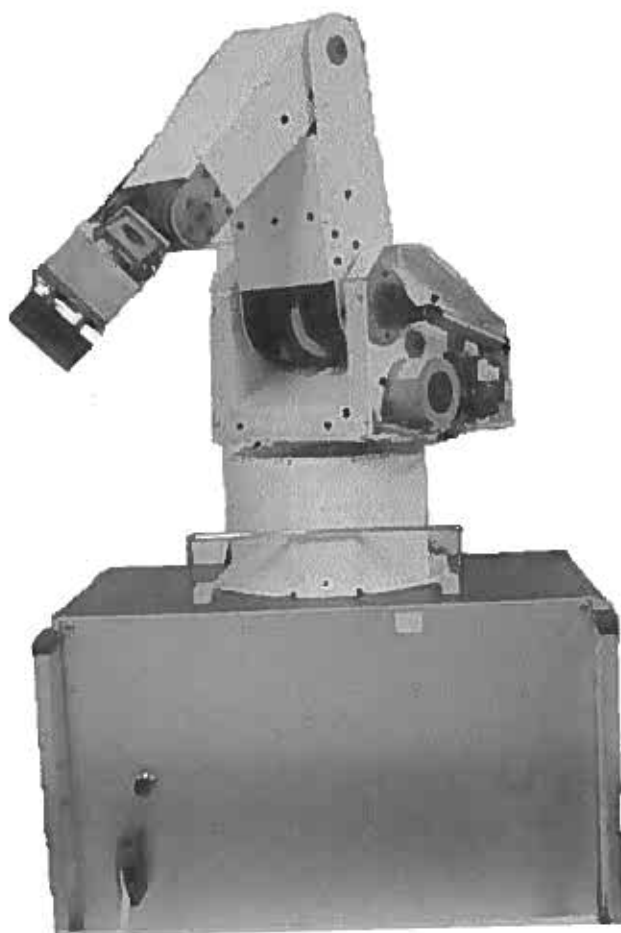


**Bellec Maël
Huon Gurvan
Dupont Marc**

TS5

TPE : Le bras manipulateur

DOSSIER



Année 2002-2003

Index

♦ Introduction

A.Etude Mécanique

- 1) Analyse Fonctionnelle**
- 2) Diagramme Pieuvre**
- 3) Cinématique**
 - a) Vue d'ensemble du robot
 - b) Les mouvements simples de rotation
 - c) Les mouvements nécessitant plusieurs moteurs
- 4) Conception**
 - a) Plan de fixation
 - b) Atèle de blocage du Bras

B.Etude Electrique

- 1) Fonctionnement général**
- 2) Organisation de la boîte**
- 3) Principe du moteur Pas à Pas**
- 4) Table de vérité des moteurs**
- 5) Capteurs embarqués**
- 6) Critiques des solutions**

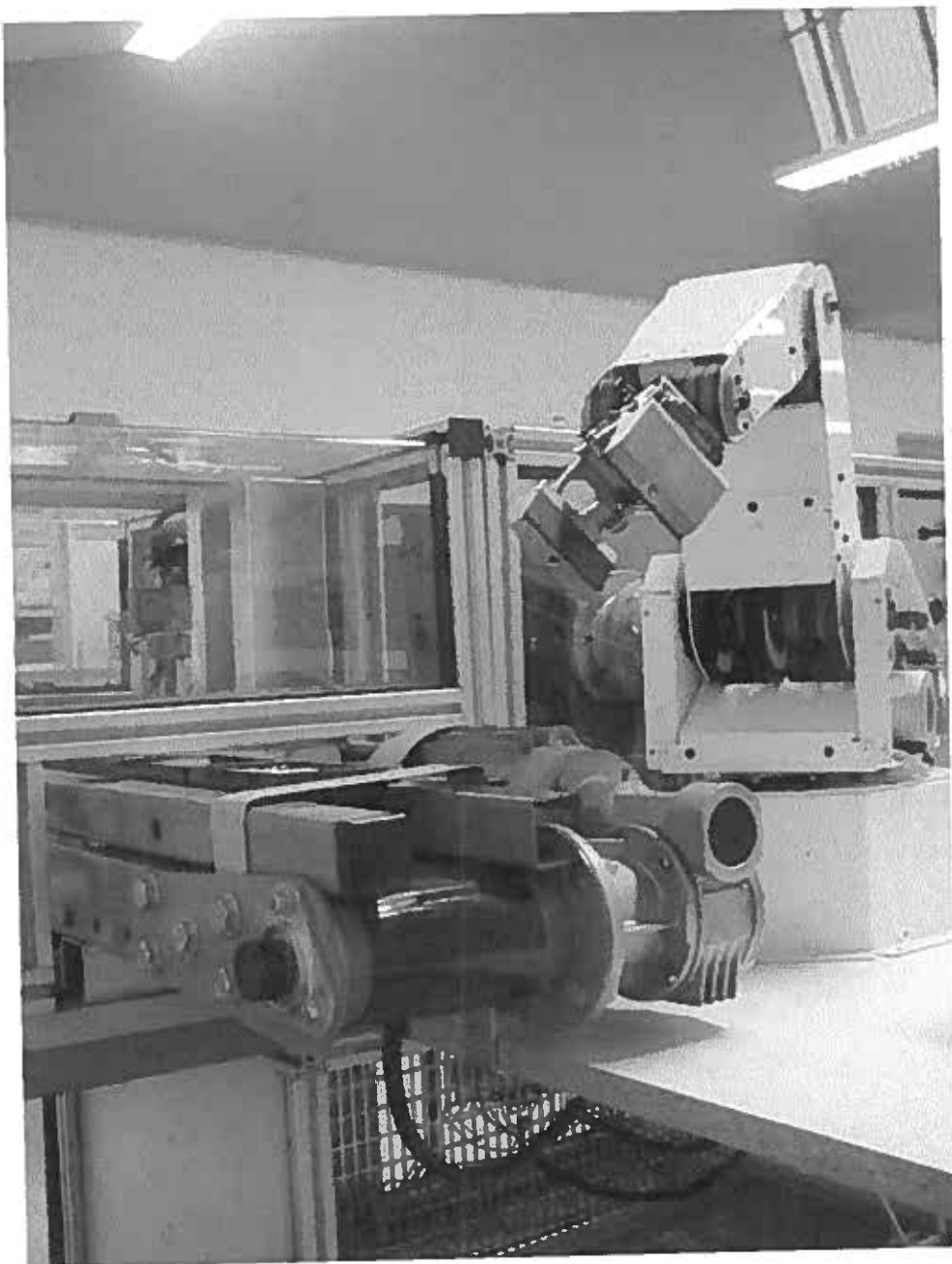
C.Programmation

- 1) Ancien programme**
- 2) Nouveau programme**
 1. Présentation
 2. Fonctions
- 3) Aide**
- 4) Sources du programme**
 1. Pour bien exploiter les sources
 2. Sources commentées

Introduction

Le but de ce projet est d'arriver à rendre fonctionnel le bras manipulateur, qui comme chacun sait, n'est plus opérationnel depuis longtemps. En effet le précédent programme de gestion est obsolète et inutilisable. Grâce à une étude approfondie de la partie mécanique et électrique du robot nous sommes parvenus à la création d'un nouveau logiciel, plus simple et plus performant. Grâce à l'utilisation du logiciel Delphi 6. Ainsi le robot peut être adapté pour une utilisation industrielle telle que l'automatisation de tâches répétitives (rangement, conditionnement etc), dangereuses (produits toxiques, milieu malsain).

Afin de mener à bien se projet, 3 grandes étapes seront successivement abordées. Nous commencerons par l'étude du robot lui même, donc la partie mécanique. Ensuite nous verrons la partie électrique ou électronique et nous finirons par le logiciel créé, donc de la programmation. Une partie conception a été rajoutée à la partie mécanique, elle s'est tout naturellement greffé à celle ci de part son contenu que vous découvrirez plus tard. Bonne lecture !

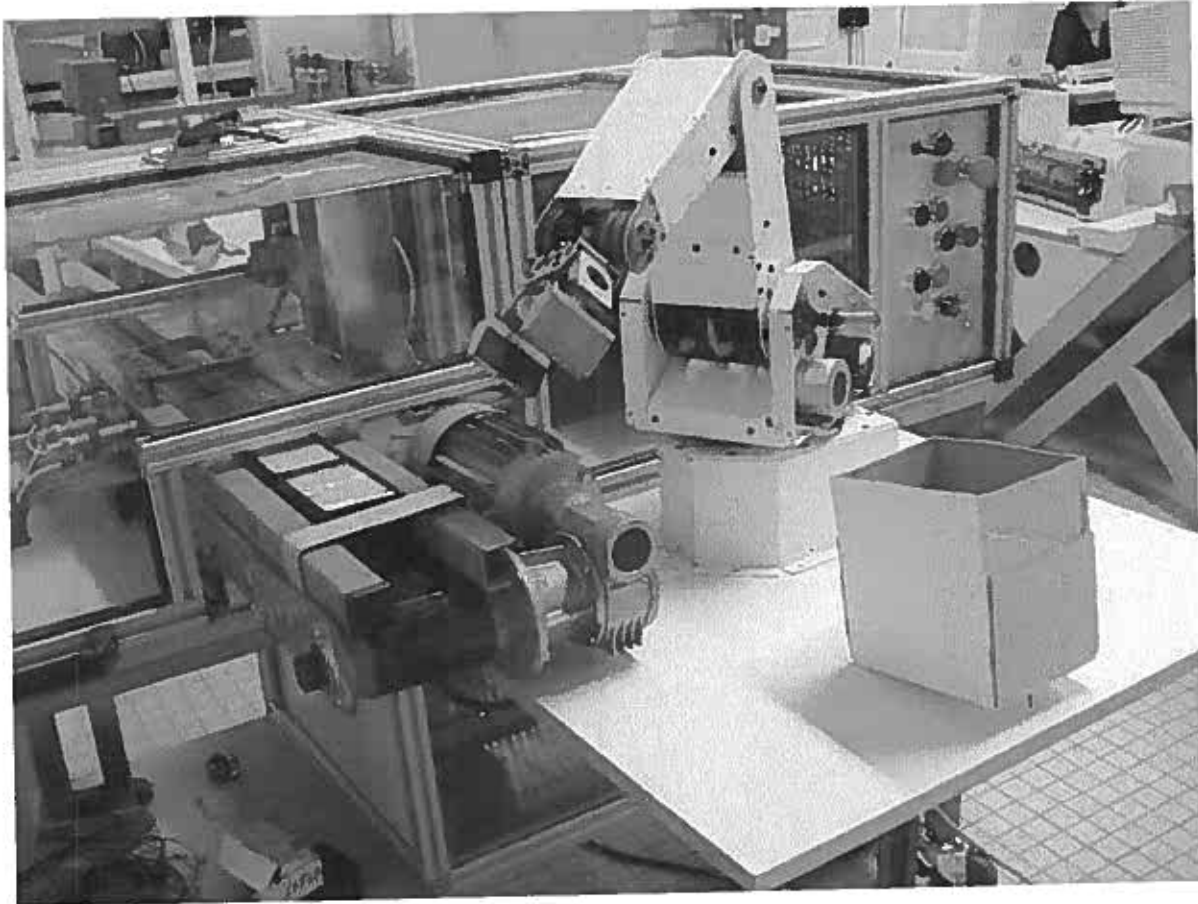


A) Etude mécanique et conception

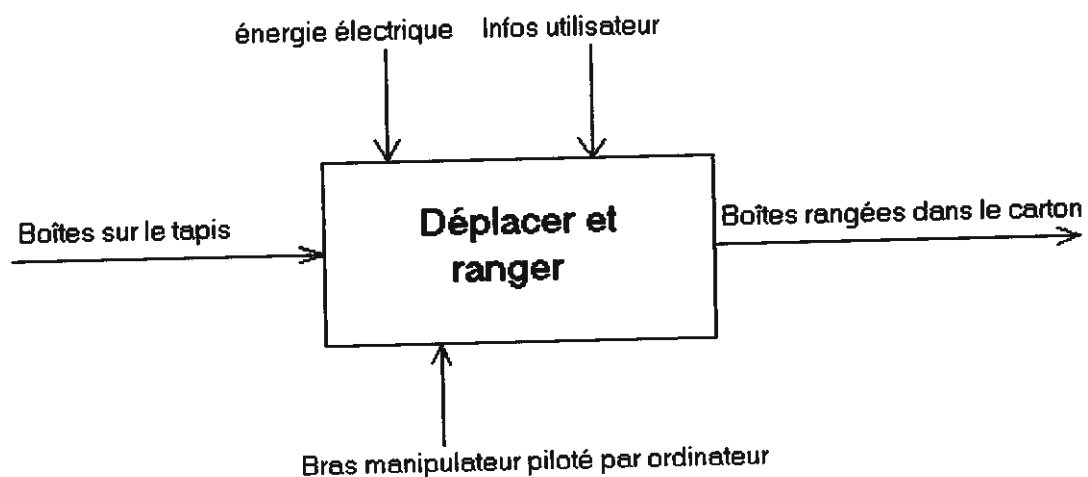
1) Analyse fonctionnelle

Présentation du Projet :

Notre TPE est orienté sur la réalisation d'un programme informatique destiné à piloter le bras manipulateur ainsi que sur l'étude mécanique du robot.



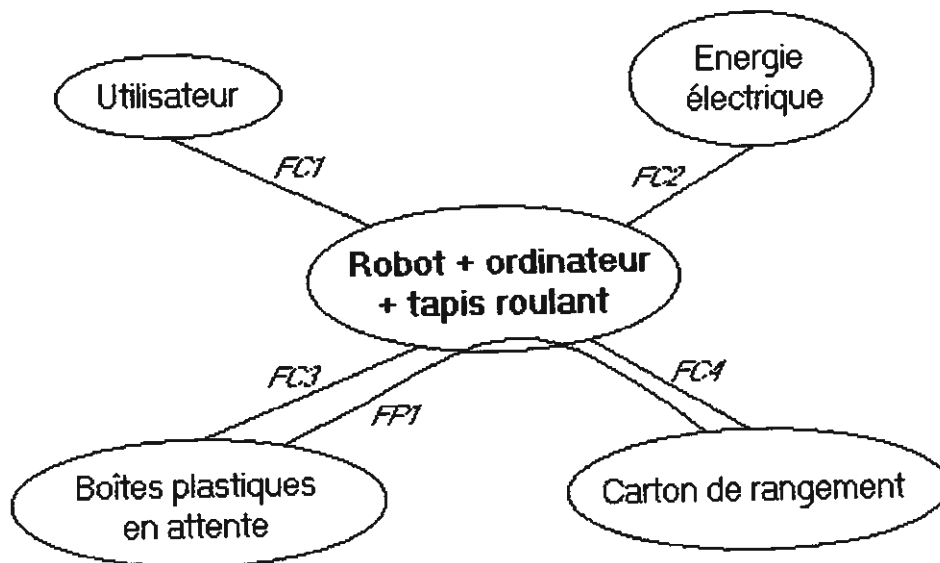
Le robot devra saisir des boîtes amenées par tapis roulant pour les ranger dans un carton.



Le robot fonctionne en mode automatique : il effectue ainsi une tâche répétitive,

comme mettre les boîtes dans le carton. Cependant, il peut être mis en mode manuel, en le commandant avec un joystick relié à l'ordinateur. Cela pourra servir dans des manipulations de produits dangereux, là où il serait trop risquer qu'une personne intervienne directement.

2) Diagramme pieuvre du système



- FP1 : Ranger les boîtes en attente sur le tapis.
- FC1 : Permettre à l'utilisateur de donner des consignes
- FC2 : Alimenter en courant électrique le système.
- FC3 : Doit arrêter les boîtes sur le tapis pour que le robot puisse les prendre.
- FC4 : Le carton de rangement doit être près du système.

