

POLYTECH ORLÉANS

École d'Ingénieurs de l'Université d'Orléans

ULRICH PARÉ - 4A GÉNIE PHYSIQUE ET SYSTÈMES EMBARQUÉS

Intégration de Capteurs de Pouls et de Température à une Plateforme ARDUINO pour la Surveillance du Vêlage en Milieu Agricole

LACME, Route du Lude 72200 La Flèche, Bureau d'étude électronique



ANNÉE 2023-2024

Maître de stage : Marc GEORGIN, Ingénieur Bureau d'étude électronique

Tuteur enseignant : Guy LAMARQUE, Directeur de la spécialité Smart Building

Remerciements

Je tiens à exprimer ma sincère gratitude à tous ceux qui ont contribué à la réussite de ce stage. Tout d’abord, je tiens à remercier M. Marc Geogin, Ingénieur au Bureau d’Etudes Electroniques de LACME. Ses conseils concernant le projet ont été très appréciés. Sa disponibilité et ses suggestions ont été précieuses.

Je tiens également à remercier sincèrement M. Guy Lamarque, directeur de la spécialisation Smart Building à Polytech Orléans, qui m’a apporté un soutien académique et des conseils précieux au cours de cette période. Ses encouragements et son expertise ont été essentiels à la progression de mon travail.

Des remerciements particuliers doivent être adressés à M. Belkacem Bouaoune, directeur de la division électronique de LACME, qui a fait preuve de soutien et de confiance à l’égard de mon stage. Son leadership et sa vision ont réellement facilité mon intégration au sein de l’équipe.

Je tiens également à remercier M. Valery Hamm, chef de projet LACME, pour son encadrement durant cette expérience. Sa confiance et ses conseils avisés ont été d’une grande utilité pour mener à bien ce projet.

Je tiens à exprimer ma gratitude à tous les membres du personnel de LACME pour leur aimable hospitalité et leur coopération, ainsi qu’à mes collègues stagiaires pour leur bonne camaraderie et leur aide précieuse dans la mise en œuvre des diverses activités.

Résumé

Ce rapport présente le travail effectué durant le stage intitulé "**Intégration de Capteurs de Pouls et de Température à une Plateforme ARDUINO pour la Surveillance du Vêlage en Milieu Agricole**". Le projet a consisté à choisir et intégrer des capteurs sur une plateforme ARDUINO, et à modifier le logiciel pour gérer ces nouveaux capteurs. Les modifications incluent la gestion des capteurs, l'ajout de colonnes de pouls et/ou de température dans le fichier CSV de sortie, et la possibilité de modifier la fréquence de lecture des capteurs. L'objectif final est d'améliorer la précision et la fiabilité de la surveillance du vêlage en milieu agricole en fournissant des données supplémentaires sur les paramètres vitaux des animaux.

Abstract

This report presents the work carried out during the internship entitled "**Integration of Pulse and Temperature Sensors into an ARDUINO Platform for Calving Monitoring in an Agricultural Environment**". The project involved selecting and integrating sensors into an ARDUINO platform, and modifying the software to manage these new sensors. The modifications include sensor management, adding pulse and/or temperature columns to the CSV output file, and enabling the modification of sensor reading frequency. The final objective is to improve the accuracy and reliability of calving monitoring in an agricultural environment by providing additional data on the animals' vital parameters.

Table des matières

1. Introduction	5
2. Présentation de l'Entreprise	7
2.1 Historique et Activités	7
2.2 Organisation	7
2.3 Nos Clients	8
3. Travail Réalisé	9
3.1 Description des Tâches	9
3.1.1 Produit Initial	9
3.1.2 Choix des Capteurs	10
3.1.3 Intégration des Capteurs sur la Plateforme ARDUINO	11
3.1.4 Modification du Logiciel de la Plateforme	11
3.1.5 Tests et Validation	11
3.1.6 Documentation et Formation	12
3.2 Méthodes et Procédés	12
3.2.1 Méthodes Utilisées	12
3.2.2 Procédés Suivis	12
3.2.3 Outils et Systèmes Appliqués	13
3.3 Organisation et Systèmes	13
3.3.1 Organisation du Travail	13
3.3.2 Systèmes en Place pour la Gestion des Projets	14
4. Analyse et Réflexion	15
4.1 Analyse des Problèmes	15
4.1.1 Absence de Documentation Initiale	15
4.1.2 Modification de la Fréquence d'Échantillonnage	15
4.1.3 Gestion des Interférences Électromagnétiques	16
4.1.4 Traitement des Données du Capteur de Pouls	16
4.1.5 Changement de Capteur Accéléromètre	16
4.2 Proposition de Solutions	17
4.2.1 Création de Documentation Complémentaire	17
4.2.2 Optimisation de la Gestion de la Fréquence d'Échantillonnage	17
4.2.3 Réduction des Interférences Électromagnétiques	18
4.2.4 Développement et Affinement des Algorithmes pour le Capteur de Pouls	18
4.2.5 Remplacement et Intégration du Nouveau Capteur Accéléromètre	19
4.3 Choix et Contraintes	19
4.3.1 Fréquence d'Échantillonnage	20
4.3.2 Développement des Algorithmes de Traitement de Données	21
4.3.3 Considérations Commerciales	21
5. Conclusion	22
6. Bibliographie	24

A	Journal du Stage	27
A.1	Contacts Préparatoires	27
A.2	Film du Stage	27
A.3	Contacts avec l’Enseignant	28
A.4	Calendrier de Réalisation du Rapport	28
B	Code Source du Projet	29
C	Résultats Expérimentaux	37
C.1	Données sur 25 minutes	37
C.2	Données sur 30 minutes	38
C.3	Données sur 1 heure	39
D	Composants Matériels	40

1. Introduction

Contexte du Stage

Ce stage s'est déroulé au sein de l'entreprise LACME, située à La Flèche et spécialisée dans le développement de solutions technologiques pour l'industrie agricole. L'objectif principal de ce stage était de contribuer à un projet de recherche et développement visant à intégrer des capteurs de pouls et de température sur une plateforme ARDUINO pour améliorer la surveillance du vêlage.

LACME est reconnue pour son expertise dans la conception de dispositifs électroniques innovants pour l'agriculture, notamment des solutions de monitoring pour le bien-être animal. L'entreprise collabore avec divers partenaires dans le domaine agricole pour développer des technologies avancées répondant aux besoins spécifiques des éleveurs.

L'industrie agricole est en constante évolution, avec une demande croissante pour des technologies qui améliorent l'efficacité et la précision des pratiques de gestion des troupeaux. La surveillance du vêlage est un aspect crucial, car elle permet de réduire les risques liés à la mise bas et d'assurer une meilleure santé pour les animaux. Dans ce contexte, l'ajout de capteurs de pouls et de température vise à fournir des données plus complètes et précises, facilitant ainsi la prise de décisions éclairées par les éleveurs.

Mon rôle durant ce stage consistait à sélectionner les capteurs appropriés, à les intégrer sur la plateforme ARDUINO, et à modifier le logiciel existant pour gérer ces nouveaux capteurs. Cette expérience m'a permis de mettre en pratique mes compétences en systèmes embarqués tout en contribuant à un projet ayant un impact direct sur l'industrie agricole.

Objectifs du Rapport

Ce rapport a pour objectifs principaux de décrire les étapes de choix et d'intégration des capteurs de pouls et de température, d'expliquer les modifications apportées au logiciel de la plateforme pour gérer ces capteurs, de présenter les résultats obtenus et les défis rencontrés, et enfin de démontrer l'impact de cette intégration sur la surveillance du vêlage.

