

RESSOURCES - PROGRAMMATION - FICHE D'ACTIVITÉ 1

# FAISONS CLIGNOTER UNE LED

POUR BIEN DÉBUTER

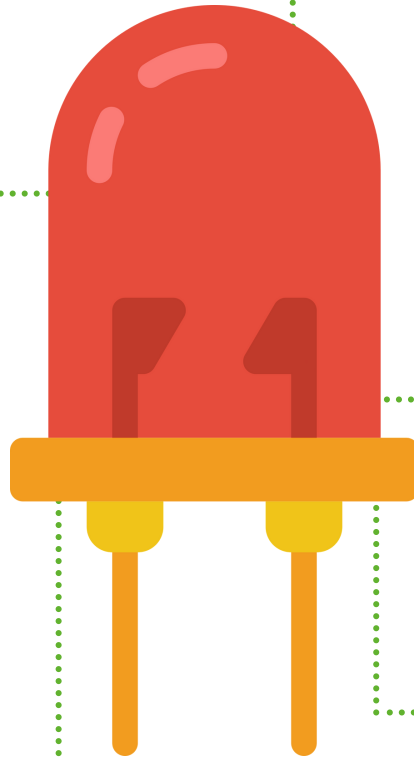
#R1AS01



**Disponible sur**

## Material

- 1 carte programmable "**STM32 IoT Node**"
- 1 câble USB Micro-B



## De quoi parle-t-on ?

Une LED est un composant électronique qui produit de la lumière lorsqu'elle est traversée par un courant. Elle peut être utilisée pour éclairer une pièce, ou pour indiquer quelque chose (un réservoir presque vide, une machine allumée, etc.). Les LED existent sous différentes formes et couleurs.

## Durée

15 minutes

## Niveau de difficulté

Débutant

## OBJECTIFS D'APPRENTISSAGE

- Programmer avec des blocs
- Apprendre les bases de MakeCode
- Utiliser la LED intégrée





Dans cette activité de prise en main, vous allez aborder le concept de **broche**. Une broche est un fil physique connecté directement au microcontrôleur. L'état d'une broche donne des informations sur le passage ou non du courant dans celle-ci. Plus précisément :

- **LOW** signifie qu'il n'y a pas de courant
- **HIGH** signifie qu'il y a du courant.

Pour rendre le courant visible, nous utilisons un composant appelé LED (light-emitting diode) déjà disponible sur la carte, qui s'allumera lorsqu'il passera par la broche.



## ETAPE 1 - CONSTRUIRE



### Connecter la carte à l'ordinateur

Avec votre câble USB, connectez la carte à votre ordinateur en utilisant le **connecteur micro-USB ST-LINK** (en bas à droite de la carte). Vous devriez voir un nouveau lecteur appelé **DIS\_L4IOT** sur votre ordinateur. 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](https://makecode.lets-steam.eu). Sur la page d'accueil, créez un nouveau projet en cliquant sur le bouton "New Project". 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)

### Organiser les blocs

Voici les différentes étapes vous permettant de faire clignoter une LED à l'aide de l'éditeur de blocs :

#### Étape 1 - Ajouter une boucle infinie

Comme nous voulons que le programme fasse clignoter la LED indéfiniment, la première étape consiste à ajouter le bloc **"forever"**. Il sera présent par défaut à l'ouverture de l'éditeur, mais vous pouvez également le trouver dans le sous-menu **BOUCLES**.

