

Filière Systèmes industriels

Orientation Power and Control

Diplôme 2008

Victor Penas

*Développement d'un
interrupteur «intelligent»*

Professeur Jean-Daniel Marcuard

Expert André Rotzetta

SI	TV	EE	IG	EST
X	X			

Confidentiel / *Vertraulich*

oui / ja non / nein

<input checked="" type="checkbox"/> FSI <input type="checkbox"/> FTV	Année académique / <i>Studienjahr</i> 2007/2008	No PS / <i>Nr. PS</i> pc/2008/37
Mandant / <i>Auftraggeber</i> <input type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Ecole hôte	Etudiant / <i>Student</i> Victor Penas	Lieu d'exécution / <i>Ausführungsort</i> <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Ecole hôte
Professeur / <i>Dozent</i> Jean-Daniel Marcuard	Expert / <i>Experte</i> (<i>données complètes</i>) André Rotzetta - HES-SO Fribourg	

Titre / *Titel*

Développement d'un interrupteur «intelligent»

Description et Objectifs / *Beschreibung und Ziele*

La libéralisation du marché de l'énergie et l'intégration dans les réseaux électriques de nombreux petits producteurs exploitant des énergies renouvelables vont poser des problèmes nouveaux aux distributeurs. Pour les résoudre il sera notamment nécessaire d'exploiter la souplesse que certains utilisateurs peuvent mettre à disposition. Pour cela il faut disposer d'interrupteurs capables de décider de la mise en marche ou de l'arrêt d'un appareil sur la base de paramètres disponibles sur un serveur distant. La loi de commutation doit être configurable par l'utilisateur, si possible par l'intermédiaire d'une page Web.

L'objectif du projet de diplôme consiste à réaliser un prototype sur la base du concept proposé dans le cadre du projet de semestre. Cela comprend :

- La réalisation et la programmation de l'interrupteur.
- La programmation du serveur distant, y compris la ou es base(s) de données nécessaire(s).
- La validation du fonctionnement de l'ensemble.

Signature ou visa / *Unterschrift oder Visum*

Resp. de la filière

Leiter des Studieng.:

Etudiant/Student:

Délais / *Termine*

Attribution du thème / *Ausgabe des Auftrags*:
01.09.2008

Remise du rapport / *Abgabe des Schlussberichts*:
21.11.2008, 12:00

Exposition publique / *Ausstellung Diplomarbeiten*:
28.11.2008

Défense orale / *Mündliche Verfechtung*:
semaine/Woche 49

Développement d'un interrupteur « intelligent »

Entwicklung eines intelligenten Umschalters

Objectif

La libéralisation du marché de l'énergie ainsi que l'intégration dans les réseaux de nombreux petits producteurs d'énergie renouvelable vont poser des problèmes nouveaux aux distributeurs. Pour y remédier, il sera nécessaire d'exploiter la souplesse que certains utilisateurs peuvent mettre à disposition. Le travail de diplôme consiste à réaliser un prototype qui démontrera qu'il est possible de créer un système capable de commuter des charges à distance en tenant compte de la loi de commutation configurable par l'utilisateur.

Résultats

La maquette qui simule une maison est construite et permet de valider le bon fonctionnement d'un interrupteur intelligent. Cela dit, la programmation du serveur distant n'est pas terminée. Il était prévu que le serveur ait une interface graphique, mais en vu des problèmes qu'elle apporte au fonctionnement du système, elle n'a pas été terminée. À sa place, la loi de commutation a été réalisée sur un environnement console. Néanmoins, toute la partie de configuration du système et l'enregistrement de l'utilisateur sur la base de donnée et terminée.

Bien entendu, comme tout prototype, ce projet devra être repris et amélioré dans le but de parvenir à des tests réels.

Mots-clés

Interrupteur intelligent, programmation, énergie renouvelable, gestion du réseau

Ziel

Die Liberalisierung des Strommarktes sowie die Anschliessung vieler kleinen erneuerbaren Stromquellen an das Netz, werden die Verteiler vor neue Probleme stellen. Um diese zu lösen, wird es zukünftig notwendig sein die Flexibilität gewisser Stromverbraucher auszunützen. Diese Diplomarbeit soll dazu dienen ein Prototyp zu entwickeln um zu zeigen dass es möglich ist ein System aufzubauen dass Lasten ferngesteuert umschalten kann. Dieses System soll vom Benutzer konfigurierbar sein.

Resultate

Der Versuchsaufbau simuliert ein haus und ermöglicht es die Funktionsweise eines Ferngesteuerten Schalter zu bestätigen. Es sollte eigentlich eine grafische Schnittstelle entwickelt werden, diese konnte aber wegen den Störungen die sie für Funktionsweise herleitet, nicht fertig entwickelt werden. An dessen Stelle werden die Umschaltungen von einer Konsole gesteuert. Das ganze konfigurierbare Teil sowie das Speicherungssystem für den Benutzer konnte fertig gestellt werden.

Wie jeder Prototyp, wird auch dieses im Rahmen eines anderen Projektes neu aufgerollt und verbessert werden, um dann reell eingesetzt werden zu können.

Schlüsselwörter

Intelligente Umschalter, Programmierung, Erneuerbare Energien, Netz Management

1 Table des matières

1	TABLE DES MATIÈRES	4
2	INTRODUCTION.....	5
3	CAHIER DES CHARGES.....	5
4	DESCRIPTION DE LA PARTIE PHYSIQUE DU SYSTÈME	6
4.1	DESCRIPTION DU DÉMONSTRATEUR	7
5	DESCRIPTION DE LA PARTIE LOGIQUE DU SYSTÈME.....	9
5.1	UTILISATIONS DU WEB	9
5.1.1	<i>Environnement du site internet développé pour l'utilisateur</i>	<i>9</i>
5.1.2	<i>Acquisition des données relatives à la bourse de l'énergie</i>	<i>14</i>
5.1.3	<i>Communication entre le serveur et l'automate.....</i>	<i>14</i>
5.2	LA BASE DE DONNÉES.....	15
5.3	LE SERVEUR	16
5.4	L'AUTOMATE	16
5.4.1	<i>Etape future : développement d'un hardware spécifique</i>	<i>16</i>
5.5	PROGRAMMATION DES DIFFÉRENTES PARTIES	17
5.5.1	<i>Choix des langages.....</i>	<i>17</i>
5.5.2	<i>Choix des programmes de développement</i>	<i>18</i>
5.5.3	<i>Développement du système</i>	<i>24</i>
6	CONCLUSIONS.....	30
6.1	ANALYSE DES PROBLÈMES RENCONTRÉS.....	30
6.2	SYNTHÈSE	30
6.3	EVOLUTIONS FUTURES	31
6.4	APPORT PERSONNEL.....	31
6.5	REMERCIEMENTS	31
7	RÉFÉRENCES.....	32
7.1	BIBLIOGRAPHIE :	32
7.2	SITES INTERNET :	32
8	TABLE DES FIGURES.....	33
9	ANNEXES.....	33

2 Introduction

A l'époque où nous vivons, le consommateur est roi. Chacun d'entre nous peut allumer son ordinateur, chauffer de l'eau ou cuisiner quand il le souhaite. Ceci est possible grâce à l'ingéniosité des distributeurs d'énergie qui gèrent le réseau électrique avec une très petite marge de manœuvre. Pour comprendre les difficultés liées à la gestion du réseau électrique, il faut savoir que l'énergie est produite en même temps qu'elle est consommée, mais que les ressources bon marché et faciles à exploiter sont limitées. D'autre part, les énergies traditionnelles comme le pétrole et le nucléaire ne sont plus acceptées comme avant par l'opinion publique.

Pour arriver à trouver une solution, il faudrait que les consommateurs soient plus flexibles (que ce soit sur les prix, les horaires, etc.) au moment d'enclencher certains appareils gourmands en énergie sans pour autant perdre du confort. De leur côté, les producteurs devront trouver des solutions pour améliorer la productivité tout en réduisant les pertes.

De nos jours, nous avons une diversification des ressources relativement importante (éolienne, solaire, biomasse, pile à combustible, etc.). Même si toutes ces nouvelles technologies ne sont pas encore tout à fait au point, elles peuvent apporter une aide considérable dans l'apport d'énergie. Si tout le monde apporte sa contribution, les pics de consommation, et en conséquence les surcoûts dus à ces pics, se verraient fortement réduits.

Avec l'apparition de nouveaux petits producteurs d'énergie renouvelable, l'exploitation de la souplesse des consommateurs sera un paramètre important à gérer. Pour cela, il faudrait disposer d'interrupteurs capables de décider de la mise en marche ou l'arrêt d'un appareil. Cette décision sera prise sur la base de paramètres disponibles sur un serveur distant, mais qui ont été configuré par l'utilisateur par l'intermédiaire d'une page web.

3 Cahier des charges

L'objectif du projet de diplôme consiste à réaliser un prototype sur la base du concept proposé dans le cadre du projet de semestre (chapitre 2). Cela comprend :

- La réalisation et la programmation de l'interrupteur
- La programmation du serveur distant, y compris la ou les bases de données nécessaires
- La validation du fonctionnement de l'ensemble

Pour y parvenir il a été nécessaire de découpler les différentes parties à traiter en deux grandes familles puis en plusieurs sous-groupes. Ces deux grandes familles sont le système physique et logique.

4 Description de la partie physique du système

Pour ce travail de diplôme il est nécessaire de créer un démonstrateur qui montrera le fonctionnement de l'interrupteur. Le démonstrateur est composé de :

- La partie interrupteur à proprement dite, construite à partir d'un automate programmable. Elle intègre les éléments suivants :
 - o Un automate Wago 750-841. Les raisons qui ont porté le choix sur cet appareil sont décrites au paragraphe « 4.2 Sélection du matériel » du travail de semestre. Les cartes couplées à cet automate modulable sont :
 - 750-841 : Il s'agit du contrôleur de l'automate. Elle gère la communication avec l'extérieur, enregistre le programme qui tournera par la suite, alimente les autres cartes et communique par le réseau de terrain avec les autres cartes.
 - 750-501 : Sortie digitale 24V. Elle permet d'allumer la lampe d'alarme ou de l'éteindre.
 - 750-513 : Sortie digitale 250V. Elle réalise la commutation de la charge. Dans notre cas, la sortie digitale permet de simuler l'état allumé ou éteint du boiler.
 - 750-611 : Carte de couplage d'alimentation.
 - 750-402 : 4 bornes d'entrées digitales. Pour ce système, l'une d'elles permet de quitter l'alarme à l'aide d'un bouton poussoir. Sur deux autres sont connectés les deux interrupteurs de forçage. L'un deux commute entre la gestion automatique de la charge et un forçage manuel, tandis que l'autre choisi si ce forçage est à On ou à Off.
 - 750-476 : Carte d'acquisition analogique en Tension (0-10V). Cette carte permet de savoir la tension que délivre le panneau solaire.
 - 750-454 : Carte d'acquisition analogique en courant (4-20mA). Cette carte permet un contrôle sur le courant de la charge.
 - 750-600 : Carte de fin de bus de terrain.

Les fiches techniques de ces différentes cartes sont dans le CD annexe.

- o Une maquette représentant une maison type où l'on a une visualisation directe des boutons, lampes et panneau solaire. Pour ce système il a été choisi de simuler le boiler. La simulation permet d'éviter certaines contraintes comme le fait de travailler avec des puissances élevées. Le but de ce travail étant de valider un concept, la simulation de la charge est amplement suffisante.

Développement d'un interrupteur « intelligent »

- Un PC (serveur) distant exécutant les programmes nécessaires à faire fonctionner tout le système. Il gère la base de données et le serveur web à travers le programme EasyPhP qui permet à l'utilisateur de voir les pages web créées. Il contient également l'application en C++ qui est le programme de base pour la gestion de l'ensemble.

4.1 Description du démonstrateur

Le démonstrateur contient une plaque de visualisation et de commande directe en alu sur laquelle est collé un dessin représentant une maison, un panneau solaire photovoltaïque, une lampe à incandescence avec son régulateur de tension et deux pc.

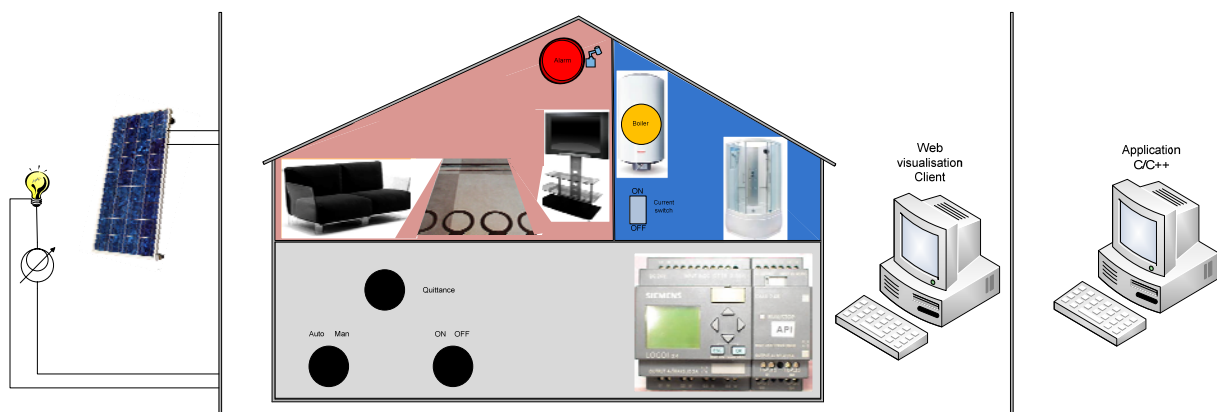


Figure 4.1 : Démonstrateur

Sur cette maquette on peut apercevoir un automate programmable, qui sera notre automate Wago, une lampe d'alarme (en haut au milieu), une lampe qui simule l'état du boiler (allumée s'il est actif), juste en dessous un interrupteur qui permettra de simuler la consommation ou non du boiler, un bouton poussoir de quittance et deux interrupteurs rotatifs à 2 positions pour le forçage du boiler.

À sa gauche, il y a un panneau solaire de 10W de puissance max. Pour simuler les effets du rayonnement solaire, une lampe à incandescence couplée à un variateur de tension agira comme source de luminosité. Ainsi, lorsque le panneau recevra suffisamment de lumière, il produira une tension suffisante sur la carte d'entrée pour que le système le détecte et transmette l'information de la mise à disposition d'énergie verte.

Afin de pouvoir adapter les plages de tension et de courant de sortie des appareils aux plages d'entrées des cartes d'acquisition analogiques, il est nécessaire de réaliser deux petits montages électroniques. Ces montages servent à réduire la tension du panneau solaire de 0-20V à 0-10V d'une part et d'autre part d'obtenir 4mA si la lampe du boiler est éteinte et 20mA si elle est allumée. Les schémas de ces montages sont en annexe 1.

À sa droite est posé un écran qui affichera la visualisation de l'utilisateur (site web). Ensuite une séparation marque la différence et la distance entre les deux PC. Bien que le premier soit l'ordinateur personnel du client, le deuxième est le serveur du distributeur. C'est là que devra fonctionner l'application en C/C++. Même si l'interface graphique ne fonctionne pas correctement, voir ch.

5.5.3.2 *Développement de l'application* en C/C++ à la page 25, le deuxième PC fera tourner l'application qui contient l'algorithme de commutation.

