

# LAMPE TACTILE

---

## Vous allez créer une lampe qui s'allumera ou s'éteindra lorsque vous toucherez un morceau de métal conducteur

Découverte : Installation de bibliothèques 'tierce-partie', création d'un capteur de contact

Durée : 45 minutes

Difficulté : ■■■■

Basé sur les projets : 1,2,5

**Vous allez utiliser la bibliothèque 'CapacitiveSensor' (Capteur Capacitif) de Paul Badger pour ce projet. Cette bibliothèque vous permettra de mesurer la capacitance de votre corps.**

La *capacitance* est la mesure de la quantité de charge électrique que quelque chose peut stocker. La bibliothèque vérifie deux broches de votre Arduino (l'une sert d'émetteur, l'autre de récepteur), et mesure le temps nécessaire pour que ces deux broches soient dans le même état. Ces broches seront connectées à un objet métallique, par exemple une feuille d'aluminium. Lorsque vous vous approcherez de cet objet, votre corps absorbera une certaine quantité de charges, provoquant l'augmentation du temps nécessaire aux deux broches pour parvenir au même état.

### Préparation de la bibliothèque

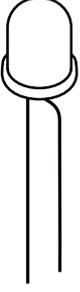
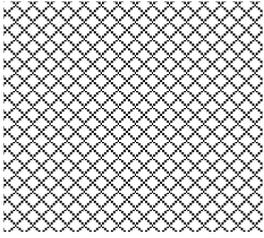
La version la plus récente de la bibliothèque 'CapacitiveSensor' est ici : <http://www.arduino.cc/capacitive>. Téléchargez le fichier sur votre ordinateur et décompressez-le.

Ouvrez le répertoire contenant les croquis Arduino (par défaut, il devrait se trouver dans votre répertoire "Documents"). Dans ce dossier, créez, et s'il n'existe pas déjà, un nouveau répertoire appelé "libraries". Déposez dans ce répertoire, le dossier "CapacitiveSensor" que vous avez décompressé et redémarrez le logiciel Arduino.

Cliquez sur le menu "Files>Exemples" dans le logiciel Arduino, et vous remarquerez la nouvelle entrée "CapacitiveSensor". En effet, la bibliothèque que vous venez d'importer contient un exemple de projet. Ouvrez l'exemple de croquis 'CapacitiveSensorSketch' et compilez-le. Si aucune erreur ne survient pendant la compilation, alors vous saurez que l'installation de la bibliothèque est réussie.

Pour plus d'informations sur les bibliothèques, visitez <http://www.arduino.cc/en/Reference/Libraries>

## INGREDIENTS

			
RESISTANCE 220 $\Omega$	RESISTANCE 1 M $\Omega$	LED	FEUILLE METALLIQUE
x1	x1	x1	x1

# CONSTRUIRE LE CIRCUIT

Fig.1 - [Circuit]

