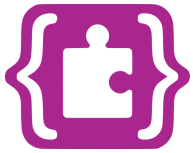


# POTENTIOMÈTRE

#R1AS05



## Disponible sur



## Prérequis

- R1AS01 - Faire clignoter une LED
- R1AS02 - Breadboard
- R1AS04 - Découverte des capteurs de lumière

## Matériel

- 1 carte programmable "**STM32 IoT Node Board**"
- 1 câble USB Micro-B
- 1 breadboard
- 1 potentiomètre
- 1 set de LED
- 1 set de résistances
- Câbles de connexion

## De quoi parle-t-on ?

Dans cette fiche d'activité, nous allons apprendre ce qu'est un potentiomètre en programmant la carte pour régler la luminosité d'une LED en tournant un bouton.

## Durée

20 minutes

## Niveau de difficulté

Intermédiaire

## OBJECTIFS D'APPRENTISSAGE

- Câbler les composants externes à la carte
- Lire une entrée analogique à l'aide d'un potentiomètre
- Utiliser une entrée analogique pour écrire une sortie analogique





Un potentiomètre est une **résistance à trois bornes** avec un contact glissant ou rotatif qui forme un diviseur de tension réglable. Si l'on utilise seulement deux bornes, une extrémité et le curseur, il agit comme une résistance variable ou un rhéostat. L'instrument de mesure appelé potentiomètre est essentiellement un **diviseur de tension** utilisé pour mesurer le potentiel électrique (tension) ; le composant est une mise en œuvre du même principe, d'où son nom.

Les potentiomètres sont couramment utilisés pour **contrôler des dispositifs électriques** tels que les commandes de volume des équipements audio. Les potentiomètres actionnés par un mécanisme peuvent être utilisés comme transducteurs de position, par exemple dans un joystick. Les potentiomètres sont rarement utilisés pour contrôler directement une puissance importante (plus d'un watt) car la puissance dissipée serait comparable à la puissance de la charge contrôlée.

*Ressource: <https://en.wikipedia.org/wiki/Potentiometer>*



## ÉTAPE 1 - CONSTRUIRE



### Câbler le potentiomètre

Connectez la **pin gauche** du potentiomètre à **GND**. La **pin de droite** doit être connectée à 3.3V. Connectez celle du milieu à **A0**.

### Câbler la LED

Connectez l'**anode (+)** de la LED sur **D9**. Connectez la **cathode (-)** de la LED à une **résistance (330 ohms)**. Puis, connectez le côté non connecté de la résistance à **GND**.

### Connecter la carte à l'ordinateur

Avec votre câble USB, connectez la carte à votre ordinateur en utilisant le **connecteur micro-USB ST-LINK** (sur le coin en haut à droite de la carte). Si tout se passe bien, vous devriez voir apparaître sur votre ordinateur un nouveau lecteur appelé **DIS\_L4IOT**. Ce lecteur est utilisé pour programmer la carte en copiant simplement un fichier binaire.

### Ouvrir MakeCode

Allez dans l'**éditeur MakeCode de Let's STEAM**. Sur la page d'accueil, créez un nouveau projet en cliquant sur le bouton "Nouveau projet". Donnez à votre projet un nom plus expressif que "Sans titre" et lancez votre éditeur.

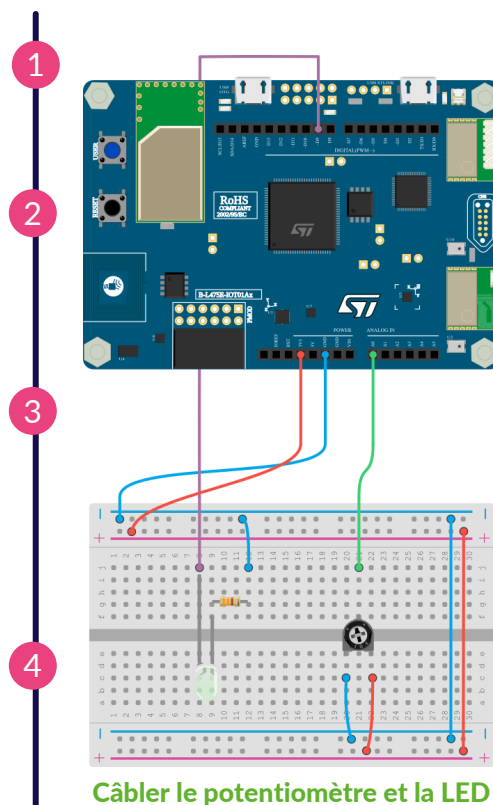
*Ressource: [makecode.lets-steam.eu](https://makecode.lets-steam.eu)*

### Programmer la carte

Dans l'éditeur JavaScript de MakeCode, copiez/collez le code disponible dans la section "**Programmer**" ci-dessous. Si ce n'est pas déjà fait, pensez à donner un nom à votre projet et cliquez sur le bouton "**Télécharger**". Copiez le fichier binaire sur le lecteur **DIS\_L4IOT** et attendez que la carte finisse de clignoter et votre carte est prête !

