NUM 760

Utility Manual

En-938603/A

MOD. 05-85
ED. 02-85

www.num.com

num
CNC HighEnd Applications

WARNING TO THE USER

This manual has been drawn up in accordance with the standards referred to in it and in accordance with the requirements imposed by the design of NUM-SA systems.

However, in the light of current standards and of technical developments, NUM-SA reserves the right to modify, accordingly, all or part of this manual as it sees fit.

TABLE OF CONTENTS

1. - GENERAL

1.1. - DEFINITION OF THE UTILITIES

To program the PC, determine the machine parameters, calibrate the axes, etc... one or several utilities, each fulfilling one of the functions described, must be loaded in the memory.

1.1.1. - Utility supervisor : Utility 0 (200067)

Utility 0 manages the memory space with respect to loading or erasure of one or several utilities. Each utility can be loaded separately and is stored at the back of the memory with those that are already present. Utility 0 must always be loaded before the other utilities. If this is not the case, loading it will automatically erase all the utilities stored in the memory.

1.1.2. - Loading and display of axes compensations : Utility 2 (200069)

Utility 2 allows measurement and entry of compensations on different axes with respect to movements along the axes.

1.1.3. - Loading fixed macros : Utility 3 (200070)

Utility 3 allows entry of special workpiece programs in EEPROM memory, such as a machining cycle specific to the machine defined by the manufacturer or by the user.

1.1.4. - Loading and display of machine parameters : Utility 5 (200072)

Utility 5 allows input, modification and display of the different machine parameters (maximum speed, gain, etc...). Tape-reader parameter loading and outputting of these to tape are carried out by this utility.

1.1.5. - Programming the PC in assembler : Utility 6 (200073)

Utility 6 allows the PC to be programmed in assembler language suited to the numerical control. It is used to input the program through the tape-reader and unload it to a punched tape or printer, as well as display, modification or input via the control keyboard.

1.1.6. - Programming the PC by relay charts :
Utility 7 (200074)

Utility 7 is used to program the PC by relay charts via the numerical control keyboard. It also allows PC program input by the tape-reader and its output to the punched tape or printer.

1.1.7. - LPC programming in assembler :
Utility 10 (200350)

Utility 10 which is similar to utility 6, allows the LPC PCB to be programmed in assembler language suited to numerical control. It also provides program input by the tape-reader and its output to a punched tape or printer, as well as its display, modification and input on the numerical control keyboard.

1.1.8. - EEPROM filing :
Utility 11 (200353)

Utility 11 is used to safeguard binary information (data and softwares) stored in EEPROM memory.

1.1.9. - Option interlock :
Utility 12 (200353)

Utility 12 is used by the machine-tool manufacturer to personalize a machine before supplying it to his client.

1.1.10 - Utility loader :
Utility 13 (200371)

As some utilities are very large (Utility 7 for example), they have had to be segmented into several tapes. Utility 13 allows loading utilities whether they are segmented or not.

1.1.11. - Final site utility (200355)

It contains utility 0, utility 9 (allowing modifications to be made to machine parameters P16, P17 and P18 and all parameters to be displayed), and utility 11.

Using this utility is described in the operator manual.

1.1.2. Display text translation

The utilities display text is in French language. In this manual the text will be shown as displayed, with English translations in brackets, viz.

"GRAVURE EN COURS" ..... [WRITING IN PROGRESS]

1.2. - LOADING A UTILITY

With the tape on the tape-reader, pressing "CHARG" [LOAD] key situated on the front face of the NC processor in the electronic rack displays the following menu :

1	INIT CNC	CNC INIT
2	RAZ RAM	CLEAR RAM
3	CHARG LECT	TAPE READ
4	MODE TRANSP	TRANSP MODE
5	CHARGE DNC	DNC LOAD
6	VERI CHG LEC	TAPE READ CHECK
7	VERI CHG DNC	DNC LOAD CHECK
8	UTIL 1	UTIL 1
9	UTIL 2	UTIL 2

Note :

Lines 4 to 9 of the menu must not at present be selected.

- Selection of menu 1 (INIT. CNC) initializes the system with display change-over to page "current point".

- When starting up for the first time (equipment delivered to the manufacturer), a reset to zero of the RAM memories is recommended. To do this select 2 on the keypad; display of RAZ RAM (O/N) ? Answer by O, RAZ is finished after successive display of two OKs on the display. To return to the previous menu, press one of the numerical keys.

  If it is a commissioned equipment, RAZ RAM (choice 2) destroys the workpiece programs in the memory.

- Selecting 3 provokes the following question :
  "LECT. PRET ? (O/N) ..... [READER READY ? (YES/NO)]

  When the answer is O, the tape-reader starts up with rapid scrolling of the program utility on the display. At the end of read mode, message "CHARG OK" ..... [LOAD OK] is displayed. Return to the menu is achieved by pressing one of the numerical keys.

- This utility is loaded at the back of the memory. If the memory space is not sufficient, part of the workpiece programs in the memory is destroyed. If despite total destruction of the workpiece programs, the memory space is still insufficient, "RAM SPACE" is displayed when loading. Then either one or several of the utilities already loaded must be destroyed to allow a new utility to be loaded, or a memory PCB must be added and the available memory space thus increased. In the latter case, all utilities must be reloaded because the top end of the memory has been relocated by the PCB addition.

- The first utility to be loaded is utility O. Once it is in the memory, the program is called by "service" key. The following menu is displayed :

  O GESTION DES UTILITAIRES ..... [UTILITY MANAGEMENT]

Loading any other utility is achieved in the same way as previously, by choice 3 of menu "TAPE READ" If for example, utilities 5 and 6 are loaded, after pressing "service" key, the following is displayed :

```
0  GESTION DES UTILITAIRES .....     [UTILITY MANAGEMENT
5  PARAMETRES MACHINE .....          MACHINE PARAMETERS
6  PRO. AUTOMATE EN ASSEMBLEUR .....[PC PROGRAM IN ASSEMBLER]
```

If access to the machine parameters is required, it is sufficient to key in 5 (LF) and the menu of utility 5 appears.

The menu of the utilities present displays 7 titles maximum. If there are more than seven, press key "continued" to display the following titles.

1.3. - UTILITY MANAGEMENT

Utility management is processed by utility 0, keying 0 (LF) displays the following menu :

```
0  ABANDON DE TOUS LES UTILITAIRES .....    [ABANDON ALL UTILITIES
1  ABANDON SELECTIF D'UN UTILITAIRE .....   SELECTIVE ABANDON OF ONE UTILITY
2  IDENTIFICATION DES UTILITAIRES .....     UTILITY IDENTIFICATION]
```

- keying in 0 (LF) provokes a confirmation request on the dialog line :
  ABANDON DES UTILITAIRES ? (O/N) ..... [ABANDON ALL UTILITIES ? (Y/N)]

  Keying in 0 (LF) destroys all utilities and returns to "current point" page. (Memory restituted for workpiece programs).

If "service" key is pressed, "PAS DE PROGRAMMES UTILITAIRES PRESENTS DDANS LE SYSTEME" ..... ["NO UTILITY PROGRAMS PRESENT IN THE SYSTEM"] is displayed.

- Keying in 1 (LF) displays on the dialog line :

  NUMERO DE L'UTILITAIRE ? ..... [UTILITY NUMBER ?]

  Keying 5 (LF), for example destroys utility 5 only. The menu of the utilities present in the system then becomes :

```
  0   GESTION DES UTILITAIRES .....     [UTILITY MANAGEMENT
  6   PROG. AUTOMATE EN ASSEMBLEUR .....[PC PROGRAM IN ASSEMBLER]
```

- Keying 2 (LF) displays a new page with "IDENTIFICATION DES UTILITAIRES" ["UTILITY IDENTIFICATION"] title, giving the references of the utilities present (number, reference with indice) as well as the memory area which is still available.

Note :

Lines 4 to 9 of the menu must not be selected.

2. - UTILITY 2 : LOADING AND DISPLAY OF AXES COMPENSATIONS

This utility allows measurement compensations to be made to all axes of the machine (linear or rotary).

The maximum number of points which can be compensated is 85 in total for a 760T and 70 for a 760F. One axis can thus comprise 85 compensation points if the other axes are not compensated (for a 760T).

