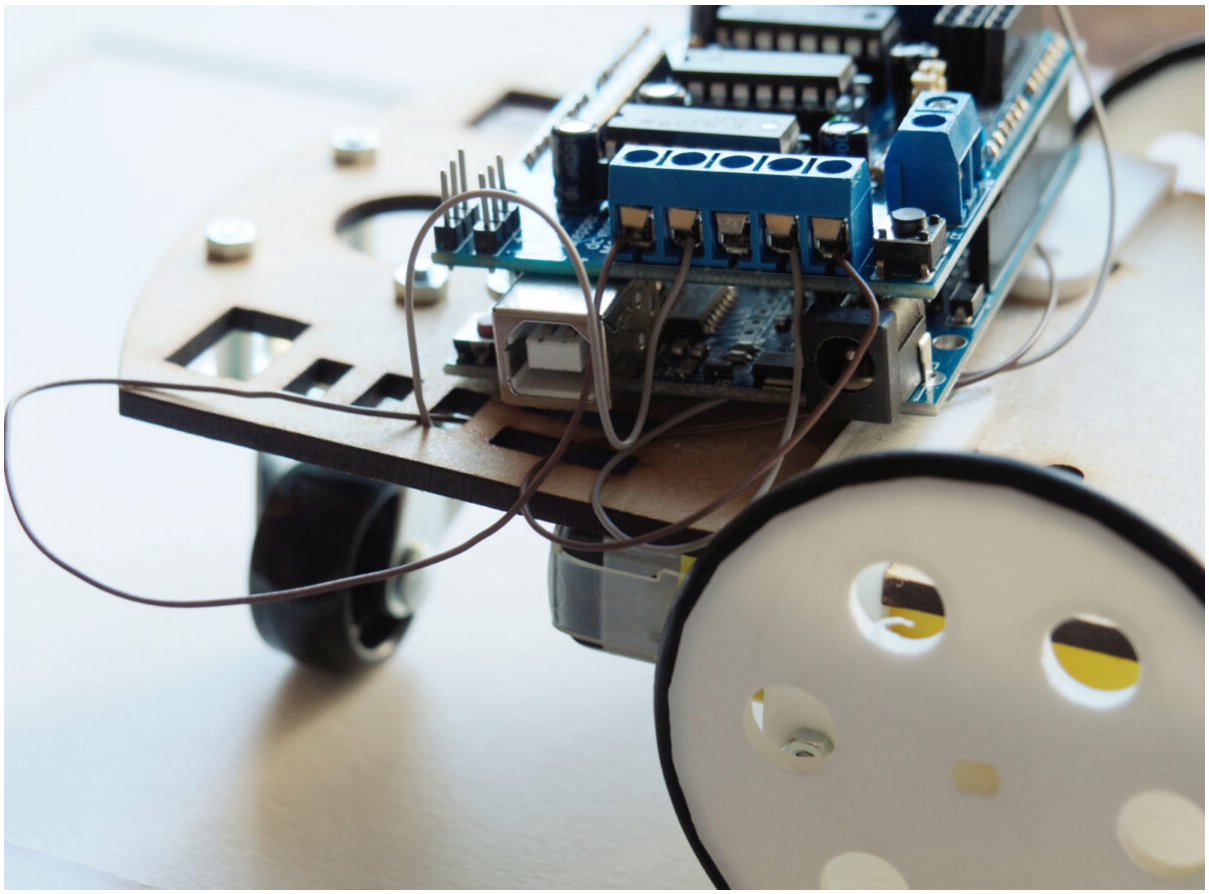
Robot Arduino



Intro

Présentation du LOV (Laboratoire Ouvert villeurbannais).

Présentation du fonctionnement d'un fablab.

Qu'est-ce qu'un fablab ?

Présentation de l'association et présentation des adhésions, des projets en cours.

Cet atelier est un atelier pilote.

C'est le premier atelier expérimental effectué au sein du Lab.

Objectifs

Cet atelier a pour objectif de comprendre les bases de programmation et de l'électronique et de bidouillage à partir d'une carte Arduino. A partir de modèle Open-source, nous développerons un robot. Nous serons amener par la suite à l'augmenter et à l'améliorer.

Public

Débutant n'ayant pas de prérequis en informatique ni en électronique. Enfants et adolescents (8-18ans).

