

Maquette mécatronique durcie d'ascenseur 5 étages

Rapport intermédiaire de projet de fin d'études

Louis CHAUCHARD
Romain IMBERT
Informatique, Microélectronique et Automatique
2015-2016

M. Blaise CONRARD
M. Xavier REDON
M. Thomas VANTROYS

Sommaire

Introduction.....	2
Présentation du projet.....	3
1. Contexte.....	3
2. Objectif du projet.....	3
Cahier des charges.....	4
1. Description du projet.....	4
2. Choix techniques.....	6
Travail effectué de septembre à décembre.....	7
1. Étude du projet.....	7
2. Choix des composants.....	7
3. Gestion et test du moteur.....	7
4. Gestion partie Arduino et partie Automate.....	9
5. Réalisation des premiers tests.....	14
6. Analyse et étude de la réalisation de la maquette.....	15
Problèmes rencontrés.....	16
Planning prévisionnel pour la période de Janvier - Février.....	17
Bibliographie.....	19

Introduction

Ce rapport a pour but d'exposer le travail réalisé dans le cadre du projet de fin d'études de cinquième année d'IMA (semestre 9-10). Ce rapport est réalisé à mi-parcours afin de détailler le travail déjà effectué ainsi que son avancement. Les projets de dernière année permettent de travailler pendant plusieurs heures sur un sujet et donc de mettre en pratique les connaissances acquises durant notre cursus ainsi que de développer de nouvelles compétences.

Pour ce projet, nous avons choisi de nous pencher sur le sujet de « La maquette mécatronique durcie d'ascenseur 5 étages ». Ce sujet initialement proposé par M. Conrard consiste à faire l'étude, mais aussi l'élaboration de deux maquettes d'ascenseur 5 étages.

Afin d'exposer au mieux le début de l'élaboration de ce projet, nous ferons tout d'abord sa présentation. Dans une seconde partie, nous développerons le plus précisément possible son déroulement et le travail déjà accompli. Enfin, nous terminerons par les difficultés rencontrées ainsi que le détail du travail qu'il reste à réaliser afin d'obtenir les maquettes finales souhaitées.

Présentation du projet

1. Contexte

Actuellement, une seule maquette d'ascenseur est proposée pendant les séances de travaux pratiques d'automatisme. D'autres maquettes permettront, tout d'abord d'augmenter le nombre de binômes travaillant sur le système de l'ascenseur en automatisme. Elles offriront aussi un exemple supplémentaire lors de portes ouvertes ou de présentation de la filière Informatique, Microélectronique et Automatique et plus précisément des cours d'automatisme.

2. Objectif du projet

L'objectif de ce projet est donc de réaliser 2 maquettes d'ascenseurs 5 étages supplémentaires destinées à être utilisées en Travaux Pratiques et elles seront commandables par automate programmable.

Dans le but de réaliser un système fiable et robuste, un soin doit être apporté à la sûreté de fonctionnement.

En effet, les différents composants utilisés doivent, soit avoir une fiabilité élevée, soit être simples et rapides à changer.

Cahier des charges

1. Description du projet

L'utilisation de redondances matérielles pour certains composants permettra d'améliorer la fiabilité du système. En effet, si un de ces composants est amené à dysfonctionner, l'ajout d'un second permettra de prendre le relais avant un dépannage.

Toujours dans l'objectif d'obtenir un système fiable et robuste, un mode permettant la détection de panne ou de défaut doit être réalisé. Ce mode déclenché par l'appui d'un bouton-poussoir "test" permettra de faire fonctionner les différents voyants. Enfin, ces maquettes doivent pouvoir être connectées à un automate fonctionnant avec des tensions industrielles (0-24V).

Une plaque en plexiglas sera installée pour la protection de l'utilisateur lors du mouvement de la partie mobile de la maquette (cabine).

Deux capteurs de fin de course bas et haut de technologie mécanique seront installés pour éviter la détérioration de la maquette. Ces deux dispositifs seront câblés directement sur l'alimentation du moteur afin d'obtenir une sécurité supplémentaire en cas de problème venant de l'Arduino et de l'automate programmable. On devra ajouter un bouton qui permettra de déplacer la cabine afin qu'elle ne soit plus en sur course pour pouvoir repartir dans un fonctionnement normal. Cette action sera réalisée à l'aide de l'Arduino. Il sera donc nécessaire de câbler les deux capteurs de fin de course sur les entrées de l'Arduino.

Un bouton d'arrêt d'urgence sera à disposition de l'utilisateur pour couper l'alimentation du moteur en cas de dysfonctionnement.

