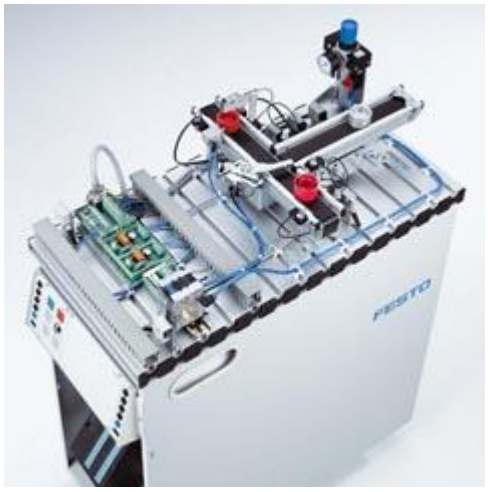
POLYTECH LILLE

Université de Lille 1 SCIENCES ET TECHNOLOGIES

Projet : Conception d'une MPS Polyvalente



BELHOUACHI Samy et COCKENPOT François-Xavier
TUTEURS : | M. MERZOUKI | M. DANEL | M. VERGEZ

IMA4 2016 - 2017

Remerciements

Nous tenons à remercier toute l'équipe pédagogique IMA ainsi que l'ARPL et plus particulièrement M. Danel, M. Vergez et M. Merzouki pour nous avoir conseillé durant toute notre période de projet et pour avoir répondu à nos questions. De plus nous remercions M. Redon pour avoir réalisé un suivi durant cette période de projet et pour sa disponibilité tout au long des séances de projet.

Introduction

Dans le cadre de notre formation, IMA (Informatique Microélectronique Automatique), nous devons réaliser un projet en quatrième année. Celui-ci ayant pour but de nous familiariser et de donner un avant-goût de l'environnement professionnel.

Lors du semestre précédent, notre binôme constitué de BELHOUACHI Samy et de COCKENPOT François-Xavier avons décidé de choisir le projet suivant « Conception d'une MPS polyvalente ». Ce projet fait appel à de l'électronique et à de l'informatique, nous avons donc réutilisé de nombreuses choses étudiées dans les semestres précédents.

L'ARPL (Association de Robotique de Polytech Lille) participe chaque année à la RoboCup compétition internationale de robotique et plus particulièrement dans la Logistics League. Pour s'entraîner pour la compétition il faut 4 MPS (Modular Production System) mais pour des raisons financières ils ne possèdent que des bases de MPS. C'est pourquoi nous devions essayer de trouver une solution pour qu'ils puissent s'entraîner. Ainsi en essayant de résoudre les problèmes éventuels, nous allons essayer de répondre aux questions suivantes :

Quelles solutions sont les plus judicieuses ? Comment procéder pour simuler 4 MPS différentes ?

Dans un premier temps, nous présenterons notre projet et expliquerons les choix vers lesquels nous nous orienterons. Ensuite dans un second temps, nous traiterons la partie conceptions et réalisations. Puis, nous expliquerons qu'avons-nous réalisé informatiquement. Enfin, nous nous attarderons sur les différents problèmes rencontrés et les perspectives de ce projet.

Table des matières

I/ Présentation générale du projet et solutions

Dans cette partie, nous allons vous présenter de façon générale notre projet et allons expliquer les différentes solutions choisies.

I.1/ Présentation du sujet et contexte

Lors du semestre 8 de la formation IMA nous avons réalisé le projet suivant « Conception d'une MPS polyvalente ». L'ARPL ne possède pas de MPS et voudrait pouvoir s'entraîner avant la compétition internationale de la RoboCup dans la Logistics League et tester les différentes combinaisons de construction. Le but de la Logistics League est de trouver des solutions robustes pour l'ajout de robots dans un environnement industriel.

Pour commencer, il faut savoir qu'une MPS (Modular Production System) est une machine permettant de construire une pièce. Il existe 4 MPS différentes qui n'ont pas la même pièce à fabriquer. Ces 4 types de MPS sont :

- Base Station
- Delivery Station
- Ring Station
- Cap Station

Figure 1: Modular Production System de Festo

Nous savons que pour reconnaitre une machine, les différents robots détectent grâce à l'AR-Code la fonctionnalité de la machine. Ces AR-Codes sont imprimés et sont collés sur la face et l'arrière de chaque MPS. Nous devons donc rechercher une solution pour que les AR-Codes ne soient pas figés ou fixés. De plus nous devons aussi gérer le feu tricolore et le convoyeur.

