

MAGNETICS

Créer des projets multicartes



Disponible sur

Durée

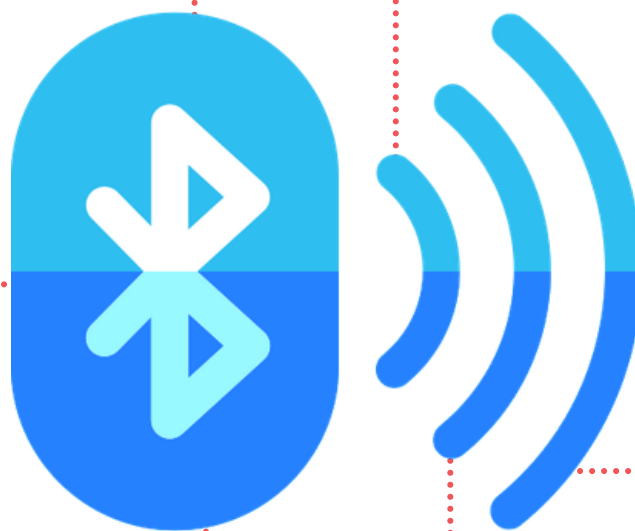
50 minutes

Matériel

- 4 cartes programmables "**STM32 IoT Node Board**"
- 1 câble USB Micro-B
- 1 écran OLED Monochrome 1.3" 128x64 OLED de Adafruit
- 1 câble QT pour connecter l'écran à la carte

De quoi parle-t-on ?

Dans cette activité, nous allons programmer plusieurs cartes électroniques et échanger des données de capteurs à l'aide de l'extension Magnetics, qui permet une communication maillée sans fil.



Coopération



Niveau de difficulté

Avancé

OBJECTIFS D'APPRENTISSAGE

- Utiliser l'extension magnetics pour construire des projets multicartes
- Appréhender le fonctionnement des modules bluetooth BLE Mesh





Cette fiche d'activité propose de créer des projets plus complexes en utilisant plusieurs cartes électroniques non connectées entre elles. Une fois les capteurs maîtrisés, nous pouvons en effet mettre en place des expériences nécessitant l'utilisation de plusieurs cartes. Afin de réaliser la collecte des données, il faut pouvoir faire communiquer les cartes entre elles par les airs. Dans cette activité nous allons programmer plusieurs cartes électroniques et échanger des données de capteurs à l'aide de l'extension **Magnetics** que permet de mettre en œuvre une communication sans fil maillée. Le projet magnetics prend la forme d'une brique technique logicielle implantée directement dans MakeCode. Ce développement est basé sur l'utilisation de la technologie de réseau maillé **Bluetooth Low Energy Mesh** (BLE Mesh) compatible avec toutes les cartes programmables disposant d'un module **Bluetooth Low Energy**.

Ressources : <https://www.magnetics.edu-up.fr/>

<https://blog.rtone.fr/bluetooth-mesh>

https://fr.wikipedia.org/wiki/Bluetooth_%C3%A0_basse_consommation



ÉTAPE 1 - CONSTRUIRE



Pour réaliser cette activité nous avons besoin de **quatre cartes STM32 IoT Nodes**. Trois d'entre elles seront **émettrices** de données de capteurs (température, humidité, pression), et la dernière sera **collectrice** des données qu'elle affichera sur un écran OLED. Mis à part l'écran de la dernière, il n'y a pas de câblage car nous utiliserons uniquement les capteurs internes. Nous allons donc vous donner la marche à suivre pour câbler et programmer en premier lieu la carte collectrice puis dans un second temps, programmer individuellement chaque carte émettrice afin de pouvoir construire votre projet.

