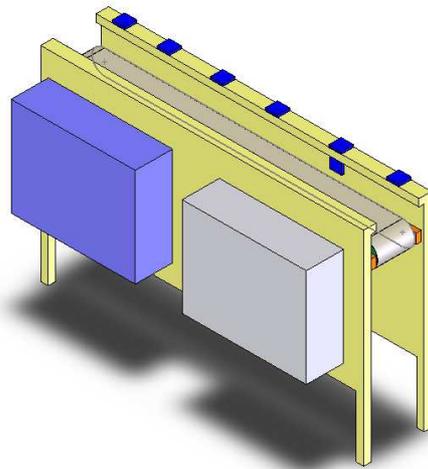


RINEAU Yoann
QUEVREUX David
LE TEXIER Nicolas

Master 1 Ingénierie Automatique
Et Informatique d'Entreprise
Promotion 2007/2008

Automatisation d'un poste de tri

Colis identifiés par code-barres et cartes RFID



Tuteurs : M. BERRUE, M. LAGRANGE, M. LAHAYE



Remerciements

Nous tenons à remercier MM LAHAYE, LAGRANGE et BERRUE pour avoir proposé ce sujet de projet. Nous les remercions aussi pour leur disponibilité et leurs conseils utiles pour l'avancement de notre projet.

Une pensée également pour M. BOULJROUFI HASSAN qui nous a beaucoup aidé sur la réalisation de la maquette. Son expérience dans le domaine électrique lui permet de nous donner des conseils judicieux sur le câblage.

Une dernière pensée aux autres groupes de projets qui travaillaient dans la même salle que nous. L'ensemble fournissait une bonne humeur et un climat tout à fait propice au travail.

Sommaire

| | |
|---|----|
| Introduction | 1 |
| I – L’environnement de travail..... | 2 |
| I.1 Le cahier des charges | 2 |
| I.2 Le code barre..... | 3 |
| I.3 Le RFID | 4 |
| I.4 Proposition d’un scénario | 5 |
| I.4.A-Synoptique de la maquette :..... | 5 |
| I.4.B-Exemple de fonctionnement avec n produit de type A :..... | 5 |
| II – La réalisation de la maquette | 8 |
| II.1 Choix des composants : | 9 |
| II.1.A-Les détecteurs de proximité..... | 9 |
| II.1.B-Les relais..... | 10 |
| II.1.C-Boite de dérivation..... | 11 |
| II.2 Construction de la maquette : | 12 |
| II.3 Schémas électriques du système :..... | 14 |
| II.4 Configuration du transpondeur RFID :..... | 18 |
| III – La programmation | 19 |
| III.1 Les nouveautés de Unity | 19 |
| III.1.A - Les particularités des variables | 19 |
| III.1.B – Les innovations dans la conception | 21 |
| III.1.C-Les logiciels spécialisés | 22 |
| III.2 Les blocs fonctions utilisées..... | 23 |
| III.2.A – Les blocs pour la gestion du temps | 23 |
| III.2.B – Les blocs pour les code-barres..... | 25 |
| III.2.C – Les blocs pour la lecture-écriture du RFID | 27 |
| III.3 L’application complète de gestion du poste de tri..... | 31 |
| III.3.A – Les parties du programme | 31 |
| III.3.B – Utilisation de notre programme sous Unity..... | 32 |
| III.3.C – Procédures et aides pour utiliser Unity Pro avec notre projet | 33 |
| Conclusion..... | 39 |

Introduction

La première année de master nous amène à travailler sur un projet. Il s'agit de mettre en application concrètement des éléments que l'on a rencontrés le long de nos études. L'objectif étant de vérifier la concordance avec nos aspirations pour notre futur métier.

Le projet se déroule sur trois mois, à raison d'un jour par semaine. Le groupe est composé de MM QUEVREUX David, RINAU Yoann et de LE TEXIER Nicolas. Les tuteurs étant MM LAHAYE, LAGRANGE et BERRUE.

Le sujet choisit est la réalisation d'un poste de tri de colis identifiés par un code barre et une carte RFID, piloté par un automate TSX PREMIUM de chez Schneider Electric. Destiné aux futurs étudiants de l'école d'ingénieur, l'ensemble se présentera en un TP qui pourra les familiariser avec les outils de traçabilité, tout en les habituant au logiciel UNITY PRO et à ses particularités.

Le poste de tri se compose d'un tapis roulant, d'une douchette pour les codes barres et d'un système de lecture/écriture pour le RFID.

Le présent rapport présente les technologies de traçabilité que sont les codes-barres et le RFID. Il expose aussi une idée de scénario que l'on a proposé qui met en œuvre l'ensemble du matériel présent afin de faire découvrir aux étudiants le maximum de chose.

Dans une seconde partie, nous présentons le travail réalisé sur la maquette afin de satisfaire notre cahier des charges. Elle présente ainsi les câblages et les choix de composants réalisés.

La dernière partie du rapport montre la partie programmation du projet : elle détaille les fonctions utilisées, les commandes à effectuer pour manipuler UNITY PRO. Elle décrit également le grafctet final qui est la réponse au cahier des charges.

I – L'environnement de travail

I.1 Le cahier des charges

L'objectif du projet est de réaliser un poste de tri automatisé de colis identifiés par un code barre et une carte RFID.

Le poste de tri sera composé :

- d'un tapis roulant
- d'une douchette de lecture de code-barres
- d'un système de lecture/écriture de cartes RFID.

Celui-ci se chargera de convoyer des colis vers une destination choisie à partir de leur identification via un code-barre et une carte RFID.

Les différentes parties à réaliser sont les suivantes :

- l'étude et la réalisation du câblage électrique du convoyeur
- l'étude et la réalisation du câblage du système d'identification RFID
- l'étude et la réalisation du câblage du système d'identification par code-barres
- la prise en main de l'atelier de programmation Unity (Schneider Electric)
- le développement d'une application automate pilotant le poste de tri

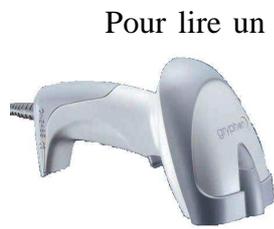
I.2 Le code barre

Le code barre est une technologie d'identification connue de tous, chaque produit que l'on achète est muni d'un code, c'est le système de traçabilité le plus répandu.

Un code barre est composé de deux parties : une série de barres blanches et noires et dessous, sa traduction dans un code alphanumérique.



Un code barre contient l'identité du produit dont il est associé. Grâce à son identité, on peut en déduire, par exemple, son prix comme à la caisse d'un supermarché.



Pour lire un code barre, il faut s'équiper d'un lecteur ou d'un scanner à infrarouge.

Celui-ci lit la série de barres blanches et noires, la traduit et l'envoie à l'ordinateur sous la forme d'une chaîne de caractère. Dans notre application, le lecteur de code barre ne sera pas connecté à l'ordinateur mais à l'automate programmable.

Bien que le code barre soit populaire dans la traçabilité, cette technologie présente quelques défauts.

Le premier est que le code barre n'est pas fiable. En effet, il peut être détérioré facilement, par exemple une tâche, ce qui rendrait son utilisation impossible par un scanner automatique.

Le second est que le code barre n'est pas évolutif. Il est impossible de modifier un code barre après une transformation subite par le produit, la solution est d'en rajouter un autre ou de le remplacer pour affecter une nouvelle identité au produit.

Le dernier défaut est que le code barre ne contient pas d'autres informations que l'identité du produit. Il aurait été intéressant pour les professionnels qu'il puisse contenir le poids, le volume, la date de péremption du produit ou encore sa date de livraison. Mais on comprend bien que l'on ne peut pas stocker toutes ces informations dans un simple code alphanumérique de treize caractères.

C'est pour palier ces défauts qu'une nouvelle technologie émerge dans le monde de la traçabilité: le RFID (Radio Frequency IDentification).

I.3 Le RFID

Le RFID (Radio Frequency Identification) est une nouvelle technologie d'identification qui va être de plus en plus répandue dans le monde industriel pour assurer la traçabilité de produit.

Comme son nom l'indique, elle fait de l'identification par radio fréquence. C'est-à-dire que chaque produit sera équipé d'un tag, une puce électronique sans alimentation électrique, qui peut être détecté, écrit et lu par un transpondeur.

Les différents modèles de tags :

Carte à puce qui est utilisée par l'homme. Exemple d'utilisation: autoriser les accès de salles.



Puce qui équipe les produits. Exemple d'utilisation: traçabilité des produits.

La communication se crée entre les deux entités lorsqu'ils sont à proximité, cette communication est sans fil.

L'apport principal de cette technologie, vis-à-vis du code barre, est la capacité d'écrire dans les tags, donc en plus de l'identité du produit la puce peut contenir diverses informations sur celui-ci. Cette nouvelle compétence va permettre au RFID de s'imposer même dans la vie de tous les jours des consommateurs. Pour exemple, le réfrigérateur RFID qui détecte tous les produits qu'il contient, de ce fait il peut alerter l'utilisateur lorsqu'un produit se périmé ou encore donner des idées de recette.

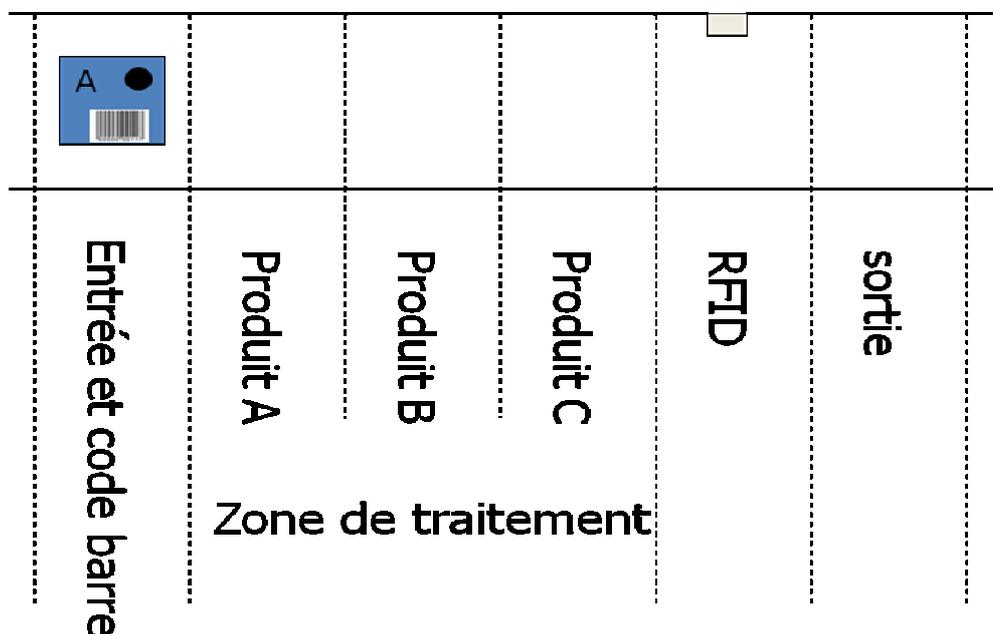


I.4 Proposition d'un scénario

Le but de ce projet est également de créer une maquette qui servira de support pour des Travaux Pratique en automatisation industrielle. De ce fait, il nous a fallu imaginer un fonctionnement que l'on puisse donner au système pour que les étudiant puissent apprendre à utiliser la lecture de code barre et la RFID.

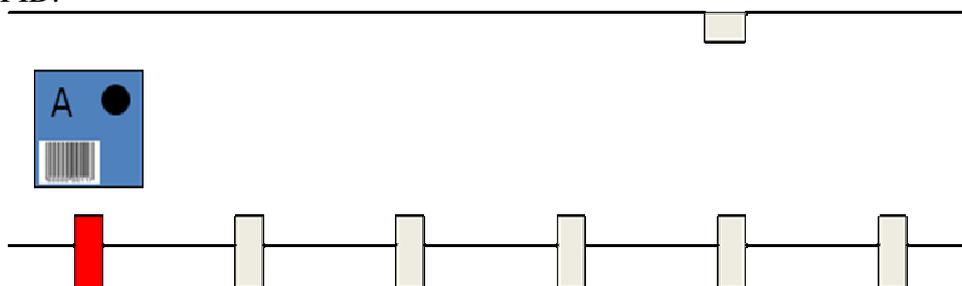
Donc nous avons pensé à un poste de tri qui reçoit trois types de colis différents (Produit A, Produit B et Produit C). Dans un premier temps l'opérateur doit enregistrer le code barre des trois types de produit. Ensuite, en fonction de code barre lu en entrée, le colis ira dans zone de traitement dédiée à son type de produit. Le système RFID écrira dans le tag du produit en cours son heure d'arrivée et son temps de traitement.

