#### Vue d'ensemble des blocs :

```
bal R1 to 1 100e3 initialize global R3 to 1 100e3
                                                               ze global 🗶 to 【 20】
                               fize global R4 to 10e3
        al R2 to ( 515
                                                                    al X before to 20 init
                                1023 / ( get global Y • × ( 4
                                                                                                                                                              get global R1 • get global R4 •
                                                                                                                get global Y • × (4 / 1023
                             get global R3 * / | get global R2 *
                                                                                                                        et global X * to O get global X * * 5

et global Y * to Call BluetoothClient1 * Receive
set global Running • to true •
pause_button • .Click
set global Running • to false •
                                                                                                                             then set R value . Text . to call calcul Rg .
  global Running • to (
  global X_before * to 20
                                                                                                                                                              10
```

# Initialisation des variables globales :

```
initialize global R1 to 1 100e3 initialize global R3 to 1 100e3 initialize global X to 1 20 initialize global Y to 1 0 initialize global VCC to 1 5 initialize global R2 to 1 515 initialize global R4 to 1 10e3 initialize global X_before to 1 20 initialize global Y_before to 1 0 initialize global Running to 1 false v initialize global R6 to 1 33000
```

Running est un booléen qui permet de connaître l'état de l'application.

## Définition des deux modes de calcul de la résistance :

```
© to calculate

O to calculate
```

Rf correspond à la résistance du *flex sensor* commercial et Rg à celle du capteur graphite.

#### Gestion du client Bluetooth :

```
when ListPicker1 v . Elements v to | BluetoothClient1 v . AddressesAndNames v

when ListPicker1 v . AfterPicking

do set ListPicker1 v . Selection v to | call BluetoothClient1 v . Connect

address | ListPicker1 v . Selection v
```

Le premier bloc définit les éléments du *listpicker* avec les noms des appareils Bluetooth qui on déjà été appairés au téléphone auparavant. Le second bloc connecte le téléphone à l'appareil sélectionné par l'utilisateur.

## Définition des évènements associés à chaque bouton :

```
when start_button v .Click
do set global Running v to true v

when pause_button v .Click
do set global Running v to false v

when refresh_button v .Click
do set global Running v to false v

call Canvas1 v .Clear
set global X_before v to 20
set global X v to 20
```

- •Quand on clique sur *start*, la variable global *Running* passe à vrai donc l'application est en mode *running*. On utilisera cette information pour faire tourner la boucle principale.
- •Quand on clique sur *pause*, on suspend l'exécution de la boucle principale.
- •Quand on clique sur *refresh*, on suspend l'exécution de la boucle principale et on réinitialise le graphique.

## Définition des modes de l'application :

```
when ModePicker . AfterPicking

do if compare texts ( ModePicker . Selection . Selection . Selection . Selection . Selection . Send1ByteNumber number ( 1 )

else call BluetoothClient1 . Send1ByteNumber number ( 2 )
```

L'utilisateur peut choisir entre le mode *capteur graphite* et le mode *capteur commercial* via un *listpicker*. Selon le mode sélectionné, le module Bluetooth envoie la valeur 1 ou 2 à la carte Arduino. Dans le code Arduino, lorsqu'on est dans le menu *Application*, on lit le mode demandé (1 → graphite, 2 → commercial) et en fonction, on envoie la valeur de tension associée au capteur graphite ou au capteur commercial.

## **Boucle principale:**

```
when Clock1 - .Timer
              get global Running -
    if 😝
          BluetoothClient1 - IsConnected -
                       call BluetoothClient1 - .BytesAvailableToReceive > 10
                      set global X_before - to
                                               get global X
                       set global Y_before - to
                                               get global Y
                       set global X to to get global X
                                                                 5
                       set global Y • to call BluetoothClient1 • .ReceiveUnsigned1ByteNumber
                                  get global Y - > 0
                                  compare texts (ModePicker -
                                                                    Selection -
                                                                                       Capteur graphite
                                                  Text • to call calcul_Rg •
                                  set R_value •
                                                 . Text • to call calcul_Rf •
                             set R_value •
                                            Text - to
                call File1 . AppendToFile
                                                     get global X
                                           🧔 join |
                                                     get global Y
                                 fileName
                                           /saved_file.txt
          call Canvas1 . DrawLine
                                     get global X_before
                                x1
                                y1
                                       get global Y_before
                                х2
                                     get global X
                                y2
                                                         10
                                       get global Y
                                    ≥ • Canvas1 •
                                                       Width -
                      get global X *
          then call Canvas1 .Clear
                 set global X_before to 20
                 set global X to (20)
```

Pour séquencer le programme, on utilise un *timer*. Toutes les 100 ms, on regarde si l'application est en mode *Running*: si ce n'est pas le cas, on ne fait rien. A l'inverse, si on est en mode *Running*, on vérifie la connexion au module Bluetooth, on lit la valeur de tension qu'il nous envoie et on met à jour nos variables globales. Selon le mode choisi par l'utilisateur, on utilise la fonction adaptée pour le calcul de résistance. Cette valeur de résistance est affichée sur l'écran puis les coordonnées (x,y) sont sauvegardées dans un fichier. Enfin, on trace sur le graphique une ligne entre les coordonnées précédentes et les nouvelles. Si on est hors intervalle, on rafraîchit le graphique et on réinitialise x.