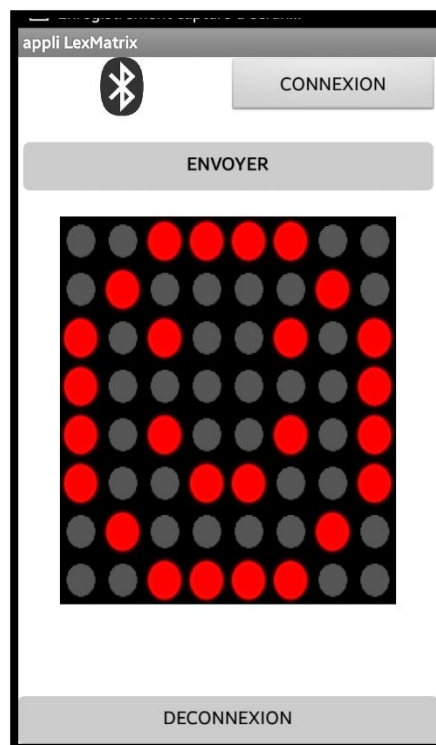
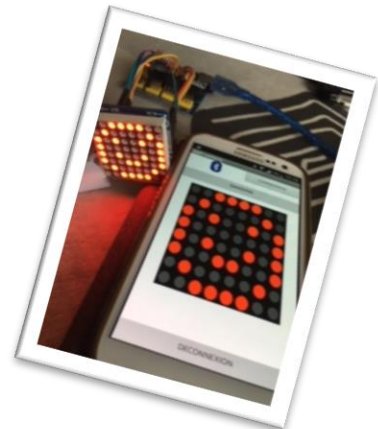


# DESSINER SUR UNE MATRICE LED AVEC SON TÉLÉPHONE



- Matrice Led 8x8  
MAX7219/MAX7221
- Module Bluetooth

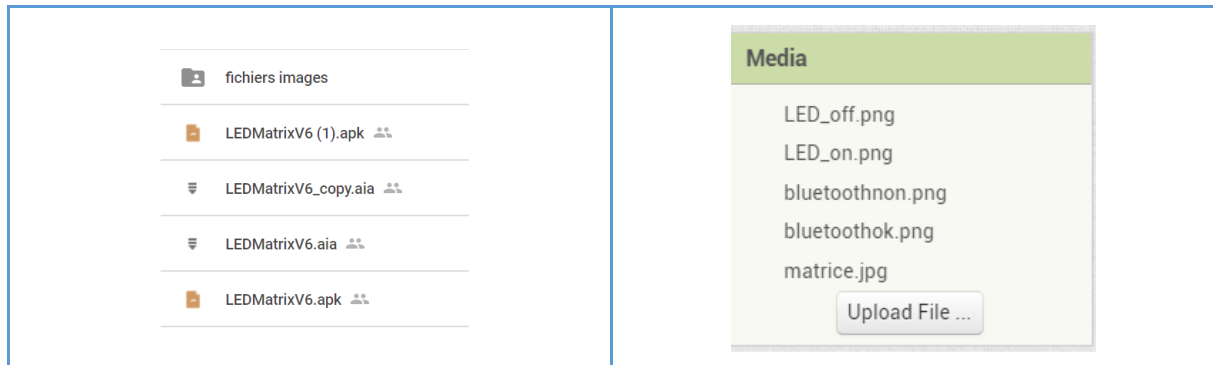


Ce tuto a été écrit par Le groupe Robot en 2019 adhérents de Makerspace56. Il est mis à disposition selon les termes de la licence Creative Commons Attribution – Pas d'Utilisation Commerciale. Les autorisations au-delà du champ de cette licence peuvent être obtenues via <http://makerspace56.org/contact/>.