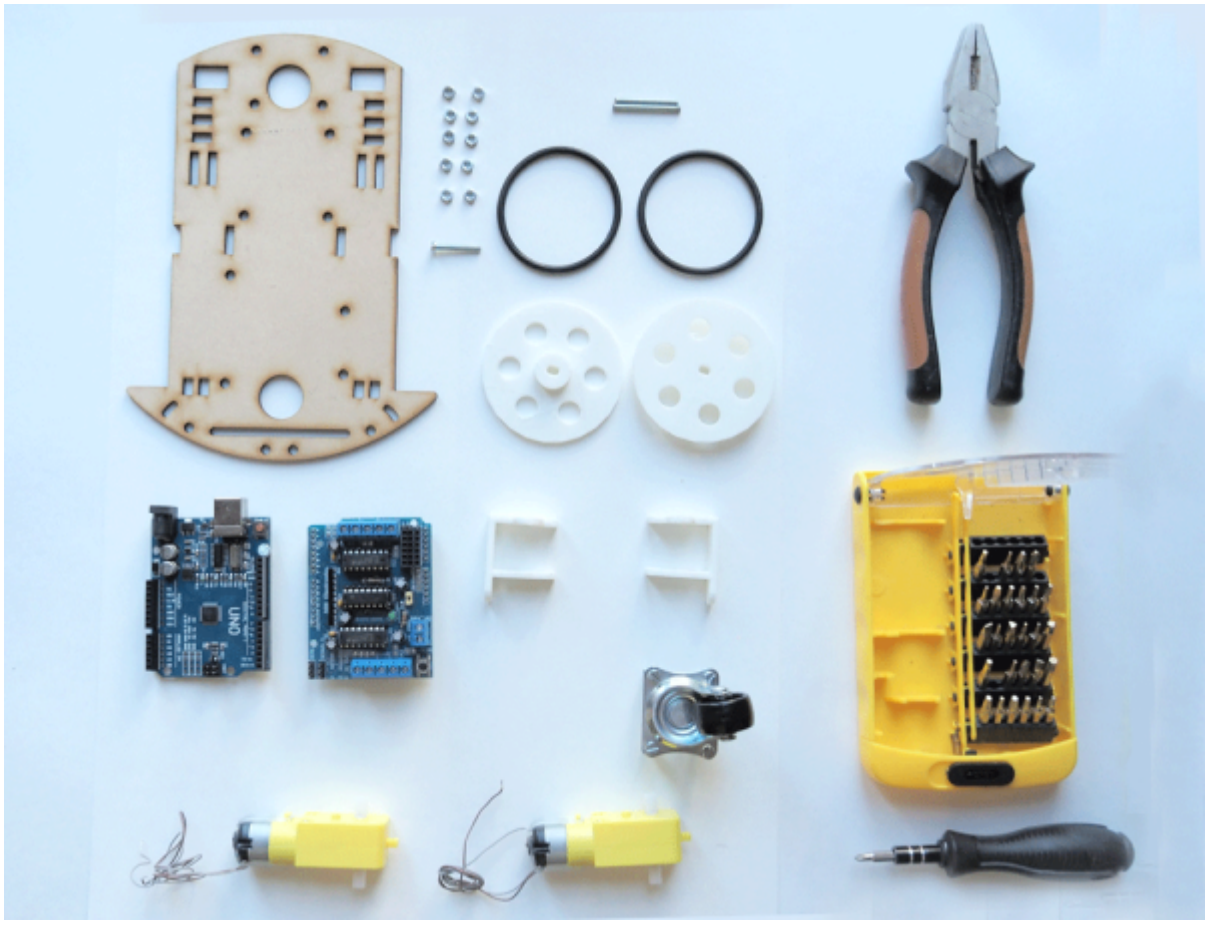
Pré-requis

Logiciel

Arduino : https://www.arduino.cc/

—

Matériel



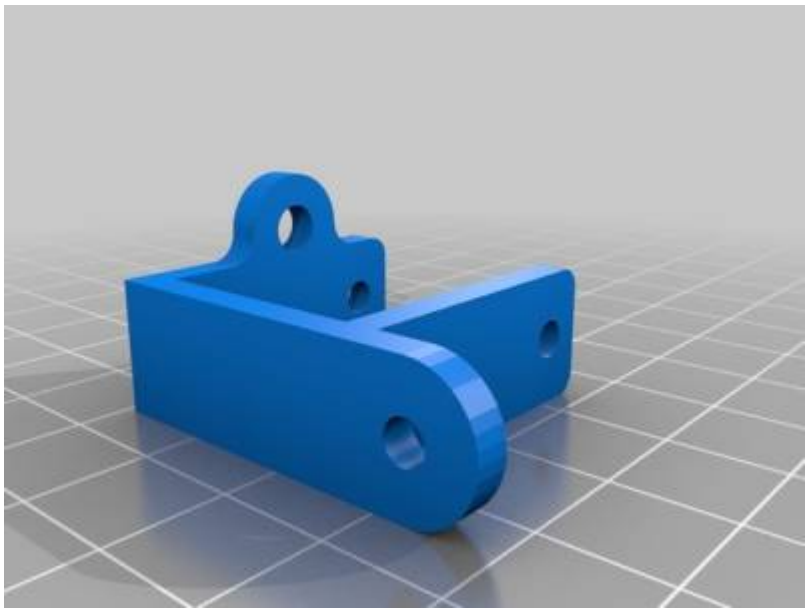
- Carte Arduino etc…
- 1 x Shield
- 1 x BreadBoard
- 1 x LED
- 1 x châssis de voiture (cf. découpage laser)
- 2 x moteur a engrenages
- 2 x roue de voiture (cf. Impression 3d)
- 2 x capteur de vitesse (optionnel)

