FABRIQUER UN THÉRÉMINE

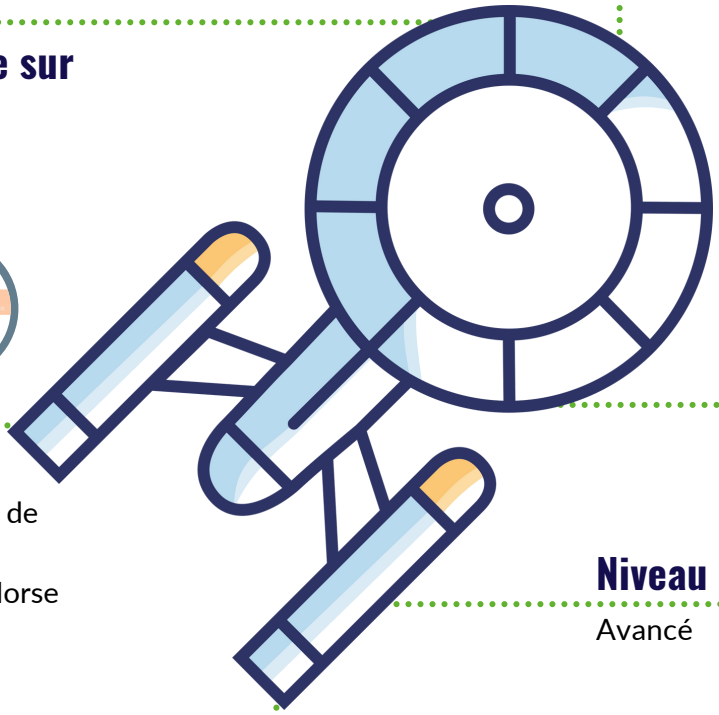
AVEC LE CAPTEUR DE DISTANCE

#R1AS07

Disponible sur

De quoi parle-t-on ?

Un thérémine est un instrument de musique électronique dont on peut jouer sans le toucher. Le concept original est basé sur l'utilisation de deux antennes pour détecter la position des mains. Une antenne est utilisée pour le volume, et l'autre pour la hauteur du son.



Durée

30 minutes

Prérequis

- R1AS04 - Détecteur de lumière basique
- R1AS06 - Le code Morse

Niveau de difficulté

Avancé

Matériel

- 1 carte programmable "***STM32 IoT Node Board***"
- 1 câble USB Micro-B
- 1 buzzer piézoélectrique ou un haut-parleur
- 1 breadboard
- Câbles de connexion

OBJECTIFS D'APPRENTISSAGE

- Utiliser un capteur de distance et comprendre son fonctionnement
- Faire de la musique avec un instrument vraiment étrange
- Utiliser la fonction map pour transformer un nombre d'une plage à une autre

Cette fiche d'activité fait partie du projet Let's STEAM financé avec le soutien de la Commission européenne par le biais du programme de partenariat stratégique Erasmus +. Son contenu n'engage que son auteur et la Commission ne peut être tenue responsable de l'usage qui pourrait être fait des informations qu'elle contient.

Cofinancé par le programme Erasmus+ de l'Union européenne

FABRIQUER UN THÉRÉMINE AVEC LE CAPTEUR DE DISTANCE

Le thérémine est un instrument de musique électronique contrôlé sans contact physique par le théréministe (interprète). Il doit son nom à son inventeur, Léon Thérémine, qui a fait breveter l'appareil en 1928. La section de contrôle de l'instrument se compose généralement de deux antennes métalliques qui détectent la position relative des mains du théréministe et contrôlent les oscillateurs pour la fréquence avec une main et l'amplitude (volume) avec l'autre. Les signaux électriques émis par le thérémine sont amplifiés et envoyés à un haut-parleur.
Notre version sera plus simple, nous ne contrôlerons que la hauteur du son, avec le capteur de distance, le volume sera prédéterminé. **Faisons de la musique !**
Ressources : https://en.wikipedia.org/wiki/Theremin, https://youtu.be/x0NVb25p1oU

ÉTAPE 1 - CONSTRUIRE

Câbler le buzzer/haut-parleur

1

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* **A2** 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* **A2** et l'autre sur la *pin* **GND**.

