CORSO AVANZATO DI INFORMATICA E ROBOTICA

LEZIONE 1: FONDAMENTI

Carlo Zambaldo | 29 ottobre 2025

Breve sondaggio

Quanti di voi userebbero il PC personale?

Dobbiamo installare:

- ARDUINO IDE
- Visual Studio Code (o equivalenti, es Python IDLE)
- FileZilla

Indice

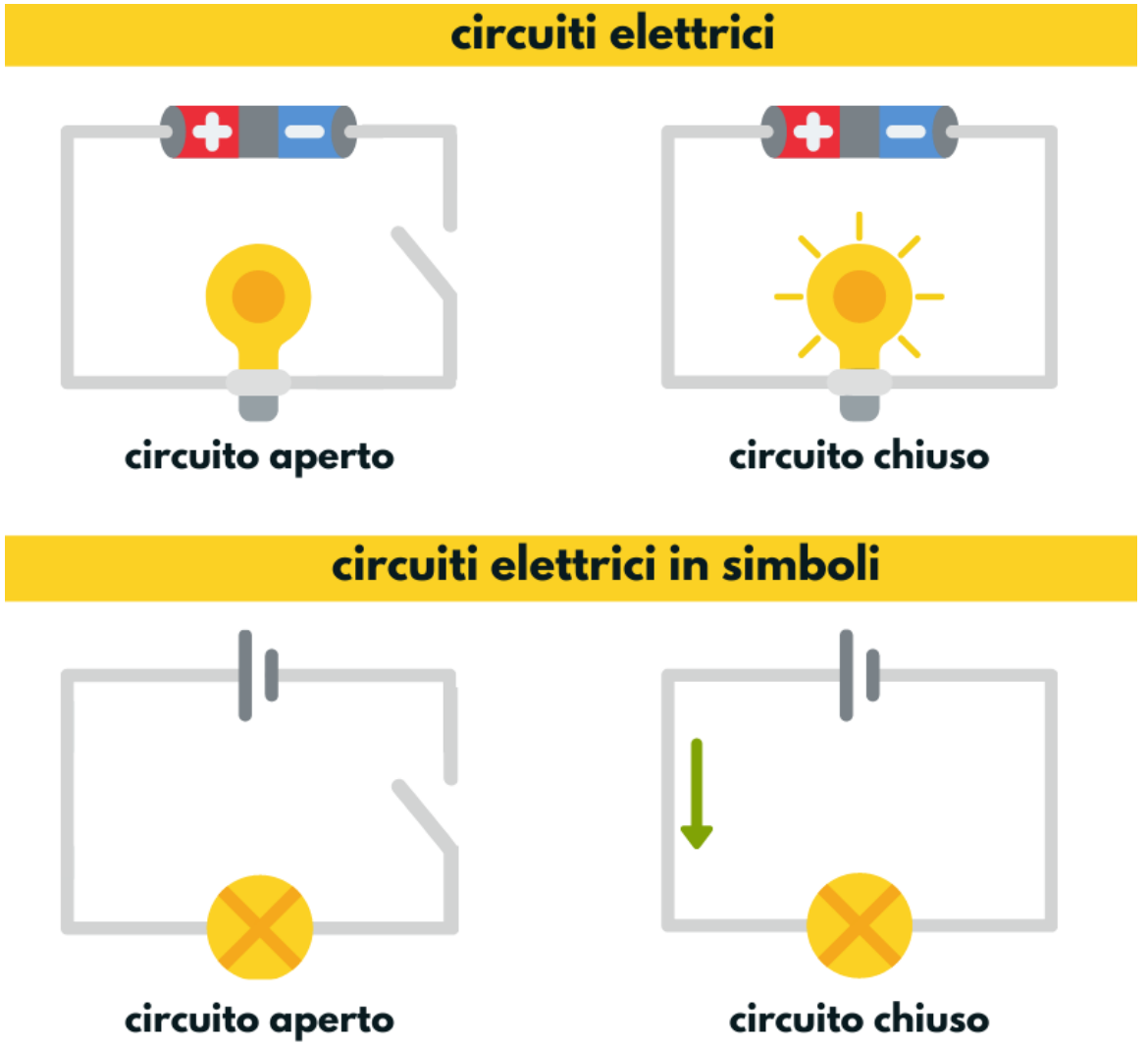
Cos'è un Circuito Elettrico?

Un circuito elettrico è un percorso chiuso in cui le cariche elettriche possono scorrere continuamente grazie a una differenza di potenziale (tensione) tra i suoi estremi.

Per funzionare, un circuito necessita di componenti come:

- un generatore,
- un utilizzatore
- dei fili conduttori che colleghino il tutto.

L'apertura o la chiusura del circuito, ad esempio tramite un interruttore, determina se la corrente può fluire o meno



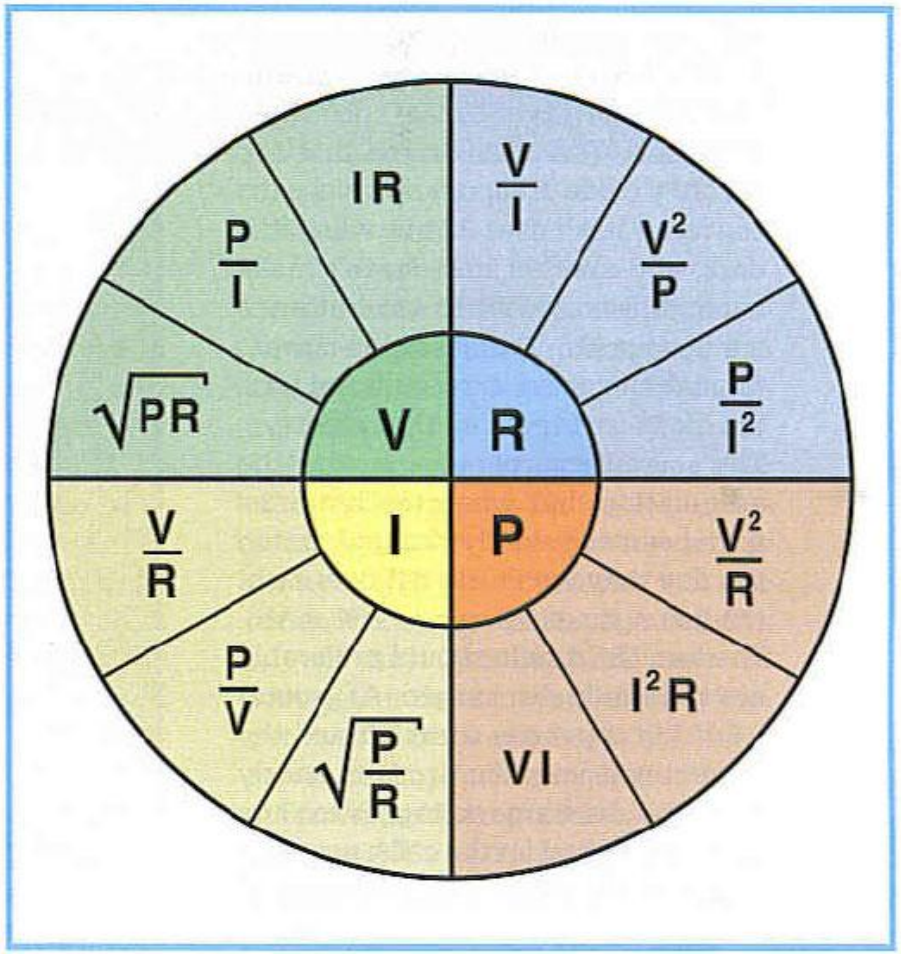
Le Grandezze Fondamentali e la Legge di Ohm

Le grandezze fondamentali dell'elettronica sono:

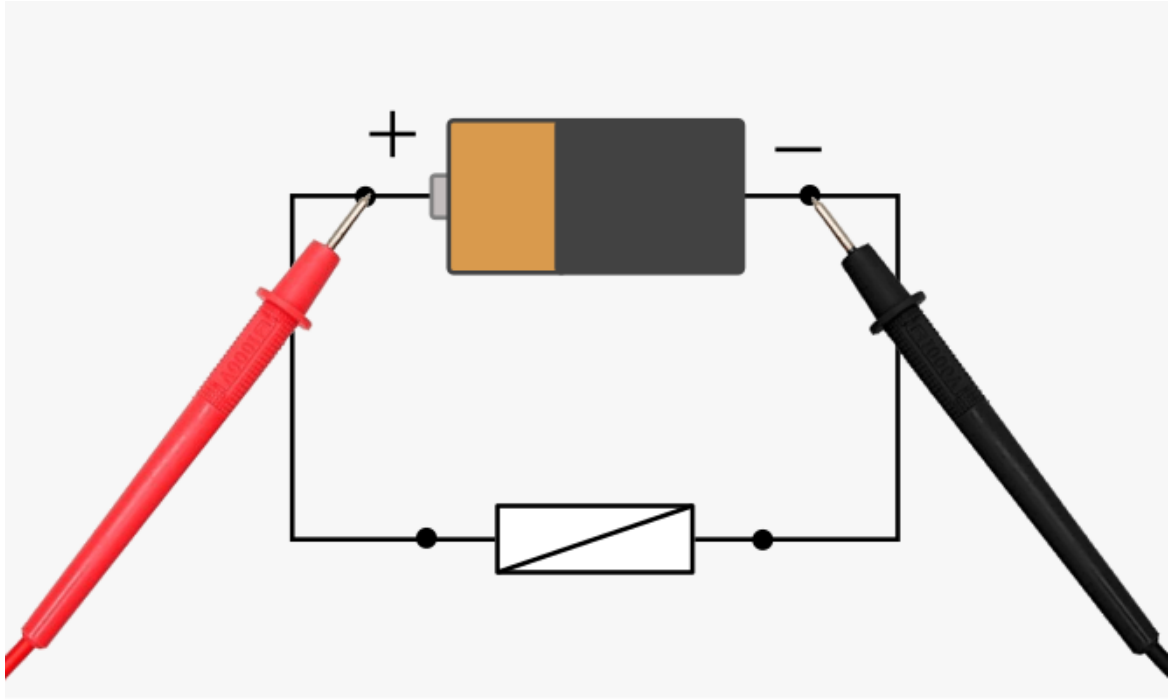
- Tensione (V): Il lavoro per unità di carica necessario per muovere una carica tra due punti. Si misura in Volt (V).
- Corrente (I): La quantità di carica che attraversa una sezione di un conduttore nell'unità di tempo (I=dQ/dt). Si misura in Ampere (A).
- Resistenza (R): L'opposizione al passaggio della corrente elettrica. Si misura in Ohm (Ω).

La relazione che governa i circuiti lineari è la Legge di Ohm:

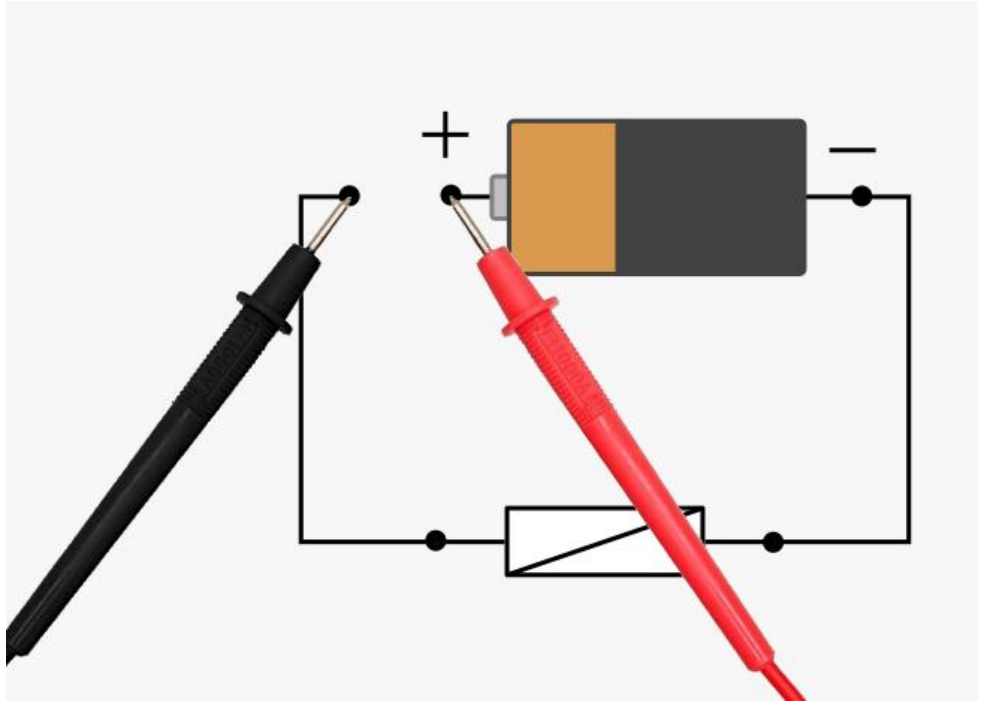
$$V = I \cdot R$$



Multimetro: come misurare tensione e corrente



Misurare una tensione
(Voltmetro)