I.4.A-Synoptique de la maquette :

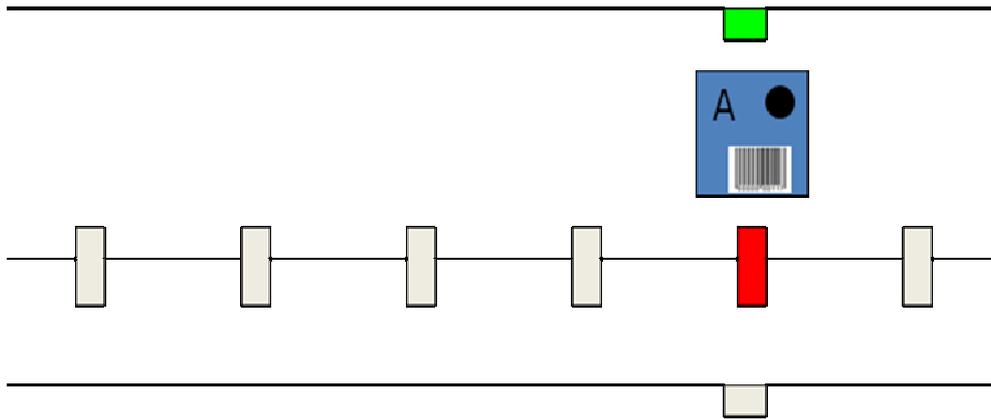


I.4.B-Exemple de fonctionnement avec n produit de type A :

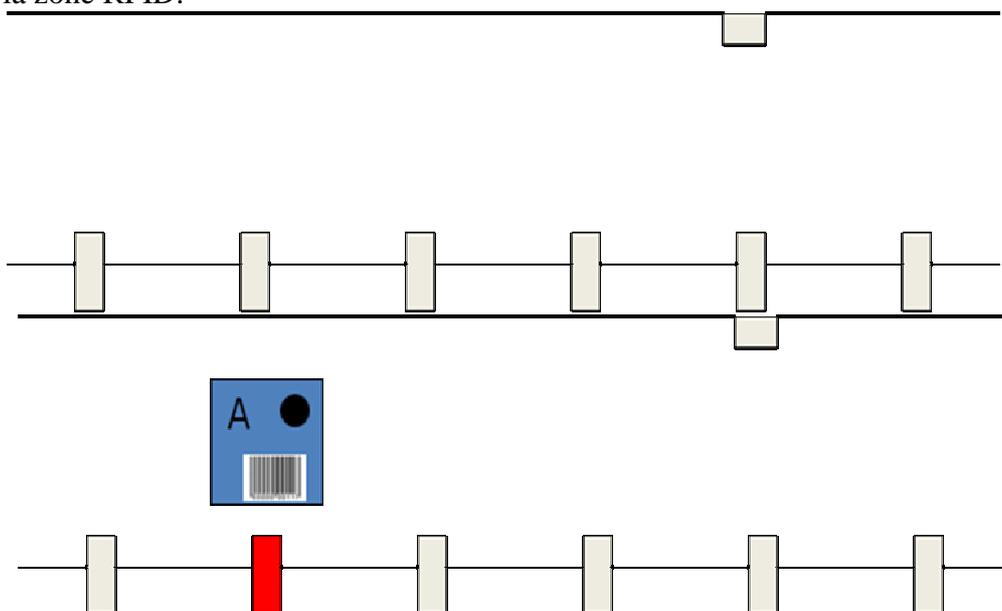
1. L'opérateur présente un colis en entrée et lit capture son code barre avec le lecteur. L'automate enregistre l'heure d'arrivée du colis et le fait avancer jusqu'à la zone RFID.



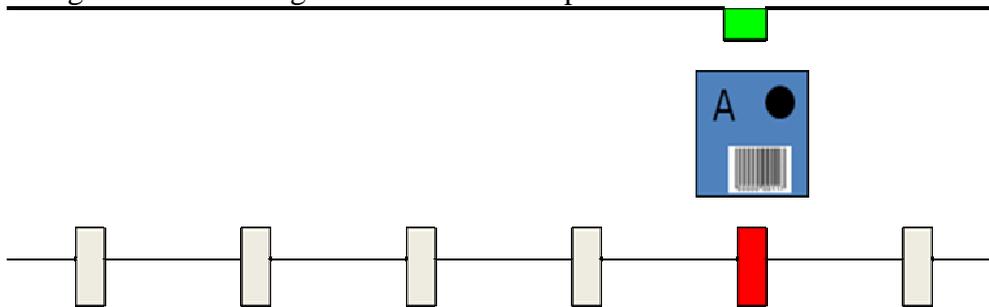
2. La station RFID écrit dans le tag du produit son heure d'arrivée précédemment enregistré dans l'automate. Le colis se rend jusqu'à sa zone de traitement.



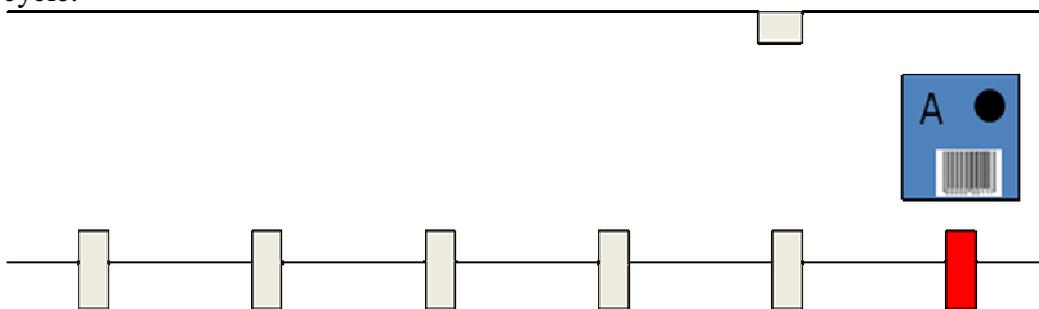
4. L'opérateur retire le colis du tapis et le remet, ce qui permet de simuler un temps de traitement sur le produit qui sera aléatoire. Ensuite le colis se rendra de nouveau dans la zone RFID.



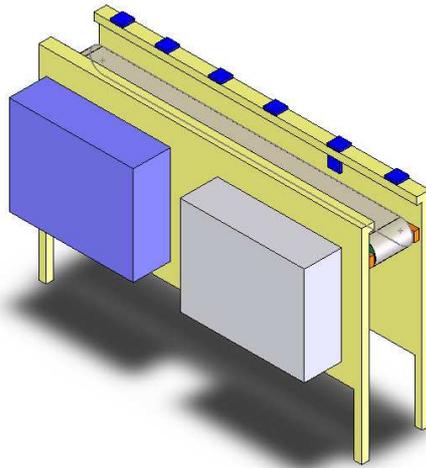
5. La station RFID lit l'heure d'arrivée du colis, l'automate la compare avec son heure courante et en déduit le temps de traitement du produit que la station RFID écrit dans son tag. Le colis se dirige vers la sortie du tapis.



6. Le système attend que le colis soit récupéré par un opérateur avant de relancer un cycle.



II – La réalisation de la maquette



Pour réaliser le fonctionnement décrit dans le chapitre précédent, il a fallu se procurer un certain nombre de matériels :

- le système d'identification RFID de télé-mécanique : Ositrack. Celui-ci est composé d'un transpondeur (il lit et écrit dans les puces RFID), d'une boîte de dérivation, d'un câble qui relie cette boîte à l'automate et d'un câble d'alimentation.
- Le lecteur code barre, qui sera connecté à l'automate via une liaison série.
- 6 détecteurs de proximité qui pourront établir la position du colis sur le tapis.
- 8 relais électriques qui feront l'interface entre les entrées et sorties de l'automate et les actionneurs et détecteurs du système.
- Une boîte de dérivation qui centralisera les commandes des actionneurs et les informations des détecteurs.
- Le convoyeur, sa motorisation et son système de commande.

Dans cette liste nous avons seulement choisi les détecteurs, les relais et la boîte de dérivation car le reste a été fourni par l'ISTIA dès le début du projet. De plus le tapis était déjà équipé de deux capteurs de proximité, donc nous les avons récupérés et nous en avons plus que quatre à commander.

II.1 Choix des composants :

II.1.A-Les détecteurs de proximité

Pour faire notre choix, on a pris en compte la largeur du tapis. Celle-ci n'étant pas grande (une vingtaine de centimètres). Nous avons donc opté pour une détection émetteur/récepteur plus réflecteur (cf figure 1). Ce système est moins fiable et aussi moins chère que le simple ensemble émetteur plus récepteur, autrement dit système à barrage (cf figure 2), mais sa précision est suffisante pour notre application.

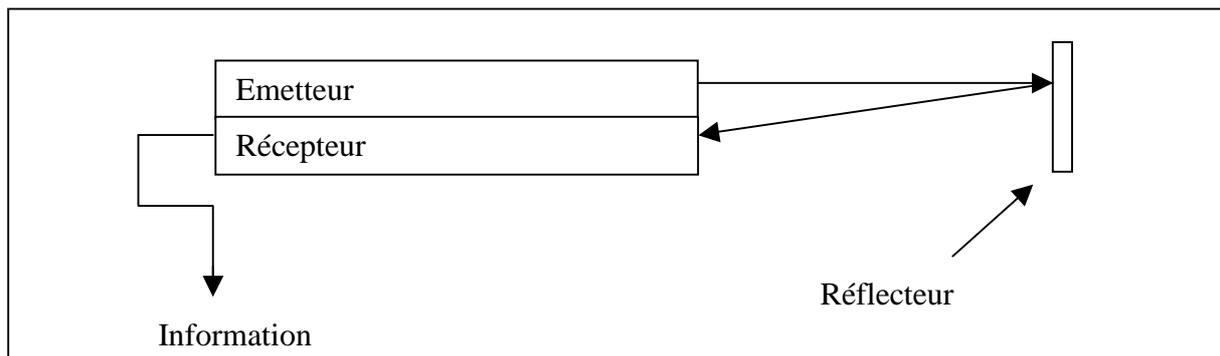


Figure 1

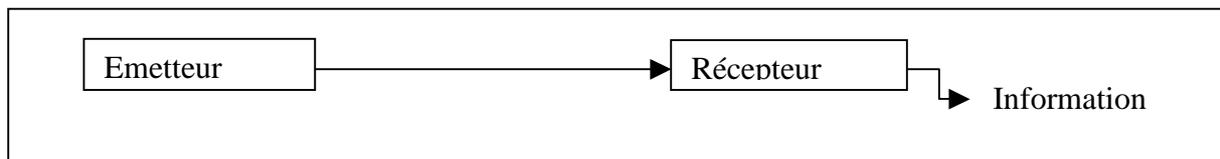
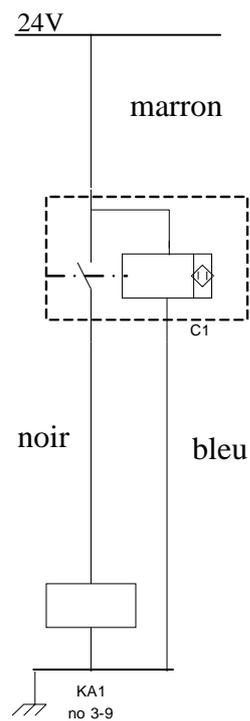


Figure 2

Nous avons donc choisi des détecteurs du constructeur TELEMECANIQUE dont la référence est : XUK1APANL2 de la famille Osiris.



Schémas de câblage des nouveaux détecteurs :



II.1.B-Les relais

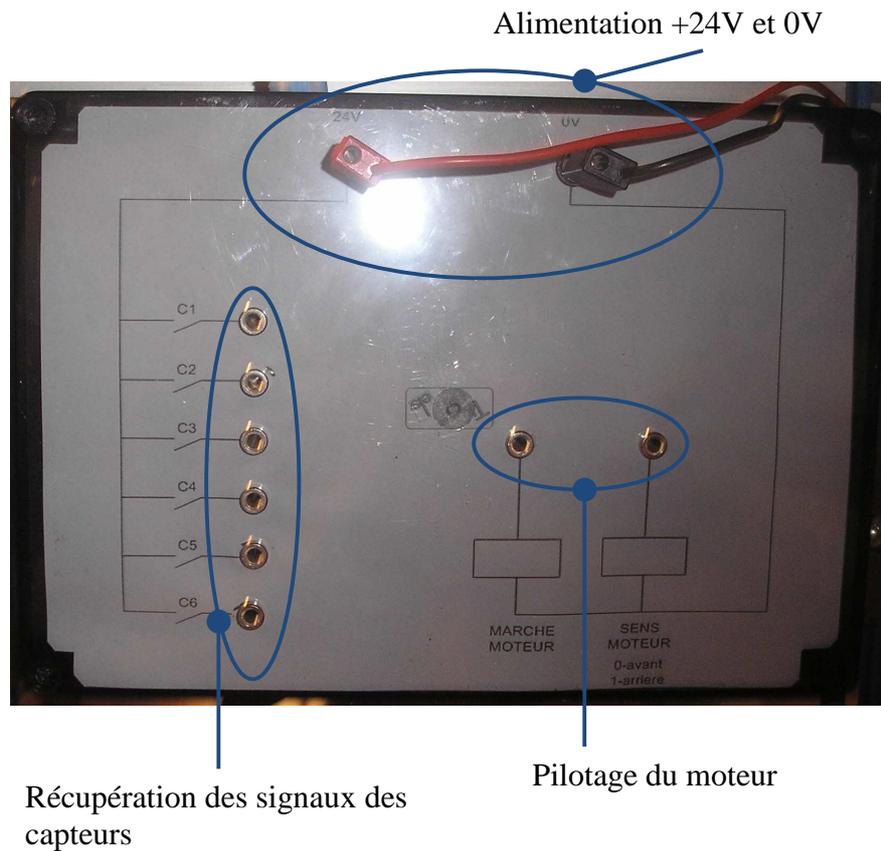
Les relais ont été choisis en fonction de la tension de leurs bobines (24V continu). Cependant, nous n'avons pas pris en compte l'intensité nominale car les relais pilotent des entrées de l'automate et du variateur de vitesse qui ne demande pas de grandes intensités.

