

Rapport de stage

Laboratoire Informatique de l'Ecole Polytechnique

Du 23 février 2009 au 27 février 2009

Alexandre GORDIEN

I. Sommaire

I.	Sommaire	2
II.	Remerciements	4
III.	Présentation du projet	5
	A. Cadre du projet.....	5
	B. Objectif final	5
IV.	Réalisation du projet	6
	A. Définition des caractéristiques physiques et mécaniques du robot.....	6
	1. Détermination des possibilités de mouvement du robot	6
	2. Modélisation 3D	6
	B. Vision par ordinateur.....	9
	1. Matériel, spécifications et contraintes.....	9
	2. Suivi des mouvements en temps réel	9
	C. Electronique	10
	1. Cartes de puissance et moteurs	10
	2. Carte de commande	11
	3. Alimentation.....	11
	D. Commande du robot	12
	1. Deux modes de commande.....	12
	2. Programmes de commandes.....	13
	E. Organigramme récapitulatif	15
	F. Conclusion	16
V.	Annexes	17
	A. Sérigraphies et schémas.....	17
	1. Carte de puissance STEP-EASY.....	17

2. Carte de commande COMIO	19
B. Banques de photos.....	20
C. Sources	23
1. Programmation.....	23
2. Mathématiques	23
3. Autres	23
Figure 1 - Blender, modélisation 3D, mode "fil de fer"	7
Figure 2 - Blender, modélisation 3D, mode "texturing"	7
Figure 3 - Blender, rendu final : 10.23 secondes.....	8
Figure 4 - tpe_opencv_vision.exe.....	9
Figure 5 - Montage de la carte STEP – EASY - 1.....	10
Figure 6 - Montage de la carte STEP - EASY - 2	10
Figure 7 - Raccord de la carte COMIO à l'ordinateur (port parallèle)	11
Figure 8 - Câble de type RJ-11 4 contacts non-inversé	11
Figure 9 - Simulateur OpenGL	13
Figure 10 - client.exe	14
Figure 11 - Sérigraphie de la carte STEP - EASY.....	17
Figure 12 - Schéma de la carte STEP - EASY	18
Figure 13 - Sérigraphie de la carte COMIO.....	19
Figure 14 - Schéma de la carte COMIO	19
Figure 15 - Le Laboratoire Informatique de l'X	20
Figure 16 - Elaboration de l'algorithme de calcul des angles.....	20
Figure 17 - L'Ecole Polytechnique	21

II. Remerciements



Je remercie Monsieur James REGIS de son enseignement, de son total soutien et de sa motivation, ainsi que l'équipe du LIX pour son excellent accueil et ses encouragements.

Je remercie le laboratoire de mécanique pour la réalisation des pièces en métal pour l'armature du robot et les membres de l'atelier qui nous ont accueillis pour le montage des cartes.

Je remercie également Monsieur le Directeur du Lycée Teilhard de Chardin, Pierre PELLE, pour la confiance qu'il a bien voulu m'accorder en me délivrant les conventions de stage pour l'École Polytechnique chaque fois que cela s'est avéré nécessaire et de l'appui de Madame Moncelet et de Madame Le Mevel.

Enfin, je remercie mes parents pour leur implication et leur engagement financier.

III. Présentation du projet

A. Cadre du projet

Ce projet s'insère dans la réalisation des TPE (Travaux Personnels Encadrés) du programme de première scientifique. Le thème choisi étant la télé-chirurgie, j'ai imaginé un moyen d'en illustrer le principe, au moyen d'un bras robotisé contrôlé par ordinateur.

B. Objectif final

L'objectif final de ce projet est d'avoir réalisé un robot opérationnel. Equipé d'une lame, il devra nous permettre de découper, à distance et sans vue directe sur le robot, un bloc de matière gélatineuse.

IV. Réalisation du projet

A. Définition des caractéristiques physiques et mécaniques du robot

1. Détermination des possibilités de mouvement du robot

a. Forme générale

Le robot devra avoir la forme d'un bras manipulateur classique, possédant une surface d'action en deux dimensions. Il comportera 3 parties et une lame.

Les deux premiers bras auront une longueur de 150mm et le troisième aura une longueur de 70mm. La longueur de lame n'est pas fixe, elle sera adaptée pour une découpe la plus facile possible.

b. Modifications ultérieures

Pour des raisons de masse et de résistance de matériaux, le robot comportera une articulation de moins (donc un moteur de moins) : la lame sera fixée directement sur le bras et non plus sur un moteur.

2. Modélisation 3D

a. Blender

Blender est un puissant logiciel de modélisation 3D. Il est gratuit et permet donc à n'importe qui de s'essayer « 3Deiste ». En plus de la modélisation, il propose diverses méthodes d'animation, gère de nombreux formats (import/export), permet des simulations de fluides, comprend un moteur physique capable de gérer collisions et déformations d'objets, un scripteur Python pour créer ses propres plug-ins et bien d'autres choses encore.

Blender m'a permis de modéliser le robot et de mieux visualiser les contraintes mécaniques auxquelles il pouvait être soumis.

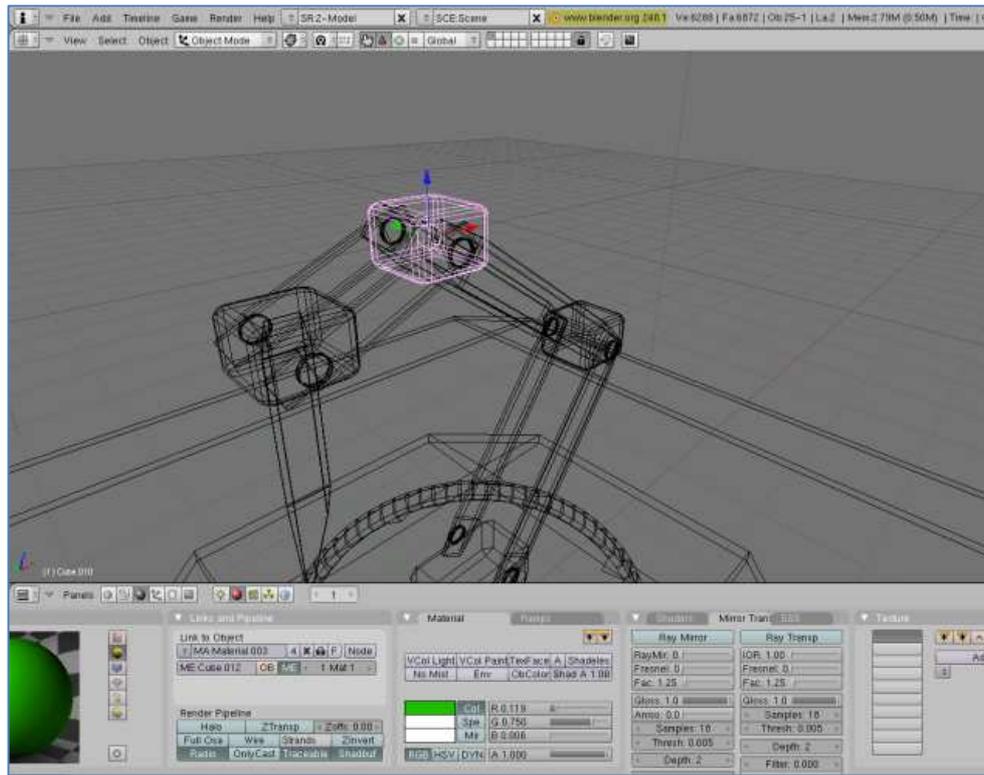


Figure 1 - Blender, modélisation 3D, mode "fil de fer"

