

---

# PROJET TUTEURE MCPC

---

2023-2024



IUT DE LILLE

2023-2024

Delcour Adrien / Cache Josse / Leroy Thibaut / Calligaro Bastien

# PROJET TUTEURE MCPC

## Sommaire :

<b>Introduction.....</b>	<b>2</b>
<b>Contexte des batteries LFB.....</b>	<b>2</b>
<b>I. Transport de matière de l'électrolyte.....</b>	<b>3</b>
<b>II. Présentation de la méthode de cyclage avec la batterie en flux.....</b>	<b>6</b>
1°) Paramètres du courant.....	6
2°) Paramètres du débit .....	7
<b>III. Résultats des premières manipulations.....</b>	<b>8</b>
<b>IV. Programmation.....</b>	<b>16</b>
1°) Température.....	16
2°) Absorbance.....	19
3°) Etat de charge .....	23
4°) Options .....	24
5°) Lancement d'une manipulation.....	27
6°) Exploitation des résultats et rapport.....	30

## Introduction :

Les batteries jouent un rôle prépondérant dans notre société moderne, alimentant une vaste gamme de dispositifs, des smartphones aux véhicules électriques. La performance d'une batterie dépend étroitement de son état de charge, un paramètre fondamental qui influe sur son efficacité, sa durée de vie et sa sécurité d'utilisation. Notre projet se concentre sur l'analyse approfondie de l'influence de plusieurs grandeurs physiques cruciales.

Ce projet est une suite de celui de l'année dernière qui était inachevée en raison de soucis liés à la programmation du spectromètre fibré. Les objectifs et les rôles restent ainsi les mêmes que l'année passée.

En premier lieu, nous allons discuter d'un nouvel aspect électrochimique qui a été étudié lors du S5. Le transport de matière qui sert à faire fonctionner l'électrolyte.

Dans un second temps, nous présenterons la méthode du cyclage avec la batterie en flux.

Nous parlerons ensuite des résultats des premières manipulations lancées. Ces manipulations nous permettent de déterminer à quel débit fixé le rendement de la batterie est le meilleur. Ce débit sera ainsi celui utilisé pour les manipulations avec le spectromètre fibré.

Enfin, nous continuerons sur l'avancée de la programmation servant à piloter tous nos capteurs extérieurs à la batterie tels que le spectromètre fibré ou encore des sondes PT100 à l'aide d'une interface graphique nommé « TechVolt »

## Contexte des batteries LFB:

Le commerce des batteries RFB (Rechargeable Flow Batteries) est un secteur en croissance, en raison de l'augmentation de la demande de solutions de stockage d'énergie à grande échelle. Les batteries RFB sont utilisées dans une variété d'applications, notamment le stockage d'énergie renouvelable, la stabilisation du réseau électrique, les applications industrielles et commerciales, ainsi que les véhicules électriques et les systèmes de secours d'urgence.

Différents électrolytes peuvent être utilisés pour le fonctionnement de ces batteries, leur prix peut varier selon leur qualité (prolongement du fonctionnement, amélioration de l'efficacité), selon la technologie de la batterie utilisée mais également en fonction du marché.

Ces batteries peuvent avoir un impact environnemental, évidemment l'extraction des matériaux servant à la confection de celle-ci en sont un, tout comme la gestion des déchets après fonctionnement de la batterie. Les efforts pour réduire cet impact incluent le développement de batteries plus durables, l'amélioration des pratiques de recyclage et l'utilisation d'énergies renouvelables pour alimenter les processus de production et de charge des batteries.

# PROJET TUTEURE MCPC

## I / Transport de matière de l'électrolyte :

Notre batterie fonctionne à l'aide de deux solutions : une solution de ferricyanure de potassium et une solution de ferrocyanure de potassium toutes deux à 0,1 mol.L

### Analyse des couples Red/Ox :

**Premier couple :**  $Fe(CN)_{6}^{3-}/Fe(CN)_{6}^{4-}$

Demi-équation :  $Fe^{3+} + 6(CN)^{-} + e^{-} \leftrightarrow Fe^{2+} + 6(CN)^{-}$

Potentiel standard :  $E_1 = E^0 + 0,06 \log \left( \frac{[Fe^{3+}][CN^{-}]^6}{[Fe^{2+}][CN^{-}]^6} \right) \leftrightarrow E_1 = E^0 = 0,41V$

### Second couple : $O_2/H_2O$

Demi-équation :  $O_2 + 4H^{+} + 4e^{-} \leftrightarrow 2H_2O$

Potentiel standard :  $E_2 = E^0 + 0,015 \log([H^{+}]^4) \leftrightarrow E_2 = E^0 - 0,06pH \leftrightarrow E_2 = 1,23 - 0,06pH$

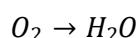
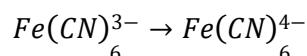
### Troisième couple : $H^{+}/H_2$

Demi-équation :  $2H^{+} + 2e^{-} \leftrightarrow H_2$

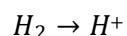
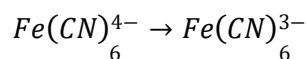
Potentiel standard : Par défaut,  $E_3 = -0,06pH$

### Attributions :

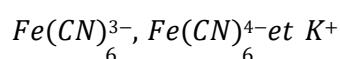
Nous avons 2 réductions qui se font à la cathode :



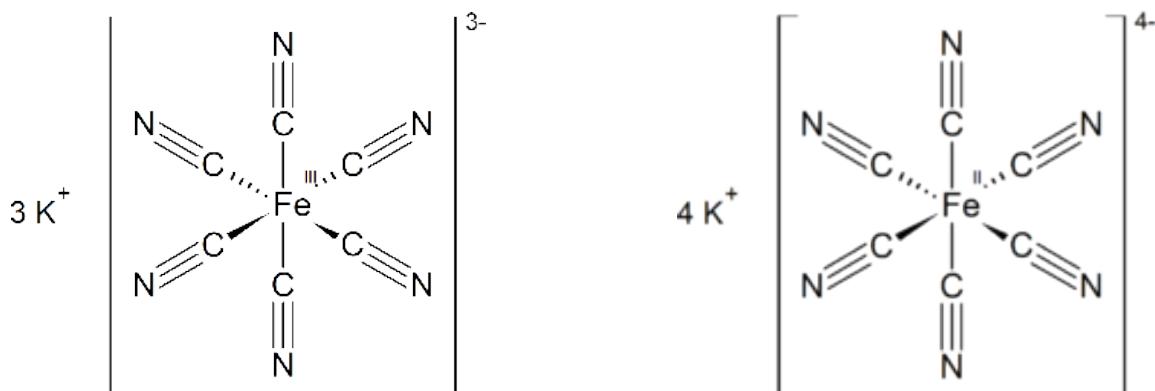
Nous avons 2 oxydations qui se font à la cathode :



Or,  $O_2$ ,  $H^{+}$  et  $H_2O$  n'interviennent pas dans les réactions électrochimiques. Les espèces qui vont intervenir sont :



## Structures chimiques des couples Red/Ox :



Lors de la dissolution, les ions  $\text{K}^+$  des complexes ferricyanure et ferrocyanure se séparent de la molécule pour assurer en partie la conduction ionique tandis que les ligands cyanure gardent leur liaison avec l'ion métallique. Ils assurent également la conduction malgré leur taille importante qui ne facilite pas le transport car ils sont fortement chargés ( $3^-$  et  $4^-$ ).

## Nombre de transport :

$$[\text{Fe}(\text{CN})_6^{3-}] = [\text{Fe}(\text{CN})_6^{4-}] = 0,1 \text{ mol. L}^{-1}$$

Il faut 4 moles de  $\text{K}^+$  pour avoir une mol de ferrocyanure de potassium et 3 mols de  $\text{K}^+$  pour avoir une mol de ferricyanure de potassium.

Ainsi,  $[\text{K}^+] = 0,7 \text{ mol. L}^{-1}$

$$\lambda_{\text{Fe}(\text{CN})_6^{3-}} = \frac{302,7 * 0,1}{302,7 * 0,1 + 441,6 * 0,1 + 73,5 * 0,7} = 0,24$$

$$\lambda_{\text{Fe}(\text{CN})_6^{4-}} = 0,35$$

$$\lambda_{\text{K}^+} = 0,41$$

Ces nombres de transport déterminent le nombre d'électron que chaque espèce transporte.

# PROJET TUTEURE MCPC

## Schéma de l'électrolyse :

Le courant est fixé à 10 électrons.

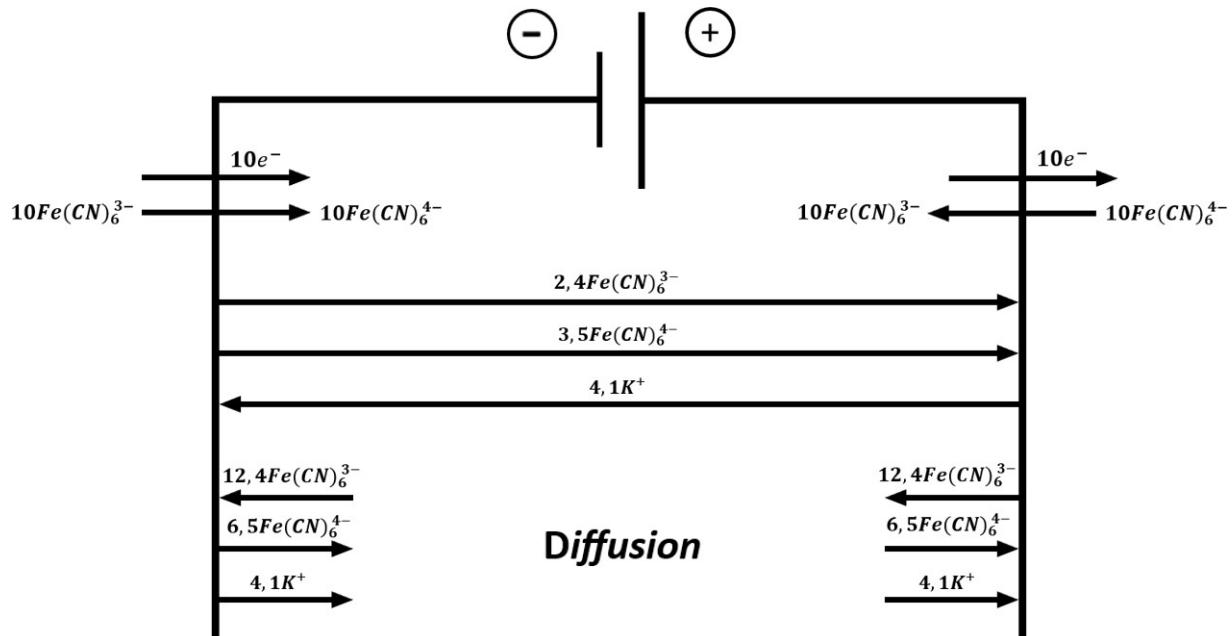


Figure 3 : Schéma du transfert de matière de l'électrolyse

En pratique néanmoins, il n'y a que les ions  $K^+$  qui assure le transport de courant. En cause la membrane de notre batterie qui est une membrane cationique : seuls les cations peuvent circuler d'une électrolyte à une autre.

## II / Présentation de la méthode du cyclage avec la batterie en flux

Pour cette partie du projet, nous allons évaluer l'impact de différents paramètres sur le rendement de la batterie et nous pourrons également décrire certains phénomènes au cours du temps.

Les paramètres pouvant être étudiés lors du cyclage de la batterie au cours du temps sont :

- La concentration
- Le volume des électrolytes
- Le débit
- Le courant
- Le pH
- La conductivité
- ...

On sait que la concentration et le volume des électrolytes feront varier un autre paramètre qui est la charge Q en Coulomb ou en mA.h.

La concentration et le volume des électrolytes resteront identiques à l'année précédente c'est-à-dire :

- 100 ml de solution de chaque électrolyte (ferro et ferri de potassium dissous)
- 0,1 mol/L

Les paramètres qui seront donc modifiés sont le courant et le débit. Pour ce qui est du pH et de la conductivité, ils seront évalués en fonction de l'avancement du projet.

### 1°) Paramètre du courant

Afin de couvrir une large gamme de courant on introduit une autre grandeur qui est « C » qui est défini comme la puissance qui fournirait la capacité d'une batterie complète sur une heure de temps. Cela signifie qu'à courant constant, la batterie peut-être complètement chargée ou déchargée en une heure.

On sait que la charge Q lorsque l'on charge à 1C est de 268,02 mA.h (cf. Rapport 2022-2023 / I. 4°). Nous allons varier les acquisitions de cycle charges/décharges de 0,2C jusque 2C lors d'une période de 17 jour théorique, c'est-à-dire 408 heures de cyclage de la batterie.

