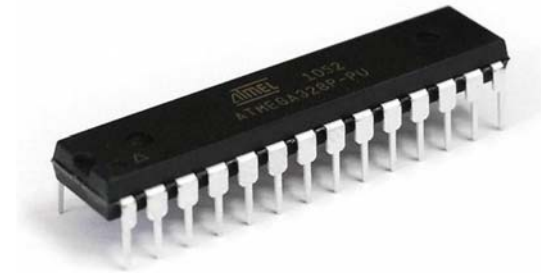


# Les microcontrôleurs

Ce sont des circuits intégrés regroupant plusieurs éléments :

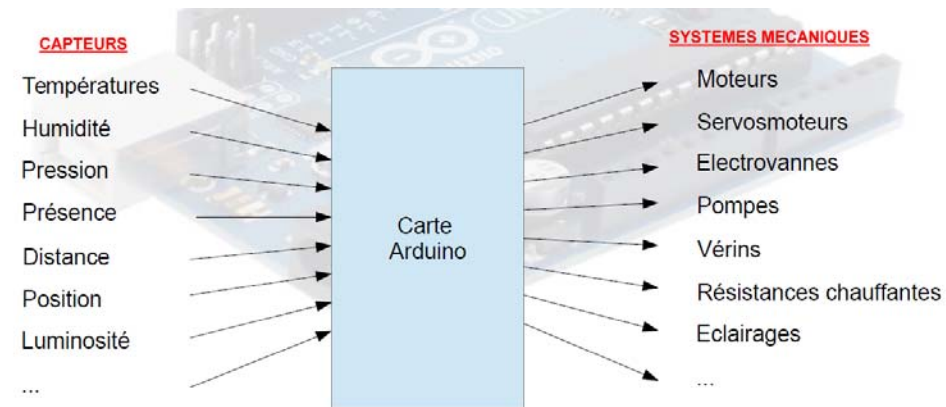
- un microprocesseur,
- de la mémoire de type RAM,
- de la mémoire non volatile pour stocker un microprogramme (firmware),
- des ports de communication de type USB, I2C, Ethernet, Bluetooth, wifi....
- des convertisseurs analogique-numériques (CAN) ...



Les microcontrôleurs sont programmables et éventuellement contrôlables avec un ordinateur.

Ils permettent, comme son nom l'indique, de contrôler des composants électroniques et mécaniques : capteurs, lumières, moteurs, vannes, etc..., afin de créer des systèmes automatisés et régulés, comme par exemple, un système d'arrosage automatique, la régulation de température d'une pièce, etc...

Généralement, dans un système automatisé avec un microcontrôleur, les données d'un capteur (température, humidité, luminosité, etc...) sont transmises au microcontrôleur qui, en fonction des valeurs reçues, donnent l'ordre à un système mécanique d'effectuer une action (allumer ou éteindre une lampe, ouvrir ou fermer une vanne, etc...)



Les microcontrôleurs sont présents dans de nombreuses applications :





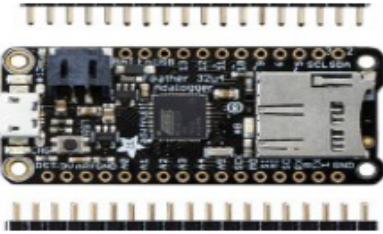
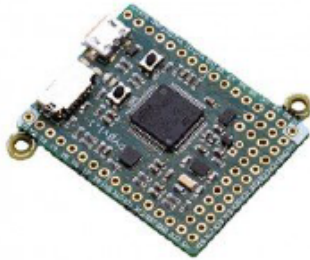
- Électronique embarquée (voiture, avion ...)

- instruments de mesures médicaux, organes artificiels,

- objets de la sécurité : alarme, caméra de surveillance ...

- les appareils électriques du quotidien : machine à laver, hifi ...

Il existe plusieurs types de microcontrôleurs :

<p>Arduino UNO</p> 	<p>BBC Micro:bit</p> 	<p>M5 Stack</p> 
<p>Pycom</p> 	<p>Adafruit Feather</p> 	<p>Pyboard</p> 

Toutes ces cartes ont des avantages et des inconvénients (puissance, langage de programmation, capteurs intégrés ...).

## Les microcontrôleurs dans les lycées

- Les microcontrôleurs dans les lycées sont utilisés comme interface physique sur un ordinateur pour piloter directement des matériels (Buzzer, DELs, ...) et récupérer des données issues de capteurs (température, luminosité, ...), à des fins de traitement et d'exploitations.

- Une nouveauté dans les programmes :

### • Classe de seconde

. Chapitre « Ondes et signaux » :

1) Émission et perception d'un son

- Utiliser un dispositif comportant un microcontrôleur pour produire un signal sonore.

