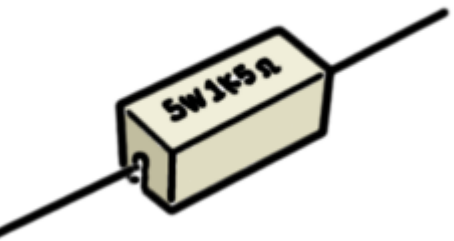
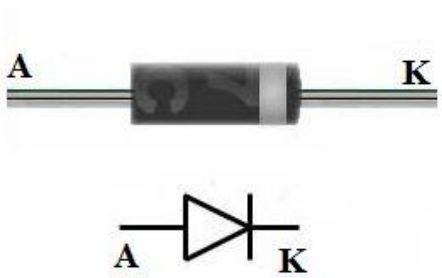
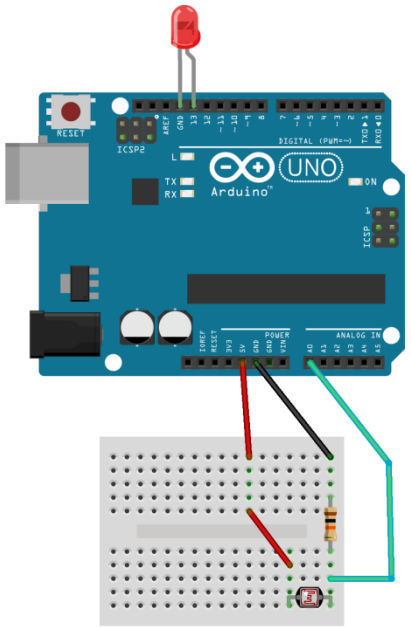
Progettare e creare con Arduino e il "physical computing"

Giorgio De Nunzio giorgio.denunzio@unisalento.it

Dipartimento di Matematica e Fisica – Università del Salento







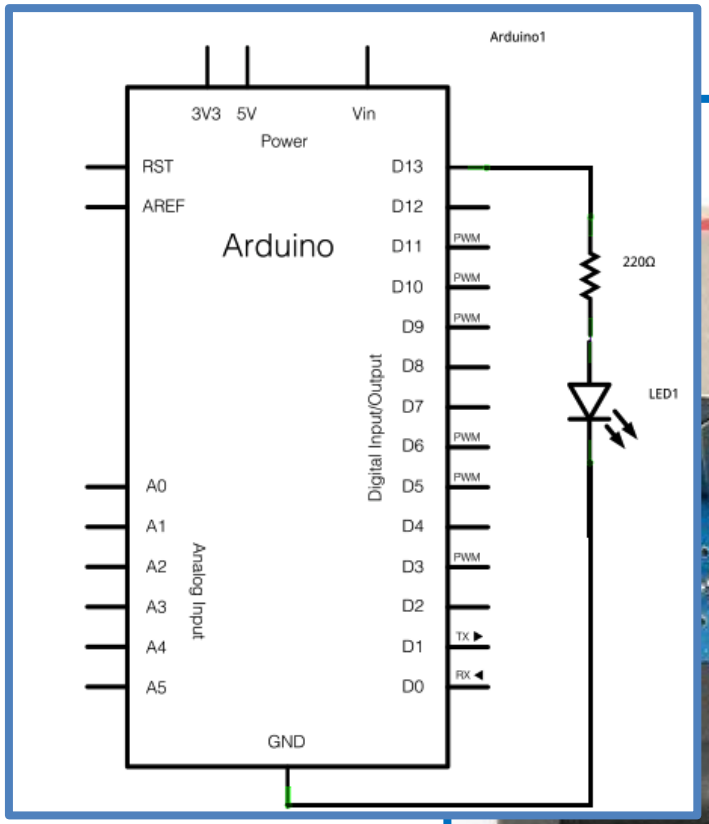
Introduzione

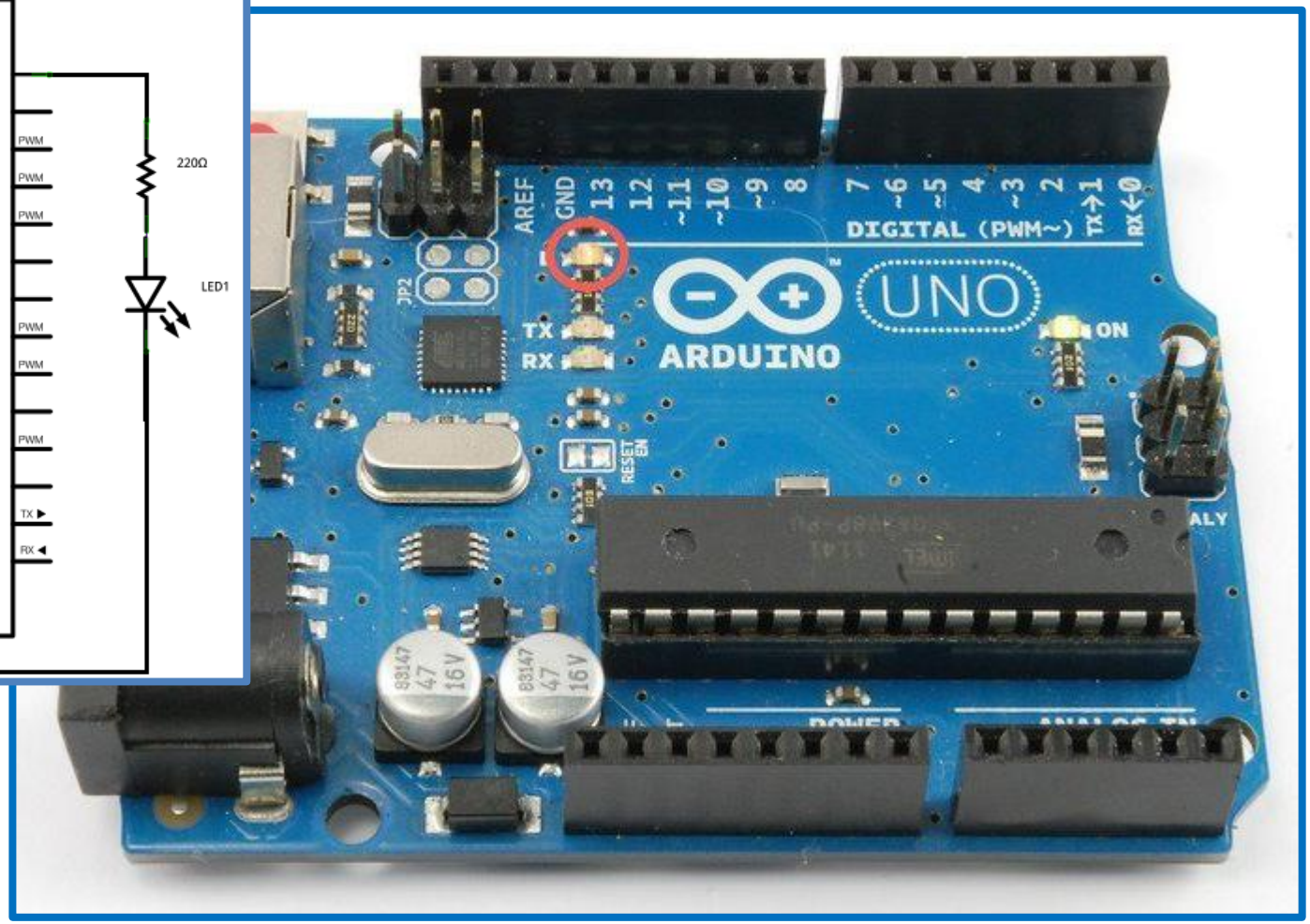
- Il corso parlerà di….
  - Elettronica di base
  - Arduino
  - Programmazione
  - Applicazioni

http://www.dmf.unisalento.it/~denunzio/allow_listing/ARDUINO/

Iniziare con Arduino (1)

Il LED “L”





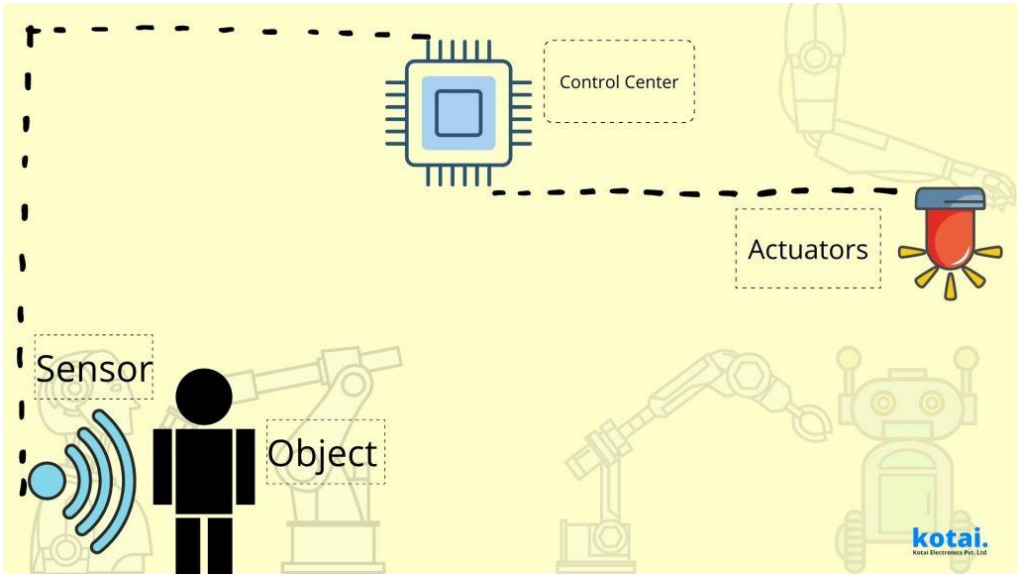
Tasto reset

Elenco argomenti su 3h

- Scheda Arduino; il led onboard
- Che cos'è un led
- TinkerCAD
- Programma vuoto: il setup() e loop()
- Blink del led onboard in TinkerCAD
- Installazione IDE di Arduino; configurazione IDE
- Blink del led onboard su arduino
- Variare la frequenza di blink
- Schema led esterno con resistore
- Che cos'è un resistore; codice dei colori
- Come funziona la breadboard
- Blink del led esterno con resistore in TinkerCAD
- Blink del led esterno con resistore su Arduino
- Collegare un secondo LED, dapprima su TinkerCAD, poi ad Arduino.
- Monitor seriale
- Fotoresistore
- Uso del fotoresistore in TinkerCAD
- Fotoresistore su Arduino
- Monitor e plotter seriale con il fotoresistore

Iniziare con Arduino (1a)

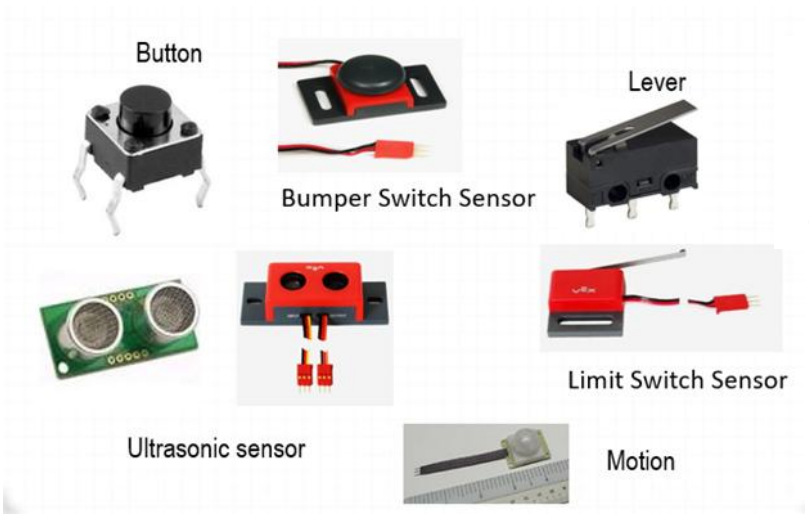
Sensori e attuatori



DC motor

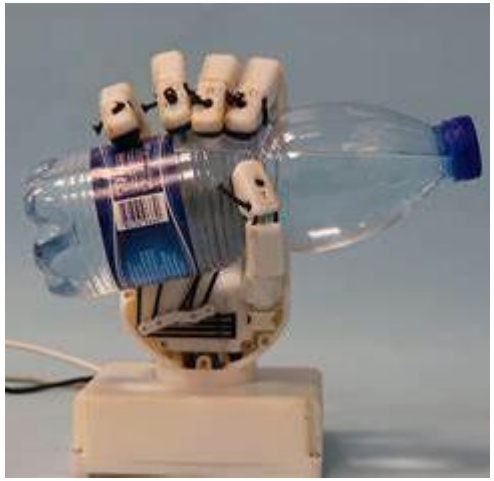
Stepper motor

Electric pump



Temperature

Infrared (obstacle)



Iniziare con Arduino (2)

CORREGGERE: SARA' NECESSARIO COMUNQUE AVERE I DIRITTI DI AMMINISTRATORE PER L'INSTALLAZIONE DEI DRIVER

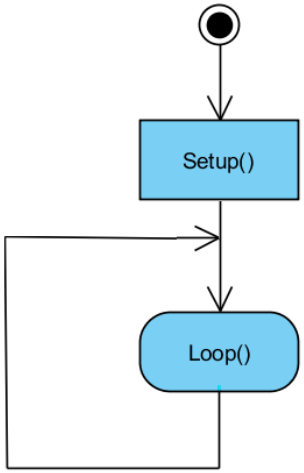
- Se si lavora su un pc sul qual
  decompattare la versione zip
  (*Integrated Development En*
  da: http://arduino.cc/en/Main/Software, scegliendo "ZIP file" e poi "JUST DOWNLOAD"
- Se si tratta del proprio pc, sul quale si dispone dei diritti di amministratore, o se comunque i diritti di amministratore sono garantiti (almeno per il proprio utente!), scaricare la versione installabile (.exe, "Win10 and newer…"), lanciarla e seguire le istruzioni.
- Collegare Arduino al computer tramite il cavo USB fornito nel kit. Se è richiesta dal computer l'installazione dei driver, seguire la procedura.
- Lanciare il programma arduino.exe dalla directory decompattata (nel caso d'uso del .zip) o dal menu di avvio del pc, o dall'icona sul desktop; dovrebbe comparire uno "sketch" arduino, contenente le funzioni (vuote) setup() e loop(); se non succede, usare il menu `File→New Sketch` (per impostare la lingua italiana: `File→Preferences→Language`):

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

- Scegliere la voce di menu "`Strumenti → Porta`" e selezionare la porta alla quale fa capo Arduino. In "`Scheda`" selezionare: Arduino UNO
- Riempire lo sketch come nella prossima slide, ricordando che la distinzione tra maiuscole e minuscole è importante! Il programma è anche già disponibile nella libreria di esempi della IDE di Arduino, con il nome *blink.ino* (menu File -> Esempi -> 01.Basics -> Blink).

Iniziare con Arduino (2a): il led onboard

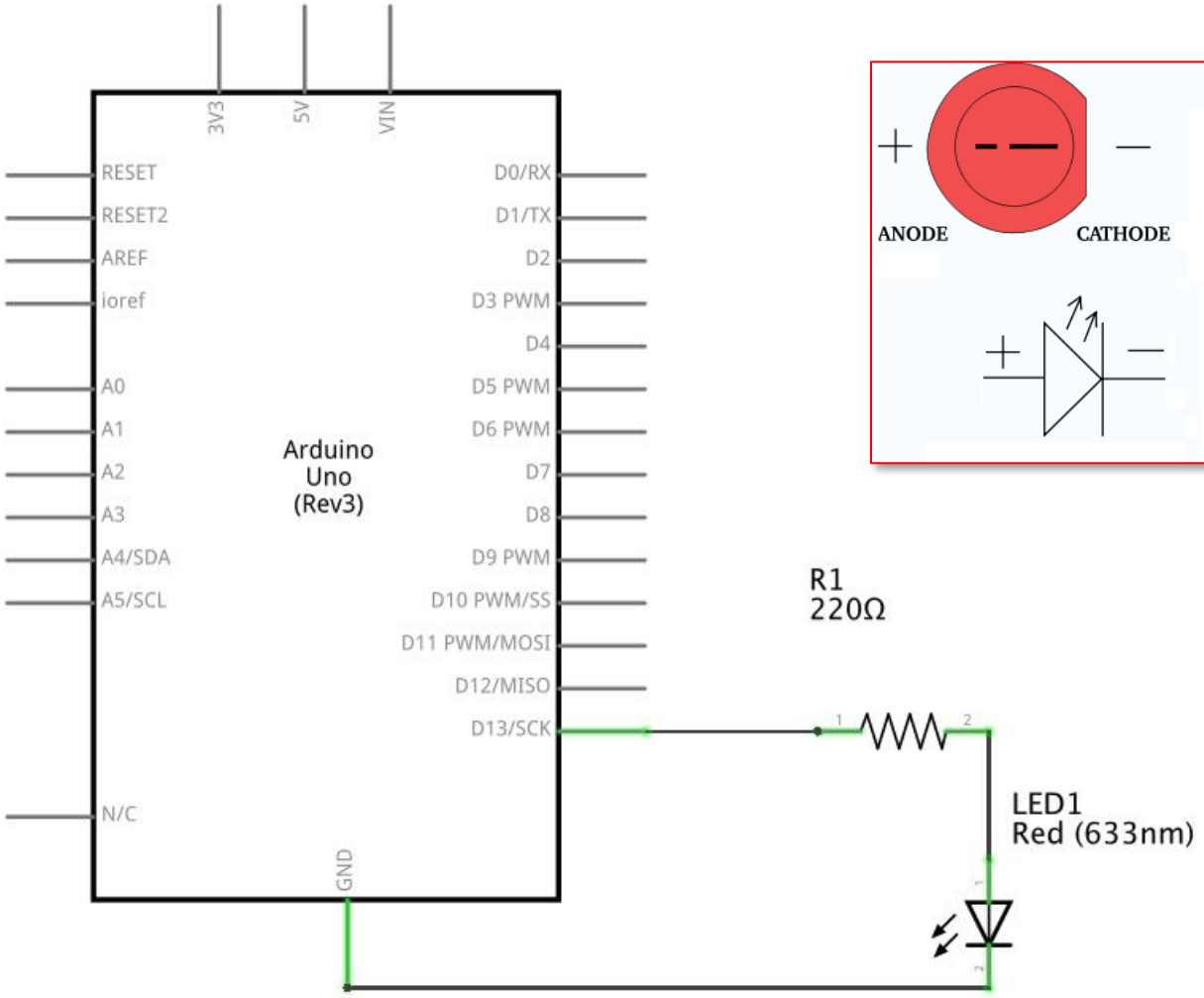
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

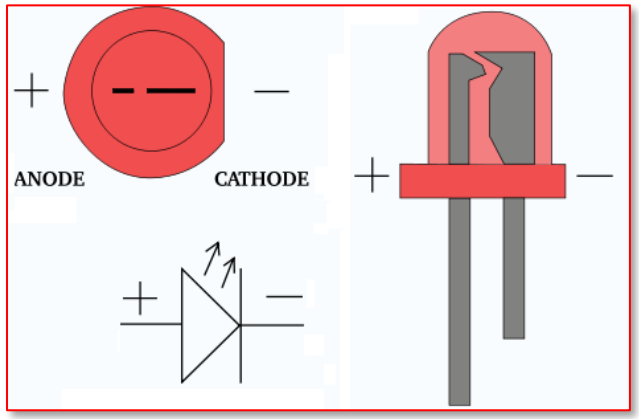


Le righe che iniziano con // sono commenti al codice!
Possono essere trascurate per brevità

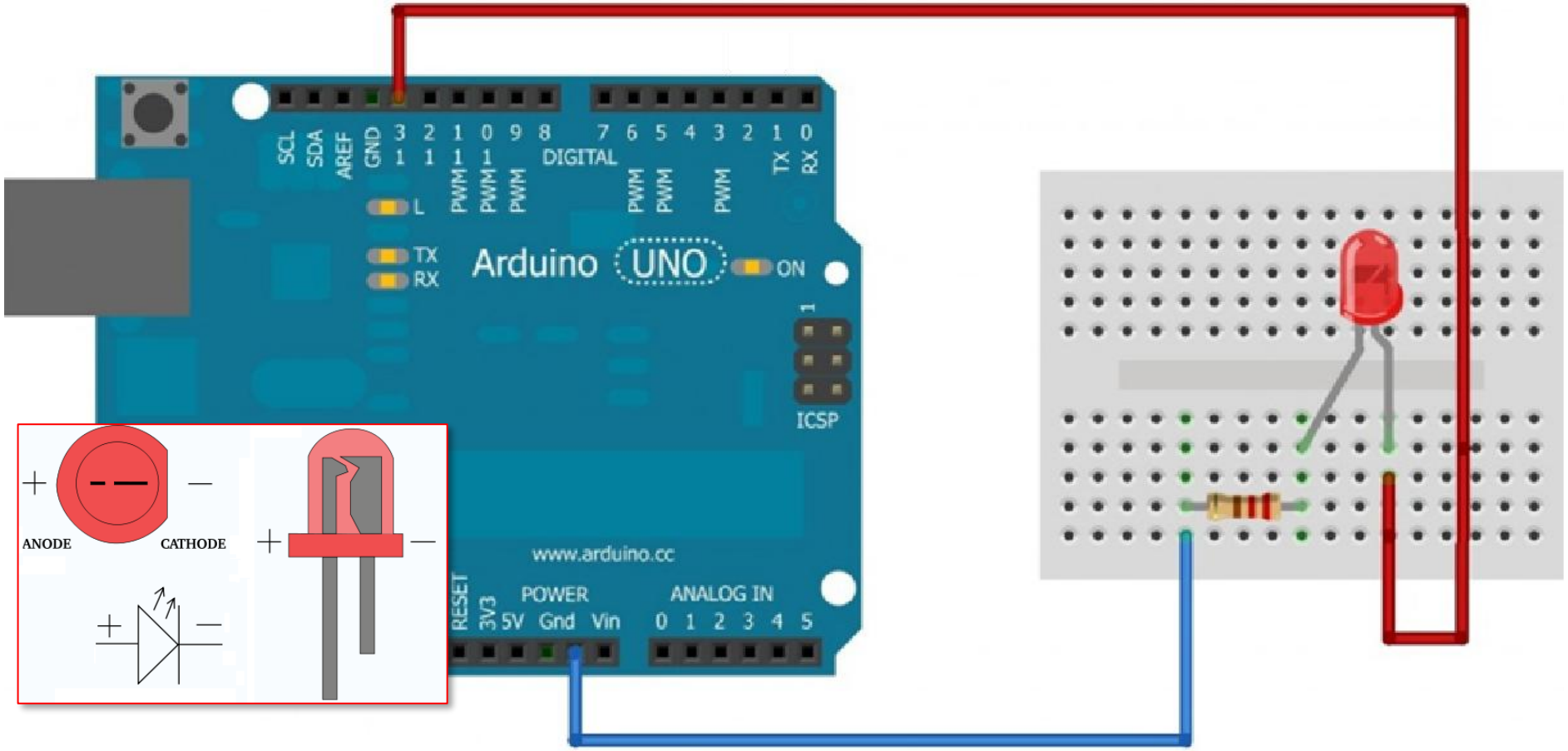
- Il blocco di codice (o "funzione") `setup()` è eseguito una sola volta, all'inizio; il blocco `loop()` è eseguito "all'infinito" dopo `setup()`.
- Cliccare sull'icona di "Verifica" (1a icona sotto il menu, contrassegnata da √) . Sarà chiesto di salvare il file sorgente. Scegliere la directory Documenti (o altro) e dare un nome al programma (esempio: test01). Automaticamente sarà aggiunta l'estensione .ino. Poi lo sketch sarà compilato e ne sarà verificata la correttezza formale.
- Cliccare su "Carica" (seconda icona sotto il menu, la freccia). Il programma sarà compilato e trasmesso ad Arduino. Il piccolo LED sulla basetta di Arduino inizierà a lampeggiare a intervalli di un secondo di accensione (on) e uno di spegnimento (off).

Iniziare con Arduino (2b): collegare un LED alla board di Arduino

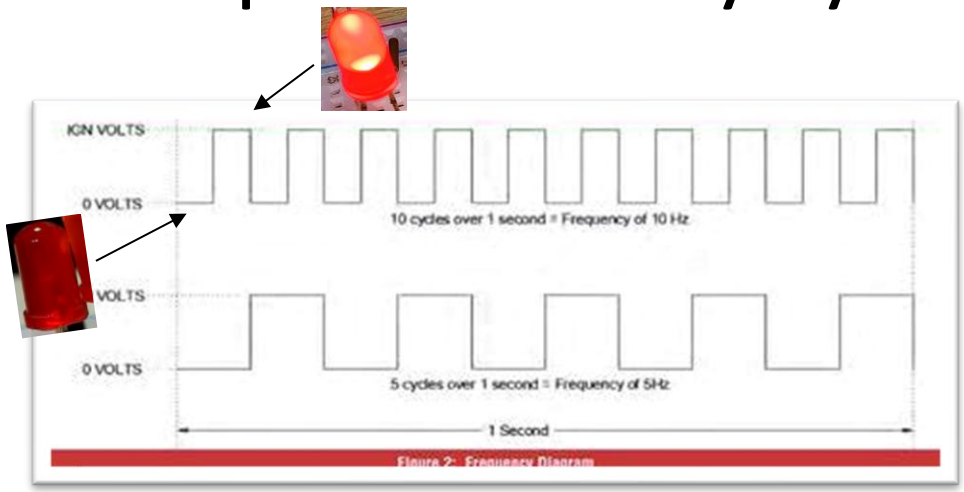




Iniziare con Arduino (2b): collegare un LED alla board di Arduino



Frequenza e duty cycle ("ciclo di lavoro") (2b)

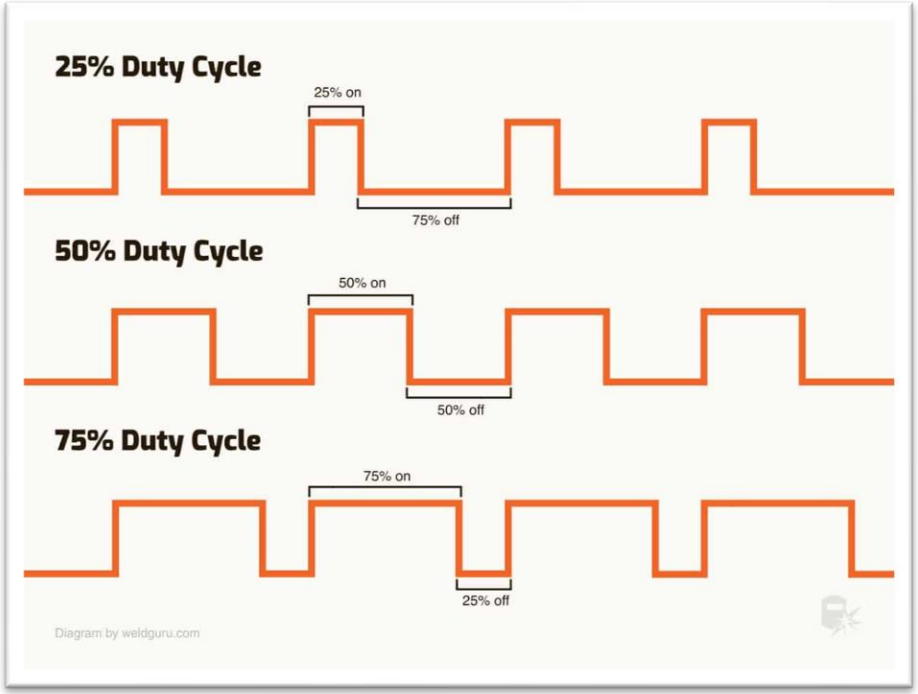


FREQUENZA (f, numero di cicli al secondo) e PERIODO (T, durata di un periodo)

f = 1 / T e quindi T = 1 / f

Esempi:
se il periodo è T = 1 sec, la frequenza è f = 1 Hz (Hertz)
se il periodo è T = 2 sec, la frequenza è f = 0,5 Hz

Duty cycle: rapporto tra la durata del segnale quando è "alto" e il periodo totale del segnale.
Esprime quant'è la parte di periodo in cui il segnale è alto.



Esercizi

- Qual è la frequenza di accensione nel codice blink? Qual è il periodo? Qual è il duty cycle (DC)?
- Modificare il codice per avere una frequenza di 1Hz con DC=50% (conviene calcolare il periodo T e ragionare su T)
- Modificare il codice per avere una frequenza di 1Hz con DC=75%
- Modificare il codice per avere una frequenza di 2Hz con DC=50%

Approfondimento

Si può immaginare che negli sketch Arduino vi sia una funzione main predefinita, come la seguente:

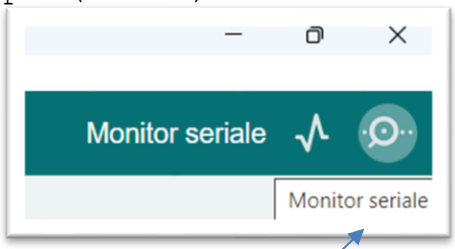
```
void main()
{
     setup();
     while(true) {
          loop();
     }
}
```

Una nota su setup e loop

Iniziare con Arduino (3): il monitor seriale

- Modificare il programma come segue:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
  Serial.begin(9600);       // open the serial port at 9600 bps (bauds)
  Serial.print("Inizio\n");
}

// the loop function runs over and over again forever
void loop() {
  Serial.print("Accendo\n");
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  Serial.print("Spengo\n");
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```



- Dare il comando di menu “Strumenti → Monitor Seriale” per aprire la finestra di output del sistema di comunicazione tramite la porta USB, oppure cliccare sull’icona in alto a destra.
- Cliccare su “Carica” (seconda icona sotto il menu, la freccia). Osservare la finestra del monitor seriale. Notare che il LED TX su Arduino lampeggia ad ogni messaggio inviato sulla porta seriale (USB).
- Nota importante: se il monitor seriale non funziona, controllare che la velocità impostata nella IDE (visibile a destra in alto nella finestra del monitor seriale) sia impostata a 9600 baud!

Iniziare con Arduino (4)

- Finora, abbiamo detto ad Arduino di accendere e spegnere un LED, e di mandare alcuni messaggi alla finestra del monitor seriale. Ora insegniamogli a contare. Immediatamente dopo la riga `Serial.print("Inizio\n")` inseriamo:

```
int j;
for (j = 0; j < 10; j++)
    Serial.print(j);
```

Concetto di variabile!

- Se la finestra del monitor di traffico seriale dovesse essere chiusa, dare il comando di menu "Strumenti → Monitor Seriale" per aprirla.
- Cliccare su "Carica" (seconda icona, la freccia). Osservare la finestra del monitor seriale.
- Informazioni su `Serial.print()` sono all'indirizzo:
  `https://www.arduino.cc/en/Serial/Print`

Il programma completo è nella prossima slide!

Iniziare con Arduino (4a)

- Ecco il programma:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
  Serial.begin(9600);      // open the serial port at 9600 bps (bauds)
  Serial.print("Inizio\n");

  int j;
  for (j = 0; j < 10; j++)
    Serial.print(j);
}

// the loop function runs over and over again forever
void loop() {
  Serial.print("Accendo\n");
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  Serial.print("Spengo\n");
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

- Dare il comando di menu “Strumenti → Monitor Seriale” per aprire la finestra di output del sistema di comunicazione tramite la porta USB, oppure cliccare sull’icona in alto a destra.
- Cliccare su “Carica” (seconda icona sotto il menu, la freccia). Osservare immediatamente la finestra del monitor seriale. Prima che inizi il funzionamento già visto, nella finestra sono scritti («stampati») i numero da 0 a 9.

