

DUT Mesures Physiques

Réalisation de mesures statiques et dynamiques
d'un champ de température sur un système de
refroidissement complexe

Hugo BONNEMAISON

Tuteur entreprise

Mr. Thomas PIERRE-EMILE

Tuteur IUT

Mr. Mojtaba JARRAHI

Remerciements

Je souhaiterais tout d'abord remercier les différentes équipes du Laboratoire Leprince-Ringuet (LLR), mécanique, informatique et électronique/instrumentation, pour les moyens matériels qui m'ont été mis à disposition tout au long de mon stage, ainsi que pour l'accueil et la sympathie régnant dans les locaux.

Et plus particulièrement à mon tuteur entreprise, Thomas PIERRE-EMILE, ingénieur de recherche, ainsi que Miguel RUBIO-ROY, ingénieur de recherche, pour leur disponibilité, leur pédagogie ainsi que la patience dont ils ont fait preuve afin de rendre mon stage intéressant et agréable, mais également très instructif.

Table des matières

I.	Introduction – Présentation.....	5
1.	Présentation du laboratoire.....	5
2.	Le projet HGICAL	5
3.	Système de refroidissement	7
4.	Module de test des capacités de refroidissement.....	8
II.	Etude et réalisation de la plaque équipée	9
1.	Description des composants.....	9
a.	Tapis chauffants	9
b.	Résistances chauffantes.....	9
c.	Sondes Pt100	10
2.	Montage électrique	10
3.	Disposition des composants	12
III.	Programmation de la communication à distance.....	14
1.	Description du matériel	14
a.	Alimentation HAMEG.....	14
b.	Châssis PXI.....	15
c.	Carte d’acquisition des températures	15
d.	Module de relais	16
2.	Programmation sous LabVIEW	16
3.	Pilotage sous Python et Pyrame	18
IV.	Prise des premières mesures	22
1.	Etalonnage des sondes Pt100	22
2.	Test de précision à la température de dégel	23
V.	Mise en condition expérimentale	25
VI.	Photographies	29

I. Introduction – Présentation

Dans le but de valider ma formation DUT Mesures physiques et d'acquérir une première expérience professionnelle, j'ai effectué un stage de 11 semaines au service mécanique du laboratoire Leprince-Ringuet (LLR) sur le site de l'Ecole Polytechnique.

1. Présentation du laboratoire

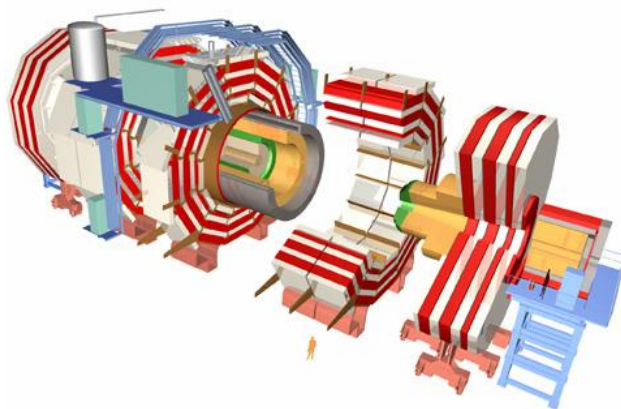
Le laboratoire Leprince-Ringuet est une unité de recherche mixte de l'Institut National de Physique Nucléaire et de Physique des Particules (IN2P3), du Centre National de la Recherche Scientifique (CNRS) rattaché au site de l'Ecole Polytechnique. Implanté à Palaiseau depuis 1936, il tient son nom du physicien Louis Leprince Ringuet, chercheur et professeur à l'Ecole Polytechnique.

Les principaux domaines de recherche traités par les 127 personnes rattachées au laboratoire sont les suivants : étude de l'interaction électrofaible, de la violation de la symétrie CP et du plasma de quarks et de gluons. Les différentes expériences effectuées ont lieu auprès des accélérateurs du CERN (Genève, Suisse), de DESY (Hambourg, Allemagne), de SLAC (Stanford, Californie, USA), de RHIC (BNL, New-York, USA) et de JPARC (Tokai, Japon).

La complexité des expériences et des études impliquent une certaine pluridisciplinarité au sein du laboratoire, qui dispose donc de différents services, mécanique, électronique et instrumentation, et informatique, ainsi qu'un service administratif.

2. Le projet HGCAL

Actuellement situé au CERN, près de Genève, le détecteur CMS (Compact Muon Solenoid) explore un large éventail de domaines de la physique, de la recherche du boson de Higgs à celle d'autres dimensions, en passant par la quête des particules qui pourraient constituer la matière noire.



Représentation du détecteur CMS

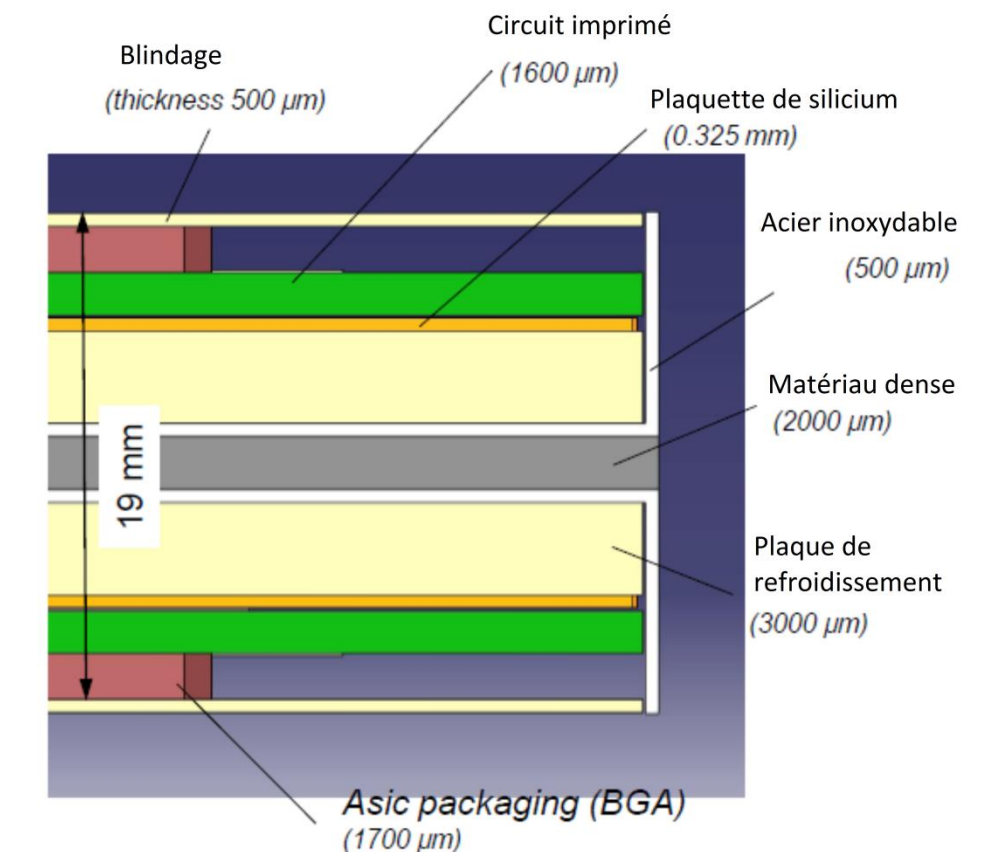
<http://public-archive.web.cern.ch/public-archive/fr/LHC/CMS-fr.html>

Initialement étudié pour une certaine luminosité (phase 0 et 1) en 2009, celle-ci devrait être multipliée par 10 à l'horizon 2024, dans le cadre de nouvelles expériences qui découlent de la découverte d'un boson de Higgs. Le but étant de comprendre le rôle et la nature exacte de cette nouvelle particule (phase 2).

Le projet HGCal (High Granularity CALorimeter) ou Calorimètre à échantillonnage de très Haute Granularité pour la phase de « haute luminosité » de CMS, a pour ambition de remplacer la calorimétrie à l'avant du détecteur CMS pour les futures expériences.

Le système de calorimétrie repose sur la détection de l'énergie d'une particule que l'on souhaite identifier. Pour cela, des paquets de particules sont entrechoqués à une vitesse proche de celle de la lumière sur des points précis appelés points d'interaction. L'explosion issue de ces collisions décrit une sphère remplie de particules qu'il faut identifier. Une partie de celles-ci est expulsée en direction du calorimètre HGCal. Celui-ci se compose des structures alvéolaires pouvant accueillir un empilement de 15 cassettes se composant chacune de :

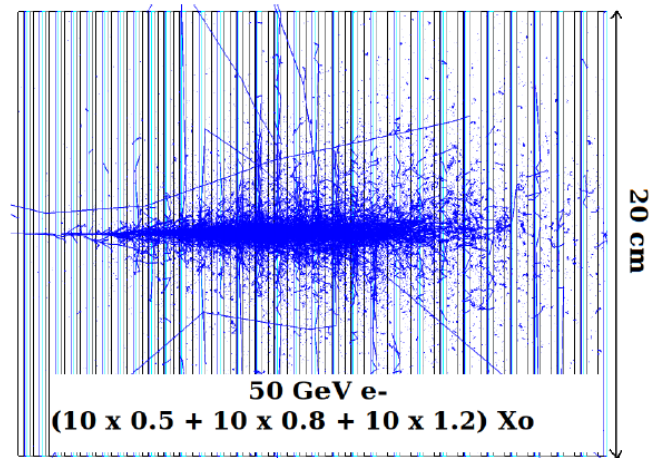
- une couche de matériau dense (tungstène, plomb ou autre), qui a pour rôle de freiner efficacement les particules
- 2 couches de Silicium chargées de détecter le passage des particules en émettant un signal électrique
- un système électronique qui récupère les informations données par le Silicium
- **un système de refroidissement au cuivre**



Exemple d'une configuration de tiroir étudiée

Propriété du LLR

Le but étant de reconstruire une gerbe électromagnétique du passage des particules et de l'énergie que celles-ci ont déposée à chaque passage dans une cassette.



Exemple d'une gerbe électromagnétique, trajectoire de gauche à droite

Les implications du service mécanique du LLR sont donc la conception générale des empilements (Choix des matériaux, de la géométrie, de l'assemblage ...), la réalisation de tests destructifs, d'irradiation et thermique, la réalisation d'études par éléments finis ainsi que l'élaboration de prototypes/démonstrateurs.

3. Système de refroidissement

Nous allons nous intéresser tout au long de ce rapport à la partie refroidissement du projet.

Pour obtenir un fonctionnement optimal, le silicium doit être refroidi à une température de -30°C , le système se chargeant de cette fonction se compose d'une plaque de cuivre dans laquelle serpente un tuyau qui assure l'écoulement de CO_2 diphasique.

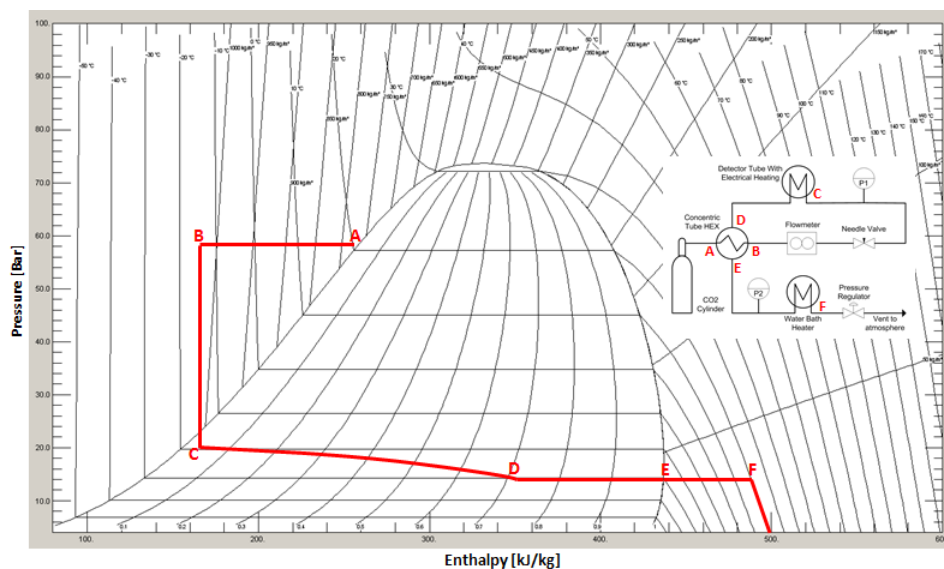


Diagramme thermodynamique pression/enthalpie, en rouge le suivi de notre cycle

Le CO₂ est conservé dans une bouteille sous forme liquide, puis il passe dans un échangeur (**segment AB**) où il subit une perte d'énergie isobare, il traverse ensuite un détendeur (**BC**), nous permettant d'atteindre une pression de 20 Bar et d'avoir un fluide à 95% liquide, celui-ci passe dans la plaque de cuivre (**CD**) d'où il extrait la chaleur puis ressort à une pression de 14 Bar et à 30% liquide.

Le fluide continue sa route en repassant dans l'échangeur (**DE**) lui permettant de récupérer une partie de l'énergie perdue précédemment, avant d'être réchauffé dans un bain d'eau (**EF**) puis évacué à l'air libre.

L'évolution au sein de la plaque de cuivre du fluide sur le palier diphasique de vaporisation nous garantit une stabilisation de sa température autour de -30 °C.

Une fois le silicium refroidit, il faudra maintenir sa température constante.

Le système de refroidissement devra donc être capable d'absorber la chaleur dissipée par les systèmes électroniques insérés dans les tiroirs, tout en garantissant un delta de température du Silicium inférieur à 1°C sur l'ensemble de la plaque pour limiter les courants de fuite.

4. Module de test des capacités de refroidissement

Ma mission au laboratoire est donc, en collaboration avec l'équipe, de mettre au point un module de test de nos capacités de refroidissement avec le système décrit ci-dessus.

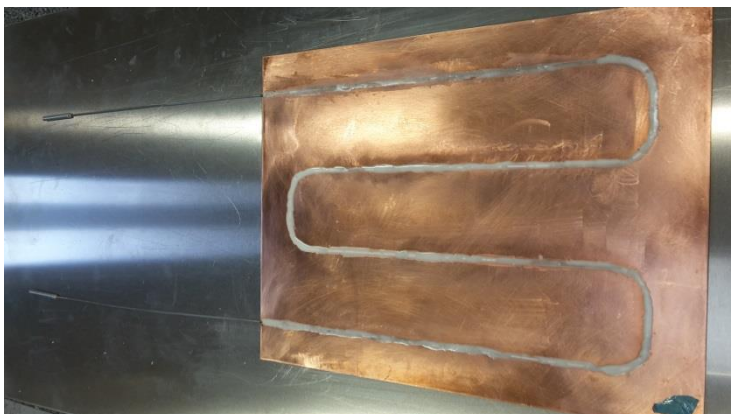
Pour cela, nous utilisons une plaque de cuivre de 30x40 cm.