3) Signaux et capteurs :

- Mesurer une grandeur physique à l'aide d'un capteur électrique résistif (**thermistance, photorésistance**).

- Utiliser un dispositif avec microcontrôleur et capteur.

### • Classe de seconde - Option Sciences et Laboratoire

- Systèmes automatisés : aides à la conduite (capteurs, microcontrôleurs et régulateurs)

### • Classe de seconde - Sciences Numériques et Technologie

- Systèmes informatiques embarqués

- Commande d'un actionneur, acquisition des données d'un capteur

### • Classe de Première : Spécialité Sciences Physiques

. Chapitre « Mouvements et interactions » :

2) description d'un fluide au repos

- Tester la loi de Mariotte, par exemple en utilisant un dispositif comportant un microcontrôleur.

. Chapitre « Ondes et signaux

1) ondes mécaniques

- Déterminer, par exemple à l'aide d'un microcontrôleur une distance ou la célérité d'une onde.

### • Classe de Première : Spécialité SVT

Étude de la propagation des ondes à travers des matériaux de nature pétrographique différente ou de comportement mécanique différent (**utilisation des capteurs de vibration, d'humidité, de température, et de lumière**).

### • Classe de Première : enseignement scientifique commun

Réalisation d'un projet expérimental et numérique :

- Utilisation d'un capteur éventuellement réalisé en classe,
- Acquisition numérique de données,
- Traitement, représentation et interprétation de ces données.

## Le choix du microcontrôleur

En considérant la facilité d'utilisation, le prix, l'accessibilité du langage et l'abondance des ressources en ligne, dans la plupart des lycées, le choix du microcontrôleur s'est porté vers la famille des cartes **Arduino**.

Il existe plusieurs types de cartes Arduino :

- **Arduino UNO R3** (la plus connue et la plus utilisée),



- **Arduino Mega 2560** (carte plus grande que l'Arduino UNO, elle comporte plus d'entrée et de sorties ainsi qu'une plus grande puissance de calcul),



- **Arduino Mega ADK** (Cette carte est principalement conçue pour s'interfacer avec des systèmes Android),



- **Arduino Nano** (condensé de la carte Arduino UNO R3),



- **Arduino Mini R5** (plus concentré encore que l'Arduino Nano),



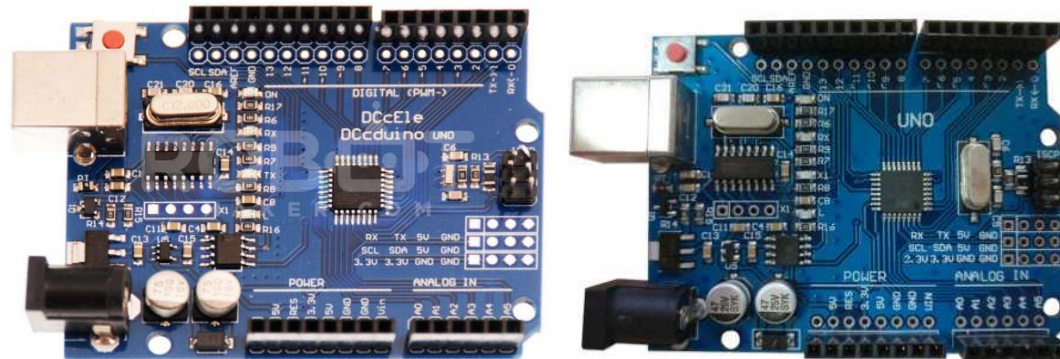
- **Arduino Leonardo** (très semblable à l'Arduino Uno mais équipé d'un processeur **ATMega32U4** qui lui permet ainsi d'être reconnu par l'ordinateur hôte comme un périphérique USB et de simuler les touches d'un clavier ou un déplacement de la souris),



- **Arduino Yun** (Cette carte Arduino possède le même processeur que l'Arduino Leonardo mais possède une différence de taille : un module générant un réseau wifi, et un système linux embarqué).