Iniziare con Arduino (5)

- Ulteriore variante (da inserire nel rettangolo arancione della slide precedente):

```
int j;
for (j = 0; j < 10; j++) {
  Serial.print(j);
  Serial.print("\n");
}
```

- Osserviamo:
  - I punti e virgola alla fine di ogni riga di codice
  - Le parentesi graffe aggiunte all'istruzione for
  - Il funzionamento dell'istruzione for
  - Il costrutto j++
  - Il "carattere" \n ; si può anche tralasciare, e usare l'istruzione `Serial.println();` oppure integrare il "vai a capo" con la variabile da stampare: `Serial.println(j);` e quindi risparmiare una linea di codice.

Esercizio!

Modificare il codice in modo che i numeri siano scritti uno accanto all'altro, come nella slide precedente, ma alla fine il programma stampi un "a capo", così:

```
123456789
Accendo
Spengo
….
```

- Insegniamo ora ad Arduino a **calcolare i quadrati dei numeri interi**; introduciamo il programma che si trova alla slide seguente (attenzione; il codice è stato semplificato rimuovendo le istruzioni che fanno accendere e spegnere il LED! Ci interessa ora solo imparare o ripassare i costrutti del linguaggio C).

Iniziare con Arduino (6)

- Ecco il programma completo, semplificato rimuovendo la gestione del led:

```
// the setup function runs once when you press reset or power the board
void setup() {
  Serial.begin(9600);       // open the serial port at 9600 bps:
  Serial.print("Inizio\n");
  int j;
  for (j = 0; j < 10; j++) {
    Serial.print(j);
    Serial.print("\t");
    Serial.print(j*j);
    Serial.print("\n");
  }
}
// the loop function runs over and over again forever
void loop() {  // questa funzione e' vuota!
}
```

- Osserviamo l'output: una tabellina dei quadrati dei primi 10 numeri interi. Qual è lo scopo del carattere "`\t`", parente di "`\n`"?
- Come **cambiare i limiti della tabella?** Possiamo compilarla tra 0 e 50? Come? Che succede se portiamo il limite superiore a 200? Tragedia! Usare `long int`!

OVER/UNDERFLOW!

Iniziare con Arduino (7)

Se utile, vedere il file “posizionali e binario.ppt”

Tipi di dato in C

Tipi di dichiarazione	*Rappresentazione*	*N. di byte*	*Intervallo*
Bolean			True – False / On –Off / High - Low
Char	Carattere	1 (8 bit)	-128 +127
Byte	Carattere	1 (8 bit)	0 +255
Int	Numero intero	2 (16 bit)	-32.768 + 32.767
Unsigned int	Numero intero	2 (16 bit)	0 + 65.535
Short	Numero intero "corto"	2 (16 bit)	
Long	Numero intero "lungo"	4 (32 bit)	-2.147.483.648 +2.147.483.647
Float	Numero reale	4 (32 bit)	-3.4028235E+38 a + 3.4028235E+38
Double	Numero reale "lungo"	8 (64 bit)	$1.7976931348623157 \times 10^{308}$

Iniziare con Arduino (8)

- Come fare un ciclo che vada da 9 a 0? Usare j--.
- Scrivere tre versioni diverse del programma, in modo che Arduino:
  1. conti **da 0 a 9 e poi da 9 a 0** (duplicare il ciclo!).
  2. **ripeta indefinitamente** il conteggio da 0 a 9 e poi da 9 a 0 (spostare i cicli nella `loop()`!).
  3. faccia **lampeggiare il led per 10 volte con periodo di un secondo complessivo e duty cycle = 50%** (0,5s *on*, seguiti da 0,5s *off*), **e poi indefinitamente** con periodo di 200ms complessivi e duty cycle = 50%

Iniziare con Arduino (9)

- Riassunto istruzione **for**:

```
for (inizial; condiz; operaz)
  istruzione;

for (inizial; condiz; operaz) {
    istruzione1;
    istruzione2;
     ...
}
```

Uso delle parentesi graffe per definire un blocco di istruzioni!
Vedere anche «Zanichelli-Mirandola Arduino.pdf» pag. 12.

Iniziare con Arduino (10)

- Istruzione **if**:

```
if (espressione)  istruzione;

if (espressione) {
    istruzione1;
    istruzione2;
}

if (j == 5) {
    Serial.print(…
    digitalWrite…
}
```

Si veda anche il file "Zanichelli-Mirandola Arduino.pdf", pag 11; vedere anche la clausola "else"!

Esercizio: scrivere il codice che, utilizzando il led sul pin 13 e il serial port per contare da 1 a 10, sfrutti un test con l'istruzione if nel codice, in modo che Arduino, nell'ordine e una sola volta:

1. spenga il led onboard
2. conti da 1 a 10 (scrivendo i numeri via seriale con ritardo di 1 secondo) e, mentre conta….
    2a. quando giunge a 5,
        2a1. scriva via seriale "Sono arrivato a 5", e
        2a2. accenda il led on board.

Il codice sarà nella setup() mentre la loop() resterà vuota.

Iniziare con Arduino (10a)

Esercizi con l'istruzione for: **Kitt supercar**!

- Prerequisito: aver realizzato il circuito (reale o su Tinkercad), per **accendere/spegnere un singolo led collegato al pin 13**
- Compito: si prepari un circuito con **6 LED**, collegati ciascuno a un pin da 2 a 7 con la propria resistenza di limitazione della corrente, ad es. 560Ω), e si scriva il **codice che accende e spegne in sequenza i LED, prima dal n.2 al n.7 e poi viceversa**, con una durata della fase di accensione di ciascun LED pari a 0,5s.

Confrontare il codice con "Zanichelli-Mirandola Arduino.pdf", pag 12

Iniziare con Arduino (10b)

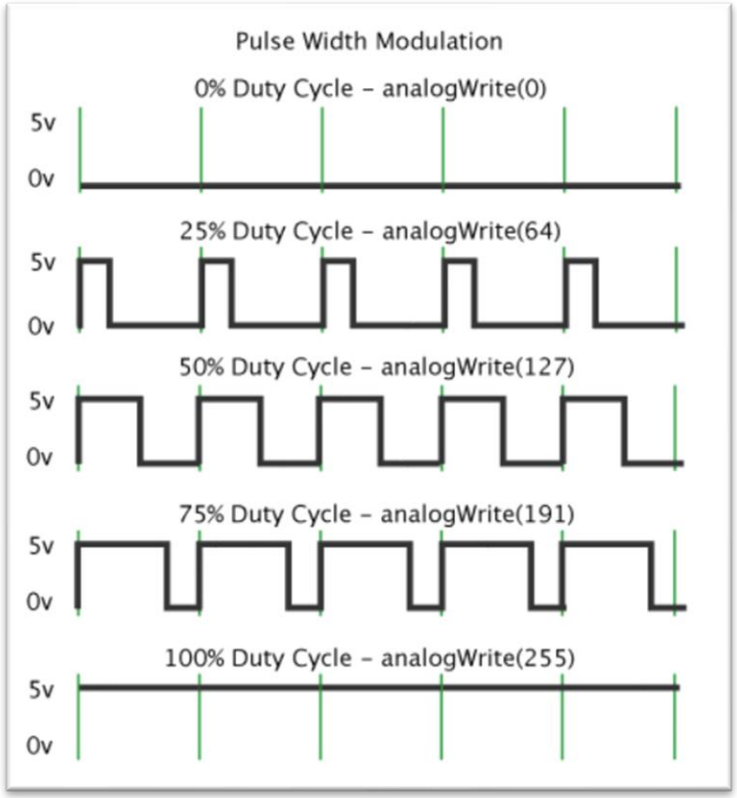
Esercizi con l'istruzione for: **Pulse Width Modulation (PWM)**!

- Compito: si prepari un circuito con 1 LED, collegato a un pin PWM (quelli con il simbolo ~) con la propria resistenza di limitazione della corrente, ad es. 560Ω).
- Si provi il codice seguente

```
// variazione della luminosità di un LED
int PWMpin = 9; // LED connected to digital pin 9

void setup() {
  pinMode(PWMpin, OUTPUT); // Set pin 9 as an output
}

void loop()
{
  analogWrite(PWMpin, 0);
  delay(1000);
  analogWrite(PWMpin, 128);
  delay(1000);
  analogWrite(PWMpin, 255);}
  delay(1000);
  analogWrite(PWMpin, 128);
  delay(1000);
}
```

Frequenza fissa (usualmente circa 490Hz)

DC da 0% a 100% per valori da 0 a 255



https://docs.arduino.cc/learn/microcontrollers/analog-output/

Iniziare con Arduino (10c)

Esercizi con l'istruzione for: **Pulse Width Modulation (PWM)**!

- Compito: si prepari un circuito con 1 LED, collegato a un pin PWM (quelli con il simbolo ~) con la propria resistenza di limitazione della corrente, ad es. 560Ω).
- Si modifichi il codice della slide precedente in base alle indicazioni qui sotto riportate:
  - Il codice, sfruttando un ciclo for, porti il led ad accendersi gradualmente dal minimo al massimo di luminosità, in un tempo complessivo di circa 1s, reiterando poi il ciclo; si sfrutti l'istruzione `delay()` con un valore calcolato in base al periodo di 1s complessivo.
  - Si modifichi il codice, duplicando il ciclo for, in modo che il led vada dapprima dal minimo al massimo della luminosità, e poi dal massimo al minimo, in un tempo complessivo di 2s.

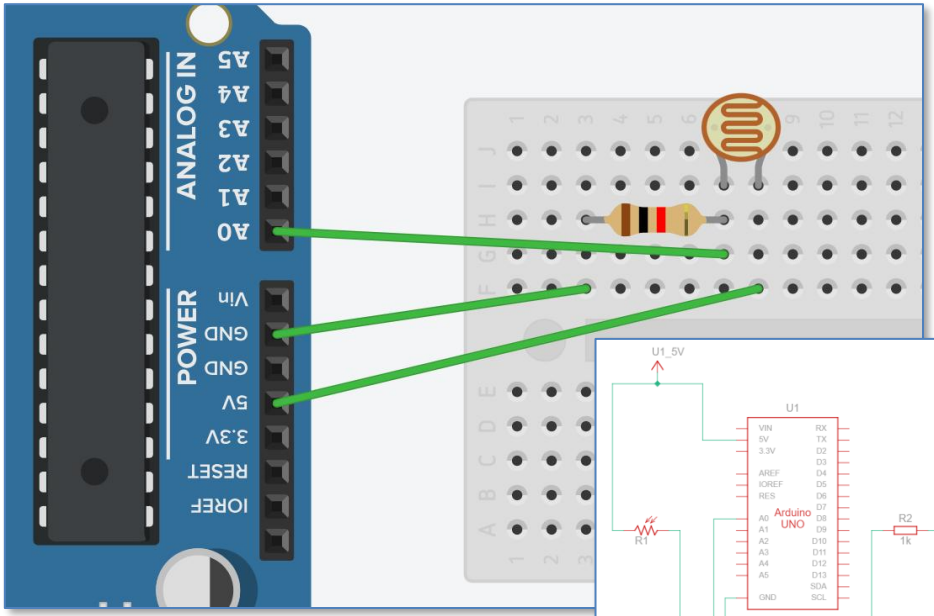
Iniziare con Arduino (10d)

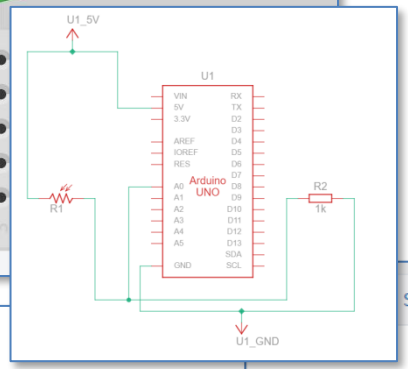
Esercizi con l'istruzione for: **Pulse Width Modulation (PWM)**!

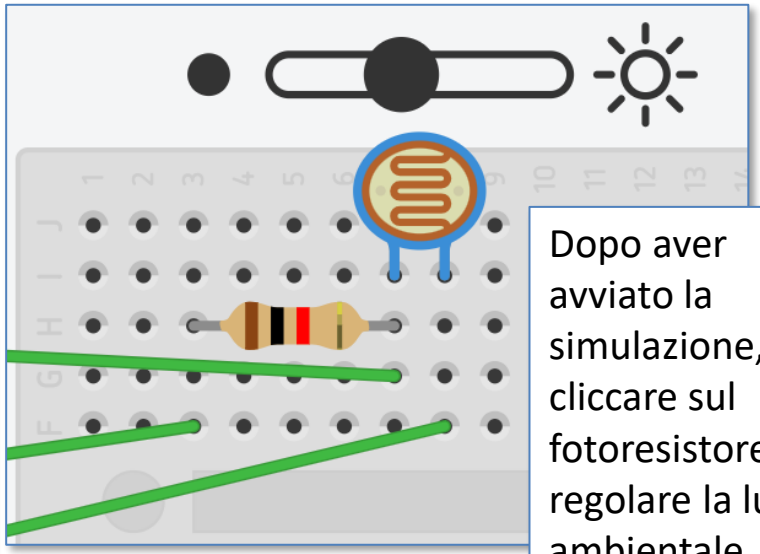
- Compito: si prepari un circuito con 1 LED, collegato a un pin PWM (quelli con il simbolo ~) con la propria resistenza di limitazione della corrente, ad es. 560Ω).
- Si provi il codice seguente

```
// variazione della luminosità di un LED
int PWMpin = 9; // LED connected to digital pin 9

void setup() {
  pinMode(PWMpin, OUTPUT); // Set pin 9 as an output
}

void loop()
{
  int x = 1;
  for (int i = 0; i > -1; i = i + x){
    analogWrite(PWMpin, i);
    if (i == 255) x = -1; // inverti la direzione agli estremi
    delay(10);
  }
}
```

Light Dependent Resistor (LDR) (fotoresistore)

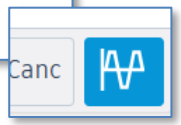






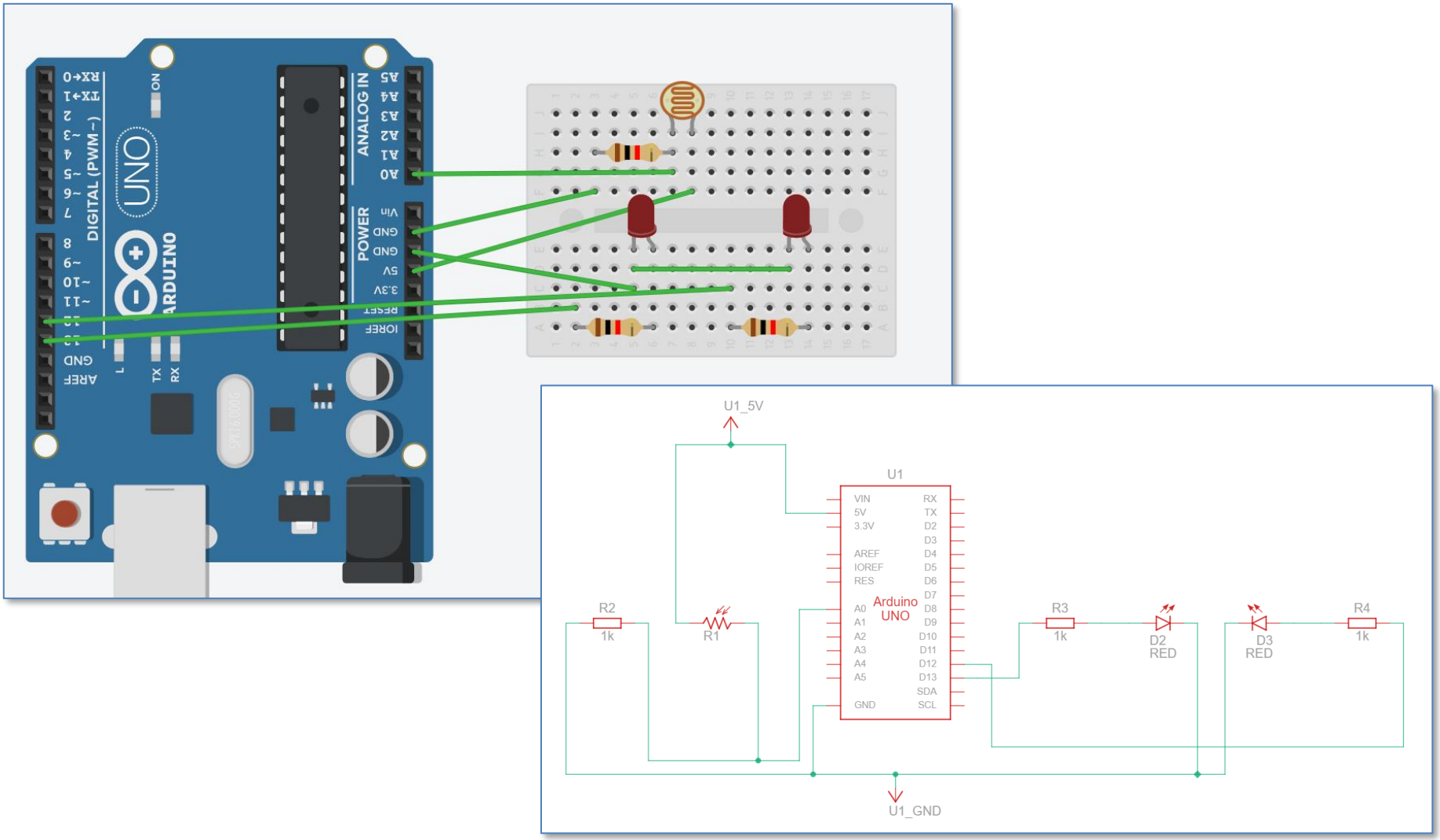
Dopo aver avviato la simulazione, cliccare sul fotoresistore e regolare la luce ambientale.

Durante la simulazione,con il codice visualizzato, cliccare su «monitor seriale» e sul simbolo del plotter seriale.

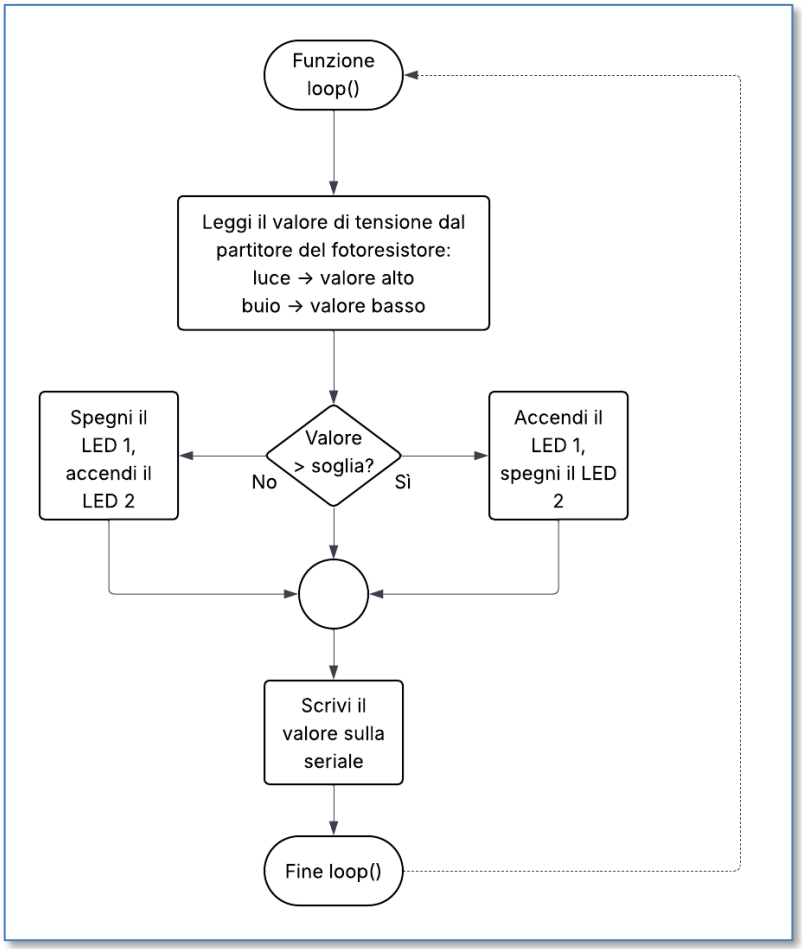



```
void setup() {
   Serial.begin(9600);
   pinMode(A0, INPUT);
}

void loop() {
  int sensorPin = A0; // select input for photocell
  int sensorValue;
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);
}
```

Light Dependent Resistor (LDR) (fotoresistore)

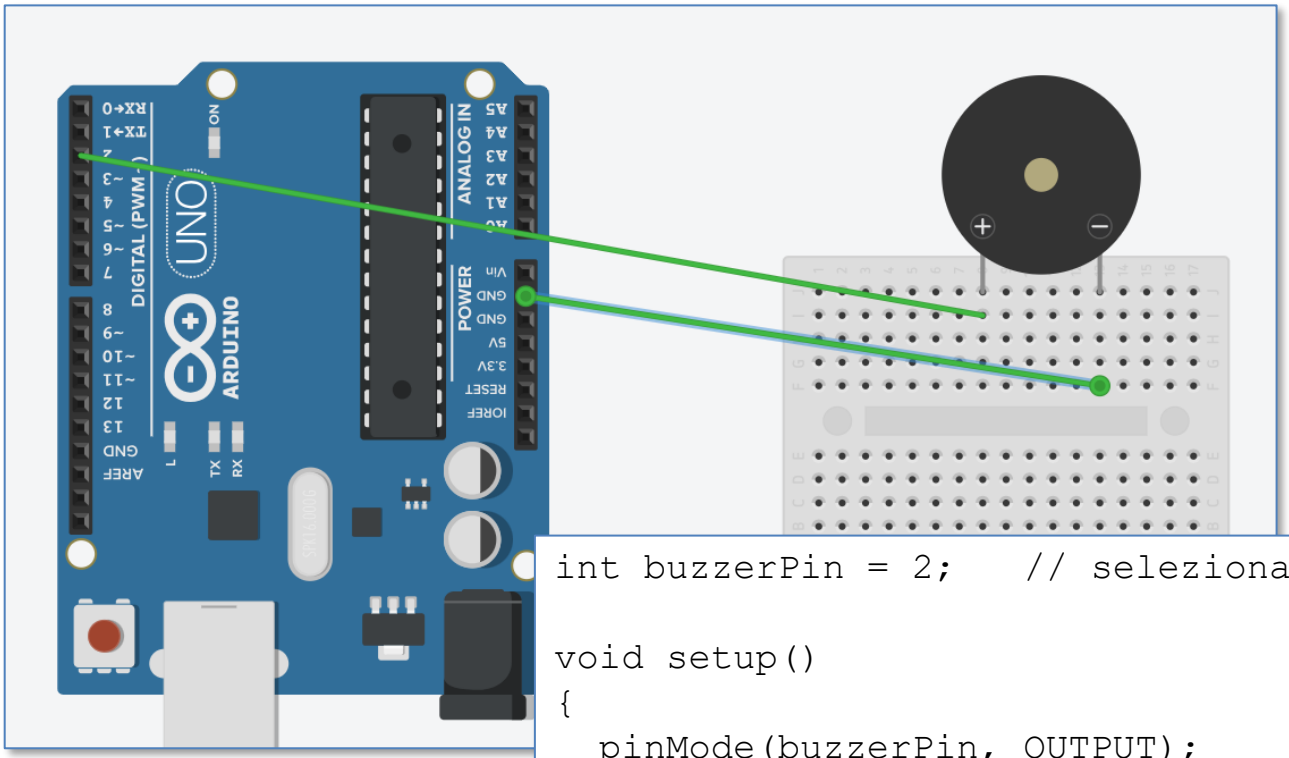


Light Dependent Resistor (LDR) (fotoresistore)



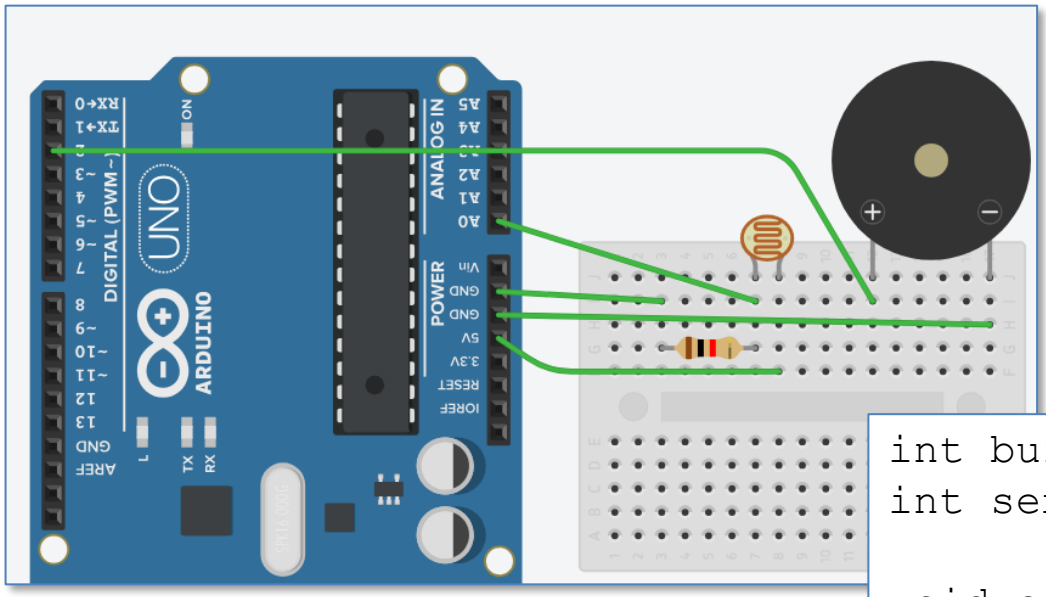

```
int ledPin1 = 12;    // select the pin for the LED
int ledPin2 = 13;    // select the pin for the LED
int sensorPin = A0;  // select input for photocell

void setup() {
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(sensorPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  int sensorValue;
  sensorValue = analogRead(sensorPin);
  if (sensorValue > 300) {
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, LOW);
  } else {
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, HIGH);
  }
  Serial.println(sensorValue);
}
```

Il *buzzer*

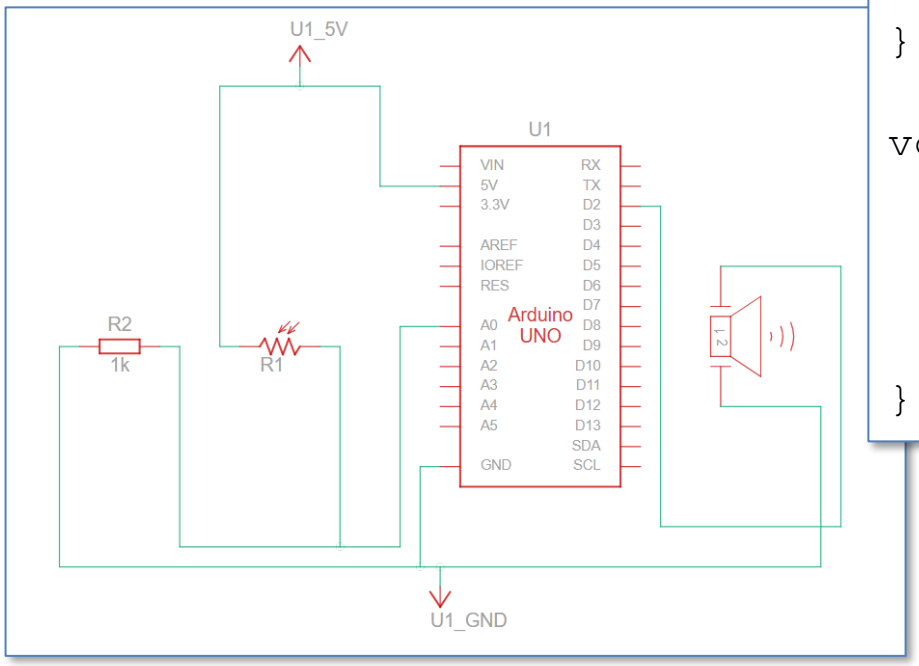



```
int buzzerPin = 2;    // seleziona il pin per il cicalino

void setup()
{
  pinMode(buzzerPin, OUTPUT);
}

void loop()
{
    tone(buzzerPin, 800, 100);
    delay(100);
}
```

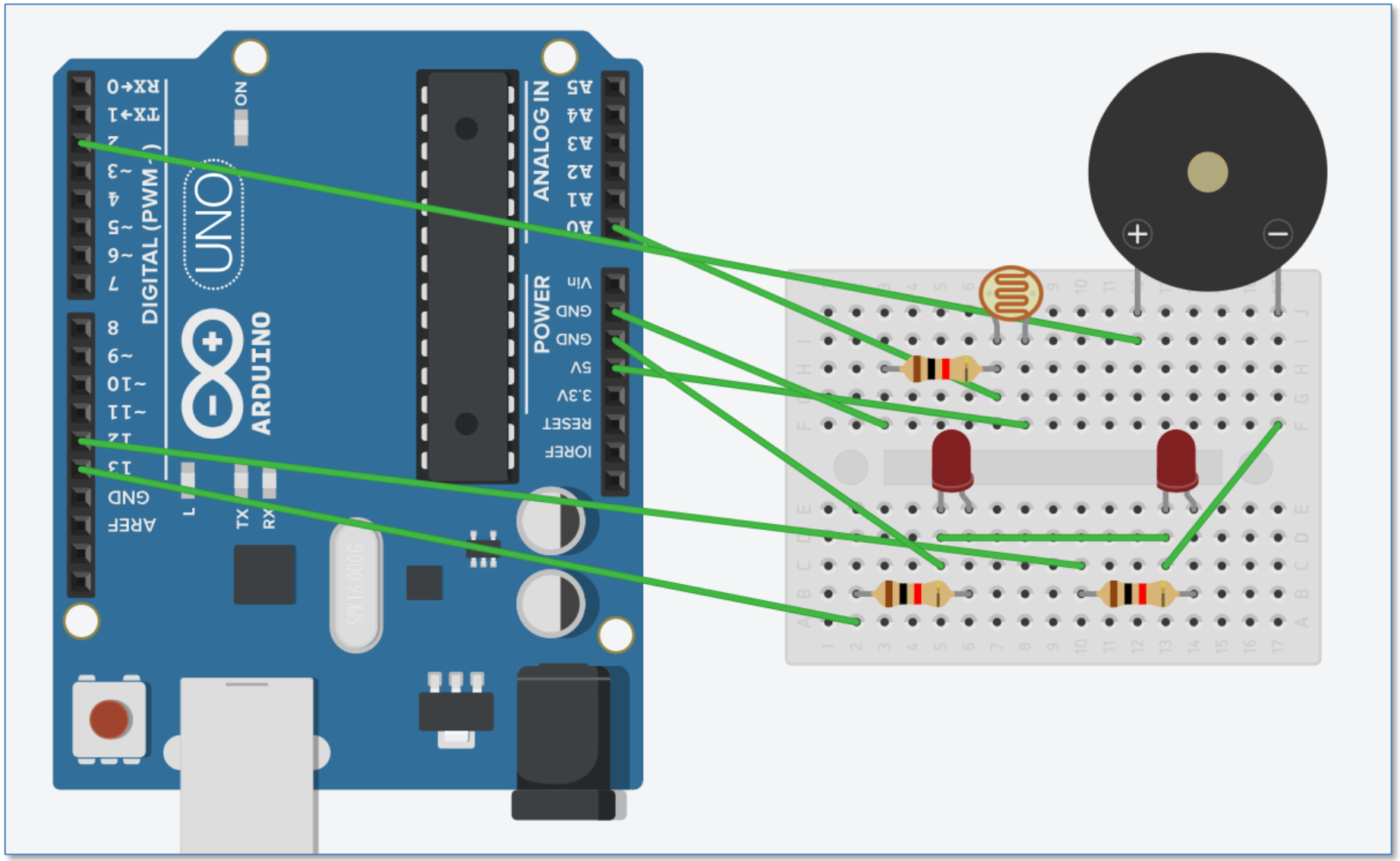
Buzzer + fotoresistore



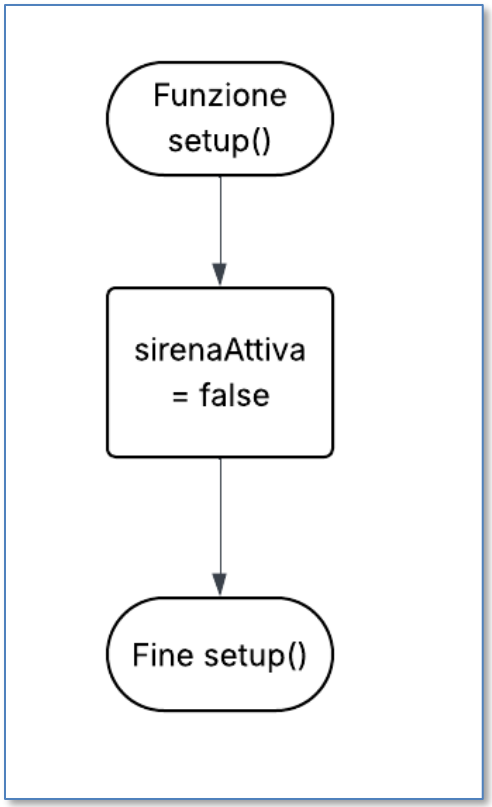

```
int buzzPin = 2; // pin cicalino
int sensPin = A0; // pin fotoresist

void setup() {
  pinMode(sensPin, INPUT);
  pinMode(buzzPin, OUTPUT);
}

void loop() {
  int sensValue;
  sensValue = analogRead(sensPin);
  tone(buzzPin, sensValue, 100);
  //delay(100);
}
```