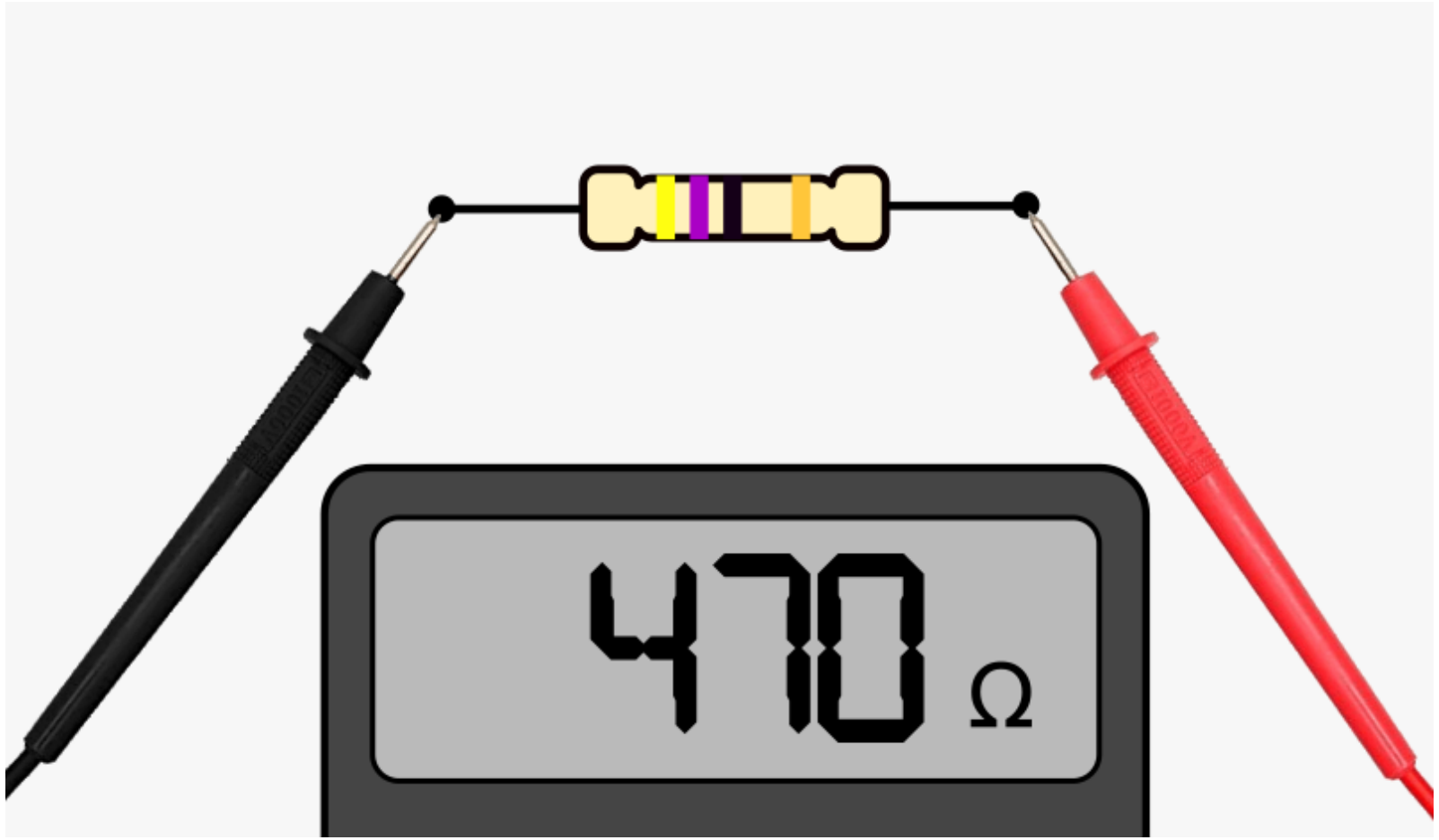


Misurare una corrente
(Amperometro)

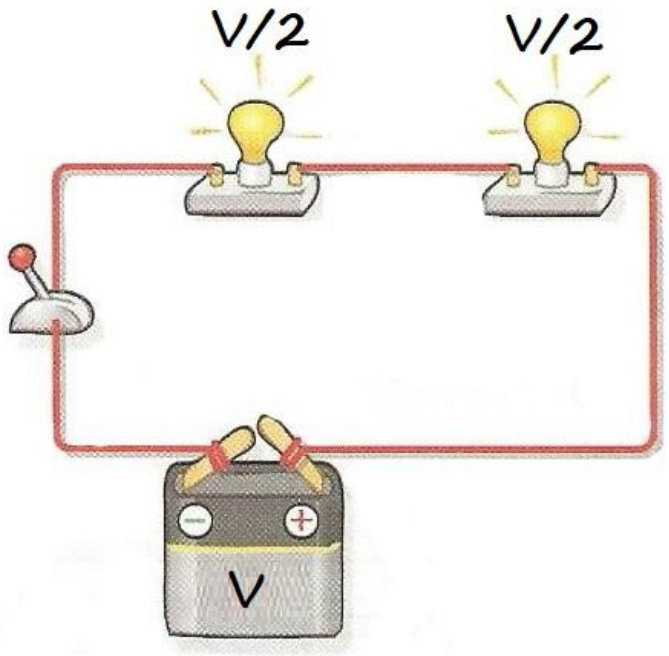
Misurare una resistenza

Scollegare sempre il componente di cui si vuole misurare la resistenza dal circuito!!

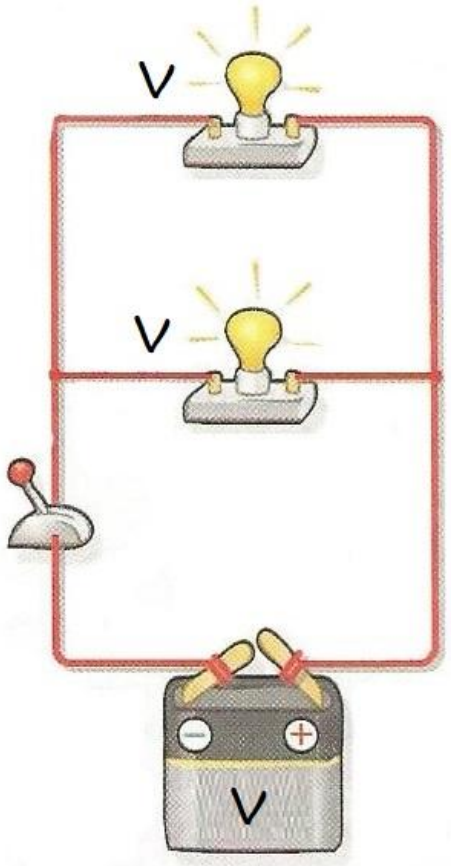
Altrimenti la misura risulta falsata dalla resistenza equivalente del circuito stesso



Circuiti in Serie e Parallelo



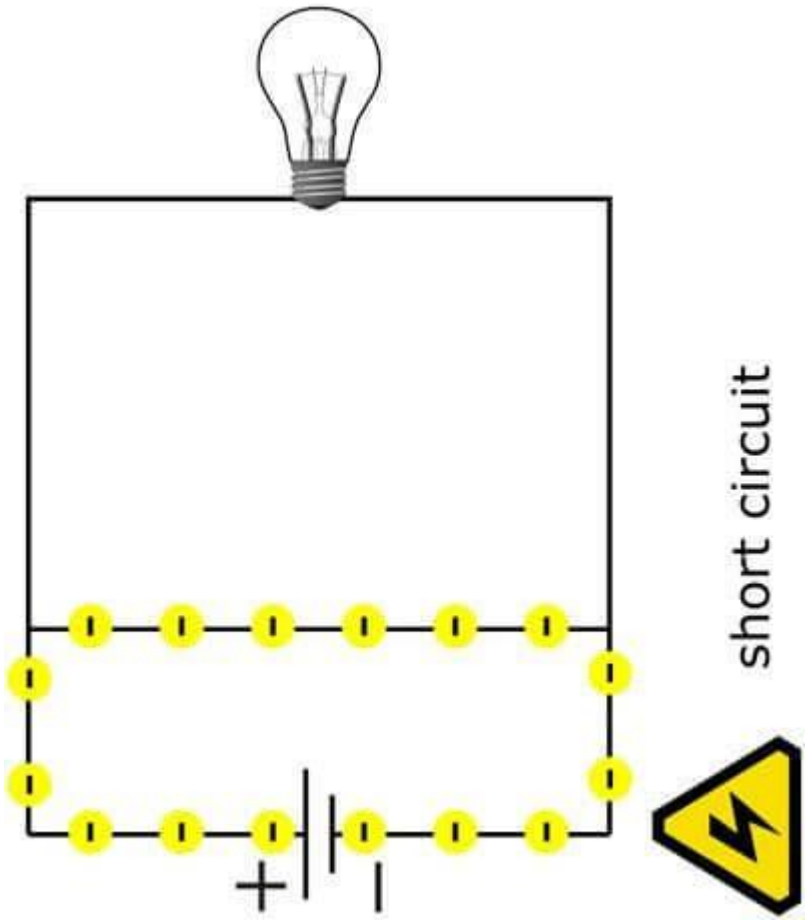
CIRCUITO IN SERIE

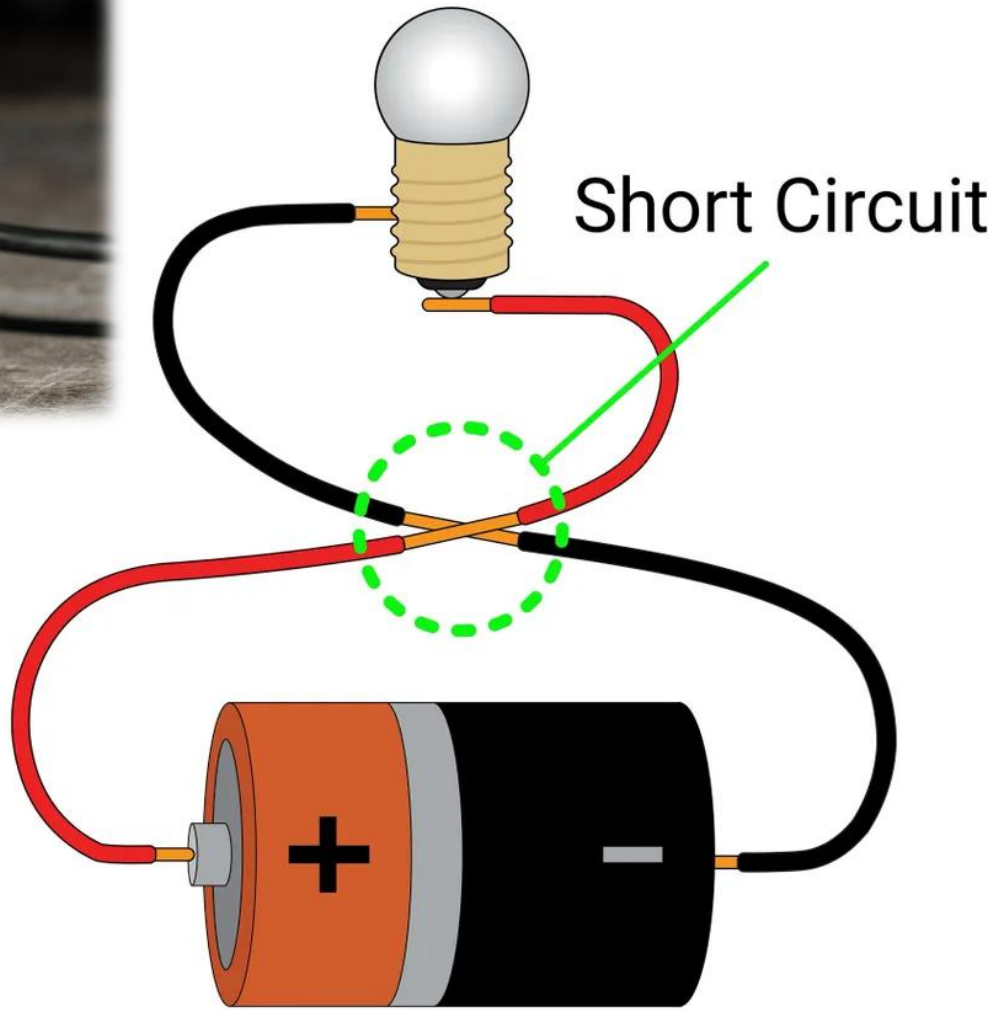


CIRCUITO IN PARALLELO

- https://youtu.be/m4jzgqZu-4s?si=cTzJr0a-QoY2Dn91
- https://youtu.be/G3H5lKoWPpY?si=UBy4Sw1Oczr3cXbO

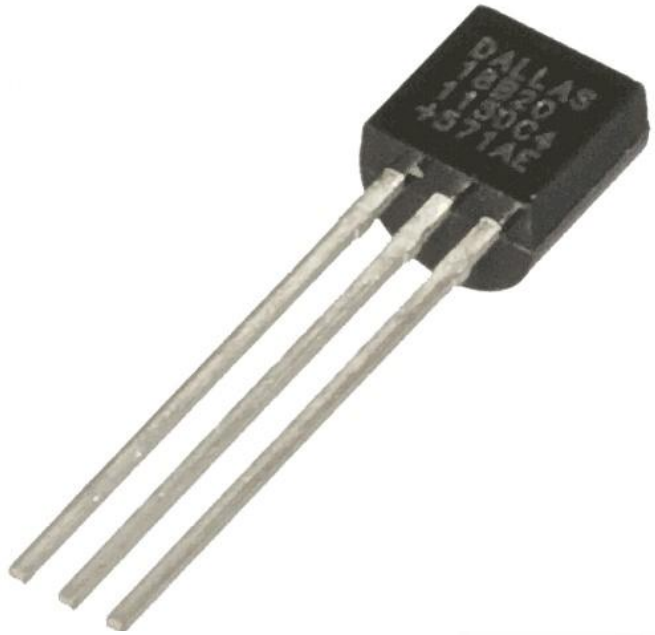
Cortocircuito

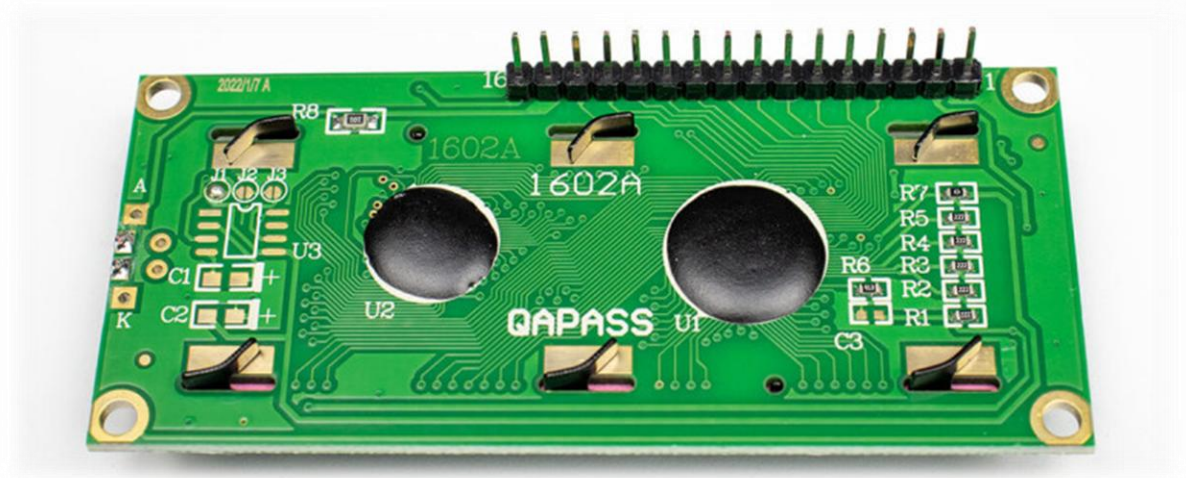


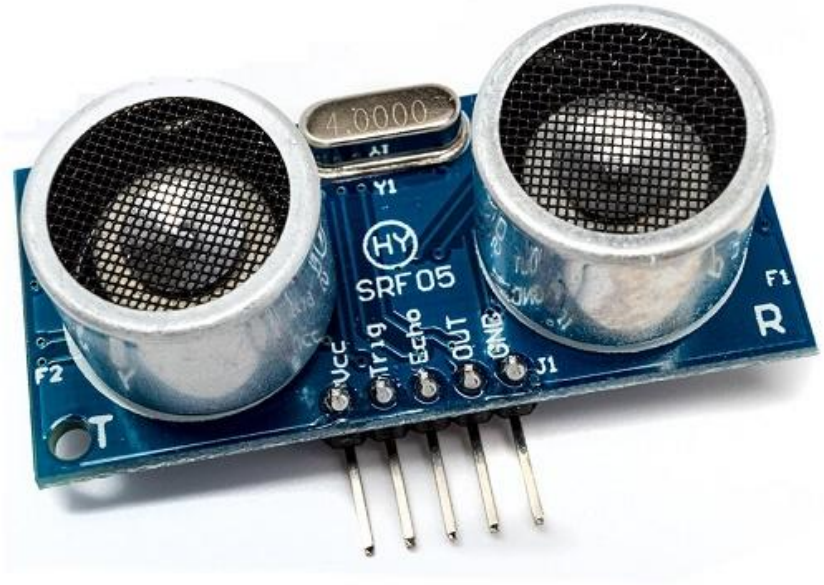


Un altro rischio: l'inversione di polarità

L'inversione della polarità può danneggiare seriamente i componenti elettronici non protetti, provocando un'ampia gamma di malfunzionamenti, dal semplice non funzionamento fino al surriscaldamento, ai cortocircuiti e alla distruzione completa del dispositivo.

