

Rapport de stage

Commande d'une cellule flexible de production robotisée



Réalisé par Alexandra GORRY POLLET

A l'attention de :

Messieurs Cyril Briand et Laurent Houssin, maîtres de stage
Monsieur Samir Medjiah, enseignant tuteur de stage

Du 06 avril au 26 juin 2015

Remerciements

Tout d'abord, je tiens à remercier Monsieur Cyril Briand et Monsieur Laurent Houssin qui m'ont donné l'opportunité de réaliser ce stage et qui, par la suite, ont été mes maîtres de stage. Je leur suis particulièrement reconnaissante du temps et de l'attention qu'ils m'ont accordés durant ces 12 semaines.

Ensuite, mes pensées se tournent vers Robin Arnal avec qui j'ai partagé un projet commun et qui m'a épaulée tout au long de ce stage. Il a su partager et me transmettre son savoir en de multiples occasions.

Je remercie également Cédric Abiven, étudiant en master qui a répondu à de nombreuses questions de ma part. J'ai de plus eu la chance de collaborer avec lui tout en profitant de ses compétences.

Enfin, il me tient à coeur de mentionner Madame Francine Fugier et Monsieur Thierry Canzoneri, présents chaque jour à l'AIP-PRIMECA et attentifs au bon déroulement de ces trois mois de stage.

Table des matières

Introduction	p. 3
I) Présentation de l'établissement d'accueil	p. 4
1) Le groupe AIP-PRIMECA	p. 4
2) Le pôle de Toulouse	p. 5
3) Le personnel et les personnes rencontrées	p. 6
II) Mon stage vu de l'extérieur	p. 8
1) L'origine du besoin, les objectifs et les enjeux	p. 8
2) Les outils utilisés	p. 9
3) Une méthode de travail particulière	p. 10
4) La présentation du matériel	p. 12
III) Travail réalisé et perspectives	p. 15
1) Première phase de prise en main	p. 15
2) Résolution des problèmes rencontrés	p. 21
3) Mise en place du premier scénario	p. 22
4) Opportunités pour la suite	p. 27
IV) Bilans	p. 29
1) Un stage en lien avec le DUT et ma poursuite d'études	p. 29
2) Un avis critique sur la méthode de travail	p. 29
3) L'influence sur mon projet professionnel et la journée de la robotique	p. 30
Conclusion	p. 32
Annexes	p. 33

Introduction

Dans le cadre de mon DUT Génie Électrique et Informatique Industrielle à l'Université Paul Sabatier de Toulouse, j'ai été amenée à effectuer un stage du 06 avril au 26 juin 2015 au sein de l'antenne AIP-PRIMECA de Toulouse.

Le sujet de ce stage est "Commande d'une cellule flexible de production robotisée". Il consiste en la programmation d'une couche logicielle permettant de faire communiquer plusieurs systèmes électroniques. Dans le cas présent, il s'agit d'un vidéo-projecteur interactif et d'un système de transport de navettes sur rail équipé d'automates programmables.

Dans ce rapport, je relate mon expérience au sein de l'AIP-PRIMECA, organisme décrit dans une première partie. Ensuite, je présente le projet réalisé de manière globale avant d'analyser son déroulement. Enfin, j'effectue un bilan sur ces 12 semaines selon différents points de vue.

I) Présentation de l'établissement d'accueil

1) Le groupe AIP-PRIMECA

L'AIP-PRIMECA, ou Atelier Inter-établissements de Productique et Pôle de Ressources Informatiques pour la MECAnique, est un service dont l'objectif est d'offrir des ressources techniques, humaines et matérielles à diverses formations mais aussi en soutien à des activités de recherches.

A proprement parler, il s'agit d'un regroupement entre les AIP et les PRIMECA. Les premiers sont des centres de ressources régionaux. Ils ont été créés en 1984 par le Ministère de l'Education Nationale, en collaboration avec les régions les accueillant mais aussi avec le Ministère de l'Industrie. Les seconds ont été fondés en 1991 pour faire la promotion des outils informatiques utilisés pour la conception de produits mécaniques. A l'initiative du Comité d'Application de l'Académie des Sciences, ils avaient également pour vocation de créer une formation en Conception Mécanique Assistée par Ordinateur. Depuis cette association, le but de l'AIP-PRIMECA est de rassembler des moyens et des compétences favorables à la recherche ainsi qu'à la formation pédagogique.

L'AIP-PRIMECA est un réseau, comptant 9 antennes réparties sur le territoire français, qui contribue au regroupement d'établissements d'enseignement supérieur et qui travaille sur trois tableaux différents : la formation, la recherche et l'industrie. Ce sont donc des lieux d'échange et de partage de connaissances liées aux domaines de la mécanique, de l'informatique et de la productique.

Localement, chaque pôle est le résultat de l'association de plusieurs établissements qui sont, pour la plupart, les universités ou des laboratoires. Ces centres d'échanges sont à l'origine de la mutualisation des ressources et par conséquent de la dynamisation de la formation et de la recherche. Ces espaces sont des lieux privilégiés pour la mise en place de travaux interdisciplinaires associant des compétences pédagogiques et scientifiques qui favorisent la formation des étudiants dans le domaine de l'industrie.

Afin d'obtenir de la visibilité mais surtout de créer une cohésion autour de leurs domaines d'activités, les pôles de l'AIP-PRIMECA sont à l'origine de l'organisation de différents événements. Il s'agit, par exemple, de manifestations nationales voire internationales ayant pour vocation de renforcer les échanges au sein du milieu technique, de journées thématiques avec des professionnels, de formations ou encore de la mise en place d'une revue internationale.

Ainsi, à titre d'illustration, l'AIP-PRIMECA permet à plus de 80 formations universitaires et écoles d'ingénieurs d'utiliser du matériel ; il est la source d'une centaine de projets avec des industriels ; il est fréquentée par plus de 400 enseignants-chercheurs et par 2500 étudiants par an.

2) Le pôle de Toulouse

Après cette vue globale de l'AIP-PRIMECA , il est temps de détailler quelles sont les caractéristiques du pôle Toulousain.

Pour commencer, il est situé dans le campus universitaire de Rangueil, plus exactement au 118 route de Narbonne à Toulouse. Il est le résultat du partenariat entre :

- l'Université Paul Sabatier ;
- l'INP (Institut National Polytechnique) de Toulouse ;
- le LAAS - CNRS (Laboratoire d'Analyse et d'Architecture des Systèmes - Centre National de la Recherche Scientifique) ;
- l'INSA (Institut National des Sciences Appliquées) de Toulouse.

Ses locaux sont situés dans la Halle Technologique et sont composés de salles de cours équipées de matériel tel que des ordinateurs mais aussi de systèmes plus élaborés. Parmi eux, nous pouvons compter de nombreux robots, un vidéo-projecteur interactif et une cellule de transport gérée par 5 automates programmables. Ces deux derniers faisant l'objet du stage, ils seront décrits par la suite. Concernant les robots présents autour de la cellule, nous trouver des bras articulés de marques Kuka et Stäubli, un robot Mitsubishi pour faire du pick-and-place (déplacement d'objets) et un robot collaboratif Baxter. Baxter est un robot humanoïde en plein essor conçu pour l'industrie et les chaînes de production. On retrouve sur la *Figure 1 - Matériel de l'AIP-PRIMECA*, la cellule flexible puis, de gauche à droite, le robot Stäubli, le robot Kuka, 3 navettes de la cellule et enfin le robot Baxter.



Figure 1 - Matériel de l'AIP-PRIMECA

On trouve également des robots tels que des Nao qui sont de petits humanoïdes programmables, ainsi que des Turtlebots, robots mobiles équipés de caméra Kinect qui leur permet de se déplacer de manière autonome.

Enfin, plusieurs salles sont équipées d'ordinateurs, plus d'une centaine au total, dotés de différents systèmes d'exploitation, Windows ou Linux. Ils sont raccordés au réseau Internet de l'université.

3) Le personnel et les personnes rencontrées

Le personnel de l'AIP-PRIMECA étant assez restreint, j'ai choisi de présenter toutes les personnes que j'ai côtoyées régulièrement lors de mon stage.

Le personnel de l'AIP-PRIMECA se limite au directeur, Monsieur Cyril Briand, qui est mon maître de stage, et de deux autres personnes :

- Monsieur Thierry Canzoneri, un ingénieur responsable du matériel présent dans les locaux ;
- Madame Francine Fugier, la secrétaire.

Il est à noter que Monsieur Cyril Briand, en plus de diriger l'AIP-PRIMECA de Toulouse, est enseignant à l'Université Paul Sabatier et chercheur au LAAS CNRS.

J'ai également rencontré Monsieur Laurent Houssin qui, à l'instar de Monsieur Cyril Briand, est simultanément enseignant à l'Université Paul Sabatier, chercheur au LAAS CNRS et mon second maître de stage.

Ensuite, j'ai eu l'occasion d'échanger avec Monsieur Philippe Truillet présent à l'AIP-PRIMECA de par son statut de maître de stage de l'étudiant avec qui je menais mon projet. Il est enseignant à l'Université Paul Sabatier et chercheur à l'IRIT (Institut de Recherche en Informatique de Toulouse).

Enfin, j'ai partagé mon quotidien avec de nombreux étudiants, en stage ou pas.

Le premier, avec qui j'ai collaboré le plus souvent, est Robin Arnal, étudiant en première année de master. Il est chargé de la partie du projet en lien avec le vidéo-projecteur interactif. Nous avons donc travaillé sur le même projet, à des niveaux différents.

Ensuite, j'ai côtoyé Alex Wattel et Joffrey Bourdieu, tous les deux en stage dans le cadre de leur licence CISPM (Conception et Intégration de Services et de Produits Multimédias).

Quant à lui, Alexis Luu est étudiant en deuxième année de master à l'Université Pierre et Mari Curie en région parisienne. Son stage consistait à acquérir des compétences sur le robot Baxter, ce dernier ayant été acquis récemment par l'AIP-PRIMECA.

Enfin, je me dois de citer les 3 étudiants en master, Cédric Abiven, Grégoire Cardone et Shengheng Gao. Ceux-ci étaient responsables des Travaux d'Etudes et de Recherches (TER) sur la cellule flexible puis sont devenus stagiaires.

II) Mon stage vu de l'extérieur

1) L'origine du besoin, les objectifs et les enjeux

L'AIP-PRIMECA dispose d'une cellule flexible de production robotisée qui restait non opérationnelle. Le désir de la remettre en service a donné lieu à un sujet de stage intitulé "Commande d'une cellule flexible de production robotisée".

Cette cellule, achetée en 2001, a fonctionné jusqu'en 2009, puis son utilisation a progressivement périclité suite au déclin des formations de productique, d'une part, et au départ d'un ingénieur qui a emporté avec lui une grande partie des compétences et du savoir-faire sur le fonctionnement cette cellule flexible, d'autre part.

Aujourd'hui, avec la montée en puissance du thème "Usine du futur", de nouveaux concepts émergent ou réapparaissent tels que : "l'industrie 4.0", "l'usine numérique" ou encore "l'usine intelligente". L'AIP-PRIMECA de Toulouse souhaite profiter de ce nouvel élan et mettre à disposition des outils opérationnels auprès des formations pour le développement de projets porteurs. C'est pourquoi, il a été mis en place des actions de réhabilitation de la cellule : d'une part avec la mise en place de TER (Travaux d'Etudes et de Recherches) d'un groupe d'étudiants en master Électronique, Électrotechnique et Automatique et d'autre part, par le présent projet mis en œuvre dans le cadre d'un stage d'une durée de 12 semaines.

L'objectif du projet est d'étendre le fonctionnement de la cellule flexible en lui permettant de pouvoir communiquer avec d'autres systèmes électroniques et logiciels. En l'occurrence, l'objectif premier est de pouvoir interagir avec un vidéo-projecteur interactif. Concernant le vidéo-projecteur, il s'agit de pouvoir commander la cellule depuis une interface projetée sur un tableau et de recevoir et visualiser des informations en temps réel sur l'activité de la cellule flexible.

Pour cela, il est nécessaire de créer une couche logicielle capable de faire communiquer tous les dispositifs en passant outre les contraintes du réseau physique. Un prérequis est que les automates programmables de la cellule flexible soient capables de communiquer, via un réseau ethernet, entre eux mais aussi avec un dispositif extérieur.

L'une des particularités de ce projet est de travailler en équipe avec un autre stagiaire, Robin Arnal, étudiant en master Systèmes Interactifs et Robotiques. Ce dernier est chargé du développement de l'interface du tableau interactif et il doit prendre en compte sa mise en relation avec la cellule flexible. C'est pourquoi les deux travaux sont extrêmement liés et sont menés de front. Cela engendre le besoin d'une communication constante entre les personnes engagées.