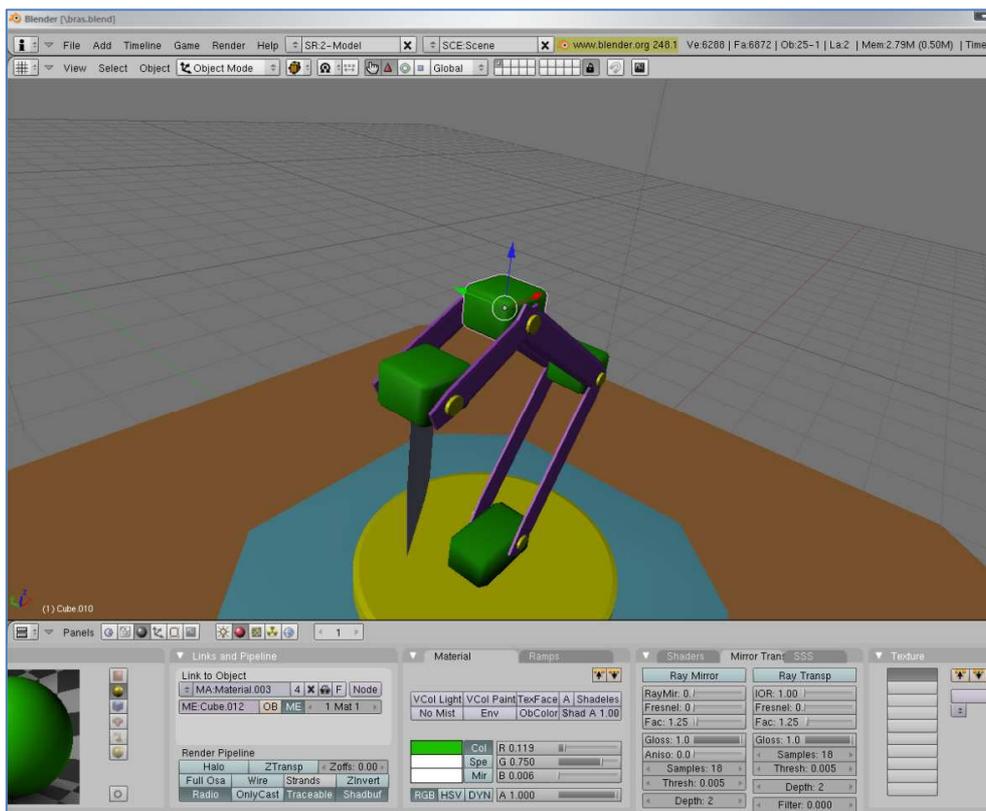


Figure 2 - Blender, modélisation 3D, mode "texturing"

b. Rendu final

Voici le rendu final de la modélisation, avec textures, jeux de lumières autres effets :

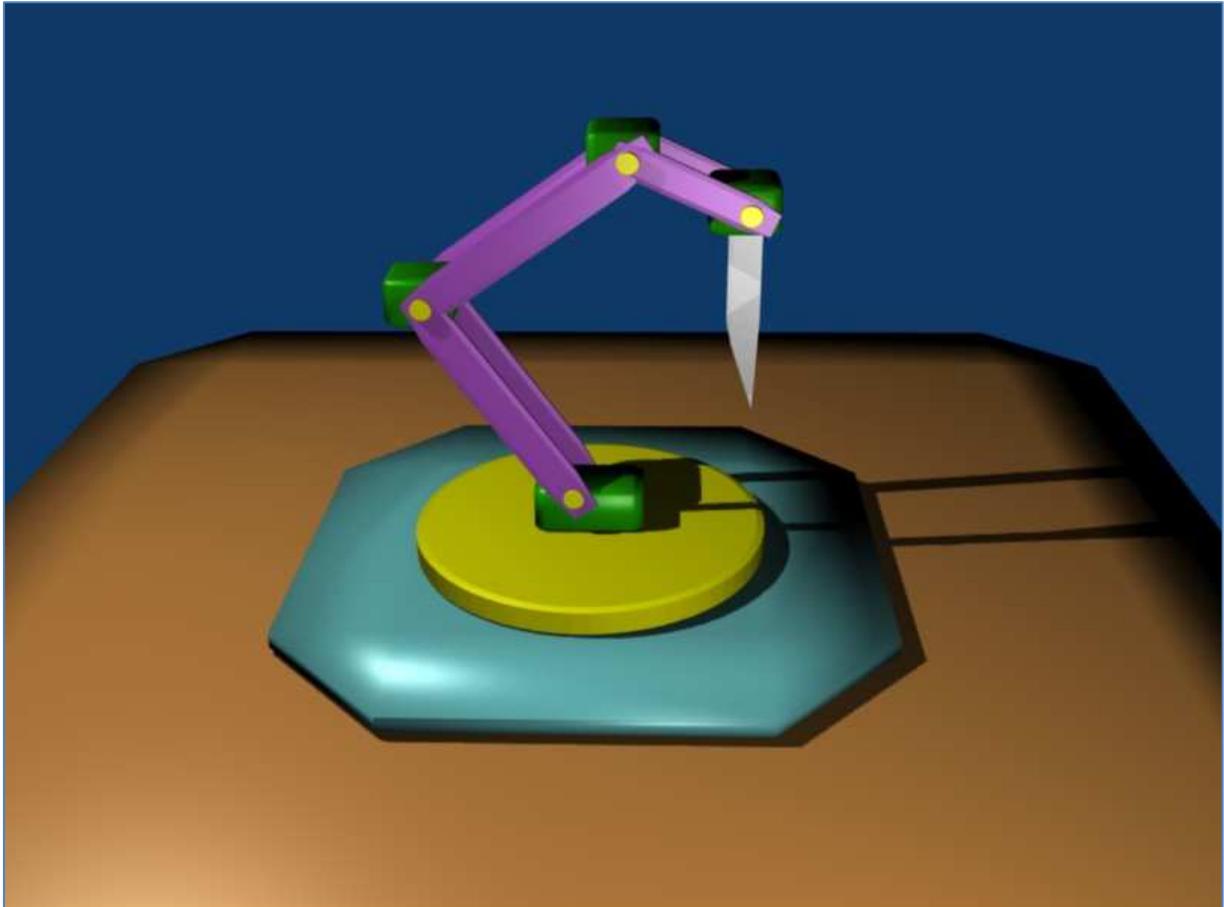


Figure 3 - Blender, rendu final : 10.23 secondes

B. Vision par ordinateur

C'est par nos yeux que nous percevons la majorité des informations qui nous permettent de comprendre et d'interagir avec le monde qui nous entoure. La vision par ordinateur (ou VAO) est un domaine technologique très dynamique et en pleine effervescence, tant du point de la recherche scientifique que des applications de la vie quotidienne. Elle permet le comptage, la reconnaissance, l'analyse et des possibilités de traitement d'images immenses.