Premièrement, il s'agira de détailler le processus de sélection des capteurs sur la plateforme Gotronic, en précisant les critères de choix, ainsi que les étapes d'intégration sur la plateforme ARDUINO. Ensuite, ce rapport expliquera les modifications effectuées sur le code source pour permettre la gestion des nouveaux capteurs, y compris la lecture des données, le traitement des informations, et l'intégration dans le fichier CSV (fichier de base des données recueillies - sans formatage particulier) de sortie.

De plus, les résultats des tests effectués avec les capteurs intégrés seront analysés, en discutant des défis techniques rencontrés lors de l'intégration et de la programmation, ainsi que des solutions apportées pour résoudre ces problèmes. Enfin, il sera démontré comment l'ajout des capteurs de pouls et de température améliore la précision et la fiabilité de la surveillance du vêlage, contribuant ainsi à une meilleure prise de décision par les éleveurs.

Ces objectifs permettront de mettre en lumière les compétences techniques et analytiques développées au cours du stage, ainsi que l'importance de ce projet pour l'industrie agricole.

2. Présentation de l'Entreprise

2.1 Historique et Activités

L'entreprise LACME, située à La Flèche, France, est spécialisée dans le développement de solutions technologiques pour l'industrie agricole. Fondée en 1956, elle a commencé par la fabrication d'électrificateurs, puis a élargi ses activités pour inclure l'assemblage de compresseurs et de chargeurs, ainsi que le négoce d'outillage pour air comprimé.

Au fil des ans, l'entreprise a continué de diversifier ses produits et services. En 1973, la production électronique a été transférée à La Flèche, suivie en 1974 par les activités de compresseurs. En 1993, la marque LAC'AIR a été créée, et en 1995, POMELAC a été fondée pour l'injection plastique. Aujourd'hui, elle dispose d'une présence internationale avec des sites en Chine (Dafeng) et en Pologne (POMELAC).

LACME emploie environ 190 personnes en France et dispose de 15 000 m² de bâtiments dédiés au développement et à l'innovation, à l'intégration industrielle et au service client. Les activités de l'entreprise couvrent une large gamme de produits, notamment l'air comprimé, les chargeurs de batterie, les colliers de dressage, et les clôtures électriques et leurs accessoires.

2.2 Organisation

LACME est organisée en plusieurs départements, chacun spécialisé dans un domaine particulier de l'entreprise. Le pôle électronique, où j'ai effectué mon stage, est responsable du développement et de l'innovation dans les dispositifs électroniques. Ce pôle travaille en étroite collaboration avec d'autres départements tels que le service client et l'intégration industrielle pour assurer la qualité et l'efficacité des produits.

En ce qui concerne ses filiales, elle comprend :

- **LACME Dafeng (Chine)** : Employant 107 personnes, cette filiale se concentre sur la production de compresseurs et de fils.
- **POMELAC (Pologne)** : Avec 94 employés, cette filiale est spécialisée dans l'injection plastique, produisant des isolateurs et des piquets.

2.3 Nos Clients

LACME sert une clientèle diversifiée incluant les secteurs du libre-service agricole, du machinisme agricole, et de la quincaillerie professionnelle. Environ 20% de la production est destinée à l'exportation, principalement des clôtures électriques. L'entreprise met un point d'honneur à fournir des produits de haute qualité adaptés aux besoins spécifiques de ses clients, qu'il s'agisse de grands exploitants agricoles ou de petits éleveurs.

3. Travail Réalisé

3.1 Description des Tâches

Durant mon stage chez LACME, j'ai été impliqué dans plusieurs tâches clés visant à intégrer des capteurs de pouls et de température sur une plateforme ARDUINO pour améliorer la surveillance du vêlage. Le projet s'est basé sur un produit initial qui comportait déjà certaines fonctionnalités. Voici une description détaillée des tâches réalisées :

3.1.1 Produit Initial

Le projet initial comprenait déjà un dispositif fonctionnel capable d'enregistrer les données d'accélération sur une carte SD. Le programme permettait d'écrire un fichier (datalog.txt) sur la carte SD, comprenant sur chaque ligne le timestamp (horodatage), le nombre de millisecondes, et les accélérations selon les axes X, Y, et Z. La fréquence d'acquisition était déterminée par la constante "période".

Le programme utilisait les bibliothèques suivantes :

- **Wire.h** : Gestion du capteur accéléromètre.
- **Adafruit_Sensor.h** et **Adafruit_ADXL345_U.h** : Gestion de l'accéléromètre ADXL345.
- **SPI.h** et **SD.h** : Gestion de la carte SD.
- **TimeLib.h** et **Timer.h** : Gestion des timers.
- **RTCLib.h** : Gestion du module RTC.
- **FlashStorage_SAMD.h** : Écriture des fuses bits dans la dataflash.

Le code initial se chargeait de l'initialisation de la carte SD et du module RTC, de l'acquisition des données de l'accéléromètre, et de leur enregistrement périodique sur la carte SD.

3.1.2 Choix des Capteurs

La première étape de mon projet a consisté à sélectionner les capteurs de pouls et de température les plus appropriés pour notre application.

Choix du Capteur de Pouls

J'ai choisi le capteur MAX30101 pour les raisons suivantes :

- **Précision et Fiabilité** : Le MAX30101 est connu pour sa précision élevée dans la mesure du pouls et de la saturation en oxygène du sang (SpO2), ce qui est crucial pour une surveillance précise des animaux.
- **Compatibilité** : Ce capteur est compatible avec la plateforme ARDUINO, ce qui facilite son intégration dans le système existant.
- **Consommation d'Énergie** : Le MAX30101 a une faible consommation d'énergie, ce qui est essentiel pour une utilisation continue sur le terrain.
- **Documentation et Support** : Il existe une abondante documentation et support communautaire pour le MAX30101, ce qui aide à résoudre les problèmes techniques rapidement.

Choix du Capteur de Température

J'ai choisi le capteur de température ADT7410 pour les raisons suivantes :

- **Précision** : L'ADT7410 offre une précision de $\pm 0.5^{\circ}\text{C}$, ce qui est suffisant pour les applications de surveillance de la température corporelle des animaux.
- **Facilité d'Intégration** : Ce capteur utilise une interface I2C, simplifiant ainsi son intégration avec la plateforme ARDUINO.
- **Robustesse** : Le capteur est conçu pour fonctionner dans des environnements difficiles, ce qui le rend adapté aux conditions agricoles.
- **Documentation et Support** : Comme le MAX30101, l'ADT7410 bénéficie d'une documentation exhaustive et d'un support communautaire, facilitant ainsi son utilisation.

3.1.3 Intégration des Capteurs sur la Plateforme ARDUINO

Une fois les capteurs sélectionnés, j'ai procédé à leur intégration sur la plateforme ARDUINO. Cette étape a impliqué plusieurs sous-tâches :

- **Montage et Soudure** : Assembler les capteurs et les souder correctement sur les circuits imprimés afin de garantir une connexion fiable.
- **Configuration des Ports** : Configurer les ports de la carte ARDUINO pour recevoir les signaux des capteurs, en tenant compte des spécifications électriques et des besoins en alimentation.
- **Tests Préliminaires** : Réaliser des tests initiaux pour vérifier que les capteurs fonctionnent correctement et qu'ils communiquent efficacement avec la carte ARDUINO avant de les intégrer au produit final.

