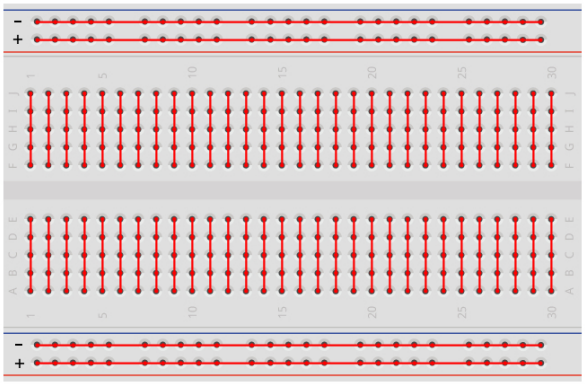
Présentation du matériel

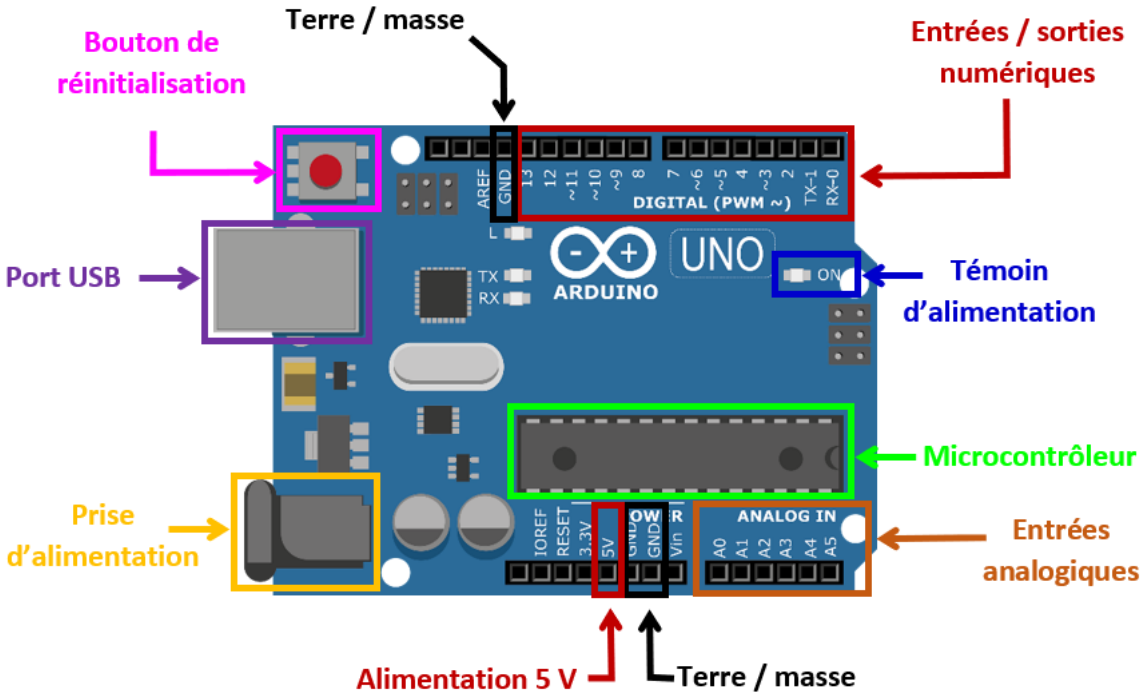
Platine

Pour réaliser des circuits avec des composants électroniques, on utilise une *breadboard* constituée de bornes connectées entre elles selon le schéma ci-dessous. Les lignes + et − sont utilisées respectivement pour la tension d'alimentation (5 V) et la masse.



Carte Arduino

Une carte Arduino est un circuit imprimé sur lequel se trouve un microcontrôleur qui peut être programmé pour mesurer et produire des signaux électriques.



L'alimentation de la carte se fait *via* le port USB, qui fournit également le moyen de communiquer avec le microcontrôleur. La tension d'alimentation de 5 V et la masse associée sont accessibles sur la carte.