Connecter la carte à l'ordinateur

2

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

3

Allez dans l'éditeur MakeCode de Let's STEAM. 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

Programmer la carte

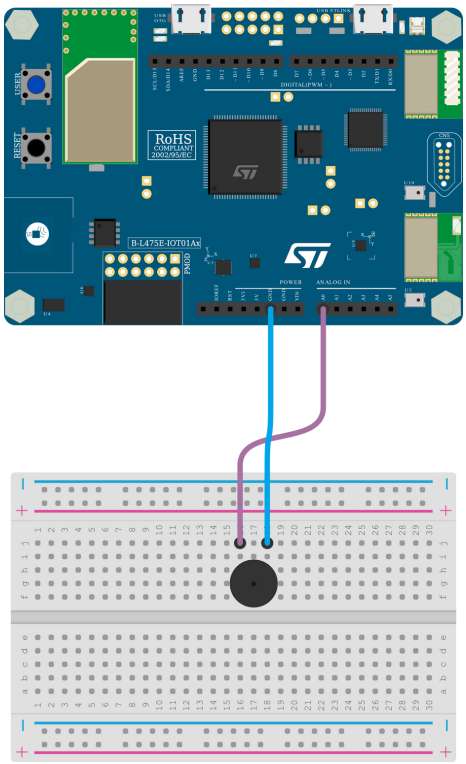
4

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 "**Download**". 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

5

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 buzzer/haut-parleur

FABRIQUER UN THÉRÉMINE AVEC LE CAPTEUR DE DISTANCE

ÉTAPE 2 - PROGRAMMER

```
let distance = 0
forever(function () {
   // Mesurer la distance
   distance = input.distance(DistanceUnit.Millimeter)

   if (distance > 500) {
      // Convertir la distance en fréquence
      let note = Math.map(distance, 0, 500, 440, 830)
      music.ringTone(note)
   } else {
      music.stopAllSounds()
   }
})
```

Variables

Dans ce programme, il y a 2 variables. La première, **distance**, est utilisée pour mémoriser la distance et déterminer la note à jouer. Ensuite, il y a **note**, qui n'est pas techniquement nécessaire/obligatoire mais permet d'introduire une plus grande compréhension de chaque étape du programme. Elle contient la transformation de la distance en fréquence du ton.

Mesurer la distance

Cette opération ne présente aucune difficulté. Nous devons appeler la fonction **input.distance(DistanceUnit.Millimeter)**. Le paramètre **DistanceUnit.Millimeter** indique à la fonction que nous voulons le résultat en millimètres (1 mètre = 1 000 millimètres).

Condition

La condition **if (distance > 500) { ... }** donne l'information que nous ne jouons un son que si la distance mesurée est inférieure ou égale à 500 millimètres.

Convertir la distance en fréquence

La partie la plus importante est la conversion. Pour la réaliser, nous utilisons une fonction mathématique appelée **map**. Cette fonction transforme une valeur d'une plage à une autre. Dans ce cas, la valeur est transférée de la plage de distances à la plage de fréquences. Comme vous pouvez le voir dans le code ci-dessus, cette fonction prend cinq paramètres, à savoir : **value**, **in_min**, **in_max**, **out_min**, **out_max**. Examinons de plus près chacun d'entre eux :

- **value :** la valeur à transformer
- **in_min :** la valeur minimale de la plage d'entrée (distance)
- **in_max :** la valeur maximale de la plage d'entrée (distance)
- **out_min :** la valeur minimale de la plage de sortie (fréquence)
- **out_max :** la valeur maximale de la gamme de sortie (fréquence)

Nous pouvons donc comprendre ce que fait cette ligne, c'est-à-dire transformer la distance (avec une plage de 0mm à 500mm) en fréquence (avec une plage de 440Hz à 830Hz).

Les fréquences choisies ne sont pas aléatoires, la gamme de fréquence de 440Hz à 830Hz représente une octave. Cela signifie que vous pouvez trouver toutes les notes : LA, SI, DO, RE, MI, FA, SOL

Nous avons maintenant une fréquence. Il est temps de la jouer, en utilisant simplement la fonction **music.ringTone(note)**.

FABRIQUER UN THÉRÉMINE AVEC LE CAPTEUR DE DISTANCE

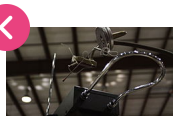
ÉTAPE 3 - AMÉLIORER

1. **Modifiez la valeur de map** pour ajouter des octaves et/ou de la distance afin d'améliorer votre musique.

2. Essayez **d'ajouter un potentiomètre** pour contrôler le volume.

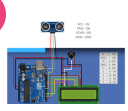
ALLER PLUS LOIN

Thérémine - - Découvrez l'histoire, les principes de fonctionnement et les utilisations du thérémine.
https://en.wikipedia.org/wiki/Theremin

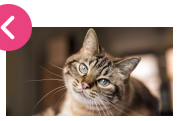


Capteur de distance à anneau LED - Découvrez un projet amusant, qui aboutira à un capteur de recul .
https://www.instructables.com/LED-Ring-Distance-Sensor/

Détecteur du niveau d'eau - Découvrez les capteurs à ultrasons qui convertissent l'énergie électrique en ondes acoustiques.
https://www.instructables.com/Water-Level-Detector-2/



Mangeoire pour chats - Utilisez un capteur à ultrasons pour construire une mangeoire automatique pour chats.
https://www.instructables.com/Cat-Feeder/



Explorer d'autres fiches d'activité

R1AS05 - Potentiomètre