



# Formation SIN

Utilisation du logiciel Basic4Android

Formateur : Pierre Aguillon



## PARTIE 3 : PROGRAMME 3

### Objectifs :

- Mettre en œuvre une communication en UDP ;
- Utiliser le logiciel gratuit Hercules ;
- Installer des librairies optionnelles;

## Troisième programme

Ce troisième programme va nous permettre de rendre communicante la tablette via le réseau Wifi en utilisant le protocole UDP.

Pour pouvoir mettre en œuvre les programmes, il nous faut l'environnement suivant :

- Une tablette connectée en Wifi sur un réseau et dont on connaît l'adresse IP attribuée par le serveur DHCP ;
- Un ordinateur connecté (peu importe le type de connexion) sur le même réseau et dont on connaît l'adresse IP attribuée par le serveur DHCP (IPCONFIG) ;
- Le logiciel Hercules de chez HWgroup que l'on peut télécharger ici [http://www.hw-group.com/products/hercules/index\\_en.html](http://www.hw-group.com/products/hercules/index_en.html) .

En UDP, le principe est simple : chacune des deux applications (Hercules et votre programme B4A) doit ouvrir un port sortant, et écouter sur son port entrant.

**Dans les exemples, il faudra penser à bien changer les adresses IP en positionnant celles correspondant à votre configuration. Elles seront surlignées dans les exemples en rouge.**

La première partie du programme va consister à faire envoyer par la tablette deux octets à destination du PC, suite à l'appui sur un bouton.

Créez un nouveau programme que vous appellerez « **programme n3** » .

L'utilisation du protocole UDP nécessite l'utilisation de la librairie **NetWork**. Cochez-la dans votre liste de librairie.

Lancez le Designer et placez dans votre feuille un bouton. Générez les membres, seul le **click** sur le bouton nous est utile. Sauvegardez votre feuille sous le nom **layout1** et fermez le Designer.

L'utilisation en émission UDP sous B4A se passe en quatre étapes : la création du socket UDP, son initialisation (adresse IP et port entrant), le formatage du paquet à envoyer et l'envoi lui-même.

### Création du socket UDP

```
1 Sub Process_Globals
2     Dim UDPSocket1 As UDPSocket
3 End Sub
```

### Initialisation du socket

```
9 Sub Activity_Create(FirstTime As Boolean)
10     Activity.LoadLayout("layout1")
11     UDPSocket1.Initialize("UDP", 0, 1024)
12 End Sub
```

*Notes : le nom des événements associés à notre socket sera **UDP** (pas utilisé pour l'émission), en plaçant le port entrant à **0** on refuse la réception, **1024** correspond à la taille mémoire en octet attribuée au buffer de réception.*

## Le formatage du paquet

```

21 Sub button1_click
22   Dim data(2) As Byte
23   data(0)=0x33
24   data(1)=0x34
25   Dim Packet As UDPPacket
26   Packet.Initialize(data, "192.168.1.90", 5500)

```

Note : avec les trois premières lignes on crée un tableau (**data**) d'octets avec les valeurs **0x33** et **0x34** (code ASCII du **3** et **4**), puis on définit une variable **Packet** de type **UDPPacket**. Enfin on formate le paquet UDP en précisant le champ de données (**data**), l'adresse IP de destination (**192.168.1.90**), et le port sortant (**5500**).

