

Mesures photovoltaïques I(V) multi-voies

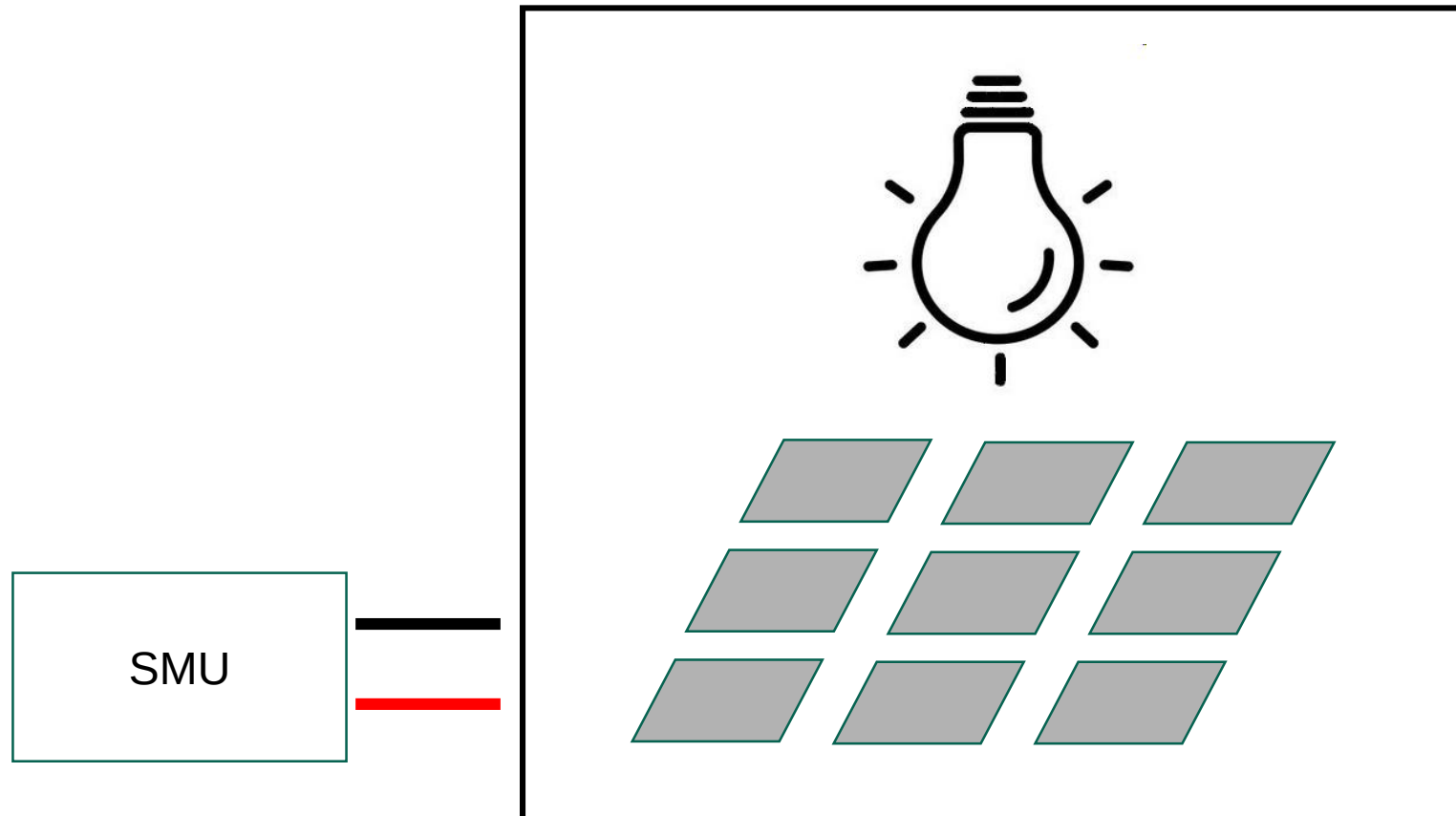
N. Charvin – G. De Moor – J.-C. Honoré – L. Flandin

laboratoire LEPMI

Journées AlpesVIEW – 15 – 16 juin 2026



But: réaliser des mesures $I(V)$ sur plusieurs cellules PV en pseudo-simultané, pendant leur vieillissement, et sans les sortir de l'enceinte climatique (80°C / 80 %RH)



1^e essai : panneau LED acheté 20€
chez Amazon

Eclairage instable, et tombe en
panne après 6 mois de manips!

