ROBOT PEDAGOGIQUE

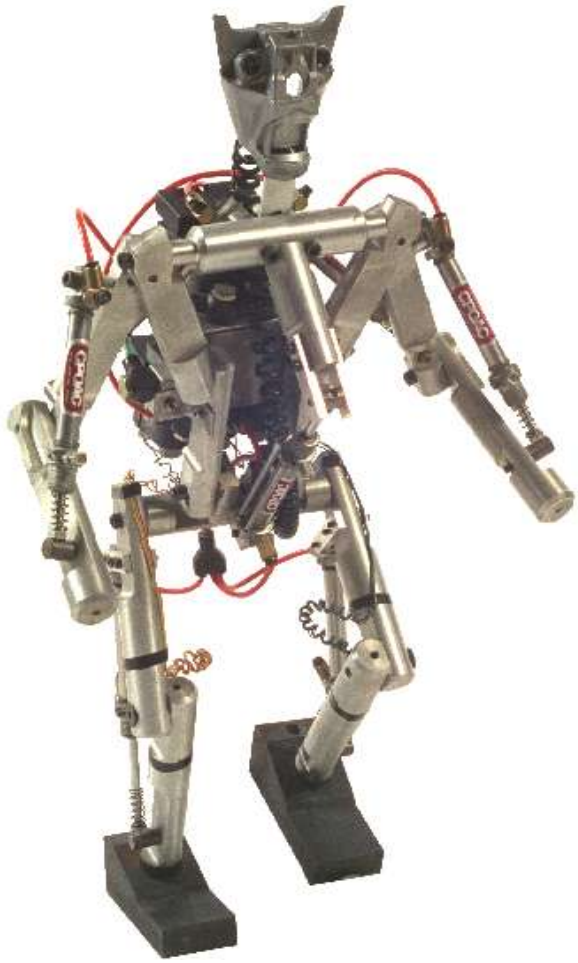


Table des matières

Chapitre 1. Introduction

1. Utilisation

J'utilise aujourd'hui cette maquette dans les cours de technologie que je dispense en classes de niveau collège. Elle permet l'apprentissage des automatismes ainsi qu'une introduction au système binaire, décimal et hexadécimal.

2. Généralités

Le système est composé de plusieurs élèments distincts. L'ensemble est cadencé par l'horloge d'un micro-ordinateur de type PC. la liaison maquette/micro-ordinateur est assurée par le port parallèle du PC via une interface électronique qui permet d'amplifier et de protéger les sorties TTL du micro-ordinateur (5V => 24V). L'interface électronique alimente une rampe de relais qui alimentent des électrovannes. ces électrovannes alimentent des verins pneumatiques.

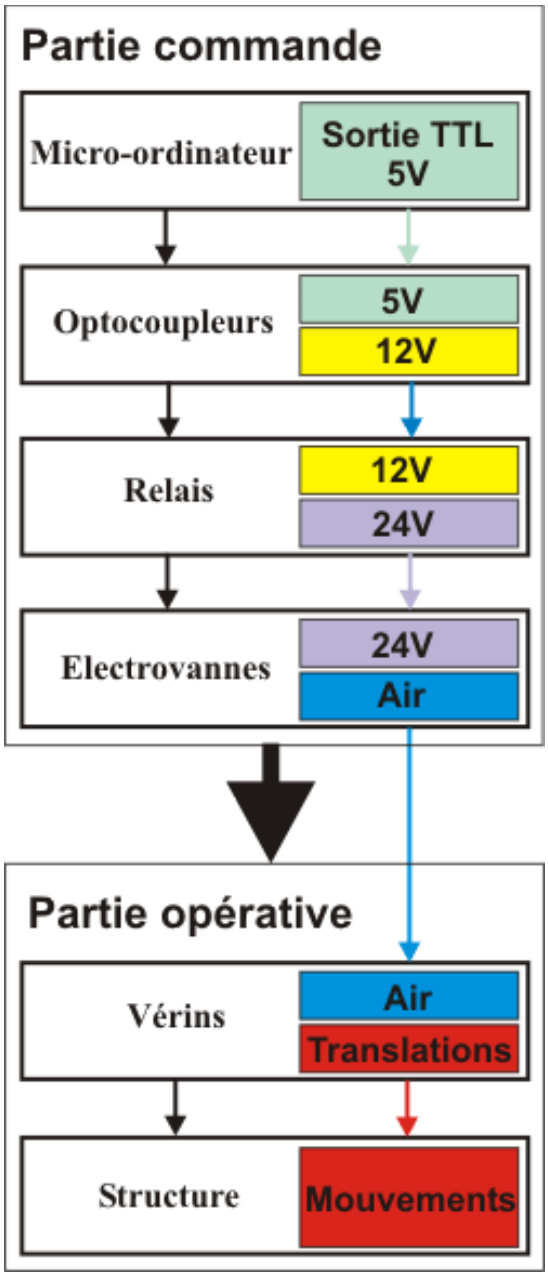
3. Actions programmables

On compte 7 actions programmables :

1. Mouvement jambes + torse
2. Mouvement épaule droite
3. Mouvement épaule gauche
4. Mouvement bras droit
5. Mouvement bras gauche
6. Allumer oeil droit
7. Allumer oeil gauche

