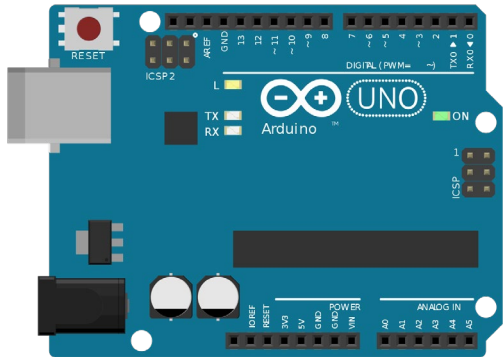


Initiation à la programmation

Avec Arduino !

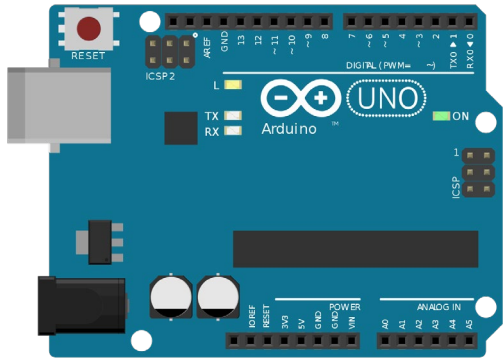
Arduino, Késako ?



Une carte

Une carte

Arduino, Késako ?



Une carte

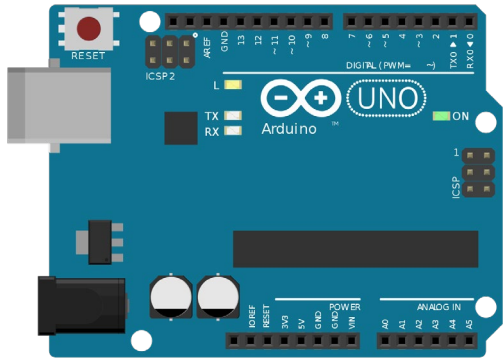


```
01010101100110001100
10100110010001010011
00101010100101101011
01010101010101010101
01010101010101001110
11010110000010111111
00110010101111010101
01010011011001010011
01101111000111000111
00110011110011010011
```

Un programme

Un programme

Arduino, Késako ?

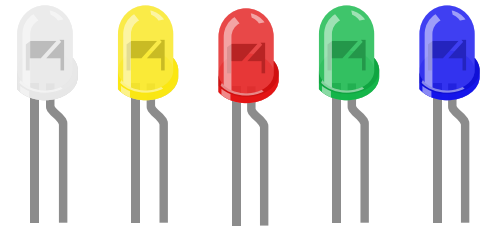


Une carte



```
01010101100110001100
10100110010001010011
00101010100101101011
01010101010101010101
01010101010101001110
11010110000010111111
00110010101111010101
01010011011001010011
01101111000111000111
00110011110011010011
```

