

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

47/87

وزارة التعليم و البحث العلمي
Ministère de l'Enseignement et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT GENIE MECANIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

2 ex

SUJET

ETUDE D'UN MINI-ROBOT
LE RB4/EV
MUNI D'UN MINI-ORDINATEUR
LE MPZ-30/EV

Proposé par :
M^r A. ZERGUERRAS

Etudié par :
A. CHOUIEUR

Dirigé par :
M^r A. ZERGUERRAS

PROMOTION JUIN 1987

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم و البحث العلمي
Ministère de l'Enseignement et de la Recherche Scientifique

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT GENIE MECANIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

ETUDE D'UN MINI-ROBOT
LE RB4/EV
MUNI D'UN MINI-ORDINATEUR
LE MPZ-830/EV

Proposé par :
M^r A. ZERGUERRAS

Etudié par :
A. CHOWIEUR

Dirigé par :
M^r A. ZERGUERRAS

PROMOTION JUIN 1967

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
المكتبة
BIBLIOTHEQUE
Ecole Nationale Polytechnique

وزارة التعليم العالي

المدرسة الوطنية المتعددة التقنيات

DEPARTEMENT : GENIE MECANIQUE

PROMOTEUR : M^r A. ZERGUERRAS

ELEVE INGENIEUR: A. CHOUIEUR

فروع الهندسة الميكانيكية

الموجه: السيد: أ. زره الراس

المهلب للمهندس: عبدالرحمن تشوير

الموضوع: دراسة قيادة إنسان آلي مشرف مبرمج.
المفهوم: هذه الدراسة تتناول تركيب وبرمجة الإنسان الآلي
المهلب RB4/EV. وانجاز تعاقبات الحركة.

Sujet: Etude de la commande d'un mini-robot programmable

Resume': Cette étude porte sur la constitution et la programmation du mini-robot RB4/EV.

Des séquences de mouvement ont été réalisées

Subject: Study of mini-robot programmable control.

Summary: This study revolves around the mini-robot RB4/EV composition and programming.

A movement sequences have been realized.

Dédicaces

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Je dédie ce modeste travail à

- mes chers parents
- ma chère épouse
- ma chère grand-mère
- mes chers frères et sœurs
- mes chers amis.

Remerciement

Je tiens à exprimer ma forte reconnaissance et mes vifs remerciements à monsieur A. ZERGUERRAS qui a eu l'aimable gentillesse de me fournir son aide à réaliser ce travail, ainsi qu'à tous les enseignants qui ont contribué à ma formation.

Que tous ceux qui ont participé de loin ou de près à la réalisation de ce projet trouveront ici ma sincère gratitude

Mouji

TABLE DES MATIERES

INTRODUCTION	page 1
Chap I: Constitution du robot	3
I.1 Shema. Le robot	3
I.2 Parties principales	4
I.3 Remarque	5
I.4 Les moteurs et leurs pas	6
Chap II La commande électronique	7
II.1 Le mini-ordinateur MPZ-80/EV	7
II.2 Branchement du robot au MPZ-80	7
II.3 Partie électronique	10
II.3.1 Carte d'interface	10
II.3.2 Carte base	11
II.3.3 Carte photo-coupleurs	12
II.4 Commande des phases des moteurs	14
II.5 Selection des moteurs	15
Chap III Description des systèmes automatisés sequentiels. Grafset.	16
III 1 Grafset de niveau 1,2	16
III 1.1 Grafset de niveau 1	16
III 1.2 Grafset de niveau 2	17
III 2 Evolution du grafset	18
III 3 Actions associées aux étapes	19
III 3.1 Actions inconditionnelles	19
III 3.2 Actions conditionnelles	19

III.3.3	Actions impulsionnelles	19
III.3.4	Temporisation	20
III.4	Divergence, Convergence	20
III.4.1	Divergence en ET	20
III.4.2	Convergence en ET	21
III.4.3	Convergence Divergence en OU	21

Chap IV Programmes de gestion du robot avec Le MPZ-80/EV 22

IV.1	Utilisation du programme	22
IV.2	Composition du programme	26
IV.2.1	Programme AROBOT	30
IV.2.2	Programme ROBCOM	36
IV.2.3	Programme MROBOT	39

Chap V Utilisation du mini-robot. 47

V.1	Description de l'operation de chargement	47
V.2	Sequence de mouvement et pas de moteurs	48
V.3	Tracé d'un grafset	49
V.4	Execution de la séquence au moyen du clavier.	52

Conclusion 54

Bibliographie 55

INTRODUCTION

- Emmergeant peu à peu du domaine de la science fiction au succès de laquelle ils contribuaient largement depuis plus de cinquante ans, les robots commencent à envahir le monde du réel. La première génération des robots industriels qui ne sont en fait que des bras mécaniques commandés par ordinateurs fonctionnent dans bon nombre d'usine des pays industrialisés. Ils y effectuent des tâches très diverses mais comprenant toujours des opérations de manipulation.

Il est pourtant curieux de constater que la réalisation pratique de robots fonctionnels n'a débuté que lorsque les ingénieurs se sont rendu compte de la difficulté et de l'inutilité qu'il y avait à donner à une machine à la fois le comportement et l'apparence humaine.

Ils ont préféré concentrer leurs efforts sur la conception de bras mécaniques travaillant d'une façon raisonnable autopomorphe et pouvant être appliqués à une grande variété de tâche.

Le but de ce projet est la mise en œuvre du mini-robot RB4/EV relié au mini computer - trainer mod HPZ 80/EV produits d'Electronica Veneta ont été conçus précisément pour répondre aux exigences d'expérimentation didactique à savoir :

- comprendre la philosophie de base du robot
- intervenir sur le logiciel de contrôle.
- appliquer ce type de robot à de petites expériences.

Il possède 6 degrés de liberté géré par 6 moteurs pas à pas pour les mouvements de la base, de l'épaule, du bras de l'avant bras, du poignet et de la main.

la détection de la position zéro est assurée par des photo-coupleurs situés sur les axes

L'interface avec le MPZ 80/EV est réalisée par des portes Entrée/Sortie, le programme de gestion écrit en basic et en assembleur s'occupe de l'étude, de la correction et de la réalisation de la séquence de mouvement.

Chapitre I

Constitution du robot

I. 1 Schéma. Le Robot.

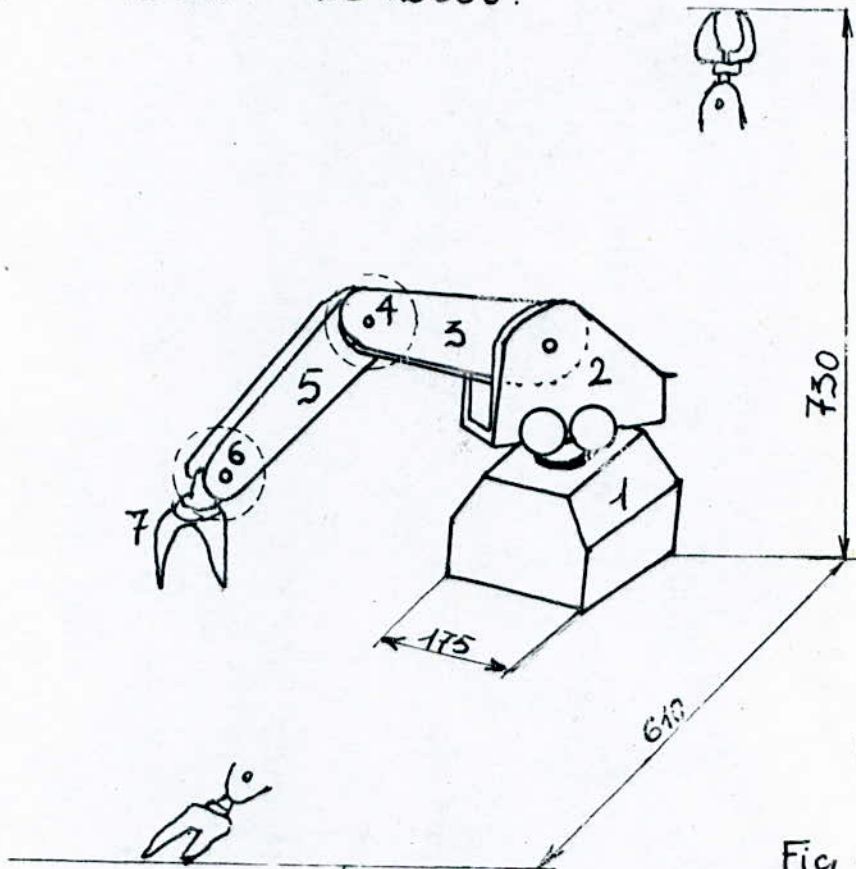


Fig 1

- | | | |
|------------------|-------------------------|--------|
| 1 - Base | - Portée en hauteur | 730 mm |
| 2 - Epau le | - Portée à l'horizontal | 610 " |
| 3 - Avant . bras | - Largeur de la base | 175 " |
| 4 - Coude | | |
| 5 - Bras | | |
| 6 - Poignet | | |
| 7 - Main | | |

I.2 Parties principales

Elles sont au nombre de cinq.

— La base : sert de support et comporte les circuits d'interface ainsi que le moteur qui assure la rotation de l'épaule.

— L'épaule : contient les cinq autres moteurs et leurs réducteurs.

— Le bras : se meut de bas en haut c'est à dire autour d'un axe horizontal. A la partie inférieure, des engrenages et des cables assurent le mouvement du coude, du poignet et de la main.

— L'avant bras : tourne autour d'un axe horizontal et porte des engrenages coniques assurant le mouvement du poignet.

— Le poignet et la main : le poignet se tourne autour d'un axe horizontal et tourne aussi autour de l'axe de la main.

Ainsi d'après la géométrie et l'espace balayé en se mouvant, On classe notre robot parmi les robots à coordonnées de révolution.

Ce sont les robots les plus employés dans l'industrie. Ils possèdent un bras articulé très semblable à celui de l'homme. En plus de l'articulation du poignet, il tourne à l'épaule et au coude.

Sa zone de travail est plus étendue que celle des autres type de robots comme ceux à coordonnées polaires, à coordonnées rectangulaires et à coordonnées cylindriques. Il peut contourner les obstacles et le bras se plie sur lui même avec plus de souplesse.

Robot à coordonnée de révolution.

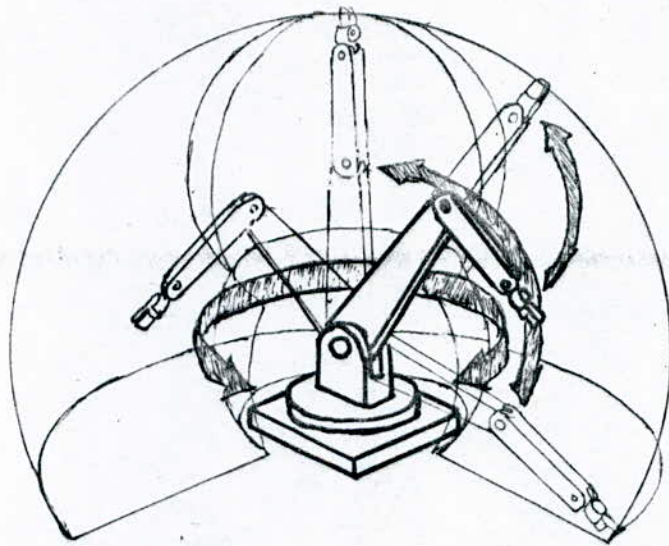


Fig 2

I-3 Remarque

Pour la machine comme pour l'homme, les mouvements du bras ont leur importance mais c'est la main (ou l'organe terminal de préhension) qui compte le plus. On constate une grande ressemblance entre le RB4/EV et l'Arm droid de Colne Robotics qui est un engin à cinq axes de mouvement et ayant une pince à trois doigts.

1.4 Les moteurs et leurs pas.

Le robot est équipé de six moteurs électriques pas à pas alimentés par courant continu de 12 V.

Chaque pas de moteur fait un angle de rotation de 7,5 degrés. Des réducteurs sont nécessaires pour commander les différentes parties du mini-robot.

- Pour la base et le poignet, on a 2 réducteurs dont les rapports sont

$$\frac{20}{72} \cdot \frac{12}{108} \quad (1)$$

- Pour l'épaule et le coude, les réducteurs ont les rapports

$$\frac{14}{72} \cdot \frac{12}{108} \quad (2)$$

- Pour la main,

On a les réducteurs (1) ainsi qu'une poulie de 26 mm de diamètre.

