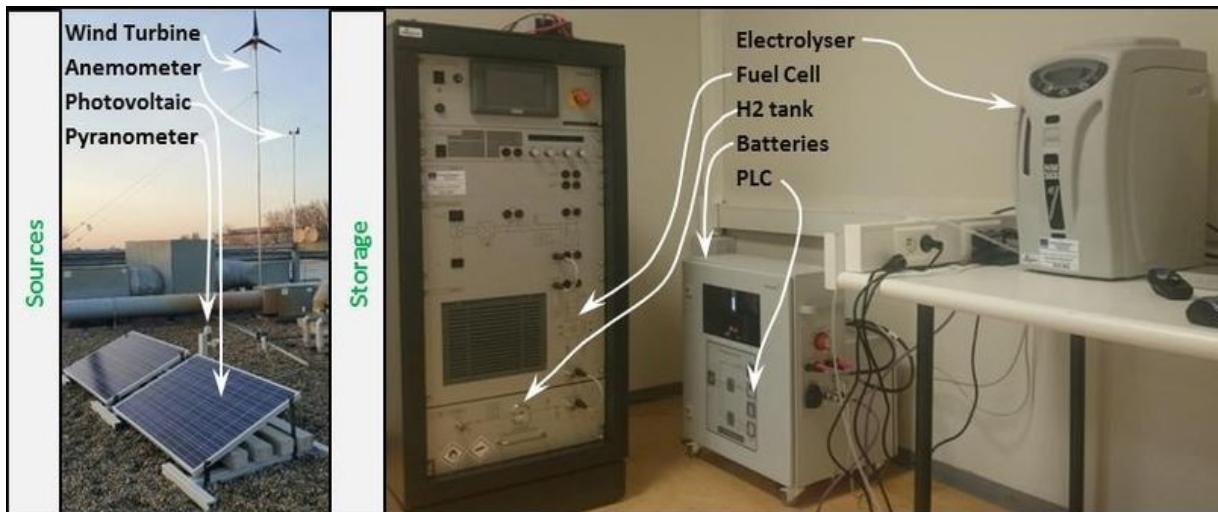


Rapport intermédiaire de PFE

P13 – Supervision d'une pile à combustible



Tutrice : Anne-Lise Gehin

IMA5
18/12/2019

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	4
1. Présentation et analyse du projet	5
1.1. Projet E2C et état de l'art	5
1.1. La pile à combustible	6
1.2. Cahier des charges	6
2. Communication avec la PAC	7
2.1. Ethernet impossible	7
2.2. Protocole CAN	8
2.3. Communication via USB	10
2.4. Liaison RS485	12
2.5. Interface Labview	13
3. Ce qu'il reste à faire	15
3.1. Récupération du reste des données	15
3.2. Ajout d'un système externe	16
3.3. Association PAC / électrolyseur	16
Conclusion	17
Annexes	18

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 planter 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 et l'élaboration de notre cahier des charges. Dans un second temps, nous expliquerons ce que nous avons déjà pu faire durant les premières semaines de projet, c'est-à-dire les différents types de communication étudiés et utilisés entre la pile à combustible et le PC ainsi que l'interface réalisée. Enfin, nous verrons ce qu'il nous reste à faire afin de mener à bien notre PFE.

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 Manche, 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.

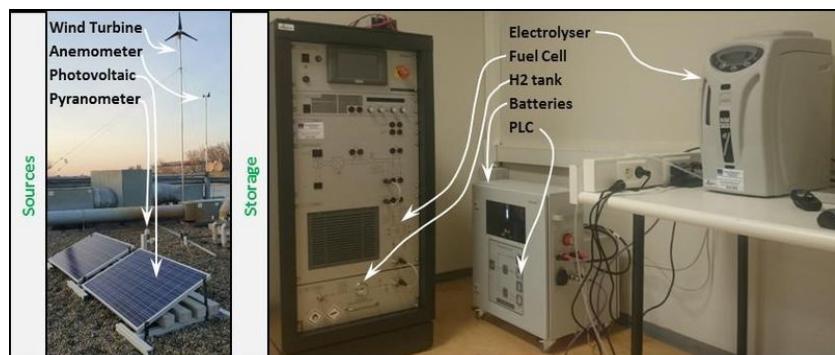


Figure 1. Plateforme E2C à Polytech Lille

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.

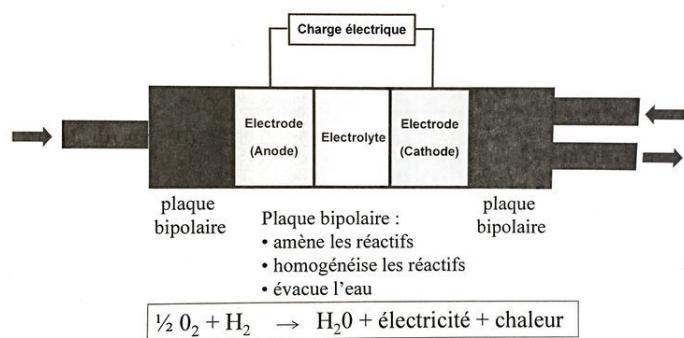


Figure 2. Procédé chimique exploité par la PAC

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 18). 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.

Dans un premier temps, le travail sera concentré sur la récupération de données de la PAC. Ensuite, une interface de commande et supervision sera réalisée sur Labview. Enfin, de nouvelles fonctionnalités seront ajoutées, et un travail de mise en commun du système sera effectué.

Nous pouvons donc entrer dans le vif du sujet, soit la récupération des données de la pile à combustible sur un ordinateur.

2. Communication avec la PAC

2.1. Ethernet impossible

Notre première idée était, de la même manière que François-Xavier l'année dernière, de récupérer les données via le port Ethernet de la PAC. En effet, ce port est utilisé pour communiquer avec le logiciel PC HEL et concentre donc la totalité des données.

Cependant, nous nous sommes vite rendu compte à l'aide de l'analyseur de paquet Wireshark que cette solution était incorrecte. L'automate contrôlant l'électrolyseur communiquant via une liaison TCP/IP avec le logiciel SHE d'Heliocentris utilise le protocole de communication Modbus, ainsi via Wireshark François-Xavier avait pu repérer les adresses des registres utilisés dans l'automate mais aussi les valeurs de ces registres lorsque l'automate répondait. La PAC utilise un protocole propriétaire créé par Heliocentris. Comme le montre la Figure 4, le port de destination n'est pas 502 comme la communication Modbus mais 4242 soit un protocole propriétaire. Ces données ne sont pas exploitable sans documentation permettant de comprendre les trames échangées.

