

Ce programme va vous permettre de rendre communicante le SmartPhone via le réseau Wifi en utilisant le protocole UDP.

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

- Un SmartPhone 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

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 le SmartPhone deux octets à destination du PC, suite à l'appui sur un bouton.

1. Créez un nouveau programme que vous appellerez « **UDP** » .
2. L'utilisation du protocole UDP nécessite l'utilisation de la librairie **NetWork**. Cochez-la dans votre liste de librairie.
3. 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.

4. Création du socket UDP

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

5. 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.*

6. 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**).*

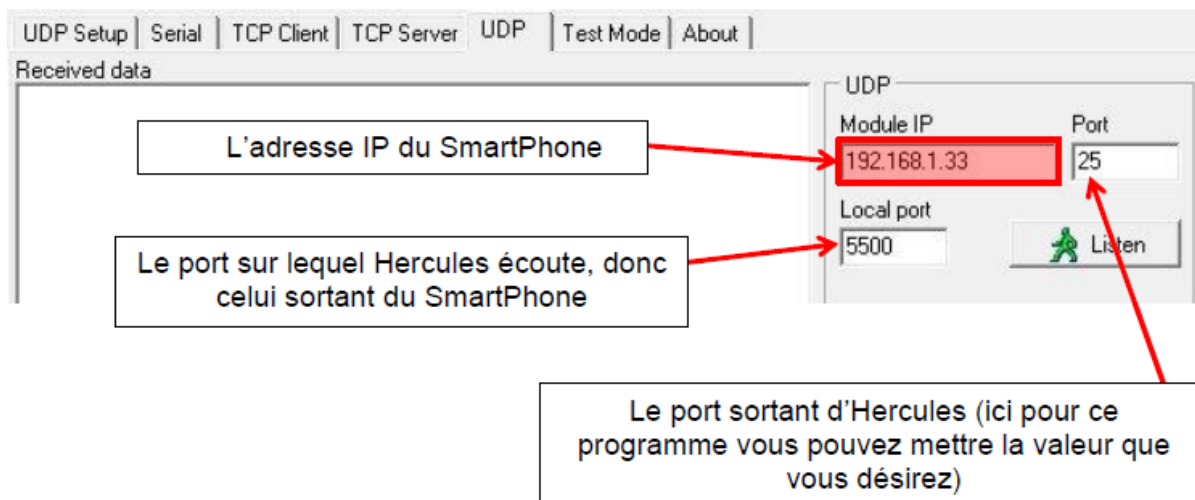
7. L'envoi

```

27     UDPocket1.Send(Packet)
28     End Sub

```

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



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

10. Lancez votre programme sur le SmartPhone. 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}**.

Modification n°1

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

```

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     UDPocket1.Send(packet)
30 End Sub

```

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

Modification n°2

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>

Vous allez maintenant voir comment installer une librairie supplémentaire.

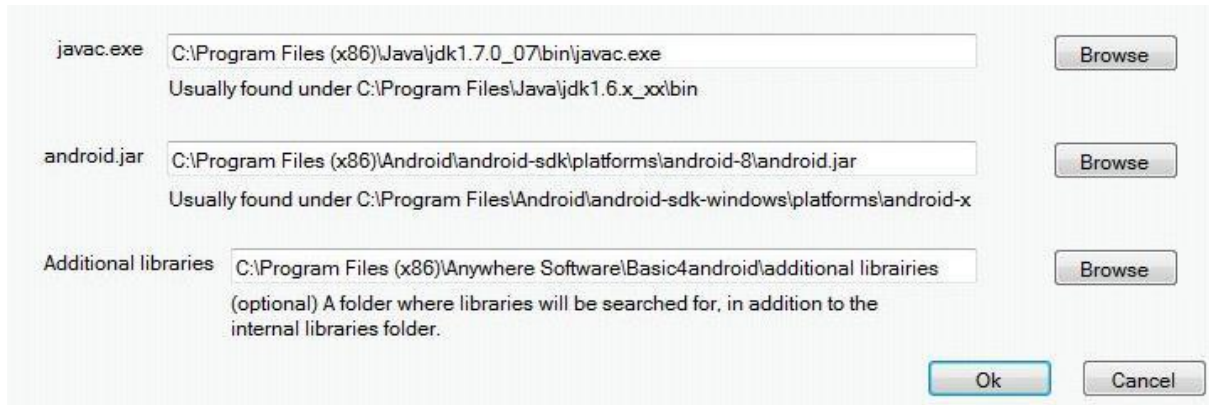
13. Quittez B4A et placez-vous dans le répertoire d'installation de B4A (par défaut **C:\ProgramFiles\Anywhere Software\Basic4android**).