4. Synoptique

Figure 1.1. Synoptique



Chapitre 2. Partie commande

1. Micro-ordinateur

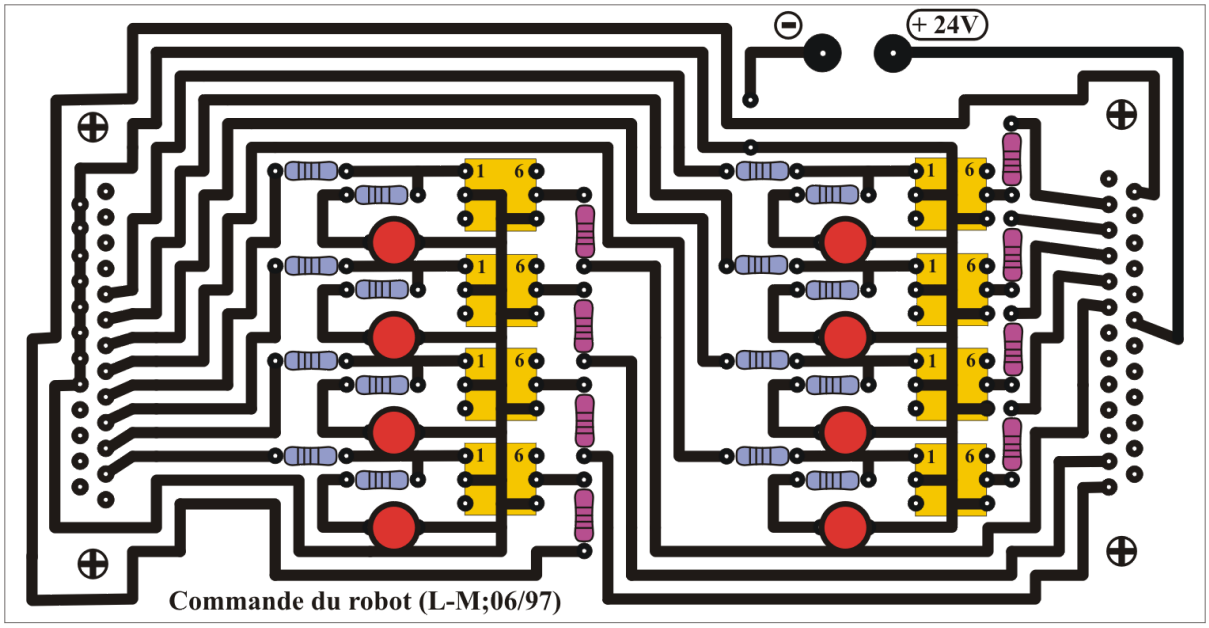
l'ordinateur est un des élèments clef de la partie commande. Comme je l'ai dit précèdemment, c'est l'horloge interne de ce dernier qui rythme le fonctionnement du système. Il permet également à l'utilisateur de commander les différentes actions à effectuer en vue de mettre le robot en mouvement. Plusieurs types de logiciels peuvent être utilisés pour programmer le robot, soir par l'intermédiaire d'une interface graphique, soit grace à un fichier de commande. ils permettent une introduction aux systèmes suivants :

- Système binaire
- Système décimal
- Technologie grafcet

2. Interface parallèle

Afin de protéger les sorties TTL du micro-ordinateur, j'ai réalisé une "carte" électronique isolant ces sorties grâce à des optocoupleurs. Les lignes D0 à D7 du port parallèle sont utilisées, sans multiplexage. les optocoupleurs amplifient les signaux TTL reçus en 12 V afin de les transmettre à une rampe de relais.

Figure 2.1. carte avec composants



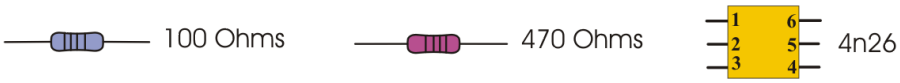
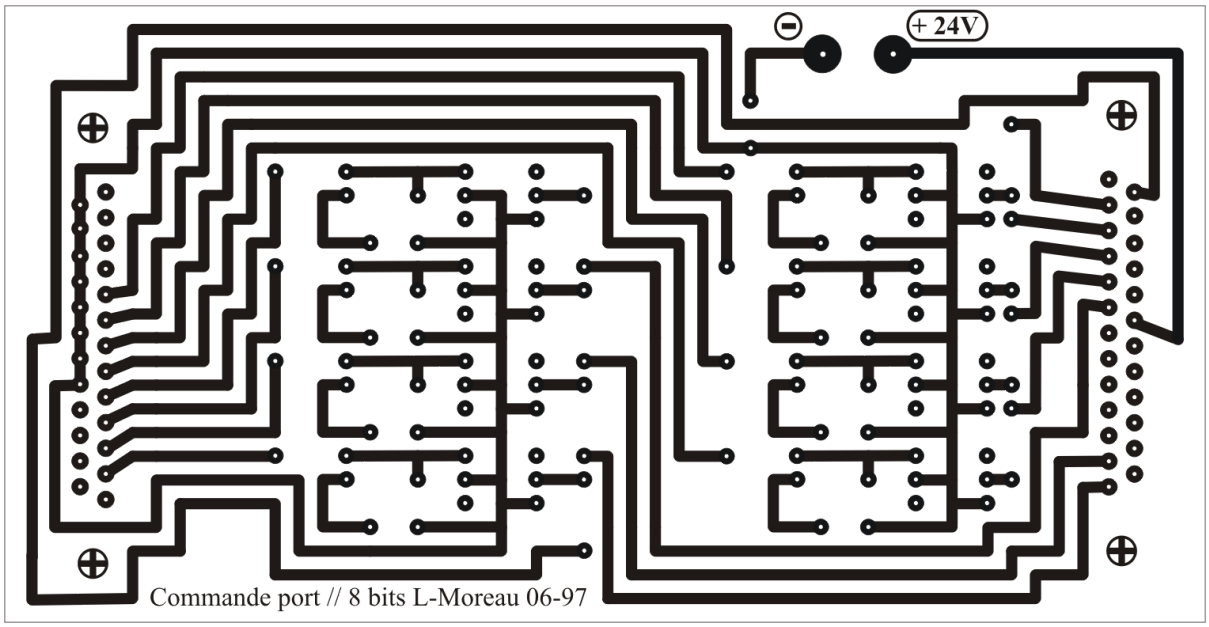


Figure 2.2. Typon



3. Relais

Les bobines des relais sont excitées par les signaux électriques transmis par les optocoupleurs (12V). Cette rampe de relais a pour mission d'exciter les bobines des électrovannes (24V)

4. Electrovannes

Les bobines des électrovannes sont excitées par les signaux électriques transmis (24V) par les relais. Ces electrovannes alimentent les verins pneumatiques en air comprimée.

Chapitre 3. Partie opérative

1. Structure mécanique

La structure modélise un "corps" de forme humaine. Cette dernière est réalisée en aluminium usiné à partir de lopins de 40 mm de diamètre. Les articulations sont assurées par des vis de type six pans montées sur des bagues en laitons.

> **Note**
>
> Je précise que cette maquette est un "prototype" ...

