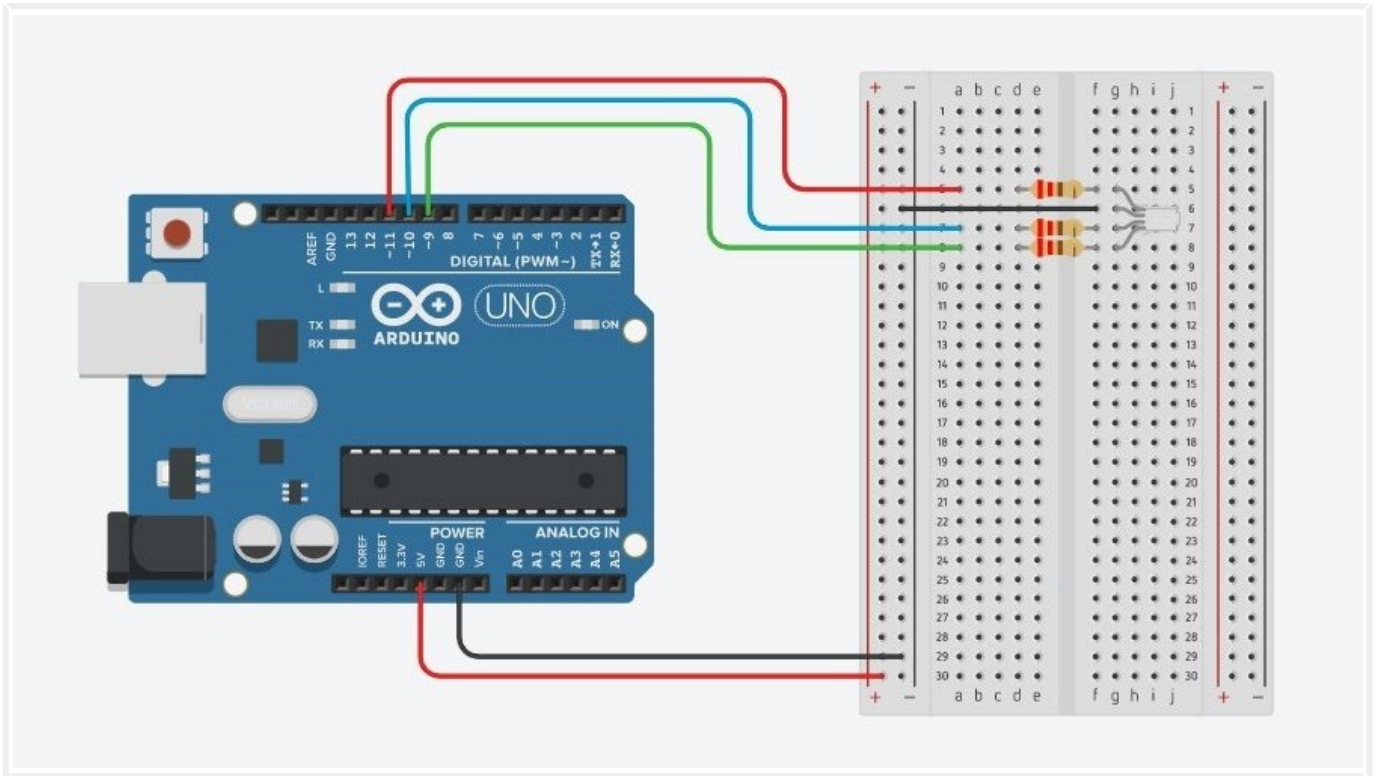


Serial - Conversion des données par la fonction "scanf()"

(conversion d'une chaîne de caractères en
nombre entier)



. Liste des composants

- . 1 DEL RVB
- . 3 résistances de 220 Ω (résistances de protection des DELs)
- . 1 plaque d'essai
- . Fils de connexion

. Objectif

Dans les programmes pour Arduino, Il est souvent nécessaire de convertir des chaînes de caractères en une variable représentant un nombre entier ou à virgule pour effectuer des calculs.

Après la fonction "atoi()", une autre méthode de conversion est d'utiliser la fonction "sscanf()" qui permet de couper et de convertir une chaîne de caractères en une série de variables allant du caractère unique (type "char") aux nombres décimaux (type "int", "float").

Exemple de conversion d'une chaîne de caractères en variables (programme "sscanf.ino") :

```
sscanf

int jour;
int an;
char str[20] = "08 Fevrier 2020";
char mois[10]="";

void setup() {
  Serial.begin(9600);
  sscanf(str, "%d %s %d", &jour, mois, &an );
  Serial.println(jour);
  Serial.println(mois);
  Serial.println(an);
}

void loop() {
}
```

Cet exemple permet d'extraire le jour, le mois et l'année d'une chaîne de caractère représentant une date :



- Le premier argument de la fonction est "str". Il représente la chaîne de caractère à convertir,
- l'argument "%d" indique à la fonction qu'il faut extraire un nombre décimal,
- l'argument "%s" pour extraire une chaîne de caractères (la fin de la chaîne est repérée par un espace, un saut de ligne),
- les autres arguments sont les variables dans lesquelles il faut stocker les valeurs converties (les variables commençant par un "&" sont destinées à recevoir un nombre).

La fonction "sscanf()" retourne le nombre de variables qu'elle a pu convertir, si cette valeur est différente du nombre d'arguments de conversion c'est qu'il y a eu une erreur.

En application, le programme suivant, utilisant la fonction **"sscanf()"**, permet de régler la luminosité de la DEL rouge du circuit d'étude, en retour d'une chaîne de caractères envoyé depuis le moniteur série dont la conversion en nombre entier doit être compris entre 1 et 255.

. Le programme

```
Serial_Conversion_sscanf

const int PinLEDR = 11;
int Luminosite = 0;

void setup() {
  Serial.begin(9600);
  pinMode (PinLEDR, OUTPUT);
  digitalWrite(PinLEDR,0);
}

void loop() {
  while(Luminosite<1 || Luminosite>255)
  {
    int Val=0;
    char tampon[10]="";
    Serial.println("Veuillez saisir la luminosité de la DEL rouge (valeur entre 1 et 255):");
    while(!Val)
    {
      delay(200);
      Val=Serial.available();
    }
    for (int i=0; i < Val; i++)
    {
      tampon[i]=Serial.read();
      delay(15);
    }
    int N=0;
    char c;
    N = sscanf(tampon, "%d %c", &Luminosite, c);
    if (N!=1){
      Luminosite=0;
    }
  }
  Serial.print("Luminosité de la DEL rouge choisie : "); Serial.println(Luminosite);
  Serial.println("");
  analogWrite(PinLEDR,Luminosite);
  Luminosite=0;
}
```

On retrouve dans ce programme, la partie de code permettant de recevoir une chaîne de caractères depuis le port série. La chaîne est convertie en nombre entier avec la fonction "`scanf()`". Si le résultat n'est pas compris entre 1 et 255 ("`while(Luminosite<1 || Luminosite>255)`"), l'utilisateur doit saisir de nouveau, une valeur.

Voici le résultat dans le moniteur série :

