

Merci d'avoir choisi le **Beginner's Lab Kit**

Bienvenue dans le kit de laboratoire pour débutants, un pack de démarrage complet conçu spécialement pour les nouveaux venus dans le monde de l'électronique et de la programmation. Ce kit comprend une gamme de composants essentiels tels que des LEDs, des résistances, un buzzer, des potentiomètres, des photo-résistances, des thermistances, des boutons poussoirs, des tubes numériques et un module ultrasonique. L'une des caractéristiques remarquables de ce kit est l'inclusion d'un multimètre, un outil précieux qui vous permet de mesurer le courant, la tension et la résistance dans vos circuits. Cet ajout est particulièrement utile pour approfondir votre compréhension du fonctionnement de chaque composant.

La séquence de cours fournie avec ce kit est structurée autour de la syntaxe de programmation Arduino, garantissant une progression logique et éducative. Cette structure vous permet de construire des circuits étape par étape tout en apprenant à écrire les programmes qui les contrôlent. Tout au long du cours, vous rencontrerez des défis de dépannage qui renforceront votre compréhension des concepts.

Pour toute question ou assistance, n'hésitez pas à nous contacter à l'adresse service@sunfounder.com. Plongez dans votre parcours d'apprentissage avec le kit de laboratoire pour débutants et commencez à construire, coder et explorer le monde passionnant de l'électronique !

Sommaire

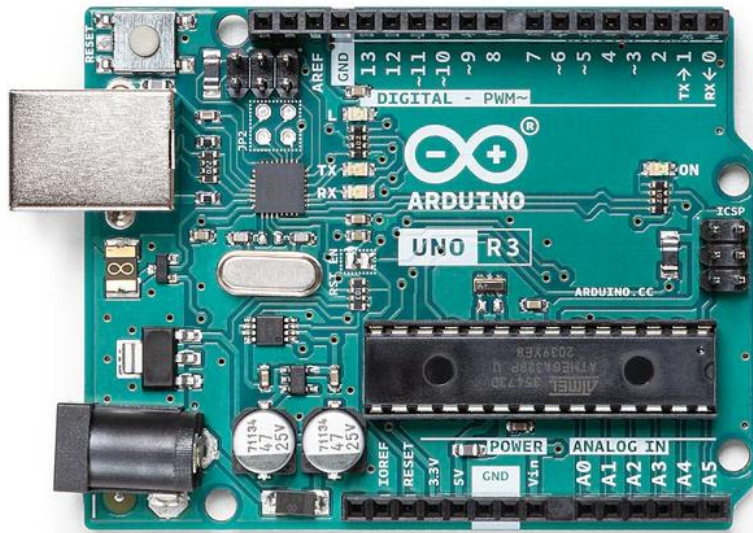
Contenu de votre kit	1
Leçon 2 : Votre premier circuit	8
Leçon 3 : Mesurer avec un multimètre	9
Leçon 4 : Loi d'Ohm	10
Leçon 5 : Circuit en série vs. circuit en parallèle	11
Leçon 6 : Faire clignoter une LED	15
Leçon 7 : Fabriquons des feux de circulation !	16
Leçon 8 : Feu de circulation avec bouton piéton	17
Leçon 9 : Lampe de bureau dimmable	19
Leçon 10 : Lampe de bureau ON/OFF	20
Leçon 11 : Contrôle des matrices LED avec un potentiomètre	21
Leçon 12 : Les Couleurs de l'Arc-en-Ciel	22
Leçon 13 : Le Spectre de la Vue	23
Leçon 14 : Couleurs aléatoires	24
Leçon 15 : Couleurs froides ou chaudes	26
Leçon 16 : Alarme de Température	27
Leçon 17 : Code Morse	28
Leçon 18 : Alarme lumineuse	29
Leçon 19 : Système d'alarme de stationnement inversé	30
Leçon 20 : Le minuteur Pomodoro	31
Leçon 21 : Son de sirène	33
Leçon 22 : Jouer « Ah ! vous dirai-je, maman »	35
Leçon 23 : Dé cybernétique	36
Leçon 24 : Lumière défilante avec le 74HC595	38
Leçon 25 : Afficher un nombre	39

Contenu de votre kit

Dans notre kit, vous trouverez une variété de composants et de pièces que vous utiliserez tout au long de ce cours pour assembler des circuits. Voici un guide rapide de ce qui est inclus.

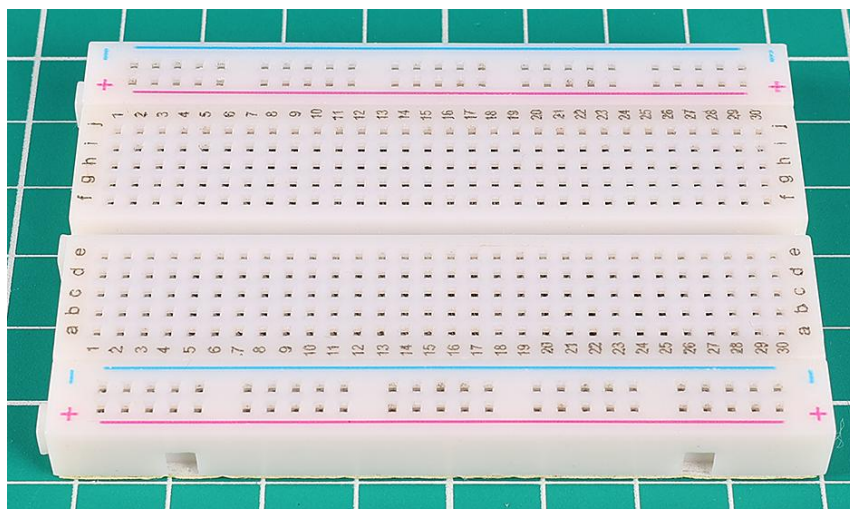
1 x Original Arduino Uno R3

Une carte microcontrôleur qui est le cerveau de vos circuits. Elle contient tout ce qu'il faut pour prendre en charge le microcontrôleur ; il suffit de la connecter à votre ordinateur avec un câble USB ou de l'alimenter avec un adaptateur AC-DC ou une batterie pour commencer.



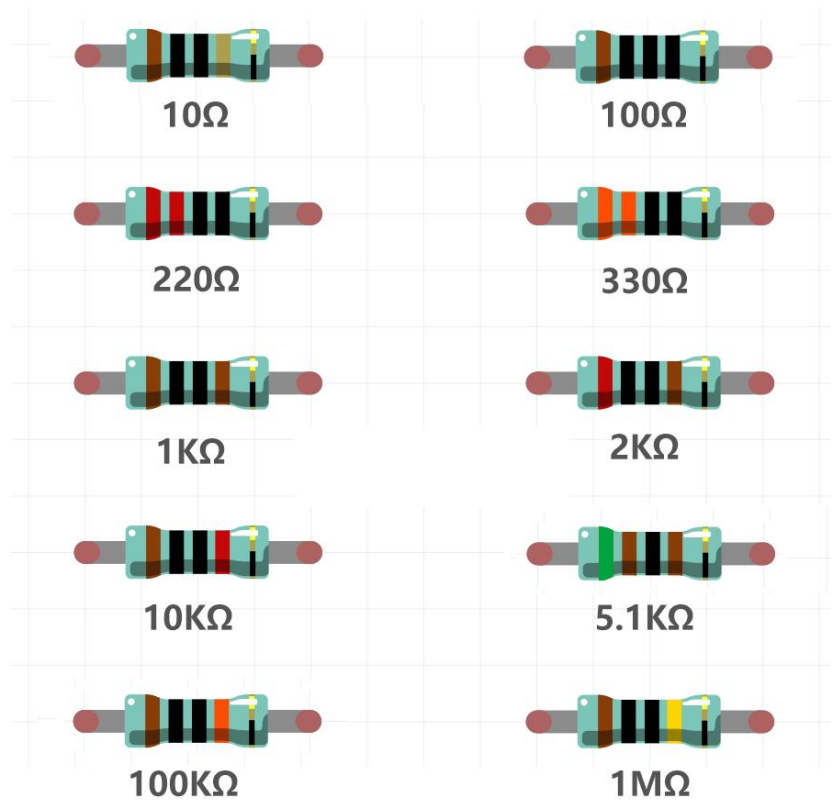
1 x Plaque d'essai 400 trous

Une plaque sans soudure qui vous permet de construire facilement des circuits électroniques. Elle est remplie de rangées de trous pour connecter des fils et des composants.



120 x Résistances (10 de chaque, 30 de 220Ω)

Une résistance est un composant qui limite le flux d'électricité, modifiant ainsi la tension et le courant dans un circuit. La valeur d'une résistance est mesurée en ohms, symbolisés par la lettre grecque oméga (Ω). Les bandes colorées sur une résistance indiquent sa valeur de résistance et sa tolérance.



25 x LEDs (5 de chaque couleur)

Cette sélection de LEDs colorées comprend cinq couleurs : rouge, vert, bleu, jaune et blanc, répondant à divers besoins d'éclairage et de signalisation. Adaptées aux applications allant des indicateurs d'état simples aux projets d'éclairage décoratif complexes, ces LEDs offrent un choix de couleurs riche pour améliorer l'attrait visuel de tout projet électronique.



2 x RGB LEDs

Ces LEDs combinent des LEDs rouge, verte et bleue dans un seul boîtier. Elles peuvent afficher différentes couleurs en ajustant la tension d'entrée, permettant de créer des millions de teintes.



1 x Photoresistor

Une photoresistance est un composant sensible à la lumière qui modifie sa résistance en fonction de l'intensité lumineuse à laquelle il est exposé. Elle est idéale pour créer des contrôles et capteurs activés par la lumière dans des projets électroniques.