Supporte mal la température et
l'humidité *a priori*

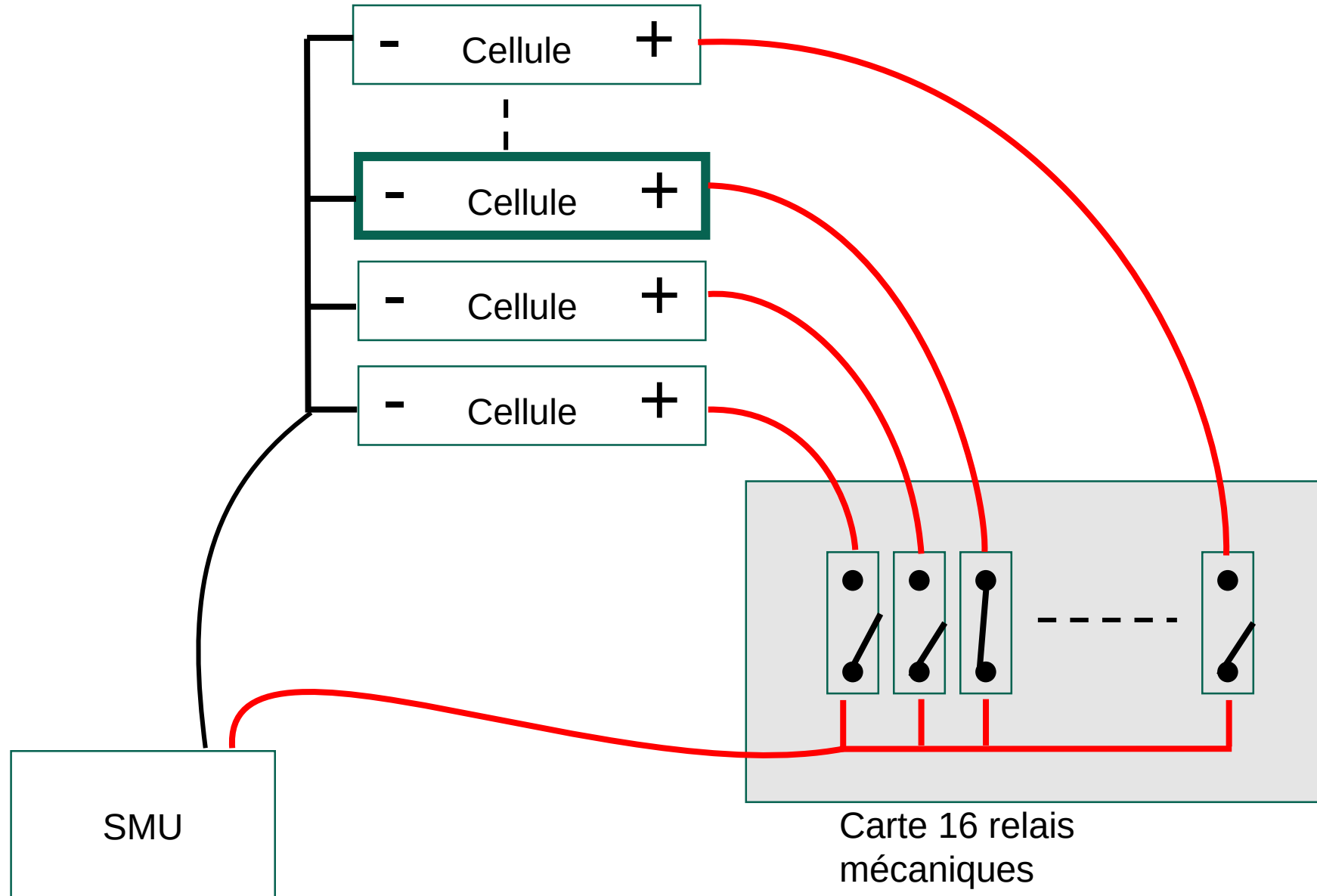


2^{ème} essai : lampe JEL Snapper
35HT

Éclairage industriel (120°C max)

