

# COURS ELE2705

## TRANSMISSION DE L'INFORMATION

### TRAVAIL PRATIQUE – Séance 5 (30 points)

#### Introduction

La combinaison de la télémétrie et de l'Internet des objets constitue une technologie évolutive de mesure à distance et de transmission de données qui s'est rapidement imposée comme une ressource incontournable dans divers secteurs industriels. Cette capacité à collecter des données en temps réel a révolutionné la manière dont nous surveillons et gérons des systèmes complexes. Dans le cadre de ce travail pratique, nous explorerons une méthodologie de transmission de ces données à partir d'un capteur vers un serveur central.

#### Explication de la Configuration

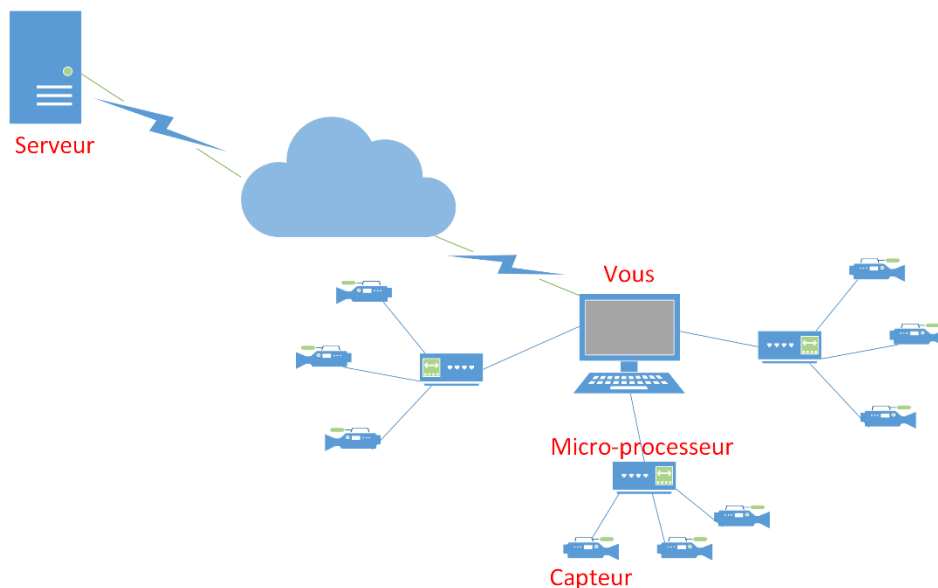


Figure 1: Architecture du réseau

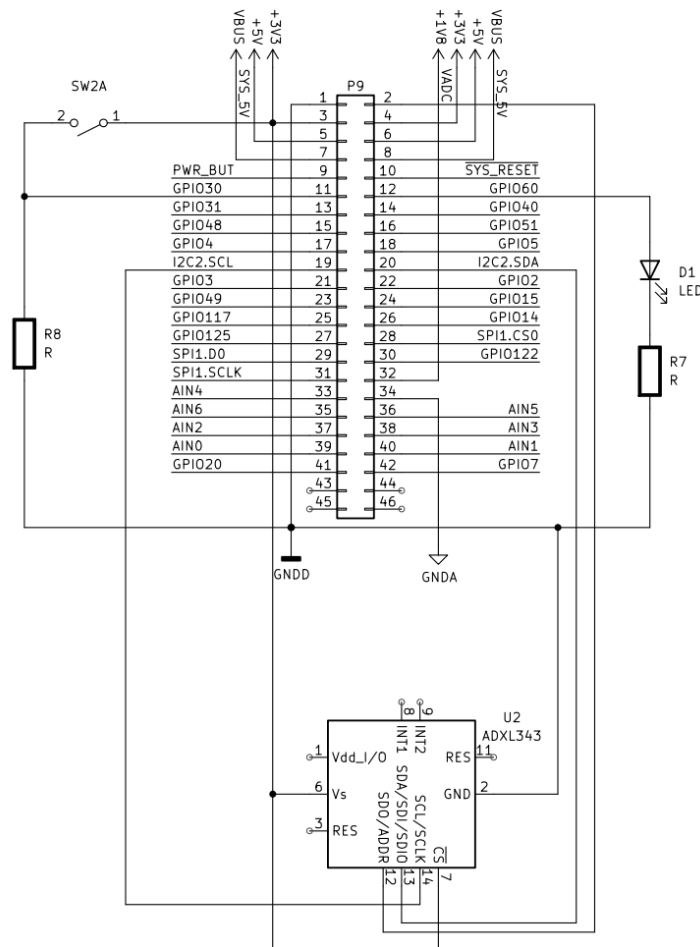
Le but de ce laboratoire est de transmettre des données d'un capteur au serveur. Comme le montre la Figure 1, le capteur est connecté à un microprocesseur via une interface de communication. Ce microprocesseur reçoit les données captées et les envoie au serveur à travers un réseau. De cette façon,