1 x NTC Thermistor

Une thermistance est une résistance sensible aux variations de température. Les thermistances NTC diminuent leur résistance lorsque la température augmente, tandis que les thermistances PTC augmentent leur résistance avec la température.



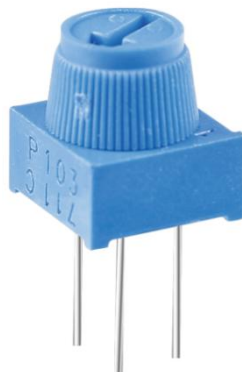
1 x Active Buzzer & 1 x Passive Buzzer

Un buzzer, disponible en version active et passive, est un dispositif de signalisation sonore qui émet un son lorsqu'un courant électrique est appliqué. Il est couramment utilisé dans les alarmes, minuteries et systèmes de notification.



1 x Potentiometer

Un potentiomètre est une résistance variable à trois broches. Deux broches se connectent aux extrémités d'une résistance, tandis que la broche centrale est reliée à un curseur mobile, divisant ainsi la résistance en deux parties. Les potentiomètres, souvent utilisés pour ajuster la tension dans les circuits, fonctionnent de manière similaire aux boutons de volume des radios.



10 x Petits boutons-poussoirs

Un petit bouton-poussoir est utilisé pour fournir une réponse physique lorsqu'il est pressé. Il est couramment utilisé dans les appareils électroniques pour déclencher des actions ou entrer des commandes.



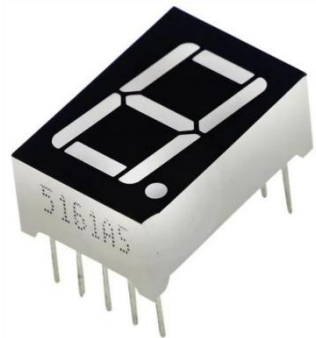
1 x Puce 74HC595

Le 74HC595 est un registre à décalage utilisé pour étendre les ports d'entrée/sortie des circuits numériques en convertissant une entrée série en sortie parallèle, réduisant ainsi le nombre de broches de connexion nécessaires. Cette puce est idéale pour contrôler un grand nombre de dispositifs de sortie, tels qu'un afficheur 7 segments, sans occuper trop de broches du microcontrôleur.



1 x Afficheur 7 segments

Un afficheur 7 segments est un composant en forme de 8 qui contient 7 LEDs. Chaque LED est appelée un segment - lorsqu'un segment est alimenté, il représente une partie du chiffre à afficher.



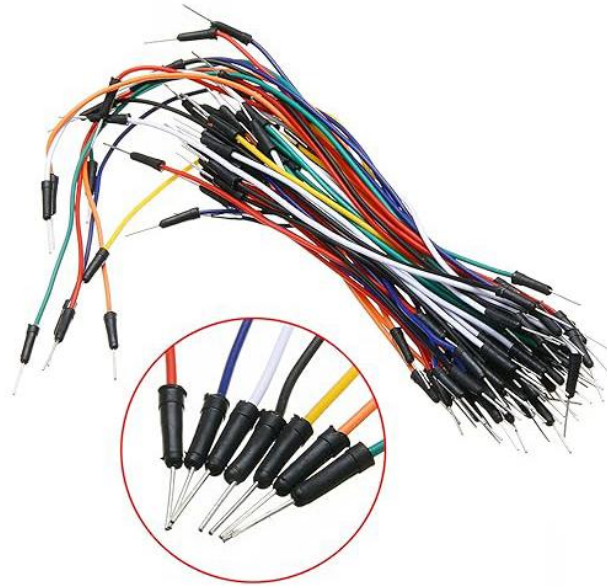
1 x Module ultrasonique

Il s'agit d'un module ultrasonique qui utilise des ondes ultrasonores pour mesurer des distances, permettant de détecter et de mesurer avec précision la position et la distance des objets. Il est largement utilisé dans la robotique, les systèmes d'évitement d'obstacles et les domaines de contrôle automatique.



65 x Fils de connexion

Permettent de connecter les composants entre eux sur la plaque d'essai ainsi qu'à la carte Arduino.



10 x Fils DuPont mâle-femelle

Les fils DuPont mâle-femelle sont spécialement conçus pour connecter des modules avec des broches mâles, comme un module ultrasonique, à une plaque d'essai. Ces fils sont essentiels pour interfacer différents composants dans des projets électroniques.



1 x Câble USB

Permet de connecter la carte Arduino à un ordinateur. Il vous permet d'écrire, compiler et transférer des programmes vers la carte Arduino. Il sert également à alimenter la carte.



1 x Pile 9V

Ceci est une pile alcaline 9V non rechargeable. Vous devez l'installer sur le multimètre.



1 x Multimètre avec fils rouge et noir

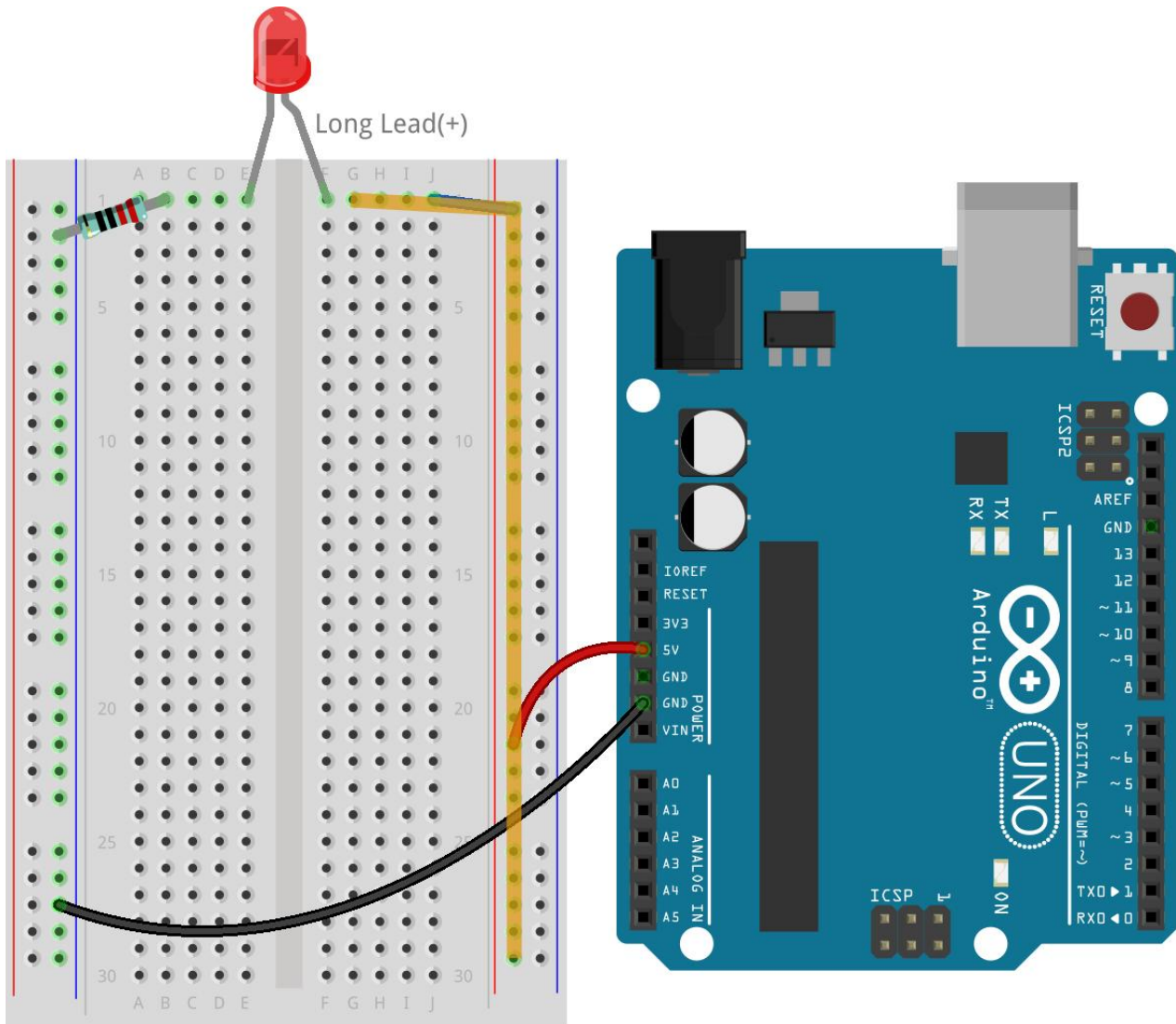
C'est un multimètre polyvalent capable de mesurer la tension, le courant et la résistance, ainsi que d'effectuer d'autres tests électriques, ce qui en fait un outil indispensable pour les travaux d'électronique et d'électricité.



Leçon 2 : Votre premier circuit

Après avoir terminé la leçon, répondez aux questions suivantes

1. Retirez le fil rouge de la plaque d'essai et faites des expériences en le plaçant dans différents trous de la plaque. Observez tout changement au niveau de la LED. Dessinez les positions des trous qui permettent à la LED de s'allumer.



2. Que se passe-t-il si vous inversez les broches de la LED ? S'allume-t-elle ? Pourquoi ou pourquoi pas ?

La LED ne s'allume pas car elle a une conductivité unidirectionnelle ; le courant doit circuler de l'anode vers la cathode pour qu'elle fonctionne.

Leçon 3 : Mesurer avec un multimètre

Répondez à cette question après avoir terminé la section "En savoir plus sur le multimètre !"

1. Maintenant que vous comprenez en détail comment utiliser un multimètre, réfléchissez à quel réglage du multimètre vous utiliseriez pour mesurer les valeurs électriques suivantes ?

