



**INGENIEUR POLYTECH LILLE**

**Informatique Microélectronique Automatique**

**Rapport – Projet de Fin d'Etudes**

# **Conception d'un bras robotisé déformable antagoniste**

**GIOVANNANGELI Florian**



<p><b>Tuteurs école :</b> M. DEQUIDT Jérémie M. VANTROYS Thomas</p>	<p><b>Tuteurs entreprise :</b> M. MORALES BIEZE Thor M. SANZ-LOPEZ Mario</p>
---	--

# Remerciements

Avant de commencer ce rapport, je souhaite remercier plusieurs personnes qui m'ont aidé, accompagné et encouragé pendant ce projet de fin d'études.

Je commence donc par remercier chaleureusement mes encadrants Thor Morales-Bièze et Mario Sanz-Lopez pour m'avoir permis de rejoindre l'équipe DEFROST pour ce projet, pour leur importante disponibilité afin de m'encadrer et m'aider tout au long de celui-ci et pour l'expérience qu'ils m'ont apportée.

Je souhaite également remercier le directeur de l'équipe DEFROST, Christian Duriez, pour m'avoir accepté dans celle-ci, et l'ensemble de ses membres pour leur sympathie et l'aide qu'ils ont pu m'apporter pendant ces cinq mois.

Je remercie aussi l'équipe pédagogique de l'école Polytech Lille qui m'a permis au cours de ces trois années d'acquérir bon nombre de connaissances et de compétences que j'ai pu utiliser et renforcer lors de ce projet, et parmi eux Jérémie Dequidt et Thomas Vantroys qui ont suivi mon travail jusqu'à la fin de ce projet.

# Abstract

Département Informatique, Microélectronique et Automatique, Polytech Lille

## **Conception d'un bras robotisé déformable antagoniste**

par Florian GIOVANNANGELI

Ce rapport rend compte de mes activités lors de mon Projet de Fin d'Etudes dans l'équipe DEFROST de l'Institut National de Recherche en Informatique et Automatique de septembre 2017 à février 2018. Il décrit le contexte dans lequel j'ai effectué ce projet, ainsi que mes réalisations et des perspectives d'évolution. Un prototype de bras robotisé déformable appelé BigMomma sera présenté, ainsi que mon rôle le concernant et les prochaines étapes à réaliser. Chronologiquement, les tâches qui me sont confiées sont de me familiariser avec les concepts opérationnels liés au système de contrôle pneumatique déjà mis en place, d'étudier des pistes d'améliorations de ce système, d'intégrer ces changements dans une nouvelle carte électronique de contrôle, d'implémenter le programme et d'effectuer des tests et améliorations.

**Mots-clefs :** Recherche et Développement – Robotique déformable – BigMomma – Carte de contrôle

This report sums up my end of studies project in the DEFROST team, part of the Inria, French National Institute for computer science and applied mathematics, from September 2017 to February 2018. It describes the context in which I carried out my project, an overview of my achievements and evolution perspectives. A prototype of a soft robotic arm called BigMomma will be presented in further pages as well as my role on that and the next steps to undertake in order to achieve it. Chronologically my different tasks are to familiarize with the operational concepts related to the pneumatic control system, to study ways of improving this system, to integrate these changes on a new electronic control card, to implement the program and to perform tests and improvements.

**Keywords:** Research and Development – Soft robotics – BigMomma – Control circuits

# Sommaire

<b>Introduction .....</b>	<b>5</b>
<b>Partie 1 : Contexte global du projet.....</b>	<b>6</b>
1.1 L'Inria et l'équipe DEFROST .....	6
1.2 Le robot « BigMomma ».....	7
1.3 Le contrôle pneumatique .....	7
<b>Partie 2 : Cahier des charges .....</b>	<b>9</b>
2.1 Objectifs.....	9
2.2 Missions.....	9
2.3 Organisation du travail.....	9
<b>Partie 3 : Présentation du travail effectué .....</b>	<b>10</b>
3.1 Recherches d'améliorations du système de contrôle .....	10
3.1.1 Système actuel .....	10
3.1.2 Recherches et améliorations.....	11
3.2 Réalisation de la carte de contrôle pneumatique .....	13
3.2.1 Schématique et PCB .....	13
3.2.2 Câblage .....	13
3.2.3 Tests.....	14
3.3 Implémentation du contrôle pneumatique.....	16
3.3.1 Problématique .....	16
3.3.2 Réglage du PID.....	17
3.3.3 Recherche de seuils d'activation .....	18
3.3.4 Problèmes rencontrés .....	19
3.4 Contrôle de la pompe à vide .....	21
3.5 Documentation.....	22
<b>Partie 4 : Perspectives d'évolution .....</b>	<b>23</b>
<b>Conclusion .....</b>	<b>24</b>
<b>Annexes.....</b>	<b>25</b>
a. Diagramme de Gantt de l'organisation initiale du travail .....	25
b. Schématique de la carte de contrôle pneumatique.....	26
c. PCB de la carte de contrôle pneumatique .....	27
d. Schématique de la carte de contrôle de la pompe à vide.....	28
e. Photos illustrées des deux cartes réalisées .....	29

# Introduction

Dans le cadre de ma 5<sup>ème</sup> année à l'école Polytech Lille au département IMA (Informatique Microélectronique et Automatique), j'ai l'occasion de réaliser mon Projet de Fin d'Etude à l'Inria Lille Nord Europe. Ce projet, réalisé de septembre 2017 à février 2018, est fait au sein de l'équipe DEFROST et consiste en la conception d'un bras robotisé déformable antagoniste.

En robotique, la majorité des machines sont composées d'éléments mécaniques, avec des structures rigides. Le mouvement s'effectue grâce à un nombre fini d'actionneurs permettant d'agir à un endroit particulier avec un degré de liberté spécifique. Constitués de matériaux souples capables de se déformer pour produire un mouvement, les robots déformables en revanche peuvent posséder un degré de liberté infini ce qui offre un certain nombre d'avantages mais pose également un certain nombre de défis scientifiques. La structure des robots déformables étant très différente de celle de robots traditionnels, les méthodes de contrôles utilisées actuellement pour ces derniers ne peuvent pas être assimilés tels quelles. Certaines équipes de recherche, telle que DEFROST, s'intéressent donc à cette question : comment contrôler le mouvement d'un robot possédant une infinité de degrés de liberté en certains points en n'agissant que sur un nombre fini de paramètres ?

Dans cette optique de recherche, je travaille pour ma part sur un bras robotisé déformable, nommé BigMomma. Ce robot fonctionne sur un système antagoniste utilisant des cavités pneumatiques contrôlées en pression, combinées à des câbles contrôlés par des moteurs. Ce système fournit au robot une bonne rigidité et une grande envergure lui permettant de s'adapter à de nombreux cas de figures. Dans le cadre de mon projet, je me suis intéressé principalement au contrôle pneumatique du robot. Mon objectif global est de mettre en place un système de contrôle du gonflage et dégonflage précis et efficace pour l'intégralité du robot.

Ce rapport fait état de mes avancées sur ce projet. Dans un premier temps, je ferai un rapide aperçu du contexte global du projet, les problématiques et les enjeux de ce secteur de recherche. Dans un second temps, je présenterai le cahier des charges, les missions qui m'ont été confiées, ainsi que l'organisation de mon temps de travail. Par la suite, je détaillerai mes réalisations, les problèmes rencontrés et les solutions mises en place. Pour finir, je présenterai les tâches qu'il reste à réaliser et les perspectives d'améliorations d'ici la fin de mon projet.

## Partie 1 : Contexte global du projet

J'ai eu la chance de réaliser mon stage au sein de l'Inria sur le site de Lille Nord Europe dans l'équipe DEFROST. Avant de décrire les différentes missions qui m'avaient été confiées et les différentes tâches que j'ai réalisées, une petite présentation de l'équipe et du contexte de recherche s'impose.

### 1.1 L'Inria et l'équipe DEFROST

L'Inria (nommé IRIA jusqu'en 1979), ou Institut National de Recherche en Informatique et Automatique, a été créé en 1967 afin de doter la France d'un centre de recherche national dans le domaine de l'informatique et d'avoir une recherche scientifique autonome de haut niveau. Ce centre possède actuellement 8 centres de recherche dans toute la France (Paris, Rennes, Sophia Antipolis, Grenoble, Nancy, Bordeaux, Lille et Saclay) et son siège social est situé à Rocquencourt près de Paris.

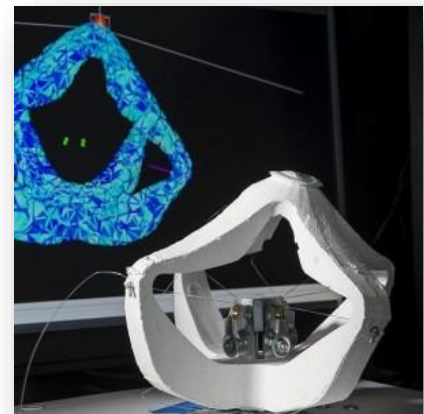
L'équipe DEFROST (DEFormable ROBot SoftWare), au sein de laquelle je travaille, est implantée sur le site de Lille et comporte actuellement une vingtaine de membres composés principalement de chercheurs, doctorants et ingénieurs. Cette équipe s'inscrit dans le domaine Perception, Cognition, Interaction et plus particulièrement dans la thématique Robotique et environnements intelligents. Elle est l'une des quelques équipes dans le monde à travailler sur la question des robots déformables. Plusieurs objectifs de recherche sont fixés au sein de l'équipe. Tout d'abord vient celui de la modélisation de robots déformables. Cela passe à la fois par la création de nouveaux modèles mais également par la simulation et l'amélioration des méthodes de simulation.

Ensuite vient la recherche sur l'interaction avec l'environnement. Cela englobe des problématiques comme la capacité d'attraper des objets, se déplacer dans un environnement fragile en minimisant les contacts avec l'extérieur ou encore l'intégration de contraintes physique et des forces agissant sur le robot comme les forces de frictions. Enfin le troisième volet des recherches est celui du contrôle et des algorithmes utilisés pour cela.

Ces objectifs sont déclinés dans différentes applications pratiques permettant l'expérimentation. D'un côté il y a la participation active au développement du logiciel SOFA qui implémente les méthodes de contrôle et les différents travaux de recherche, permettant entre autre la modélisation et la simulation de robot et de leur environnement. De l'autre le travail autour de différents prototypes permettant l'expérimentation sur des modèles réels mettant en place différents mécanismes de contrôle et ayant différentes spécificités. J'ai pour ma part travaillé sur ce dernier axe.



Logo Inria depuis juillet 2011



Robot déformable Diamond

L'équipe DEFROST possède plusieurs prototypes ayant différents aspects, objectifs, fonctionnements et donc différentes contraintes. Le prototype sur lequel je travaille est celui d'un bras articulé surnommé « BigMomma » (Big Mou Manipulator). Il est inspiré de différents modèles existants mais est plus de dix fois plus grand que ceux-ci, ce qui pose de nouveaux enjeux et de nouvelles problématiques.

