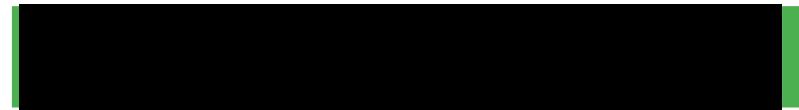




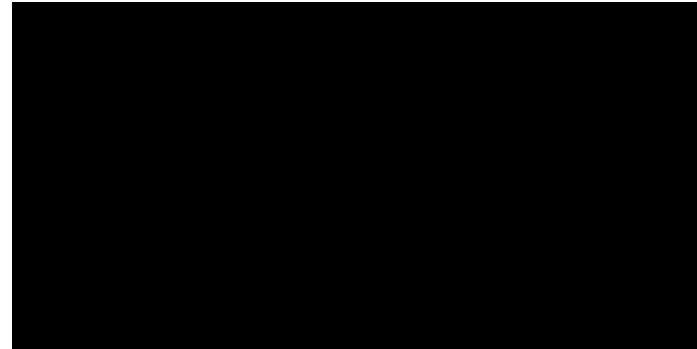
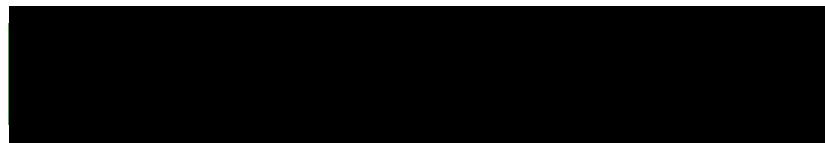
POLYTECHNIQUE
MONTRÉAL

UNIVERSITÉ
D'INGÉNIERIE



Conception d'un instrument de mesure électrochimique

(Potentiostat)



Département de génie biomédical

Polytechnique Montréal

Vendredi 18 avril 2025

Table des matières

1. Introduction.....	3
2. Organisation de l'équipe.....	4
2.1. Répartition des tâches et responsabilités	4
2.2. Organisation du travail.....	5
2.3. Plan de travail élaboré	7
3. Principe de fonctionnement du Potentiostat	10
3.1. Élaboration du circuit.....	10
3.1.1. Explication du circuit de base	10
3.2. Génération et modulation du signal envoyé au potentiostat	11
3.3. Traitement du courant en sortie de la cellule électrochimique.....	13
3.4. Calculs des valeurs pour les composantes du circuit	15
3.4.1. Circuit de génération du signal.....	15
3.4.2. Circuit de traitement du signal	15
3.5. Temps caractéristique et réponse temporelle du circuit	17
3.6. Mesure de protection de la plateforme.....	17
3.7. Photo du circuit complet sur PCB	18
4) Programmation	19
4.1. Organisation de la communication sérielle.....	19
4.2. Envoi de différents types de signaux modulables à partir de l'Uno	21
4.3. Gestion de la mesure simultanée du courant (échantillonnage)	23
4.4. Affichage des données courant – potentiel et discussion sur leur résolution.....	24
Résolution théorique de la tension	24
Résolution théorique du courant	24
4.5. Méthode de test pour valider la précision de l'appareil	25
5. Mesures chimiques.....	29
5.1 Voltammetrie.....	29
5.2 Chronamperométrie	35
5.3 Limites de votre système et les sources d'erreurs	37
6 Développement durable	40
Bibliographie	41
Annexe	41

1. Introduction

L'homéostasie du corps humain est un équilibre précaire concernant une multitude de molécules différentes et qui permet à l'organisme de remplir ses fonctions. La détection d'une variation quelconque dans cet équilibre est une des premières étapes dans le diagnostic d'une pathologie puisque c'est une des choses les plus faciles à vérifier chez un patient, et parce que tout changement de l'homéostasie peut impliquer des conséquences drastiques. Plusieurs technologies permettent d'analyser la composition des fluides biologiques, notamment le sang, de façon fiable et rapide, en ne nécessitant qu'un petit échantillon. Parmi les plus utilisées, on note par exemple les centrifugeuses, la spectroscopie, l'analyse par microscope et le sujet de ce travail, les capteurs électrochimiques. Parmi ces derniers, les potentiostats maintiennent un voltage précis entre deux électrodes plongées dans la solution à analyser tout en mesurant le courant créé par la réaction d'oxydo-réduction du milieu.

On a alors deux courants; un faradique, lié à la concentration de la molécule à analyser, et un capacitif, puisque l'électrode de travail est non polarisable et se comporte comme un condensateur en solution. En tout, un potentiostat est composé de trois électrodes; une de travail, une de référence et une contre-électrode. Le circuit contrôle la différence de tension entre l'électrode de travail et l'électrode de référence, qui est stable; la réaction d'oxydation ou de réduction se produit à l'électrode de travail et la contre-électrode offre un chemin par où le courant peut passer afin que l'électrode de référence ne soit pas affectée. Pour maintenir la différence de tension désirée entre les électrodes, le circuit prend une source de tension connue et agit comme un amplificateur opérationnel, avec comme fonction le fait de contrôler ce potentiel avec précision. La deuxième partie du circuit mesure le courant entre les électrodes et le transforme en potentiel afin qu'il puisse être lu facilement. Le fonctionnement de notre potentiostat, dont la source de potentiel est le microcontrôleur Arduino, sera décrit plus en détail dans la section 3 de ce rapport.

L'intérêt d'un tel potentiostat est dans sa simplicité, son cout de fabrication et sa taille. Les senseurs électrochimiques utilisés dans le domaine médical peuvent être très chers, encombrant et difficiles à déplacer. Or, des patients pourraient avoir une pathologie nécessitant de faire des tests chimiques réguliers, l'exemple le plus évident étant les patients diabétiques. Également, certains patients en région rurale ou dans des pays en voie de développement habitent potentiellement à plusieurs heures d'un hôpital. La nécessité d'un appareil pouvant mesurer la concentration d'un substrat chimique rapidement et de façon fiable est donc un réel enjeu pour accommoder les patients et alléger le rôle du personnel médical. D'autres réalités rendent également ces appareils nécessaires, tel que les cliniques mobiles. Enfin, un potentiostat comme celui de ce projet présente un potentiel d'analyse de données automatisée sans besoin de l'intervention d'un professionnel de la santé, en le connectant par exemple à un appareil mobile appartenant au patient. Bref, le

développement de senseurs électrochimiques miniaturisés, de faible cout et de conception simple est d'un grand intérêt médical pour ses nombreuses applications possibles.

2. Organisation de l'équipe

2.1. Répartition des tâches et responsabilités

Avant de débuter le travail, nous nous sommes concentrés sur une analyse commune des grandes lignes du projet. À notre première rencontre du mercredi 26 février, nous avons fait une lecture commune du document de présentation de projet, afin d'en ressortir les principaux thèmes. Le but de cette rencontre était de diviser le projet en sous-divisions claires en accord avec les objectifs à atteindre. Cette première étape nous a permis d'avoir une compréhension d'ensemble du travail à réaliser et de s'assurer que nous étions tous sur la même longueur d'onde. Le résultat de cet exercice est présenté à la Figure 2.1.1.

Modélisation: <ul style="list-style-type: none">• Compréhension des composantes principales d'un potentiostat• Compréhension des deux méthodes de tests électriques, soit la voltanométrie et la chronampérométrie	Traitemenent de données <ul style="list-style-type: none">• Compréhension de la forme et des caractéristiques des données à chaque niveau du circuit électrique• Comment traiter les données reçues en provenance de la solution numériquement?
Programmation <ul style="list-style-type: none">• Élaborer une structure du programme Arduino permettant de répondre aux requis de l'énoncé du projet.• Etablir la logique du code pour les deux méthodes de tests électriques	Plateforme Arduino <ul style="list-style-type: none">• Compréhension du fonctionnement de l'Arduino dans le cadre des objectifs du projet• Compréhension des limitations de l'Arduino dans le contexte du signal reçu et envoyé.

Figure 2.1.1. Schéma des sous-divisions principales du projet

Suite à cet exercice, l'équipe a obtenu une compréhension commune des grandes divisions du projet. Sur cette base commune, des rôles de gestion et des tâches spécifiques au sein de l'équipe pour assurer le bon déroulement du projet ont été établis. Les tâches ont été distribués en fonction des intérêts et forces de chaque membre.

a. Rôles :



Le chef d'équipe a comme responsabilité de superviser les travaux de l'équipe ainsi que de coordonner la planification du travail avec les membres de l'équipe.

[REDACTED]

Le secrétaire a comme responsabilité de suivre l'avancement du travail de chaque membre comme prévu et d'organiser les réunions régulières. Il organise et catalogue tous les documents du projet tout au long de sa réalisation dans le groupe Teams créé à cet effet.

b. Tâches :

Équipe Circuit électrique :

- Lead Circuit d'envoi du signal électrique

A comme tâche de concevoir le circuit électrique permettant de transmettre, traiter et stabiliser le signal en provenance de l'Arduino vers la solution chimique.

- Lead Circuit de réception du signal électrique

A comme tâche de concevoir le circuit électrique permettant de recevoir, traiter et stabiliser le signal en provenance de la solution chimique vers l'Arduino.

- Lead Programmation:

A comme tâche de concevoir le programme Arduino pouvant fournir les tests de voltanométrie et chronampérométrie, de recevoir les signaux en provenance de la solution chimique et d'afficher les résultats.

2.2. Organisation du travail

Afin d'organiser les différents volets du projet, chaque *Lead* a élaboré des étapes précises en fonction des objectifs de chaque volet. À chaque rencontre d'équipe, tel que présenté à la figure 2.2.1., nous discutions des avancements de chaque volet du projet, des problèmes rencontrés et nous ajustions des objectifs à atteindre pour chaque *Lead* avant la prochaine rencontre la semaine suivante.

FÉVRIER 2025							MARS 2025						
SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
26	27	28	29	30	31	1	23	24	25	26	27	28	1 RENCONTRE INITIALE
2	3	4	5	6	7	8	3	4	5	6	7	8	9
9	10	11	12	13	14	15	10 RENCONTRE A	11	12 RENCONTRE B	13	14	15	
16	17	18	19	20	21	22	16	17 RENCONTRE A	18	19 RENCONTRE B	20	21	22
23	24	25	26 DÉBUT DU PROJET	27	28	1	23 RENCONTRE A	24	25 RENCONTRE B	26	27	28	29
							30	31 RENCONTRE A	1	2	3	4	5

AVRIL 2025						
SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
30	31	1	2 RENCONTRE B	3	4	5
8	9 MESURES CHIMIQUES - 1	10	11 MESURES CHIMIQUES - 2	12		
15	16 MESURES CHIMIQUES - 3	17	18	19 REMISE DU RAPPORT		
22	23	24	25	26		
29	30	1	2	3		

Figure 2.2.1. Calendrier global des rencontres et dates importantes du projet

Le calendrier de la figure 2.2.1. présente trois types de rencontre que l'on a eu lors de la session de projet.

a. Rencontre Initiale :

Après avoir lu le document de projet, l'équipe s'est rencontré une première fois via la plateforme Teams pour discuter des objectifs du projet. Les sous-divisions principales ainsi que la répartition des tâches du projet ont été établis lors de cette rencontre initiale.

b. Rencontre A :

Les rencontres de type A se déroulaient au début de la période de laboratoire du lundi. Nous nous réunissions pour discuter des avancements de la semaine en cours. Ces rencontres étaient courtes et étaient centralisées autour de la mise à jour commune de l'équipe sur l'avancement des différentes divisions du projet.

c. Rencontre B :

Les rencontres de type B se déroulaient au début de la période laboratoire du mercredi. Lors de ces rencontres, nous discutions en détails des objectifs à atteindre pour la semaine qui suit de chaque sous-division du projet. Une fois ces objectifs mis en place, l'équipe se remettait au travail.

2.3. Plan de travail élaboré

Le tableau suivant présente la progression de chaque *Lead* durant l'entièreté de la réalisation du projet ainsi que la complétion progressive des objectifs de chaque sous-division.