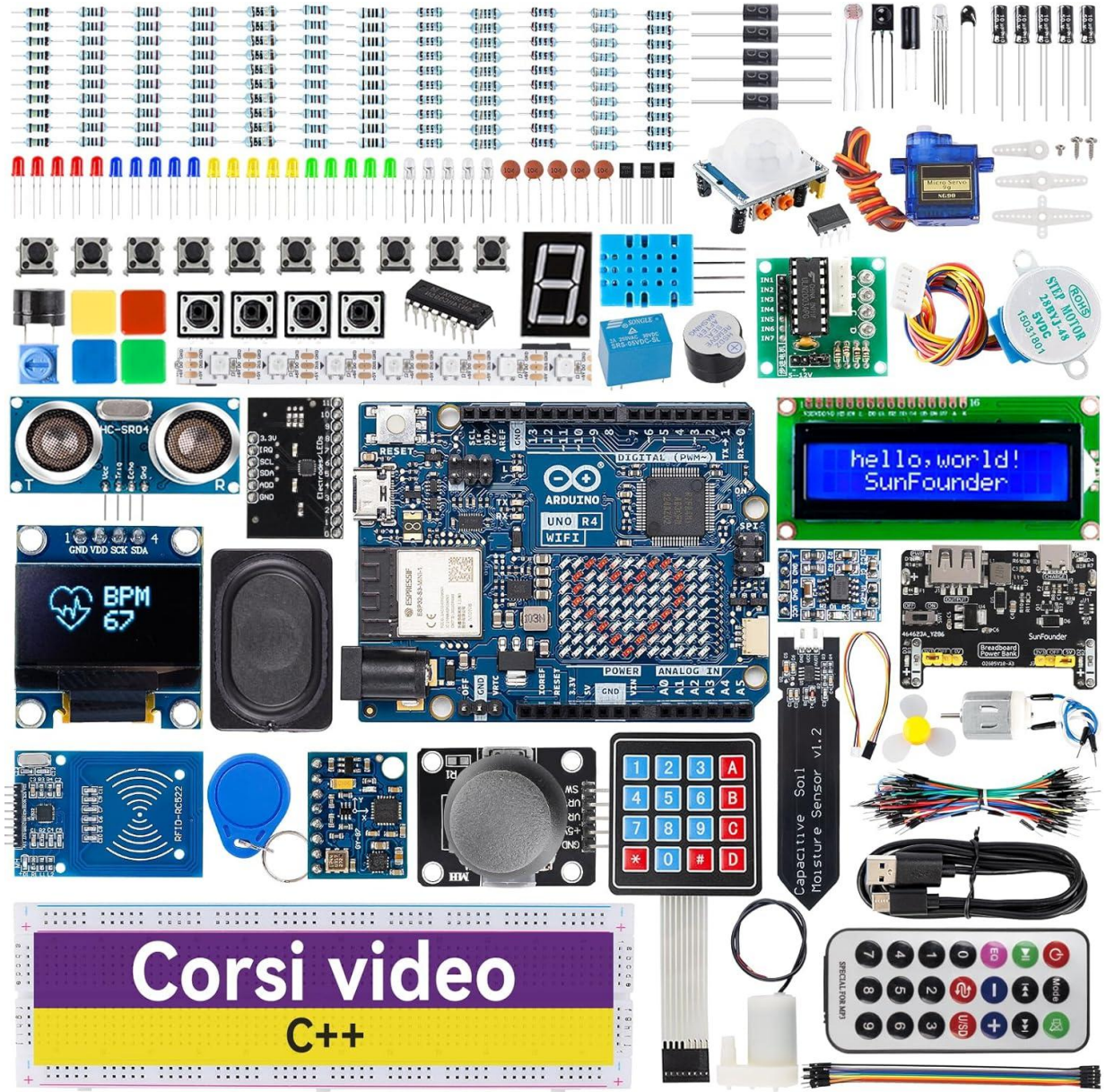






Componenti

- Resistori
- Lampadine ad incandescenza
- LED e diodi
- Condensatori
- Induttori
- Display LCD
- Display LED
- Relè
- Microfono
- Sensori Vari

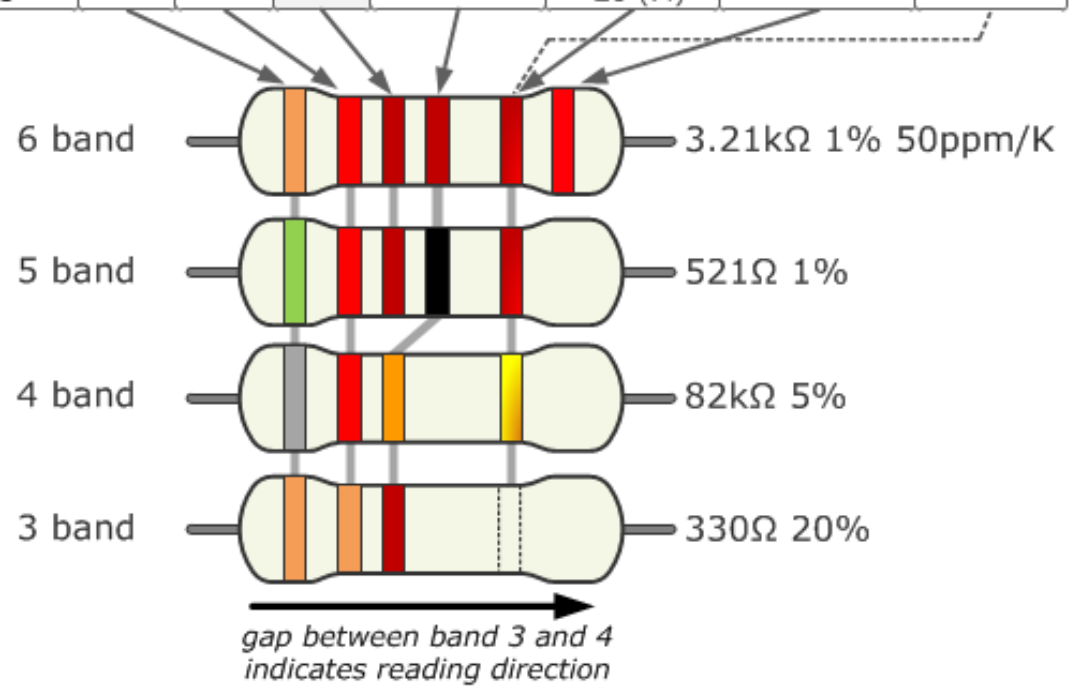


Resistori

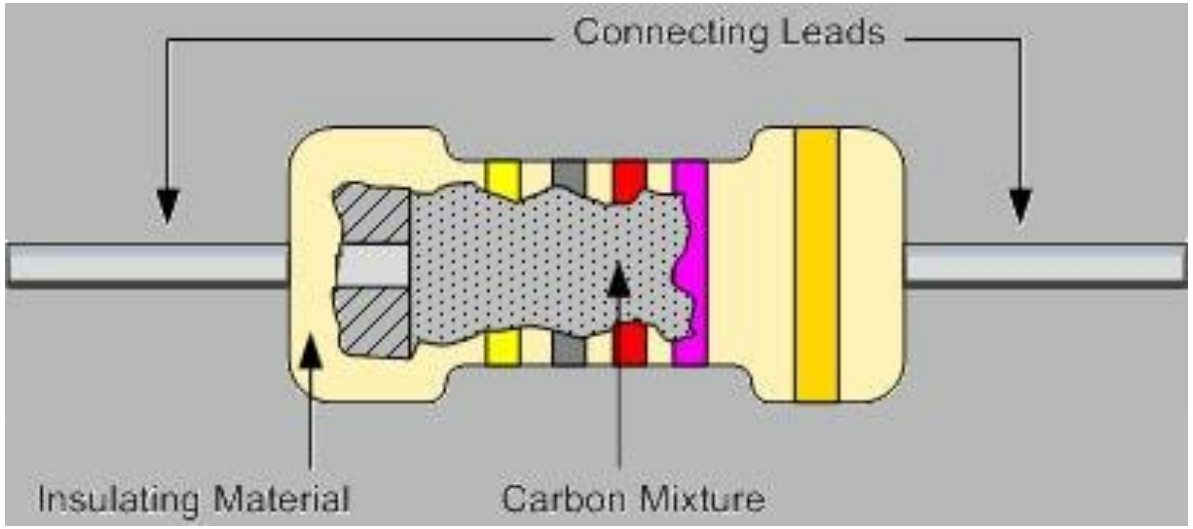
American version

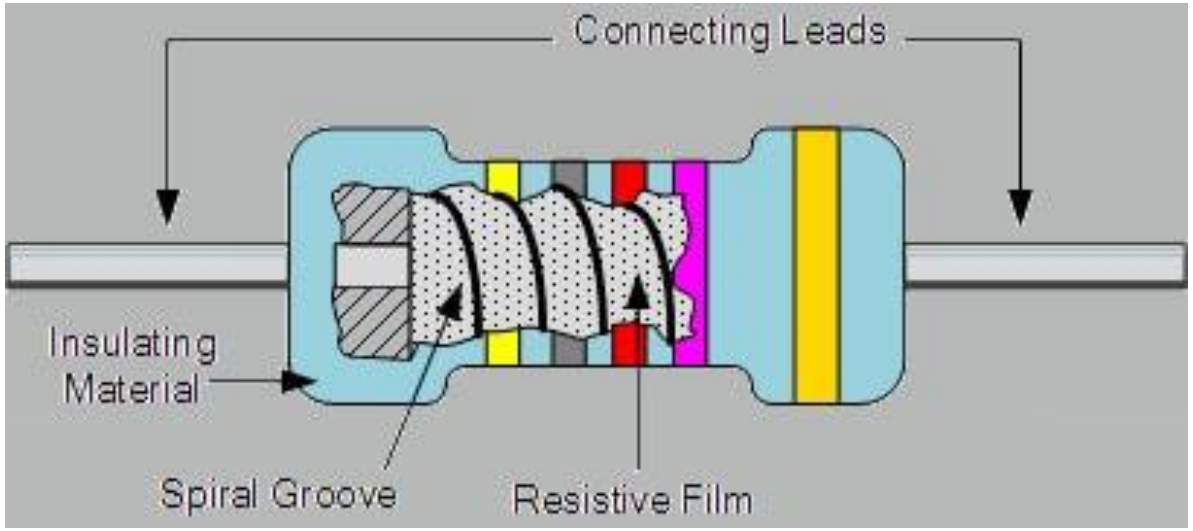
International Electrotechnical Commission version

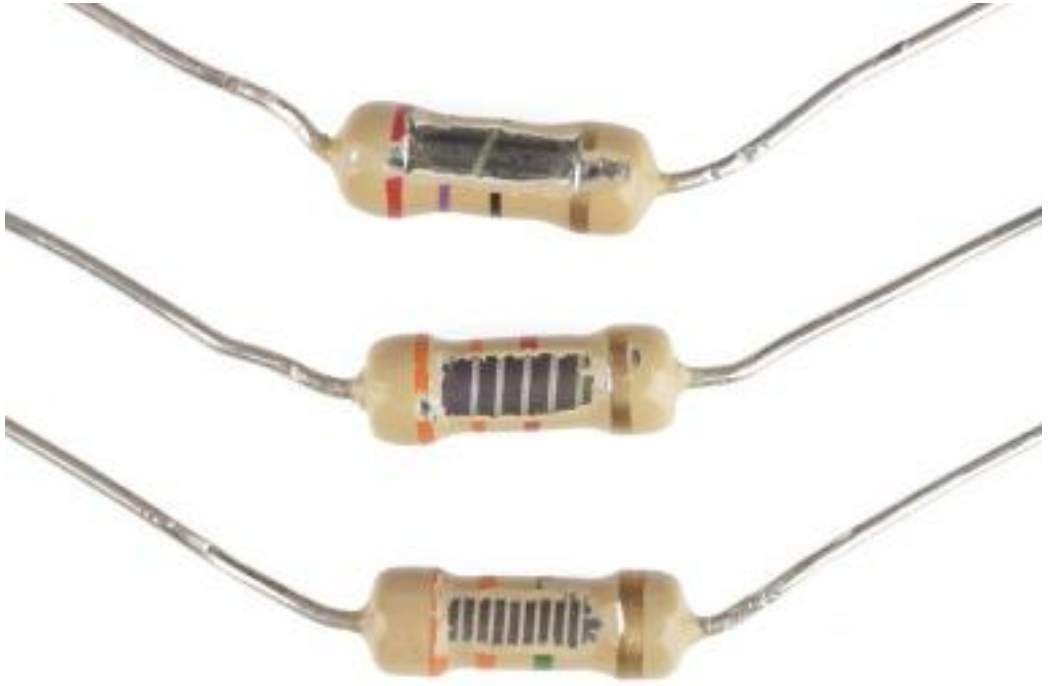
	Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		



