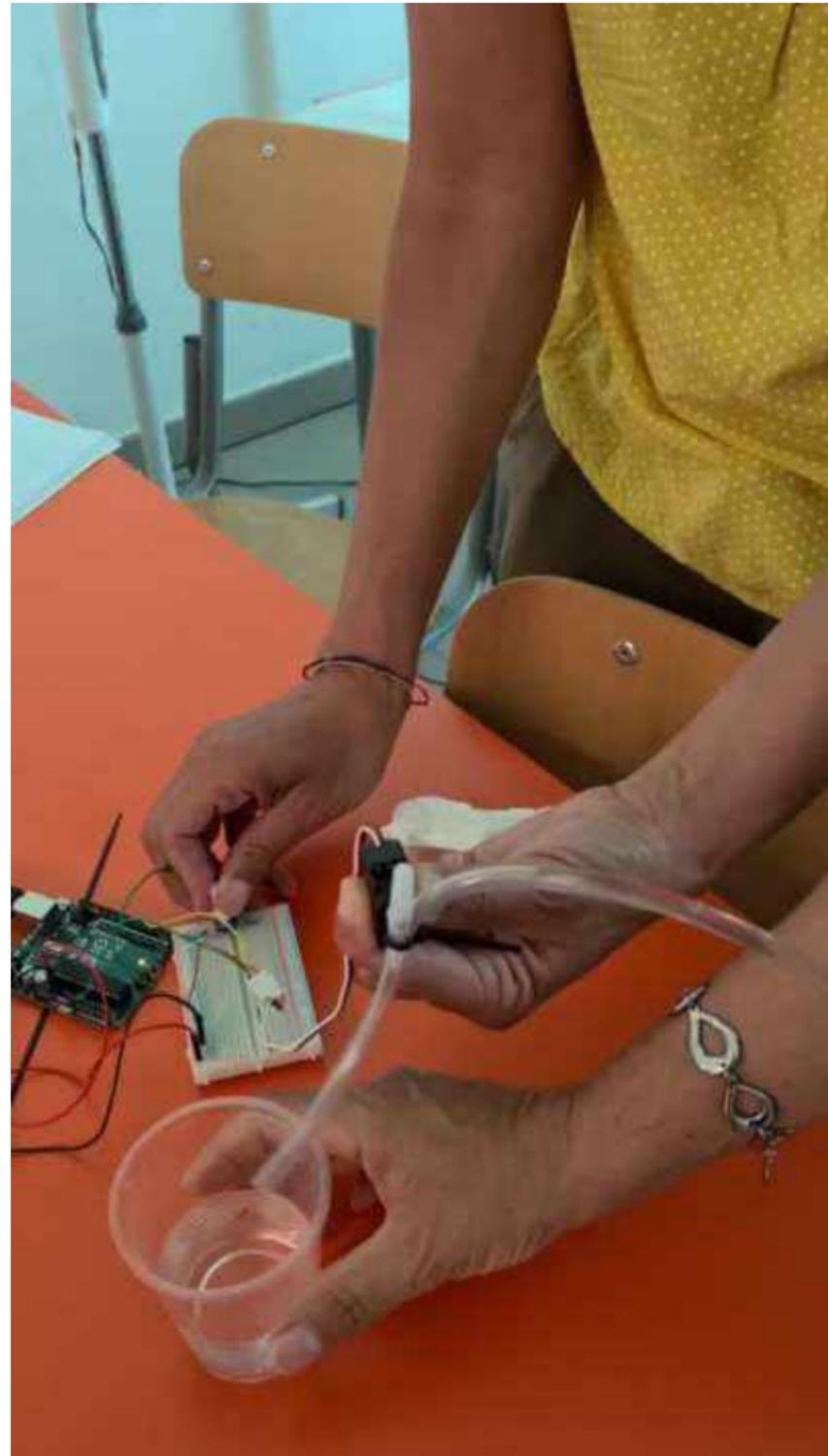
MITICI



IRRIGAZIONE SERRA



MITICI



IRRIGAZIONE SERRA



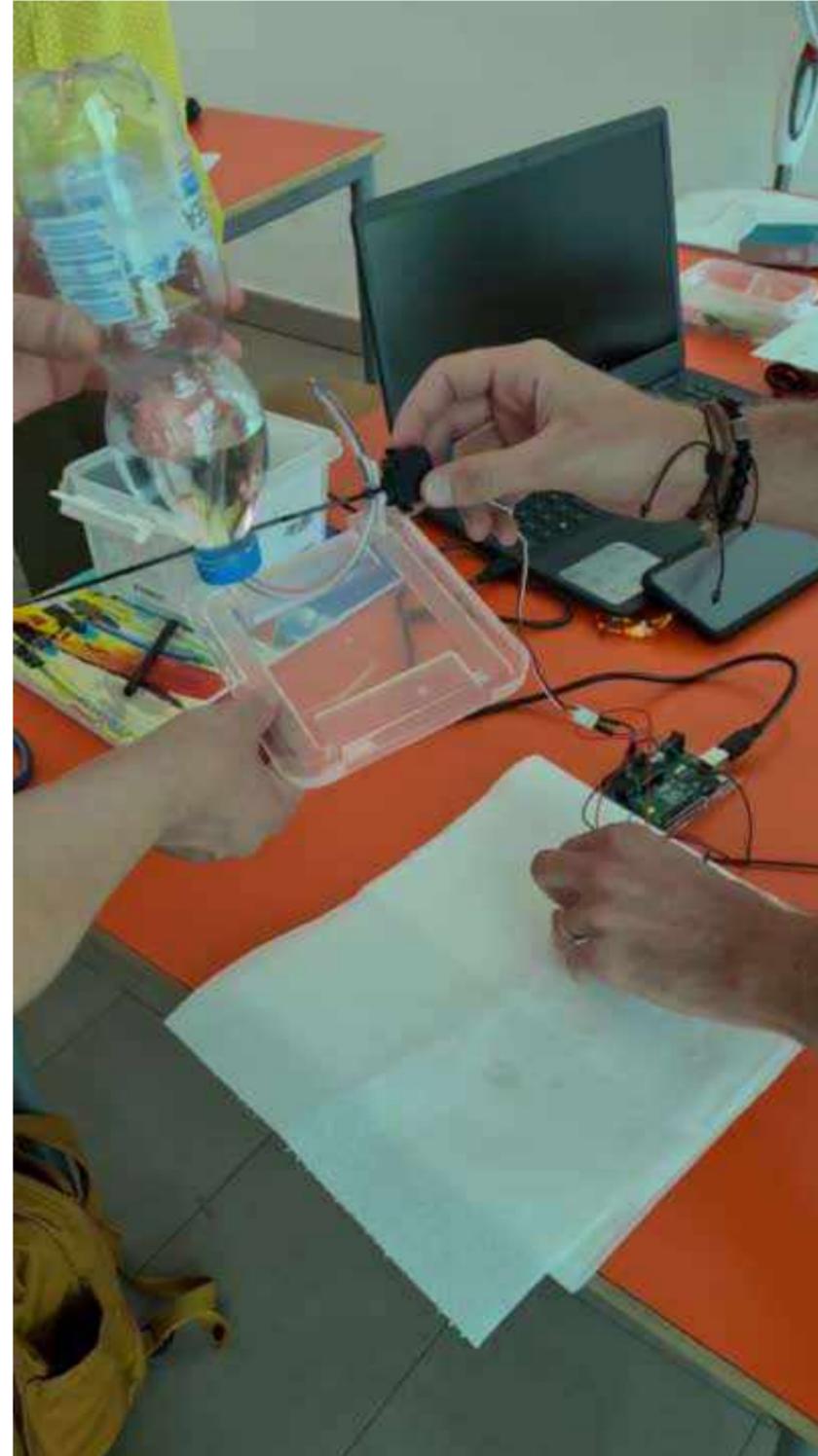
MITICI



IRRIGAZIONE SERRA



MITICI





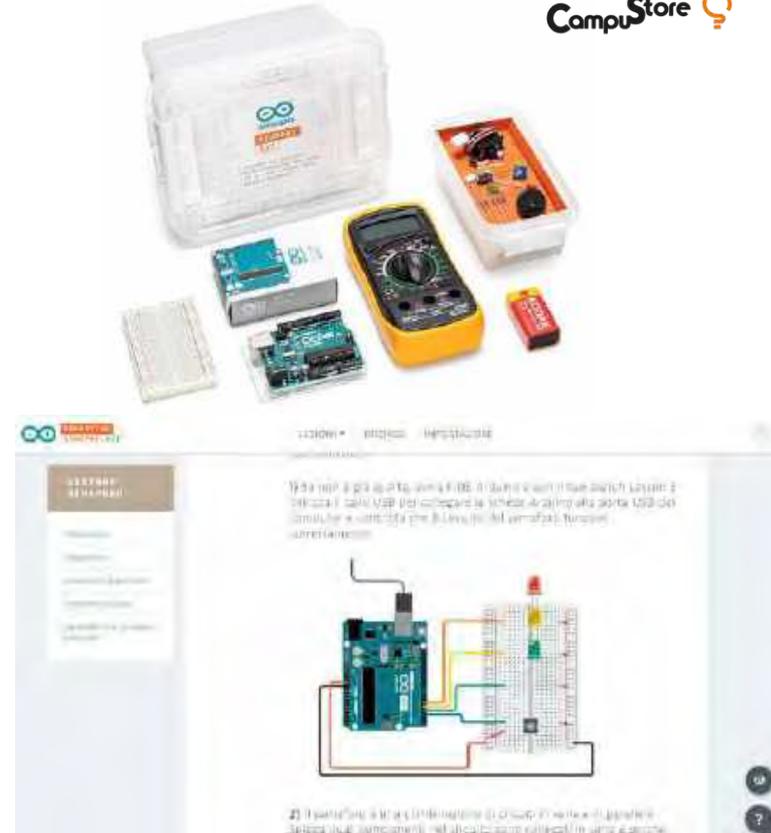
BENVENUTI

**Arduino Student Kit
Sperimentazione didattica**

COS'E' LO STUDENT KIT?

- Kit pensato per studenti **dagli 11 anni in su**
- **Hardware** (scheda Arduino Uno, multimetro, componenti, ...)
- **Piattaforma online** (11 Lezioni per un totale di circa 25 ore di lezione, con istruzioni passo-passo, note per gli insegnanti e studenti, quaderno degli esercizi e risorse varie)

<https://studentkit.arduino.cc/>



DOVE TROVARE LO STUDENT KIT



Arduino Student Kit

Codice: 331642

Codice MEPA: 331642CS

x12



**Arduino Student Kit
Kit per mezza classe**

Codice: 333212

Codice MEPA: 333212CS

x24



**Arduino Student Kit
Kit per la classe**

Codice: 333211

Codice MEPA: 333211CS

LEZIONE 10 - PROGETTO-APERTO: PROGRAMMA, COSTRUISCI E PROGETTA LA SERRA INTELLIGENTE

Ora conosciamo molti componenti elettronici e funzioni di programmazione, è tempo di usare di nuovo la creatività e progettare la propria serra con il nuovo Arduino Student Kit.

22|09|2020



STUDENT KIT SYLLABUS

LEZIONE 10: SERRA INTELLIGENTE - PROGETTO APERTO

DURATA: 180 min

QUADERNO DEGLI ESERCIZI: pag.26-30

INVENZIONI IMPORTANTI: Oltre l'atmosfera terrestre

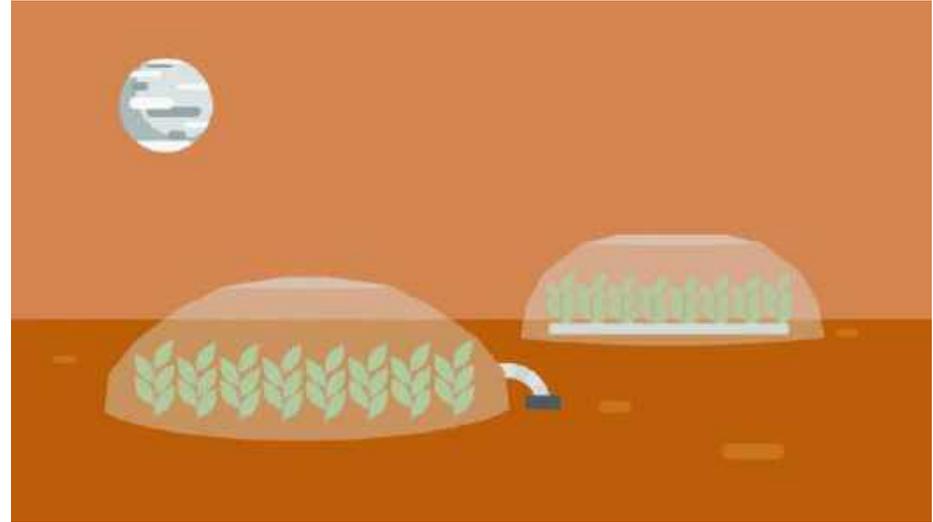
Descrizione: Un progetto aperto per ideare, costruire e programmare un sistema di controllo del clima per una serra.