2. Vérins pneumatiques

Les vérins sont de type double-effets mais ne sont utilisés qu'en mode simple effet. leurs retours en position initiale sont assurés par des ressorts.

3. Cinématique

On compte cinq mouvements d'ordre cinématique :

1. Lever du robot : déploiement simultané des deux jambes associé au redressement du torse.
2. Mise en mouvement du bras droit.
3. Mise en mouvement du bras gauche.
4. Mise en mouvement de l'épaule droite.
5. Mise en mouvement de l'épaule gauche.

Chapitre 4. Logique de programmation

1. Principe

Le robot se programme de manière séquentielle. C'est à dire que c'est le temps qui conditionne le passage d'un pas à une autre. Pour chaque pas, il est possible de définir huit actions logiques. chaque action logique est associée à un mouvement (bras, épaule ...) ou à l'allumage des yeux. Un pas est codé sur un octet, soit 8 bits. chaque action est associée à un bit de donné. Lorque qu'un bit porte la valeur "zero" (0), l'action associée n'est pas réalisée lors du pas. A l'inverse, lorsqu'un un bit porte la valeur "un" (1), l'action associée sera réalisée lors du pas.

2. Codage des actions

```
Bit N° 1 => Epaule gauche
Bit N° 2 => Bras gauche
Bit N° 3 => Oeil gauche
Bit N° 4 => Oeil droit
Bit N° 5 => Epaule droite
Bit N° 6 => Bras droit
Bit N° 7 => Jambes + torse
Bit N° 8 => Non attribué
```

3. Définiton d'une séquence

Un mouvement correspond à l'état logique de l'actionneur

```
En écrivant [11111111], toutes les actions seront réalisées.
En écrivant [00000000], aucune action ne sera réalisée.
En écrivant [00001100], j'allume les deux yeux.
```

4. Mouvement séquentiel

Pour élaborer un mouvement complexe, il faut définir une suite de pas temporisés qui seront lues de manière linéaire.

Chapitre 5. Interface logicielle

1. Interface

En démarrant le logiciel de programmation du robot, la fenêtre suivante apparaît :

Figure 5.1. Menu

1. Chargement d'une animation = Permet d'éxécuter une animation déjà enregistrée

2. Saisie d'une nouvelle animation = Création d'une nouvelle animation.

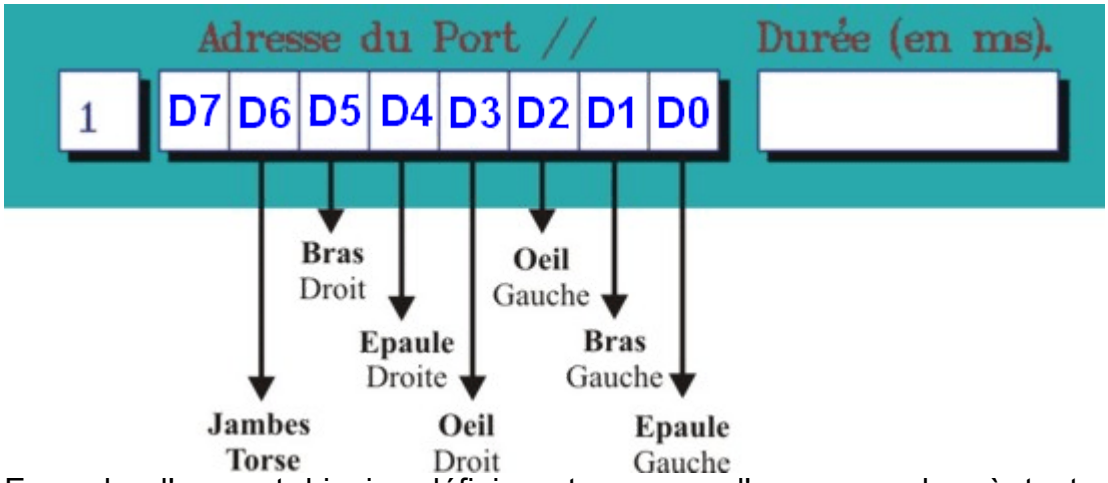
3. Quitter (Retour sous DOS) = Quitter l'interface de programmation.

Pour créer une nouvelle animation, choisir l'option N°2. (Entrer « 2 » puis valider)

2. Correspondance des actions

1. Avant de commencer à programmer le robot, il est nécéssaire de connaître les correspondances entre les actions et les bits qui leur sont associés :

Figure 5.2. Correspondance des actions



2. Exemple d'un mot binaire définissant un pas d'une seconde où toutes les actions seront éxécutées à l'exeption de l'épaule gauche.

Figure 5.3. Détail d'un mot binaire

3. Nombre de séquences

Dans la fenêtre qui suit, nous allons spécifier le nombre de pas que comportera l'animation. Dans ce tutoriel, nous allons réaliser une animation comportant 3 pas.

1. Entrer le chiffre « 3 » (3 pas), puis valider.

4. Définition des actions au sein d'un pas

Chaque pas comprend des actions. Dans cet exemple, nous allons commander les actions suivantes au sein de 3 pas :

```
Pas 1 : Allumer l'œil gauche pendant 1 seconde
Pas 2 : Allumer l'œil droit pendant 1 seconde
Pas 3 : Allumer les deux yeux pendant 1 seconde
```