En cours d'acquisition





Enceinte Temp + %RH
SMU Keithley 2602
Platine 16 Relais USB Numato
Lampe
LabVIEW

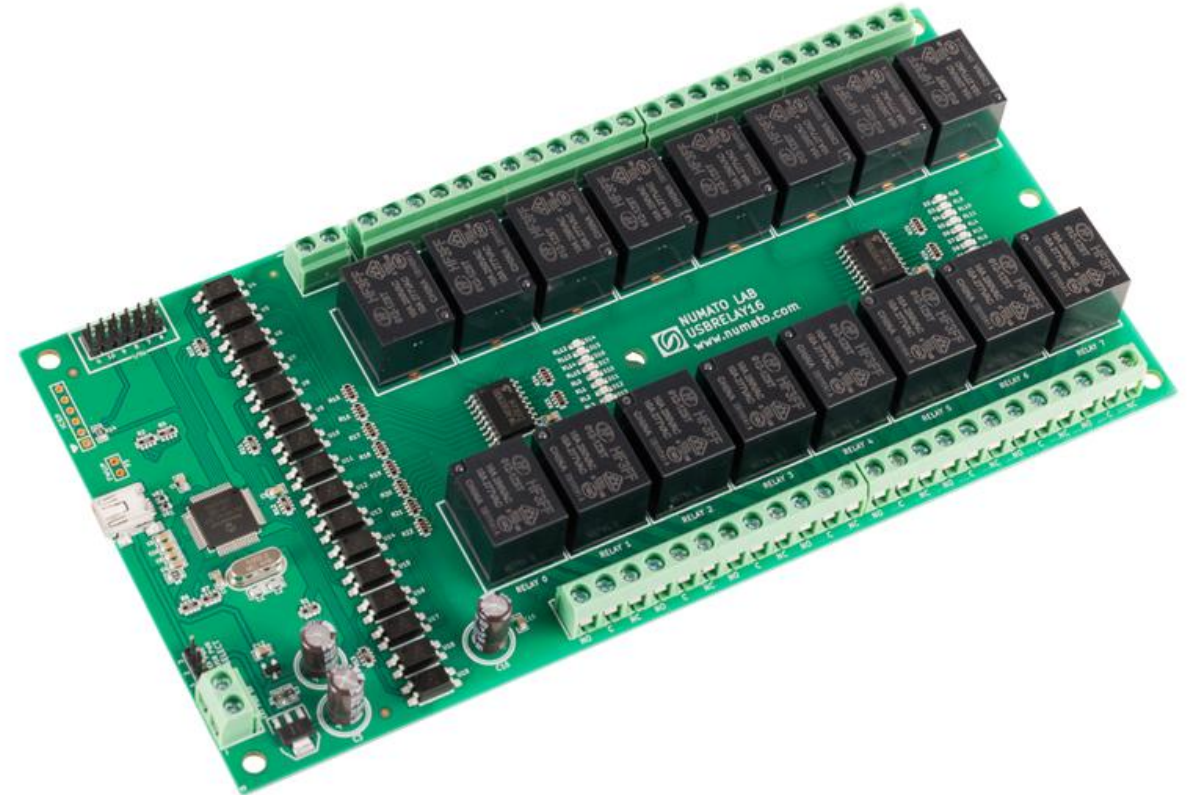
16 Mechanical Relay with contact rating up to 7A
10 TTL (3.3V) compatible GPIOs
5 Analog inputs with 10-bit resolution (multiplexed with digital IOs)

Pilotage via commande ASCII sur port série:
"relay 4 ON" , "relay 7 OFF"