3.1.4 Modification du Logiciel de la Plateforme

Pour gérer les nouveaux capteurs, il a été nécessaire de modifier le logiciel de la plateforme ARDUINO. Les principales modifications apportées ont été d'ajouter la Gestion des Capteurs qui impliquait donc l'écriture et l'intégration des routines spécifiques pour lire les données des capteurs de pouls et de température.

Ensuite adapter le format de sortie des données était nécessaire pour inclure les nouvelles mesures, ajoutant des colonnes pour le pouls et la température, en plus des colonnes existantes pour le timestamp et les accéléromètres.

Pour finir il fallait régler de la fréquence d'échantillonnage en implémentant une fonctionnalité permettant de modifier dynamiquement la fréquence de lecture des capteurs selon les besoins spécifiques des utilisateurs.

3.1.5 Tests et Validation

Après l'intégration matérielle et logicielle, j'ai conduit une série de tests approfondis pour valider le bon fonctionnement des capteurs et du système global :

- **Tests en Conditions Réelles** : Tester le dispositif dans des conditions réelles d'utilisation sur le terrain pour s'assurer de sa robustesse et de sa fiabilité.
- **Analyse des Données** : Collecter et analyser les données obtenues pour vérifier leur précision et leur cohérence.

- **Détection et Correction des Erreurs** : Identifier les éventuels problèmes rencontrés lors des tests et apporter les corrections nécessaires pour améliorer le système.

3.1.6 Documentation et Formation

Enfin, j'ai rédigé une documentation détaillée du projet pour faciliter la compréhension et l'utilisation future du dispositif, incluant un manuel d'utilisateur qui explique comment utiliser le dispositif, de l'installation à la configuration, ainsi que l'interprétation des données.

Ces tâches m'ont permis de développer mes compétences en systèmes embarqués tout en contribuant de manière significative à un projet ayant un impact direct sur l'industrie agricole.

3.2 Méthodes et Procédés

Pour accomplir les tâches décrites ci-dessus, j'ai utilisé diverses méthodes et procédés, ainsi que des outils et systèmes spécifiques :

3.2.1 Méthodes Utilisées

J'ai conduit une recherche documentaire approfondie dans la littérature technique et les ressources en ligne afin d'identifier les capteurs les plus appropriés et de comprendre leurs caractéristiques et modes de fonctionnement. Par la suite, j'ai mené une analyse comparative des différentes options disponibles, en évaluant leur précision, compatibilité, consommation d'énergie et robustesse, afin de sélectionner les solutions les mieux adaptées au projet. Enfin, j'ai réalisé des prototypes pour chaque étape du développement et effectué des tests rigoureux pour valider les choix techniques et les modifications apportées.

3.2.2 Procédés Suivis

- **Intégration Matérielle** : Assemblage et soudure des capteurs sur les circuits imprimés, configuration des ports ARDUINO, et tests de connectivité.
- **Développement Logiciel** : Écriture de nouvelles routines de gestion des capteurs, modification des fichiers CSV pour inclure les nouvelles données, et réglage de la

fréquence d'échantillonnage.

- **Validation et Tests** : Tests en conditions réelles sur le terrain, collecte et analyse des données, et ajustements nécessaires pour améliorer la fiabilité et la précision du système.

3.2.3 Outils et Systèmes Appliqués

- **Plateforme ARDUINO** : Utilisée pour le développement et les tests du système intégré.
- **Librairies ARDUINO** : Utilisation des librairies spécifiques pour la gestion des capteurs, la carte SD, le module RTC, et la communication I2C.
- **Éditeur de Code** : Utilisation d'un éditeur de code comme l'IDE ARDUINO et Eclipse pour écrire et tester le code.
- **Outils de Mesure** : Multimètre et autres outils de mesure pour vérifier les connexions électriques et la qualité des signaux.
- **Logiciels d'Analyse de Données** : Utilisation de logiciels comme Excel et MATLAB pour analyser les données collectées et évaluer la performance du système.

3.3 Organisation et Systèmes

L'organisation du travail chez LACME repose sur une structure bien définie et des systèmes efficaces pour gérer les projets, assurant une coordination optimale entre les différentes équipes et départements.

3.3.1 Organisation du Travail

LACME est organisée en plusieurs départements spécialisés, chacun ayant des responsabilités distinctes mais interconnectées :

- **Département de Recherche et Développement (RD)** : Ce département est le moteur de l'innovation au sein de l'entreprise, responsable de la conception et du développement de nouveaux produits. Il est subdivisé en équipes dédiées à différents domaines technologiques, notamment l'électronique, le logiciel, et les capteurs.

- **Département de Production** : Chargé de la fabrication et de l'assemblage des produits. Il assure également le contrôle qualité pour garantir que les produits finis répondent aux normes de l'entreprise.
- **Département de Service Client** : Ce département est dédié au support technique, offrant une assistance aux clients et gérant les retours ainsi que les réparations des produits.

3.3.2 Systèmes en Place pour la Gestion des Projets

Pour gérer les projets efficacement, l'entreprise utilise plusieurs systèmes et méthodologies :

- **Gestion de Projet Agile** : Utilisation de méthodes agiles pour la gestion des projets, permettant une adaptation rapide aux changements et une meilleure collaboration entre les équipes. Les réunions de suivi régulières sont utilisées pour maintenir le projet sur la bonne voie.
- **Systèmes de Gestion de la Qualité (SGQ)** : Mise en place de procédures de contrôle qualité à chaque étape de la production. Les tests et les vérifications sont systématiquement effectués pour s'assurer de la conformité aux spécifications.
- **Outils de Collaboration** : Utilisation de logiciels de gestion de projet comme Jira et Trello pour la planification des tâches, le suivi de l'avancement, et la communication entre les membres de l'équipe. Ces outils permettent de centraliser les informations et de faciliter la coordination.
- **Documentation et Standards** : Adoption de standards de documentation pour toutes les phases de développement et de production. Chaque projet est documenté de manière exhaustive, incluant les spécifications techniques, les plans de test, et les manuels d'utilisation.

Ces systèmes et méthodologies permettent à LACME de gérer efficacement les projets, en garantissant une coordination fluide entre les différents départements et une haute qualité des produits livrés aux clients.

4. Analyse et Réflexion

4.1 Analyse des Problèmes

Au cours de ce projet, j'ai rencontré plusieurs défis techniques et méthodologiques, impactant à la fois le déroulement des travaux et les résultats obtenus.

4.1.1 Absence de Documentation Initiale

Dès le début du projet, j'ai constaté l'absence de documentation complète et précise, en dehors du code Arduino existant. Cette absence de documents de référence m'a privé d'une base solide pour appréhender le projet dans son ensemble, ainsi que pour anticiper les problèmes potentiels. L'absence de schémas, de guides d'intégration ou d'informations sur les problèmes connus a entraîné une courbe d'apprentissage abrupte et a compliqué la résolution de problèmes techniques. Cette situation a également rendu difficile l'établissement d'une feuille de route claire pour le projet.

4.1.2 Modification de la Fréquence d'Échantillonnage

Un des changements majeurs intervenus au cours du projet a été l'augmentation de la fréquence d'échantillonnage des capteurs de 200 ms à 10 ms. Cette modification, bien qu'indispensable pour obtenir une meilleure résolution des données, a ajouté une complexité significative au projet. Une fréquence d'échantillonnage plus élevée a entraîné une augmentation substantielle du volume de données générées, ce qui a exigé non seulement une optimisation des routines de collecte des données, mais également une gestion plus efficace de l'écriture des données sur la carte SD.