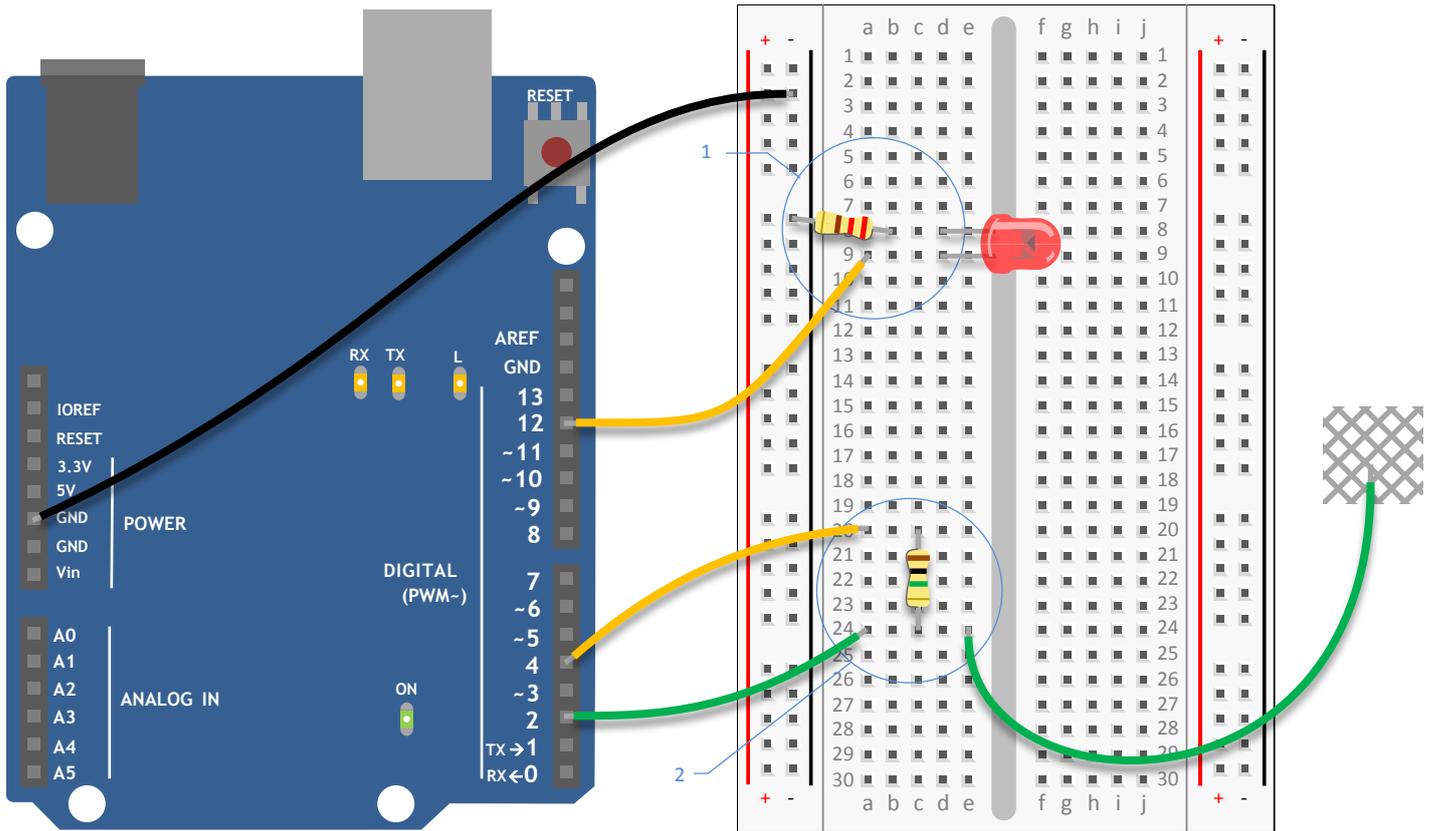
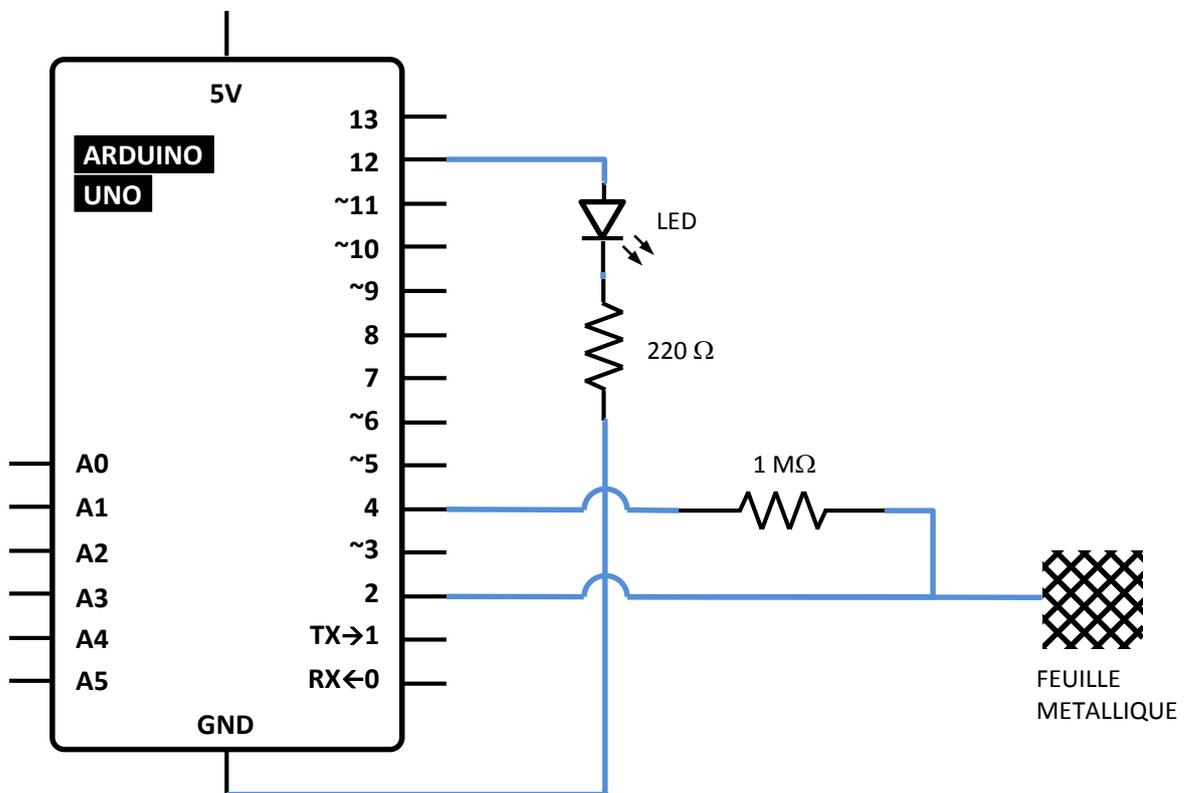


