



Parcours : 2SN-T

Module : Programmation de mobiles

Projet : Communication Bluetooth

Outils : Android Studio

Auteur : RHAYOUTE Abdelmalek, Aymane Rbaati

Date : 03 juin 2021

Introduction : L'objectif principal de notre projet est de faire en sorte que deux smartphones Android communiquent bien afin qu'ils soient capables d'échanger des informations via la communication Bluetooth.

Pour créer une connexion entre deux appareils, nous avons mis en œuvre les mécanismes côté serveur et côté client, car un appareil doit ouvrir un socket serveur et l'autre doit initier la connexion en utilisant l'adresse MAC de l'appareil serveur. Le périphérique serveur et le périphérique client obtiennent chacun le BluetoothSocket requis de différentes manières. Le serveur reçoit des informations sur le socket lorsqu'une connexion entrante est acceptée. Le client fournit des informations sur le socket lorsqu'il ouvre un canal RFCOMM vers le serveur. Le serveur et le client sont considérés comme connectés l'un à l'autre lorsqu'ils ont chacun un BluetoothSocket connecté sur le même canal RFCOMM. À ce stade, chaque dispositif peut obtenir des flux d'entrée et de sortie, et le transfert de données peut commencer.

Se connecter en tant que serveur

Pour connecter deux appareils, l'un d'eux doit agir comme un serveur en tenant un BluetoothServerSocket ouvert. L'objectif du socket serveur est d'écouter les demandes de connexion entrantes et de fournir un BluetoothSocket connecté après l'acceptation d'une demande. Lorsque le BluetoothSocket est acquis à partir du BluetoothServerSocket, le BluetoothServerSocket peut - et doit - être jeté.

Se connecter en tant que client

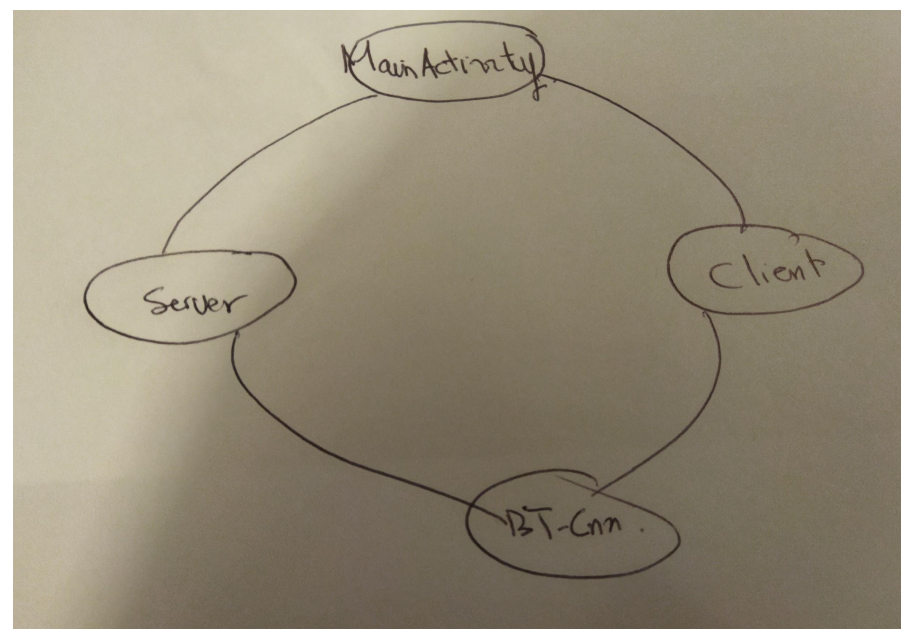
Afin d'initier la connexion, on doit obtenir un objet BluetoothDevice qui représente le périphérique distant.

Pré-configuration : Taper 7 fois sur la description du 'numéro de build' pour activer le mode debug.

#Monitoring de processus à distance

Description de l'application :

Notre application sera divisée en trois activités :
La première sera l'activité principale (MainActivity) ou l'utilisateur aura le choix de se connecter en tant que serveur et client.
Si l'utilisateur choisit de se connecter en tant que client, le clic du bouton est capturé et l'activité client est lancée.
Au sein de cette activité est défini le socket client, et le thread de connexion lancé.
Au contraire, si l'utilisateur clique sur le bouton serveur, l'activité est lancée, un serveur socket est créé et est mis à l'écoute en attente d'une demande de connexion.