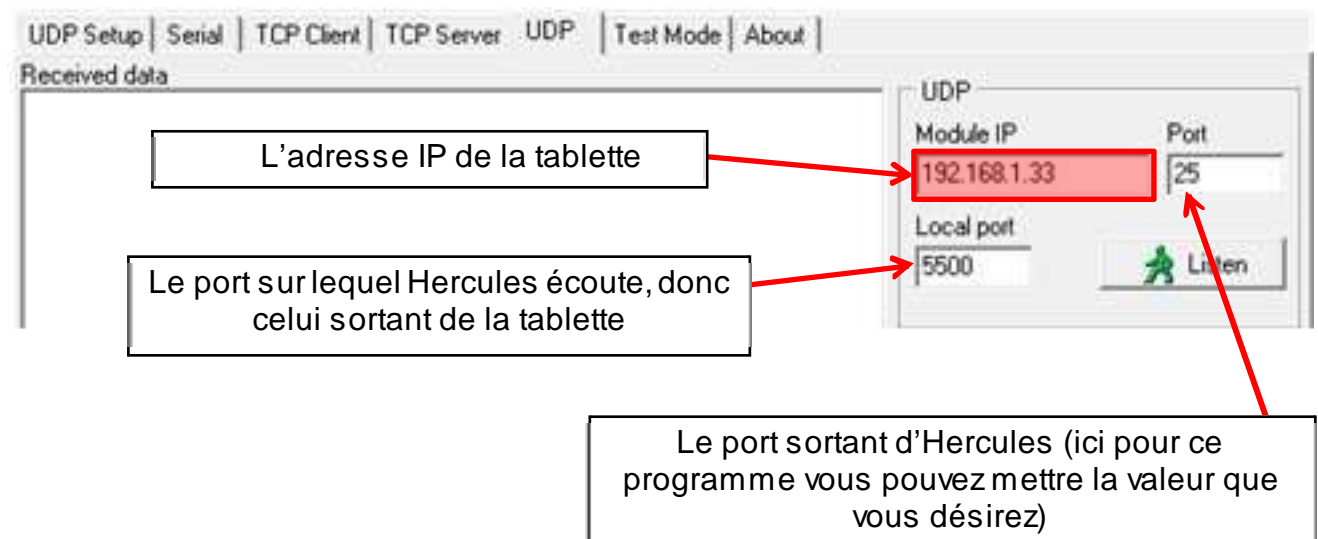
## L'envoi

```

27   UDPsocket1.Send(Packet)
28 End Sub

```

Configurez maintenant le logiciel Hercules pour qu'il assure une liaison UDP avec les paramètres suivants :



Dans Hercules, cliquez sur le bouton **Listen** .

Lancez votre programme sur la tablette (après l'avoir installé). En appuyant sur le bouton vous devriez recevoir **34** (en fait 3 puis 4) sur Hercules dans la fenêtre de réception. En cliquant avec le bouton droit de la souris dans cette même fenêtre vous pouvez décider d'afficher la réception en hexadécimal **{33}{34}**.

Maintenant que l'on a envoyé deux octets, on va réaliser la même chose mais pour la transmission d'un message de type texte « Bonjour STI2D ».

L'opération au niveau UDP reste la même. C'est au niveau de la construction du tableau Data que cela change. Pour cela on va utiliser la fonction **GetBytes** dont le rôle est de construire un tableau d'octet ayant pour valeur les codes ASCII d'un message de type texte (String).

```

24 Sub button1_click
25     Dim data() As Byte
26     data = "Bonjour STI2D".GetBytes("ASCII")
27     Dim packet As UDPPacket
28     packet.Initialize(data, "192.168.1.90", 5500)
29     UDPsocket1.Send(packet)
30 End Sub

```

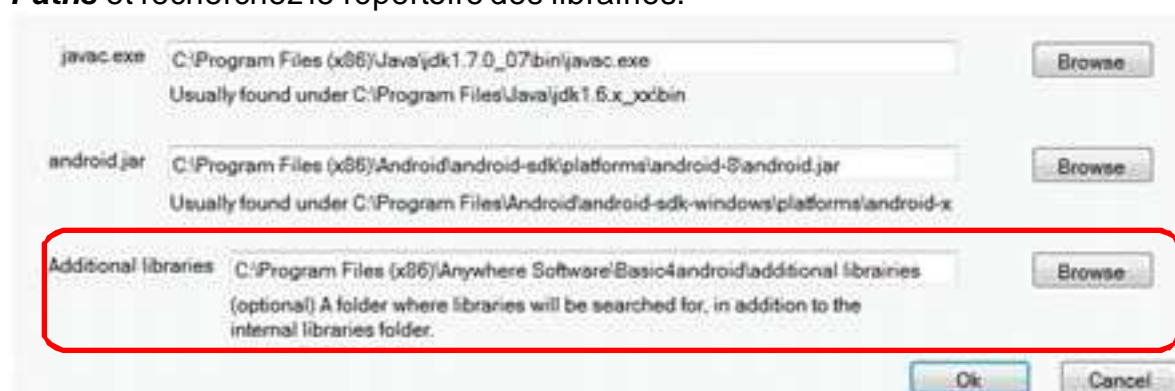
La manipulation de variable, et le changement d'un format à l'autre, sont très fréquents avec B4A. Pour cela une librairie non officielle (donc non livrée avec le logiciel) permet de nous simplifier la tâche. Il s'agit de la librairie **ByteConverter** que l'on trouve ici :

