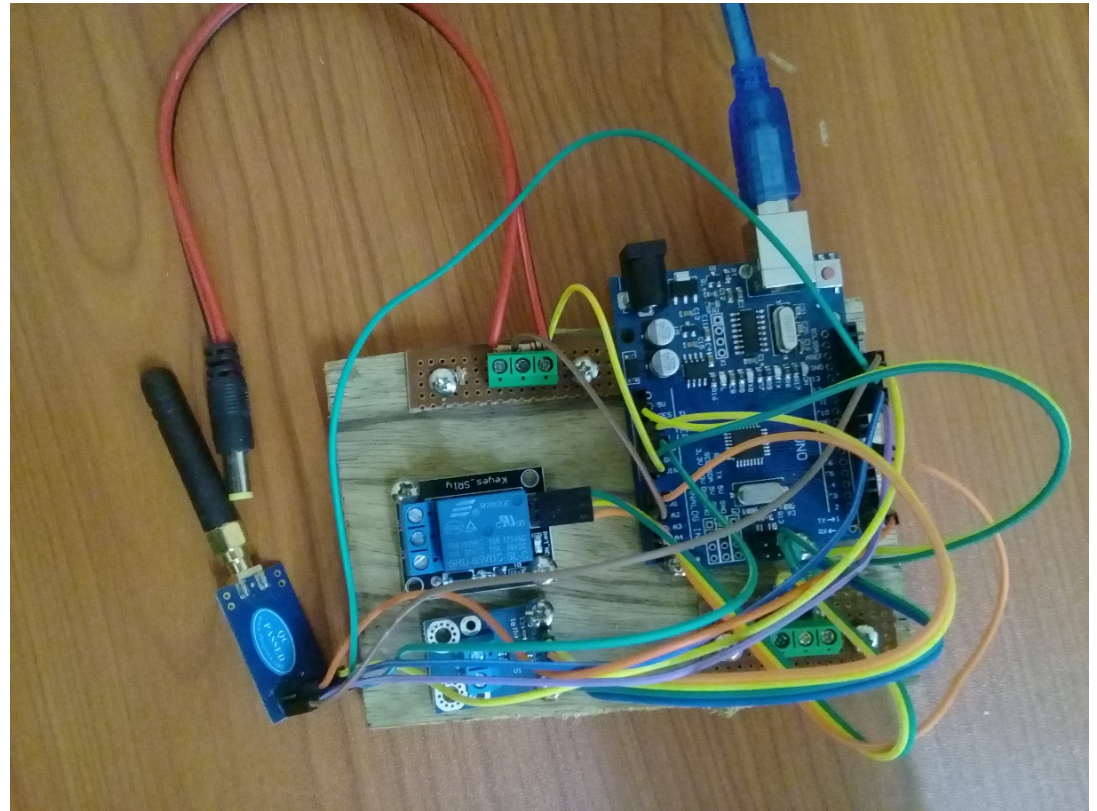


Introduction à L'internet des objets IoT(The Internet Of Things):