On remarque qu'il y a un réducteur identique à toute les parties du bras.

L'angle maximum réalisé par chaque doigt de la pince est de 50 degrés et sa course est de 20 mm.

	Base	Epaule	Coude	Poignet	Main
Degré par pas	0,2314	0,162	0,162	0,2314	0,0655
Pas par degré	4,32152	6,17284	6,17284	4,32152	15,2672
course par pas					0,0262

Fig 3

Chapitre II

La commande électronique.

II - 1 Le mini-ordinateur mod MPZ 80/EV

Le système mod MPZ 80/EV, fondé sur le microprocesseur Z 80 a été conçu et développé par Elettronica Veneta pour répondre aux exigences de l'enseignement moderne, pour garantir le maximum de diversification et possibilité d'extension du système. L'unité centrale se fonde sur une architecture avec bus grâce à des modules sur circuits imprimés au format Simple Europe. C'est cette modularité qui le rend expansible et par conséquent approprié aux plus diverses applications.

En ce qui concerne le logiciel, il utilise le système opérationnel CP/M qui est devenu un standard de l'industrie et supporte une large gamme de langages de programmes d'application. Sa souplesse d'emploi fait du MPZ-80/EV un instrument didactique puissant.

II - 2 Branchement du mini-robot au MPZ-80/EV

Le robot se branche au mini-ordinateur MPZ-80/EV au moyen de deux portes parallèles d'un octet, 8 bits. Ce sont les lignes de commande des moteurs et de renseignement sur l'état des capteurs. Voir Fig 5.

La porte A est porte de sortie. C'est le chemin par lequel on envoie les données pour mouvoir le robot.

Elle est constituée par 8 bits dont 4 d'entre eux numérotés (A5, A6, A7, A8) commandent les 4 phases des moteurs. 3 bits d'adresse (A2, A3, A4) pour adresser les moteurs et 1 bit pour valider les latches des données.

La porte B est porte d'entrée. C'est le chemin par lequel on reçoit les informations sur la position des ailes (photo-coupleurs et micro interrupteur). 2 bits d'entre eux sont libres et c'est à l'utilisateur d'utiliser d'autres capteurs dans les cas nécessaires. voir (fig 4)

la porte A

A8	phase 4
A7	Phase 3
A6	Phase 2
A5	Phase 1
A4	Adresse
A3	Adresse
A2	Adresse
A1	Validité

La porte B

B8	Photo-coupleur main
B7	Photo-coupleur poignet 2
B6	Photo-coupleur poignet 1
B5	Photo-coupleur coude
B4	Photo-coupleur épaule
B3	Micro-interrupteur base
B2	libre
B1	libre

Fig 4

Les portes parallèles A et B.

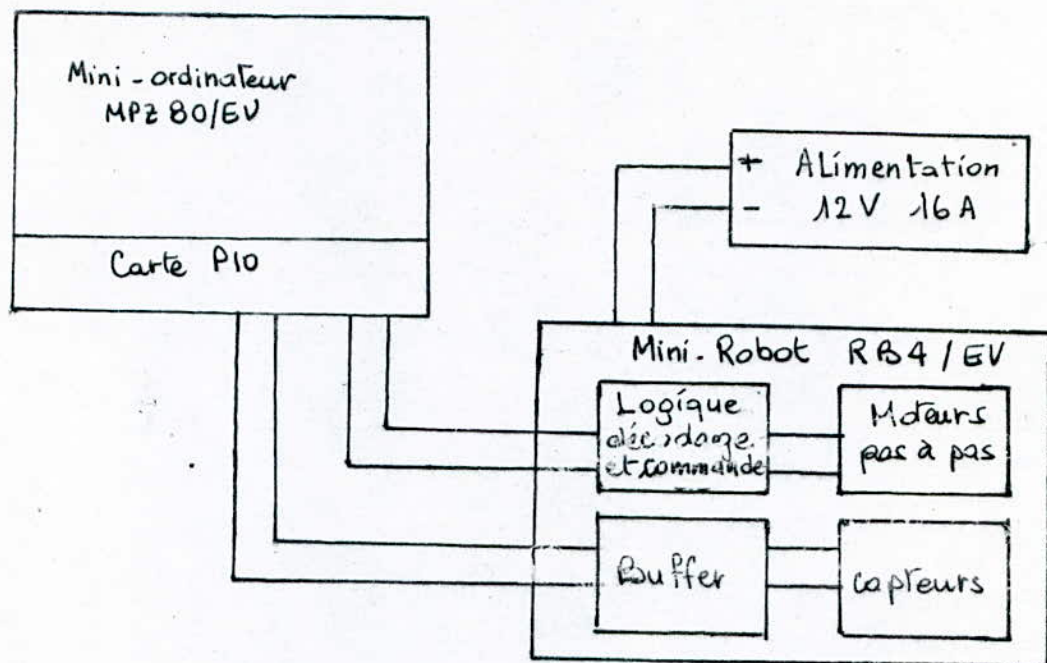


Fig 5

Branchement du mini-robot

Ainsi, la porte A et la porte B constitue l'interface parallèle E/S. C'est à dire que les informations en provenance, ou à destination du microsysteme sont transmises en même temps sous forme de « mots binaires ». Par exemple pour transmettre un mot de 8 bits ou 1 octet il faut disposer de 8 fils chacun véhiculant 1 bit (information logique 0 et 1).

II-3 Partie électronique.

Elle est disposée physiquement en trois cartes distinctes.

- Carte d'interface avec le connecteur de G/S pour le branchement direct au mini-ordinateur MPZ-80/EV
- Carte base avec toute la logique de commande des moteurs et la partie de puissance.
- Carte de Support des photo-coupleurs (facultative) pour leur utilisation comme capteurs de zéro.

Se référer au schéma électrique complet Fig 6

II-3.1 Carte d'interface.

Elle sert de support au connecteur de E/S pour le branchement de la carte PIO (Porte A et B) du MPZ-80/EV
la porte A est préparée comme sortie au moyen du branchement opportun des signaux de strobe =

$$ASTB = HIGH (1)$$

$$BSTB = LOW (\phi)$$

alors que la porte B est préparée pour l'entrée en posant :

$$CP1 = LOW (\phi)$$

$$CP2 = LOW (\phi)$$

Tous les signaux de la porte A vont directement à la carte base.

Les signaux qui proviennent des capteurs sont élaborés au contraire par un circuit intégré CMOS 4069, pour fournir les seuils d'entrée appropriés et envoyés ensuite à la carte PIO par l'intermédiaire d'un buffer du type 74367.

I.3.2 Carte base

Elle contient la section de commande des six moteurs pas à pas du robot.

Les 8 bits de commande provenant de la porte A sont "bufferés" par les circuits intégrés IC1 et IC2.

Le buffer du bit A1 est toujours validé et sa sortie est envoyée à un monostable IC3 pour produire une impulsion d'horloge. De cette manière, le décodeur IC4 produit une impulsion de mémorisation qui dure environ 500 msec au latch de contrôle du moteur adressé IC5 à IC10.

A1 est branché à une résistance de pull-up (R1), afin que cette ligne soit toujours haute, à moins qu'elle ne soit commandée par le micro-ordinateur. Les buffers IC1 et IC2 sont validés par la sortie "bufferée" du bit 1, pour que les données soient envoyées à l'entrée seulement quand le bit 1 est bas.

Les données relatives à la commande des phases des moteurs bits A5, A6, A7 et A8 vont, en parallèle, à tous les latches. Seul celui qui est adressé par les 3 bits A2, A3 et A4 les transférera ensuite à la sortie, pour commander effectivement les 4 enroulements.

Ces signaux à la sortie des 6 latches sont envoyés à autant de circuits intégrés translateurs de niveau IC11, IC12, IC13, IC14, IC15 et IC16. Ici les entrées à 5 volts sont

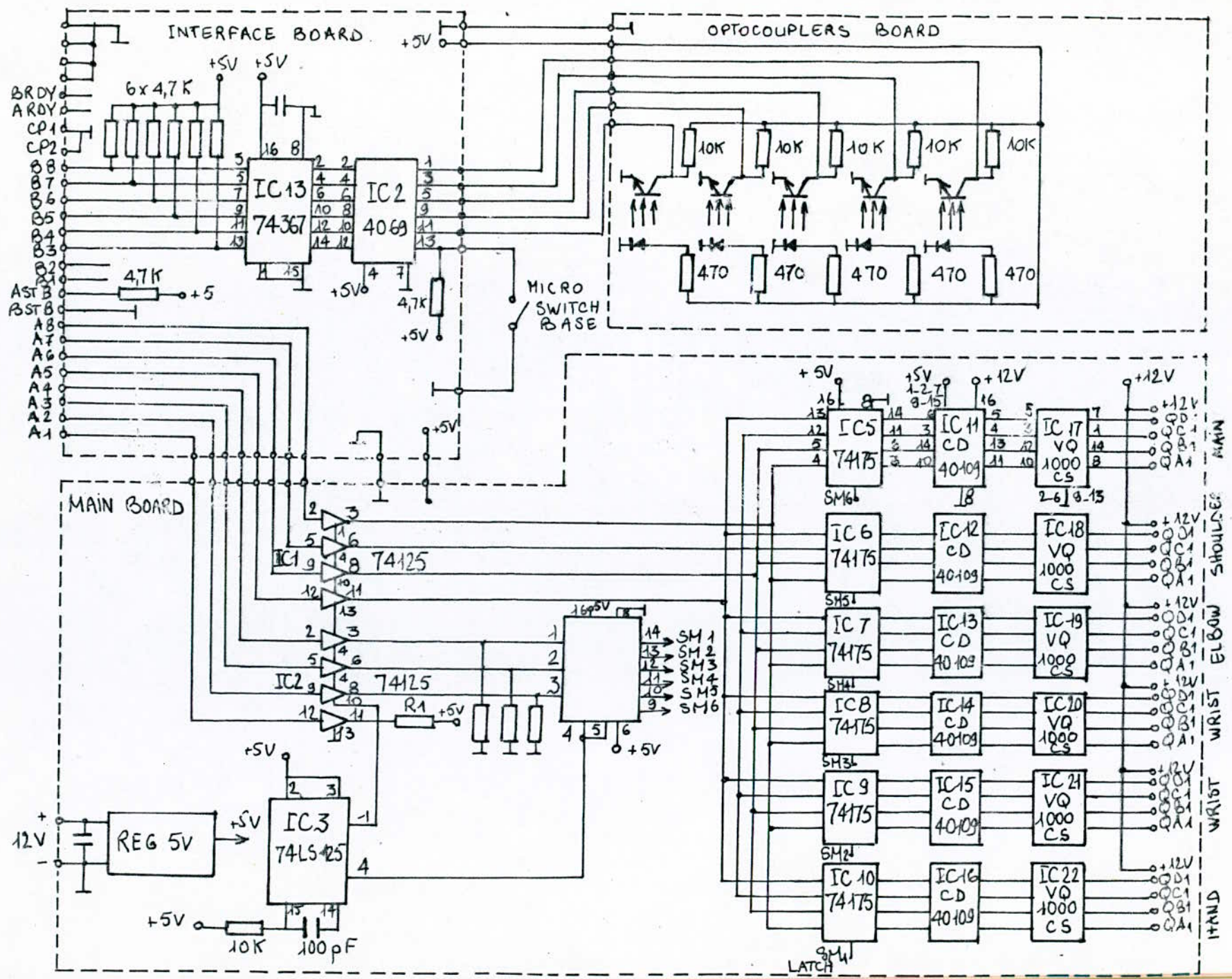
transformées en sorties à 12 volts capables de piloter les circuits de puissance IC 17, IC 18, IC 19, IC 20, IC 21 et IC 22. Ce sont des VFET quadruples à courant élevé qui produisent des sorties pour les enroulements du moteur. Le courant de pilotage pour les enroulements est égal à environ 300 mA et 12V

II 3.3 Carte photo-coupleurs.

Cette carte (facultative) supporte les photo-coupleurs qui ont la fonction de capteurs de la position zéro pour les mouvements de l'épaule, de l'avant bras, du poignet et de la main.

On utilise un micro-interrupteur branché directement à la carte d'interface, pour détecter la position zéro de la base.