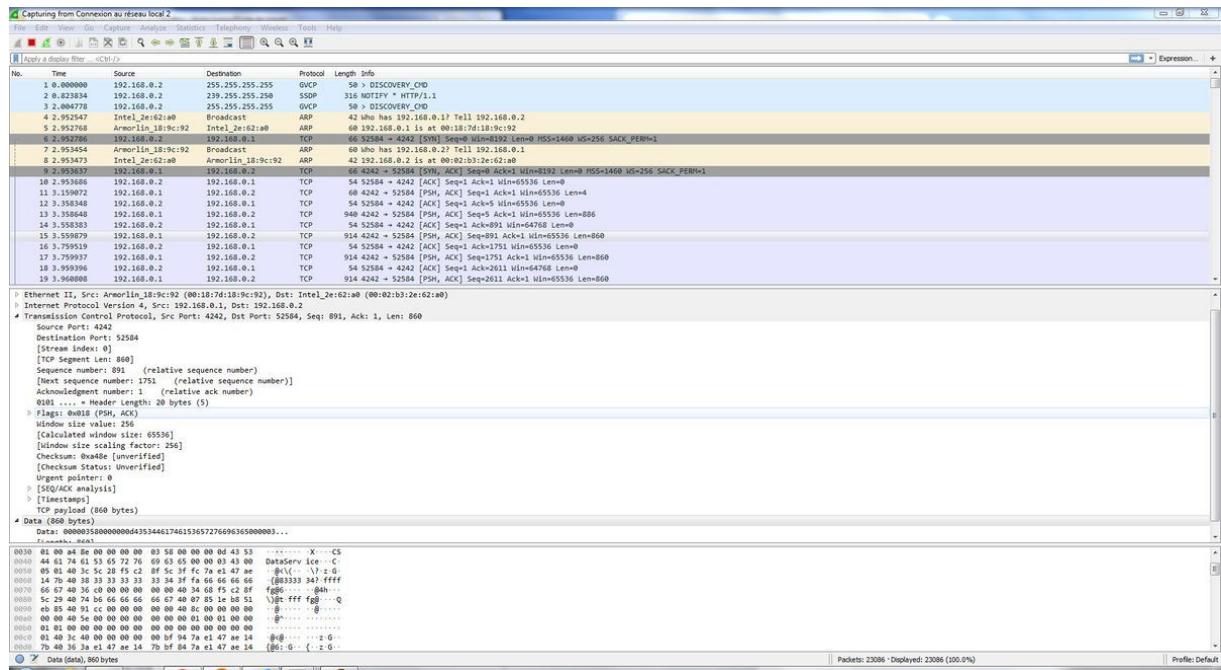


Figure 4. Récupération via Wireshark des trames reçues sur le logiciel HEL depuis une communication ethernet

Nous avons également tenté d'installer Wireshark sur le panel PC pour voir si plus d'informations sur les trames y circulaient, mais le logiciel ne détecte pas la connexion et donc les trames qui transitent entre le panel PC et notre PC.

Après avoir contacté l'entreprise Heliocentris en vain, M. Mathieu Bressel de l'équipe E2C nous a aidé en nous fournissant un grand nombre de documentations décrivant d'autres protocoles de communication entre différentes parties de la pile et le panel PC.

2.2. Protocole CAN

Un des protocoles principal dans cette communication est le CAN. Comme le montre la photo ci-dessous, il y a trois possibilités de récupération des données CAN :

- en haut du bus, juste avant le PC panel
- en entrée/sortie du convertisseur DC/DC
- en fin de bus, au niveau de la pile à combustible

La première possibilité permet de vérifier le bon envoi des trames vers le PC Panel, et la troisième de vérifier la bonne réception des trames.



Figure 5. 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/envoie des trames de données.

L'annexe A2 à la page 18 montre le résultat du sniffer 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. Aucune échelle n'est utilisée comme la documentation pourrait le faire penser. Il suffit de convertir en décimal puis éventuellement de diviser par un multiple de 10 pour retrouver les valeurs affichées sur le PC panel.

Pour pouvoir recevoir et envoyer des trames cette fois à partir de Labview afin de réaliser la commande et la supervision du système, le module d'interface CAN Ni-9862 (National Instrument) a été commandé. En effet, il est compatible avec le module CompactDAQ déjà en place dans la salle, dispose d'un port CAN avec une impédance

interne de 120 ohms et il peut supporter des réseaux CAN communiquant à des vitesses maximum d'1Mbits/s, ce qui correspond au bus de notre système.

Après avoir installé le driver logiciel NI XNET nous avons donc essayé de recevoir les données via le câble D-Sub directement relié au bus CAN de la pile. Cependant, il s'avère que l'interface NI9862 requiert une alimentation continue entre 9 et 30V (consommation d'1W) en entrée, et le bus CAN de la pile ne fournit pas cette alimentation. Nous avons utilisé un adaptateur mâle vers femelle en tant qu'élément intermédiaire (pour ne pas travailler directement sur le câble CAN), sur lequel nous avons soudé des câbles pour ressortir chaque pin. L'autre extrémité de chaque pin est alors soudée vers un autre adaptateur qui lui est connecté à la carte d'acquisition. Entre les deux adaptateurs se trouve donc une zone tampon où nous pouvons placer l'alimentation appropriée sur les pin 6 (masse) et 9 (VSup). Dans un premier temps nous avons travaillé avec les grosses alimentations stabilisées-réglables des salles C20x pour tester notre montage. Puis, nous avons utilisé l'alimentation du châssis CompactDAQ. Pour cela, nous avons donc coupé le câble pour placer un bornier entre les deux extrémités. De ce fait, la tension récupérée est directement celle fournie par l'alimentation du châssis. L'interface NI XNET Bus Monitor (Figure 6) permet facilement de vérifier la réception et envoie des données de la même manière qu'avec le sniffer CAN.

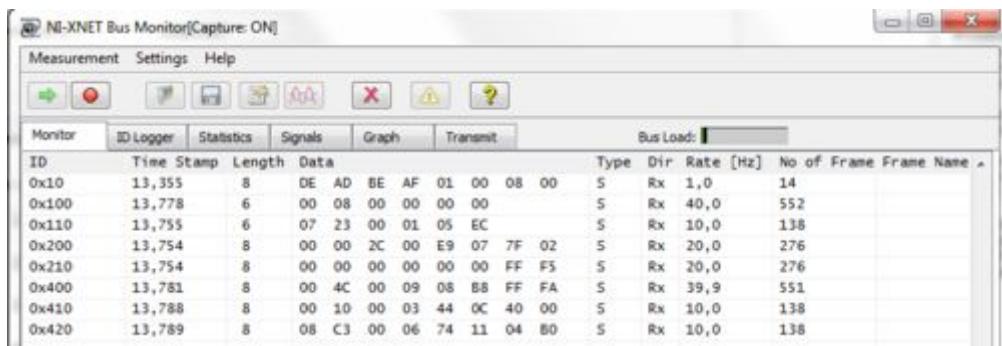


Figure 6. réception des trames CAN sur le logiciel NI-XNET

Ci-dessous le montage final, avec la carte d'acquisition à gauche, l'alimentation entourée en violet et le bornier en haut à droite.



Figure 7. Montage de la carte d'acquisition NI9862

2.3. Communication via USB

Le deuxième protocole utilisé dans l'échange des données est l'USB, entre la charge électronique et le panel PC. Pour être plus précis, en connectant un PC à l'interface IF-U1 (ci-contre) située à l'arrière de la charge via un câble USB-USB (mâle mâle), le PC reconnaît la charge comme un port COM virtuel.

Cependant, à la différence du CAN où un port était libre pour ajouter un câble, l'interface USB de la charge ne propose qu'un seul port USB et deux ports RJ45.