The point to be compensated and the compensation value are expressed in metric to the micron without decimal point. Maximum compensation is 256μ.

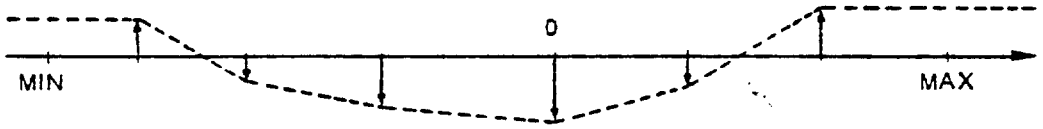
The compensation variation between two successive compensated points must not exceed 128 and the compensation is linear between these two points. There is no restriction on the distance between two compensated points.

A maximum and minimum compensation relative to the datum switches is advised (defined within machine parameter P17) otherwise the value of the last compensation will be applied to these points.

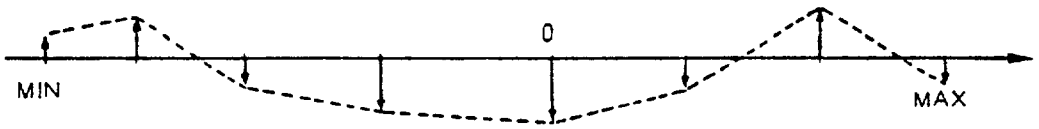
Point 0 (measurement 0) can be compensated.

Examples :

- Without compensation at the max and min positions



- With compensation at the max and min positions



Note :

For a rotary axis, compensation of points 0° and 360° must always mandatorily be the same.

With the utility in the memory, pressing "service" key gives access to the menu of the utilities which are present in the system.

Key in of 2 followed by (LF) displays the following menu :

CORRECTIONS DE MESURE	[MEASUREMENT COMPENSATIONS]
0 VISUALISATION - MODIFICATION	DISPLAY - MODIFICATION
1 DECHARGEMENT PAR RUBAN	LOADING BY TAPE
2 DECHARGEMENT SUR RUBAN	UNLOADING ONTO TAPE

To leave the menu, press X OFF. If compensations or modifications have been made "GRAVURE EN COURS" ..... ["WRITING IN PROGRESS"] is displayed, then a return to "current point" at the end of the write mode.

2.1. - DISPLAY - MODIFICATION

Key in 0 (LF). Question "AXE ?" ..... ["AXIS ?"] is displayed on the dialog line.

Specify the number of the axis to be compensated, this number being given by machine parameter P0 (0 to 7 for the lathe, 0 to 8 for the milling machine).

Key in the axis number followed by (LF) : "CORRECTIONS DE MESURE AXE" ..... ["AXIS MEASUREMENT COMPENSATIONS"] table is displayed.

This table is configured from the most negative dimension to the most positive dimension for the values of compensation.

M ± dimension of the point to be compensated ␣ C ± compensation value.

If no compensation is present, the table is blank, only the heading is displayed.

Compensation input is achieved in accordance with the following format :

M ± dimension ␣ C ± compensation (LF)

Sign + is taken by default

Dimensional input can be in any order, the system classifies them automatically from the most negative to the most positive.

To delete a compensation, point to the line which must be erased then key in -(LF).

To modify or add a line, it is not necessary to point to the line to be modified or to the line preceding the line to be inserted.

Modification of a line requires a complete rewrite of this line.

To compensate another axis, leave the present axis by pressing "service" key, which returns to the menu, and selection of the new one.

All axes having been compensated, press X OFF which initiates write mode and "current page" is automatically displayed.

2.2. - LOADING BY TAPE

Key in 1 (LF); display of "PRET ?" ..... ["READY ?"] on the dialog line.

With the tape on the tape-reader, answer "0 (LF)", which starts read mode.

If the business number corresponds to the numerical control in which the compensations are input, read mode is carried out in one go, with return to the menu at the end of read mode.

If the business number does not correspond to the numerical control, read mode stops after the business number and this number is displayed on the dialog line. After correction of the number, by superimposing over the number displayed, press (LF). Read mode is restarted and return to the menu occurs at the end of read mode.

Write into EEPROM is initiated when X OFF is keyed.

Should there be an error, the line is displayed on the dialog line. After compensation restart with (LF). (A character is deleted by - when the cursor is positioned on it).

Punched tape structure

Punching is done on a teletype.

```
% 0 0 0 8 4 2 7 2 ; 0 9                    (business number)
A X E   :   0 ; 0 7                        AXIS
M - 1 0 0 0 0   C - 1 0 ; 0 C
M - 3 5 0 0   C - 2 ; 0 A
M - 5 0 0   C 3 ; 0 8
M 1 2 0 0   C 6 ; 0 8
M 4 2 0 0   C 1 0 ; 0 9
M 1 0 0 0 0   C 5 ; 0 9
A X E   :   1 ; 0 7
M - 5 0 0 0 0   C 8 ; 0 A
M - 2 5 0 0 0   C 3 ; 0 A
M - 1 0 0 0 0   C - 2 ; 0 B
M - 2 0 0 0   C - 6 ; 0 A
M 0   C - 8 ; 0 6
M 1 0 0 0   C - 3 ; 0 9
M 2 0 0 0 0   C - 2 ; 0 A
M 5 0 0 0 0   C 1 ; 0 9
X O F F
```

The two digits after ; indicate the number of characters of each line, space included (in hexadecimal code). They must be calculated for each line.

The syntax for the axis number must be respected : 1 space between E and :, and a space between : and the first digit of the axis.

If comments are present after the business number, they are not taken into account by the NC.

If the tape is punched from an NC, the dimensions are justified and the total per line is always 17 (in hexadecimal code).

2.3. - OUTPUT TO A TAPE PUNCH

Key in 2 (LF) : display of "PRET ?" ..... ["READY ?"] on the dialog line.

The tape punch is now operational. Answer "0 (LF)" which starts tape punching.

Punching a tape starts with a 1 m leader and display of % business number of the numerical control. This number can be modified by superimposing a new number (% included), with the aid of the management cursor and the alphanumerical keyboard.

Punching is restarted by (LF). The tables of all axes which are compensated are punched. Return to the menu at the end of punched tape.

The structure of the punched tape has the format below. The two digits at the end of the line represent the number of characters of each line in hexadecimal code, space included; 23 characters always, i.e. 17 in hexadecimal code).

```
% 00084001 ; 0A
AXE :  0;07
M-    10000     C-     2;17
M-      789     C+     1;17
M+        0     C+     8;17
M+      500     C-     5;17
M+    12000     C+     4;17
AXE :  1;07
M-    50000     C+     8;17
M-    32000     C+     5;17
M-    16000     C+     3;17
M+       10     C-     1;17
M+    20000     C+     5;17
AXE :  2;07
M-   100000     C+    10;17
M-    50000     C+     2;17
M+        0     C+     5;17
M+    30000     C-     1;17
M+    80000     C-     5;17
;0
```

Comments can be added after the business number before restarting punching, for example "% 00084001 LE 20 JUIN 1984" ..... ["% 00084001 20th JUNE 1984".]

3. - UTILITY 3 : LOADING FIXED MACROS

This utility allows transference of special programs from RAM to EEPROM, or when modification of these programs is required, from EEPROM to RAM.

The available memory is 4Kbyte.

Before being transferred into EEPROM, these programs must previously be programmed in RAM (workpiece program).

Using this utility requires the presence of "resident macros" option.

When the utility is in the memory, pressing "service" key gives access to the menu of the utilities present in the system.

Key in of 3 followed by (LF) displays the following menu :

MACROS RESIDENTES	RESIDENT MACROS
0 DUPLICATION (%%) RAM - (DEBUT) EEPROM	(%%) RAM - EEPROM DUPLICATION
1 DUPLICATION EEPROM - RAM	EEPROM - RAM DUPLICATION
2 EFFACEMENT PROGRAMME EN EEPROM	ERASING PROGRAMS IN EEPROM
3 RAZ PROGRAMME EN EEPROM	CLEARANCE OF PROGRAMS IN EEPROM

To leave the menu, press X OFF which returns display to "current point" page.

When one of the requests has been effective, pressing X OFF key reinitializes the NC software.

