ESME sudria Paris

M1B

Rapport de stage

Tuteur de stage : Eric Regnard

Quentin Mercier

A2V Mécatronique

2014-2015

Sommaire :

Remerciements :

Mes remerciements s'adressent en premier lieu à mon maître de stage, Monsieur Eric REIGNARD, président d'A2v pour m'avoir permis d'intégrer la société A2V et de participer activement à ses projets, dans un second temps à Monsieur Albin MARCHEWKA, Ingénieur Bureau d'Etude, pour m'avoir fait partager toute son expérience et ses compétences ; pour le temps qu'il m'a consacré tout au long de cette période de stage, sachant répondre à toutes mes interrogations ; sans oublier sa participation à la réalisation de ce mémoire.

C'est grâce aux missions que l'on m'a confié que j'ai choisi d'aborder ce sujet dans mon rapport de stage. Ce stage a nécessité tout au long de sa durée l'aide et le soutien de plusieurs personnes.

J'exprime également ma gratitude à l'ensemble du personnel d'A2V Mécatronique pour avoir concouru à rendre ce passage en entreprise agréable et avoir facilité mon intégration dans cette équipe jeune, dynamique et sympathique.

Je souhaiterais également remercier Monsieur AIT ABDERRAHIM Karim pour toute l'aide qu'il m'a apportée lors de ma formation et de l'attention qu'il a eue à mon égard.

Introduction :

1. Présentation de l'entreprise A2V Mécatronique

Dans le cadre de ma dernière année d'études, j'ai effectué mon stage au sein de la société A2V (Applications à Vitesse Variable) Mécatronique située à Gazeran. Créée en 1991, A2V est une société par actions simplifiée dirigée par Eric Regnard. Elle est spécialisée dans l'asservissement des servomoteurs brushless, des moteurs à courant continu et pas à pas. Sur l'année 2014, elle réalise un chiffre d'affaires de 3,5 Millions d'euros et emploie actuellement 13 personnes.

A ce jour, la société est désormais divisée en trois grands pôles qui sont : la mécatronique, la nanotronique et la flowtronique. Elle offre une large gamme de produits dans ces différents domaines et propose à ses clients de développer des projets sur mesure grâce à son bureau d'études. La formation des techniciens, le support technique et le service après-vente font aussi partie du travail de l'entreprise.

La motorisation est le cœur de métier d'A2V. Les moteurs commandés sont de type brushless, pas à pas, piezo, linéaires ou couples. Pour les piloter de nombreuses solutions sont proposées au travers de cartes de commande. Celles-ci sont capables de contrôler plusieurs axes et de communiquer entre elles ou bien avec un ordinateur par exemple par le biais de différents protocoles de communications. L'ensemble des éléments indispensables au fonctionnement de ces systèmes est également proposé comme les codeurs, les collecteurs, les encodeurs et les interfaces IHM.

Les projets développés sont extrêmement variés pour les secteurs de la recherche, de l'industrie, de l'évènementiel…

1) Le contrôle des moteurs

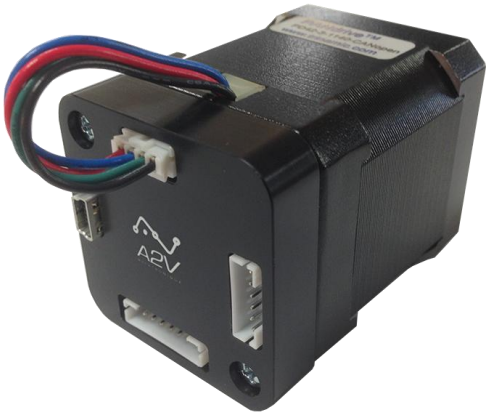
Les moteurs brushless sont les plus couramment utilisés et conviennent pour de nombreuses applications. Un contrôleur d'axes et un variateur sont nécessaires à son fonctionnement. Ceux-ci peuvent être intégrés directement au moteur avec un codeur par exemple. L'ensemble des moteurs brushless balaient une large gamme de puissances avec différents encombrements. Les communications sont de type CANopen, ProfiBUS, DeviceNEt, RS232/RS485…

Les moteurs pas à pas sont eux aussi largement proposés.

Les moteurs piezo permettent un positionnement très précis, de l'ordre du nanomètre. En céramique, ils ne génèrent pas de champ magnétique et peuvent être utilisés dans des applications ou le vide est nécessaire. A2V conçoit des tables très précises de positionnement.

Les contrôleurs d'axes permettent, pour certains, le contrôle de différents types de moteurs. Il est possible de les programmer et de communiquer avec, à l'aide de commandes. L'asservissement peut être réglé ainsi que les accélérations, les décélérations et les vitesses maximales. Ils proposent également de nombreuses entrées/ sorties pour gérer le Homing et les Switch par exemple.

Moteur Brushless avec variateur intégré



Moteur pas à pas avec variateur intégré

2) Les développements du bureau d'études

Le bureau d'études d'A2V Mécatronique développe de nombreux projets pour des applications variées comme des roues à filtres, des systèmes pour des positionnement nanométriques, des chariots mobiles autonomes, ou bien des caméras suspendues par câble.

Pour réaliser ces projets, les ingénieurs du bureau d'études, conçoivent la mécanique, ainsi que l'électronique et le logiciel embarqué. De nombreux projets sont pilotés grâce à une interface Codesys qui permet de développer l'architecture automatique et le contrôle commande des actionneurs présents sur le projet. Codesys est un standard de programmation offrant une grande souplesse et reconnu dans l'industrie.

Les personnes travaillant au bureau d'études assurent donc la qualité des produits livrés, l'accompagnement technique, les essais de mise en œuvre et le service après-vente. Un support est également disponible en ligne pour les clients qui souhaitent disposer de manuels, logiciels, plans de câblage…

2. Sujet de stage

Au cours de ces 6 mois au sein du Bureau d'Etude de l'entreprise, j'ai pu m'intéresser au développement de différents produits et également au lancement d'une nouvelle gamme.

Plus largement, ce stage à été l'opportunité pour moi d'appréhender une partie du secteur de la Mécatronique, et de découvrir plus particulièrement le rôle et les missions d'un ingénieur Bureau d'Etude.

Au delà d'enrichir mes connaissances et de compléter ma formation technique, ce stage m'a permis de comprendre le fonctionnement d'une petite entreprise, mais également de voir l'ensemble de la chaine de conception d'un produit ; de l'idée du client au produit final.

Mon stage au sein du bureau d'étude a consisté en différentes missions, telles que la conception d'une carte électronique permettant le pilotage de roues à filtres, la conception mécanique de capots moteurs, ou encore la création et le lancement d'une nouvelle gamme de produit.

Mon maître de stage étant Ingénieur Bureau d'Etude, j'ai pu aborder les différentes missions que l'on m'a confiées avec une grande sérénité, sachant qu'à tout moment il été présent pour répondre à mes interrogations et me former sur les logiciels techniques que l'on utilise au sein d'A2V.

Ce stage a été une opportunité pour moi de percevoir comment cette entreprise évolue dans un secteur complexe. L'élaboration de ce rapport a pour principale source les différents enseignements tirés de la pratique journalière des tâches auxquelles j'étais affecté. Enfin, les nombreux entretiens que j'ai pu avoir avec les employés des différents services de la société m'ont permis de donner une cohérence à ce rapport.

I. Le projet Tour EIFFEL

1. Présentation du projet

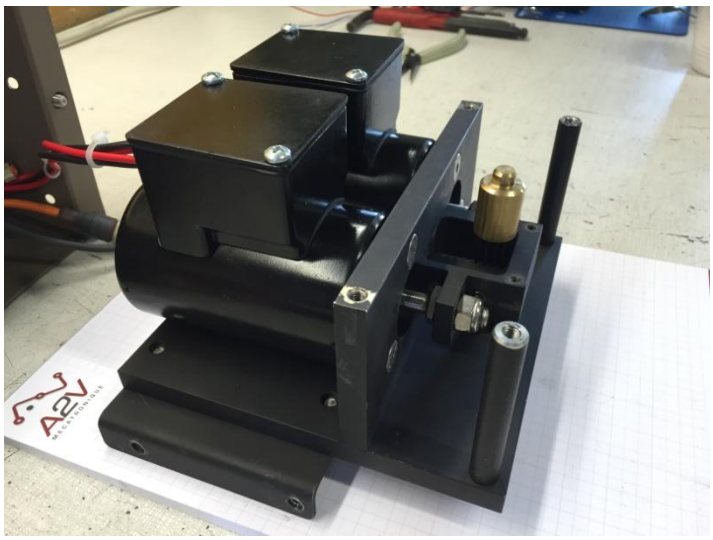
Tous les soirs, la tour Eiffel se pare de son habillage doré et scintille 5 minutes au début de chaque heure, tandis que son phare rayonne sur Paris.

Le premier phare de la tour Eiffel a été mis en place en 1952. Il est constitué de deux phares tournant sur des rails, un regardant vers le haut, l'autre vers le bas pour une portée de 300 km.

Pour le passage à l'an 2000, la tour a été équipée d'un faisceau lumineux tournant à la manière d'un phare rappelant ainsi le projet initial de Gustave Eiffel. Les deux faisceaux lumineux balayant le ciel parisien jusqu'à une distance de 80 kilomètres sont produits par quatre projecteurs motorisés de type « marine » munis de lampes au xénon de 6 000 W. Les projecteurs effectuent chacun un quart de tour synchronisés de façon que l'ensemble forme un double faisceau en croix pivotant à 360°. Lors du fonctionnement du phare, les 4 projecteurs sont allumés en permanence, et un système de shutter permet de laisser passer ou non la lumière afin de donner l'impression d'une rotation discontinue et de pouvoir remettre les phares en position sans éclairer l'intérieur de la tour.

Ce système ayant donc été mis en place en l'an 2000 souffrait de pannes régulières et demandait une maintenance quasi constante. Pour palier à ce problème, la société A2V a été mandatée afin de remplacer le système de shutter.

Les shutters sont composés de 2 solénoïdes permettant d'ouvrir et de fermer les battants. J'ai été chargé de réaliser la partie électronique permettant de contrôler l'ouverture et la fermeture des volets.



