

# Systemes embarqués

Introduction à la programmation microcontrôleurs  
et à l'impression 3D.



**Pape Abdoulaye BARRO, Ph.D,**

.....  
**Enseignant-chercheur**

UFR des Sciences et Technologies  
Département Informatique

.....  
E-LabTP, Laboratoire des TP à Distance, UFR-SET,  
Marconi-Lab, Laboratoire de Télécommunications, ICTP, Italie

.....  
**Email:** [pape.abdoulaye.barro@gmail.com](mailto:pape.abdoulaye.barro@gmail.com)

- ❑ Généralités
- ❑ Architecture et familles de microcontrôleur
- ❑ Capteurs et actionneurs
- ❑ Programmation des microcontrôleurs
  - ❑ **Programmation sous Arduino**
    - ❑ **Etudes de cas**
  - ❑ Programmation sous Raspberry Pi
    - ❑ Etudes de cas
- ❑ Initiation à l'impression 3D

# PLAN

The background of the slide features a series of thin, vertical, light blue lines of varying heights and positions, creating a textured, rain-like effect. A solid, medium-blue horizontal bar spans the width of the slide, positioned in the lower half. The text 'Etudes de cas: Arduino' is centered within this bar.

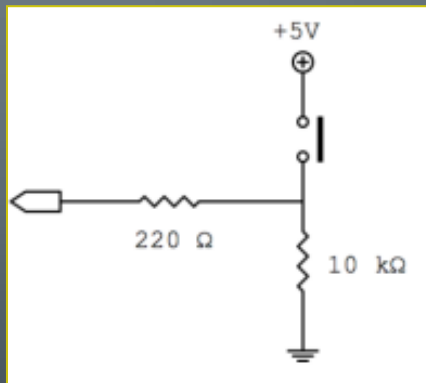
# Etudes de cas: Arduino

# Etude de cas: Arduino

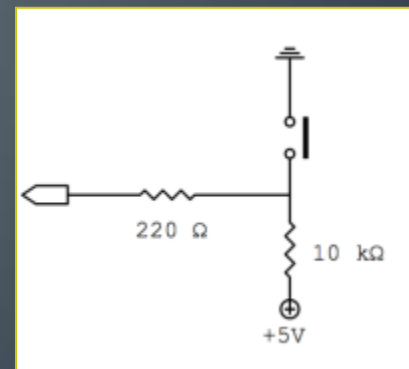
## Bouton poussoir et Led

Mécaniquement, lorsqu'on appuie ou relâche un bouton-poussoir, il y a toujours des rebonds car le contact n'est jamais instantané ou ni parfait. Ces phénomènes sont des parasites qui peuvent tromper l'Arduino.

- ❑ Le signal n'est pas forcément clair: le bouton poussoir peut se comporter comme une antenne dans un environnement pollué par des ondes électromagnétiques et va donc générer un courant dans le circuit qui peut être interprété comme un signal (phénomène d'induction).
- ❑ Pour remédier à cela, il suffit d'utiliser des résistances pour effectuer des montages en **Pull-Down** ou en **Pull-Up**.



