



STARTER KIT OTRONIC ARDUINO UNO

# Kit de démarrage Arduino UNO : tutoriel et 16 projets

Pas à pas : qu'est-ce qu'un Arduino, que contient ton kit, comment fonctionne Codey, et 16 projets à construire toi-même.

[codey.online/fr/starter-kits/uno/tutorial](https://codey.online/fr/starter-kits/uno/tutorial)



# Sommaire

|                                           |           |
|-------------------------------------------|-----------|
| <b>Bienvenue ! Tu vas le faire ! 🚀</b>    | <b>3</b>  |
| <b>Qu'est-ce qu'un Arduino UNO ?</b>      | <b>4</b>  |
| <b>Qu'y a-t-il dans ton starter kit ?</b> | <b>5</b>  |
| Breadboard                                | 6         |
| Câbles jumper                             | 7         |
| LED (petite lampe)                        | 8         |
| Résistance 220 Ω                          | 9         |
| Bouton-poussoir                           | 10        |
| Buzzer actif                              | 11        |
| Capteur de lumière (LDR)                  | 12        |
| Capteur de température DHT11              | 13        |
| Capteur de distance HC-SR04               | 14        |
| Capteur de mouvement (PIR)                | 15        |
| Petit écran OLED 0,91"                    | 16        |
| Servomoteur SG90                          | 17        |
| Anneau LED RGB (8 LEDs)                   | 18        |
| Relais 5V                                 | 19        |
| Clip pour pile 9V                         | 20        |
| <b>C'est quoi, le vibe coding ?</b>       | <b>21</b> |
| <b>Crée ton compte gratuit</b>            | <b>22</b> |
| <b>Voici comment fonctionne Codey</b>     | <b>23</b> |
| <b>Connecter et téléverser</b>            | <b>24</b> |
| <b>C'est parti ! 🎉</b>                    | <b>25</b> |
| Ton premier montage sur breadboard        | 26        |
| Clignotant                                | 27        |
| Feu tricolore                             | 28        |
| Bouton lumière                            | 29        |
| Sonnette                                  | 30        |
| Anneau arc-en-ciel                        | 31        |
| Mini-station météo                        | 32        |
| Capteur de stationnement                  | 33        |
| Petite veilleuse                          | 34        |
| Alarme de mouvement                       | 35        |
| Barrière                                  | 36        |
| Jeu de réflexes                           | 37        |
| Dé électronique                           | 38        |
| Capteur de parking luxe                   | 39        |
| Télémètre avec barre lumineuse            | 40        |
| Ventilateur malin                         | 41        |
| Barrière automatique                      | 42        |

## STARTER KIT OTRONIC ARDUINO UNO

# Bienvenue ! Tu vas le faire ! 🚀

Avec ce starter kit, tu construis tes propres petits appareils intelligents. Une station météo, un capteur de parking, une alarme... Et le plus beau : tu n'as pas besoin de savoir programmer. Codey t'aide.

Lis vite ce petit tutoriel. Comme ça, tu sauras ce qu'il y a dans ton kit et ce que tu peux faire avec. Ensuite, tu commences tout de suite ton premier projet.

**Voilà comment ça se passe :**

- 2 **Qu'est-ce qu'un Arduino et que peux-tu faire avec ?**
- 3 **Toutes les pièces de ton kit, une par une**
- 4 **Qu'est-ce que le vibe coding ? Programmer avec l'IA**
- 5 **Créer ton compte gratuit**
- 6 **Voici comment fonctionne Codey**
- 7 **Brancher ton Arduino et téléverser ton code**
- 8 **Commencer avec 16 projets**

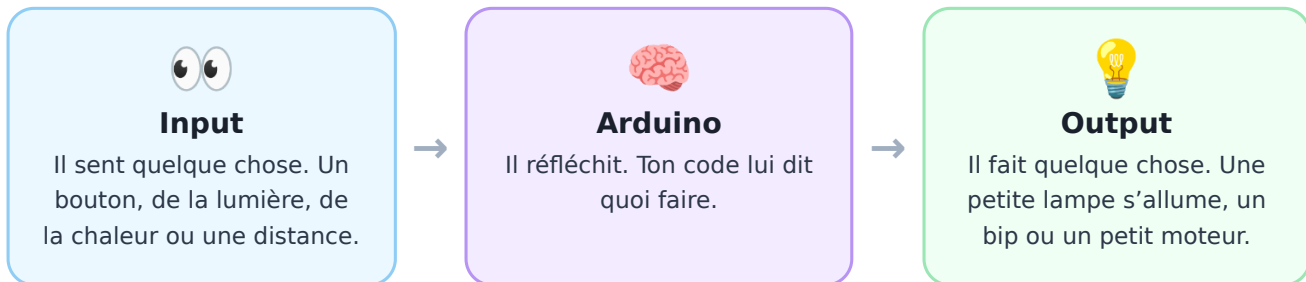


🕒 Cela prend environ 15 minutes. Tu peux toujours passer une partie et revenir plus tard.

# Qu'est-ce qu'un Arduino UNO ?

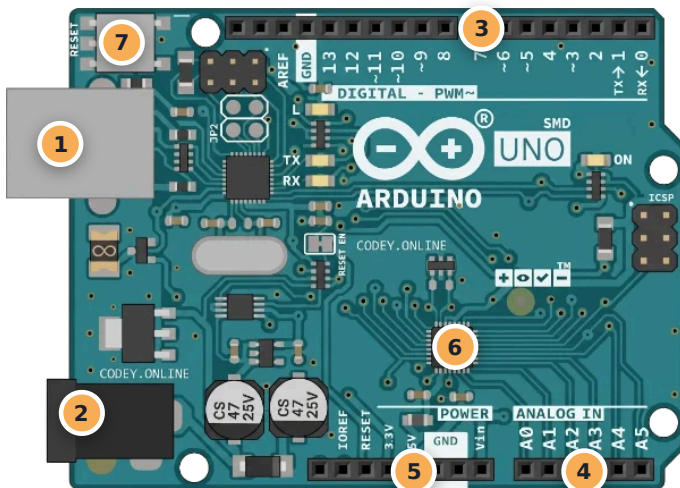
L'Arduino est un tout petit ordinateur. Tu lui dis quoi faire, et il le fait. Encore et encore, sans se fatiguer.

## Voici comment pense un Arduino



**Exemple :** le capteur de distance voit que tu t'approches (input). L'Arduino pense : "trop près !" Et le buzzer se met à biper (output). C'est comme ça que fonctionne un capteur de parking dans une voiture.

## Qu'y a-t-il sur la carte ?



- 1 Port USB : c'est ici que va le câble vers ton ordinateur. Il reçoit le courant et le code par là.
- 2 Prise d'alimentation : pour une batterie ou un adaptateur (7-12V), si tu n'utilises pas d'ordinateur.
- 3 Pins digitales D0 à D13 : allumé ou éteint. Pour les petites lampes, les boutons et les buzzers.
- 4 Pins analogiques A0 à A5 : elles mesurent une quantité. Par exemple combien il y a de lumière.
- 5 Pins d'alimentation : 5V (plus) et GND (moins) donnent du courant à tes composants.
- 6 La puce : le cerveau. C'est ici que ton code est placé.
- 7 Bouton reset : il redémarre ton code.

### Deux mots que tu vas voir souvent :

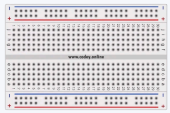
- **5V** est le plus. C'est de là que vient le courant (5 volts, comme un chargeur de téléphone).
- **GND** (ground) est le moins. C'est par là que le courant repart. Chaque composant a besoin d'un fil vers GND.

Les **pins** sont les petits trous sur le côté. Tu y branches des fils. Une pin peut être une **entrée** (elle écoute) ou une **sortie** (elle donne du courant). Tu le décides dans le code.

## Qu'y a-t-il dans ton starter kit ?

Ton kit est plein de composants. Certains ressentent quelque chose (capteurs). D'autres font quelque chose (petites lampes, buzzer, petit moteur). Clique sur Suivant. Tu vas les découvrir un par un.

### Base : avec ça, tu construis tout



#### Breadboard

La plaque de montage : construire sans souder

**Dans ton kit : 1x**



#### Câbles jumper

Câbles Dupont : les petits fils qui relient tout

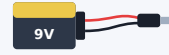
**Dans ton kit : 40x**



#### Résistance 220 Ω

Elle ralentit le courant

**Dans ton kit : 10x**

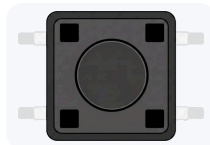


#### Clip pour pile 9V

Faire fonctionner ton projet sans ordinateur

**Dans ton kit : 1x**

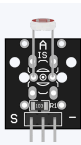
### Input : ces composants ressentent ou mesurent quelque chose



#### Bouton-poussoir

Un petit bouton sur lequel appuyer

**Dans ton kit : 5x**



#### Capteur de lumière (LDR)

Mesure s'il fait clair ou sombre

**Dans ton kit : 1x**



#### Capteur de température DHT11

Mesure la température et l'humidité de l'air

**Dans ton kit : 1x**



#### Capteur de distance HC-SR04

Mesure la distance avec le son, comme une chauve-souris

**Dans ton kit : 1x**

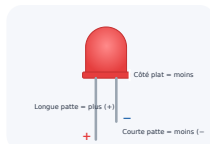


#### Capteur de mouvement (PIR)

Voit si quelqu'un bouge

**Dans ton kit : 1x**

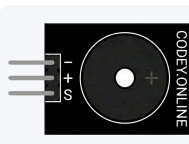
### Output : ces composants font quelque chose



#### LED (petite lampe)

Une petite lampe en 5 couleurs

**Dans ton kit : 10x**



#### Buzzer actif

Fait des bips

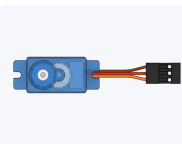
**Dans ton kit : 1x**



#### Petit écran OLED 0,91"

Un petit écran pour du texte et des nombres

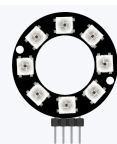
**Dans ton kit : 1x**



#### Servomoteur SG90

Un petit moteur qui tourne vers une position

**Dans ton kit : 1x**



#### Anneau LED RGB (8 LEDs)

8 petites lampes intelligentes de toutes les couleurs

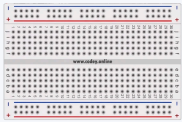
**Dans ton kit : 1x**



#### Relais 5V

Un interrupteur que ton Arduino peut actionner

**Dans ton kit : 1x**



## Breadboard

La plaque de montage : construire sans souder

Dans ton kit : 1x

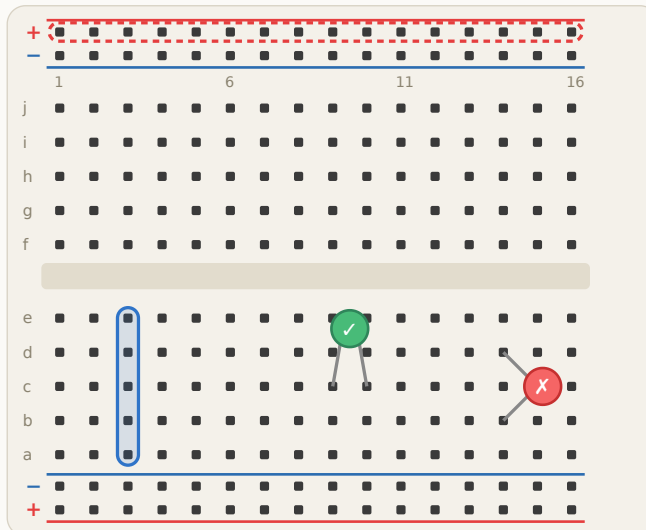
### Qu'est-ce que c'est ?

Un breadboard est une petite plaque pleine de trous. Tu y mets des composants et des fils. Sous les trous, il y a des bandes en métal qui relient certains trous entre eux. Comme ça, tu fais un circuit sans souder. Tu t'es trompé ? Tu retires simplement et tu recommences.

### Voici comment le brancher

Voici comment c'est fait :

- **Rangées de 5** : dans la partie du milieu, 5 trous côte à côte sont reliés (a-e et f-j). Tout ce que tu mets dans la même rangée est relié ensemble.
- **La rainure au milieu** sépare les deux côtés. À gauche (a-e) et à droite (f-j), ce n'est donc *pas* relié.
- **Bandes d'alimentation (+ et -)** sur les grands côtés sont reliées sur toute la longueur. Relie le + au 5V et le - au GND de ton Arduino. Maintenant, tu as du courant partout.



- Bande d'alimentation : toute la longueur est reliée
- 5 trous dans une rangée sont reliés
- X Erreur : les deux pattes sont dans la même rangée

- Rainure : la gauche et la droite ne sont pas reliées
- ✓ Bien : les pattes sont dans des rangées différentes

### Que peux-tu faire avec ?

- Essayer des circuits sans souder
- Alimenter plusieurs composants en même temps avec les bandes + et -
- Mettre une LED avec une résistance bien en série

### Astuce

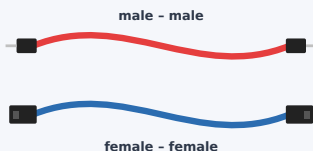
Ne mets jamais les deux pattes d'un composant dans la même rangée de 5. Elles seraient reliées entre elles et le courant passerait juste à côté. Mets plutôt un composant par-dessus la rainure.

Essaie tout de suite ! Dans l'exercice "Ton premier montage sur breadboard", tu construis une LED avec une résistance sur la breadboard, puis tu l'allumes et tu la fais clignoter. →

### Projets avec cette pièce

★ Ton premier montage sur breadboard

## Partie 2 sur 15



## Câbles jumper

Câbles Dupont : les petits fils qui relient tout

Dans ton kit : 40x

### Qu'est-ce que c'est ?

Avec des câbles jumper, tu relieras ton Arduino aux composants. On les appelle aussi **câbles Dupont**, à cause des petits connecteurs noirs aux bouts. Il y a deux sortes dans ton kit :

- **Male-male** (petite broche des deux côtés) : pour les mettre dans l'Arduino et le breadboard.
- **Female-female** (petit trou des deux côtés) : pour les glisser sur les broches d'un module.

### Voici comment le brancher

Enfonce bien le câble dans le trou ou sur la broche. Si tu combines un câble male et un câble female, tu as un câble male-female : pratique pour brancher un module directement sur ton Arduino.

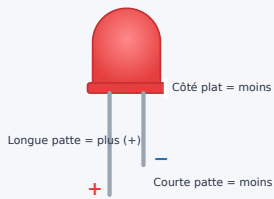
### Que peux-tu faire avec ?

- Relier ton Arduino au breadboard
- Brancher des modules capteurs
- Amener le courant (5V) et GND vers tes composants

### Astuce

Utilise toujours les mêmes couleurs : rouge pour 5V et noir pour GND. Comme ça, tu vois tout de suite si le courant est bien branché.