L'intégration de nouveaux capteurs dans ce contexte a également été plus difficile, car le temps disponible pour traiter les données et les stocker s'est trouvé considérablement

réduit. La nécessité d'assurer une acquisition de données fiable tout en respectant les contraintes temporelles strictes a représenté un défi de taille.

4.1.3 Gestion des Interférences Électromagnétiques

L'intégration du capteur de pouls MAX30101 a révélé un problème récurrent d'interférences électromagnétiques, principalement causées par le module d'accélération. Ces interférences ont eu pour conséquence des lectures erronées ou des interruptions de la communication I2C entre les différents composants du système. La résolution de ce problème a nécessité plusieurs essais d'isolation et de blindage, ainsi que des ajustements dans la gestion logicielle des interruptions. Toutefois, malgré ces efforts, le problème d'interférences n'a pas été totalement éliminé, affectant la fiabilité des données collectées.

4.1.4 Traitement des Données du Capteur de Pouls

Le capteur MAX30101, contrairement au capteur de température utilisé dans le projet, ne dispose pas d'un algorithme intégré pour interpréter les données brutes qu'il génère. Ce capteur mesure les variations de rayons infrarouges, qui doivent ensuite être converties en données exploitables pour déterminer le rythme cardiaque. En l'absence de cette fonctionnalité intégrée, j'ai dû développer moi-même les algorithmes nécessaires pour traiter ces données.

Ce travail s'est révélé particulièrement complexe, car il a fallu non seulement interpréter correctement les signaux bruts, mais aussi mettre en place des filtres adéquats pour éliminer les bruits et les erreurs de mesure. Plusieurs itérations ont été nécessaires pour parvenir à une précision acceptable, mais des erreurs de calcul subsistaient, entraînant des valeurs parfois aberrantes. Ce processus a été chronophage et a nécessité une analyse approfondie des données pour identifier et corriger les sources d'erreurs.

4.1.5 Changement de Capteur Accéléromètre

Enfin, au cours du projet, il a été nécessaire de remplacer le capteur accéléromètre initialement choisi par un modèle identique à celui utilisé sur un PCB développé en parallèle (à un stade plus avancé du projet mais sans le capteur de pouls). Ce changement a été dicté par la nécessité de garantir une compatibilité entre les différents systèmes matériels

utilisés dans le projet.

Cependant, ce remplacement n'a pas été sans conséquence. Il a impliqué une réévaluation de certains paramètres de fonctionnement et des ajustements dans l'interface de communication entre les composants. L'intégration de ce nouveau capteur a également nécessité des tests supplémentaires pour s'assurer de la cohérence des données avec celles du prototype précédent.

4.2 Proposition de Solutions

Pour surmonter les défis identifiés au cours du projet, plusieurs solutions techniques et méthodologiques ont été mises en œuvre.

4.2.1 Création de Documentation Complémentaire

Pour pallier l'absence de documentation initiale, j'ai pris l'initiative de créer des documents techniques détaillés tout au long du projet. Cela comprenait des schémas des connexions des capteurs, des descriptions des flux de données, ainsi que des guides pour l'installation et le débogage des composants. Cette documentation a servi non seulement de référence pour mes propres travaux, mais aussi de base pour les futures itérations du projet. En documentant chaque étape, j'ai pu mieux comprendre l'ensemble du système et identifier rapidement les sources potentielles de problèmes.

4.2.2 Optimisation de la Gestion de la Fréquence d'Échantillonnage

Pour gérer la transition vers une fréquence d'échantillonnage de 10 ms, j'ai procédé à une optimisation rigoureuse du code, en particulier pour la collecte des données et l'écriture sur la carte SD. Cela a inclus :

- La mise en place de techniques d'échantillonnage synchrone, permettant de coordonner la collecte des données entre les différents capteurs sans provoquer de surcharge du système.
- L'amélioration des routines d'écriture sur la carte SD, en utilisant des tampons pour accumuler les données avant de les écrire par blocs, réduisant ainsi le nombre d'écritures et optimisant le temps de traitement.

- L'ajustement de la taille des données collectées en éliminant les informations redondantes, ce qui a permis de réduire la charge de traitement.

Ces optimisations ont permis de réduire les délais de traitement et de garantir une acquisition de données fiable à la nouvelle fréquence d'échantillonnage.

4.2.3 Réduction des Interférences Électromagnétiques

Pour résoudre les problèmes d'interférences électromagnétiques affectant le capteur J'ai modifié les routines de gestion des interruptions dans le code pour mieux synchroniser la communication I2C, minimisant ainsi les erreurs de transmission dues aux interférences.

Grâce à ces mesures, les interférences électromagnétiques ont été significativement atténuées, permettant d'obtenir des données plus stables et fiables.

4.2.4 Développement et Affinement des Algorithmes pour le Capteur de Pouls

Pour traiter les données brutes du capteur MAX30101, j'ai développé un algorithme personnalisé permettant de convertir les variations de rayons infrarouges en mesures de fréquence cardiaque exploitables. Les étapes suivantes ont été essentielles dans ce processus :

- **Mise en place de filtres numériques** : J'ai implémenté un **filtre médian** pour éliminer les pics aberrants et un filtre **Moyenne mobile exponentielle (Exponential Moving Average, EMA)** pour lisser les données. Ces filtres ont été choisis pour leur capacité à atténuer les bruits tout en préservant les tendances réelles des signaux.
- **Algorithme de détection de pics** : Un algorithme de détection de pics a été développé pour identifier les battements cardiaques à partir des variations des signaux lumineux captés par le capteur. Cet algorithme a été optimisé pour minimiser les faux positifs et les erreurs de détection.
- **Test et validation** : J'ai procédé à des tests rigoureux avec des jeux de données simulées et réelles pour affiner les coefficients des filtres et les seuils de détection. Cela a permis d'améliorer la précision des mesures tout en réduisant les erreurs dues aux artefacts de mouvement ou aux interférences résiduelles.

Ces développements ont permis d'améliorer considérablement la fiabilité et la précision des données de fréquence cardiaque obtenues.

4.2.5 Remplacement et Intégration du Nouveau Capteur Accéléromètre

En ce qui concerne le changement de capteur accéléromètre, j'ai procédé à une intégration soignée du nouveau modèle en adoptant les mesures suivantes :

- **Mise à jour du code** : J'ai ajusté les paramètres de configuration du nouveau capteur dans le code pour s'assurer qu'il soit compatible avec les autres composants du système. Cela incluait la réécriture de certaines fonctions pour correspondre aux spécifications du nouveau capteur.
- **Tests d'intégrité** : Des tests comparatifs ont été réalisés pour vérifier que les données obtenues avec le nouveau capteur étaient cohérentes avec celles du précédent prototype, garantissant ainsi la continuité et la fiabilité du projet.
- **Documentation des modifications** : Les changements apportés ont été soigneusement documentés pour faciliter la prise en main du projet par d'autres membres de l'équipe ou pour de futures mises à jour.

Ces actions ont permis une transition fluide vers l'utilisation du nouveau capteur tout en maintenant la qualité des données collectées.

4.3 Choix et Contraintes

Dans le cadre du développement de ce projet, plusieurs choix ont été dictés par des contraintes techniques, commerciales, et logistiques. Cette section présente une analyse détaillée des principales décisions prises, ainsi que les raisons qui les sous-tendent.

Choix du Matériel

Capteurs

Le choix des capteurs a été fortement influencé par les contraintes techniques suivantes :

- **Compatibilité** : Le capteur de fréquence cardiaque MAX30101 et l'accéléromètre étaient compatibles avec les protocoles de communication I2C déjà utilisés dans

le projet. Cette compatibilité a simplifié leur intégration et réduit la complexité du code.