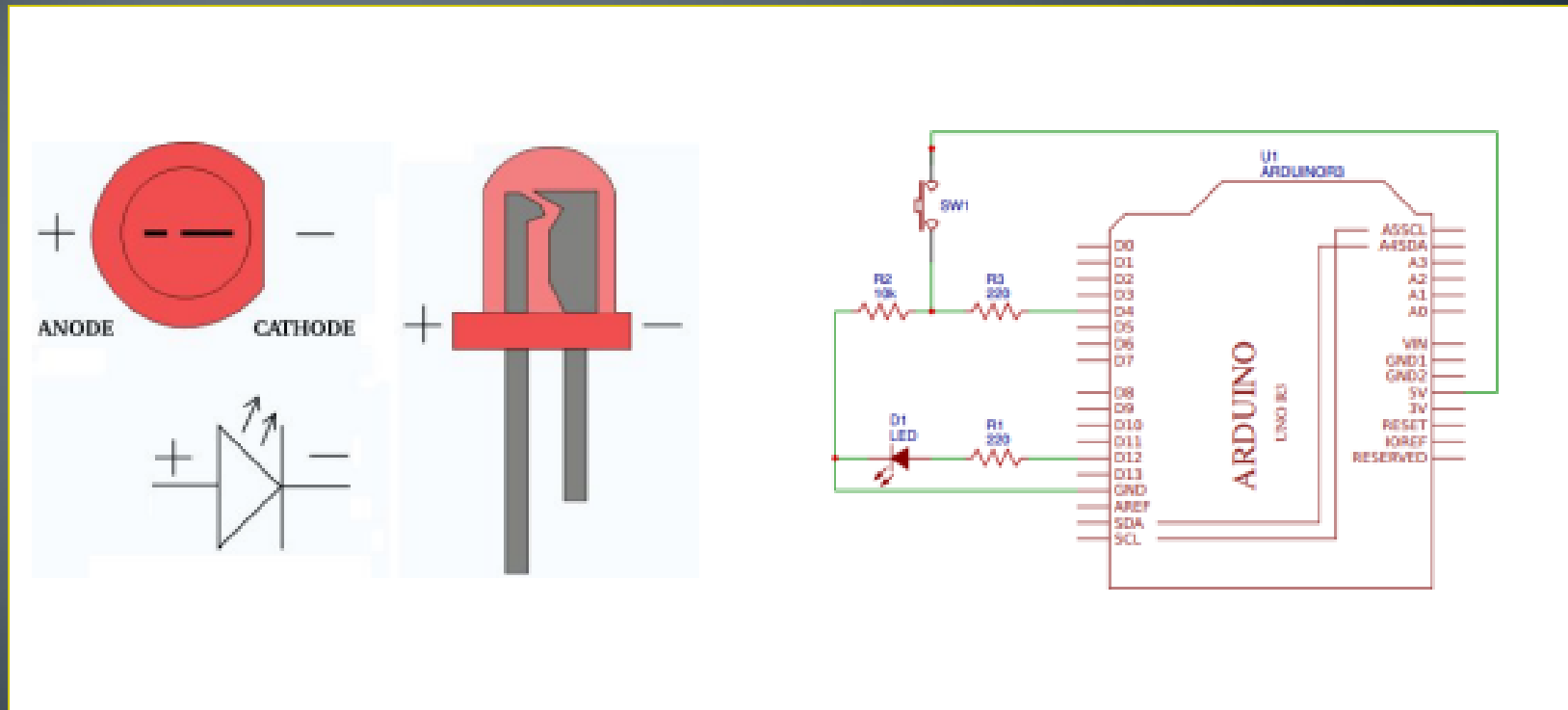
Résistance pull down par défaut



Résistance pull up par défaut

# Etude de cas: Arduino

- ❑ Utiliser un bouton poussoir pour allumer/éteindre une led;