## Partie 3 sur 15



## LED (petite lampe)

Une petite lampe en 5 couleurs

Dans ton kit : 10x

### Qu'est-ce que c'est ?

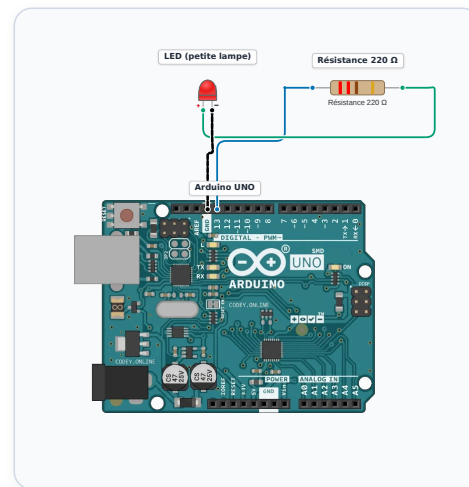
Une LED est une petite lampe qui utilise très peu de courant. Ton kit a 10 LEDs en 5 couleurs. Une LED fonctionne dans un seul sens, donc tu dois la mettre dans le bon sens.

### Voici comment le brancher

- La **longue patte** est le plus (+). Elle va vers une pin de ton Arduino avec une résistance de 220  $\Omega$ .
- La **courte patte** est le moins (–). Elle va vers GND. De ce côté, le bord de la petite lampe est aussi plat.

**Utilise toujours une résistance !** Sans résistance, la LED reçoit trop de courant et elle se casse.

### Schéma de branchement



- Arduino UNO · D13 → Résistance 220  $\Omega$  · 1
- Résistance 220  $\Omega$  · 2 → LED (petite lampe) · ANODE
- LED (petite lampe) · CATHODE → Arduino UNO · GND

### Que peux-tu faire avec ?

- Une lumière qui clignote
- Un feu de circulation
- Une petite lampe qui montre si quelque chose est allumé ou éteint

### Astuce

Ta LED ne fait rien ? Retourne-la. Elle fonctionne dans un seul sens.

### Projets avec cette pièce

★ Ton premier montage sur breadboard

★ Clignotant

★ Feu tricolore

★ Bouton lumière

★★ Alarme de mouvement

★★ Jeu de réflexes

## Partie 4 sur 15

Rouge, rouge, marron = 220  $\Omega$ 

# Résistance 220 $\Omega$

Elle ralentit le courant

Dans ton kit : 10x

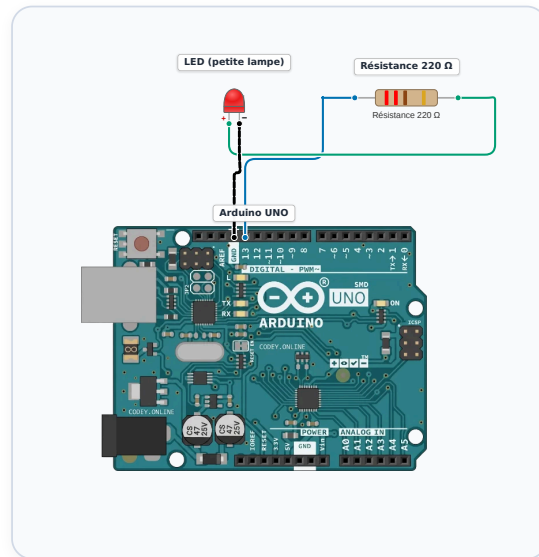
## Qu'est-ce que c'est ?

Une résistance ralentit un peu le courant. Comme un robinet que tu fermes à moitié. Comme ça, une LED reçoit juste assez de courant et elle ne se casse pas. Les résistances de ton kit font 220 ohm ( $\Omega$ ). Tu le vois aux petits anneaux de couleur : rouge, rouge, marron.

## Voici comment le brancher

Une résistance n'a pas de plus ni de moins : tu peux la mettre dans les deux sens. Mets-la entre la pin de l'Arduino et la longue patte de la LED. Un côté va vers la pin, l'autre côté vers la LED.

## Schéma de branchement



- Arduino UNO · D13 → Résistance 220  $\Omega$  · 1
- Résistance 220  $\Omega$  · 2 → LED (petite lampe) · ANODE
- LED (petite lampe) · CATHODE → Arduino UNO · GND

## Que peux-tu faire avec ?

- Protéger chaque LED
- Limiter le courant vers un composant

## Astuce

Une résistance par LED. Tu utilises trois LEDs ? Alors il te faut aussi trois résistances.

## Projets avec cette pièce

★ Ton premier montage sur breadboard

★ Clignotant

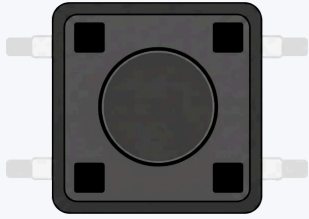
★ Feu tricolore

★ Bouton lumière

★★ Alarme de mouvement

★★ Jeu de réflexes

## Partie 5 sur 15



## Bouton-poussoir

Un petit bouton sur lequel appuyer

Dans ton kit : 5x

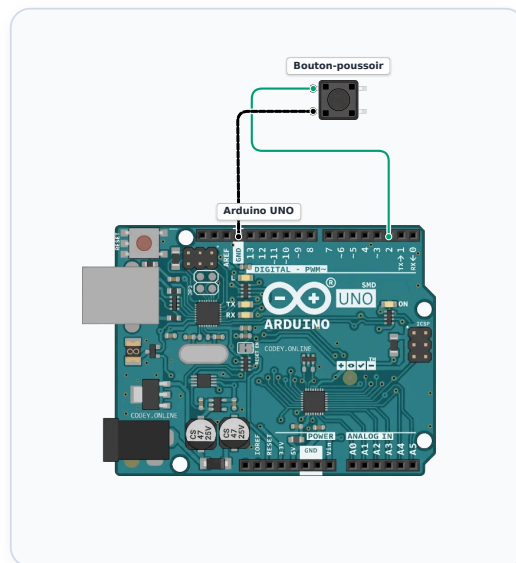
### Qu'est-ce que c'est ?

Un bouton-poussoir fait contact tant que tu appuies dessus. Quand tu le relâches, le contact disparaît. Ton Arduino peut sentir si le bouton est appuyé. Ton kit a 5 boutons.

### Voici comment le brancher

Le bouton a 4 pattes. Tu n'en as besoin que de 2 : une vers une pin (par exemple D2) et une vers GND. Dans le code, tu mets la pin en `INPUT_PULLUP`. Alors la pin est HIGH quand tu n'appuies pas, et LOW quand tu appuies.

### Schéma de branchement



● Arduino UNO · D2 → Bouton-poussoir · 1

● Arduino UNO · GND → Bouton-poussoir · 2

### Que peux-tu faire avec ?

- Allumer et éteindre une petite lampe
- Une sonnette
- Un petit jeu : qui appuie le plus vite ?

### Astuce

Tu ne sais pas quelles pattes choisir ? Prends deux pattes qui sont en diagonale l'une en face de l'autre. Comme ça, ça marche toujours. Mets le bouton au-dessus de la fente de ta breadboard.

### Projets avec cette pièce

★ Bouton lumière

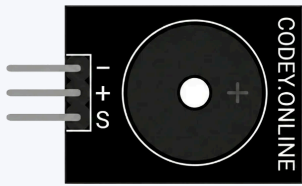
★ Sonnette

★★ Barrière

★★ Jeu de réflexes

★★ Dé électronique

## Partie 6 sur 15



## Buzzer actif

Fait des bips

Dans ton kit : 1x

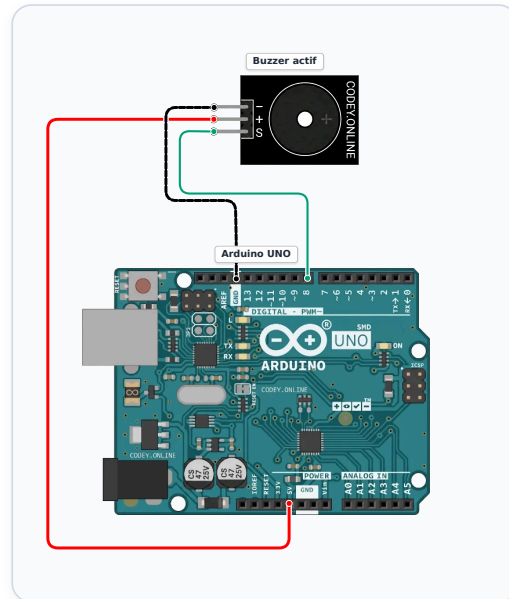
### 🔍 Qu'est-ce que c'est ?

Le buzzer fait un petit bip. C'est un buzzer **actif** : il bipe tout seul dès qu'il reçoit du courant. Tu dois seulement l'allumer et l'éteindre, comme une petite lampe.

### 🔌 Voici comment le brancher

- S (signal) vers une pin, par exemple D8
- + vers 5V
- - vers GND

### Schéma de branchement



● Arduino UNO · GND → Buzzer actif · -

● Arduino UNO · 5V → Buzzer actif · +

● Arduino UNO · D8 → Buzzer actif · S

### ✨ Que peux-tu faire avec ?

- Une alarme
- Un capteur de parking qui bipe plus vite quand tu te rapproches
- Une sonnette ou un minuteur

### 💡 Astuce

Tu veux un bip court ? Allume le buzzer, attends 100 millisecondes, puis éteins-le.

## Projets avec cette pièce

★ Sonnette

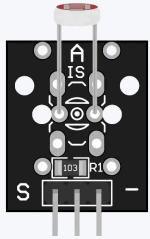
★★ Capteur de stationnement

★★ Alarme de mouvement

★★★★ Capteur de parking luxe

★★★★ Barrière automatique

## Partie 7 sur 15



## Capteur de lumière (LDR)

Mesure s'il fait clair ou sombre

Dans ton kit : 1x

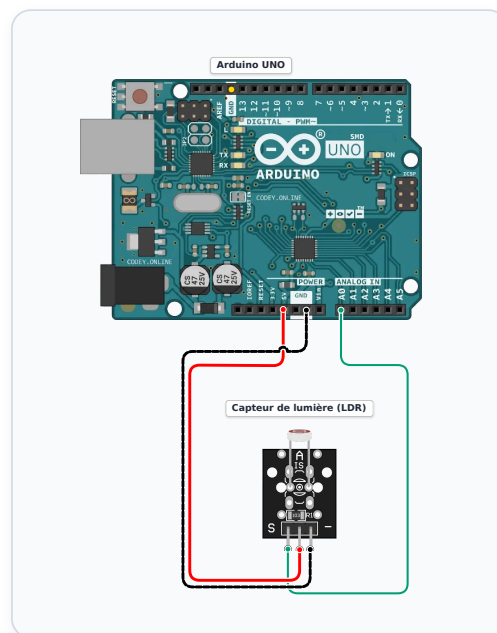
### Qu'est-ce que c'est ?

Ce capteur mesure la quantité de lumière. Ton Arduino lit un nombre entre 0 et 1023. Avec beaucoup de lumière, tu obtiens un autre nombre que dans le noir. Comme ça, ton Arduino sait s'il fait jour ou nuit.

### Voici comment le brancher

- S vers une pin analogique, par exemple A0
- + vers 5V
- - vers GND

### Schéma de branchement



- Arduino UNO · A0 → Capteur de lumière (LDR) · S
- Arduino UNO · 5V → Capteur de lumière (LDR) · +
- Arduino UNO · GND → Capteur de lumière (LDR) · -

### Que peux-tu faire avec ?

- Une veilleuse qui s'allume toute seule
- Un luxmètre
- Une alarme qui se déclenche quand quelqu'un ouvre le tiroir

### Astuce

Regarde dans le moniteur série quel nombre tu vois. Mets ensuite ta main au-dessus du capteur. Comme ça, tu vois quel nombre correspond à "sombre".

## Projets avec cette pièce

### ★★ Petite veilleuse

## Partie 8 sur 15



# Capteur de température DHT11

Mesure la température et l'humidité de l'air

Dans ton kit : 1x

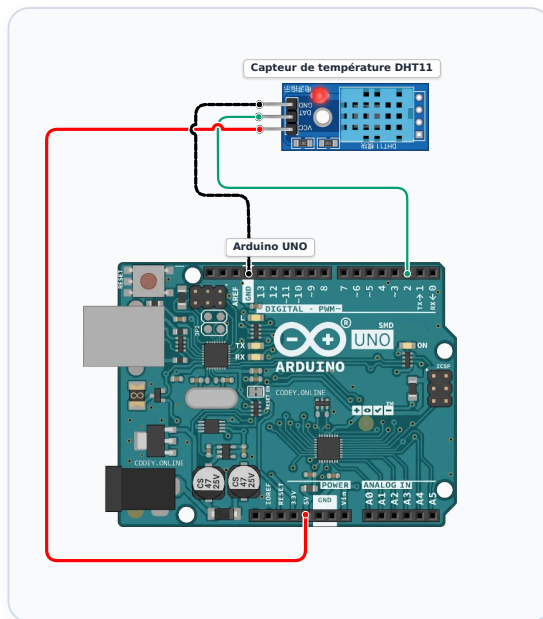
## Qu'est-ce que c'est ?

Le DHT11 mesure la chaleur qu'il fait (en degrés Celsius) et l'humidité de l'air (en pourcentage). Il donne une nouvelle mesure toutes les 2 secondes.

## Voici comment le brancher

- **DAT** (données) vers une pin, par exemple D2
- **VCC** vers 5V
- **GND** vers GND

## Schéma de branchement



- Arduino UNO · 5V → Capteur de température DHT11 · VCC
- Arduino UNO · GND → Capteur de température DHT11 · GND
- Arduino UNO · D2 → Capteur de température DHT11 · DAT

## Que peux-tu faire avec ?

- Une mini-station météo
- Un ventilateur qui s'allume quand il fait trop chaud
- Un avertissement quand il y a trop d'humidité dans la salle de bain

## Astuce

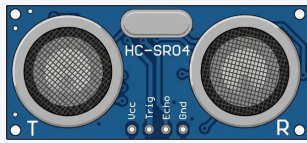
Souffle un peu sur le capteur. Tu verras l'humidité de l'air monter tout de suite !