Semaine	Tâches de la semaine	Progression des objectifs
10 mars	Circuit de génération du signal <ul style="list-style-type: none"> - Définir les contraintes à respecter dans le cadre de la conception du circuit (PWM Arduino, alimentation ampliops) - Définir la nature du signal qu'on doit moduler (onde triangulaire entre 0 et 5 V) - Définir le signal qui doit arriver aux électrodes (onde triangulaire lisse entre -1 et 1V) 	1 Compréhension du circuit – 50% 2 Définitions des valeurs de condensateurs et de résistances – 0% 3 Test sur <i>breadboard</i> du circuit – 0% 4 Construction sur <i>PCB</i> du circuit – 0%
	Circuit de réception du signal <ul style="list-style-type: none"> - Définir les contraintes à respecter dans le cadre de la conception du circuit de réception du signal (principalement Arduino) - Définir la nature du signal obtenue en provenance de la solution - Poser une approche de conversion du courant en tension 	5 Compréhension du circuit – 50% 6 Définitions des valeurs de condensateurs et de résistances – 0% 7 Test sur <i>breadboard</i> du circuit – 0% 8 Construction sur <i>PCB</i> du circuit – 0%
	Programmation <ul style="list-style-type: none"> - Faire et terminer le Pseudo-code - Déterminer comment faire l'échantillonnage en 10bits pour lissage - Générer un signal triangulaire dans une fonction - Générer un créneau (signal rectangulaire) dans une fonction - Pensée à comment synchroniser temps et lecture 	1. Pseudo-code – 100% 2. Section communication serielle du code – 0% 3. Section gestion des fonctions du code 4. Code voltammetrie cyclique – 0% 5. Code chronampérométrie – 0% 6. Section échantillonnage – 0% 7. Rédaction section 4 du rapport – 0%

17 mars	Circuit de génération du signal	<ul style="list-style-type: none"> - Définir et comprendre les rôles des trois ampli-ops - Comprendre le fonctionnement du premier filtre RC - Commencer les calculs pour justifier les valeurs des résistances 	<ol style="list-style-type: none"> 1. Compréhension du circuit – 100% 2. Définitions des valeurs de condensateurs et de résistances – 75% 3. Test sur <i>breadboard</i> du circuit – 0% 4. Construction sur <i>PCB</i> du circuit – 0%
	Circuit de réception du signal	<ul style="list-style-type: none"> - Analyse des deux amplificateurs à utiliser dans le circuit ; utilité, fonctionnement, paramètres en lien avec l'Arduino - Tentative préliminaire de calculs des valeurs des condensateurs et résistances. 	<ol style="list-style-type: none"> 5. Compréhension du circuit – 100% 6. Définitions des valeurs de condensateurs et de résistances – 50% 7. Test sur <i>breadboard</i> du circuit – 0% 8. Construction sur <i>PCB</i> du circuit – 0%
	Programmation	<ul style="list-style-type: none"> - Écrire la section de la communication sérielle avec une machine état - Écrire la fonction de temps, de création de graphique, de chronoampérométrie et voltammetrie cyclique - Tester le code sur le circuit tinkerCad et sur le filtre RC - Répondre à la question 4.1 - Penser à comment faire les tests 	<ol style="list-style-type: none"> 1. Pseudo-code – 100% 2. Section communication sérielle du code – 80% 3. Section gestion des fonctions du code – 80% 4. Code voltammetrie cyclique – 100% 5. Code chronampérométrie – 100% 6. Section échantillonnage – 80% 7. Rédaction section 4 du rapport – 20%
24 mars	Circuit de génération du signal	<ul style="list-style-type: none"> - Terminer les calculs - Construire le circuit et le tester sur <i>breadboard</i> 	<ol style="list-style-type: none"> 1. Compréhension du circuit – 100% 2. Définitions des valeurs de condensateurs et de résistances – 100% 3. Test sur <i>breadboard</i> du circuit – 100% 4. Construction sur <i>PCB</i> du circuit – 0%
	Circuit de réception du signal	<ul style="list-style-type: none"> - Finalisation des calculs des valeurs de résistance et de condensateurs - Construction du modèle de circuit sur <i>breadboard</i> - Tests sur <i>breadboard</i> du gain de tension de 0V à 5V en sortie pour une entrée de -200 uA à 200 uA - Début de la construction 	<ol style="list-style-type: none"> 5. Compréhension du circuit – 100% 6. Définitions des valeurs de condensateurs et de résistances – 100% 7. Test sur <i>breadboard</i> du circuit – 75% 8. Construction sur <i>PCB</i> du circuit – 0%

	Programmation <ul style="list-style-type: none"> - Effectuer les tests de la question 4.5 pour différents paramètres de vitesse et d'amplitude (enregistrer les fichier txt) - Préparer le test global avec une modélisation par une résistance - Corriger la fréquence d'échantillonnage pour une mesure par marche et réduire freq d'affichage 	<ol style="list-style-type: none"> 1. Pseudo-code – 100% 2. Section communication sérielle du code – 80% 9. Section gestion des fonctions du code – 80% 3. Code voltammetrie cyclique – 100% 4. Code chronampérométrie – 100% 5. Section échantillonnage – 100% 6. Rédaction section 4 du rapport – 40%
31 mars	Circuit de réception du signal <ul style="list-style-type: none"> - Validation finale du circuit complet avec tests sur le circuit d'envoi du signal - Construction du circuit PCB 	<ol style="list-style-type: none"> 1. Compréhension du circuit – 100% 2. Définitions des valeurs de condensateurs et de résistances – 100% 3. Test sur <i>breadboard</i> du circuit – 100% 4. Construction sur <i>PCB</i> du circuit – 100%
	Programmation <ul style="list-style-type: none"> - Répondre question 4.2 - Corriger le bogue du reset - Effectuer le test complet du circuit avec la programmation, envoie et acquisition avec une modélisation de la solution par une résistance (enregistrer fichier txt) - Vérifier pour s'assurer que le code fonctionne pour les premier test chimique 	<ol style="list-style-type: none"> 1. Pseudo-code – 100% 2. Section communication sérielle du code – 100% 3. Section gestion des fonctions du code – 100% 4. Code voltammetrie cyclique – 100% 5. Code chronampérométrie – 100% 6. Section échantillonnage – 100% 7. Rédaction section 4 du rapport – 50%
7 avril – 18 avril	TESTS CHIMIQUES ET RÉDACTION DU RAPPORT	<ol style="list-style-type: none"> 1. Test préliminaires – 2. Correction de bug dans le code afin de réduire le bruit 3. Prise de mesures 4. Reprises des mesures aberrantes

3. Principe de fonctionnement du Potentiostat

3.1. Élaboration du circuit

3.1.1. Explication du circuit de base

Le circuit de base d'un potentiostat est composé d'un amplificateur opérationnel suivi de trois électrodes à fonctions dédiées.

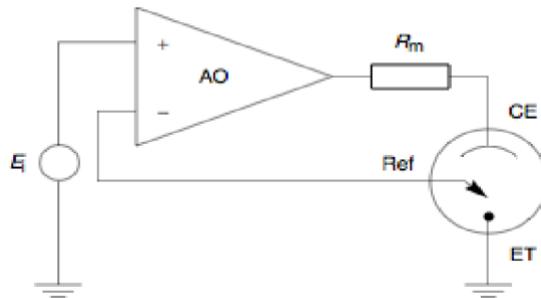


Figure 3.1.1.1. Schéma du circuit de base d'un potentiostat [1]

Les composantes de ce circuit travaillent ensemble pour effectuer les deux fonctions suivantes :

1. Induire une réaction d'oxydo-réduction dans la solution d'analyse
2. Mesurer le courant faradique en réponse à la réaction d'oxydo-réduction

Les 3 électrodes présentés dans la Figure 3.1.1 sont les électrodes de travail, de référence et la contre-électrode. Chaque électrode occupe une fonction spécifique dans le circuit de base.

ET : Électrode sur laquelle la réaction d'oxydo-réduction se fera. Elle est mise directement en contact avec la solution et réagit avec les molécules désirées. Cette électrode ne doit pas réagir lorsqu'elle est mise en solution et son potentiel ne doit pas changer. Elle est généralement mise à la terre. On mesure le courant produit par l'échange d'électrons de la réaction d'oxydo-réduction à cette électrode.

Réf : Électrode qui fournit le potentiel permettant à la réaction d'oxydo-réduction de se faire. Ce potentiel va agir entre **Réf** et **ET**. Ainsi, le potentiel fournit permettra aux électrons d'une molécule spécifique de circuler entre **ET** et la solution, causant ainsi le courant faradique que l'on peut mesurer. Le potentiel fournit à travers **Réf** est contrôlé et doit être constant, c'est pourquoi l'on utilise un amplificateur opérationnel suiveur dont l'utilité sera définie dans une section subséquente.

CE : Électrode qui n'agit pas directement dans la mesure du courant faradique et la prise des données. Elle permet simplement de fournir un passage aux électrons générés par la réaction d'oxydo-réduction hors de la solution. Si cette électrode n'était pas présente, le courant passerait à travers **Réf** et modifierait ainsi son potentiel qui est à la source de la réaction d'oxydo-réduction. **CE** permet d'absorber ce courant, laissant ainsi **Réf** intact et avec un potentiel constant et connu.

L'amplificateur suiveur **A0** permet de stabiliser et maintenir le potentiel appliqué à l'électrode de référence **Réf**. Dans le cas idéal, la tension appliquée à la borne positive de l'ampli est directement transmise à la borne négative. Le courant est nul aux entrées de l'amplificateur en raison de l'impédance d'entrée élevée, préservant ainsi l'intégralité de la tension. La sortie de l'amplificateur suiveur est de faible impédance, ce qui permettra au courant de la contre-électrode **CE** de passer préférentiellement. Cette différence d'impédance est à la source de la sélectivité entre **CE** et **Réf**.

3.2. Génération et modulation du signal envoyé au potentiostat

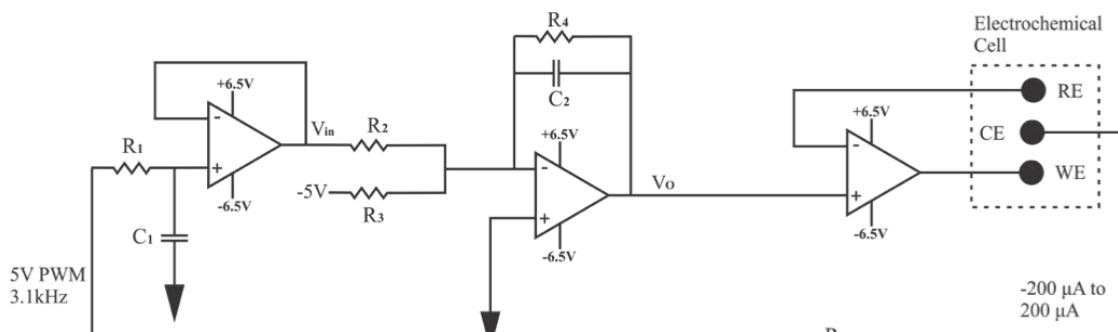


Figure 3.2.1. Schématisation de la première moitié du circuit, de l'Arduino aux électrodes [2]

La première moitié du circuit (figure x) doit adapter le voltage envoyé par l'Arduino pour transmettre le signal désiré au potentiostat. Elle doit pour cela remplir deux fonctions principales, soit de lisser le signal PWM du microcontrôleur et moduler la plage de tension envoyée (0 à 5V) pour qu'elle corresponde aux besoins des tests chimiques. Au total, cette partie du circuit comporte deux filtres, deux amplificateurs opérationnels suiveurs et un amplificateur opérationnel sommateur.

Filtre 1 (C1 R1). L'Arduino envoie la tension dans le circuit sous la forme d'un signal PWM qui alterne entre 0 et 5V à une fréquence de 980 Hz. Il faut transformer ce signal puisqu'il faut envoyer une rampe de potentiel à la cellule pour l'expérience de voltammétrie. On utilise un filtre passe-bas pour couper les hautes fréquences à 980 Hz du PWM et obtenir un signal lisse et continu. Afin de pouvoir lisser le signal PWM, le filtre doit ne “percevoir” que le voltage moyen du PWM, comme tout autre élément électronique.

Il faut pour cela que la période de chaque cycle du PWM soit confortablement plus petite que la constante de temps du filtre, qui se calcule ainsi: $R \times C$. Avec une fréquence de 980Hz, la période du PWM est d'environ 1,02 ms. Il faut également que la fréquence de coupure du filtre soit beaucoup plus basse que 980 Hz, afin d'éliminer efficacement la fréquence PWM et limiter l'effet ondulatoire causé par la variation rapide entre 0 et 5 V.

La fréquence de coupure se calcule ainsi: $\frac{1}{2\pi RC}$

. Nos valeurs choisies sont de 10K ohms et 470 nF. Cela nous donne une constante de temps de 4,7 ms, considérablement plus grande que la période du PWM, et une fréquence de coupure de 34 Hz, largement plus basse que 980 Hz. Enfin, une dernière chose doit être considérée dans le choix du filtre. On sait que la constante du temps du filtre est le temps que le filtre met pour que sa sortie atteigne environ 63% de la nouvelle valeur de tension après un saut de voltage. Arduino envoie 256 valeurs différentes pour faire sa plage de 0 à 5V, le filtre doit donc être assez rapide pour se charger à chacun de ces pas. La période nécessaire pour passer de 10% à 90% de la nouvelle valeur du PWM est ce qu'on appelle le temps de montée du filtre, et puisque $\ln(0,9/0,1) = 2,197$, le temps de montée équivaut à environ 2,2 fois la constante de temps. Il équivaut donc à environ 10 ms dans notre cas. En connaissant la plage de voltage désirée, ainsi que le fait que le signal PWM consiste en 256 échelons, on pourra calculer la vitesse de balayage maximale de notre circuit en fonction du temps de montée. Bref, le choix de R1 et de C1 est donc ultimement un compromis entre la vitesse de balayage possible et la qualité du filtrage du signal. Ainsi, une fréquence initiale de 980Hz est préférable à 490 Hz puisqu'elle permet un jeu supplémentaire au niveau de la fréquence de coupure du filtre et de la période du cycle PWM.

Ampli suiveur 1. Le premier amplificateur opérationnel fait office de circuit suiveur et ne modifie pas la tension. La grande impédance d'entrée et faible impédance de sortie font en sorte que le reste du circuit n'affecte pas la tension du filtre. Ses fonctions sont principalement d'isoler les étages du circuit et de stabiliser le signal.

Ampli sommateur, filtre 2 (C2 R4) et -5V. L'amplificateur sommateur, aidée de l'alimentation de -5V et comportant un deuxième filtre RC, a pour fonction de moduler la plage de voltage envoyée par l'étage précédent du circuit. Les expériences de voltammétrie nécessitent un potentiel alternant entre une valeur négative et positive, or l'Arduino ne peut qu'envoyer une plage de 0 à 5V. De plus, le potentiel créé dans une solution aqueuse sur une électrode dépasse rarement 1.5V. On désire donc déterminer des valeurs de résistances et une valeur de tension supplémentaire pour passer d'une plage de 0 à 5 V à une plage de -1 à 1 V. On envoie un courant négatif dans l'étage grâce à une tension de -5V en parallèle au courant sortant du filtre précédent, afin que le courant résultant, soit l'addition des deux, varie entre une valeur négative et positive. Puisque le courant passant par R2 varie entre 0

et $5V/R2$ et que le courant passant par R3 est constant à $-5V/R3$, la valeur de R2 doit être deux fois plus petite que celle de R3 pour que le courant résultant ait une valeur minimale de même norme que sa valeur maximale (À titre d'exemple: $-5 + 0 = -5$: valeur minimale. $-5 + 10 = 5$: valeur maximale). Par les propriétés de l'ampli-op, le courant est ensuite obligé de passer par R4, et le voltage à l'entrée est nul: la loi d'ohm permet donc de calculer facilement le voltage à la sortie de l'étage grâce à la formule suivante:

$$V_o = -R_4 \left(\frac{V_{in}}{R_2} + \frac{V_{-5}}{R_3} \right)$$

On voit donc que pour passer de 0-5V à -1 à 1V, il faut donc également que R4 soit cinq fois plus petite que R3. Enfin, pour ce qui est du filtre RC, le condensateur C2 agit comme filtre passe-bas avec la R4, afin d'éliminer des oscillations parasites et de stabiliser le signal. Puisque $C2 = 100nF$ et que $R4 = 200$ ohms, la fréquence de coupure est d'environ 7958 Hz et n'interfère pas du tout avec le signal principal.

