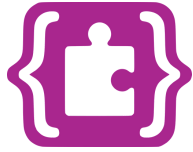


BOUTONS ET AFFICHAGE LED

#R1AS03

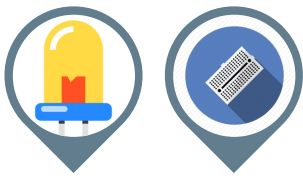


Disponible sur



De quoi parle-t-on ?

Nous allons apprendre à interagir avec la carte en utilisant un simple bouton-poussoir. Il en existe de toutes les formes et de toutes les tailles, mais ils nécessitent tous l'interaction la plus simple : appuyer dessus !

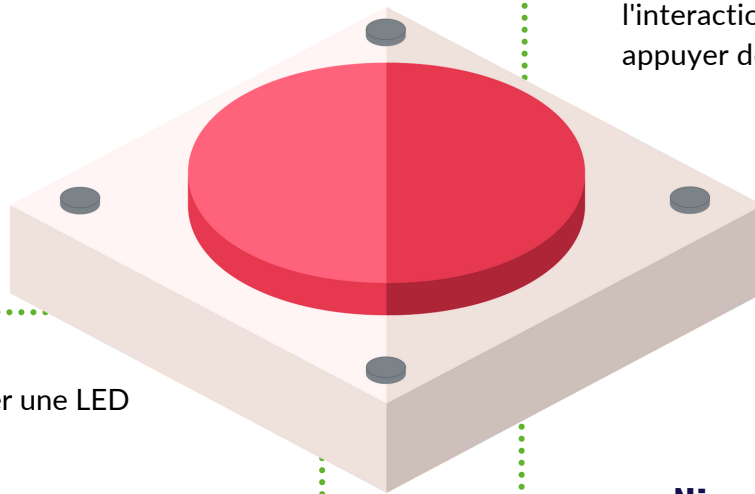


Prérequis

- R1AS01 - Faire clignoter une LED
- R1AS02 - Breadboard

Durée

25 minutes



Niveau de difficulté

Intermédiaire

Matériel

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

OBJECTIFS D'APPRENTISSAGE

- Ajouter de l'interactivité
- Gérer un événement déclenché par un bouton
- Utiliser une variable pour stocker l'état actuel du programme
- Câbler un circuit simple sur une breadboard avec des boutons et des LED
- Utiliser le simulateur MakeCode





Pour apprendre à utiliser un bouton, **jouons à un jeu de questions-réponses !**

L'idée est assez simple : **un animateur, deux joueurs, un bouton, et une LED pour chaque joueur.** Lorsque l'animateur pose une question, les joueurs doivent appuyer en premier sur leur bouton pour donner la bonne réponse. Les LED indiquent quel joueur a appuyé sur son bouton en premier et autorisé à répondre.



ETAPE 1 - CONSTRUIRE



Câbler les boutons et les LED

Connectez une patte de chaque bouton à la broche **GND** de la carte. Connectez ensuite l'autre patte sur la **broche D0** pour le joueur 1, et sur la **broche D1** pour le joueur 2. Connectez l'anode de la LED du joueur 1 sur la **broche A0** et celle du joueur 2 sur la **broche A1**. Connectez la **cathode** de chaque LED à une résistance (330 ohms). Branchez ensuite les pattes non connectées des résistances sur la broche **GND**.



La **LED** a une orientation. Pour désigner l'orientation correcte, chaque branche a un nom. Voici comment trouver la différence entre l'anode et la cathode : **Anode** : C'est le '+' de la LED. La branche anodique est plus longue que la branche cathodique. **Cathode** : C'est le '-' de la LED. La branche de la cathode est plus courte que le fil de l'anode.

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

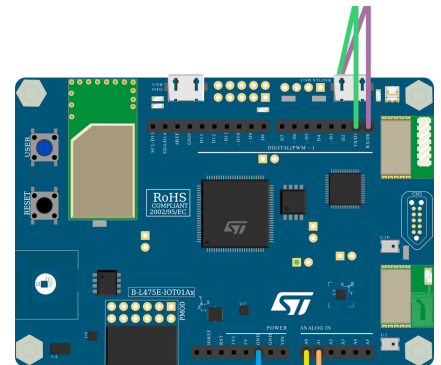
Programmer la carte

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, votre buzzer de quiz est prêt !