Les bornes ANALOG IN sont des bornes d'entrée analogique, qui convertissent la tension lue en nombre binaire codé sur 10 bits. C'est-à-dire qu'elle renvoie renvoie l'entier $k \in [\![0;1023]\!]$ lorsque la tension lue vaut $\dfrac{k}{1023} \times 5\,\text{V}$.

Les bornes DIGITAL numérotées de 1 à 13 sont des connecteurs numériques (utilisés le plus souvent comme sortie) :

- celles dont le numéro est précédé d'une vague fournissent des signaux de tension comprise entre 0 et 5 V sous forme d'une fraction codée sur 8 bits (c'est-à-dire de valeurs $\dfrac{k}{255} \times 5\,\text{V}$ avec $k \in [\![0;255]\!]$) ;
- les autres n'existent que sous deux états : état bas (LOW) de tension 0 V et état haut (HIGH) de tension 5 V.

Logiciel

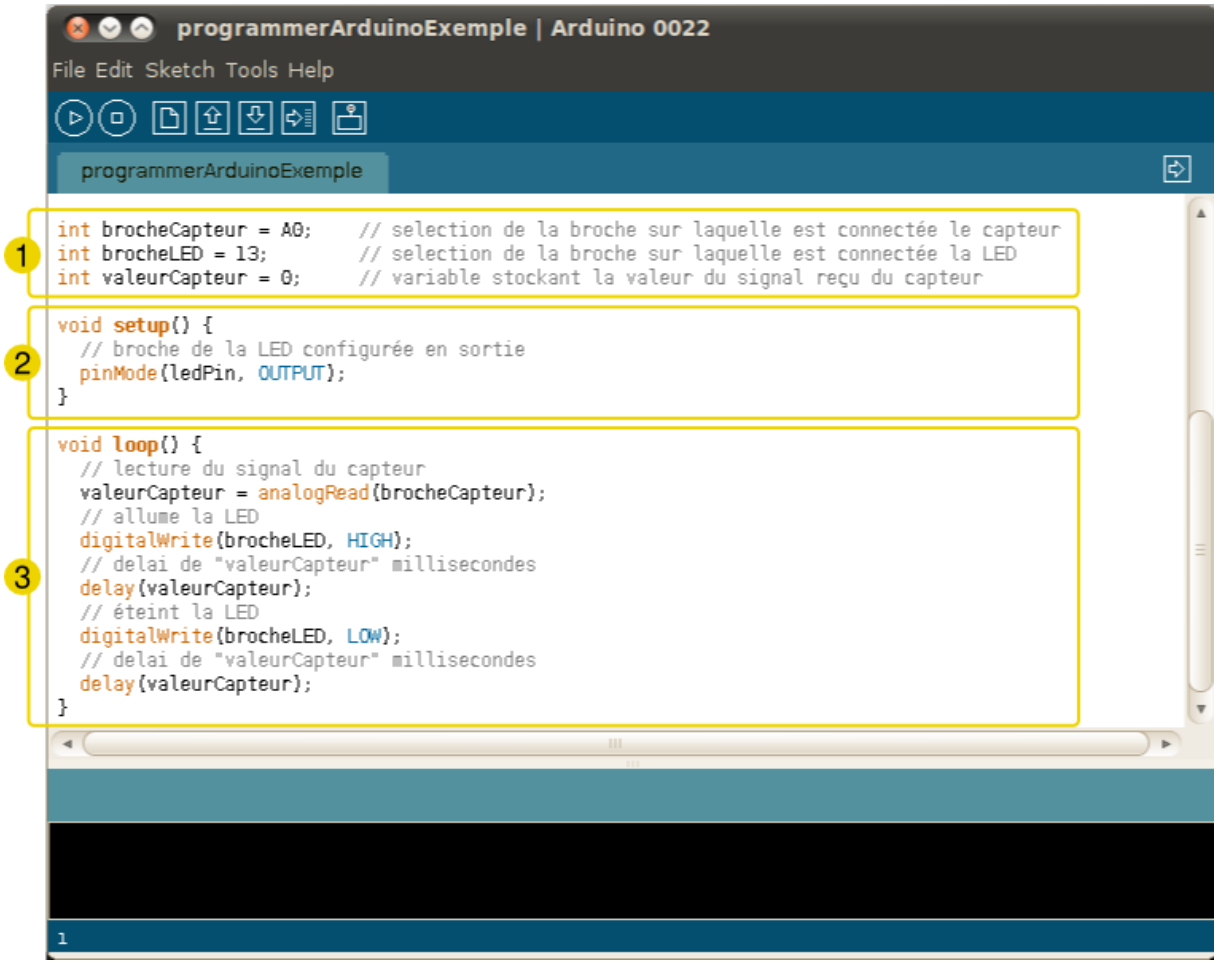
Le logiciel *arduino* (téléchargeable gratuitement sur le site officiel arduino.cc) permet d'écrire les programmes que l'on souhaite installer sur le microcontrôleur.