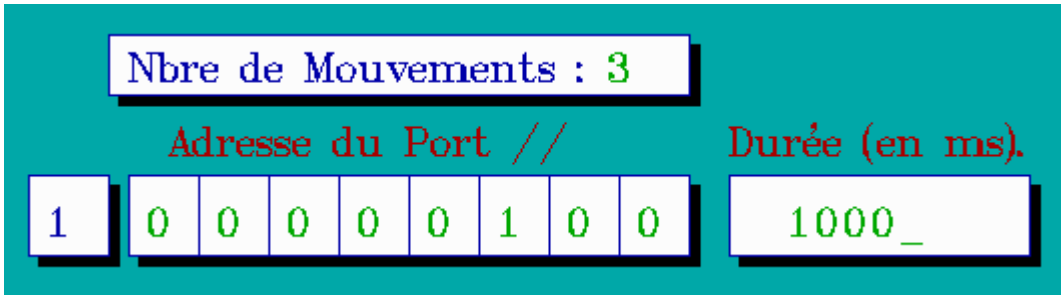
4.1. Séquence N°1

1. L'action à effectuer lors de la première séquence est d'allumer l'oeil gauche pendant 1 seconde.

2. Je me reporte au schéma des correspondances afin de déterminer à quel bit correspond l'action "oeil gauche".

3. Action oeil gauche = bit N°3

4. J'indique donc "1" comme valeur pour le bit N°3 (état "haut" = action éffectuée)



5. Toutes les autres actions restent à "0" (étabt "bas" = pas d'action)

6. Je spécifie la durée du pas : 1 s = 1000 ms

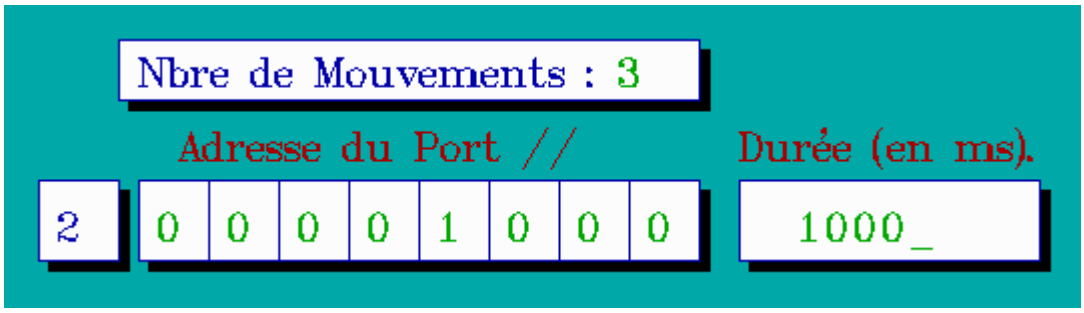
4.2. Séquence N°2

1. L'action à éffectuer lors du deuxième pas est d'allumer l'oeil droit pendant 1 seconde.

2. Je me reporte au schéma des correspondances afin de déterminer à quel bit correspond l'action "oeil droit".

3. Action oeil droit = bit N°4

4. J'indique donc "1" comme valeur pour le bit N°4(état "haut" = action éffectuée)



5. Toutes les autres actions restent à "0" (étabt "bas" = pas d'action)

6. Je spécifie la durée du pas : 1 s = 1000 ms

4.3. Séquence N°3

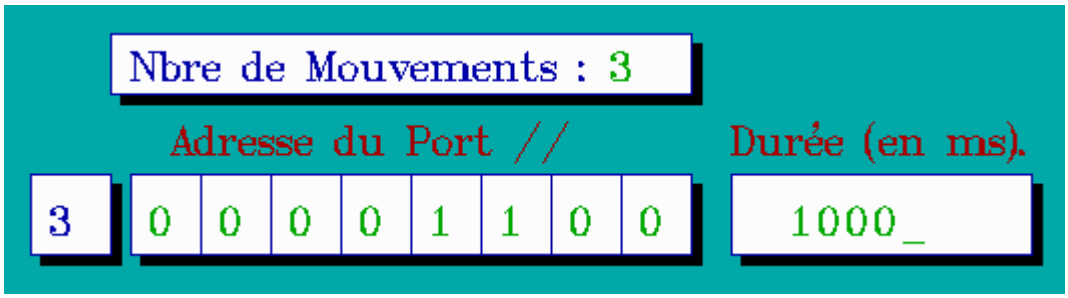
1. L'action à éffectuer lors du troisième pas est d'allumer l'oeil gauche et l'oeil droit pendant 1 seconde.

2. Je me reporte au schéma des correspondances afin de déterminer à quel bit de donnée correspond l'action "oeil droit" et "oeil gauche"