Travaux Pratiques N°1

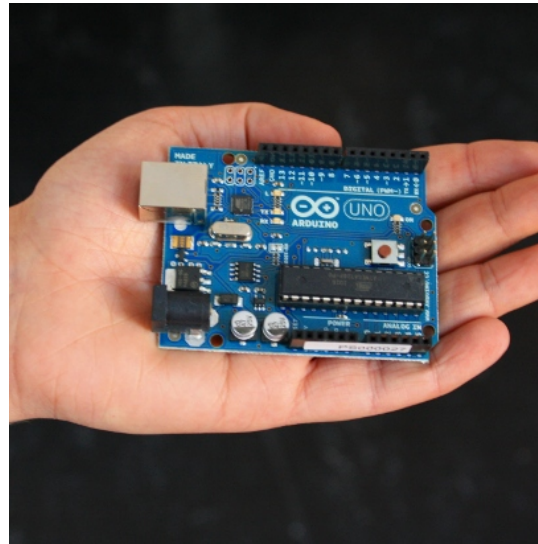
Pr Moussa DIALLO

<http://edmi.ucad.sn/~moussadiallo/>

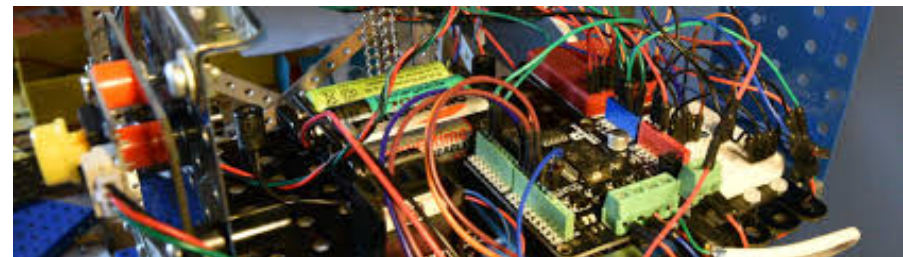
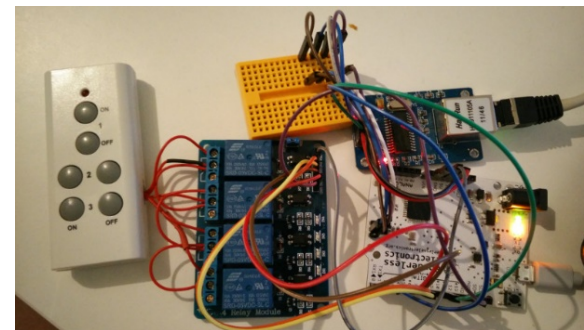
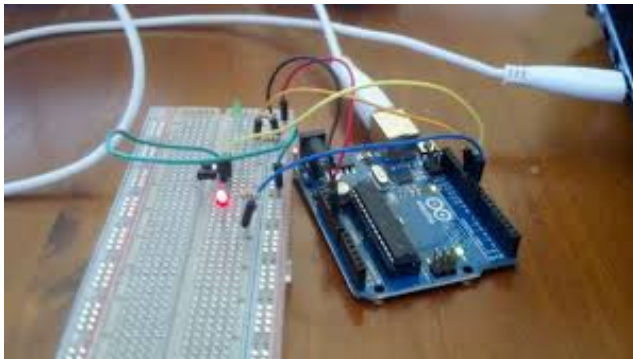


Qu'est ce que c'est Arduino?

Arduino est un projet créé par une équipe de développeurs, composée de six individus : Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti.



Cette équipe a créé le "système Arduino". C'est un outil qui permet aux **débutants, amateurs ou professionnels** de créer des systèmes électroniques plus ou moins complexes.



Que peut-t-on faire avec?

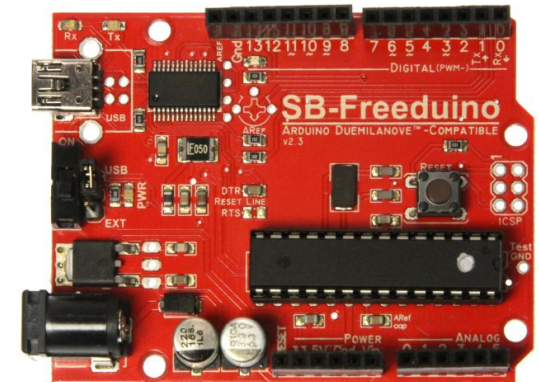
Créer des systèmes électroniques en programmant et en utilisant des composants électroniques

- ❑ Le matériel : cartes électroniques pas cher dont les schémas sont en libre circulation sur internet.
- ❑ Le logiciel : gratuit et open source, développé en Java et avec une simplicité d'utilisation. Compatibilité sous toutes les plateformes, à savoir : Windows, Linux et Mac OS

Les types de cartes

Il y a trois types de cartes :

- ❑ Lesdites « officielles » qui sont fabriquées en Italie par le fabricant officiel (Smart Projects).
- ❑ Lesdits « compatibles » qui ne sont pas fabriqués par Smart Projects, mais qui sont totalement compatibles avec les Arduino officielles.
- ❑ Les « autres » fabriquées par diverse entreprise et commercialisées sous un nom différent (Freeduino, Seeduino, Fentoduino, ...).



