

Présentation et principe de l'Arduino

1. Généralités

Un Arduino représente des cartes électroniques regroupant plusieurs composants électroniques afin de réaliser des objets électroniques interactifs. Il peut être vu comme un ordinateur.

On retrouve de la **mémoire** sous deux types :

- la mémoire morte, qui contiendra les instructions que l'Arduino devra exécuter
- la mémoire vive, contenant les informations qui changent : les variables

On aura également un **calculateur** qui traitera et exécutera les instructions.

Tous ces éléments sont contenu dans un micro contrôleur. Il est d'architecture soit **AVR** soit d'architecture **ARM** comme le Cortex m3.

Les cartes Arduino sont construits autour d'un microcontrôleur **Atmel AVR** pour les modèles :

- ATmega328,
- ATmega32u4,
- ATmega2560,
- ATmega168,
- ATmega1280
- ATmega8

D'autres cartes Arduino sont construits autour des microcontrôleur **Cortex** d'architecture **ARM**. C'est la société **STMicroelectronics** qui se lance dans l'aventure en mai 2016 avec les modèles **STM32** ¹⁾

Les schémas des cartes électroniques Arduino sont publiés en licence libre. Le microcontrôleur peut analyser et produire des signaux électriques de format analogique ou de format numérique. On utilisera les entrées sortie de la carte électronique.

Pour le programmer, il faut utiliser la prise USB qui permettra de le relier à l'ordinateur. Elle permet également d'alimenter l'Arduino. Plus tard, l'alimentation s'effectuera sur la broche **+Vin (7-12 V)** et **0V**.

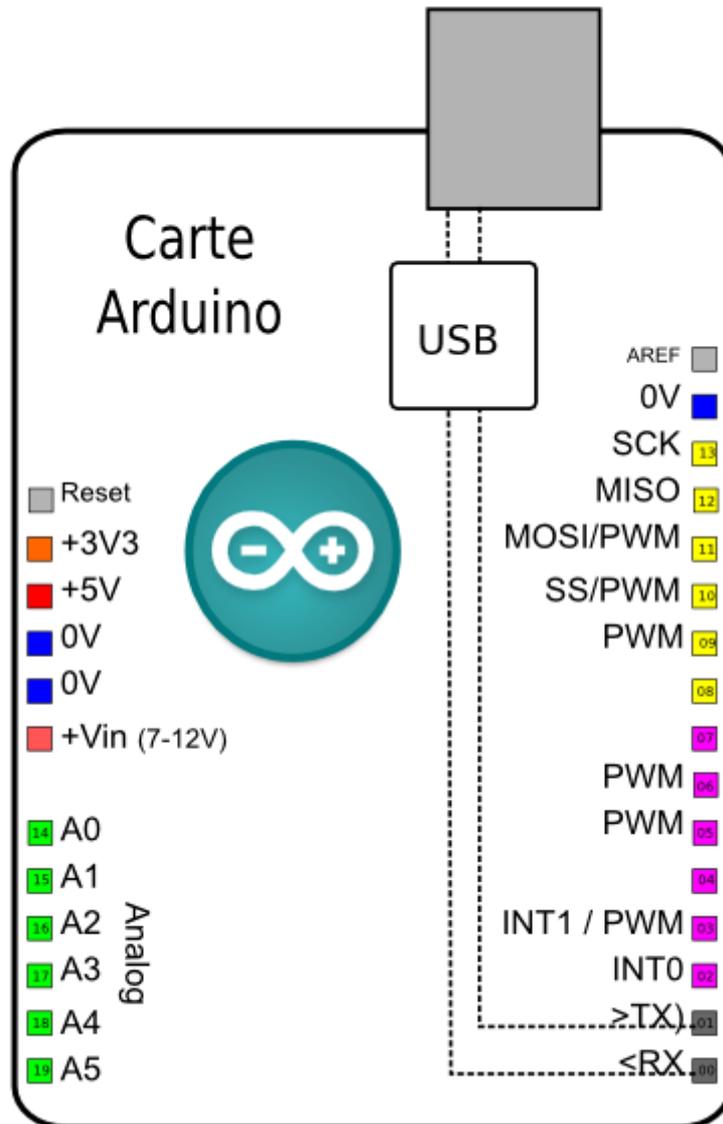
Une fois programmé, l'Arduino sera relié par un prise secteur ou une batterie via le port d'alimentation.

Les broches sont des entrées/sorties qui permettent de relier des capteurs, des LED, des moteurs ou des cartes d'extension. Cela permet le contrôle des appareils domestiques - éclairage, chauffage..., le pilotage d'un robot, de l'informatique embarquée, etc.