- 2 x attaches
- 1 x roue universelle
- 1 x tournevis
- 1 x boite de la batterie
- 10 x vis nécessaires et noix

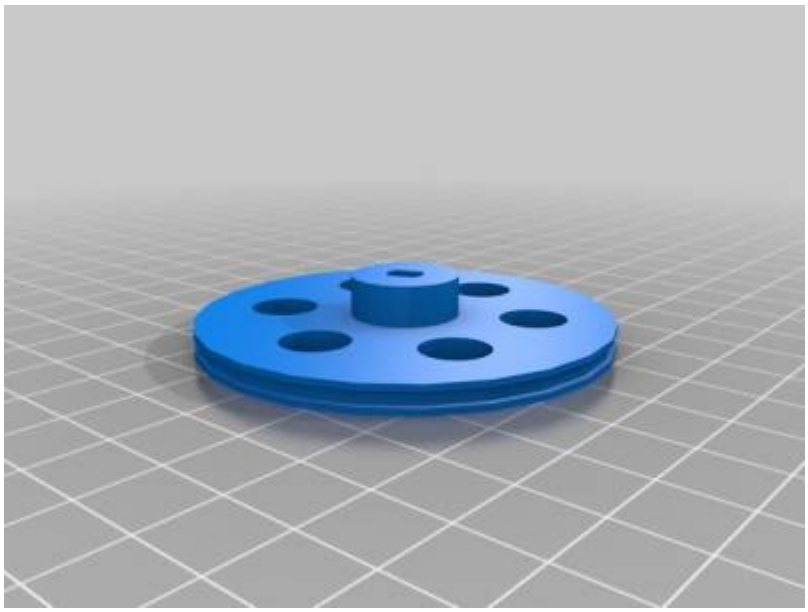
—

Pièces 3d à imprimer DIY

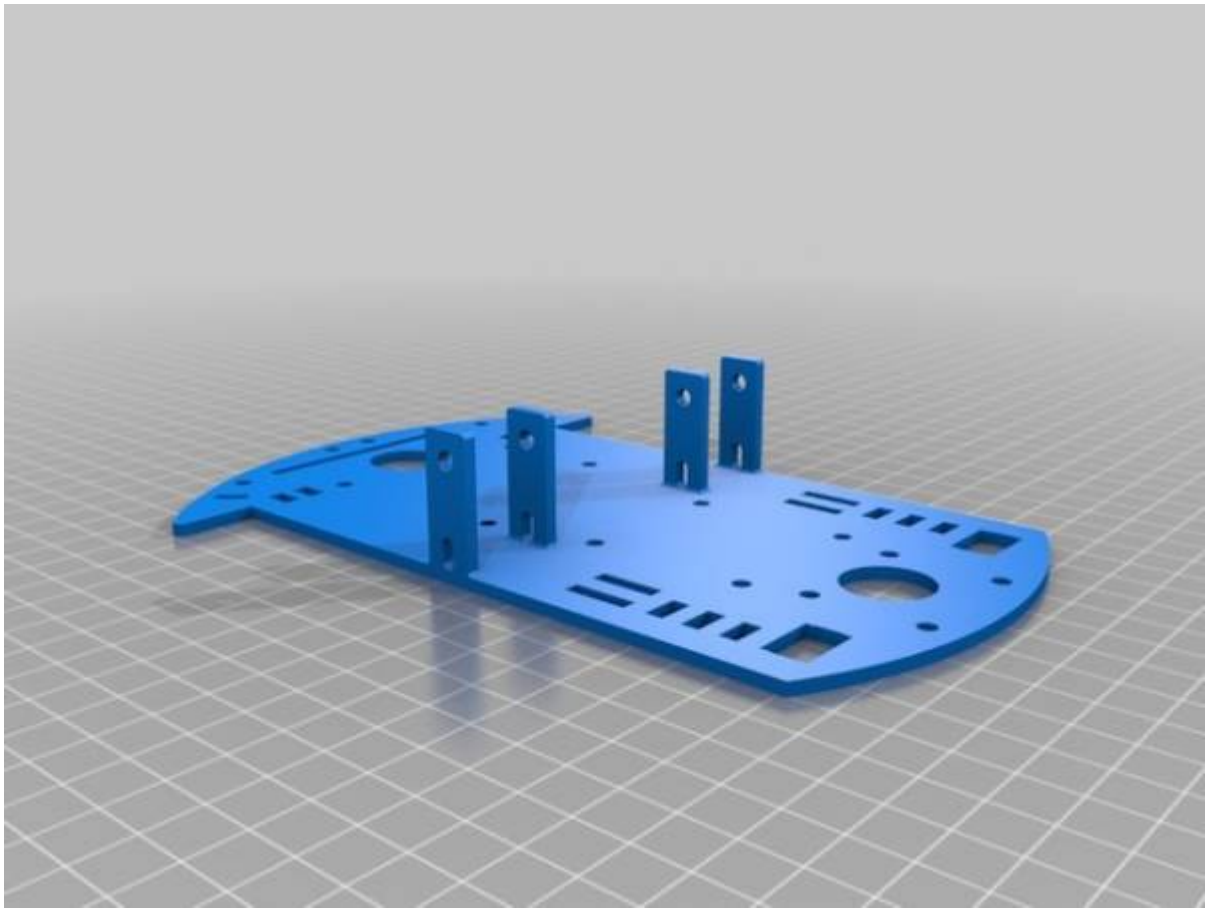
Clips moteur: https://www.thingiverse.com/thing:750963