- **Précision des mesures :** Le MAX30101 a été choisi pour sa capacité à fournir des mesures précises de la fréquence cardiaque via des variations de rayons infrarouges. Bien qu'il n'inclue pas d'algorithme de traitement intégré, cela a permis une plus grande flexibilité dans le développement d'algorithmes personnalisés adaptés aux besoins spécifiques du projet.
- **Disponibilité et coût :** Le capteur d'accélération sélectionné a été choisi non seulement pour ses caractéristiques techniques, mais aussi pour sa disponibilité sur le marché et son coût abordable. En effet, le coût des composants a été un facteur clé, car il était nécessaire de maintenir le budget global du projet dans des limites acceptables pour garantir sa viabilité commerciale.
- **Évolution du projet :** Le changement de capteur d'accélération pour un modèle compatible avec un PCB déjà en cours de développement a été motivé par la nécessité d'assurer une cohérence matérielle entre les différentes étapes du projet. Ce choix a permis de limiter les ruptures dans la chaîne de développement et d'optimiser les ressources déjà investies.

4.3.1 Fréquence d'Échantillonnage

- **Exigences de performance :** L'augmentation de la fréquence d'échantillonnage à 10 ms a été nécessaire pour répondre aux exigences de performance du projet, notamment en termes de réactivité et de précision des données collectées. Cependant, ce choix a impliqué des compromis, notamment en termes de gestion des ressources système et du traitement des données.
- **Limites matérielles :** Les contraintes imposées par la vitesse de traitement du microcontrôleur et la capacité de stockage de la carte SD ont nécessité une optimisation du code pour garantir que le système puisse gérer la collecte et le stockage de données en temps réel sans perte d'information.
- **Fiabilité des données :** Pour assurer la fiabilité des données à cette nouvelle fréquence, j'ai dû intégrer des techniques de filtrage avancées, comme le filtre médian et la Moyenne mobile exponentielle (EMA), afin de réduire le bruit et les erreurs potentielles dans les mesures.

4.3.2 Développement des Algorithmes de Traitement de Données

- **Flexibilité algorithmique** : En choisissant de développer un algorithme personnalisé pour le capteur MAX30101, j'ai pu adapter précisément le traitement des données aux spécificités du projet. Ce choix a permis de surmonter les limitations des algorithmes génériques et d'optimiser la précision des mesures.
- **Complexité de développement** : Le développement de ces algorithmes a ajouté une complexité supplémentaire au projet, entraînant une phase d'expérimentation et de validation plus longue. Bien que ce compromis ait prolongé le temps de développement, il était indispensable pour garantir les performances attendues.

4.3.3 Considérations Commerciales

- **Optimisation des coûts** : Tout au long du projet, le choix des composants a été guidé par un souci d'équilibre entre performance et coût. En sélectionnant des capteurs fiables mais abordables, et en optimisant l'usage des ressources existantes, j'ai pu développer un prototype répondant aux exigences techniques tout en restant compétitif sur le plan financier.
- **Scalabilité** : Les décisions prises quant aux composants et aux algorithmes tiennent également compte de la scalabilité du produit final. Le système a été conçu pour être facilement extensible, permettant ainsi des adaptations futures sans nécessiter de refonte complète, ce qui est un atout commercial important.

En conclusion, les choix effectués tout au long du projet ont été le résultat d'un équilibre complexe entre les contraintes techniques et les objectifs de performance. Chaque décision a été soigneusement pesée pour maximiser la viabilité du projet tout en respectant les limites imposées par les ressources disponibles.

5. Conclusion

Bilan du Stage

Ce stage a été extrêmement enrichissant tant sur le plan technique que professionnel. J'ai acquis des compétences avancées en intégration de capteurs comme le MAX30101 et les accéléromètres, ainsi qu'en traitement des signaux avec des techniques telles que le filtre médian et la Moyenne mobile exponentielle (EMA). La résolution des problèmes liés aux interférences électromagnétiques et l'écriture d'algorithmes personnalisés ont renforcé ma compréhension des défis pratiques en électronique.

J'ai également appris à gérer des projets complexes, en adaptant les solutions aux contraintes techniques et en améliorant mes compétences en communication et en travail en équipe. Ce stage a permis d'appliquer les connaissances théoriques acquises en cours à des situations réelles, consolidant ainsi mon savoir-faire technique et ma préparation professionnelle. En résumé, cette expérience a grandement contribué à ma formation en ingénierie et m'a préparé à relever des défis futurs dans ma carrière.

Perspectives d'Avenir

Cette expérience de stage a significativement enrichi mes perspectives de carrière en ingénierie. En travaillant sur un projet innovant d'intégration de capteurs et de traitement de données en milieu agricole, j'ai renforcé mes compétences techniques en systèmes embarqués, en traitement du signal, et en gestion de projets. La résolution des problèmes complexes et l'adaptation aux contraintes spécifiques du projet m'ont appris à aborder les défis techniques avec une approche analytique et créative.

De plus, la collaboration avec des professionnels du secteur et l'application pratique de concepts théoriques ont confirmé mon intérêt pour les technologies de surveillance et

l'innovation en ingénierie. Cette expérience m'a également ouvert des perspectives sur des domaines spécialisés tels que l'Internet des objets (IoT) et les systèmes de monitoring. En conséquence, je me sens mieux préparé à poursuivre une carrière dans le développement de solutions technologiques avancées, avec une préférence marquée pour les projets combinant ingénierie et applications pratiques dans des environnements industriels ou agricoles.

6. Bibliographie

- **Adafruit**, *Adafruit Feather M0 Adalogger*, Adafruit Industries, New York, 2016. Disponible en ligne : <https://www.adafruit.com/product/2796>, consulté le 7 mai 2024.
- **SparkFun**, *SparkFun MAX3010x Sensor Library Documentation*, SparkFun Electronics, Boulder, Colorado, 2019. Disponible en ligne : https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library/tree/master, consulté le 30 mai 2024.
- **SparkFun**, *MAX30105 Datasheet*, SparkFun Electronics, Boulder, Colorado, 2016. Disponible en ligne : https://cdn.sparkfun.com/assets/learn_tutorials/5/7/7/MAX30105_3.pdf, consulté le 30 mai 2024.
- **Adafruit**, *Adafruit ADT7410 Library Documentation*, Adafruit Industries, New York, 2020. Disponible en ligne : https://github.com/adafruit/Adafruit_ADT7410/tree/master, consulté le 30 mai 2024.
- **Adafruit**, *ADT7410 Breakout Datasheet*, Adafruit Industries, New York, 2016. Disponible en ligne : <https://cdn-learn.adafruit.com/downloads/pdf/adt7410-breakout.pdf>, consulté le 30 mai 2024.
- **DFRobot**, *DFRobot LIS2DW12 Library Documentation*, DFRobot, Shanghai, 2021. Disponible en ligne : https://github.com/DFRobot/DFRobot_LIS/tree/master/examples/LIS2DW12, consulté le 14 juin 2024.
- **DFRobot**, *LIS2DW12 Triple Axis Accelerometer Datasheet*, DFRobot, Shanghai, 2021. Disponible en ligne : https://wiki.dfrobot.com/LIS2DW12_Triple_Axis_Accelerometer_SKU_SEN0405, consulté le 14 juin 2024.