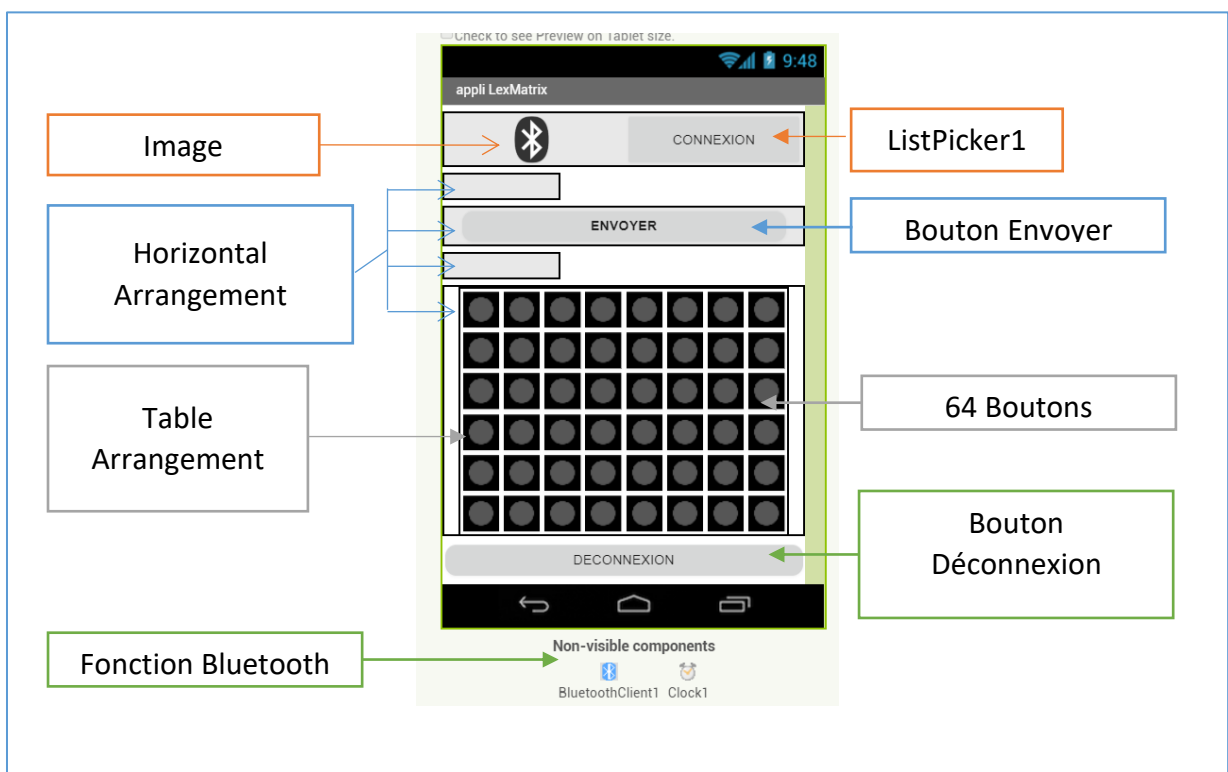


# PARTIE APP INVENTOR

## A. Fichier téléchargeables sur le Drive



## B. Interface DESIGNER



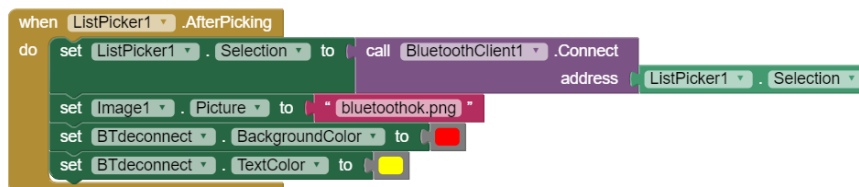
## C. Interface BLOCK

### 1. ListPicker – connexion Bluetooth

- ListPicker permet d'afficher une liste. Ici, on choisit de rechercher parmi les connexions bluetooth disponibles sur le téléphone.
- On définit une couleur de police et d'arrière-plan du bouton.

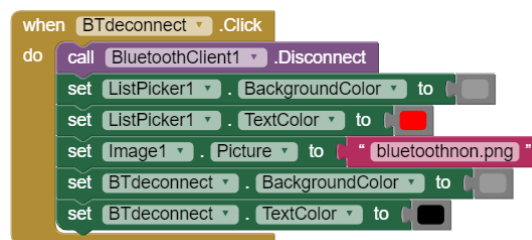


- Lorsque la liste est choisie, on se connecte à une adresse et on définit les paramètres du bouton « Connexion », « Déconnexion » ainsi que l'icône signifiant la bonne liaison.



### 2. Déconnexion

- Lorsque le bouton « Déconnexion » est appuyé, on coupe la liaison Bluetooth et on modifie l'interface.



### 3. Variables

- Il y a 8 lignes (indexées de 0 ..7) et 8 colonnes (également indexées de 0 ..7) dans la matrice.
- Dans Arduino, la librairie LedControl permet de contrôler toute les leds d'une ligne grâce à l'instruction « setRow(addr,row,value) »
  - Addr : numéro de la matrice (si plusieurs) ;
  - Row : numéro de la ligne ;
  - Value : valeur de la ligne en hexa, binaire ou décimal.

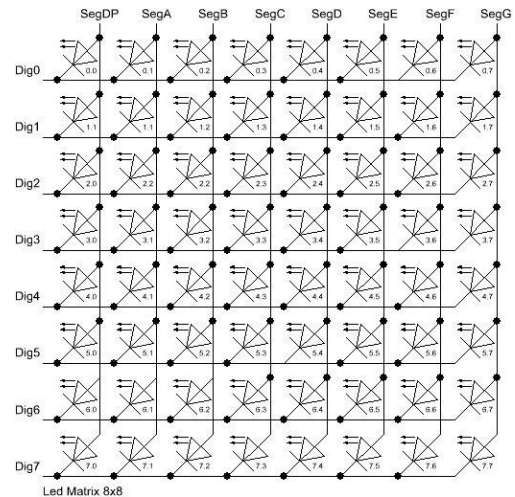
La fonction utilisée Arduino :maMatrice.setRow(0, lg, valLigne)

valLigne est en décimal en base 2.