Nous réalisons deux plaques équipées, le paramètre changeant étant la colle avec laquelle les tuyaux sont collés dans le cuivre. Une première plaque utilise de la colle chargée avec des billes d'alumine, l'autre des billes d'argent.

La deuxième possède une conductivité 10x supérieure à la première, cependant la différence de prix est conséquente, l'objectif sera donc également de déterminer quelle colle offre le meilleur rapport qualité/prix. Ainsi, si les résultats expérimentaux sont sensiblement les mêmes, nous ferons un choix économique.

Pour que le procédé soit validé, il faut que le système de refroidissement soit capable d'extraire une puissance dissipé d'environ 60W, répartie sur la plaque de cuivre.

Les composants électroniques sont simulés par des tapis chauffants et des résistances, pilotés par une alimentation, en parallèle, nous disposons équitablement sur la plaque 18 sondes Pt100 qui nous donneront la température exacte du cuivre.



Photographie de la plaque de cuivre avec le tuyau

II. Etude et réalisation de la plaque équipée

1. Description des composants

a. Tapis chauffants



Photographie d'un de nos tapis

Taille : 25 x 50 mm

Résistance : 115,2 ohms

Intervalle de température de fonctionnement : -60 à +200 °C

Fixation : support adhésif

Puissance nominale : voir annexe A

Pour convenir à notre utilisation, les tapis chauffants doivent être en état de fonctionnement à des températures avoisinant les -30°C, cet intervalle de température de fonctionnement nous le certifie. La fixation par un support adhésif est adapté pour notre simulation, on peut voir sur la documentation (annexe A) que celui-ci résiste à une température continue de +150°C, cependant le constructeur ne donne pas d'indication sur le comportement de l'adhésif dans les basses températures.

b. Résistances chauffantes



uk.rs-online.com

Taille : 30 x 16,5 x 8,8 mm

Résistance : 10 ± 5% ohms

Intervalle de température de fonctionnement : -55 à +200 °C

Comme décrit dans l'introduction, les résistances simulent des points chauds, nous devons donc avoir une forte dispersion de puissance sur un espace réduit, la transmission des 10W pouvant être produit à notre plaque de cuivre est assurée par le boîtier en aluminium.

c. Sondes Pt100



Photographie d'une des sondes Pt100

Taille : 2 x 2,3 x 1,4 mm

Support : céramique plat

Domaine de température : -100 à +500 °C

Type : F220 classe A

Montage : 4 fils

Précision : $\pm 0.20^{\circ}\text{C}$ à -30°C

La précision de la mesure des températures étant très importante dans notre cas, le choix de Pt100 de classe A s'impose (une classe B aurait doublé notre incertitude), le montage 4 fils est également de rigueur, afin de s'affranchir de la résistance des fils assez longs (environ 2m de câble).

Des sondes plus précises existent, pouvant avoir une précision sous les 0.1°C , mais disposer d'une sonde ultra précise n'est réellement utile que si l'équipement électronique est lui-même de précision suffisante, surtout si l'interface mécanique entre la plaque et la sonde n'est pas parfaite.

2. Montage électrique

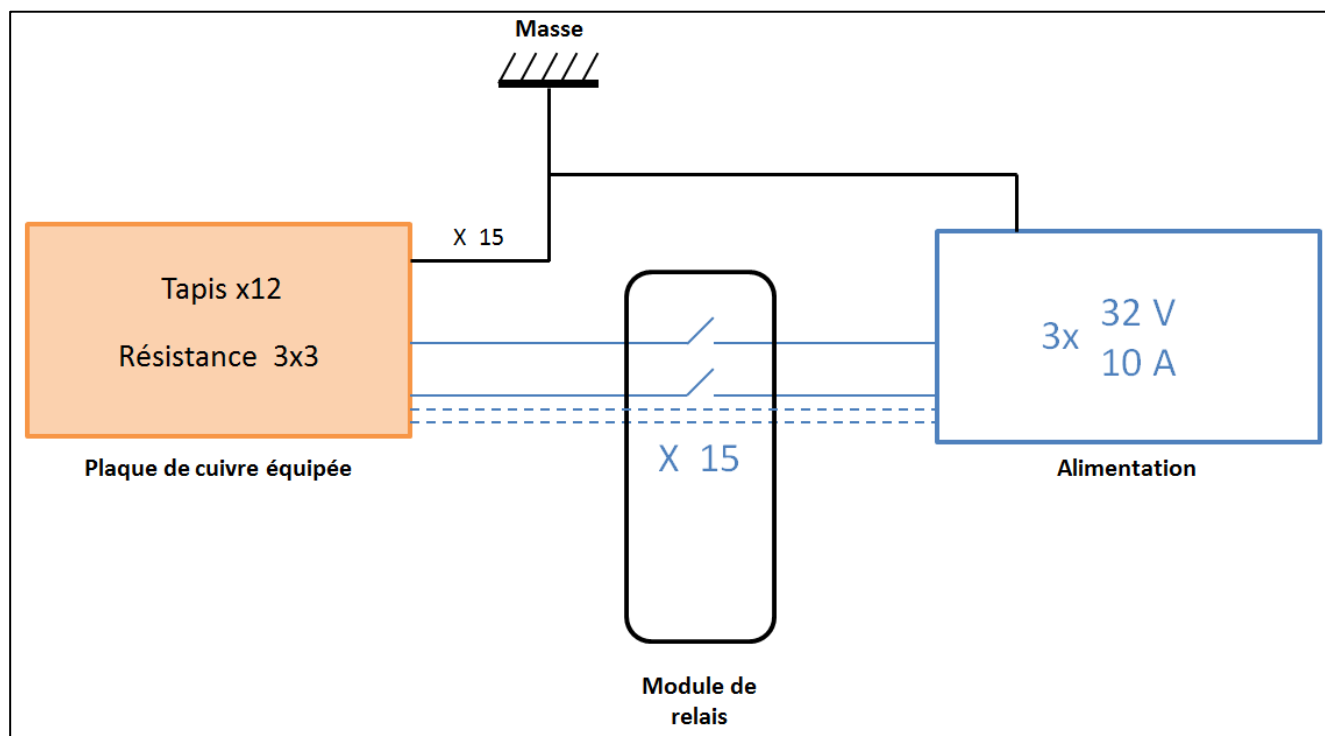


Schéma du montage électrique global simplifié

L'objectif étant de réussir à extraire 60W sur la surface de la plaque, nous avons choisi de pouvoir produire au maximum 160W de puissance à l'aide des tapis chauffants, sans compter les résistances.

Pour se faire, sont disposés sur la plaque 12 tapis chauffants et 9 résistances.

L'alimentation à notre disposition est équipée de 3 voies, pouvant produire chacune jusqu'à 32 Volts et 10 Ampères. Deux sorties sont montées en série, elles desservent l'ensemble des tapis, la troisième nous sert donc à alimenter les résistances.

Les tapis sont montés en parallèle, ainsi la puissance générée par l'alimentation est répartie équitablement entre les tapis que nous souhaitons allumer. Un module de relais nous permet d'allumer ou d'éteindre chaque tapis indépendamment. Les deux voies de l'alimentation produisent chacune 20 V et 4A, pour une puissance totale de 160W. Cette puissance sert de plafond pour le montage, elle peut être diminuée selon les besoins de l'expérience.

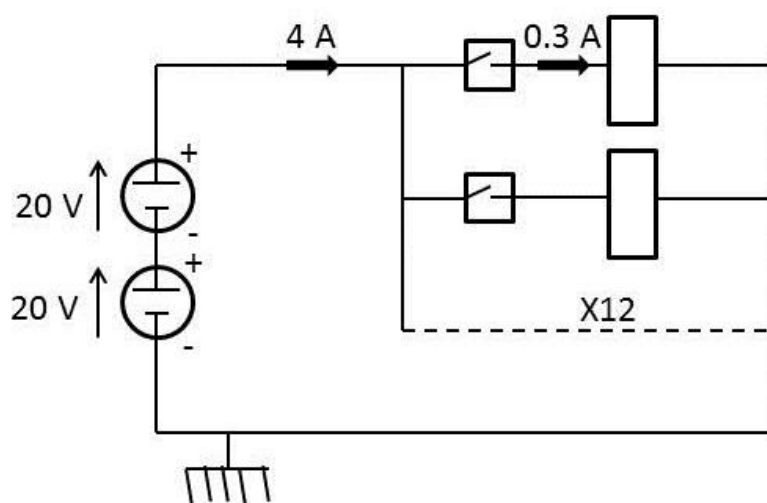


Schéma électrique du circuit des tapis chauffants

En ce qui concerne les résistances, nous avons opté pour un montage de trois branches en parallèle de trois résistances en série chacune, nous pouvons ainsi les piloter par ligne. Chaque résistance consomme 10W, soit une alimentation de 30V et 3A.

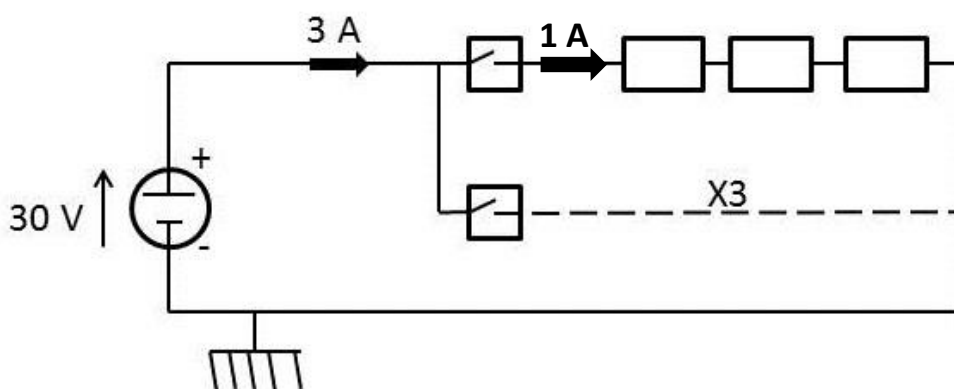
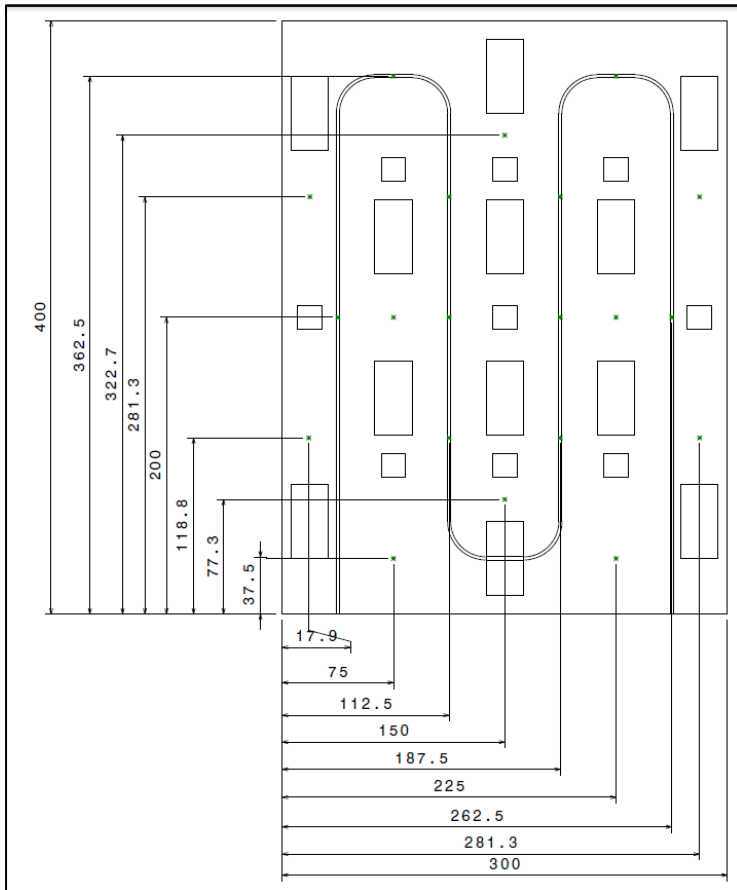


Schéma électrique du circuit des résistances chauffantes

3. Disposition des composants

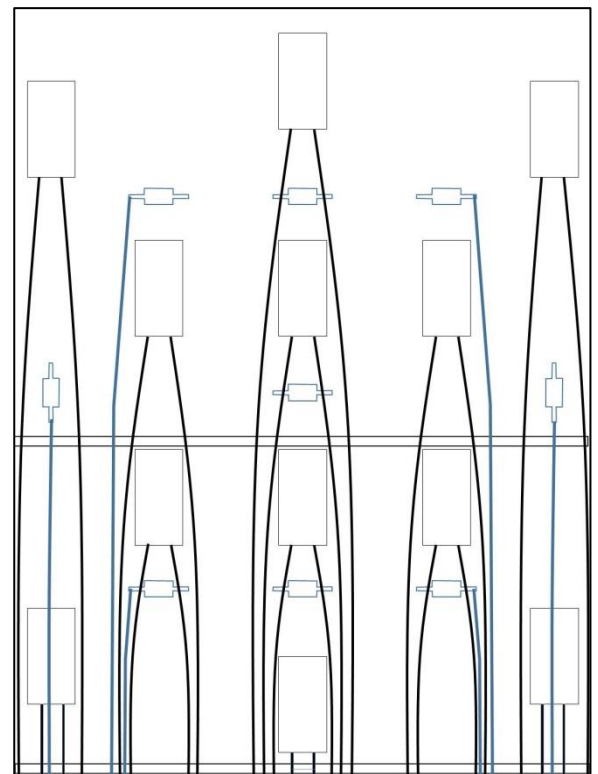


Plan de disposition des composants

Chaque composant ayant besoin d'une paire de câble, il a fallu imaginer un système de guidage des fils pour limiter l'encombrement :

Celui-ci est représenté par les deux rectangles qui traversent la plaque, il se compose simplement de plusieurs réglettes, qui viendront enfermer les fils pour les maintenir. Celles-ci sont fabriquées à l'aide d'une imprimante 3D.

Il faut rajouter, en plus, les fils qui relient les résistances en série horizontalement.



Plan de câblage

Pour que la simulation soit des plus proches de la réalité, nous avons disposé les composants de manière équitable.