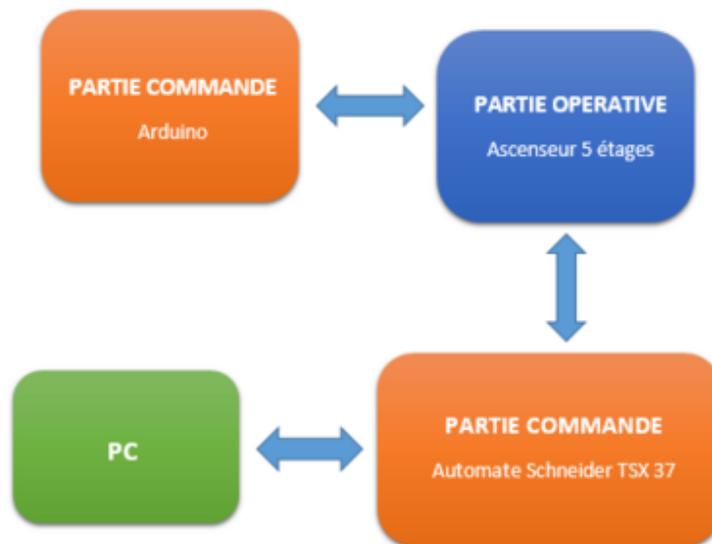


Figure 1 : Synoptique de fonctionnement

Fonctionnement général de la maquette

La maquette possédera deux modes de fonctionnement en plus d'un mode "défaut". Ces deux modes de fonctionnement seront soit l'ascenseur relié à l'automate pour réaliser un TP d'automatismes, soit un mode "démonstration" où l'ascenseur ne sera géré que par le biais de l'Arduino.

Ce dernier mode ne nécessitera donc pas de programmation préalable de l'automate pour fonctionner.

➤ Fonctionnement de l'ascenseur relié à l'automate (pour les TP)

Ce mode de fonctionnement permettra de réaliser un scénario particulier en créant un programme qui sera transféré ensuite dans l'automate qui lui-même actionnera les différents composants de la maquette.

➤ Fonctionnement de l'ascenseur en mode "démonstration" qui sera utilisé pour les journées portes ouvertes de Polytech' Lille.

L'utilisateur appelle l'ascenseur à un étage x . Il choisit entre le bouton poussoir bas ou haut. Si l'utilisateur est à l'étage 1, il n'aura que le choix d'appuyer sur le bouton poussoir d'appel haut. Si l'utilisateur est à l'étage 5, il n'aura que le choix d'appuyer sur le bouton poussoir d'appel bas. C'est deux obligations permettent d'éviter un dysfonctionnement de la maquette. Dès que l'ascenseur est arrivé à cet étage x , l'utilisateur entre dans la cabine et il appuiera sur le bouton poussoir concernant l'étage désiré. L'ascenseur monte ensuite à l'étage désiré. L'utilisateur sort alors de la cabine. Après une durée t , cet utilisateur appelle à nouveau la cabine avec le bouton poussoir appel palier.

➤ Fonctionnement de l'ascenseur en mode "détection de défauts"

A tout moment de l'utilisation de la maquette, il doit être possible via un bouton "test" de vérifier l'état de tous les voyants lumineux de la maquette afin de vérifier leur bon fonctionnement.

2. Choix techniques

Nous utiliserons des capteurs inductifs à effets Hall dans le but d'éviter les contacts et une usure des capteurs et ainsi réduire la maintenance de ceux-ci.

Chaque étage possède deux boutons poussoirs lumineux d'appel palier. Un bouton servira à l'appel palier vers les étages inférieurs et le second gèrera l'appel palier vers les étages supérieurs. L'étage 1 ne possèdera pas de bouton poussoir d'appel vers les étages inférieurs. L'étage 5 ne possèdera pas de bouton poussoir d'appel vers les étages supérieurs.

Les 5 étages seront équipés d'un détecteur de positionnement cabine. Il y aura aussi un capteur de fin de course position basse à l'étage 1 et un capteur de fin de course position haute à l'étage 5 afin de limiter le déplacement de la cabine en cas de problème. Ces deux capteurs de fin de course seront choisis avec une technologie différente dans le but d'ajouter une sécurité en cas de dysfonctionnement. Nous utiliserons des capteurs mécaniques, car la technologie diffère de celle utilisée pour chaque étage.

Il y aura 5 boutons poussoirs lumineux sur un pupitre matérialisant les appels provenant de la cabine.

Un bouton d'arrêt d'urgence sera implanté sur le pupitre pour couper l'alimentation du moteur permettant la translation de la cabine.

Nous utiliserons un moteur pouvant être piloté dans les deux sens de rotation permettant la montée ou la descente de la cabine.

Le châssis de la maquette sera réalisé avec du bois et du plexiglas.

Travail effectué de septembre à décembre

1. Étude du projet

La toute première étape du projet était la pré étude du sujet qui se décomposait en plusieurs étapes.