Impareremo a conoscere: Progettazione e costruzione di circuiti, riutilizzo del codice esistente, debugging, creatività, pensiero critico e problem solving

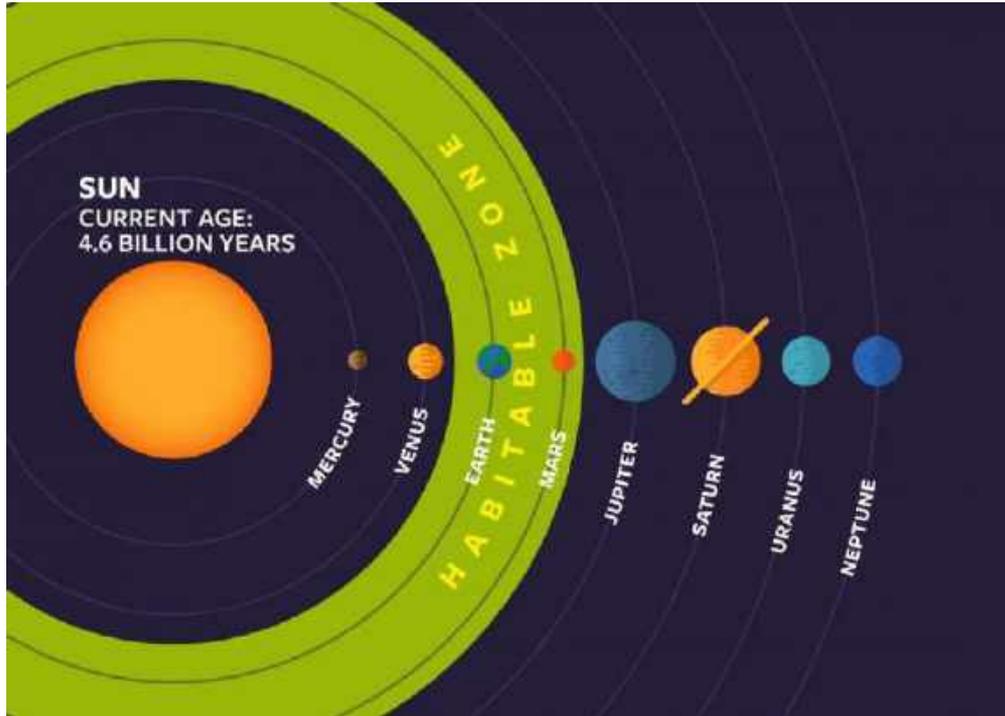
Per vedere il programma completo clicca [qui](#)

CHE COSA È UN PROGETTO APERTO?

Un **progetto aperto** significa che spetta a noi decidere come funziona il nostro circuito della serra intelligente. Useremo i concetti che abbiamo imparato nelle lezioni precedenti per completare il progetto, ma il risultato potrebbe cambiare.



PERCHE' LA SERRA?

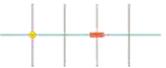
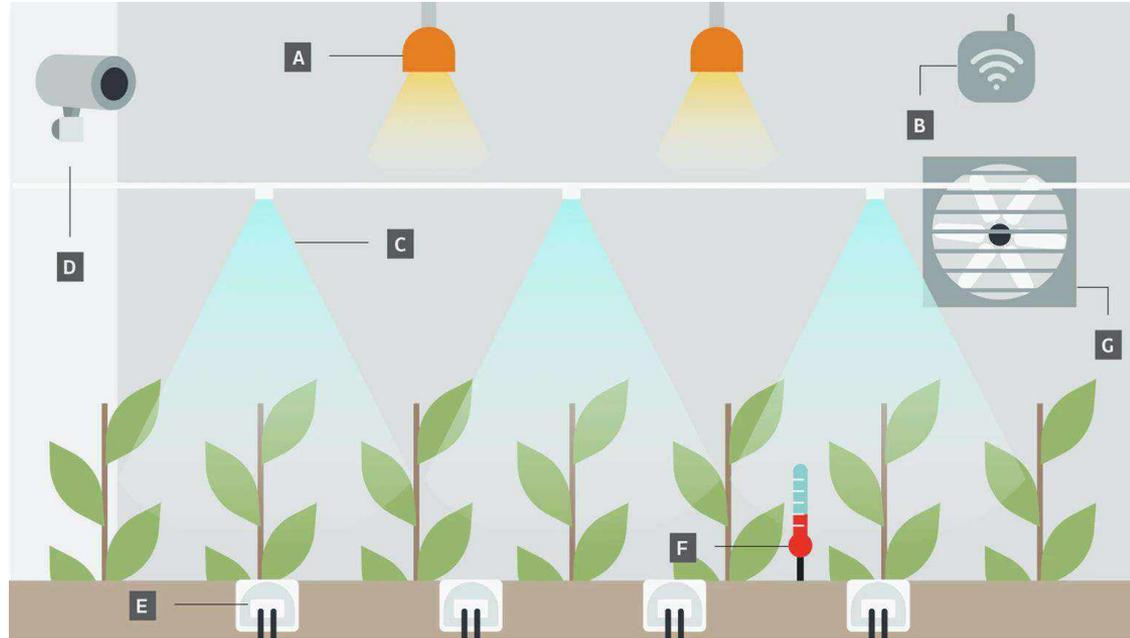


Courtesy of ESA



COS'E' UNA SERRA?

- A. Controllo della **luce**.
- A. Controllo a **distanza**.
- A. **Irrigazione** automatizzata.
- A. Sistemi di **allarme**.
- A. **Controllo dell'umidità** del terreno.
- A. Controllo della **temperatura**.
- A. **Aerazione**.



LINEE GUIDA PER IL PROGETTO: OBIETTIVI + CRITERI

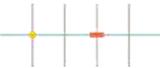
- Ci sono **sette obiettivi** per questo progetto, che bisogna raggiungere.
- Per soddisfare questi obiettivi, dobbiamo essere **creativi** e pensare in modo critico ai problemi che affronteremo.
- Teniamo a mente i **criteri** e i **parametri** del progetto. I criteri e i parametri sono **schemi precisi** che debbono essere seguiti affinché il progetto abbia successo.
- **Creatività e Rigore**

7 OBIETTIVI DEL PROGETTO

1. **Idee per il progetto** - raccogliere idee per la progettazione e la programmazione della vostra serra
2. **Prototipo del circuito** - creare un circuito che controlla il clima della serra e rispetti tutti i criteri richiesti.
3. **Schema Elettrico** - disegnare uno schema elettrico che corrisponda al design del progetto della serra.
4. **Pseudocodice** - scrivere lo pseudocodice del nostro programma
5. **Programma** - creare uno sketch con Arduino IDE partendo dallo pseudocodice
6. **Modello serra** - Costruire un prototipo fisico della serra con vari materiali forniti o raccolti
7. **Dimostrazione** - sviluppare una breve presentazione per spiegare il design del circuito.

CRITERI E PARAMETRI

- A volte, aiuta pensare ai **criteri** e ai parametri come una lista di ciò che si deve o non si deve fare per il progetto.
- I **parametri** sono dei fattori limitanti ai quali il progetto deve attenersi.



I CRITERI DEL PROGETTO CI AIUTANO A IMPOSTARLO

Il nostro progetto dovrebbe contenere:

- **LED** – Almeno due LED. Ogni LED utilizzato dovrebbe essere collegato in serie con una resistenza. Sta a noi decidere come usare i LED.
- **Controlli manuali**– Ad esempio includere interruttori e pulsanti, potenziometri o la possibilità di inserire un comando attraverso il monitor seriale.
- **Servomotore e condensatore** – Il nostro sistema di controllo del clima dovrebbe incorporare un servomotore che esegue un compito specifico mentre si muove.
- **Pin digitali e analogici** – il circuito della serra deve includere l'uso di almeno un pin digitale ed uno analogico sulla scheda Arduino.
- **Resistenza** – Ogni LED nel circuito deve avere delle resistenze incluse nella serie. Pulsanti devono essere collegati con una resistenza pull-down.
- **Sensori** – Il circuito dovrebbe includere almeno due sensori che monitorano l'ambiente nella serra - un sensore di luce e un sensore di temperatura.
- **Piezo** – Il piezo dovrebbe suonare un allarme per indicare un problema con la serra o una specifica azione
- **Richiamare una funzione** – Lo sketch che controlla il sistema della serra deve includere almeno una funzione.

OPERAZIONI UNITARIE

PROCESSO

- Una **sequenza di operazioni** che attraverso cambiamenti porta ai risultati progettati

DIAGRAMMI DI FLUSSO

- **Schematizzazione** delle operazioni, dei flussi di dati e delle informazioni con attribuzioni, istruzioni e risultati

OPERAZIONI UNITARIE

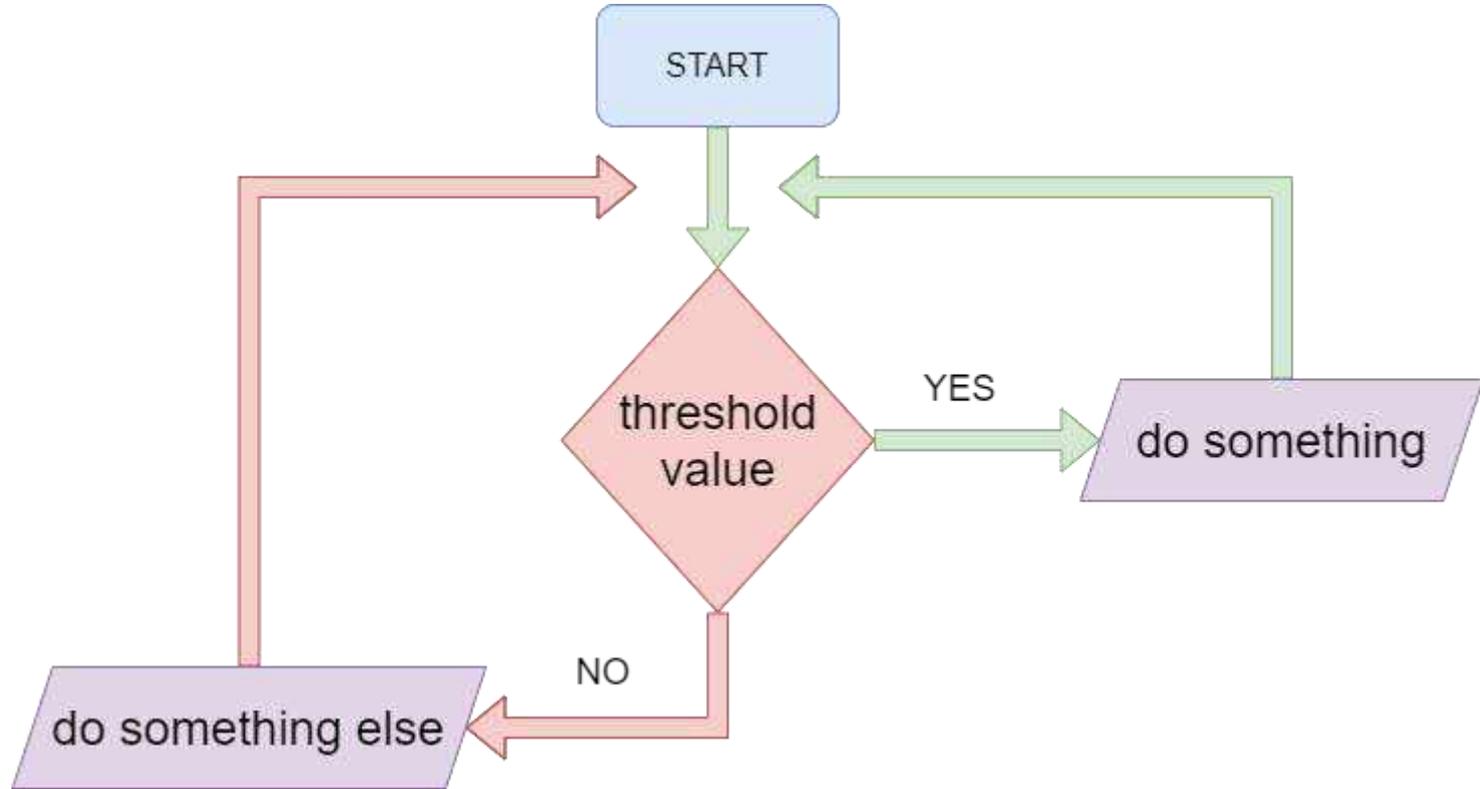
- Ciascuno **stadio** che può essere studiato singolarmente e i cui principi sono indipendenti dal caso specifico

OPERAZIONI UNITARIE

In breve, quello che dobbiamo fare nella nostra serra è:

1. **Controllare la luce:** nel caso di luce bassa accenderla.
1. **Controllare l'umidità:** nel caso di umidità sotto soglia, innaffiare.
1. **Controllare la temperatura:** nel caso avviare un allarme e aumentare l'aerazione.
1. **Controllare il gas:** nel caso avviare un allarme e aumentare l'aerazione.

