

# Manuel d'installation

JUMEAU NUMÉRIQUE DANS  
L'INDUSTRIE DU FUTUR  
**L2G1-2021**



# Manuel d'installation

## PROJET JUMEAU NUMÉRIQUE

### Identification du document

<b>Référence du document :</b> L2G1.I
<b>Version du document :</b> 1.3
<b>Date du document :</b> 10/04/2022
<b>Auteurs :</b> Wattrelos Tigran Boka Ricardo Abou Assaf Mahyr-Florian Daumont-Ouk Ilan'

# Sommaire

<b>Introduction</b>	<b>4</b>
<b>Installation du matériel côté serveur</b>	<b>5</b>
Installation de InfluxDB	5
Installation de Telegraf	6
Installation de Grafana	8
Configuration de la messagerie (alerting)	9
Installation des scripts fourni par le groupe	9
<b>Installation du matériel côté raspberry</b>	<b>10</b>
Installation du système d'exploitation	10
Première connexion	11
Configuration matérielle et logicielle	12
Paramétrer le programme	16
Côté Python	16
Côté Java	17
<b>Sources</b>	<b>18</b>
InfluxDB/Grafana	18
Raspberry	18

# Introduction

Un Digital Twin se compose en deux parties: le côté client et le côté serveur.

Contrairement au développement côté client, tout ce qui concerne le côté serveur est invisible pour l'utilisateur et comprend les composants suivants :

- Le **langage de programmation** est utilisé pour écrire le code permettant de faire le lien entre le serveur, le Raspberry Pi et la base de données.
- Le **serveur** gère les données qu'il reçoit du Raspberry Pi, et l'affichage des données du tableau de bord Grafana.
- La **base de données** est utilisée pour stocker et archiver les données des capteurs, grâce à InfluxDB/Prometheus.
- Le **broker de messages** permet la communication des valeurs relevées par les capteurs entre le raspberry et le serveur

Par la suite, nous vous montrerons comment installer Grafana, Influx DB (ainsi que ses applications nécessaires au bon fonctionnement Telegraf, MQTT) et à les configurer avec le Raspberry Pi Zero.

Dans le dossier L2g1\_serveur sont fournis le code permettant de faire fonctionner le côté serveur, une Image de Raspberry Pi avec un Sense Hat-B et un fichier de configuration permettant de paramétrer le script avec les valeurs souhaitées.

Dans le dossier L2g1\_raspberry sont fournis le code python permettant de faire fonctionner le Raspberry Pi et un fichier de configuration permettant de paramétrer le script avec les valeurs souhaitées.

Ces deux dossiers sont accompagnés des scripts permettant la réalisation des tests.

# Installation du matériel côté serveur

- Installer curl :

```
- $ sudo apt install curl
```

## 1. Installation de InfluxDB



- Sur une machine sous Linux, installer InfluxDB grâce à cette commande :

```
- $ sudo apt-get install influxdb influxdb-client
```

- Il faut maintenant exécuter un ensemble de commandes pour lancer le service :

```
- $ sudo systemctl unmask influxdb.service
```

```
- $ systemctl start influxdb
```

- Dans le shell, pour lancer la base de données, écrire : "influx".

Pour pouvoir collecter les données, nous allons configurer InfluxDB :

- Créer une nouvelle base de données avec l'utilisateur "telegraf" et le mot de passe "pass-telegraf" :

```
- $ create user telegraf with password 'pass-telegraf'
```

- Remplacer " telegraf " et " pass-telegraf " par le nom de user et le mot de passe souhaité

InfluxDB est installé et fonctionnel sur votre environnement, et accessible sur le port 8086.

## 2. Installation de Telegraf



Telegraf permet de collecter les métriques et de les envoyer à InfluxDB.