### Exécuter, modifier, jouer

Votre programme s'exécutera automatiquement chaque fois que vous le sauvegarderez ou que vous réinitialiserez votre carte (appuyez sur le bouton intitulé RESET). Essayez de comprendre l'exemple et commencez à le modifier.



Câbler le potentiomètre et la LED



## ÉTAPE 2 - PROGRAMMER

```
forever(function () {  
  pins.D9.analogWrite(pins.A0.analogRead())  
})
```

### Comment cela fonctionne-t-il ?

Le code se compose de trois éléments :

- un bloc **forever** ;
- un bloc **analogRead** ;
- un bloc **analogWrite**.

Le bloc **forever** met en œuvre "une boucle" qui continue à exécuter les instructions jusqu'à ce que la carte soit éteinte.

Le bloc **analogRead** est utilisé pour obtenir la valeur du potentiomètre sur la *pin* A0. Cette valeur est un nombre entier compris entre 0 et 1023. En tournant le bouton du potentiomètre, on modifie la valeur.



**Le potentiomètre agit comme un diviseur de tension réglable. En changeant la position du bouton du potentiomètre, vous modifiez la tension appliquée sur A0. Plus vous le tournez vers la gauche, plus la tension sera proche de 0V. Plus vous le tournez vers la droite, plus la tension sera proche de 3,3V.**



**Une *pin* d'entrée analogique peut être utilisée pour lire une valeur comprise entre 0 et 1023. Cette valeur est proportionnelle à la tension appliquée à la *pin*, qui doit être comprise entre 0V et 3,3V.**

Le bloc **analogWrite** est utilisé pour allumer la LED sur D9. En utilisant **analogWrite**, la carte est capable de limiter la tension à une certaine valeur pour rendre la LED plus ou moins brillante. La luminosité est définie par la valeur d'**analogRead** sur la *pin* A0 : plus la valeur est élevée, plus la LED est lumineuse.



**En utilisant la *pin* D9, nous sommes en mesure d'écrire une valeur analogique via une broche numérique sur la carte. La *pin* D9, comme quelques autres *pin* de la carte, supporte la modulation de largeur d'impulsion (Pulse Width Modulation ou PWM). Cette technique utilise des fonctionnements tout ou rien pour simuler différentes tensions et donc différents signaux analogiques. La valeur passée à **analogWrite** doit être comprise entre 0 et 255. 0 correspond à une tension de 0V et 255 à 3.3V.**

Comme vous le verrez en utilisant ce programme, vous n'utiliserez pas la gamme complète du potentiomètre. Vous pouvez transformer la plage de valeurs du potentiomètre (0...1023) en plage de valeurs du PWM (0...255) avec la fonction **map**.



## ÉTAPE 3 - AMÉLIORER



En utilisant la fonction **map**, essayez **d'utiliser toute la plage du potentiomètre**. Vous pouvez définir deux variables pour être plus précis et séparer la lecture, la transformation et l'écriture dans des cas spécifiques.

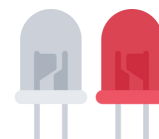
**Ajoutez une autre LED** et inversez la valeur du potentiomètre, de sorte que la deuxième LED s'éteigne lorsque la première s'allume.

Utilisez le potentiomètre pour **contrôler le volume du son d'un buzzer**. Utilisez un potentiomètre pour **contrôler la position d'un servomoteur**.

1



2



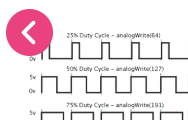
3



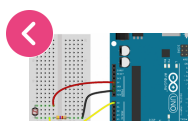
## ALLER PLUS LOIN



**Modulation de largeur d'impulsion** - Tutoriel Arduino sur l'utilisation d'une sortie analogique (PWM) pour faire clignoter une LED. <https://www.arduino.cc/en/Tutorial/Foundations/PWM>



**Diviseurs de tension** - Découvrez comment les diviseurs de tension se comportent. <https://learn.sparkfun.com/tutorials/voltage-dividers>



**Jeu de pong Arduino avec une matrice de 24x16 LED avec MAX7219** - Construire une petite console de pong. <https://www.youtube.com/watch?v=dK9F5AJM2XI>



**Jeu du potentiomètre** - Contrôlez le personnage dans un jeu à l'aide d'un potentiomètre. <https://www.hackster.io/matejadjukic03/potentiometer-game-05ee93?f=1#>



### Explorer d'autres fiches d'activité

**R1AS11 - Fabriquer un thermomètre ... très, très lisible**



**R1AS15 - Collecter des données**