Sembrano tanti processi, vero?



OPERAZIONI UNITARIE



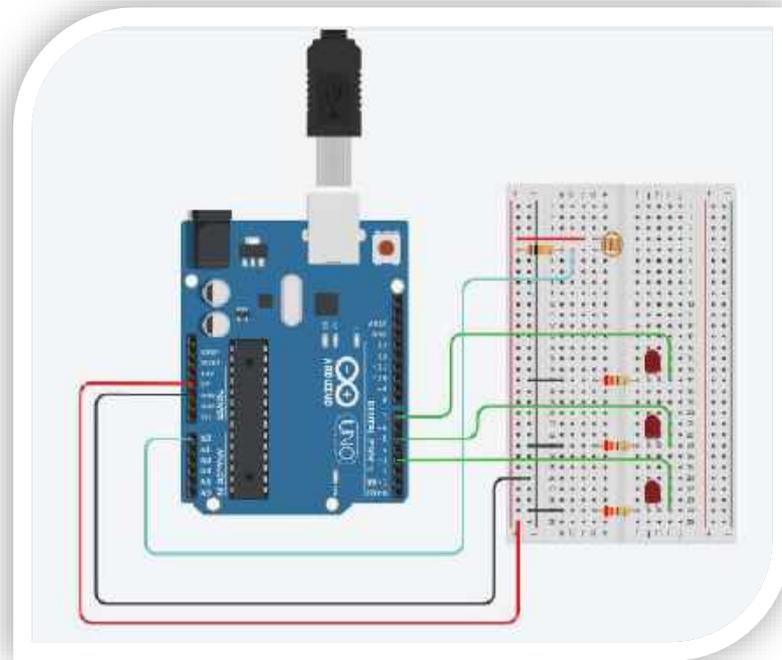
Non sono tre **processi** diversi.
Siamo davanti allo stesso **processo!**

ILLUMINAZIONE

Cosa succede se non ho la giusta **illuminazione**?

Se la
luce è
bassa

- Segnale di allarme
- Accendi LED



ILLUMINAZIONE

1. Definiamo variabili e costanti

1. Definiamo i LED come OUTPUT

1. Inizializziamo il monitor seriale

1. Leggiamo il valore di luce

1. Se siamo sotto soglia

1. Se siamo sopra soglia

1. Stampa

```
Arduino IDE | Arduino 1.8.12
File Modifica Sketch Strumenti Aiuto

fototransistor

#define PIN_SENSOR 10
#define PIN_LED1 13
#define PIN_LED2 12
#define PIN_LED3 11

const int sensorPin = 10;
#define LED1 13
#define LED2 12
#define LED3 11

//Variabile che misura la quantità di luce
int lightAmount = 0;

void setup() {
  // si definiscono i LED come OUTPUT
  pinMode (LED1, OUTPUT);
  pinMode (LED2, OUTPUT);
  pinMode (LED3, OUTPUT);

  Serial.begin (9600);
}

void loop() {
  //nella variabile salviamo il valore letto dal fototransistor
  lightAmount = analogRead(sensorPin);

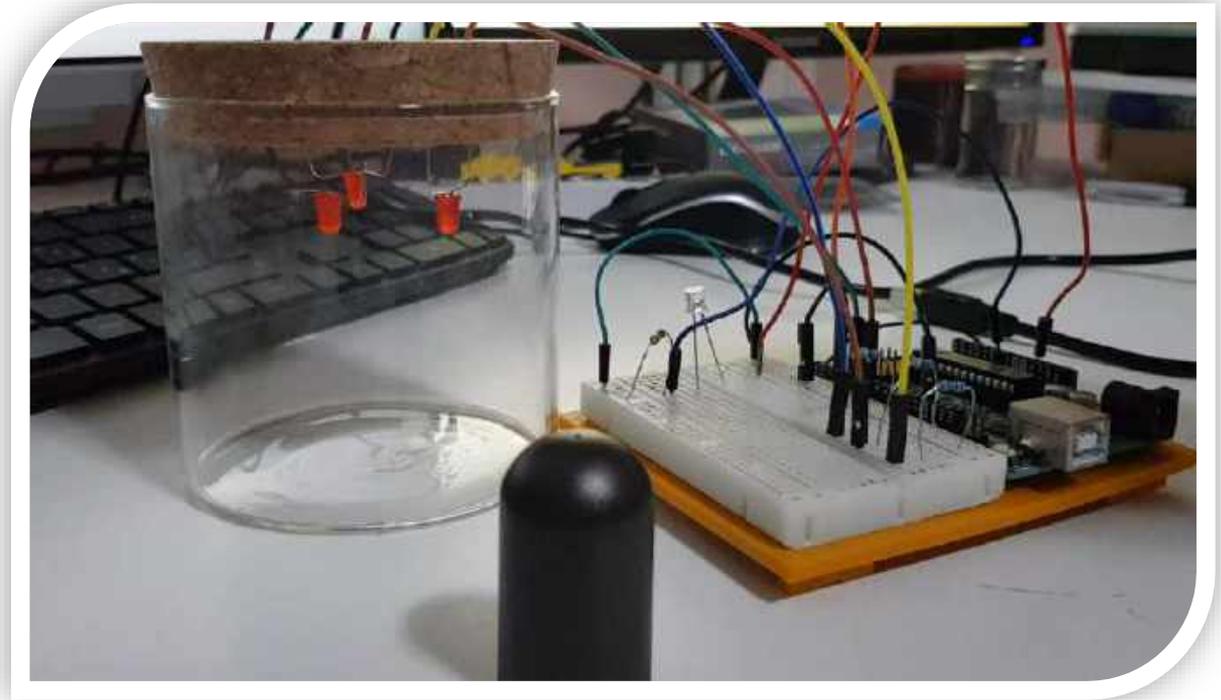
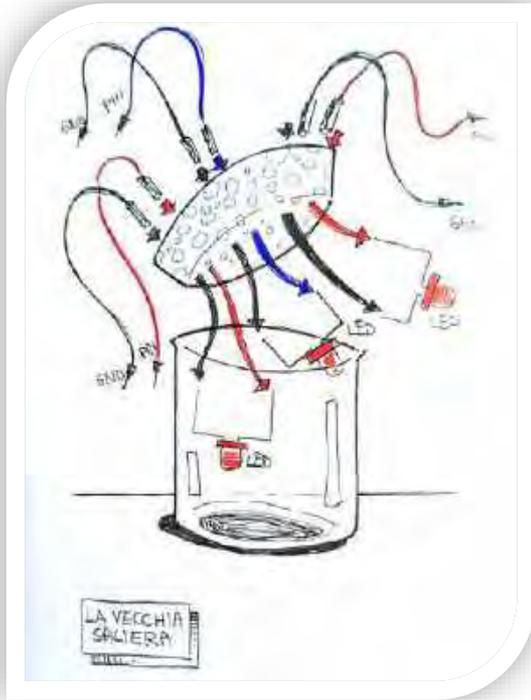
  //ciclo condizionato che accende i LED se la luce è troppo bassa
  if (lightAmount < 30) {
    //Nello schema si bloccò questo con un TRI QUALCOSA
    digitalWrite (LED1, HIGH);
    digitalWrite (LED2, HIGH);
    digitalWrite (LED3, HIGH);
  }
  else {
    //Nello schema si bloccò questo con un TRI QUALCOSA/ALTRIO
    digitalWrite (LED1, LOW);
    digitalWrite (LED2, LOW);
    digitalWrite (LED3, LOW);
  }

  //STAMPA sul seriale il valore di luminosità
  Serial.println ("L'intensità è pari a: ");
  Serial.println (lightAmount);

  delay (500);
}
```

Per visualizzare lo sketch clicca [qui](#)

ILLUMINAZIONE

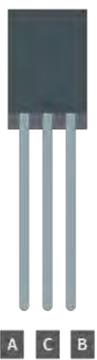


SENSORE DI TEMPERATURA

Il sensore di temperatura è un **sensore analogico**. Ha tre pin - uno per l'alimentazione (A), uno per la terra (B) e uno per l'uscita (C) del segnale analogico alla scheda Arduino. Non esiste un simbolo schematico standard per un sensore di temperatura.

Dato che leggeremo da un sensore analogico, useremo il comando **analogRead()** nell'IDE di Arduino, che invia un valore analogico compreso tra 0 e 1.023 in base alla temperatura della stanza.

Per convertire il **valore in temperatura utilizziamo una formula**. Questa si basa sulle informazioni fornite nella scheda tecnica del sensore di temperatura (un manuale che descrive il sensore e il suo funzionamento). Questa formula converte il campo analogico da 0 a 1.023 nel campo del sensore di temperatura, che è da -40°C a +125°C.


$$tempC = (((sensorVal/1024) * 5) - .5) * 100;$$

Temperatura
in Celsius

Valore di analogRead()
0 to 1023

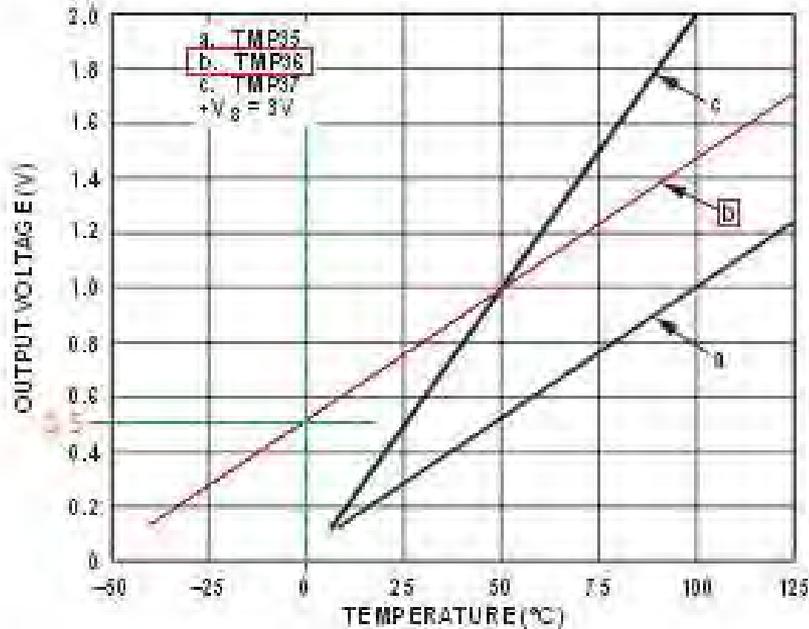


Figure 6. Output Voltage vs. Temperature

DATI

1. 0,5 mV corrisponde a 0 °C;
1. 1 °C di variazione corrisponde a 10 mV;

$$^{\circ}\text{C} = ((\text{ValoreLetto} * \text{PrecisioneADC}) - \text{Tens } 0^{\circ}\text{C}) / \text{stepTens}$$

ValoreLetto = analogRead()

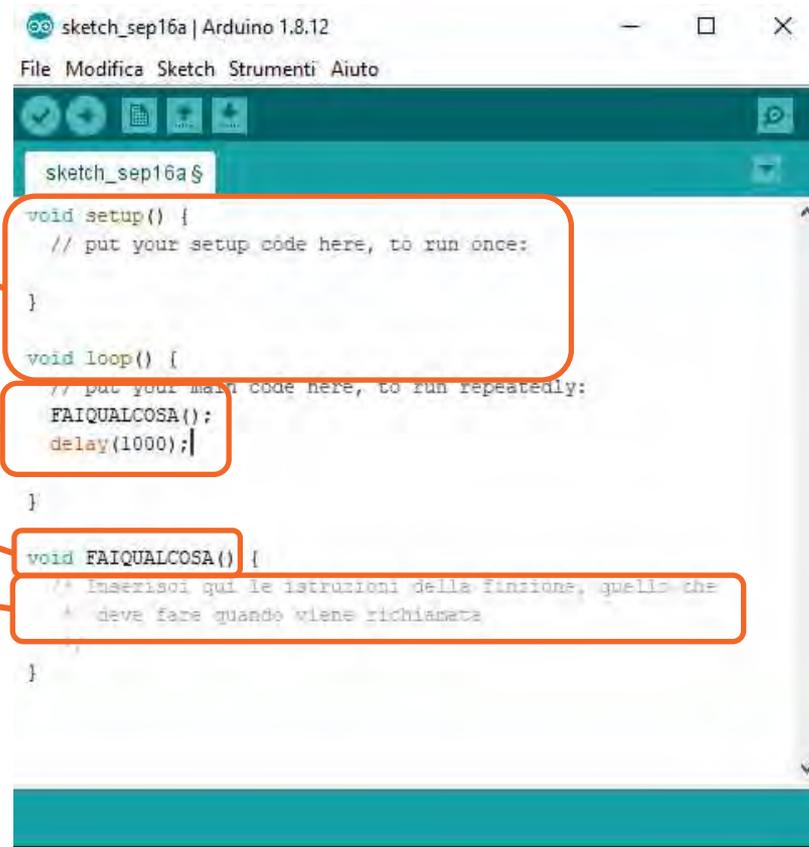
PrecisioneADC = $5\text{V} / 1024\text{V} = 0.0048$

