MARTIGNAS

Stage
Initiation à la programmation et à l'électronique avec ARDUINO

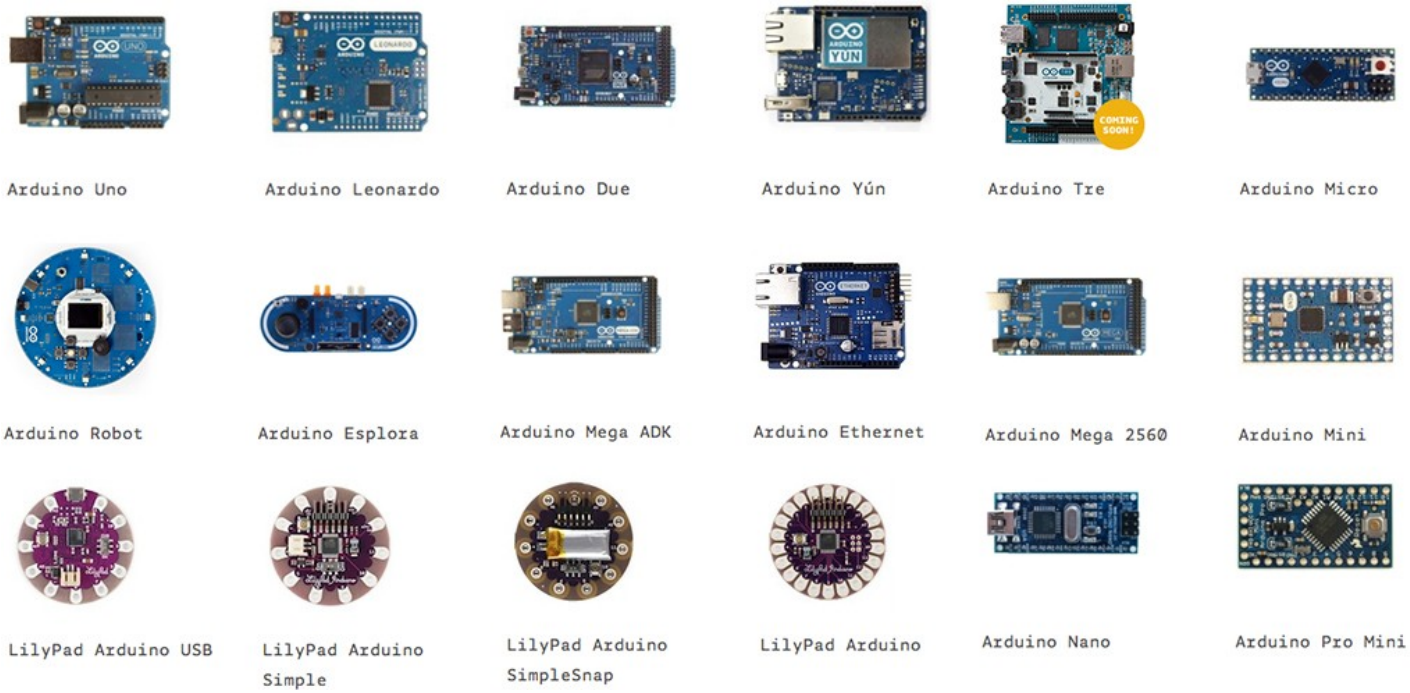
Jour 1

Programme du jour 1

- Découverte de la plateforme ARDUINO UNO R3
  - Ressources, E/S, Alimentation
- L'environnement de développement ARDUINO IDE
- Les bases de l'électronique
  - Tension, courant
  - Résistance, capacité
  - Diode LED, Microcontrôleur ATmega328
- PROJET 1: Faire clignoter une LED
  - Câblage
  - Programmation