Bloc solénoïdes

Le cahier des charges imposé par le client nous donnait les contraintes suivantes :

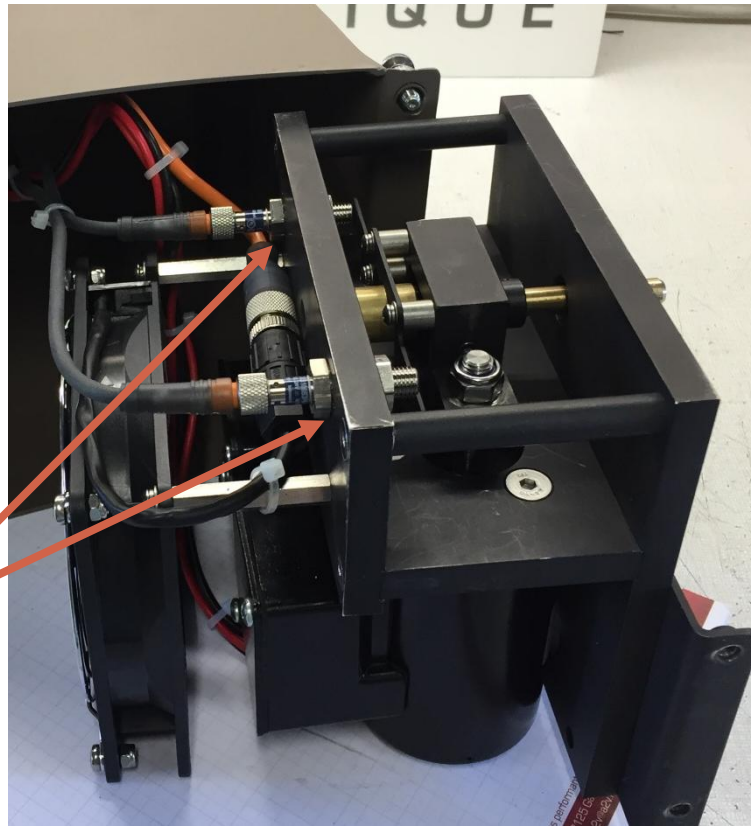
- Utilisation d'un automate Wago CANopen pour le pilotage des solénoïdes
- La commande d'ouverture ou de fermeture est un signal 230V AC
- Le changement d'état doit être instantané lors du changement d'état de la consigne
- Il faut pouvoir repérer un dysfonctionnent du système

Un cycle complet du phare est le suivant :

- Le phare se place en position prêt à tourner
- Le phare commence à tourner et on ouvre le shutter
- Fermeture du shutter au bout de 45°
- Accélération du phare pour finir un tour complet et se remettre en place pour un nouveau cycle

Le cycle de fonctionnement du shutter impose donc d'autres contraintes comme le temps minimal entre une ouverture et une fermeture de shutter (Environ 15 secondes).

Pour le bon fonctionnement du système, le bloc solénoïde permettant l'ouverture et la fermeture des volets dispose de capteurs de proximités pour détecter les fins de courses.



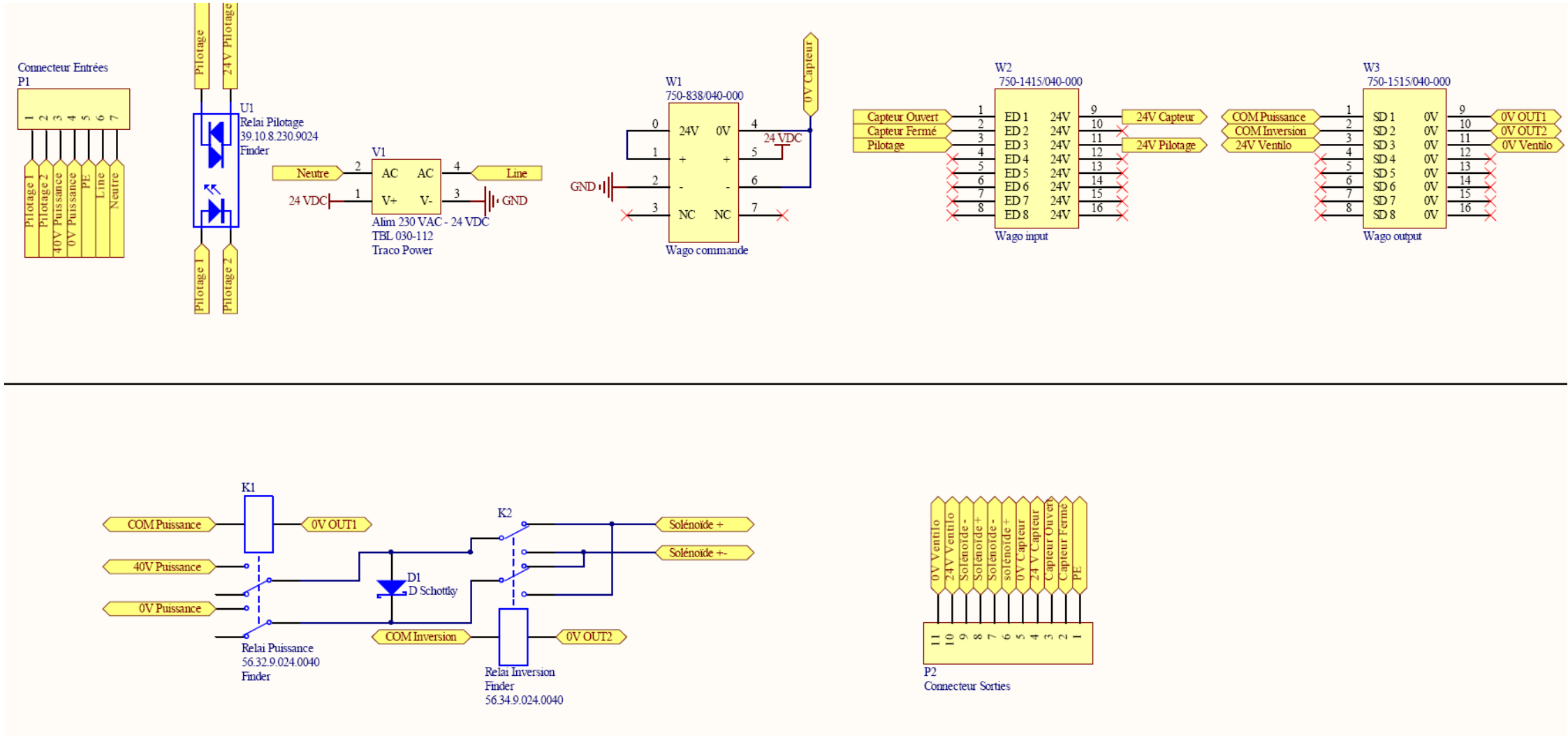
Capteurs fins de courses

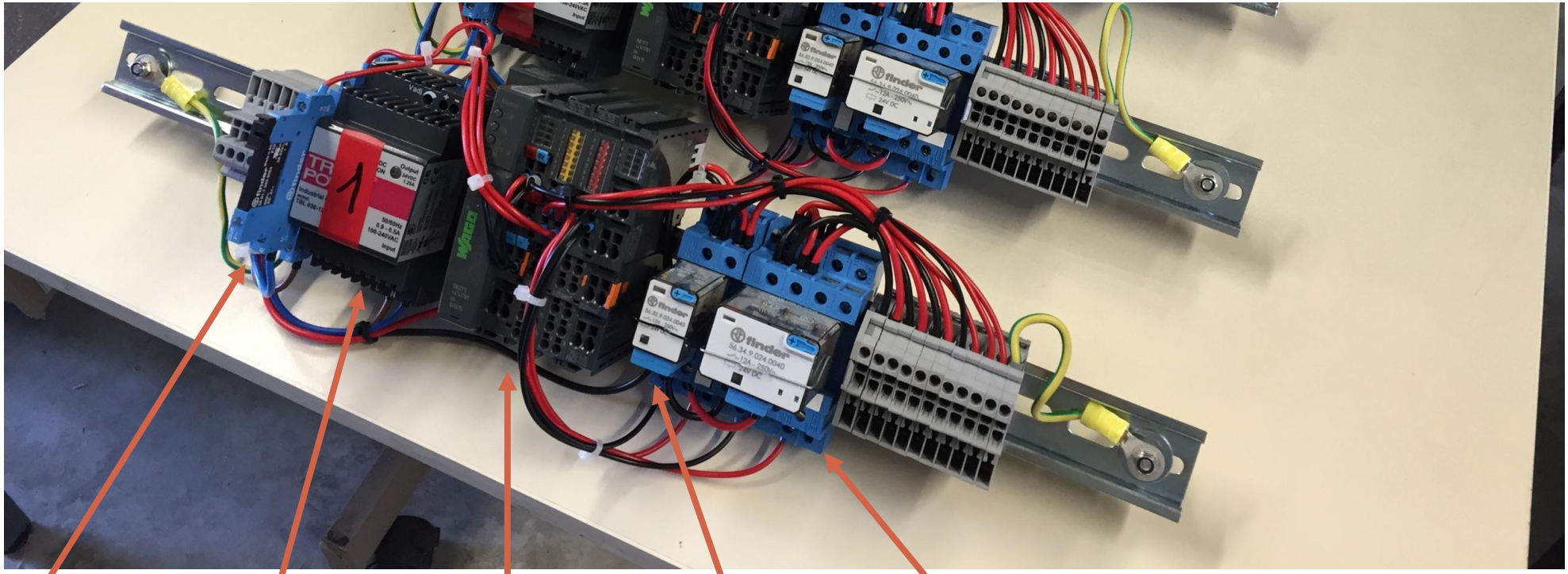
C’est ainsi que j’ai réalisé un rail DIN permettant le pilotage des solénoïdes avec l’automateWago.