Un programme



Du matériel

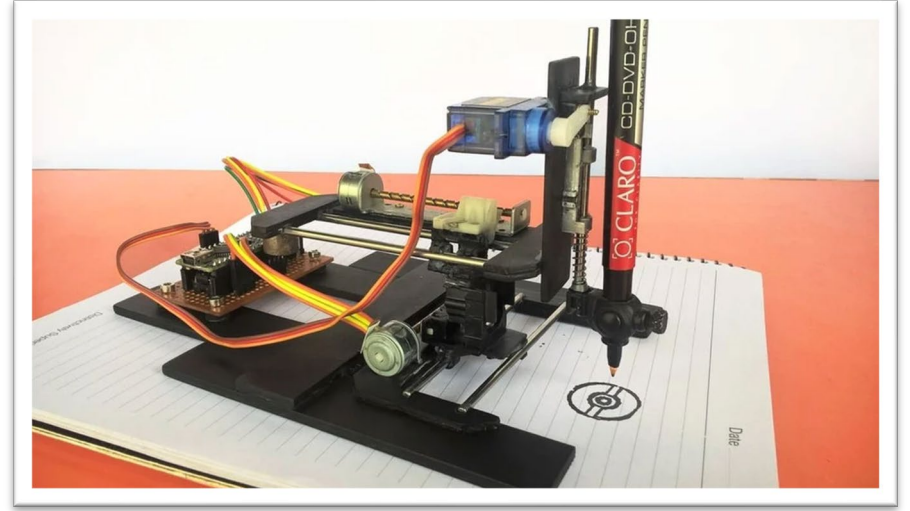
Du matériel



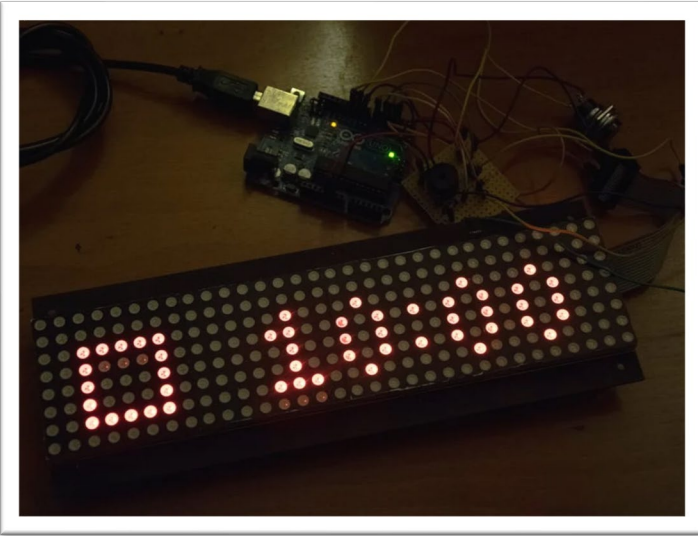
Une main robotique



Une serre intelligente



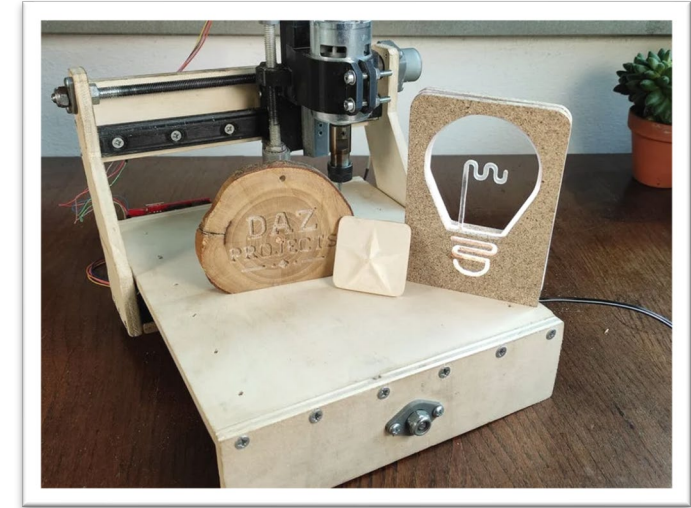
Robot dessinateur



Pendule d'échecs



Un moniteur
cardiaque

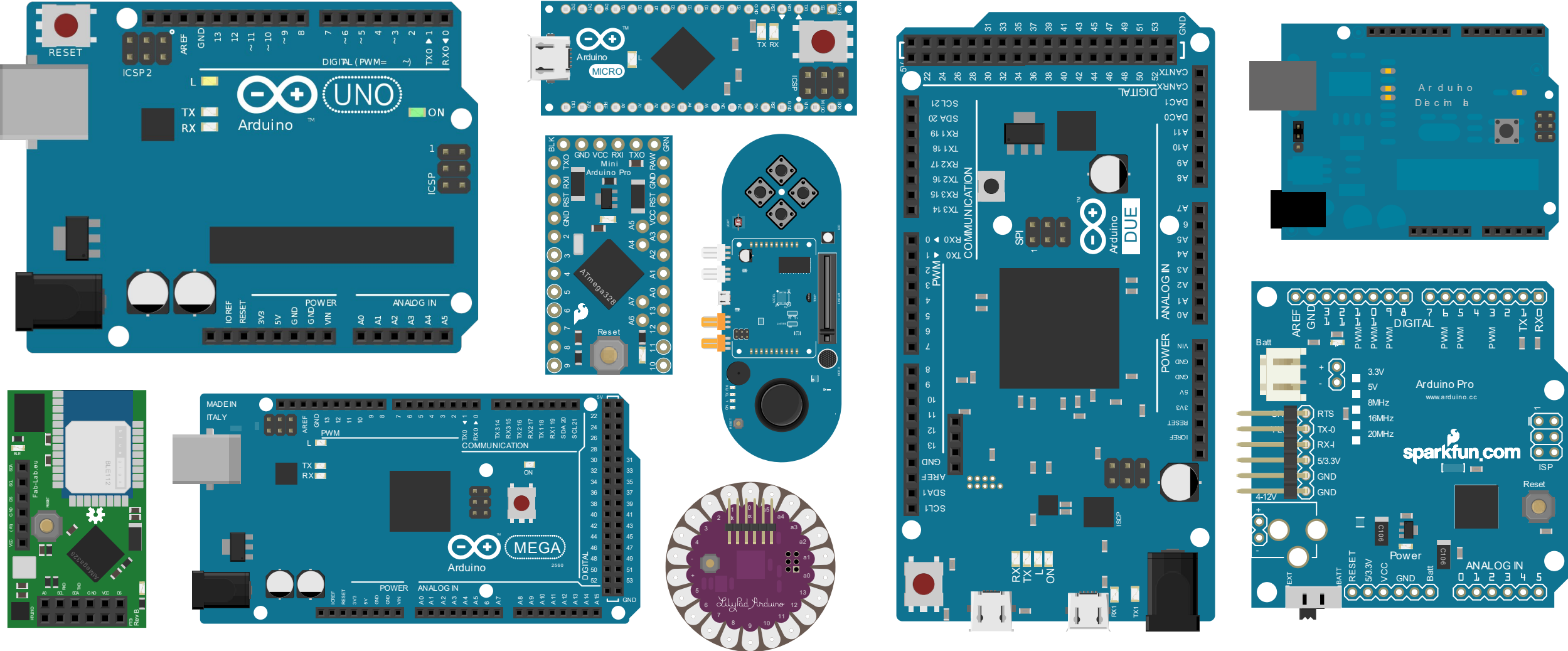


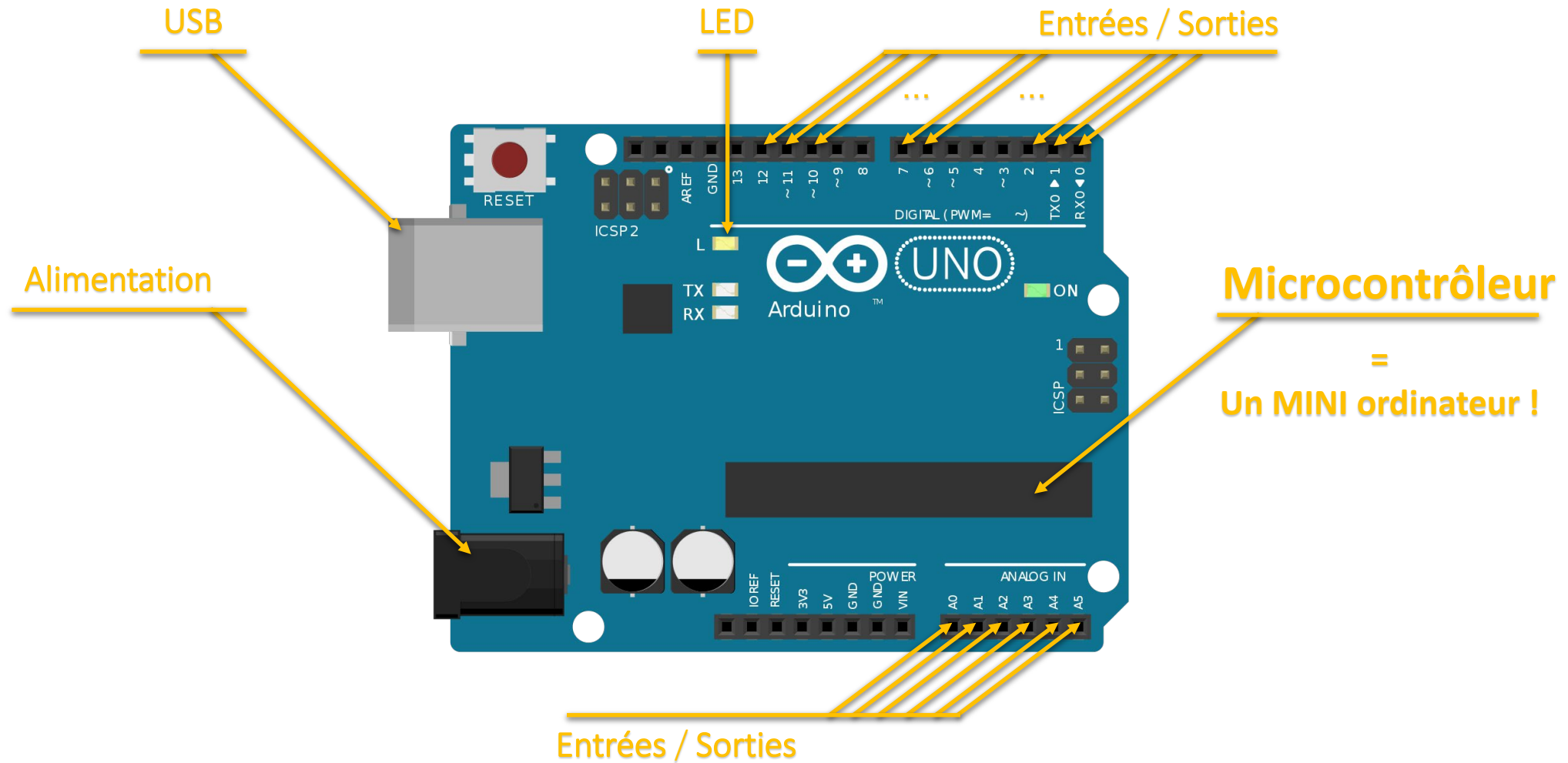
Fraiseuse

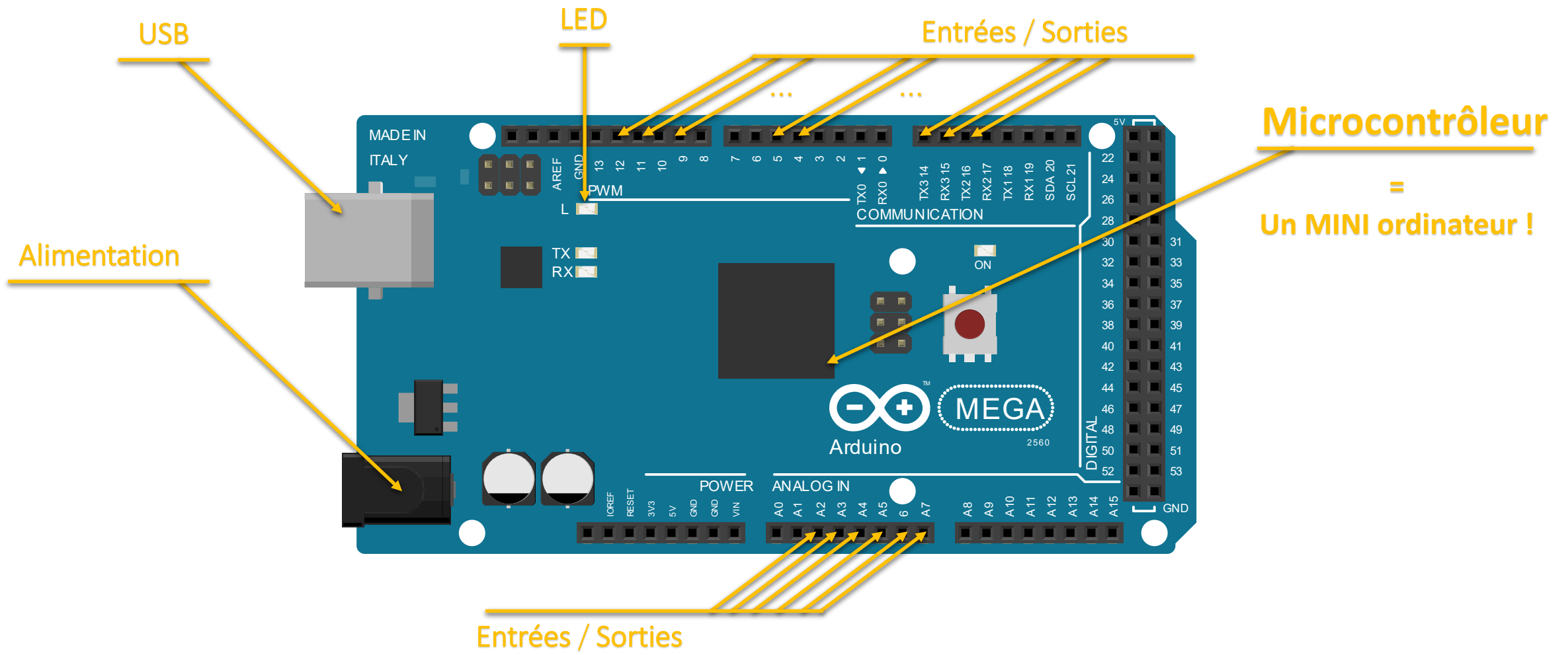
Le matériel

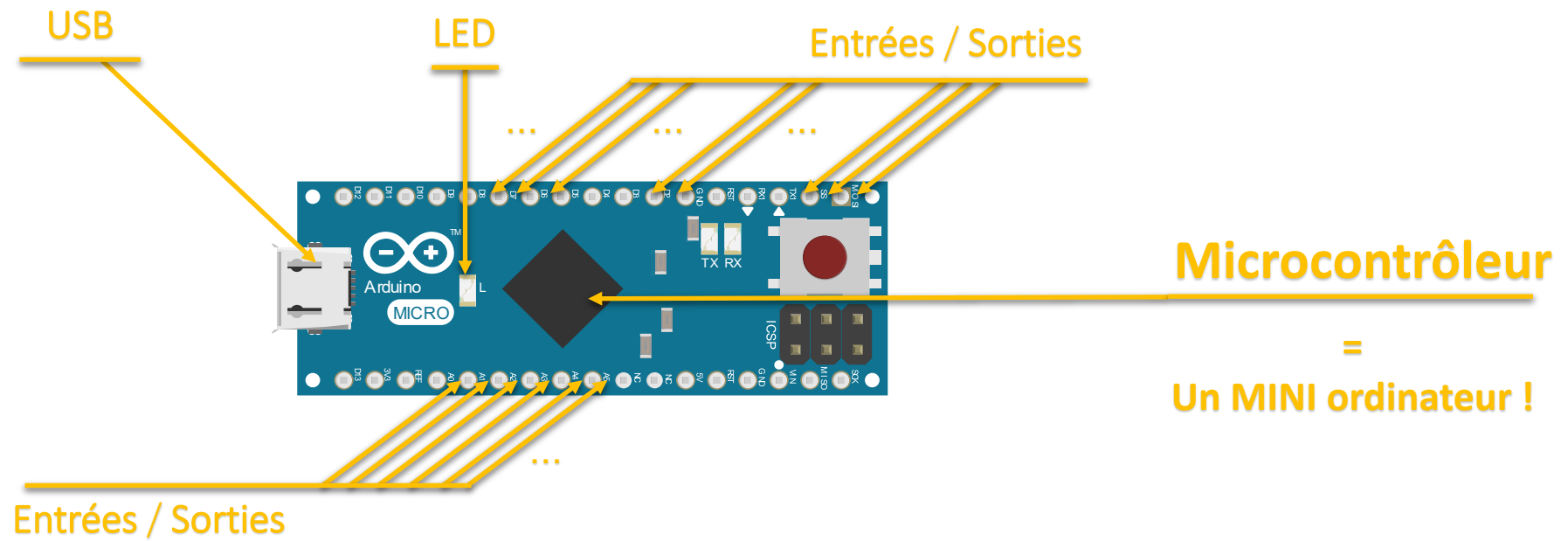
L'univers Arduino

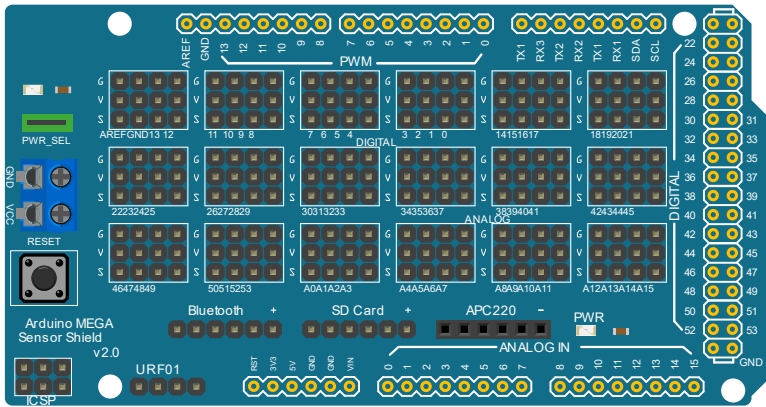
Des Cartes



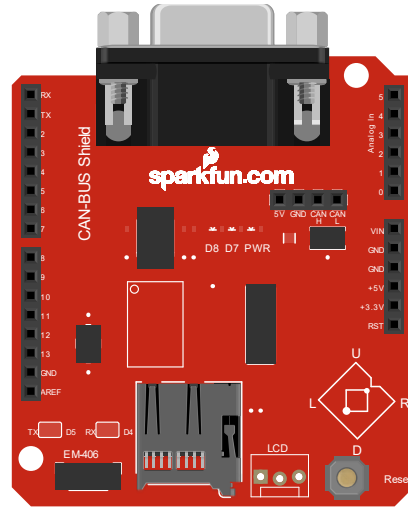




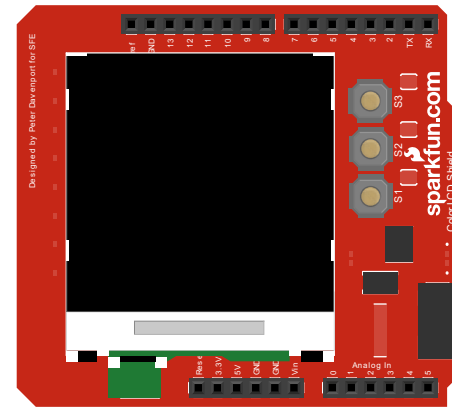




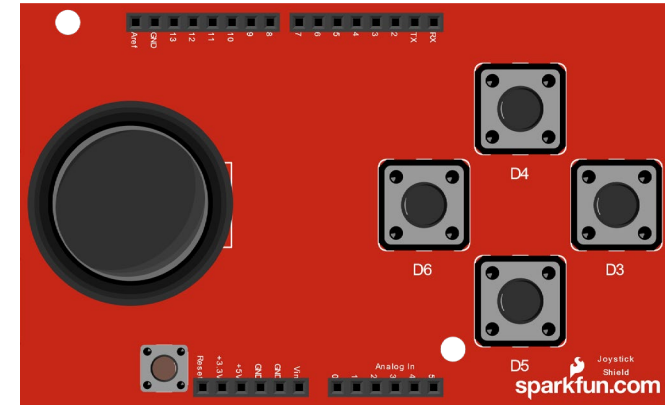
Shield capteur



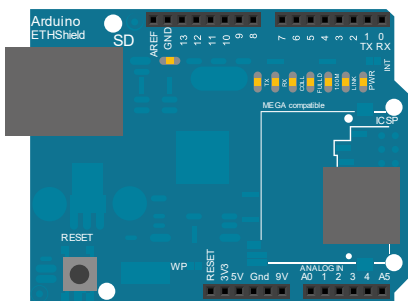
Shield communication



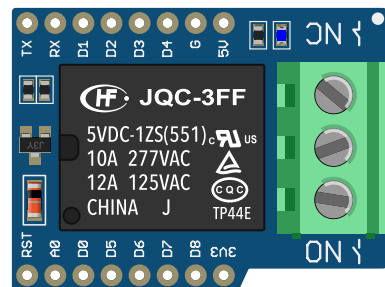
Shield écran



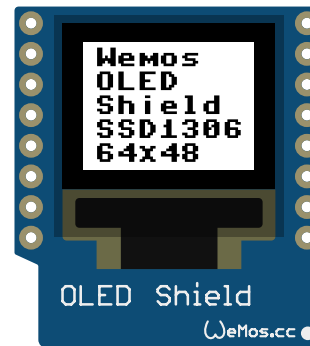
Shield manette



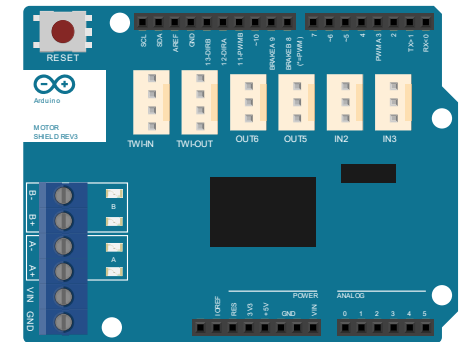
Shield Ethernet



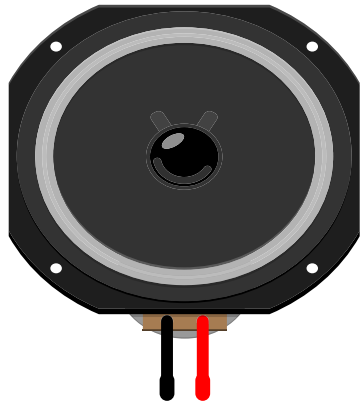
Shield relais



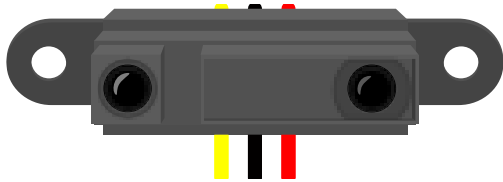
Shield écran



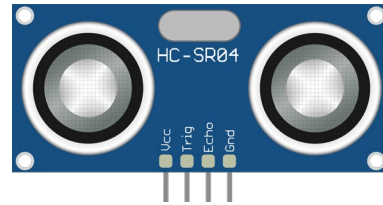
Shield moteur



Enceintes



Capteur de distance infrarouge



Capteur de distance ultrasons

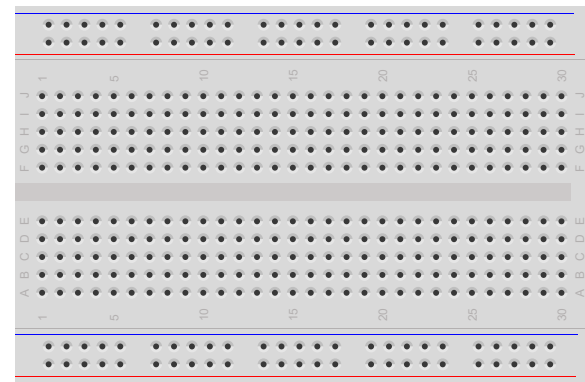
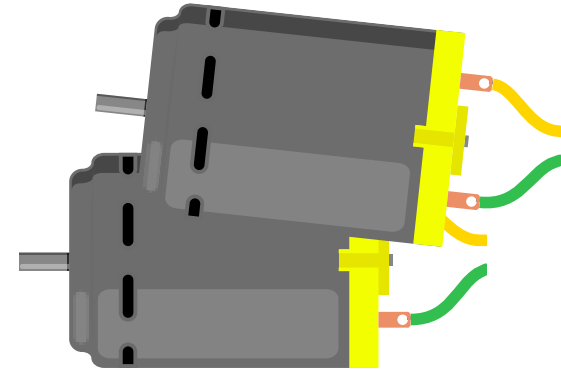
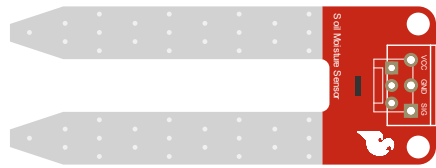


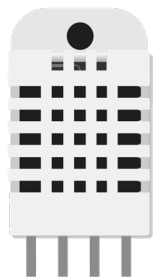
Planche à pain



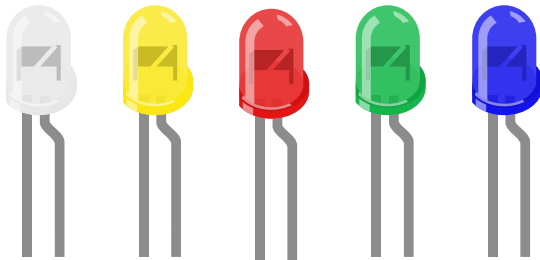
Des moteurs



Capteur d'humidité

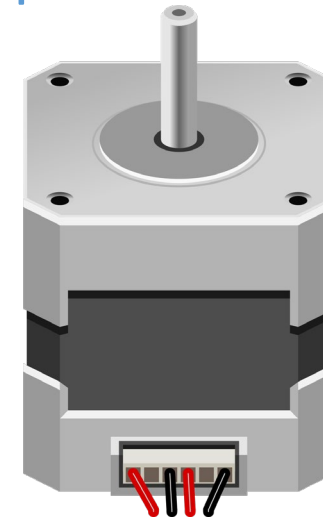
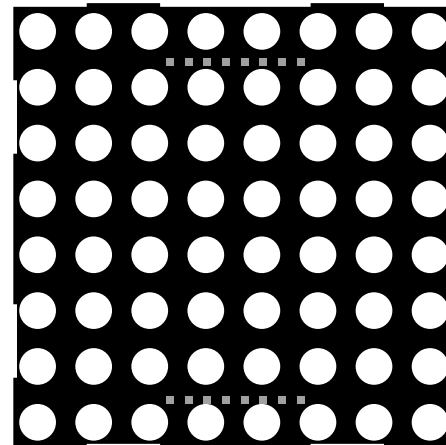


Capteur de température



LED

Afficheurs

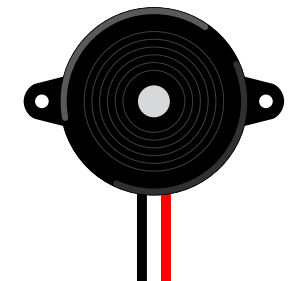


Des moteurs



Résistance

Piezzo



Un Programme

Arduino, Késako ?