La structure de base d'un programme est la suivante [1] :

— au début (zone 1), on définit les variables globales utilisées dans la suite (optionnel).

— la fonction *void setup()* (zone 2) permet d'initialiser la carte au démarrage. On y définit les entrées et sorties utilisées.

— la fonction *void loop()* (zone 3) contient les instructions qui sont répétées en boucle après initialisation.

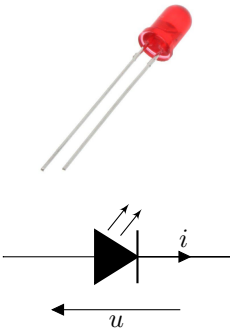


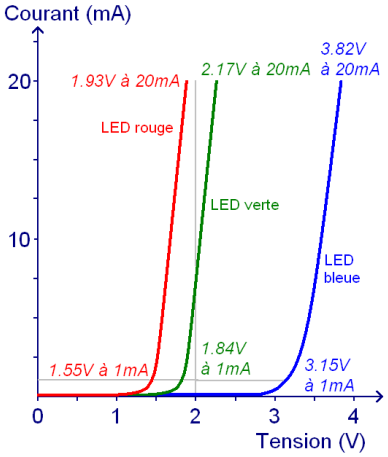
Pour téléverser le programme sur le microcontrôleur, il faut tout d'abord choisir le port sur lequel est branchée la carte (dans le menu *Outils*) puis cliquer sur l'icône . Un menu propose d'enregistrer le programme, c'est optionnel.

1. L'information complète sur la syntaxe et les structures du langage sont disponibles sur le site : http ://arduino.cc/fr/Main/Reference

Partie A. Premier projet : feu tricolore

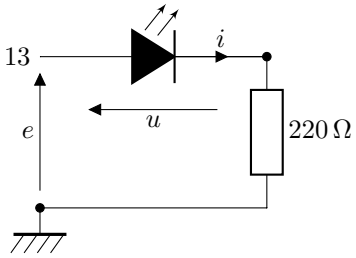
Une diode est un dipôle constitué d'une anode (symbolisée par un triangle) et d'une cathode (symbolisée par une barre, et correspondant à la patte courte du composant). Elle ne laisse passer le courant que dans le sens direct $i > 0$ (anode $\rightarrow$ cathode) lorsque la tension u imposée dépasse un certain seuil caractéristique. Une diode électroluminescente (LED) émet de la lumière lorsqu'elle est passante. La tension seuil est liée à la couleur de la LED.





L'objectif est d'éclairer une LED de façon intermittente.

❏ Brancher le circuit ci-dessous sur la platine. La masse et l'entrée numérique 13 correspondent à celles du microcontrôleur.



❏ Pour une tension d'entrée $e = 5\,\text{V}$, évaluer le courant circulant dans une LED rouge en suppposant que sa tension est égale à la tension seuil. Sachant que l'intensité maximale supportée par une LED est de 20 mA, en déduire l'intérêt de la résistance introduite dans le circuit.

❏ Recopier le programme suivant dans le logiciel *Arduino* :

```
void setup(){
  pinMode(13,OUTPUT);
}
void loop(){
  digitalWrite(13,HIGH);
  delay(1000);
  digitalWrite(13,LOW);
  delay(1000);
}
```

Explications :

- lors de l'initialisation, la broche 13 sur laquelle est branchée la LED est configurée en sortie.
- digitalWrite(13,HIGH) alimente cette borne avec la tension de service de 5 V, ce qui allume la LED.
- digitalWrite(13,LOW) passe la tension à 0, la LED s'éteint.
- delay(n) crée un temps d'attente de n millisecondes avant de passer à l'instruction suivante.