Présentation de la carte UNO

C'est la carte la plus couramment utilisée et que nous utilisons dans les TPs de ce cours

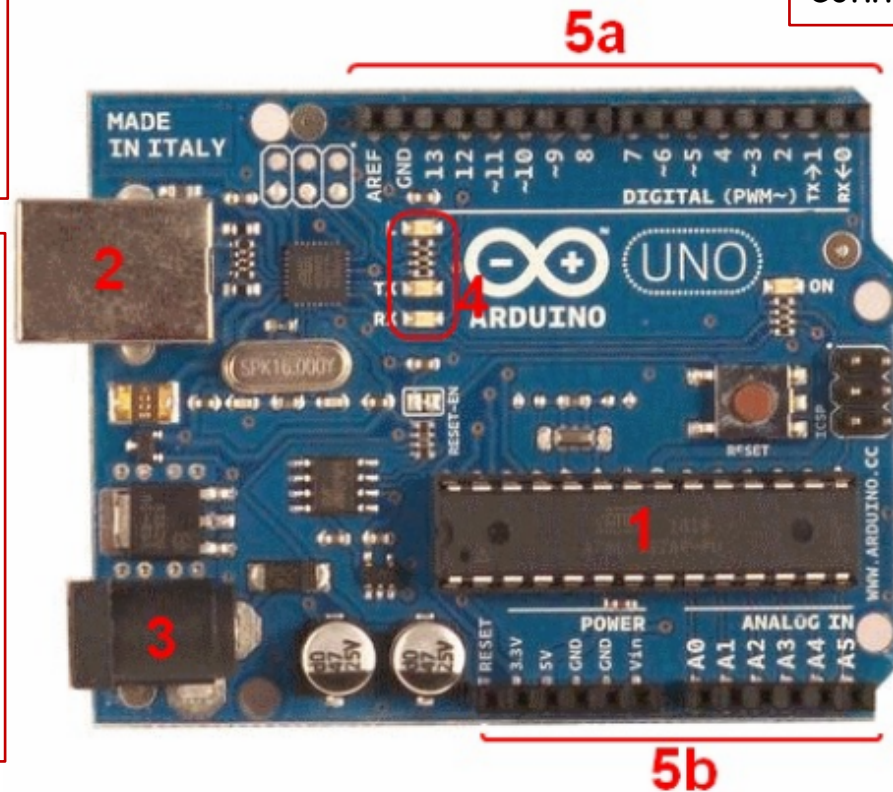
1: Le microcontrôleur est le cerveau de notre carte. C'est lui qui va recevoir le programme que vous aurez créé et qui va le stocker dans sa mémoire puis l'exécuter.

Pour fonctionner, la carte a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB (en **2**) ou bien par une alimentation externe (en **3**) qui est comprise entre 7V et 12V. Cette tension doit être continue et peut par exemple être fournie par une pile 9V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte.

4: 3 LED

- ❑ Celle tout en haut du cadre est connectée à une broche du microcontrôleur et va servir pour tester le matériel. Nota : Quand on branche la carte au PC, elle clignote quelques secondes.
- ❑ Les deux LED du bas servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

5a et 5b: la connectique.