Les cartes Arduino et les schémas de ces cartes sont publiés en licence libre.

2. Principe général

Les différentes versions des Arduino fonctionnent sous le même principe général :



Les broches de **1** à **13**, ce sont les broches dites numériques (**0** ou **1**) ou « tout ou rien » ; elles offrent en sortie du 5 V et acceptent en entrée du 5 V sur le même principe. Dans le code, on utilise les fonctions `digitalWrite()` et `digitalRead()`.

Les broches de **14** à **19**, ce sont les broches dites analogiques, valeur entre **0 V** et **5 V**. Dans le code, on utilise les fonctions `analogWrite()` et `analogRead()`

Enfin, on y trouve également les différentes broches d'alimentation :

- Rouge : sortie 5 V (+5V)
- Orange : sortie 3,3 V (+3V3)
- Bleue : les masses (0V)
- Saumon : entrée reliée à l'alimentation de **+Vin (7-12V)**

Il y a des variations entre les différentes cartes (par exemple : UNO, la patte 13 est équipée d'une résistance).

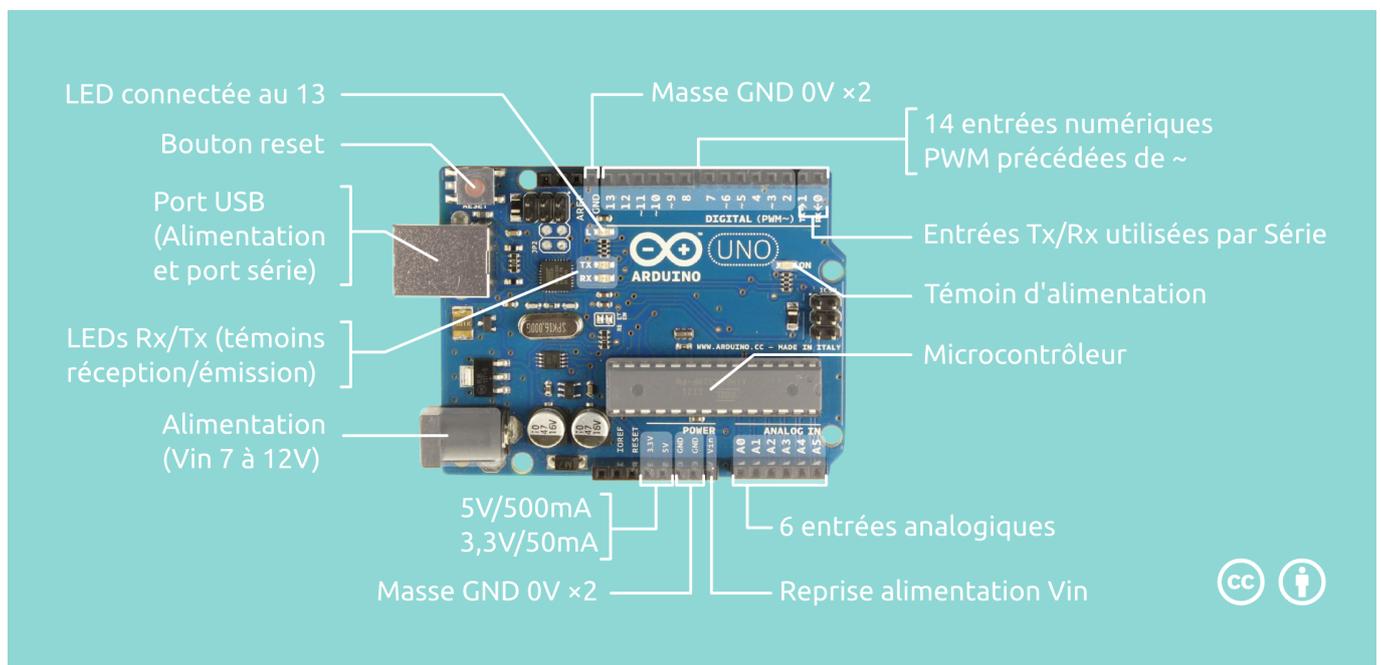
Les tensions admises des entrées/sorties sont strictement comprise entre **0 V** et **5 V**. Pas de tension négative !

Il existe plein de variantes de cartes Arduino :

- Arduino Nano
- Arduino NanoPro
- Arduino NanoPro mini
- Arduino NanoMega
- Arduino NanoDiecimila
- Arduino NanoDuemilanove
- Arduino NanoLeonardo
- Arduino NanoDue
- Arduino NanoFio

3. Présentation de l'Arduino Uno

L'**Arduino Uno** est basé sur un microcontrôleur ATME152 de 32 registres, cadencé à 166 MHz (un cycle de 6ns).



