

# PRESENTATION DE LA CARTE ARDUINO ET DE SON INTERFACE LOGICIELLE

## 1- LA CARTE A MICROCONTROLEUR

### Un peu d'histoire

Une équipe de développeurs a imaginé un projet répondant au nom de Arduino et mettant en œuvre une petite carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tout le monde en téléchargement gratuit. On dit que le matériel et le logiciel sont « open source ».

Le système Arduino a de nombreuses applications possibles :

- ⇒ contrôler des appareils domestiques
- ⇒ donner une "intelligence" à un robot
- ⇒ réaliser des jeux de lumières
- ⇒ permettre à un ordinateur de communiquer avec une carte électronique et différents capteurs
- ⇒ télécommander un appareil mobile (modélisme)

### La carte Arduino

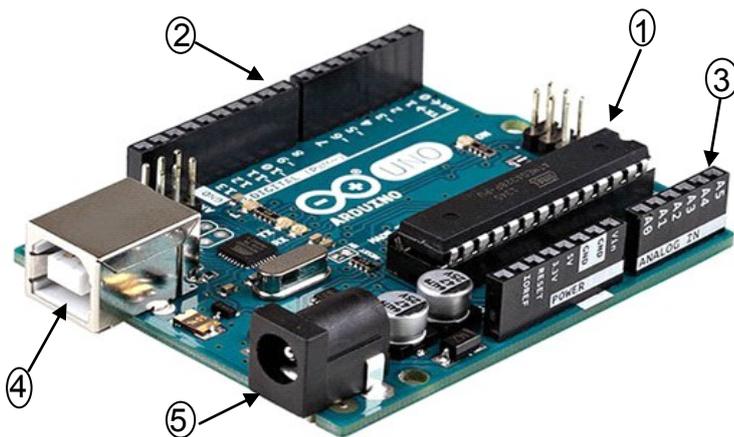
Les cartes Arduino font partie de la famille des microcontrôleurs.

Un microcontrôleur est une petite unité de calcul accompagnée de mémoire, de ports d'entrée/sortie et de périphériques permettant d'interagir avec son environnement.

Une carte Arduino est une interface programmable capable de piloter des capteurs et des actionneurs afin de simuler ou créer des systèmes automatisés.

Elle peut stocker un programme et le faire fonctionner.

La carte reçoit des informations analogiques ou numériques sur ces entrées. Le microcontrôleur traitera ces informations et les transmettra vers les sorties numériques.



### LEGENDE

- 1 : .....
- 2 : .....
- 3 : .....
- 4 : .....
- 5 : .....

Le microcontrôleur : .....

Le connecteur jack : .....

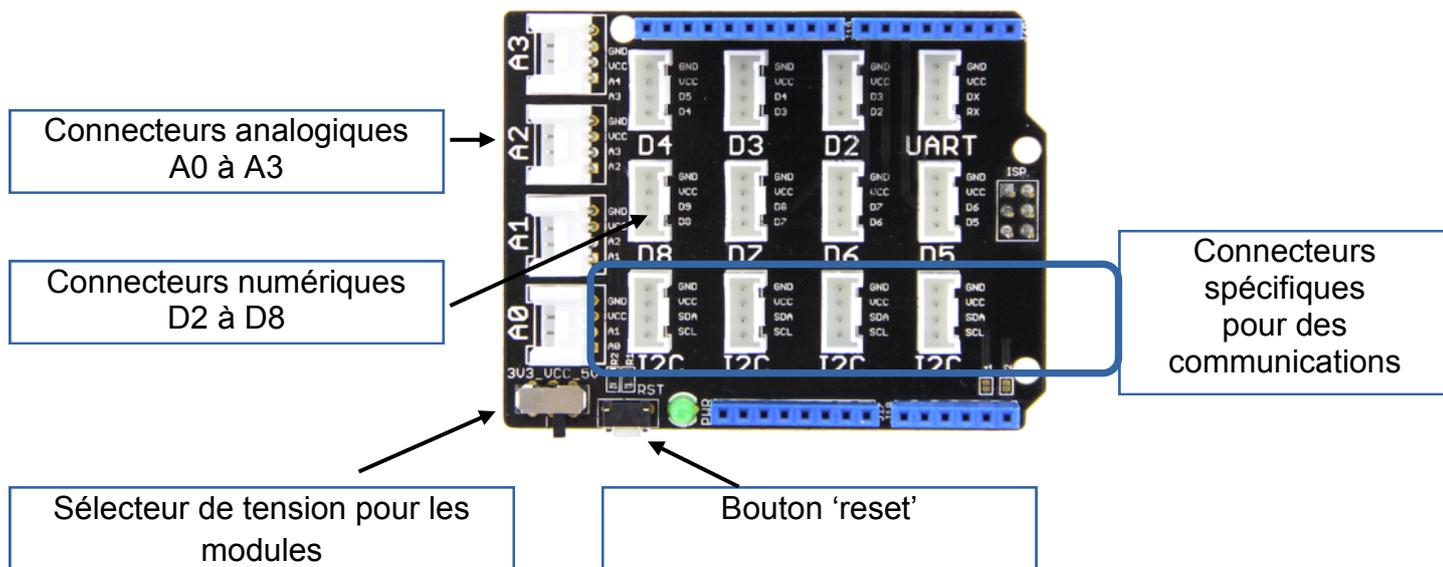
Le port USB : .....

