08-02 Programmer l'interface ARDUINO:

Se fait après avoir penser le montage, **mis les bonnes résistances pour protéger le matériel**(surtout les led en fonction de leur couleur).

Aide en élec :

L'additivité des tensions s'écrit : UMatériel + UResistance = 5V et U=R.I.

exemple pour une **led verte**(ou jaune) fonctionnant de façon nominale avec une **tension de 2,2 V et un courant de 20 mA** .(Rouge : 1,6V pour 20mA, Bleue 3,3 pour 20mA)

R à ajouter en série = (Ualim - Uled) / Iled = (5 - 2,2) / 0,02 = 340 Ohm

ULED est égale à la tension de seuil (supposée indépendante du courant) on cherche donc à éviter que notre led prennent une tension trop éloignée de sa valeur seuil.

0- on importe les éventuelles bibliothèques contenant les drivers nécessaires en fonction du matériel utiliser.

1- d'abord on **affecte** des valeurs :

- aux variables tout en indiquant qu'il s'agit du type de variable :

exemple avec un int :

```
int led=13 ;
```

→ ne pas oublier le ; Les points virgules terminent les instructions.

→ il existe plein de types de variables principales : int(nombre de -32000 à +32000, long nombre de – beaucoup à + bcp), char(chaîne de caractères)

un entier nommé led prend la valeur 13. On s'en servira pour affecter la broche 13 à une instruction.

- des constantes :

```
#define PI 3,14        doute sur l'absence de signe =
```

→ il ne faut pas de ;

2- 1ere fonction obligatoire pour programmer l'Arduino:

void setup() → **fonction d'installation s'exécute une fois au début !**

{ ….}

3- 2eme fonction obligatoire pour programmer l'Arduino:

void loop() → fonction en boucle dés que l'on branche l'Arduino

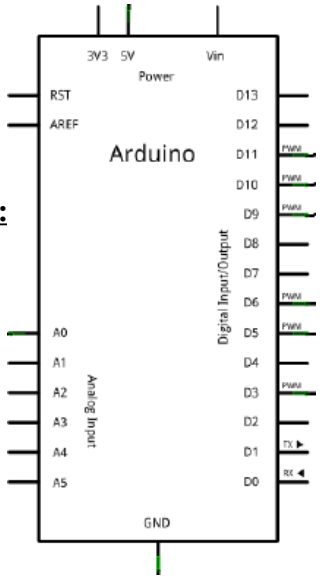
{ …} boucle

4- on branche l'arduino relié au montage effectué, on téléverse sur l'arduino : c'est lancé !

Exemple 1: gérer un allumage alternatif d'une led rouge reliée à la liaison digitale 5(5V) et protégée par une résistance(calculer sa valeur!)

Ressources schéma :

plateforme Arduino :



LED :

à gauche à brancher côté -, à droite côté +.

Résistance :

Ressources programmation :

pinMode(8,OUTPUT); → on a configuré la broche 8 en mode sortie

digitalWrite(X;Y) ; → est une fonction qui donne un état électrique Y(0 ou 1) à une broche X.

delay(30); → est une instruction commandant un temps d'attente en ms dans la lecture du programme.

ATTENTION c'est en langage « presque C », deux trois bases :

affectation :

```
    int trou =5   # affecte la valeur entier 5 à trou
```

conditions :

```
    if(x>y)
    {……..}
    for(n=0 ; n<10 ; n++) #de 0 à 10 ajoute 1 à n à chaque fois.
    {……..}
    while(x>=r)
    {……..}
```

le ET logique s'écrit &&

le OU logique ||

entre accolades, il y a ce qui se passe dans les fonctions setup() ou loop()

→ →Projet : réaliser un feu tricolore autonome.(Programme Arduino ET montage électrique!!!)