Pour continuer avec la description, le fonctionnement du démonstrateur est assez simple. Comme mentionné plus haut, l'automate ne réalise que des fonctions de contrôle et de forçage laissant toute la régulation et les décisions à l'application sur le serveur. Voici le fonctionnement des différentes parties « palpables » du démonstrateur :

Boutons de forçage :

Lorsque l'interrupteur « automatique-manuel » est sur automatique, c'est l'application sur le serveur qui prend les décisions. Si l'application force la sortie de l'automate à On, celui-ci incrémentera le compteur dédié aux forçages à On dut au distributeur dans sa mémoire. Il en sera de même lorsque l'application force la sortie de l'automate à Off. Cela dit, le fonctionnement détaillé de l'application sera décrit dans le paragraphe 5.5.3.2 *Développement de l'application* en C/C++ à la page 25.

Lorsque cet interrupteur est sur manuel, chaque changement sur le deuxième interrupteur à On ou Off incrémentera le compteur correspondant dans la mémoire de l'automate.

Quittance d'alarme:

C'est un bouton poussoir. Lorsque l'alarme s'enclenche, il faut appuyer sur le bouton pour redémarrer le programme normalement. Tant que l'on n'appuie pas sur la quittance, l'alarme continue.

Lampes :

Il y a 2 lampes. L'une est l'alarme (rouge) et l'autre est le boiler (verte).

Alarme

Elle peut s'enclencher de deux façons différentes.

- Lorsqu'il n'y a plus de communication avec l'application (le bit d'activité n'a pas été remis à 1 après un certain temps) la lampe s'allume sans clignoter.
- Lorsque l'on allume le boiler, si celui-ci ne consomme pas de courant pendant un certain temps, l'alarme se met à clignoter. Cette alarme est créée car un boiler peut être actif sans consommer de courant, si l'eau qu'il contient est assez chaude. Par contre, si après un délai suffisant, il n'a toujours pas consommé de courant, cela pourrait être signe d'une défaillance. S'il n'y a pas de défaillance, la quittance permettra de relancer le timer.

Boiler

La lampe du boiler est en fait une simulation de l'état actif ou non du boiler. Elle est allumée lorsque le boiler est en fonction. Dans le cas contraire, la lampe est éteinte. Pour simuler qu'il est allumé, mais ne consomme pas, il y a un interrupteur sous la lampe du boiler. Lorsque ce dernier est placé à On, cela veut dire qu'il consomme et vis versa.

5 Description de la partie logique du système

Comme indiqué plus haut, la deuxième famille est le système logique. Il comprend les liens entre chaque partie physique à travers la programmation des différents éléments. Le système comprend une base de données ainsi qu'une application de gestion stockées dans le serveur, un automate programmable, une interface web pour l'utilisateur et une interface homme-machine pour le distributeur. Voici ci-après le schéma de base du système complet à implémenter:

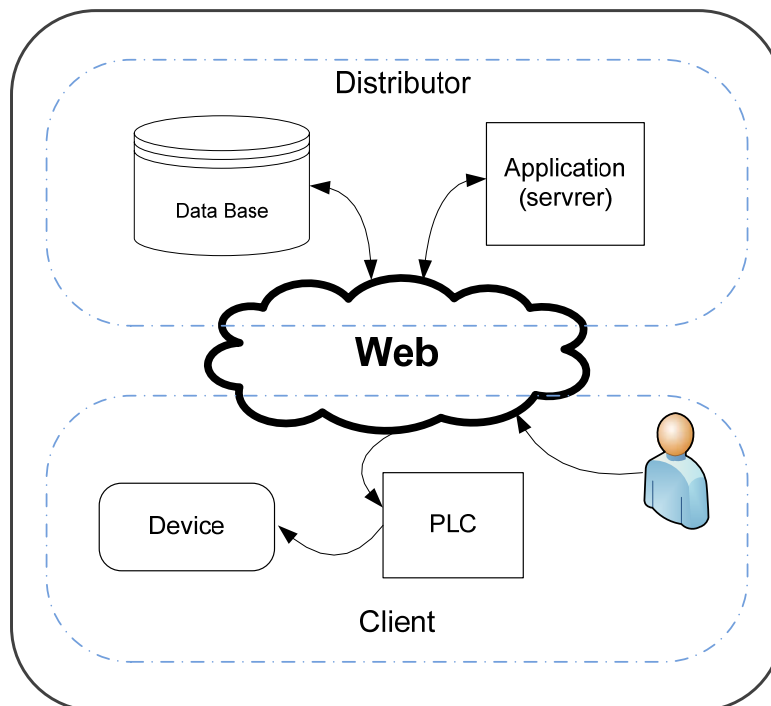


Figure 5.1 : Schéma de base du système

5.1 Utilisations du web

Dans le cas présent, le web est utilisé de plusieurs façons. D'une part, il fonctionne en tant qu'interface à travers un site pour que l'utilisateur puisse s'enregistrer sur la base de données du distributeur et qu'il choisisse la loi de commutation de son interrupteur. D'autre part, pour pouvoir satisfaire certaines exigences des consommateurs, il est nécessaire de savoir le prix du kWh. Il est donc nécessaire de rechercher cette information sur le site de la bourse de l'énergie. Finalement, internet est le support de communication entre le serveur et l'automate. Ci-après, une explication détaillée de chaque élément.

5.1.1 Environnement du site internet développé pour l'utilisateur

Le site développé pour ce projet contient trois parties. La première partie est dédiée à l'utilisateur, la deuxième à l'installation de ce dernier et la troisième à la visualisation « pratique » du système. Pour qu'il soit dynamique le choix du langage utilisé s'est porté sur le PHP accompagné de l'html. Les raisons de ce choix sont expliquées au paragraphe 5.5.1 *Choix des langages* à la page 17.

5.1.1.1 Enregistrement d'un utilisateur sur la base de données

La page d'accueil propose deux possibilités, l'enregistrement pour les nouveaux utilisateurs et le login pour les utilisateurs déjà enregistrés.

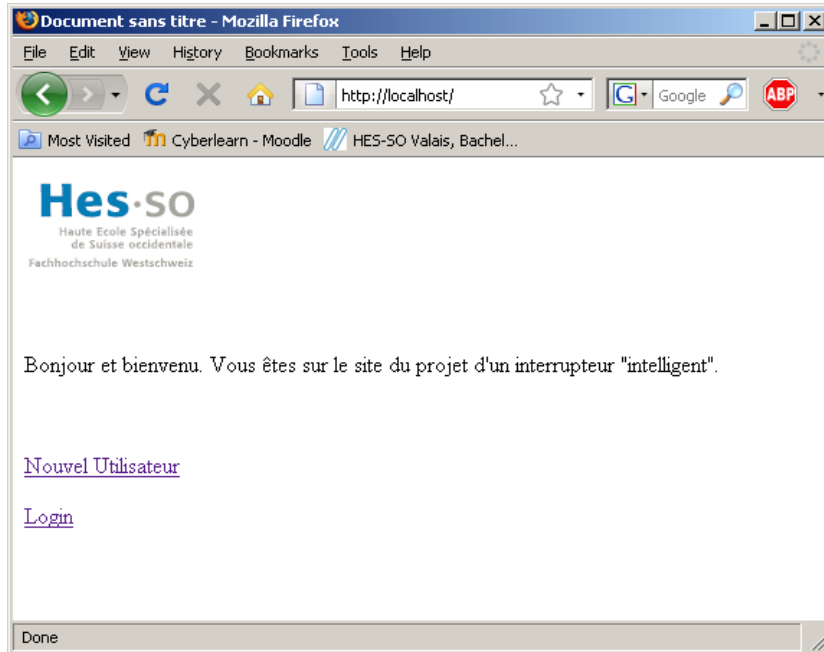


Figure 5.2 : Page d'accueil

Pour se loguer il faut cliquer sur [Login](#) puis insérer son n° d'abonné. Pour s'enregistrer, il suffit de cliquer sur [nouvel utilisateur](#), de rentrer ses coordonnées puis de cliquer sur envoyer.

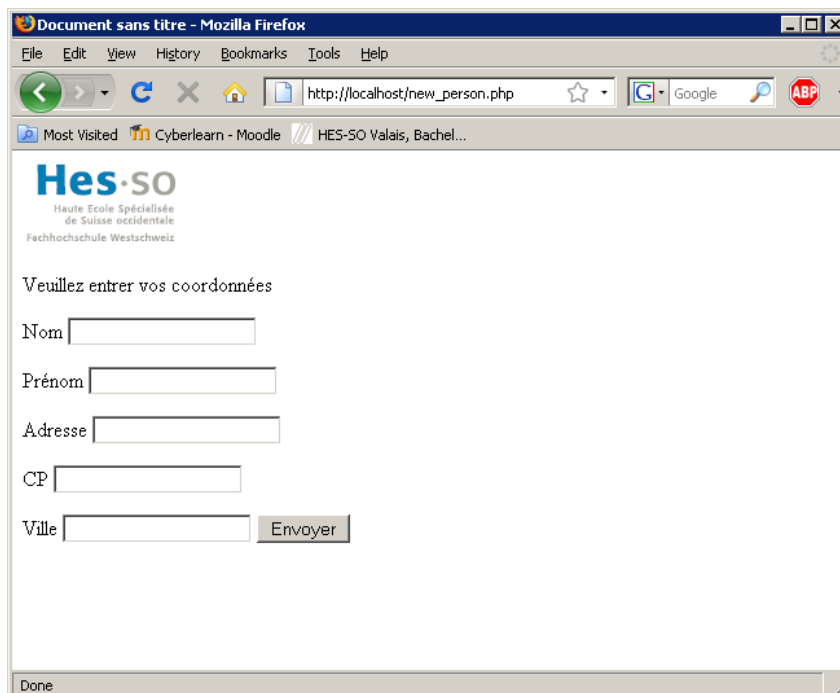


Figure 5.3 : Page d'enregistrement d'un nouveau client

5.1.1.2 Affichage, configuration et modifications du système a travers le site web

Une fois les données personnelles enregistrées, l'interface affiche le système personnel de l'utilisateur.

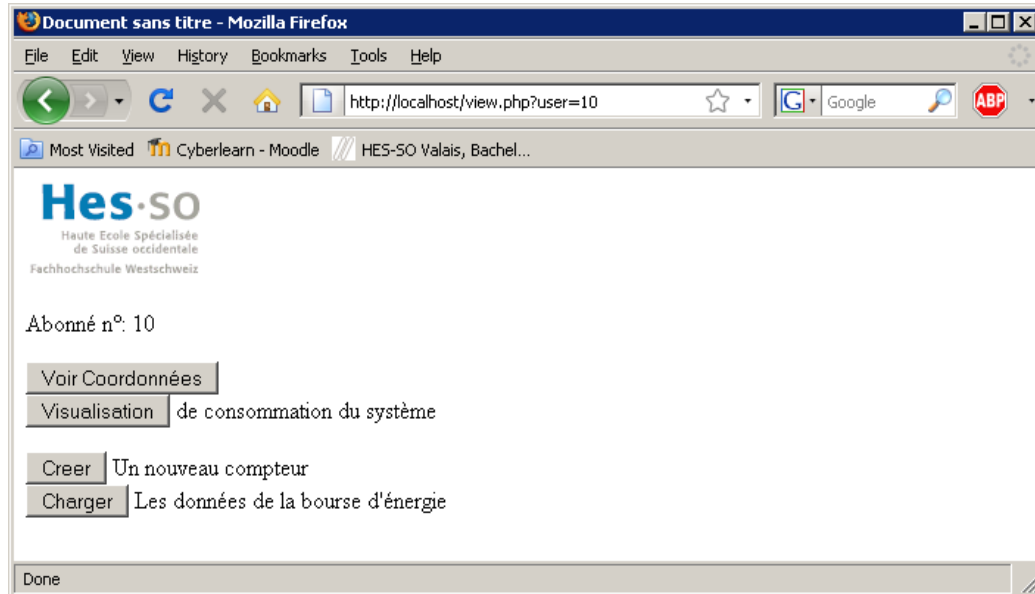


Figure 5.4 : Première visualisation principale

A partir de cette page, l'utilisateur pourra recréer son système en appuyant sur les boutons adéquats. Il commencera par créer un nouveau compteur avec son automate et son appareil auquel ils sont connectés.

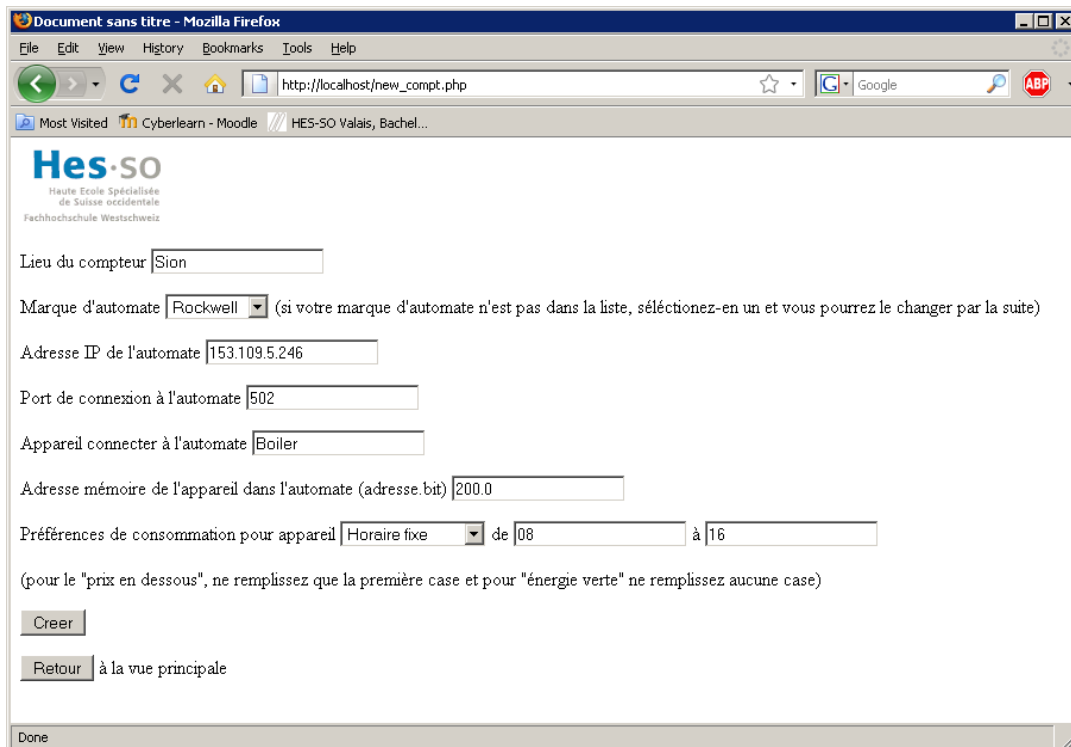


Figure 5.5 : Création d'un nouveau compteur

En cliquant sur « Créer » le programme enregistre dans la base de données les informations et revient sur la visualisation du système avec les modifications apportées.

