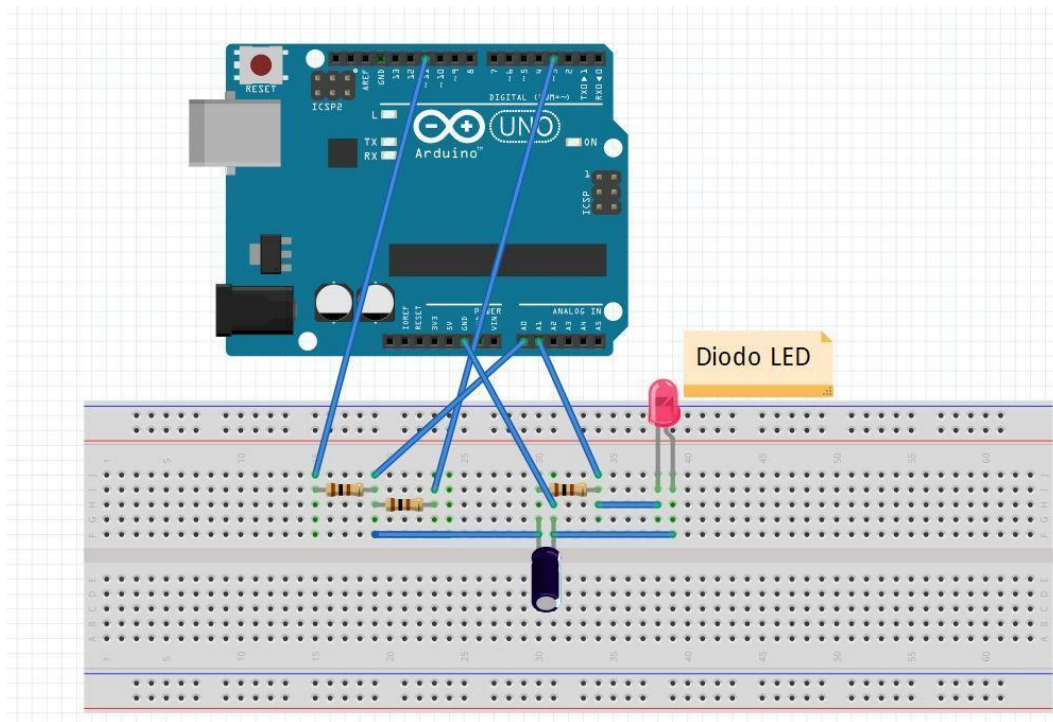


CARATTERISTICA I-V DI UN DIODO CON ARDUINO

Configurare il sistema come mostrato in figura:

- Tre resistenze (da 100 Ω) di cui una utilizzata per il filtro passa basso RC con frequenza di taglio $f = 100$ Hz;
- Un condensatore $C = 100 \mu\text{F}$ - il piedino A0 di Arduino collegato al polo positivo del condensatore;
- Il polo positivo del condensatore collegato alla terra GND;
- Un diodo LED con il catodo collegato a GND;
- Le due resistenze che non fanno parte del filtro collegate rispettivamente ai pin A11 e A3;
- Una Scheda Arduino UNO



L'esperimento consiste nel generare una tensione crescente da 0 a 5 Volt, da applicare al diodo. Una tensione variabile si può ottenere filtrando con un filtro RC, un segnale "Fast PWM" (pin A11 e Pin A3 scheda Arduino). La lettura della corrente I assorbita dal diodo è possibile grazie all'inserimento di una resistenza in serie al componente e leggendone la tensione ai suoi capi. Quest'ultima grandezza risulterà essere proporzionale alla corrente che attraversa il diodo. Pertanto la lettura con Arduino di V_0 e $V_0 - V_1$ (si ricorda che $I = (V_0 - V_1)/R$) consente di ottenere la coppia di valori corrente-tensione al fine di tracciare la caratteristica del diodo stesso.

I valori sono registrati in un file Numbers. I diversi gruppi possono lavorare su dati ottenuti con diodi tra loro diversi (es. LED, ZENER, SHOCKLEY, etc...)

Analisi dello Sketch Caratteristica I-V di un diodo

```
void setup() {
```

```
Serial.begin(9600);
```

```
pinMode(11,OUTPUT);
```

Qui mettiamo il codice di inizializzazione.