## 1.2 Le robot « BigMomma »

Le prototype BigMomma est un robot dont le comportement peut être assimilé à un bras articulé inspiré du fonctionnement d'un tentacule de pieuvre. Il est constitué de plusieurs cavités reliées ensemble. Le mouvement dans ce robot est ainsi contrôlé par différents mécanismes. La longueur du bras est gérée par le contrôle de la pression dans celui-ci : plus la pression est forte plus le bras est long et inversement. Ce contrôle pression agit donc sur une seule dimension. Le déplacement dans les autres directions est contrôlé par un système de câbles permettant de contracter la cavité dans différentes directions permettant un mouvement dans l'espace.

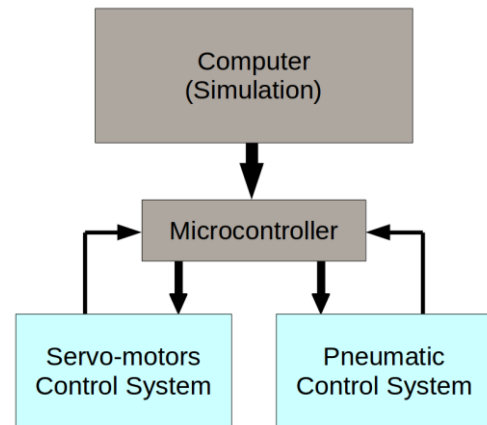


Figure 1 Principe de fonctionnement du robot

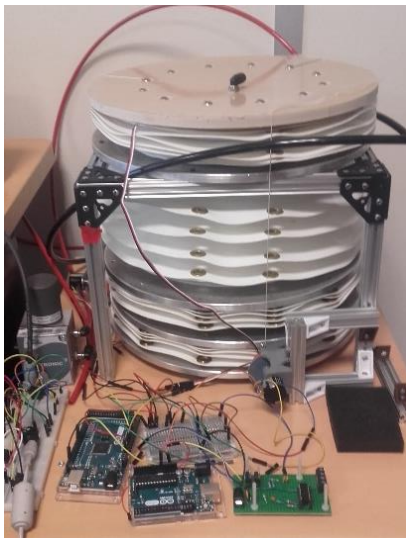


Figure 2 Le bras robotisé BigMomma

La combinaison de ces deux méthodes de contrôle forme ainsi un système dit antagoniste. En effet, elles agissent dans des directions différentes, ce qui fournit au robot une rigidité plus importante. Cette rigidité est l'aspect le plus intéressant de ce robot car cela lui permet de s'adapter à de nombreux cas de figures comme des charges plus ou moins lourdes et des perturbations environnementales tout en gardant une mobilité et une envergure importantes.

Actuellement, le système des câbles n'est pas encore installé sur le robot, car un travail sur simulation est nécessaire afin de déterminer la meilleure implantation des câbles et des moteurs. Mon projet s'intéresse donc principalement au contrôle pneumatique.

## 1.3 Le contrôle pneumatique

Les cavités pneumatiques qui composent le robot sont une sorte d'accordéon qui peut ainsi se replier ou se déplier facilement. Afin d'alimenter les cavités en pression, celles-ci sont reliées à un distributeur (vanne pneumatique), lui-même relié à un compresseur et une pompe à vide. Le contrôle pneumatique du robot se fait donc par le contrôle de cette vanne.

En parallèle de cela, il est nécessaire d'avoir des mesures en temps réel de la pression, afin de pouvoir affiner le contrôle et mettre en place notre propre boucle. Une première option utilisée était de récupérer directement la pression mesurée par la vanne pneumatique en son sein, mais elle s'est révélée peu précise car les mesures étaient sensiblement différentes de la pression réelle dans le robot (tuyaux, pertes etc..). Des capteurs de pressions ont donc été mis en place directement dans les cavités.

Le système de contrôle pneumatique actuellement mis en place est donc le suivant :

Un prototype de carte de contrôle a été réalisé et est globalement fonctionnel, mais n'est pas optimal. En effet, l'intérêt a surtout été porté vers la rapidité de gonflement et dégonflement, mais pose des problèmes de précision. Les éléments qui composent le circuit ont été parfois choisis arbitrairement dans des situations de fonctionnement parfaites.

Un fonctionnement en situation réel, prenant en compte les contraintes extérieures et les autres éléments du robot, révèle des réactions différentes comme des signaux bruités, des dépassements de consigne menant à une alternance d'utilisation des appareils pouvant les endommager etc... De plus, le système de contrôle en question n'est fonctionnel que pour une seule cavité. Or le robot en possède quatre qu'il est nécessaire de contrôler séparément.

Mon travail dans le cadre de ce projet consiste donc à trouver des solutions d'améliorations et de les mettre en place afin d'obtenir un système de contrôle pneumatique complet et fonctionnel sur l'intégralité du robot.

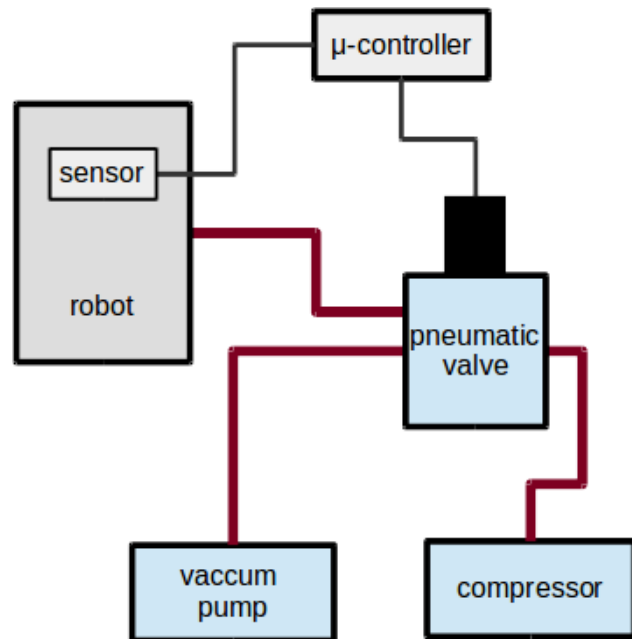


Figure 3 Schéma de fonctionnement du contrôle pneumatique



## Partie 2 : Cahier des charges

Suite à cette présentation du contexte global de mon projet, je vais pouvoir décrire ici les objectifs du projet, les tâches et missions qui en découlent, ainsi que l'organisation de mon travail. Cette partie fait donc office de cahier des charges de mon PFE.

### 2.1 Objectifs

L'objectif global est d'améliorer le modèle actuel du corps du robot, et de mettre en place l'intégralité du système de contrôle pneumatique développé à travers de la conception de cartes PCB en forme des modules. Également, il sera nécessaire de mettre en place un système embarqué de contrôle et pilotage des actionneurs, avec le développement du code pour la communication.

### 2.2 Missions

A partir de l'objectif global du projet, plusieurs missions ont été définies avec les encadrants dans le cadre de mon Projet de Fin d'Etudes :

1. Etudes préalables
  - les robots déformables
  - le robot, son fonctionnement et son but
  - travaux réalisés, état actuel du projet
2. Carte de contrôle pneumatique
  - recherche de solution d'amélioration
  - choix des composants
  - réalisation du schématique et PCB
  - tests et optimisation
3. Programme de contrôle pneumatique
  - recherche de solutions d'amélioration
  - implémentation
  - tests et optimisation

### 2.3 Organisation du travail

Le diagramme de Gantt expliquant l'organisation de mon travail est disponible en Annexes :

(cf. a. Diagramme de Gantt de l'organisation initiale du travail)

Il indique mes prévisions de temps de travail pour chaque partie du cahier des charges, ainsi que le temps réel impliqué.

## Partie 3 : Présentation du travail effectué

### 3.1 Recherches d'améliorations du système de contrôle

#### 3.1.1 Système actuel

Le système de contrôle pneumatique a été défini et réalisé par de précédentes recherches sur le sujet. Néanmoins, des pistes d'amélioration sont encore à trouver et le premier prototype de carte de contrôle a été réalisé à la main. Le premier travail à réaliser est donc d'étudier le système actuel, de dépouiller les données mesurées sur celui-ci et de proposer des améliorations afin d'accroître la performance du contrôle pneumatique.

Le système actuel qui a été défini est celui-ci :

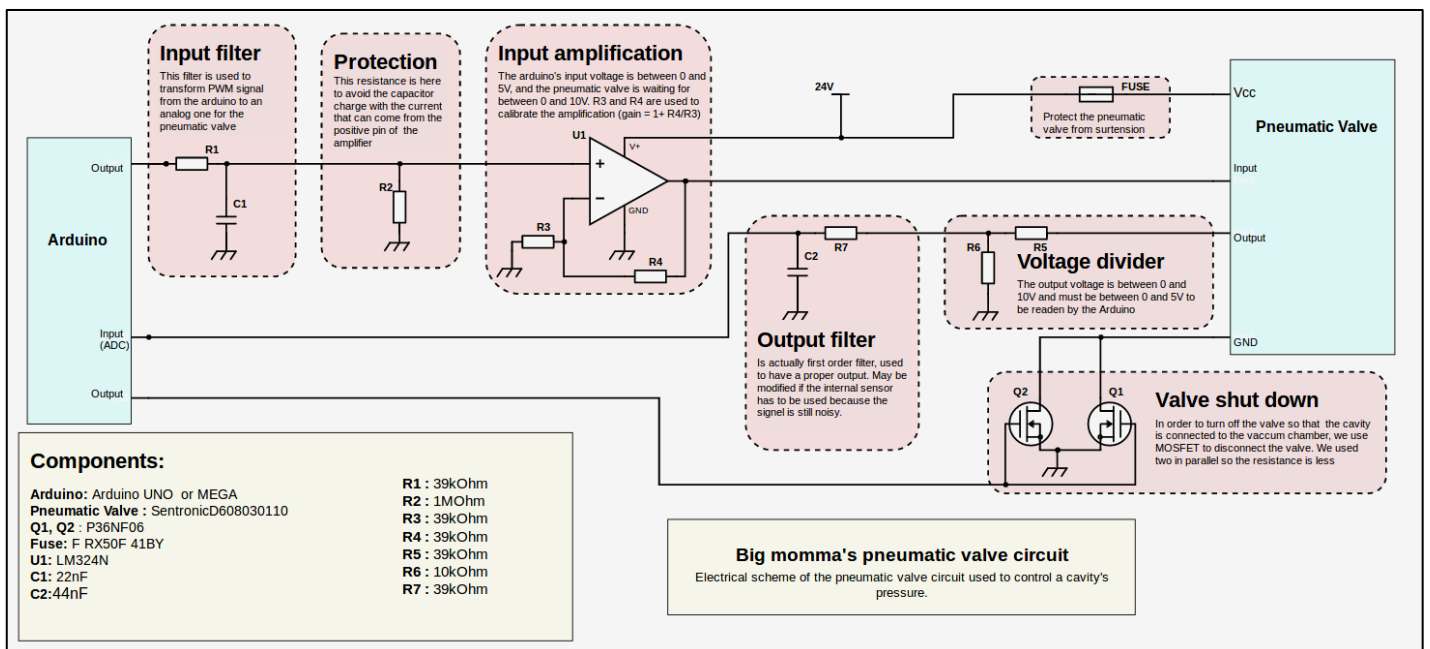


Figure 4 Schéma électrique de la carte de contrôle pneumatique actuelle

La première sortie de l'Arduino correspond au signal PWM de contrôle de la valve. Il est suivi d'un filtre permettant d'avoir un signal analogique, qui est ensuite amplifié à l'aide d'un montage en amplificateur non-inverseur afin d'avoir les bonnes tensions de commande de la valve (0-10V). La valve envoie en retour les valeurs mesurées de pression qu'elle a fournie au robot. Ce signal passe par un diviseur de tension et un filtre de façon à fournir un signal lisible et interprétable par l'Arduino. Enfin, l'Arduino contrôle un système d'interrupteur composé de 2 MOSFET en parallèle pour connecter ou déconnecter la valve pneumatique en fonction des besoins.