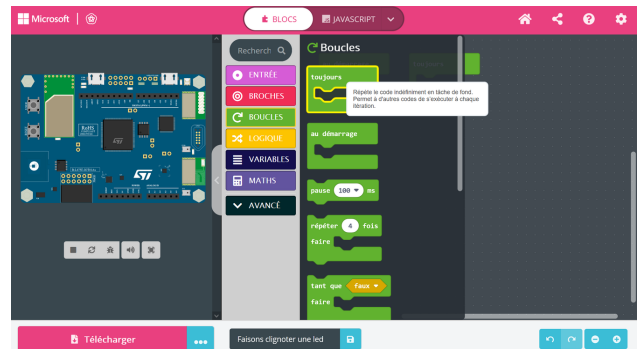
#### Étape 2 - Allumer la LED

Contrôler une LED est une tâche simple car elle ne peut être qu'allumée (le courant la traverse) ou éteinte (le courant ne circule pas). Pour y parvenir, nous devons définir l'état de la broche (plus communément appelée **pin**) où la LED est connectée. Dans notre cas, si nous voulons allumer la LED, nous devons mettre l'état de la broche sur **HIGH**. L'état de la broche sur **LOW** l'éteindra alors. Dans MakeCode, pour contrôler l'état d'une broche, sélectionnez le sous-menu **BROCHES**, puis faites glisser le bloc broche **digital write** à l'intérieur de la boucle **forever**.

1

2

3



Ajouter une boucle infinie en utilisant le bloc **FOREVER**



Sélectionner et éditer le bloc de broches (ou pin) **digital write** pour allumer la LED



## ETAPE 1 - CONSTRUIRE

### Étape 3 - Créer le clignotement

Pour créer le clignotement, il est nécessaire que nous puissions voir la LED s'allumer et s'éteindre pendant une durée similaire. Pour créer ce clignotement, nous devons suivre les étapes suivantes :

1) **Créer une pause lorsque la LED est allumée pour voir la lumière** : avant d'éteindre la LED, nous devons attendre un petit temps, une demi-seconde (500 millisecondes) par exemple, avec la lumière allumée. Pour ce faire, ajoutez le bloc **pause** (à l'intérieur du sous-menu LOOPS), et définissez la valeur à 500 (pour 500 millisecondes).

**i** Vous pouvez choisir une **valeur** dans la liste, ou saisir directement une valeur personnalisée.

2) **Éteignez la lumière pendant une durée similaire pour créer le clignotement** : jusqu'à présent, vous avez fait la moitié du travail ! Ajoutez un autre bloc **digital write** et un autre bloc **pause** pour éteindre la LED et attendre à nouveau 500 ms, permettant de créer cet effet de clignotement. Combiné avec la boucle infinie, on peut voir ce clignotement se répéter à l'infini.

**i** Au lieu de choisir des **blocs** dans les sous-menus, vous pouvez cliquer avec le bouton droit de la souris sur un bloc et le "dupliquer".

Grâce à cette activité facile, vous avez découvert comment créer un morceau de code en utilisant la programmation par blocs. Vous pouvez jeter un coup d'œil à l'éditeur JavaScript pour voir le code généré tel qu'il est indiqué dans la section "**Programmer**" ci-dessous. Dans les prochaines fiches d'activité, n'hésitez pas à copier/coller directement dans l'éditeur JavaScript de MakeCode le code fourni pour voir le résultat en blocs.

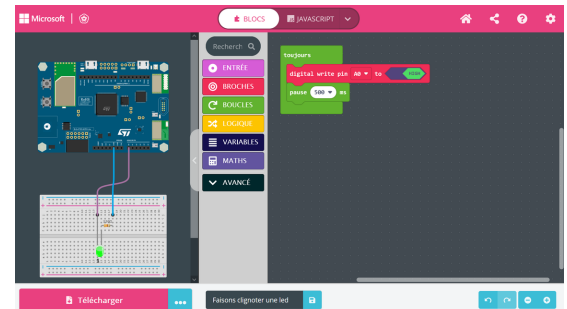
### Programmer la carte

Si ce n'est pas déjà fait, pensez à donner un nom à votre projet et cliquez sur le bouton "**Download**". Copiez le fichier binaire sur le lecteur **DIS\_L4IOT** et attendez que la carte finisse de clignoter. Votre premier programme est maintenant en cours d'exécution et la LED intégrée devrait clignoter !

