



Transformation d'un banc de compression de piles à combustible, la suite...

Nicolas CHARVIN – Christophe CARRAL

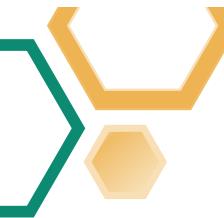


- Produit commercialisé par *Pragma Industries*
- PC embarqué + interface tactile
- Logiciel de contrôle développé en LabVIEW, distribué en .exe
- Régulation du vérin en force et déplacement

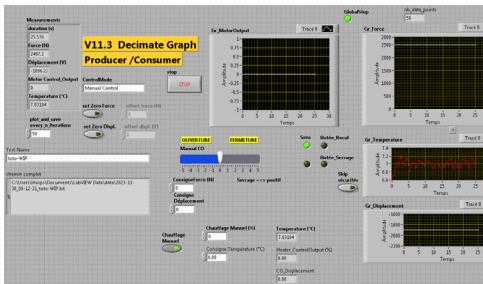
Nos objectifs:

- Pilotage plus personnalisé des cycles de compression/décompression
- Couplage avec des mesures électriques
- Intégration avec un système d'humidification





Résumé des épisodes précédents...



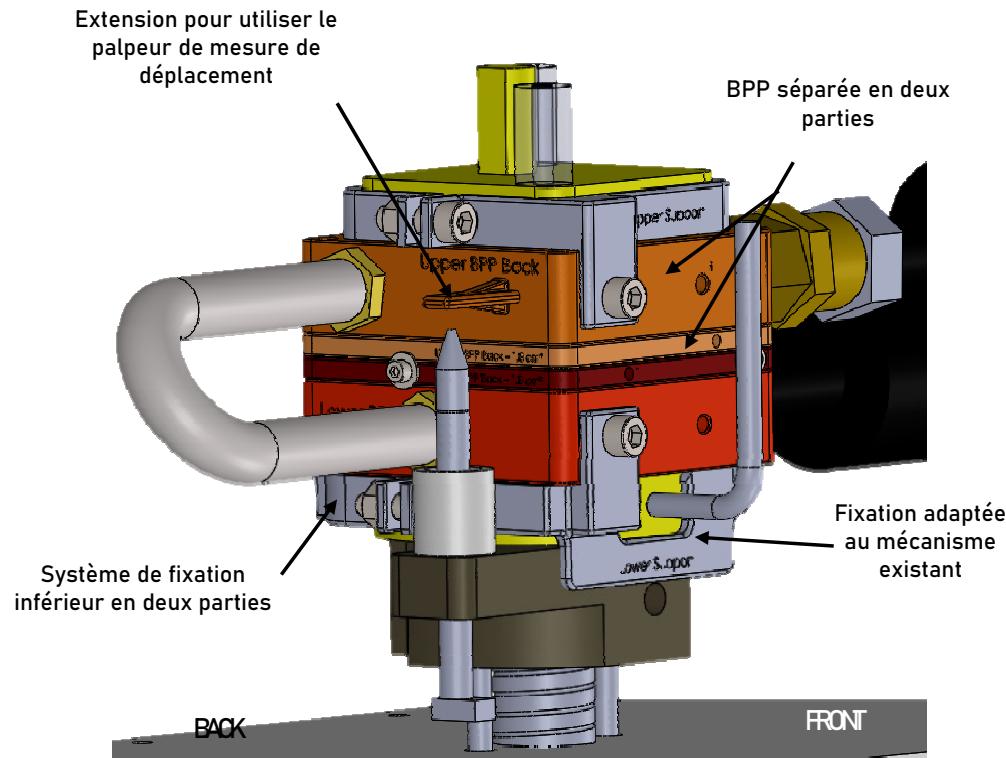
Un circulateur-refroidisseur



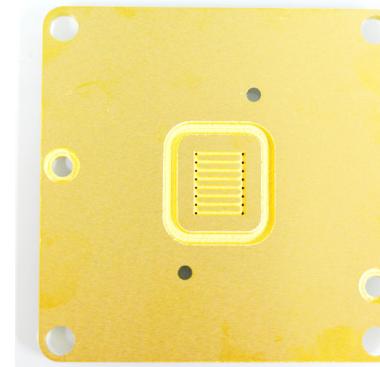
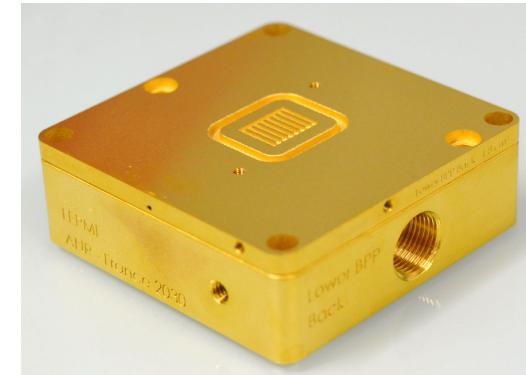
Une seule et même application ?

De ma difficulté à développer une application d'envergure...





Daphné MOLÉ (stagiaire UTC 2024)



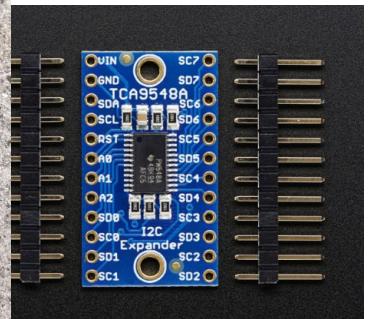
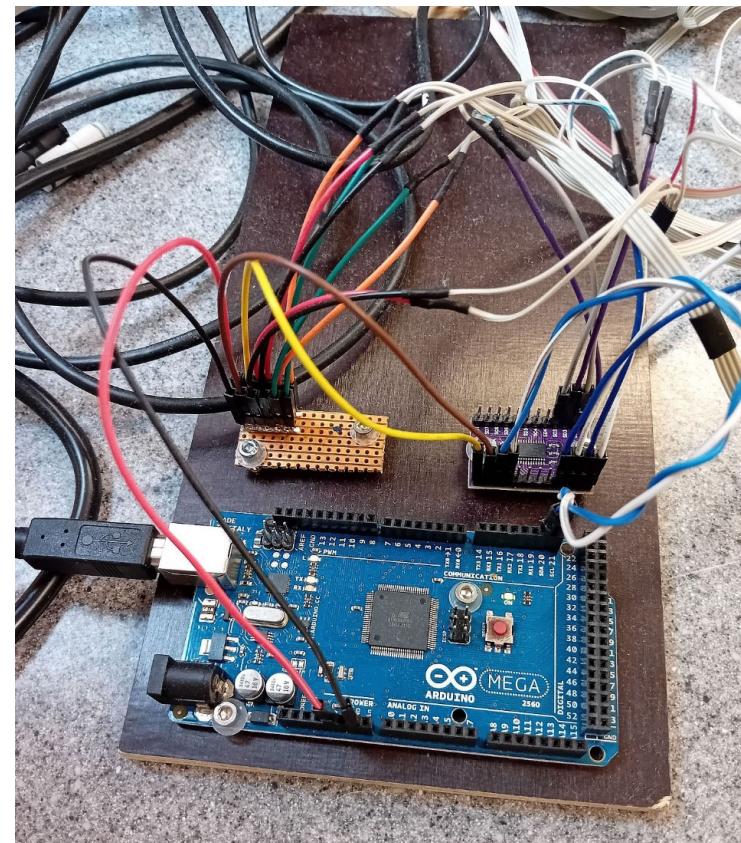
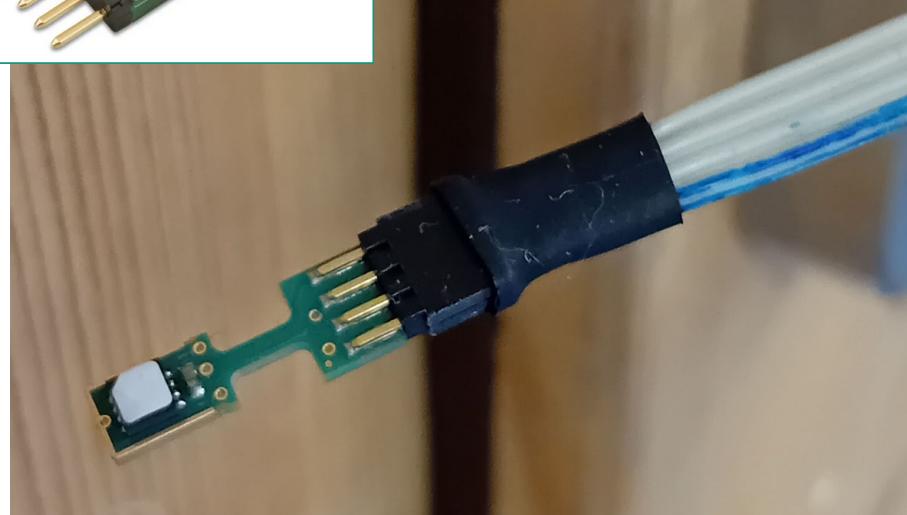
Fabrication : inox 316L;
fusion laser + surfaçage,
dorure 2µm