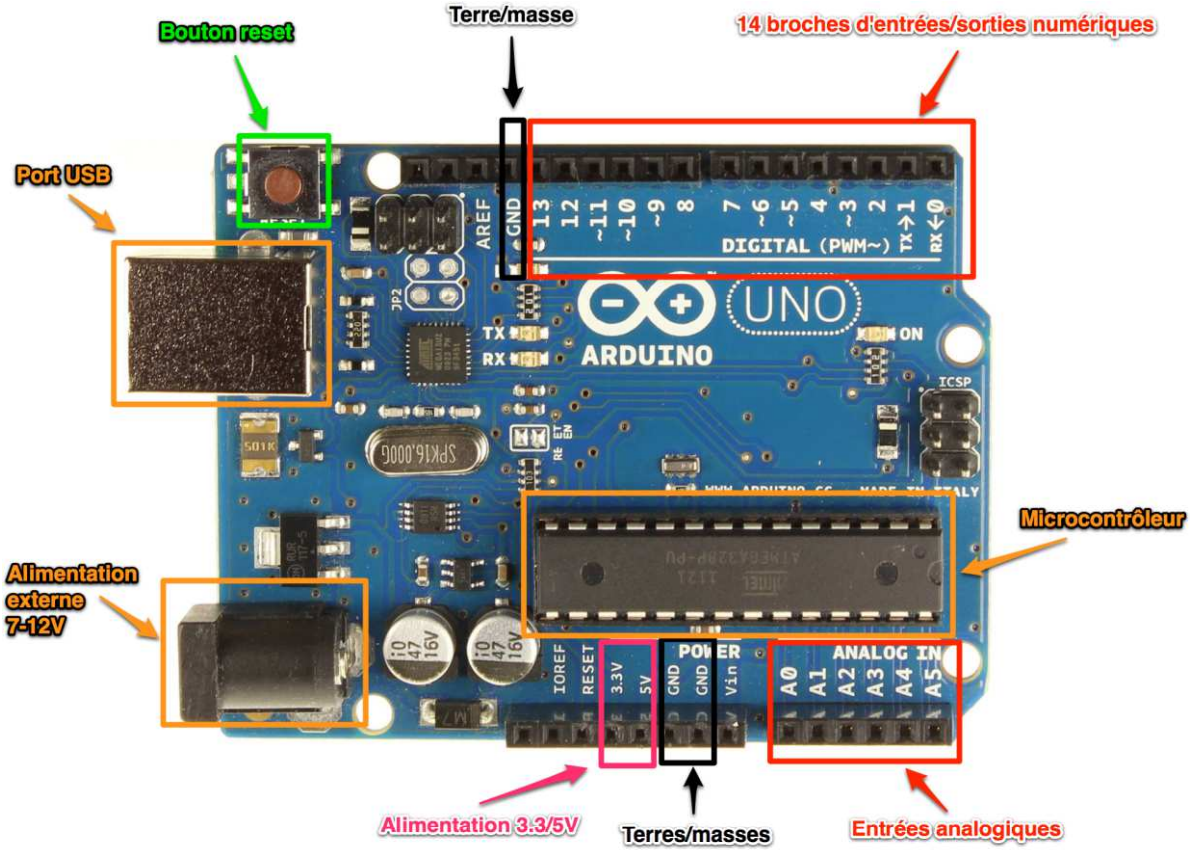
Plateformes ARDUINO

- Arduino est une platine de développement composée d'un circuit imprimé équipé d'un microcontrôleur.
- Arduino est open source, ainsi il existe un grand nombre de clones et de platines compatibles, tout comme il existe de nombreux modèles d'Arduino officiels, avec des fonctions particulières.



- Pour ce cours, nous nous baserons sur le plus connu: l'Arduino Uno

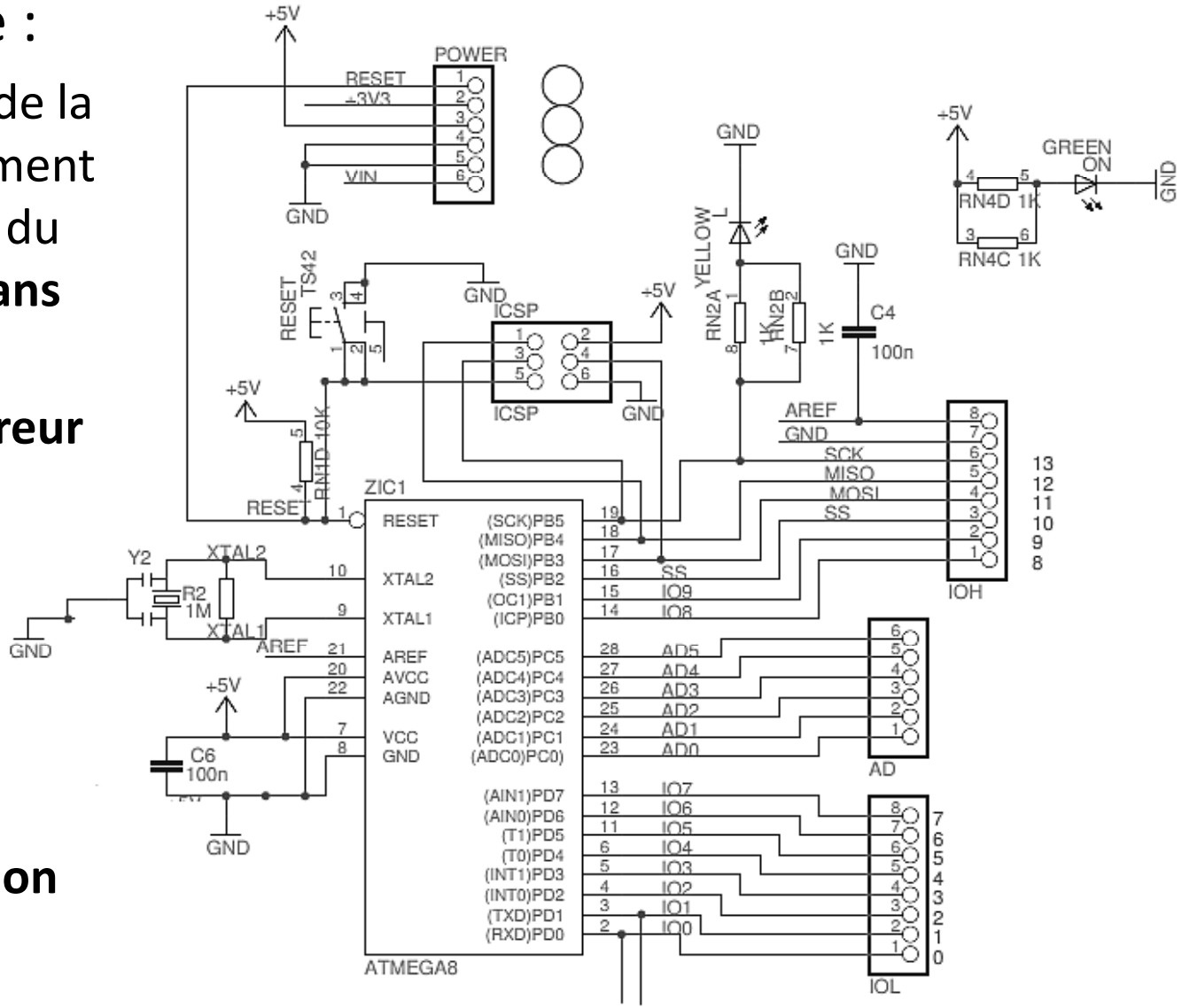
ARDUINO UNO



- Le microcontrôleur ATMega328 est le cerveau de notre carte. Il va recevoir le programme que nous allons créer et va le stocker dans sa mémoire avant de l'exécuter. Grâce à ce programme, il va savoir faire des choses, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur, mettre en route ou arrêter un moteur …