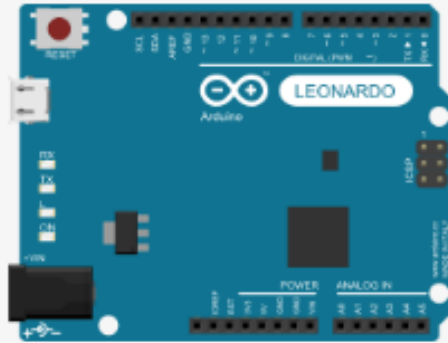


Les autres cartes:

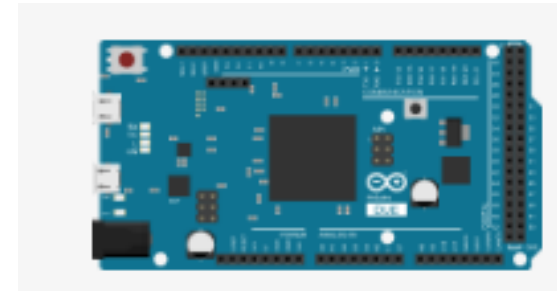
Il existe une bonne centaine de cartes Arduino différentes, certains constructeurs modifient le modèle original afin d'améliorer certains points, d'autre pour une question d'esthétisme ou encore pour l'orienter (expérimentations, meilleur voltage...). Le choix de la carte dépend du projet que l'on veut réaliser.



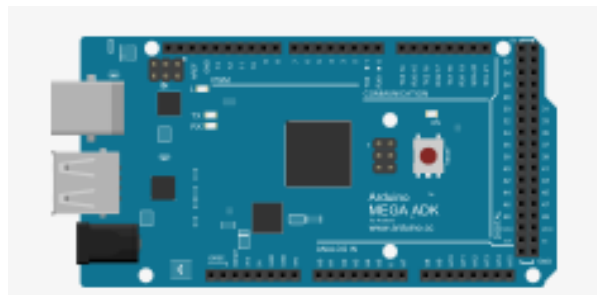
Arduino Mega
Plus complète de UNO



Arduino Leonardo
successeur de UNO



Arduino DUE avec puissance
de calcul plus élevée



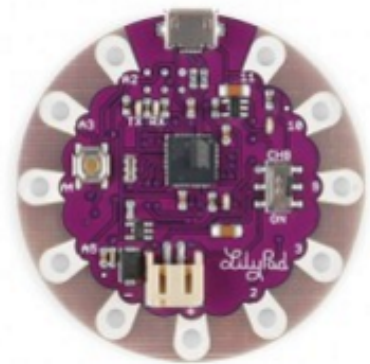
Arduino Mega
Avec USB



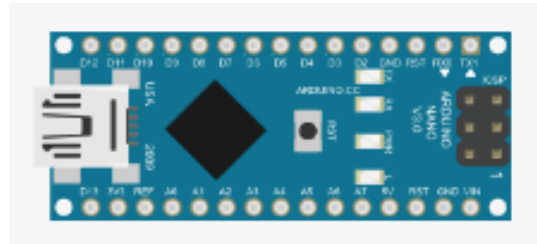
Arduino yun
Linux et Wifi

Les autres cartes

Des cartes de plus petites tailles et de formes différentes



Arduino lilypad USB



Arduino Nano
Pour des projet en
miniature



Arduino mini
Pour finaliser un projet



Arduino esplora

Installation

Arduino est un Espace de Développement Intégré (EDI) qui vous permet d'écrire, de compiler et d'envoyer du code sur le circuit imprimé du même nom.

Télécharger et installer l'IDE d'Arduino sur votre machine Windows, Linux ou Mac.

Installation sous Linux

❑ Methode 1:

(il faut avoir une connexion internet)

Exécuter en mode super utilisateur la commande :

❑ Debian/Ubuntu:

`apt-get install arduino arduino-core`

❑ Fedora:

`dnf install arduino arduino-core`

❑ Red Hat:

`yum install arduino arduino-core`

❑ Methode 2:

❑ Décompresser le dossier téléchargé

❑ Installer arduino

❑ Pour les permissions

`sudo usermod -a -G dialout <username>`

`sudo chmod a+rw /dev/ttyUSB0`

Pour chercher les pilotes sous Windows

-Brancher la carte sur un des ports USB de votre PC

-Clic droit sur Ce PC

-Propriétés

-gestionnaire de périphériques

-USB inconnue puis chercher les pilotes (il faut avoir une connexion internet)