1. Matériel, spécifications et contraintes

Dans ce projet la vision par ordinateur me permet de vaincre la distance qui me sépare du robot physique. J'ai écrit deux programmes en langage C++ qui me permettent d'utiliser un périphérique de type Camera. J'ai utilisé la *library* OpenCV créée par la société Intel, qui propose un ensemble d'objets et de fonctions en C/C++ liés à la vision par ordinateur. Elle a pour mérite d'être gratuite, libre et multi-plate-formes, c'est-à-dire qu'elle est utilisable aussi bien sous Windows que sous GNU/Linux ou Mac OS. Les deux programmes interagissent avec la webcam Logitech QuickCam E3500 Series, connectée via un port USB 2.0 de l'ordinateur. Malgré la portabilité d'OpenCV, les programmes ne peuvent fonctionner que sur une plate-forme Microsoft Windows car je me suis également servi de fonctions mises à disposition par l'API de Microsoft.

2. Suivi des mouvements en temps réel

Le suivi des mouvements du robot se fait en temps réel grâce au second programme : *tpe_opencv_vision.exe*. Une fois la connexion établie avec la webcam, l'utilisateur choisit des points sur l'image renvoyée en cliquant sur les articulations. Le programme calcule les changements de positions des objets et replace les points sur chaque frame du flux vidéo de sorte qu'un point indique toujours la même articulation. Il fonctionne indépendamment de la luminosité de l'image.

Figure 4 - tpe_opencv_vision.exe



C. Electronique

1. Cartes de puissance et moteurs

Pour ce projet, les cartes de puissance sont au nombre de 3. Il s'agit de cartes STEP – EASY achetées non montées sur le site Web www.conrad.fr. Elles sont livrées avec chacune un moteur pas-à-pas Japan Servo /200 pas et nécessitent 12 Vdc / 0,5 A. Les trois cartes ont été assemblées à la main.

La carte STEP – EASY est une interface électronique qui permet le contrôle manuel ou par PC (ou automate, microprocesseur, ordinateur) d'un moteur pas-à-pas unipolaire (5, 6 ou 8 fils), dont la tension d'alimentation peut varier sur une large plage de 2 à 24V avec un courant maximal de 1A par bobine.

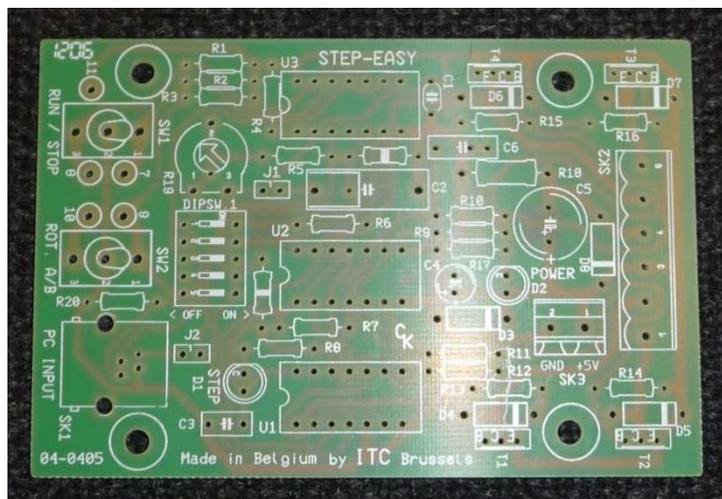


Figure 5 - Montage de la carte STEP – EASY - 1

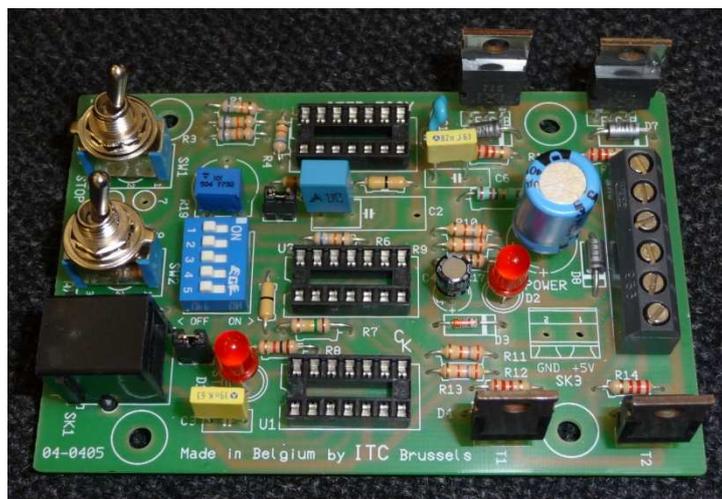


Figure 6 - Montage de la carte STEP - EASY - 2

2. Carte de commande

Les cartes STEP – EASY présentées ci-dessus sont, en mode PC, contrôlées par une carte de commande COMIO. Il s'agit d'une interface offrant 8 sorties et 4 entrées digitales toutes indiquées par des LED.

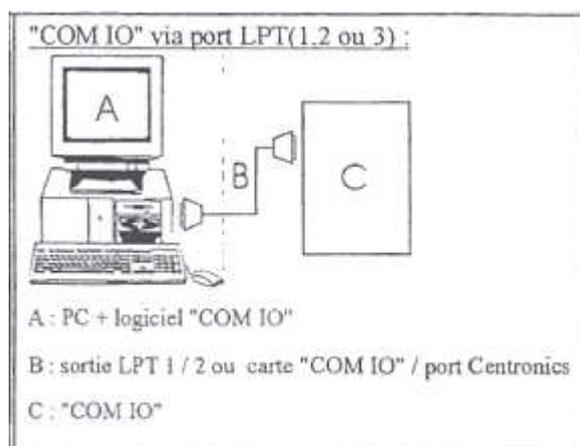


Figure 7 - Raccord de la carte COMIO à l'ordinateur (port parallèle)

Le raccord avec les cartes STEP – EASY se fait via les connecteurs MJ-1, MJ-2, MJ-3, MJ-4 à l'aide d'un câble de type RJ-11 4 contacts **non-inversé**.

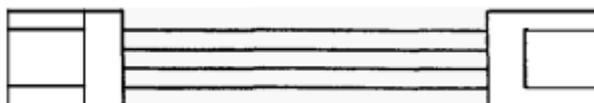


Figure 8 - Câble de type RJ-11 4 contacts non-inversé

3. Alimentation

La carte de commande, les cartes de puissance et les moteurs sont alimentés par deux alimentations Schaltnetzteil-Baustein SP1500/01 (une pour alimenter la carte de commande et une pour alimenter cartes de puissance et moteurs). Elles sont conçues pour des appareils d'une consommation maximale de 1500 mA. Chacun des moteurs nécessitant une alimentation de 500 mA, ce module est donc *théoriquement* capable de fournir de l'électricité aux 3 moteurs en même temps. Pour prévenir tout risque d'accident, **IL EST ABSOLUMENT NECESSAIRE DE RELIER CET APPAREIL A LA TERRE.**

D. Commande du robot

1. Deux modes de commande

a. Mode Manuel

Le mode manuel consiste à alimenter ou non chacun des moteurs et à définir le sens de rotation des moteurs à l'aide des interrupteurs présents sur les cartes de puissance.

b. Mode PC

Calculs de trajectoire, cinématiques directe et inverse

En mode PC, j'utilise une partie du principe de cinématique inverse. Le simulateur OpenGL comprend un algorithme permettant de trouver les coordonnées de chacun des points du bras en ne connaissant que les coordonnées du point de départ et celles du point de l'extrémité du bras ainsi que les longueurs de chacune de ses parties.