Un ordinateur

Créer un programme pour Arduino

Arduino, Késako ?



Un ordinateur



Logiciel Arduino

Créer un programme pour Arduino

Arduino, Késako ?



Un ordinateur



Logiciel Arduino



```
01010101100110001100
10100110010001010011
00101010100101101011
01010101010101010101
01010101010101001110
11010110000010111111
00110010101111010101
01010011011001010011
01101111000111000111
00110011110011010011
```

Un programme

Créer un programme pour Arduino

Un Programme

Une recette = une suite d'instructions

- 1 Laissez ramollir le beurre à température ambiante. Dans un saladier, malaxez-le avec le sucre.
 - 2 Ajoutez l'œuf et éventuellement le sucre vanillé.
 - 3 Versez progressivement la farine, la levure chimique, le sel et les pépites de chocolat. Mélangez bien.
 - 4 Beurrez une plaque allant au four ou recouvrez-la d'une plaque de silicone. À l'aide de deux cuillères à soupe ou simplement avec les mains, formez des noix de pâte en les espaçant car elles s'étaleront à la cuisson.
- 👉 Pour finir**
Faites cuire 8 à 10 minutes à 180°C soit thermostat 6. Il faut les sortir dès que les contours commencent à brunir.



Un Algorithme

Un Programme



Recette en français



Recette en C++



Recette en « 0 » et « 1 »

1 Laissez ramollir le beurre à température ambiante. Dans un saladier, malaxez-le avec le sucre.

2 Ajoutez l'oeuf et éventuellement le sucre vanillé.

3 Versez progressivement la farine, la levure chimique, le sel et les pépites de chocolat. Mélangez bien.

4 Beurrez une plaque allant au four ou recouvrez-la d'une plaque de silicone. À l'aide de deux cuillères à soupe ou simplement avec les mains, formez des noix de pâte en les espaçant car elles s'étaleront à la cuisson.

ψ **Pour finir**
Faites cuire 8 à 10 minutes à 180°C soit thermostat 6. Il faut les sortir dès que les contours commencent à brunir.

```
void loop()
{
    addFlour(375);
    addSoda(0, 5);
    addEggs(3);
    addSugar(1000);

    while(!Mixed)
        Mix();
    bakeCookies();
}
```

```
01010101100110001100
10100110010001010011
00101010100101101011
01010101010101010101
01010101010101001110
11010110000010111111
00110010101111010101
01010011011001010011
01101111000111000111
00110011110011010011
```

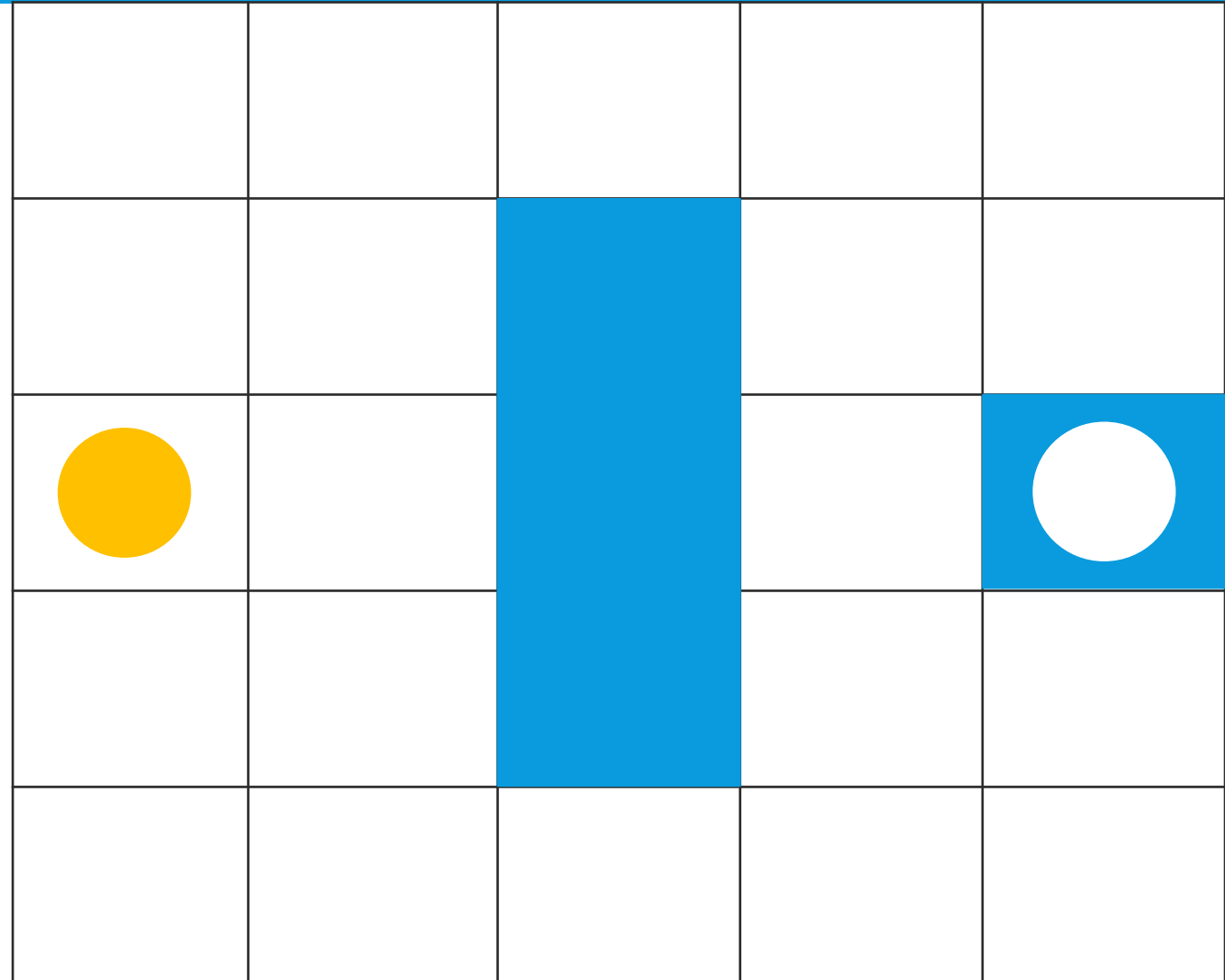

Un Programme

Exercice 1

Instructions disponibles :

```
North(); // Monter d'une case  
South(); // Descendre d'une case  
East(); // Aller à droite d'une case  
West(); // Aller à gauche d'une case
```

```
Programme()  
{  
  East();  
  North();  
  North();  
  East();  
  East();  
  East();  
  South();  
  South();  
}
```