Les données mesurées sur ce système montrent qu'il y a néanmoins des possibilités d'amélioration, notamment au niveau de la propreté et la précision des signaux échangés ainsi que de la rapidité d'exécution. C'est dans cette optique que j'ai commencé mes recherches.

### 3.1.2 Recherches et améliorations

#### ❖ Amplificateur opérationnel

Les premières perspectives d'amélioration se sont portées sur l'amplificateur opérationnel. J'ai effectué plusieurs recherches afin de vérifier que celui utilisé (LM324N) était le plus adapté à l'application voulue. Il s'est avéré qu'il était difficile de trouver un meilleur AOP. En effet, le circuit demande tout d'abord une alimentation assez importante (24V) ce qui restreint le choix des AOP qui demandent généralement moins de tension en entrée (~5-10V). L'utilisation d'un régulateur de tension combiné à un autre AOP pourrait être envisagée, mais dans notre cas ce n'était pas pertinent. Un aspect important de l'amplificateur est sa tension de sortie à l'état bas. L'objectif est d'avoir la valeur la plus proche de 0 possible pour envoyer les commandes à la valve (celle-ci attend du 0-10V). Dans notre cas, le composant utilisé est de type rail-to-rail et cette valeur est de 5 à 10mV ce qui est suffisamment proche. Enfin, il est nécessaire d'avoir 4 AOP pour contrôler, à terme, les 4 cavités du robot. Les recherches effectuées n'ont donc pas permis de trouver un meilleur composant pour nos attentes, ou en tous cas pas suffisamment pour envisager un changement.

#### ❖ Filtres et programmes de tests

D'autres perspectives d'amélioration ont ensuite été étudiées sur les filtres d'entrée et de sortie. J'ai cherché à trouver la configuration optimale afin d'avoir des signaux plus propres et donc plus de précision. Afin d'avoir une idée du meilleur filtre à utiliser pour le circuit, j'ai dû me lancer dans la réalisation d'un code de test pour simuler les signaux de commande PWM et de retour de la valve pneumatique. Ces signaux seront générés par un Arduino Uno et on pourra donc, à l'aide d'un oscilloscope, visualiser le contenu harmonique et déterminer le filtre le mieux adapté.

Le programme ainsi réalisé crée un signal PWM qui effectue un balayage sur une certaine gamme de fréquence. On utilise pour cela le Timer1 de l'Arduino UNO qui permet plusieurs modes de PWM (Fast, Phase Correct, Phase and Frequency Correct...). Dans un premier temps nous utilisons le Phase Correct PWM Mode qui est suffisant pour notre application. Initialement, le balayage se fait entre 20 et 30kHz, mais l'utilisateur peut utiliser la console de l'IDE Arduino pour modifier la valeur des registres OCR1A et OCR1B de façon à modifier la gamme de fréquence du balayage et ainsi observer la réaction du filtre.

Le programme de tests a fonctionné correctement, en revanche il a été impossible d'exploiter le résultat et la réaction du filtre car la carte avait des défauts. En effet, il s'est avéré qu'il y avait des faux contacts et même certaines pistes involontairement reliées, ce qui risquait d'endommager certains composants. Ces défauts viendraient de l'âge du circuit ou des dernières modifications de la carte (réalisées par une précédente stagiaire) qui n'avaient pas été correctement réalisées ni testées. J'ai tenté de corriger ces défauts en

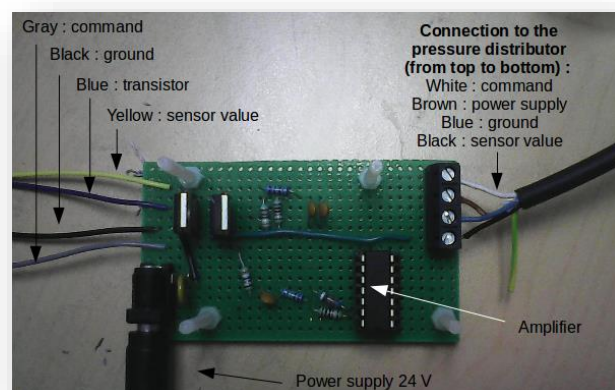


Figure 5 Prototype actuel de la carte de contrôle

refaisant certaines soudures, mais il y avait toujours des problèmes de faux contacts, notamment au niveau de l'amplificateur opérationnel (probablement déjà endommagé). Nous avons donc abandonné l'idée des tests par simulation de signaux.

A l'aide des données que j'ai pu retrouver dans les archives et les spécifications des composants, j'ai néanmoins changé les valeurs pour le filtre de sortie de façon à mettre une résistance R7 plus importante que celles du régulateur ( $R7 \gg R5/R6$  cf. Figure 4), et fixer le filtre à une fréquence de coupure  $f_0 \sim 100\text{Hz}$  pour améliorer le filtrage. Cette configuration permettra également, à termes, de by-passer le régulateur dans le cas de l'utilisation d'un capteur pour la valeur de retour, comme expliqué plus bas. Nous avons envisagé l'idée d'utiliser un filtre du 2<sup>nd</sup> ordre de type Sallen & Key pour améliorer encore le signal de retour, mais la carte étant déjà très chargée, nous avons décidé de rester sur un filtre du 1<sup>er</sup> ordre pour l'instant et tenter éventuellement un 2<sup>nd</sup> ordre plus tard.

### ❖ Capteurs de pression

Actuellement, comme le montre le schéma présenté précédemment, la valeur de retour utilisée par la carte de contrôle est celle fournie par la valve pneumatique. Or cette valeur est très bruitée et peu représentative de la pression réelle à l'intérieur des cavités car elle est mesurée par la valve, mais il y a des pertes et des pressions sensiblement différentes à cause des tuyaux et autres imperfections. Une autre approche qui a été envisagée est d'utiliser un capteur installé à l'intérieur de chaque valve et qui permettrait donc d'obtenir une valeur plus proche de la réalité. Des premiers essais ont été réalisés avec un capteur de pression absolue étant le capteur le plus facile à installer. Cependant ce capteur pose quelques problèmes au niveau de l'offset et de la précision. En effet, dans le cas où l'on arrête et redémarre le robot, la pression dans les cavités peut changer et il est donc difficile de régler un offset universel, une référence, qui prenne en compte ces changements, sans risquer d'endommager le robot (sur-pression, sous-pression...). L'environnement pose également problème (pression extérieur, poids futurs que le robot devra porter...).

Nous avons donc envisagé l'utilisation des capteurs de pression relative (différentielle) qui peut utiliser la pression atmosphérique comme référence et sa pression interne pour se calibrer au démarrage. Ce type de capteur est donc plus fiable, bien que plus difficile à installer car il faudra utiliser 2 tuyaux traversant les cavités, ce qui peut engendrer des pertes. L'installation de ceux-ci ne sera donc probablement pas possible d'ici la fin de mon projet, mais leur utilisation future a été prévue et intégrée à la carte de contrôle.

### ❖ Transistors MOSFET

L'utilisation des deux transistors MOSFET en parallèle est une solution qui avait été décidée précédemment. Cette solution permet de couper la valve pneumatique afin de vider plus rapidement les cavités. Pour cela, l'Arduino envoie aux transistors un signal qui va les rendre soit saturés, ce qui aura pour effet de mettre la masse de la valve à la masse du circuit pour la faire fonctionner ; soit coupés ce qui privera la valve d'une masse et l'éteindra donc. Cette configuration nécessite que la tension de sortie à l'état bas soit la plus proche de la masse possible. Pour ce faire, il faut que la

résistance  $R_{ds(on)}$  des transistors soit la plus faible possible. C'est pour cette raison qu'il a été décidé auparavant d'utiliser 2 transistors en parallèle pour réduire encore cette résistance.

Suite à plusieurs recherches, nous avons pu trouver un nouveau transistor, le FDP8880, qui possède une  $R_{ds(on)}$  très faible ( $0,0116\Omega$ ), suffisamment pour n'avoir à utiliser qu'un seul transistor. Nous avons cependant laissé dans le circuit la possibilité de mettre 2 transistors parallèles dans le cas d'une autre utilisation.

## 3.2 Réalisation de la carte de contrôle pneumatique

### 3.2.1 Schématique et PCB

Comme cela a été défini dans le cahier des charges, l'objectif de cette nouvelle carte de contrôle est d'être la plus polyvalente possible pour que les applications futures puissent être testées sans qu'il soit nécessaire de refaire une carte, ou du moins avec peu de modifications. La carte réalisée initialement ne permet de contrôler qu'une seule cavité du robot. Or, à terme, le système de contrôle devra être capable de contrôler les 4 cavités. Il faut donc prendre cet aspect en compte pour la nouvelle carte et le schéma doit être répété 4 fois pour contrôler les 4 valves pneumatiques.

Nous avons également ajouté 2 rangés de 4 commutateurs. La première rangée permettra de by-passer les résistances R3 (cf. Figure 4) des amplificateurs du circuit dans le cas où nous voudrions utiliser un gain de 1 au lieu de 2 (dans le cas d'un éventuel changement de valve pneumatique, d'AOP, ou d'alimentation). La deuxième rangée de commutateurs permettra de by-passer le diviseur de tension ( $R5//R6$ ) dans le cas de l'utilisation d'un capteur de pression interne comme valeur de retour. En effet, comme expliqué précédemment, la valeur de retour est actuellement transmise par la valve pneumatique, mais une autre option envisagée est d'utiliser des capteurs de pression dans les cavités afin d'avoir plus de précision.

Annexe : b. Schématique de la carte de contrôle pneumatique

Annexe : c. PCB de la carte de contrôle pneumatique

### 3.2.2 Câblage

Pour commencer le câblage, il a fallu attendre la réception des composants commandés par l'Inria ainsi que la carte imprimée à Polytech. Cette attente a été plutôt longue en raison des démarches administratives à effectuer pour commander du matériel, ce qui a retardé la demande et la réception de mes composants.

Une fois la carte imprimée et les premiers composants reçus, j'ai pu d'abord constater une erreur au niveau de l'impression : les trous pour les pins du connecteur DC 24V n'avaient pas été faits et ceux des connecteurs de l'AOP étaient trop petits (problème d'empreinte sur Kicad). J'ai donc dû utiliser la Dremel du labo de l'Inria pour effectuer minutieusement les trous à la main. Le câblage a ensuite pu être fait sans encombre. Il manquait dans un premier temps quelques composants qui étaient sur une autre commande comme par exemple les switches. J'ai donc dû câbler un fil pour compenser cela et faire des tests en attendant la réception.