Il se compose des éléments suivants :

- Une alimentation 24V pour la logique
- Un optocoupleur piloté en 220V pour la commande
- L’automate Wago
- Un relai de puissance pour les solénoïdes
- Un deuxième relai de puissance afin d’inverser la tension d’alimentation des solénoïdes

Voici les schémas de câblage du RAIL-DIN


Connecteur Entrées
P1
1
2
3
4
5
6
7
Pilotage 1
Pilotage 2
40V Puissance
0V Puissance
PE
Line
Neutre
Pilotage
24V Pilotage
U1
Relai Pilotage
39.10.8.230.9024
Finder
Pilotage 1
Pilotage 2
V1
Neutre
24 VDC
2
1
AC
AC
V+
V-
4
3
Line
GND
Alim 230 VAC - 24 VDC
TBL 030-112
Traco Power
W1
750-838/040-000
0
1
2
3
24V
0V
+
+
-
-
NC
NC
4
5
6
7
24 VDC
GND
0V Capteur
Wago commande
W2
750-1415/040-000
Capteur Ouvert
Capteur Fermé
Pilotage
1
2
3
4
5
6
7
8
ED 1
ED 2
ED 3
ED 4
ED 5
ED 6
ED 7
ED 8
24V
9
10
11
12
13
14
15
16
24V Capteur
24V Pilotage
Wago input
W3
750-1515/040-000
COM Puissance
COM Inversion
24V Ventilo
SD 1
SD 2
SD 3
SD 4
SD 5
SD 6
SD 7
SD 8
0V
0V OUT1
0V OUT2
0V Ventilo
Wago output
K1
COM Puissance
40V Puissance
0V Puissance
0V OUT1
Relai Puissance
56.32.9.024.0040
Finder
K2
Solénoïde +
Solénoïde +-
D1
D Schottky
COM Inversion
0V OUT2
Relai Inversion
Finder
56.34.9.024.0040
0V Ventilo
24V Ventilo
Solénoïde -
Solénoïde +
Solénoïde -
solénoïde +
0V Capteur
24 V Capteur
Capteur Ouvert
Capteur Fermé
PE
11
10
9
8
7
6
5
4
3
2
1
P2
Connecteur Sorties


Optocoupleur
Alimentation 24V
Automate Wago
Relai commande
Relai inversion

L'automate Wago étant un automate compatible Codesys, cela m'a permis de prendre en main le logiciel et les différents types de programmation possible. Mon choix s'est arrêté sur le langage ST qui est composé d'instructions simples tel que « IF », « ELSE », …

Le cycle de l'automate fonctionne de la façon suivante :

Afin que la commande puisse être changée instantanément en fonction de la consigne, il faut anticiper en inversant le relai d'inversion à la fin d'un cycle. En effet, il faut prévoir un délai entre l'envoi de la commande sur le relai d'inversion et sur le relai de puissance pour ne pas risquer d'alimenter en -40V pendant 0.1 secondes puis 40V par exemple.

De ce fait le cycle démarre lors du changement de consigne par l'alimentation de la bobine du relai envoyant la puissance sur les solénoïdes. On attend ensuite le déclenchement du capteur de fin de course. Si celui-ci n'est pas détecter au bout de 5 secondes alors c'est qu'il y a un problème d'ouverture ou de fermeture des volets et on enregistre alors un défaut. Si tout se passe normalement, alors on continu d'alimenter encore 1 seconde afin que le solénoïde « colle » correctement et verrouille la position des volets, en effet la position des capteurs permet de s'assurer que le volet a bien changé d'état mais pas que celui est verrouillé.

Afin de ne pas trop faire chauffer le système et consommer inutilement, les solénoïdes ne sont alimentés que lors des changements d'états donc une fois celui-ci considéré comme verrouillé, on coupe l'alimentation, on attend une seconde de roue libre puis on inverse l'état du relai d'inversion, enfin on attend encore une seconde pour s'assurer du changement d'état du relai puis un nouveau cycle peut recommencer.

Lors de nos premiers tests sur le terrain, nous nous sommes rendu compte que l'arrivée de la puissance, utilisée pour alimenter le ventilateur permettant de refroidir les solénoïdes était allumée en permanence, et qu'en conséquence le ventilateur fonctionnait inutilement pendant la journée. Nous avons donc rajouté dans l'automate une sortie pour alimenter celui-ci, également à chaque fois que la consigne change nous réinitialisons un compteur de 5 minutes. Par conséquent quand le phare est arrêté, le système ne reçoit plus de nouvelles consignes et le ventilateur s'arrête de lui-même au bout de 5 minutes.

Voici le programme codesys :

```
0015
0016 (* PHASE 5 :  On inverse le relai puis on attend une seconde avant de pouvoir recommencer un cycle*)
0017 IF WAITING_READY THEN
0018     (* Si la tempo est finie *)
0019     IF TEMPO.Q = FALSE THEN
0020         TEMPO(IN:=FALSE, PT :=TEMPO_END_CYCLE);
0021         WAITING_READY := FALSE;
0022     ELSE
0023     (* sinon on continu d'attendre *)
0024         TEMPO(IN:= TRUE, PT:=TEMPO_END_CYCLE);
0025     END_IF
0026
0027 (*PHASE 4 : on attends 1 seconde de roue libre avant de pouvoir inverser*)
0028 ELSIF WAITING_INVERSION THEN
0029     (* si la seconde de roue libre est terminée : *)
0030     IF TEMPO.Q = FALSE THEN
0031
0032         (* alors on Inverse le relai en prévision du prochain changement d'état *)
0033         IF POLARITY = TRUE THEN
0034             POLARITY := FALSE;
0035         ELSE
0036             POLARITY := TRUE;
0037         END_IF;
0038         WAITING_INVERSION := FALSE;
0039
0040         (*On lance une tempo de 1 seconde  pour s'assurer du changement d'état du relai d'inversion*)
0041         TEMPO(IN:=FALSE, PT:=DELAY_INVERSION);
0042         TEMPO(IN:=TRUE, PT:=TEMPO_END_CYCLE);
0043         WAITING_READY := TRUE;
0044     ELSE
0045         (* sinon on continue d'attendre *)
0046         TEMPO(IN:=TRUE, PT:=DELAY_INVERSION);
0047     END_IF;
```