Tout d'abord la recherche d'informations a été portée sur le sujet et l'analyse de l'état de l'art. Une fois cette étape préliminaire réalisée, une seconde étape fut nécessaire. Cette étape résidait dans la recherche des divers moyens et solutions possibles, mais surtout envisageables pour réaliser à bien notre projet. Cette étape nous a amenés à avoir différentes réflexions et à rédiger des questions et interrogations sur le sujet, en fonction des attentes de notre responsable de projet M. Conrard. Plusieurs échanges avec lui, nous ont permis de spécifier de nombreux points dans le but d'élaborer une première ébauche de cahier des charges.

Les points abordés étaient la définition de la taille de la maquette, l'utilisation de la maquette selon deux modes et la nécessité de développer une procédure de test pour détecter certains défauts.

Suite à ces spécifications, nous avons effectué la rédaction du cahier des charges.

2. Choix des composants

Une fois la rédaction du cahier des charges achevée, l'étape suivante fut la recherche des composants à utiliser pour la conception des deux maquettes.

Une première étape assez conséquente en termes de temps fut donc l'analyse et la recherche des composants à utiliser selon les critères de rapport qualité/prix et de fiabilité étant donné que l'on recherche à concevoir un système assez robuste, dans le but d'effectuer le plus rapidement possible une commande.

3. Gestion et test du moteur

Après l'obtention du servomoteur, nous avons débuté l'étude de celui-ci.

Suite à l'analyse de la documentation technique pour comprendre le fonctionnement de ce moteur, il apparaît que l'actionneur devra être alimenté entre 7 et 10 V. Il est contrôlé via un bus série et un système d'adresse.

Le moteur possède 2 connecteurs de 3 PINs:

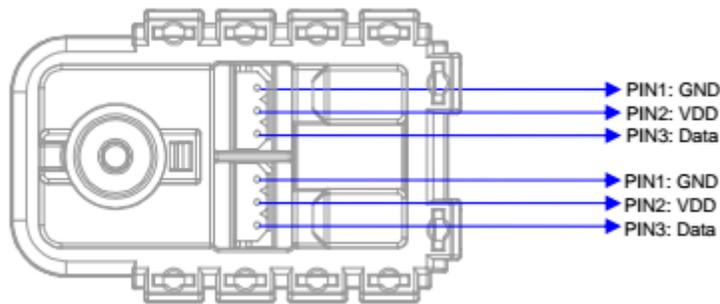


Figure 2 : Illustration du moteur AX12+

Le second connecteur est nécessaire pour l'utilisation de plusieurs moteurs en parallèle.

Au vu de notre application, ce second connecteur ne sera pas utilisé.

Cet actionneur sera donc commandé par l'Arduino. Il sera alors nécessaire d'utiliser les pin 0(RX0) et pin 1 (TX0) par exemple. Le connecteur Data transmettra les données série TX et RX alternativement.

Pour notre application, l'utilisation de la pin 0 (RX0) n'est pas réellement nécessaire, car nous ne souhaitons pas obtenir d'informations de retour.

Le câblage de l'Arduino est le suivant, mais les entrées "Data Control" et "RX" ne seront pas utilisées. La commande du moteur sera effectuée par l'entrée TX (verte). L'alimentation se fait par un dispositif extérieur.

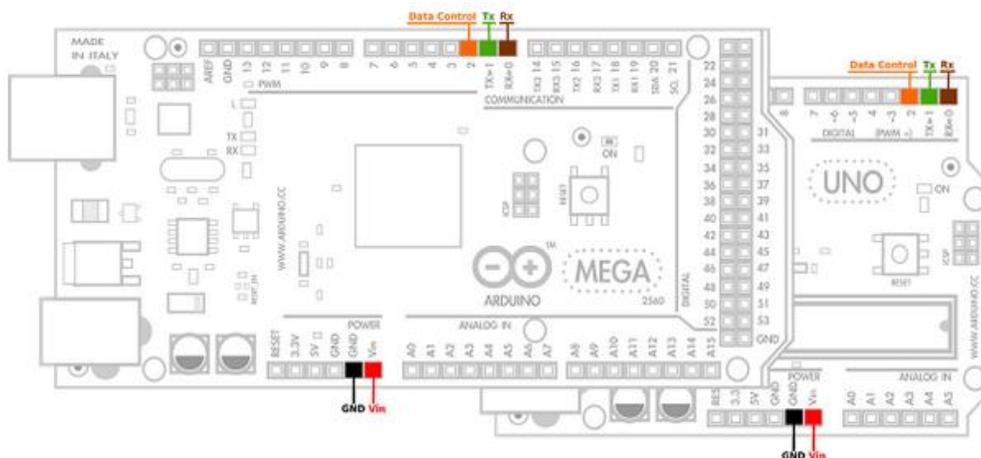


Figure 3 : Câblage du moteur AX12 avec Arduino

Nous avons effectué des tests pour la commande du moteur AX12+ via l'Arduino. Le servomoteur ne sera commandé que par l'Arduino.