I.2/ Solutions

Dans un premier temps, nous avons réalisé une étude de l'environnement. Et nous nous sommes attardés sur 3 grands axes :

- La gestion du convoyeur
- La gestion du feu tricolore
- Et la recherche de solutions concernant la détection des AR-Codes

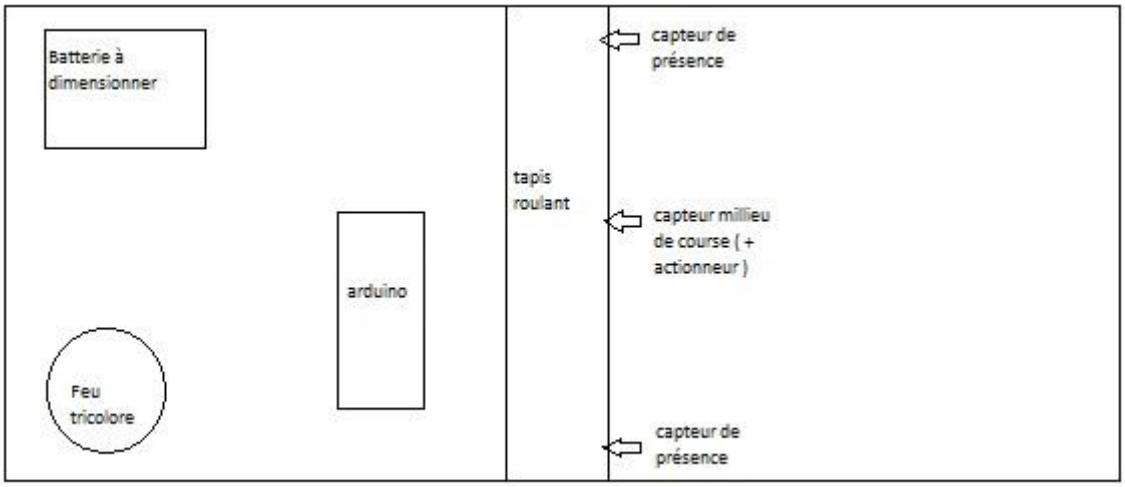


Figure 2: Vue du dessus de la MPS

Ainsi, nous voulons gérer le déplacement du convoyeur à savoir les deux types de déplacement :

- Déplacement avant
- Déplacement arrière

Le but du convoyeur est d'amener une pièce en début ou en fin de zone de production. Ensuite nous devons gérer les différents états du feu tricolore :

- Feu vert → En production
- Feu rouge → En arrêt
- Feu rouge et orange clignotant → en phase de détection

Concernant la gestion des AR-Codes, nous allons rechercher plusieurs solutions, et choisir la meilleure d'entre elles.

1) Nous voulions dans un premier temps mettre des tablettes électroniques pour ensuite changer les différents codes pour ensuite la manipuler avec une arduino.

2) Puis nous avons pensé à acheter deux matrices de leds 5*5 pour gérer les différentes leds permettant de générer le code souhaité.

3) Construire nous-même la matrice de leds pour ensuite la gérer de la même manière que la solution 2.

Parmi ces trois solutions la première était trop élevée financièrement c'est pourquoi nous ne pouvions pas la choisir. Nous n'avons pas pris la seconde solution car les leds sont trop proches et le robot ne détecterait pas le code représentait par cette matrice de leds. C'est pourquoi nous avons décidé de prendre la dernière pour choisir l'écartement de nos leds. Voici l'endroit où nous mettrons notre matrice de leds sur la MPS.

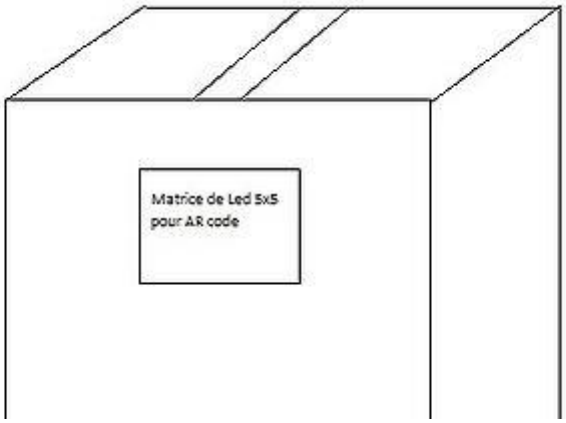


Figure 3: Matrice de leds sur la MPS

Problème : Nous voulions tout gérer avec l'Arduino mais ce n'est pas possible car le moteur du tapis roulant et le feu tricolore sont alimentés en 24V.