Tens 0°C = Tensione in uscita a 0 °C = 0,5 mV

stepTens = salto di tensione per ogni °C = 10 mV (0,01 V)

CREIAMO UNA NUOVA FUNZIONE

1. Quelle indicate sono già funzioni
1. Qui richiamiamo la nuova funzione semplicemente nominandola
1. Scriviamo VOID NOME() {...}
1. All'interno delle parentesi graffe inseriamo le istruzioni che la nuova funzione deve eseguire quando viene richiamata



```
sketch_sep16a | Arduino 1.8.12
File Modifica Sketch Strumenti Aiuto

sketch_sep16a$

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

void loop() {
  // put your main code here, to run repeatedly:
  FAIQUALCOSA();
  delay(1000);
}

void FAIQUALCOSA() {
  // Inserisci qui le istruzioni della funzione, quello che
  // deve fare quando viene richiamata
}

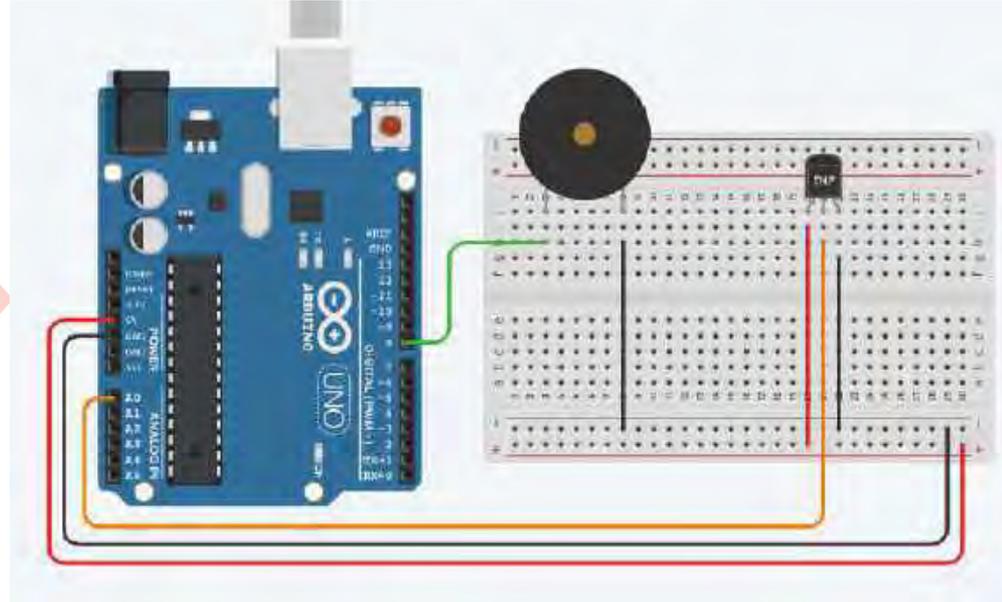

```

SCHEMA DI COLLEGAMENTO

Cosa succede se non ho la giusta **temperatura**?

Se la
Temp.
è alta

- Segnale di allarme
- Aerazione manuale



TEMPERATURA

1. Definiamo variabili e costanti
1. Definiamo il buzzer come OUTPUT e Inizializziamo il monitor seriale
1. Leggiamo il valore di Temperatura
1. Conversioni
1. Stampa sul Monitor Seriale
1. Verifica sul valore di soglia e richiamo funzione Allarme
1. Funzione Allarme

```
Alarm_Temp3
File Modifica Sketch Strumenti Aiuto

// make the Arduino Pin
const int tempSensorPin = A0;
const int buzzerPin = 8;

// declare variables
float tempC = 0;
float tempF = 0;
float sensorVal = 0;

void setup() {
  // open a serial connection to display values
  pinMode (buzzerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // read the analog value from the sensor and store it in a variable
  sensorVal = analogRead(tempSensorPin);

  tempC = ((sensorVal*0.0048) - 0.5) / 0.01; // calculates the temperature in Celsius
  tempF = (tempC * 9) / 5 + 32; // calculates the temperature in Fahrenheit

  Serial.println("Sensor Value: ");
  Serial.println(sensorVal);
  Serial.println(" degrees C: ");
  Serial.println(tempC);
  Serial.println(" degrees F: ");
  Serial.println(tempF);
  delay(2000);

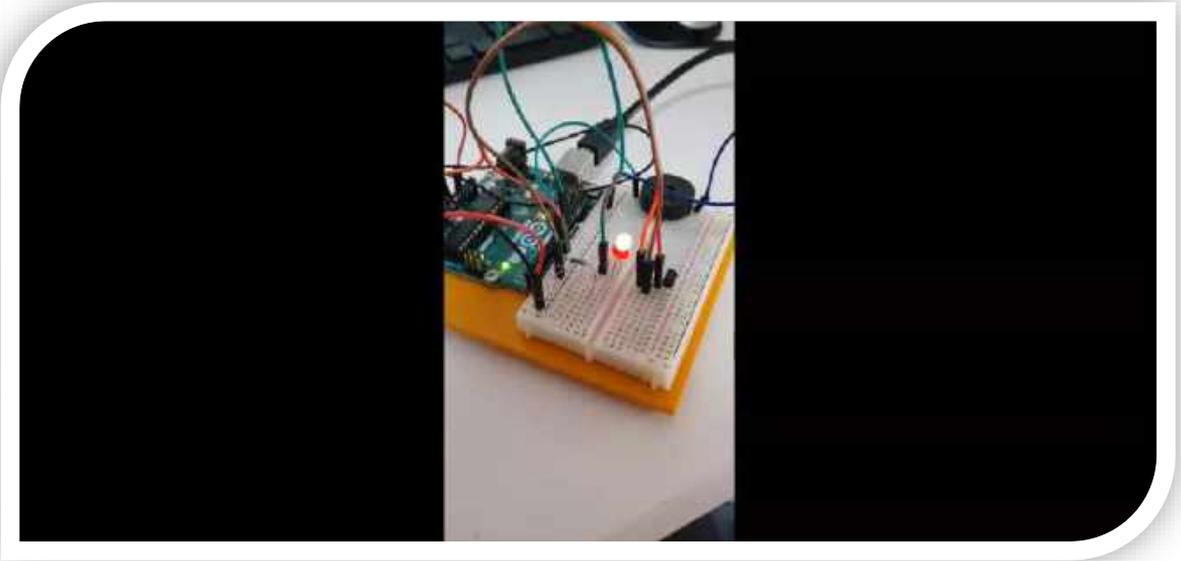
  if (tempC > 35)
    alarm();
  else
    delay(500);
}

void alarm() {
  // Allarme Temperatura troppo alta

  digitalWrite(buzzerPin, HIGH); // turn on the buzzer
  delay(500);
  digitalWrite(buzzerPin, LOW); // turn off the buzzer
  delay(500);
}
```

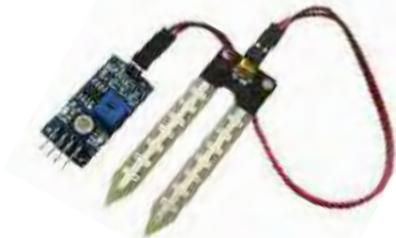
Per visualizzare lo sketch clicca [qui](#)

TEMPERATURA



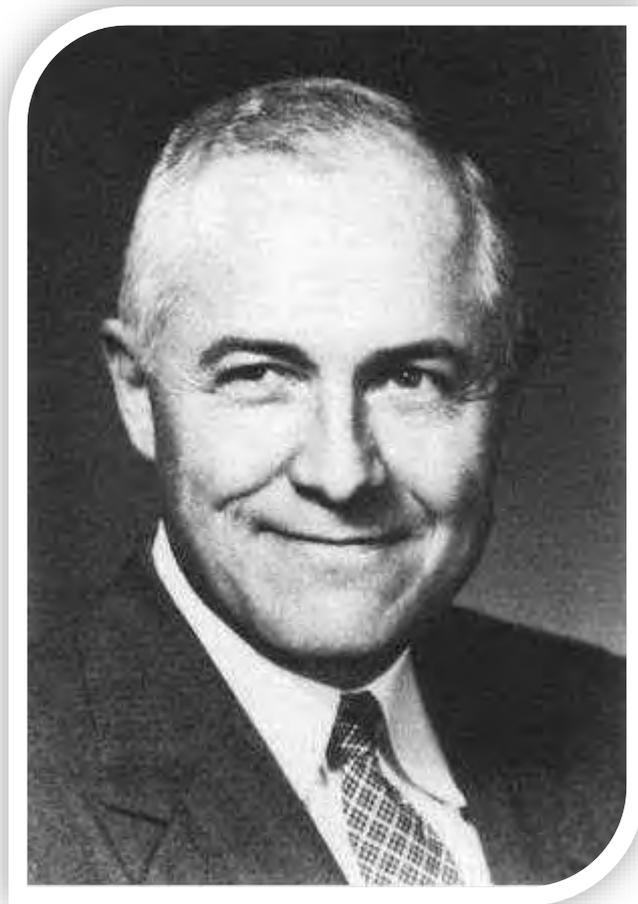
SENSORE DI UMIDITÀ DEL TERRENO

NOI NON ABBIAMO **UN SENSORE DI UMIDITÀ.**



«Non chiedere i mezzi per raggiungere quello che vuoi. Vedi cosa hai a disposizione e cerca di immaginare cosa puoi ottenere.»

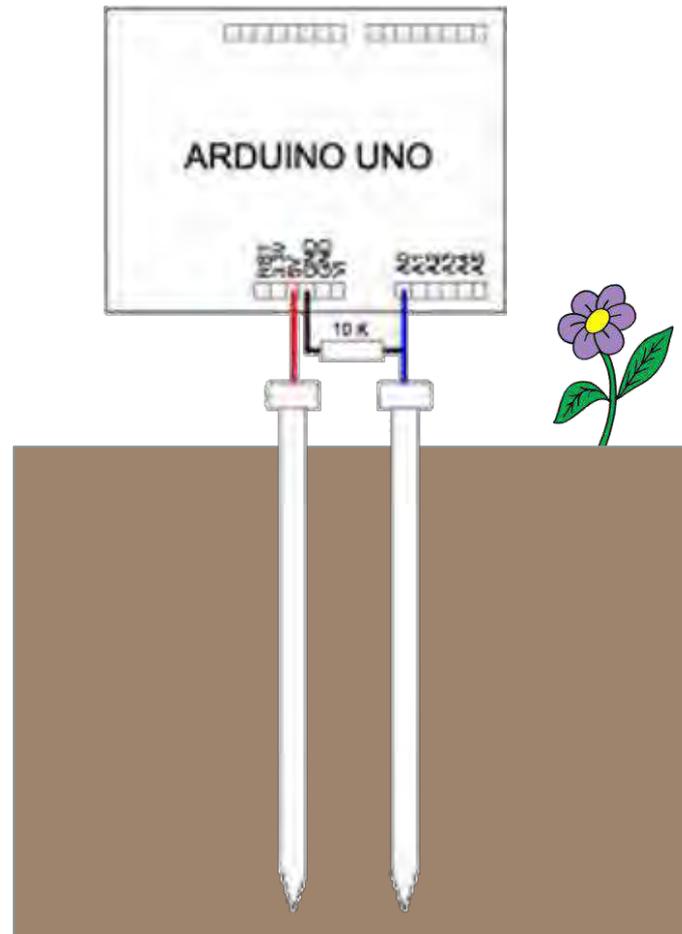
Thomas Kilgore Sherwood



SENSORE DI UMIDITÀ DEL TERRENO

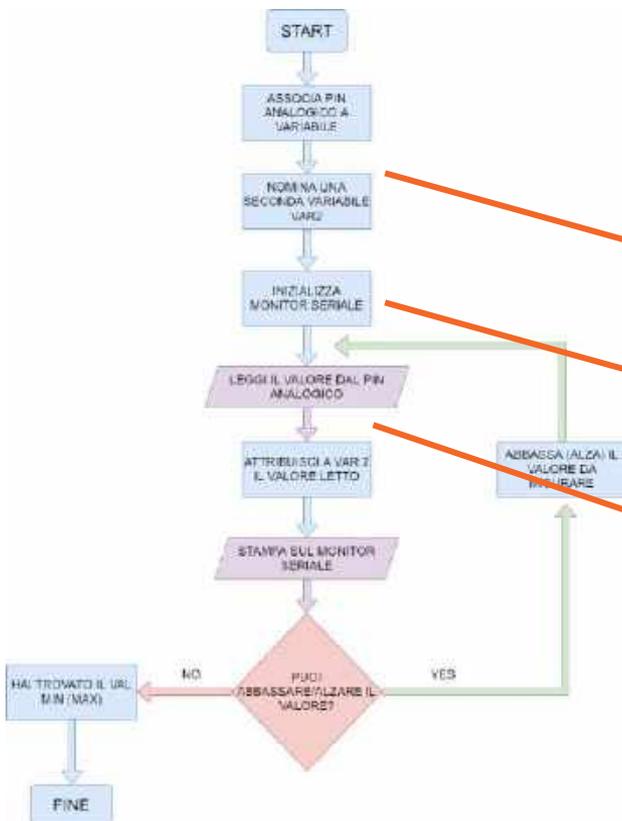
1. Due **chiodi zincati**
2. Una resistenza da **10 kOhm**
3. **Terreno secco** il sensore leggerà **0 V**
4. **Terreno saturo** di acqua il sensore leggerà **1023 V**
5. Con l'istruzione **map ()** possiamo trasformare la lettura nel **range di umidità** che vogliamo

