

# Documentation script Python

Hugo WENDJANEH, Matéo PÉPIN, Tilian HURÉ  
et Vincent MIQUEU-DENJEAN (**G2A-**)

## Structure du code

### Principales fonctions

- `get_config()` : lit le fichier de configuration `config.json`
- `write_data(data)` : écrit les données récupérées via le paramètre `data` dans le fichier JSON `data.json`
- `on_connect(client, userdata, flags, rc)` : abonne le client donné à un flux lors de la connexion (*fonction surchargée*)
- `on_message(client, userdata, msg)` : convertit les données reçues du payload (`msg.payload`) dans une variable `payload` au format JSON (*fonction surchargée*)
- `on_alarm(signum, frame)` : envoie dans la fonction `write_data` les données reçues qui correspondent à celles demandées dans le fichier de configuration et indique si elles dépassent le seuil renseigné lors de la réception d'un signal (*fonction surchargée*)

### Fonctionnement

Le script abonne le client au réseau *chirpstack* de l'IUT (fonction `on_connect`) afin récupérer les données depuis les capteurs physiques. Il envoie ensuite un signal périodique **SIGALRM** afin de déclencher la fonction `on_alarm` qui s'appellera en boucle. Cette dernière va récupérer les données du fichier de configuration `config.json` via la fonction `get_config`, ainsi que les données du réseau MQTT et les réécrire dans un fichier `data.json` en fonction des spécificités renseignées (données à prendre en compte et seuils).

Peu importe le fichier de configuration (fichier mal-formé voire absent), le script crée toujours un fichier de données au format JSON.

## Installation

Le script nécessite au préalable d'avoir installé **Python3** sur le poste concerné, ainsi que les librairies Python suivantes :

- `sys` : manipulation de l'environnement d'exécution Python (path, nom du fichier exécuté, paramètres, etc)
- `os` : manipulation du système d'exploitation du poste (lecture de fichiers, écriture, etc)
- `json` : manipulation de fichiers JSON (lecture, écriture, conversion en dictionnaire, etc)
- `paho.mqtt.client` (mqtt) : connexion à un réseau MQTT (celui des capteurs de l'IUT)
- `signal` : envoi de signaux à des processus

## Lancement du code

Le script peut être lancé dans un terminal via les commandes suivantes (si l'une ne marche pas, essayer l'autre) et sans paramètre :

```
python script.py
```

```
python3 script.py
```