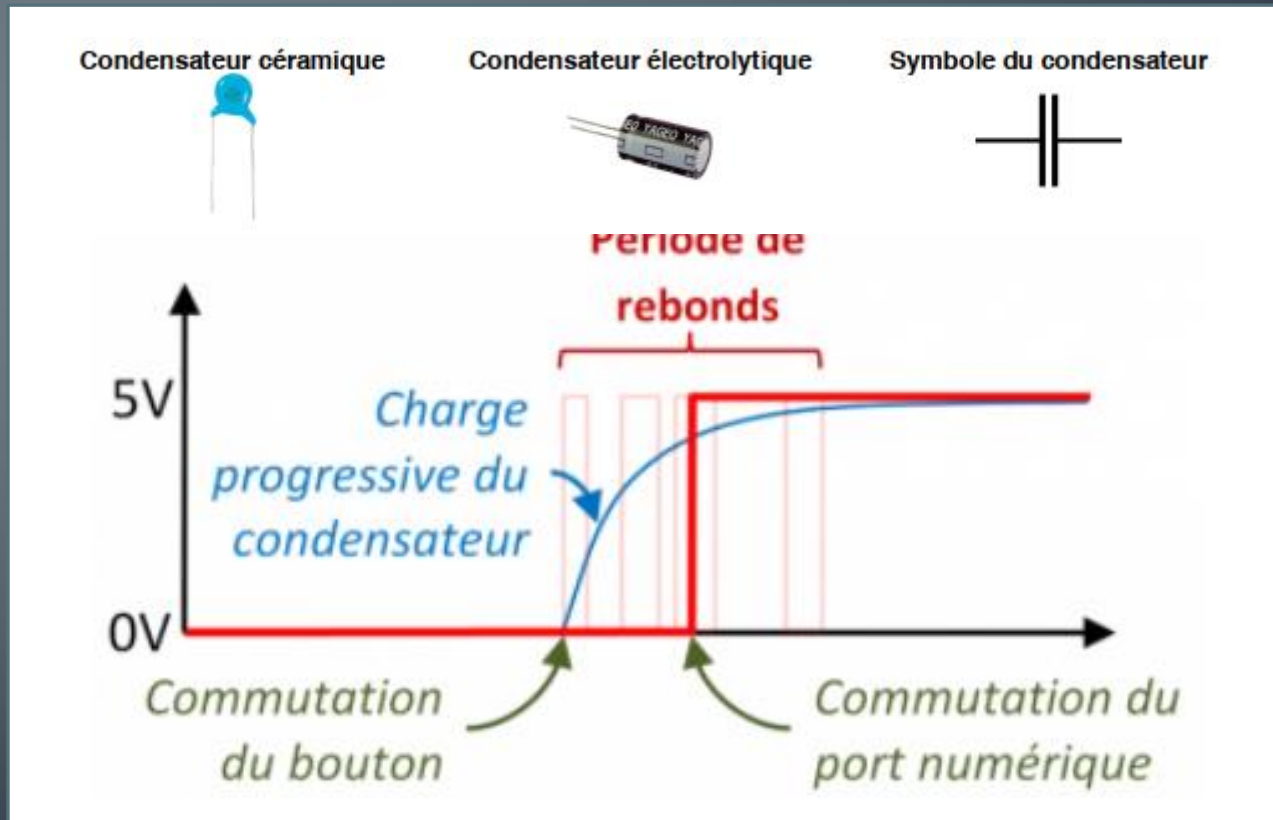


- ❑ Ajouter un Buzzer pour capter le clic sur le bouton

# Etude de cas: Arduino

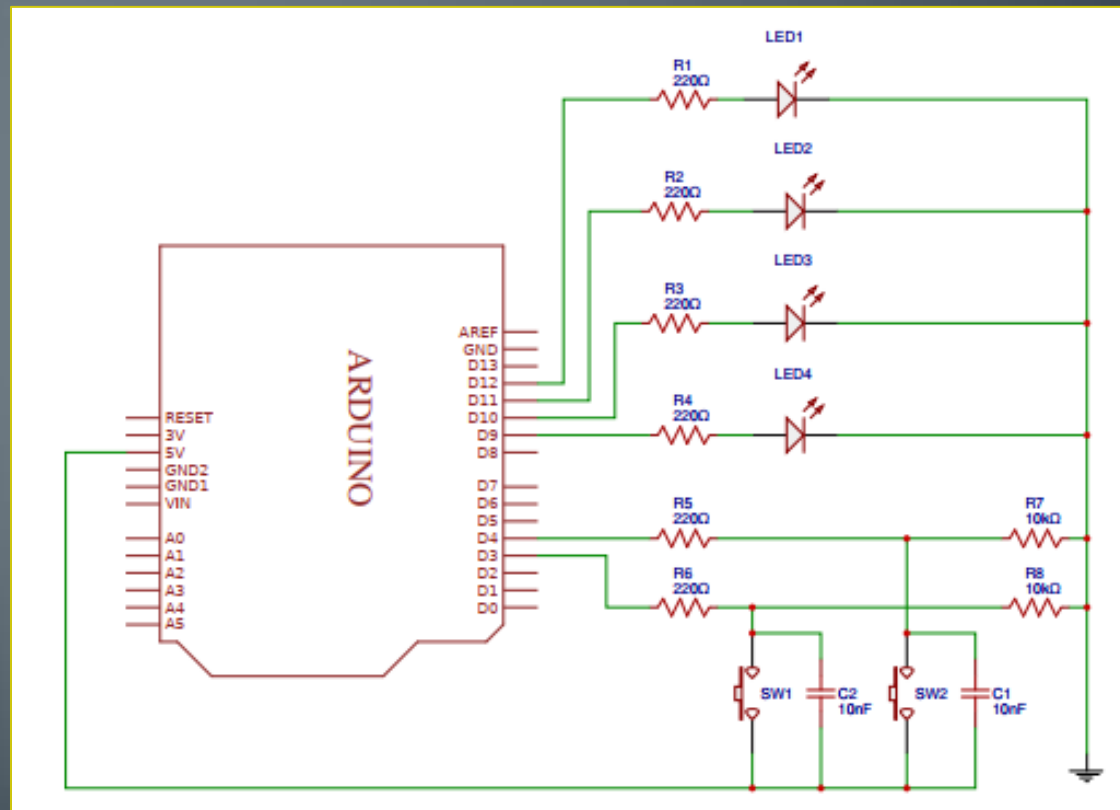
## Utilisation d'un condensateur pour déparasiter

**Constat** : Malgré des pull-down, les boutons poussoirs sont peu précis. Il y a moyen de encore déparasiter et d'absorber ces rebonds en montant en parallèle du bouton-poussoir, un condensateur de faible capacité (ex : 10nF).



# Etude de cas: Arduino

- ❑ Faire un bargraphe de 4 LED avec deux boutons poussoirs. L'un d'eux servira à augmenter la valeur du bargraphe, tandis que l'autre servira à la diminuer.