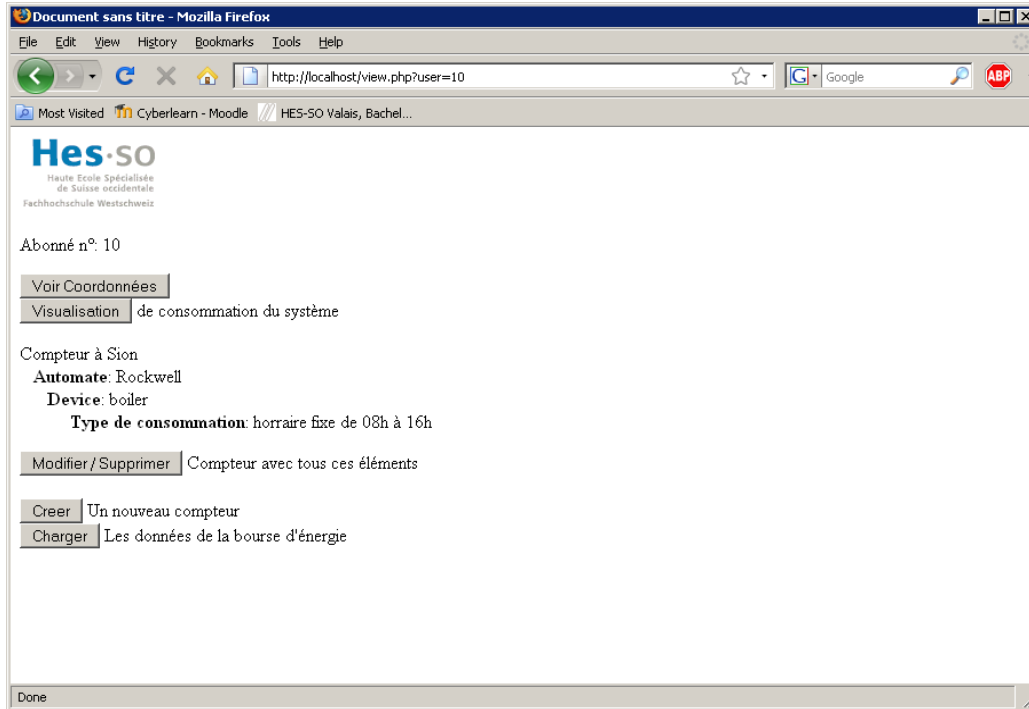


Figure 5.6 : Visualisation système

Enfin l'utilisateur, il peut ajouter, supprimer ou modifier un compteur, un automate puis un appareil à l'aide des pages mises à disposition. Voir le restant des pages en annexe 2.

Développement d'un interrupteur « intelligent »

5.1.1.3 Visualisation du côté utilisateur de la consommation du système

Une fois le système recréé, l'abonné pourra visualiser le prix du kWh en cours ainsi que l'état de ses appareils.

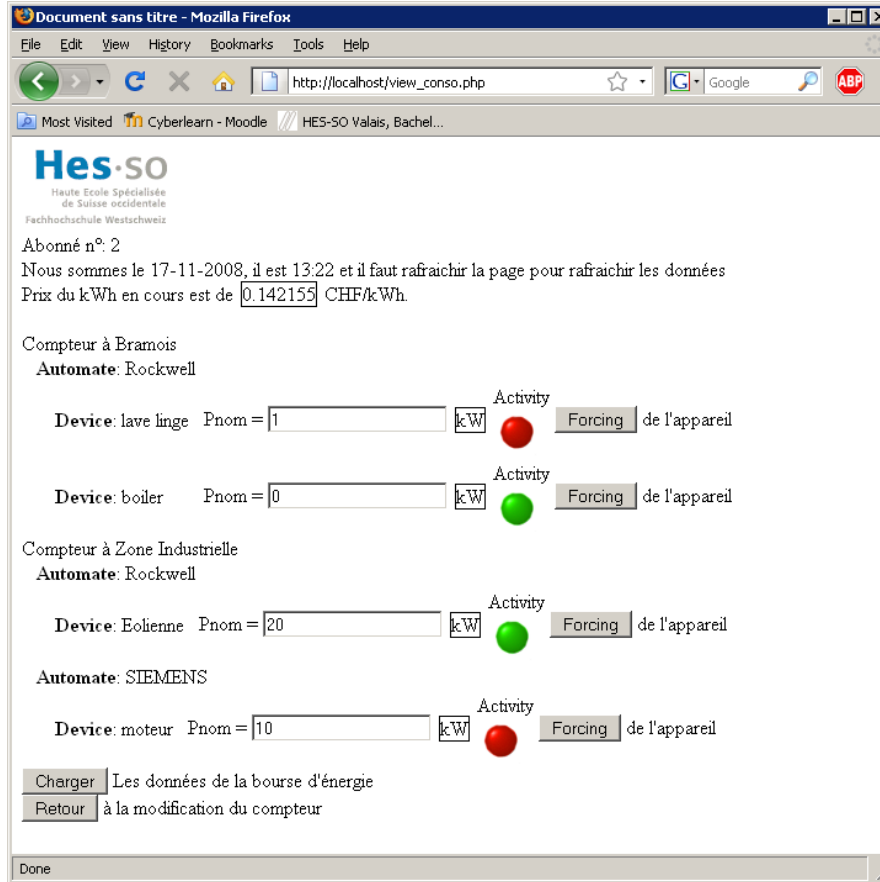


Figure 5.7 : Visualisation consommation

Dès lors, il pourra accéder à une dernière page qui lui permet d'observer le nombre de fois que son distributeur à forcer l'appareil sélectionné à s'enclencher ou se déclencher ainsi que le nombre de fois que lui-même a utilisé le forçage.

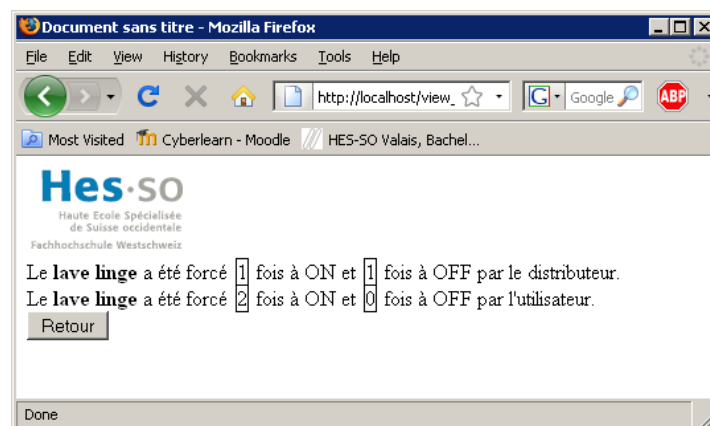


Figure 5.8 : Visualisation forçages

5.1.2 Acquisition des données relatives à la bourse de l'énergie

L'année 2008 qui est en train de se terminer, a vu naître et aboutir des projets d'unification des bourses européennes. Dans ce sens les bourses Powernext et EEX, basées respectivement à Paris et Leipzig, ont décidé de s'unir pour former une seule bourse Spot de l'électricité immatriculée EPEX spot SE. Les deux bourses intégreront toutes leurs activités électriques spot dans cette bourse de l'électricité EPEX Spot SE avant la fin de l'année. Cette nouvelle bourse au comptant (spot) représente une étape majeure sur le chemin vers l'intégration des marchés électriques spot Européens. S'il est avantageux pour le consommateur final de pouvoir choisir entre plusieurs fournisseurs d'électricité, il est également bénéfique pour le développement du marché de gros que les plateformes de négociation se consolident dans la perspective d'un marché européen unique.

Pour les besoins de ce projet il a fallu trouver un site internet sur lequel on pourrait recueillir le prix du kWh qui est donné par la bourse de l'énergie pour les 24 heures à venir. Le site choisi est: <http://eex.com/en/Market Data/Trading Data/Power/Hour Contracts | Spot Hourly Auction/spot-hours-table/> . Ce site obtient ses informations dans une base de données à partir du système commercial de « Power Trading ». A partir de là, les données sont traitées et mises à disposition sur leur site. Le temps de rafraîchissement entre l'obtention du prix du kWh à la bourse et la publication sur le site est de 5 à 15 minutes selon les produits. Dans notre cas, il est intéressant d'utiliser le tableau qui offre le prix du kWh pour chaque heure de la journée un jour à l'avance, le « Swissix Day Ahead spot ». Ce tableau est mis à jour aux alentours de 10h30 tous les jours.

Dans le cadre de ce projet, les données concernant le prix du kWh de ce site doivent être enregistrées dans notre base de données. Un programme en PHP, dont le code est dans le CD en annexe, a été créé pour les enregistrées. Celui-ci récupère le code source du site où sont hébergées les données, le nettoie des balises Html et sélectionne uniquement la colonne de données nécessaire. Pour exécuter le programme, il suffit de cliquer sur le bouton de l'une des visualisations (Charger les données de la bourse). Cela dit, les données devraient aussi être chargées automatiquement depuis le programme en C. Quelques problèmes, décrits plus loin, ont empêché que cette manipulation se fasse automatiquement.

En vu des changements qui vont survenir sur les sites de bourses d'énergies, il faudra revoir le code du programme prévu pour télécharger le prix du kWh. Si le site depuis lequel on télécharge les données change son format, ou si on veut utiliser un autre site pour récupérer le prix du kWh, il faudra modifier le programme en conséquence.

5.1.3 Communication entre le serveur et l'automate

Le choix de la communication entre le serveur et l'automate s'est porté sur un transfert de données à travers le web avec le Protocol Modbus/TCP. Les raisons de ce choix se trouvent dans le travail de semestre à la page 12 chapitre 4.1.1.3 *Modbus*.

5.2 La base de données

La base de données a été développée avec EasyPHP qui est une plateforme de développement Web. Ce programme regroupe deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'une administration SQL PhpMyAdmin. C'est à travers cette administration que les différentes tables de la base de données ont été créées.

Pour l'instant la base de données est gérée localement. C'est-à-dire que la base est accessible uniquement sur la machine porteuse du programme EasyPHP. Pour pouvoir y accéder depuis l'extérieur, il suffirait de la mettre à disposition sur le web avec un autre serveur SQL adapté à internet. On peut interroger la base de données avec des requêtes MySQL standards.

La structure de la base de données est la suivante :

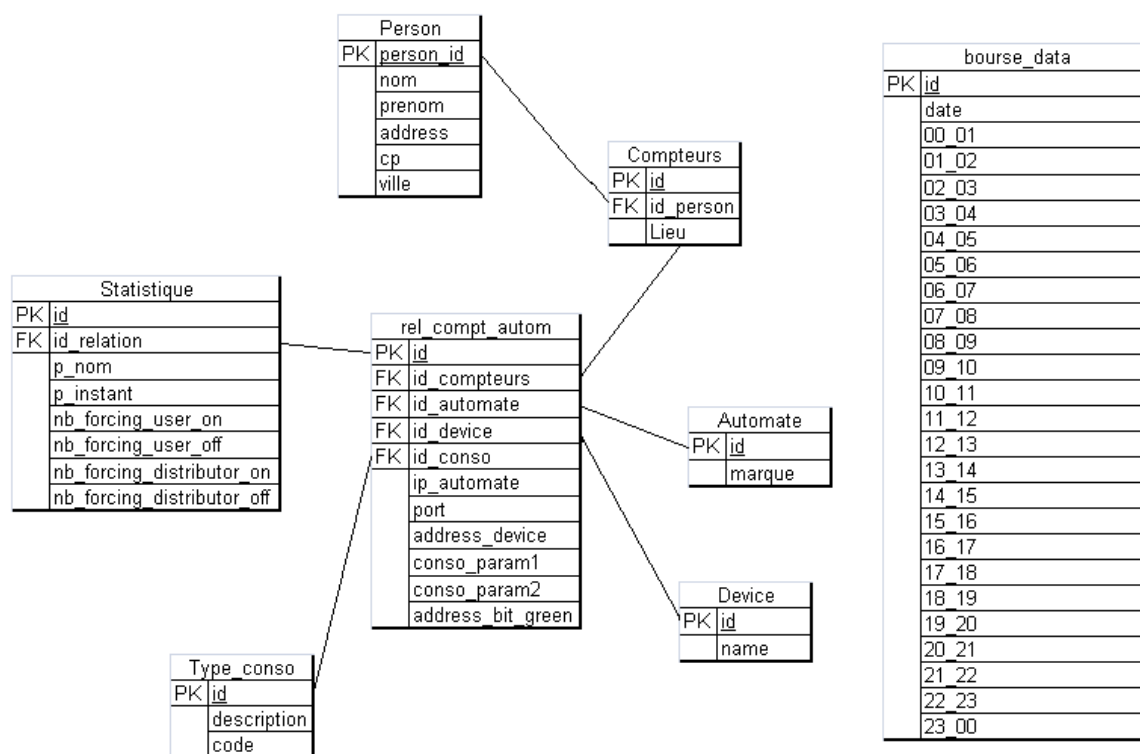


Figure 5.9 : Structure de la base de données

Cette structure nous permet de réaliser toutes les fonctionnalités requises pour ce travail de diplôme. Cela dit, elle n'est pas optimale. Comme on peut le constater, la table rel_compt_autom pourrait être remplacée par deux tables bien distinctes. Une table contiendrait les relations entre les compteurs, automates, appareils et choix de consommations et l'autre contiendrait les informations relatives à chaque appareil (IP de l'automate, adresse de l'appareil dans l'automate, ...). Finalement, on a décidé de ne pas restructurer la base de données. Le temps restreint ainsi que la modification de tout le code (si la restructuration avait été faite) sont deux paramètres qui ont influencé cette décision.

5.3 Le serveur

Le serveur est un simple PC. Il est le support sur lequel s'exécute l'application en C++ qui gère la loi de commutation, les communications avec l'automate et la base de données. Le PC contient le programme EasyPHP dont sont issues les bases de données, et les fichiers contenant les codes sources de l'application. Ce serveur doit gérer les communications entre les différents automates, la base de données et l'application.

5.4 L'automate

Pour ce travail de diplôme l'automate sélectionné est de la marque Wago. Le choix de l'automate c'est produit durant le travail de semestre (voir ch. 4.2.1 *Le matériel* p. 14 du rapport de semestre). Les fonctions principales de l'automate, sont des fonctions de contrôle et d'exécution des ordres donnés par le serveur. Comme la logique de l'automate ne nécessite pas d'être très pointue, il serait préférable pour un futur projet de développer un hardware spécifique pour ce type d'applications. Dans le cadre de ce projet, nous avons limité la recherche à la validation du concept avec des éléments déjà présents sur le marché. Voir chapitre 4. *cahier des charges* à la page 10 du travail de semestre.

5.4.1 *Etape future : développement d'un hardware spécifique*

Dans le cas d'une continuation de ce projet, il serait préférable de créer un automate spécifique pour ce genre d'application car, comme expliqué auparavant, les seules tâches que doit réaliser l'automate sont des tâches de contrôle et d'exécution. La loi de commutation et la base de données sont implémentées dans le serveur, en conséquence, la troisième contrainte à réaliser par l'automate est de communiquer à travers internet pour pouvoir se relier au serveur. L'avantage avec un hardware spécifique est que les développeurs n'auraient pas à attendre que les fournisseurs d'automates leur envoi les dernières mises à jour des logiciels et des librairies.

D'autre part, les avancées technologiques permettent aujourd'hui d'avoir des applications d'envoi de SMS comme moyen de communication pour alarmer le client d'une éventuelle défaillance. Dans le même ordre d'idée, le client pourrait forcer une sortie de son automate à distance en envoyant un message à travers un SMS à son automate. Cette technique est déjà utilisée chez certains constructeurs d'automates comme Siemens ou Comat.

