

# ANALISI DELLA CARATTERISTICA V-I DEL LED

## Obiettivo

L'esperimento si consiste nel costruire un piccolo circuito e acquisire i dati relativi all'andamento della corrente che circola sul LED in funzione della tensione applicata ai suoi capi. L'esperimento ha come scopo la determinazione della lunghezza d'onda emessa dal LED (misura indiretta) e la verifica del risultato ottenuto tramite lo spettrometro (misura diretta)

Nell'esperimento utilizzeremo la scheda Arduino programmandola per alimentare il circuito, simulando una tensione crescente da 0 a 5V, sia per acquisire i valori di tensione, che per calcolare la corrente che circola sul LED indirettamente.

## Obiettivi dell'esperienza:

### Prima parte

1. ottenere da un circuito la curva caratteristica dell'intensità di corrente in funzione della tensione per un LED rosso e per un LED blu.
2. determinare la tensione di soglia della curva V-I e calcolare la lunghezza d'onda ( $\lambda$ ) della luce.

### Seconda parte

3. misurare la lunghezza d'onda emessa dal LED tramite uno spettrometro.
4. confrontare i risultati 2 e 3.

## Prima parte

### Esecuzione dell'esperimento

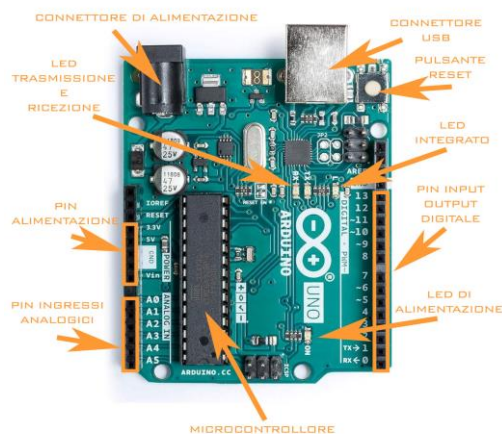
La prima cosa da fare è preparare il circuito collegato alla scheda Arduino.

### Materiali:

Microcontrollore Arduino e una bread-board;

Pc e cavo usb;

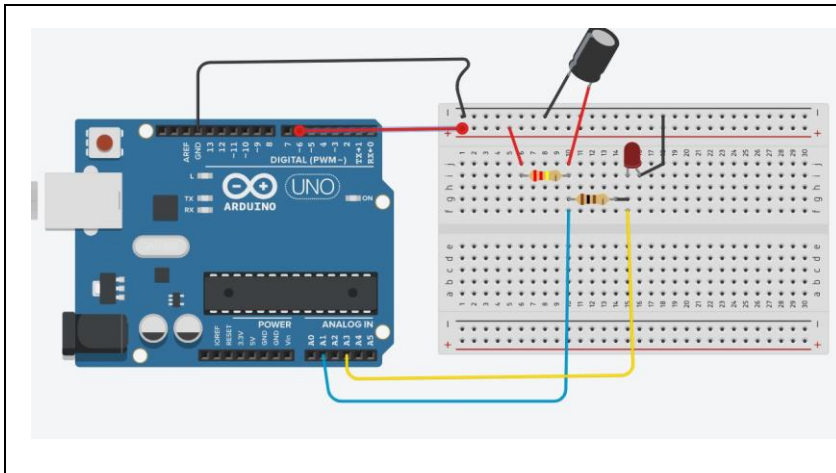
Componenti per il circuito da montare sulla bread-board:



|                |                  |               |                            |
|----------------|------------------|---------------|----------------------------|
|                |                  |               |                            |
| $R=220 \Omega$ | $R_D =22 \Omega$ | $C=470 \mu F$ | Led / diodo<br>ROSSO e BLU |