Recherche de solutions : Utilisation de relais statiques facilitant la gestion du tapis roulant et du feu tricolore par l'Arduino.

On utilisera un transistor pour contrôler des relais en 24V ce qui permettra d'avoir des relais plus résistants en courant et en tension.

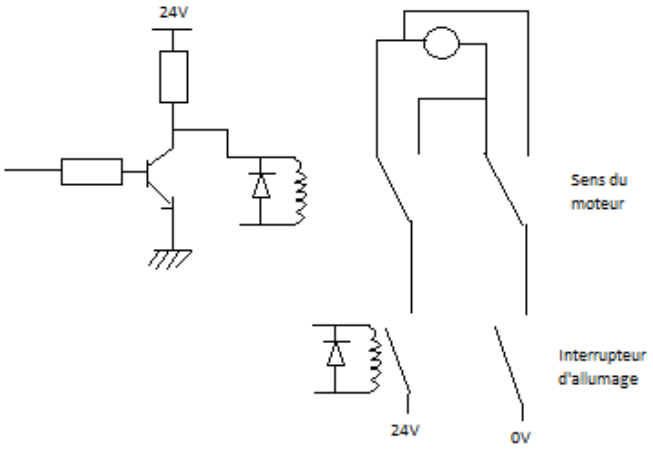


Figure 4:Schéma de câblage pour la gestion du tapis roulant bidirectionnel

Pour éviter les cas où l'on alimente par erreur le moteur dans les deux sens, nous avons pensé au câblage précédent qui, avec deux relais statiques différents, permet de gérer le sens du tapis roulant pour le premier cas et d'allumer ou non le tapis roulant pour le deuxième.

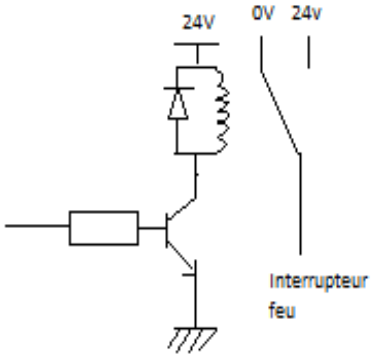


Figure 5: Schéma de câblage pour une couleur de feu

Pour la partie capteur, nous allons utiliser les capteurs robotino. Ce sont des capteurs alimentés en 24V. De plus, dans le cas où la tension de sortie est de 24V cela signifie qu'il y a une détection d'objet, sinon dans le cas échéant nous avons une tension de sortie égale à 0V. Les entrées de l'arduino étant alimentés en 5V maximum, nous choisissons alors d'utiliser un transistor pour abaisser la tension.

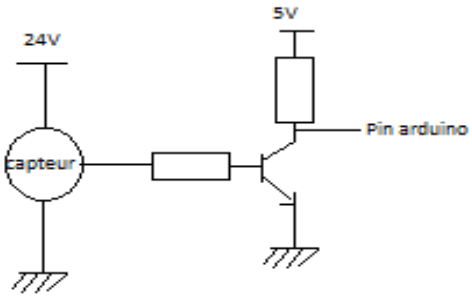


Figure 6: Schéma de câblage pour capteur Robotino

II/ Conceptions et réalisations

Tout au long de cette partie, nous allons expliquer les différentes conceptions et réalisations que nous avons réalisées.

II.1/ Circuits électroniques

Après avoir décidé de réaliser une matrice de leds, nous avons choisi d'utiliser le logiciel Altium Designer. Nous avons alors commencé par rechercher les valeurs des résistances de notre matrice. Puis nous avons effectué le « schematic » que nous pouvons observer à la figure suivante.