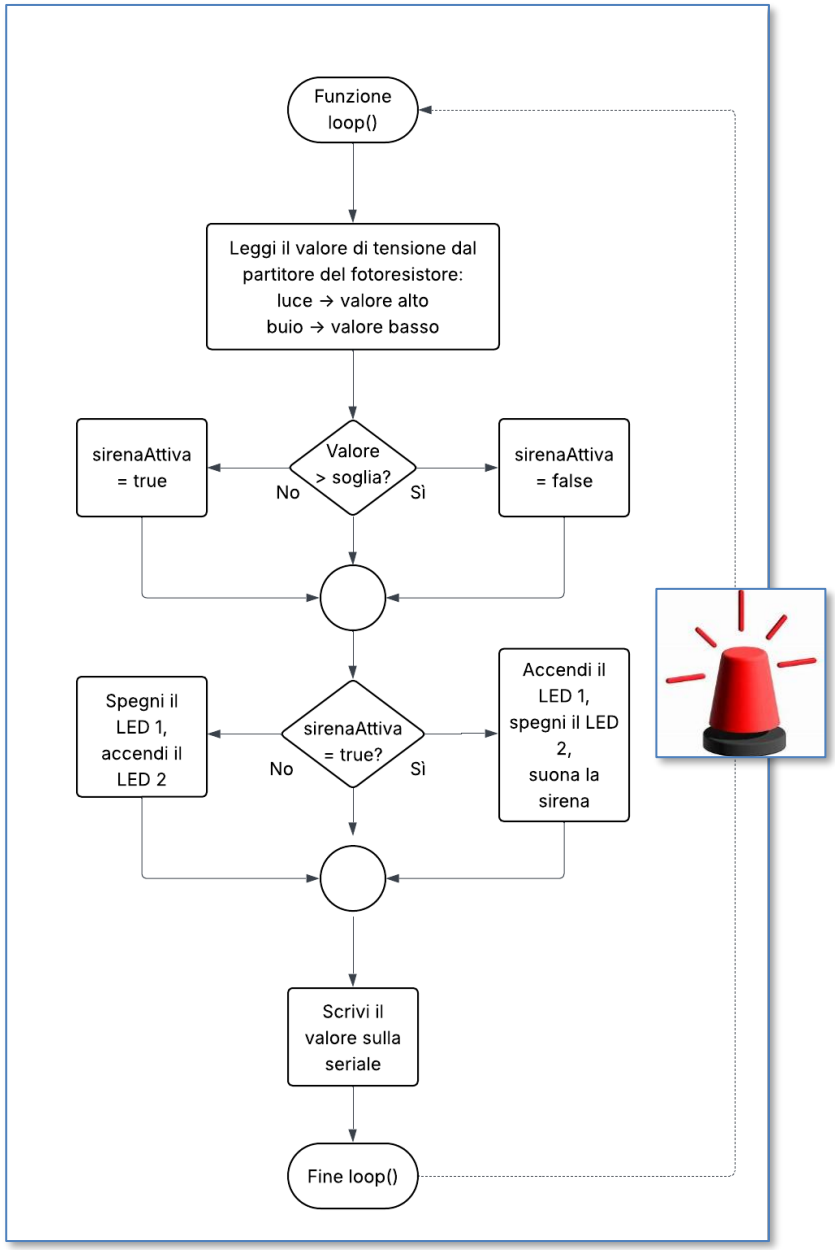


Antifurto



Antifurto


Funzione setup()
sirenaAttiva = false
Fine setup()


Funzione loop()
Leggi il valore di tensione dal partitore del fotoresistore:
luce → valore alto
buio → valore basso
Valore > soglia?
No
Sì
sirenaAttiva = true
sirenaAttiva = false
sirenaAttiva = true?
No
Sì
Spegni il LED 1, accendi il LED 2
Accendi il LED 1, spegni il LED 2, suona la sirena
Scrivi il valore sulla seriale
Fine loop()

```
int ledPin1 = 12;    // seleziona il pin per il LED
int ledPin2 = 13;    // seleziona il pin per il LED
int buzzerPin = 2;   // seleziona il pin per il cicalino
int sensorPin = A0;  // seleziona il pin per il fotoresistore

bool sirenaAttiva;   // status

void setup() {
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(sensorPin, INPUT);
  Serial.begin(9600);
  sirenaAttiva = false; // status "non attivo"
  digitalWrite(ledPin1, LOW); // spegni il LED
  digitalWrite(ledPin2, LOW);
  delay(10000); // aspetta un po'
}
```

```
void loop() {
  int sensorValue;
  sensorValue = analogRead(sensorPin);

  if (sensorValue > 300)
    sirenaAttiva = false; // "inattivo"
  else
    sirenaAttiva = true; // "attivo"

  if (sirenaAttiva == true) {
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, LOW);
    tone(buzzerPin, 800, 100);
    delay(100);
  } else {
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, HIGH);
  }
  Serial.println(sensorValue);
}
```

Iniziare con Arduino (11)

- FUNZIONI/FUNCTIONS

Studiare dapprima dal file "Zanichelli-Mirandola Arduino.pdf", pag 15

Esercizio: dopo aver ripetuto l'esempio dal testo citato (funzione "moltiplica"), scrivere una funzione che accenda e spenga una volta il led on board, con il periodo di un secondo.
Si richiami questa funzione prima dalla setup(), per vedere che il led compie un ciclo "accendi-spegni" una volta sola, e poi dalla loop(), per farla funzionare indefinitamente.

La funzione potrebbe chiamarsi blink1Time().

```
void blink1Time(void) { // do not forget to set pin 13 as output in setup
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(500);             // wait for half a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(500);             // wait for half a second
}
```

Iniziare con Arduino (11a)

- FUNZIONI/FUNCTIONS

Esercizio: scrivere una funzione che riceva due parametri: il numero di impulsi (n) e il periodo degli impulsi (t), e accenda/spenga il led n volte, ogni volta con periodo t (attenzione, il delay è t/2!). La funzione potrebbe chiamarsi blinkNTimes(int n, int t).

```
void blinkNTimes(int n, int t) { // don't forget to set pin 13 as output!
 int j;
 for (j = 0; j < n; j++)      {
         turn led on, wait for t/2 ms, turn led off, wait for t/2 ms
         // fill with the right commands!
    }
}
```

Usare questa funzione per realizzare un programma che esegua nell'ordine le seguenti operazioni:

1) chiami blinkNTimes per accendere/spegnere il led 5 volte con periodo di 2 secondi
2) faccia una pausa di un secondo
3) chiami blinkNTimes per accendere/spegnere il led 5 volte con periodo di 1 secondo
4) faccia una pausa di un secondo
5) chiami blinkNTimes per accendere/spegnere il led indefinitamente (quindi, una volta sola, ma inserendo la chiamata nella funzione loop!) con periodo di 1 secondo

Iniziare con Arduino (12)

Lettura dei tasti del pc

- Vedremo come **usare la tastiera** del computer per:
  - **accendere e spegnere il LED** situato sulla scheda di Arduino, e…
  - **cambiare il periodo di accensione** dello stesso LED
- A **cosa può servire**?
  - Per creare una **postazione di controllo dal proprio pc** per **gestire** qualsiasi **supporto elettronico** come luci, motori, pompe d'acqua per annaffiare, elettrodomestici…
- **Come funziona**?
  - legge il **tasto premuto sulla tastiera del pc** tramite seriale, con **Serial.read()**

Ispirato da http://www.marcopucci.it/tutorial-arduino-3-comandi-da-tastiera/

Iniziare con Arduino (13)

Lettura dei tasti del pc

Lettura del **tasto premuto sulla tastiera del pc** tramite seriale, con **Serial.read**(), e stampa a video con Serial.print() :

```
void setup() {
  Serial.begin(9600);            // open the serial port at 9600 bps:
  Serial.print("Inizio\n");
}

void loop() {
  char tasto;  // la variabile "tasto" conterra' un singolo carattere
               //   inviato da tastiera, quindi un "char"
  tasto = Serial.read();      // read a key from the keyboard
  Serial.print("Carattere inviato: ");
  Serial.println(tasto);
}
```

La IDE manda sulla connessione seriale (USB) i caratteri premuti; questi arrivano ad Arduino, che li legge con `Serial.read()` e poi li rimanda sulla connessione seriale con `Serial.print()`

Iniziare con Arduino (14)

Lettura dei tasti del pc

- Aprire il monitor seriale, con “Strumenti → Monitor Seriale” o con l’icona a forma di lente, in alto a destra
- Caricare il programma in Arduino: cosa si vede nella finestra del monitor seriale? (“Tasto premuto:” seguito da un carattere… poco stampabile! Arduino legge qualcosa anche quando non inviamo nulla.)
- Digitare una lettera nel campo di input in alto nel monitor seriale, e cliccare su **Send/Invia** oppure premere il tasto <invio>,
- Immediatamente si vedrà la lettera comparire nel flusso di output del programma.
- Digitando più lettere, o una frase, e poi premendo <invio> o <enter>, esse compariranno in sequenza.
- Se il programma stampa e va avanti troppo velocemente, inserire delay(500);

Iniziare con Arduino (15)

Lettura dei tasti del pc, prima variante

Lettura del **tasto premuto sulla tastiera del pc** tramite seriale, con **Serial.read**(), e stampa a video con Serial.print() **solo dei caratteri corrispondenti a lettere minuscole**:

```
void setup() {
  Serial.begin(9600);            // open the serial port at 9600 bps:
  Serial.print("Inizio\n");
}

void loop() {
  char tasto;
  tasto = Serial.read();       // read a key from the keybord
  if (tasto >= 'a' && tasto <= 'z') {  // notare: apici singoli (sono
                                       // caratteri!) e non doppi (usati
                                       // per le stringhe!)
    Serial.print("Carattere inviato: ");
    Serial.println(tasto);
  }
}
```

Iniziare con Arduino (16)

Lettura dei tasti del pc, prima variante; note al codice

```
void setup() {
  Serial.begin(9600);           // open the serial port at 9600 bps:
  Serial.print("Inizio\n");
}
void loop() {
  char tasto;
  tasto = Serial.read();      // read a key from the keybord
  if (tasto >= 'a' && tasto <= 'z') {      // notare: apici singoli (sono caratteri!)
                                           //  e non doppi (usati per le stringhe!)
    Serial.print("Carattere inviato: ");
    Serial.println(tasto);
  }
}
```

Codice uguale a quello della slide precedente

I caratteri da **'a'** fino a **'z'** sono in realtà dei numeri interi piccoli, secondo il codice ASCII (slide seguente), e possono essere inseriti in operazioni di confronto.

Notare gli apici singoli, usati per i singoli caratteri come in **'a'**, invece degli apici doppi come in **"Inizio"** usati per le stringhe (parole). Si tratta, inoltre, del carattere **'** e non **‘ che spesso esce da un copia-incolla!!**

Notare l'uso di && per verificare contemporaneamente due condizioni:
tasto >= 'a' && tasto <= 'z' equivale a **'a' ≤ tasto ≤ 'z'** che non è sintassi corretta in C.

logica!

Provare, dopo aver consultato la tavola dei codici ASCII, condizioni diverse più articolate, come:
(tasto >= 'a' && tasto <= 'z') || (tasto >= 'A' && tasto <= 'Z') || (tasto >= '0' && tasto <= '9') || (tasto == '-')

ASCII table

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	Del

asciichars.com

Iniziare con Arduino (17)

- Istruzione **switch** / **case**:

Si veda anche il file "Zanichelli-Mirandola Arduino.pdf", pag 11

```
switch (variabile) {
  case 1:
      //blocco di istruzioni da eseguire quando variabile == 1
      break;
  case 2:
      // blocco 2 di istruzioni da eseguire quando variabile == 2
      break;
  default:
      // se variabile non corrisponde ai casi elencati, eseguire queste istruzioni
      // (il caso default è opzionale)
}
```

Iniziare con Arduino (18)

Lettura dei tasti del pc: Modificare come segue; cosa fa il codice?

```
void setup() {
  pinMode(13, OUTPUT);
  Serial.begin(9600);            // open the serial port at 9600 bps:
  Serial.print("Inizio\n");
}

void loop() {
  char tasto;
  tasto = Serial.read();       // read a key from the keybord
  switch (tasto) {
    case 'a':
      Serial.print("Inviato carattere a\n");
      digitalWrite(13, HIGH);
      break;
    case 's':
      Serial.print("Inviato carattere s\n");
      digitalWrite(13, LOW);
      break;
    default:
      Serial.flush();
  }  // switch
}
```

Esercizio: modificare il codice in modo che, oltre ad accendere e spegnere il led con i tasti a/s, usi il tasto *b* per far "blinkare" (accendere/spegnere) il led 10 volte con periodo 1 s (usare la funzione blinkNTimes realizzata in precedenza, slide "Iniziare con Arduino (11a)"!)

Iniziare con Arduino (19)

Lettura dei tasti del pc

Esercizio… difficile: modificare il codice in modo che sfrutti l'input da tastiera per modificare il periodo di blinking del led.

Suggerimento
nell'ordine:
• dichiarare una variabile t, che conterrà il periodo del blinking e partirà per esempio da 1000; la variabile t dev'essere globale (dichiarata fuori da tutte le funzioni) o `static`, dichiarata nella stessa `loop()`, perché così conserverà il valore tra una chiamata e l'altra di loop
• usare un costrutto switch/case per incrementare t di 200 quando si preme e si invia il carattere '+', e decrementarla di 200 quando si preme e si invia il carattere '-'.
• chiamare poi la funzione `blinkNTimes(1, t)` .

Attenzione: occorre prevedere dei test per evitare che t diventi nulla o negativa, ed eventualmente anche che diventi troppo grande (per esempio, t >200 e t < 2000, sottraendo o sommando 200 a t solo se le due condizioni sono rispettivamente soddisfatte).

Iniziare con Arduino (20)

Istruzione DO…WHILE

Si veda il file "Zanichelli-Mirandola Arduino.pdf", pag 13

```
void setup() {
  Serial.begin(9600);
  Serial.print("Inizio\n");

  int i = 9;
  do {
    Serial.println(i);
    i = i - 1;   // oppure i--
  } while (i >= 0);

  Serial.print("Fine\n");
}  // setup

void loop() {
}
```

Programma che conta da 9 a 0, scrivendo i numeri via seriale, con un ciclo do..while anziché con il for

Costrutto alternativo all'uso di for: il ciclo do..while esegue un blocco di istruzioni e poi fa un test, tornando ad eseguire il blocco solo se il test è verificato.

Si può uscire da un ciclo do…while con l'istruzione break

Iniziare con Arduino (21)

Istruzione DO…WHILE

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  float somma = 0;  // float perche' dopo
                    // dividiamo per 2
  int n = 0;
  char tasto;
  do {
    tasto = Serial.read();
    if (tasto >= '0' && tasto <= '9') {
      somma = somma + (tasto - 48);
      n++;
    } //  if
  }  // do
  while (tasto != '-');
  Serial.println(somma/n);
}
```

Programma che calcola la media dei numeri di una cifra introdotti da tastiera (media dei voti).

Una volta lanciato il programma, si possono introdurre dei numeri (di una cifra) nella finestra del monitor seriale, seguiti dal carattere '-'.
Il carattere '-' fa da "tappo", ossia conclude l'inserimento dei numeri.
Il programma somma i numeri, e fornisce in output (con il monitor seriale) la media dei numeri introdotti.

A cosa serve la riga colorata in celeste?

Iniziare con Arduino (22)

```
void setup() {
  Serial.begin(9600);
  Serial.print("Inizio\n");
}  // setup

void loop() {
  if (Serial.available()) {
    char tasto = Serial.read();
    Serial.print(tasto);
  }
}
```

Programma che legge i caratteri inviati da tastiera, controllando dapprima (con Serial.available()) che effettivamente ci siano caratteri da leggere

Evita che vengano stampati caratteri senza senso quando non si invia nulla. Può in certi casi sostituire i test sui limiti dei valori ASCII dei caratteri (commentando con // la riga che contiene il Serial.available, e la riga con la corrispondente parentesi graffa chiusa, si ritorna alla situazione in cui sono stampati a video i caratteri strani…)

Iniziare con Arduino (23)

```
void setup() {
  Serial.begin(9600);
  Serial.print("Inizio\n");
}  // setup

void loop() {
  if (Serial.available()) {
    char tasto = Serial.read();
    Serial.print(tasto);
    Serial.print('\t');
    int tastoInt = tasto;
    Serial.println(tastoInt);
  }
}
```

Programma che legge i caratteri inviati da tastiera, controllando dapprima (con Serial.available) che effettivamente ci siano caratteri da leggere, e stampa sia il carattere che il suo codice ASCII

Iniziare con Arduino (24)

Esercizi in linguaggio C, difficoltà bassa/media

1) Riprendere il programma che conta da 0 a 9, inviando il risultato tramite il canale di comunicazione seriale; una volta verificato che funzioni, modificarlo in modo che conti solo numeri pari, compreso lo zero, da 0 a 20, e scriva quindi i numeri 0 2 4 … 20; per farlo, sostituire all'istruzione j++ l'istruzione j = j + 2, e poi provare anche con il costrutto j+=2. [uso di for]
2) Scrivere un programma che conti da 0 a 10 e scriva, via comunicazione seriale, una tabellina dei numeri e delle loro radici quadrate. Ispirarsi a "Iniziare con Arduino (6)", capendo bene come funziona, e considerare che la radice quadrata di un numero si calcola con sqrt(x). [uso di for]
3) Come al punto precedente, però quando il numero che state considerando è un quadrato perfetto, fate scrivere "quadrato perfetto" dopo il valore della radice quadrata; per testare se un numero è un quadrato perfetto, usare if (sqrt(x) == floor(sqrt(x))) che controlla se la radice quadrata del numero è pari alla sua parte intera; meglio sarebbe mettere la radice quadrata di x in una variabile, per non ripetere il calcolo: float s = sqrt(x); if (s == floor(s))... [uso di for e if]
4) Scrivere un programma composto di setup()/loop() e di una funzione void pippo() che, una volta richiamata da loop(), stampi sul canale seriale la frase "Ciao sono Pippo". Modificare il programma in modo che la funzione pippo abbia un parametro: void pippo(int x) ; una volta richiamata da loop(), stampi "Ciao sono Pippo" e poi "Il parametro vale" seguito dal numero x [function]
5) Scrivere un programma che invii sul canale di comunicazione seriale un numero casuale, generato nella funzione loop(); un numero casuale tra 50 e 100 (per esempio) si genera con int randNumber = random(50, 100); notare che la sequenza di numeri casuali è fissa (si tratta di numeri pseudocasuali) e può essere variata dando inizialmente il comando randomSeed(millis()); Infine, fare in modo che il programma, quando sceglie un numero situato tra 70 e 90, avvisi esplicitamente sul canale seriale, e faccia lampeggiare il led alcune volte.
6) Come al punto 4, però la funzione pippo() (versione senza parametro) deve generare numeri casuali, e stamparli tramite canale seriale
7) Come al punto 6, però la funzione pippo(), dopo aver scritto a video il numero casuale, deve dire anche se è pari o dispari; un numero m è pari se m%2==0.

Iniziare con Arduino (25)

Esercizi in linguaggio C, difficoltà medio-alta

1) Scrivere una funzione, chiamata leggiCifraDallaSeriale(), che – una volta chiamata – legga caratteri dalla tastiera e torni solo quando il carattere corrisponde a una cifra tra 0 e 9; restituisca questa cifra, trasformata in numero (sottraendo 48 che equivale allo 0). Il programma principale, nella funzione setup(), chiamerà due volte la leggiCifraDallaSeriale() scrivendo via seriale i due numeri letti
2) Sfruttare la funzione leggiCifraDallaSeriale scritta al punto precedente, per scrivere un programma che legga tre numeri dalla seriale, e li usi come i coefficienti di una equazione di secondo grado, ne calcoli le due soluzioni se il delta è >= 0, e restituisca dette soluzioni; se il delta è negativo, restituisca la frase “Delta negativo, non ci sono soluzioni reali”; considerare che i calcoli devono essere fatti con variabili di tipo float, alle quali si possono inizialmente assegnare i valori interi letti con la leggiCifraDallaSeriale(); considerare poi la funzione sqrt() che restituisce la radice quadrata di un numero; se la variabile a contiene un numero intero, float s = sqrt(a2); ne calcola la radice quadrata.
3) Scrivere un programma che, dati i due cateti di un triangolo rettangolo, che supponiamo essere numeri piccoli ad una cifra, introdotti tramite il monitor seriale come ai punti precedenti con la funzione leggiCifraDallaSeriale, stampi in uscita tramite il monitor seriale l’ipotenusa, l'area e il perimetro del triangolo stesso;
4) Scrivere un programma che, dopo aver letto dall'utente un numero intero con leggiCifraDallaSeriale, stabilisca se e' pari o dispari; per farlo, considerare che l’espressione c % 2 restituisce il resto della divisione di c per 2, quindi se il resto è 0 il numero è pari, se è 1 il numero è dispari: if (c % 2 == 0){…

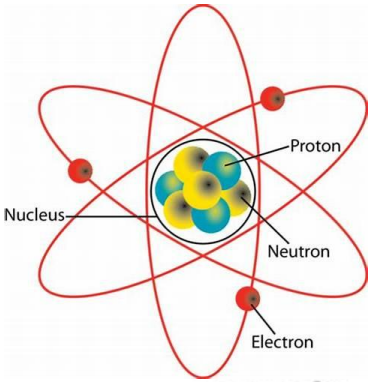
Iniziare con Arduino (26)

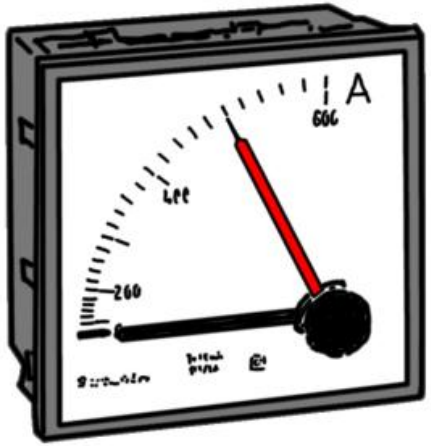
```
void setup() {
  Serial.begin(9600);
  Serial.print("Inizio\n");
  int a1, a2;
  a1 = leggiCifraDallaSeriale();
  Serial.println(a1);
  a2 = leggiCifraDallaSeriale();
  Serial.println(a2);
}  // setup