14. Créez un répertoire (**Additional Libraries Formation** par exemple) et copiez-y les fichiers **ByteConverter.jar** et **ByteConverter.xml**.

15. 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.

16. Utilisez le menu **Tools-Configure Paths** et recherchez le répertoire des librairies.



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

18. Sélectionnez-la.

La description complète de la librairie est ici :

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

19. Modifiez votre programme de la façon suivante :

```

23
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).

20. Exécutez votre programme sur le SmartPhone et constatez la réception du message texte sous Hercules.

Modification n°3

Vous allez maintenant recevoir des données depuis Hercules à destination de le SmartPhone. Après l'initialisation de votre socket le SmartPhone écoute son port entrant. Hélas, pour le moment, la valeur 0 dans le champ port, indique au SmartPhone de ne pas écouter.

21. Vous allez 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

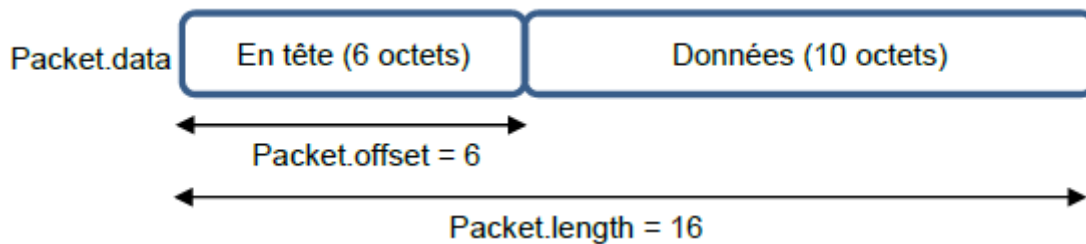
```

IHM : UDP

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 :

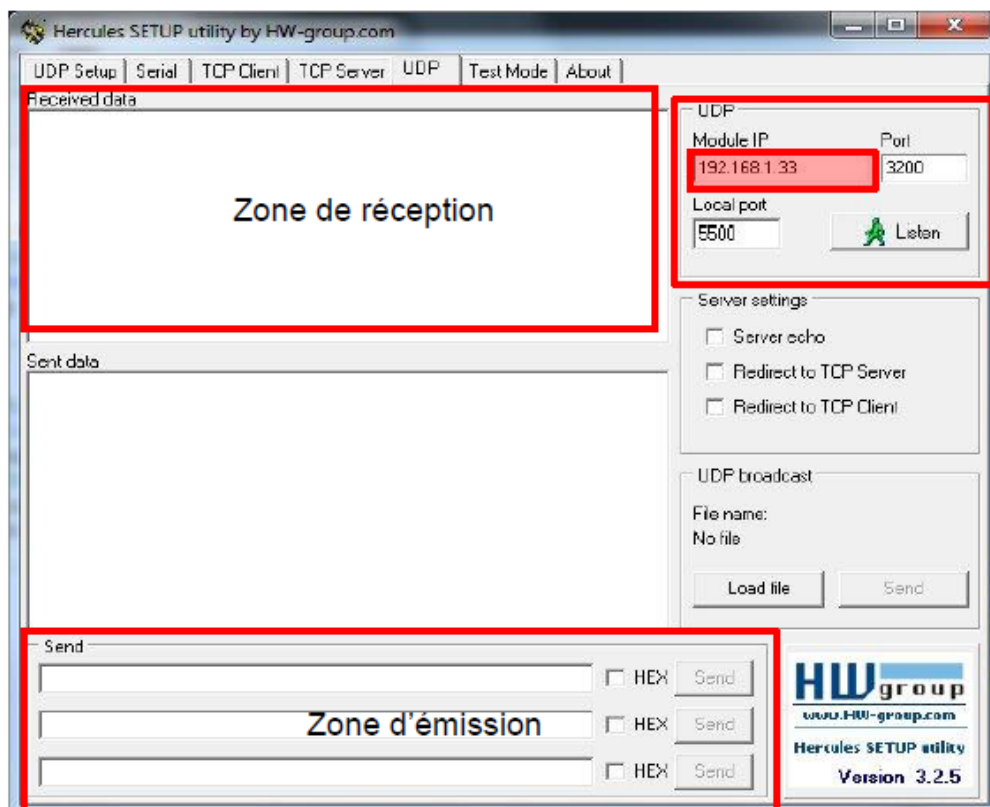
- **Packet.Length** : c'est la longueur de Packet.Data
- **Packet.offset** : c'est la longueur du décalage avant les données