ARDUINO UNO

- Schéma de la carte :
  - Les entrés sorties de la carte sont directement reliées aux pattes du microcontrôleur **sans protection**
  - **Attention : une erreur de câblage peut amener à la destruction du microcontrôleur**
  - **Ne pas faire de modification de câblage sous tension**

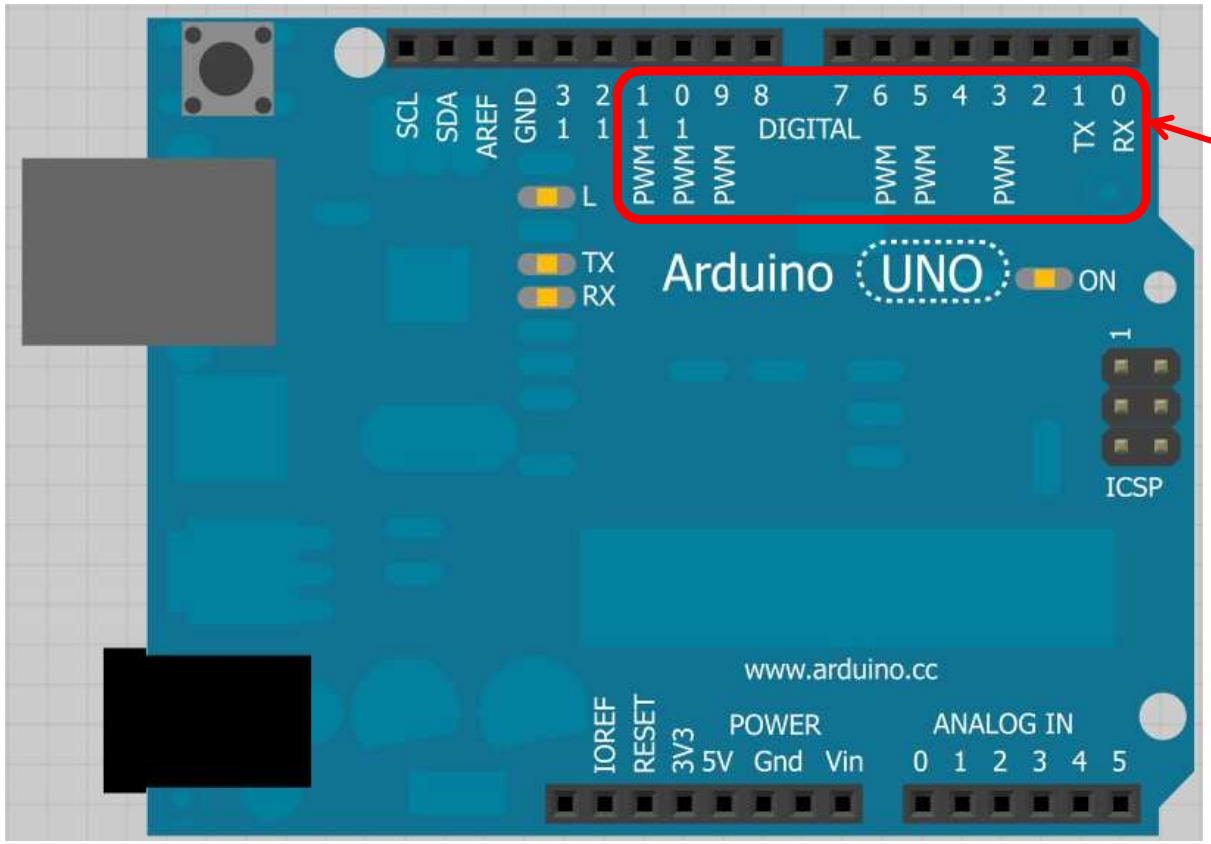


Alimentation et E/S

- **L'alimentation**
  - Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB ou bien par une alimentation externe qui est comprise entre 7V et 12V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte.
- **La connectique Entrées / Sorties**
  - **0 à 13** Entrées/sorties numériques (pin multifonctions voir planche suivante)
  - **A0 à A5** Entrées analogiques
  - **GND** Masse (0V)
  - **5V** Sortie Alimentation +5V
  - **3.3V** Sortie Alimentation +3.3V
  - **Vin** Alimentation non stabilisée (= le même voltage que celui à l'entrée de la carte)
  - **USB** Connexion au PC pour alimentation et transfert du programme