Sinon installer directement les pilotes (CH34x_Install_Windows_v3_4.exe)

Fonctionnement global

Nous allons étudier le fonctionnement global à partir de l'exemple blink

```
const int lampe_pin=12;
```

```
void setup() {  
    pinMode(lampe_pin, OUTPUT); //initialises de lampe_pin comme sortie  
}
```

```
void loop() {  
    digitalWrite(lampe_pin, HIGH);    // Allumes la led  
    delay(1000);                      // patientes une seconde  
    digitalWrite(lampe_pin, LOW);    // Eteins la led  
    delay(1000); // patientes une seconde  
}
```

Voir fichier [exemple_blink.ino](#) en annexe

Fonctionnement global

La première zone

La première zone sert principalement à dire à la carte de garder en mémoire quelques informations: déclarations.

On peut par exemple déclarer:

- ☐ l'emplacement d'un élément connecté à la carte.
- ☐ une valeur quelconque qui sera utile dans le programme.
- ☐ Ect...

Selon votre version définissez une variable `lampe_pin` et affectez le PIN N°12 de la carte

```
const int lampe_pin=12;
```

Fonctionnement global

La deuxième zone

La zone secondaire est l'endroit où l'on va initialiser certains paramètres du programme.

- ☐ Définir une liaison série pour communiquer avec l'ordinateur
- ☐ Configurer un PIN en entrée ou en sortie
- ☐ ect...

Selon votre version configurez le PIN N°12 en sortie

```
void setup() {  
    pinMode(lampe_pin, OUTPUT); //initialises de lampe_pin comme sortie  
}
```

L'exécution se fera à chaque fois que la carte est mise sous tension ou après un RESET

Fonctionnement global

La dernière zone

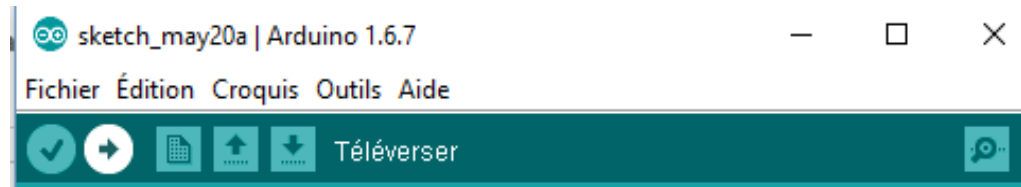
La dernière zone est la zone principale où se déroulera le programme. Tout ce qui va être écrit dans cette zone sera exécuté par la carte, se sont les actions que la carte fera.

```
void loop() {  
    digitalWrite(lampe_pin, HIGH);    // Allumes la led  
    delay(1000);                      // patientes une seconde  
    digitalWrite(lampe_pin, LOW);    // Eteins la led  
    delay(1000); // patientes une seconde  
}
```