Resistori – come sono fatti



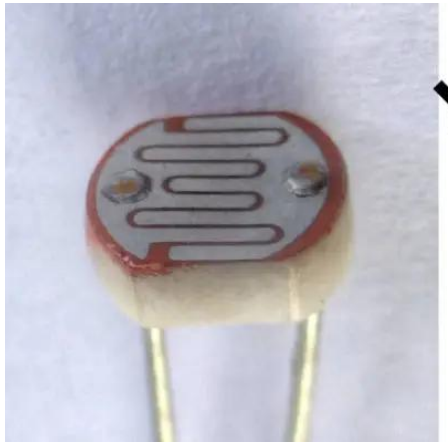


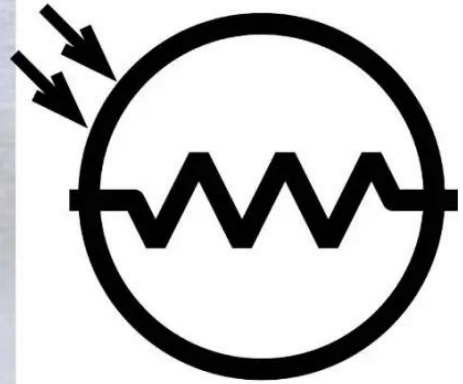


Componenti a resistenza variabile

Fotoresistenza

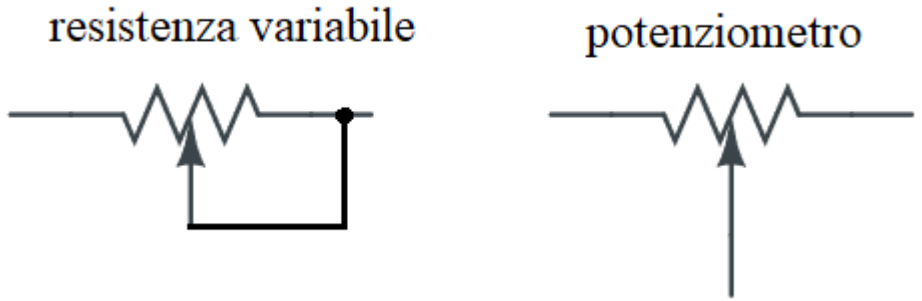
Componente elettronico la cui resistenza elettrica varia in modo inversamente proporzionale all'intensità della luce che la colpisce.





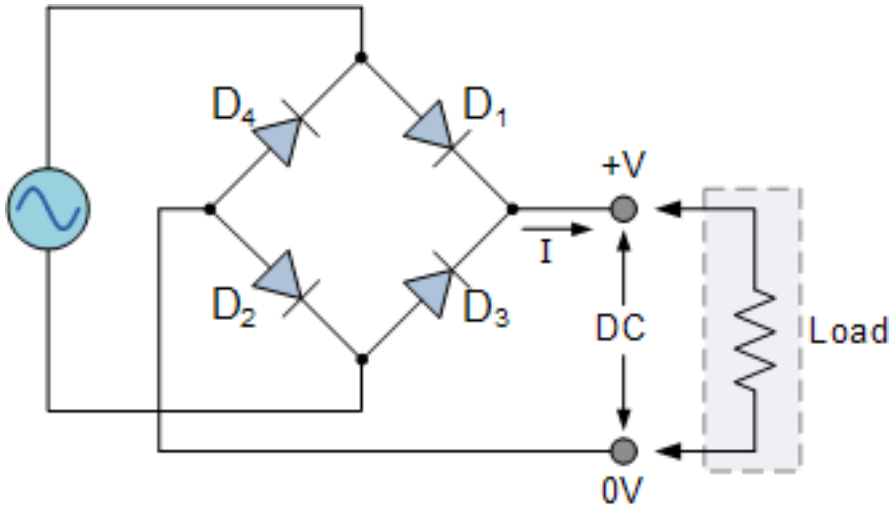
Potenziometro

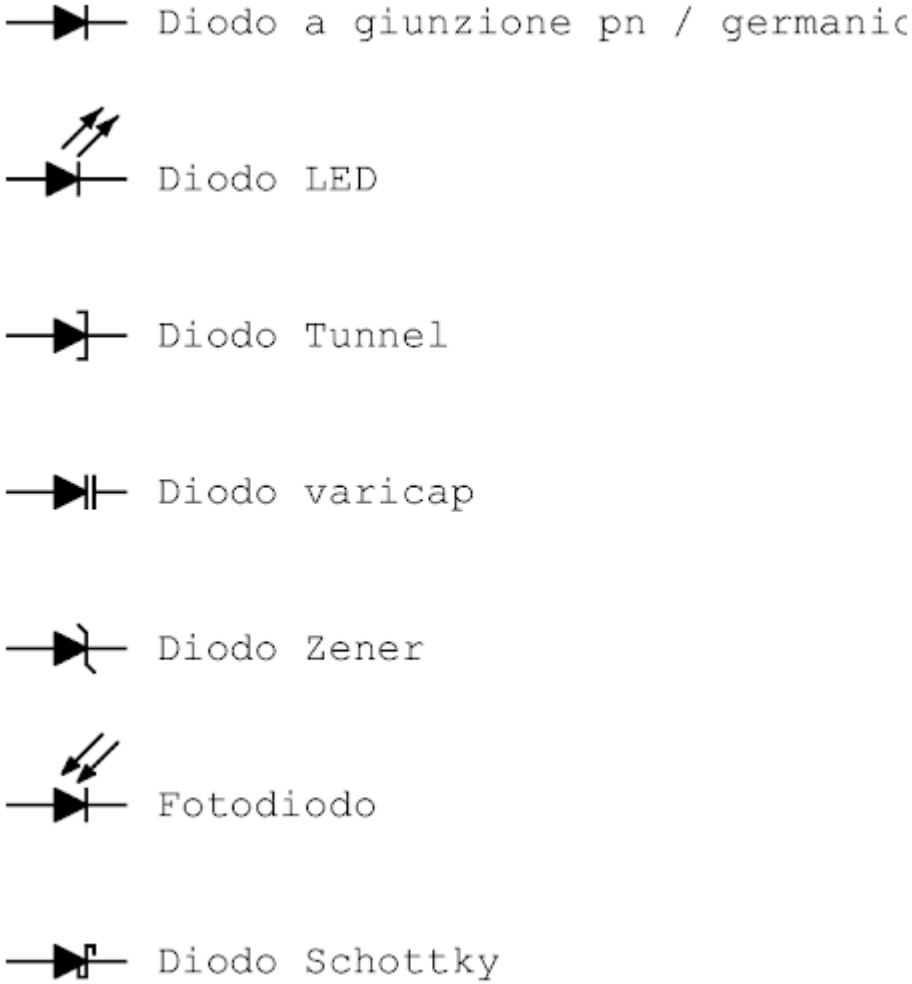
Un potenziometro è una resistenza variabile. Serve a dividere una tensione elettrica. Controlla la tensione in uscita, non la corrente. Possiede tre terminali e viene spesso usato per regolare l'intensità di volume, luce ecc.



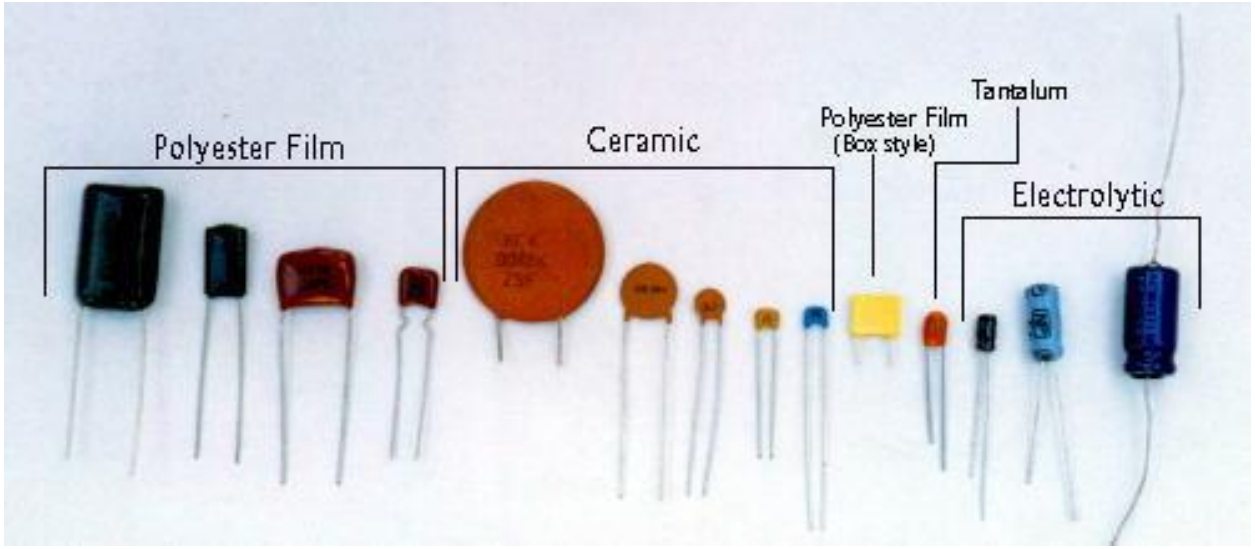
Diodi e LED

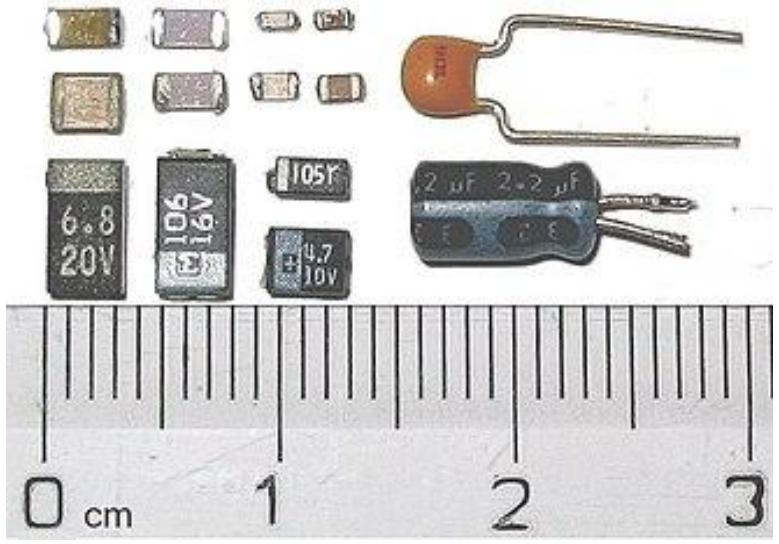

Generic Diode
Zener Diode
Schottky Diode
LED
Tunnel Diode
Photo Diode

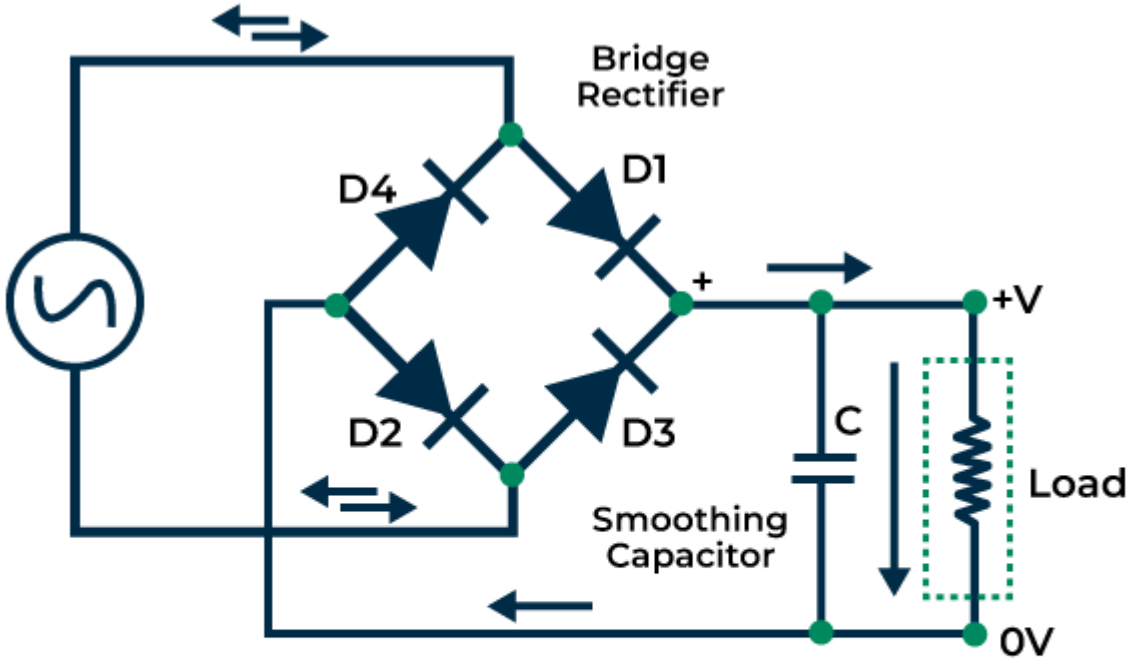

D4
D1
D2
D3
+V
I
DC
Load
0V

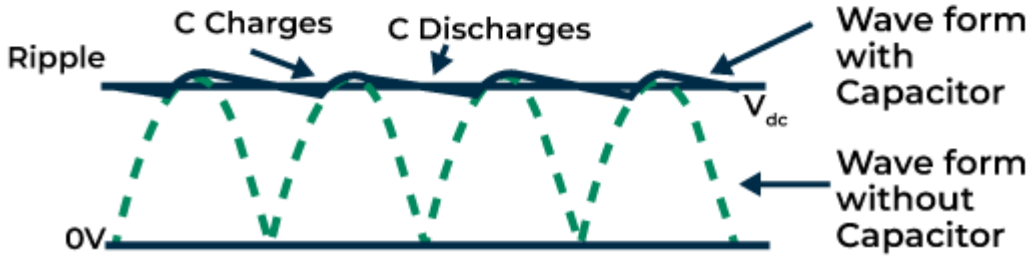

Diodo a giunzione pn / germanic
Diodo LED
Diodo Tunnel
Diodo varicap
Diodo Zener
Fotodiodo
Diodo Schottky