● Finalité du robot et de son système de commande

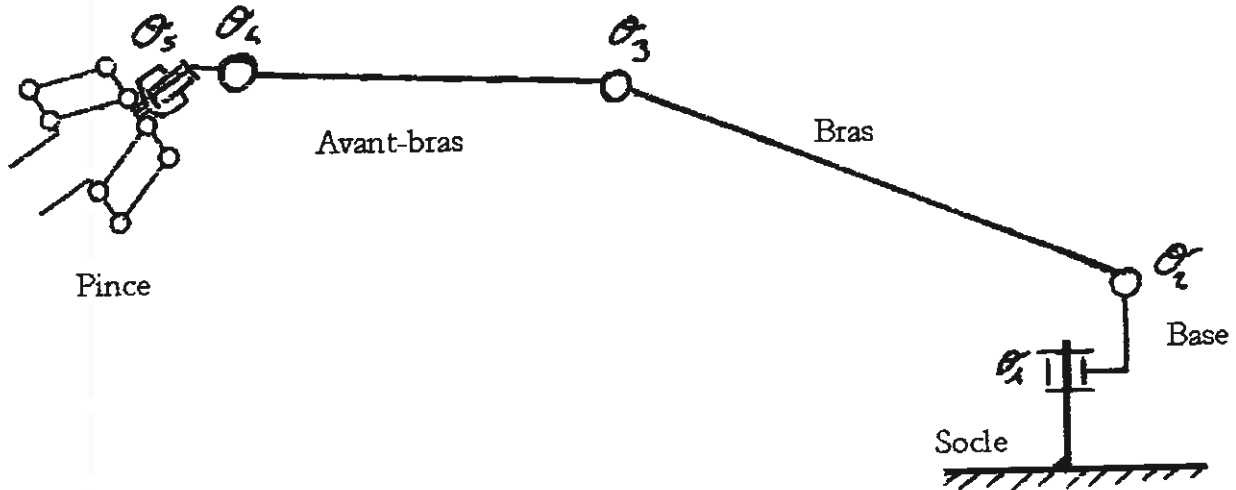
Le robot est destiné à l'environnement industriel. Il devra répondre aux exigences de ce milieu, à savoir :

- la résistance
- la fiabilité
- la sécurité
- la précision

Pour mettre notre système dans les conditions, nous l'avons installé auprès d'un tapis roulant. Afin qu'il puisse marcher automatiquement, un capteur est installé et détecte la présence de pièce à saisir. Un ordinateur se charge de la commande.

2) Cinématique

a) Vue d'ensemble du robot



Le robot a 5 axes montés en liaisons pivot. Le socle est fixé au support, lui-même solidaire de la machine avec le tapis roulant.

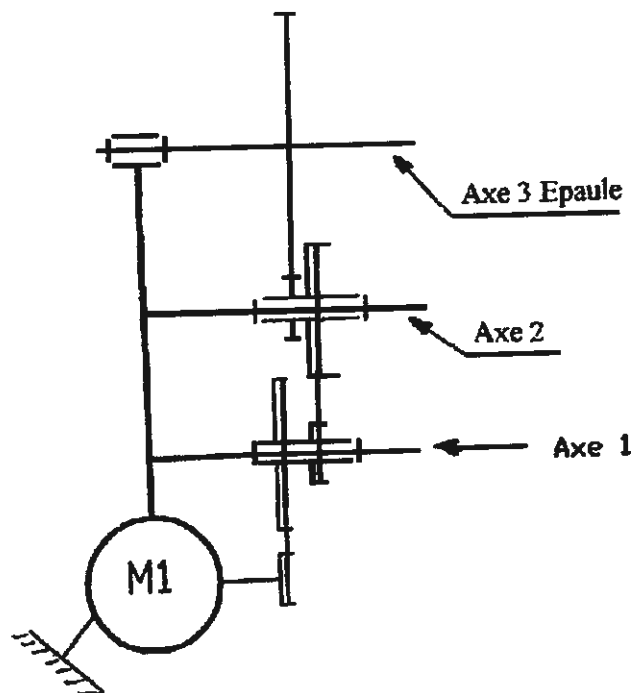
b) Les mouvements simples de rotation (ne nécessitant qu'un moteur)

→ La base :

La liaison pivot σ_1 est réalisée par un système poulie/courroie simple.

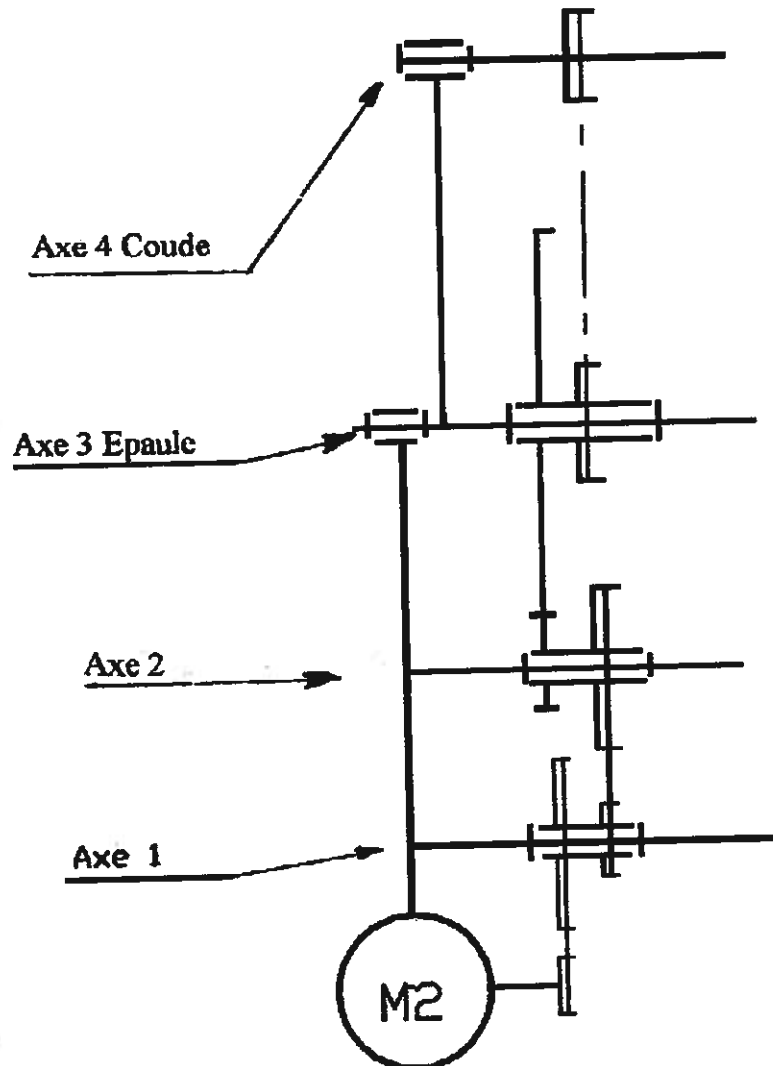
→ Rotation autour de l'axe de l'épaule (liaison σ_2) :

Grâce à une rotation en σ_2 , le bras peut s'abaisser ou remonter. Seul le moteur M1 intervient pour faire ce mouvement.



Nous trouvons une transmission de type poulie/courroie, avec aussi un petit pignon qui entraîne une roue. On peut remarquer que le sens de rotation de la roue fixée sur l'axe 3 est l'inverse du sens à la sortie du moteur. Cet inversement est dû au pignon et la roue, les transmissions poulie/courroie conservant le sens de rotation. Ces deux système de transmission servent à découpler l'effort : on augmente ainsi le couple, et on réduit la vitesse.

→ Rotation en σ_3 :

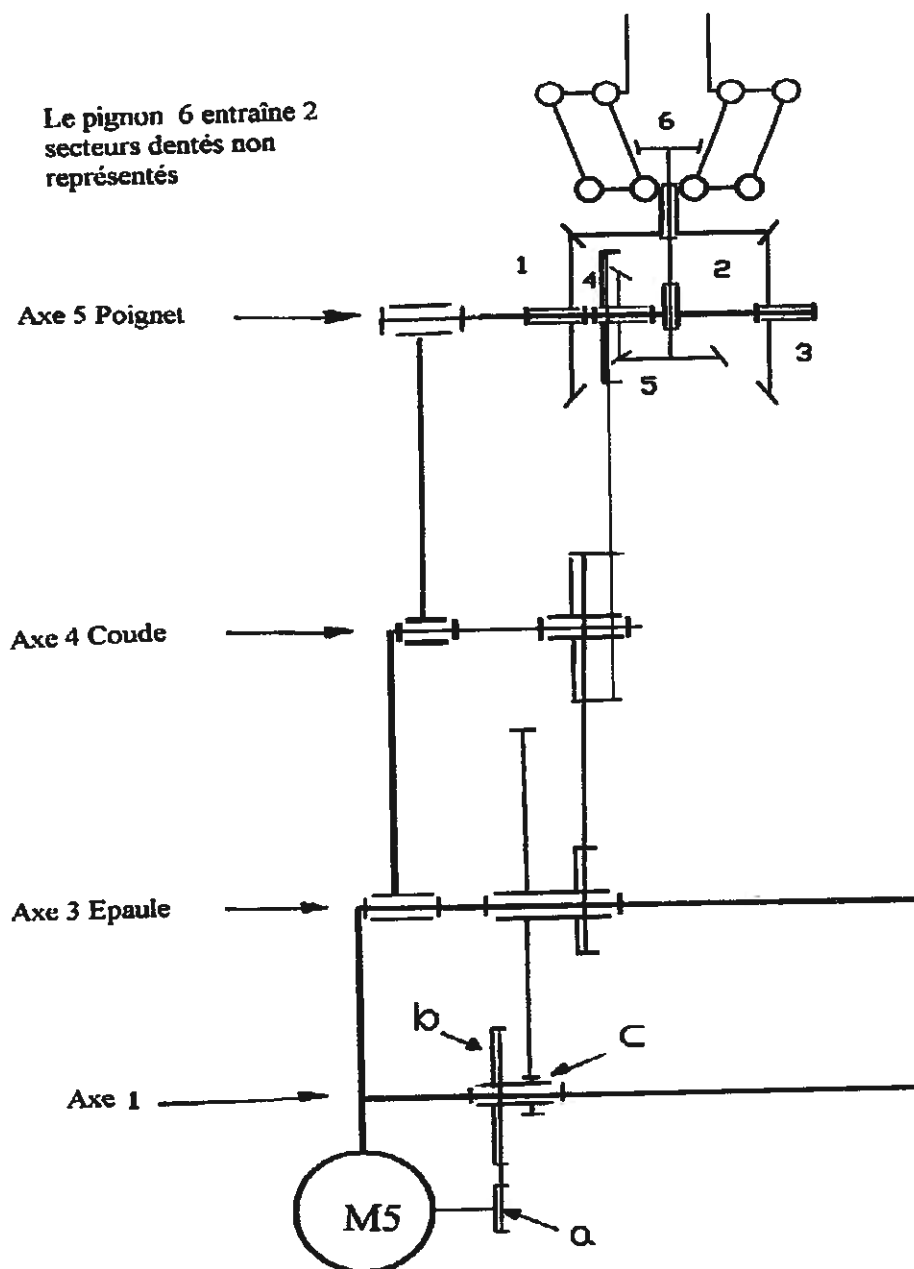


Même commentaire que précédemment (il y a aussi une inversion de sens). Lorsque l'on fait tourner le moteur M2, on fait s'abaisser ou se lever l'avant-bras. Les efforts pouvant être assez conséquents (il s'agit de monter par exemple un objet lourd), une chaîne a été mise à la place de la courroie pour la liaison de l'axe 3 à l'axe 4.

c) Les mouvements nécessitant plusieurs moteurs :

On peut remarquer la présence de 6 moteurs installés dans le robot. En effet, il faut aussi un moteur pour l'ouverture de la pince, en plus des 5 autres pour les pivots σ_1 à σ_5 (voir schéma du début).

Le robot est doté d'un différentiel à la pince. Mais il rend les mouvements plus compliqués à produire car on doit pour une simple rotation mettre en marche 2 voire 3 moteurs différents.



La rotation du moteur M5 permet l'ouverture de la pince en faisant tourner le pignon 4 qui lui-même entraîne le pignon 5. L'axe sur lequel est fixé le pignon 6 se met en rotation, le pignon 6 entraîne les deux secteurs dentés : la pince s'ouvre (ou se ferme).

Mais pour tous les autres mouvements faisant appel au différentiel, au moins deux moteurs sont utiles pour un mouvement :

➡ Rotation autour de l'axe de la pince :

Il faut provoquer la rotation simultanée et de sens contraires des pignons 1 et 3. Pour cela, les moteurs 4 et 5 interviennent. Mais il se pose alors un problème s'il y a juste ces deux moteurs : la pince tourne très bien, par contre elle s'ouvre au fur et à mesure de sa rotation.

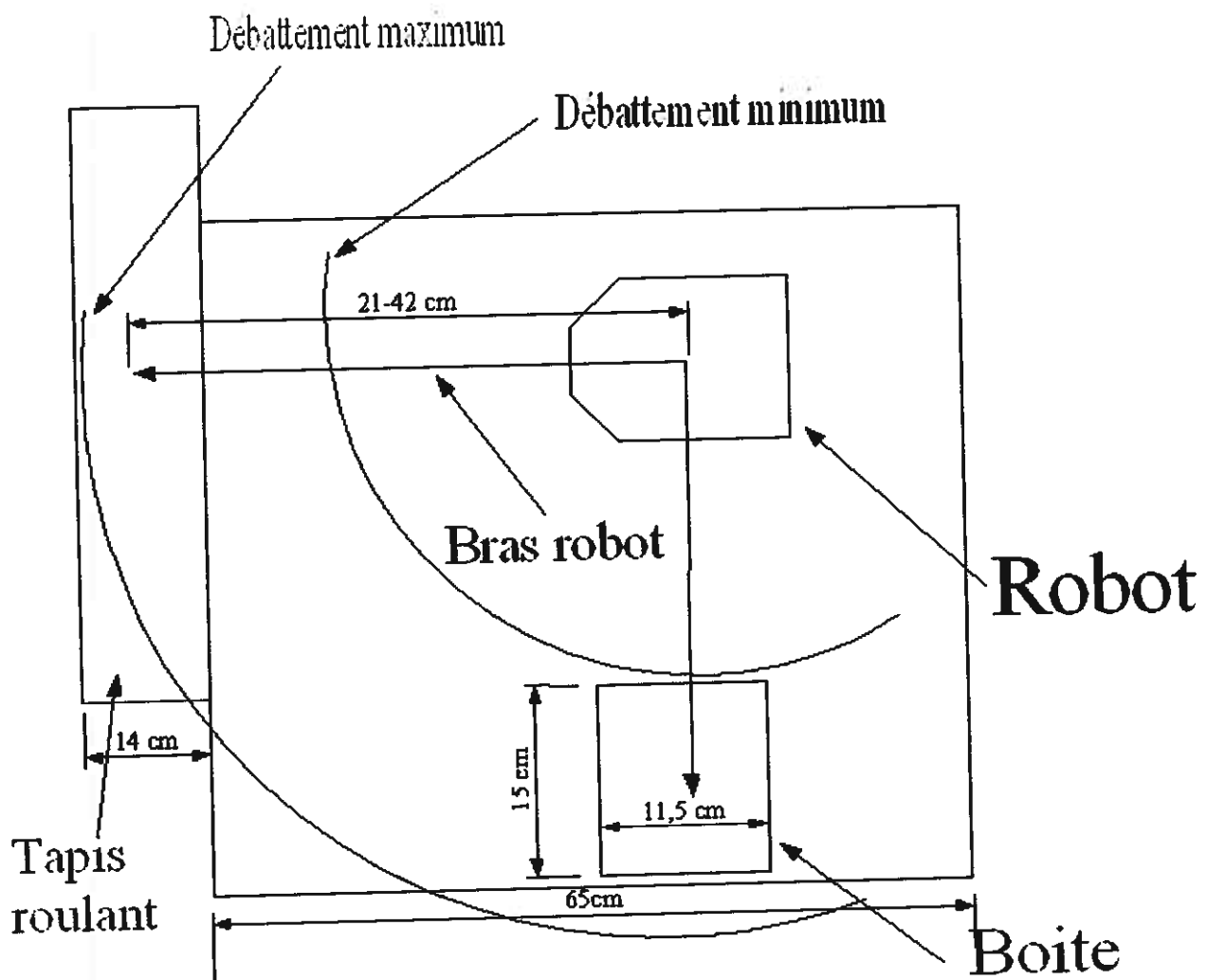
Pour résoudre ce problème, il suffit de faire tourner le moteur 5 dans le même sens que le 6

→ Rotation autour de l'axe du poignet avec la pince restante dans sa position :

Ce mouvement est possible lorsque l'on fait tourner le pignon 1 et 3, cette fois dans le même sens. On retrouve le problème identique que précédemment : la pince ne reste pas dans sa position. En effet, la pince fait une rotation autour de l'axe du poignet comme souhaitée, mais l'axe soutenant le pignon 6 va aussi s'incliner. Si on ne donne aucun ordre au moteur M5, il va resté bloqué car il sera toujours sous tension. Le pignon 4 va être alors lui aussi bloqué, et lorsque la pince va faire sa rotation au niveau du poignet, la pince va s'ouvrir. Donc la solution consiste à faire marcher en même temps le moteur M5 et dans le même sens que les deux autres qui entraînent les pignons 1 et 3.