5.5 Programmation des différentes parties

Avant de commencer à développer le système, il a d'abord fallu apprendre à programmer dans des langages que je ne connaissais pas et dans des programmes inconnus jusqu'au début de ce projet. Pour ce faire, j'ai utilisé des sites internet et j'ai surtout demandé à mes collègues qui sont en filière Infotronic de bien vouloir m'aider à débiter dans ces nouveaux langages. Comme ce sera mentionné plus tard, beaucoup de petits problèmes m'ont fait perdre du temps car je n'avais pas les bons réflexes de programmeur pour le débogage des différents programmes.

5.5.1 Choix des langages

Pour pouvoir commencer la programmation, il a tout d'abord fallu définir le langage de chaque programme.

5.5.1.1 Site web

Les pages web ont été créées en HTML associé à du PHP afin de les rendre plus dynamiques. Le HTML étant un langage statique, il affiche les pages désirées, mais il ne permet pas d'interaction avec l'utilisateur. Le PHP apporte cette dynamique qui permet de naviguer entre les pages et de remplir des formulaires en ligne. Une autre possibilité aurait été de programmer du JavaScript, cependant, les conseils de mes collègues plus expérimentés m'ont permis de réaliser qu'il était plus simple d'apprendre à programmer des sites web en PHP.

5.5.1.2 Base de données

La base de données choisie pour le stockage des informations est une base de données relationnelle. Les deux langages les plus courants pour ce type de bases de données sont le TCL (*Transaction control language*, langage de contrôle des transactions) et le SQL (*Structured Query Language*, langage structuré de requêtes). SQL étant le langage de base, c'est envers lui que le choix s'est porté. Toutefois, la base de données n'a pas été créée avec ce langage. La création de la base et de ses tables s'est faite avec une application fournie avec EasyPHP qui s'appelle phpMyAdmin. Cette application permet de réaliser et modifier facilement les tables. Son fonctionnement sera décrit dans le chapitre suivant (5.5.2 *Choix des programmes de développement*).

5.5.1.3 Application

Le but à long terme de l'application étant de pouvoir être utilisable par tous les distributeurs d'énergie, il fallait choisir un langage universel, qui n'est pas de licence propriétaire et qui soit utilisable sous tous les systèmes opératifs. Le choix s'est donc porté sur le C/C++ qui est le langage fréquent pour les programmeurs.

5.5.2 Choix des programmes de développement

Dans ce paragraphe le but est de montrer les programmes avec lesquels ce projet a été réalisé. Ces programmes demandent parfois certaines manipulations précises pour qu'ils fonctionnent correctement. Ces manipulations sont décrites ci-dessous. Il est aussi intéressant de voir l'interface que ces différents programmes possèdent. Comme au début du projet je ne savais pas réellement vers quels programmes me tourner, j'ai demandé conseil auprès de mes collègues d'Infotronic et de mon professeur. Ainsi, j'ai pu m'orienter vers les quatre programmes suivant : Adobe Dreamweaver, EasyPHP, Visual Studio et Codesys.

Le programme Adobe Dreamweaver

Ce programme fait partie de la suite Adobe Creative (version CS3 pour ce projet). Il m'a été recommandé par un collègue de la filière Infotronic qui programme des sites web. Comme on peut le voir dans la figure suivante, il permet de programmer dans tous les langages web les plus répandus (HTML, PHP, JavaScript, XML,...).

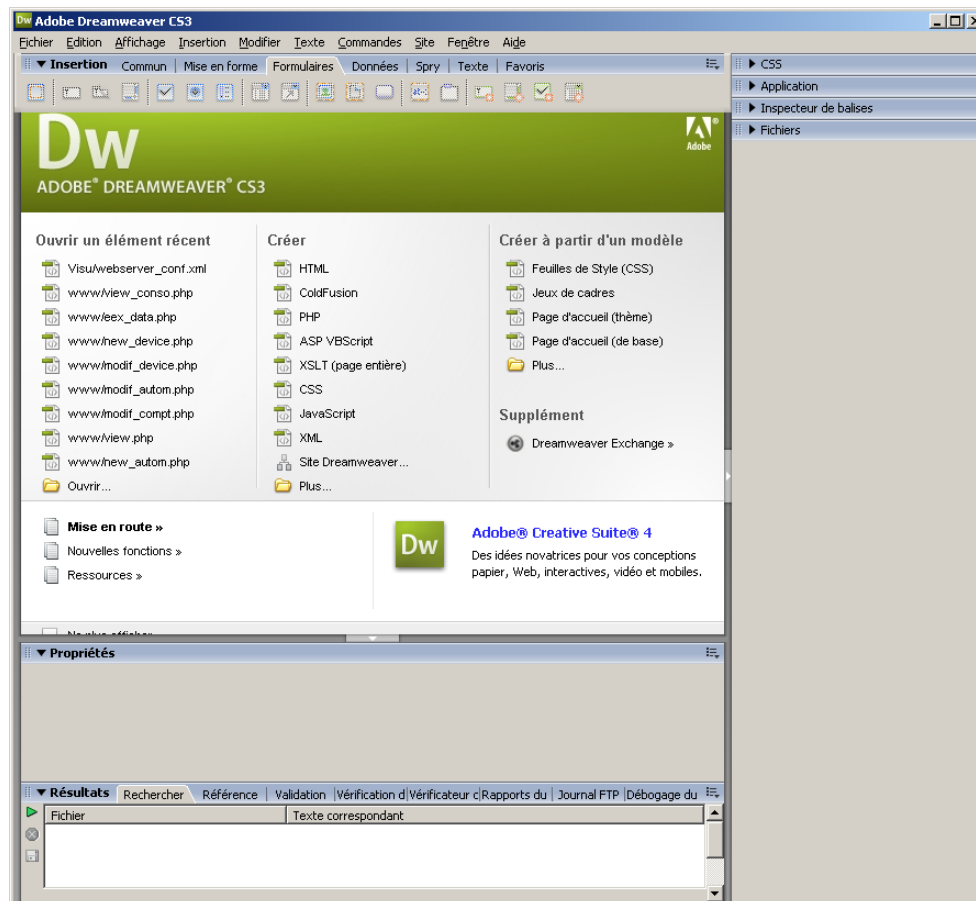


Figure 5.10 : Langages de programmation avec dreamweaver

Ce programme est simple à installer et ne requiert pas de manipulation particulière pour la mise en route d'un projet.

Dans le cadre de ce travail de diplôme, c'est donc pour le PHP que l'on a opté pour programmer le site web. Comme décrit plus haut, le site web est l'interface que possède l'utilisateur pour

Développement d'un interrupteur « intelligent »

s'enregistrer. L'affichage d'une page web se fait en HTML, toutefois pour rendre la page dynamique et pouvoir interagir avec, cette dernière doit disposer d'un autre langage, qui sera interprété par un serveur installé sur la machine (dans notre cas il s'agit d'Apache, qui sera expliqué au paragraphe suivant). J'ai choisi de prendre le PHP plutôt que du JavaScript car mes collègues d'Infotronic m'ont informé qu'il était plus facile à apprendre.

Après les premières démarches hasardeuses avec ce programme, la méthode de travail devient de plus en plus facile. De plus, comme on peut le voir ci-après, il est possible de partager la fenêtre pour pouvoir coder dans une partie et avoir la partie visuelle dans l'autre.

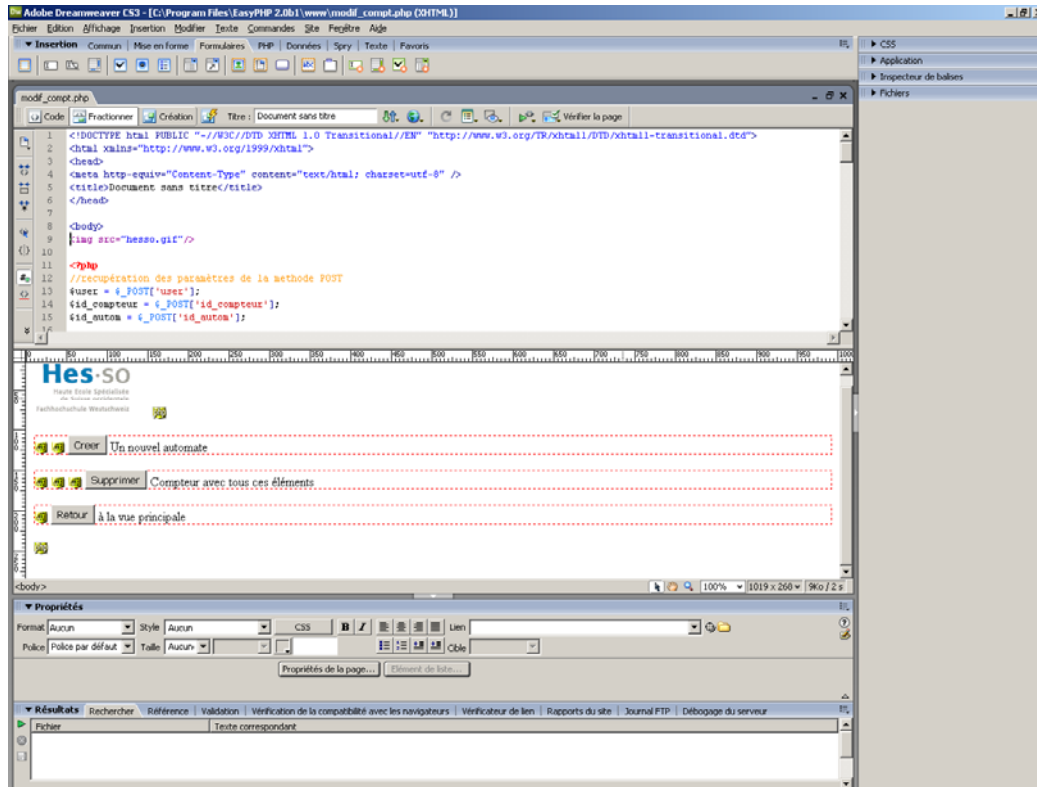


Figure 5.11 : visualisation lors de la programmation avec Dreamweaver

Ainsi, s'il faut ajouter des champs de formulaire, il suffit de sélectionner le bouton correspondant dans la barre d'outils, le placer graphiquement où l'on désire et le code apparaît dans la partie haute de la fenêtre. De la même manière, il est possible de coder manuellement le texte qui apparaîtra sur le bouton et on le verra changer dans la partie du bas.

Une fois le code terminé, il suffit de placer le code source de la page dans le répertoire du serveur web. Pour la visualiser, il suffira finalement de noter *http://localhost/* sur une URL, puis d'y ajouter le nom de la page. Par exemple, *http://localhost/new_compt.php*.

EasyPHP est dans notre cas le programme contenant le serveur web qui permettra d'afficher la page web sur internet. Ci après nous procéderons à la description de ce dernier.

Le programme EasyPHP

Comme son nom l'indique, EasyPHP permet de simplifier le travail sur des fichiers au format PHP. En effet, ce format nécessite d'être interprété, car un simple navigateur Web ne pourra pas le faire. EasyPHP vous offre donc la possibilité de travailler dans un environnement serveur complet. L'application comprend un serveur Apache, une base de données MySQL, la dernière version de PHP, et tous les outils nécessaires afin de pouvoir travailler sur le code PHP ainsi que de créer un site.

Ce programme se télécharge et s'installe très facilement. Lors de l'exécution du programme, ce dernier démarre les deux serveurs, Apache et MySQL. Apache est le serveur http le plus répandu sur internet et MySQL est un serveur de bases de données relationnelles SQL. Toutefois, pour mieux gérer les bases de données, EasyPHP fournit une interface graphique, nommée phpMyAdmin, qui nous aide à créer des bases de données.

phpMyAdmin

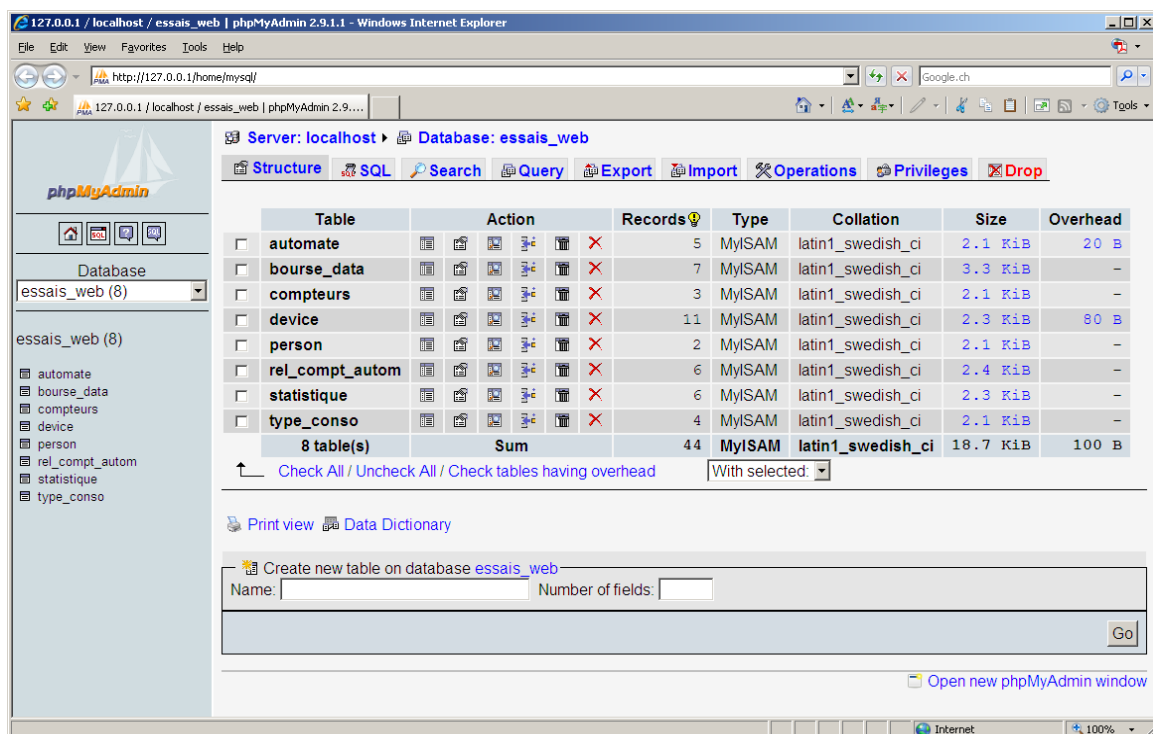


Figure 5.12 : Visualisation de phpMyAdmin

Comme on peut le constater sur l'image ci-dessus, la prise en main de phpMyAdmin est très intuitive. Pour créer une nouvelle table, il suffit de lui donner un nom, ainsi que le nombre de colonnes et d'appuyer sur la case Go au fond de la fenêtre. Si on veut en effacer, il suffit de cliquer sur la croix de la ligne correspondante à la table à effacer. Pour accéder à une table, il suffit de cliquer sur le nom de la table désirée qui se situe sur la partie gauche de l'application.