## 2- LES SHIELDS

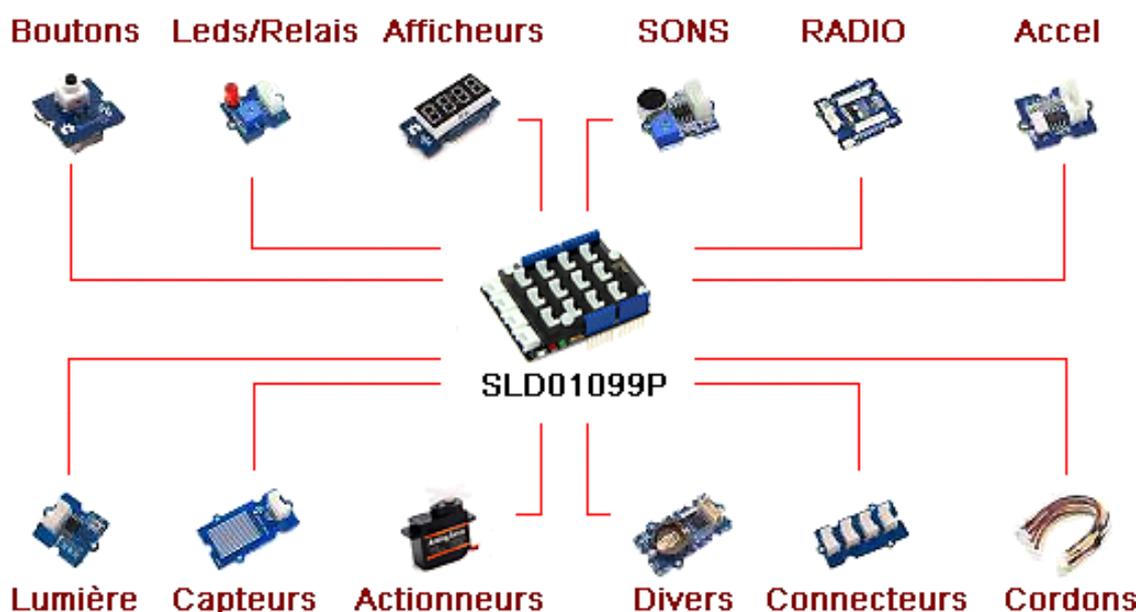
Un shield est une carte que l'on connecte directement sur la carte Arduino qui a pour but d'ajouter des composants sur la carte et augmenter ainsi ses fonctionnalités.

Le principal avantage d'un shield est sa simplicité d'utilisation. Il suffit de l'emboîter sur la carte Arduino pour le connecter. Cela facilite le câblage des composants. Les entrées et sorties sont repérées par des numéros.

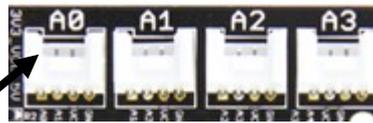
On raccorde rapidement et sans soudure tous les capteurs et les actionneurs qui servent à faire fonctionner le système.



Nous allons utiliser des « modules grove » qui sont des modules électroniques (capteurs ou actionneurs) prêts à l'emploi

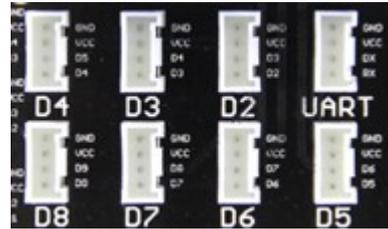


## 2.1 : les connecteurs analogiques :



Les entrées analogiques sont repérées par Elles sont capable de lire la valeur d'une tension renvoyée par un capteur, comme un potentiomètre, un capteur d'humidité, ou de température.

## 2.2 : Les connecteurs numériques :



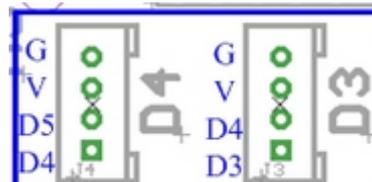
Ces connecteurs peuvent être utilisés comme Entrée ou Sortie et sont repérées par D2 .... D8 les entrées numériques détectent tout signal électrique (de type « tout ou rien ») renvoyé par un capteur, comme un bouton.

