الجـــــمـــهـــوريـــــــــــة الجـــــــــزائـــريـــة الديمـــــــقـــراطـيـــة
الـشـــــــــــعـبـيـــة

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التـــــــــــعـــليـــــم العـــــــــــــالي

والبـــــــحث العـــــــــلـــمـــــــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb – Blida –

Faculté de Science

MEMOIRE

Présenté pour l'obtention du

Diplôme de **MASTER**

en : **Informatique**

Spécialité : **GSI**

par : **HAMMOUDA Inaam**

Sujet :

Mise en Place d'un Système de Supervision des Machines de Fabrication dans une Unité de Production d'Emballage

Dr Abderrahmane MENACER	Université de Blida 1 Département d'Electronique	Directeur de mémoire
M. Ibrahim KHELIFA	Directeur Technique IECO-Emballage	Encadreur.

Juin 2016

Dédicaces :

A cœur vaillant rien d'impossible

A conscience tranquille tout est accessible

Espérant des lendemains épiques

Un avenir glorieux et magique

Souhaitant que le fruit de nos efforts

Fournis jour et nuit, nous mènera vers le bonheur fleuri.

Je dédie ce mémoire à…

Docteur F. BRAHIMI-DRAI…

Mon symbole de bonté par excellence, exemple du dévouement, de réussite et d'une femme au-delà de la race humaine.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que vous méritez, ce projet n'aurait jamais vu le jour sans vous, je n'aurais jamais continué à exister sans vous.

Je vous dois la vie et la personne que je suis.

Ma Mère Hadjira…

Affable, honorable, aimable : pour tous nos moments d'énervement, de tendresse, de tension, d'écoute et de désaccord, sache que je t'aimerai toujours coûte que coûte, je te dois la vie que j'ai eu …

Mon père Abd EL Fettah…

Aucune dédicace ne saurait exprimer ce que je ressens pour vous, père. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation, le présent travail est le fruit des sacrifices que vous avez, avec générosité, bien voulu consentis pour mon éducation.

Hafidh, Amel et Yasmine…

Mon cher frère qui est, pour moi, aussi bien le père que la mère, aucun mot, aucune phrase ne sont suffisants pour exprimer mon attachement, mon amour et mon affection envers toi.

Mon ange gardiens, mon fidèle compagnon dans les moments délicats de ma vie.

ELLE …

Celle qui ne m'a jamais abandonné, même lorsque je lui tournais le dos tant de fois, je te dédie ce travail …

***Torkia,** pour ton amour inconditionnel.*

***Halima et Mohamed,** merci d'avoir apporté un gout à la vie.*

***Saida,** le chouchou de mon existence, toujours présente dans ma vie, avec sa sérénité et sa joie de vivre.*

***Sarah** et **Leila,** malgré la distance, vous êtes toujours dans mon cœur, je vous souhaite un avenir plein de joie, de bonheur, de réussite et de sérénité.*

***Akli,** "the one and only ", son aide, son humeur, son intelligence, ses encouragements ,merci d'avoir apporté un gout à la fin de ce projet.*

Tous les membres de ma famille, petits et grands recevez, tous, cette modeste mais sincère attention.

Remerciements :

Je remercie Dieu le tout Puissant qui m'a donné la force et la volonté pour réaliser ce modeste travail.

Avant d'entamer la rédaction de ce rapport et à travers ce travail, je tiens à remercier tous ceux qui ont orienté les différentes étapes de ce travail jusqu'à son terme, par leurs estimables conseils et contributions, en particulier :

Monsieur BENNILA Nabil, Professeur d'automatisme.

J'ai eu le privilège de vous rencontrer, de travailler avec vous, et d'apprécier vos qualités, vos valeurs, votre aide et votre disponibilité.

Monsieur MENACER Abderrahmane, Directeur de Mémoire.

D'avoir accepté d'être mon Promoteur durant ce travail, et pour la confiance que vous m'avez donné.
Votre sérieux, votre compétence, votre confiance en moi, vos conseils et vos encouragements m'ont énormément marqué.

Veuillez trouver ici l'expression de ma respectueuse considération et ma profonde admiration pour toutes vos qualités scientifiques et hu-

maines. Ce travail est pour moi l'occasion de vous témoigner ma profonde gratitude.

Monsieur KHELIFA Ibrahim, Directeur Technique d'IECO et Encadreur de ce projet.

Vous avez bien voulu me confier ce travail riche d'intérêt et me guider à chaque étape de sa réalisation.

Vous m'avez toujours réservé le meilleur accueil, malgré vos obligations professionnelles, vos encouragements et votre gentillesse tout ceci mérite l'admiration.

J'en profite de cette occasion pour vous exprimer ma profonde gratitude toute en vous témoignant mon respect.

*Toute l'équipe d'**IECO.** Plus particulièrement **Samra** et **Mohamed** pour avoir facilité mon intégration.*

***Monsieur Hammouda Mohamed,** Professeur de base de données.*

***Les membres du jury,** d'avoir accepté de juger mon travail.*

***Toute ma famille,** qui m'a toujours supporté moralement et financièrement pendant toutes mes longues années d'étude.*

Résumé :

Pour survivre et faire face aux évolutions accélérées d'un marché de plus en plus concurrentiel, les entreprises doivent donc arriver à produire bien, vite et bon marché, tout en étant capables de s'adapter rapidement à l'évolution des produits. Sur le plan industriel, les technologies numériques, les systèmes automatisés et supervisés jouent un rôle primordial dans ce processus d'amélioration de productivité. En effet, la maîtrise de tels systèmes permet, aux entreprises, le pilotage des processus de production pour répondre aux attentes des clients et aux besoins du marché.

Dans le cadre d'une vision stratégique de l'amélioration du fonctionnement de supervision des unités de production, le groupe **IECO** (***I**ndustrie des **E**mballages en **C**arton **O**ndulé*) a opté pour procéder à l'automatisation de la supervision de ses machines de fabrication.

La solution proposée prend en considération non seulement l'intégration des nouvelles technologies ; la faisabilité de la solution ainsi que les contraintes technico-économiques.

Dans ce mémoire, nous avons présenté le contenu du stage qui avait pour sujet la réalisation ***d'un système de supervision des équipements automatisés***. On présentera l'automatisation ainsi que la supervision industrielle d'un point de vue théorique, afin de donner les clés de compréhension nécessaires. Puis tout au long de ce rapport, sont exposées et détaillées les différentes étapes accomplies pour réaliser et parfaire le système projeté. Enfin, on conclura par l'exposé de la solution (matérielle et logiciel) retenue et qui est opérationnelle sur site.

---o0O0o---

Abstract :

To survive and cope with the accelerated evolution of an increasingly competitive market, companies must therefore be able to produce good, fast and cheap while being able to adapt quickly to changing products. On the industrial side, digital technologies, automated systems and supervised ones play a key role in this process of productivity improvement. Indeed, the control of such systems allows companies, management of production processes to meet customer expectations and market needs.

As part of a strategic vision of improving the functioning of supervision of production units, the group **IECO** (Industry Packaging Corrugated) opted to proceed with automating the supervision of its manufacturing machines.

The proposed solution takes into consideration the integration of new technologies, the feasibility of the solution as well as the technical and economic constraints.

In this report, we will present the subject which is ***the realization of supervisory system for automated equipment***. We will introduce automation and industrial supervision from a theoretical point of view to provide the most necessary keys to this subject. And through this report, all the steps to achieve this system will be detailed. Finally, we conclude by stating the results of the proposed solution.

---o0O0o---

Sommaire

Chapitre 2 : Présentation de l'environnement technique

Introduction……

Volet 1 : Généralité sur les Systèmes Automatisés

Volet 2 : la Supervision industrielle

Volet 3 : Protocole de communications FINS

Chapitre 3 : Les outils de développement

Chapitre 4 : Etude conceptuelle

Chapitre 5 : Réalisation

Liste des figures :

Introduction générale.

Dans la société d'aujourd'hui, l'informatique a envahi de nombreux secteurs de la vie. Où que vous soyez, vous êtes, d'une manière ou d'une autre, amené à avoir une certaine relation avec une machine, de plus en plus intelligente. Ces relations vont du respectable à son extrême contraire. Mais une chose est sure, on se voit mal vivre sans cette merveille que nous avons inventée.

Le secteur industriel n'a pas échappé à cette communion scientifique. Au contraire, pendant la dernière décennie, il a connu des évolutions permettant aux industries d'atteindre un haut niveau de croissance, entrainant une complexité des systèmes industriels, qui est dû aussi à la sensibilité et la quantité massive d'informations à gérer pour assurer le bon fonctionnement.

De nos jours, l'implémentation des systèmes automatisés suppose donc la mise en place d'outils pour la supervision. Une chaine de production automatisée est indissociable de sa supervision pour aider les entreprises dans leurs recherches permanentes d'une meilleure productivité et qualité. Elle permet aussi, aux unités de production de garantir, préserver et gérer la sureté de fonctionnement de la production, des équipements, et leur personnel.

Notre Projet de Fin d'Etudes (PFE) (Mémoire de Master) s'inscrit dans ce contexte. Il a été réalisé au sein de l'entreprise IECO Emballage, qui a acquis une unité de production de carton ondulé, complétement automatisée, mais sans système de supervision intégré. Notre travail consiste, donc, à développer une solution SCADA (Supervision, Commande et acquisition de données) capable de collecter, mémoriser et mettre à disposition sous forme conviviale, toutes les informations souhaitées.

Ce mémoire de PFE est composé de cinq chapitres et d'une conclusion auxquels s'ajoute la bibliographie.

Le premier chapitre présente la problématique et le tour d'horizon du cadre du projet ainsi que l'environnement du stage.

Le deuxième chapitre est décomposé en trois volets. Le premier volet est consacré à l'introduction aux systèmes automatisés. Le deuxième volet est consacré aux systèmes de supervision et d'acquissions des données. Le dernier volet présente les protocoles de communication.

Le troisième chapitre est entièrement consacré aux méthodes de modélisation ainsi les outils de développement utilisés afin de réaliser ce PFE.

Le quatrième chapitre étudie la conception du logiciel de supervision développé, en se focalisant sur les différents formalismes et les méthodes proposés pour l'analyse et la modélisation de ce système de supervision et d'acquisition de données.

Finalement, dans le dernier chapitre, nous présentons le résultat de ce projet, la réalisation de ce logiciel ainsi que des captures-écran de l'application.

Chapitre 1 : Contexte de réalisation du projet.

1. Introduction :

L'étude du contexte générale du projet est une étape primordiale et décisive pour la conception de tout système, ce chapitre contiendra un aperçu sur l'organisme IECO, et par la suite le procédé de l'emballage du carton ondulé à la fin une présentation de la problématique ainsi que les objectifs désirés de ce projet.

2. Présentation de l'environnement du stage :

La **SARL I.E.C.O** est une société spécialisée dans la conception et la réalisation d'emballage en carton assurant une recherche permanente sur les matériaux et produits nouveaux ainsi que la création de tous type d'emballage. Sa mission principale est d'apporter à sa clientèle un produit fini, résistant à toutes les contraintes et avec une extrême rapidité d'exécution, réalisé par un personnel hautement qualifié munis d'une longue expérience et sachant manier un équipement moderne et un matériel adapté à toutes les exigences.

2.1 Historique :

La société des emballages **I.E.C.O.** ***(Industrie des Emballages en Carton Ondulé)*** créée en 1996, est le fruit d'une expérience de trois décennies dans le domaine du cartonnage. La société des emballages S.I.F.E.C, créée dans les années soixante-dix, avait pour vocation première la fabrication des emballages en carton ondulé, vierges ou imprimés, à partir de plaques de carton qu'elle achetait auprès d'onduleurs. Au début des années quatre-vingt, alors que l'entreprise et la ville de Blida grandissaient et s'épanouissaient, le siège de l'usine a été transféré au lieu-dit, Berge de l'Oued Sidi El-Kebir, ce qui a permis une extension de l'unité et l'acquisition d'un train onduleur, équipement qui mit fin à notre dépendance vis à vis de nos fournisseurs de plaques ; une ère nouvelle s'ouvrit alors devant la société. Devant l'impossibilité d'envisager une nouvelle extension de la société **S.I.F.E.C**., en raison d'un manque d'espace, les dirigeants ont décidé de créer en 1996 une seconde entreprise, toujours dans le même créneau ; il s'agit de la société **I.E.C.O,** localisée – zone industrielle – site 2 – Ouled-Yaich / Blida. La société qui s'étend sur une superficie de 20 000 m² (entrepôts compris), dispose d'équipements de production modernes lui permettant de développer une large gamme d'emballages avec impression de haute qualité.

Figure 1 : Localisation de IECO.

2.2 Moyens acquis :

- Trains onduleurs : Pour la fabrication du carton ondulé en plaque
- Machines de transformations :
- Auto platines.
- Platines de découpe
- Imprimeuses
- Case maker dotés du groupes imprimeurs 4 couleurs et de groupes découpeurs rotatifs
- Slotters imprimeurs
- Plieuses colleuses
- Piqueuses
- Colleuses
- Ficeleuses

L'unité est équipée d'un dispositif d'aspiration de déchets, qui sont déchiquetés et mis en ballots avant d'être expédiés vers les entreprises spécialisées dans la récupération et le recyclage de carton et papier ou exportés.

Employant à la date d'entrée en production 73 ouvriers, les effectifs de la société **I.E.C.O.,** ont significativement évolué jusqu'à atteindre le nombre de 427 (toutes catégories confondues).Cette montée en cadence s'explique par la politique de l'entreprise qui, pour palier l'obsolescence technologique et l'usure temporelle des matériels et équipements existants, a décidé de lancer dans les années 2012,2013 et 2014 un ambitieux programme d'investisse-

ment, parallèlement à la recherche continuelle d'une efficacité économique et compétitivité par la réduction des coûts de production et une meilleure capacité organisationnelle.

L'entreprise IECO dispose d'un service commercial qualifié qui, en collaboration avec un service d'infographie veillent à satisfaire tous les besoins et attentes, notamment :

- La réalisation et la conception des maquettes.
- La réalisation de la fonction communication de votre produit.
- Assurer le bon cheminement de la relation commerciale entre l'entreprise et sa clientèle.

Avec ses deux unités de production complémentaires. Elle est en mesure de satisfaire pleinement les exigences de modernité et de performance.

3. Procédé de production du carton ondulé :

3.1 Définition d'emballage :

Vient de l'ancien allemand balla, l'emballage est défini comme tout objet constitué de matériaux de toute nature, destiné à contenir et à protéger des marchandises données allant des matières premières aux produits finis, à permettre leur manutention et leur acheminement du producteur au consommateur ou à l'utilisateur, et à assurer leur présentation. D'une façon plus globale, l'emballage d'un produit peut se définir comme : « dans le produit, tout ce qui n'est pas le produit lui-même ».

3.1.1 Familles d'emballage :

Les familles d'emballage sont identifiées par les mots-clés suivants : feuille, sac, caisse, boîte, fût, bidon, bouteille, tube. Chaque famille se décompose en genres d'emballage déterminés en fonction de la forme ou de la conception de l'emballage. Les familles sont définies comme suit :

a) Feuille : ensemble de feuilles de matériau souple employées pour l'enveloppage, après, par exemple, découpage à partir d'une bobine (feuille en plaque, en rouleau)

b) Sac : ensemble de moyens d'emballage préformés à partir d'un matériau souple, généralement quadrangulaires (sac auto-ouvrant)

c) Caisse : ensemble de moyens d'emballage rigides généralement parallélépipédiques. (Caisse américaine, bac, plateau, cagette) ;

d) Boîte : ensemble de moyens rigides, de petite contenance, de formes variées, ne comportant ni col, ni goulot (boîte pliante, boîte deux ou trois pièces, boîte cylindrique composite, boîtier, pot).

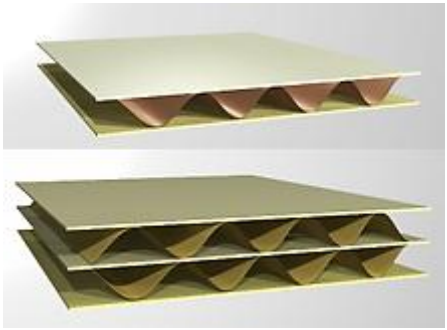
e) Fût : ensemble de moyens d'emballage rigides constitués d'un corps généralement cylindrique pourvu d'extrémités planes et comportant une ou plusieurs ouvertures.

f) Bidon : ensemble de moyens d'emballage rigides comportant un dessus plat ou galbé muni d'un goulot, sans raccordement par un col. (Bidon cylindrique, Jerrican).

g) Bouteille : ensemble de moyens d'emballage rigides se terminant par un col et une bague destinée à recevoir un système de fermeture. (Bouteille champenoise, bouteille à poignée intégrée).

h) Tube : ensemble de moyens d'emballage semi-rigides de section généralement circulaire, dont une extrémité est prévue pour permettre le prélèvement du produit par compression.

3.2 Le carton ondulé :



Le carton ondulé est constitué par un ou plusieurs feuilles de papier canneté collées sur une ou plusieurs feuillent de papier cannelé collées sur une ou plusieurs feuilles de papier plan, utilisé la première fois en 1871 aux Etats-Unis. La fabrication du carton ondulé est assurée par une onduleuse, il faut aussi, du papier, de la colle et de la chaleur.

3.2.1 Structure :

L'étude de la résistance des matériaux a permis, dans le cas des matériaux de construction, de remplacer de lourdes poutres massives par des structures profilées aussi rigides mais plus légères. De la même façon, le carton ondulé permet de remplacer un lourd carton massif par plusieurs feuilles planes maintenues équidistantes par une ou plusieurs entretoises de forme ondulée :

- Les feuilles planes externes sont appelées couvertures ;
- Les feuilles planes internes sont appelées médianes ;

- Les feuilles ondulées formant entretoise sont appelées cannelures.

La simple face est constituée d'une couverture et d'une cannelure solidarisée par des joints de colle sur les crêtes de cannelure en contact avec la couverture ; il est utilisé comme papier d'emballage renforcé, calage, support pour panneaux divers et se stocke en rouleaux.

3.3 Emballage en cartons ondulés :

3.3.1 Fabrication :

Les trois grandes familles d'articles sont :

- Les caisses à rabats.
- Les découpes
- Le petit façonnage.

Les opérations d'impression et d'assemblage sont communes aux deux premières familles. Dans tous les cas, la matière première est la plaque de carton ondulé. Les dimensions des plaques de carton sont toujours données dans l'ordre : laize (parallèle aux cannelures) et longueur de coupe (perpendiculaire aux cannelures).

Les dimensions des emballages sont toujours données dans l'ordre suivant : ***L* × *B* × *H*,** elles sont indiquées dans chaque description du boitier avec :

- *L (longueurs)* = dimension la plus longue à l'ouverture.
- *B (largeurs)* = dimension la plus courte à l'ouverture.
- *H (hauteur)* =la dimension du niveau de l'ouverture jusqu'à la base.

Figure 2 : Caisse à rabats

4. Cadre général du projet :

L'intérêt de la question c'est de passer d'une supervision manuelle à un système informatisé.

4.1 Présentation générale du projet :

Ce projet de fin d'étude est dans le cadre de la supervision industrielle, il a pour finalité de mettre en place un système de supervision des machines de fabrication dans une unité de production d'emballage.

4.2 Problématique :

IECO est une usine spécialisée dans la production du carton ondulé, plus précisément, dans les caisses et les plaquettes de carton, divisée en trois ateliers responsables de la fabrication, chaque atelier à son tour est composé de cellules de production qui regroupent des machines fortement interactives.

Les deux ateliers concernés par ce projet renferment trois machines. Les deux premières du premier atelier : la **618** s'occupe de la production des caisses américaine, l'autre **2200** des barquettes, la troisième dans l'autre atelier la **4500** s'occupe d'emballages des électroménagers.

Il faut savoir que cette unité lors de son acquisition d'**Italie** était dépourvue de tout système de supervision obligeant un traitement manuel des données, par exemple :

- La sauvegarde manuelle des arrêts de la machine.
- Le suivi des états de la machine.
- L'absence de l'historique de la production.

Tous ces problèmes pouvaient être réglés par l'achat onéreux des systèmes de supervision ne pouvant être modifiés donc inexploitable pratiquement.

Au paravent, la saisie des commandes se faisait :

Par la réception d'un bon de commande par le commercial qui le transmet au chef d'équipe pour une deuxième validation et transmis à l'opérateur en tant que bon de production et les saisies par la machine pour l'exploitation.

4.3 Objectifs :

Ce logiciel a pour objectif d'offrir la possibilité d'accéder en temps réel aux données contenus dans les machines de productions, pour assurer une supervision à distance de l'état des machines :

- ***La vitesse.***
- ***La quantité produite.***
- ***L'état de la machine (ON /OFF),***
- ***La possibilité de garder la traçabilité.***
- ***Visualiser l'historique de production***

Tout cela pour analyser et améliorer le processus de fabrication de carton.