4
6
24
26
3
25
1
2
8
7
10
8
12
11
14
13
23
6
21
19
18
20
17
18
16



II - 4 Commande des phases des moteurs.

Le pilotage des moteurs se fait par des signaux sur l'interface parallèle le mouvement est à demi pas ou à pas complet. Les tableaux ci après montrent les sequences qu'il faut envoyer sachant que les adresse A_8 , A_7 , A_6 et A_5 sont destinées à la commande des phases des moteurs.

Enroulement	A	B	C	D		
Bits	A_8	A_6	A_7	A_5	Pas	Donnée
	1	0	1	0	1	192
	1	0	0	0	1.5	128
	1	0	0	1	2	144
	0	0	0	1	2.5	16
	0	1	0	1	3	48
	0	1	0	0	3.5	32
	0	1	1	0	4	96
	0	0	1	0	4.5	64

Fig 9

Adresse	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1
binaire	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1

Fig 10

Le tableau fig 10 montre comment a été formé le nombre figurant aux données partant de la correspondance entre adresses et nombres binaires.

II. 5 Selection des moteurs

La combinaison entre les adresses A_4 , A_3 et A_2 permet de choisir le ou les moteurs à piloter parmi les 6.

Le tableau ci-après montre la correspondance entre adresses et moteurs.

A_4	A_3	A_2	Donnée	Moteur	N° moteur
0	0	1	2	Avant-bras	3
0	1	0	4	Poignet 2	4
0	1	1	6	Base	1
1	0	0	8	Main	6
1	0	1	10	Epaule	2
1	1	0	12	Poignet 1	5

CHAPITRE III

DESCRIPTION DES SYSTEMES AUTOMATISES SEQUENTIELS . Grafcet .

III.1 Grafcet de niveau 1,2

Lorsque l'on analyse un système automatisé, les actions qui permettent l'évolution du processus sont prévisibles, et peuvent être consignées dans l'ordre chronologique où elle vont se dérouler. A chaque action simultanée est associée une étape. Pour glisser d'une étape à l'autre, il faut qu'un événement se soit produit (fin d'une ou plusieurs action, ect.)

III 1.1 Grafcet de niveau 1.

Lorsque l'on aborde l'analyse et la description d'un système, on ne sait pas quelle technologie sera retenue pour les actionneurs, les capteurs et la commande. On décrira les actions et les événements en termes généraux. Par exemple (fig 3.2), On écrira : Avancer la pièce (étape 6) si la pièce a été avancée (et sa position contrôlée) Bider la pièce (étape 7) ect.

On trace ainsi un grafcet de niveau 1. Dans son graphisme élémentaire un grafcet comporte : (fig 3.1)

→ des étapes (carrés) repérés par un numéro tracé dans sa partie haute.

→ des étiquettes (rectangles) dans lesquelles sont notés les descriptions des actions à effectuer (en terme d'action : verbe à l'infinitif).

- des arcs qui relient les étapes de haut en bas.
- des transitions, traits horizontaux sur les axes auxquelles sont associées les receptivités, ou condition logique d'évolution.

III.1.2 Grafcet de niveau 2.

Le grafcet de niveau 2 s'établit après avoir déterminé :

- la technologie des actionneurs et leurs interfaces (mouvements élémentaires, commande ou bistable, possibilité d'arrêt sous énergie, etc),
- la technologie et le comportement des capteurs (action maintenue ou impulsionnelle).

En accord avec le grafcet de niveau 1, défini préalablement, on choisit la technologie. Le grafcet de niveau 2 transcrit ces choix. Exemple (fig 3.3) pour avancer la pièce on choisit un verin A, dont la tige sortira lorsque le piloté A+ de son distributeur est sollicité. La fin de course de sortie de tige de A est détectée par un capteur a₁.

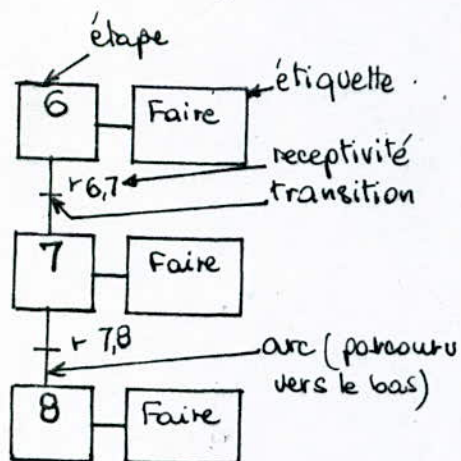


fig 3.1

Graphisme général du grafcet

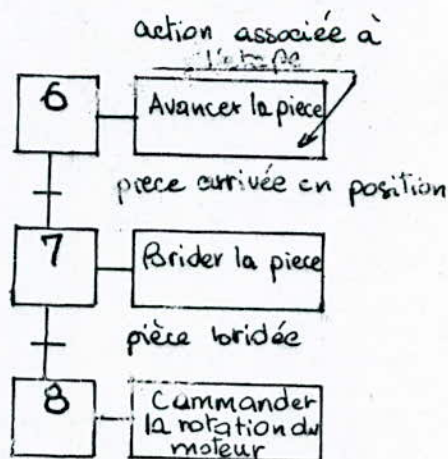


fig 3.2

Grafcet de niveau 1

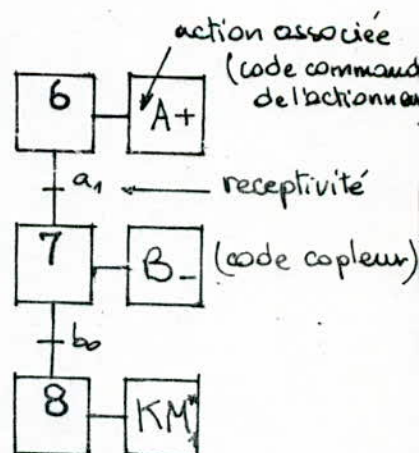


fig 3.3

Grafcet de niveau 2

III 2 Evolution du grafcet (fig 3.4)

Pour évoluer de l'étape 13 à l'étape 14, il faut franchir la transition $t_{13,14}$. Pour que la transition soit franchissable, il faut :

- que l'étape d'entrée (ou amont), soit active. On dira que la transition est validée.
- et que la réceptivité associée soit vraie (équation logique = 1)

Lorsque ces 2 conditions sont simultanément présentes, la transition est franchie, c'est à dire que :

- l'étape suivante ou aval est activée
- et en même temps, l'étape l'étape d'entrée (ou amont) est désactivée

Le graphe de la fig 3.5 montre que si la réceptivité associée à la transition 14,15, est vraie alors que l'étape 14 n'est pas active, cette transition n'est pas franchie.

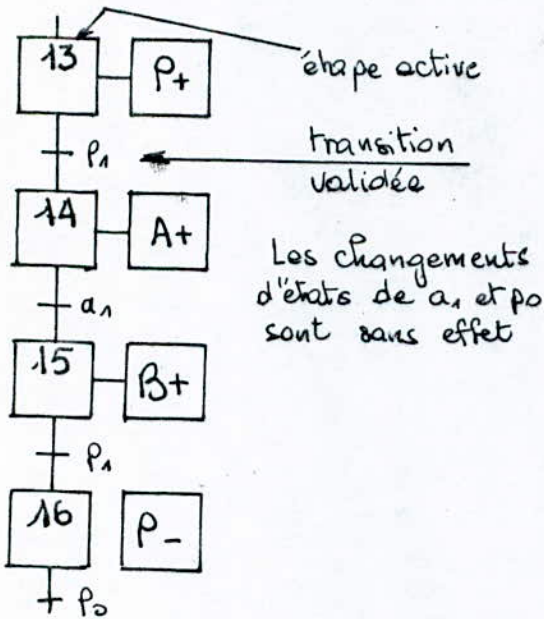


fig 3.4
Grafcet

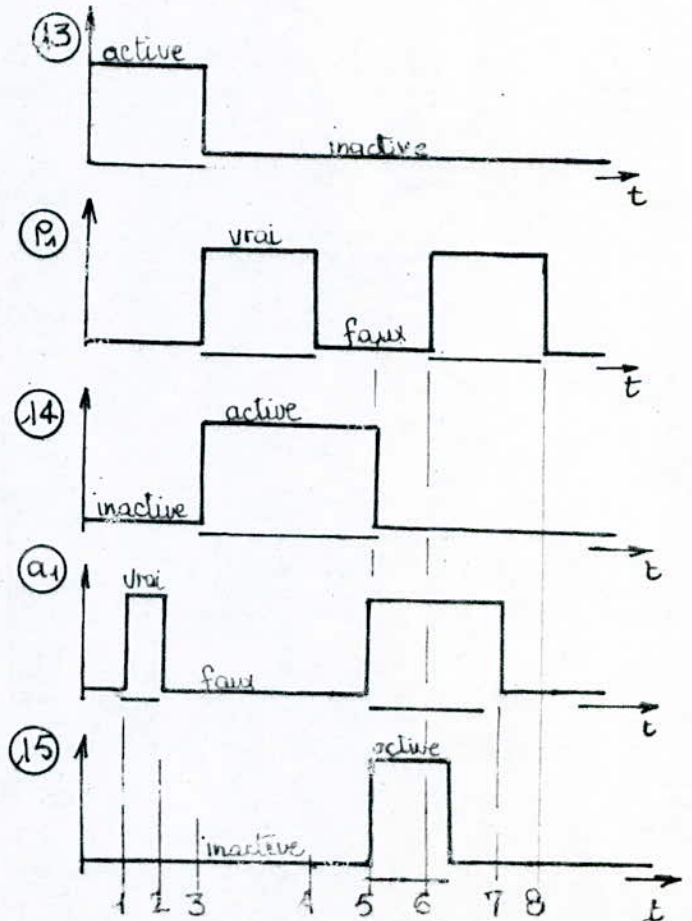


Diagramme de fonctionnement

fig 3.5

III 3. ACTIONS ASSOCIEES AUX ETAPES.

Dans le cas général, une action est associée à chaque étape, mais peut tout aussi bien être associée à plusieurs étapes d'un même graphe.

III 3.1 Actions inconditionnelles

Elles sont exécutées lorsque l'étape associée est active.

III 3.2 Action conditionnelles fig 3.6

Si, porte ouverte engager le chariot, est une action conditionnelle, elle est associée à l'étape 8, et ne sera effectivement réalisée que lorsque le constat de «porte ouverte» sera fait même si 8 est active bien avant.

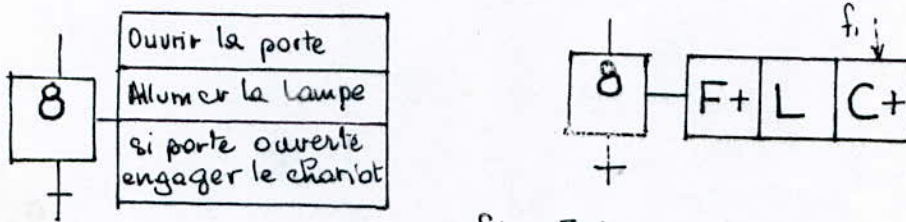


fig 3.6

Fonctions conditionnelles.

III 3.3 Actions impulsionnelles. fig 3.7

Une action impulsionnelle est signalée par le symbole * et sa durée. Par opposition à une action à niveau, une action impulsionnelle est réalisée dès que l'étape est active, mais sa durée est inférieure ou égale au temps d'activité de l'étape.

Par exemple une action L, pour maintenir une lampe allumée, est une action à niveau, mais L^* qui n'est que l'action d'allumer la lampe est une action impulsionnelle de durée limitée.

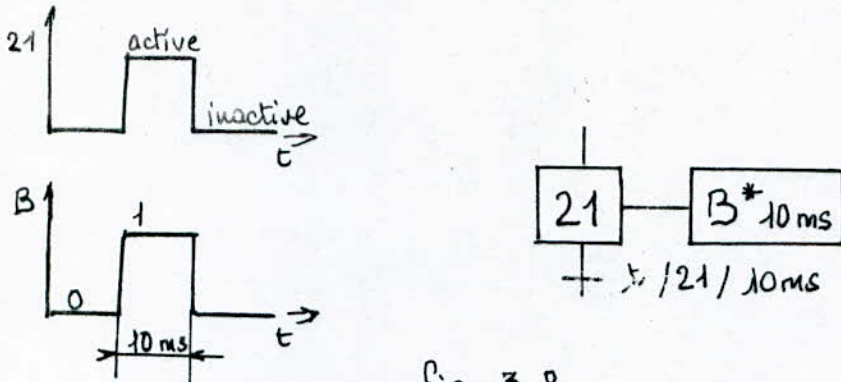


fig 3.8

Actions impulsionnelles.