A noter : Quand on regarde de plus près les connecteurs, On s'aperçoit qu'ils ont des broches communes. Ainsi, Par exemple les connecteurs D3 et D4 ont en commun les bornes G, V et D4.

De même, les connecteurs D2 et D3 ont en commun les bornes G, V et D3.....

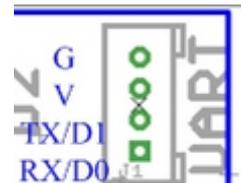
Les bornes D3,D5, D6, D9, D10, D11 sont configurées par défaut en PWM :

obtenir des effets d'allure analogique avec des broches numériques, pour atténuer la luminosité d'une LED par exemple



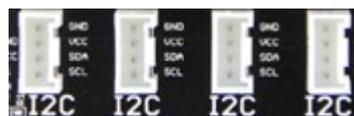
### Cas du connecteur « UART »:

Ce connecteur utilisant les bornes D0 et D1 de la carte UNO, est configuré par défaut pour servir de liaison série entre la carte et un port USB, un lecteur RFID, lecteur RF433 Mhz, lecteur Bluetooth, ... par exmple La borne D1 est alors dédiée à la Transmission (TX) et D0 à la réception (RX)



Ce matériel permet à la puce Atmega de recevoir une communication série même en travaillant sur d'autres tâches , tant qu'il ya place dans le tampon de série de 64 octets .

La bibliothèque SoftwareSerial a été développée pour permettre la communication série sur d'autres Arduino , en utilisant un logiciel pour ( d'où le nom " SoftwareSerial " ) .



répliquer la fonctionnalité broches numériques de l'

Il est possible d'avoir plusieurs logiciels ports série avec des vitesses allant jusqu'à 115200 bps . Un paramètre permet la signalisation inversée pour les appareils qui nécessitent ce protocole.

## 2.3 : Les connecteurs I2C :

I2C est le sigle d'Inter-Integrated Circuit.

À la fin des années 1970, la division des semi-conducteurs de chez Philips (maintenant devenue NXP) avait vu la nécessité de simplifier et standardiser les échanges de données entre les différents circuits intégrés dans leurs produits. Leur solution fut le bus I2C, elle réduisait le nombre de lignes nécessaires à seulement deux lignes, SDA - Serial DATA, et SCL - Serial CLock.

On utilise ce connecteur notamment pour l'utilisation des modules grove LCD



### 3 - LES DIFFERENTS CAPTEURS ET ACTIONNEURS ARDUINO

Connectez-vous sur le site de notre fournisseur « TECHNOLOGIE SERVICES » à l'aide des liens suivant :

Vous obtenez la liste de tous les modules possibles à utiliser

<http://www.technologieservices.fr/fr/r-edr1000001/recherche.html?motCle=module%20grove>

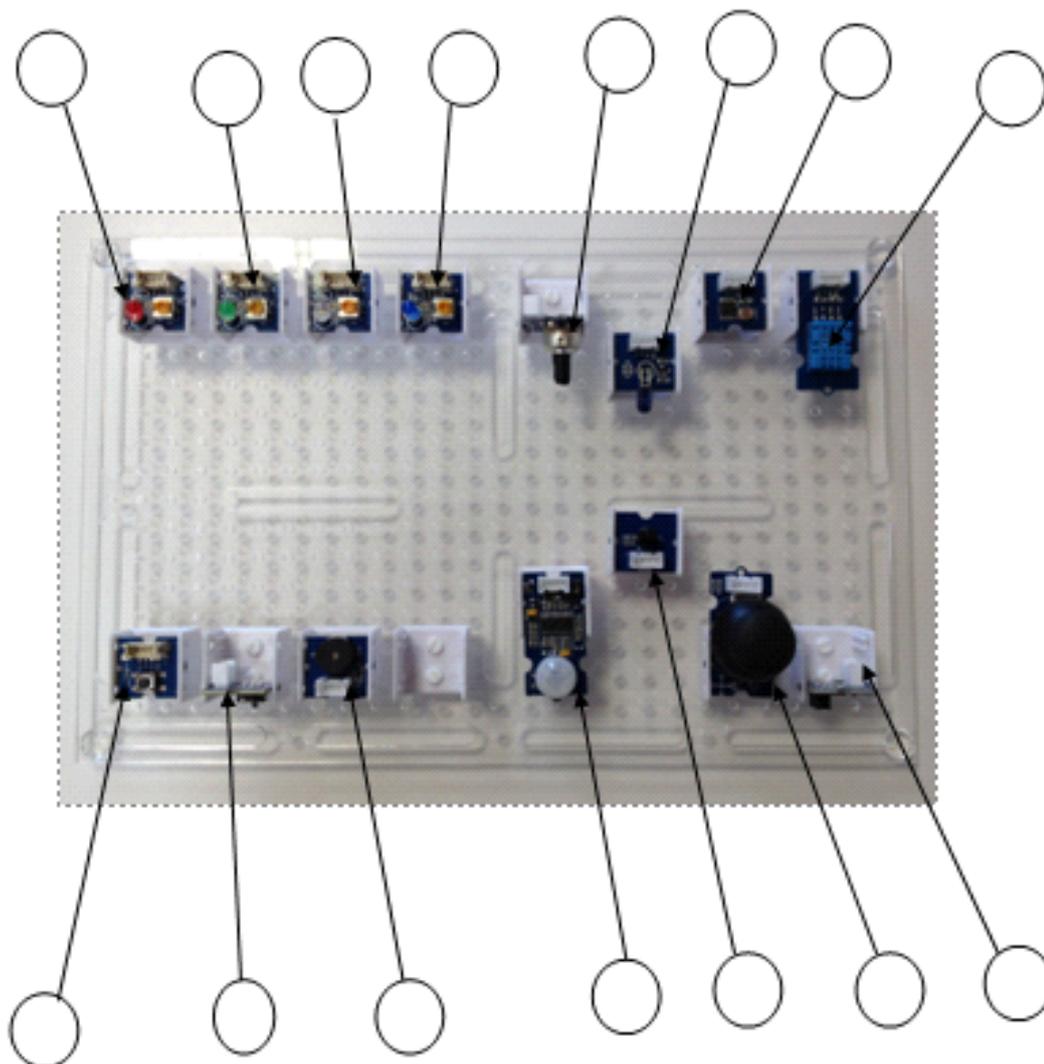
<http://www.technologieservices.fr/fr/r-edr1000001/recherche.html?motCle=servomoteur>