- **Adafruit**, *Adafruit PCF8523 Real-Time Clock Library Documentation*, Adafruit Industries, New York, 2019. Disponible en ligne : <https://learn.adafruit.com/adafruit-pcf8523-real-time-clock>, consulté le 19 mai 2024.
- **Arduino**, *Arduino SD Library Documentation*, Arduino, Ivrea, 2020. Disponible en ligne : <https://www.arduino.cc/en/Reference/SD>, consulté le 19 mai 2024.

Table des Annexes

- **Annexe A** : Journal du Stage . . .27
- **Annexe B** : Code Source du Projet . . .29
- **Annexe C** : Ressources et Outils . . .37
- **Annexe D** : Composants matériels utilisés . . .40

Annexe A

Journal du Stage

A.1 Contacts Préparatoires

- **Date :** 21 mars 2024
Contact : Monsieur Belkacem BOUAOUNE, Directeur du Pôle Électronique
Détails : Envoi du mail de demande de stage.
- **Date :** 28 mars 2024
Contact : Recrutement LACME
Détails : Premier rendez-vous avec Monsieur Marc GEORGIN et Monsieur Belkacem BOUAOUNE pour discuter des objectifs du stage, des attentes et des ressources disponibles.
- **Date :** 5 avril 2024
Détails : Confirmation de l'intégration au stage.
- **Date :** 30 avril 2024
Détails : Signature de la convention de stage.

A.2 Film du Stage

- Le "film du stage" présente les principales étapes du travail réalisé :
- **Début du Stage :** Introduction au projet et prise en main des outils (6 mai 2024).
 - **Développement :** Intégration des capteurs, modifications du logiciel, et tests.
 - **Validation :** Analyse des données et ajustements nécessaires.
 - **Clôture :** Préparation et révisions du rapport.

A.3 Contacts avec l'Enseignant

— **Date :** 26 juin 2024

Contact : Monsieur Belkacem BOUAOUNE

Détails : Entretien annuel et évaluation de la performance.

— **Date :** 9 juillet 2024

Contact : Monsieur Guy LAMARQUE, Tuteur enseignant

Détails : Réunion avec Monsieur Marc GEORGIN pour faire le point sur l'avancée du projet.

— **Date :** 11 juillet 2024

Contact : Monsieur Belkacem BOUAOUNE et Monsieur Marc GEORGIN

Détails : Réunion pour discuter du produit final.

A.4 Calendrier de Réalisation du Rapport

— **Début du Stage :** 6 mai 2024

— **Fin du Stage :** 12 juillet 2024

— **Début de Rédaction :** 15 juillet 2024

— **Première Ébauche :** 25 juillet 2024

— **Révisions :** 30 juillet - 2 août 2024

— **Validation Finale :** 5 août 2024

— **Soumission du Rapport :** 7 août 2024

Annexe B

Code Source du Projet

Fichier main.cpp

```
1 #include <Wire.h>
2 #include <Adafruit_Sensor.h>
3 #include <SPI.h>
4 #include <SD.h>
5 #include <TimeLib.h>
6 #include <Timer.h>
7 #include "RTCLib.h"
8 #include <FlashStorage_SAMD.h>
9 #include <MAX30105.h>
10 #include "heartRate.h"
11 #include <Adafruit_ADT7410.h>
12 #include <DFRobot_LIS2DW12.h>
13
14 MAX30105 particleSensor;
15 Adafruit_ADT7410 tempsensor = Adafruit_ADT7410();
16 DFRobot_LIS2DW12_I2C acce(&Wire, 0x19);
17
18 const int PERIODE = 10;
19 const int CHIP_SELECT = 4;
20 const int LED_PIN = 13;
21 const int MAXROW = 50;
22 const int MAXCOLUMN = 70;
23 char bufferSD[MAXROW][MAXCOLUMN];
24 int nbRowsBuffer = 0;
```

```
25
26 RTC_PCF8523 rtc ;
27 const byte RATE_SIZE = 4;
28 byte rates[RATE_SIZE];
29 byte rateSpot = 0;
30 long lastBeat = 0;
31 float beatsPerMinute;
32 int beatAvg;
33
34 long lastDisplayTime = 0;
35 long lastAccelTime = 0;
36 long lastTempTime = 0;
37 long lastBPMUpdateTime = 0;
38
39 long experimentStartTime = 0;
40 const long EXPERIMENT_DURATION = 60 * 60 * 1000; //in milliseconds
41
42 FlashStorage(fuseStorage , int);
43
44 float accX, accY, accZ;
45 float tempValue;
46 File dataFile;
47
48 const int WINDOW_SIZE = 15;
49 float bpmWindow[WINDOW_SIZE];
50 int windowIndex = 0;
51 bool windowFull = false;
52
53 void setup() {
54     Serial.begin(115200);
55     Serial.println("Initializing ...");
56
57     if (!particleSensor.begin(Wire , I2C_SPEED_FAST)) {
58         Serial.println("MAX30105 was not found. Please check wiring /
59             power.");
60         while (1);
61     }
62     Serial.println("Place your index finger on the sensor with steady
63         pressure.");
```

```
62
63     particleSensor . setup () ;
64     particleSensor . setPulseAmplitudeRed (0x7F) ;
65     particleSensor . setPulseAmplitudeGreen (0) ;
66
67     Serial . println ("ADT7410 demo") ;
68     if (! tempsensor . begin ()) {
69         Serial . println ("Couldn't find ADT7410!") ;
70         while (1) ;
71     }
72     delay (250) ;
73     tempsensor . setResolution (ADT7410_16BIT) ;
74     Serial . print ("Resolution = ") ;
75     Serial . println (tempsensor . getResolution () == ADT7410_16BIT ? "16
        bits" : "?? bits") ;
76
77     initRtcSDcard () ;
78
79     if (! acce . begin ()) {
80         Serial . println ("No LIS2DW12 detected") ;
81     } else {
82         Serial . println ("LIS2DW12 detected") ;
83     }
84
85     acce . softReset () ;
86     acce . continRefresh (true) ;
87     acce . setDataRate (DFRobot_LIS2DW12 :: eRate_100hz) ;
88     acce . setRange (DFRobot_LIS2DW12 :: e2_g) ;
89     acce . setFilterPath (DFRobot_LIS2DW12 :: eLPF) ;
90     acce . setFilterBandwidth (DFRobot_LIS2DW12 :: eRateDiv_4) ;
91     acce . setPowerMode (DFRobot_LIS2DW12 :: eContLowPwrLowNoise2_14bit) ;
92
93     pinMode (LED_PIN , OUTPUT) ;
94
95     dataFile = SD . open ("datalog . txt" , FILE_WRITE) ;
96     if (! dataFile) {
97         Serial . println ("Error opening datalog . txt") ;
98         while (1) ;
99     }
```



```
100
101     experimentStartTime = millis();
102 }
103
104 void loop() {
105     long currentTime = millis();
106     if (currentTime - experimentStartTime >= EXPERIMENT_DURATION) {
107         Serial.println("Experiment duration has ended.");
108         dataFile.close();
109         while (1);
110     }
111
112     readHeartRate(currentTime);
113     updateBPMAverage(currentTime);
114     readSensorData(currentTime);
115     displayValues(currentTime);
116     writeCSV();
117 }
118
119 void readHeartRate(long currentTime) {
120     long irValue = particleSensor.getIR();
121
122     if (irValue > 50000) { // Threshold to detect if a finger is placed
123                           // on the sensor
124         if (checkForBeat(irValue)) {
125             long delta = currentTime - lastBeat;
126             lastBeat = currentTime;
127
128             beatsPerMinute = 60 / (delta / 1000.0);
129
130             if (beatsPerMinute < 255 && beatsPerMinute > 20) {
131                 rates[rateSpot++] = (byte)beatsPerMinute;
132                 rateSpot %= RATE_SIZE;
133
134                 bpmWindow[windowIndex++] = beatsPerMinute;
135                 if (windowIndex >= WINDOW_SIZE) {
136                     windowFull = true;
137                     windowIndex = 0;
138                 }
139             }
140         }
141     }
```

```
138         }
139     }
140     } else {
141         // If no finger is detected, reset the heart rate values
142         beatsPerMinute = 0;
143         beatAvg = 0;
144         memset(rates, 0, sizeof(rates));
145         memset(bpmWindow, 0, sizeof(bpmWindow));
146         rateSpot = 0;
147         windowIndex = 0;
148         windowFull = false;
149     }
150 }
151
152 void updateBPMAverage(long currentTime) {
153     if (currentTime - lastBPMUpdateTime >= 3000) {
154         beatAvg = 0;
155         int validRates = 0;
156         for (byte x = 0; x < RATE_SIZE; x++) {
157             if (rates[x] != 0) {
158                 beatAvg += rates[x];
159                 validRates++;
160             }
161         }
162         if (validRates > 0) {
163             beatAvg /= validRates;
164         }
165
166         if (windowFull) {
167             beatAvg = applyLowessSmoothing(bpmWindow, WINDOW_SIZE);
168         } else {
169             beatAvg = applyLowessSmoothing(bpmWindow, windowIndex);
170         }
171
172         lastBPMUpdateTime = currentTime;
173     }
174 }
175
176 float applyLowessSmoothing(float* window, int size) {
```

```
177     float sum = 0;
178     for (int i = 0; i < size; i++) {
179         if (window[i] != 0) {
180             sum += window[i];
181         }
182     }
183     return size > 0 ? sum / size : 0;
184 }
185
186 void readSensorData(long currentTime) {
187     if (currentTime - lastAccelTime >= PERIODE) {
188         readAccelerometer();
189         lastAccelTime = currentTime;
190     }
191
192     if (currentTime - lastTempTime >= 3000) {
193         processTemperature();
194         lastTempTime = currentTime;
195     }
196 }
197
198 void displayValues(long currentTime) {
199     if (currentTime - lastDisplayTime >= PERIODE) {
200         DateTime now = rtc.now();
201
202         Serial.print(now.unixtime());
203         Serial.print(",");
204         Serial.print(accX, 2);
205         Serial.print(",");
206         Serial.print(accY, 2);
207         Serial.print(",");
208         Serial.print(accZ, 2);
209         Serial.print(",");
210         Serial.print(beatAvg);
211         Serial.print(" BPM,");
212         Serial.print(tempValue, 1);
213         Serial.println(" C ");
214
215         lastDisplayTime = currentTime;
```

```
216     }
217 }
218
219 void readAccelerometer () {
220     accX = acce.readAccX () / 1000.0;
221     accY = acce.readAccY () / 1000.0;
222     accZ = acce.readAccZ () / 1000.0;
223 }
224
225 void processTemperature () {
226     tempValue = tempsensor.readTempC ();
227 }
228
229 void writeCSV () {
230     int i = 0;
231     char oneRow[100];
232
233     DateTime now = rtc.now();
234
235     snprintf(oneRow, sizeof(oneRow), "%d,%f,%f,%f,%d,%f",
236             now.unixtime(), accX, accY, accZ, beatAvg, tempValue);
237
238     if (nbRowsBuffer < MAXROW) {
239         strcpy(bufferSD[nbRowsBuffer], oneRow);
240         nbRowsBuffer++;
241     }
242
243     if (nbRowsBuffer == MAXROW) {
244         for (i = 0; i < MAXROW; i++) {
245             dataFile.println(bufferSD[i]);
246         }
247         nbRowsBuffer = 0;
248     }
249 }
250
251 void initRtcSDcard () {
252     if (!rtc.begin()) {
253         Serial.println("Couldn't find RTC");
254         while (1);
255     }
```