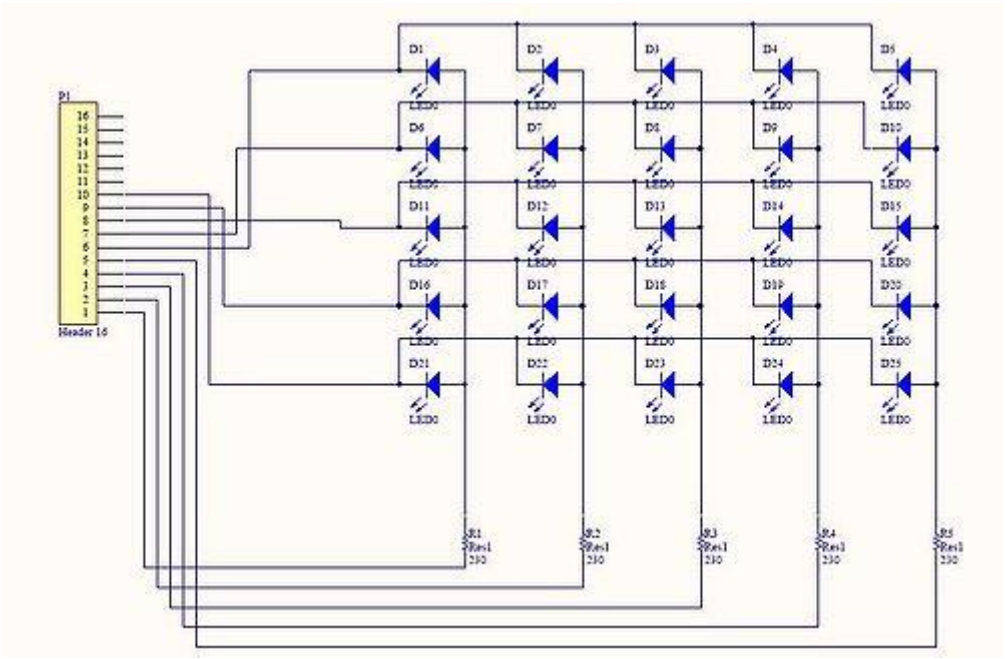


Figure 7: Schematic sur Altium Designer

De plus, une fois avoir fait notre « schematic », nous nous sommes attardés sur le routage de la carte. Nous avons d'abord dimensionné notre carte 80 mm X 80 mm, et avons espacé les composants de 1 cm entre eux pour obtenir les AR-Codes utilisés durant la compétition. Après quelques vérifications nous avons pu constater qu'il n'y avait plus de problèmes. Nous pouvons observer à la figure suivante notre routage.

Malgré la réalisation de la conception de la matrice de leds, nous ne sommes pas allés l'imprimer après une discussion avec nos tuteurs qui préféraient avoir un code fonctionnel et la conception de la carte pour un futur proche.

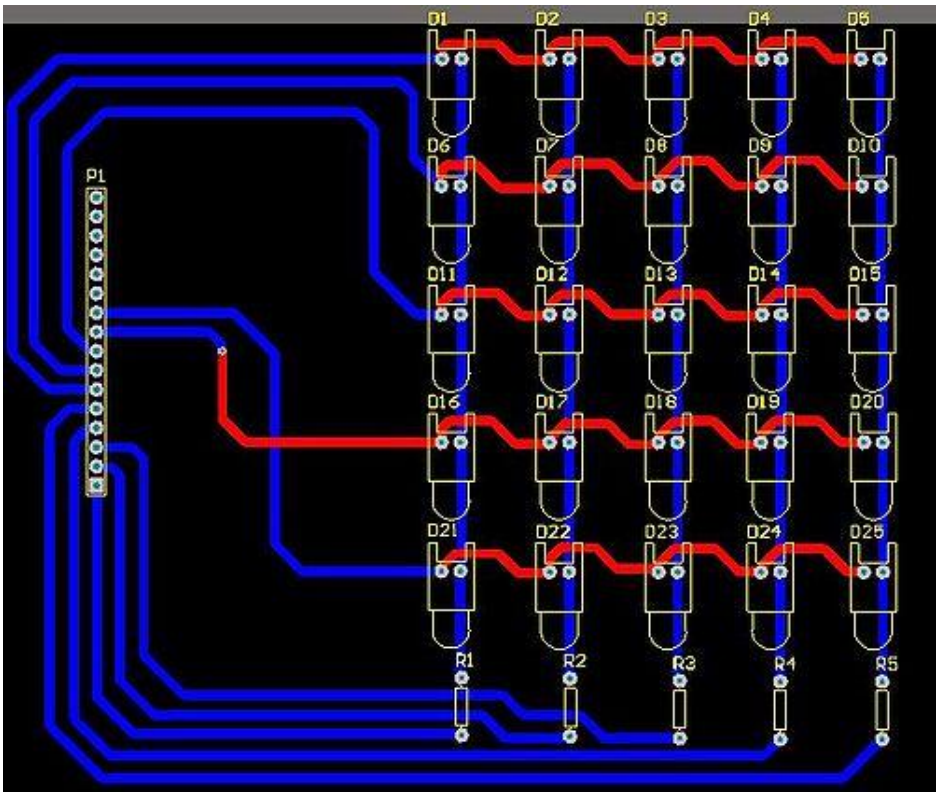


Figure 8: Routage de la Matrice de leds

II.2/ Modifications physique

Dans un second temps, nous sommes allés opérer différents réglages sur notre MPS c'est-à-dire que nous nous sommes plus intéressés aux placements du feu, du convoyeur, des ajouts etc… C'est pourquoi nous avons percé à la base du feu tricolore pour faire passer un câble par exemple.