<http://www.technologieservices.fr/fr/c-c1000000847-epc1000003/categorie/Lycees-Technologiques-LEGT-Professionnels-CFA,Systemes-et-cartes-programmables,Modules-Grove.html>

Repérez les différents modules grove.

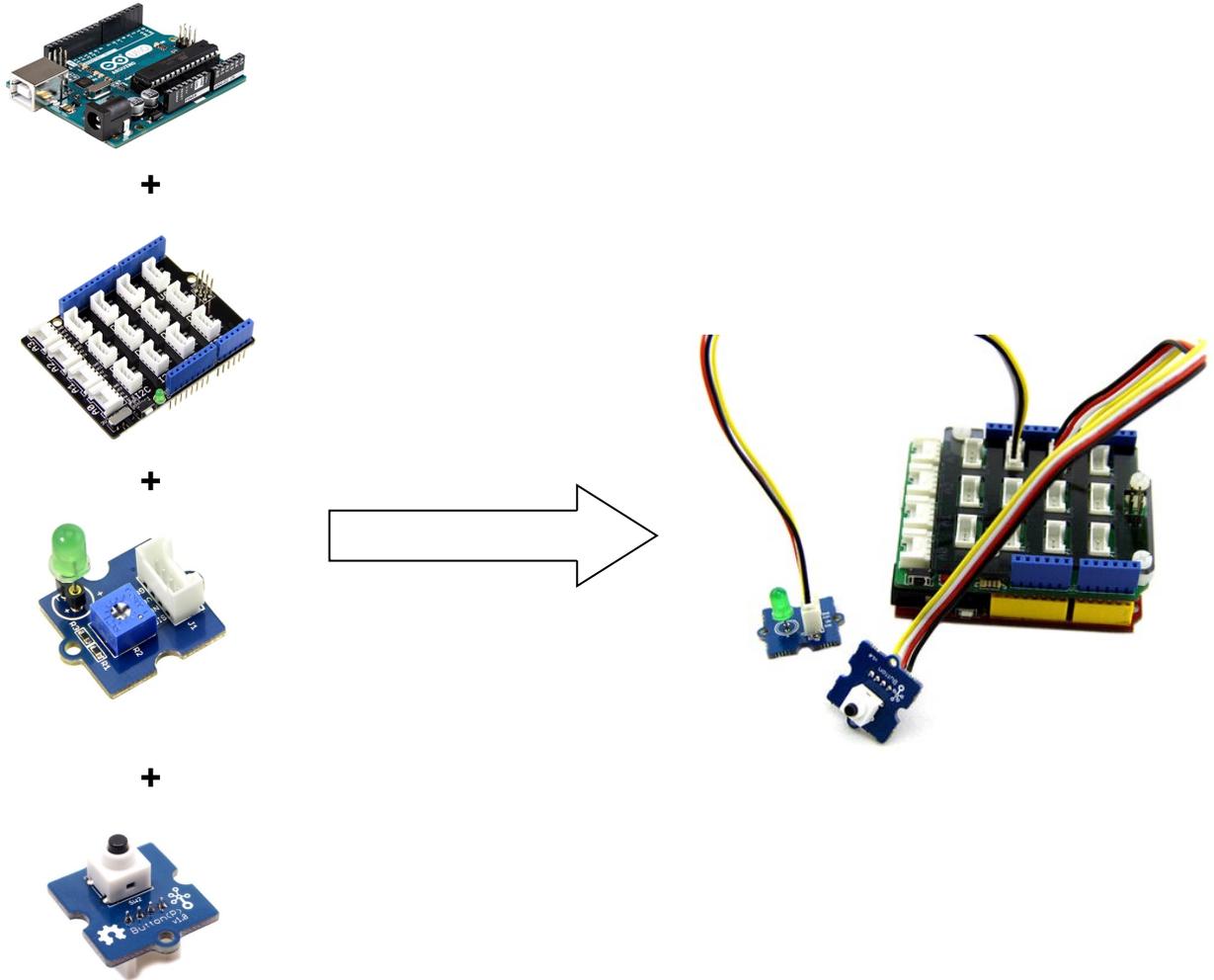
Puis complétez les colonnes vides du tableau en notant les informations demandées.