De plus, d'autres travaux, liés à la commande bas niveau de la cellule flexible, sont en cours de réalisation par les étudiants chargés des TER sus-cités. Leur projet étant déjà avancé au commencement de ce stage, il fut nécessaire de prendre connaissance du travail déjà réalisé et de

s'informer régulièrement de celui développé en parallèle avec ce stage. Notons que les étudiants en TER ont prolongé leur projet par un stage ayant lieu au mois de juin.

Ce cadre est l'occasion d'échanger et de mettre en commun les connaissances et compétences de chacun ce qui est un véritable atout. Cependant, la contrainte se pose de devoir se coordonner et aussi de se mettre d'accord sur des idées communes. Mais, la limite la plus importante est le temps restreint qui induit une compréhension rapide du sujet et un apprentissage efficace de l'outil principal de mise en oeuvre.

Parmi les enjeux de ce projet, on trouve tout d'abord le fait de regagner la maîtrise perdue de la cellule flexible. En effet, comme dit plus haut, les compétences antérieures n'ont pas pu être conservées.

Retrouver cette maîtrise et relancer des travaux en lien avec la cellule flexible implique aussi de cibler ses obsolescences, c'est-à-dire d'adapter les logiciels et le matériel la concernant, après un certain temps passé à ne pas s'en occuper. De plus, le fait de permettre la communication entre la cellule et les autres systèmes alentours offrira à l'AIP-PRIMECA une toute nouvelle façon de travailler avec les dispositifs présents dans les locaux.

Enfin, remettre en fonction la cellule flexible permettra aux étudiants de l'Université Paul Sabatier de s'exercer sur un "nouveau" système complexe opérationnel pour l'apprentissage de disciplines telles que l'informatique industrielle, la commande de systèmes à événements discrets, la robotisation, l'interaction.

2) Les outils utilisés

Afin d'optimiser le fonctionnement de la cellule flexible et de pouvoir arriver au résultat escompté, l'emploi de certains moyens techniques s'est imposé.

L'un d'eux est l'utilisation du middleware ROS (Robot Operating System) qui est un outil informatique permettant de développer des logiciels dans le domaine de la robotique. ROS permet de faire communiquer différents dispositifs informatiques en passant outre les contraintes physiques, en d'autres mots cela veut dire que l'on utilise un bus logiciel. L'avantage de ROS est de pouvoir faire transiter des messages entre les différents processus, ce qui est très utile dans la résolution du présent projet. Ce middleware étant opensource, il existe un certain nombre de fonctions déjà construites et pouvant être réutilisées. Le plus appréciable est de trouver une vraie communauté supportant ROS. Cela permet de trouver nombre de tutoriels et textes explicatifs sur le sujet. Conçu en 2007, ROS a connu plusieurs versions. Celle utilisée dans ce contexte est la huitième et porte de nom le Hydro Medusa, qui est fonctionnelle sous une distribution Linux, Ubuntu dans notre cas.

Ensuite, nous pouvons citer le logiciel PL7 Pro. Ce dernier a été développé par Schneider Electric qui est, entre autres, le fabricant de certains automates programmables présents à l'AIP-PRIMECA. PL7 Pro sert donc à programmer les automates et offre la possibilité d'avoir recours à 4 langages différents : le langage à contact (LD), la liste d'instructions (IL), le littéral structuré (ST) et le Grafcet (G7). Dans le cas présent, seuls les deux derniers ont été utilisés. En plus de pouvoir programmer les automates au sens propre du terme, le logiciel PL7 Pro permet de configurer le réseau qu'ils constituent. Cela est une étape cruciale dans le déroulement de ce stage comme expliqué ultérieurement dans de ce rapport. On note cependant que ce logiciel fait partie de ceux qui devront être mis à jour à l'avenir (passage à Unity Pro).

Toujours en rapport avec les automates, nous avons choisi de les faire communiquer via le protocole TCP/IP grâce à la librairie Modbus : celle-ci a été conçue pour les automates programmables. Ce protocole de transfert de données est largement utilisé sur Internet mais dans notre cas, il suffit de connaître les adresses IP des constituants du réseau pour pouvoir échanger des trames. Ces dernières permettront, par exemple, de lire ou d'écrire dans la mémoire des automates, ce qui est un impératif dans la mise en place de la communication entre la cellule flexible et le tableau interactif. Le protocole Modbus est normalisé : il est inclus de base sur les cartes réseau des automates et reste l'un des plus fiables et des plus simples parmi ses semblables. C'est pourquoi le choix s'est tourné vers lui.

Concernant la programmation en elle-même, les deux outils présentés ci-dessus nécessitent la connaissance de différents langages. Pour le premier, ROS, la programmation s'effectue en C++. Néanmoins, nous pouvons avoir recours au Python qui sert à effectuer des prototypages de manière plus et plus rapide. En ce qui concerne la programmation des automates, les langages utilisés sont le ST et Grafcet. Il est tout de même à noter que l'utilisation du Grafcet a été plus fréquente dans ce travail.

Enfin, ce stage fait aussi appel à des compétences plus globales, comme la connaissance de l'environnement Ubuntu qui n'est pas familier à tout le monde. Nous pouvons aussi citer la connaissance des commandes Unix car le travail en console occupe une grande place dans ce stage. Et, de manière encore plus large, nous pouvons citer les compétences liées à la gestion de projet. Pour illustrer ces propos, il est possible de mentionner la rédaction de documents, l'organisation dans le temps de son travail ou la réactivité face aux problèmes rencontrés.

3) Une méthode de travail particulière

L'organisation de ce stage était basée sur la méthode agile Scrum. Il s'agit d'un fonctionnement dont le principe est de fournir des livrables fréquemment. Ainsi, il a fallu produire du contenu régulièrement, dans des laps de temps relativement courts. Ces intervalles de temps s'appellent des sprints durant lesquels il faut produire de manière la plus efficace possible en alliant qualité et quantité. Dans le cas présents, les sprints avaient une durée d'environ deux

semaines. C'est donc à cette fréquence qu'étaient organisées des réunions entre stagiaires et maîtres de stage. Durant ces rencontres, des objectifs et des échéances datées étaient fixés en tenant compte du volume de travail restant à effectuer. A chaque nouvelle réunion, le travail effectué sur la période se terminant était examiné et commenté. Cela permettait, si nécessaire, de livrer une version corrigée à la réunion suivante. Comme dans toute structure, l'organisation des réunions avait un rôle prédominant : élaboration des ordres du jour et rédaction des comptes-rendus avec la planification des opérations restant à mener, transféré à chacun des participants.

En début de stage, avant de passer à l'avancement concret de l'aspect technique du projet, les deux premiers livrables à rendre furent les cahiers des charges et le plan de développement.

Le cahier des charges était à réaliser par le stagiaire. Cela a nécessité une réflexion profonde aboutissant à compréhension du sujet, jusqu'alors exprimé seulement à l'oral. En plus de cela, il a fallu cibler les besoins logiciels engendrés par la demande, les points techniques auxquels il faudrait avoir recours, les limites à respecter, les améliorations à envisager ... En fait, le sujet devait être présenté dans sa globalité. De plus, ce document devait servir "d'introduction" au stage. C'est-à-dire qu'il avait aussi pour vocation de présenter le contexte du stage ainsi que ses enjeux et toute chose pouvant amener un néophyte à la compréhension de tous les tenants et les aboutissants de ce stage. Au final, ce document, composé d'une petite dizaine de pages, comporte les parties suivantes :

- Présentation du projet ;
- Caractérisation du matériel ;
- Caractérisation de la demande.

Il a été remis, dans sa première version le 16 avril 2015, soit moins de deux semaines après le démarrage du stage. Grâce aux retours des maîtres de stage, sa version finale a été achevée pour la réunion suivante.

Suite à cela, et en collaboration avec le second stagiaire, un plan de développement a été rédigé. Celui-ci devait établir une vue globale du projet détaillant les différentes étapes à suivre pour sa réalisation. Ce document avait la particularité d'être évolutif. Il en existe autant de versions que de mises à jour, chacune contenant chacun un scénario d'utilisation supplémentaire à la précédente. Ce document contient la vue finale du projet, ce à quoi il ressemblera une fois abouti. Pour cela, il a fallu identifier tout ce qui était possible de faire dans ce contexte précis. Ce qui est présenté dans ce document n'est pas forcément réalisable dans le délai qui nous est imparti. Néanmoins, il montre ce à quoi pourraient aboutir les travaux entamés une fois optimisés. Forcément, cela implique la nécessité d'identifier ce qui était faisable durant cette courte période, prioriser certaines choses, en mettre d'autres de côté et surtout prendre en compte toutes ces améliorations et les prévoir. Les parties qui le composent sont :

- Vue finale du projet ;
- Planning ;
- Les différents scénarii, ajoutés de manière itérative à chaque nouvelle version.

Ces deux documents, cruciaux dans la compréhension et le développement du sujet, fournissent tous les détails traitant la résolution, le déroulement et l'avancement du projet. Ensemble, ils ont été un véritable guide, autant pour nous stagiaires, que pour nos maîtres de stage et le seront potentiellement pour toute personne externe au projet qui voudrait s'y intéresser.

Néanmoins, bien que ces documents constituent les seuls textes rédigés à fournir, chaque réunion était l'occasion de faire le point sur le travail effectué et de vérifier si les objectifs étaient atteints ou non. Les rencontres étaient de plus des occasions de faire des démonstrations de nos travaux.

4) La présentation du matériel

La cellule flexible de l'AIP-PRIMECA est un système de transport intelligent nommé Montrac. Ce système de transport est monorail et possède des navettes nommées "Shuttles" qui sont en fait des porte-palettes. Ces derniers sont autonomes et intelligents : ils se déplacent seuls et ils ne peuvent pas être programmés : ce sont les automates qui gèrent la commande des différents actionneurs présents sur le rail. C'est pourquoi les navettes possèdent un capteur de proximité qui évite les collisions avec les autres. Les palettes transportées pèsent jusqu'à 30kg et mesurent jusqu'à 300 par 550 mm. Les navettes se déplacent grâce au rail qui les alimente en énergie. Celle-ci leur est transmise par des bandes métalliques accolées au rail. Leur vitesse est elle aussi gérée par le rail. En effet, sont présents sur ce support des petits ergots métalliques. Il en existe de différentes tailles et c'est cela qui indique à la navette la vitesse de déplacement sur chaque tronçon du rail. Par exemple, une pièce longue va la faire accélérer et une pièce courte va la faire ralentir tandis qu'une pièce verticale la fera s'arrêter.

Montrac est aussi un système modulaire. En effet, il existe plusieurs types de rails (rectilignes, courbes, aiguillages ...) qui peuvent être assemblés de différentes manières. On remarque qu'une fois assemblés, ils n'existent qu'un sens de circulation. Les risques de collisions frontales sont évitées grâce à cet avantage mais il faut en tenir compte lors de la programmation du dispositif.

La cellule flexible de l'AIP-PRIMECA comporte 5 "secteurs" dont un central et 4 desservants des postes de travail. Ceux-ci permettent à d'autres systèmes (robots Baxter ou Kuka, par exemple) d'avoir accès aux porte-palettes et donc d'interagir avec. La cellule est représentée en vue de dessus sur la *Figure 2 - Schéma de la cellule flexible*.