Ampli suiveur 2. Le potentiel passe ensuite au travers d'un deuxième ampli-op suiveur avant de se rendre aux électrodes, ce qui ne modifie pas la tension, mais comme pour le premier ampli-op suiveur, isole le potentiostat des étages précédents du circuit et stabilise le signal.

Le signal qui arrive aux électrodes est donc maintenant une rampe lisse variant de -1V à 1V.

3.3. Traitement du courant en sortie de la cellule électrochimique

Le courant en sortie de la cellule électrochimique est un courant faradique généré par la réaction d'oxydo-réduction entre la solution et l'électrode de travail. Ce courant se situera entre -200 uA et 200 uA. Ainsi, la seconde partie du circuit devra être en mesure de traiter ce courant en assurant les fonctions suivantes :

1. Convertir les mesures de courant faradique en des mesures de tension
2. Assurer un gain de tension conforme aux contraintes de l'Arduino
3. Stabiliser le signal

Afin de mesurer des données expérimentales, nous utilisons les broches analogiques 10 bits de l'Arduino comme entrées analogiques. Ces broches ont une résolution de 4.88 mV et peuvent lire des valeurs de tension allant de **0V à 5V**. Ces deux bornes de tension seront utilisées comme conditions limites nous permettant de transformer le courant faradique en tension (voir section 3.2.2).

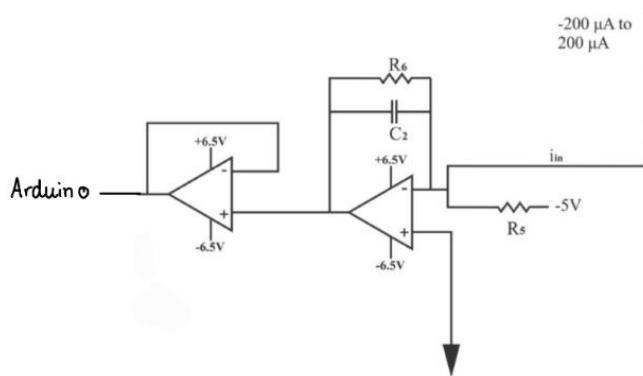


Figure 3.3.1. Schéma du circuit de traitement du courant [2]

Le circuit de traitement de la Figure 3.1.3.1. permet d'adapter le signal original en provenance de la cellule électrochimique à l'Arduino utilisé pour l'enregistrement des données. L'Arduino ne peut capter que des tensions de 0V à 5V, la première partie du circuit de traitement permet de transformer ce courant d'une gamme de courant de **-200 uA à 200 uA** à une gamme de tension de **0V à 5V**.

a) Borne inférieure de 0V

Lorsque le courant faradique sera à son minimum de -200 uA, le circuit de traitement doit le transformer en une source de tension de **0V**. La résistance **R5** avec un potentiel d'entrée de **-5V** permet de fournir un courant connu venant interagir avec le courant faradique expérimentale. Pour avoir une tension de **0V** à la sortie du premier amplificateur, le courant à la borne négative doit être nulle lorsque le courant faradique atteint -200 uA. En choisissant la bonne valeur de **R5**, un courant opposé mais égal en amplitude de 200 uA est fourni à l'intersection des branches. Ce courant constant permet d'avoir un courant net à la borne négative de l'amplificateur de 0 A pour la plus basse valeur de courant faradique. Ceci se traduit en une tension de **0V** à la sortie de l'amplificateur.

b) Borne supérieure de 5V

Lorsque le courant faradique atteint un maximum de 200 uA, le circuit de traitement doit le transformer en une source de tension de **5V**. L'amplificateur inverseur RC permet d'effectuer une transformation du courant en un gain de tension. En utilisant des valeurs précises pour **R6** et **C2**, nous pouvons obtenir un gain adéquat du courant vers la borne supérieure de tension. Le circuit RC agit aussi comme un filtre passe-bas, permettant d'éliminer l'interférence 60 Hz du secteur électrique tout en gardant le signal du courant faradique de basse fréquence.

L'amplificateur suiveur qui suit permet de stabiliser le signal en tension obtenu à la sortie de l'amplificateur inverseur RC.

3.4. Calculs des valeurs pour les composantes du circuit

3.4.1. Circuit de génération du signal

Le choix des valeurs pour **R1** (10K ohms) et **C1** (470 nF) a été justifié dans la section 3.3, ainsi que pour **C2** (100 nF).

Les valeurs de **R2 (500 ohms)**, **R3 (1K ohms)** et **R4 (200 ohms)** peuvent être mises dans l'équation de l'amplificateur sommateur expliquée plus haut pour tester leur validité.

$$V_o = -R_4 \left(\frac{V_{in}}{R_2} + \frac{V_{-5}}{R_3} \right) = -200 \left(\frac{V_{in}}{500} + \frac{V_{-5}}{1000} \right)$$

Si **Vin** = 0V, **V0** = 1V. Si **Vin** = 5V, **V0** = -1V. On a donc bien la plage de potentiel désirée.

3.4.2. Circuit de traitement du signal

La valeur de la résistance **R5** a été obtenue en appliquant une loi d'Ohm pour fournir un courant de **200uA** à travers la résistance.

$$\begin{aligned}\Delta V &= Vin - Vout = R_5 \cdot I \\ \Delta V &= 0V - (-5V) = R_5 \cdot (200 \cdot 10^{-6} A) \\ R_5 &= \frac{5V}{200 \cdot 10^{-6} A} = 25 k\Omega\end{aligned}$$

Les valeurs de **R6** et **C2** ont été déterminés selon le contexte présenté à la section 3.1.3. en transformation le circuit parallèle en impédance et en appliquant une analyse de gain dans le domaine fréquentiel.

i. Transformation du circuit en impédances :

$$\begin{aligned}R_6 &\Rightarrow R_6 \\ C_2 &\Rightarrow \frac{1}{Cj\omega}\end{aligned}$$

ii. Transformation du circuit parallèle en un circuit d'impédance équivalente :

$$R_6 // C_2 = \left(\frac{1}{R_6} + Cj\omega \right)^{-1}$$

$$Z_{eq} = R_6 // C_2 = \frac{R_6}{1 + R_6 C j\omega}$$

iii. Application de la loi d'Ohm pour trouver la fonction de transfert :

$$\Delta V = V_{out} - V_{in} = Z_{eq} \cdot I$$

$$\Delta V = V_{out} - 0V = Z_{eq} \cdot I$$

$$V_{out} = Z_{eq} \cdot I = \frac{R_6}{1 + R_6 C j\omega} \cdot I$$

$$\frac{V}{I} = \frac{R_6}{1 + R_6 C j\omega}$$

iv. Gain de courant-tension nécessaire selon les conditions du circuit :

Objectif : $V = 5V$

Courant d'entrée : $I = 200\mu A + 200\mu A = 400\mu A$

Gain requis :

$$\frac{V}{I} = \frac{5V}{400 \cdot 10^{-6} A} = 12500 = \frac{R_6}{1 + R_6 C j\omega}$$

v. Valeurs de R_6 et C_2 :

Pour déterminer l'effet de R_6 et C_2 sur le gain du circuit, nous devons analyser l'amplitude de la fonction de transfert.

$$\left| \frac{R_6}{1 + R_6 C_2 j\omega} \right| = \frac{R_6}{\sqrt{1 + R_6^2 \cdot C_2^2 \cdot \omega^2}} = 12500$$

Cette fonction de transfert est telle que le facteur dominant de l'équation est la résistance R_6 . Si nous désirons avoir un gain de 12500, nous devons utiliser une résistance R_6 de 12500Ω . Nous posons cette valeur de résistance afin d'isoler la valeur de C_2 . La fréquence $\omega = 2\pi f$ du courant entrant dans l'amplificateur est pris en référence à la fréquence maximale de stimulation de la solution par voltanométrie, soit de 0.5 V/s , donnant ainsi

une période de 8 secondes et une fréquence de 0.125 Hz. Suite à l'isolation de C_2 , nous trouvons la valeur suivante :

$$C_2 = 100 \text{ nF}$$

3.5. Temps caractéristique et réponse temporelle du circuit

Tel que mentionné plus haut, la vitesse de balayage maximale du signal est limitée par la constante de temps du premier filtre RC. Dans notre cas, avec une valeur de 10K ohms et 470 nF, la constante de temps, et donc le temps caractéristique, est de 4,7 ms. Afin d'avoir une réponse temporelle décente, on veut toutefois que le condensateur puisse se rendre à au moins 90% de la nouvelle tension à chaque incrément du PWM avant qu'il se rende au suivant. On a déterminé plus tot que le temps de montée correspondait à environ 10 ms. Considérant que notre plage de voltage envoyée est de 5 Volts et que le PWM la divise en 256 incréments, on se retrouve avec des incréments d'environ 0,0195 V et la possibilité d'en filtrer 100 par seconde, ce qui donne une vitesse de balayage maximale de 1,953 V/s. Toutefois, en pratique, on veut que le signal soit le plus lisse possible (dans la limite du raisonnable), et non seulement acceptable. Pour atteindre 99% de la nouvelle tension après un saut de voltage, on sait qu'un filtre RC a besoin de 5 fois sa constante de temps. Si on utilise cette valeur, on obtient une vitesse de balayage maximale d'environ 0,83 V/s. Pour les tests de voltammétrie, la vitesse de balayage prescrite était de 0,1 V/s. On conclue donc que notre vitesse de balayage permet amplement d'obtenir une bonne réponse temporelle.

3.6. Mesure de protection de la plateforme

L'Arduino ne peut prendre en entrée que des tensions allant de 0V à 5V. Si la tension dépasse 5V, la plateforme risque une surtension et une rupture du fonctionnement de sa circuiterie. Pour prévenir l'entrée d'un courant de tension trop élevé suite au gain de l'amplificateur, nous utiliserons une diode qui s'active dès que la tension en entrée est plus grande que 5V. La diode sera liée à une LED qui s'allumera lorsqu'un courant passe à travers son circuit.

Une diode laisse passer un courant en fonction de la différence de tensions entre sa borne d'entrée et de sortie. Si la tension à sa borne de sortie dépasse la tension à sa borne d'entrée,

la diode laissera passer le courant dans le sens inverse à celui permis par défaut. Le sens d'orientation de la diode est crucial pour le bon fonctionnement du système.

Ainsi, la limite de la diode sera placée à 5 V. Si la tension en sortie dépasse cette valeur, la diode laissera passer le courant qui allumera la LED, signifiant une surcharge du circuit. Pour protéger la diode et la LED, nous allons limiter le courant passant à travers celles-ci en mettant une résistance de $200\text{ k}\Omega$ lié à la masse.

3.7. Photo du circuit complet sur PCB

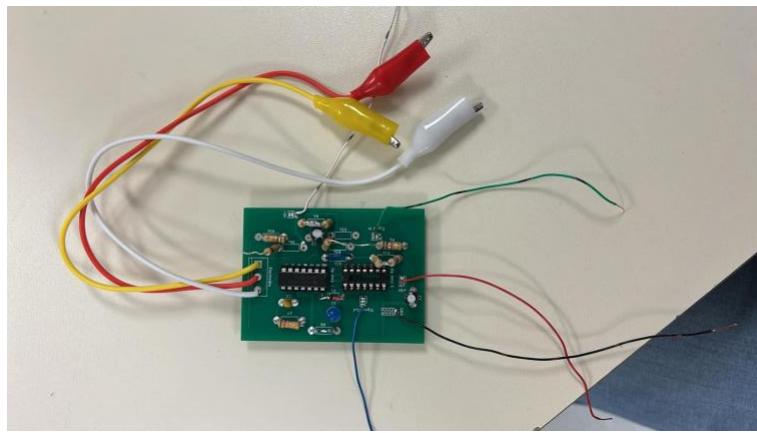


Figure 3.7.1 Photo du circuit

4) Programmation

La programmation du potentiostat constitue une étape essentielle du projet, assurant le contrôle précis des protocoles électrochimiques ainsi qu'une acquisition fiable des données dans les unités appropriées. Celle-ci a été réalisée en langage C++ à l'aide de l'interface Arduino IDE. Le code développé remplit plusieurs fonctions clés : il permet à l'utilisateur de sélectionner la méthode électrochimique désirée (chronampérométrie ou voltammetrie cyclique) et de saisir les paramètres expérimentaux correspondants, soit E1, E2 et t0 pour la chronampérométrie, ou E1, E2, la vitesse de balayage et le nombre de cycles pour la voltammetrie. Ces paramètres sont stockés dans des variables globales permettant de les réutiliser. Le programme génère ensuite le signal approprié (créneau ou triangle) à l'aide du PWM et transmet en temps réel, via le port série, les valeurs du temps écoulé, du potentiel appliqué et du courant mesuré. Les données de sortie sont organisées en un tableau à trois colonnes, facilement exploitable, notamment pour une intégration directe dans le tableur PuTTY utilisé. Enfin, comme l'Arduino ne peut gérer que des tensions comprises entre 0 et 5 V, des conversions ont été effectuées par produit croisé afin d'afficher les valeurs réelles de tension et de courant appliqués dans la cellule électrochimique :

$$[0V, 5V]_{sortie \text{ } PWM} \rightarrow [-1V, 1V]_{solution \text{ } triangl.}$$

$$[0V, 5V]_{entrée \text{ } analogique} \rightarrow [-200\mu A, 200\mu A]_{solution}$$

À l'annexe 2, on retrouve le schéma (réalisé à la semaine du 10mars) de l'organisation du code et des fonctions qui a été réalisé lors de rencontres.