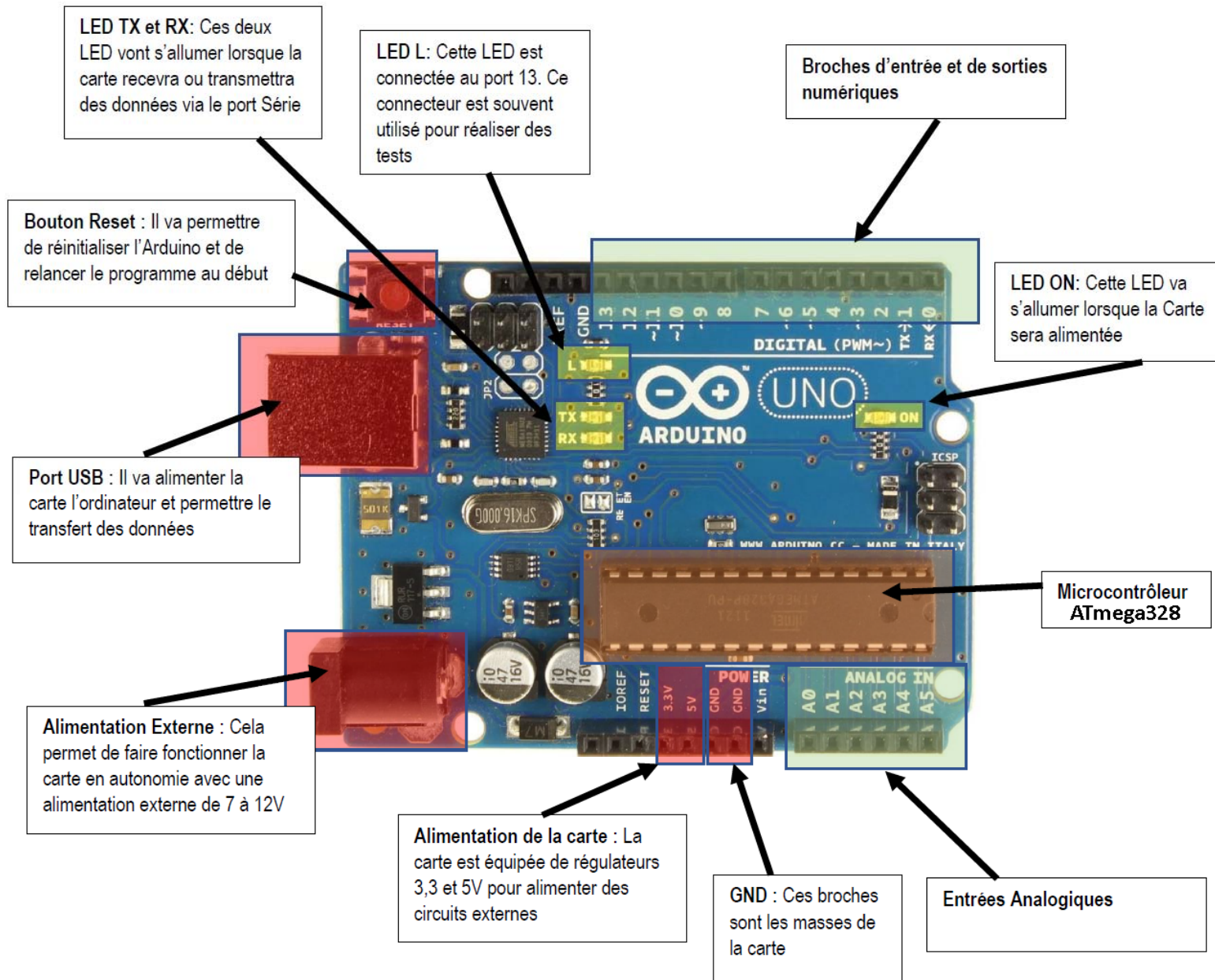
Et comme Arduino est open source, il existe un grand nombre de clones et de platines compatibles :



Nous allons nous intéresser plus particulièrement à la carte **Arduino UNO R3** car c'est celle que vous rencontrerez le plus souvent.



# La carte Arduino UNO R3



## . Le microcontrôleur

La puce ATmega328 (le microprocesseur) est le cerveau de l'Arduino. C'est un microprocesseur de la série AVR produite par la société ATMEL (16 MHz d'horloge et 32 Ko de mémoire). Il va recevoir le programme que nous allons créer et va le stocker dans sa mémoire avant de l'exécuter.

## . Les entrées et sorties numériques

Il y a 14 entrées/sorties numériques notées de 0 à 13 sur l'Arduino Uno. Elles sont situées sur la grande rangée du haut de la carte. Ces connecteurs (ou broches) sont numériques car le signal sur ces broches ne connaît que deux états (niveau logique) : **haut** ou **bas** (**1** ou **0**). Électriquement, cela se traduit, respectivement, par une tension de **5 V** ou **0 V**.



Ces broches sont configurées en entrées ou sorties numériques par programmation :

- . Configurées en sortie, elles ne peuvent délivrer que des niveaux logiques bas (**0 V**) et des niveaux logiques haut (**5 V**).
- . Configurées en entrée, elles ne peuvent recevoir que des niveaux logiques bas (**0 V**) et niveaux logiques haut (**5 V**).

### Attention :

. Les tensions appliquées sur les broches d'entrées/sorties numériques doivent être comprise entre 0 et 5 V. Au-dehors de ces limites, le microcontrôleur sera endommagé.