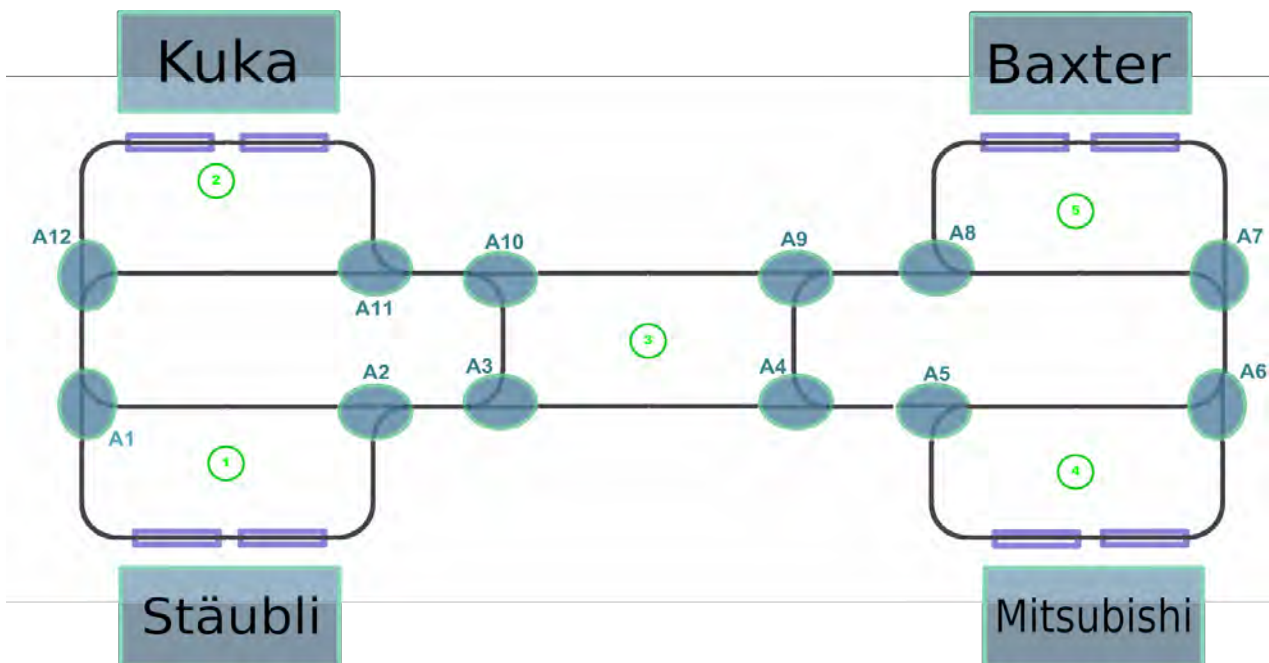


Figure 2 - Schéma de la cellule flexible

De plus, chaque "secteur" est contrôlé par un automate programmable qui agit sur les actionneurs de la cellule (essentiellement les aiguillages) via un système d'air comprimé. Ici, il y a 3 automates de la marque Schneider Electric, situés dans les zones 1 à 3, et 2 de la marque Siemens gérant les zones 4 et 5. Ceux-ci donnent accès à un jeu de capteurs et d'actionneurs. Les principaux sont des capteurs de présence et des actionneurs permettant d'arrêter les shuttles ou d'orienter les aiguillages. Chaque automate n'ayant accès qu'à une partie de la cellule, il peut être nécessaire de les faire communiquer entre eux afin de faire circuler les navettes sur l'ensemble du rail. En effet, ils pourront, par exemple, s'envoyer l'état de leurs capteurs respectifs pour savoir si une zone est accessible ou pas.

Dans un deuxième temps, il est important de faire une brève description du vidéo-projecteur interactif. En effet, même si aucun travail ne sera réalisé dessus à proprement parlé dans le sujet de stage présenté ici, il est un élément important du cahier des charges et des scénarii du plan de développement.

Le tableau interactif de l'AIP-PRIMECA est le modèle Epson EB-595 WI, représenté sur la Figure 3 - Vidéo-projecteur interactif Epson EB-595 WI.



Figure 3 - Vidéo-projecteur interactif Epson EB-595 WI.

Il possède 3 modes de fonctionnement :

- le mode tableau blanc ;
- le mode annotation ;
- le mode interactif sur ordinateur.

Il peut prendre en charge jusqu'à 6 points de contact. Ces points de contact sont généralement assurés par les doigts ou les stylets fournis. Ces stylets possèdent un bouton qui permet de simuler le clic droit de la souris. Le mode qui nous servira le plus ici est le mode interactif sur ordinateur qui permettra d'offrir une interface visuelle et tactile à l'utilisateur.

III) Travail réalisé et perspectives

1) Première phase de prise en main

Avant de commencer le travail effectif, j'ai d'abord procédé à une sorte d'étude du matériel, des outils et des travaux déjà réalisés sur la cellule flexible.

Premièrement, l'objectif était d'appréhender le fonctionnement du matériel. Pour cela, la première étape, cruciale, a été d'observer la cellule, de bien se la représenter et de comprendre son fonctionnement. J'ai donc dans un premier temps posé beaucoup de questions à son sujet. Cette phase d'apprentissage relativement courte mais très dense a été déterminante pour la suite du stage. En effet, possibilités offertes par la cellule flexible et son comportement devaient être connus. Quelques scénarii d'utilisations ont alors émergé et d'autres se sont révélés impossibles. Grâce aux travaux en cours, j'ai eu l'opportunité de la voir en fonctionnement, avec des tâches simples. Cela fut très intéressant. D'une part, pour une représentation encore plus complète de ce dont elle était capable et d'autre part dans la compréhension des jeux de capteurs et d'actionneurs. Le système de tronçons à sens unique et d'aiguillages rotatifs étant assez complexe, rien n'a été plus instructif que de voir les navettes circuler sur le système.

Ayant acquis une vue extérieure du système, je me suis penchée sur la façon de commander ce dispositif. Plus précisément, je me suis plongée dans la méthode de programmation des automates. Pour cela, j'ai eu recours aux connaissances produites par le groupe d'étudiants responsables des TER. Grâce aux documents qu'ils avaient pris soin de rédiger, j'ai compris la manière dont il fallait programmer les automates dirigeant la cellule flexible. Ce qui m'a le plus aidé est le schéma représentant le système et son jeu de capteurs et d'actionneurs illustrés sur la *Figure 4 - Schéma de la cellule flexible avec capteurs et actionneurs* et sur la *Figure 5 - Signification des capteurs et actionneurs* ci-dessous ainsi qu'un des Grafcet qu'ils avaient codés. Ce Grafcet est représenté sur la *Figure 6 - Grafcet de fonctionnement d'un automate*.

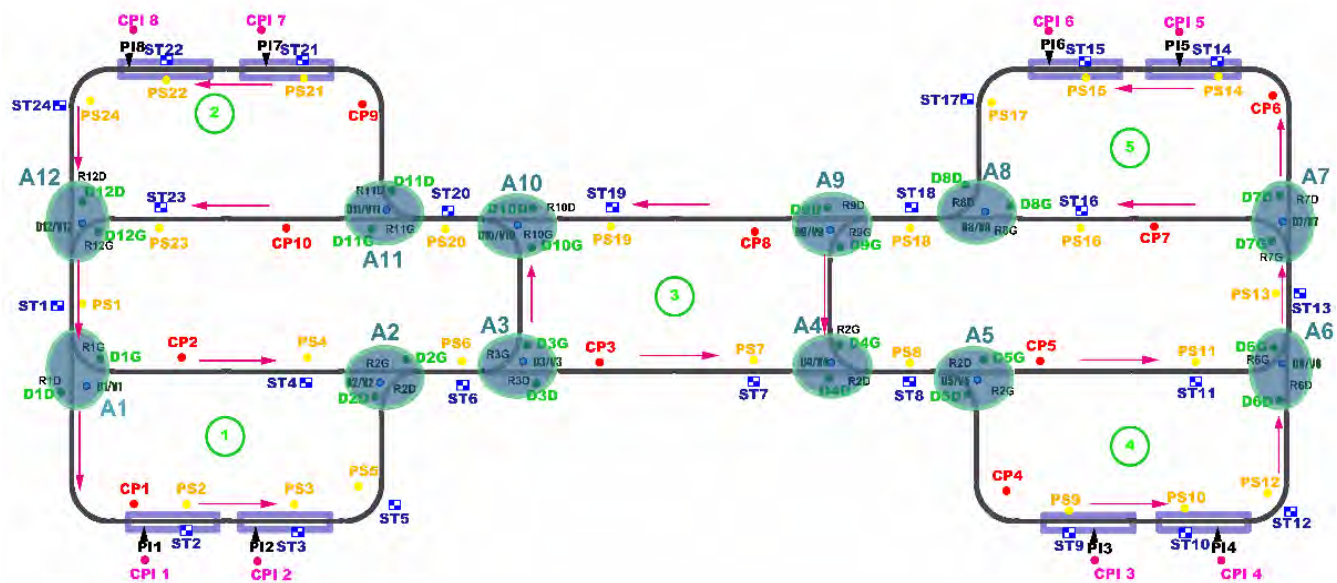


Figure 4 - Schéma de la cellule flexible avec capteurs et actionneurs

Capteurs:

CP	Capteur de position
PS	Capteur de position en face d'un start/stop
CPI	Capteur de maintien d'une navette dans un start/stop

Actionneurs:

DxD & DxG	Orientation droite et gauche des aiguillages
ST	Start/stop
PI	Maintien d'une navette à un poste de travail
Dx & Vx	Verrouillage et déverrouillage des aiguillages

Figure 5 - Signification des capteurs et actionneurs

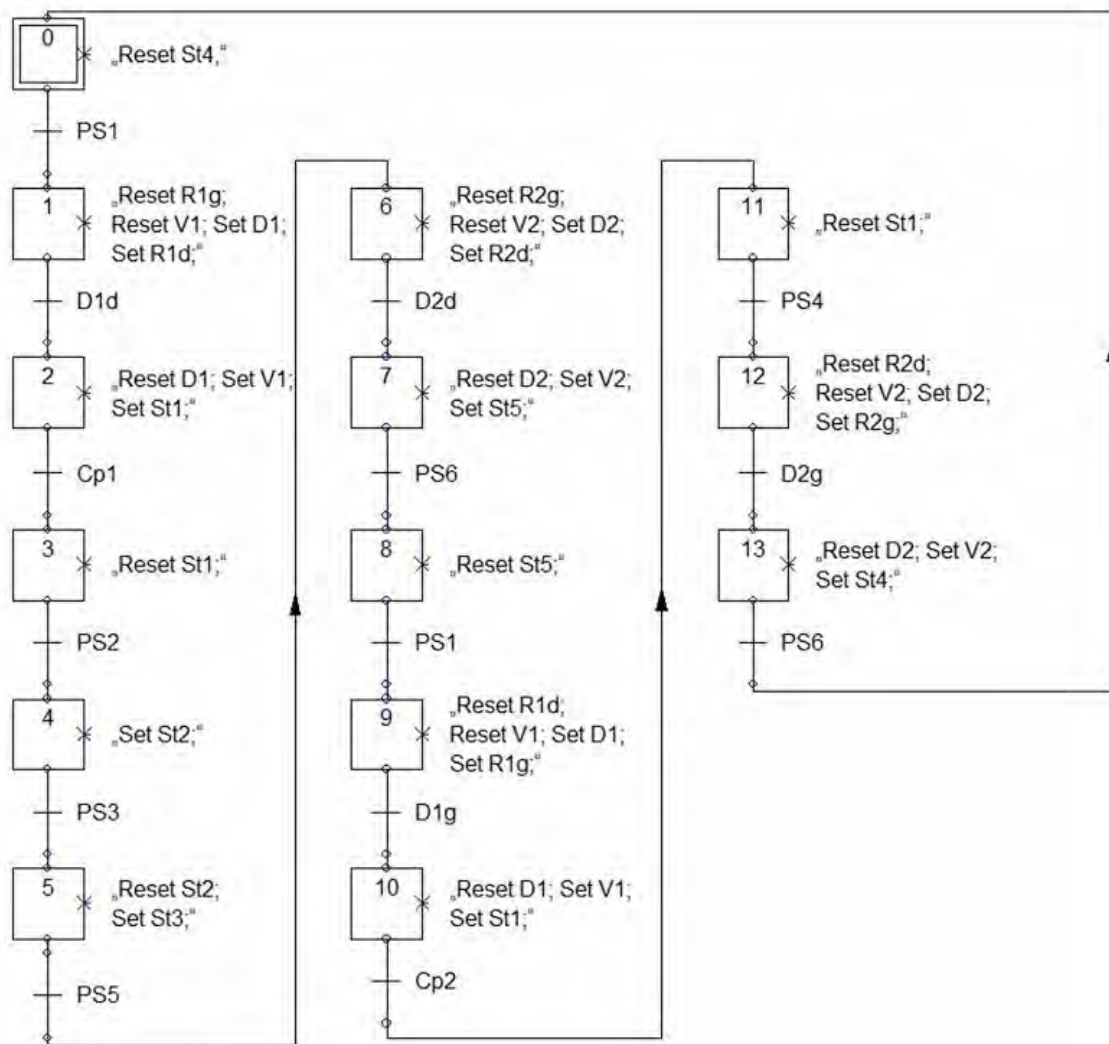


Figure 6 - Grafset de fonctionnement d'un automate

Concrètement, j'ai interprété les différentes étapes du Grafset ainsi que les conditions de transition. J'ai reconstitué pas à pas les séquences d'actions à réaliser pour le bon fonctionnement de la cellule flexible. Par exemple, pour faire bouger un aiguillage vers la droite, il faut respecter la séquence suivante :

- indiquer à celui-ci qu'il va être en mouvement en activant une variable "déverrouiller" et désactivant une variable "verrouiller" ;
- ensuite, mettre à 1 la commande l'orientant vers la droite et à 0 celle l'orientant vers la gauche pour éviter les conflits ;
- puis, lorsque l'aiguillage est en place et que le capteur le fait savoir, verrouiller à nouveau en faisant l'inverse de la première étape.