4.1. Organisation de la communication serielle

La communication serielle constitue l'interface centrale entre l'utilisateur et le système. Elle est initialisée avec `Serial.begin(9600);` dans la fonction `setup()`, permettant une communication bidirectionnelle via le port USB, compatible avec l'IDE Arduino ou un terminal externe (PuTTY). L'utilisateur interagit avec le système uniquement par saisies textuelles, ce qui permet de configurer entièrement la méthode sans avoir à modifier le code source. Les valeurs saisies par l'utilisateur sont lues à l'aide de la fonction `Serial.readStringUntil('\n')`, qui permet de capter les instructions ligne par ligne, tandis que les messages et les données sont envoyés vers le moniteur série à l'aide de la fonction `Serial.println()`, utilisée à la fois pour afficher les instructions, les paramètres saisis et les résultats expérimentaux sous forme de tableau.

La logique de saisie est structurée par une machine d'état, gérée par la variable globale « `etape` », qui permet de guider l'utilisateur à travers chaque étape de configuration dans

une interface lucide avec un code très simple à comprendre. Cette structure est implantée à l'intérieur d'un switch(etape) dans la boucle loop(). Voici le rôle de chaque case :

- case 0 : L'utilisateur est invité à choisir la méthode électrochimique (taper « 1 » pour la chronoampérométrie, ou « 2 » pour la voltammetrie cyclique). Cette sélection est traitée par la fonction handleMethodSelection() qui met à jour la variable methode et passe à l'étape suivante.
- case 1 : L'utilisateur saisit la première tension (E1, en volts).
- case 2 : Saisie de la seconde tension (E2). Le cheminement diverge (if else) ensuite selon la méthode sélectionnée. Si la méthode est chronoampérométrie, on passe à l'étape 3. Si c'est voltammetrie cyclique, on passe directement à l'étape 4.
- case 3 : Pour la chronoampérométrie, l'utilisateur entre la durée t0 (temps à E1 avant de passer à E2). Une fois les paramètres saisis, la fonction afficherParametres() est appelée pour récapituler les valeurs, et on passe à l'étape 6.
- case 4 : Pour la voltammetrie cyclique, on demande à l'utilisateur la vitesse de balayage en V/s.
- case 5 : L'utilisateur saisit le **nombre de cycles**. Ensuite, les paramètres sont affichés, et le programme passe également à l'étape 6.
- case 6 : L'utilisateur est invité à entrer un **facteur d'affichage** (par défaut : 1). Cette valeur permet de contrôler la fréquence à laquelle les données sont imprimées sur le port série, ce qui pourrait permettre de limiter la surcharge de transmission. Une valeur de 4, par exemple, signifie qu'un point sur 4 sera affiché. Une fois la saisie confirmée et pendant que le système affiche les paramètres, le code attend la commande "ok" (while (true){if (entree == "ok") {break;}}) pour démarrer la mesure.

Une fois "ok" saisi, un message "Début" est affiché, et le programme exécute la fonction de mesure correspondante (activée via les booléens chrono_aff ou volto_aff). À chaque itération de mesure, les données sont affichées sous forme de tableau à trois colonnes : temps (s), tension (mV), courant (μ A). Il est important de noter qu'en écrivant **Again** dans le port série après l'acquisition complétée, il est possible de relancer une nouvelle série de données. Cela est rendu possible grâce à la fonction resetProgramme(), qui réinitialise les paramètres initiaux et remet la variable de la machine d'état à 0. Ainsi, l'ensemble du processus peut être contrôlé via le port série.

4.2. Envoi de différents types de signaux modulables à partir de l'Uno

Chaque signal est produit à l'aide de la modulation de largeur d'impulsion (PWM) sur la broche 9 (pwmPin), contrôlée par le Timer1 configuré en mode Fast PWM 10 bits (décris à la question 4.3.), permettant une tension de sortie analogique simulée avec une résolution de 1024 niveaux pour permettre un meilleur contrôle du potentiel appliqué à la cellule électrochimique (et lissage).

→ Signal triangulaire : Avec les paramètres entrée dans le moniteur série, la fonction voltammetrieFunction() va générer un **signal triangulaire** :

Dans le programme, la fonction voltammetrieFunction() génère un signal triangulaire en faisant varier progressivement la valeur du PWM 10 bits, à l'aide de la fonction analogWrite10bit(value). La valeur value est incrémentée ou décrémentée pas à pas dans deux boucles for, représentant la montée et la descente du triangle. Chaque modification du PWM correspond à un changement de duty cycle, et la fonction delay(interval) contrôle le temps entre chaque pas selon la vitesse de balayage. Ce signal PWM est ensuite transformé en tension analogique triangulaire par le circuit du filtre passe-bas RC externe qui agit comme un intégrateur (Duty 100% → 5V et Duty 0% → 0V). Si le duty varie linéaire, alors le triangle aura également des pentes linéaires

Le pas de tension correspond à l'incrément minimal entre deux niveaux successifs du signal triangulaire. Puisque le PWM est en 10 bits, il dispose de 1024 niveaux (de 0 à 1023) :

$$\text{pas_mV} = \frac{|E_2 - E_1|}{1023} \Rightarrow \text{pour } E_1 = -1000\text{mV et } E_2 = 1000\text{mV : pas_mV} \cong 2 \text{ mV/bit}$$

Ce calcul permet de déterminer avec quelle précision la tension sera augmentée ou diminuée à chaque itération du triangle. Un plus petit pas permet une meilleure résolution du signal appliqué. Ainsi, on peut exploiter pleinement le 10 bits disponible en envoyé une valeur de tension par plateau pour le signal d'amplitude maximale (-1 à 1V).

Le délai entre chaque pas est appelé par la variable interval. Pour respecter la vitesse de balayage définie par l'utilisateur, on calcule le temps nécessaire entre deux pas en utilisant (diviser l'incrément en y par la pente) :

$$\text{interval (ms)} = \frac{\text{pas_mV}}{\text{vitesse en mV/s}} \times 1000$$

Ainsi, plus la vitesse est élevée, plus la pente est importante et plus l'intervalle entre les pas est court.

Détermine les valeurs de PWM associées à E1 et E2 : Étant donné que le signal doit osciller entre -1 V et +1 V, mais que l'Arduino sort du 0 à 5 V, on applique une transformation linéaire pour faire correspondre l'affichage. En utilisant les valeurs entrées dans la machine état du loop on trouve les bornes supérieur et inférieur des boucles for. La transformation linéaire est la suivante :

$$pwm = \frac{-E_x + 1000}{pas_{mV}}$$

Ensuite il ne suffit que de faire deux boucle for (une pour la partie croissante et une pour la partie décroissante) le tout dans un while qui a comme condition que le compteur de tour soit inférieur au nombre de cycle entrée en paramètre dans le sériele. À l'intérieur de chaque boucle for, on envoie la tension par la fonction analogWrite10bit(value) dans la pin9, on affiche la ligne associé du tableau et on met le délai « interval » pour prendre une mesure par marche.

```
for (value = pwmMin; value <= pwmMax; value++) { ... } // Descente
for (value = pwmMax; value >= pwmMin; value--) { ... } // Montée
```

→ Signal carré :Avec les paramètres entrés dans le moniteur série, la fonction chronoamperometrieFunction() génère un signal carré (un seul créneau) composé de deux niveaux de tension définis par l'utilisateur : E₁ et E₂. Ces niveaux sont maintenus pendant des durées précises, simulant ainsi un signal de type créneau, typique des expériences de chronoampérométrie.

La tension est appliquée sous forme d'un signal PWM 10 bits, puis convertie en tension analogique à l'aide d'un filtre RC. Les valeurs PWM correspondant à E₁ et E₂ sont calculées à partir d'une transformation linéaire expliquée précédemment.

Le temps entre chaque acquisition est contrôlé par une fréquence d'échantillonnage définie dans le code et convertie en intervalle de temps :

$$intervalle = \frac{1000}{f_{échantillonnage}} \quad (\text{en ms})$$

Dans le code, le signal carré est généré en deux phases distinctes.

Phase 1 — Application de E₁ : La première phase consiste à maintenir la tension E₁ pendant une durée t₀ spécifiée par l'utilisateur. Cela est réalisé à l'aide d'une boucle while (temps < t₀). À chaque itération, la tension E₁ est appliquée à l'aide de la fonction analogWrite10bit(pwmE1), qui génère un signal PWM correspondant. Le temps est incrémenté selon une fréquence d'échantillonnage prédéfinie (ici 34,48 Hz), soit un

intervalle d'environ 29 ms. À chaque point, la tension appliquée (recalculée à partir du PWM), le temps et le courant mesuré (getPWMCURRENT()) sont envoyés via le port série.

Phase 2 — Application de E₂ : La deuxième phase applique la tension E₂ pendant une durée fixe de 5 secondes. Elle est implémentée par une boucle for basée sur la fréquence d'échantillonnage :for (int i = 0; i < (5 * frequencyHz_ech); i++) { ... }. À chaque itération, la tension E₂ est appliquée avec analogWrite10bit(pwmE2), et les valeurs de temps, tension et courant sont enregistrées de la même manière que dans la phase précédente.

4.3. Gestion de la mesure simultanée du courant (échantillonnage)

Pour assurer un suivi temporel précis du courant mesuré en réponse au signal de tension appliquée, le code utilise une fonction d'échantillonnage temps_actuel(float frequencyHz_ech) synchronisée avec une fréquence calculée ou fixée. Chaque mesure est déclenchée uniquement si l'intervalle correspondant est écoulé, garantissant un espacement temporel constant entre les points. La fonction creationTableau() orchestre ensuite l'acquisition du courant via l'ADC (getPWMCURRENT()), le calcul de la tension PWM à partir de la valeur globale « value », et l'impression des données. Cette approche assure que chaque point de mesure est associé à un instant précis du protocole expérimental, permettant une interprétation fiable et synchrone des résultats. Un schéma montrant la gestion des différentes fonctions est présenté à l'annexe 1.

La fréquence d'échantillonnage est définie par la variable float frequencyHz_ech et utilisée dans la fonction de gestion du temps (temps_actuel). La fonction gestion de temps utilise l'horloge du arduino par millis(). Cette fonction retourne -1 lorsqu'il n'est pas encore temps d'échantillonner, ce qui est ignoré par la condition if (temp > 0) { ... } dans le code principal. Lorsqu'il est temps de prendre une mesure, elle retourne la valeur temporelle courante. En voltamétrie cyclique, cette fréquence est calculée dynamiquement à partir du pas de tension, lui-même déterminé par les paramètres entrés par l'utilisateur (E₁, E₂, et vitesse), comme expliqué à la question précédente. En revanche, pour la chronoampérométrie, la fréquence est fixée dans le code à une valeur constante de 34 Hz.

```
Serial.print(temp, 4);
Serial.print(";\t");
//PWM=0 ->+1000mV PWM=1023 ->-1000mV
float tension_affichee = -pas_mV * value + 1000;
Serial.print(tension_affichee, 2);
Serial.print(";\t");
Serial.println(getPWMCURRENT(), 2);
```

Figure 4.3.1 : code de l'affichage

Le code permet d'afficher les valeurs mesurées à chaque cycle d'échantillonnage, ou de réduire la fréquence d'affichage par un facteur ajustable (facteur_affichage, par exemple 1 ou 4). Ce mécanisme a été initialement mis en place pour éviter une surcharge de transmission série, tout en permettant une visualisation fluide des données en temps réel en contexte expérimental. Lors de la journée des tests chimiques, un facteur de réduction de 4 a été utilisé, car il permettait d'obtenir les mêmes résultats tout en améliorant la lisibilité. Il est important de noter que le facteur d'affichage n'a aucun impact sur la génération du signal PWM qui est toujours généré à 1 valeur par marche.

Ainsi, l'acquisition du courant est parfaitement synchrone avec le signal de stimulation, permettant une reconstitution fidèle des courbes courant-potentiel.

4.4. Affichage des données courant – potentiel et discussion sur leur résolution

Les données courant-potentiel sont affichées en temps réel sur le moniteur série de l'ordinateur comme à la figure 4.3.2 , sous forme tabulaire, avec trois colonnes : temps (en secondes), tension appliquée (en millivolts) et courant mesuré (en microampères). Pour ce faire on utilise la fonction génération de tableau expliqué à la question 4.3. La tension est générée à l'aide d'un signal PWM sur 10 bits (via le Timer1 de l'Arduino), tandis que le courant est mesuré par lecture analogique avec l'ADC 10 bits intégré au microcontrôleur.

Résolution théorique de la tension

La modulation PWM couvre une plage de -1 V à $+1$ V (la plus grande amplitude pour évaluer la « pire » résolution), soit une amplitude totale de 2 V répartie sur 1023 intervalles (car 10 bits $\rightarrow 2^{10} = 1024$ niveaux, donc 1023 pas entre eux). La résolution théorique est donc donnée par :

$$\text{Résolution}_{\text{tension}} = \frac{2 \text{ V}}{1023} \approx 1,96 \text{ mV}$$

Résolution théorique du courant

La conversion du courant dans le code est effectuée selon la relation :

$$I (\mu\text{A}) = \text{ADC} \times \left(\frac{400}{1023} \right) - 200$$

Temps (s) tension (mV) courant (mA)
 0.0196;1000.00;-200.00
 0.0391;998.04;-200.00
 0.0587;996.09;-200.00
0.0782;994.13;-200.00
 0.0978;992.18;-200.00
 0.1173;990.22;-200.00
 0.1369;988.27;-200.00
0.1564;986.31;-200.00
 0.1760;984.36;-199.22

Temps (s);tension (mV);courant (uA)
0.0782;994.13;44.39
0.1564;986.31;44.00
 0.2346;978.49;43.61
 0.3128;970.67;42.83
 0.3910;962.85;41.66
 0.4692;955.03;41.27

Figure 4.3.2 affichage pour différentes fréquences

Ce qui suppose une gamme de mesure allant de $-200 \mu\text{A}$ à $+200 \mu\text{A}$, soit une amplitude totale de $400 \mu\text{A}$. La résolution du courant devient alors :

$$\text{Résolution}_{\text{courant}} = \frac{400 \mu\text{A}}{1023} \approx 0,40 \mu\text{A}$$

Même si les résolutions obtenues théoriquement sont relativement fines ($\approx 2 \text{ mV}$ pour la tension et $\approx 0.40 \mu\text{A}$ pour le courant), elles sont difficiles à atteindre en conditions expérimentales. En effet, plusieurs sources d'erreur réduisent la précision réelle du système :

- le bruit de fond sur la ligne de mesure (alimentations, perturbations électromagnétiques),
- les imprécisions des composants analogiques (filtrage RC, résistances),
- la quantification et le bruit thermique de l'ADC,
- l'ondulation du signal PWM malgré le filtrage,
- les limitations liées au taux d'échantillonnage ou à la résolution temporelle.

