

Envoi des données depuis le microbit à Processing (**attention, pas encore fonctionnel en mode REPL**).

Connecter le Microbit à un autre programme tel que Processing via le port série. Infos tirés d'ici : <https://support.microbit.org/support/solutions/articles/19000022103-outputting-serial-data-from-the-micro-bit-to-a-computer>

- Prérequis :

- Savoir programmer le microbit
- Connaître les bases des instructions 'ligne de commande' (command-line interface) dans un terminal.

1. La communication par port série est un protocole qui permet d'échanger les données entre périphériques et ordinateurs.

2. Chargez ce programme dans votre microbit :

https://makecode.microbit.org/_3uj4xdAhhCJK

3. L'ordinateur reconnaît l'interface série du microbit et lui attribue un port lorsqu'on le branche. Pour obtenir la liste des interfaces, faites **ls /dev/cu***
Celle du microbit ressemble à ceci :

```
alx@AirdeAlexandre /dev % ls /dev/cu*  
/dev/cu.Bluetooth-Incoming-Port /dev/cu.usbmodem142402  
alx@AirdeAlexandre /dev %
```

4. Sur mac osx démarrez 'screen'. **screen / dev/cu.usbmodem142402 115200**

Attention, il faut remplacer le numéro de l'exemple qui suit «usbmodem» par le vôtre. Il se peut que ce numéro change après une session, il est bien de revérifier.

```
alx@AirdeAlexandre ~ % screen /dev/cu.usbmodem14240/ 115200
```

Le 115200 correspond à la vitesse de communication exprimée en 'bauds' (qui a été établie pour unifier deux protocoles de communication par télégraphe).

La sortie de 'screen' sur le mac :

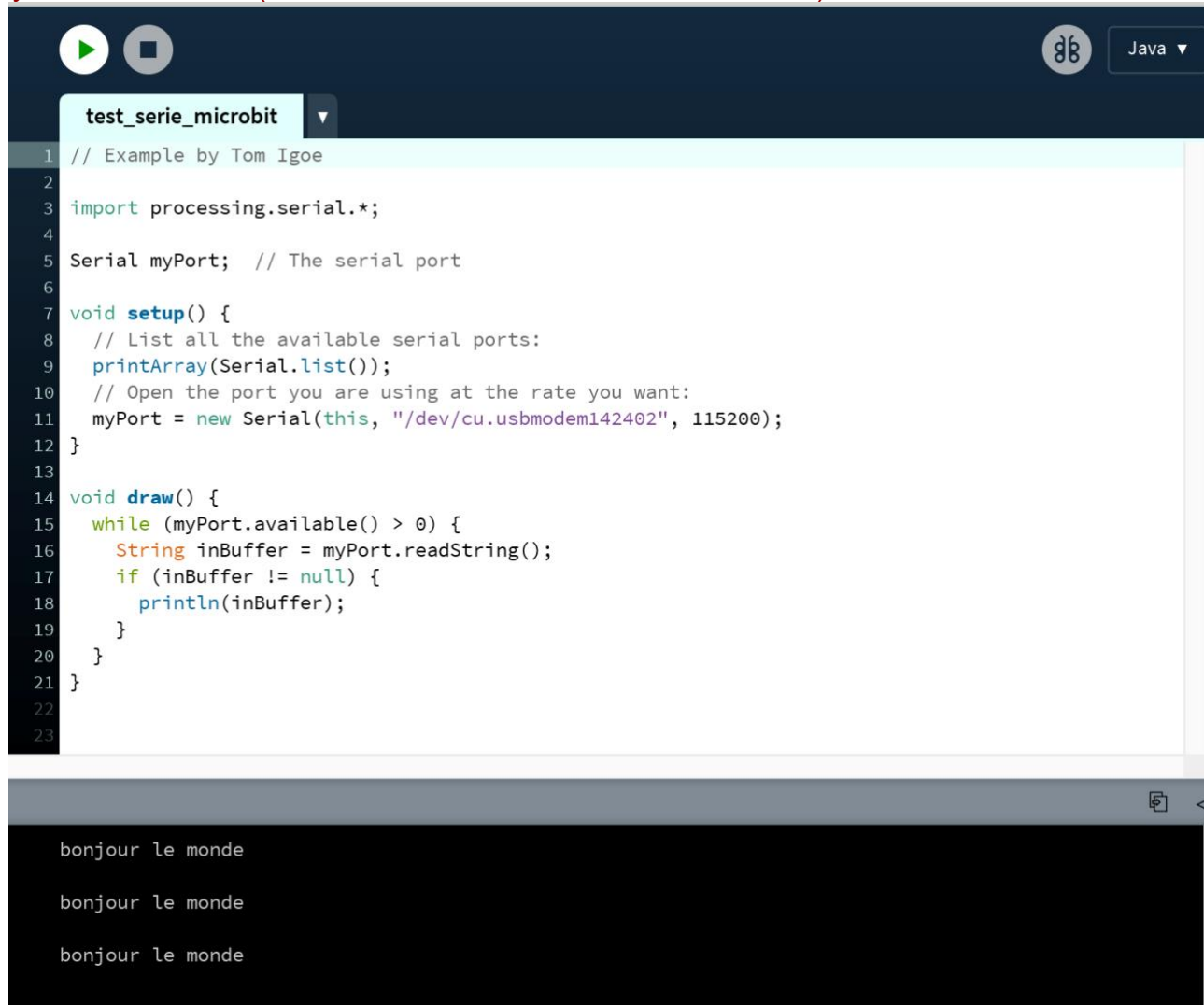
```
bonjour le monde  
bonjour le monde  
bonjour le monde  
bonjour le monde  
bonjour le monde  
bonjour le monde  
bonjour le monde  
bonjour le monde  
bonjour le monde  
bonjour le monde
```