Le logiciel Arduino

Le logiciel Arduino

 <https://www.arduino.cc/>



Vérifier

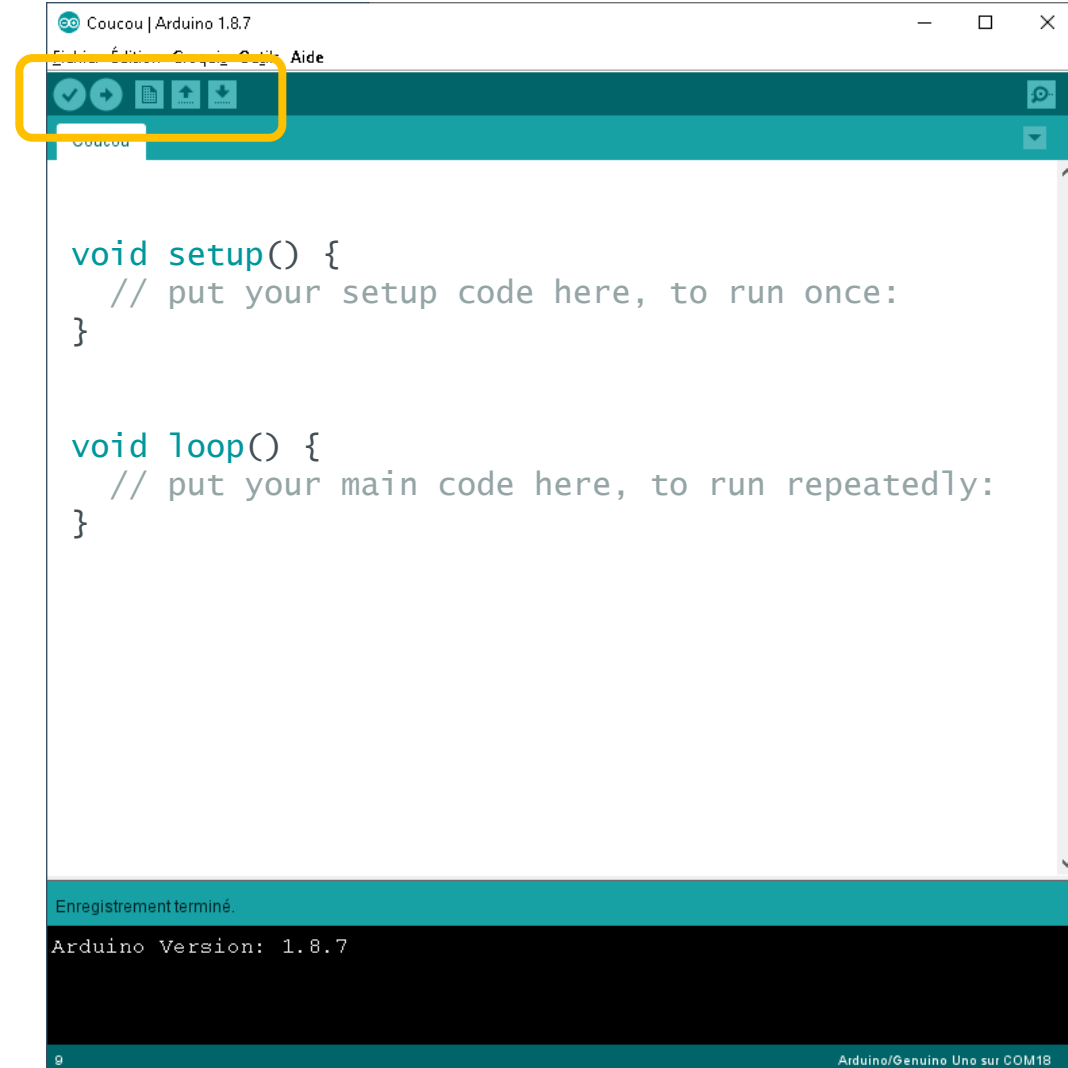
Téléverser

Nouveau

Ouvrir

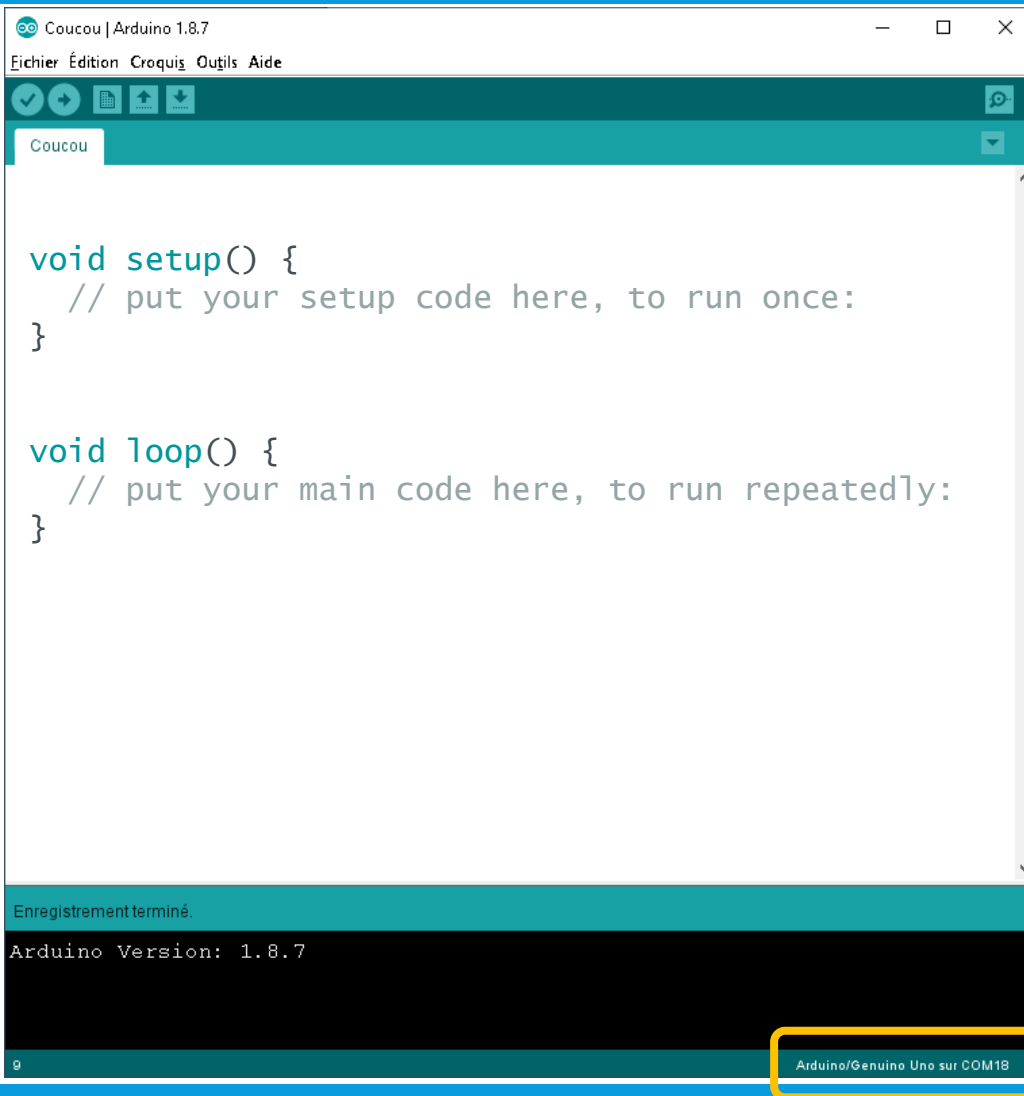
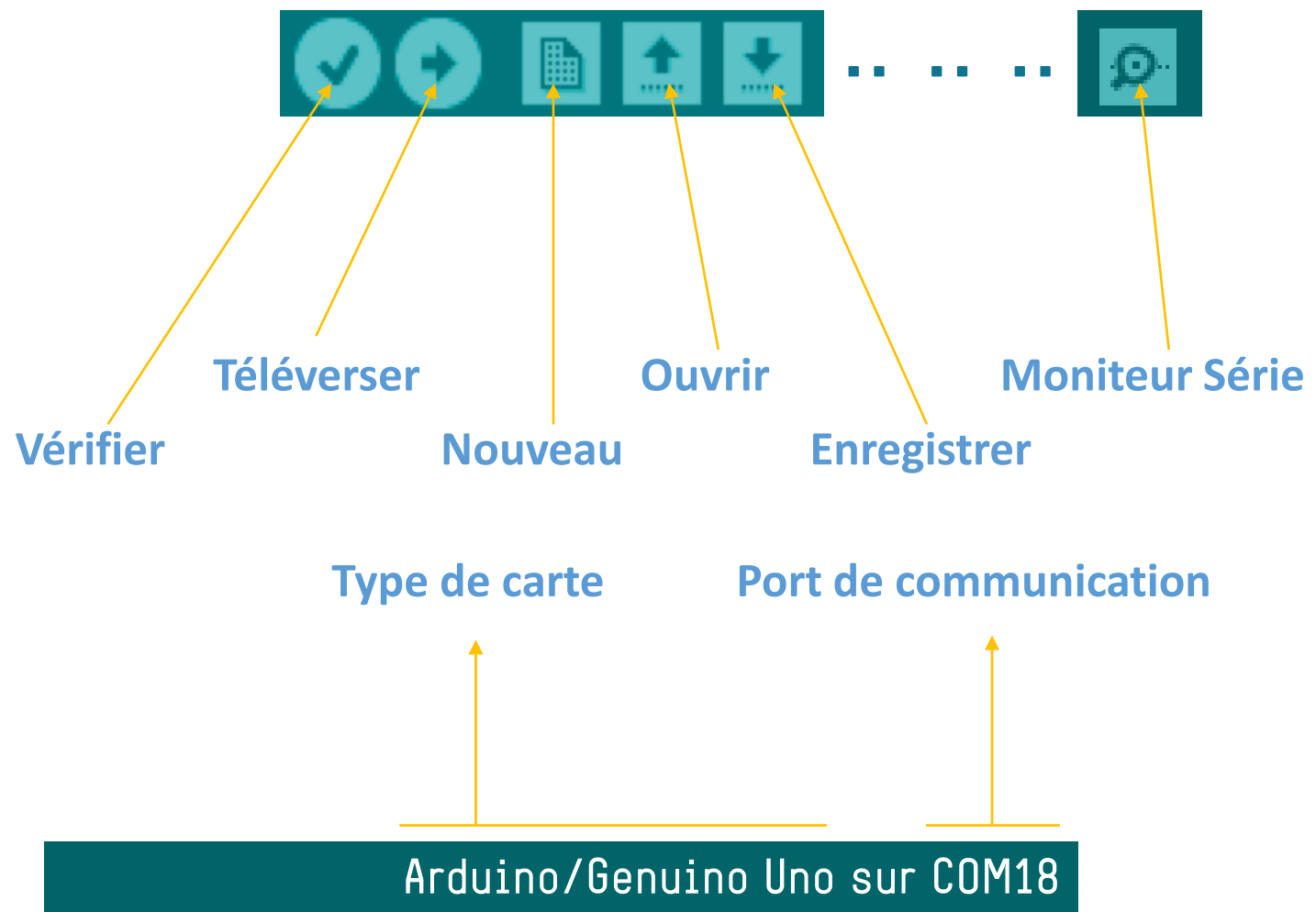
Enregistrer

Moniteur Série



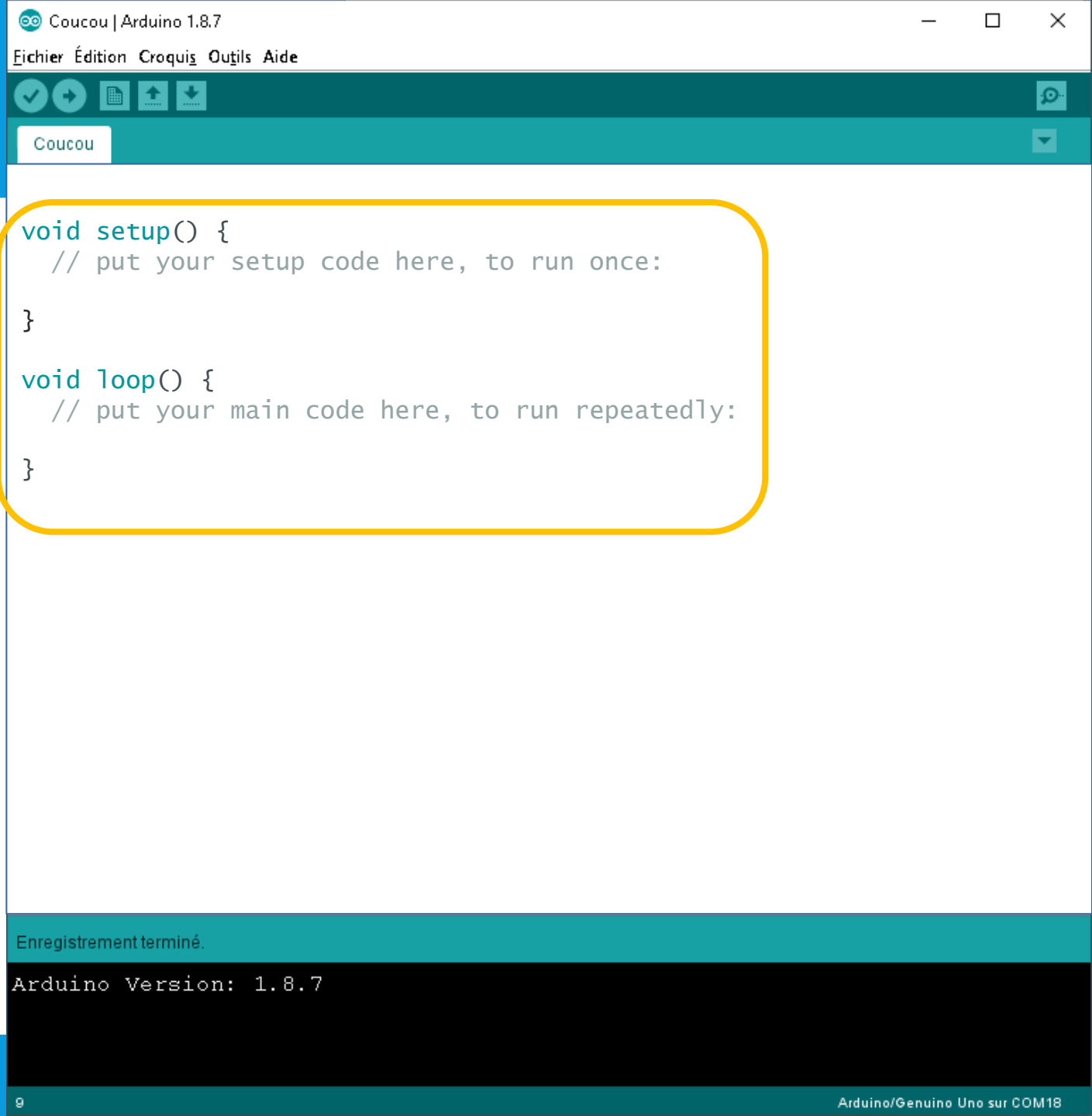
Le logiciel Arduino

 <https://www.arduino.cc/>



Le logiciel Arduino

Langage C++
Code minimal !



Le logiciel Arduino

Une fonction

```
void setup()
```

```
{
```

```
  // put your setup code here, to run once:
```

```
}
```

```
void loop()
```

```
{
```

```
  // put your main code here, to run repeatedly:
```

```
}
```

Le logiciel Arduino

Une fonction :

- Type
- Nom
- Paramètres
- Commence par { et finis par }

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

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

Le logiciel Arduino

Une fonction :

- **Type** : `void`
- **Nom** : `setup`
- **Paramètres** : aucun
- **Commence par** { **et finis par** }

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

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

Le logiciel Arduino

Une fonction :

- Type : `void`
- Nom : `setup`
- Paramètres : aucun
- Commence par `{` et finis par `}`

Un commentaire :

- Commence par `//`
ou
- Commence par `/*` et finis par `*/`

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

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

Le logiciel Arduino

Une fonction :

- Type : `void`
- Nom : `setup`
- Paramètres : aucun
- Commence par `{` et finis par `}`

```
void setup()
{
  // Commentaire
  Instruction1
  Instruction2
}
```

Un commentaire :

- Commence par `//`
ou
- Commence par `/*` et finis par `*/`

```
void loop()
{
  /* Super
  commentaire sur
  plusieurs lignes
  */
}
```

Le logiciel Arduino

Initialisation

```
void setup()  
{  
  
}
```

```
void setup()  
{  
  Préchauffer le four  
}
```

The screenshot shows the Arduino IDE interface. The title bar reads "Coucou | Arduino 1.8.7". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for saving, undo, redo, and other functions. The file name "Coucou" is visible in the top right. The main editor area contains the following code:

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

The `void setup()` function is highlighted with a yellow rounded rectangle. Below the editor, a status bar shows "Enregistrement terminé." and "Arduino Version: 1.8.7". The bottom right corner of the IDE displays "Arduino/Genuino Uno sur COM18".

Le logiciel Arduino

Boucle

```
void loop()  
{  
  
}
```

```
void loop()  
{  
  Préparer les ingrédients  
  Mélanger les ingrédients en suivant la recette  
  Cuire les cookies  
  Manger les cookies  
}
```

The screenshot shows the Arduino IDE interface. The title bar reads "Coucou | Arduino 1.8.7". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for saving, running, and other functions. The file name "Coucou" is visible in the top right. The code editor displays the following code:

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

The `void loop()` function is highlighted with a yellow rounded rectangle. Below the code editor, a teal status bar displays "Enregistrement terminé." and "Arduino Version: 1.8.7". The terminal window at the bottom shows "9" and "Arduino/Genuino Uno sur COM18".

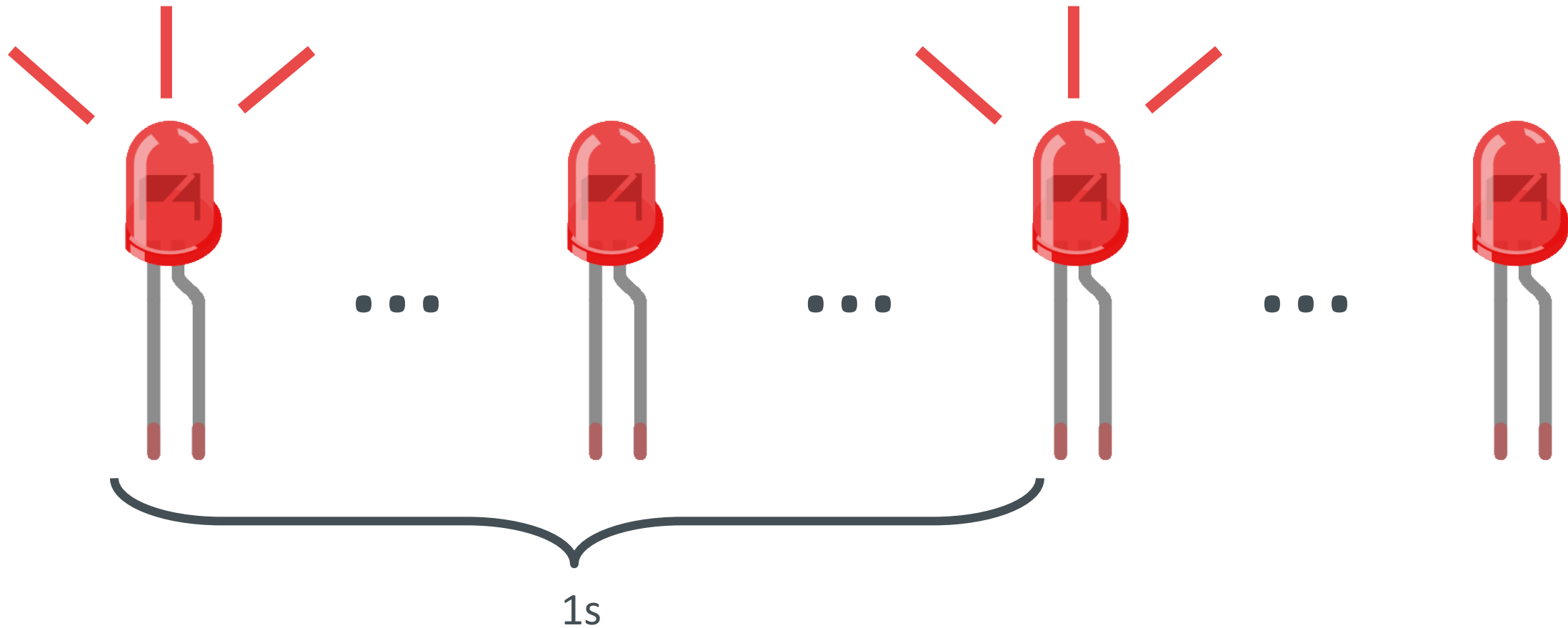
Enregistrement terminé.