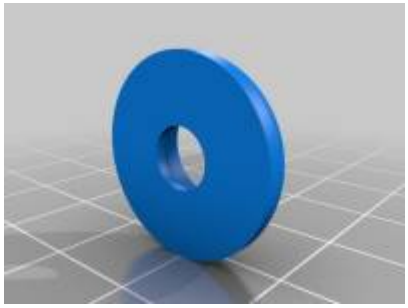
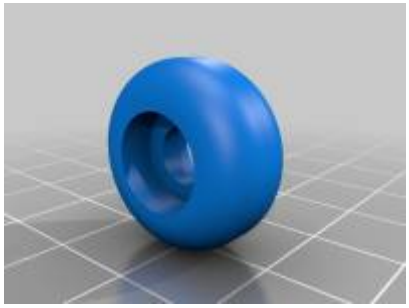
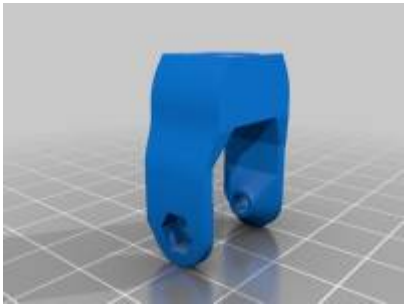
Roues: https://www.thingiverse.com/lluiscolomer/about