Suite au câblage et aux premiers tests, deux problèmes ont été révélés. Le premier est que le signal de contrôle des valves s'est trouvé être plus bruité qu'attendu. Après plusieurs études du circuit, nous avons compris que le problème était dû à certaines pistes un peu longues qui peuvent produire un effet antenne, notamment au niveau des résistances R3 de l'AOP qui, lorsqu'elles ne sont pas câblées (switchs ouverts), laissent place à de longues pistes non utilisées. Ces longues pistes inutilisées et entourées de la masse, peuvent produire alors un effet de capacité parasite. Pour pallier à ce problème, nous avons décidé de rajouter un condensateur aux bornes de la résistance R4 (boucle de l'AOP) de façon à réaliser un nouveau filtre qui ne pourra qu'être bénéfique quelle que soit l'application.

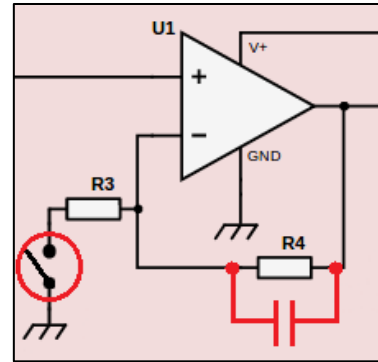


Figure 8 Problème et solution des parasites

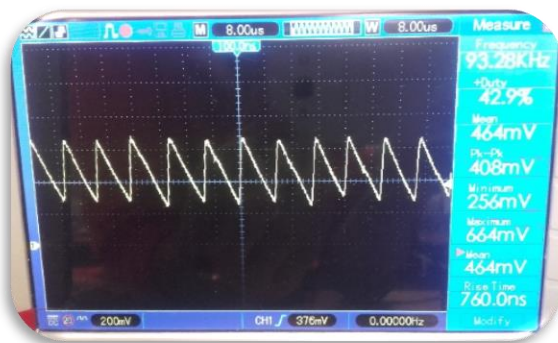


Figure 7 Signal de commande bruité

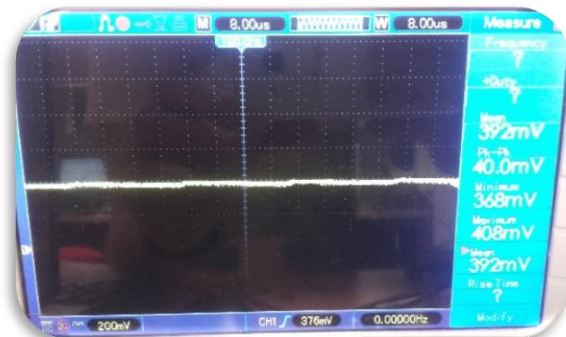


Figure 6 Signal de commande corrigé

Le deuxième problème plus important est une erreur d'empreinte des transistors MOSFET. Alors que je constatais des tensions anormales à leurs bornes, j'ai vérifié le circuit et il s'est avéré que l'empreinte fournie par Kicad et qui était censée être universelle (package TO-220) était en vérité fautive, ou du moins inadaptée pour mon application. L'effet de cette erreur est que les pins de la base et du collecteur se retrouvent échangés et le fonctionnement des MOSFET n'est donc pas celui souhaité. Pour corriger cela, j'ai dans un premier temps échangé manuellement les pins afin d'effectuer des tests, mais j'ai donc dû par la suite réimprimer une carte.

### 3.2.3 Tests

Suite aux modifications manuelles expliquées ci-dessus, et en attendant l'impression de la nouvelle carte corrigée, j'ai pu effectuer quelques tests. J'ai utilisé pour cela une petite cavité de test réalisée en silicone et un programme simple permettant un gonflage et dégonflage progressif de la cavité en contrôlant uniquement le compresseur et la valve pneumatique avec un signal PWM changeant. J'ai également connecté un capteur de pression tel que celui utilisé sur BigMomma pour tester la récupération de données. Après quelques corrections les tests ont bien fonctionné. Cependant, à cause du problème des MOSFETs, je n'ai pu tester qu'un seul circuit de contrôle. Nous avons également remarqué quelques soucis de faux contacts qui se sont créés à force d'utilisation notamment sur certains condensateurs. Ces problèmes devront être évités pour la prochaine carte.

Après avoir reçu et câblé la nouvelle carte contenant les modifications décidées précédemment, j'ai pu à nouveau effectuer ces mêmes tests. La carte est désormais fonctionnelle sur la cavité de test et les différentes erreurs ont été corrigées. J'ai donc ensuite pu passer sur le robot réel.

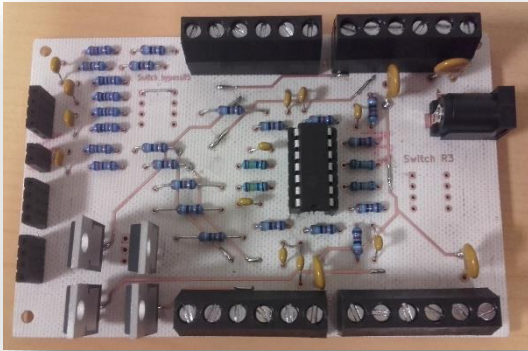


Figure 10 Carte de contrôle pneumatique

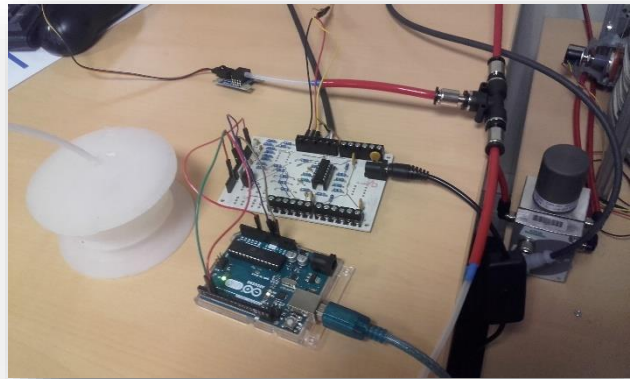


Figure 9 Essais avec la cavité de test

Pour que les tests sur le robot réel se passent dans les meilleures conditions, j'ai dans un premier temps dû me lancer dans le calibrage des différents éléments de contrôle (capteurs, PID, potentiomètres...). J'ai tout d'abord réalisé le calibrage du capteur Phidget 1140 en utilisant un autre et en faisant des tests hors robot. En modifiant le programme et en effectuant plusieurs mesures, j'ai pu arriver à un calibrage proche de la valeur réelle.

Cependant, dès la tentative de passage sur le robot, j'ai été confronté à un problème. Les données envoyées par le capteur interne n'étaient pas celles attendues. Nous avons donc dû ouvrir le robot pour vérifier et il s'est avéré que le capteur installé n'était pas celui annoncé. En effet, le Phidget avait déjà été remplacé par le capteur relatif MPX4250 et mes encadrants n'en avaient pas été informés. D'un certain point de vue, cette modification est bénéfique car elle devait de toute façon être faite à terme, l'utilisation d'un capteur relatif étant dans notre cas plus efficace que celle d'un capteur absolu. Mais le fait de ne pas être courant avant a posé 2 problèmes : les connexions entre les deux capteurs étant différentes, nous avons dégradé inconsciemment celui installé et avons dû le changer; et le calibrage fait au préalable ne servait donc plus et j'ai dû la recommencer avec le nouveau capteur.

Après l'installation et le calibrage du nouveau capteur relatif, j'ai pu procéder aux tests. Les 2 cartes (contrôle pneumatique et pompe à vide) se sont révélées toujours fonctionnelles sur le robot réel, avec des données envoyées et collectées satisfaisantes (valeurs attendues, peu de bruit etc...). J'ai donc pu me lancer sur le réglage du PID et la recherche de seuils.

## 3.3 Implémentation du contrôle pneumatique

### 3.3.1 Problématique

Le programme de contrôle informatique, qui a été réalisé précédemment, est très complet et fonctionnel pour une cavité. La compréhension de celui-ci m'a pris un certain temps. La documentation a cependant bien été réalisée, j'ai donc pu finir m'y retrouver dans les fonctions, et le raisonnement global de la personne l'ayant codé avant moi. Après donc avoir analysé tous les fichiers pour bien prendre en main la partie informatique, et en discutant avec mes encadrants, nous avons pu cibler un problème dans le contrôle de la pression dans la cavité.

Actuellement, le gonflage se fait par l'ouverture de la vanne reliée au compresseur et cette ouverture est gérée par une boucle de contrôle contenant un PID. Cette méthode a permis de grandement accélérer le gonflage et ainsi avoir une réponse rapide. A l'inverse, le dégonflage se fait en ouvrant la vanne reliée à la pompe à vide, une méthode qui permet également d'améliorer la vitesse de réponse. Si en théorie ce système devrait fonctionner normalement, en pratique il existe un délai entre l'émission de la consigne et l'ouverture ou la fermeture de la vanne. Cela a pour effet de créer des dépassements de consignes et donc des alternances d'utilisation entre le compresseur et la pompe à vide. Autre problème, dans le cas d'une nouvelle consigne proche de la précédente, le robot utilisera tout de même la pompe ou le compresseur pour un changement dérisoire. En plus de rendre la stabilisation à la consigne plus longue, cet effet a de fort risque d'endommager, à termes, les équipements.

Une solution envisagée est de définir des seuils autour de la valeur demandée qui permettront d'atténuer les ondulations. Il faudra dans un premier temps modifier les valeurs du PID pour que le gonflage reste rapide mais qu'il se stabilise en évitant de trop dépasser la consigne. Lors du dégonflage, on peut couper la pompe à vide et ouvrir les vannes menant à l'extérieur pour que l'air se vide naturellement et donc plus lentement. Cette solution permettra également de palier le problème de commande très peu changeante ne nécessitant pas l'utilisation des pompes, ainsi que le problème des interactions extérieures qui peuvent fausser les valeurs (ex: une personne qui appuie sur le robot, faisant changer la pression).

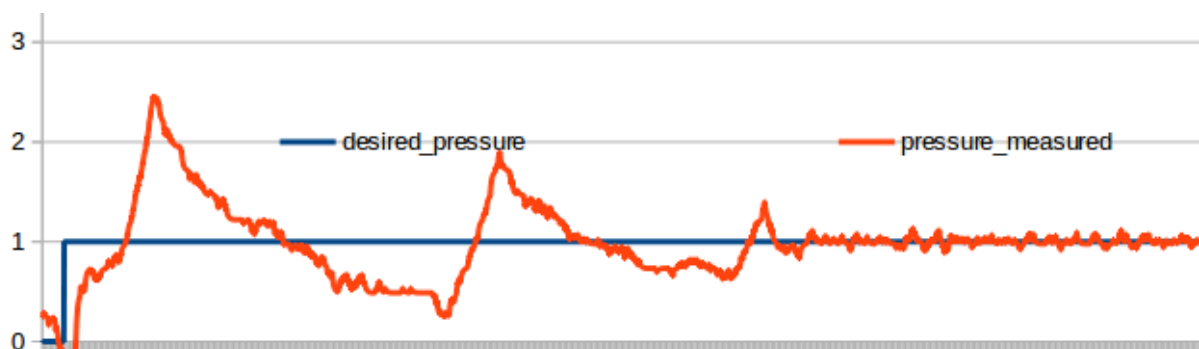


Figure 11 Premier test de la réponse en pression du robot (kPa)



### 3.3.2 Réglage du PID

La première étape pour améliorer le contrôle était de redéfinir les valeurs du PID avec la nouvelle carte et les nouvelles contraintes. En effet, le réglage du PID avait déjà été entamé mais dépend cependant beaucoup de la carte de contrôle et des capteurs de pression car la qualité des signaux de retour contenant les valeurs de pressions mesurées peut varier d'une carte ou d'un capteur à l'autre. Il était donc nécessaire d'effectuer de nouveaux tests en situation réelle avec les nouvelles cartes de contrôle.