4.4 Besoins fonctionnels :

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée/sortie du système. Le logiciel à concevoir doit permettre à l'utilisateur d'effectuer les opérations suivantes :

- **Traitement des commandes** : c'est la fonctionnalité sur laquelle le logiciel tourne, il s'agit de l'outil permettant de collecter et superviser les états de la machine en plein production.
- **Gestion des clients** : cette fonctionnalité permet de gérer les clients, en ajoutant, modifiant et supprimant ces derniers.
- **Gestions des produits** : il s'agit de l'outil permettant de gérer les produits, en ajoutant, supprimant et modifiant un produit
- **Gestion des commandes** : il s'agit de l'outil permettant de gérer les commandes, en ajoutant, supprimant et modifiant une commande.
- **Gestion des comptes** : un outil permettant des opérations telles que l'ajout et la suppression des utilisateurs.

- **Gestion des arrêts** : un outil qui permet la sauvegarde des arrêts des machines de production.

4.5 Les besoins non fonctionnels :

Les besoins non fonctionnels concernent les contraintes à prendre en considération pour mettre en place une solution adéquate aux attentes des utilisateurs, Cette application doit nécessairement assurer ces besoins :

- **L'extensibilité :** dans le cadre de ce travail. L'application devra être extensible. C'est à dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.
- **La sécurité :** l'application devra être hautement sécurisée. Les informations ne devront pas être accessibles à tout le monde. C'est à dire que ce logiciel est accessible par un identifiant et un mot de passe attribué à une personne physique.
- **La performance :** l'application devra être performante i.e. le logiciel doit réagir dans un délai précis, en d'autre termes la consolidation des données et l'obtention des calculs doit être fournit dans un espace de temps juste.
- **La convivialité :** ce logiciel doit être simple et facile à manipuler même par des non experts.
- **L'ergonomie :** le thème adopté par l'application doit être inspiré des couleurs et du logo type de l'entreprise d'accueil.

5. Diagramme de cas d'utilisation globale :[1]

Ce diagramme illustre toutes les fonctionnalités assurées par ce logiciel, ce diagramme sera bien détaillé dans le quatrième chapitre (Etude conceptuelle)

[1] Le chapitre 3 est dédié à la présentation des diagrammes.

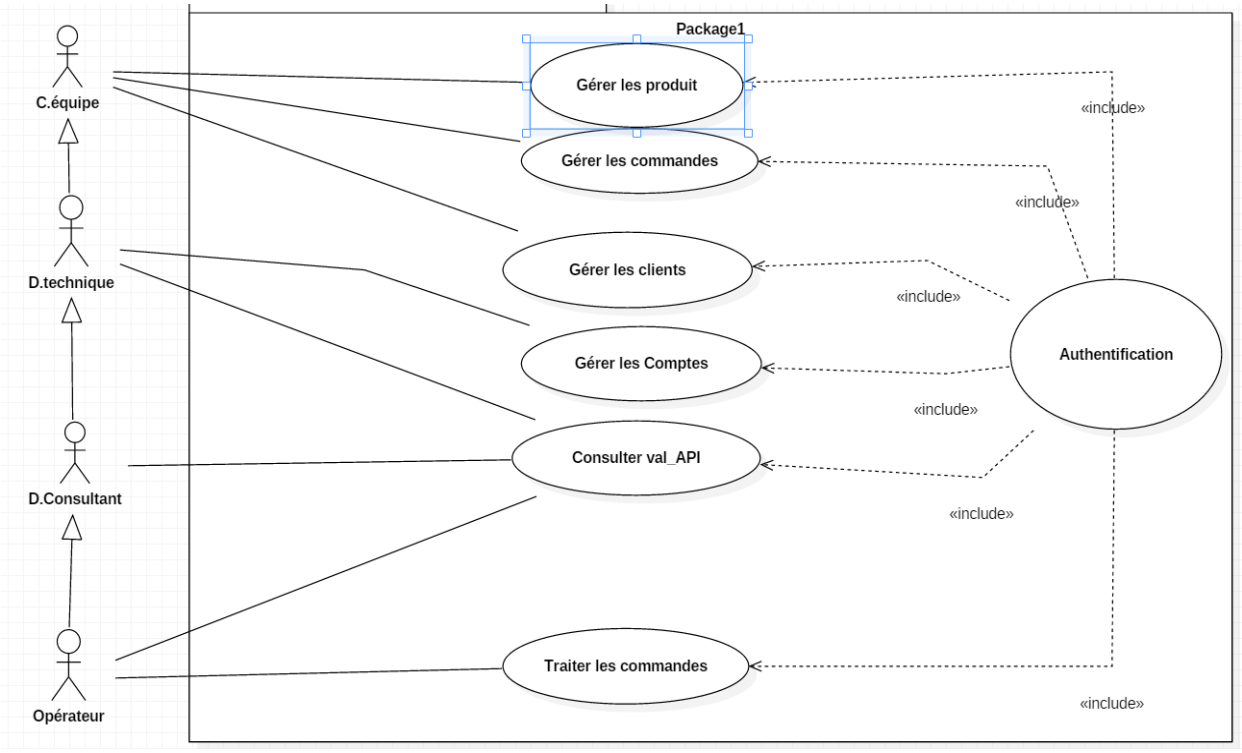


Figure 3 : Diagramme de cas d'utilisation globale.

6. Conclusion :

Ce chapitre nous a permet de décrire le contexte général dans lequel s'inscrit ce projet de fin d'étude, aussi il a bien mis sous les projecteurs l'une des étapes les plus indispensables dans la réalisation d'un projet c'est bien 'analyse et la détermination des besoins fonctionnel et non fonctionnel illustrés dans le diagramme de cas d'utilisation générale.

Chapitre 2 : Présentation de l'environnement technique.

Introduction :

Cette partie présente le travail de recherche mené en vue d'acquérir la connaissance nécessaire à la compréhension des aspects importants du projet : **Systèmes de supervision industrielle** qui seront détaillés dans le reste du mémoire, ce chapitre est devisé en volets. Dans un premier temps nous présenterons les **systèmes automatisés** dans un cadre général, puis nous nous intéresserons aux systèmes **d'acquisition des données**. Nous élargirons ensuite nos recherches aux protocoles de communication **API-PC** (**PLC-PC)** qui seront l'un des constituants importants de ce projet. Nous terminerons par une synthèse du travail effectué

Volet 1 : Généralité sur les Systèmes Automatisés

1. Introduction :

De nos jours, les constructeurs de commande et les ingénieurs automaticiens n'ignorent plus rien des automates programmables, ce point d'intersection à partir duquel ces systèmes de commande relativement récents sont d'un prix comparable ou même inférieur à celui des commandes traditionnelles à logique câblée recule cependant constamment. Les **Automates Programmables Industriels (API)** sont apparus aux Etats-Unis vers 1969, où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués. Le but de cet partie est l'étude théorique des systèmes automatisés précisément l'automate programmable industriel.

Dans ce volet, nous décrirons **les systèmes automatisés** en général, l'histoire des systèmes de contrôle, nous présenterons ensuite une étude sur les automates programmables industriels plus une conclusion pour clôturer ce premier volet.

2. Structure d'un système automatisée

Un système est dit automatisé lorsque le processus qui permet de passer d'une situation initiale à la situation finale, se fait sans intervention humaine et que ce comportement est répétitif. Un système automatisé réalise un certain nombre d'actions appelées « **tâches** ».

Il accomplit une suite d'opérations, appelée « **cycle** », depuis un état initial jusqu'à un état final. On peut distinguer deux types de cycle :

a) **Cycle ouvert** : les taches s'enchaînent et sans aucune vérification comme les feux rouges (leur fonctionnement est identique le jour ou la nuit)

b) **Cycle fermé** : les taches ne se déclenchent pas que lorsque c'est nécessaire. Comme le distributeur de boisson.

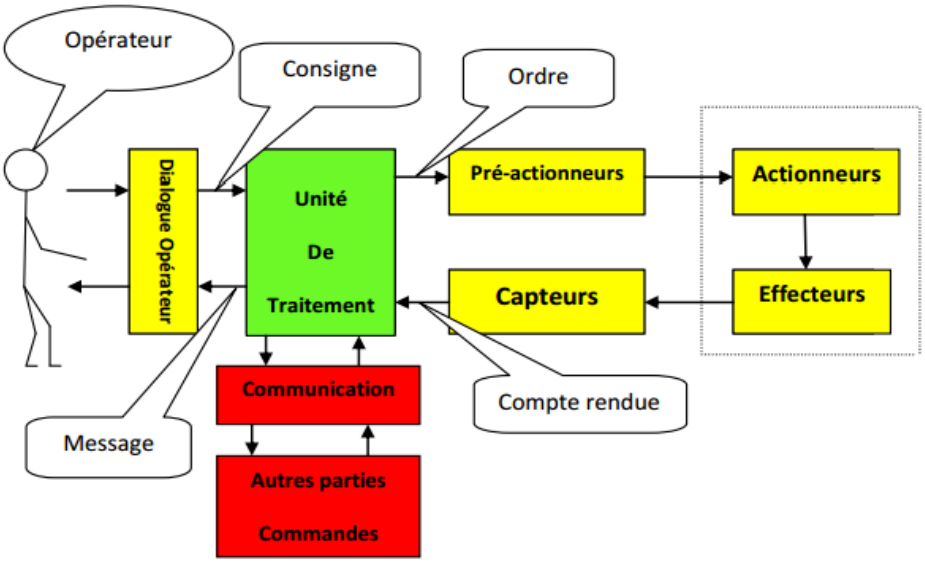


Figure 4 : Structure d'un Système Automatisé

Deux parties importantes sont distinguées :

a) La Partie opérative :

- Elle exécute les ordres qu'elle reçoit de la partie commande grâce aux **ACTIONNEURS.**
- Elle possède des **CAPTEURS** qui permettent de recueillir des informations.
- Elle reçoit des messages et envois des consignes vers la partie commande. Elle comporte les éléments suivants :
- **Pré-actionneur** : est un constituant dont le rôle est de distribuer, sur ordre de la partie commande, l'énergie utile aux actionneurs. Les pré-actionneurs les plus utilisés sont les contacteurs (pour les moteurs électriques) et les distributeurs (pour les vérins pneumatiques).
- **Actionneur** (moteur...) : Objet technique qui transforme l'énergie d'entrée qui lui est appliquée en une énergie de sortie (généralement mécanique) utilisable par un Effecteur pour fournir une action définie.
- **Effecteur** : qui agissent sur la matière d'ouvre (pales de ventilateurs…) (tout organe en contact avec la matière d'œuvre).

- **Capteur** : est un élément de prélèvement et de codage d'informations sur un processus ou sur l'environnement du système. Il convertit une grandeur physique (position, vitesse,) en une information appelée compte-rendu et compréhensible par la Partie Commande.

b) La partie commande :

Elle joue le rôle du cerveau du système, et pilote la partie opérative et reçoit des informations venant des capteurs de la Partie Opérative, et les transmet vers cette même partie opérative en direction des pré-actionneurs et actionneurs. La partie de commande est une unité de traitement ou un automate programmable industriel.

3. Les automates programmables industriels (API) :

A la fin des années soixante, ***GENERAL MOTORS*** a passé un appel d'offre pour la conception d'un système, pour remplacer les armoires à relais et lui permettre de faire des modifications de cycle de fabrication à moindre coût, plus rapidement, et en conservant une possibilité d'évolution. C'est la société ***BEDFORD ASSOCIATES***, et plus particulièrement ***Richard E. MORLEY*** qui, en créant le concept d'automate programmable, emporta le marché.

Richard MORLEY et son équipe, créèrent la société ***MODICON*** (**MOdularDIgitalCONtrol**). Le premier automate fut baptisé ***Modicon 084*** car il concrétisait le 84ème projet de la société, et fut présenté à la fin du 1969 ***Modicon 084*** avec :

- 255 E/S.
- Mémoire : 4 Ko.
- Programmé en : LADDER[2].
- Dimension : L 500 x H 1200 x P 340.
- Poids :46 kg.

3.1 Définition d'un API :

API (Automate Programmable Industriel) ou en anglais PLC (Programmable Logic Controller) c'est un appareil électronique (matériel, logiciel, processus, un ensemble des machines ou un équipement industriel) destiné à la commande de processus industriels par

[2] Ce langage sera présenté dans le point suivant.

un traitement séquentiel (Il contrôle les actionneurs grâce à un programme informatique qui traite les données d'entrée recueillies par des capteurs). Qui comporte une mémoire programmable par un utilisateur automaticien (et non informaticien) à l'aide d'un langage adapté (Le langage List, Le langage Ladder…etc.) pour le stockage interne des instructions donnée pour satisfaire un objectif donné.

L'automate permet de contrôler, coordonner et agir sur l'actionneur comme par exemple un robot, un bras manipulateur, il peut être donc utiliser pour automatiser des processus.

L'API est structurée autour d'une unité de calcul (processeur), de cartes d'entrées-sorties, de bus de communication et de modules d'interface et de commande. Dans notre stage, un automate nommé **OMRON CJ2M** est utilisé.

Figure 5 : L'Automate Industrielle Programmable (API).

3.2 Avantages des API :

- Améliorer les conditions de travail en éliminant les travaux répétitifs.
- Améliorer la productivité en augmentant la production.
- Améliorer la qualité des produits ou en réduisant les coûts de production.

- Programme facile à programmer et à modifier par rapport à la logique câblée
- Simplification du câblage.
- Puissance et rapidité.
- Facilité de maintenance (l'API par lui-même est relativement fiable et peut aider l'homme dans sa recherche de défauts).
- Possibilités de communication avec l'extérieur (ordinateur, autre API)
- Plus économique.

4. Structure d'un Automate Programmable Industriel :

Les caractéristiques principales d'un Automate Programmable Industriel (**API**) sont : **coffret**, **rack**, **baie** ou **cartes.** Il se compose de plusieurs parties et notamment d'une mémoire programmable dans laquelle l'opérateur écrit à l'aide d'un langage dédié aux automates, des directives concernant le déroulement de la fonction à automatiser.

Il comporte quatre parties principales reliées par des bus :

1) **Une mémoire** : conçue pour recevoir, gérer et stocker des informations issues des secteurs du système (le terminal de programmation : PC ou console) et le processeur, ainsi que les informations en provenance des capteurs. Il existe des types de mémoires selon les différentes fonctions des API.

   - **EEPROM** : (Electrically Erasable Programmable Read Only Memory, ROM qui peut être supprimée par un signal électrique et reprogrammée) pour la conception et l'élaboration du programme.

   - **EPROM** : pour la conservation du programme pendant l'exécution.

2) **Un processeur** : a pour but d'organiser les différentes relations entre la zone mémoire et les interfaces d'E/S et gérer les instructions de programme.

3) **Des interfaces d'entrées/sorties** : celle d'entrée comporte des adresses d'entrée, une pour chaque capteur relié, l'interface de sortie contient des adresses de sortie, une pour chaque pré-actionneur. Le nombre d'E/S varie suivant le type d'automate.

4) **Un module d'alimentation** : transforme l'énergie externe provenant du réseau en en la mettant en forme, afin de la fournir aux différents modules de l'API, les niveaux de tension nécessaires à leur bon fonctionnement. Plusieurs niveaux de tension peuvent être utilisés par les circuits internes (3v, 5v, 12v, 24v…). Il sera dimensionné en fonction des consommations des différentes parties. Actuellement, les API utilisent un bloc d'alimentation de 240 V et délivre une tension de 24 V.

La structure interne d'un automate programmable industriel (API) est assez voisine de celle d'un système informatique simple, L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

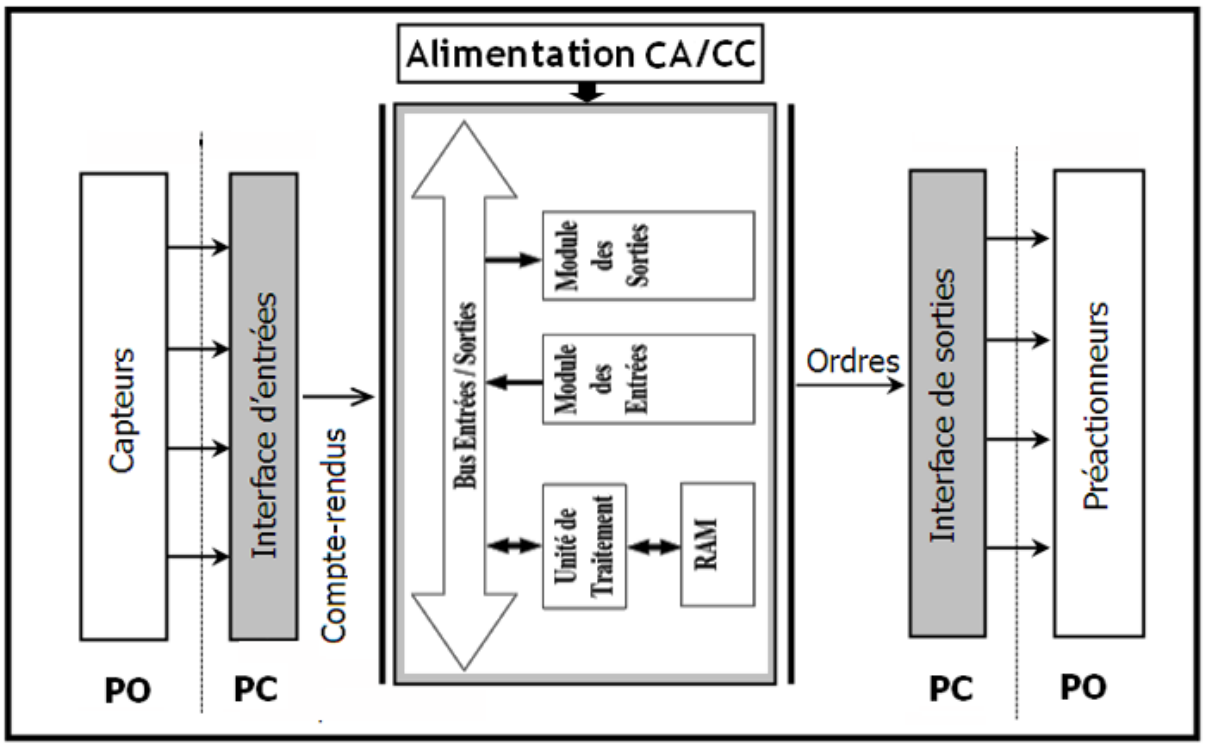


Figure 6 : Structure d'un API.

5. Le fonctionnement d'un automate programmable industriel :

L'API fonctionne par déroulement cyclique du programme. Le cycle comporte trois opérations successives qui se répètent normalement comme suit :

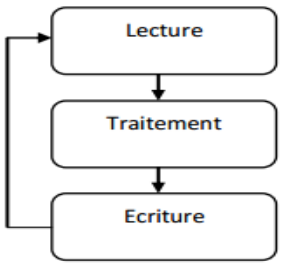


Figure 7 : Fonctionnement d'un API.

- **Phase 1 : Lecture (Photographie des entrées)**

Durant cette phase qui dure quelques microsecondes :

- Les entrées sont photographiées et leurs états logiques sont stockés dans une zone spécifique de la mémoire de donnée.
- Le programme n'est pas scruté.
- Les sorties ne sont pas mises à jour.

- **Phase 2 : Traitement (exécution de programme)**

Durant cette phase qui dure quelques millisecondes :

- Les instructions de programme sont exécutées une à une. Si l'état d'une entrée doit être lu par le programme, c'est la valeur stockée dans la mémoire de données qui est utilisée.
- Le programme Détermine l'état des sorties et stocke ces valeurs dans une zone de la mémoire de données réservée aux sorties.
- Les entrées ne sont pas scrutées.
- Les sorties ne sont pas mises à jour.

Notant que pendant cette phase, seules la mémoire de données et la mémoire programme sont mises à contribution. Si une entrée change d'état sur le module d'entrées, l'API ne voit pas ce changement.

- **Phase 3 : Ecriture (mise à jour des sorties)**

Durant cette phase qui dure quelques microsecondes :

- Les états des sorties mémorisés précédemment dans la mémoire de données sont reportés sur le module de sorties.
- Les entrées ne sont pas scrutées.
- Le programme n'est pas exécuté.

6. Les langages de programmation d'un API :

Les langages destinés à la programmation des automates programmables industriels ont pour objectifs d'être facilement mis en œuvre par tout technicien après une courte formation.

L'écriture d'un programme consiste à créer une liste d'instructions permettant l'exécution des opérations nécessaires au fonctionnement du système. La programmation de ces automates se fait soit à partir de leur propre console, soit à partir du logiciel de programmation propre à la marque. Actuellement les API disposent en tout ou partie des langages de programmation suivants :

a) Langages LITTERAUX :

- **Langage liste d'instructions «IL » (Instruction List)** : très proche du langage assembleur, on travaille au plus près du processeur en utilisant l'unité arithmétique et logique, ses registres et ses accumulateurs. C'est un langage textuel de bas niveau.
- **Langage littéral structuré «ST » (Structured Text)** : Ce langage structuré ressemble au langage C utilisé pour les ordinateurs. C'est un langage textuel de haut niveau. Il permet la programmation de tout type d'algorithme plus ou moins complexe.

b) Langages GRAPHIQUES :

- **Langage à contacts ou diagramme en échelle (LD : Ladder diagram**) : ressemble aux schémas électriques, développé pour les électriciens, ce langage graphique est essentiellement dédié à la programmation d'´équations booléennes (true/false). C'est le plus utilisé. Les automates de l'usine IECO sont programmées à l'aide de ce langage.

