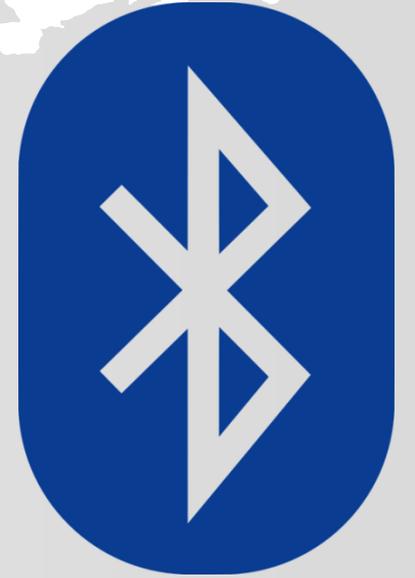


# App Inventor

## Communiquer par Bluetooth



(entre 2 smartphones)

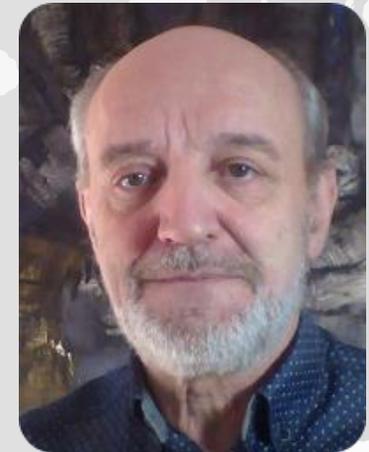
# PLAN DU TUTORIEL



- Site du cours : <https://onvaessayer.org/appinventor?app=bluetooth>
- Video : <https://onvaessayer.org/appinventor?video=bluetooth>