Légendez ensuite la photo de la carte Arduino en inscrivant le numéro de repère de chaque module listé dans le tableau.



Rep	Module Grove	photo	Capteur	Actionneur
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

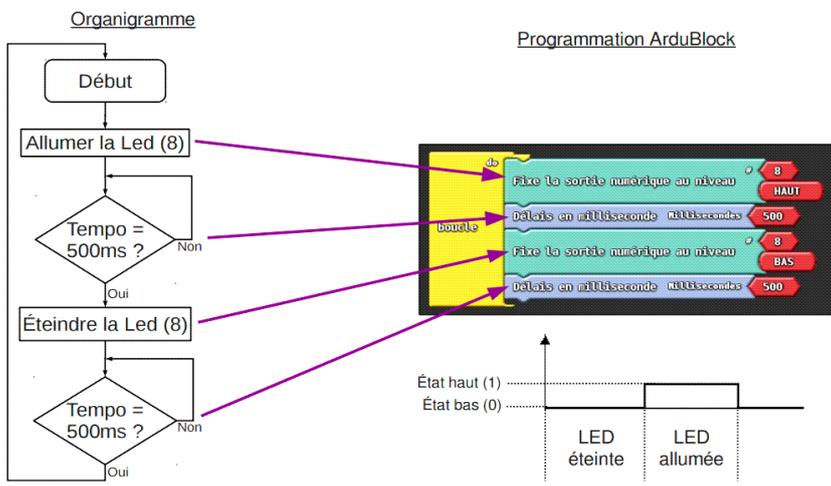
## 4 - ASSEMBLAGE DES ELEMENTS



## 5 - LA PROGRAMMATION

On programme la carte Arduino à l'aide du langage de programmation « ArduBlock ». Ce langage utilise des organigrammes que l'on retranscrit en insérant des blocks à la manière du langage « scratch ».

Voici un exemple d'organigramme associé à un programme :



Le logiciel Arduino IDE transforme ces blocks en langage PYTHON

```
int ledPin = 13;
// initialisation
void setup()
{
  // patte en sortie
  pinMode(ledPin, OUTPUT);
}
// boucle sans fin
void loop()
{
  // on allume la led
  digitalWrite(ledPin, HIGH);
  // on attends une seconde
  delay(1000);
  // on éteint la led
  digitalWrite(ledPin, LOW);
  // on attends une seconde
  delay(1000);
  // et on recommence (ne jamais oublier la dernière attente)
}
```

# PRESENTATION DE LA CARTE ARDUINO ET DE SON INTERFACE LOGICIELLE

## 1- LA CARTE A MICROCONTROLEUR

### Un peu d'histoire

Une équipe de développeurs a imaginé un projet répondant au nom de Arduino et mettant en œuvre une petite carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tout le monde en téléchargement gratuit. On dit que le matériel et le logiciel sont « open source ».