## Projets avec cette pièce

☆☆ Mini-station météo

☆☆☆☆ Ventilateur malin

## Partie 9 sur 15



## Capteur de distance HC-SR04

Mesure la distance avec le son, comme une chauve-souris

Dans ton kit : 1x

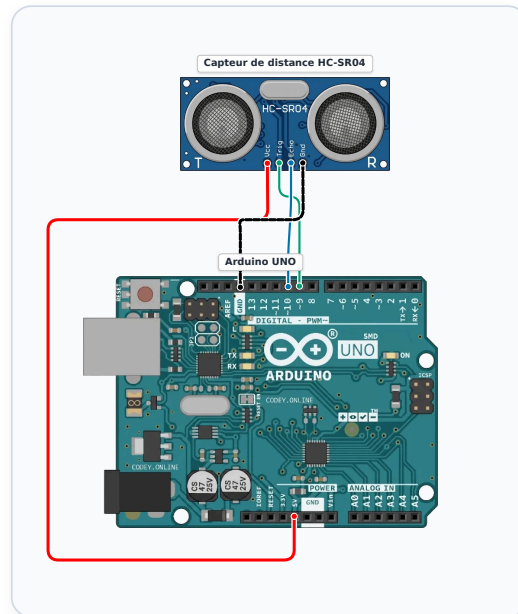
### Qu'est-ce que c'est ?

Ce capteur envoie un petit son que tu ne peux pas entendre. Le son rebondit sur quelque chose et revient. Le capteur mesure combien de temps cela prend. Comme ça, ton Arduino sait à quelle distance se trouve un objet : de 2 cm à environ 4 mètres.

### Voici comment le brancher

- **Trig** (envoyer) vers D9
- **Echo** (recevoir) vers D10
- **Vcc** vers 5V
- **Gnd** vers GND

### Schéma de branchement



- Arduino UNO · 5V → Capteur de distance HC-SR04 · Vcc
- Arduino UNO · GND → Capteur de distance HC-SR04 · Gnd
- Arduino UNO · D9 → Capteur de distance HC-SR04 · Trig
- Arduino UNO · D10 → Capteur de distance HC-SR04 · Echo

### Que peux-tu faire avec ?

- Un capteur de parking
- Un télémètre
- Une barrière qui s'ouvre toute seule

### Astuce

Les deux "yeux" sont le haut-parleur et le micro. Dirige-les bien vers ce que tu veux mesurer.

## Projets avec cette pièce

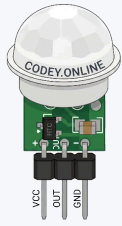
★★★ Capteur de stationnement

★★★★ Capteur de parking luxe

★★★★ Télémètre avec barre lumineuse

★★★★ Barrière automatique

## Partie 10 sur 15



## Capteur de mouvement (PIR)

Voit si quelqu'un bouge

Dans ton kit : 1×

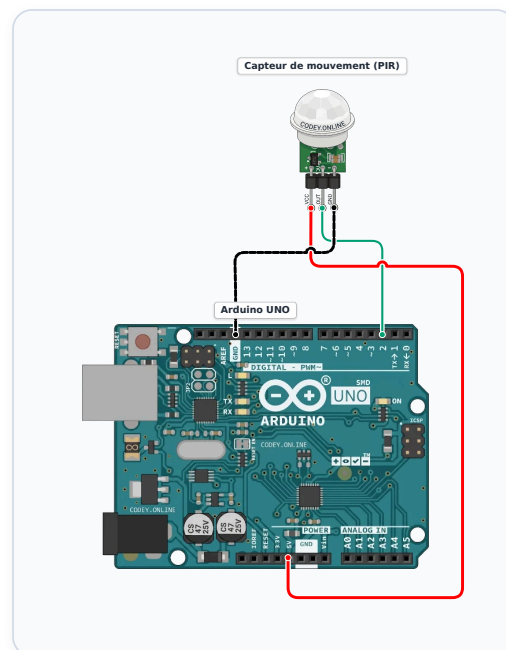
### Qu'est-ce que c'est ?

Un capteur PIR sent la chaleur des personnes et des animaux. Quelque chose de chaud bouge devant le capteur ? Alors sa sortie devient HIGH. Les lampes qui s'allument toutes seules quand tu passes fonctionnent aussi comme ça.

### Voici comment le brancher

- **OUT** vers une pin, par exemple D2
- **VCC** vers 5V
- **GND** vers GND

### Schéma de branchement



- Arduino UNO · 5V → Capteur de mouvement (PIR) · VCC
- Arduino UNO · GND → Capteur de mouvement (PIR) · GND
- Arduino UNO · D2 → Capteur de mouvement (PIR) · OUT

### Que peux-tu faire avec ?

- Une alarme anti-intrusion
- Une lampe qui s'allume quand tu entres
- Compter combien de fois quelqu'un passe

### Astuce

Après le démarrage, le capteur a besoin de quelques secondes pour s'habituer à la pièce. Reste alors immobile un petit moment.

## Projets avec cette pièce

### ★★ Alarme de mouvement

## Partie 11 sur 15



## Petit écran OLED 0,91"

Un petit écran pour du texte et des nombres

Dans ton kit : 1x

### Qu'est-ce que c'est ?

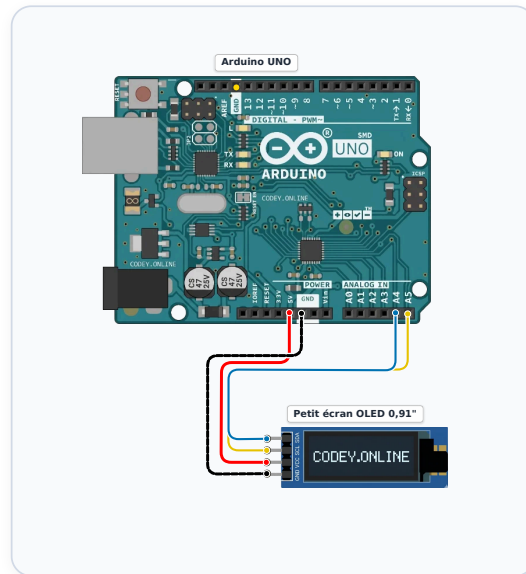
Un petit écran bien net de 128 par 32 pixels. Tu peux y afficher du texte, des nombres et des dessins simples. Par exemple la température ou une distance.

### Voici comment le brancher

L'écran parle via **I2C** : seulement deux fils de données.

- **SDA** vers A4
- **SCL** vers A5
- **VCC** vers 5V
- **GND** vers GND

### Schéma de branchement



- Arduino UNO · A4 → Petit écran OLED 0,91" · SDA
- Arduino UNO · A5 → Petit écran OLED 0,91" · SCL
- Arduino UNO · 5V → Petit écran OLED 0,91" · VCC
- Arduino UNO · GND → Petit écran OLED 0,91" · GND

### Que peux-tu faire avec ?

- Afficher la température
- Afficher la distance en centimètres
- Afficher un score ou un temps de réaction

### Astuce

L'écran reste noir ? Vérifie que SDA et SCL ne sont pas inversés.

## Projets avec cette pièce

★★ Mini-station météo

★★ Jeu de réflexes

★★★★ Capteur de parking luxe

★★★★ Ventilateur malin

## Partie 12 sur 15



## Servomoteur SG90

Un petit moteur qui tourne vers une position

Dans ton kit : 1x

### Qu'est-ce que c'est ?

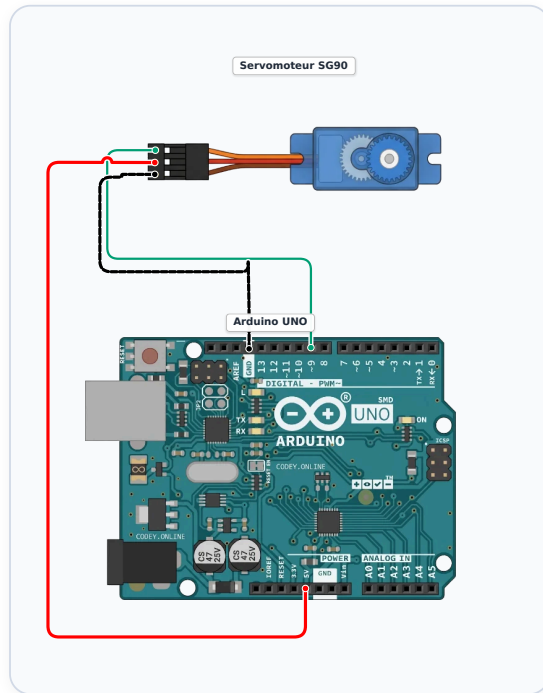
Un servomoteur est un petit moteur que tu peux faire tourner vers une position précise, de 0 à 180 degrés. Tu dis par exemple "va à 90 degrés" et il y va exactement. Sur son axe, tu mets un petit bras.

### Voici comment le brancher

Le servomoteur a trois fils :

- **Orange ou jaune** (signal) vers une pin, par exemple D9
- **Rouge** vers 5V
- **Marron** vers GND

### Schéma de branchement



- Arduino UNO · GND → Servomoteur SG90 · GND
- Arduino UNO · 5V → Servomoteur SG90 · VCC
- Arduino UNO · D9 → Servomoteur SG90 · PWM

### Que peux-tu faire avec ?

- Une barrière
- Une aiguille qui montre la température
- Un petit bras de robot qui fait coucou

### Astuce

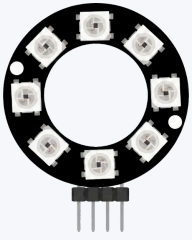
Ne tourne pas le bras à la main. Sinon, les petites roues dentées à l'intérieur peuvent se casser.

## Projets avec cette pièce

★★ Barrière

★★★★ Barrière automatique

## Partie 13 sur 15



## Anneau LED RGB (8 LEDs)

8 petites lampes intelligentes de toutes les couleurs

Dans ton kit : 1x

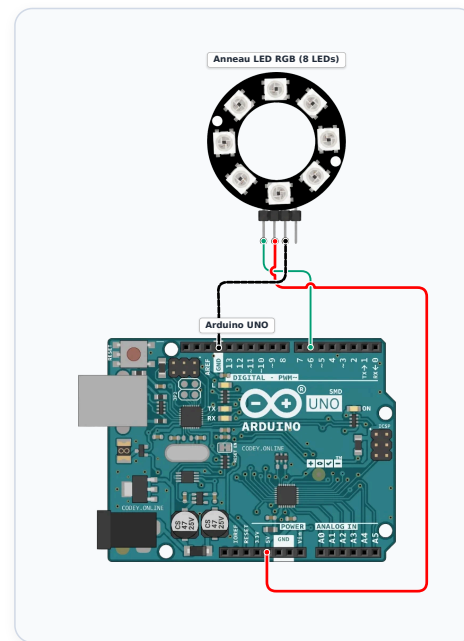
### Qu'est-ce que c'est ?

Un anneau avec 8 petites lampes intelligentes. Chaque petite lampe peut prendre n'importe quelle couleur. Et tu les commandes toutes séparément. Avec un seul fil de données ! Super chouette pour faire des effets lumineux.

### Voici comment le brancher

- **DI** (data in) vers une broche, par exemple D6
- **5V** vers 5V
- **GND** vers GND
- **DO** tu le laisses libre (tu peux y ajouter un autre anneau derrière)

### Schéma de branchement



● Arduino UNO · 5V → Anneau LED RGB (8 LEDs) · 5V

● Arduino UNO · GND → Anneau LED RGB (8 LEDs) · GND

● Arduino UNO · D6 → Anneau LED RGB (8 LEDs) · DI

### Que peux-tu faire avec ?

- Une lampe arc-en-ciel
- Une barre lumineuse qui montre à quelle distance est quelque chose
- Un dé lumineux

### Astuce

Ne mets pas la luminosité trop forte dans le code. C'est plus agréable pour tes yeux et pour ton Arduino.

### Projets avec cette pièce

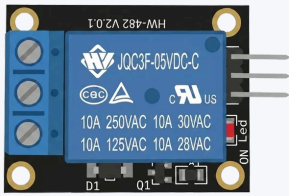
★ Anneau arc-en-ciel

★★ Petite veilleuse

★★ Dé électronique

★★★★ Télémètre avec barre lumineuse

## Partie 14 sur 15



## Relais 5V

Un interrupteur que ton Arduino peut actionner

Dans ton kit : 1x

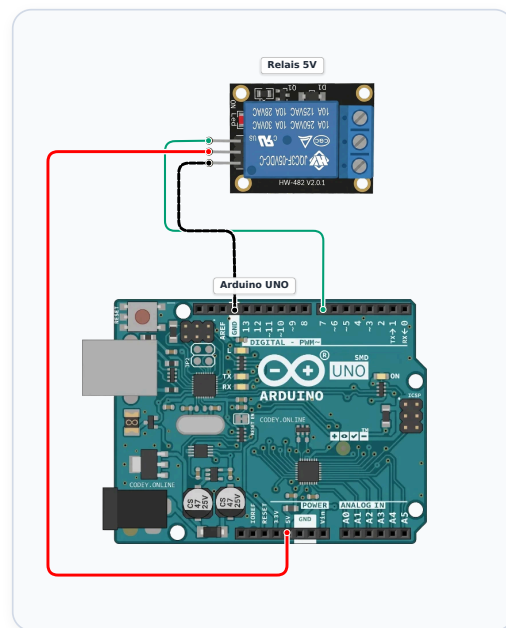
### 🔍 Qu'est-ce que c'est ?

Un relais est un interrupteur que ton Arduino peut allumer et éteindre. Tu l'entends faire « clic ». Avec un relais, ton Arduino peut allumer et éteindre un autre appareil, par exemple un petit ventilateur ou une lampe sur une batterie.

### 🔌 Voici comment le brancher

- **S** vers une broche, par exemple D7
- **+** vers 5V
- **-** vers GND
- L'appareil que tu veux commander se branche sur **COM** et **NO**. NO veut dire : normalement éteint, allumé quand le relais clique.

### Schéma de branchement



● Arduino UNO · GND → Relais 5V · -

● Arduino UNO · 5V → Relais 5V · +

● Arduino UNO · D7 → Relais 5V · S

### ✨ Que peux-tu faire avec ?

- Un ventilateur qui s'allume quand il fait chaud
- Une lampe avec un minuteur
- Une petite pompe pour les plantes

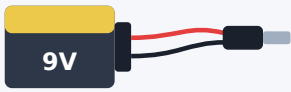
### 💡 Astuce

La sécurité d'abord : utilise le relais seulement avec des batteries ou une basse tension (jusqu'à 12V). Jamais avec une prise secteur (230V) !

## Projets avec cette pièce

★★★★ Ventilateur malin

## Partie 15 sur 15



## Clip pour pile 9V

Faire fonctionner ton projet sans ordinateur

Dans ton kit : 1x

### Qu'est-ce que c'est ?

Avec ce clip, tu branches une pile 9V sur ton Arduino. Comme ça, ton projet fonctionne aussi sans ordinateur. Pratique si tu veux le montrer ou le poser quelque part.

### Voici comment le brancher

Clipse la pile sur le clip et mets la fiche ronde dans la prise d'alimentation noire de ton Arduino. Ton code est encore sur l'Arduino, donc il démarre tout de suite.