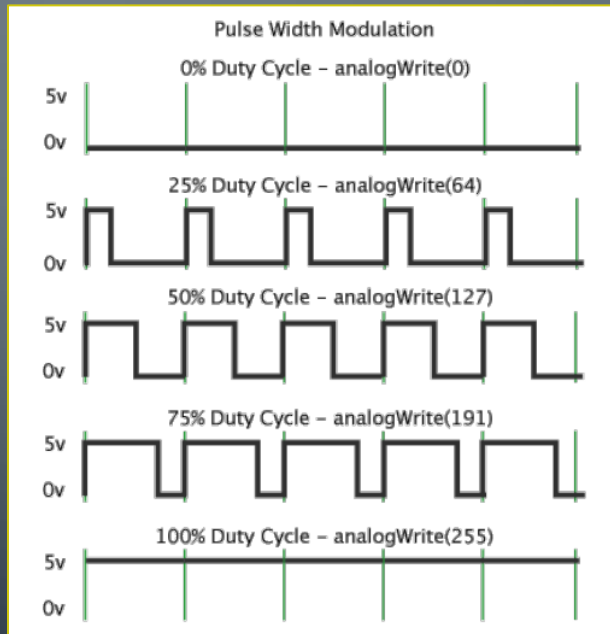


# Etude de cas: Arduino

## Utilisation de la modulation PWM

**PWM** pour Pulse Width Modulation (modulation de largeur d'impulsion en français) vous permet de faire varier les périodes haute (on) et basse (off) des broches à une fréquence élevée.

❑ **Exemple:** Pour un cycle de 25% en position haute et 75% en position basse, la LED sera moins lumineuse par rapport à un cycle de 50% sur 50% (voir quelques phases ci-dessous).



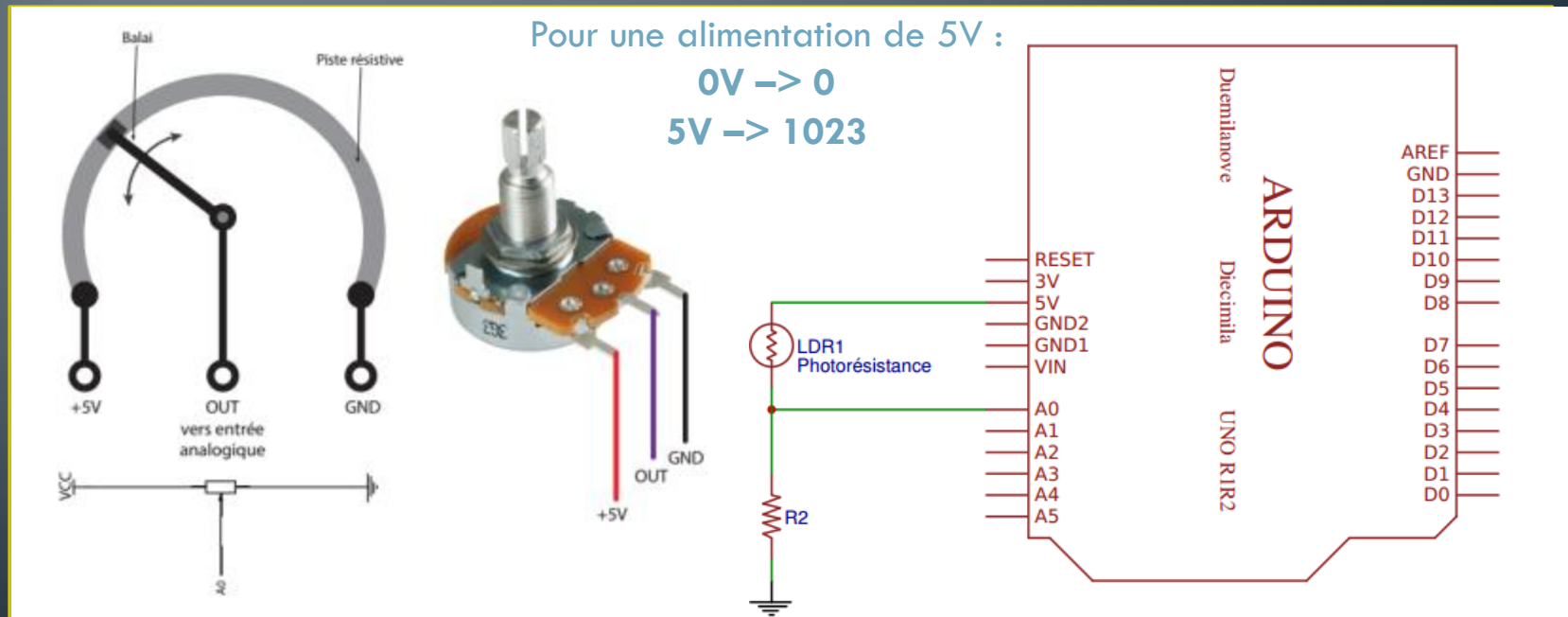
- ❑ Les broches capables de prendre en charge la fonction PWM sont identifiées par un (~) et pour l'utiliser, on utilise AnalogWrite au lieu de l'instruction DigitalWrite. Le rapport cyclique n'est pas donné de 0 à 100 mais de 0 à 255.
- ❑ Utilisez un bouton pour faire varier la luminosité d'une LED en 5 clics:
  - ❑ Clique 1 : 25% en position haute ;
  - ❑ Clique 2 : 50% en position haute ;
  - ❑ Clique 3 : 75% en position haute ;
  - ❑ Clique 4 : 100% en position haute ;
  - ❑ Clique 5 : 0% en position haute.



