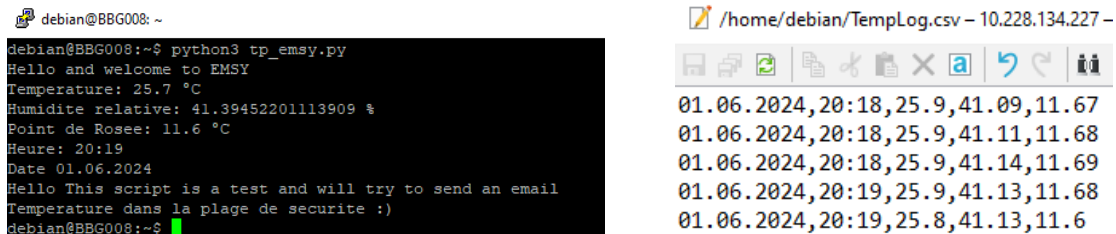


Lors de l'exécution du code il est censé envoyer un email que lorsque la température dépasse les 28°C. Dans ce cas il n'y a pas de problème, nous sommes dans la plage de sécurité et donc pas d'email d'alerte, de toute façon à chaque fois que le code est lancé toutes les informations sont enregistrées dans un fichier .csv nommé TempLog.csv dans un format :

Date/Heure/Temperature/Humidité/Point de rosée



The image shows a terminal window on the left and a CSV file view on the right. The terminal output shows a script execution where the temperature is 25.7°C, which is below the 28°C threshold, so no alert is sent. The CSV file on the right contains five rows of data, all with temperatures below 28°C.

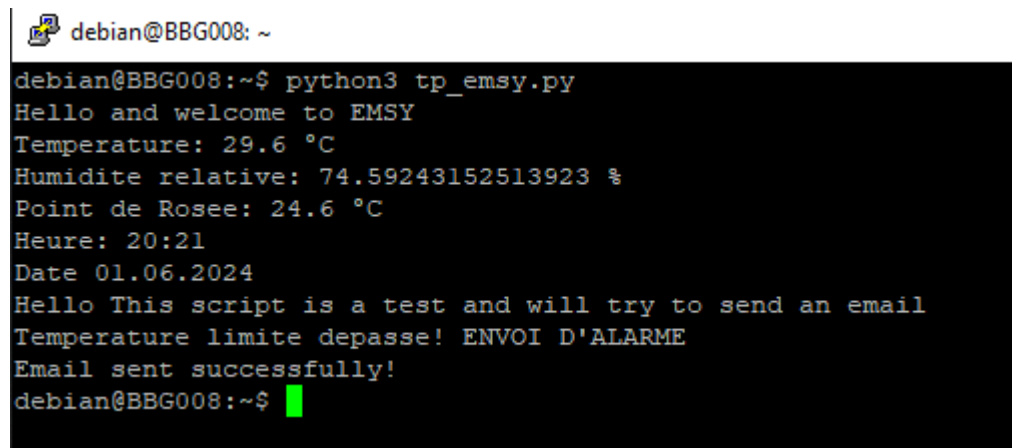
```
debian@BBG008: ~  
debian@BBG008:~$ python3 tp_emsy.py  
Hello and welcome to EMSY  
Temperature: 25.7 °C  
Humidite relative: 41.39452201113909 %  
Point de Rosee: 11.6 °C  
Heure: 20:19  
Date 01.06.2024  
Hello This script is a test and will try to send an email  
Temperature dans la plage de securite :)  
debian@BBG008:~$
```

/home/debian/TempLog.csv – 10.228.134.227 –

Date	Heure	Temperature	Humidité	Point de rosée
01.06.2024	20:18	25.9	41.09	11.67
01.06.2024	20:18	25.9	41.11	11.68
01.06.2024	20:18	25.9	41.14	11.69
01.06.2024	20:19	25.9	41.13	11.68
01.06.2024	20:19	25.8	41.13	11.6

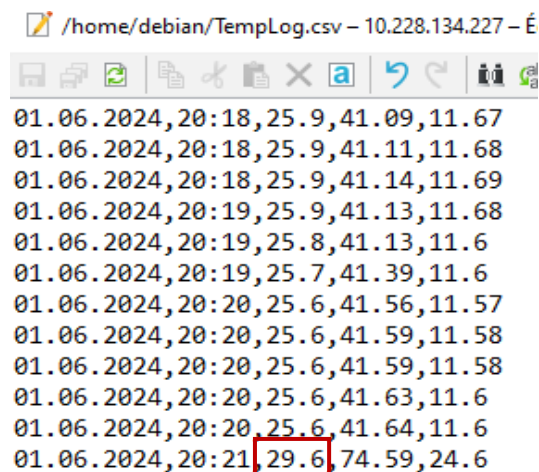
Envoi de l'email d'alerte :

Dans ce cas, le capteur a détecté une température supérieure à 28°C et donc un email d'alerte sera envoyé (**Email sent successfully**) et nous aurons aussi l'information dans le fichier .csv



The terminal window shows the script execution where the temperature is 29.6°C, which is above the 28°C threshold. The script outputs a message indicating that the temperature limit has been exceeded and that an alert email has been sent successfully.

```
debian@BBG008: ~  
debian@BBG008:~$ python3 tp_emsy.py  
Hello and welcome to EMSY  
Temperature: 29.6 °C  
Humidite relative: 74.59243152513923 %  
Point de Rosee: 24.6 °C  
Heure: 20:21  
Date 01.06.2024  
Hello This script is a test and will try to send an email  
Temperature limite depasse! ENVOI D'ALARME  
Email sent successfully!  
debian@BBG008:~$
```