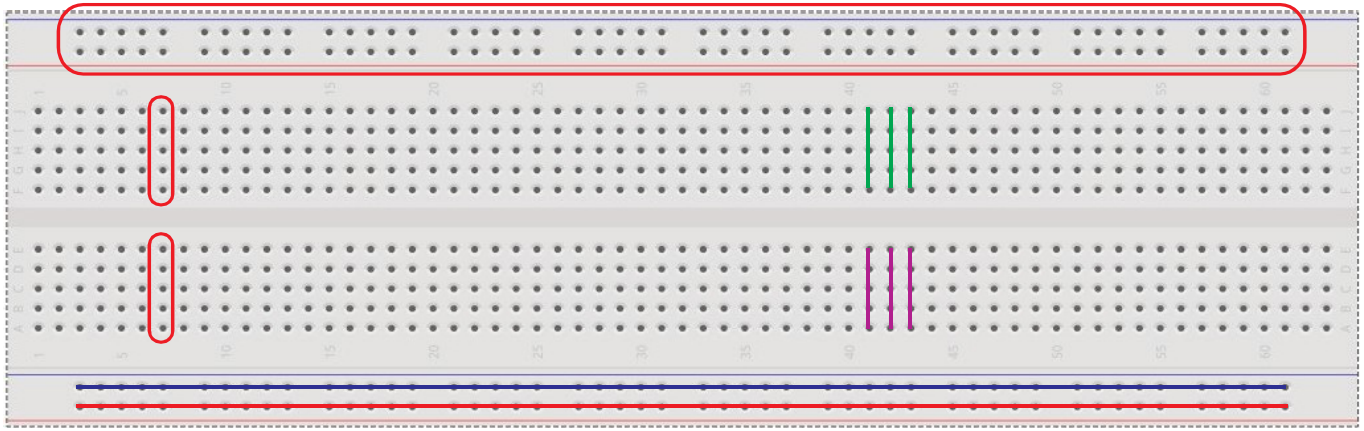
Alimentation et E/S

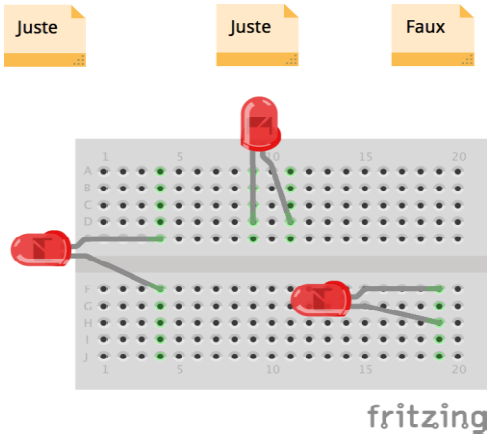
- Les entrées sorties numériques sont multifonctions, et donnerons accès au besoin à une liaison série (TX, RX) ou à des sorties «analogiques » (PWM)



Pin multifonctions

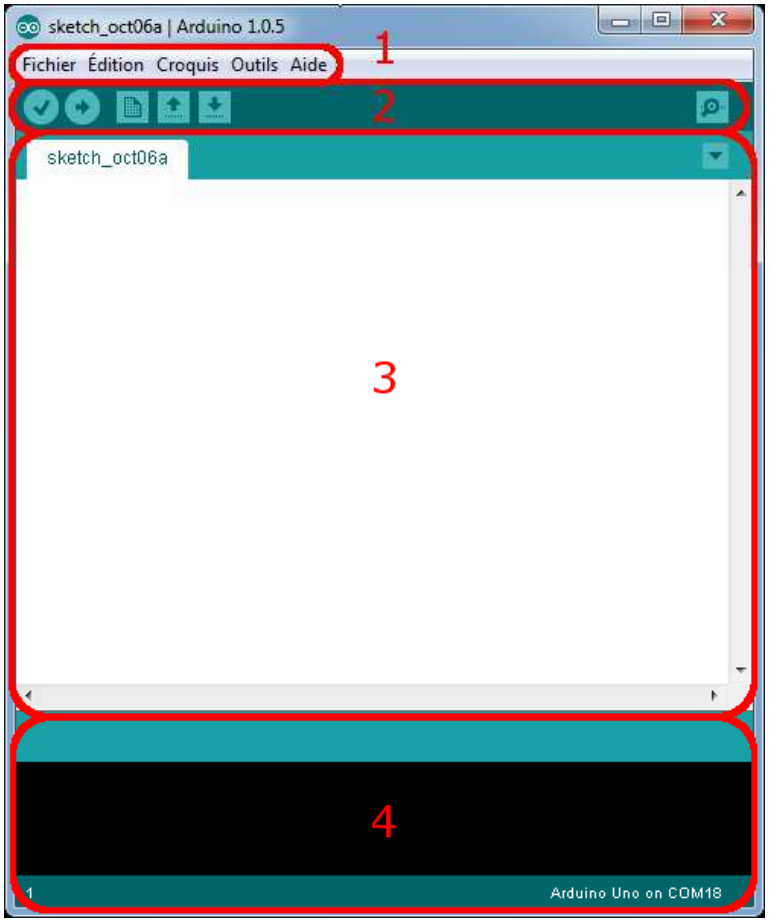
Platine d'expérimentation

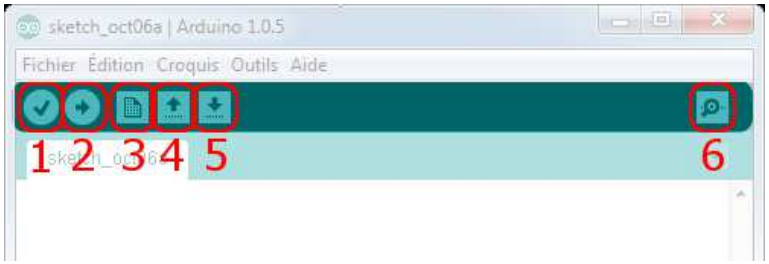




- Tous les connecteurs d'une ligne extérieure sont électriquement reliés entre eux
- Tous les connecteurs dans une rangée de 5 sont reliés entre eux.
- Si on branche deux composants dans un groupe de cinq connecteurs, ils seront reliés entre eux.
- En haut et en bas les alignements de connecteurs rouges sont pour l'alimentation et les bleus pour la masse
- Les composants doivent ainsi être placés à cheval sur des connecteurs qui n'ont pas de liens électriques entre eux