POSSIAMO CREARE UNA **PROCEDURA DI TARATURA?**



VALORE DI SOGLIA - PROCEDURA DI TARATURA

Anche per il **range di misura** di un **sensore** posso utilizzare lo stesso approccio modulare.



```

sketch_sep05e | Arduino 1.8.12
File Modifica Sketch Strumenti Aiuto

sketch_sep05e$

/*Iniziamo due variabili. La prima leggerà del sensore,
 * La seconda memorizzerà il valore letto.
 */
int SENSORE = A0;
int VALORE = 0;

void setup() {
  //Inizializziamo il monitor seriale
  Serial.begin (9600);
}

void loop() {
  // leggiamo il valore e stampiamolo sul monitor serial=
  VALORE = analogRead(SENSORE);
  Serial.println ("VALORE: ");
  Serial.println (VALORE);
  delay (1000);
}
  
```

Per visualizzare lo sketch clicca [qui](#)

SISTEMA DI IRRIGAZIONE

NOI NON ABBIAMO **UNA POMPA DI ALIMENTAZIONE.**



SISTEMA DI IRRIGAZIONE - PRIMA SOLUZIONE

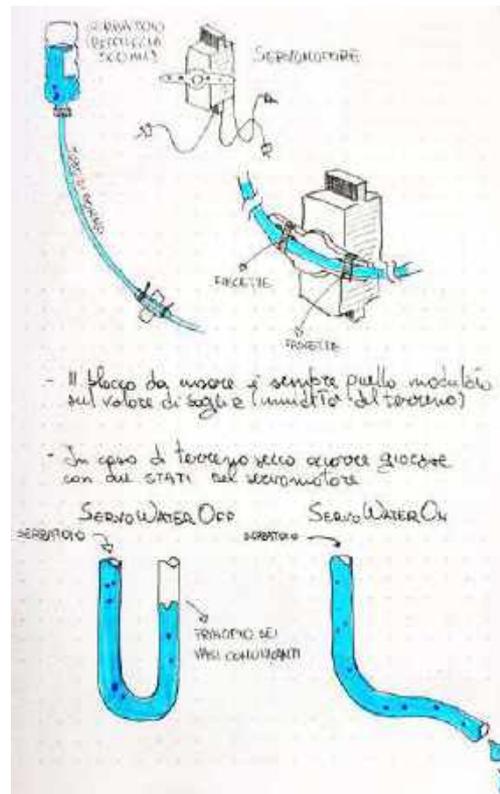
TEORIA DEL SERBATOIO

1. velocità di efflusso
2. cono delle particelle
3. Tempo di bucciaratura

EQUAZIONE DI BERNOULLI

$$P_1 + \rho g h_1 + \frac{1}{2} \rho v_1^2 = P_2 + \rho g h_2 + \frac{1}{2} \rho v_2^2$$

$$P + \rho g h + \frac{1}{2} \rho v^2 = K$$



- Il blocco da usare è sempre quello modulare sul valore di soglia e l'umidità del terreno
- In caso di terreno secco occorre giocare con due STATI nel servomotore

SERVO WATER OFF SERVO WATER ON

FRANGITIO DEI VASI COLLEGAMENTI

SISTEMA DI IRRIGAZIONE PRIMA SOLUZIONE

1. Definiamo variabili per la lettura del sensore e per l'inizializzazione del servomotore;
1. Definiamo le posizioni di apertura e chiusura del servo;
1. Leggiamo dal sensore
1. Ciclo condizionato che apre e chiude il servo;

```

#include <Servo.h>
//nomiamo il nostro servomotore
Servo waterServo;

// impostiamo le variabili per il sensore di umidità
int soilSensorPin = 0;
int servoPin = 3; //rimoviamoci che il pin del servo deve essere PWM
int soilMoisture;

//impostiamo le posizioni di apertura e chiusura del servo
int waterOn = 0; //posizione di apertura
int waterOff = 120; //posizione di chiusura;

void setup() {
  // installiamo il monitor seriale
  Serial.begin(9600);

  //collegiamo il servo al pin
  waterServo.attach(servoPin);
}

void loop() {
  // Leggiamo l'umidità del terreno
  soilMoisture = analogRead(soilSensorPin);

  //ciclo di controllo dell'irrigazione
  if (soilMoisture > 200) {
    //stampa i valori sul seriale
    Serial.println("Valore del sensore di umidità = ");
    Serial.println(soilMoisture);

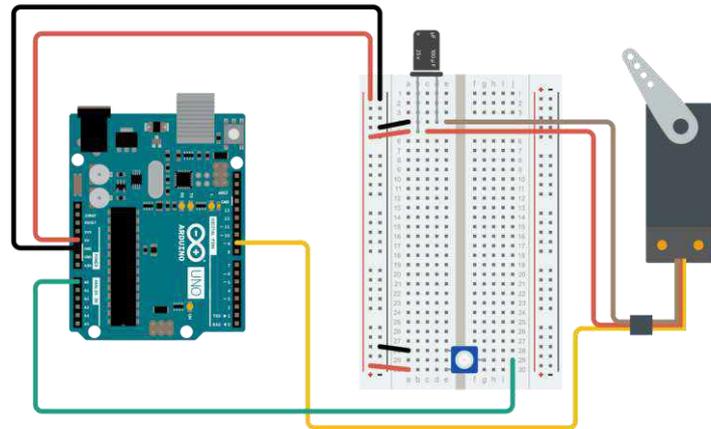
    //metti il servo in posizione off
    waterServo.write(waterOff);
    delay(2000);
  }
  else {
    //stampa i valori sul seriale
    Serial.println("Valore del sensore di umidità = ");
    Serial.println(soilMoisture);

    //metti il servo in posizione on
    waterServo.write(waterOn);
    delay(2000);
  }
}

```

Per visualizzare lo sketch clicca [qui](#)

SISTEMA DI IRRIGAZIONE - SECONDA SOLUZIONE



SISTEMA DI IRRIGAZIONE SECONDA SOLUZIONE

```
innaffiamento_soluzione_2 | Arduino 1.8.12
File Modifica Sketch Strumenti Aiuto

innaffiamento_soluzione_2

//incluiamo la libreria per usare i servo
#include <Servo.h>
//nominiamo il nostro servomotore
Servo waterServo;

//Buzzer
const int buzzerPin = 8;

//potenziometro
const int potPin = A0;
int potValue = 0;
int servoAngle = 0;

// Impostiamo le variabili per il sensore di umidità
int soilSensorPin = 0;
int servoPin = 3; //ricordarsi che il pin del servo deve essere PWM
int soilmoisture;

void setup() {
  //buzzer come output
  pinMode (buzzerPin, OUTPUT);

  // inizializziamo il monitor seriale;
  Serial.begin (5000);

  //colleghiamo il servo al pin
  waterServo.attach(servoPin);
}
```

```
innaffiamento_soluzione_2 | Arduino 1.8.12
File Modifica Sketch Strumenti Aiuto

innaffiamento_soluzione_2

void loop() {
  // Leghiamo l'umidità del terreno e il valore del potenziometro
  soilmoisture = analogRead(soilSensorPin);
  potValue = analogRead(potPin);
  servoAngle = map(potValue, 0, 1023, 0, 179);

  //moviamo il servo
  waterServo.write(servoAngle);

  //stampa sul seriale
  Serial.print("potValue: ");
  Serial.print(potValue);
  Serial.print(", servoAngle: ");
  Serial.println(servoAngle);

  //ciclo di controllo dell'umidità
  if (soilmoisture < 200) {
    //Stampa i valori sul seriale
    Serial.print ("valore del sensore di umidità = ");
    Serial.println (soilmoisture);
    //Allarme
    alarm ();
    delay (2000);
  }
  else {
    //Stampa i valori sul seriale
    Serial.print ("valore del sensore di umidità = ");
    Serial.println (soilmoisture);
    delay (2000);
  }
}

void alarm () {
  //Allarme
  digitalWrite (buzzerPin, HIGH);
  delay (500);
  digitalWrite (buzzerPin, LOW);
  delay (500);
}

Salva tutto e compila
```

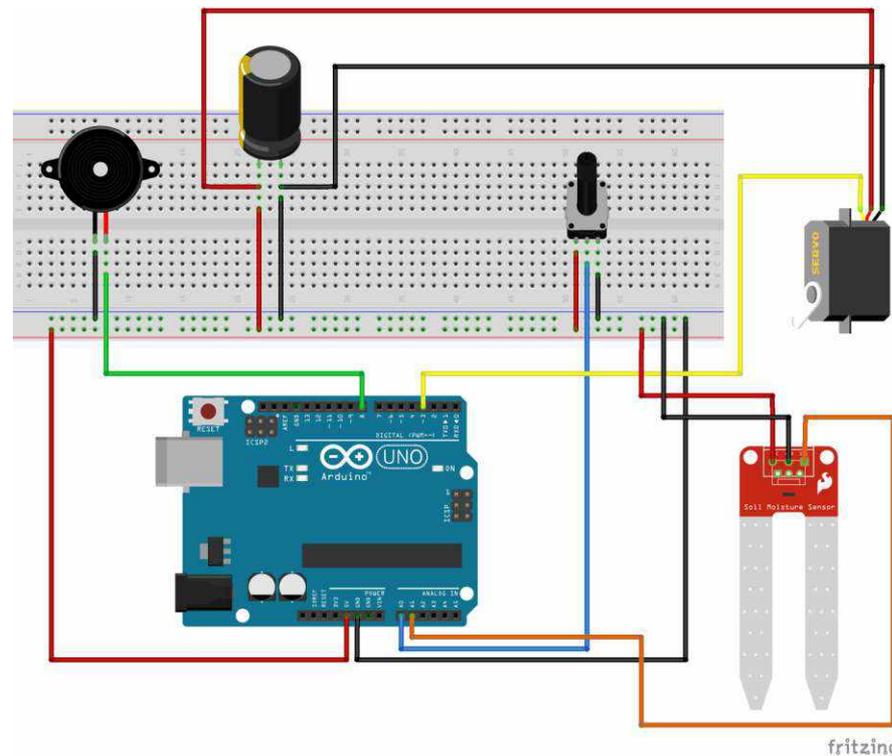
Per visualizzare lo sketch clicca [qui](#)

SCHEMA DI COLLEGAMENTO

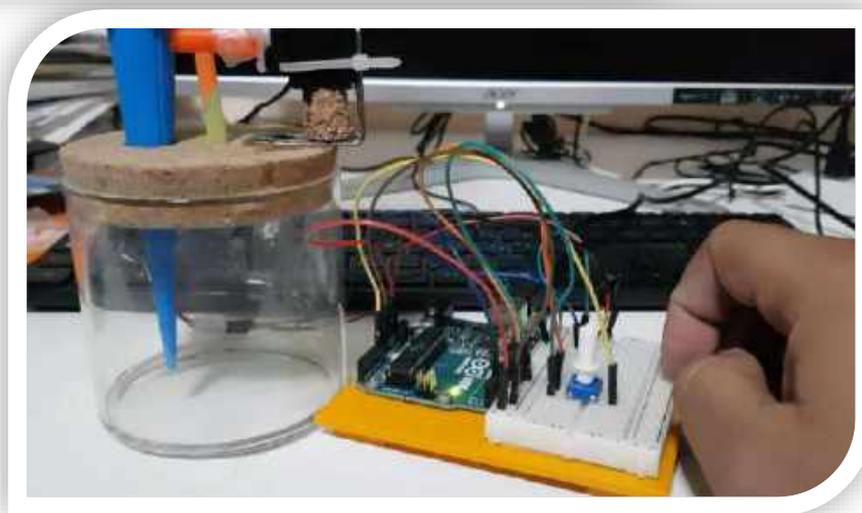
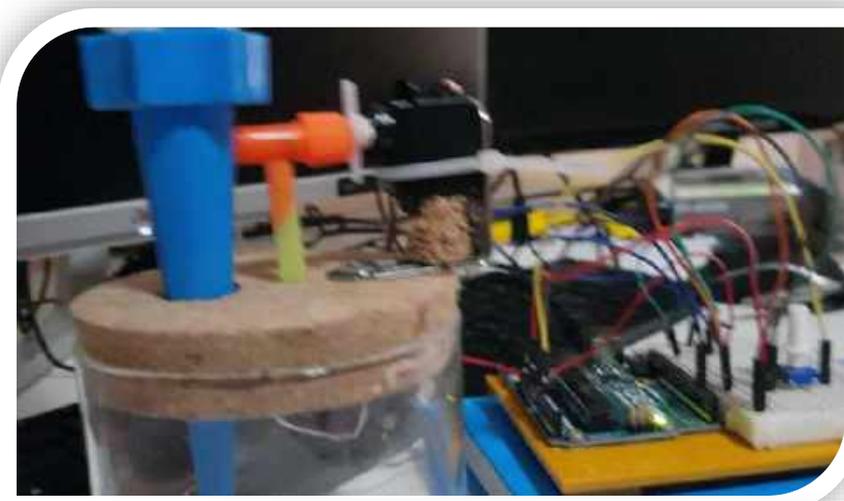
Cosa succede se non ho la giusta **umidità**?