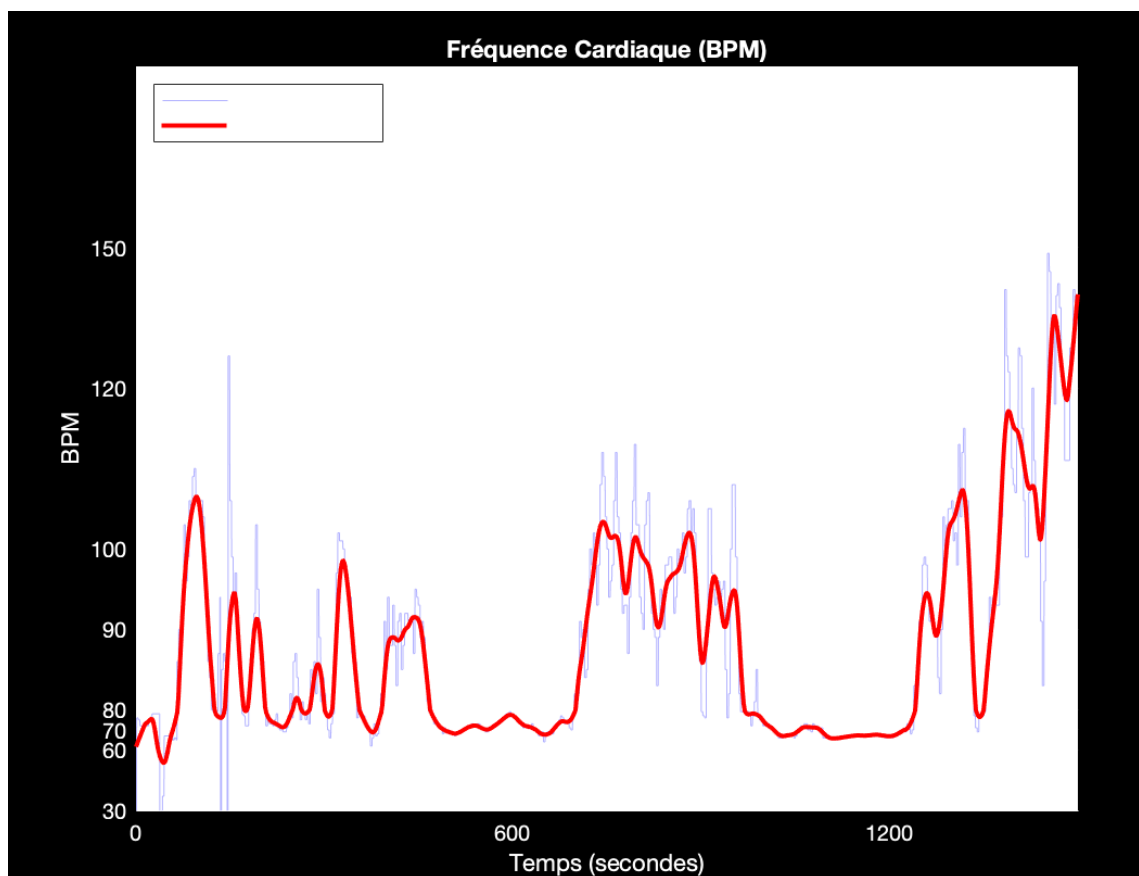
```
255     }
256
257     if (!rtc.initialized() || rtc.lostPower()) {
258         Serial.println("RTC is NOT initialized, let's set the time!");
259         rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
260     }
261
262     if (!SD.begin(CHIP_SELECT)) {
263         Serial.println("Card failed, or not present");
264         while (1);
265     }
266
267     File dataFile = SD.open("datalog.txt", FILE_WRITE);
268     if (dataFile) {
269         dataFile.println("UnixTime,accX,accY,accZ,BPM,TempC");
270         dataFile.close();
271     }
272 }
```

