



## Liaison radio et récupération des données de la carte Micro:bit

Matériel : par binôme : une carte micro:bit, une connexion internet, un ordinateur région muni du navigateur chrome, un afficheur LCD.

1

### Liaison Radio de la carte Micro:bit (30min)

Les technologies sans fil sont basées sur les concepts de la physique : les ondes radio ont certaines propriétés (comme l'amplitude, la pulsation ou la période) modulées par un émetteur de façon à ce que cette information puisse être encodée et ainsi diffusée.

Lorsque des ondes radio rencontrent un conducteur électrique (c'est-à-dire une antenne), elles provoquent l'apparition d'un courant alternatif duquel l'information contenue dans les ondes peut être extraite et retraduite dans sa forme originale.

#### a) Utilisation de la radio et longueur des données

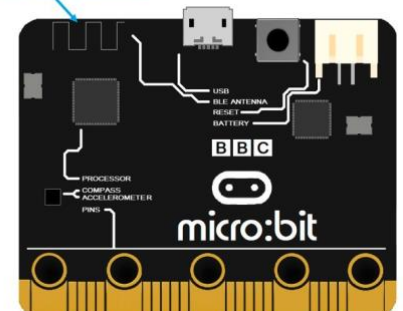
Le but est de recevoir sur une 1<sup>ère</sup> carte la donnée envoyée par une 2<sup>ème</sup> carte en appuyant sur le bouton A de la micro:bit.

Connectez vous au WIFI du lycée et ouvrir « Vittascience » → programmation micro:bit → mode bloc/python

Brancher les cartes une par une et transférer le code suivant :

**Attention** : Choisissez un numéro de canal différent de tous les autres groupes !

Antenne Radio et Bluetooth



Le code Python visible à droite de l'interface est le suivant :

```

1 from microbit import *
2 import radio
3
4 def radio_receiveData():
5     data = radio.receive()
6     if data:
7         if data.find('&&int:') != -1:
8             return int(data[7:-3])
9         elif data.find('&&float:') != -1:
10            return float(data[9:-3])
11        elif data.find('&&bool:') != -1:
12            return bool(data[8:-3])
13        else:
14            return data
15    else:
16        return None
17
18 radio.on()
19
20 radio.config(channel = 7, power = 6, length = 8)
21
22 while True:
23     if button_a.is_pressed():
24         radio.send('Les ondes se promènent !!')
25         radioData = radio_receiveData()
26     if radioData:
27         display.scroll(str(radioData))
  
```

Appuyer sur le bouton A d'une des cartes. Observer l'autre carte après l'avoir branchée à une batterie externe.

Le message « Les ondes se promènent ! » a-t-il été reçu et s'affiche-t-il ?

Le bloc "Configurer la radio" permet de modifier certains paramètres de transmission radio. Il est possible de choisir la taille des données, la puissance de l'émetteur/récepteur radio et le groupe sur lequel est transmis les données.

Premièrement, nous modifions la taille des données. Elle est définie en nombre d'octets pouvant aller de 0 à 251. On envoie ici une chaîne de caractère assez longue (Ex : "Les ondes se promènent") avec 8 octets comme taille de données.

Q1 : Voyez- vous toute la chaine de caractère ?.....

En effet, avec une taille de 8 octets, la chaine de caractère n'a pas pu être envoyée totalement.

Q2 : Quel paramètre faut-il modifier pour obtenir toute la phrase ? ..... . Faites-le pour vérifier !

**b) Portée de la radio**

Il est également possible de paramétrer la puissance de l'émission/transmission de la radio.

La puissance joue sur la distance maximale entre les cartes où l'on peut recevoir des données.

Il est possible de définir la puissance de 0 à 7 (inclus), sachant qu'une augmentation de puissance augmente la portée mais aussi la consommation énergétique de la carte.

Éloignez-vous et tenter régulièrement d'envoyer des données.

Q3 : Au bout de combien de mètres, la transmission ne s'effectue plus, c'est-à-dire que la carte n'affiche plus la chaîne de caractères ?.....