- **Le GRAFCET (GRAphe Fonctionnel de Commande par Etapes et Transitions)** : ou SFC (Sequential Fonction Chart) c'est un outil graphique qui décrit les différents comportements de l'évolution d'un automatisme. C'est un mode de représentation et d'analyse d'un automatisme, particulièrement bien adapté aux systèmes à évolution séquentielle, c'est à dire décomposable en étapes.

- **Blocs Fonctionnels (FBD : Function Bloc Dia gram)** : c'est une suite de blocs, liés entre eux, réalisant tout type de fonctions des plus simples au plus sophistiquées, ce langage permet de programmer graphiquement à l'aide de blocs, représentant des variables, des opérateurs ou des fonctions. Les blocs sont programmés ou programmables.

7. Critères pour le choix d'un API :

Le choix d'un API se fait selon la partie commande à programmer. Plusieurs critères sont tenus ont en compte :

- Nombre d'entrées / sorties.
- Le temps de traitement.
- La capacité de la mémoire.
- Le nombre d'étapes ou d'instructions.
- Le nombre de temporisateurs.
- Le langage de programmation.

Dans notre stage, un automate nommé par **Omron CJ2M** est utilisé. Dans ce qui suit, on parlera sur le côté matériel de l'automate **Omron CJ2M.**

8. Sécurité d'un API :

Les systèmes automatisés sont, par nature, source de nombreux dangers (tensions utilisées, déplacements mécaniques, jets de matière sous pression ...). Placé au cœur du système automatisé, l'automate se doit d'être un élément fiable car un dysfonctionnement de celui-ci pourrait avoir de graves répercussions sur la sécurité des personnes, de plus les coûts de réparation et un arrêt de la production peuvent avoir de lourdes conséquences sur le plan financier. Aussi, l'automate fait l'objet de nombreuses dispositions pour assurer la sécurité :

- **Contraintes extérieures** : l'automate est conçu pour supporter les différentes contraintes du monde industriel et a fait l'objet de nombreux tests normalisés.
- **Coupures d'alimentation** : l'automate est conçu pour supporter les coupures d'alimentation et permet, par programme, d'assurer un fonctionnement correct lors de la réalimentation (reprises à froid ou à chaud)
- **Mode RUN/STOP** : Seul un technicien peut mettre en marche ou arrêter un automate et la remise en marche se fait par une procédure d'initialisation (programmée).
- **Contrôles cycliques** : Procédures d'autocontrôle des mémoires, de l'horloges, de la batterie, de la tension d'alimentation et des entrées / sorties
- **Vérification du temps de scrutation :** à chaque cycle appelée ***Watchdog*** (chien de garde), et enclenchement d'une procédure d'alarme en cas de dépassement de celui-ci (réglé par l'utilisateur)
- **Visualisation** : Les automates offrent un écran de visualisation où l'on peut voir l'évolution des entrées / sorties

9. La famille Omron :

Omron est une entreprise japonaise d'électronique basée à ***Kyoto***. Créé par ***Kazuma Tateish*** en 1933. Son secteur d'activité était initialement la vente et fabrication de systèmes d'automatisme. Elle est connue en particulier pour les équipements médicaux comme les thermomètres et les tensiomètres. Omron a développé et mis au point le premier portillon d'accès électronique, ainsi que les premiers distributeurs automatiques de billets de banque avec carte magnétique.

10. Omron CJ2M :

La série **CJ2M** est idéale pour les besoins en automatisation des machines d'emballage ou d'utilisation générale. La connectivité est assurée par un port USB intégré et le choix entre des ports Ethernet et RS-232C/422/485 sur l'UC.

Figure 8 : Omron CJ2M

Toujours accessible via le port USB standard et port Ethernet standard. Une large gamme de capacités de programme de 5K étapes à 60K étapes Serial carte option. La mémoire spéciale du bloc fonction assure l'exécution efficace des modules logiciels de blocs fonctionnels.

11. Conclusion :

D’après ce qui précède, le développement scientifique a laissé sa trace sur les systèmes de production donnant naissance au Système Automatisé de Production, qui s’avère être plus ou moins un remède au paradoxe des paramètres coût-qualité visés généralement par la gestion de production (Optimisation du coût, qualité et délai). Le rôle de l’automatisme industriel est prépondérant puisque les systèmes automatisés occupent et contrôlent l’ensemble des secteurs de l’économie, il a comme objectif d’améliorer la productivité, la qualité, la sécurité et autres variables qui peuvent influencés les objectifs de l’entreprise. L’**API** est un bon équipement s'il est bien choisi et bien employé. Dans le volet qui suit, nous parlerons des systèmes de supervisons.

Volet2 : la supervision industrielle

1. Introduction :

Il est aujourd'hui vital pour un industriel d'exploiter au maximum tous les potentiels d'optimisation, que permet l'ensemble du cycle de vie d'une machine. La supervision est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés, elle concerne l'acquisition de données et des paramètres de commande des processus confiés à des automates programmables.

2. Définition de la supervision industrielle :

La supervision des automates permet le suivi du procédé ainsi qu'une interaction de l'utilisateur sur celui-ci. Il peut en effet rentrer des commandes mais aussi avoir des données en retour. Une supervision est donc composée d'un ou plusieurs ordinateurs en réseau équipés du logiciel adéquat. Mais celle-ci se trouve aussi sur des consoles ou des écrans tactiles, de plus en plus de supervision se font à l'aide d'ordinateurs car leur coût de revient de plus en plus faible font d'eux des outils plus compétitifs et pratiques.

La mise en place d'un système de supervision permet de visualiser en temps réel la bonne marche de l'installation et d'être alerté immédiatement en cas de défaut ou d'alarme.

La supervision offre beaucoup d'avantage :

- Le contrôle de la disponibilité des services et des fonctions.
- Le contrôle de l'utilisation des ressources et la vérification de leurs suffisances.
- Détection, localisation, diagnostic et prévention des défauts et pannes.

3. SCADA :

SCADA, acronyme pour ***Supervisory Control and Data Acquisition*** c'est-à-dire **Supervision *Commande et Acquisition des Données,*** il est utile pour surveiller et commander des équipements ou une installation dans les industries. Il englobe le transfert des données entre le logiciel central et les **API** aussi entre l'ordinateur central et les terminales de commandes. Ils sont l'épine dorsale de l'industrie moderne. Ce sont des systèmes de contrôle de l'automatisation industrielle au cœur de nombreuses industries modernes, y compris :

- Énergie.
- Nourriture et boisson.
- Fabrication.
- Pétrole et gaz.
- Recyclage.

- Transport…et beaucoup plus.

Ces systèmes englobent le transfert de données entre la centrale **SCADA** et un certain nombre d'Automates Programmables Industrielles (**API),** et entre la plateforme SCADA et les autres interfaces de supervisions.

La complexité de ces systèmes varie, elle peut être bien simple, tel que la supervision des conditions environnementales d'un petit immeuble de bureaux, ou complexe, comme un système de supervision de toute activité dans un centrale nucléaire. Aujourd'hui, de nombreux systèmes sont surveillés en utilisant l'infrastructure du réseau local d'entreprise LAN (Réseau local) / WAN (Réseau étendu).

Les systèmes SCADA sont composés de :

- Un ou plusieurs appareils de terrain d'interfaçage de données i.e. les API, dans notre système, il est composé de 3 API.
- Un système de communications pour transférer les données entre les interfaces de données, et entre les API et les interfaces de supervision.
- Une centrale de SCADA (appelé une station maitre SCADA, ou ***Master Terminal Unit*** (MTU)), dans notre système, on aura trois interfaces centrales, chaque interface collectera les données depuis un API.
- Le logiciel de supervision communément appelé logiciel IHM (Interface Homme-Machine).

Les IHM présentent généralement l'information pour le personnel d'exploitation (le directeur technique, le superviseur, le chef d'équipe), à l'aide d'un synoptique, Cela signifie que l'opérateur peut voir une représentation schématique de l'usine contrôlé.

3.1 L'évolution de SCADA :

Dans les années 1950, les premiers mini-ordinateurs ont d'abord été développés et utilisés à des fins industrielles. Dans les années 1960, ce qui était autrefois des mini-ordinateurs ont été maintenant considéré de taille moyenne et ils ont été utilisés pour la surveillance à distance et le contrôle de surveillance.

Le terme « **SCADA** » a été inventé au début des années 1970, et la hausse des microprocesseurs et les contrôleurs logiques programmables (**PLC**) au cours de cette décennie, a donné aux entreprises une plus grande capacité de surveiller et de contrôler les processus automatisés que jamais. Dans les années 1980 et 1990, SCADA évolué encore une

fois avec la large utilisation de réseaux locaux (LAN), qui a permis à des systèmes SCADA d'être connecté à d'autres systèmes, et l'introduction du logiciel IHM sur PC.
Dans les années 1990 et au début des années 2000, ***Structured Query Language*** (**SQL**) des bases de données sont devenus la norme pour les bases de données IT, mais n'a pas été adopté par les développeurs **SCADA**. Cela a abouti à un désaccord entre les domaines de contrôles et de l'informatique, et de la technologie **SCADA** est devenu désuet au fil du temps.

3.2 SCADA modernes :

Les systèmes **SCADA** modernes permettent aux données en temps réel de l'atelier de production d'être accessibles à partir de n'importe où dans le monde. Cet accès à l'information en temps réel permet aux gouvernements, aux entreprises et aux individus de prendre des décisions basées sur les données sur la façon d'améliorer leurs processus. Sans logiciel **SCADA**, il serait extrêmement difficile, voire impossible, de recueillir suffisamment de données pour toujours des décisions éclairées. En outre, la plupart des applications de créateurs **SCADA** modernes ont des capacités de développement rapide d'applications (RAD) qui permettent aux utilisateurs de concevoir des applications relativement facilement, même si elles ne possèdent pas une connaissance approfondie du développement de logiciels.

L'introduction de normes informatiques modernes et pratiques tels que SQL et les applications basées sur le Web dans le logiciel **SCADA** a grandement amélioré l'efficacité, la sécurité, la productivité et la fiabilité des systèmes **SCADA.**
Les logiciel **SCADA** qui utilisent la puissance de bases de données SQL offre d'énormes avantages par rapport aux logiciels **SCADA** archaïque. Un grand avantage de l'utilisation de bases de données **SQL** avec un système **SCADA** est que cela rend plus facile à intégrer dans des systèmes **ERP** (de l'anglais « Enterprise Ressource Planning » c'est-à-dire « Progiciel de Gestion Intégré »), permettant aux données de circuler de façon transparente à travers toute une organisation. Les données historiques d'un système **SCADA** peuvent également être enregistrés dans une base de données SQL, qui permet de faciliter l'analyse des données par le biais de tendances de données.

4. Architecture des systèmes SCADA :

Les systèmes SCADA ont été évolué en parallèle avec la croissance et la sophistication des technologies de l'information. Dans cette partie, l'évolution de ces systèmes sera traitée.

a) Première génération : [monolithique]

Dans cette génération, le concept de l'informatique était appliqué par une unité centrale, les réseaux n'existaient pas, ainsi les système SACADA étaient autonome avec pratiquement aucune connexion à un autre système.

b) Deuxième génération : [distribué]

Cette génération a bien profité des développements dans le domaine de la miniaturisation et de la technologie de réseaux locaux pour répartir le traitement entre plusieurs stations reliées par un réseau local et partager l'information en temps réel.

Chaque station est responsable d'une tache particulière rendant ainsi la taille et le cout de chaque station inférieure à celle utilisée dans la première génération

c) Troisième génération : [en réseau]

C'est la génération actuelle, elle adopte une architecture réseau, qui est étroitement liée à l'architecture distribuée, l'amélioration majeur dans la troisième génération vient de l'utilisation des protocoles WAN comme le protocole IP, pour la communication entre la station maitresse et les équipements de communication

5. Le fonctionnement de SCADA :

SCADA a déployé des éléments matériels et logiciels multiples qui permettent aux entreprises industrielles de : Surveiller, rassembler, et les données de processus Interagir avec les machines et les dispositifs de contrôle tels que les vannes, pompes, moteurs, etc., qui sont reliés par **IHM** (Interface Homme-Machine) (logiciels qui enregistrent des événements dans un fichier journal).

Dans l'architecture **SCADA** de base, destiné pour ce projet, les informations provenant des capteurs ou des entrées manuelles sont envoyés aux automates (les contrôleurs logiques programmables), ces informations sont envoyées au logiciel **SCADA**, il affiche les données et les sauvegarde dans des bases de données, afin d'aider les opérateurs et le personnel d'exploitation d'améliorer l'efficacité et garder la traçabilité de processus de fabrication.

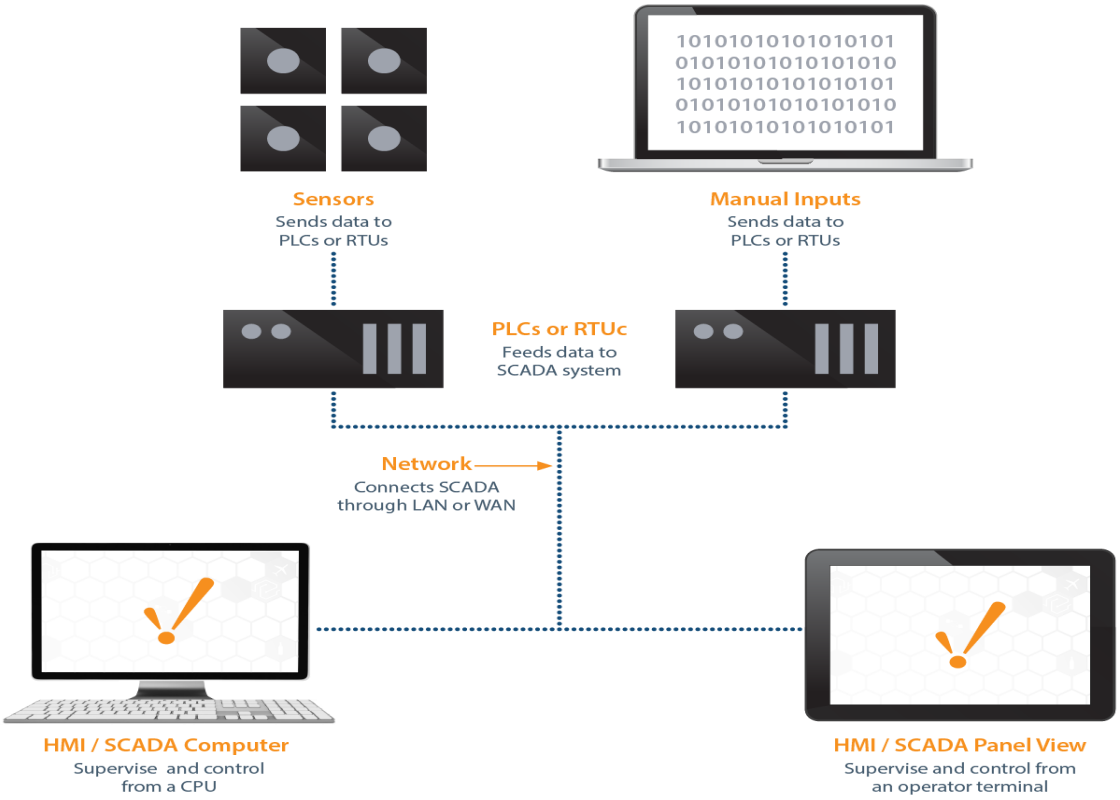


Figure 9 : Diagramme basique de SCADA

6. Conclusion :

L'objectif de ce chapitre est d'illuminer la nécessité de mettre en place des systèmes de supervisons pour réduire les couts économiques et répondre aux contraintes environnementales due à l'évolutions des systèmes automatisés de productions.

SCADA est un outil puissant permettant de mettre en place des plateformes de supervisons et acquisitions capables de prendre en charge la supervision d plusieurs installation en même temps et à des différents endroits grâce au réseau industriel reliant les différentes installations.

Volet 3 : Protocole de communication FINS

1. Introduction :

Le **Networking** est devenu l'un des exigences essentielles des systèmes automatisés parce que l'avantage concurrentiel de demain vient du plancher de l'usine et les données de l'entreprise.

Le Réseau industriel d'**Omron** fournit un outil à mettre en œuvre les connexions des contrôleurs aux **SCADA** qui n'est pas homogène avec l'unité de production.

Omron supporte un ensemble de protocoles de communications de sorte qu'ils peuvent être facilement reliés à un contrôleur tels que le logiciel SCADA, écrans tactiles etc. Il reconnait automatiquement le type de format de commande et envoie la correcte réponse, ceux-ci est réaliser sans aucune intervention de l'utilisateur i.e. sans qu'il soit nécessaire de modifier le code de l'automate.

Ce volet expliquera le protocole choisit afin d'établir une connexion entre les **API** de l'usine **IECO** du type **CJ2M** et les **PC.**

2. Liste de protocoles d'automatisations :

Selon les manuels fournis par **Omron**, il existe un ensemble de protocole assurant la communication avec un **Automate** et un **hôte**, nommés **Process automation protocols** i.e. **protocoles d'automatisation des processus,** selon cette liste, il existe 23 protocoles permettant d'établir une communication **API-PC** (**PLC-PC)**.

Le choix d'un protocole à implémenter dépend des ports existants dans un API. Dans notre cas, les API **CJ2M** de l'usine **IECO** contiennent un port **USB** et deux ports **Ethernet** contrairement à d'autres version de **CJ2M**, nos 3 **API** ne contiennent pas un module d'option de série pour ajouter les ports de communications **Série RS-232** ou **RS-485.**

Figure 10 : Les ports de CJ2M

Durant ce stage, nous avons eu affaire avec seulement deux protocoles, ***ModBus*** et ***FINS***, d'autres détails seront expliqués dans les points suivants.

3. Principes des cartes Ethernet Omron :

Les cartes Ethernet Omron **CS1W-ETN01** et **CV500-ETN01** (appartenant à **CJ2M**) supportent plusieurs protocoles de communication Ethernet : **UDP/IP, TCP/IP, FTP, FINS (protocole réseau Omron)**.

a) Protocole IP : c'est le Protocole de la couche Internet qui se charge de l'adressage logique. IP reçoit les données de la couche accès réseau, vérifie que le datagramme est en bon état et qu'il est arrivé à la bonne adresse en comparant l'adresse IP du message avec celle de la carte réseau de la machine réceptrice. L'un des champs d'en-tête contient l'adresse du protocole (TCP, UDP, ICMP) auquel IP devra livrer le contenu utile des données.

b) Protocole ICMP : Le Protocole Internet de Contrôle de Messages avertit l'expéditeur en cas de non livraison de datagramme, de destinataire introuvable, etc.

c) Protocole ARP : Le Protocole de Résolution d'Adresse est chargé de trouver l'adresse physique correspondant à une adresse logique IP.

d) Protocole TCP : Protocole de contrôle de transmission chargé d'établir une connexion avec le destinataire, d'expédier des segments de données de longueur variable assortis d'accusés de réception, mettre à jour le statut de la transmission puis de clôturer la connexion. La complexité de l'en-tête révèle la richesse des fonctions de TCP.

e) **Protocole UDP :** Protocole Datagramme Utilisateur est un mécanisme de dialogue à l'usage des programmes applicatifs débarrassé de l'en-tête TCP. UDP ne réémet pas les données manquantes ou erronées, n'envoie pas d'accusés réception des datagrammes, n'établit ni n'interrompt les connexions. FINS utilisant l'entête UDP, l'utilisateur doit prendre des dispositions pour s'assurer de la bonne livraison et le cas échéant de la ré-expédition des messages.

f) **Protocole SMTP :** Protocole d'échange entre deux hôtes sur un réseau TCP/IP limité ici à l'envoi de message.

4. ModBus :

C'est un protocole de communication série rappelons qu'une communication série est la modalité de transmission de données dans laquelle les éléments d'information se succèdent, les uns après les autres sur une seule voie entre deux points, publié par ***Modicon*** en 1979 (c'est le premier Automate programmable, crée en 1968, appartenant à Schneider Electric depuis 1996). Aujourd'hui ; il est le moyen le plus couramment utilisé pour communiquer des dispositifs industriels.

Ses avantages se résument dans : Sa gratuité, c'est un protocole Open Source et sa simplicité de mettre en œuvre.