Châssis en 3D: https://www.thingiverse.com/thing:1011890



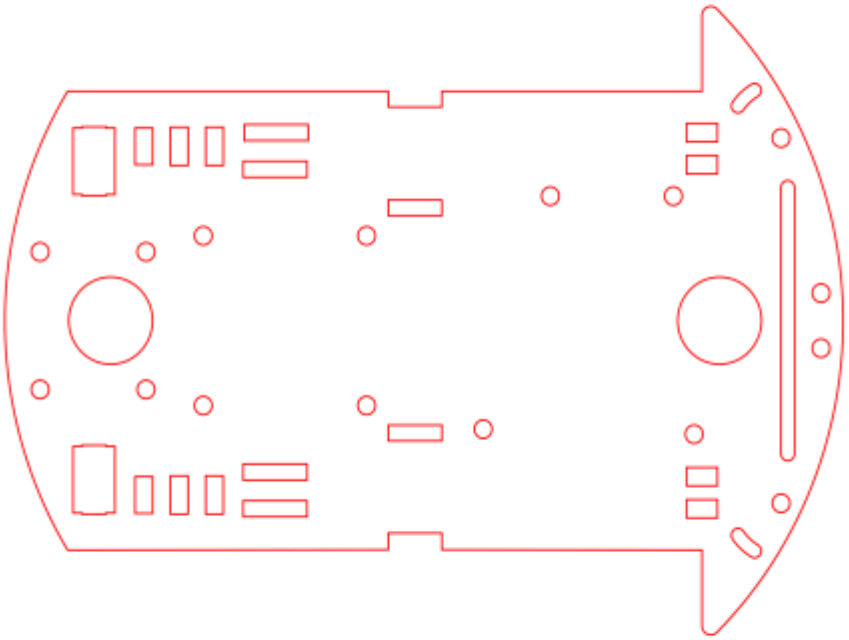
Roue avant

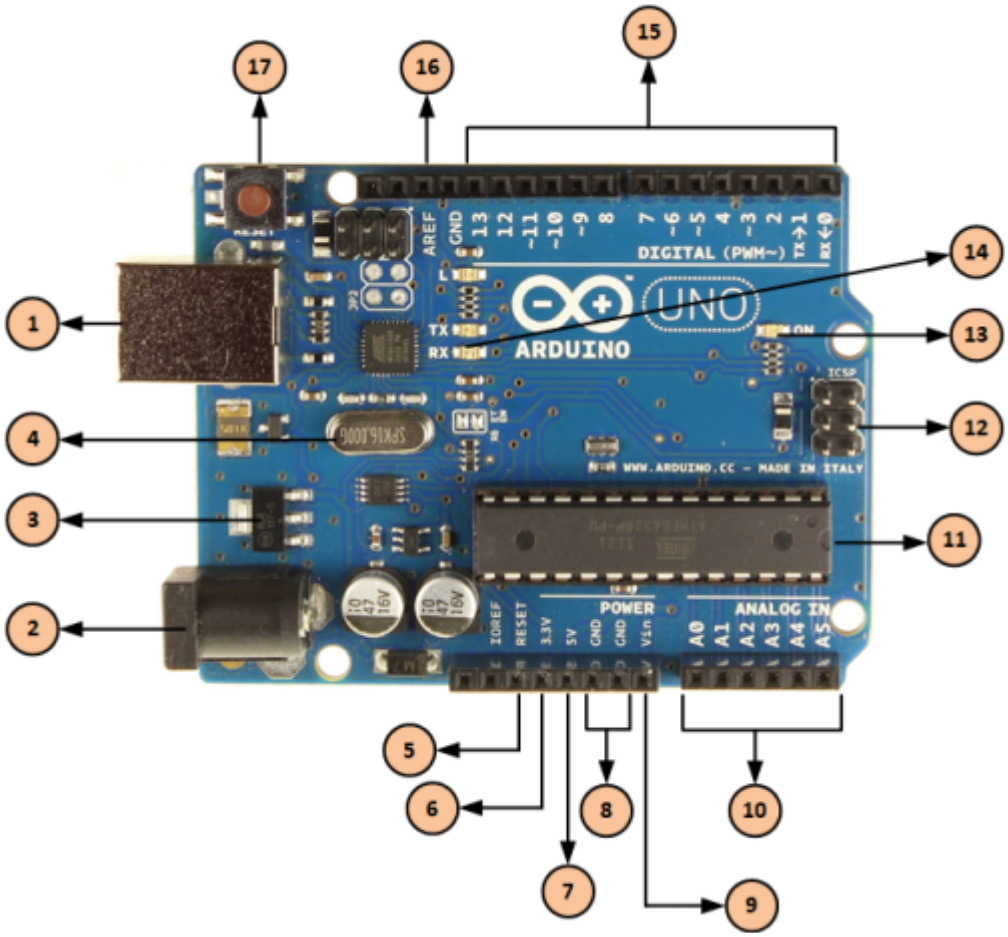
—

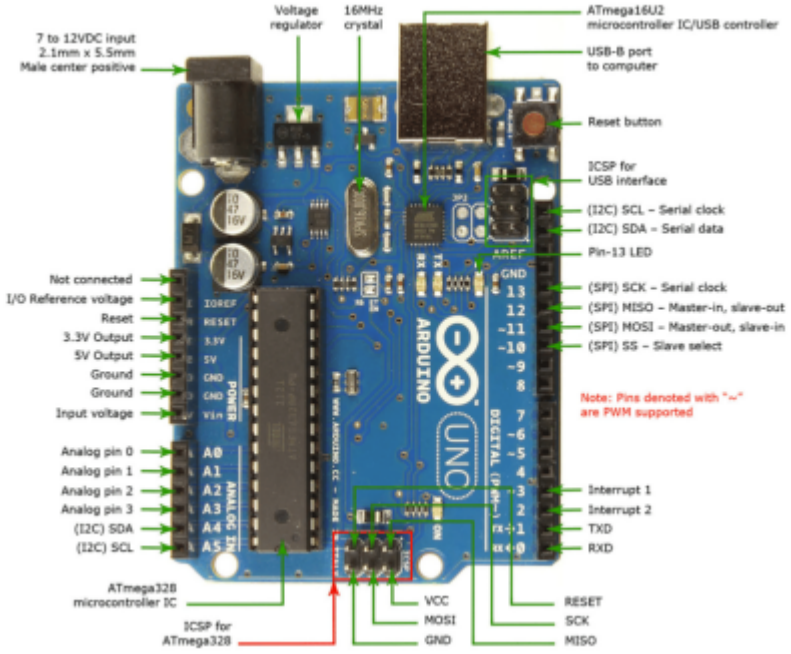
Pièce à découper

Plateau / Châssis



I. Présentation de la carte Arduino





Explications basiques sur l'intérêt de piloter un circuit électrique avec un ordinateur.

Présenter la carte arduino: carte qu'on connecte avec un câble USB à un PC et qui peut piloter ou lire des signaux électriques, pour lire ou écrire ces signaux on utilise le logiciel Arduino IDE.

Présenter Arduino IDE: on écrit du code, le code est transformé et envoyé à la carte qui le mouline. Deux blocs indispensables pour faire un programme: `setup()` quand la carte démarre et `loop()` qui tourne à l'infini.