3) Conception

a) Plan de fixation



Nous avons réalisé ce plan pour montrer les différentes contraintes auxquelles nous étions soumis. En effet, il nous a fallu calculer la taille de la planche qui fait support (le rectangle blanc sur l'image) car le positionnement du robot ne doit pas être le fruit du hasard, tout comme la place de la boîte. Le bras a un débattement minimum (21 cm) et maximum (42cm). Cela détermine la surface où il sera capable d'agir. Si l'on passe outre ces bornes, on risque de faire des dégâts sur le robot, et

peut être aussi dans la boîte de commande des moteurs si ceux-ci doivent forcer.

Le tapis roulant apporte les boîtes qui sont saisies ensuite par le bras et sont placées dans le carton . On peut en mettre dans chaque 6 boîtes, 2 rangées où sont empilées 3 boîtes. Le fait de mettre aussi peut de rangées dans un carton a pour avantage de faire travailler le bras dans ses possibilités par rapport à ses bornes de débattement. Si l'on avait mis 3 rangées, le bras aurait dû alors aller plus loin pour pouvoir mettre la dernière rangée, ce qui n'aurait pas été possible.

Il y avait aussi une autre contrainte : il s'avérait peu aisé de faire une rotation autour de l'axe de la pince, ce mouvement étant assez imprécis. Mais avec la configuration actuelle, ce n'est plus nécessaire. Le bras n'a plus qu'à se positionner au dessus de la boîte en plastique, de refermer sa pince sur un bord, de soulever la boîte, et de la lâcher au dessus du carton incliné par des cales.

b) Une autre contrainte qui déboucha sur une pièce à concevoir :

Depuis le début du TPE, nous avions des doutes sur le bon fonctionnement du moteur qui servait à faire monter ou baisser le l'avant bras du robot. Après des tests (inversion des cartes de commandes), nous avions la déception de constater que ce moteur n'était plus valide. En commander un autre aurait pris énormément de temps (on ne l'aurait pas eu pour le passage devant le jury). Ainsi, il a fallu trouver au plus vite une solution qui fut ce que nous appelons une 'atlèlle'. Nous avons été contraints de débrancher le moteur défectueux, mais l'avant bras n'était alors plus bloqué. Nous avons pris une pièce plate en métal, l'avons travaillé pour l'adapter sur le robot. Elle sert à condamner l'avant bras, tout en pouvant régler sa position après avoir dévisser une vis.

Le fait d'avoir l'avant bras inutilisable n'était pas très grave, car il servait à monter ou baisser les boîtes. Désormais, on peut encore le faire en n'agissant plus qu'avec le bras qui suffit pour notre utilisation. La hauteur du carton a été calculée pour que le bras puisse aller jsuque là pour déposer les boîtes en plastique.

B/ Etude Electrique

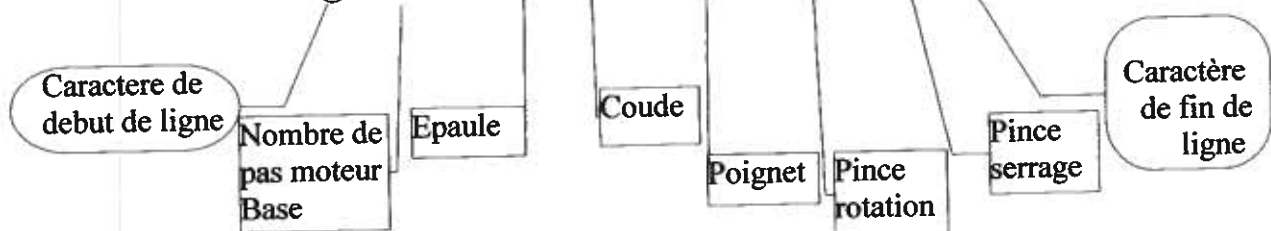
1) Fonctionnement général :

Le principe de fonctionnement est le suivant :

- 1) On envoie de l'ordinateur une ligne de commande
- 2) Cette ligne est réceptionnée et traitée par la boîte de commande
- 3) Les moteurs sont commandés

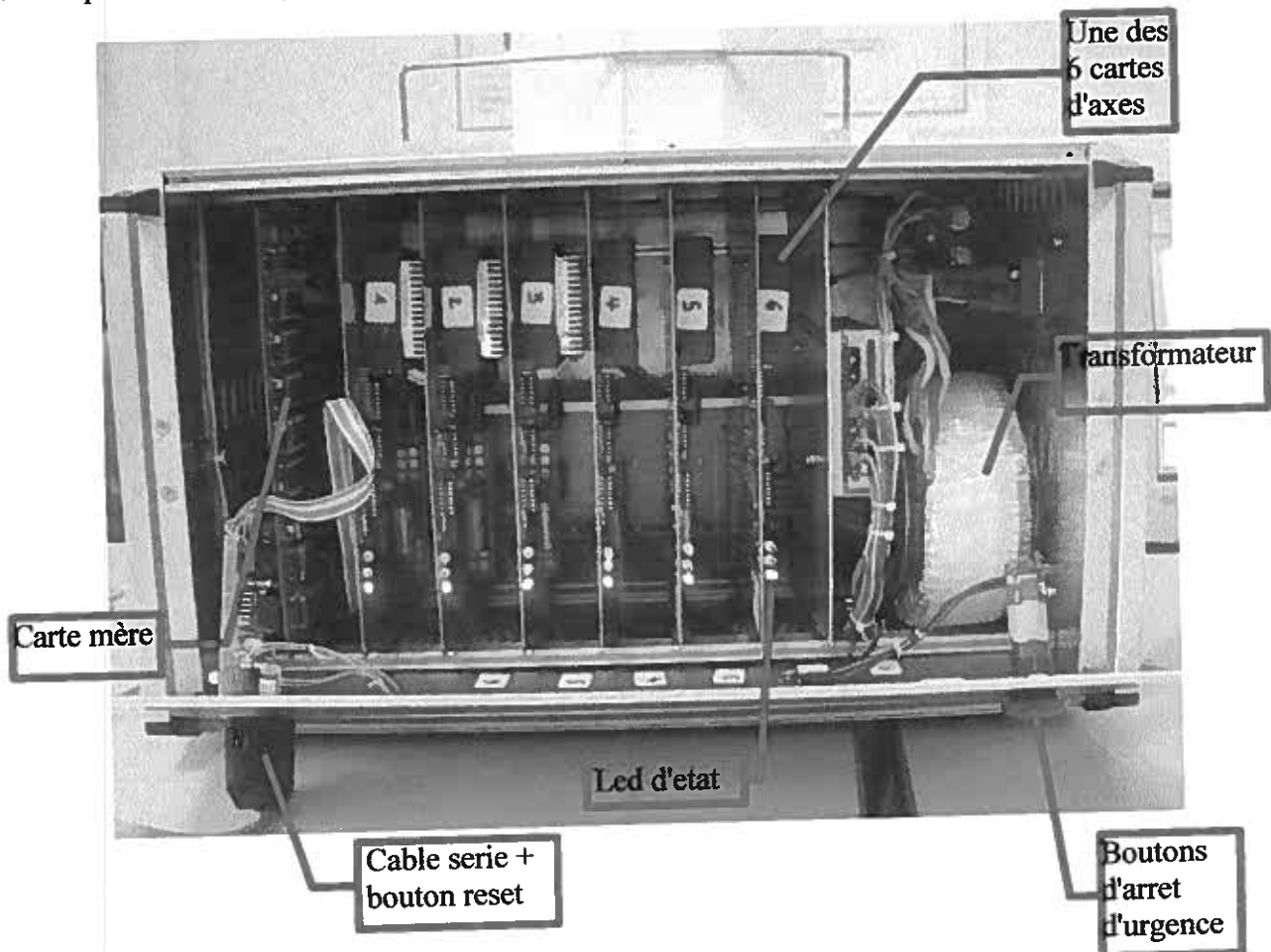
On envoie à l'aide de l'ordinateur une ligne de commande formatée de ce type :

@+00500+00000+00000B+00000+00000+00000 ?



Remarque : - Le + peut être remplacé par un – pour effectuer une rotation dans le sens inverse.
- Ici on fera tourner la base de 500 pas dans le sens positif.

Cette ligne est réceptionnée par la boîte d'alimentation et de commande qui va décomposer la ligne afin d'agir séparément sur chaque moteur. Ce boîtier permet également d'alimenter en énergie électrique les moteurs pas à pas.



2) Organisation de la la boîte :

Le câble de communication se branche sur une prise de type ancien (Sub-D 25 broches). Derrière se trouve un circuit d'adaptation de type RS232. Cette norme garanti l'universalité des signaux à envoyer. L'adaptation faite, elle se retrouve dans la carte mère qui va se charger de découper les informations. Chaque partie de la commande a son moteur, il ne reste plus qu'a envoyer via le bus (carte de fond de panier) l'information aux cartes d'axes qui vont à leur tour travailler tout ça en passant à de l'électronique de puissance : alimentation du moteur avec ses contraintes de tensions, courants et puissances à dissiper, le séquenceur qui gère la chronologie des impulsions et l'oscillateur.

3) Principe des moteurs pas à pas :

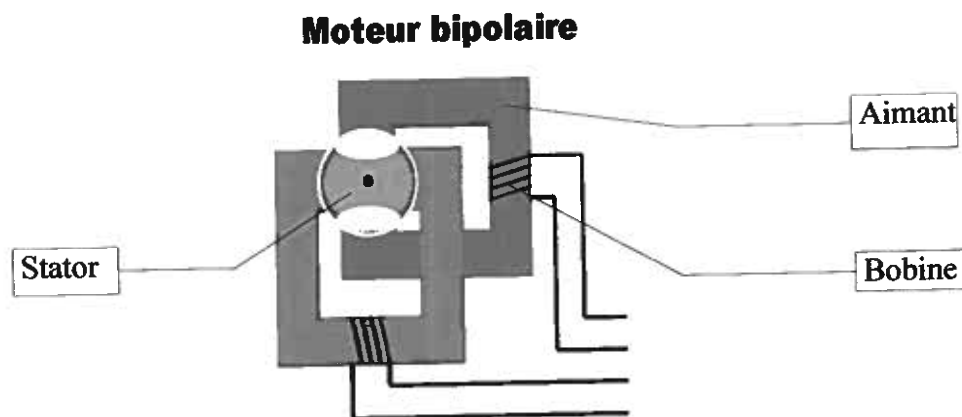
Le robot fonctionne avec 6 moteurs pas à pas. Pratiques et précis, ils permettent beaucoup, néanmoins ils sont assez lourds à gerer.

Deux types de moteurs sont utilisés pour ce robot :

- 4 Moteurs bipolaires
- 2 moteurs unipolaires, mais attention, ils sont branchés en comme des unipolaires.

Les raisons de ce choix de mettre deux moteurs differents n'est pas connu. On peut tout de même faire quelques suppositions : Il y avait sûrement besoin de moteurs plus petits (question d'intégration), peut être d'efforts moins importants, mais surtout l'électronique derrière qui n'est pas la même pour les deux types de moteurs. C'est pourquoi, on suppose, qu'ils ont été branchés comme des bipolaires.

Les moteurs pas à pas utilisé ici sont des moteurs de 100 pas, ce qui veut dire qu'il faut faire 100 pas pour faire un tour. Chaque impulsion envoyée au moteur le fait tourner d'un centième de tour. Pour les tourner, on doit alimenter les bobinages une fois dans un sens, une fois dans l'autre. On créer ainsi un pôle nord, puis un pôle sud, d'où le nom de bipolaire.



Les cartes d'axes s'occupent donc d'envoyer du courant là ou il faut, au moment voulu, tour à tour que chaque pôle, en les inversants, et le moteur tourne d'un nombre de pas bien déterminé.

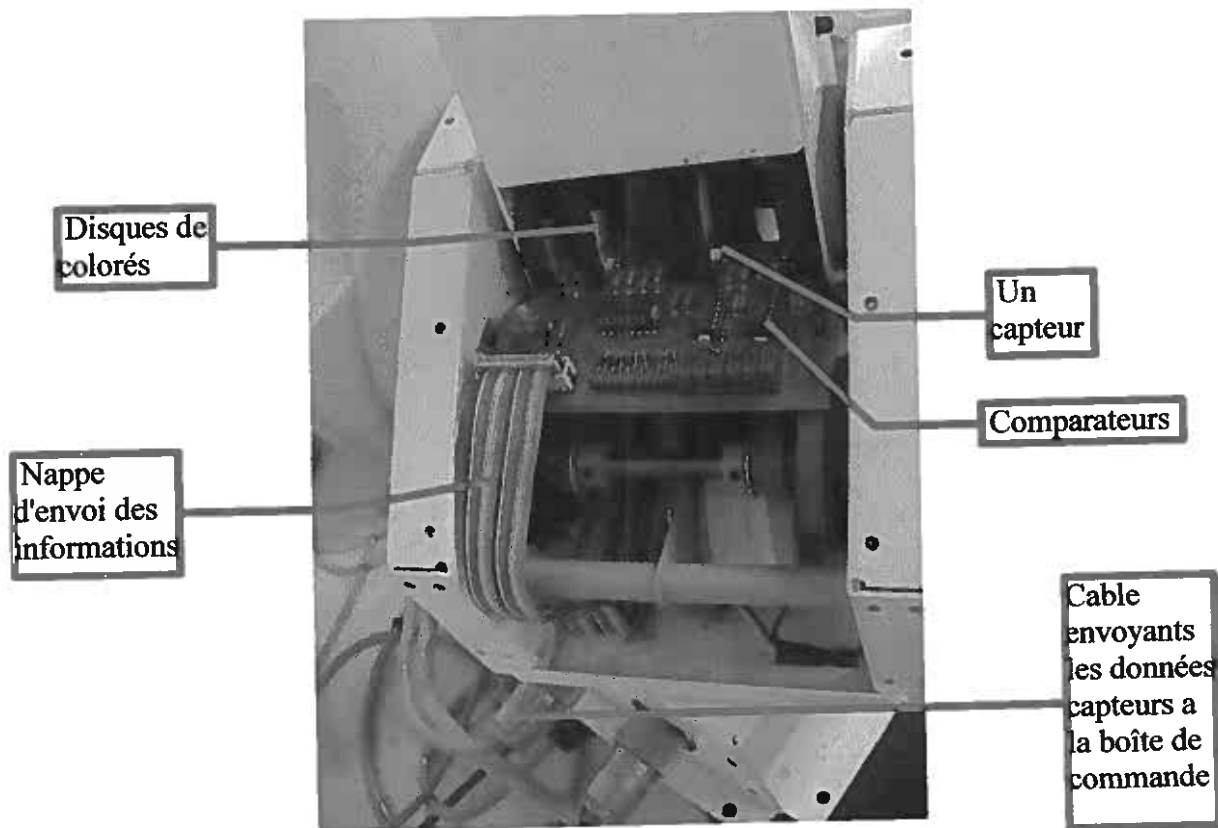
4)Table de Vérité des moteurs :

Comme nous l'avons vu dans l'étude mécanique, la présence d'un différentiel entraîne irrémédiablement des mouvements de moteurs complexes pour imposer un seul mouvement réel. Il faut souvent pour la pince, compenser un mouvement, ce qui explique la rotation de 3 moteurs pour bouger la pince.

	<i>M1</i>	<i>M2</i>	<i>M3</i>	<i>M4</i>	<i>M5</i>	<i>M6</i>
Pivot en $\sigma 1$ (base)	1	0	0	0	0	0
Pivot en $\sigma 2$	0	1	0	0	0	0
Pivot en $\sigma 3$	0	0	1	0	0	0
Biasser la pince tout en la laissant dans son ouverture initiale	0	0	0	1	1	1
Rotation (simple) de la pince en $\sigma 5$ (sens trigo)	0	0	0	1	-1	-1
Ouverture pince	0	0	0	0	1	0

5) Capteurs embarqués :

le robot possède une série de 6 capteurs qui permettront d'initialiser le robot. Il faut savoir que la partie commande fait "confiance" à la mécanique : à partir du moment où le robot est initialisé, l'envoi de pas est considéré comme effectif. Il n'y a pas de capteur de position proprement dit, seule la valeur totale des pas envoyés permet de supposer la position. Ceci est critiquable : imaginons une erreur de 2 pas lors d'une rotation de la base. Multiplié par 50, par exemple en quelques heures, l'écart devient conséquent...



Le fonctionnement est le suivant :

Pour initialiser le robot dans sa position d'origine, des cercles de couleurs ont été tracés. A la limite supérieure de la position dite '0', le disque est noir, blanc dans l'autre cas. Grâce à des capteurs détectant la lumière, il est aisé de savoir dans quel zone on se trouve. Il ne reste plus qu'à faire tourner les moteurs un par un, et les arrêter juste à la limite du blanc et du noir.