Arduino Version: 1.8.7

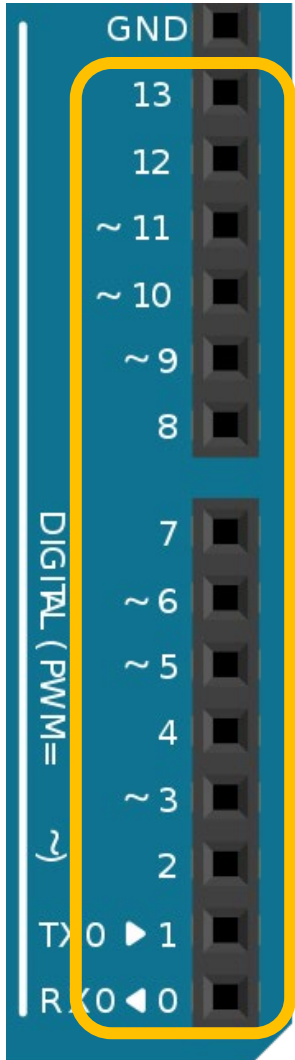
Notre premier programme

Hello World !

Objectif : Faire clignoter une LED à la seconde



Les entrées et sorties numériques



GPIO = General Purpose Input Output

Modes de fonctionnement

Entrée numérique :

Permet de lire une valeur appliquée en tension sur la broche :

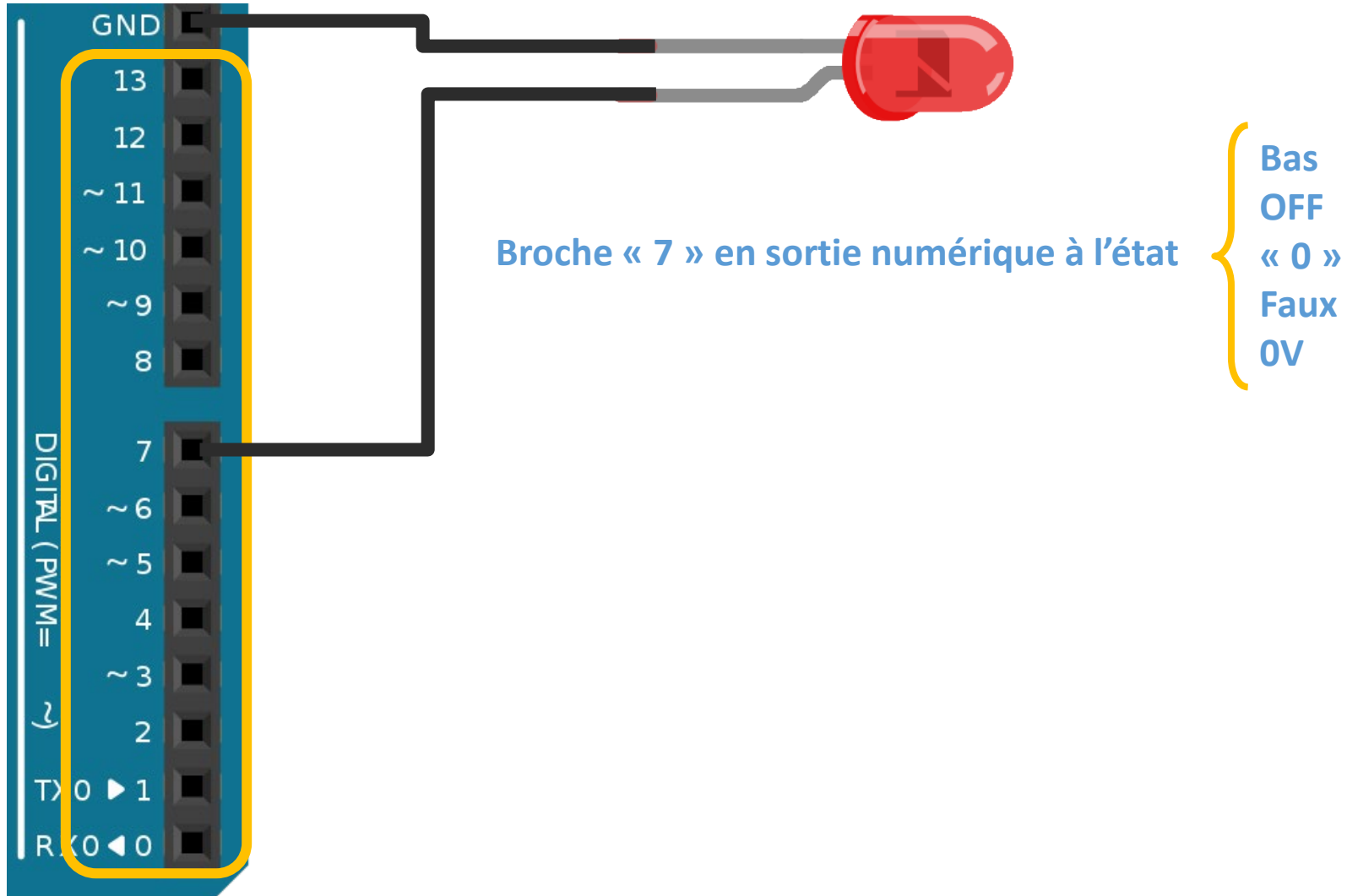
- 0V → État bas / Faux / « 0 » / OFF
- 5V → État haut / Vrai / « 1 » / ON

Sortie numérique :

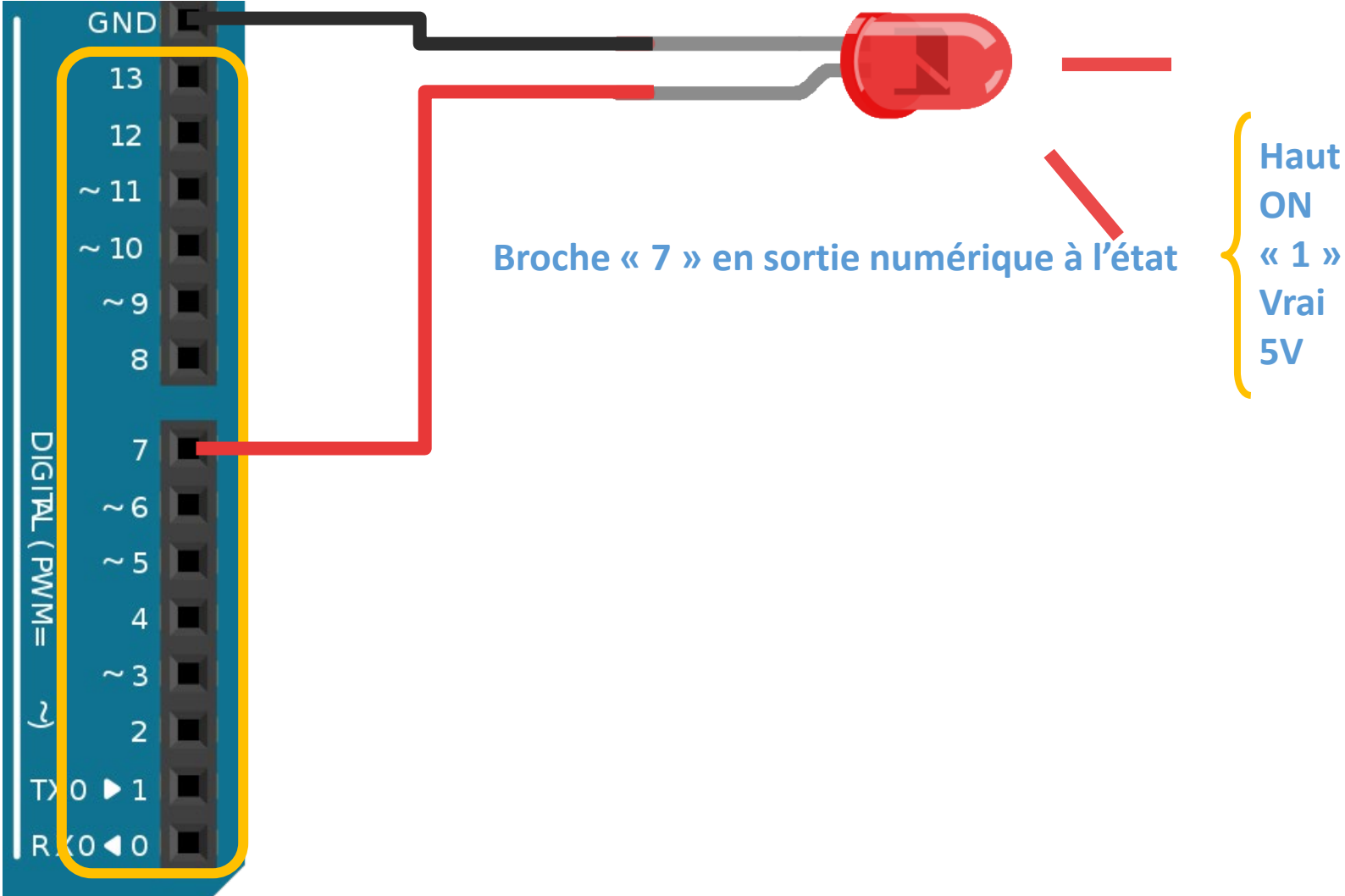
Permet d'imposer une tension sur la broche :

- État bas / Faux / « 0 » → 0V
- État haut / Vrai / « 1 » → 5V

Les entrées et sorties numériques

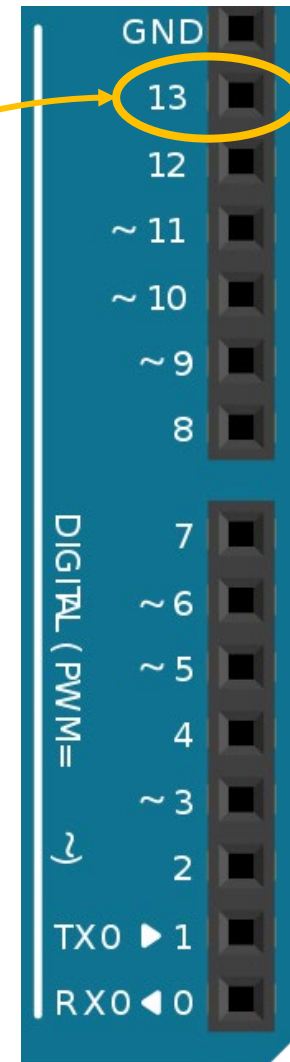
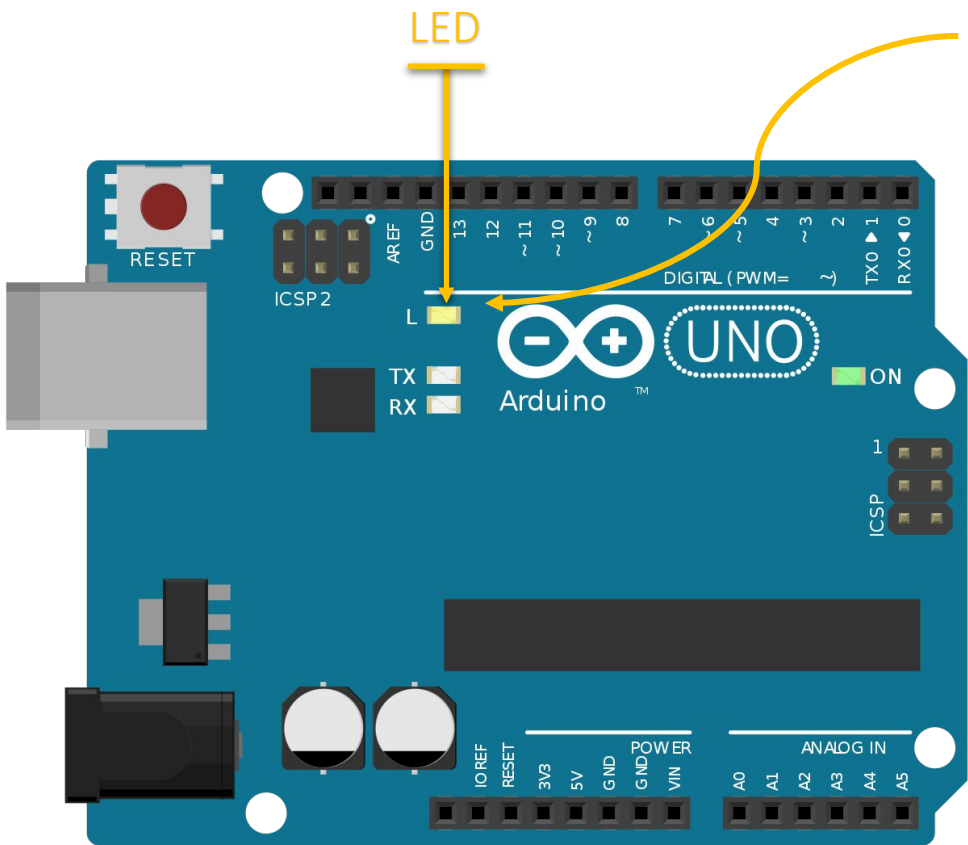


Les entrées et sorties numériques



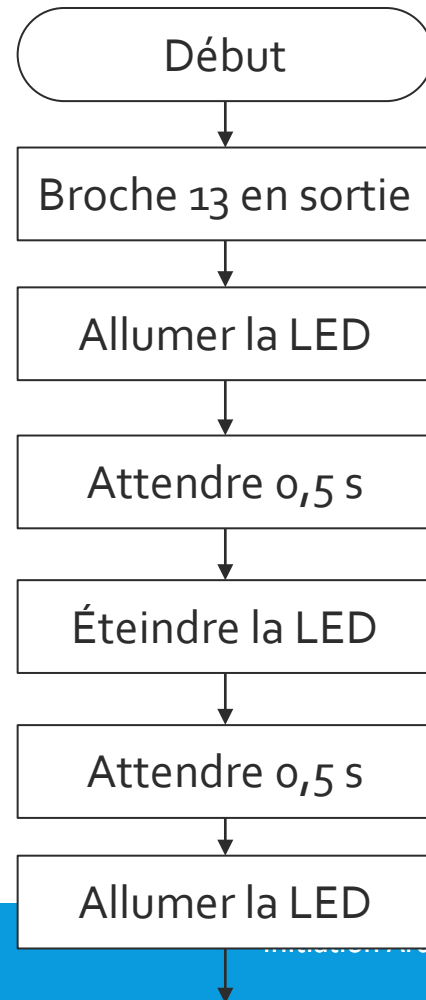
Hello World !