Exercice pour les participants: lancer l'IDE, faire le montage du tutoriel Blink et le faire fonctionner avec la LED intégrée (sortie 13).

```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

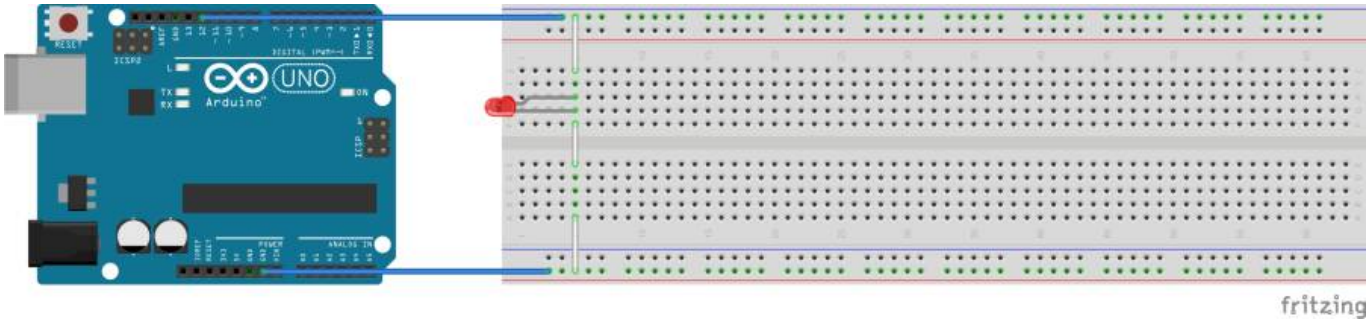
Exercice 2: le refaire avec la plaque à trous avec une LED externe branchée sur une autres sortie(par exemple la pin 12).

II. Notions de base

Exemple : Montrer qu'on peut allumer une LED avec une pile en série avec une résistance. Montrer que ça ne marche pas dans l'autre sens (principe d'une diode) Expliquer les notions de continuité, montrer une plaque à trou, montrer les trous qui sont connectés et ceux qui ne le sont pas.

Exo n°1: Allumer la LED avec une plaque à trous. Laisser les gens chercher comment marche une plaque à trous.

Matériel nécessaire : - Une LED - Une breadboard - Deux mini câbles - Une carte Arduino - Un câble USB ou une pile



III. Un peu plus loin avec la carte Arduino

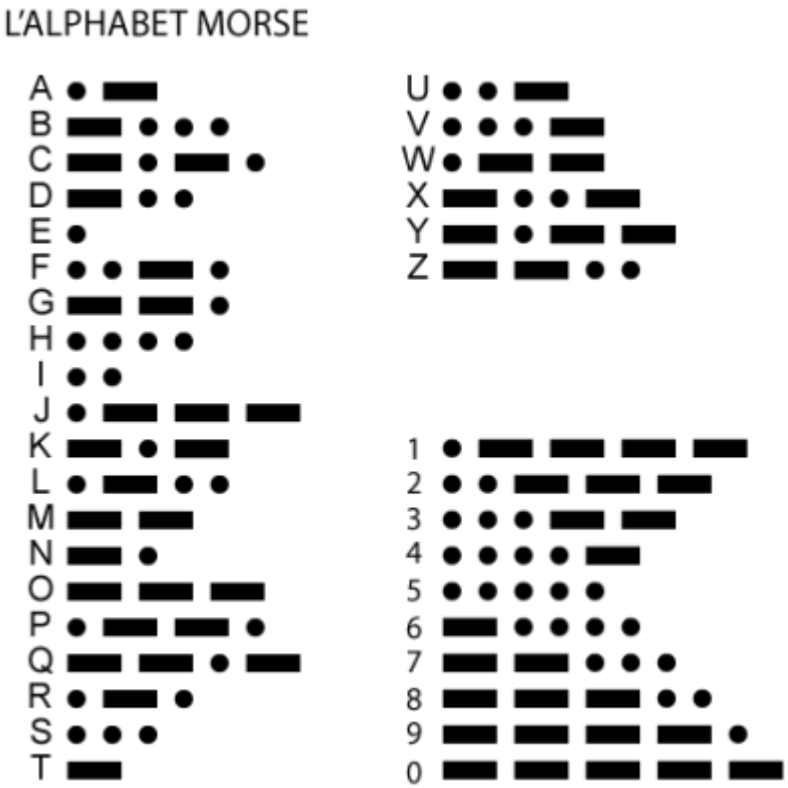
Expliquer les sorties (exemple avec LED pin 12) et les entrées (exemple avec un bouton pin 2).

```
void setup() {
  pinMode(12, OUTPUT);
  pinMode(2, INPUT_PULLUP);
}

void loop() {
  if(digitalRead(2) == LOW) {
```

```
    digitalWrite(12, HIGH);
    delay(50);
    digitalWrite(12, LOW);
    delay(50);
  }
}
```

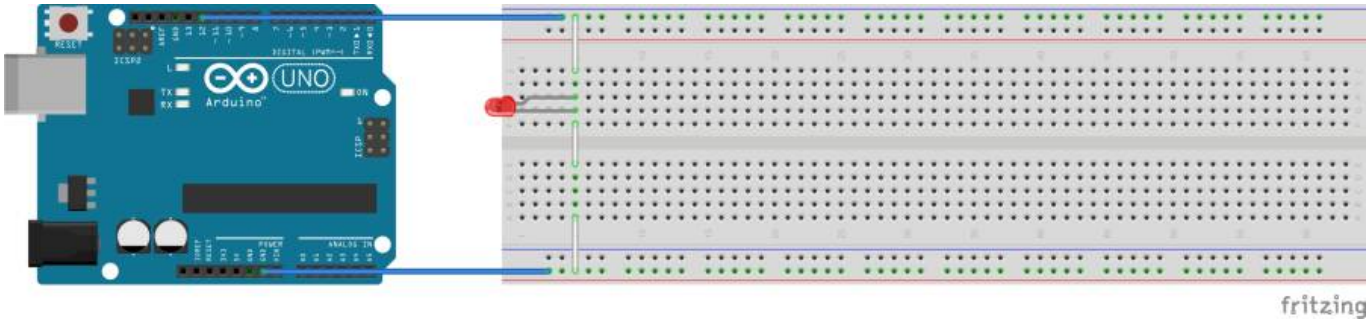
Exercice: faire un programme qui clignote un signal SOS avec une LED quand on appuie sur un bouton.



Aide >

Puis alimenter un moteur directement avec une pile.





V. Présentation du shield pour piloter les moteurs

Expliquer pourquoi on ne peut pas connecter directement un moteur à la carte Arduino.

Démo avec un moteur, avancer puis reculer avec une temporisation.