Les capteurs délivrent une tension suivant la lumière qu'ils reçoivent. Il suffit à l'aide d'un simple comparateur observer cette valeur, et ainsi savoir précisément la position. Malheureusement, le robot n'étant pas tout jeune (date de fabrication : 1988), il n'est pas rare que la détection se fasse mal.

6) Critiques des Solutions :

A travers cette étude, plusieurs points se révèlent mal choisis ou en tout cas peu commodes. Il faut bien sur tenir compte de l'âge du robot (technologie moins avancée) des coûts et utilisations.

On peut tout d'abord souligner l'imposante boîte de commande. Elle est très lourde, contient beaucoup d'éléments et n'est pas très évolutive. Ce poids est dû au fait que chaque élément est monté sur une carte spécifique, et que l'armature est en elle même lourde. On peut rajouter que la carte mère gère beaucoup de chose, peut être même trop. Ceci amène à la multiplication de composants et surtout un regard sur le bon fonctionnement peu aisé. Si une panne survient, la réparation s'annonce compliquée. Toute la gestion (notamment la décomposition pour chaque moteur) aurait facilement pu être faite logiciellement, ce qui est tout de suite beaucoup plus souple. La partie des capteurs est elle aussi plus que perfectible. On aurait très bien imaginé un système optique genre roue percée de petits trous et capteur optique pour savoir à tous moments la position réelle du robot, et éviter quelques imprécisions.

Il faut avouer que la conception du robot et des cartes est quand même à souligner : finition soignée,

carte sur rack, facilement remplaçable, mécanique robuste, ingénieux système différentiel pour la pince, intégration bien pensée... Tout ceci nous mène quand même à un certain respect vis à vis des concepteurs, étant donné les moyens et le but recherché.

C/ PROGRAMMATION

1) Ancien Programme

Un programme de gestion du robot existait déjà. Ecrit en Qbasic, il ne fonctionne que rarement et n'est pas très souple d'utilisation. De plus, étant natif DOS, il ne peut fonctionner dans un environnement Windows NT ou supérieur. En effet, l'accès direct au port COM ne peut se faire sous ces environnements. L'évolutivité de ce programme est donc limitée, il est tout simplement dépassé. L'améliorer, ou le reprendre n'est pas non plus la meilleure des solutions : le temps de comprendre le fonctionnement entier du programme (34 pages tout de même), apporter des améliorations et les valider prendrait trop de temps et ne seraient pas intéressantes. Nous avons donc décidé de créer un nouveau programme.

2) Nouveau programme

1) Présentation

L'idée lancée, il ne reste plus qu'à appliquer. Il est évident que tout est à refaire si l'on veut un résultat à la hauteur. Bon nombre d'heures ont été nécessaires au bon fonctionnement et à la mise au point de ce logiciel. Au final, on obtient un logiciel capable de gérer les fonctions principales à savoir la gestion des fonctions principales (mouvements, séquences), et beaucoup plus simple à utiliser. Vous pouvez maintenant prétendre vraiment utiliser ce robot tel qu'il a été conçu !

2) Fonctions

Voici ses fonctions :

- Initialisation du robot
- Choix du port COM
- Mise en mouvement de chaque "gestes"
- Butées configurables
- Pas global ou indépendant avec possibilité de configurer un pas pour un mouvement
- Gestion du Joystick
 - ★ Totale transparence (reprise des commandes "à la main" sans problèmes)
 - ★ Temps de réaction configurable
- Création automatique d'un fichier de logs
- Sauvergarde des logs
- Sauvegardes automatiques de tous les paramètres (butées, pas, vitesse etc)
- Choix des paramètres du robot
 - ★ Vitesse
 - ★ Accélération
 - ★ Allumage / Extinction des moteurs
- Gestion des Séquences

```

end;
if (JoyStickInfo.wButtons = 1) AND (Comport1.Connected=false) then
Showmessage('Veuillez vous connecter et initialiser le robot')

end;

// if JoyStickInfo.wButtons = 1 then bouton1.Checked:=true;
if JoyStickInfo.wButtons = 2 then begin
  if form5.Visible then Form5.Button1.Click;
end;
if JoyStickInfo.wButtons = 4 then begin
  bouton3.Checked:=true;
  if radiogroup1.ItemIndex = 0 then pincerota := trackbar1.Position
  else pincerota := Form2.TrackBar5.Position;
  if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot')
else
  if (pincerota + strtoint(label11.caption)) > strtoint(Form4.Edit10.Text) then else
    position(0,0,0,pincerota,0);
    Form3.Memo1.Lines.Add(Timetostr(time) + ' : Rotation de la Pince ' + inttostr(pincerota));
  end;
  if JoyStickInfo.wButtons = 8 then begin
    bouton4.Checked:=true;
    if radiogroup1.ItemIndex = 0 then pincerota := -trackbar1.Position
    else pincerota := -Form2.TrackBar5.Position;
    if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot')
  else
    if (pincerota + strtoint(label11.caption)) < strtoint(Form4.Edit9.Text) then else
      position(0,0,0,pincerota,0);
      Form3.Memo1.Lines.Add(Timetostr(time) + ' : Rotation de la Pince ' + inttostr(pincerota));
    end;
    if JoyStickInfo.wButtons = 16 then Button1.Click;
    if JoyStickInfo.wButtons = 32 then Button3.Click;
  if JoyStickInfo.wButtons = 64 then begin
    bouton1.Checked:=true;
    if radiogroup1.ItemIndex = 0 then poignet := trackbar1.Position
    else poignet := Form2.TrackBar4.Position;
    if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot')
  else
    if (poignet + strtoint(label10.caption)) > strtoint(Form4.Edit8.Text) then else
      position(0,0,0,poignet,0,0);
      Form3.Memo1.Lines.Add(Timetostr(time) + ' : Poignet ' + inttostr(poignet));
    end;
    if JoyStickInfo.wButtons = 128 then begin
      bouton2.Checked:=true;
      if radiogroup1.ItemIndex = 0 then poignet := -trackbar1.Position
      else poignet := -Form2.TrackBar4.Position;
      if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot')
    else
      if (poignet + strtoint(label10.caption)) < strtoint(Form4.Edit7.Text) then else
        position(0,0,0,poignet,0,0);
        Form3.Memo1.Lines.Add(Timetostr(time) + ' : Poignet ' + inttostr(poignet));
      end;
    end;
  end;
end;

```

```
end;  
end;  
end.
```

Sequence.pas

```
unit sequence;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Grids, StdCtrls, ComCtrls, ExtCtrls, inifiles;
```

```
type
```

```
TForm5 = class(TForm)
```

```
Tableau: TStringGrid;
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
GroupBox1: TGroupBox;
```

```
CheckBox1: TCheckBox;
```

```
TrackBar1: TTrackBar;
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Timer1: TTimer;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
Button3: TButton;
```

```
Label6: TLabel;
```

```
Button4: TButton;
```

```
Button5: TButton;
```

```
Button6: TButton;
```

```
SaveDialog1: TSaveDialog;
```

```
OpenDialog1: TOpenDialog;
```

```
Button7: TButton;
```

```
Button8: TButton;
```

```
procedure FormCreate(Sender: TObject);
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure Button2Click(Sender: TObject);
```

```
procedure TrackBar1Change(Sender: TObject);
```

```
procedure Timer1Timer(Sender: TObject);
```

```
procedure Button3Click(Sender: TObject);
```

```
procedure Button4Click(Sender: TObject);
```

```
procedure Button5Click(Sender: TObject);
```

```
procedure Button6Click(Sender: TObject);
```

```
procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
procedure TableauSelectCell(Sender: TObject; ACol, ARow: Integer;  
var CanSelect: Boolean);
```

```
procedure Button7Click(Sender: TObject);
```

```
procedure Button8Click(Sender: TObject);
```

```
private
```

```

    { Déclarations privées }
public
    { Déclarations publiques }
    procedure SauveSeq(Etape:string; Moteur:string; Datas:string);
        end;

var
    Form5: TForm5;
    etapecourante : integer;
    valeur : string;
    fichier : string;
implementation

uses Main, telecommande, logs;

{$R *.dfm}

procedure TForm5.FormCreate(Sender: TObject);
begin
    etapecourante := 0;
    Tableau.Cells[1,0] := 'BASE';           //Affichage dans le tableau
    Tableau.Cells[2,0] := 'EPAULE';
    Tableau.Cells[3,0] := 'BRAS';
    Tableau.Cells[4,0] := 'POIGNET';
    Tableau.Cells[5,0] := 'POIGNET ROTA.';
    Tableau.Cells[6,0] := 'PINCE SERRAGE';
    Tableau.Cells[0,0] := 'Etape N°';

    Label2.caption := inttostr(Trackbar1.Position);
    Timer1.Interval := Trackbar1.Position;

end;

procedure TForm5.Button1Click(Sender: TObject);
var
    i: integer;
    signe : string;
begin
    Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Ajout étape ' + inttostr(Tableau.RowCount - 1));
    Tableau.Tag := Tableau.RowCount;
    Tableau.RowCount := Tableau.RowCount + 1; //On rajoute une etape dans la séquence
    Tableau.Cells[0,Tableau.RowCount-1] := inttostr(Tableau.RowCount - 1);
    For i:= 1 to 6 do
        begin
            if i = 1 then signe := p;
            if i = 2 then signe := q;
            if i = 3 then signe := r;
            if i = 4 then signe := s;
            if i = 5 then signe := t;
            if i = 6 then signe := u;
            Tableau.cells[i,Tableau.RowCount-1] := signe + sub[i];
        end;
    end;
end;

```

end;

```
procedure TForm5.Button2Click(Sender: TObject);
begin
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Lecture de la séquence');
```

```
if tableau.RowCount = 1 then Showmessage('Aucune Position Enregistrée');
if not Form1.ComPort1.Connected then ShowMessage('Veuillez Initialiser le Logiciel !')
else Timer1.Enabled := True;
```

end;

```
procedure TForm5.TrackBar1Change(Sender: TObject);
begin
label2.Caption := inttostr(Trackbar1.position) + 'ms';  //Vitesse entre les etapes
Timer1.Interval := Trackbar1.Position;
end;
```

```
procedure TForm5.Timer1Timer(Sender: TObject);
```

```
begin
```

```
inc(etapecourante);
Form6.Caption := 'Télécommande - Etape N°'+ inttostr(etapecourante);
```

```
label5.Caption := inttostr(etapecourante);
Form1.position((strtoint(Form5.Tableau.Cells[1,etapecourante])-
strtoint(Form1.label7.caption)),(strtoint(Tableau.Cells[2,etapecourante])-
strtoint(Form1.label8.Caption)),(strtoint(Tableau.Cells[3,etapecourante])-
strtoint(Form1.label9.Caption)),(strtoint(Tableau.Cells[4,etapecourante])-
strtoint(Form1.label10.Caption)),(strtoint(Tableau.Cells[5,etapecourante])-
strtoint(Form1.label11.Caption)),(strtoint(Tableau.Cells[6,etapecourante])-
strtoint(Form1.label12.Caption)));  //Manipulation pour pouvoir la procedure précédente (dans le
main)
```

```
Label3.Caption := '@' + Tableau.Cells[1,etapecourante] + Tableau.Cells[2,etapecourante] +
Tableau.Cells[3,etapecourante] + 'B' + Tableau.Cells[4,etapecourante] +
Tableau.Cells[5,etapecourante] + Tableau.Cells[6,etapecourante] + '?';
```

//Suivant : gestion des enregistrements, arret reprise...

```
if etapecourante = Tableau.rowcount-1 then
begin
if checkbox1.Checked = true then etapecourante := 0 else
timer1.Enabled := false;
etapecourante := 0;
end;
end;
procedure TForm5.Button3Click(Sender: TObject);
begin
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Séquence arrêtée');
```

```
Timer1.Enabled := False;
end;
```

```
procedure TForm5.Button4Click(Sender: TObject);
begin
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Reprise de la séquence');
```

```
Timer1.Enabled := True;
end;
```

```
procedure TForm5.SauveSeq(Etape:string; Moteur:string; Datas:string);
```

```
var FIni:TiniFile;
```

```
begin
  FIni:=TiniFile.Create(fichier);                                //Procédure de Sauvegarde de la séquence
  FIni.WriteString(Etape,moteur,Datas);
  FIni.Free;
end;
```

```
procedure TForm5.Button5Click(Sender: TObject);
```

```
var
```

```
eta : integer;
```

```
begin
```

```
If SaveDialog1.Execute then Fichier:=(SaveDialog1.FileName);
```

```
SAuveSeq('Count','Nbretapes',inttostr(Tableau.RowCount - 1));
```

```
For eta := 1 to Tableau.RowCount - 1 do
```

```
begin
```

```
SauveSeq(inttostr(eta),'Base',Tableau.Cells[1,eta]);
```

//Sauvegarde de la séquence

```
SauveSeq(inttostr(eta),'Epaule',Tableau.Cells[2,eta]);
```

```
SauveSeq(inttostr(eta),'Bras',Tableau.Cells[3,eta]);
```

```
SauveSeq(inttostr(eta),'Poignet',Tableau.Cells[4,eta]);
```

```
SauveSeq(inttostr(eta),'Pincerota',Tableau.Cells[5,eta]);
```

```
SauveSeq(inttostr(eta),'Pinceserre',Tableau.Cells[6,eta]);
```

```
end;
```

```
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Séquence sauvegardée dans : ' +
Savedialog1.FileName);
```

```
end;
```

```
procedure TForm5.Button6Click(Sender: TObject);
```

```
Var
```

```
FIni: TiniFile;
```

```
j : integer;
```

```
nbretapes : integer;
```

//Chargement de la séquence

```
begin
```

```
Tableau.RowCount := 1;
```

```
if opendialog1.Execute then Fichier:= opendialog1.FileName;
```

```
FIni:=TiniFile.Create(fichier);
```

```
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Fichier Séquence ouvert : ' + fichier);
```

- ★ Ajout d'étapes
- ★ lecture pas à pas
- ★ Modification des étapes à chaud
- ★ Sauvegarde et Chargement des séquences
- ★ Bouclage des séquences
- ★ Délai entre chaque étapes
- ◆ Affichage des positions actuelles du robot
- ◆ Envoi de commandes “manuelles”
- ◆ Aide
- ◆ Heure ;-)

3)Aide

Afin de faciliter l'utilisation du logiciel, et l'apprentissage de ses fonctions, nous avons créé un fichier d'aide qui vous aidera dans votre démarche. Celui-ci relate toutes les fonctions du logiciel, tout est là pour mener à bien l'utilisation du logiciel !

Bellec Maël
Huon Gurvan
Dupont Marc
TS5
TPE 2003

AIDE

du logiciel

Boby le robot v1.0

Sommaire

1. Introduction

2. Fonctionnement général

- a) Communication avec le robot
- b) Initialisation
- c) Utilisation générale
- d) Sauvegarde des paramètres

3. Gestion Séquences

4. Fonctions Diverses

- a) Joystick
- b) Butées
- c) Logs

5. Limitations du logiciel

1. Introduction

Bienvenue dans le logiciel Bobby le robot v1.0 !

Développé tout spécialement pour le bras manipulateur de Multisoft Robotique, ce logiciel vous aidera à exploiter ses nombreuses possibilités. Du simple mouvement à la gestion de séquences en passant par le Joystick ou encore les différents paramètres, nous passerons en revue ici tout ce dont vous devez savoir.

Nous vous rappelons que tout ceci a été créé pour le projet TPE de Bellec Maël, Huon Gurvan et Marc Dupont pour l'année 2002-2003. Nous vous demandons donc une certaine tolérance vis à vis des dysfonctionnements possibles (et il y en a), ou sur l'oubli d'éléments. Sur ce, bonne lecture !

2. Fonctionnement général

a) Communication.

La communication du robot s'effectue par la liaison série de l'ordinateur. Pour relier l'ordinateur à la boîte de commande, il vous faut un câble avec à un bout une prise SubD 9 broches femelle, et à l'autre bout une prise SubD 25 broches femelle également. Ce changement de genre est dû à l'ancienneté du robot, en ce temps le port série était en 25 Broches.

Une fois connecté physiquement, et la boîte de commande sous tension, vous pouvez lancer le logiciel "Bobby le robot !". Appuyer sur le bouton "Init logiciel" sur le panneau principal pour ouvrir le port COM. Si tout se passe bien, le port COM sera ouvert sans message d'erreur, une petite ampoule s'allumera dans la barre d'état. De plus, si la connectique avec le robot est correcte, un voyant vert "Etat du robot", toujours dans la barre

des tâches, s'allumera. Tout est en ordre ! Très bien, nous allons pouvoir continuer.