## 1) costruire il circuito e collegare la scheda Arduino

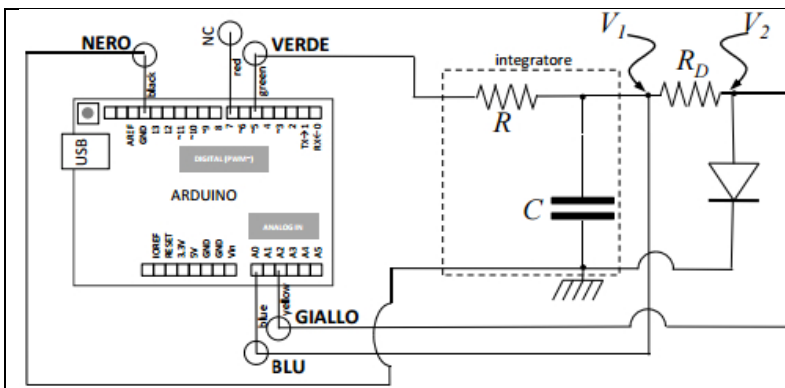
Utilizzando la breadboard posiziona la resistenza di  $R=220\ \Omega$  e il condensatore  $C$ , e una resistenza di polarizzazione del led  $R_D=22\ \Omega$  come indicato in figura



- *il PIN 5 alla linea predisposta nella breadboard (+) per alimentare il circuito;*
- *le porte A1 e A3 ai capi della resistenza di polarizzazione  $R_D$ , per leggere i valori della tensione  $V_0$  e  $V_1$*
- *la linea indicata nella breadboard (-) al pin GND;*

Fate attenzione a realizzare correttamente i collegamenti con la scheda e a controllare preliminarmente che le connessioni con i vari piedini della scheda siano corretti.

Una volta effettuati i collegamenti abbiamo realizzato il circuito in schema e possiamo testarlo con lo sketch di prova che accende il LED



// SKETCH DI PROVA DEL CIRCUITO

```
int RampPin = 5; //pin 5 come uscita
void setup() {
  //pin LED_BUILTIN as an output.
  pinMode(RampPin, OUTPUT);
}
void loop() {
  //accende il LED
  digitalWrite(RampPin, HIGH);
  delay(1000);
}
```

## 2) Il codice per programmare la scheda Arduino

Analizziamo in breve il programma che gestisce il microcontrollore Arduino e il segnale PWM

### Il Pulse Width Modulation (PWM)

PWM è l'acronimo di Pulse Width Modulation ed è una tecnica utilizzata in genere per ottenere un'uscita analogica con mezzi digitali, ad esempio per controllare la luminosità dei LED, o la velocità del motore DC, o un servomotore etc..etc. Tutti i pin digitali possono generare un segnale in tensione: quando sono attivati HIGH danno un segnale in tensione di 5V e quando sono attivati 0V. L'uscita è un segnale ad onda quadra.

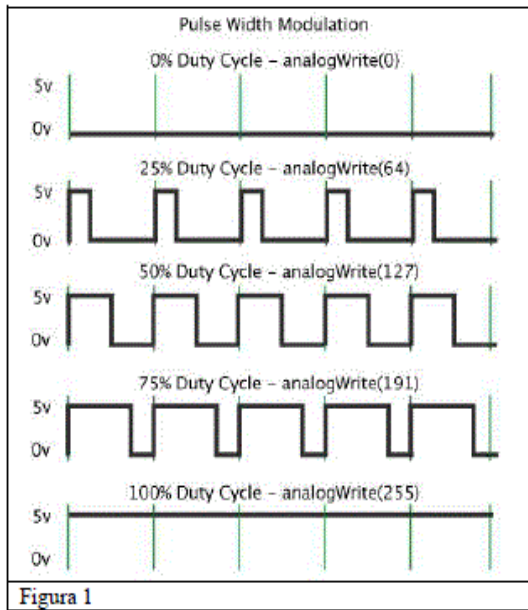


Figura 1

Consideriamo la frequenza del segnale di 1000 Hz, per cui ogni periodo dura circa 1 millisecondo.

All'interno del periodo abbiamo un impulso a 5 V che ha una durata temporale variabile (larghezza dell'impulso, pulse width), ed una parte in cui l'impulso si azzerava.

