

# Orange

# Challenge Ada Lovelace

Initiation à la programmation Arduino

Matthieu Salvat



# Avant de commencer

Il vous faudra télécharger l'environnement de développement (IDE) Arduino sur le site officiel :

<https://www.arduino.cc/en/Main/Software>

Une fois l'IDE téléchargé et installé il vous faudra charger les bibliothèques nécessaires pour l'écran LCD et le moteur pas-à-pas. Pour cela dans l'IDE Arduino allez dans le menu "Croquis/Inclure une bibliothèque/Gérer les bibliothèques" puis cherchez et installez les bibliothèques suivantes :

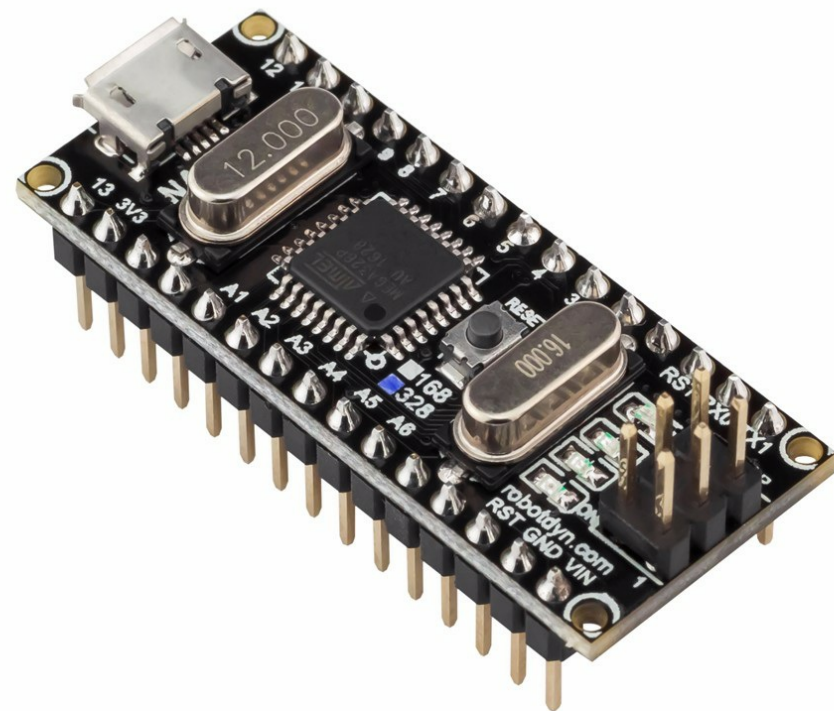
- AccelStepper v1.57.1 de Mike McCauley
- LiquidCrystal I2C v1.1.2 de Frank de Brabander



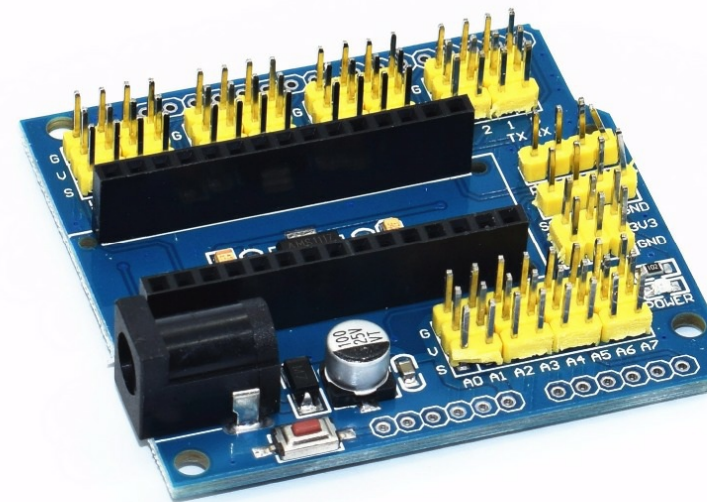
# Arduino

L'Arduino est une carte construite autour d'un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage...), le pilotage d'un robot, de l'informatique embarquée, etc.

RobotDyn



FEIYANG



Ici on le branche sur un "shield" (bouclier) permettant d'avoir une alimentation (+5V, Ground) pour chaque broche afin de faciliter le câblage des composants.