Nous avons donc choisi des 55.32 de chez Finder qui peuvent supporter jusqu'à 10 A sur leurs contacts ce qui est largement suffisant pour notre application.



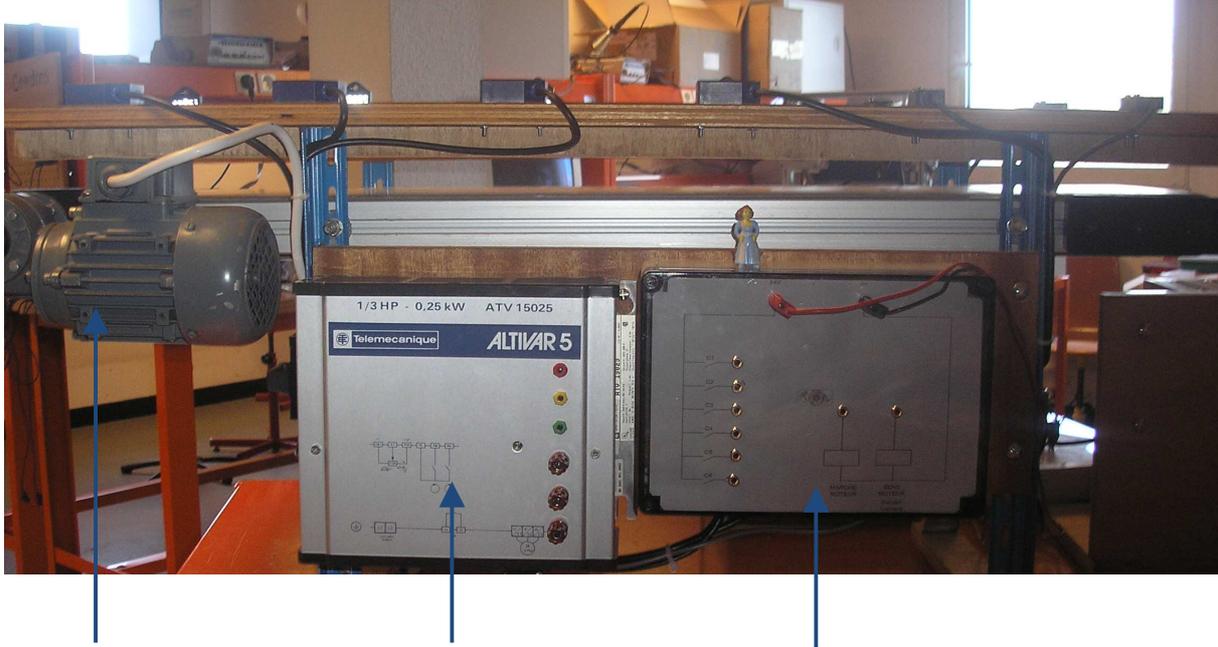
II.1.C-Boite de dérivation

La boîte choisie mesure 178 x 253 x 109mm et possède un couvercle transparent afin que l'on puisse placer un synoptique modélisant toutes les entrées et sorties mises en jeu dans le système.



II.2 Construction de la maquette :

Vue de face sur la maquette:



Variateur de vitesse

Variateur de vitesse

Boite de dérivation

Vue de haut sur la maquette:



Boite de dérivation
RFID

Station RFID

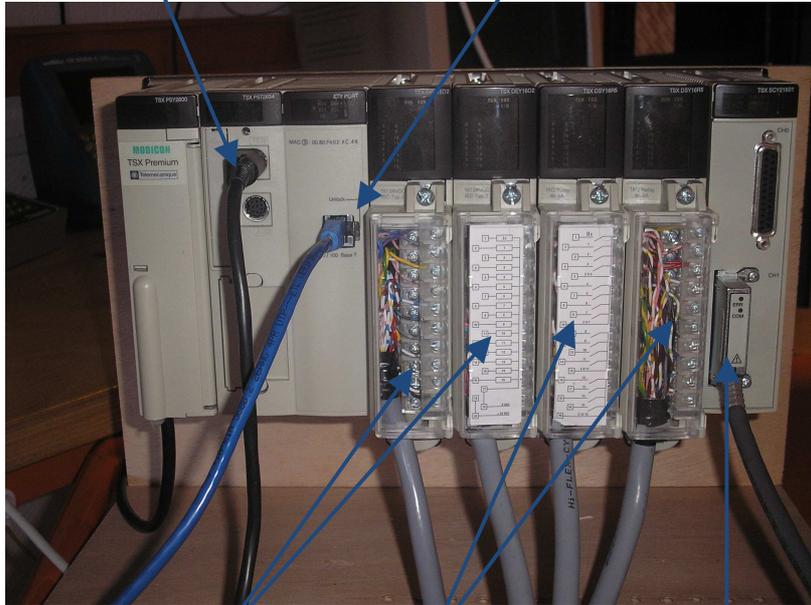
Anciens capteurs

Nouveaux capteurs

Vue de face sur l'automate :