Un duty cycle del

- 100% indica una tensione continua a 5 V,
- 0% indica una tensione fissa a 0 V
- 50% indica che per metà del periodo abbiamo una tensione a 5V e per l'altra metà a 0 V (in questo caso la tensione effettiva in uscita è 2.5 V).

255 corrisponde al 100%

127 corrisponde al 50%,

Se si vuole alimentare un LED, con una tensione "variabile" è possibile modulare, su porte abilitate, il tempo di accensione e spegnimento del segnale nel ciclo di lavoro, il *duty cycle*, tramite la funzione integrata *analogWrite(pin ~, i%)*.

*analogWrite(pin, 0)* indica un segnale del 0% del ciclo di funzionamento

*analogWrite(pin, 127)* indica un segnale del 50% del ciclo di funzionamento.

*analogWrite(pin, 255)* indica un segnale del 100% del ciclo di lavoro.

Sono abilitate all'uso della PWM le uscite digitali contrassegnate sulla scheda col simbolo ~  
3, 9, 10 e 11 con frequenza di 490Hz  
5 e 6 con frequenza di 980Hz

Cioè la scheda Arduino permette di scegliere il *duty cycle* con un valore compreso fra 0 e 255 dove il 255 corrisponde al 100% e 127 corrisponde al 50%, potendo perciò avere una precisione di circa lo 0.5%, sul valore della tensione.

### Sketch di ARDUINO

Il codice che viene eseguito da Arduino provveda a:

- 1) alimentare il circuito tramite il PIN5 attraverso una tecnica chiamata modulazione di larghezza di impulso - Pulse Width Modulation (PWM).
- 2) prelevare i valori delle tensioni V1 e V2 ai capi della resistenza di polarizzazione  $R_D$  tramite le porte A1 e A2 di Arduino
- 3) calcolare in modo indiretto l'intensità di corrente che attraversa il diodo, in base al valore della resistenza  $R_D$ ;
- 4) stampare sul Serial Monitor i dati della tensione e corrente applicata al LED;

Il codice:

```
const unsigned int RampPin = 6; // porta di uscita Segnale PWM
```

```

const unsigned int analogPin_uno = 0; // d.d.p. prima della Rd
const unsigned int analogPin_due = 1; // d.d.p. dopo la Rd
unsigned int i = 0;
float medio_1 = 0;
float medio_2 = 0;
int start = 1; //flag per fermare il ciclo
int rd = 22; //Resistenza Rd da modificare
void setup()
{
  pinMode(RampPin, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  delay(1000); // attende 1s
  Serial.println("tensione[V], corrente[mA]");
  for (i = 0; i < 256; i = i + 4) {
    analogWrite(RampPin, i);
    delay(50);
    //media di 5 letture
    for (int j = 0; j < 5; j++) {
      medio_2 = medio_2 + analogRead(analogPin_due);
      medio_1 = medio_1 + analogRead(analogPin_uno);
    }
    medio_2 = medio_2 / 5; //valor medio di 5 letture
    medio_1 = medio_1 / 5;

    // stampa Tensione media sul LED
    Serial.print(5.0 * medio_2 / 1023.0, 4);
    //Serial.print(5.0 * (analogRead(analogPin_due)) / 1023.0, 4);
    Serial.print(" ; ");

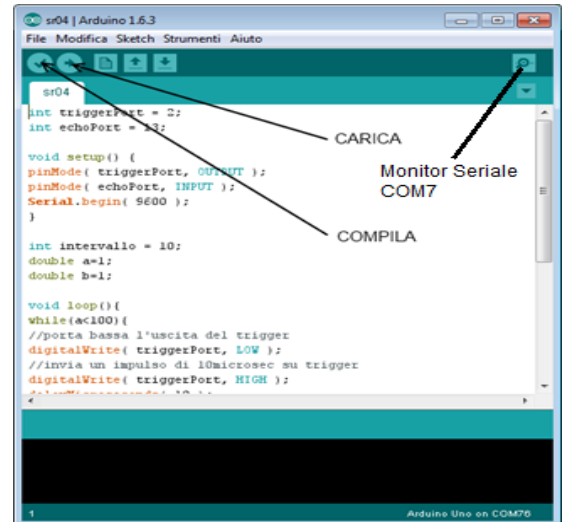
    // stampa Intensità di corrente media sul LED
    Serial.println((5.0 / 1.023) * ((medio_1 - medio_2)) / rd, 4);
    // Serial.println((5.0 / 1.023) * ((analogRead(analogPin_uno))
- (analogRead(analogPin_due))) / rd, 4);
    medio_1 = 0;
    medio_2 = 0;
  }
  delay(100); // l'intervallo di tempo campionamento
  analogWrite(RampPin, 0);
  while (start = 1) {};
}