Il fonctionne sur la couche réseau Ethernet **TCP**, chaque dispositif destiné à communiquer en utilisant Modbus est donnée une adresse unique. Maximum de 247 stations peut être connecté sur un réseau Modbus. La longueur maximale du fil est limitée à 1200mtr. Sa configuration est généralement en mode : maître / esclave.

Tout dispositif peut envoyer une commande **ModBus**, bien que généralement un seul dispositif maître fait. Une commande Modbus contient l'adresse Modbus de l'appareil destiné. La plage valide pour l'adresse est de 0 à 247. Une adresse est égale à 0 signifie une diffusion à tous les périphériques du réseau. Seul le dispositif destiné agira sur la commande, même si d'autres périphériques peuvent la recevoir. Toutes les commandes Modbus contiennent des informations de contrôle, pour assurer qu'une commande arrive en bon état

Les commandes de base **ModBus** peuvent charger un **API** pour modifier une valeur dans un de ses registres, ainsi que commander le dispositif de renvoyer une ou plusieurs valeurs contenues dans ses registres.

Comme mentionné dans le point précédent, l'absence des ports de communications série dans les **API** d'**IECO** nous a empêché de mettre en œuvre ce protocole, notre deuxième alternative était le protocole FINS qu'on va présenter prochainement.

5. Le protocole FINS :

FINS (*Factory Interface Network Service*) est le protocole de communication réseau natif de tous les automates omron des séries **alpha/CS1/CJ1/CJ2/CP1L/CP1H.**

FINS permet à un automate ou bien un PC (équipé de FinsGateway ; qui est un programme de communication de middleware qui est fourni par **Omron** pour réaliser un environnement transparent afin d'échanger de informations dans les réseaux), de communiquer avec un autre Automate Programmable industrielle (ou PC) du réseau ou appartenant à un autre réseau, via un réseau intermédiaire si nécessaire. On peut ainsi envoyer une commande **FINS** depuis un réseau Controller-Link, passer ensuite sur un réseau Ethernet pour atteindre un autre réseau Controller-Link. Un réseau peut comporter jusqu'à 32 nœuds.

Ce protocole d'**Omron** peut être utilisé par un programme API (PLC) pour transférer des données et effectuer d'autres services avec un automate distant connecté sur un réseau Ethernet, il peut également être utilisé par des périphériques distants, tels que les PC pour transférer des données entre **API-PC (PLC-PC).** Il utilise les protocoles Ethernet UDP ou TCP pour transporter les messages de Fins depuis et vers l'API.

Avec le service de communication **Fins**, quand il y a plusieurs unités Ethernet connectées au réseau Ethernet, les unité Ethernet sont identifiées par des numéros de nœuds rappelons que, dans un réseau, un nœud est un point de connexion, soit un point de redistribution ou un point pour les transmissions de données de fin, d'une manière générale, un nœud est programmé ou conçu à reconnaitre et traiter les transmissions vers l'avant ou vers d'autres nœuds. Les nœuds sur un réseau Ethernet sont identifiés par des adresses IP, chaque adresses IP est définies avec 32 bits de données binaires, sont divisés en quatre champs de 8 bits.

5.1 Fonctionnement basique :

Les commandes **FINS** peuvent être envoyées ou reçues d'autres automates ou ordinateurs sur le même réseau Ethernet, en exécutant **SEND** ou **RECV**. Ceci permet de lancer les différentes opérations de contrôles telles que l'écriture te la lecture depuis la mémoire.

Exécutées, à partir de l'ordinateur hôte, les commandes **FINS** avec **UDP/IP** ou **TCP/IP**. La fonction **FINS Gateway** permet d'accéder à des automates trouvant sur le même réseau Ethernet.

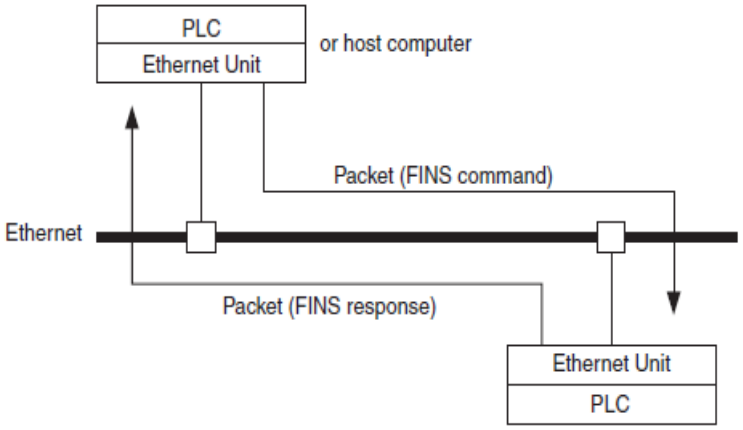


Figure 11 : Fonctionnement basique de FINS.

Pour les communications en utilisant les messages **FINS,** il faut bien spécifier la commande à exécuter et l'adresse de destination afin de pouvoir envoyé les données, ceci est appelé la commande **FINS**, l'unité recevant la commande **FINS** (**FINS Command**), exécute la commande et renvoie les résultats d'exécution en réponse à la source, cela est bien évidement la réponse FINS (**FINS Response**).

Les applications exécutent généralement cette communication appariée d'une commande FINS et une réponse FINS, par exemple, sur des ordinateurs équipés d'un FINSGateway, si le numéro d'unité de la destination est 0, la commande FINS : ***Read Data Area*** (**lire la zone des données**) est utilisée (le code de la commande est **0101**). Par conséquent l'application envoie la commande ***Read Data Area*** à l'unité **0**, cette dernière renvoie les données dans sa zone de données comme la réponse.

Cela permet à l'application de lire les données dans la zone de données de l'unité de destination.

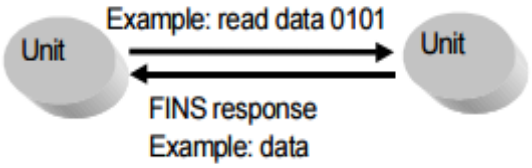


Figure 12 : Commande et réponse FINS

6. FINS TCP /IP :

Les séries de l'API **OMRON** utilisent le protocole **TCP/IP** pour la communication via Ethernet, le protocole utilisé est bien **FINS** (la couche application pour **Omron**), toutes couches produites pour ce protocole peuvent donc être représenté selon le modèle **OSI** suivant :

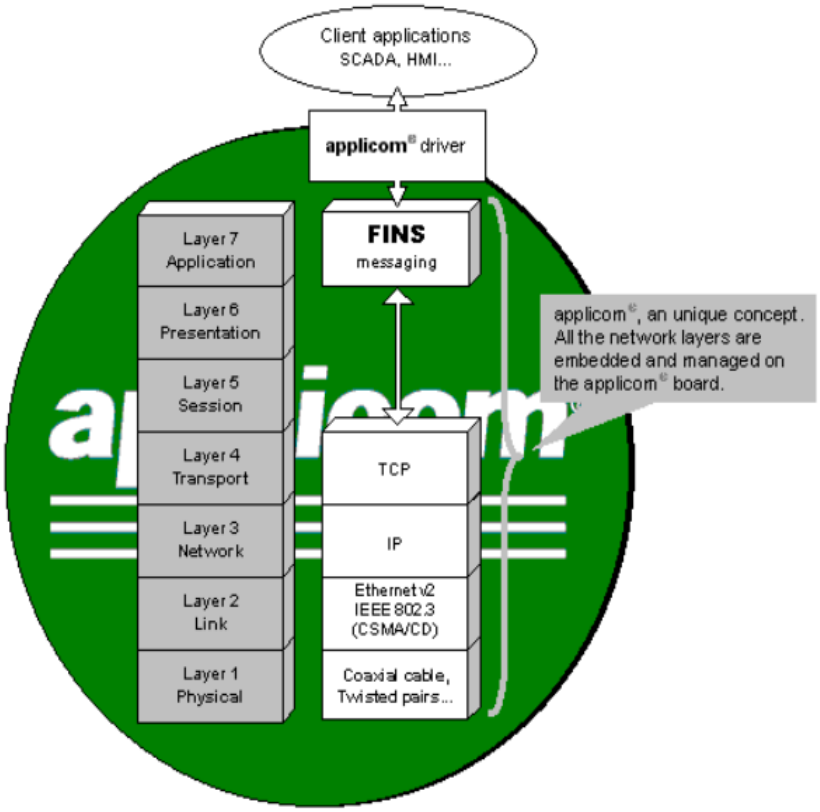


Figure 13 : Les couches d'Omron selon OSI.

La méthode **FINS TCP/IP** est une méthode de communication **FINS** qui utilise le protocole **TCP/IP**. Rappelons que **TCP/IP** est un protocole de communication, avant qu'un message soit envoyé d'un nœud à un autre, il est nécessaire d'établir un circuit virtuel, i.e. une connexion. Une fois qu'une connexion est établie, les communications sont assez fiables.

L'arrivée des données envoyées via la connexion est confirmée par un accusé de réception (ACK), et les tentatives sont exécutées automatiquement au besoin.

Dans le service de communication **FINS**, une adresse **IP** (la couche Internet) et une adresse de nœud **FINS** (la couche d'application) sont utilisés pour le périphérique distant. En outre,9600 est utilisé comme paramètre par défaut pour le numéro de port TCP (i.e. la couche de transport).

La méthode **Fins TCP/IP** a été récemment ajoutée aux unités ***CS1W-ETN21*** et ***CJ1W-ETN21*** Ethernet (rappelons que c'est bien notre api cible). Quand **FINS /TCP** est utilisé, il faut déterminer quel nœud est le serveur et qui est le client, dans notre projet, pour une communication entre un ordinateur et l'Automate Programmable Industrielle **Omron**, l'ordinateur est défini comme un client et l'**API** en tant que serveur. Pour une communication entre deux **API**, soit un peu être défini comme un client et l'autre en tant que serveur.

Avec **FINS/TCP**, la transmission des données est plus fiable, en raison de facteurs tels que le traitement de nouvelle tentative au niveau de la couche **TCP/IP,** cela la rend mieux adapté à traiter les erreurs de communications dans un réseau.

Les adresses des nœuds **FINS** et les adresses IP dans la table interne sont modifiées à chaque connexion établie, par conséquent, même lorsqu'une **commande FINS** a été reçue à partir d'un ordinateur personnel (un client DHCP ordinateur), dont l'adresse IP est modifiée dynamiquement, une réponse peut encore être renvoyé à l'ordinateur à partir de lequel la commande est originaire.

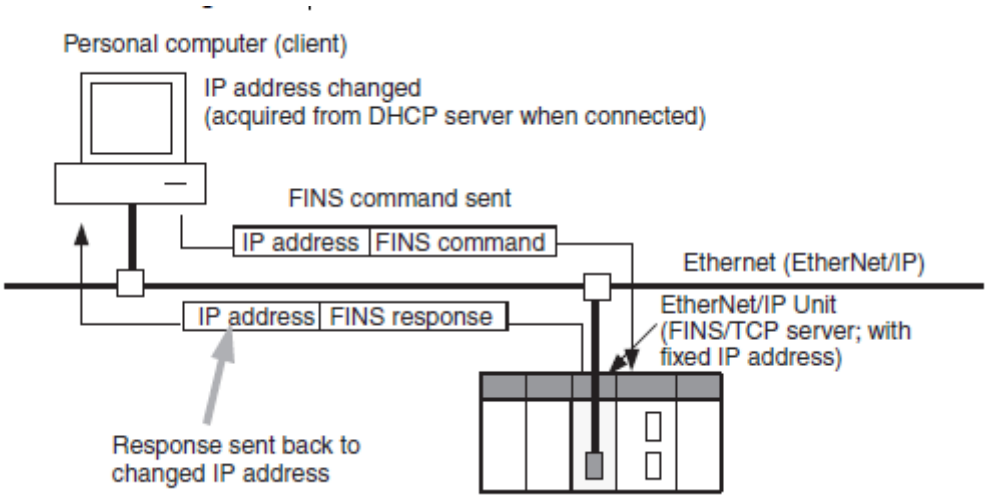


Figure 14 : Le chemin d'une commande FINS.

6.1 La trame FINS /TCP :

Le service de communication **FINS** est effectué par l'échange de trame de commande **FINS** et leurs réponses (il y a des commandes sans réponses). Les trames de commandes et de réponses sont constituées d'un **en-tête** (**FINS Header**). Pour mémoriser des informations de commande de transfert, un champ FINS de commande pour stocker une commande et un champ de **paramètre /données** FINS pour stocker les paramètres de commande et de transmission de **données / réponses.**

Les commandes FINS forment un système de commandes pour les services de messageries à travers les réseaux **Omron**. Elles peuvent être utilisés pour diverses opérations de contrôles, telles que l'envoi et la réception de données, le changement des modes de fonctionnements et ainsi de suite.

Elles ont les caractéristiques suivantes :

- Elles sont définies dans le niveau d'application et ne dépendent pas des niveaux inférieurs (i.e. la couche physique et la liaison de données).
- Elles peuvent être utilisés pour accéder aux différents types d'appareils en plus des CPU Units, des dispositifs tels que les ordinateurs personnels.
- Elles supportent **Network Relay Operating**, de sorte qu'elles peuvent passer à travers une hiérarchie de réseau pour accéder aux périphériques sur un maximum de trois niveaux de réseau.

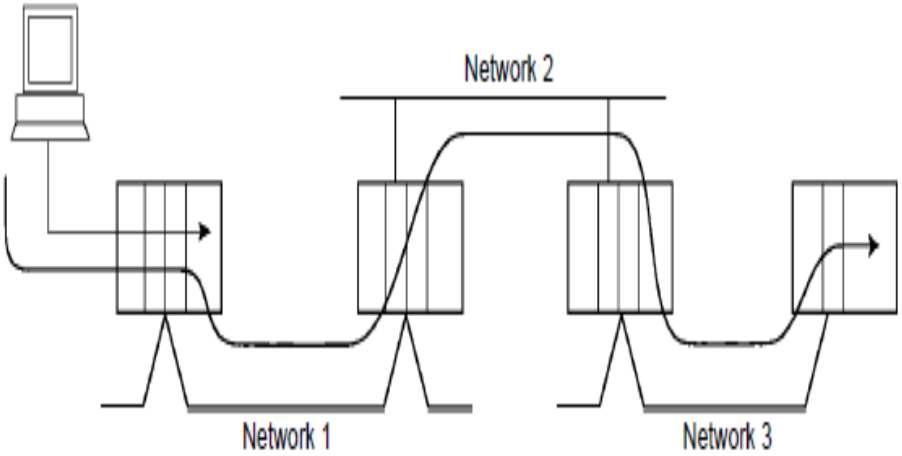


Figure 15 : Hiérarchie d'un réseau supporté par FINS.

6.2 Le format de la commande FINS :

La figure suivante illustre le format de la commande FINS :

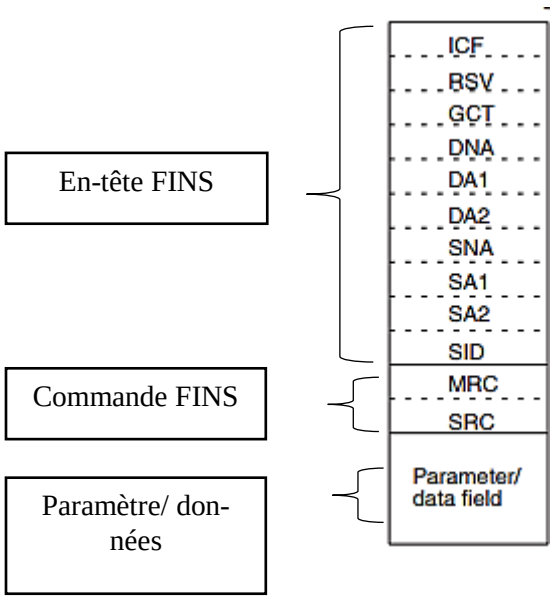
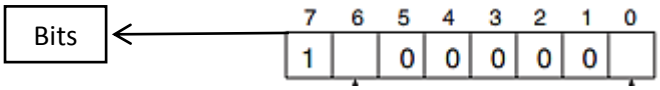


Figure 16 : Format d'une commande FINS.

a) L'en tête de la trame (**FINS Header**) est composé des dix octets suivants :

- **ICF** : (***Information Control Field***), champs de contrôle de l'information, affiche les informations de la trame.



-le bit numéro 0 : (***Response Request Bit***), c'est le bit de requête de réponse, la valeur 0 indique que la réponse est nécessaire, tandis que la valeur 1 indique que la réponse n'est pas nécessaire.

-les bits **1,2, 3,4** et **5** toujours fixés à **0.**

-le bit numéro **6** : (***type of Data***), représente le type de données, la valeur 0 référence une commande et la valeur 1 représente une réponse.

- **RSV** : (***Reserved***), réserver par le système, fixé à 00 (Hexa).
- **GCT** :(***Permissible Number of Gateways***) nombre admissible de la passerelles (Gateway). Sa valeur est égale à 02.
- **DNA** : (***Destination Network Address***), adresse de réseau de destination. Spécifie le numéro du réseau dans lequel le nœud de destination se trouve.

  Si le réseau est local, sa valeur est égale 00, sinon elle appartient à l'intervalle {01 à 7F}.
- **DA1** : (***Destination Node Address***), adresse de nœud de destination, indique le numéro du nœud où la commande est envoyée.
- **DA2** : (***Destination Unit Address***), adresse d'unité de destination, indique le nombre de l'unité au niveau du nœud de destination.
- **SNA** : (***Source Network Adress***), adresse de réseau source, indique le numéro du réseau où le nœud local est situé.
- **SA1** : (***Source Node Address***), adresse de nœud source. Indique l'adresse du nœud local
- **SA2** :(***Source Unit Address),*** adresse d'unité source.

- **SID** : Service ID, utilisé pour identifier le processus qui envoie les données, tout nombre peut être réglé entre 00 et FF (hexa) pour le SID, la même valeur définie pour le SID dans la commande est renvoyée par le nœud envoyant la réponse, ce qui permet aux commandes et réponses à mettre en correspondance lorsque les commandes sont envoyées successivement à la même unité.

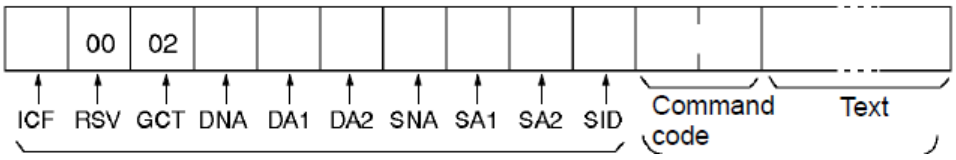
b) La commande FINS est composé de :

- **MRC** :(***Main Request Code***), code de la requête principale, sur 1 octet.
- **SRC** : (***Sub-Request Code***), code de sous réponse, aussi, sur 1 octet.

c) La zone de paramètres/données est composée de :

- Champ paramètre/données : codé sur 2000 octets au max, il contient les paramètres de la commande et les données envoyées, la longueur des données dépend de **MRC** et **SRC**.

Quand une commande de **FINS** est envoyée, son en tête sera automatiquement enlevé et les données de sa réponse seront spécifiées



Exemple : Lecture à partir d'une mémoire E/S :

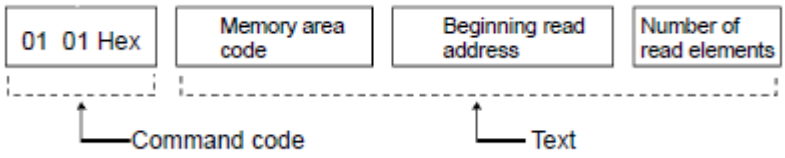
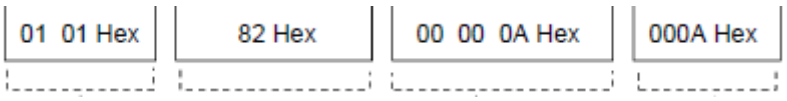


Figure 17 : Exemple de commande FINS

Les données suivantes liraient 10 mots à partir de D00010.



-1- -2- -3- -4-

1- : code de la commande. 2- : code de la zone mémoire.

3- : première adresse à lire. 4 - : numéro d'élément à lire

6.3 Le FORMAT DE LA REPONSE FINS :

La figure suivante illustre le format de la réponse FINS :

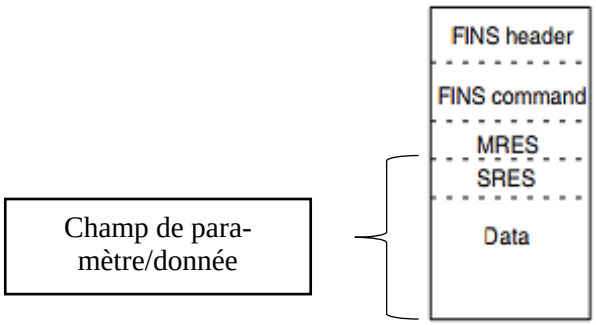


Figure 18 : Format d'une réponse FINS.

a) L'en tête de la réponse (**FINS Header**) est le même que la commande FINS, montré précédemment.

b) La zone de de la commande FINS (**FINS command)** est la même que celle dans la trame de commande, introduite précédemment.

c) Le champ **paramètre/données** composé de :

- **MRES** : (***Main Response Code***) : code de réponse principale, sur 1 octet.
- **SRES** : (***Sub-Response Code***) : code de sous – réponse, codé sur 1 octet.
- **DATA** : les données de réponse, codé sur maximum 1998 octets.