- Mettre les paquets à jour:

```
$ sudo apt update && sudo apt -y upgrade
```

- Télécharger les paquets Telegraf :

```
- $ sudo apt install -y gnupg curl wget  
$ wget -qO- https://repos.influxdata.com/influxdb.key  
| sudo apt-key add -
```

- Ajouter le dépôt InfluxDB APT en utilisant la commande ci-dessous :

```
- $ echo "deb https://repos.influxdata.com/debian buster  
stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
```

- Installer Telegraf :

```
- $ sudo apt update  
- $ sudo apt -y install telegraf
```

- Vérifier le service Telegraf :

```
- $ sudo systemctl status telegraf
```

- Si tout fonctionne cela devrait s'afficher et le système est noté actif :

- telegraf.service - The plugin-driven server agent for reporting metrics into InfluxDB  
 Loaded: loaded (/lib/systemd/system/telegraf.service; enabled; vendor preset: enabled)  
 Active: active (running) since Sun 2021-06-20 11:31:22 CEST; 13s ago  
 Docs: <https://github.com/influxdata/telegraf>  
 Main PID: 9466 (telegraf)  
 Tasks: 10 (limit: 4915)  
 Memory: 26.9M  
 CGroup: /system.slice/telegraf.service  
         └─9466 /usr/bin/telegraf -config /etc/telegraf/telegraf.conf -config-directory /etc/

- Accéder au fichier de configuration Telegraf :

```
- $ sudo nano /etc/telegraf/telegraf.conf
```

- Modifier dans le fichier le morceau ci-dessous. En modifiant le username et le mot de passe par ceux défini dans la partie d'installation d'InfluxDB. Remplacer également "database-name" par le nom de base de donnée voulu et urls par "[<http://influxdb:8086>]" :

```
#####
#                                OUTPUTS                                #
#####

[[outputs.influxdb]]
  urls = ["http://influxdb-ip:8086"]
  database = "database-name"
  timeout = "0s"
  username = "auth-username"
  password = "auth-password"
  retention_policy = ""
```

- Redémarrer Telegraf :

```
- $ sudo systemctl restart telegraf
```

### 3. Installation de Grafana



Grafana est un tableau de bord qui permet de visualiser l'affichage de données en temps réel.

- Vérifier la dernière version de Grafana grâce à cette suite de commandes :

```
- $ sudo wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add  
- $ sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable  
main"  
- $ sudo apt update  
- $ sudo apt-cache policy grafana
```

- Effectuer son installation :

```
- $ sudo apt install grafana
```

- Lancer le serveur Grafana, et activer le lancement automatique :

```
- $ sudo systemctl start grafana-server  
- $ sudo systemctl enable grafana-server
```

- Installer le plug-in du panneau Worldmap, puis redémarrer le service :

```
- $ grafana-cli plugins install grafana-worldmap-panel  
- $ sudo systemctl restart grafana-server
```

- Ouvrir un navigateur web puis entrer l'adresse "localhost:3000" pour accéder à la page de connexion du site Grafana.



Par défaut se connecter avec les identifiants user: "pi", mdp : "raspberry".  
Pour une question de sécurité, il est vivement conseillé de modifier votre mot de passe.

- Passer votre souris sur le "+" à gauche de l'écran puis sur "Import".
- Dans le panneau "import via panel json", copier-coller le fichier json "dashboard.json" fourni avec ce manuel.

### Configuration de la messagerie (alerting)

Pour l'envoi d'alertes il faut mettre en place une méthode nous permettant de les transférer par mail. Nous allons configurer un protocole smtp qui est une solution répondant parfaitement à ce cas là.

- Entrer dans le fichier grafana.ini :

**- \$ sudo nano /etc/grafana/grafana.ini**

- Rechercher « smtp » avec combinaison « ctrl + w »
- Configurer la section [smtp] comme ci-dessous:  
(ici nous utilisons le smtp gmail)

```
##### SMTP / Emailing #####
[smtp]
enabled = true
host = smtp.gmail.com:587
user = inserer_adresse_mail@gmail.com
# If the password contains # or ; you have to wrap it with triple quotes. Ex "
password = mot_passe_adresse_mail
cert_file =
key_file =
skip_verify = false
from_address = inserer_adresse_mail@gmail.com
from_name = Grafana
# EHLO identity in SMTP dialog (defaults to instance_name)
ehlo_identity =
# SMTP startTLS policy (defaults to 'OpportunisticStartTLS')
#startTLS_policy = NoStartTLS

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^J Justifier
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^T Orthograp.
```

- Enregistrer, quitter et redémarrer le serveur grafana:

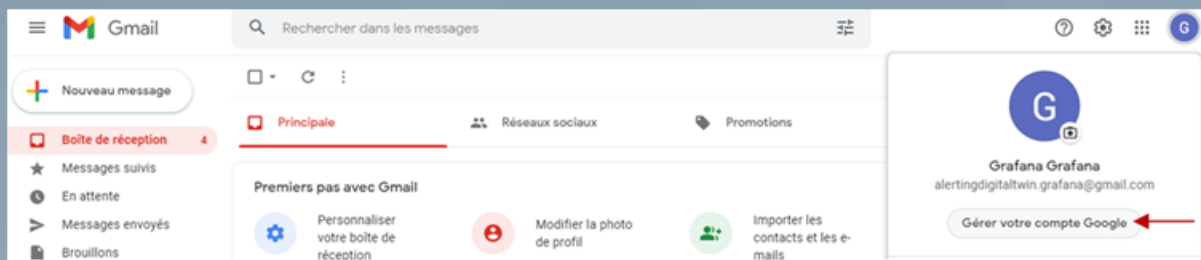
**- \$ sudo systemctl restart grafana-server**

La configuration smtp est complète, l'adresse mail est reliée à l'alerting.

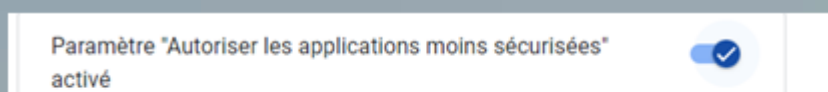
NB: Il se peut que des erreurs surviennent lors de la configuration des points de contact (voir manuel d'utilisation). Dans le cas du smtp gmail, il s'agit d'un paramètre de sécurité qui empêche l'accès d'autres applications au transfert de mails.

Suivre les étapes suivantes pour activer l'accès aux applications moins sécurisées:

- Ouvrir Gmail:



Gérer votre compte Google -> Sécurité -> Accès aux applications moins sécurisées -> activer



La configuration des points de contact ne devrait plus indiquer d'erreur serveur smtp.

#### 4. Installation des scripts fourni par le groupe

Télécharger sur votre serveur le dossier " L2g1\_serveur " fourni par le groupe sur le répertoire personnel.

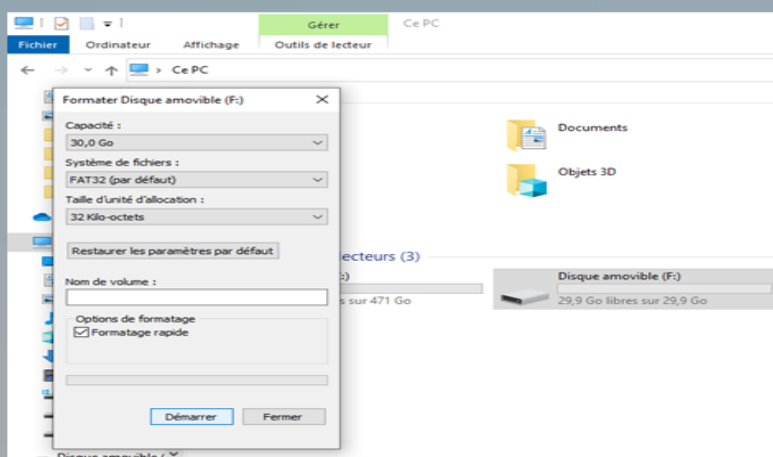
- Ajouter l'image du raspberry dans le bon dossier

```
- $ mv ~/L2g1_serveur/Image/RPISENSE.png  
/usr/share/grafana/public/img
```

# Installation du matériel côté raspberry

## Installation du système d'exploitation

- Retirer la carte SD du Raspberry puis l'insérer dans un ordinateur
- Formater la carte SD avant l'installation du système, pour cela :
  - Aller sur Ce PC
  - Cliquer droit sur l'icône représentant le stockage carte sd
  - Cliquer sur formater
  - Laisser les paramètres de formatage par défaut et exécuter le formatage



- Pour procéder à l'installation du système, télécharger Raspberry Pi Imager sur le site: <https://www.raspberrypi.com/software/>
- Installer un OS compatible avec le Raspberry pi zero sur la carte sd avec Raspberry Pi Imager (Raspberry Pi OS Lite 32 bits est vivement recommandé pour ce modèle).
- Après l'installation si le logiciel éjecte la carte SD, la retirer puis la réinsérer dans l'ordinateur pour configurer une première connexion en ssh à votre machine.