```

I parametri che è possibile modificare all'interno del programma sono l'intervallo di tempo  $\Delta t$  fra due campionamenti successivi, da scegliere in funzione della frequenza di taglio dell'integratore e il valore del resistore  $R_D$ .

## Caricare gli sketch su Arduino e lanciare la misura

Per caricare gli sketch su Arduino si usa il software Arduino che si trova sul desktop. Una volta aperto bisogna copiare lo sketch precedente al suo interno e poi salvare il file su una cartella. Poi si può verificare la corretta scrittura del codice con il pulsante compila. Se la compilazione va a buon fine si può caricare il programma sulla scheda con il pulsante carica. Se il programma di caricamento invia un messaggio di errore occorre controllare dal Menù *Strumenti* la porta di comunicazione con Arduino COM7(Arduino Uno), ovviamente la scheda deve essere connessa al computer tramite la porta USB.



## Analisi dei dati sperimentali.

Apri il monitor seriale per monitorare l'esecuzione del programma. I dati che appaiono sul Monitor seriale vanno salvati in un file di testo per la successiva rielaborazione. Per fare ciò:

- seleziona e copia i dati dal monitor seriale
- salva su un file in *formato ASCII*: *nome\_file.txt*

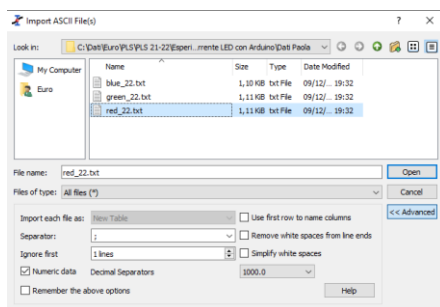
(si possono usare i tasti rapidi: Ctrl+A seleziona tutto, Ctrl+C copia ,Ctrl+V incolla).

## Analisi dei dati sperimentali tramite il software SciDavis

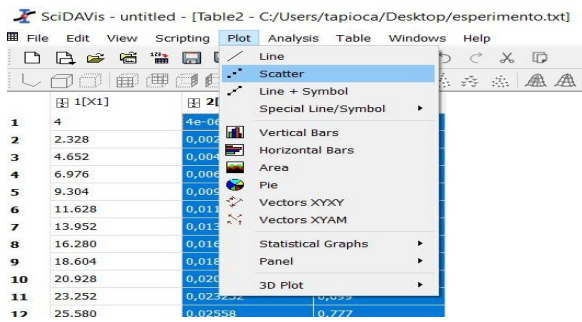
Il software SciDAVis è libero, open source, per scaricarlo vai al <http://scidavis.sourceforge.net/>.