### Que peux-tu faire avec ?

- Accrocher ton capteur de parking dans le garage
- Poser ta station météo sur le rebord de la fenêtre
- Montrer ton projet à tes amis

### Astuce

Enlève la pile quand tu as fini. Comme ça, elle dure beaucoup plus longtemps.

# C'est quoi, le vibe coding ?

Avant, il fallait d'abord apprendre à programmer pendant des mois. Maintenant, tu dis simplement avec tes mots ce que tu veux fabriquer. L'IA écrit le code pour toi.

Ça s'appelle le **vibe coding** : toi, tu as l'idée, et l'IA fait le travail de taper. Tu peux dire par exemple : *"Crée une veilleuse qui s'allume quand il fait noir."* En quelques secondes, le code est prêt, avec un schéma de branchement.

L'IA d'aujourd'hui est très forte pour programmer. Souvent, elle est même meilleure et plus rapide qu'un humain. Elle connaît toutes les pièces, fait moins d'erreurs de frappe et explique tout calmement.

## Construire un projet : avant et maintenant

### Programmer soi-même

Chercher, taper, trouver les erreurs, réessayer...

± 10 uur

### Avec Codey

Dire ce que tu veux, téléverser, et c'est prêt.

± 5 minuten ⚡

## Voilà comment ça marche



Tu dis ce que tu veux fabriquer.



Codey écrit le code et dessine comment tout brancher.



Tu construis la même chose et tu téléverses le code.



Tu veux autre chose ? Dis-le, et Codey l'adapte.

Et en plus, tu apprends ! Codey explique ce que fait le code. Comme ça, tu comprends de mieux en mieux comment ça marche.

# Crée ton compte gratuit

Pour travailler avec Codey, tu as besoin d'un compte. C'est gratuit et cela ne prend qu'une petite minute. Avec ton kit de démarrage, tu peux ensuite utiliser Codey sans limite.

1

## Inscris-toi

Clique sur "Créer un compte" et entre ton adresse e-mail et un mot de passe. Tu as un compte Google ? Clique alors sur "Continuer avec Google". Et voilà, c'est prêt.

2

## Confirme ton adresse e-mail

Tu vas recevoir un e-mail de Codey Online. Clique sur le lien dans cet e-mail. Tu ne vois pas l'e-mail ? Regarde aussi dans tes courriers indésirables.

3

## Connecte-toi

Connecte-toi avec ton adresse e-mail et ton mot de passe. Tu reviendras alors tout seul dans ce tutoriel, à l'étape suivante.

**J'ai déjà un compte : me connecter →**

👉 Va sur [codey.online/uno](https://codey.online/uno) pour créer ton compte gratuit.



Tu es encore jeune ? Alors crée le compte avec ton papa, ta maman ou un autre adulte.

# Voici comment fonctionne Codey

Codey est ton copain de programmation. Tout est dans un seul écran. Voici les boutons les plus importants.

The screenshot shows the Codey Online web interface. On the left is a sidebar with a chat window, project files, and navigation links. The main area is divided into three sections: a code editor with a C++ sketch for an Arduino Uno, a chat window on the right, and a bottom toolbar. Numbered callouts 1 through 6 highlight key features: 1. Chat button, 2. Code editor, 3. Compile & upload button, 4. Schematic diagram, 5. Serial monitor, and 6. My kit button.

## 1 Le chat

Ici, tu parles avec Codey. Écris ce que tu veux créer ou changer, avec des mots simples.

## 2 Le code

Ici, tu vois le code que Codey écrit pour toi. Tu peux aussi écrire dedans toi-même.

## 3 Compiler & téléverser vers la carte

Un seul bouton pour tout : Codey vérifie ton code, le prépare (compiler) et l'envoie avec le câble USB vers ton Arduino (téléverser). Il y a une erreur ? Alors Codey la corrige souvent tout seul.

## 4 Schéma de branchement

Il montre quel fil va où. Tu le vois dans le chat ; clique dessus pour le voir en grand.

## 5 Moniteur série

Ici, tu vois les messages que ton Arduino envoie, par exemple la température ou une distance.

## 6 Mon kit

Toutes les pièces de ton starterkit. Codey utilise seulement les choses que tu as.

**Bon à savoir :** avec ton starterkit, tu peux utiliser Codey sans limite dans tes chats du kit. Tu peux tout demander à Codey, même "que fait cette ligne ?" ou "pourquoi ça ne marche pas ?".

# Connecter et téléverser

Ton code est prêt ? Alors maintenant, il va aller dans ton Arduino. Tu fais cela en trois étapes.

**Important :** utilise **Google Chrome** ou **Microsoft Edge** sur ton ordinateur. Seuls ces navigateurs ont le droit de parler avec ton Arduino par USB. Safari, Firefox et la plupart des téléphones ne peuvent pas (encore) le faire.

## 1 Connecte ton Arduino

Branche le câble USB dans ton Arduino et dans ton ordinateur. Une petite lumière verte (ON) s'allume.

## 2 Clique sur Compiler & téléverser vers la carte

Codey vérifie ton code et le prépare (compiler). La première fois, cela prend parfois une petite minute.

## 3 Choisis ton Arduino dans la fenêtre des ports

Une fenêtre avec des ports apparaît. Choisis ton Arduino et clique sur Connecter. Les petites lumières RX et TX clignotent un instant : ton code entre dedans (téléversement). C'est prêt !

## Comment s'appelle mon Arduino dans la fenêtre des ports ?

Cela dépend de ton ordinateur. Tu en vois plusieurs ? Débranche le câble un instant et regarde lequel disparaît. C'est ton Arduino.

### Windows

Un **port COM** avec un numéro, par exemple `Arduino Uno (COM4)` ou `USB-SERIAL CH340 (COM5)`. COM1 n'est presque jamais ton Arduino.

### Mac

Quelque chose avec `usbmodem` ou `usbserial`, par exemple `/dev/cu.usbmodem14101`, `cu.usbserial-1420` ou `cu.wchusbserial1420`. Choisis toujours la version `cu.`

### Linux

`/dev/ttyACM0` ou `/dev/ttyUSB0`. Pas d'accès ? Ajoute-toi au groupe `dialout` et reconnecte-toi.

### Chromebook

Chrome montre le nom de l'appareil, par exemple `Arduino Uno`, `USB2.0-Serial` ou `USB Serial`.

**Moniteur série :** certains projets envoient des messages à ton ordinateur, par exemple la température. Tu les vois si tu cliques sur Moniteur série dans Codey. Comme ça, tu vois ce que ton Arduino "pense".

## Ça ne marche pas ? Pas de panique !

### ▼ Je ne vois pas mon Arduino dans la fenêtre des ports.

Essaie un autre câble USB. Certains câbles peuvent seulement charger et ne peuvent pas envoyer de données. Essaie aussi un autre port USB sur ton ordinateur.

### ▼ Je vois CH340 ou USB2.0-Serial, mais le téléversement ne marche pas (Windows/Mac).

Ton Arduino utilise alors une puce CH340. Installe une seule fois le driver CH340 gratuit (cherche "CH340 driver") et redémarre ton ordinateur.

### ▼ Le téléversement échoue au milieu.

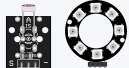
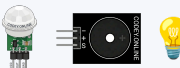
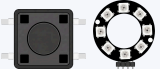
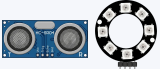
Ferme les autres programmes qui utilisent l'Arduino, comme l'Arduino IDE ou un deuxième onglet Codey. Débranche le câble un instant, rebranche-le et réessaie.

### ▼ Ça marche, mais mon montage ne fait rien.

Vérifie les fils avec le schéma de branchement. 5V et GND sont-ils bien branchés ? Demande ensuite à Codey : "Ma LED ne s'allume pas, qu'est-ce que ça peut être ?"

## C'est parti ! 🎉

Choisis un projet et clique sur FAIS CECI. Codey prépare le code et le schéma de branchement pour toi. Commence par un projet avec une étoile ★ si tu n'as encore jamais fait quelque chose avec Arduino.

|                                                                                                                                                                                                                                           |                                                                                                                                                                                                                           |                                                                                                                                                                                                                                 |                                                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p>★ ⌚ 10 minutes</p> <p><b>Ton premier montage sur breadboard</b></p> <p>Entraîne-toi avec le breadboard : allume une LED, puis fais-la clignoter.</p> |  <p>★ ⌚ 10 minutes</p> <p><b>Clignotant</b></p> <p>Ton tout premier projet : faire clignoter une LED.</p>                                |  <p>★ ⌚ 15 minutes</p> <p><b>Feu tricolore</b></p> <p>Trois LEDs qui fonctionnent comme un vrai feu tricolore.</p>                             |  <p>★ ⌚ 15 minutes</p> <p><b>Bouton lumière</b></p> <p>Appuie sur le bouton et la lampe s'allume. Encore une fois : elle s'éteint.</p> |
|  <p>★ ⌚ 10 minutes</p> <p><b>Sonnette</b></p> <p>Appuie sur le bouton pour un joyeux ding-dong.</p>                                                      |  <p>★ ⌚ 10 minutes</p> <p><b>Anneau arc-en-ciel</b></p> <p>L'anneau LED tourne avec toutes les couleurs de l'arc-en-ciel.</p>            |  <p>★★ ⌚ 20 minutes</p> <p><b>Mini-station météo</b></p> <p>Température et humidité sur ton propre petit écran.</p>                            |  <p>★★ ⌚ 20 minutes</p> <p><b>Capteur de stationnement</b></p> <p>Plus tu es près, plus ça bipe vite. Comme dans une voiture.</p>      |
|  <p>★★ ⌚ 15 minutes</p> <p><b>Petite veilleuse</b></p> <p>Il fait sombre ? Alors la lampe s'allume toute seule.</p>                                     |  <p>★★ ⌚ 20 minutes</p> <p><b>Alarme de mouvement</b></p> <p>Quelqu'un passe ? Alors l'alarme se déclenche !</p>                        |  <p>★★ ⌚ 20 minutes</p> <p><b>Barrière</b></p> <p>Appuie sur le bouton et la barrière s'ouvre.</p>                                            |  <p>★★ ⌚ 25 minutes</p> <p><b>Jeu de réflexes</b></p> <p>À quelle vitesse réagis-tu ? Mesure ton temps de réaction.</p>               |
|  <p>★★ ⌚ 15 minutes</p> <p><b>Dé électronique</b></p> <p>Appuie sur le bouton et lance un dé de lumière.</p>                                           |  <p>★★★ ⌚ 30 minutes</p> <p><b>Capteur de parking luxe</b></p> <p>Ça bipe et montre la distance en centimètres sur un petit écran.</p> |  <p>★★★ ⌚ 25 minutes</p> <p><b>Télémètre avec barre lumineuse</b></p> <p>Plus c'est proche, plus il y a de lampes : vert, orange, rouge.</p> |  <p>★★★ ⌚ 30 minutes</p> <p><b>Ventilateur malin</b></p> <p>Trop chaud ? Le relais allume le ventilateur.</p>                        |
|  <p>★★★ ⌚ 30 minutes</p> <p><b>Barrière automatique</b></p> <p>Une voiture arrive ? Alors la barrière s'ouvre toute seule.</p>                         |                                                                                                                                                                                                                           |                                                                                                                                                                                                                                 |                                                                                                                                                                                                                           |

**Tu as ta propre idée ?** Encore mieux ! Ouvre Codey et dis ce que tu veux fabriquer. Par exemple : "Fabrique une petite lampe qui clignote plus vite quand il fait plus chaud."

★ Débutant ⌚ 10 minutes

## Ton premier montage sur breadboard

Un exercice pour découvrir comment marche un breadboard. Tu construis une LED avec une résistance sur le breadboard. D'abord, tu l'allumes avec la barrette 5V, sans code. Ensuite, tu la fais clignoter avec la broche 13.

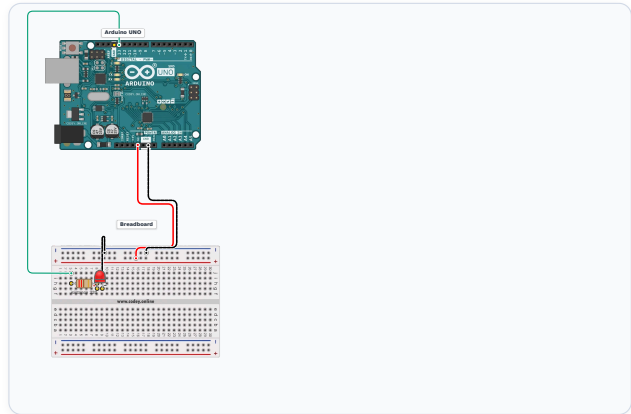
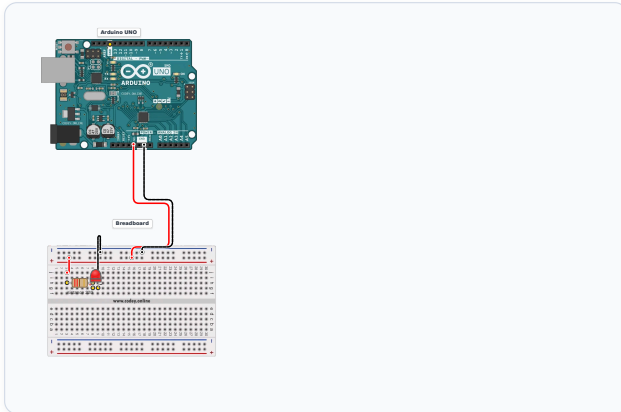
### Schéma de branchement

#### Étape A : allume la LED (sans code)

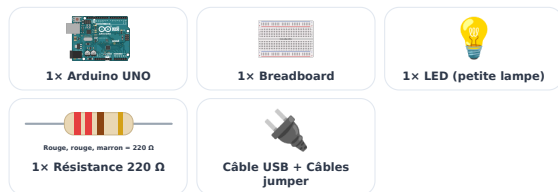
Le petit fil rouge va de la barrette + à la rangée 3. Le courant passe alors comme ça : 5V → barrette + → résistance → LED → barrette - → GND. La LED s'allume ? Alors tu as bien utilisé le breadboard !

#### Étape B : fais clignoter la LED avec la broche 13

Enlève le petit fil rouge entre la barrette + et la rangée 3. À la place, mets un fil de la broche 13 vers la rangée 3. Maintenant, c'est Arduino qui décide quand la LED reçoit du courant.