Objet de mesure	Réglage du multimètre
9V volts DC	<i>20V</i>
1K ohms	<i>2kΩ</i>
40 milliamps	<i>200mA</i>
110 volts AC	<i>200V~</i>

Remplissez ce tableau pendant la section "Mesurer avec un multimètre"

Type	Unités	Résultats de mesure	Remarques
Tension	Volts	<i>≈ 5.13 volts</i>	
Courant	Milliamps	<i>≈ 13.54 milliamps</i>	
Résistance	Ohms	<i>≈ 378.88 ohms</i>	

Leçon 4 : Loi d'Ohm

Remplissez le tableau suivant pendant la section "**Explorer la loi d'Ohm avec des expériences pratiques**"

1. Remplacez la résistance de 220 ohms par d'autres résistances de valeurs différentes comme indiqué ci-dessous. Notez les changements de luminosité de la LED à chaque substitution pour observer comment la résistance affecte le courant et, par conséquent, l'intensité lumineuse.

Résistance	Observations
100Ω	<i>Plus lumineux</i>
1KΩ	<i>Luminosité normale</i>
10KΩ	<i>Plus faible</i>
1MΩ	<i>Presque éteint</i>

Vous remarquerez que seule la résistance de 100Ω rend la LED plus lumineuse que la résistance précédente de 220Ω. Avec des résistances plus élevées, la luminosité de la LED diminue jusqu'à s'éteindre complètement à 1MΩ. Pourquoi cela se produit-il ?

Selon la loi d'Ohm ($I = V/R$), lorsque la résistance augmente alors que la tension reste constante, le courant traversant la LED diminue, réduisant ainsi la luminosité de la LED. À 1MΩ, le courant est trop faible pour allumer la LED.

2. Après avoir observé les effets du changement de résistance, maintenez la résistance à 220 ohms et changez l'alimentation du circuit de 5V à 3,3V. Notez tous les changements dans la luminosité de la LED.

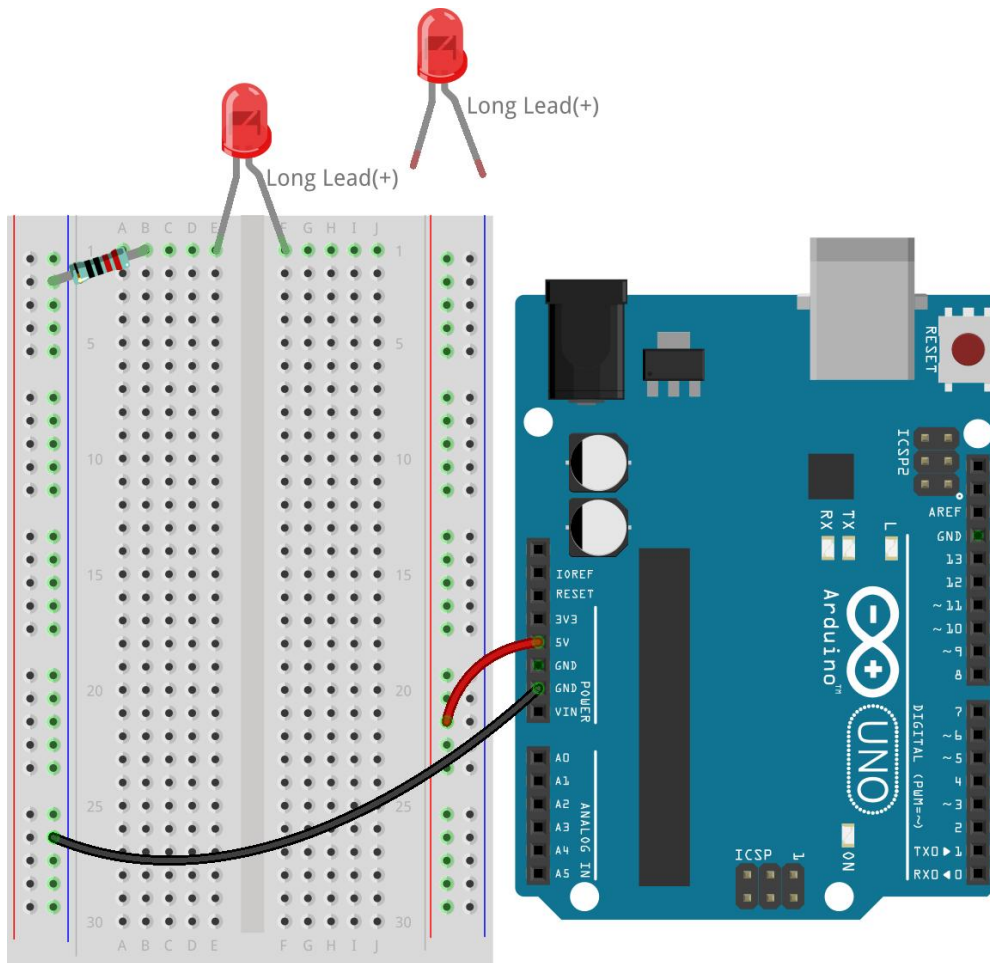
Vous constaterez que la LED est légèrement moins lumineuse à 3,3V qu'à 5V. Pourquoi cela ?

Selon la loi d'Ohm, en connaissant la résistance et la nouvelle tension, le courant devrait être $I = V/R$. Avec une baisse de la tension alors que la résistance reste la même, le courant diminue, ce qui réduit la luminosité de la LED.

Leçon 5 : Circuit en série vs. circuit en parallèle

Répondez aux questions suivantes pendant la section "Plongée dans les circuits en série"

1. Que se passe-t-il si vous retirez une LED ? Pourquoi cela se produit-il ?



Dans un circuit en série, si vous retirez une LED, l'autre LED ne s'allumera pas. En effet, dans un circuit en série, le courant doit traverser tous les composants du circuit. Retirer une LED interrompt le circuit, empêchant ainsi le courant de circuler à travers la LED restante.

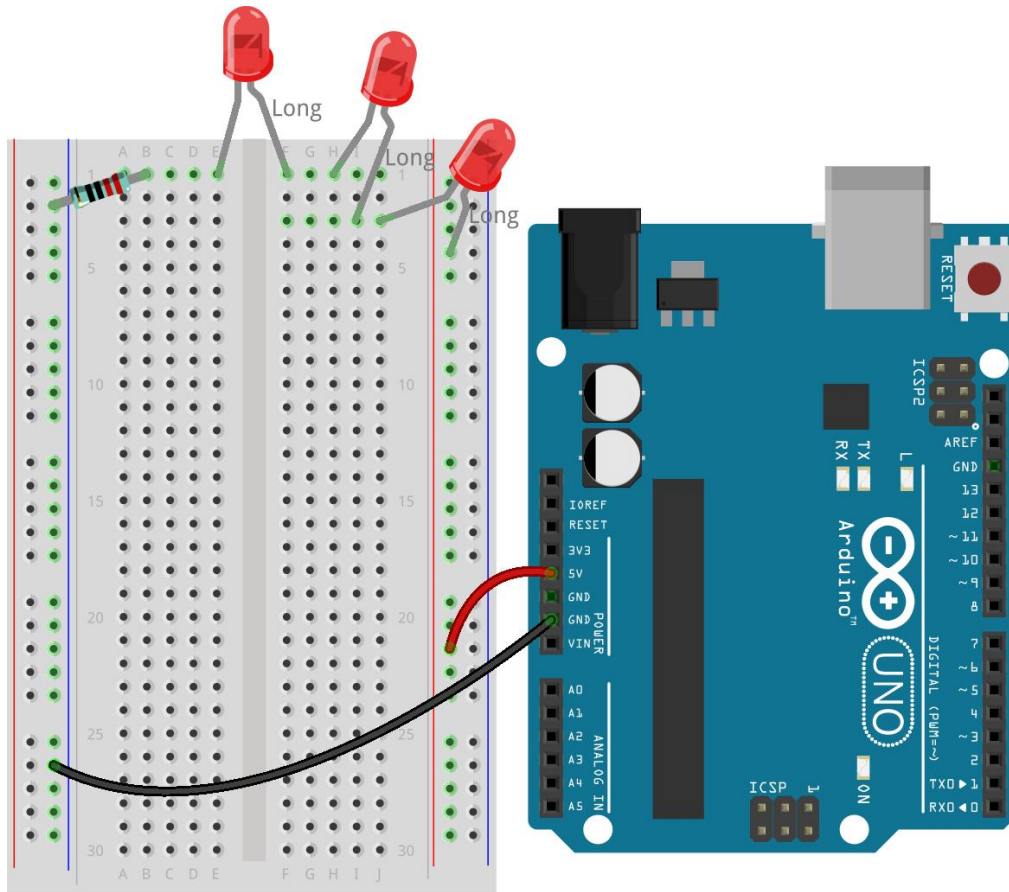
2. Mesurez la tension de chaque composant dans le circuit en série.

Circuit	Tension de la résistance	Tension de la LED1	Tension de la LED2	Tension totale
2 LEDs	$\approx 1.13 \text{ volts}$	$\approx 1.92 \text{ volts}$	$\approx 1.92 \text{ volts}$	$\approx 4.97 \text{ volts}$

3. Mesurez le courant de chaque composant dans le circuit en série.

Circuit	Courant LED1	Courant LED2
2 LEDs	≈ 4.43 milliamps	≈ 4.43 milliamps

4. Si une autre LED est ajoutée à ce circuit, pour un total de trois LEDs, comment la luminosité des LEDs change-t-elle ? Pourquoi ? Comment les tensions aux bornes des trois LEDs changent-elles ?