L'environnement de développement Visual Studio

Visual Studio est un environnement de développement intégré (IDE). Pour ce projet, la version 2005 a été utilisée. Ce programme m'a été recommandé par le professeur, M. Marquard, qui supervisait ce travail de diplôme. Alors que l'installation de ce logiciel n'a posé aucun problème, la programmation s'est avérée plus problématique. Certains problèmes concernant les liens avec des dll ou des librairies externes au projet ont été rencontrés. Lorsqu'il fallait compiler, le linker ne trouvait pas les librairies standards ou il ne savait pas où chercher la dll utilisée dans certaines parties du code. Pour trouver des solutions à ces ennuis, il m'a fallu contacter un collègue plus expérimenté dans la programmation en C/C++. Ensemble on a pu forcer le linker à prendre les liens requis en lui indiquant où les dossiers se trouvaient. Dans la figure suivante, on aperçoit cette fenêtre sur laquelle on réalise ce forçage.

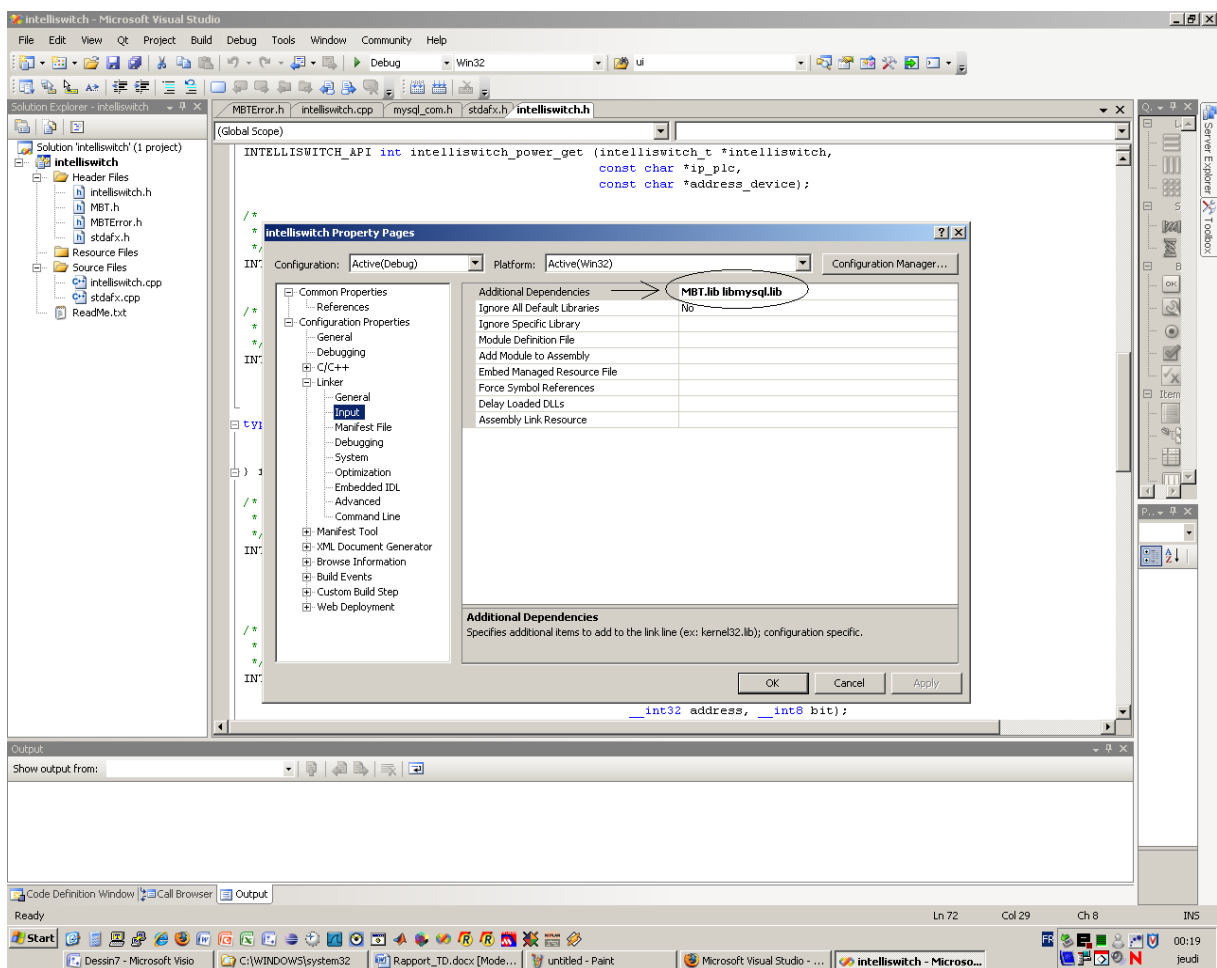


Figure 5.13 : Visualisation du linker de Visual Studio

Une autre astuce qui a été faite pour pouvoir compiler intelliswitch.dll fut de placer la MBT.dll dans le dossier C:\WINDOWS\system32. Vu la méthode de sauvegarde des trois parties développées (intelliswitch, intelliswitch_debug et intelliswitch_gui), lors de la réouverture des projets (.sln) toutes ces astuces sont sauvegardées dans l'environnement de Visual Studio. Il est nécessaire que les trois dossiers soient sauvegardés au même niveau hiérarchique car les liens entre eux sont définis par des acheminements relatifs.

Développement d'un interrupteur « intelligent »

Tous ces petits embarras prennent du temps pour les détecter et pour les résoudre. Finalement, on a obtenu une base de travail pour la personne qui reprendra ce travail de diplôme pour le continuer.

Le programme Codesys

Codesys est le nom de l'environnement de développement fourni par Wago pour créer le programme qui tournera sur l'automate. L'installation s'est accomplie sans problème et la programmation aussi.

Avant de commencer à programmer, il est nécessaire de configurer la cible où se chargera le programme. Pour ce faire, Wago met à disposition une application appelée « Wago-EthernetSettings ».

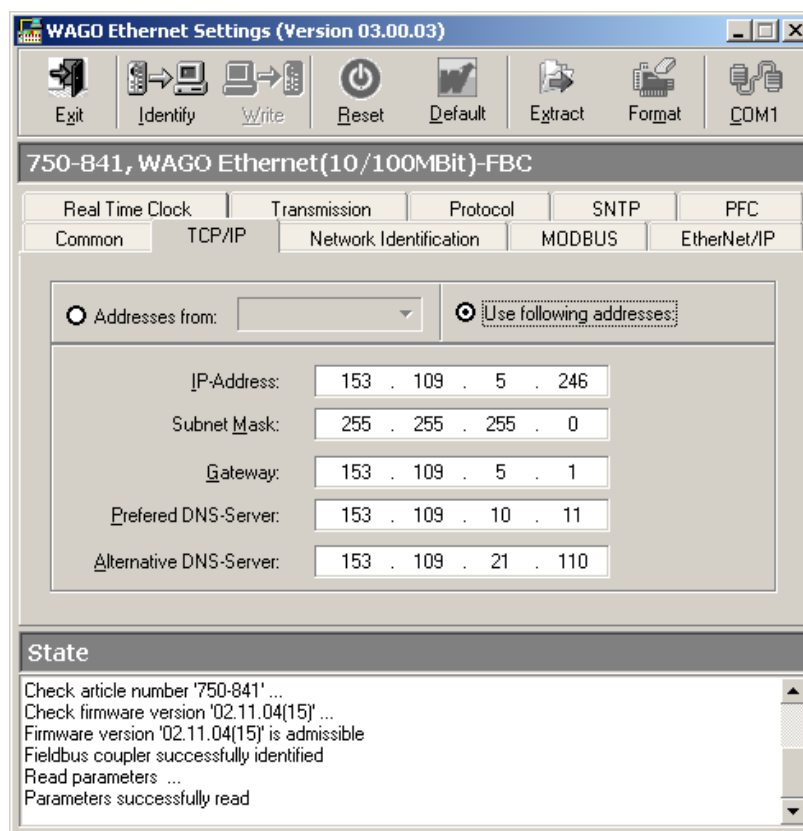


Figure 5.14 : Configuration de l'automate cible

Il y a seulement deux pas à suivre pour configurer l'automate cible. La première est de définir les paramètres TCP/IP comme on peut le voir dans la figure ci-dessus. La deuxième est la synchronisation de l'horloge du système avec celle de l'automate (onglet « Real Time Clock »). Une fois ces deux manipulations effectuées, il faut cliquer sur Extract puis sur Format pour enregistrer les changements (« Format » n'efface pas le contenu, mais le met en forme).

Développement d'un interrupteur « intelligent »

L'environnement de Codesys est assez simple, comme on peut le voir sur la figure suivante.

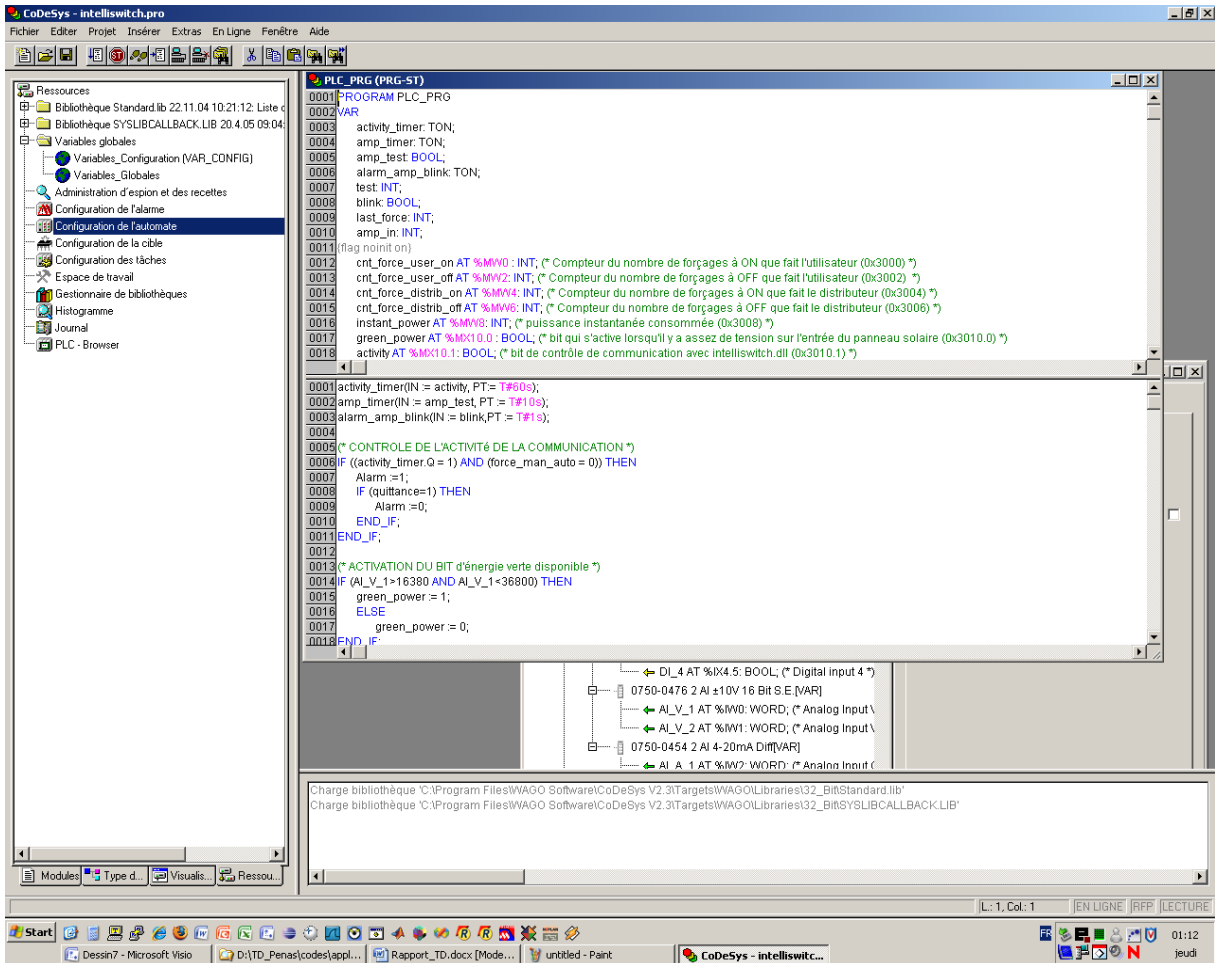


Figure 5.15 : Prise en main de Codesys (Wago)

La fenêtre principale (divisée en deux parties) contient le programme (affiché dans la partie du bas) et les variables (affiché dans la partie du haut). Au début, il faut bien se mettre dans la philosophie des développeurs de chez wago pour comprendre comment fonctionne l'adressage des entrées/sorties. A noter qu'en cas de difficultés, l'assistance technique de la marque est compétente, effective et toujours disponible.

De plus, le manuel d'utilisation de l'automate est très complet et permet de s'acclimater rapidement à l'environnement de développement. Il contient toutes les étapes à suivre pour la mise en service de l'automate, ainsi que le guide des premiers pas à suivre dans l'environnement de Codesys.

5.5.3 Développement du système

Le système comprenant la totalité des éléments (automate, maquette, PC) est un prototype. Comme tout prototype, il y a beaucoup de choses qui peuvent être améliorées. Cela dit, il peut servir de base pour de futurs projets d'améliorations.

5.5.3.1 Développement du site Web et de la base de données

Dans un premier temps, il est important de savoir quels éléments le distributeur nécessite, afin que le client puisse s'inscrire dans la base de données. Pour commencer, il faut que le client s'enregistre et qu'il reproduise son système dans la base de données à travers le site web mis à disposition. Ensuite le système doit être prévu pour un client qui aurait une grande installation, voir plusieurs installations. Les systèmes doivent pouvoir se modifier si le client ajoute, modifie ou supprime un élément de son installation. Finalement, il faut que le client puisse voir l'état de ses appareils et si le système proposé lui est bénéfique, c'est-à-dire si son distributeur à souvent, ou peu, utiliser la flexibilisation d'enclenchement des appareils en question.

Pour cela, il faut créer pour le client des formulaires prêts à remplir et des pages de visualisation. Il ne manque plus qu'à savoir quel champs il faut utiliser dans ces formulaires, à quel endroit les placer et créer des boutons pour naviguer entre les différentes pages.

Une fois les champs nécessaires définis, il faut écrire le code et faire le design du site. Le programme Dreamweaver permet, comme expliquer dans le chapitre précédent, de visualiser le code et de l'afficher en même temps. Ainsi, il est possible de créer parallèlement les champs du formulaire à remplir graphiquement dans la partie création, et de continuer à écrire le code PHP qui dynamise l'affichage dans la partie code.

Mais avant de se lancer dans l'écriture du code, il faut établir la structure de la base de données. Si cette étape n'est pas réalisée correctement, il y a de fortes chances que, lors de l'écriture du code, surviennent des problèmes d'accès à la base. Ces problèmes peuvent être, par exemple, de ne plus pouvoir accéder à une donnée parce qu'elle est inatteignable ou de remplacer des champs qu'il ne fallait pas, en faisant une requête.

Une fois tous ces points éclaircis, l'écriture des pages se fait plus facilement. Le code est dans le CD annexe. Il contient des commentaires qui expliquent le déroulement de chaque partie.