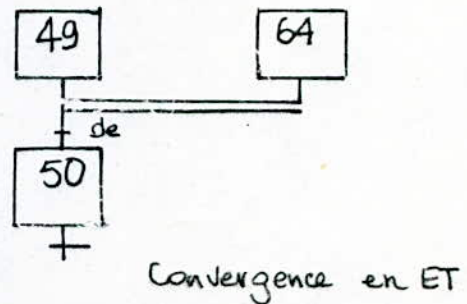
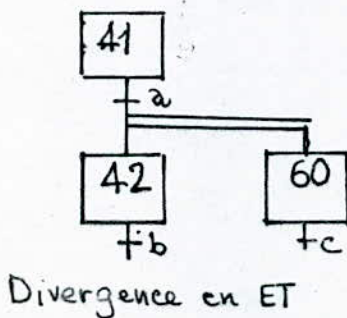
III 3.4 Temporisation.

La temporisation est une action impulsionnelle particulière. Comme elle, une temporisation est déclenchée avec l'activité de l'étape à laquelle elle est associée, et a une durée limitée, mais lorsque le temps est écoulé, la temporisation délivre un signal qui reste vrai tant que l'étape associée est active.

III 4. Convergence Divergence.

III 4.1 convergence en ET

Le graphisme 3.9 montre une transition unique, suivie d'un double trait horizontal. Lorsque cette transition est franchie, elle active simultanément toutes les étapes immédiatement en aval.



Sequences paralleles.
Fig 3.9

III 4.2 convergence en ET

Le graphisme est symétrique du précédent et se caractérise de même par une transition unique. Pour que cette transition soit franchissable, il faut que toutes les étapes immédiatement en amont soient actives et que la réceptivité associée soit vraie.

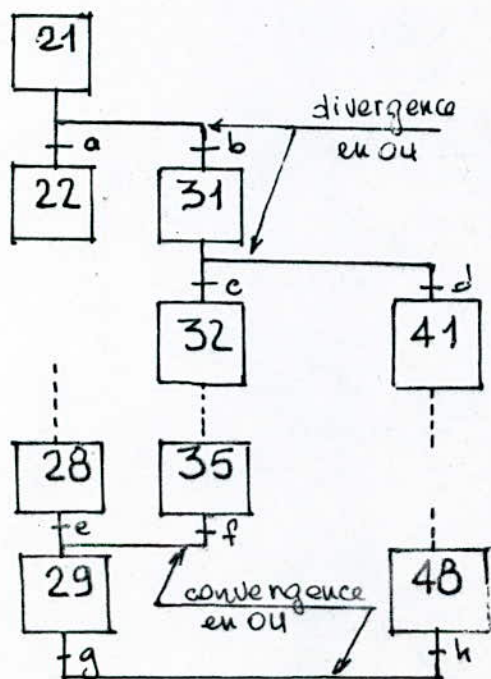
III 4.3 Divergence convergence en OU

- Divergence en OU (fig 3.10)

Le graphisme montre une étape suivie de plusieurs transitions. A l'issue de cette étape, on peut évoluer vers une branche ou l'autre. Ce sont les transitions qui graphiquement discriminent une évolution par rapport à l'autre.

- Convergence en OU (fig 3.10)

Le processus ayant évolué dans la branche sélectionnée, on converge vers une étape commune. On remarque que la convergence de toutes les branches ne se fait pas obligatoirement au même endroit.



La divergence en OU se comporte comme une divergence en ET

Fig 3.10
Aiguillage

Programmes de gestion du Robot avec le MPZ 80/EV

Le programme de gestion complète du robot avec le mini-ordinateur MPZ 80/EV est appelé MROBOT.COM.

Il permet de mouvoir le robot au moyen du clavier, d'apprendre une séquence de mouvement, de l'enregistrer sur disque, de la modifier ou de la faire répéter à l'échance.

Les routines de mouvement sont écrites assembleur Z80 qui est un langage qui contient toutes les instructions de l'assembleur 8080 plus d'autres instructions, c'est un langage plus récent et plus sophistiqué. La section de mouvement est écrite en assembleur Z80 pour la vitesse d'exécution. L'utilisation d'un autre langage évolué le basic par exemple aurait diminué considérablement la vitesse de mouvement.

Mais qu'on utilise ce paquet de routines dans le programme Basic qu'on nomme le Basic 80.

Pour rappeler le programme complet de gestion on compose les instructions suivantes :

A > MROBOT < Cr >

Ainsi le menu principal de sélection des différentes fonctions apparaît sur l'écran de visualisation.

IV.1 Utilisation du programme.

Une fois qu'on rappelle le menu, il visualise sur l'écran les différentes fonctions du programme numérotées à 7.

- 1 - Simulation libre de mouvement.
- 2 - Etude d'une séquence de mouvement.
- 3 - Mouvement sur Sequence.
- 4 - Edit d'une séquence
- 5 - Mise à zéro des mouvements effectués.
- 6 - Mise à zéro manuel.
- 7 - Fin.

1 - Simulation libre des mouvements :

Avec cette section du menu, on peut mouvoir notre robot au moyen du clavier ainsi en fait correspondre chaque axe à une certaine touche.

La touche contrôle (CTRL) nous permet de changer le sens de mouvement. Et si, on voudrait sortir de cette section, On tape sur la touche (+) ou la touche (-)

2 - Etude d'une séquence de mouvement.

On rappelle cette section chaque fois qu'on veuille exécuter ou enregistrer une certaine séquence qui est une chaîne de 6 caractères, chiffres ou lettres.

On donne un nom à la séquence et le programme se prépare pour accepter les pas de la séquence et leur mise en mémoire sur disque. Pour passer au pas qui suit on presse sur la touche (-) Pour fermer la séquence de mouvement, on appuie sur la touche (+)

Le programme exécute le mouvement simultané des axes lorsqu'on rappelle le mouvement sur séquence.

3. Mouvement sur séquence.

Il permet d'exécuter une séquence enregistrée par le menu n° 2 et cela en l'appelant par son nom. d'autant qu'il faut spécifier le mode de mouvement qu'il soit :

a) un mouvement continu = (C), sans arrêt.

b) un mouvement pas à pas : (P), mouvement avec arrêt après chaque pas de séquence et au moyen du clavier on continue le mouvement.

c) un mouvement simple : (S), une fois la séquence terminée, on retourne au menu.

d) un mouvement cyclique : (C), On reporte le robot à la position zéro à la fin de la séquence et on répète la même séquence.

Chaque type de mouvement peut être arrêté en appuyant sur la touche (U)

4. Edit d'une séquence.

Cette séquence permet le contrôle et la modification des données d'une séquence déjà enregistrée. Cela est très efficace car on a pas à établir une nouvelle séquence de mouvement mais d'en faire quelques modifications.

Remarque :

Il faut faire la distinction entre pas de séquence et pas de moteur.

Exemple :

Dans une séquence enregistrée précédemment on a les données suivantes :

Pas de la séquence.	Mot 1	Mot 2	Mot 3	Mot 4	Mot 5	Mot 6
3	100	0	20	-150	150	0

La séquence se fait en 3 pas . Le moteur de la base n° 1, fait 100 pas dans le sens compté positivement . Le moteur n° 2, de l'épaule n'exécute aucun pas dans cette séquence ainsi que le moteur n° 6, de la main demeure au repos. Le moteur n° 3 tourne de 20 pas dans le sens (+), c'est le moteur de l'avant-bras. Le moteur n° 4, du poignet 2 tourne de 150 pas dans le sens (-) quant au moteur n° 5, il tourne de 150 pas dans le sens (+).

A présent on a la séquence avec ses 3 pas et ses nombre de pas de moteur. Dans cette section de menu, On a 4 touches sur lesquelles il faut appuyer pour procéder ou non à une telle modification de la séquence enregistrée précédemment et qu'on a rappelée. Ce sont les touches S, M, F et Q

S : En pressant sur cette touche, On passe d'un pas de séquence à un autre sans procéder à une modification en supposant la visualisation examinée est exacte -

M : Avec cette touche, on modifie les pas de moteurs associés au pas de séquence examinée.

F : Au moyen de cette touche, On élimine tous les pas de séquence à partir du point visualisé.

Q : Cette touche permet de garder la séquence inchangée et retourne au menu.

5. Mise à zéro des mouvements effectués.

Ce présent programme met à zéro tous les mouvements exécutés ultérieurement.

6. Mise à zéro manuelle.

Cette section de menu permet de choisir arbitrairement la position zéro du robot et de la mettre manuellement sans mettre les moteurs en mouvement. elle sert la position zéro de tous les mouvements successifs.

IV.2 Composition du programme.

Le programme MROBOT-BAS est écrit en Basic (partie de gestion du menu principal et faisant appel à des sous programmes ou routines écrites en assembleur ce sont les routines base de mouvement du mini-robot. quant aux routines de service elle sont mises en œuvre au moyen du clavier elle sont aussi écrites en assembleur.

Donc le programme de gestion complet le MROBOT-COM se compose de 3 parties distinctes qui sont,

MROBOT-BAS : programme de gestion du menu principal en basic

ROBCOM.MAC : routines base de mouvement en assembleur.
 AROBOT.MAC : routines de service au moyen du clavier ecrite en assembleur.

Les sections en langage assembleur du programme se trouvent dans les points fixes de la memoire, Mais pour connaître les points d'entree exacts on travaille avec l'interprete Basic c'est le buffer qui doit echanger les informations avec les programmes Basic

On indique dans les figures ci-apres la carte de memoire Fig 4.1 et le buffer des donnees Fig 4.2

Systeme Operationnel	C000
Buffer des Donnees	A4FF
AROBOT	A300
ROBCOM	A000
Interprete BASIC + Programme MROBOT.BAS	9FFF

Fig 4.1

Carte de memoire des programme de gestion du Robot.

- On note a l'interieur de l'aire buffer des donnees les tableaux suivants qui sont au nombre de 9.

- FINDT** : Cette aire contient l'adresse des moteurs et doit être chargée au début du programme avant d'appeler les routines qui l'utilisent.
- FTABL** : contient les données de commande des phases des moteurs pour le mouvement à pas complet et doit être chargée au début du programme.
- POSAR** : contient les compteurs absolus des pas exécutés par chaque moteur ; chaque compteur occupe 2 octets pour une gamme de comptage de -32000 à $+32000$; elle se met à jour après chaque pas de mouvement.
- CTPOS** : contient, pour chaque moteur, le curseur au tableau **FTABL**, déterminant ainsi la position actuelle des phases.
- TBUF** : contient pour chaque moteur un compteur à 2 octets où sont mémorisés les pas à effectuer.
(elle est utilisée par la routine `command`)
- MOTBF** : contient les paramètres déterminant la vitesse de mouvement des axes, le mouvement de rotation des axes peut être différent d'un axe à un autre.
- CHARIM** : position de mémoire contenant l'information du caractère composé pendant le mouvement au moyen du clavier.
- MOTPIL** : position de mémoire contenant l'information relative aux

PACOM: position de memoire contenant le mot de comande du moteur en question.

				Donnée
TRAM	A4FF	moteur	6	8
			5	4
		Adresse moteur	4	12
			3	2
			2	10
			1	6
FINDT	A4FA	Sequence de mouvement des phases moteur		96
				48
				144
FTABL	A4F6	Moteur		182
		Total pas moteur		
POSAR	A4EA	Curseur à FTABL moteur		
CTPOS	A4E4	Partiel pas moteur		
TBLUF	A4D8	Parametre vitesse moteur		
MOTBF	A4D2	Caractère lu par console		
CHARIM	A4D1	Mot de commande des moteurs		
PACOM	A4D0	Numéro position de mémoire		
MOTPIL	A4CF			

Fig 4.2

Le programme AROBOT.MAC contient la routine de mouvement au moyen du clavier. cette routine lit le clavier en utilisant les fonctions base du système opérationnel CP/M qu'on utilise dans le mini-ordinateur MPZ 80/EV pour gérer les ressources du système.

En fonction de la touche pressée, l'axe correspondant se déplace dans un sens ou dans l'autre.