### De quoi as-tu besoin ?



### Voici comment le construire

1. Relie 5V de l'Arduino à la barrette + en haut du breadboard (fil rouge).
2. Relie GND de l'Arduino à la barrette - juste à côté (fil noir). Maintenant, tu as deux barrettes de courant.
3. Mets la résistance de 220 Ω du trou H3 au trou H8. Elle est alors dans deux rangées différentes.
4. Mets la longue patte de la LED dans G8 (la même rangée que la résistance) et la petite patte dans G9.
5. Relie le trou J9 à la barrette - (petit fil noir). Comme ça, le courant peut revenir vers GND.
6. Étape A : relie la barrette + au trou J3 (petit fil rouge). La LED s'allume !
7. Étape B : enlève ce petit fil rouge et relie la broche 13 au trou J3 (fil vert).
8. Laisse Codey créer le code : clique sur CRÉE CECI, ou écris toi-même dans le chat : "Fais clignoter la LED sur la broche 13 chaque seconde." Clique ensuite sur Compiler & téléverser vers la carte.

### Le code

```
// Ton premier montage sur breadboard
// La LED sur le breadboard s'allume et s'éteint. Elle reçoit du courant via la broche 13.

const int LED_PIN = 13; // Le fil vert va de la broche 13 au breadboard

void setup() {
  pinMode(LED_PIN, OUTPUT); // La broche 13 est une sortie : elle donne du courant
}

void loop() {
  digitalWrite(LED_PIN, HIGH); // LED allumée
  delay(1000);                // Attendre 1 seconde
  digitalWrite(LED_PIN, LOW);  // LED éteinte
  delay(1000);                // Attendre 1 seconde
}
```

#### Que fait le code ?

- La broche 13 est une sortie. Avec `HIGH`, elle donne 5V. Avec `LOW`, elle ne donne rien.
- Dans `loop()`, on allume la LED, on attend 1 seconde, on l'éteint, puis on attend encore.
- Le courant passe par le breadboard : broche 13 → rangée 3 → résistance → rangée 8 → LED → rangée 9 → barrette - → GND.

#### Essaie un peu...

- Fais clignoter la LED deux fois plus vite
- Ajoute une deuxième LED sur le breadboard qui clignote à tour de rôle
- Fais clignoter la LED trois fois très vite, puis laisse-la éteinte un petit moment

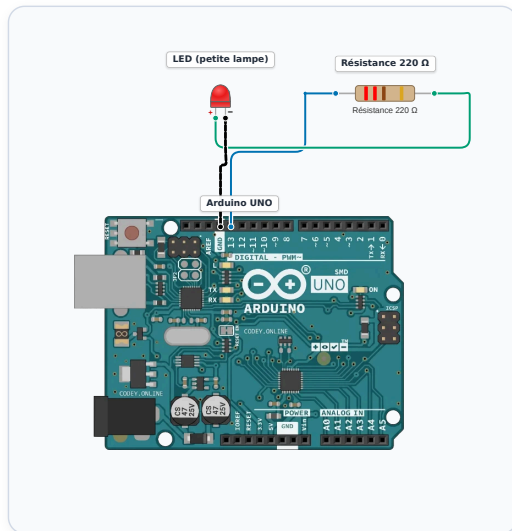
🔧 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/breadboard-led](https://codey.online/fr/starter-kits/uno/projects/breadboard-led) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★ Débutant ⌚ 10 minutes

# Clignotant

Ton premier projet ! Une LED s'allume et s'éteint sans arrêt, comme un clignotant. Tu apprends ainsi comment l'Arduino allume et éteint une sortie.

## Schéma de branchement



## De quoi as-tu besoin ?



## Voici comment le construire

1. Mets la LED dans le breadboard, avec les pattes dans deux rangées différentes.
2. Mets une résistance de 220 Ω depuis la rangée de la longue patte vers une autre rangée.
3. Relie cette autre rangée avec un fil à la pin 13 de ton Arduino.
4. Relie la courte patte avec un fil à GND.

## Le code

```
// Clignotant
// Une LED s'allume et s'eteint sans arret, comme un clignotant.

const int LED_PIN = 13; // La LED est sur la pin 13 (avec la resistance de 220 ohm)
const int TEMPS_ATTENTE = 500; // Duree ou la LED reste allumee ou eteinte (en millisecondes)

void setup() {
  // setup() tourne une fois quand l'Arduino démarre
  pinMode(LED_PIN, OUTPUT); // La pin 13 est une sortie : elle donne du courant
}

void loop() {
  // loop() tourne encore et encore, pour toujours
  digitalWrite(LED_PIN, HIGH); // LED allumee
  delay(TEMPS_ATTENTE); // Petite pause
  digitalWrite(LED_PIN, LOW); // LED eteinte
  delay(TEMPS_ATTENTE); // Petite pause
}
```

### Que fait le code ?

- Dans `setup()`, nous disons que la pin 13 est une sortie.
- Dans `loop()`, nous allumons la LED, nous attendons un peu, nous l'éteignons, puis nous attendons encore.
- Ensuite, `loop()` recommence tout simplement. Comme ça, la LED continue à clignoter.

### Essaie un peu...

- Fais clignoter la LED deux fois plus vite
- Transforme-la en signal SOS : court-court-court, long-long-long, court-court-court
- Fais devenir la LED doucement forte puis faible, comme si elle respirait

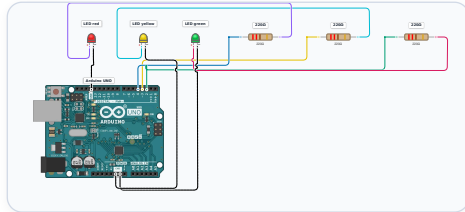
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/blink](https://codey.online/fr/starter-kits/uno/projects/blink) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★ Débutant ⌚ 15 minutes

## Feu tricolore

Tu construis un vrai feu tricolore avec une LED rouge, une LED jaune et une LED verte. Elles s'allument chacune leur tour, comme dans la rue.

### Schéma de branchement



### De quoi as-tu besoin ?



### Voici comment le construire

1. Place les trois LEDs côte à côte sur le breadboard.
2. Donne à chaque LED sa propre résistance de 220 Ω vers sa broche : rouge vers 4, jaune vers 3, verte vers 2.
3. Relie toutes les pattes courtes à la bande moins (-) du breadboard.
4. Relie la bande moins à GND de ton Arduino.

### Le code

```
// Feu tricolore
// Trois LEDs font exactement comme un vrai feu tricolore : vert, orange, rouge.

const int ROUGE = 4; // LED rouge sur broche 4
const int ORANGE = 3; // LED jaune/orange sur broche 3
const int VERT = 2; // LED verte sur broche 2

void setup() {
  pinMode(ROUGE, OUTPUT);
  pinMode(ORANGE, OUTPUT);
  pinMode(VERT, OUTPUT);
}

// Allume une petite lampe et éteint les deux autres
void mettreFeu(int allume) {
  digitalWrite(ROUGE, allume == ROUGE ? HIGH : LOW);
  digitalWrite(ORANGE, allume == ORANGE ? HIGH : LOW);
  digitalWrite(VERT, allume == VERT ? HIGH : LOW);
}

void loop() {
  mettreFeu(VERT); // Vert : tu peux avancer
  delay(5000);
  mettreFeu(ORANGE); // Orange : attention, ça va être rouge
  delay(2000);
  mettreFeu(ROUGE); // Rouge : stop !
  delay(5000);
}
```

#### Que fait le code ?

- Chaque LED a sa propre broche : rouge sur 4, jaune sur 3 et verte sur 2.
- La fonction `mettreFeu()` allume une petite lampe et éteint les deux autres.
- Dans `loop()`, on passe du vert à l'orange puis au rouge, avec un temps d'attente entre chaque.

#### Essaie un peu...

- Fais clignoter la petite lampe orange trois fois avant qu'elle devienne rouge
- Ajoute un bouton : quand tu appuies, le feu passe plus vite au rouge (un bouton piéton)
- Crée un mode nuit où seule la petite lampe orange clignote

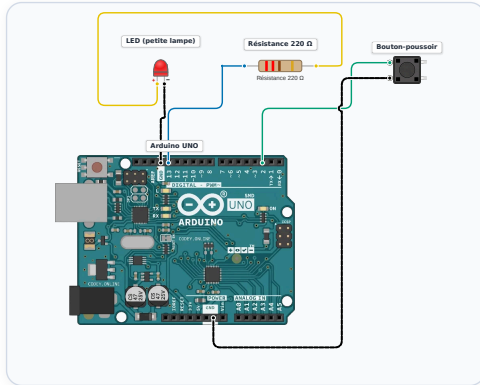
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/traffic-light](https://codey.online/fr/starter-kits/uno/projects/traffic-light) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★ Débutant ⌚ 15 minutes

# Bouton lumière

Tu fais un bouton lumière. Tu appuies une fois : la LED s'allume. Tu appuies encore une fois : la LED s'éteint. Comme ça, tu apprends comment l'Arduino "sent" un bouton.

## 🔌 Schéma de branchement



## 🛒 De quoi as-tu besoin ?



1x Arduino UNO



1x Bouton-poussoir



1x LED (petite lampe)



1x Résistance 220 Ω



Câble USB + Câbles jumper

## 🔧 Voici comment le construire

1. Place le bouton-poussoir au-dessus de la fente de ta breadboard.
2. Relie une patte à la pin 2 et une patte de l'autre côté à GND.
3. Place la LED avec une résistance de 220 Ω sur la pin 13, la patte courte vers GND.

## 💻 Le code

```
// Bouton lumière
// Appuie une fois sur le bouton : la LED s'allume. Encore une fois : la LED s'éteint.

const int BOUTON_PIN = 2; // Le bouton est sur la pin 2 (l'autre côté vers GND)
const int LED_PIN = 13; // La LED est sur la pin 13 (avec la résistance)

bool ledAllumee = false; // Retient si la LED est allumée ou éteinte
bool boutonPrecedent = HIGH; // Comment était le bouton la fois d'avant

void setup() {
  pinMode(BOUTON_PIN, INPUT_PULLUP); // Entrée avec résistance intégrée : relâché = HIGH, appuyé = LOW
  pinMode(LED_PIN, OUTPUT);
}

void loop() {
  bool bouton = digitalRead(BOUTON_PIN);

  // Réagir seulement au moment où tu appuies sur le bouton
  if (boutonPrecedent == HIGH && bouton == LOW) {
    ledAllumee = !ledAllumee; // Inverser : allumé devient éteint, éteint devient allumé
    digitalWrite(LED_PIN, ledAllumee ? HIGH : LOW);
    delay(50); // Petite pause contre les rebonds du bouton
  }

  boutonPrecedent = bouton;
}
```

### 🧠 Que fait le code ?

- Le bouton est sur la pin 2 avec `INPUT_PULLUP`. Relâché, la pin est HIGH. Appuyé, elle est LOW.
- Nous gardons en mémoire si la LED est allumée ou éteinte dans `ledAllumee`.
- Seulement au moment où tu appuies sur le bouton, nous inversons `ledAllumee`.

### 🎯 Essaie un peu...

- Fais clignoter la LED au lieu de la laisser allumée
- Fais s'éteindre la LED toute seule après 10 secondes
- Compte combien de fois on appuie sur le bouton et montre-le dans le moniteur série

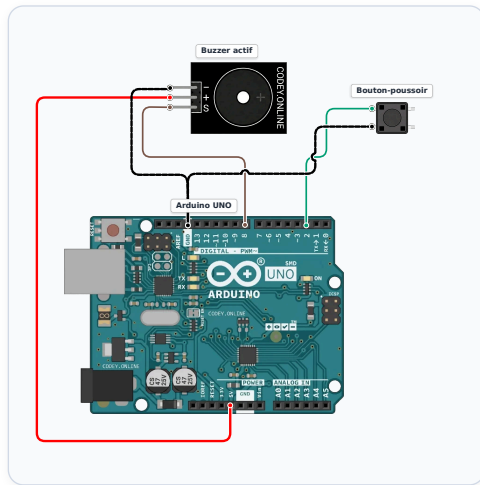
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/light-switch](https://codey.online/fr/starter-kits/uno/projects/light-switch) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★ Débutant ⌚ 10 minutes

# Sonnette

Tu fabriques ta propre sonnette. Appuie sur le bouton et le buzzer fait "ding-dong" !

## 🖨 Schéma de branchement



## 🛒 De quoi as-tu besoin ?



## 🔧 Voici comment le construire

1. Place le bouton-poussoir par-dessus la fente de la breadboard : une patte vers pin 2, et une patte de l'autre côté vers GND.
2. Branche le buzzer : S vers pin 8, + vers 5V et – vers GND.

## 💻 Le code

```
// Sonnette
// Appuie sur le bouton et le buzzer fait un joyeux "ding-dong".

const int BOUTON_PIN = 2; // Bouton-poussoir sur pin 2 (autre côté vers GND)
const int BUZZER_PIN = 8; // Buzzer actif : S sur pin 8

void setup() {
  pinMode(BOUTON_PIN, INPUT_PULLUP); // Libre = HIGH, appuyé = LOW
  pinMode(BUZZER_PIN, OUTPUT);
}

// Fait bip le buzzer un petit moment
void bip(int duree) {
  digitalWrite(BUZZER_PIN, HIGH);
  delay(duree);
  digitalWrite(BUZZER_PIN, LOW);
}

void loop() {
  if (digitalRead(BOUTON_PIN) == LOW) { // Quelqu'un appuie sur la sonnette !
    bip(300); // "Ding"
    delay(150);
    bip(600); // "Dong"
    delay(1000); // On attend un peu, pour que la sonnette ne sonne pas sans arrêt
  }
}
```

### 🧠 Que fait le code ?

- Le bouton est sur pin 2 et le buzzer sur pin 8.
- La fonction `bip()` allume le buzzer un petit moment, puis l'éteint.
- Quand le bouton est appuyé, on fait entendre un bip court et un bip long.

### 🎯 Essaie un peu...

- Joue une petite mélodie joyeuse de cinq bips
- Fais aussi clignoter une LED quand quelqu'un sonne
- Compte combien de fois on a sonné et montre-le dans le moniteur série

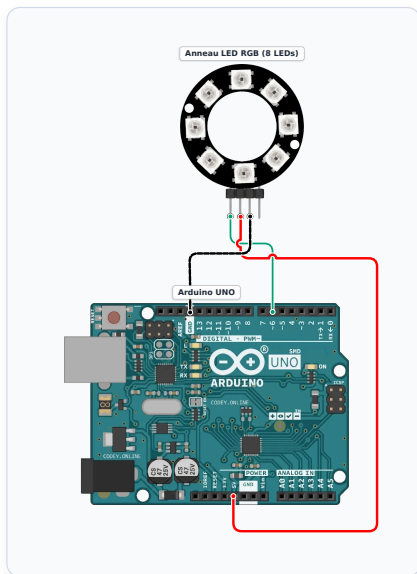
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/doorbell](https://codey.online/fr/starter-kits/uno/projects/doorbell) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★ Débutant ⌚ 10 minutes

## Anneau arc-en-ciel

Avec l'anneau LED, tu fais un arc-en-ciel qui tourne. Chaque LED reçoit une autre couleur. Et toutes les couleurs avancent en rond.

