

MUSIQUE

CRÉONS UNE MÉLODIE

#R1AS07



Disponible sur



Prérequis

- R1AS02 - Breadboard
- R1AS06 - Code Morse

Matériel

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

De quoi parle-t-on ?

Créons une mélodie agréable à nos oreilles, inspirée des consoles 8 bits.

Durée

30 minutes

Niveau de difficulté

Avancé

OBJECTIFS D'APPRENTISSAGE

- Jouer de la musique avec une carte programmable





Alors que nous faisons beaucoup de bruits à l'aide de buzzers et de haut-parleurs dans diverses fiches d'activité, comme lors de la fabrication d'un thérémine avec un capteur de distance ou le jeu de questions-réponses avec des boutons et des LED, voyons ce qui peut être fait pour créer une mélodie plus agréable pour les oreilles. Nous allons apprendre à jouer quelques notes et tonalités à l'aide d'un programme pour diffuser une mélodie bien connue. Pour rester dans l'ambiance des sons électroniques, nous commencerons par une musique inspirée des consoles 8 bits. Le **chiptune**, également connu sous le nom de musique chip ou musique 8 bits, est un style de musique électronique synthétisée réalisée à l'aide des puces sonores ou synthétiseurs PSG (Programmable sound generator ou générateur de son programmable) des machines d'arcade, ordinateurs et consoles de jeux vidéo vintage.

Resource: <https://en.wikipedia.org/wiki/Chiptune>



ÉTAPE 1 - CONSTRUIRE



Câbler le buzzer/haut-parleur

En théorie, un buzzer ou un haut-parleur n'est pas polarisé (cela signifie qu'il n'y a pas de "+" ni de "-"), mais il y a souvent une paire de fils noir/rouge ou des signes ("+" et/ou "-") sur l'appareil. Si vous êtes dans cette configuration, attachez le fil du côté "+" du buzzer à la *pin* **D3** et l'autre à la *pin* **GND**. S'il n'y a pas de couleur ou d'indication, branchez simplement un fil sur la *pin* **D3** et l'autre sur la *pin* **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.