A ce fait, le tableau ci apres fig illustre la correspondance entre les touches leur code ASCII et les axes à mouvoir.

Touche	Code ASCII	Fonction
CTRL / Q	81	Rotation base en arriere
CTRL / Q	17	Rotation base en avant
CTRL / W	87	Rotation épaule vers le bas
CTRL / W	23	Rotation épaule vers le haut
CTRL / E	69	Rotation bras vers le haut
CTRL / E	5	Rotation bras vers le bas
CTRL / R	82	Rotation pince vers le haut
CTRL / R	18	Rotation pince vers le bas.
CTRL / T	84	Rotation pince sens anti-horaire
CTRL / T	20	Rotation pince sens horaire
CTRL / Y	89	fermeture pince
CTRL / Y	25	Ouverture pince.
CTRL / +	43	Sortie de la routine
CTRL / -	45	Sortie de la routine.

Le programme AROBOT.MAC est rappelé par l'instruction

A > M80 AROBOT, AROBOT = AROBOT/Z

Le point d'entrée de la routine est à l'adresse A300

On dresse la liste complète du programme AROBOT.MAC, ensuite on illustre le programme de la routine MOVIM, avec leurs adresses codes et mnémoniques.

Mouvement par clavier

0000	ASEG		
	ORG 0A300H		
	EXTRN	MOTUP, MOTDW, CONUP, CONDW	
	PUBLIC	MOVIM	
	MOVIM	routine base.	
	MOTUP	routine ;	1 pas vers le haut
	MOTDW	routine ;	1 pas vers le bas
	CONUP	routine ;	augmentation du compteur
	CONDW	routine ;	diminution du compteur.
A4FF	TRAM EQU	0A4FFH	; Sommet de la aire des donnée
A4D1	CHARIN EQU	TRAM-46	; Caractère de console du buffer
A4CF	MOTPIL EQU	TRAM-48	; moteur à piloter du buffer.
0005	BOOS EQU	0005H	; CP/M entry - point
0051	M1UP EQU	81	; code ASCII pour Q
0011	M1DW EQU	17	; CTRL Q
0057	M2UP EQU	87	; W
0017	M2DW EQU	23	; CTRL W
0045	M3UP EQU	69	; E
0005	M3DW EQU	5	; CTRL E
0052	M4UP EQU	82	; R
0012	M4DW EQU	18	; CTRL R
0054	M5UP EQU	84	; T
0014	M5DW EQU	20	; CTRL T
0059	M6UP EQU	89	; Y
0019	M6DW EQU	25	; CTRL Y
0020	MINUS EQU	45	; -
0028	PLUS EQU	43	; +


```

;
;*****
;*****+ ROUTINE MOVIM *****
;MOVEMENT BY KEY-BOARD
;

```

```

A300 00          MOVIM: NOP          ;
A301 0E 06      START: LD C,06H      ;
A303 1E FF      LD E,0FFH          ;
A305 CD 0005    CALL B00S          ;console input
A308 FE 00      CP C0H            ;
A30A 2B F5      JR Z,START        ;jump if no input
A30C 32 A4D1    LE (CHARIN),A     ;save character
;
A30F FE 51      CP M1UP          ;Key Q : forward base rotation ?
A311 20 0E      JR NZ,CON1       ;jump if not
A313 3E 01      LD A,01H        ;
A315 32 A4CF    LD (MOTPIL),A    ;select motor number 1
A318 CD 0000*   CALL MOTUP          ;move motor
A31B CD 0000*   CALL CONUP        ;increment counter
A31E C3 A301    JP START         ;jump to loop
;
A321 FE 11      CON1: CP M1DW     ;Key CTRL/Q: backward base rot. ?
A323 20 0E      JR NZ,CON2       ;
A325 3E 01      LD A,01H        ;
A327 32 A4CF    LD (MOTPIL),A    ;
A32A CD 0000*   CALL MOTDW     ;
A32D CD 0000*   CALL CONDW     ;
A330 C3 A301    JP START         ;
;
A333 FE 57      CON2: CP M2UP     ;Key W : shoulder down rotation ?
A335 20 0E      JR NZ,CON3       ;
A337 3E 02      LD A,02H        ;
A339 32 A4CF    LD (MOTPIL),A    ;
A33C CD 0000*   CALL MOTUP     ;
A33F CD 0000*   CALL CONUP     ;
A342 C3 A301    JP START         ;
;
A345 FE 17      CON3: CP M2DW     ;Key CTRL/W: shoulder up rotation ?
A347 20 0E      JR NZ,CON4       ;
A349 3E 02      LD A,02H        ;
A34B 32 A4CF    LD (MOTPIL),A    ;
A34E CD 0000*   CALL MOTDW     ;
A351 CD 0000*   CALL CONDW     ;
A354 C3 A301    JP START         ;
;
A357 FE 45      CON4: CP M3UP     ;Key E: arm up rotation ?
A359 20 0E      JR NZ,CON5       ;
A35B 3E 03      LD A,03H        ;
A35D 32 A4CF    LD (MOTPIL),A    ;
A360 CD 0000*   CALL MOTUP     ;
A363 CD 0000*   CALL CON4P     ;

```

```

A36B 20 0E          JR NZ,CON6      ;
A36D 3E 03          LD A,03H        ;
A36F 32 A4CF        LD (MOTPIL),A  ;
A372 CD 0000*      CALL MOTDW      ;
A375 CD 0000*      CALL CONDW      ;
A378 C3 A301        JP START        ;
A37B FE 52          CON6: CP M4UP    ;Key R : hand up rotation ?
A37D 20 19          JR NZ,CON7      ;
A37F 3E 04          LD A,04H        ;
A381 32 A4CF        LD (MOTPIL),A  ;
A384 CD 0000*      CALL MOTUP      ;motor 4 up
A387 CD 0000*      CALL CONUP      ;
A38A 3E 05          LD A,05H        ;
A38C 32 A4CF        LD (MOTPIL),A  ;
A38F CD 0000*      CALL MOTDW      ;motor 5 down
A392 CD 0000*      CALL CONDW      ;
A395 C3 A301        JP START        ;

A398 FE 12          CON7: CP M4DW    ;Key CTRL/R : hand down rotation ?
A39A 20 19          JR NZ,CON8      ;
A39C 3E 04          LD A,04H        ;
A39E 32 A4CF        LD (MOTPIL),A  ;
A3A1 CD 0000*      CALL MOTDW      ;motor 4 down
A3A4 CD 0000*      CALL CONDW      ;
A3A7 3E 05          LD A,05H        ;
A3A9 32 A4CF        LD (MOTPIL),A  ;
A3AC CD 0000*      CALL MOTUP      ;motor 5 up
A3AF CD 0000*      CALL CONUP      ;
A3B2 C3 A301        JP START        ;

A3B5 FE 54          CON8: CP M5UP    ;Key T : hand counter-clockwise?
A3B7 20 19          JR NZ,CON9      ;
A3B9 3E 04          LD A,04H        ;
A3BB 32 A4CF        LD (MOTPIL),A  ;
A3BE CD 0000*      CALL MOTUP      ;motor 4 up
A3C1 CD 0000*      CALL CONUP      ;
A3C4 3E 05          LD A,05H        ;
A3C6 32 A4CF        LD (MOTPIL),A  ;
A3C9 CD 0000*      CALL MOTUP      ;motor 5 up
A3CC CD 0000*      CALL CONUP      ;
A3CF C3 A301        JP START        ;

A3D2 FE 14          CON9: CP M5DW    ;Key CTRL/T : hand clockwise ?
A3D4 20 19          JR NZ,CON10     ;
A3D6 3E 04          LD A,04H        ;
A3D8 32 A4CF        LD (MOTPIL),A  ;
A3DB CD 0000*      CALL MOTDW      ;motor 4 down
A3DE CD 0000*      CALL CONDW      ;
A3E1 3E 05          LD A,05H        ;
A3E3 32 A4CF        LD (MOTPIL),A  ;
A3E6 CD 0000*      CALL MOTDW      ;motor 5 up
A3E9 CD 0000*      CALL CONDW      ;
A3ED C3 A301        JP START        ;

```



```

A3F3 3E 06          LD A,06H          ;
A3F5 32 A4CF       LD (HOTPIL),A    ;
A3FB CD 0000*     CALL HOTUP       ;
A3FB CD 0000*     CALL CONUP      ;
A3FE C3 A301      JP START        ;

```

```

A401 FE 19        CON11: CP M6DW          ;Key CTRL/Y : hand open ?
A403 20 0E        JR NZ,CON12         ;
A405 3E 06        LD A,06H          ;
A407 32 A4CF       LD (HOTPIL),A    ;
A40A CD 0000*     CALL HOTDN       ;
A40B CD 0000*     CALL CONDN      ;
A410 C3 A301      JP START        ;

```

```

A413 FE 2D        CON12: CP MINUS        ;Key - : go out ?
A415 20 02        JR NZ,CON13         ;
A417 1E 05        JR CON199         ;

```

```

A419 FE 2B        CON13: CP PLUS          ;key + : go out ?
A41B C2 A301      JP NZ,START        ;
A41E C9          CON99: RET              ;
                     ENI

```

Macros:

Symbols:

0005	BDDS	A4D1	CHARIM	A321	CON1
A3EF	CON10	A401	CON11	A413	CON12
A419	CON13	A333	CON2	A345	CON3
A357	CON4	A369	CON5	A37B	CON6
A398	CON7	A385	CON8	A3D2	CON9
A41E	CON99	A40E*	CONDW	A3FC*	CONUP
0011	M1DW	0051	M1UP	0017	M2DW
0057	M2UP	0005	M3DW	0045	M3UP
0012	M4DW	0052	M4UP	0014	M5DW
0054	M5UP	0019	M6DW	0059	M6UP
0	MINUS	A40B*	MOTDW	A4CF	MOTPIL
A3F9*	MOTUP	A3001	MOVIM	002B	PLUS
1	START	A4FF	TRAM		

No Fatal error(s)

ROBCOM.MAC, contient les routines base de mouvement du robot, sont générales et peuvent être utilisées dans d'autres programmes. On les illustre une à une ce sont :

MOTUP : (Point d'entrée = A000)

Déplace en avant d'un pas le moteur déterminé par le contenu de la position MOTFIL

MOTDW : (Point d'entrée = A02C)

Déplace en arrière d'un pas le moteur déterminé par le contenu de la position MOTFIL.

COMUP : (Point d'entrée = A0EC)

Augmente les compteurs partiels et absolus (Tableaux TBUF et POSAR) du moteur déterminé par le contenu de la position MOTFIL.

CONDW : (Point d'entrée = A109)

Diminue les compteurs partiels et absolus (Tableau TBUF et POSAR) du moteur déterminé par le contenu de la position MOTFIL.

COMMAND: (Point d'entrée = A09F)

Commande le mouvement simultané de tous les moteurs avec le nombre de pas spécifié par le contenu du tableau TBUF.

RESTOR: (Point d'entrée = A058)

Reporte tous les axes du robot à zéro; prend les données du tableau POSAR qui contient le nombre de pas effectués et commande les moteurs jusqu'à la mise à zéro de ce tableau.

Le programme ROBCOM.MAC est appelé par l'instruction : $A > M80 \text{ ROBCOM}, \text{ROBCOM} = \text{ROBCOM}/2$

La liste complète est indiquée ci-après ainsi que les routines qui composent ce programme.


```

;
;-----
;====  MINI-ROBOT  GENERAL PURPOSE ROUTINES  =====
;====  used by all programs  =====
;-----
;
;

```

0000

```

ASEG
ORG 0A000H
;
PUBLIC MOTUP,MOTDN,RESTOR,COMAND,CONUP,CONDW
;
;

```

- ```

; MOTUP Motor one step up
; MOTDN Motor one step down
; RESTOR Restore zero-position
; COMAND Move all motors together
; CONUP Increment counter
; CONDN Decrement counter
;
;
;

```

A4FF  
A4FA  
A4F6  
A4EA  
A4E4  
A4DB  
A4D2  
A4D0  
A4CF

- ```

;
;
; TRAM      EQU 0A4FFH      ;TOP of data area
; FINDT     EQU TRAM-5     ;Motor address table
; FTABL     EQU TRAM-9     ;Motor sequence table
; POSAR     EQU TRAM-21    ;Absolute counter table
; CTPOS     EQU TRAM-27    ;Motor sequence pointer table
; TBUF      EQU TRAM-39    ;Relative counter table
; MOTBF     EQU TRAM-45    ;Speed parameter table
; PACOM     EQU TRAM-47    ;Drive motor Buffer
; MOTPIL    EQU TRAM-48    ;Motor Buffer
;
;
;

```

0000

```

;
;
; POUT      EQU 00H        ;Port address
;
;
PAGE

```

Le programme MROBOT.BAS, au moyen de l'appel des routines bases gère complètement le robot.

le programme est écrit en langage BASIC, mais toutefois il doit trouver les routines qui résident en mémoire et sont écrites en assembleur.