Le système Arduino a de nombreuses applications possibles :

- ⇒ contrôler des appareils domestiques
- ⇒ donner une "intelligence" à un robot
- ⇒ réaliser des jeux de lumières
- ⇒ permettre à un ordinateur de communiquer avec une carte électronique et différents capteurs
- ⇒ télécommander un appareil mobile (modélisme)

### La carte Arduino

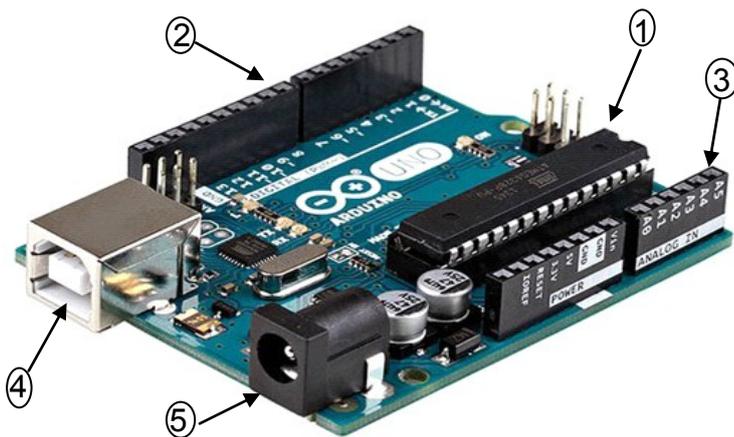
Les cartes Arduino font partie de la famille des microcontrôleurs.

Un microcontrôleur est une petite unité de calcul accompagnée de mémoire, de ports d'entrée/sortie et de périphériques permettant d'interagir avec son environnement.

Une carte Arduino est une interface programmable capable de piloter des capteurs et des actionneurs afin de simuler ou créer des systèmes automatisés.

Elle peut stocker un programme et le faire fonctionner.

La carte reçoit des informations analogiques ou numériques sur ces entrées. Le microcontrôleur traitera ces informations et les transmettra vers les sorties numériques.



### LEGENDE

- 1 : Microcontrôleur
- 2 : Entrées et sorties numériques
- 3 : Entrées analogiques
- 4 : Port USB
- 5 : Connexion jack alimentation

Le microcontrôleur : stocke et exécute le programme

Le connecteur jack : permet de rendre autonome la carte en y connectant une batterie ou une pile

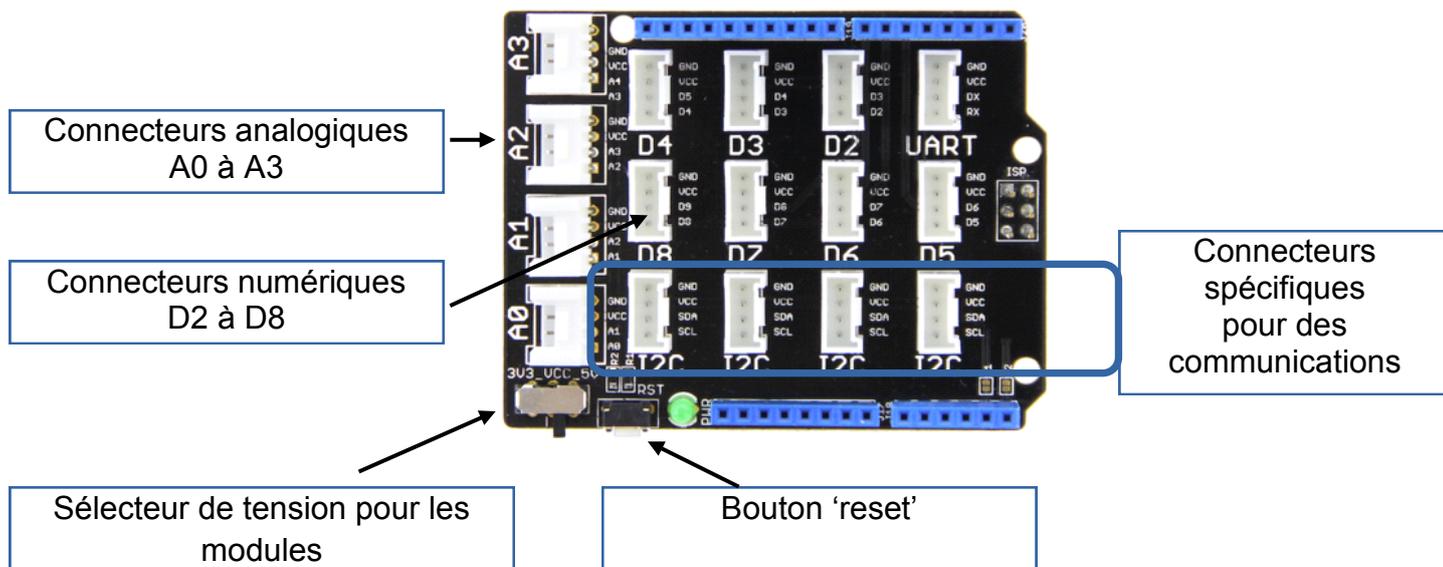
Le port USB : permet de communiquer avec la carte et de l'alimenter en 5 V