Activité 1 - Préparer, câbler et programmer la carte collectrice

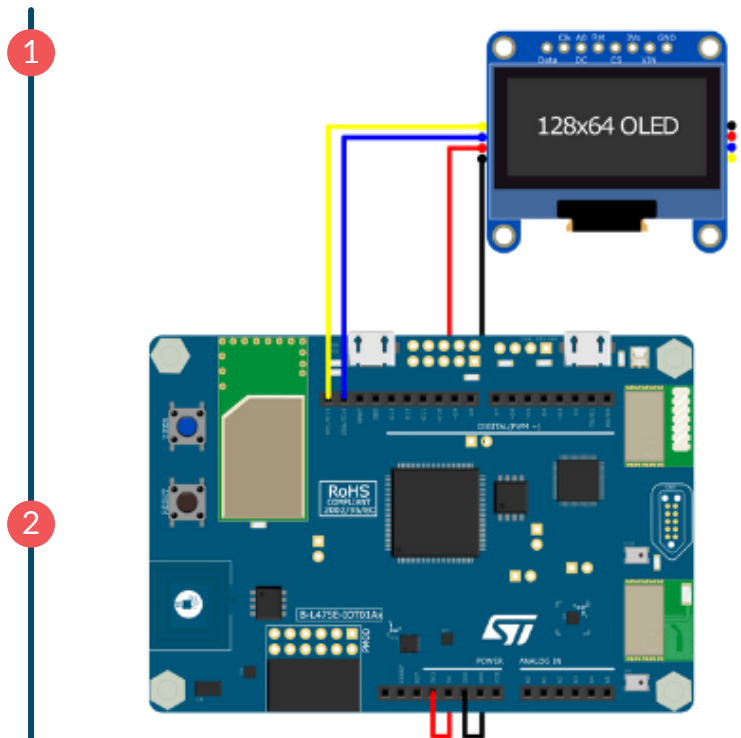
Câbler l'écran OLED

Nous devons en premier lieu câbler l'écran OLED directement à la carte collectrice. Il y a deux façons de câbler l'écran **OLED SSD1306** à une carte, soit avec une connexion **I2C** ou **SPI**. Pour notre écran, nous utilisons la connexion **I2C** via le câble **QWIIC/STEMMA** avec la convention suivante :

- **Noir** pour **GND**
- **Rouge** pour **V+ (3V3)**
- **Bleu** pour **SDA (D14)**
- **Jaune** pour **SCL (D15)**

Connecter la carte à l'ordinateur

Avec votre câble USB, connectez la carte collectrice à votre ordinateur en utilisant le connecteur micro-USB. Si tout se passe bien, vous devriez voir apparaître sur votre ordinateur un nouveau lecteur.



Câblage de l'écran OLED sur la carte collectrice



ÉTAPE 1 - CONSTRUIRE

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 les extensions

Après avoir créé votre nouveau projet, vous obtiendrez l'écran par défaut et vous devrez installer deux extensions pour ce projet spécifique : "magnetics" et "oled".



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.

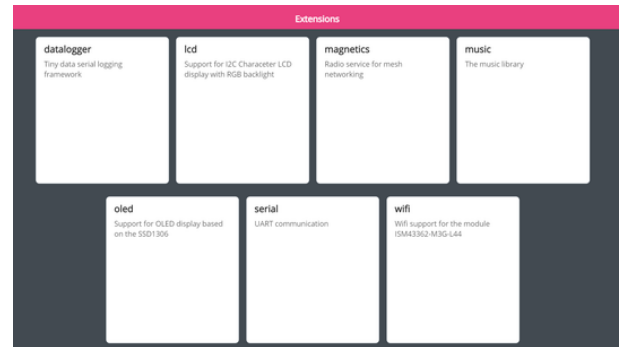
Vous voyez le bouton noir **AVANCÉ** en bas de la colonne des différents groupes de blocs. En cliquant sur ce bouton, vous verrez apparaître des groupes de blocs supplémentaires. En bas, il y a une boîte grise appelée **EXTENSIONS**. Cliquez sur ce bouton.

Dans la liste des extensions disponibles, vous pouvez trouver l'extension **Magnetics** qui sera utilisée pour cette activité. Si elle n'est pas directement disponible sur votre écran, vous pouvez la rechercher en utilisant la barre de recherche. Cliquez sur l'extension et un nouveau groupe de blocs apparaîtra sur l'écran principal. Réitérez cette action pour installer l'extension **OLED** également disponible. Vous pouvez commencer à programmer.

Programmer la carte

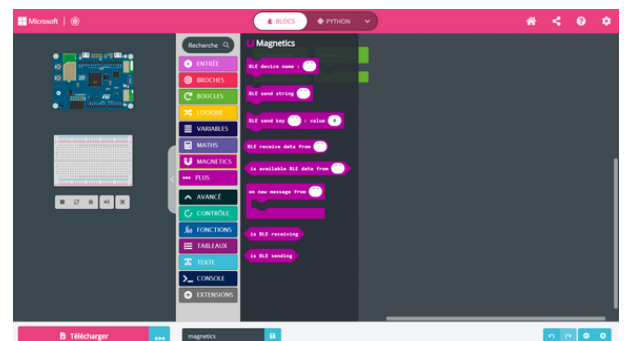
Dans l'éditeur JavaScript de MakeCode, copiez/collez le code disponible dans la section "**Programmer**" ci-dessous. Cliquez sur le bouton "**Télécharger**" et attendez que la carte finisse de clignoter.