Ainsi, bien que le système permette en théorie une résolution inférieure à 2 mV et à $0.4 \mu\text{A}$, ces valeurs doivent être considérées comme idéales. En pratique, la résolution effective est souvent réduite, et un post-traitement numérique (filtrage, moyennage) pourrait être nécessaire pour améliorer la qualité des mesures. On peut voir ce problème de bruit au graphique de la section 4.5.

4.5. Méthode de test pour valider la précision de l'appareil

Afin de valider la précision de notre code, nous avons réalisé plusieurs tests au fur et à mesure de l'avancement du circuit et de la programmation, selon une démarche séquentielle.

1. Lors de la semaine de relâche, nous avons utilisé Tinkercad pour modéliser la première partie du circuit. Cette simulation nous a permis de visualiser les résultats approximatifs attendus à la sortie de chaque amplificateur opérationnel (grâce aux oscilloscopes intégrés), et de tester notre code dans un environnement virtuel. (Voir annexe 1)
2. À l'aide d'un générateur de fonctions, nous avons appliqué différents signaux (sinusoïdaux, triangulaires, en créneaux) dans l'intervalle de **0 à 5 V** sur l'entrée analogique. L'objectif était de vérifier que la fonction `getVoltage()` permettait de

restituer fidèlement la forme du signal appliqué. Les courbes obtenues via le tracé série de l’Arduino correspondaient aux formes attendues, confirmant la validité de notre méthode de lecture.

3. Une fois la première partie du circuit complétée sur la breadboard, nous avons testé le signal PWM afin de nous assurer qu’il génère bien une onde triangulaire avec l’amplitude et la pente souhaitées. Plusieurs pentes et amplitudes ont été essayées. Les résultats ont été mesurés et compilés dans un tableau comportant trois colonnes : temps (s), tension_théorique et tension_expérimentale. L’objectif était de vérifier que la tension affichée (tension_théorique) correspondait au signal effectivement envoyé à la solution. La lecture de la tension a été effectuée via la broche A0, et les données ont été enregistrées sous forme de fichiers .txt. Ces données ont ensuite été utilisées pour tracer des graphiques. Par exemple, pour une pente de 0,1 V/s et une amplitude de -1 V à 1 V, on obtient une montée en 20 secondes et une période totale de 40 secondes, ce qui est conforme aux attentes (figure 4.5.1). Ce test a également permis de valider la partie transmission série de notre code et d’évaluer qualitativement le bruit du signal. Des graphiques supplémentaires correspondant à d’autres cas de test (amplitude de -0,5 à 0,5 V, pente de 0,5 V/s, etc.) sont présentés en annexe3.

(Voir fichiers texte : test1.txt, test2.txt et test3.txt).

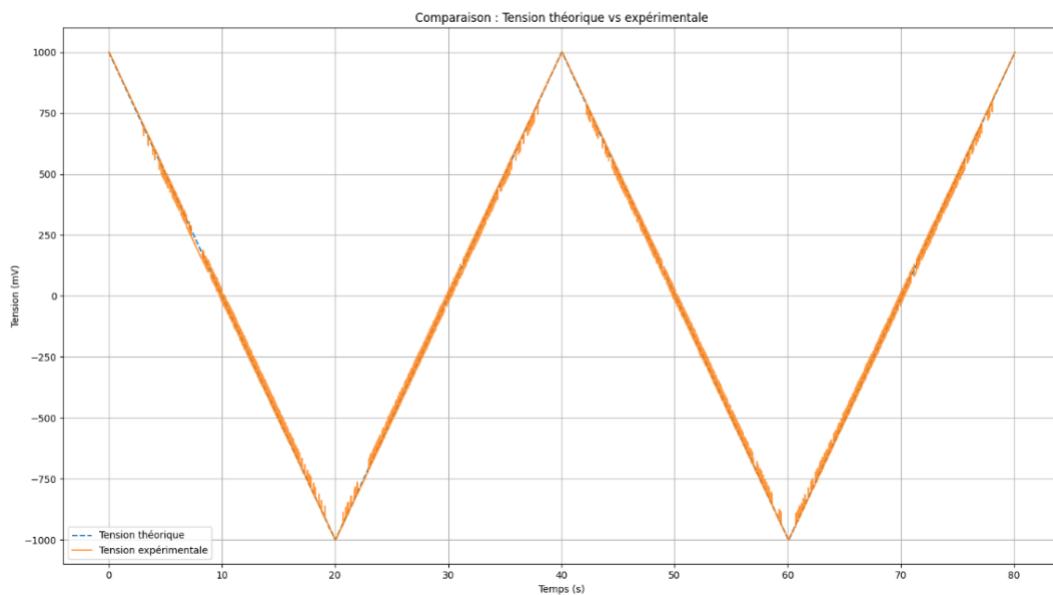


Figure 4.5.1 verification du signal envoyé

Sur le graphique on constate la présence de bruit (noté que c’était le circuit sur breadbord), mais le signal reste très fiable

4. Pour vérifier la deuxième partie du circuit et la lecture, nous avons injecté un signal sinusoïdal allant de -2 V à 2 V aux bornes d'une résistance de $10\text{ k}\Omega$ à l'aide d'un générateur de fonctions. Selon la loi d'Ohm, cela correspond à un courant de $-200\text{ }\mu\text{A}$ à $200\text{ }\mu\text{A}$ (amplitude maximale à mesurer). Nous nous attendions donc à observer sur l'oscilloscope un signal sinusoïdal d'amplitude 0 à 5 V. Le gain a été validé, et la correspondance dans le code — $0 = -200\text{ }\mu\text{A}$ et $1023 = 200\text{ }\mu\text{A}$ — a été établie sans nécessiter d'ajustement supplémentaire. (voir figure 4.5.2)

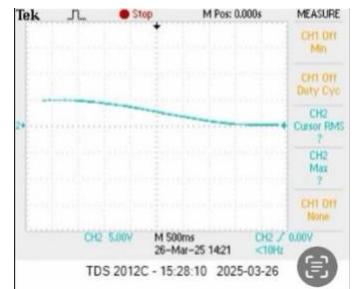


Figure 4.5.2 : lecture sur l'oscilloscope

5. Quelques jours avant le premier test chimique, nous avons procédé à un test global du circuit (assemblé au complet) ainsi que du code. Pour simuler le comportement électrochimique de la solution dans le bêcher, nous avons remplacé celle-ci par une résistance de $5\text{ k}\Omega$ (figure 4.5.3). Une tension triangulaire, variant de -1 V à $+1$ V avec une vitesse de balayage de $0,1$ V/s, a été appliquée entre la sortie du signal (ET) et la masse. Ce montage nous a permis de vérifier le bon fonctionnement de l'ensemble du système, notamment en enregistrant la courbe tension-courant. En traçant la tension en fonction du courant mesuré, nous avons obtenu une droite, ce qui correspond au comportement attendu selon la loi d'Ohm ($V=RI$). En effet, pour une résistance pure, la tension est directement proportionnelle au courant. Ainsi, le graphique V en fonction de I (figure 4.5.4) doit être une droite dont la pente représente la résistance. Cette visualisation a permis de valider expérimentalement la cohérence du signal généré et mesuré. L'objectif du test est double :

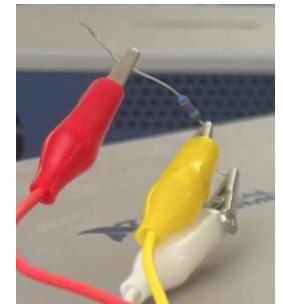


Figure 4.5.3 : modélisation

1. Vérifier la **linéarité** et la **fidélité** du signal de tension appliqué en comparant la tension générée par l'Arduino (via PWM) et celle mesurée à l'entrée de la cellule simulée à l'aide d'un multimètre ou d'un oscilloscope.

2. Évaluer la **précision** de la mesure de courant, en injectant un courant connu dans une résistance de précision et en comparant la valeur mesurée par le système avec la valeur attendue (calculée par la loi d'Ohm).

La pente de la courbe tension-courant est négative car la tension appliquée varie de $+1$ V à -1 V pendant que le courant suit linéairement en sens inverse, selon la loi d'Ohm $V=RI$. L'ordonnée à l'origine provient des imperfections expérimentales (décalage de zéro, bruit, etc.). Idéalement, la droite passerait par l'origine.

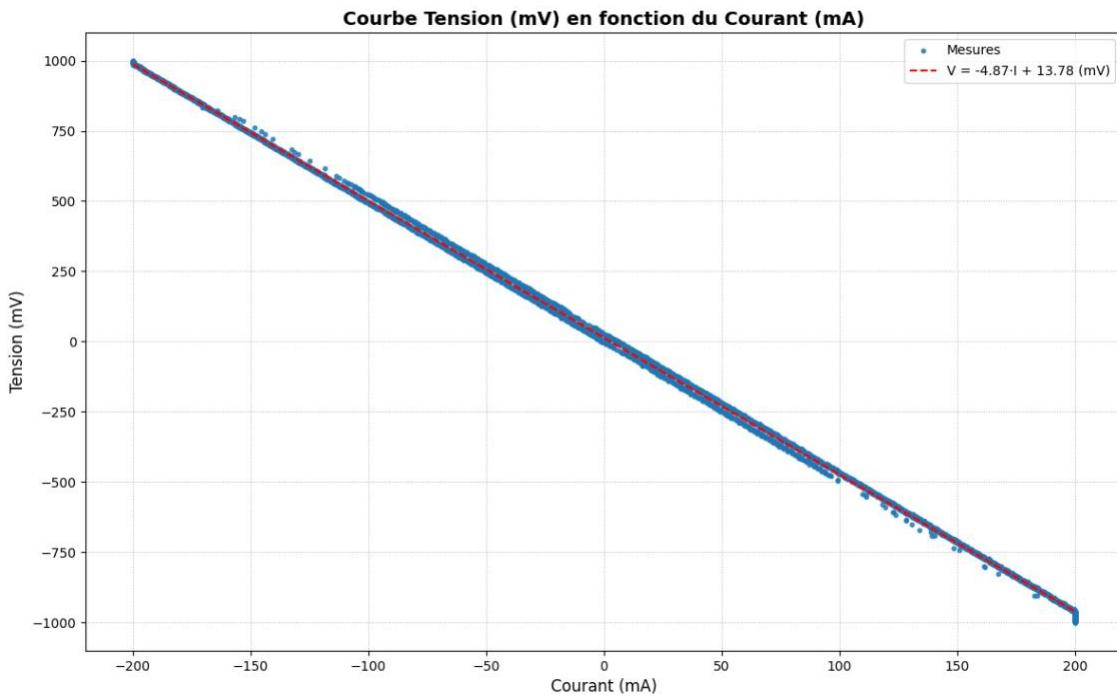


Figure 4.5.4: Vérification de la loi d'ohm par le test global

1. Écart relatif de la pente par rapport à la résistance réelle :

$$\text{Écart relatif (\%)} = \left| \frac{R_{\text{mesurée}} - R_{\text{réelle}}}{R_{\text{réelle}}} \right| \times 100 = \left| \frac{4,87 \text{ k}\Omega - 5,00 \text{ k}\Omega}{5,00 \text{ k}\Omega} \right| \times 100 = 2,60\%$$

La faible valeur d'écart relatif nous permet de confirmer notre circuit et notre code.

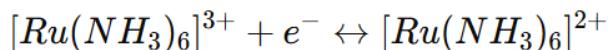
En annexe 4, on retrouve un graphique obtenu en modélisant la cellule par une résistance en parallèle avec un condensateur, afin d'illustrer le comportement capacitif. Cette réponse en boucle traduit une situation où le courant dépend non seulement de la tension instantanée, mais aussi de sa variation temporelle, ce qui est caractéristique des milieux électrochimiques réels. (voir fichiers texte: test4.txt et test5.txt)

5. Mesures chimiques

5.1 Voltammetrie

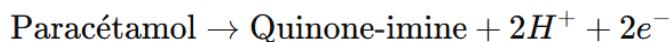
Lorsqu'on analyse une courbe voltampérométrique avec deux pics, il est essentiel d'identifier celui qui correspond à la réaction d'intérêt. Si l'espèce analysée subit une oxydation, on utilise le pic d'oxydation, qui apparaît généralement sur l'aller et se manifeste par un courant positif maximal (pic vers le haut). En revanche, si l'espèce subit une réduction, on s'intéresse au pic de réduction, observé lors du retour, sous la forme d'un courant négatif minimal (pic vers le bas). Le choix du pic dépend donc directement de la nature de la réaction redox que l'on cherche à mesurer.

Les mesures électrochimiques ont été réalisées sur deux espèces d'intérêt biomédical : le ruthénium hexaammine et le paracétamol. Le ruthénium hexaammine subit une réaction de réduction réversible, décrite par l'équation suivante :



Cette réaction implique le transfert d'un électron et se produit typiquement autour de -0,2 V vs Ag|AgCl. Le paracétamol, quant à lui, subit une oxydation irréversible en solution aqueuse, menant à la formation d'un produit de type quinone-imine selon :

Cette réaction se manifeste par un pic d'oxydation aux alentours de +0,9 V vs Ag|AgCl. Ces deux réactions redox servent de base pour l'analyse voltampérométrique et permettent de relier les courants de pic mesurés à la concentration des espèces en solution.