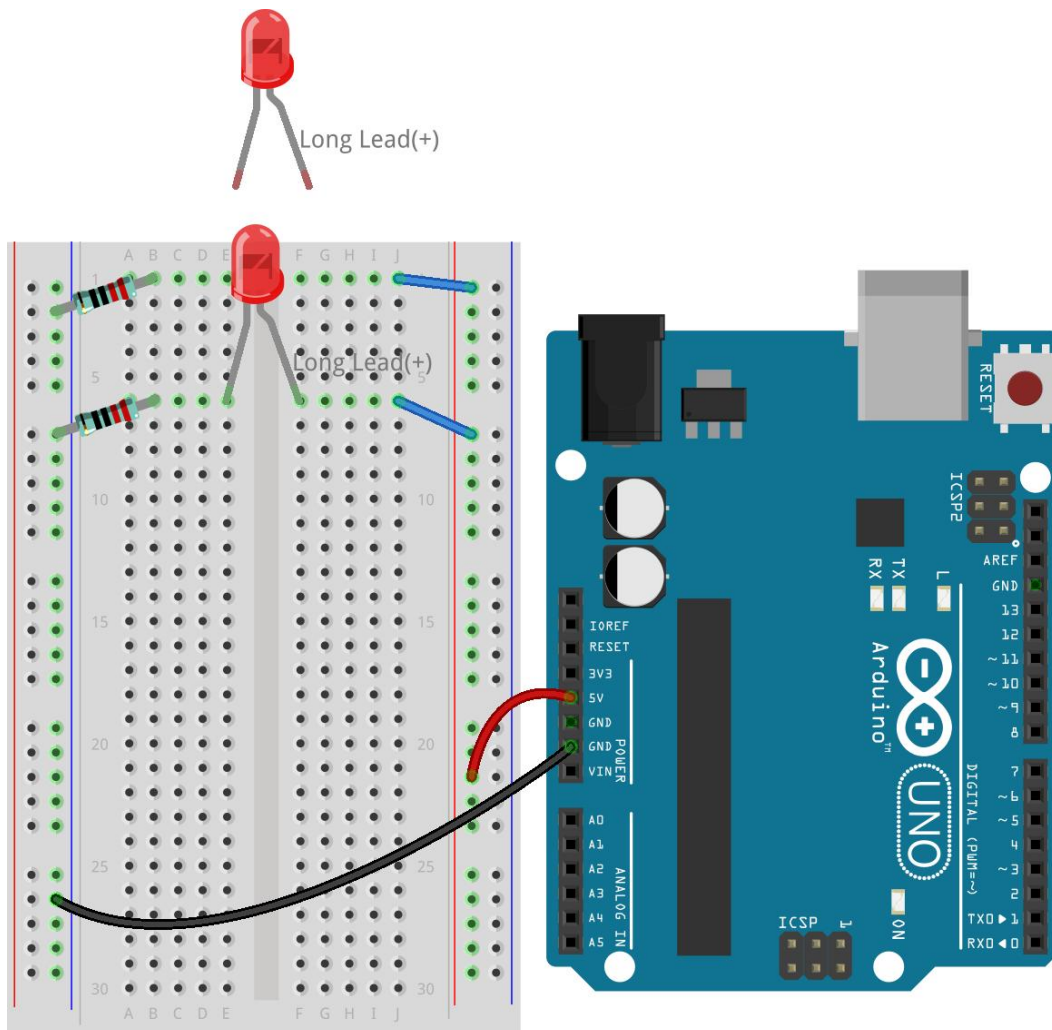


Ajouter une autre LED à un circuit en série comportant déjà deux LEDs entraînera généralement une diminution de la luminosité de chaque LED. Cela se produit parce que la tension totale de la source d'alimentation est divisée entre plus de composants, ce qui se traduit par une chute de tension plus faible à travers chaque LED par rapport à un circuit avec seulement deux. Par conséquent, moins de courant circule à travers chaque LED, réduisant ainsi leur luminosité.

En ce qui concerne les tensions aux bornes des trois LEDs, chaque LED aura désormais une plus petite portion de la tension totale du circuit. Si la tension de la source d'alimentation reste la même, cette tension sera divisée par trois, en supposant que toutes les LEDs ont des caractéristiques électriques similaires. Ainsi, la tension aux bornes de chaque LED dans le circuit sera approximativement d'un tiers de la tension totale fournie par la source d'alimentation.

Répondez aux questions suivantes pendant la section "**Plongée dans les circuits en parallèle**"

1. Dans ce circuit en parallèle, que se passe-t-il si une LED est retirée ? Pourquoi cela se produit-il ?



Dans un circuit en parallèle, si une LED est retirée, les autres LEDs du circuit continueront à s'allumer. Cela se produit parce que chaque LED dans un circuit en parallèle a son propre chemin indépendant vers la source d'alimentation. Retirer une LED n'interrompt pas le flux de courant vers les autres LEDs, elles restent donc intactes et continuent de fonctionner normalement. Cette configuration permet à chaque composant d'un circuit en parallèle de fonctionner indépendamment des autres.

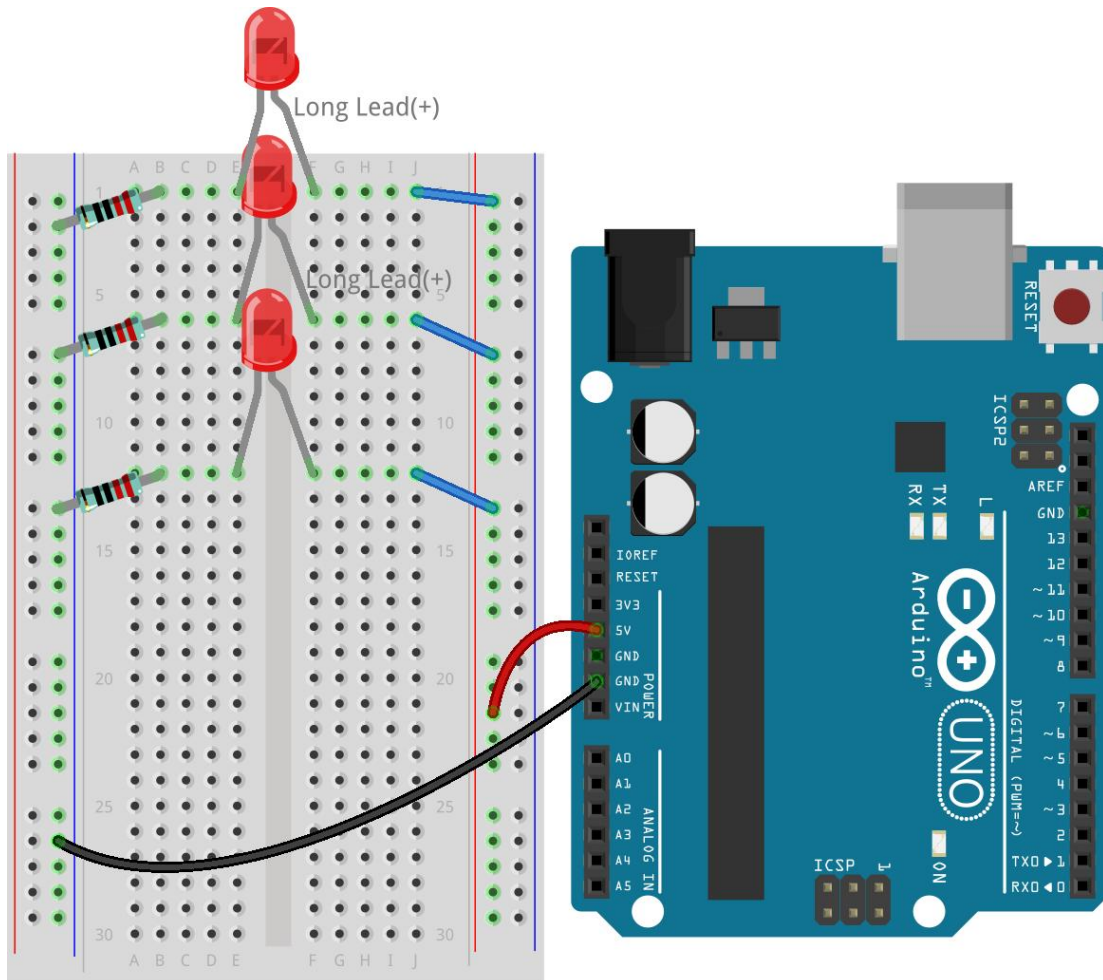
1. Notez la tension mesurée dans le tableau.

Circuit	Tension du chemin 1	Tension du chemin 2
2 LEDs	≈ 5.00 volts	≈ 5.00 volts

2. Notez le courant mesuré dans le tableau.

Circuit	Courant LED1	Courant LED2	Courant total
2 LEDs	≈ 12.6 milliamps	≈ 12.6 milliamps	≈ 25.3 milliamps

3. Si une autre LED est ajoutée à ce circuit, que se passe-t-il avec la luminosité des LEDs ? Pourquoi ? Notez votre réponse dans votre manuel.



Lorsque vous ajoutez une autre LED à un circuit en parallèle, la luminosité des LEDs existantes reste généralement inchangée. En effet, chaque LED dans un circuit en parallèle possède son propre chemin direct vers la source d'alimentation, donc la tension aux bornes de chaque LED reste constante, peu importe le nombre de LEDs ajoutées. Chaque LED reçoit la pleine tension nécessaire pour fonctionner à sa luminosité optimale. Ainsi, l'ajout de LEDs supplémentaires n'affecte pas la luminosité des LEDs déjà présentes, à condition que l'alimentation électrique puisse supporter la demande totale de courant du circuit. Assurez-vous de noter cela dans votre manuel pour référence future.

