

# Communication Bluetooth

## Projet INKSMob

Master 2

Génie Informatique, Mobile et Embarqué

31 janvier 2018



# Plan du cours

- 1 Plan du cours
- 2 Introduction
  - Généralités
- 3 Fonctionnement
  - Procédure
- 4 Mise en œuvre sous Android
  - Permissions
  - BluetoothAdapter
  - Activation
  - Visibilité
  - Scan local
  - Appairage
  - Demo



# Introduction

## Généralités

- Inventé par Ericsson en 1994 pour remplacer le RS232
- Optimisé pour les communications à courte distance
- Récepteurs basse consommation peu coûteux
- Bande radio de 2.4GHz à 2.485GHz
- Portée de 10 à 100m selon les versions



# Fonctionnement

## Procédure

- Activation du Bluetooth
- Rendre son appareil visible
- Scan local des appareils visibles
- Demande d'appairage
- Communication sécurisée via des sockets

## Code source

<https://github.com/CaillPa/bluetooth.git>



# Permissions

Utilisation du Bluetooth :

```
<uses-permission android:name="android.permission.BLUETOOTH" />
```

Scan local, demande d'appairage, (des)activation du Bluetooth :

```
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Utilisation du Bluetooth pour la localisation :

```
<uses-permission  
android:name="android.permission.ACCESS_COARSE_LOCATION" />
```



# Objet BluetoothAdapter

La manipulation de la carte bluetooth se fait via un objet `BluetoothAdapter`.  
Cet objet est obtenu grâce à la méthode `BluetoothAdapter.getDefaultAdapter()`

## Exemple

```
BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();  
if (mBluetoothAdapter == null) {  
    // Device doesn't support Bluetooth  
}
```



# Activation du Bluetooth

2 méthodes pour activer le Bluetooth :

## Popup interactif

```
if ( !mBluetoothAdapter.isEnabled()) {  
    Intent enableBtIntent = new  
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);  
    startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);  
}
```

## Direct (nécessite permission admin)

```
if(bluetoothAdapter.isEnabled()) {  
    bluetoothAdapter.disable();  
} else {  
    bluetoothAdapter.enable();  
}
```



# Rendre son appareil visible

Via un intent (interactif)

```
Intent discoverableIntent =  
    new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);  
discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION,  
    DISCOVERABLE_DURATION_SECONDS);  
startActivityForResult(discoverableIntent, BT_DISCOVER);
```





# Scan local des appareils visibles

## Recuperation des appareils déjà appairés

```
Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();  
if (pairedDevices.size() > 0) {  
    // There are paired devices. Get the name and address of each paired device.  
    for (BluetoothDevice device : pairedDevices) {  
        String deviceName = device.getName();  
        String deviceHardwareAddress = device.getAddress(); // MAC address  
    }  
}
```





## Scan local des appareils visibles (suite)

### Enregistrement de l'IntentFilter

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
    registerReceiver(mReceiver, filter);
}
```

### Libère l'IntentFilter

```
@Override
protected void onDestroy() {
    super.onDestroy();
    unregisterReceiver(mReceiver);
}
```





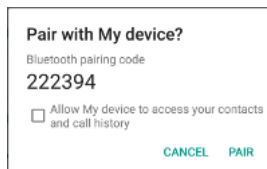
## Scan local des appareils visibles (suite bis)

### Broadcast Receiver

```
private final BroadcastReceiver mReceiver = new BroadcastReceiver() {  
    public void onReceive(Context context, Intent intent) {  
        String action = intent.getAction();  
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {  
            // Discovery has found a device. Get the BluetoothDevice  
            // object and its info from the Intent.  
            BluetoothDevice device =  
                intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);  
        }  
    }  
};
```

# Demande d'appairage

Méthode `BluetoothDevice.createBond()`



## Demo