

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR

PRINCIPES ET ARCHITECTURE DES RÉSEAUX

(BUT1 RT TP G2A)

SAÉ - 1.02 : Réseaux informatiques

Élèves :

Alexis Déhu

Martin Motz

Enseignant :

Stéphane Mascaron

Préambule

Lors de cette SAE nous avons volontairement pas usé d'applications et/ou de logiciels pour effectuer nos manipulations. Ainsi nous utilisons uniquement des commandes bash présentes nativement sur la plupart des systèmes d'exploitation utilisant ce shell comme Ubuntu ou Debian par exemple. En conséquent, le nombre d'étapes et de manipulations se veut plus conséquent que si nous avions utilisé des outils pour effectuer les mêmes tâches.

Table des matières

1	Introduction	3
2	Approche du projet	3
2.1	Attentes du SAE	3
2.1.1	Matériel utilisé	3
2.1.2	Solutions que nous apportons	3
2.2	Schématisation de notre solution	3
2.2.1	Infrastructure réseau résumée	3
2.2.2	Illustration connexion du capteur au Raspberry Pi	4
3	Préparation du Raspberry Pi	5
3.1	Mise en place du SE	5
3.1.1	Récupération de l'image du SE	5
3.1.2	Découverte de la location du périphérique de stockage	5
3.1.3	Flash de l'image du SE sur la carte micro SD	5
3.2	Configuration du SE avant déploiement	5
3.2.1	Création d'un utilisateur et d'un mot de passe	5
3.2.2	Configuration réseau du SE	6
3.2.3	Activation du service SSH	6
4	Déploiement du serveur DHCP	6
4.1	Explications sur notre serveur DHCP	6
4.2	Création de la machine virtuelle	6
4.3	Configuration de l'interface côté réseau de l'IUT	6
4.3.1	Attribution d'une adresse IP statique à l'interface	6
4.3.2	Reinseignement de l'adresse IP du serveur DNS	6
4.4	Installation du service DHCP	6
4.4.1	Mise à jour des paquets du système	6
4.4.2	Installation d'un paquet de service de DHCP	7
4.5	Configuration de l'interface côté appareils	7
4.5.1	Sauvegarde de l'ancienne configuration de l'interface	7
4.5.2	Sauvegarde et modification du fichier de configuration DHCP	7
4.5.3	Vérification de l'état du service réseau	7
4.5.4	Vérification de l'état du service DHCP	7
5	Configuration d'un NAT	7
5.1	Explication de l'intérêt du NAT	7
5.2	Activation du NAT	7
5.3	Configuration du routeur NAT	7
5.3.1	Configuration de l'interface pour accepter le NAT	7
5.4	Contrôles supplémentaires après configuration	7
5.4.1	Vérification des routes IP	7
5.4.2	Initialisation d'une connexion SSH	7
5.5	Script d'automatisation optionnel	7
5.5.1	Script shell de configuration des services	7

6	Utilisation du capteur DHT20	7
6.1	Explications sur l'utilisation du capteur	7
6.2	Branchement sur le Raspberry Pi	7
6.2.1	Schéma de branchement	7
6.2.2	Photographie du branchement	7
6.3	Programme Python pour utiliser le capteur	7
6.4	Exemple de sortie du programme Python	8

1 Introduction

Le principe de cette SAÉ consiste à configurer un micro-ordinateur Raspberry Pi ("RPI") ainsi qu'une machine virtuelle ("VM") dans le réseau de l'IUT afin d'y connecter le RPI et de pouvoir y accéder depuis n'importe quelle machine du réseau local de l'IUT (LAN) via la VM.

Le RPI sera accessible depuis une connexion SSH et devra accéder à Internet, donc aux réseaux extérieurs (WAN).

Nous avons utilisé une VM Ubuntu 18.04 comme système d'exploitation pour la préparation du RPI et pour la VM.

Les commandes précédées d'un hashtag seront à être exécutées avec les droits administrateurs.