La phase suivant cette découverte du matériel a été celle de l'apprentissage de l'outil ROS. Cette étape a demandé un certain temps. En effet, j'ai voulu bien maîtriser son fonctionnement, les possibilités qu'il offre et ses limites afin d'avoir une vue claire du travail que je pourrai réaliser grâce à lui.

Pour cela, j'ai eu recours au wiki explicatif de ROS qui propose de nombreux tutoriels.

Le premier concerne la création d'un espace de travail. En effet, pour compiler un projet ROS, il faut lancer des commandes spécifiques directement depuis le terminal. C'est pourquoi tous les fichiers composant le projet doivent être organisés suivant une certaine hiérarchie bien précise. L'espace de travail ainsi créé s'appelle un *workspace* tandis qu'un projet s'appelle un *catkin*. Chaque tutoriel fournit les commandes à taper dans le terminal et explique le rôle de chacune.

Une fois mon *workspace* créé, j'ai conçu mes premiers noeuds ou *nodes* en anglais. Sous ROS, un noeud est comme un objet avec une fonction particulière qui va communiquer avec d'autres noeuds. En fait, lorsqu'on monte un projet ROS, on crée plusieurs noeuds regroupés dans un *package* qui formeront un réseau et construiront un tout fonctionnel.

Les premiers noeuds que j'ai créés ont été un *subscriber* et un *publisher*. Ces derniers ont vocation de communiquer sur un *topic*. Un *topic* est, de façon imagée, un canal où vont transiter des messages. Ces messages vont être envoyés en boucle par le *publisher* à une fréquence donnée. Par exemple, l'état d'un capteur. Dès qu'un message arrive sur le *topic*, le subscriber est notifié et récupère le message. Un *topic* peut accepter plusieurs *subscribers* et plusieurs *publishers*, il suffit que la fonction de chaque noeud soit précisée dans son code. Pour illustrer mes propos, le premier tutoriel est décrit dans la *Figure 7 - Code d'un subscriber en C++* et dans la *Figure 8 - Code d'un publisher en C++*.

```
#include "ros/ros.h"
#include "std_msgs/String.h"

/* On crée une fonction qui sera appelée à chaque nouveau message publié dans le topic */
void chatterCallback(const std_msgs::String::ConstPtr& msg)
{
    /* On affiche sur la console "I heard" puis les données de la variable msg */
    ROS_INFO("I heard: [%s]", msg->data.c_str());
}

int main(int argc, char **argv)
{
    /* Ici, listener est le nom de notre noeud. Cette ligne initialise ROS */
    ros::init(argc, argv, "listener");

    /* On initialise le noeud */
    ros::NodeHandle n;

    /* On abonne le noeud au topic nommé chatter en tant que subscriber. On spécifie que
    dès qu'un message est publié, on appelle la fonction chatterCallback */
    ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);

    /* On rentre dans une boucle qui sonde le topic le plus vite possible à la recherche
    de nouveaux messages */
    ros::spin();

    return 0;
}
```

Figure 7 - Code d'un subscriber en C++

```

#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>

int main(int argc, char **argv)
{
    /* Ici, talker est le nom de notre noeud. Cette ligne initialise ROS */
    ros::init(argc, argv, "talker");

    /* On initialise le noeud */
    ros::NodeHandle n;

    /* On abonne le noeud au topic nommé chatter en tant que publisher*/
    ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);

    /* On spécifie la fréquence à laquelle la boucle est appelée. Ici 10Hz */
    ros::Rate loop_rate(10);

    /* Déclaration d'une variable count servant à compter le nombre de messages envoyés */
    int count = 0;

    /* ros::ok() est vrai lorsque le noeud est actif dans la console */
    while (ros::ok())
    {
        /* Déclaration d'une variable msg de type "message", transitant dans le topic */
        std_msgs::String msg;

        /* Déclaration d'une variable ss de type chaîne de caractères*/
        std::stringstream ss;

        /* La variable ss reçoit "hello world" et le compteur de messages*/
        ss << "hello world " << count;

        /* La variable msg reçoit dans ses données le contenu de la variable ss*/
        msg.data = ss.str();

        /* On affiche sur la console les données de la variable msg*/
        ROS_INFO("%s", msg.data.c_str());

        /* On publie le message msg dans le topic*/
        chatter_pub.publish(msg);

        /* Le noeud "dort" jusqu'à ce qu'il soit appelé à nouveau, ici à une fréquence de 10 Hz*/
        loop_rate.sleep();

        /* On compte un message en plus*/
        ++count;
    }
    return 0;
}

```

Figure 8 - Code d'un publisher en C++

Néanmoins, j'ai préféré, par la suite, créer mon propre *topic*. Dans ce dernier, le *publisher* envoyait aléatoirement des 0 et des 1. Alors, le subscriber devait récupérer ces messages et les analyser. Si c'était un 0, il affichait "Au revoir" sur la console, si c'était un 1, il affichait "Bonjour". Cela peut paraître très simple et très basique, cependant, cela m'a servi à analyser le contenu d'un message pour ensuite adapter le comportement du *subscriber*. Il a fallu gérer le type des messages envoyés (ici, une chaîne de caractères) et trouver le moyen de tester son contenu, étape obligatoire pour la suite du projet.

En plus des *topics*, il existe aussi des *services*. La différence entre *topics* et *services* réside dans le fait que pour qu'une information soit publiée dans un *service*, il faut qu'un noeud en fasse la demande. Le noeud qui publie le message en retour rend alors un "service" à celui qui le demande. Il y a donc communication entre un *client* et un *server* qui acquitte son bon fonctionnement en retour. Là encore tout le code était fourni pour pouvoir créer un système qui

permet d'entrer deux entiers dans le *client* qui va demander au *server* de les additionner et de lui retourner la somme.

Cependant, je ne me suis pas arrêtée là. J'ai poussé le concept plus loin. J'ai modifié les noeuds précédents pour que la moyenne de dix chiffres générés aléatoirement soit calculée. Cela m'a permis de ne pas m'arrêter à un code déjà fourni et donc de me faire la main sur le fonctionnement de ROS.

Une fois que j'avais jugé ma connaissance de ROS suffisante pour la suite du projet, je me suis penchée sur une toute autre chose : la communication entre les automates.

Le travail des TER permettait déjà de charger des programmes sur les automates. Cependant, cela n'était possible que par communication série entre un ordinateur et l'automate. La première étape a été de charger un programme sans passer par un câble série mais directement par le réseau grâce aux adresses IP.

Dans cette partie du travail, l'aide de mes maîtres de stage Messieurs Cyril Briand et Laurent Houssin a été très précieuse. Pour sa part, Monsieur Cyril Briand, grâce à des recherches communes avec moi, a su identifier le *driver* manquant au logiciel PL7 Pro qui permet la communication ethernet. Ensuite, Monsieur Laurent Houssin m'a consacré une après-midi pour parvenir à débloquer ce problème. Grâce à lui, j'ai par la suite collaboré avec les étudiants de master pour faire fonctionner les fonctions "read_var" et "write_var". La première permet à un automate d'aller lire des informations dans la mémoire d'un autre automate et de les stocker dans sa propre mémoire. Les paramètres de cette fonction sont présentés dans la *Figure 9 - Description de la fonction "read_war"*. La seconde permet à un automate d'écrire dans la mémoire d'un autre. Cette étape nous a permis de savoir que la communication entre automates était possible par le réseau, ce qui n'était pas chose évidente jusque là.

- Fonction READ_VAR
- ```
READ_VAR{adr#{s}SYS,type,i,j,typek:n,%MWm:4)
```
- {s}            numéro de la station objet de l'échange, du type {n°réseau.n°station}.
- Type            type de données lues '%MW' pour un mot, '%M' pour un bit, etc.
- i                adresse des données à lire
- j :              nombre de données à émettre
- %typek:n        adresse de stockage des données reçues  
                  avec k l'adresse de base et n le nombre de données
- %MWm:4         4 mots de gestion, m décrit l'adresse de base des mots utilisés.
- Parmi ces 4 mots, les deux premiers mots de gestion sont utilisés comme suit :
- %MWm : l'octet de poids fort contient un identifiant de l'échange, le bit de poids faible %MWm : x0 est le bit d'activité (à 1 lorsqu'un échange est en cours).
  - %MWm+1 : l'octet de poids fort contient un rapport d'opération, l'octet de poids faible un rapport relatif à la communication. Ces codes renseignent sur le déroulement de la communication.

*Figure 9 - Description de la fonction "read\_war"*

Cette avancée m’a poussée à l’étape suivante : communiquer via le réseau avec un automate depuis un ordinateur non équipé de PL7 Pro. Pour cela, il suffisait de créer un programme en C chargé de créer un lien entre l’ordinateur et l’automate. J’ai eu recours à la librairie Modbus ou Libmodbus. Cette librairie contient les fonctions adéquates pour développer une application utilisant le protocole TCP/IP qui est intégré de base sur les automates. Les fonctions utiles s’appellent “modbus\_read\_registers” et “modbus\_write\_registers” et ont, à quelques choses près, le même fonctionnement que les fonctions “read\_var” et “write\_var” citées plus haut. Ceci a été un pas décisif qui a été la dernière étape avant de pouvoir commencer à mettre en place des scénarii de tests de gestion de la cellule par le tableau interactif.

## 2) Résolution des problèmes rencontrés

La plupart des problèmes que j’ai rencontrés ont eu lieu lors de la première phase de travail. Par conséquent, celle qui suit et qui vise à mettre en place la forme finale du projet, n’a débuté que lorsque toutes les conditions de sa réussite furent réunies.

Lors de la découverte du matériel, il m’a fallu un certain temps pour bien comprendre son fonctionnement. Cependant, Monsieur Cyril Briand a su répondre à mes questions et sa présence fut nécessaire à ce moment-là. J’ai aussi pu m’appuyer sur la collaboration avec les étudiants en master qui m’ont apporté de nombreux éléments et m’ont donné accès à tous les documents qu’ils avaient produits et regroupés sur un Drive. Dans l’ensemble, on peut dire que je n’ai pas eu de problèmes majeurs pour découvrir et comprendre le fonctionnement du matériel.

Concernant l’apprentissage de ROS, les problèmes rencontrés ont été moindres. C’était une phase de découverte, les problèmes étaient faciles à surmonter et bénéfiques pour un apprentissage efficace. De plus, j’ai eu la chance d’être accompagné par Robin Arnal, l’autre stagiaire, qui avait déjà une certaine maîtrise de cet outil et qui m’a fourni une aide précieuse lors de mes premiers pas dans ce nouvel environnement. Je dirai donc que cette partie ne m’a pas vraiment posé de problème.

Les problèmes plus sérieux ont commencé lorsque j’ai utilisé le logiciel PL7 Pro pour faire fonctionner les automates en réseau. Ce qui a engendré le plus de difficultés est le fait que je n’avais jamais travaillé avec ce logiciel. Durant les premiers jours, j’ai pu apprendre à faire des choses basiques comme compiler un programme et le charger sur un automate programmable, mais mes connaissances s’arrêtaient là. Malgré différentes tentatives, je n’arrivais à rien de concluant. De plus, bien qu’ayant écumé Internet pour trouver des solutions à mes problèmes, rien ne semblait régler ces difficultés. Néanmoins, dès qu’un élément me semblait intéressant, je le transmettais à Monsieur Cyril Briand. Ainsi, il a pu identifier un élément clef : le *driver* XIP qui est nécessaire à PL7 Pro pour la communication réseau. Il a donc pris de son temps pour l’installer et comprendre son fonctionnement. J’ai, pour ma part, essayé de faire fonctionner la fameuse fonction “read\_var”, mais sans résultat. J’en ai conclu que cela était en partie dû au fait que les

descriptions de la fonction trouvées sur Internet étaient toutes différentes dans la syntaxe des paramètres. Je n'ai malheureusement pas réussi à passer l'étape de la compilation ... C'est à ce moment que j'ai eu besoin de l'intervention de Monsieur Laurent Houssin est arrivée. Grâce à sa connaissance des automates et de PL7 Pro, il a su m'expliquer comment faire fonctionner le lien réseau. A ce stade, même si la fonction "read\_var" ne marchait pas, nous avions déjà des résultats probants : il était possible d'envoyer un *ping* aux automates sans passer par la liaison série, ce qui était déjà probant. Nous avons aussi trouvé comment configurer le logiciel pour que le chargement du programme sur l'automate se fasse par le réseau. C'est suite à cela que les étudiants en master et moi avons réussi à faire fonctionner la fonction "read\_var".