3.1. - RAM - EEPROM DUPLICATION

Allows transfer of one or all RAM programs into EEPROM.

Key in 0 (LF) ; display of (%% or %---> &) % ■ on the dialog line.

3.1.1. - Program duplication

Key in the number of the program to be transferred followed by (LF) ?. The value will be written on the dialog line. Four cases can occur :

- The program does not exist in the RAM : display of "PROGRAMME INEXISTANT" ..... ["PROGRAM DOES NOT EXIST"] above the dialog line, and the cursor flashes on the program number.

- The program number is already in the EEPROM : "PROGRAMME EXISTANT" ..... ["PROGRAM EXISTS"] is displayed above the dialogue line, and the cursor flashes on the program number. In that case, if this request is to be maintained, the existing program must be erased (2 on the menu), then return to 0 (LF)..

- the EEPROM area is not sufficient to contain the program. In that case, "RAM ou EEPROM SATURABLE" ..... ["RAM OR EEPROM MAY BE SATURATED"] message is displayed above the dialog line, and the cursor flashes on the program number.

- If none of the above three cases appears, when (LF) is pressed, a flashing message "DUPLICATION EN COURS" ..... ["DUPLICATION IN PROGRESS"] appears above the dialog line.

  At the completion of write-in, there is a steady display of "FIN DUPLICATION" ..... ["END OF DUPLICATION"] and the cursor points to the program number. At that time, either the procedure can be relinquished by "service" key which returns to the menu, or by X OFF, or duplication of another program can be started.

3.1.2. - Duplication of several programs

If there are several programs in the RAM, these can be duplicated in one single handling. Instead of displaying the program number, write % (LF).

The previously mentioned fault cases and display messages can occur.
Note :

- If writing difficulties occur when starting duplication, "DIFFICULTE ECRITURE" ..... ["WRITE-IN DIFFICULTY"] message is displayed and remains displayed during duplication. At the end of write-in, two messages - "FIN DUPLICATION" ..... ["END OF DUPLICATION"] and "DIFFICULTE ECRITURE" ..... ["WRITING DIFFICULTY"] are displayed, but program transfer has occurred.

- If " DUPLICATION EN COURS" ..... ["DUPLICATION IN PROGRESS"] or "DIFFICULTE ECRITURE" ....["WRITE-IN DIFFICULTY"] are present and steady, there is a fault in the memories.

3.2. - EEPROM - RAM DUPLICATION

Allows transferance of a program from EEPROM memory to RAM memory if the "protected macros" option is not present.

Key in 1 (LF) ; display of "(& %..>--->%..) & % ■ " on the dialog line.

Key in the number of the program to be transferred followed by (LF) ; the value will be written in the location of the cursor. Four cases can occur :

- The program does not exist in the EEPROM : "PROGRAMME INEXISTANT" ..... ["PROGRAM DOES NOT EXIST"] is displayed above the dialog line, and the cursor flashes on the program number.

- The program number already exists in the RAM : "PROGRAMME EXISTANT" ..... ["PROGRAM EXISTS"] is displayed above the dialog line, and the cursor flashes on the program number.

- The RAM memory area is not sufficient to contain the program : "RAM ou EEPROM SATURABLE" ..... ["RAM or EEPROM MAY BE SATURATED"] is displayed above the dialog line, and the cursor flashes on the program number.

- Transference of the program occurs, when (LF) is keyed and a fixed message "DUPLICATION EN COURS" ..... ["DUPLICATION IN PROGRESS"] is displayed.

  At the end of the transfer there is a display of fixed message "FIN DUPLICATION" ..... ["END OF DUPLICATION"], and the cursor points to the program number. At that time, either the procedure can be relinquished by "service" key which returns to the menu, or by X OFF, or duplication of another program can be started.

3.3. - ERASING PROGRAMS IN EEPROM

This procedure allows one or several programs in EEPROM to be erased.

Key in 2 (LF) ; display on the dialog line of "(- & % ..) - & % ■ ".

Key in the number of the program to be erased, followed by (LF), the value is written in the location of the cursor.

If the program does not exist, "PROGRAMME INEXISTANT" ..... ["PROGRAM DOES NOT EXIST"] is displayed above the dialog line, and the cursor flashes on the program number.

If the program exists, flashing display of "EFFACEMENT EN COURS" ..... "ERASING IN PROGRESS" , and when erasing is finished, steady display of "PROGRAMME EFFACE" ..... ["PROGRAM ERASED"] with the cursor pointing to the program number. Another program can be erased or the procedure can be left by "service" key or X OFF.

If the program which is erased is not the last one in the EEPROM memory, the following programs are transposed :

Erasing of all EEPROM programs

This command allows the entire EEPROM memory to be erased. It can be done by the previous method, but the procedure must be repeated for each program whereas the following method is almost immediate.

Key in 3 (LF) ; display of "(- & %%) - & % ■ " on the dialog line.

Key in % followed by (LF), % is written in the cursor location.

Very rapid display of "EFFACEMENT EN COURS" ..... ["ERASING IN PROGRESS",] then display of "PROGRAMME EFFACE" ..... ["PROGRAM ERASED"].

Relinquish the procedure by "service" key which returns to the menu, or by X OFF.

This procedure can be used only if the "protected macros" option is present.

3.4. - DISPLAY OF THE PROGRAMS IN THE EEPROM

After exiting from utility 3 by X OFF, press key LIST. (&) (%% or %.. N..)■ is displayed on the dialog line.

Key in & (shift Y) %% (LF). The list of programs in the EEPROM is displayed with the number of bytes taken up by each program, the number of bytes used and the number of bytes available.

Display of the content of a program in EEPROM (if "protected macros" option is not enabled) is done by cursor management in front of the program desired and pressing LIST key. The beginning of the selected program is displayed, the remainder of this program is obtained by pressing "continued" key.

4. - UTILITY 5 : MACHINE PARAMETER INPUT

Loading, display or modification of the machine parameters requires utility 5 to be present in the memory. See 1.2 - loading a utility.

When the utility is in the memory, pressing "service" key displays the utility choice menu or abort. By keying in 5 (LF), the following menu is displayed.

PARAMETRES MACHINE	MACHINE PARAMETERS
> 0 VISUALISATION	DISPLAY
1 MODIFICATION	MODIFICATION
2 CHARGEMENT PAR RUBAN	LOADING BY TAPE
3 DECHARGEMENT SUR RUBAN	UNLOADING UNTO TAPE

To leave the menu, key in X OFF which returns the display to "current point" page. When the content of a parameter has been modified, keying in X OFF reinitializes the NC software.

4.1. - DISPLAY OF PARAMETERS

CALL "MACHINE PARAMETER" menu, as previously described. When displaying the menu, the pointer is automatically on 0. Pressing "continued" key displays "PARAMETRE ? " ..... ["PARAMETER ? "] question on the dialog line.

Choose the parameter to be displayed through the alphanumerical keyboard for example P10 (LF). On the screen is a display of :

```
(DISPLAY)  PARA : 10   TYPE : 0   WORD NB : 2
           > N0 00
             N1 00
```

To display all parameters, when "PARAMETRE ?" ..... ["PARAMETER ?"] question is displayed, press "continued" key which displays P0. Pressing "continued", again displays P1, and so on.

Pressing "service" key returns to the menu.

Page displayed for one parameter

When modification or display of one parameter has been requested, the following appears on the display page :

```
(MODIF)  PARA :     TYPE :     WORD NB :

OR

(DISPLAY)

     > N0
       N1
       N2
       N3
       .
       .
       .
```

- After PARA, the number of the parameter requested is displayed.

- After TYPE, are written either 0, 1, 2, or 3.

  . 0 corresponds to a word written in hexadecimal code from 00 to FF
  . 1 corresponds to a word written in decimal code from - 32767 to + 32767
  . 2 corresponds to a word written in decimal code from - 99999999 to + 99999999
  . 3 means the word corresponds to a character, X, for example.

- WORD NB indicates the number of words included in the parameter.

- N0 - N1 - N2 ..... gives the word number. 5 words maximum are displayed, the following words are obtained by pressing " ↓ " key.

4.2. - MODIFICATION OF PARAMETERS

Call "PARAMETRE MACHINE" ..... ["MACHINE PARAMETER"] menu, key in 1 (LF). This positions the cursor on "MODIFICATION" line and "PARAMETRE" ..... ["PARAMETER"] question on the dialog line.