La cinématique directe correspond à une méthode de calcul qui permet de calculer la position dans l'espace de l'extrémité d'un « bras » articulé à partir de la position dans laquelle se trouve chacune de ses articulations. La cinématique inverse est « tout simplement » le contraire Elle permet de piloter tous les axes du robot uniquement en partant de la position que l'on désire atteindre.

Cependant, il arrive, selon la structure du bras et selon la coordonnée désirée, que les équations de cinématique inverse ne puissent trouver de solution. Cela peut venir de 3 causes principales :

- Les équations n'ont pas été établies correctement
- Les coordonnées à atteindre et/ou le vecteur d'approche sont hors de la zone couverte par le bras.
- Les équations butent sur une redondance : la structure du bras permet d'atteindre une certaine coordonnée avec un vecteur d'approche donné selon plusieurs configurations différentes

Il a donc fallu adapter l'algorithme pour parer à d'éventuelles absences de solutions.

2. Programmes de commandes

a. Contrôle par le logiciel fourni avec la carte de commande

La carte de commande est livrée avec un logiciel permettant de contrôler les sorties de cette carte vers les moteurs. Il nous permet de réaliser des tests et d'ajuster robot et algorithme.

b. Simulateur OpenGL et console système

Le simulateur OpenGL est un programme écrit en langage C utilisant les *librairies OpenGL* et *SDL* qui permettent de dessiner sur une surface. Il sert à définir les mouvements du bras robotisé : soit le contrôle se fait moteur par moteur à l'aide du clavier, soit il se fait seul, grâce à la cinématique inverse, le point de l'extrémité étant défini par un clic de souris.

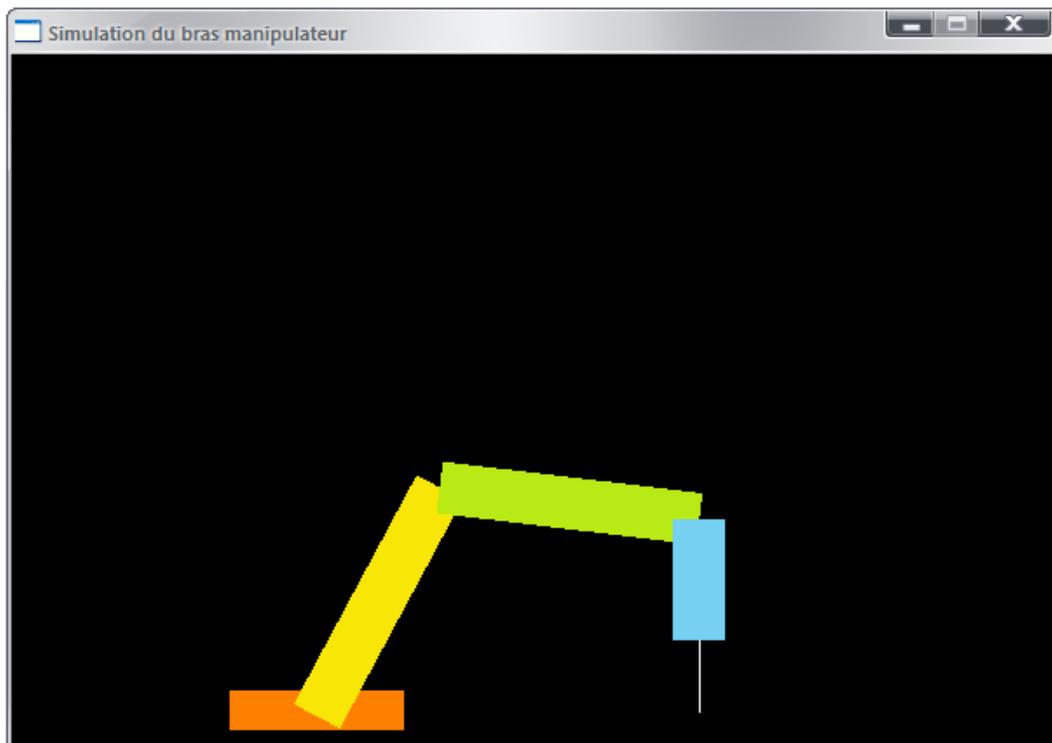
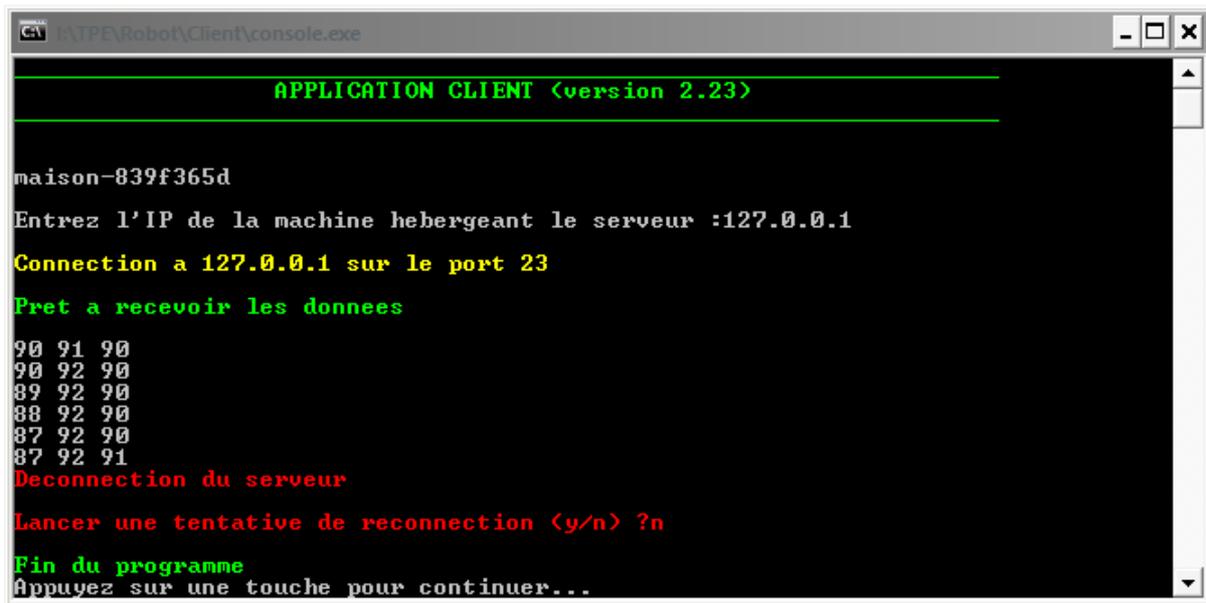


