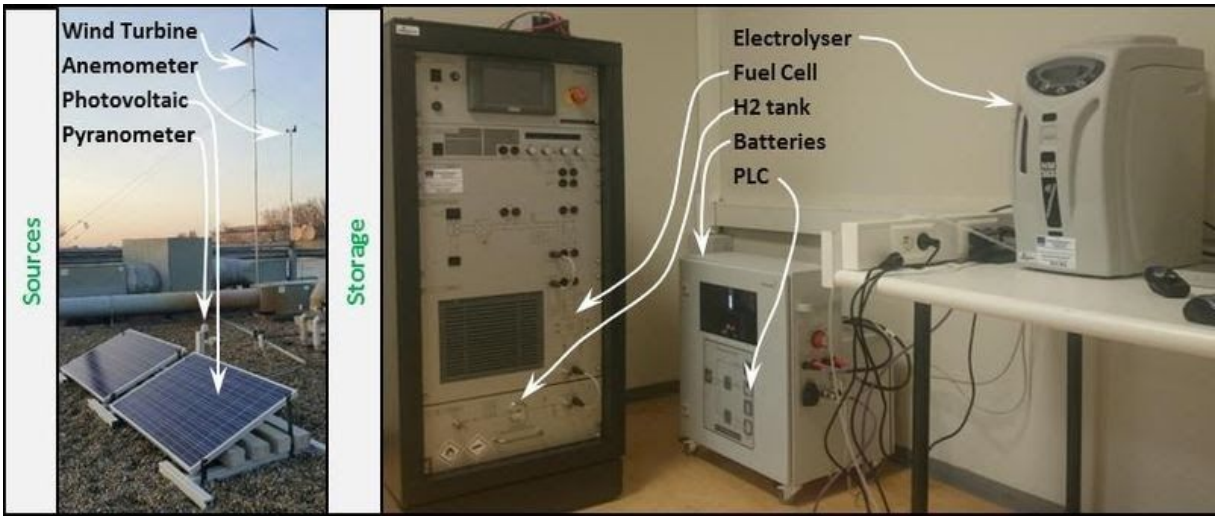
Rapport final de PFE

P13 - Supervision d'une pile à combustible



Tutrice : Anne-Lise Gehin

IMA5
20/02/2020

Antoine Branquart
Juliette Obled

Remerciements

Nous tenons à remercier notre encadrante, Mme. Gehin pour sa gentillesse et sa disponibilité. Nous remercions également M. Conrard pour son aide très précieuse. Merci à M. Vantroys et M. Redon pour leur réactivité ainsi que M. Bressel, M. Granjon et M. Pollart pour leurs connaissances sur le système. Nous remercions aussi M. Flamen, qui nous a conseillé avec beaucoup de bienveillance. Enfin, merci à nos camarades qui n'ont pas hésité à nous apporter de l'aide quand nous en avions besoin.

Sommaire

Introduction

De nos jours, le réchauffement climatique est au centre de toutes les problématiques mondiales. Les ressources fossiles que nous utilisons, en plus de s'épuiser, polluent énormément. Le monde cherche donc à se tourner vers l'utilisation d'énergies renouvelables. Cependant, aussi prometteuses qu'elles soient, les énergies renouvelables (issues du rayonnement solaire, du vent, de l'eau) ne proposent pas d'énergie à la demande car dépendent de nombreux paramètres tels que la météo au moment dit, l'heure de la journée et le moment de l'année. Il se pose donc un problème de stockage de ces énergies pour réussir à se libérer totalement des énergies fossiles. C'est dans ce cadre que s'est constitué le projet européen Electrons to high value Chemical products (E2C). Pour la partie située au sein de l'Université de Lille, le projet consiste à étudier la combinaison de plusieurs énergies renouvelables afin d'en sortir une production la moins variable possible, un stockage du surplus d'énergie vers des batteries et des bouteilles d'hydrogène grâce à l'électrolyse de l'eau.
Les avantages de ce mode de stockage sont les suivants : en l'associant au stockage par batterie disponible dans l'armoire de commande de la plateforme, on obtient un gain non seulement en autonomie mais aussi en disponibilité en énergie.

Dans le cadre de notre dernière année d'étude d'ingénieur dans la spécialité IMA, nous avons donc décidé de prendre part à ce projet. Notre objectif est de développer des algorithmes et implanter une interface de supervision pour gérer de manière optimale les différents modes de fonctionnement d'une pile à combustible.

Tout d'abord, nous développerons comment notre PFE s'inscrit dans le projet E2C, notre cahier des charges et le plan d'action associé. Dans un second temps, nous détaillerons chaque étape du plan d'action, avec notamment l'analyse de l'existant, le travail de récupération des données, la réalisation de l'interface de supervision, l'ajout de fonctionnalités et la mise en place de livrables. Nous ferons ensuite un retour d'expérience sur notre projet en mentionnant les difficultés rencontrées ainsi que les perspectives envisageables sur le système à la suite de notre PFE. Enfin, nous conclurons sur ce que ce projet nous a apporté.

1. Présentation et analyse du projet

1.1. Le projet E2C

L'école, en partenariat avec le laboratoire CRIStAL, dispose d'une plate-forme technologique permettant d'illustrer des enseignements dans le domaine des énergies propres. Comme le montre la Figure 1, cette plateforme est constituée d'une éolienne, de deux panneaux photovoltaïques, d'un électrolyseur, d'une unité de stockage de l'hydrogène et d'une pile à combustible. L'idée est d'utiliser l'énergie produite par les sources renouvelables lorsqu'elles sont disponibles pour produire de l'hydrogène à partir de l'électrolyse de l'eau puis de réutiliser ultérieurement cet hydrogène pour produire de l'électricité via la pile à combustible. Ceci permet alors de pallier l'intermittence des sources primaires.
La réalisation de ce projet de fin d'études s'inscrit dans le projet européen Electrons to high value Chemical products (E2C). Ce projet, commun à plusieurs universités et centres de recherche localisés dans les régions côtières de la mer du Nord, a pour objectif de convaincre l'industrie d'investir dans le développement et la mise en œuvre de technologies qui utilisent les énergies renouvelables pour remplacer les sources d'énergie du pétrole et du gaz dans la production de produits chimiques.

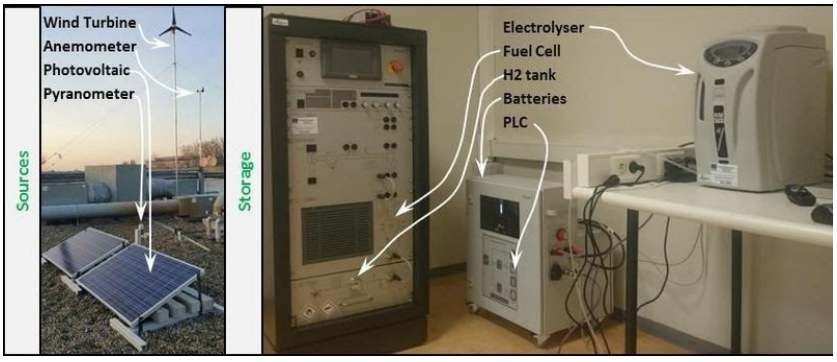


Figure 1. Plateforme E2C à Polytech Lille

L'entreprise *Heliocentris* fournit la majorité des composants ainsi que des interfaces de lecture des données. Cependant, ces logiciels sont non-libres et ne peuvent pas être modifiés. De nouveaux composants comme des capteurs ou même une deuxième éolienne ne peuvent donc pas être ajoutés, et la commande ou la supervision ne peuvent pas être adaptées. Dans le cadre d'un projet de recherche ils ne sont donc pas exploitables.

L'année dernière, l'étudiant François-Xavier Cockenpot a également pu inscrire son PFE dans le projet européen E2C, afin de réaliser l'interface de commande et de supervision de la production de l'hydrogène (électrolyseur). Il a tout d'abord travaillé sur la communication avec l'automate, en récupérant et identifiant les trames grâce au logiciel Wireshark. Une fois les variables récupérées il a automatisé la commande de l'électrolyseur et réalisé l'interface de supervision du système complet avec le logiciel Labview.

1.2. La pile à combustible

L'objectif de notre PFE est donc de continuer sur les pas du PFE 2018 du point précédent. Alors que le système étudié l'année dernière permet donc d'obtenir et de stocker de l'hydrogène à partir de sources d'énergies renouvelables, le système sur lequel nous allons travailler cette année utilisera donc cet hydrogène comme carburant pour la pile à combustible afin d'alimenter la charge électronique et donc de constituer une partie de la plateforme technologique.
En effet, la "PAC" permet d'obtenir de l'électricité à partir de l'oxydation sur une électrode de l'hydrogène couplée à la réduction sur l'autre électrode d'un oxydant, tel que l'oxygène de l'air. Ce fonctionnement revient finalement à l'électrolyse inverse de l'eau. Les équations qui résultent de cette réaction se trouvent à la Figure 2.

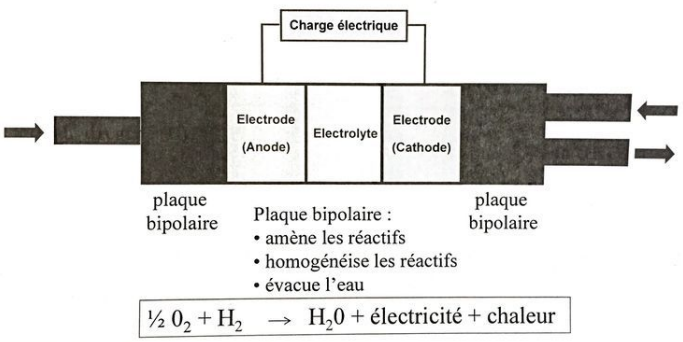


Figure 2. Procédé chimique exploité par la PAC

1.3. Cahier des charges

Lors de ce projet, l'étude est centrée sur la pile à combustible. Pour palier au logiciel de commande et supervision trop contraignant déjà existant, le but principal est de développer une interface plus flexible de l'automate, en reprenant un design similaire à celle d'*Heliocentris* déjà existante (Annexe A1 à la page 27). Si cela est possible, la commande et/ou l'interface seront mises en commun avec le reste du système (PLC, électrolyseur, batteries). L'ajout d'un système externe pourra être mis en place afin de tester l'efficacité du contrôle de la production d'électricité via notre interface, ainsi que d'autres nouvelles fonctionnalités à l'interface.