Condensatore

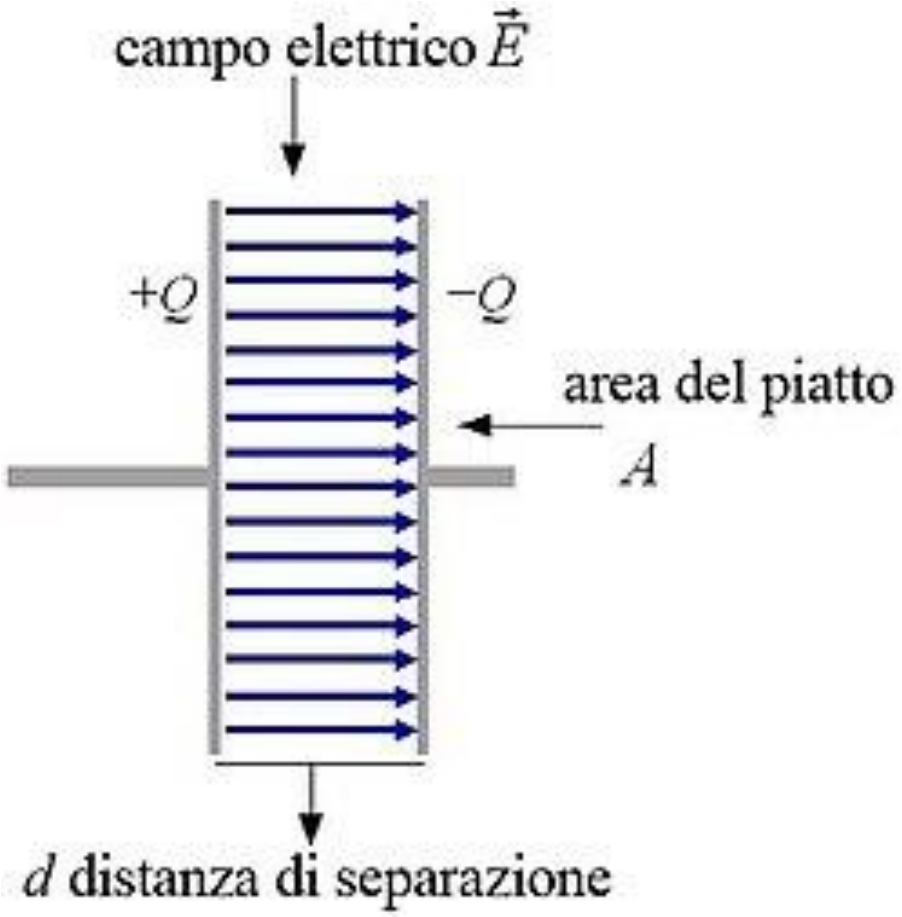


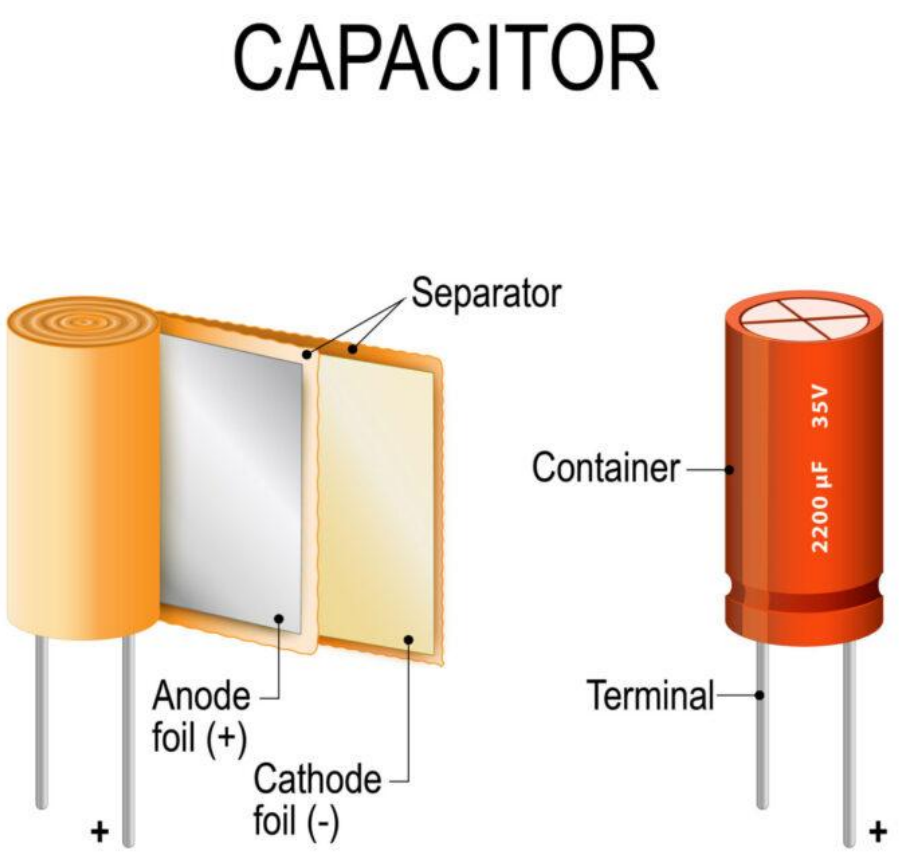




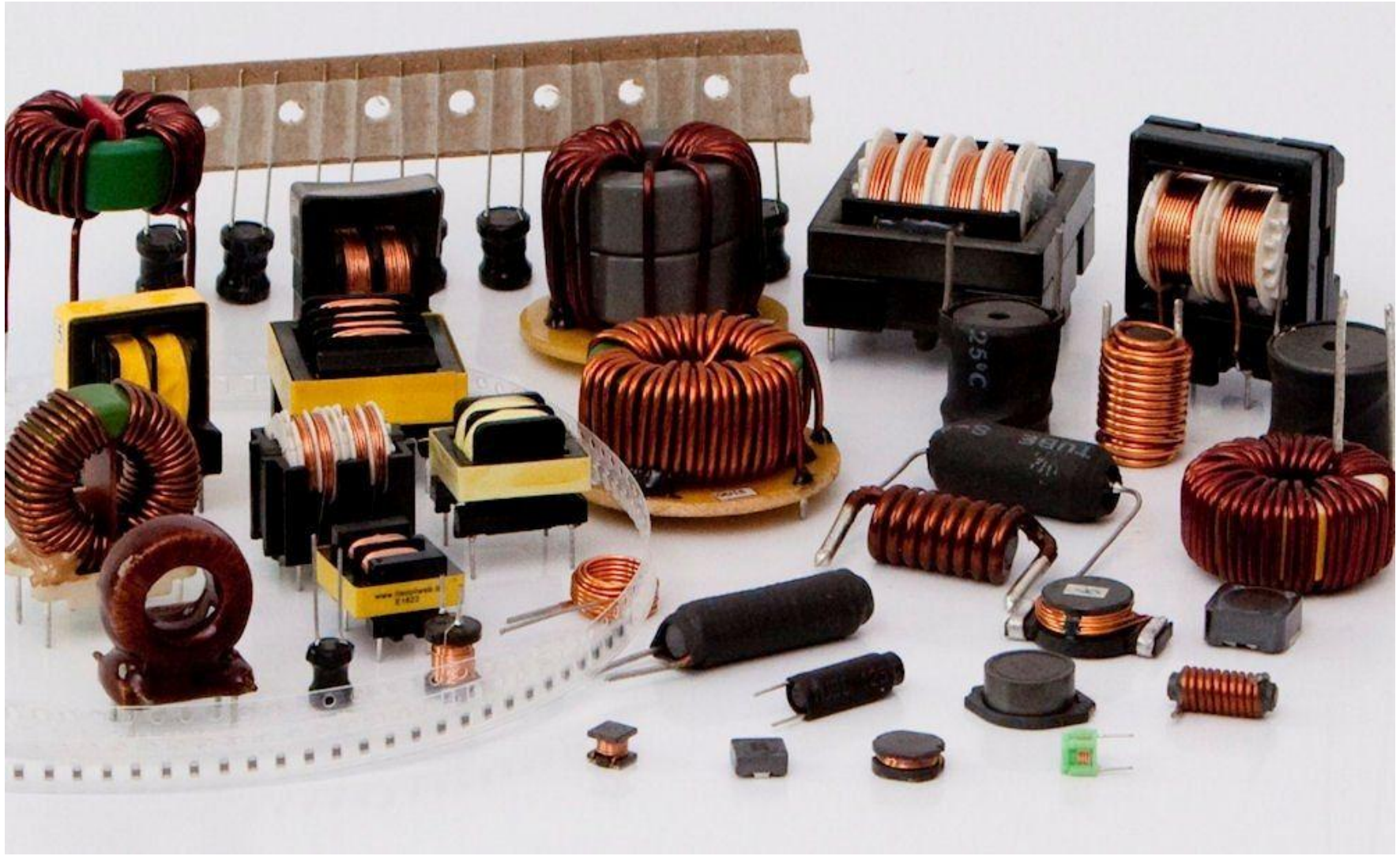


Condensatore – com'è fatto

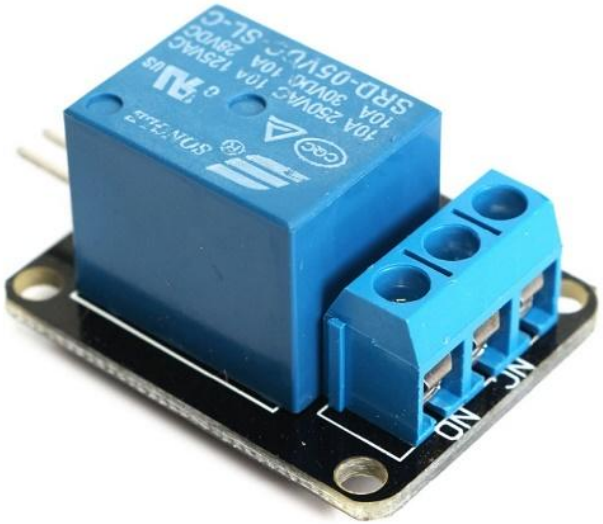


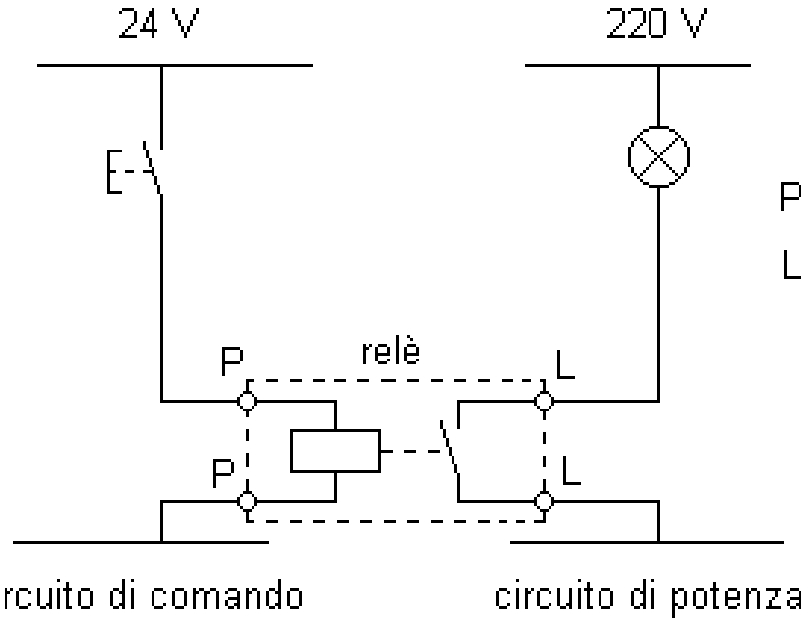


Induttore



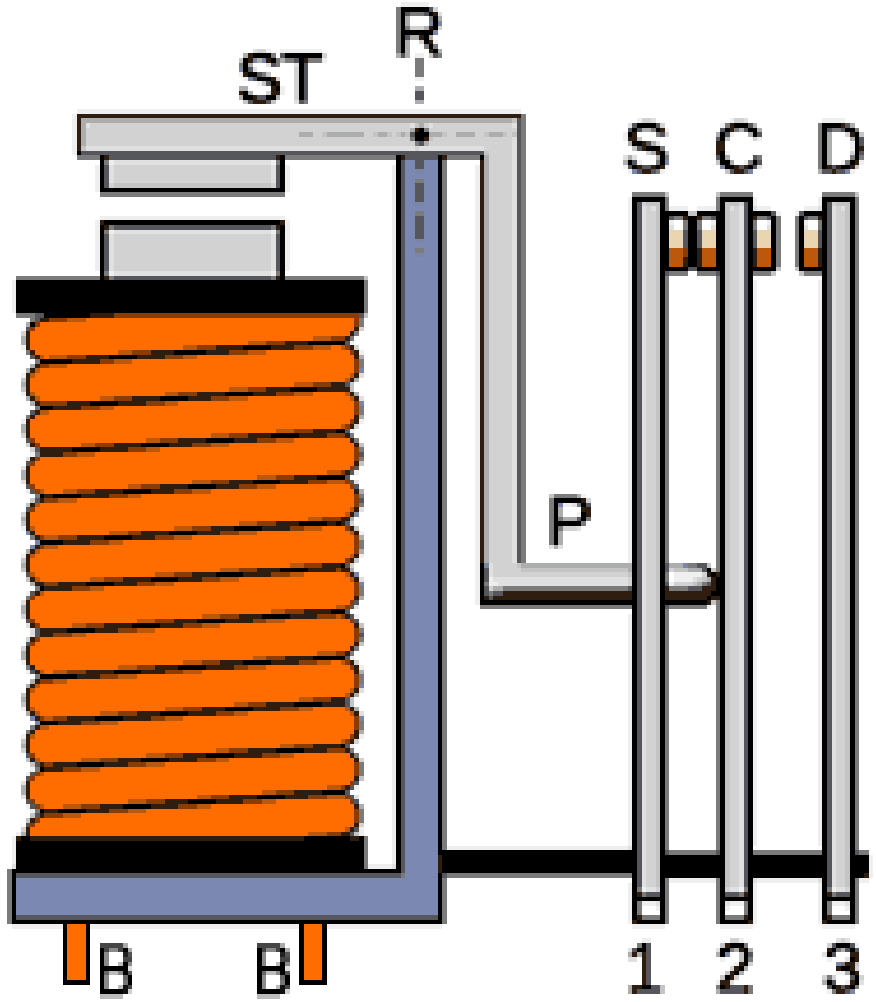
Relè



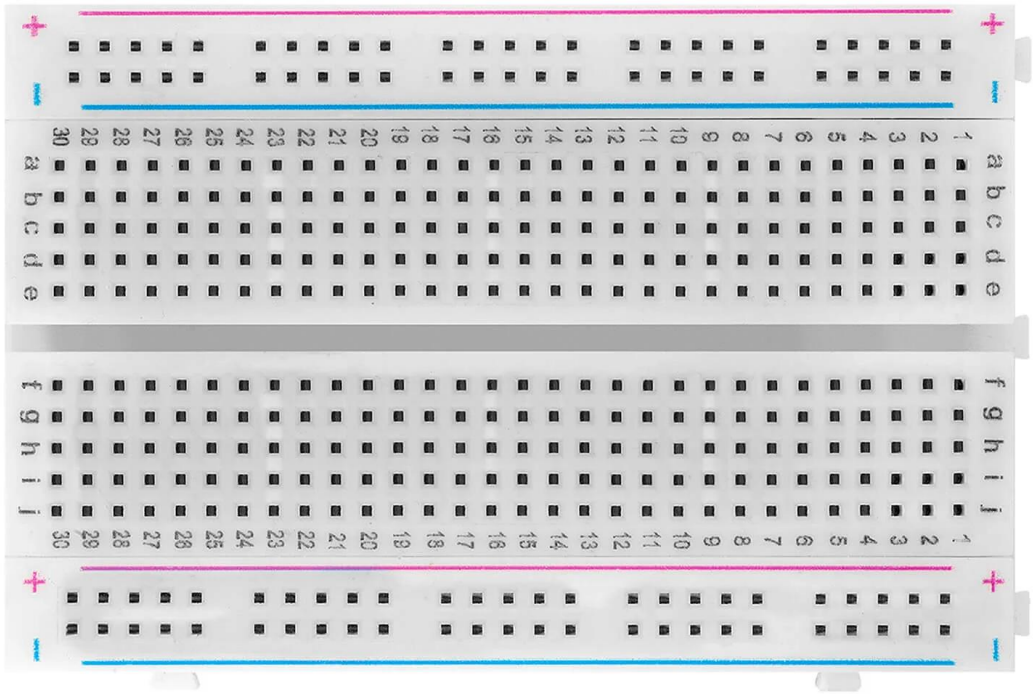


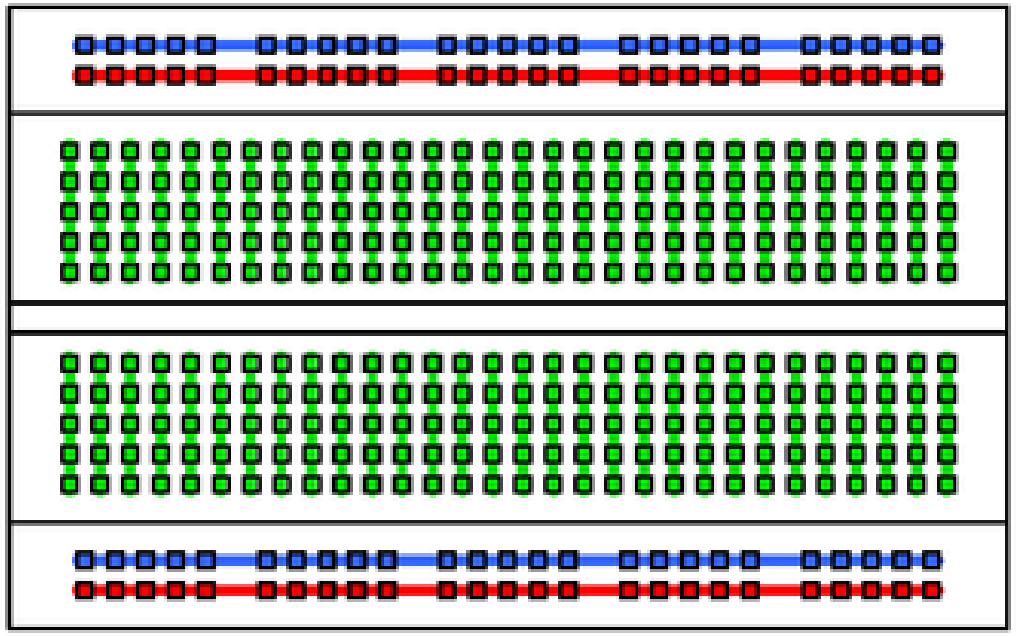
P = morsetti pulsanti

L = morsetti lampade



Breadboard – Basetta sperimentale





Piattaforme di Prototipazione

Arduino è una piattaforma basata su microcontrollore, ideale per progetti che richiedono il controllo di sensori e motori, mentre Raspberry Pi è un mini-computer più potente, adatto a progetti complessi che necessitano di un sistema operativo e connessioni di rete.
La scelta tra le due dipende dall'applicazione: Arduino per il controllo in tempo reale e Raspberry Pi per compiti che richiedono maggiore potenza di calcolo, come un server

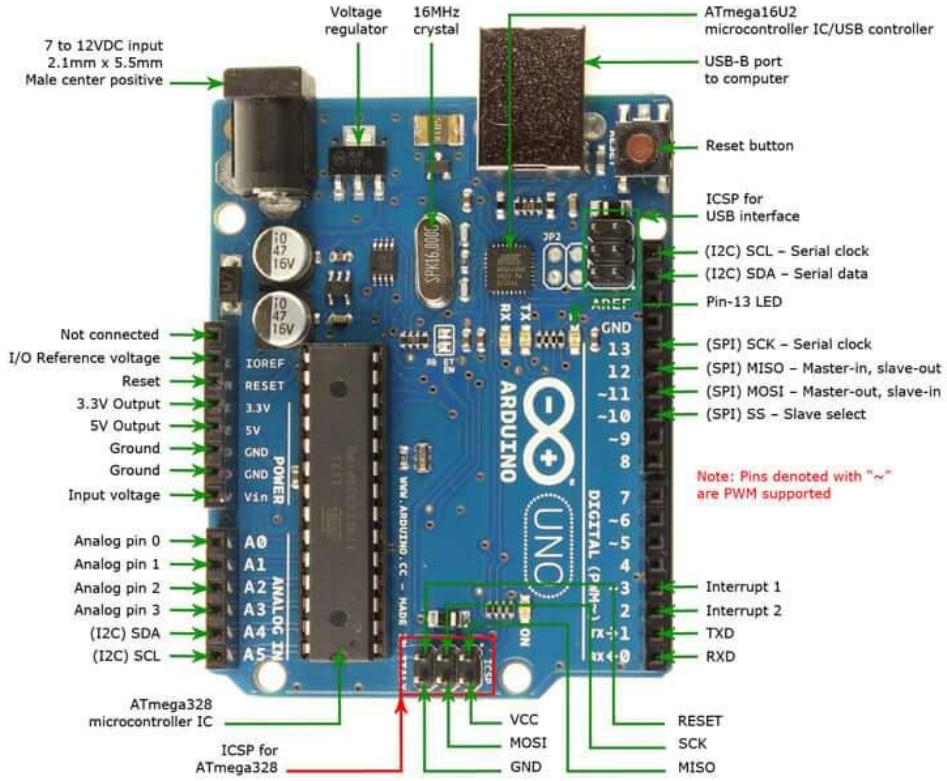
Architettura di un Microcontrollore (MCU) - Arduino

Il cuore di Arduino UNO è il microcontrollore ATmega328P, un sistema completo su un singolo chip.