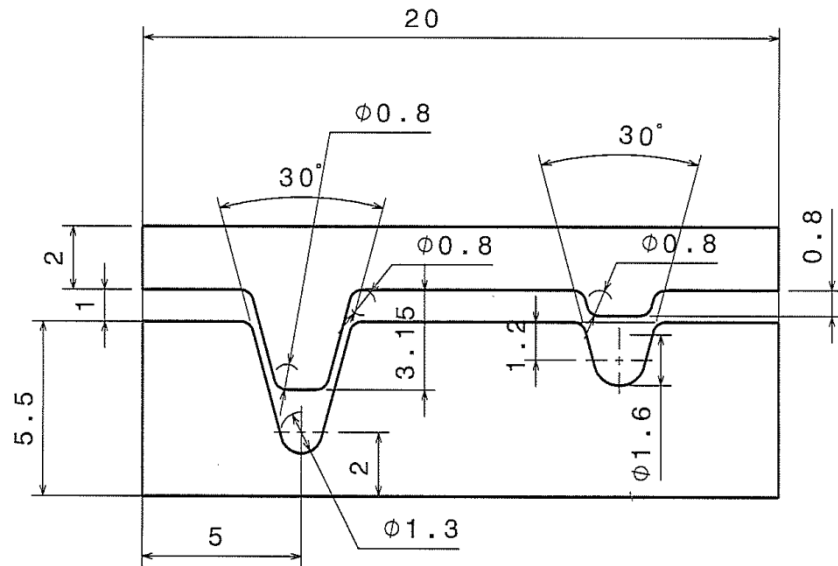
Les carrés représentent les résistances, les rectangles sont les tapis et les points verts indiquent la position des Pt100 (qui sont disposées de l'autre côté de la plaque).

On peut également voir la position du tuyau.

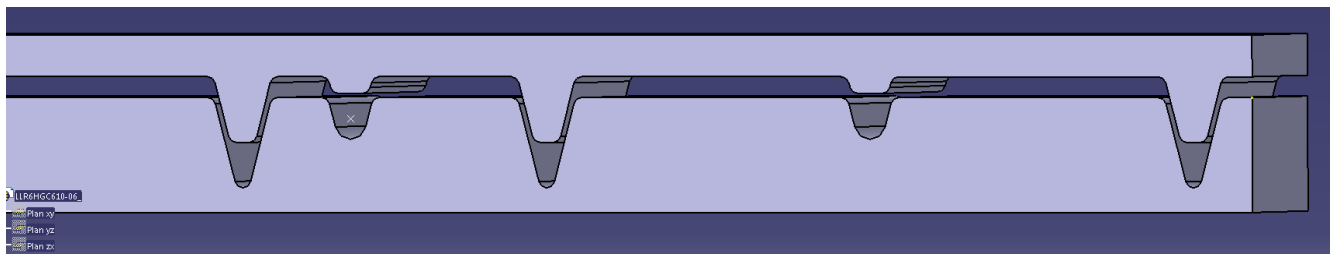
Certaines sondes sont disposées en face du tuyau, de façon à avoir une indication de la température du fluide, car nous n'avons pas de moyens de la connaître exactement. En revanche les températures d'entrée et de sortie du CO₂ seront connues.

Réalisation des réglettes :

La conception 3D est faite à partir du logiciel Catia, légèrement différent de SolidWorks que j'ai utilisé à l'IUT. En partant du plan présenté ci-dessous et du plan de câblage vu juste avant, j'ai donc dessiné puis imprimé les 6 réglettes, que nous avons ensuite percé pour y insérer des vis.



Plan de base



Simulation sous Catia



Photographie d'un bout de réglette après impression et perçage

Fixation des composants :

Les tapis sont uniquement collés grâce à l'adhésif prévu dessus, quant aux résistances et réglettes, elles sont collées à l'aide d'une colle Epoxy, résistante aux températures que nous allons atteindre.

Les Pt100 sont disposées sur un point de pâte thermique, puis scotché à l'aide d'un scotch isolant électrique mais non thermique. Il est important d'isoler les pâtes des Pt100 de la plaque de cuivre conductrice tout en ayant une bonne conductivité thermique sur les matériaux qui l'entoure.

III. Programmation de la communication à distance

1. Description du matériel

a. Alimentation HAMEG



Nom : HAMEG HMP4030

Limite par voie : 32V / 10A / 160W

Limite totale : 384W

Communication : GPIB / Ethernet / USB

<http://www.farnell.com/datasheets/1808907>

La puissance maximale délivrable par cette alimentation nous est suffisante, et le fait que chaque voie soit pilotable indépendamment convient parfaitement à notre utilisation. Son grand nombre de modes de communication nous permet de palier à toutes éventualités.

b. Châssis PXI



Nom : NI Pxie-1071

Emplacement : 3

Bande passante : 3Go/s

Communication : Ethernet / USB

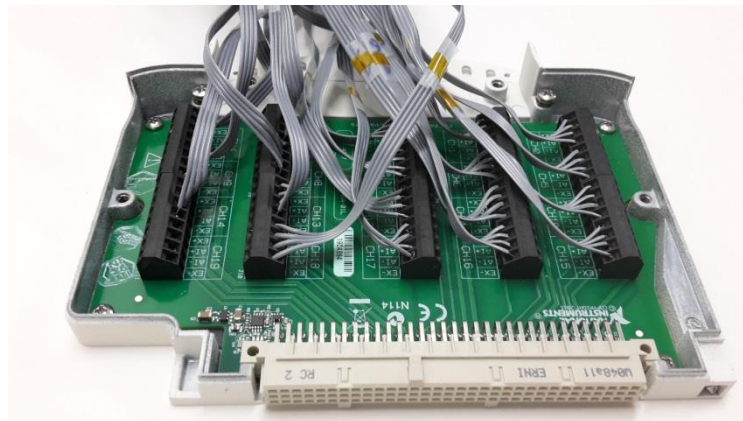
<http://www.ni.com/datasheet/pdf/en/ds-257>

Elément central des traitements de données, le châssis Pxie de National Instrument nous permet de faire le pont entre la plaque de cuivre et l'ordinateur. Totalement compatible avec LabVIEW, il nous permet de prendre les mesures de résistance des Pt100 et de transmettre les températures équivalentes à l'opérateur.

L'intérêt principal de ce matériel est qu'il possède son propre mini-ordinateur, pouvant ainsi enregistrer et faire tourner des programmes LabVIEW.

Nous venons « plugger » 2 modules décrits ci-dessous, en lien avec les sondes Pt100 et les relais.

c. Carte d'acquisition des températures

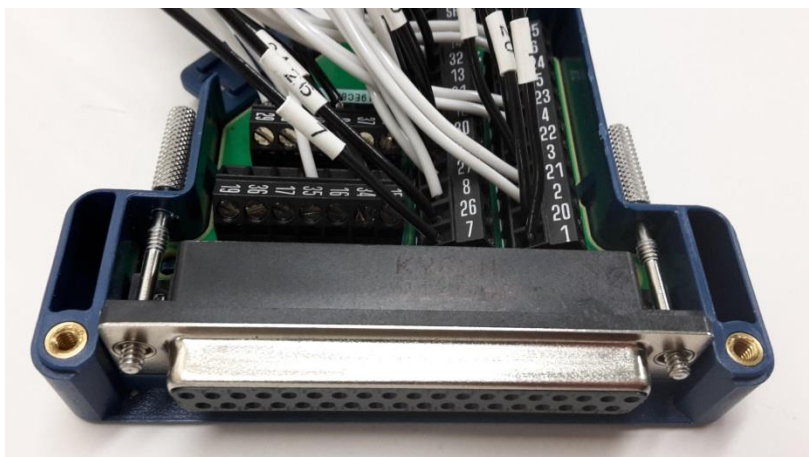


Photographie de la partie extérieur de la carte d'acquisition

On peut voir sur la photographie les branchements 4 fils des 18 sondes Pt100, chaque nappe doit être étiquetée car nous avons plusieurs mètres de câbles avant d'arriver sur les Pt100, ainsi les données récupérées pour une chaîne correspondent toujours à la même sonde.

Cette carte peut être retirée du châssis facilement, à chaud (lorsque le châssis est en train de travailler) sans aucun risque, ce qui nous permet une manipulation beaucoup plus simple et rapide.

d. Module de relais



Photographie du module de relais

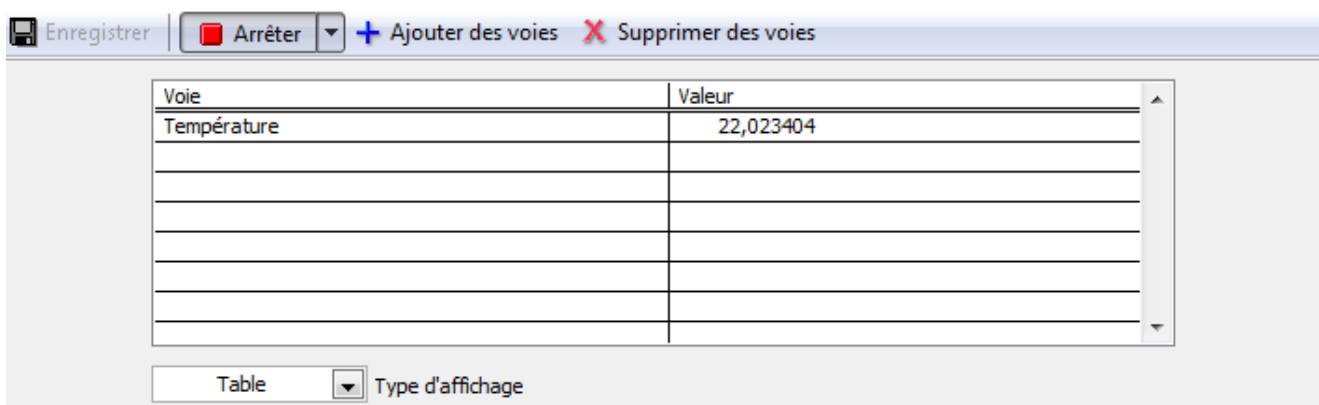
Egalement pilotable par le châssis Pxi, ce module de relais nous permet d'alimenter les 12 tapis et 3 lignes de résistances indépendamment, LabVIEW reconnaît ce type de matériel et nous permet une configuration grâce à des fonctions « switch on » ou « switch off ».

Chaque relais possède une limitation de 1A qu'il faudra surveiller lors de nos manipulations, en effet une erreur de configuration peut amener à suralimenter une voie.

2. Programmation sous LabVIEW

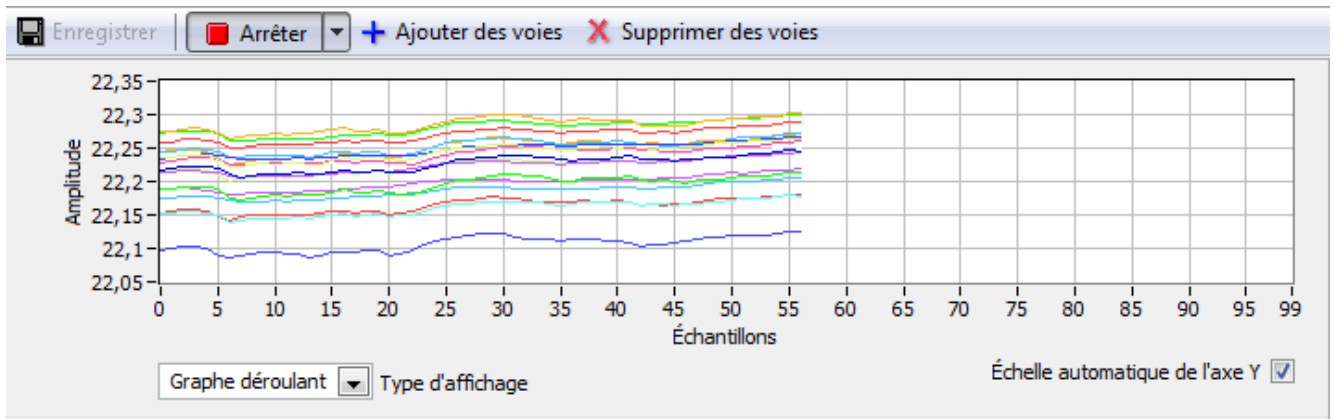
Ma principale mission au laboratoire a donc été d'initialiser la communication entre les appareils ainsi que de programmer le module de test final.

Pour ce faire, il a fallu commencer par communiquer avec le châssis, sous le logiciel *NI Max* dans un premier temps, le but étant d'arriver à obtenir la température d'une sonde en direct :



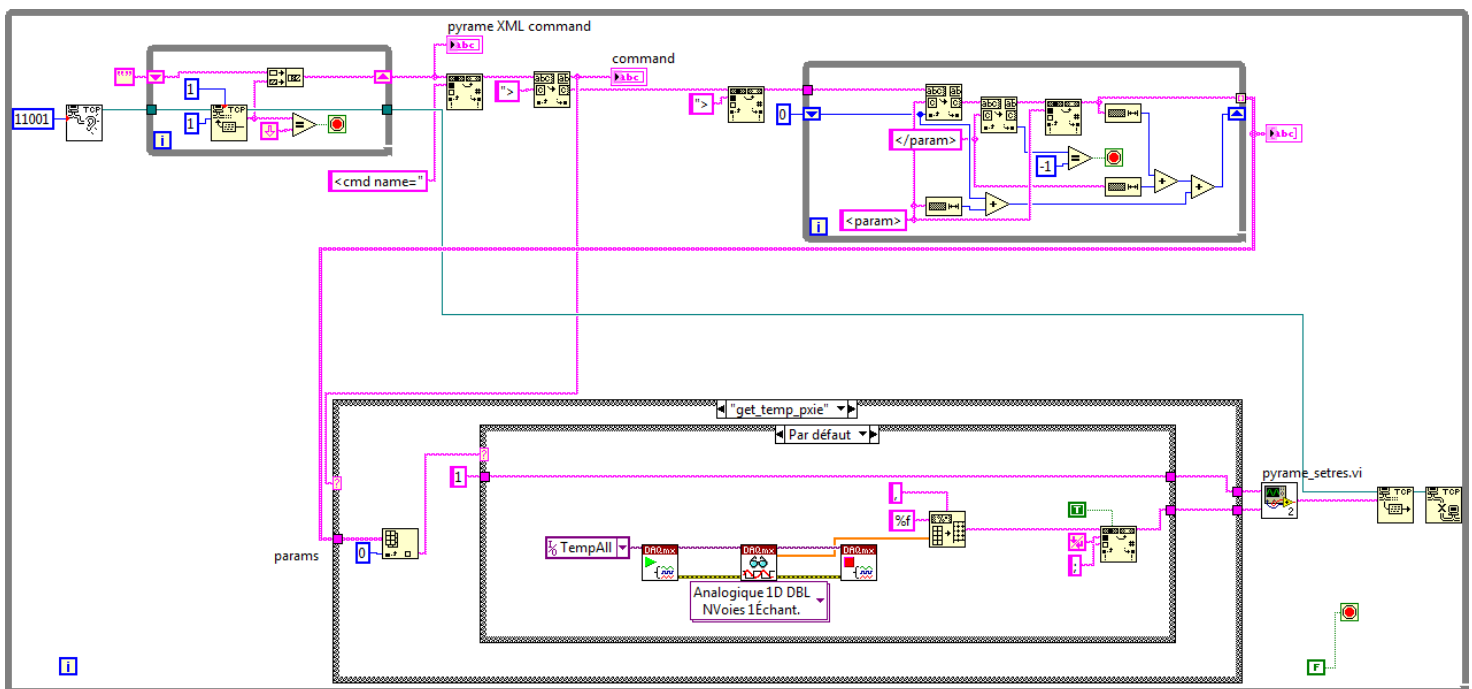
Capture d'écran du logiciel NI Max, première prise de température

Puis de construire un graphe déroulant comprenant l'ensemble des sondes :



Capture d'écran du logiciel NI Max, 18 sondes sur graphe déroulant

NI Max utilisant le système de « tâche » pour communiquer avec le châssis, j'ai donc cherché à reproduire cette fonction sous LabVIEW, il faut tout d'abord ouvrir une connexion entre l'ordinateur et le périphérique, branché via Ethernet, puis créer la tâche de lecture de température et l'utiliser grâce à la palette Daqmx .