Avec l'instruction A > MROBOT ;

On charge en mémoire les 2 paquets AROBOT et ROBCOM suivant leur fonction avec le programme link 80,

A > L 80 / P : A000, ROBCOM, AROBOT, MROBOT/E/N

Et avec l'instruction ; A > BASIC / M : 2H9FFF ,

on charge le langage BASIC avec la limite de mémoire max.

A ce moment on peut changer le disque pour réaliser le programme MROBOT : RUN "MROBOT BAS"

On décrit ainsi les fonctions exécutées par le programme.

Ligne :

2 - 100 : Définition des fonctions, dimensionnement des vecteur utilisés et attribution des paramètres de contrôle du vidéo ; voir appendice 1 pour plus de compréhension.

120 : Définition des points d'entrée des routines assembleur.

130 : Programmation des portes A et B de la carte P10

150 - 400 : Chargement des tableaux en MEV.

1000 - 1300 : Menu principal.

- 3000 - 3250 : Mouvement commandé par le clavier; appeler la routine MOVIM.
- 4000 - 4800 : Étude d'une séquence de mouvement. les données de chaque séquence sont mémorisées dans une archive appropriée RANDOM, dont le nom est spécifié par l'utilisateur. le nombre maximum de pas de séquence est limité à 50.
- 8000 - 8900 : Mouvement sur séquence. On demande le nom de la séquence enregistrée sur le disque. on contrôle si la séquence existe si oui, on procède au mouvement.
- 10000 - 10950: Edit des données d'une séquence.
- 16000 - 16110: Mise à zéro des mouvement effectués précédemment c'est la routine RESTOR.
- 17000 - 17350: Mise à zéro manuelle, on libère les moteurs en permettant le mouvement manuel, on met le tableau POSAR.
- 18000 - 35500: Routines de service.

```

2 REM ***** MROBOT.BAS *****
5 REM
30 DEF FNRV$(X$) = R$+" "+X$+" "+N$
40 DEF FNP$(XZ,YZ)=CHR$(13)+CHR$(27)+*"+CHR$(31+YZ)+CHR$(31+YZ)
80 DIM PAS$(12),PASZ(12),MATRX(50,12),INDZ(9)
90 DIM EDT!(10)
  00 GOSUB 35000
120 MOVIM=&HA300:RESTOR=&HA058:COMAND=&HA09F:'entry-point
  30 OUT(1),255:OUT(1),0:OUT(3),255:OUT(3),255 ' PIO Programming
150 DATA 4,10,2,12,4,8
160 DATA 192,144,48,96
165 DATA 15,22,22,18,18,20
190 TRAM=&HA4FF
300 REM Load RAM data
310 FOR I=TRAM-5 TO TRAM:READ AZ:POKE I,AZ:NEXT 'address
315 FOR I=TRAM-9 TO TRAM-6:READ AZ:POKE I,AZ:NEXT 'sequence
320 FOR I=TRAM-45 TO TRAM-40:READ AZ:POKE I,AZ:NEXT 'speed
400 GOSUB 18100:GOSUB 18120:GOSUB 18140 'zero position
999 REM ##### menu
1000 PRINT CLS$:STI$R$
1010 PRINT R$;TAB(9)"***** MINI-ROBOT UTILIZATION PROGRAM *****"
1015 PRINT R$;SPACE$(79)
1070 PRINT TAB(27)R$" MAIN MENU "N$
1080 PRINT
1090 PRINT TAB(10)FNRV$("1");R$;REI$;" KEY-BOARD SINGLE MOTIONS";TAB(60)N$STI$
1100 PRINT:PRINT TAB(10)FNRV$("2");R$;REI$;" MOTIONS SEQUENCE LEARNING";TAB(60)N$STI$
1070 PRINT:PRINT TAB(10)FNRV$("3");R$;REI$;" SEQUENCE EXECUTION";TAB(60)N$STI$
1080 PRINT:PRINT TAB(10)FNRV$("4");R$;REI$;" SEQUENCE EDIT ";TAB(60)N$STI$
1100 PRINT:PRINT TAB(10)FNRV$("5");R$;REI$;" PARTIAL RESET";TAB(60)N$STI$
1110 PRINT:PRINT TAB(10)FNRV$("6");R$;REI$;" MANUAL RESET";TAB(60)N$STI$
1120 PRINT :PRINT TAB(10)FNRV$("E");R$;REI$;" E N D ";TAB(40)N$STI$
1150 PRINT " ";STRING$(78,"=")
1160 PRINT FNP$(24,33);"> SELECTION : ";:FL=1:GOSUB 21100:IF IN$="" THEN 1160
1170 ON INSTR("123456E",IN$) GOTO 3000,4000,8000,10000,16000,17000,1300
1190 GOTO 1160
1300 PRINT CLS$:END

```



```

2999 REM ##### key-board motions
3000 PRINT CLS$
3020 PRINT TAB(20)R$*KEY-BOARD CONTROLLED MOTION "N$:PRINT LL$
3030 PRINT"The MINI-ROBOT is key-board controlled using the following keys"
3040 PRINT:PRINT
3050 GOSUB 18500
3110 PRINT FNP$(20,30)*"/+ = back to MAIN MENU"
3200 CALL MOVIM
3250 GOTO 1000
3999 REM ##### sequence learning
4000 PRINT CLS$
4010 PRINT TAB(15)R$*MOTIONS SEQUENCE LEARNING "N$:PRINT LL$
4020 CALL RESTOR : GOSUB 18100
4050 PRINT :PRINT "SEQUENCE NAME (CTRL/A = back to MAIN MENU) : ";:FL=6:GOSUB 2 10):IF CF=1 THEN 1000
4060 NOMEF$=IN$
4150 GOSUB 18400:GET 2,1:IF LOF(2)<=0 THEN 4250
4200 PRINT:PRINT"Sequence already stored, want to erase it S/N : ";:FL=1:GOSUB 2!100
4210 IF IN$="S" THEN 4250
4220 IF IN$="N" THEN CLOSE 2:GOTO 4000 ELSE 4200
4250 PRINT CLS$:PRINT TAB(17)R$*SEQUENCE LEARNING : << "NOMEF$;" >> "N$:PRINT LL$
4270 PRINT:PRINT"Sequences is MADE of steps produced moving the MINI ROBOT"
4280 PRINT"with the following keys :":PRINT
4300 GOSUB 18500
4320 PRINT:PRINT:PRINT
4330 PRINT" Press (-) al the end of every step":PRINT" Press (+) to stop learning"
4340 PRINT :PRINT LL$:PRINT
4400 NUMB=0
4410 PRINT
4420 NUMB=NUMB+1:FOR JJ=1 TO 16:PRINT BI$;:NEXT
4460 PRINT FNP$(23,60)*STEP NUMBER ";RIGHT$( " *STR$(NUMB),5);
4470 FOR I=TRAM-39 TO TRAM-28:POKE I,0:NEXT
4500 CALL MOVIM
4550 I1=0:FOR I=TRAM-39 TO TRAM-28:I1=I1+1:PASZ(I1)=PEEK(I):NEXT
4600 FOR I=1 TO 12:LSET PAS$(I)=MKI$(PASZ(I)):NEXT:PUT 2,NUMB+1
4650 IF PEEK(TRAM-46)=43 THEN 4800
4680 IF NUMB=50 THEN 4800 ELSE 4450
4800 LSET PAS$(1)=MKI$(NUMB):PUT 2,1:CLOSE 2:GOTO 1000

```

```

7999 REM ##### sequence execution
8000 PRINT CLS$
8020 PRINT TAB(23)R$"SEQUENCE AUTOMATIC EXECUTION"N$;PRINT LL$
8025 CALL RESTOR;GOSUB 18100
8030 PRINT
8100 PRINT "SEQUENCE NAME (CTRL/A = 'back to MAIN MENU) : ";FL=6;GOSUB 21100;IF CF=1 THEN 1000
8110 NOMEF$=IN$;GOSUB 18400;IF LOI(2)>0 THEN 8200
8120 PRINT TAB(46)"**** Non existing Sequence ****";CLOSE 2;GOTO 8100
8200 GET 2,1;NUMC=CVI(PAS$(1));IF NUMC<=0 OR NUMC>50 THEN 8120
8250 PRINT CLS$
8260 PRINT TAB(20)"PARAMETER CHOICE";PRINT LL$;PRINT;PRINT;PRINT
8270 PRINT"Continuous motion (C) or step by step (P) : ";FL=1;GOSUB 21100;COP$=IN$
8275 IF COP$<>"C" AND COP$<>"P" THEN PRINT;GOTO 8270
8290 PRINT"Single sequence (S) or continuous sequence (C) : ";FL=1;GOSUB 21100;MOS$=IN$
8295 IF MOS$<>"S" AND MOS$<>"C" THEN PRINT;GOTO 8290
8350 FOR J=1 TO NUMC
8360 GET 2,J+1
8380 FOR K=1 TO 12;MATRZ(J,K)=CVI(PAS$(K));NEXT
8400 NEXT
8500 PRINT CLS$
8510 PRINT"SEQUENCE EXECUTION << "NOMEF$;" >>";TAB(50)"Press U to go back to MAIN MENU"
8520 PRINT LL$
8530 FOR I=1 TO 10;PRINT;NEXT
8540 IF COP$="P" THEN PRINT"Press <RETURN> to go on after each step";
8650 FOR J=1 TO NUMC
8700 I1=0;FOR I=TRAN-39 TO TRAN-28;I1=I1+1;POKE I,MATRZ(J,I1);NEXT
8750 CALL COMAND
8755 M$=INKEY$;IF M$<>" " THEN IF M$="U" THEN 8900
8760 IF COP$="C" THEN 8780
8770 M$=INKEY$;IF M$="" THEN 8770 ELSE IF ASC(M$)<>13 THEN IF M$="U" THEN 8900 ELSE 8770
8780 M$=INKEY$;IF M$<>" " THEN IF M$="U" THEN 8900
8800 NEXT
8810 IF MOS$="S" THEN 8900
8820 CALL RESTOR
8825 IF COP$="C" THEN 8830
8827 M$=INKEY$;IF M$="" THEN 8827 ELSE IF ASC(M$)<>13 THEN IF M$="U" THEN 8900 ELSE 8827
8830 GOTO 8650
8900 PRINT CLS$;PRINT TAB(25)"*OTID: END";CALL RESTOR;GOSUB 18100;CLOSE 2;GOTO 1000