Listing B.1 – Code source principal

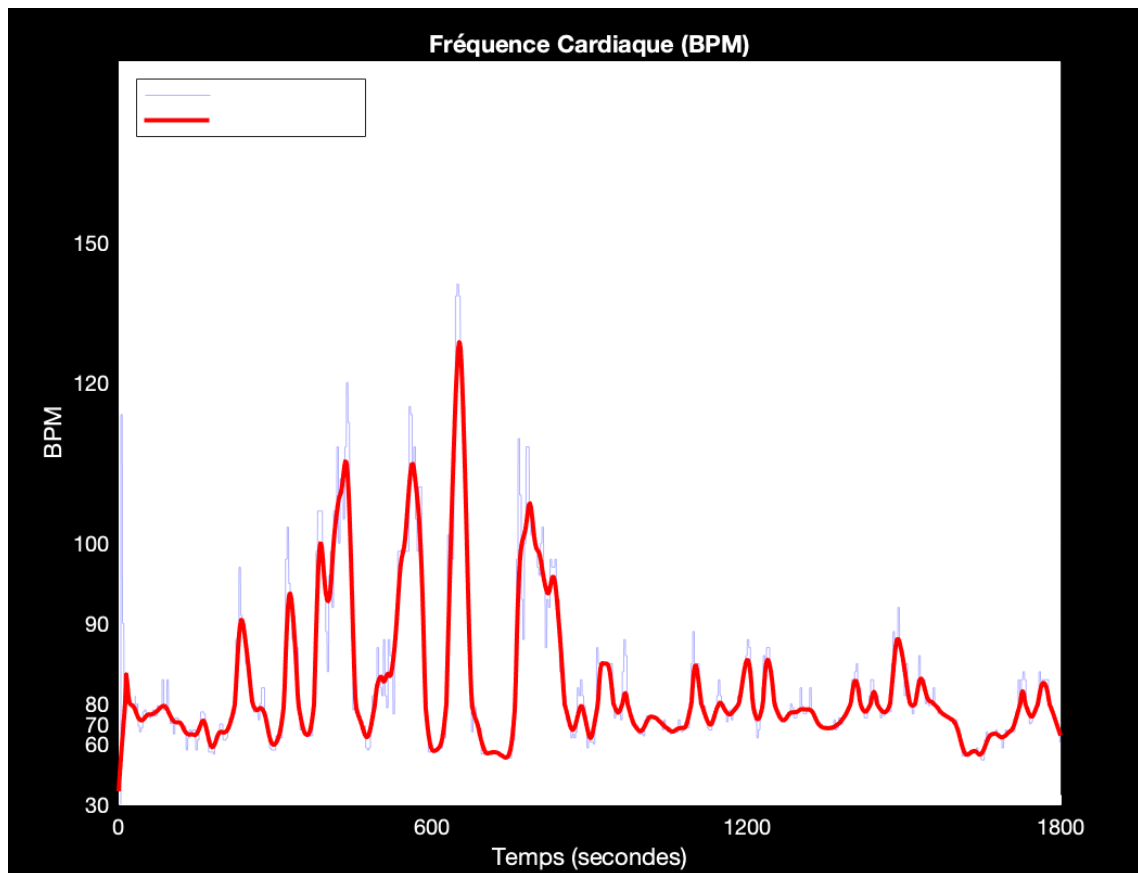
Annexe C

Résultats Expérimentaux

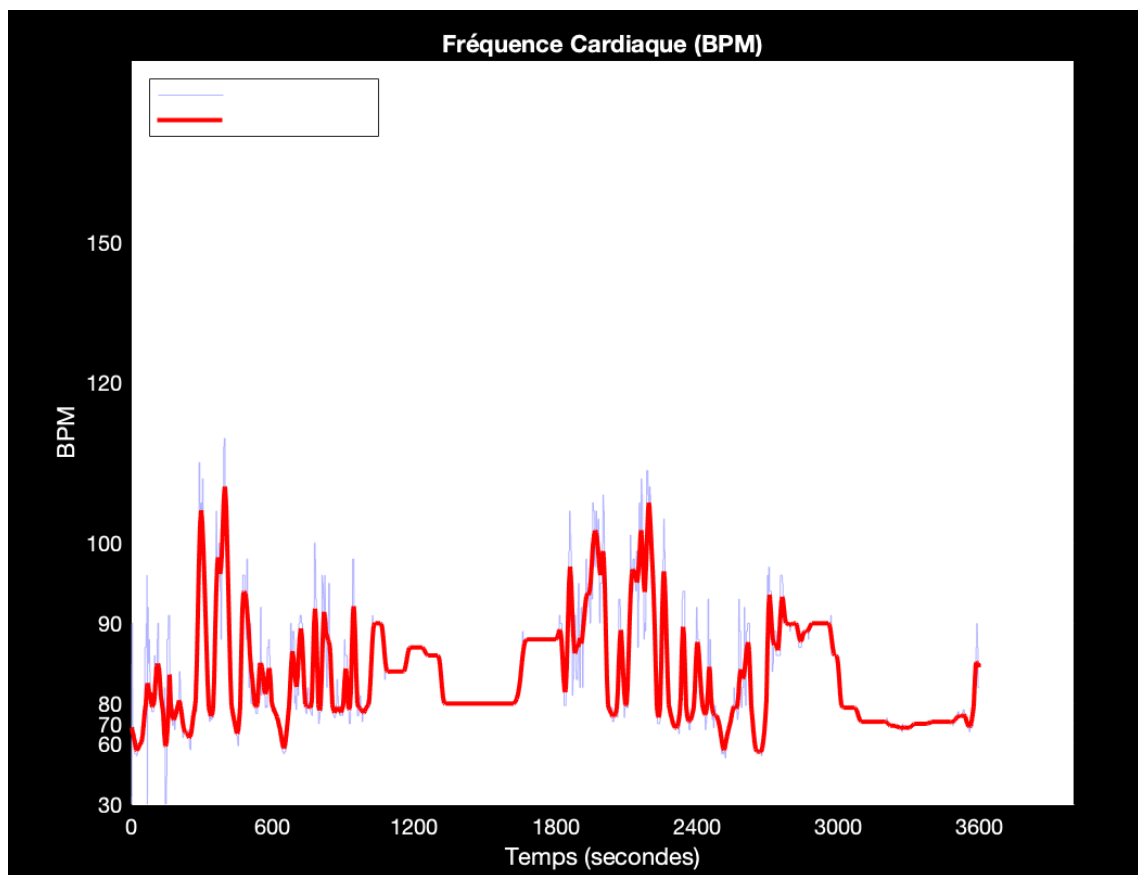
C.1 Données sur 25 minutes



C.2 Données sur 30 minutes



C.3 Données sur 1 heure



Annexe D

Composants Matériels

- **Arduino** : Un microcontrôleur pour exécuter le programme.
- **Capteur MAX30105** : Utilisé pour mesurer la fréquence cardiaque.
- **Capteur ADT7410** : Utilisé pour mesurer la température.
- **Accéléromètre DFRobotLIS2DW12** : Utilisé pour mesurer les mouvements.
- **Module RTC (PCF8523)** : Utilisé pour garder l'heure exacte.
- **Module SD et carte SD** : Utilisés pour stocker les données.
- **LED** : Utilisée comme indicateur visuel.