Capture d'écran du diagramme du programme LabVIEW final

Le premier bloc (en haut à gauche), nous permet d'initialiser l'écoute de communications sur le port TCP choisi (on peut voir le port 11001), ce programme est donc à l'écoute de ce port et chaque information qui y passera sera traitée par la boucle. La partie du rapport suivante décrira avec plus de précision les syntaxes d'envoi de commande, mais on peut déjà distinguer (entre le premier et le deuxième bloc) les fonctions nous servant à isoler les commandes.

Ces fonctions sont suivies par le deuxième bloc (en haut à droite) qui détermine les paramètres des précédentes commandes. Nous avons donc, pour résumer, un ordre (la commande), et la ou les cible(s) de cet ordre (les paramètres).

Ces informations cheminent ensuite vers le dernier bloc (en bas), qui reconnaît les différentes commandes, sur cette capture d'écran on peut voir la commande « `get_temp_pxie` », qui correspond à la prise de température, et qui applique cet ordre à une certaine voie, ici « Par défaut », qui correspond à l'ensemble des voies. En sortie de programme, nous avons donc les températures des 18 sondes, qui sont envoyées dans le port 11001 afin de pouvoir les récupérer sur n'importe quel ordinateur branché en Ethernet.

La deuxième commande utilisée dans notre banc de test est « `set_relays_pxie` », qui sert à configurer la matrice d'ouverture/fermeture des relais.

Ce programme LabVIEW est ensuite chargé dans le châssis Pxie, pour que celui-ci l'exécute à chaque allumage, cette étape est réalisée grâce aux fonctions LabVIEW qui permettent de compiler et déployer sur le châssis un instrument virtuel (VI).

Une fois ces étapes effectuées, le système est totalement indépendant, sauf modification de programme, il n'est plus nécessaire d'utiliser LabVIEW et la mise en route du banc de test n'est donc plus soumise à la condition d'un ordinateur sous licence LabVIEW.

TCP : Transmission Control Protocol (littéralement, « protocole de contrôle de transmissions »)

3. Pilotage sous Python et Pyrame

Une fois le programme LabVIEW final chargé dans le châssis, il faut piloter ce dernier, grâce à un outil développé par les ingénieurs de recherche du service informatique, le protocole Pyrame, basé sur du XML.

Il permet, à travers le langage python le plus souvent, d'envoyer des commandes au châssis pour que celui-ci fasse tourner le programme LabVIEW, récupère les données demandées, et nous les transmette dans un langage compréhensible via TCP.

Référence Pyrame :

Publication :

Frédéric Magniette et al 2015 J. Phys. : Conf.Ser.664 082028

Site web: <http://lir.in2p3.fr/sites/pyrame/>

Cette partie demandant des connaissances de programmation interne au laboratoire, j'ai travaillé une grande partie de mon stage avec des ingénieurs de recherche du service informatique/instrumentation pour suivre leur politique de programmation, et ainsi produire un travail s'inscrivant dans le temps et dans les accoutumées.

En utilisant une machine Linux et un terminal de commande, on peut communiquer avec le châssis avec la commande « `chkpyr2.py 10.220.0.3 11001` », correspondant donc à la fonction pyrame qui construit la commande en format XML, puis à l'adresse IP du châssis et enfin au port Ethernet utilisé, il suffit ensuite d'y ajouter la commande et les paramètres souhaités, exemple :

Sur le terminal :

```
# chkpyr2.py 10.220.0.3 11001 get_temp_pxie all  
Retcode=1 res=22.35,22.39,22.32,22.36 ....
```

La première ligne correspondant à l'envoi et la seconde au retour

Le programme Labview reçoit ceci :

```
<cmd name= <get_temp_pxie><param>all</param></cmd>
```

D'où les différentes fonctions servant à isoler les commandes en se basant sur le langage récurrent de type `<param>` et d'autres (bloc n°2)

Ici on envoie une ligne de commande composé de la fonction, de l'adresse IP, du port, de la commande et du paramètre. Cette ligne va ensuite être transférée dans le programme LabVIEW interne au châssis via TCP, qui va prendre la commande et le paramètre (identiques à ceux présentés plus haut) et renvoyer une chaîne de caractères composée des 18 températures de Pt100. Cette chaîne, via le même protocole de transmission, est transmise à l'ordinateur qui nous l'affiche. Le programme est prévu pour pouvoir prendre une seule sonde, en changeant « all » par « x » où x est le numéro de la sonde (de 0 à 17) qui nous intéresse.

Cet exemple est en pyrame pur, mais pour pouvoir faire une utilisation plus évoluée du système, il faut écrire un programme en python qui va faire le travail à notre place, à savoir envoyer la commande, traiter les données (par exemple déterminer si les sondes sont stables, faire une moyenne des températures...), et les enregistrer dans un tableur Excel.

Programme complet en **annexe B**.

Il nous permet, de manière générale, de programmer différentes phases thermiques lors d'un essai. La première étant la phase de stabilisation, où le programme récupère les températures et détermine si celles-ci ne varient pas trop au cours du temps. Lorsque la plaque est considérée comme stable, le programme allume l'alimentation, c'est la phase de chauffage, puis la 3^{ème} phase est de nouveau une attente de stabilisation, alimentation toujours allumée.

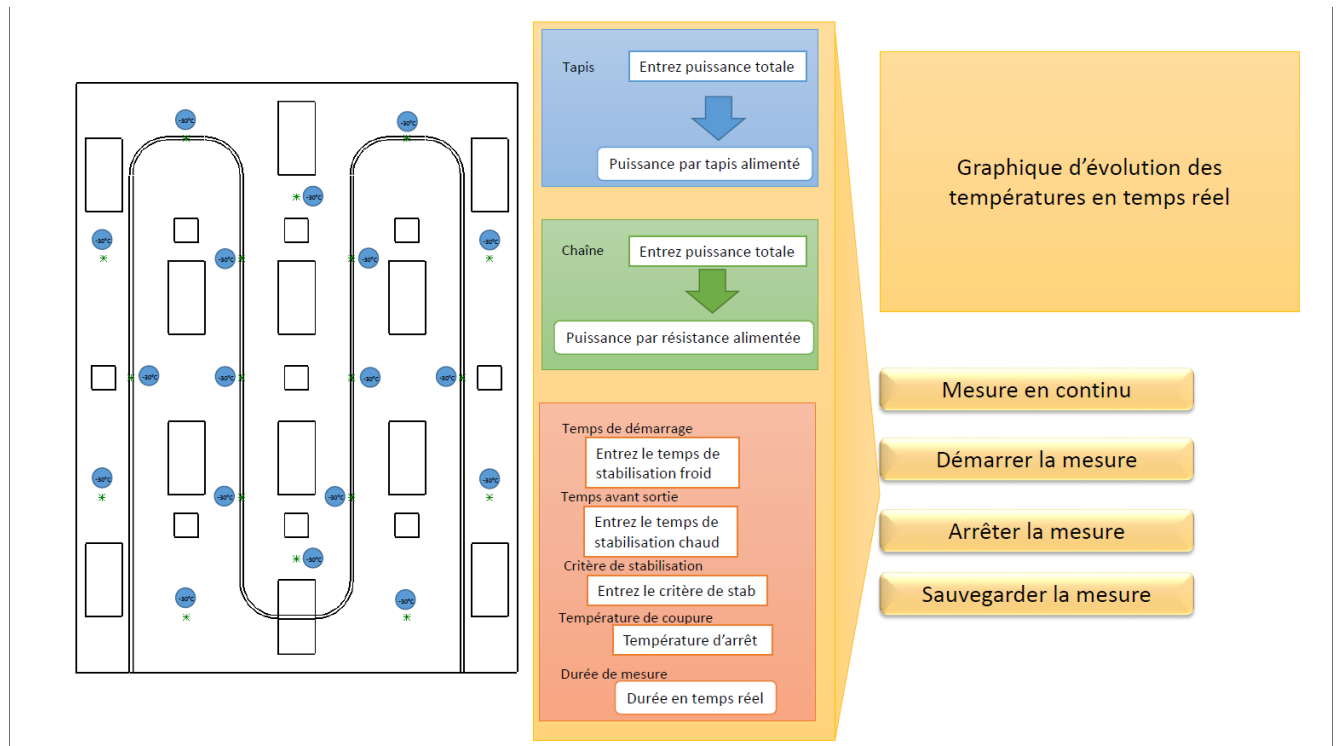
En entrée de programme, il faut indiquer le ou les tapis que l'on souhaite allumer ainsi que la puissance totale en Watt que l'on souhaite envoyer dans les tapis chauffants, le programme se charge de déterminer la résistance équivalente pour calculer l'ampérage et le voltage que l'alimentation doit produire.

Pour résumer la fonction de chaque bloc (délimité par un #) :

- **# Set relays based on power matrix** récupère les positions (en booléens) de chaque relai, 0 signifie fermé et 1 ouvert, ces informations sont ensuite transmises au programme LabVIEW interne pour que celui-ci ouvre ou ferme les relais.
- **# Current/Voltage calculation** calcul, à partir de la puissance souhaitée et du nombre de tapis ouverts, la résistance équivalente et les intensité/tension à fournir.
- **# Initialization of module PS** ouvre une connexion entre l'ordinateur et l'alimentation
- **# Power off PS** nous assure que l'alimentation est éteinte lors de la première phase de stabilisation.
- **# Set current limit on PS** envoie l'information du courant souhaité aux différentes voies de l'alimentation.
- **# Set voltage on PS** fait de même avec la tension.
- **# Take start time** et **# First phase** permette respectivement de démarrer une horloge virtuelle et d'annoncer le début de la première phase de stabilisation.
- **# Buffers for stabilization criteria** calcul en temps réel l'écart de mesure maximal sur un certain nombre de valeurs, afin de déterminer s'il y a stabilisation ou non.
- **# Main loop** contient la boucle qui récupère les températures en temps réel, de plus, nous imprimons ces informations dans le terminal ainsi que dans un tableur Excel, lorsque la stabilisation est atteinte :
- **# Power on PS** démarre l'alimentation et la chauffe commence, quand la deuxième stabilisation est repérée :
- **# Power off PS** coupe l'alimentation.
- **# Deinitialize modules** coupe la connexion entre alimentation et ordinateur.

PS : Power Supply (alimentation)

Ce programme sera par la suite inséré dans un **GUI *Graphical User Interface***, « interface graphique », qui pourrait ressembler à ceci :



Aperçu du GUI provisoire

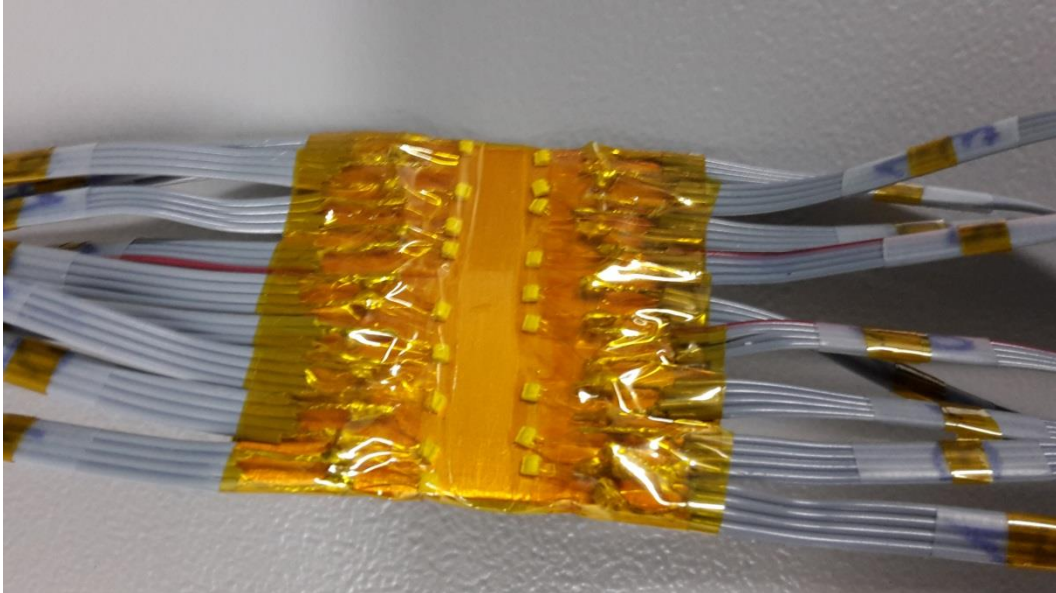
On peut y voir une représentation de la plaque et des différents composants, un simple clic sur un tapis ou une chaîne de résistance permettra de les allumer/éteindre, et la température des sondes sera indiquée à l'emplacement virtuel de celle-ci (ronds bleus). Il faudra ensuite rentrer la puissance souhaitée, les différents critères de stabilisation et le programme affichera un graph déroulant des températures.

IV. Prise des premières mesures

1. Etalonnage des sondes Pt100

Pour être certain du bon comportement des sondes Pt100, j'ai effectué un étalonnage de l'ensemble des sondes dans une enceinte thermo régulée et étalonnée.