Câble connectant
l'automate à la boîte
de dérivation RFID

Câble connectant
l'automate au PC grâce
à un réseau Ethernet

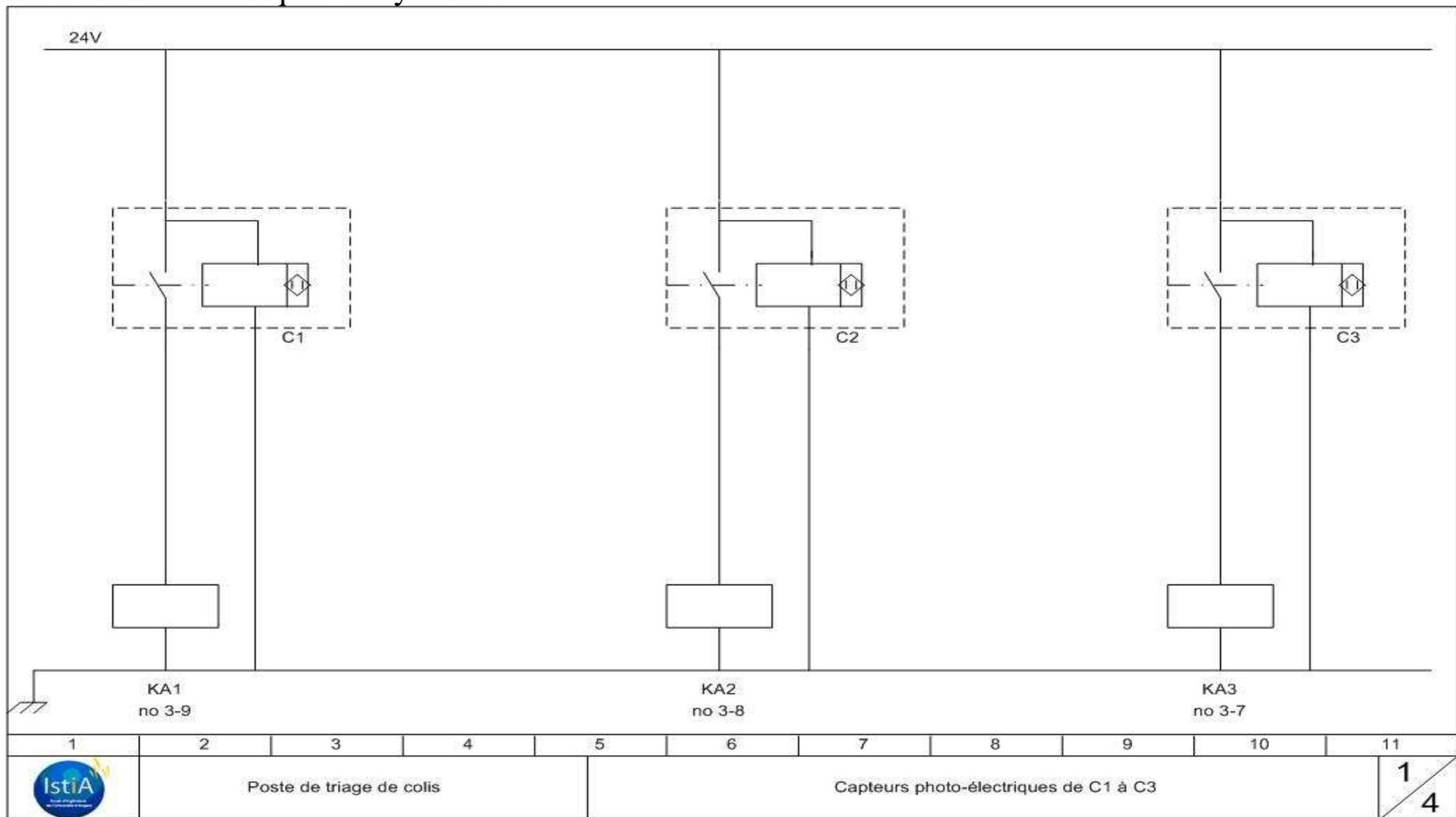


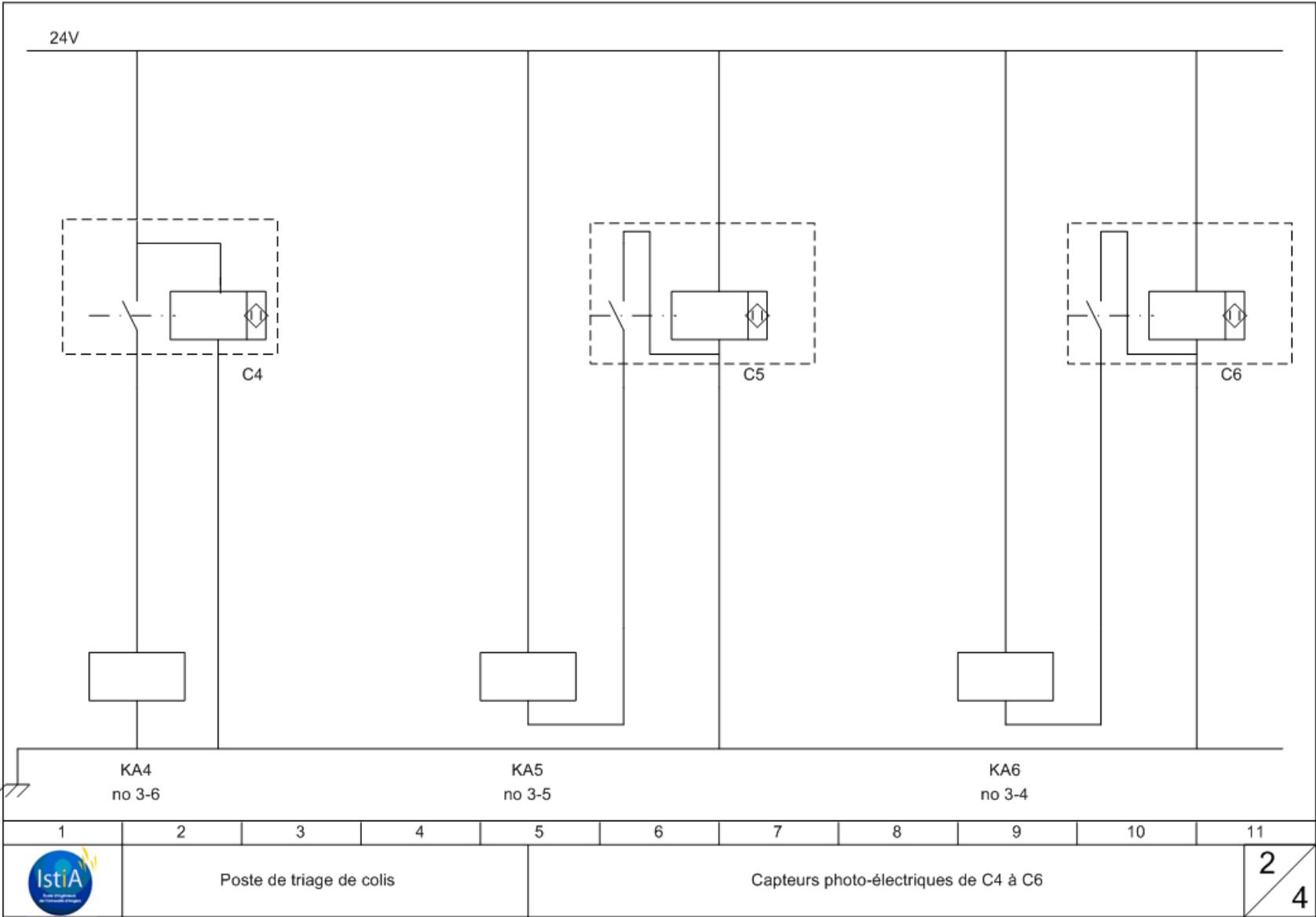
Cartes d'entrées de
l'automate

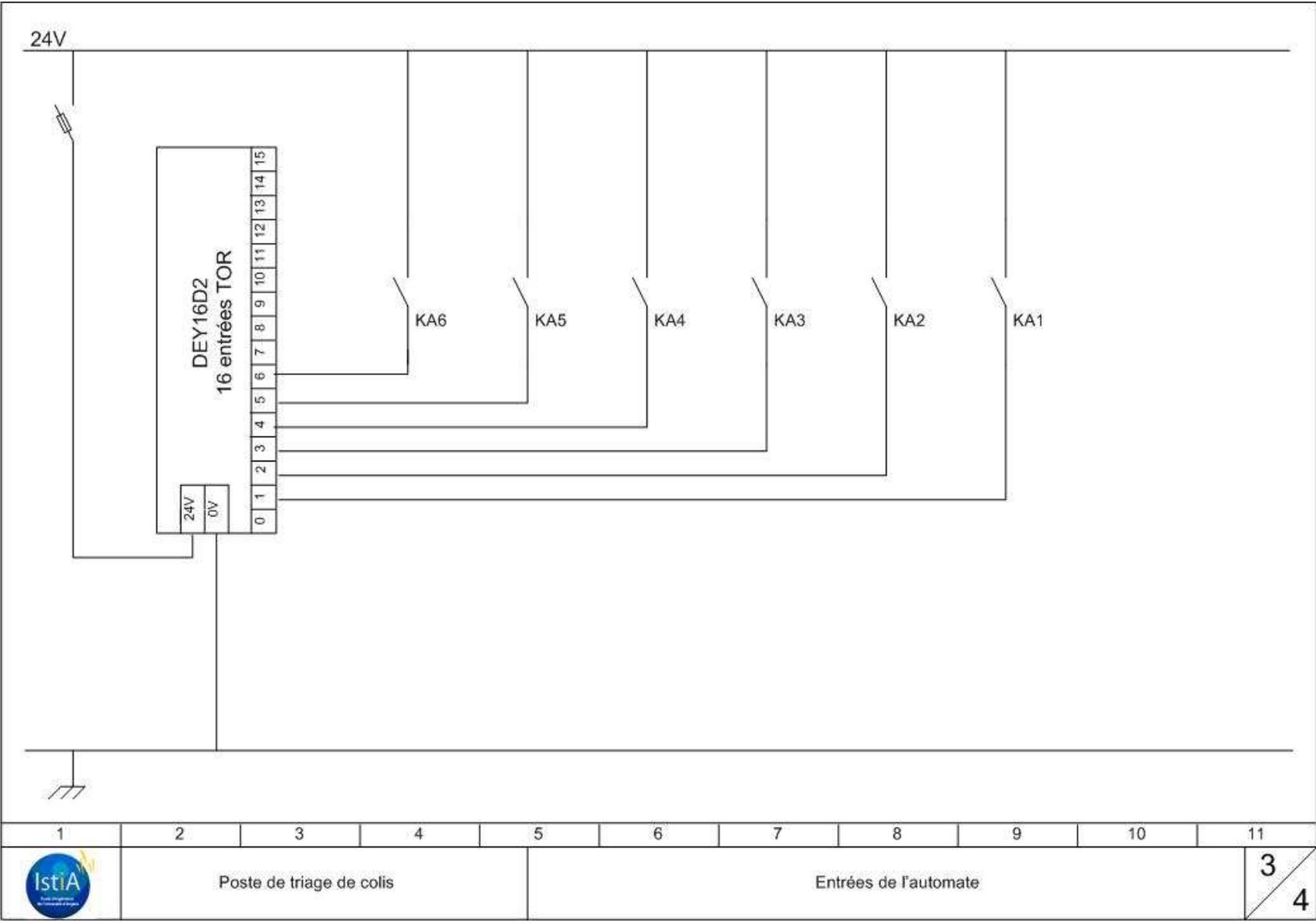
Cartes de sorties de
l'automate

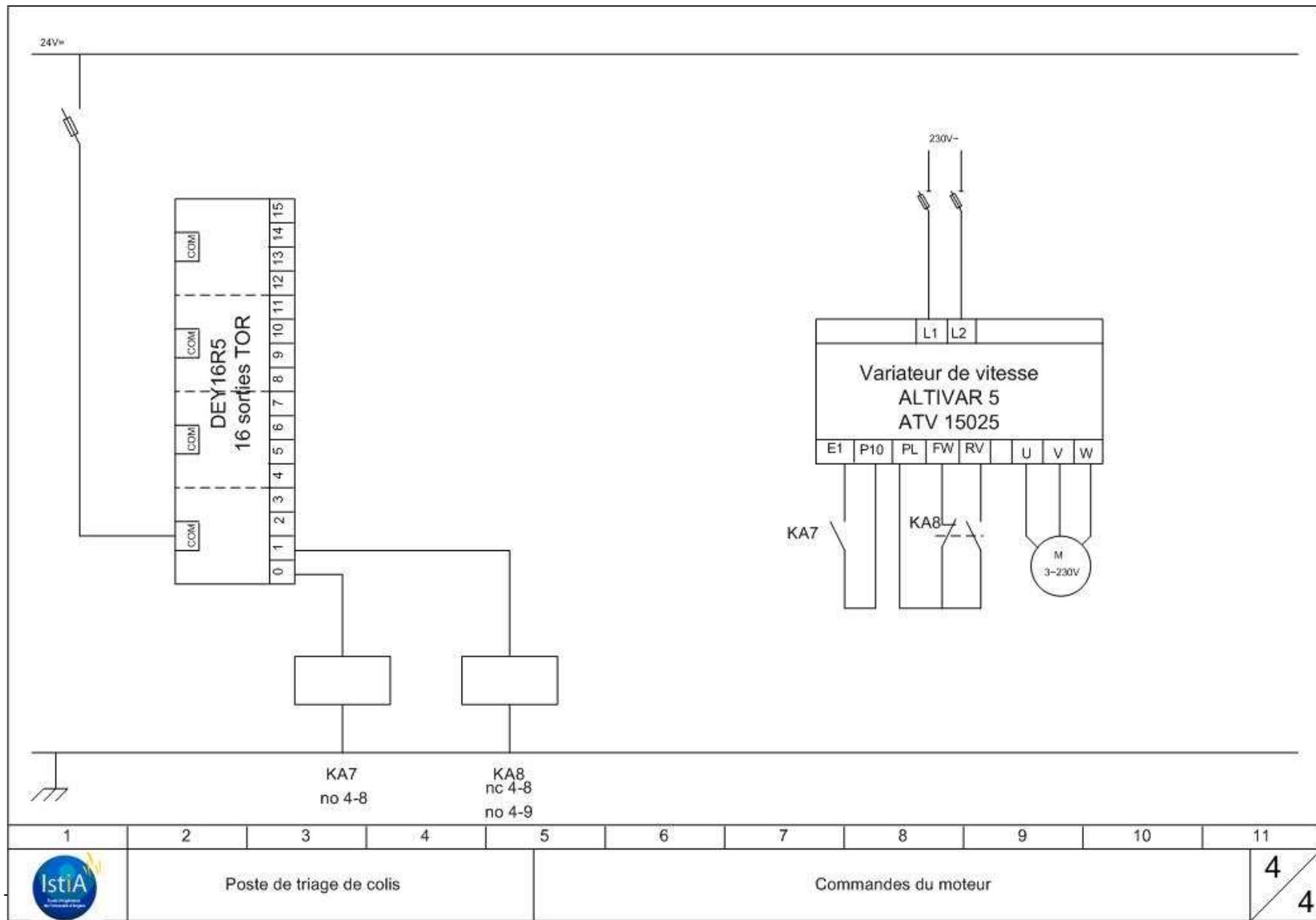
Câble connectant le lecteur de
code barre à l'automate grâce
une liaison série en mode
caractère (le lecteur renvoie
des string à l'automate)

II.3 Schémas électriques du système :









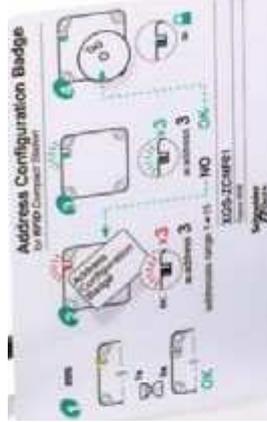
Poste de triage de colis

Commandes du moteur

4
4

II.4 Configuration du transpondeur RFID :

Pour que la maquette soit prête d'utilisation, il ne reste plus qu'à configurer le transpondeur RFID. En effet, celui-ci communiquera avec l'automate grâce à un réseau UNITELWAY, il est donc indispensable de donner sa propre adresse au transpondeur même si dans notre application il est le seul membre de ce réseau. Avec le système Ositrack est fournie une carte de configuration :



Attention cette carte est indispensable, il ne faut surtout pas la perdre.

Démarche à suivre :

- alimentez le système Ositrack et attendez 5 secondes.
- Approchez la carte de configuration du transpondeur, le nombre de clignotement **ROUGE** que son voyant 'tag' fera sera l'adresse que vous lui affecterez. Retirez la carte de configuration.
- Le voyant 'tag' clignote en **VERT** le nombre de fois correspondant à la valeur de l'adresse affectée.
- Approchez un tag quelconque du transpondeur pour valider l'opération, le voyant 'tag' doit s'allumer en **VERT**.

Les numéros d'adresse sont compris entre 1 et 15.

Le voyant 'tag' du transpondeur clignotera régulièrement en vert le nombre de fois correspondant à la valeur de l'adresse affectée.

III – La programmation

III.1 Les nouveautés de Unity

Les concepteurs peuvent utiliser Unity pro comme s'il s'agissait de PI7 pro sans être obligé de connaître les nouveautés. Il s'agit donc d'une continuité dans la série afin de réduire au minimum les possibles coût de formation.

Ces nouveautés nous le verrons sont plus axées sur la programmation en elle-même et elle resserre le fossé qu'il y a entre un programmeur d'automate et un programmeur informatique. Ce resserrement est significatif notamment via l'introduction d'objet dans le logiciel.

III.1.A - Les particularités des variables

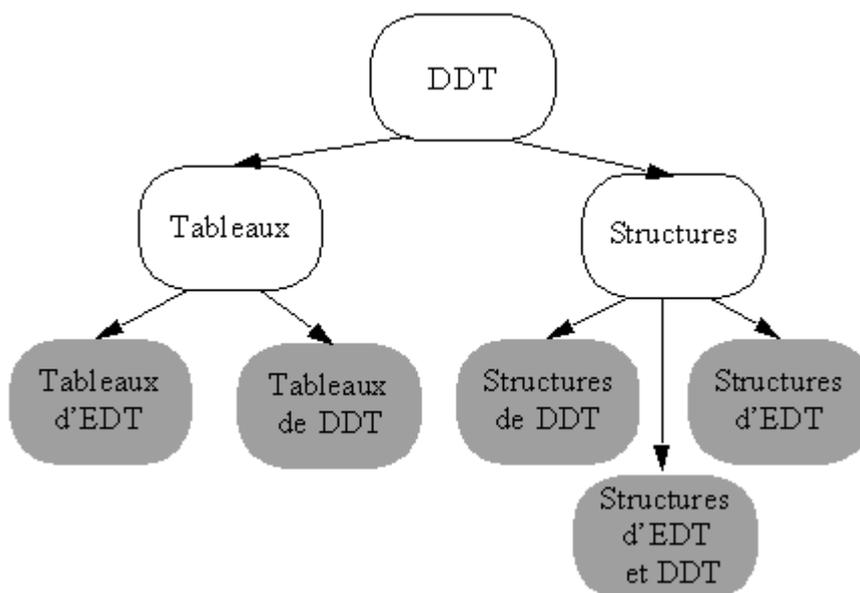
Les types de variables déjà existantes sont classés dans les EDT (type de données élémentaires)

- Binaires : Booléen, entier et heure
- BCD : Date, TOD(Time Of Date) et DT (Date and Time)
- les réels : Real
- les chaînes de caractères : String
- les chaînes de bits : Byte, Word, Dword

Les nouveaux types de données dérivés sont classés dans les DDT(Derived Data Types) :

- les tableaux
- Les structures

On peut schématiser l'utilisation de ces types dérivés de la façon suivante :



On peut ainsi voir que l'on peut réaliser des données structurées imbriquées : des structures de structures, des structures de tableaux, des tableaux de tableaux, des tableaux de structures...