Se
l'umidità
è bassa

- Segnale di allarme
- Irrigazione manuale



SISTEMA DI IRRIGAZIONE



A. IDEE PER IL PROGETTO



Raccogliete in una lista le idee possibili su come programmare e ideare un circuito per la serra con sistema di controllo del clima. Prendete nota delle idee nello spazio sottostante.

Le idee possono includere:

- Numero e colori dei LED utilizzati e per cosa vengono utilizzati.
- Come il servomotore è utilizzato e controllato.
- Come i sensori di temperatura e luminosità vengono utilizzati e cosa accade in base ai dati forniti dai sensori.
- Come vengono utilizzati dispositivi come potenziometri, pulsanti o dati dal serial monitor per controllare manualmente parti del circuito.

B. SCHEMA ELETTRICO

Disegnate uno schema elettrico del vostro circuito della serra con clima controllato. Controllate di prendere nota di ogni cambiamento che effettuate al vostro circuito durante la sua costruzione e che venga rappresentato anche qui nel vostro diagramma.

Gli schemi elettrici variano da progetto a progetto. Adesso, dovremmo avere maggiore familiarità con i simboli e la rappresentazione di ogni componente eccetto il se

LOGBOOK

TABELLA PER LA VALUTAZIONE

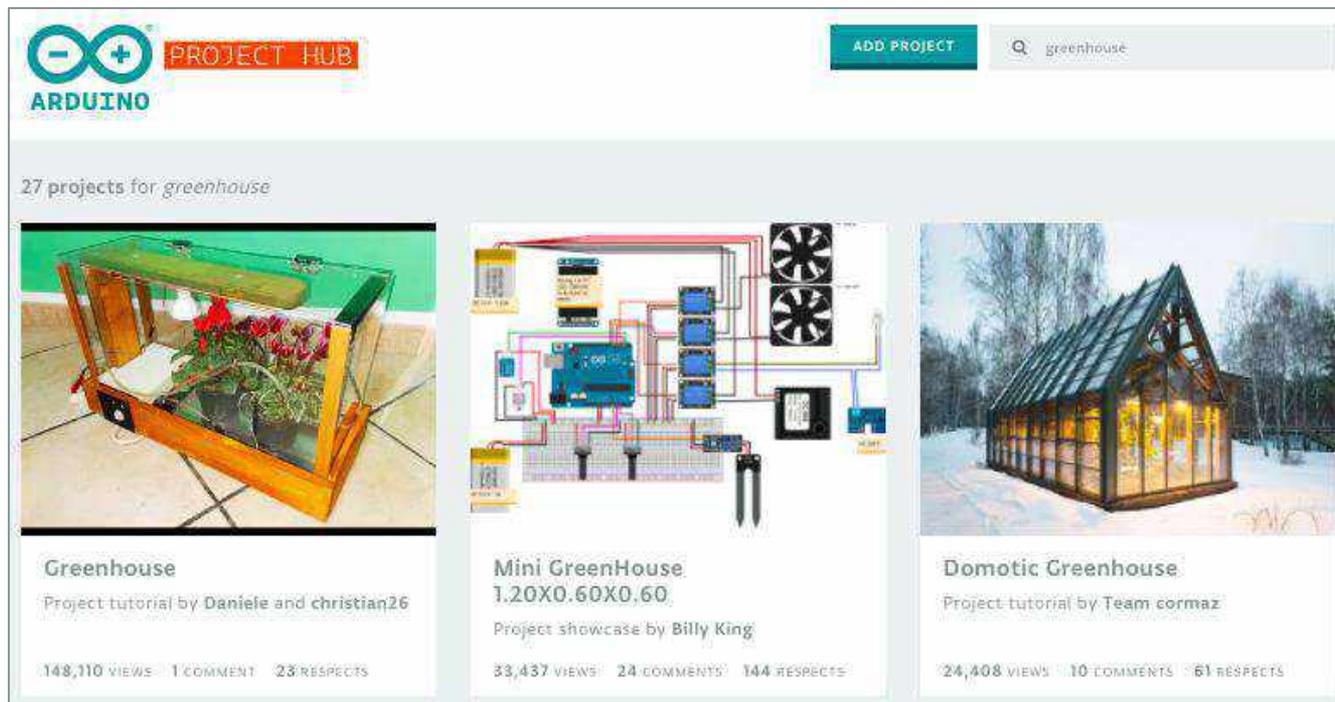
*G. PROJECT SCORING RUBRIC (OPTIONAL)



**Complete this section if you were instructed so by an educator*

Category	3	2	1	0	Total
Schematic Diagram	Diagram accurately represents the circuit developed.	Diagram mostly represents the circuit developed but has minor mistakes.	Diagram was attempted and somewhat represents the circuit developed with some mistakes.	Diagram was not attempted or does not represent the circuit developed.	
Circuit Prototype	Circuit prototype was developed, meeting all criteria and constraints	Circuit prototype was developed, meeting 75% or more of the criteria and constraints.	Circuit prototype was developed, meeting 50% or more of the criteria and constraints.	Circuit prototype was developed, meeting less than 50% of the criteria and constraints.	

ARDUINO PROJECT HUB - PER ISPIRARSI



The screenshot shows the Arduino Project Hub interface. At the top left is the Arduino logo with a minus and plus sign, and the text 'PROJECT HUB' in a red box. To the right is an 'ADD PROJECT' button and a search bar containing the text 'greenhouse'. Below the search bar, it says '27 projects for greenhouse'. Three project cards are displayed:

- Greenhouse**: Project tutorial by Daniele and christian26. 148,110 views, 1 comment, 23 respects. The image shows a wooden frame with a glass top containing plants.
- Mini GreenHouse 1.20X0.60X0.60**: Project showcase by Billy King. 33,437 views, 24 comments, 144 respects. The image shows a breadboard with an Arduino Uno, various sensors, and a fan.
- Domotic Greenhouse**: Project tutorial by Team cormaz. 24,408 views, 10 comments, 61 respects. The image shows a wooden greenhouse with a glass roof, illuminated from within, in a snowy outdoor setting.

<https://create.arduino.cc/projecthub/search?q=+greenhouse>

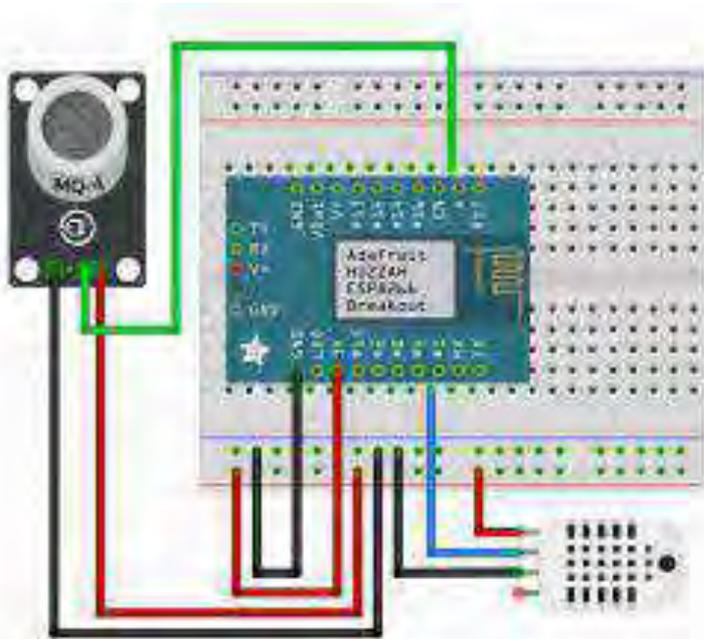


SERRA IDROPONICA



ACQUARIO

IOT E FOTOSINTESI CLOROFILLIANA



```
esp8266 Pinout V1.0
File: Monitor Sketch, Source: Ada:
esp8266

#include <DHTesp.h>
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"

#define DHTTYPE DHT11
#define DHTPIN D5

int pin1 = 1;
int pin2 = 2;
int pin3 = 3;
int pin4 = 4;
int pin5 = 5;
int pin6 = 6;
int pin7 = 7;
int pin8 = 8;
int pin9 = 9;
int pin10 = 10;
int pin11 = 11;
int pin12 = 12;
int pin13 = 13;
int pin14 = 14;
int pin15 = 15;
int pin16 = 16;
int pin17 = 17;
int pin18 = 18;
int pin19 = 19;
int pin20 = 20;
int pin21 = 21;
int pin22 = 22;
int pin23 = 23;
int pin24 = 24;
int pin25 = 25;
int pin26 = 26;
int pin27 = 27;
int pin28 = 28;
int pin29 = 29;
int pin30 = 30;
int pin31 = 31;
int pin32 = 32;
int pin33 = 33;
int pin34 = 34;
int pin35 = 35;
int pin36 = 36;
int pin37 = 37;
int pin38 = 38;
int pin39 = 39;
int pin40 = 40;
int pin41 = 41;
int pin42 = 42;
int pin43 = 43;
int pin44 = 44;
int pin45 = 45;
int pin46 = 46;
int pin47 = 47;
int pin48 = 48;
int pin49 = 49;
int pin50 = 50;
int pin51 = 51;
int pin52 = 52;
int pin53 = 53;
int pin54 = 54;
int pin55 = 55;
int pin56 = 56;
int pin57 = 57;
int pin58 = 58;
int pin59 = 59;
int pin60 = 60;
int pin61 = 61;
int pin62 = 62;
int pin63 = 63;
int pin64 = 64;
int pin65 = 65;
int pin66 = 66;
int pin67 = 67;
int pin68 = 68;
int pin69 = 69;
int pin70 = 70;
int pin71 = 71;
int pin72 = 72;
int pin73 = 73;
int pin74 = 74;
int pin75 = 75;
int pin76 = 76;
int pin77 = 77;
int pin78 = 78;
int pin79 = 79;
int pin80 = 80;
int pin81 = 81;
int pin82 = 82;
int pin83 = 83;
int pin84 = 84;
int pin85 = 85;
int pin86 = 86;
int pin87 = 87;
int pin88 = 88;
int pin89 = 89;
int pin90 = 90;
int pin91 = 91;
int pin92 = 92;
int pin93 = 93;
int pin94 = 94;
int pin95 = 95;
int pin96 = 96;
int pin97 = 97;
int pin98 = 98;
int pin99 = 99;
int pin100 = 100;

#define STASSID
#define STASSID "iPhone di Pasquale"
#define STAPSK "01091c1234"
#define STAPSK "01091c1234"
#define STAPSK "01091c1234"

float TEMP_EXT;
float HUMID_EXT;
float CO2_EXT;

DHTesp dht;

WiFiClient client;

const char* ssid = STASSID;
const char* password = STAPSK;
unsigned long myChannelNumber = 636750 ;
char * myWriteAPIKey = "44VZWPq68V3K6C0N?";

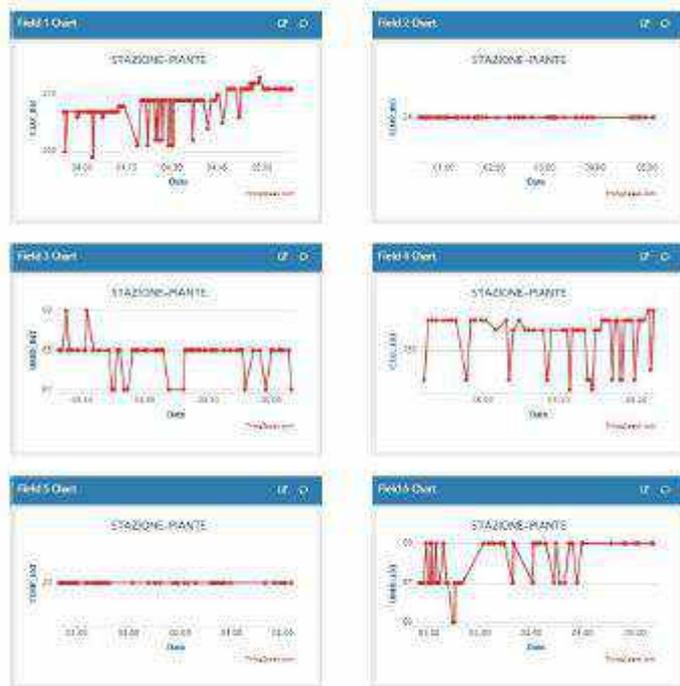
void setup() {
  // put your setup code here, to run once!
  dht.setup(DHTPIN, DHTTYPE);

  WiFi.disconnect();
  Serial.begin(115200);
  Serial.println("Connessione a:");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  ThingSpeak.begin(client);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println(".");
  }
  Serial.println("WiFi connesse");
  Serial.println("IP WiFi =");
  Serial.println(WiFi.localIP());
}

}


```

IOT E FOTOSINTESI CLOROFILLIANA



IOT E FOTOSINTESI CLOROFILLIANA

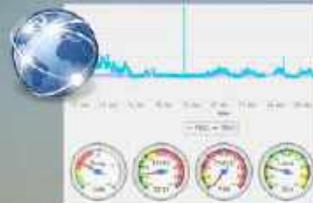


IOT E INQUINAMENTO

<http://www.coderschoolitalia.it/ebook-una-stazione-ambientale-a-basso-costo-per-il-controllo-polveri-sottili/>

Una Stazione Ambientale a basso costo per il controllo polveri sottili.