3. Action :

   - oeil gauche = bit N°3
   - oeil droit = bit N°4

4. J'indique donc "1" comme valeur pour le bit N°3 et N°4 (état "haut" = action éffectuée)

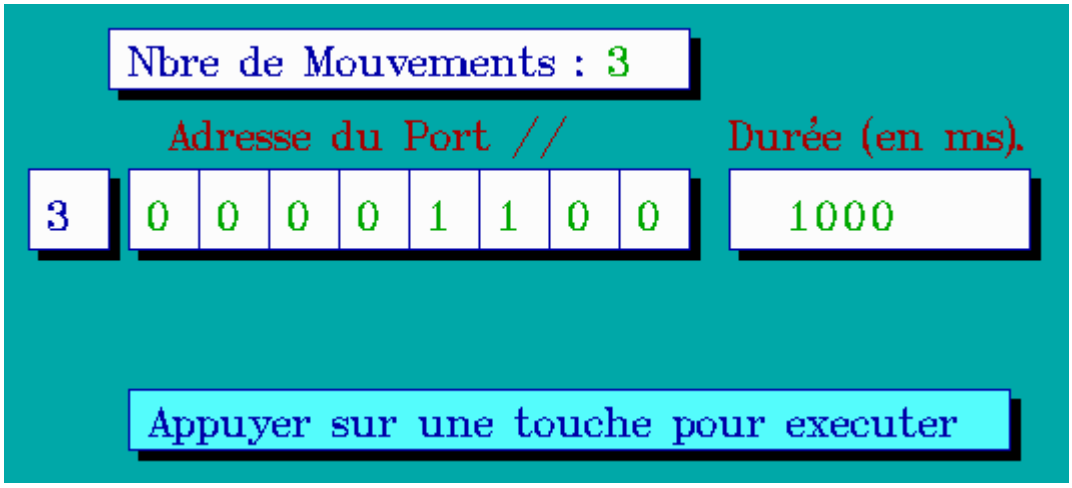


5. Toutes les autres actions restent à "0" (étabt "bas" = pas d'action)

6. Je spécifie la durée du pas : 1 s = 1000 ms

5. Exécuter une suite de mouvements

1. Quand les trois pas sont correctement définis, la fenêtre suivante s'affiche :



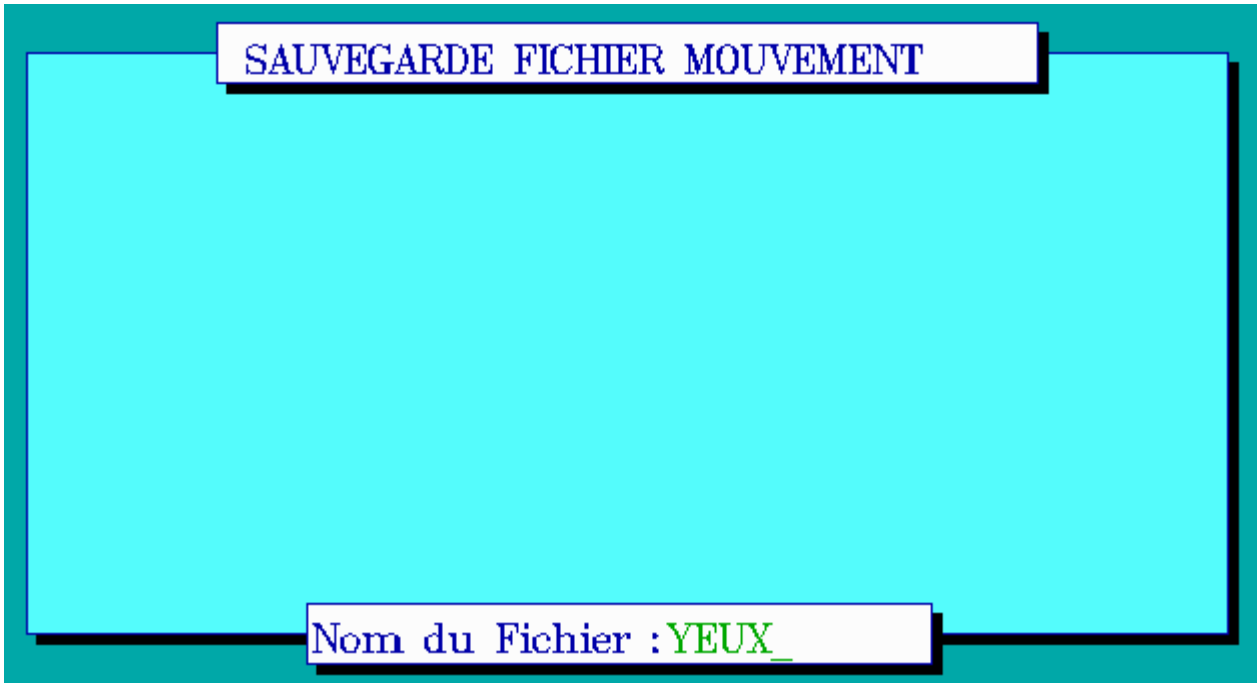
2. Valider en appuyant sur une touche pour lancer l'animation créée.

6. Sauvegarder une animation

Lorsque l'éxécution de l'animation est terminée, je peux sauvegarder l'animation :

1. Appuyer sur la touche "espace" pour commander la sauvegarde de l'animation

2. Donner un nom à l'animation sauvegardée (ex : "yeux"):



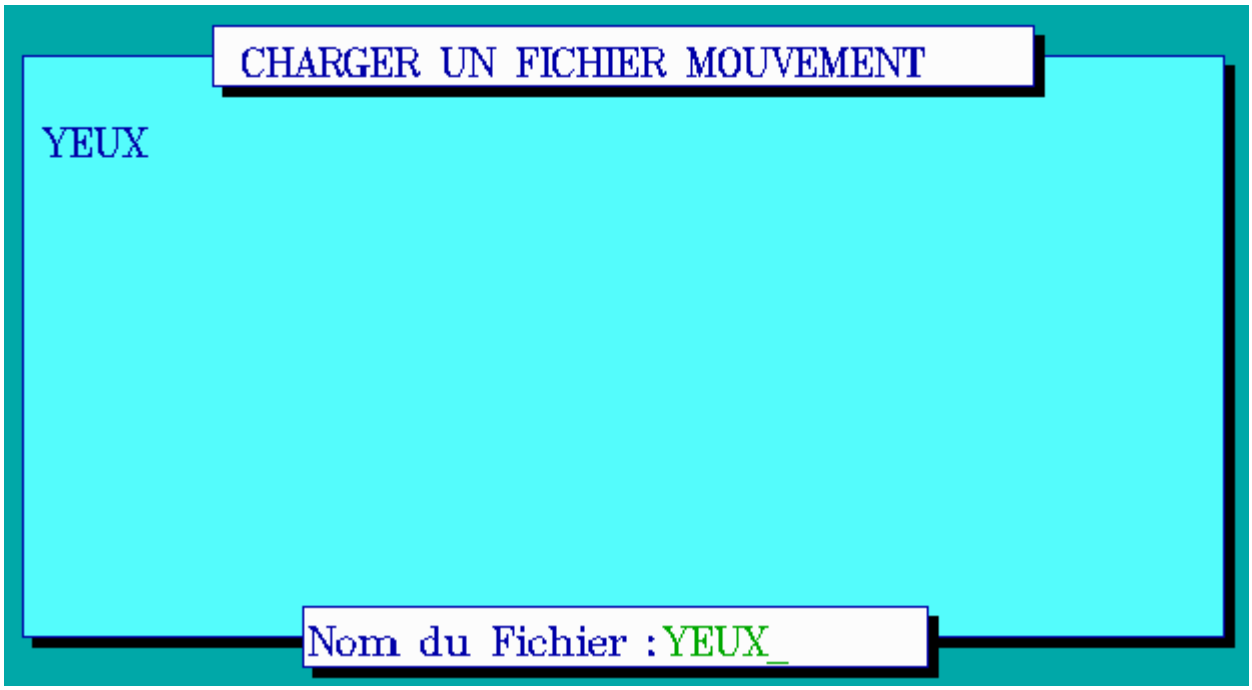
3. Valider en appuyant sur la touche "entrée"

> **Note**
>
> L'extension du fichier est générée automatiquement : *.MSP (MSP : Mouvement Séquentiel Programmable)

7. Ouvrir une animation sauvegardée

1. Relancer le logiciel de programmation du robot
2. Choisir l'option 1 du menu = Charger une animation
3. Saisir le nom de l'animation à charger