Les tableaux : élément de données incluant un ensemble de données de type identique

Elle se définit de la façon suivante : ARRAY[dimensions] OF type

| | | | | |
|-------------|--------------------|--|--|--|
| [-] tablire | ARRAY[0..3] OF INT | | | |
| [-] tabl... | INT | | | |
| [-] tabl... | INT | | | |
| [-] tabl... | INT | | | |
| [-] tabl... | INT | | | |

Dans cet exemple, il s'agit d'un tableau de 4 entiers.

Les tableaux à plusieurs dimensions sont définis comme suit :

Tab_3 : ARRAY[1..10, 1..20] OF INT qui est un tableaux à 2 dimensions contenant 10 x 20 nombres entiers

Tab_4 : ARRAY[0..2, -1..1, 201..300, 0..1] OF REAL qui est un tableaux à 4 dimensions contenant 3 x 3 x 100 x 2 nombres réels

Une structure : C'est une donnée qui contient un ensemble de données de type différent.

Définition d'une structure :

```
Ident ; nom du nouveau type
    Nom : STRING[12]
    Prenom : STRING[16]
    Age : UINT
```

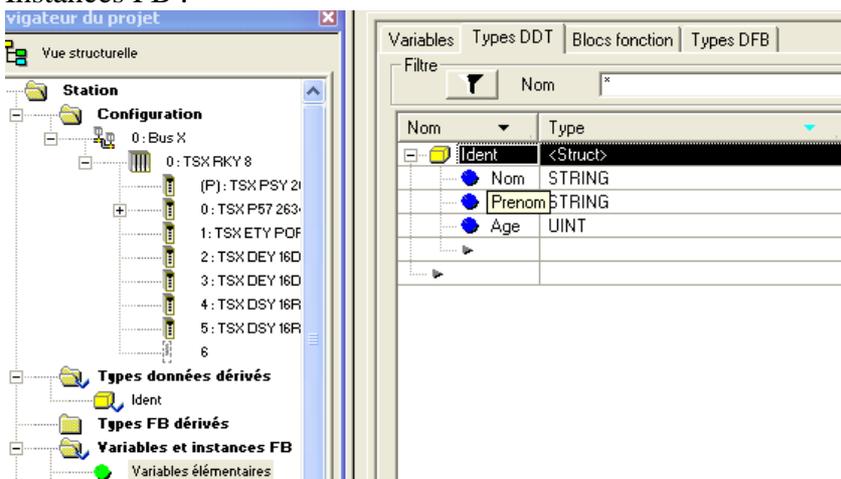
Définition d'une Personne de type Ident :

Personne : Ident

Accès à la structure :

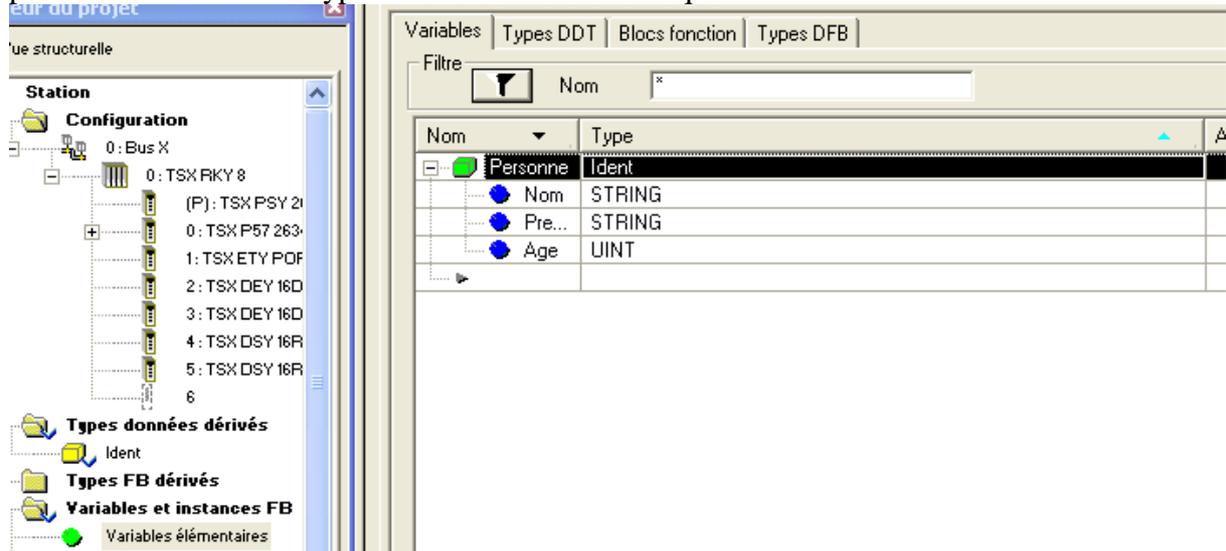
Personne.Nom

Une autre possibilité est de déclarer la structure dans l'onglet « Types DDT » de Variable et Instances FB :



On fait un clic droit sur le nom de la nouvelle structure et on la copie dans la bibliothèque.

Enfin, pour déclarer la variable, on insère le nom et on va dans le sélecteur de type de donnée pour sélectionner notre type dans la bonne bibliothèque :



Autre particularité : les données peuvent être symbolique : définis sans adresse :



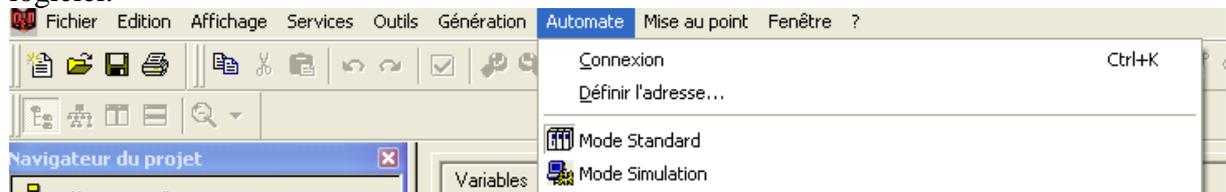
Ces données non localisées permettent de supprimer le mapping mémoire.

III.1.B – Les innovations dans la conception

Le simulateur

Il existe un simulateur :

Pour l'actionner, il suffit de sélectionner le mode Simulation dans l'onglet « Automate » du logiciel.



Il s'agit de la même démarche que pour l'automate : sauf qu'il faut le mettre en mode simulation.

En exécution, on peut forcer les valeurs d'entrée avec une table d'animation.

Les sections

La conception est plus claire : il peut exister plusieurs sections pour les graficets :



Il y a également un gain de temps lors de la programmation et en particulierité des tests puisqu'il y a maintenant possibilité de faire une modification en ligne des variables et des sections programmes sans transfert de PC à l'automate.

III.1.C - Les logiciels spécialisés

- de nouveaux outils de mise au point qui permettent l'accroissement de la qualité en phases de tests et de mise en service.
- une traçabilité des actions opérateurs.
- la recherche de causes de défauts.
- l'intégration plus aisée d'outils tiers comme les hyperliens dans le projet, l'import/export XML sur tout élément du projet et un accès à la base de données et au serveur Unity Pro.

Les concepteurs de télémécanique accentuent leur orientation sur un rapprochement entre l'automatisme et l'informatique grâce à des logiciels avancés de Unity Pro :

- Unity Studio structure les applications distribuées qui communiquent entre eux via internet.
- Unity Application Generator est spécialisé pour les applications de régulation, plus particulièrement pour les contrôles continus dans les industries batch/process.
- Unity Developer's Edition permet l'accès à partir d'application VBA, VB ou C++ aux serveurs Unity.
- Unity EFB Toolkit développe en C des fonctions personnalisées pour qu'elles soient intégrées dans les bibliothèques de Unity Pro.
- Unity SFC permet la surveillance des applications en SFC à partir de station de dialogue homme/machine IHM.

III.2 Les blocs fonctions utilisées

Nous présentons, dans cette partie, les blocs fonctions utilisées pour répondre au besoin que nous avons. Les premiers développés touchent à la question du temps. Il faut, en effet, enregistrer dans la carte RFID l'heure d'arrivée du produit sur le poste de tri. Ensuite, après le traitement, le temps écoulé depuis l'arrivée de l'objet doit être aussi inscrit dans la carte.

Selon le scénario voulu, à l'arrivée du produit sur le tapis, on lit le code barre pour identifier le type de produit. Les deuxièmes types de blocs fonctions présentés seront donc pour la gestion du code barre.

Pour manipuler les temps de traitement présents sur les cartes RFID, le troisième type de blocs concerne la lecture/écriture des cartes RFID présentes sur les produits.

III.2.A – Les blocs pour la gestion du temps

Mise à l'heure de l'automate par le PC

Pour rafraîchir l'heure de l'automate, on se dirige vers la fenêtre schématisant le rack, on double clic sur le processeur. Une fenêtre s'ouvre : dans la partie « Animation », on trouve l'onglet « Horodateur ». Une fonction propose de mettre à jour l'automate à partir de la date du PC.

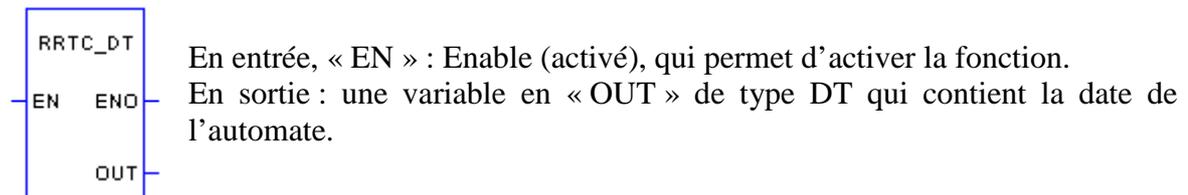
Choix du type des variables

Quatre types de variables nous sont proposés pour intégrer la notion de temps dans le projet : le type Time, le type DT (Date and Time), le type Date et le type TOD (Time Of Day) :

- Le type Date, codé sur 32 bits contient les informations suivantes : l'année codée sur 16 bits, le mois codé sur 8 bits et le jour codé sur 8 bits. Souhaitant disposer d'une durée qui aura pour grandeur la seconde, voir la minute, on peut écarter ce type pour notre projet.
- Le type TOD, au contraire, nous informe sur l'heure, la minute et la seconde : codée sur 32 bits, elle est composée de 8 bits pour l'heure, 8 bits pour les minutes, 8 bits pour les secondes et les 8 derniers bits sont inutilisés. Aucun bloc contenant ce type n'est intéressant à manipuler pour récupérer l'heure courante donc nous écartons TOD.
- Le type TIME est utilisé juste pour exprimer une durée : exprimé en milliseconde, elle représente environ la durée maximale de 49 jours. Ce type aurait pu être utilisé (manipulant une durée) mais problème pour manier de tel variable avec les cartes RFID (cf. plus loin).
- Le type DT (Date and Time) contient toutes les données relatives au temps : codées sur 64 bits, elle possède 16 bits pour l'année, 8 bits pour le mois, 8 bits pour le jour, 8 bits pour l'heure, 8 bits pour les minutes et 8 bits pour les secondes. Ce dernier type

possède beaucoup de données facilement vérifiables et surtout il est aisé de l'extraire de l'automate.

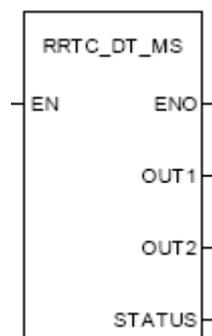
Nous voulions donc utiliser le bloc suivant « RRTC_DT » permettant d'obtenir la date de l'automate au format DT :



Ce format était pratique. Nous avons même réussi à soustraire 2 variables de type DT avec le bloc « SUB_DT_DT » (pour calculer temps de traitement). Le problème est arrivé quand il a fallu écrire ces variables dans les cartes RFID. En effet, on écrit dans les cartes par mot de 16 bits et ici il était difficile de séparer une variable de 64 bits en mots de 16 bits.

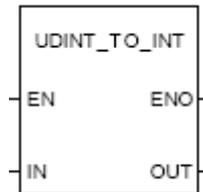
La durée de traitement d'un produit est de l'ordre de quelques secondes voir de quelques minutes. Nous avons donc trouvé un autre bloc dérivé de « RRTC_DT »: « RRTC_DT_MS » qui nous donne aussi une durée mais pas au format DT, au format UDINT. Ce format double entier de 32 bits est plus facile à manipuler puisqu'il suffit de le séparer en deux pour obtenir 2 mots de 16 bits écrivables dans les cartes RFID.