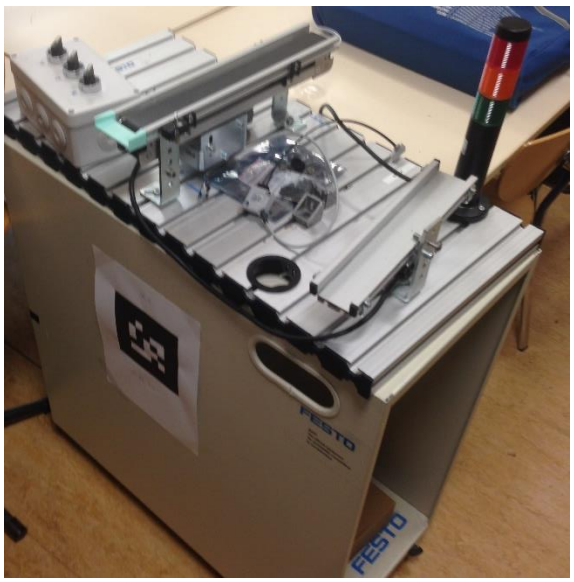


Figure 9: MPS après modifications

Puis nous avons conçu une pièce en 3D sur Onshape, cette pièce était nécessaire pour le maintien des barrières du convoyeur. Nous pouvons voir ci-dessous la conception logicielle et son rendu une fois nos pièces sorties de l'imprimante 3D.

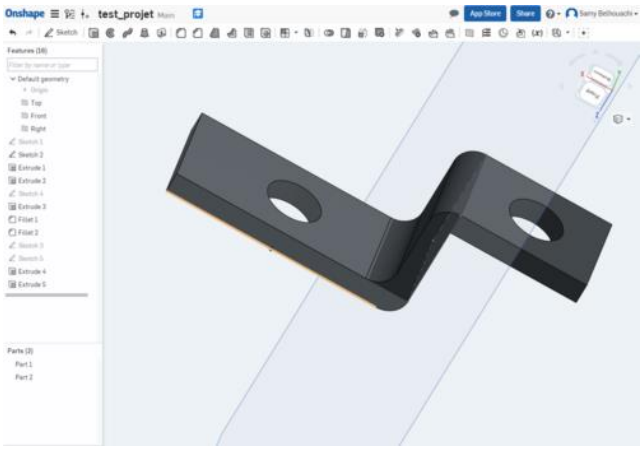


Figure 10a: Conception logicielle

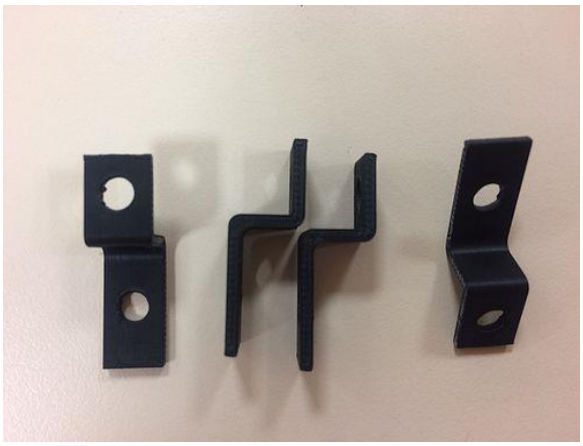


Figure 10b: Pièces 3D imprimées

III/ Partie Informatique

Dans cette partie, nous allons nous attarder sur tous les élements qui concernent la programmation mais aussi les moyens mis en œuvre avec l'arduino. Le programme que nous avions à réaliser a pour but de simuler le comportement des 4 machines MPS selon le choix de l'utilisateur. Le choix sera fait directement sur pc via une liaison série. Un ordinateur portable sera donc utilisé pour alimenter l'arduino ce qui nous amène à ne pas avoir besoin de gérer l'alimentation de l'arduino.

III.1/ La liaison série et la communication avec l'arduino

Nous commencerons par expliquer la liaison série. Elle a pour but de permettre la communication directe avec l'arduino, elle nous permet aussi de décider simplement si l'on souhaite mettre la MPS en mode Base Station, Cap Station, Ring Station ou encore Delivery Station. Pour cela, nous avons décidé de créer une interface graphique. Bien évidemment, nous n'avons pas essayé de faire une interface complexe, simplement quelque chose de fonctionnel.

```
void send_serial( int c , int fd )
    {
        write( fd ,  &c , 1 );
    }
```

Cette simple fonction nous permet d'envoyer un entier lorsque l'on entre un ordre.