Leçon 6 : Faire clignoter une LED

Remplissez le tableau suivant pendant la section "Donner vie aux LEDs"

1. Notez la tension mesurée dans le tableau pour la broche 3.

État	Tension à la broche 3
HIGH	<i>≈4.95 volts</i>
LOW	<i>0.00 volts</i>

Après avoir terminé la leçon, répondez à la question suivante

1. Téléchargez le code ci-dessus, et vous verrez que la LED clignote de manière répétée à un intervalle de 3 secondes. Si vous souhaitez qu'elle ne s'allume et ne s'éteigne qu'une seule fois, que devriez-vous faire ?

Vous pouvez déplacer les commandes qui allument et éteignent la LED de la fonction `loop()` vers la fonction `setup()`. La fonction `setup()` ne s'exécute qu'une seule fois lorsque le programme démarre, ce qui fera en sorte que la LED s'allume et s'éteigne une seule fois. Voici comment vous pouvez ajuster votre code :

```
void setup() {  
  // Setup code here, to run once:  
  pinMode(3, OUTPUT); // set pin 3 as output  
  
  digitalWrite(3, HIGH); // Light up the LED on pin 3  
  delay(3000);           // Wait for 3 seconds  
  digitalWrite(3, LOW);  // Switch off the LED on pin 3  
}  
  
void loop() {  
  // Main code here, to run repeatedly:  
}
```

Leçon 7 : Fabriquons des feux de circulation !

Répondez à la question suivante pendant la section "Écrire le pseudo-code pour un feu de circulation"

Réfléchissez à ce qui doit se passer pour que votre circuit fonctionne comme un feu de circulation. Dans l'espace fourni dans votre journal de bord, écrivez le pseudo-code décrivant le fonctionnement de votre feu de circulation. Utilisez un langage simple.

Pour simuler un feu de circulation avec un Arduino, vous aurez besoin d'un montage avec trois LEDs (rouge, jaune et verte) et d'une séquence qui contrôle l'allumage de manière à imiter les feux de circulation réels. Voici un simple schéma de pseudo-code que vous pouvez noter dans votre journal pour décrire le fonctionnement de ce circuit de feu de circulation :

Setup:

```
Define pins for the red, yellow, and green LEDs.  
Set all these pins as outputs.
```

Main Loop:

```
Turn on the red LED for 5 seconds.  
Turn off the red LED.  
Turn on the yellow LED for 2 seconds.  
Turn off the yellow LED.  
Turn on the green LED for 5 seconds.  
Turn off the green LED.  
Repeat the cycle.
```

Répondez à la question suivante pendant la section "Écrire le pseudo-code pour un feu de circulation"

Regardez les intersections autour de chez vous. Combien de feux de circulation y a-t-il généralement ? Comment se coordonnent-ils entre eux ?

Dans les zones urbaines, les intersections sont souvent équipées de feux de circulation pour gérer efficacement le flux des véhicules et des piétons. Le nombre de feux à une intersection peut varier considérablement en fonction de sa taille et de sa complexité. Une intersection simple à quatre voies comporte généralement au moins quatre feux de circulation, un pour chaque direction. Les intersections plus complexes peuvent avoir des feux supplémentaires pour les voies de virage, les passages piétons et d'autres besoins de gestion du trafic.

Leçon 8 : Feu de circulation avec bouton piéton

Après avoir terminé "Construire le circuit", répondez à la question suivante

1. Votre feu de circulation est un mélange de circuits en série et parallèles. Discutez des parties de votre circuit qui sont en série et pourquoi. Ensuite, expliquez quelles parties sont en parallèle et pourquoi.

Dans ce circuit, le bouton et sa résistance de tirage de 10K sont connectés en série. Cette configuration garantit que lorsque le bouton est pressé, il modifie correctement l'état de la broche 8 en le connectant directement à la masse lorsqu'il n'est pas pressé, évitant ainsi les entrées flottantes.

Les trois LED connectées aux broches 3, 4 et 5 sont en parallèle les unes avec les autres. Chaque LED fonctionne de manière indépendante car elles sont connectées à des broches de contrôle distinctes et partagent une alimentation commune. Cette configuration permet à chaque LED de fonctionner sans affecter les autres, ce qui est essentiel pour un système de feux de circulation.

Remplissez ce tableau pendant la phase de "Création de Code"

1. Remplissez le tableau avec la tension mesurée sur la broche 8 lorsque le bouton est pressé et lorsqu'il n'est pas pressé. Ensuite, indiquez les états correspondants de niveau haut et bas.

État du bouton	Tension sur la broche 8	État de la broche 8
Release	0.00 volts	LOW
Press	≈4.97 volts	HIGH

Répondez à la question suivante à la fin de cette leçon.

1. Pendant les tests, vous remarquerez peut-être que la LED verte ne clignote que lorsque le bouton piéton est maintenu enfoncé, mais les piétons ne peuvent pas traverser la route en maintenant le bouton enfoncé en continu. Comment pouvez-vous modifier le code pour garantir qu'une fois le bouton piéton pressé, la LED verte s'allume suffisamment longtemps pour permettre une traversée en toute sécurité sans nécessiter un maintien

continu ? Veuillez inscrire la solution en pseudo-code dans votre carnet.

Pour garantir que la LED verte s'allume pour les piétons sans nécessiter un maintien continu du bouton, et pour continuer le cycle normal des feux de circulation par la suite, vous pouvez ajuster votre pseudo-code afin de vérifier la pression du bouton, puis de modifier l'état de fonctionnement en fonction de cette pression. Voici une version optimisée et plus claire du pseudo-code qui reflète ces changements :

Setup:

- Define pins for red, yellow, and green LEDs as output
- Define the button pin as input

Main Loop:

- Check if the button is pressed

- If button is pressed:

 - Turn off all LEDs

 - Turn on green LED for pedestrians

 - Delay 10 seconds

- Else:

 - Execute normal traffic light cycle:

 - Turn on green LED (for vehicles), turn off other LEDs

 - Delay 10 seconds

 - Turn on yellow LED, turn off other LEDs

 - Delay 3 seconds

 - Turn on red LED, turn off other LEDs

 - Delay 10 seconds

Leçon 9 : Lampe de bureau dimmable

Remplissez ce tableau lors de "Construire le circuit"

1. Faites tourner le potentiomètre dans le sens des aiguilles d'une montre de la position 1 à 3 et mesurez la résistance à chaque point, puis enregistrez les résultats dans le tableau.

Point de mesure	Résistance (kilohms)
1	1.52
2	5.48
3	9.01

2. Comment pensez-vous que la tension au niveau de A0 changera lorsque le potentiomètre est tourné dans le sens des aiguilles d'une montre et dans le sens inverse ?

On peut considérer le potentiomètre comme étant composé de deux résistances connectées en série dans le circuit. Selon la mesure des résistances, la résistance entre A0 et GND augmente lorsque le potentiomètre est tourné dans le sens des aiguilles d'une montre. Comme le courant reste constant dans un circuit en série, selon la loi d'Ohm ($\text{tension} = \text{courant} \times \text{résistance}$), une augmentation de la résistance entraîne une augmentation de la tension au niveau de A0. Par conséquent, tourner le potentiomètre dans le sens des aiguilles d'une montre augmente la tension à A0, tandis que le tourner dans le sens inverse la diminue, car la résistance diminue.

Complétez la question suivante à la fin de cette leçon.

1. Si vous connectez la LED à une autre broche, comme la broche 8, et que vous faites tourner le potentiomètre, la luminosité de la LED va-t-elle toujours changer ? Pourquoi ou pourquoi pas ?

Si vous connectez la LED à la broche 8 d'un Arduino UNO et que vous faites tourner le potentiomètre, la luminosité de la LED ne changera pas. Cela est dû au fait que la broche 8 ne prend pas en charge le PWM (Modulation de Largeur d'Impulsion), qui est nécessaire pour ajuster les niveaux de luminosité en utilisant la fonction `analogWrite()`. Sur un Arduino UNO, les broches qui prennent en charge le PWM et qui peuvent donc être utilisées pour contrôler la luminosité d'une LED via `analogWrite()` sont les broches 3, 5, 6, 9, 10 et 11.

Leçon 10 : Lampe de bureau ON/OFF

Répondez aux questions suivantes à la fin de cette leçon.

1. What would happen if you set digital pin 7 to INPUT only? Why?

```
void setup() {  
  pinMode(9, OUTPUT); // Set pin 9 as output  
  pinMode(7, INPUT);  // Set pin 7 as input with an internal pull-up resistor  
  Serial.begin(9600); // Serial communication setup at 9600 baud  
}
```

Configurer la broche numérique 7 en mode INPUT dans votre sketch Arduino, plutôt qu'en mode INPUT_PULLUP, peut entraîner une instabilité potentielle du signal lu depuis la broche. Lorsqu'une broche est configurée uniquement en tant qu'INPUT et qu'elle n'est connectée ni à une tension haute ni à une tension basse à travers un circuit externe, elle devient ce que l'on appelle "flottante". Une broche flottante n'est pas dans un état haut ou bas stable ; son état peut fluctuer en fonction du bruit électrique ou des interférences de l'environnement. Cette fluctuation peut entraîner des lectures imprévisibles lorsque vous essayez de lire l'état de la broche via des fonctions d'entrée numérique, ce qui peut donner des données erronées ou incohérentes au microcontrôleur.

2. Si la broche 7 est configurée uniquement en tant qu'INPUT, quelles modifications doivent être apportées au circuit ?

Si la broche 7 de votre Arduino est configurée en mode INPUT et que vous souhaitez garantir des lectures stables et prévisibles, vous devez ajouter une résistance de tirage externe au circuit. Cela consiste à connecter une résistance de 10 k Ω entre la broche 7 et l'alimentation 5V de l'Arduino. La résistance de tirage garantit que la broche d'entrée est à un état élevé (niveau logique 1) lorsqu'aucun autre signal d'entrée n'est présent.