Crédit image : Mines Telecom - Programmer un objet avec Arduino, cours 04017

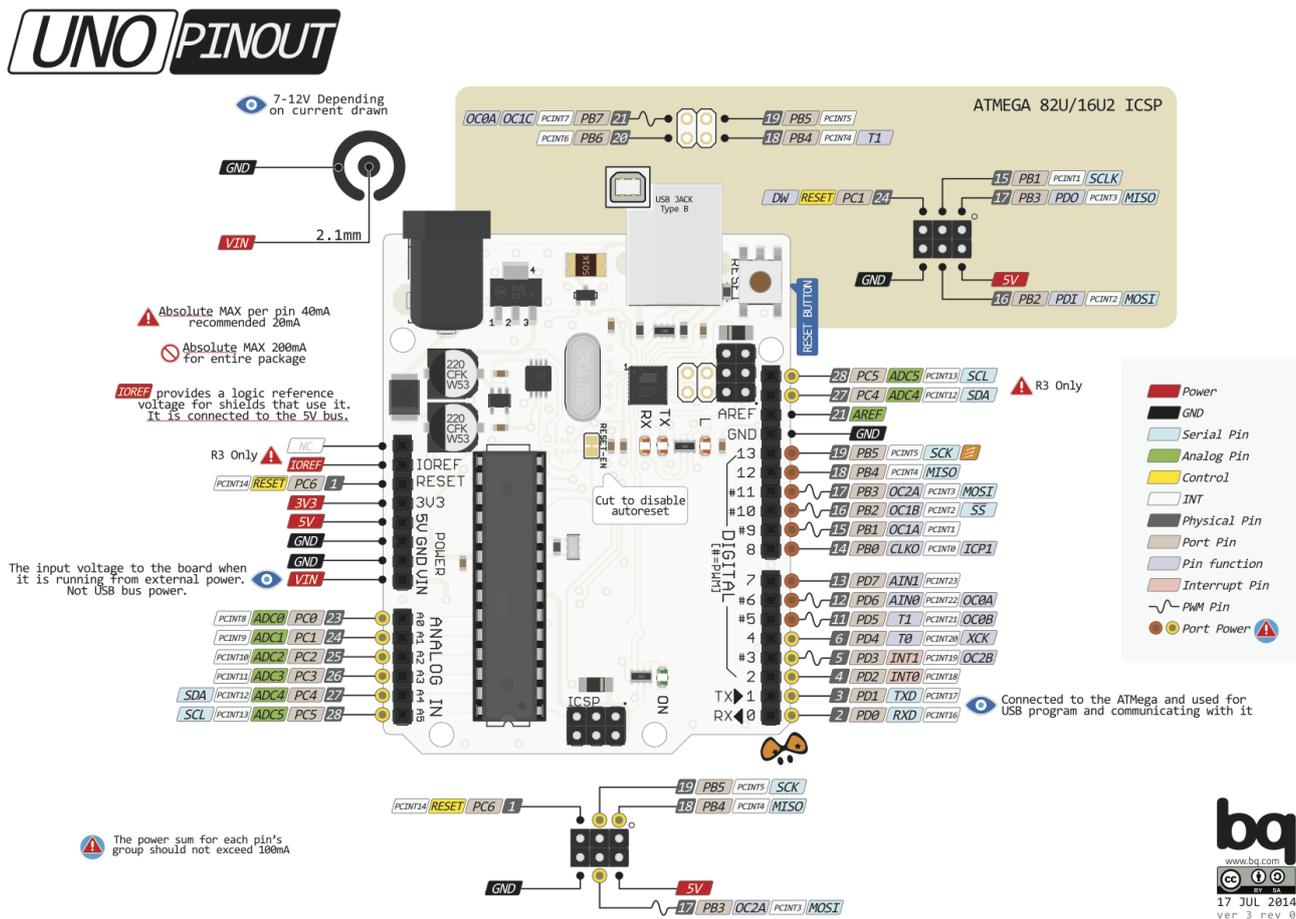
L'**Arduino Uno** propose :

- 14 entrées numériques
- 6 entrées analogiques
- 1 sortie 5 V à 500 mA
- 1 sortie 3,3 V à 50 mA
- 1 ports série Rx/Tx avec Led indépendantes

- 1 port USB, qui sert également d'alimentation et de connexion série
- 1 alimentation complémentaire (Vin 7 à 12 V)

On notera la présence d'une LED connectée sur la broche 13.