5.5.3.2 Développement de l'application en C/C++

Comme pour tous les programmes, la première étape consiste à définir les buts de l'application à développer. L'application en C/C++ doit permettre de communiquer avec la base de données et avec la dll (dynamic link library) fournie par Wago. Cette dll permet la communication via Modbus/TCP (donc sur un câble réseau RJ45 et à travers le réseau internet) entre le serveur et l'automate. L'application est divisée en deux parties, une dll et une interface graphique, qui sera celle du distributeur. La première partie est une définition et implémentation des fonctions qui pourront être utilisées par l'interface graphique utilisateur (GUI). La deuxième partie est justement ce GUI. Le schéma suivant représente la structure de base de l'application.

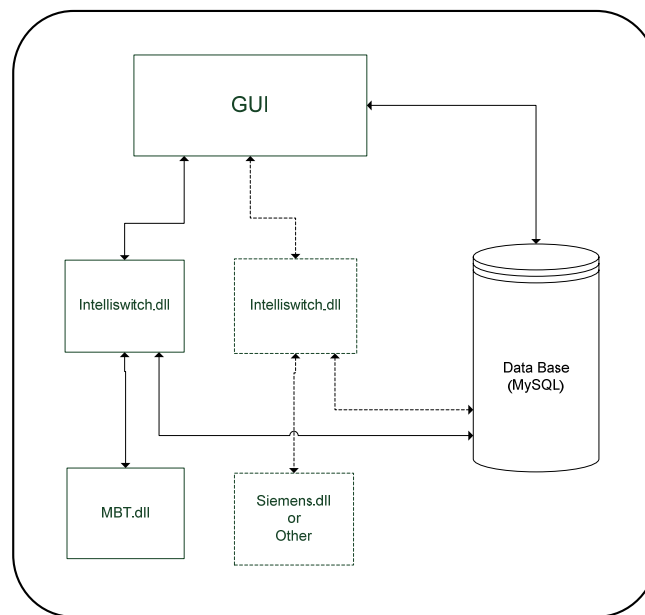
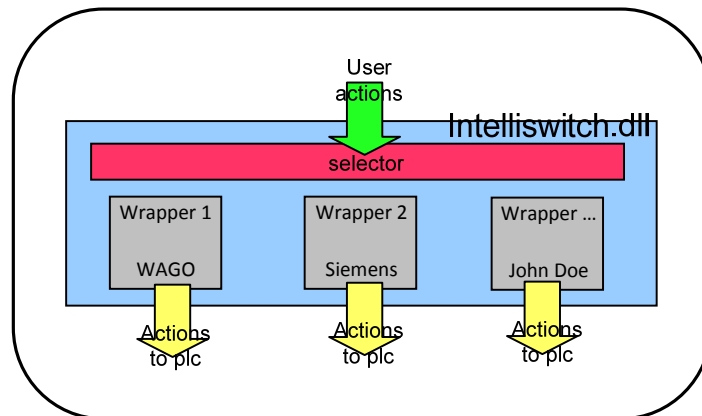


Figure 5.16 : Structure application

Idéalement la dll créée, doit être utilisable avec n'importe quelle dll de communication vers les automates. Cela dit, faute de temps et d'expérience dans la programmation, elle n'est utilisable qu'avec celle de Wago pour l'instant. Pour qu'elle soit compatible avec les autres marques d'automates, il faudrait implémenter la librairie intelliswitch.dll comme une couche d'abstraction indépendante des marques. Chaque marque serait donc représentée sous la forme d'un wrapper accessible uniquement par l'intermédiaire d'un sélecteur.

Développement d'un interrupteur « intelligent »



Le schéma ci-dessus représente l'idée du wrapper. La librairie `intelliswitch.dll` est représentée en bleu. Elle englobe le tout. Les wrappers sont intégrés à la librairie, et c'est la partie rouge (le sélecteur) qui permet de faire l'abstraction entre les actions de l'utilisateur (la flèche verte) et les actions réellement envoyées aux automates (les flèches jaunes).

Le principe de la structure est de créer une instance (un « handle ») via `intelliswitch.dll` pour chaque automate qu'on veut contrôler. Cette création se fait automatiquement depuis le GUI, qui sera expliqué plus loin.

Les fonctions de la dll doivent être choisies en fonction de ce qu'on veut voir sur le GUI et de ce qu'on veut faire depuis l'interface graphique aux automates. Ainsi, les fonctions de bases sont :

- Initialisations et fermeture de la communication avec l'automate
- Chercher la liste des automates et appareils dans la base de données
- Recevoir ou imposer l'état d'une sortie de l'automate
- Recevoir la puissance consommée par l'appareil connecté à l'automate

Cela dit, deux fonctions sont prioritaires sur les autres, à savoir `Intelliswitch_preinit()` et `intelliswitch_exit()`. Elles initialisent et terminent respectivement l'accès à la dll.

Comme décrit auparavant, la deuxième partie de l'application est l'interface graphique du côté distributeur. Voici, ci-après, le schéma de principe de ce GUI.

Développement d'un interrupteur « intelligent »

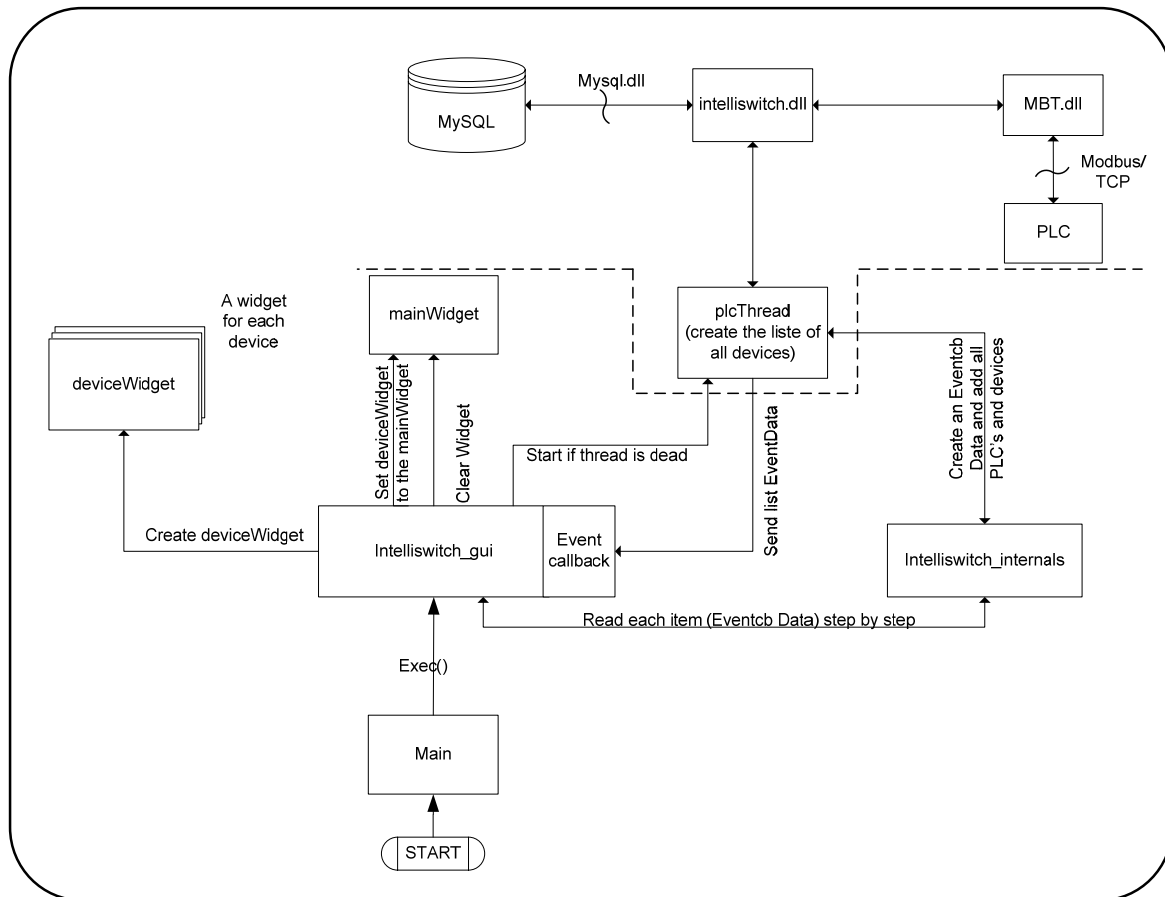


Figure 5.17 : Structure visualisation distributeur

Le fonctionnement de l'interface graphique est le suivant :

Au lancement du programme, on entre dans le Main. Se crée alors la fenêtre de base, qui s'affiche à l'écran. Dès lors, se démarre la boucle événementielle qui permet d'agir sur la fenêtre (exec()).

Ensuite on rentre dans la boucle principale par intelliswitch_gui. Dans ce bloc, on commence par construire le container qui aura les widget dans la fenêtre, le thread (processus parallèle) et le timer. Pour mettre à jour les widgets, on commence par effacer les anciens du container, puis on va chercher un par un les nouveaux dans intelliswitch_internals et si le thread est mort, on le réinitialise. Le thread est un processus parallèle qui permet de communiquer avec la dll qu'on a créé auparavant. Il demande à la dll de lui fournir la liste des automates et des appareils et les stocks dans une structure de données dans intelliswitch_internals. Une fois toutes les données reçues, le thread envoie l'adresse du pointeur vers les données à l'event callback de l'intelliswitch_gui. Dès que celui-ci reçoit cette information, il efface les anciens widget de la fenêtre et la remplit à nouveau avec les nouvelles données stockées.

Le grand problème posé par ce système est le temps d'attente lorsque la dll veut se connecter à un automate qui n'est pas actif. Au début, il était prévu que ce soit directement l'intelliswitch_gui qui fasse les requêtes à la dll, mais ses temps d'attente faisaient geler la fenêtre (la fenêtre se bloque et on ne peut plus rien faire). Pour éviter cela, il a donc fallu créer un processus parallèle qui irait faire ces requêtes à la place. A présent la fenêtre ne se gèle plus, mais le temps de rafraichissement des

Développement d'un interrupteur « intelligent »

données affichées est long. Le problème vient du fait que lors de la connexion à un automate, la dll attend que celui-ci lui réponde pendant un très long temps. Ce temps d'attente se reporte sur le thread qui attend à son tour les réponses. En conséquence, l'intelliswitch_gui ne modifie pas la liste affichée dans la fenêtre.

Pour éviter ce problème, il existe deux solutions. La première serait de créer un superviseur qui recueille (sans être bloquant) toutes les requêtes qui doivent être passées à intelliswitch.dll. Celui-ci les traite les unes après les autres dans le cas où elles concernent un même automate, sinon en parallèle. Ainsi, même si un automate reste bloqué (timeout), les autres peuvent agir en parallèle. La deuxième serait de refaire l'intelliswitch.dll en la créant Thread Safe. C'est-à-dire que plusieurs threads pourraient accéder en même temps à toutes les fonctions publiques de la dll. Cela dit, il faudrait s'assurer que toutes les bibliothèques utilisées par intelliswitch.dll le soient aussi, car dans le cas contraire ce seront ces bibliothèques qui empêcheront le système de fonctionner correctement.

En voyant que l'interface graphique ne fonctionnerait pas, un programme contenant l'algorithme de commutation a néanmoins été réalisé (intelliswitch_debug). Cet algorithme permet de démontrer que le concept d'interrupteur intelligent est réalisable. Le structogramme d'intelliswitch_debug se situe en annexe 3.

Par faute de temps, la partie qui devait télécharger les données de la bourse de l'énergie dans la base de données n'a pas pu être finalisée. Pour télécharger ces données, il faut le faire à la main en appuyant sur le bouton correspondant dans la visualisation du système de l'utilisateur (site web).

Chaque partie de code est commentée pour expliquer à quoi servent les méthodes et le code source se trouve dans le CD annexe à ce rapport.

Pour pouvoir faire fonctionner les dll ainsi que les différents programmes, il faut pouvoir exécuter les codes sources avec des programmes adaptés. Durant ce travail de diplôme, le temps restreint n'a pas permis de créer les fichiers exécutables nécessaires à l'exécution des programmes sur une machine vierge.

Développement d'un interrupteur « intelligent »

5.5.3.3 Développement du programme de l'automate

Le programme implémenté dans l'automate est assez simple. Pour le coder il a fallu tenir compte des fonctionnalités gérées par l'automate et qui avait été définies dans le chapitre 2.2 *Analyse fonctionnelle* à la page 6 du travail de semestre.

Dans notre cas, ces fonctions sont :

- Contrôle de l'état de la communication
- Contrôle de l'état du producteur
- Contrôle de consommation de la charge
- Contrôle des forçages
- Calcul de la puissance produite par le producteur
- Calcul de la puissance consommée par la charge

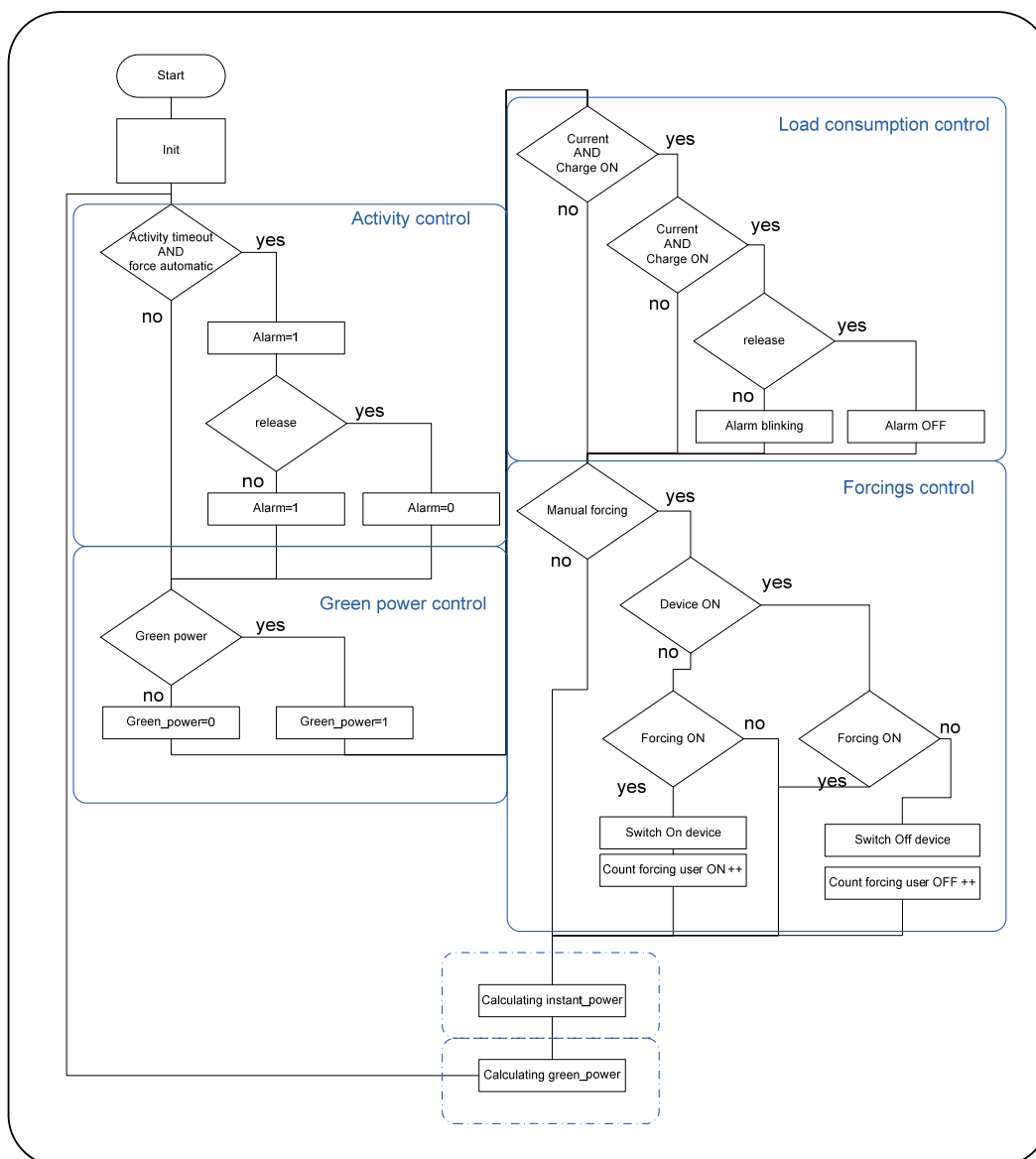


Figure 5.18 : Structogramme du programme de l'automate

Développement d'un interrupteur « intelligent »

Grâce au structogramme ci-dessus, il est plus aisé de coder le programme qui tournera sur l'automate. Cela dit, la programmation sur les automates peut se faire avec des langages différents, mais je ne m'étendrai pas là dessus (voir ch.8 programmation des API du cours MCR2). Comme il y a beaucoup de conditions dans cette application, le Structured text me semble le plus approprié. L'implémentation des conditions dans ce langage requiert moins d'effort que dans les autres.