3



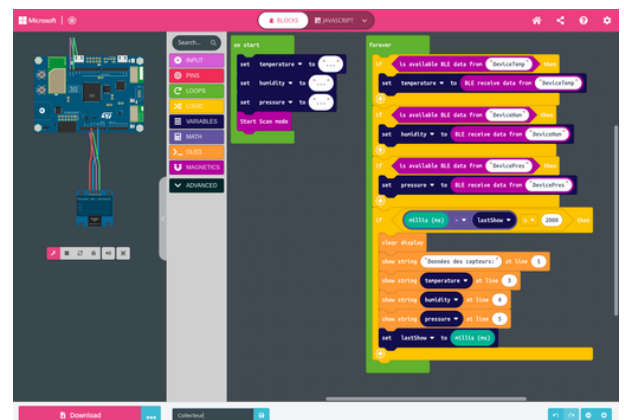
Liste d'extensions et outil de recherche

4



Blocs associés à l'extension magnetics

5



Capture d'écran de MakeCode avec les blocs du programme du collecteur



ETAPE 1 - CONSTRUIRE

Activité 2 - Programmer chaque carte émettrice individuellement

Une fois la carte collectrice câblée avec l'écran OLED et programmée, nous pouvons préparer les trois cartes émettrices en suivant les mêmes étapes de programmation que lors de l'étape 1. Pour chaque carte, il faudra donc effectuer les tâches suivantes :

Connecter la carte à l'ordinateur

Avec votre câble USB, connectez la carte émettrice que vous souhaitez programmer à votre ordinateur en utilisant le **connecteur micro-USB**. Si tout se passe bien, vous devriez voir apparaître sur votre ordinateur un nouveau lecteur. 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

De la même manière qu'à l'étape 1, ajoutez l'extension magnetics à la liste de bloc via le menu "**AVANCÉ**" puis "**EXTENSIONS**".

Programmer la carte

Dans l'éditeur JavaScript de MakeCode, copiez/collez le code disponible dans la section "**Programmer**" ci-dessous relatif à la carte émettrice que vous êtes en train de programmer. Si ce n'est pas déjà fait, pensez à donner un nom à votre projet et cliquez sur le bouton "**Télécharger**" et attendez que la carte finisse de clignoter.

6

7

8

9

```

on start
  BLE device name : "DeviceTemp"
  Start Send mode

forever
  BLE send key "Temp (°C)" : value temperature in °C
  pause 500 ms
  
```

Exemple de la programmation par bloc relative à la carte émettrice collectant les données de température

Activité 3 - Exécuter, modifier, jouer

Une fois votre carte collectrice reliée à votre écran, et vos quatre cartes programmées, votre programme est prêt à être utilisé. Votre programme s'exécutera automatiquement chaque fois que vous les mettez sous tension ou que vous réinitialiserez votre carte (appuyez sur le bouton intitulé RESET). Essayez de comprendre le fonctionnement de votre code et commencez à le modifier en créant vos propres projets.



ETAPE 2 - PROGRAMMER



Code de la carte collectrice

```
let temperature = "...";
let humidity = "...";
let pressure = "...";
let lastShow = 0;

magnetics.startScanning()
forever(function () {

  if (magnetics.availableDataFromName("DeviceTemp")) {
    temperature = magnetics.readDataFromName("DeviceTemp")
  }
  if (magnetics.availableDataFromName("DeviceHum")) {
    humidity = magnetics.readDataFromName("DeviceHum")
  }
  if (magnetics.availableDataFromName("DevicePres")) {
    pressure = magnetics.readDataFromName("DevicePres")
  }

  if( control.millis() - lastShow >= 2000 ){
    oled.clear()
    oled.showString("Données des capteurs:", 1)
    oled.showString(temperature, 3)
    oled.showString(humidity, 4)
    oled.showString(pressure, 5)

    lastShow = control.millis();
  }
})
```