Objectif : Faire clignoter une LED à la seconde



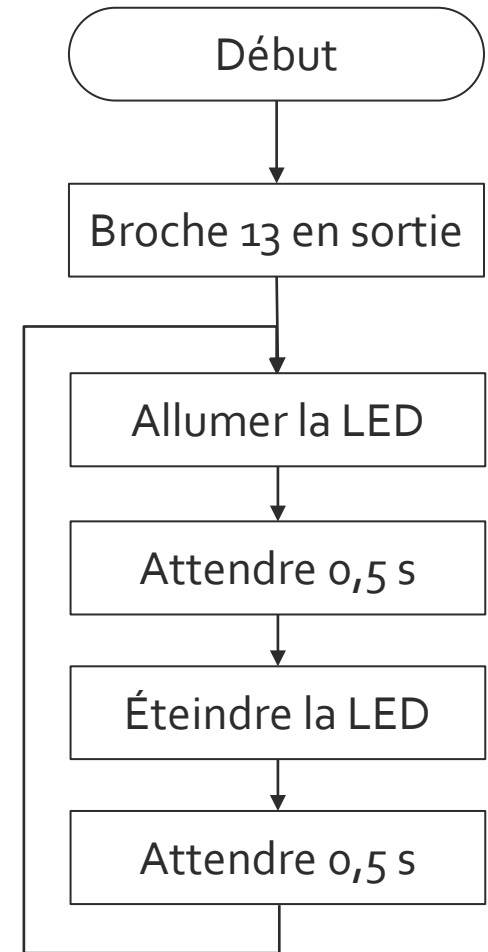
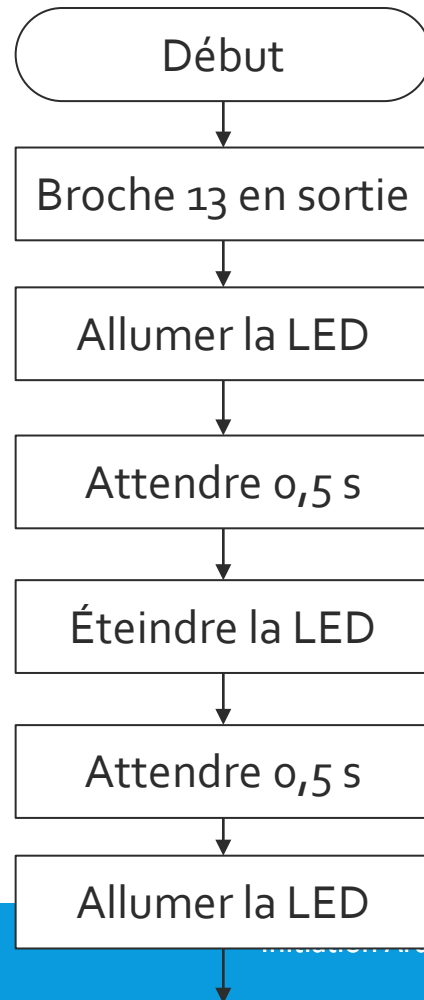
Hello World !

Objectif : Faire clignoter une LED à la seconde



Hello World !

Objectif : Faire clignoter une LED à la seconde



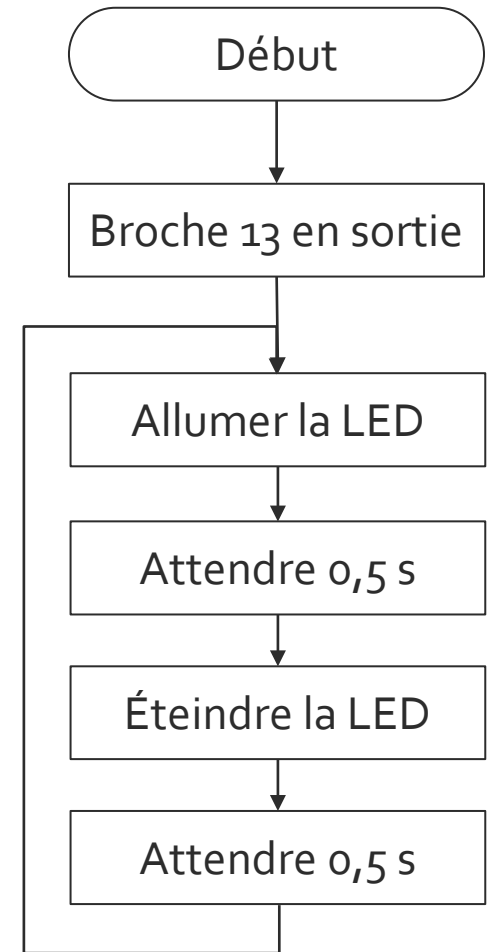
Hello World !

Objectif : Faire clignoter une LED à la seconde



Initialisation

Boucle



Hello World !

Objectif : Faire clignoter une LED à la seconde

Initialisation : `pinMode(pin, mode);`

`pin` : Numéro du pin à initialiser

`mode` :

INPUT : Pour récupérer une information

OUTPUT : Pour envoyer une commande



Hello World !

Objectif : Faire clignoter une LED à la seconde



Boucle : `digitalWrite`(pin, état);

pin : Numéro du pin à commander

état :

HIGH : Allumé

LOW : Éteint

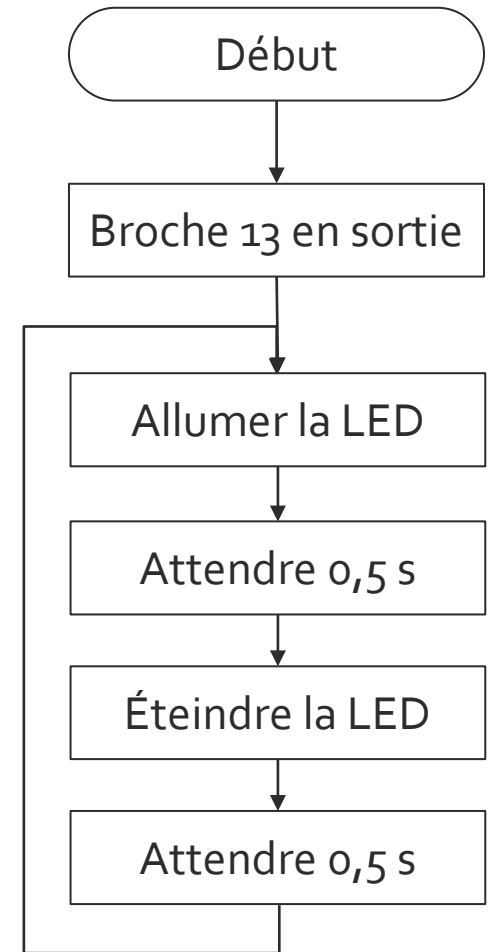
`delay`(temps);

temps : temps d'attente en millisecondes (1000ms = 1s)

