

Laboratorio di Fisica

Esperienza con Arduino: esponenziale e intensità luminosa
– Liceo Massimo D’Azeglio di Torino –

Descrizione Esperienza di laboratorio di fisica in cui lo studente usa Arduino per esplorare un esempio di andamento esponenziale applicato alla realtà: la luminosità al variare di strati di copertura semi-trasparente.

Scopi dell’esperimento Esplorare l’applicazione della funzione esponenziale ad un processo fisico; studiare l’andamento della luminosità al variare di strati di copertura; usare la scala logaritmica per linearizzare i dati.

Strumenti e materiale:

- scheda Arduino Uno e computer
- un resistore da 10 k Ω , un resistore sensibile alla luce (foto-resistenza)
- strati di copertura semi-trasparente (es. una busta porta fogli A4 tagliata a strisce)

Richiami teorici La fisica alla base della luminosità è troppo complessa ed esula dagli obiettivi di questo esperimento, ma sarà sufficiente un approccio intuitivo-matematico.

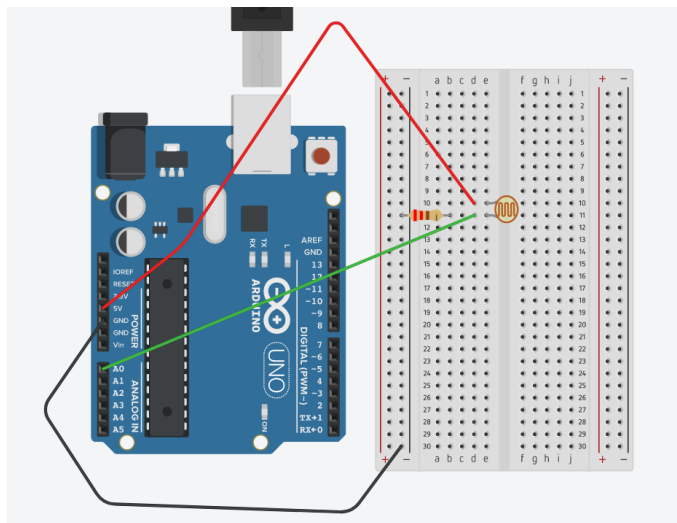
Il sensore che misura la luminosità viene successivamente coperto con la sovrapposizione di strati identici. All’aumentare del numero di strati, ci si aspetta che la luminosità diminuisca, senza tuttavia andare a zero (perché le coperture semi-trasparenti faranno sempre passare un po’ di luce) e, soprattutto, senza raggiungere valori negativi. Una qualsiasi funzione polinomiale decrescente (come una retta) non è quindi adatta allo scopo, ma serve una funzione matematica che sia sempre positiva e che ammetta un asintoto orizzontale di equazione $y = 0$:

$$V(n) = V_0 e^{-kn} \quad (1)$$

Dove $V(n)$ è il valore di differenza di potenziale misurato che è direttamente proporzionale alla luminosità, V_0 è il valore di differenza di potenziale massimo (luminosità massima, senza coperture), n è la variabile espressa come numero di strati di copertura che, se sono tutti uguali, è direttamente proporzionale allo spessore, k è una costante in questo caso adimensionale.

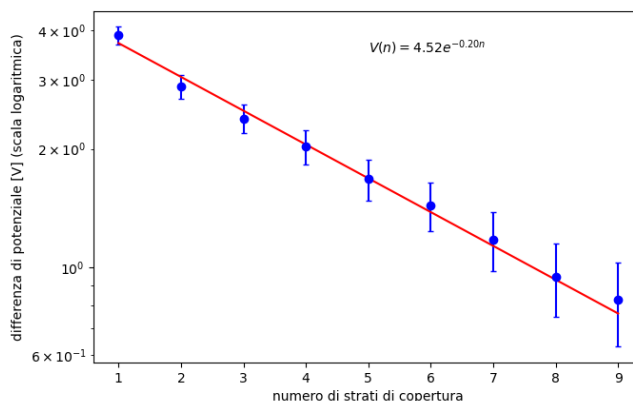
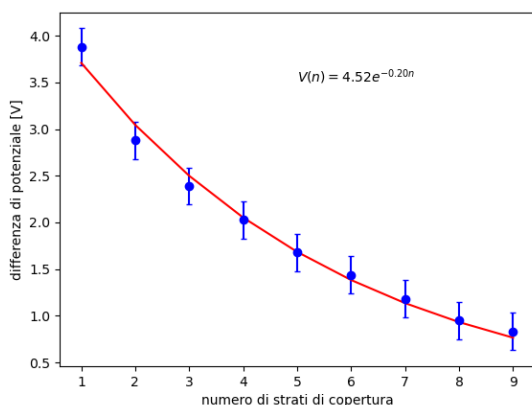
Svolgimento: presa dati La scheda Arduino fornisce un valore iniziale di differenza di potenziale di circa 5 V e permette la lettura della differenza di potenziale $V(n)$ ai capi della fotoresistenza e al variare della copertura.

1. Implementare un circuito analogo a quello mostrato nella seguente figura.
2. Visualizzare i valori di $V(n)$ sul monitor seriale e scriverli sia su carta che su file, calcolando il valor medio di almeno dieci misure.
3. Aggiungere uno strato sopra la fotoresistenza e ripetere le misure, facendo attenzione a non variare la luminosità ambientale (es. non fare ombra con il proprio corpo).
4. Ripetere la procedura per almeno 8 strati e compilare una tabella come quella seguente (tutte le misure sono in volt, si è ommesso il simbolo per maggiore leggibilità):



| n | V_1 | V_2 | V_3 | ... | V_{10} | \bar{V} | ΔV |
|-----|-------|-------|-------|-----|----------|-----------|------------|
| 0 | 4,91 | 4,92 | 4,89 | ... | 4,90 | 4,91 | 0,05 |
| 1 | 4,21 | 4,22 | 4,19 | ... | 4,20 | 4,91 | 0,04 |
| 2 | 3,71 | 3,72 | 3,69 | ... | 3,60 | 3,83 | 0,03 |
| ... | ... | ... | ... | ... | ... | ... | ... |

Svolgimento: analisi dati Rappresentare su un diagramma differenza di potenziale-numero strati i dati trovati usando il foglio di calcolo o Python come nella seguente figura di sinistra.



Trovare la funzione matematica esponenziale che rappresenta al meglio i dati, da essa ricavare la costante di tempo k . Rappresentare gli stessi dati su scala logaritmica, o usando le funzioni del software o calcolando il logaritmo dei rapporti $V(n)/V_0$ e mettendo questi valori come ordinate.

Trarre delle conclusioni sul confronto dei valori, sui valori di incertezza, sui possibili errori sistematici, casuali o accidentali commessi. Confrontare i risultati ottenuti con quelli degli altri gruppi.

Codice Si riporta un esempio di codice. Si è fatto uso di una funzione `voltage` che misura il valore di differenza di potenziale dal pin con conversione da bit a volt.

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

void loop() {
  float V = analogRead(A0) * 5.0 / 1024.0;
  Serial.println(V);
  delay(10);
}
```