ARDUINO IDE

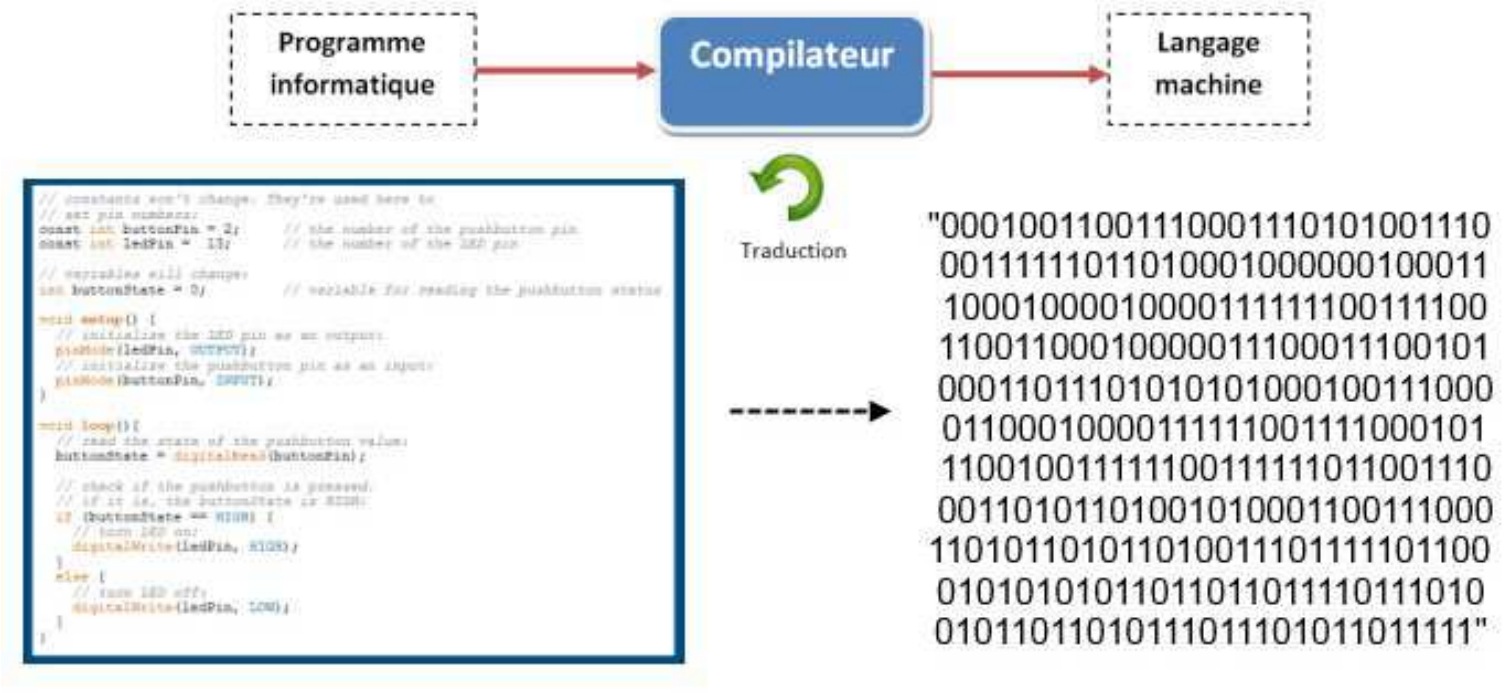




- Cadre numéro 1 : contient les menus de configuration du logiciel
- Cadre numéro 2 : contient les boutons qui vont nous servir lorsque l'on va programmer la carte
- Cadre numéro 3 : va contenir le programme que nous allons créer
- Cadre numéro 4 : est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur

- Bouton 1 : permet de compiler le programme, et de trouver d'éventuelles erreurs
- Bouton 2 : compile et charge le programme dans la carte Arduino.
- Bouton 3 : crée un nouveau fichier (croquis).
- Bouton 4 : ouvre un fichier (croquis).
- Bouton 5 : enregistre le fichier (croquis) en cours.
- Bouton 6 : ouvre le moniteur liaison série.

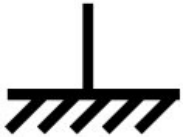
Compilation du programme



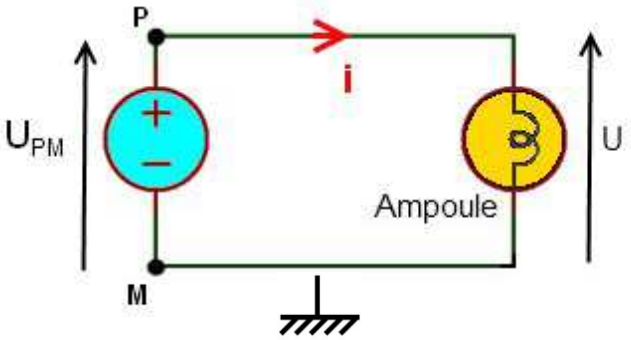
- Au départ, le programme est sous forme de texte (langage C), puis il est transformé en langage machine (sous forme de 1 et 0)
- L'envoi du programme est géré par le PC: le programme passe, dans le câble USB qui relie l'ordinateur à la carte, arrive dans la carte et est stocké dans la mémoire Flash du microcontrôleur.

Bases de l'électronique

- La masse
  - La masse est, en électronique, le point de référence à 0 Volts.



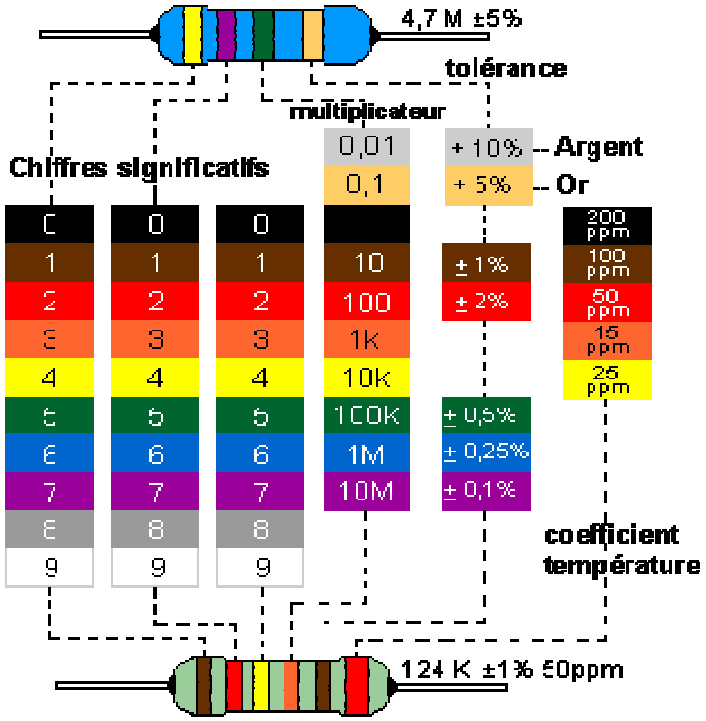
- La tension : Unité = Volts (V)
  - Sur le schéma, on a au point M une tension de 0V et au point P une tension de 5V.
    Le potentiel au point P, soustrait par le potentiel au point M vaut : $U_P - U_M = 5 - 0 = 5V$. On dit que la différence de potentiel entre ces deux points est de 5V