1.4. Plan d'action

Notre plan d'action pour la réalisation de ce projet se décompose en cinq catégories:

- La préparation du projet
- La récupération des données du système
- La réalisation de l'interface de commande et de supervision à partir de ces données
- L'ajout de fonctionnalités à cette interface
- La réalisation des livrables

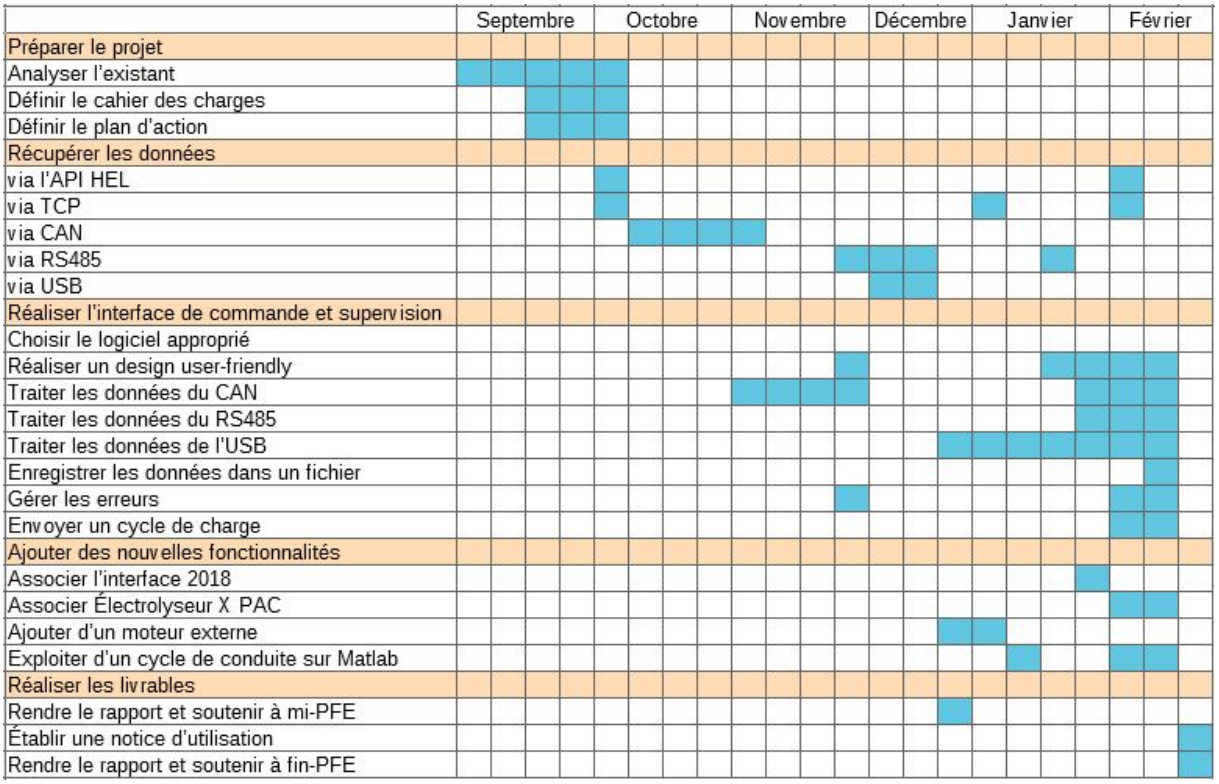


Figure 3. Diagramme de Gantt du projet

2. Réalisation du projet

2.1. Analyse de l'existant et matériel à disposition

2.1.1. Système de pile à combustible

La première étape du projet a été d'analyser ce qui existe pour mieux comprendre le fonctionnement de l'ensemble du système et de faire les bons choix quant aux méthodes à adopter pour réaliser ce projet.

Le système sur lequel porte le projet est composé de plusieurs modules dont une pile à combustible qui dispose du principe de fonctionnement décrit plus haut. L'ensemble de ces modules assemblés compose le système à la figure 4.

Sur la partie la plus basse se trouve un réservoir où l'on dispose les bouteilles d'hydrogène.
En remontant se situe d'abord une pile à combustible, puis un module de gestion de l'énergie. Ce module comporte un convertisseur DC/DC pour soit recharger les batteries soit alimenter une charge continue, et un convertisseur DC/AC.
S'en suit les batteries 24V au plomb.
Enfin les deux modules sur la partie la plus haute sont la charge électronique qui permet de simuler le consommateur et le module de contrôle du système avec l'ordinateur de commande (panneau tactile), l'interrupteur principal et l'arrêt d'urgence.

Figure 4. Système de pile à combustible

Tous ces modules communiquent avec le module de contrôle (panel PC) grâce à différents bus de communications comme le montre l'annexe A3 page 28.
Les informations liées entre autres au module de stockage d'hydrogène sont remontées au panel PC via un bus RS485. Il y a ensuite le bus CAN qui lie les modules gestion de l'énergie et pile à combustible au panel PC. Le bus CAN est le principal bus de communication puisqu'il permet d'effectuer la majorité des commandes et de remonter les deux tiers des variables du système au panel PC. Enfin il reste la communication avec la charge électronique qui s'effectue grâce à une liaison mâle-mâle de type USB.

2.1.2. Exécutable Heliocentris

Le panel PC qui est lié aux modules via les différents bus décrits précédemment contient un exécutable, fourni par la société ayant produit le système. Il est en charge de la supervision et du contrôle de ce dernier.

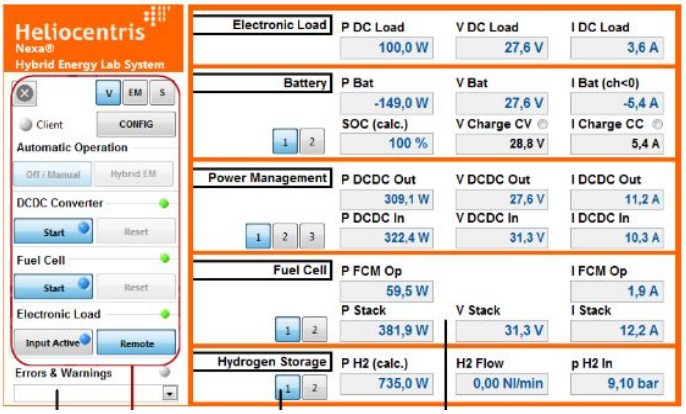


Figure 5. Exécutable du panel PC

On retrouve les différentes valeurs des variables du système (courant, tension, puissance, débit) ainsi que la possibilitée de contrôler la charge électronique, la pile à combustible ainsi que le convertisseur DC/DC du système.
Un logiciel utilisateur sur PC externe a été livré avec le système afin de piloter et de commander ce dernier, et de paramétrer les modules à distance. Son interface est disponible à l'annexe A1 page 27.

Nous nous inspirons de cet exécutable pour reproduire une interface souple et libre d'accès. De ce fait elle pourra évoluer et s'optimiser en y ajoutant des fonctionnalités indisponibles sur les logiciels HEL, ciblées selon le projet ou l'application voulu.

2.1.3. Système de production d'hydrogène

Le projet de François-Xavier Cockenpot réalisé l'année dernière porte sur le système hybride de production d'hydrogène qui est composé de :

- Deux panneaux photovoltaïques et les capteurs associés
- Une éolienne et les capteurs associés
- Une armoire de commande disposant d'une batterie, d'un onduleur et d'un automate
- Un électrolyseur

Ce système est à disposition pour notre projet et son principe de fonctionnement est simple. Il récupère l'énergie produite par les sources d'énergies renouvelables afin de produire de l'hydrogène en alimentant l'électrolyseur. L'électrolyseur peut être alimenté soit par la batterie de l'armoire de commande soit par les sources d'énergies renouvelables. La production d'énergie est automatisée grâce à l'automate de l'armoire de commande qui fait en sorte qu'il y ait toujours de l'énergie pour alimenter l'électrolyseur.
Un exécutable pour visualiser les différentes variables est fournie par Heliocentris avec le système (Voir annexe A2 page 27).

Pour réaliser son projet François-Xavier a également commandé du matériel qui a été laissé à notre disposition pour le projet.

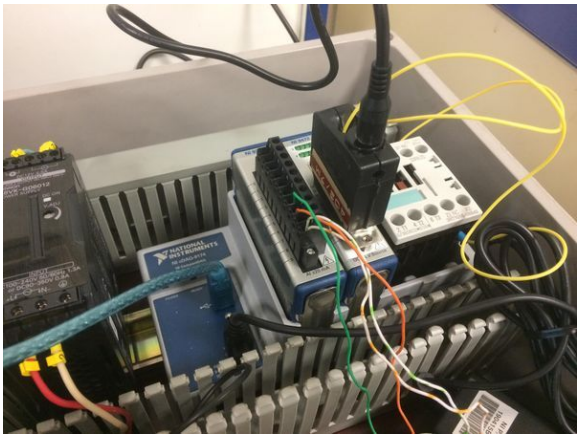


Figure 6. Châssis compactDAQ

Ce matériel est un châssis CompactDAQ (figure XX) relié à un PC via un câble USB. Il est équipé de deux modules d'E/S conditionnés qui fournissent une connectivité directe aux capteurs. Les deux modules sont:

- Le module 9203, un module d'entrée de courants à 8 voies
- Le module 9472, un module numérique à 8 voies

Le châssis disposant de 4 ports, deux modules d'E/S peuvent donc être encore installés pour notre projet.

2.2. Récupération des données

Dans un premier temps, l'objectif est de récupérer toutes les valeurs des différentes variables qui composent l'interface fournie par Heliocentris afin de les réutiliser dans notre propre interface sur Labview. Plusieurs pistes sont alors étudiées.

2.2.1. API Heliocentris

Pour cela nous nous sommes orientés vers l'interface de programmation disponible dans l'onglet outils du logiciel HEL.

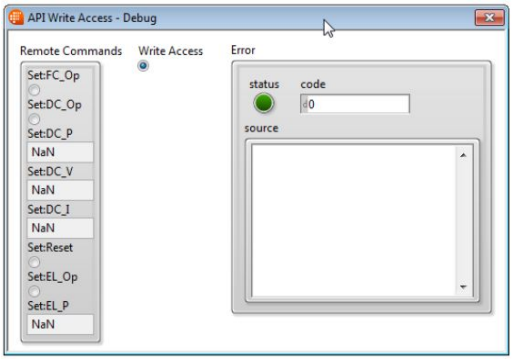


Figure 7. API disponible sur l'executable HEL.exe

Il y a deux accès au niveau de cette API:

- Un accès en lecture qui affiche les différentes valeurs de variables du système.
- Un accès en écriture qui traite l'envoi de commandes.

Il s'avère qu'il existe une application "HET_Remote.vi" à titre d'exemple sur Labview avec le code source fournie par Heliocentris.

Le "HET_Remote.vi" utilise des variables PSP. Ce sont des variables partagées qui sont publiées sur un réseau au moyen du protocole NI-PSP (NI Publish-Subscribe Protocol) par un autre programme Labview par le biais d'un URL qui permet à d'autres programmes Labview de récupérer les valeurs associées à ces variables partagées afin de les exploiter.
Le réseau en question dans notre cas est la liaison établie entre le panel PC et notre PC grâce à un câble ethernet. Cependant, en lançant l'API en lecture et en écriture nous n'avons pas réussi à récupérer les valeurs de ces variables partagées. La liaison entre le panel PC est notre PC était pourtant bien établie. En effet, sur le logiciel HEL de notre PC, elle est nécessaire pour accéder aux valeurs et commande du système, et la connexion était bien établie.