void loop() {
}

int leggiCifraDallaSeriale() {
  char tasto;
  do {
    tasto = Serial.read();
  } while (tasto < '0' || tasto > '9');
  int valore = tasto - 48;
  return valore;
}
```

Programma che legge due numeri piccoli (due numeri di una cifra) inviati da tastiera, ristampandoli tramite il canale di comunicazione seriale

Nota: potremmo usare solo una int tasto.

Corrente e flusso d'acqua (1)



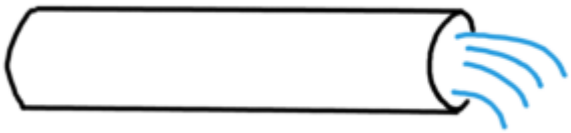


Corrente elettrica: paragonabile a un flusso d'acqua.

Si genera per il **movimento degli elettroni in un materiale conduttore** come il rame o il ferro in cui gli elettroni possono muoversi facilmente.

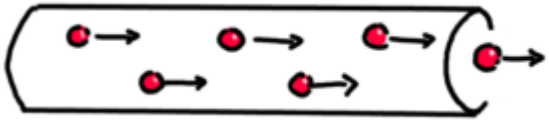
Gli elettroni creano un **flusso** e possono essere visualizzati come un **flusso d'acqua che scorre in un tubo**.

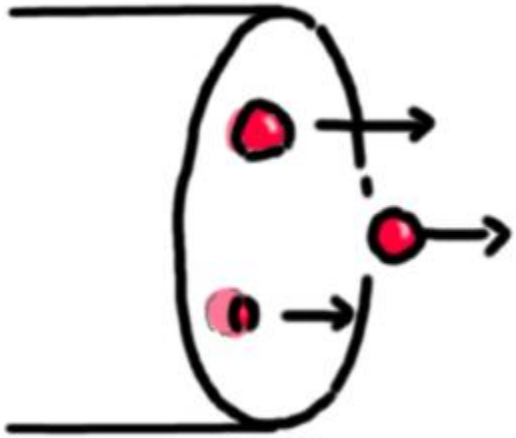
I **fili di un circuito elettrico** possono essere immaginati come dei tubi in cui scorre l'acqua.



ATTENZIONE: si tratta di una rozza **semplificazione**!!!

Corrente e flusso d'acqua (2)





La **corrente** in un "conduttore" è la **quantità di elettroni che passano in un filo elettrico** (o meglio da una **sezione del filo**) **in un secondo**.
Paragone con l'acqua: quantità di acqua che passa nel tubo in un secondo.

Unità di misura è **l'Ampère** (da André-Marie Ampère, 1775–1836).

L'Ampere si indica con la lettera **A** (maiuscola)

Esempi "pratici":
treno/tram 100/500A
Forno elettrico alcuni A
lettore mp3, radiosveglia 0.01-0.3A

Quando la corrente è molto piccola si può esprimere in **milliAmpère (mA),** cioè un millesimo di Ampère

Le misure di corrente si eseguono con uno strumento chiamato **Amperometro**.

Tensione e cascate (1)

“A quale **tensione** funziona quest’elettrodomestico?” “A 220 volt”.
“Che batteria devo mettere nella radiosveglia?” “Una pila da 9 volt”.
“Qual è la tensione fornita dalla batteria dell’auto?” “12 volt”

Unità di misura: il **volt** (indicato con la lettera V maiuscola), dal nome del conte, e scienziato, **Alessandro Volta** (1745 – 1827): invenzione della pila (e scoperta del metano…). **Multipli e sottomultipli del V: milliVolt (mV), kiloVolt (kV)… Lo strumento** che misura la tensione è **il Voltmetro.**

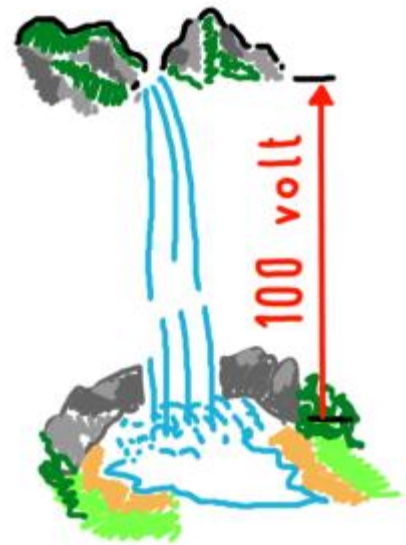
Tensione e cascate (2)

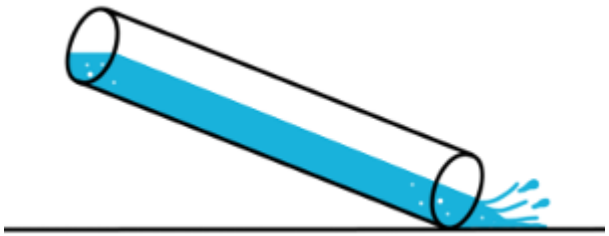
La **metafora dell'acqua**!

L'acqua scorre se c'è un **dislivello**.
La **tensione elettrica** è **l'equivalente del dislivello** che deve esserci per far **scorrere dell'acqua** in un tubo: maggiore è l'altezza e maggiore è la velocità, e dunque l'energia, con cui l'acqua arriva in fondo.

Se prendiamo un **tubo molto lungo**, lo **posiamo a terra** e lo riempiamo d'acqua, questa uscirà dall'altra estremità lentamente, con **poca energia**.
Se ora **solleviamo una delle estremità**, l'acqua uscirà con **maggior energia.**

La tensione si può immaginare come il dislivello da cui cade l'acqua.



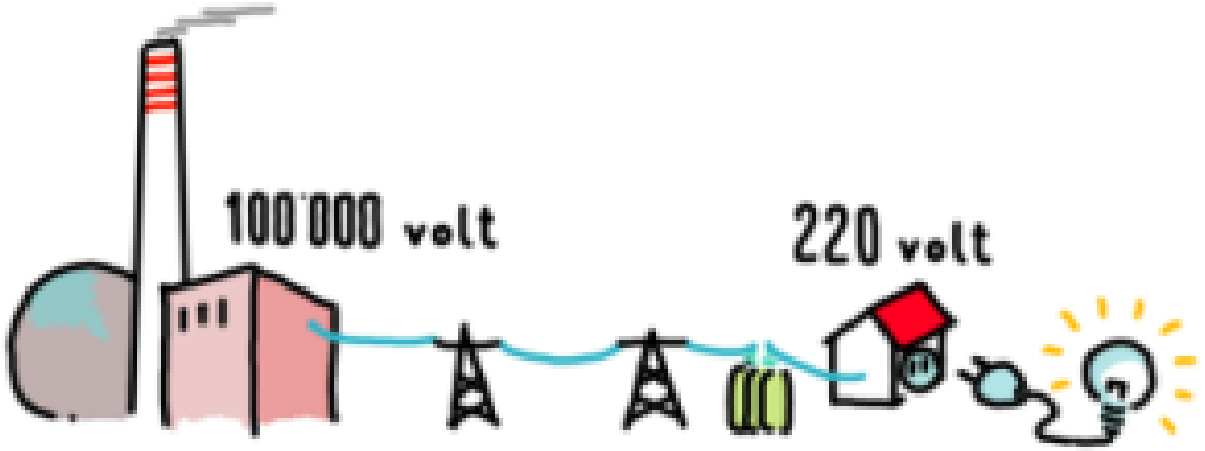


Tensione e cascate (3)

Se vogliamo usare un **tubo molto lungo**, per far uscire l'acqua con un certo **vigore**, è necessario che la **differenza d'altezza tra le estremità del tubo sia notevole**.

E' quello che accade nelle **centrali elettriche che generano correnti con alto voltaggio** (anche **centinaia di migliaia di volt**), per poi immetterla negli elettrodotti che percorrono **centinaia di chilometri**.

Al termine dell'elettrodotto la tensione **è ridotta** prima di essere portata nelle case (a 220 V) o nelle industrie.



Tensione e cascate (4)

La **tensione** della corrente elettrica nelle nostre case è di circa **220 V** in **corrente alternata** ed è **direttamente** impiegata per gli **elettrodomestici più grandi**.

Piccoli elettrodomestici necessitano di una "cascata" ancora più piccola: interponendo dei circuiti elettrici detti "**alimentatori**" si scende usualmente a **12, 9** o **5 V**. In più, la corrente in uscita è **continua** (come quella delle pile e batterie).

Nelle prese **USB** troviamo **5 volt** utilizzabili per alimentare piccoli dispositivi.

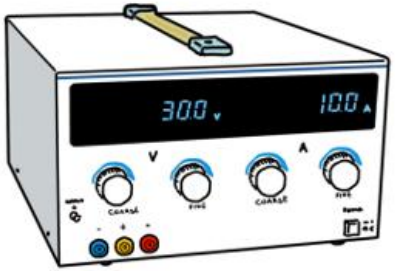
Maggiore è la tensione, maggiore è la pericolosità (in realtà conta la **combinazione di corrente e tensione**, ossia il loro **prodotto** che dà la **potenza**).

Le **tensioni elevate** possono superare gli ostacoli (o gli isolamenti) in modo più facile: per questo **sono più pericolose**.

Tensione e corrente (1)

Perché un **circuito** o un dispositivo elettrico **possa funzionare** è necessario collegarlo a "qualcosa" in grado di "**fornire la corrente**".

Può essere un **alimentatore o un gruppo di batterie**: in generale, un **generatore**, che genera una certa **tensione** (alle volte indicata come **differenza di potenziale**, sebbene i concetti e i contesti siano diversi) ed eroga una **corrente**.



La tensione è una "forza" che spinge gli elettroni a muoversi, dando origine alla corrente (flusso ordinato di cariche).

Per **evitare di danneggiare** il dispositivo, è necessario verificare che:
– la **tensione** fornita dal generatore sia corretta (ossia quella richiesta dall'apparecchiatura): né maggior né minore (con piccola tolleranza)
– ci sia sufficiente **corrente** affinché il dispositivo funzioni.

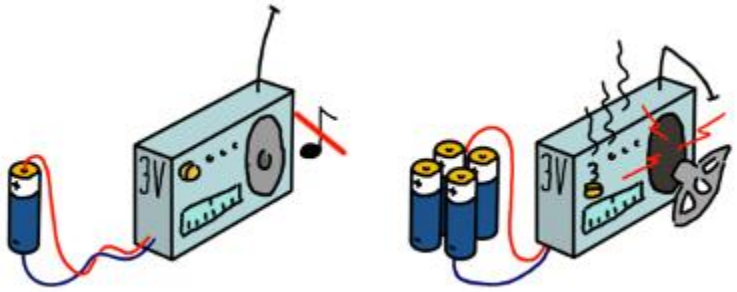
Tensione e corrente (2)

Immaginiamo che il **generatore** sia una **piccola cascata** d'acqua e che il **circuito sia un mulino** (una ruota):
– se la **ruota è troppo grande**, la cascata **non riuscirà** a coprire la ruota del mulino e a farla ruotare
– se la **cascata è troppo alta** e la **ruota** del mulino è **molto piccola**, la caduta dell'acqua **danneggerà** la ruota.

(L'eventuale danneggiamento dipende anche dalla presenza di circuiti di protezione…)

Se la **tensione** dell'alimentatore è **inferiore alla tensione richiesta** dal circuito, il circuito (probabilmente) **non funziona**. Se è **superiore**, è possibile che il circuito **resti danneggiato**.

Dispositivo elettrico che richiede di essere alimentato con 3 batterie da 1.5 V:



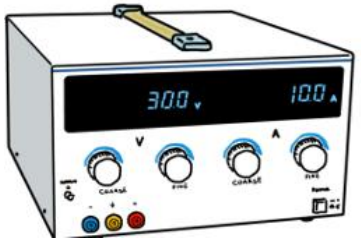
Tensione e corrente (3)

Immaginate di essere sulle rive di un fiume. **Immergete un tubo nel fiume** e prelevate dell'acqua per far **girare il mulino**. Il tubo si riempirà di acqua che farà quindi **girare la ruota** del mulino.

Ora immaginiamo di **prelevare con un tubo l'acqua** da un **rivolo** che scorre pigramente… Il **tubo farà fatica a colmarsi d'acqua**. La **ruota** del mulino **non ruoterà** perché non c'è abbastanza acqua.

In conclusione, **se al circuito forniamo poca corrente**, il circuito **non funzionerà** o funzionerà in modo non corretto.

Tensione e corrente (4)



Con un **alimentatore regolabile** si può impostare a piacimento **tensione e corrente** massima (nei limiti dello strumento).

- **Colleghiamo** un **circuito** che funziona a **5 V** e che ha bisogno di **1 A**.

- **Impostiamo l'alimentatore su 5 V e 0 A**. Il **circuito** è **spento** perché non gli arriva "acqua". Il tubo pesca da un torrente in secca (0 A).

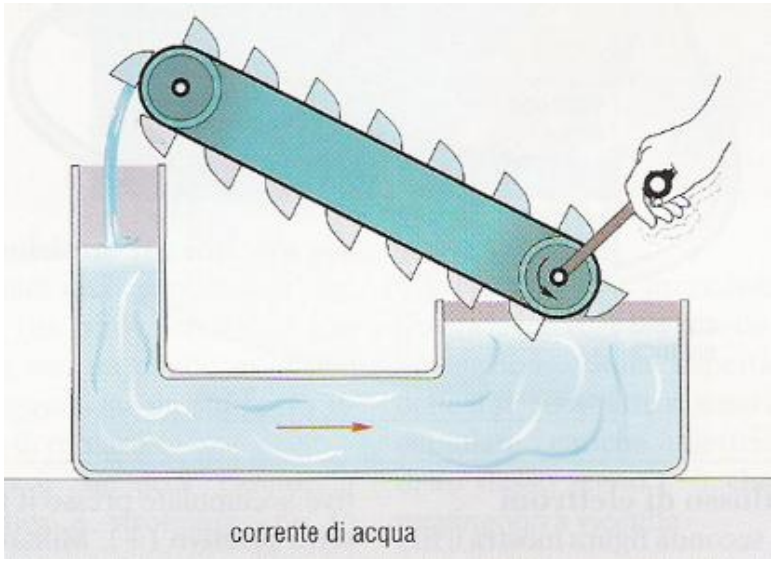
- **Alziamo la corrente: 0.5 A**. Il circuito forse **si accende**, ma **non funziona ancora bene**. Il tubo pesca in un piccolo torrente, insufficiente a riempirlo.

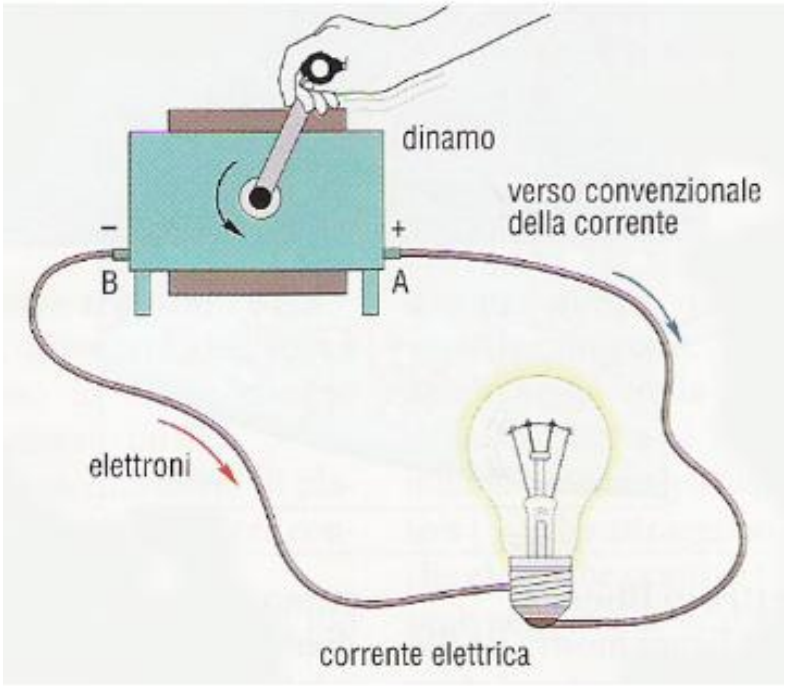
- Ora **impostiamo esattamente 1 A** e il circuito **funziona**: la condotta pesca in un fiume e il tubo si riempie a dovere.

- Infine **aumentiamo la corrente a 15 A**. Esplode tutto? No! È un limite superiore! Nel circuito passa comunque solo la corrente di cui esso ha bisogno per funzionare.

- Se invece **portassimo la tensione a 7 volt**… allora il circuito **potrebbe danneggiarsi**!

Tensione e corrente (5)





Generatore

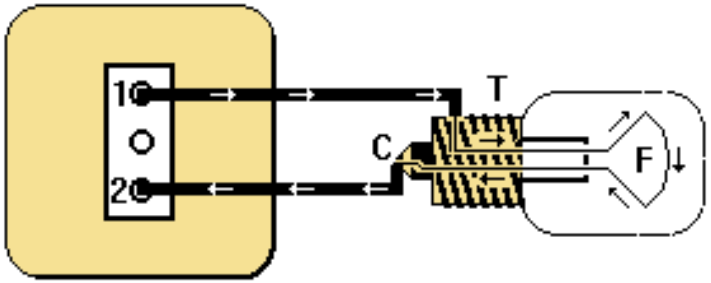
http://www.itchiavari.org/chimica/lab/condelet.html

Tensione e corrente (6)

Riassunto/Esempio

Tra i due terminali di una presa di corrente (fori esterni), o di **una pila**, c'é **tensione**, ovvero una "forza" che spinge gli elettroni a muoversi, dando origine alla **corrente (flusso ordinato di cariche)**.
La tensione nelle prese di casa è pari a 220V.

Quando inseriamo la spina di una lampada, creiamo un collegamento tra i fori, e **nel filo della lampada comincia a scorrere corrente elettrica**, che ha per effetto l'accensione della lampadina.



http://www.anarchia.com/link_in_frame.php?link=8676&c=

Lavoro – Energia – Potenza

Lavoro: (meccanica) grandezza fisica data dal prodotto scalare del vettore forza **F** per il vettore spostamento **s**.

$$L = F \cdot s.$$

L'unità SI del lavoro è il **joule** (**J**).
Una forza **F** di 1 N compie il lavoro di 1J quando il suo punto di applicazione si sposta di 1m nella stessa direzione e nello stesso verso della forza: 1J =1N x 1m.

L'**energia** E è una grandezza fisica che misura la capacità di un corpo o di un sistema di compiere lavoro, a prescindere dal fatto che tale lavoro sia effettivamente svolto. Siccome corrisponde potenzialmente ad un lavoro, si misura in **J**.

La **potenza** quantifica il trasferimento, la produzione e l'utilizzo dell'energia. È definita come **energia** (e quindi lavoro) **nell'unità di tempo**: **P = E/t**.
Nel SI la potenza si misura in **watt W**, rapporto tra unità di energia in joule (J) e unità di tempo in secondi (s):

$$1\ \mathrm{W} = 1\ \frac{\mathrm{J}}{\mathrm{s}}.$$

La **potenza elettrica assorbita da un carico** (la lampada) è data dal prodotto **tensione x corrente:** **P = V x I**
e si misura naturalmente anch'essa in **watt (W)**

Ricapitolazione: Tensione, Corrente, Potenza

Tensione V: **volt (V)**

Corrente I: **ampère (A)**

Potenza P = E / t = **V x I**: **watt (W)**

Energia E = P x t: joule (J) e watt-ora Wh (o kilowatt-ora kWh)

1 Wh = 1 J/s x 1 h = 1 J/s x 3600 s = 3600 J

1 kWh = 10^3 J/s x 1 h = 10^3 J/s x 3600 s = 10^3 x 3600 J = 3.6 10^6 J

Ricapitolazione: Tensione, Corrente, Potenza

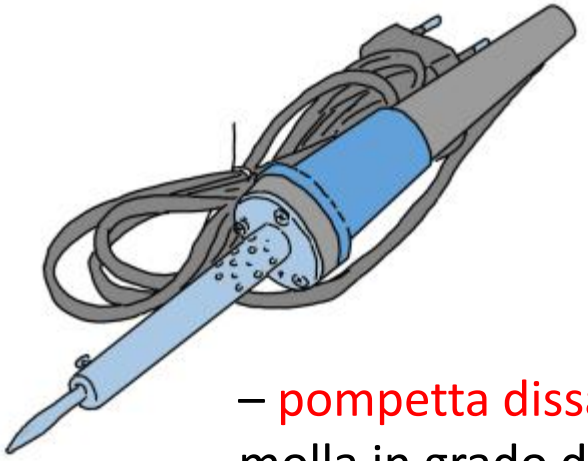
Esempio:

In **un'auto** (**tensione della batteria: 12V**), i fari anteriori sono equipaggiati di una lampada da **40W** ciascuno, mentre le luci posteriori hanno lampade da **5W**.

Quanta corrente è assorbita dalla batteria quando sono accesi i fari e le luci posteriori?

Quanta energia in Wh e in J è consumata in un viaggio di due ore?

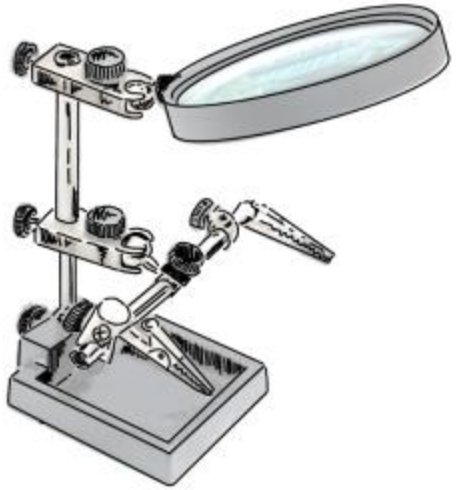
Attrezzi



– saldatore da 20/30 watt – il costo va da poche euro in su. Volendo investire di più, considerare una stazione saldante con controllo della temperatura (Weller, LaFayette…)

– pompetta dissaldante – è un cilindretto con una molla in grado di aspirare lo stagno in eccesso

– Spugnetta naturale – inumidita si utilizza per ripulire la punta del saldatore.



– Stagno – una volta si usava una lega di stagno e piombo. Il piombo facilita la fusione, ma è dannoso. Oggi esistono “leghe” *lead-free* che fondono a temperature maggiori. All’interno del filo di stagno ci sono dei micro forellini colmi di rosin o pasta saldante. Il rosin è un composto che facilita la fusione e aiuta a fare un lavoro “pulito”. Il rosin si liquefa e evapora (da **non inspirare**!!).

– Terza mano –base dotata di due mollette in cui fermare il pezzo che si sta lavorando

Breadboard (1)

La **Breadboard** è un dispositivo che facilita la costruzione e il test dei circuiti.

Tipicamente è utilizzata per fare **prototipazione** del design dei circuiti.

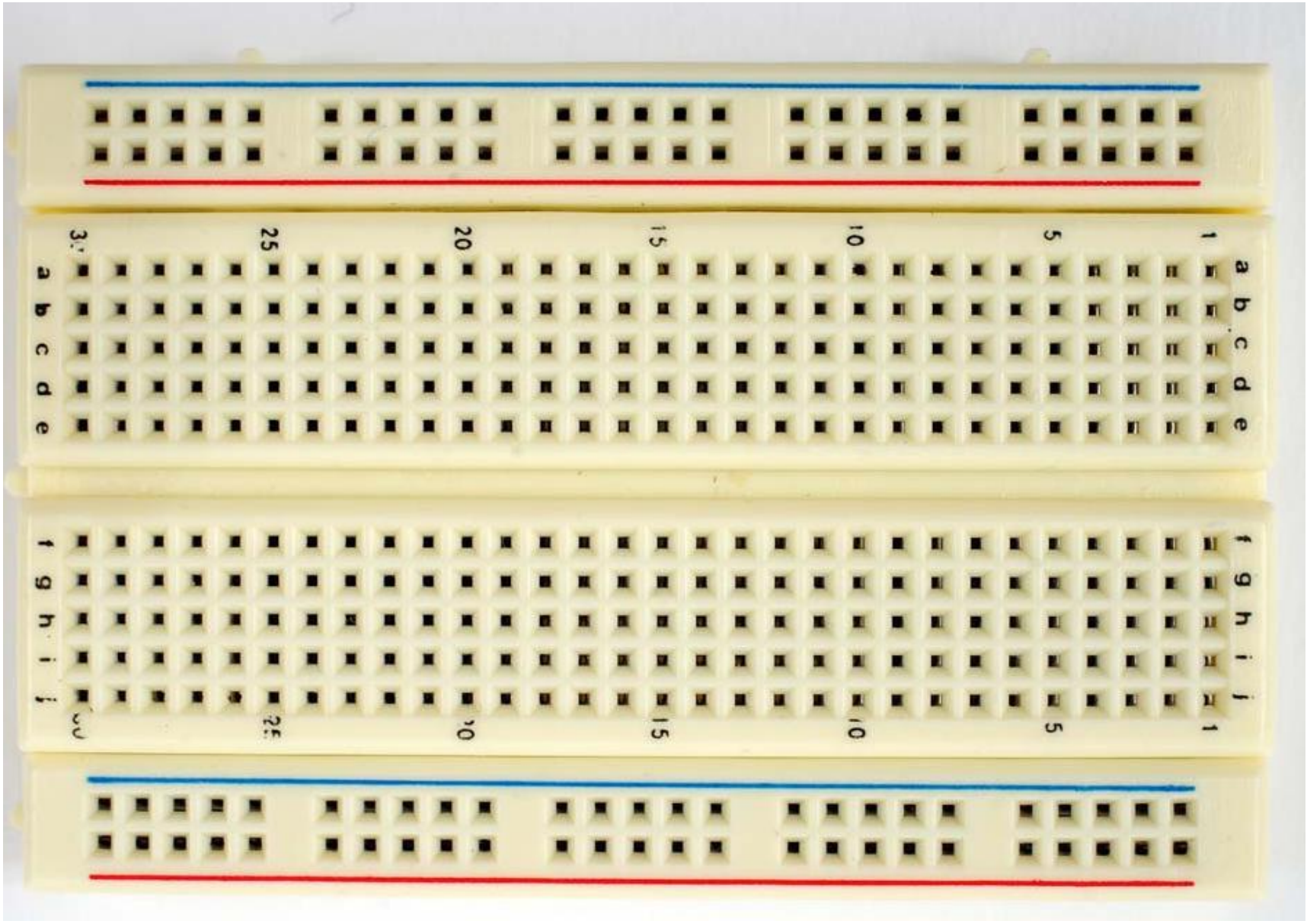
La Breadboard si può presentare come una **basetta di plastica con una griglia** di **fori** in cui si possono **inserire** i componenti o i cavi.

I componenti fanno contatto elettrico con delle **strisce di materiale conduttivo presenti sul fondo della basetta**. In questo modo non è necessario saldare ogni singolo componente.

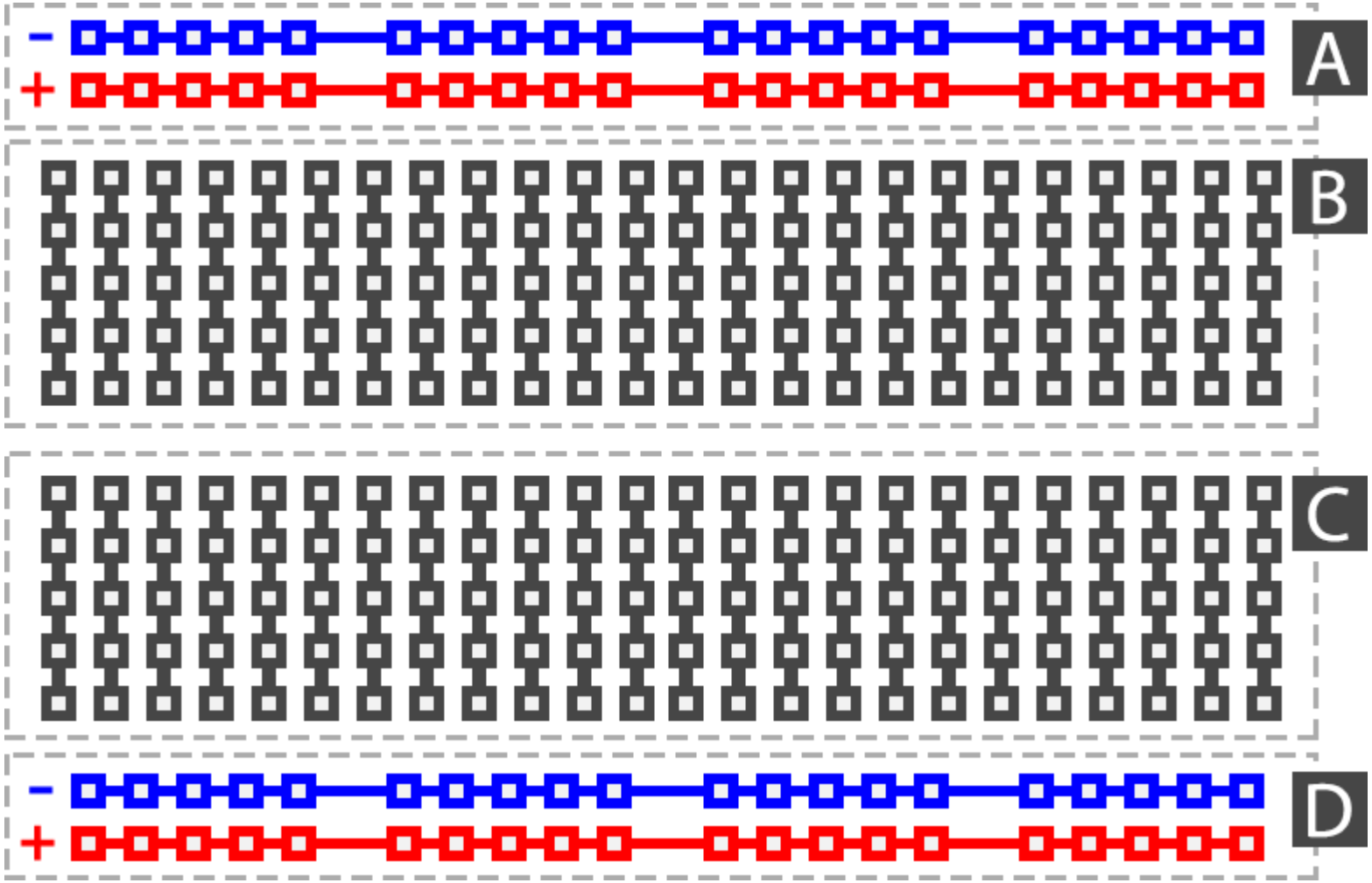
Di solito su due lati della Breadboard sono presenti **due righe per lato** identificate da **due strisce colorate (rosso per il +, blu per il -)** a cui si possono connettere i terminali positivo e negativo della tensione che alimenta il circuito (spesso detti rispettivamente **alimentazione e massa**). In questo modo si hanno sempre a disposizione (in entrambi i lati) il polo positivo e il polo negativo del circuito, a cui collegare i componenti.

http://www.giuseppecaccavale.it/corso-arduino/componenti-base-dellelettronica/

Breadboard (2)

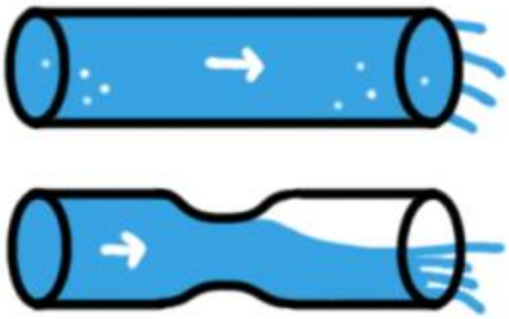


Breadboard (3)



Resistenza e strettoie (1)

Immaginiamo un **tubo di gomma** in cui scorre dell'acqua: se **schiacciamo** il tubo con un piede, il flusso d'acqua diminuisce: il tubo è più stretto e passa meno acqua. Il tubo esercita una certa **resistenza** sul **flusso d'acqua**.



Analogamente la **resistenza elettrica di un conduttore** è un proprietà che indica quanta **difficoltà** la corrente elettrica incontri nel percorrerlo.