Fig.2 - [Schéma]



- 1) Connecter une LED, l'anode à la borne 12 de Arduino, et la cathode à la masse via une résistance de  $220 \Omega$  comme illustré sur les *Fig.1* et *Fig.2*.
- 2) Connecter les bornes numériques 2 et 4 à votre platine d'expérimentation. Connecter ces 2 bornes avec une résistance de  $1 M\Omega$ . Sur la même colonne que celle où est connectée la borne 2, insérer un long fil (au moins 8-10 cm) dont l'une des extrémités sera libre et sortira de votre platine d'expérimentation. Cela constituera votre capteur de contact.

**Dans ce projet, il n'y a pas besoin de fournir 5V à votre platine de test. Le courant nécessaire au fonctionnement du capteur est directement fourni par la borne numérique 4 de Arduino.**



Comme pour les autres projets utilisant des LEDs, diffuser la lumière rendra tout cela beaucoup plus attrayant. Des balles de ping-pong, de petits abat-jours fait de papier ou de plastique, ou quoi que ce soit que vous ayez à portée de main fera l'affaire.

Vous pouvez cacher le capteur derrière quelque chose de solide (couverture de livre, plaque de carton, sous-main, set de table ...), il continuera à fonctionner. En effet, la capacitance peut être mesurée à travers des matériaux non conducteurs tels que le bois et le plastique. Augmenter l'aire du capteur avec une surface conductrice plus grande le rendra plus sensible; essayer de connecter à votre fil libre une feuille d'aluminium ou un fin grillage de cuivre. Vous pourriez aussi faire un pied pour votre lampe, à l'aide de carton, de fines planches de bois, ou de tissu, et recouvrir la surface interne de cette base avec une feuille de métal reliée au fil de votre capteur. Tout le pied de la lampe se comporterait alors comme un capteur de contact. Si vous effectuez ces modifications, ajustez la variable de seuil (`seuilDeclenchement`) dans le code afin de vous assurer que vous obtenez toujours un résultat fiable.