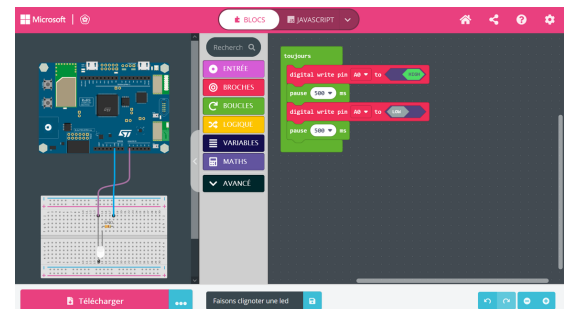
### Exécuter, modifier, s'amuser

Votre programme s'exécutera automatiquement chaque fois que vous le sauvegarderez ou que vous réinitialiserez votre carte (en appuyant sur le bouton RESET). Essayez de comprendre le code et commencez à le modifier en changeant la période entre deux clignotements. N'hésitez pas à essayer de faire clignoter à différents rythmes ou à faire un **SOS** visuel en **morse**.

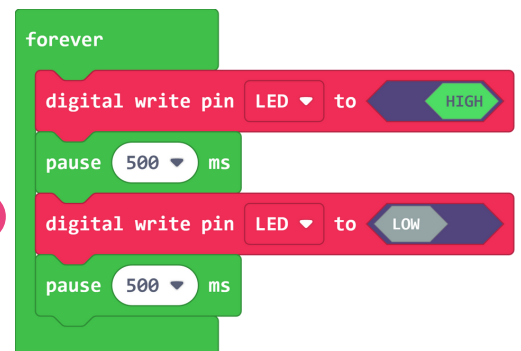
Ressource: <https://en.wikipedia.org/wiki/SOS>



Créez une pause lorsque la LED est allumée pour voir la lumière



Éteignez la lumière pendant une durée similaire à celle du clignement des yeux



Blocs permettant de programmer l'activité d'une LED clignotante

4

5



## ÉTAPE 2 - PROGRAMMER



```
forever(function () {
  pins.LED.digitalWrite(true)
  pause(500)
  pins.LED.digitalWrite(false)
  pause(500)
})
```

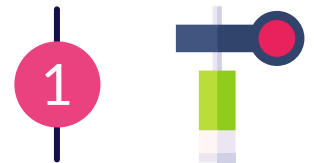
### Comment cela fonctionne-t-il ?

Voici la traduction en JavaScript de notre programme en blocs. Les mots-clefs sont un peu différents, la fonction **digitalWrite** prend un paramètre booléen (**true** ou **false**). Mais la traduction est facile : **true** signifie **HIGH** et **false** signifie **LOW**.

## ÉTAPE 3 - AMÉLIORER



Tentez de créer un **feu de signalisation ferroviaire** en utilisant l'autre LED intégrée appelée LED2



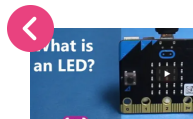
## ALLER PLUS LOIN



**Diode électroluminescente** - Apprenez-en plus sur l'histoire des LED, les principes physiques qui les sous-tendent, les typologies et les couleurs.  
[https://en.wikipedia.org/wiki/Light-emitting\\_diode](https://en.wikipedia.org/wiki/Light-emitting_diode)



**Derrière MakeCode - LED sur micro:bit** - Comment les LED fonctionnent-elles sur la carte micro:bit ? Apprenez tout cela avec Shawn Hymel (en).  
<https://www.youtube.com/watch?v=qqBmvHD5bCw>,  
<https://shawnhymel.com>



**Courant et tension - Bases de l'électricité** - Tutoriel à destination des débutants en électronique pour explorer le courant, la tension, leur différence et leur fonctionnement.  
<https://www.codrey.com/dc-circuits/current-and-voltage/>



**Boucles** - Apprenez-en plus sur les boucles dans MakeCode. <https://makecode.st.com/blocks/loops>



### Explorer d'autres fiches d'activité

**R1AS03 - Bouton et affichage LED**



**R1AS06 - Code Morse**