Avec les temps associés, on obtient :

C	Charge (mA.h)	Temps d'un cycle (h)	14 cycles charge/décharge (h)
0,2	53,60	5,00	180
0,4	107,21	2,50	90
0,6	160,81	1,67	60
1,0	268,02	1,00	36
1,5	402,03	0,67	24
2,0	536,04	0,50	18

**Figure 4 :** Tableau des paramètres de charge

# PROJET TUTEURE MCPC

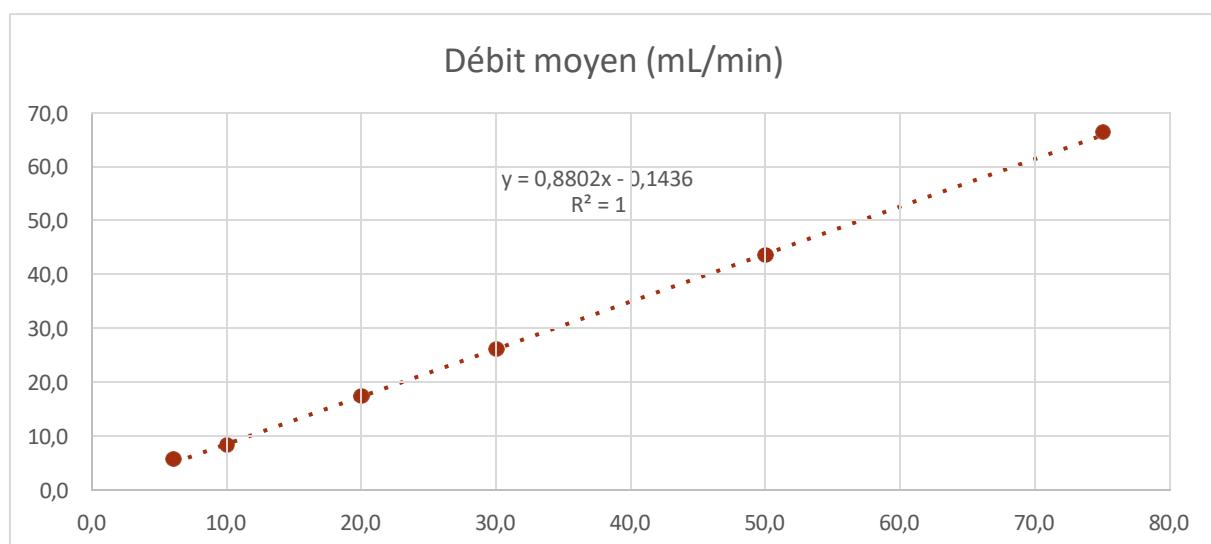
## 2°) Paramètre du débit

Ensuite, l'autre paramètre étudié sera le débit, nous gardons les paramètres précédents pour le courant qui sont des paramètres « programmables » et allons reproduire la même manipulation à différents débits.

En effet, l'inverse aurait été plus contraignant (courant constant) car pour changer la vitesse de rotation de la pompe péristaltique, il faut le faire manuellement avec la commande numérique ([voir I. 2°- Présentation du matériel : pompe rotative multicanaux] du rapport BUT2).

Les manipulations se dérouleront donc à débit constant.

La vitesse de rotation de la pompe n'ayant pas d'unité, la conversion en débit est effectuée en amont (manipulation et données extraites par Monsieur Beaucamp).



Grâce à l'équation de la régression linéaire, nous pouvons en déduire le débit pour une vitesse de rotation choisie. Afin de couvrir une large gamme de débit et d'évaluer rapidement et efficacement l'impact de celui-ci, nous allons travailler avec des débits d'environ 10, 40 et 73 mL/min qui correspondent respectivement à des vitesses de rotation (en tours/min) de 12, 46 et 84.

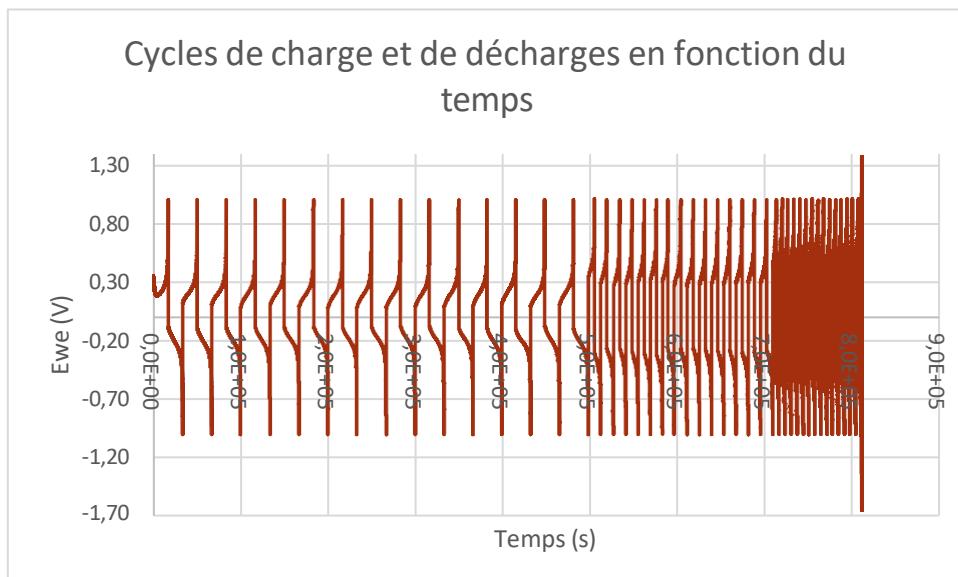
## III / Résultats des premières manipulations

Comme rappeler dans plus tôt, nous cherchons à étudier la façon dont réagit la batterie lorsque que nous sommes menées à modifier certains paramètres. Aussi lors des premières manipulations, nous avons modifier l'état de charge ainsi que le débit. Ces modifications nous mènerons à définir les paramètres idéals de fonction de la batterie afin d'obtenir le rendement le plus élevé.

Dans un premier temps nous avons lancé une manipulation à un débit fixé de 10 mL/min et nous avons effectué des cycles à des états de charge différents. Pour ce qui est des états de charge et les courants associés, nous avons balayer la plage suivante :

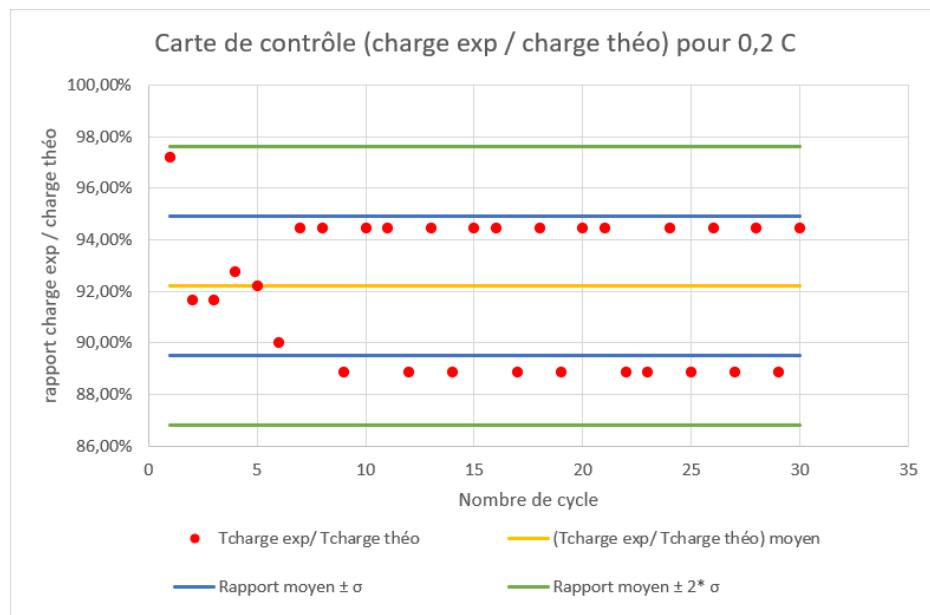
C	Charge (mA.h)	Temps d'un cycle (h)
0,2	53,60	5,00
0,4	107,21	2,50
0,6	160,81	1,67
1,0	268,02	1,00
1,5	402,03	0,67
2,0	536,04	0,50

Nous avons effectué des cycles de charge et décharge. Nous obtenons à la suite de cette manipulation le graphique suivant :



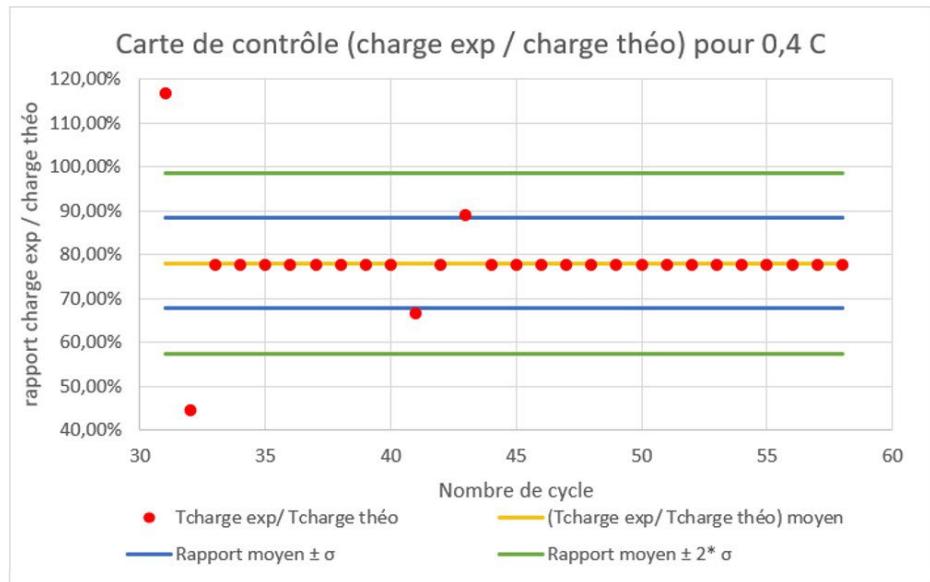
Nous pouvons bien identifier la succession de la charge et de la décharge de la batterie. Aussi nous pouvons identifier clairement les différents états de charges, quoique l'arrêt soudain de cette succession et de la manipulation nous éveille sur un possible problème durant cette dernière. Afin de vérifier le bon déroulement de la manipulation, nous avons tiré les cartes de contrôle de chacun des états de charges. Pour cela nous pouvons dans un premier temps comparer le temps de charge + décharge théorique à celui observé expérimentalement.

# PROJET TUTEURE MCPC



A un état de charge de 0,2 C, le temps de charge/décharge théorique est de 5h. Ici, on observe le rapport Tcharge expérimentale /Tcharge Théorique moyen à 92%, le rapport idéal étant normalement de 100%, nous avons une efficacité correcte. Aussi la dispersion des points n'est pas aberrante ce qui nous rassure quant à l'utilisation de batterie dans ces paramètres.

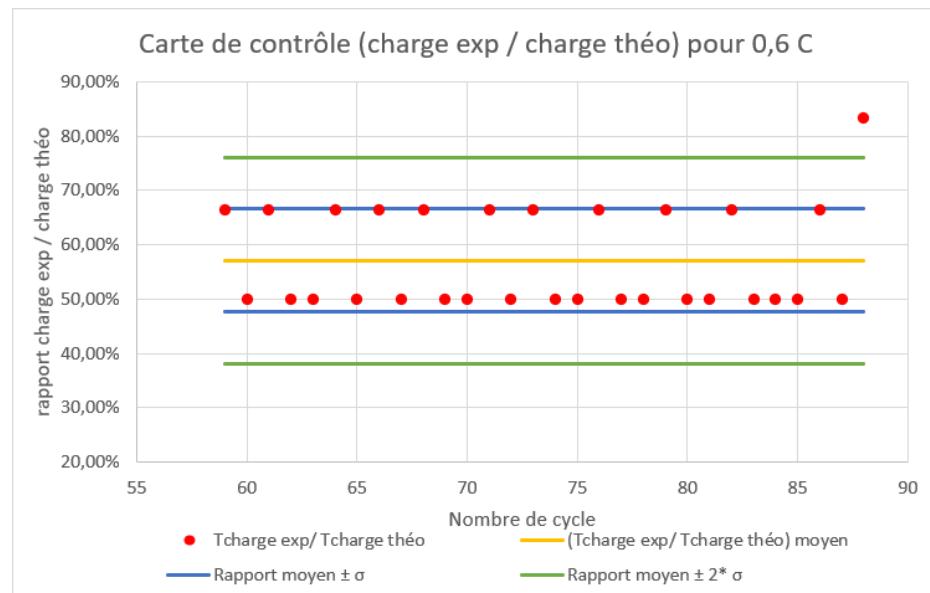
Les cycles suivants ont été réalisés à un état de charge de 0,4 C la carte de contrôle est la suivante :