J'ai donc réalisé les premiers tests sur le robot complet directement avec le programme qui avait été fait par les précédentes recherches afin de vérifier son fonctionnement et mieux assimiler l'état actuel du contrôle. Le contrôle était fonctionnel mais il y avait des différences avec les données précédemment collectées (cf. Figure 11 Premier test de la réponse en pression du robot (kPa) ). Dans un premier temps, le gain de l'AOP a été modifié sur ma carte (passé de 2 à 1) ce qui a induit, comme attendu, de forts dépassements de la réponse. Il est donc nécessaire pour cela de recalibrer le PID. De plus, un prototype de fonction dans le programme permettant de désigner des seuils d'activation avait été entamé, mais il ne correspondait pas à nos attentes. Nous avons donc décidé de le supprimer et de reprendre cette fonction plus tard, une fois le PID réglé et les seuils définis.

Pour régler le PID, j'ai commencé par mettre en place une action proportionnelle seule et de déterminer le gain maximum qui induit un dépassement. En procédant à plusieurs essais, il s'est cependant avéré que, même avec un très gros gain, la réponse avec un P seul ne produisait pas de dépassement et que le temps de montée ne s'améliorait plus à partir d'un certain gain. Cela est dû au temps de réponse global du système qui est très long, ainsi qu'aux capacités limitées du compresseur et de la valve. La réponse est globalement déjà très satisfaisante. Mais on observe cependant une petite erreur statique et des comportements parfois un peu brusques, qui devraient être corrigés par l'action intégrale.

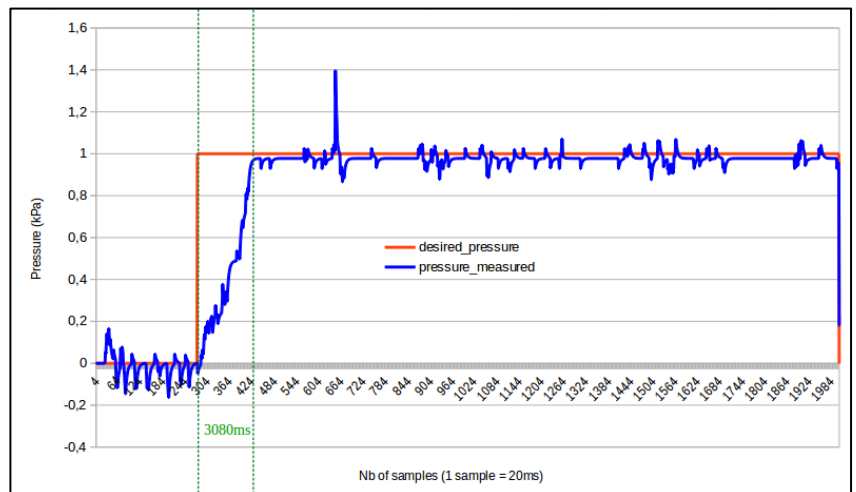


Figure 12 Réponse en pression (kPa) avec un proportionnel P

J'ai ensuite procédé à de nouveaux tests avec un contrôleur PI. Le temps de réponse du système étant long et l'erreur assez faible, il est nécessaire de mettre un coefficient  $K_i$  très petit. J'ai réalisé plusieurs essais qui ont permis d'obtenir un régime stationnaire stable et précis. En revanche on constate que l'action intégrale induit un petit dépassement qui ne disparaît pas, même en réduisant grandement (voire presque totalement) le gain  $K_p$ . Ce gain est négligeable étant donné les faibles pressions qu'il représente (de l'ordre de 0,2 kPa = 0,002 bar) mais il doit être surveillé car à terme il pourrait avoir un effet multiplié sur le robot complet avec les 4 cavités. L'action dérivative devrait théoriquement réduire également ce dépassement ;

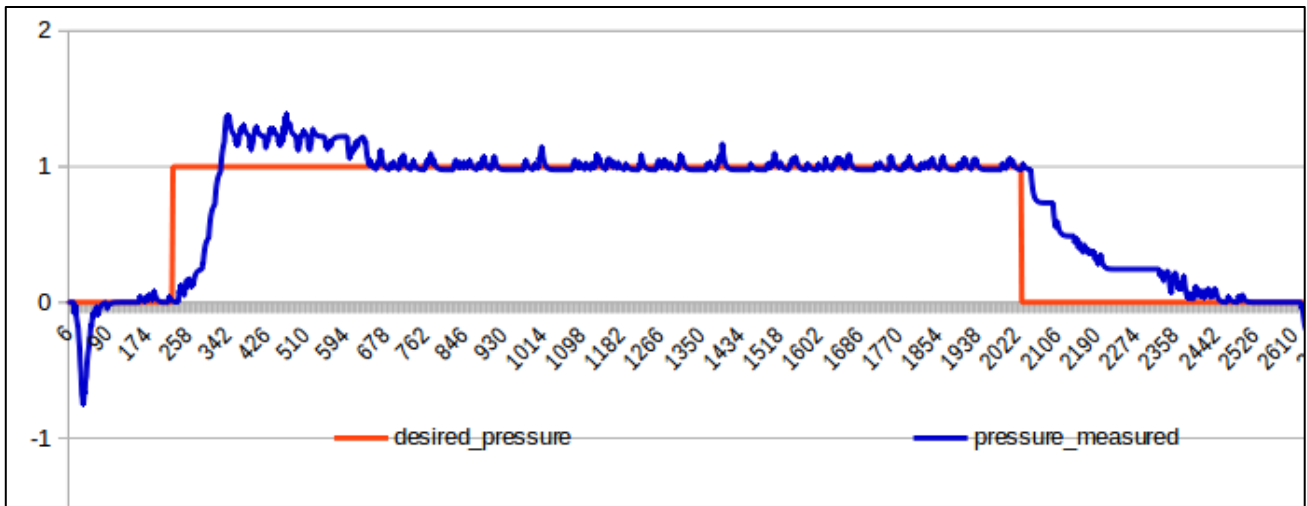


Figure 13 Réponse en pression (kPa) avec un PI

L'action dérivative n'a été que très peu étudiée car j'arrivais au terme de mon projet de fin d'étude. De plus, j'ai été confronté à plusieurs résultats inattendus qui ont nécessité du temps pour les étudier et en déterminer la cause. Ces problèmes sont expliqués ci-dessous. J'ai tout de même effectué plusieurs essais mais je n'ai pas été en mesure de déterminer les coefficients optimaux pour le régulateur PID.

### 3.3.3 Recherche de seuils d'activation

Comme il est expliqué plus haut, une solution proposée pour éviter d'alterner trop souvent entre la pompe et le compresseur alors que ce n'est pas nécessaire, était de définir des seuils autour de la commande. Je me suis donc lancé dans cette recherche.

Un premier seuil a été fixé arbitrairement pour éteindre la valve (ce qui la met en échappement total) à l'aide des MOSFETs lorsque la pression est de 0,2 kPa au-dessus de la commande ( $\text{measured\_value} - \text{desired\_value} > 0.2$ ). Cela fournit deux avantages. Dans un premier temps, cela constitue une sécurité en cas de dépassement imprévu de la commande (par exemple un problème de régulation ou une intervention extérieure sur le robot) permettant d'éviter un déchirement du robot lors d'un gros gonflage. Dans cette situation, la valve se coupe, il ne peut donc plus y avoir de pression supplémentaire et l'air peut s'échapper. Cette incident a d'ailleurs failli arriver avant que nous ne mettions en place ces seuils d'arrêt, comme il est expliqué plus bas dans les problèmes rencontrés.

Dans un second temps, cela permet théoriquement de dégonfler plus rapidement le robot si une nouvelle commande demande une pression bien inférieure à la précédente car la valve éteinte est à priori plus ouverte à l'échappement que lors d'une commande à 0. Ce second aspect est cependant actuellement théorique. J'ai en effet réalisé plusieurs tests et il s'avère que le fait de couper la valve ne change pas (ou très peu) la vitesse de dégonflage. Cela est probablement dû à la capacité de la pompe à vide qui ne peut pas aspirer plus d'air, malgré la plus grande ouverture de la valve. Il sera nécessaire de réaliser de nouveaux tests avec une pompe à vide plus performante pour valider cette théorie.

Une autre solution a été évoquée pour corriger l'actionnement alterné de la pompe à vide et du compresseur, mais elle nécessite l'installation d'un nouvel appareil : une électrovanne. Cette vanne serait placée du côté de l'échappement, entre la valve pneumatique et la pompe à vide, permettant d'utiliser soit la pompe soit un tuyau allant directement à l'air libre. Ainsi, dans le cas de d'un besoin important de dégonflage, on utilise la pompe à vide, mais lors de la stabilisation à la consigne on utilise l'air libre de façon à éviter les dégonflements trop brutaux. Cette installation ne fait pas partie des priorités, mais elle sera probablement faite à terme.

### 3.3.4 Problèmes rencontrés

Lors de mes recherches pour le réglage du PID, j'ai été confronté à plusieurs problèmes qui ont rendu la tâche plus difficile que prévue.

Le premier est un souci de programmation qui aurait pu être très dangereux pour le robot. J'ai souhaité réaliser un test de dégonflage de la cavité en débranchant la pompe à vide, de façon à observer la réaction. J'ai constaté qu'à un certain moment du dégonflage la commande passe de 0 (pour que la vanne soit à l'échappement) à une valeur extrêmement grande, sans raison apparente. Cela a pour effet de regonfler le robot sans limite, pouvant mener à son éclatement si on ne coupe pas la valve à temps (ou si la sécurité des MOSFET est inactive, comme c'était le cas à ce moment). Après différents tests j'ai dans un premier temps constaté que la durée y était pour quelque chose, ce qui expliquait que cet incident ne s'était pas produit auparavant grâce à la pompe qui accélérât le processus. Malgré mes recherches personnelles, ce n'est qu'avec l'aide de mon tuteur Mario que nous avons pu découvrir la raison difficilement décelable. Lors du dégonflage, la valeur de la commande descend dans les négatifs puisque la pression dans la cavité est supérieure à celle désirée. Or la valve ne prenant pas en compte les valeurs négatives, le programme réalisé par la précédente chercheuse n'affichait pas ces valeurs et les ramenait à 0, raison pour laquelle je n'ai pas situé le problème. Le temps de dégonflage étant plus long sans la pompe et le PID faisant son travail, la commande diminue rapidement dans les négatifs et passe finalement en débordement ce qui change la valeur en une autre très grande. Pour contrer ce problème, nous avons modifié le type de la valeur de commande d'un entier à un entier long. Cette solution ne corrige pas le passage dans les négatifs, mais elle rallonge considérablement le temps d'arrivée au dépassement (plusieurs centaines d'heures), ce qui évitera cet incident car il ne sera à priori jamais atteint compte tenu de la taille du robot, de son temps de marche et de l'utilisation de la pompe en cas d'écart important avec la consigne.