Le problème semblerait venir de l'accès aux variables qui ne se réalise pas à cause d'un mauvais URL ou bien un fichier manquant. Au vu du manque de documentation sur cette API et à l'accès à seulement quelques variables, d'autres pistes plus prometteuses sont étudiées.

2.2.2. Trames TCP

Une des idées initiale était, de la même manière que François-Xavier en 2018, de récupérer les données via le port Ethernet de la PAC. En effet, les variables qui transitent sur le réseau CAN, RS485 et USB sont centralisées sur le panel PC qui les envoi ensuite de manière continu au PC via la liaison TCP. Cependant, le contenu des trames est non documenté ce qui nécessite de faire un rétro-engineering. De plus, le protocole de communication n'est pas de type requête/réponse, ce qui permettrait de faire correspondre les valeurs que l'on modifie sur le système avec celles que l'on reçoit, ce qui complique la tâche.

Après un travail de reverse engineering mené en isolant chaque type de connexion, le résultat est le suivant : ces trames ne sont pas exploitables.
En effet, les données du CAN sont trop nombreuses (une trentaine) et non isolables, sans mentionner que nous n'avons pas réussi à retrouver le type de conversion entre les données de la trames et les valeurs réelles.
Pour les données issues du RS485 et celles de l'USB, alors que les paquets associés aux variables transmises sont facilement identifiables, l'analyse des paquets montre une nécessité de connaître plus amplement le type de conversion utilisé par HEL, qui n'est pas divulgué par l'entreprise.

Un document regroupant l'ensemble des tests menés et détaillant beaucoup plus le travail fourni pour conclure de tels résultats est repris dans la partie 2.5 (page 23).

M. Mathieu Bressel de l'équipe E2C nous a alors conseillé de travailler sur les communications entre le système et le panel PC directement, soit le CAN, le RS485 et l'USB. La difficulté réside dans le fait de devoir réussir à établir ces trois connexions, qui sont complémentaires, pour pouvoir avoir l'ensemble des valeurs sur l'interface.

2.2.3. via CAN

Un des protocoles principal du système est le CAN. Comme le montre la photo ci-dessous, il y a trois ports CAN In sur le système, dont un libre (celui en fin de chaîne). Il est donc possible de récupérer les valeurs du bus CAN sans perturber le panel PC.

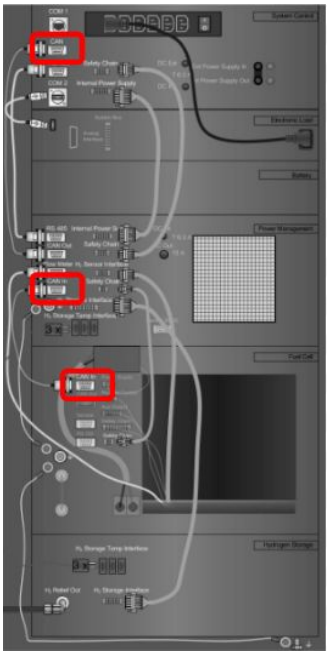


Figure 8. Ports de communication du bus CAN à l'arrière de la PAC

Pour sniffer cette communication CAN, la première solution est d'utiliser un shield arduino. Cependant cette solution prend énormément de temps (commande du matériel, réalisation du code, etc). Heureusement, Polytech dispose de PC sniffeur CAN dans la salle C001, que nous avons utilisé l'année dernière dans le cadre des TP Réseaux industriels. Ce PC comporte le logiciel NSi527 qui permet d'établir une communication CAN et tester la récupération/envoi des trames de données.

L'annexe A4 à la page 29 montre le résultat du sniffeur CAN. On retrouve bien les trames de la documentation, qui sont exploitables. En comparant les données reçues sur le logiciel NSi527 et sur le PC panel, on comprend que ce sont des données en hexadécimal signé. Il suffit de convertir en décimal puis éventuellement de diviser par un multiple de 10 pour retrouver les valeurs affichées sur le panel PC.

Pour pouvoir recevoir et envoyer des trames sur un logiciel de supervision, le module d'interface CAN NI-9862 (National Instrument) a été commandé.

Cependant, il s'avère que l'interface NI-9862 nécessite une alimentation continue entre 9 et 30V (consommation d'1W) en entrée, et le bus CAN de la pile ne la fournit pas. Un câblage a été réalisé à partir de câbles de type D-SUB afin de pouvoir récupérer la tension délivrée par l'alimentation du châssis CompactDAQ en entrée du module NI-9862. De cette manière, le bus CAN du système est relié au module d'interface tout en alimentant la carte d'acquisition.

L'interface NI-XNET Bus Monitor (Figure 6) permet facilement de vérifier la réception et envoi des données de la même manière qu'avec le sniffer CAN.

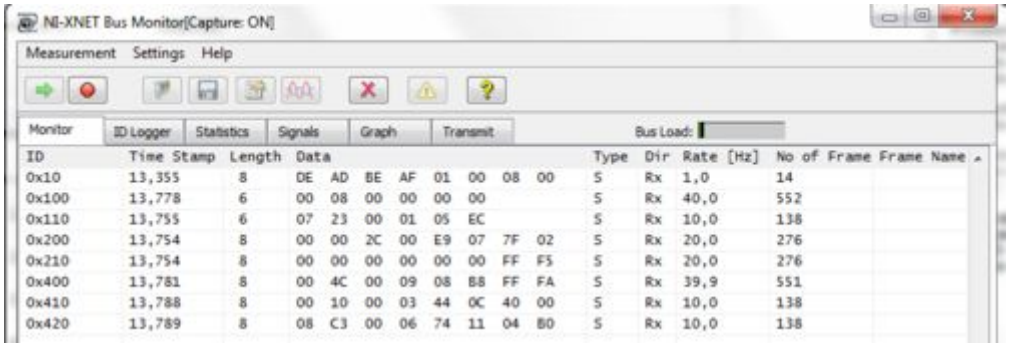


Figure 9. réception des trames CAN sur le logiciel NI-XNET

Ci-dessous le montage final du châssis CompactDAQ avec en bas à droite l'arrivée du bus CAN.

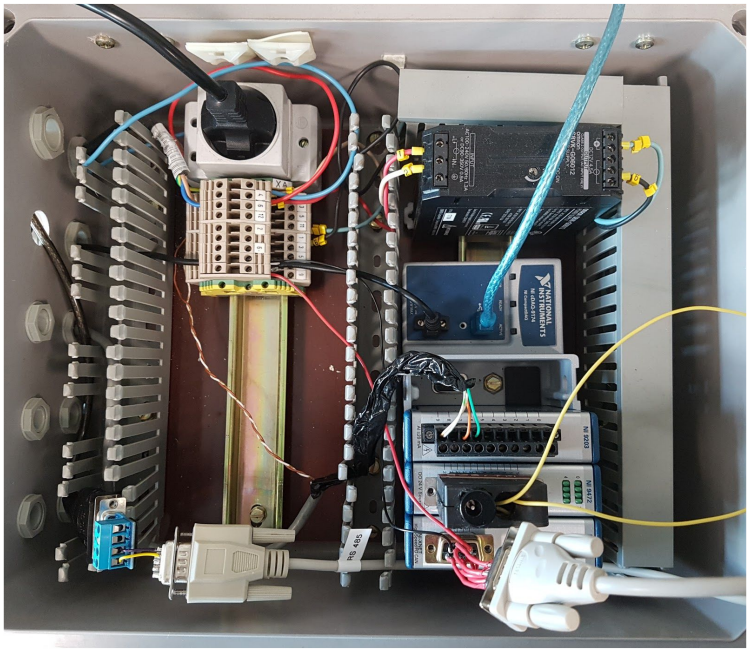


Figure 10. Montage de la carte d'acquisition NI9862

2.2.4. via RS485

La liaison RS485 permet de remonter des données de mesures supplémentaires comme le débitmètre, la température des réservoirs d'hydrure, etc. Toutes ces données proviennent de capteurs indépendants du système qui ont des sorties analogiques en 4..20mA ou 0..10V qui sont ensuite envoyées vers le module de mesure *ICPCON* 7019R qui est relié au panel PC par le bus RS485.

Figure 11. Module I-7019R

Dans un premier temps, nous avons mis en place un diviseur de bus qui permet de connecter le bus RS485 via un convertisseur RS485 vers USB à notre PC et de conserver la liaison entre le PC panel et le module I-7019R. L'idée était de pouvoir communiquer et recevoir les données des capteurs tout en laissant l'affichage des données sur le panel PC, de la même manière que le CAN. Cependant, après plusieurs semaines de travail, nous avons n'avions toujours pas de résultats de connexion valide.

La configuration du module I-7019R a été vérifiée depuis le logiciel DCON utility lancé sur le panel PC, qui permet la configuration du module. Après une étude du protocole de communication *DCON*, nous avons établi une requête permettant de récupérer toutes les valeurs analogiques des différents capteurs connectés au module. Cette requête a été envoyée depuis le terminal *PuTTY* au module toujours depuis le panel PC et les valeurs ont bien été récupérées.

Les mêmes actions ont ensuite été réalisées sur notre PC sans essayer de diviser les connexions, en utilisant le convertisseur RS485 vers USB, ce qui a fonctionné (figure 12). Certaines valeurs (comme les températures) n'ont besoin d'aucun traitement mais d'autres (comme le débit d'H2) ont nécessité un travail de rétro-engineering pour retrouver leur valeur réelle.

L'idée de diviser le bus est abandonnée et notre interface devient donc une alternative au panel PC.

Figure 12. Réception des données RS485 sur le terminal PuTTY

2.2.5. via USB

La liaison USB entre la charge électronique et le panel PC permet à ce dernier de commander la charge ainsi que de récupérer - entre autres - les valeurs de courant, tension, puissance dans la charge. La liaison est possible grâce à l'interface IF-U1 (ci-contre) dont dispose la charge électronique. Cette interface permet de détecter la charge comme un port COM virtuel lorsqu'elle est connectée à un PC.

Pour réaliser l'action de récupération des données via USB nous nous sommes tout d'abord placé dans une configuration où la liaison entre le panel PC et la charge reste disponible et où notre PC est connecté entre la charge et le panel PC. Après plusieurs échecs (port RJ45 non adapté à la communication, hub USB, ...) nous avons compris qu'il n'était pas possible que deux PC différents soient connectés en même temps à la charge.

Grâce à un sniffer USB *Beagle 12* fourni par M. Vantroys, on peut lire le bus de communication sans avoir d'impact sur celle-ci. On retrouve sur le logiciel associé *Total Phase Data Center* toutes les trames circulant entre le panel PC et la charge, et on reconnaît alors des requêtes de la documentation de l'IF-U1.