```



```

10000 REM ##### sequence data edit
10010 PRINT CLS$
10020 PRINT TAB(30)"SEQUENCE EDIT":PRINT LL$
10030 PRINT
10100 PRINT "SEQUENCE NAME (CTRL(A)=uscita) : ";:FL=6:GOSUB 21100:IF CF=1 THEN 1000
10110 NOMEF$=IN$:GOSUB 18400:IF LOF(2)>0 THEN 10200
10120 PRINT TAB(46)"**** Non existing sequence ****":CLOSE 2:GOTO 10100
10200 GET 2,1:NUMC=CVI(PAS$(1)): IF NUMC<=0 OR NUMC>200 THEN 10120
10250 PRINT CLS$:PRINT R$"SEQUECE EDIT << ";NOMEF$;" >> "N$:PRINT LL$:PRINT
10270 PRINT"Every motor amount of steps is reported for each sequence step"
10280 PRINT:PRINT"STEP  MOT.1  MOT.2  MOT.3  MOT.4  MOT.5  MOT.6":PRINT
10380 NUMM=0
10400 NUMM=NUMM+1:IF NUMM>NUMC THEN FOR J=1 TO 12:MATRX(1,J)=0:NEXT:GOTO 10450
10410 GET 2,NUMM+1
10420 FOR J=1 TO 12:MATRX(1,J)=CVI(PAS$(J)):NEXT
10450 FOR J=1 TO 6:I1=MATRX(1,(J-1)*2+2):I2=MATRX(1,(J-1)*2+1)
10460 IF I1<127 THEN M!=I1*256+I2 ELSE M!=-65536!-I1*256-I2
10470 EDT!(J)=M!
10480 NEXT
10550 PRINT:IN$=RIGHT$(" "+STR$(NUMM),4):IF NUMM>NUMC THEN MID$(IN$,1,1)="*
10555 PRINT IN$;
10560 FOR J=1 TO 6:PRINT RIGHT$(" "+STR$(EDT!(J)),8);:NEXT
10600 PRINT " O.K. (S/M/F/Q) : ";
10620 W$=INKEY$:IF W$="" THEN 10620 ELSE IF ASC(W$)<>13 THEN PRINT W$;
10630 IF W$="S" THEN 10760 ELSE IF W$="M" THEN 10660
10640 IF W$="Q" THEN 10950 ELSE IF W$="F" THEN 10820
10645 IF ASC(W$)<>13 THEN PRINT B!$;
10650 GOTO 10620
10660 FOR K=1 TO 67:PRINT B!$;:NEXT
10665 FOR J=1 TO 6:ED$=LEFT$(STR$(EDT!(J))+ " ",8):FL=8:GOSUB 21110
10670 EDT!(J)=VAL(IN$)
10680 NEXT
10700 FOR J=1 TO 6:IF EDT!(J)<0 THEN EDT!(J)=65536!+EDT!(J)
10720 IF EDT!(J)>255 THEN I1=FIX(EDT!(J)/256) ELSE I1=0
10725 I2=EDT!(J)-FIX(EDT!(J)/256)*256
10740 MATRX(1,(J-1)*2+1)=I2:MATRX(1,(J-1)*2+2)=I1
10750 NEXT
10760 FOR J=1 TO 12:LSET PAS$(J)=MKI$(MATRX(1,J)):NEXT:PUT 2,NUMM+1
10800 GOTO 10400
10820 GET 2,1:LSET PAS$(1)=MKI$(NUMM-1):PUT 2,1
10950 CLOSE 2:GOTO 1000
15999 REM ##### movement reset
16000 PRINT CLS$
16010 PRINT TAB(18)"EXECUTION MOTIONS RESET"
16020 PRINT LL$
16100 CALL RESTOR
16105 GOSUB 18100
16110 GOTO 1000
16999 REM ##### manual position
17000 PRINT CLS$
17030 PRINT TAB(20)R$" MANUAL POSITION RESET "N$
17040 PRINT:PRINT LL$:PRINT
17050 PRINT "Drive the MINI-ROBOT to the assumed zero position then press"
17060 PRINT "any key to go on"
17150 DATD=0
17160 FOR I=1 TO 6:IND=INDZ(I):GOSUB 18000:NEXT "rilascio motori

```



```

17999 REM -----
18000 OUT(0),DATO+IND+1:OUT(0),DATO+IND:OUT(0),DATO+IND+1:RETURN
18100 FOR I=TRAM-39 TO TRAM-28:POKE I,0:NEXT:RETURN
18120 FOR I=TRAM-21 TO TRAM-10:POKE I,0:NEXT:RETURN
18140 FOR I=TRAM-27 TO TRAM-22:POKE I,1:NEXT
18145 RETURN
18399 REM -----
18400 OPEN "R",2,NOMEF$,50
18410 I1=-2:FOR I=1 TO 12:I1=I1+2:FIELD 2,(I1)AS I1$,2 AS PAS$(I):NEXT
18450 RETURN
18499 REM -----
18500 PRINT"Q = BASE rotation clockwise";TAB(50)*CTRL(Q) = opposite movement"
18510 PRINT"W = SHOLDER lowering";TAB(50)*CTRL(W) = opposite movement"
18520 PRINT"E = ARM elevation";TAB(50)*CTRL(E) = opposite movement"
18530 PRINT"R = GRIPPER elevation";TAB(50)*CTRL(R) = opposite movement"
18540 PRINT"T = GRIPPER rotation anti clockwise";TAB(50)*CTRL(T) = opposite movement"
18550 PRINT"Y = GRIPPER closure";TAB(50)*CTRL(Y) = opposite movement"
18560 RETURN
18999 REM -----
21100 FOR K5=1 TO 8:Y$=INKEY$: NEXT :IN$=MID$(F$,1,ABS(FL)):WE=0: GOTO 21112
21110 WE=1:IN$=MID$(S$,1,ABS(FL)):MID$(IN$,1,LEN(IN$))=ED$
21112 CF=0:W1$=R$: IF FL>0 THEN W1$=""
21114 W=1: PRINT W1$:IN$: :WEEZ=ABS(FL): FOR K5=1 TO WEEZ: PRINT B1$: ; NEXT : GOTO 21116
21115 INPUT Z$
21116 W$=INKEY$: IF W$="" THEN 21116
21118 IF W>ABS(FL) THEN 21126 ELSE IF W$="" AND W$<="8" THEN 21140
21120 IF W$=CHR$(29) THEN IF W<=ABS(FL) THEN PRINT MID$(IN$,W,1):W=W+1: GOTO 21142
21122 IF W$=CHR$(9) THEN IF W<=ABS(FL) THEN MID$(IN$,W+1,ABS(FL)-W)=MID$(IN$,W,ABS(FL)-W):MID$(IN$,W,1)=" ": GOTO 21144
21124 IF W$=CHR$(4) THEN IF W<=ABS(FL) THEN MID$(IN$,W,ABS(FL)-W)=MID$(IN$,W+1,ABS(FL)-W):MID$(IN$,ABS(FL),1)=" ": GOTO 21144
21126 CF=INSTR(CC$,W$): IF CF<>0 THEN W$=CHR$(13): GOTO 21134
21128 IF W=1 THEN 21134 ELSE IF W$=CHR$(8) THEN PRINT B1$:W=W-1:GOTO 21116
21130 IF W$=CHR$(27) THEN PRINT MID$(B$,1,W-1): ON WE+1 GOTO 21100,21110
21132 IF W$=CHR$(28) THEN PRINT B1$:W=W-1: GOTO 21116
21134 IF W$<>CHR$(13) THEN 21116 ELSE IF WE=0 THEN IN$=MID$(IN$,1,W-1)
IF CF=0 AND W=1 THEN CF=5
21138 WEEZ=W-1:FOR K5=1 TO WEEZ:PRINT B1$:;NEXT:PRINT IN$:SPC(ABS(FL)-LEN(IN$));
21139 RETURN
21140 MID$(IN$,W,1)=W$:W=W+1: PRINT W$:
21142 IF ABS(FL)=1 THEN W$=CHR$(13): GOTO 21134 ELSE 21116
21144 PRINT MID$(B$,1,W-1);IN$:MID$(B$,1,ABS(FL)-W+1):; GOTO 21116
18999 REM -----
19000 N$=CHR$(27)+"GO" : R$=CHR$(27)+"64" : LL$=STRING$(78,"-") : LM$=STRING$(131,"-")
19010 EL$=CHR$(27)+"T" : ES$=CHR$(27)+"Y"
19020 ERROR$=R$+"ERROR"+N$
19030 FL$="." : F$=STRING$(80,FL$) : CF$=CHR$(2) : CS$=CHR$(1) : B1$=CHR$(8)
19040 B$=STRING$(80,B1$) : CC$=CS$+CF$+CHR$(30)+CHR$(31) : S$=STRING$(80,32)
19070 STI$=CHR$(27)+"(*:REI$=CHR$(27)+*)"
19080 CLS$=CHR$(27)+"+"
19500 RETURN

```


ROUTINES, FUNCTIONS and CONTROL PARAMETERS

ROUTINES:

21100..... INPUT ROUTINE

Entry-point 21100 : input data with no assigned string
 Input parameter : FL = field length

Entry-point 21110 : input data with assigned string
 Input parameter : FL = field length
 ED\$ = input string

Outputs: IN\$ = output string

CF = parameter = 0 or 5 out with key <return>
 = 1 out with key CTRL/A
 = 2 out with key CTRL/B
 = 3 out with key <up arrow>
 = 4 out with key <down arrow>

FUNCTIONS:

 FNRV\$(X\$) : put in reverse video the string X\$

FNP\$(XZ,YZ) : cursor position on screen
 XZ = line YZ = column

CONTROL PARAMETER:

 N\$ normal video
 R\$ reverse video
 STI\$ standard intensity
 REI\$ reduced intensity
 EL\$ erase line
 ES\$ erase screen
 CLS\$ clear screen

VARIABLES:

 FL\$ point "."
 F\$ 80 points
 B1\$ backspace
 B\$ 80 backspace
 S\$ 80 blank
 CF\$ chr\$(1)
 CO\$ chr\$(2)
 CC\$ CO\$+CF\$+chr\$(30)+chr\$(31)

Chapitre V

Utilisation du robot à la charge d'une machine-outil

V-1 Description de l'opération de chargement

Soit à charger une machine-outil exemple une perceuse transversale automatique à commande numérique, On dispose d'une pièce cylindrique de moyenne dimension dont le diamètre est $d = 26 \text{ mm}$.

la hauteur $h = 60 \text{ mm}$

et on desire percer cette pièce sur une longueur de 30 mm avec forêt de 10 mm de diamètre.

On décrit alors toutes les opérations de mouvement ou pas de séquence exécutés par le robot lors du chargement et déchargement de la machine-outil.

1- Déplacement du robot de la position zéro vers la pièce (pièce entre les pinces)

2- La saisie de la pièce. (fermeture des pinces)

On peut calculer approximativement la force de serrage des pinces, et cela suivant le poids de la pièce à soulever.

masse de la pièce : $m = V \cdot d$

$V =$ volume de la pièce $V = \frac{\pi d^2}{4} h$

$d =$ densité de l'acier. $d = 7,8$

$$m = \frac{1}{4} \pi \cdot (26)^2 \cdot 10^{-6} \cdot 10^{-3} \cdot 60 \cdot 7800 \approx 0,25 \text{ kg}$$

on applique alors une force inférieure ou égale à 20 daN

- 3 - Déplacement de la pièce vers les mors de la machine (point de fixation de la pièce pour être usinée).
- 4 - Emplacement de la pièce entre les mors
- 5,6 Retour à une position d'attente
- 7,8 Reprise de la pièce (serrage des pinces).
- 9 - Dépot de la pièce fin de l'opération
- 10 - Retour à la position zéro. x_0, y_0, z_0, T_0, U_0 et V_0

VI-2 Séquence de mouvement et pas de moteurs

PAS DE MOTEUR

pas de séquence	Moteur 1 Base	Moteur 2 Epaule	Moteur 3 avant bras	Moteur 4 Poignet 2	Moteur 5 Poignet 1	Moteur 6 Pinces..
1	x_1	y_1	z_1	T_1	u_1	v_1
2	0	0	0	0	0	v_2
3	x_3	y_3	z_3	T_3	u_3	v_3
4	x_4	0	0	T_4	0	0
5	0	0	0	0	0	$v_5 = -v_2$
6	x_6	0	0	0	0	0
7	$x_7 = -x_6$	0	0	$T_7 = -T_4$	0	0
8	0	0	0	0	0	$v_8 = -v_5$
9	x_9	y_9	z_9	T_9	u_9	v_9

Les vitesses de rotation des moteurs sont de telle sorte qu'il ait le même temps pour la rotation de tous les moteurs durant le pas de séquence.

On estime que les temps d'exécution des pas de séquence sont les suivants

$$T_1 = 2 \text{ s}$$

$$T_2 = 1 \text{ s}$$

$$T_3 = 2 \text{ s}$$

$$T_4 = 1 \text{ s}$$

$$T_5 = 1 \text{ s}$$

$$T_6 = 1 \text{ s}$$

$$T_7 = 1 \text{ s}$$

$$T_8 = 1 \text{ s}$$

$$T_9 = 2 \text{ s}$$

$$T_{10} = 3 \text{ s}$$

$$\sum T_i = 15 \text{ s}$$

- L'opération de perçage se situe entre les pas 6 et 7

elle comprend :

- le serrage de la pièce.

- mise en position de perçage

- perçage et retour en position initiale.

- desserrage se situe entre la phase 8 et 9.

L'opération de perçage ainsi citée dure approximativement 10 secondes

Ainsi toute l'opération de perçage avec manipulations du robots a une durée de 25 secondes.