Figure 9 - Simulateur OpenGL

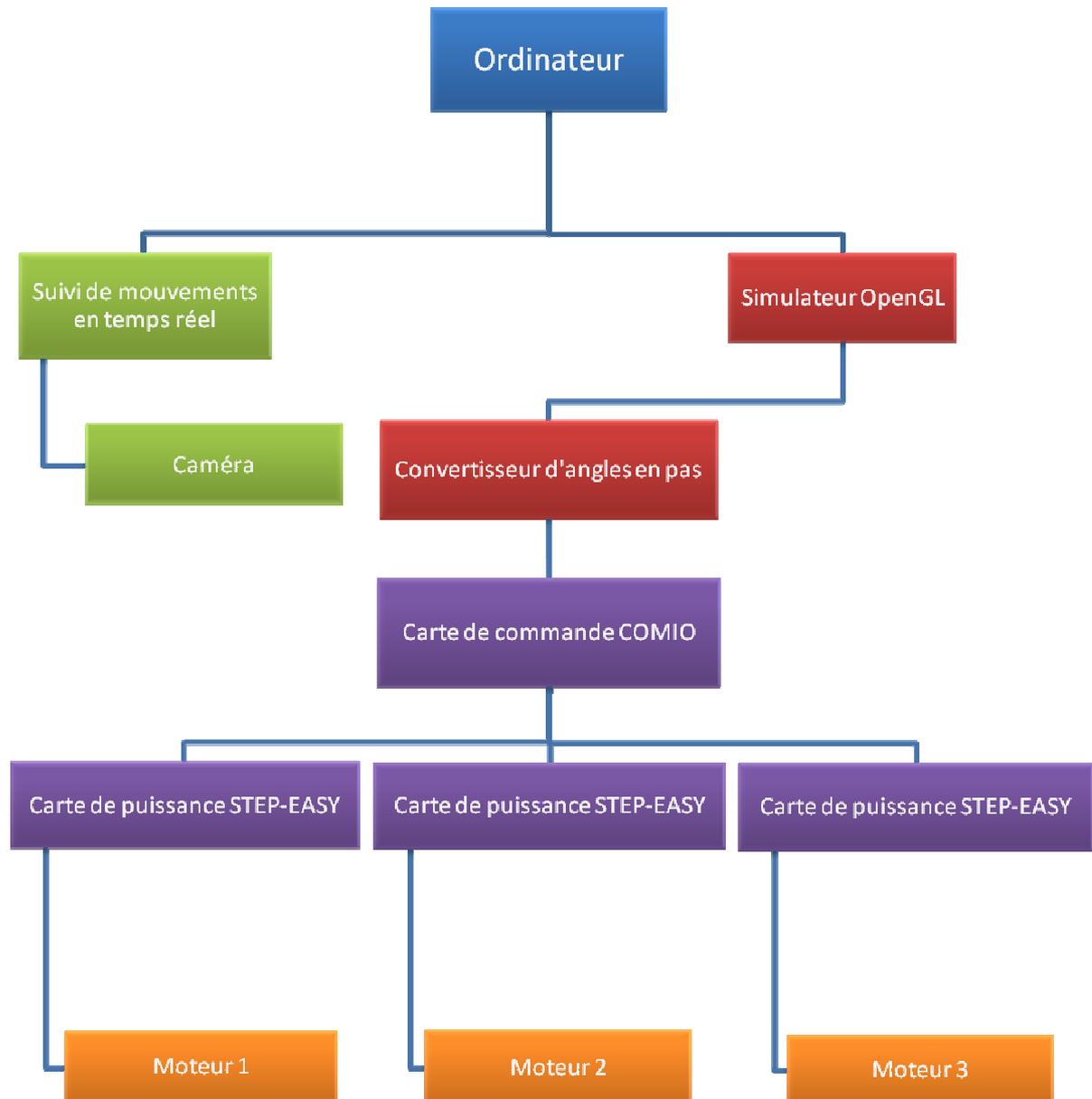
Le simulateur OpenGL envoie les angles de chacune des articulations à un autre programme en C, client.exe, au moyen des *sockets*. Celui convertit les valeurs reçues en pas et les envoie à la carte de commande COMIO via le port parallèle de l'ordinateur.



```
C:\ATPE\Robot\Client\console.exe
APPLICATION CLIENT (version 2.23)
maison-839f365d
Entrez l'IP de la machine hebergeant le serveur :127.0.0.1
Connection a 127.0.0.1 sur le port 23
Pret a recevoir les donnees
90 91 90
90 92 90
89 92 90
88 92 90
87 92 90
87 92 91
Deconnection du serveur
Lancer une tentative de reconnection (y/n) ?n
Fin du programme
Appuyez sur une touche pour continuer...
```

Figure 10 - client.exe

E. Organigramme récapitulatif



F. Conclusion

Durant ce court stage, j'ai pu utiliser et acquérir des notions mathématiques, physiques, mécaniques, algorithmiques, informatiques et électroniques. Guidé par Monsieur REGIS, j'ai élaboré la structure physique d'un robot ayant la forme d'un bras manipulateur et réalisé une grande partie de son système de commande.

Une fois le projet mené à terme, il me permettra non seulement d'illustrer le principe de la télé-chirurgie lors ma soutenance, mais également d'avoir un point de départ pour mes projets de robotiques personnels. Il s'agira pour moi de développer ce bras manipulateur, notamment au niveau de la vision par ordinateur. J'envisage aussi une saisie d'objets à l'aide d'une pince, etc...

Ce second stage au Laboratoire Informatique de l'Ecole Polytechnique renforce encore ma motivation à faire de la robotique mon métier et partager cette passion.