La documentation des IF-U1 insiste sur le fait que ces **ports RJ45** sont des *System Link Ports* c'est à dire des ports à n'utiliser que pour associer l'alimentation de plusieurs charges, et uniquement des charges de type PSI9000 (notre charge est une de type EL3000). L'idée de récupérer la communication sur ces ports n'est donc pas retenue.

Nous avons donc pensé à un **hub USB**. Un premier obstacle est survenu : connecter deux PC en USB (le PC panel au PC) peut provoquer des courts-circuits et abîmer les systèmes si le câble ne propose pas une inversion de son RX/TX. De plus, le hub USB ne permet que de recevoir les données de la pile uniquement dans une configuration : la charge en entrée du hub et la sortie du hub vers le pc panel.

Nous avons donc conçu un câble USB vers USB (mâle mâle) depuis deux câbles USB basiques, avec inversion interne des bus de transmission. Mais l'ordinateur n'a pas du tout reconnu la connexion USB dans le gestionnaire de périphériques. En effet, pour établir une connexion USB, le PC envoie une requête au système et attend une réponse, et là aucune réponse n'est générée.

Le hub étant potentiellement un problème si il empêche les communications de se transmettre dans les deux sens, M. Flamen nous aide en préparant un **câble à 3 ports USB** (mâles) dont un dénudé. Les deux ports USB vont respectivement à la charge et au PC panel, et la partie dénudée est sniffée via un oscilloscope comme le montre la photo ci-dessous. La vérification que des trames circulent est bien faite mais il faut trouver une interface pour traiter les données avant de les communiquer au PC comme expliqué plus tôt. La solution de l'arduino ou d'une carte d'acquisition NI instrument se présente. Cependant, elle n'est pas viable car la fréquence à laquelle nous recevons les trames est bien trop rapide pour être reçue via une carte d'acquisition, sans compter le fait que le protocole USB est très complexe à décoder manuellement.

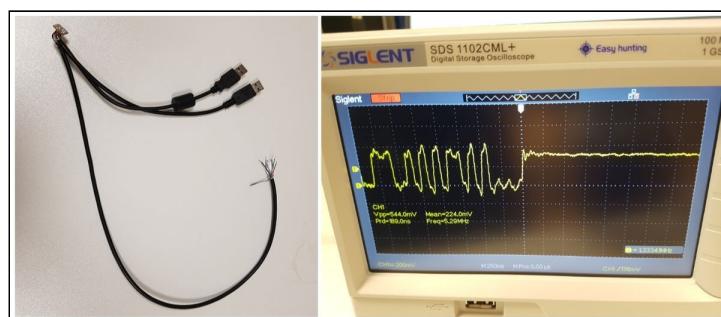


Figure 8. Réception des trames de la communication USB sur l'oscilloscope

Nous nous attardons alors sur l'interface de transmission de données de la charge, le IF-U1 dont la documentation comprend un certain nombre d'informations sur la programmation de ces interfaces. 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 depuis la documentation de l'IF-U1 la requête 71 (47 en hexadécimal sur le troisième Byte) qui permet de récupérer l'état de tension, courant et puissance de la charge.

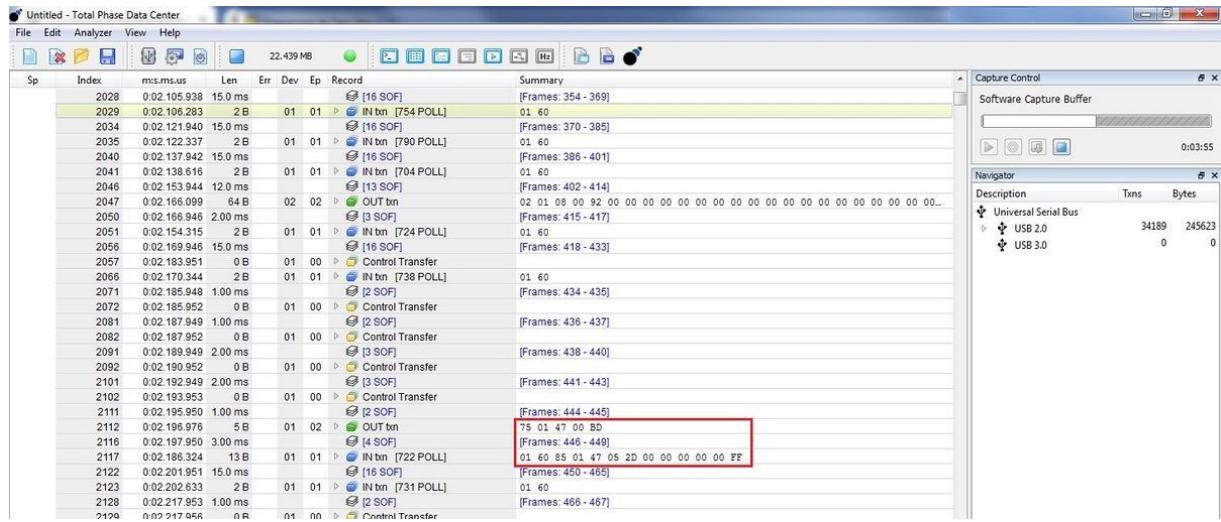


Figure 9. Réception des trames USB sur le logiciel Total Phase Data Center

Pour obtenir les valeurs réelles, on convertit de hexadécimal à décimal, puis on divise par la valeur nominale associée (160V pour la tension, 60A pour le courant, 400W pour la puissance). Nous avons répertorié un certain nombre de requêtes comme l'état du système, ses valeurs, engager le remote mode, etc.

Cependant, cette configuration est limitée : par définition, ce sniffer n'a aucun impact sur le bus et de ce fait aucune donnée ne peut être envoyée depuis notre PC de cette manière. Nous avons donc par la suite uniquement connecté la charge à notre PC, sans prendre en compte le panel PC. Depuis le logiciel Serial Port Monitor, on peut visualiser et envoyer des trames circulant sur le bus. En envoyant la requête 71 évoquée plus tôt, on obtient le même résultat que lorsqu'on observait les trames envoyées par la charge vers le panel PC.

Ces tests nous permettent de se positionner dans la configuration où le panel PC n'a plus accès aux données de la charge car nous connectons directement la charge électronique à notre PC. De cette manière on peut se passer du Beagle USB et utiliser Labview pour contrôler la charge.

2.4. Liaison RS485

La liaison RS485 entre le "power management system" et le panel PC permet de remonter des données de mesures supplémentaires comme le débitmètre, la température des réservoirs d'hydrure, Icc et Ucc de l'onduleur 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 10. Module I-7019R

M. Conrard nous a fourni un convertisseur RS485 vers USB afin de pouvoir communiquer avec ce module, et le fabricant ICPDAS fournit le logiciel DCON Utility Pro pour se connecter. Cependant, en se connectant directement du bas de la pile vers le PC via le convertisseur, le port COM est bien reconnu mais aucun module n'est reconnu.

Nous avons alors lancé la recherche de module via le même logiciel directement sur le panel PC pour observer le résultat qu'il fallait obtenir sur notre PC pour certifier la bonne connexion au I-7019R. Le résultat a été rapide depuis le panel PC la détection s'est faite directement et on a pu tester l'envoi de requêtes.