Par une analyse sur Excel de nos courbes de voltammetrie, il a été possible d'extraire la valeur des courants de pic faradiques, correspondant aux réactions redox étudiées. Pour ce faire, le courant capacitif, présent en raison du comportement non-polarisable de l'électrode de travail, a été soustrait du courant total. Ce courant capacitif a été approximé au moment du pic en extrapolant une tangente à la courbe dans la région précédant le pic. La différence entre le courant total au pic et cette valeur extrapolée a permis d'estimer la composante faradique nette, proportionnelle à la concentration de l'espèce en solution.

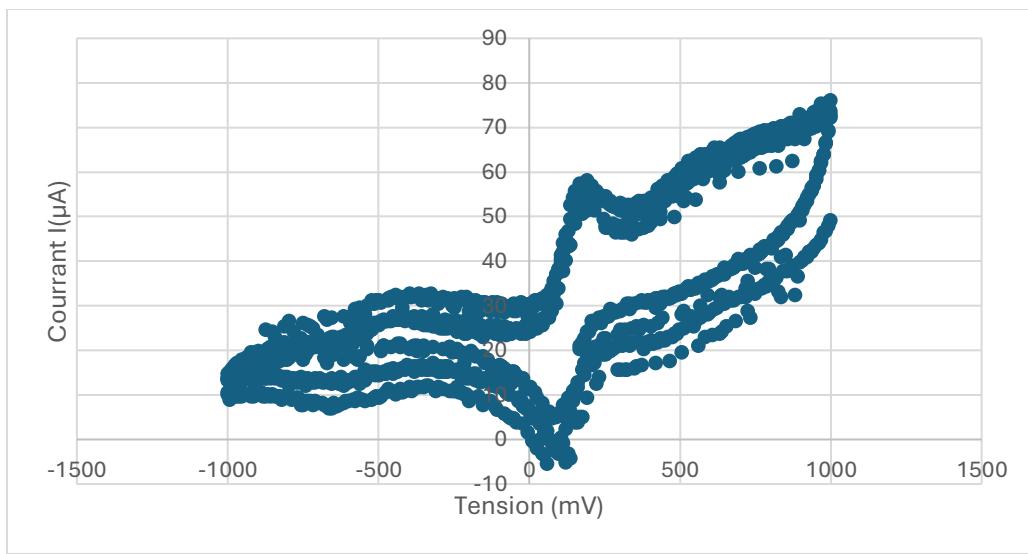


Figure 5.1 Voltamétrie cyclique de $\text{Ru}(\text{NH}_3)_3^{3+}$ 1 mM dans 1 M KCl – 3 cycles à $v = 100$ mV/s

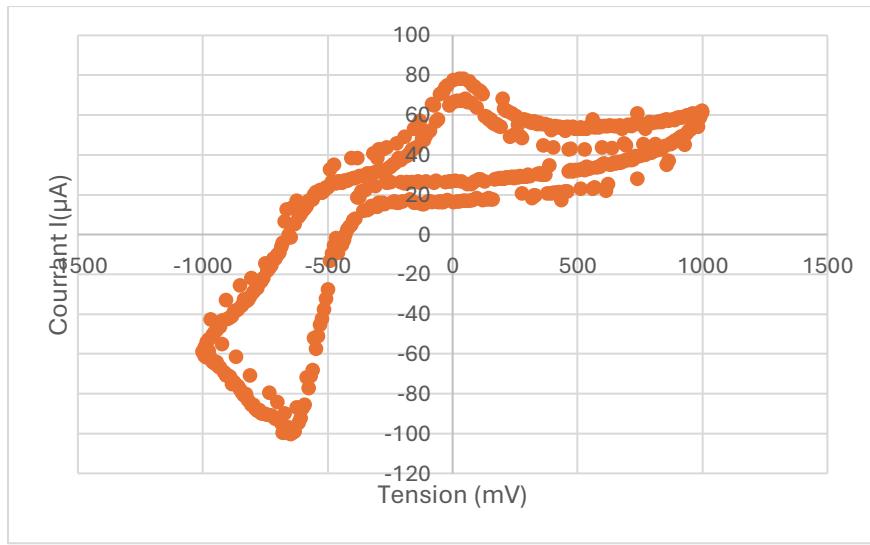


Figure 5.2. Voltamétrie cyclique du paracétamol 1 g/L – 1 cycle à $v = 100$ mV/s

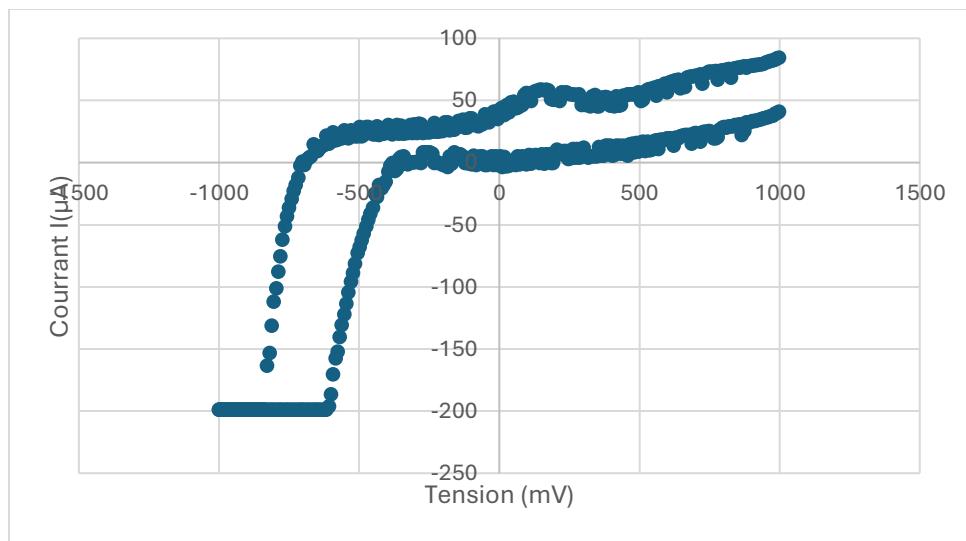


Figure 5.3. Voltamétrie cyclique du paracétamol 5g/L – 1 cycle à $v = 100 \text{ mV/s}$

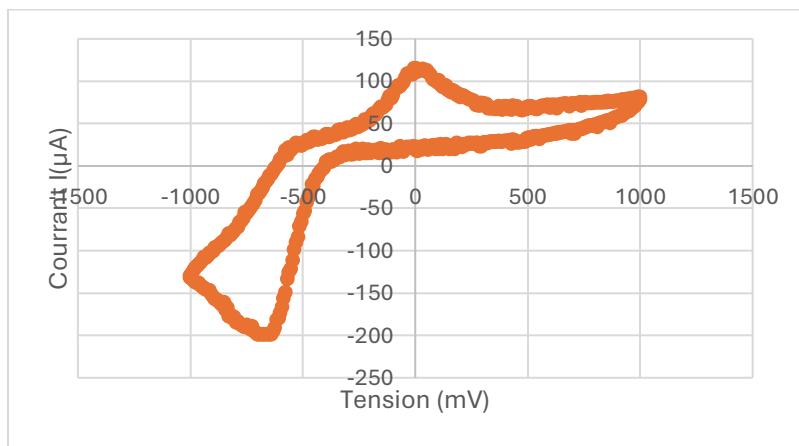
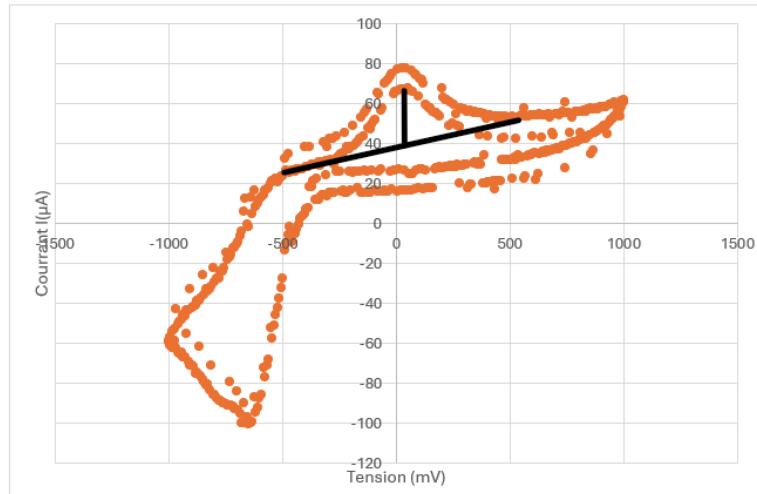


Figure 5.4 Voltamétrie cyclique du paracétamol ?g/L – 1 cycle à $v = 100 \text{ mV/s}$

Ainsi en suivant l'approche précédentes, les tracés suivants on été effectués .



Ce qui permet d'extraire une valeur de pic de 32,84 μA

Figure 5.5. Tracé de Voltamétrie cyclique du paracétamol 1 g/L – 1 cycle à $v = 100 \text{ mV/s}$

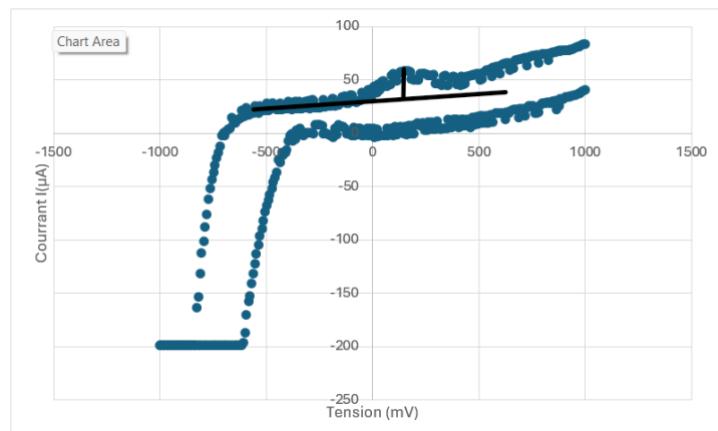


Figure 5.6. Voltamétrie cyclique du paracétamol 5g/L – 1 cycle à $v = 100 \text{ mV/s}$

Ce qui permet d'extraire une valeur de pic de 25,83 μA .

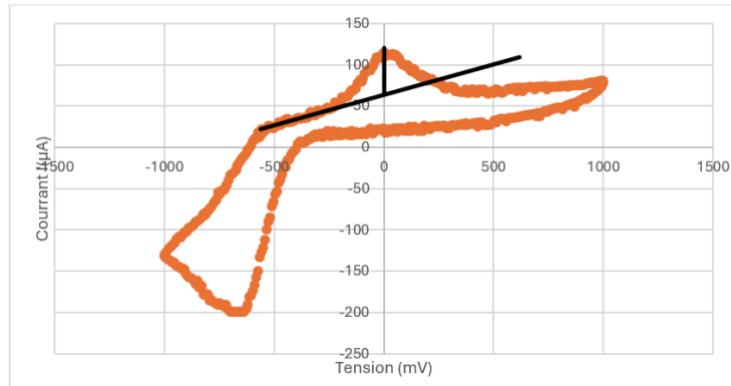


Figure 5.7 Tracé de Voltamétrie cyclique du paracétamol X g/L – 1 cycle à $v = 100 \text{ mV/s}$

Ce qui permet d'extraire une valeur de pic de $45,18 \mu\text{A}$.

Finalement pour le RuHex, les 3 cycles de voltamétrie ont été moyennés pour obtenir le graphique suivant :

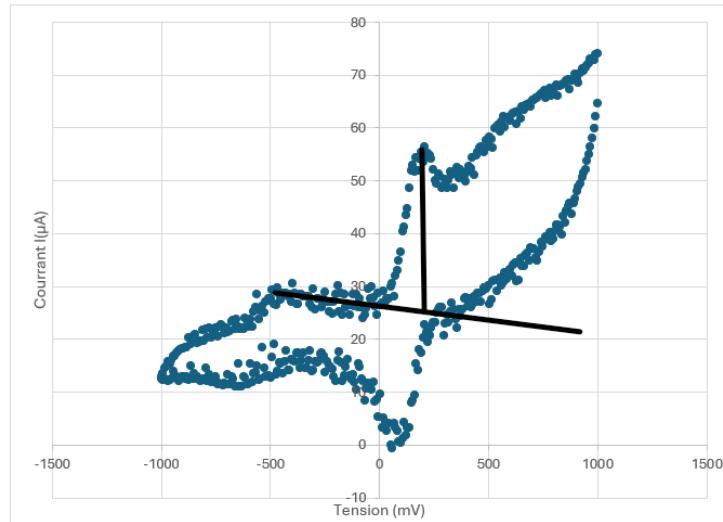


Figure 5.8 Tracé de Voltamétrie cyclique moyennée de $\text{Ru}(\text{NH}_3)_6^{3+}$ 1 mM dans 1 M KCl

Ce qui permet d'extraire une valeur de pic de $32,83 \mu\text{A}$

Pour ce qui est de la solution de paracétamol 5g/L, les données vont à l'encontre de la théorie puisque pour une concentration plus élevée, le pic est plus faible.