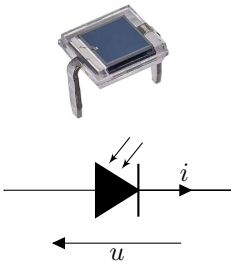
❏ Téléverser le programme. Observer et interpréter.

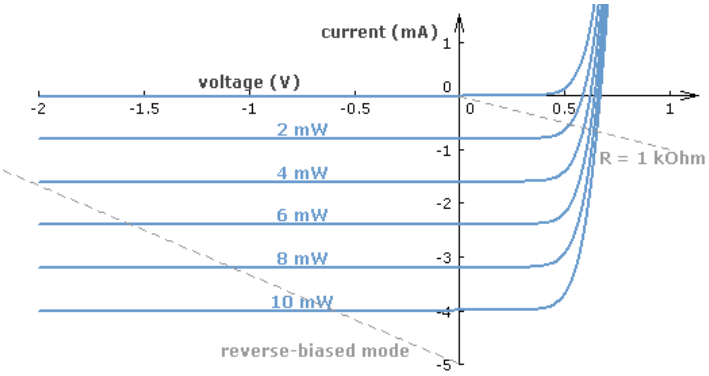
❏ Compléter le montage et le programme afin de réaliser un feu tricolore : le feu vert dure 2 secondes, l'orange une demi-seconde et le rouge 2 secondes.

Partie B. Deuxième projet : veilleuse

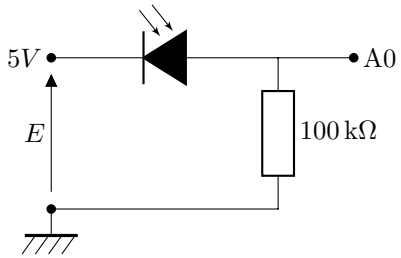
Une photodiode est constitué comme une diode. L'anode est reconnaissable à la bande métallisée sur le dessus.

Une photodiode fonctionne de façon inverse d'une LED. Branchée en inverse ($u < 0$), elle produit un courant lorsqu'elle reçoit de la lumière. L'intensité du courant inverse est proportionnelle à l'intensité lumineuse.





❏ Brancher le circuit ci-dessous, qui est alimenté par une tension de 5 V fournie le microcontrôleur, et qui contient également une borne de sortie permettant au microcontrôleur de mesurer la tension aux bornes de la résistance.



❏ Quel est le double rôle de la résistance dans le montage ?

❏ Recopier le programme suivant dans le logiciel :

```
int eclairement;
void setup(){
  Serial.begin(9600);
}
void loop(){
  eclairement = analogRead(A0);
  Serial.println(eclairement);
  delay(100);
}
```

Explications :

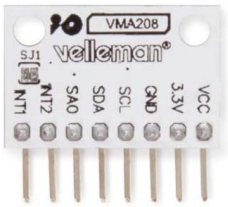
- on définit la variable *eclairement* contenant la valeur entière correspondant à l'intensité lumineuse.
- Serial.begin(9600) initialise la communication par le port série qui permet de transmettre les valeurs mesurées à l'ordinateur. La valeur 9600 correspond au débit (grosso modo en bit/seconde), que l'on peut augmenter si besoin.
- analogRead(A0) lit la valeur de la tension sur l'entrée A0 (sous forme d'un entier compris entre 0 et 1023).
- Serial.println(variable) transmet la variable par le port série.

❏ Téléverser le programme. Afficher les résultats des mesures en lançant le *traceur série* à partir du menu *Outils* : on obtient un graphe de l'éclairement en fonction du temps. Vérifier le bon fonctionnement de ce capteur de lumière.