Dans l'étape suivante, la découverte de la librairie Modbus, l'intervention de Monsieur Cyril Briand a une fois de plus été déterminante. D'une part car c'est lui qui a trouvé les fonctions qui permettent de lire et écrire sur les automates (je n'avais pas identifié les bonnes) et, d'autre part, car son intervention efficace m'a permis de comprendre comment configurer les options de compilation pour que la librairie soit bien référencée dans la console.

Quant au travail final de mise en place du premier scénario de test, il ne m'est pas possible de citer les éventuels problèmes à résoudre au moment où j'écris ce rapport.

En conclusion, on peut dire que j'ai été bien accompagnée durant ce stage et que recevoir de l'aide extérieure a été très appréciable et décisive.

### 3) Mise en place du premier scénario

Après avoir mis en place tous les éléments nécessaires au bon fonctionnement du projet, nous avons posé nos idées, avec Robin Arnal, pour avoir une vue claire du projet final. Nous avons donc décidé certaines conventions à respecter jusqu'à la fin du projet et mis en oeuvre un premier scénario, présenté dans cette partie : dans un premier temps, l'interface, puis les spécificités techniques.

La *Figure 10 - Premier prototype de l'interface* représente une maquette de l'interface principale du vidéo-projecteur interactif. Celle-ci est partagée en trois parties générales. Sur la gauche, il y a un indicatif des performances ou des objectifs du dispositif. Au milieu, on retrouve un schéma de la chaîne permettant de suivre l'évolution des navettes et des zones de travail. Enfin, sur la droite sont présents une série de rappels généraux, concernant les stocks à réapprovisionner, les maintenances à effectuer ou autre.

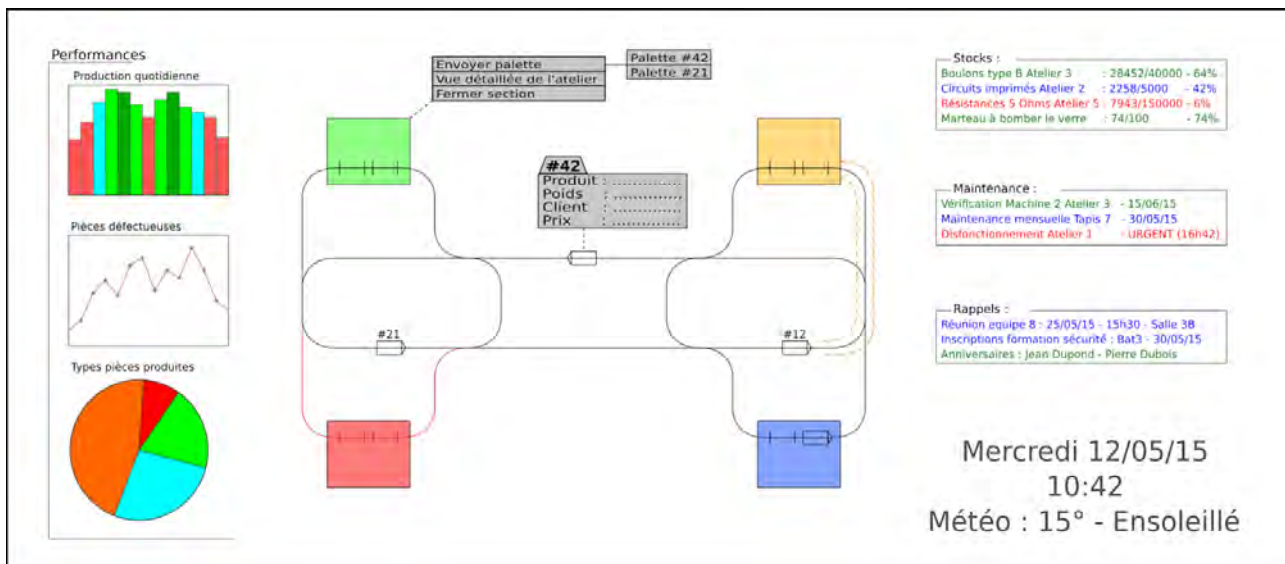


Figure 10 - Premier prototype de l'interface

Au centre du tableau ce trouve donc l'élément principal : un schéma représentant la cellule de production. Sont présentes sur le schéma les navettes identifiées par des numéros. En cliquant sur une navette, on peut voir un certain nombre d'informations la concernant (produit transporté, poids, prix, client, retard, échéance, priorité, ...). Ces informations ne sont pas statiques et peuvent différer selon le type de produit transporté. Il apparaît aussi le trajet que la navette réalise actuellement (en pointillé orange sur la Figure 10 - Premier prototype de l'interface).

Les différentes zones de travail sont représentées par des carrés de couleurs. Les couleurs ont une signification :

- la couleur verte signifie que le poste de travail est libre et inutilisé ;
- l'orange qu'une navette est en cours d'acheminement vers cet atelier ;
- le bleu que l'atelier est actuellement occupé par une ou deux navettes ;
- le rouge que l'atelier est indisponible (en panne ou en maintenance). Lorsqu'un atelier est indisponible, toute la section est indisponible, et est alors affichée en rouge.

En cliquant sur un atelier, un menu apparaît, permettant d'interagir avec des ateliers. On peut par exemple demander l'envoi d'une navette vers cet atelier, on peut voir les statistiques et performances détaillées de cet atelier, ou encore fermer la section.

Sur le côté gauche du tableau se trouvent un certain nombre de schémas décrivant la performance de la chaîne de production. On pourra par exemple trouver le nombre de pièces produites aujourd'hui, mis en comparaison avec le nombre produit la veille ainsi que l'objectif quotidien. On pourra aussi afficher la proportion de pièces défectueuses, les différents types de pièces produites, etc.

Sur le côté droit sont présents un certain nombre de rappels. On peut par exemple trouver une section de gestion des stocks (indiquant les pièces, les ateliers à réapprovisionner, le pourcentage par rapport à la capacité totale ...), une section de maintenance (indiquant les opérations à effectuer sur les ateliers ou machines et la date avant laquelle faire cette



maintenance), ou encore une section de rappels généraux (les prochaines réunions, évènements importants ou des informations plus légères, comme les anniversaires du personnel). Les différents éléments de ces sections seront soumis à un code couleur, en fonction de leur importance (rouge pour urgent, bleu pour important, vert pour correct). Sera aussi présente une zone indiquant la date et l'heure, ainsi que quelques autres informations.

Afin de récupérer des informations de la cellule, comme la position des navettes, il faudra créer des échanges avec cette dernière. Ils seront assurés par la couche logicielle ROS dans laquelle transiteront des messages à destination des deux extrémités.

Nous allons maintenant présenter le premier scénario qui est en développement au moment de la rédaction de ce rapport. S'agissant du scénario de base qui constituera nos premières expérimentations, celui-ci reste simple, tout en permettant de tester et de faire appel à un nombre conséquent de fonctionnalités. Mais, plus important, il servira de base à nos travaux futurs car c'est avec lui que nous allons choisir certaines spécificités que nous garderons jusqu'à la fin du projet. Il nous permettra aussi de voir si nos ambitions sont réalisables ou si nous allons devoir les adapter.

Dans la situation initiale, une navette circule sur l'anneau situé entre les deux zones de travail gérées par les automates Schneider. Le vidéo projecteur affiche le schéma de la maquette, ainsi que des carrés de couleurs pour les différentes zones de travail. L'utilisateur clique (avec un stylet ou son doigt) sur une zone de travail (dans un premier temps, seule une zone sera cliquable et accessible). La navette se met alors en route vers la zone de travail en question. Le carré de couleur la représentant devient orange pour signifier à l'utilisateur que la demande de trajet à bien été prise en compte et est en cours de réalisation. Une fois arrivée, la navette attend 10 secondes sur le poste de travail. Le carré de couleur devient alors bleu pour toute la durée du traitement, signifiant que le poste est actuellement occupé par une navette. Une fois le délai écoulé, la navette repart vers l'anneau pour reprendre son trajet initial. Le carré redevient alors vert, signifiant que la zone de travail est inoccupée.

Pour mettre en oeuvre ce scénario, il faut que l'ordinateur relié au tableau soit capable d'envoyer des messages, à l'instar de celui relié aux automates. Pour cela un *service* et un *topic* ROS seront créés :

- un *server* transmettant les demandes de trajets que l'on appellera "demande\_trajet" ;
- un *topic* donnant l'état de la navette en temps réel (Libre, En déplacement vers la destination, Arrivée à destination, Retour au trajet normal). Celui-ci portera le nom de "état\_navette".

Les messages transitant dans chacun d'eux doivent répondre aux mêmes caractéristiques pour avoir une certaine uniformité. Pour cela, nous nous sommes mis d'accord sur des messages types.

Pour le *service* "demande\_trajet", il y aura des messages composés d'un entier de deux chiffres. Le premier codant le numéro de l'automate dirigeant la zone de destination et le second

pour le numéro de poste (1 ou 2). Par exemple, pour se rendre au deuxième poste de travail de l'automate 1, le message envoyé sera "12". Dès que le *server* recevra une demande, il renverra soit 1, pour signifier que la demande a bien été prise en compte, soit 0 pour signifier que le déplacement n'est pas réalisable (poste de travail occupé par exemple).

Concernant le *topic* "état\_navette", nous aurons différentes possibilités. Chacun des messages est constitué de deux entiers distincts, comme expliqué ci-dessous :

- le message "1;0" indique que la navette est libre ;
- le message "2;0" indique que la navette repart d'un poste de travail ;
- le message "3;X" indique que la navette est en déplacement vers le poste de travail X ;
- le message ""4;X" indique que la navette est arrivée au poste de travail X.

Ici, X prendra des valeurs telles que "21" ou "32".

Les communications lors du déroulement du scénario s'effectueront donc de la manière suivante : dans la situation initiale, la maquette publie "1;0" dans le *topic* "etat\_navette" pour faire savoir à l'interface que la navette est libre. L'interface va envoyer une demande de déplacement via le *server* "demande\_trajet". La maquette renverra alors un entier (0 ou 1) pour indiquer si la demande a bien été prise en compte. Lorsque du trajet de la navette, la maquette publiera différents messages pour tenir compte à l'interface de l'avancée de la navette : "3;X" lors du déplacement, puis "4;X" lors de l'attente, puis "2;0" lors du déplacement vers la position initiale, et enfin "1;0" lors du retour en position initiale. La *Figure 11 - Schématisation des communications ROS* et la *Figure 12 - Représentation des différents bus et couches mis en oeuvre* illustre ce fonctionnement.