L'automate programmable aura deux sorties qui permettront de donner l'information de rotation dans le sens (+) et le sens (-) du moteur.

Pour la commande de ce servomoteur, nous avons utilisé la librairie "Dynamixel.h". Cette librairie met à disposition des fonctions suivantes :

`Dynamixel.setEndless(ID,Status)`: permet d'activer le fonctionnement du moteur en "mode continu".

`Dynamixel.turn(ID,Side,Speed)` : permet de faire tourner le moteur de manière continue en précisant le sens et la vitesse.

`Dynamixel.move(ID,Position)` : permet de réaliser une rotation du moteur en indiquant la position en paramètre.

Le moteur utilisé est repéré par son identifiant (ID), car il est possible de commander plusieurs moteurs simultanément.

Pour notre application, nous allons seulement envoyer des commandes de l'Arduino vers le moteur donc nous utiliserons seulement le port TX0 de l'Arduino qui sera connecté au port DATA du servomoteur.

Nous avons réalisé une protection pour l'alimentation du moteur. Nous avons ajouté un fusible de 2000 mA pour limiter le courant en cas de dysfonctionnement et une diode de redressement pour protéger l'alimentation contre les tensions négatives.

Pour notre application, il est nécessaire d'utiliser le servomoteur en fonctionnement continu. Après de multiples essais, le moteur a enfin un fonctionnement convenable pour notre application. Il tourne en continu grâce à la fonction "setEndless()".

Nous avons ensuite utilisé une interruption après l'appui sur un bouton poussoir (génération d'un front montant) dans le but de réaliser la fonction "appel cabine". Après cet appui, le moteur se met à tourner dans un des deux sens de rotation.

Ayant reçu les composants commandés, nous avons réalisé un premier montage, servant de prototype pour la pré-soutenance. (Commande du moteur selon l'entrée, capteur à effet Hall ou bouton poussoir)

4. Gestion partie Arduino et partie Automate

Dans cette partie, nous allons développer l'étude de la gestion partie automate et partie Arduino.

Comme nous l'avons remarqué ultérieurement, une "difficulté" du projet est la gestion de deux tensions de fonctionnement. En effet, la partie automate doit fonctionner en 24V tandis que la partie gérée par l'Arduino a une tension de fonctionnement de 5V.

Il faut donc trouver un moyen de manipuler ces deux tensions en évitant surtout que l'Arduino reçoive des tensions supérieures à 5V.

Une première solution envisagée est l'utilisation d'un relais Reed, qui permettrait d'éviter des liens entre 24V et 5V. Cependant, étant donné le nombre d'entrées-sorties

nécessaires (boutons, voyants et capteurs) et le prix (un peu plus de deux euros pour un seul composant), cette solution est trop coûteuse et donc nous ne l'avons pas retenue.

Une seconde solution possible et envisageable en termes de coût est l'utilisation d'optocoupleurs.

Nous avons utilisé cette seconde solution qui est la plus judicieuse du fait du nombre de composants que nous devons utiliser pour la gestion des deux tensions.

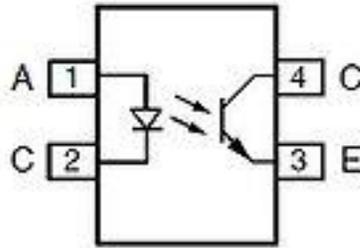


Figure 4 : Schéma d'un Optocoupleur

Ces composants permettent l'échange d'un signal électrique tout en réalisant une isolation galvanique, c'est-à-dire qu'il n'existe aucune liaison par conducteur électrique. Cela va donc permettre de limiter les tensions sur l'Arduino.

Le premier montage ci-dessous utilise un optocoupleur pour fournir une information à l'automate en 0-24V. La tension envoyée à l'Arduino est directement la tension fournie par le capteur à effet Hall.

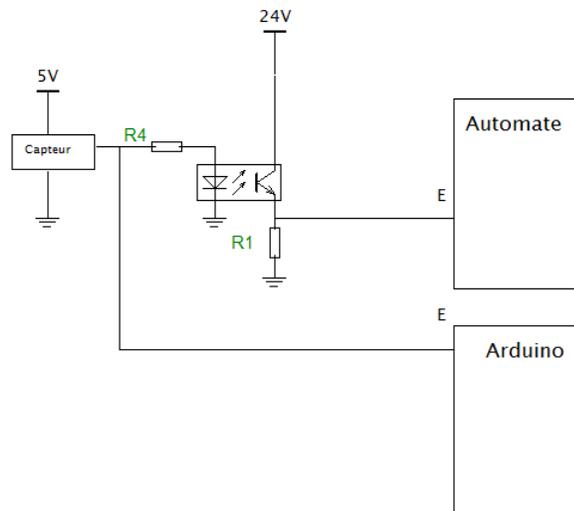


Figure 5 : Adaptation des niveaux de tensions pour le capteur à effet Hall

Loi des mailles 1:

V_d = tension aux bornes de la diode

Vcapteur: tension délivrée par le capteur à effet Hall

i4: courant traversant la résistance R4

$$V_{\text{capteur}} = R4 \cdot i4 + V_d$$

On fixe i4= 20mA, on peut donc calculer la valeur de la résistance R4.

$$R4 = \frac{V_{\text{capteur}} - V_d}{i4} = \frac{5 - 1.2}{0.020} = 190\Omega$$

Lois des mailles 2:

$$V_{cc} = 24V$$

Vce0: tension entre collecteur et émetteur du transistor NPN

i1=courant traversant la résistance R1

$$V_{cc} = V_{ce0} + R1 \cdot i1$$

On fixe le courant i1 à 1mA

$$R1 = \frac{V_{cc} - V_{ce0}}{i1} = 23.9 k\Omega$$

Pour valider ce montage, nous avons réalisé une simulation de celui-ci à l'aide du logiciel LT Spice:

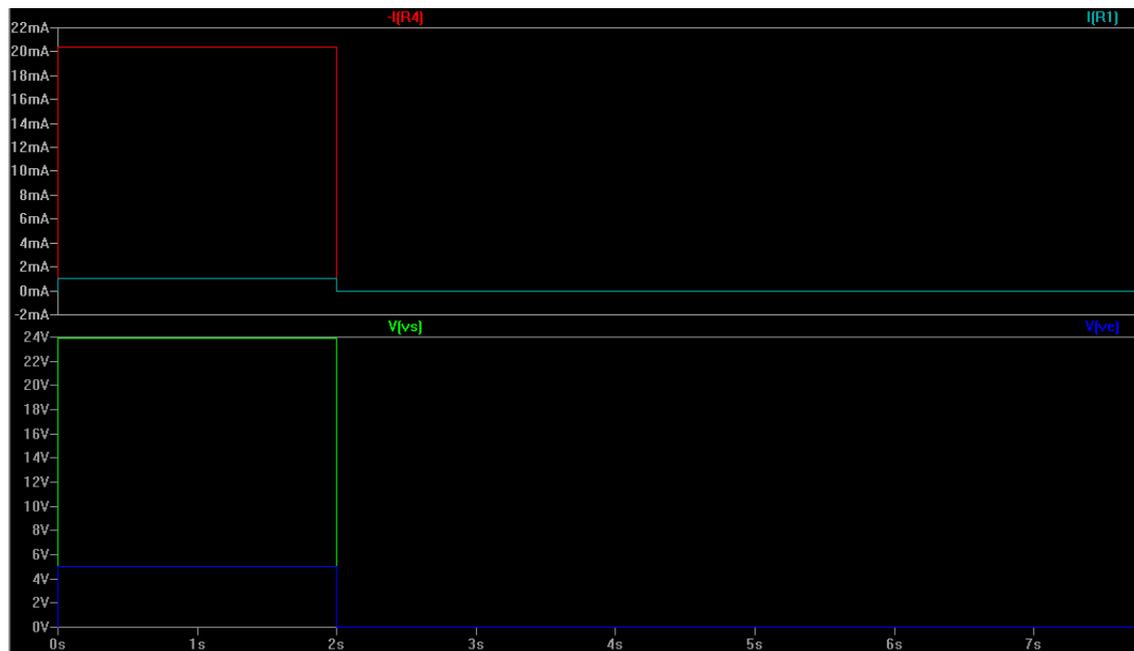


Figure 6 : Résultats de simulation

Le premier graphe donne les allures de courant. La courbe rouge indique la valeur du courant présente à l'entrée de l'optocoupleur. La courbe turquoise est l'allure du courant en sortie de l'optocoupleur.

Le second graphe donne les allures de tension. La courbe en bleu est la tension en sortie du capteur à effet Hall et elle sera directement utilisée par l'Arduino. La courbe en vert est la tension en sortie de l'optocoupleur qui sera utilisée par l'automate programmable.

Le schéma électrique ci-dessous utilise la même solution technologique que le schéma précédent. Nous utilisons deux optocoupleurs. Le premier sert à envoyer l'information si l'utilisateur a appuyé sur le bouton poussoir et le second permet d'allumer le voyant de ce bouton poussoir grâce à l'information fournie par l'automate. Ces deux composants servent toujours pour l'adaptation de tension entre les deux dispositifs.