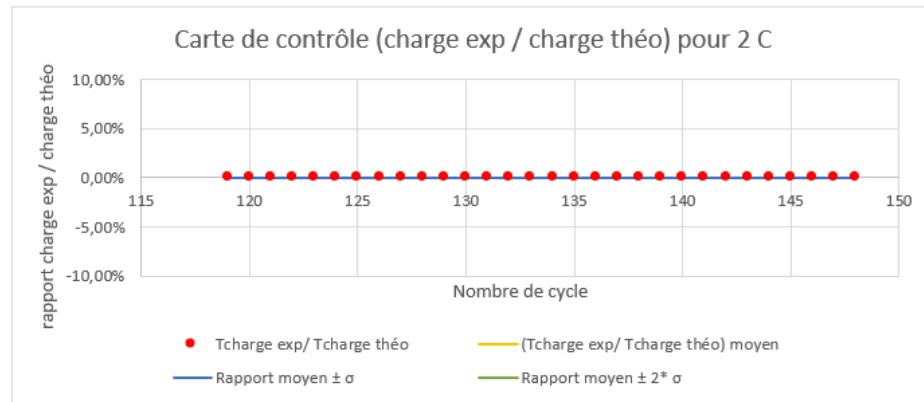
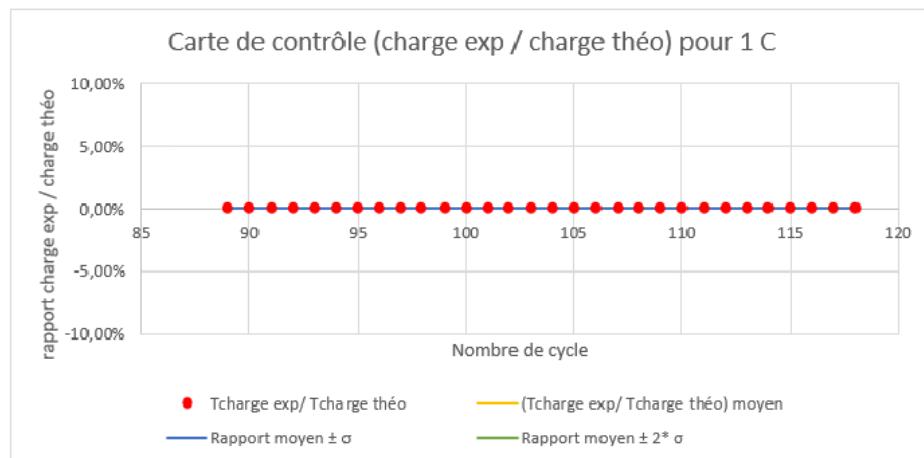
Sur cette carte de contrôle, nous observons un rapport Tcharge expérimentale /Tcharge Théorique moyen à environ 78%. Cela représente une efficacité moindre que lors des cycles à un état de charge de 0,2 C. Cependant la dispersion est plus faible ici. Seuls les deux premiers points ont un comportement différent des autres, on peut l'expliquer par la transition entre les cycles à 0,2 C et les cycles 0,4 C.

Pour ce qui est de l'efficacité, l'allure que à l'air de prendre celle-ci est que plus l'état de charge augmente est plus l'efficacité diminue.

# PROJET TUTEURE MCPC



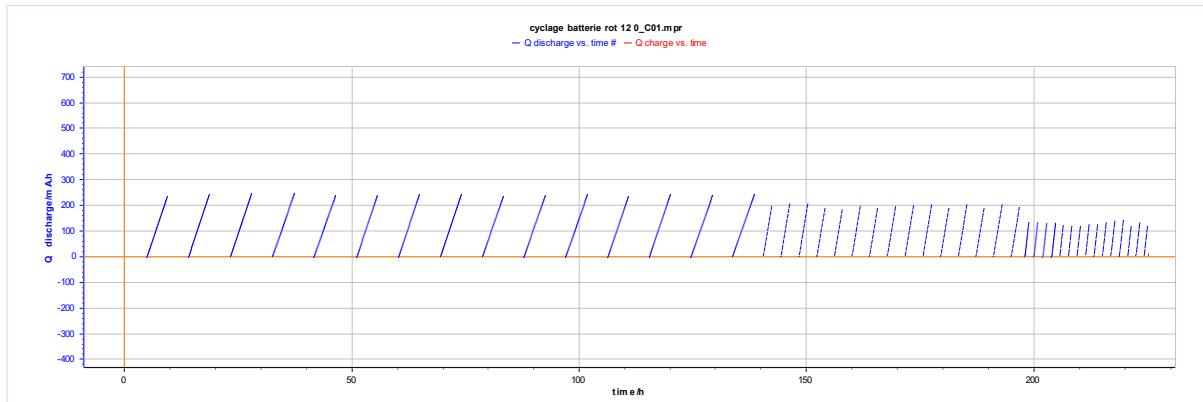
Cette carte de contrôle vient certifier la supposition que nous avons émis plus tôt, en effet plus l'état de charge augmentent plus l'efficacité diminue.



Les carte de contrôle des cycles aux états de charge de 1 C et de 2 C permettent de conclure que le débit choisi lors de cette première manipulation est trop faible pour permettre la batterie de fonctionner efficacement, en effet, la convection n'est pas assez importante pour amener assez de

# PROJET TUTEURE MCPC

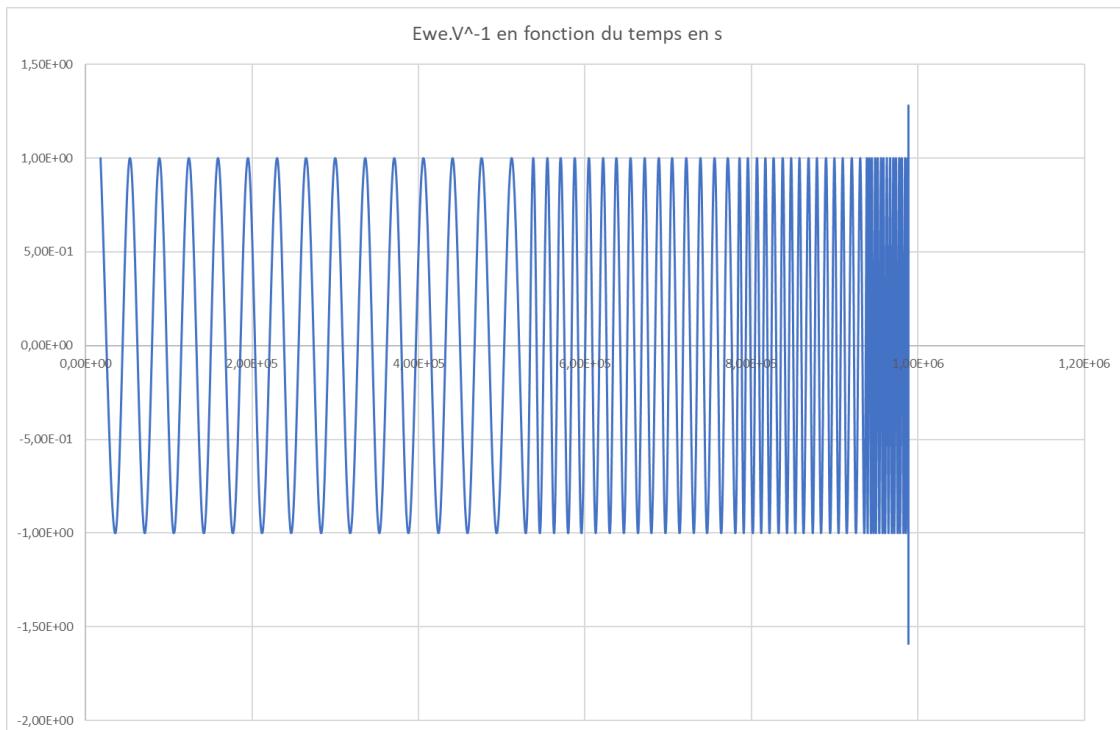
matière (et donc de réactifs) aux sièges des réactions électrochimiques. De plus lors de ces derniers cycles, la batterie atteint son *Cut off* avant même d'avoir pu commencer à se charger ou à se décharger. En effet la batterie atteint la tension limite inférieure prescrite à laquelle la décharge de la batterie est considérée comme complète ou inversement pour la charge.



**Figure 5 :** Graphique des cycles de charges/décharge (mA.h) en fonction du temps

Ce graphique nous permet de vérifier l'hypothèse que nous avons émis ci-dessus, Une fois les cycles à un état de charge de 0,6 terminés, on ne peut plus observé de cycles dû au *Cut off* de la batterie. On remarque bien que plus l'état de charge augmente et plus l'efficacité diminue, jusqu'à être égale à 0.

Suite à cette première manipulation, nous avons effectué deux nouvelles manipulations mais à des débits différents. La première est effectuée à un débit de 40 ml/min :



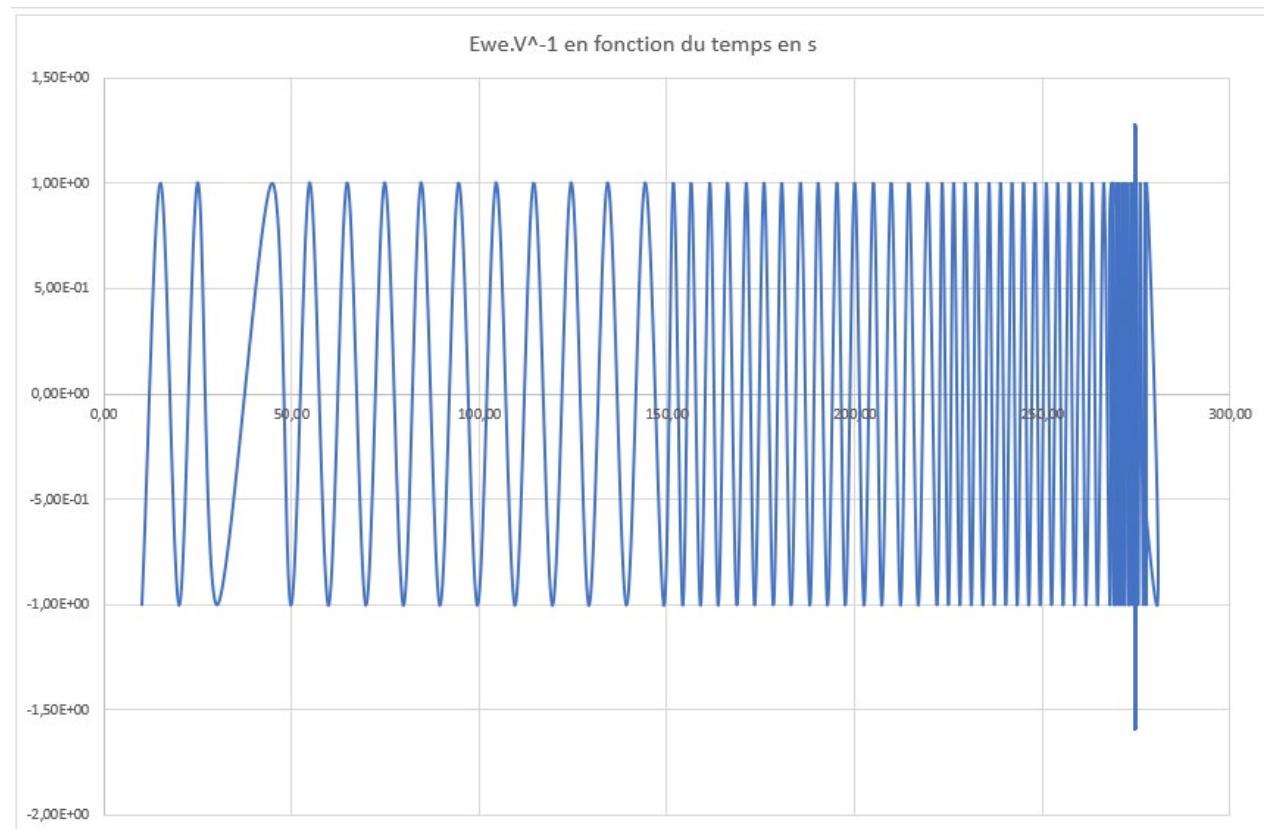
## PROJET TUTEURE MCPC

Contrairement au premier débit, nous avons davantage de données. En effet, à 10 mL/min, l'acquisition a duré environ 220 heures, maintenant, on est quasiment à 280 heures. Le débit augmente bien l'apport en réactifs ce qui permet à la batterie en flux de fonctionner à des "C-rate" plus élevés.