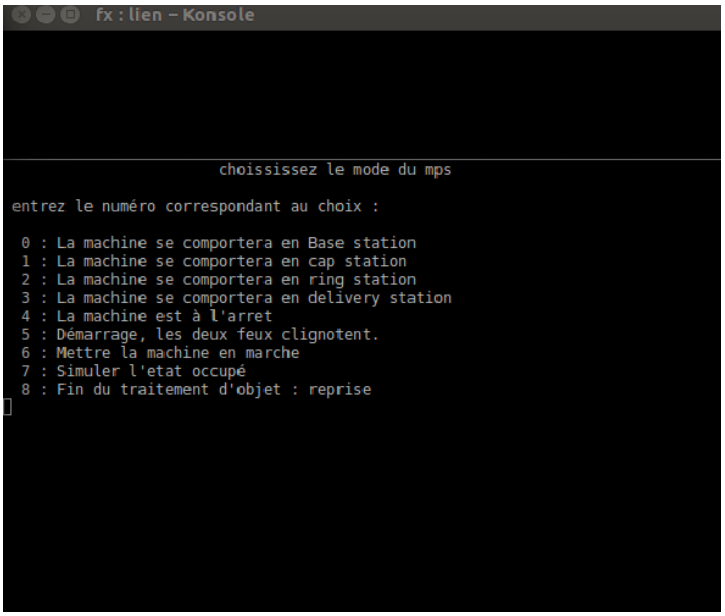


Figure 11: Interface

Nous voyons ci-dessus l'interface de communication avec l'arduino. Nous pouvons donc choisir le type de machine et l'état que celle-ci devra simuler.

Nous pouvons donc maintenant facilement envoyer des données à l'arduino. Cependant, la réception de données du côté de l'arduino ne peut être bloquant puisque nous n'envoyons des données que très rarement. Le fait d'avoir une fonction bloquante rendrait le programme complètement inutilisable.

Pour détecter la réception de données, nous allons utiliser le test suivant:

```
if ( ( UCSR0A & ( 1 << RXC0 ) ) != 0 )
        ordre = get_serial();
```

Nous utilisons donc ici la fonction Get_serial() qui est bloquante de base, cependant lorsque nous utilisons la condition précédente nous pouvons détecter si une donnée est reçue où non ceci grâce aux registres.

La liaison série et l'interface de communication ont donc été réalisé et testé avec des résultats concluants. La plus grande difficulté de cette partie était de faire que la réception de la liaison série ne soit pas bloquante dans l'arduino. Certaine modification pourrait être réalisé dans le but de rendre l'interface utilisateur plus pratique, comme par exemple d'afficher en permanence le dernier choix de machine qui a été demandé ou encore le dernier ordre qui lui a été émis (arrêt , marche , occupé …).

III.2/ Gestion et affichage des matrices de leds

Le travail d'affichage de matrice de LED a été un des points les plus compliqués durant ce projet et nous allons expliquer pourquoi.

Tout d'abord, au tout début du projet, en accord avec nos encadrants, nous avons choisi de travailler avec une Arduino Uno. Sachant que le nombre d'entrées/sorties d'une arduino Uno étant de 23 Pins, il était quand même possible de réaliser notre projet en y mettant un expandeur. Cette solution fut alors retenue et nous avons alors commencé à programmer. Cependant, après plusieurs heures de recherche et de travail, la solution de l'expandeur était un mauvais choix. Nous avons alors trouvé une autre solution qui était d'utiliser un shift register qui est plus simple d'utilisation et de programmation. Après plusieurs heures de programmation, nous avons détecté une petite erreur dans notre programmation qui nous a coûté plusieurs heures de recherche, le shift register fût alors contrôlé.