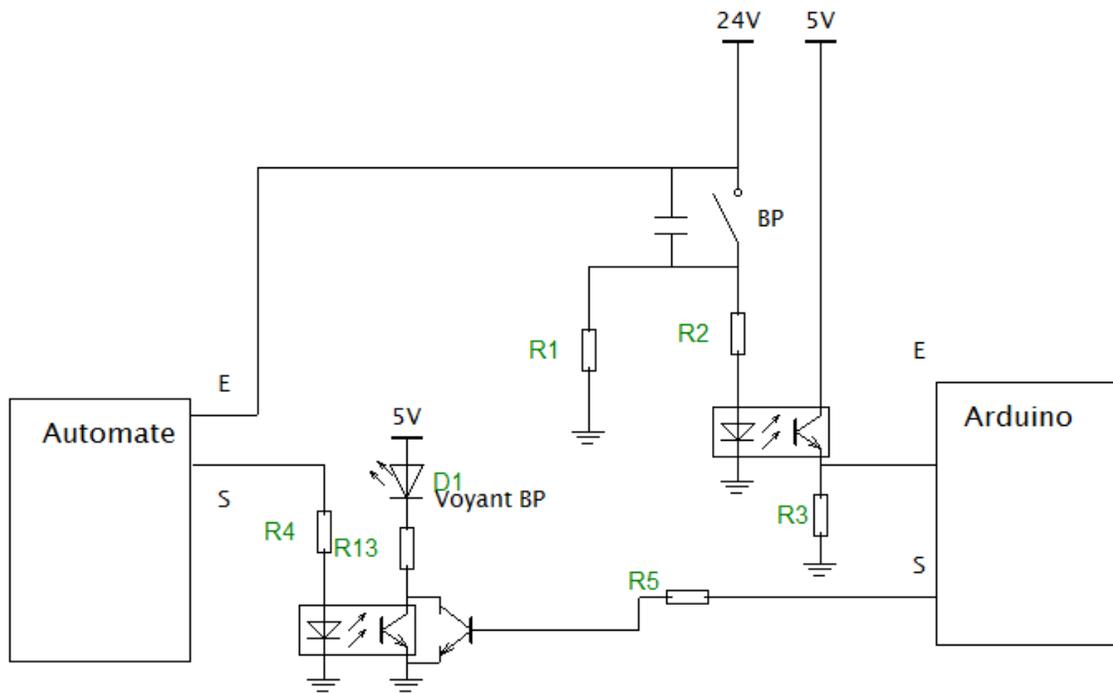


Figure 7 : Schéma pour la commande des boutons poussoirs avec voyant

Calcul des résistances pour la partie bouton-poussoir

Lois des mailles 1:

$V_{cc} = 24V$

V_d = tension aux bornes de la diode

i_2 : courant traversant la résistance R2

$V_{cc} = R_2 \cdot i_2 + V_d$

On fixe $i_2 = 20mA$, on peut donc calculer la valeur de la résistance R2.

$$R_2 = \frac{V_{cc} - V_d}{i_2} = \frac{24 - 1.2}{0.020} = 1.14k\Omega \text{ (valeur choisie pour les tests : } 1.2k\Omega)$$

L'ajout d'un condensateur de 10nF aux bornes du bouton poussoir a été ajouté au schéma initial, car les tests ont montré qu'il y avait des rebonds.

Loi des mailles 2:

$$V_{cc}=5V$$

$$V_{cc}= V_{ce0}+R_3*i_3$$

On fixe le courant i_3 à 1mA

$$R_3 = \frac{V_{cc} - V_{ce0}}{i_3} = 4.9 \text{ k}\Omega \text{ (valeur choisie pour les tests : } 5.6\text{k}\Omega)$$

Calcul des résistances pour la partie voyant lumineux

Loi des mailles 1:

$$V_{\text{automate}}=0- 24V$$

V_d = tension aux bornes de la diode

i_4 : courant traversant la résistance R_4

$$V_{\text{automate}} = R_4*i_4 + V_d$$

On fixe $i_4= 20\text{mA}$, on peut donc calculer la valeur de la résistance R_4 .

$$R_4 = \frac{V_{\text{automate}} - V_d}{i_4} = \frac{24-1.2}{0.020} = 1.14\text{k}\Omega \text{ (valeur choisie pour les tests : } 1.2\text{k}\Omega)$$

Loi des mailles 2:

$$V_{cc}=5V$$

V_{led} :tension aux bornes du voyant lumineux

V_{ce0} :tension aux bornes du transistor NPN et du phototransistor

$$V_{cc}= V_{led}+V_{ce0}+R_{13}*i_{13}$$

On fixe le courant i_{13} à 20mA

$$R_{13} = \frac{V_{cc} - V_{led} - V_{ce0}}{i_{13}} = \frac{5-2.1-0.1}{0.020} = 140\Omega \text{ (valeur choisie pour les tests : } 150\Omega)$$

Loi des mailles 3:

$$V_{\text{arduino}}=0-5V$$

V_{arduino} : tensions fournies par la sortie de l'Arduino

V_{led} :tension aux bornes du voyant lumineux

V_{be0} :tension base émetteur du transistor NPN

$$V_{\text{arduino}}= V_{be}+R_5*i_5$$

On fixe le courant i_5 à 20mA

$$R_5 = \frac{V_{\text{arduino}} - V_{be0}}{i_{13}} = \frac{5-0.7}{0.020} = 215\Omega \text{ (valeur choisie pour les tests : } 220\Omega)$$

5. Réalisation des premiers tests

En attendant la réception des composants commandés, nous avons réalisé les premiers tests, sur plaques d'essais, avec d'autres composants afin de valider les différents montages réalisés précédemment.