La dernière manipulation est effectué un débit fixé de 73 mL/min nous avons effectué des cycles à des états de charge différents. Pour ce qui est des états de charge et les courants associés, nous avons balayer la même plage que précédemment :

C	Charge (mA.h)	Temps d'un cycle (h)
0,2	53,60	5,00
0,4	107,21	2,50
0,6	160,81	1,67
1,0	268,02	1,00
1,5	402,03	0,67
2,0	536,04	0,50

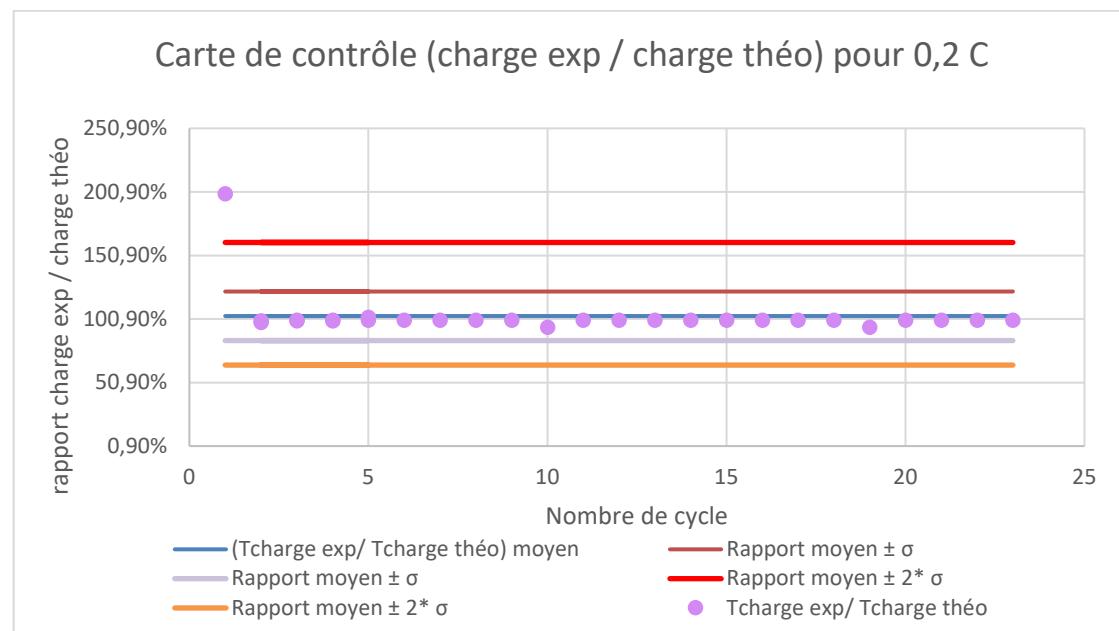
Nous avons effectué des cycles de charge et décharge. Nous obtenons à la suite de cette manipulation le graphique suivant :



Nous pouvons remarquer qu'il manque les données d'un cycle de charge/décharge, nous ne l'expliquons pas mais cela à peu d'influence sur le reste des données. Aussi la durée de l'acquisition est

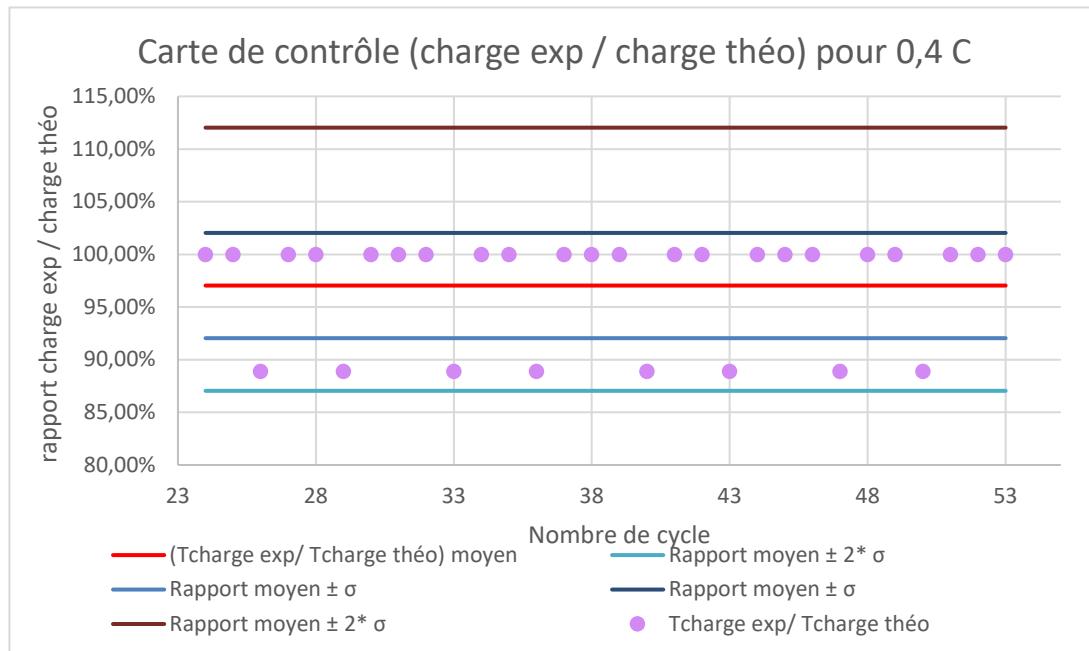
## PROJET TUTEURE MCPC

un peu près la même que celle à une vitesse de rotation de 40 t/min. De la même manière que précédemment, afin de vérifier le bon déroulement de la manipulation, nous avons tiré les cartes de contrôle de chacun des états de charges. Pour cela nous pouvons dans un premier temps comparer le temps de charge + décharge théorique à celui observé expérimentalement.



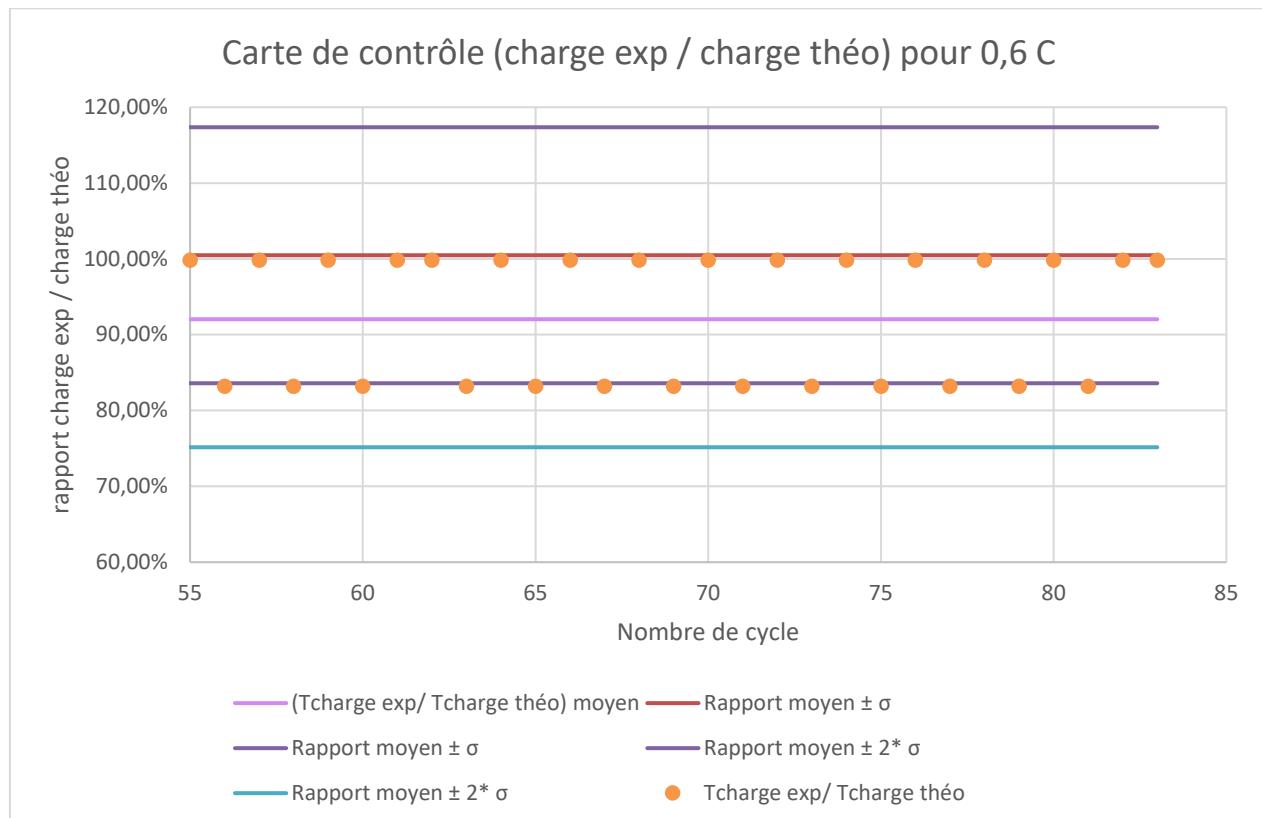
A un état de charge de 0,2 C, le temps de charge/décharge théorique est de 5h. Ici, on observe le rapport Tcharge expérimentale /Tcharge Théorique moyen à environ 100% , le rapport idéal est atteint, nous obtenons une efficacité parfaite. Aussi la dispersion des points est très bonne ce qui nous rassure quant à l'utilisation de batterie dans ces paramètres et nous concilie à utiliser la batterie dans ces conditions.

Les cycles suivants ont été réalisés à un état de charge de 0,4 C la carte de contrôle est la suivante :



## PROJET TUTEURE MCPC

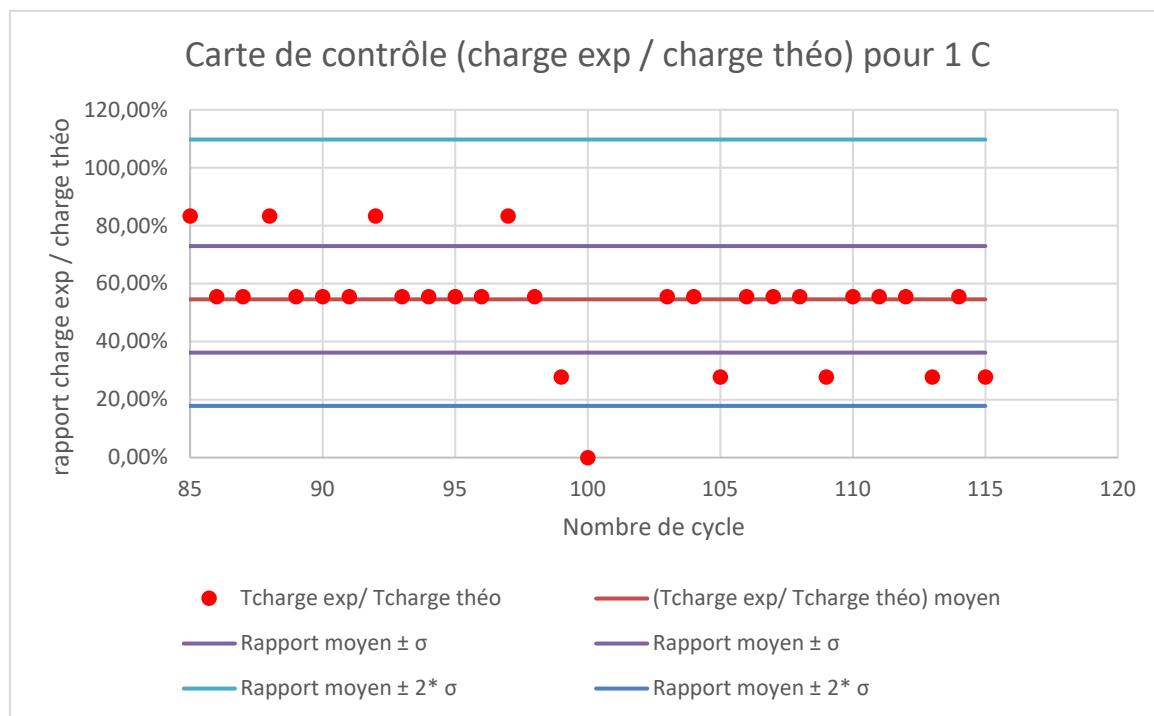
Sur cette carte de contrôle, nous observons un rapport  $T_{charge \text{ expérimentale}} / T_{charge \text{ Théorique}}$  moyen à environ 97%. Cela représente une efficacité moindre que lors des cycles à un état de charge de 0,2 C. Mais cela reste très satisfaisant. Cependant la dispersion est plus grande ici. Aussi de la même manière que avec les autres débits, l'allure que à l'air de prendre l'efficacité est que plus l'état de charge augmente est plus l'efficacité diminue. Nous pouvons vérifier cela à l'aide de la carte de contrôle suivante :



Cette carte de contrôle vient certifier la supposition que nous avons émis plus tôt, en effet plus l'état de charge augmentent plus l'efficacité diminue. Aussi nous observons une dispersion plus grande que précédemment, aussi cette dispersion ne semble pas aléatoire. Nous ne pouvons pas l'expliquer mais les cycles semblent alterner entre le rapport moyen + l'écart type et le rapport moyen - l'écart type.

## PROJET TUTEURE MCPC

La carte de contrôle pour 1C montre un rapport  $T\text{charge expérimentale}/T\text{charge Théorique}$  moyen à environ 53%. Cela représente une efficacité moindre que lors des cycles à un « taux de charge moins important, la batterie devient moins intéressante pratiquement .



Les derniers états de charge ne nous apportent aucune information en plus. En effet, le cut off est atteint trop vite ce qui empêche l'acquisition.

Pour résumer, nous pouvons tracer le tableau suivant :

Vitesse de charge C	Débit (mL/min)		
	10	40	73
0,2	92,22	97,96	103,22
0,4	77,98	93,25	97,04
0,6	57,11	86,69	92,04
1	0	49,07	54,60
1,5	0	0	0
2	0	0	0

Grâce au tableau, on remarque très clairement que entre le débit à 10 et à 40, pour des vitesses de charge comprises entre 0.2 et 0.6 C, l'efficacité est toujours supérieure pour le débit à 40. On relève également que l'écart entre les deux débits augmente au fur et à mesure. Pour une vitesse à un 1C, l'apport en réactifs n'est pas assez rapide et important à un débit de 10. En revanche, à 40, on a tout de même 49% d'efficacité encore un point important à soulever en faveur de ce débit. Par la suite, les vitesses de charge à 1.5 et 2 C sont beaucoup trop importantes et rapides pour apercevoir les résultats des réactions électrochimiques. En effet, on aurait besoin davantage de quantité de matière (molaire) aux pôles ou un apport plus rapide.