. Pour une broche configurée en entrée, toute tension inférieure à **0,3 x Vcc**, Vcc étant égale à 5V, soit **1,5 V**, sera comprise comme un niveau logique bas (0 V) et toute tension supérieure à **0,6 x Vcc**, soit **3 V**, sera comprise comme un niveau logique haut (5 V).

Entre les deux, c'est incertain. L'Arduino renverra de toutes façons un 0 ou un 1 mais de manière plus ou moins aléatoire.

. Pour une broche configurée en sortie, il est préférable de limiter l'intensité du courant dans le circuit électrique à **20 mA** et absolument nécessaire de ne pas dépasser **40 mA** sous peine de destruction de la sortie (Il faut donc au minimum une résistance de  $200\ \Omega$  dans le circuit électrique alimenté par la sortie). On veillera aussi à ne pas dépasser une intensité totale de **200 mA** dans les circuits électriques reliés à ces broches.

. Il est déconseillé d'utiliser les broches **0** (RX) et **1** (TX) qui sont initialement prévues pour la communication série (broche 1 pour l'émission et broche 0 pour la réception des données), notamment avec certains modules externes (par exemple, le module Bluetooth HC-05). Mais, en l'absence de communication série, il est possible de les utiliser classiquement en entrée ou sortie numérique.

## . Les entrées analogiques

Il y a six entrées analogiques notées de A0 à A5 en bas à droite de la carte.

Les tensions, entre 0 et 5V, appliquées sur ces broches, sont numérisées via un convertisseur analogique-numérique CAN ou ADC (Analog Digital Converter).



Le convertisseur analogique-numérique des Arduino effectue une conversion sur 10 bits, c'est à dire qu'il convertit la tension analogique en un nombre entier ayant une valeur de 0 à 1023 (**1023** en binaire s'écrit : **1111111111**).

0 correspond à une tension de 0 V et 1023 à une tension de 5V.

La résolution, c'est à dire la différence entre deux valeurs successives de la tension correspondant à une différence de 1 sur l'entier résultat de la conversion analogique-numérique, est donc d'environ 5mV ( $5/1024 = 4,9\ \text{mV}$ ).

### Attention :

. Appliquer une tension supérieure à 5 volts ou inférieure à 0 volt sur une broche analogique endommagera immédiatement et définitivement votre carte Arduino,

. La mesure prend environ  $100\ \mu\text{s}$ , cela fait un maximum de 10 000 mesures par seconde,

. Effectuer une mesure d'une tension sur une broche non connectée retourne des valeurs de l'ordre de 300 à 500, même s'il n'y a pas de signal,

. Les broches notées de A0 à A5 peuvent également être configurées en entrées et sorties numériques.



## . Les sorties analogiques (PWM)

Il n'y a pas de sortie analogique à proprement parler sur un Arduino. Par contre, six des connecteurs numériques (les connecteurs 3, 5, 6, 9, 10 et 11) sont capables de simuler des sorties analogiques et fonctionnent donc comme telles pour délivrer une tension entre 0 et 5 V.



Elles sont marquées du symbole tilde ~ et du sigle PWM, qui veut dire Pulse Width Modulation (Modulation de Largeur d'Impulsion en français).

Le PWM fonctionne ainsi : comme il n'est possible que d'envoyer des informations binaires (haut ou bas, c'est à dire 5 V ou 0 V) sur ces broches, l'Arduino va faire varier la durée pendant laquelle ces deux valeurs sont appliquées afin d'obtenir la tension souhaitée.

L'Arduino génère donc un signal carré caractérisé par deux paramètres :

- . L'amplitude du signal est de 5V ou de 0V,
- . Le rapport entre la durée où la tension est à 5V et celle où elle est à 0V (ce rapport est appelé Duty cycle et est exprimé en %).

La fréquence du signal PWM est de 490 Hz, ce qui est suffisamment rapide pour que l'on puisse dire que l'amplitude du signal d'une sortie PWM est égale à la valeur moyenne du signal carré généré :

$$\text{Tension sortie analogique (en V)} = 5 \times (\text{Duty Cycle}/100)$$

Exemple :

Duty Cycle à 0% : Tension sortie analogique =  $5 \times 0 = 0 \text{ V}$

Duty Cycle à 25% : Tension sortie analogique =  $5 \times 0,25 = 1,25 \text{ V}$

Duty Cycle à 50% : Tension sortie analogique =  $5 \times 0,5 = 2,5 \text{ V}$

Duty Cycle à 75% : Tension sortie analogique =  $5 \times 0,75 = 3,75 \text{ V}$

Duty Cycle à 100% : Tension sortie analogique =  $5 \times 1 = 5 \text{ V}$