Un **valore basso** indica un **conduttore che non oppone molta difficoltà al passaggio degli elettroni** ("buon conduttore") e fa passare **molta corrente**.
Un **valore alto** di resistenza caratterizza un conduttore che fa passare **poca corrente**.

La **resistenza si misura in Ohm** (simbolo **Ω**) dal nome dello scienziato tedesco George Simon Alfred Ohm (1789 – 1854).
Multipli: **kΩ, MΩ.**

Resistenza e strettoie (2)

Il **rame** è **buon conduttore** di corrente elettrica; ha un valore di resistenza molto basso:
0,000000017 Ω per un metro di filo di sezione di 1 mm.

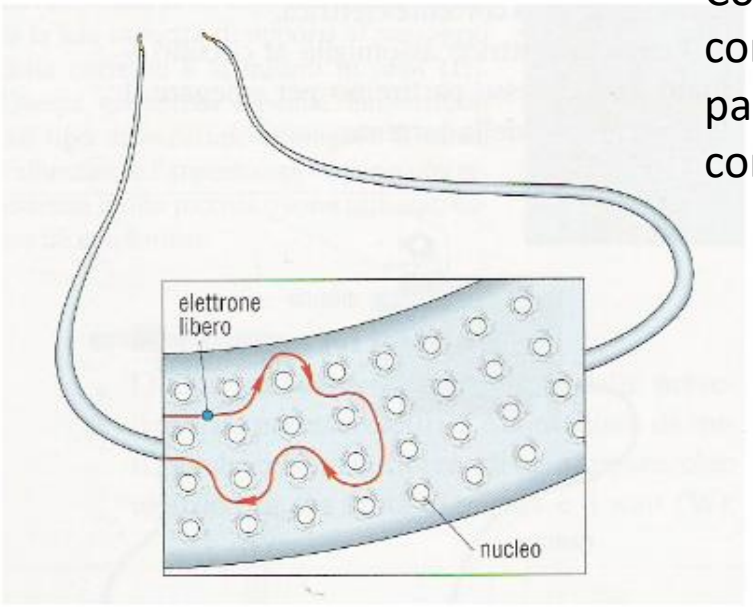
Il **ferro** conduce la corrente, ma lo fa un po' meno bene del rame; ha una resistenza di **0,000000097 Ω** per 1m di filo di sezione di 1 mm.

Il **vetro conduce pochissimo** la corrente. La sua resistenza arriva a **10^{14} Ω = 100.000.000.000.000 Ω** per 1m di filo di sezione di 1 mm.

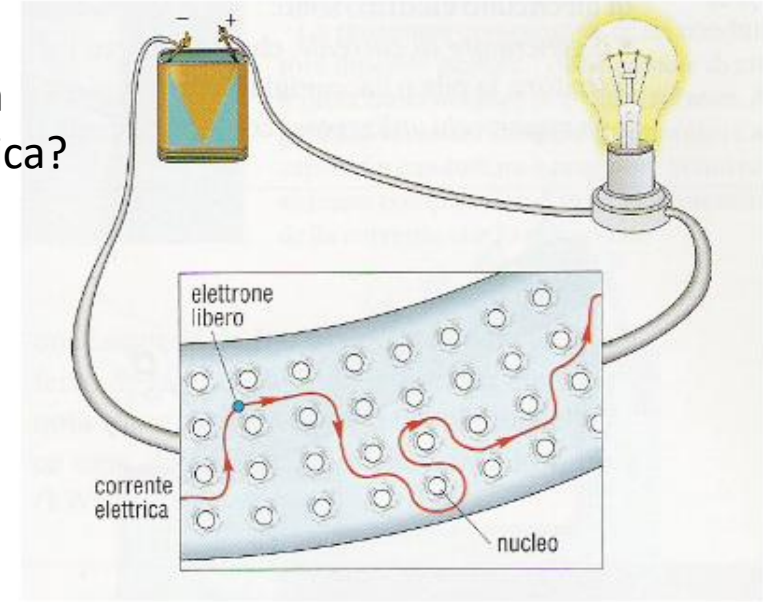
Conduttori e isolanti (1)

I materiali possono essere così suddivisi:

- buoni conduttori sono i metalli (ad esempio il rame, l'argento e l'alluminio, che vengono usati per costruire i cavi elettrici), i gas ionizzati (come quelli presenti all'interno dei tubi fluorescenti), le soluzioni elettrolitiche (acqua e sale) ed i tessuti organici, vale a dire il nostro corpo (purtroppo anche noi siamo dei conduttori!);
- conduttori meno buoni, isolanti deboli sono il legno e la carta;
- buoni isolanti come le ceramiche, il vetro, il marmo, la porcellana;
- isolanti buonissimi come l'ambra, la gomma.



Cosa succede in un conduttore al passaggio della corrente elettrica?



Conduttori e isolanti (2)

Zinco e **rame** sono dei *metalli*. Questi, come è noto, conducono corrente elettrica in quanto nei loro atomi gli elettroni sono "***delocalizzati*** " e, per questo, in grado di spostarsi da un atomo all'altro (figura nella slide precedente).

Lo **zolfo** è un *nonmetallo* ovvero un *solido molecolare* i cui atomi sono legati da legami covalenti ; gli elettroni possono, tutt'al più, essere condivisi tra gli atomi e non possono muoversi liberamente. I solidi molecolari covalenti sono, perciò, in massima parte, degli isolanti.

Il **carbonio grafite** è un buon conduttore, pur essendo anch'esso un *solido covalente*. In un atomo dei quattro elettroni di valenza, tre sono legati covalentemente con tre atomi vicini su un piano, formando una struttura stabile, mentre il quarto elettrone, nell'orbitale p, è delocalizzato e quindi in grado di spostarsi da un atomo all'altro. Questa struttura determina la conducibilità di questa sostanza.

http://www.itchiavari.org/chimica/lab/condelet.html

Conduttori e isolanti (3)

Per controllare la corrente che scorre in un conduttore, si può:

resistività

•usare un **materiale che abbia minore o maggiore "resistenza" elettrica** e quindi presenti una diversa attitudine ad essere attraversato dalla corrente

•a parità di materiale, si può usare un **filo più grosso o più sottile**: più è grosso il filo, maggiore è la corrente che riesce a passare

• rendere **minore o maggiore** la **lunghezza del filo**: più corto è il filo, più corrente passa.

$$R = \rho \frac{l}{S}$$

Seconda legge di Ohm: La resistenza dipende dalla **lunghezza** e dalla **sezione** del conduttore, e dalla **resistività**.

Il simbolo ρ indica la **resistività**, caratteristica fisica specifica di ciascun materiale: il rame ha una resistività minore del ferro e quindi è più adatto a far passare la corrente. Il nichelcromo ha una resistività elevata, e così risulta adatto per la costruzione di resistenze elettriche.

Resistori (1)

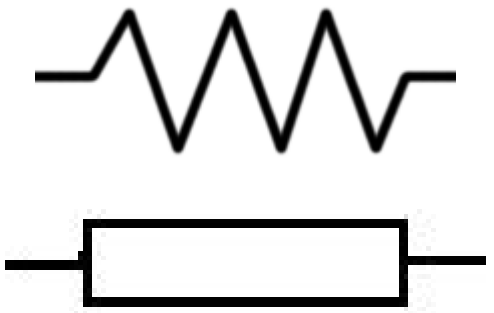
Un **resistore** è un componente elettrico dotato di una **ben determinata resistenza**.

Scopo dei resistori nei circuiti è **ridurre il flusso di corrente**.
Reti di resistori diminuiscono il valore della tensione nei circuiti.

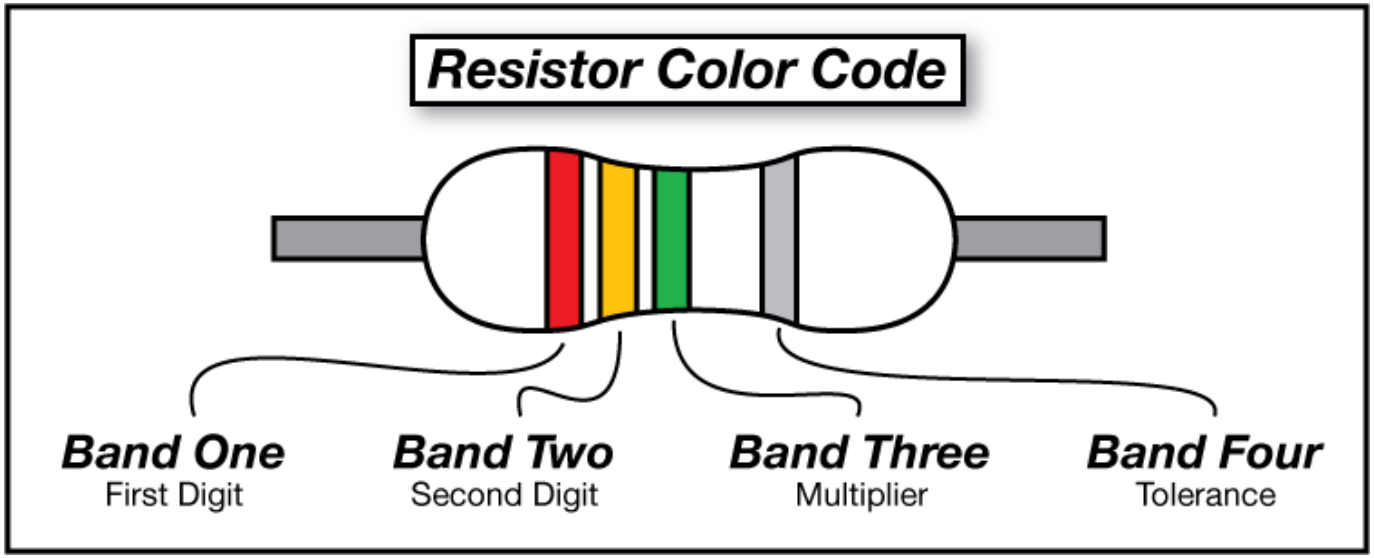
I **resistori** che si utilizzano nei circuiti elettronici **hanno valori di resistenza** che vanno da **qualche ohm fino a qualche milione di ohm** (MΩ)

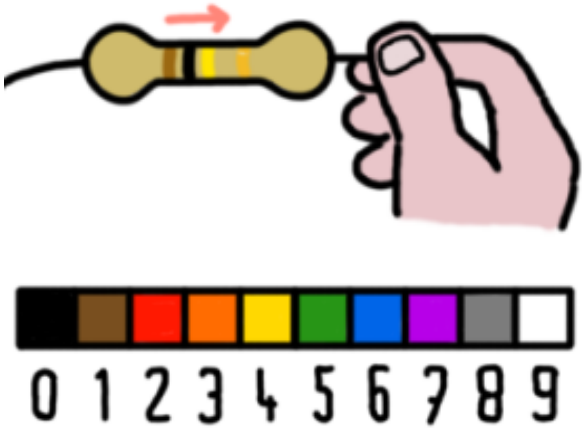
Di solito un resistore ha la forma di un bastoncino con le estremità rigonfie e con delle **strisce colorate**, che ne esprimono il valore con il cosiddetto "**codice dei colori**".

Simboli del resistore nei circuiti



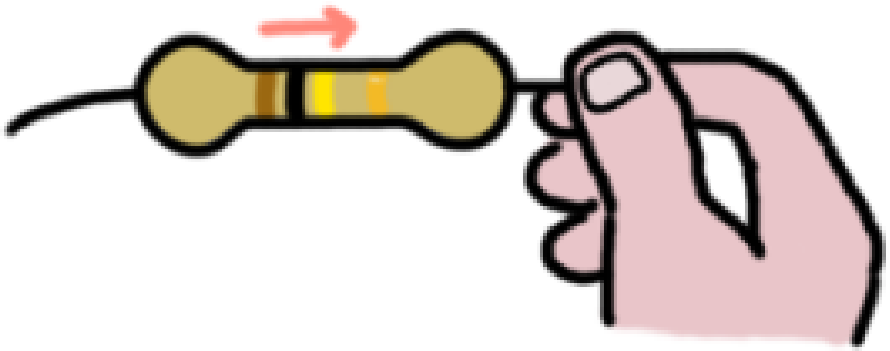
Il codice dei colori per i resistori (1)

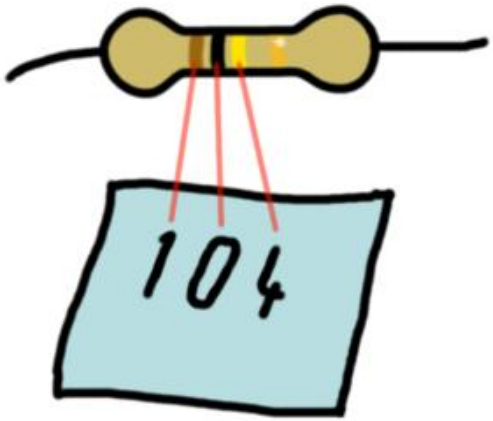




Normalmente troverete **quattro bande colorate**. C'è sempre **una banda a una estremità, un po' separata dalle altre, di colore dorato o argentato**. Per **leggere** il **valore di resistenza** tenete la **banda dorata a destra** e poi **leggete le bande colorate partendo da sinistra verso destra** secondo la tabella dei colori.

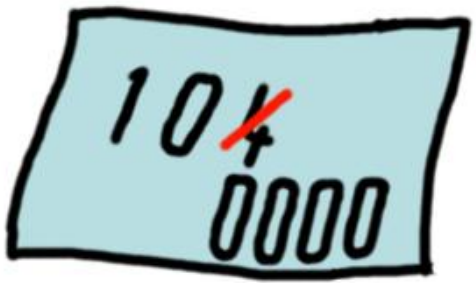
Il codice dei colori per i resistori (2)





Teniamo la banda dorata a destra. Quindi leggendo da sinistra verso destra troviamo una **banda marrone**. Scriviamo "**1**" su un foglio. La seconda **banda** è **nera** e quindi scriviamo "**0**". La terza é **gialla**, quindi scriviamo "**4**". Sul foglio avremo scritto "**1 0 4**", ma questa resistenza **non vale 104 ohm**!

Ora **cancelliamo** la **terza cifra** e sostituiamola con **un numero di zeri pari al valore della cifra**. Cancelliamo il numero "4" e scriviamo al suo posto quattro zeri: "0 0 0 0". Ora sul foglio dovremmo avere:



Cioè **10 x 10^4,**
ossia 100.000 Ω o,
meglio, **100 kΩ**

Nota: a volte il simbolo "k" o "M" si usa anche come virgola decimale!
Ad esempio :
2k2 = 2,2 kΩ = 2200 Ω
4k7 = 4700 Ω
3M3 = 3.300.000 Ω

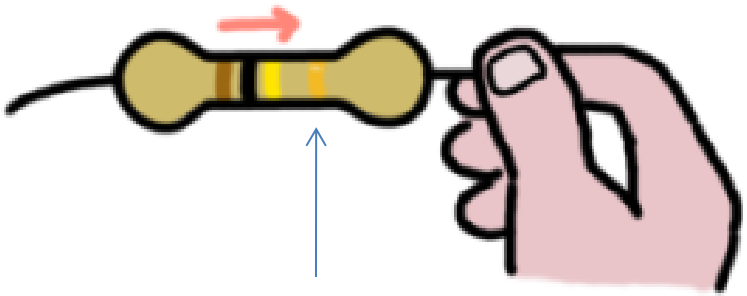
Il codice dei colori per i resistori (3)

L'ultima banda colorata indica la **precisione del valore della resistenza** (o "**tolleranza**").

Tipicamente è di **colore dorato** che significa che il valore **può variare in più o in meno del 5%** rispetto a quello nominale.

Se la banda è **d'argento** la tolleranza è del **10%**.

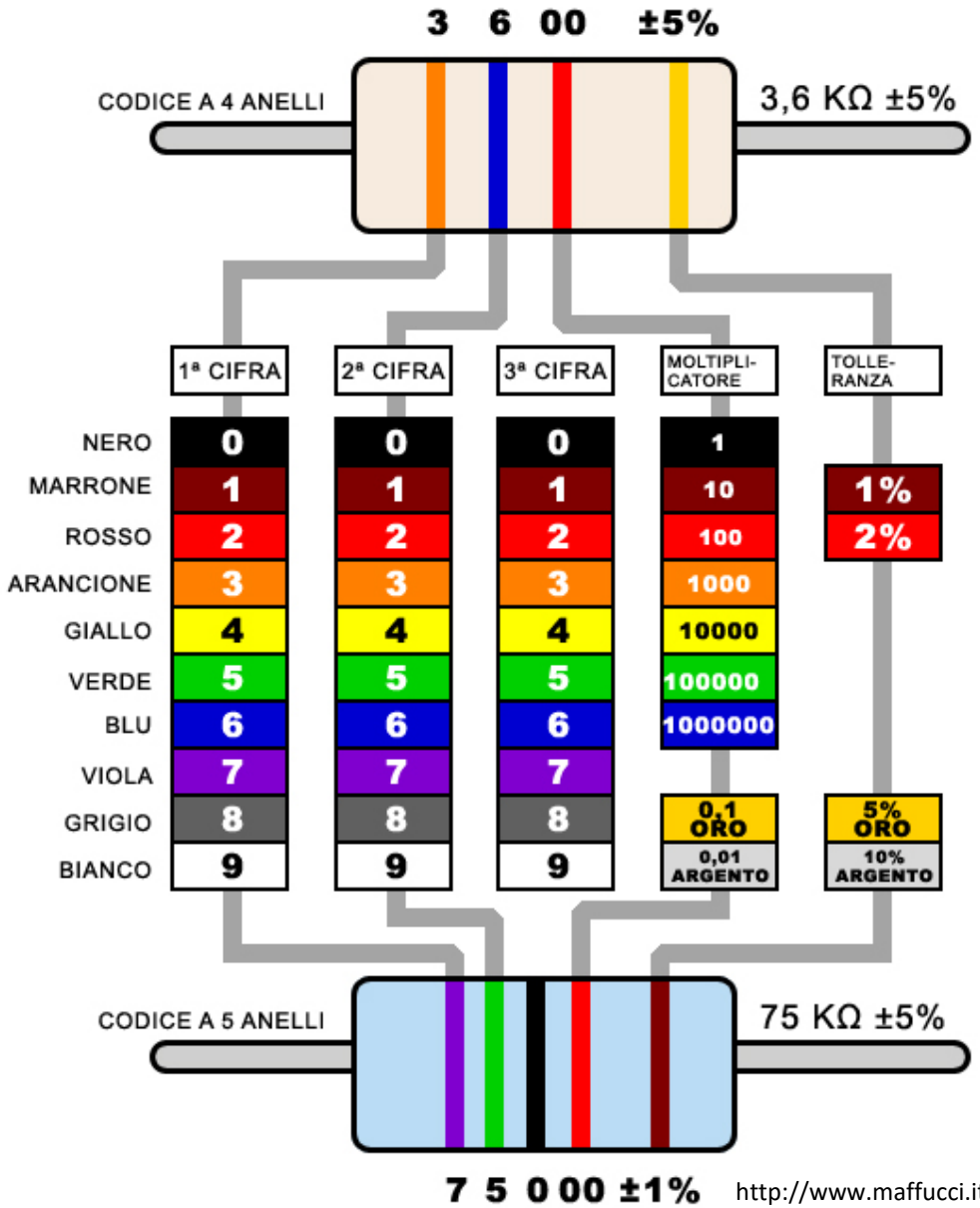
Esistono anche resistenze con valori più precisi: **2% o 1%,** ma sono leggermente più complicate da leggere.



Tolleranza

Esempio: un resistore da 1000 Ω con tolleranza del 5%, ha un'incertezza del valore di resistenza pari al 5% di 1000 Ω, ovvero 50 Ω, quindi il valore vero è tra 950 Ω e 1050 Ω.

Il codice dei colori per i resistori (4)



Codici dei colori a 4 (visto in precedenza) e a 5 anelli (o bande)

http://www.maffucci.it/2011/09/29/usare-il-multimetro-per-misurare-la-resistenza-elettrica/

Il codice dei colori per i resistori (5)

©Audiovalvole.it 2015

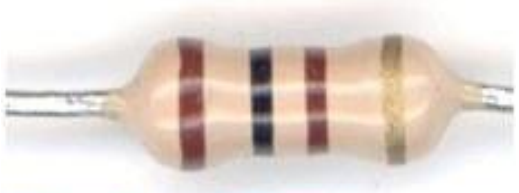
?

Il codice dei colori per i resistori (6)

Marrone/rosso/marrone/oro: quattro bande di colore
Marrone=1, Rosso=2, Marrone= moltiplica x 10^1
Quindi 120 ohm con la tolleranza del 5% (ultima banda color oro)
Il valore del resistore è compreso fra 120-((120/100)*5)=114 e 120+((120/100)*5)=126

Marrone/rosso/giallo/oro: quattro bande di colore
Marrone=1, Rosso=2, Giallo= moltiplica x 10^4
Quindi 120000 ohm con la tolleranza del 5% (ultima banda color oro)
Il valore del resistore è compreso fra 120000-((120000/100)*5)=114000 e 120000+((120000/100)*5)=126000

Marrone/nero/rosso/oro: quattro bande di colore
Marrone=1, Nero=0, Rosso = moltiplica x 10^2
Quindi 10000 ohm con la tolleranza del 5% (ultima banda color oro)
Il valore del resistore è compreso fra 10000-((10000/100)*5)=9900 e 10000+((10000/100)*5)=10100



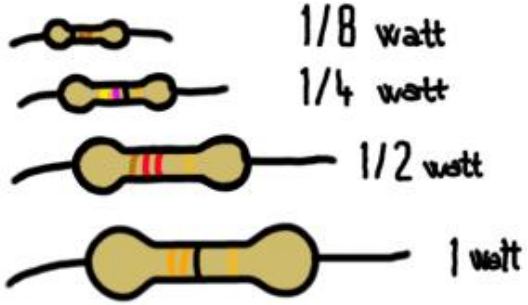
Marrone/nero/marrone/oro: quattro bande di colore
Marrone=1, Nero=0, Marrone= moltiplica x 10^1
Quindi 100 ohm con la tolleranza del 5% (ultima banda color oro)
Il valore del resistore è compreso fra 100-((100/100)*5)=95 e 100+((100/100)*5)=105

Resistori (2)

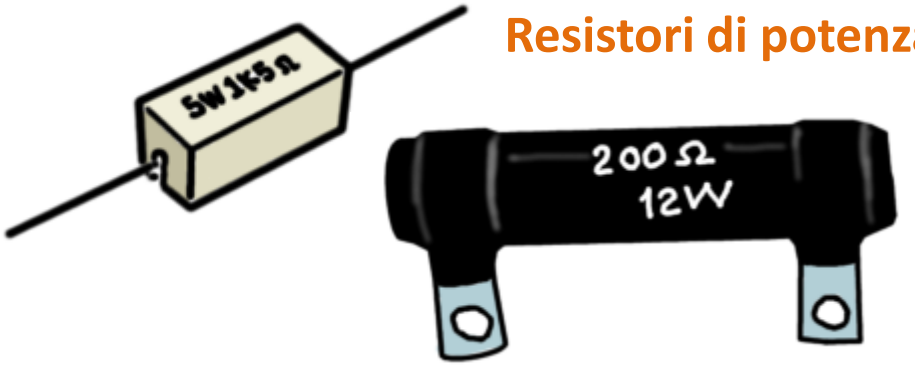
I **resistori** oppongono resistenza al passaggio della corrente e **generano quindi calore**. Se la **corrente che vi passa è rilevante**, la resistenza potrebbe **scaldarsi** molto.

Si può **calcolare la quantità di potenza dissipata**, che deve sempre essere minore della massima potenza sopportabile dal componente.

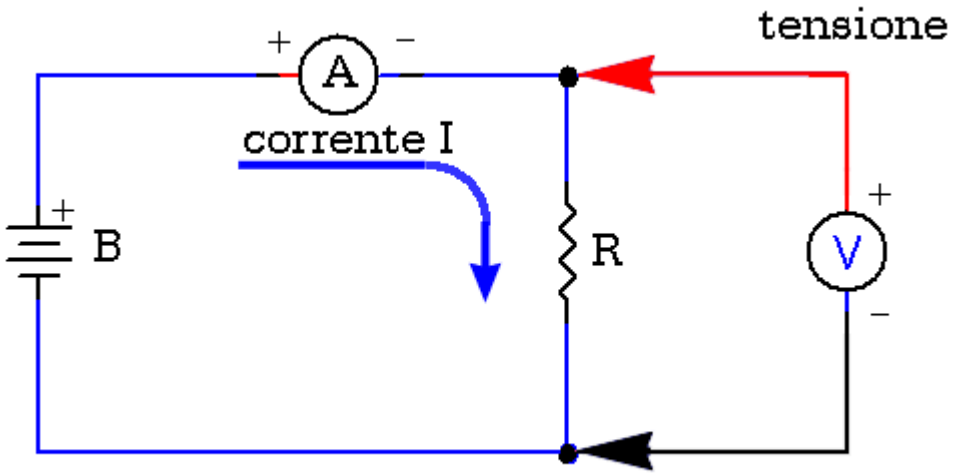
I **resistori più comuni sono quelli da 1/4 di watt**. E' possibile trovarne anche da **1/8** (sono molto piccole) oppure un po' più grandi per potenze da **1/2**, **1** o **2 W** e più



Resistori di potenza



(Prima) legge di Ohm



$V=R \cdot I$ $\quad I=V/R$

$R=V/I$

Esercizio: Un resistore (o un carico) ha una resistenza R = 56 Ω**.** Determinare la corrente che circola in esso, sapendo che ai suoi capi è applicata una tensione di 200 V:

$$R = 56\ \Omega \quad V = 200\ V \quad I = ?$$

Soluzione: Circuito formato da un generatore di tensione da 200 volt e un carico resistivo da 56 ohm, quindi basta applicare la legge di Ohm…

Corrente continua e corrente alternata (1)

Esiste una differenza fra la tensione erogata da una batteria (o pila o accumulatore) e quella delle prese elettriche.

A parte il diverso valore (220 V della presa di casa, 12 volt della batteria dell'auto, 1.5 V per le pile a stilo…), **la pila fornisce tensione continua**, mentre **la presa elettrica fornisce tensione alternata**.

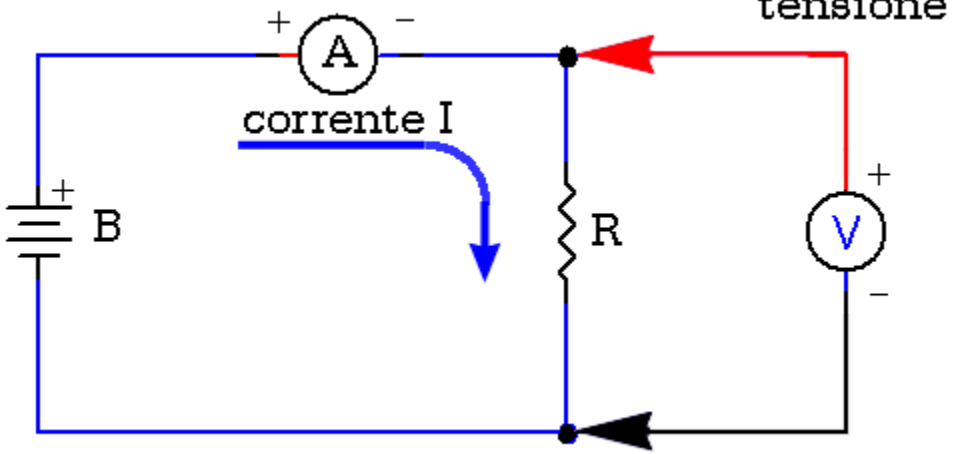
Entrambe sono disponibili tra i due "poli", ma nel primo caso i poli sono fissi, nel secondo si scambiano di segno (positivo + / negativo -) 50 volte al secondo.

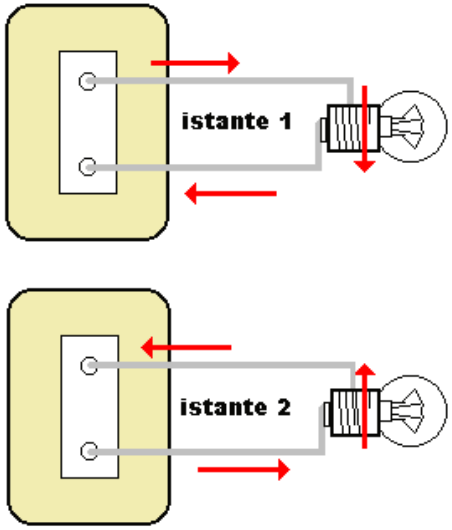
La corrente esce dal polo positivo, attraversa l'utilizzatore (per es. una lampadina) e rientra dal polo negativo. Finché la **pila** è carica ed eroga corrente, questa fluisce sempre nella stessa direzione e con un valore (almeno inizialmente, e fissato il "carico") costante: una corrente con tali caratteristiche viene definita "**corrente continua**".

Diversa è la corrente che preleviamo dalle **prese (corrente "di rete").** Per un breve tempo la corrente esce da un foro della presa e rientra nell'altro (consideriamo i due fori esterni), poi i ruoli si invertono, e così via. Oltre a cambiare direzione, la corrente fluisce con un valore che varia da zero ad un massimo e poi di nuovo a zero. Una corrente con tali caratteristiche viene definita "**corrente alternata**".

Corrente continua e corrente alternata (2)

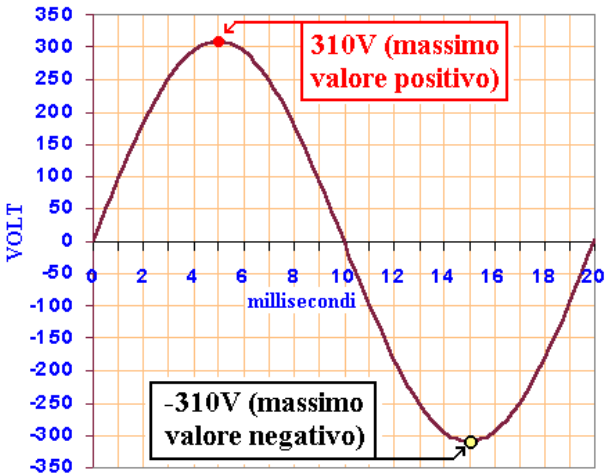
Corrente continua





Corrente alternata

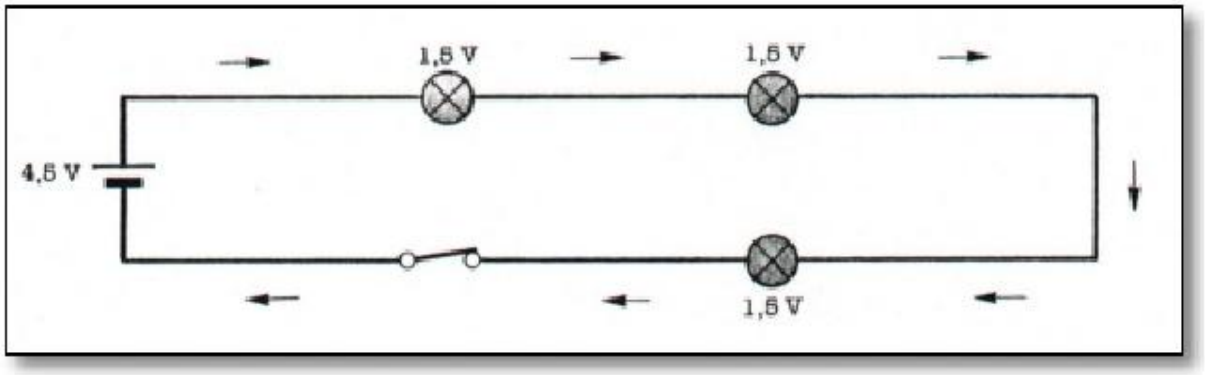
Come funzionerebbe una lampada se la corrente alternata di rete avesse una frequenza troppo bassa



Elettricità: concetti di base – 7

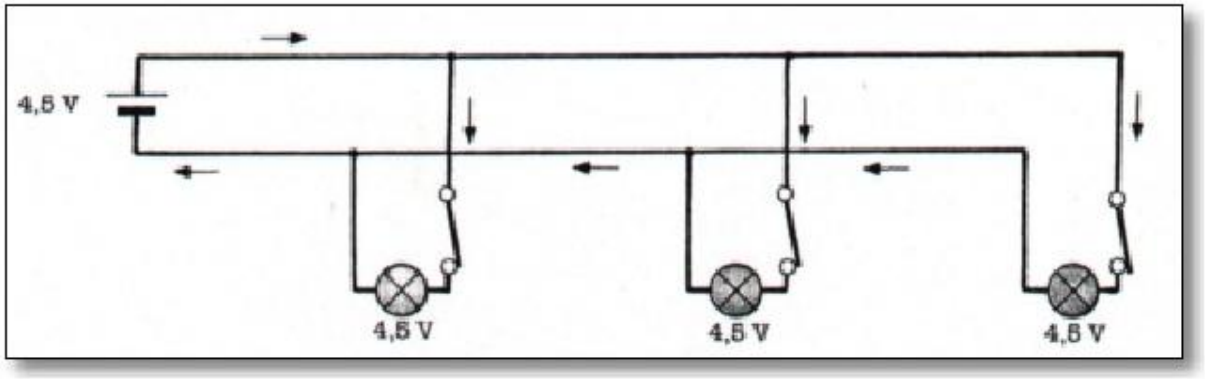
Circuito in serie

È quello nel quale gli utilizzatori si susseguono su di un unico conduttore (lampadine dell'albero di Natale) e l'avaria di uno di essi compromette l'intero circuito.



Circuito in parallelo

È formato da un conduttore, detto *primario*, dal quale si dipartono una serie di altri conduttori cui sono collegati i singoli utilizzatori. In tal caso il guasto di uno di essi é ininfluente.



Elettricità: concetti di base – 7.1

Collegamento di resistenze in serie e in parallelo.

Due o più resistori si dicono collegati in serie quando sono attraversati dalla stessa corrente. In fig. 1 sono riportate tre resistenze collegate in serie e sulla destra è riportato il simbolo della resistenza equivalente R_T, ossia di quella resistenza che dal punto di vista circuitale si comporta in modo perfettamente analogo alle tre resistenze. Quanto vale tale resistenza?

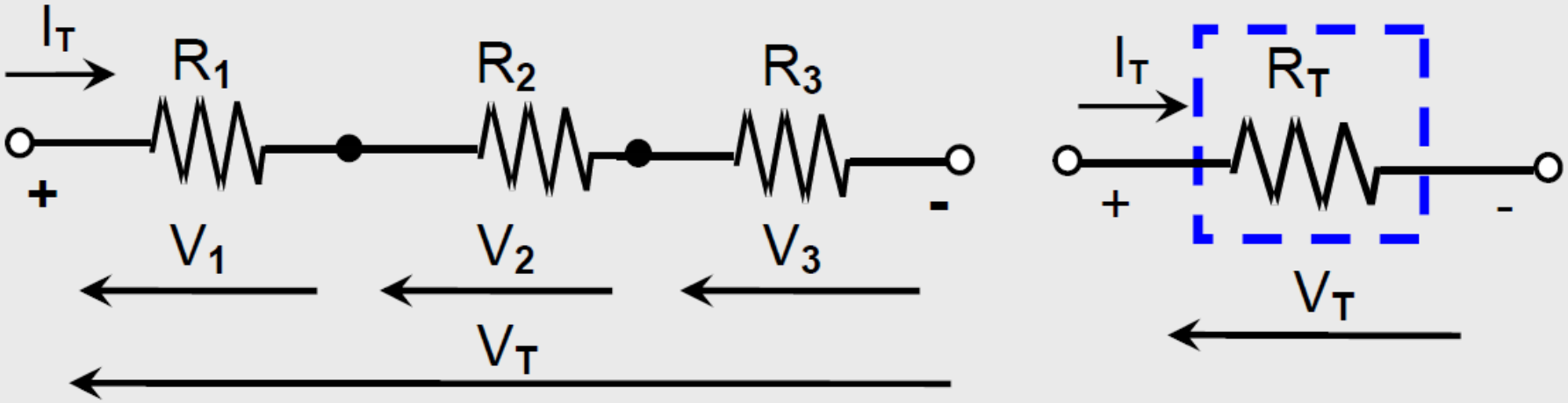


Fig. 1 – Resistenze in serie

Elettricità: concetti di base – 7.2

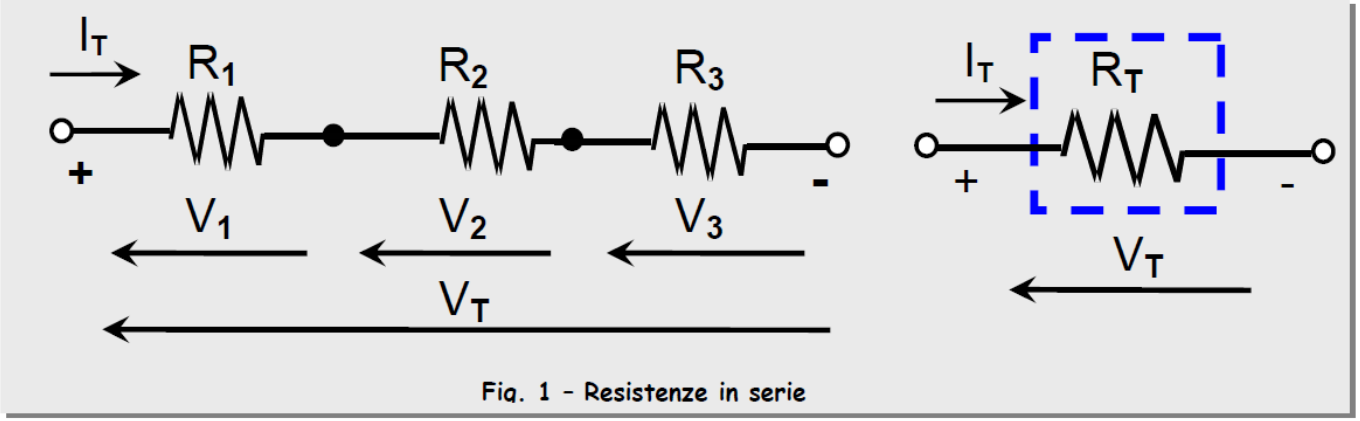


Fig. 1 - Resistenze in serie

$$V_1 = R_1 * I_T \qquad V2 = R_2 * I_T \qquad V_3 = R_3 * I_T$$

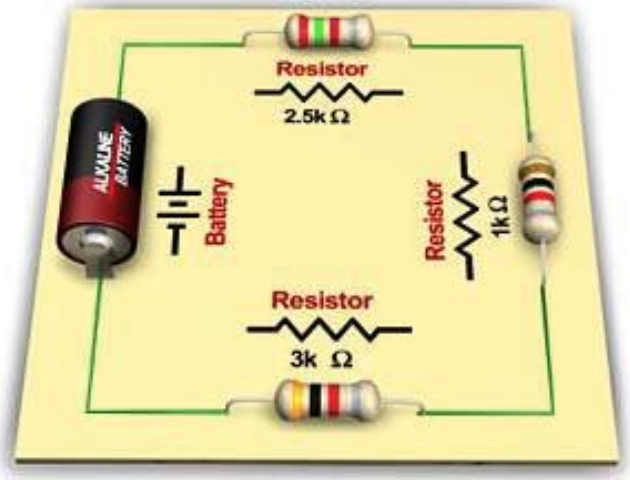
$$V_T = V_1 + V_2 + V_3$$

$$V_T = R_T * I_T$$

$$V_T = R_1 * I_T + R_2 * I_T + R_3 * I_T =$$

$$= (R_1 + R_2 + R_3) * I_T$$

$$R_T = R_1 + R_2 + R_3$$



Elettricità: concetti di base – 7.3

Ecco ora una sintesi delle principali caratteristiche di questa configurazione.

1. Resistenza totale è la somma delle singole resistenze

$$R_T = R_1 + R_2 + R_3 + \ldots$$

2. La corrente è la stessa in ogni resistenza

$$I_T = I_1 = I_2 = I_3 = \ldots$$

3. La tensione totale è la somma della tensione su ogni singola resistenza

$$V_T = V_1 + V_2 + V_3 + \ldots$$

Elettricità: concetti di base – 7.4

RESISTENZE IN PARALLELO

Due o più resistori si dicono collegati in parallelo quando ai loro capi è applicata la stessa differenza di potenziale. In fig. 2 sono riportate tre resistenze collegate in parallelo e sulla destra è riportato il simbolo della resistenza equivalente R_T. Ossia di quella resistenza che dal punto di vista circuitale si comporta in modo perfettamente analogo alle tre resistenze. Quanto vale tale resistenza?