<http://www.basic4ppc.com/forum/additional-libraries-classes-official-updates/6787-byteconverter-library.html#post39339>

On va maintenant voir comment installer une librairie supplémentaire. Pour cela quittez B4A et placez-vous dans le répertoire d'installation de B4A (par défaut

**C:\Program Files\Anywhere Software\Basic4android**). Créez un répertoire (**Additional Libraries** par exemple) et copiez-y les fichiers **ByteConverter.jar** et **ByteConverter.xml**.

Relancez B4A et ouvrez votre dernier programme (programme n°3). Sous l'onglet **Libs** la librairie **ByteConverter** n'apparaît pas. C'est normal puisque vous n'avez pas précisé le chemin d'accès des librairies additionnelles. Pour cela, utilisez le menu **Tools-Configure Paths** et recherchez le répertoire des librairies.



Validez avec **OK**. Maintenant la librairie doit apparaître dans l'onglet **Libs**. Sélectionnez-la.

La description complète de la librairie est ici :

<http://www.basic4ppc.com/android/help/byteconverter.html>

On va voir ce que cela peut donner dans notre programme précédent.

```

24 Sub button1_click
25     Dim data() As Byte
26     Dim bc As ByteConverter
27     data = bc.StringToBytes("Bonjour STI2D", "ASCII")
28     Dim packet As UDPPacket
29     packet.Initialize(data, "192.168.1.90", 5500)
30     UDPsocket1.Send(packet)
31 End Sub

```

Cette librairie est très utile (elle devrait même me semble-t-il être intégrée par défaut avec B4A).

On va maintenant recevoir des données depuis Hercules à destination de la tablette. Après l'initialisation de votre socket la tablette écoute son port entrant. Hélas, pour le moment, la valeur 0 dans le champ port, indique à la tablette de ne pas écouter. On va donc déjà modifier cette ligne et attribuer un port entrant valide (3200 par exemple) :

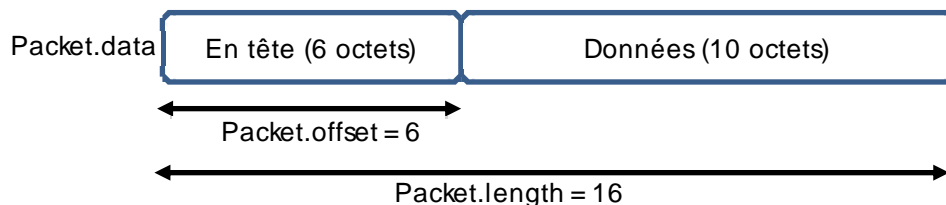
```
12 Sub Activity_Create(FirstTime As Boolean)
13     Activity.LoadLayout("layout1")
14     UDPSocket1.Initialize("UDP", 3200, 1024)
15 End Sub
```

L'évènement qui correspond à l'arrivée de paquets sur le port entrant est :

**UDP\_PacketArrived (Packet As UDPPacket)**. C'est **UDP** car vous avez défini ce nom dans l'initialisation du socket.

Lancez le Designer et placez un Label (laissez son nom par défaut **Label1**) suffisamment grand pour contenir le texte que vous allez recevoir. Fermez le Designer en sauvegardant votre feuille.

Les données brutes reçues par le buffer **Packet** sont placées dans **Packet.data**. Plus que les données effectives le **Packet.data** peut contenir des octets d'en-tête. Il faut donc extraire les seules informations correspondant au message reçu.



Pour cela on dispose de deux champs supplémentaires :

- **Packet.Length** : c'est la longueur de Packet.Data
- **Packet.offset** : c'est la longueur du décalage avant les données

Il faut donc modifier notre programme de la manière suivante :

```
18
19
20 Sub UDP_PacketArrived (Packet As UDPPacket)
21     Dim msg As String
22     msg = BytesToString(Packet.data, Packet.offset, Packet.length, "ASCII")
23     Label1.Text=msg
```

Lancez votre programme sur la tablette. Dans Hercules sur le PC modifiez les paramètres :