Dans le même temps nous avons mis en place un diviseur de bus qui permet de connecter le bus RS-485 via le convertisseur à notre PC et de laisser en place 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.

A ce jour, le problème n'est toujours pas résolu. La détection du module devrait vraisemblablement passer par l'initialisation et la configuration de celui-ci directement via notre PC au risque qu'il ne soit plus détectable par le panel PC. En effet, un seul "maître" peut récupérer les données des capteurs indépendants par le biais de la liaison série, c'est la source de notre problème.

2.5. Interface Labview

L'interface Labview actuelle a été réalisée à partir des données CAN, seules données pour lesquelles nous avons pu aller aussi loin à cette date.

Le VI (code Labview) principal est donc composé d'une partie connexion : depuis une session XNET définie auparavant dans le projet, un module de récupération des données est disponible. Tant qu'il n'y a pas d'erreur de connexion, toutes les trames qui transitent sont alors récupérées. Chaque trame disposant de données particulières, elles sont triées par adresse afin de correctement les exploiter. La figure 11 montre un extrait de ce VI principal.

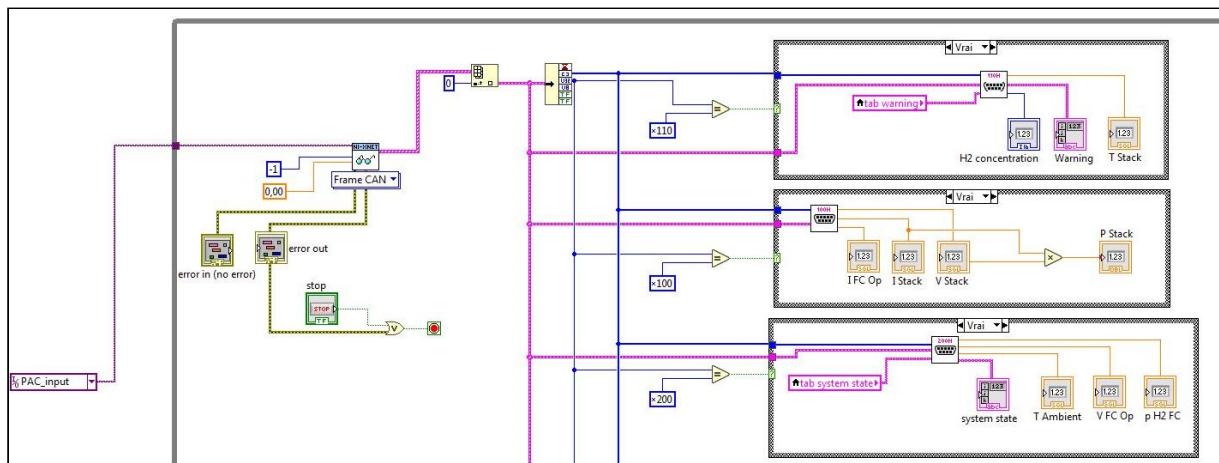


Figure 11. Extrait du VI Labview permettant la récupération, le traitement et l'affichage des données issues de la communication CAN

Pour plus de clarté, chaque type de trame est traitée dans un sous-VI associé à son adresse, et ces traitements sont issus de la documentation de la pile. L'annexe A3 à la page 19 montre un exemple de sous-VI de récupération des données.

Contrairement aux valeurs chiffrées qui n'ont qu'à être affichées après avoir été récupérées, les messages d'erreurs ne sont pas encore exploitables. En effet, en sortie du sous-VI ils sont à l'état de valeur numérique. La valeur étudiée est alors par la suite stockée dans un tableau binaire. Si le bit vaut 1, le message associé à l'indice doit s'afficher, sinon non. Il faut noter que plusieurs bits peuvent être à 1 et donc plusieurs messages doivent pouvoir s'afficher. Pour cela on initialise en parallèle, un tableau global recensant tous les messages possibles de la valeur associée. Par exemple, il peut y avoir 16 Warnings, donc un tableau contenant ces 16 messages possibles est créé. À la page 19 l'annexe A4 détaille tous les tableaux de messages communiqués par la PAC et l'annexe A5 le sous-VI d'affichage des messages.

Les annexes A6 et A7 à la page 20 correspondent à l'interface Labview réalisée avec les données CAN, et notamment les deux premiers onglets que l'on peut retrouver sur l'interface HEL. L'annexe A8 à la page 20 est le VI principal global.

Dans le VI principal ne figure pas d'envoi de trames. En effet, celle-ci n'est pas encore mise en place mais est déjà réalisée dans un VI à part comme le montre la Figure 11 ci-dessous. L'annexe A9 à la page 21 est un exemple de tests prouvant de la fonctionnalité de ce dernier.

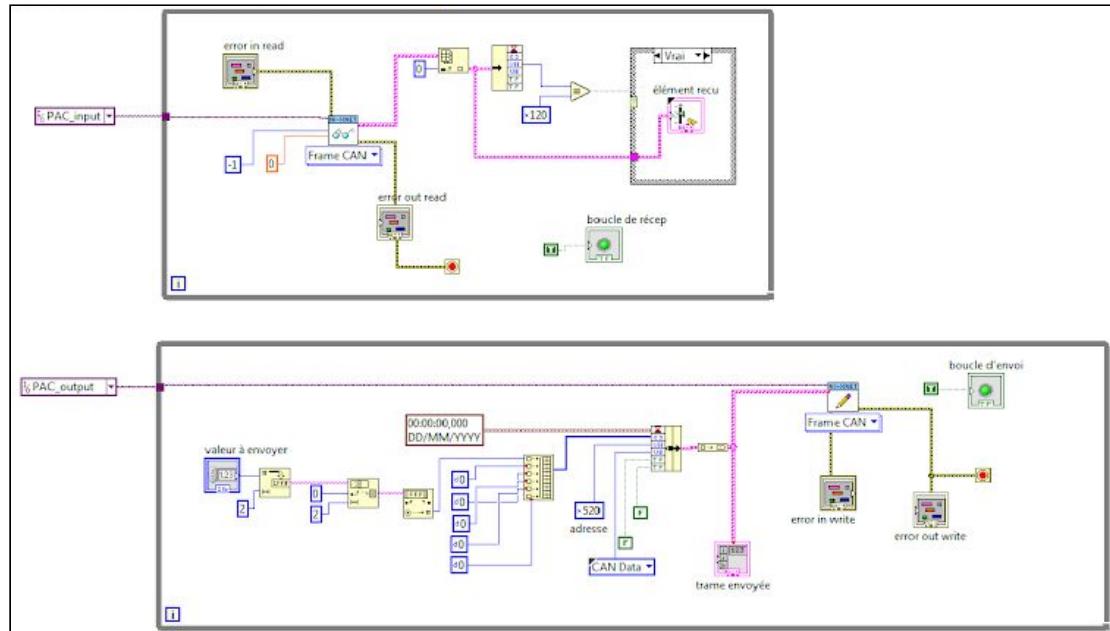


Figure 12. VI Labview d'envoi et réception simultanée de trames CAN

L'interface Labview est donc finie au niveau de la récupération des données issues du protocole CAN.