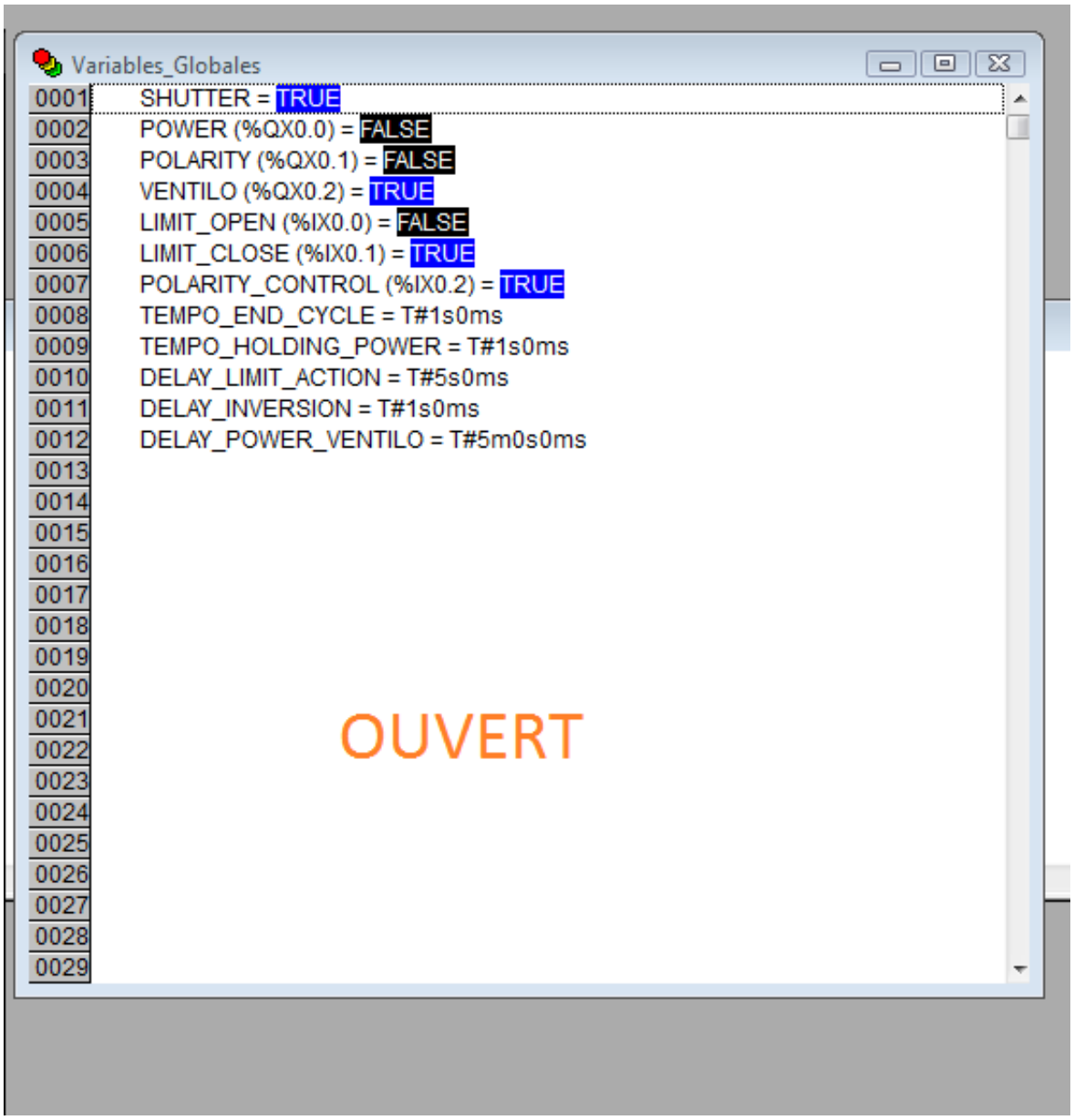
```
0049 (*PHASE 3 : on attends 1 secondes après détections pour arrêter l'alim*)
0050 ELSIF DELAY_POWER = TRUE THEN
0051     (* si la tempo de maintient de puissance après détection est finie *)
0052     IF TEMPO.Q = FALSE THEN
0053         (* On coupe l'alim, on associe la variable SHUTTER avec l'état du shutter*)
0054         SHUTTER := POLARITY;
0055         POWER := FALSE;
0056
0057         (* On lance la tempo de roue libre *)
0058         TEMPO(IN:=FALSE, PT:=TEMPO_HOLDING_POWER);
0059         TEMPO(IN:=TRUE, PT:=DELAY_INVERSION);
0060         WAITING_INVERSION := TRUE;
0061         DELAY_POWER := FALSE;
0062     ELSE
0063         (* sinon on continu d'attendre *)
0064         TEMPO(IN:=TRUE, PT:=TEMPO_HOLDING_POWER);
0065     END_IF;
0066
0067 (*PHASE 2 : on attends que le module se ferme ou fin de tempo *)
0068 ELSIF WAITING_CLOSE = TRUE OR WAITING_OPEN = TRUE THEN
0069
0070     (*Shutter en cours de fermeture, on attend détection du capteur ou que la tempo de sécurité de 5 secondes soit écoulée *)
0071     IF WAITING_CLOSE = TRUE AND (LIMIT_CLOSE = TRUE OR TEMPO.Q = FALSE) THEN
0072
0073         (* Si on a dépassé le délai de 5 secondes, on enregistre un défaut *)
0074         IF TEMPO.Q = FALSE THEN
0075             DEFAULT := DEFAULT +1;
0076         END_IF;
0077
0078         (* On lance la tempo de maintient de la puissance après détection *)
0079         TEMPO(IN:=FALSE, PT:=DELAY_LIMIT_ACTION);
0080         TEMPO(IN:=TRUE, PT:=TEMPO_HOLDING_POWER);
0081         DELAY_POWER := TRUE;
0082          WAITING_CLOSE := FALSE;
0083
0084     (* Shutter en cours d'ouverture, on attend détection du capteur ou que la tempo d" sécurité de 5 secondes soit écoulée *)
0085     ELSIF WAITING_OPEN = TRUE AND (LIMIT_OPEN = TRUE OR TEMPO.Q = FALSE )THEN
0086
0087         (* Si on a dépassé le délai, on enregistre un défaut *)
0088         IF TEMPO.Q = FALSE THEN
0089             DEFAULT := DEFAULT +1;
0090         END_IF;
0091
0092         (* On lance la tempo de maintient de la puissance après détection *)
0093         TEMPO(IN:=FALSE, PT:=DELAY_LIMIT_ACTION);
0094         TEMPO(IN:=TRUE, PT:=TEMPO_HOLDING_POWER);
0095         DELAY_POWER := TRUE;
0096          WAITING_OPEN := FALSE;
0097     ELSE
0098         (* sinon on continue de compter les 5 secondes de temps limite pour éxécuter l'action *)
0099         TEMPO(IN:=TRUE, PT:=DELAY_LIMIT_ACTION);
0100     END_IF;
0101
```

```
0102 (*PHASE 1 : Quand la consigne change on amorce le processus de changement d"état du shutter*)
0103 (* Et lors de l'init du système *)
0104 ELSIF POLARITY_CONTROL <> SHUTTER  OR INIT = FALSE THEN
0105     INIT := TRUE;
0106
0107     (* Relai d'inversion positioné en fonction de la consigne, utile uniquement lors de l'init *)
0108     POLARITY := POLARITY_CONTROL;
0109     (* On envoit la puissance *)
0110     POWER:=TRUE;
0111
0112     (* On attends l'ouverture ou la fermeture du shutter en focntion de la consigne et on lance la tempo de sécurité de 5 secondes pour détecter les problèmes *)
0113     IF POLARITY = FALSE THEN
0114         WAITING_CLOSE := TRUE;
0115     ELSE
0116         WAITING_OPEN := TRUE;
0117     END_IF;
0118     TEMPO(IN:=TRUE, PT:=DELAY_LIMIT_ACTION);
0119
0120     (* Réinit du timer pour ventilo*)
0121     TEMPO_VENTILO (IN:= FALSE, PT:=DELAY_POWER_VENTILO);
0122     TEMPO_VENTILO (IN:= TRUE, PT:=DELAY_POWER_VENTILO);
0123     VENTILO := TRUE;
0124 END_IF;
0125
0126 (* Gestion de l'alim du ventilo*)
0127 (* Si il n'y a pas eu de changement d'état du shutter depuis 5 minutes *)
0128 IF TEMPO_VENTILO.Q = TRUE THEN
0129     (* On coupe le ventilo *)
0130     VENTILO := FALSE;
0131     TEMPO_VENTILO (IN:= FALSE, PT:=DELAY_POWER_VENTILO);
0132 ELSE
0133     (* on incrémente la tempo de 5 minutes *)
0134     TEMPO_VENTILO (IN:= TRUE, PT:=DELAY_POWER_VENTILO);
0135 END_IF;
0136
0137
```

```
PROGRAM PLC_PRG
VAR
    INIT:BOOL:=FALSE;
    WAITING_CLOSE: BOOL:=FALSE;
    WAITING_OPEN: BOOL:=FALSE;
    WAITING_READY:BOOL:=FALSE;
    WAITING_INVERSION:BOOL:=FALSE;
    WAITING_STOP:BOOL:=FALSE;
    DELAY_POWER:BOOL:=FALSE;
    TEMPO: TP;
    TEMPO_VENTILO: TON;
END_VAR
VAR RETAIN
    DEFAULT: INT ;
END_VAR
```

```
0001 VAR_GLOBAL
0002     SHUTTER: BOOL;
0003     POWER AT %QX0.0: BOOL;
0004     POLARITY AT %QX0.1: BOOL;
0005     VENTILO AT %QX0.2: BOOL;
0006     LIMIT_OPEN AT %IX0.0:BOOL;
0007     LIMIT_CLOSE AT %IX0.1:BOOL;
0008     POLARITY_CONTROL AT %IX0.2 :BOOL;
0009     TEMPO_END_CYCLE: TIME := t#1s;
0010     TEMPO_HOLDING_POWER: TIME := t#1s;
0011     DELAY_LIMIT_ACTION: TIME := t#5s;
0012     DELAY_INVERSION: TIME := t#1s;
0013     DELAY_POWER_VENTILO: TIME :=t#5m;
0014 END_VAR
0015
0016
0017
```

Le logiciel codesys permet également de visualiser en temps réel tous les changements de variables ainsi que de simuler des entrées ce qui peut s'avérer très pratique pour tester le système lors de la conception.



Le projet incluait un test de chaque shutter en simulant les conditions réelles dans les locaux d'A2V, il a donc également fallu réaliser un banc de test permettant d'envoyer les consignes aux shutters comme le système de la tour Eiffel le ferait.

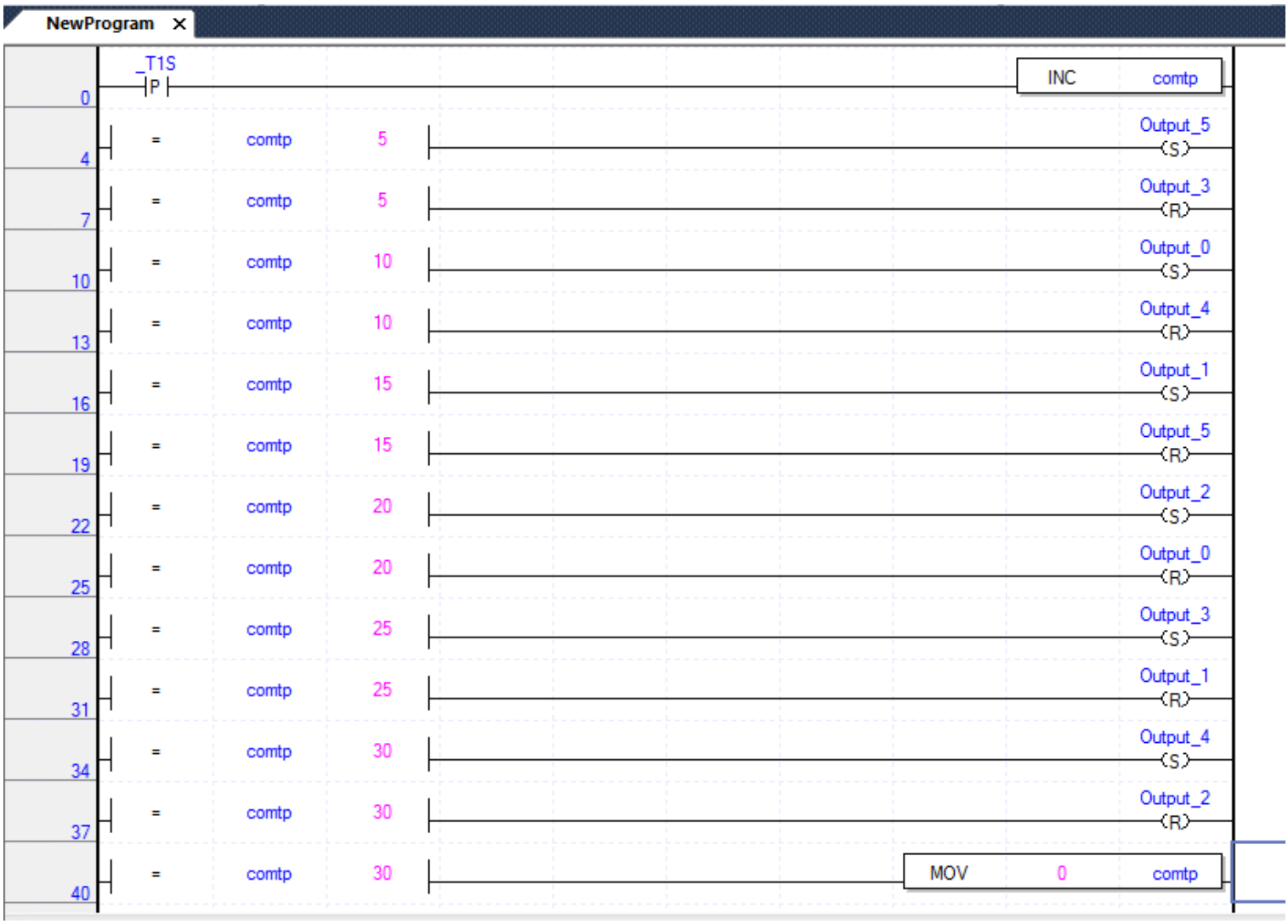
Pour se faire j'ai utilisé un automate LS-IS, et ceux pour différentes raisons :

- Disponible directement chez A2V
- M'a permis de découvrir un autre logiciel automate et un autre type de codage

- Toutes les sorties de l'automates sont des sorties relais, il m'a donc permis d'alimenter directement les relais en 230V AC.