Si jamais la connection ne se fait pas correctement, plusieurs sources d'erreurs peuvent être à l'origine :

- Mauvais port COM ouvert, dans ce cas, dans le menu préférences, sélectionnez le port où est effectivement connecté le robot. Vérifiez également qu'aucun logiciel ne vient utiliser le port COM, auquel cas il serait occupé, donc inutilisable.
- Mauvais branchement, si le voyant vert n'est pas allumé alors que tout semble correct, vérifiez bien les branchements.

b) Initialisation du robot.

Après avoir ouvert le port COM avec succès, vous pouvez maintenant initialiser le robot mécaniquement. Attention, c'est une phase très délicate. En effet, les capteurs parfois peu coopératifs ou une position mal placée rend difficile l'initialisation. Elle consiste à placer tout les moteurs à leurs points d'origines. Appuyez sur le bouton "Init Robot", puis observez le robot s'initialiser en se tenant prêt à appuyer sur l'arrêt d'urgence (vous constaterez sa grande utilité...). Pour favoriser cette étape, nous nous conseillons de placer tous les moteurs au plus prêt de leurs origines. En effet, ils trouveront plus facilement ainsi. Pour ce qui est de la pince, il est inscrit 'dessus' sur une partie métallique. Mettez la comme son nom l'indique au dessus, et la pince s'initialisera plus facilement. Pour le reste il faut avouer que c'est une question d'habitude, on sait à la longue s'il "est dans la bonne voie". Si vous avez un doute, arrêtez le, c'est plus prudent.

Une remarque très importante est à faire. Lorsque nous avons utilisé ce robot, nous avons dû l'adapter, à cause d'un de ces moteur défectueux. Il s'agit de l'avant bras, qui comporte une "Atèle" derrière le ressort de maintien. Elle permet donc de bloquer dans une position le robot afin qu'il ne bouge plus. Le moteur a été débranché. Pour initialiser le robot, il faut donc, quand c'est au tour de ce moteur, appuyer légèrement sur le capteur correspondant (vous trouverez facilement, j'en suis sûr) et le tour est joué !

Bien, le robot est initialisé, nous allons pouvoir commencer à l'utiliser !

c) Utilisation.

L'utilisation est très simple : pour faire bouger tel ou tel moteur il suffit d'appuyer sur le bouton correspondant. A chaque fois que vous allez appuyer, le logiciel enverra le pas. Celui ci est configurable. Soit vous le voulez global, auquel cas à l'aide du curseur réglez le en fonction de vos besoins, soit vous le voulez indépendant, vous pourrez donc choisir le nombre de pas à envoyer pour chaque moteur. Dans ce cas, dans le menu de configuration des butées vous trouverez de quoi régler tout cela. Une petite remarque, n'y aller pas trop vite, le robot risquerait de "perdre les pédales". Dans ce cas, seul un Reset peut vous sauver.

d) Sauvegarde des paramètres.

Tous vos réglages sont automatiquement sauvegardés dans un fichier INI. A chaque modification de paramètres tel que les pas, les butées, la vitesse, accélération etc .

Ainsi vous n'aurez pas à refaire vos réglages à chaque fois. Si vous changez d'ordinateur vous perdrez toutes ces informations. Pour les conserver, récupérez le fichier "params.ini" qui se situe dans le répertoire racine de Windows. Vous pouvez également modifier ces valeurs à la main pour gagner du temps. En cas d'absence de ce fichier, un nouveau sera recréé automatiquement.

3. Gestion Séquences

a) Aperçu global.

Les séquences sont des suites de positions enregistrées dans des étapes, dans lesquelles sont mémorisées chaque positions de moteur à un instant précis. Une séquence pourra donc être une application spécifique au robot tel que prendre un objet et le déplacer. Ce n'est qu'une succession d'étapes simples qui seront lues dans l'ordre pour recréer l'action. Vous pouvez Charger et Sauvegarder toute séquence, même incomplète, et les éditer directement dans le logiciel. Un fichier de séquence n'est en fait qu'un fichier INI dans lequel vous pouvez ajouter, modifier les étapes sans problèmes.

b) Enregistrement de séquence.

Vous devez d'abord être connecté au robot et avoir initialisé ce dernier. Ensuite, c'est très simple. Dans le menu Fichier, cliquez sur Nouvelle Séquence. Orientez le robot là où vous le souhaitez, et appuyez sur le bouton d'enregistrement. Vous devez voir une nouvelle étape apparaître dans le tableau. Les valeurs actuelles du robot doivent s'inscrire dans les cases. Nous vous conseillons vivement de faire des étapes SIMPLES. En effet, faire bouger plusieurs moteurs d'un coup et ainsi créer une composition complexe fonctionne, mais engendre des bruits suspects du robot. A ne pas trop abuser donc. Pour plus de commodités, utilisez la télécommande (bouton télécommande dans le gestionnaire de Séquences). Vous pouvez éditer directement les valeurs dans le tableau, si besoin est. Une fois votre séquence terminée, vous pouvez l'enregistrer dans un fichier.

c) Lecture de séquence.

Utilisez la télécommande, vous gagnerez du temps. Cliquez sur l'étape voulue et appuyez sur les boutons d'avance et recul pas à pas. Vous avancerez ainsi d'une étape à la fois, afin de bien contrôler l'avancement du robot. Ne négligez pas cette étape, elle est très importante. Décomposez bien les mouvements, préférez mettre plus d'étapes que de vouloir avancer trop vite ! Ne brûlez pas les étapes !

Si vous êtes sûr de votre séquence, réglez le délai entre chaque étape aux alentours de quelques secondes, vous éviterez d'aller plus vite que le robot. Si jamais une étape arrive alors que le robot n'a pas fini la précédente, il se perdra à tous les coups ! Une fois la séquence rodée, vous pourrez réduire ce temps au plus juste, l'expérience vous y aidera (notamment en ce qui concerne l'ouverture et fermeture de pince).

6. Fonctions Diverses

a) Joystick

Ce logiciel est capable de gérer un joystick pour faciliter la manipulation du robot. Cette fonction n'est à utiliser qu'avec prudence. La composition de moteur est très délicate, nous vous conseillons de procéder doucement. Le fonctionnement est le suivant : Après avoir installer (et étalonner !) le joystick, il vous suffit de le bouger et d'appuyer sur la gâchette pour envoyer la commande. Le robot se déplacera dans la position voulue. La rotation de la pince se fera avec les boutons 3 et 4, le mouvement du poignet avec les boutons 7 et 8. Le bouton 5 initialisera le logiciel, tandis que le bouton 6 initialisera le robot. Enfin le bouton 2 enregistrera la position courante en temps qu'étape dans une séquence. La trop grande précision du joystick face au robot crée un fossé assez grand, et il n'est pas facile de vraiment bien faire bouger le robot. Toutefois, il permet un positionnement assez ludique.

b) Butées

Vous pouvez configurer les butées de sorte que l'on ne puisse pas forcer sur la mécanique, ou par exemple dans un espace limité, éviter de toucher les obstacles. Celles-ci sont nécessairement symétriques (pour des raisons de programmation). Attention, si vous les configurez mal, vous pouvez endommager le robot. Prenez donc soin de Bobby, fabriqué en 1988, il n'est plus tout jeune !

c) Logs

Afin de déceler les bugs (plus qu'éventuels) le logiciel intègre une création d'un fichier de log, qui répertorie toutes opérations effectuées. Vous pouvez y accéder via le menu paramètres puis logs. Vous pouvez sauvegarder sous forme de fichier texte afin d'y accéder ultérieurement.

5.Limitations du logiciel

Comme tout logiciel, il contient des bugs. De plus livré dans une version presque finale, des erreurs apparaissent. Le manque de temps en est à l'origine. Dans l'ensemble, les fonctions de bases fonctionnent, pas de problèmes particuliers, des améliorations peuvent être apportés, vous trouverez les sources jointes. Encore une fois, n'allez pas trop vite avec ce robot, si jamais il se trouve réticent, donnez lui du temps ;-)
Voilà, vous savez tout (ou presque), n'hésitez pas à améliorer le logiciel, il ne demande que ça !

FIN de l'aide

4) Sources du programme

3. Pour bien exploiter les sources

Ce logiciel a été écrit sous Delphi 6 version Personnelle. En ce qui concerne le fonctionnement des sources, vous devrez installer une librairie de composant nommé ComPortLib. C'est une suite de composants gratuits, que vous trouverez sur le cd avec le dossier ou sur le site de l'auteur : <http://www2.arnes.si/~sopecrni/>
Une fois cette librairie installée, vous comprendrez aisément l'architecture du programme, dont le nom du projet est ; latotale.dpr
Voilà, de nombreuses améliorations sont possibles, libre à vous de continuer !

2.Sources commentées

Main.pas

```
{*****
*****
***** Gestion du Bras Manipulateur 'Boby' *****
*****-----Par Bellec Maël, Huon Gurvan et Dupont Marc-----*****
***** TPE ANNEE 2002-2003 *****
*****}

unit Main;
```

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, ComCtrls, ExtCtrls, StdCtrls, CPort, inifiles, CPortCtl,
Gauges, mmsystem;

type

TForm1 = class(TForm)

MainMenu1: TMainMenu;
Fichier1: TMenuItem;
N1: TMenuItem;
Quitter1: TMenuItem;
Paramtres1: TMenuItem;
Prfrences1: TMenuItem;
Aide1: TMenuItem;
Utilisationdulogiciel1: TMenuItem;
N2: TMenuItem;
Aproposde1: TMenuItem;
GroupBox1: TGroupBox;
GroupBox2: TGroupBox;
Label1: TLabel;
Label2: TLabel;
Label3: TLabel;
Label4: TLabel;

Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
GroupBox3: TGroupBox;
TrackBar1: TTrackBar;
Label18: TLabel;
Label20: TLabel;
Button1: TButton;
ComPort1: TComPort;
Button3: TButton;
Timer2: TTimer;
GroupBox4: TGroupBox;
Button5: TButton;
Button17: TButton;
Button12: TButton;
Button15: TButton;
Button11: TButton;
Button9: TButton;
Button6: TButton;
Button8: TButton;
Button10: TButton;
Button14: TButton;
Button13: TButton;
Button16: TButton;
Button18: TButton;
RadioGroup1: TRadioGroup;
Logs1: TMenuItem;
GroupBox5: TGroupBox;
ComLed1: TComLed;
ComLed2: TComLed;
Timer1: TTimer;
GroupBox6: TGroupBox;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Gauge1: TGauge;
Gauge2: TGauge;
Gauge3: TGauge;
Gauge4: TGauge;
CheckBox1: TCheckBox;
TrackBar2: TTrackBar;
Label32: TLabel;
Label33: TLabel;

Button2: TButton;
StatusBar1: TStatusBar;
ComLed3: TComLed;
ComLed4: TComLed;
ConfigurationdesButes1: TMenuItem;
NouvelleSequence1: TMenuItem;
ExcuterSequence1: TMenuItem;
Button4: TButton;
Button7: TButton;
Timer3: TTimer;
GroupBox7: TGroupBox;
boutton1: TRadioButton;
boutton2: TRadioButton;
boutton3: TRadioButton;
boutton4: TRadioButton;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;

procedure Button1Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button3Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure Button13Click(Sender: TObject);
procedure Button14Click(Sender: TObject);
procedure Button15Click(Sender: TObject);
procedure Button16Click(Sender: TObject);
procedure Button17Click(Sender: TObject);
procedure Button18Click(Sender: TObject);
procedure RadioGroup1Click(Sender: TObject);
procedure Logs1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure TrackBar2Change(Sender: TObject);
procedure ConfigurationdesButes1Click(Sender: TObject);
procedure NouvelleSequence1Click(Sender: TObject);
procedure ExcuterSequence1Click(Sender: TObject);
procedure Prfrences1Click(Sender: TObject);

```

procedure UtilisationduLogiciel1Click(Sender: TObject);
procedure AProposde1Click(Sender: TObject);
procedure ComPort1AfterOpen(Sender: TObject);
procedure Timer3Timer(Sender: TObject);

private
  { Déclarations privées }
public
  procedure validation;
  procedure position(base: integer; bras : integer; epaule : integer; poignet : integer; pincerota :
integer; pincereste : integer);
  procedure ecriture;
  procedure SauveIni(SectionIni:string; Param:string; Datas:string);
  Procedure LitIni(SectionIni:string; Param:string; default: string);
end;
type
Datastring = String[20];
var
  Form1: TForm1;           //Déclaration des Variables (nombreuses...)
  Datacomp : Datastring;
  NewCmd,Char17,temp : Boolean;
  j: integer;
  chaine : string;
  base : integer;
  epaule : integer;
  bras , poignet, pincereste, pincerota : integer ;
  base1 : string;
  pos : array[1..12] of integer;
  sub : array[1..6] of string;
  boucle,initi : integer;
  jopilou : integer;
  new : integer;
  valeur : string;
  Pasglobal : integer;
  p,q,r,s,t,u : string;
implementation

uses params, logs, butees, sequence, preferences, apropos;

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject); //Connection Port COM et prépara
begin
  //tion du robot
  Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Connection et préparation du Robot');
  If comport1.Connected = False then
  begin
    Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Ouverture du Port COM');
    //Label15.Caption := 'Ouverture du port COM1';
    Comport1.Connected := True;
    //Label15.Caption := 'Initialisation du robot';
  end;
  validation;
  base := Trackbar1.Position;

```

```
Button3.Enabled := True;
end;
```

```
Procedure TForm1.validation; //Envoi du char 15 indispensable a la comunica-
var                               //tion du robot
init : String;
```

```
Begin
```

```
for j := 0 to 3 do
begin
init := Chr(15) + #13#10;
sleep(200);
comport1.WriteStr(init);
end;
j := 0;
end ;
```

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
```

```
var                               //Juste pour savoir ce qui se passe
data: string;                     //(logs)
begin
Comport1.Readstr(data, count);
Form3.Memo1.Text := Form3.Memo1.text + data ;
If data = chr(17) then
begin
char17 := True;

end;
```

```
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
var
f:integer;                        //A l'ouverture du logiciel, bcp de
begin                             //paramètres à mettre a jour
```

```
comled1.Parent := StatusBar1;
Comled1.SetBounds(38,1, StatusBar1.Panels[0].Width, StatusBar1.Height-2);
comled2.Parent := StatusBar1;
Comled2.SetBounds(128,2, StatusBar1.Panels[1].Width, StatusBar1.Height-2);
Comled3.Parent := StatusBar1;
Comled3.SetBounds(172,2, StatusBar1.Panels[2].Width, StatusBar1.Height-2);
Comled4.Parent := StatusBar1;
Comled4.SetBounds(235,2, StatusBar1.Panels[3].Width, StatusBar1.Height-2);
```

```
Litini('PASGLOBAL', 'Pasglobal','1'); //Lecture des valeurs sauvegardés
Pasglobal:= strtoint(valeur);         //dans un fichier ini
Litini('PASGLOBAL', 'Valeur','100');
Trackbar1.Position := strtoint(valeur);
```

```

label20.Caption := inttostr(trackbar1.position);
valeur := '';
Litini('JOYSTICK', 'Activé', '0');
if valeur = '1' then checkbox1.Checked := true
else checkbox1.Checked := False;
Litini('JOYSTICK', 'Ping', '200');
Trackbar2.Position := strtoint(valeur);
Label33.Caption := valeur + ' ms';
Newcmd := False;
Char17 := False;
temp := True;
timer1.Tag := 0;
j := 0;
base := 0;
for f:= 1 to 12 do
pos[f]:=0;
pos[1] := 1;
new := 0;

```

```

end;

```

```

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Comport1.Close;    //Fermeture ;- )
end;

```

```

procedure TForm1.Button3Click(Sender: TObject);
var
f : integer;          //Initialisation du robot
begin
for f:= 1 to 12 do
pos[f]:=0;
pos[1] := 1;
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Initialisation du Robot : ');
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Envoi commande V U X S10000 J00500 K00003
R45321 F?');
chaine := '@+00000+00000+00000B+00000+00000+00000?'; //Initilialisation chaine
validation;
sleep(500);
comport1.WriteString('V U X S10000 J00500 K00003 R45321 F?' + #13#10); //Chaine d'initialisation
// Demande d'initialisation au robot //avec retour chariot
validation;
Label7.Caption := '+00000';
Label8.Caption := '+00000';    //Initialisation de l'affichage
Label9.Caption := '+00000';
Label10.Caption := '+00000';
Label11.Caption := '+00000';
Label12.Caption := '+00000';

end;

```

```

procedure TForm1.Button2Click(Sender: TObject);

```

begin

```
Form4.visible:= true;           //Chargement de toute les valeurs de
if timer1.Enabled then
begin
timer1.enabled := False;
timer3.enabled := False;
timer3.tag := 1;
end;
```

```
litini('BUTEES','Base-','-6000');    //butées
Form4.edit1.text := valeur;
litini('BUTEES','Base+', '6000');
Form4.edit2.text := valeur;
```