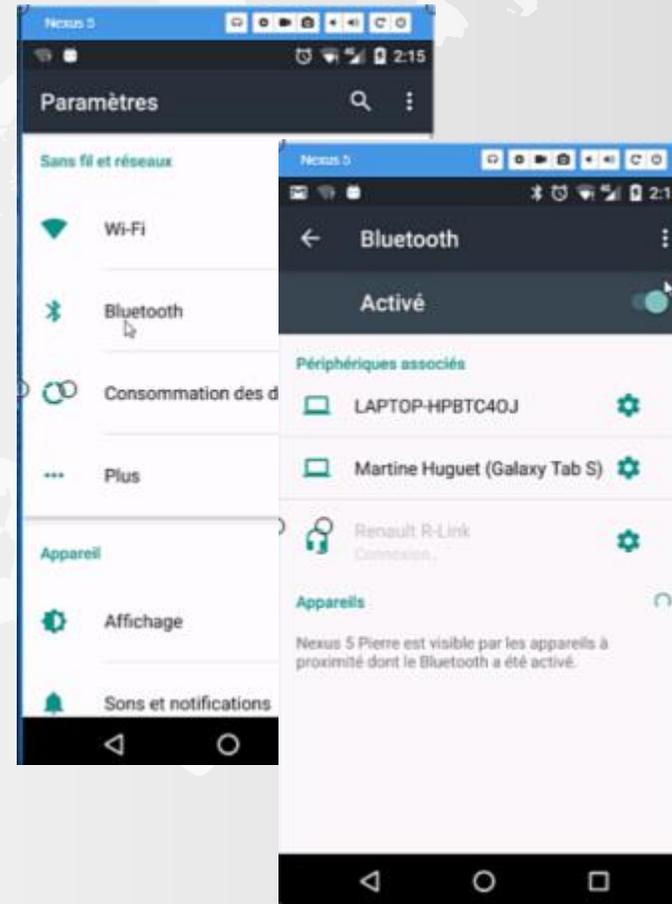
## Plan

1. Appairage de deux appareils Bluetooth
2. Design de l'application de test
3. Programmation
4. Essais



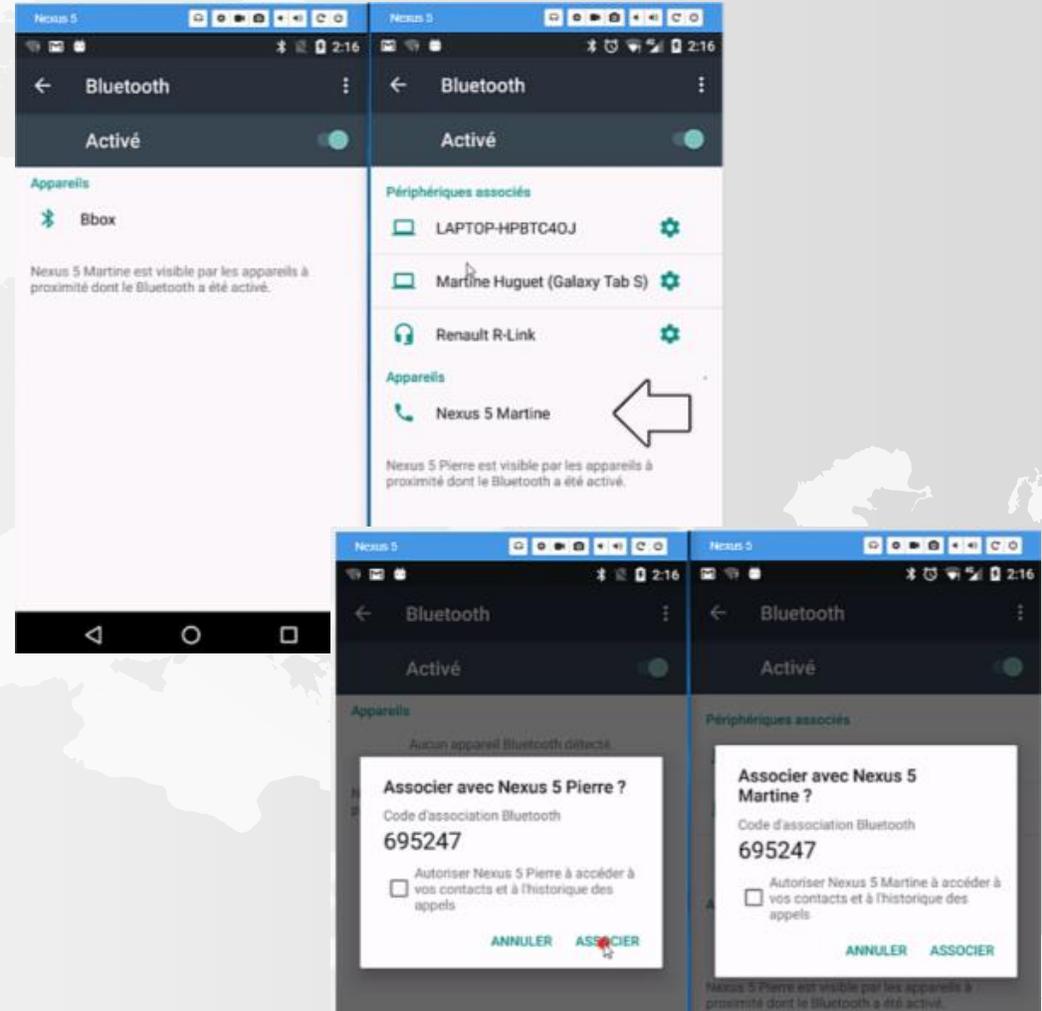
# APPAIRAGE-ASSOCIATION DES SMARTPHONES

- activation du bluetooth  
on voit la liste des appareils BT
  - déjà associés (ou appairés)
  - ou seulement détectés