Utilisable avec n'importe quel langage de programmation

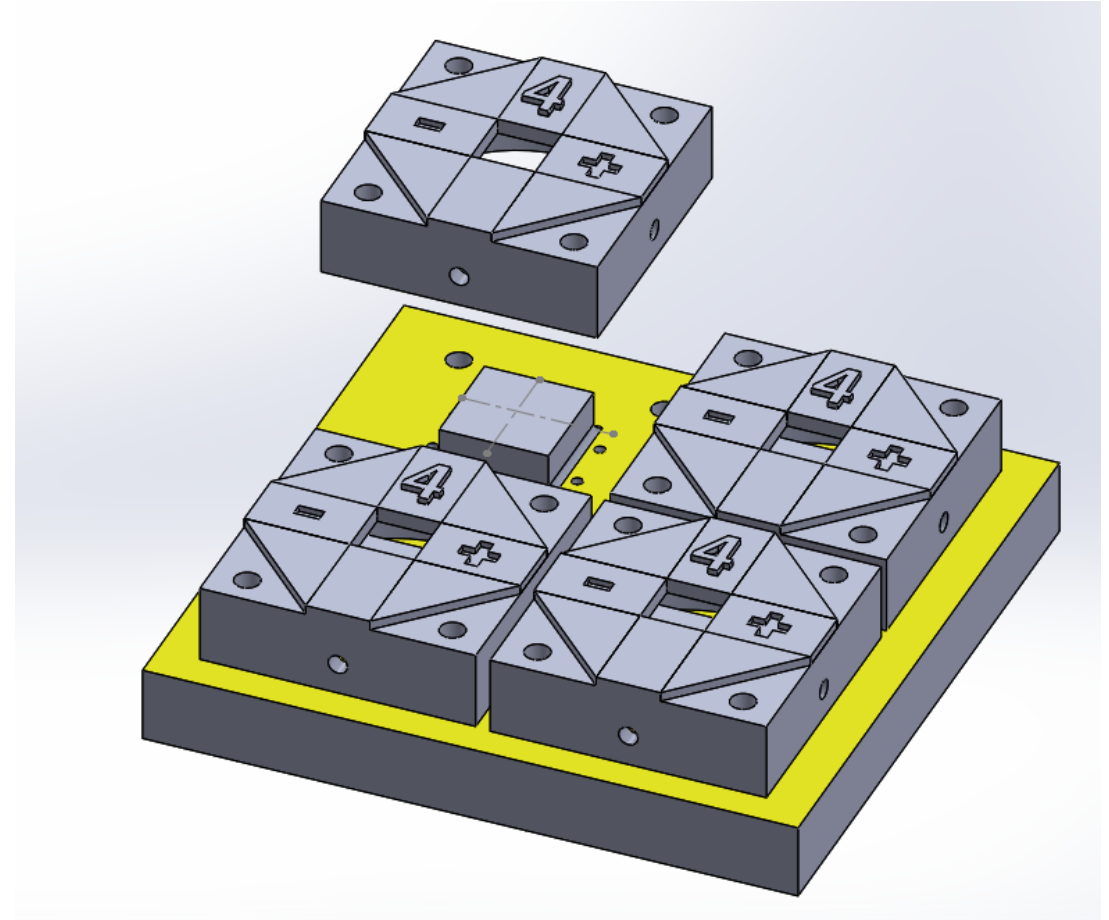
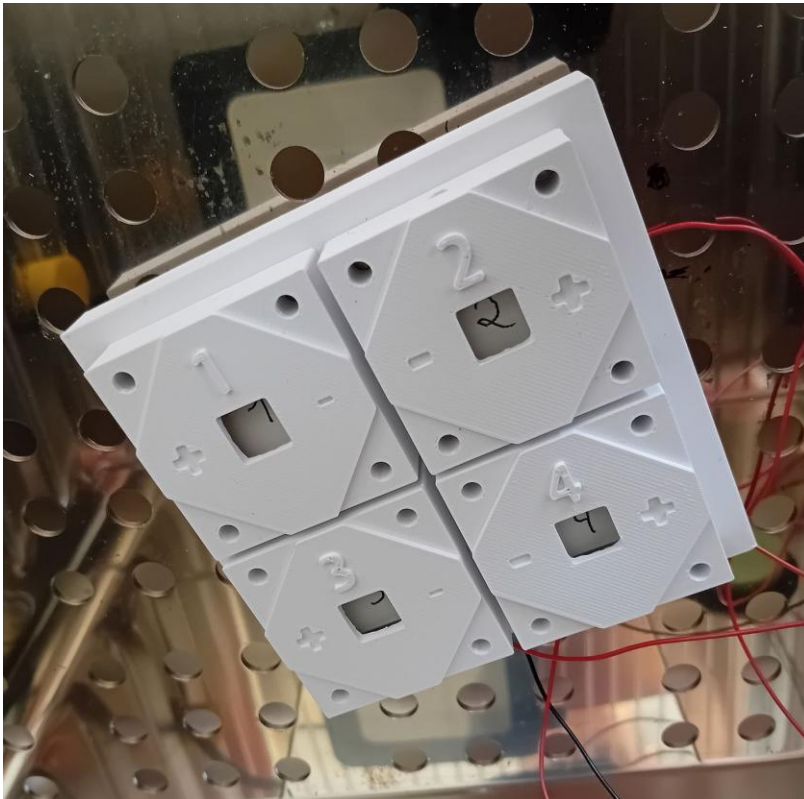
<https://numato.com/product-category/usb-relay-modules>

Relais mécaniques ou SSR



Impression 3D des supports:

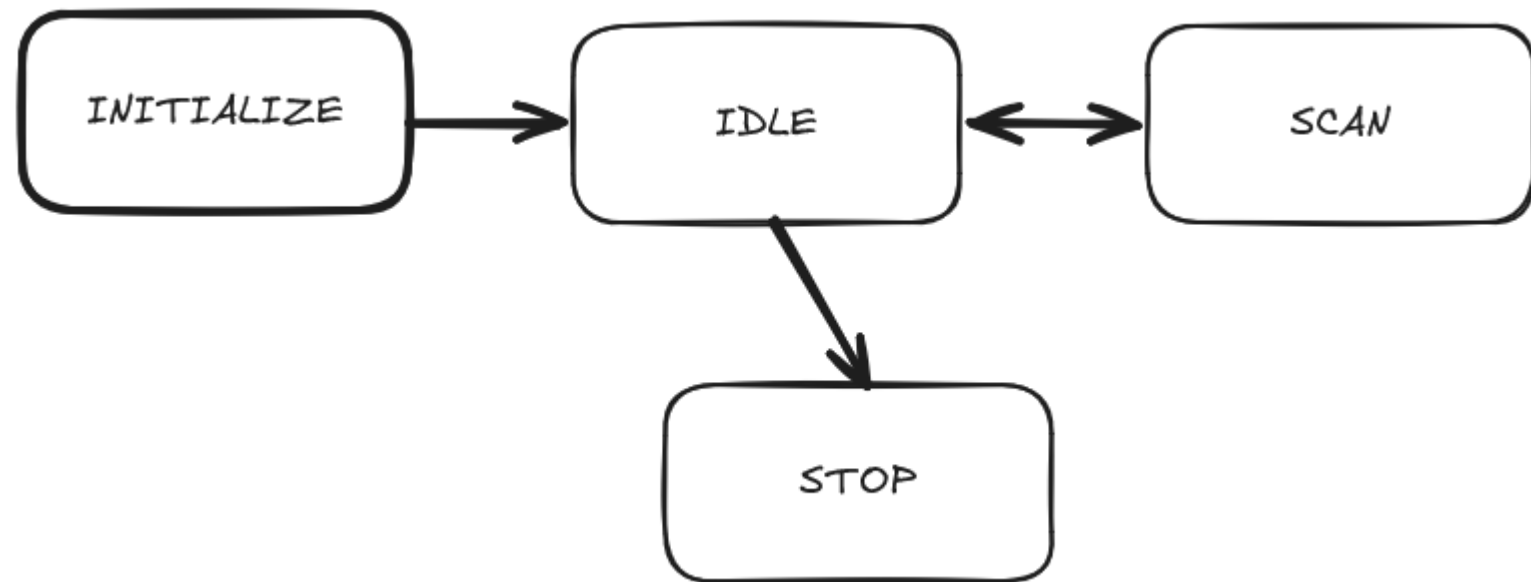
- Choix du matériau
- Système de fixation:
 - Capot aimanté
 - Pointe à ressort pour les contacts