Exemple de la réponse :

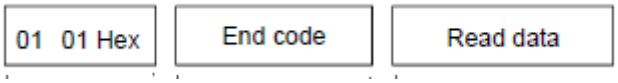
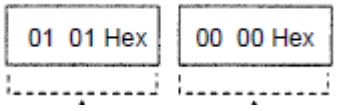


Figure 19 : Exemple d'une réponse Fins

Le End Code i.e. le code final est un code de 2 octet qui montre le résultat de l'exécution de la commande, le premier octet indique la catégorie générale et le second octet indique le résultat en détails.



1 : code de la commande. 2 : code de la réponse.

6.4 Le port TCP pour Fins/TCP :

Le numéro de port TCP est pour identifier la couche d'application (i.e. le service de communication, FINS dans notre cas). Lorsque les communications sont exécutées en utilisant **TCP/IP**, ce numéro de port doit être alloué pour le réglage par défaut de la communications service. Le réglage par défaut de l'unité Ethernet pour le numéro de port **TCP** de **FINS/TCP** est **9600**.

Pour définir un autre numéro, une modification de port est effectuée pour le port FINS/TCP en utilisant l'onglet de configuration dans la configuration dans l'unité, en utilisant le logiciel [3]*CX-Programmer*.

6.3 Nombre de connexion FINS/TCP :

Fins/TCP permet jusqu'à **16** connexions simultanément, ces 16 connexions sont gérées dans l'unité Ethernet, par des numéros de connexion. Lors de la configuration des connexions à l'aide des paramètres FINS/TCP.

7. Procédure de communication FINS/TCP :

Avec FINS/TCP, les adresses de nœud FINS sont échangées immédiatement après une connexion est établie. Cela permet de déterminer les adresses de nœud de FINS à lesquelles les numéros des 16 connexions sont connectées et de les gérer dans une table interne.

Avec les services de communications FINS sur un réseau Ethernet, les adresses IP et les numéros de port TCP sont jumelés avec les adresses des nœuds de FINS pour spécifier les nœuds du réseau.

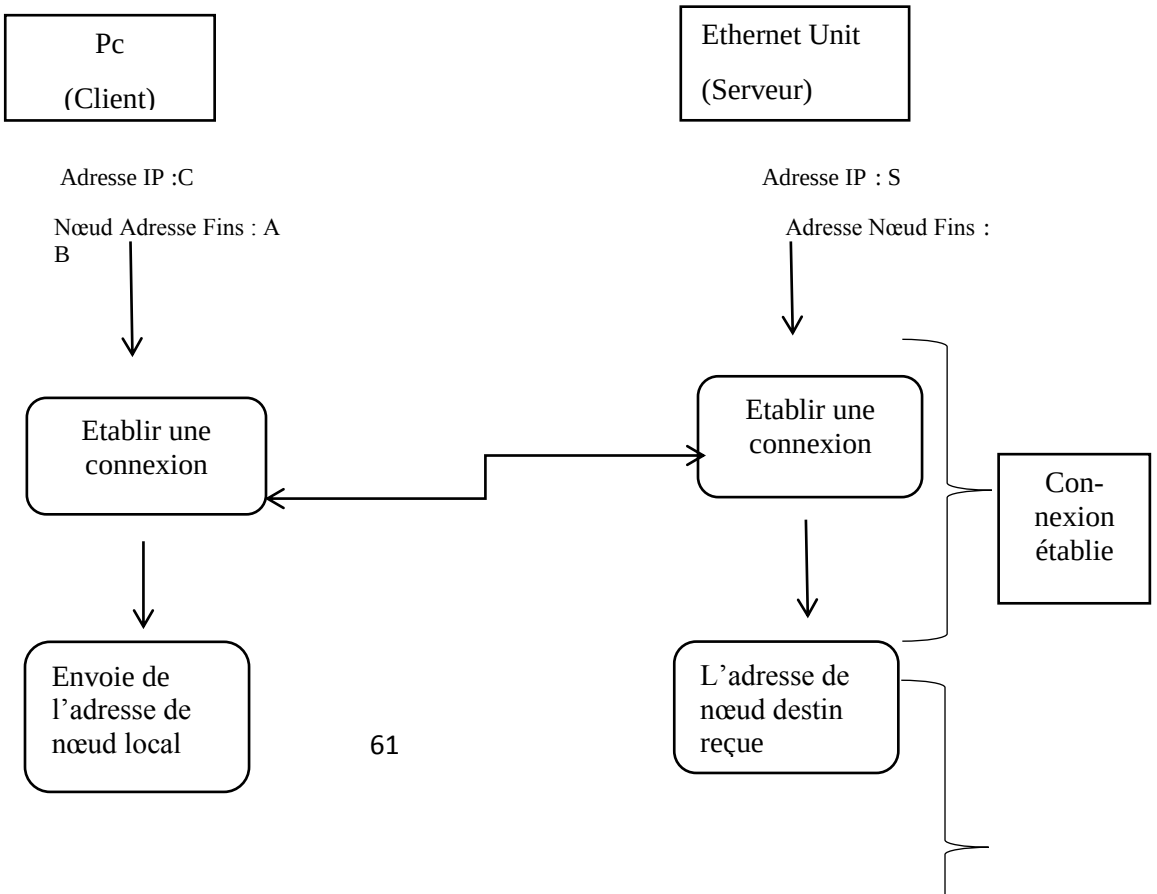
[3] Le logiciel CX-Programmer sera présenté dans le chapitre trois. Et la configuration de FINS/TCP sera illustré dans le dernier chapitre.

- **Adresse Ethernet** : un nombre fixe est attribué à chaque Ethernet /IP unité et il ne peut être modifié.
- **Adresse IP** : Fins utilise l'adresse de l'API (à récupérer en utilisant cx-programmer).

Les nœuds FINS sont attribués individuellement pour chaque application sur ordinateur, et les numéros de port TCP respectifs qui sont utilisés sont également alloués individuellement, chaque application est positionnée avec un client **FINS/TCP** et demande l'ouverture d'une connexion avec le serveur **FINS / TCP**, quand la connexion est établie, l'adresse IP distante et le numéro de port distant dans la table interne sont modifiés dynamiquement.

Après une connexion a été établie, elle est interrompue dans les cas suivants :

- Lorsque la connexion est fermée par le client.
- Quand une commande FINS pour fermer la connexion (le nœud à distance change la requête : le code 27 30 de commande en hexadécimal) est envoyé par le client.
- Quand il n'y a pas de réponse de l'api.
- Si une autre commande autre que la trame de commande FINS est reçu du client, la connexion sera fermée après la trame de commande FINS envoie une notification d'erreur est envoyée.



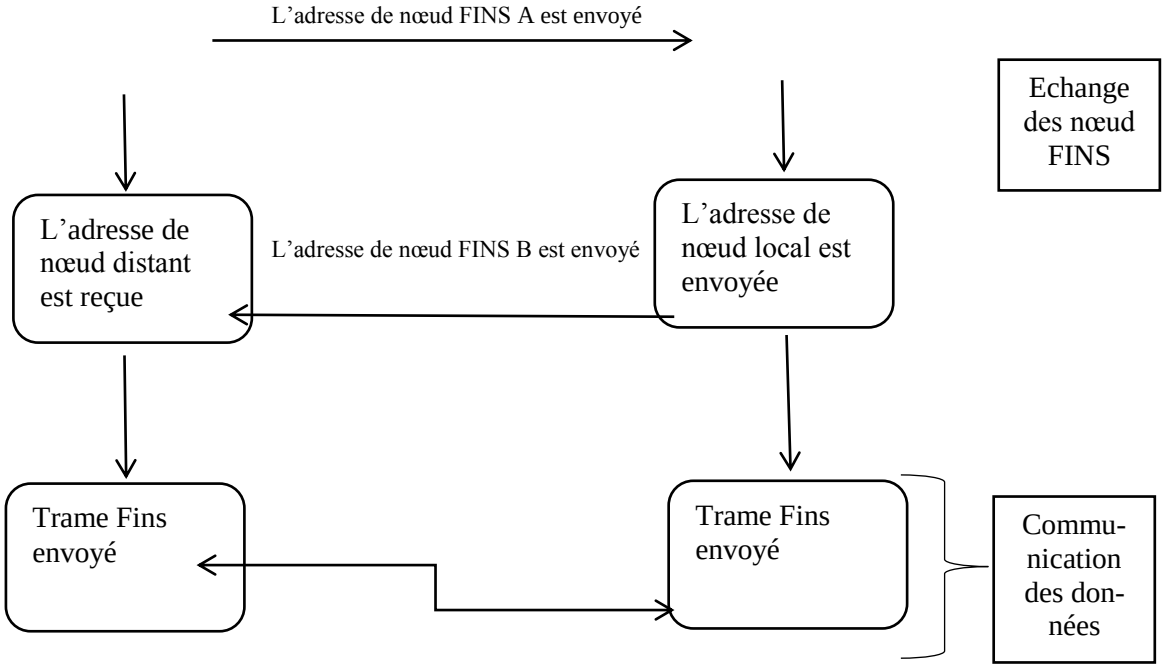


Figure 20 : Schéma résumant la communication FINS.

8. Conclusion :

Ce Volet nous a permis d'introduire le protocole de communication Fins avec qui on a pu établir une communication API **– PC** (**PLC-PC**), le cœur de ce projet SCADA.

Sur ceci, on conclut ce deuxième chapitre, il nous a donné un aperçu sur l'aspect automatisme de ce projet en présentant d'une manière simple et brève les systèmes automatisés, les systèmes de supervisons et les protocoles de communications.

Le chapitre suivant concentrera sur la présentation des méthodes de conception et les logiciels utilisés afin d'évité les présenter au fils des chapitres suivants.

Chapitre 3 : les outils de développements.

1. Introduction :

Une fois l'étude de l'existant est terminée, il s'agit maintenant de chercher quels outils logiciels adéquats utiliser et quelles méthodologies suivre. Les outils logiciels se divisent en trois types :

- le système d'exploitation sur lequel le projet a été effectuer.

- l'outil de développement de l'interface utilisateur et de tous les contrôles qui suivent.

- le SGBD qui va permettre la création et la gestion de la base de données.

Quant aux méthodologies, deux seront adaptées, une méthode de conception et une méthode de développement.

Le présent chapitre a pour but de présenter ces outils et la méthode de travail utilisée pour la réalisation de ce projet.

2. Les méthodes de conception :

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse.

- **UML** :

UML (« Unified Modeling Language » ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. Il permet donc de modéliser les objets et ainsi représenter l'application sous forme de diagramme.

A) Diagramme de Cas d'utilisation (Use Case) :

Le diagramme de cas d'utilisation décrit les utilisations requises d'un système, ou ce qu'un système est supposé faire. Les principaux concepts de ce diagramme sont les acteurs,

cas d'utilisation et sujets. Un sujet représente un système avec lequel les acteurs et autres sujets interagissent.

Use Cases décrivent les services les plus importants rendus par un système. Partant des acteurs, participants externes qui interagissent avec le système, ils représentent les cas les plus importants du système en cours d'utilisation.

B) Diagramme d'activité :

Un diagramme d'activité permet de mettre l'accent sur les traitements, il est donc particulièrement adapté à la modélisation du cheminement de flots de contrôle et de flots de données. Il permet ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

C) Diagramme de classe :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classes montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il s'agit d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation. Une classe-association possède les caractéristiques des associations et des classes : elle se connecte à deux ou plusieurs classes et possède également des attributs et des opérations. Une classe-association est caractérisée par un trait discontinu entre la classe et l'association.

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations ou méthodes) et de relations avec d'autres objets (les associations et les agrégations).

Une classe de conception est composée de :

- **Attribut :** chaque attribut d'une classe est le même pour chaque instance de cette classe
- **Méthodes :** elle définit le comportement d'une classe elle-même, et non le comportement de ses instances qui peut être différent.

3. Les outils utilisés :

3.1 Le système d'exploitation :

L'usine **IECO** est déjà équipée d'un réseau local qui relie ses différent services Ces divers services partagent un certain nombre d'informations, autrement dit, les employés de différents niveaux vont être connectés aux applications relatives à leur cadre de spécialité. Et puisqu'on a la notion de partage de données, il est nécessaire que les postes soient équipes d'un système d'exploitation serveur. Le choix primordial est fixé sur l'environnement WINDOWS7.

3.2 L'OUTIL DE DEVELOPPEMENT :

- **C#** :

C'est un langage de programmation moderne et orienté objet permettant de créer des programmes simples et robustes. Spécifiquement conçu pour exploiter les spécifications CLI, C# est le principal langage du **Framework Microsoft .NET**. C# était le langage à choisir grâce aux différentes bibliothèques **Open Source** disponible qui ont faciliter le développement de ce logiciel.

3.3 LES OUTILS LOGICIELS :

3.3.1 Outil de modélisation UML :

- StarUml :

Nous avons exploité pour la modélisation UML de l'application l'outil Entreprise Architect qui est flexible, complet et puissant, conçu pour les plateformes Windows. C'est un outil de création de modèles dont le langage est UML.

3.3.2 Outil d'administration de la base de données :

- MySQL Workbench :

On s'est servi de MySQL Workbench comme un logiciel de gestion et d'administration de bases de données MySQL, il possède une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL.

3.3.3 Outil de programmation du logiciel :

- Visual Net :

C'est un jeu complet d'outils de développement permettant de générer des applications **Web ASP**, des services **Web XML**, des applications bureautiques et des applications mobiles. **Visual Basic .NET**, **Visual C++ .NET**, **Visual C# .NET** et **Visual J# .NET** utilisent tous le même environnement de développement intégré (**IDE, *Integrated Development Environment***), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du .NET Framework, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de services **Web XML.**

3.3.4 Outils d'interaction avec les automates :

- CX-Programmer :

CX -Programmer, le logiciel de programmation pour toute la série d'**API** d'Omron, est entièrement intégré dans la suite logicielle **CX-One**. CX -Programmer comprend une grande variété de fonctionnalités pour accélérer le développement du programme **PLC.** C'est un outil de programmation, configuration et extraction de code de l'automate Omron.

4. Conclusion :

Ce chapitre nous a permis d'avoir un aperçu des langages de modélisation et de logiciel de développement utilisés dans la suite de ce travail, afin d'évité les introduire à chaque étape. Le chapitre suivant sera consacré à la conception de ce système de supervision.

Chapitre 4 : étude conceptuelle.

1. Introduction :

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation en s'appuyant sur le formalisme UML introduit dans le chapitre précédent.

2. Les diagrammes des cas d'utilisations :

Dans le premier chapitre, on a pu voir le diagramme de cas d'utilisation globale, dans ce qui suit on va détailler ce cas.

2.1 GERER LES COMPTES :

La gestion des comptes ayant accès au système est assurée par le Directeur Technique. Celui-ci se charge d'***ajouter les différents utilisateurs***, ce cas permets à chaque utilisateur du système d'avoir son compte, comme il ***attribut les droits d'accès à l'utilisateur*** afin de garantir la fiabilité du système. Ceci est modélisé par le diagramme de cas d'utilisation suivant :

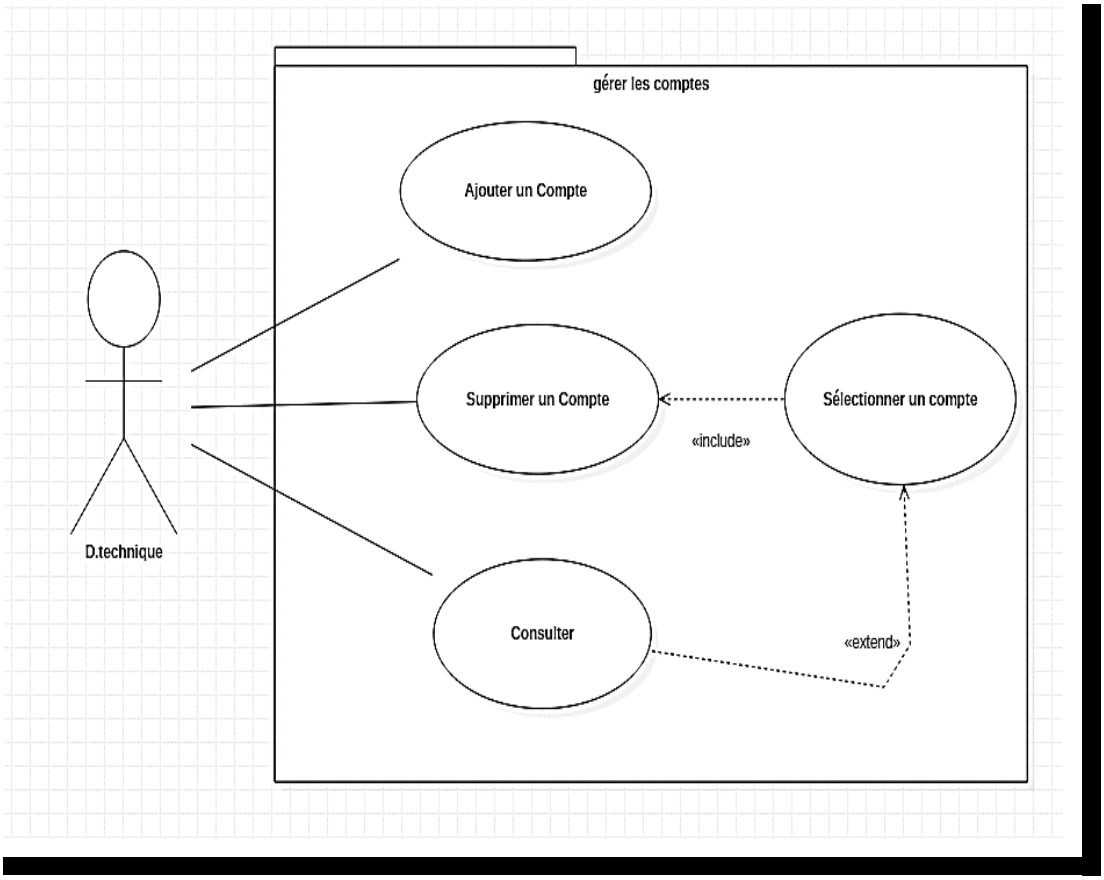


Figure 21 : Cas d'utilisation : Gérer les comptes.

2.2 GERER LES CLIENTS :

Effectué par le directeur technique et le chef d'équipe, ce cas d'utilisation comporte ***la gestion des clients***. Aussi, il permet de ***visualiser la variation de ces derniers***.

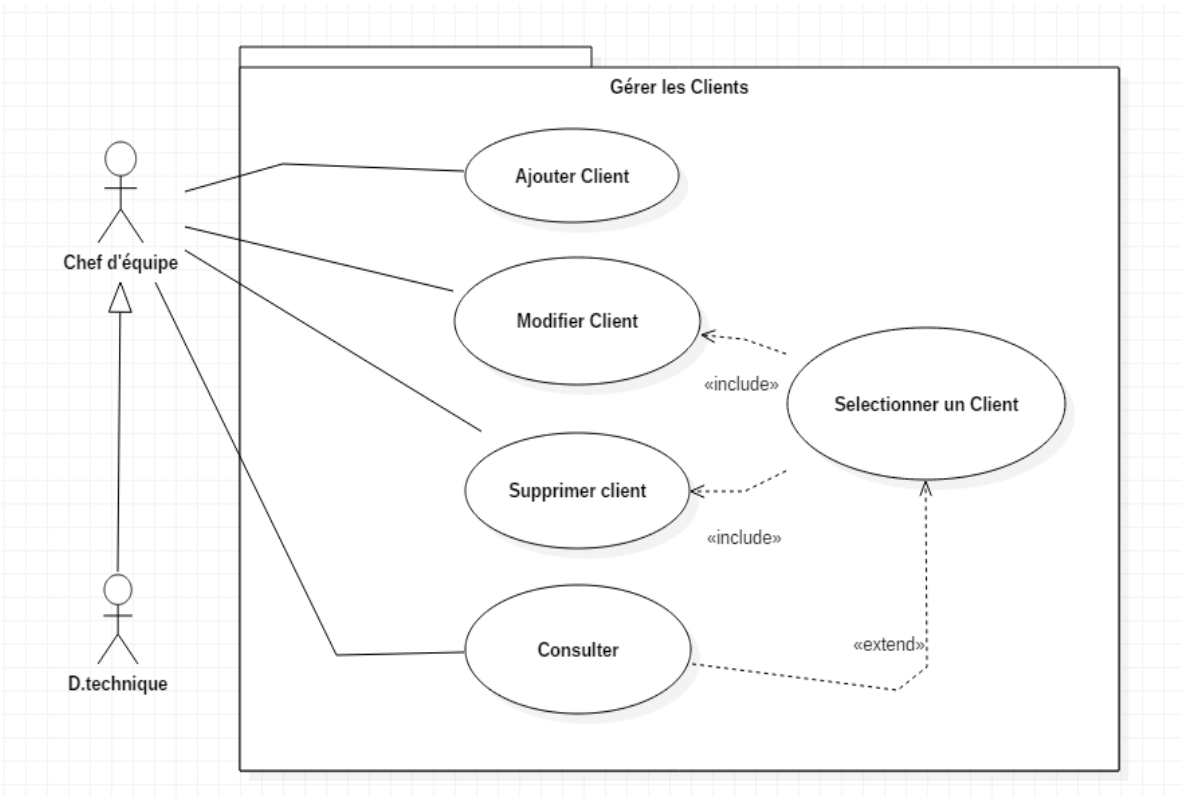


Figure 22 : Cas d'utilisation : Gérer les clients.