Input the parameter to be modified on the keyboard, followed by (LF). For example, P10 (LF). Display of :

```
(MODIF)     PARA : 10     TYPE : 0     WORD NB : 2
            > N0 00
              N1 02
```

Bring the cursor in front of the line to be modified, then modify the word by means of # "New value" (LF). Up to ten values in succession can be modified from the line pointed to. To enter this key in # followed by N values ($N \leq 10$), separated from one another by a space. If N exceeds 10 values, the first 10 only will be taken into account and the following ones will be ignored.

If there is only one character to be modified on the line, pressing # followed by (LF) brings the line pointed to, onto the dialog line and the modification will occur by cursor management.

Pressing "service" key returns display to the menu.

4.3. - TAPE LOADING OF PARAMETERS

Call the menu, key in 2 (LF). This positions the pointer in front of 2, and question "PRET ? O/N["READY ?"] on the dialog line.

When the program tape is on the tape-reader, answer "0 (LF)". This starts tape read mode, and rapid scrolling on the display of what is being read. At the end of read mode, the menu is displayed.

If the tape identification does not correspond to the business number in the NC memory, read mode is stopped and identification is displayed on the dialog line. After the number has been corrected, key in (LF). Read mode is restarted, and the menu is displayed when read mode is complete.

4.4. - UNLOADING ON THE TAPE PUNCH

Call the menu, key in 3 (LF). This positions the pointer in front of 3, and question "PRET" ? O/N["READY ?"] is displayed on the dialog line.

When the tape-punch is connected, answer "0 LF". The punch is started with sprocket punching over about 1.50 m leader and display of the business number on the dialog line (eight digits are possible), for example : % 00082575.

The business number can be modified and a comment can be written; upon keying in (LF), the number, comment and all parameters are punched on the tape.

At the end of punch data about 1 m trailer feeds out before stopping with return to the display menu.

4.5. - LISTING FORMAT

```
%00000001  Comments;12
    PON0;04
    #X Z 0 U V 0 C 0 0 0 ; 14
    PON10 ; 05
    #0 0 0 0 0 ; 0A
    P1N0;04
    #0 0;04
    P2N0;04
    #0 3;04
    "
    "
    "
    "
    P17N0;05
    #-220000 +5000 -450000 +5000 -10000 +50000 +0 +0 +0 +0;36
    "
    "
    "
    P46N0;05
    #0 0 66 1226 200 3680 0 0 0;1D
    P46N10;06
    #0 0;04
    P47N0;05
    #0 0 50 600 100 1500 300 2500 0 0;21
    P47N10;06
    #0 0;04
    P48N0;05
    #0 0;04
    X OFF
```

- % 00000001 indicates the business number.

- At the end of each line, the figure representing the number of characters on the line, space or # included, in hexadecimal code, must mandatorily be keyed in. VIS. ; 12

- Sign # means that it is a data line.

- N followed by one or two digits indicates the number of the first word on the following data line.

  So for P46, the first line concerns the value of words 0 to 9, the second line concerns the value of words 10 and 11.

- A data line generally does not exceed 10 words. Parameters P0, P9 P11, P17, P30, P46, P47 and P60 are parameters which exceed this number.

- Parameter P50 must not be modified. The basic values are 10000 for the monoslide lathe or milling machine, 12000 for the lathe with three groups of axes and 15000 for the lathe with 5 groups of axes.

- It is not necessary to enter parameters, which are not required to be modified, from their present value.

5. - UTILITY 6 : PROGRAMMING THE PC IN ASSEMBLER LANGUAGE

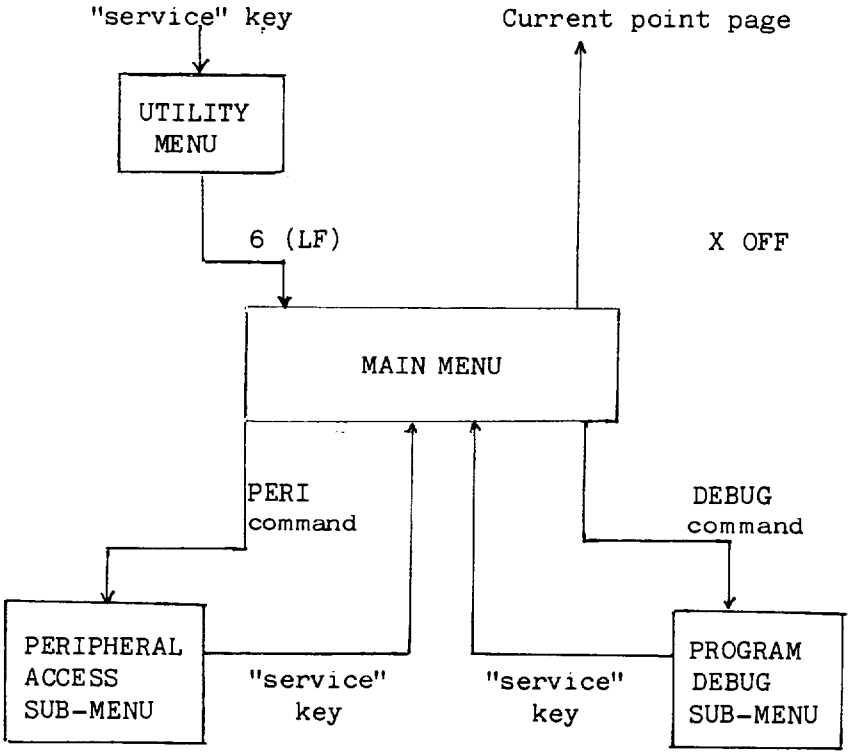
Loading, display, modification or debugging of the PC program in assembler require the presence of utility 6 in the memory. To load the utility see 1.2.

5.1. - COMPOSITION OF THE UTILITY

This utility includes three different menus : the main menu displayed when 6 (LF) is selected in the utility menu, and two sub-menus called from the main menu. The main menu allows the program to be written and the PC system to be configured. One of the sub-menus concerns the peripherals : loading or unloading the program in binary, unloading the program on a printer, printing the cross references ; the other sub-menu is intended for program debugging.

Exit from the main menu occurs by pressing X OFF key which returns display to "current point" page, with reinitialization of the NC software should a write-in have occurred.

Exit from a sub-menu, is done by pressing "service" key which returns to the main menu.



Main menu

Writing 6 (LF) in the utility menu displays the following menu :

PROG. AUTOMATE EN ASSEMBLEUR	[PC PROG. IN ASSEMBLER]
- EDITION MODULE ....< NOM DU MODULE >	MODULE OUTPUT ..... < MODULE NAME >
- IDENTIFICATION LOGICIEL ...	SOFTWARE IDENTIFICATION .... IDEN
- CONFIGURATION MEMOIRE	MEMORY CONFIGURATION ..... MEM
- CONFIGURATION SORTIES / OCTETS	BYTE/OUTPUT CONFIGURATION ... BYT
- DEFINITION TACHES HARD	HARD TASK DEFINITION ..... THD
- ACCES PERIPHERIQUE	PERIPHERAL ACCESS ..... PERI
- TABLE D'EQUIVALENCE	EQUIVALENCE TABLE ..... EQUI
- MISE AU POINT PROGRAMME	PROGRAM DEBUG ..... DEBU]

If the program appearing in the memory is not enabled, the system displays : "MEMOIRE AUTOMATE A INITIALISER (INIT)" ..... ["PC MEMORY TO BE INITIALIZED (INIT)"].

Write-in of INIT (LF) erases the PC memory with display of "GRAVURE EN COURS" ..... ["WRITING IN PROGRESS"] on the dialog line.

At the end of initialization, return to the menu above.

Note :

- If utilities 6 and 7 are used simultaneously, write in INI 2 (LF) instead of INIT, otherwise the complete program would be destroyed : program is cleared in 7 if display is in 6 and vice versa.

- Erasure of all tasks is possible at any time, by keying INIT (LF) on the main menu page.

5.2. - MEMORY CONFIGURATION : MEM

Before any program is loaded, an EEPROM memory area must be reserved for each task used, i.e. sequential, segment, background tasks, hardware tasks. Likewise, memory areas must be reserved for the constant, label and message tables. The whole of the reserved areas must not exceed the total capacity of the PC memory.