Hello world !

```
void setup()
{
  // put your setup code here, to run once:
  pinMode(13, OUTPUT);
}

void loop()
{
  // put your main code here, to run repeatedly:
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```



Des fils en pagaille

Représenter un montage électronique

De l'électro-quoi ?

Objectif : Brancher une LED

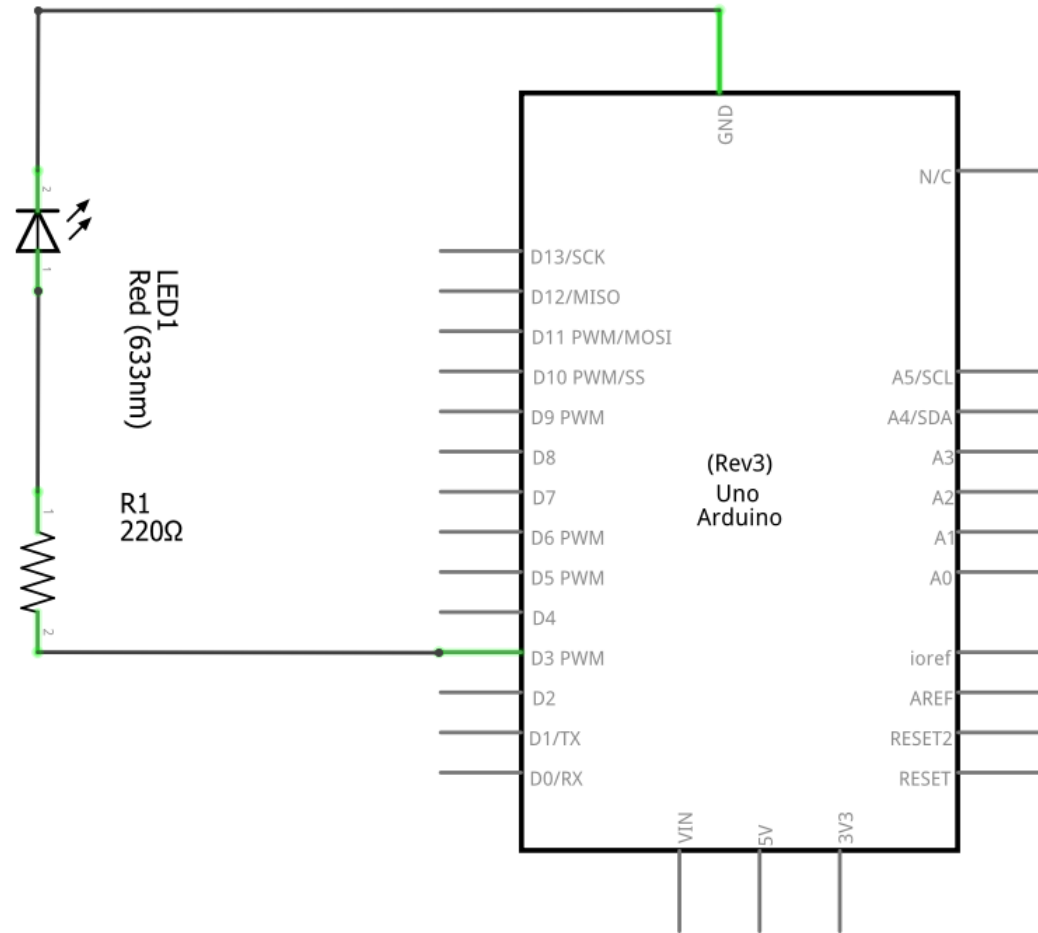
Calcul de la résistance R1

$$\begin{cases} V_f = 1,2V \\ I_N = 20mA \\ V_{CC} = 5V \end{cases}$$

$$R_1 = \frac{V_{CC} - V_f}{I_N} = \frac{5 - 1,2}{20 \times 10^{-3}} = 190\Omega$$

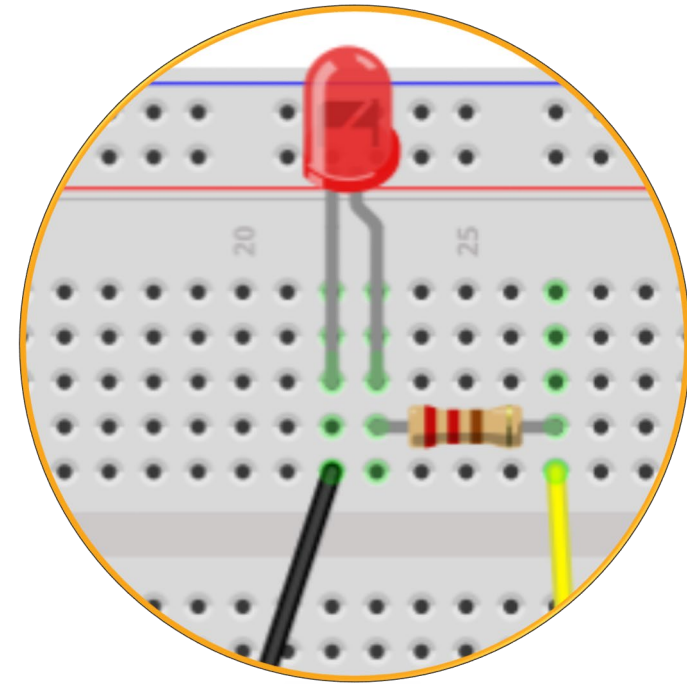
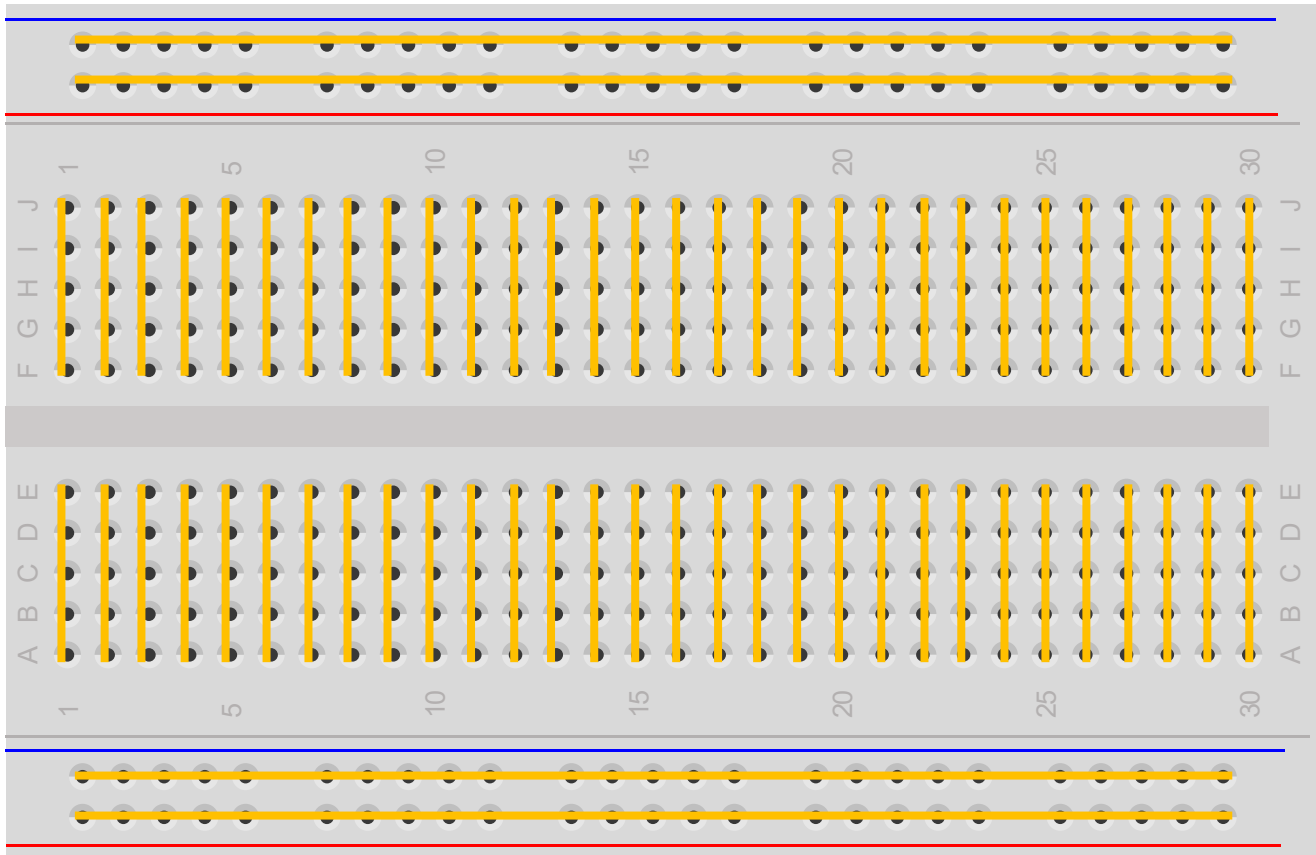
Valeur normalisée supérieure : $R_1 = 220\Omega$

Vue schématique



La Planche à pain - Reliez c'est gagné

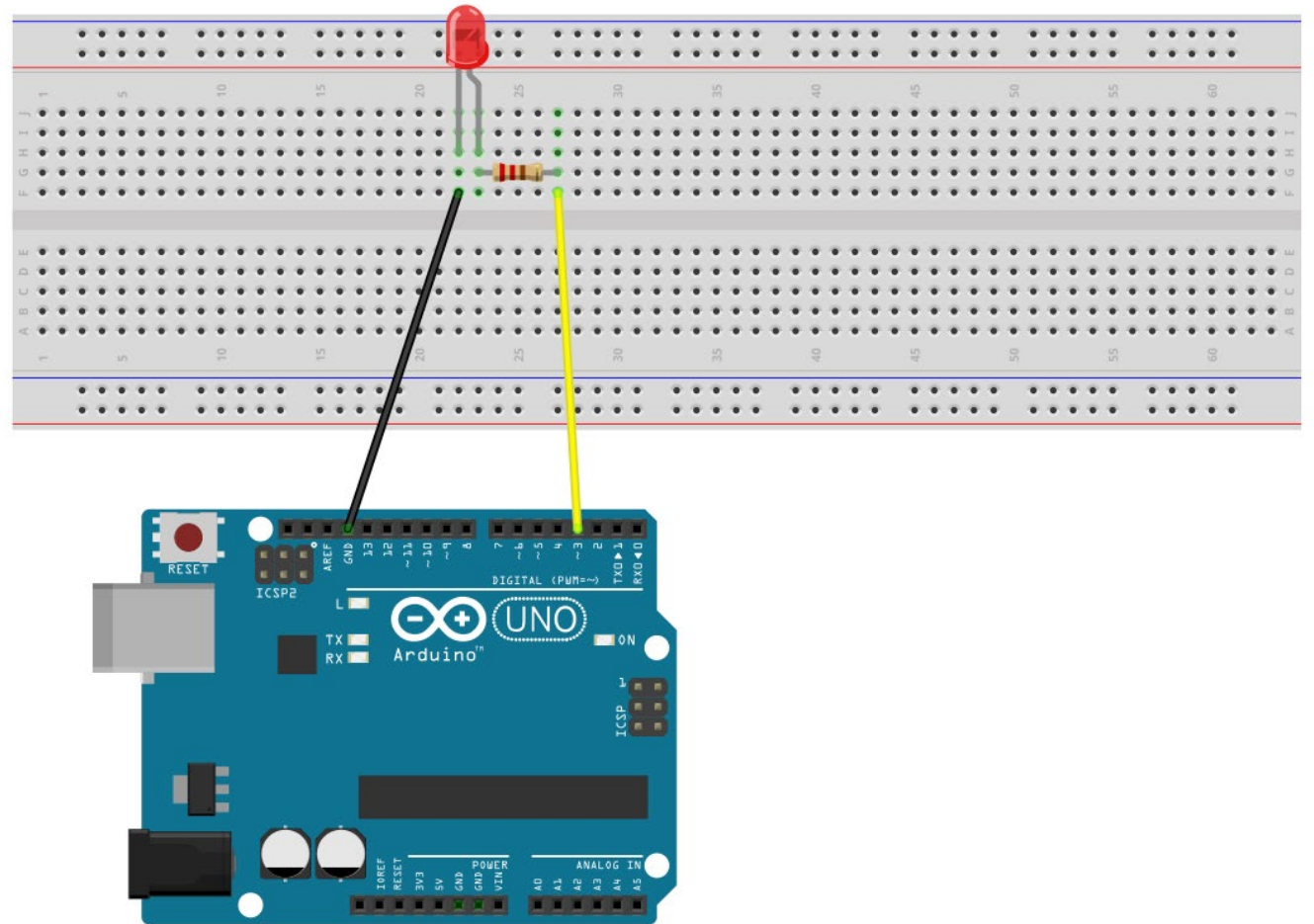
Objectif : Brancher une LED



La Planche à pain

Objectif : Brancher une LED

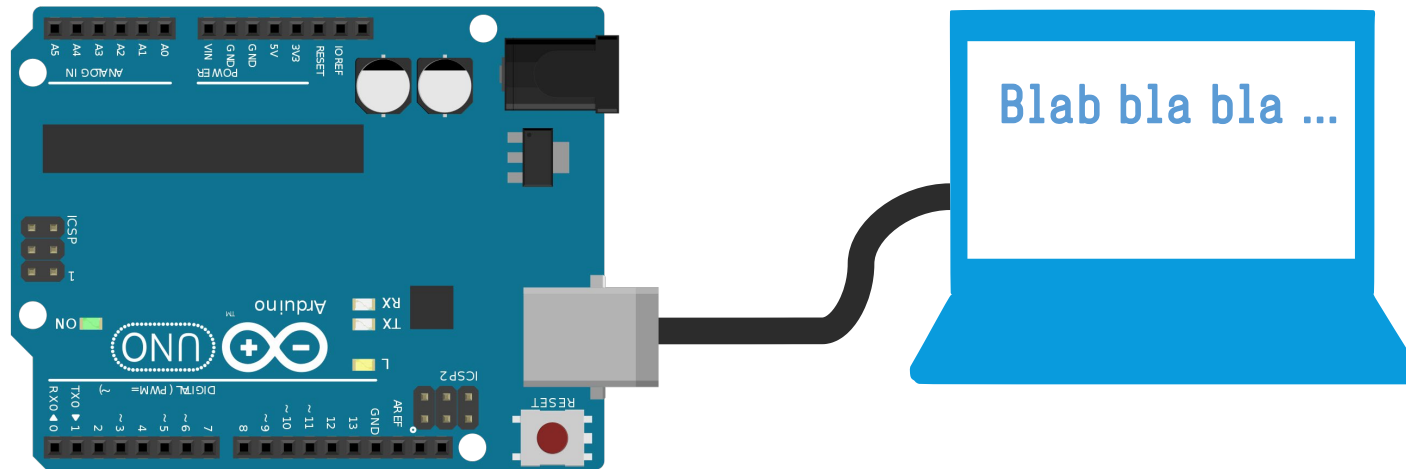
Vue



Communiquer avec l'ordinateur

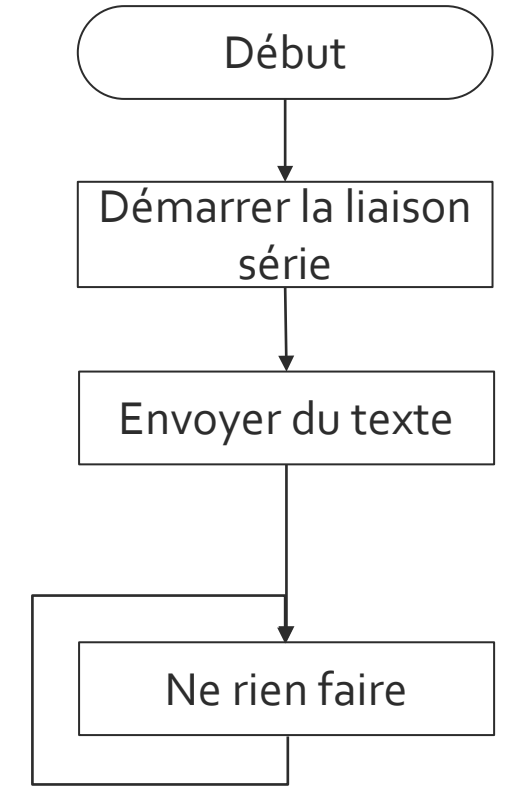
Liaison Série

Objectif : Faire clignoter une LED à la seconde



Initialisation

Boucle



Comment tu t'appelles ?

Objectif : Afficher son nom et son âge dans le moniteur série

Initialisation