V. Annexes

A. Sérigraphies et schémas

1. Carte de puissance STEP-EASY

a. Sérigraphie

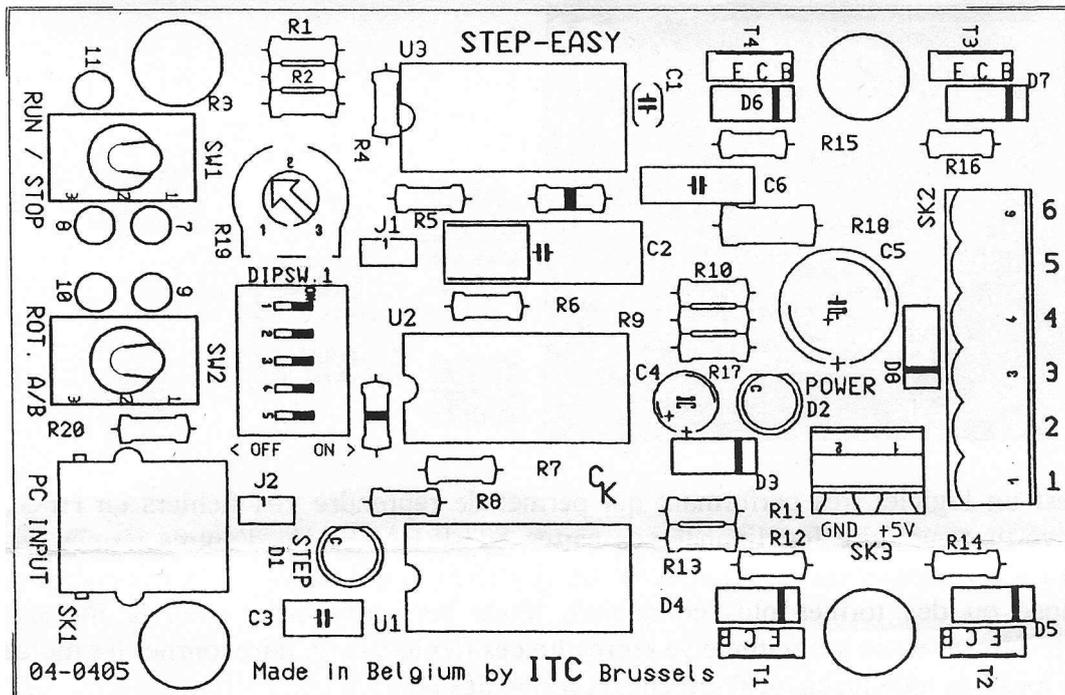


Figure 11 - Sérigraphie de la carte STEP - EASY

b. Schéma

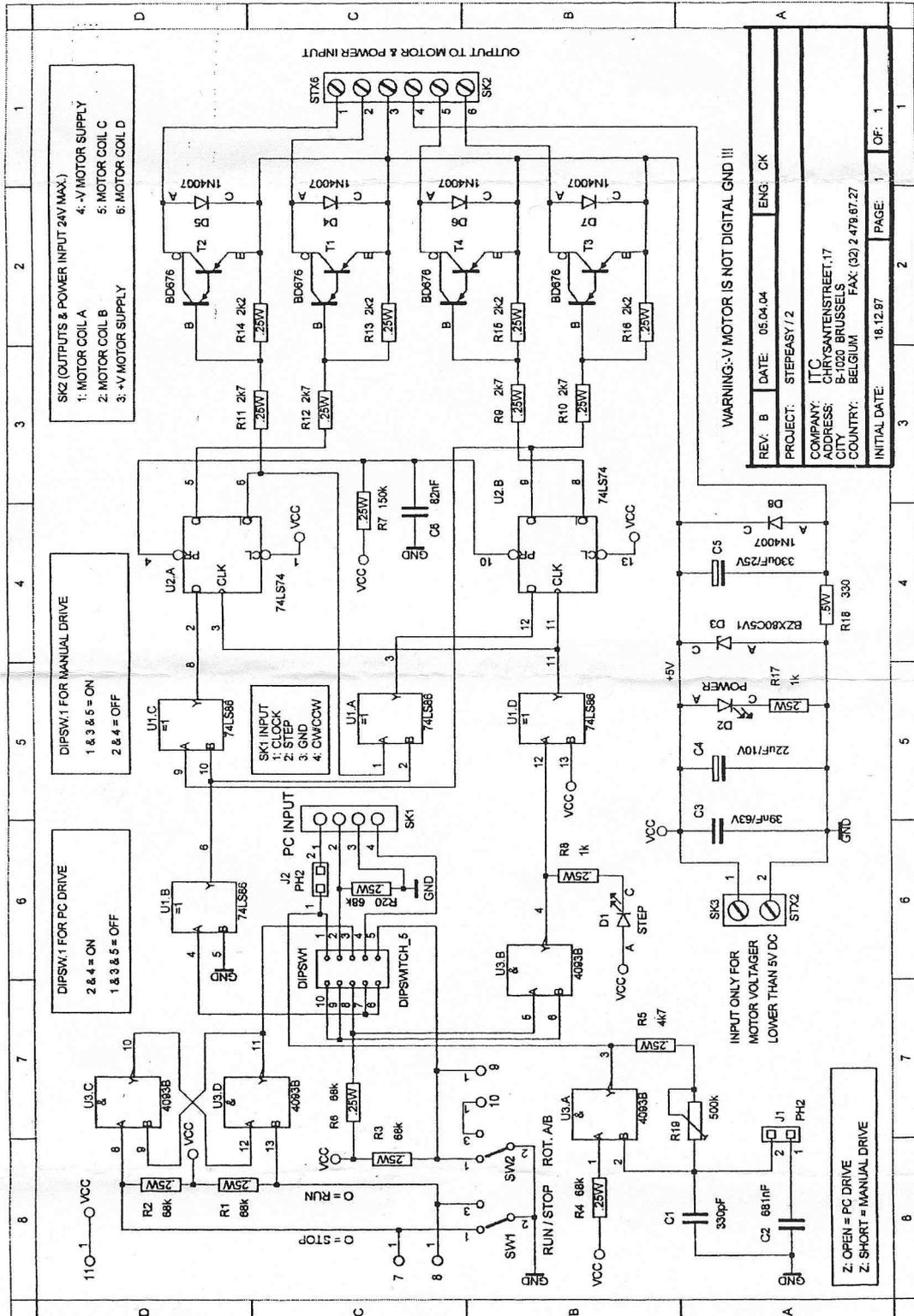


Figure 12 - Schéma de la carte STEP - EASY

2. Carte de commande COMIO

a. Sérigraphie

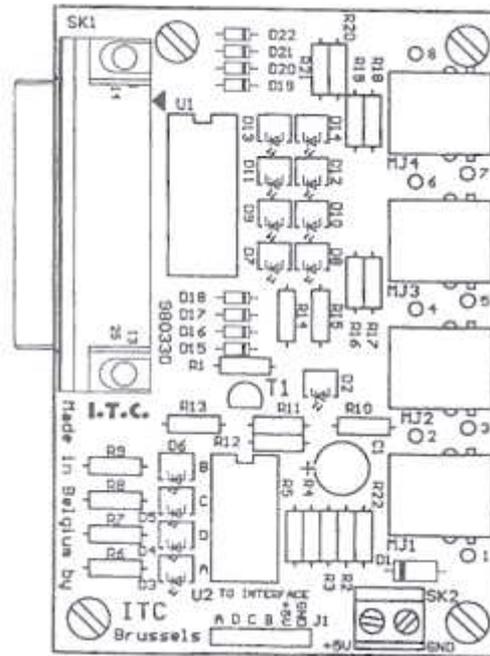
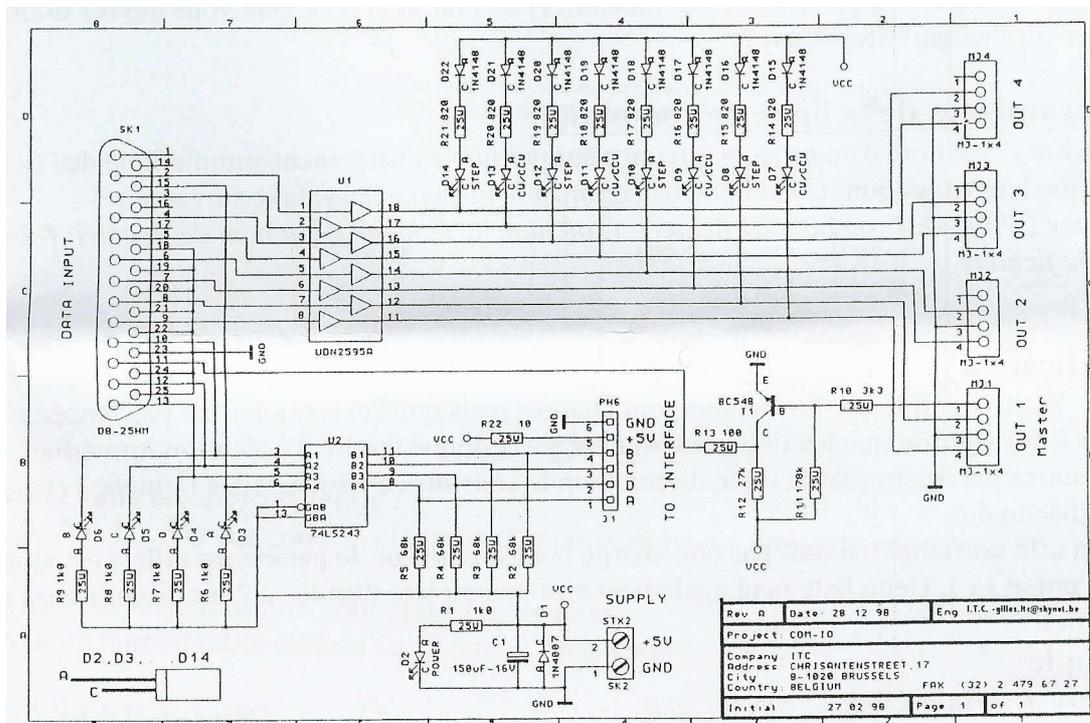