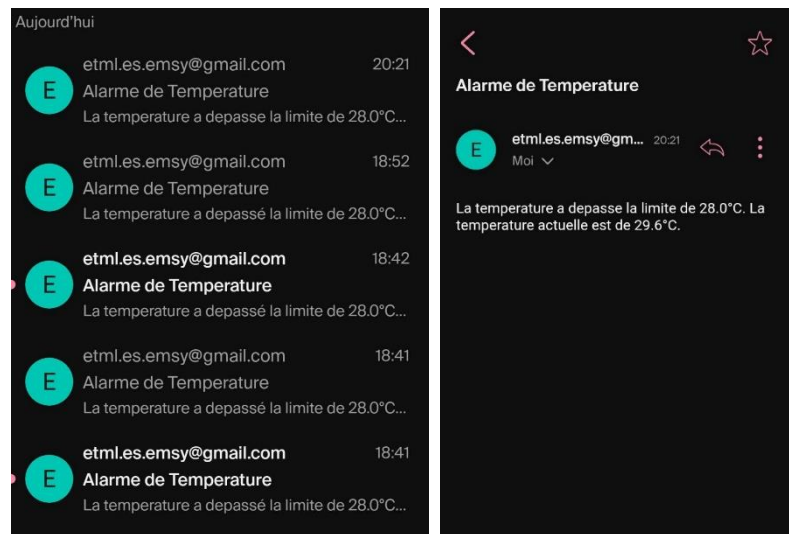


The CSV file view shows the same data as before, but with an additional row at the bottom where the temperature is 29.6°C. This row is highlighted with a red box, indicating the alert condition.

/home/debian/TempLog.csv – 10.228.134.227 – É

Date	Heure	Temperature	Humidité	Point de rosée
01.06.2024	20:18	25.9	41.09	11.67
01.06.2024	20:18	25.9	41.11	11.68
01.06.2024	20:18	25.9	41.14	11.69
01.06.2024	20:19	25.9	41.13	11.68
01.06.2024	20:19	25.8	41.13	11.6
01.06.2024	20:19	25.7	41.39	11.6
01.06.2024	20:20	25.6	41.56	11.57
01.06.2024	20:20	25.6	41.59	11.58
01.06.2024	20:20	25.6	41.59	11.58
01.06.2024	20:20	25.6	41.63	11.6
01.06.2024	20:20	25.6	41.64	11.6
01.06.2024	20:21	29.6	74.59	24.6

Dans ces images nous voyons que nous avons bien reçu les messages d'alerte.



Automatisation de l'envoi de l'email d'alerte

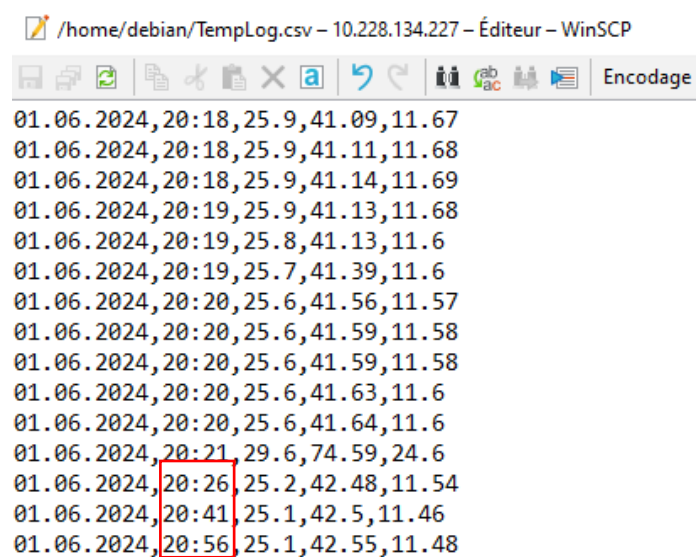
Grâce à la commande **crontab -e** nous avons automatisé notre code, chaque 15min il sera exécuté et si la température est supérieure à 28°C un email d'alerte sera envoyé.

```
debian@BBG008: ~  
GNU nano 3.2 /tmp/crontab.xpTb6m/crontab  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
*/15 * * * * /usr/bin/python3 /home/debian/tp_emsy.py
```

Et avec la commande **crontab -l** nous voyons les taches programmées déjà en place, ici nous voyons celle qu'on avait ajoutée.

```
debian@BBG008: ~  
debian@BBG008:~$ crontab -l  
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').  
#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
*/15 * * * * /usr/bin/python3 /home/debian/tp_emsy.py  
debian@BBG008:~$
```

Et dans le fichier TempLog.csv nous avons les infos chaque 15 minutes.



The screenshot shows a WinSCP editor window titled "/home/debian/TempLog.csv - 10.228.134.227 - Éditeur - WinSCP". The window contains a CSV file with 15 rows of data. The data is as follows:

Date	Time	Temp (°C)	Humid (%)	Press (hPa)
01.06.2024	20:18	25.9	41.09	11.67
01.06.2024	20:18	25.9	41.11	11.68
01.06.2024	20:18	25.9	41.14	11.69
01.06.2024	20:19	25.9	41.13	11.68
01.06.2024	20:19	25.8	41.13	11.6
01.06.2024	20:19	25.7	41.39	11.6
01.06.2024	20:20	25.6	41.56	11.57
01.06.2024	20:20	25.6	41.59	11.58
01.06.2024	20:20	25.6	41.59	11.58
01.06.2024	20:20	25.6	41.63	11.6
01.06.2024	20:20	25.6	41.64	11.6
01.06.2024	20:21	29.6	74.59	24.6
01.06.2024	20:26	25.2	42.48	11.54
01.06.2024	20:41	25.1	42.5	11.46
01.06.2024	20:56	25.1	42.55	11.48

The time column (20:18, 20:19, 20:20, 20:21, 20:26, 20:41, 20:56) is highlighted with a red box in the original image.