la connexion est établie entre le capteur et le serveur. Pour permettre d'utiliser l'infrastructure réseau du laboratoire et pour facilement interagir avec le microprocesseur, nous utilisons un ordinateur pour interfacer le microprocesseur avec le réseau. Dans la configuration utilisée, l'ordinateur joue le rôle de « proxy ». C'est-à-dire que lorsque le serveur veut communiquer avec un microprocesseur, il envoie les données à l'adresse IP de l'ordinateur correspondant vers un port choisi auparavant. L'ordinateur effectue un « port forwarding » au microprocesseur ciblé. Un code python est fourni pour vous pour faire ce port forwarding. Vous pouvez le lancer sur votre machine en utilisant la commande :

```
python port_forwarding.py
```

Vous devez spécifier les adresses IP et les ports sources et destinataires (Plus de détail dans [Partie 1 : Établissement de la Connexion \(23 points\)](#)).

Vous pouvez constater qu'un circuit a été ajouté au microprocesseur pour lui permettre d'interfacer avec différents composants et capteurs. Ce circuit est illustré dans la figure ci-dessous.



Les boutons activent leur capteur correspondant en invoquant du code dans le microprocesseur. La DEL s'allume quand le microprocesseur reçoit une réponse du serveur.

## Partie 1 : Établissement de la Connexion (23 points)

Nous voulons établir une connexion UDP entre le serveur et notre microprocesseur. Le serveur est déjà programmé pour recevoir vos messages et envoyer des réponses. De plus, les fonctions permettant d'interfacer avec les capteurs sont aussi fournis. Dans cette partie du laboratoire, vous devez programmer le client UDP sur le microprocesseur. Nous vous FORTEMENT conseillons d'écrire votre code localement sur votre machine puis l'envoyer au microprocesseur pour éviter d'oublier de récupérer votre fichier à la fin de la séance.

Avant de brancher le microprocesseur, utilisez la commande `ipconfig`. Maintenant, connectez votre BeagleBone Black à votre machine. Une autre fois, utilisez la commande `ipconfig`.

1. (2 pts) Quelle est l'adresse IP de l'interface de votre machine ajoutée après le branchement du BeagleBone ? Le BeagleBone prend automatiquement l'adresse IP 192.168.7.2.

Écrivez l'adresse IP du BeagleBone dans un navigateur web. Vous êtes maintenant à l'intérieur du système de fichier du microprocesseur.

2. (1 pt) Quel est le dossier par défaut qui est affiché ? Vous pouvez mettre votre code dans ce dossier pour pouvoir le voir dans le navigateur (Consultez [Annexe 1 : Téléchargement des données](#)).

Téléchargez le code trouvé dans Moodle, et mettez-le dans le dossier trouvé ci-dessus.

Le serveur est en train d'écouter des paquets, et il va vous répondre dès qu'il reçoit vos données. Alors, vous avez besoin de **(1)** programmer le client pour envoyer les données reçues par le capteur, puis **(2)** recevoir et afficher la réponse du serveur.

3. (10 pts) Complétez le code qui vous est fourni. Vous allez principalement utiliser la librairie « socket » de python. Pour vous aider, cherchez la documentation de ces fonctions, et utilisez celles dont vous en avez besoin.

- `socket()`
- `bind()`
- `sendto()`
- `recvfrom()`

1. Indice : Vous aurez besoin des fonctions `string.encode()` et `string.decode()`.

5. (3 pts) Ajoutez les adresses et ports dans le script `port_forward.py`. Celui-ci va rouler sur votre machine, pas sur le microprocesseur.

3. Pour exécuter le code, il faut rouler le script `port_forward.py` sur votre machine. Puis, vous devez rouler le script `client_UDP.py` sur votre microprocesseur. Quand tout est prêt, appuyez sur le bouton pour activer le capteur.

6. (4 pts) Utilisez Wireshark pour analyser les paquets transmis et reçus pour chacun des capteurs sur l'interface entre le réseau local et l'ordinateur et entre l'ordinateur et le microprocesseur. Est-ce que les adresses IP correspondent au serveur et client ? Expliquez. Ajoutez des captures d'écrans et élaborez sur les similarités ou différences dans ces captures.

7. (3 pts) Expliquez le fonctionnement du code `port_forwarding.py` et, en utilisant les captures Wireshark, expliquez à quoi correspondent les paramètres que vous avez configuré (`LISTEN_ADDRESS`, `CONNECT_ADDRESS...`). Pourquoi est-ce que nous avons besoin de la fonction `bind()` pour `private_socket` mais pas pour `public_socket` ?

