

## 2.3 TP Ethernet Client

### Prés requis :

Ouvrir un projet java avec Eclipse

La communication Ethernet est un mode connecté (socket / prises), ici nous utilisons des flux de communications (stream) en entrée et en sortie (InputStream / OutputStream comme en bluetooth), le protocole de communication est le protocol TCP/IP ( Transfert contrôle protocole sur IP (Internet Protocole) )

Le protocole de communication nécessite pour initier le dialogue de créer une prise de communication (socket) par conséquent d'utiliser un port de communication ainsi que l'adresse IP du serveur.

Ce protocole de communication est différent du protocole UDP (mode non connecté) qui ne nécessite pas de connaître l'adresse du poste distant (utilisation du broadcasting pour récupérer l'adresse distante).

a) On crée un nouvel objet de connexion de type Socket en précisant l'adresse du serveur et le port de communication qu'utilise le serveur.

***SocketClient = new Socket(strAdresseServeur, intPort); //Port 1234***

b) On crée un flux de sortie sur le socket pour envoyer des données au serveur.

***FluxSortieEthernet = SocketClient.getOutputStream();***

c) On crée un flux d'entrée sur ce socket pour recevoir des données du serveur.



### ***FluxEntreeEthernet = SocketClient.getInputStream()***

d) Il est donc possible ensuite de lire des tableaux de bytes en provenance du serveur et de connaître le nombre de bytes reçus

*intNbreCarlusEthernet = FluxEntreeEthernet.read(byteTabAccuseReception);*

e) Il est aussi possible d'envoyer des données (tableau de bytes) au client, par exemple des accusés de réception.

*FluxSortieEthernet.write(RequeteEnvoyee.getBytes());*



### 2.3.1 Créer un Client Ethernet en Java sur PC avec Eclipse

- a) Vous ouvrirez le projet java (**Client\_Java\_TCP\_Socket\_v1.0ac**) avec l'IDE Eclipse
- b) Vous compléterez le programme avec les informations précédentes
- c) Ici pas besoin d'ajouter de librairies (pas comme en bluetooth)
- d) Vous exécuterez le programme. Si tout se passe correctement vous devriez obtenir ceci.

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows "Graphical Layout" and "client\_android\_tcp\_activity.xml".
- Bottom Bar:** Shows tabs for "Problems", "Javadoc", "Declaration", "Console", and "Progress".
- Console View:** Displays the output of the Java application "Client\_Java\_TCP\_Socket [Java Application] C:\Program Files\Java\jre1.8.0\_151\bin\javaw.exe (19 déc. 2017 22:57:07)".
- Output:**

```
#                      Client TEST TCP#
##
# Auteur : Wilfrid Grassi#
# Date : 25/11/2017#
# Version : 1.0#
```
- Text Prompt:** "Entrez le port de communication du serveur :"

entrez le port de communication (1234)



The screenshot shows the Android Studio interface with the 'client\_android\_tcp\_activity.xml' file open in the graphical layout editor. The console tab is active, displaying the following text:

```
# Client TEST TCP
#
# Auteur : Wilfrid Grassi
# Date : 25/11/2017
# Version : 1.0
```

Below this, the console displays user prompts:

```
Entrez le port de communication du serveur :
1234
Entrez l'adresse du serveur :
```

**Entrez l'adresse du serveur pour cela vous lancerez un serveur sur un autre poste**



```
Problems @ Javadoc Declaration Console Progress
Client_Java_TCP_Socket [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (19 déc. 2017 23:06:09)

#
# Client TEST TCP
#
# Auteur : Wilfrid Grassi
# Date : 25/11/2017
# Version : 1.0

-----
Entrez le port de communication du serveur :
1234
Entrez l adresse du serveur :
192.168.1.43
Attente de l accuse de reception Serveur : (Connexion) Accuse de reception du serveur recu.
Serveur connecte

Saisissez les donnees ou commandes a envoyer :
```



## Envoyons maintenant des données au serveur distant

```
Graphical Layout client_android_tcp_activity.xml
Problems @ Javadoc Declaration Console Progress
Client_Java_TCP_Socket [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (19 déc. 2017 23:06:09)
-----
# Client TEST TCP #
#
# Auteur : Wilfrid Grassi
# Date : 25/11/2017
# Version : 1.0
#
-----
Entrez le port de communication du serveur :
1234
Entrez l adresse du serveur :
192.168.1.43
Attente de l accuse de reception Serveur : (Connexion) Accuse de reception du serveur recu.
Serveur connecte

Saisissez les donnees ou commandes a envoyer :
envoyons des données au serveur
Attente de l accuse de reception Serveur : (OK) Accuse de reception du serveur recu.

Saisissez les donnees ou commandes a envoyer :
```



## Déconnectons le client du serveur

```
Problems @ Javadoc Declaration Console Progress
Client_Java_TCP_Socket [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (19 déc. 2017 23:11:52)
-----
Entrez le port de communication du serveur :
1234
Entrez l adresse du serveur :
192.168.1.43
Attente de l accuse de reception Serveur : (Connexion) Accuse de reception du serveur recu.
Serveur connecte

Saisissez les donnees ou commandes a envoyer :
Envoyons des données au serveur
Attente de l accuse de reception Serveur : (OK) Accuse de reception du serveur recu.

Saisissez les donnees ou commandes a envoyer :
Cmd : ClientDeConnecte
Attente de l accuse de reception Serveur : (OK) Accuse de reception du serveur recu.
Fermeture du Client !!!
-----
#                      Client  TEST TCP
#
# Auteur : Wilfrid Grassi
# Date : 25/11/2017
# Version : 1.0
#
-----
Entrez le port de communication du serveur :
```



### 2.3.2 Exécuter le Client Ethernet Java sur Raspberry

Pour cela vous utiliser le partage réseau de votre raspberry et copierez le dossier <Client\_Java\_TCP\_Socket\_v1.0ac> dans le dossier <Pi3-xx/Pi/Document> puis vous utiliserez le script "StartX\_Client\_Java\_TCP\_Socket.sh" pour lancer votre programme Client java

### 2.3.3 Connecter le Client Ethernet Raspberry avec le Serveur Java d'un autre poste