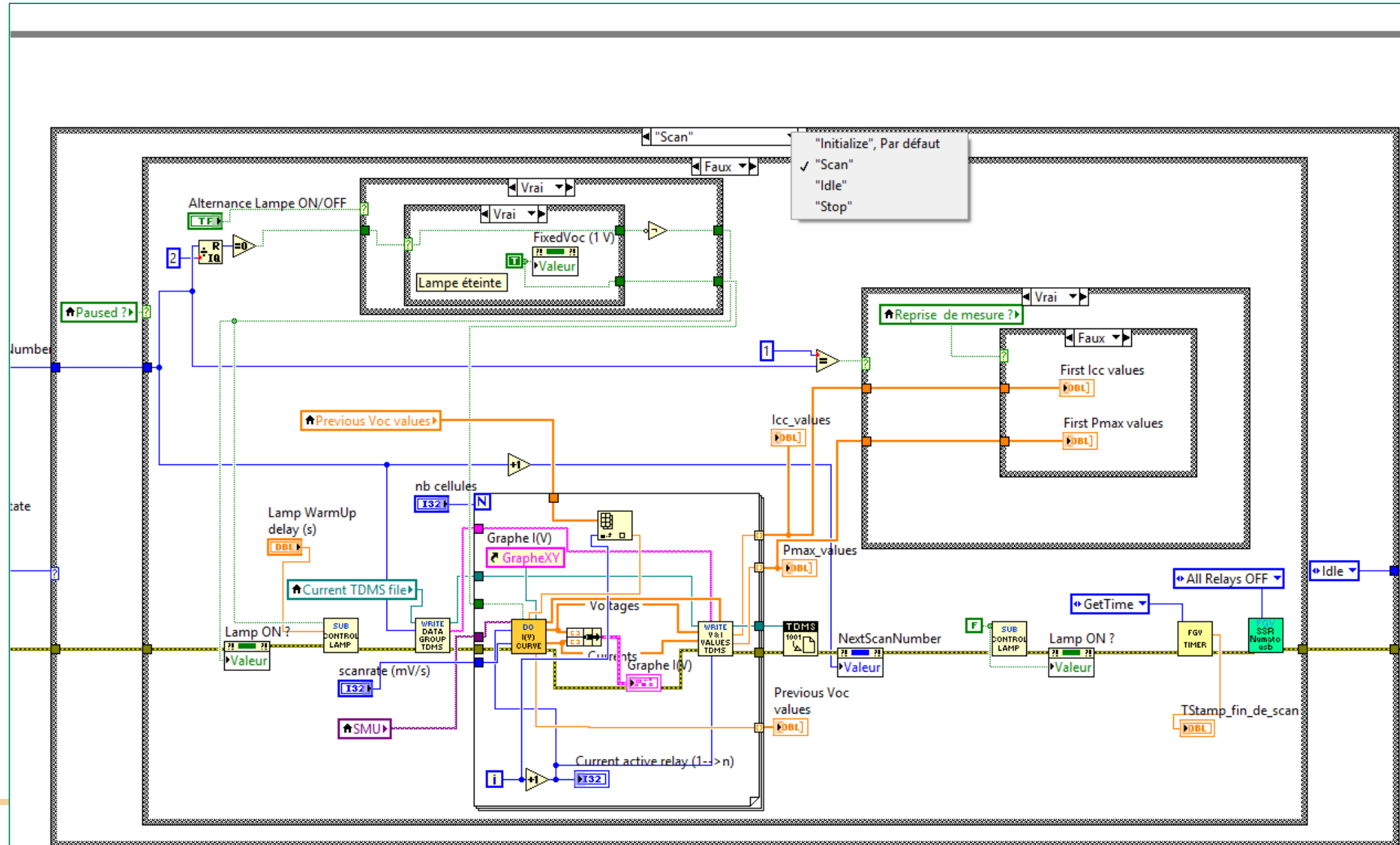


Machine d'états : Initialize, Scan, Idle, Stop

À chaque scan: on effectue une mesure $I(V)$ pour les différents cellules. La platine de relais "aiguille" le SMU sur la cellule concernée

Enregistrement TDMS





<https://github.com/adamreeve/npTDMS>



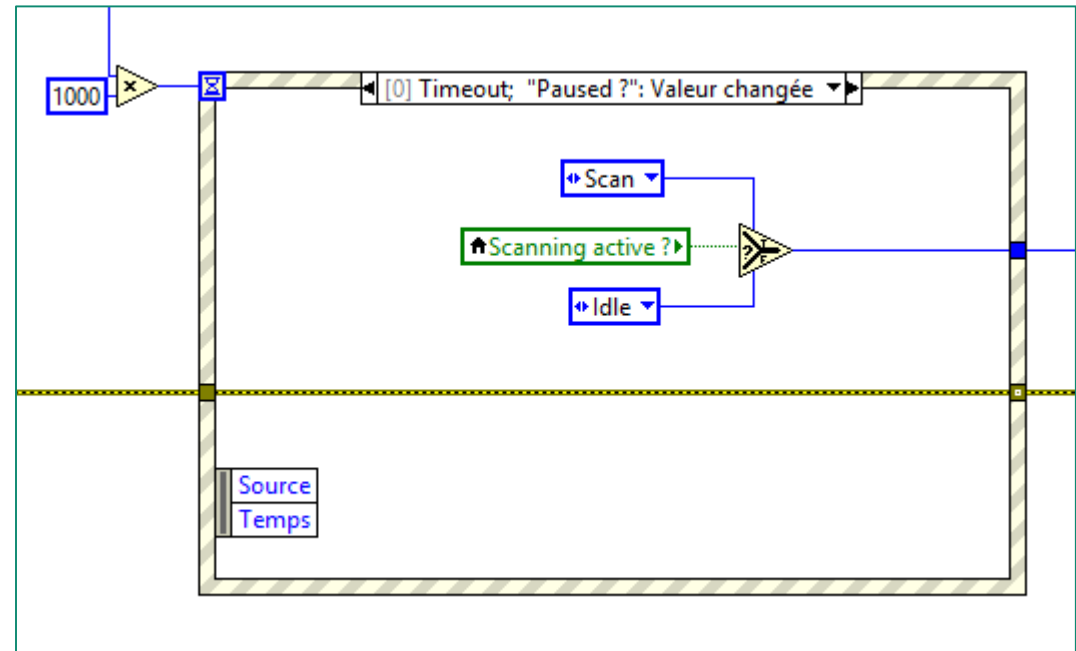
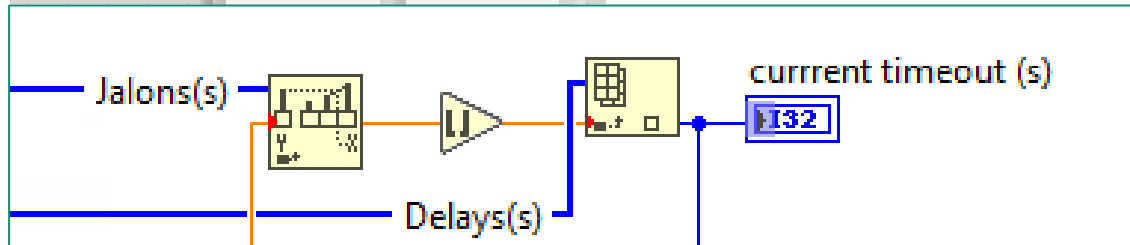
Des mesures rapprochées en début de vieillissement, puis de plus en plus éloignées