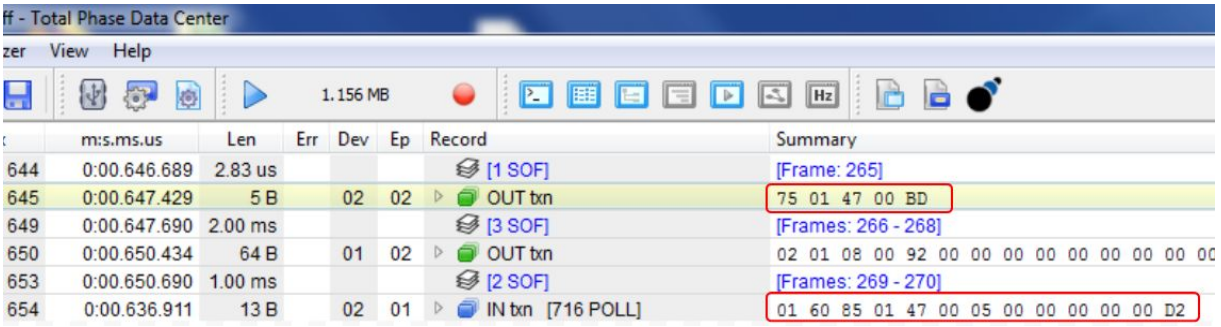


Figure 13. Réception des trames USB sur le logiciel Total Phase Data Center

En connectant ensuite uniquement la charge à notre PC, le logiciel *Serial Port Monitor* est utilisé pour envoyer des requêtes sur le bus et en voir les réponses, qui correspondent avec celles lues sur le *Beagle 12*.

L'utilisation de notre interface se place donc dans une configuration où le panel PC ne pourra pas utiliser les données provenant du RS485 et de l'USB, mais les données du CAN seront toujours disponibles à la fois sur le panel PC et sur le PC.
Toutes les communications étant désormais établies sur le PC, l'action récupération des données est terminée. L'action suivante est donc d'utiliser ces données pour réaliser l'interface de commande et de supervision.

2.3. Interface de commande et de supervision

2.3.1. Choix du logiciel

Le but étant de réaliser une interface de commande, une analyse sur les différents logiciels a été réalisée, et notamment Labview et Matlab ont été retenu. Matlab propose de meilleurs modules de commandes et de simulation, mais Labview est plus intuitif et possède de meilleures fonctionnalités dans la récupérations des données et le design d'interface. De plus, Labview a été utilisé pour réaliser l'interface de 2018 donc une mise en commun est plus pratique sur ce logiciel. Des possibilités de symbiose Matlab/Labview existent pour profiter des meilleurs points de chacun, mais celles-ci sont chères et non indispensables. Il est donc important de savoir que cela est possible, mais la solution choisie est l'utilisation du Logiciel Labview uniquement.

2.3.2. Code Labview

Le VI Labview (code) est donc séparé en plusieurs parties comme le montre la figure ci-dessous.

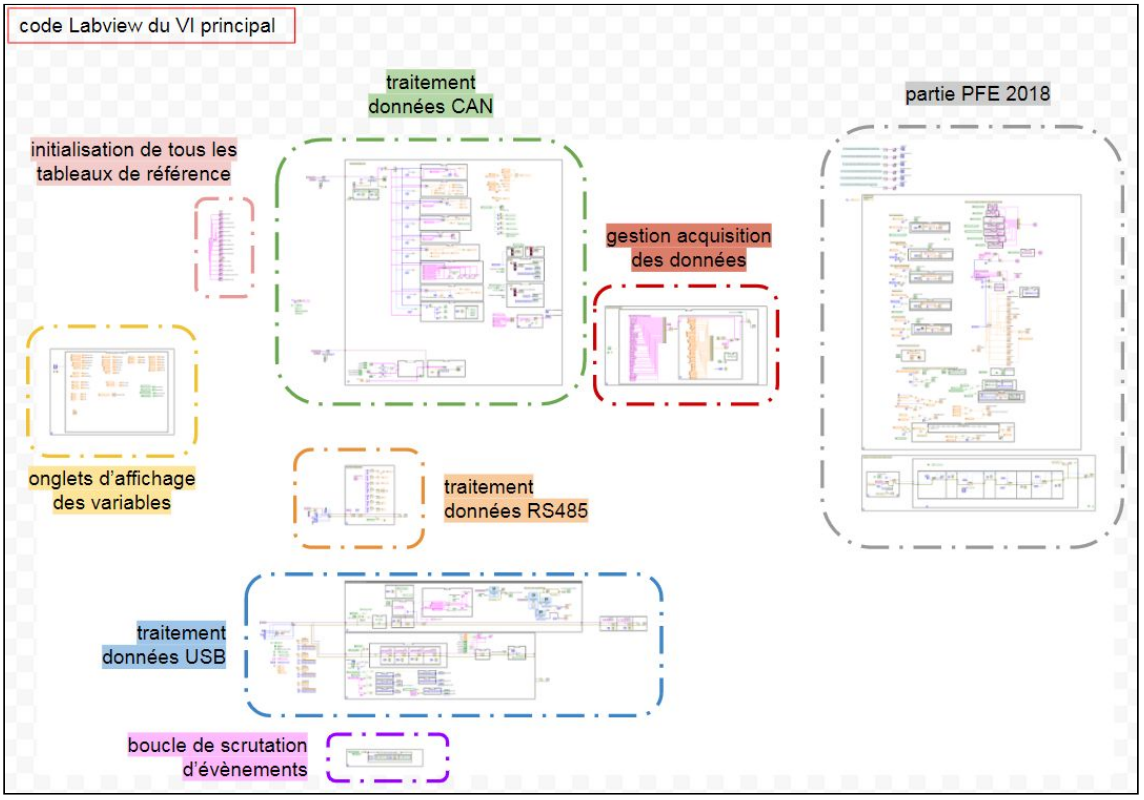


Figure 14. code Labview du VI principal

La partie PFE 2018 reprend le code de l'élève François-Xavier Cockenpot avec quelques modifications de boucles. L'onglet d'affichage regroupe l'ensemble des indicateurs qui s'actualisent dans leur boucle de traitement associée, afin de s'afficher dans l'onglet voulu sur la face-avant. La gestion d'acquisition de données permet d'enregistrer de manière périodique les valeurs de la vue *Aperçu du système* dans un fichier .csv à la demande de l'utilisateur.

Le traitement de données CAN est détaillé dans un schéma similaire à la figure 14, à l'annexe A5 page 29. Il comprend un module de connexion NI-XNET CAN en lecture et un en écriture. Celui en écriture réagit à l'appui de boutons de commande. Celui en lecture établit une lecture continue des trames, les trie selon leur adresse puis retrouve les valeurs des variables associées (établi depuis la documentation).

Le traitement des données RS485 est constitué d'un module de connexion VISA (détails sur l'annexe A6 page 30). Il récupère les données sur son bus après avoir envoyé une requête spécifique de lecture. S'en suit un traitement peu complexe pour retrouver les valeurs réelles.

Le traitement des données USB est plus complexe que les deux premiers, et est toujours détaillé sur l'annexe A7 page 30. Il commence par une connexion VISA au port COM4 (port de la charge) qui fait alors tourner deux boucles. La première boucle, après une initialisation reçoit (après requête en conséquence) en continu les valeurs actuelles d'état, de courant, de tension et de puissance. Dans cette même boucle se situe la connexion au module DAQ Assistant de la partie PFE 2018.
Dans la deuxième boucle dédiée à la charge, se situe toute la gestion de commande de changement de mode, de valeur de charge, etc. Une partie de cette commande est régie par la bouche de scrutation d'événement. La boucle contient également la possibilité d'envoyer un cycle de charge, c'est à dire un step ou un cycle de conduite (voir la partie 2.4.4. page XX).

2.3.3. Design user-friendly

La face avant de l'interface a été travaillée de telle sorte à être intuitive, agréable à utiliser et empêcher les erreurs, notamment en grisant des boutons lorsque leur utilisation est inadéquate, à utiliser des leds d'état ainsi que des menus déroulants et des onglets pour éviter la surcharge d'information.

La figure 15 montre le résultat final, avec :

1. Zone de connexion et déconnexion
2. Zone d'arrêt (à utiliser avant la déconnexion)
3. Zone de commande
4. Vues :
   - Vue Aperçu du système
   - Vue Pile à combustible
   - Vue Power management monitor
5. Zone de messages d'erreurs et d'avertissements
6. Zone d'acquisition des données

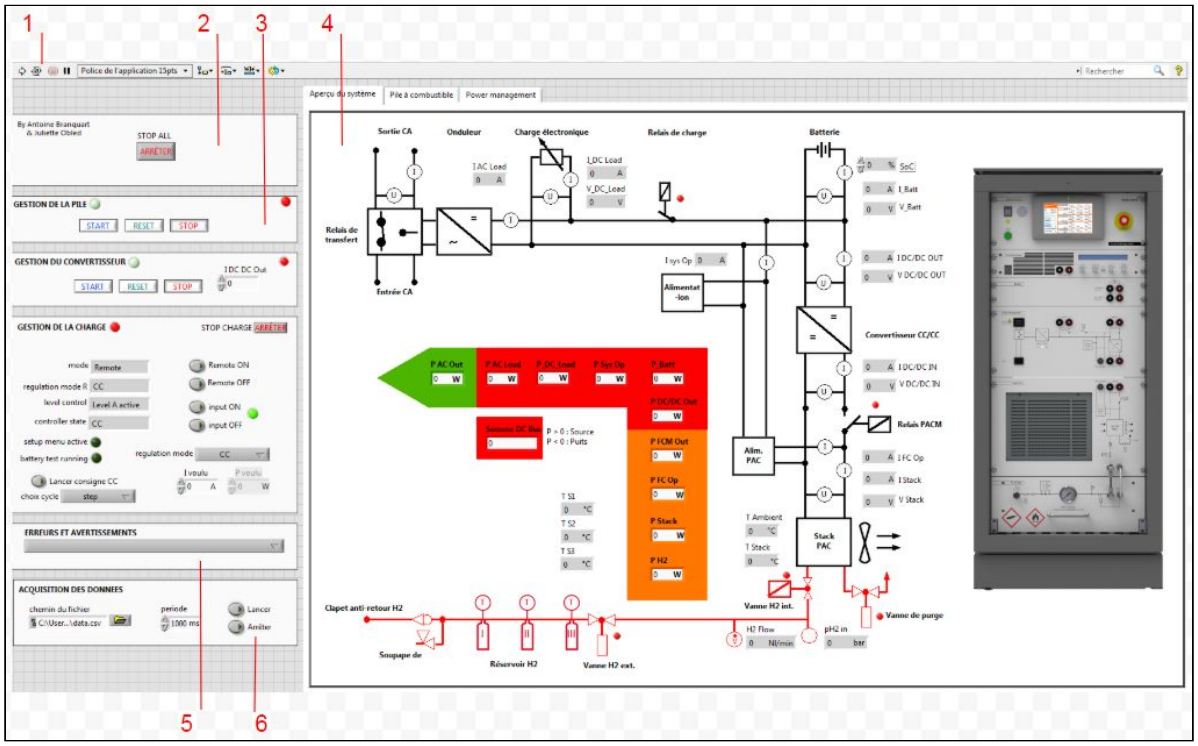


Figure 15. Ensemble de l'interface Labview

2.4. Ajout de nouvelles fonctionnalités

2.4.1. Interface PFE 2018

Pour réaliser l'action d'ajout de nouvelles fonctionnalités à notre interface, nous nous sommes orientés tout d'abord vers la possibilitée de lier notre interface Labview développée sur le système de pile à combustible avec celle développée l'année dernière sur le système hybride de production d'hydrogène.