```
litini('BUTEES','Epaule-','-4000');
Form4.edit3.text := valeur;
litini('BUTEES','Epaule+', '4000');
Form4.edit4.text := valeur;
```

```
litini('BUTEES','Bras-','-4000');
Form4.edit5.text := valeur;
litini('BUTEES','Bras+', '4000');
Form4.edit6.text := valeur;
```

```
Form1.SauveIni('BUTEES','Poignet-','-500');
Form4.edit7.text := valeur;
Form1.SauveIni('BUTEES','Poignet+', '500');
Form4.edit8.text := valeur;
```

```
litini('BUTEES','Pincerota-','-500');
Form4.edit9.text := valeur;
litini('BUTEES','pincerota+', '500');
Form4.edit10.text := valeur;
```

```
litini('BUTEES','Pincaserre-','-500');
Form4.edit11.text := valeur;
litini('BUTEES','Pincaserre+', '500');
Form4.edit12.text := valeur;
```

end;

```
procedure TForm1.Button4Click(Sender: TObject);
begin
Comport1.WriteStr(chr(15) + #13#10);
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Envoi Char15');
end;
```

end;

```
procedure TForm1.Timer2Timer(Sender: TObject);
begin
Statusbar1.Panels.Items[3].Text := timetostr(time) ; //Affichage de l'heure ;-
end;
```

```

procedure TForm1.Button5Click(Sender: TObject);           //Suivant : toutes les procedures pour
                                                         //chaque gestes

begin
if radiogroup1.ItemIndex = 0 then base := trackbar1.Position
else base := Form2.TrackBar1.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (base + strtoint(label7.caption)) > strtoint(Form4.Edit2.Text) then showmessage('Attention,
valeur trop grande !') else                               //On gere les butées
position(base,0,0,0,0,0); //Procedure pour faire bouger un moteur du pas
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Rotation de +' + inttostr(base)); //Affichage dans les
end;                                                         //logs

```

```

procedure TForm1.TrackBar1Change(Sender: TObject);
begin
label20.Caption:= inttostr(trackbar1.position);
sauveini('PASGLOBAL','Valeur',inttostr(Trackbar1.Position));
end;

```

```

procedure TForm1.Button6Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then base := -trackbar1.Position
else base := -Form2.TrackBar1.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (strtoint(label7.caption) + base) < strtoint(Form4.Edit1.Text) then showmessage('Attention,
valeur trop grande !') else
position(base,0,0,0,0,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Rotation de -' + inttostr(base));
end;

```

```

procedure TForm1.Button7Click(Sender: TObject);
begin
validation;
Form3.Memo1.Lines.Add(TimetoStr(Time) + ' : Initialisation de la Commande');

end;

```

```

procedure tform1.position (base: integer; bras : integer; epaule : integer; poignet : integer; pincerota
: integer; pincaserre : integer);
var
moteur : integer;
                                //Etape très importante : Formatage des données
i : integer;
signe : string;
begin
jopilou := 1;
moteur := 0;
for jopilou := 1 to 12 do
begin
if jopilou mod 2 <> 0 then begin //tests valeurs impaire

```

```

if jopilou = 1 then moteur := base; // cas moteur 1
if jopilou = 3 then moteur := bras; // cas moteur 2
if jopilou = 5 then moteur := epaule; // cas moteur 3
if jopilou = 7 then moteur := poignet; // cas moteur 5
if jopilou = 9 then moteur := pincerota; // cas moteur 7
if jopilou = 11 then moteur := pincesserre; // cas moteur 9

```

```

if moteur < 0 then // gestion des des signes et valeurs
begin
if pos[jopilou] = 0 then begin
new := - pos[(jopilou + 1)] + moteur;
end
else
new := pos[jopilou+1] + moteur; //manipulation complexes mais efficaces...
if new < 0 then
begin
pos[jopilou] := 0;
new := abs(new);
end;
pos[jopilou+1] := new;

```

```

end
Else begin
if pos[jopilou] = 0 then //gestion des signes
begin
new := - pos[jopilou+1] + moteur ;
if new < 0
then new := abs(new)
else pos[jopilou] := 1;
pos[jopilou+1] := new;
end
else begin
new := pos[jopilou+1] + moteur;
pos[jopilou] := 1;
pos[jopilou+1] := new
end;

```

```

end;
end;
end;
ecriture;

```

```

For i:= 1 to 6 do
begin
if i = 1 then begin
signe := p; //Affichage...
Label7.caption := signe + sub[i];
end;

```

```

    if i = 2 then begin
    signe := q;
    Label8.caption := signe + sub[i];
    end;
    if i = 3 then begin
    signe := r;
    Label9.caption := signe + sub[i];
    end;
    if i = 4 then begin
    signe := s;
    Label10.caption := signe + sub[i];
    end;
    if i = 5 then begin
    signe := t;
    Label11.caption := signe + sub[i];
    end;
    if i = 6 then begin
    signe := u;
    Label12.caption := signe + sub[i];
    end;
end;
end;
end;

```

//gestion avec des tableaux de type
// array

```

procedure tform1.ecriture;
var

```

```

z : integer;
val : string;
begin

```

```

for z := 1 to 11 do
begin
for initi := 1 to 6 do
sub[initi] := '0';

```

```

if (z = 1) and (pos[z] = 1) then p := '+'; //Gestion des signes
if (z = 1) and (pos[z] = 0) then p := '-';

```

```

if (z = 3) and (pos[z] = 1) then q := '+';
if (z = 3) and (pos[z] = 0) then q := '-';

```

```

if (z = 5) and (pos[z] = 1) then r := '+';
if (z = 5) and (pos[z] = 0) then r := '-';

```

```

if (z = 7) and (pos[z] = 1) then s := '+';
if (z = 7) and (pos[z] = 0) then s := '-';

```

```

if (z = 9) and (pos[z] = 1) then t := '+';
if (z = 9) and (pos[z] = 0) then t := '-';

```

```

if (z = 11) and (pos[z] = 1) then u := '+';
if (z = 11) and (pos[z] = 0) then u := '-';
end;

```

```
for boucle := 1 to 12 do
begin
if boucle mod 2 = 0 then begin
```

```
if pos[boucle] <= 9999 then sub[(boucle div 2)] := '0'+inttostr(pos[boucle]);
if pos[boucle] <= 999 then sub[(boucle div 2)] := '0'+sub[(boucle div 2)];
if pos[boucle] <= 99 then sub[(boucle div 2)] := '0'+sub[(boucle div 2)];
if pos[boucle] <= 9 then sub[(boucle div 2)] := '0'+sub[(boucle div 2)];
end;    //petite astuce pour toujours arrivé a une position avec 5 chiffres
```

```
end;
```

```
val := '@'+ p + sub[1] + q + sub[2] + r + sub[3] + 'B' + s + sub[4] + t + sub[5] + u + sub[6] + '?';
//Chaine reconstituée
//Envoi de la chaine reconstituée
comport1.WriteString(val);
```

```
end;
procedure TForm1.Button8Click(Sender: TObject); //Suite de actions pour chaque
begin
    //moteurs
if radiogroup1.ItemIndex = 0 then epaule := -trackbar1.Position
else epaule := -Form2.TrackBar2.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (epaule + strtoint(label8.caption)) < strtoint(Form4.Edit3.Text) then showmessage('Attention,
valeur trop grande !') else
position(0,epaule,0,0,0,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' Epaule ' + inttostr(epaule));
end;
```

```
procedure TForm1.Button10Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then bras := trackbar1.Position
else bras := Form2.TrackBar3.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (bras + strtoint(label9.caption)) > strtoint(Form4.Edit6.Text) then showmessage('Attention, valeur
trop grande !') else
position(0,0,bras,0,0,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Bras +' + inttostr(bras));
end;
```

```
procedure TForm1.Button9Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then epaule := trackbar1.Position
else epaule := Form2.TrackBar2.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (epaule + strtoint(label8.caption)) > strtoint(Form4.Edit4.Text) then showmessage('Attention,
valeur trop grande !') else
```

```
position(0,epaule,0,0,0,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Epaule +' + inttostr(epaule));
end;
```

```

procedure TForm1.Button11Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then bras := -trackbar1.Position
else bras := -Form2.TrackBar3.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (bras + strtoint(label9.caption)) < strtoint(Form4.Edit5.Text) then showmessage('Attention, valeur
trop grande !') else
position(0,0,bras,0,0,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Bras ' + inttostr(bras));
end;

```

```

procedure TForm1.Button12Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then pincerota := -trackbar1.Position
else pincerota := -Form2.TrackBar5.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (pincerota + strtoint(label11.caption)) < strtoint(Form4.Edit9.Text) then showmessage('Attention,
valeur trop grande !') else

position(0,0,0,0,pincerota,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Rotation de la Pince ' + inttostr(pincerota));
end;

```

```

procedure TForm1.Button13Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then pincerota := trackbar1.Position
else pincerota := Form2.TrackBar5.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (pincerota + strtoint(label11.caption)) > strtoint(Form4.Edit10.Text) then
showmessage('Attention, valeur trop grande !') else
position(0,0,0,0,pincerota,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Rotation de la Pince +' + inttostr(pincerota));
end;

```

```

procedure TForm1.Button14Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then poignet := trackbar1.Position
else poignet := Form2.TrackBar4.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (poignet + strtoint(label10.caption)) > strtoint(Form4.Edit8.Text) then showmessage('Attention,
valeur trop grande !') else
position(0,0,0,poignet,0,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Poignet +' + inttostr(poignet));
end;

```

```

procedure TForm1.Button15Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then poignet := -trackbar1.Position
else poignet := -Form2.TrackBar4.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (poignet + strtoint(label10.caption)) < strtoint(Form4.Edit7.Text) then showmessage('Attention,
valeur trop grande !') else

```

```

position(0,0,0,poignet,0,0);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Poignet ' + inttostr(poignet));
end;

```

```

procedure TForm1.Button16Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then pincaserre := trackbar1.Position
else pincaserre := Form2.TrackBar6.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (pincaserre + strtoint(label12.caption)) > strtoint(Form4.Edit12.Text) then
showmessage('Attention, valeur trop grande !') else

```

```

position(0,0,0,0,0,pincaserre);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Pince ' + inttostr(pincaserre));
end;

```

```

procedure TForm1.Button17Click(Sender: TObject);
begin
if radiogroup1.ItemIndex = 0 then pincaserre := -trackbar1.Position
else pincaserre := -Form2.TrackBar6.Position;
if not Comport1.connected then Showmessage('Veuillez vous connecter et initialiser le robot') else
if (pincaserre + strtoint(label12.caption)) < strtoint(Form4.Edit11.Text) then
showmessage('Attention, valeur trop grande !') else

```

```

position(0,0,0,0,0,pincaserre);
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Pince +' + inttostr(pincaserre));
end;

```

```

procedure TForm1.SauveIni(SectionIni:string; Param:string; Datas:string);

```

```

var FIni:TiniFile;           //Automatisation de la sauvegarde
                             //dans un fichier ini

```

```

begin
  FIni:=TiniFile.Create('params.ini');
  FIni.WriteString(SectionIni,Param,Datas);
  FIni.Free;
end;
Procedure TForm1.litIni(SectionIni:string; Param:string; default: string);
Var FIni: TiniFile;

```

```

begin
  FIni:=TiniFile.Create('params.ini');      //Automatisation de la lecture
  Valeur := FIni.ReadString(Sectionini,Param,"");
  if valeur = " then
  begin
    FIni.WriteString(SectionIni,Param,default);
    Valeur := FIni.ReadString(Sectionini,Param,"");
  end;

```

```

  FIni.Free;
end;

```

```

procedure TForm1.Button18Click(Sender: TObject);

begin
Form2.visible := True;           //Lecture des valeurs précédentes

Form1.litIni('PAS','1','500');
Form2.TrackBar1.Position:= strtoint(valeur);
Form2.label7.caption := valeur;

Form1.litIni('PAS','2','500');
Form2.TrackBar2.Position:= strtoint(valeur);
Form2.label8.caption := valeur;

Form1.litIni('PAS','3','500');
Form2.TrackBar3.Position:= strtoint(valeur);
Form2.label9.caption := valeur;

Form1.litIni('PAS','4','100');
Form2.TrackBar4.Position:= strtoint(valeur);
Form2.label10.caption := valeur;

Form1.litIni('PAS','5','100');
Form2.TrackBar5.Position:= strtoint(valeur);
Form2.label11.caption := valeur;

Form1.litIni('PAS','6','100');
Form2.TrackBar6.Position:= strtoint(valeur);
Form2.label12.caption := valeur;

end;


procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
case radiogroup1.ItemIndex of    //Gestion des pas (global/individuels)
0:
begin
PasGlobal := 1;
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Le Pas est maintenant Global');
end;
1:
begin
PasGlobal := 0;
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Les Pas sont maintenant indépendants');
end;
end;
sauveini('PASGLOBAL','Pasglobal',inttostr(pasglobal));
sauveini('PASGLOBAL','Valeur', inttostr(Trackbar1.position));
end;
procedure TForm1.Logs1Click(Sender: TObject);
begin

```

```
Form3.visible:= True;
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);
```

```
var JoyStickInfo : JOYINFOEX;      //Gestion du joystick
    Result:MMRESULT;
```

```
begin
```

```
    JoyStickInfo.dwSize:=SizeOf(JoyStickInfo);
    JoyStickInfo.dwFlags:= JOY_RETURNALL;
    Result := joyGetPosEx(JOYSTICKID1, @JoyStickInfo);
    case Result of
```

```
        MMSYSERR_NODRIVER   : ShowMessage('pas de driver');
        MMSYSERR_INVALIDPARAM : ShowMessage('paramètre invalide');
        MMSYSERR_BADDEVICEID : ShowMessage('Mauvais identificateur de joystick');
        JOYERR_UNPLUGGED    : ShowMessage('Joystick non connecté');
    end;
```

```
    case Radiogroup1.ItemIndex of
        0:
            begin
```

```
                // temp:=((abs(JoyStickInfo.wXpos-32767) * (strtoint(Form4.Edit2.text)))div 32767);
                {bras} Label27.Caption := inttostr(strtoint(Form4.Edit6.text)-((JoyStickInfo.wXpos *
                strtoint(Form4.Edit6.text))div 32767));
                {epaule} Label28.Caption:= inttostr(strtoint(Form4.Edit4.text)-((JoyStickInfo.wYpos *
                strtoint(Form4.Edit4.text))div 32767));
                {pincesserre} Label29.Caption:= inttostr(strtoint(Form4.Edit12.text)-((JoyStickInfo.wZpos *
                strtoint(Form4.Edit12.text))div 32767));
                {base} label31.Caption:= inttostr(strtoint(Form4.Edit2.text)-((JoyStickInfo.dwRpos *
                strtoint(Form4.Edit2.text))div 32767));
                gauge1.Progress := 100- (JoyStickInfo.dwRpos*100)div 65535 ;
                gauge2.Progress := 100 -(JoyStickInfo.wZpos*100)div 65535 ;
                gauge3.Progress := 100 -(JoyStickInfo.wYpos*100)div 65535 ;
                gauge4.Progress := (JoyStickInfo.wXpos*100)div 65535 ;
```

//Conversion des valeurs et adaptations pour aller dircetement aux butées

```
if JoyStickInfo.wButtons = 0 then
begin
```

```
    bouton1.Checked:=false;
    bouton2.Checked:=false;
    bouton3.Checked:=false;
    bouton4.Checked:=false;
```

```
end;
```

```

end;
end;
end;
procedure TForm1.CheckBox1Click(Sender: TObject);
begin
if checkbox1.Checked = true then
begin
Sauveini('JOYSTICK', 'Activé', '1');
Groupbox6.Visible := true;
Groupbox7.visible := true;
//Form3.Memo1.Lines.Add(Timetostr(time) + ' : Joystick Activé')
Timer1.Enabled := True;
Timer3.Enabled := True;
end
else begin
Sauveini('JOYSTICK', 'Activé', '0');
Groupbox6.Visible := False;
Groupbox7.visible := False;
Form3.Memo1.Lines.Add(Timetostr(time) + ' : Joystick Désactivé');
Timer1.Enabled := false;
Timer3.Enabled := false;
end;
end;
end;

```

```

procedure TForm1.TrackBar2Change(Sender: TObject);
begin
Sauveini('JOYSTICK','Ping', inttostr(Trackbar2.Position));
Timer3.Interval := Trackbar2.Position;
Label33.Caption := inttostr(Trackbar2.position) + ' ms';
end;

```

```

procedure TForm1.ConfigurationdesButes1Click(Sender: TObject);
begin
button2.OnClick(Sender);
end;

```