## Première connexion

- Créer un fichier nommé "ssh", veiller à ne pas donner d'extension au fichier ("exemple.txt").
- Créer un fichier nommé "wpa\_supplicant.conf"
- Entrer dans le fichier wpa\_supplicant.conf pour configurer votre réseau comme ci-dessous:

```
country=FR # Le code 2-digit de votre pays  
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev  
update_config=1
```

```
network={  
    ssid="nom de votre réseau"  
    psk="mot de passe du réseau"  
    key_mgmt=WPA-PSK  
}
```

- Enregistrer et quitter le fichier.
- Éjecter la carte SD.
- Insérer la carte SD dans le raspberry pi.
- Se munir d'un câble usb/micro-usb pour alimenter le raspberry pi à un ordinateur.
- Ouvrir un terminal de commande.
- Saisir dans le terminal la commande `$ ssh pi@raspberrypi` pour se connecter au rpi.

- Par défaut se connecter avec les identifiants suivants :
  - User : "pi"
  - MDP : "raspberry"(Pour une question de sécurité, il est vivement conseillé de modifier votre mot de passe)

```
- $ sudo passwd
```

Vous venez à présent de vous connecter pour la première fois au raspberry. Par précaution, créer à nouveau un fichier "wpa\_supplicant.conf" avec les même configurations réseaux dans le répertoire "/etc/wpa\_supplicant/" afin de conserver nos paramètres de connexions:

```
- $ sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

## Configuration matérielle et logicielle

Nous allons ici importer le dossier 'Raspberry' dans le raspberry pi pour l'envoi des relevés capteurs au serveur et installer les modules essentiels au bon fonctionnement des capteurs du Sense Hat (B).

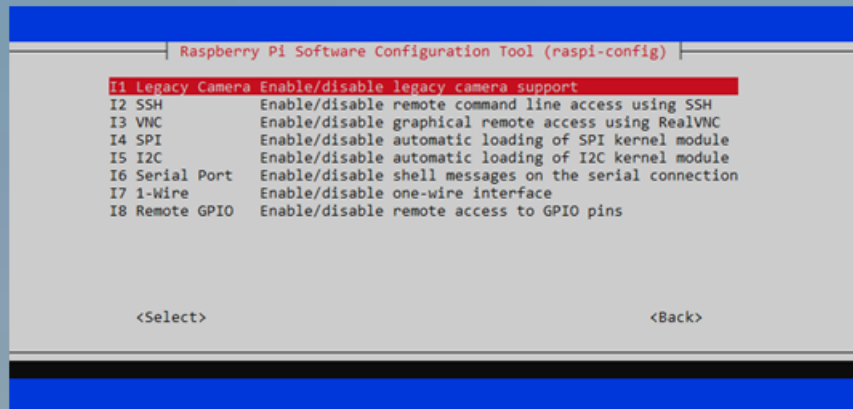
Veiller à ce que le Sense Hat soit branché au raspberry de la manière suivante:



- Pour activer le bus i2c et s'assurer que ssh est bien actif, entrer dans l'outil de configuration logicielle raspberry pi :

```
- $ sudo raspi-config
```

### 3 Interface Options -> I2C -> Enable -> SSH -> Enable



- S'assurer que l'I2C est activé en entrant dans le fichier "config.txt":

- `$ sudo nano /boot/config.txt`

- Vérifier que le fichier contient la ligne : "dtparam=i2c\_arm=on" (si celle-ci est commentée "#", il faut la décommenter)
- Vérifier les adresses des capteurs avec la commande :

- `$ sudo i2cdetect -y 1`

```
pi@raspberrypi:~ $ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- 29 -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- 48 -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- 5c -- -- --
60:  -- -- -- -- -- -- -- 68 -- -- -- -- -- --
70:  70 -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

Adresse du capteur de température et d'humidité: 0x70

Adresse du capteur de pression: 0x5c

Si les adresses des capteurs sont indiquées alors i2c est bien actif et les capteurs sont reconnus.