Aussi, en comparaison avec le fonctionnement de la batterie à un débit de 73 ml/min, on remarque que qu'importe la vitesse de charge, la batterie est plus efficace quand le débit est plus important.

## IV / Programmation

Aujourd’hui nous avons une interface graphique regroupant tous les capteurs qui relève des données lorsque l’on lance une manipulation à partir d’une fenêtre de dialogue qui permet de rentrer différents paramètres, qui synchronise les capteurs et écrit les données dans un fichier sous format « .txt » ou « .csv ».

Nous avons maintenant deux capteurs de température étalonné et un spectromètre qui sont opérationnels avec des options tels que :

- Pouvoir modifier l’intervalle de l’axe des abscisses (Température (°C) ou Absorbance)
- Pouvoir modifier la vue de l’axe des ordonnées (Temps (seconde))
- Pouvoir créer des seuils modifiables qui envoie une notification d’alertes lorsqu’elles sont dépassées
- Pouvoir modifier les paramètres ci-dessus en mode « Grouper » ou « Séparer » (Uniquement pour la partie température)

Nous allons voir dans un premier temps tout ce qui est en rapport avec la température, après cela nous verrons la partie absorbance, suite à ceci nous verrons avec les différentes options citer en détails, nous poursuivrons par voir comment lancer une manipulation et nous finirons par la section exploitation des résultats et rapports.

### 1°) Température

En reprenant le travail de l’année dernière nous avons premièrement intégrer le graphique dans la nouvelle interface qui est permis avec un GraphicsScene intégrer dans un GraphicsView.

La classe QGraphicsScene fournit une surface permettant de gérer un grand nombre d’éléments graphiques 2D et la classe QGraphicsView fournit un widget pour afficher le contenu d’un QGraphicsScene (Figure IV.1).

```
self.scene_1 = QGraphicsScene()
self.scene_1.addWidget(self.airGraphCanvas.figure.canvas)

self.scene_2 = QGraphicsScene()
self.scene_2.addWidget(self.tempeGraphCanvas.figure.canvas)

# Graphique Air
self.graphicsView_4 = self.ui.graphicsView_4
self.graphicsView_4.setScene(self.scene_1)
self.graphicsView_4.viewport().installEventFilter(self)
self.graphicsView_4.setContentsMargins(0, 0, 0, 0)
self.graphicsView_4.setHorizontalScrollBarPolicy(Qt.ScrollBarAlwaysOff)
self.graphicsView_4.setVerticalScrollBarPolicy(Qt.ScrollBarAlwaysOff)

# Graphique Electrolyte
self.tempeGraph = self.ui.tempeGraph
self.tempeGraph.setScene(self.scene_2)
self.tempeGraph.viewport().installEventFilter(self)
self.tempeGraph.setContentsMargins(0, 0, 0, 0)
self.tempeGraph.setHorizontalScrollBarPolicy(Qt.ScrollBarAlwaysOff)
self.tempeGraph.setVerticalScrollBarPolicy(Qt.ScrollBarAlwaysOff)
```

Figure IV.1 : Programme permettant l’intégration des graphiques Matplotlib dans l’interface

## PROJET TUTEURE MCPC

Nous avons rajouté un deuxième capteur de température car cela permet d'avoir un pont de comparaison entre la température de l'air et des électrolytes. Pour intégrer tous ces changements nous avons placer les différents éléments à l'aide du logiciel QtDesigner, voici le résultat à l'heure actuelle :

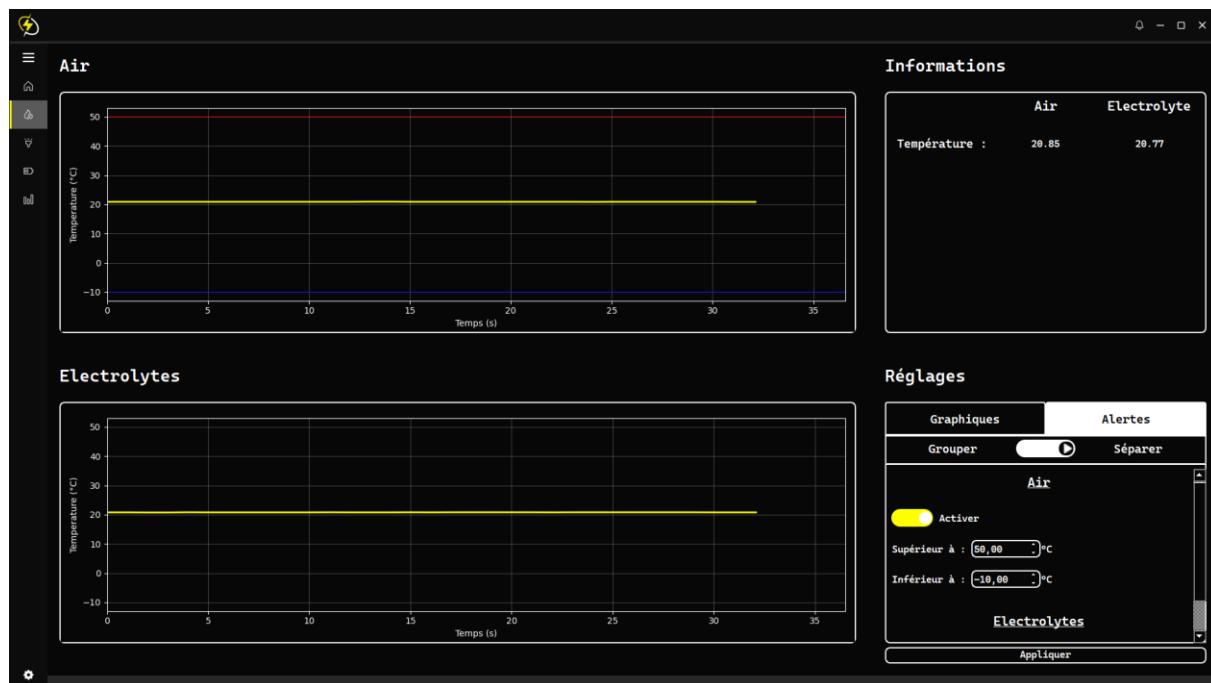


Figure IV.2 : Vue de l'interface graphique TechVolt pour la

Comme nous pouvons le constater il y a maintenant deux graphiques Matplotlib, un pour la température de l'air et un autre pour la température des électrolytes qui trace la température en temps réel. Nous avons aussi directement la valeur des températures affichés dans l'onglet « Informations » et nous réfléchissons actuellement aux différentes données que nous pourrions rajouter tels que la moyenne, le maximum, le minimum et encore d'autres qui pourraient être utiles à l'utilisateur. Nous avons aussi un onglet « Réglages » avec un bouton « Appliquer » que nous détaillerons dans la partie dédiée à cela.

Pour envoyer les données des deux capteurs simultanément nous avons dû revoir le câblage du système et le programme Arduino. En effet le câblage ne permettait que d'envoyer des données pour un capteur et nous avons dû revoir le programme Arduino car ce dernier stocker les valeurs sur la carte Arduino et donc au bout d'un certain temps (qui dépend de l'intervalle de la prise de mesure) la mémoire saturée et plus aucune valeur n'était affichée.

Pour la partie câblage nous avons donc d'abord dédoubler l'Adafruit MAX31865 avec sonde Pt100 (Voir rapport BUT2, page 25, figure 30) afin d'avoir un deuxième capteur.

Ensuite nous avons effectuée des recherches pour comprendre comment la communication d'informations entre la carte Arduino vers l'ordinateur s'effectue. Nous avons trouvé que nous devions utiliser le protocole dit SPI (Serial Peripheral Interface) qui est un protocole de communication série synchrone utilisé pour la communication entre plusieurs périphériques sur le même bus.

# PROJET TUTEURE MCPC

Voici le programme Arduino résultant de ces recherches :

```
#include <Adafruit_MAX31865.h>
Adafruit_MAX31865 mon_capteur1 = Adafruit_MAX31865(10);
Adafruit_MAX31865 mon_capteur2 = Adafruit_MAX31865(9);

void setup() {
    Serial.begin(9600);
    mon_capteur1.begin(MAX31865_2WIRE);
    mon_capteur2.begin(MAX31865_2WIRE);
}

void loop() {
    float R1 = mon_capteur1.lecture_resistance();
    float T1=(R1-100.31)/0.3795;

    float R2 = mon_capteur2.lecture_resistance();
    float T2=(R2-100.80)/0.3618;

    Serial.print ("Température Capteur 1: ");
    Serial.print(T1);
    Serial.println("°C");
    Serial.print ("Température Capteur 2: ");
    Serial.print(T2);
    Serial.println("°C");

    delay (1000);
}
```

**Figure IV.3 :** Programme Arduino permettant le fonctionnement des capteurs de

On utilise d'abord la bibliothèque Adafruit\_MAX31865 pour interagir avec deux capteurs MAX31865, qui mesurent la température en lisant la résistance de thermocouples.

Le protocole SPI utilise plusieurs lignes de communication, dont certaines sont partagées entre les périphériques connectés. Ces lignes comprennent généralement :

- MOSI (Master Out Slave In) : C'est la ligne par laquelle le maître envoie des données aux esclaves.
- MISO (Master In Slave Out) : C'est la ligne par laquelle les esclaves renvoient des données au maître.
- SCLK (Serial Clock) : C'est la ligne qui génère l'horloge pour synchroniser la communication.
- CS (Chip Select) : Cette ligne est utilisée pour sélectionner le périphérique avec lequel le maître souhaite communiquer.

Dans votre programme, chaque capteur est connecté à un port CS différent.

Nous avons :

```
Adafruit_MAX31865 mon_capteur1 = Adafruit_MAX31865(10);
Adafruit_MAX31865 mon_capteur2 = Adafruit_MAX31865(9);
```

La ligne CS est utilisée pour activer ou désactiver un périphérique spécifique sur le bus SPI. Lorsqu'un capteur est sélectionné (CS est actif), il est prêt à recevoir des commandes et à transmettre des données. Lorsqu'il n'est pas sélectionné, il est en mode veille et n'interfère pas avec les autres

## PROJET TUTEURE MCPC

périphériques sur le bus. En résumé, la raison pour laquelle chaque capteur a une ligne CS distincte est de permettre au microcontrôleur de choisir quel capteur il souhaite interroger à un moment donné. Chaque capteur est indépendant dans la communication, mais ils partagent les autres lignes SPI (MOSI, MISO, SCLK) pour économiser des broches et simplifier le câblage.

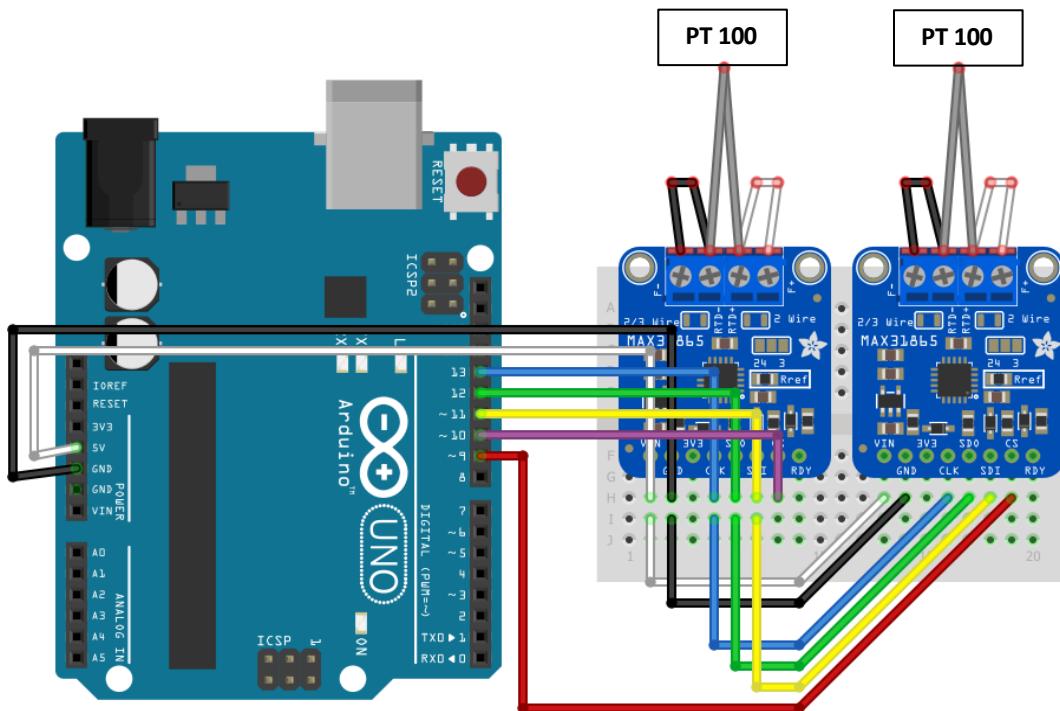


Figure IV.4 : Câblage des différents composants pour la partie

Nous avons maintenant deux capteurs de température opérationnels qui envoient les données vers l'ordinateur afin de tracer la température en temps réel. Pour ce qui est de l'étalonnage on réutilise la même méthode que l'année dernière.