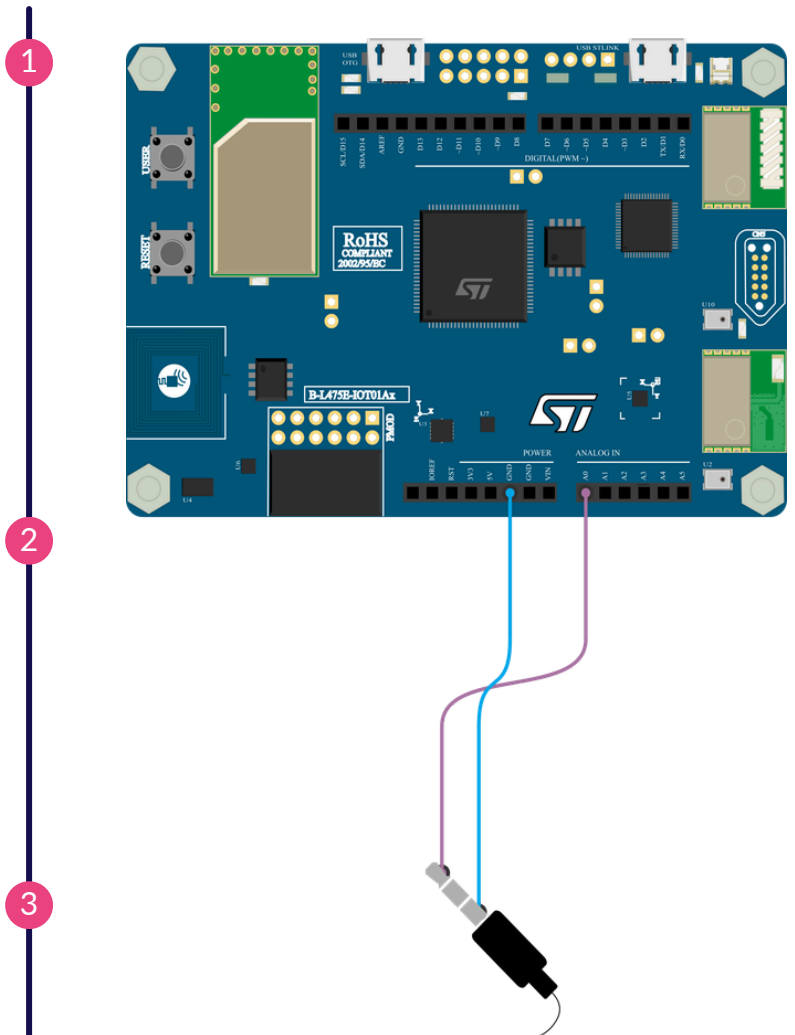
Ouvrir MakeCode

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

Installer l'extension

Après avoir créé votre nouveau projet, vous obtiendrez l'écran par défaut "prêt à l'emploi" illustré ici et vous devrez installer une extension.



Simulateur MakeCode



ÉTAPE 1 - CONSTRUIRE



Les extensions dans MakeCode sont des groupes de blocs de code qui ne sont pas directement inclus dans les blocs de code de base que l'on trouve dans MakeCode. Les extensions, comme leur nom l'indique, ajoutent des blocs pour des fonctionnalités spécifiques. Il existe des extensions pour un large éventail de fonctionnalités très utiles, ajoutant des capacités de manette de jeu, de clavier, de souris, de servomoteurs, de la robotique et bien plus encore.

Vous voyez le bouton noir **AVANCÉ** en bas de la colonne des différents groupes de blocs. Si vous cliquez sur **AVANCÉ**, vous verrez apparaître des groupes de blocs supplémentaires. En bas, il y a une boîte grise appelée **EXTENSIONS**. Cliquez sur ce bouton.

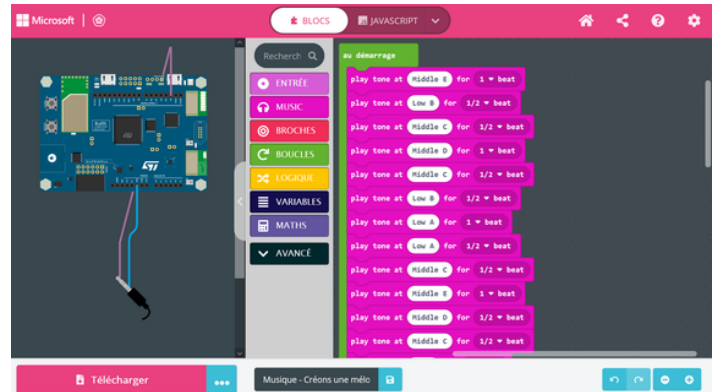
Dans la liste des extensions disponibles, vous pouvez facilement trouver l'extension **Music** qui sera utilisée pour cette activité. Si elle n'est pas directement disponible sur votre écran, vous pouvez la rechercher à l'aide de l'outil de recherche. Cliquez sur l'extension que vous souhaitez utiliser et un nouveau groupe de blocs apparaîtra sur l'écran principal.

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 programme 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). Essayez de comprendre l'exemple et commencez à le modifier en changeant la période entre deux notes.



L'éditeur MakeCode avec l'extension Musique

```

on start
  play tone at Middle E for 1 beat
  play tone at Low B for 1/2 beat
  play tone at Middle C for 1/2 beat
  play tone at Middle D for 1 beat
  play tone at Middle C for 1/2 beat
  play tone at Low B for 1/2 beat
  play tone at Low A for 1 beat
  play tone at Low A for 1/2 beat
  play tone at Middle C for 1/2 beat
  play tone at Middle E for 1 beat
  play tone at Middle D for 1/2 beat
  play tone at Middle C for 1/2 beat
  play tone at Low B for 1 beat
  play tone at Low B for 1/2 beat
  
```

Blocs permettant l'exécution du programme

5

6



ETAPE 2 - PROGRAMMER



```
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(220, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Whole))
music.playTone(349, music.beat(BeatFraction.Half))
music.playTone(440, music.beat(BeatFraction.Half))
music.playTone(440, music.beat(BeatFraction.Half))
music.playTone(392, music.beat(BeatFraction.Half))
music.playTone(349, music.beat(BeatFraction.Half))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(262, music.beat(BeatFraction.Whole))
music.playTone(330, music.beat(BeatFraction.Whole))
music.playTone(294, music.beat(BeatFraction.Half))
music.playTone(262, music.beat(BeatFraction.Half))
music.playTone(247, music.beat(BeatFraction.Whole))
music.playTone(247, music.beat(BeatFraction.Half))
```



ETAPE 2 - PROGRAMMER



Comment cela fonctionne-t-il ?

