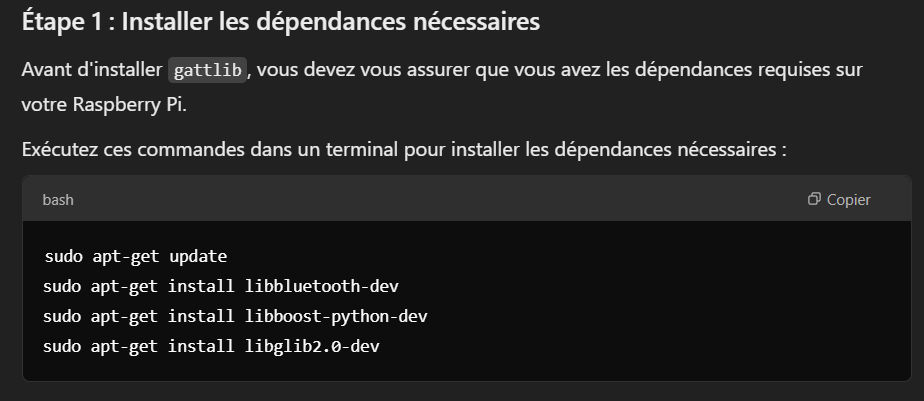
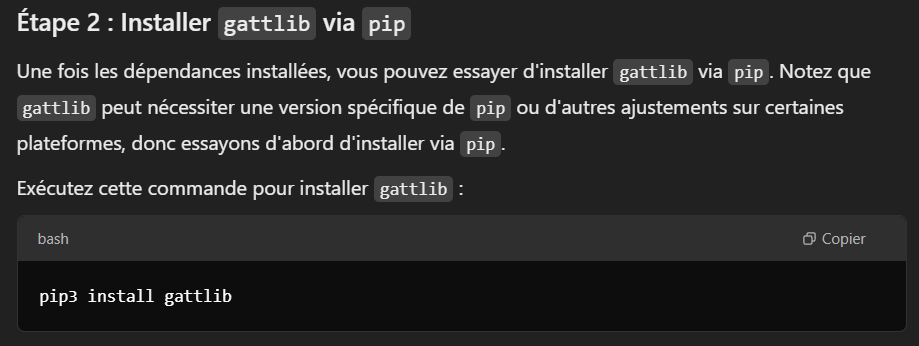
# Historique des recherches

<https://github.com/scivision/pybluez-examples/blob/main/bluetooth_scan.py>

<https://pybluez.readthedocs.io/en/latest/api/discover_devices.html>





Installation de gattlib (pour detecter les appareil bluetooth ble) :

<https://stackoverflow.com/questions/43721463/cant-install-gattlib-on-raspberry-pi-error-stray-xxx-in-program-on-usr-inc>

# Mise en place d’une machine virtuelle sur Ubuntu (Linux)

J’ai choisi de mettre en place un machine virtuelle Ubuntu (Linux) sur ma machine personnelle afin de tester l’execution de package python et des outils spécifiques à des environements Linux. Pour cela je me suis aider de cours OpenClassroom pour me guider dans la mise en place et le parametrage de la amchine virtuelle.

Une fois inistallée, j’ai pu utiliser un environement Linux.

J’y ai ensuite installer python3, pycharm et un broker mosquitto afin de realiser des tests pour gagner du temps lors des sessions sur le Raspberry PI.

Sitographie :

Installation et deploiement de mosquitto sur Ubuntu : (cmd similaires au raspberry PI)

<https://docs.vultr.com/install-mosquitto-mqtt-broker-on-ubuntu-20-04-server>

Use paho mqtt with python :

<https://www.hivemq.com/blog/mqtt-client-library-paho-python/>

Utilisation de loop\_start & loop\_stop :

Permet de gerer les msg, la connexio au topic et tout le toutim relatif à MQTT de facon asynchrone avec le code python => le code python va continuer de s’executer sans se preoccuper des informations relatives à MQTT (sauf erreur !)

# Installation d’un environement virtuel Windows

Pourquoi faire ? Nous essayons d’utiliser DeepFace qui est une bibliotheque python permettant de faire de la reconnaissance faciale. Cependant cette bibliotheque se base en partie sur TensorFlow, une autre bibliotheque python. Cette derniere n’est pas supportee sur python 3.13, c’est pourquoi nous devons regresser notre version de python afin de pouvoir installer DeepFace. La création d’un environement virtuel devient alors nécéssaire pour l’execution du code et l’installation des package sans boulverser les environements déjà existant en 3.13.

1.Installation d’un python inférieur (compatibilité DeepFace)

2.Check des versions de python installée : py -0

3.a. Se placer dans le dossier où l’on souhaite créer l’environement virtuel : cd C:/environements\_python

3.b.Création d’un venv python : py -3.12 -m venv deepface\_env

4.Activation de l’environement virtuel : deepface\_env\Scripts\activate

5.installation des package :

- pip install deepface opencv-python

- pip install pillow

- paho-mqtt

- tf-keras

- psycopg2

6.Désactivation de l’environement : deactivate