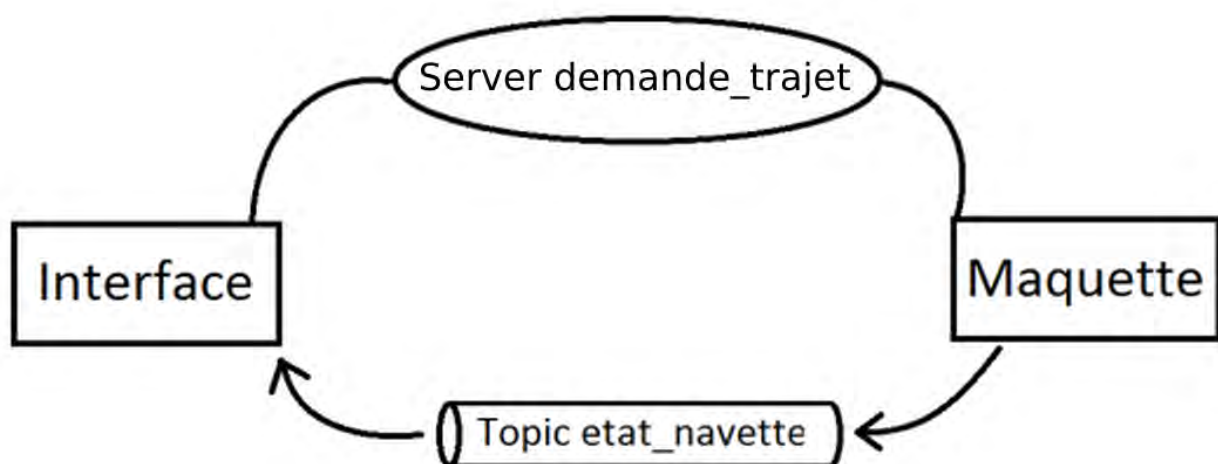


Figure 11 - Schématisation des communications ROS

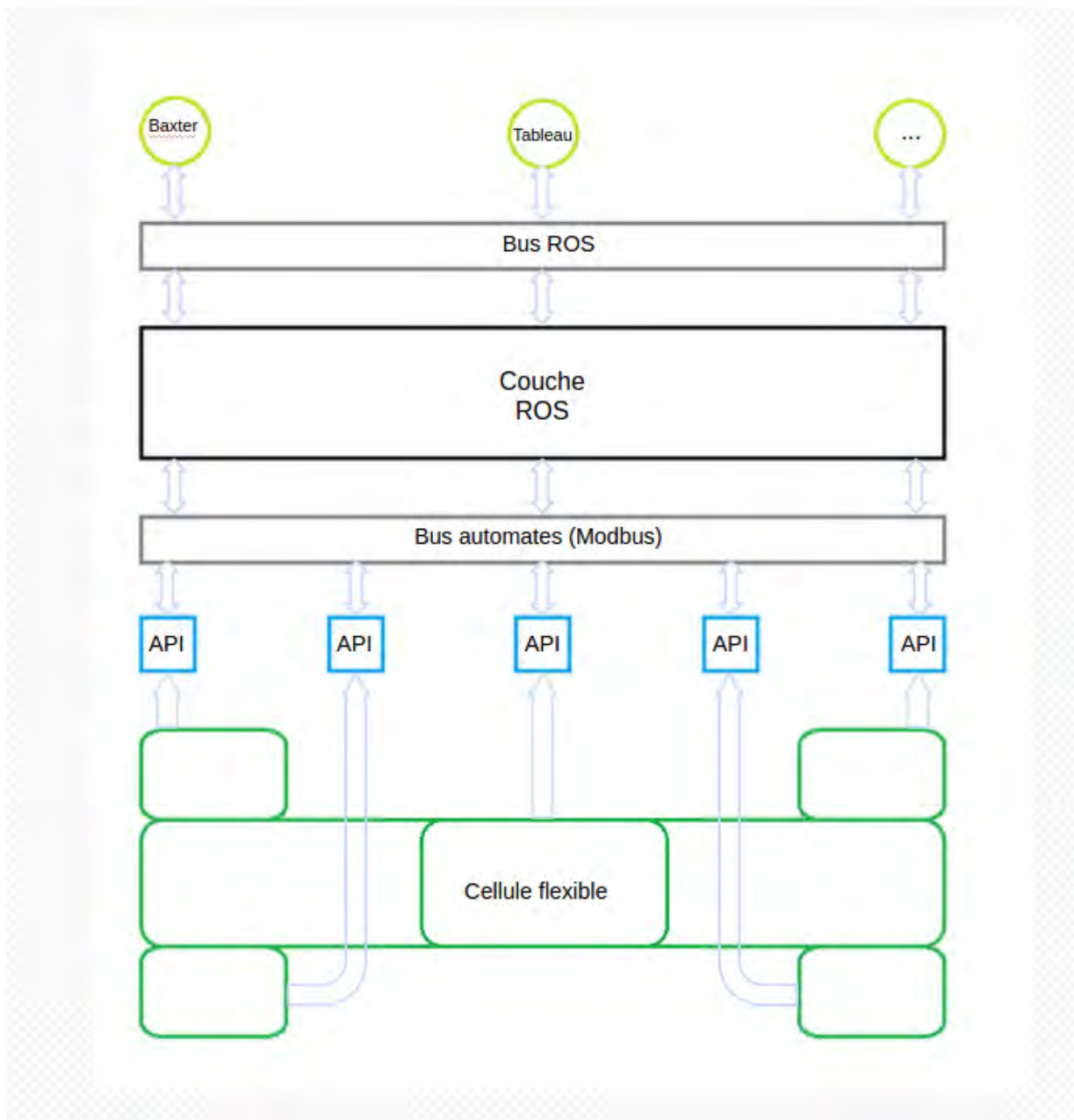


Figure 12 - Représentation des différents bus et couches mis en oeuvre

Enfin, un des acteurs clefs de ce scénario est la mise en place d'une machine à état permettant la circulation de la navette. Il faut être capable d'en faire une évolutive en fonction de la demande. Celle-ci est représentée sur la *Figure 13 - Machine à états du premier scénario*.

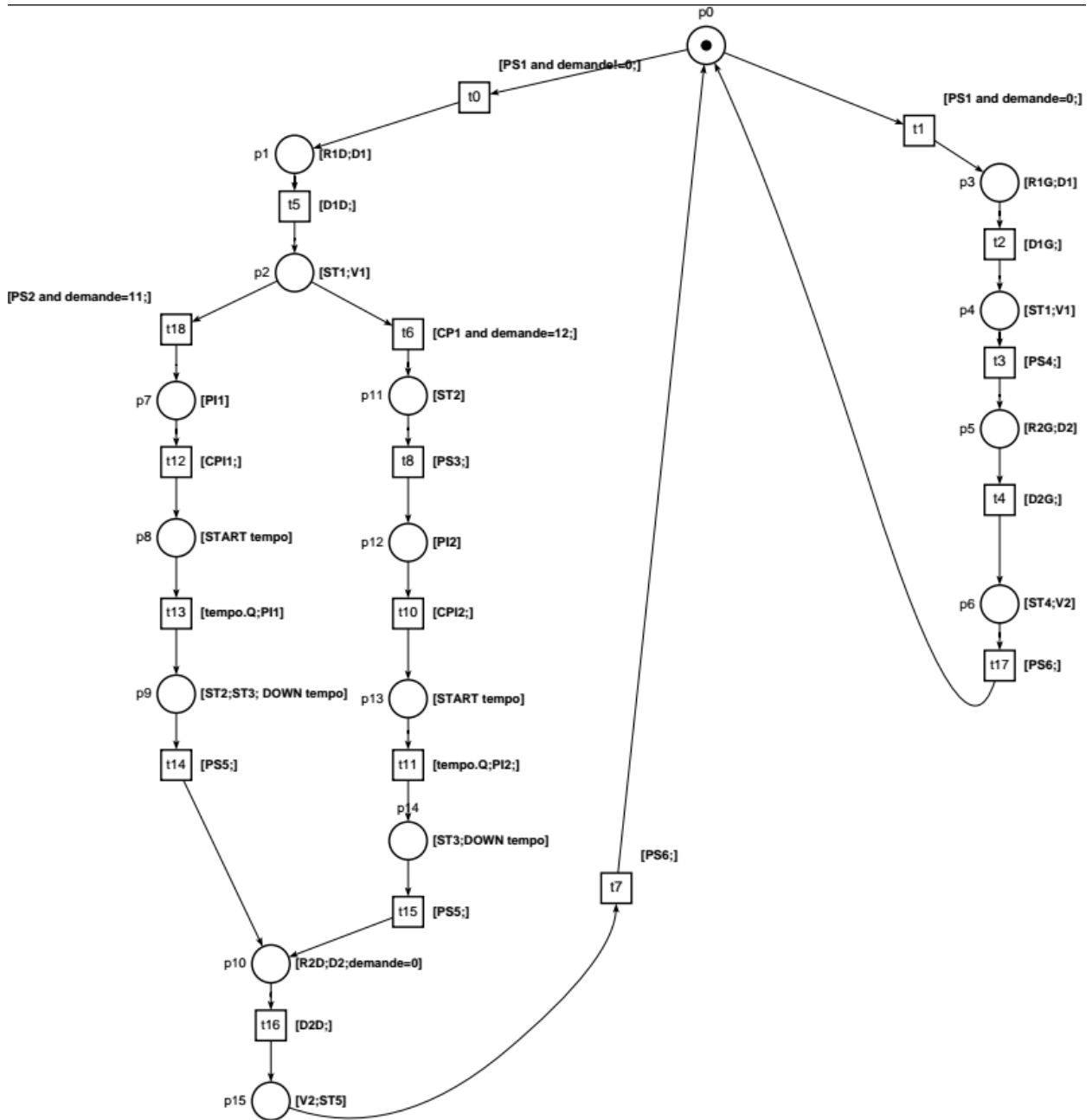


Figure 13 - Machine à états du premier scénario

Lorsque ce scénario sera fonctionnel, il sera modifié et amélioré. Les améliorations sont présentées dans la partie suivante.

#### 4) Opportunités d'amélioration pour la suite

Lorsque le premier scénario sera traité, beaucoup d'améliorations seront envisageables. Elles sont présentées dans cette partie par ordre de complexité croissante.

En premier lieu, l'étape suivante pourrait consister à faire fonctionner le même cas d'utilisation mais en intégrant un deuxième automate. Cela permettra de desservir plus de postes de travail tout en posant les bases de la communication entre plusieurs automates. Logiquement, les étapes suivantes seront d'ajouter tous les automates en prenant soin de faire communiquer ceux de marque Schneider avec ceux de marque Siemens.

Après cela, il faudrait intégrer la navigation de plusieurs navettes. Celles-ci intégreraient l'utilisation de puces RFID, des sortes de radio-étiquettes, associées aux navettes. Ainsi, on pourrait demander à une palette en particulier de rejoindre un atelier. Les identifier servira aussi à afficher la place de chacune sur l'interface, ce qui n'est pas possible dans le cas d'une utilisation avec plusieurs navettes non différenciées.

Ensuite, on pourra intégrer la vue des postes de travail sur l'interface, chacun d'eux étant équipé de caméra. Il faudra d'ailleurs faire des choix par rapport à l'interface. Il pourra être décidé d'obtenir la vue d'une zone de travail par un clic de l'utilisateur ou à des moments précis (quand une navette y est arrêtée par exemple). Ou encore, afficher les images de la caméra en grand si un problème est détecté dans une zone.

De plus, il serait très intéressant de faire interagir le robot Baxter, ou d'autres robots, avec une navette lorsque celle-ci est arrêtée dans sa zone de travail. Il pourrait poser ou prendre de la marchandise par exemple. Cela semble réalisable étant donné que le robot Baxter est programmable avec ROS.

Enfin, si on se projette plus loin, les interactions homme/machine pourraient être optimisées. Par exemple, l'utilisateur pourrait commander l'interface par les gestes. Pour cela, on peut penser à l'utilisation d'une Kinect, qui est une caméra capable de reconnaître et situer des mouvements dans l'espace. Mieux encore, les bracelets Myo utilisent l'activité musculaire et les mouvements du bras pour contrôler un programme informatique.

En conclusion, les améliorations possibles sont quasiment infinies tant ce système et les technologies contemporaines offrent de possibilités. Il semble peut-être probable que toutes soient réalisées dans notre cas mais cela prouve que, de nos jours, toute interaction avec un procédé peut être réinventée sans vraies limites, hormis celles de l'imagination humaine.

## IV) Bilans

### 1) Un stage en lien avec le DUT et ma poursuite d'études

Ce stage a été très intéressant de différents points de vue. Il a tout d'abord fait appel à de nombreuses compétences développées dans le DUT Génie Électrique et Informatique Industrielle.

Premièrement, je me suis servi de ma connaissance du C et du C++, langages utilisés tout le long du cursus de DUT. Ensuite, j'ai eu recours à l'utilisation de Grafcet et machines à états, modèles de commande très présents dans cette formation ainsi que dans le stage.

De plus, j'ai réinvesti des connaissances apprises dans des cours tels que le Réseau ou l'Ethernet Industriel. Parmi elles, on peut compter les protocoles de communication à l'instar du TCP/IP. Je peux également citer la connaissance des commandes Unix, ou du moins la manière d'évoluer dans une console.

Ensuite, j'ai pu mettre en oeuvre mes compétences dans la gestion de projet. En effet, à l'IUT chaque projet induisait la création de différents livrables tels que des plannings et des cahiers des charges. Ces notions m'ont beaucoup servi lorsque j'ai dû fournir les miens durant ce stage. Cependant, j'ai pu compléter cela avec la rédaction d'un plan de développement.

En plus de réutiliser des compétences, j'en ai développé de nouvelles. Notamment, j'ai découvert l'outil ROS qui je pense me sera très utile par la suite. En effet, la formation que j'aimerais intégrer à la suite de ce DUT a recours à l'utilisation de ce middleware. Avoir travaillé sur cet outils pendant ce stage ne pourra être qu'un atout si cette école retient ma candidature. De plus, là encore, je peux citer l'élaboration d'un plan de développement qui semble être une pratique récurrente dans les études supérieures.

L'école à laquelle j'ai postulé et qui, je l'espère vivement, me comptera l'année prochaine parmi ses étudiants est l'UPSSITECH avec son programme Systèmes Robotiques et Interactifs. Il faut savoir que cette école fait partie de l'Université Paul Sabatier. Comme expliqué précédemment dans ce rapport, l'Université Paul Sabatier est un des partenaires à la tête de l'AIP-PRIMECA. Avoir fait ce stage dans cette structure a je pense mis toutes les chances de mon côté. Les professeurs que j'y ai rencontrés auront peut-être la possibilité de faciliter mon admission. Je pense avoir eu une chance inestimable d'évoluer dans ce milieu pendant 3 mois et je mesure ce privilège qui m'a été offert.