Ce programme représente une séquence de notes avec une durée. La compréhension de cette activité est davantage liée à la musique qu'à la programmation.

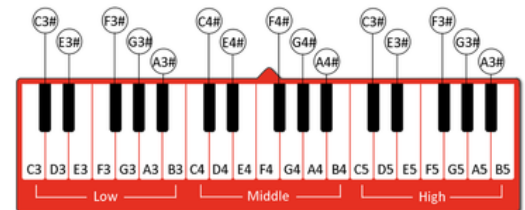
La bibliothèque musicale intégrée à MakeCode nous permet de jouer de la musique sur notre carte. Pour jouer une note, nous utilisons la commande suivante : `play tone at Middle C for 1/2 ▼ beat` Où Middle C = note et 1 beat = 1 temps.

Transcription de chansons à partir de partitions

Si nous voulons recréer nos chansons préférées, nous devons d'abord avoir une connaissance de base des partitions. Voici un rappel des notes les plus courantes utilisées dans une partition musicale :



Pour choisir la bonne note sur MakeCode, vous pouvez cliquer sur le nom de la note et faire apparaître le piano virtuel. Chaque touche correspond à une note spécifique :



Durée de la note

Si nous regardons à nouveau les notes d'une partition de musique, vous remarquerez qu'elles ont des formes et des couleurs différentes. Ces différentes formes et couleurs indiquent les différentes durées appelées valeur de note et exprimées en nombre de temps (beat).

Notes	Name	Value	Code
	Semibreve Whole note	4 beat	4 ▼ beat
	Minim Half note	2 beat	2 ▼ beat
	Crotchet Quarter note	1 beat	1 ▼ beat
	Quaver Eighth note	1/2 beat	1/2 ▼ beat
	Semiquaver Sixteenth note	1/4 beat	1/4 ▼ beat



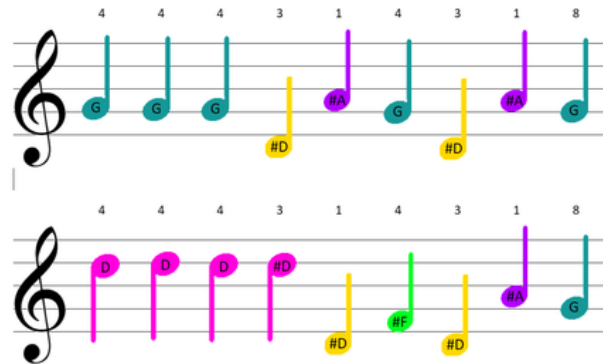
ÉTAPE 3 - AMÉLIORER



Écrivez un programme qui **joue la partition suivante** :



Essayez de **reproduire le thème de Dark Vador** avec MakeCode à partir de cette partition :



En utilisant un **capteur de distance** comme **détecteur de présence**, créez un programme qui **joue la musique de votre choix** chaque fois qu'il détecte quelque chose.

1

2

3

ALLER PLUS LOIN



233 projets musicaux utilisant Arduino.
<https://create.arduino.cc/projecthub/projects/tags/music>

**Comment faire de la musique avec des micro:bits**

- En utilisant des pinces crocodiles, vous pouvez connecter toutes sortes de choses à votre micro:bit, y compris un haut-parleur.

<https://www.youtube.com/watch?v=bm7MGKspk0o>

**Codage avec micro:bit - Partie 4 - Faire de la musique**

- Jetez un oeil aux fonctionnalités son et audio de micro:bit et testez une variété de différents buzzers et haut-parleurs.

https://www.youtube.com/watch?v=6hxvLZSM_pM



Faire de la musique avec micro:bit - Utiliser la bibliothèque musicale intégrée à MakeCode pour jouer de la musique sur notre micro:bit.

<https://www.teachwithict.com/microbit-music.html>



Explorer d'autres fiches d'activité

R1AS12 - Alarme de détecteur de mouvement



R1AS08 - Fabriquer un thérémine avec le capteur de distance