8 séries de valeurs seront envoyées correspondant à chaque ligne

lg/col	Led 2.0	Led 2.1	Led 2.2	Led 2.3	Led 2.4	Led 2.5	Led 2.6	Led 2.7
Valeur	128	64	32	16	8	4	2	1

- Une ligne vide aura une valeur de 0
- Une ligne entière aura une valeur de 255.
- Chaque Led se verra attribuée une valeur de 1 si elle est allumée et 0 si elle est éteinte.
- Les numéros de colonne ont un facteur de 1 à 128.  
Il faut 8 variables r1, ..., r8 pour les rangées et 64 variables n1,...,n64 pour chaque led.



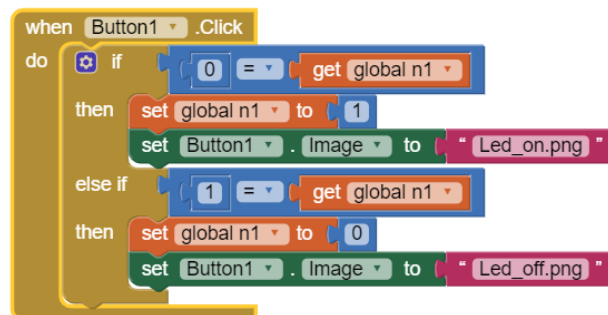
- Chaque variable est initialisée à 0

initialize global r1 to 0

initialize global n25 to 0

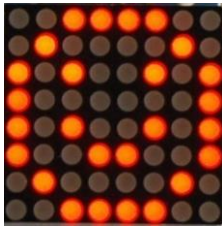
#### 4. Déclaration de chaque bouton Led

- Quand le bouton de la led est actionné, on lui affecte la valeur 1 et on modifie son apparence. Si ce même bouton est appuyé de nouveau, sa valeur devient zéro et il retrouve son aspect éteint.



#### 5. Calcul de la valeur des lignes quand le bouton « envoyer » est appuyé





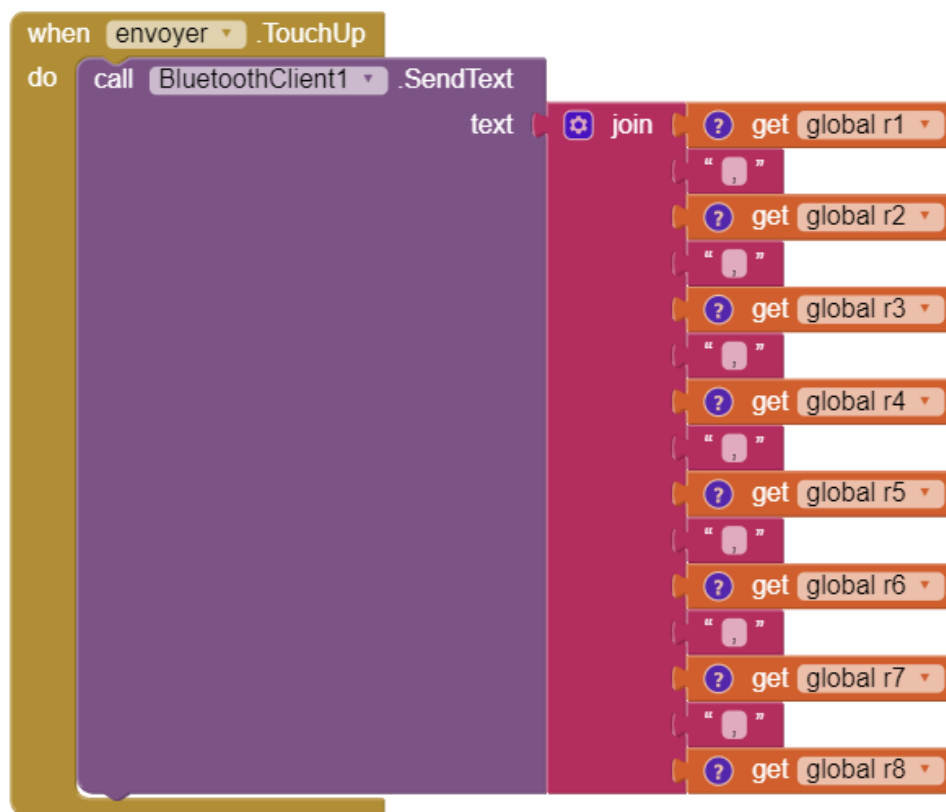
### Exemple :