# Branchement de l'Arduino

Pour programmer l'Arduino il suffit de le brancher sur un des ports USB de votre ordinateur. Vous devriez voir apparaître dans le menu "Outils/Port" une nouvelle entrée, sélectionnez là. Vérifiez ensuite que c'est bien un "Arduino Nano" qui est sélectionné dans le menu "Outils/Type de carte".

Tapez le programme suivant :

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("setup");  
}  
int i=0;  
void loop() {  
  Serial.print("loop ");  
  Serial.println(i);  
  i++;  
  delay(1000);  
}
```

Téléversez le programme sur l'Arduino (bouton en forme de flèche en haut de l'IDE). Ouvrez le moniteur série (menu "Outils/Moniteur série" ou via le bouton en haut à droite de l'IDE), vous devriez voir apparaître toutes les secondes une nouvelle ligne.

# La structure d'un programme Arduino

Tous les programmes Arduino doivent respecter cette structure :

```
// Vous pouvez déclarer les variables globales de votre script ici
int entier = 0;
void setup() {
  // Cette fonction sera lancée une seule fois au démarrage de l'Arduino
  // C'est ici que vous initialiserez les entrées et sorties de l'Arduino
  // Typiquement vous pouvez initialiser la liaison série
  // entre l'Arduino et votre ordinateur :
  Serial.begin(9600);
}

void loop() {
  // Cette fonction sera appelée continuellement après le setup
  // Pour déboguer la liaison série est pratique :
  Serial.println("Test");
  // La fonction "delay(X)" permet de mettre en pause
  // le programme pendant X ms
  delay(200);
}
```

A noter : quand vous utilisez la liaison série USB vous ne devez pas utiliser les broches 0 et 1.

# Allumer une DEL

La DEL (ou LED en anglais) est un composant simple s'allumant quand elle reçoit du courant. Nous allons commencer par brancher une DEL sur notre arduino. Pour cela prenez 2 fils.

Les DEL ont des tensions précises de fonctionnement (entre 2.2V et 3V selon leur couleur) mais les broches de l'Arduino fournissent du +5V, il faudra donc réduire l'alimentation des DEL en utilisant une résistance. Pour gagner du temps nous allons brancher directement nos DEL. Cela va réduire leur durée de vie mais dans le cadre de ce TP ça ne sera pas gênant.

Branchez un fil entre la patte courte de la DEL et la broche G du port 2 de l'Arduino

Branchez un fil entre la patte longue de la DEL et la broche S du port 2 de l'Arduino

Téléversez le programme suivant. La DEL devrait clignoter!

Jouez avec la valeur inscrite dans les fonctions delay pour faire clignoter plus ou moins vite la DEL

```
#define LED1 2

void setup() {
  pinMode(LED1, OUTPUT);
}

void loop() {
  digitalWrite(LED1, HIGH);
  delay(500);
  digitalWrite(LED1, LOW);
  delay(500);
}
```



# Écran LCD I2C

Les écrans LCD permettent d'afficher des informations facilement. La version présentée ici utilise une interface I2C (bus informatique) afin de ne pas utiliser un trop grand nombre de ports de l'Arduino pour l'affichage.



```
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
    lcd.init();
    lcd.backlight();
}
int index=0;
void loop() {
    lcd.clear();
    lcd.print("Ligne 1");
    lcd.setCursor(0, 1);
    lcd.print("Ligne 2 : ");
    lcd.print(index++);
    delay(500);
}
```

# Branchement de l'écran LCD I2C

Détachez un groupe de 4 fils :

- Branchez un fil entre la broche **SCL** de l'écran et la broche **SCL** de l'Arduino
- Branchez un fil entre la broche **SDA** de l'écran et la broche **SDA** de l'Arduino
- Branchez un fil entre la broche **GND** de l'écran et la broche **GND** de l'Arduino (sur la même ligne que les broches SDA et SCL)
- Branchez un fil entre la broche **VCC** de l'écran et la broche **5V** de l'Arduino (sur la même ligne que les broches SDA et SCL)

Vérifier le contraste de l'écran : ce dernier doit s'allumer au démarrage. Si vous ne voyez rien, pas de panique c'est peut-être tout simplement parce que le contraste est mal réglé. Pour changer ce paramètre il suffit de tourner le petit potentiomètre derrière l'écran avec un tournevis cruciforme jusqu'à obtenir un bon contraste entre l'affichage des caractères et le fond de l'écran

Entraxes: 75mm et 31mm



# Module Bluetooth

Les modules Bluetooth permettent de communiquer des informations entre votre Arduino et un téléphone mobile ou une tablette.