Pour obtenir une prise de mesure homogène et le plus proche de la réalité, j'ai donc scotché les sondes sur une plaque de cuivre :



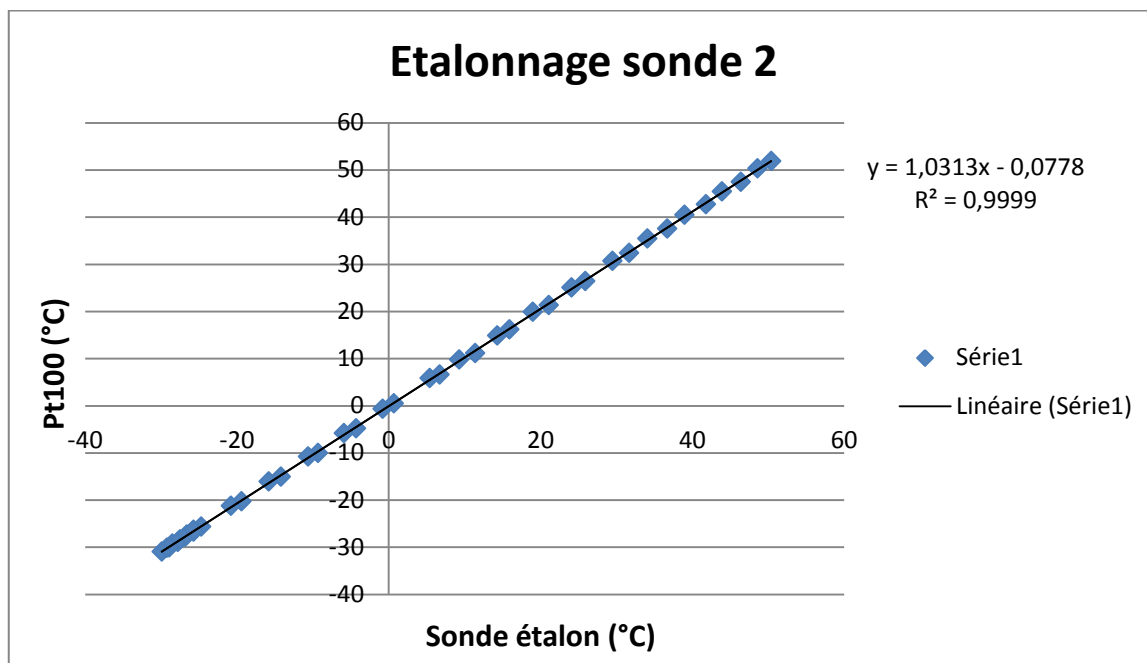
Photographie du montage effectué pour l'étalonnage

La sonde étalonnée est ensuite scotchée sur la plaque de cuivre.

Le montage a subi deux cycles thermiques, de +50°C à -30 °C, puis de -30°C à +50°C avec une prise de mesure tous les 5 degrés, ou 1°C dans les très basses températures, l'idée étant de reproduire les températures que la sonde pourra rencontrer au cours de son utilisation.

J'ai ensuite effectué une régression linéaire sur chaque sonde, en prenant les points de l'aller et du retour, afin de limiter le phénomène d'inertie thermique.

Cette équation de droite nous servira donc à corriger la température des Pt100.



Régression linéaire pour une Pt100 (étalonnage à l'enceinte)

On remarque que, comme attendu, le coefficient de corrélation linéaire est très bon, ce qui est plutôt rassurant car une correction polynomiale aurait été plus compliquée, la pente est en revanche relativement éloignée de 1, ce qui prouve que la correction s'impose. En annexe C et D sont présentées le tableau de valeurs pour quelques sondes ainsi que l'ensemble des équations de correction obtenues.

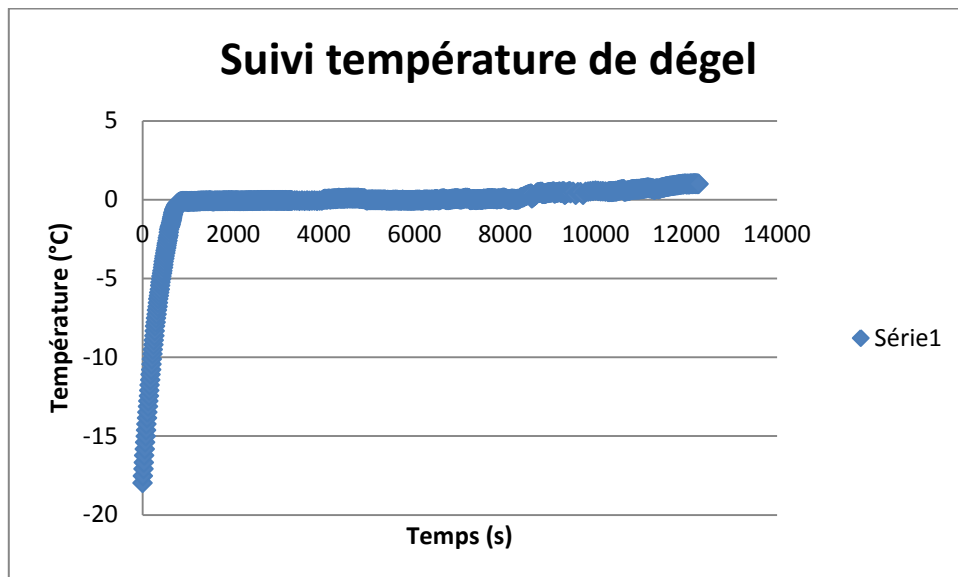
Pour vérifier le décalage à 0 des droites obtenues, nous avons imaginé un petit test.

2. Test de précision à la température de dégel

La température de gel de l'eau étant constante quelle que soit les conditions ambiantes (contrairement à la température d'ébullition qui varie fortement en fonction de la pression atmosphérique), un test en eau gelée est parfait pour vérifier la précision de nos sondes à 0°C, à condition d'utiliser de l'eau déionisée.

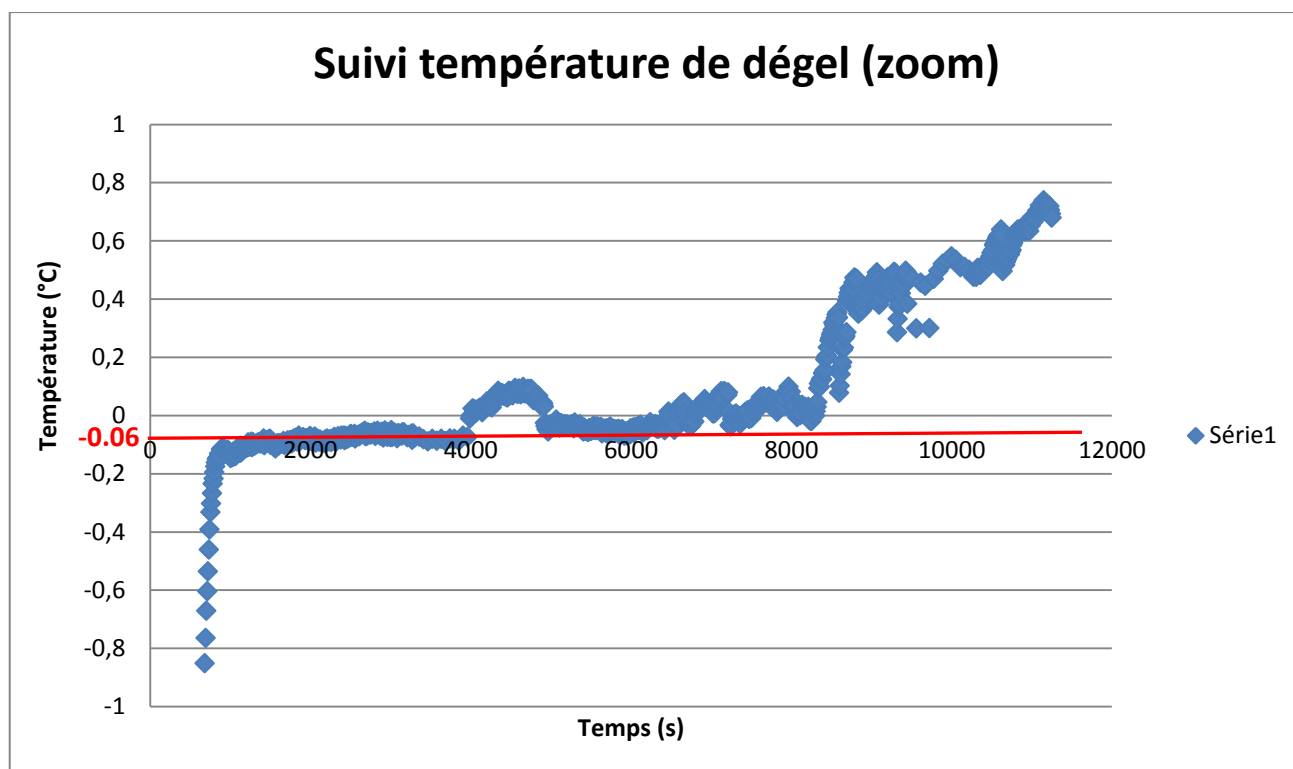
Initialement, l'idée était de tester la Pt100 dans un bain eau+glace, mais difficile d'être sûr de l'homogénéité lors de ce genre de test, nous avons donc placé la Pt100 dans un volume d'eau puis gelé le tout au congélateur à -30°C.

J'ai ensuite mis en place un programme de mesure de température, qui a pris une mesure toutes les 10 secondes tout au long de la fonte du bloc de glace, la température de dégel étant identifiable grâce à l'apparition d'un palier (la température stop sa montée durant plusieurs minutes).



Graphique du suivi de la température de dégel

Le test a duré plus de 3 heures, le phénomène de palier est clairement mis en évidence mais difficilement observable à cette échelle, en jouant sur celle-ci on obtient :



Graphique zoomé du suivi de la température de dégel

La température de dégel a donc été estimée à $-0,06^{\circ}\text{C}$, très proche, donc, du $-0,078^{\circ}\text{C}$ vu précédemment avec la sonde n°2.

Les droites de correction semblent donc être totalement légitimes pour corriger nos Pt100 à l'avenir.

V. Mise en condition expérimentale

Le test final de nos capacités de refroidissement doit être effectué à Lyon, dans des locaux possédant une installation régulée de CO₂, mais les délais étant trop longs pour insérer cette étape dans mon stage, nous avons donc imaginé dans un premier temps un test comparatif de l'efficacité de chaque colle, ainsi qu'un test nous permettant d'avoir un suivi temporel de la réaction de la plaque à une impulsion de chauffe.

Test comparatif :

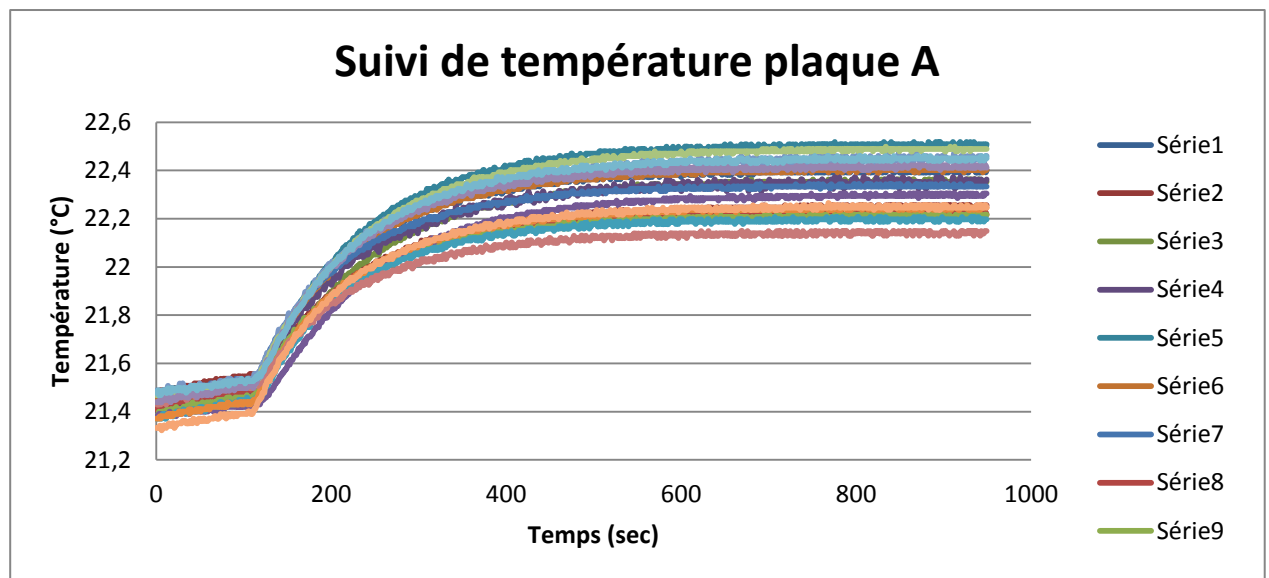
Pour comparer l'efficacité des colles, nous avons fait s'écouler de l'eau à température ambiante dans les tuyaux, en ayant en parallèle une chauffe des tapis.