## 2- LES SHIELDS

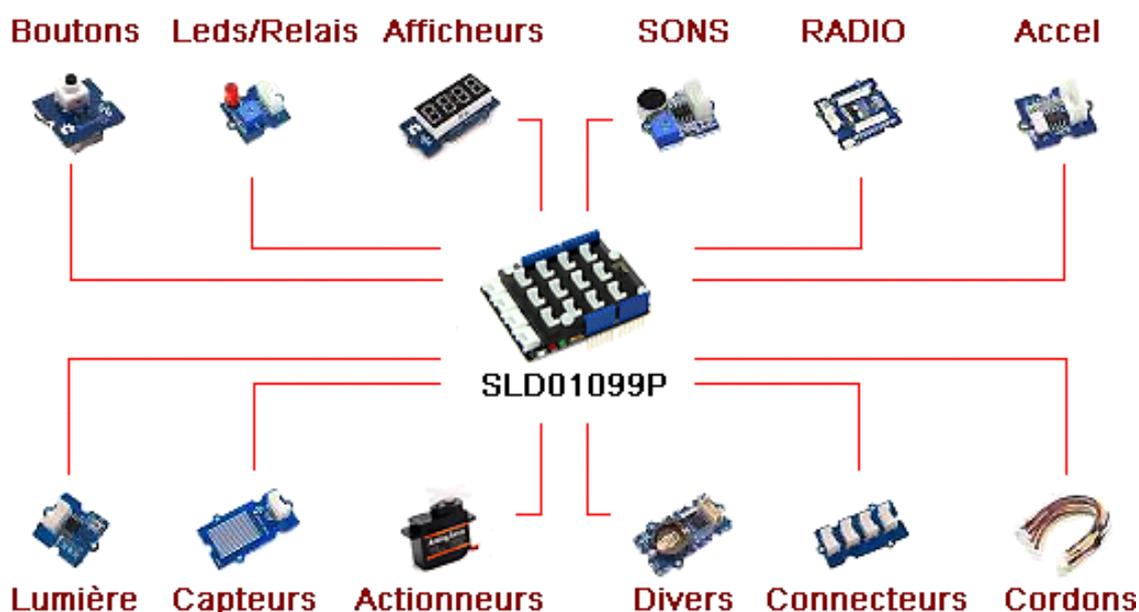
Un shield est une carte que l'on connecte directement sur la carte Arduino qui a pour but d'ajouter des composants sur la carte et augmenter ainsi ses fonctionnalités.

Le principal avantage d'un shield est sa simplicité d'utilisation. Il suffit de l'emboîter sur la carte Arduino pour le connecter. Cela facilite le câblage des composants. Les entrées et sorties sont repérées par des numéros.

On raccorde rapidement et sans soudure tous les capteurs et les actionneurs qui servent à faire fonctionner le système.



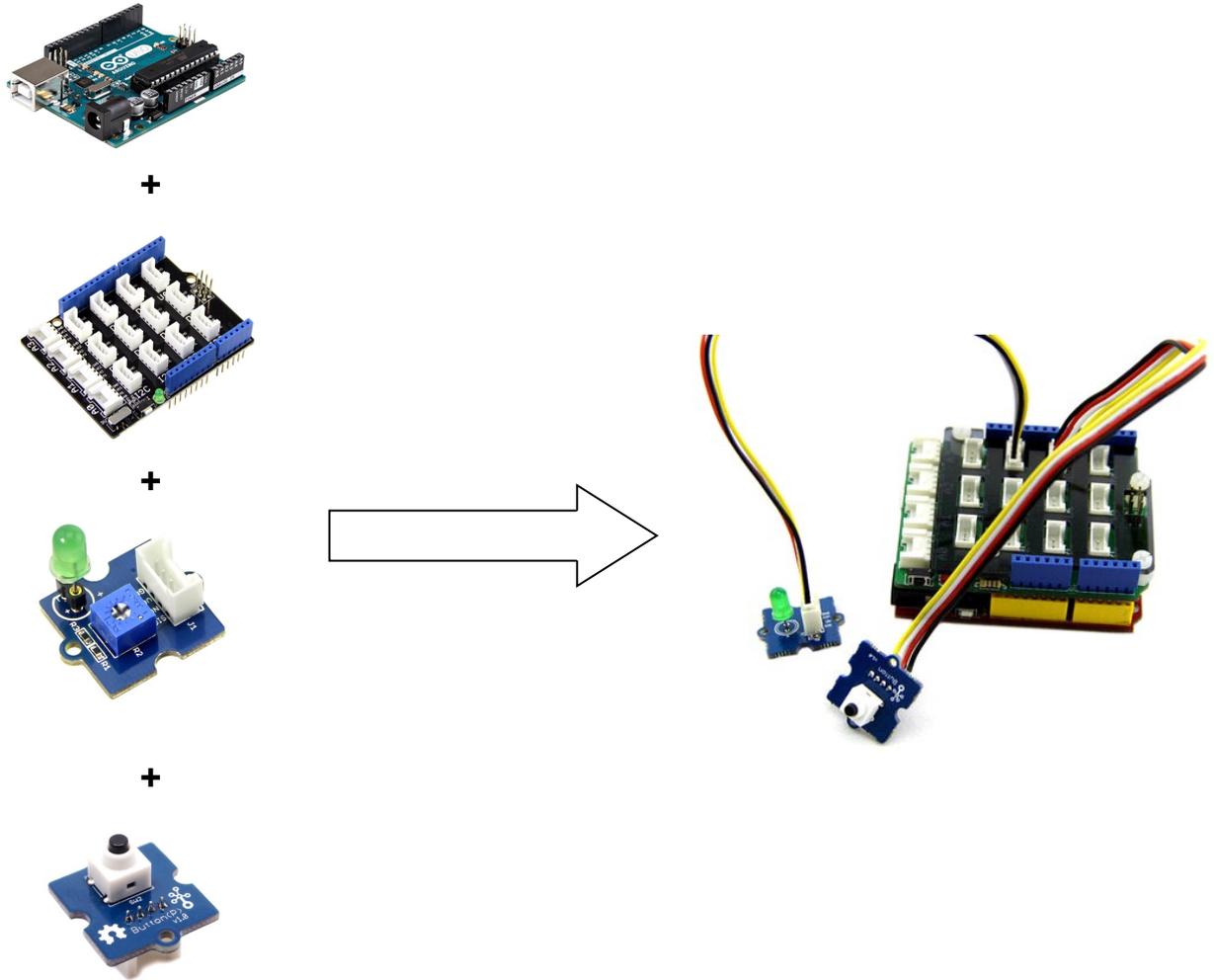
Nous allons utiliser des « modules grove » qui sont des modules électroniques (capteurs ou actionneurs) prêts à l'emploi