```
#include <SoftwareSerial.h>
#define RxD 12
#define TxD 13
SoftwareSerial BTSerie(RxD,TxD);
void setup() {
    BTSerie.begin(9600);
    Serial.begin(9600);
}
void loop() {
    BTSerie.println("hello");
    delay(300);
    char c = 0;
    while (BTSerie.available()) {
        c = BTSerie.read();
        Serial.print(c);
    }
}
```

# Branchement du module Bluetooth

Détachez un groupe de 4 fils :

- Branchez un fil entre la broche **GND** du module Bluetooth et la broche **G** du port 12 de l'Arduino
- Branchez un fil entre la broche **+5V** du module Bluetooth et la broche **V** du port 12 de l'Arduino
- Branchez un fil entre la broche **RX** du module Bluetooth et la broche **S** du port 13 de l'Arduino
- Branchez un fil entre la broche **TX** du module Bluetooth et la broche **S** du port 12 de l'Arduino

Quand vous allumerez l'Arduino la première fois le module Bluetooth clignotera, cela veut dire qu'il attend d'être appairé avec un autre appareil. Depuis votre téléphone/tablette cherchez les appareils disponibles et réalisez l'appairement avec le module HC-06 (code PIN 1234 par défaut). Une fois l'appairement réalisé le module Bluetooth ne clignotera plus.

Vous pouvez télécharger une application Android du type "Bluetooth Serial Controller" pour vérifier que le programme précédent fonctionne bien.

Proposition d'exercice :

- Afficher sur l'écran LCD les lignes envoyées par Bluetooth à votre Arduino

# Potentiomètre "slider"

Ce composant permet d'avoir une valeur analogique comprise entre 0 et 1023 selon la position du curseur. Il faut brancher ce type de composant analogique sur une des broches de A0 à A7.

**RobotDyn**



```
int pot = A0;
void setup() {
  pinMode(pot, INPUT);
  Serial.begin(9600);
}

void loop() {
  int val = analogRead(pot);
  Serial.print("val=");
  Serial.println(val);
  int percent = map(val, 0, 1023,
    0, 100);
  Serial.print("percent=");
  Serial.println(percent);
  delay(400);
}
```

# Branchement du potentiomètre

Détachez un groupe de 3 fils et choisissez un des côtés du potentiomètre (A ou B):

- Branchez un fil entre la broche **GND** du potentiomètre et la broche **G** du port A0
- Branchez un fil entre la broche **VCC** du potentiomètre et la broche **V** du port A0
- Branchez un fil entre la broche **Out** du potentiomètre et la broche **S** du port A0

Téléversez le programme précédent et vérifiez que les valeurs s'affichent bien sur le moniteur série.

Propositions d'exercices :

- Afficher la valeur du potentiomètre sur l'écran LCD
- Faire varier le temps de clignotement d'une LED avec le potentiomètre
- Brancher votre deuxième potentiomètre sur la broche A1 de l'Arduino

entraxes : 60mm et 16mm

# Servomoteur

Un servomoteur est un petit moteur pouvant tourner précisément sur 180°.



```
#include <Servo.h>
Servo myservo;

void setup() {
  myservo.attach(3);
  myservo.write(0);
}
int deg = 0;
void loop() {
  deg = deg + 2;
  if (deg >= 180) {
    deg = 0;
  }
  myservo.write(deg);
  delay(10);
}
```

# Branchement du servomoteur

Branchez directement la prise du servomoteur sur le port 3 de l'Arduino en faisant attention à ce que le fil marron soit sur la broche **G** du port 3, le fil rouge sur la broche **V** et le fil orange sur la broche **S**

Entraxe: 27mm

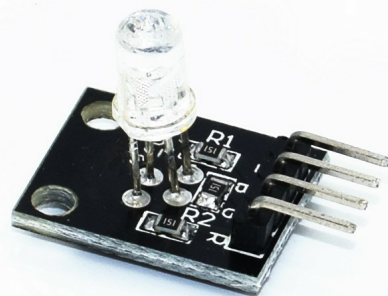
Proposition d'exercice :

- Faire tourner le servomoteur à partir de la valeur d'un potentiomètre (utilisation de la fonction [map](#) pour convertir les valeurs d'entrées du potentiomètre - de 0 à 1023 - en une valeur en degré - de 0 à 180)



# DEL RVB

Une DEL RVB (Rouge, Vert, Bleu ou RGB en anglais pour Red, Green, Blue) permet de choisir la couleur de la lumière. Chaque composante de la lumière (rouge, verte et bleue) peut prendre une valeur de 0 à 255.