2 Approche du projet

2.1 Attentes du SAÉ

2.1.1 Matériel utilisé

- Micro-ordinateur Raspberry Pi 4 (8 Go) et son alimentation – "RPI"
- Carte micro SD 32 Go
- Adaptateur micro SD vers USB A – "Adaptateur"
- Machine virtuelle Ubuntu 18.04 sous VI4RT – "VM"
- Câbles RJ45 Ethernet droits
- Capteur DHT20 avec brins de connexion
- Résistance 10 Kilo Ohm

2.1.2 Solutions que nous apportons

Afin de répondre aux attentes du SAÉ nous allons monter une image de Raspberry Pi OS Lite, système d'exploitation propriétaire de Raspberry sur notre RPI avec un nom d'utilisateur et un mot de passe personnalisés. Nous devrons aussi ouvrir le port 22 du RPI afin de pouvoir y accéder via une connexion SSH. Sa configuration réseau sera aussi à vérifier.

En suivant, sur une VM créée sur VI4RT nous allons configurer un serveur DHCP auquel nous rattacherons notre RPI afin qu'il puisse se connecter au réseau local de l'IUT (LAN) via l'adresse IP que notre VM lui aura attribuée. Nous allons aussi configurer cette VM pour permettre au RPI d'accéder aux réseaux extérieurs à celui de l'IUT (WAN).

Le RPI sera accessible via une connexion SSH depuis le LAN de l'IUT et pourra être utilisé comme une machine normale du réseau avec un accès à l'extérieur.

En supplément de l'accessibilité du RPI sur le réseau, un capteur sera branché à ses pins afin de récupérer la température et l'humidité ambiantes. Un programme Python lira et affichera les données reçues du capteur sur la sortie standard de la machine ayant initiée la connexion SSH sur le RPI.

2.2 Schématisation de notre solution

2.2.1 Infrastructure réseau résumée

Le RPI est ici branché sur l'interface Passthrough1 de la VM, pas de restriction quant au masque de sous-réseau ou les adresses choisies car étant notre propre réseau local. L'adresse de la VM sera considérée comme celle du routeur par les équipements de ce LAN (192.168.0.0/24 dans notre cas).

L'interface Bridge1 est reliée au réseau de l'IUT avec une adresse IP statique définie sur la plage d'adresses IP que le switch principal de l'IUT nous accorde. Ici en exemple .4 la plage d'adresse étant de 10.2.18.4 à 10.2.18.7 sur le poste TITI de la salle TP Réseaux sur lequel la configuration a été faite.

Le NAT sera aussi activé sur l'interface Bridge1 de notre VM pour que le RPI puisse communiquer avec le réseau le LAN de l'IUT (10.2.18.0/24) et le WAN.

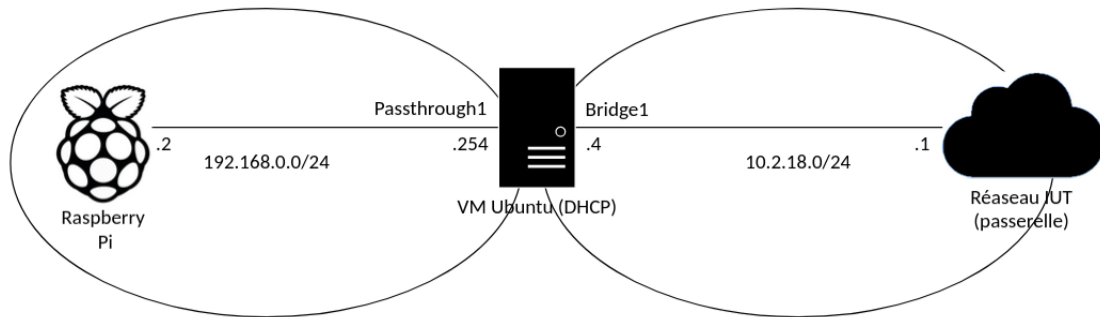


FIGURE 1 – Diagramme réseau simplifié de notre installation

2.2.2 Illustration connexion du capteur au Raspberry Pi

Le PIN le plus à gauche du capteur vu de haut est appelé VCC, c'est-à-dire l'alimentation en +5V que va lui fournir le pin numéro 2 dit 5V PWR du RPI.