Un problème nous a alors été émis sur la faisabilité de cette solution. En effet, la matrice de LED ayant pour but de faire un AR-Tag pour permettre au robot de détecter quelle machine est simulée. De plus, la reconnaissance se faisant par traitement d'image, la différence d'intensité entre une colonne où une seule LED est allumée et une colonne où deux LEDs sont allumées pourrait ne pas permettre un traitement d'image fiable. Il fallut donc trouver une

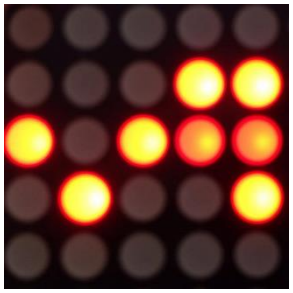
solution où l'on allume une seule LED à la fois et permettant de contrôler les deux matrices de LEDs séparément. La solution était d'utiliser des multiplexeurs pour gérer les colonnes et des shift registers pour gérer les lignes des matrices. Cette solution était toujours applicable à l'aide d'un arduino UNO mais le nombre de sortie étant très restreint, il n'y aurait pas la place pour plus de capteurs si l'on souhaitait développer la MPS dans le futur. Il est toujours possible d'obtenir plus de sortie en mettant des shift registers en cascade mais cela complexifie fortement le programme.

Alors que toutes les parties de code étaient à ce moment-là réalisées, une discussion avec nos encadrants nous a amené à finir le projet sur une arduino MEGA malgré le travail supplémentaire que cela allait engendrer. Cette solution nous a rendu le projet plus viable et plus facilement développable. Ensuite, lors de la modification du makefile et du programme principale nous eurent un problème pour uploader le programme. Il nous fallut plusieurs heures de recherche pour trouver une solution à ce problème. Un simple -D dans le makefile était nécessaire, mais il est plus facile dire ceci après avoir trouvé la solution. Une fois ce problème-là résolu, tout le travail effectué sur le shift register et le multiplexeur n'était plus nécessaire puisque l'arduino MEGA possède un nombre de sorties bien suffisantes pour gérer les deux matrices de LEDs, les capteurs et les relais.

Pour la gestion, nous avons donc choisi de gérer les colonnes et les lignes de chaque matrices par des Pins différentes. Lorsque nous activons une colonne, si l'on met la ligne à l'état haut, la LED reste éteinte, alors que si nous la mettons à l'état bas, la LED correspondant à la ligne s'allume.

```
if ( tabDSA[i][j]==1)
                {
            if ( j == 0)
                {
                 PORTF |= 0x1F;                    // Allumage de LED ligne n°1
                 PORTF &= 0xFE;
                }
            ….
               }
```

La programmation d'affichage de la matrice de LED n'a rien de compliqué, elle prend simplement du temps.



Voici l'exemple d'un AR-Tag obtenu. L'affichage obtenu ici est directement en lien avec l'ordre envoyé depuis la liaison série.

III.3/ Gestion et affichage des différents capteurs et relais

Dans cette partie, nous allons expliquer la manière dont nous contrôlons le feu tricolore, le tapis roulant et les capteurs présents sur le tapis roulant.

Cette partie-là est gérée grâce à une variable " d'état " que l'on choisit directement depuis l'interface graphique. Selon l'état choisi par l'utilisateur, nous pouvons décider de simuler un état " à l'arrêt", " en marche ", " occupé " ou " phase de détection ". Chaque état correspond à un comportement du feu tricolore, du tapis et des capteurs différent.

Les capteurs sont simplement des pins mise en mode " Entrées" avec pour simple test :

```
if ( PORTD && 0x02 == 0x02) //          on arrête le tapis au milieu
        PORTF &= 0xF7;
```

Cet exemple correspond au test en mode " occupé " ce qui équivaut au passage d'un objet devant le capteur présent au milieu du tapis.

Concernant le feu tricolore et le tapis roulant, nous avons vu qu'ils étaient tous deux gérés à l'aide de relais. La gestion d'un relais en sortie d'un arduino est toute simple puisqu'il suffit de faire passer une pin de l'état haut à l'état bas pour changer la position de l'interrupteur.

IV/ Difficultés rencontrées et perspectives

IV.1/ Difficultés rencontrées

Durant ce projet, nous nous sommes confrontés à de nombreux problèmes. Certains dus à un manque de compétence dans un certain domaine, d'autre venant d'une prise de décision trop hâtive. Certains de ces problèmes nous ont fait perdre de longues heures mais nous avons pu apprendre de nos erreurs.

Tout d'abord, lors du tout début de notre projet, nous sommes directement dirigés vers l'utilisation d'une arduino UNO, qui nous permettait de débuter directement notre projet. Cependant, cela nous a rajouté une charge de travail supplémentaire assez conséquente. De plus, nous n'avions quasiment aucune marge sur le nombre de Pin.