Pour calculer la puissance que délivre le panneau, on a mesuré le courant et la tension maximale qu'il délivrait, puis on a simplifié les calculs pour la simulation en choisissant que la puissance était proportionnelle à la tension.

Le calcul de la puissance instantanée se fait aussi en simulant une vraie charge. Ainsi, lorsque la lampe qui simule le boiler est éteinte on a 0W et lorsqu'elle est allumée on obtient 162 W.

Dans le cas présent, si la communication est interrompue, il n'y a qu'une alarme qui s'enclenche. Cela dit, il serait préférable que l'automate réagisse à cette alarme. La réaction ne devra pas être la même si l'alarme survient dans un automate qui a juste un boiler qui y est connecté que si elle arrive sur un automate où sont connectés des appareils dangereux. De la même manière qu'il y a des bits ou des mots réservés pour le contrôle de l'activité, le calcul de la puissance, ... on pourrait aussi prévoir de réserver un espace mémoire pour un code de comportement en cas de coupure.

Comme pour les autres parties du développement, le code est commenté avec les explications de ce que fait chaque bout de programme.

Le code de cette partie est dans le CD annexe.

6 Conclusions

6.1 Analyse des problèmes rencontrés

Plusieurs problèmes sont apparus tout au long du projet. Certains ont pu être résolus, alors que d'autres sont restés irrésolus. Les petits problèmes résolus ont été de trouver les astuces nécessaires à faire fonctionner Visual Studio par exemple. Une autre difficulté qui s'est résolue au fur et à mesure que je prenais de l'expérience fut l'habileté à déboguer les morceaux de code avec plus d'efficacité. Les erreurs ont pu être trouvées dans un laps de temps plus restreint.

Mais le problème le plus récurrent a été le manque d'expérience dans les différents langages de programmation. Bien que la formation d'ingénieur en électrotechnique contienne un cours de programmation, les notions apprises lors de ce cours ne sont pas suffisantes pour attaquer un projet de cette envergure. Ce manque d'expérience a finalement eu comme conséquence de ne pas avoir pu finir l'interface graphique de l'application en C/C++.

6.2 Synthèse

Malgré ces problèmes, le prototype a pu être réalisé. Bien que le système ne soit pas définitif, la maquette réalisée permet de valider le concept d'interrupteur intelligent. Il était prévu que le serveur ait une interface graphique, mais en vu des problèmes mentionnés ci-dessus elle n'a pas pu

être terminée. À sa place, la loi de commutation a été réalisée sur une console de commandes, moins jolie, mais qui permet la validation du concept. Néanmoins, toute la partie de configuration du système et l'enregistrement de l'utilisateur sur la base de données sont terminées.

6.3 Evolutions futures

Bien entendu, comme tout prototype, ce projet devra être repris et amélioré dans le but de parvenir à des tests réels. D'ailleurs, toute la programmation de la partie en C/C++ pourrait être réalisée en un projet à part entière par une personne de la filière Infotronic. Un autre projet pourrait être de réaliser un hardware spécifique à ce type d'applications. Ainsi, les programmes pourraient être simplifiés. Intelliswitch.dll, par exemple, n'aurait plus besoin de wrapper car il n'y aurait qu'un seul type d'automates avec lequel communiquer.

6.4 Apport personnel

Personnellement, j'ai trouvé ce projet très intéressant car les énergies renouvelables m'ont toujours interpellé. Il est aussi vrai que les difficultés rencontrées tout au long du projet m'ont parfois baissé le moral. Cela dit, je souhaiterais devenir un producteur d'énergie renouvelable dans le futur et ce projet me paraît une excellente option. Je suivrais donc avec intérêt l'évolution de ce projet.

6.5 Remerciements

Je tiens à remercier toutes les personnes qui m'ont aidé et suivis tout au long de ce projet. Les résultats obtenus ne seraient sans doute pas les mêmes sans leur collaboration. Mes remerciements vont particulièrement à mes collègues de la filière Infotronic Petrovic Darko pour son aide dans la programmation des pages web ; ainsi que Schroeter Mathieu pour son aide précieuse avec la programmation en C/C++. Je termine en remerciant mon professeur Marcuard Jean-Daniel pour son coaching tout au long du projet.

Sion le 21 novembre 2008

Penas Victor

7 Références

7.1 BIBLIOGRAPHIE :

Php & MySQL	Luke Welling & Laura Thomson	CampusPress
C en action	Yves Mettier	O'REILLY
Manuel Wago		
Bulletin electrosuisse 18/2008		

7.2 SITES INTERNET :

www.lesiteduzero.com	pour apprendre php, mySQL
www.developpez.com	site de tutoriaux, codes sources
www.codes-sources.com	site de tutoriaux, codes sources
http://www.powernext.fr/modules.php?op=modload&name=PwnNews&req=actualite&file=article&sid=1017&bid=1&cld1=2&cld2=-1&cld3=-1	communiquer de presse nouvelle bourse énergie
http://www.powernext.fr/modules.php?op=modload&name=Powernext&file=index&req=ahead&bid=9&cld1=10&cld2=-1&cld3=-1	fixing prices in powernext platform
www.eex.com	bourse européenne de l'énergie à Leipzig
http://msdn.microsoft.com/fr-fr/visualc/msdn.coachc++.atelier4.aspx	Formation interactive C++

8 Table des figures

Figure 4.1 : Démonstrateur	7
Figure 5.1 : Schéma de base du système	9
Figure 5.2 : Page d'accueil	10
Figure 5.3 : Page d'enregistrement d'un nouveau client	10
Figure 5.4 : Première visualisation principale	11
Figure 5.5 : Création d'un nouveau compteur	12
Figure 5.6 : Visualisation système	12
Figure 5.7 : Visualisation consommation	13
Figure 5.8 : Visualisation forçages	13
Figure 5.9 : Structure de la base de données	15
Figure 5.10 : Langages de programmation avec dreamweaver	18
Figure 5.11 : visualisation lors de la programmation avec Dreamweaver	19
Figure 5.12 : Visualisation de phpMyAdmin	20
Figure 5.13 : Visualisation du linker de Visual Studio.....	21
Figure 5.14 : Configuration de l'automate cible.....	22
Figure 5.15 : Prise en main de Codesys (Wago)	23
Figure 5.16 : Structure application	25
Figure 5.17 : Structure visualisation distributeur.....	27
Figure 5.18 : Structogramme du programme de l'automate	29
Figure 9.1 : Page de login	35
Figure 9.2 : Page de visualisation des données personnelles	35
Figure 9.3 : Page de modification de compteur	36
Figure 9.4 : Page de nouvel automate	36
Figure 9.5 : Page de modification d'automate	37
Figure 9.6 : Page de nouveau device	37
Figure 9.7 : Page de modification de device	38

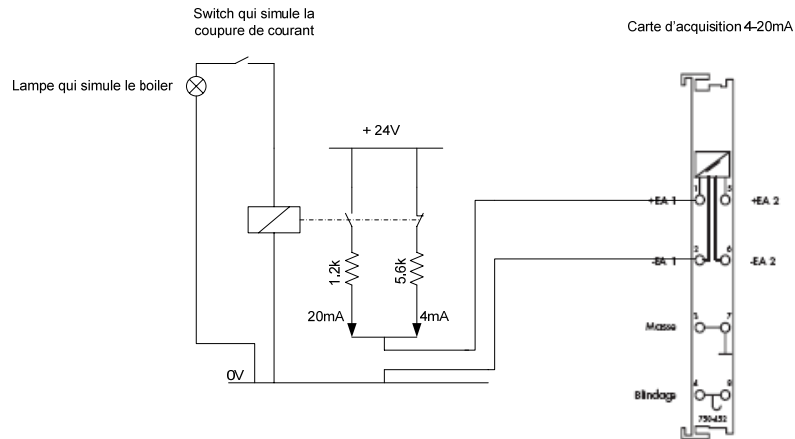
9 Annexes

Table des annexes

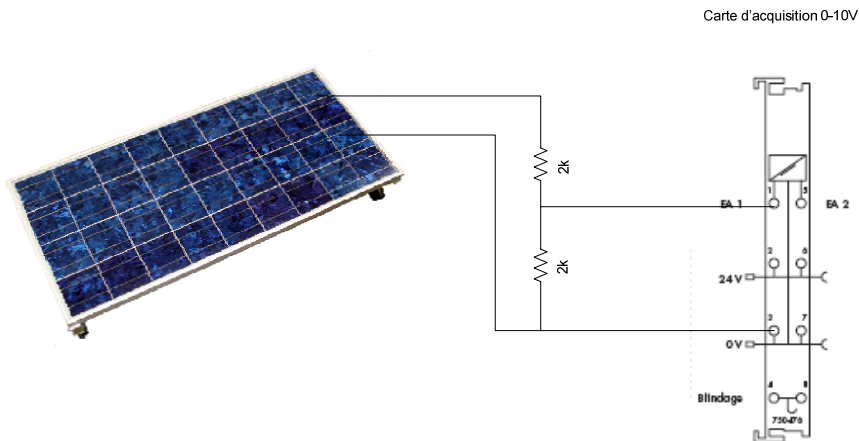
9.1	SCHÉMAS ÉLECTRONIQUES DE LA PLAQUE D'EXPÉRIMENTATION.....	34
9.2	VISUALISATION DE TOUTES LES PAGES WEB	35
9.3	STRUCTOGRAMME INTELLISWITCH_DEBUG	39
9.4	CONTENU DU CD.....	40

9.1 Schémas électroniques de la plaque d'expérimentation

Adaptation à la carte d'acquisition 4-20mA



Adaptation à la carte d'acquisition 0-10V



9.2 Visualisation de toutes les pages web

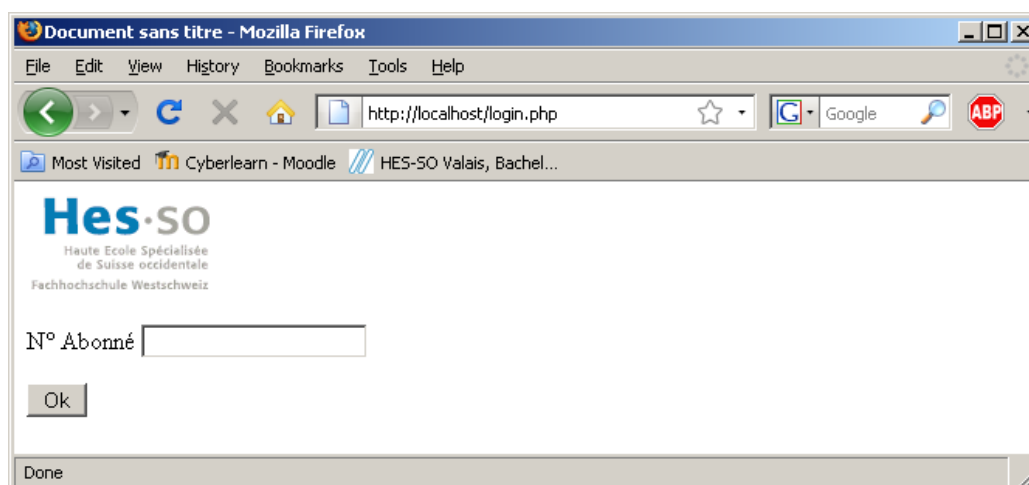


Figure 9.1 : Page de login

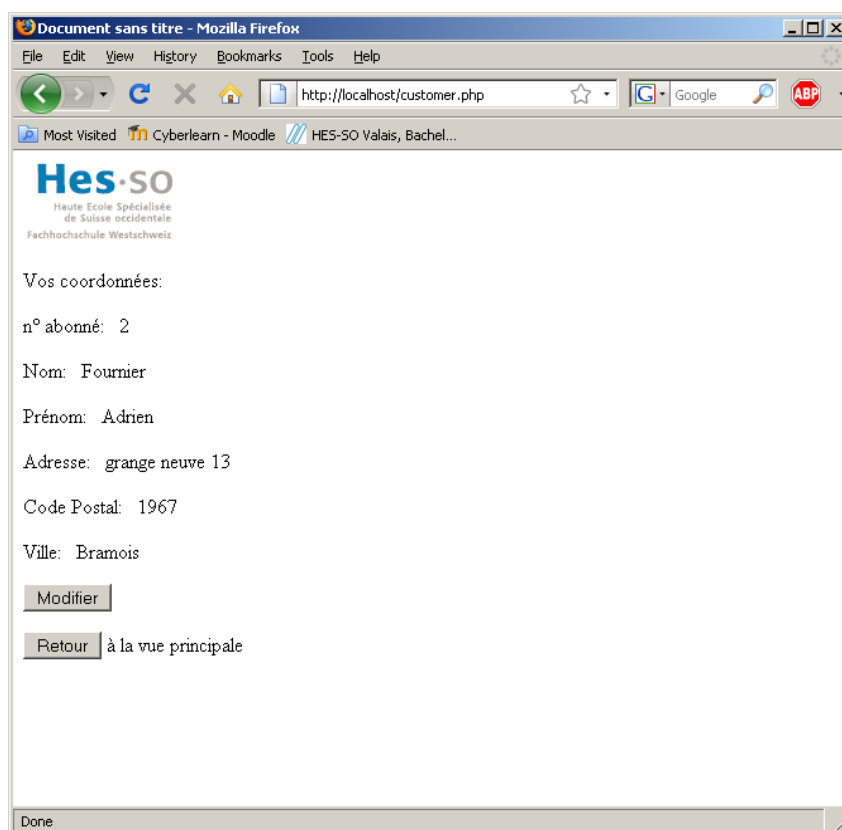


Figure 9.2 : Page de visualisation des données personnelles

Développement d'un interrupteur « intelligent »