- CPU: Esegue le istruzioni dello sketch.
- Memoria Flash: Memoria non volatile dove risiede il programma.
- SRAM: Memoria volatile per le variabili usate a runtime.
- EEPROM: Memoria non volatile per dati da conservare allo spegnimento.

- Periferiche Integrate:
  - Convertitore A/D (ADC): per leggere segnali analogici.
  - Timers/Counters: Per generare segnali PWM e gestire interrupt temporizzati.
  - Interfacce di Comunicazione: UART (seriale), SPI, I²C.

- Non ha un sistema operativo, ma esegue un singolo programma (sketch) alla volta.

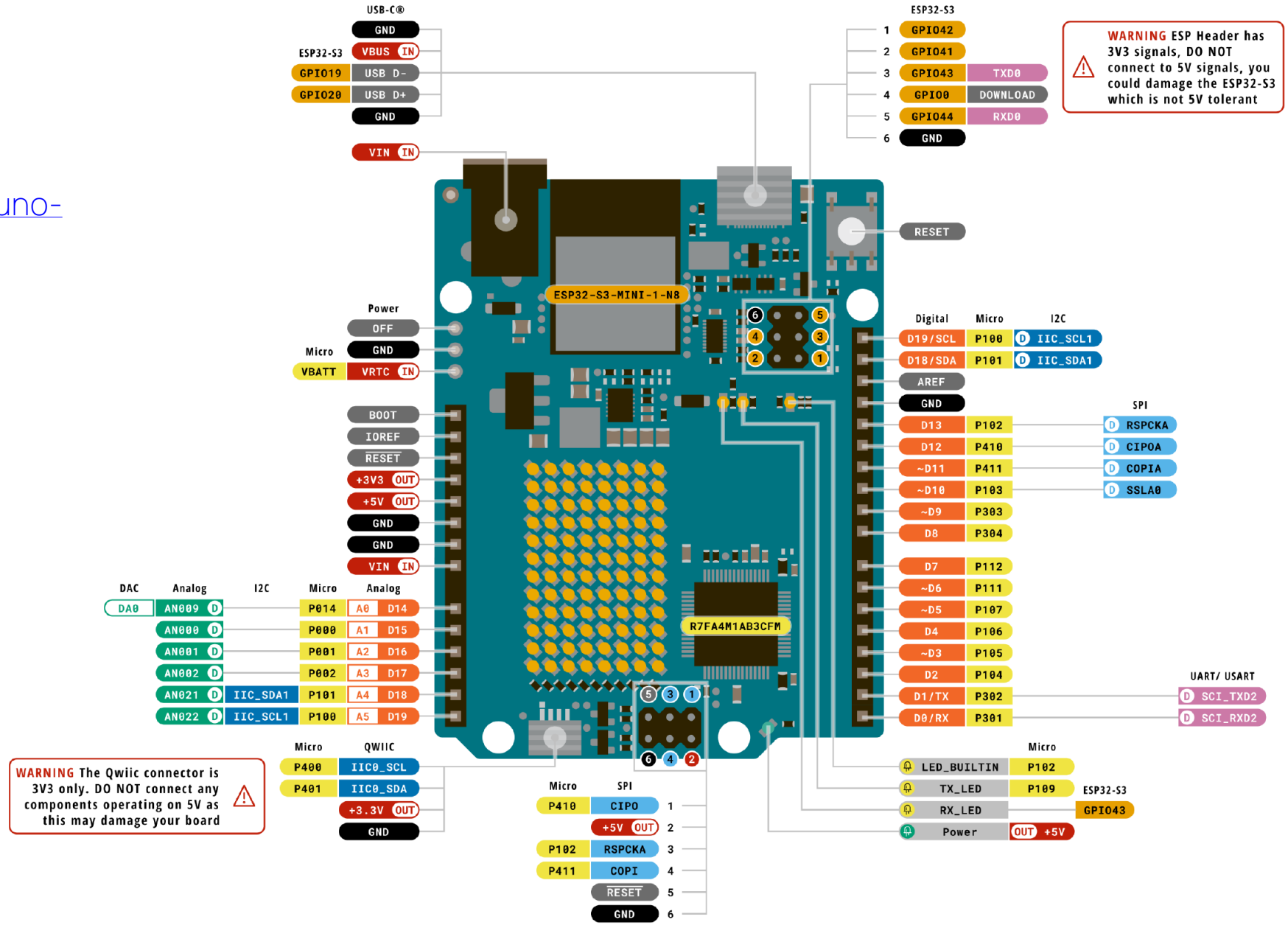


Architettura di un Single-Board Computer (SBC) - Raspberry Pi

- Raspberry Pi è un mini-computer completo basato su un System-on-Chip (SoC).
- SoC (es. Broadcom BCM2711): Integra più componenti in un unico chip:
  - CPU (Multi-core ARM): Processore ad alte prestazioni per eseguire un sistema operativo.
  - GPU (VideoCore): Unità di elaborazione grafica.
  - Controller I/O: Gestisce USB, Ethernet, GPIO.
- RAM: Memoria di sistema esterna al SoC
- Sistema Operativo: Esegue un OS completo (es. Raspberry Pi OS, una derivata di Debian Linux), che gestisce i processi, la memoria e le periferiche.