Extraire les données temporelles de l'automates dans une variables de type UDINT :



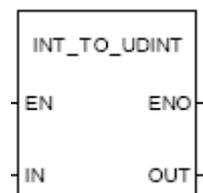
Ainsi, pour extraire la date de l'automate, on utilise la fonction « RRTC_DT_MS » :

- En entrée, EN : Enable (activé), qui permet d'activer la fonction.
- En sortie 1 : une variable en « OUT 1 » de type Display_NTTPC (structure) qui contient la date au format BCD et le nombre de milisecondes écoulées depuis 1980.
- En sortie 2 : une variable en « OUT 2 » de type Calc_NTTPC (structure) qui contient le nombre de secondes écoulées depuis 1980. C'est ce nombre (Seconds : élément de la structure) qu'on utilise pour le temps de traitement.

Séparer la variable UDINT 32 bits en deux mots de 16 bits :

Le bloc UDINT_TO_INT élimine les 16 bits de poids forts de la variable UDINT(32 bits) en entrée pour aboutir à une variable de 16 bits :

- L'entrée EN active la fonction UDINT_TO_INT.
- L'entrée IN reçoit la variable UDINT à transformer.
- La sortie OUT rend une variable INT.

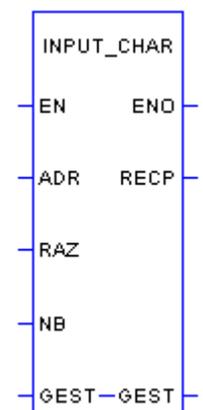
Transformer les mots lus sur la carte RFID au format UDINT :

Ce bloc fonctionne à l'inverse du précédent : il transforme une variable de 16 bits en entrée (IN) en une variable de 32 bits (UDINT) en sortie (OUT). Les 16 nouveaux bits de poids forts sont nuls.

Le temps de traitement inscrit dans la carte est séparé en deux mots de 16 bits, eux-mêmes transformés en deux variables de 32 bits grâce à cette fonction. Nous avons donc les poids faibles et forts du temps dans deux variables différentes. Cette « formule » permet de les assembler (HeureSortie.Seconds est le nombre de secondes courant, à la fin du traitement, écoulé depuis 1980 (bloc RRTC_DT_MS)) :

TempsTraitement := HeureSortie.Seconds - (PFaibleHeureLue + (PFortHeureLue*65536));

III.2.B – Les blocs pour les code-barres

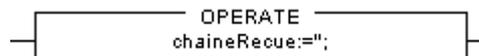


La fonction « INPUT_CHAR » est utilisée pour envoyer une requête de lecture de chaîne de caractères vers un module de communication en mode caractère. Description du bloc :

- L'adresse du module à distance est : « ADDR('0.6.1.SYS') ». Elle se décompose comme suit : le premier terme (0) pour le numéro du rack, le deuxième terme (6) pour le numéro du module, le troisième terme (1) pour le numéro de la voie et le dernier (SYS) pour commander tout objet sur cette voie (1).
- L'entrée RAZ=1 permet de réinitialiser le buffer de réception. Il ne fait rien s'il est à 0.
- NB permet d'informer du nombre de caractère à lire. Si sa valeur est à zéro, la condition d'arrêt est un caractère fin de chaîne.
- GEST est un tableau de gestion : il informe de la situation de l'échange. Il s'agit d'un tableau de quatre entiers. L'utilisateur informe du troisième entier qui est le temps d'attente maximum (0 pour l'infini) et du quatrième entier qui est le nombre de caractère à envoyer. Au retour, l'automate indique via le premier entier un numéro d'échange et un bit d'activité et via le second entier un compte rendu d'opération et de communication qui informent du résultat de l'opération. La signification des valeurs des différents entiers est accessible par l'aide du logiciel Unity.
- En sortie, RECP est une chaîne de caractère qui reçoit ici le code barre.

Il peut exister un problème de chevauchement : par exemple, après avoir lu un long code barre, on fait la lecture d'un petit code barre. Ce dernier sera falsifié car la lecture ne remplace que le strict minimum, ainsi, le petit code barre comportera la fin du long code barre.

Pour y remédier, on remet à zéro la chaîne de caractère lorsqu'on ne l'utilise plus :



III.2.C – Les blocs pour la lecture-écriture du RFID

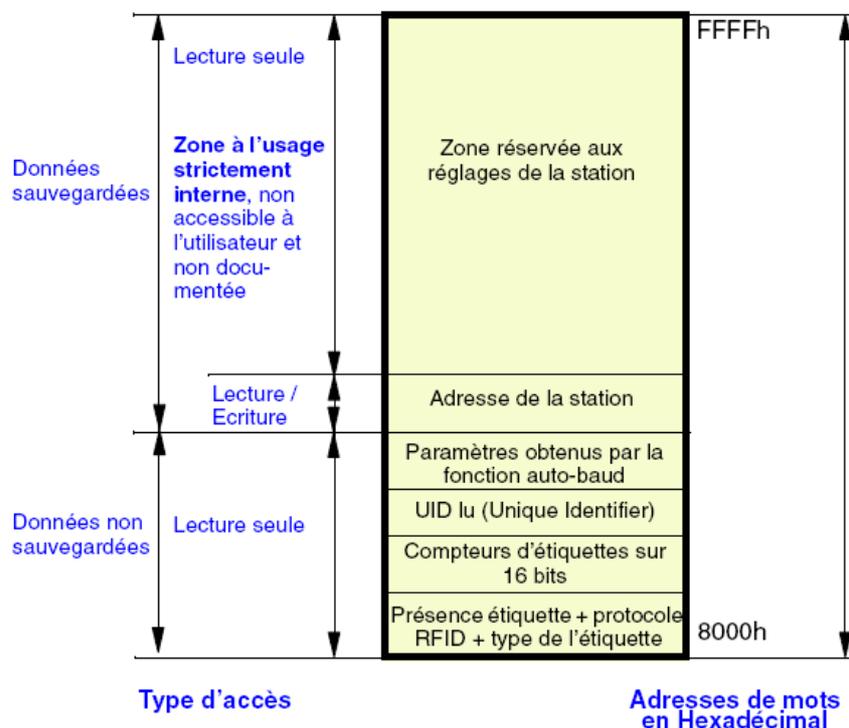
Petite présentation d'une carte RFID (manipulation) :

On accède aux données d'une carte RFID. La zone mémoire adressage est découpée en deux zones : la zone mémoire étiquette et la zone mémoire station. Définition des zones d'adresses de mots utilisées :



Plus précisément chaque zone est définie ainsi :

1) La zone mémoire Station :



Avec l'aide d'un tableau descriptif, on en déduit que pour savoir si l'étiquette est présente devant le transpondeur RFID, nous devons lire le premier bit de la zone mémoire Station. Ce bit présence étiquette est donc le premier bit du mot %MW32768.

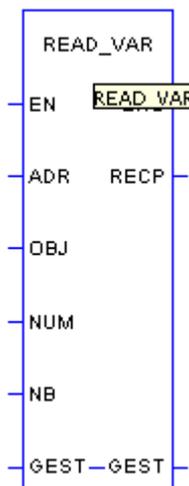
2) La zone mémoire Etiquette :

| Types d'étiquettes | XGH B90E340 | | XGH B221346 | | XGH B320345 | | XGH B211345 | | XGH B444345 | | XGH B445345 | | |
|--------------------|-------------|------|-------------|------|-------------|------|-------------|------|-------------|------|--------------|------|---|
| | 256 octets | | 256 octets | | 112 octets | | 256 octets | | 3408 octets | | 13632 octets | | |
| Adresses | Dec. | Hex. | Dec. | Hex. | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | à | à | à | à | à | à | à | à | à | à | à | à | à |
| | 127 | 7F | 127 | 7F | 55 | 37 | 127 | 7F | à | à | à | à | à |
| | (1) | (1) | (1) | (1) | (1) | (1) | (1) | | | | | | |
| | | | | | | | | | 1703 | 6A7 | | | |
| | | | | | | | | (2) | | | | | |
| | | | | | | | | | | | 6815 | 1A9F | |
| | | | | | | | | | | | (2) | | |

Ce tableau nous montre que chaque type d'étiquette possède une plage de mémoire différente. Ces plages allant de 112 octets à 13632 octets sont largement suffisantes dans notre application. Nous devons donc écrire ou lire les données de la carte à partir du mot %MW0.

Nous devons écrire des données sur les cartes RFID. Pour réaliser ceci nous avons le choix entre des blocs WRITE/READ VAR et des blocs REQUEST. Dans un souci de facilité d'utilisation nous choisissons les blocs READ_VAR et WRITE_VAR :

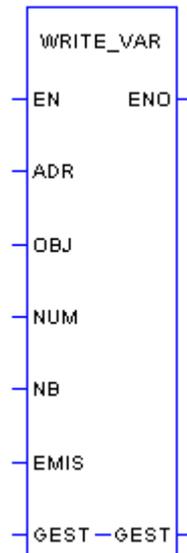
Manipulation de la fonction READ_VAR :



La fonction « READ_VAR » est utilisée pour envoyer une requête de lecture d'objets (mots ou bits) vers un module de communication à distance. Description du bloc :

- L'adresse du module à distance est : « ADDR('0.0.0.3') ». Elle se décompose comme pour le bloc INPUT_CHAR : le premier terme (0) pour le numéro du rack, le deuxième terme (0) pour le numéro du module, le troisième terme (0) pour le numéro de la voie et le dernier (3) correspond à l'adresse de l'objet à distance. L'adresse de l'objet, dans notre cas, est l'adresse du transpondeur RFID de lecture/écriture configurable. On configure l'adresse 3 par la méthode décrite précédemment dans la partie « Réalisation de la maquette ».
- L'entrée OBJ configure le type des variables à lire et donc le format des valeurs reçues. Pour notre application on lit des mots donc OBJ='%MW'.
- L'entrée NUM correspond au numéro du premier objet lu. NUM=0 pour la lecture des temps de traitement. NUM=32768 pour savoir si une étiquette est présente devant le transpondeur RFID.
- NB permet d'informer du nombre d'objets à lire à partir de NUM. NB=2 pour la lecture des deux mots de 16 bits contenant les poids faibles et forts du temps de traitement (UDINT). NB=1 pour la lecture du mot %MW32768.
- GEST est un tableau de gestion : même utilité que pour le bloc INPUT_CHAR.
- En sortie, RECP est un tableau de NB objets de type OBJ.

Manipulation de la fonction WRITE_VAR :



La fonction « WRITE_VAR » est utilisée pour envoyer une requête d'écriture d'objets (mots ou bits) vers un module de communication à distance. Description du bloc :

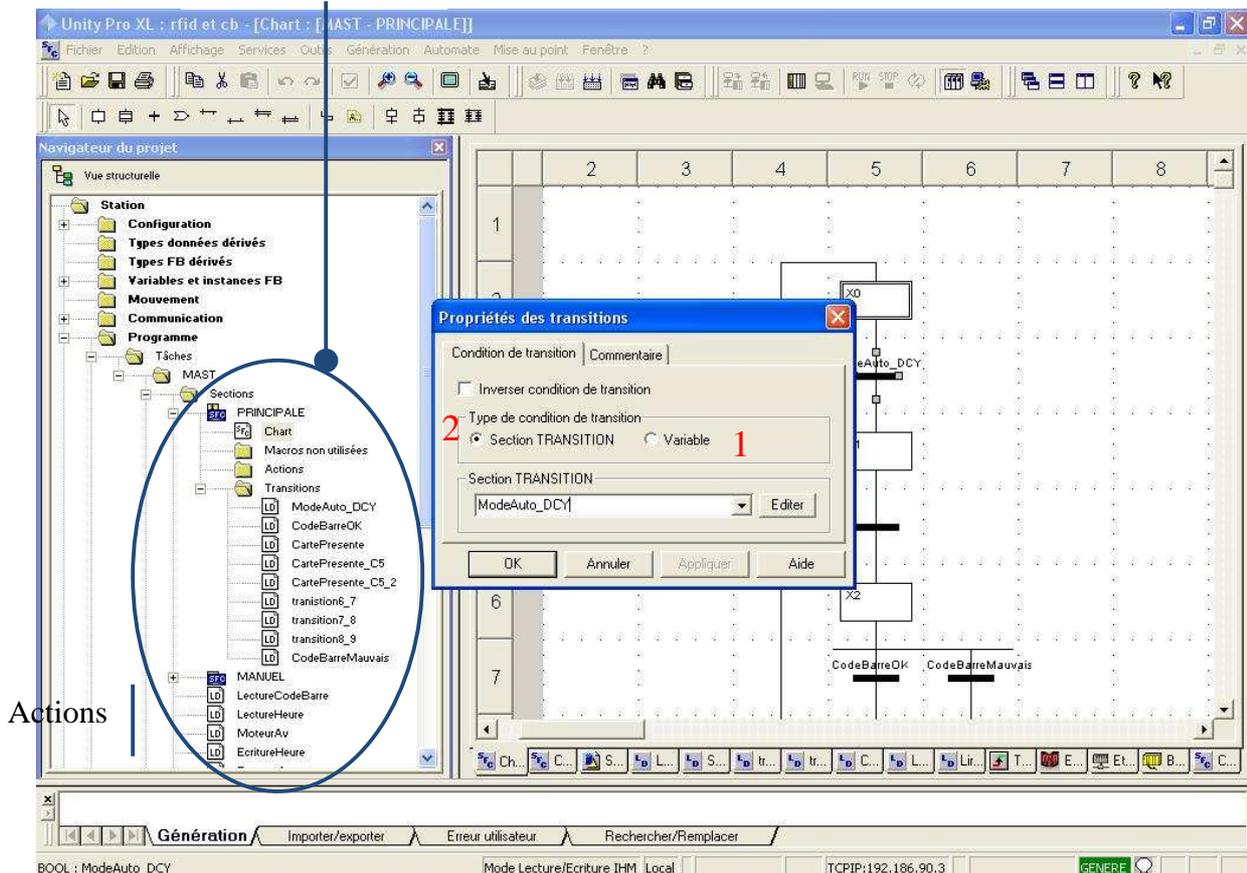
- L'adresse du module à distance est : « ADDR('0.0.0.3') ». Elle se décompose de la même manière que pour la lecture de données à distance.
- L'entrée OBJ configure le type des variables à écrire et donc leur format. Pour notre application on écrit des mots donc OBJ='%MW'.
- L'entrée NUM correspond au numéro du premier objet écrit. NUM=0 car on écrit à partir du premier mot de la zone mémoire de la carte.
- NB permet d'informer du nombre d'objets de type OBJ à écrire à partir de NUM. NB=2 pour écrire les 16 bits de poids faible puis les 16 bits de poids forts de l'heure courante (plus précisément, au moment de l'arrivée du produit sur le tapis, la durée écoulé en secondes depuis 1980).
- En entrée, EMIS est un tableau contenant les données à écrire. Il doit être créé en concordance avec les valeurs du nombre NB et du type OBJ.
- GEST est un tableau de gestion : même utilité que pour le bloc précédent.