Nous avons eu de nombreux changement dans notre cahier des charges. Mais encore un problème rencontré non prévu initialement tel que l'intensité des LEDs qui doivent être uniforme. De ce fait le traitement d'image depuis une caméra, entraîne des modifications ce qui faussait notre idée première. Ajoutons à cela que l'on nous a demandé de prévoir un peu de marge pour pouvoir continuer le projet et ajouter par la suite d'autres capteurs et d'autres sorties, nous n'avons pas eu d'autre choix que de changer d'arduino ce qui nous a fait perdre un certain temps.

Ensuite, nous avons pu rencontrer d'autres problèmes lorsque nous avons eu à faire toute la partie électronique de puissance. Nous avons passé un bon nombre d'heures à essayer de trouver les bonnes valeurs des composants permettant d'avoir les tensions voulus en entrée de l'arduino, cela concernait aussi les éléments extérieurs (capteurs, moteur, feu tricolore). De plus, il fallait faire attention à la commande des relais puisqu'il fallait réussir à les faire commuter. Nous avons reçu de l'aide pour cette partie, quelqu'un travaillant à Polytech est venu voir ce que l'on avait fait et a pu confirmer certaines valeurs mais aussi en changer quelques-unes.

Pour finir, nous avons rencontré d'autres problèmes lorsque nous avons testé les capteurs. En effet, lorsque nous alimentons le capteur à 24V et que nous approchons un objet du capteur, alors les leds présentes sur le capteur nous indiquaient bien la présence d'un objet, mais sur la broche de réception nous ne recevions rien. Le test avait d'abord été fait directement relié à notre arduino avec un transistor au milieu pour abaisser la tension mais lors des vérifications à l'aide d'un voltmètre, rien n'était observé en sortie. Nous avons testé plusieurs capteurs mais aucuns résultats concluant n'ont été observé.

De plus, nous avons rencontré un dernier problème lorsque l'on a essayé d'assembler toutes les pièces de la machine. En effet, il manquait beaucoup de pièces (vis, écrous, etc…) ce qui rendait impossible l'assemblage de la machine. Néanmoins, ce souci nous a permis de travailler un peu le logiciel de CAO vu en cours.

IV.1/ Perspectives

Nous avons pu avancer dans un bon nombre de domaine, mais il reste encore du travail à effectuer. Pour la partie informatique, il reste à simuler le fonctionnement des différents états de la MPS. Cette tâche devait être réalisée s'il nous restait du temps en fin de projet, mais nous n'avons pas pu faire cette partie-là. Il nous reste à faire un compteur non bloquant pour faire clignoter le feu tricolore pendant la phase de recherche.

Pour la construction de la MPS et de tous les composants qui y sont associés, la matrice de LED a été routé mais n'a pas pu être faite car nous n'avions pas les leds nécessaires. Toute la partie électronique de puissance est normalement faite, il reste à trouver les raisons du problème de capteur, certains composants sont encore manquants pour finir la machine.

Conclusion

Pour conclure, lors de ce projet nous avons pu allier la théorie à la pratique et plus particulièrement dans le domaine de l'informatique, de la CAO, de l'électronique et de l'électronique de puissance. Nous avons donc pu acquérir de nouvelles connaissances dans tous ces domaines. Mais nous avons aussi acquis de nombreuses compétences humaines grâce au travail en équipe.

Grâce aux différentes erreurs et aux mauvais choix durant notre projet, nous savons désormais ce qu'il faut faire et ne pas faire. De plus, le travail que nous avons réalisé est utile pour l'ARPL qui pourra finaliser le projet très prochainement. La recherche de solutions fut l'une des phases les plus importantes de notre projet car il n'y avait pas qu'une solution possible pour notre problème, ce qui nous a donc amené à prendre beaucoup de recul.

Ce fut très intéressant de travailler avec l'ARPL, nous avons pu découvrir de nouvelles façons de travailler et de nouvelles choses que nous ne connaissions pas avant ce projet.

Bibliographie / Sitographie

Ci-dessous vous pouvez lire le livre consulté durant la phase de recherche :

- BARRY, Jean. *Schémas d'électricité*. Eyrolles, 1980.

Voici les sites visités durant la phase de recherche :

- Astuces pratiques. [En ligne]. Astuces pratiques, 2017. Disponible à l'adresse : https://www.astuces-pratiques.fr/electronique/le-relais-principe-de-fonctionnement
- Mon club élec. [En ligne]. David Gilbert, 2013. Disponible à l'adresse :http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielMega2560
- ZEM. [En ligne]. ZEM, 2017. Disponible à l'adresse : http://www.zem.fr/decouverte-du-composant-74hc595-8-bit-shift-register/