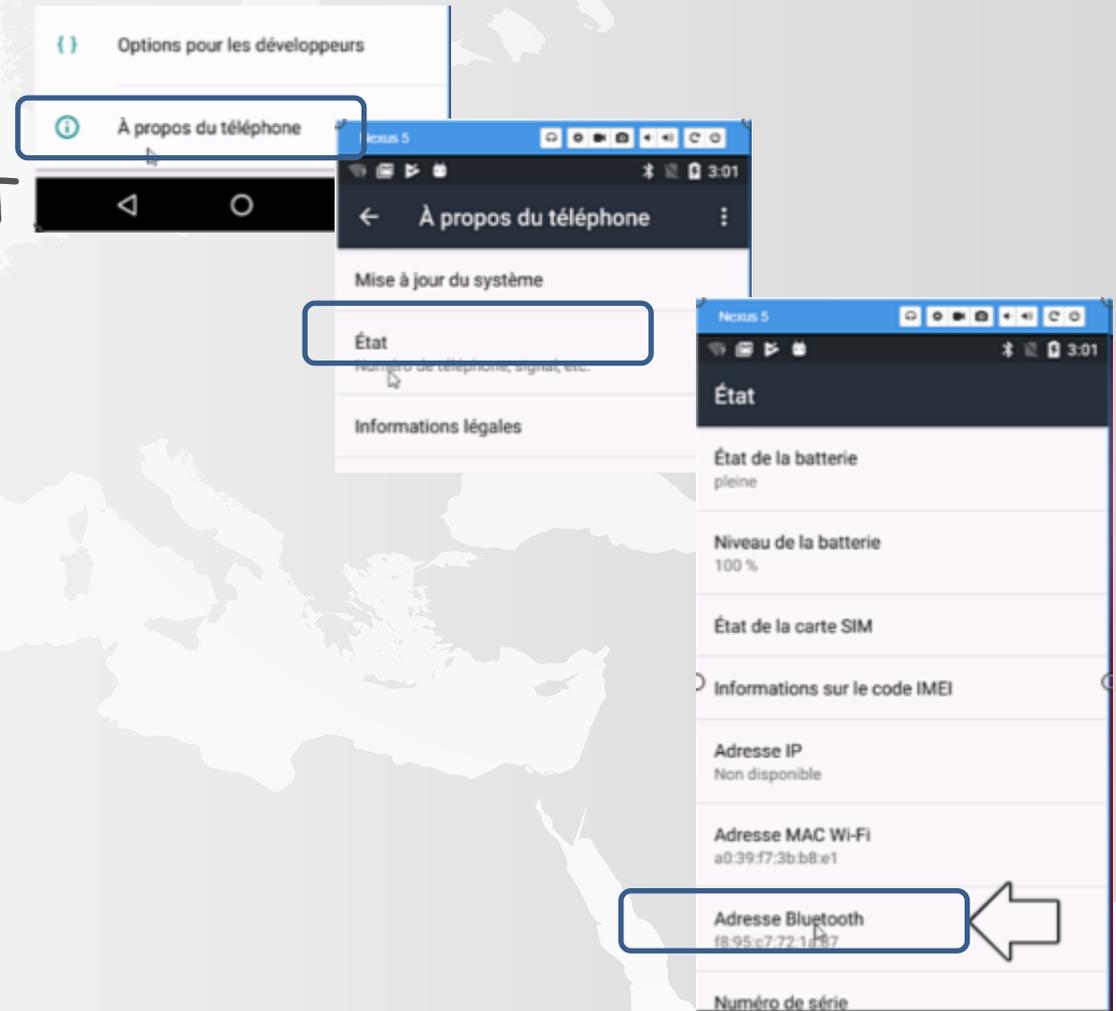
# APPAIRAGE-ASSOCIATION DES SMARTPHONES

- activation du bluetooth  
on voit la liste des appareils BT
  - déjà associés (ou appairés)
  - ou seulement détectés
- Après activation BT des deux appareils
  - Clic sur un appareil détecté
  - Confirmation de l'association