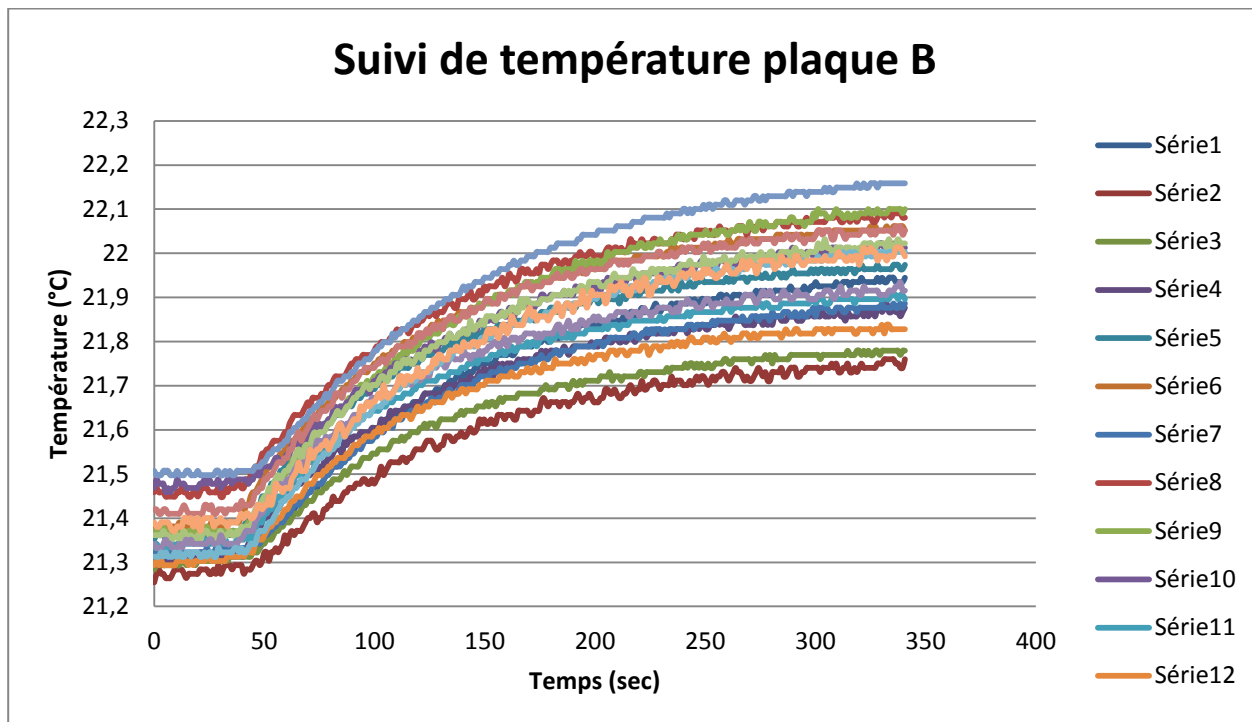
Protocole expérimental :

Les tuyaux sont branchés sur un circuit de pompage d'eau, ainsi, le débit volumique reste constant durant les expériences, une fois que le circuit est mis en route, on attend quelques minutes l'équilibre thermique, puis on lance notre programme. Celui-ci vérifie que l'équilibre est atteint, puis allume l'alimentation à une puissance de 10W répartie sur les 12 tapis (l'équilibre est déterminé par une variation de température inférieure à 0.03°C dans une période de 20sec), le programme attend ensuite le deuxième équilibre thermique, alimentation toujours allumée.

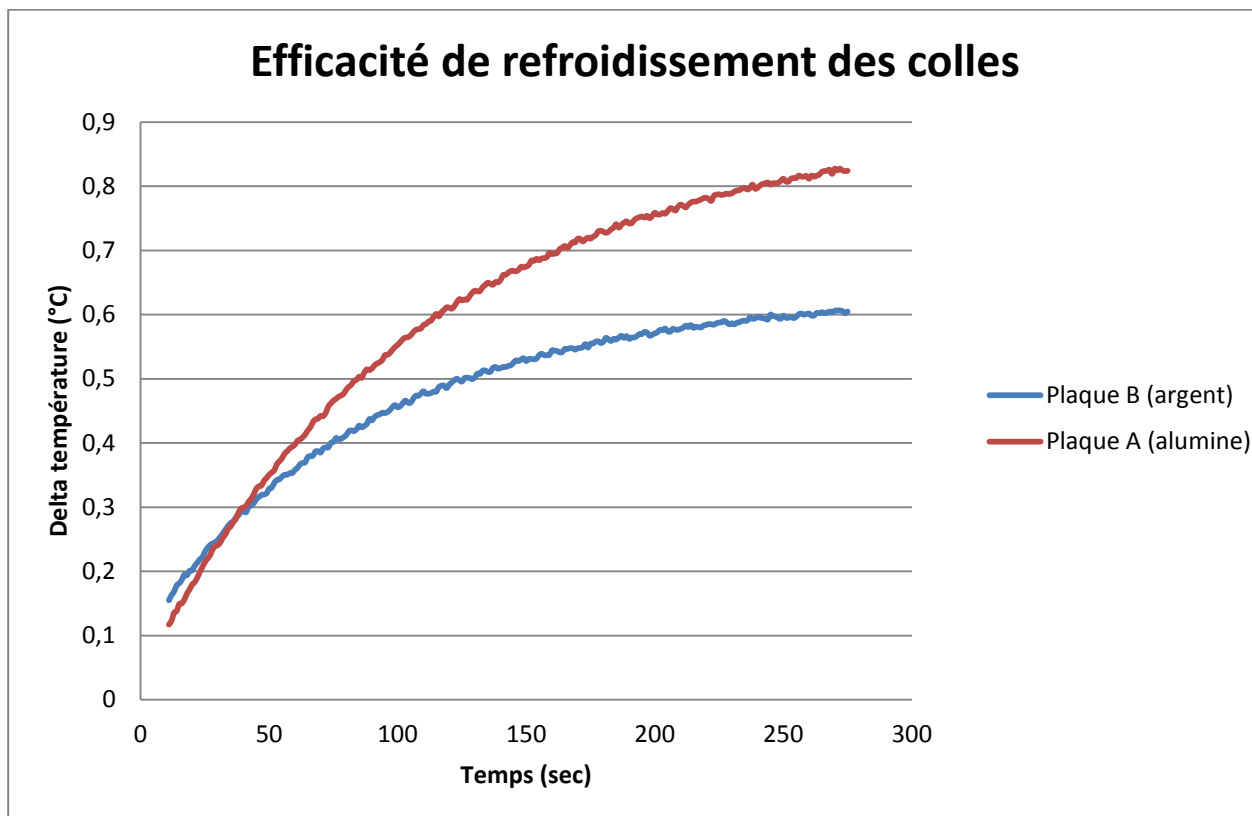
Durant ces cycles thermiques, un suivi des températures est effectué toutes les 1 sec, le tout consigné dans un tableur Excel.

En sortie d'expérience, nous obtenons les deux courbes suivantes (plaque A et B) :





La température ambiante ayant légèrement changé entre les 2 tests, on résonne en différence de température, et en moyennant les températures des 18 sondes, ainsi nous obtenons le graphe comparatif suivant :



Comme attendu, la colle chargée en bille d'argent, qui possède une conductivité plus élevée, nous permet d'obtenir un refroidissement plus efficace. La différence de température des plaques est d'approximativement 0.2 °C pour une montée en température inférieure à 1 °C. Ce qui amène donc, pour ce test, une variation d'efficacité d'environ 20%.

Il faudra effectuer plusieurs tests, par exemple avec de l'eau proche de 0 °C, pour comparer les variations d'efficacité.

Test de corrélation entre modèle mathématique et expérimental :

Les logiciels de simulation permettent d'obtenir des approximations de la réponse de notre plaque de cuivre à différentes impulsion électrique. Cependant il reste difficile de simuler parfaitement le comportement des tapis chauffants, des résistances, ou bien même d'une plaque de cuivre. En effet il peut y avoir par exemple des aspérités sur les contacts mécaniques entre les composants, ou des variations de résistances dans ces derniers.

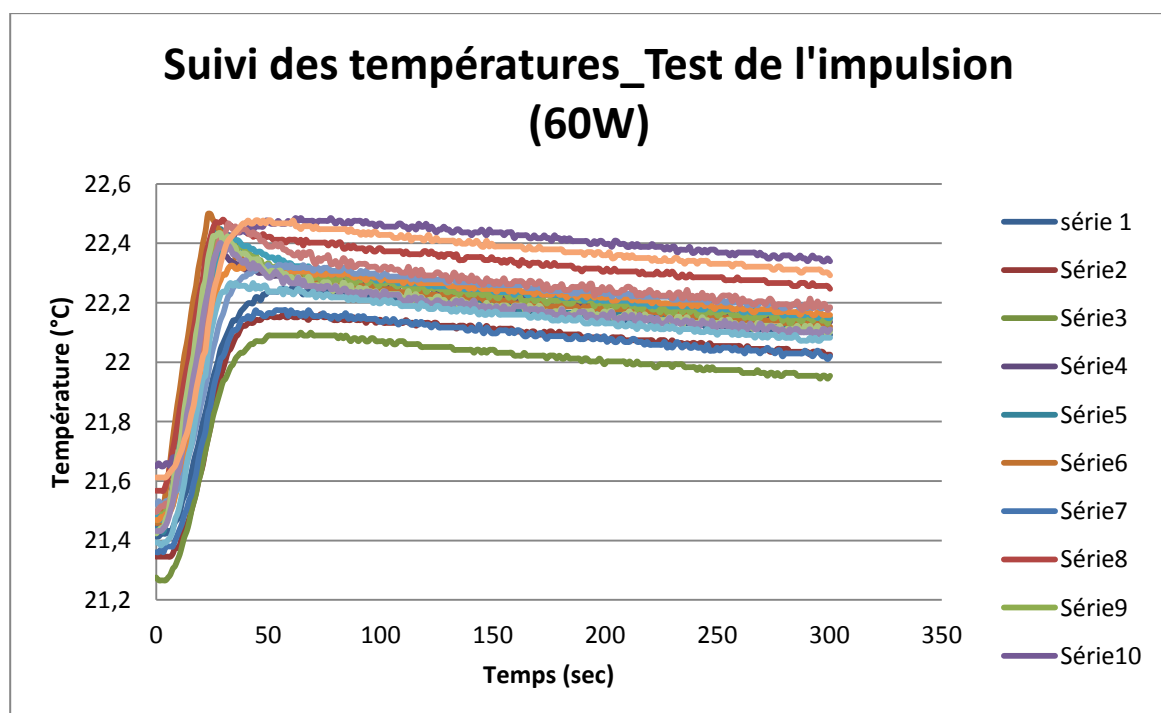
Le but de l'expérience suivante est donc d'obtenir une bonne corrélation entre la simulation mathématique et les conditions expérimentales.

Pour cela, nous n'avons pas besoin du système de refroidissement.

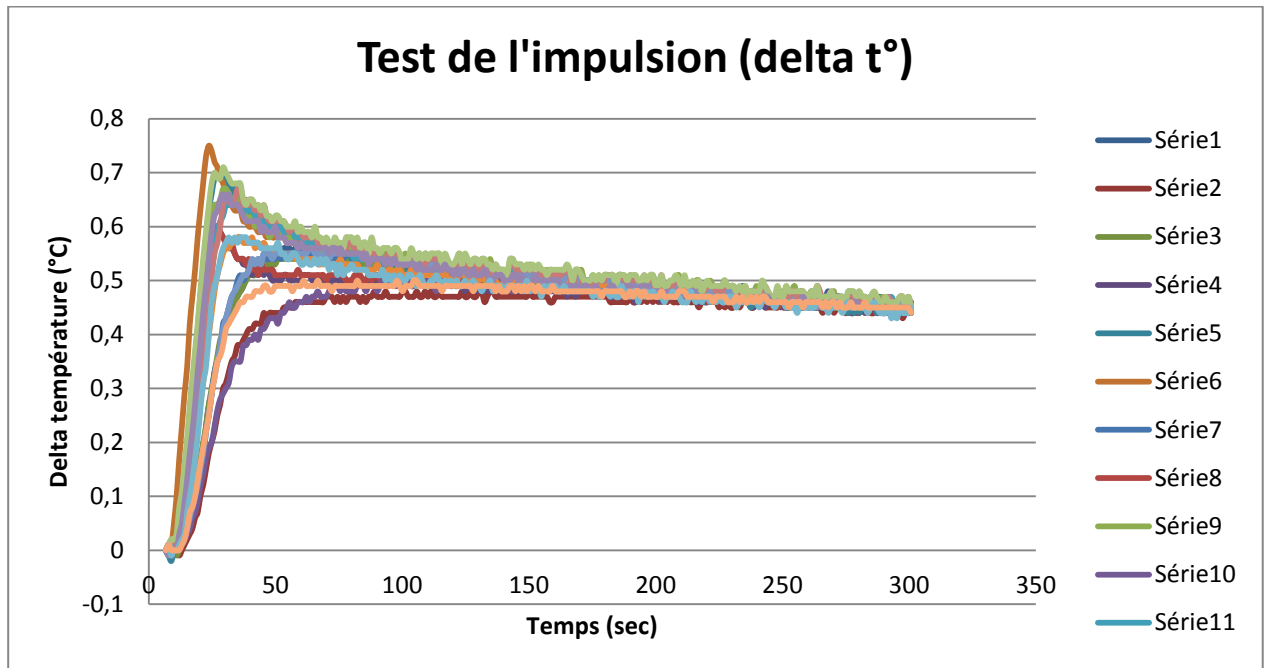
Protocole expérimental :

La plaque de cuivre est soumise à une impulsion de puissance chauffante, après établissement de l'équilibre thermique, une puissance de 60W répartie sur les 12 tapis est envoyée pendant une courte période de 20 secondes, un suivi des températures a lieu chaque secondes avant et après la chauffe, on pourra donc visualiser le régime de chauffe ainsi que le régime de retour à la température ambiante.

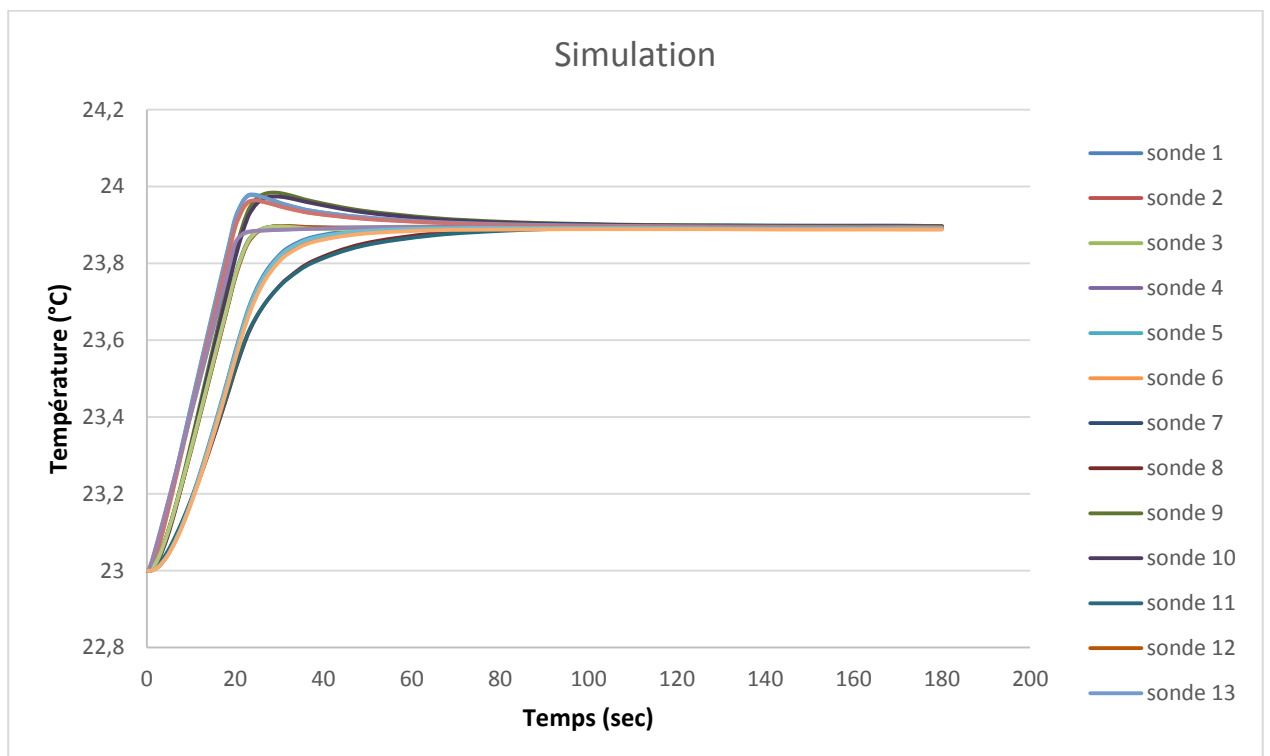
Résultat du test :



Le résultat n'est pas celui que l'on attendait, en effet, sur la simulation, les températures durant le régime de stabilisation à température ambiante converge, on remarque par ailleurs que l'écart des températures avant le test est assez conséquent. On décide donc de résonner en delta de température comme précédemment, afin d'obtenir la courbe ci-dessous, plus représentative.