Figure 5.4. Ouvrir une animation existante



4. Valider une appuyant sur la touche entrée.

8. Source du programme (Turbo C++)

L'interface logicielle a été écrite en Turbo C++. Les sources comportent 9 fichiers :

- Robot.exe
- Robot.C
- Robot.H
- Robot_rs.H
- EGAVGA.BGI
- GOTH.CHR
- LITT.CHR
- SANS.CHR
- TRIP.CHR

Vous pouvez télécharger l'archive contenant l'ensemble des fichiers robot.zip

Chapitre 6. Programmation en mode texte (base 10)

1. principe

Il est possible d'écrire une animation sans passer par l'interface logicielle présentée ci-dessus. Ce mode de programmation vise à introduire le processus de conversion binaire/décimal. En effet, dans ce mode de programmation orienté "texte", les actions d'une séquence ne seront plus définies par un "mot binaire" mais par un mot exprimé en base 10 (décimale)

Pour écrire une animation en mode texte, nous utiliserons la commande "edit"

2. Exemple

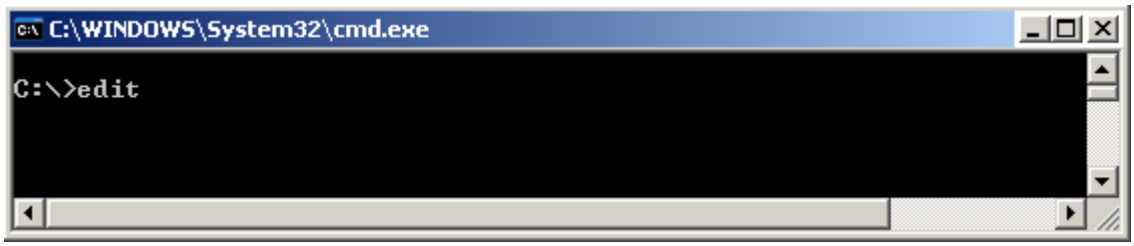
Dans cet exemple, nous reprendrons les trois séquences étudiées lors de l'apprentissage de l'interface graphique.

Rappel :

Pas 1 : Allumer l'œil gauche pendant 1 seconde
Pas 2 : Allumer l'œil droit pendant 1 seconde
Pas 3 : Allumer les deux yeux pendant 1 seconde

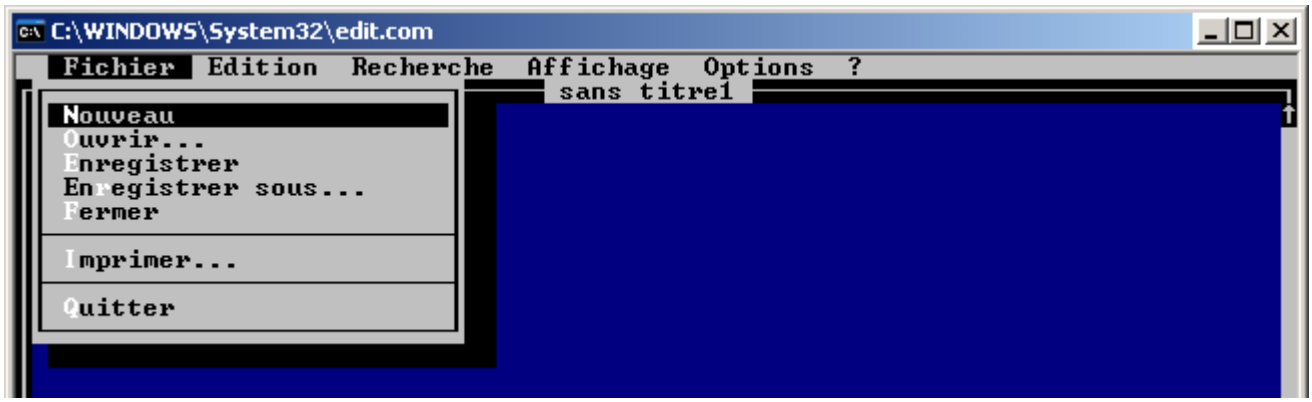
1. lancer "Edit"

Figure 6.1. lancer Edit



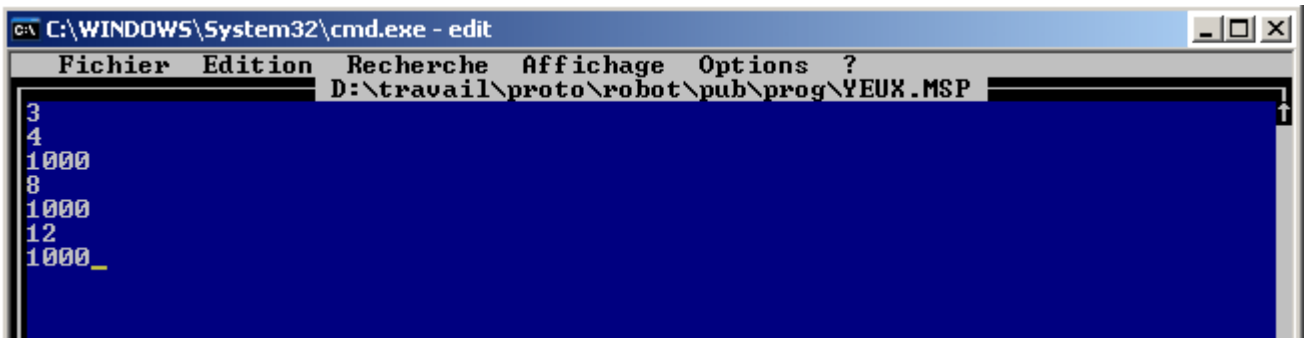
2. Commander la création d'un nouveau fichier

Figure 6.2. Nouveau fichier



3. Saisir le programme

Figure 6.3. programme



Détail :