Global r1 =  $(128 \times 0) + (64 \times 0) + (32 \times 1) + (16 \times 1) + (8 \times 1) + (4 \times 1) + (2 \times 0) + (1 \times 0) = 60$   
 Global r2 =  $(128 \times 0) + (64 \times 1) + (32 \times 0) + (16 \times 0) + (8 \times 0) + (4 \times 0) + (2 \times 1) + (1 \times 0) = 66$   
 Global r3 =  $(128 \times 1) + (64 \times 0) + (32 \times 1) + (16 \times 0) + (8 \times 0) + (4 \times 1) + (2 \times 0) + (1 \times 1) = 165$   
 Global r4 =  $(128 \times 1) + (64 \times 0) + (32 \times 0) + (16 \times 0) + (8 \times 0) + (4 \times 0) + (2 \times 0) + (1 \times 1) = 129$

...

## 6. Envoi de la liste

- Quand le bouton « envoyer » est relâché, on envoie un message sous format texte. Ce message regroupe les 8 valeurs de lignes séparées par une virgule.



### Exemple :

Message envoyé : « 60,66,165,129,...,...,... »

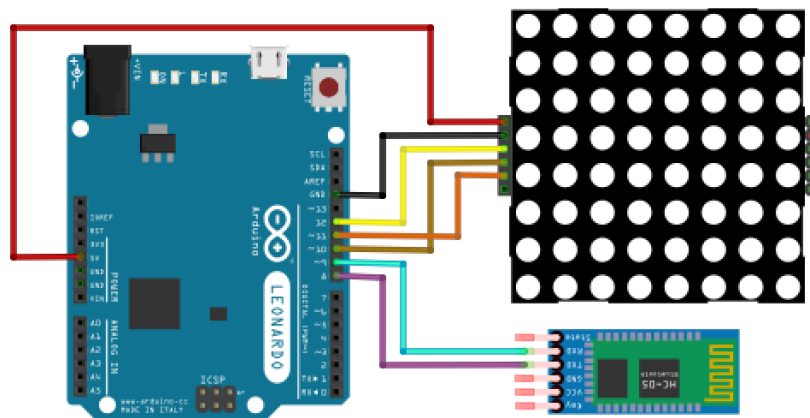
# PARTIE ARDUINO

## A. Montage

- Utilisation de la librairie <LedControl.h>

Elle permet des contrôles simples et des connections : LedControl(DIN, CLK, CS, 1);

- Pin 10 CS : Chip Select pour l'activation du modèle
- Pin 11 CLK : l'horloge synchrone pour les données. L'horloge définit la vitesse de transmission des données série
- Pin 12 DIN : La donnée série sur 8 bits
- Le dernier paramètre correspond au nombre de MAX7219 utilisé. Ici un (1).



```
// Définition de la matrice de led
```

```
#include <LedControl.h>
```

```
LedControl maMatrice = LedControl(12,11,10,1);
```

## B. Bluetooth

- Utilisation de la librairie <SoftwareSerial.h> pour la connexion Bluetooth

```
// Définition d'une liaison série BlueTooth
```

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial monBT(8,9);
```

```
void setup()
```

```
{
```

```
  // initialisation du BlueTooth
```

```
  monBT.begin(9600);
```

## C. Les fonctions de bases :

- La fonction shutdown();

Elle permet d'allumer ou d'éteindre le MAX 7219. Si la valeur est false, le circuit fonctionne normalement et à l'inverse, true le circuit est éteint.

- La fonction setIntensity();

Elle permet de définir l'intensité lumineuse de l'afficheur en 16 valeurs différentes; 15 étant notamment le maximum.

- La fonction clearDisplay();

Elle éteint l'afficheur et supprime toutes les données précédentes.

```
// Initialisation de la matrice
maMatrice.shutdown(0, false); // réveil !
maMatrice.setIntensity(0, 8); // intensité
maMatrice.clearDisplay(0); // effacement
}
```

## D. Le LOOP

```
void loop()
{
    int valLigne = 0; // valeur reçue pour une ligne de matrice

    // Lecture des données BlueTooth
    while(monBT.available() > 0)
    {
        // Boucle pour lire les 8 valeurs
        for (int lg = 0; lg < 8; lg++){
            valLigne = monBT.parseInt(); // recherche une variable entière jusqu'à un séparateur (,)
            maMatrice.setRow(0, lg, valLigne); // affiche la valeur trouvée sur la ligne correspondante de la matrice
        }

        // Purge du buffer série (lecture de la fin de chaine pour rien !)
        if (monBT.readString()){/* Rien ! */}
    }
}
```