```
#include <AFMotor.h>

AF_DCMotor motor(1, MOTOR12_64KHZ); // Créer le moteur #2, PWM à 64KHz

void setup() {
  motor.setSpeed(128);     // Mettre la vitesse à 200 (max 255)
}

void loop() {
  motor.run(FORWARD);      // mettre en marche avant (forward)
  delay(1000);

  motor.run(BACKWARD);     // Dans l'autre sens
  delay(1000);

  motor.run(RELEASE);      // Arrêt
  delay(1000);
}
```

Expliquer le principe d'une bibliothèque de fonction (#include) : mode d'emploi pour la carte arduino : elle saura quoi faire des instruction qu'on lui donne pour utiliser les moteurs et leur contrôleur.

Expliquer pourquoi un mécanisme de réduction est nécessaire. Présenter les modules moteurs. Expliquer comment on les utilise pour faire un châssis.

Connecter le moteur 1 à la carte, connecter une pile de 9V et placer le jumper PWR (pour que les moteurs soient alimentés via le 9V de l'Arduino).

Exemple avec des boutons connectés sur A0 et A1:

```
#include <AFMotor.h>

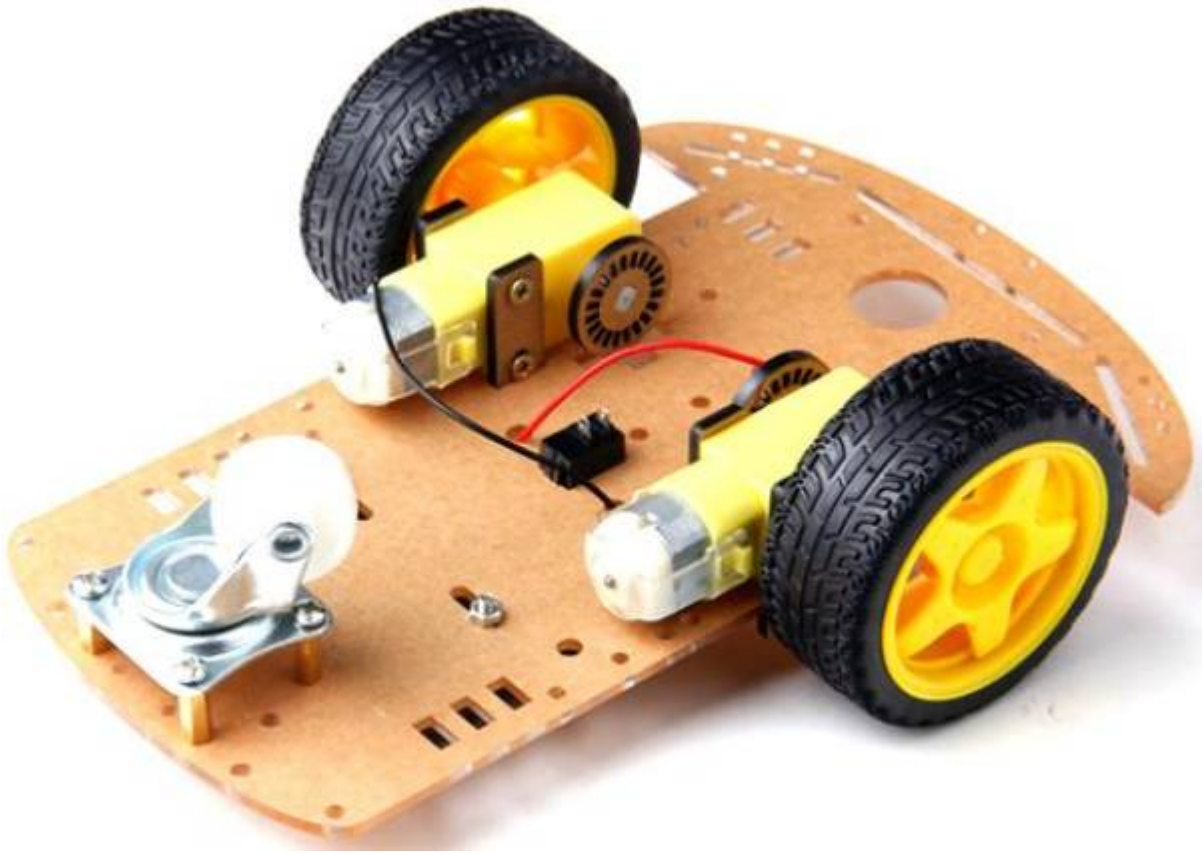
AF_DCMotor motor(1, MOTOR12_64KHZ);