### Schéma de branchement



### De quoi as-tu besoin ?



### Voici comment le construire

1. Branche l'anneau LED : DI sur pin 6, 5V sur 5V et GND sur GND.
2. C'est tout ! Tu laisses DO libre.

### Le code

```
// Anneau arc-en-ciel
// L'anneau LED fait tourner toutes les couleurs de l'arc-en-ciel.

#include <Adafruit_NeoPixel.h>

const int BROCHE_ANNEAU = 6; // DI de l'anneau est sur pin 6
const int NB_LEDS = 8; // L'anneau a 8 LED
const int VITESSE = 20; // Plus c'est petit, plus ça va vite

Adafruit_NeoPixel anneau(NB_LEDS, BROCHE_ANNEAU, NEO_GRB + NEO_KHZ800);

void setup() {
  anneau.begin();
  anneau.setBrightness(40); // Pas trop fort : 0 = éteint, 255 = super fort
  anneau.show();
}

void loop() {
  // Décale l'arc-en-ciel petit à petit
  for (long couleur = 0; couleur < 65536; couleur += 512) {
    for (int i = 0; i < NB_LEDS; i++) {
      // Chaque LED reçoit une couleur un peu différente
      long couleurLed = couleur + (i * 65536 / NB_LEDS);
      anneau.setPixelColor(i, anneau.gamma32(anneau.ColorHSV(couleurLed)));
    }
    anneau.show();
    delay(VITESSE);
  }
}
```

#### Que fait le code ?

- Nous utilisons la bibliothèque `Adafruit_NeoPixel` pour commander l'anneau.
- `ColorHSV()` crée une couleur de l'arc-en-ciel. Chaque LED reçoit une couleur un peu plus loin dans l'arc-en-ciel.
- En décalant toujours un peu la couleur, l'arc-en-ciel semble tourner en rond.

#### Essaie un peu...

- Fais tourner l'arc-en-ciel deux fois plus vite
- Fais une lumière de police qui clignote en rouge et bleu
- Fais tourner une seule LED en rond, comme une petite lumière qui court

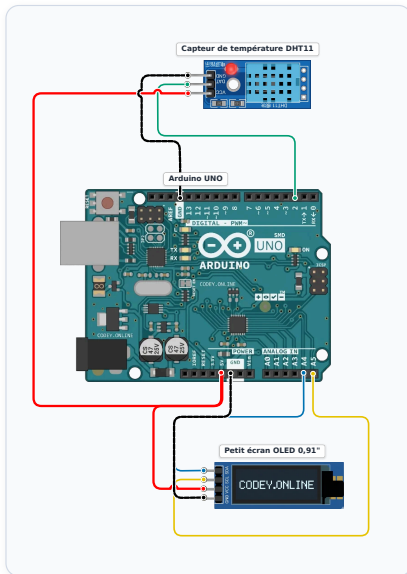
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/rainbow-ring](https://codey.online/fr/starter-kits/uno/projects/rainbow-ring) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★ Débutant avancé ⌚ 20 minutes

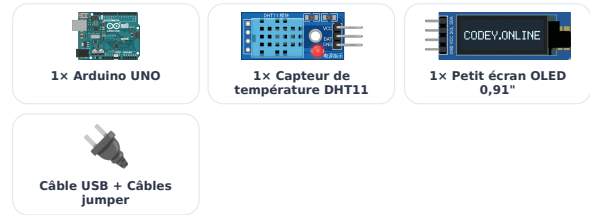
## Mini-station météo

Tu fabriques une mini-station météo. Le DHT11 mesure la température et l'humidité, et le petit écran OLED les affiche.

### Schéma de branchement



### De quoi as-tu besoin ?



### Voici comment le construire

1. Branche le DHT11 : DAT vers pin 2, VCC vers 5V et GND vers GND.
2. Branche l'écran OLED : SDA vers A4, SCL vers A5, VCC vers 5V et GND vers GND.
3. Astuce : deux pièces ont besoin de 5V et GND. Utilise les bandes + et - de ta breadboard.

### Le code

```
// Mini-station meteo
// Le DHT11 mesure la temperature et l'humidite. Le petit ecran OLED les
affiche.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

const int DHT_PIN = 2;          // DAT du DHT11 est sur la pin 2
DHT dht(DHT_PIN, DHT11);

Adafruit_SSD1306 ecran(128, 32, &Wire, -1); // OLED de 128 x 32 pixels (SDA =
A4, SCL = A5)

void setup() {
  Serial.begin(9600);
  dht.begin();
  if (!ecran.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // 0x3C est l'adresse de
l'ecran
    Serial.println(F("Ecran non trouve. Verifie les fils de SDA et SCL."));
    while (true); // Arret : sans ecran, cela
ne sert a rien
  }
  ecran.setTextColor(SSD1306_WHITE);
}

void loop() {
  float temperature = dht.readTemperature(); // En degres Celsius

  float humidite = dht.readHumidity();        // En pour cent

  ecran.clearDisplay();
  ecran.setTextSize(2);
  ecran.setCursor(0, 0);

  if (isnan(temperature) || isnan(humidite)) { // La mesure ne marche pas ?
    ecran.setTextSize(1);
    ecran.print(F("DHT11 introuvable"));
  } else {
    ecran.print(temperature, 1);
    ecran.print((char)247); // Le signe degre
    ecran.println(F("C"));
    ecran.print(humidite, 0);
    ecran.print(F("% hum."));

    Serial.print(F("Temperature: "));
    Serial.print(temperature);
    Serial.print(F(" C Humidite: "));
    Serial.print(humidite);
    Serial.println(F(" %"));
  }

  ecran.display(); // C'est seulement maintenant que tout arrive vraiment sur
l'ecran
  delay(2000); // Le DHT11 ne peut mesurer qu'une fois toutes les 2
secondes
}
```

#### Que fait le code ?

- Le DHT11 est sur la pin 2. Avec `readTemperature()` et `readHumidity()`, nous lisons les valeurs.
- L'écran OLED parle avec SDA (A4) et SCL (A5). Avec `print()`, nous mettons du texte sur l'écran.
- Toutes les 2 secondes, il y a une nouvelle mesure. La mesure ne marche pas ? Alors tu vois un message.

#### Essaie un peu...

- Affiche aussi la température en Fahrenheit
- Allume une LED s'il fait plus de 25 degrés
- Garde en mémoire la température la plus haute d'aujourd'hui et affiche-la aussi

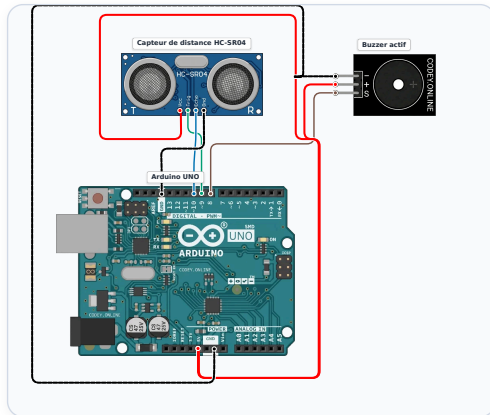
🔧 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/weather-station](https://codey.online/fr/starter-kits/uno/projects/weather-station) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★ ★ Débutant avancé ⌚ 20 minutes

# Capteur de stationnement

Tu construis un capteur de stationnement, comme dans une voiture. Plus tu t'approches du capteur, plus le buzzer bipe vite. Très près ? Alors il fait un long son.

## Schéma de branchement



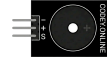
## De quoi as-tu besoin ?



1x Arduino UNO



1x Capteur de distance HC-SR04



1x Buzzer actif



Câble USB + Câbles jumper

## Voici comment le construire

1. Branche le HC-SR04 : Trig sur la broche 9, Echo sur la broche 10, Vcc sur 5V et Gnd sur GND.
2. Branche le buzzer : S sur la broche 8, + sur 5V et – sur GND.
3. Utilise les bandes + et – de ta breadboard pour 5V et GND.

## Le code

```
// Capteur de stationnement
// Plus tu t'approches du capteur, plus le buzzer bipe vite. Comme
// dans une voiture !

const int TRIG_PIN = 9; // Trig du HC-SR04 sur pin 9
const int ECHO_PIN = 10; // Echo du HC-SR04 sur pin 10
const int BUZZER_PIN = 8; // S du buzzer sur pin 8

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
}

// Mesure la distance en centimètres
long mesureDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH); // Envoie un petit son (que tu
  // n'entends pas)
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duree = pulseIn(ECHO_PIN, HIGH, 30000); // Combien de
  // temps avant le retour de l'écho ?

  if (duree == 0) return 999; // Pas d'écho :
  // rien tout près
  return duree / 58; // Transformer le
  // temps en centimètres
}

void loop() {
  long distance = mesureDistance();
  Serial.print(F("Distance: "));
  Serial.print(distance);
  Serial.println(F(" cm"));

  if (distance < 5) {
    digitalWrite(BUZZER_PIN, HIGH); // Très près : un long son
    delay(100);
  } else if (distance < 50) {
    digitalWrite(BUZZER_PIN, HIGH); // Petit bip
    delay(50);
    digitalWrite(BUZZER_PIN, LOW);
    delay(distance * 10); // Plus près = pause plus
    // courte = bips plus rapides
  } else {
    digitalWrite(BUZZER_PIN, LOW); // Loin : silence
    delay(100);
  }
}
```

### Que fait le code ?

- `mesureDistance()` envoie un petit son et mesure le temps que met l'écho pour revenir. On le transforme en centimètres.
- Plus loin que 50 cm : le buzzer est silencieux.
- Plus près : la pause entre les bips devient plus courte. Sous 5 cm, il bipe sans s'arrêter.

### Essaie un peu...

- Fais commencer les bips du capteur dès 1 mètre
- Ajoute une LED rouge qui s'allume quand c'est trop près
- Montre la distance dans le moniteur série, avec une petite barre d'étoiles

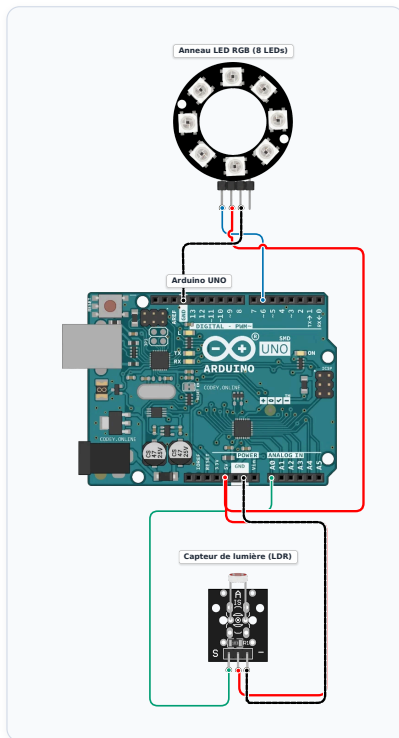
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/parking-sensor](https://codey.online/fr/starter-kits/uno/projects/parking-sensor) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★ Débutant avancé ⌚ 15 minutes

## Petite veilleuse

Tu fabriques une veilleuse intelligente. Quand il fait sombre, l'anneau LED s'allume doucement. Quand il fait clair, il s'éteint.

### Schéma de branchement



### De quoi as-tu besoin ?



### Voici comment le construire

1. Branche le capteur de lumière : S vers A0, + vers 5V et - vers GND.
2. Branche l'anneau LED : DI vers pin 6, 5V vers 5V et GND vers GND.
3. Ouvre le moniteur série et regarde quel nombre tu vois quand il fait clair et quand il fait sombre. Adapte `SEUIL_SOMBRE` avec ça.

### Le code

```
// Veilleuse
// Il fait sombre ? Alors l'anneau LED s'allume doucement. Il fait clair ? Alors
il s'éteint.

#include <Adafruit_NeoPixel.h>

const int LUMIERE_PIN = A0; // S du capteur de lumière sur A0
const int ANNEAU_PIN = 6; // DI de l'anneau LED sur pin 6
const int NOMBRE_LEDS = 8;

// Avec cette valeur, on trouve qu'il fait "sombre". Regarde dans le moniteur
série quel nombre tu vois
// quand il fait clair et quand il fait sombre, puis mets ici un nombre entre les
deux.
const int SEUIL_SOMBRE = 500;

Adafruit_NeoPixel anneau(NOMBRE_LEDS, ANNEAU_PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  Serial.begin(9600);
  anneau.begin();
  anneau.setBrightness(60);
  anneau.show();
}

void loop() {
  int lumiere = analogRead(LUMIERE_PIN); // Un nombre de 0 à 1023
  Serial.print(F("Capteur lum.: "));
  Serial.println(lumiere);

  // Avec ce capteur, le nombre monte quand il fait plus sombre.
  // Chez toi, c'est l'inverse ? Alors remplace > par <.
  bool sombre = lumiere > SEUIL_SOMBRE;

  for (int i = 0; i < NOMBRE_LEDS; i++) {
    if (sombre) {
      anneau.setPixelColor(i, anneau.Color(255, 140, 40)); // Lumière jaune
      // chaude
    } else {
      anneau.setPixelColor(i, 0); // Éteint
    }
  }
  anneau.show();
  delay(200);
}
```

#### Que fait le code ?

- Le capteur de lumière est sur A0. `analogRead()` donne un nombre de 0 à 1023.
- Le nombre est plus grand que `SEUIL_SOMBRE` ? Alors on trouve qu'il fait sombre.
- Dans le noir, on donne à toutes les petites lampes une couleur jaune chaude. Sinon, elles s'éteignent.

#### Essaie un peu...

- Fais briller la petite lampe de plus en plus fort quand il fait de plus en plus sombre
- Fais changer la couleur doucement, comme une lampe d'ambiance apaisante
- Fais s'éteindre la petite lampe toute seule après 1 minute

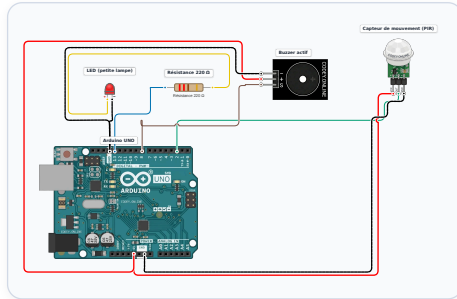
🔧 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/night-light](https://codey.online/fr/starter-kits/uno/projects/night-light) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★ Débutant avancé ⌚ 20 minutes

# Alarme de mouvement

Tu construis une alarme antivol. Le capteur de mouvement voit quelqu'un ? Alors le buzzer sonne et la LED rouge clignote.

## Schéma de branchement



## De quoi as-tu besoin ?



## Voici comment le construire

1. Branche le capteur PIR : OUT vers pin 2, VCC vers 5V et GND vers GND.
2. Branche le buzzer : S vers pin 8, + vers 5V et – vers GND.
3. Mets la LED rouge avec une résistance de 220 Ω sur pin 13, la petite patte vers GND.