Suite aux tests, il apparaît que les différents montages fonctionnent et valident les simulations effectuées précédemment.

En effet, nous avons bien, pour le capteur et le bouton, une alternance d'état 1 (24V ou 5V) et d'état 0 (quelques mV) selon l'entrée choisie (information du capteur ou appui du bouton).

N'ayant pas de capteur à effet hall lors de la première phase de test, l'information est remplacée par une tension 5V. Enfin, selon l'entrée (présence ou absence de tension), le voyant (simulée par une LED dans nos tests) s'allume ou reste éteint.

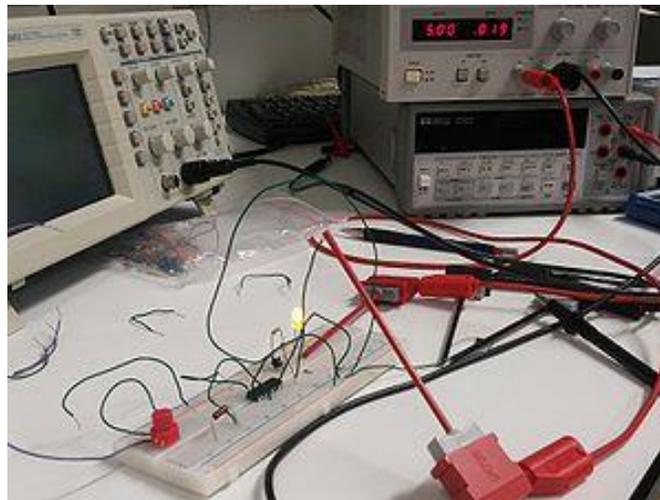


Figure 8 : Montage de test sur plaque d'essai pour le voyant

Dès la réception des capteurs numériques à effet Hall, nous avons réalisé des tests identiques et obtenons bien une variation de tension: 0V lorsque l'on approche un aimant près du dispositif et une tension d'environ 5V si l'aimant est non présent devant le capteur. Ce capteur est alimenté en 5V, la sortie (Output sur le schéma) de celui-ci doit donc être câblée à une résistance dite de "pull up".

Cette résistance est nécessaire pour fixer les deux niveaux de tensions disponibles selon les deux scénarios. On ajoutera ensuite un inverseur logique pour avoir un niveau logique haut (tension de 5V) lorsque l'aimant est devant le capteur.

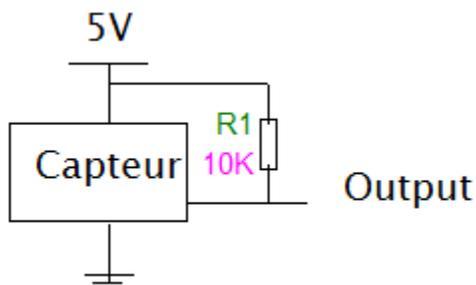


Figure 9 : Schéma du capteur à effet Hall

6. Analyse et étude de la réalisation de la maquette

Nous avons tout d'abord réalisé une première partie de la maquette en carton pour analyser la largeur et la hauteur de la future maquette.

Cette première esquisse nous a permis de réfléchir où nous allons implanter les boutons poussoirs d'appel cabine ainsi que les différents capteurs (à effet Hall pour les capteurs d'étage et de portes et mécaniques pour les capteurs de fin de course).

Nous avons modélisé le début de la maquette finale sur le logiciel de CAO Solidworks.

Problèmes rencontrés

Lors des premiers tests qui ont suivi la réception de notre commande, nous avons été confrontés à un premier problème concernant les capteurs à effet Hall.

En effet, ceux commandés donnent une réponse “numérique” c’est-à-dire que s’il y a présence d’un aimant ou d’un matériel modifiant le champ magnétique, la sortie va être de type tout ou rien (la tension de sortie va alors correspondre à la tension d’alimentation). La finalité est donc ce que l’on souhaitait. C’est concernant le mode de fonctionnement et donc le mode de détection ou non d’un objet que nous n’avons pas obtenu les résultats souhaités.

Les capteurs à effet Hall commandés sont de type bipolaire, il est alors possible de les déclencher selon n’importe quel type de pôles que ce soit un pôle Nord ou Sud. Cependant, le capteur, une fois déclenché, garde l’état et nécessite la présence du pôle opposé pour revenir à l’état initial.