<https://www.initial.fr/>

Capteur SHT85 + multiplexeur I2C TCA9584A

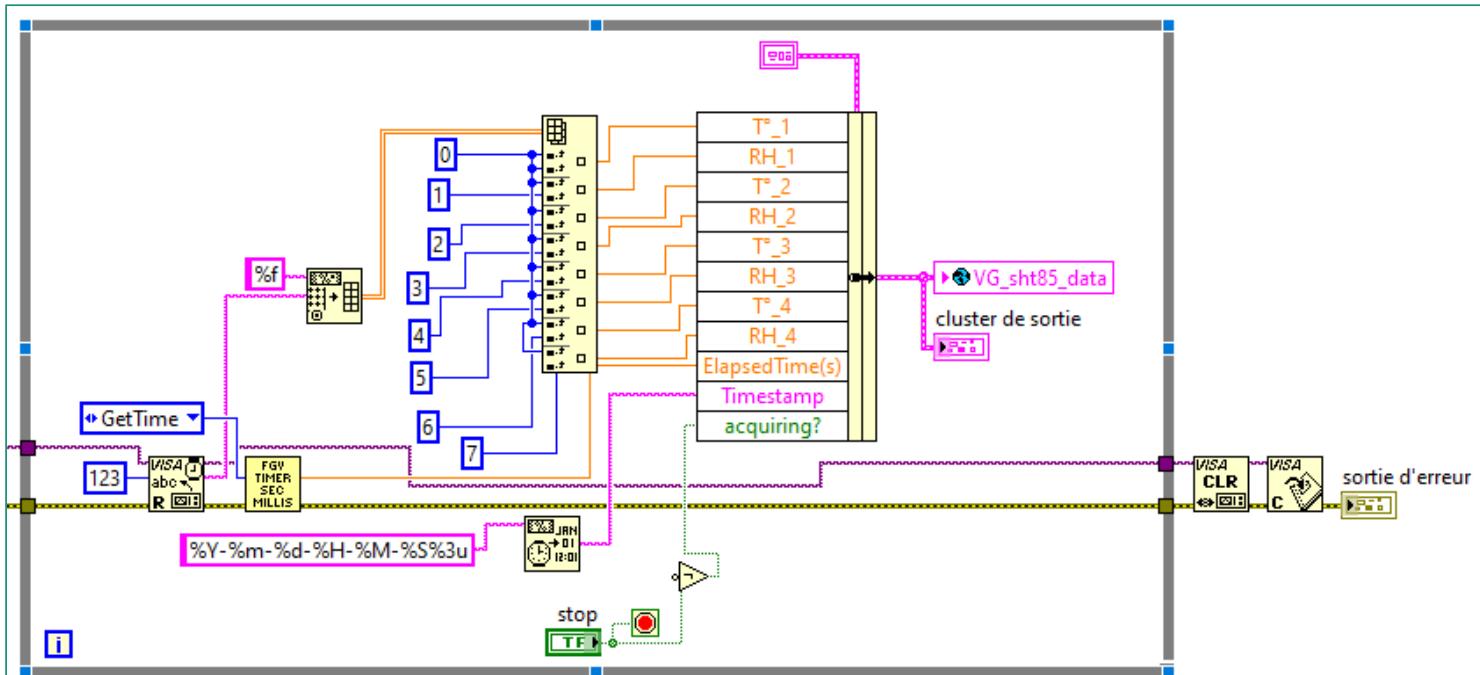
Montage sur câble en nappe.