### 2°) Absorbance

Pour la partie absorbance nous nous étions arrêtés au simple fonctionnement « brut » du spectromètre donc les objectifs étaient les mêmes que pour la température c'est-à-dire avoir un graphique qui trace en temps réel l'absorbance dans l'interface graphique TechVolt et de récupérer les données.

Notre spectromètre qui est un Flame-S-UV-VIS-ES de Ocean Insight (voir rapport 1<sup>ère</sup> année, page 26), lorsque l'on connecte ce dernier à l'ordinateur il est détecté directement.

# PROJET TUTEURE MCPC

Nous avons fait un programme permettant de tout d'abord détecter le spectromètre qui est connecté à l'ordinateur avec la fonction suivante :

```
def initialize_spectrometer(self):
    devices = sb.list_devices()
    if not devices:
        raise ValueError("Aucun spectromètre trouvé.")

    spectrometer = sb.Spectrometer(devices[0])
    spectrometer.integration_time_micros(500000)
    return spectrometer
```

**Figure IV.5 : Programme permettant de détecter le spectromètre**

La fonction cherche les spectromètres qui peuvent être connecter à l'ordinateur, s'il y en a un alors on renvoie ce dernier à d'autres fonctions et sinon on indique qu'aucun spectromètre n'est détecté.

Avant de faire une manipulation il faut mesurer ce que l'on appelle un « Blanc » et un « Noir ». Voici ce que signifient généralement ces termes :

## Blanc :

Définition : Le blanc est une mesure de référence effectuée sans échantillon. Cela implique de mesurer le signal de l'instrument lorsque la source de lumière ou de rayonnement passe à travers un espace vide ou un matériau transparent.

Utilité : Le blanc permet de corriger les effets de fond, tels que la lumière parasite ou le bruit électronique, présents dans le système de mesure. En soustrayant le blanc du spectre total (échantillon + instrument), on peut isoler le signal provenant uniquement de l'échantillon.

## Noir :

Définition : Le noir est une mesure de référence effectuée lorsque la source de lumière ou de rayonnement est complètement bloquée, de sorte qu'aucun signal n'atteint le détecteur.

Utilité : Le noir permet de mesurer le bruit de fond maximal de l'instrument. En soustrayant cette mesure du spectre total, on peut identifier et corriger les variations indésirables du signal qui ne proviennent pas de l'échantillon.

Donc en bref le blanc et le noir en spectroscopie servent à éliminer les contributions indésirables au signal, à corriger les variations de l'instrument et à garantir la précision des mesures réalisées sur un échantillon donné.

Pour faire ces deux mesures nous avons ces deux fonctions qui mesurent l'intensité lumineuse et qu'ils les sauvegarde dans un fichier sous format «.pkl» afin de les réutiliser pour le calcul d'absorbance (Figure IV.6).

# PROJET TUTEURE MCPC

```
def measure_white(self):
    spectrometer = self.initialize_spectrometer()

    self.measure_dialog = MeasureDialog()
    self.measure_dialog.exec()
    time.sleep(0.25)
    white_measurement = self.measure_intensity(spectrometer)

    measurements = {'white': white_measurement, 'black': None}
    self.save_measurement_to_pkl(measurements)

    spectrometer.close()
    return white_measurement

def measure_black(self):
    spectrometer = self.initialize_spectrometer()

    self.measure_dialog = MeasureDialog()
    self.measure_dialog.exec()
    time.sleep(0.25)
    black_measurement = self.measure_intensity(spectrometer)

    measurements = self.load_measurement_from_pkl()
    measurements['black'] = black_measurement
    self.save_measurement_to_pkl(measurements)

    spectrometer.close()
    return black_measurement
```

**Figure IV.6 :** Programme permettant la mesure du blanc et du noir

Une fois que nous avons ces deux données nous pouvons lancer la manipulation afin de tracer l'absorbance en temps réel or le spectromètre ne relève pas directement l'absorbance, il relève une intensité lumineuse donc nous devons faire un calcul avant de tracer un point de mesure.

L'absorbance ( $A$ ) en spectroscopie peut être calculée à l'aide de la formule suivante :

$$A = -\log_{10} \left( \frac{I}{I_0} \right)$$

Où :

- $A$  est l'absorbance
- $I$  est l'intensité lumineuse mesurée à travers l'échantillon
- $I_0$  est l'intensité lumineuse de référence (le blanc)

La formule est basée sur le principe que l'absorbance est liée négativement au logarithme décimal du rapport entre l'intensité lumineuse traversant l'échantillon ( $I$ ) et l'intensité lumineuse de référence ( $I_0$ ). Plus l'absorbance est élevée, plus l'échantillon absorbe de lumière.

# PROJET TUTEURE MCPC

## Mesure de l'intensité lumineuse de référence ( $I_0$ ) :

Prendre une mesure de l'intensité lumineuse en l'absence d'échantillon, généralement en utilisant un blanc (par exemple, un espace vide ou un matériau transparent). C'est notre intensité de référence ( $I_0$ ).

## Mesure de l'intensité lumineuse à travers l'échantillon ( $I$ ) :

On place notre échantillon dans le spectromètre et on mesure l'intensité lumineuse à travers lui.

## Calcule de l'absorbance (A) :

On utilise la formule  $A = -\log_{10} \left( \frac{I}{I_0} \right)$  pour calculer l'absorbance en utilisant les valeurs mesurées.

Voici le programme qui en découle :

```
def calculate_absorbance(self, I, I0, I_black=None):
    if I_black is not None:
        return -np.log10((I - I_black) / I0)
    else:
        return -np.log10(I / I0)

def read_spectrometer_start(self):
    measurements = self.load_measurement_from_pk1()
    white_measurement = measurements['white']
    black_measurement = measurements['black']

    self.spectrometer = self.initialize_spectrometer()
    self.thread = Thread(target=self.background_thread, args=(white_measurement, black_measurement))
    self.thread.start()

    while not self.isReceiving:
        time.sleep(0.1)

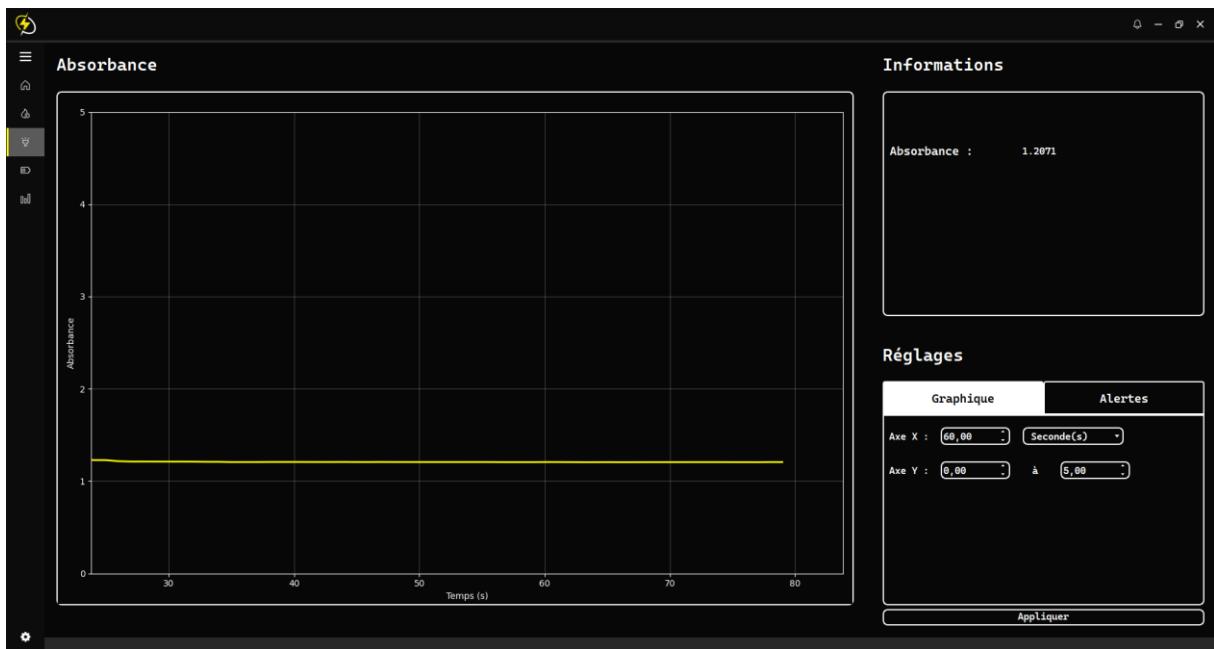
    print("Démarrage de la lecture du spectromètre.")

def background_thread(self, white_measurement, black_measurement):
    time.sleep(1.0)
    if self.spectrometer is not None:
        while self.isRun:
            intensity = self.measure_intensity(self.spectrometer)
            absorbance = self.calculate_absorbance(intensity, white_measurement, black_measurement)
            self.data3.append(absorbance)
            time.sleep(1)
            self.isReceiving = True
    else:
        print("Le spectromètre n'est pas correctement initialisé")
```

**Figure IV.7 : Programme permettant la mesure de l'absorbance**

Puis comme pour la température, il ne reste plus qu'à placer les éléments (GraphicsScene, GraphicsView, ...) à l'aide de QtDesigner et nous avons notre fenêtre pour l'absorbance au complet qui permet de tracer l'absorbance en fonction du temps :

# PROJET TUTEURE MCPC

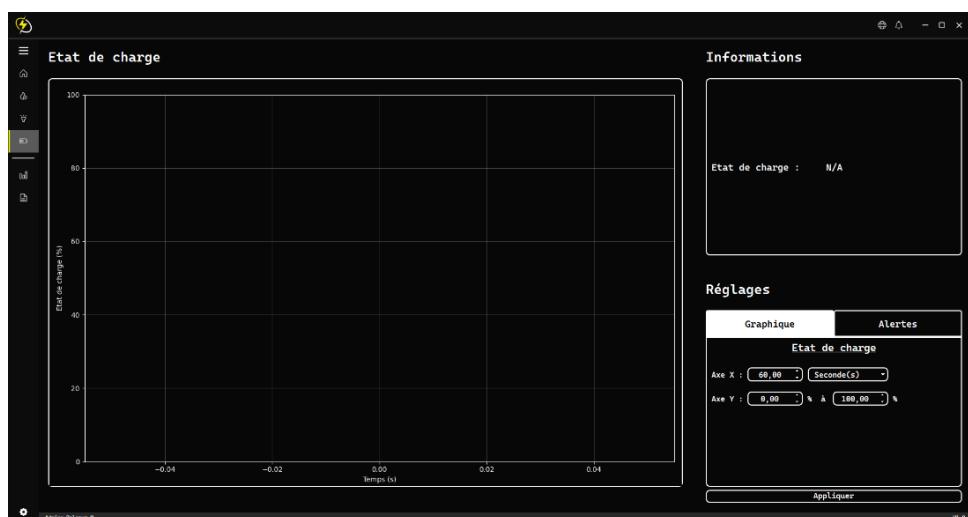


**Figure IV.8 :** Vue de l'interface graphique TechVolt pour l'absorbance

Nous avons maintenant un spectromètre opérationnel qui envoie des données vers l'ordinateur afin de tracer l'absorbance en temps réelle.

### 3°) Etat de charge

La partie Etat de charge n'a pas encore été traitée dû à des difficultés pour l'implémentation du calcul qui prends en compte la variable  $\epsilon$  mais elle sera opérationnelle d'ici Juin 2024. Voici l'interface pour l'état de charge :



**Figure IV.9 :** Vue de l'interface graphique TechVolt pour l'absorbance

# PROJET TUTEURE MCPC

## 4°) Les options

Lorsque l'on souhaite lancer une nouvelle expérience nous arrivons sur le menu Figure IV.17. Nous avons 3 choix qui sont « Température », « Absorbance » ou « Etat de charge ».

Si on sélectionne « Température » alors une nouvelle partie s'affiche dans la fenêtre (Figure IV.10) où l'on peut d'abord voir le modèle de la carte Arduino connecté avec un bouton à droite qui permet de rafraîchir ce dernier. Il y a également la possibilité de choisir les capteurs de température que l'on souhaite avec leur étalonnage (voir Figure IV.19). Si on sélectionne « Absorbance » alors il est possible de voir le modèle du spectromètre, on peut sélectionner la longueur d'onde à laquelle on souhaite mesurer durant l'expérience, on peut choisir également le temps d'intégration (en ms ou en sec) et enfin de faire la mesure du blanc et du noir avec des indicateurs à droite. Si on change le temps d'intégration ou la longueur d'onde alors les indicateurs se remettent au rouge et il est nécessaire de refaire des mesures du blanc et du noir afin d'avoir les valeurs correspondant aux paramètres choisis. Enfin il y a la possibilité de sélectionner « Etat de charge » mais uniquement si « Température » et « Absorbance » sont sélectionnés car en effet, il n'est pas possible de mesurer l'état de charge sans les paramètres de température et d'absorbance. Dans cette section il est demandé d'indiquer la concentration des électrolytes et l'équation  $\epsilon$  qui permet d'adapter ce paramètre dans le calcul de l'état de charge car il dépend de la température de l'électrolyte.

Il est tout à fait possible de sélectionner « Température » et « Absorbance » sans sélectionner « Etat de charge ».