Ainsi au lieu de faire une courbe d'étalonnage pour déterminer la concentration de la

solution inconnue, nous avons utilisé la proportionnalité entre la concentration et le courant pour faire le calcul proportionnel suivant.

$$\frac{I_{inconnu}}{I_{référence}} = \frac{C_{inconnu}}{C_{référence}}$$

$$C_{inconnu} = \frac{1g}{L} * \frac{46,18}{32,84} = 1.406 \frac{g}{L}$$

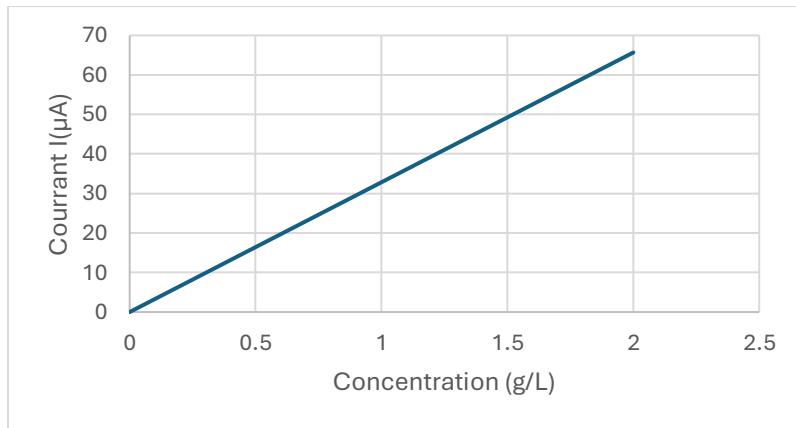


Figure 5.9 Courbe d'étalonnage du paracétamol

5.2 Chronamperométrie

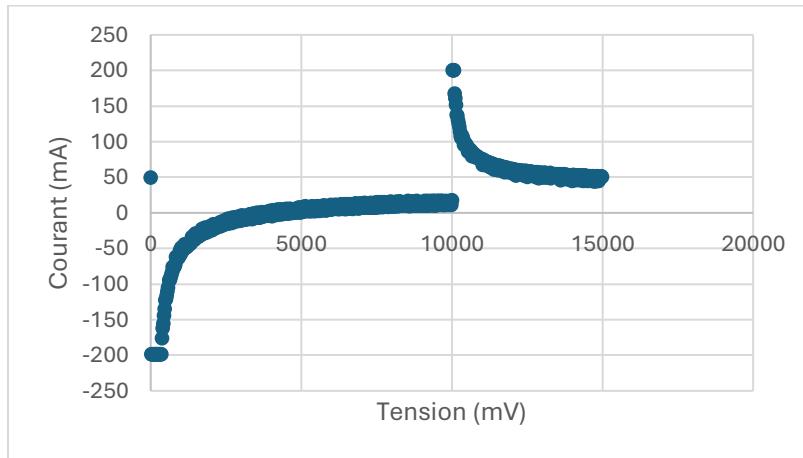


Figure 5.10 Chronoampérométrie avec ruthénium hexaammine

Le graphique du courant selon le temps est gouverné par la loi de Cottrel, qui s'écrit :

$$i(t) = \frac{(n \cdot F \cdot A \cdot C \cdot \sqrt{D})}{\sqrt{\pi \cdot t}}$$

Il est possible de tracer le courant $i(t)$ en fonction de $\frac{1}{\sqrt{t}}$. Ce graphique devrait donner une relation linéaire, dont la pente correspond à :

$$pente = \frac{(n \cdot F \cdot A \cdot C \cdot \sqrt{D})}{\sqrt{\pi}}$$

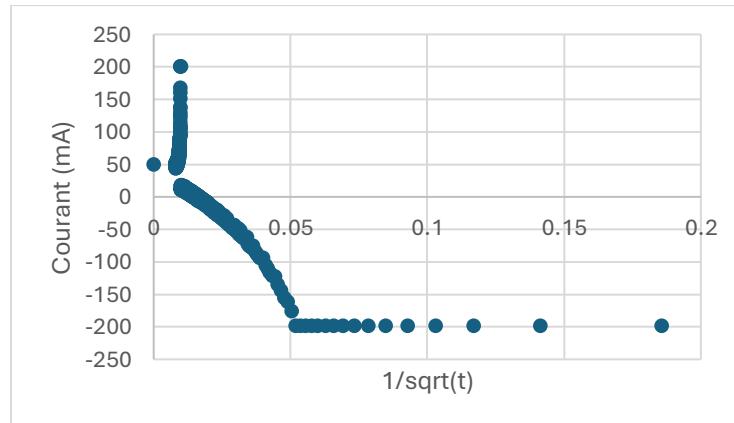


Figure 5.11 Courbe $i(t)$ en fonction de $\frac{1}{\sqrt{t}}$

Une régression linéaire a été effectué sur la région d'intérêt pour trouver la pente.

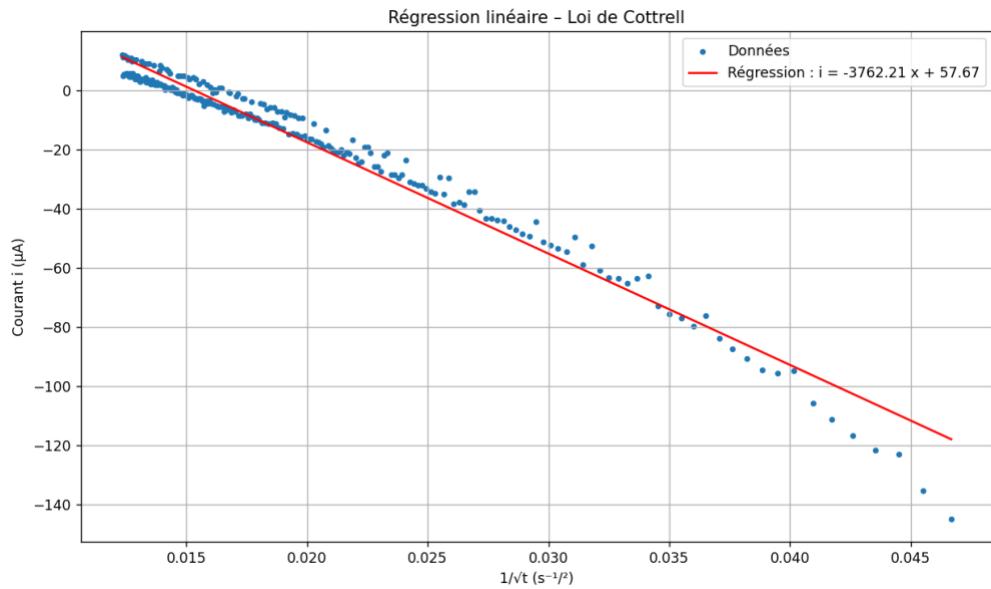


Figure 5.12 Régression linéaire – Loi de Cottrell

La pente est donc de $-3762.21 \mu\text{As}^{1/2}$

En réorganisant cette équation, on peut isoler le coefficient de diffusion D si la concentration C est connu, ou inversement :

$$D = \left[\frac{(pente \cdot \sqrt{\pi})}{(n \cdot F \cdot A \cdot C)} \right]^2$$

Les valeurs utilisées dans notre expérience sont les suivantes :

$$C = 1 \text{ mol/m}^3$$

$$n = 1$$

$$F = 96485 \text{ C/mol}$$

$$\begin{aligned} A &= \text{aire de l'extrémité correspondant à l'aire de l'extrémité d'une mine de graphite} \\ &= \pi (0,25 \text{ mm})^2 = 0,196 \text{ mm}^2 \end{aligned}$$

Ainsi en remplaçant on trouve D :

$$D = \left[\frac{(-3762.21 \mu s^{1/2} \cdot \sqrt{\pi})}{(1 \cdot 96485 C/mol \cdot 0.196 mm^2 \cdot 1 mol/m^3)} \right]^2 = 1,243 * 10^{-7} \frac{m^2}{s}$$

5.3 Limites de votre système et les sources d'erreurs

Des mesures de contrôle ont été réalisées dans une solution de KCl 1 M, ne contenant aucune molécule rédox active. En Chronoampérométrie, un potentiel constant de 0 V vs Ag|AgCl a été appliqué, et le courant mesuré se stabilise avec le temps.

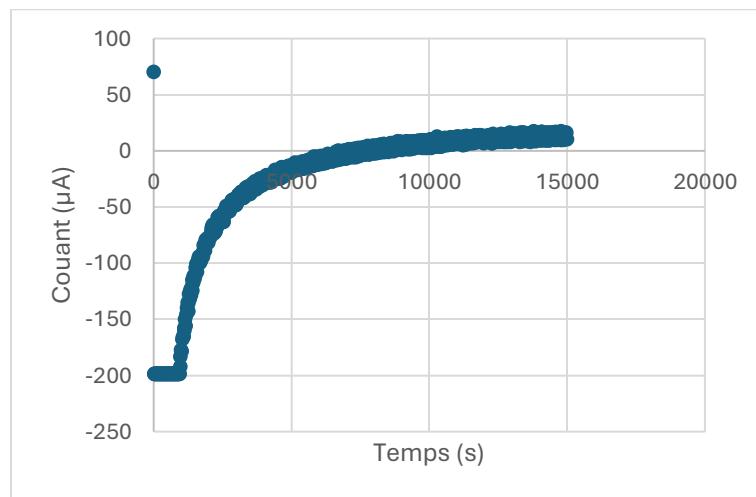


Figure 5.13. Chronoampérométrie de contrôle dans 1 M KCl

En Voltamétrie cyclique, bien que le courant varie en fonction du potentiel, aucune réaction électrochimique n'est attendue, et l'ensemble du signal représente le comportement capacitif et le bruit de fond du système. Or on observe des pics d'oxydoréduction ce qui indique la présence de contaminants dans le système, probablement des composites de la mine de crayon.

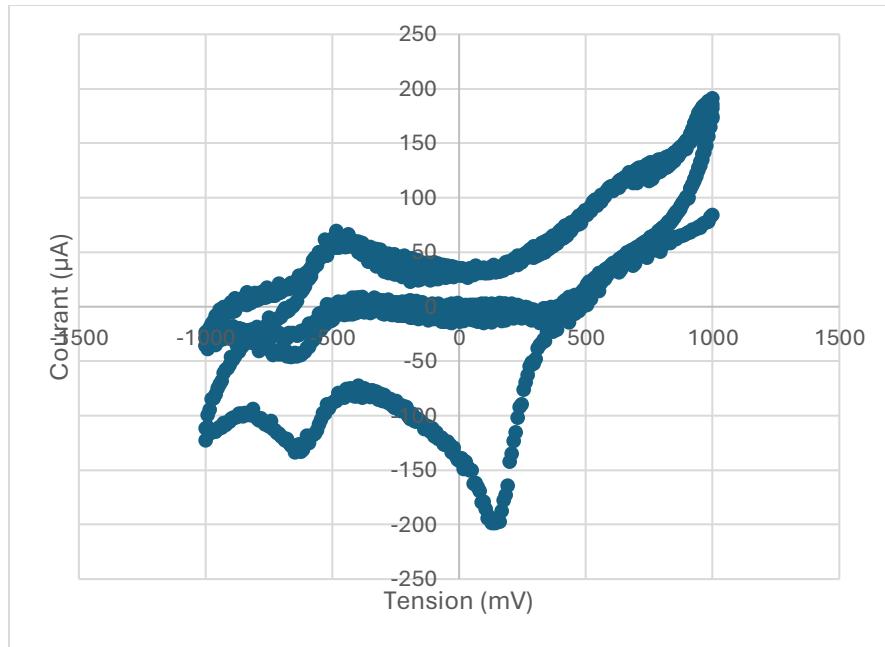


Figure 5.14 Voltamétrie cyclique de contrôle dans 1 M KCl – 3 cycles à $v = 100 \text{ mV/s}$

Dans les deux cas, le bruit a été quantifié à l'aide de l'écart-type du courant mesuré, ce qui permet d'évaluer la stabilité du système en l'absence de réaction redox. Pour la Voltamétrie l'écart type est de $68,60 \mu\text{A}$ et pour la Chronoampérométrie il est de $4,83 \mu\text{A}$ dans la zone stabilisée.

Pour ce qui est des moyennes, la Voltamétrie à une valeur moyenne de $20,76 \mu\text{A}$ et la Chronoampérométrie une valeur de $8,12 \mu\text{A}$.

En utilisant la moyenne et l'écart type, il est possible de définir la limite de détection comme suit :

$$\text{Bruit} = i_{\text{moy}} + 3\sigma$$

Pour la Voltamétrie :

$$\text{Bruit} = (20,76 + 3 * 68,60) \mu\text{A} = 226,56 \mu\text{A}$$

Pour la Chronoampérométrie :

$$\text{Bruit} = (8,12 + 3 * 4,83) \mu\text{A} = 22,61 \mu\text{A}$$

Finalement, pour ce qui est de la sensibilité du système, elle est définie en Voltamétrie par la pente de notre courbe d'étalonnage.

Sachant que notre courbe d'étalonnage se base sur la proportionnalité entre courant et concentration, notre approche donne directement la sensibilité. Puisque les 2 points sont (0,0) et (1,32.84).

$$\text{Sensibilité} = \frac{\text{Limite de détection}}{\text{pente de la courbe d'étalonnage}} = \frac{226,56 \mu\text{A}}{32,84 \left(\frac{\mu\text{A} * \text{L}}{\text{g}} \right)} = 6,899 \frac{\text{g}}{\text{L}}$$

Ainsi il est donc possible d'affirmer que le bruit de la Voltamétrie est plus élevé que celui de la Chronampérométrie. Ceci peut être expliqué par le fait qu'en Voltamétrie, le courant de pic correspond à une variation rapide du signal, ce qui rend la mesure plus sensible au bruit et donc moins précise. En revanche, la Chronoampérométrie repose sur une décroissance temporelle plus progressive, ce qui permet une analyse plus stable et reproductible.

Les sources d'erreur attribuables au bruit peuvent être minimisées de plusieurs façons. La première consiste à choisir un amplificateur opérationnel de précision plus élevée. Celui utilisé dans notre montage, le LMC6484, présente un niveau de bruit raisonnable pour un système de base, mais laisse place à des améliorations.

Une deuxième source potentielle d'erreur provient du signal PWM généré par la plateforme Arduino pour créer les rampes de potentiel. Ce signal est en réalité une approximation numérique d'un signal analogique, et n'est donc pas parfaitement lisse. Les variations rapides entre les états hauts et bas du PWM peuvent introduire du bruit, surtout si le filtrage n'est pas optimal. L'ajout d'un filtre passe-bas RC à la sortie du PWM permet de lisser le signal, mais une solution plus fiable consisterait à utiliser un convertisseur numérique-analogique (DAC) dédié, offrant une résolution et une stabilité bien supérieures.