Leçon 11 : Contrôle des matrices LED avec un potentiomètre

Répondez aux questions suivantes avant de passer à la "Création de code".

1. Écrivez votre pseudocode pour le tableau LED.

Le pseudocode sert de croquis pour le programme, rédigé en langage simple pour faciliter la compréhension. Votre tâche consiste à créer un pseudocode pour un tableau LED qui réagit à un potentiomètre. À mesure que la valeur du potentiomètre augmente, davantage de LED s'allument.

Voici un pseudocode simplifié qui décrit le contrôle d'un tableau LED en fonction de l'entrée d'un potentiomètre :

```
Declare readValue variable.  
Setup.  
Declare 3 digital pin outputs.  
Main Loop.  
If the potentiometer's value is below 200, all LEDs should be off.  
If the value is between 200 and 600, the first LED should be on.  
If the value is between 600 and 1000, the first two LEDs should be on.  
If the value exceeds 1000, all LEDs should be on.  
Delay for a short time.
```

Après avoir terminé la leçon, répondez à la question suivante.

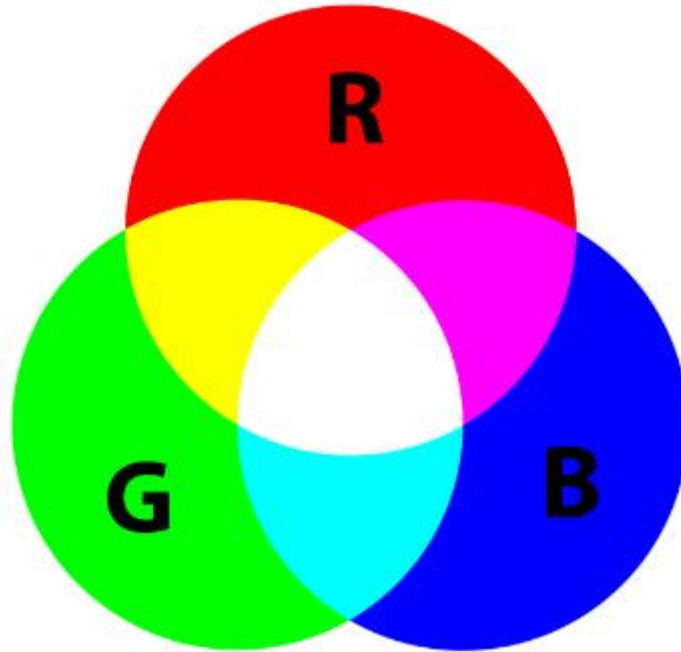
1. Dans le dernier code, nous déterminons le nombre de LED à allumer en fonction de la valeur du potentiomètre. Comment pouvons-nous modifier le code afin que, tout en allumant les LED, leur luminosité change en fonction du potentiomètre ?

Pour modifier votre code afin que la luminosité des LED change en fonction de la valeur du potentiomètre, vous pouvez utiliser la fonction `analogWrite()` au lieu de `digitalWrite()`. La fonction `analogWrite()` vous permet de contrôler la luminosité des LED par le biais du PWM (Modulation de Largeur d'Impulsion).

Leçon 12 : Les Couleurs de l'Arc-en-Ciel

Remplissez ce tableau pendant la "Création du Code"

1. Si vous souhaitez d'autres couleurs, que devez-vous faire ? Reportez-vous au diagramme ci-dessous et notez vos idées dans votre carnet.



Couleur	Broche Rouge	Broche Verte	Broche Bleue
Rouge	<i>ÉLEVÉ</i>	<i>BAS</i>	<i>BAS</i>
Vert	<i>BAS</i>	<i>ÉLEVÉ</i>	<i>BAS</i>
Bleu	<i>BAS</i>	<i>BAS</i>	<i>ÉLEVÉ</i>
Jaune	<i>ÉLEVÉ</i>	<i>ÉLEVÉ</i>	<i>BAS</i>
Rose	<i>ÉLEVÉ</i>	<i>BAS</i>	<i>ÉLEVÉ</i>
Cyan	<i>BAS</i>	<i>ÉLEVÉ</i>	<i>ÉLEVÉ</i>
Blanc	<i>ÉLEVÉ</i>	<i>ÉLEVÉ</i>	<i>ÉLEVÉ</i>

Leçon 13 : Le Spectre de la Vue

Remplissez ce tableau pendant la "Création de Code"

1. Maintenant, vous pouvez ajuster les valeurs des broches 9, 10 et 11 séparément, et noter les couleurs observées dans votre carnet.

Broche Rouge	Broche Verte	Broche Bleue	Couleur
0	128	128	<i>Bleu foncé</i>
128	0	255	<i>Violet</i>
128	128	255	<i>Bleu clair</i>
255	128	0	<i>Orange</i>

2. Choisissez quelques-unes de vos couleurs préférées et remplissez le tableau avec leurs valeurs RGB.

Couleur	Rouge	Vert	Bleu

Leçon 14 : Couleurs aléatoires

Après avoir terminé la leçon, répondez aux questions suivantes.

1. Si vous modifiez le code de `randomSeed(analogRead(A0))` à `randomSeed(0)`, comment les couleurs de la LED RGB vont-elles changer et pourquoi ?

2.

Si vous changez le code de `randomSeed(analogRead(A0))` à `randomSeed(0)`, cela affectera l'aléatoire des couleurs de la LED RGB. La fonction `randomSeed(seed)` est utilisée pour initialiser le générateur de nombres pseudo-aléatoires dans Arduino, ce qui influence la séquence des nombres aléatoires générés par des fonctions telles que `random()`.

Utiliser `analogRead(A0)` comme valeur de départ lit une valeur légèrement aléatoire depuis la broche analogique A0, généralement influencée par le bruit ambiant et d'autres facteurs, ce qui entraîne des valeurs de départ différentes à chaque démarrage du programme. Cela signifie que la séquence des nombres aléatoires (et donc les couleurs) variera à chaque réinitialisation de l'Arduino.

En revanche, en définissant la valeur de départ avec une valeur fixe comme `randomSeed(0)`, le générateur de nombres aléatoires est initialisé avec le même point de départ à chaque exécution du programme. Cela entraîne la même séquence de nombres aléatoires, ce qui signifie que la LED RGB affichera le même motif de couleurs à chaque réinitialisation ou mise sous tension de l'Arduino. Cela élimine le caractère aléatoire des changements de couleur de la LED.

1. Quelles sont certaines situations où le hasard est utilisé pour résoudre des problèmes dans la vie quotidienne, en dehors du choix aléatoire des couleurs pour la décoration et des numéros de loterie ?

Le hasard est utilisé dans divers contextes quotidiens pour résoudre des problèmes, notamment :

- **Jeux de société:** Lancer des dés pour déterminer les déplacements, garantissant que chaque partie soit différente et équitable.

- **Playlists musicales:** Mélanger les chansons pour garder votre expérience d'écoute fraîche et imprévisible.
- **Choix alimentaires:** Choisir un restaurant ou un plat au hasard peut faciliter la prise de décision et rendre le moment plus amusant lorsque vous ne savez pas quoi manger.
- **Arrangements de sièges:** Tirer au sort les places lors d'événements pour mélanger les invités et encourager les interactions sociales.
- **Soirées cinéma:** Utiliser un tirage au sort pour sélectionner un film lorsque chacun a des préférences différentes.

Leçon 15 : Couleurs froides ou chaudes

Remplissez ce tableau pendant la « Création de code »

1. Ouvrez Paint ou tout autre outil de sélection de couleurs, trouvez ce que vous considérez comme les couleurs les plus chaudes et les plus froides, puis notez leurs valeurs RGB dans votre manuel.

Type de Couleur	Rouge	Vert	Bleu
Couleur Chaude	246	52	8
Couleur Froide	100	150	255

Après avoir terminé la leçon, répondez à la question suivante.

1. Remarque que les « limites inférieures » d'un intervalle peuvent être supérieures ou inférieures aux « limites supérieures », donc la fonction map() peut être utilisée pour inverser une plage de nombres, par exemple :

```
y = map(x, 1, 50, 50, 1);
```

La fonction gère également bien les nombres négatifs, de sorte que cet exemple est valide et fonctionne correctement.

```
y = map(x, 1, 50, 50, -100);
```

Pour $y = \text{map}(x, 1, 50, 50, -100)$, si x est égal à 20, quelle devrait être la valeur de y ? Référez-vous à la formule suivante pour le calculer.

$$\frac{\text{Value} - \text{From Low}}{\text{From High} - \text{From Low}} = \frac{\text{Y} - \text{To Low}}{\text{To High} - \text{To Low}}$$

$$\text{Y} = \frac{\text{Value} - \text{From Low}}{\text{From High} - \text{From Low}} \times (\text{To High} - \text{To Low}) + \text{To Low}$$

Pour $x=20$ en utilisant la formule de mapping $y = \text{map}(x, 1, 50, 50, -100)$; la valeur de y serait d'environ -8.16.

Leçon 16 : Alarme de Température

Remplissez ce tableau lors de la "Construction du Circuit"

1. Lisez la valeur de la résistance à différentes températures et enregistrez-la dans le tableau ci-dessous.

Environment	Resistance (kiloohm)
Température actuelle	9.37
Température élevée	6.10
Température basse	12.49

Après avoir terminé la leçon, répondez à la question suivante.

1. Dans le code, les températures en Kelvin et en Celsius sont calculées. Si vous souhaitez également connaître la température en Fahrenheit, que devez-vous faire ?