2.3 GERER LES PRODUITS :

Comme le cas d'utilisation précédent, la gestion des produits se fait par les mêmes acteurs, ce cas permet ***d'ajouter, modifier, supprimer*** et ***consulter la liste des produits***.

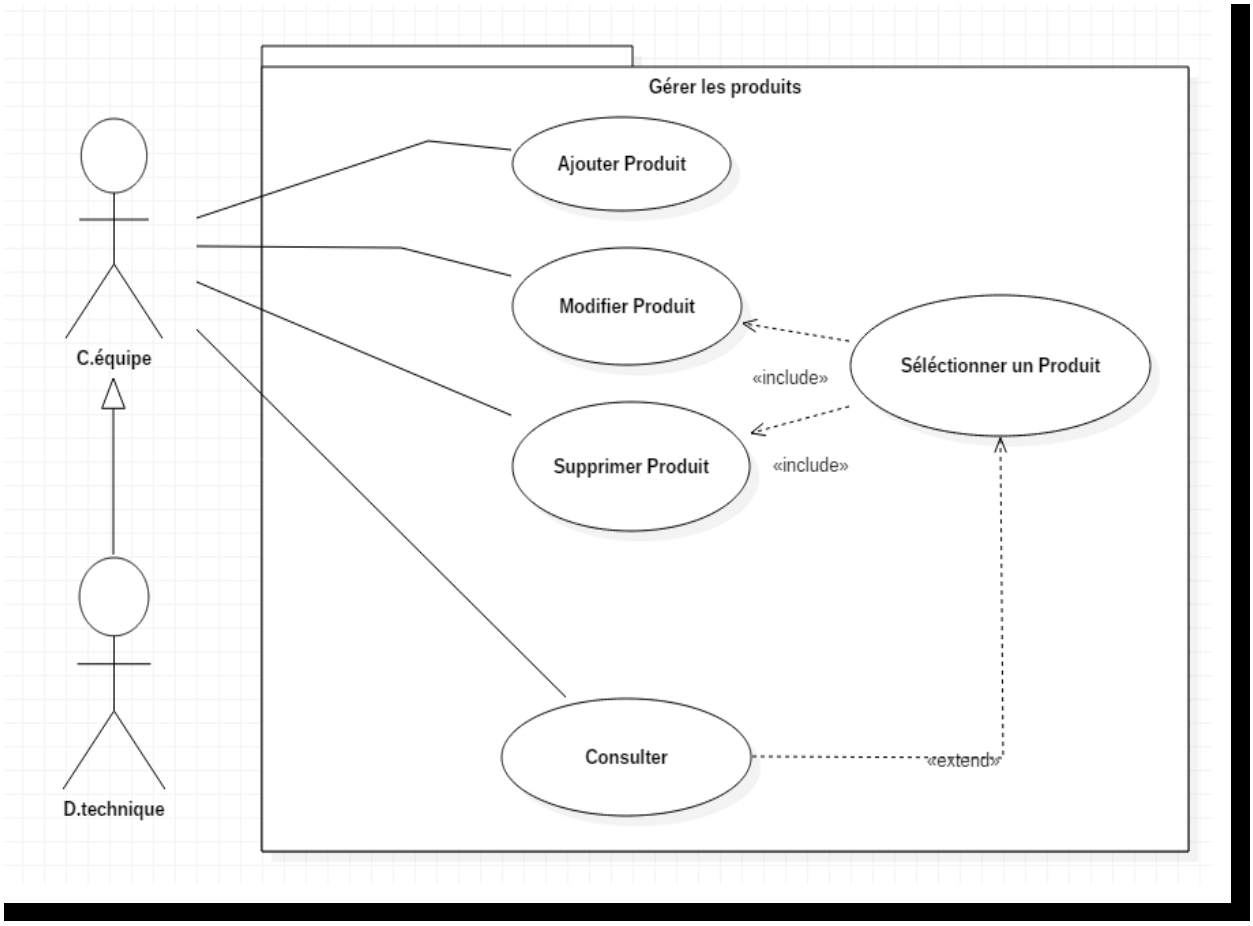


Figure 23 : Cas d'utilisation : Gérer les produits.

2.4 Gérer les commandes :

Ce cas modélisé dans la figure suivante, permet de ***saisir la commande*** du carton qui sera produite, une commande a son propre code, une quantité, et trois dimensions (X, Y, Z), gérer les commandes permet aussi ***la modification***, ***la suppression,*** et ***la consultation*** des commandes.

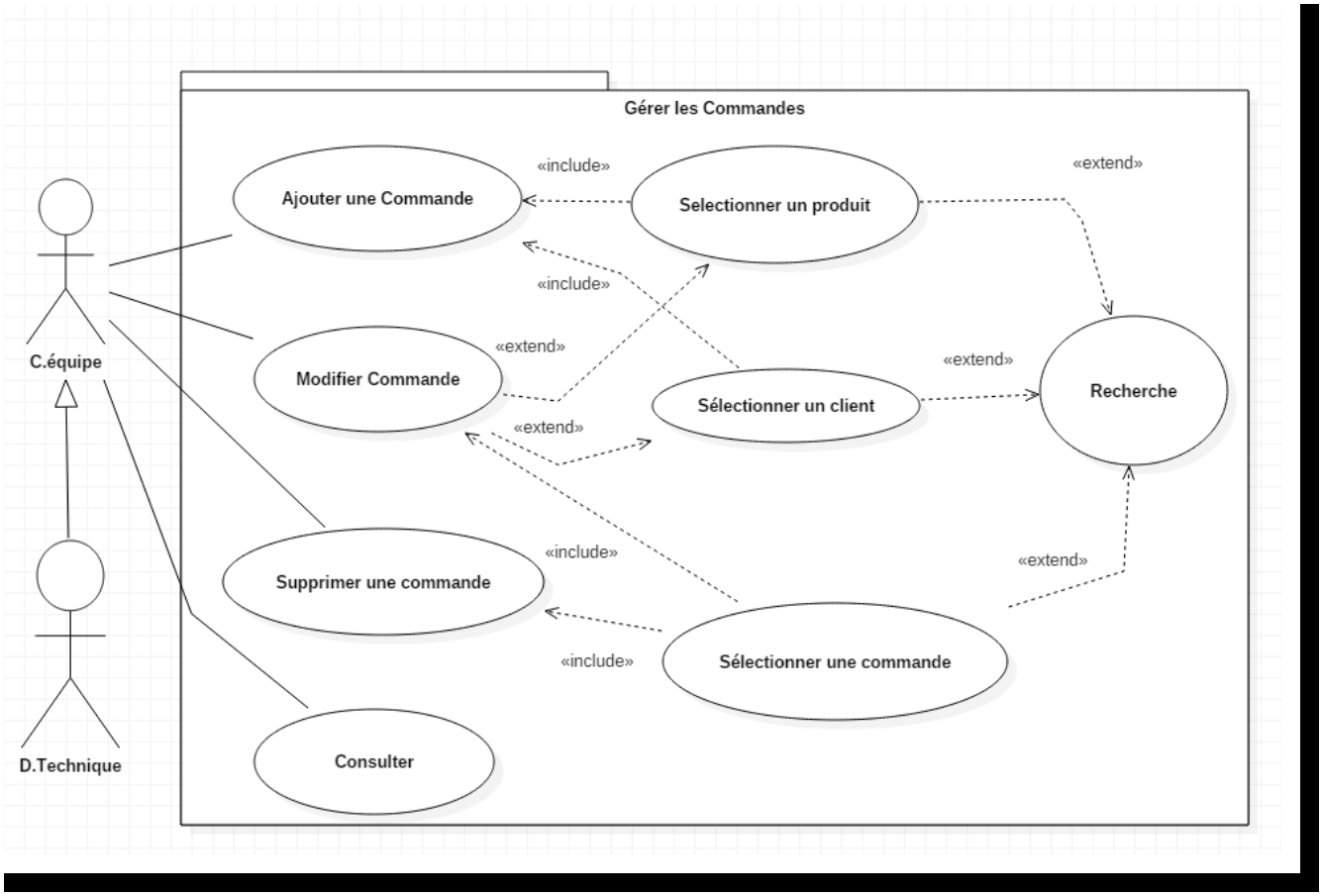


Figure 24 : Cas d'utilisation : Gérer les Commandes

2.5 Traiter les commandes :

Le traitement des commandes, un cas d'utilisation réalisé par l'opérateur. Avant de lancer la production du carton, une commande doit être sélectionnée ou insérée. Ce traitement se fait au niveau des trois **API**, c'est-à-dire les trois machines :618,2200 et 4500.

Le lancement de la commande permet au logiciel d'être prêt à collecter et sauvegarder depuis l'API les valeurs de :

- ***La vitesse.***
- ***Le nombre des boites.***
- ***Le nombre des paquets***.

Le traitement des commandes suit le même scénario pour les trois machines.

- Le cas d'utilisation ***Fin session*** : ce cas permet à l'opérateur de quitter sa session de travail, l'interface d'authentification sera directement affichée.

- Le cas d'utilisation ***Fin journée*** : ce cas permet à l'opérateur de quitter lorsqu'une journée de production est achevée, cette Fin génère automatiquement un rapport de production de la journée, contenant des informations telles que :
  - ***Les commandes exécutées***.
  - ***La vitesse moyenne***.
  - ***La totalité du cartons produits***.
  - ***L'opérateur effectuant la commande.***
  - ***(Les arrêts et leurs durés) //reformulation.***
- Le cas d'utilisation ***gérer les arrêts*** : ce cas permet d'enregistrer les arrêts qui se produisent lors de l'exécution d'une commande X, selon le cahier de charge, la production s'arrête pour une des raisons suivantes :
  - ***Changement d'ordre, c'est-à-dire, fin de production d'une commande X.***
  - ***Une Panne***
  - ***Une Recharge.***
  - ***Autre (pose de déjeuner par exemple).***

  La gestion des arrêts a pour objectifs de sauvegarder :
  - ***Le type d'arrêts d'une commande X***
  - **La durée de cet arrêt.**

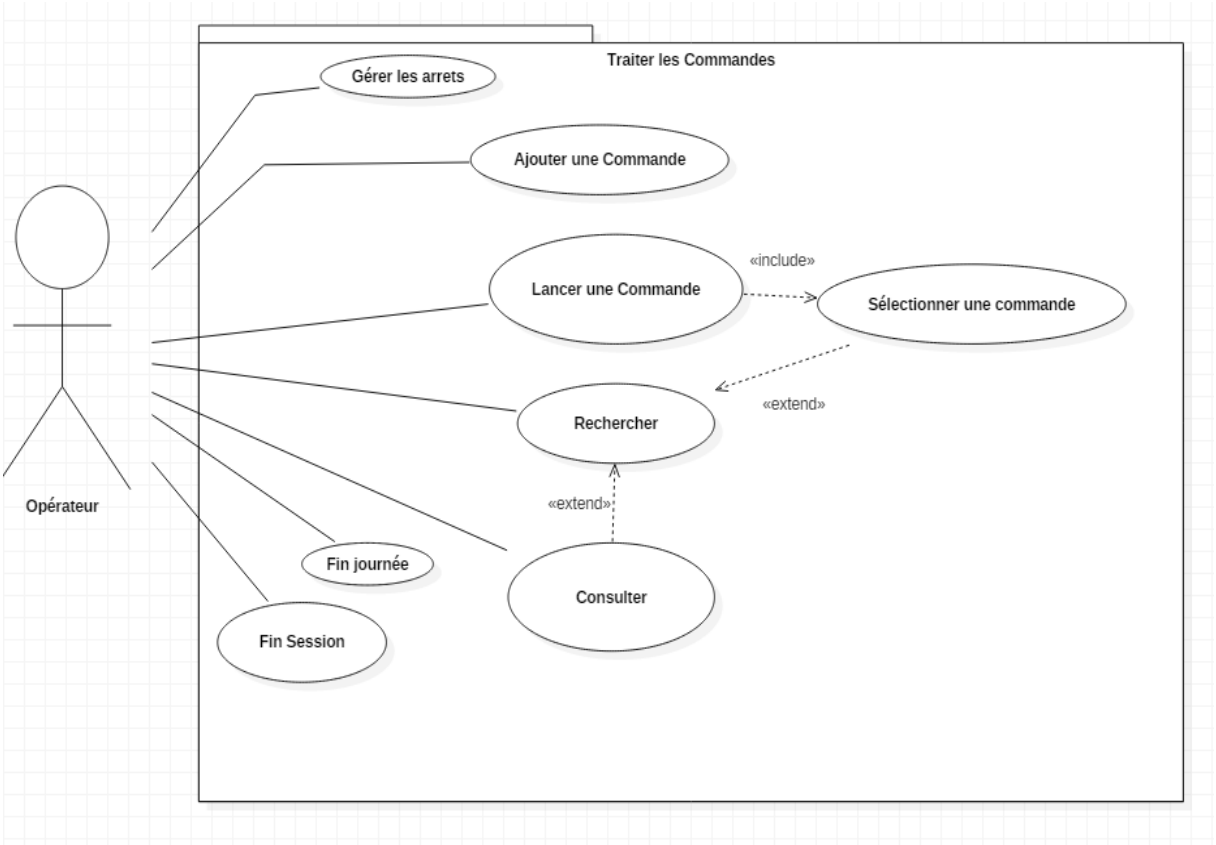


Figure 25 : cas d'utilisation : Traiter les commandes.

3. Diagramme d'activité :

Le diagramme d'activité illustré dans la figure suivante (figure 27), permet de modéliser le cas d'utilisation ***: traitement de commande***, effectué par un Opérateur.

Les commandes d'une journée sont saisies par le chef d'équipe.

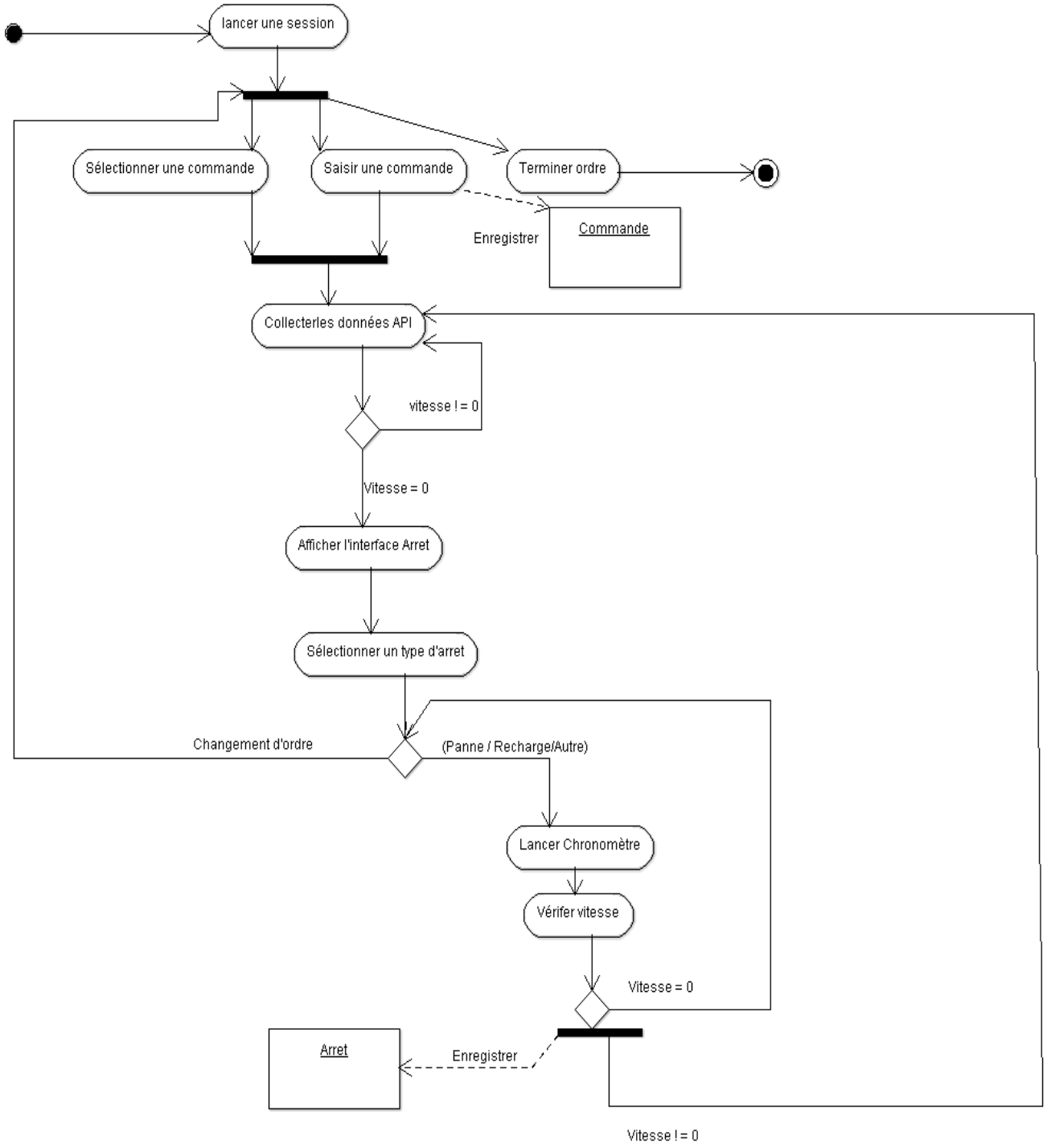
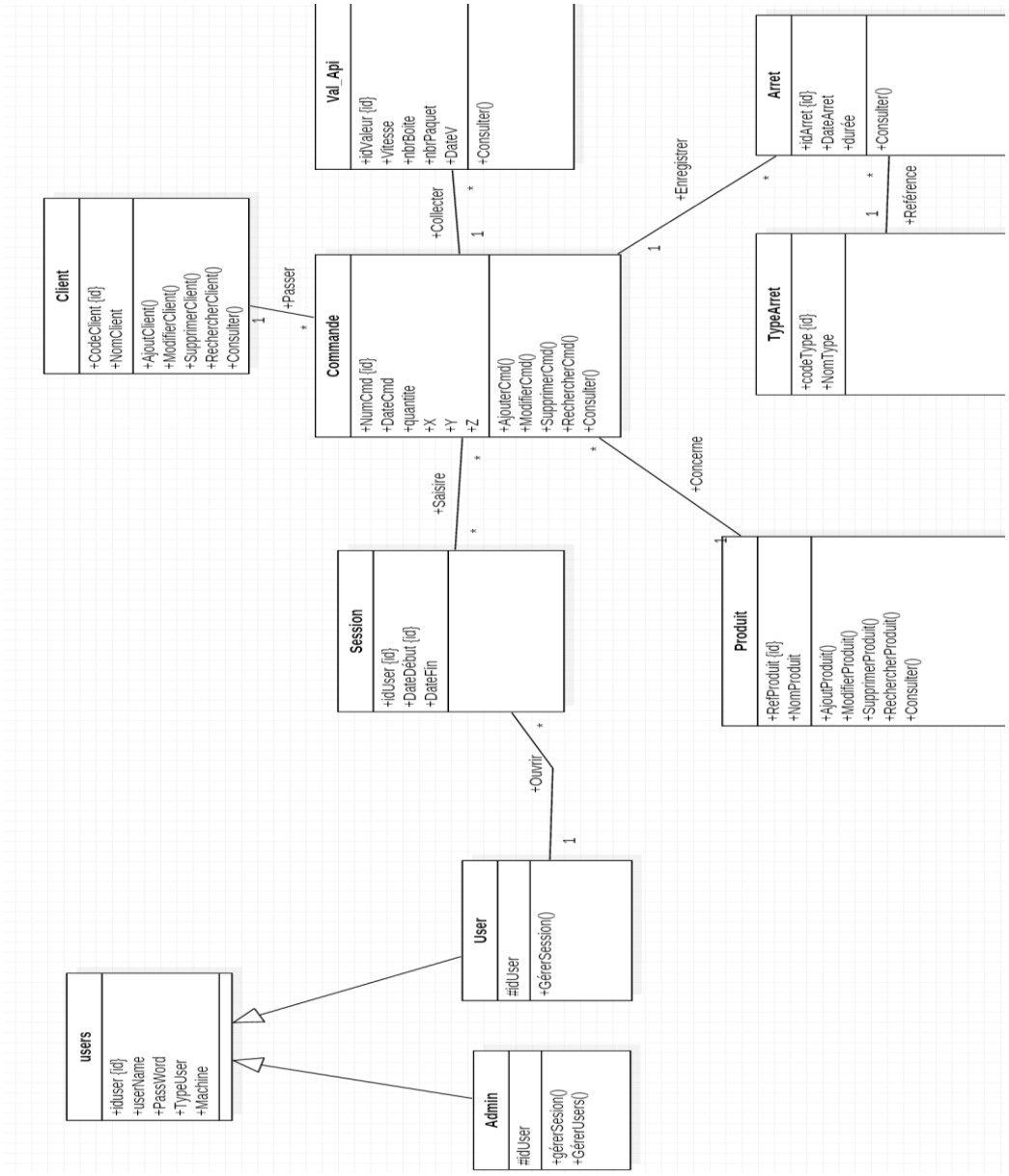


Figure 26 : Diagramme d'activité

4. Diagramme de classe :



[4]Figure 27 : Diagramme de classe.