Le deuxième pin du capteur est celui des données DATA rattaché à une résistance pull-up de 10 K Ohms dans notre cas d'usage (10 K Ohms étant le maximum, 4.7 K Ohms étant le minimum possible).

Ce deuxième pin est aussi placé sur le pin GPIO 4 afin de récupérer les données, à noter qu'il peut être placé sur n'importe quel pin GPIO du RPI.

Le quatrième est dernier pin du capteur est celui de la masse GND rattaché à celui du RPI.

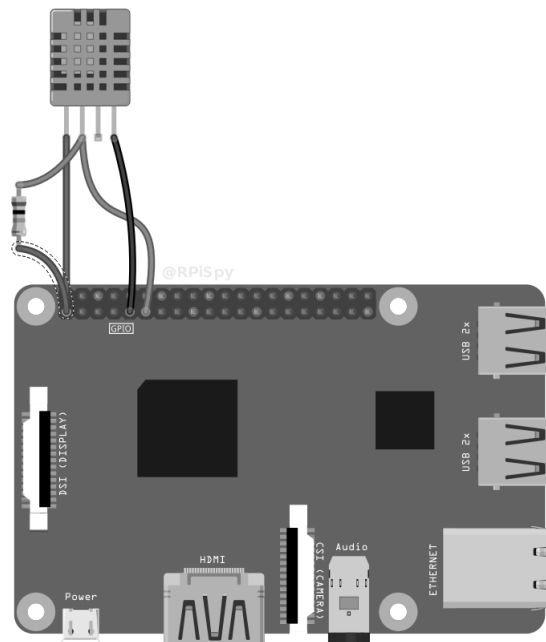


FIGURE 2 – Schéma des branchements du capteur DHT20 sur le RPI

3 Préparation du Raspberry Pi

3.1 Mise en place du SE

3.1.1 Récupération de l'image du SE

Nous allons récupérer la dernière image en date du système d'exploitation propriétaire de Raspberry pour ses Raspberry Pi : Raspberry Pi OS. Nous utiliserons sa version Lite, ne contenant pas d'environnement de bureau, elle utilisera donc moins de stockage et de ressources matérielles (ne contenant pas les paquets pour et ne l'exécutant pas).

```
$ wget https://downloads.raspberrypi.org/raspios_lite_armhf/images/raspios_lite_armhf-2022-09-26/2022-09-22-raspios-bullseye-armhf-lite.img.xz
```

3.1.2 Découverte de la location du périphérique de stockage

Pour mettre le système d'exploitation sur la carte micro SD du RPI par la suite, nous devons connecter sa carte micro SD sur notre machine via l'adaptateur.

Après cela nous devons trouver l'emplacement de la carte micro SD sur notre machine ainsi que son nom de lecteur.

(les commandes précédées d'un hashtag et marquée en rouge doivent être lancées avec les droits administrateurs)

```
# lshw -C disk
```

Voici un exemple de sortie de cette commande.

```
CHANGER SORTIE DE COMMANDE LSHW
```

3.1.3 Flash de l'image du SE sur la carte micro SD

Nous avons l'image à "flasher" sur la carte micro SD ainsi que l'emplacement du périphérique de stockage. Nous devons maintenant écrire le contenu du fichier récupéré sur la carte micro SD.

Le premier argument de la commande est l'emplacement du fichier lu, dans cette démonstration elle se situe à l'emplacement où le terminal a été lancé.

La carte micro SD a comme emplacement /dev/sda vu 3.1.2.

```
# xzcat 2022-09-06-raspios-bullseye-armhf-lite.img.xz | sudo dd  
bs=4M of =/dev/sda
```

3.2 Configuration du SE avant déploiement

3.2.1 Création d'un utilisateur et d'un mot de passe

Afin de créer un utilisateur et un mot de passe personnalisés sur notre RPI, nous devons modifier un fichier de configuration du système d'exploitation du RPI avant son déploiement.