Cet inconvénient pouvait cependant être rectifié au niveau de la programmation de l’Arduino. Malheureusement, concernant la partie automate ce n’était pas réalisable étant donné que l’automate attend à ses entrées, une valeur 1 (état haut).

Finalement, nous avons pu régler assez facilement ce problème, en commandant des échantillons gratuits sur le site Texas Instruments.

Les nouveaux capteurs sont cette fois-ci, unipolaires et donc ne réagissent qu’à un seul pôle. Le capteur ne se déclenche qu’à l’approche d’un aimant et il suffit simplement d’éloigner l’aimant pour revenir à l’état initial.

Nous avons également rencontré des problèmes de rebonds lors de la génération de fronts montants avec le bouton poussoir, il a donc été nécessaire d’ajouter un condensateur de 10nF aux bornes du bouton poussoir pour corriger ce phénomène.

Planning prévisionnel¹ pour la période de Janvier - Février

Après les différentes études des composants et des montages à utiliser, il est nécessaire de passer à la réalisation concrète des maquettes.

Dès le début du projet, nous avons fait le choix de ne pas faire la conception des deux maquettes en parallèle, mais réaliser tout d'abord une première maquette puis une fois celle-là totalement validée et fonctionnelle, faire de même pour la seconde maquette.

Actuellement, les différents montages étant validés et fonctionnels, il nous reste à réaliser les PCB des cartes gérant les entrées sorties de l'automate, le circuit pour l'arrêt d'urgence ainsi que les adaptations des tensions utilisables par l'Arduino et par l'automate programmable.

Concernant la conception matérielle de la maquette, nous avons été plus lents que prévu. En effet, nous voulions commencer la réalisation avant les vacances de fin d'année or nous n'avons pour le moment que commencé à réfléchir aux différents moyens de réalisation. Cette réalisation comprend à la fois, la conception du bâti général de la maquette (avec les 5 étages), mais aussi la conception de la cabine et des différentes portes.

En parallèle, il nous reste également la réalisation de la partie « translation de la cabine » avec la réalisation d'une poulie à fixer que la partie mobile du moteur AX12+. Elle sera réalisée avec l'imprimante 3D du Fabricarium de l'école. Nous avons déjà conçu le patron de cette pièce sous le logiciel Solidworks.

Enfin nous devons réaliser la programmation complète et finale de la maquette sur l'Arduino afin de réaliser le mode « démonstration ».

Il nous reste donc encore du travail à effectuer afin d'obtenir des maquettes fonctionnelles et de répondre aux diverses spécifications du sujet.

¹ Vous trouverez à la page suivante (page 18), le planning prévisionnel ainsi que le planning réel détaillant l'avancement du projet et ses différentes étapes de réalisation.

	Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5	Semaine 6	Semaine 7	Semaine 8	Semaine 9	Semaine 10	Semaine 11	Semaine 12	Semaine 13	Semaine 14	Semaine 15	Semaine 16	Semaine 17	Semaine 18	Semaine 19	Semaine 20	Semaine 21	
Compréhension du sujet et état de l'art	Orange																					
Élaboration du cahier des charges		Orange	Orange																			
Choix et commande des composants			Orange																			
Étude et commande du moteur par arduino				Orange	Orange																	
Gestion fonctionnement Arduino - Automate							Orange	Orange														
Réalisation des premiers tests de validation des composants								Orange														
Réalisation d'un "prototype" pour la pré-soutenance									Orange	Orange												
Réalisation du châssis de la maquette 1											Orange	Orange	Orange	Orange								
Gestion de la partie mécanique (translation de la cabine)																						
Gestion de la partie sécurité (arrêt d'urgence, capteurs fin de course)																Orange	Orange					
Tests et validations (Résolution des problèmes si nécessaire)																		Orange				
Réalisation de la maquette 2																				Orange	Orange	Orange
Réalisation de la vidéo de fin de projet																						Orange

Planning prévisionnel
 Planning réel

Bibliographie

- Mapping de l'Arduino 2560

<https://www.arduino.cc/en/Hacking/PinMapping2560>

- Librairie Dynamixel.h

<http://savageelectronics.blogspot.fr/2011/01/arduino-y-dynamixel-ax-12.html>

- Description de la librairie Dynamixel.h

<http://austinpalmer.com/Projects/Documentr/#/home>

- Datasheet Servomoteur AX12+

<http://www.trossenrobotics.com/images/productdownloads/AX-12%28English%29.pdf>

- Datasheet Capteur effet Hall

<http://www.ti.com/lit/ds/symlink/drv5023.pdf>

- Datasheet Optocoupleur

<http://docs-europe.electrocomponents.com/webdocs/13e4/0900766b813e4118.pdf>