## Partie 2 : Exploration du Protocole I2C (7 points)

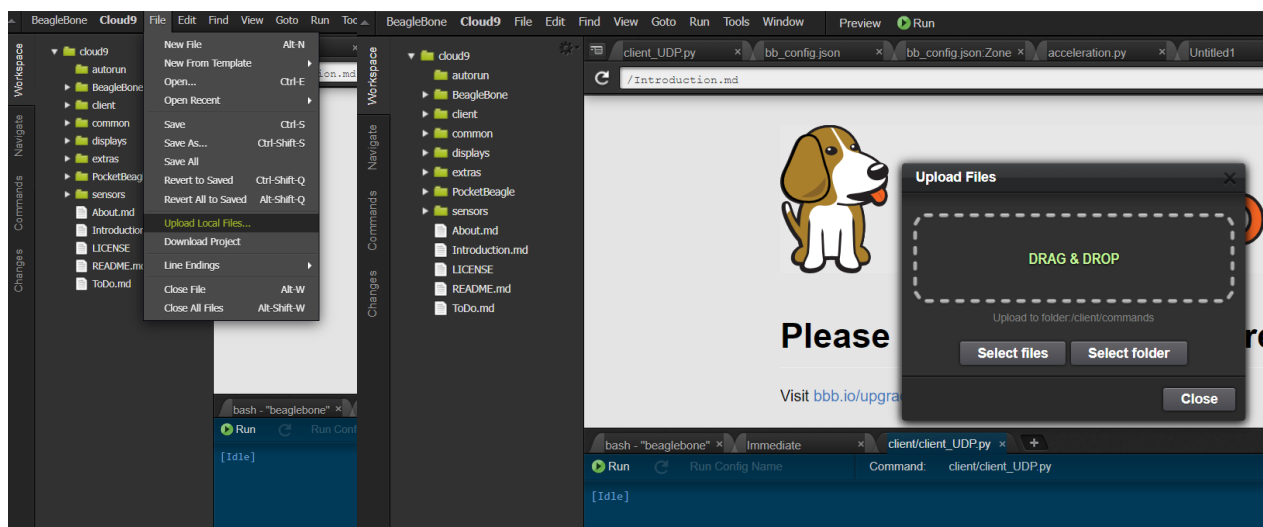
Vous pouvez trouver ici la documentation de l'accéléromètre : [ADXLS343 \(Rev. A\) \(analog.com\)](#). La section I2C (pages 16 et 17) explique le fonctionnement du protocole de communication utilisé pour envoyer les données.

1. (3 pts) L'accéléromètre communique avec le micro-processeur en utilisant le protocole I2C. Expliquez la structure d'une trame du protocole I2C. Incluez des références aux sites web que vous avez consulté (Wikipédia n'est pas une source fiable).
2. Branchez l'Analog Discovery à votre machine, et lancez le logiciel Waveforms. Dans WaveForms, cliquez sur l'onglet « Logic » pour ouvrir le Logic Analyser. Ajoutez un canal I2C, et notez les pins utilisés (par défaut DIO0 et DIO1 pour SCL et SDA respectivement). Sur votre circuit, il y a deux pins dédiés pour la capture des signaux SCL et SDA. Branchez-les avec les pins DIO0 et DIO1.
3. (4 pts) Utilisez l'Analog Discovery avec le logiciel WaveForms pour afficher les signaux SCL et SDA de I2C entre le microprocesseur et l'accéléromètre. Prenez plusieurs captures. Comparez-les. Qu'est-ce que vous constatez ? À votre avis, pourquoi y a-t-il plusieurs parties dans le signal ? Comparez avec le code.

## Annexe 1 : Téléchargement des données

Pour copier les fichiers au BeagleBone, vous pouvez :

- (1) utiliser l'interface graphique de BeagleBone. Taper l'adresse IP de BeagleBone 192.168.7.2 dans votre navigateur. Après aller à File -> Upload Local Files. Après vous pouvez faire Drag-and-Drop ou sélectionner des fichiers ou des dossiers (voir la figure ci-après).



ou

(2) utiliser la commande `scp` (Secure Copy) dans Linux. Ouvrez WSL, et lancez la commande suivante :

```
scp -r client debian@192.168.7.2:\[path/to/destination\]
```

où `[path/to/destination]` est le dossier que vous avez trouvé dans la question [2](#). Si vous trouvez l'erreur REMOTE HOST IDENTIFICATION HAS CHANGED, ça veut dire qu'un autre BeagleBone était connecté à votre machine avant. Utilisez la commande `ssh-keygen -f` générée pour enlever l'ancienne entrée. Maintenant, vous pouvez copier les fichiers au microprocesseur.