Making tra attività didattica, monitoraggio civico e salvaguardia ambientale.



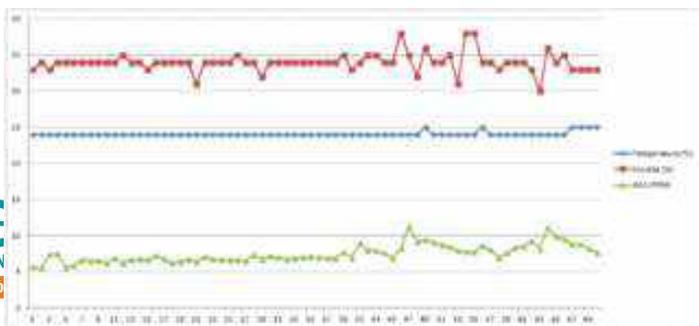
**Francesco Piersoft Paolicelli,
Domenico Aprile, Luca Scalzullo**

OPEN DATA E INQUINAMENTO



https://umap.openstreetmap.fr/it/map/rilievi-ambientali-nocera-inferiore_140212

<https://www.professionistiscuola.it/open-data-e-cittadinanza-digitale/un-amico-un-divano-ed-un-computer-open-data-ed-ambiente-per-un-progetto-di-cittadinanza-digitale.html>



OPEN DATA E INQUINAMENTO

<https://www.piersoft.it/thinkspeak-telegram-click-leggere-valori-del-vostro-arduino/>





GRAZIE MILLE!

A PRESTO!

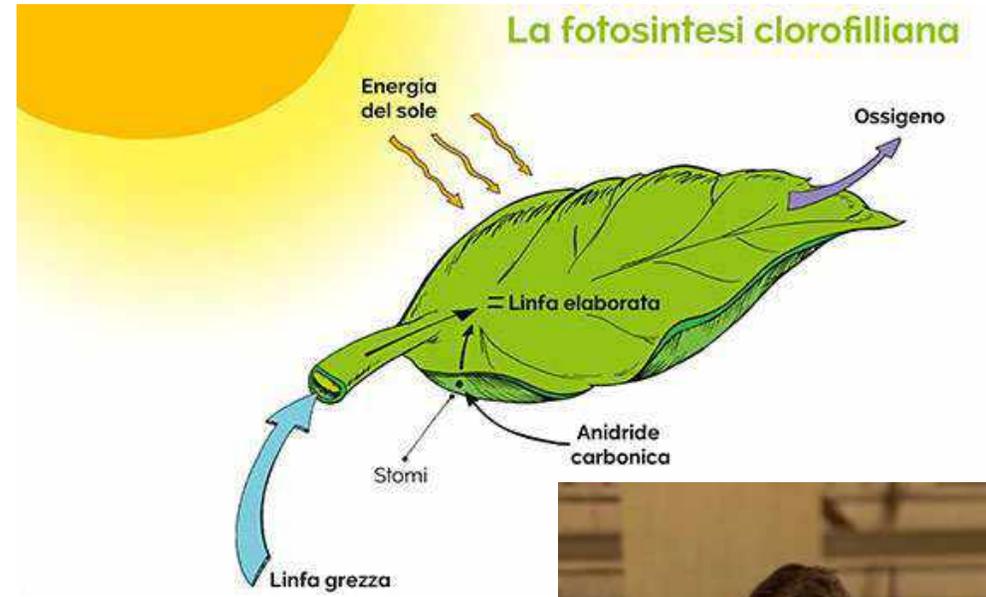
Laboratori Green
creiamo delle piante musicali:
utilizzare i dati per dar voce alla
natura intorno a noi

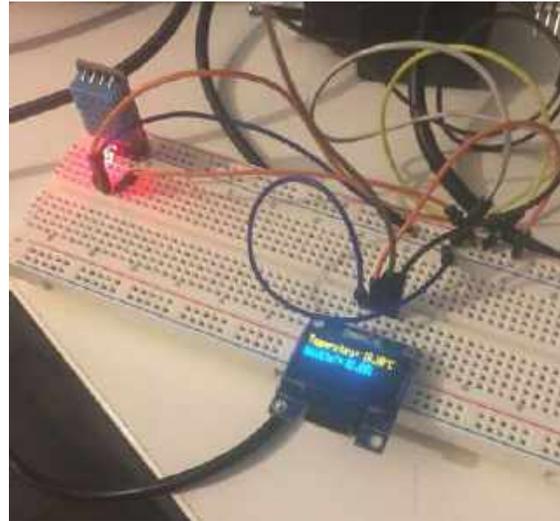
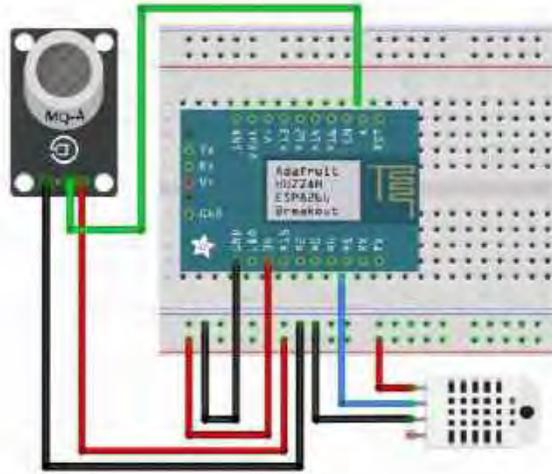


LUCA SCALZULLO



Bandi Green e STEM





ThingSpeak



About ThingSpeak

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.



Collect

Send sensor data privately to the cloud.



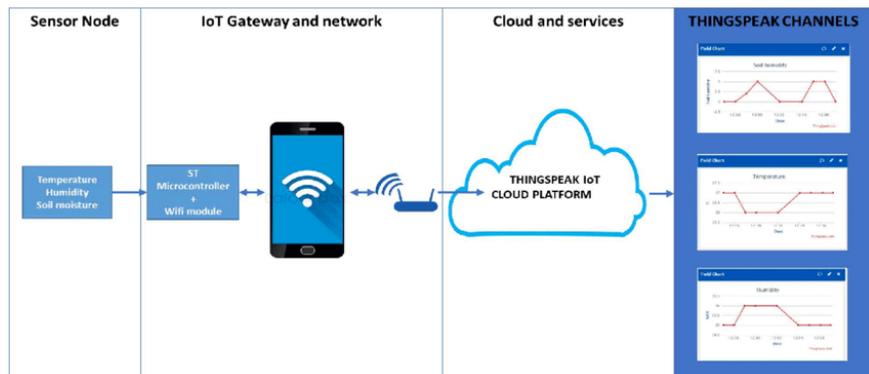
Analyze

Analyze and visualize your data with MATLAB.



Act

Trigger a reaction.



Channel Settings

Percentage complete 70%

Channel ID 738622

Name Respirazione di una pianta

Description L'idea del canale è quella di misurare quanto respira una pianta tenendo sotto controllo la

Field 1 Temperature

Field 2 Humidity

Field 3 CO2_OUT

Field 4 Temperature

Field 5 Humidity

Field 6 CO2_IN

Write API Key

Key U6WFQW8NGTSPR40T

Generate New Write API Key

Read API Keys

Key E1SRKNIU8WTQLYM1

Note

Save Note

Delete API Key

Generate New Read API Key

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=U6WFQW8NGTSPR40T&field=
```

Read a Channel Feed

```
GET https://api.thingspeak.com/channels/738622/feeds.json?api_key=E1
```

Read a Channel Field

```
GET https://api.thingspeak.com/channels/738622/fields/1.json?api_key=
```

Read Channel Status Updates

```
GET https://api.thingspeak.com/channels/738622/status.json?api_key=E
```

```

esp2$
#include <DHTesp.h>
#include <ESP8266WiFi.h>
#include "ThingSpeak.h"

#define DHT1PZ DHT11
#define DHTPIN D5

#ifndef STASSID
#define STASSID "iPhone di Pasquale"
#define STAPSK "Glorgiol234"
#endif

float TEMP_EXT;
float UMID_EXT;
float CO2_EXT;

DHTesp dht;

WiFiClient client;

const char* ssid = STASSID;
const char* password = STAPSK;
unsigned long myChannelNumber = 856758;
char * myWriteAPIKey = "34VINFDQ8HV3KCCV";

void setup() {
  // put your setup code here, to run once:
  dht.setup(DHTPIN, DHTesp::DHT1PZ);

  WiFi.disconnect();
  Serial.begin(115200);
  Serial.print("Connessione a:");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  ThingSpeak.begin(client);
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connessa");
  Serial.print("IE WiFi =");
  Serial.println(WiFi.localIP());
}

```

```

void loop() {
  // put your main code here, to run repeatedly:
  TEMP_EXT = dht.getTemperature();
  UMID_EXT = dht.getHumidity();
  CO2_EXT = analogRead(0);
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println(CO2_EXT);
  ThingSpeak.setField(4, CO2_EXT);
  int x = ThingSpeak.writeField(myChannelNumber, 4, CO2_EXT, myWriteAPIKey);
  ThingSpeak.setField(5, TEMP_EXT);
  int y = ThingSpeak.writeField(myChannelNumber, 5, TEMP_EXT, myWriteAPIKey);
  ThingSpeak.setField(6, UMID_EXT);
  int z = ThingSpeak.writeField(myChannelNumber, 6, UMID_EXT, myWriteAPIKey);