## Le code

```
// Alarme de mouvement
// Quelqu'un passe ? Alors la LED rouge clignote et le buzzer sonne.

const int PIR_PIN = 2;      // OUT du capteur de mouvement sur pin 2
const int BUZZER_PIN = 8;   // S du buzzer sur pin 8
const int LED_PIN = 13;     // LED rouge sur pin 13 (via la résistance)

void setup() {
  Serial.begin(9600);
  pinMode(PIR_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(LED_PIN, OUTPUT);
  Serial.println(F("Le capteur chauffe... reste immobile !"));
  delay(10000);           // Le capteur a besoin d'un moment pour s'habituer à la pièce
  Serial.println(F("Alarme activée."));
}

void loop() {
  if (digitalRead(PIR_PIN) == HIGH) {    // Mouvement vu !
    Serial.println(F("Mouvement ! ALARME !"));
    for (int i = 0; i < 5; i++) {        // Sonner et clignoter 5 fois
      digitalWrite(LED_PIN, HIGH);
      digitalWrite(BUZZER_PIN, HIGH);
      delay(150);
      digitalWrite(LED_PIN, LOW);
      digitalWrite(BUZZER_PIN, LOW);
      delay(150);
    }
  }
  delay(100);
}
```

### Que fait le code ?

- Le capteur PIR est sur la broche 2. S'il voit un mouvement, la broche est HIGH.
- Au démarrage, on attend 10 secondes. Comme ça, le capteur s'habitue à la pièce.
- Quand il y a un mouvement, on fait sonner et clignoter 5 fois.

### Essaie un peu...

- Ajoute un bouton pour allumer et éteindre l'alarme
- Compte combien de fois il y a eu un mouvement et montre-le dans le moniteur série
- Fais sonner l'alarme plus longtemps et plus fort au deuxième mouvement

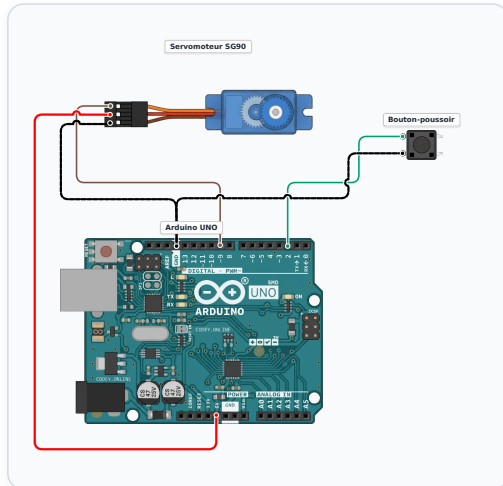
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/motion-alarm](https://codey.online/fr/starter-kits/uno/projects/motion-alarm) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★ Débutant avancé ⌚ 20 minutes

# Barrière

Tu construis une barrière, comme dans un parking. Appuie sur le bouton : le servo fait monter la barrière, attend un peu, puis la referme.

## 🖨 Schéma de branchement



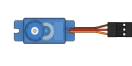
## 🛒 De quoi as-tu besoin ?



1x Arduino UNO



1x Bouton-poussoir



1x Servomoteur SG90



Câble USB + Câbles jumper

## 🔧 Voici comment le construire

1. Place le bouton-poussoir par-dessus la fente de ta breadboard : une patte vers pin 2, une patte de l'autre côté vers GND.
2. Branche le servo : orange vers pin 9, rouge vers 5V et marron vers GND.
3. Colle une paille ou un bâtonnet de glace sur le bras du servo : c'est ta barrière !

## 💻 Le code

```
// Barrière
// Appuie sur le bouton : la barrière (servo) monte, attend un
// peu puis se referme.

#include <Servo.h>

const int BOUTON_PIN = 2;    // Bouton-poussoir sur pin 2
                             // (l'autre côté vers GND)
const int SERVO_PIN = 9;    // Fil orange/jaune du servo sur pin
                             // 9

const int FERME = 0;        // Angle du servo quand la barrière
                             // est fermée
const int OUVERT = 90;      // Angle du servo quand la
                             // barrière est ouverte

Servo barriere;

void setup() {
  pinMode(BOUTON_PIN, INPUT_PULLUP);
  barriere.attach(SERVO_PIN);

  barriere.write(FERME);    // On commence avec la barrière
                             // fermée
}

void loop() {
  if (digitalRead(BOUTON_PIN) == LOW) { // Bouton appuyé
    // Tourner doucement pour ouvrir, degré par degré
    for (int angle = FERME; angle <= OUVERT; angle++) {
      barriere.write(angle);
      delay(15);
    }
    delay(3000);              // 3 secondes ouverte :
    // la voiture passe
    // Refermer doucement
    for (int angle = OUVERT; angle >= FERME; angle--) {
      barriere.write(angle);
      delay(15);
    }
  }
}
```

### 🧠 Que fait le code ?

- Nous utilisons la bibliothèque `Servo`. Avec `write()`, tu dis vers quel angle le servo doit aller.
- Après un appui sur le bouton, nous tournons degré par degré de 0 à 90. Comme ça, c'est tout doux.
- Après 3 secondes, nous revenons à 0 : la barrière est fermée.

### 🎯 Essaie un peu...

- Fais ouvrir et fermer la barrière plus vite
- Fais biper un buzzer pendant que la barrière bouge
- Allume une LED rouge quand la barrière est fermée et une verte quand elle est ouverte

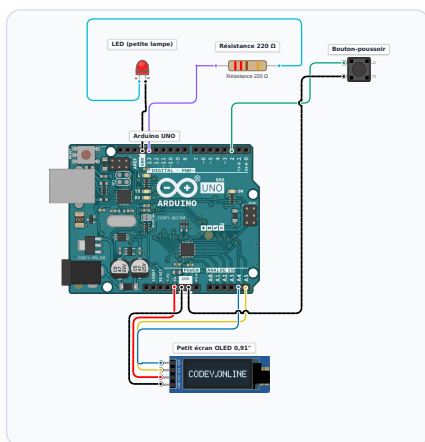
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/barrier](https://codey.online/fr/starter-kits/uno/projects/barrier) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★ Débutant avancé ⌚ 25 minutes

## Jeu de réflexes

À quelle vitesse réagis-tu ? Attends que la LED s'allume. Puis appuie le plus vite possible sur le bouton. L'écran montre ton temps en millisecondes. Défie tes amis !

### Schéma de branchement



### De quoi as-tu besoin ?



### Voici comment le construire

- Place le bouton-poussoir au-dessus de la fente du breadboard : une patte vers pin 2, une patte de l'autre côté vers GND.
- Place la LED avec une résistance de 220 Ω sur pin 13, la patte courte vers GND.
- Branche l'écran OLED : SDA vers A4, SCL vers A5, VCC vers 5V et GND vers GND.

### Le code

```
// Jeu de réflexes
// Attends que la LED s'allume, puis appuie le plus vite possible sur le bouton.
// L'écran montre à quelle vitesse tu as réagi !

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const int BOUTON_PIN = 2; // Bouton sur pin 2 (autre côté vers GND)
const int LED_PIN = 13; // LED sur pin 13 (avec la résistance)

Adafruit_SSD1306 ecran(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

unsigned long meilleurTemps = 0; // Ton meilleur temps jusqu'ici

// Affiche un texte en grand sur l'écran
void affiche(const __FlashStringHelper* texteHaut, unsigned long nombre) {
  ecran.clearDisplay();
  ecran.setTextSize(1);
  ecran.setCursor(0, 0);
  ecran.print(texteHaut);
  if (nombre > 0) {
    ecran.setTextSize(2);
    ecran.setCursor(0, 14);
    ecran.print(nombre);
    ecran.print(F(" ms"));
  }
  ecran.display();
}

void setup() {
  pinMode(BOUTON_PIN, INPUT_PULLUP);
  pinMode(LED_PIN, OUTPUT);
  randomSeed(anaLogRead(A0)); // Donne des temps d'attente vraiment au
  hasard
  ecran.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  ecran.setTextColor(SSD1306_WHITE);
}

void loop() {
  affiche(F("Presse bouton..."), 0);
  while (digitalRead(BOUTON_PIN) == HIGH); // Attendre que tu appuies sur le bouton
  while (digitalRead(BOUTON_PIN) == LOW); // ...puis que tu le relâches
  affiche(F("Pret? LED!"), 0);

  delay(random(2000, 5000)); // Un temps secret entre 2 et 5 secondes

  if (digitalRead(BOUTON_PIN) == LOW) { // Appuyé trop tôt ? Tricheur !
    affiche(F("Trop tot! Encore."), 0);
    delay(2000);
    return;
  }

  digitalWrite(LED_PIN, HIGH); // MAINTENANT !
  unsigned long start = millis();
  while (digitalRead(BOUTON_PIN) == HIGH); // Attendre ton appui
  unsigned long temps = millis() - start;
  digitalWrite(LED_PIN, LOW);

  if (meilleurTemps == 0 || temps < meilleurTemps) {
    meilleurTemps = temps;
    affiche(F("NOUV. RECORD!"), temps);
  } else {
    affiche(F("Ton temps:"), temps);
  }
  delay(3000);
}
```

#### Que fait le code ?

- Après un appui sur le bouton, on attend un temps secret entre 2 et 5 secondes ( `random()` ).
- Puis la LED s'allume et on lance un chronomètre avec `millis()`.
- Quand tu appuies, on calcule combien de millisecondes se sont écoulées. On garde ton meilleur temps.

#### Essaie un peu...

- Fais-en un jeu pour deux joueurs, avec un bouton pour chacun
- Fais biper un buzzer quand il y a un nouveau record
- Joue 5 manches, puis affiche ton temps moyen

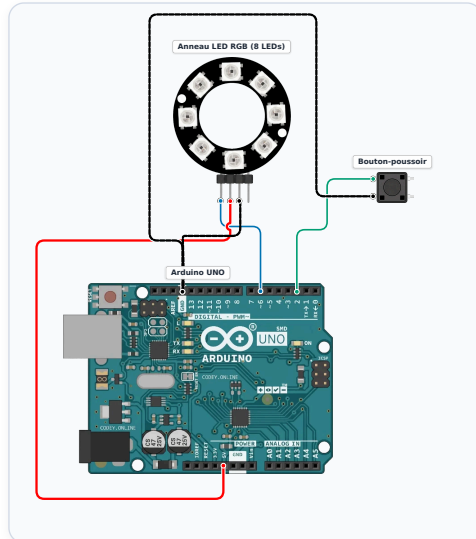
🔧 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/reaction-game](https://codey.online/fr/starter-kits/uno/projects/reaction-game) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★ Débutant avancé ⌚ 15 minutes

# Dé électronique

Tu fabriques un dé de lumière. Appuie sur le bouton : les lumières roulent un instant, puis tu vois combien tu as lancé. De 1 à 6 lumières !

## Schéma de branchement



## De quoi as-tu besoin ?



## Voici comment le construire

1. Place le bouton-poussoir par-dessus la fente du breadboard : une patte vers pin 2, une patte de l'autre côté vers GND.
2. Branche l'anneau LED : DI vers pin 6, 5V vers 5V et GND vers GND.

## Le code

```
// Dé électronique
// Appuie sur le bouton : l'anneau LED roule un peu puis allume 1 à 6 lumières.

#include <Adafruit_NeoPixel.h>

const int BOUTON_PIN = 2; // Bouton-poussoir sur pin 2 (autre côté vers GND)
const int RING_PIN = 6; // DI de l'anneau LED sur pin 6
const int NOMBRE_LEDS = 8;

Adafruit_NeoPixel ring(NOMBRE_LEDS, RING_PIN, NEO_GRB + NEO_KHZ800);

// Allume quelques lumières avec une seule couleur
void afficherNombre(int nombre, uint32_t couleur) {
  ring.clear();
  for (int i = 0; i < nombre; i++) {
    ring.setPixelColor(i, couleur);
  }
  ring.show();
}

void setup() {
  pinMode(BOUTON_PIN, INPUT_PULLUP);
  randomSeed(analogRead(A0)); // Pour que le dé soit vraiment au hasard
  ring.begin();
  ring.setBrightness(50);
  ring.show();
}

void loop() {
  if (digitalRead(BOUTON_PIN) == LOW) {
    // Petit « roulement » : nombres rapides, puis de plus en plus lents
    for (int i = 0; i < 15; i++) {
      afficherNombre(random(1, 7), ring.Color(0, 80, 255));
      delay(40 + i * 10);
    }
    int lancer = random(1, 7); // Un nombre de 1 à 6
    afficherNombre(lancer, ring.Color(0, 255, 60)); // Le vrai nombre en vert
    delay(500);
  }
}
```

### Que fait le code ?

- `random(1, 7)` choisit un nombre au hasard de 1 à 6.
- D'abord, nous montrons des nombres qui changent vite, en bleu. Comme ça, on dirait que le dé roule.
- Ensuite, `afficherNombre()` montre le vrai nombre en vert.

### Essaie un peu...

- Donne une couleur à chaque nombre
- Fais biper un buzzer un court instant quand le dé s'arrête
- Fais-en un dé de 1 à 8

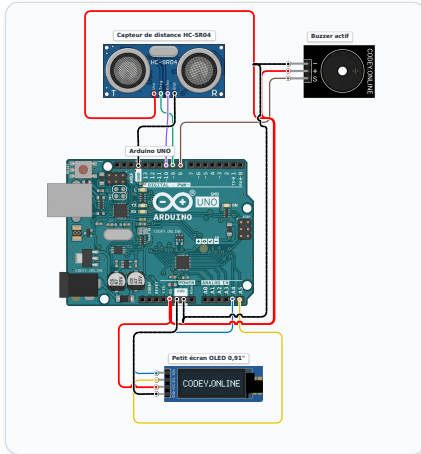
🔧 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/dice](https://codey.online/fr/starter-kits/uno/projects/dice) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★★ Défi ⌚ 30 minutes

## Capteur de parking luxe

La version luxe du capteur de parking ! Il bippe plus vite quand tu t'approches. Et l'écran OLED montre la distance en centimètres. Avec une petite barre qui devient plus longue quand tu t'approches.

### Schéma de branchement



### De quoi as-tu besoin ?



### Voici comment le construire

1. Branche le HC-SR04 : Trig vers la broche 9, Echo vers la broche 10, Vcc vers 5V et Gnd vers GND.
2. Branche le buzzer : S vers la broche 8, + vers 5V et - vers GND.
3. Branche l'écran OLED : SDA vers A4, SCL vers A5, VCC vers 5V et GND vers GND.
4. Trois pièces ont besoin de 5V et de GND. Utilise les bandes + et - de ta breadboard.