Rep	Module Grove	photo	Capteur	Actionneur
1	Buzzer			X
2	Emetteur			X
3	Joystick			X
4	Capteur humidité et température		X	
5	Capteur de Mouvement		X	
6	Capteur de lumière		X	
7	Récepteur		X	
8	Suiveur de ligne			X
9	LED Blanche			X
10	LED Bleue			X
11	LED Verte			X
12	LED Rouge			X
13	Bouton Poussoir		X	
14	Interrupteur		X	
15	Potentiomètre			X
16	Servo moteur + roues			X

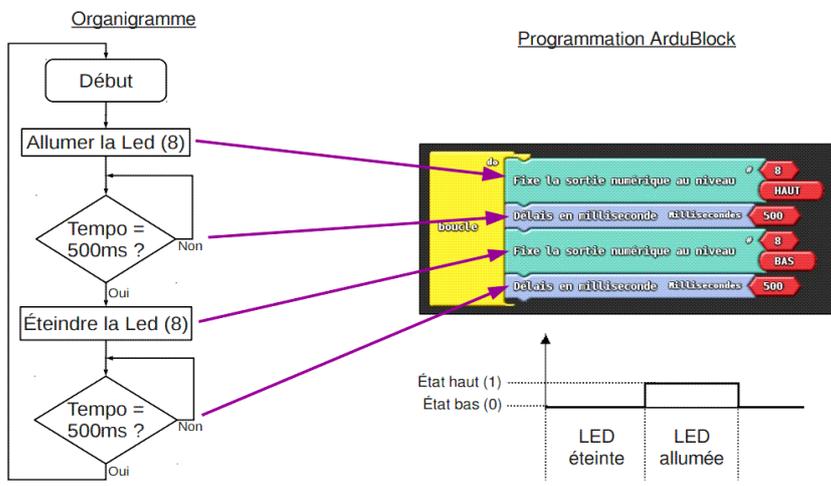
## 4 - ASSEMBLAGE DES ELEMENTS



## 5 - LA PROGRAMMATION

On programme la carte Arduino à l'aide du langage de programmation « ArduBlock ». Ce langage utilise des organigrammes que l'on retranscrit en insérant des blocks à la manière du langage « scratch ».

Voici un exemple d'organigramme associé à un programme :



Le logiciel Arduino IDE transforme ces blocks en langage PYTHON

```
int ledPin = 13;
// initialisation
void setup()
{
  // patte en sortie
  pinMode(ledPin, OUTPUT);
}
// boucle sans fin
void loop()
{
  // on allume la led
  digitalWrite(ledPin, HIGH);
  // on attends une seconde
  delay(1000);
  // on éteint la led
  digitalWrite(ledPin, LOW);
  // on attends une seconde
  delay(1000);
  // et on recommence (ne jamais oublier la dernière attente)
}
```