### 2) Un avis critique sur la méthode de travail

Dans cette partie, j'ai décidé de faire un retour sur la méthode de travail adoptée durant ce stage et présentée dans le II) 3) *Une méthode de travail particulière* de ce rapport.

Cette méthode, visant à faire des réunions, à deux semaines d'intervalle, durant lesquelles le travail de la période se terminant était examiné et où des objectifs pour la suivante étaient fixés, était relativement particulière et je reste assez partagée à son sujet.

D'une part, je l'ai trouvée très intéressante et assez efficace. Le fait d'avoir des objectifs m'a beaucoup motivée. Les réunions étaient très stimulantes car l'envie d'avoir le plus de contenu à présenter était bien réelle. De plus, le fait d'être suivis dans nos travaux était très plaisant parce que cela leur accordait une véritable importance. Enfin, la rédaction de livrables tels que le cahier des charges et le plan de développement étaient vraiment un atout. En effet, le fait de mettre en oeuvre une réflexion poussée sur le travail à réaliser est, d'après moi, un passage inévitable. Je suis d'avis de dire qu'il vaut mieux analyser une situation en détails avant de se plonger dans le travail. Je juge donc cela comme quelque chose qui m'a beaucoup aidé dans mon stage car même si j'avais voulu le faire de ma propre initiative, je n'aurais peut-être pas eu de retours aussi constructifs par mes maîtres de stage sur les documents produits ; et les réunions sont avant tout un lieu d'échange et de rencontre.

Par ailleurs, j'ai trouvé l'encadrement que j'ai eu assez "scolaire" au début. Il me semble que le stage en entreprise est l'occasion d'être confronté au monde du travail et je me demande si, en entreprise, j'aurais rencontré un contexte similaire. Concrètement, j'ai trouvé étrange, bien que formateur, qu'il soit à la charge du stagiaire de rédiger lui-même un cahier des charges sur le travail à effectuer. Il me semble qu'en général, les besoins auxquels doit répondre le stagiaire sont préalablement identifiés par le client. Néanmoins, je réalise aujourd'hui que mes maîtres de stage m'ont témoigné leur confiance en me donnant cette mission. De plus, je me rends compte que si je réalise au moins trois ans d'études supplémentaires, ces pratiques seront choses courantes de mon quotidien lors de ma vie professionnelle. Malgré un premier regard mitigé, je reconnais finalement que cette expérience a été une véritable chance.

### 3) L'influence sur mon projet professionnel et la journée de la robotique

Comme évoqué dans la partie précédente, je souhaite poursuivre mes études à l'UPSSITECH. La formation que je souhaite rejoindre traite de robotique. Grâce à ce stage, j'ai pu développé de nombreuses compétences relatives à ce domaine. Cela a confirmé mon envie de poursuivre dans cette voie. En effet, j'ai approfondi ma connaissance de ce milieu qui n'était jusqu'alors que très limitée.

Notamment, j'ai pu apprendre beaucoup sur la robotique lors d'une journée organisée à l'AIP-PRIMECA le 21 mai 2015. Il s'agissait de la "Journée découverte - Robotique Industrielle et PME" proposée par l'organisme Madeeli. Madeeli signifie Mon Agence du Développement Economique, de l'Export et de l'Innovation en Midi-Pyrénées et anime de nombreux événements comme celui-ci.

Cette journée avait pour but de réunir toute sorte de personnes ayant un lien avec la robotique et plus particulièrement celles évoluant dans des PME. L'enjeu était de montrer que la robotisation est accessible à tous, même aux plus petites entreprises.

Lors de cette journée, j'ai eu la chance d'assister à différentes conférences concernant ce sujet. Parmi elles, des patrons de PME racontaient leur expérience dans l'achat d'un robot. Cela a été vraiment constructif et j'ai pu voir de plus près le monde du travail dans ce domaine.

Ensuite, en dehors de ces présentations, j'ai pu échanger avec toutes ces personnes. Dialoguer avec eux a été une expérience très positive car ce n'est pas une occasion qui se présente fréquemment lorsque nous sommes étudiants. En effet, le monde du travail nous semble vraiment loin de notre quotidien alors que nous nous y rendons à grand pas. En plus d'avoir découvert la vie dans une entreprise précise lors de ce stage, ces rencontres m'ont donné un aperçu de ce qu'il peut se passer dans d'autres sociétés.

Ce jour-là, j'ai assisté à la présentation du matériel technique de l'AIP-PRIMECA. J'ai découvert les enjeux de posséder de tels équipements compte tenu de l'état du marché de nos jours, des innovations actuelles ou encore des possibilités offertes par ces dispositifs.

En conclusion, ce stage a eu un impact bénéfique sur mon projet professionnel. J'ai pu découvrir plus en profondeur un milieu déjà très attrayant à mon goût. Les expériences que j'ai pu vivre lors de ce stage auront certainement été déterminantes pour ma vie future et je pense que c'est une réelle chance d'avoir eu ces opportunités.



## Conclusion

En définitive, je retiens beaucoup de positif de ce stage. La majorité des situations que j'ai rencontrées m'ont appris de nouvelles choses, qu'elles aient un aspect technique, relatif au monde du travail ou même humain.

Durant ces 12 semaines, j'ai pu m'atteler à un projet dans lequel j'ai mis en oeuvre des compétences développées pendant mon DUT. Néanmoins, je pense que celles acquises pendant ce stage me seront d'une aide précieuse pour la suite.

Je retiendrai de cette période la persévérance dont j'ai fait preuve face à de nombreux problèmes mais aussi la maturité que j'ai gagnée face à ces situations.

Quant à l'avancement du projet, il m'est impossible de dire que ce dernier est abouti car peu de scénarii de tests ont été mis en oeuvre à l'heure à laquelle est rédigé ce rapport. J'espère sincèrement que le travail que j'ai fourni saura être réutilisé pour mener à bien ce projet, qui je pense, a un bel avenir dans les locaux de l'AIP-PRIMECA.

# Annexes

Annexe I : Table des figures

Annexe II : Diagramme de Gantt

Annexe III : Offre de stage

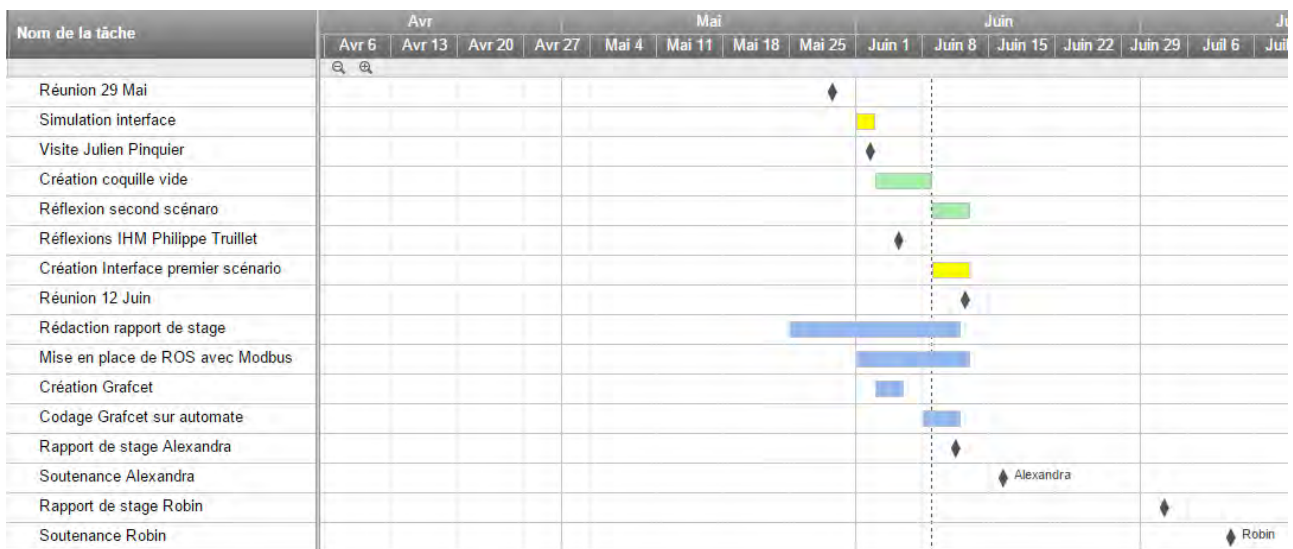
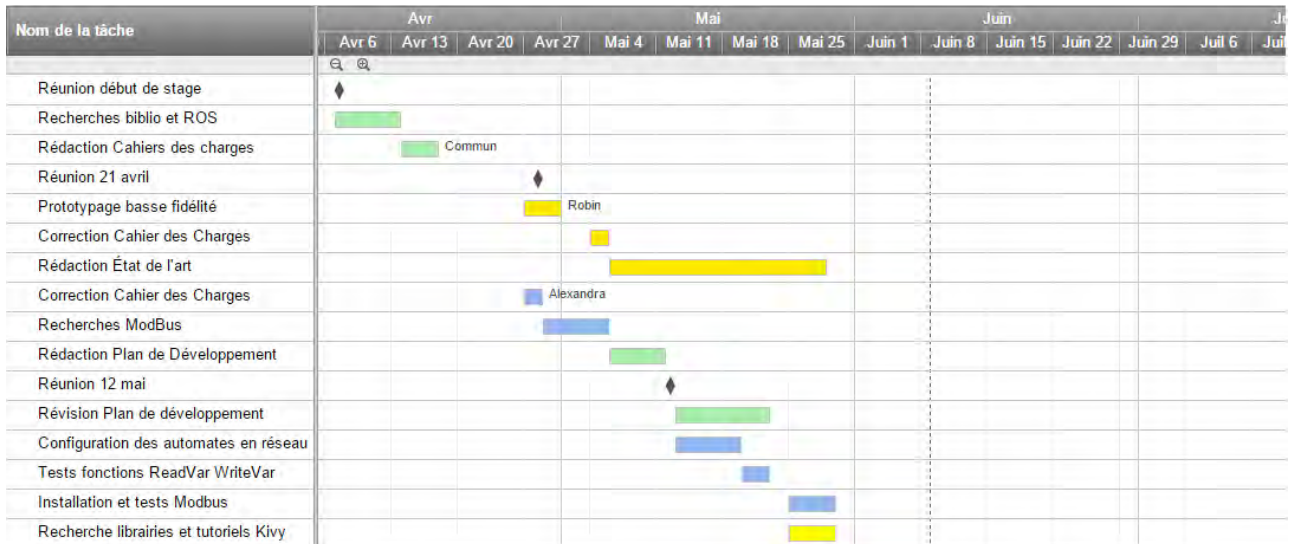
Annexe IV : Programme de la Journée Découverte de la Robotique Industrielle

Annexe V : Fiche de bilan et d'archivage



## Tables des figures

|                                                                        |       |
|------------------------------------------------------------------------|-------|
| Figure 1 - Matériel de l'AIP-PRIMECA                                   | p. 5  |
| Figure 2 - Schéma de la cellule flexible                               | p. 13 |
| Figure 3 - Vidéo-projecteur interactif Epson EB-595 WI                 | p. 14 |
| Figure 4 - Schéma de la cellule flexible avec capteurs et actionneurs  | p. 16 |
| Figure 5 - Signification des capteurs et actionneurs                   | p. 16 |
| Figure 6 - Grafcet de fonctionnement d'un automate                     | p. 17 |
| Figure 7 - Code d'un subscriber en C++                                 | p.18  |
| Figure 8 - Code d'un publisher en C++                                  | p. 19 |
| Figure 9 - Description de la fonction "read_war"                       | p. 20 |
| Figure 10 - Premier prototype de l'interface                           | p. 23 |
| Figure 11 - Schématisation des communications ROS                      | p. 25 |
| Figure 12 - Représentation des différents bus et couches mis en oeuvre | p. 26 |
| Figure 13 - Machine à états du premier scénario                        | p. 27 |

# Diagramme de Gantt



## Offre de stage



PROPOSITION DE STAGE  
2015

1

|                        |                                                                                                                                     |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Service                | Université Paul Sabatier – Atelier Inter-universitaire de Productique – Pole de ressources Informatiques en Mécanique (AIP-PRIMECA) |
| Maître de stage        | Cyril Briand, Sylvain Durola et Laurent Houssin                                                                                     |
| Contact                | <a href="mailto:briand@laas.fr">briand@laas.fr</a> - 05 61 33 78 18                                                                 |
| Lieu                   | AIP-PRIMECA, Halle Technologique – ESPE, UPS, 118 Route de Narbonne, Toulouse                                                       |
| Domaine                | Informatique industrielle, Robotique, Automatismes                                                                                  |
| Durée                  | 4 à 6 mois – À partir de mars 2015                                                                                                  |
| Rémunération           | Gratification de stage (500,51€/mois)                                                                                               |
| Niveau d'étude         | Licence Pro - Bac +4/5 Ecoles et Universités                                                                                        |
| Compétences souhaitées | Programmation C/C++/Python, Grafcet, Unity Pro, Réseaux de Petri.                                                                   |

### **Titre : Commande d'une cellule flexible de production robotisée**

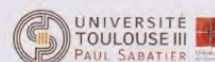
Ce stage s'intéresse au domaine de l'automatisation des systèmes de production et de la robotique. Il concerne la cellule flexible Montech® disponible au pôle AIP PRIMECA (Halle technologique - ESPE, Université Paul Sabatier). Celle-ci est constituée d'un anneau de transport desservant 4 postes de travail robotisés. Le système de transport est constitué d'un réseau monorail en aluminium, de chariots motorisés autonomes (navettes), d'un système modulaire de capteurs de présence et d'actionneurs liés aux aiguillages, d'automates programmables de marques Schneider Télémécanique et Siemens.