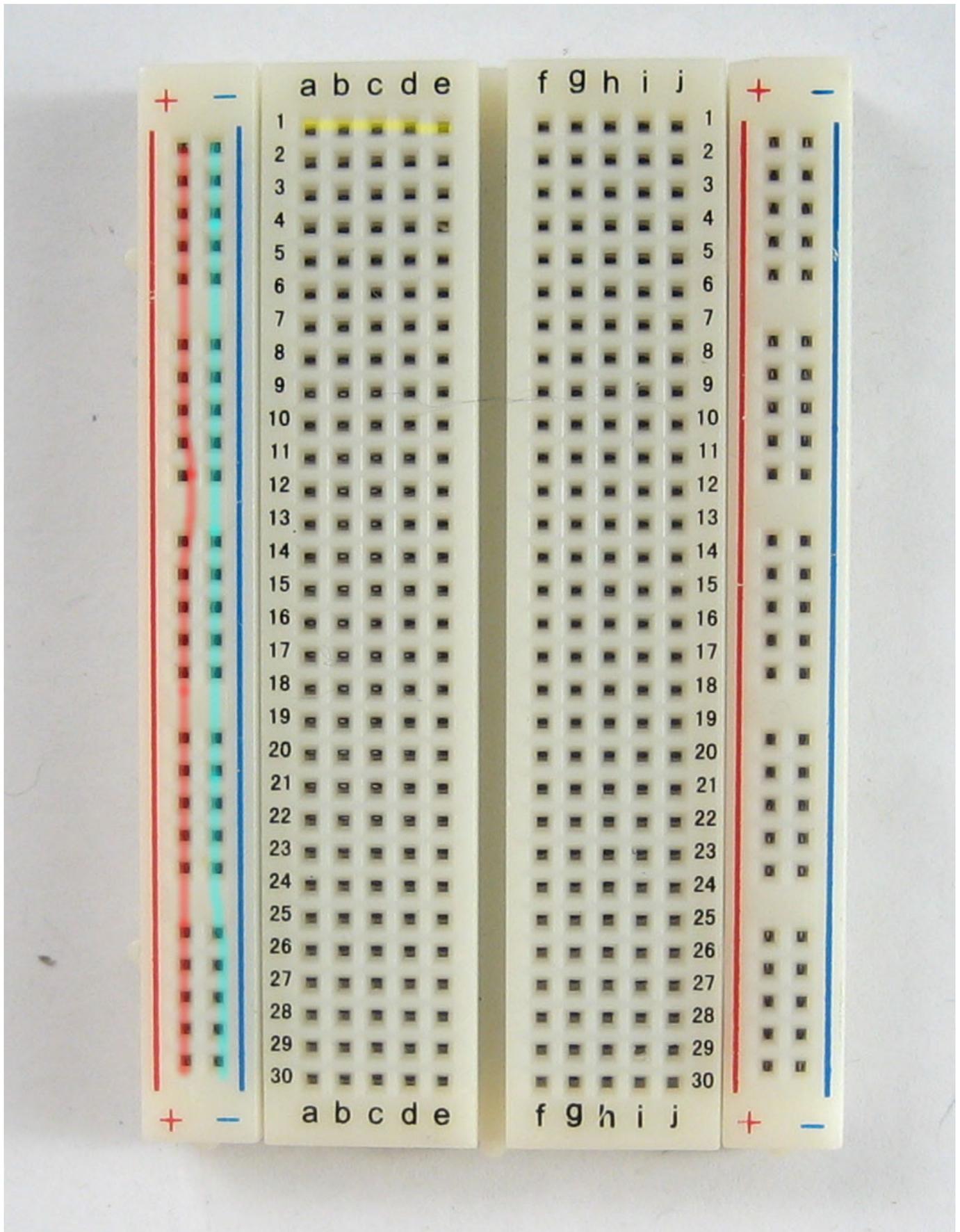
Les broches **Rx** et **Tx** de l'Arduino permettent d'effectuer une communication série. On veillera à ne pas utiliser ces broches pour brancher des LED ou d'autres composants.



4. Utiliser une plaque de prototypage

Pour tester ses branchements directement sur l'Arduino sans souder, il faut utiliser une platine d'essai (ou breadboard, plaque d'essai, plaque de montage rapide).