Le fonctionnement de ce banc de test est quand à lui très simple, l'automate envoie les consignes pour ouvrir et fermer les shutters les un après les autres à raison de deux ouvertures/fermetures par minute.

Le choix a été fait d'utiliser du graphset pour le programme de l'automate afin de me faire découvrir quelque chose de nouveau et car il se prêtait plutôt bien à l'application.



J’ai donc pu découvrir que parfois il fallait développer des outils dans l’unique but de tester un autre outil que l’on est également en train de développer.

Station de test des shutters de la tour Eiffel

II. Switch Ethernet pour moteur modbus

1. Présentation du projet

A2V est comme expliqué plus haut spécialisé dans les moteurs avec électroniques intégrées. En effet nous développons des cartes de commandes que nous fixons directement à l'arrière du moteur.

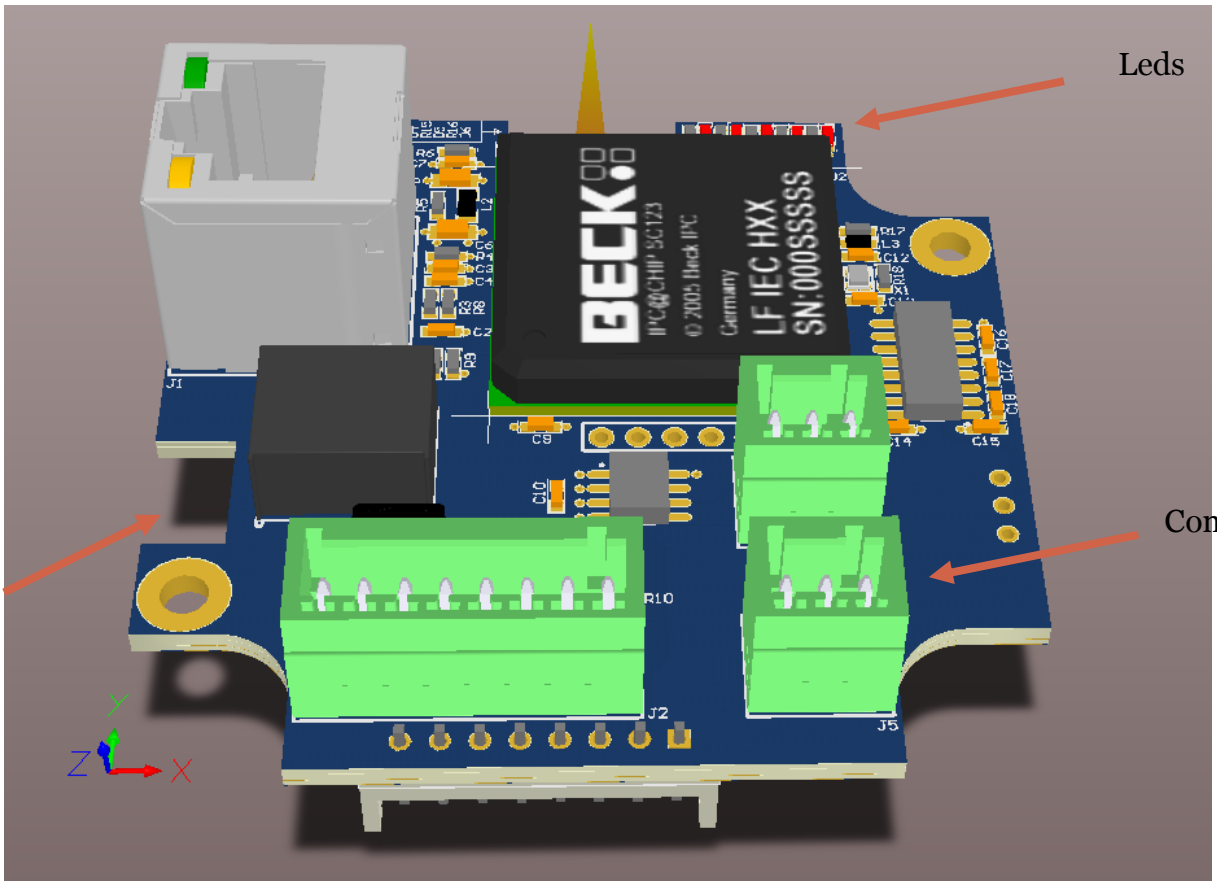
Récemment un membre du BE a développé une carte pour un client, pour un moteur pas à pas, incluant un automate codesys, bus CANopen, modbus et un internet.

Dans la suite de ce projet, il a été décidé de développer une carte mezzanine qui se fixerait par-dessus l'ancienne afin de rajouter une option switch ethernet qui permettrait de chainer les moteurs.

Mon travail a donc consisté à :

- Choisir un composant switch ethernet
- Réaliser les schémas électriques de la carte
- Router la carte de telle façon à ce qu'elle puisse se fixer directement sur l'ancienne carte

Il faut également prendre en compte que dans un souci d'encombrement pour le client la carte ne peut se permettre de dépasser en largeur du moteur.



Carte existante pour le moteur

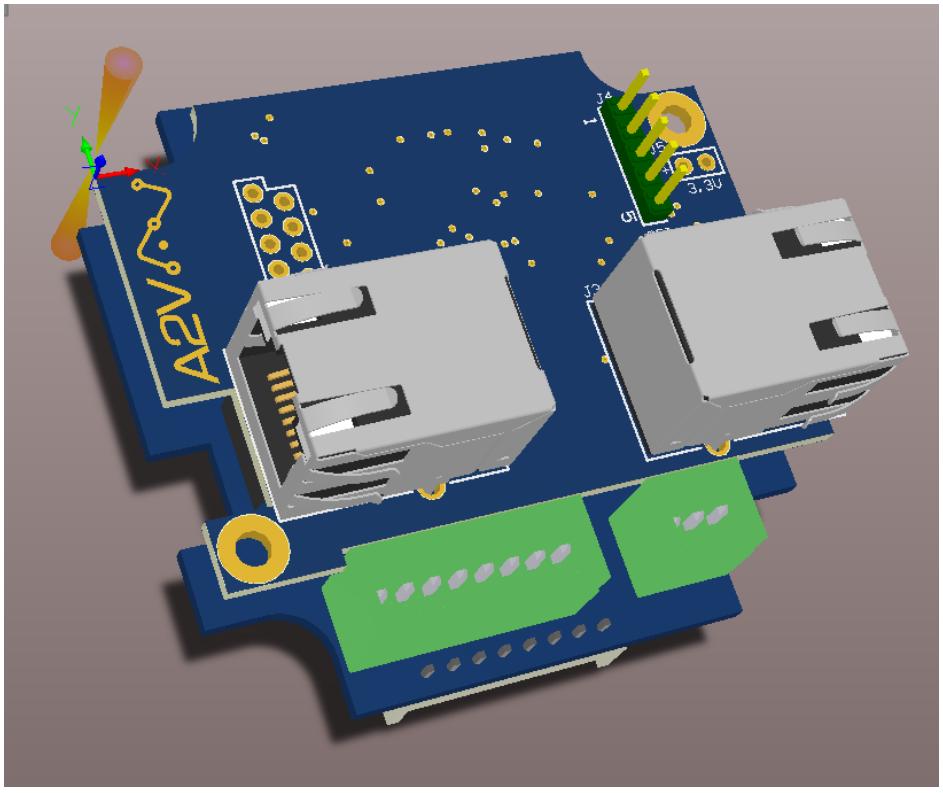
Comme on peut le voir sur la photo précédente, la carte sur laquelle sera fixé le Switch Ethernet impose quelques contraintes quand à la forme de cette dernière. En effet les connecteurs verts devront rester accessibles, la fente sur la gauche de la carte doit rester accessible, et il ne faut pas recouvrir les leds du fond de la carte.

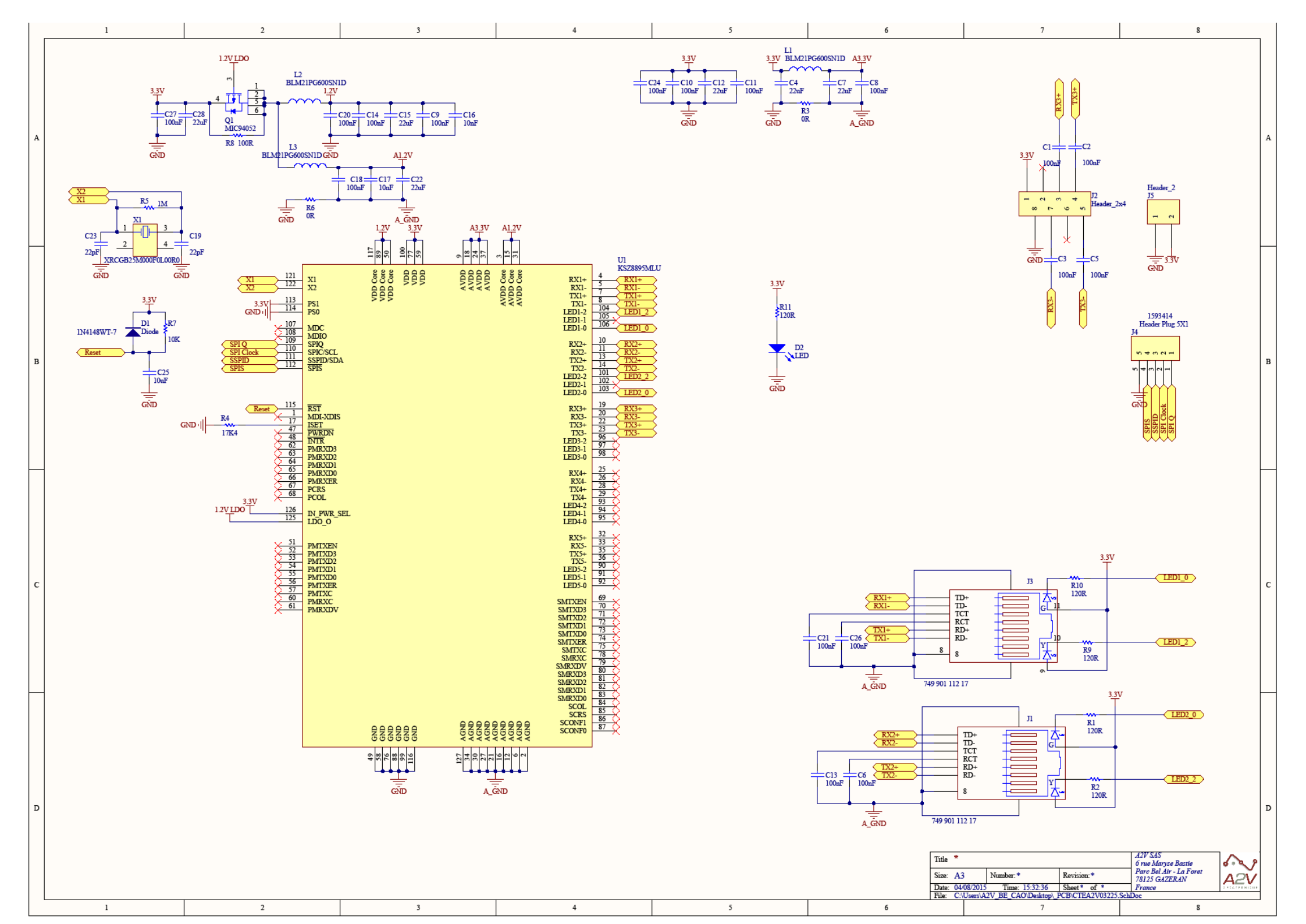
Aussi les signaux Ethernet seront récupérés sur le PCB directement à la place du connecteur Ethernet, imposant alors un emplacement pour un connecteur de type « header » à cet endroit.