Méthode:

- À chaque fin de scan, j'entre dans l'état "Idle", et je récupère le temps écoulé
- Je compare ce temps avec un tableau de jalons associant "temps écoulé → temps d'attente entre scans"
- Je modifie le Timeout de la structure Évènement en conséquence

jalons + delays (s)

0	0	20
0	120	60
	3600	1800
	1800	1800

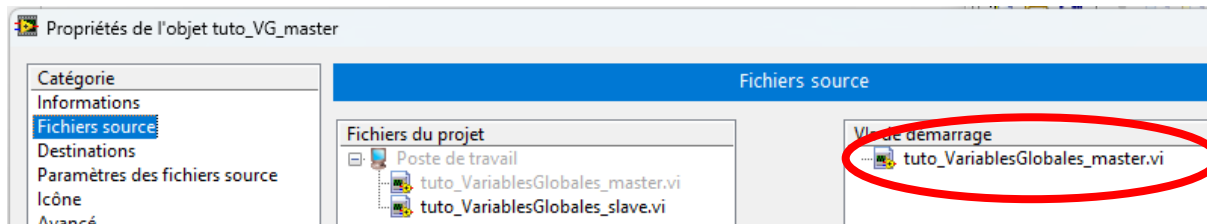


Profil de temps d'attente
Mise en pause

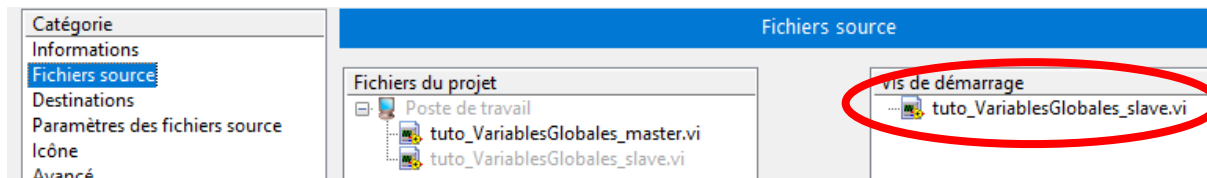
Digression : AppBuilder, fenêtres infermables!

Variable globale (un VI qui écrit, un VI qui lit) → OK

Si on "builde" les VI séparément, la variable globale ne fonctionne pas !
→ 2 fichiers .EXE séparés



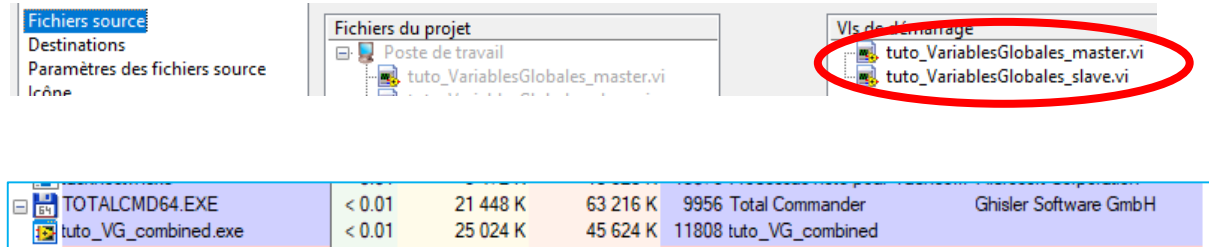
Au final, je me retrouve avec 2 fichiers .EXE (**tuto_VG_master.exe** et **tuto_VG_slave.exe**). Si j'ouvre ces 2 programmes, j'ai bien 2 processus dans le gestionnaire de tâches.