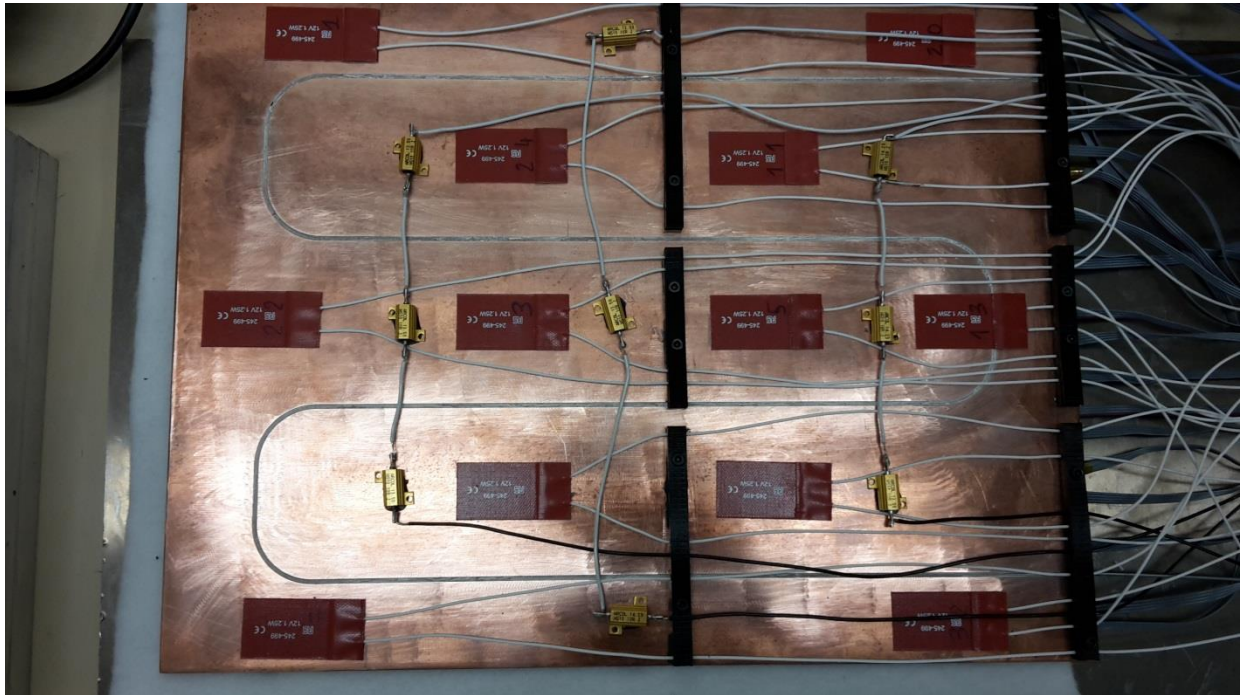


Nous obtenons ainsi un résultat proche de la simulation suivante :

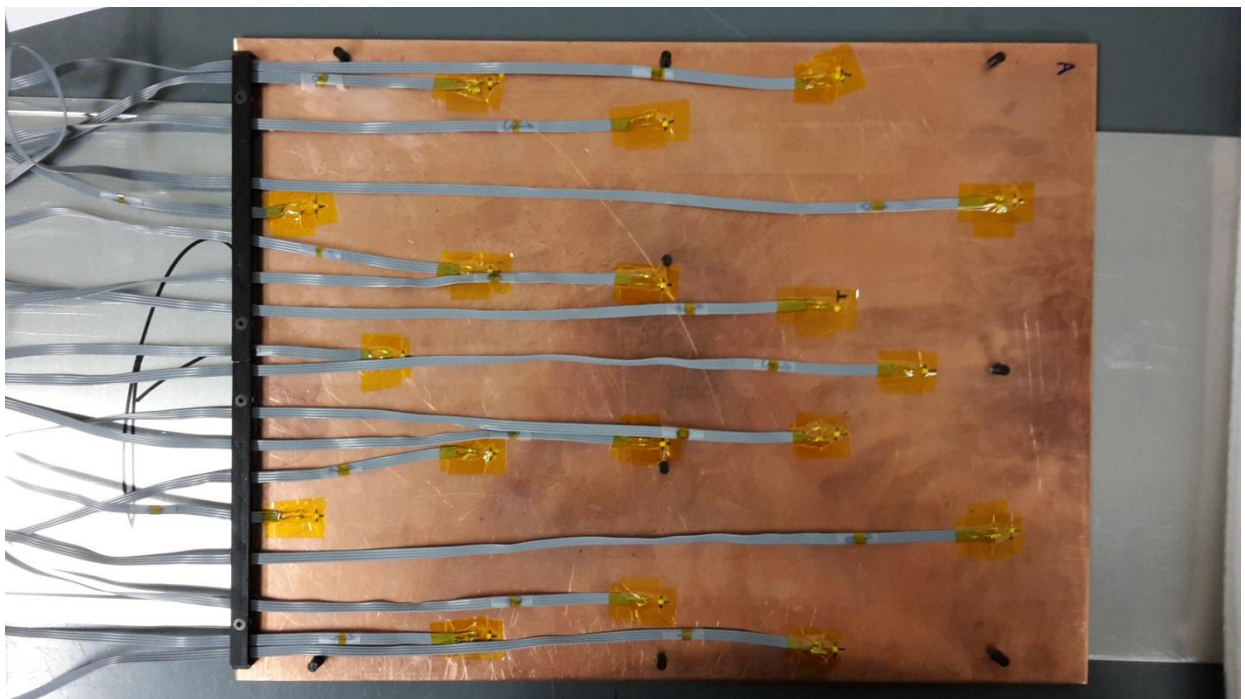


Dans la simulation, la plaque est parfaitement isolée, ce qui explique que les sondes convergent de manière idéale, alors que pendant l'expérience l'isolation est contrainte par la masse des câbles qui doivent sortir.

VI. Photographies



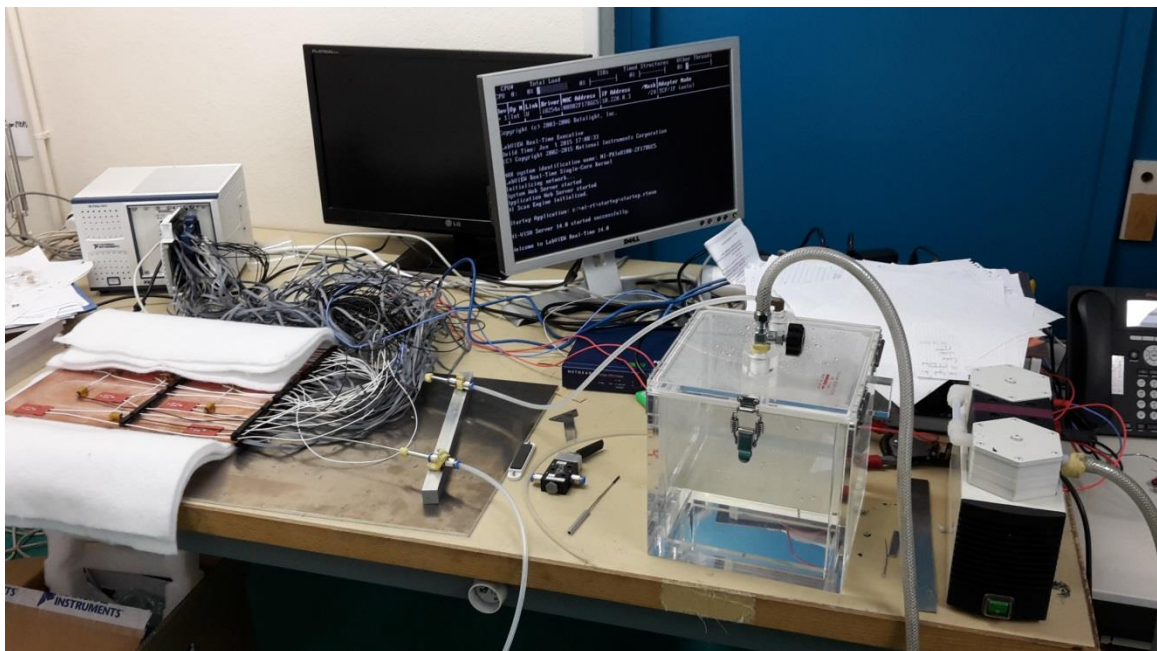
Photographie d'une plaque côté chauffe



Photographie d'une plaque côté sondes



Photographie embranchement tuyaux/arrivée d'eau



Photographie montage, en bas à droite cuve et pompe

Conclusion

Le test sur l'installation de CO₂ étant nécessaire pour conclure avec certitude sur les capacités de refroidissement du module, nous ne pouvons que tirer des conclusions partielles après ces 11 semaines de stages et les différentes manipulations réalisées.

Sur le comparatif entre la colle chargée en bille d'Argent et celle en bille d'Alumine, les données récoltées nous permettent de dire qu'il ne faut pas négliger l'impact de la conductivité de la colle, mais un test à température ambiante peut ne pas être représentatif des conditions de refroidissement à -30 °C.

Sur la simulation mathématique du comportement de la plaque de cuivre à des cycles thermiques, le résultat obtenu est satisfaisant, il semble que le modèle soit en accord avec la réalité, à l'avenir, les simulations (plus rapides) pourront se baser sur ce modèle pour créer des cas de figure variables (dimension de la plaque, des tuyaux etc...).

The test on the CO₂ installation being necessary to conclude on the cooling capacities, we can only deduce partial conclusions after these 11 weeks of internships and different manipulations performed.

On the comparison between the glue loaded with Silver and Alumina, the data collected allow us to say that we should not ignore the impact of the conductivity of the glue, but a test at room temperature may not be representative of the conditions of cooling at -30 °C.

On the mathematical simulation of the behavior of the copper plate thermal cycles, the result is satisfactory, it seems that the model is consistent with the reality, in the future, simulations (faster) will be based on this template to create variables scenarios (dimension of the plate, pipes...).

Annexes

Annexe A : Documentation constructeur des tapis chauffants, le point rouge indique celui que nous utilisons

Etched Foil Silicone Heaters - RS Range													
RS Part Number	Size (mm)	Total Area (cm.sq)	Nom Resistance (Ohms)	Nom Power (W), produced at :-					Adhesive Backed ?	Thermostat Fixing ?	Approx Watts Required To Achieve Temperature Rise Above Ambient (Guidance only - assumes minimal losses)		
				12V	24V	48V	120V	240V			+ 100 degC (0.3W/cm.sq)	+ 200 degC (0.8W/cm.sq)	+ 300 degC (1.6W/cm.sq)
245-499	25 x 50	12.5	115.2	1.25	5	20			Yes	-----	3.8	10	20
245-506	50 x 50	25.0	57.6	2.50	10	40			Yes	-----	7.5	20	40
245-512	50 x 75	37.5	38.4	3.75	15	60			Yes	-----	11.3	30	60
245-528	50 x 100	50.0	28.8	5.00	20	80			Yes	-----	15	40	80
245-556	50 x 150	75.0	19.2	7.50	30	120			Yes	-----	22.5	60	120
245-534	75 x 100	75.0	19.2	7.50	30	120			Yes	-----	22.5	60	120
245-562	75 x 200	150.0	9.6	15.00	60	240			Yes	-----	45	120	240
245-540	100 x 100	100.0	14.4	10.00	40	160			Yes	-----	30	80	160
731-350	100 x 150	150.0	9.6	15.00	60	240			Yes	-----	45	120	240
245-613	100 x 150	150.0	1152.0	0.13	0.5	2	12.5	50	Yes	Yes	45	120	240
245-629	100 x 150	150.0	576.0	0.25	1	4	25	100	Yes	Yes	45	120	240
731-366	150 x 200	300.0	4.8	30.00	120	480			Yes	-----	90	240	480
245-635	150 x 200	300.0	576.0	0.25	1	4	25	100	Yes	Yes	90	240	480
245-641	150 x 200	300.0	288.0	0.50	2	8	50	200	Yes	Yes	90	240	480
731-372	200 x 300	600.0	2.4	60.00	240	960			Yes	-----	180	480	960
245-657	200 x 300	600.0	288.0	0.50	2	8	50	200	Yes	Yes	180	480	960
245-663	200 x 300	600.0	144.0	1.00	4	16	100	400	Yes	Yes	180	480	960
245-578	200 x 400	800.0	1.8	80.00	320	1280			Yes	-----	240	640	1280
245-679	200 x 400	800.0	216.0	0.70	2.7	10.7	67	267	Yes	Yes	240	640	1280
245-685	200 x 400	800.0	108.0	1.30	5.3	21.3	133	533	Yes	Yes	240	640	1280
245-584	50 dia	19.6	72.0	2.00	8	32			Yes	-----	6	16	32
245-590	75 dia	44.2	36.0	4.00	16	64			Yes	-----	13.3	35	70
245-607	100 dia	78.6	18.0	8.00	32	128			Yes	-----	24	63	126
731-388	150 x 300	450.0	6.0	24.00	96	384			Yes	-----	135	360	720
(Heater Kit) or													
Comprises :-	50 x 150	75.0	36.0	4.00	16	64			Yes	-----	22.5	60	120
	75 x 75	56.3	48.6	3.00	12	47			Yes	-----	17	45	90
	75 x 150	112.5	23.7	6.00	24	97			Yes	-----	34	90	180
	100 x 150	150.0	18.0	8.00	32	128			Yes	-----	45	120	240
	75 dia	44.2	48.6	3.00	12	47			Yes	-----	13.3	35	70
Description :-													
Construction = Etched foil element, encapsulated between layers of glass fibre re-enforced silicone rubber.													
Thickness = 1.4mm													
Connections = 2 x 500mm PTFE flying leads													
Fixing = Peel & place adhesive backing													
Operating temperature range = -60 degC / +200 degC													
(Adhesive 150 degC continuous / 180 degC intermittent)													
Colour Key, to aid selection of power requirement :-													
Green = Low/Medium power density. Suitable for operation at normal ambients and above. Thermal control recommended.													
Yellow = High power density. Suitable for cold weather / sub-zero low ambient. Thermal control is essential.													
Red = V High power density. Only suitable for very low ambient or where rapid heat up is essential. Close thermal control is essential to prevent material breakdown over time. (Please consult office)													