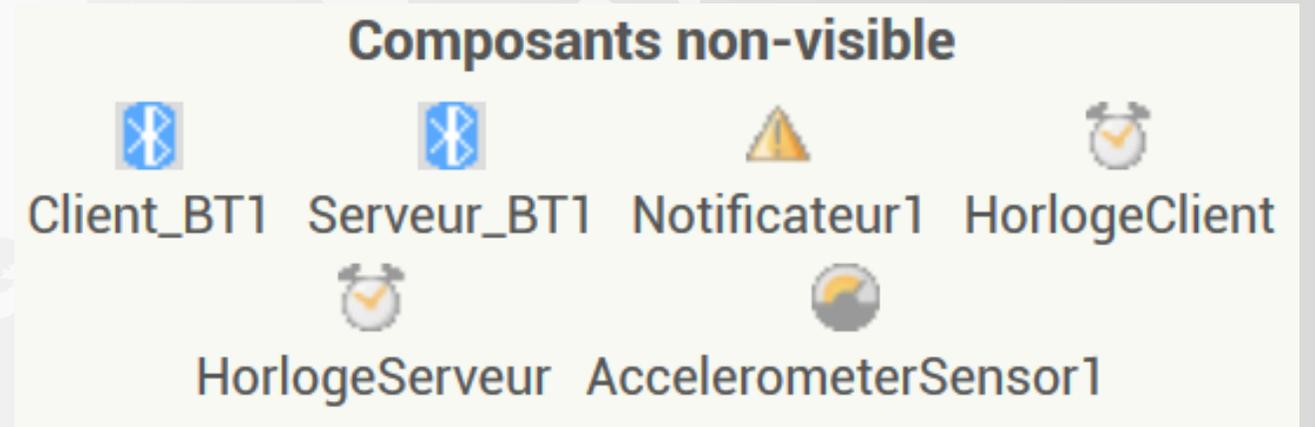
# APPAIRAGE-ASSOCIATION DES SMARTPHONES

- activation du bluetooth  
on voit la liste des appareils BT
  - déjà associés (ou appairés)
  - ou seulement détectés
- Après activation BT  
des deux appareils
  - Clic sur un appareil détecté
  - Confirmation de l'association
- Pour info :
  - Accès à l'adresse BT  
d'un smartphone



# CODAGE DU PROGRAMME : DESIGN

## 1. Composants non visibles



# CODAGE DE L'APPLI : DESIGN

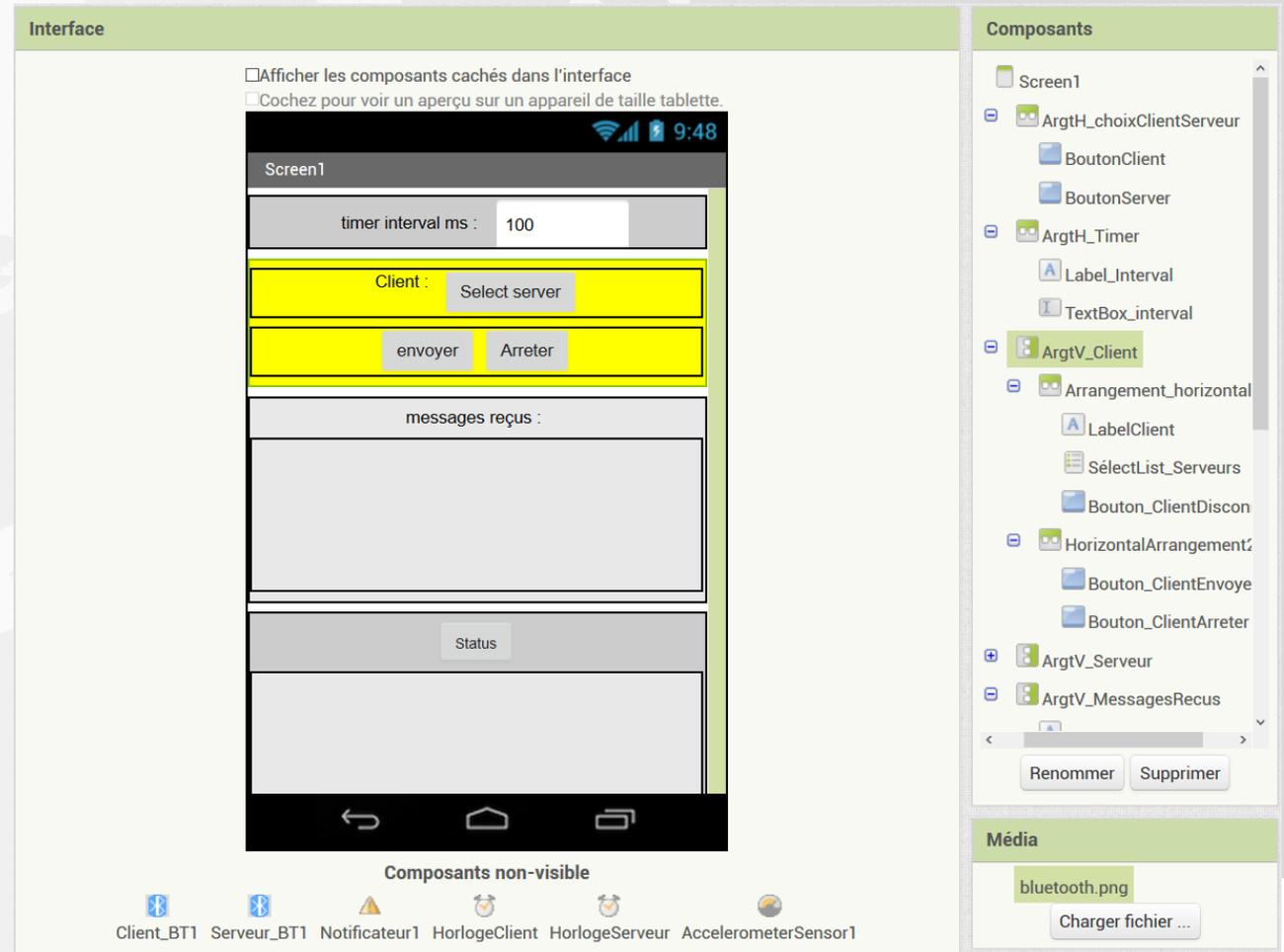
1. Composants non visibles
2. Choix du mode client/serveur
3. Cadence d'envoi
4. Affichage des messages reçus
5. Affichage statut et erreurs