Arduino UNO R4 WiFi

Cheat-sheet:
https://docs.arduino.cc/tutorials/uno-r4-wifi/cheat-sheet/



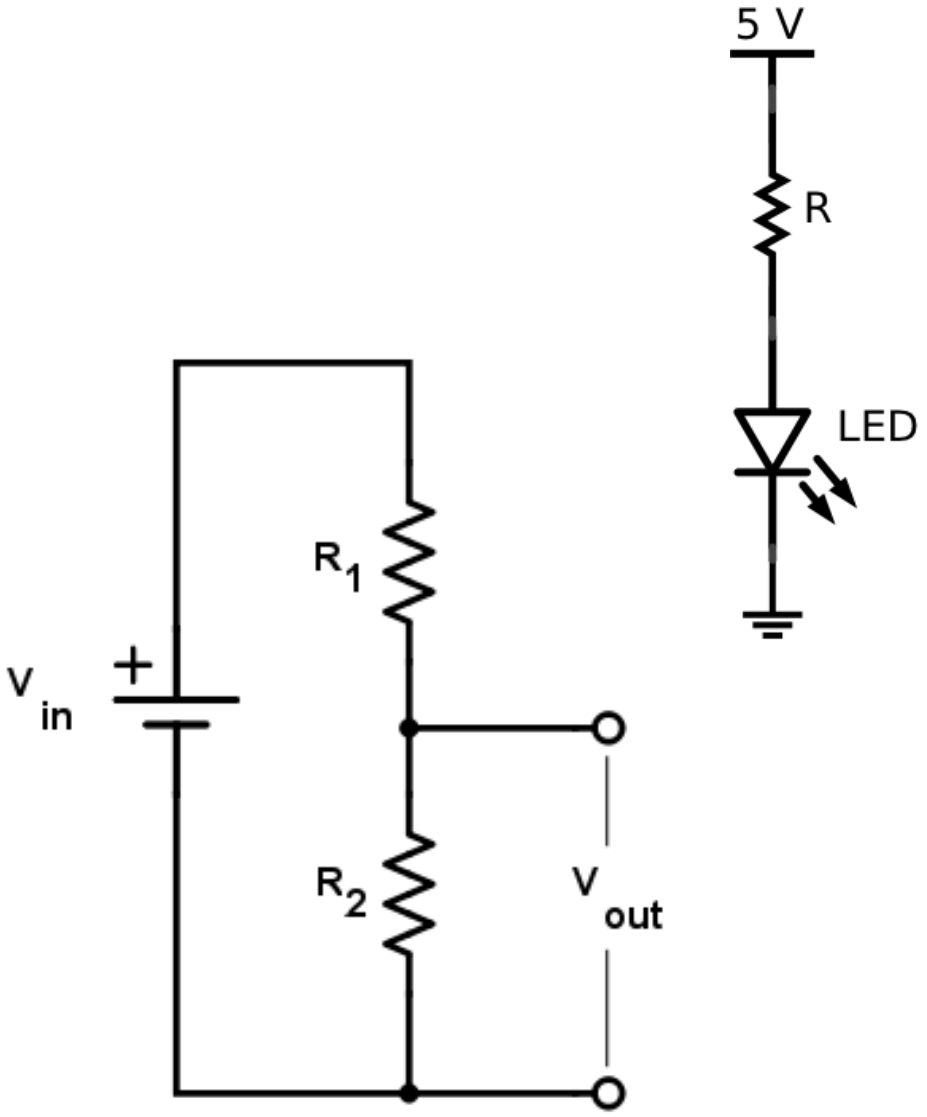
ARDUINO

I pin GPIO sono l'interfaccia fisica tra il controller e il mondo esterno.
Modalità del Pin:

- INPUT: Alta impedenza, legge lo stato logico esterno.
- OUTPUT: Bassa impedenza, impone uno stato logico (HIGH/LOW).
- INPUT_PULLUP: Attiva una resistenza interna verso VCC, utile per leggere pulsanti senza resistore esterno.

Caratteristiche Elettriche:

- Livelli Logici: Le tensioni che definiscono HIGH e LOW (es. TTL, CMOS). Su Arduino 5V, su Raspberry Pi 3.3V. Attenzione: un segnale a 5V può danneggiare un pin di Raspberry Pi!
- Corrente Massima (Source/Sink): Ogni pin può erogare/assorbire una corrente limitata (es. ~20-40 mA per ATmega328P). Superarla causa danni permanenti.



Segnali Digitali vs. Analogici

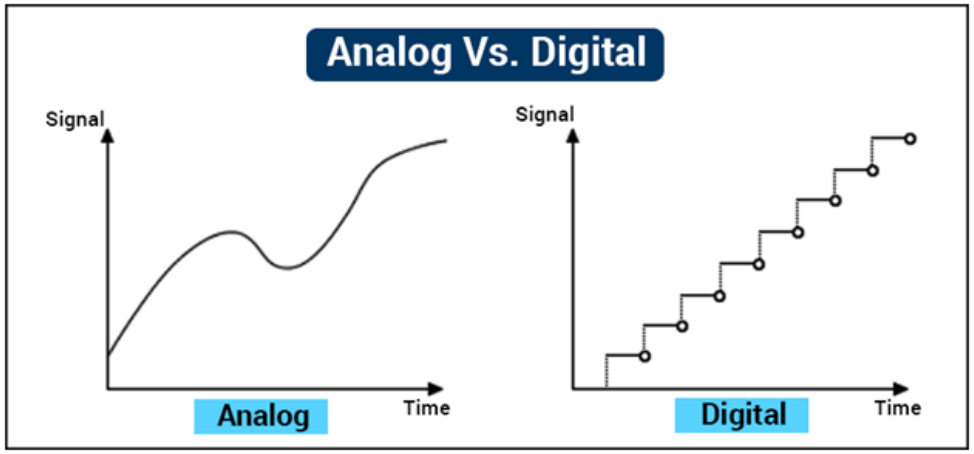
Segnale Digitale:

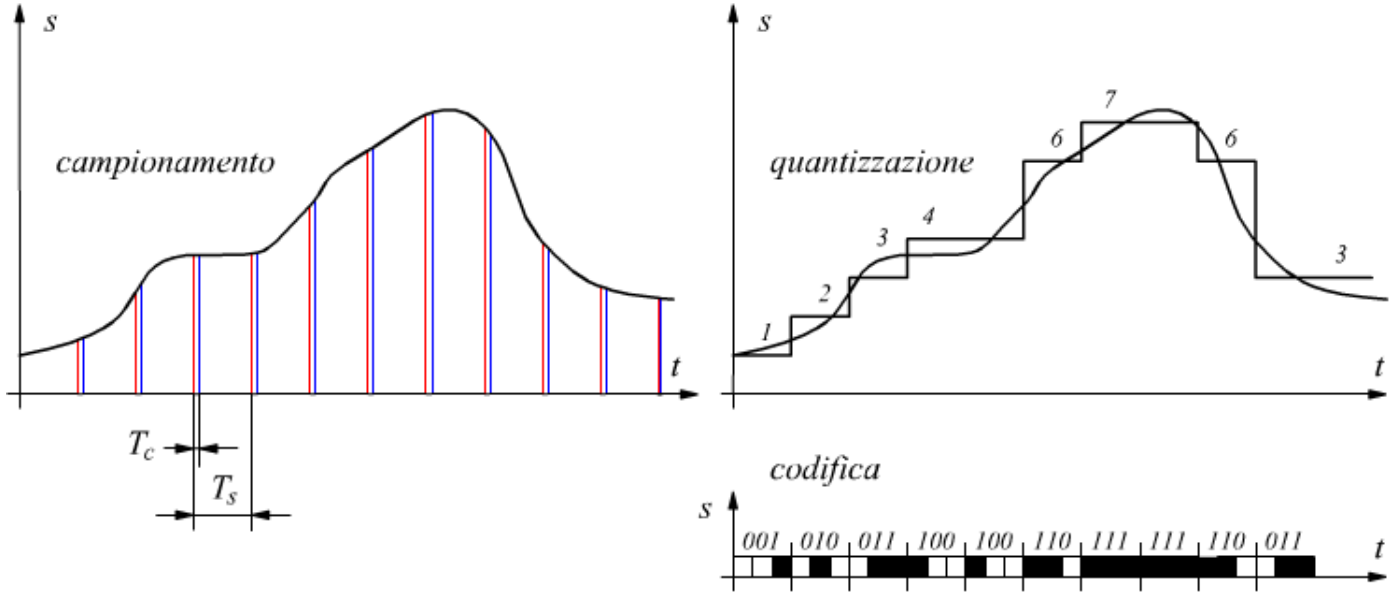
- Discreto nel valore e nel tempo.
- Rappresentato da due livelli di tensione: HIGH (es. > 3V) e LOW (es. < 0.8V).
- Utilizzato per comunicare stati binari (ON/OFF, 1/0).

Segnale Analogico:

- Continuo nel valore e nel tempo.
- La tensione può assumere qualsiasi valore in un range definito (es. da 0V a 5V).
- Necessario per misurare grandezze fisiche continue (temperatura, luminosità).

- Vedi convertitori A/D e D/A





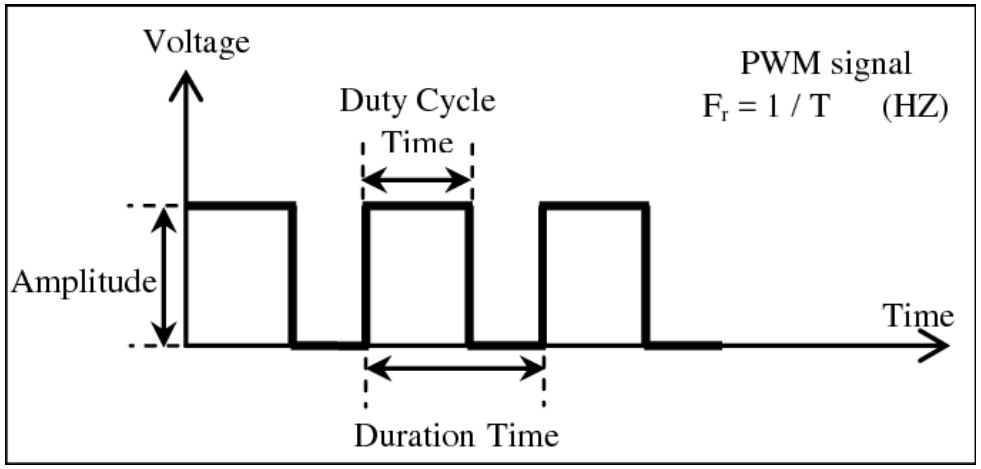
Da Analogico a Digitale e Viceversa

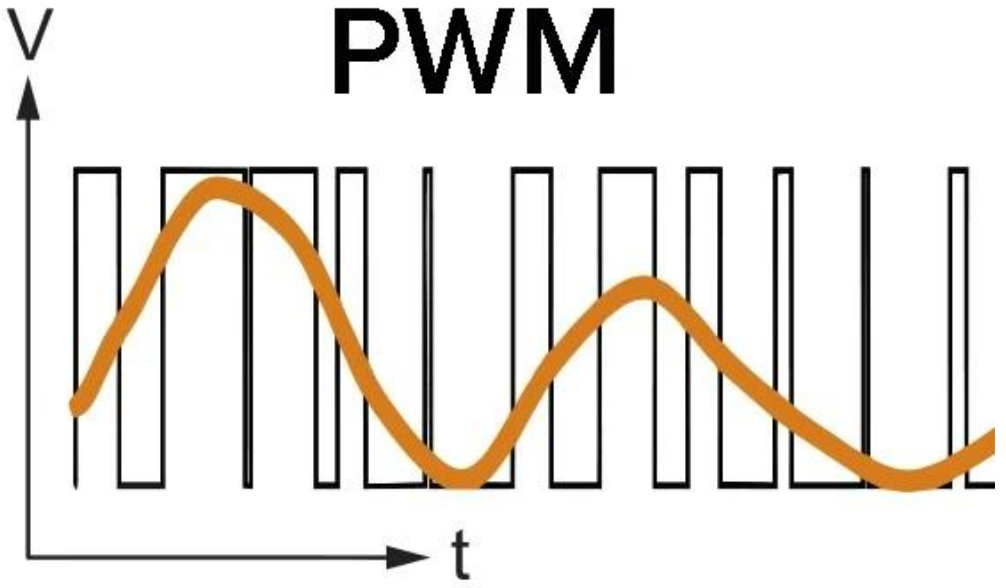
ADC (Analog-to-Digital Converter):

- Converte una tensione analogica in un valore numerico.
- Risoluzione: Il numero di bit usati per la conversione. L'ADC di Arduino ha (ad esempio) 10-bit, quindi può rappresentare 210=1024 livelli discreti (da 0 a 1023).

PWM (Pulse Width Modulation):

- Una tecnica per simulare un'uscita analogica usando un pin digitale.
- Si basa sulla variazione del Duty Cycle di un'onda quadra a frequenza fissa.
- Duty Cycle: La percentuale di tempo in cui il segnale rimane HIGH all'interno di un periodo. Un duty cycle del 100% equivale a VCC, del 50% a VCC/2, dello 0% a GND





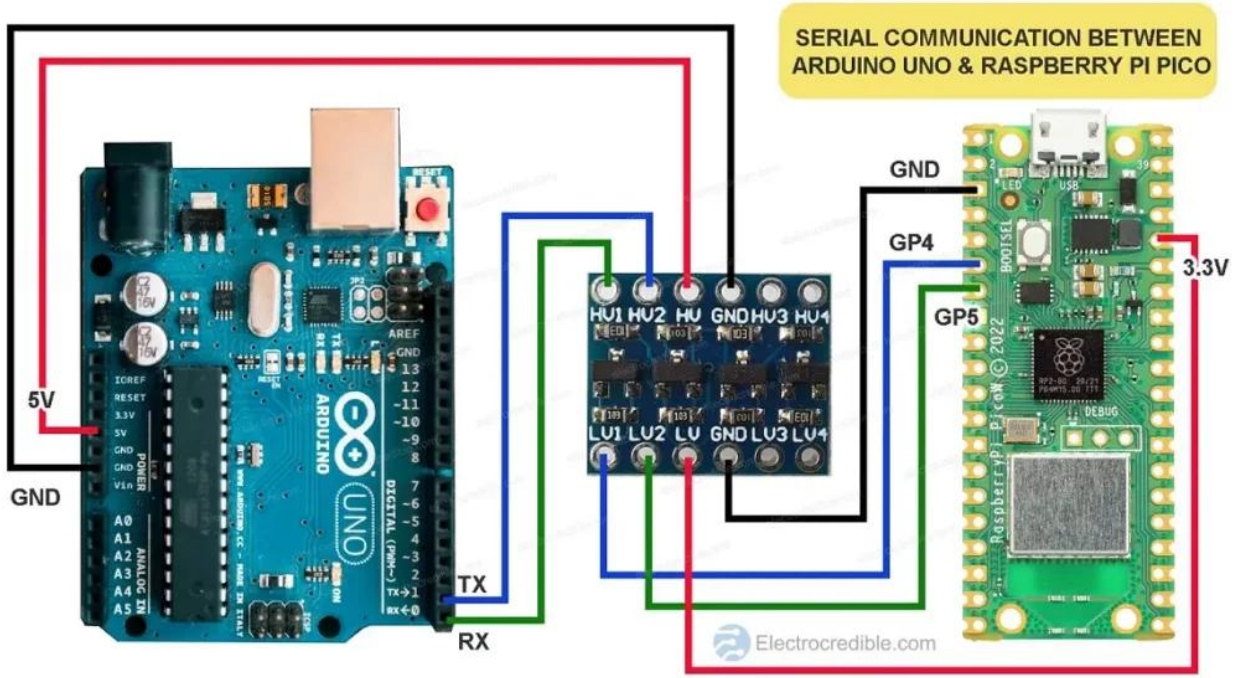
Codifica dell'informazione

I microcontrollori non capiscono "acceso" o "spento", capiscono solo livelli di tensione.
Arduino UNO (logica a 5V):

- LOW: Una tensione vicina a 0V (<2.5 V).
- HIGH: Una tensione vicina a 5V (>2.5 V).