3. Ce qu'il reste à faire

3.1. Récupération du reste des données

L'objectif est toujours de réussir à récupérer les données issues du bus RS485 et du bus USB et à les afficher / contrôler sur Labview. Nous envoyons toujours régulièrement des mails à Heliocentris au cas où ils acceptent de communiquer plus d'informations sur la communication TCP/IP, mais sans réponse.

Pour le bus USB, deux choix sont possibles :

- Soit nous décidons que la partie charge électronique (charge présente sur le système) n'est pas commandable sur notre PC, et dans ce cas on ne fait que lire les valeurs grâce au sniffer Beagle 12. Le sniffer devra donc faire parti du système et un VI Labview communicant grâce à ce système doit être développé. Le module Beagle 12 offre des modules de lecture sur Labview dans le CD d'installation, l'association des données de la charge à notre interface doit donc être possible sans trop de problème. Cependant, le point négatif est que le sniffer n'a pas d'impact sur le bus et donc la commande de la charge ne pourra pas être possible depuis notre interface.
- Soit nous décidons que le panel PC n'aura plus accès aux données de la charge car nous la connectons directement à notre PC. De cette manière, on peut utiliser les modules lecture de port série sur Labview et se passer du sniffer Beagle. De ce fait, l'interface sera fonctionnelle à la place du panel PC.

Nous n'écartons cependant pas encore entièrement la possibilité de réussir à trouver un moyen d'interconnecter les deux PC avec la charge.

Pour le bus RS485, il faudra trouver le moyen d'initialiser et configurer le module d'entrées/sorties I-7019R depuis notre PC pour détecter et échanger des trames de données via le protocole DCON. Il serait aussi intéressant de trouver la localisation du module afin de pouvoir analyser son câblage et peut-être trouver une autre solution matérielle.

Une fois que nous arriverons à récupérer les données, la programmation sur Labview ne devrait également pas poser de problème puisque, comme évoqué plus tôt, des modules de gestion de port série y sont déjà disponibles et bien documentés.

3.2. Ajout d'un système externe

Nous aimerions, dans la deuxième partie de projet, 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 employé dans le secteur des transport et notamment de l'automobile. Cela permettrait de voir plus facilement comment une bonbonne d'hydrogène arrive à lancer un moteur par exemple. Il faudra penser au système à installer entre le moteur et la pile à combustible afin de créer un système complet PAC + moteur, contrôlable et supervisable à partir d'une même interface Labview.

Nous n'y avons pas encore énormément réfléchi, cependant nous anticipons déjà les difficultés que nous allons rencontrer : la PAC a un problème au niveau du convertisseur DC/DC, ce qui engendre un mauvais fonctionnement du système. La tension délivrée par le convertisseur est trop faible pour pouvoir recharger les batteries ou alimenter la charge électronique. Des erreurs indiquant un niveau trop faible de tension sont identifiables dans le logiciel Heliocentrism. Ces problèmes devront donc être réglés avant d'entamer le travail de mise en place d'une nouvelle charge.

3.3. Association PAC / electrolyseur

Une fois que nos objectifs principaux seront réalisés, nous aimerions associer notre interface de commande/supervision avec celle réalisée par l'étudiant M. Cockenpot en 2018 sur le PLC relié aux éoliennes et autres sources d'énergie renouvelables. Pour cela, il faut reprendre ses anciennes configurations.

Nous avions également évoqué le fait de rendre possible la recharge des bonbonnes par l'électrolyseur dans la PAC, et donc d'avoir un seul et même système ne nécessitant pas d'intervention manuelle. Si cette configuration est possible, nous aimerions ajouter une fonctionnalité permettant de travailler dans différents fonctionnements :

- Si la puissance offerte est très supérieure à celle demandée, alors le surplus est stocké dans les batteries ainsi que sous forme hydrogène (tant que la pression dans les bombonnes est inférieure à 10 bar).
- Si la puissance proposée est toujours supérieure à la demande mais de manière moins importante alors le stockage est uniquement dirigé vers les batteries.
- Si la puissance offerte est inférieure à la demande, le stockage dans les batteries est utilisé puis l'hydrogène et enfin le réseau électrique si cela est nécessaire.

Cependant, nous allons potentiellement rencontrer un petit contre temps : en effet, un groupe de Master est assigné à un projet sur l'électrolyseur et donc utilisent le PLC évoqué. La mise en commun ne sera pas forcément pratique pour tout le monde.

De plus, la pile à combustible sur laquelle nous travaillons permet d'ajouter une source d'hydrogène externe (comme l'électrolyseur) mais n'a pas l'air de pouvoir recharger les bouteilles à l'intérieur. Nous devrons donc réfléchir à tout cela.

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 est de réaliser une interface de supervision de la pile à combustible mise à notre disposition afin d'en gérer différents modes de fonctionnement.

Tout au long de ces premières semaines, nous avons pu travailler sur les bus de communication présents sur cette pile à combustible afin de récupérer les données et réaliser l'Interface Homme-Machine. Comme nous avons pu l'expliquer dans ce rapport, cette partie fut plus longue et difficile que prévue et devra donc être continuée dans les prochaines semaines. La majorité des données est tout de même déjà traitée et affichée sur l'interface Labview que nous avons développée.

Par la suite, il est souhaité qu'une charge externe soit associée à la pile à combustible, et que potentiellement une mise en commun avec l'électrolyseur soit réalisée afin d'avoir le système complet supervisable depuis la même interface.