C'est la méthode standard pour convertir les Celsius en Fahrenheit, et elle vous donnera la température en Fahrenheit en fonction de la valeur Celsius que vous avez déjà obtenue lors de vos calculs.

$$F = C * 1.8 + 32$$

1. Pouvez-vous envisager d'autres situations ou lieux où un système de surveillance de la température, comme celui que nous avons construit aujourd'hui, pourrait être utile ?

Les systèmes de surveillance de la température sont largement applicables dans des situations quotidiennes et divers environnements. Voici quelques exemples simplifiés :

- **Confort domestique:** Ajustez automatiquement le chauffage ou la climatisation de votre maison en fonction des relevés de température en temps réel pour maintenir un espace de vie confortable.
- **Jardinage:** Surveillez les températures dans les serres pour garantir que les plantes se développent dans des conditions optimales. Ajoutez des systèmes automatisés pour ajuster les températures selon les besoins.
- **Sécurité alimentaire:** Suivez les températures des réfrigérateurs et des congélateurs pour garantir que les aliments restent sûrs à consommer, notamment dans les restaurants ou lors du transport des aliments.
- **Santé:** Surveillez et enregistrez les températures dans les zones de stockage pour les médicaments et vaccins sensibles à la température afin de garantir leur efficacité.

Leçon 17 : Code Morse

Répondez à la question suivante lors de la « **Construction du circuit** »

1. Que se passera-t-il si vous connectez la cathode d'un buzzer actif directement à la masse (GND) et l'anode à 5V ? Pourquoi ?

Si vous connectez la cathode d'un buzzer actif directement à la masse (GND) et l'anode à 5V, le buzzer émettra un son continu. Cela se produit parce que l'oscillateur interne du buzzer est activé par l'alimentation de 5V, ce qui le fait générer un son jusqu'à ce que le circuit soit déconnecté.

Après avoir terminé la leçon, répondez à la question suivante.

1. En utilisant le tableau du code Morse fourni, écrivez un code pour envoyer le message « Bonjour ».

En code Morse, « Bonjour » serait codé comme suit en fonction des caractères :

- H:
- E: .
- L: .-..
- L: .-..
- O: ---

En rassemblant le tout, le code Morse pour « Bonjour » est :

.... .-.. .-.. ---

Dans la communication pratique, il y a généralement une pause plus longue entre les mots pour les différencier clairement, mais comme « Bonjour » est un mot unique, le code est continu avec des espaces séparant uniquement les lettres individuelles.

Leçon 18 : Alarme lumineuse

Remplissez ce tableau lors de la construction du circuit.

1. Lisez la valeur de la résistance sous la lumière ambiante actuelle et enregistrez-la dans le tableau ci-dessous.

Environment	Resistance (kilohm)
Lumière normale	≈5.48
Lumière vive	≈0.16
Obscurité	≈1954

Après avoir terminé la leçon, répondez à la question suivante.

1.Des voleurs rusés pourraient choisir de voler la nuit, et si un tableau disparaît, le photoresistor pourrait ne pas détecter de changement de luminosité, empêchant ainsi le déclenchement de l'alarme. Que peut-on faire pour remédier à ce problème ?

Une solution consiste à installer une source lumineuse devant le tableau. Cela garantit non seulement que le tableau est clairement visible et éclairé, mais cela signifie également que toute perturbation ou retrait du tableau déclenchera immédiatement l'alarme du photoresistor en modifiant le niveau de luminosité détecté.

Leçon 19 : Système d'alarme de stationnement inversé

Après avoir terminé la leçon, répondez à la question suivante.

1. Si vous souhaitez que la distance détectée par cet appareil soit plus précise au niveau des décimales, comment devez-vous modifier le code ?

Pour rendre la mesure de distance plus précise au niveau des décimales, vous pouvez modifier le calcul dans la fonction `measureDistance` pour utiliser l'arithmétique à virgule flottante au lieu de l'arithmétique entière. Cela permettra d'inclure des valeurs décimales dans le résultat, offrant ainsi des mesures plus précises.

```
float distance = duration * 0.034 / 2.0;
```

La variable de `distance` dans la fonction `measureDistance` est modifiée en float pour prendre en charge les valeurs décimales.

Le calcul de la `distance` utilise l'arithmétique à virgule flottante en divisant par `2.0` au lieu de `2`.

Leçon 20 : Le minuteur Pomodoro

Répondez à la question suivante lors de "Création de code - millis()"

1. Que se passera-t-il si `delay(100)` est remplacé par `delay(1000)` dans le programme ? Pourquoi ?

```
if (currentMillis - previousMillis >= interval) {  
  previousMillis = currentMillis; // Save the last time the buzzer beeped  
  digitalWrite(buzzerPin, HIGH); // Make a voice  
  delay(100);  
  digitalWrite(buzzerPin, LOW); // silence  
}
```

Dans le code original, le buzzer émet un bip pendant environ 100 millisecondes toutes les 1000 millisecondes (1 seconde, comme défini par la variable d'intervalle), suivi d'un silence de 900 millisecondes. Après la modification, le buzzer bippera pendant 1000 millisecondes toutes les 1000 millisecondes, et émettra presque immédiatement un nouveau bip, car le prochain `interval` commence presque instantanément. Ainsi, changer le délai de 100 à 1000 millisecondes transforme le buzzer d'un bip bref à un son continu, ce qui devient plus agaçant et inapproprié par rapport à l'intention originale.

Changer `delay(100)` en `delay(1000)` dans votre code fera en sorte que le buzzer sonne pendant une seconde entière au lieu d'un bref bip, car cela augmente le temps de pause lorsque le buzzer est activé. Cela entraîne des sons de buzzer plus longs et des cycles de programme moins fréquents, rendant potentiellement le programme moins réactif aux autres tâches durant ces intervalles.

Après avoir terminé la leçon, répondez à la question suivante.

Réfléchissez à d'autres endroits de votre vie où vous pouvez « entendre » le temps. Dressez une liste de quelques exemples et inscrivez-les dans votre carnet !

Entendre le temps est un concept intrigant, et il existe plusieurs scénarios quotidiens où nous pouvons faire cette expérience. Voici quelques exemples que vous pourriez noter :

- **Horloges et montres :** Le tic-tac des horloges analogiques ou les bips spécifiques des montres numériques signalant chaque seconde ou minute écoulée.
- **Minuteries de cuisine :** Le tic-tac et l'alarme finale d'une minuterie mécanique ou

numérique qui compte le temps de cuisson ou de pâtisserie.

- **Cloches d'école** : Le son des cloches dans les écoles qui marquent le début et la fin des cours ou des pauses.
- **Annonces dans les transports publics** : Les bips ou carillons qui précèdent les annonces dans les gares ou dans les bus, signalant le départ ou l'arrivée imminente.
- **Micro-ondes** : Le bruit de bip lorsque le minuteur se termine, signalant que le processus de chauffage est terminé.
- **Montres de fitness ou traqueurs sportifs** : Les bips ou alarmes indiquant la fin d'un temps défini pendant les entraînements ou les intervalles.

Leçon 21 : Son de sirène

Répondez à la question suivante pendant "**Construire le circuit**"

1. Que se passera-t-il si vous connectez la cathode d'un buzzer passif directement à la masse (GND) et l'anode à 5V ? Pourquoi ?

Si vous connectez la cathode d'un buzzer passif directement à la masse (GND) et l'anode à 5V, contrairement à un buzzer actif, le buzzer passif ne produira aucun son par lui-même car il n'a pas d'oscillateur intégré.

Un buzzer passif nécessite un signal externe pour générer du son. En général, vous devez le piloter avec une onde carrée (tension oscillante) à la fréquence souhaitée pour créer des sons audibles.

Répondez aux questions suivantes pendant "**Création de code - Faire sonner le buzzer passif**".

1. Si vous changez le code et les broches du circuit pour utiliser les broches 7 ou 8, qui ne sont pas des broches PWM, le buzzer émettra-t-il toujours un son ? Vous pouvez tester, puis noter votre réponse dans le manuel.

Même si la broche 8 n'est pas une broche PWM, la fonction `tone()` peut toujours générer une onde carrée précise sur celle-ci, permettant ainsi de piloter efficacement un buzzer passif pour produire du son. Cette flexibilité vous permet d'utiliser n'importe quelle broche numérique pour la sortie sonore sans être limité aux broches compatibles PWM. Lorsque vous appelez la fonction `tone(pin, fréquence)`, Arduino configure un minuteur pour changer l'état de la broche (de HAUT à BAS et retour à HAUT) à la fréquence spécifiée, créant ainsi une onde carrée. Cette onde carrée active le buzzer passif, le faisant émettre du son à la fréquence de l'onde générée.

Pour explorer comment la fréquence et la durée dans tone(pin, fréquence, durée) affectent le son du buzzer, veuillez modifier le code sous deux conditions et noter les phénomènes observés dans votre manuel :

- En maintenant la fréquence à 1000, augmentez progressivement la durée, de 100 à 500, puis à 1000. Comment le son du buzzer change-t-il et pourquoi ?
 - Durée de 100 ms : Le son est un bref bip.
 - Durée de 500 ms : Le son est un bip plus long, clairement audible et qui dure une demi-seconde.
 - Durée de 1000 ms: Le son est encore plus long, il dure une seconde entière.

En augmentant la durée, le son émis par le buzzer dure plus longtemps. La hauteur ou la fréquence du son reste constante (car elle est réglée à 1000 Hz), ce qui signifie que la "note" du ton ne change pas, mais la durée pendant laquelle vous l'entendez augmente. Cela est utile pour signaler différentes durées d'alertes où l'urgence ou le type d'alerte peut être distingué par la longueur du ton.