The screenshot displays the design tool interface for an Android application. The main window, titled "Interface", shows a preview of the application screen. At the top, there are two checkboxes: "Afficher les composants cachés dans l'interface" and "Cochez pour voir un aperçu sur un appareil de taille tablette". The preview shows a screen with a status bar at the top displaying "9:48" and signal icons. Below the status bar, there are two buttons labeled "client" and "Serveur". A text input field is labeled "timer interval ms : 100". Below this is a label "messages reçus :" followed by a large empty text area. At the bottom of the preview is a label "Status" followed by another large empty text area. The bottom of the preview shows the Android navigation bar with back, home, and recent apps icons. Below the preview, there is a section titled "Composants non-visible" with icons for "Client\_BT1", "Serveur\_BT1", "Notificateur1", "HorlogeClient", "HorlogeServeur", and "AccelerometerSensor1". To the right of the main window is a "Composants" panel showing a tree view of the application's components. The root is "Screen1", which contains "ArgtH\_choixClientServeur" (with sub-components "BoutonClient" and "BoutonServer"), "ArgtH\_Timer" (with "Label\_Interval" and "TextBox\_interval"), "ArgtV\_Client", "ArgtV\_Serveur", "ArgtV\_MessagesRecus" (with "LabelMessagesRecus"), "ArgtVScroll\_TexteRecu" (with "LblRecu"), "ArgtV\_statusErr" (with "Bouton\_Status"), and "ArgtVScroll\_Status" (with "Lbl\_statusMess"). At the bottom of the "Composants" panel are "Renommer" and "Supprimer" buttons. Below the "Composants" panel is a "Média" section with a file named "bluetooth.png" and a "Charger fichier ..." button.