Pour cela nous devons d'abord débrancher et rebrancher l'adaptateur sur la machine afin que la carte micro SD soit reconnue comme un dispositif de stockage utilisable par le SE que nous utilisons. Dans notre cas la carte micro SD est reconnue avec comme chemin d'accès **/run/media/alexis/**, "alexis" étant le nom d'utilisateur de ma session. Elle peut être observée dans l'explorateur de fichier de votre SE.

Nous devons ensuite modifier le fichier se trouvant dans /boot/userconf.txt sur la carte micro SD, le chemin absolu de ce fichier dans mon cas est donc **/run/media/alexis/boot/userconf.txt**.

Pour éviter de fournir le mot de passe en clair dans le fichier de configuration nous allons utiliser la commande **openssl** pour générer un mot de passe chiffré étant **"mypassword"** et l'attribuer à l'utilisateur **"alexispi"** tout cela dans le fichier **userconf.txt**.

```
$ echo "alexispi:(echo 'mypassword' | openssl passwd -6 -stdin)"  
> /run/media/alexis/boot/userconf.txt
```

3.2.2 Configuration réseau du SE

La configuration réseau initiale de Raspberry Pi OS repose sur l'attente d'une réponse DHCP sur l'interface RJ45 Ethernet.

Cependant si vous souhaitez configurer le RPI sur son interface Wi-Fi IEEE lors son premier démarrage vous pouvez configurer le fichier **etc/wpa_supplicant/wpa_supplicant.conf** présent sur la carte micro SD.

Dans notre cas d'usage, nous laissons le RPI se connecter sur le LAN de l'IUT via son interface RJ45 Ethernet, nous utiliserons une VM faisant notamment office de serveur DHCP pour lui attribuer une adresse.

Pas de manipulation à faire dans notre cas.