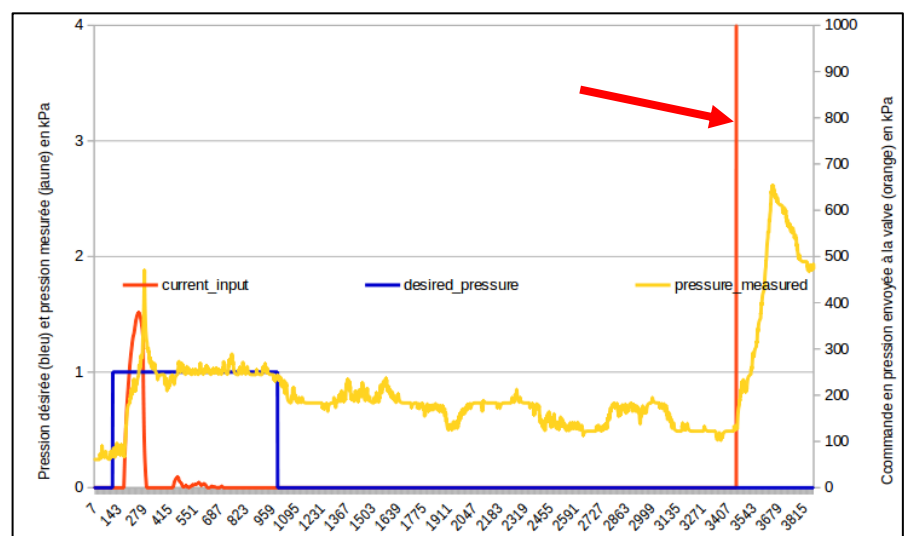


Figure 14 Débordement de la variable « current\_input »

Un autre problème rencontré est celui de faux-contacts ou de bruit au niveau des branchements du capteur. Lors de la réalisation de la carte, nous avons décidé d'utiliser des borniers à vis pour connecter la valve pneumatique et les capteurs, permettant ainsi d'utiliser tout type de capteur sur la carte. Mais le contact des borniers avec les fils fins est moins efficace qu'avec des connecteurs mâle/femelle de 2,54mm. J'ai donc constaté qu'il y avait des faux-contacts et que lorsque l'on bougeait ou touchait la carte, la valeur reçue par le capteur pouvait changer. Cela nécessite, dans un premier temps, une calibration au début de chaque test pour vérifier que la valeur reçue est la bonne. Et dans un second temps, cela cause parfois des pics, des valeurs insensées ou des changements d'offset lors des tests. Car lorsque la cavité se gonfle, elle tire avec elle les câbles de connexion du capteur, ce qui peut induire des faux-contacts. Actuellement, les seules solutions que je vois pour palier à ce problème seraient de modifier le type de connexion fil-à-carte, ou de stabiliser le câble du capteur interne (passable par l'intérieur du robot et non l'extérieur), ou d'implémenter le programme pour qu'il réalise régulièrement un calibrage et une correction de la valeur renvoyée.

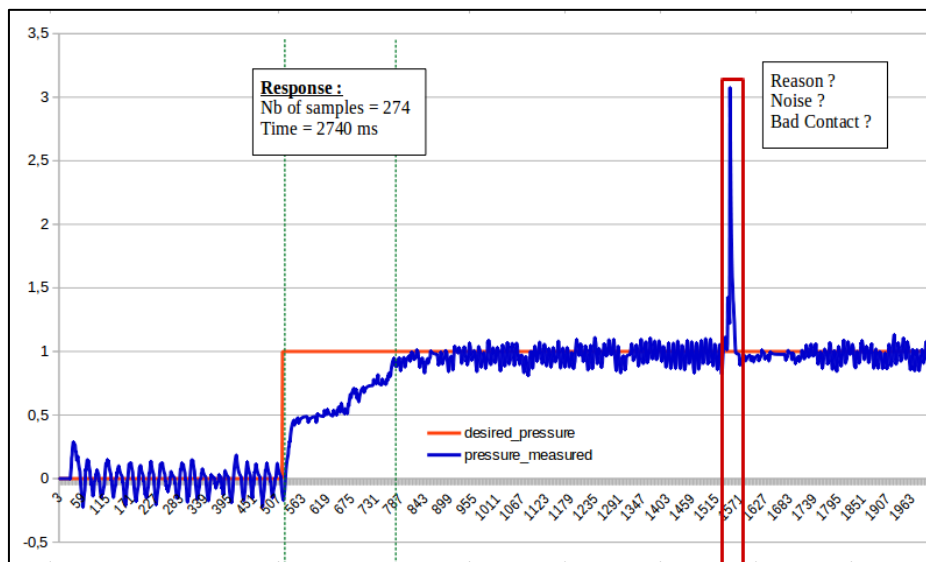


Figure 15 Pic de pression anormal

Enfin, les derniers problèmes sont liés à la structure même du robot. Celui-ci n'est pas encore totalement hermétique et compte actuellement de nombreuses fuites. Cela a pour effet de rendre difficile le réglage du PID car même dans le régime stationnaire, la pression aura tendance à diminuer ce qui implique des variations de gonflage/dégonflage incessantes. Pour que le contrôle soit optimal, il sera donc nécessaire de refaire des tests et régler le PID lorsque le robot sera terminé (préparation des autres cavités, installation de tuyaux, fermeture bien réalisée...) et donc le plus hermétique possible.

### 3.4 Contrôle de la pompe à vide

Durant l'attente de l'arrivée des composants et des circuits imprimés, j'ai pu me lancer sur une autre tâche qui ne faisait pas réellement partie de mon cahier des charges, mais qui devait à l'avenir être réalisée sur le robot. Cette tâche concerne le contrôle de la pompe à vide.

La pompe à vide installée actuellement sur le robot a été « fabriquée » à partir d'un compresseur. En effet, les pompes traditionnelles ne disposant pas de contrôle de pression au sein de leur réservoir, il a été décidé de créer une pompe à vide qui s'enclencherait ou s'arrêterait lorsque la pression dans son réservoir est suffisamment faible ou trop haute. Cela permet ainsi d'avoir, indépendamment du robot, une pompe à vide toujours opérationnelle avec une pression capable de subvenir à tout moment au besoin des cavités. Pour contrôler cette pompe, il a été décidé de réaliser un circuit électronique simple (sans utilisation de microcontrôleur) composé d'un capteur de pression, d'une bascule de Schmitt, d'un relai, d'un transistor, d'un potentiomètre et d'un régulateur de tension.

Ce circuit a cependant été réalisé manuellement sur une breadboard et il était limité au niveau des valeurs de commutation. J'ai donc décidé de réaliser un nouveau circuit imprimé plus propre en reprenant ces fonctions, et en rajoutant un nouveau potentiomètre afin d'avoir la possibilité de déplacer ou écarter le cycle d'hystérésis. Cela permet au circuit de s'adapter dans le cas d'un changement de pompe ou de capteur. Le schéma et le PCB de la carte sont en annexe (cf.28)

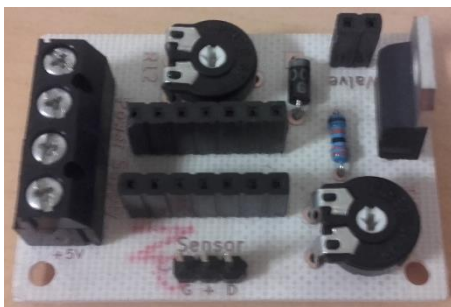


Figure 16 Circuit de contrôle de la pompe à vide

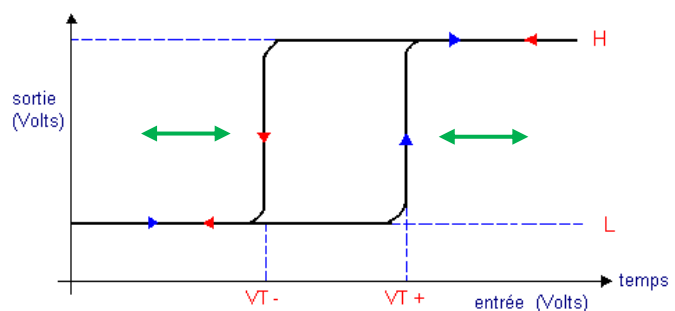


Figure 17 Cycle d'hystérésis d'une bascule de Schmitt

Après réception de la carte et des composants, j'ai pu les câbler et réaliser les premiers tests. J'ai été confronté à plusieurs problèmes dus à des faux contacts ou des valeurs de potentiomètres difficiles à régler. Mais après les avoir identifiés et corrigés, j'ai pu réaliser les tests sur la pompe à vide. J'ai pour cela réalisé un court programme afin d'afficher la valeur en pression mesurée dans le réservoir et ainsi procéder au réglage du cycle d'hystérésis d'actionnement de la pompe. Le fait d'ajouter un second potentiomètre nous a permis, comme souhaité, d'avoir plus de libertés sur ce réglage. Nous avons donc pu définir un cycle d'arrêt/allumage plus précis de la pompe : celle-ci fait désormais en sorte d'avoir en permanence une pression entre 0,6 et 0,7 bar dans son réservoir.

Cette carte d'actionnement de la pompe à vide est fonctionnelle mais il reste un faux contact au niveau de l'AOP. Celui-ci ne pose pas de problème au fonctionnement, mais il nécessite de trouver la bonne position de la carte pour l'éviter, et de ne plus la toucher. Un nouveau soudage des connecteurs ou l'impression d'une nouvelle carte seront donc nécessaires.

### 3.5 Documentation

Lors du commencement de mon PFE, je suis arrivé sur un projet qui avait débuté depuis un certain temps et auquel différentes personnes avaient contribué. Il s'est révélé que la compréhension d'un projet de la sorte est très compliquée car nous ne connaissons pas forcément toutes les réflexions, tentatives, échecs et méthodes qui ont conduit à l'état actuel du projet. Je me suis donc rendu compte de l'importance de la documentation laissée par les précédent(e)s contributeur(s) qui m'a permis d'assimiler plus facilement les avancées, les enjeux et les objectifs du projet. Je devais donc m'atteler méticuleusement à cette tâche sur mes réalisations de façon à ce que le projet puisse être poursuivi facilement et qu'il soit mené à terme.

J'ai globalement repris le format de stockage qui avait été prévu jusque-là, et j'y ai ajouté mes fichiers, mes choix et explications, les informations sur le matériel, et surtout un maximum de données. Les données étaient très peu présentes ou mal renseignées lorsque j'ai pris en main ce projet, ce qui m'a forcé à refaire entièrement certains tests moi-même. Me rendant compte de cette perte de temps, je me suis donc efforcé de ne pas faire la même erreur et d'en stocker un maximum, avec des explications et les conclusions qui en ont découlé.

Le dossier final contient donc :

- Les codes Arduino (contrôle pneumatique global, réglage de la pompe à vide et tests moteurs)
- Les projets Kicad pour les circuits imprimés (fichiers utiles et détail des composants actuels)
- Les données récoltées sur des tableurs (tableaux, graphes, calculs et explication des tests)
- Les documents utiles au projet (photos, schémas, ainsi que tous les fichiers explicatifs pour les codes, les circuits, l'état actuel du contrôle, les conclusions et les pistes d'améliorations...)

Chaque partie contient ainsi des fichiers explicatifs au format MD de façon à pouvoir stocker le projet dans un GitHub.

