Une image contenant texte, capture d’écran

Description générée automatiquement**Script principal**

**Calcul du *POINT DE ROSEE***

On a utilisé la fonction read\_sensor() proposée.

***Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement***

**Script :**

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

* Il y a deux variables :  **COEFF1** (β) et **COEFF2**(γ) avec des valeurs constantes que nous utilisons dans la formule.
* On appel la fonction **read\_sensor()** pour obtenir les donnés du capteur, c’est-à-dire, la température et l’humidité relative pour assigner ces valeurs aux variables t et rh.
* On récupère la valeur de l’humidité relative du capteur et on le met dans la variable **rh\_value.**
* Pour **t\_value :** Avec ces 3 lignes de code :

Une image contenant texte, Police, capture d’écran, typographie

Description générée automatiquement

On convertit la valeur de la température en chaine de caractère pour ensuite extraire seulement la partie numérique de la valeur de la température grâce à la méthode ***split,*** et finalement on le convertit en un nombre flottant

* Pour le calcul on a partagé la grosse formule en deux parties pour que ça soit plus compréhensible. La variable ***part1*** représente la partie du numérateur et la ***partie2*** représente la partie du dénominateur. Ensuite dans la variable ***Prosee*** on fait le calcul final, donc la division de **part1/part2.**
* Finalement on imprime les valeurs de la température, humidité relative et point de rosee dans le terminal pour vérifier que tout s’est bien passé.

**Enregistrement des données dans le fichier .csv**

On a utilisé la fonction save\_to\_csv() proposée.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Dans le script nous avons ajouté quelques lignes de code pour obtenir la date et l’heure pour ensuite les enregistrer dans un fichier .csv

Une image contenant texte, capture d’écran, Police

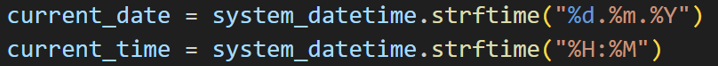
Description générée automatiquement

* La variable **system\_datetime** nous permet d’obtenir l’heure et la date actuelles du système.

**nomVariable = module.classe.méthode()**

Dans le module **datetime** il y a une classe nommé **datetime** aussi, dans la classe **datetime** il y a quelques méthodes pour gérer les dates et les heures, **now()** est une méthode qui renvoie l’heure et la date actuelles du système.

* Ces deux lignes de code :



Nous permettent de prend l’heure et la date pour les mettre au format

**jour/mois/année** et **heure/minute**

* Ensuite on imprime les valeurs de **current\_date** et **current\_time** dans le terminal.
* Cette ligne de code :



Nous permet d’enregistrer l’heure, la date, l’humidité relative et le point de rosée dans le fichier .csv placé dans le chemin : /home/debian/TempLog.csv, de cette façon lors de chaque exécution du code toutes les informations seront enregistrées dans ce fichier .csv

**Envoi de l’email**

On a utilisé la fonction send\_email() proposée.

**Une image contenant texte, capture d’écran

Description générée automatiquement**

Et nous avons rajouté quelques lignes de code pour gérer l’envoi de l’email

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Si la température est supérieure à 28°C (selon cahier de charge) un email d’alerte doit être envoyé à l’utilisateur. Dans le cas contraire si nous sommes dans la plage de sécurité, c’est-à-dire, inférieur ou égal à 28°C il n’y a pas d’email d’alerte.