```

procedure TForm1.NouvelleSequence1Click(Sender: TObject);
begin
Form5.visible:=true;
Form5.Tableau.RowCount := 1;
end;

```

```

procedure TForm1.ExcuterSequence1Click(Sender: TObject);
begin
Form5.Visible := True;
Form5.Tableau.RowCount := 1;
Form5.Button6.Click;
end;

```

```

procedure TForm1.Prferences1Click(Sender: TObject);
begin
Form7.visible := true;

end;

```

```

procedure TForm1.Utilisationdulogiciel1Click(Sender: TObject);
begin
Showmessage('Veuillez vous référer à la documentation fournie dans le dossier accompagnant le
robot.');
```

```

end;

procedure TForm1.Aproposde1Click(Sender: TObject);
begin
Form8.visible := true;
end;

```

```

procedure TForm1.ComPort1AfterOpen(Sender: TObject); //Gestion du port COM
begin
//(affichage...
If Comled1.State = IsOn then Form3.Memo1.Lines.Add(TimetoStr(Time) +' : Port ouvert !');
```

```

If Comled2.State = IsOn then Form3.Memo1.Lines.Add(TimetoStr(Time) +' : Robot physiquement
connecté !');
```

```

end;

```

```

procedure TForm1.Timer3Timer(Sender: TObject);
var JoyStickInfo : JOYINFOEX;           //Gestion du joystick
    Result:MMRESULT;
```

```

begin

```

```

JoyStickInfo.dwSize:=SizeOf(JoyStickInfo);
JoyStickInfo.dwFlags:= JOY_RETURNALL;           //2eme procedure de gestion
Result := joyGetPosEx(JOYSTICKID1, @JoyStickInfo); //des envois des positions
case Result of

```

```

MMSYSERR_NODRIVER   : ShowMessage('pas de driver');
MMSYSERR_INVALIDPARAM : ShowMessage('paramètre invalide');
MMSYSERR_BADDEVICEID : ShowMessage('Mauvais identificateur de joystick');
JOYERR_UNPLUGGED    : ShowMessage('Joystick non connecté');
end;

```

```

if Comport1.Connected then begin
if (JoyStickInfo.wButtons = 1) then begin
position((strtoint(label31.Caption))-strtoint(label7.caption),(strtoint(label28.Caption)-
strtoint(label8.Caption)),(strtoint(label27.Caption)-
strtoint(label9.Caption)),0,0,(strtoint(label29.caption)-strtoint(label12.Caption)));
Form3.Memo1.Lines.Add(Timetostr(time) +' : Envoi Cmd (Joystick) '+
label31.Caption+', '+label27.Caption+', '+label28.Caption+',0,0,'+label29.caption);

```

```

nbretapes := strtoint(FIni.ReadString('Count','nbretapes','20'));
Tableau.Tag := nbretapes;
For j := 1 to nbretapes do
begin
if FIni.SectionExists(inttostr(j)) then
begin
Tableau.RowCount := Tableau.RowCount + 1;
Tableau.Cells[0,j]:= inttostr(j);
Tableau.Cells[1,j]:= FIni.ReadString(inttostr(j),'Base','');
Tableau.Cells[2,j]:= FIni.ReadString(inttostr(j),'Epaule','');
Tableau.Cells[3,j]:= FIni.ReadString(inttostr(j),'Bras','');
Tableau.Cells[4,j]:= FIni.ReadString(inttostr(j),'Poignet','');
Tableau.Cells[5,j]:= FIni.ReadString(inttostr(j),'Pincerota','');
Tableau.Cells[6,j]:= FIni.ReadString(inttostr(j),'Pinceserre','');
end;
end;

Fini.Free;
end;

procedure TForm5.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Timer1.Enabled := False;
end;

procedure TForm5.TableauSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);
begin
Label5.Caption := inttostr(ARow);
etapecourante := ARow;
end;

procedure TForm5.Button7Click(Sender: TObject);
begin
Label3.Caption := '@' + Tableau.Cells[1,etapecourante] + Tableau.Cells[2,etapecourante] +
Tableau.Cells[3,etapecourante] + 'B' + Tableau.Cells[4,etapecourante] +
Tableau.Cells[5,etapecourante] + Tableau.Cells[6,etapecourante] + '?';
Form1.ComPort1.WriteStr(label3.caption);
end;

procedure TForm5.Button8Click(Sender: TObject);
begin
Form6.visible := true;
Form3.Memo1.Lines.Add(TimetoStr(Time) +' : Télécommande ouverte');

end;