5. Faites Ctl-A, Ctl-D pour quitter l'application 'screen'. Le port série ne sera peut-être pas

fermé donc il sera peut-être nécessaire de débrancher physiquement votre microbit pour réinitialiser la connection (à éditer avec de l'info pour fermer la communication sérieelle avec le microbit gracieusement depuis 'screen').

6. Dans Processing, copier le sketch d'exemple de Tom Igoe pour 'serial.readString()' https://processing.org/reference/libraries/serial/Serial_readString_.html. Adaptez la ligne de définition du port série comme ceci :

`myPort = new Serial(this, "/dev/cu.usbmodem142402", 115200);`



```
1 // Example by Tom Igoe
2
3 import processing.serial.*;
4
5 Serial myPort; // The serial port
6
7 void setup() {
8   // List all the available serial ports:
9   printArray(Serial.list());
10  // Open the port you are using at the rate you want:
11  myPort = new Serial(this, "/dev/cu.usbmodem142402", 115200);
12 }
13
14 void draw() {
15   while (myPort.available() > 0) {
16     String inBuffer = myPort.readString();
17     if (inBuffer != null) {
18       println(inBuffer);
19     }
20   }
21 }
22
23
```

bonjour le monde

bonjour le monde

bonjour le monde

8. Pour l'envoi de données de type numériques et non de type de chaînes de caractères, remplacez la fonction readString() par la fonction read(). La documentation est ici : https://processing.org/reference/libraries/serial/Serial_read_.html

N.B. Processing en mode 'REPL' retourne une erreur, la communication sérieelle n'est peut-être pas adaptée. D'autres tests doivent être faits.

Un exemple d'envois de données entre Micro:bit et Processing :

Lorsque Processing reçoit un feedback de données numériques de la part du Micro:bit, un carré vert apparaît, sinon un carré rouge.

```
# =====
# AUTHOR :TRISTAN FRANC
#OGR : COLLEGE LIONEL-GROULX
#DATE:12_05_2020
#OPEN_SOURCE
# =====

void setup() {
  size (640, 360);
  // List all the available serial ports:
  //affiche la liste de ports disponibles
  printArray(Serial.list());
  //Ouverture du port série
  myPort = new Serial(this, "/dev/tty.usbmodem14302", 115200);
}

void draw() {
  background (0);
  //vérifie que le port recois une donnée
  while (myPort.available() > 0) {
    //stockage reçu sur le port série
    String x = myPort.readString(); //myPort.readString(); = Micro:bit que je définis par « x »
    /*vérifie que la valeur lue n'est pas n « NULL»(absence de donnée , incluant 0)
    //le « ! » est l'équivalent de dire n'est pas*/
    if (x!=null ) { //si «x», le Micro:bit, donne un feedback
      switch(x)
      case"1": //si x = 1, affiche un carré vert
        fill (0, 255, 0);
        rect (100, 100, 100, 100);
        break;
      case"0": // si x = 0, affiche un carré rouge
        fill (255,0,0);
        rect (100,100,100,100);
        break;
      }
    }
    println(x); //feedback de x sur le moniteur
    x="";
  }
}
```