Socket Bluetooth : connexion

Lorsque notre application est lancée, nous instancions les boutons pour demander à l'utilisateur de choisir entre Client et Serveur et les configurer pour écouter :

```

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.clt:
            Button btn = (Button) findViewById(R.id.clt);
            // Création d'une intention
            Intent cltIntent = new Intent( packageContext: this, Client.class);
            // Lancer l'activité client
            startActivity(cltIntent);
            break;

        case R.id.srv:
            Button server = (Button) findViewById(R.id.srv);
            // Création d'une intention
            Intent servIntent = new Intent( packageContext: this, Server.class);
            // Lancer l'activité serveur
            startActivity(servIntent);
            break;
    }
}

```

Figure – Bouton serveur /Client

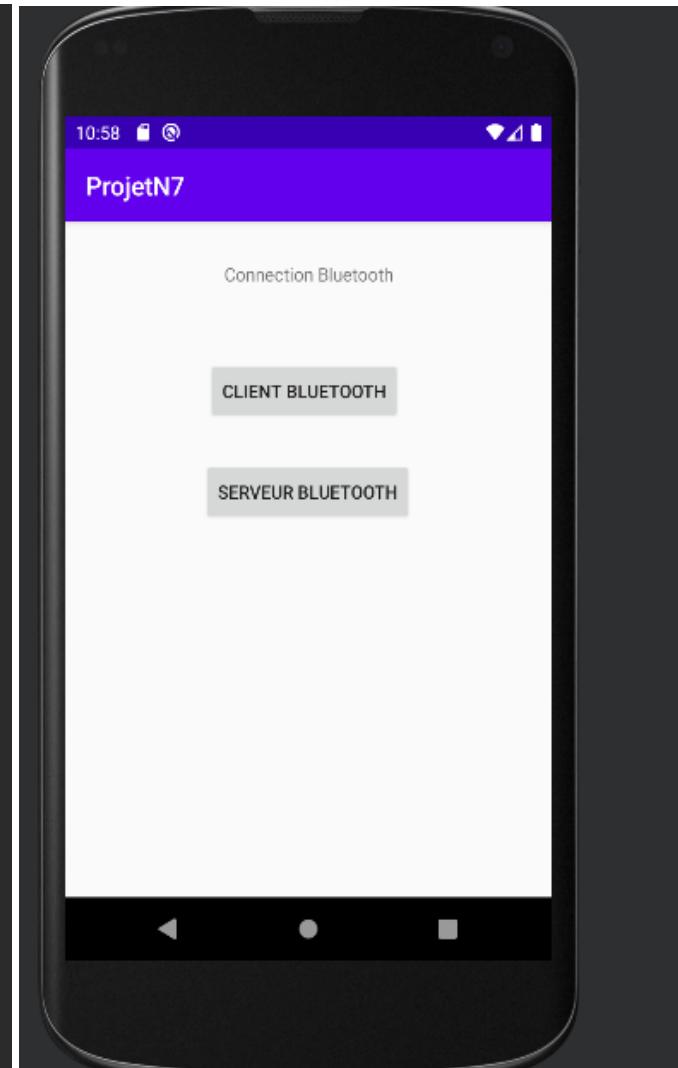


Figure – Écran d'invite de connexion

Nous capturons le clic sur le bouton et selon l'id, nous modifions l'affichage en attendant que la connexion soit établie. Pour établir la connexion, du côté client, nous devons récupérer la liste des appareils appariés par Bluetooth et choisir le bon appareil. Enfin, nous lançons un thread client et un thread serveur.

```
@SuppressWarnings("MissingPermission")
public ServerClass() {

    BluetoothServerSocket tmp = null;
    try {

        tmp = MyBtHandler.listenUsingRfcommWithServiceRecord( name: "Projet_N7", MY_UUID); // Server socket
        Log.d(TAG, msg: "Le serveur se connecte");

    } catch (IOException e) {
        Log.e(TAG, msg: "Socket's listen() method failed");
    }
    monServerSocket = tmp;
}

public void run(){
    Log.d(TAG, msg: "Accept thread running...");
    BluetoothSocket socket=null;
    try{
        Log.d(TAG, msg: "Server starts...");
        socket=monServerSocket.accept(); // Accepte la connexion du client
        Log.d(TAG, msg: "Connection accepted...");
        Button btn = (Button) findViewById(R.id.serv_bt);
        btn.setText("Ready !");

    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figure – Thread Serveur

Remarque : L'exécution du thread serveur se fait par un appel à la méthode accept().

```
@SuppressWarnings("MissingPermission")
public ClientClass(BluetoothDevice mon_appareil) {
    device = mon_appareil;
    // Obtenez un BluetoothSocket pour vous connecter avec le BluetoothDevice donné.
    try {

        socket = device.createRfcommSocketToServiceRecord(MY_UUID); // Creation du socket client

    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```
@SuppressWarnings("MissingPermission")
public void run() {

    try {
        Log.d(TAG, msg: "Starting to connect");
        socket.connect();
        Log.d(TAG, msg: "Connection made");

    } catch (IOException e) {

        e.printStackTrace();
    }
}
```

Figure – Thread Client

Le thread Client effectue un appel à la méthode de connect sur le téléphone choisi avant l'exécution du thread

```
public void BT_Conn (BluetoothSocket socket) {  
    bluetoothSocket = socket;  
    InputStream temponIn = null;  
    OutputStream temponOut = null;  
  
    try {  
        temponIn = bluetoothSocket.getInputStream();  
        temponOut = bluetoothSocket.getOutputStream();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
  
    inputStream = temponIn;  
    outputStream = temponOut;  
}
```

L'instanciation du thread BT-connexion permet les échanges de messages via le socket Bluetooth

Références :

<https://developer.android.com>

<https://developer.android.com/guide/topics/connectivity/bluetooth/connect-bluetooth-devices>