3.2.3 Activation du service SSH

Nous initierons par la suite une connexion SSH (

4 Déploiement du serveur DHCP

4.1 Explications sur notre serveur DHCP

4.2 Création de la machine virtuelle

4.3 Configuration de l'interface côté réseau de l'IUT

4.3.1 Attribution d'une adresse IP statique à l'interface

4.3.2 Reinseignement de l'adresse IP du serveur DNS

4.4 Installation du service DHCP

4.4.1 Mise à jour des paquets du système

dire pb libappstream4 aucasou

4.4.2 Installation d'un paquet de service de DHCP

4.5 Configuration de l'interface côté appareils

4.5.1 Sauvegarde de l'ancienne configuration de l'interface

4.5.2 Sauvegarde et modification du fichier de configuration DHCP

4.5.3 Vérification de l'état du service réseau

4.5.4 Vérification de l'état du service DHCP

5 Configuration d'un NAT

5.1 Explication de l'intérêt du NAT

5.2 Activation du NAT

5.3 Configuration du routeur NAT

5.3.1 Configuration de l'interface pour accepter le NAT

5.4 Contrôles supplémentaires après configuration

5.4.1 Vérification des routes IP

5.4.2 Initialisation d'une connexion SSH

5.5 Script d'automatisation optionnel

5.5.1 Script shell de configuration des services

6 Utilisation du capteur DHT20

6.1 Explications sur l'utilisation du capteur

6.2 Branchement sur le Raspberry Pi

6.2.1 Schéma de branchement

6.2.2 Photographie du branchement

6.3 Programme Python pour utiliser le capteur

```
import time, board, adafruit_dht

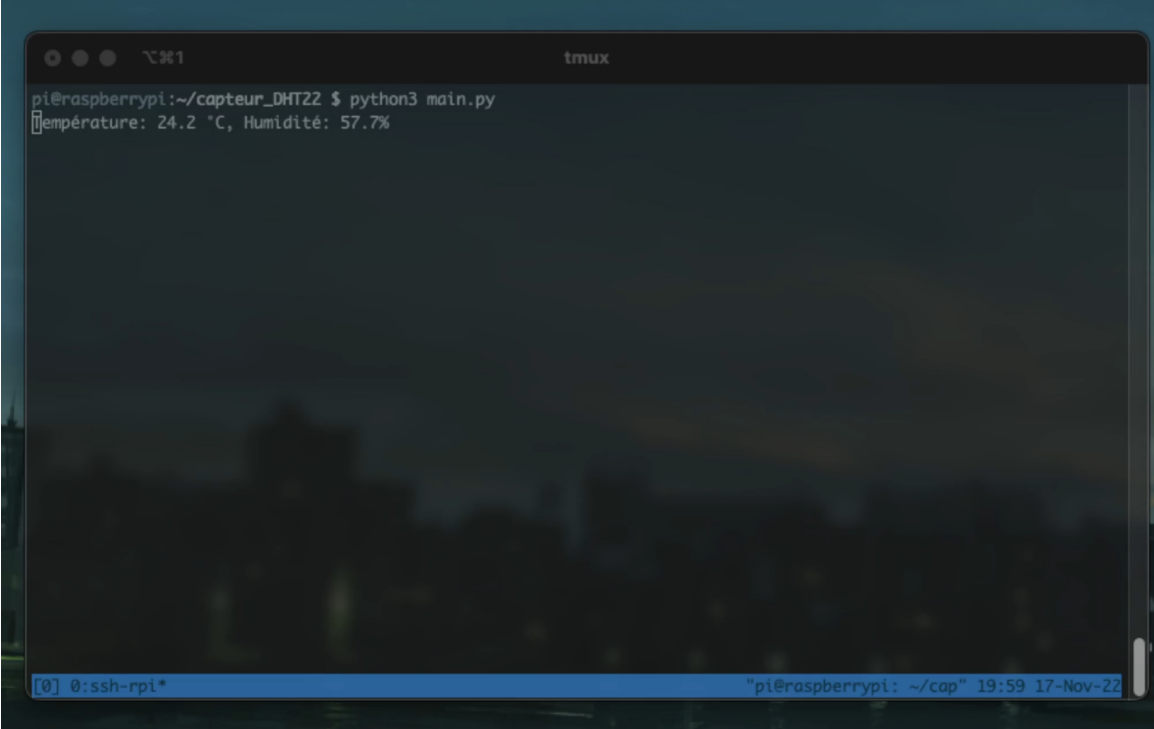
dht_device = adafruit_dht.DHT22(board.D4) # Charge le capteur sur le PIN GPIO 4

while True:
    try:
        temperature, humidity = dht_device.temperature, dht_device.humidity
        # Recupere les donnees du capteur
        print("Temperature: " +str(temperature) + " C, Humidite: " +str(humidity) + "%", end="\r")
        # Affiche ces donnees dans la console
    except RuntimeError:
        # Il arrive que les donnees du capteur arrivent dans le Raspberry en etant corrompues, on ignore
        # donc cette erreur

        time.sleep(2.0) # Pause de 2 secondes
        continue # On continue la boucle
    except Exception as error:
        # Cette erreur est consideree comme grave, elle n'est pasensee arriver mais protection par
        # precaution
        dht_device.exit() # Ferme proprement la liaison capteur-programme si erreur
        raise error # Laisse l'erreur interrompre le programme
    time.sleep(2.0) # Raffraichissement toutes les 2 secondes
```


6.4 Exemple de sortie du programme Python

Temperature: 24.2 C, Humidite: 57.7%



The image shows a terminal window titled 'tmux' with a session identifier '1'. The prompt is 'pi@raspberrypi:~/capteur_DHT22 \$'. The command 'python3 main.py' has been executed, resulting in the output 'Température: 24.2 °C, Humidité: 57.7%'. The terminal window has a dark background and a blue status bar at the bottom. The status bar on the left shows '[0] 0:ssh-rpi*' and on the right shows '"pi@raspberrypi: ~/cap" 19:59 17-Nov-22'.

```
pi@raspberrypi:~/capteur_DHT22 $ python3 main.py
Température: 24.2 °C, Humidité: 57.7%
```