La platine d'essai est composée d'une multitude de trous. La plupart d'entre eux sont reliés. Voici une représentation :



En position portrait (comme sur la photo), la plaque est divisée à la verticale en 2 parties égales. Les 5 trous d'une ligne, d'une partie sont interconnectés. C'est-à-dire qu'il y a une liaison électrique pour les trous **a**, **b**, **c**, **d**, et **e** de la ligne 1, indépendante des lignes voisines et de l'autre partie de la plaque.

Sur notre plaque d'essai, nous avons 30 x 2 lignes de connexion. L'espace entre les deux parties est standardisé afin de positionner la plupart des circuits intégrés et de pouvoir leurs câbler toutes les broches.

Enfin, tous les trous de la colonne + d'une partie de la plaque sont interconnectés entre eux. Ceci représente une ligne d'alimentation et il faudra l'utiliser tel quel. Il en va de même pour la colonne - et l'autre partie de la plaque.

Sur la partie gauche de la photo:

- (trait rouge) à gauche, tous les trous de la colonne + sont reliés entre eux
- (trait bleu) tous les trous de la colonne -, sont reliés entre eux
- (trait jaune) tous les trous de la ligne 1 sont reliés entre eux, sur 5 colonnes (a, b, c, d et e). On peut reproduire ce schéma sur les 29 autres lignes.
- la partie de droite, reproduit le même schéma par effet miroir, avec un axe de symétrie vertical coupant la plaque en deux moitié égale.

Par convention, je branche l'alimentation 5 V sur la colonne + et la masse (0 V) sur la colonne -.

5. Logiciel de programmation d'un Arduino

Pour programmer un **Arduino** ou une carte programmable, il faut un éditeur qui fonctionne sur un ordinateur et un programme qui permettra de téléverser le code vers la carte.

Le programme libre de droit **Arduino Software (IDE)** permet d'écrire du code et de le téléverser dans la carte. C'est l'outil privilégié si vous avez acheté du matériel. Il fonctionne sous Windows, Mac OS X, et Linux. Le [guide d'installation du logiciel IDE Arduino sous Linux](#) permet de mieux appréhender une installation de la dernière version en date.

Il existe également un simulateur appelé **Tinkercad circuits** qui permet de tester votre code et les montages électroniques virtuellement.

6. Principe de programmation Arduino

Le langage Arduino est basé sur les langages C et C++. Les programmes Arduino sont intégrés dans un schéma (scketch) **setup / loop**. Il s'agit de deux blocs de fonctions obligatoirement présents dans tous programmes Arduino.

Un bloc commence par le symbole { et se termine par le symbole }. Le nom d'une fonction est toujours suivie par les symboles ().

La fonction **setup()** est appelée systématiquement au démarrage de l'Arduino, une seule fois, après un **reset** ou **une mise sous tension**. Il est utilisé pour initialiser des variables, démarrer des bibliothèques, modifier le paramétrage des broches, etc...

Après avoir utilisé la fonction **setup()**, la fonction **loop()** exécute de manière infinie le code à l'intérieur de ce bloc afin de répondre aux interactions demandées.

Afin de rendre plus intelligible le code écrit, il est possible d'écrire du texte qui ne sera pas interprété comme du code. Une ligne de commentaire commencera par les symboles `//` alors qu'un bloc de commentaires sera encadré par les symboles `/*` et `*/`.

Par exemple :

```
void setup() {
  // Ceci est une ligne de commentaire
}

void loop() {
  /*
   Voici un bloc
   de commentaire
   sur plusieurs lignes
  */
}
```

On peut utiliser d'autres fonctions qui permettent d'exécuter une ou plusieurs actions. Les fonctions sont définies avec :

- un nom représentant l'utilité du bloc.
- une ou plusieurs entrées. Il s'agit de paramètres ou arguments placés entre parenthèses.
- une sortie qui est le résultat de la fonction.

Par exemple, prenons le code suivant `pinMode(buttonPin, INPUT)`. Dans ce cas, la fonction est **pinMode** qui contient deux variables : **buttonPin** et **INPUT**.

```
int buttonPin = 3;

void setup() {
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

void loop() {
  // ...
}
```

7. Référentiel des fonctions, variables et structures du Langage Arduino

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

1)

https://www.st.com/content/st_com/ja/about/media-center/press-item.html/t3829.html

Last update: 2022/01/28 10:51
electronique:arduino:100-presentation-et-principe-de-l-arduino <https://www.abonnel.fr/electronique/arduino/100-presentation-et-principe-de-l-arduino>

From:
<https://www.abonnel.fr/> - **notes informatique & technologie**

Permanent link:
<https://www.abonnel.fr/electronique/arduino/100-presentation-et-principe-de-l-arduino>

Last update: **2022/01/28 10:51**