Three bytes are reserved per constant, for the constant table.

For the table of labels, each label reservation reserves ten bytes : two bytes for the address of the variables and eight bytes for the label in ASCII.

For the table of messages, each message reservation corresponds to an automatic reservation of 32 bytes.

Memory area reservation is done with command MEM (LF). The following table appears on the display (if no program had been introduced previously) :

SECTOR	UTIL	RESERV
TC	0	0
TS0	0	0
TS1	0	0
⋮		
TS4	0	0
TF0	0	0
⋮		
TF15	0	0
TH0	0	0
⋮		
TH7	0	0
SP0	0	0
⋮		
SP32	0	0
TE	0	0
SP33	0	0
⋮		
SP61	0	0
TM	0	0
TE2	0	0

RESERV column indicates the number of bytes the programmer wishes to reserve for different sectors. Column UTIL indicates the number of bytes used by each program. It is increased as the PC programs are input.

There are 95 sectors, 9 sectors being displayed per page, the remainder is obtained by pressing "continued" key.

To reserve a sector, key in on the alphanumerical keyboard with display on the dialog line :

Name of the sector ␣ Number of bytes reserved (LF)

For example : TS0 ␣ 2500 (LF)

When pressing LF, "GRAVURE EN COURS" ..... ["WRITING IN PROGRESS"] is displayed and, at the end of write-in the number of reserved bytes appears in the table in front of the sector concerned.

If the reservation is greater than the available memory area available, the command will be refused with display of "ZONE MEMOIRE SATUREE" ..... ["MEMORY AREA SATURATED"] message.

When sectors have already been programmed, reservation of an area for a new sector or an extension of an existing sector does not destroy the other sectors nor the sector itself.

When the various reservations have been carried out, pressing "service" key returns displayto the main menu.

Note :

Subprograms SP60 and 61 are reserved should there be some inter-PC exchanges (NUM BUS procedure).

5.3. - MODULE OUTPUT : < NOM DU MODULE >.....[< MODULE NAME >]

Before entering the program, the various modules used must have their memory area reserved (MEM). It is only possible to program module by module.

To write any module task, it is necessary to enable it in edit mode by <module name > (LF) ; TSO (LF) for example.

When editing, the operator has the following commands available :

SYNTAX	PROCEDURE
+ instruction (LF)	Instruction insertion between the line pointed to and the following line
- LF	Deletion of the line pointed to
≠ instruction (LF)	Modification of the line pointed to
≠ (LF)	Display on the dialog line of ≠ followed by the instruction pointed to for any modification
; instruction (LF)	Search for first occurrence of the instruction following ;
; operand (LF)	Search for first occurrence of the operand following ;
; (LF)	Search for the following occurrence of the instruction or operand
U n (LF)	Cursor backtrack by n lines $0 \leqslant n < 9999$
D n (LF)	Cursor advance by n lines
↓	Advance of the cursor by one line
Key to the left of key →	Backtrack of the cursor by one line
.../...	Cursor advance by one page
Sn (LF) or ; Sn (LF)	Search for sequence Sn
EQUI	Modification of one or severable variables according to the table of equivalence
MERG	Insertion of a module in another module
PERF (LF)	Symbolic perforation of the sector
LIST (LF)	Symbolic listing of the sector
LECT (LF)	Sector read-in by tape
INIT (LF)	Erasing the edited program
QUIT (LF)	Exit from edit without edited program safeguard

Note :

A module which is written into utility 7 cannot be called in utility 6 and at the bottom of display page "NON EDITABLE SOUS CET UTILITAIRE" ..... ["CANNOT BE EDITED IN THIS UTILITY"] message is displayed.

5.3.1. - Program input on the keyboard

Select main menu of utility 6.
Key in (with display on the dialog line) :

TSO (LF)

At the head of the page will appear :

TSO UTIL : 0 RESERV : 2500 (for example)

The cursor will be pointing to centre line of display. To write the program, each line must start with sign +. For example :

+ S10␣MAR BRO (LF)

This line is written on the dialog line. Keying LF causes the line to be displayed opposite the cursor, and the dialog line is erased. Current page is justified automatically.

The following program line will be written , for example :

+ LD␣E.2 (LF)

When keying in (LF), this new line is written in opposite the cursor, and the preceding line rises by one line. There can only be one instruction or sequence per line.

Should there be an error in the instruction (does not exist or is incorrectly entered), non respect of space or absence of separating point in the operator, keying in (LF) brings the cursor ■ to the beginning of the line and there is no transfer of the dialog line to the current line.

With each line transfer, UTIL is incremented by the number of bytes used for the line instruction.

When the whole program has been written, (there is no special character to indicate the end of a program), to safeguard it and leave this output mode, press "service" key. "ANALYSE PROGRAMME EN COURS" ..... ["PROGRAM ANALYSIS IN PROGRESS"] is then displayed on the dialog line.

This analysis checks whether there is a jump to sequences which do not exist (error 3), or the same sequence number occurs several times (Error 2).

Should there be an error, the analysis is stopped and the faulty sequence is displayed on the current line with the error number. After correction (see program modification) pressing "service" key restarts the analysis of the program from the beginning.

At the end of the analysis the system automatically starts memory write-in. "GRAVURE EN COURS" ..... ["WRITING IN PROGRESS"] is displayed on the dialog line.

Write-in execution time is 2 min 40 sec for an 8K program. The system returns to the menu at the end of write-in.

When the program is being written, if the part used becomes greater than the area reserved, message "ZONE MEMOIRE SATUREE" ..... ["MEMORY AREA SATURATED"] is displayed on the dialog line.

Press "service" key which will allow analysis and write-in of the program which has already been written. After having returned to "AUTOMATE PROGRAMMABLE" ..... ["PROGRAMMABLE PC"] menu, configure the sector concerned again, reselect sector editing and search for the sequence which had been partially written (by Sxxx (LF)), then carry on writing the program.

5.3.2. - Program input in symbolic language by the tape-reader

The program is in ISO code and its format is required to be :

```
% TSO   Comments      (CR LF)
S10     DOOR SECU     (CR LF)
LD      E.3           (CR LF)
AND     B.80          (CR LF)
ST      A.12          (CR LF)
S20     DATUM SWITCH  (CR LF)
 .
 .
 .
 .
ST      A.25          (CR LF)
X OFF
```

With the program tape on the tape-reader and a memory area reserved for the module to be input, call the module name from the menu , for example : TSO (LF).

Display of :

```
TSO     UTIL : 0     RESERV : 4000 (for example)
```

Key in LECT which is displayed on the dialog line. Then by pressing (LF), the tape-reader reads the tape and the program scrolls rapidly on the dialog line. At the end of read-in with character X OFF, the page displayed is :

```
TSO     UTIL : 3 652 (for example)   RESERV : 4000
```

together with the first sequences of the program.

Note :

If the programmer wishes to copy one task into another, when the tape is read, the name of the module read is displayed on the dialog line and can be replaced with the aid of the NC keyboard by the name of the module selected from the menu. Restarting the tape-reader is engaged by (LF).

Pressing "service" key initiates the program analysis with display of "ANALYSE PROGRAMME EN COURS" ..... ["PROGRAM ANALYSIS IN PROGRESS"] on the dialog line.

If no error is detected, the system changes automatically to memory write-in and displays "GRAVURE EN COURS" ..... ["WRITE-IN IN PROGRESS"] on the dialog line.

Automatic return to the menu page at the end of write-in.

Should there be an error, the analysis is stopped and the sequence at fault with the error number is displayed on the current line.

The various types of errors encountered when inputting the program are :

1 - Syntax error or instruction not recognized
2 - Double definition of the sequence number
3 - Jump to a sequence which is not defined
4 - Operand is inaccessible
5 - Wrong sub-progran address

Should there be a syntax error or an unrecognized instruction, the line is replaced by "NOP". For example, if END␣E.3 has been written on the tape NOP will be displayed, hence the necessity of inputting the complete line through the keyboard.

As an error is corrected, reinitiate the analysis of the program with "service" key.

The program is automatically justified by the system.