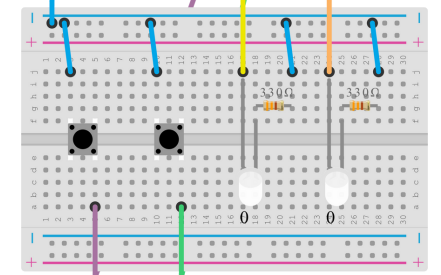
Exécuter, modifier, jouer

Votre programme s'exécutera automatiquement chaque fois que vous le sauvegarderez ou que vous réinitialiserez votre carte (appuyez sur le bouton intitulé RESET). Utilisez les connaissances acquises sur cette fiche d'activité pour réaliser des projets plus ou moins complexes et explorez les fiches d'activité suivantes.

1



2



3

Câbler les boutons et les LED

4

5



ÉTAPE 2 - PROGRAMMER

```
//Initialisation
let weCanPushIt = true
pins.A0.digitalWrite(false)
pins.A1.digitalWrite(false)
```

Initialisation

Dans un premier temps, nous devons déclarer une variable appelée **weCanPushIt**, de type booléen - un type de données avec seulement deux valeurs possibles, généralement "**true**" et "**false**". Cette variable sera utile pour savoir si notre appui sur le bouton sera pris en compte, ou si l'autre joueur est déjà en train de le faire. Les deux dernières lignes donnent l'information que toutes les LED sont éteintes.



Une **variable** est un moyen de nommer et de stocker une valeur pour une utilisation ultérieure par le programme, comme les données d'un capteur ou une valeur intermédiaire utilisée dans un calcul. Une **variable** a un nom et un type. Le type permet de spécifier le type de données que la variable peut contenir. Sur les langages dynamique comme python ou javascript, le type n'est pas forcément explicite.

```
input.buttonD0.onEvent(ButtonEvent.Down, function () {
  if (weCanPushIt) {
    weCanPushIt = false
    pins.A0.digitalWrite(true)
    pause(3000)
    pins.A0.digitalWrite(false)
    weCanPushIt = true
  }
})

input.buttonD1.onEvent(ButtonEvent.Down, function () {
  if (weCanPushIt) {
    weCanPushIt = false
    pins.A1.digitalWrite(true)
    pause(3000)
    pins.A1.digitalWrite(false)
    weCanPushIt = true
  }
})
```

Interactions

Le code principal concerne les interactions des boutons réalisées avec les fonctions **input.buttonXX.onEvent**.



Une **fonction** est un bloc de code qui exécute une tâche spécifique. Elle est vraiment utile pour simplifier le code et rendre un bloc de code plus compréhensible.

La ligne la plus importante ici est la condition **if (weCanPushIt) { ... }** qui teste si les joueurs ont déjà appuyé sur leur bouton ou non. Si cela est le cas (**weCanPushIt** est égal à **true**), nous :

- Affectons **weCanPushIt** à **false**, pour empêcher l'adversaire de répondre,
- Allumons la LED du joueur pour montrer qui est le gagnant,
- Attendons 3 secondes (3 000 millisecondes),
- Éteignons la LED du gagnant,
- Affectons **weCanPushIt** à **true**, pour permettre aux joueurs d'appuyer sur leurs boutons.

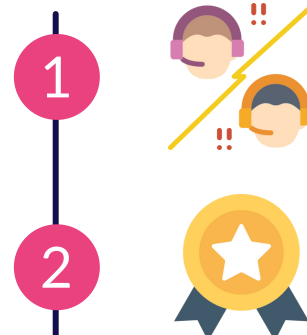


ÉTAPE 3 - AMÉLIORER



Ajoutez d'autres boutons et LED et modifiez votre programme afin de jouer avec plus de participants !

Modifiez votre programme pour **faire clignoter la LED du gagnant** en utilisant la feuille d'activité "Faire clignoter une LED".



ALLER PLUS LOIN



Bouton-poussoir - En savoir plus sur les utilisations du bouton-poussoir.

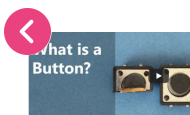
<https://en.wikipedia.org/wiki/Push-button>



Derrière le matériel MakeCode - Boutons sur micro:bi - Tout sur les boutons et leur utilisation dans MakeCode avec Shawn Hymel, influenceur en nouvelles technologies.

https://www.youtube.com/watch?v=t_Qujjd_38o,

<https://shawnhymel.com>



Jeu de réflexes - Créez un jeu de réflexes avec des interrupteurs que vous pouvez frapper aussi fort que vous le souhaitez !

<https://microbit.org/projects/make-it-code-it/reaction-game/>



Découvrez ce qu'est une **variable** - Apprenez-en plus sur les variables et ce qu'est une **fonction** en programmation ? En savoir plus sur les fonctions.

<https://www.computerhope.com/jargon/v/variable.htm>, <https://www.makeuseof.com/what-is-a-function-programming/>



Explorer d'autres fiches d'activité

R1AS04 - Découverte des capteurs de lumière