[4] Les attributs (date) sont de type DateTime.

5. Schéma de la base de données relationnelle :

La transcription d'un MCD en modèle relationnel s'effectue selon quelques règles simples qui consistent d'abord à transformer toute entité en table, avec l'identifiant comme clé primaire, puis à observer les valeurs prises par les cardinalités maximum de chaque association pour représenter celle-ci soit par l'ajout d'une clé étrangère dans une table existante, soit par la création d'une nouvelle table dont la clé primaire est obtenue par concaténation de clés étrangères correspondant aux entités liées.

La supervision concerne trois machines différentes, 618 est une case-maker, la 2200 pour les barquettes et la 4500 pour l'emballage des électroménagers. Cette différence nous incite à modéliser trois bases de données indépendantes :

Schéma 1 : **Machine 618**

Users (idUser, nomUser, Type, machine).

Session (#idUser, DateDebut, DateFin).

Produit (RefProduit, nomproduit).

Client (CodeClient, nomClient).

Commande (NumCmd, DateCmd, quantité, X, Y, Z, #RefProduit, #CodeClient).

Saisir (#idUser, #DateDebut, #NumCmd).

Val_Api (idV, vitesse, nbrBoite, nbrPaquet, DateV, # NumCmd).

TypeArret (codeType, NomType).

Arrêt (idA, dateArret, durée, #codeType)

Ce schéma est le même pour les trois bases de données, sauf que la différence entre le premier et les deux autres (schéma de la base de données 2200 et le schéma de la base de données 4500) l'absence de l'attribut ***nombre de paquet,*** la table Val_Api devient donc :

Val_Api (idV, vitesse, nbrBoite , DateV, # NumCmd).

6. Conclusion :

Dans ce chapitre, en utilisant UML, nous avons fait la description des diagrammes des cas d'utilisation, de classe et d'activité, afin de délimiter le cadre de notre travail et de préparer un terrain favorable pour la prochaine étape. Maintenant, notre application est prête à être codée. Dans le chapitre suivant, nous allons nous intéresser à l'implémentation de notre système en se basant sur la conception détaillée de ce chapitre.

Chapitre 5 : Réalisation.

1. Introduction :

Dans le chapitre 4 (Etude conceptuelle) précédent, la conception de cette application a été entreprise avec :

- Les diagrammes des différents cas d'utilisation.
- Les diagrammes d'activité.
- Le diagramme de classe.
- Et le schéma relationnel des bases de données.

Dans le présent chapitre nous allons traiter les étapes clés de la réalisation et de l'implémentation de l'application. Nous commençons par la description des étapes de réalisation. Suivi par la suite, d'un aperçu pratique sur le travail accompli (au cours de la période de développement), sous forme de prises d'écran.

2. Etapes de réalisation :

La clé, pour réaliser un **système de supervision** des **Automates Programmables**, c'est d'assurer une connexion fiable avec ces automates et de localiser les adresses des valeurs à superviser. Dans cette partie nous présenterons la procédure suivie pour localiser les valeurs à superviser et pour récupérer les adresses IP d'un automate

Dans la deuxième partie, nous expliquerons la configuration faite sur le terrain pour se connecter au **CJ2M**.

2.1 Localisation des adresses mémoires :

Cette étape est primordiale, il faut connaitre l'adresses exacte des zones mémoires des variables que l'on voudrait superviser. La localisation de ces adresses pour les trois machines nécessite l'extraction du programme des machines en utilisant *cx-programmer* et l'analyse du code écrit en langage ***Ladder.***

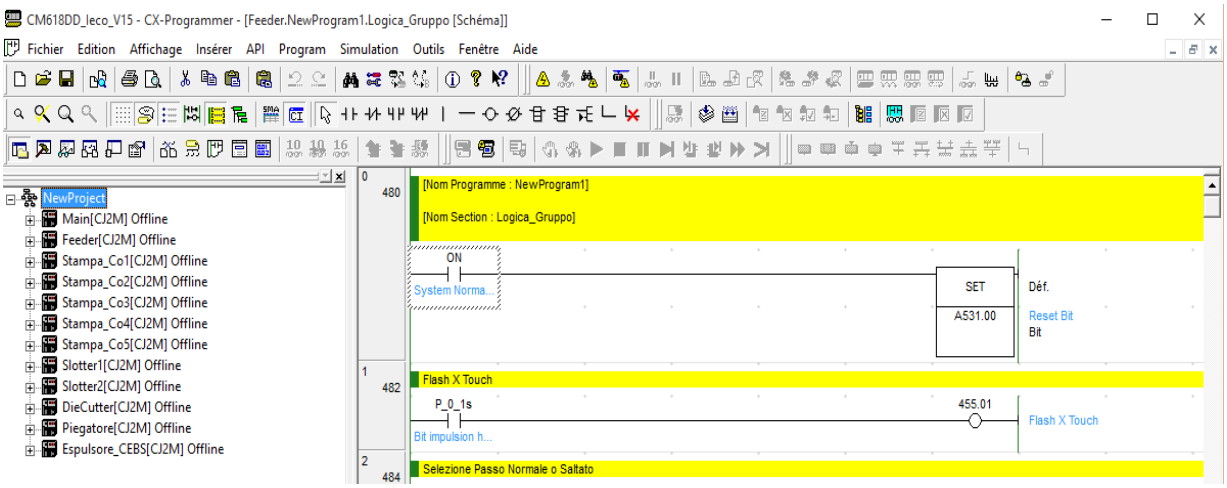


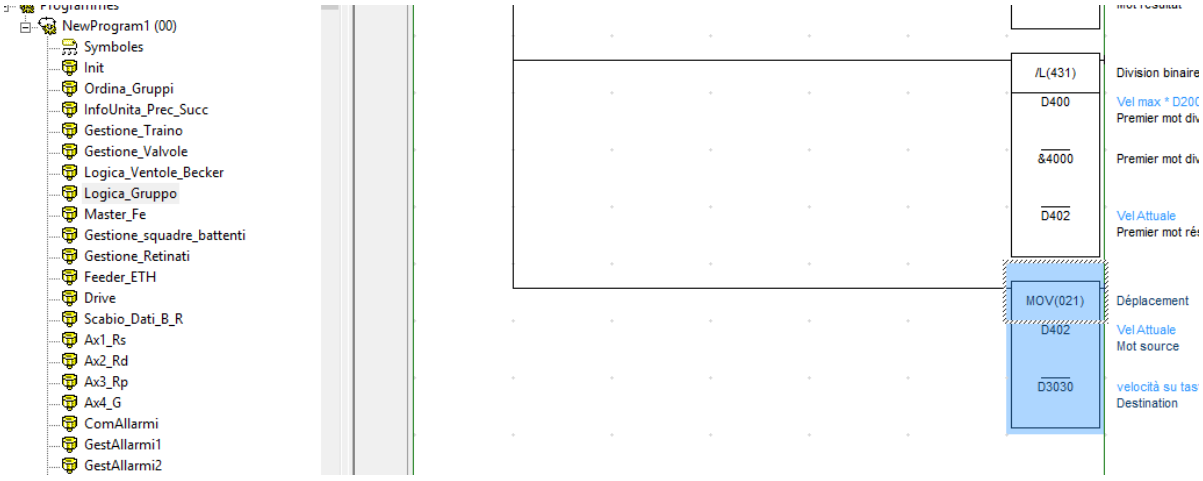
Figure 28 : Programme de la machine 618.

Comme illustré sur la figure 1, le programme d'une machine est composé de sous-programmes plus le Main (le programme principale). Le but est de trouver les adresses mémoires qui stockent :

a. ***La valeur finale de la vitesse de la machine***.

b. ***Le nombre de boites.***

c. ***Le nombre de paquets*** (dans le cas de **Case-Maker**)



5

Figure 29 : Adresse mémoire contenant la valeur de la vitesse-machine618-.

La figure 29 montre l'adresse mémoire contenant la valeur finale de la vitesse qui est **DM030,** qui se trouve dans le sous-programme **Feeder**. En parcourant le reste du code de la

[5] « **Velocità su tastiera »** signifie *vitesse sur le clavier* et « **vel attuale »** signifie *vitesse actuelle*.

machine 618 on trouvera que **D3020** contient le nombre de boites et **D3022** stocke le nombre de paquets.[6]Ces adresses seront utilisées dans le code de notre logiciel développé.

Après la localisation des adresses mémoires et les adresses IP des API (la figure 3), en utilisant *cx-programmer*, nous passerons à la connexion **API-PC** (**PLC-PC).**

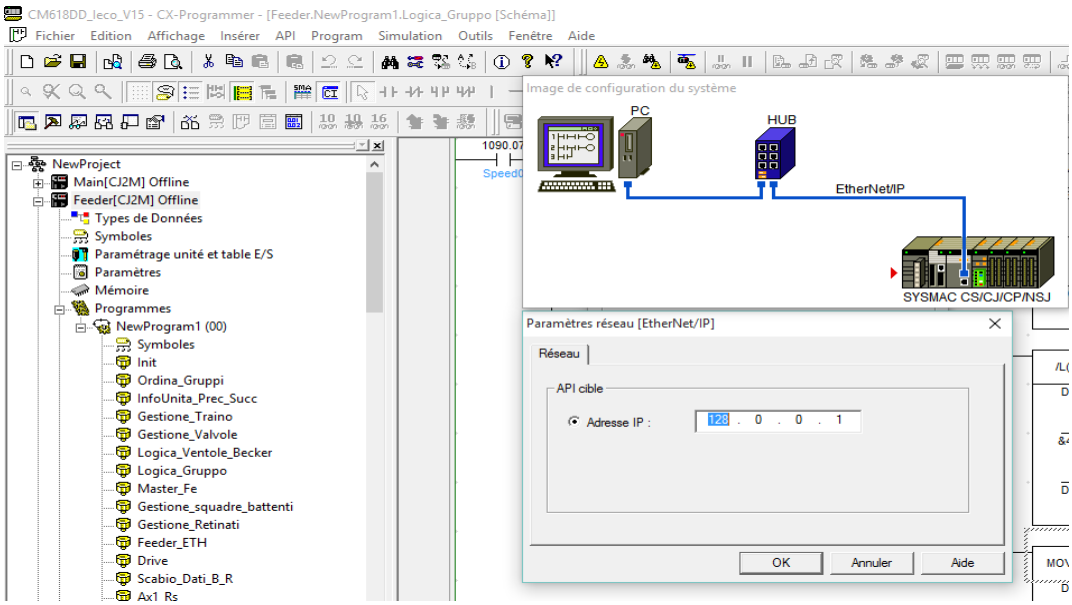


Figure 30 : Adresse IP du bloc contenant les adresses cibles.

2.2 Connexion PC-PLC :

La liaison PC-PLC est possible à réaliser parce que les modules de communications fournissent l'intelligence pour le routage des commandes ou des données. La mémoire de l'API est organisée de telle sorte que les demandes de communication peuvent accéder aux zones de données dans le processeur sans interrompre la fonction de commande de CPU.
La mémoire de données fournit un Scratch Pad, c'est-à-dire, un bloc-notes pour obtenir des informations à écrire et lire. Cela permet à l'utilisateur de désigner les données souhaitées sans interférer avec l'exécution du programme de commande.

La dernière clé pour assurer cette liaison est l'unique middleware disponible sur Omron, c'est l'outil de communication FINS, il permet aux messages et aux données d'être acheminés en toute transparence, comme cela a été expliqué dans le chapitre 2, volet 2-.

[6] A noter que les mêmes procédures ont été exécutées sur les deux autres machines.

2.2.1 PLC Setup :

En utilisant cx-programmer, ces changements doivent être effectués pour vérifier ou modifier les paramètres de la communication FINS (numéro de port par exemple), Les étapes montrées dans les figures suivantes ont été faites sur les trois machines avant de lancer la connexion depuis les IHM **(Interface Homme Machine)** liés à ces API :

1) Se connecter à l'API Omron via CX-Programmer et activer le mode « Online »

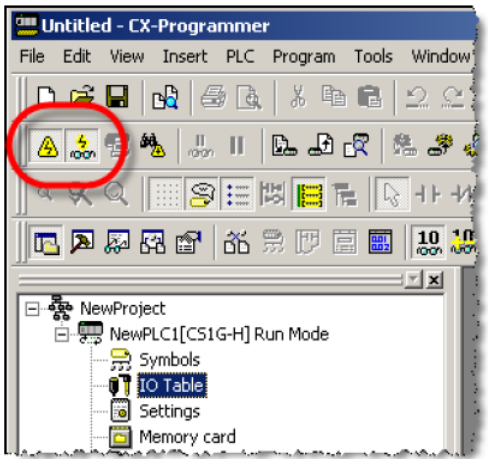


Figure 31 : Activation du mode en ligne.

2) Transformation de toute toutes les configurations à partir de l'API.

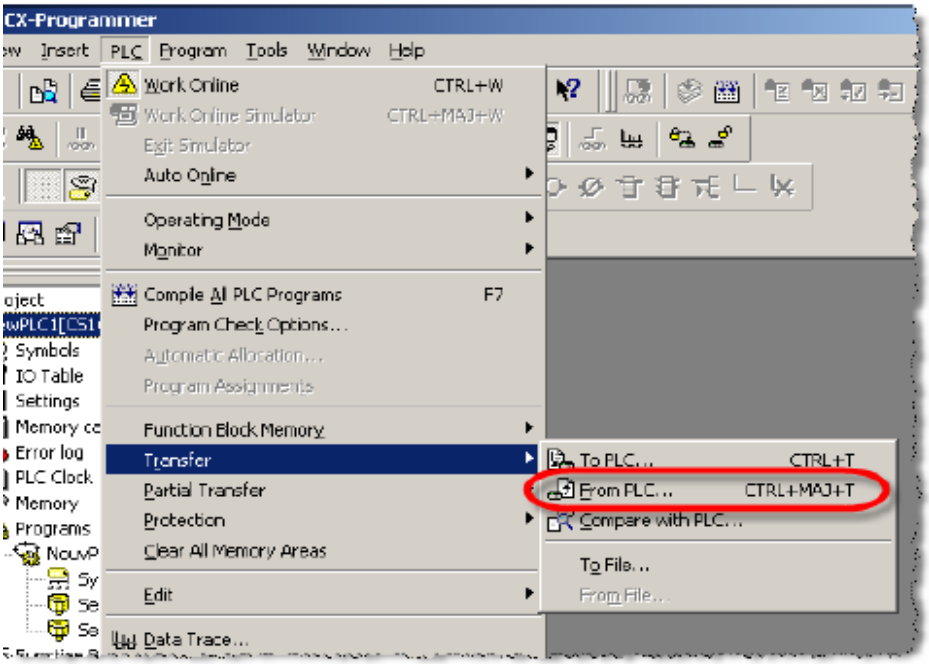


Figure 32 : Transformation des configurations.

3) Arrêter l'api en basculant au « program mode »

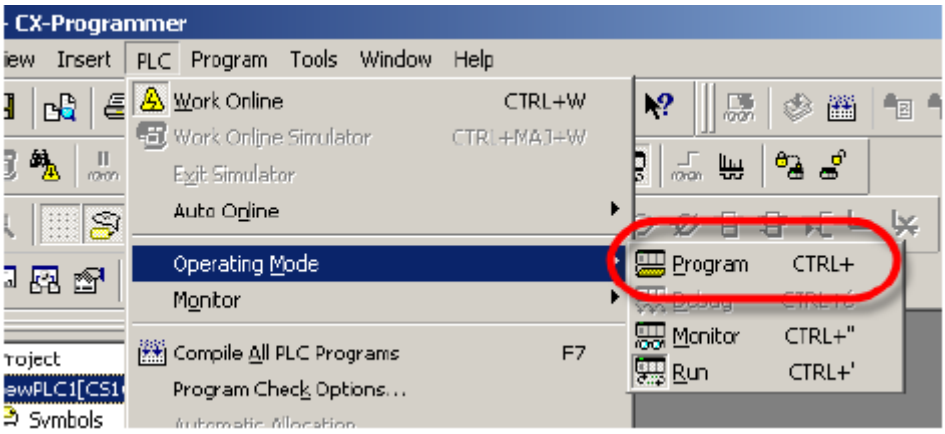


Figure 33 : Passage au "program mode ".

4) Accéder aux paramètres de la carte Ethernet :

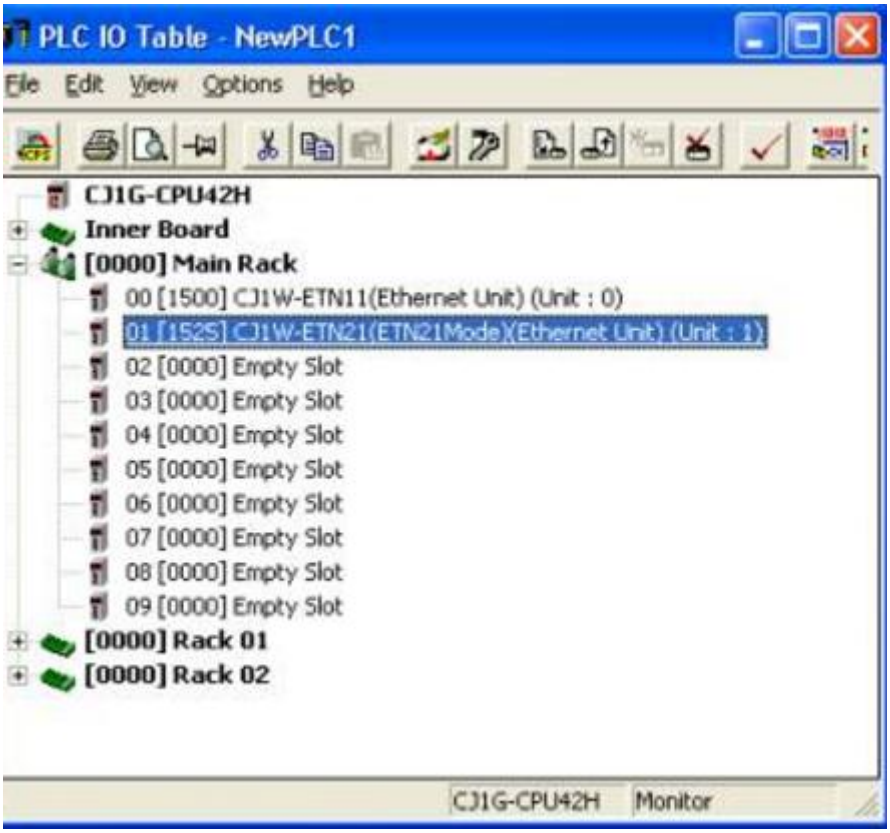


Figure 34 : Paramètres de la carte Ethernet.