5.3.3. - Output of the program in symbolic language on the tape-punch or printer

When the module to be output has been chosen, two commands are possible :
- PERF (LF) for output on the tape-punch (ISO code)
- LIST (LF) for output on the printer

Choice of the sector to output, TSO for example. In menu page, key in TSO (LF).

Display of :

TSO UTIL : 3652 (for example) RESERV : 4000 (for example)

as well as the beginning of the program.

Write the command chosen on the dialog line.

Display of % TSO on the dialog line of the display page.

A message can be written at that time, for example, date, name of the programmer, machine to which this program relates, etc... This message can have any length and is not stored in the PC memory. At the end of the message pressing (LF) punches or prints the message and the program.

Printing the program occurs in the same format as the page displayed, i.e. tabulations included.

If another module must be printed, carry out the same manipulations as previously.

All modules can be printed.

Notes :
In tape punch mode, the system generates a 1 m sprocket leader before data outpout and also a 1 m trailer at the end of data. Listing the program can also be carried out from the PERI module.

5.3.4. - Program modification

As long as utility 6 is present a program in the memory can be modified.

From the menu, choose the module that is to be modified.

- Search for a required sequence by entering Sxxx (LF). The sequence called will be displayed opposite the cursor.

- Searching for an instruction is done by entering ; LD␣E.3 (LF) (for example search for input 3 loading instruction).

  The first line encountered which includes this instruction starting from the beginning of the sector will be displayed on the current line.

  To obtain other lines which include the same instruction key in ; (LF). The following line will be displayed, and so on.

- Searching for an operand is done in the same manner as above :

  ; E.3 (LF) to search for the first occurrence and ; (LF) to point to the following ones.

- Cursor backtrack by n lines : Un (LF)

- Cursor advance by n lines : Dn (LF)

- Character modification in one line.
  Key in * (LF) which transfers the line pointed to onto the dialog line. Modify the term or terms. Pressing (LF) modifies the line pointed to and erases the dialog line.

- Modification of a complete line
  Key in # and the new line on the dialog line.
  Pressing (LF) replaces the line pointed to by the new line and erases the dialog line.

- Deletion of a line
  Bring the line to be deleted in line with the cursor > then key in - (LF) ; the following line will become the new current line.

- Adding one line
  + instructions↵Operator (LF), this line will be inserted after the current line and will itself become the current line.

- Cursor command keys, character insertion, character cancelling, page continued or line continued are used to modify, or input a program through the keyboard.

- Key-in of the word QUIT allows leaving editing without modifications being taken into account.

5.3.5. - Module-insertion : MERG

With this command the operator can insert any module in another module.

Place the cursor at the insertion point. Key in command MERG followed by a blank and the name of the module to be inserted. If the size reserved is not sufficient to allow this insertion, "ZONE RESERVEE INSUFFISANTE" ..... ["AREA RESERVED IS NOT SUFFICIENT"] message is displayed.

Once the insertion is done, the cursor is positioned on the last instruction of the module inserted, to allow a new insertion should the occasion arise.

This command cannot be used for TC, TM, TE and TE2 (for TE and TE2 see table of labels paragraph - 5.3.8.).

5.3.6. - Program display

- Display in static mode

  When the program is in the memory, if one wants a module to be displayed, utility 6 must be loaded, then, through the processes described above, call the main menu.

  Choose the module to be displayed, then, through page and cursor management, do the check wished for. If no modifications are to be made to this program or if the modifications made must not be taken into account, writing QUIT allows return to the main menu. "Current point" page can be be returned to by X OFF.

- Display in dynamic mode

  The status of the internal variables (B), memories (M), constants (V), inputs (E), outputs (A), timing (T), counters (C), bytes (EN, AN) can be displayed. For some operators such as constants, timings, etc... the values are displayed in decimal and hexadecimal code.

Pressing Input/Output key displays menu :

0	MESSAGE DE DIAGNOSTIC	[DIAGNOSTIC MESSAGE
1	PARAMETRES PERIPHERIQUES	PERIPHERAL PARAMETERS
2	MEMOIRE AUTOMATE	PC MEMORY
3	MODE TRANSPARENT	TRANSPARENT MODE]

2 (LF) displays "ADRESSES ?" ..... ["ADDRESSES ?"] on the dialog line.

Write-in the operators wished for while observing the syntax (10 maximum). For example B.14 ␣ M.12 ␣ E.1A ␣ T.3 (LF). These operators will be displayed with their values - fixed or evolving.

Pressing "continued" key allows obtaining other operators, the preceding ones being erased. If one wishes to keep them, press "continued" then key + and the new operators.

5.3.7. - Table of constants : TC

The table of constants is output by command TC (LF) ; to reserve 5 constants, for example, the table is displayed as follows :

TABLE DES CONSTANTES NOMBRE : 5
[TABLE OF CONSTANTS NUMBER : 5]

One display line is thus made up of :

1 - N° of the constant
2 - Value of the constant in absolute mode ($0 \leq V \leq 65535$)
3 - Value of the constant in relative mode ($-32768 \leq V \leq +65535$)
4 - Value in hexadecimal code ($\$0000 \leq V \leq \$FFFF$)
5 - ASCII representation of the constant if it exists, if a byte has no ASCII correspondance, no display

Input of a byte can thus be done in four different ways :

- Absolute input	N° of the constant ␣ # value
- Incremental input	N° of the constant ␣ #± value
- Hexadecimal input	N° of the constant ␣ $ value
- ASCII input	N° of the constant ␣ 2 characters

4 byte decimal value can be input (= 2 successive constants) to use constants as a table of values, such as travels, references, speeds, etc...

To do this write on the dialog line :

N° of the constant (LF)

Display of :

N° of the constant ≠ ± value $ value ≠ ■

After the last the operator writes the signed value he wishes to input followed by (LF). This value, limited to 8 decades, is broken down into 2 x 2 bytes and carried back on the 2 constants concerned.

Example : Value - 126200 in constant N° 1

- ≠ 1 (LF) keyed in by the operator
- 1 ≠ + 72810495 $ 0456 FFFF ≠ ■ is displayed on the dialog line
- the operator keys in -126200 on the keyboard which is written in the place of ■ and after pressing (LF), the table of constants becomes :

TABLE DES CONSTANTES NOMBRE : 5
[TABLE OF CONSTANTS NUMBER : 5]

0	≠ 12594	≠ + 12594	$ 3132	12
1	≠ 65534	≠ - 2	$ FFFE	
2	≠ 4872	≠ + 4872	$ 1308	
3	≠ 14132	≠ + 14132	$ 3734	74
4	≠ 65458	≠ - 78	$ FFB2	

Command PERF (LF) enables output of the table of constants on the tape.

Command LIST (LF) starts output on the printer.

Command LECT (LF) enables input of the constants by tape. In that case the syntax which must be observed is :

```
% TC                    Comment (CR LF)
0 ␣ ≠ - 1                      (CR LF)
1 ␣ $ FC                       (CR LF) All syntaxes accepted in keypad
2 ␣ 12                         (CR LF) input are accepted in tape read
3 ␣ ≠ 4283                     (CR LF) mode.
X OFF                          (CR LF)
```

Command INIT (LF) resets all constants to zero.

Command QUIT (LF) exits from editing without safeguard.

Note :

Printer output can also be done from the menu of PERI command.

5.3.8. - Table of labels : TE and TE2

Two tables are available to the programmer : one in EEPROM (TE) to retain the labels in the memory, the other in RAM (TE2) to output listings and cross references.

A table written in a module can be transferred to the other module by command MERG (the module zone must have been reserved before carrying out this command).

For example, labels written for inputs (E), actuators (A), variables (B), etc... in module TE2 : transfer occurs by selecting TE in the main menu, then write-in of MERG. The table will be transferred in TE but not destroyed in TE2.

All operands available for program write mode can be input to the table of labels (except immediate values I and II).

The label comprises 8 characters maximum and the operands are classified in increasing order by type.

Write-in of a label is done by :

A.10 ⏎ MAR BRO (LF)

When outputting a module listing on a printer, if the labels have been input in TE2, each time an operand has a corresponding mnemonic, the latter will be printed on the line.

Commands PERF, LIST, LECT, QUIT and INIT are possible. Command INIT (LF) clears the table in which it has been written.