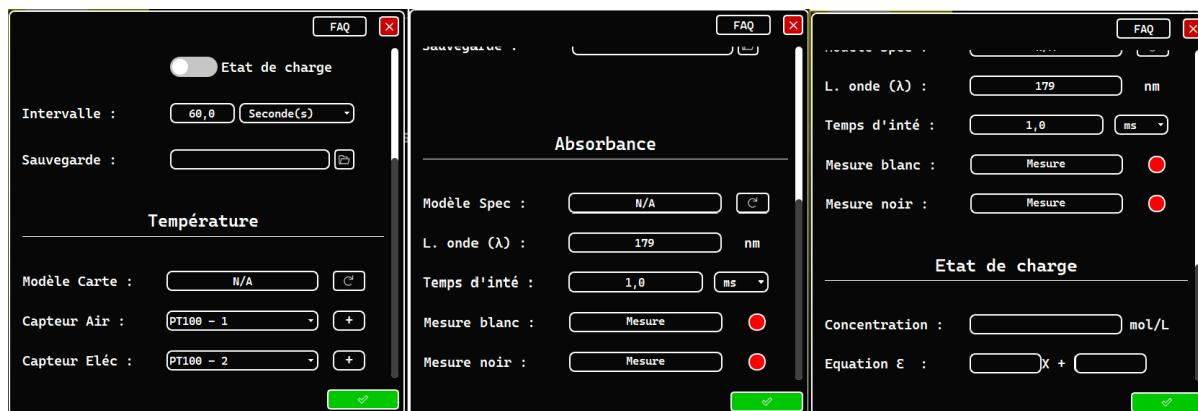


Figure IV.10 : Interface de démarrage en fonction des sélections

On peut voir une autre option sur les figures IV.11 et IV.12, on remarque qu'il y a des onglets « Réglages ».

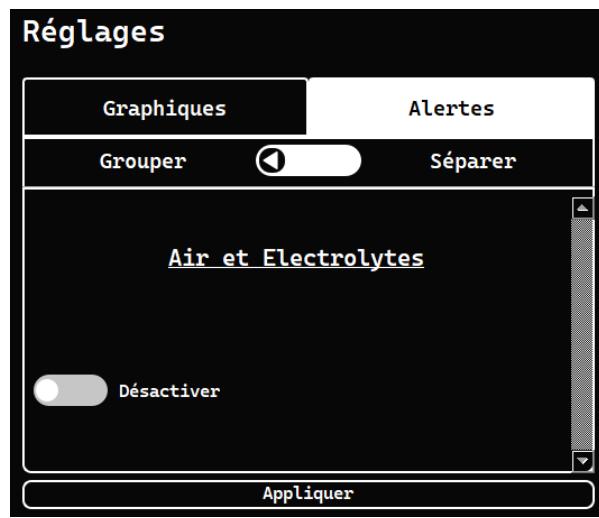
L'onglet « Réglages » permet de faire plusieurs modifications et a plusieurs fonctionnalités :

Dans la partie « Graphique » nous avons, comme son nom l'indique toutes les modifications possibles relatives aux graphiques. Nous avons la possibilité de modifier la vue d'intervalle de temps de l'axe X en écrivant le chiffre souhaité (entre 1 et 60) et en choisissant l'ordre de grandeur (Secondes, Minutes, Heures ou Jours). Pour ce qui est de l'axe Y nous pouvons indiquer la vue dont nous souhaitons en indiquant les limites de vue donc par exemple à la figure X on remarque que l'on a mis l'axe Y de -10°C à 50°C et donc cela résulte sur le graphique de la vue de -10°C à 50°C (Figure IV.2).

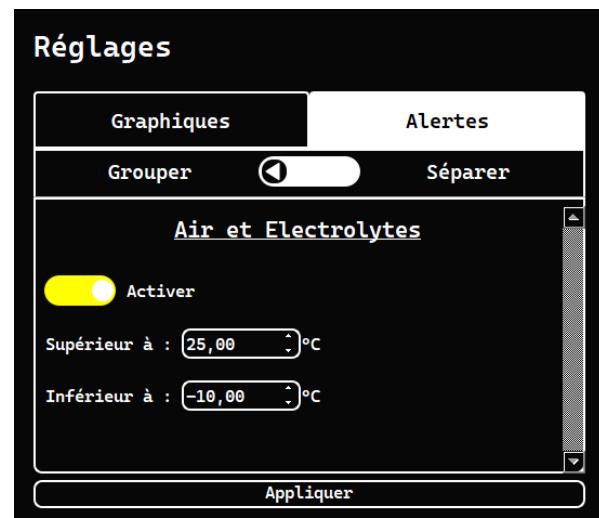
# PROJET TUTEURE MCPC

Dans la partie « Alertes » nous avons la possibilité d'activer comme son nom l'indique des alertes qui permettent d'être notifier lorsque la valeur de la température et/ou de l'absorbance dépasse un seuil imposé par l'utilisateur représenté par des lignes rouge et bleues sur les graphiques (Figure IV.2 et IV.13). Nous avons la possibilité de les activer ou de les désactiver afin que dans certains cas d'expérience le canal de notifications ne soient pas trop encombrés avec des alertes inutiles.

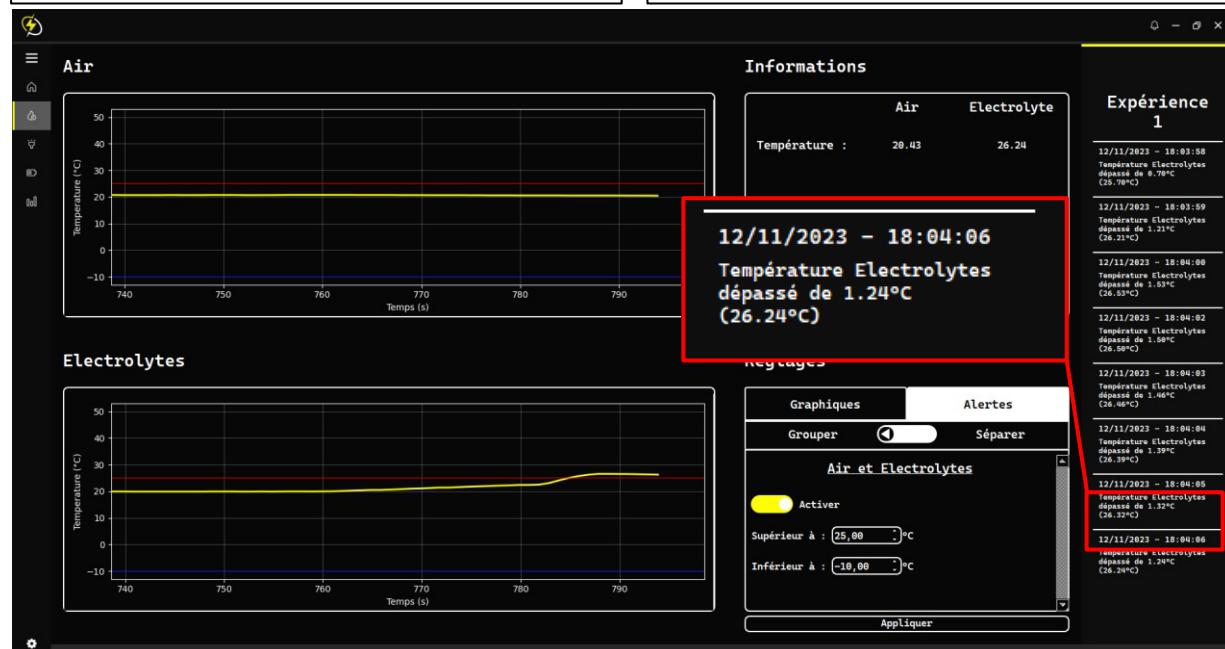
Pour le cas de la température il y a une option en plus qui est « Grouper » ou « Séparer », cette option est déterminée par le côté où se situe le rond blanc. Si le rond blanc est à gauche donc sur l'option « Grouper » alors on peut modifier les paramètres des graphiques Air et Electrolytes et/ou des alertes en même temps afin de ne pas être obligé à écrire deux fois la même chose et donc l'option « Séparer » permet d'avoir des paramètres différents pour chaque graphique que ce soit sur la vue ou sur les alertes.



**Figure IV.11 : Alertes de température désactiver**



**Figure IV.12 : Alertes de température activer**



**Figure IV.13 : Vue des graphiques avec les alertes de température**

# PROJET TUTEURE MCPC

Dans le logiciel Techvolt nous avons la possibilité de planifier des expériences afin d'optimiser et d'automatiser au maximum les expériences à effectuer. On remarque sur la Figure IV.16 en arrière-plan qu'il y a un bouton « Planifier ». Ce bouton à les mêmes fonctions que le bouton « Démarrer » mais avec une section planification en plus qu'on peut voir sur la figure IV.14.

Dans cette section nous pouvons choisir la date du début de l'expérience et la fin de celle-ci avec l'heure. Nous avons en-dessous un indicateur qui nous renseigne sur la durée de l'expérience et enfin la possibilité de choisir la couleur de l'expérience (Figure IV.15) qui sera ensuite affiché dans la section planification et sur le calendrier pour avoir un affichage visuel (Figure IV.16).

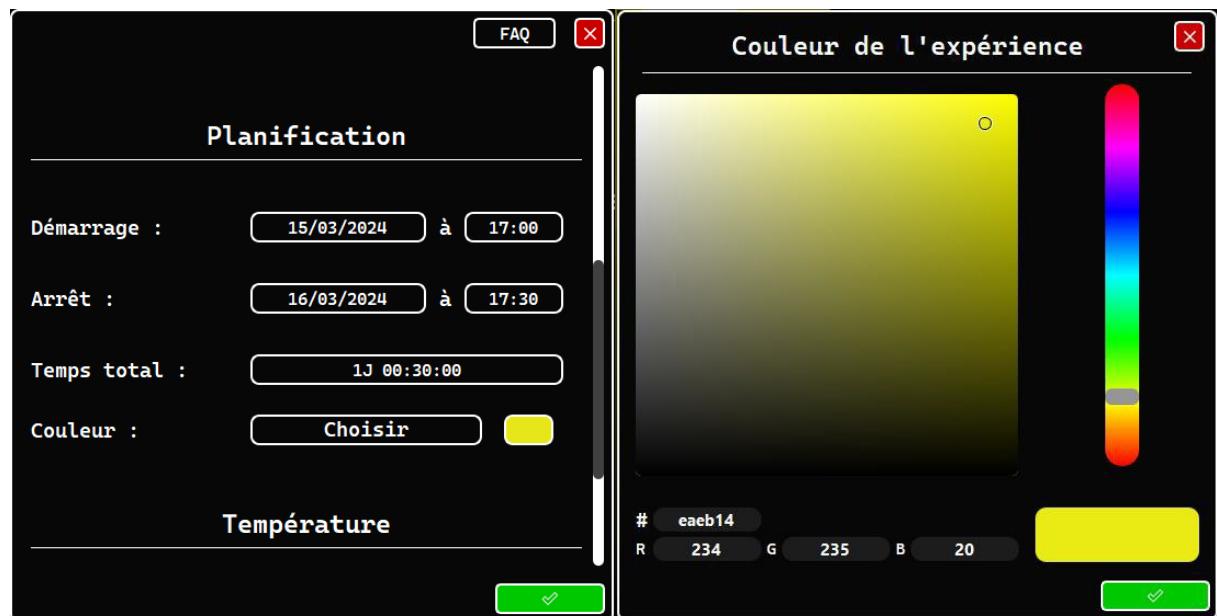


Figure IV.14 : Section de planification

Figure IV.15 : Sélecteur de couleur

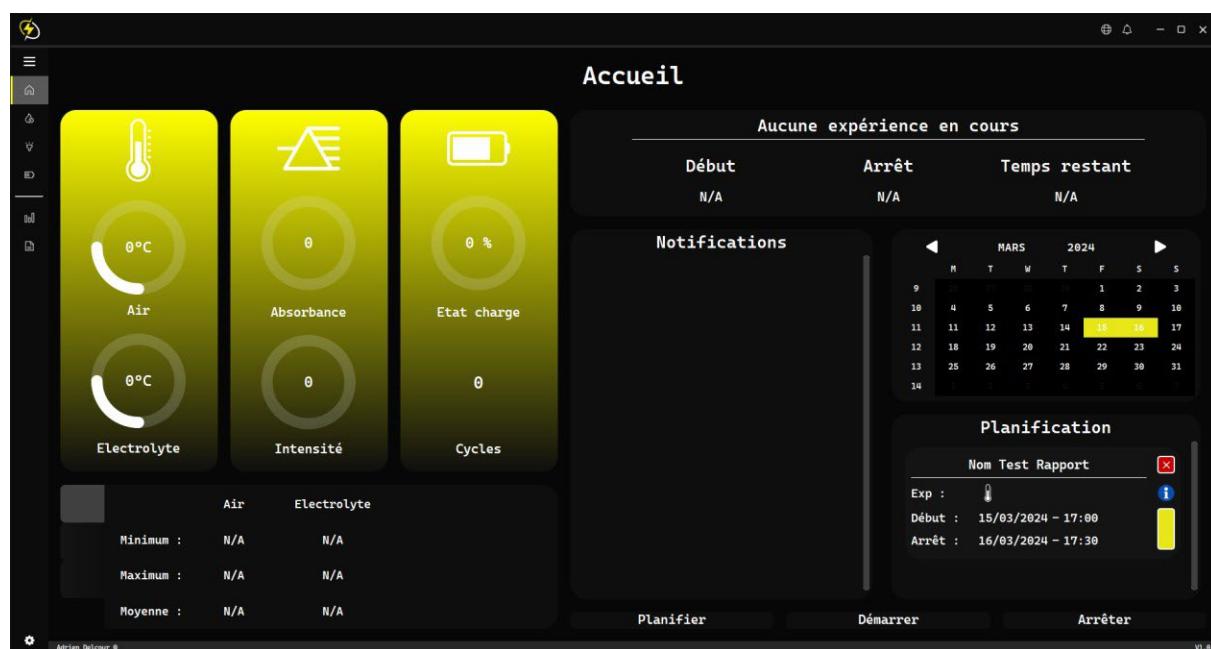


Figure IV.16 : Interface avec une expérience planifier

# PROJET TUTEURE MCPC

On remarque qu'une fenêtre s'est créée dans la section « Planification » avec quelques informations tels que les expériences choisi (Température, Absorbance, Etat de charge) avec les dates de début et de fin de l'expérience. Si l'on souhaite avoir toutes les informations relatives à l'expérience il suffit de cliquer sur le « i » afin d'avoir toutes les informations.