Figure 13 - Sérigraphie de la carte COMIO

b. Schéma

Figure 14 - Schéma de la carte COMIO



B. Banques de photos



Figure 15 - Le Laboratoire Informatique de l'X

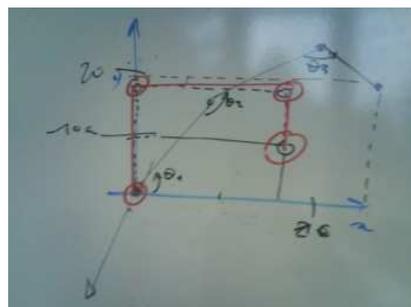
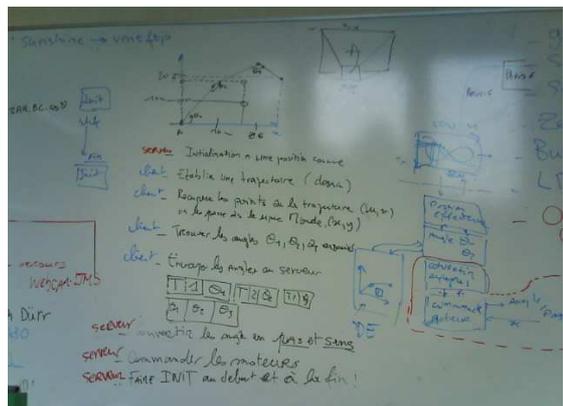
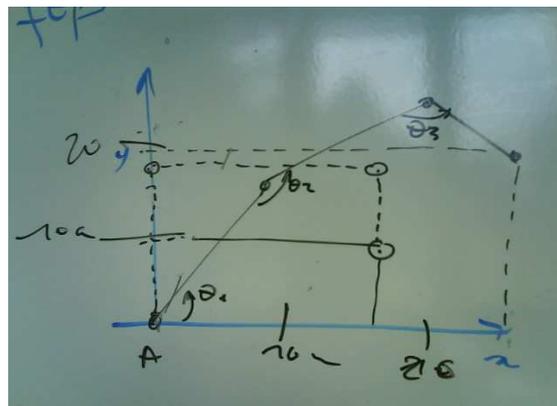
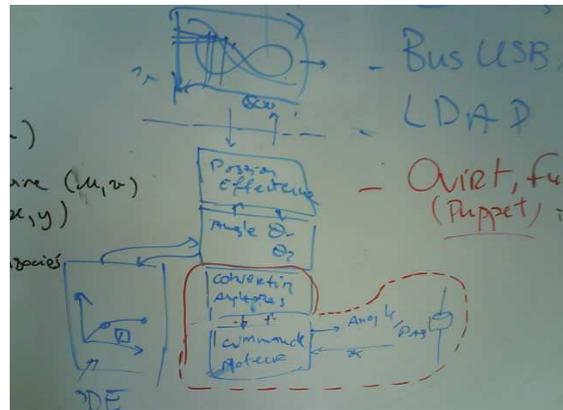
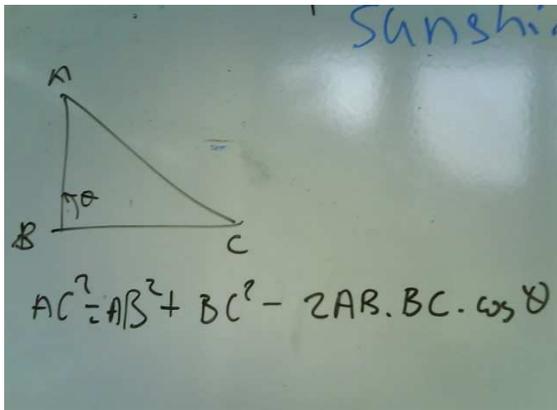