5.3.9. - Table of messages : TM

Input of the message is done by :

N° of the message ⏎ Message LF (32 characters maximum)

Commands PERF, LIST, LECT, QUIT and INIT are possible. Command INIT initializes the table of messages by blanks.

Should there be an error in the text of a message, it can be recalled on the dialog line by command # followed by the number of the message.

5.4. - SOFTWARE IDENTIFICATION : IDEN

When the PC program has been input, it can be referenced and identified by command IDEN (LF).

The following page is displayed :

```
          N O   L O G I C I E L / I D E N T I F I C A T E U R

  N O   L O G  ( % )

  I D E N :

  Z O N E   M E M O I R E   O C C U P E E :  5 8 1

 ┌          S O F T W A R E   N O / I D E N T I F I E R          ┐
 │ S O F T   N O   ( % ) :                                       │
 │ I D E N   :                                                   │
 │ M E M O R Y   A R E A   O C C U P I E D   :   5 8 1           │
 └                                                               ┘
```

- The software number is generated by writing on the dialog line :

  % 8 characters maximum (LF)

- Identification by :

  IDEN ␣ 32 characters maximum (LF)

- The memory zone occupied indicates to the user the total memory which is occupied by the PC program, in the knowledge that this size cannot exceed :

  8191 basic bytes
  or 40959 with PC 40K option

  If no other program is in the memory, the occupied area is 581 bytes (value pertaining to the PC).

- Command LIST (LF) allows printer output of the identifier, memory configuration and finally byte output configuration.

- If the operator wishes to output onto tape, he can command PERF (LF).

NOTE :

Printer output can also be commanded from the peripheral menu PERI.

5.5. - BYTE OUTPUT CONFIGURATION : OCT

PC outputs to the machine, via the 32 output PCB, can be configured either in bits or in bytes; to do this key in command C OCT (LF).

The following page is displayed :

C O N F I G U R A T I O N S O R T I E S / O C T E T S

[B Y T E / O U T P U T C O N F I G U R A T I O N]

A N .	0	1	2	3	4	5	6	7
	0	0	0	1	1	0	0	0
A N .	8	9	A	B	C	D	E	F
	0	0	0	1	0	0	0	0
A N .	10	11	12	13	14	15	16	17
	0	0	0	0	0	0	0	0
A N .	18	19	1A	1B	1C	1D	1E	1F
	0	0	0	0	0	0	0	0

- AN.0 = 0 means that outputs 0 to 7 of the 32 output PCB addressed by 0-1 are in bits.

- AN.3 = 1 means that byte outputs 18 to 1F of the PCB addressed by 0-1 form a byte output.

Modification is made by AN. N° ⌴ 0 or 1 (LF)

Command INIT (LF) configures all bit outputs.

Note :

If 8 outputs are declared as a byte, it is not possible to use individual bits and vice versa.

5.6. - HARDWARE TASK DEFINITION : THD

Hardware tasks are used to process priority interrupts, hardware task 0 has the highest priority and hardware task 7 the least priority.

An interrupt input is associated with each hardware task. Its status is represented by one of the following words :

EN.3E	ACIA1 status register
EN.40	ACIA2 status register
EN.42 to EN.45	IT status word

Assigning a status word to a hardware task must be carried out in HARDWARE TASK DEFINITION page called by command THD (LF).

Display of page :

D E F I N I T I O N D E S T A C H E S H A R D

[H A R D W A R E T A S K D E F I N I T I O N]

T H 0 E N . 4 3

T H 1 E N . 3 E

T H 7

To assign EN.** to the hardware task the following must be keyed in

THn ⌴ EN.** (LF)

keying in followed in display by

THn EN.**

To reinitialize all the hardware tasks and thus enable a new configuration, key in INIT (LF).

5.7. - ACCESS TO THE PERIPHERALS : PERI

When the PC program has been entered in the memory, it is useful to file it on tape. This allows the program to be reloaded should there be incidents, or another identical machine to be programmed.

Likewise, it is worthwhile outputting the program on the printer to constitute the dossier of the machine.

To do this, the user can request the peripheral menu, from the main menu, which displays the following sub-menu :

```
A C C E S  A U X  P E R I P H E R I Q U E S

- D E C H A R G E M E N T  B I N A I R E  S U R  R U B A N  . .  P E R F
- V E R I F.  P E R F O R A T I O N  B I N A I R E . . . . . .  V E R I
- C H A R G E M E N T  B I N A I R E  P A R  R U B A N . . . . .  L E C T
- I M P R I M E R  T O U S  L E S  S E C T E U R S . . . . . .  L I S T
- I M P R I M E R  C R O S S - R E F E R E N C E . . . . . .  C R E F
```

```
[          A C C E S S  T O  T H E  P E R I P H E R A L S

 - B I N A R Y  P U N C H  O U T P U T  T O  T A P E . .  P E R F
 - B I N A R Y  P U N C H  V E R I F I C A T I O N . . . .  V E R I
 - B I N A R Y  L O A D I N G  B Y  T A P E . . . . . .  L E C T
 - P R I N T  A L L  S E C T O R S . . . . . .  L I S T
 - P R I N T  C R O S S - R E F E R E N C E . . . . . .  C R E F]
```

To exit, press "service" key which returns display to the main menu.

5.7.1. - Binary output to tape : PERF

The system transfers the whole content of the EEPRON onto tape in binary code. During the whole transfer, message "PERFO EN COURS" ..... ["PUNCHING IN PROGRESS"] is displayed.

1 m sprocket leader and trailer is automatically punched before and after the program on the tape. The program can be interrupted at any time by pressing "service" key, but cannot be resumed at the location of the stop.

5.7.2. - Binary punch verification : VERI

When the program has just been output to tape in binary code, its content must be checked with respect to the EEPROM content. When the tape is placed on the tape-reader, key in VERI (LF).

Tape read mode occurs with display of "VERIFICATION EN COURS" ..... ["VERIFICATION IN PROGRESS"]. If a punching error is detected, the tape-reader stops, and "MAUVAISE PERFORATION DU RUBAN" ..... ["INCORRECT TAPE PUNCHING"] is displayed.

The program must be punched again , or there is a fault on the tape-reader.

If no error is detected, "RUBAN BINAIRE CONFORME" ..... ["BINARY TAPE CONFORMS"] is displayed at the end of read mode. The contents of the tape and of the EEPROM memory are identical.

5.7.3. - Binary loading by tape : LECT

Command LECT (LF) allows loading the entire PC memory directly in binary. In that case, it is not necessary to reserve memory areas for the various tasks or tables (automatically done by tape).

LECT (LF) command reads the tape and displays "GRAVURE EN COURS" ..... ["WRITING IN PROGRESS"] on the dialog line. At the end of read mode, the dialog line is erased and the menu returned to.

Should there be a transfer fault or incorrect tape reading, the system sends an error message "ERREUR DE CHECKSUM" ..... ["CHECKSUM ERROR"] and the EEPROM (INIT) must be reinitialized.

5.7.4. - Sector output on the printer : LIST

This command allows outputting all PC program sectors to a printer (TSx, THx, SPx, ...) or outputting selected sectors.

Write-in of LIST (LF) displays the following message on the dialog line :

TOUT OU PARTIE (ALL / SELECT) ..... [ALL OR PART (ALL / SELECT) ?]

- Output of all sectors present by answering A (LF). A display of "LISTING EN COURS" ..... ["LISTING IN PROGRESS"] on the dialog line up to the end of print-out.

- To output one or several sectors, answer S (LF). Automatically, the system will present the sectors used one after the other.

  The first question after answer S is always :

  LISTING IDENTIFICATEUR ? ..... [IDENTIFIER LISTING ?]

  If listing is requested, answer with (LF). "LISTING EN COURS" ..... ["LISTING IN PROGRESS"] display up to the end of printing, then return to the following listing request with :

  LISTING DU SECTEUR ? ..... [SECTOR LISTING () ?]

  () name of the first sector which is not present (TSO for example). If the sector is not to be printed, pressing key ← causes a skip to the next sector. Answer with (LF) prints the sector.

  Printing or sector search can be interrupted at any time by pressing "service" key.

Note :

If utility 6 and utility 7 are used simultaneously, it will only be possible to print sectors written in utility 6 (except TE, TE2, TC, TM).