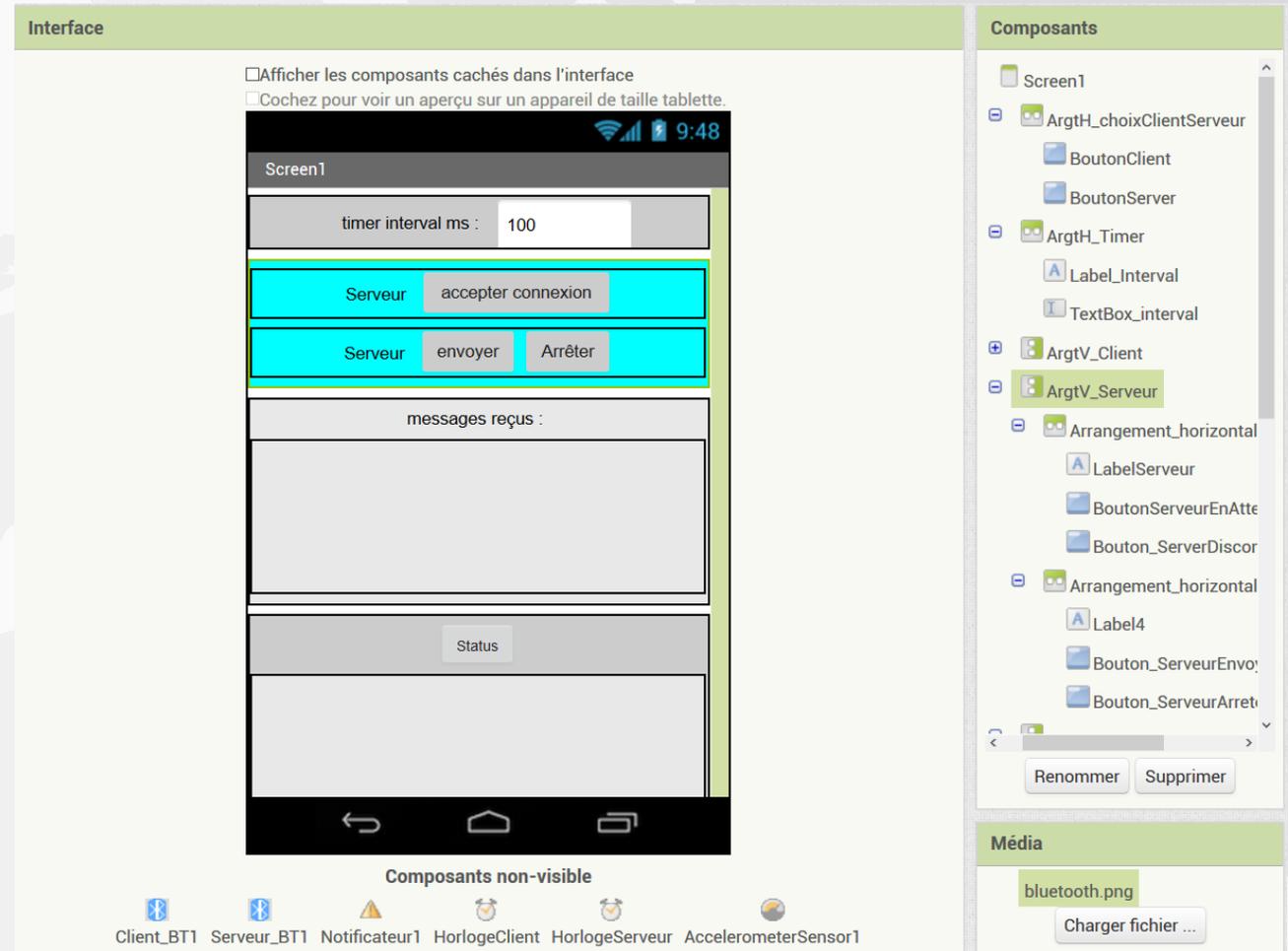
# CODAGE DE L'APPLI : DESIGN

1. Composants non visibles
2. Choix du mode client/serveur
3. Cadence d'envoi
4. Affichage des messages reçus
5. Affichage statut et erreurs
6. Interface client :
  - Choix du serveur, début et fin envoi



# CODAGE DE L'APPLI : DESIGN

1. Composants non visibles
2. Choix du mode client/serveur
3. Cadence d'envoi
4. Affichage des messages reçus
5. Affichage statut et erreurs
6. Interface client :
  - Choix du serveur, début et fin envoi
7. Interface serveur
  - Attente connexion, début et fin envoi