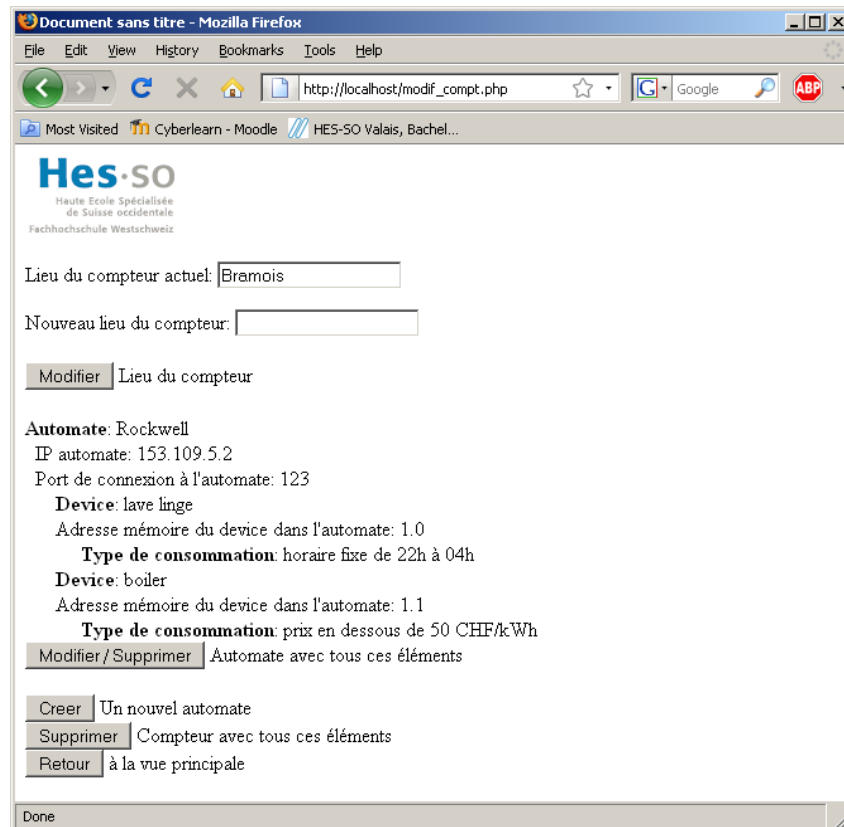


Figure 9.3 : Page de modification de compteur

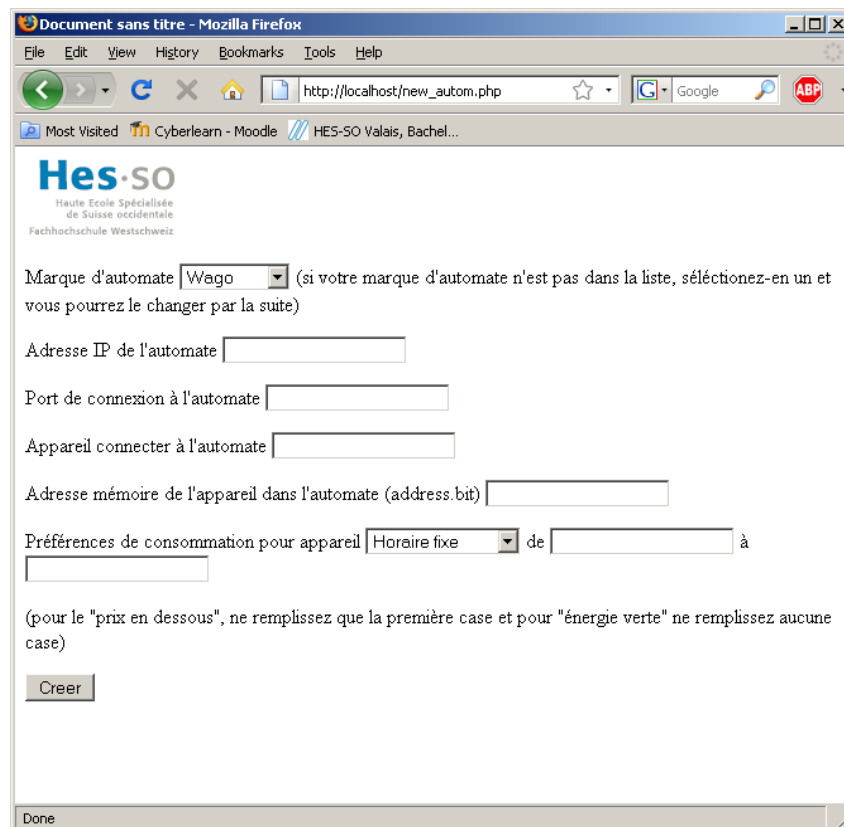


Figure 9.4 : Page de nouvel automate

Développement d'un interrupteur « intelligent »

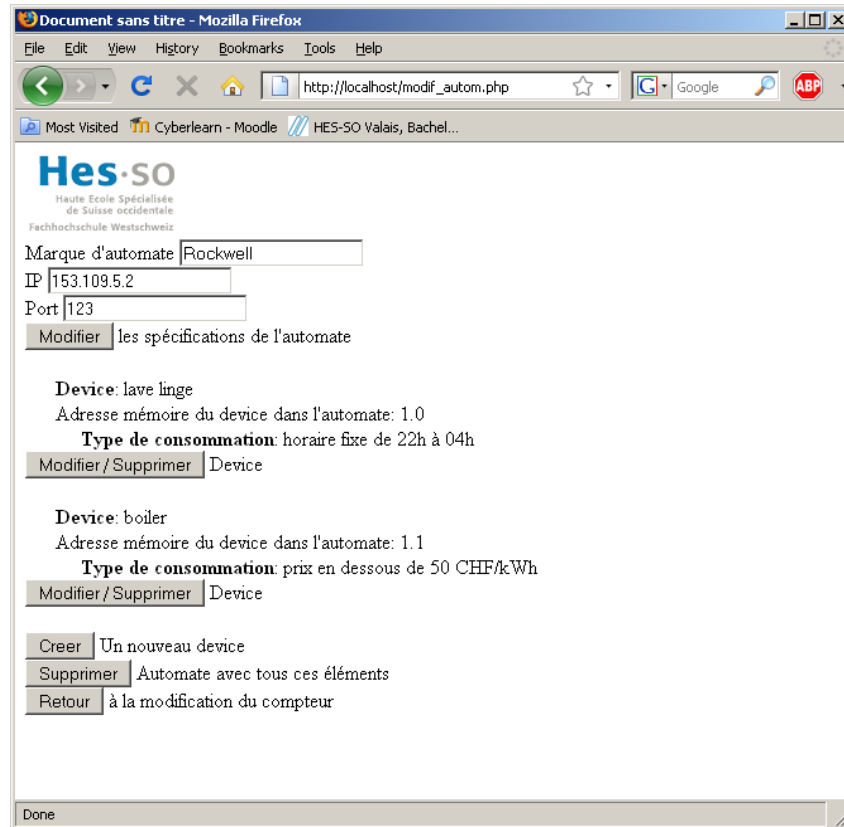


Figure 9.5 : Page de modification d'automate

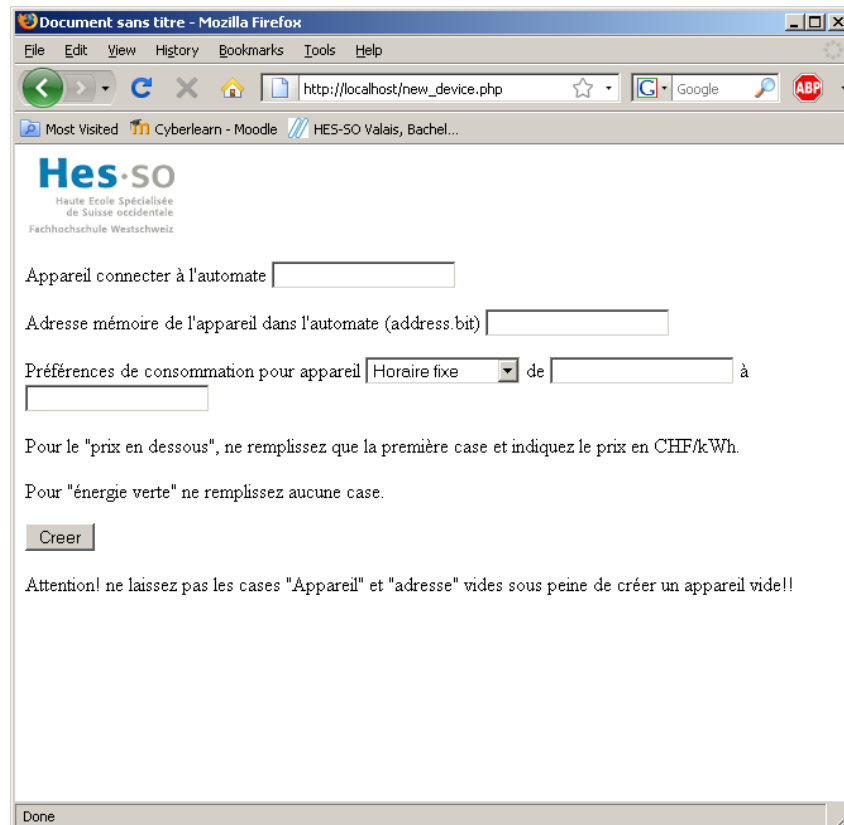


Figure 9.6 : Page de nouveau device

Développement d'un interrupteur « intelligent »

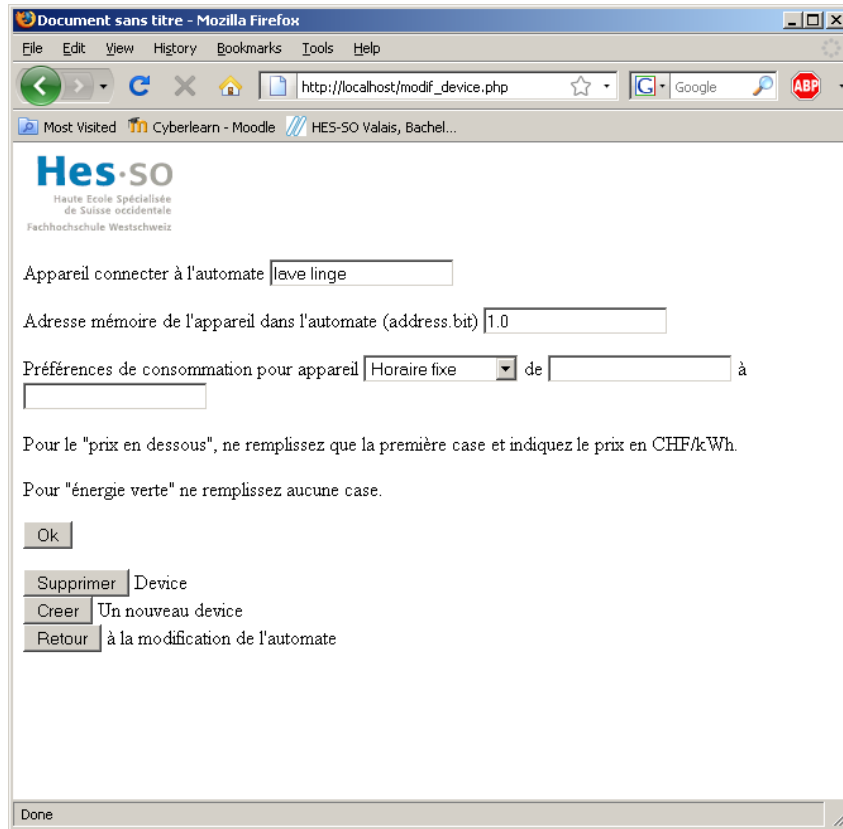
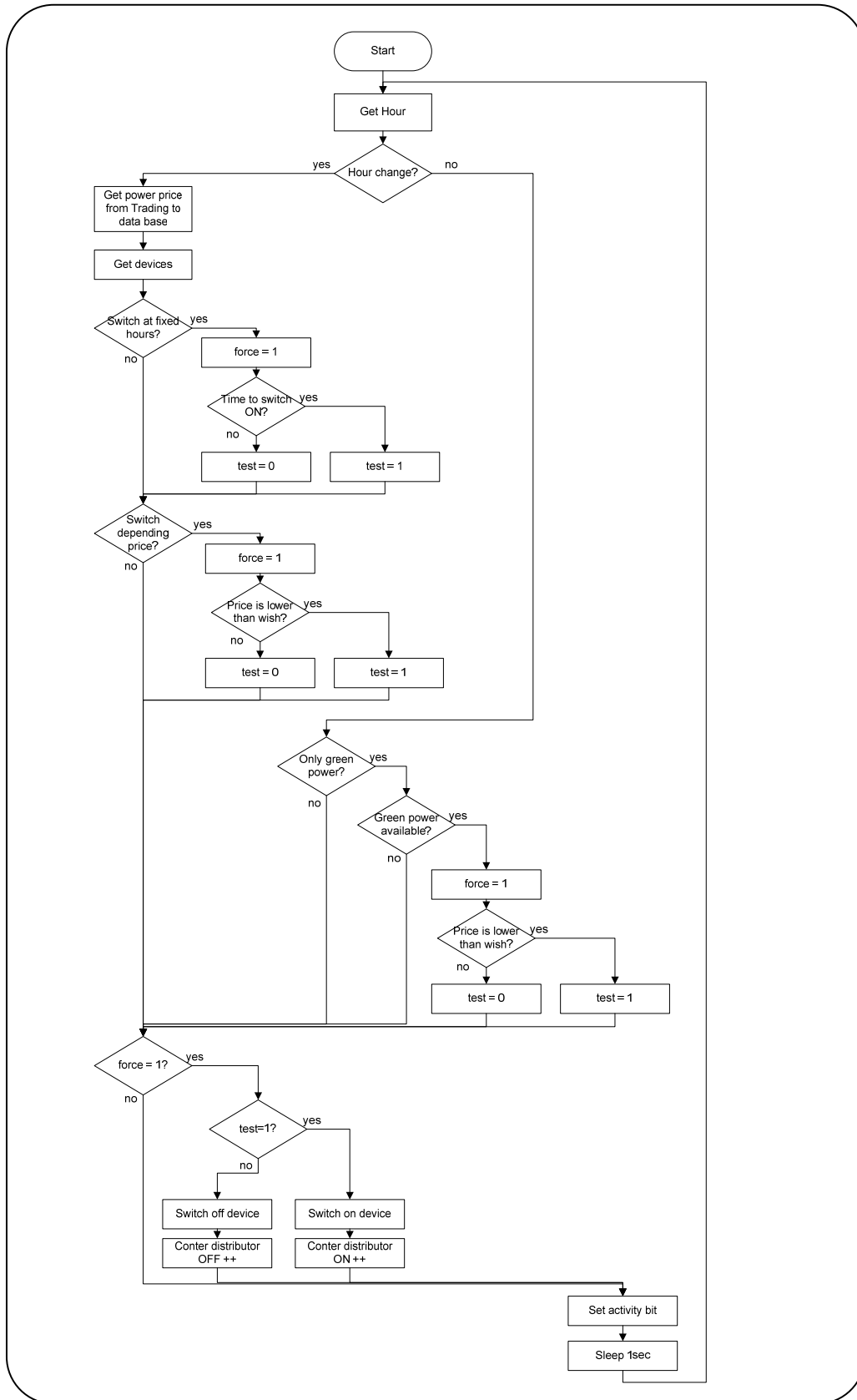


Figure 9.7 : Page de modification de device

9.3 Structogramme intelliswitch_debug



Développement d'un interrupteur « intelligent »

9.4 Contenu du CD