2. Réalisation technique

Le chip qui a été choisi pour réaliser le Switch Ethernet est le KSZ8895MLU de chez Micrel. Il permet d'avoir nos 3 ports Ethernet en Switch et d'être configurable à l'aide de résistance de pull-down et de pull-up. Il ne demande donc pas de configurations complexes. Le vrai défi de la carte réside dans le placement des composants et le routage.

Souhaitant avoir nos deux connecteurs Ethernet en vis-à-vis, nous n'avons eu d'autres choix que de les placer de la manière suivante sur la carte :

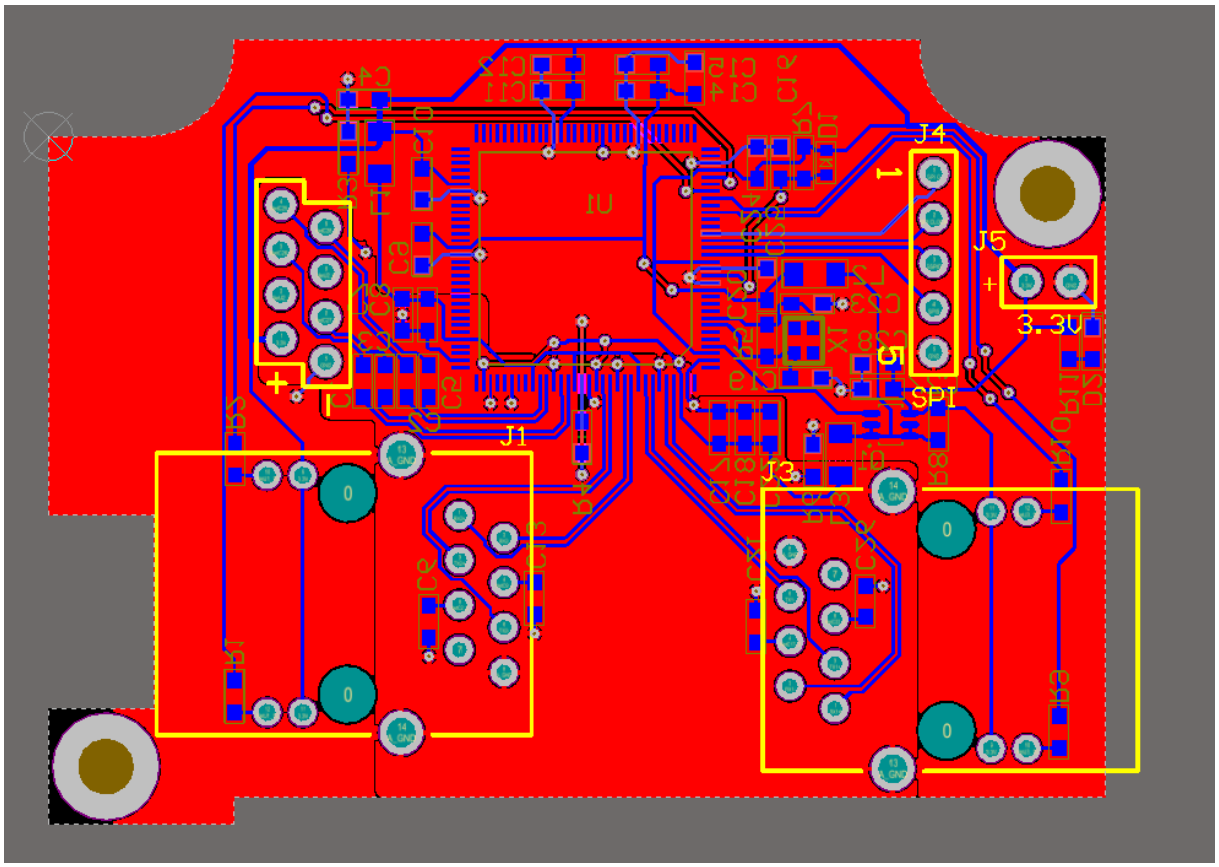




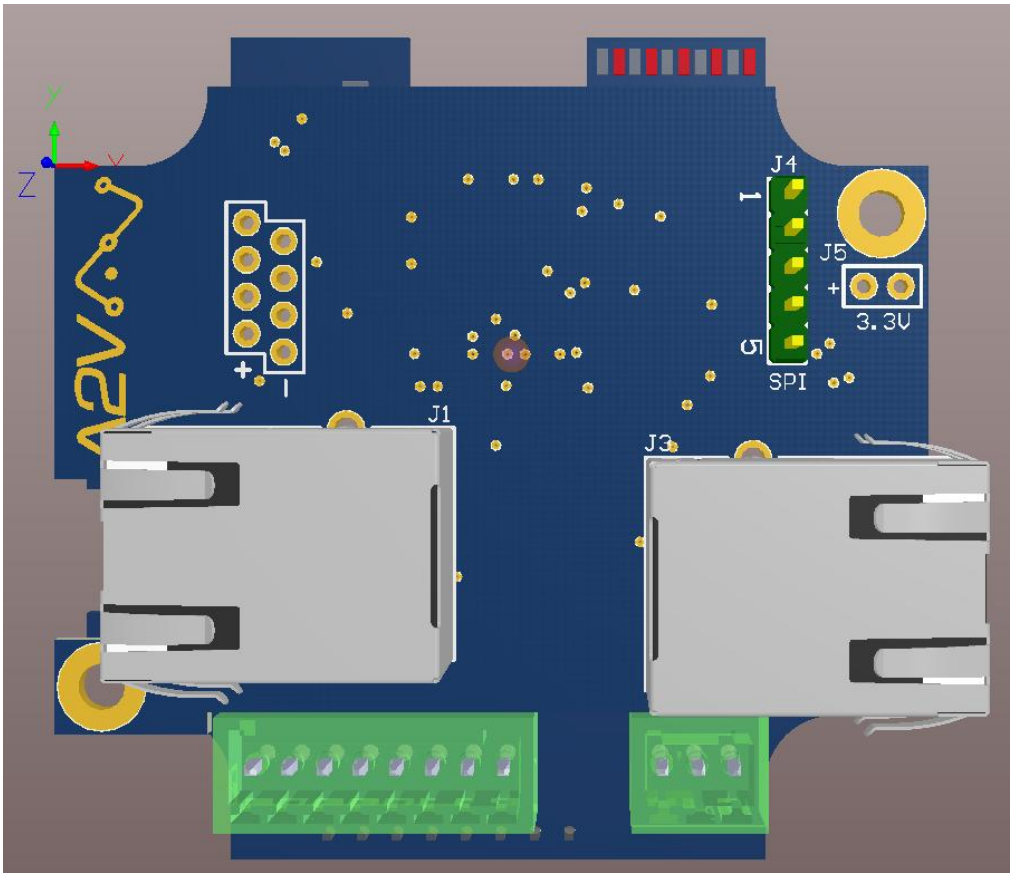
Title *			A2V SAS 6 rue Maryse Bastie Parc Bel Air - La Foret 78125 GAZERAN France
Size: A3	Number: *	Revision: *	
Date: 04/08/2015	Time: 15:32:36	Sheet * of *	
File: C:\Users\A2V_BE_CAO\Desktop_PCB\CTEA2V03225.SchDoc			

Nous avons choisi de réaliser un PCB deux couches afin de minimiser les coûts, également il m'a été imposé lors de ce projet de ne placer les composants que sur une seule face du PCB, comme cela lors de la fabrication en série, notre sous-traitant n'aura pas à retourner la carte pour placer les composants non traversant ce qui est relativement chère.