```
Serial.begin(9600);
```

```
Serial.print("Texte a écrire");
```

```
Serial.println("Texte a écrire");
```

print : Écrit un texte à l'écran du moniteur série

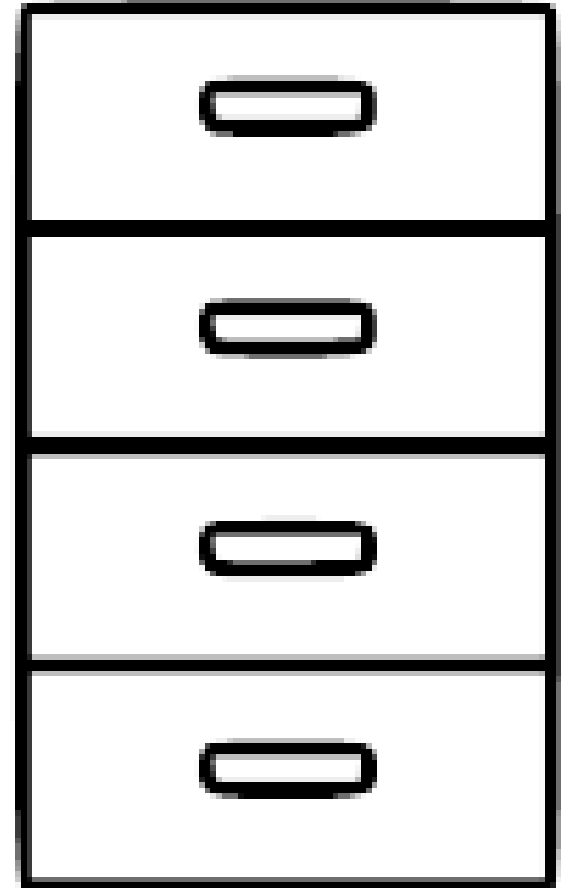
println : Écrit un texte à l'écran et **rajoute un saut de ligne**

Point chaussettes ! Ou, les variables

Des tiroirs dans la tête

Chaque tiroir à un nom

Chaque tiroir ne peut contenir qu'un seul type d'objet



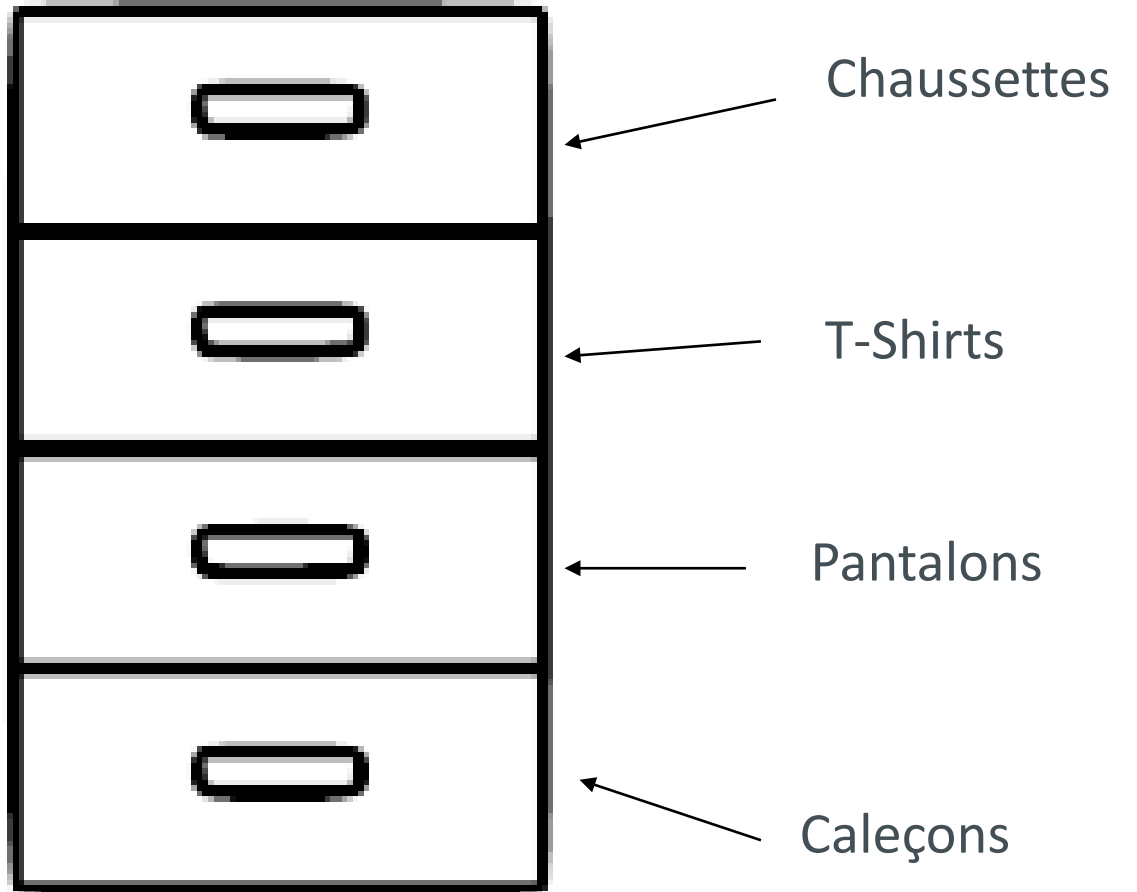
Point chaussettes ! Ou, les variables

Chaque tiroir à un nom

Chaque tiroir ne peut contenir qu'un seul type d'objet



Nom du tiroir



Point chaussettes ! Ou, les variables

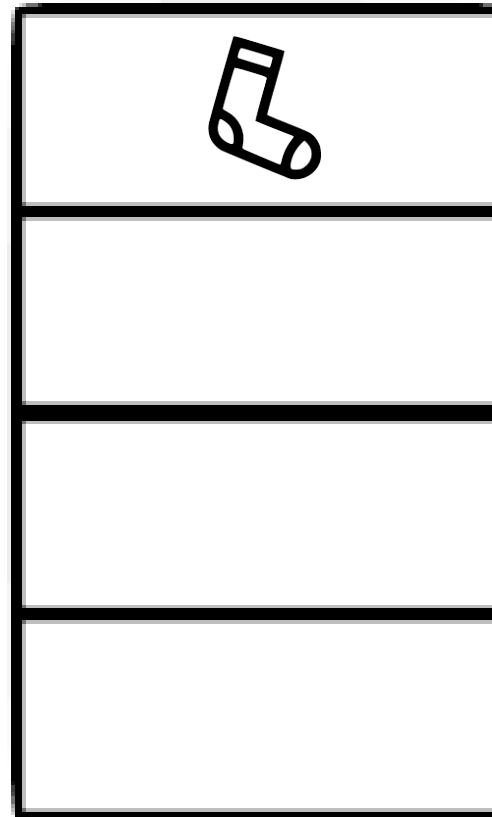
Chaque tiroir à un nom

Chaque tiroir ne peut contenir qu'un seul type d'objet

Nombre d'objets

Nom du tiroir

1



Chaussettes

T-Shirts

Pantalons

Caleçons

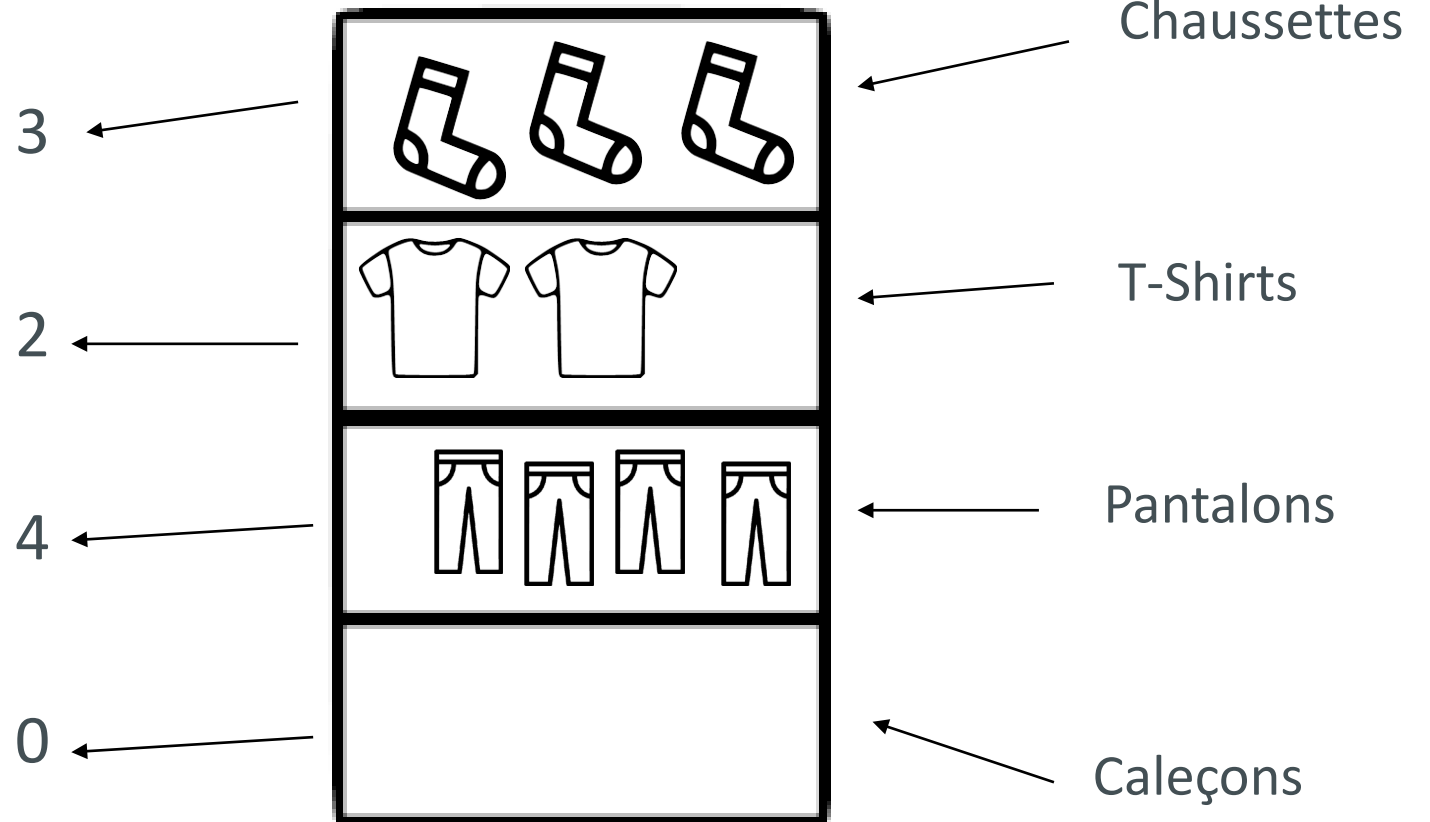
Point chaussettes ! Ou, les variables

Chaque tiroir à un nom

Chaque tiroir ne peut contenir qu'un seul type d'objet

Nombre d'objets

Nom du tiroir



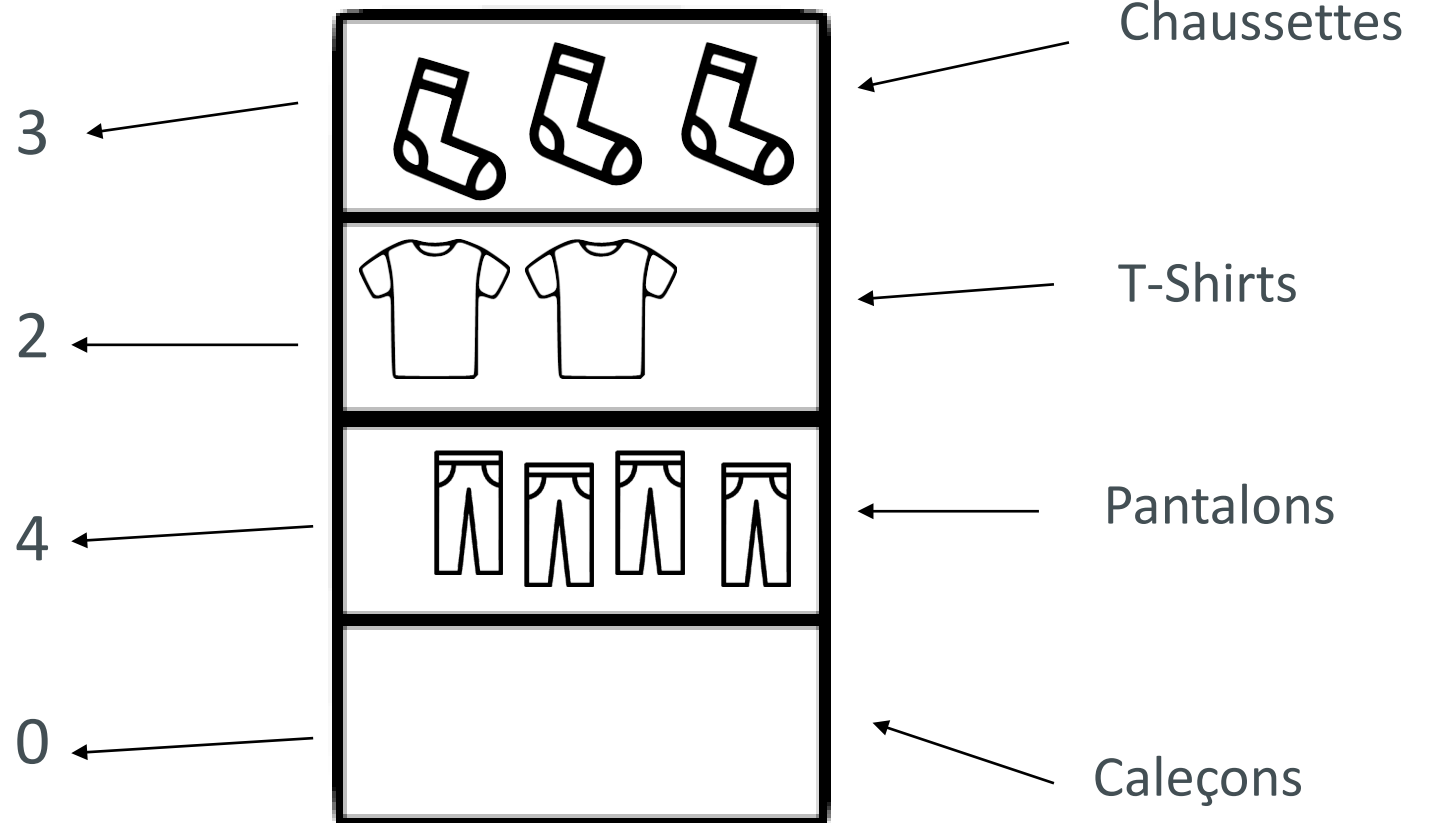
Point chaussettes ! Ou, les variables

Chaque **variable** à un nom

Chaque **variable** ne peut contenir qu'un seul type d'objet

Valeur ou Contenu

Nom de la variable



Le logiciel Arduino

Création d'une variable :

- Type : `int`
- Nom : NombreDeChaussette
- Valeur : 0

```
int NombreDeChaussette;
```

Création et initialisation d'une variable:

- Type : `int`
- Nom : NombreDePantalons
- Valeur : 4

```
int NombreDePantalons = 4;
```

Le logiciel Arduino

Types de variables :

Type	Représente	Minimum	Maximum
<code>int</code>	Un entier	-32 768	32 768
<code>unsigned int</code>		0	65 535
<code>long</code>		-2,147,483,648	2,147,483,648
<code>char</code>	Un caractère	-127	127
<code>unsigned char</code>		0	255
<code>float</code>	Nombre à virgule flottante	- 3.4028235 E +38	3.4028235E+38
<code>double</code>			

Point chaussettes ! Ou, les variables

Créer une variable: `int NombreDeChaussettes;`

Type : `int` Représente un nombre entier (-32 768 à 32 768)

Stocker dans une variable :

```
int NombreDeVetements = NombreDeChaussettes + NombreDePantalons;
```

Operations sur les variables :

Point chaussettes ! Ou, les variables

Créer une variable: `int NombreDeChaussettes;`

Stocker dans une variable : `NombreDeChaussettes = 2;`

`int NombreDePantalons = 3;`

Opérations sur les variables :

`int NombreDeVetements = NombreDeChaussettes + NombreDePantalons;`

Que la lumière soit

Objectif : Allumer une LED si il fait jour !



```
pinMode(pin, mode);
```

pin : Numéro du pin à initialiser

mode:

- **INPUT** : Pour récupérer une information (ex : capteur)
- **OUTPUT** : Sortie

```
valeur = analogRead(pin);
```

pin : Numéro du dont on veut lire la valeur

Que la lumière soit !

Objectif : Allumer une LED **SI** il fait jour !



Si Luminosité > 400
Éteindre la LED

Sinon
Allumer la LED

```
if(Luminosite > 400)
{
    Éteindre la LED
}
else
{
    Allumer la LED
}
```

A votre tour !

pinMode(pin, mode);

pin : Numéro du pin à initialiser
mode:

INPUT : Pour récupérer une information (ex : capteur)
OUTPUT : Sortie

digitalWrite(pin, état);

pin : Numéro du pin à commander
état :

HIGH : Allumé
LOW : Éteint

vaieur = **analogRead**(pin);

pin : Numéro du dont on veut lire la valeur

delay(temps);

temps : temps d'attente en millisecondes (1000ms = 1s)

Les variables

Créer une variable: `int NombreDeChaussettes;`

Stocker dans une variable : `NombreDeChaussettes = 2;`

`int NombreDePantalons = 3;`

Opérations sur les variables :

`int NombreDeVetements = NombreDeChaussettes + NombreDePantalons;`

Les conditions

```
if(Luminosite > 400)
{
    Éteindre la LED
}
else
{
    Allumer la LED
}
```