**c) Défi 1 : A vous de jouer...**

Le but est de manipuler les LED de l'écran d'une carte voisine avec la transmission radio. On manipule donc un tableau de 5\*5 LED. On a les conditions suivantes :

Carte émettrice :

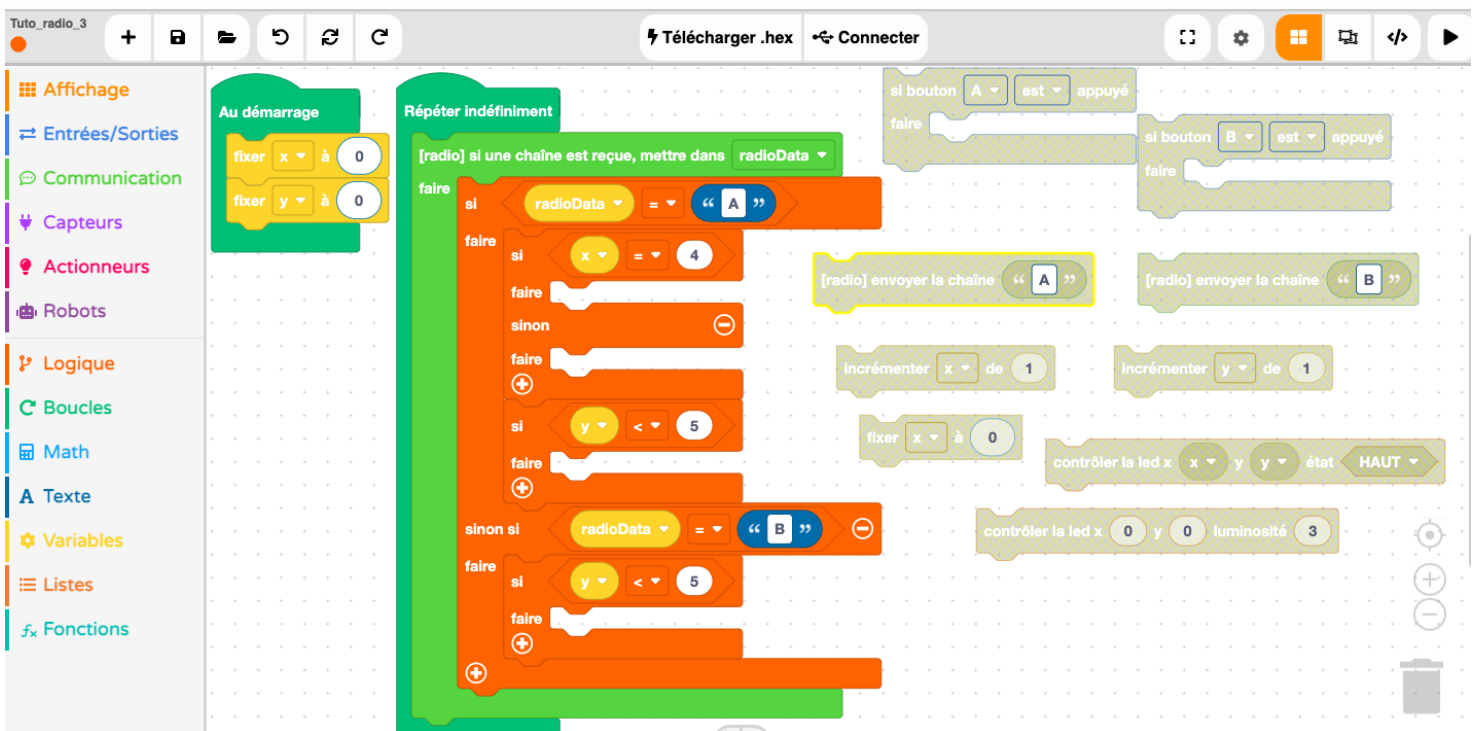
- La commande "A" est envoyée lorsqu'on clique sur le bouton A
- La commande "B" est envoyée lorsqu'on clique sur le bouton B

Carte réceptrice :

- La commande "A" déplace le curseur.
- La commande "B" allume la LED où se trouve le curseur.



- Astuce : Utiliser les blocs "incrémenter de 1" de la catégorie "Variables".
- Aide : il faut utiliser les blocs ci-dessous en modifiant parfois les variables...



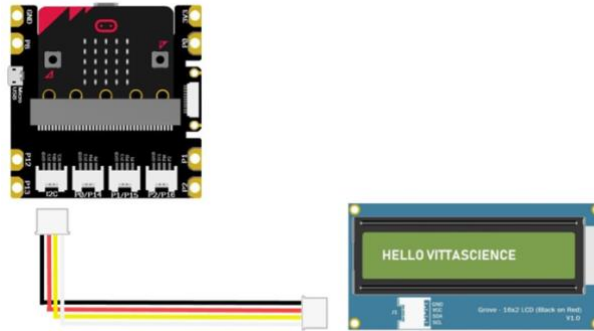
Compléter le code avec les blocs au bon endroit et transférer-le vers les deux cartes micro:bit (N'oubliez pas de connecter une des cartes à une batterie externe !)