# CODAGE DE L'APPLI : BLOCS

```
initialise global messageNumber à 0
initialise global envoiEnCours à faux

quand BoutonClient .Clic
faire
  si pas Client_BT1 .Activé
  alors
    appeler Notificateur1 .Afficher Alerte notice " BT client inactif! "
  sinon
    mettre Argv_Client .Visible à vrai
    mettre Argv_Serveur .Visible à faux
    mettre ArgvH_choixClientServeur .Visible à faux

quand BoutonServer .Clic
faire
  si pas Serveur_BT1 .Activé
  alors
    appeler Notificateur1 .Afficher Alerte notice " BT Serveur inactif! "
  sinon
    mettre Argv_Client .Visible à faux
    mettre Argv_Serveur .Visible à vrai
    mettre ArgvH_choixClientServeur .Visible à faux
```



# CODAGE DE L'APPLI : BLOCS

```
quand SélectList_Serveurs . Avant prise
faire
  mettre SélectList_Serveurs . Éléments à Client_BT1 . Adresses et noms

quand SélectList_Serveurs . Après prise
faire
  mettre Lbl_statusMess . Texte à " "
  si
    appeler Client_BT1 . Se connecter adresse SélectList_Serveurs . Sélection
  alors
    appeler Client_BT1 . Envoyer texte texte " client connecté\n "
    mettre SélectList_Serveurs . Visible à faux
    mettre Bouton_ClientDisconnect . Visible à vrai
    mettre HorlogeClient . ChronomètreActivé à vrai
    mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte
    " connexion réussie "
    "\n "
  sinon
    mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte
    " échec connexion "
    "\n "
```

```
quand BoutonServeurEnAttenteConnexion . Clic
faire
  appeler Serveur_BT1 . Accepter connexion Nom service " "
  mettre BoutonServeurEnAttenteConnexion . Couleur de fond à
  mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte
  " acceptation démarrée "
  "\n "
```

```
quand Serveur_BT1 . Connexion acceptée
faire
  appeler Serveur_BT1 . Envoyer texte texte " connexion acceptée par le serveur\n "
  mettre BoutonServeurEnAttenteConnexion . Couleur de fond à
  mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte
  " connexion acceptée "
  "\n "
  mettre HorlogeServeur . ChronomètreActivé à vrai
  mettre BoutonServeurEnAttenteConnexion . Visible à faux
  mettre Bouton_ServerDisconnect . Visible à vrai
```



# CODAGE DE L'APPLI : BLOCS

```
quand Bouton_ClientDisconnect .Clic
faire
appeler Client_BT1 .Envoyer texte texte " client Déconnecté\n "
appeler Client_BT1 .Déconnecter
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte
" déconnexion client "
" \n "
mettre SelectList_Serveurs . Visible à vrai
mettre Bouton_ClientDisconnect . Visible à faux
mettre HorlogeClient . ChronomètreActivé à faux
```

```
quand Bouton_ServerDisconnect .Clic
faire
appeler Serveur_BT1 .Envoyer texte texte " serveur Déconnecté\n "
appeler Serveur_BT1 .Déconnecter
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte
" déconnexion serveur "
" \n "
mettre BoutonServeurEnAttenteConnexion . Visible à vrai
mettre Bouton_ServerDisconnect . Visible à faux
mettre HorlogeServeur . ChronomètreActivé à faux
```



# CODAGE DE L'APPLI : BLOCS

The image displays three sets of Scratch-style code blocks for an Android application:

- Client Send Button:** When `Bouton_ClientEnvoyer` is clicked, set `HorlogeClient` interval timer to `TextBox_interval` text, set `global messageNumber` to 0, and set `global envoiEnCours` to true.
- Client Stop Button:** When `Bouton_ClientArreter` is clicked, set `global envoiEnCours` to false.
- Client Timer:** When `HorlogeClient` timer ticks, if `global envoiEnCours` is true, increment `global messageNumber` by 1 and call `Client_BT1` to send the `messageAenvoyer` text. If `Client_BT1` is connected and has more than 0 bytes available for reception, call `afficher` to receive text from `Client_BT1` and call `Client_BT1` to receive more bytes.
- Server Send Button:** When `Bouton_ServeurEnvoyer` is clicked, set `HorlogeServeur` interval timer to `TextBox_interval` text, set `global messageNumber` to 0, and set `global envoiEnCours` to true.
- Server Stop Button:** When `Bouton_ServeurArreter` is clicked, set `global envoiEnCours` to false.
- Server Timer:** When `HorlogeServeur` timer ticks, if `global envoiEnCours` is true, increment `global messageNumber` by 1 and call `Serveur_BT1` to send the `messageAenvoyer` text. If `Serveur_BT1` is connected and has more than 0 bytes available for reception, call `afficher` to receive text from `Serveur_BT1` and call `Serveur_BT1` to receive more bytes.



# CODAGE DE L'APPLI : BLOCS

The screenshot displays two event listener code blocks in an Android Studio environment. At the top, there are two purple blocks: 'recu appeler Client\_BT1 .Recevoir texte nombre d'octets' and 'appeler Client\_BT1 .Octets disponibles pour le réception'. Below these, there are two main event listener blocks:

- Left Block (Client\_BT1):** Triggered by 'quand Client\_BT1 BluetoothError'. It contains a 'faire' block with 'mettre Lbl\_statusMess Texte à' followed by a series of 'joint' (join) blocks. The first 'joint' block connects to 'Lbl\_statusMess Texte' with the value 'BT Client Error:'. The second 'joint' block connects to a 'fonction:' block with the value '\n'. The third 'joint' block connects to 'obtenir Nom fonction' with the value '\n'. The fourth 'joint' block connects to 'obtenir message' with the value '\n'.
- Right Block (Serveur\_BT1):** Triggered by 'quand Serveur\_BT1 BluetoothError'. It follows the same structure as the left block but with the value 'BT Server Error:' in the first 'joint' block.



# CODAGE DE L'APPLI : BLOCS

```
à messageAenvoyer
résultat joint
  obtenir global messageNumber
  " "
  modulo de appeler HorlogeClient .Obtenir millisecondes instant appeler HorlogeClient .Maintenant
  u00F7 100000
  " "
  AccelerometerSensor1 . XAccel
  " "
  AccelerometerSensor1 . YAccel
  " "
  AccelerometerSensor1 . ZAccel
  "\n"

à afficher recu
faire mettre LblRecu . Texte à joint obtenir recu
  LblRecu . Texte
```



# CODAGE DE L'APPLI : BLOCS

```
quand Bouton_Status .Clic
faire
mettre Lbl_statusMess . Texte à " Client : \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " activé : " Client_BT1 . Activé " \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " disponible : " Client_BT1 . Disponible " \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " connecté : " Client_BT1 . Est connecté " \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " adresses : \n " liste vers ligne csv liste Client_BT1 . Adresses et noms " \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " \n " Serveur : \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " activé : " Serveur_BT1 . Activé " \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " disponible : " Serveur_BT1 . Disponible " \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " connecté : " Serveur_BT1 . Est connecté " \n "
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " accepte : " Serveur_BT1 . Accepte " \n "

quand Screen1 .Erreur survenue
Composant Nom fonction Numéro erreur message
faire
mettre Lbl_statusMess . Texte à joint Lbl_statusMess . Texte " Error : " \n "
joint " component : " obtenir Composant " \n "
joint " fonction : " obtenir Nom fonction " \n "
joint " error number : " obtenir Numéro erreur " \n "
joint " message : " obtenir message " \n "
```





# CODAGE DE L'APPLI : ESSAIS

- Lancement de l'application sur deux smartphones
- 1° smartphone en serveur  
2° smartphone en client
- 1° : accepter connexion  
2° : sélection serveur  
(Nexus 5 Martine)

Connexion validée, messages échangés

- Envoyer depuis le serveur : OK
- Envoyer depuis le client : OK