- Le courant : Unité = Ampères (A)
  - Le courant est la quantité d'électrons qui traversent la charge, la puissance dissipée dans la charge sous forme de chaleur sera P = U x I, exprimée en Watts (W).
    (sur le schéma $U = U_{PM} = 5V$)

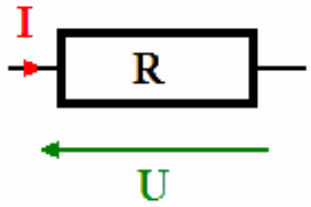
Bases de l'électronique

- La résistance : Unité = Ohms (Ω)
  - Sa principale fonction est de réduire l'intensité du courant (mais pas uniquement).
  - Ce composant se présente sous la forme d'un petit boitier fait de divers matériaux et repéré par des anneaux de couleur indiquant la valeur de la résistance.



Bases de l'électronique

- La loi d'ohms
  - La loi d'ohm est une loi physique très importante dans le domaine électrique. Cette loi met en relation 3 éléments : la valeur d'une résistance (en ohms), le courant qui la traverse (en Ampère) et la tension entre ses bornes (en Volt).



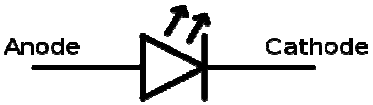
Pour le montage ci-dessus, la formule de la loi d'Ohm est : $U = R \times I$

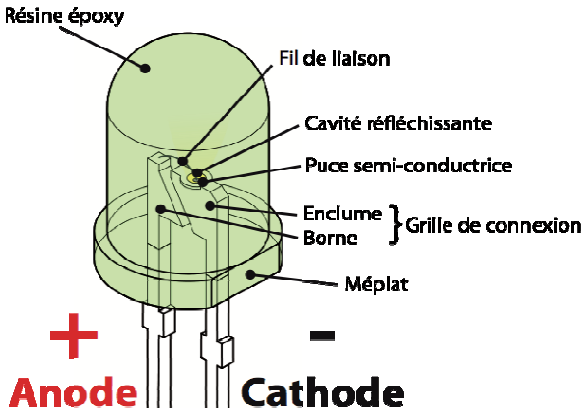
A noter que cette dernière formule peut s'écrire en fonction du courant : $I = \frac{U}{R}$

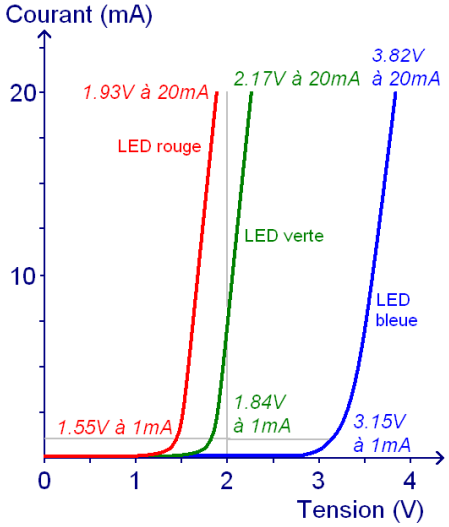
Ou alors on peut l'écrire en fonction de la résistance : $R = \frac{U}{I}$

Bases de l'électronique

- La diode LED
  - Il est possible de remplacer l'ampoule par une diode électroluminescente, aussi appelée LED
  - Le courant électrique ne peut traverser la diode que dans le sens de l'anode vers la cathode.
  - On reconnaît l'anode, car il s'agit de la broche la plus longue. Lorsque les deux broches sont de même longueur, on peut distinguer l'anode de la cathode, par un méplat du côté de la cathode.
  - Le courant maximum que peut supporter une LED est de 20 mA. **Une résistance en série avec la LED sera obligatoire pour ne pas la griller**.