  delay(5000);
}

```


The Sonification Handbook

Edited by

Thomas Hermann, Andy Hunt, John G. Neuhoff



λογος

cost SI

SONIFY Google News Initiative

2001
2002
2003
2004
2006
2008
2009

Welcome to TwoTone

A free open-source web app to turn data into music

Launch App

or

View on GitHub



xSonify

Java-based sonification data analysis

Status: **Alpha** Brought to you by: [candley](#), [scheran7B](#)

GT Sonification Lab

School of Psychology - Georgia Institute of Technology

Sonification Sandbox

ASTRONIFY

A Python package for sonifying astronomical data -
turning telescope observations into sound!



ASTRONIFY

ESOPIANETI



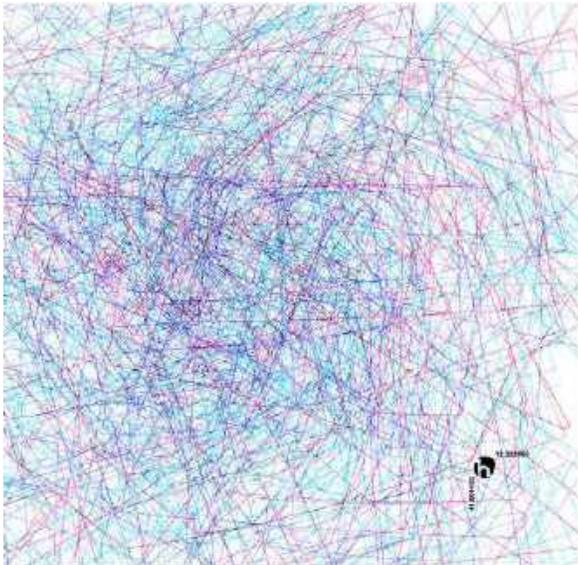
<https://www.iguzzini.com/it/notizie/light-for-future/>

CIAO.
IL MIO NOME È hArt
E SONO 1/2 PERFORMER...

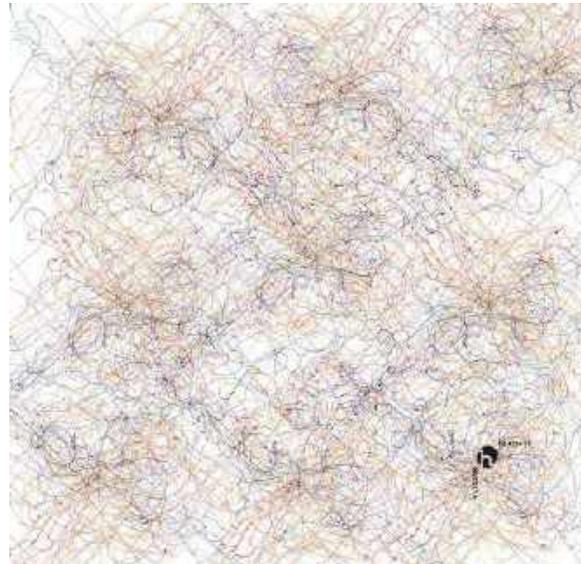
CONOSCIAMOCI MEGLIO



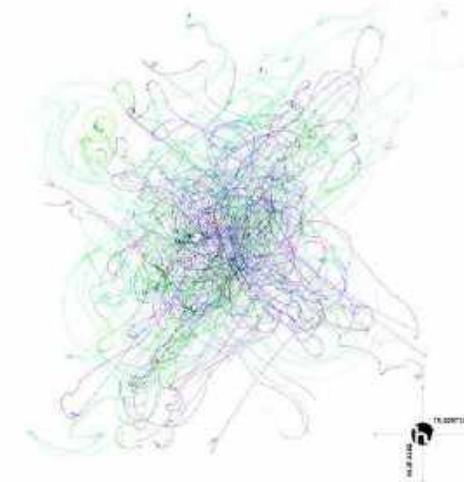
<https://www.hartdesign.eu/>



ROMA, 41.804455, 12.322483



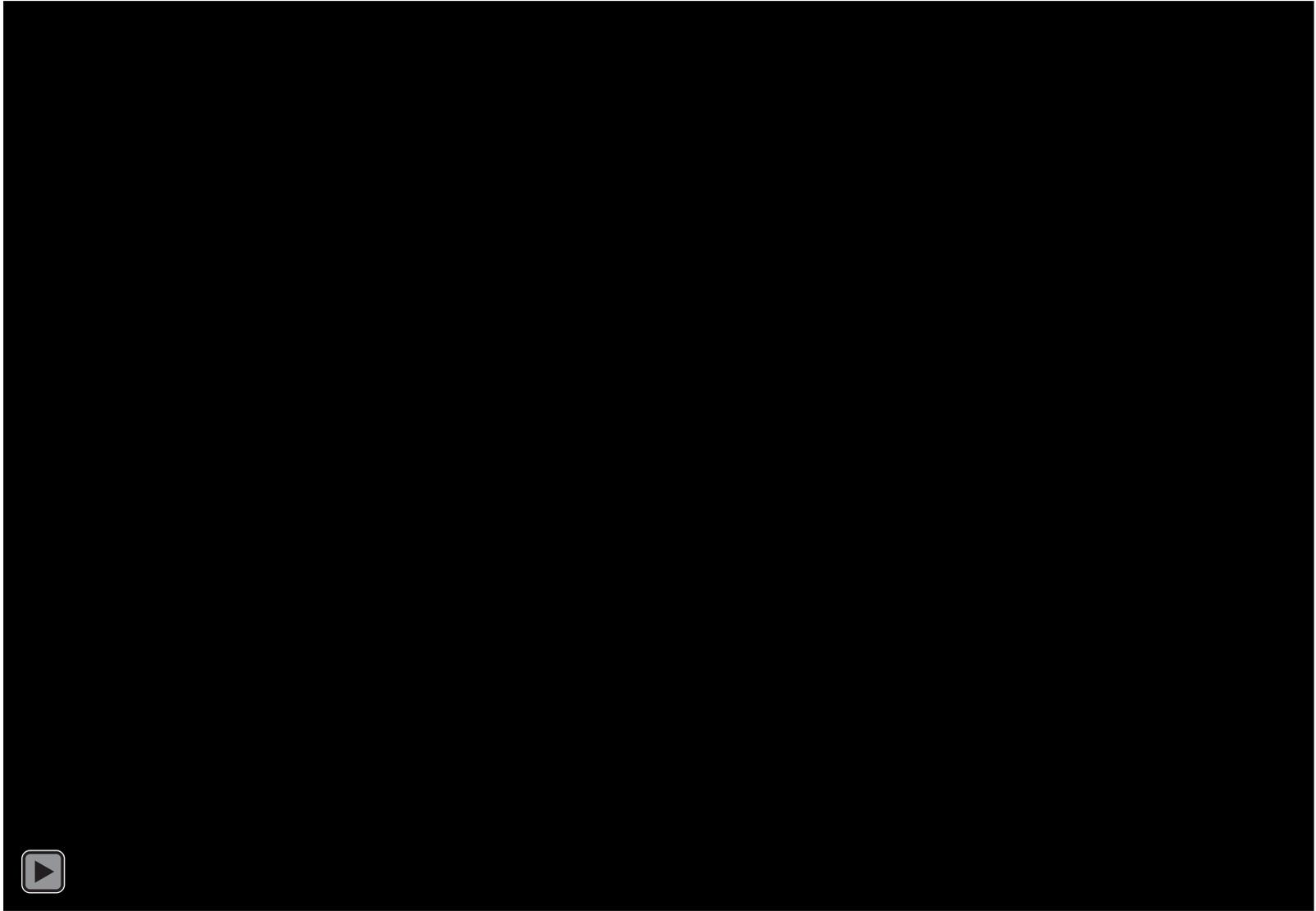
ROMA, 41.910748, 12.476417



SIENA, 43.318382, 11.331720

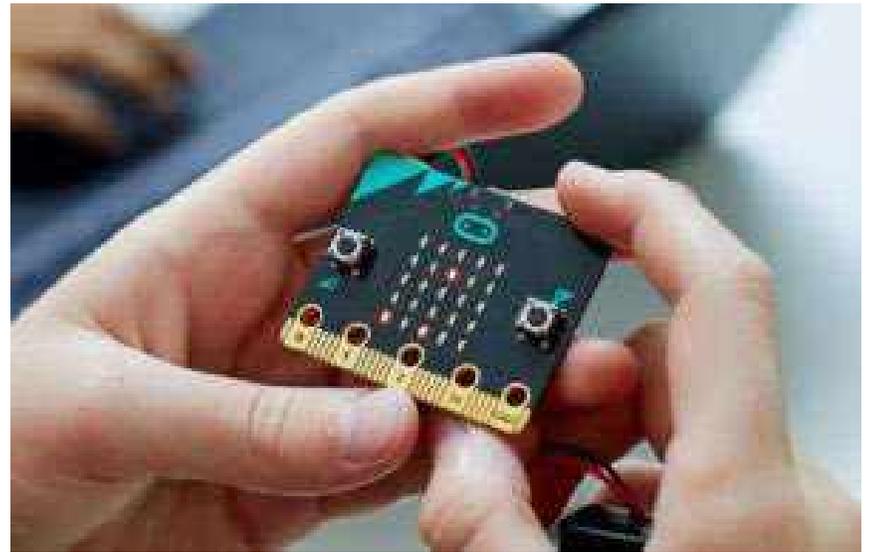
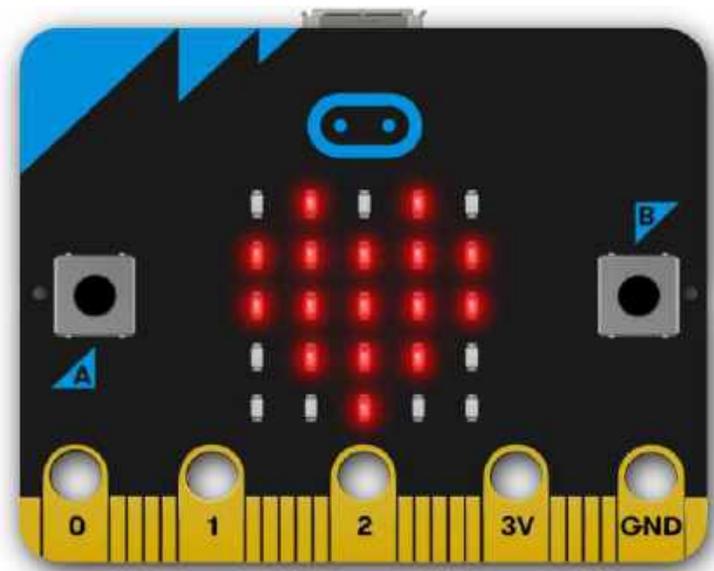
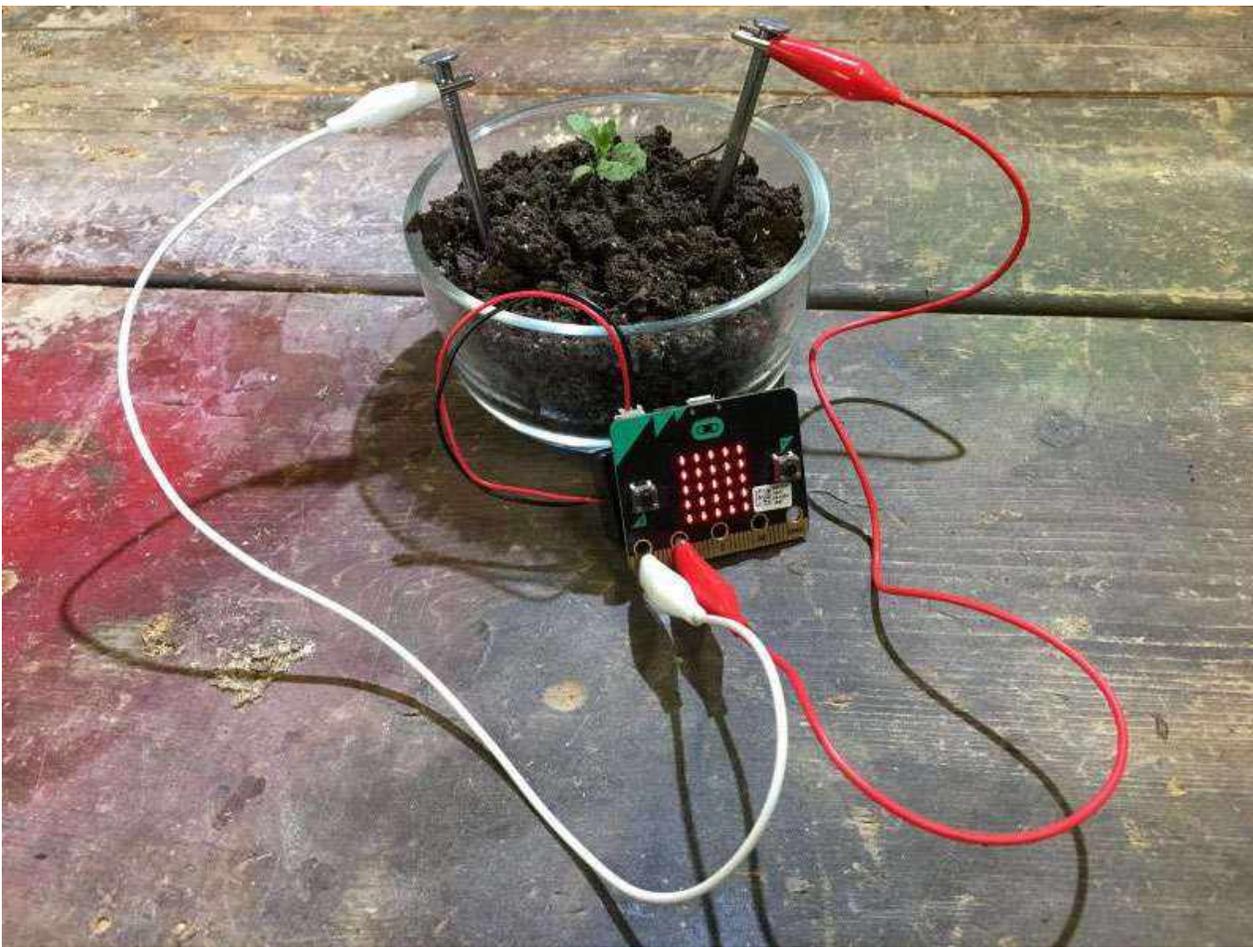


E-Rhythms



<http://individual.utoronto.ca/jboase/software.html>







5.1 Agrumino Lemon (fronte)





<https://www.innovationforeducation.it/academy/come-fare-lezione-a-scuola-e-a-casa-con-arduino-student-kit-lezione-10-progetto-2/>



Titolo

Sottotitolo / parte in evidenza

Testo

Grazie per l'attenzione