### *Procedi come segue*

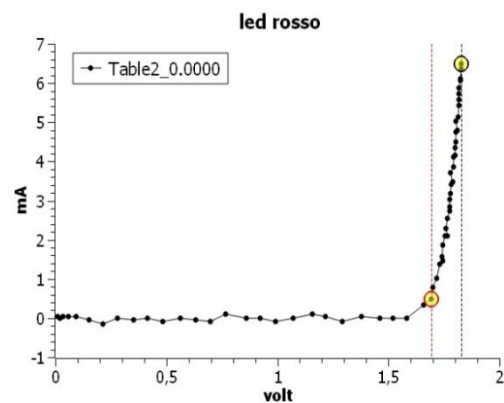
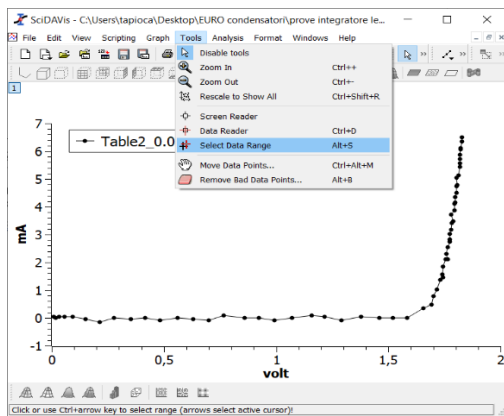
- Apri il programma SciDAVis nel menù *File* salva progetto in una cartella *Nome\_progetto* nel menù *File* cerca il comando *Import ASCII* e seleziona il *Nome\_file.txt* scegliendo come *Separator* il “;” e impostando di ignorare la prima riga (*Ignore first 1 lines*)



- Verifica che la prima colonna sia X e la seconda Y utilizzando il tasto destro del mouse;
- Selezionare entrambe le colonne mouse+Ctrl e sceglie dal menù *Plot* → *Scatter*. Si aprirà il grafico dei valori selezionati in una nuova finestra.



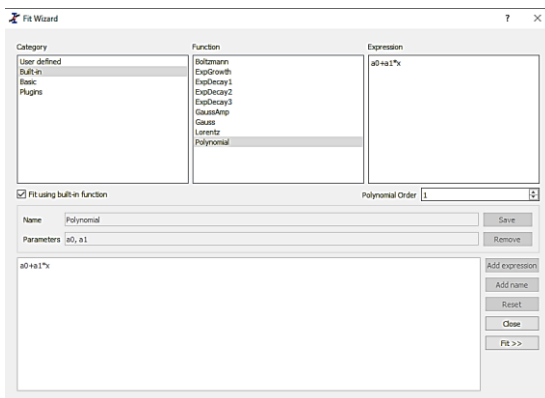
D. Si può notare che per valori di differenza di potenziale ( $\Delta V$ ) minori della *tensione di soglia* ( $V_0$ ) la corrente che passa nel LED è trascurabile, poi per tensioni maggiori ( $V > V_0$ ) il LED inizia a condurre e quindi a emettere luce. Per determinare il valore di  $V_0$  occorre fare un fit lineare della parte quasi rettilinea della curva: dal menù *Tool* selezionare l'opzione *Select Data Range* come mostrato in figura.



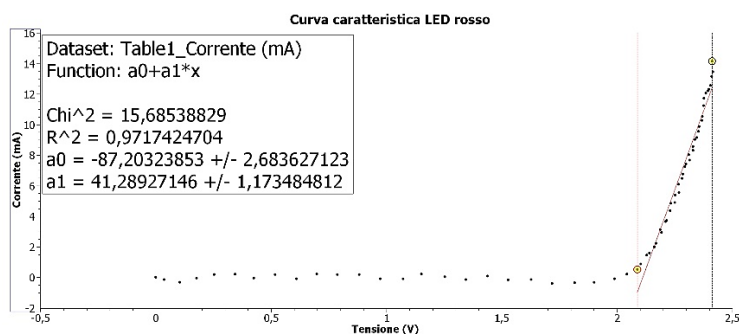
E. Mantenendo attiva la finestra del plot, selezionare dal menù *Analysis* l'opzione *fit Wizard*.. e scegliere nella categoria *Built-in* la funzione *Polynomial Order 1*.

Ricorda di spuntare il flag [v] *fit using built-in function* prima di eseguire il comando *Fit*;

Al termine completa il grafico inserendo il titolo, l'unità di misura sugli assi etc.. utilizzando un doppio click sugli elementi del grafico.