III.3 L'application complète de gestion du poste de tri

III.3.A – Les parties du programme

Pour réaliser le programme de l'automate nous nous sommes donc servi des blocs fonctions précédents. Conformément au scénario voulu, un grafcet a été mis en place sous Unity Pro. Présentons d'abord les parties de notre programme :

Grafcet principal



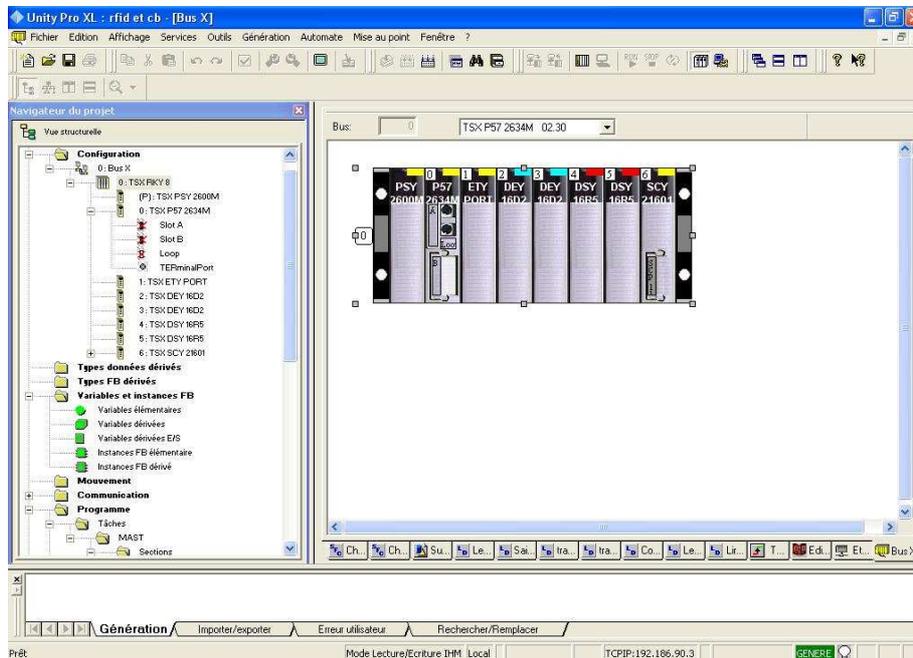
Le grafcet principal correspond au scénario mis en place précédemment. Il comporte 3 parties :

- une partie « Chart » avec le grafcet lui-même. Les transitions peuvent être, comme on le voit ci-dessus, soit une variable unique (1), soit une section Transition à éditer (2).
- on retrouve dans la deuxième partie « Transitions », les sections créés. On peut y mettre plusieurs conditions.
- La troisième partie est les actions : l'activation des étapes du grafcet active ou désactive des bits correspondants (X4.x=1 par exemple quand Etape 4 active). Ceci nous permet de lancer des actions.

La deuxième partie du programme automate est le grafcet secondaire de saisie de code barre. En effet, le départ d'une pièce, sur le poste de traitement, ne peut se faire que si les code-barres des 3 types de produit ont été saisis par le grafcet secondaire.

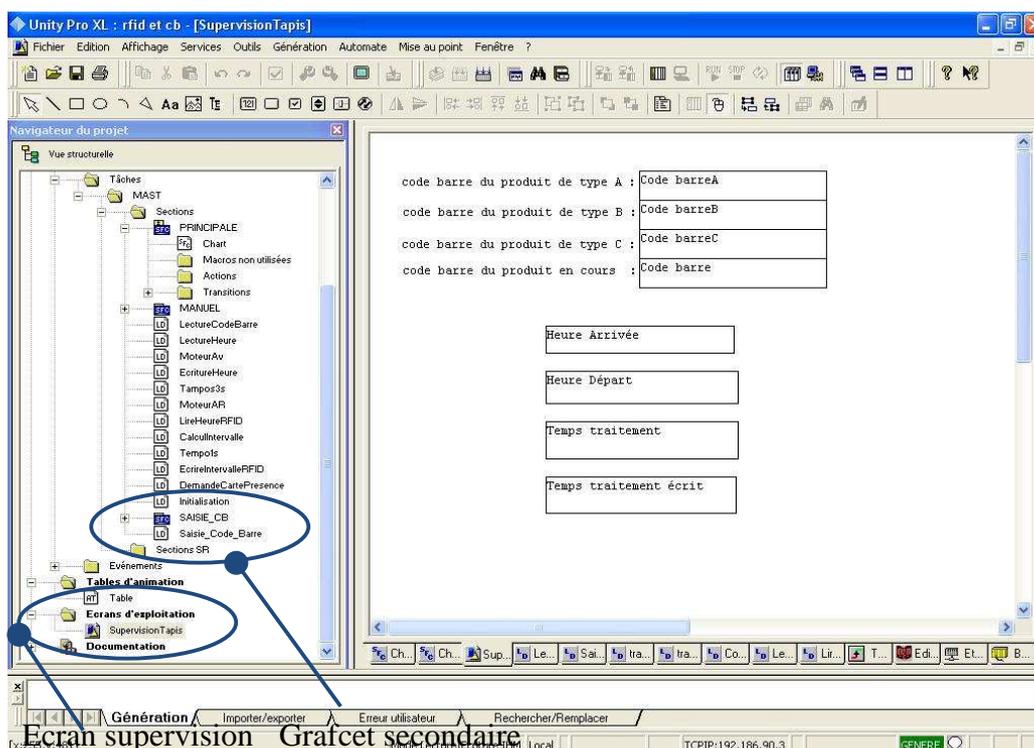
III.3.B – Utilisation de notre programme sous Unity

Configuration des racks de l'automate, avec de gauche à droite : l'alimentation, le processeur (ports TER), le port Ethernet, les cartes d'entrées, les cartes de sorties et le module de communication.



N.B. : Nos variables élémentaires sont dans la partie « Variables et instances FB »

Une supervision simple nous permet aussi de tester rapidement différentes variables de notre projet :

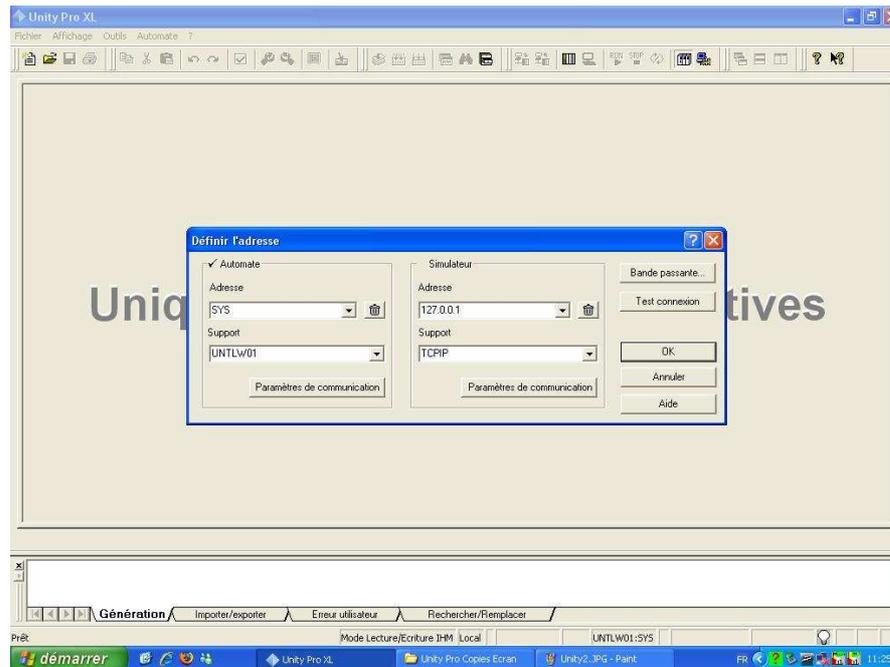


Ecran supervision Grafcet secondaire

III.3.C – Procédures et aides pour utiliser Unity Pro avec notre projet

Procédure de connexion du PC à l'automate :

- 1) Il faut d'abord vérifier que le logiciel n'est pas en Mode Simulation mais bien en Mode Standard.
- 2) Définition de l'adresse de l'automate : menu Automate / Définir l'adresse :



Deux possibilités de configuration d'adresse s'offre à vous. En effet, deux systèmes de communication sont possibles :

- Par câble USB TSXCUSB485 par support UNITELWAY :
 - Adresse = SYS
 - Support = UNITLW01
- Par câble réseau croisé en liaison Ethernet :
 - Adresse = 192.186.90.3
 - Support = UNITLW01

Il est conseillé d'utiliser la connexion Ethernet avec la communication entre le transpondeur RFID et l'automate. En effet, Le transpondeur RFID est branché sur un port TER de l'automate tout comme le câble USB. Il y a pourtant deux ports TER (TER et AUX) mais lorsque les deux sont branchés (vers RFID et vers PC) la communication automate-RFID ne fonctionne pas.

3) Quand l'adresse est configuré, il suffit ensuite de faire : Menu Automate / Connexion --> La communication doit être bonne.

Procédure d'installation du câble USB TSXCUSB485 :

CDROM utile : le CD d'installation de drivers « CD Driver Pack »

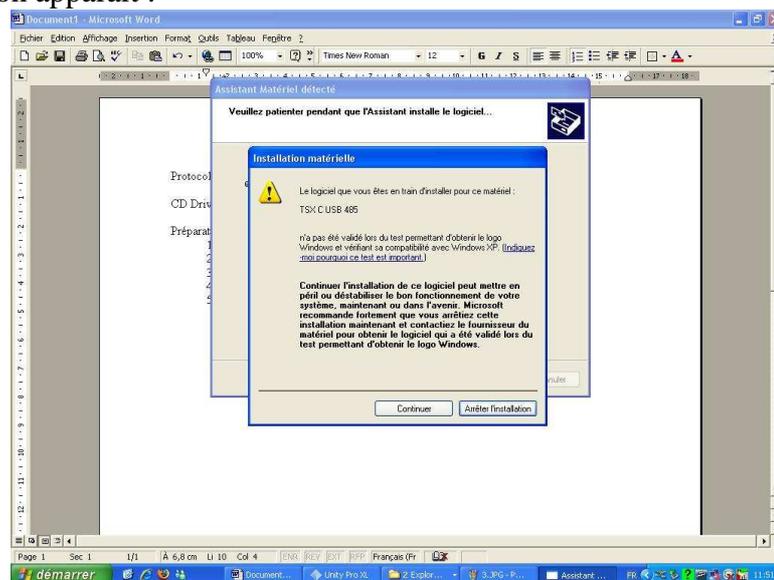
1^e étape : Nouveau matériel

- 1) Branchement câble du port TER de l'automate à n'importe quel port USB du PC
→ l'installation de nouveau matériel se lance automatiquement
- 2) Assistant Ajout de nouveau matériel :



Cochez « Non pas pour cette fois » → Suivant

- 3) Cochez « Installer à partir d'une liste ou d'un emplacement spécifié » → Suivant
- 4) Cochez « Rechercher le meilleur pilote des ces emplacements »
Puis « Rechercher dans les médias amovibles (CD ROM) » → Suivant
- 5) Information apparaît :



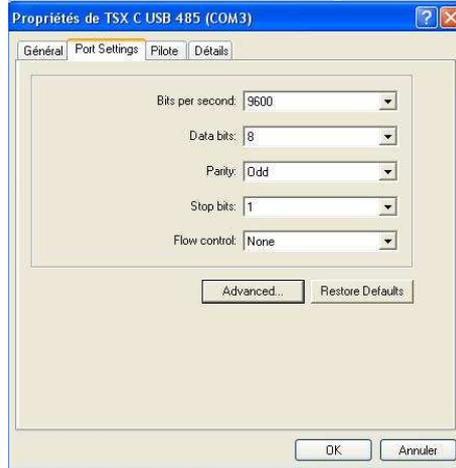
Valdez « Continuer »

- 6) Enfin, installation terminée → « Terminer »

Information : Le PC peut relancer l'Assistant « ajout de matériel détecté » car le matériel n'est pas complètement installé → recommencez donc la même procédure que ci-dessus.