Exemple2 : Acquisition de données :
Pour utiliser un capteur capteur(transducteur) qui mesure la luminosité. Le transducteur qui opère avec cette grandeur est une *photorésistance* ou *LDR* (Light Depending **Resistor). C'est une résistance photo-sensible** La relation établie par la photorésistance entre la luminosité et sa résistance de sortie permet d'avoir une image de l'intensité lumineuse par une résistance électrique qui varie selon cette
intensité.

Ressources Montage et réflexion :
Une LDR(photo et représentation schématique) :

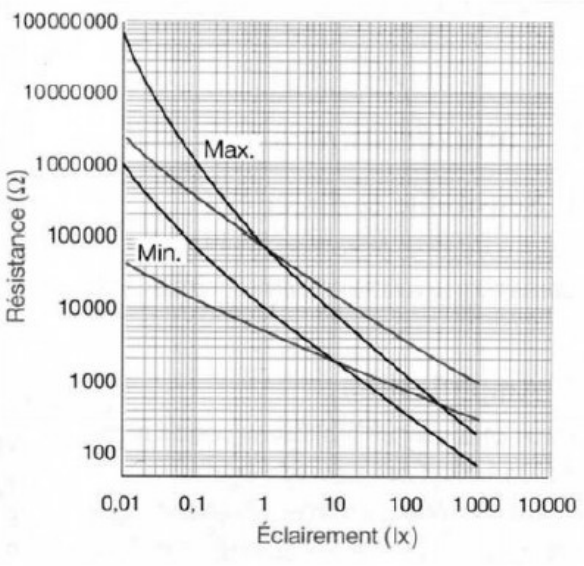
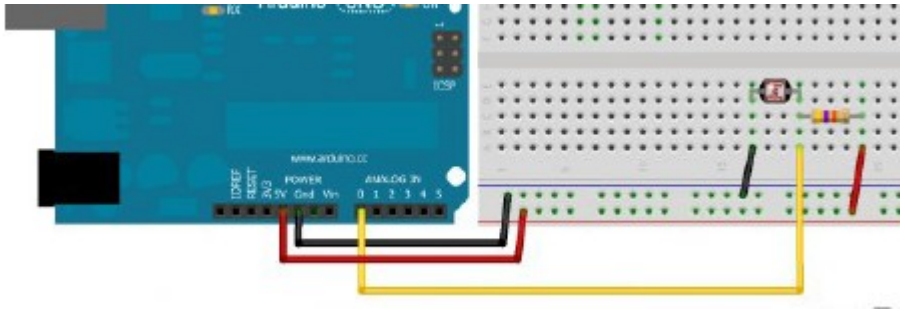


FIGURE 22.9. – Caractéristique d'une photorésistance



Pour des raisons de pont diviseur de tension… le montage étant complexe : je le donne.

Ressources programmation :
Serial.begin(9600); // ouvre le port 9600 pour que l'ordi affiche la température relevée
pinMode(8,INPUT); → on a configuré la broche 8 en mode entrée
analogRead(pin); → lis la valeur d'entrée du pin analogique, valeurs de 0 à 1024
digitalRead(pin); → lis la valeur d'entrée du pin digital, valeurs de 0 ou1

Serial.print("un truc à écrire");
Serial.print(variable); //envoie de la valeur de la tension lue vers l'ordinateur via la liaison série

Projet : un appareil qui éclaire d'autant plus qu'il fait sombre…. Et qui s'éteint lorsqu'il fait jour !
Ressources programmation :

```
if(variable <= valeur)
{
digitalWrite(led1, LOW); // On allume la LED branché sur pin led1 digital, soit LOW, soit HIGH
}
else
{
analogWrite(led, variable); // On envoie un courant de valeur comprise entre 0 et 255 (variable amis moins précis qu'en réception !!! 8 bits au lieu de 10) vers la LED branchée sur le pin analogique pin2tension la LED
}
```

→ →Projet : réaliser un montage éclairant de plus en plus fort en fonction de l'obscurité et qui s'éteint au dessus d'une valeur seuil. (Programme Arduino ET montage électrique!!!)