# Etude de cas: Arduino

## Utilisation du potentiomètre

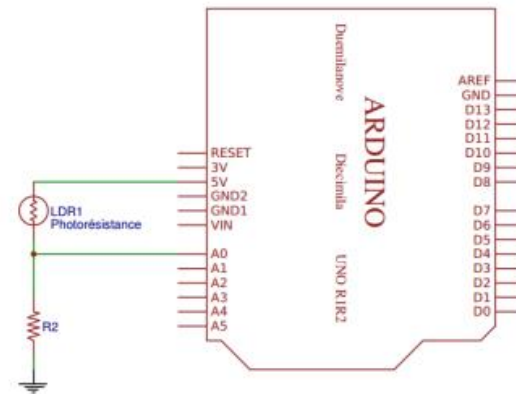
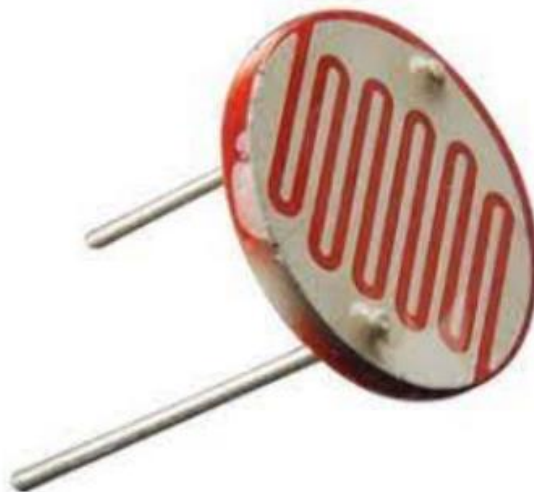
Un potentiomètre est une résistance variable. Il permet donc de modifier la tension d'un circuit. Il est analogique, ce qui veut dire qu'il peut être branché à une broche analogique (A0 à A5) de l'Arduino. Ce qui veut dire aussi qu'on va récupérer les valeurs comprises entre 0 et 1023 (car l'entrée analogique de l'Arduino Uno est codé sur 10 bits c'est-à-dire  $2^{10} = 1024$  valeurs).



# Etude de cas: Arduino

## Utilisation d'un capteur analogique

Les capteurs analogique produisent en sortie une grandeur physique (tension ou courant) dont la valeur est proportionnelle à la grandeur physique mesurée. Il délivre ainsi en sortie une infinité de valeurs continues. En Arduino, les capteur analogique sont lus via les ports analogiques (A0, A1, A2, A3, A4 et A5) grâce à la fonction ***analogRead()*** et les valeurs retournées sont comprises entre 0 et 1023. Ces valeurs sont les images des tensions mesurées, elles-mêmes comprises entre 0V et +5V.



# Etude de cas: Arduino

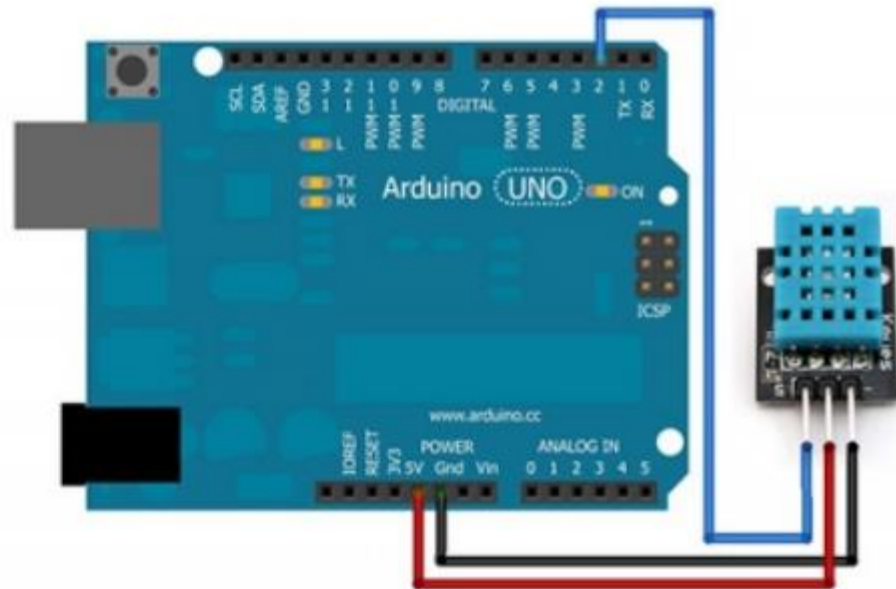
## Utilisation d'un capteur numérique

Les capteurs numériques produisent en sortie une séquence d'états logiques. Les capteurs numériques donnent en sortie une valeur finie (des valeurs discrètes). En Arduino, les capteurs numériques sont lus via les ports numérique (0, 1, 2, ..., 13) grâce à la fonction `digitalRead()`.