- En maintenant la durée à 100, augmentez progressivement la fréquence, de 1000 à 2000, puis à 5000. Comment le son du buzzer change-t-il et pourquoi ? Avec l'augmentation de la fréquence, le son devient plus aigu. La fréquence du son, qui mesure le nombre de vibrations par seconde, influe directement sur la hauteur du ton entendu. Ainsi, plus la fréquence est élevée, plus le son perçu est aigu, ce qui modifie la perception de l'alerte émise par le buzzer.
 - Fréquence de 1000 Hz: Le son émis est un bip de tonalité moyenne.
 - Fréquence de 2000 Hz: Le son présente une tonalité plus haute comparée à celle de 1000 Hz.
 - Fréquence de 5000 Hz: Le son est nettement plus aigu, perçu comme plus perçant et potentiellement inconfortable à courte distance.

Augmenter la fréquence tout en maintenant la durée constante entraîne un changement de tonalité du son. Les fréquences plus élevées produisent des sons plus aigus. Ce principe est utile pour distinguer différents types de notifications ou de signaux en fonction de leur urgence ou de leur importance, les tonalités plus élevées étant souvent utilisées pour les alertes plus urgentes.

Leçon 22 : Jouer « Ah ! vous dirai-je, maman »

Répondez à la question suivante lors de la « **Création de code - Tableau** »

1. Vous pouvez également effectuer des opérations sur les éléments du tableau, comme changer pour `Serial.println(melody[i] * 1.3),``. Quelles données obtiendrez-vous et pourquoi ?

Le nombre 1.3 est un nombre à virgule flottante. Lorsqu'un entier du tableau `melody` (qui est de type `int`) est multiplié par 1.3, le résultat de l'opération est automatiquement converti en un nombre à virgule flottante (`float`).

Pour chaque fréquence de note dans ce tableau, la multiplier par 1,3 puis imprimer le résultat donnera :

```
340.6  
340.6  
509.6  
509.6  
572.0  
572.0  
509.6  
...
```

Après avoir terminé la leçon, répondez à la question suivante :

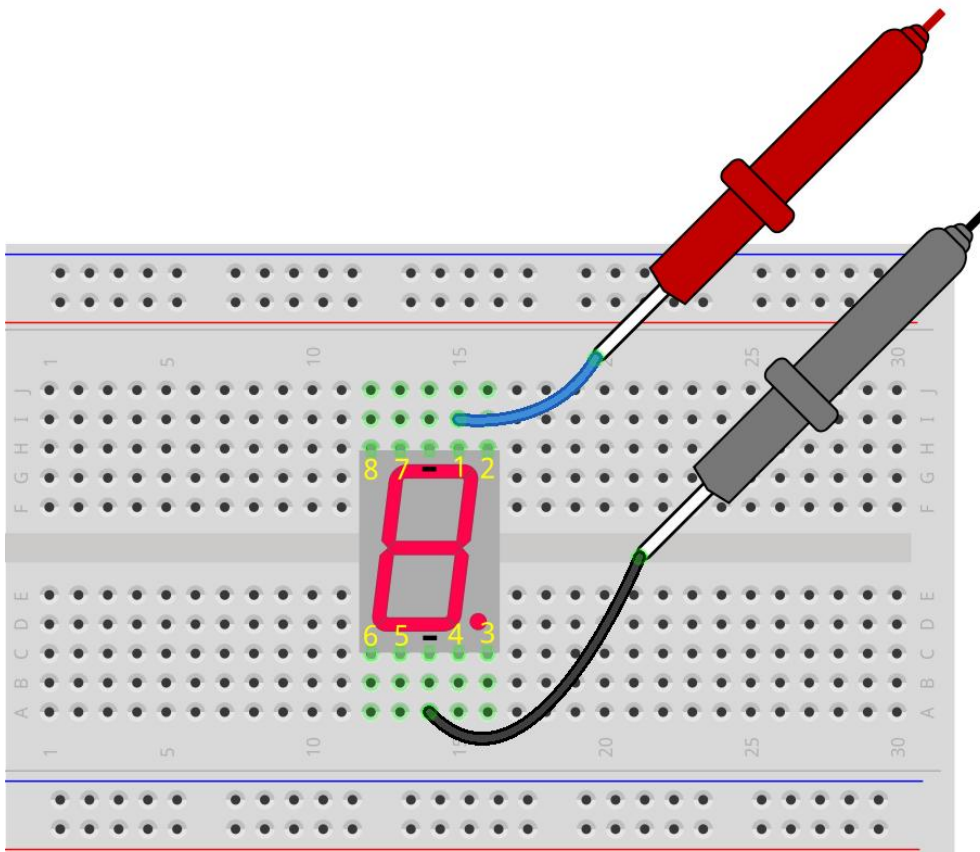
1. Si vous remplacez le buzzer passif dans le circuit par un buzzer actif, pouvez-vous jouer positivement « Twinkle Twinkle Little Star » ? Pourquoi ?

Si vous remplacez le buzzer passif par un buzzer actif pour jouer « Twinkle Twinkle Little Star », cela ne fonctionnera pas comme prévu. Les buzzers actifs ne peuvent produire qu'un seul ton, car ils possèdent un oscillateur intégré. Par conséquent, vous ne pouvez pas contrôler la hauteur pour jouer la mélodie avec précision ; vous entendriez seulement un bip répétitif au rythme de la chanson, et non les notes réelles.

Leçon 23 : Dé cybernétique

Répondez aux questions suivantes pendant « Comprendre l'affichage à 7 segments »

1. Si un segment s'allume, reportez-vous à ce diagramme pour enregistrer le numéro de broche du segment et sa position approximative dans le tableau du manuel.



Broche	Numéro de segment	Position
1	A	Le haut
2	B	Le coin supérieur droit
3	C	Le coin inférieur droit
4	D	Le bas
5	E	Le coin inférieur gauche
6	F	Le coin supérieur gauche
7	G	Le centre
8	Point décimal	Le point

1. D'après les tests ci-dessus, il est établi que l'afficheur du kit utilise une cathode commune. Cela signifie que vous devez simplement connecter la broche commune à la masse (GND) et appliquer une tension élevée aux autres broches pour illuminer les segments correspondants. Si vous souhaitez que l'afficheur affiche le chiffre 2, sur quelles broches devez-vous appliquer une tension élevée ? Pourquoi ?



Pour afficher le chiffre 2, les segments a, b, d, e et g doivent être activés (placés sous une tension élevée) car ce sont eux qui forment le chiffre 2 sur l'afficheur. Les segments f, c et dp (point décimal, s'il est présent) doivent rester éteints (sous une tension faible), car ils ne font pas partie de la représentation du chiffre 2. Ainsi, les broches auxquelles il faut appliquer une tension élevée sont celles connectées aux segments a, b, d, e et g pour afficher correctement le chiffre 2.

Leçon 24 : Lumière défilante avec le 74HC595

Répondez à la question suivante lors de la section "Création de code - Allumage des LED"

1. Que se passe-t-il si nous remplaçons `MSBFIRST` par `LSBFIRST` dans la commande `shiftOut(DS, SHcp, MSBFIRST, B11101110);` ? Pourquoi ?

Si vous remplacez `MSBFIRST` par `LSBFIRST`, l'ordre des bits sera inversé, et l'octet sera décalé en commençant par le bit le moins significatif (celui le plus à droite). Si vous utilisez le registre à décalage pour contrôler des LED, changer l'ordre des bits inversera l'ordre d'allumage des LED. Au lieu de s'allumer dans l'ordre prévu initialement, elles s'allumeront dans l'ordre inverse.

Après avoir terminé la leçon, répondez à la question suivante

1. Si nous voulons avoir trois LED allumées en même temps et qu'elles semblent "défiler", comment faut-il modifier les éléments du tableau `dataArray[]` ?

Vous commenceriez avec les trois premières LED allumées, puis vous décaleriez une LED vers la droite à chaque motif suivant jusqu'à ce que les trois dernières LED soient allumées. Voici comment vous pourriez définir ces motifs en binaire :

```
B11100000: LEDs 1, 2, 3 are on; others are off.  
B01110000: LEDs 2, 3, 4 are on; others are off.  
B00111000: LEDs 3, 4, 5 are on; others are off.  
B00011100: LEDs 4, 5, 6 are on; others are off.  
B00001110: LEDs 5, 6, 7 are on; others are off.  
B00000111: LEDs 6, 7, 8 are on; others are off.
```

```
byte dataArray[] = { B11100000, B01110000, B00111000, B00011100, B00001110,  
B00000111 };
```

Leçon 25 : Afficher un nombre

Remplissez ce tableau pendant la section **“Nombres binaires pour les chiffres de 0 à 9”**

1. Maintenant que nous connaissons les représentations binaires des chiffres 0 et 2, veuillez remplir les nombres binaires pour les autres chiffres dans le tableau ci-dessous.

Nombre	Binaire
0	<i>B00111111</i>
1	<i>B00000110</i>
2	<i>B01011011</i>
3	<i>B01001111</i>
4	<i>B01100110</i>
5	<i>B01101101</i>
6	<i>B01111101</i>
7	<i>B00000111</i>
8	<i>B01111111</i>
9	<i>B01101111</i>

Remplissez ce tableau pendant la section **“Conversion binaire”**

1. Veuillez convertir les nombres binaires représentant les chiffres de 0 à 9 en nombres décimaux et hexadécimaux à l'aide d'une calculatrice, puis complétez le tableau. Cela vous fournira un guide de référence rapide pour les conversions de bases.

Nombre	Binaire	Décimal	Hexadécimal
0	B00111111	63	0x3F
1	B00000110	6	0x06
2	B01011011	91	0x5B
3	B01001111	79	0x4F

4	B01100110	102	0x66
5	B01101101	109	0x6D
6	B01111101	125	0x7D
7	B00000111	7	0x07
8	B01111111	127	0x7F
9	B01101111	111	0x6F