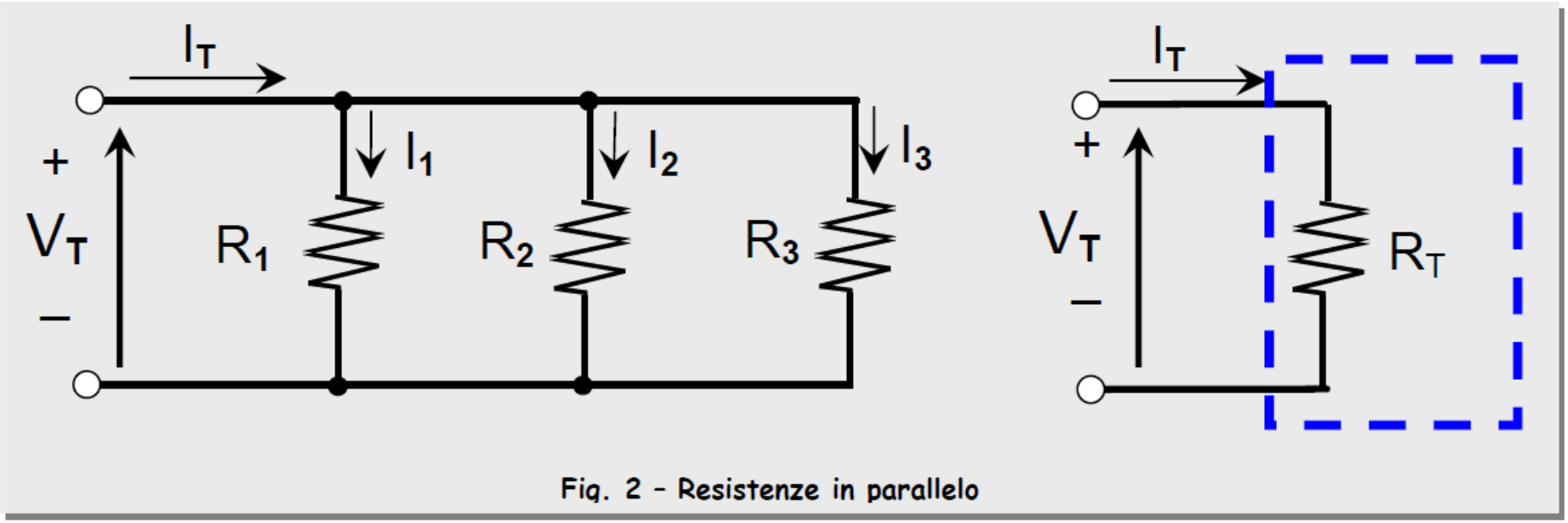


Fig. 2 – Resistenze in parallelo

Elettricità: concetti di base – 7.5

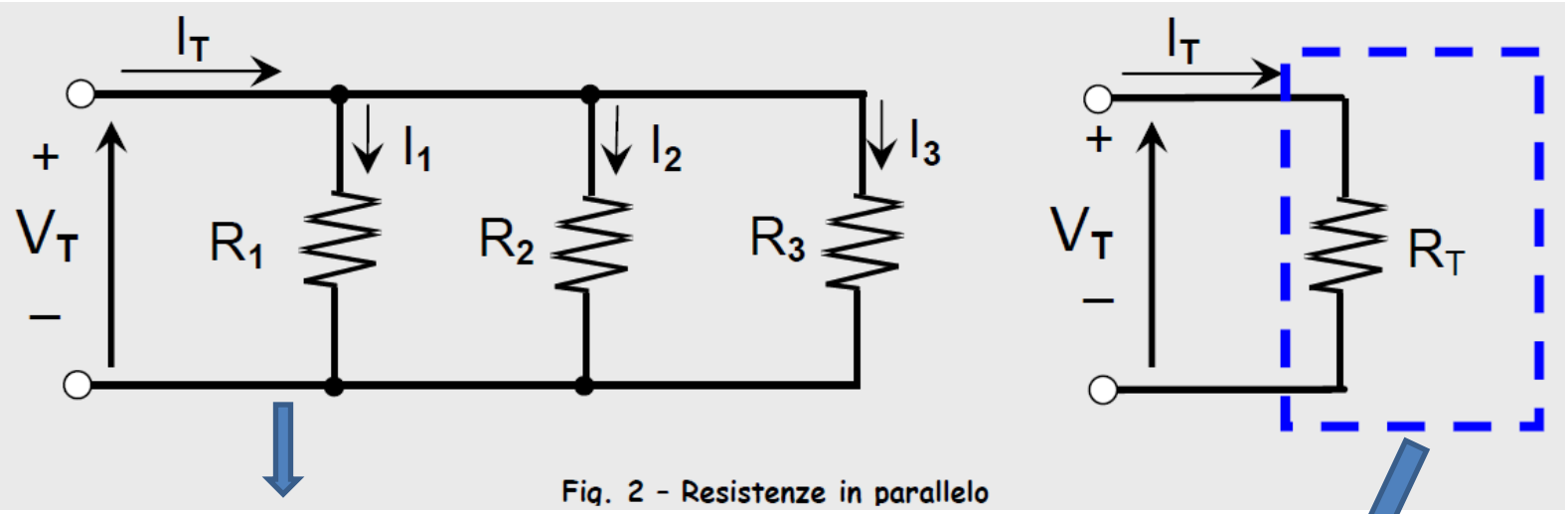


Fig. 2 - Resistenze in parallelo

$$I_T = I_1 + I_2 + I_3$$

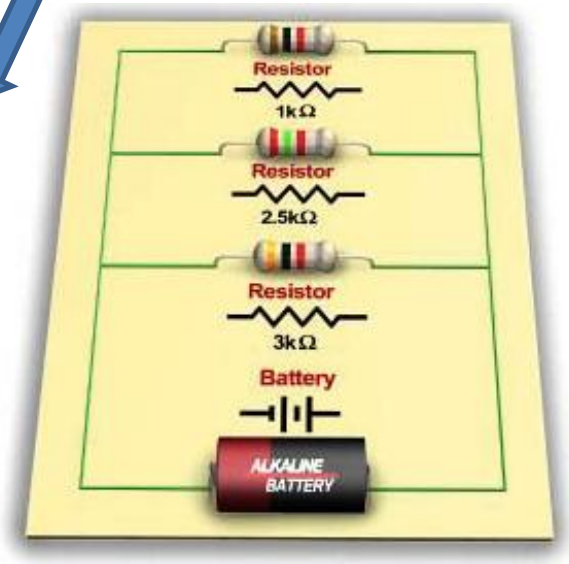
$$I_1 = V_T/R_1 \qquad I_2 = V_T/R_2 \qquad I_3 = V_T/R_3$$

$$V_T = R_T * I_T \quad \rightarrow I_T = V_T/R_T$$

$$I_T = V_T/R_1 + V_T/R_2 + V_T/R_3 =$$

$$= V_T / (1/R_1 + 1/R_2 + 1/R_3)$$

$$R_T = 1/(1/R_1 + 1/R_2 + 1/R_3)$$



Elettricità: concetti di base – 7.6

Ecco ora una sintesi delle principali caratteristiche di questa configurazione.

1. Il reciproco della resistenza totale è uguale alla somma dei reciproci delle singole resistenze

$$1/R_T = 1/R_1 + 1/R_2 + 1/R_3$$

2. La corrente totale è uguale alla somma delle correnti che attraversano le singole resistenze

$$I_T = I_1 + I_2 + I_3$$

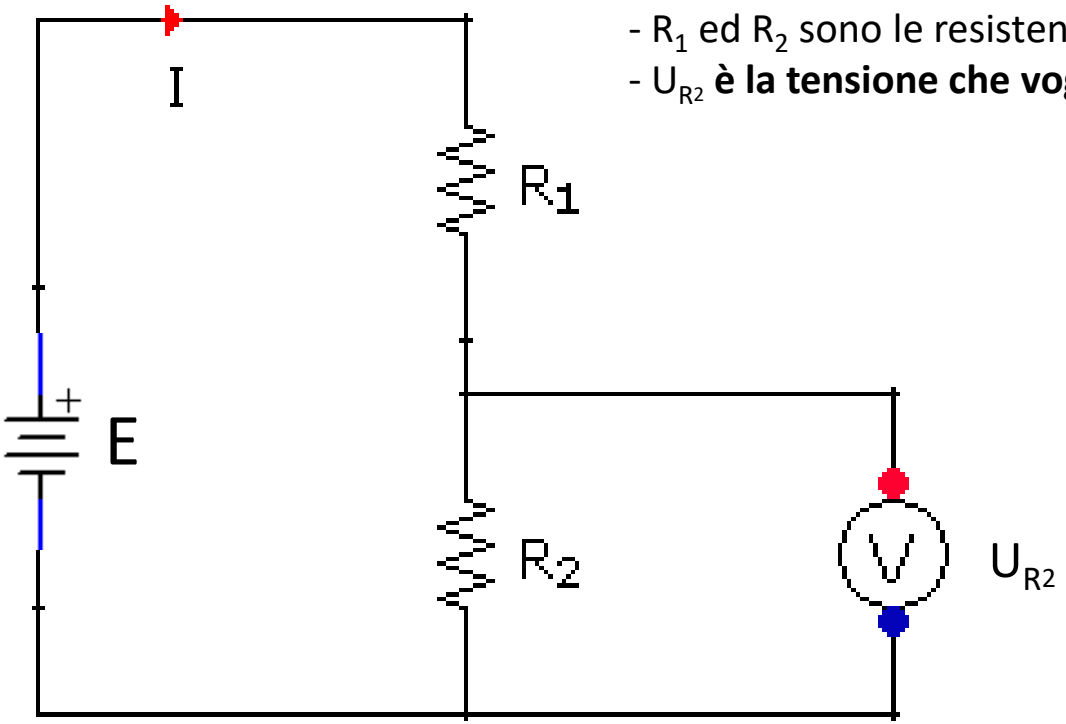
3. La differenza di potenziale è la stessa per tutte le resistenze

$$V_T = V_1 = V_2 = V_3$$

Elettricità: concetti di base – 7.7

IL PARTITORE DI TENSIONE

- E è la tensione del generatore;
- I è la corrente;
- R_1 ed R_2 sono le resistenze del partitore;
- U_{R2} **è la tensione che vogliamo calcolare**, misurata su R_2.

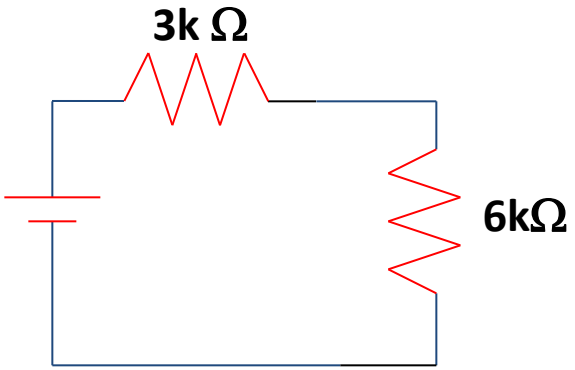


sostituendo la corrente I nella formula della legge di Ohm abbiamo:

Elettricità: concetti di base – 7.8

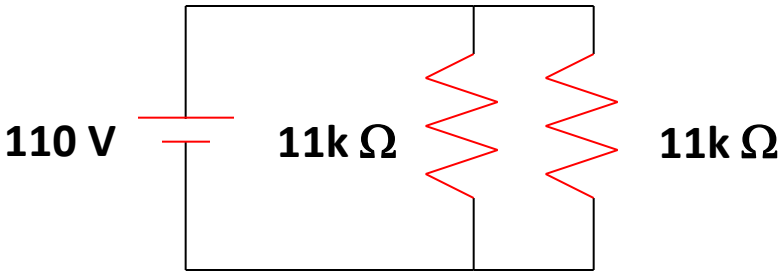
Esercizio n.1

Qual'è il valore della resistenza equivalente ai due resistori in serie? Quanta corrente scorre nel circuito?



Esercizio n.2

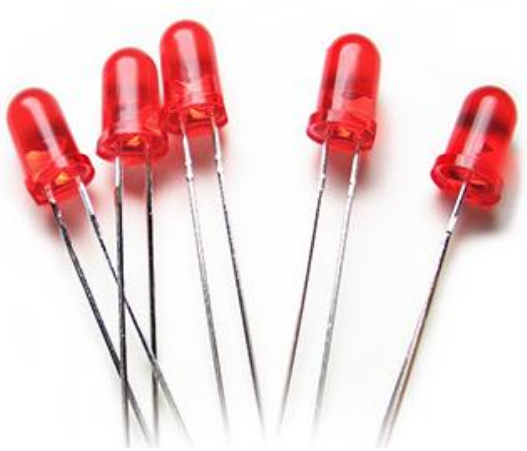
Calcolare la corrente nel seguente circuito. Qual è la resistenza equivalente dei due resistori in parallelo? Calcolare il voltaggio a cavallo di ciascun resistore (per un generatore ideale!!). Quanta potenza assorbe il circuito?



Diodi LED (1)

Un **diodo** è un dispositivo che permette il passaggio della corrente elettrica **in un solo verso** (**polarizzazione diretta**).

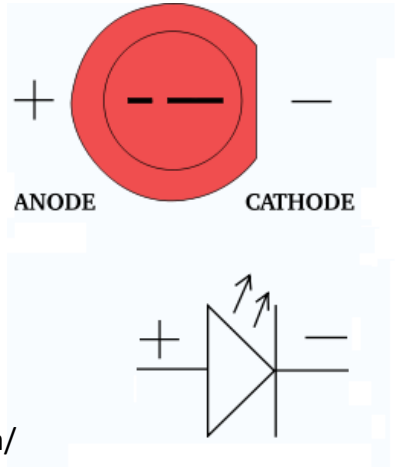
Un **diodo LED** (**Light Emitting Diode**) è un dispositivo che **emette luce** ad una specifica lunghezza d'onda (**colore**) quando è ad esso applicata una tensione.

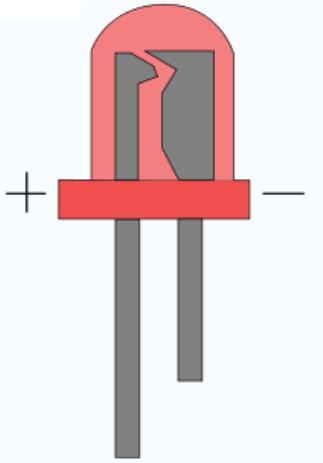


Un LED non ha limitatore di corrente, per cui **applicare una tensione** più **alta** rispetto alla sua tensione limite "forward voltage" causerà probabilmente in esso una corrente eccessiva e il surriscaldamento del LED con eventuale **rottura** dello stesso.

Solitamente il **terminale più lungo** di un led **è l'anodo (+)** e quello **più corto** è il **catodo (-)**.

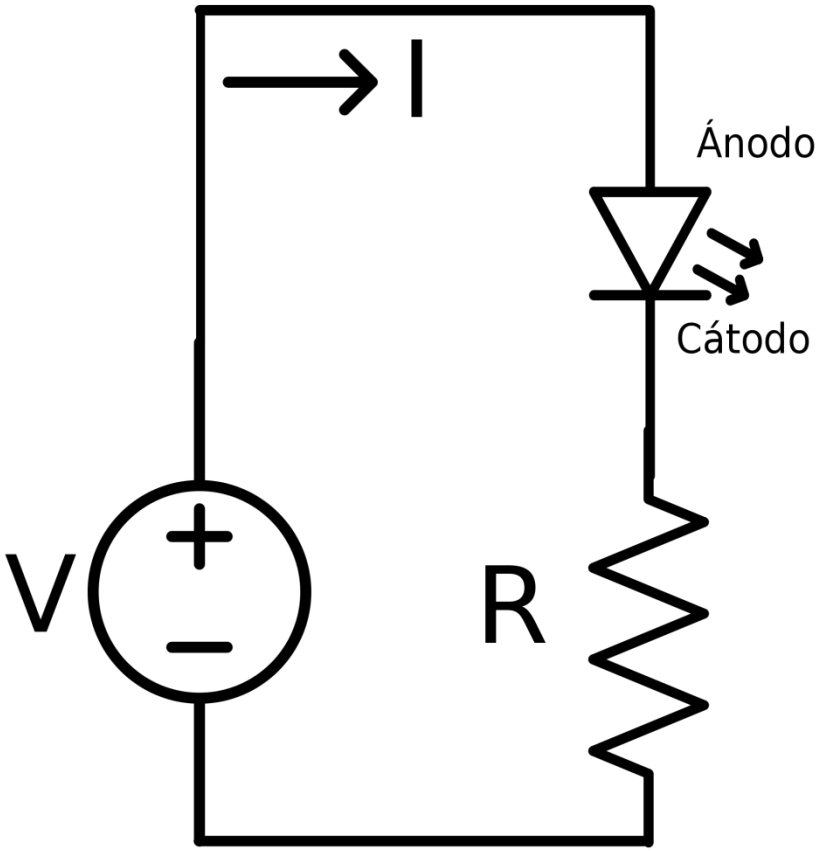
Se si guarda il led dall'alto, si può notare come la parte laterale del package sia squadrata da un lato: questa "**squadratura**" identifica il **catodo (-)**.

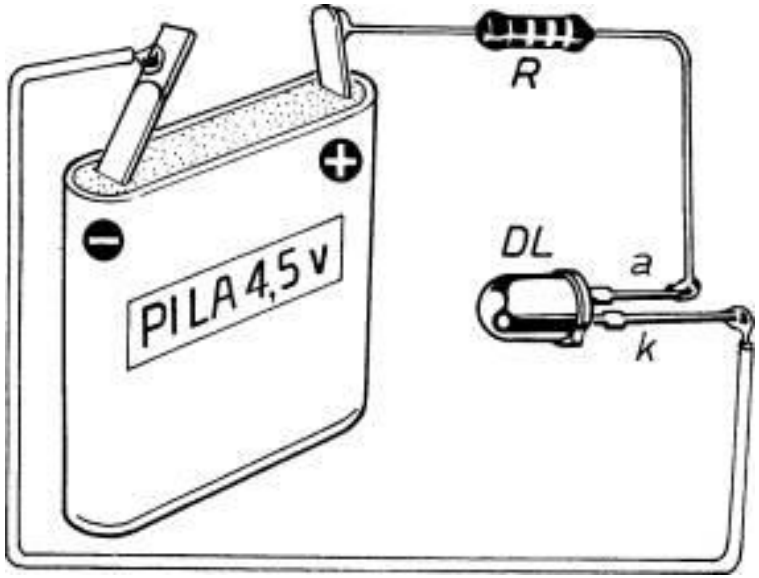




http://www.giuseppecaccavale.it/corso-arduino/componenti-base-dellelettronica/

Diodi LED (2)





R = 100 Ω - 500 Ω

http://www.giuseppecaccavale.it/corso-arduino/componenti-base-dellelettronica/
http://www.microst.it/Tutorial/led_fd_2.html

Diodi LED (3)

Come calcolare il valore della resistenza da inserire in serie al led?

La corrente per accendere un diodo LED oscilla tra i 15 e i 20 mA.
La tensione ai capi del LED ("caduta di tensione ") deve essere scelta in funzione del colore:
colore rosso: 1,8 V / colore giallo: 1,9 V / colore verde: 2,0 V
colore arancio: 2,0 V / colore blu: 3,0 V / colore bianco: 3,0 V

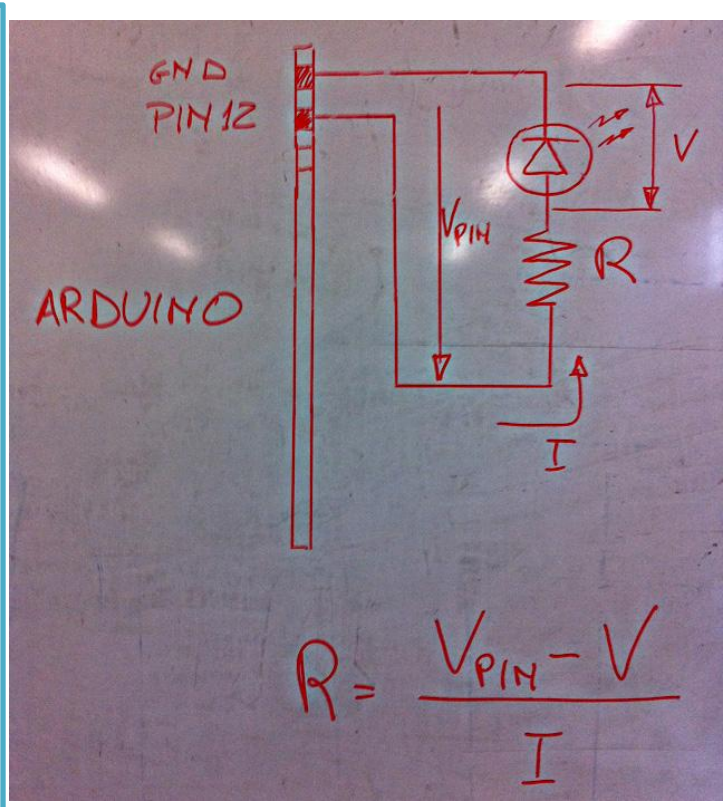
Per calcolare la resistenza R in serie bisogna utilizzare la seguente formula:

$R = (V_{in} – V_{led})/I$

dove R è la resistenza da inserire, V_{in} la tensione nella situazione di valore logico ALTO (+5V), V_{led} la caduta di tensione ai capi del diodo LED, I la corrente.

Supponiamo di utilizzare un diodo LED rosso;
la tensione sui pin digitali nella condizione di "ALTO" è circa +5V
la corrente di esercizio del led è di circa 20 mA

Il valore della resistenza serie sarà:
R = (5 – 1,8)/0,02 = 160 Ω
160 Ω non è un valore commerciale, il primo valore disponibile è 180 Ω e il successivo è 220 **Ω**.

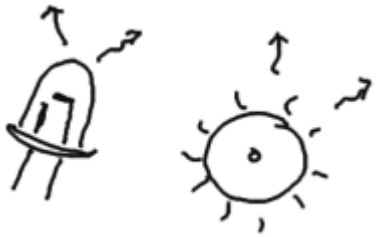


Per valori superiori, fino a circa 1 KΩ il diodo LED funziona senza alcun problema, per valori più elevati di resistenza il diodo avrà una luminosità inferiore.

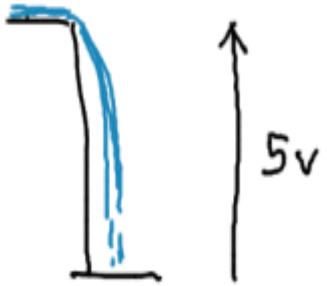
http://www.maffucci.it/2011/04/14/arduino-dimensionare-la-resistenza-serie-di-protezione-per-un-diodo-led/
http://playground.arduino.cc/Italiano/Led
http://golem.linux.it/wiki/Arduino,_led_e_resistenze
https://zmaker.wordpress.com/2012/12/06/che-resistenza-ci-metto/

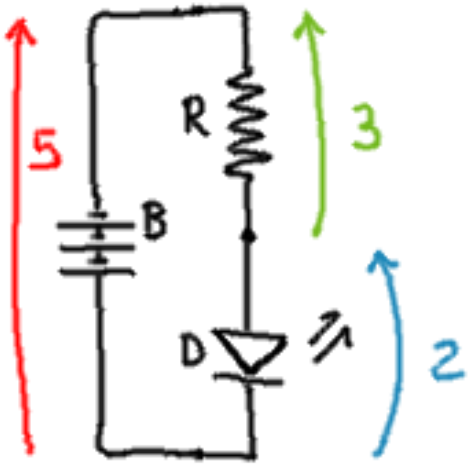
Diodi LED (4)

Come calcolare il valore della resistenza da inserire in serie al led?

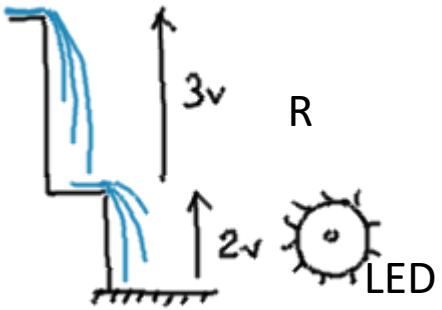


$V_{led} = 2V$





Posso pensare ad un alimentatore da 5 volt come una piccola cascata; la posso anche disegnare e la faccio alta 5 quadretti del foglio. Il LED richiede una tensione di circa 2 volt quindi una cascata adatta al LED dovrà essere di 2 volt (o due quadretti); se fosse più alta, il LED si danneggerebbe, se fosse più bassa non si accenderebbe (tutto ciò lo trovo nel manualetto di istruzioni del LED detto “datasheet”. Dal datasheet ricavo anche che al LED serve una corrente di 20 milli Ampere (0,020 Ampere).

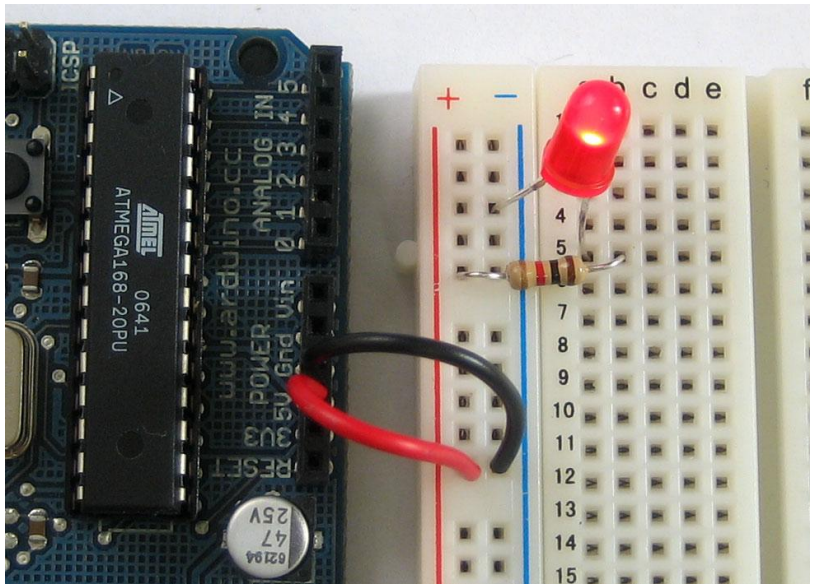


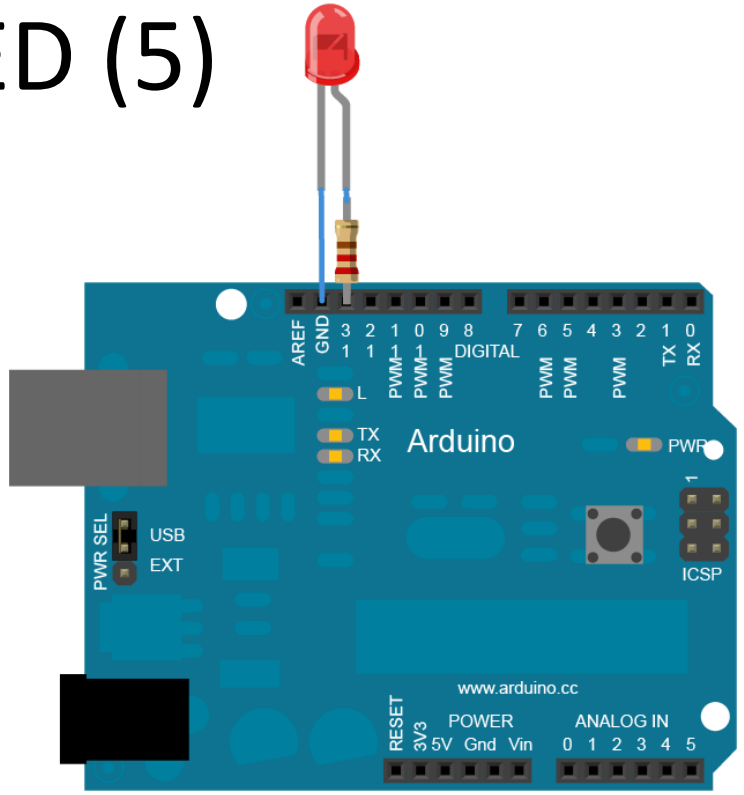
Devo cercare di spezzare la caduta dell’acqua e far fare 2 salti alla cascata: il primo da 3 e il secondo da 2 volt. Per ottenere il salto da 3 volt metterò la resistenza, cioè un tubo che si restringe. La resistenza limiterà la quantità d’acqua che scenderà per la cascata.

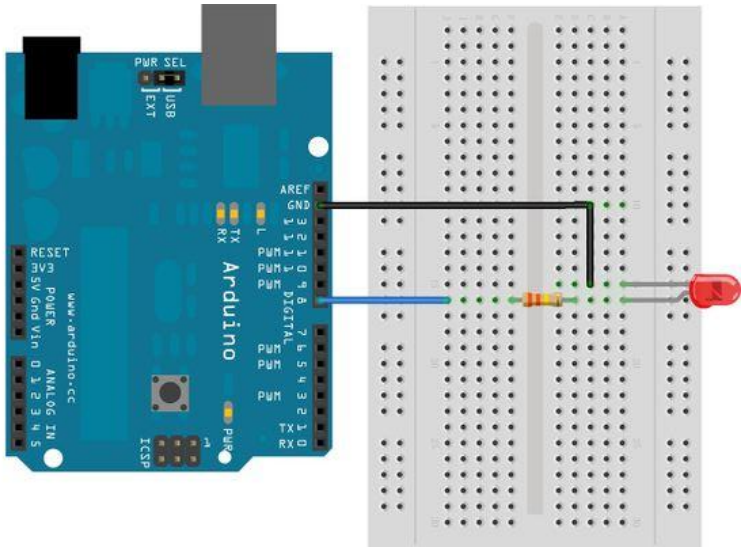
$$R = V / I = 3v / 20\ mA = 3v / 0.002\ A = 150\ ohm$$

http://www.maffucci.it/2011/04/14/arduino-dimensionare-la-resistenza-serie-di-protezione-per-un-diodo-led/
http://playground.arduino.cc/Italiano/Led
http://golem.linux.it/wiki/Arduino,_led_e_resistenze
https://zmaker.wordpress.com/2012/12/06/che-resistenza-ci-metto/

Diodi LED (5)


ATMEGA168-20PU
0641
www.arduino.cc
POWER
ANALOG IN


AREF
GND
DIGITAL
L
TX
RX
Arduino
PWR
PWR SEL
USB
EXT
ICSP
www.arduino.cc
RESET
POWER
ANALOG IN


PWR SEL
USB
EXT
Arduino
ICSP

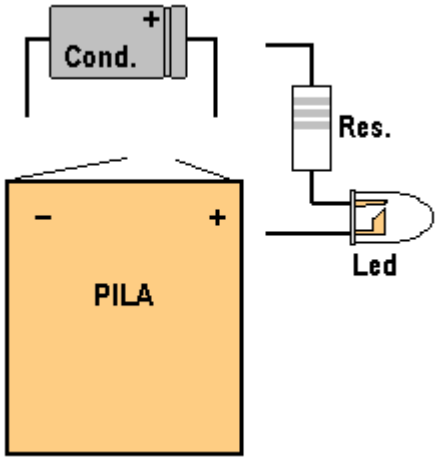
Elettricità: concetti di base – 8

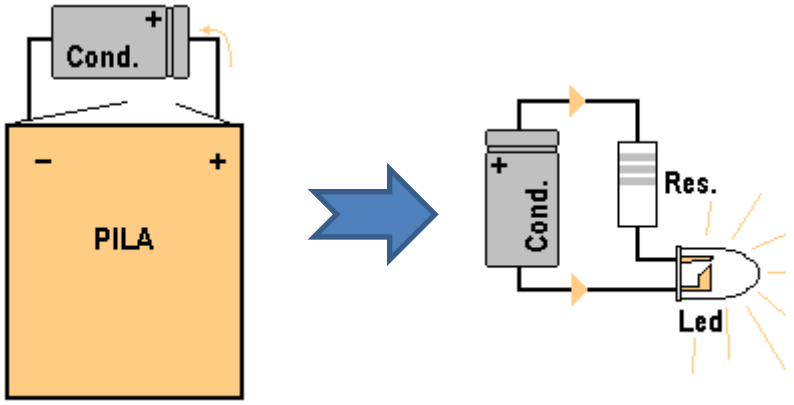
CHE COS'E' UN CONDENSATORE

La caratteristica del condensatore è la **capacità**, che si misura in **Farad** (F), o, meglio, in μF o nF.

Il condensatore è un dispositivo in grado di **immagazzinare carica** (e quindi energia) **elettrica**. La carica che si accumula in un condensatore dipende dalla capacità e dalla tensione di lavoro: **Q = C x V** [Q si misura in Coulomb]

Semplice esperimento: occorre una pila, un condensatore elettrolitico da circa 1000 μF e un led cui aggiungeremo in serie una resistenza da 100 ohm:





Elettricità: concetti di base – 8.1

Un condensatore è un sistema di due conduttori affacciati, le armature, separati da un isolante.

Serve per immagazzinare energia elettrica nel campo elettrico che si stabilisce fra le armature.

La grandezza che caratterizza un condensatore è la capacità C, ossia il rapporto tra quantità di carica Q deposta sulle sue armature e la differenza di potenziale V tra loro:

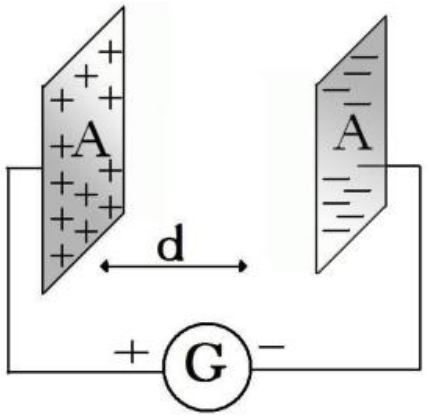
$$C = \frac{Q}{V}$$

Carica Q misurata in Coulomb

1 Coulomb = $6{,}24 \times 10^{18}$ volte la carica di un elettrone

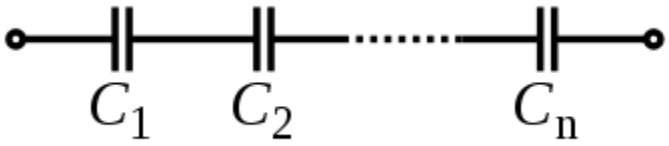
Un condensatore piano è costituito da due lastre metalliche piane di superficie A, separate da un dielettrico (un isolante) con costante ε, poste a distanza d.

$$C = \varepsilon \frac{A}{d}$$

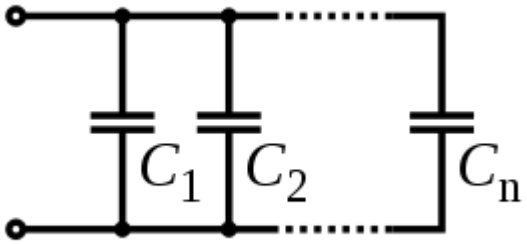


Elettricità: concetti di base – 8.2

Condensatori in serie e in parallelo



$$\frac{1}{C_{\text{equivalente}}} = \frac{1}{C_1} + \frac{1}{C_2} + \cdots + \frac{1}{C_n}$$

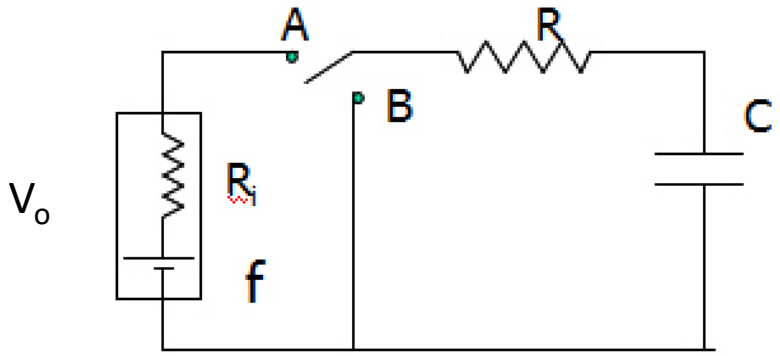


$$C_{\text{totale}} = C_1 + C_2 + \cdots + C_n$$

Elettricità: concetti di base – 8.3

Carica e Scarica di un condensatore

Consideriamo il processo di carica di un condensatore C, inserito nel circuito seguente:



Portando il commutatore sulla posizione A, si ottiene il circuito relativo al processo di carica

La carica elettrica che si accumula sulle armature quando si sottopone il condensatore alla d.d.p., non raggiunge istantaneamente il suo valore massimo $Q = C V_o$.

Questo avviene perché man mano che la carica si accumula sull'armatura, aumenta la forza di repulsione tra le cariche e perciò aumenta il lavoro necessario al generatore per accumulare altre cariche.

Elettricità: concetti di base – 8.4

Carica e Scarica di un condensatore

La legge che esprime il valore della carica in funzione del tempo q(t), durante il processo di carica, è:

$$q(t) = Q(1 - e^{-\frac{t}{RC}})$$

$Q = C V_o$

in cui Q indica il valore finale della carica, R la resistenza elettrica del circuito con il quale il condensatore si carica e C la capacità del condensatore. Pertanto, il grafico relativo alla legge sarà di tipo esponenziale.

La rapidità con cui la carica del condensatore aumenta dipende dal prodotto RC, che rappresenta la costante di tempo t.

Dopo un tempo t=RC, il condensatore raggiungerà il 63% della carica totale Q, dopo t=2RC l'86%, dopo t=3RC il 95% e dopo t=4RC il 98%.

Elettricità: concetti di base – 8.5

Carica e Scarica di un condensatore

$$q(t) = Q(1 - e^{-\frac{t}{RC}})$$

CARICA!

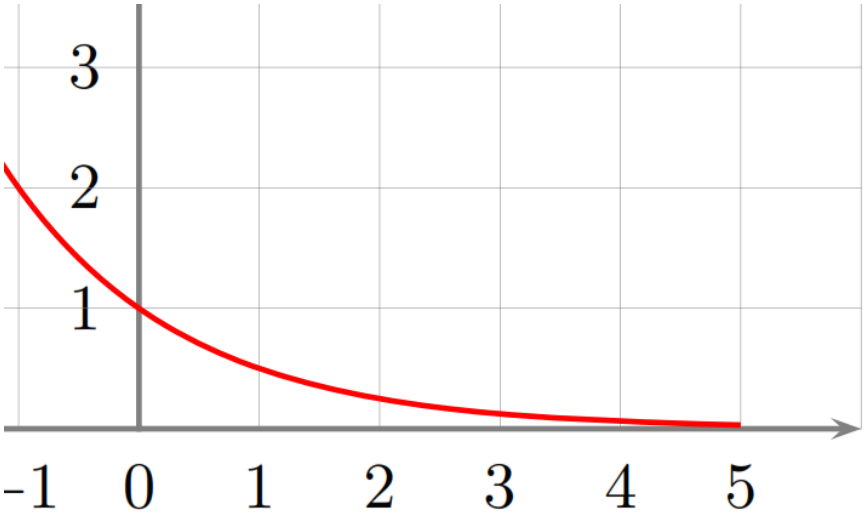
$$V(t) = V_o(1 - e^{-\frac{t}{RC}})$$

Elettricità: concetti di base – 8.6

Carica e Scarica di un condensatore

Com'è fatta la funzione: $V(t) = V_o(1 - e^{-\frac{t}{RC}})$???

Un'esponenziale con esponente negativo ha quest'aspetto:



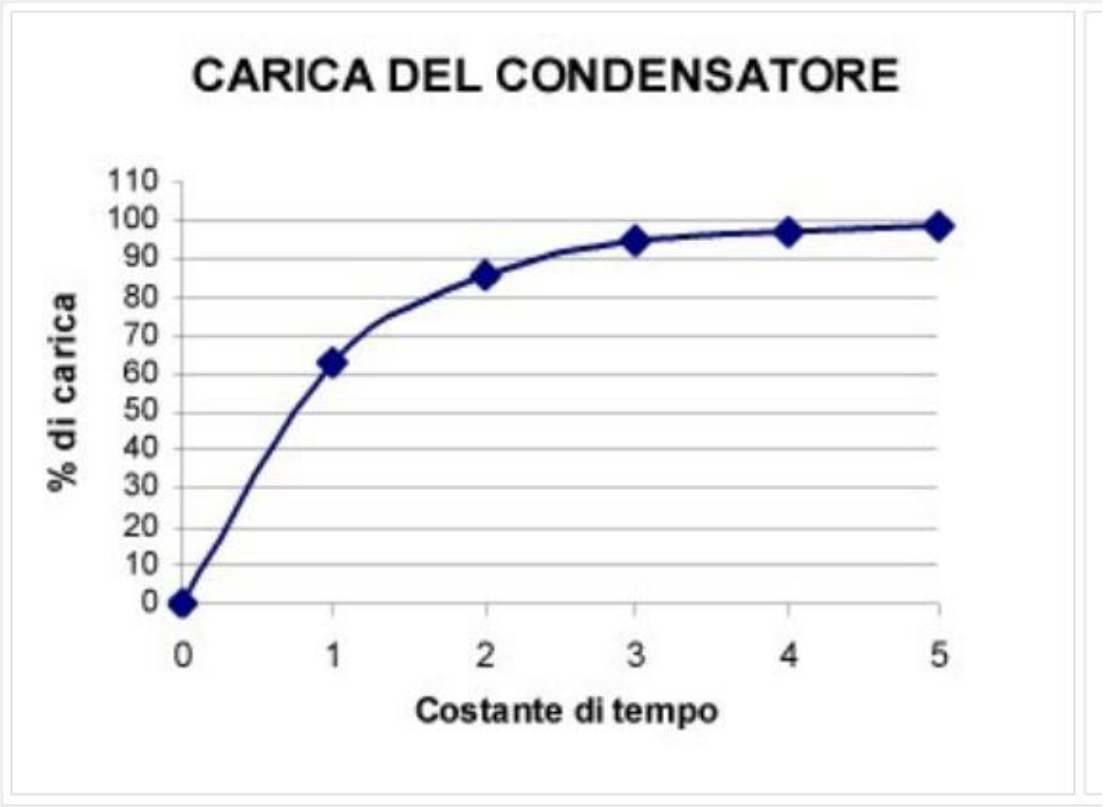
Elettricità: concetti di base – 8.7

Carica e Scarica di un condensatore

Quindi la funzione:

$$V(t) = V_o\left(1 - e^{-\frac{t}{RC}}\right)$$

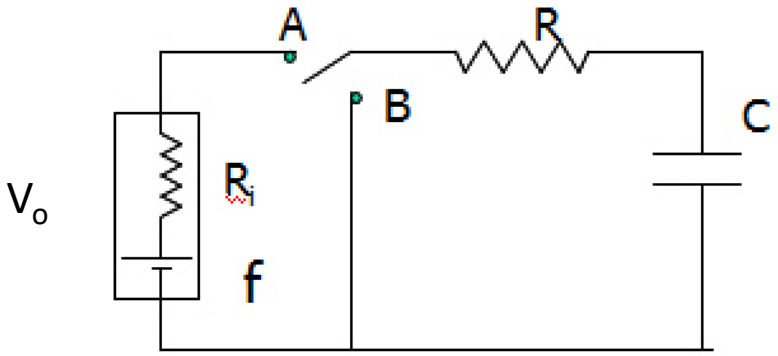
è così:



Elettricità: concetti di base – 8.8

Carica e Scarica di un condensatore

Consideriamo ora il processo di scarica del condensatore C:



Portando il commutatore sulla posizione B, si ottiene il circuito relativo al processo di scarica.

Analizzando la fase di *scarica si giunge a stabilire che il* valore della carica q rilevabile sulle armature del condensatore dopo un tempo t, è espresso dalla relazione

$$q(t) = Qe^{-\frac{t}{RC}}$$

da cui…

$$V(t) = V_o e^{-\frac{t}{RC}}$$

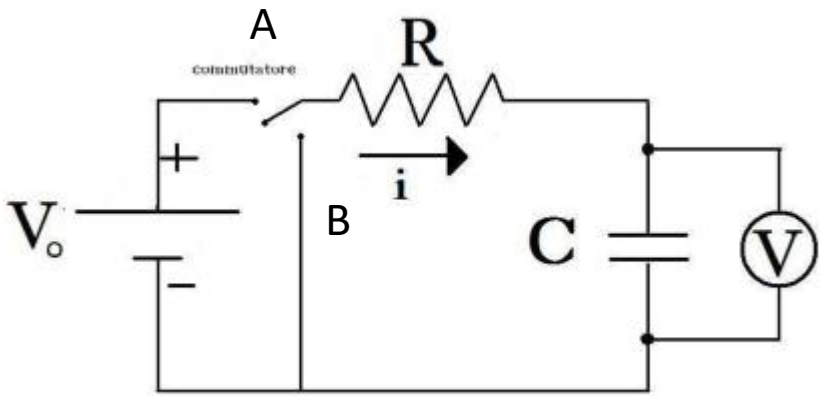
Elettricità: concetti di base – 8.9

Carica e Scarica di un condensatore

$$V(t) = V_o e^{-\frac{t}{RC}}$$

Elettricità: concetti di base – 8.95

Esperimento: Carica e Scarica di un condensatore



Portando il commutatore sulla posizione A, si ottiene il circuito relativo al processo di carica. Nella posizione B, la scarica.

Valori dei componenti:

R = 47 Ω
C = 2200 μF
τ = RC = …… s

$\ln V = \ln V_o - t/RC$

Logaritmi

definizione		
il logaritmo di un numero è l'esponente x da dare alla base a per ottenere l'argomento b cioè: $a^x = b$		
$\log_a b = x$	a si chiama **base** b si chiama **argomento** x è il **logaritmo** in base a di b	la base a deve essere > 0 e $\neq 1$ l'argomento b deve essere > 0 il logaritmo x è un numero reale $\mathbb{R}$
proprietà		
$\log_a a = 1$	$\log_a 1 = 0$	$a^x > 0$

teoremi principali sui logaritmi		
$\log_a(b \cdot c) = \log_a b + \log_a c$	teorema del prodotto	$\log_2 3x = \log_2 3 + \log_2 x$
$\log_a\left(\frac{b}{c}\right) = \log_a b - \log_a c$	teorema del rapporto	$\log_2 \frac{x}{3} = \log_2 x - \log_2 3$
$\log_a b^c = c \log_a b$	teorema della potenza	$\log_2 x^3 = 3 \log_2 x$

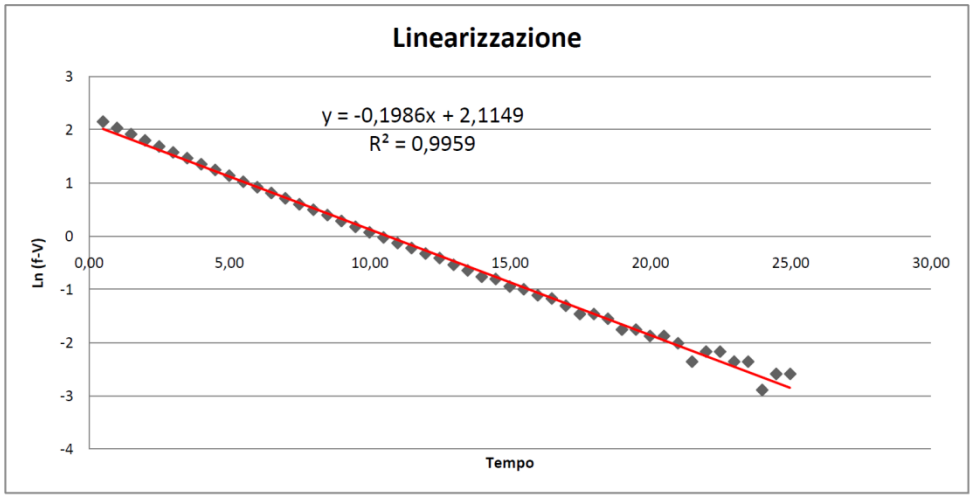
1. $\log_2 32 = 5; \quad \log_3 81 = 4; \quad \log_{10} 100 = 2; \quad \log_{\frac{2}{3}} \frac{8}{27} = 3.$
2. $\log_2 \frac{1}{8} = -3; \quad \log_5 \frac{1}{5} = -1; \quad \log_{\frac{2}{3}} \frac{9}{4} = -2.$
3. $\log_5 5 = l; \quad \log_{0,8} 0,8 = 1; \quad \log_3 1 = 0; \quad \log_{\frac{3}{3}} 1 = 0.$
4. $\log_8 2 = \frac{1}{3}; \quad \log_4 8 = \frac{3}{2}; \quad \log_9 \frac{1}{3} = -\frac{1}{2}.$

Elettricità: concetti di base – 8.96

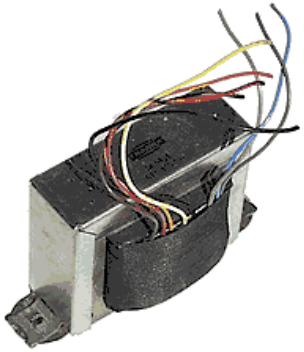
Esperimento: Carica e Scarica di un condensatore

$$V(t) = V_o e^{-\frac{t}{RC}}$$

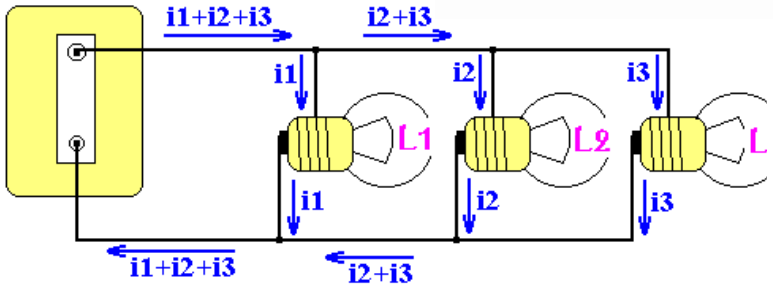
$$\ln V(t) = \ln Vo - t/RC$$

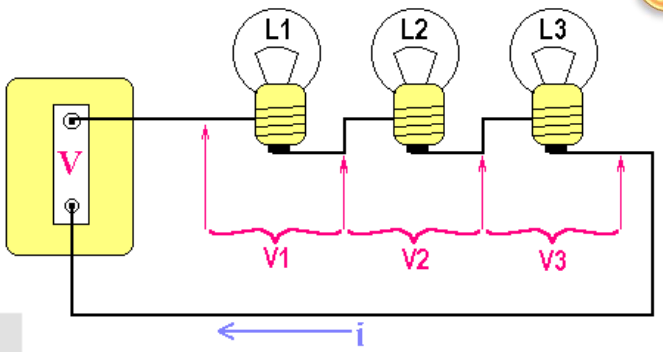


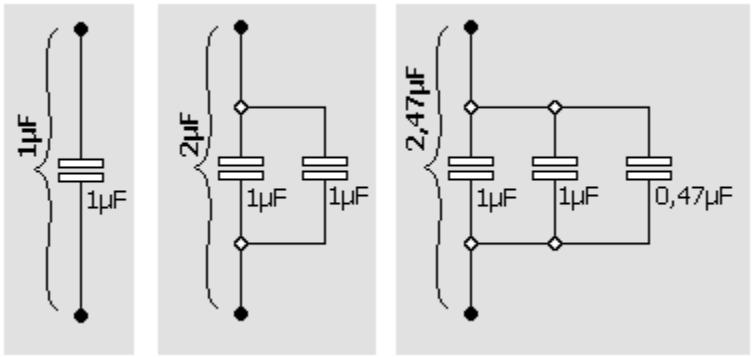
Elettricità: concetti di base – 7

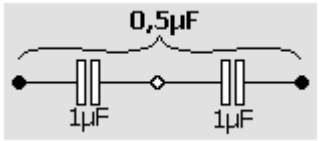


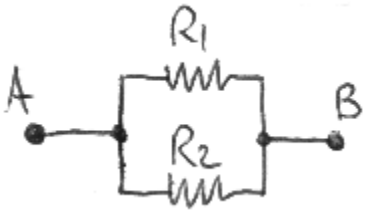
Elettronica!

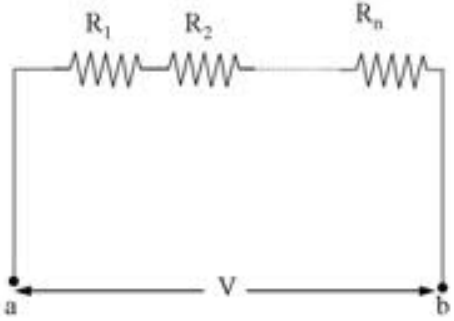












Warning

- Effettuare e/o modificare i collegamenti solo quando la scheda Arduino è scollegata dall’alimentazione!!!

LED comandati da un carattere inviato tramite porta seriale, con e senza toggle