L'objectif est de n'avoir qu'un seul programme Labview pouvant superviser les deux systèmes, ce qui n'est pas possible avec les interfaces *Heliocentris* qui ont leur propre exécutable non-modifiable.

Les modifications apportées au code labview de François-Xavier pour qu'il puisse être utilisé dans notre diagramme Labview se résument à des mises à jour de driver, des simplifications et suppressions de certaines boucles qui empêchaient les deux interfaces de fonctionner correctement. Des paramètres de connexion au châssis compactDAQ ont également été ajouté car ils manquaient sur le programme que nous avions récupéré.

L'intégration sur la face-avant du programme Labview a été mise en place grâce à la création d'onglets qui permettent d'avoir plusieurs visuels sur une même face-avant.

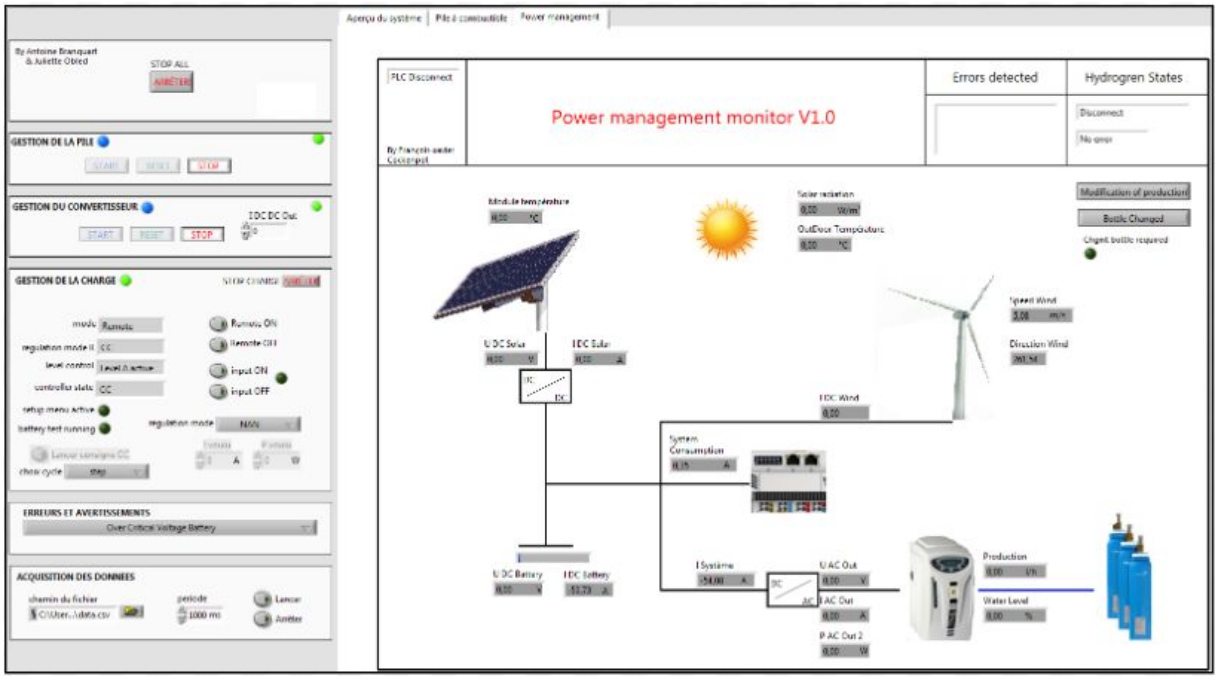


Figure 16. Face avant de l'interface avec intégration du Power management monitor

2.4.2. Electrolyseur et la PAC

L'électrolyseur est le seul système qui n'est pas contrôlé ou supervisé. En effet le travail réalisé l'année dernière ne permet pas de contrôler ni de visualiser les erreurs liées à l'électrolyseur mais seulement de visualiser deux valeurs de variables (H2 production et pourcentage de volume d'eau restant).
Pour l'ouverture du projet, l'électrolyseur reste donc une piste mais contrôler ce système serait réellement utile si l'intervention humaine pour le changement de bouteilles dans l'électrolyseur n'était pas nécessaire. Cela permettrait réellement d'automatiser les recharges des bouteilles.
C'est pour cette raison que nous avons testé la recharge des bouteilles directement en connectant l'électrolyseur au module de stockage d'hydrogène de la pile à combustible plutôt qu'aux bouteilles sorties du module directement.

Ce n'avait jamais été testé et pourtant la recharge s'effectue bien. On évite donc l'intervention d'une personne pour sortir les bouteilles du module de stockage.

2.4.3. Moteur externe

Dans nos premières idées de nouvelles fonctionnalités, nous pensions ajouter une charge externe plus adaptée aux étudiants ou aux personnes qui ne connaissent pas le système de la pile. Les piles à combustible sont aujourd'hui principalement employées dans le secteur des transport et notamment de l'automobile Cela permettrait de voir plus facilement comment une pile à combustible permet de lancer un moteur par exemple.
Cependant, alors que sur papier cette idée paraît intéressante, il faudrait en réalité installer un variateur de vitesse entre le moteur et le bus DC/DC du système, qu'il faudrait réussir à contrôler. Pour garder l'idée de la voiture électrique sans les contraintes pratiques que cela engendre, nous pouvons toujours utiliser la charge électronique déjà présente sur le système pour simuler le moteur.
Nous avons donc laissé de côté l'idée du moteur réel pour plutôt travailler sur une simulation de celui-ci.

2.4.4. Cycle de conduite Matlab

La charge électronique étant commandable depuis l'interface Labview, le but est de lui envoyer une consigne de type cycle de conduite.
Un cycle de conduite est un test s'appliquant aux voitures entrant sur le marché pour déterminer leur consommation en carburant ainsi que leurs émissions de substances polluantes. Ces tests sont effectués sur des bancs à rouleaux et sont censés simuler des conditions de conduite plus ou moins réalistes. Différents tests existent, notamment le ECE (cycle urbain), NEDC (nouveau cycle européen de conduite) et WLTP (Procédure d'essai mondiale harmonisée pour les voitures particulières et véhicules utilitaires légers) qui durent respectivement 3 minutes, environ 20 minutes et environ 30 minutes.

Cependant, ces cycles imposent une vitesse afin de connaître la consommation/autonomie du véhicule. Pour notre charge, nous pensons imposer un courant et il faut donc travailler sur une conversion vitesse vers courant.

Pour cela, nous réalisons une REM (Représentation Energétique Macroscopique) d'un véhicule électrique, qui permet de mettre sur papier les différents éléments composants un système (vision globale) tout en respectant la causalité des événements. Elle permet ensuite de réaliser une SMC (Structure Maximale de Commande) qui correspond à la REM inverse afin de retrouver la commande du système pour une valeur de sortie souhaitée. Grâce à cela, on retrouve le courant à imposer à la charge associé à une vitesse. Ci-dessous le courant (à gauche) associé à la vitesse (à droite) du cycle ECE.

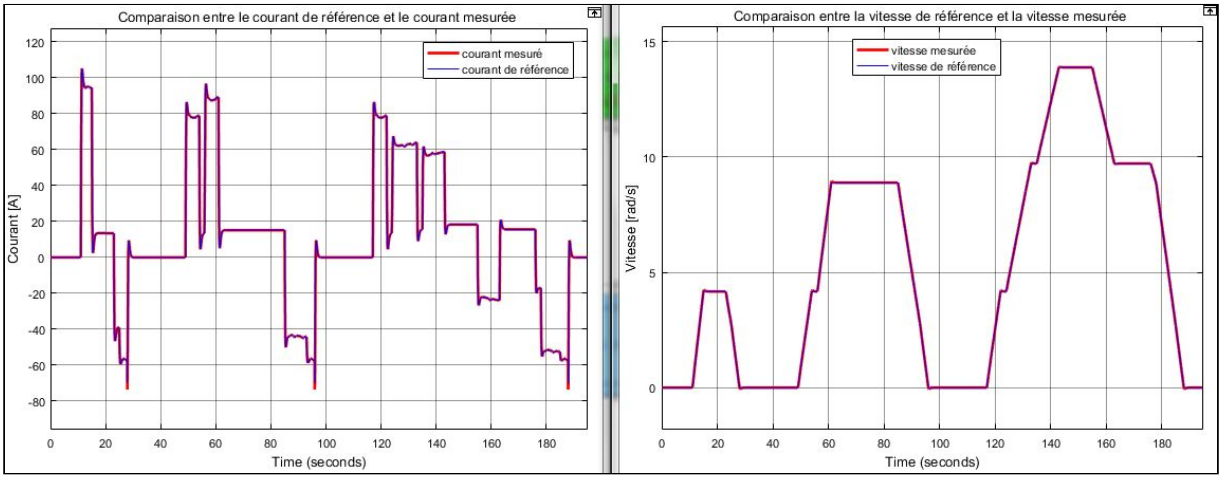


Figure 17. consignes en courant et en vitesse associées au cycle de conduite ECE

A l'annexe A8 page 31 se trouve le simulink Matlab correspondant. Une documentation est également disponible reprenant la mise en oeuvre de ce simulink, avec notamment les équations utilisées. Elle est mentionnée plus en détails dans la partie 2.5. page 23.

La charge ne supportant pas de valeurs de consigne négative, la valeur de courant est légèrement modifiée avant d'être imposée à la charge depuis Labview. Les cycles ECE, NEDC et WLTP sont donc disponibles en tant que consigne en courant sur l'interface finale.

2.5. Rédaction de livrables

Pour finir, la dernière partie du plan d'action est la mise en place de documentation. En effet, le but principal de notre projet est d'établir une interface ouverte qui peut mener à des améliorations et ajouts constants. De ce fait, quel est l'intérêt de proposer une interface à première vue flexible mais sans la possibilité de comprendre son fonctionnement et la modifier ? C'est dans cette optique que nous avons réalisé plusieurs livrables.