Inizializza tutte le impostazioni e le istruzioni della scheda (gli INPUT e OUTPUT) prima che il ciclo principale del programma si avvii

Prepara Arduino a mandare ed a ricevere dati tramite porta seriale. Generalmente si impongono 9600 bits

Imposta il pin A11 come OUTPUT

```

pinMode(3,OUTPUT);
pinMode(A0,INPUT);
pinMode(A1,INPUT);
TCCR2A = B10100011;
TCCR2B = B00000001;
}
void loop() {

if(Serial.available()>=3){
  if(Serial.read()>=255){
    byte val=Serial.read();

    OCR2A=val;
    OCR2B=val;
    byte nrep=Serial.read();

    delay(30);

    unsigned int v0=0;
    unsigned int v1=0;
    for(byte i=0; i<nrep; i++){

      v0+=analogRead(A0);

      v1+=analogRead(A1);

    }
    if(nrep>0){
      Serial.write(255);
      Serial.write(v0/256);
      Serial.write(v0%256);
      Serial.write(v1/256);
      Serial.write(v1%256);
    }
  }
}
}
}

```

Imposta il pin A3 come OUTPUT
 Imposta il pin A0 come INPUT di lettura
 Imposta il pin A1 come INPUT di lettura
 Abilitato il "FAST PWM" per le porte A e B
 Viene disabilitato il Prescaler del timer2

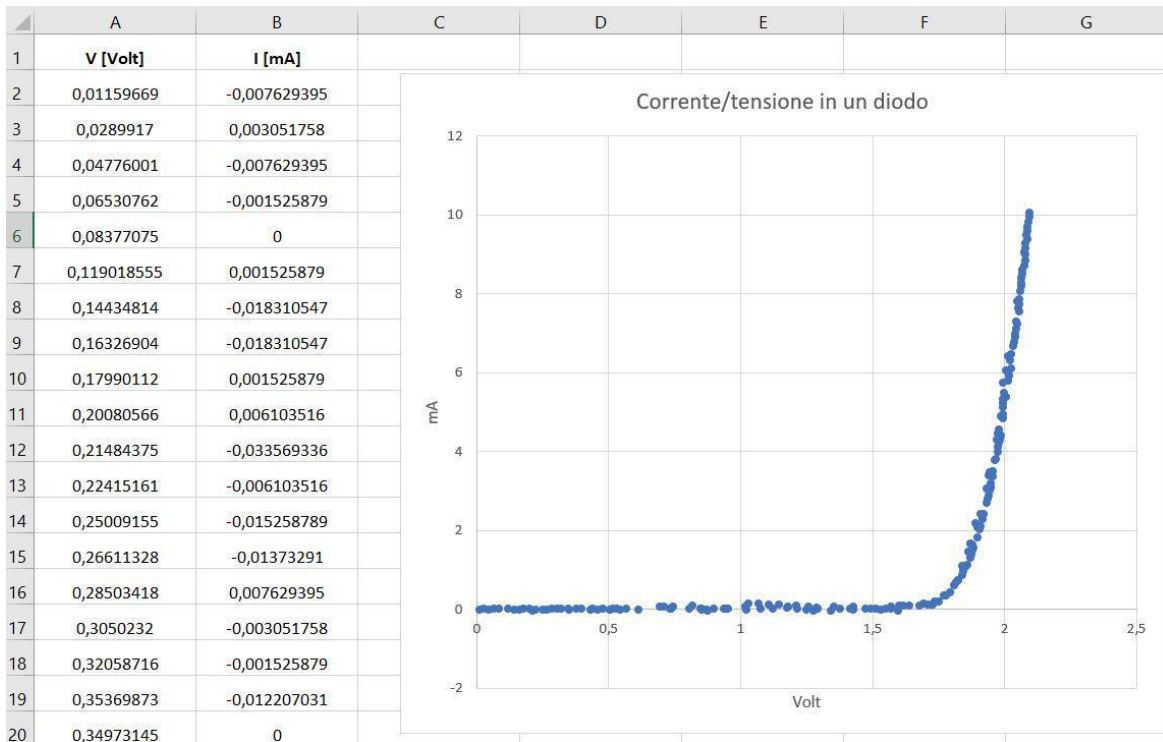
Rappresenta il contenitore del codice principale del programma. Contiene una serie di istruzioni che possono essere ripetute una dopo l'altra fino a quando non spegniamo la scheda Arduino

Dichiarazione di una variabile su cui memorizzare i valori letti
 Registro A
 Registro B
 Dichiarazione variabile e inizio lettura della seriale
 Indica il ritardo di lettura. Nel nostro caso 30 ms
 dichiarazione variabile v0
 dichiarazione variabile v1
 Ciclo for per la lettura dei valori delle tensioni ai capi del diodo e del resistore da cui si deduce la corrente assorbita dal diodo stesso
 lettura e aggiornamento valore della tensione su PIN A0
 lettura e aggiornamento valore tensione su PIN A1

La funzione scrive dati binari sulla porta seriale. I dati vengono inviati come byte o serie di byte.

<https://drive.google.com/open?id=1gfhvw0PFCP2l2S21LkSjMePreryCQKXI>

Dati acquisiti



Dati trasferiti su Geogebra per far lavorare i ragazzi