- Modifiez votre 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 le SmartPhone.

- Dans Hercules sur le PC modifiez les paramètres :



26. Saisissez dans la zone d'émission d'Hercules un texte et constatez la bonne réception sur le SmartPhone. De même vérifiez que l'émission depuis le SmartPhone vers Hercules est toujours opérationnelle.

Utilisation des bibliothèques simplifiées

Le concept de socket UDP peut être quelque peu déroutant pour nos élèves de STI2D. Aussi pour simplifier la transmission en UDP, j'ai compilé une bibliothèque permettant de rendre transparentes certaines étapes.

Vous allez maintenant installer la bibliothèque supplémentaire UDP.

27. Quittez B4A et placez-vous dans le répertoire des bibliothèques additionnelles de B4A (celui que vous avez créé précédemment).

28. Copiez-y les fichiers **UDP.jar** et **UDP.xml**.

29. Relancez B4A pour créer un nouveau programme (programme 4) et cochez la bibliothèque additionnelle UDP

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

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

L'utilisation en émission UDP en utilisant la bibliothèque sous B4A se passe en deux étapes : l'initialisation UDP, et l'envoi du paquet.

31. Saisissez les lignes suivantes :

```
28 Sub Activity_Create(FirstTime As Boolean)
29     'Do not forget to load the layout file created with the visual designer. For example:
30     Activity.LoadLayout("Layout1")
31     UDP.Initialise(0)
32 End Sub
```

En précisant **0** dans l'appel de la fonction **UDP.Initialise**, vous ne précisez aucun port entrant pour le SmartPhone. C'est logique, puisque dans ce premier programme, on désire que celle-ci ne réalise que l'envoi de données.

32. On va maintenant construire le tableau des données que l'on désire transmettre :

```
43 Sub send_Click
44     Dim data(2) As Byte
45     data(0)=0x33
46     data(1)=0x34
```

Ici donc les octets 0x33 et 0x34.