2^e étape : Configuration du câble USB

- 1) Ouvrez Panneau de configuration / Outils d'administration / Gestion de l'ordinateur
- 2) Dans la fenêtre « Gestion de l'ordinateur » :
Click sur Gestionnaire de périphériques / Ports (COM et LPT)
Puis sur TSXCUSB485 click droit / Propriétés / Onglet « Port Settings » :

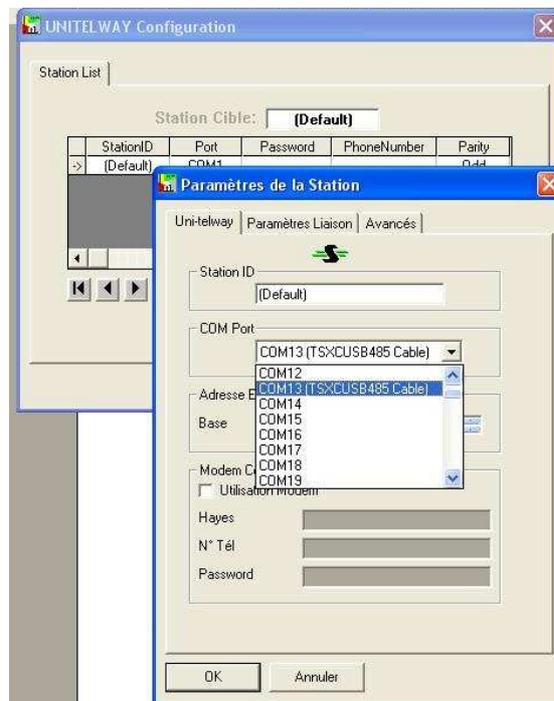


Les paramètres de communication doivent être rentrés ainsi.

- 3) Click sur « Advanced » dans cette même fenêtre
- 4) Dans la fenêtre suivante :
Sélectionnez un autre port COM : COM13 par exemple
- 5) OK
- 6) OK

3^e étape : Configuration de Unity Pro

- 1) Ouvrez Automate / Définir l'adresse / Paramètres de communication / Config Drivers
- 2) Dans la fenêtre « Gestion des drivers SCHNEIDER » :
Click sur onglet « UNITELWAY Driver » puis « Configuration »
- 3) Dans la fenêtre « UNITELWAY Configuration » : Click sur la ligne puis « Modifier »



Sélectionnez dans « COM Port » : COM13 qui doit être reconnu.

- 4) Click onglet « paramètres liaison » :
Même configuration de la communication qu'au 2) de l'étape 2
- 5) →OK
- 6) →OK
- 7) →OK
- 8) →OK
- 9) Click sur « Test connexion » : connexion réussie normalement

Il ne reste plus qu'à effectuer la première procédure décrite de connexion PC-Automate.

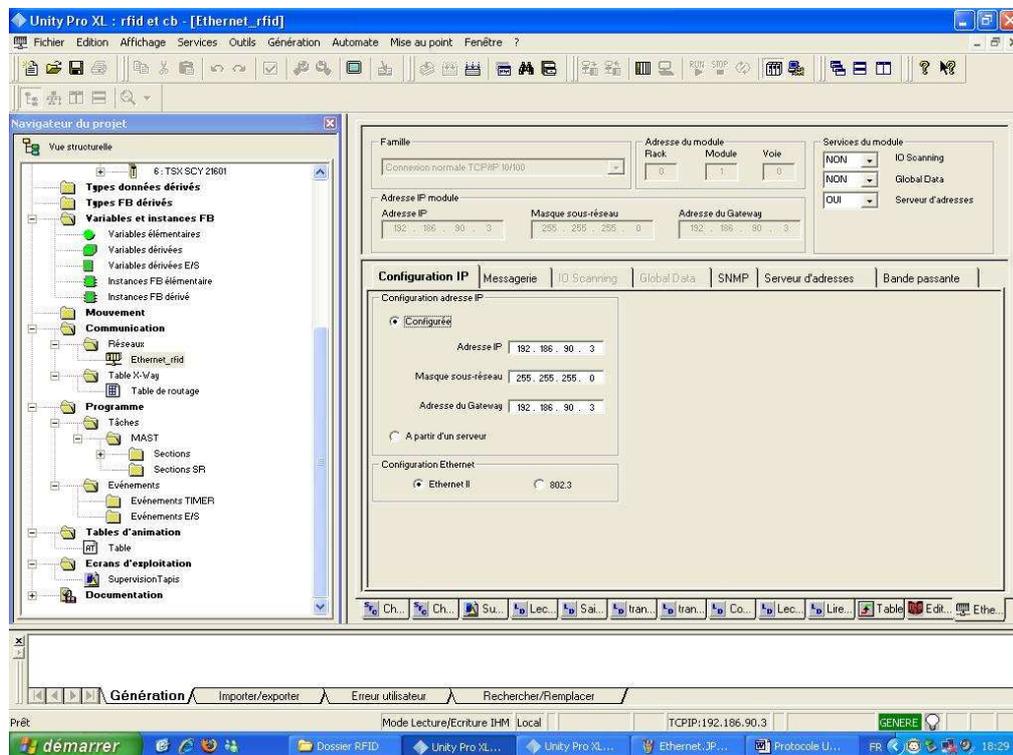
Procédure d'installation de la communication Ethernet PC-Automate :

1^{er} étape : Installation du matériel

- 1) Branchement d'un câble réseau **croisé** RJ45 à partir du port du module ETH_PORT de l'automate vers la carte réseau du PC
- 2) Configuration de la carte réseau :
 - a. Ouvrez Panneau de configuration / Connexion réseau
→ Click droit sur le réseau : « Propriétés »
 - b. Sur « Protocole Internet (TCP/IP) » : cliquez sur « Propriétés »
 - c. Entrez l'adresse IP suivante : 192.186.90.1
 - d. Entrez le Masque de sous réseau : 255.255.255.0
- 3) OK puis Fermer
- 4) La connexion PC-automate doit alors se faire automatiquement au niveau de la carte réseau du PC

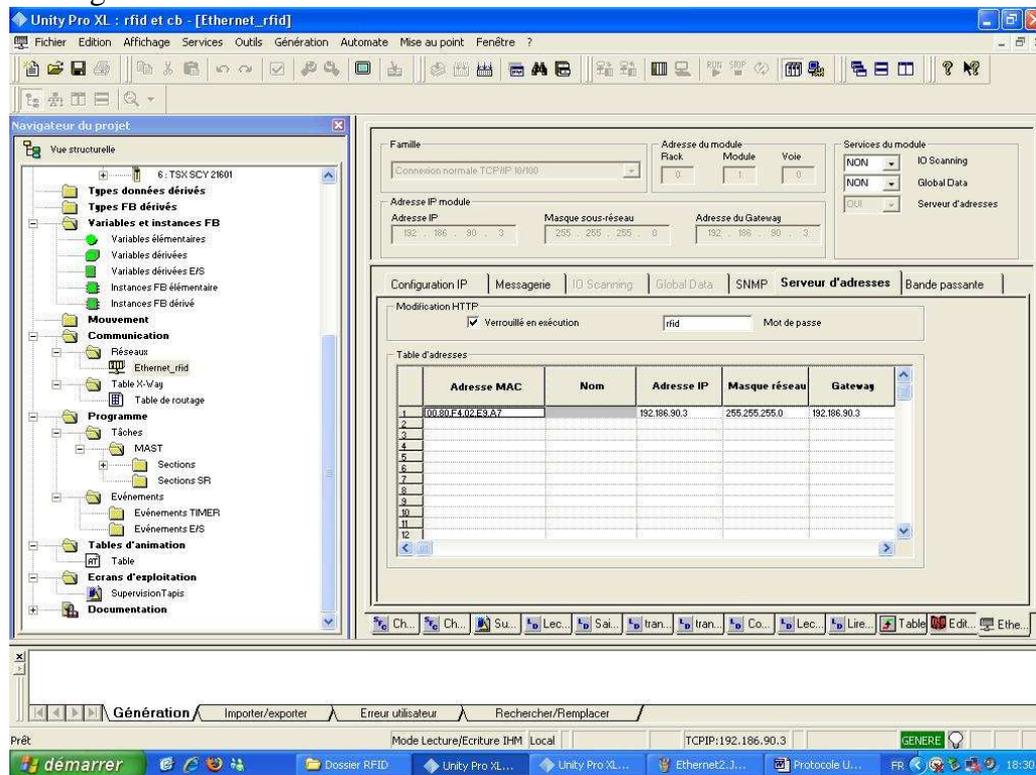
2^e étape : Configuration de Unity Pro

- 1) Dans le navigateur du projet : click droit sur Communication / Réseau
→ « Nouveau Réseau »
→ Dans la fenêtre « Ajouter Réseau » :
 - Choix du réseau disponible : Ethernet
 - Nom du réseau à changer → OK
- 2) Allez dans votre réseau nouvellement créé :



On y configure l'adresse IP de l'automate : 192.186.90.3
 Le masque de sous-réseau : 255.255.255.0
 L'adresse du Gateway : 192.186.90.3

3) On coche « OUI » en haut à droite pour le serveur d'adresse et on clique sur son onglet :



Comme ci-dessus, on entre l'adresse MAC du module ETH_PORT (inscrite sur la façade) ainsi que l'adresse IP, le masque et le Gateway comme précédemment. Les autres onglets ne sont pas utilisés.

4) Au niveau de la configuration matériel, il ne reste plus qu'à choisir ce réseau nouvellement créé pour la communication du module ETH_PORT

Enfin : on effectue la première procédure décrite de connexion PC-Automate.

Informations : l'adresse IP proposé n'est pas anodine. Le réseau 192.186.90 est un réseau de classe C conformément à la documentation pour les réseaux Ethernet de ce type et de cette taille ci. Bien sûr, l'automate et le PC ne possède pas la même adresse IP mais doivent être sur le même réseau et sous-réseau. Les numéros 1 et 3, ensuite, sont moins importants.

Conclusion

L'ensemble du projet nous a permis de nous immerger dans le domaine de l'automatisme. Nous avons réussi à proposer aux futurs étudiants d'école ingénieur un TP viable qui leur permettra d'une part de découvrir les outils industriels de traçabilité via l'utilisation de codes-barres et l'utilisation de la technologie RFID et d'une autre part de faire un TP d'automatisme dans un environnement nouveau pour eux : UNITY PRO.

En effet le scénario que l'on a proposé réussi dans un contexte industriel à employer le matériel déjà existant tout en utilisant au maximum les possibilités qu'offrent le code barre et le système de lecture/écriture RFID.

Au terme du temps impartis, la maquette est opérationnelle, les capteurs et les réflecteurs sont installés, le câblage est effectué, les informations nécessaires sont présentes. Cela rend l'ensemble clair ce qui supprime toute possibilité de confusion pour un futur utilisateur.

Du point de vue de la programmation, nous avons appris à manier le logiciel UNITY PRO. Malgré quelques déboires avec la connexion PC-Automate, nous avons quand même réussi à établir une connexion via le port ethernet. Au final, La communication avec la station RFID se fait par le module de communication et celle avec la douchette se fait par une carte PCMI. L'utilisation de ces éléments nécessitera à l'étudiant de découvrir des fonctions qu'il n'aura sans doute jamais utilisé mais qui sont triviaux pour quelqu'un dans le domaine. Le grafset final est fait et pourrait servir d'exemple la correction des TPs.

L'ensemble maquette-programme en exécution répond tout à fait au cahier des charges que l'on s'était proposé. Le temps alloué a donc été raisonnable.

Du point de vue personnel, ce projet nous a permis de nous situer vis-à-vis du monde de l'automatisme. Il nous aura donné l'occasion de découvrir une toute nouvelle technologie de traçabilité qui est le RFID.

Le travail en équipe a été la clé de la réussite dans ce projet. Cela a ainsi été l'occasion de consolider notre savoir-être l'un vis-à-vis de l'autre. Le savoir-faire et l'expérience de chacun formait un ensemble complémentaire, ce qui était bénéfique à notre travail et à l'ensemble du groupe.