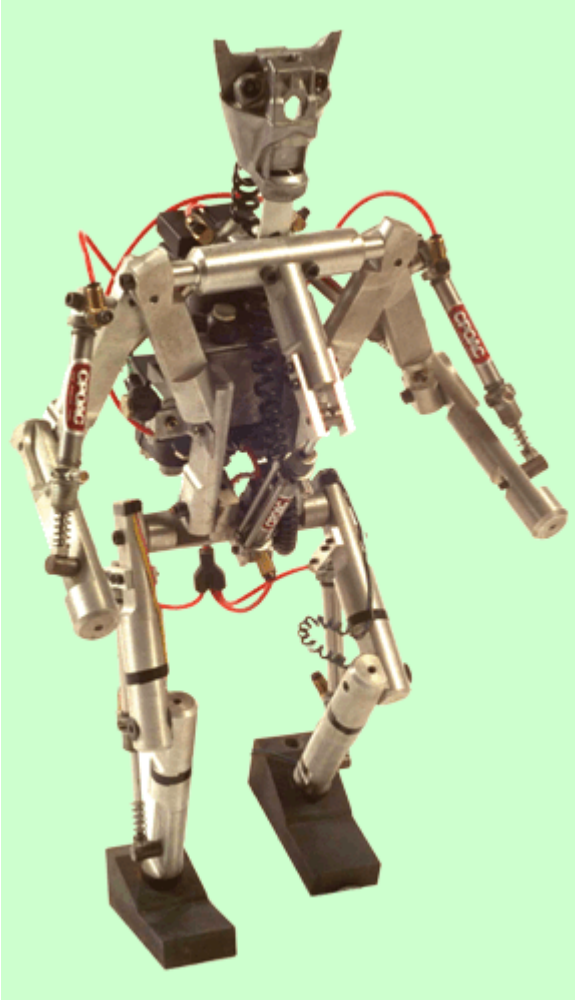
```
3       // nombre de pas
4       // actions du 1er pas (Mot binaire = 100 ; 4 en base 10)
1000     // durée du 1er pas
8       // actions du 2ème pas  (Mot binaire = 1000 ; 8 en base 10)
1000     // durée du 2ème pas
12      // actions du 3ème pas  (Mot binaire = 1100 ; 12 en base 10)
1000     // durée du 3ème pas
```

4. Enregistrer le programme en lui donnant l'extension "MSP" (EX : yeux1.MSP)
5. Lancer l'interface graphique de programmation du robot
6. Choisir l'option 1 du menu = charger une animation
7. Sélectionner "yeux1"
8. Appuyer sur une touche pour lancer l'éxécution de l'animation