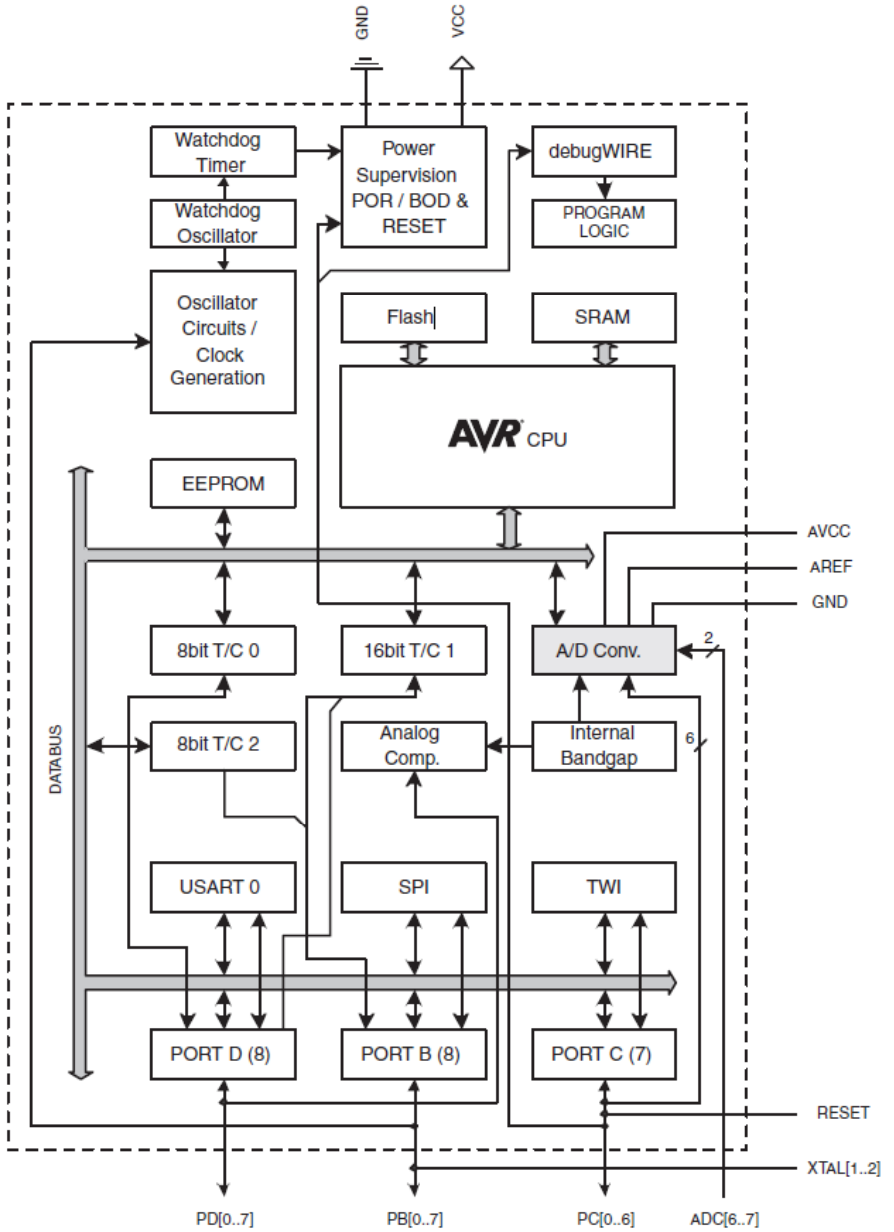






Bases de l'électronique

- Le microcontrôleur ATMega328
  - AVR CPU : processeur qui exécutera le programme
  - Mémoire Flash : contiendra le programme à exécuter
  - Mémoire SRAM : contiendra les variables du programme
  - Timers 8 et 16 bits : pour générer les PWM
  - A/D Conv : pour acquérir des signaux analogiques
  - USART 0 : liaison série pour communiquer avec le PC
  - PORT BCD : pour gérer les entrées/sorties numériques