Annexes

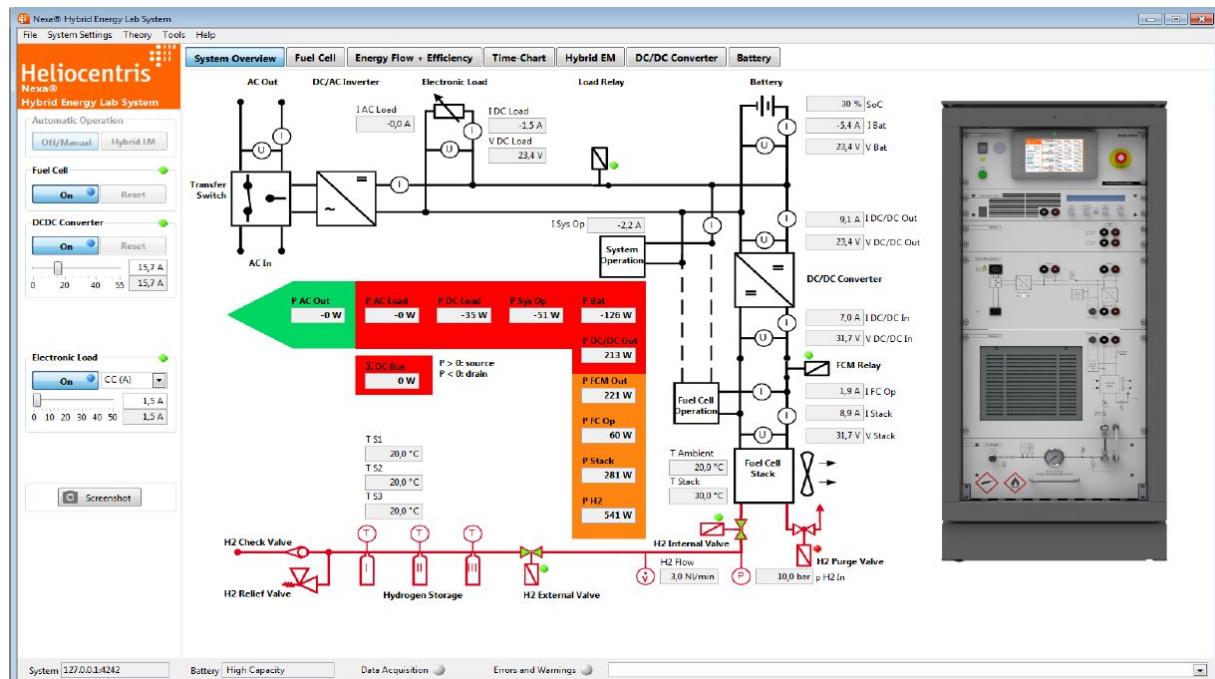


Figure A1. Logiciel Heliocentris (HEL) de la PAC

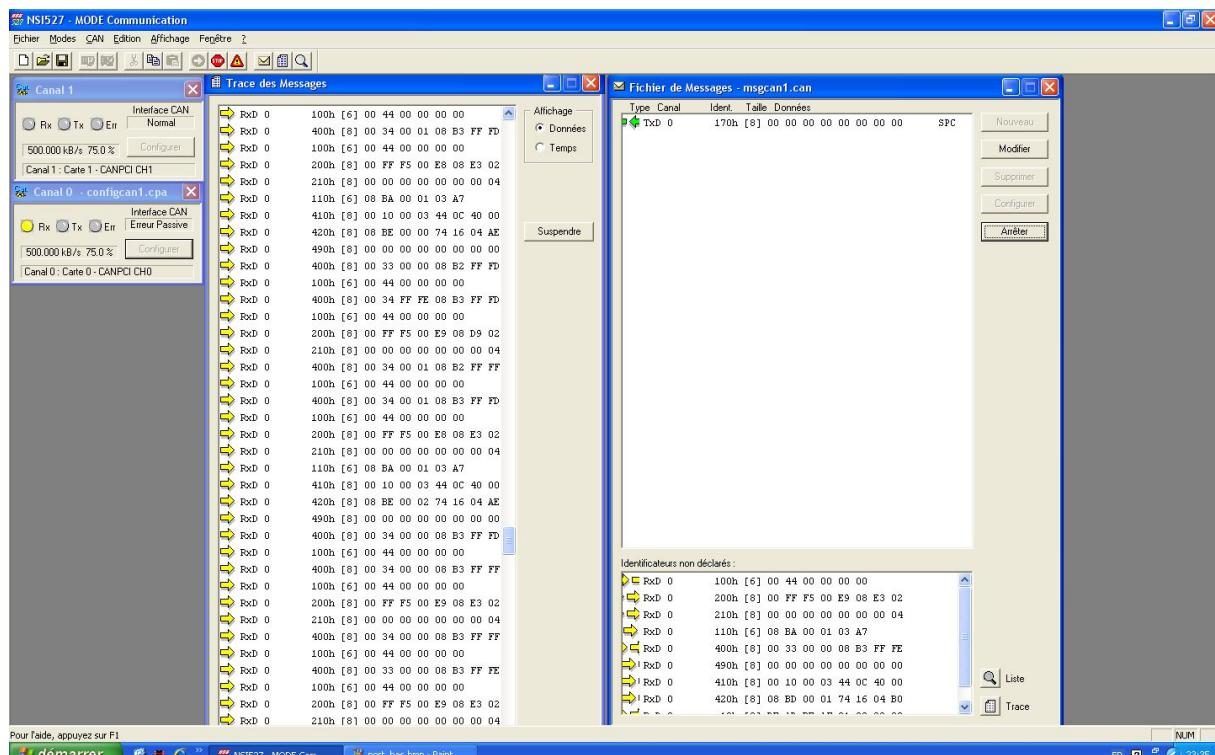


Figure A2. Trames issues du sniffer CAN

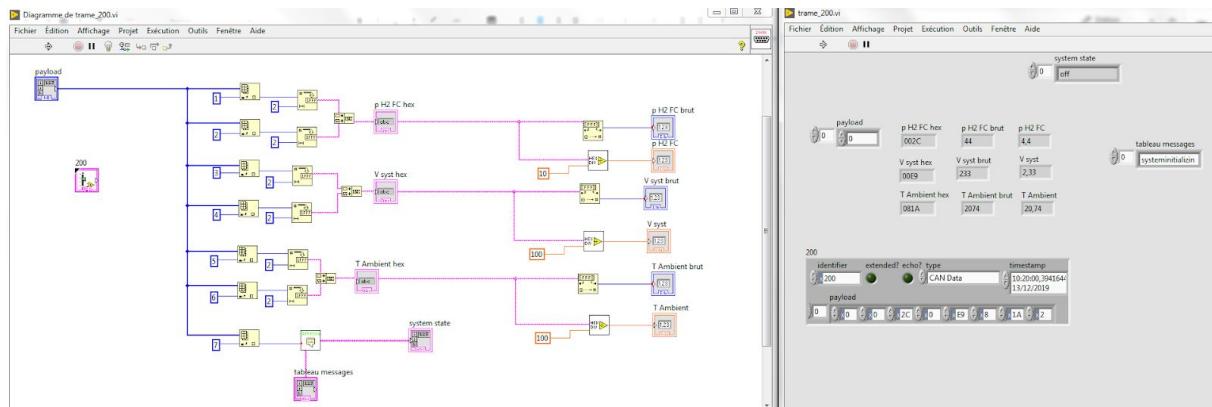


Figure A3. VI et face avant VI Labview associés au traitement des données issues de la trame CAN x200