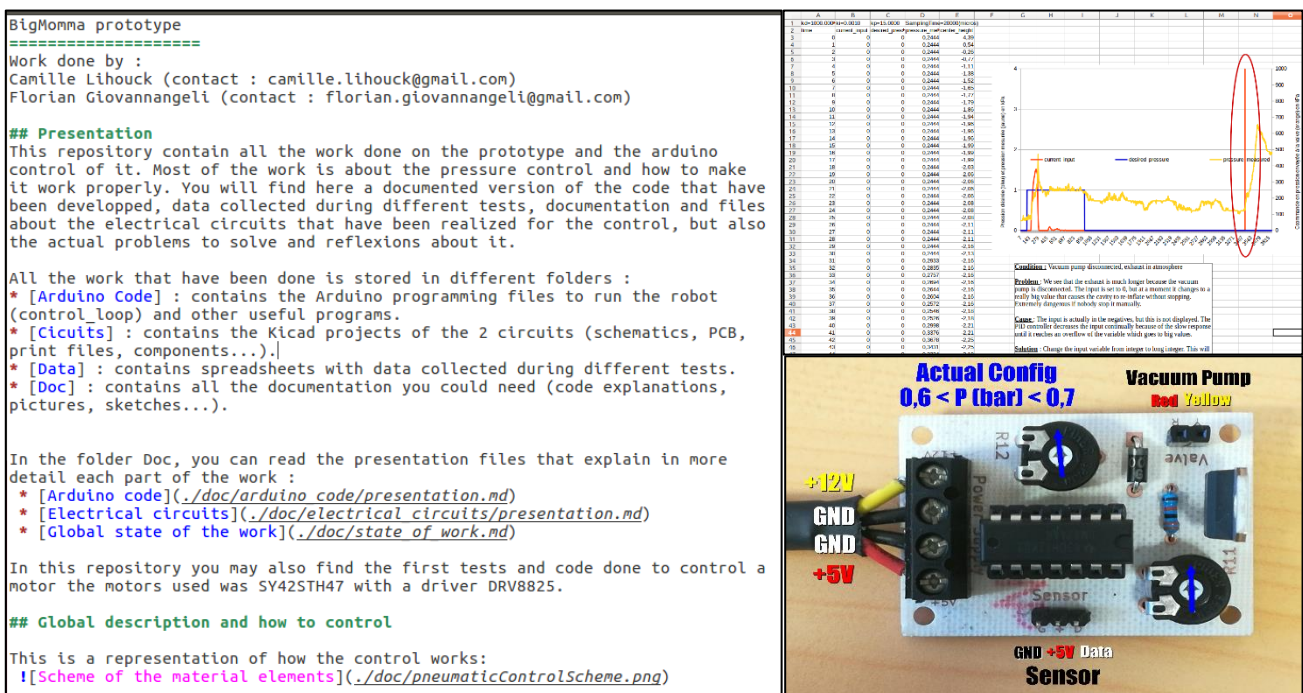


Figure 18 Exemples de fichiers de documentation du projet

## Partie 4 : Perspectives d'évolution

Mon travail sur ce projet concernait principalement le contrôle pneumatique. Ce contrôle, qui représente une grosse partie dans le contrôle global du robot, a été bien avancé mais n'est pas optimal. J'ai pu notamment constater plusieurs imperfections dans mes circuits qui peuvent parfois fausser les données collectées. Pour la suite de ce contrôle pneumatique, il sera donc nécessaire dans un premier temps de réimprimer les circuits, voire de les commander à des entreprises spécialisées, de façon à éviter les faux-contacts et les bruits parasites, et pouvoir travailler sereinement avec. Il pourrait être intéressant de profiter de cette réimpression pour apporter des améliorations au circuit, comme par exemple tenter de réduire la longueur de certaines pistes, ou modifier le type de connexion qui, comme montré précédemment, peut poser problème.

Outre les erreurs dues aux circuits imprimés, nous avons vu que la structure du robot a également une grande influence sur la réponse du contrôle pneumatique. Cette dernière est limitée notamment par les appareils utilisés. Si l'on veut encore améliorer la réponse en pression, il sera donc nécessaire d'utiliser une pompe à vide plus performante (avec un plus gros réservoir) pour dégonfler rapidement le robot. L'utilisation d'une électrovanne pour faire l'intermédiaire entre la valve, la pompe à vide et l'air libre pourra être étudiée car celle-ci devrait permettre de stabiliser la réponse et de limiter l'utilisation de pompe, facilitant l'état stationnaire. Le compresseur semble pour l'instant suffisant pour une seule cavité, mais il risque d'être également problématique lorsqu'il faudra gonfler rapidement 4 cavités.

Concernant les cavités justement, il va falloir procéder à leur installation afin de pouvoir rapidement tester le contrôle pneumatique complet. Cela comprend le passage des tuyaux, l'installation des capteurs et le changement des disques intermédiaires, le tout en essayant de garder le robot le plus hermétique pour faciliter son contrôle.

Une fois que toutes ces modifications physiques auront été faites, le programme de contrôle pourra alors être implémenté. Dans un premier temps il faudra effectuer un nouveau réglage du PID qui devrait, avec ces améliorations, se faire plus facilement. Le programme actuel n'étant fonctionnel que pour une seule cavité, il sera nécessaire de mettre en place le contrôle pour les 4 cavités. Cette partie devrait être assez rapide car le programme change peu, seules les contraintes de chaque cavité devront être définies et intégrées.

Une fois que toutes ces étapes seront faites, le contrôle pneumatique devrait alors être complet. Il faudra alors faire le parallèle avec l'utilisation des servomoteurs pour le contrôle de la direction. Les deux parties une fois fonctionnelles, elles pourront être modélisées dans le logiciel SOFA pour réaliser des simulations et ensuite mettre en place un contrôle complet par ordinateur.

## Conclusion

Au travers de ce Projet de Fin d'Etudes, j'ai pris part à un projet de recherche assez pointu faisant appel à un certain nombre de connaissances et compétences techniques.

Dès le début, je me suis rendu compte qu'il n'est pas aisé de rejoindre un projet déjà bien entamé par plusieurs autres personnes, et qu'une description approfondie du travail effectué est primordiale. Bien que j'en ai déjà un peu fait l'expérience, j'ai à nouveau constaté que la recherche se résume souvent à des études ou des tentatives qui ne sont pas toujours concluantes, mais qui permettent d'avancer et de cibler de plus en plus l'objectif. Cela se solde souvent par des écarts entre la théorie et la pratique, faisant ainsi apparaître des problèmes inattendus qu'il faut comprendre et résoudre par un raisonnement et des expériences.

Mon travail a débuté par des études de circuit, des tests et des recherches d'améliorations. Celles-ci ont conduit à la réalisation d'une carte de contrôle pneumatique complète et polyvalente qui permettra d'effectuer des essais plus poussés sur le robot afin d'affiner encore son contrôle. Une seconde carte a également été réalisée afin de contrôler l'actionnement de la pompe à vide. Ces circuits imprimés ont ensuite pu être testés en situation réelle. Bien que globalement fonctionnels, ils nous ont permis de mettre en lumière d'autres problèmes et imperfections inattendu(e)s que nous avons pu ou qu'il faudra résoudre. Du côté programmation, j'ai peu modifié le programme car il s'est trouvé être plutôt complet et bien organisé. Je me suis en revanche plutôt intéressé au réglage du PID et à la recherche de méthodes d'amélioration grâce à des seuils afin de rendre le contrôle pneumatique plus précis et performant. Pour que mon travail puisse être compris, réutilisé et amélioré, il a enfin fallu que je le stocke, le documente et l'explique minutieusement. De cette manière, comme je l'ai fait au départ, le prochain utilisateur pourra rapidement assimiler l'état actuel du projet et prendre la suite sans tarder.

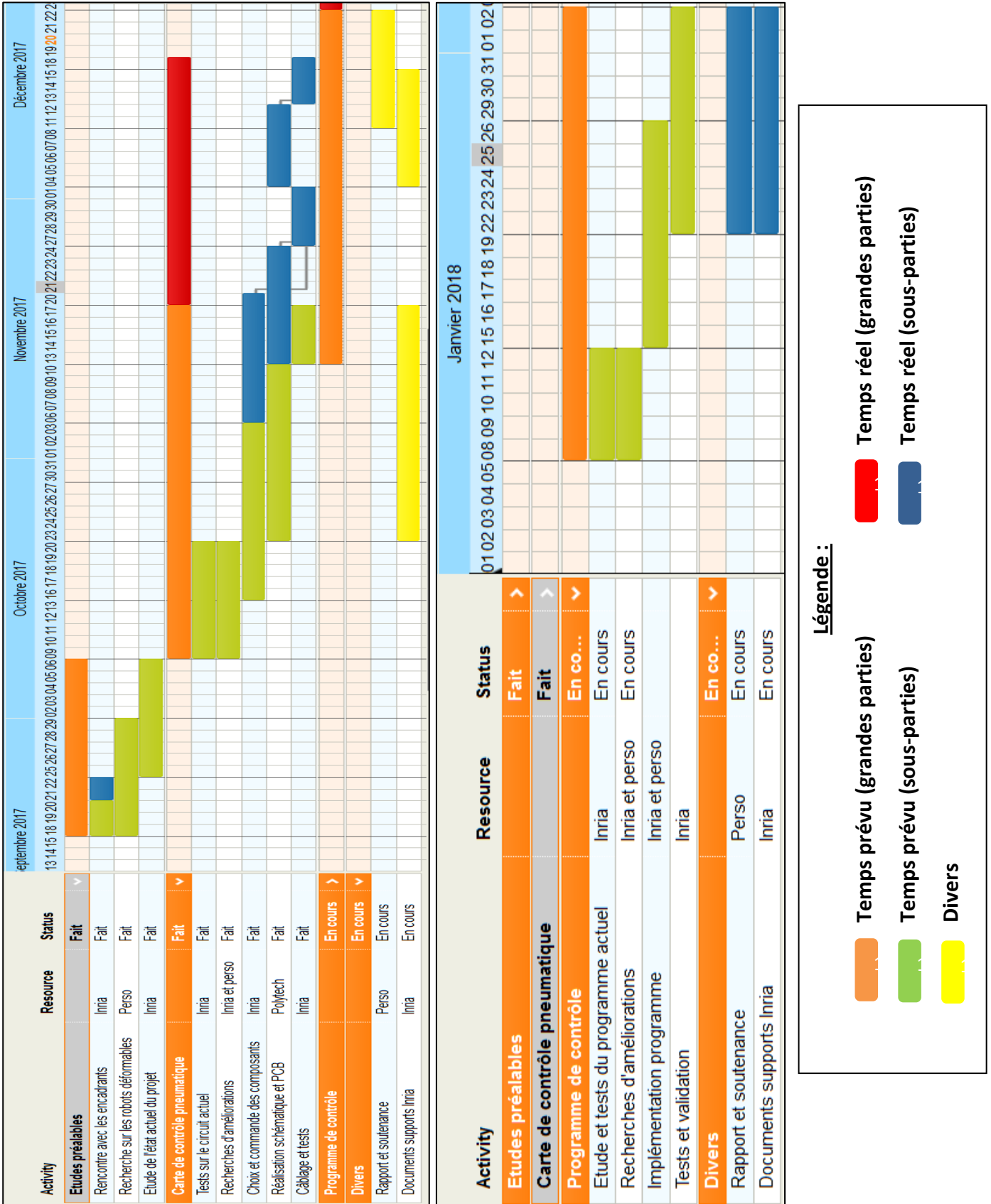
Le système global de contrôle pneumatique ainsi réalisé durant mon projet est fonctionnel et nous donne l'opportunité d'avancer dans les recherches sur ce robot. Il se révèle être un prototype efficace pour réaliser des tests et déceler les problèmes et modifications à apporter. Il possède quelques erreurs et imperfections qu'il faudra corriger afin d'avoir un contrôle pneumatique performant et de pouvoir passer sur les autres aspects du robot tels que les servomoteurs.