Chapitre 7. Photographies

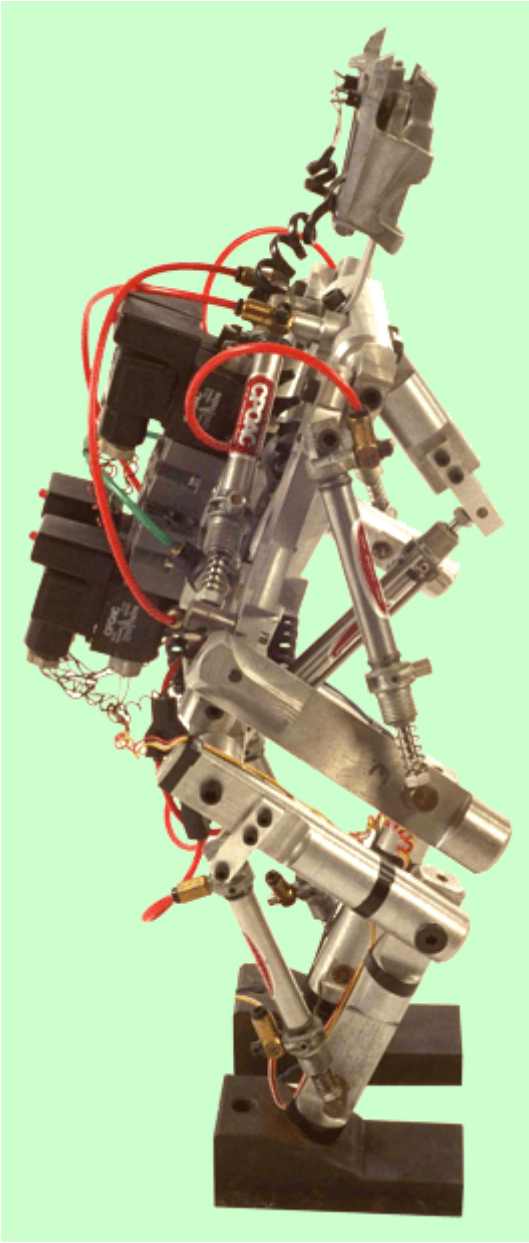
1. Vue de face

Figure 7.1. Vue de face



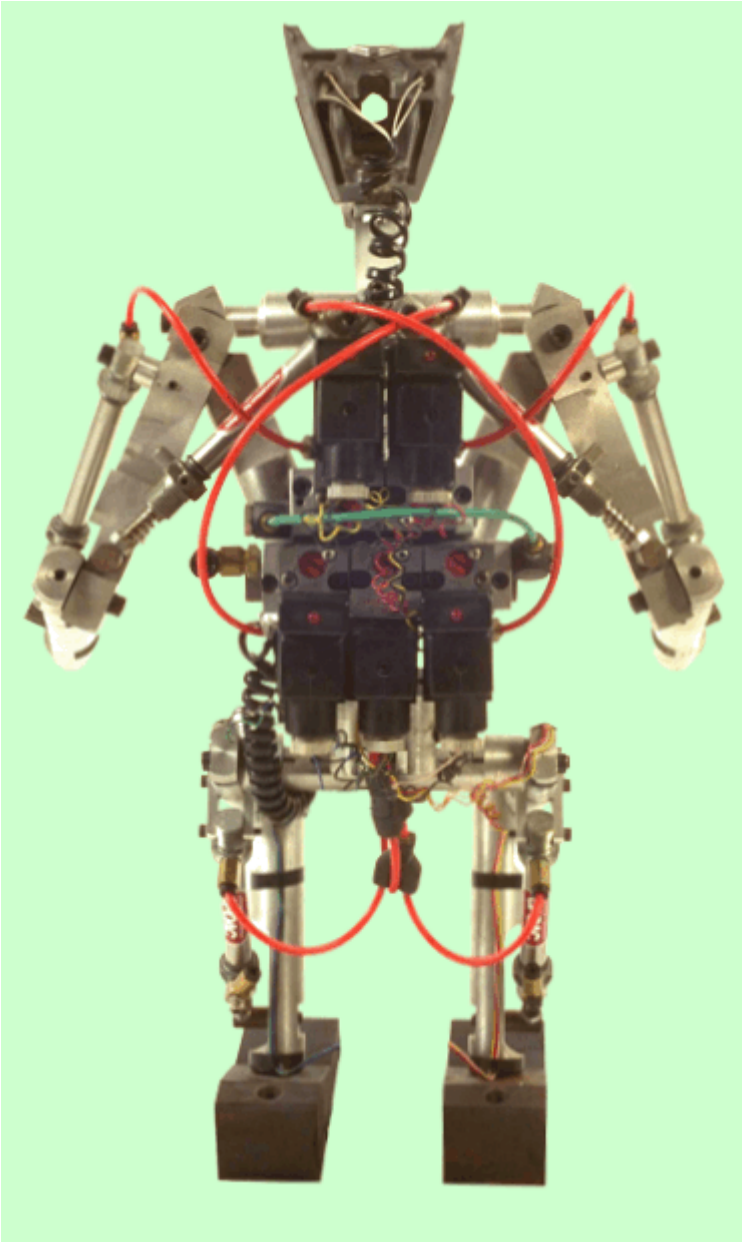
2. Vue de profil

Figure 7.2. Vue de profil



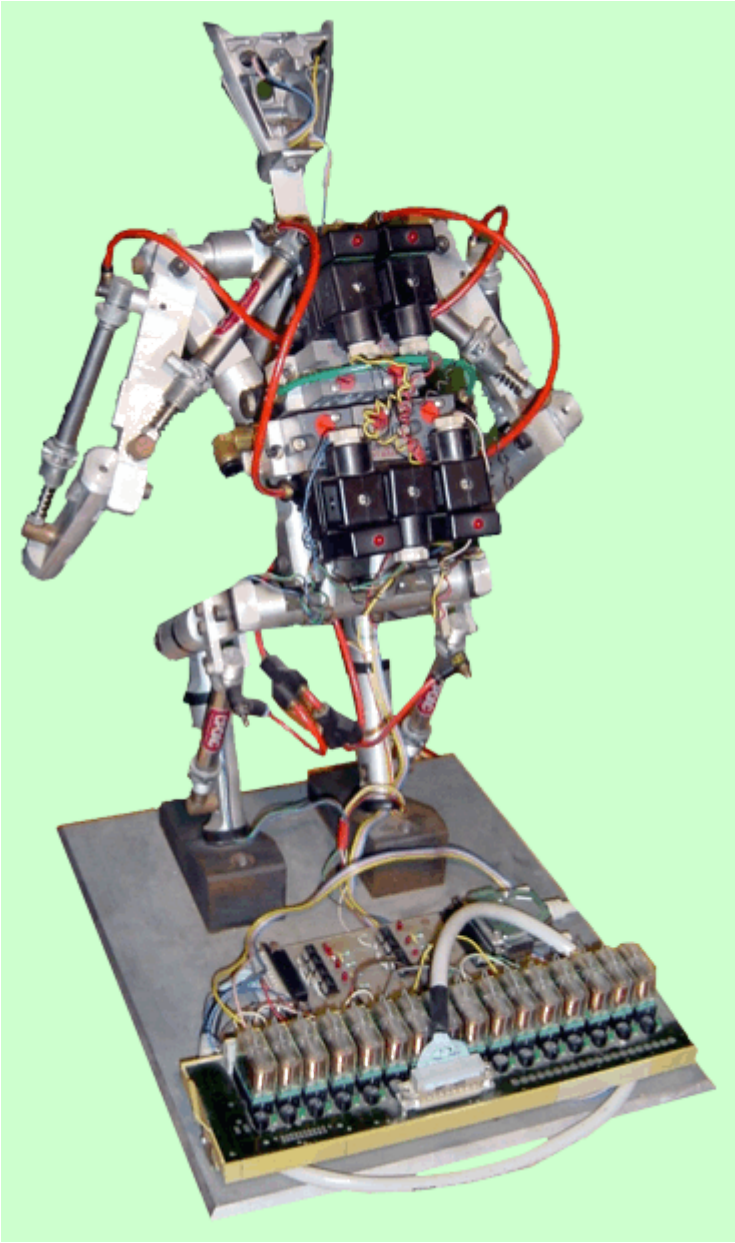
3. Vue de dos

Figure 7.3. Vue de dos



4. Ensemble robot-platine

Figure 7.4. Vue de dos avec platine



Chapitre 8. Vidéos

Vous trouverez ci-dessous deux vidéos de la maquette en fonctionnement (format mpeg)

- robot_01.MPG (taille ~ 5 Mo)
- robot_02.MPG (taille ~ 5 Mo)

Chapitre 9. Evolutions

J'ai beaucoup d'idées concernant les perspectives d'évolution de cette maquette. Ces évolutions entraîneraient la réalisation d'une nouvelle maquette, plus évoluée. Cependant, je compte réaliser certaines des évolutions suivantes dans un avenir proche, sans changer la structure de la maquette.

1. Accès internet + visioconference

A l'avenir, la programmation de cette maquette sera accessible depuis le réseau internet. Après l'écriture d'un programme, l'élève aura la possibillité d'animer la maquette et d'observer le résultat de son programme à l'écran via un processus de type visioconference (une caméra filme le robot et diffuse l'image sur le poste client). Pour cette fonctionnalité, seule la partie logicielle est à mofifier.

2. Ajout de capteur

Aujourd'hui, la maquette se programme uniquement de manière séquentielle. C'est donc uniquement la notion temps qui rythme le passage d'un pas à un autre. L'ajout de capteurs permettrait un mode de programmation plus complet admettant des interactions avec le monde extérieur. dans ce cas de figure, c'est au niveau de l'interfaçage qui'il faudrait apporter des modifications (multiplexage)

3. Module de préhension

Il serait intéressant d'ajouter une pince au robot lui permettant de saisir un objet. Pour cette fonctionnalité, aucune modification majeure n'est à prévoir.