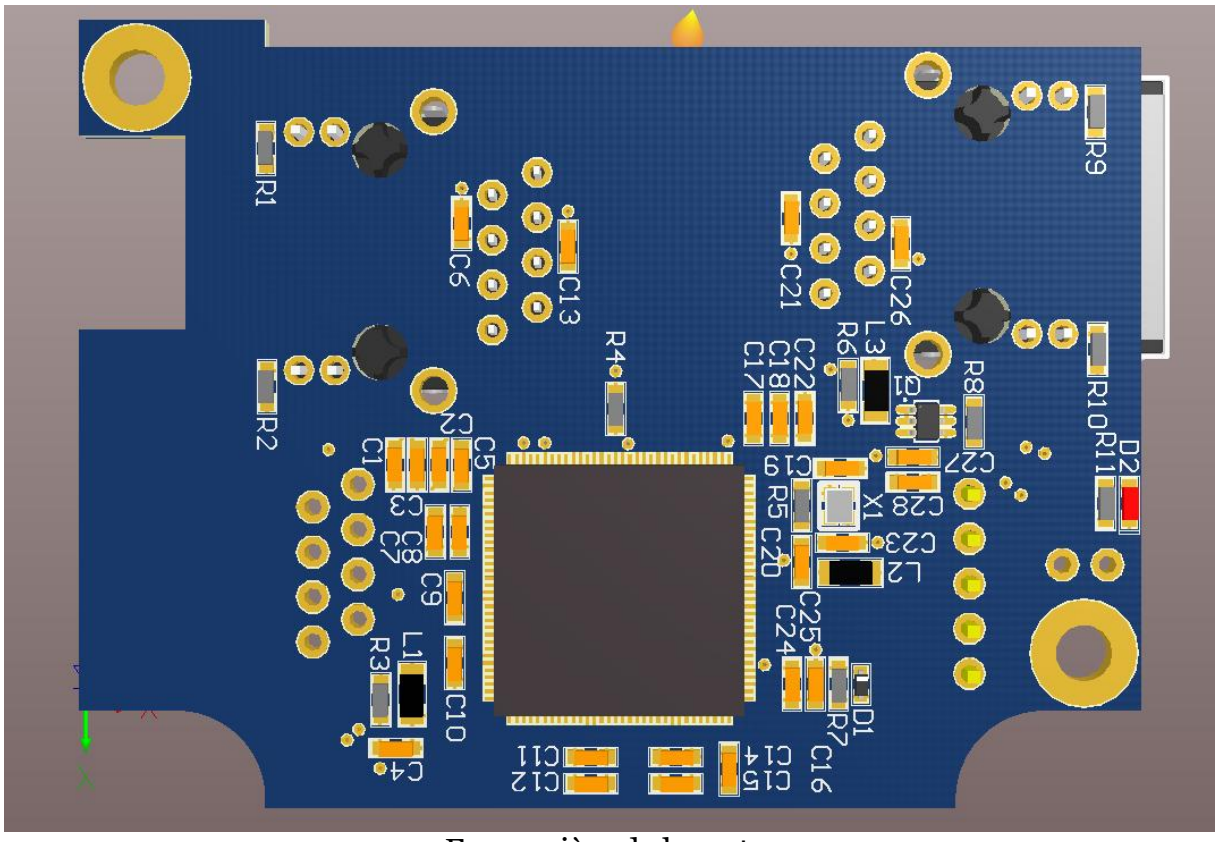
Après placement de tous les composants puis du routage à la main, je suis arrivé au résultat suivant :



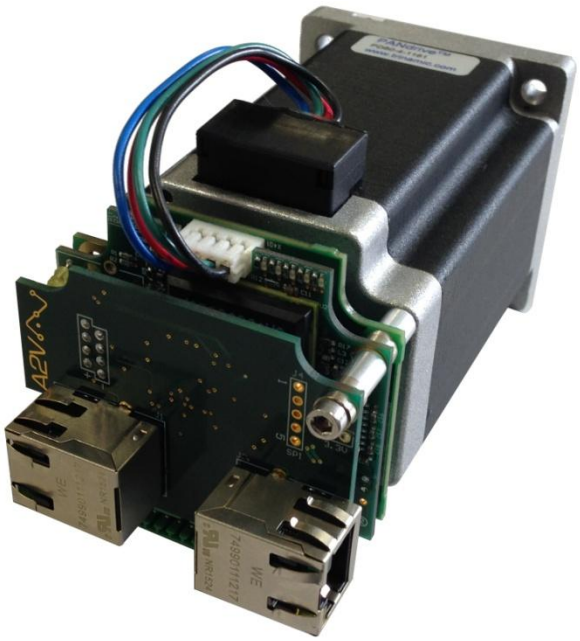
L'Ethernet est constitué de 2 paires différentiels, on peut donc voir en prêtant attention sur la carte que les signaux allant aux connecteurs Ethernet on été routé de telle sorte à être les plus proches deux à deux et de longueur la plus similaire possible. On peut également apercevoir que la zone rouge est divisée en deux zones distinctes, ce sont les plans de masses. L'Ethernet étant relativement sensible il dispose de son propre plan de masse et de ces propres alimentations filtrés afin de ne pas perturber toute la carte avec ces signaux hautes fréquences et ne pas être lui-même perturbé.

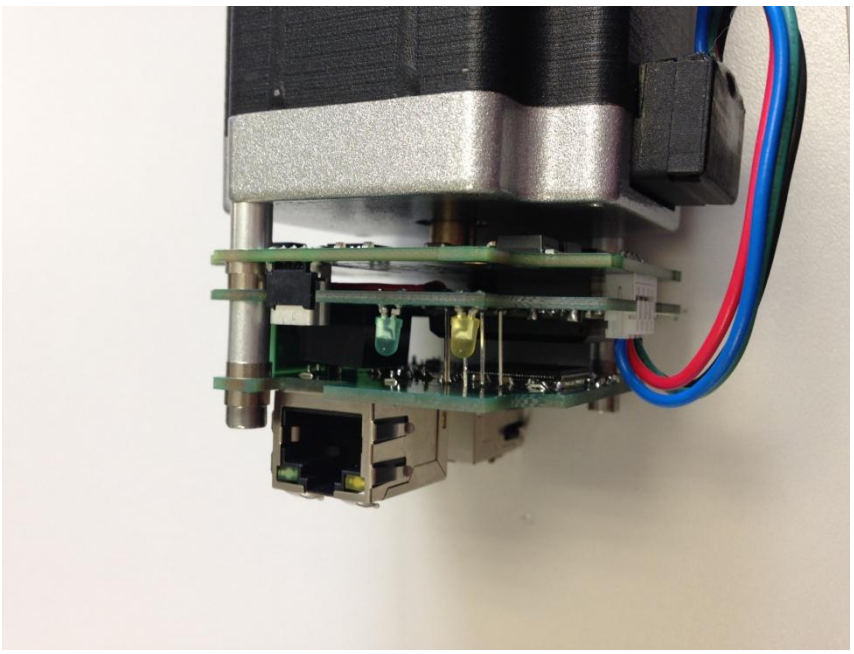


Face avant de la carte



Face arrière de la carte





Photos du moteur assemblée avec la carte

3. Programmation du moteur

Une fois la carte réalisée et montée sur le moteur, nous nous sommes attelés à la validation de celle-ci. Nous nous sommes donc assurés du bon fonctionnement du switch ethernet. Nous aurions pu nous arrêter là, le projet était pour ainsi dire validé, seulement nous avons pensés judicieux développer une partie une partie du code codesys de ce moteur afin d'avoir un exemple à montrer à notre client.

Ce client achetait pour le moment des moteurs ne disposant pas de la fonctionnalité modbus et encore moins d'un switch, ce produit l'intéressait donc particulièrement puisqu'en chainant les moteurs les uns aux autres avec un seul câble, son câblage deviendrait bien plus simple et moins couteux.

Je me suis donc attelé à porter le programme de ces anciens moteurs pour celui-ci en utilisant les fonctionnalités modbus et en lui developpant des blocs "codesys".

Mon travail consistait donc à développer un asservissement en position pour le moteur, et de permettre à l'utilisateur de choisir deux points consignes : une position haute et une position basse. L'utilisateur devait pouvoir actionner la rotation sur la position haute ou basse depuis n'importe où en envoyant des commandes modbus sur le réseau, il devait également être capable de modifier les valeurs de positions consignes, la vitesse de rotation, ou encore la fenêtre de validité de la consigne. Egalement deux variables modbus permettent de savoir si le moteur est dans les positions consignes hautes/basses ou non. Avec un affichage de ces variables sur le moteur à l'aide de LED.

Mon programme se décompose donc en plusieurs parties, d'une part les blocs de motion control permettant d'avoir un asservissement en position sur le moteur :

```
0001 PROGRAM Controller
0002 VAR
0003     position_Error: DINT;
0004     moving: BOOL;
0005     speed_Cmd: DINT;
0006     Inst_CurTime: CurTime;
0007     timevalold: SysTime64;
0008     cycletime: DWORD;
0009 END_VAR
```

```
0001 Inst_CurTime(SystemTime := timevalold);
0002
0003 positionAbsolute_Current := TMCM_ReadPositionAbsolute();
0004
0005 IF ((positionAbsolute_Previous - positionAbsolute_Current) < -512) THEN
0006     position_Current := position_Current - 1023;
0007 ELSIF ((positionAbsolute_Previous - positionAbsolute_Current) > 512) THEN
0008     position_Current := position_Current + 1023;
0009 END_IF
0010
0011 position_Current := position_Current + (positionAbsolute_Current - positionAbsolute_Previous);
0012
0013 IF (PID_Enable) THEN
0014     position_Error := position_Target - position_Current;
0015
0016     IF (ABS(position_Error) > position_Window) THEN
0017         speed_Cmd := REAL_TO_DINT(position_Error * PID_Divisor);
0018         speed_Cmd := MIN(speed_Cmd, speed_Maximum);
0019         speed_Cmd := MAX(speed_Cmd, -speed_Maximum);
0020         TMCM_MoveVelocity(speed_Cmd);
0021         moving := TRUE;
0022     ELSIF (moving) THEN
0023         PID_Enable := FALSE;
0024         TMCM_Stop();
0025         moving := FALSE;
0026     END_IF;
0027 ELSIF (moving) THEN
0028     TMCM_Stop();
0029     moving := FALSE;
0030 END_IF;
0031
0032
0033 positionAbsolute_Previous := positionAbsolute_Current;
0034
0035 Inst_CurTime(SystemTime := timeval);
0036 cycletime := timeval.ulLow - timevalold.ulLow;
0037
0038
0039
0040
```

```
0001 VAR_GLOBAL
0002     positionAbsolute_Current: DINT;
0003     position_Current: DINT;
0004     position_Target: DINT;
0005     positionAbsolute_Previous: DINT;
0006     position_Window: DINT := 2;
0007     PID_Enable: BOOL := FALSE;
0008     PID_Divisor: DINT := 5;
0009     speed_Maximum: DINT := 2047;
0010
0011
0012     timeval: SysTime64;
0013 END_VAR
0014
0015
```

Ainsi que le programme permettant de communiquer en modbus, et de répondre au cahier des charges :

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     init: INT :=0;
0004     input :BOOL;
0005     position1 : DINT :=269;
0006     position2 : DINT := 755;
0007     windows : DINT :=3;
0008
0009     command: INT;
0010     command_ID: USINT;
0011     command_Type: USINT;
0012     command_Data: DINT;
0013 END_VAR
0014
```