void setup() {
```

```
  motor.setSpeed(128);
  pinMode(A0, INPUT_PULLUP);
  pinMode(A1, INPUT_PULLUP);
}

void loop() {
  if(digitalRead(A0) == LOW){
    motor.run(FORWARD);
    delay(2000);
  } else if(digitalRead(A1) == LOW){
    motor.run(BACKWARD);
    delay(2000);
  } else {
    motor.run(RELEASE);
  }
}
```

VII. Construction du robot



Montage de l'arduino dans le châssis, connexion.

Exercice: faire avancer, reculer, tourner le robot autour d'un obstacle fixe (utiliser des delays pour coder un chemin en dur).

VIII. Récupérer des informations avec le moniteur série

```
#include <AFMotor.h>

AF_DCMotor motor(1, MOTOR12_64KHZ);

void setup() {
  Serial.begin(9600);           // Initialiser la communication série à 9600
bauds
  Serial.println("Test Moteur!");

  motor.setSpeed(128);
  pinMode(A0, INPUT_PULLUP);
  pinMode(A1, INPUT_PULLUP);
}

void loop() {
  if(digitalRead(A0) == LOW){
    Serial.println("en avant");
    motor.run(FORWARD);
    delay(2000);
  } else if(digitalRead(A1) == LOW){
    Serial.println("en arriere");
    motor.run(BACKWARD);
    delay(2000);
  } else {
    motor.run(RELEASE);
  }
}
```

IX. Capteur de luminosité

Démo de résistance photo sensible avec le moniteur série:

```
void setup() {
  Serial.begin(9600);           // Initialiser la communication série à 9600
bauds

  pinMode(A0, INPUT);
  digitalWrite(A0, HIGH);
}

void loop() {
  Serial.println(analogRead(A0));
  delay(1000);
}
```

Exercice: placer deux résistances photo sensibles et faire avancer le robot dans la direction avec le plus de lumière.

X. Capteur de distance à Ultrasons

```
/* Constantes pour les broches */
const byte TRIGGER_PIN = A0; // Broche TRIGGER
const byte ECHO_PIN = A1;    // Broche ECHO

/* Constantes pour le timeout */
const unsigned long MEASURE_TIMEOUT = 25000UL; // 25ms = ~8m à 340m/s

/* Vitesse du son dans l'air en mm/us */
const float SOUND_SPEED = 340.0 / 1000;

void setup() {

  /* Initialise le port série */
  Serial.begin(9600);

  /* Initialise les broches */
  pinMode(TRIGGER_PIN, OUTPUT);
  digitalWrite(TRIGGER_PIN, LOW); // La broche TRIGGER doit être à LOW au
repos
  pinMode(ECHO_PIN, INPUT);
}

void loop() {

  /* 1. Lance une mesure de distance en envoyant une impulsion HIGH de 10µs
sur la broche TRIGGER */
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER_PIN, LOW);

  /* 2. Mesure le temps entre l'envoi de l'impulsion ultrasonique et son
écho (si il existe) */
  long measure = pulseIn(ECHO_PIN, HIGH, MEASURE_TIMEOUT);

  /* 3. Calcul la distance à partir du temps mesuré */
  float distance_mm = measure / 2.0 * SOUND_SPEED;

  /* Affiche les résultats en mm, cm et m */
  Serial.print(F("Distance: "));
  Serial.print(distance_mm);
  Serial.print(F("mm ("));
  Serial.print(distance_mm / 10.0, 2);
  Serial.print(F("cm, "));
  Serial.print(distance_mm / 1000.0, 2);
  Serial.println(F("m)"));

  /* Délai d'attente pour éviter d'afficher trop de résultats à la seconde
*/
```

```
  delay(500);
}
```

XI. Liens utiles

Tuto Blink officiel: https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink/

Datasheet atmega: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

Schéma arduino: https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf

Motor shield: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-motor-shield.pdf

Schéma du motor shield: https://raw.githubusercontent.com/adafruit/Adafruit-Motor-Shield-for-Arduino/master/mshieldv12schem.png

Doc HC-SR04: https://www.carnetdumaker.net/articles/mesurer-une-distance-avec-un-capteur-ultrason-hc-sr04-et-une-carte-arduino-genuino/

Clips moteur: https://www.thingiverse.com/thing:750963

Roues: https://www.thingiverse.com/lluiscolomer/about

Chassis en 3D: https://www.thingiverse.com/thing:1011890

From:
https://labovilleurbanne.fr/dokuwiki/ - **DokuWiki du LOV**

Permanent link:
https://labovilleurbanne.fr/dokuwiki/ateliers:robot_arduino?rev=1690209853

Last update: **2023/07/24 14:44**