Tout fonctionne ? Pour vérifier, la solution se trouve en fin de document !

2

## Afficher du texte sur un écran LCD avec Micro:bit (15 min)

### a) Schéma du montage :



### b) Utilisation de l'afficheur LCD :

Le bloc "Afficher sur la ligne le texte" de la catégorie "Affichage", rubrique modules externes, permet de générer le code nécessaire à l'affichage de texte sur un écran LCD Grove compatible en 3,3V. Brancher la carte micro:bit au shield Grove ainsi que l'afficheur LCD sur le port I2C, puis transférer le code pour afficher un premier texte sur l'écran LCD : HELLO VITTASCIENCE

L'alphabet est stocké dans la mémoire EEPROM de l'écran LCD. Dès qu'on souhaite écrire un caractère, celui-ci est traduit en une adresse mémoire, et c'est ainsi que le LCD parvient à l'afficher.

L'afficheur disposant de 2 lignes, il est possible de choisir la ligne sur laquelle sera affichée la chaînes de caractères (0 pour la ligne du haut – 1 pour la ligne du bas).

### c) Défi 2 : A vous de jouer !

L'objectif est d'afficher sur l'écran LCD la température  $T$  (en degré) du processeur de la carte micro:bit. Penser à y afficher le nom de la variable ainsi que l'unité de la grandeur mesurée.

**Astuce :** Utiliser le bloc "Créer un texte avec" pour chaque ligne ainsi que le bloc "Nettoyer l'afficheur LCD"

**Aide :** utiliser tous les blocs ci-dessous en modifiant certains parfois.

Tout fonctionne ? Pour vérifier, la solution se trouve en fin du document.

3

## Récupération des données de la carte Micro:bit (10 min)



Ouvrir maintenant l'interface Makecode (choisir mode français si possible) → outils → Accélération du tracé  
Faire le tutorial « accélération du tracé » et exporter les données au format .csv et au format .txt ainsi que réaliser une capture d'écran du graphe obtenu.



micro:bit Accélération du tracé

Quitter le tutorial Microsoft

Précédent

Click on the **Show Console** button by the simulator to see a chart of the values plotted by the block over a period of time. Hover over the board in the simulator to make a force in the x dimension.

Suivant

Exporter les données en .csv ou en .txt

← Revenir en arrière

Simulateur

Afficher la console Simulateur

```

2 44
2 37
2 31
2 26
276 0

```

Solution du Défi 1 de la partie 1

Tuto\_radio\_solution

+ [Télécharger .hex] [Con]

**Au démarrage**

- fixer x à 0
- fixer y à 0

**Répéter indéfiniment**

- si bouton A a été appuyé
  - faire [radio] envoyer la chaîne " A "
- si bouton B a été appuyé
  - faire [radio] envoyer la chaîne " B "
- [radio] si une chaîne est reçue, mettre dans stringData
  - faire
    - si radioData = " A "
      - faire
        - si x = 4
          - faire
            - fixer x à 0
            - incrémenter y de 1
          - sinon
            - incrémenter x de 1
        - si y < 5
          - faire contrôler la led x x y y luminosité 3
      - sinon si radioData = " B "
        - faire
          - si y < 5
            - faire contrôler la led x x y y état HAUT

Solution du Défi 2 partie 2

unamed

+ [Télécharger .hex] [Connecter]

**Au démarrage**

- [LCD1602] afficher le texte " température " sur la ligne 0
- [LCD1602] afficher le texte créer le texte température en (°C) Celsius sur la ligne 1
- pause 500 milliseconde(s)
- [LCD1602] nettoyer l'écran

**Répéter indéfiniment**

- [LCD1602] afficher le texte " température " sur la ligne 0
- [LCD1602] afficher le texte créer le texte température en (°C) Celsius sur la ligne 1
- pause 500 milliseconde(s)
- [LCD1602] nettoyer l'écran

```

1 from microbit import *
2 from lcd_i2c import LCD1602
3
4 lcd = LCD1602()
5
6 while True:
7     lcd.setCursor(0,0)
8     lcd.writeTxt('température')
9     lcd.setCursor(0,1)
10    lcd.writeTxt(str(temperature()) + 'Celsius')
11    sleep((500)*1)
12    lcd.clear()
13
    
```