Ce projet de bras déformable antagoniste présente de grands enjeux quant à la robotique déformable. En effet, avec ce système d'actions réciproques, il permet d'assimiler toujours plus les problématiques de ces robots et nous avance sur la possible fabrication et le contrôle de robots déformables de grande taille possédant une rigidité et donc une force importante.

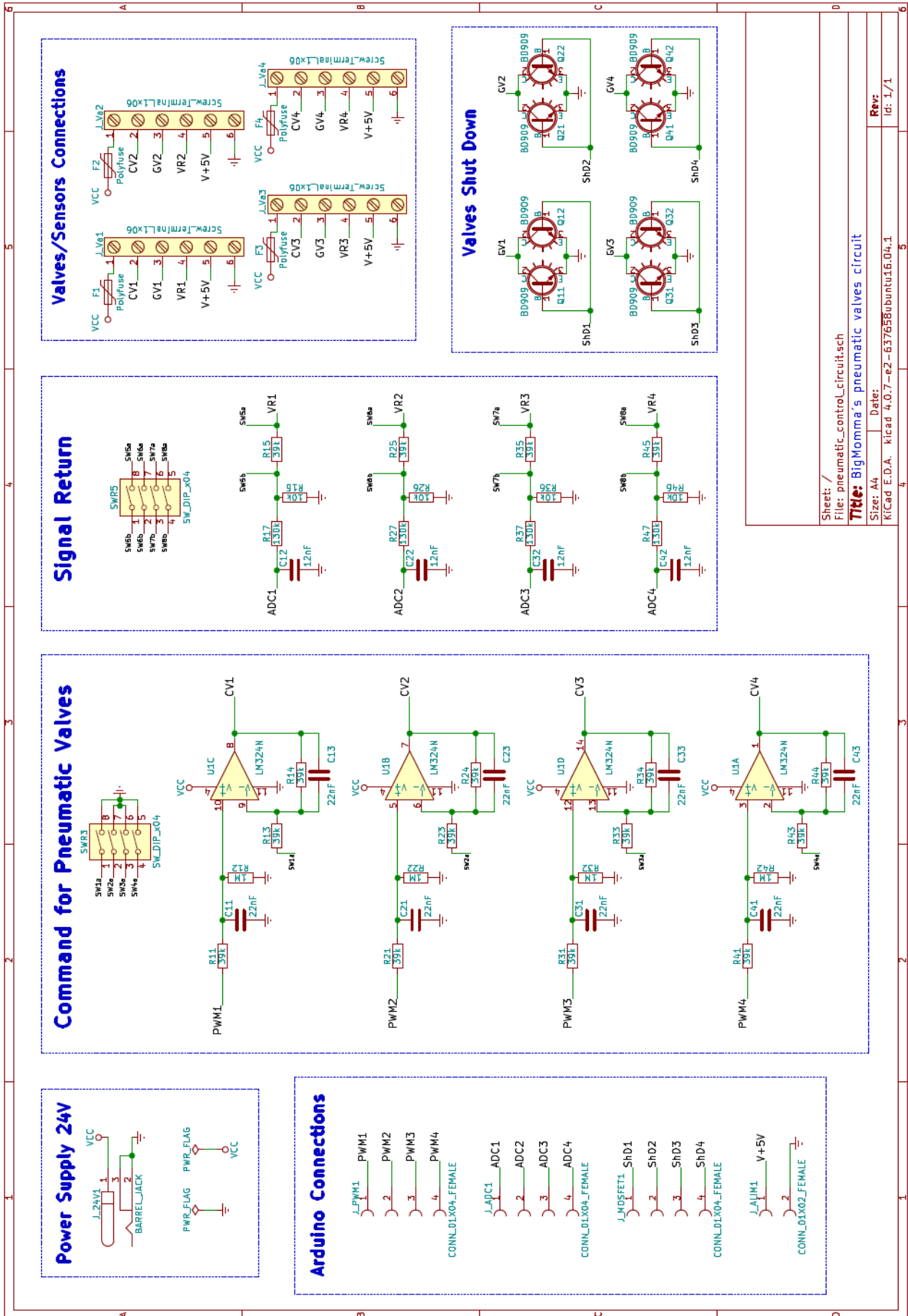


# Annexes

## a. Diagramme de Gantt de l'organisation initiale du travail

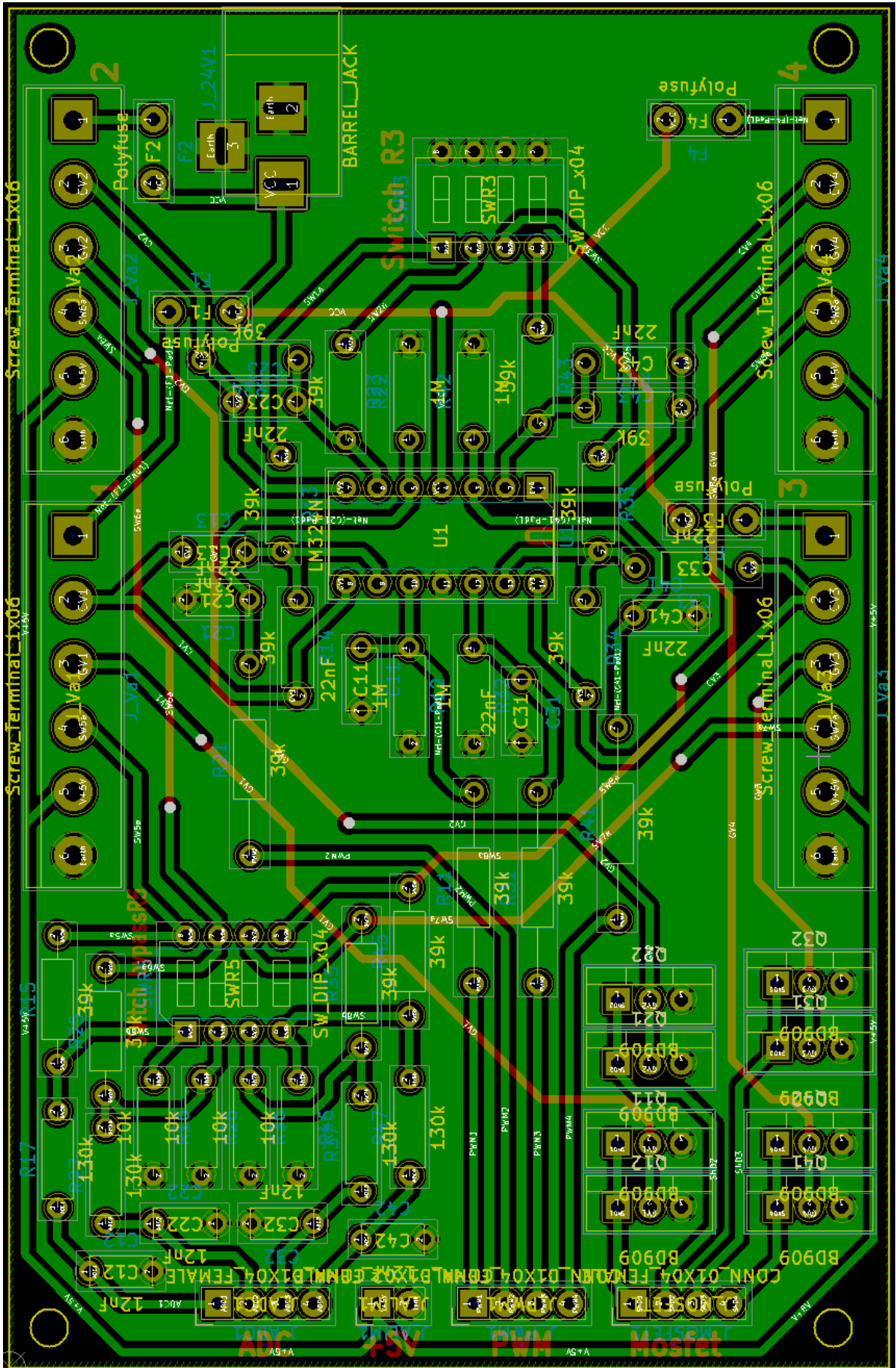


## b. Schématique de la carte de contrôle pneumatique

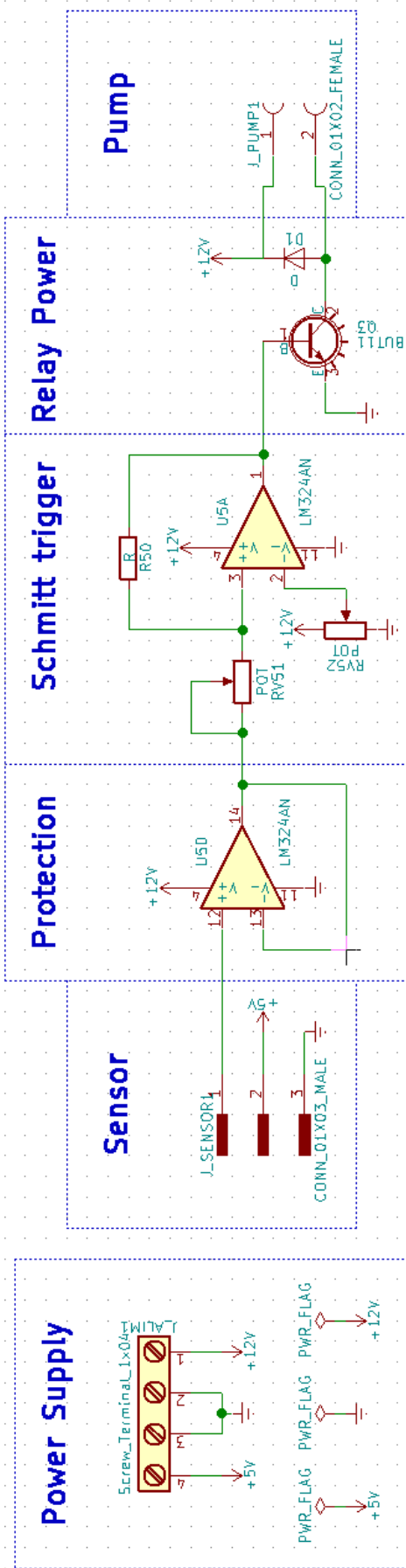


Sheet: /  
 File: pneumatic\_control\_circuit.sch  
**Title:** BigMamma's pneumatic valves circuit  
 Size: A4  
 KiCad E.D.A. kicad 4.0.7-e2-637658ubuntu16.04.1  
 Date:  
 Rev: 1/1  
 Id: 1/1

### c. PCB de la carte de contrôle pneumatique



### d. Schématique de la carte de contrôle de la pompe à vide



**e. Photos illustrées des deux cartes réalisées**