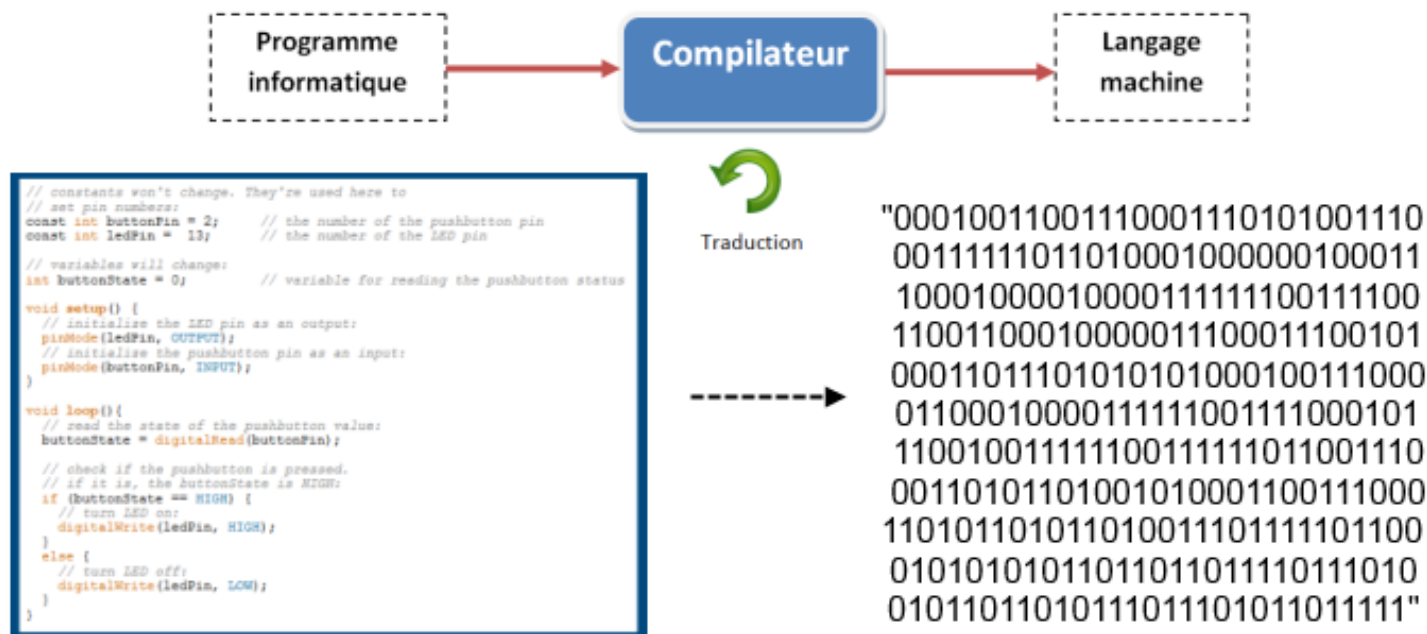
Tant que la carte est sous alimentation, toutes les actions dans la fonction loop sont continuellement exécutées.

Envoi du programme vers la carte: Téléverser

Pour envoyer le programme sur la carte.



Le logiciel Arduino va alors vérifier si le programme ne contient pas d'erreur et ensuite le compiler (le traduire) pour l'envoyer dans la carte.



Si tout fonctionne bien, on a le message televersement terminé.

Le langage de programmation: c++

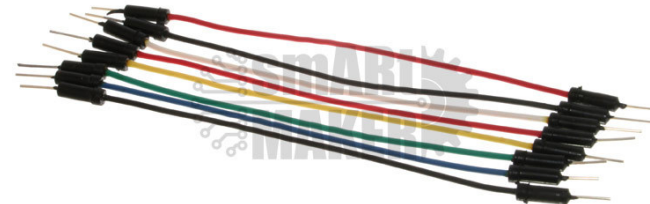
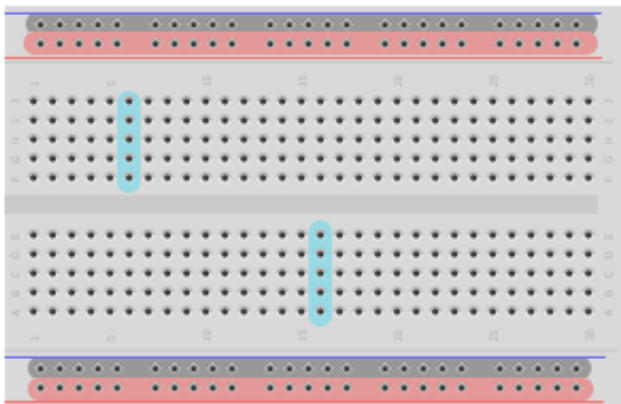
Lire les deux tutoriels suivants

<http://eskimon.fr/83-arduino-105-le-langage-arduino-12>

<http://eskimon.fr/86-arduino-106-le-langage-arduino-22>

Réalisation du câblage:

Le matériel



Quelques exercices

- ☐ Changer le programme et le câblage pour allumer et éteindre la Led à partir du PIN 2.