```
int redpin=11;
int bluepin=10;
int greenpin=9;
void setup() {
  pinMode(redpin, OUTPUT);
  pinMode(bluepin, OUTPUT);
  pinMode(greenpin, OUTPUT);
  randomSeed(analogRead(0));
}
void loop() {
  analogWrite(redpin,
              random(255));
  analogWrite(bluepin,
              random(255));
  analogWrite(greenpin,
              random(255));
  delay(500);
}
```

# Montager de la DEL RVB

Il faut brancher les broches R,G et B de la DEL sur des ports PWM de l'Arduino (typiquement les port 9, 10 et 11). Détachez un groupe de 4 fils :

- Branchez un fil entre la broche **G** de la DEL sur la broche **S** du port 9
- Branchez un fil entre la broche **B** de la DEL sur la broche **S** du port 10
- Branchez un fil entre la broche **R** de la DEL sur la broche **S** du port 11
- Branchez un fil entre la broche - de la DEL sur la broche **G** du port 9, 10 ou 11

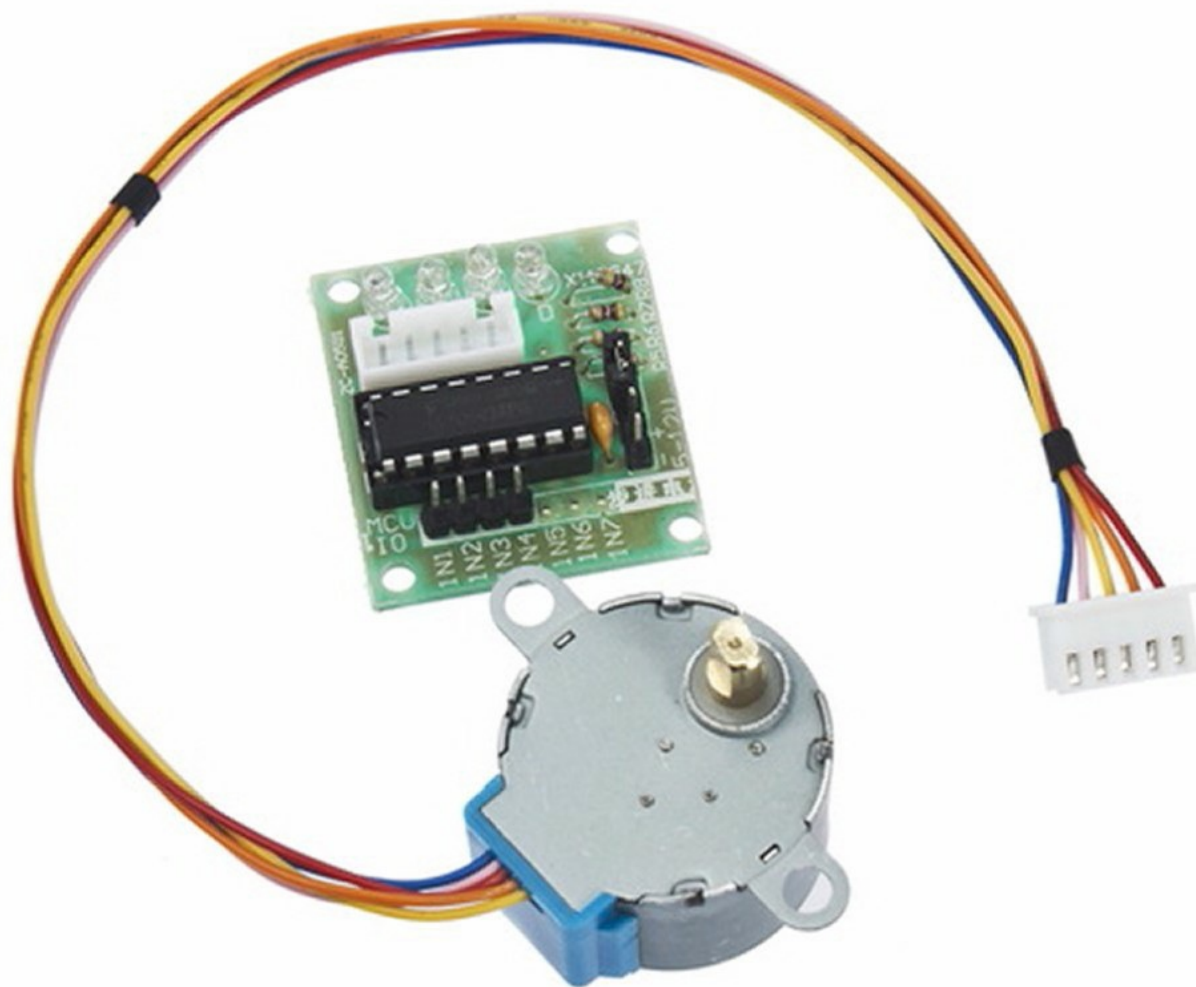
Propositions d'exercices :

- Faire varier la lumière de façon plus naturelle
- Utiliser 3 potentiomètres pour fixer la valeur de chaque couleur de la DEL

Entraxe: 10mm

# Moteur pas-à-pas

Un moteur pas-à-pas permet de gérer des rotations précises ([documentation](#))



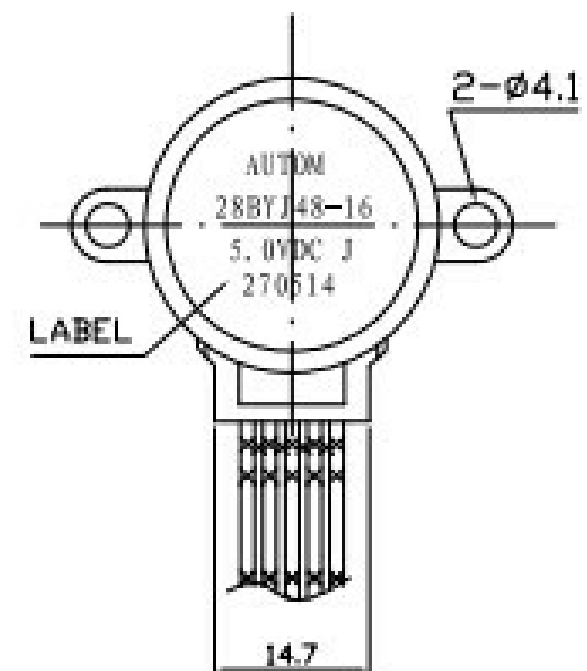
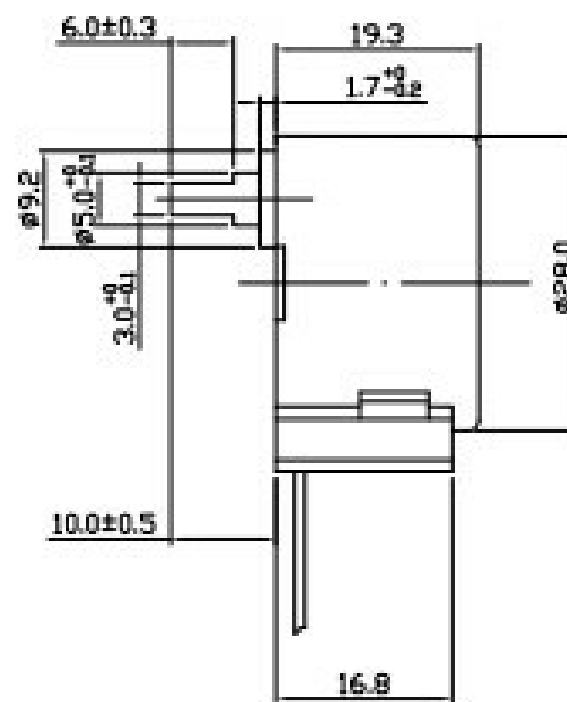
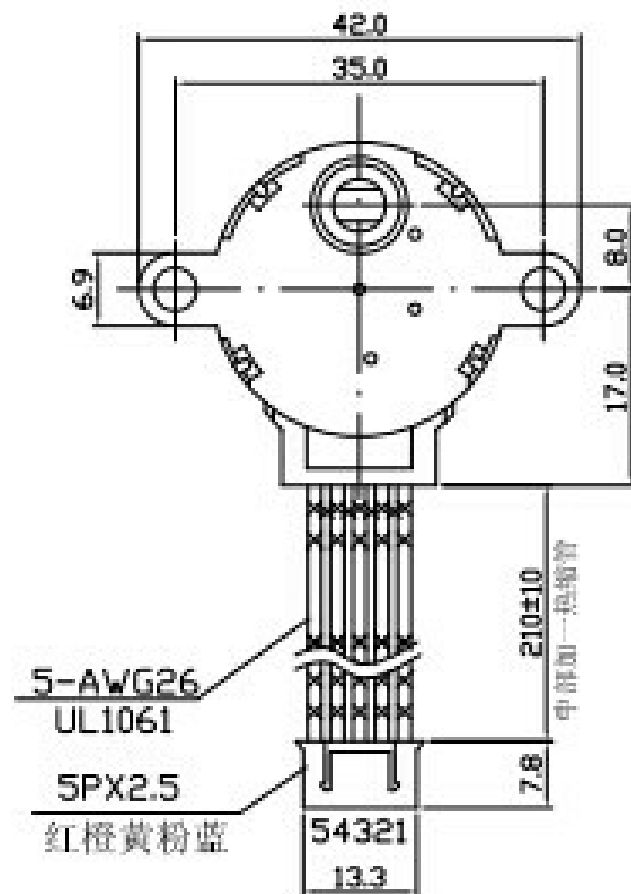
```
#include <AccelStepper.h>
#define motorPin1 4
#define motorPin2 5
#define motorPin3 6
#define motorPin4 7
AccelStepper stepper(
    AccelStepper::HALF4WIRE,
    motorPin1, motorPin3,
    motorPin2, motorPin4);