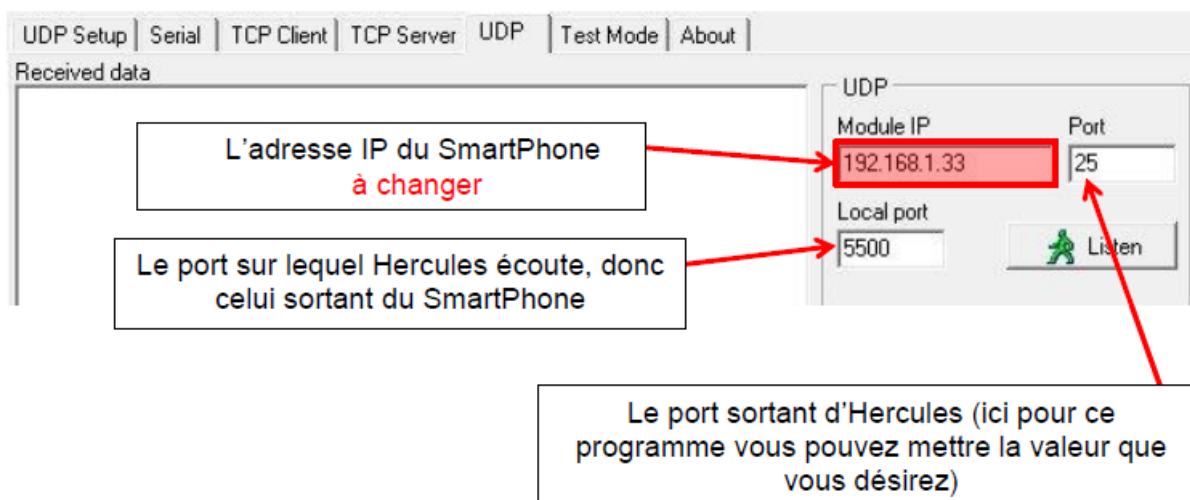
33. Reste à appeler la fonction réalisant l'envoi de notre tableau de données :

```
43 Sub send_Click
44     Dim data(2) As Byte
45     data(0)=0x33
46     data(1)=0x34
47     UDP.emission("192.168.1.91",5500,data)
48 End Sub
```

Ici on décide d'envoyer sur un poste ayant l'adresse IP 192.168.1.91 et ayant ouvert le port 5500. Il vous faut changer cette adresse et préciser celle de la clé Wifi de votre PC.

34. Compilez votre programme et installez-le dans le SmartPhone.

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



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

37. Lancez votre programme sur le SmartPhone (après l'avoir installé). En appuyant sur le bouton vous devriez voir **34** sur Hercules dans la fenêtre de réception.

La transmission d'un message texte n'est pas différente.

38. Cochez la librairie **ByteConverter** dans votre liste de librairie.

39. Vous allez utiliser la fonction **StringToBytes** permettant de réaliser la conversion en ASCII et la construction du tableau **Data** :

```
43 Sub send_Click
44     Dim data() As Byte
45     Dim bc As ByteConverter
46     data=bc.StringToBytes("Bonjour STI2D","ASCII")
47
48     UDP.emission("192.168.1.91",5500,data)
49 End Sub
```

40. Compilez votre programme et testez-le.

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

41. Modifiez la ligne :

```
28 Sub Activity_Create(FirstTime As Boolean)
29     'Do not forget to load the layout file created with the visual designer. For example:
30     Activity.LoadLayout("Layout1")
31     UDP.Initialise(3200)
32 End Sub
```

42. 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.

43. L'évènement qui correspond à l'arrivée de paquets sur le port entrant est : **UDP_PacketArrived (Packet As UDPPacket)**. Saisissez la fonction :

```
51 Sub UDP_packetArrived(packet As UDPPacket)
52
53
54
55 End Sub
```

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

44. Pour cela on commence par créer une variable de type **String** qui contiendra les données réelles :

```
51 Sub UDP_packetArrived(packet As UDPPacket)
52     Dim message As String
53
54
55 End Sub
```

45. Puis on extrait les données réelles de celles reçues :

```
51 Sub UDP_packetArrived(packet As UDPPacket)
52     Dim message As String
53     message=UDP.reception(packet)
54
55
56 End Sub
```

46. Et enfin on les place dans le **Label** :

```
51 Sub UDP_packetArrived(packet As UDPPacket)
52     Dim message As String
53     message=UDP.reception(packet)
54     Label1.text=message
55 End Sub
```

47. Dans Hercules sur le PC modifiez les paramètres pour préciser le port entrant 3200.

48. Compilez votre programme et testez-le.