- Menù *Analysis*
- Opzione *Fit Wizard*
- *Built-in* → *Function Polynomial Order 1*
- *Fit* >>
- *Close*



È possibile determinare la tensione di soglia  $V_0$  mediante l'equazione della retta che meglio interpola i dati;  $V_0$  è il valore in cui la retta del fit interseca l'asse delle ascisse, e si ottiene dai parametri  $a_0$  e  $a_1$  del fit.

## La lunghezza d'onda della luce

La tensione di soglia ( $V_0$ ) moltiplicata per la carica dell'elettrone ( $e = 1,602 \cdot 10^{-19}$  C), fornisce l'energia dei fotoni emessi:

$$E = e V_0$$

Per determinare la lunghezza d'onda ( $\lambda$ ) della luce emessa dal LED basta combinare la formula precedente con la formula di Planck:

$$E = h f$$

dove  $f$  è la frequenza e  $h = 6,6 \cdot 10^{-34}$  J·s è la costante di Planck.

Poi mediante la relazione  $\lambda \cdot f = c$  (dove  $c = 3 \cdot 10^8$  m/s è la velocità della luce nel vuoto), si ottiene la lunghezza d'onda  $\lambda$ :

$$\lambda = \frac{c h}{e V_0}$$

Ripetere l'intera procedura per il LED blu.

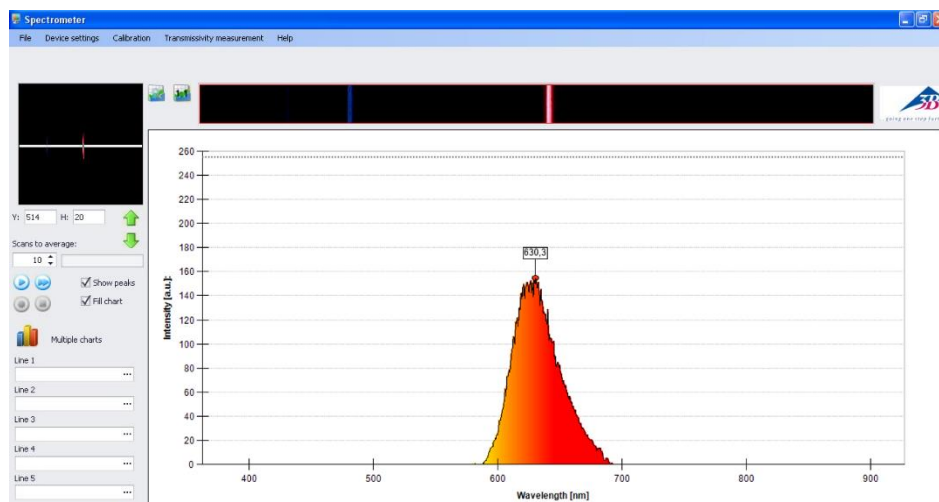
## Seconda parte dell'esperimento: misura della lunghezza d'onda emessa dal LED con lo spettrometro

- Spettrometro Spectra-1 della Kvant (range 360-940 nm, risoluzione < 2nm) con cavo flessibile in fibra ottica flessibile per ricevere il segnale ottico



- Aprire il software *Spectrometer*, impostare il campo *Scans to average* al valore 10, selezionare il campo *Show peaks* e *Fill chart* e avviare la misura cliccando sul tasto "Record spectrum", quindi cliccare sulla *freccia verde in basso*. Si dovrebbe ottenere uno spettro simile a quello mostrato nella figura seguente, in cui viene mostrato il valore massimo del picco che corrisponde al valore

della lunghezza d'onda dello spettro (l'incertezza è data dalla larghezza del massimo a metà altezza).



## Conclusioni

- Confrontare tale valore con quello ottenuto nella prima parte dell'esperimento dal valore di soglia.
- Ripetere l'esperimento usando il LED blu.