## LE PROGRAMME

### Déclaration de la bibliothèque *CapacitiveSensor*

Au début du programme, importez la bibliothèque **CapacitiveSensor**. Cette déclaration s'effectue de la même façon que pour une bibliothèque Arduino native comme la bibliothèque **Servo** des projets précédents.

```
#include <CapacitiveSensor.h>
CapacitiveSensor capteurCapa = CapacitiveSensor(4,2);
```

### Configuration du seuil de déclenchement

Créez une variable pour définir le seuil de sensibilité qui déclenchera l'allumage de lampe. Vous modifierez cette valeur après avoir testé la plage de fonctionnement du capteur. Enfin, définissez la broche sur laquelle la LED est connectée.

```
int      seuilDeclenchement = 1000;
const int brocheLED        = 12;
```

Dans la fonction **setup()** (=configuration), ouvrez une connexion série à 9600 bps. Vous l'utiliserez pour afficher les valeurs mesurées par le capteur. Déclarez aussi la `brocheLED` comme **OUTPUT** (=sortie).

```
void setup() {
    Serial.begin(9600);
    pinMode(brocheLED, OUTPUT );
}
```

### Détection d'un contact

Dans la fonction **loop()** (=boucle), créez une variable de type **long** qui stockera la mesure faite par le capteur. La bibliothèque renvoie la valeur mesurée en utilisant la commande appelée **capacitiveSensor()** qui prend un argument donnant le nombre d'échantillons que vous souhaitez lire pour effectuer et lisser la mesure. Si vous utilisez seulement quelques échantillons, il est possible que vous constatiez de nombreuses variations dans les mesures du capteur. Si vous prenez trop d'échantillons, vous introduirez, malgré vous, une latence sensible dans votre programme puisque la fonction lira les valeurs aux bornes du capteur de très nombreuses fois avant de renvoyer la valeur de la mesure. 30 échantillons semblent être une bonne valeur pour commencer. Affichez la valeur mesurée par le capteur et retournée par la fonction dans le moniteur série.

```
void loop() {
    long mesureCapteur = capteurCapa.capacitiveSensor(30);
    Serial.print("Mesure : ");
    Serial.println(mesureCapteur);
}
```

### Contrôle de la lampe

Avec une instruction **if()...else** (=si()...sinon), vérifiez si la mesure faite par le capteur est supérieure à la limite définie pour le seuil de détection. Si oui, allumez la LED. Si non, éteignez-la.

```
if (mesureCapteur > seuilDeclenchement) {
    digitalWrite(brocheLED, HIGH);
}
else {
    digitalWrite(brocheLED, LOW );
}
```

Enfin, insérez une courte pause avec l'instruction **delay()** (=retarder) avant de clore la fonction **loop()**.

```
delay(10);
}
```

## UTILISEZ LE MONTAGE

Après avoir programmé Arduino, vous voudrez trouver quelles sont les valeurs retournées par le capteur quand il est touché. Ouvrez le moniteur série et notez la valeur donnée par le capteur lorsque vous ne le touchez pas. Appuyez doucement sur la partie du fil laissée libre et dépassant de votre platine d'expérimentation, ou bien si vous avez procédé à la modification du montage, sur la feuille d'aluminium. Le nombre affiché dans le moniteur série devrait augmenter. Essayer d'appuyer plus fermement et voyez comment évolue la mesure.

Une fois que vous avez une idée de la plage des valeurs qui peuvent être retournées par le capteur, retournez au croquis et modifiez la variable `seuilDeclenchement`, qui définit le seuil de déclenchement de l'allumage de la LED, en lui donnant une valeur plus grande que celle renvoyée par le capteur quand il n'est pas touché, mais aussi plus petite que celle renvoyée quand vous le touchez. Rechargez le croquis avec la nouvelle valeur dans Arduino. La lumière devrait s'allumer de manière fiable quand vous touchez votre capteur (le fil libre, ou bien la feuille métallique), et s'éteindre quand vous ne touchez plus le capteur. Si la lampe ne s'allume pas lorsque vous touchez le capteur, essayez de baisser encore un peu la valeur du seuil de déclenchement.