### Le code

```
// Capteur de parking luxe
// Comme le capteur de parking normal, mais maintenant tu vois aussi la
// distance sur l'écran OLED.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

const int TRIG_PIN = 9; // Trig du HC-SR04 sur la broche 9
const int ECHO_PIN = 10; // Echo du HC-SR04 sur la broche 10
const int BUZZER_PIN = 8; // S du buzzer sur la broche 8

Adafruit_SSD1306 ecran(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

// Mesure la distance en centimètres
long mesurerDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long temps = pulseIn(ECHO_PIN, HIGH, 30000);
  if (temps == 0) return 999; // Pas d'écho : rien tout près
  return temps / 58;
}

// Met la distance en grand sur l'écran, avec une petite barre dessous
void afficherDistance(long distance) {
  ecran.clearDisplay();
  ecran.setTextSize(2);
  ecran.setCursor(0, 0);
  if (distance >= 999) {
    ecran.print(F("..."));
  } else {
    ecran.print(distance);
    ecran.print(F(" cm"));
  }
  // Petite barre : plus c'est près, plus elle est longue
  int longueur = map(constrain(distance, 0, 100), 0, 100, 128, 0);
  ecran.fillRect(0, 24, longueur, 8, SSD1306_WHITE);
  ecran.display();
}

void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  ecran.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  ecran.setTextColor(SSD1306_WHITE);
}

void loop() {
  long distance = mesurerDistance();
  afficherDistance(distance);

  if (distance < 5) {
    digitalWrite(BUZZER_PIN, HIGH); // Très près : un long son
    delay(100);
  } else if (distance < 50) {
    digitalWrite(BUZZER_PIN, HIGH); // Petit bip court
    delay(50);
    digitalWrite(BUZZER_PIN, LOW);
    delay(distance * 10); // Plus près = bips plus vite
  } else {
    digitalWrite(BUZZER_PIN, LOW);
    delay(100);
  }
}
```

#### Que fait le code ?

- `mesurerDistance()` mesure la distance en centimètres avec le HC-SR04.
- `afficherDistance()` met le nombre en grand sur l'écran et dessine une petite barre avec `fillRect()`.
- Le buzzer bippe plus vite quand la distance devient plus petite.

#### Essaie un peu...

- Affiche "STOP!" sur l'écran si tu es à moins de 5 cm
- Affiche la distance en mètres si quelque chose est à plus de 1 mètre
- Ajoute l'anneau LED qui passe du vert au rouge

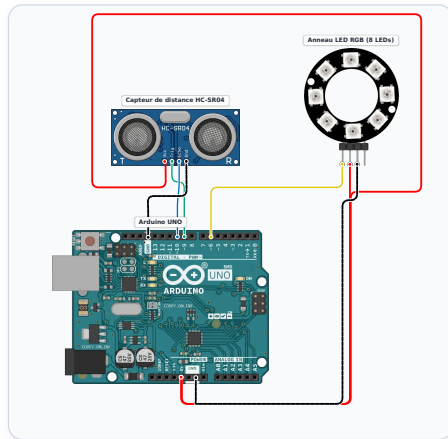
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/parking-sensor-deluxe](https://codey.online/fr/starter-kits/uno/projects/parking-sensor-deluxe) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★★ Défi ⌚ 25 minutes

# Télémètre avec barre lumineuse

Tu fabriques une barre lumineuse qui montre à quelle distance se trouve un objet. Loin : aucune lampe. Plus c'est proche, plus il y a de lampes qui s'allument. D'abord vert, puis orange, et tout près rouge.

## Schéma de branchement



## De quoi as-tu besoin ?

  
1x Arduino UNO  
1x Capteur de distance HC-SR04  
1x Anneau LED RGB (8 LEDs)  
Câble USB + Câbles jumper

## Voici comment le construire

1. Branche le HC-SR04 : Trig vers la pin 9, Echo vers la pin 10, Vcc vers 5V et Gnd vers GND.
2. Branche l'anneau LED : DI vers la pin 6, 5V vers 5V et GND vers GND.

## Le code

```
// Télémètre avec barre lumineuse
// Plus tu t'approches, plus il y a de lampes allumées : vert, puis orange, puis rouge.

#include <Adafruit_NeoPixel.h>

const int TRIG_PIN = 9;    // Trig du HC-SR04 sur la pin 9
const int ECHO_PIN = 10;   // Echo du HC-SR04 sur la pin 10
const int RING_PIN = 6;    // DI de l'anneau LED sur la pin 6
const int NB_LEDS = 8;
const int DISTANCE_MAX = 80; // Plus loin que ça (en cm) : toutes les lampes éteintes

Adafruit_NeoPixel anneau(NB_LEDS, RING_PIN, NEO_GRB + NEO_KHZ800);

long mesurerDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duree = pulseIn(ECHO_PIN, HIGH, 30000);
  if (duree == 0) return 999;
  return duree / 58;
}

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  anneau.begin();

  anneau.setBrightness(40);
  anneau.show();
}

void loop() {
  long distance = mesurerDistance();
  Serial.print(F("Distance: "));
  Serial.print(distance);
  Serial.println(F(" cm"));

  // Combien de lampes doivent s'allumer ? Près = 8, loin = 0
  int nombre = map(constrain(distance, 0, DISTANCE_MAX), 0, DISTANCE_MAX, NB_LEDS, 0);

  for (int i = 0; i < NB_LEDS; i++) {
    if (i < nombre) {
      if (i < 3) anneau.setPixelColor(i, anneau.Color(0, 255, 0));
      // Vert
      else if (i < 6) anneau.setPixelColor(i, anneau.Color(255, 120, 0));
      // Orange
      else anneau.setPixelColor(i, anneau.Color(255, 0, 0));
      // Rouge
    } else {
      anneau.setPixelColor(i, 0);
      // Éteint
    }
  }
  anneau.show();
  delay(100);
}
```

### Que fait le code ?

- `mesurerDistance()` mesure la distance avec le HC-SR04.
- `map()` transforme la distance en nombre de lampes : 0 à 8.
- Les 3 premières lampes sont vertes, les 3 suivantes orange et les 2 dernières rouges.

### Essaie un peu...

- Fais clignoter toutes les lampes en rouge si tu es à moins de 10 cm
- Ajoute le buzzer qui bippe quand c'est rouge
- Rends le mètre plus sensible : toutes les lampes allumées à 40 cm au lieu de 80 cm

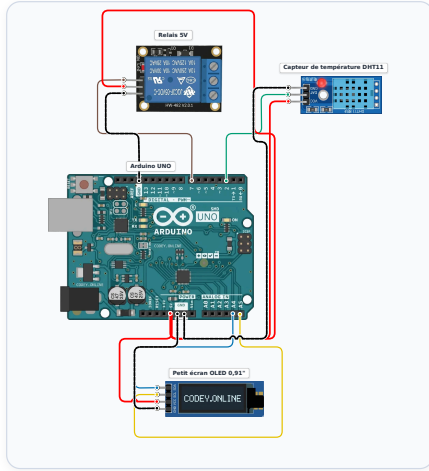
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/distance-light-bar](https://codey.online/fr/starter-kits/uno/projects/distance-light-bar) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★★ Défi ⌚ 30 minutes

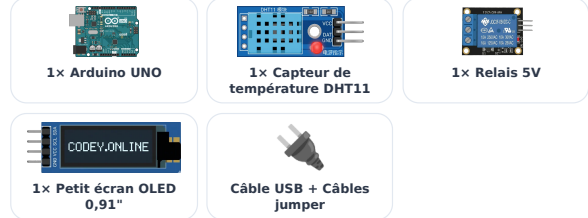
## Ventilateur malin

Tu construis un thermostat. S'il fait plus chaud que 25 degrés, le relais clique et s'allume. Comme ça, ton Arduino peut par exemple allumer un petit ventilateur sur une pile. Sur l'écran, tu vois la température et si le ventilateur est allumé.

### Schéma de branchement



### De quoi as-tu besoin ?



### Voici comment le construire

1. Branche le DHT11 : DAT vers pin 2, VCC vers 5V et GND vers GND.
2. Branche le relais : S vers pin 7, + vers 5V et - vers GND.
3. Branche l'écran OLED : SDA vers A4, SCL vers A5, VCC vers 5V et GND vers GND.
4. Tu veux vraiment allumer quelque chose ? Utilise seulement une pile ou une basse tension sur COM et NO, jamais la prise de courant.

### Le code

```
// Ventilateur malin (thermostat)
// Fait-il trop chaud ? Alors le relais s'allume, par exemple pour un ventilateur.
// Sur l'écran, tu vois la température et si le ventilateur est allumé.

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <DHT.h>

const int DHT_PIN = 2;      // DAT du DHT11 sur pin 2
const int RELAIS_PIN = 7;   // S du relais sur pin 7
const float MARCHE_AU_DESSUS = 25.0; // Au-dessus de cette température, le ventilateur s'allume
const float ARRET_EN_DESSOUS = 24.0; // Sous cette température, il s'éteint de nouveau

DHT dht(DHT_PIN, DHT11);
Adafruit_SSD1306 ecran(128, 32, &Wire, -1); // OLED: SDA = A4, SCL = A5

bool ventilateurAllume = false;

void setup() {
  Serial.begin(9600);
  pinMode(RELAIS_PIN, OUTPUT);
  digitalWrite(RELAIS_PIN, LOW); // Commence avec le relais éteint
  dht.begin();
  ecran.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  ecran.setTextColor(SSD1306_WHITE);
}

void loop() {
  float temperature = dht.readTemperature();

  if (!isnan(temperature)) {
    // Deux limites, pour que le relais ne clique pas sans arrêt
    if (temperature > MARCHE_AU_DESSUS) ventilateurAllume = true;
    if (temperature < ARRET_EN_DESSOUS) ventilateurAllume = false;
    digitalWrite(RELAIS_PIN, ventilateurAllume ? HIGH : LOW);
  }

  ecran.clearDisplay();
  ecran.setTextSize(2);
  ecran.setCursor(0, 0);
  if (isnan(temperature)) {
    ecran.print(F("DHT11 ?"));
  } else {
    ecran.print(temperature, 1);
    ecran.print((char)247);
    ecran.print(F("C"));
  }
  ecran.setTextSize(1);
  ecran.setCursor(0, 22);
  ecran.print(ventilateurAllume ? F("Ventilo: ON") : F("Ventilo: off"));
  ecran.display();

  Serial.print(F("Temperature: "));
  Serial.print(temperature);
  Serial.println(ventilateurAllume ? F(" C -> ON") : F(" C -> off"));

  delay(2000);
}
```

#### Que fait le code ?

- Le DHT11 mesure la température. Au-dessus de 25 degrés, nous allumons le relais.
- Le relais s'éteint seulement sous 24 degrés. Avec ces deux limites, il ne clique pas sans arrêt.
- L'écran OLED affiche la température et "Ventilo: ON" ou "off".

#### Essaie un peu...

- Ajoute deux boutons pour monter et baisser la température limite
- Allume une LED tant que le ventilateur est allumé
- Fais aussi démarrer le ventilateur si l'humidité de l'air dépasse 70%

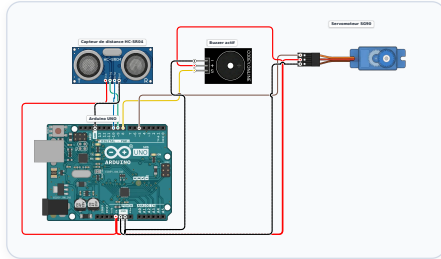
🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/thermostat](https://codey.online/fr/starter-kits/uno/projects/thermostat) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.

★★★★ Défi ⌚ 30 minutes

## Barrière automatique

Voici la version maligne de la barrière ! Le capteur de distance voit qu'une voiture arrive. Alors le buzzer fait bip et la barrière s'ouvre toute seule. La voiture est partie ? Alors elle se referme.

### Schéma de branchement



### De quoi as-tu besoin ?



### Voici comment le construire

1. Branche le HC-SR04 : Trig vers la broche 9, Echo vers la broche 10, VCC vers 5V et GND vers GND.
2. Branche le servo : orange vers la broche 5, rouge vers 5V et marron vers GND.
3. Branche le buzzer : S vers la broche 8, + vers 5V et - vers GND.
4. Colle une paille ou un bâtonnet de glace sur le servo comme barrière et place le capteur devant.

### Le code

```
// Barriere automatique
// Une voiture arrive (quelque chose a moins de 15 cm) ? Alors le buzzer
// fait bip et la barriere s'ouvre.

#include <Servo.h>

const int TRIG_PIN = 9;      // Trig du HC-SR04 sur la broche 9
const int ECHO_PIN = 10;     // Echo du HC-SR04 sur la broche 10
const int SERVO_PIN = 5;     // Fil orange/jaune du servo sur la broche 5
const int BUZZER_PIN = 8;    // S du buzzer sur la broche 8
const int PROCHE = 15;      // A partir de cette distance (cm), la
                             // barriere s'ouvre

const int FERME = 0;         // Angle quand la barriere est fermee
const int OUVERT = 90;       // Angle quand la barriere est ouverte

Servo barriere;

long mesurerDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long temps = pulseIn(ECHO_PIN, HIGH, 30000);
  if (temps == 0) return 999;
  return temps / 58;
}

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  barriere.attach(SERVO_PIN);
  barriere.write(FERME);
}

void loop() {
  long distance = mesurerDistance();

  if (distance < PROCHE) {
    Serial.println(F("Voiture vue ! Barriere ouverte.));
    digitalWrite(BUZZER_PIN, HIGH); // Petit bip
    delay(100);
    digitalWrite(BUZZER_PIN, LOW);

    barriere.write(OUVERT);
    delay(1000);
    // Rester ouvert tant que la voiture est encore la
    while (mesurerDistance() < PROCHE) {
      delay(200);
    }
    delay(2000); // Attendre encore un peu que la
    // voiture soit bien partie
    barriere.write(FERME);
    Serial.println(F("Barriere fermee.));
  }
  delay(100);
}
```

#### Que fait le code ?

- Le HC-SR04 mesure sans arrêt la distance. Moins de 15 cm ? Alors il y a une voiture.
- Le buzzer fait un petit bip et le servo tourne à 90 degrés : ouvert.
- Tant que la voiture est là, la barrière reste ouverte. Après, on attend 2 secondes et elle se ferme.

#### Essaie un peu...

- Compte combien de voitures sont passées et montre-le dans le moniteur série
- Ajoute une LED rouge et une LED verte comme feu près de la barrière
- Fais ouvrir la barrière seulement si on appuie aussi sur un bouton

🚀 Construis ce projet avec Codey : ouvre [codey.online/fr/starter-kits/uno/projects/automatic-barrier](https://codey.online/fr/starter-kits/uno/projects/automatic-barrier) et clique sur FAIS ÇA. Alors le code et le schéma de branchement sont tout de suite prêts.