- Redémarrer le raspberry pi :

- `$ sudo reboot`

***Installer les paquets ci-dessous pour assurer une bonne fonctionnalité du programme “envoi\_capteurs.py” :***

Si python3.8 et le gestionnaire de paquets python3 (pip3) ne sont pas installés par défaut, saisir les lignes de commande ci-dessous:

- python3.8 : `$ sudo apt-get install python3.8`
- pip3 : `$ sudo apt-get install python3-pip`

Librairie Adafruit pour les paquets de capteurs et modules contrôle bus:

- `$ sudo pip3 install --upgrade Adafruit-PlatformDetect`
- `$ sudo pip3 install --upgrade adafruit-blinka`
- `$ sudo pip3 install adafruit-circuitpython-shtc3`
- `$ sudo pip3 install adafruit-circuitpython-lps2x`

Module de gestion de temps / horloge serveur NTP :

- `$ sudo pip3 install ntpdatetime`

Module de conversion de données de localisation au format geohash :

- `$ sudo pip3 install python-geohash`

Module client avec prise en charge MQTT :

- `$ sudo pip3 install paho-mqtt`

Le raspberry est à présent bien configuré et le programme “envoi\_capteurs.py” devrait pouvoir fonctionner correctement.

Télécharger le dossier “L2g1\_raspberry” fourni par le groupe dans votre ordinateur.

Depuis une nouvelle fenêtre terminal, copier le dossier avec les programmes python de votre pc (source) vers le raspberry (cible):

```
- $ scp -r C:\Users\<nom d'utilisateur>\<répertoire où se situe le  
programme dans votre pc>\L2G1 Program\L2g1_raspberry  
pi@raspberrypi:~/
```

Une fois la copie effectuée, revenir sur le terminal du raspberry pi.  
L'exécution du programme s'effectue avec la commande suivante:

```
- $ sudo python3 envoi_capteurs.py
```

Après exécution, le client parvient à se connecter au serveur et publie les données, le programme devra afficher les messages suivants :

```
pi@raspberrypi:~/L2g1_raspberry/Capteurs_python $ sudo python3 envoi_capteurs.py  
Connecté  
Envoi des données en cours:
```

NB : En cas d'erreur de type " attributeError: module 'board' has no attribute 'I2C' " avec le module board, vous pouvez la résoudre en effectuant une mise à niveau des librairies adafruit avec la commande suivante :

```
- $ sudo pip3 install --upgrade --force-reinstall adafruit-blinka  
Adafruit-PlatformDetect
```

## Démarrage automatique du Script python

Le script peut être lancé automatiquement au démarrage. Pour mettre en place cela, suivre les étapes suivantes :

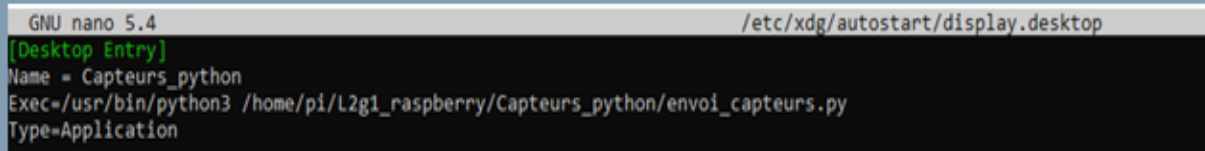
Si vous souhaitez que le programme s'exécute automatiquement au démarrage du raspberry, suivre les étapes suivantes:



Créer le fichier display.desktop dans répertoire /etc/xdg/autostart:

```
- $ sudo nano /etc/xdg/autostart/display.desktop
```

Configurer le fichier de la manière suivante:



```
GNU nano 5.4 /etc/xdg/autostart/display.desktop
[Desktop Entry]
Name = Capteurs_python
Exec=/usr/bin/python3 /home/pi/L2g1_raspberry/Capteurs_python/envoi_capteurs.py
Type=Application
```

Enregistrer (“ctrl +o”) et quitter (“ctrl + x”)

Lorsque le raspberry pi va redémarrer, le programme devrait s’exécuter automatiquement :

```
- $ sudo reboot
```

Pour vérifier que l’exécution automatique du programme soit fonctionnel, regarder sur le dashboard grafana si de nouvelles données sont récupérées.