Vous avez probablement remarqué que les mesures du capteur dépendent de la manière dont vous le touchez et en particulier de la taille de la surface de votre doigt touchant le capteur. Pouvez-vous utiliser cette caractéristique pour créer d'autres interactions avec la LED? Et si vous utilisiez plusieurs capteurs pour augmenter ou réduire l'intensité de la lumière produite par la LED? Et si vous placiez une résistance d'une valeur différente entre les broches 2 et 4, cela modifiera la sensibilité du capteur. Est-ce que cela pourrait être utile pour votre interface? Vous pourriez essayer de détecter la proximité de votre main sans contact, essayer de détecter un simple effleurement, ou bien à l'inverse, faire en sorte que vous deviez plaquer la paume de votre main contre le capteur pour déclencher l'allumage, ...

*Les bibliothèques tierce-partie comme la **CapacitiveSensor** de Paul Badger sont des outils utiles pour étendre les capacités de Arduino. Une fois installées, elles se comportent de la même manière que les librairies fournies avec le logiciel de base.*

# EXTRAIT DU LIVRE DE PROJETS ARDUINO

## EDITEURS

Projets et texte par Scott Fitzgerald et Michael Shiloh  
Revue de texte complémentaire par Tom Igoe

## DESIGN ET DIRECTION ARTISTIQUE

### TODO

Giorgio Olivero, Mario Ciardulli, Vanessa Poli, Michelle Nebiolo  
todo.to.it

## EDITION NUMERIQUE ET GESTION DE PROJET

Officine Arduino Torino  
Katia De Coi, Enrico Bassi

## CONSEILLERS ET SUPPORTERS

Massimo Banzi, Gianluca Martino, Smart Projects

## TESTEURS DES PROJETS ET RELECTEURS

Michael Shiloh, Michelle Nebiolo, Katia De Coi, Alessandro Buat, Frederico Vanzati, David Mellis

## REMERCIEMENTS

Un grand merci à toute la communauté des utilisateurs Arduino pour leurs contributions continues, leur soutien, et leurs retours.

Nous remercions particulièrement l'équipe Fritzing: Quelques-unes des illustrations de composants électroniques utilisés dans le livre sont issues ou inspirées par le projet open-source de Fritzing ([www.fritzing.org](http://www.fritzing.org)).

Un grand merci à Paul Badger pour la bibliothèque *CapacitiveSensor* utilisée dans le projet 13.

Le texte du Livre de Projets Arduino est distribué sous licence Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License 2012 par Arduino LLC. Cela signifie que vous pouvez copier, réutiliser, adapter et vous appuyer sur le texte de ce livre en citant l'œuvre originale (mais pas d'une manière qui suggérerait que nous souscrivons à vous ou votre utilisation de l'œuvre) et seulement si le résultat de votre travail est transmis sous la même licence Creative Commons.

Les termes complets de la licence sont disponibles à : [creativecommons.org/licenses/by-nc-sa/3.0/](http://creativecommons.org/licenses/by-nc-sa/3.0/)

© 2012 Arduino LLC. Le nom *Arduino* et le logo sont des marques de Arduino, déposées aux États-Unis et dans le reste du monde. Les autres noms de produits et de sociétés mentionnés dans ce document sont des marques commerciales de leurs sociétés respectives.

Les informations contenues dans ce livre sont distribuées «telles quelles» sans aucune garantie supplémentaire. Bien que toutes les précautions aient été prises dans la conception de ce livre, ni les auteurs ni Arduino LLC ne pourraient endosser une quelconque responsabilité envers toute personne ou entité à l'égard de toutes pertes ou dommages causés ou déclarés causés directement ou indirectement par les instructions contenues dans ce livre ou par le logiciel et le matériel qu'il décrit.

Ce livre ne peut être vendu séparément du 'Kit de Démarrage Arduino'.

Conçu, imprimé et relié à Turin, Italie  
Septembre 2012

Première réimpression, Décembre 2012

Traduit de l'anglais par Nicolas PONCET