Lecture %RH

Library SHT85 <https://github.com/RobTillaart/SHT85>

L'arduino écrit toutes les 50ms une chaîne avec les 4 couples [T°, %RH]
LabVIEW lit cette chaîne et écrit les valeurs dans une variable globale.

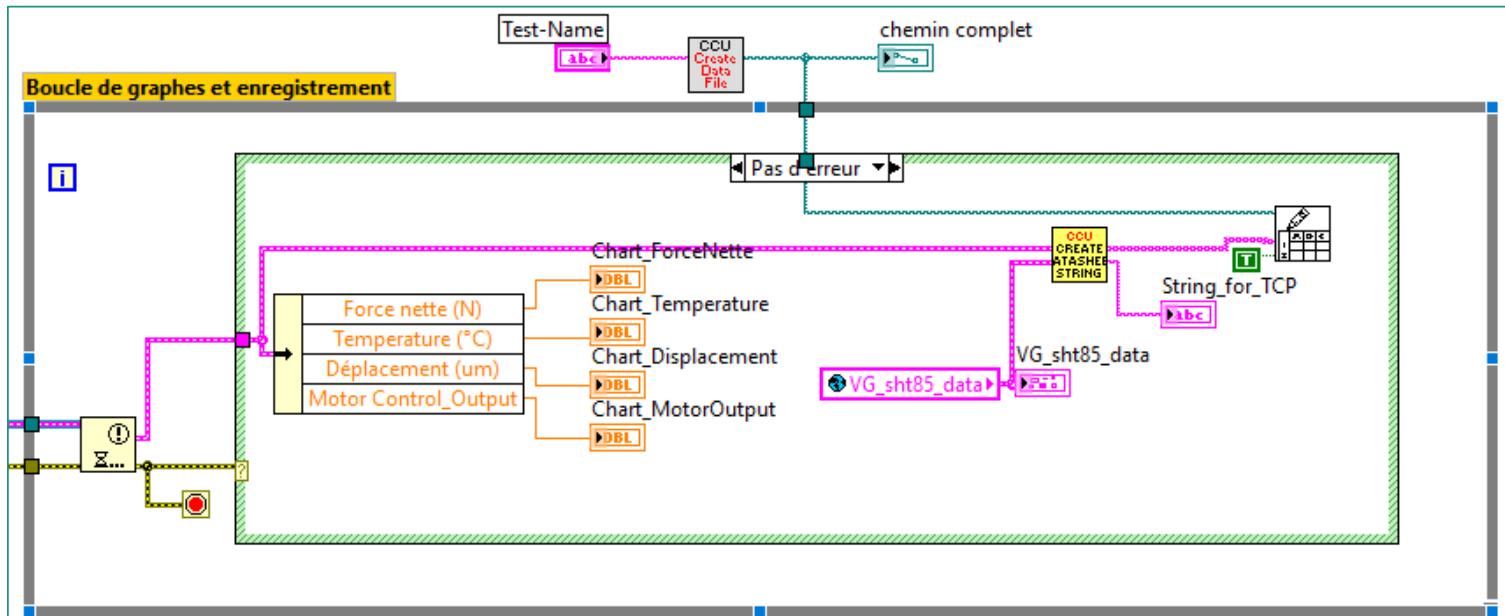


Partager les mesures d'un VI à l'autre

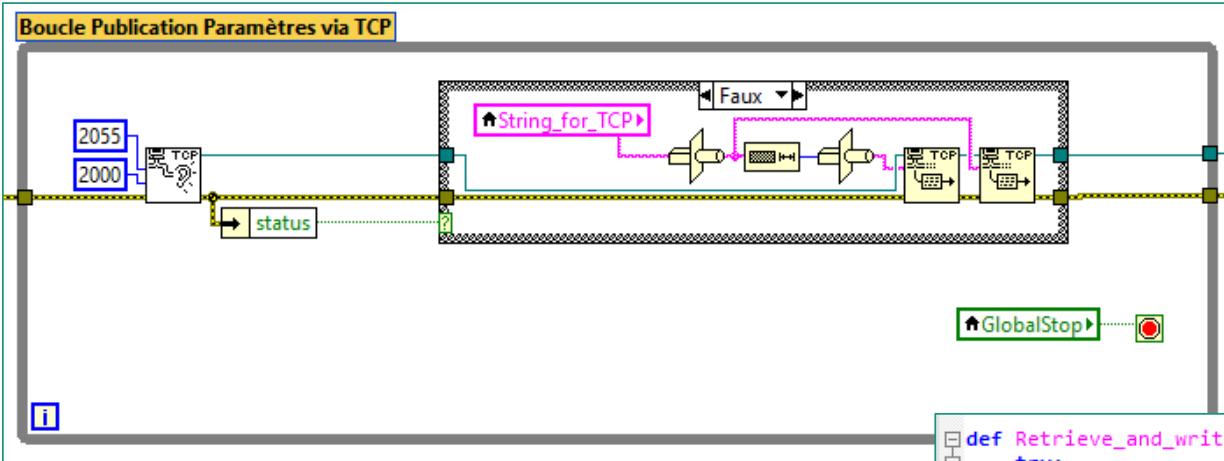
Grenoble - Chambéry

Objectif: que le VI "contrôle du banc de compression" puisse accéder aux mesures de %RH issus du VI "contrôle de l'humidité"

- Utilisation d'une variable globale, "publiée" par un VI dédié, et qui sera relu par les VIs "contrôle Force" et "contrôle RH"

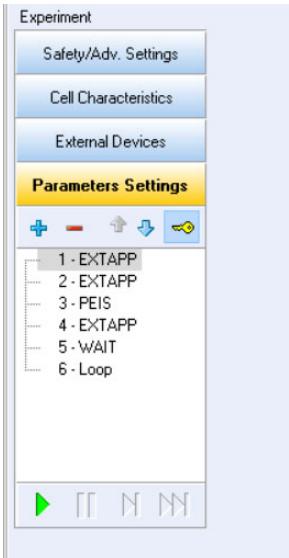


Couplage LabVIEW ↔ EC-Lab



Publication TCP, inspiré de
 "LabVIEW 2010
 \examples\comm\TCP.llb\Simple
 Data Server.vi"