## Paramétrer le programme

Des fichiers de paramétrages sont mis à disposition dans le dossier du programme

### Côté Python

Nom du fichier : parametrages.py

Pour y accéder, saisir la commande suivante:

```
- $ sudo nano parametrages.py
```

```
# nom des topics
topic_temperature = "l2g1_temperature"
topic_humidite = "l2g1_humidite"
topic_pression = "l2g1_pression"
topic_coordonnees = "l2g1_coordonnees"
topic_test = "l2g1_test"

# délai d'envoi des données (en seconde)
temps = 1

# connexion au broker
url = "broker.hivemq.com"
port = 1883
```

*contenu du fichier de configuration*

Voici les différents paramètres:

topic\_temperature : Le topic auquel se connecter pour envoyer les données de température

topic\_humidite : Le topic auquel se connecter pour envoyer les données d'humidité

topic\_pression : Le topic auquel se connecter pour envoyer les données de pression

topic\_coordonnees : Le topic auquel se connecter pour envoyer les données de géolocalisation

topic\_test : Le topic auquel se connecter pour les phases de test.

temps : Le temps entre deux envois de données par le raspberry

url : L'url du Mqtt auquel se connecter

port : Le port du Mqtt auquel se connecter

## Côté Java

nom du fichier : ConfigJava

```
ipServeur = localhost
portInflux = 8086
portMqtt = 1883
lienMqtt = tcp://broker.hivemq.com
topic_temperature = l2g1_temperature
topic_humidite = l2g1_humidite
topic_pression = l2g1_pression
topic_coordonnees = l2g1_coordonnees
topic_test = l2g1_test
database = metrics
userInflux = telegraf
mdpInflux = password
```

*contenu du fichier de configuration*

Voici les différents paramètres:

ipServeur : L'ip du serveur de Influxdb

portInflux : Le port sur lequel est connecté InfluxDB

portMqtt : Le port où est connecté le Mqtt

lienMqtt : Le lien du Mqtt auquel se connecter

topic\_temperature : Le topic auquel se connecter pour recevoir les données de température

topic\_humidite : Le topic auquel se connecter pour recevoir les données d'humidité

topic\_pression : Le topic auquel se connecter pour recevoir les données de pression

topic\_coordonnees : Le topic auquel se connecter pour recevoir les données de géolocalisation

topic\_test : Le topic auquel se connecter pour les phases de test

database : Le nom de la base de données où seront stocké les métriques ( elle se crée automatiquement )

userInflux : Le nom de l'utilisateur qui se connectera à InfluxDB

mdpInflux : Le mot de passe de l'utilisateur InfluxDB associé

# Sources

## InfluxDB/Grafana

- <https://nwmichl.net/2020/07/14/telegraf-influxdb-grafana-on-raspberrypi-from-scratch/>
- <https://grafana.com/grafana/plugins/grafana-worldmap-panel/?tab=installation>
- <https://static.cinay.eu/2021/06/20/InfluxDB-Telegraf-Grafana.html>

## Raspberry

- <https://raspberry-lab.fr/Tutoriels/Installer-Raspberry-Pi-OS-sur-carte-SD/>
- <https://raspberry-lab.fr/Tutoriels/Installer-Raspberry-Pi-OS-sur-carte-SD/#config-rapide>
- <https://www.raspberrypi-spy.co.uk/2014/11/enabling-the-i2c-interface-on-the-raspberry-pi/>
- <https://pypi.org/project/Adafruit-Blinka/>
- <https://learn.adafruit.com/adafruit-sensirion-shtc3-temperature-humidity-sensor/python-circuitpython>
- <https://learn.adafruit.com/adafruit-lps25-pressure-sensor?view=all#python-installation-of-lps2x-library-3056734-6>
- <https://pypi.org/project/ntpdatetime/>
- <https://pypi.org/project/paho-mqtt/>
- <https://github.com/vinsci/geohash#:~:text=README.rst-,Geohash,from%20latitude%20and%20longitude%20coordinates.&text=The%20last%20two%20values%20are,the%20latitude%20and%20longitude%20respectively.>
- <https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/faq-troubleshooting>