5) Configuration / vérification des paramètres FINS / TCP :

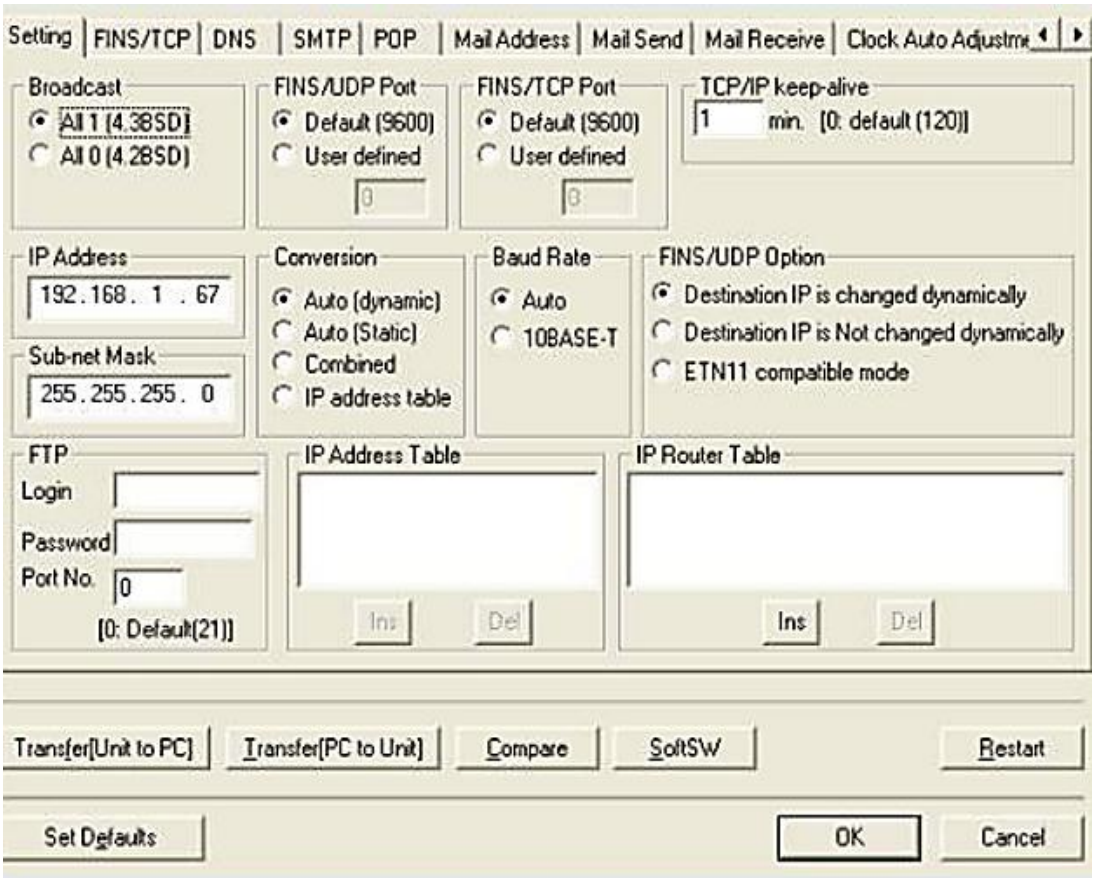


Figure 35 : Paramètres FINS / TCP.

Cette figure montre l'interface du logiciel *Cx-programme* qui nous permet de configurer / vérifier les paramètres du FINS / TCP :

- **Adresse IP :** c'est l'adresse IP de l'unité (l'API)**.**
- **Masque du sous réseau :** souvent, **255.255.255.0.**
- **TCP/ IP Keep -Alive :** cette fonctionnalité permet de spécifier une valeur (1 ou 5minutes) qui est le délai d'attente du socket, et de reconnecter en cas ou des problèmes de réseau sont détectées.
- **Port TCP/IP :** par défaut 9600.

Après cette configuration, il est recommandé de transférer les paramètres et de redémarrer l'Automate[7].

3. Les captures d'écran de l'application :

L'interface graphique est une partie très importante pour la réalisation d'un logiciel convenable et conviviale offrant du plaisir à l'utilisateur lors de la navigation. Ainsi, ce critère peut faire la différence entre une application et d'autres, bien qu'elles aient les mêmes fonctionnalités.

Voici maintenant un ensemble de captures d'écrans sur les principaux points d'entrées de l'application.

A. Authentification :

Figure 36 : Interface authentification.

C'est la première interface qui s'affiche après l'exécution du logiciel. La saisie du bon ID transportera l'utilisateur à son espace. Chaque connexion valide insérera dans la table Session de la base de données appropriée l'utilisateur ainsi que sa date de connexion.

[7] Ces configurations ont été mentionné dans les Data Sheet de l'API CJ2M fournies par Omron.

B. Espace Admin :

Figure 37 : Interface Admin.

Comme son nom l'indique, cet espace est dédié à l'administrateur du logiciel i.e. le directeur technique. L'espace est illustré dans la Figure 37, cette interface permet les ***ajouts***, les ***modifications*** et les ***suppressions*** des comptes des ***utilisateurs***, des ***clients,*** des ***produits*** et des ***commandes***.

Figure 38 : Interface gestion des comptes

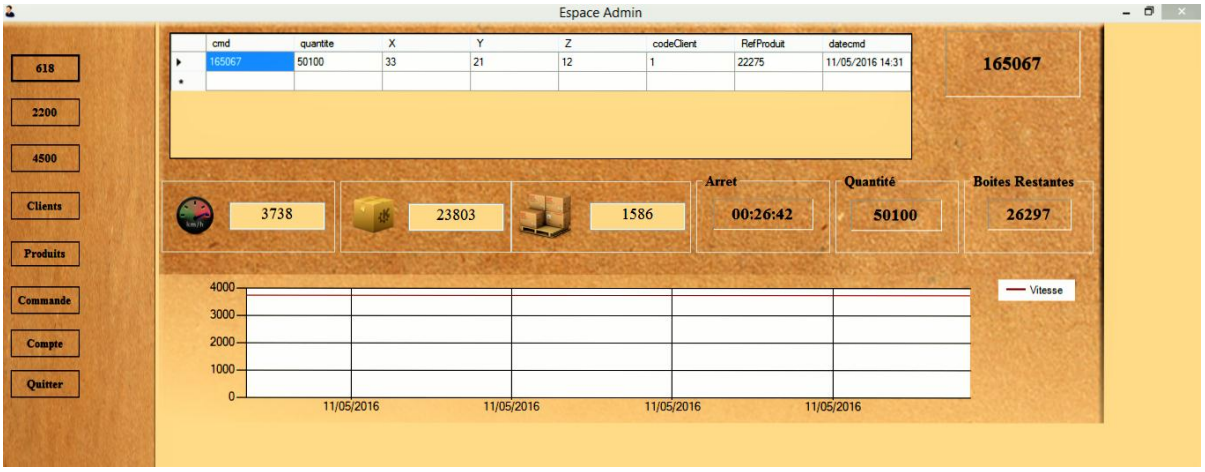


Figure 39 : Interface de supervision de la machine 618.

Le Directeur Technique peut aussi superviser en temps réel la production d'une commande, les boutons 618 2200 et 4500 lancent les interfaces de supervisions pour chaque machine.

Comme le montre la douzième figure, le Directeur Technique pourra consulter la commande en cours de traitement :

- ***La vitesse de production.***
- ***Le nombre de boites.***
- ***Le nombre de paquets produits*** par la machine
- ***La durée totale des arrêts***.
- ***Le nombre de boites restantes à fabriquer.***

L'interface affiche aussi un graphe permet de schématiser la vitesse de la machine en fonction du temps ainsi une table contenant la liste des commandes de la journée.

C. Espace Chef d'équipe :

Son espace lui permet de ***créer***, ***modifie***r, ***supprimer*** et ***consulter*** :

- ***Les clients.***
- ***Les produits.***
- ***Les commandes d'une journée.***

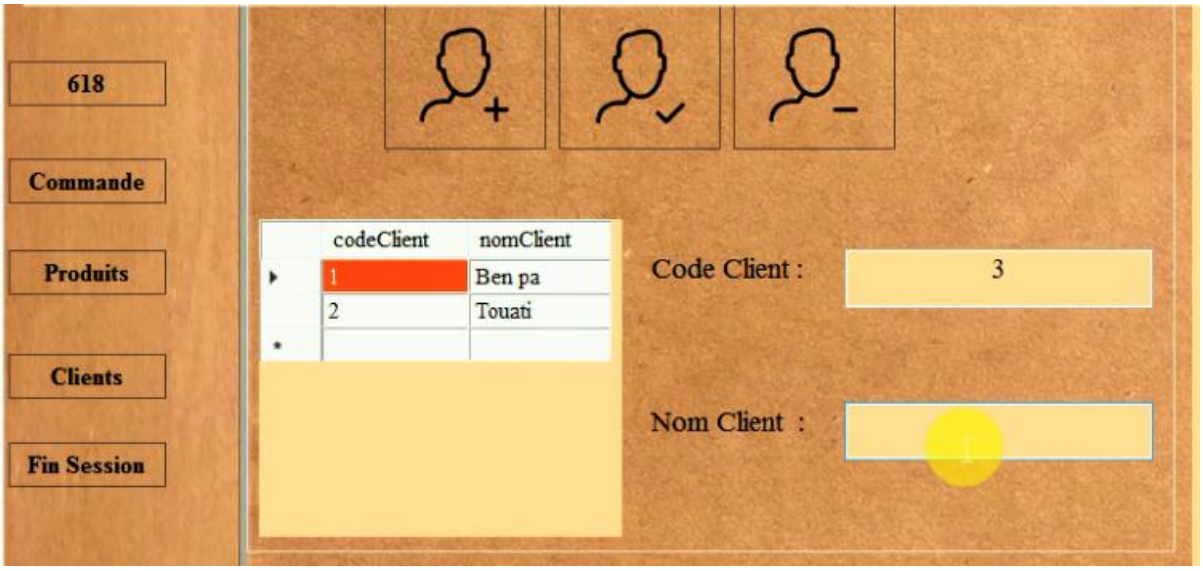


Figure 40 : Interface de gestion les clients

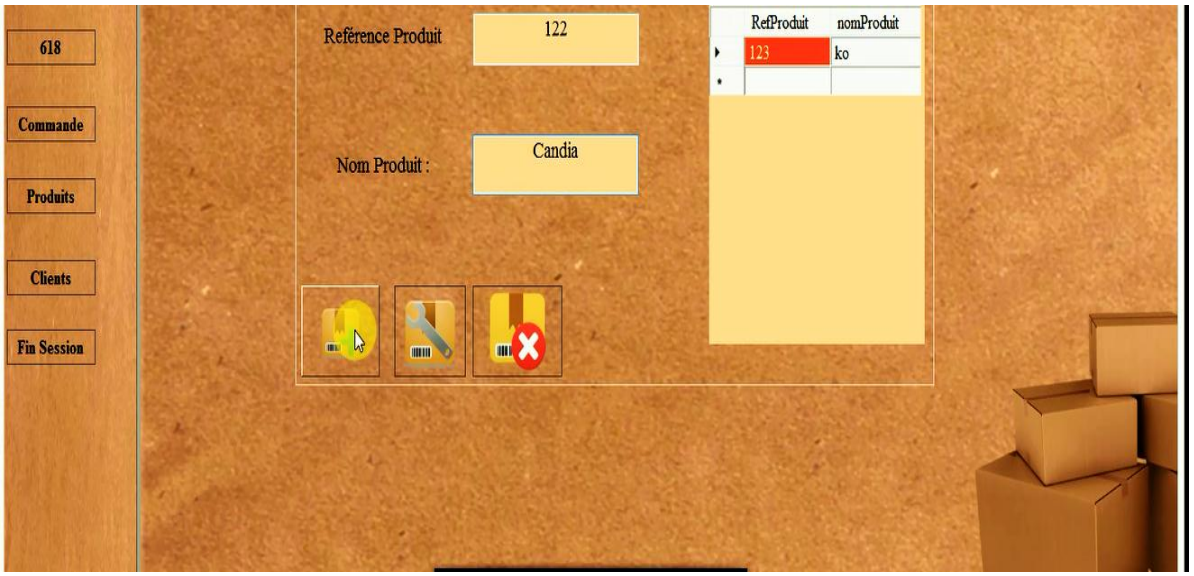


Figure 41 : Interface de gestion des produits

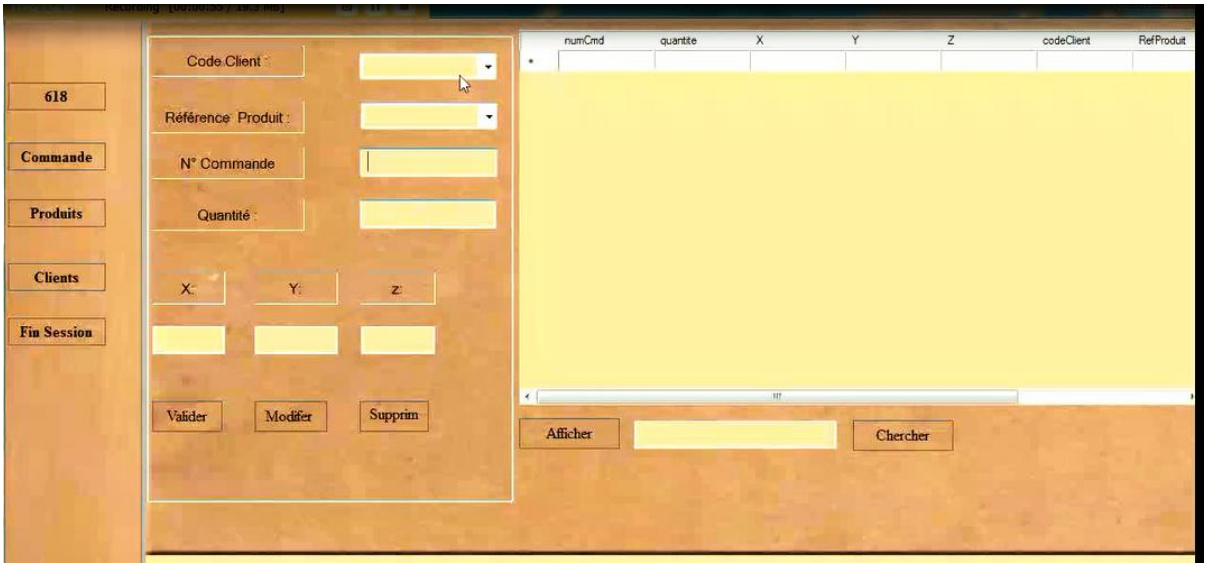


Figure 42 : Interface de gestion de commande

D. Espace Opérateur :

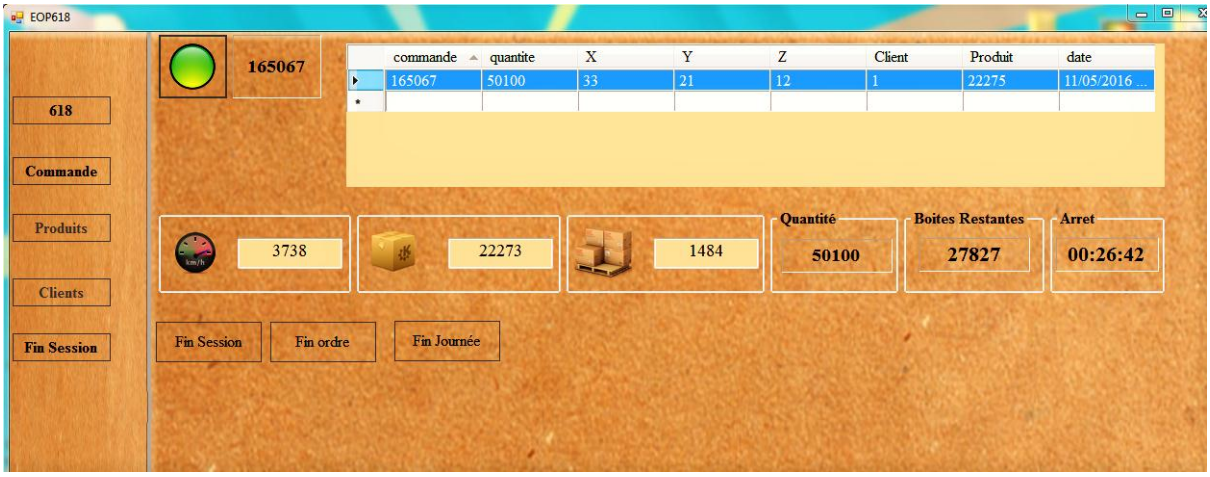


Figure 43 : interface espace opérateur.

L'opérateur est responsable de la production. Son espace lui permet de sélectionner une commande existante dans la table des commandes de la journée. La sélection de cette commande permet d'activer le bouton vert afin de lancer la collecte des données depuis l'API.

- L'outils ***Fin Session*** : à l'issu de la séance de travail, l'utilisateur met un terme à sa session, en quittant l'interface. L'outil "Fin de Session" envoie alors une requête à la table ***Session*** de la base de données appropriée pour insérer la date de sortie
- L'outils ***fin ordre*** : permet à l'opérateur de mettre Fin à un traitement de commande X.

4. Conclusion :

La dernière partie de ce projet est dédiée :

1) Aux étapes principales de la réalisation.

2) À la navigation dans notre application,

Ce rapport a été illustré par la présentation des impressions écrans décrivant nombreuses interfaces de notre outil.

Conclusion Générale :

Tout au long de ce rapport, nous avons présenté les différentes étapes de réalisation d'un ***système de supervision d'une unité de production d'emballage de carton ondulé***, au sein de la société **IECO Emballage**. Nos travaux de recherche et de développement ont abouti à la création d'une plateforme de supervision SCADA, comportant un ensemble de stations reliées entre elles, ainsi qu'une application conviviale, en soutien à la production :

- Prise en charge informatisée des commandes de production, de la direction centralisée jusqu'à l'opérateur exécutant sur la machine ;
- Actualisation et enrichissement de la base de données par les informations quotidiennes de production ;
- Gestion des arrêts sur machine ;
- Génération automatique du rapport quotidien de production…

Le présent mémoire est structuré sur cinq chapitres dont les 3 premiers sont consacrés à **la revue bibliographique du matériel et des outils logiciels** utilisés pour le développement de notre travail. Tandis que les deux derniers chapitres portent sur **la présentation de l'application développée** pour le bénéfice de l'usine d'emballage **IECO**.

Durant toute la période du stage, nous avons eu la possibilité de découvrir le milieu industriel d'emballage en carton mais le plus important pour nous c'était de voir au plus près et de manipuler beaucoup de choses et beaucoup de facettes en relation avec notre domaine de spécialité qu'est **l'informatique**

Ce projet nous 'a donné l'opportunité de s'initier à la vie professionnelle dans un cadre industriel enrichissant, complémentaire du cadre universitaire. Ainsi, ce projet nous a permis d'avoir un début d'expérience significatif. Il nous a permis de prendre conscience sur des facettes concernant le **Facteur Humain** qui est très important en **milieu coopératif interactif** tel que celui que l'on retrouve en industrie. Ce projet nous a appris ainsi :

- Comment dès le matin on peut prendre le parti de la gaieté ;
- Comment réussir de bonnes relations pour s'assurer et garantir un travail de groupe productif ;

- Comment compter sur soi, et prendre des initiatives, pour résoudre les problèmes qui se présentent ;
- Comment être méticuleux dans le travail ;
- Comment être attentif aux directives et aux indications des supérieurs ;
- Comment être bien organisé pour accomplir dans les meilleurs délais, et dans les meilleures conditions, les tâches qui nous sont confiées

- ***Perspectives :***

Le travail présenté est arrivé à sa première phase de maturité. Ce travail reste, tout de même, à parfaire et à parachever, en effet il y a plusieurs perspectives qui sont ouvertes qui peuvent être résumées dans les points suivants :

- Le premier point consiste à créer une application de pilotage des machines automatisée à distance
- Créer une application qui permet de prédire une éventuelle défaillance à partir de la table des arrêts.
- Développer un outil logiciel qui permettrait de piloter notre application (celle développée dans le cadre du projet de Master) via (à travers) un réseau internet sécurisé.

Bibliographie

BOUAMAMA, B. O. (s.d.). ***Supervision des système industriels (.ppt).***

Cx-One. (s.d.). ***Cx-one Introduction Guide.***

EskoArtwork. (s.d.). ***Emballages en carton compact & carton ondulé.***

H.LECOCQ, D. I. (2005). ***Les Automates Programmables*** *-Tome I-*. Université de Liege ,Faculté des Sciences Appliquées.

LAURENT, Q. (2011). ***Etude et mise en service d'un système de controle-commande pour une installation de pré-concentration de gaz.*** Rapport de Mémoire de Fin d'étude.

Modbus. (2004, June 04). ***Modbus application protocol specification***. Récupéré sur http://www.Modbus-IDA.org

Omron. (2000). ***Réseau FINS -Manuel STA 24-***.

Omron. (2005). ***Sysmac CS and CJ Series ,CS1W-ETN21 Ethernet Units Construction of Application ,Operation Manual.***

Omron. (2007). ***Connecting a PLC to a PC for programming using Cx One-QuickStart Note-***.

Omron. (2009). ***SYSMAC CS and CJ series, CS1W-ETN21 Ethernet units Construction of Networks. Dans OPERATION MANUAL*** (pp. 115-202).

Omron. (2010). ***Fiche Technique 006.***

Omron. (4 mars 2008). ***Ethernet Setup of Omron PLC .***

Omron. (s.d.). ***CJ2M PLC DataSheet***.

Omron industrial Automation. (s.d.). Récupéré sur http://www.omron-ap.com/service_support/

Omron. (s.d.). Omron CS CJ1W-ETN21 User Manual. Dans ***FINS / TCP Method*** **(pp. 195-208).**

Omron. (s.d.). ***SYSMAC CJ Series, CJ2M , CJ2CPU Unit Hardware , user's Manuel.***

Sébastien, R. (2002). ***Introduction aux API.***