Tout d'abord, comme mentionné dans la partie 2.2.2., il y a le document reprenant l'ensemble des tests menés sur les trames TCP qui est disponible à l'url suivante :
https://www.cjoint.com/c/JBtjDpMA0cN

Ensuite, également mentionné mais cette fois dans la partie précédente 2.4.4., le document expliquant le simulink Matlab utilisé pour la REM et SMC d'un véhicule électrique. Ce document est disponible à cette adresse :
https://www.cjoint.com/c/JBtjESYP3ZN
L'ensemble des livrables ainsi que les fichiers Matlab utilisés pour établir les cycles de conduite sont mis en évidence sur la page wiki du projet dans la section *Livrables*. :
https://projets-ima.plil.fr/mediawiki/index.php/IMA5_2019/2020_P13#Livrables

De plus, le code Labview pouvant être très indigeste (pas de dézoom possible), nous avons schématisé et annoté le code de manière générale (figure 14 à la page 17) et sommes rentrés dans les détails pour chaque communication (annexes A5, A6 et A7 aux pages 29 et 30) afin que la compréhension du code soit plus facile. Ces schémas sont également disponibles dans la partie *Livrables* du wiki.

Enfin, un des documents qui servira certainement le plus est la notice d'utilisation. Cette notice reprend l'ensemble du système, les configurations possibles et le matériel à disposition. Il explique comment démarrer chaque module, dans quel ordre ainsi que la connexion à l'interface Labview. Il reprend également toutes les variables et possibilités de commande que l'interface Labview offre. Cette notice est disponible sur le wiki ainsi que sur ce lien :
https://www.cjoint.com/c/JBtjPNU1ARN

3. Retour d'expérience

3.1. Analyse critique

Les résultats obtenus présentés tout au long du rapport répondent bien au plan d'action défini en première partie. Cependant, nous allons revenir sur plusieurs problèmes rencontrés durant ce projet.

Tout d'abord, nous pensions mettre seulement quelques semaines pour réaliser l'action de récupération des données du système de pile à combustible. En s'inspirant du projet de François-Xavier, nous avions mis en place la même démarche de sniffing de communication TCP. En revanche, nous n'avons pas eu le même succès, dû à la complexité de retrouver des valeurs dans notre configuration. En effet, contrairement au système PAC, le système de production d'hydrogène utilise le TCP/modbus qui permet d'accéder directement depuis *Wireshark* aux valeurs contenues dans les registres utilisés.

Le second problème a été de vouloir diviser les bus de communication USB et RS485 dans la même optique que celle du CAN afin de pouvoir connecter notre PC entre les différents modules tout en gardant la configuration de base du système (avec le panel PC). Ces bus de communications ne peuvent avoir qu'un seul maître, cette configuration n'est pas réalisable. Cela nous a fait perdre du temps en milieu de projet avant de prendre la décision de définir notre interface comme une alternative au panel PC et donc de ne pas diviser ces bus de communication.

De nombreux soucis matériels ont également ralenti le projet.
Le convertisseur DC/DC ne démarrait plus et générait constamment une erreur. Ce n'est qu'en fin de projet, une fois après avoir eu une réponse d'un contact de *Heliocentris*, que nous avons pu réinitialiser la configuration du convertisseur pour le faire fonctionner normalement. La décharge des batteries nous a également ralenti ainsi que la mise en défaut régulière de la sécurité électrique sur le système. La cause de ce dernier problème reste inconnu à ce jour.

Enfin, quelques imperfections sur l'interface sont notifiables. Deux sur les dix valeurs de puissance n'ont pas été retrouvées. Ceci s'explique par l'absence de documentation quant au calcul de ces puissances à partir des données récupérées. Nous n'avons également pas trouvé la bonne conversion (valeur du bus vers valeur réelle) de deux valeurs sur les sept transitant sur le bus RS485.
Cependant, ces erreurs représentent seulement quatre valeurs faussées sur la cinquantaine lue et affichée sur l'interface, sans compter l'ensemble des messages d'erreurs qui sont tous correctement récupérés.
D'autres points positifs sont tout de même à retenir puisque le code de notre interface est libre d'accès, souple, avec la partie commande de la PAC, du convertisseur DC/DC et de la charge. Ces points permettent d'ailleurs de s'ouvrir sur de nouvelles perspectives et améliorations de l'interface.

3.2. Perspectives

Notre contribution à ce projet dans le cadre du projet E2C peut donc amener de nouvelles possibilités.

Dans un premier temps, nous avons volontairement enlevé quelques fonctionnalités pour le contrôle de la charge. Cela permet de coller à l'interface d'Heliocentris, de ne pas surcharger la partie commande de la charge (objectif interface agréable à utiliser) et dans le même temps limiter les erreurs. Cependant, il est important de noter que ces fonctionnalités, telles que le choix d'autres modes de régulation (contrôle en tension, contrôle en résistance) ou le choix du level, peuvent être ajoutées. Elles ont en effet été testées et approuvées pendant le projet.

De plus, il est toujours intéressant de pouvoir associer au système une charge réelle comme un moteur, comme mentionné dans la partie 2.4.3 à la page 21. Comme cela a été expliqué, nous nous sommes penchés sur l'étude de cycles de conduite appliqués à la charge électronique mais une étude plus profonde de la mise en place d'un système contrôleur de vitesse et moteur reste envisageable.

La partie 2.4.2 à la page 21, résume ce qui a été fait au niveau de la démarche d'association entre le système de pile à combustible et l'électrolyseur. Ceci ouvre de très nombreuses possibilités d'automatisation de la recharge des bouteilles, en fonction de la production d'énergie des sources renouvelables.
Cette perspective est de loin la plus intéressante puisque, en finalisant la supervision de l'ensemble des systèmes PAC, électrolyseur et armoire de commande, ces systèmes seraient totalement autonomes.
Les algorithmes de contrôle pourront être réalisés à partir de Labview et s'ajouter à notre interface.

Pour réussir à créer cette interface et cette association, il sera sans doute utile d'abord de réaliser une interface pour la supervision de l'électrolyseur uniquement, et pour cela d'analyser l'interface du système qui communique via une liaison RS232.

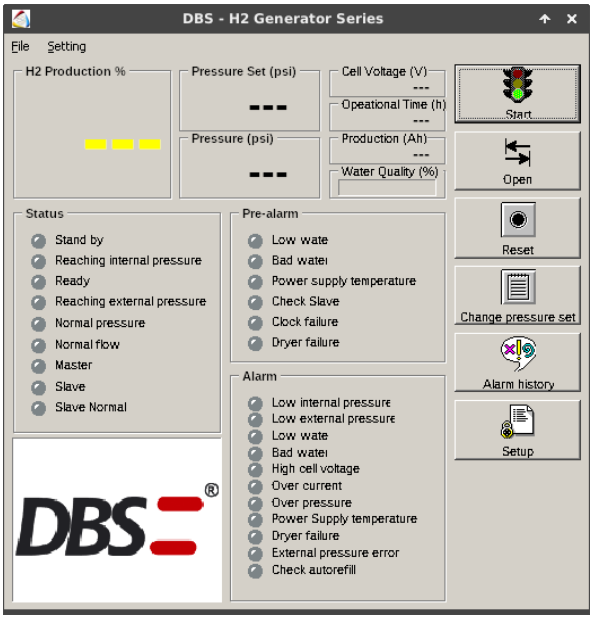


Figure 18. Interface de contrôle de l'électrolyseur

Conclusion

Ce projet de supervision d'une pile à combustible nous donne donc l'opportunité d'intégrer un projet européen qui s'inscrit dans les problématiques énergétiques actuelles. Le travail se focalise sur une pile à combustible et l'enjeu du stockage de l'hydrogène appliqué à l'utilisation des énergies renouvelables.

L'objectif de notre PFE de dernière année d'ingénieurs a été de réaliser une interface de supervision ouverte du système pile à combustible mise à notre disposition afin d'en gérer différents modes de fonctionnement. Nous avons pu voir tout au long de ce rapport que nous avons réussi à mener à bien ce projet, et qu'il s'ouvre à de nombreuses nouvelles perspectives prometteuses.

Ces plusieurs mois de PFE ont également été très formateurs.
La partie technique d'abord, où nous avons appris à coder sur Labview, ré-utiliser les connaissances sur la REM et sur les communications CAN et série. Les notions d'automatisme, énergétique et réseau abordées durant notre cursus ont donc pu être complétées. Nous avons eu l'opportunité de développer une vraie interface user-friendly, ce que nous n'avions jamais réellement fait et nous a donc beaucoup plu. De plus, avoir un impact sur un projet tel que le projet E2C est très gratifiant, puisqu'il contribue à une problématique majeure du monde d'aujourd'hui.
Mais ce PFE nous a aussi permis de développer une vraie démarche projet. En effet, le même projet sur plusieurs mois nous apprend à revoir notre manière de travailler et notre gestion du temps. En tant que binôme, nous avons su coordonner notre travail et être très autonomes, notamment sur la deuxième partie du PFE. Enfin, toujours dans cette démarche projet, documenter notre étude tout au long des deux semestres nous a permis de gagner beaucoup de temps pour établir les livrables, tout en apportant de nombreuses ressources aux futures personnes sur le projet.