```
int led1Acceso = 0;  // spento! sara' 1 quando acceso!
....
void loop() {
...... leggi da tastiera
  switch (tasto) {
    case '1':
      if (led1Acceso == 0) {
        digitalWrite(1, HIGH);
        led1Acceso = 1;
      }
      else {
        digitalWrite(1, LOW);
        led1Acceso = 0;
      }
      break;
     case '2'
       .....
     }
}
ALTERNATIVA PIU' SEMPLICE
case '1':
      led1Acceso = 1 - led1Acceso;
      digitalWrite(1, led1Acceso);
      break;
```

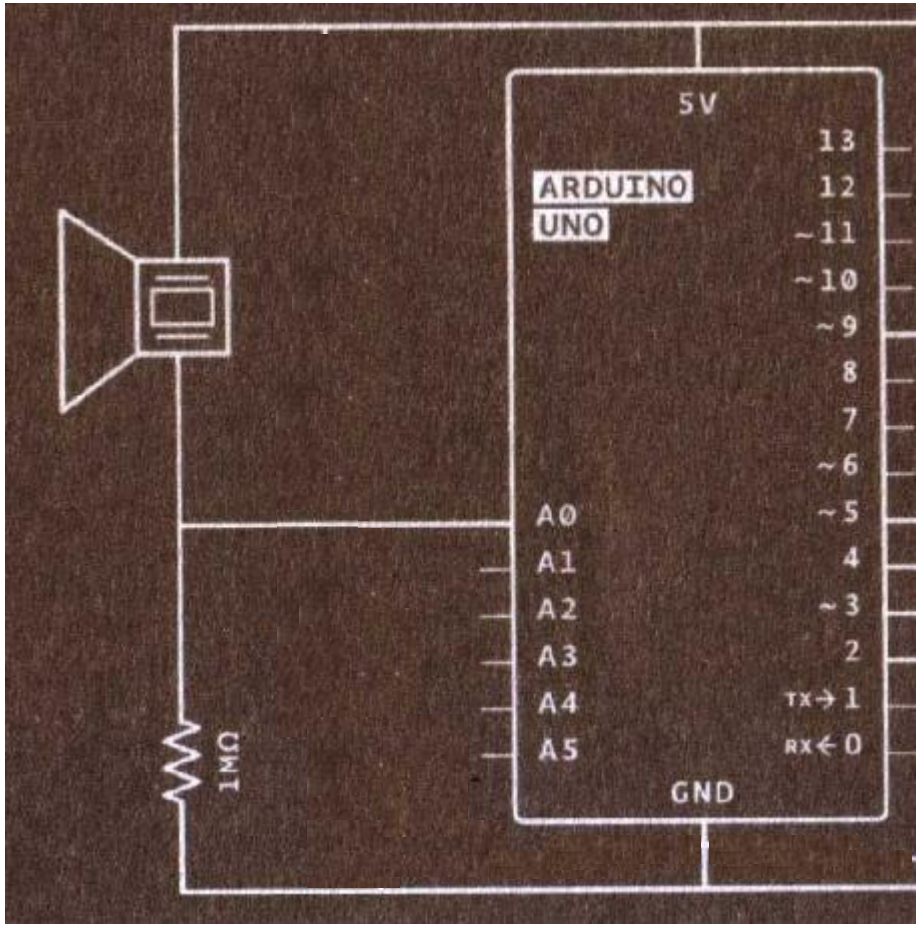
Uso di un pulsante come input

- Realizzare un circuito con arduino, in cui premendo un pulsante collegato ad un pin (per esempio D8), si accenda un LED (esterno, collegato ad un altro pin, per esempio D13); rilasciando il pulsante, il LED si spenga, e così via. Per il collegamento del pulsante, vedere il progetto “spaceship interface” del libro Arduino Projects Book.

Pulsante e toggle dello stato del LED

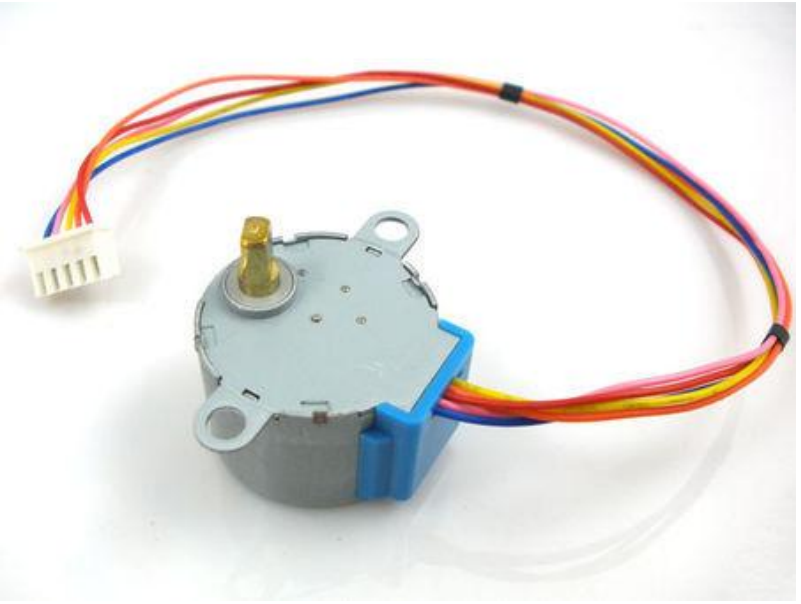
- Come nell'esercizio precedente, però il pulsante ora agirà da "toggle" dello stato, ossia una pressione accenderà il LED, una seconda pressione lo spegnerà, e così via.
- Può essere necessario inserire, nell'if che cambia lo stato del circuito, un delay di circa mezzo secondo in modo che la stessa pressione del pulsante non sia vista come più pressioni successive e quindi cambi più volte lo stato.

Buzzer piezoelettrico come input



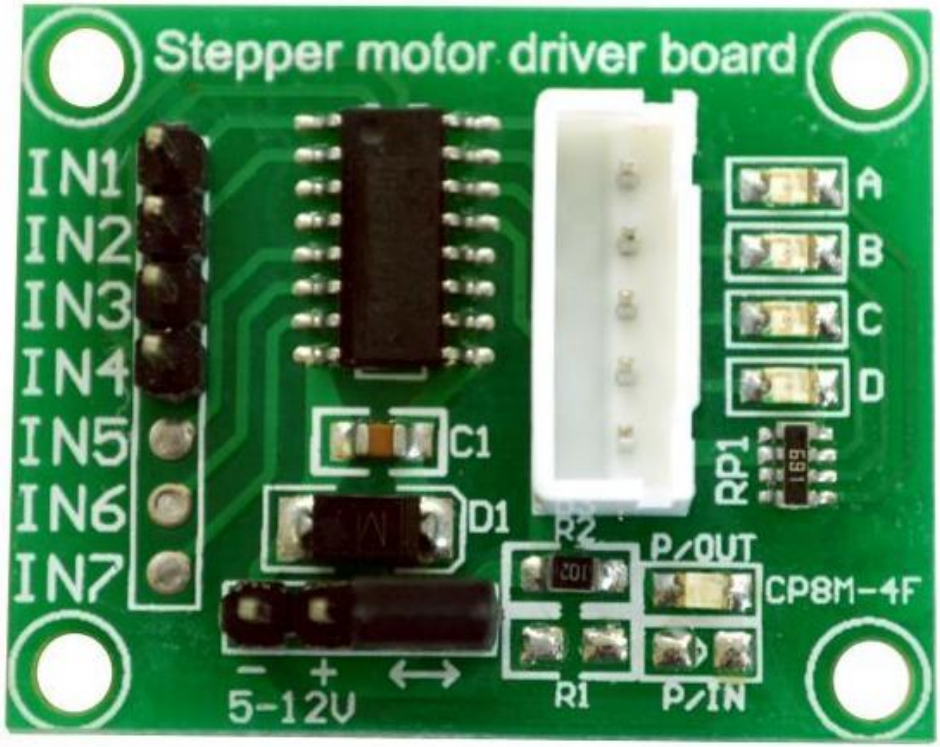