Code de la carte émettrice de la température

```
magnetics.setLocalName("DeviceTemp")
magnetics.startEmitting()
forever(function () {
magnetics.setAdvertisingKeyValueData("Temp (°C)", input.temperature(TemperatureUnit.Celsius))
  pause(500)
})
```



ETAPE 2 - PROGRAMMER



Code de la carte émettrice de l'humidité

```
magnetics.setLocalName("DeviceHum")
magnetics.startEmitting()
forever(function () {
  magnetics.setAdvertisingKeyValueData("Hum (%)", input.humidity())
  pause(500)
})
```

Code de la carte émettrice de la pression

```
magnetics.setLocalName("DevicePres")
magnetics.startEmitting()
forever(function () {
  magnetics.setAdvertisingKeyValueData("Pres (hPa)", input.pressure(PressureUnit.HectoPascal))
  pause(500)
})
```

Comment cela fonctionne-t-il ? Initialiser la collecte de données :

Cela fait beaucoup de code, mais rien de très complexe, puisque le code des **cartes émettrices** est quasiment identique (il n'y a que le capteur qui change) et reste simple à comprendre.

Pour commencer, on donne un nom à notre carte à l'aide de la fonction `setLocalName`, ce qui permettra au collecteur de reconnaître la donnée envoyée. Pour que la carte puisse émettre des données, il faut lui préciser son rôle, c'est ce que fait la fonction `startEmitting`. Enfin, via la fonction `setAdvertisingKeyValueData`, nous envoyons la donnée du capteur toutes les 500 millisecondes (c'est-à-dire deux fois par seconde).

Enfin, intéressons-nous au code du collecteur. Au début, nous déclarons 4 variables:

- `temperature`, `humidity` et `pressure` qui contiendront les données émises par les émetteurs (respectivement pour la température, l'humidité et la pression atmosphérique)
- `lastShow` qui permet de savoir à quand remonte le dernier affichage des données (en millisecondes).

Une fois les variables initialisées, nous faisons appel à `startScanning`, pour rechercher les cartes émettrices qui sont à proximité. Le reste du code se trouve dans la fonction `forever` ce qui implique, que nous allons répéter indéfiniment le code suivant. Les trois premiers `if(...)` permettent de savoir si nous avons reçu des données de la part des émetteurs (grâce à la fonction `availableDataFromName`), si tel est le cas, alors on enregistre cette donnée dans la variable associée (en utilisant `readDataFromName`).

Maintenant que nous avons les données, il faut les afficher sur notre écran OLED, c'est le rôle de la dernière condition `if`, qui nous permet de mettre à jour notre écran avec les nouvelles données, seulement si cela fait plus de deux seconds qu'il n'a pas été rafraîchi.



ETAPE 3 - AMÉLIORER



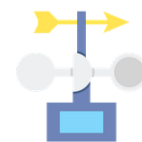
Cette activité ne fait qu'afficher les informations sur un écran, mais il est possible **d'utiliser le datalogger, pour enregistrer nos données et les analyser.**

Utiliser une boîte étanche et **faire une station météo**, la connexion étant sans fil, on pourrait avoir un écran en intérieur nous affichant **les données environnementales de l'extérieur.**

1



2



ALLER PLUS LOIN



Bluetooth - Découvrir plus d'informations sur cette norme de télécommunication si présente dans notre vie quotidienne

<https://fr.wikipedia.org/wiki/Bluetooth>



CircuitPython BLE Libraries on Any Computer - Utilisez le code BLE de CircuitPython sur les ordinateurs de bureau, les ordinateurs portables et les Raspberry Pi grâce aux bibliothèques Adafruit

<https://learn.adafruit.com/circuitpython-ble-libraries-on-any-computer>



Bouton de volume BLE avec CircuitPython - Transformez votre Bluefruit de Circuit Playground en bouton de volume BLE sans fil

<https://learn.adafruit.com/bluetooth-le-hid-volume-knob-with-circuitpython>



Explorer d'autres fiches d'activité du projet Let's STEAM

R1AS15 - Collecter les données

https://github.com/letssteam/Resources/blob/main/3_AS_Programming/FR/LS_R1AS15_Data_FR.pdf



R1AS11 - Fabriquer un thermomètre

https://github.com/letssteam/Resources/blob/main/3_AS_Programming/FR/LS_R1AS11_Thermometer_FR.pdf