## 5°) Lancement d'une manipulation

Nous avons maintenant tous les capteurs qui sont opérationnels mais il faut que les mesures du blanc et du noir soient effectuées avant la manipulation, il faut également créer un fichier qui permet de sauvegarder toutes les données et enfin d'avoir un bouton qui lance tous les capteurs en même temps afin d'être synchroniser. Voici l'interface créer afin de répondre à toutes ces exigences :



Figure IV.17 : Vue de la fenêtre permettant de lancer une manipulation

La fenêtre visible figure IV.17 permet de sélectionner un preset d'expériences ou de choisir les expériences que l'on souhaite faire avec l'intervalle de temps entre chaque mesure et le chemin de sauvegarde de notre fichier txt ou csv.

Les preset permettent de sélectionner automatiquement les boutons pour une certaines expérience afin de commencer cette dernière le plus rapidement possible et pour simplifier la vie de l'utilisateur. Il est possible de créer un nouveau preset en sélectionnant nos options d'expériences et de cliquer sur le bouton bleu. A ce moment il est possible de saisir un nom de preset qui sera trouvable dans le sélecteur de preset (bouton à la gauche du bouton bleu de sauvegarde de preset) qu'on peut voir figure IV.18. Il y a également une barre de recherche dans la sélection du preset afin de retrouver celui que l'on souhaite au plus vite.

Il y également l'ajout d'un bouton « FAQ » qui, lorsque l'on clique dessus, permet de nous rediriger vers le site de Techvolt afin de répondre à un maximum de questions que les utilisateurs pourraient se poser.

# PROJET TUTEURE MCPC

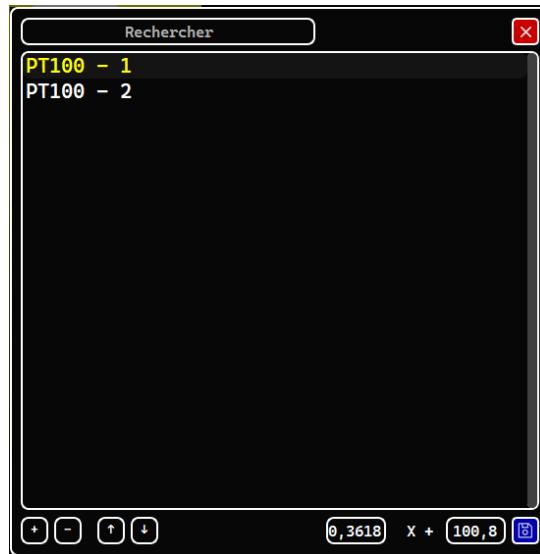
Lien de la FAQ : <https://techvolt-infos.github.io/FAQ.html>

Lien pour accéder à l'accueil du site TechVolt : <https://techvolt-infos.github.io/>



**Figure IV.18 :** Vue de la fenêtre permettant de configurer les presets

Dans la partie « Température » nous avons l'indication où l'Arduino est détecter (ici Port COM 5), cette détection se fait automatiquement comme la méthode pour le spectromètre. Dans les sections « Air » et « Eléc. » nous avons la possibilité de choisir les différents capteurs que nous pouvons rajouter avec la fenêtre de dialogue (Figure IV.19).



**Figure IV.19 :** Vue de la fenêtre permettant de configurer les capteurs

On peut y voir qu'on peut assigner une fonction linéaire à chaque capteur afin de modifier l'étalonnage du capteur si besoin, cela permet d'éviter de modifier directement le programme Arduino et il y a également une barre de recherche afin de trouver rapidement le capteur que l'on

# PROJET TUTEURE MCPC

souhaite utiliser. Malheureusement nous avons quelques difficultés à transmettre l'information vers la carte Arduino donc pour l'instant cette option ne fonctionne pas et les capteurs gardent leur étalonnage d'origine inscrit dans le programme Arduino.

Dans la partie « Absorbance » nous avons le modèle du spectromètre détecter qui est ici FLMS17687 qui est détecter automatiquement avec la méthode Figure IV.20. Nous avons la mesure du blanc et du noir que l'on peut effectuer en cliquant sur les boutons assignés à ces derniers. Il y a des indicateurs à droite des boutons qui permettent de savoir si la mesure du blanc et/ou du noir a été correctement effectué, sur la figure IV.20 on remarque que la mesure du blanc a été faites mais pas celle du noir. Il y a aussi un bouton à côté du modèle afin de rafraîchir la connexion, cela permet que même si l'on ouvre la fenêtre de manipulation sans connecter le spectromètre alors il suffit de rafraîchir une fois que l'on connecte le spectromètre. Cela évite de fermer et de rouvrir la fenêtre pour l'initialisation du spectromètre.

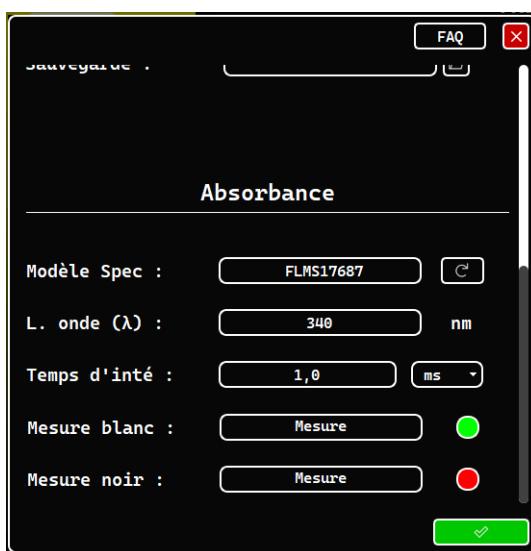


Figure IV.20 : Fenêtre permettant de saisir les paramètres du spectromètre

Enfin nous avons le chemin de sauvegarde qui s'écrit automatiquement si l'on clique sur le bouton « Parcourir » car cela ouvre une fenêtre d'explorateur de fichier Windows qui permet d'entrer un nom pour notre fichier et le format (.txt ou .csv).

Temps	Temperature Air	Temperature Electrolyte	Absorbance
1.02	19.82	20.61	0.7344
2.02	19.82	20.57	0.7344
3.01	19.82	20.61	0.7283
4.01	19.79	20.57	0.7268
5.03	19.79	20.57	0.7268
6.04	19.82	20.61	0.7205
7.04	19.82	20.57	0.7167
8.04	19.79	20.57	0.7167

Figure IV.21 : Données enregistrer dans un fichier .txt pendant une manipulation

Ici, les capteurs prennent une mesure toute les secondes mais on remarque qu'il y a de légères fluctuations, nous ne connaissons pas la cause exacte du problème dans le programme mais on peut supposer que le programme doit charger plusieurs fonctions à la fois (comme la fonction pour

## PROJET TUTEURE MCPC

calculer l'absorbance) et donc il y a un temps de décalage qui se créer au fur et à mesure de la manipulation.

Pour lancer la manipulation il suffit de cliquer sur le bouton « ✓ » si tous les paramètres sont corrects mais si ce n'est pas le cas alors une alerte s'affiche afin d'indiquer à l'utilisateur ce qui ne va pas et ce qu'il doit faire avec une image pour savoir directement ce qui ne va pas, voici les différentes alertes possibles :



Figure IV.22 : Types d'erreur possibles lors du démarrage d'une manipulation

### 6°) Exploitation des résultats et rapports

Cette partie du rapport traitera d'une section qui n'est pas encore abouti. En premier lieu il y a l'exploitation des résultats, c'est une partie du logiciel qui servira à importer une expérience et de pouvoir glisser-déposer à partir d'un répertoire directement intégrer dans le logiciel (voir Figure IV.23). Une fois cela fait alors le logiciel trace les différentes courbes et l'on pourra sélectionner les paramètres que l'on souhaite afficher. Il y aura la possibilité de sélectionner certaines portions du graphique, de zoomer/dezoomer dessus afin d'appliquer des calculs et enfin de pouvoir sauvegarder l'image du graphique et les résultats souhaiter.

Ensuite il y a une nouvelle section qui permettra de créer des rapports de façon automatique en important tout d'abord l'expérience que l'on souhaite rédiger. A droite de la fenêtre se situera toutes les variables qui correspondent aux résultats de l'expérience. Il suffira d'en sélectionner une et de la glisser déposer afin que celle-ci s'écrivent sur le rapport. Il y aura des modèles pré-enregistrer afin qu'ils ne restent plus qu'à glisser-déposer.

A l'heure actuelle la partie du traitement des résultats d'une expérience n'est pas opérationnel et nous font fassent à différents bugs qui compliquent le développement de cette fonctionnalité et la partie du logiciel sur les rapports automatiques ne permet que de faire du traitement de texte simple, la fonction de glisser-déposer et les modèles ne sont pas encore incorporer au logiciel mais on peut voir sur la Figure IV.24 à quoi ressemble à l'heure actuelle cette partie du logiciel :

# PROJET TUTEURE MCPC



Figure IV.23 : Section du logiciel TechVolt sur le traitement des résultats

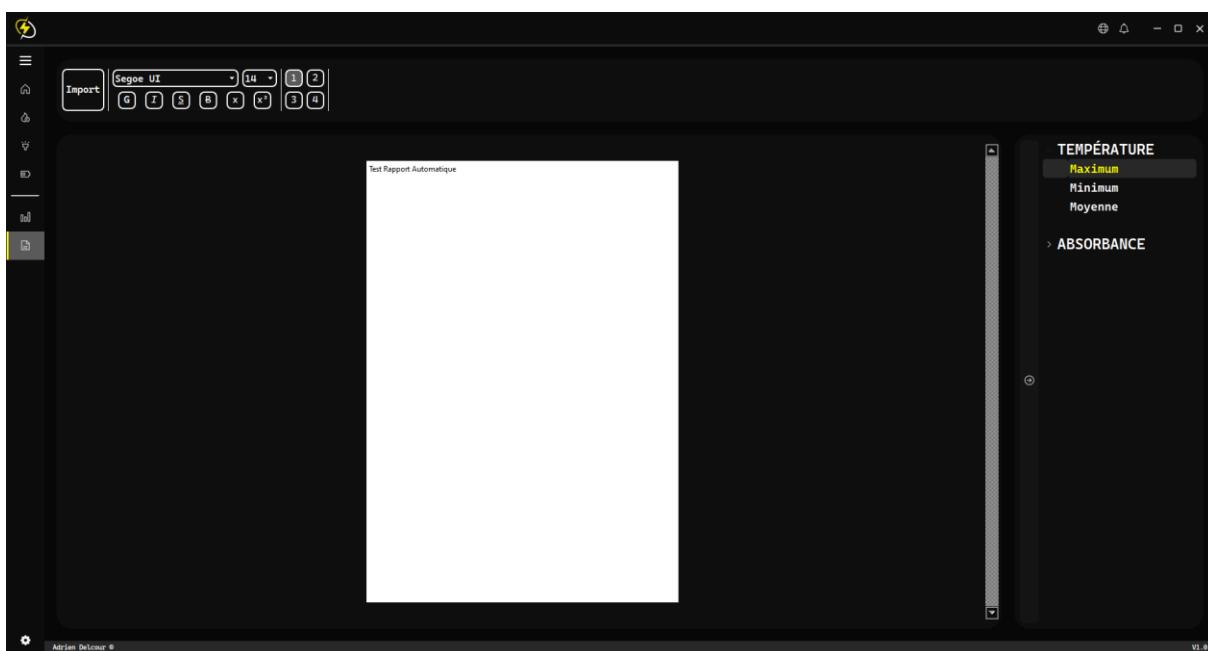


Figure IV.24 : Section du logiciel TechVolt sur les rapports automatiques

Pour résumer nous avons aujourd’hui la possibilité de lancer ou de planifier une manipulation avec deux capteurs de température et un spectromètre où leurs données sont enregistrées dans un fichier txt ou csv et qui sont tracer en temps réel avec la mesure de l’état de charge bientôt disponible avec pour complément l’analyse des résultats et de la rédaction automatique de rapports.