```
0001
0002 IF init = 0 THEN
0003     positionAbsolute_Previous := positionAbsolute_Current;
0004     position_Current := positionAbsolute_Current;
0005     position_Target := 0;
0006     PID_Enable := TRUE;
0007     init := 1;
0008 ELSIF init = 1 THEN
0009     (* on attends qu'il est fait sa prise d'origine *)
0010     IF (PID_ENABLE = FALSE) THEN
0011         init := 2;
0012     END_IF;
0013 ELSE
0014
0015 input := MDB_IN_000.0;
0016
0017 IF (WORD_TO_DINT(MDB_IN_001) = 1) THEN
0018     position1 := WORD_TO_DINT(MDB_IN_002);
0019     position2 := WORD_TO_DINT(MDB_IN_003);
0020     windows := WORD_TO_DINT(MDB_IN_004);
0021     speed_Maximum := WORD_TO_DINT (MDB_IN_005);
0022 END_IF
0023
0024
0025 MDB_OUT_102 := DINT_TO_WORD(position1);
0026 MDB_OUT_103 := DINT_TO_WORD(position2);
0027 MDB_OUT_104 := DINT_TO_WORD(windows);
0028 MDB_OUT_105 := DINT_TO_WORD(speed_Maximum);
0029
0030
0031 IF (input) THEN
0032     position_Target := position2;
0033 ELSE
0034     position_Target := position1;
0035 END_IF;
0036
0037 IF (position_Current < position_Target + Windows AND position_Current > position_Target - Windows) THEN
0038     IF (input) THEN
0039         USER_LED_1 := FALSE;
0040         USER_LED_2 := TRUE;
0041     ELSE
0042         USER_LED_1 := TRUE;
0043         USER_LED_2 := FALSE;
0044     END_IF
0045 ELSE
0046     PID_Enable:=TRUE;
0047     USER_LED_1 := FALSE;
0048     USER_LED_2 := FALSE;
0049 END_IF;
0050
0051 MDB_OUT_100.0 := USER_LED_1;
0052 MDB_OUT_100.1 := USER_LED_2;
0053
0054 END_IF
```

III. Shunt Regulator

1. Présentation du projet

Une résistance de shunt est un circuit de protection qui permet d'éviter des surtensions sur bus d'alimentation. Ce type de protection est particulièrement importante et utilisé chez A2V pour les raisons suivantes : lorsqu'un moteur en rotation freine, ou alors lorsqu'un moteur est entrainé en rotation par un autre moteur par exemple alors il va récupérer de l'énergie sur le bus d'alimentation. Le problème est que cette énergie n'est pas régulée naturellement pouvant provoquer de grosses surtensions et faire chuter le bus d'alimentation, ou faire cramer des composants alimenter par celui-ci.

La carte qui m'a été demander de développer ici fonctionne de la manière suivante : A l'aide d'un circuit analogique (un AOP), on compare la tension d'alimentation à une tension de référence (obtenue avec une diode zener), si celle-ci est dépassée, alors on active un mosfet qui dissipe l'énergie dans une résistance de puissance en utilisant l'effet Joule. La résistance chauffe en consommant de l'énergie, ce qui permet de faire descendre la tension du bus, ainsi on évite toute surtension.

2. Réalisation technique

Je ne dispose malheureusement pas des schémas électroniques de la carte ni des 3D de cette dernière. Cette carte ayant été terminée la dernière semaine de mon stage et n'ayant pu la tester moi même.

Cependant j'ai, depuis la fin de mon stage eu des retours de mon maitre de stage m'indiquant qu'elle était fonctionnelle et que des clients en avaient déjà commandé pour test. Un avenir radieux s'annonce donc peut être pour ce projet.

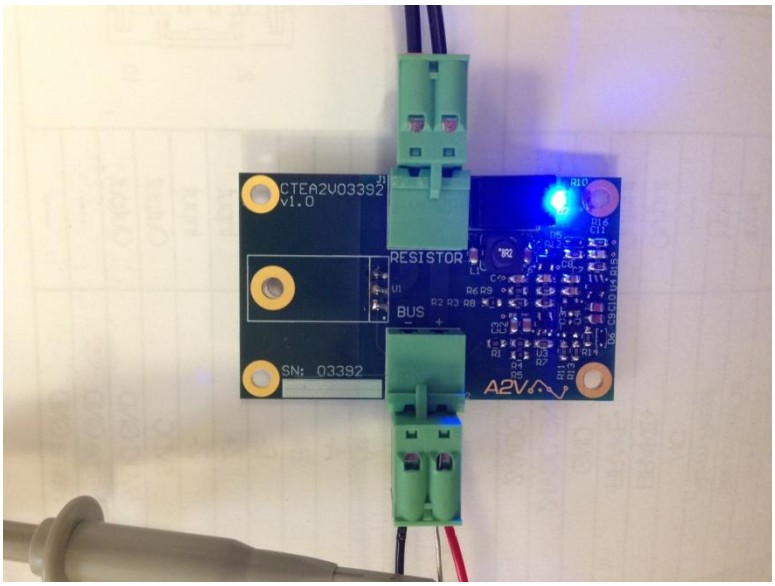


photo de la carte lors des tests

Conclusion :

Au travers de ce stage, j'ai eu la possibilité de participer à de nombreux projets très enrichissants qui ont constitué pour moi une nouvelle expérience pleine d'intérêts. Ils ont également contribué à renforcer mes approches de la vie active et du monde du travail.

J'ai pu acquérir des compétences dans de nombreux domaines. Tout d'abord, j'ai pu appréhender une nouvelle approche des cartes de commandes et des automates à l'aide des nombreux automates que j'ai utilisé, j'ai notamment appris à utiliser le logiciel codesys et les différents langages de programmation associés. Le tuning de moteurs fut également une très bonne mise en pratique des connaissances acquises à l'école comme l'automatisme et le pilotage de servomoteurs.

Par ailleurs, j'ai pu développer ma connaissance du logiciel Altium Designer et de la réalisation de PCB. Du point de vue intégration des éléments et conception mécanique, j'ai pu apprendre beaucoup grâce à l'équipe qui m'entourait.

D'un autre côté, l'expérience humaine a également été très enrichissante. J'ai pu découvrir la façon dont sont gérés les projets dans une entreprise, participer à des réunions.

Mon stage m'a permis d'avoir une très large vision de ce que sera mon métier d'ingénieur. J'ai été ravi de le faire au sein de la société A2V, où l'ambiance de travail est très agréable. Les projets étant tous vraiment intéressants, cela m'a permis de travailler tout en étant motivé d'obtenir de bons résultats.

Annexe :

Sources :

A2V Mécatronique :

http://www.a2v.fr/

Trinamic :
http://trinamic.com/

Arcus Technology :
http://www.arcus-technology.com/

Altium Designer :
http://www.altium.com/

Visual Studio :
http://www.microsoft.com/france/visual-studio/

L'étudiant évalue son stage :

ESME sudria Paris

L'ETUDIANT(E) EVALUE SON STAGE

Vous devez, en complétant ce document, donner votre opinion sur le déroulement de votre stage en entreprise. A restituer obligatoirement au service des stages S.V.P :
vauxbidon@esme.fr ou 38 Rue Molière -94200 Ivry sur Seine

Prénom NOM : Quentin Mercier **CLASSE :** M1B

Intitulé du stage :

NOM ENTREPRISE : A2V Mécatronique

Merci d'entourer la lettre choisie – NE = non évalué

Respect du projet pédagogique défini par l'Ecole :	A	B	(C)	D	NE
Accueil et moyens pour réussir la mission :	(A)	B	C	D	NE
Suivi par le tuteur :	(A)	B	C	D	NE
Informé des règles, codes et culture de l'entreprise :	(A)	B	C	D	NE
Aide à l'Intégration :	(A)	B	C	D	NE
Accès aux informations nécessaires :	(A)	B	C	D	NE
Aide dans l'acquisition des compétences nécessaires :	(A)	B	C	D	NE
Suivi régulier des travaux :	(A)	B	C	D	NE
Evaluation du travail effectué :	(A)	B	C	D	NE
Conseils sur le projet professionnel :	(A)	B	C	D	NE

Signature : 

Important !! Pour les Master 2 Vous devez remettre impérativement ce document au service des stages avant la soutenance et compléter les éléments ci-dessous :

- Avez-vous, à ce jour , une ou des propositions d'embauche en cours ?
- Si oui chez qui ?
- Si oui salaire annuel ?

Service des stages -CdVB

Evaluation par l'entreprise :

ESME sudria Paris

Evaluation d'expérience professionnelle en entreprise

(stage ou CDD de BAC +1 à Bac +4)

Nom de l'ETUDIANT : MERCIER **Prénom :** Quentin

Classe : TT1B

Nom de la SOCIETE : A2V - Mecatronique .

	Très bien	*Bien*	*Moyen*	*Insuffisant*	*Non évalué*
Ponctualité/assiduité	X				
Respect des délais	X				
Adaptation aux usages de l'entreprise	X				
Autonomie		X			
Sens de la responsabilité	X				
Sens de l'organisation		X			
Maîtrise technique ou sens pratique		X			
Intégration rapide de nouvelles connaissances		X			
Sens critique		X			
Initiative		X			
Clarté d'expression	X				
Ecoute	X				
Souci de s'améliorer	X				
Attitude vis à vis de l'équipe	X				
Rapport de stage					

Appréciation globale sur le déroulement du stage : très bon stagiaire
Compétent
Sérieux
Facile à vivre.

Note sur 20 : 17

Nom du signataire et fonction :
Eric REGNARD
Président

Signature :

A2V SAS
6 rue Marcel Paul Parc Bel Air la Clef Saint Pierre 78990 ELANCOURT
Tél. : 01 34 83 97 38 - Fax : 01 34 83 97 38
www.a2v.fr - RCS B380995217
Eric REGNARD Président

A restituer au service des stages : Claudine de Vaux-Bidon ESME Sudria 38 rue Molière 94200 Ivry sur Seine- email : vauxbidon@esme.fr
Fax : 33 (0)1 56 20 62 62

Service des stages ESME Sudria 2015