- ❑ Utiliser un DHT11 et afficher les valeurs de température et d'humidité sur le moniteur série.

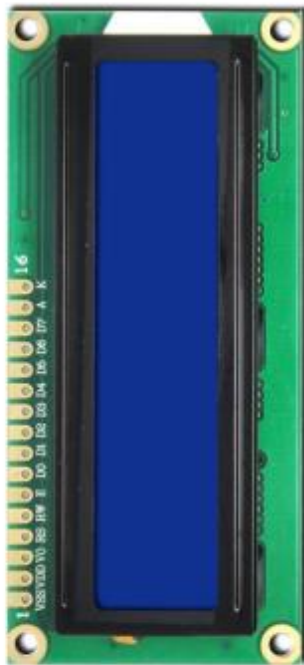


Utilisation de la bibliothèque DHTLib : Dans Croquis - Inclure une bibliothèque - Gérer les bibliothèques.



# Etude de cas: Arduino

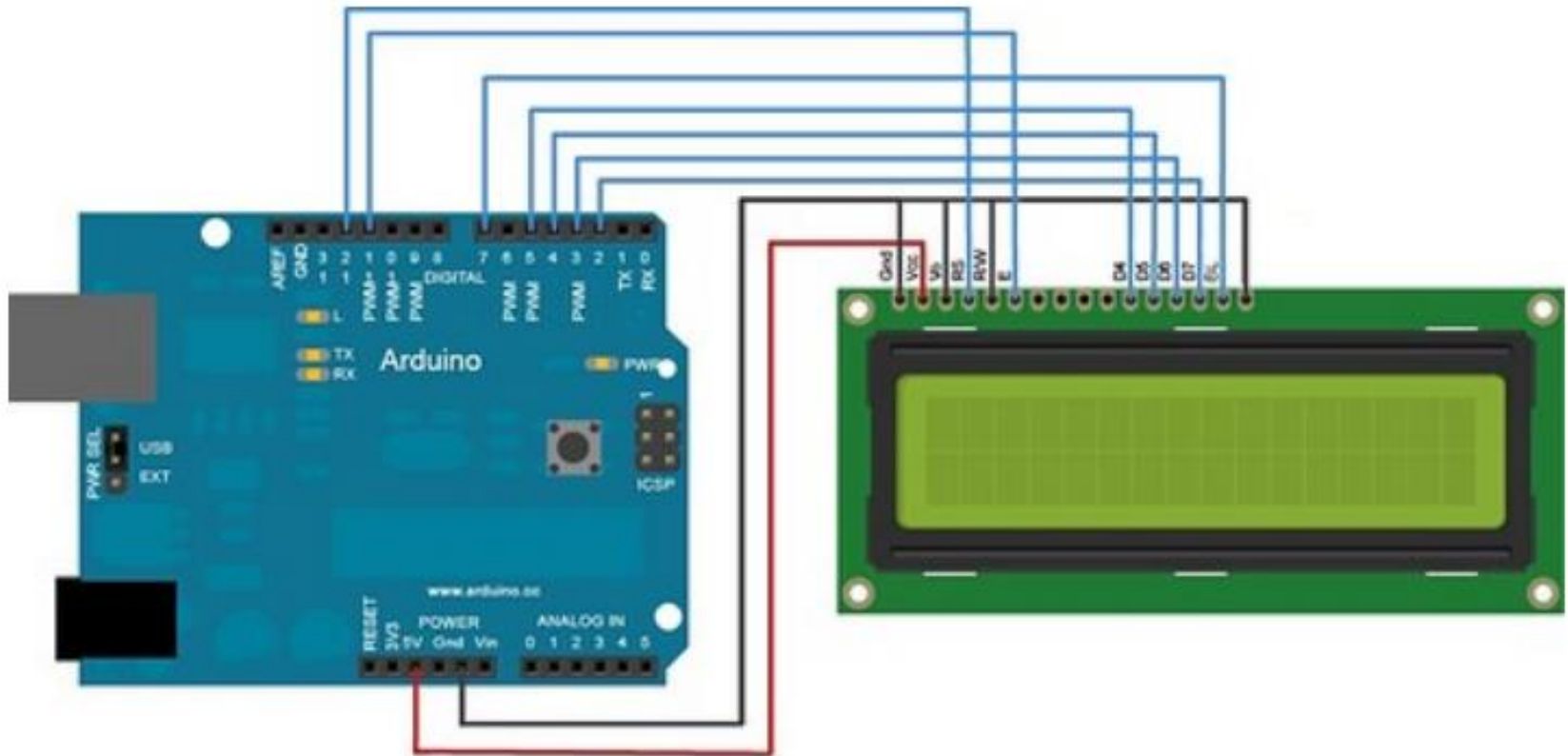
Un écran LCD nous permet d'afficher des informations de notre système sans être nécessaire de le brancher sur notre ordinateur. Les LCD 16×2 sont les plus utilisés dans le monde Arduino et sont très suffisant pour un grand nombre d'application. Il est ainsi composé:



- K katode. borne – de la LED de rétroéclairage ;
- A anode. borne + de la LED de rétroéclairage ;
- D4, D5, D6 et D7 pour la transmission des données à afficher ;
- D0, D1, D2 et D3 sont reliés au ground (GND) ;
- E enable. active ou non l'affichage ;
- RW Read or Write. Toujours à la masse ;
- RS Register Select. Permet de sélectionner la zone mémoire ;
- V0 Broche de contraste. Connecté à une sortie PWM ou à un potentiomètre ;
- VDD Broche d'alimentation. Typiquement connectée à la broche 5V de l'Arduino ;
- VSS Relier à la masse de l'écran.

# Etude de cas: Arduino

- ❑ Utiliser un écran LCD pour afficher les valeurs de température et d'humidité Collectées par le DHT11.



# Etude de cas: Arduino

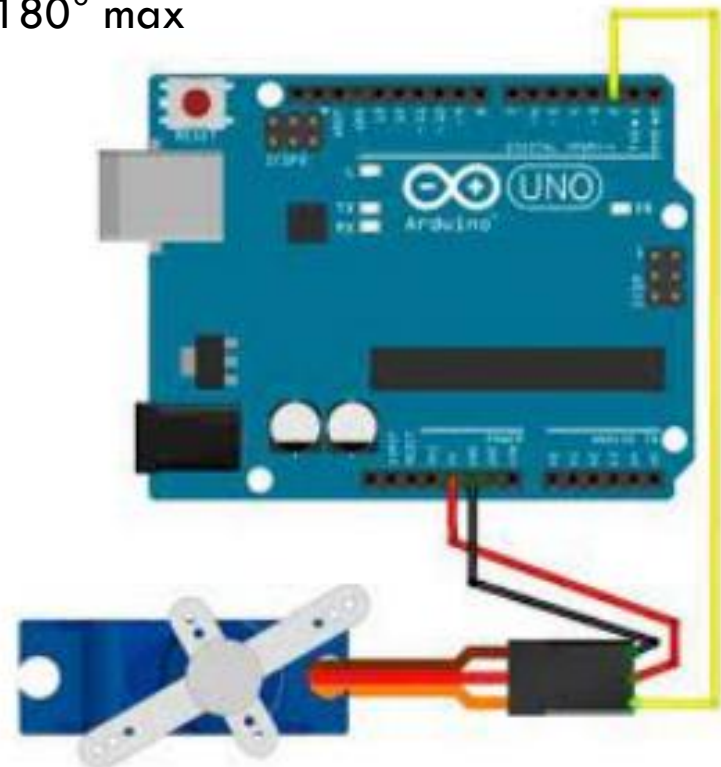
## Utilisation d'un servomoteur

..