Le logiciel EC-Lab, qui pilote le potentiomstat, s'exécute sur un autre ordinateur



```
def Retrieve_and_write_CCU_values(cycle_number=-999):
    try:
        sock = socket.create_connection(('192.168.135.172', 2055),timeout=2)
        buf_bytes = sock.recv(4)

        nbybytes = int.from_bytes(buf_bytes, "big")
        print(f'There are {nbybytes} bytes to be read.')
        buf_bytes = sock.recv(nbybytes)

        data_string = buf_bytes.decode()

        sock.close()

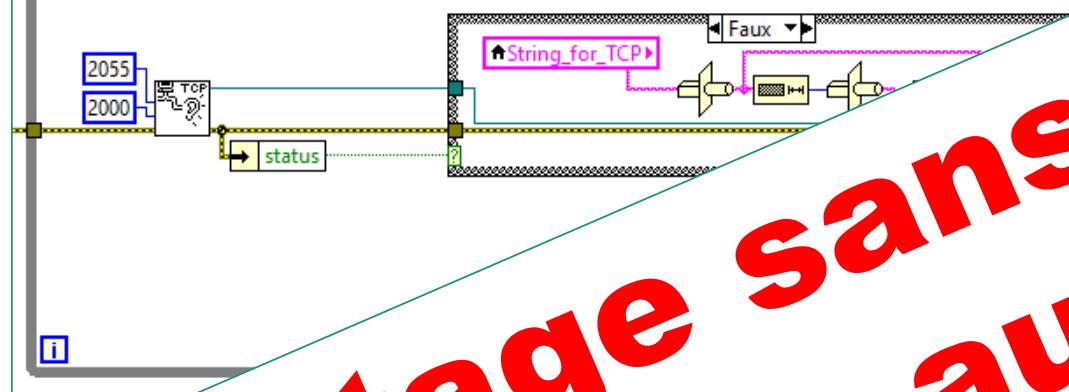
    except socket.timeout:
        print("Time OUT !!!")

    data = data_string.split('\t')
```

Côté LabVIEW, il y a probablement plus simple, plus propre et plus moderne...

Couplage LabVIEW ↔ EC-Lab

Boucle Publication Paramètres via TCP

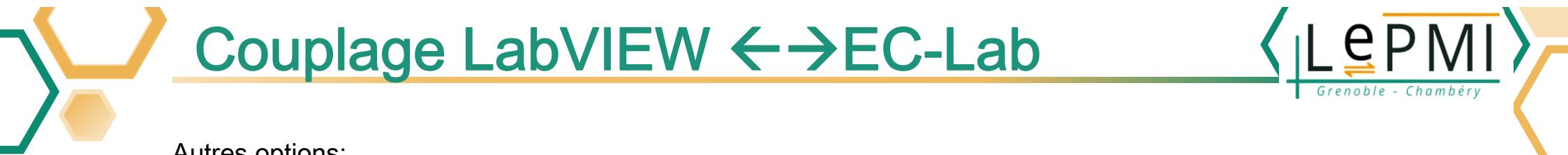


Le logiciel EC-Lab
le programme
s'exécute sur
un autre ordinateur

Plantage sans raison
apparente au bout de
3 jours...

```
135.172.1.2055, timeout=2)
    .bytes, "big")
    bytes to be read.')
    .decode()
    sock.close()
except socket.timeout:
    print("Time OUT !!!")
data = data_string.split('\t')
```

Côté LabVIEW, il y a probablement plus simple, plus propre et plus moderne...



Couplage LabVIEW ↔ EC-Lab

Autres options:

1/

Le PC LabVIEW → écrit un fichier texte dans un dossier partagé par Windows

Le PC EC-Lab accède à ce dossier partagé

2/

Démarrer un serveur web minimalist (<https://busybox.net/>) sur le PC LabVIEW

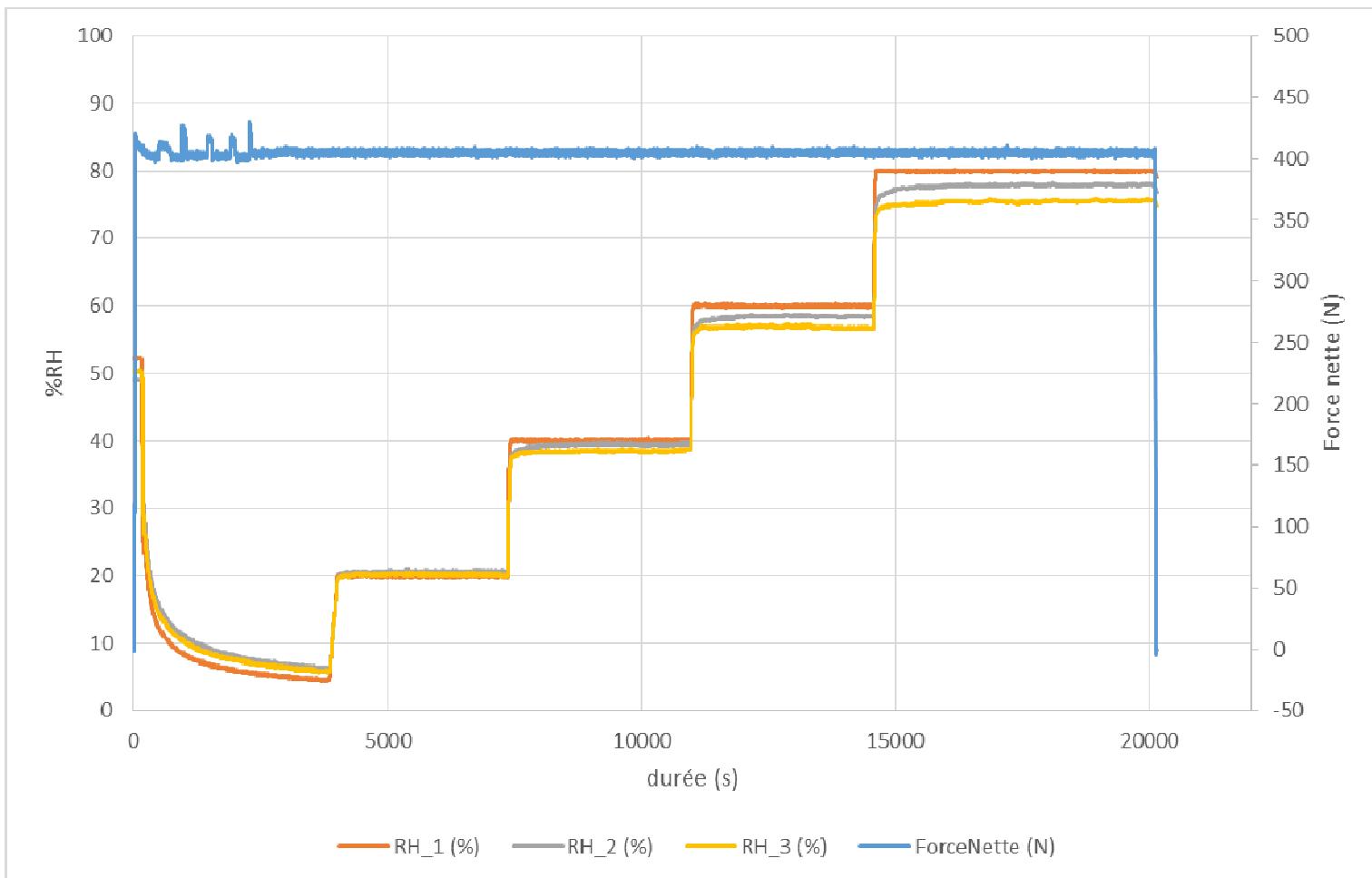
Ecrire et publier un fichier texte

Récupérer ce fichier depuis le PC EC-Lab

3/

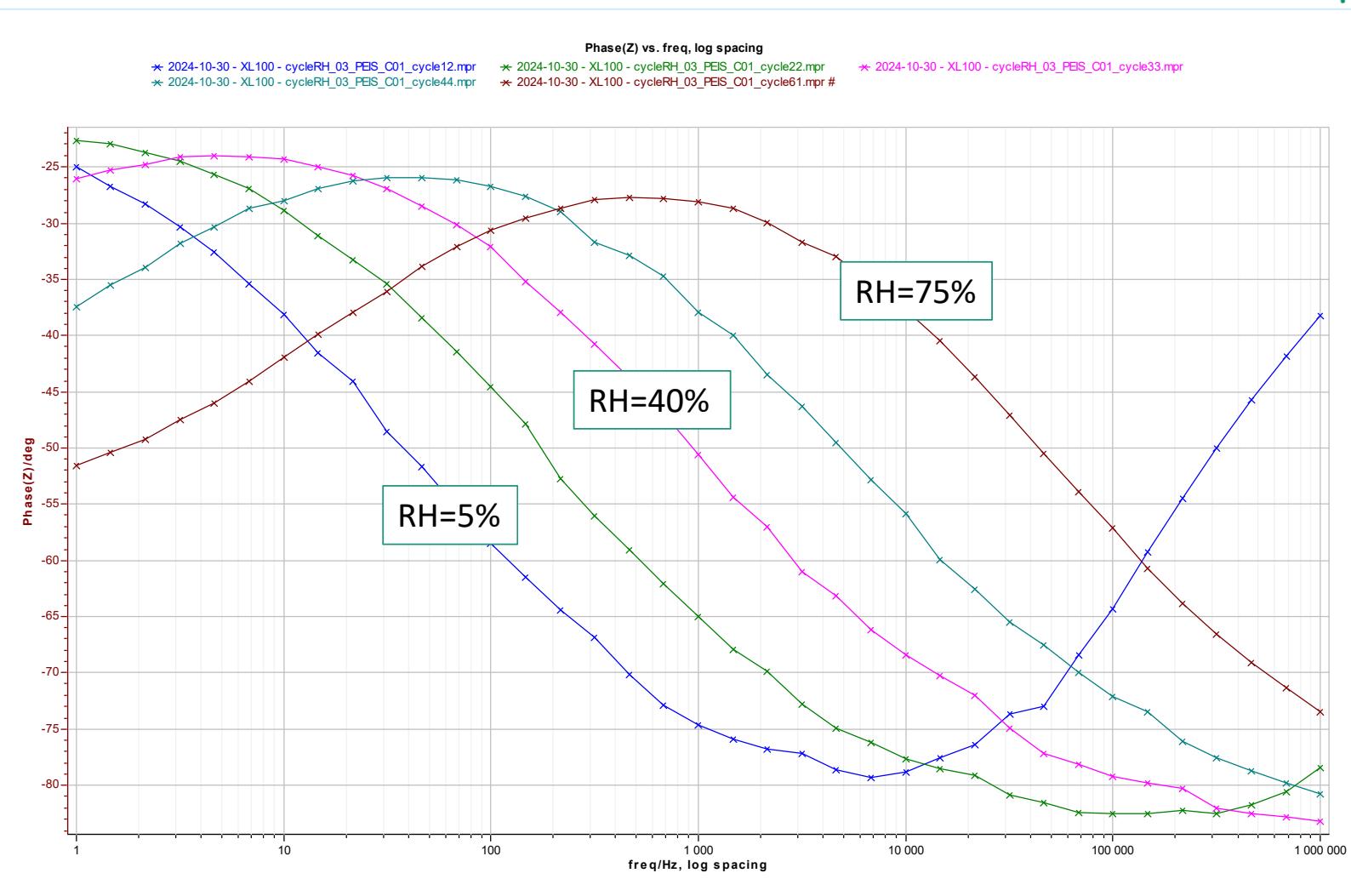
Faire tourner EC-Lab et LabVIEW sur le même PC (crainte de "surcharger" le port USB, ou le PC en général)

Résultats 1





Résultats 2





La suite

- Mesures de longue durée (plusieurs jours).
Actuellement, ajout d'une ligne dans un fichier CSV, avec les pour et les contre
- Intégrer le pilotage du refroidisseur Julabo, pour faire des cycles.