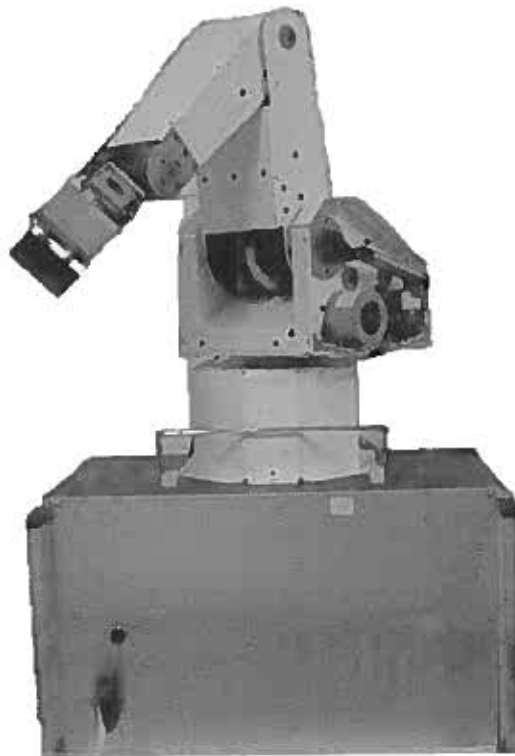
end.

```

FIN des Sources et du Dossier

Bobby le robot

Le bras manipulateur



Laurent Fabien
Paranthöen Manuel
Salomon Loeiz
Couzigou Stéphane

1^{ère} S7
Année 2003-04

Sommaire :

I. Introduction

II. Etude mécanique et conception

Analyse fonctionnelle
Diagramme pieuvre du système
Cinématique du robot
Remise en état du robot

III. Etude électrique

Principe de fonctionnement
Les capteurs
Le choix des moteurs Pas à Pas
Les liaisons séries

IV. Difficultés rencontrées

Communication PC-ROBOT : Port Séries
Initialisation du bras manipulateur
Les mouvements de la pince

V. Conclusion

I. Introduction

Le but de notre TPE était de rendre le bras manipulateur parfaitement opératif c'est-à-dire :

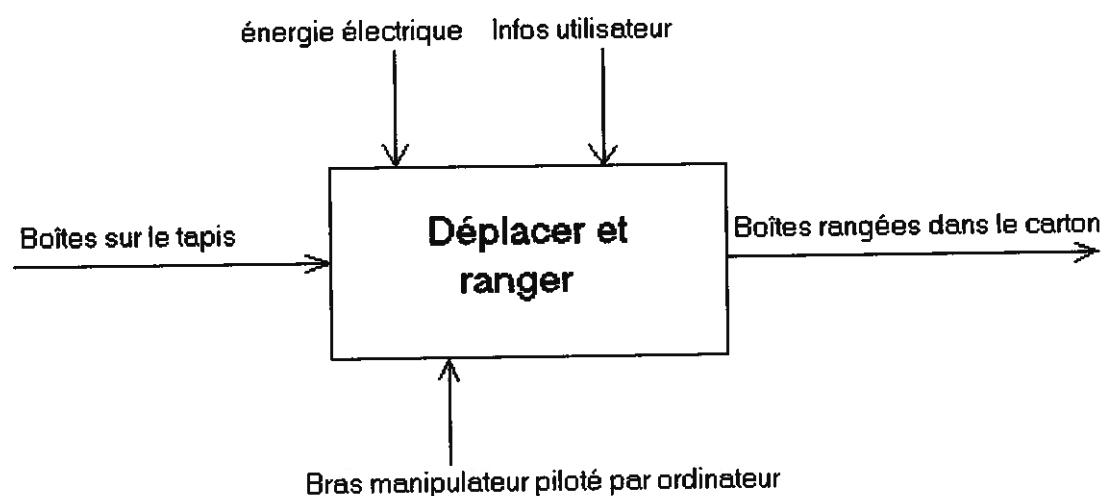
- Rendre le robot programmable par l'utilisateur.
- Création de boucle.

Pour cela nous avons effectué des études approfondies sur le fonctionnement du robot. Cependant au cours de nos recherches nous avons dues faire face à divers problèmes inattendues (liaisons entre PC et boîtier de commande, composant défaillant, bug du programme etc...). Cette perte de temps nous a considérablement aidé à comprendre et à mieux appréhender le système et son fonctionnement. Nous avons néanmoins réussi à rendre le bras manipulateur à nouveau opérationnel (création d'une séquence automatisée) malgré quelques bugs qui subsistent.

II. Etude mécanique et conception

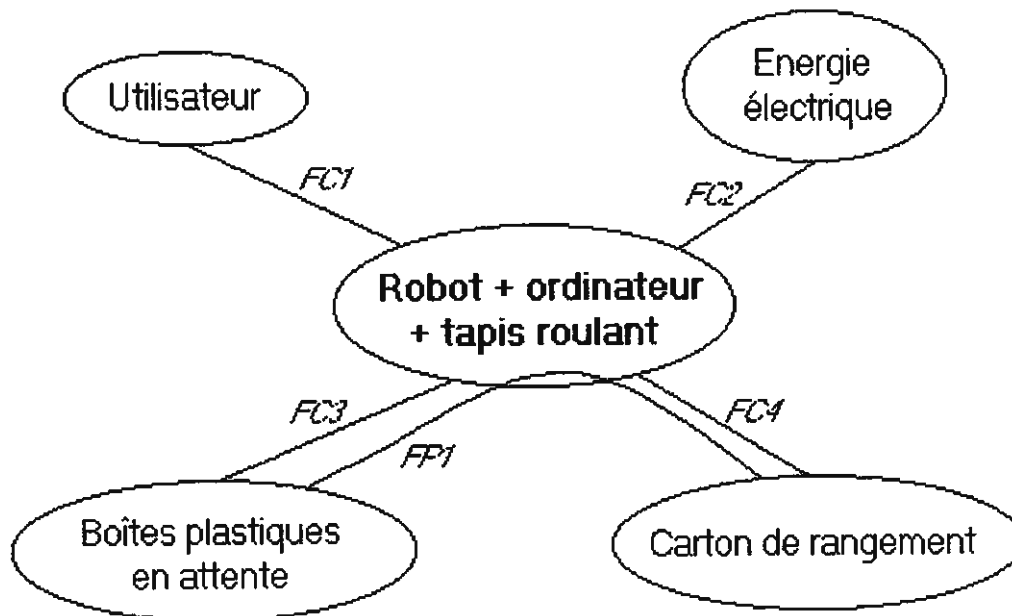
➤ Analyse fonctionnelle

Le bras manipulateur a pour but premier d'effectuer une séquence qui peut se traduire par la réalisation d'un cycle dans lequel il viendra agripper une pièce sur le tapis roulant et la déposera par la suite dans un carton, par exemple.



Il est donc question de créer un automatisme en associant le bras manipulateur avec un tapis roulant ...

➤ Diagramme pieuvre du système.



- FP1 : Ranger les boîtes en attente sur le tapis.
- FC1 : Permettre à l'utilisateur de donner des consignes
- FC2 : Alimenter en courant électrique le système.
- FC3 : Doit arrêter les boîtes sur le tapis pour que le robot puisse les prendre.
- FC4 : Le carton de rangement doit être près du système.

✓ En outre, il est important de préciser que le robot a été conçu pour répondre à certaines exigences industrielles, à savoir :

-la fiabilité :

Ce bras manipulateur a la capacité d'effectuer un grand nombre de tâches et il a été prévu pour pouvoir réaliser le maximum d'exigences de l'utilisateur et ce avec une fiabilité très convenable, du moins au temps où il a été créé.

-la résistance :

Les matériaux utilisés pour la conception de ce robot s'avèrent être très résistants puisqu'ils sont pour la plupart de fer ce qui le rend robuste.

-la sécurité :

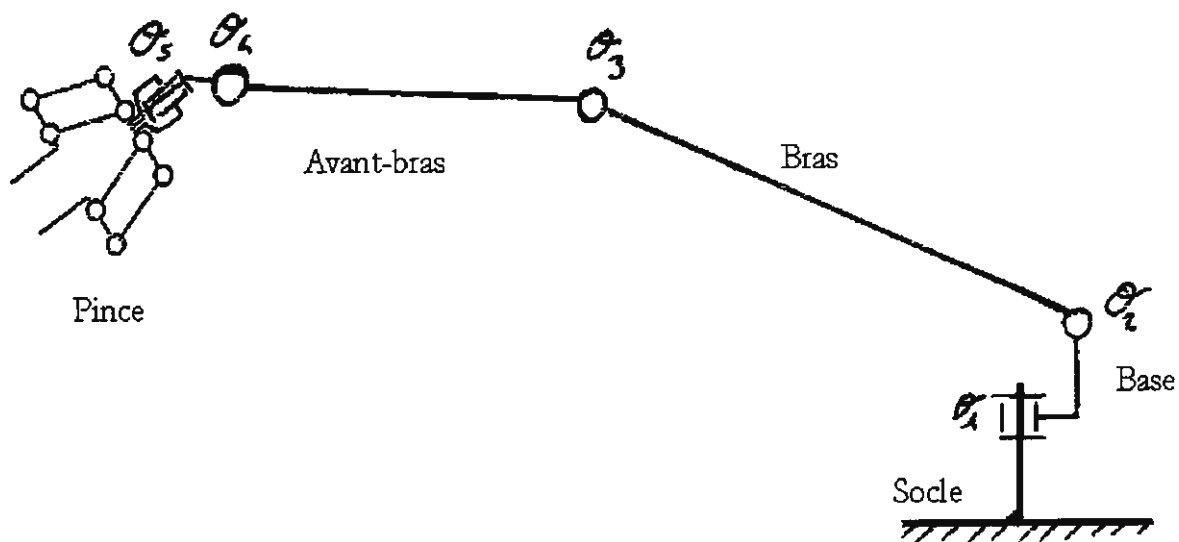
On remarque la présence de nombreux caches et carters sur l'ensemble du robot, ce qui permet de protéger l'utilisateur et le robot d'éventuels problèmes. A part ça, il faut ajouter que le boîtier séparant le bras de l'ordinateur est également muni de boutons coup de poings en cas d'arrêt d'urgence.

-la précision :

Sur l'interface de contrôle du bras, on trouve un certain nombre de possibilités de réglages de pas des moteurs afin d'augmenter considérablement la précision des gestes. D'autre part, la complexité du robot prouve sa précision malgré son ancienneté.

➤ Schémas Cinématiques.

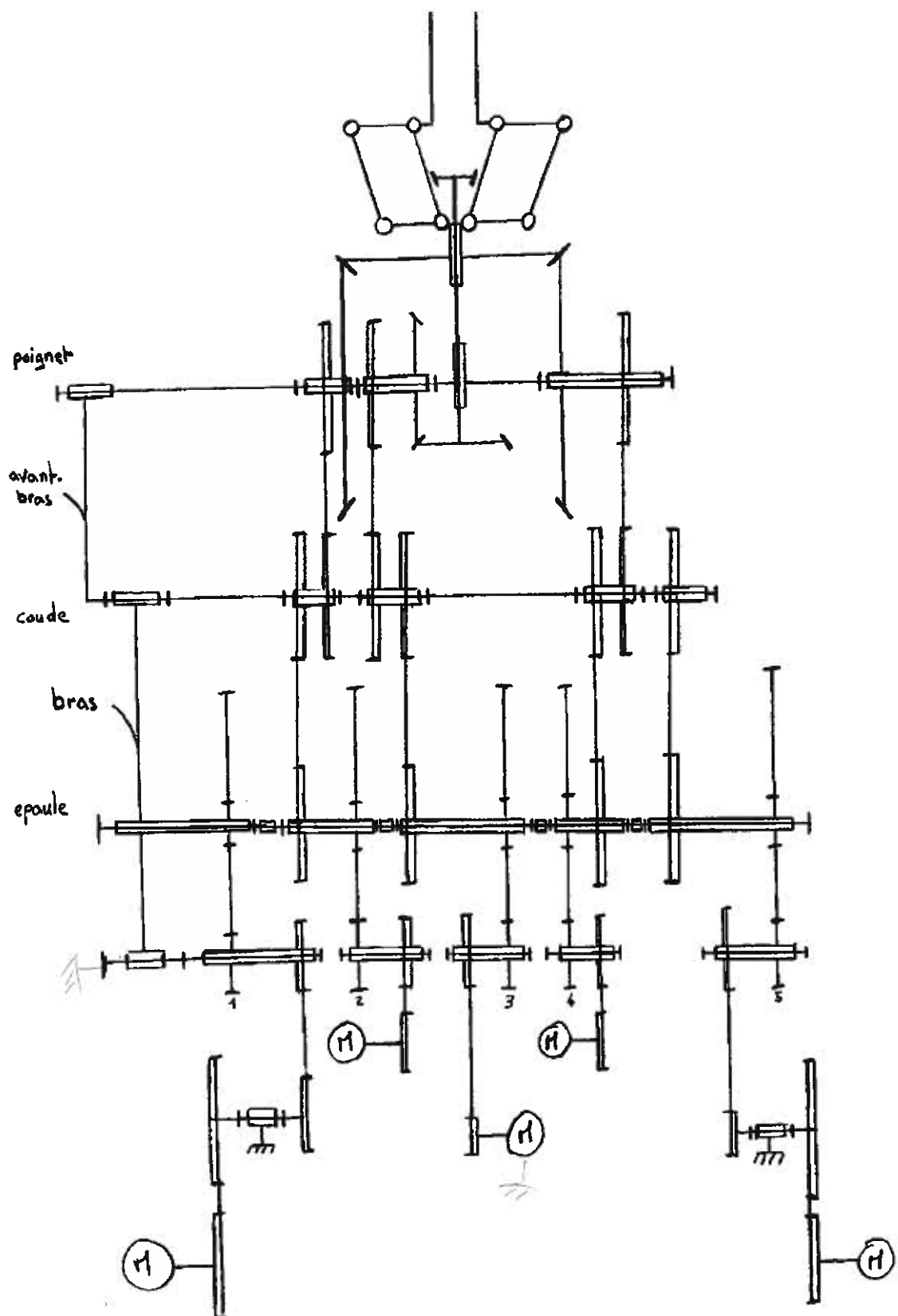
- Vue d'ensemble 1 du Robot :



Le robot a 5 axes montés en liaisons pivot. Le socle est fixé au support, lui-même solidaire de la machine avec le tapis roulant.

Cf vue du schéma cinématique complet du robot page suivante ...

- Vue d'ensemble 2 du robot:



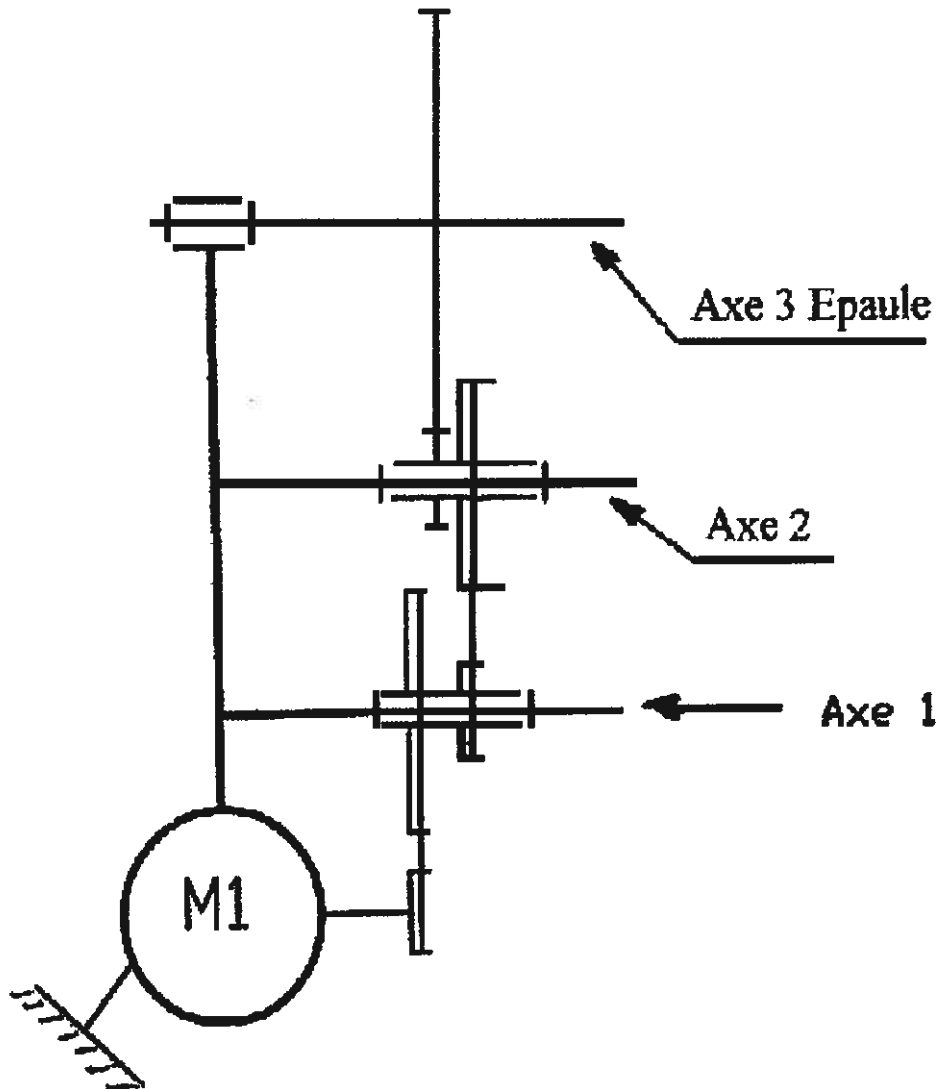
- Les mouvements simples de rotation (ne nécessitant qu'un moteur)

-La base :

La liaison pivot σ_1 est réalisée par un système poulie/courroie simple.

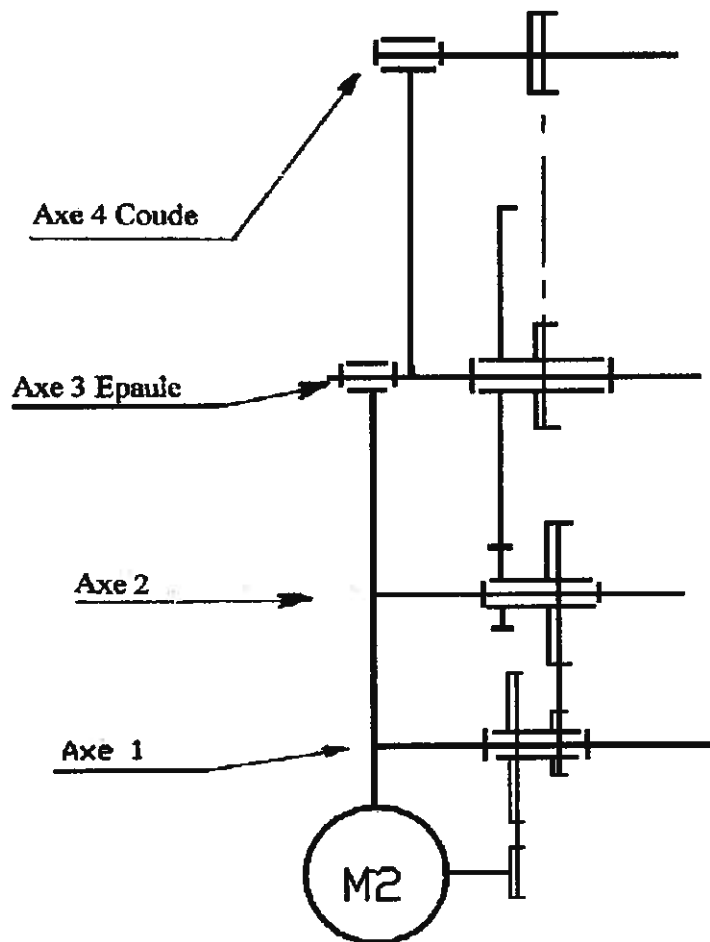
-Rotation autour de l'axe de l'épaule (liaison σ_2) :

Grâce à une rotation en σ_2 , le bras peut s'abaisser ou remonter. Seul le moteur M1 intervient pour faire ce mouvement.



Nous trouvons une transmission de type poulie/courroie, avec aussi un petit pignon qui entraîne une roue. On peut remarquer que le sens de rotation de la roue fixée sur l'axe 3 est l'inverse du sens à la sortie du moteur. Cet inversement est dû au pignon et la roue, les transmissions poulie/courroie conservant le sens de rotation. Ces deux système de transmission servent à découpler l'effort . Cependant la vitesse est réduite.

-Rotation en σ_3 :

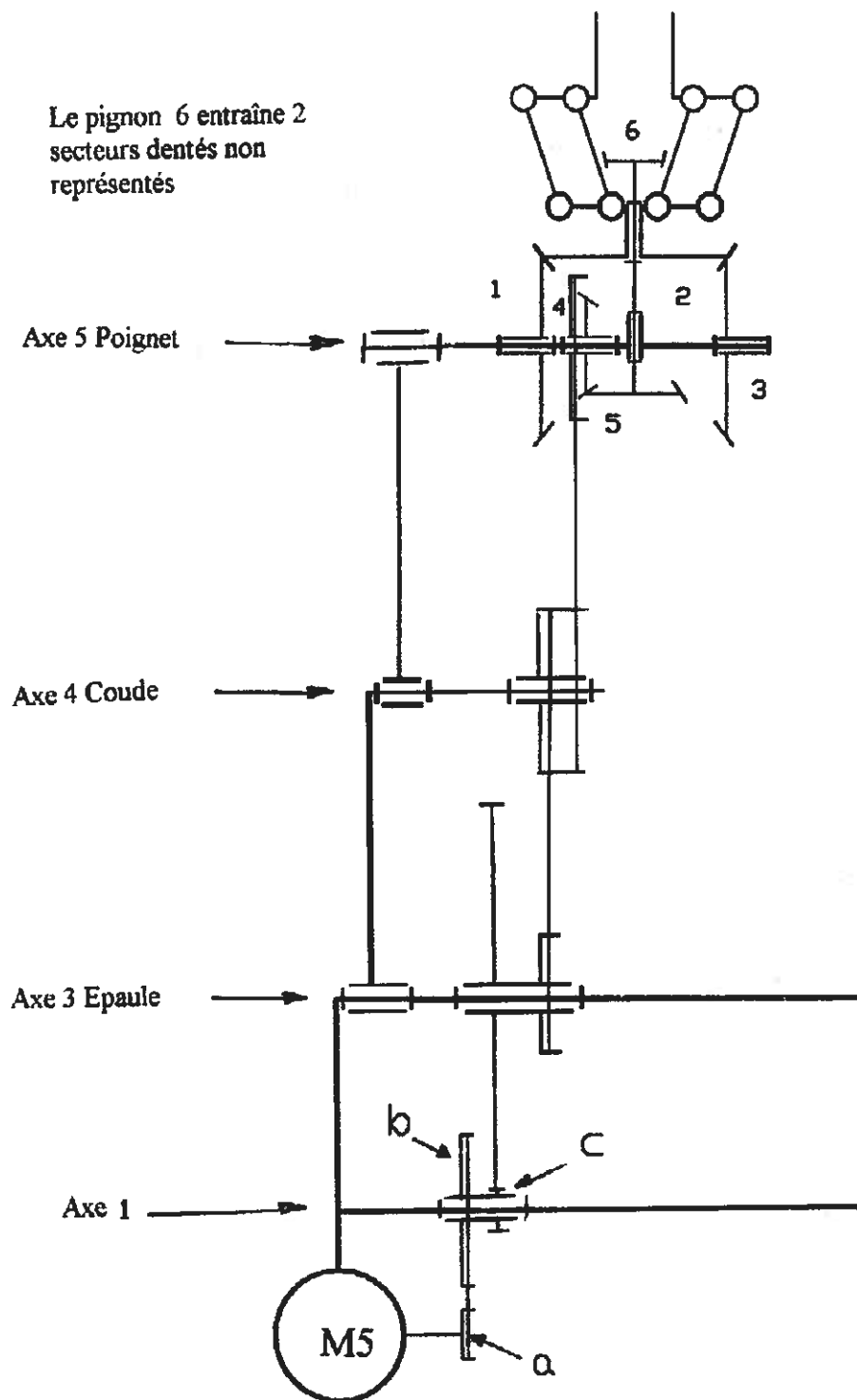


Même commentaire que précédemment (il y a aussi une inversion de sens). Lorsque l'on fait tourner le moteur M2, on fait s'abaisser ou se lever l'avant-bras. Les efforts pouvant être assez conséquents (il s'agit de monter par exemple un objet lourd), une chaîne a été mise à la place de la courroie pour la liaison de l'axe 3 à l'axe 4.

- Les mouvements nécessitant plusieurs moteurs :

On peut remarquer la présence de 6 moteurs installés dans le robot. En effet, il faut aussi un moteur pour l'ouverture de la pince, en plus des 5 autres pour les pivots σ_1 à σ_5 (voir schéma du début).

Le robot est doté d'un différentiel à la pince. Mais il rend les mouvements plus compliqués à produire car on doit pour une simple rotation mettre en marche 2 voire 3 moteurs différents. En effet, il nous est arrivé à plusieurs reprises d'avoir affaire à une défaillance de la pince (principalement lors de la fermeture de celle-ci) entraînant l'emballement du robot qui se met à effectuer tous les mouvements à la fois. C'est pourquoi il est recommandé de toujours se tenir prêt à le stopper lors de l'emploi de la pince ! Un problème que nous n'avons pu qu'atténuer mais qui est toujours probable, notamment lorsque la séquence atteint la prise de la boîte sur le tapis roulant.



La rotation du moteur M5 permet l'ouverture de la pince en faisant tourner le pignon 4 qui lui-même entraîne le pignon 5. L'axe sur lequel est fixé le pignon 6 se met en rotation, le pignon 6 entraîne les deux secteurs dentés : la pince s'ouvre (ou se ferme).

Cependant, pour tous les autres mouvements faisant appel au différentiel, au moins deux moteurs sont utiles pour un mouvement :

	M1	M2	M3	M4	M5
M1	M1+ : bras sens + M1- : bras sens -	M1+M2+ : bras+ et Fermeture pince M1+M2+ : bras+ et Ouverture pince	M1+M3+ : bras+ et rotation pince bas-gauche M1+M3- : bras+ et rotation pince haut-droite	M1+M4+ : bras+ et Rotation pince bas-droite avec ouverture pince M1+M4- : bras+ et Rotation pince haut-gauche avec fermeture pince	M1+M5+ : bras et avant bras sens+ (mvt accordé) M1+M5- : bras + et avant bras - (mvt inverse)
M2	M2+M1- : bras- et fermeture pince M2-M1- : bras- et ouverture pince	M2+ : fermeture pince M2- : ouverture pince	M2+M3+ : fermeture pince et rotation pince bas-gauche M2+M3- : fermeture pince et rotation pince haut-gauche	M2+M4+ : rotation pince bas-droite M2+M4- : rotation pince haut-gauche avec fermeture de la pince	M2+M5+ : avant bras + et fermeture pince M2+M5- : avant bras - et fermeture pince
M3	M3+M1- : bras- et rotation pince haut-droite M3-M1- : bras- et Rotation pince bas-gauche	M3+M2- : ouverture pince et rotation pince haut-droite M3+M2- : ouverture pince et rotation pince bas-gauche	M3+ : rotation pince Bas-gauche M3- : rotation pince haut-droite	M3+M4+ : pince vers le bas et ouverture pince M3+M4- : rotation gauche avec fermeture de la pince	M3+M5+ : avant bras + et rotation pince bas-gauche M3+M5- : avant bras - et rotation pince bas-gauche
M4	M4+M1- : bras- et Rotation pince bas-droite avec ouverture de la pince M4-M1- : bras- et Rotation pince haut-gauche avec fermeture de la pince	M4+M2- : rotation pince Bas-droite avec ouverture de la pince M4-M2- : rotation pince haut-gauche	M4+M3- : rotation droite et ouverture pince M4+M3- : pince vers la haut avec fermeture pince	M4+ : rotation pince bas- droite avec ouverture de la pince M4- : rotation pince haut gauche avec fermeture de la pince	M4+M5+ : avant bras + et rotation pince bas-droite avec ouverture pince M4+M5+ : avant bras moins et rotation pince bas droite avec ouverture pince
M5	M5+M1- : bras- et avant-bras+ (mvt inverse) M5-M1- : bras- et avant-bras- (mvt accordé)	M5+M2- : avant-bras+ et Ouverture pince M5-M2- : avant-bras- et ouverture pince	M5+M3- : avant-bras+ et Rotation pince haut-droite M5-M3- : avant-bras- et Rotation pince haut-droite	M5+M4- : avant-bras + et rotation pince haut-gauche avec fermeture pince M5-M4- : avant-bras – et rotation pince haut-gauche avec fermeture pince	M5+ : avant bras sens + M5- : avant bras sens -