Une troisième source d'erreur provient de la qualité des électrodes utilisées. En effet, si celles-ci sont fabriquées à partir de matériaux hétérogènes, comme des mines de crayon contenant des traces d'autres métaux réactifs, cela peut induire des réactions électrochimiques parasites non souhaitées. Notamment ceci est observable dans le tracé du KCl.

6 Développement durable

Pour ce qui est du pilier environnemental, le potentiostat a été développé sur une plateforme Arduino Uno, une breadboard, ainsi qu'un PCB custom et des composants électroniques de base. L'Arduino, la breadboard et les composants comme les condensateurs et les résistances sont réutilisables, ce qui permet de minimiser l'empreinte écologique du système, notamment pendant la phase de prototypage. Cela évite l'utilisation immédiate de matériaux non recyclables. Une fois la solution validée et prête pour la production, un PCB personnalisé est utilisé, avec des composants soudés de manière permanente. Cette étape a toutefois une empreinte environnementale plus élevée, car le PCB est difficilement recyclable. De plus, une fois les pattes coupées, les résistances et condensateurs deviennent difficilement réutilisables. Enfin, des déchets chimiques sont générés par les solutions de paracétamol et de ruthénium hexaammine. Bien que les volumes utilisés soient faibles, leur accumulation sur la durée et leur utilisation par plusieurs équipes rendent leur impact non négligeable.

Pour ce qui est du pilier social, ce projet permet d'augmenter l'accès à des outils d'analyse électrochimique, normalement réservés à des laboratoires bien équipés. En réduisant les coûts, il devient envisageable d'utiliser ce type de dispositif dans des milieux éducatifs, communautaires ou même dans des contextes de santé à ressources limitées. De plus l'utilisation des mines de crayons comme électrodes est une approche intéressante, puisqu'il s'agit d'un item facilement procurable et à faible coût.

De plus, le projet encourage l'apprentissage interdisciplinaire en combinant électronique, programmation et chimie ce qui permet de faire évoluer les compétences des membres du projet.

Finalement pour ce qui est du pilier économique, le projet permet de créer un potentiostat expérimental pour une fraction du prix d'un potentiostat retrouvé sur le marché pour une performance satisfaisante. Bien que la performance soit satisfaisante, elle est imparfaite et seulement adéquate à un usage pédagogique et non professionnel.

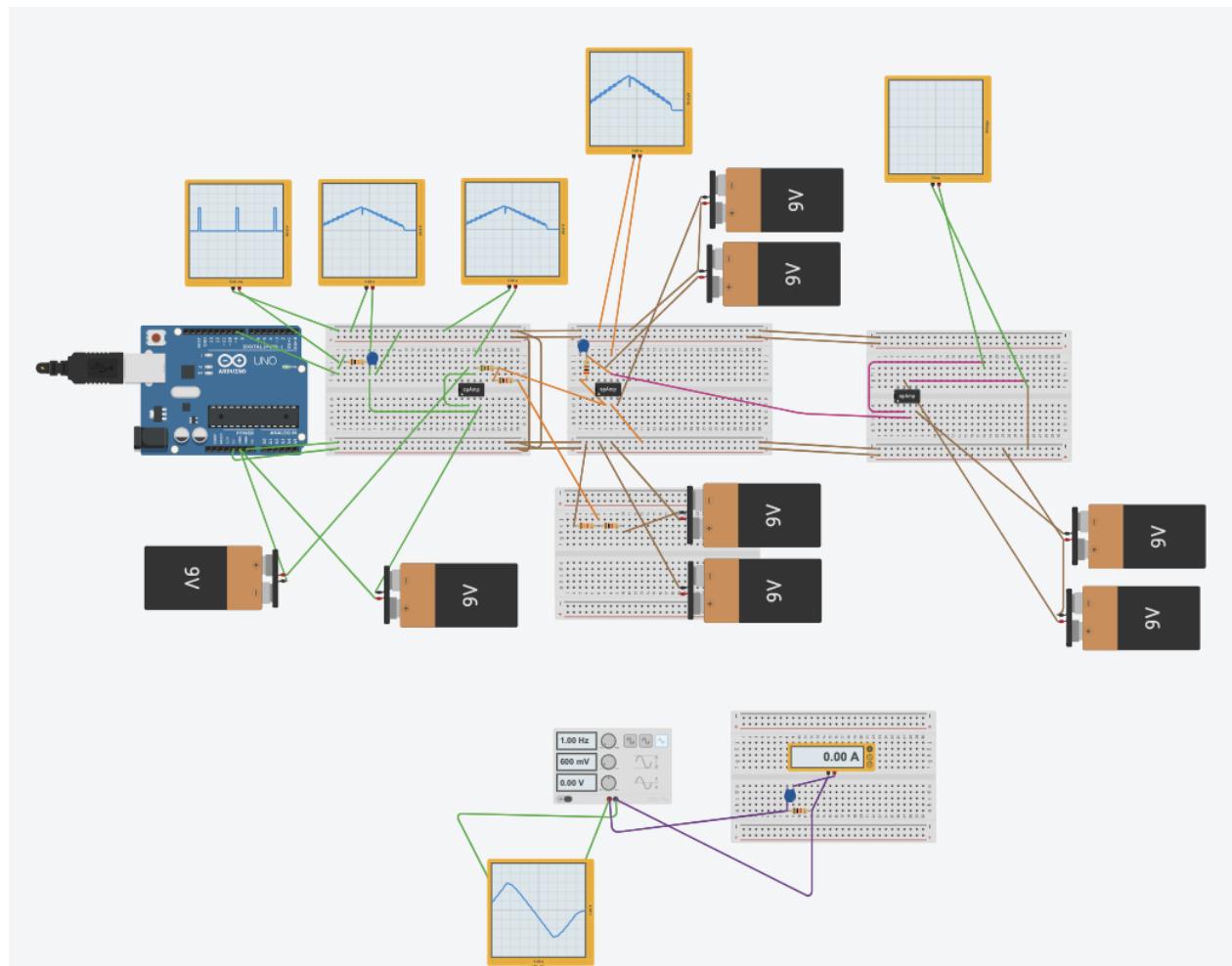
Bibliographie

[1] Wikipédia 2020.

[2] G. N. Meloni, *Supporting Information for "Building a Microcontroller Based Potentiostat: An Inexpensive and Versatile Platform for Teaching Electrochemistry and Instrumentation"*, Instituto de Química, Universidade de São Paulo, São Paulo, Brazil. [Online]. Available: <http://www.ti.com/lit/ds/symlink/lm324.pdf> (accédé le 18 avr. 2025).

Annexe

Annexe 1 : circuit test tinkerCad et code première partie du circuit



Lien TinkerCad :

<https://www.tinkercad.com/things/79aYovd4ekB/edit?returnTo=%2Fdashbaard%2Fdesigns%2Fcircuits&sharecode=hg7rGuiPAcgXg9DvQQhvi9POSZsDg-7fsMnimA-RWRA>

Annexe 2 : Planification de la structure du code

LOOP

- o Choix méthodes:
 - 1 - chronampérométrie
 - 2 - Voltammetrie cyclique

```

if (Choix = 1) | if (Choix = 2)
    • Saisir et enregistrer | • Saisir et enregistrer
    dans variable | dans variable
    E1, E2, t0 | E1, E2, vitesse, cycles
    • bool chronoaff=true | • bool voltoaff=true
    if (chronoaff=true) | if (voltoaff=true)
        •
    
```

Code

FUNCTION

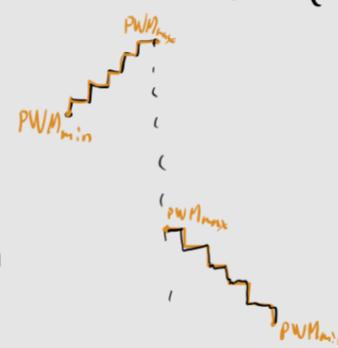
chronoamperometrieFunction()

- convertir 0 et 5V en PWM_{E1} et PWM_{E2}
- if (temps_act < t₀)
 - analogwrite(PWM_{E1})
- else
 - analogwrite(PWM_{E2})
- creationTableau()

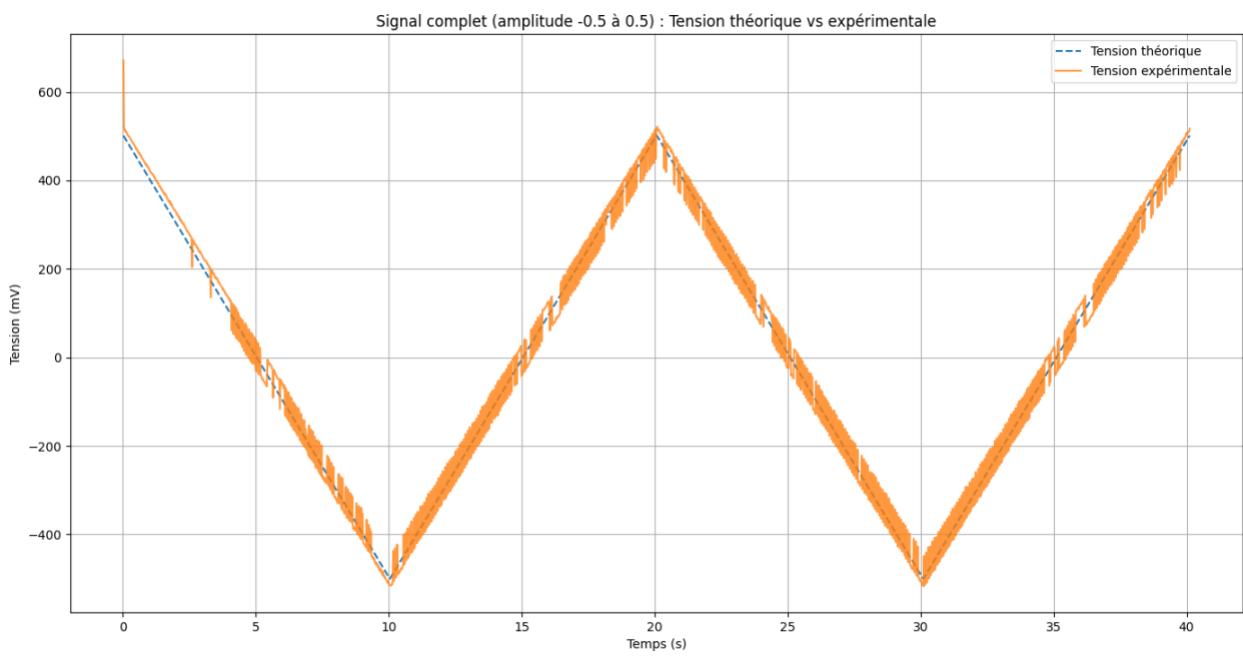
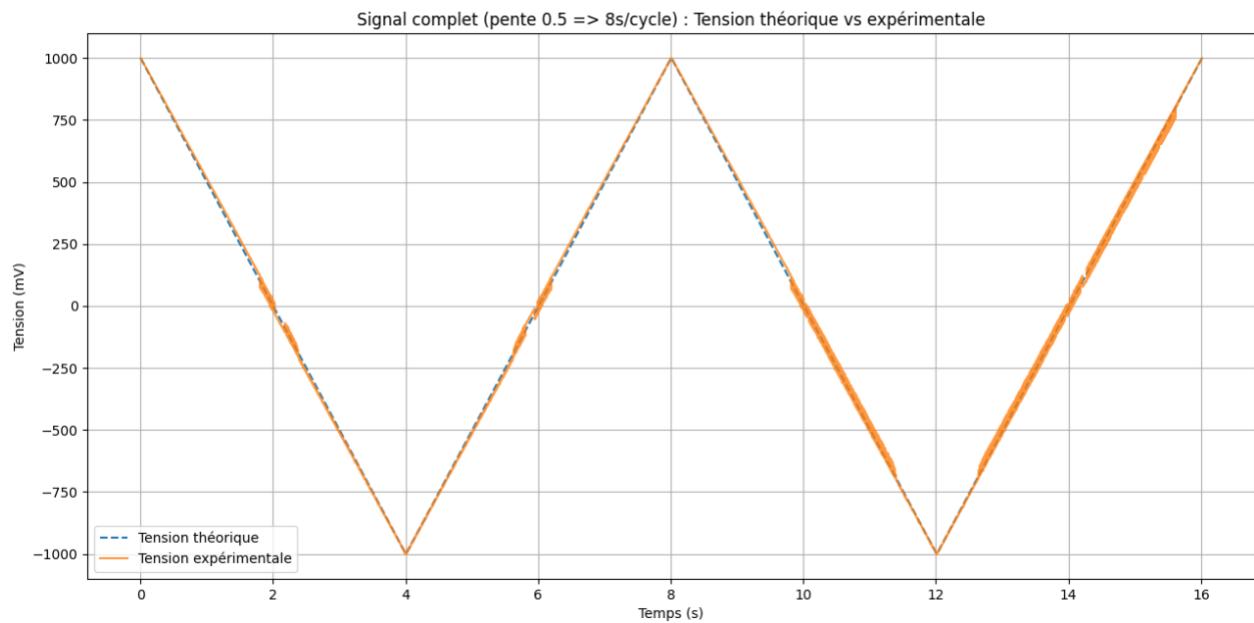


VoltammetrieFunction()

- Convertir 0 à 5V → PWM_{min} à PWM_{max}
- while (compteur_cycle < cycles)
 - Monter : for (value++)
 - AnalogWrite
 - creationTableau()
 - delay()
 - Descendre for (value--)
 - AnalogWrite
 - creationTableau()
 - delay()
 - compteur à jour



Annexe 3 : Test génération du signal triangulaire pour différents paramètres



Annexe 4 : Test global supplémentaire avec modélisation du courant capacitif d'un potentiostat