❏ Compléter le montage et le programme afin de réaliser une veilleuse qui allume une LED lorsque l'éclairement de la photodiode devient trop faible. Pour gérer le seuil, on utilisera une structure conditionnelle dont la syntaxe est :

```
if (condition)
{
  action quand la condition est vraie (TRUE);
}
else
{
  action quand la condition est fausse (FALSE);
}
```

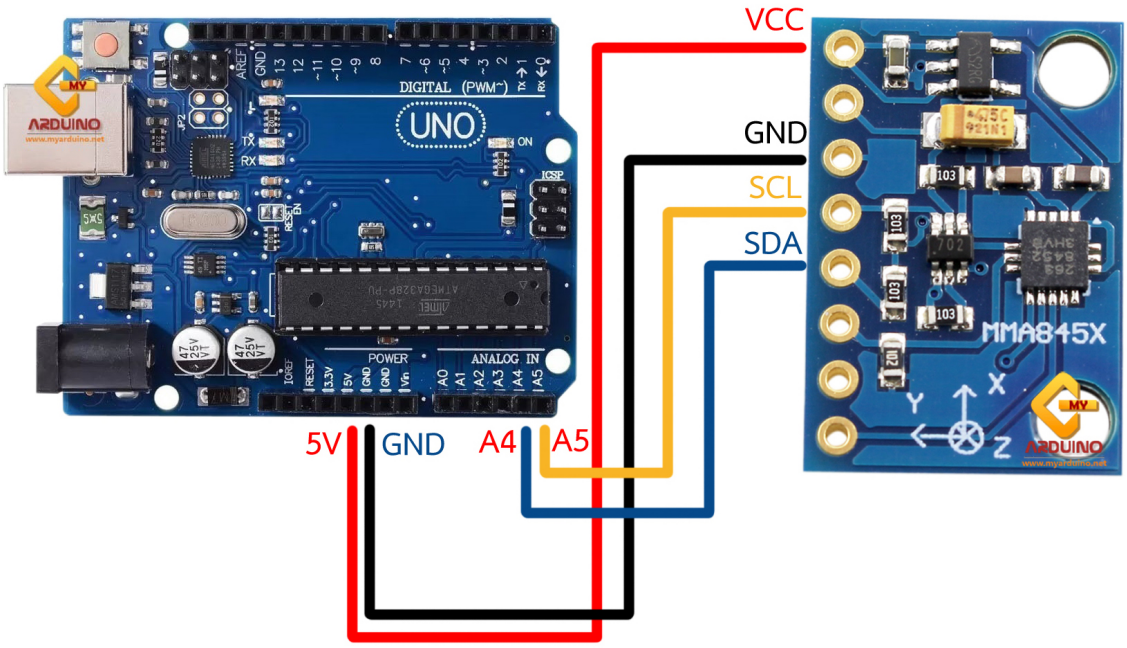
Partie C. Troisième projet : accéléromètre



Un accéléromètre (voir photo ci-contre) mesure les composantes de l'accélération absolue selon trois axes. En pratique il mesure l'étirement d'un ressort microscopique qui maintient un solide en équilibre dans le boîtier. De ce fait, la mesure inclut l'accélération de la pesanteur.

Les branchements à l'accéléromètre nécessitent 4 fils : 2 pour l'alimentation (GND = masse et VCC = 5 V) et 2 pour recueillir les données de l'accélération (SDA) et de l'horloge (SCL).

❏ Réaliser le montage ci-dessous.



❏ Enregistrer le fichier « SFE-MMA8452Q.zip » sur l'ordinateur, ainsi que le fichier « accelerometre.ino ». Ces fichiers sont disponibles sur cahier-de-prepa, dans le sous-menu TP/Arduino.

❏ Dans le logiciel *Arduino*, sous l'onglet *Croquis*, cliquer sur *Inclure une bibliothèque*, puis *Ajouter la bibliothèque .ZIP*. Sélectionner le fichier « SFE-MMA8452Q.zip »

❏ Ouvrir le fichier « accelerometre.ino », puis le téléverser sur le microcontrôleur.

❏ Afficher les résultats des mesures en lançant le *moniteur série*, et vérifier le bon fonctionnement de l'accéléromètre.

❏ Compléter le programme de façon à émettre une lumière lorsqu'un mouvement est détecté.