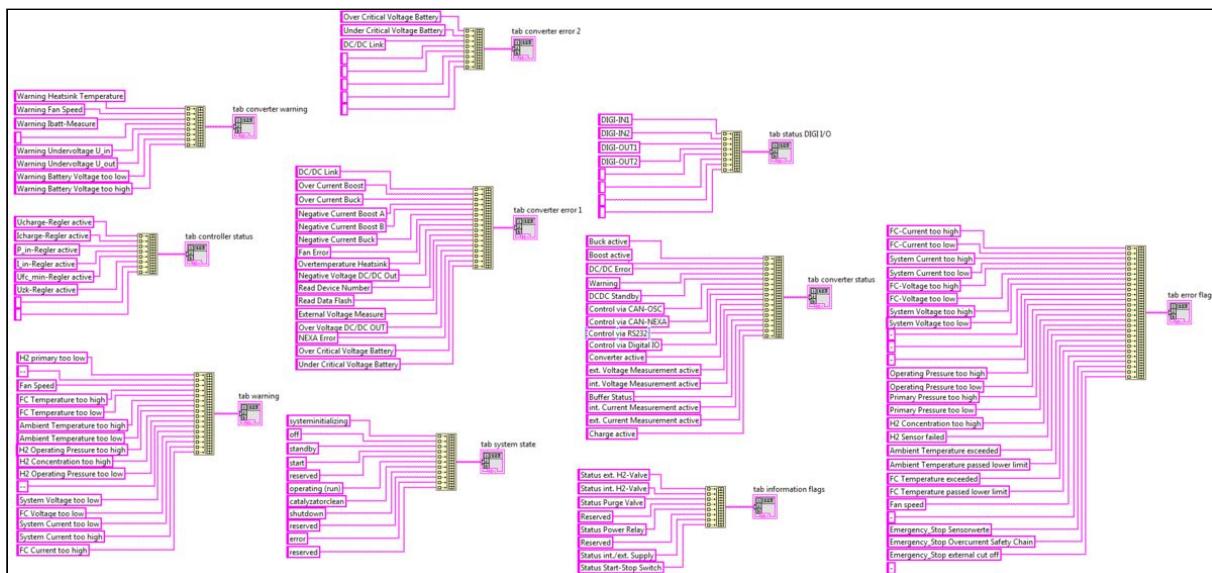


Figure A4. Initialisation Labview de tous les messages reçus par la PAC depuis le CAN

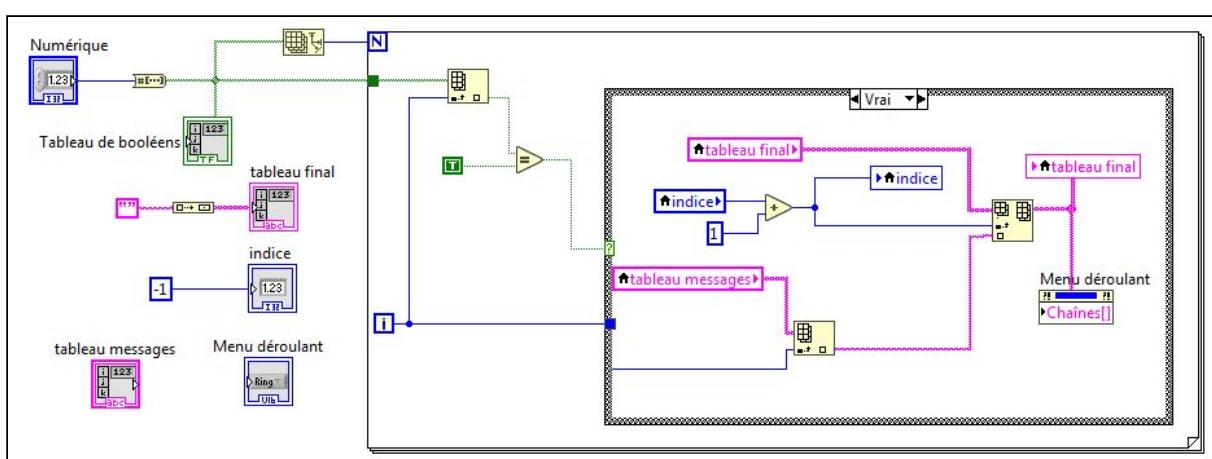


Figure A5. Sous VI Labview permettant d'associer une liste de messages à un tableau binaire