void setup() {
    stepper.setMaxSpeed(1000);
    stepper.setAcceleration(100.0);
    stepper.enableOutputs();
}
void loop() {
    stepper.runToNewPosition(0);
    stepper.runToNewPosition(1000);
}
```

# Branchement d'un moteur pas-à-pas

Branchez le cable du moteur pas-à-pas sur la carte de contrôle. Détachez un groupe de 4 fils pour les broches **IN** et un groupe de 2 fils pour les broches - et +):

- Branchez un fil entre la broche **IN1** du contrôleur et la broche **S** du port **4** de l'Arduino
- Branchez un fil entre la broche **IN2** du contrôleur et la broche **S** du port **5** de l'Arduino
- Branchez un fil entre la broche **IN3** du contrôleur et la broche **S** du port **6** de l'Arduino
- Branchez un fil entre la broche **IN4** du contrôleur et la broche **S** du port **7** de l'Arduino
- Branchez un fil entre la broche - du contrôleur et la broche **G** du port **7** de l'Arduino
- Branchez un fil entre la broche + du contrôleur et la broche **V** du port **7** de l'Arduino



# Mode non-bloquant

Dans le code précédent à chaque appel à la fonction ***runToNewPosition*** le programme attend la fin de la rotation du moteur pour exécuter la ligne de code suivante. Voici ici un exemple de code permettant de faire tourner le moteur sans bloquer le programme :

```
#include <AccelStepper.h>
#define motorPin1 4
#define motorPin2 5
#define motorPin3 6
#define motorPin4 7
AccelStepper stepper(
    AccelStepper::HALF4WIRE,
    motorPin1, motorPin3,
    motorPin2, motorPin4);

void setup() {
    stepper.setMaxSpeed(1000);
    stepper.move(1);
    Serial.begin(9600);
}
```

```
void loop() {
    if (isCommandTerminated()) {
        Serial.println("fin");
        delay(500);
        stepper.move(300);
    }
}

boolean isCommandTerminated() {
    stepper.setSpeed(1000);
    int d = stepper.distanceToGo();
    stepper.runSpeedToPosition();
    if (d == 0) {
        return true;
    } else {
        return false;
    }
}
```

# D'autres composants sympas pour Arduino

- Un buzzer
- Un lecteur de MP3
- Des rubans à LED
- Des capteurs de présence
- Des capteurs de température et d'humidité
- ...

Où commander : [aliexpress.com](https://www.aliexpress.com), [gearbest.com](https://www.gearbest.com) ... Faites des recherches puis triez les par nombre de commandes, choisissez ensuite les vendeurs avec les meilleurs notes. Il faut compter entre 3 et 6 semaines pour la livraison.