Annexes

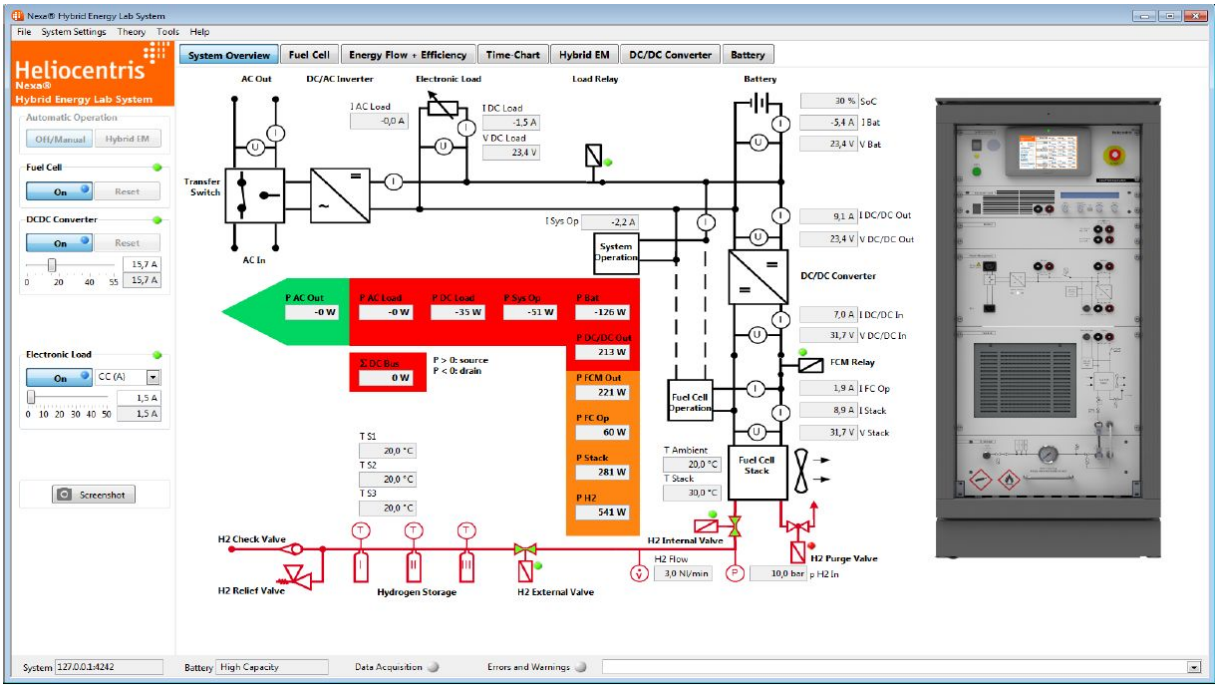


Figure A1. Logiciel Heliocentris (HEL) de la PAC

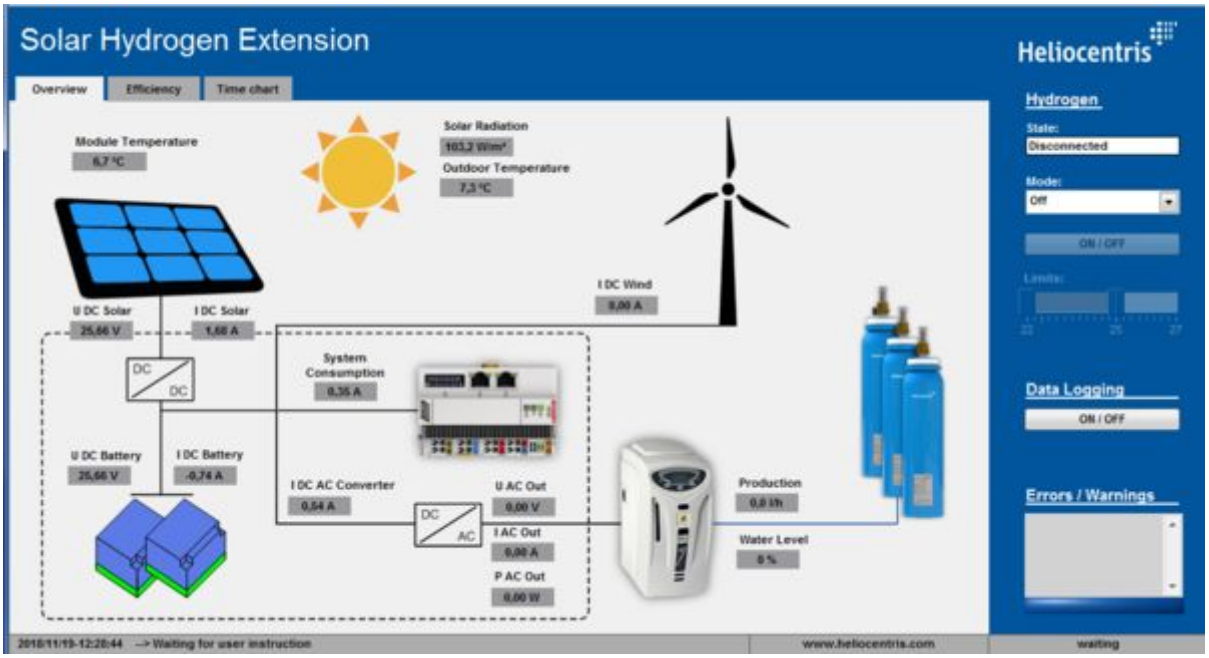


Figure A2. Exécutable de l'interface du système de production d'hydrogène

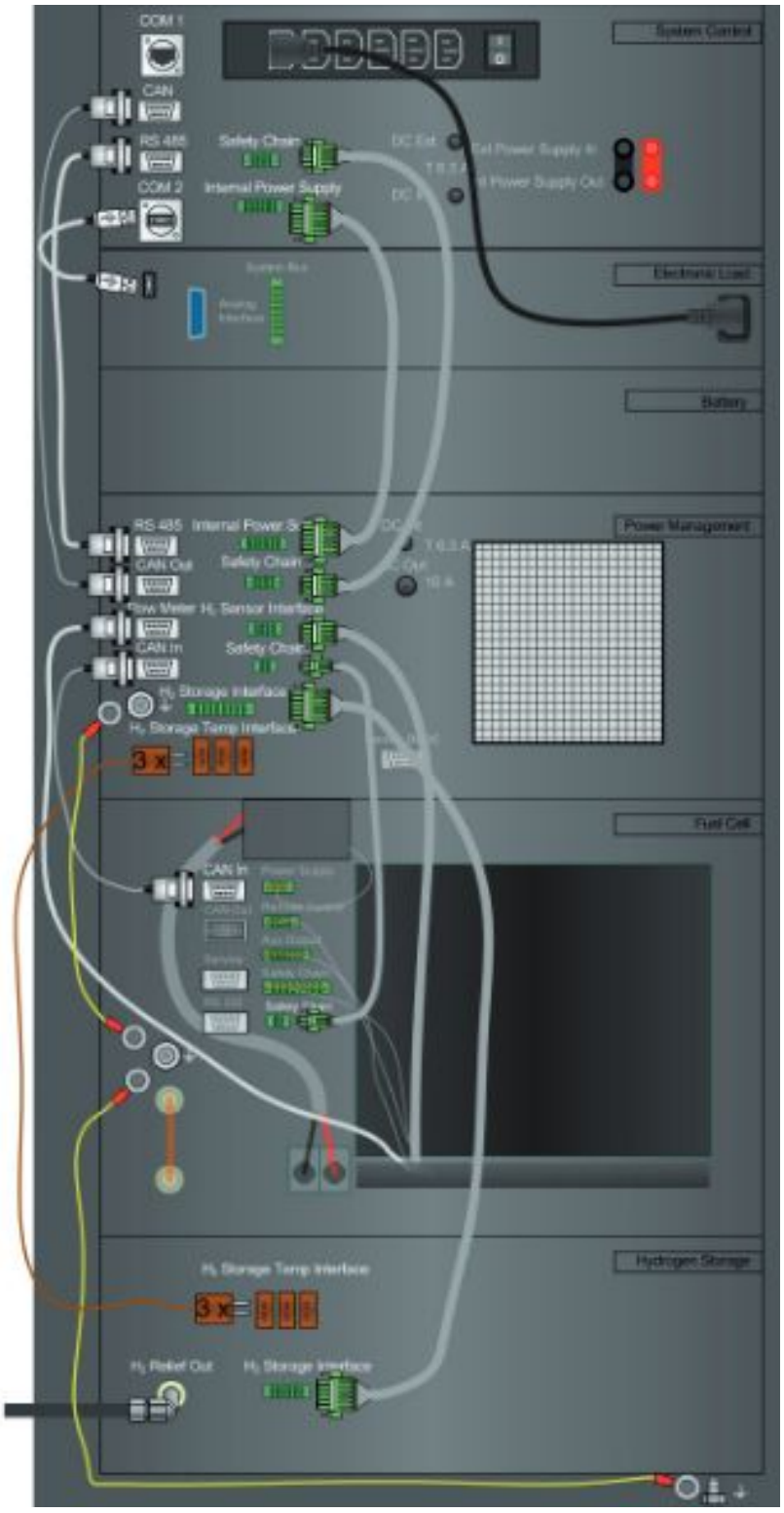


Figure A3. Face arrière du système de pile à combustible

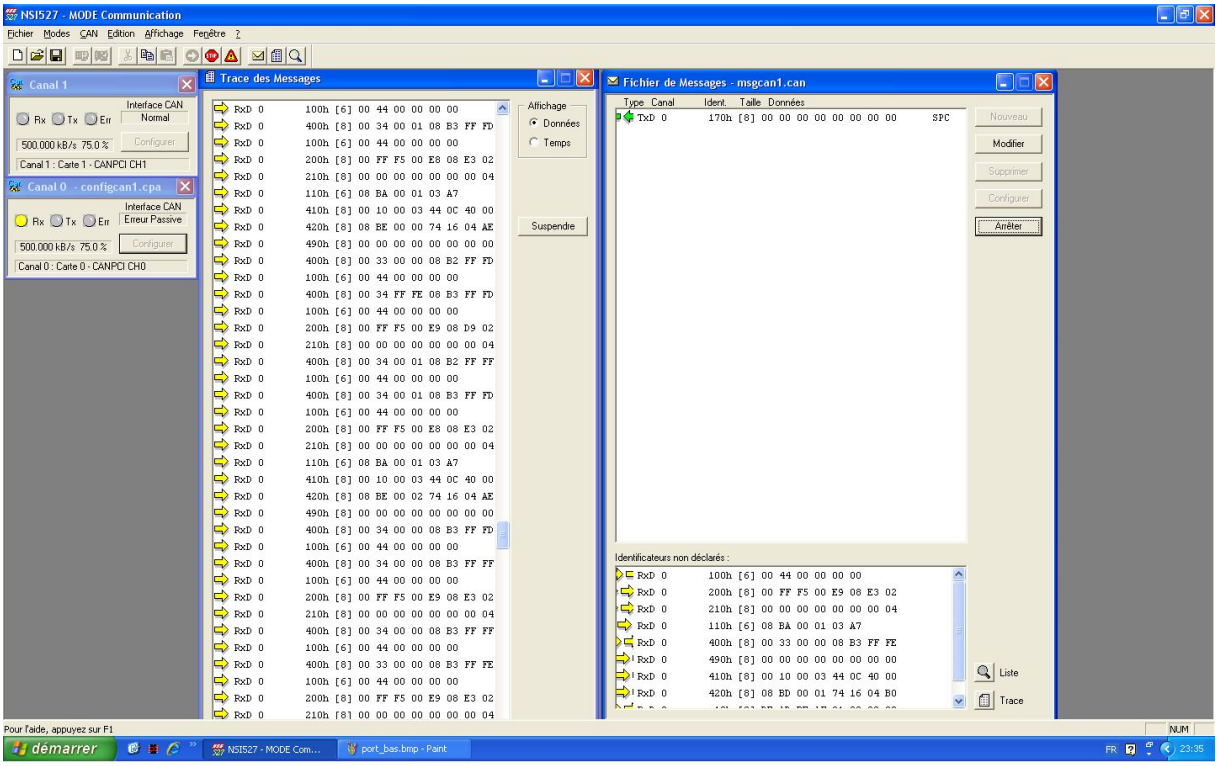