Raspberry Pi (logica a 3.3V):

- LOW: Una tensione vicina a 0V.
- HIGH: Una tensione vicina a 3.3V.



Attenzione: Questa differenza non è un dettaglio, è una regola critica. Se un pin di Arduino in modalità OUTPUT (che eroga 5V) viene collegato a un pin GPIO di un Raspberry Pi, lo danneggerà permanentemente.
I pin del Raspberry Pi non sono "tolleranti" ai 5V. Per far comunicare questi due dispositivi, servono dei circuiti appositi chiamati level shifter (convertitori di livello logico).

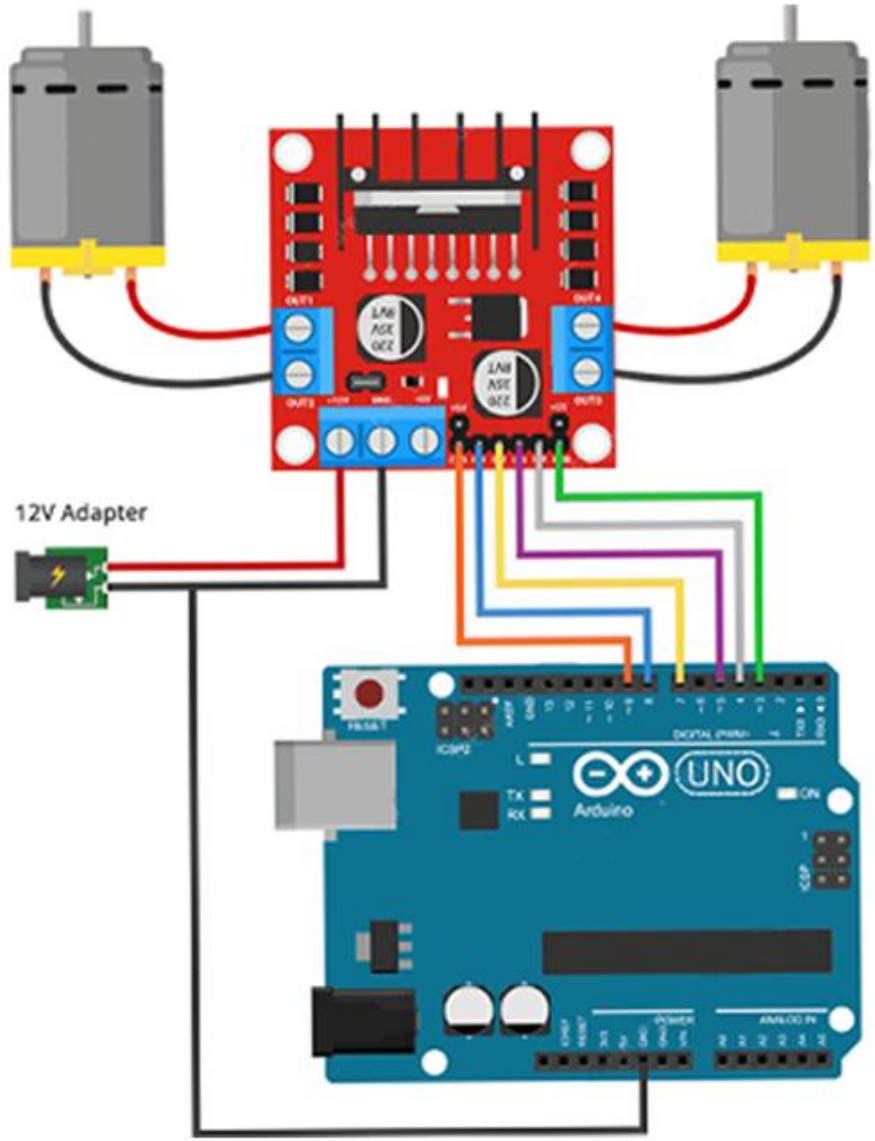
Interfacce di Input e Output

Sensori (Input):

- Analogici: Forniscono una tensione continua (es. fotoresistenza, sensore di temperatura LM35). Si leggono con l'ADC.
- Digitali: Forniscono uno stato HIGH/LOW (es. pulsante, sensore PIR).
- Basati su Protocollo: Comunicano tramite protocolli seriali standard (es. sensore di umidità DHT11, sensori I²C/SPI).

Attuatori (Output):

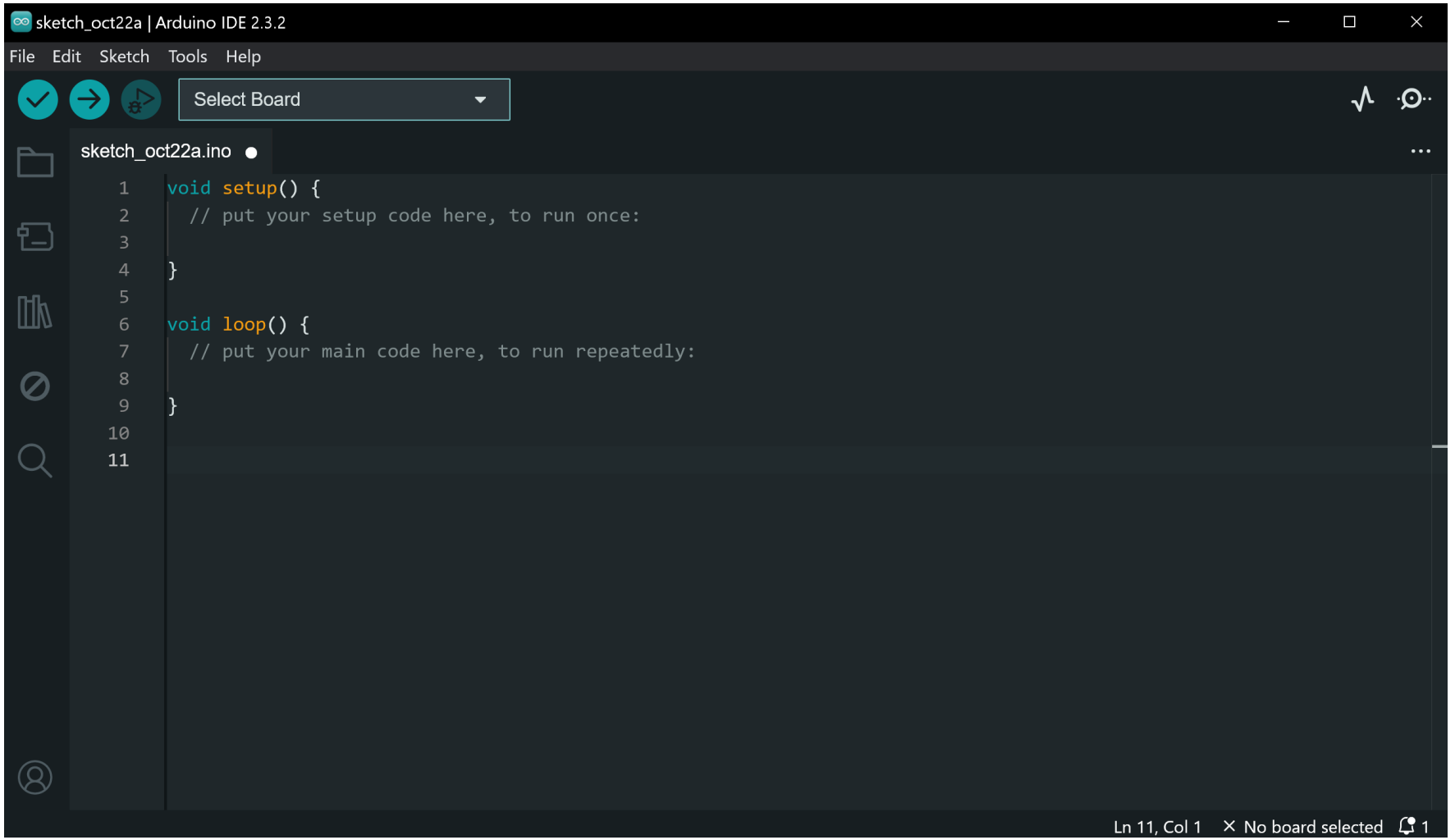
- Semplici: Controllati direttamente da un pin GPIO (con resistenza di limitazione) (es. LED).
- Complessi: Richiedono circuiti di pilotaggio esterni (driver) per gestire correnti elevate (es. Motori DC, servomotori, relay).



Codice Arduino

1. Installare Arduino IDE da: https://www.arduino.cc/en/software/
2. Aprire Arduino IDE
3. Selezionare la Board

Se non avete ancora Arduino potente sempre “smanettare” su Tinkercad: https://www.tinkercad.com/



Review generale del codice C++

- Functions:

```
<return type> <function name>(<input args>)}
        // code to execute
}
```

- Arrays

```
int primes[5] = {2,3,5,7,11}
int number = primes[0];
```

- Arduino ha la Comunicazione Seriale:
  - Porta seriale
  - Serial.begin(9600); // baudrate determina la velocità di trasmissione
  - Serial.print() oppure Serial.println()

I/O pins

- Digital Pins
  - Leggere tensioni LOW (<2.5V) e HIGH (>2.5V or 5V)
  - Per leggere usare digitalRead(<pin number>)
- PWM Pins
- Analog Pins

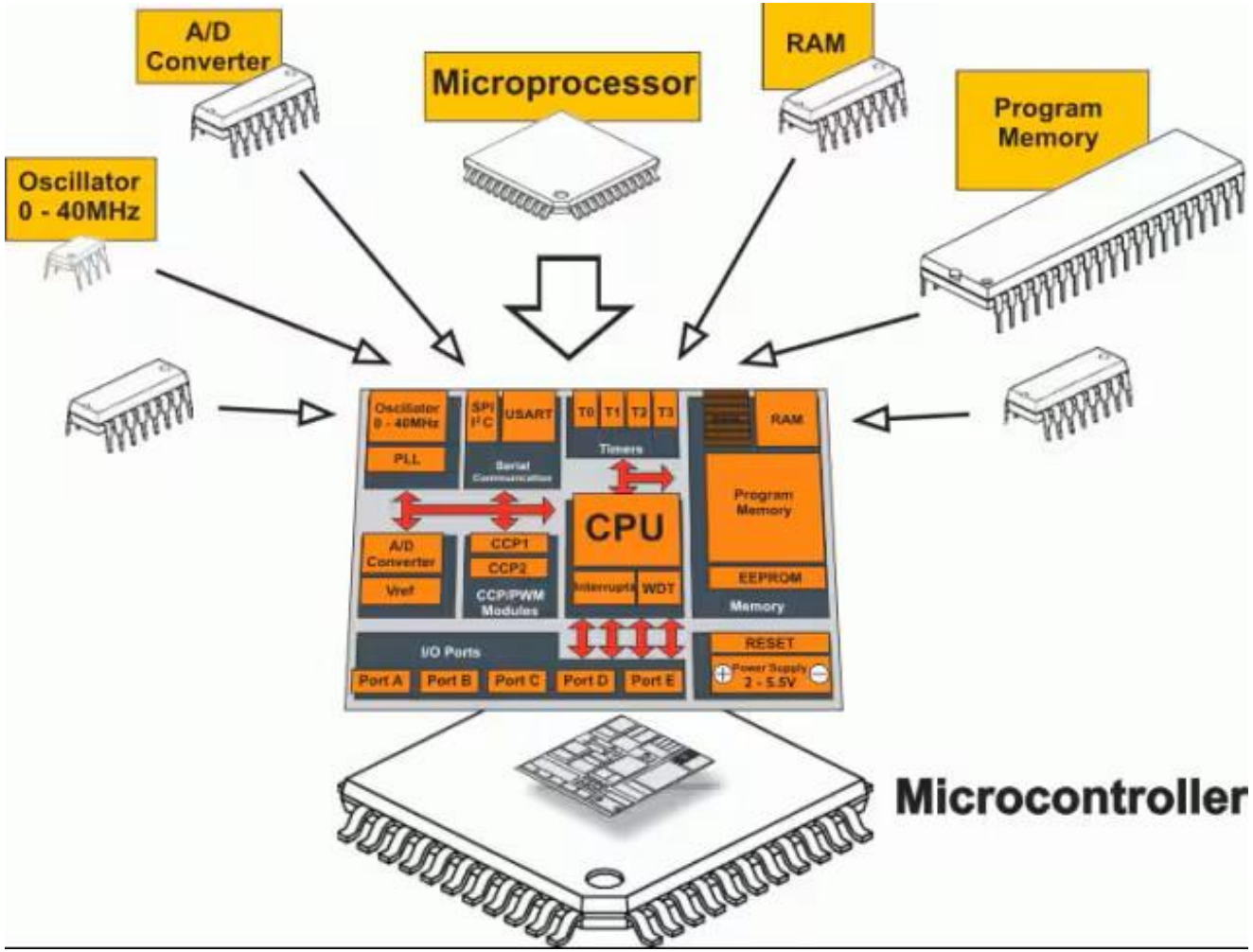
```
#define BUTTON_PIN 2

int variabileGlobale = 1;

void setup(){
    pinMode(BUTTON_PIN,INPUT);
    pinMode(8,OUTPUT);
}

void loop(){
    int buttonState = digitalRead(BUTTON_PIN);
    Serial.println(buttonState);
}
```

Domande, Dubbi, Perplessità?


A/D Converter
Microprocessor
RAM
Program Memory
Oscillator 0 - 40MHz
Oscillator 0 - 40MHz
PLL
SPI I²C
USART
Serial Communication
T0 T1 T2 T3
Timers
RAM
CPU
Program Memory
A/D Converter
Vref
CCP1
CCP2
CCP/PWM Modules
Interrupts
WDT
EEPROM Memory
I/O Ports
Port A
Port B
Port C
Port D
Port E
RESET
Power Supply 2 - 5.5V
Microcontroller