Rotation sur 180° max



Inclure la bibliothèque Servo : **Croquis -**  
Inclure une bibliothèque - Servo

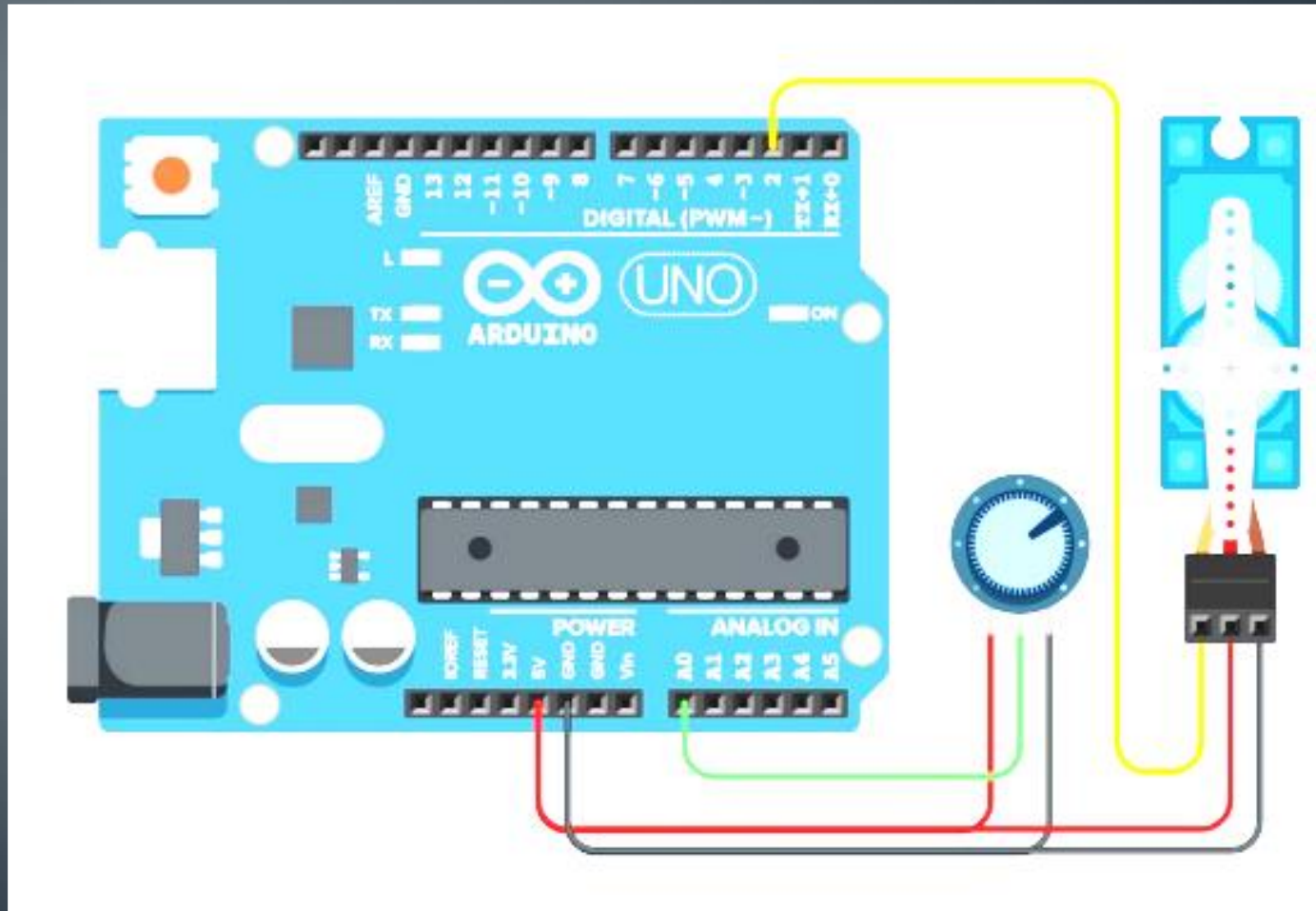




# Etude de cas: Arduino

□ Servo et potentio

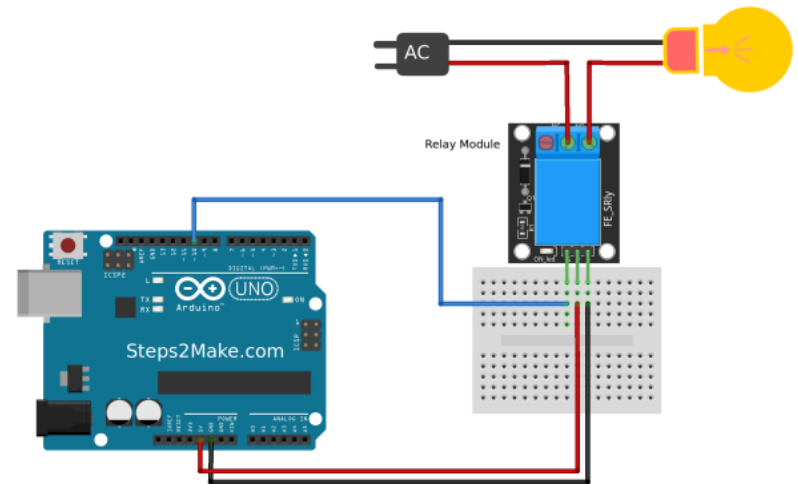
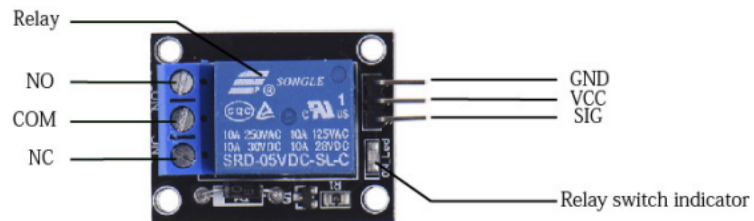
..



# Etude de cas: Arduino

Utilisation d'un relais (commutateur électrique) pour commander un second circuit ;

..





**À suivre**

**Feedback sur:**

**[pape.abdoulaye.barro@gmail.com](mailto:pape.abdoulaye.barro@gmail.com)**