V. 3 Tracé d'un Grefcet.

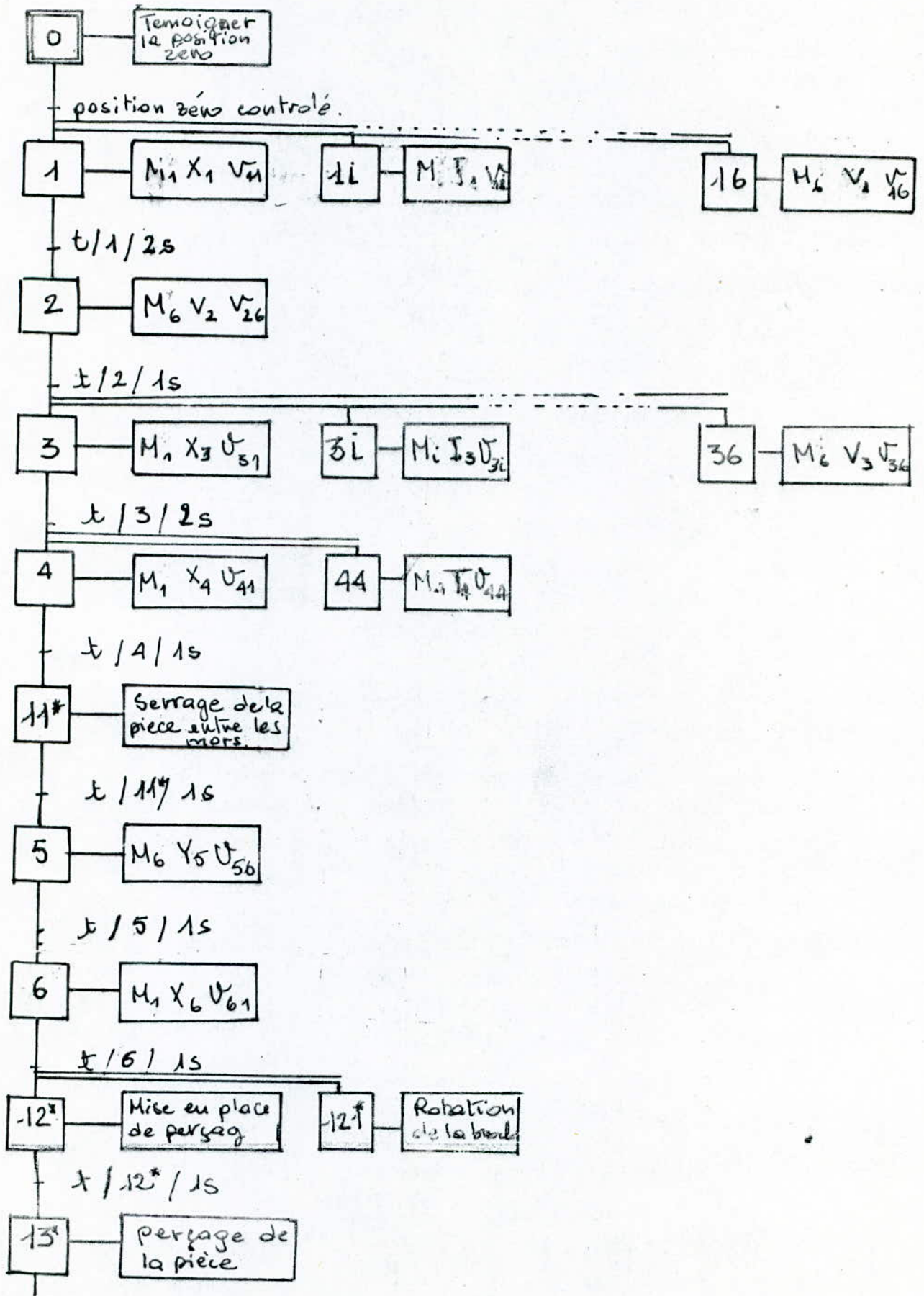
Dans l'étiquette de chaque étape, on mentionne par :

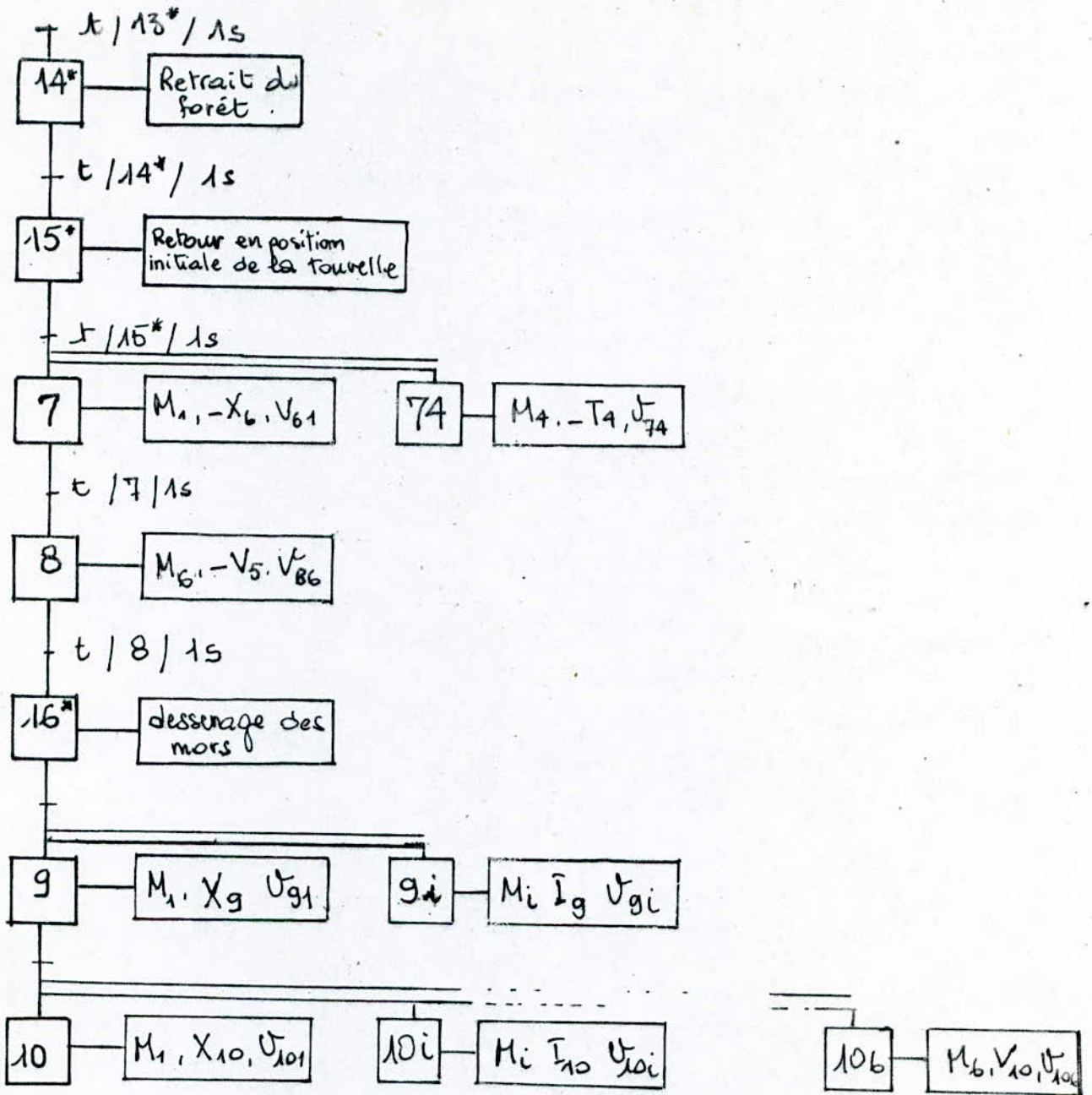
M_i le numéro de moteur qu'il faut piloter

X_i, Y_i, Z_i, T_i, U_i et V_i les pas de moteurs associés aux pas de séquences.

T_i Temps alloué à chaque opération.

V_i Vitesse de rotation des moteurs.





IV.4 Execution de la séquence au moyen du clavier.

Comme expliqué en page 30 c'est le programme AROBOT qui contient les routines de mouvement au moyen du clavier.

Pour l'exécution des mouvements de séquence, on doit rappeler le Programme AROBOT en formulant l'instruction

A > 80 AROBOT, AROBOT = AROBOT / Z

Pour réaliser le premier pas de séquence, on doit appuyer sur la touche.

- Q - on a rotation du moteur 1 en arrière -
- W - on a rotation de l'épaule vers le bas. (moteur 2)
- E - rotation de l'avant bras vers le haut (moteur 3)
- R - rotation de la pince vers le haut
- T - rotation de la pince dans le sens inverse des aiguilles d'une montre.
- Y - Permeture de la pince.

On peut utiliser chaque mouvement de moteur seul pour arrêter le mouvement on presse sur la touche U la touche controle permet d'inverser le sens de rotation. par exemple en appuyant sur la touche Q, on a rotation du moteur 1 en arrière on pressant sur la touche CTRL Q on a le sens inverse de la rotation.

si l'on veut enregistrer seulement la séquence on doit lui donner un nom de 6 caractères. et spécifier ensuite le mode de mouvement (pas à pas, continu, cyclique, simple)

une fois qu'on a enregistré sur disque tous les mouvements pas par pas c'est ce qu'on appelle mouvement sur séquence.

Ainsi rien qu'au moyen du clavier sans faire appel à une programmation manuelle et enregistrer les séquences de mouvement du bras-mécanique. et c'est à l'intérieur du micro-ordinateur que les routines en mémoire sont interpellées chaque fois qu'on appuie sur une touche.

Conclusion

Dans cette étude, on a eu l'occasion de connaître comment fonctionne un bras-mécanique ou manipulateur programmable au moyen d'un micro-ordinateur muni d'un microprocesseur le Z80.

Ainsi on a abordé en spécifiant les différentes manières de mouvoir le mini-robot. Il reste à l'utilisateur de l'exploiter en le livrant à diverses tâches. Certes, ce n'est pas un robot industriel proprement dit mais son rôle est didactique, mais toutefois le principe de fonctionnement demeure le même.

Il est à signaler que ce travail reste incomplet du fait d'absence de manipulations qui pourtant est le thème essentiel de ce sujet. Mais nous suggérons l'installation du mini-robot dans les très brefs délais pour permettre aux étudiants de "toucher" un domaine qui est la robotique et de les préparer à s'intégrer dans les complexes industrielles où tout a tendance à s'automatiser.

Bibliographie

1 - Documentation fournisseur

Edition Electronica Veneta 1986

Elle se compose de 8 volumes se rapportant essentiellement à l'utilisation du micro ordinateur MPZ-80/EV

1. a Manuel R₂ Mini-computer-trainer mod MPZ-80
Mini-robot RB4 /EV.1. b Manuel matériel du système base M₁1. c Notice de fonctionnement M₂1. d Manuel du système opérationnel CP/M M₃1. e Notice du programme Monitor MON-80 M₄1. f Notice de fonctionnement M₆1. g Manuel terminal video M₇1. h Unité d'alimentation M₈

1. i Unité d'Extension industrielle.

2. Projet de fin d'étude ENP

Mini-robot Pneumatique RB3/EV

Etudié par A. MEDJAHED et Dirigé par M^{me} ROBL

promotion : janvier 87

3. Automatique industrielle H Longeot

Edition Dunod 1985

4. Les systèmes microprogrammés automatisés

T Maurin Ed 1978

5. Encyclopedie le temps des robots

Bordas Ed 1986

6. Les Systèmes automatisés - C Bourbonne - J Cojean
Tome 1 et 2 Editions Foucher Paris 1982
7. Introduction aux circuits logique Létocha
édition 1985
- 8* - L Automatique par les problèmes. R chappert
L. Liberville. J-cojean
édition Paris 1983
- 9* - L'Automatisation des machines-outils
Ford PT éd Foucher 1966
- 10* - Automatisation industrielle HORNAUER
Edition 1959.
- 11 - Logique Combinatoire et Séquentielle
Lagasse Jean Dunod 1971
- 12 - Logique Combinatoire
Lagasse Jean Dunod 1976
- 13* - Logique binaire et Ordinateur
Tome 1: fonction logique et Arithmétique
binaire -
Michèle Amiax Masson 1977

(*) Documents non consultés.