TOTALCMD64.EXE	< 0.01	21 144 K	66 880 K	9956 Total Commander
tuto_VG_master.exe	0.01	25 432 K	46 228 K	12724 tuto_VG_master
tuto_VG_slave.exe	0.01	25 300 K	45 388 K	14944 tuto_VG_slave

Digression : AppBuilder, fenêtres infermable!

Si on "builde" ces deux VI dans un seul et même fichier .EXE :

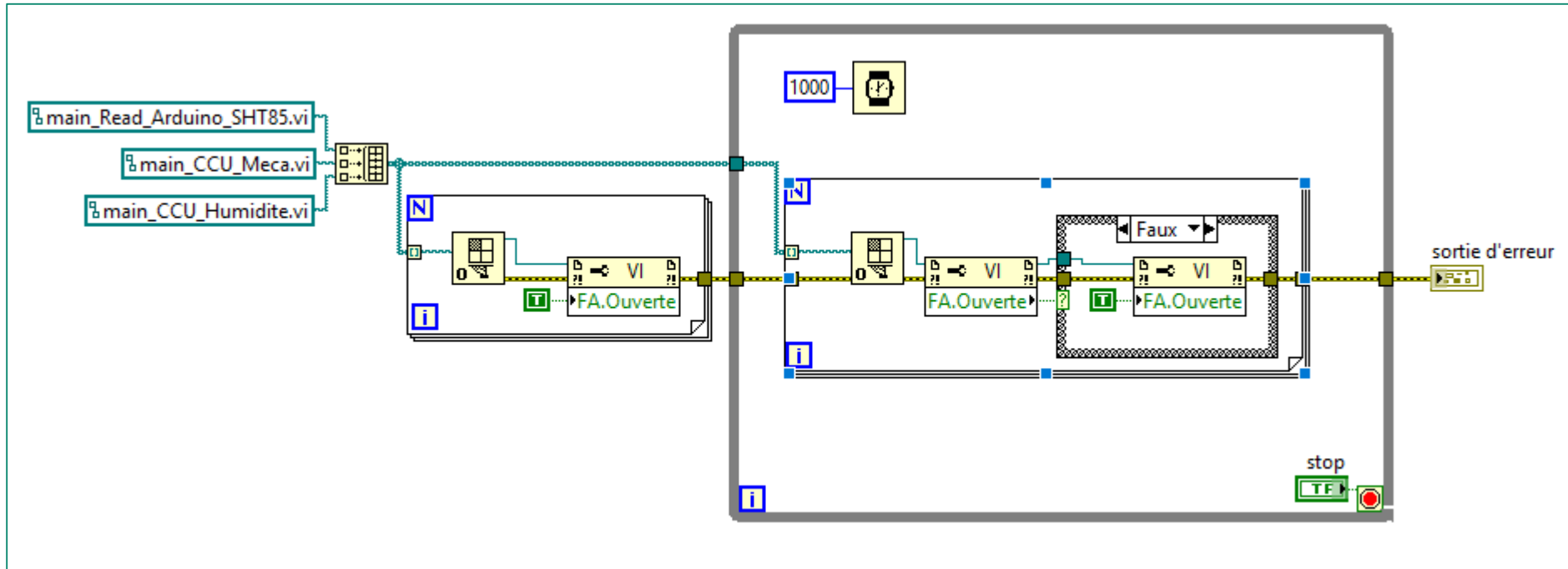


Au final, je me retrouve avec 1 fichier .EXE, mais bien deux fenêtres, et la variable globale fonctionne.

Un même fichier .EXE ouvre 2 fenêtres VI.

Si, par erreur, je ferme une de ces fenêtres, comment la ré-ouvrir sans perturber les autres ?

Avec un VI maître qui ouvre les différentes faces-avant, et surveille



<https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000YIQtCAO&l=fr-FR>

- Avec une structure "Évènement", on peut capturer l'évènement "Face-avant fermée" (Panel Close), et demander confirmation
- Empêcher d'utiliser le bouton fermeture
 - Using the VI Properties:
 1. Go to **File»VI Properties**.
 2. Choose the **Window Appearance** category.
 3. Click **Customize**.
 4. Uncheck **Allow user to close window**.
- Allez encore plus loin, et cacher la barre de menus pendant l'exécution