Structure d'un programme

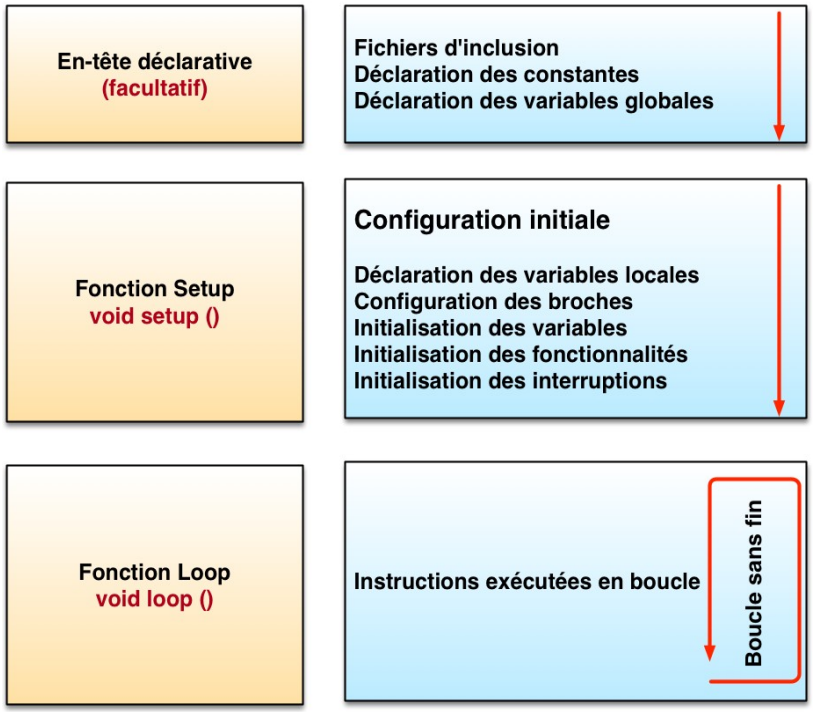
```
sketch_oct09a
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Déroulement du programme

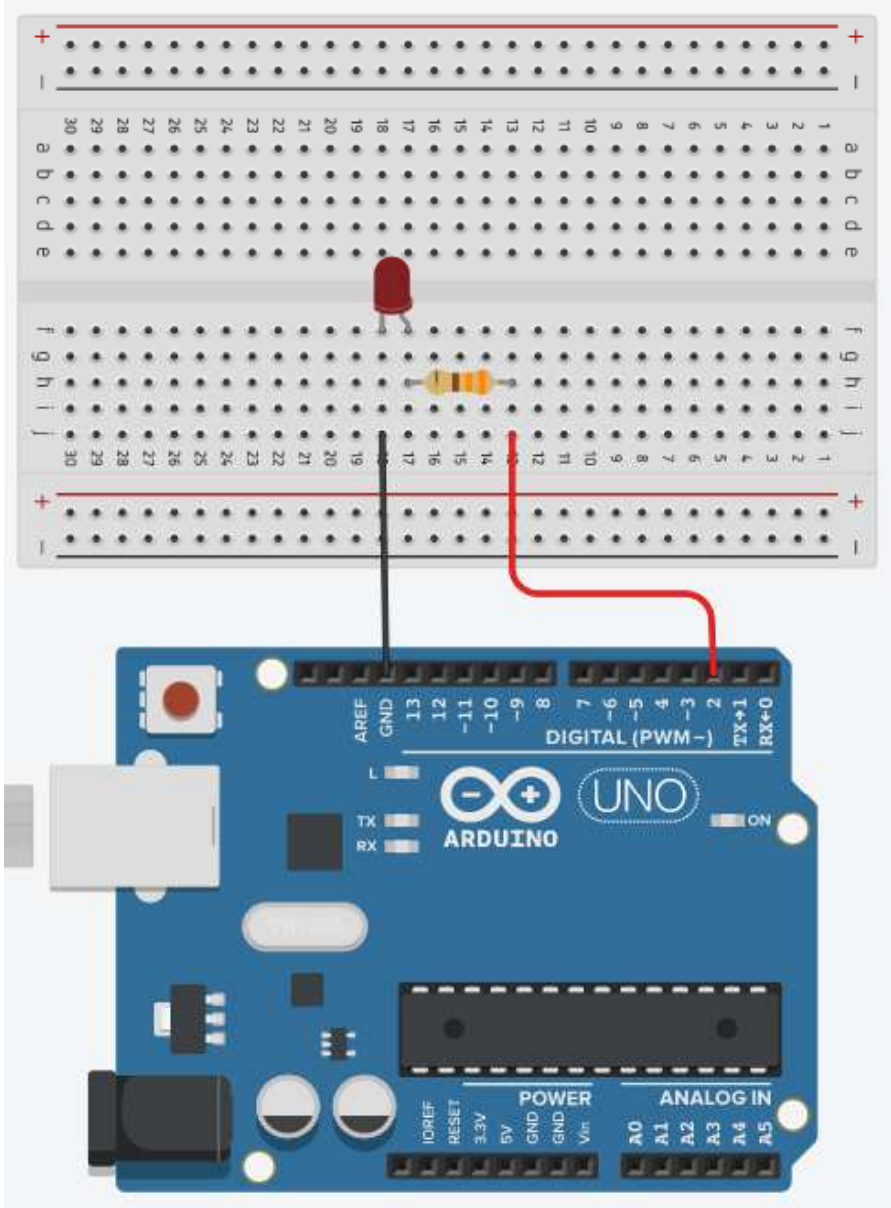


- La fonction setup() est appelée **une seule fois** lorsque le programme commence. C'est pourquoi c'est dans cette fonction que l'on va écrire le code qui n'a besoin d'être exécuté qu'une seule fois (initialisations).
- Dans la fonction loop() on va écrire le contenu du programme. Il faut savoir que cette fonction est appelée en permanence, c'est-à-dire qu'elle est exécutée encore et encore. On parle de *boucle infinie*

Projet 1 : Faire clignoter une LED

- Données d'entrée :
  - LED rouge (tension de seuil ≈ 1.6 V)
  - A connecter sur la pin 2 de la carte (par défaut les pin sont des entrées)
  - Résistance de limitation de courant à calculer (alimentation 5V et courant 10 mA)
  - Fréquence de clignotement ≈ 1 Hz

- Travail à effectuer:
  - Calculer la valeur de la résistance
  - Déclarer la pin 2 en sortie dans la section « Setup » du programme
  - Ecrire le code pour animer la LED dans la section « Loop » du programme



Projet 1 : Faire clignoter une LED

- Fonction : pinMode()
  - **Description**
    - Configure la broche spécifiée pour qu'elle se comporte soit en entrée, soit en sortie.
  - **Syntaxe**
    - pinMode(*broche, mode*);
  - **Paramètres**
    - ***broche***: le numéro de la broche de la carte Arduino dont le mode de fonctionnement (entrée ou sortie) doit être défini.
    - ***mode***: soit INPUT (entrée en anglais) (=0) ou OUTPUT (sortie en anglais) (=1)
  - **Valeur retournée**
    - Aucune

Projet 1 : Faire clignoter une LED

- Fonction : digitalWrite()
  - **Description**
    - Met un niveau logique HIGH (HAUT en anglais) ou LOW (BAS en anglais) sur une broche numérique. Si la broche a été configurée en SORTIE avec l'instruction pinMode(), sa tension est mise à la valeur correspondante : 5V (ou 3.3V sur les cartes Arduino 3.3V) pour le niveau HAUT, 0V (masse) pour le niveau BAS.
  - **Syntaxe**
    - digitalWrite(*broche, valeur*);
  - **Paramètres**
    - ***broche***: le numéro de la broche de la carte Arduino
    - ***valeur*** : HIGH ou LOW (ou bien 1 ou 0)
  - **Valeur retournée**
    - Aucune

Projet 1 : Faire clignoter une LED

- Fonction : delay(ms)
  - **Description**
    - Réalise une pause dans l'exécution du programme pour la durée en millisecondes indiquée en paramètre.
      (Pour mémoire, il y a 1000 millisecondes dans une seconde)
  - **Syntaxe**
    - delay (ms);
  - **Paramètres**
    - ***ms:*** le nombre de millisecondes que dure la pause
  - **Valeur retournée**
    - Aucune

Projet 1 : Faire clignoter une LED

- Code du programme:

```
1  void setup()
2  {
3    pinMode(2, OUTPUT); // Positionne la pin 0 en sortie
4  }
5
6  void loop()
7  {
8    digitalWrite(2, HIGH);    // Met la sortie 0 à 5V (allume la LED)
9    delay(500);               // Attend 500 millisecondes
10   digitalWrite(2, LOW);     // Met la sortie 0 à 0V (eteint la LED)
11   delay(500);               // Attend 500 millisecondes
12 }
```

MARTIGNAS

Merci pour votre attention

Avez-vous des questions ?