Figure 16 - Elaboration de l'algorithme de calcul des angles

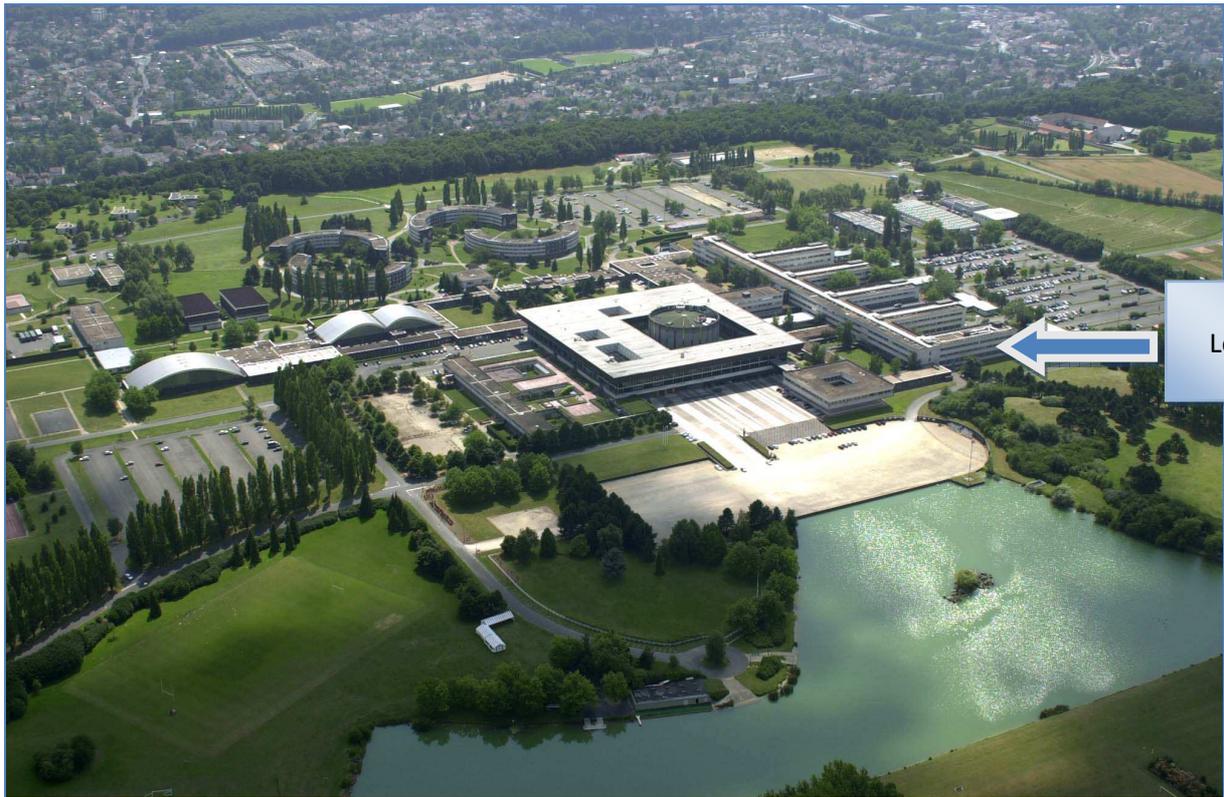
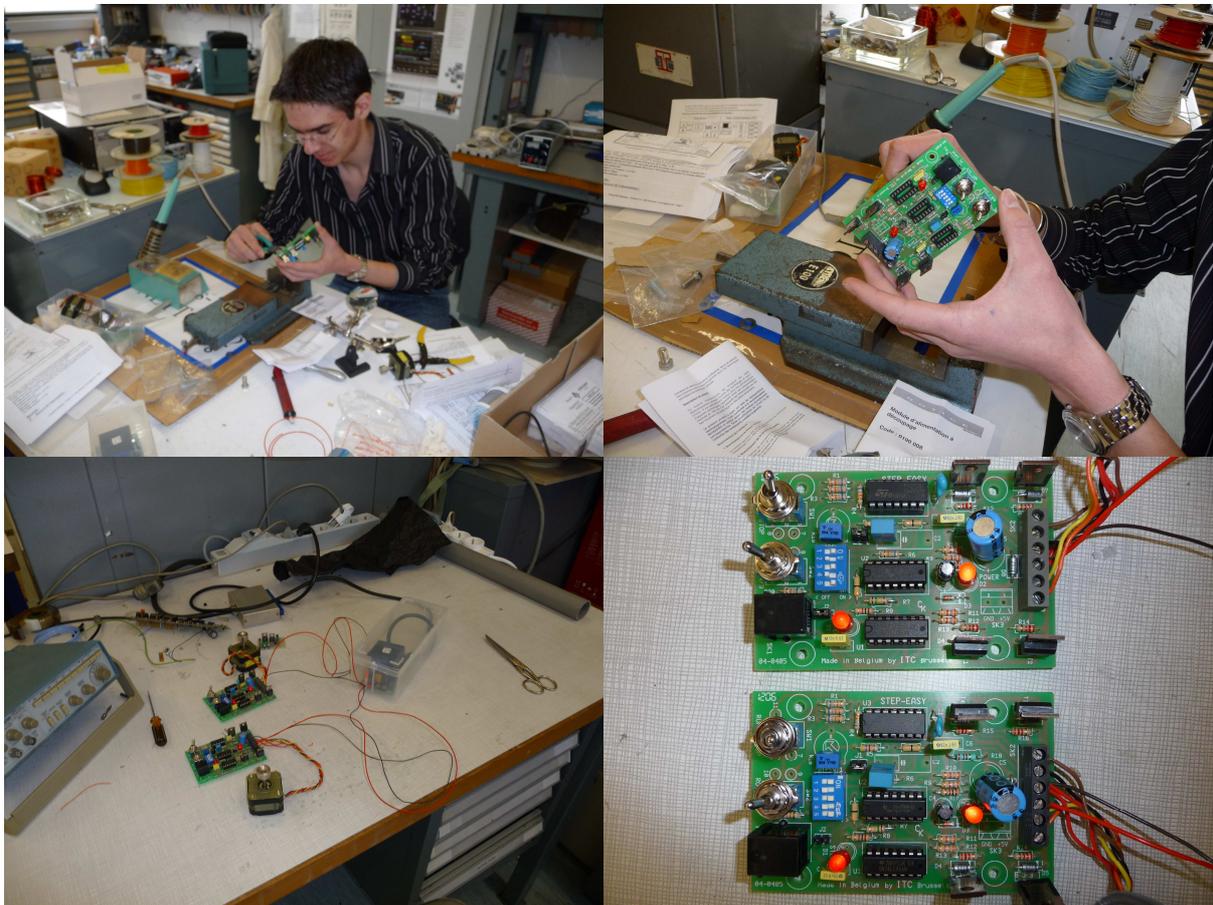
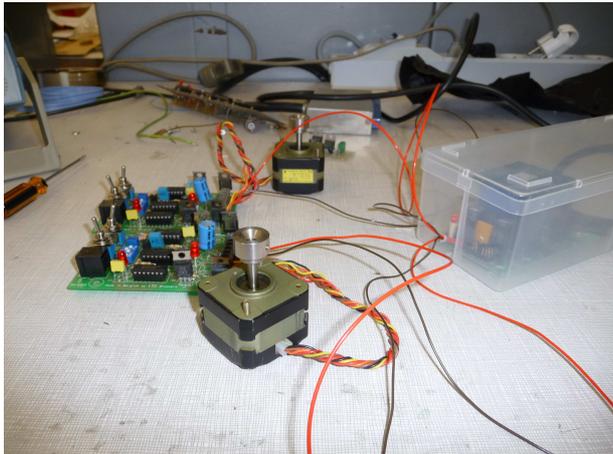
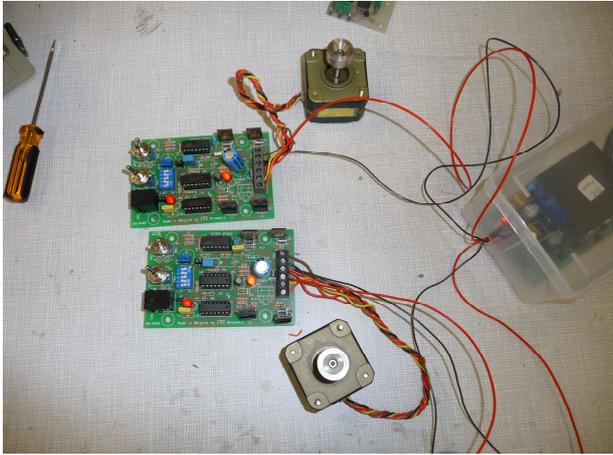


Figure 17 - L'Ecole Polytechnique





C. Sources

1. Programmation

www.siteduzero.com

www.cppfrance.com

www.developpez.com

<http://opencv.willowgarage.com/wiki/>

<http://msdn.microsoft.com/fr-fr/default.aspx>

www.polymorphe.org

www.vbfrance.com

Programmation en Langage C – 3^{ème} édition de Stephen Kochan - Editions Pearson Education

Le langage C Norme ANSI – 2^{ème} édition de Brian W.Kernighan et Dennis M.Ritchie - Editions Dunod - Sciences Sup

C++ Pour Les Nuls - 2^{ème} édition de Stephen Randy Davis

2. Mathématiques

http://fr.wikipedia.org/wiki/Cin%C3%A9matique_inverse

<http://www.learnaboutrobots.com/inverseKinematics.htm>

http://fr.wikipedia.org/wiki/Applications_de_la_trigonometrie

3. Autres

www.wikipedia.org

www.conrad.fr

www.sphynx.be