Figure A4. Trames issues du sniffeur CAN

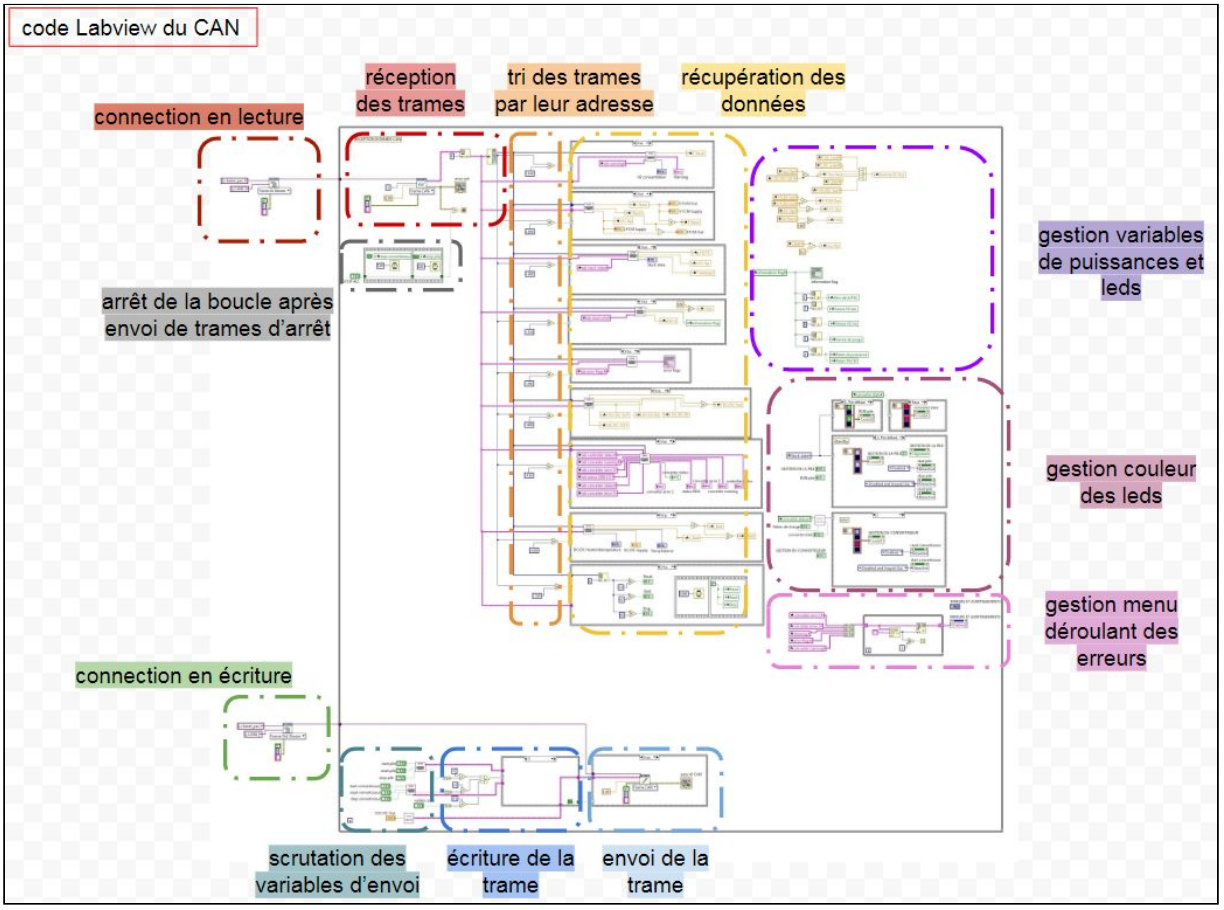


Figure A5. partie CAN du code Labview

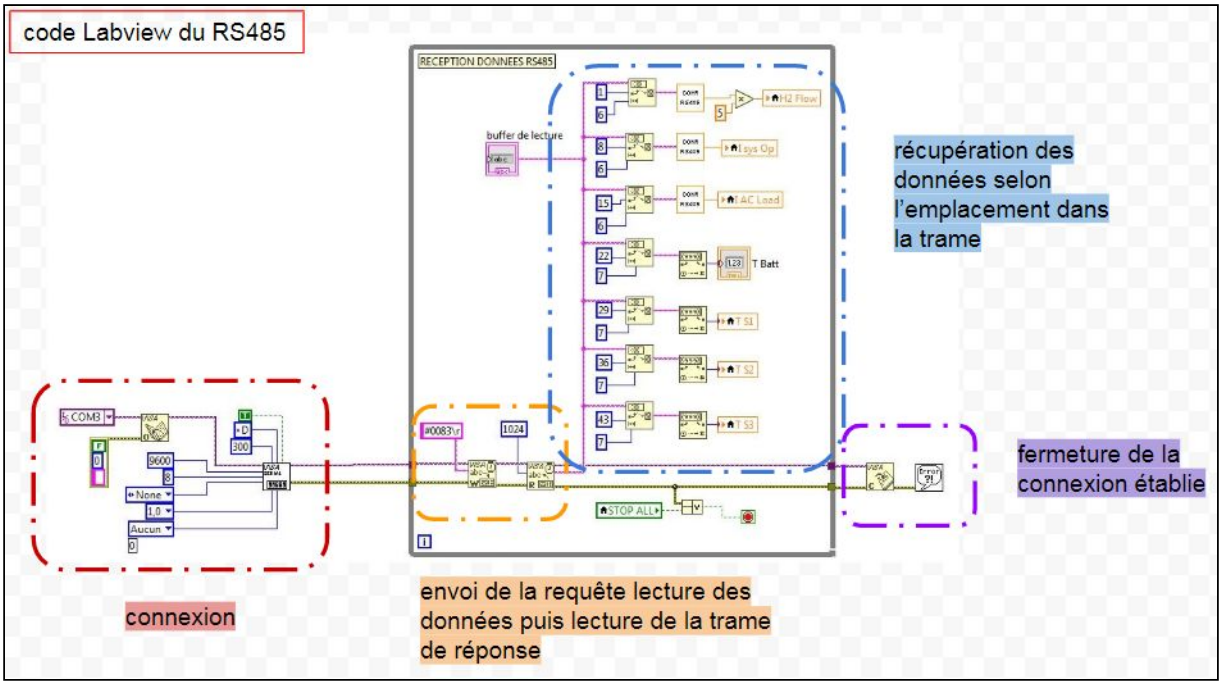


Figure A6. partie RS485 du code Labview

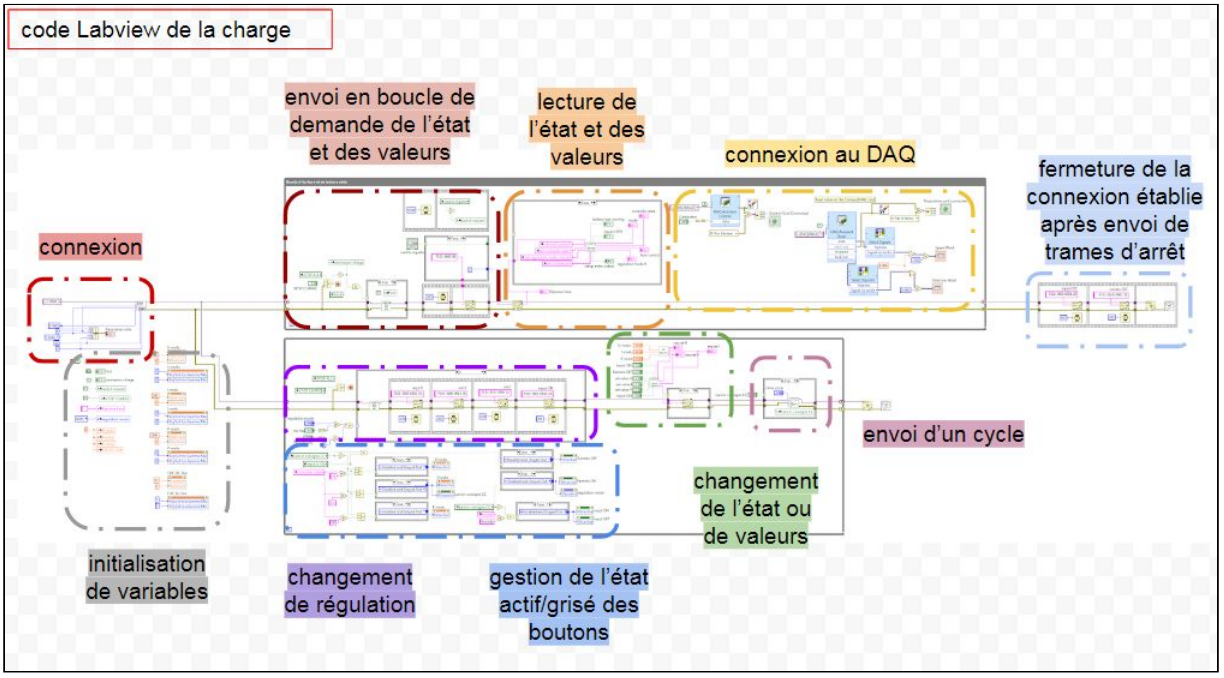


Figure A7. partie USB du code Labview

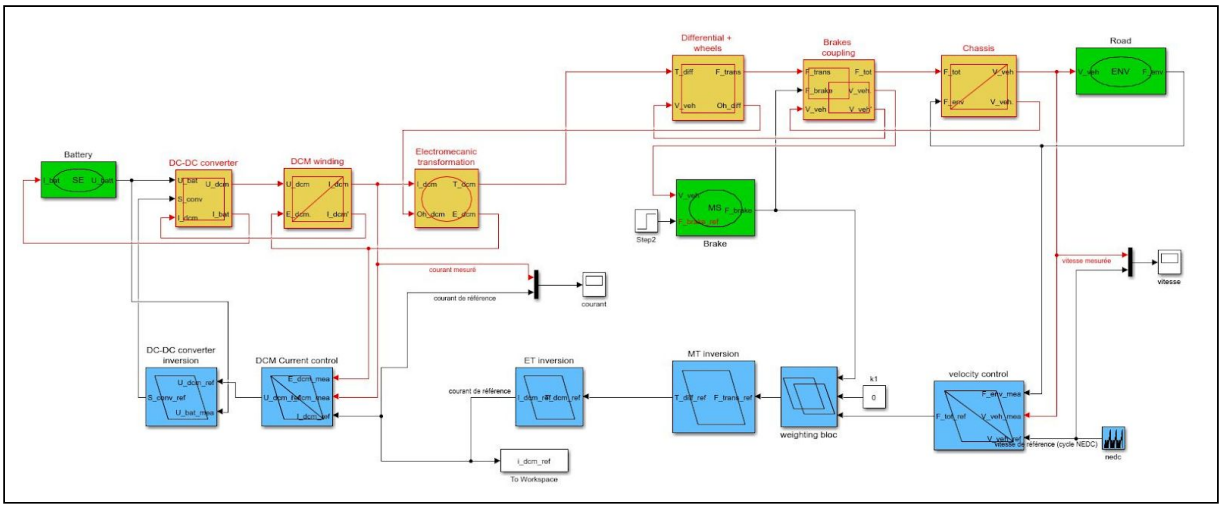


Figure A8. simulink Matlab de la REM et SMC d'un véhicule électrique