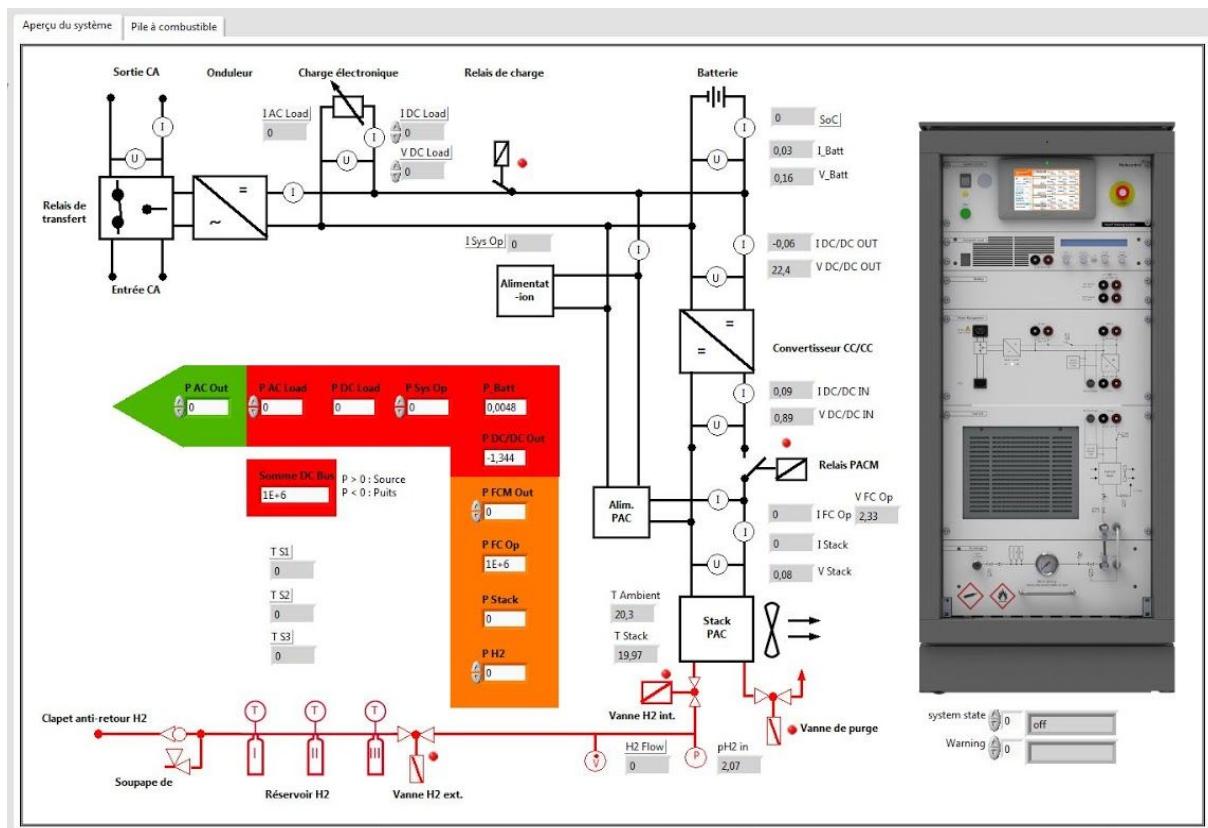


Figure A6. Onglet “Aperçu du système” réalisé sur Labview

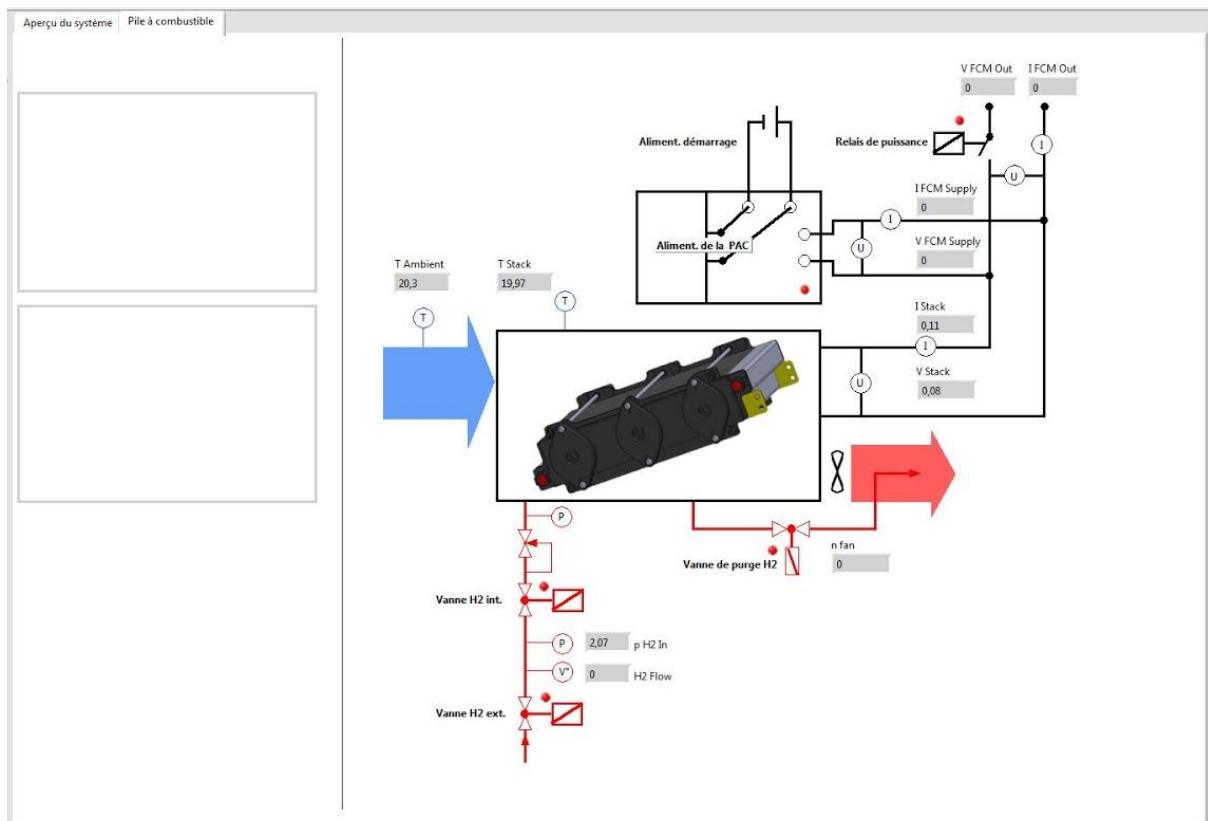


Figure A7. Onglet “Pile à Combustible” réalisé sur Labview

Figure A4. VI Labview principal

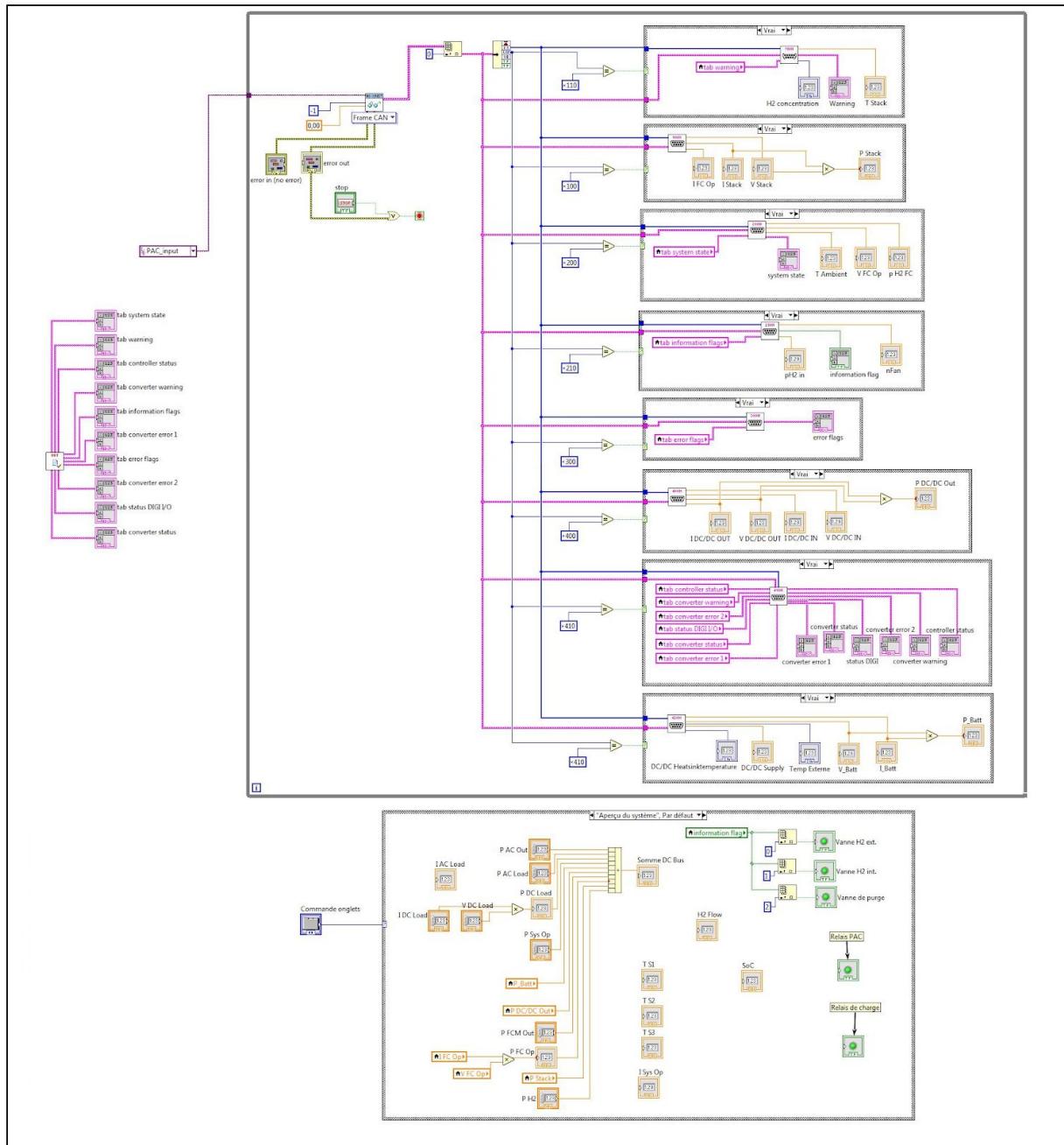


Figure A8. VI principal contenant le traitement de la communication CAN



Figure A9. Face-avant Labview montrant la fonctionnalité d'envoi et réception simultanée de trames CAN

La trame x120 est envoyée (par la PAC vers le PC) en réponse à la trame x520 (envoyée du PC vers la PAC). Sur la capture ci-dessus, on voit en bleu que la valeur envoyée par la trame 520 est bien retrouvée dans la trame 120 de réponse (avec le reste des données associées).