```
const int piezo = A0;
int knock;

void setup() {
  Serial.begin(9600);
}

void loop() {
  knock = analogRead(piezo);
  Serial.println(knock);
}
```

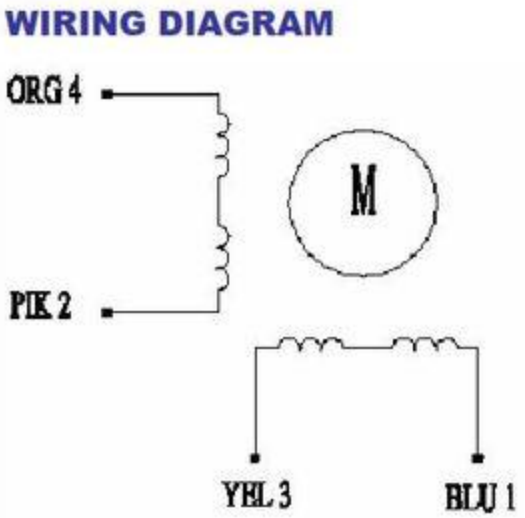
Base robotica con 28-BYJ48 Stepper Motor e driver ULN2003

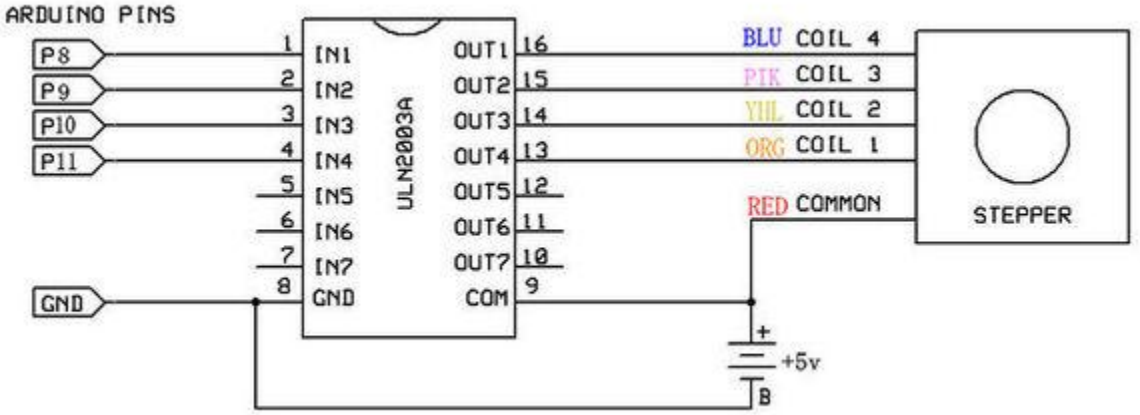


http://www.instructables.com/id/BYJ48-Stepper-Motor/

Base robotica con 28-BYJ48 Stepper Motor e driver ULN2003







http://www.instructables.com/id/BYJ48-Stepper-Motor/

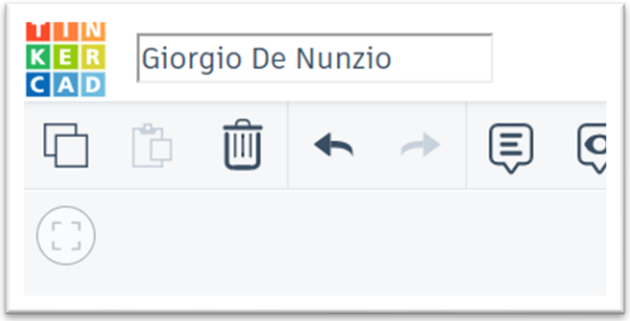
Base robotica con 28-BYJ48 Stepper Motor e driver ULN2003

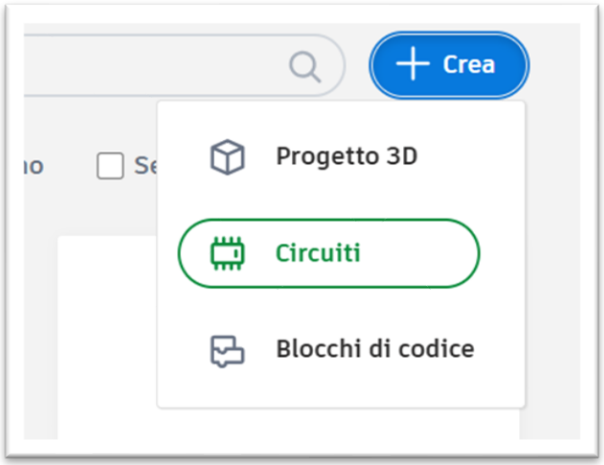
Filmato

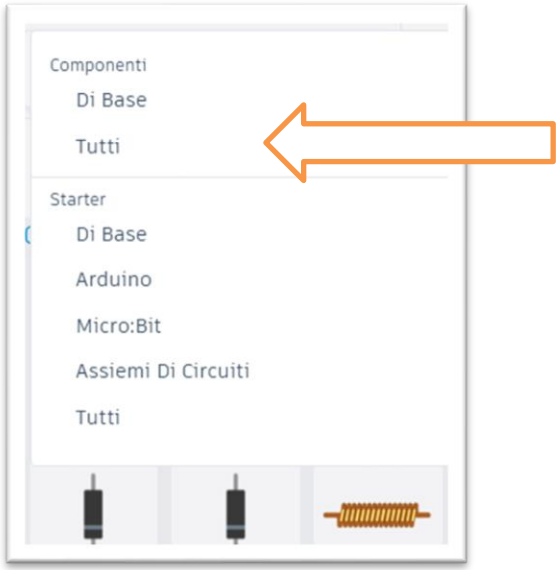
http://42bots.com/tutorials/28byj-48-stepper-motor-with-uln2003-driver-and-arduino-uno/

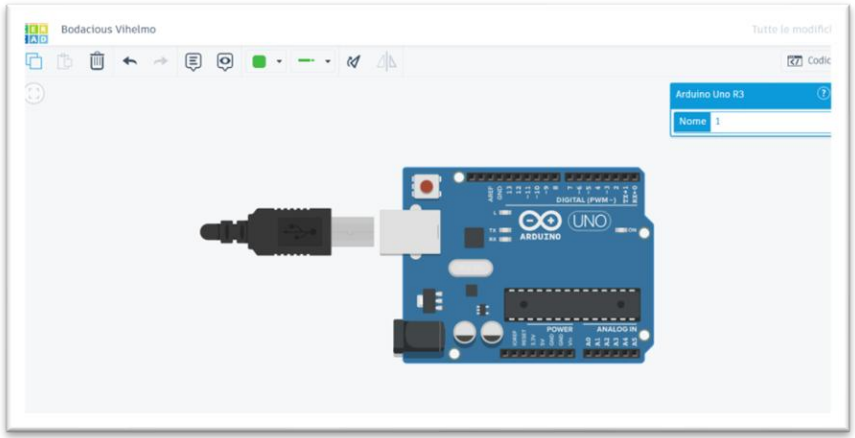
http://www.instructables.com/id/BYJ48-Stepper-Motor/

Tinkercad https://www.tinkercad.com/

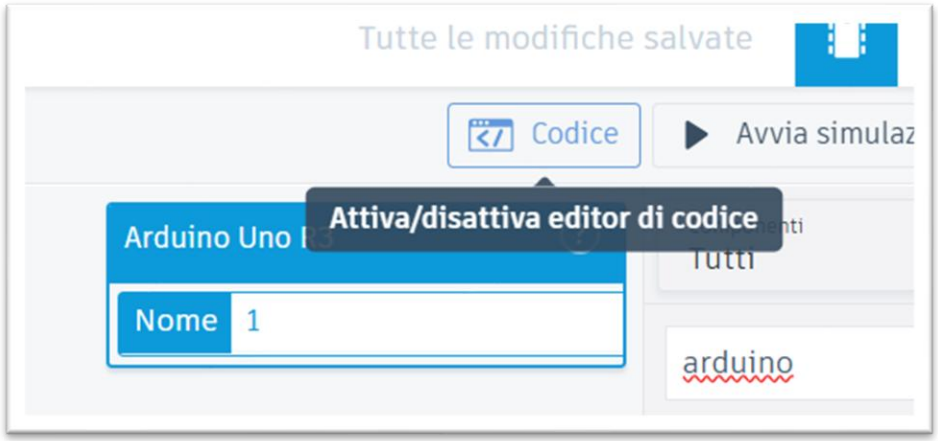






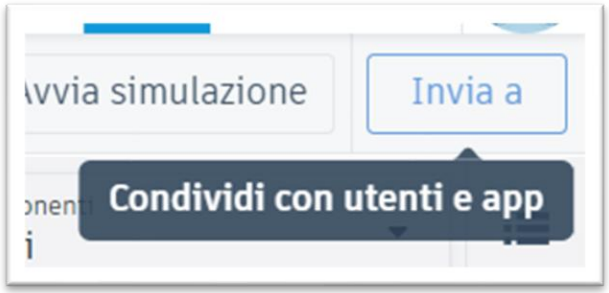


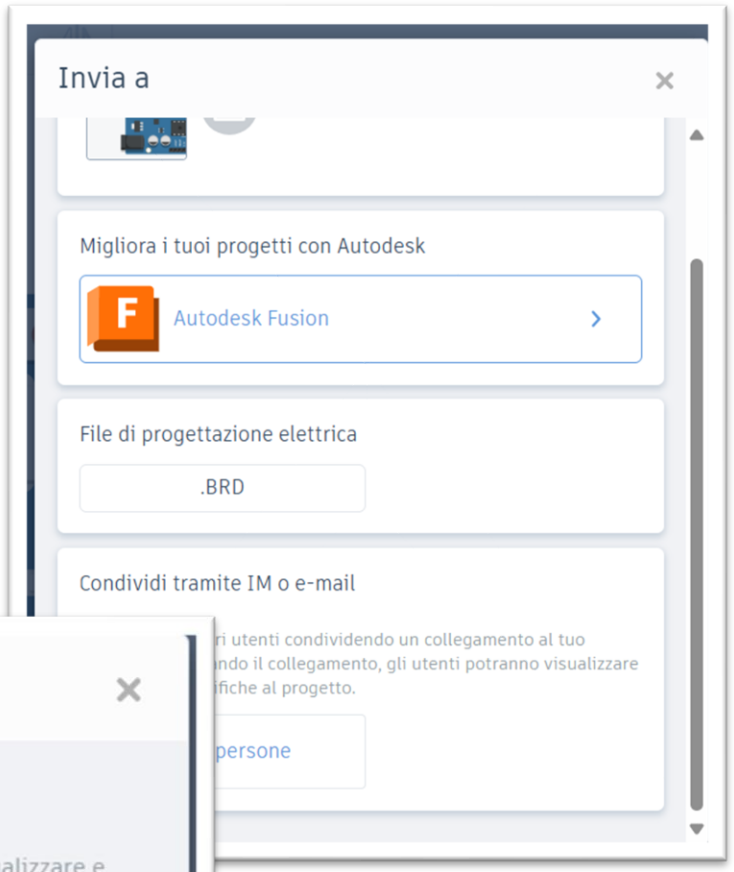
Tinkercad

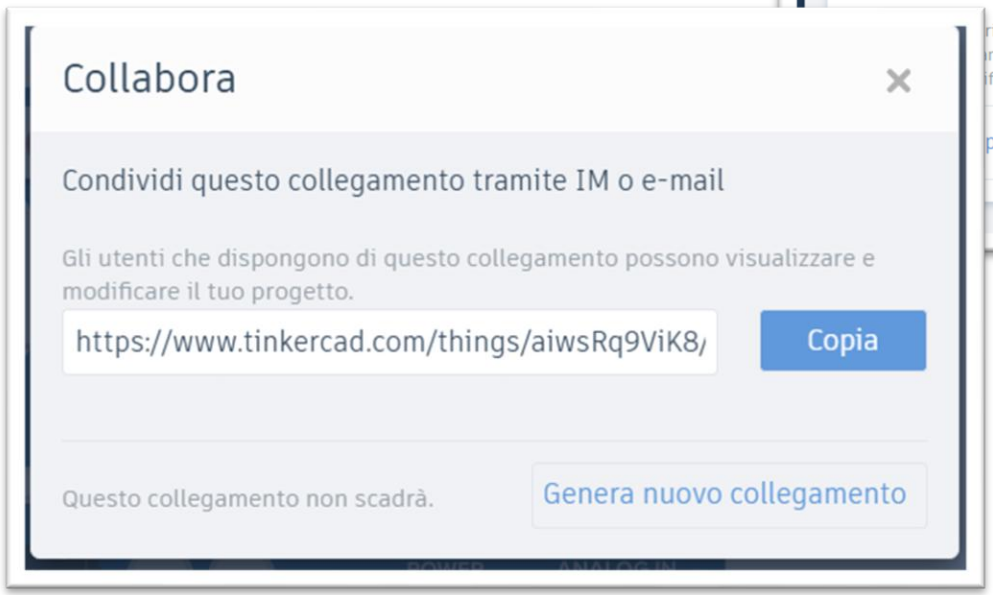

Tutte le modifiche salvate
Codice
Avvia simulaz
Attiva/disattiva editor di codice
Arduino Uno
Nome
1
arduino

Codice
Avvia simulazione
Arduino Uno R3
Nome
1
Testo
1 (Arduino
// C++ code
//
void setup()
{
pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
digitalWrite(LED_BUILTIN, HIGH);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(LED_BUILTIN, LOW);
delay(1000); // Wait for 1000 millisecond(s)
}
AREF
GND
DIGITAL (PWM~)
UNO
ARDUINO
TX
RX

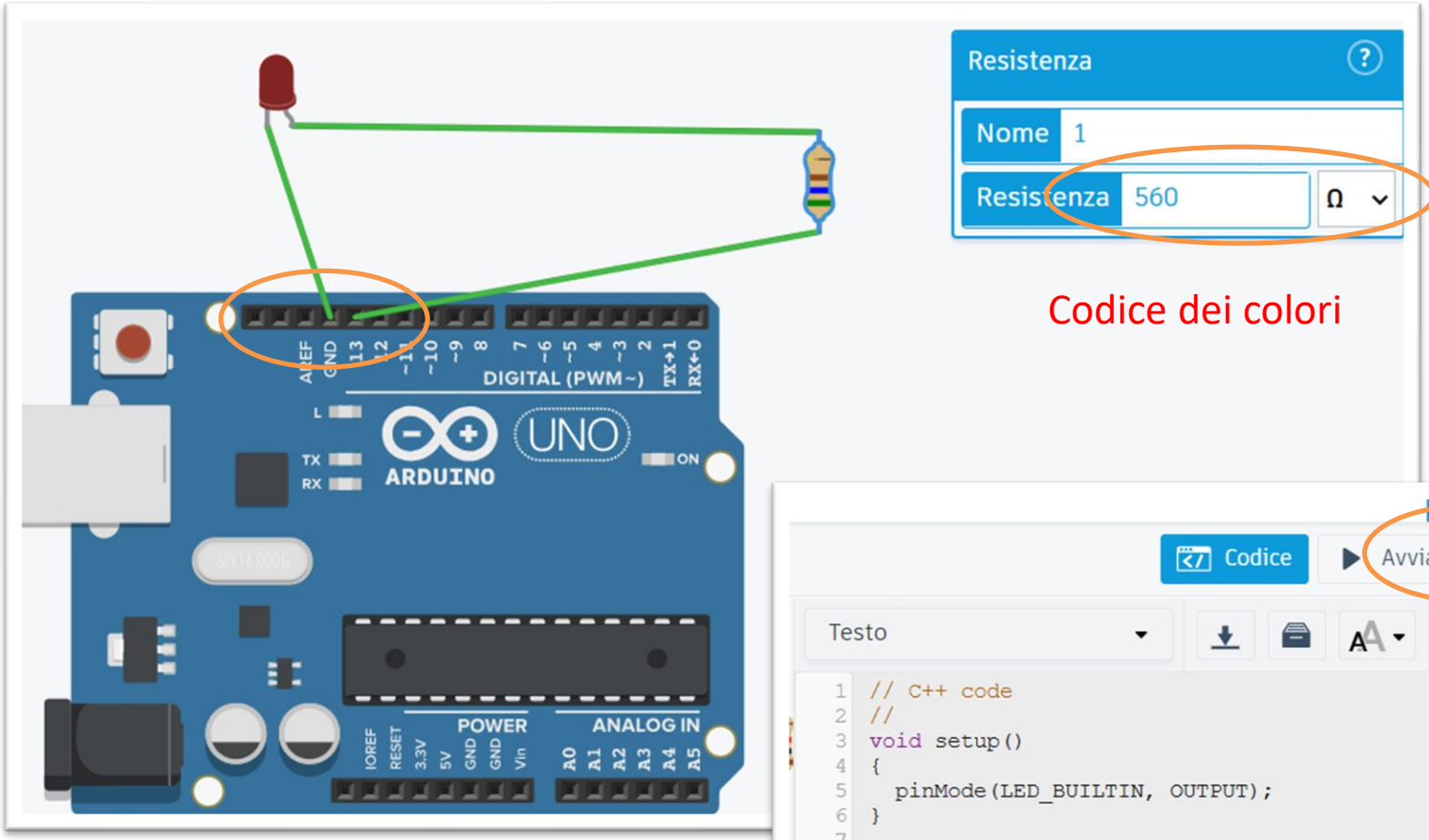
Tinkercad


Avvia simulazione
Invia a
Condividi con utenti e app

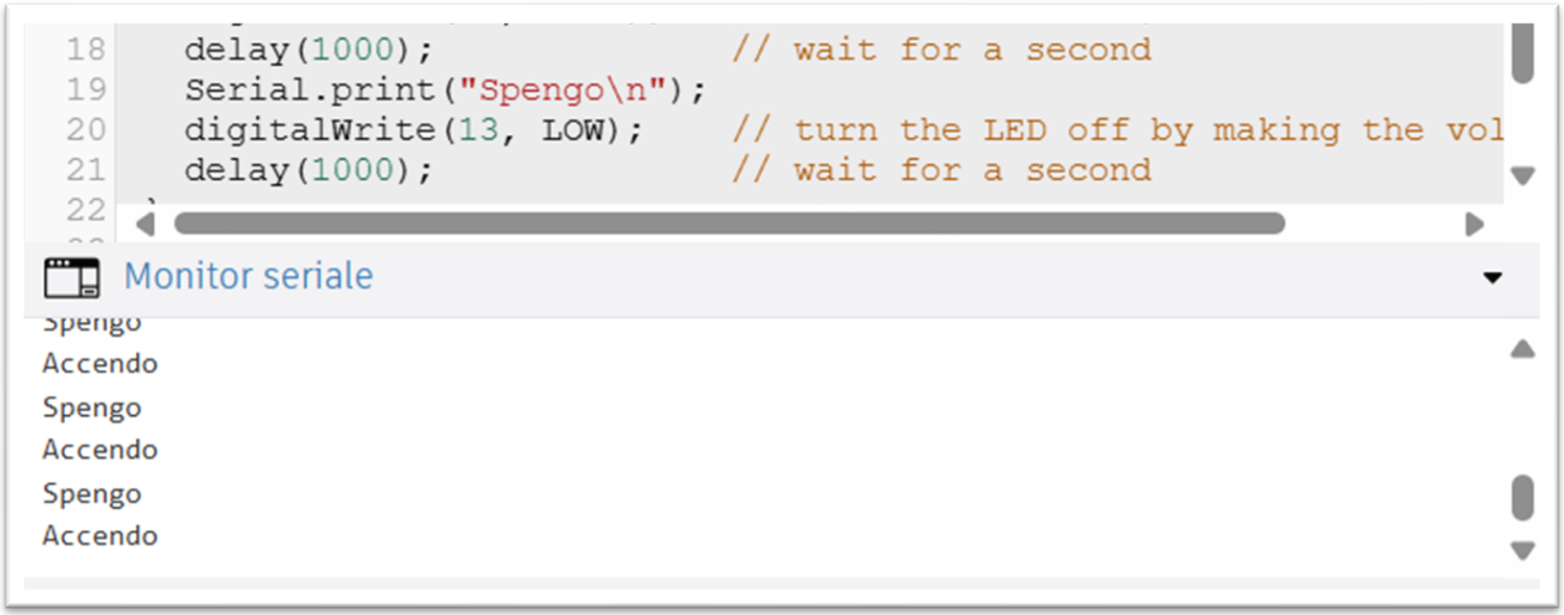

Invia a
Migliora i tuoi progetti con Autodesk
Autodesk Fusion
File di progettazione elettrica
.BRD
Condividi tramite IM o e-mail
persone


Collabora
Condividi questo collegamento tramite IM o e-mail
Gli utenti che dispongono di questo collegamento possono visualizzare e modificare il tuo progetto.
https://www.tinkercad.com/things/aiwsRq9ViK8/
Copia
Questo collegamento non scadrà.
Genera nuovo collegamento

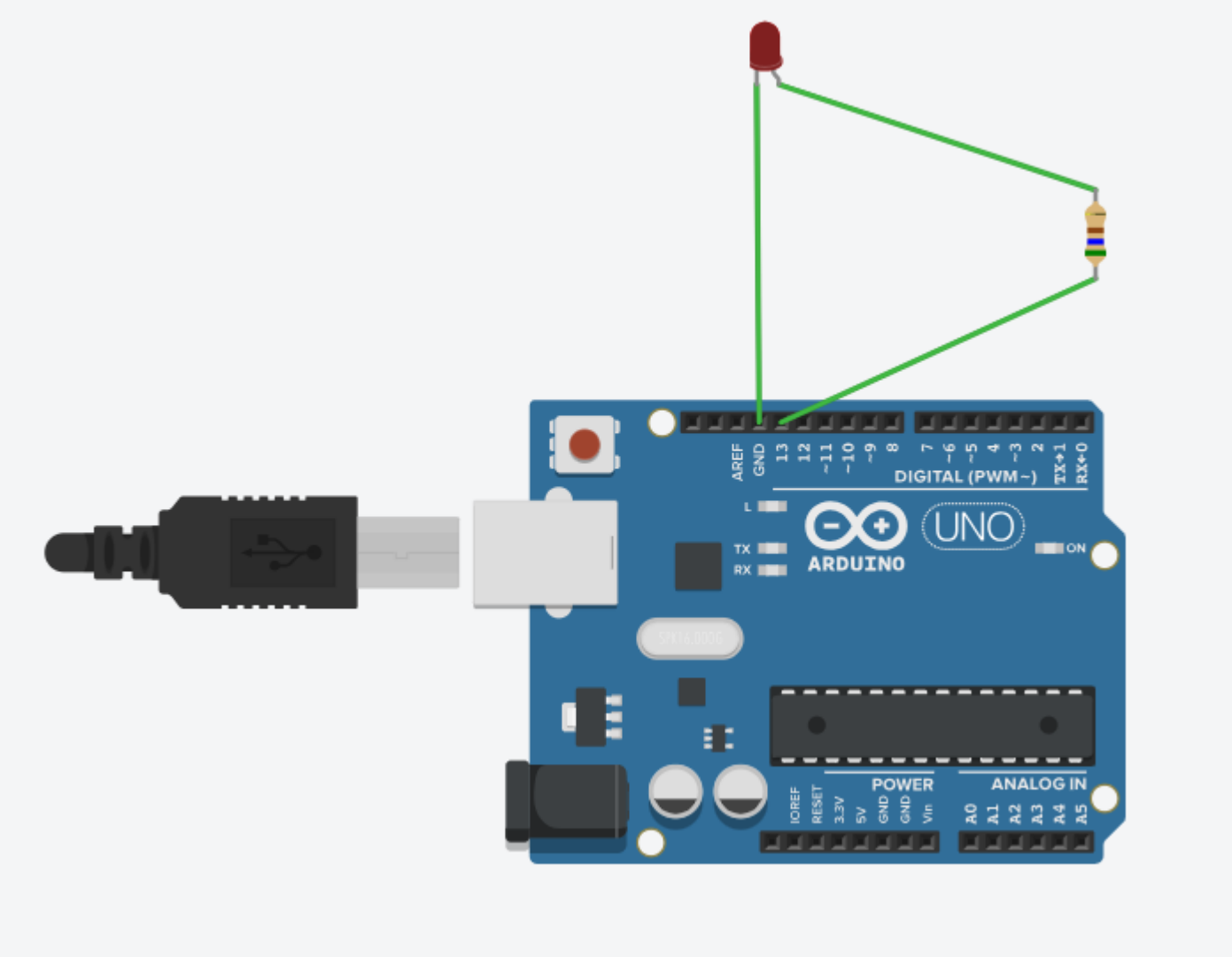
Tinkercad


Resistenza
Nome 1
Resistenza 560
Ω
Codice dei colori
DIGITAL (PWM~)
UNO
ARDUINO
POWER
ANALOG IN
Codice
Avvia simulazione
Invia a
Testo
1 (Arduino Uno R3)
// C++ code
//
void setup()
{
pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
digitalWrite(LED_BUILTIN, HIGH);
delay(1000); // Wait for 1000 millisecond(s)
digitalWrite(LED_BUILTIN, LOW);
delay(1000); // Wait for 1000 millisecond(s)
}

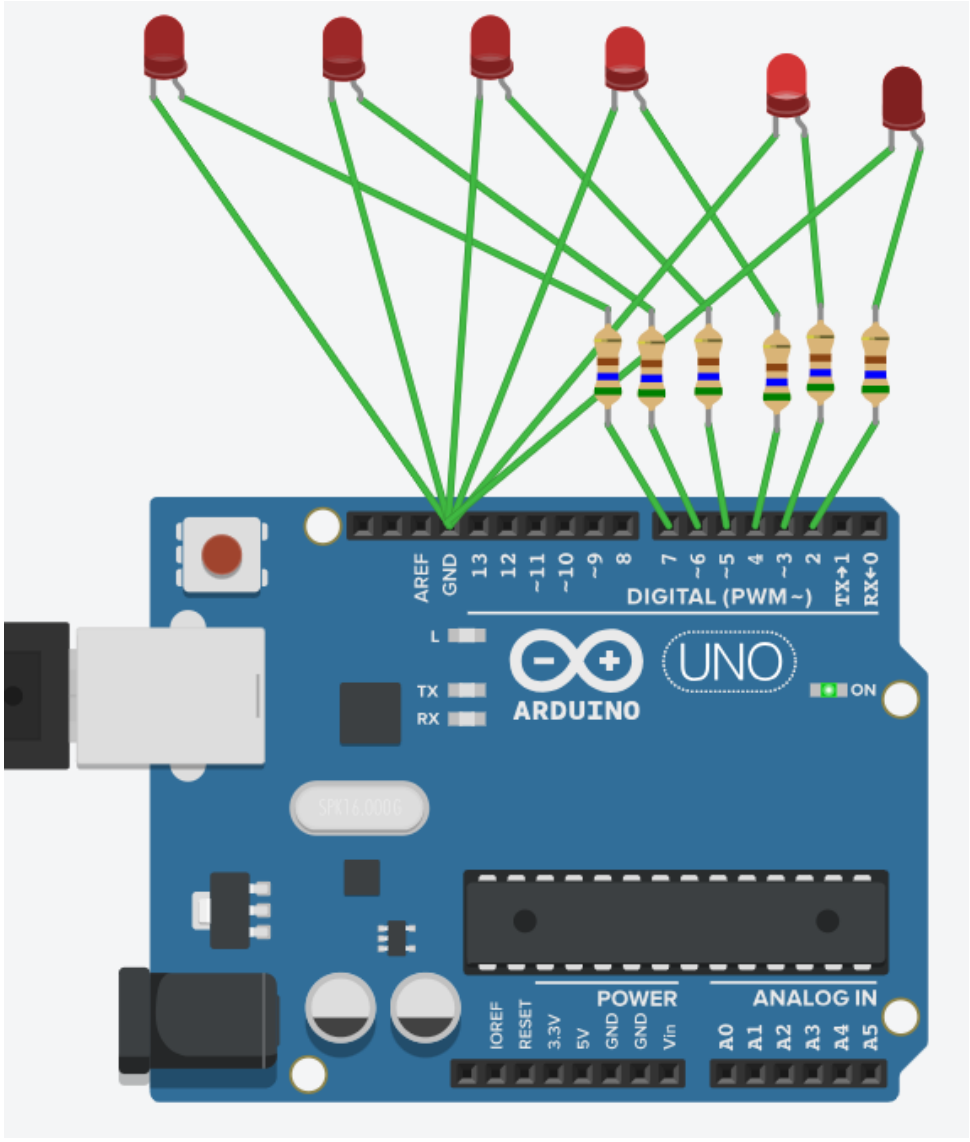
Tinkercad


18 delay(1000); // wait for a second
19 Serial.print("Spengo\n");
20 digitalWrite(13, LOW); // turn the LED off by making the vol
21 delay(1000); // wait for a second
22
Monitor seriale
Spengo
Accendo
Spengo
Accendo
Spengo
Accendo

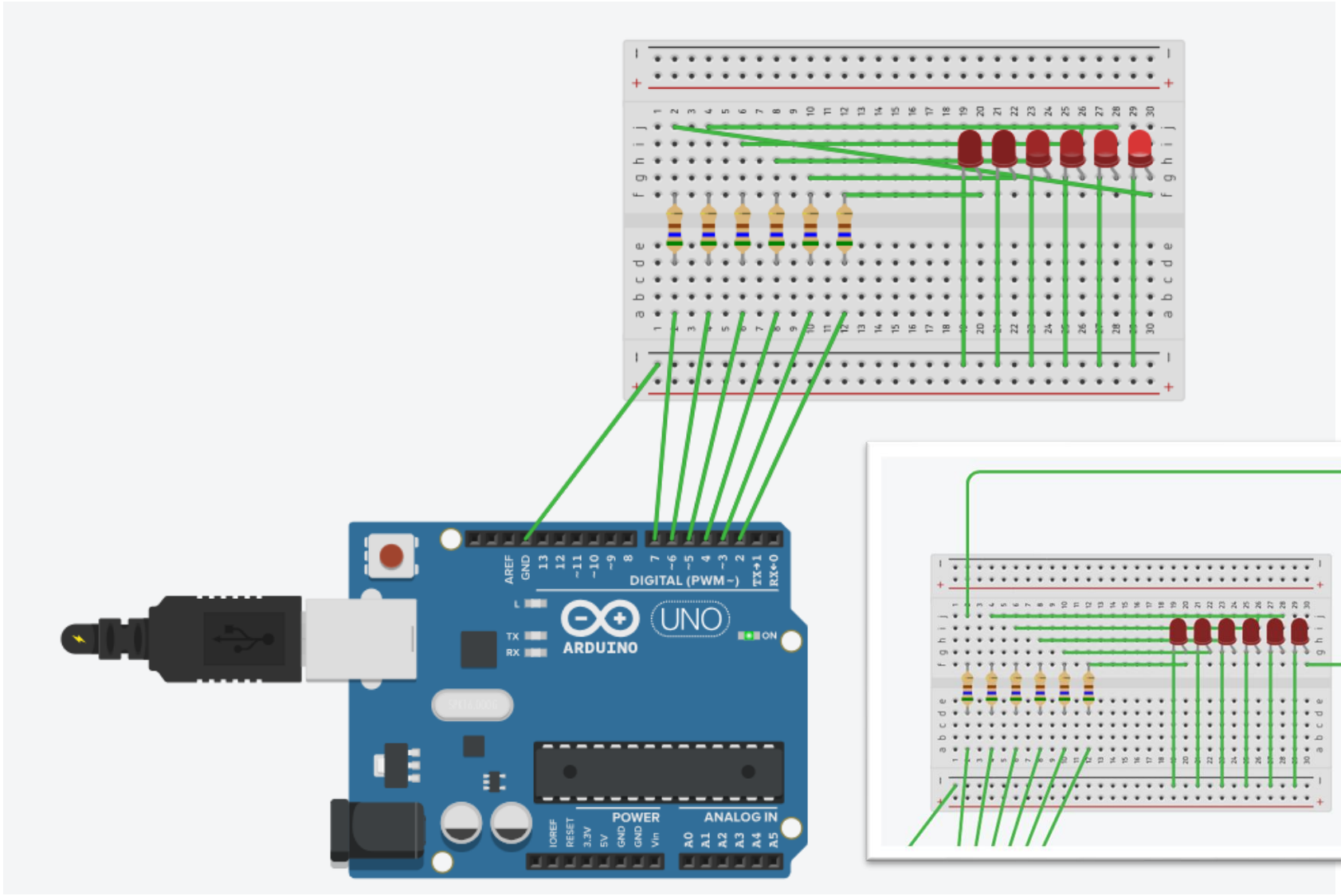
Tinkercad


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Tinkercad


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5

Tinkercad



Biblio/sitografia

- E’ stato largamente attinto (con piccole modifiche) dal sito
  http://www.zeppelinmaker.it/indice-degli-argomenti/
- Sito per la pulizia del codice in caso di copia-incolla dal ppt o pdf: https://tools360.net/it/text-cleaner (selezionare solamente «Rimuovi i caratteri non ASCII»)
- Open collector: https://www.build-electronic-circuits.com/open-collector-outputs/
- Audio Analysis: https://www.compadre.org/osp/pwa/soundanalyzer/, https://www.qsl.net/dl4yhf/spectra1.html, https://scope.software.informer.com/download/#downloading (scope 1.40)