Source: <http://docs-europe.electrocomponents.com>

Annexe B : Programme python complet

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
#
# Copyright 2012-2015 Frédéric Magniette, Miguel Rubio-Roy
# This file is part of Pyrame.
#
# Pyrame is free software: you can redistribute it and/or modify
# it under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# Pyrame is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU Lesser General Public License for more details.
#
# You should have received a copy of the GNU Lesser General Public License
# along with Pyrame. If not, see <http://www.gnu.org/licenses/>

import sys,os,time,datetime
import bindpyrame
from math import sqrt

if len(sys.argv)<9:
    print("usage: %s power_carpets power_points| sampling_time warming_time stab_time exit_time\nconvergence_temp power_matrix"%(sys.argv[0]))
    sys.exit(1)

f_data=open("/home/data/"+datetime.datetime.now().strftime("%Y%m%d%H%M%S")+"_hgcal.csv","w",0)

power_carpets=float(sys.argv[1])
power_points=float(sys.argv[2])
sampling_t=float(sys.argv[3])
warming_t=float(sys.argv[4])
stab_t=float(sys.argv[5])
exit_t=float(sys.argv[6])
conv=float(sys.argv[7])
pm_file=sys.argv[8]

# Connectivity parameters to power supply (ps)
conf_string_ps="ha_hmp4030(bus=tcp(host=10.220.0.20,port=5025))"
ps_port=bindpyrame.get_port("PS_PORT")
# Parameters of PXI chassis

host_pxie="10.220.0.3"
port_pxie=11001
# Relays
carpet_relays=[0,1,2,3,5,6,9,10,11,12,14,15]
point_relays=[4,7,13]
# Resistances
res_carpet=115.2
res_point=30.0

#####
# disable stdout buffering
sys.stdout=os.fdopen(sys.stdout.fileno(),"w",0)
#####

# Set relays based on power matrix
with open(pm_file,"r") as f_pm:
    pm=f_pm.read()
    position_string=pm.strip("\n")
    position_list=position_string.split(",")
    nb_carpets=0
    for relay in carpet_relays:
        if position_list[relay].strip()=="0":
            nb_carpets+=1
    print("nb_carpets=%d"%(nb_carpets))
    nb_points=0
    for relay in point_relays:
        if position_list[relay]=="0":
            nb_points+=1
```

```

print("nb_points=%d"%(nb_points))
retcode,res=bindpyrame.sendcmd
(host_pxie,port_pxie,"set_relays_pxie",position_string,"0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15")
if retcode==0:
    print("error setting relays <- %s"%(res))
    sys.exit(1)

#Current/Voltage calculation
power_for_1_carpet=power_carpets/nb_carpets
current_for_1_carpet=sqrt(power_for_1_carpet/res_carpet)
if current_for_1_carpet>0.95:
    print("error current limit for carpets too close to relay limit")
    sys.exit(1)
power_for_1_point=power_points/nb_points
current_for_1_point=sqrt(power_for_1_point/res_point)

if current_for_1_point>0.95:
    print("error current limit for points too close to relay limit")
    sys.exit(1)
current_limit=
[current_for_1_point*nb_points,current_for_1_carpet*nb_carpets,current_for_1_carpet*nb_carpets]
req_carpets=(1/(1/res_carpet*nb_carpets))
req_points=(1/(1/res_point*nb_points))
voltage=[current_limit[0]*req_points,(current_limit[1]*req_carpets)/2,(current_limit
[2]*req_carpets)/2]

# Initialization of module PS
retcode,res=bindpyrame.sendcmd("localhost",ps_port,"init_ps",conf_string_ps)
if retcode==0:
    print("error initializing PS <- %s"%(res))
    sys.exit(1)
ps_id=res

# Configure modules
retcode,res=bindpyrame.sendcmd("localhost",ps_port,"config_ps",ps_id,"fast")
if retcode==0:
    print("error configuring PS <- %s"%(res))
    sys.exit(1)

# Power off PS
for chan in range(3):
    retcode,res=bindpyrame.sendcmd("localhost",ps_port,"power_off_ps",ps_id,str(chan+1))
    if retcode==0:
        print("error powering off channel %d <- %s"%(chan+1,res))
        sys.exit(1)

# Set current limit on PS
for chan in range(3):
    retcode,res=bindpyrame.sendcmd("localhost",ps_port,"set_current_limit_ps",ps_id,str
(current_limit[chan]),str(chan+1))
    if retcode==0:
        print("error setting current limit in channel %d <- %s"%(chan+1,res))
        sys.exit(1)

    # Set voltage on PS
    retcode,res=bindpyrame.sendcmd("localhost",ps_port,"set_voltage_ps",ps_id,str(voltage[chan]),str
(chan+1))
    if retcode==0:
        print("error setting voltage in channel %d <- %s"%(chan+1,res))
        sys.exit(1)

# Take start time
start_time=datetime.datetime.now()

# First phase
phase="cold_stab"

# Buffers for stabilization criteria
nb_probes=18
stab_samples=int(round(stab_t/sampling_t))
print("stab_samples=%d",stab_samples)
buf=[[999]*stab_samples]*nb_probes
diffs=[0]*nb_probes

# Main loop
print("# time(s)    temp(C)")

```

```

f_data.write("# time(s)    temp(C)\n")
while True:
    retcode,res=bindpyrame.sendcmd(host_pxie,port_pxie,"get_temp_pxie","undef")
    if retcode==0:
        print("error measuring temp of channel <- %s"%(res))
        break
    temps=map(float,res.split(";")[0].split(","))
    now=datetime.datetime.now()
    td=now-start_time
    elapsed_s=(td.microseconds + (td.seconds + td.days * 24 * 3600) * 10**6) / 10.**6
    print("%.2f\t"%(elapsed_s)),
    f_data.write("%.2f\t"%(elapsed_s))
    for temp in temps[:nb_probes]:
        print("%.2f\t"%(temp)),
        f_data.write("%.2f\t"%(temp))
    print("")
    f_data.write("\n")
    # Wait sampling time
    time.sleep(sampling_t)
    # Exit criteria:
    if phase=="hot_stab" or phase=="cold_stab":
        for i in range(nb_probes):
            tmp=buf[i][1:]
            tmp.append(temps[i])
            buf[i]=tmp
            diffs[i]=max(buf[i])-min(buf[i])
        if max(diffs)<conv:
            if phase=="hot_stab":
                break
            if phase=="cold_stab":
                print("warming up")
                # Power on PS
                for chan in range(3):
                    retcode,res=bindpyrame.sendcmd("localhost",ps_port,"power_on_ps",ps_id,str(chan+1))
                    if retcode==0:
                        print("error powering on channel %d <- %s"%(chan+1,res))
                        sys.exit(1)
                    td=datetime.datetime.now()-start_time
                    warming_start=(td.microseconds + (td.seconds + td.days * 24 * 3600) * 10**6) / 10.**6
                    phase="warming"
            if phase=="warming" and elapsed_s>warming_start+warming_t:
                print("stabilizing")
                phase="hot_stab"
            if elapsed_s>exit_t:
                print("exit without being stabilized")
                f_data.write("exit without being stabilized\n")
                break

# Power off PS
for chan in range(3):
    retcode,res=bindpyrame.sendcmd("localhost",ps_port,"power_off_ps",ps_id,str(chan+1))
    if retcode==0:
        print("error powering off channel %d <- %s"%(chan+1,res))
        sys.exit(1)

# Deinitialize modules
retcode,res=bindpyrame.sendcmd("localhost",ps_port,"deinit_ps",ps_id)
if retcode==0:
    print("error deinitializing PS <- %s"%(res))
    sys.exit(1)

```

Annexe C : Tableau de valeurs pour 6 sondes

t° consigne (°C)	t° sonde (°C)	t° PT100 (°C)					
		0	1	2	3	4	5
50	50,4	51,852	51,856	51,845	51,78	51,826	51,82
45	46,4	47,452	47,483	47,436	47,393	47,458	47,43
40	41,8	42,68	42,723	42,67	42,641	42,698	42,674
35	36,7	37,522	37,571	37,522	37,494	37,549	37,53
30	31,7	32,375	32,434	32,372	32,356	32,416	32,389
25	25,9	26,416	26,477	26,407	26,399	26,462	26,435
20	21,1	21,327	21,401	21,319	21,325	21,38	21,362
15	15,9	16,139	16,222	16,133	16,145	16,194	16,176
10	11,4	11,125	11,226	11,12	11,149	11,209	11,18
5	6,7	6,55	6,651	6,555	6,58	6,638	6,603
0	0,7	0,49	0,605	0,496	0,531	0,588	0,559
-5	-4,3	-4,834	-4,712	-4,831	-4,788	-4,742	-4,763
-10	-9,3	-10,016	-9,89	-10,013	-9,946	-9,906	-9,926
-15	-14,2	-15,099	-14,952	-15,092	-15,023	-14,973	-14,994
-20	-19,4	-20,274	-20,114	-20,264	-20,176	-20,122	-20,158
-25	-24,7	-25,703	-25,547	-25,692	-25,604	-25,553	-25,586
-26	-25,7	-26,766	-26,599	-26,765	-26,665	-26,619	-26,637
-27	-26,8	-27,844	-27,689	-27,831	-27,747	-27,706	-27,723
-28	-27,8	-28,911	-28,749	-28,9	-28,811	-28,766	-28,787
-29	-28,9	-30,048	-29,887	-30,046	-29,955	-29,904	-29,93
-30	-29,9	-30,968	-30,802	-30,956	-30,859	-30,824	-30,838
-29	-29,2	-30,123	-29,982	-30,113	-30,038	-30	-30,013
-28	-28,5	-29,27	-29,129	-29,26	-29,186	-29,146	-29,166
-27	-27,5	-28,359	-28,222	-28,347	-28,28	-28,238	-28,253
-26	-26,6	-27,364	-27,226	-27,368	-27,279	-27,253	-27,26
-25	-25,7	-26,331	-26,205	-26,32	-26,256	-26,228	-26,23
-20	-20,8	-21,253	-21,141	-21,256	-21,19	-21,163	-21,168
-15	-15,8	-16,113	-15,999	-16,109	-16,066	-16,022	-16,029
-10	-10,6	-10,792	-10,694	-10,784	-10,755	-10,722	-10,723
-5	-5,9	-5,85	-5,765	-5,851	-5,828	-5,791	-5,783
0	-0,8	-0,666	-0,583	-0,66	-0,646	-0,608	-0,614
5	5,4	5,817	5,886	5,816	5,823	5,861	5,871
10	9,3	9,765	9,828	9,76	9,761	9,801	9,802
15	14,3	14,884	14,944	14,889	14,874	14,916	14,921
20	19	19,904	19,94	19,896	19,876	19,919	19,917
25	24,1	25,082	25,125	25,077	25,05	25,095	25,101
30	29,5	30,66	30,684	30,654	30,615	30,65	30,657
35	34,1	35,441	35,466	35,439	35,389	35,429	35,432
40	39	40,445	40,464	40,435	40,393	40,436	40,424
45	43,9	45,428	45,437	45,419	45,356	45,399	45,399
50	48,6	50,324	50,325	50,319	50,244	50,289	50,285

Annexe D : Equation de correction de chaque sonde

y=t° Pt100 / x=t° réelle

SERIE A	
sonde 0	y = 1,028775x - 0,100935 R ² = 0,999961
sonde 1	y = 1,029115x - 0,125883 R ² = 0,999964
sonde 2	y = 1,027520x - 0,076977 R ² = 0,999959
sonde 3	y = 1,029081x - 0,092071 R ² = 0,999961
sonde 4	y = 1,029492x - 0,066793 R ² = 0,999957
sonde 5	y = 1,028078x - 0,019101 R ² = 0,999961
sonde 6	y = 1,029150x - 0,099353 R ² = 0,999960
sonde 7	y = 1,028181x + 0,066224 R ² = 0,999958
sonde 8	y = 1,027728x - 0,022727 R ² = 0,999960
sonde 9	y = 1,030452x - 0,000778 R ² = 0,999955
sonde 10	y = 1,029290x - 0,044244 R ² = 0,999922
sonde 11	y = 1,027594x - 0,069033 R ² = 0,999961
sonde 12	y = 1,029567x + 0,067440 R ² = 0,999960
sonde 13	y = 1,027967x + 0,030021 R ² = 0,999963
sonde 14	y = 1,028114x - 0,042477 R ² = 0,999958
sonde 15	y = 1,029305x - 0,053783 R ² = 0,999959
sonde 16	y = 1,029176x - 0,107597 R ² = 0,999959
sonde 17	y = 1,029355x - 0,029408 R ² = 0,999959

SERIE B	
sonde 0	y = 1,031572x - 0,079326 R ² = 0,999906
sonde 1	y = 1,029751x + 0,019372 R ² = 0,999916
sonde 2	y = 1,031341x - 0,077802 R ² = 0,999906
sonde 3	y = 1,029450x - 0,047427 R ² = 0,999914
sonde 4	y = 1,029600x - 0,002843 R ² = 0,999918
sonde 5	y = 1,029682x - 0,016345 R ² = 0,999911
sonde 6	y = 1,031469x - 0,022623 R ² = 0,999906
sonde 7	y = 1,029907x - 0,032309 R ² = 0,999899
sonde 8	y = 1,029880x - 0,050403 R ² = 0,999899
sonde 9	y = 1,029397x - 0,086401 R ² = 0,999915
sonde 10	y = 1,030034x - 0,063713 R ² = 0,999898
sonde 11	y = 1,029385x - 0,080039 R ² = 0,999913
sonde 12	y = 1,031803x - 0,016969 R ² = 0,999895
sonde 13	y = 1,029806x - 0,016444 R ² = 0,999908
sonde 14	y = 1,029787x - 0,011521 R ² = 0,999912
sonde 15	y = 1,031456x - 0,056207 R ² = 0,999911
sonde 16	y = 1,029938x + 0,006743 R ² = 0,999909
sonde 17	y = 1,029151x - 0,132138 R ² = 0,999914