L'objectif du stage est de concevoir une commande modulaire du dispositif permettant une intégration des postes robotisés et du système de transport. Le projet se déroulera selon une méthode agile (SCRUM) consistant à intégrer les fonctionnalités pas à pas, en privilégiant la livraison régulière d'un logiciel opérationnel selon des cycles de quelques semaines.





# Programme de la Journée Découverte de la Robotique Industrielle



## Journée découverte Robotique industrielle & PME

jeudi 21 mai 2015 à Toulouse  
Pôle AIP/Primeca, ESPE de Toulouse Rangueil - 118 route de Narbonne

La robotique industrielle n'est pas réservée aux grandes entreprises. PME, la nouvelle agence régionale du développement économique, de l'export et de l'innovation **Madeeli**, le **pôle AIP/PRIMECA** de l'Université Paul Sabatier et le cluster **Robotics Place** proposent aux PME de découvrir pourquoi et comment utiliser des robots au sein de leur outil de production.

### Programme :

8h30 Café et accueil des participants

9h **Ouverture de la journée par Bernard Plano, Conseiller régional et Président Délégué de Madeeli**

9h30 **Pourquoi intégrer un robot ?**

**Introduction par le professeur Gabriel Colletis – UTS-LEREPS**

**Démystifier la robotique industrielle par quelques éléments sociaux-économiques**

**Un robot dans le cadre de la stratégie industrielle d'une PME par Dominique Naud – ADTronik**

**Témoignages d'industriels utilisateurs et intégrateurs de robots**

- Alexandre Vidal, société Vidal
- Thierry Orgue, société Solev, groupe Pochet
- Xavier Barthou, société SPIE
- Pierre Rouch, société Aviacomp
- Jérémie Pedros, société Actemium

**Comment intégrer un robot ?**

Présentation de Madeeli, Christophe Nicot

Présentation du cluster Robotic Place, Laurent Latorse

Présentation de la plateforme AIP/PRIMECA et de l'Offre formation robotique UPS, Cyril Briand

12h45 **Repas et Stands des acteurs régionaux de la robotique (Intégrateurs, cluster, acteurs de la formation)**

*Halle technologique ESPE, site Toulouse Rangueil de l'Université Paul Sabatier*

14h **Visite de la plateforme robotique-productive AIP/PRIMECA**

**Présentation du plan régional Usine du Futur par Nathalie Nouzies – Conseil Régional et Hervé Serres -Madeeli**

**Quelques exemples de robotique collaborative pour applications industrielles par Michel Devy – LAAS-CNRS**

16h Fin de la journée

## Fiche de bilan et d'archivage

N° d'archivage   
(ne pas remplir)

### FICHE DE BILAN ET D'ARCHIVAGE DU RAPPORT DE STAGE

Cette fiche, disponible sur le site web du département dans la rubrique « Espace Etudiants/Personnels GEII » est à remplir par chaque étudiant(e) et doit être :

- **Envoyée obligatoirement par mail à échéance du 8 juin 2015 :**
  - à votre enseignant de culture et communication pour évaluation
  - à [cathy.ibanez@iut-tlse3.fr](mailto:cathy.ibanez@iut-tlse3.fr) pour archivage

**NOM de l'étudiant(e) :** Alexandra GORRY POLLET

**Titre du rapport :** Commande d'une cellule flexible de production robotisée

**A - Nom de l'entreprise** (expliciter les sigles)  
AIP PRIMECA (Atelier Inter-Etablissements de Productique)

**Adresse :**  
Université Paul Sabatier  
Halle Technologique IUFM  
118 route de Narbonne  
31078 TOULOUSE Cedex 4

**Domaine(s) d'activité(s) :**  
Mécanique, productique, robotique, informatique

#### **B - Critères de classement de stage** \* Cocher les cases correspondantes

| <b>Lieu*</b>                                   | <b>Type*</b>                                                          | <b>Spécialité*</b>                                                    |
|------------------------------------------------|-----------------------------------------------------------------------|-----------------------------------------------------------------------|
| 1 <input checked="" type="checkbox"/> Toulouse | 1 <input checked="" type="checkbox"/> Etude                           | 1 <input type="checkbox"/> Automatique                                |
| 2 <input type="checkbox"/> Midi-Pyrénées       | 2 <input type="checkbox"/> Production                                 | 2 <input type="checkbox"/> Electronique analogique                    |
| 3 <input type="checkbox"/> Métropole           | 3 <input type="checkbox"/> Maintenance                                | 3 <input type="checkbox"/> Electronique de puissance                  |
| 4 <input type="checkbox"/> DOM-TOM             | 4 <input type="checkbox"/> Intégration                                | 4 <input type="checkbox"/> Traitement du signal                       |
| 5 <input type="checkbox"/> Etranger            | 5 <input type="checkbox"/> Autre : (préciser)<br><input type="text"/> | 5 <input type="checkbox"/> Hyperfréquences                            |
|                                                |                                                                       | 6 <input checked="" type="checkbox"/> Informatique industrielle       |
|                                                |                                                                       | 7 <input checked="" type="checkbox"/> Réseaux                         |
|                                                |                                                                       | 8 <input checked="" type="checkbox"/> Développement logiciel          |
|                                                |                                                                       | 9 <input type="checkbox"/> Autre : (préciser)<br><input type="text"/> |

#### **C - Informations diverses sur le stage**

Offre d'emploi prolongeant le stage ☐ OUI ☒ NON  
Si oui type et durée :

## **D - Travail effectué par le (la) stagiaire**

### **1- Thème du stage**

**Libellé technique de l'objet du stage** (400 caractères (espaces compris) maximum sur 4 lignes maximum)

Développer une couche logicielle permettant le dialogue entre des automates programmables et des applications informatiques pour commander un système de transport desservant des postes de travail robotisés. L'objectif est de contrôler le système de transport à l'aide un vidéo projecteur interactif via une couche logicielle développée avec le middleware ROS (Robot Operating System) en langage C.

**Libellé vulgarisé et explicatif** (400 caractères (espaces compris) maximum sur 4 lignes maximum)

Contrôler un système de transport sur rails grâce à des interactions humaines avec une interface représentant ce système sur un tableau "tactile".

### **2- Synthèse du travail effectué**

*Présentation de la démarche d'analyse et de traitement du thème, bilan des résultats*

**Résumé** (1100 caractères (espaces compris) maximum sur 10 lignes maximum)

Dans un premier temps, une découverte des outils qui allaient servir au traitement du sujet a été réalisée (middleware ROS, logiciel PL7 de programmation des automates ...). Une fois qu'une vue globale des possibilités offertes par ces outils a été construite, la communication entre les automates a été réalisée. L'objectif est de permettre à chacun d'entre eux de recevoir dans sa mémoire des informations provenant d'autres automates mais aussi d'en envoyer, en passant uniquement par le réseau. L'étape suivante consiste à créer une couche logicielle permettant le transit des messages entre le groupe d'automates et le tableau interactif. Les scénarii à réaliser consistent à commander le déplacement des navettes, leur immobilisation ou encore connaître leur emplacement.

**Mots clés** (5 maximum / 200 caractères (espaces compris) / 2 lignes maximum)

ROS, automates programmables, tableau interactif, interactions homme/machine, programmation

**Abstract** (1100 characters (including spaces) in a maximum of 10 lines)

In a first place, I had to discover the tools that will permit to answer to my internship's objectives (the middleware, the software PL7 for the automates programming ...). Then, thanks to a global view of these tools' possibilities in the realisation of my work, I had to resolve the problems of communication between the programmable controllers, that is to say to permit to each one to receive in its memory informations from the other controllers but also to send them, just using the network. The next stage is to create a software layer on which the messages between the controllers and the interactive board will transit. The scenarios that will be made will consist of asking shuttles' displacements, immobilize them or else of knowing their movements on the rails.

**Key-words** (5 maximum / 200 characters (including spaces) / 2 lines maximum)

ROS, programmable controllers, interactive board, human/computer interactions, programming



### 3- Analyse des compétences mises en œuvre

A partir des principales activités exercées, explicitez dans le tableau ci-dessous les compétences mobilisées (ce tableau ne doit pas excéder 1 page).

| Activités                                                               | Disciplines<br>(enseignées ou non enseignées au département GEII) | Compétences <ul style="list-style-type: none"> <li>- Connaissances utilisées : matériel, outils, logiciels....</li> <li>- Savoirs faire mobilisés</li> <li>- Qualités/aptitudes personnelles requises</li> </ul>                                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------|-------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Activité 1 :</b><br>Rédaction d'un cahier des charges                | Gestion de projet                                                 | Connaissances : utiliser un traitement de texte, rôle d'un cahier des charges, son contenu<br>Savoir faire : élaborer le cahier des charges en fonction d'un délai et d'un volume horaire, élaborer un planning / diagramme de Gantt, identifier les tâches nécessaires au bon fonctionnement du projet<br>Qualités/aptitudes : savoir rédiger un texte, organiser ses propos, synthétiser                                                                    |
| <b>Activité 2 :</b><br>Programmation des automates avec le logiciel PL7 | Programmation, réseau                                             | Connaissances : fonctionnement du logiciel PL7, savoir configurer un réseau d'automates, Grafcet<br>Savoir faire : élaborer un Grafcet (notion d'étapes, de transitions ...), programmer un Grafcet avec PL7, tester le fonctionnement sur site<br>Qualités/aptitudes : s'adapter au matériel, identifier les comportements dangereux                                                                                                                         |
| <b>Activité 3 :</b><br>Mise en place de la couche logicielle avec ROS   | Programmation, réseau                                             | Connaissances : fonctionnement de ROS, fonctionnement du protocole TCP/IP, langage C, environnement Ubuntu et commandes Unix, rechercher et identifier des fonctions existantes utiles<br>Savoir faire : création de l'espace de travail ROS et de ses composantes, programmation en C, installation de librairie (Modbus en l'occurrence), savoir utiliser des fonctions déjà existantes<br>Qualités/aptitudes : faire des choix logiques sur la réalisation |
| <b>Activité 4 :</b>                                                     |                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Activité 5 :</b>                                                     |                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

**E - Schémas ou Photos illustrant votre stage :** (pas plus d'une page).

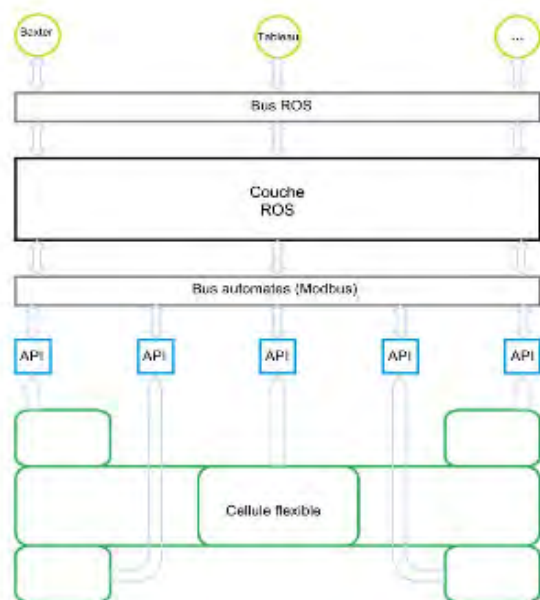


Schéma des différentes couches permettant le fonctionnement du système



Photo de la cellule flexible