- ☐ Changer le programme pour que la Led s'allume pendant 3 secondes et ensuite s'éteint 1,5 secondes.

- ☐ Réaliser un feu de signalisation routière.
 - ☐ Rouge: 50 secondes
 - ☐ Vert: 1minute 10 secondes
 - ☐ Orange: 5 secondes

Voir fichier `exemple_feurouge.ino` en annexe

Quelques exercices

La fonction `millis()`

A l'intérieur du cœur de la carte Arduino se trouve un chronomètre. Ce chrono mesure l'écoulement du temps depuis le lancement de l'application. `millis()` nous sert à savoir quelle est la valeur courante de ce compteur.

- ❑ Changer le programme en utilisant `millis()`.

```
//on initialise le temps  
temps = millis();
```

```
if((millis() - temps) > 1000)  
{  
    actions  
}
```



- ❑ Refaire la fonction `blink` avec `millis()`.

Voir fichier `exemple_blink_avecmillis.ino` en annexe

Communications filaire sur port série



- ❑ Pour établir une communication entre le PC et une carte arduino:

```
void setup() {  
    Serial.begin(9600);  
}
```

Il y a 8 débits possibles

- ❑ 1200 Bauds
- ❑ 2400 Bauds
- ❑ 4800 Bauds
- ❑ 9600 Bauds (par défaut)
- ❑ 19200 Bauds
- ❑ 38400 Bauds
- ❑ 57600 Bauds
- ❑ 115200 Bauds

- ❑ La peut tout lire (dans la **loop**) dans une variable string ou caractère par caractère:

```
if(Serial.available() > 0) {  
    String message = Serial.readString();  
}
```

```
if(Serial.available() > 0) {  
    Char message = Serial.read();  
}
```

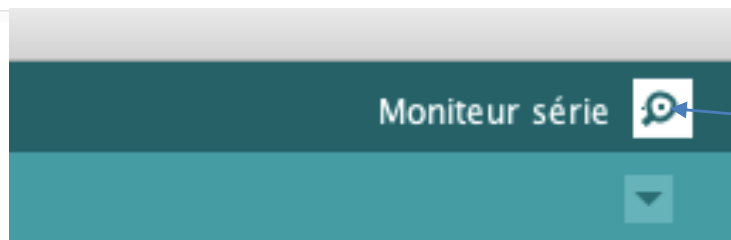
- ❑ Pour envoyer de la carte vers le PC:

```
Serial.print("texte ou donnees");
```

Exemple

Ce programme permet de transmettre via le port série un message à la carte. A la réception du message, la carte répond en envoyant: message bien reçu suivi du message reçu.

```
void setup() {  
  Serial.begin(9600); // ouverture du port série avec un débit de 9600 Bauds  
}  
  
void loop() {  
  
  if (Serial.available() > 0) // s'il y a des caractères sur le port série  
  {  
    String message = Serial.readString(); // lire le message en entier  
    Serial.print("Message reçu: "); // affichage sans aller à la ligne  
    Serial.println(message); // affichage avec saut de ligne  
  }  
}
```



Ouverture du port série

Voir fichier **exemple_comserie.ino** en annexe

Quelques exercices

- ❑ Ecrire un programme qui permet
 - ❑ D'allumer la led en envoyant "on" à la carte
 - ❑ D'éteindre la led en envoyant "off" à la carte

Voir fichier `exemple_blink_commande.ino` en annexe



- ❑ Améliorer d'exercice précédent de sorte que, la carte envoie vers le PC le message « saisie incorrecte » s'il reçoit autre chose que « on » ou « off ».

Voir fichier `exemple_comserie_filtre.ino` en annexe