5.7.5. - Output of cross references : CREF

For the programmer or the user, it is interesting to know in which module and which sequence the different variables, memories, inputs, actuators ... are generated and used as well as the mnemonic associated with them. To do this, CREF (LF) command allows printer output of these variables.

The variables with a mnemonic are classified alphabetically and then the variables without mnemonic are classified by type and increasing number. For the cross references to be output with their menmonics, these must have been programmed in table of labels TE2.

Note :

- To obtain the list of variables, with or without mnemonics, classified by type of variable and increasing number, module TE2 must be printed using LIST (LF).

- In the case of programs written in utility 6 and 7, the cross references will only concern modules written in utility 6.

- Most printers use the 8th bit (parity bit) of the characters received for selection of another set of characters. Sending this parity bit limits the printer possibilities; to avoid this punching and listing outputs will henceforth be transmitted without the parity bit.

Example :

MNEMONIQUE MNEMONIC	ETIQUETTE LABEL	GENERATION GENERATION	UTILISATION USE
AC AAX+	E.22		TS0 S9006
AC AAX-	E.23		TS0 S9006
AC AAZ+	E.24		TS0 S9006
AC AAZ-	E.25		TS0 S9006
AOB VAR	A.1	TS0 S20	TS0 S20 S60 S9000 S9001 TS0 S9002 S9005 TS1 S1
BCD VAR	E.28		TS0 S3800
BC ROT	E.29		TS0 S100 S3805 S3835 TS0 S3850 TS1 S1100 S1130
BLD VAR	A.F	TS4 S105 S110	
BO MOT	A.2	TS0 S55 S100 S3825 S3830 S3840 S3845	TS0 S15 S3805 S3835
CY ADEP	A.14	TS0 S55 S100 S3520 S3574 S3580	TS0 S3566 S3568 S3580
EC MOT	E.11		TS1 S3610
EO AV	A.A	TS0 S55 S100 TS1 S3605 S3625 S3630	

MENMONIQUE MNEMONIC	ETIQUETTE LABEL	GENERATION GENERATION	UTILISATION USE
VAR INT	B.12	TS0 S250 S3725 TS4 S1250	TS0 S250 TS1 S1225 S4310
	E.13		TS0 S3028 S3036 S3056
	E.1E		TS3 S2015 S2130
	E.26		TS3 S2040 S2102
	A.9	TS0 S55 S100	
	A.D	TS0 S55 S100	
	A.E	TS0 S55 S100	
	A.15	TS0 S55 S100 S8011 S8013	
	A.16	TS0 S152 S3236 S3250	TS0 S3212
	B.10	TS0 S3805	TS0 S1105 S3003 S3104 TS0 S3212 S3502
	B.11	TS0 S1000	TS0 S3024 S3044 S3162
	B.27	TS0 S155 S3113 S3114 S3115 S3124	TS0 S3168 S3174 S3176
	B.28	TS0 S155	
	M.12	TS0 S3210	TS0 S3210
	M.14	TS0 S8001 S8005	TS0 S8002 TS1 S1005 S1010
	M.15	TS1 S1010	TS0 S8002
	EN.20		TS0 S3560 S9002
	EN.23		TS0 S3005 S3112 S3510 TS0 S4010 S8010
	AN.22	TS4 S105 S120	
	AN.23	TS4 S105 S120	
	AN.24	TS0 S9050	
	C.2	TS0 S8002 TS1 S1131 S1132	TS1 S1132
	HC.2	TS1 S3902	TS1 S3902
	QC.2	TS0 S55 TS1 S3900 S3903	TS1 S3902 S3904
	EN.30		TS0 S9050
	EN.31		TS0 S9040
	V.1		TS0 S8000 TS1 S1000 S1016
	V.3		TS0 S3200 S3210 S3230
	V.4		TS0 S3000 S3004 S3028

5.8. - EQUIVALENCE TABLE : EQUI

When one or several modules have been written with some functions having different variables it is possible to modify these to make a program compatible with different machine types (for example E.100 transformed to E.23).

All variables can be modified : E, A, M, T, C ... The maximum number of variables which can be modified is 512.

Section 5.9 - Program debug

To be inserted later.

- Main menu

Writing 7 (LF) in the utility menu displays the following menu :

```
          P R O G.  A U T O M A T E  P A R  S C H E M A S

- E D I T I O N  M O D U L E ..... N O M  D U  M O D U L E
- I D E N T I F I C A T I O N  L O G I C I E L ..... I D E N
- C O N F I G U R A T I O N  M E M O I R E ..... M E M
- C O N F I G U R A T I O N  S O R T I E S / O C T E T S .. O C T
- D E F I N I T I O N  T A C H E S  H A R D ..... T H D
- A C C E S  P E R I P H E R I Q U E S ..... P E R I
- T A B L E  D ' E Q U I V A L E N C E ..... E Q U I
- M I S E  A U  P O I N T  P R O G R A M M E ..... D E B U

          P C  P R O G R A M  B Y  R E L A Y  C H A R T

- M O D U L E  E D I T I N G ..... M O D U L E  N A M E
- S O F T W A R E  I D E N T I F I C A T I O N .....I D E N
- M E M O R Y  C O N F I G U R A T I O N ..... M E M
- B Y T E / O U T P U T  C O N F I G U R A T I O N ..... O C T
- H A R D W A R E  T A S K S  C O N F I G U R A T I O N .. T H D
- P E R I P H E R A L  A C C E S S .....P E R I
- T A B L E  O F  E Q U I V A L E N C E .....E Q U I
- P R O G R A M  D E B U G G I N G .....D E B U
```

If the program in the memory is not initialized, the system displays : "MEMOIRE AUTOMATE A INITIALISER (INIT)" ..... ["PC MEMORY TO BE INITIALISED (INIT)].

Writing INIT (LF) erases the PC memory with display on the dialog line of "GRAVURE EN COURS" ..... ["WRITING IN PROGRESS"].

At the end of initialization, return to the menu above.

Note :

- In the case of simultaneous use of utilities 7 and 6, write INI 2 (LF) instead of INIT, otherwise the complete program would be destroyed : program erased in 6 if display is in 7 and vice versa.

- Erasure of all tasks is possible at any time, by keying in INIT (LF) on the main menu page.

6.2. - MEMORY CONFIGURATION : MEM

Before any program is loaded, an EEPROM memory area must be reserved for each task used, i.e. sequential, segment, background task, hardware task. Likewise, memory areas must be reserved for the tables of constants, labels and messages. The total of these reserved areas must not exceed the total capacity of the PC memory.

For the table of constants, 2 bytes are reserved per constant.

for the table of labels , each label reservation reserves 10 bytes : 2 bytes for the address of the variable and 8 bytes for the ASCII label.

For the table of messages, each message reservation corresponds to an automatic reservation of 32 bytes.

The memory area reservation is done by command MEM (LF).

The following table is displayed (if no program had been introduced previously) :

SECTOR	UTIL	RESERV
TC	0	0
TS0	0	0
TS1	0	0
'		
'		
'		
TS4	0	0
TF0	0	0
'		
'		
'		
TF15	0	0
TH0	0	0
'		
'		
TH7	0	0
SP0	0	0
'		
'		
SP32	0	0
TE	0	0
SP33	0	0
'		
'		
SP61	0	0
TM	0	0
TE2	0	0

RESERV column indicates the number of bytes the programmer wishes to reserve for the various sectors, UTIL column indicates the number of bytes used by each program; this column is increased as PC programs are input.

There are 95 sectors, seven sectors being displayed per page. To scroll all sectors, use commands :

↑ : gives the seven preceding sectors (left key on the second display control keypad line)

↓ : gives the next seven sectors.

To reserve a sector, it is sufficient to key in the name of the sector followed by (LF). If the name of the sector is correct, the system asks "RESERVATION:". Key in the reserved number of sectors in decimal code, followed by (LF). "RESERVATION EN COURS" ...["RESERVATION IN PROGRESS"]is displayed at the bottom of the page.

Three cases can occur :

1 - The size reserved is greater than the maximum size of a sector (8 K), in that case the cursor re-positions itself at the beginning of the reservation and waits for another more consistent size.

2 - The size reserved is less than the length used, the system re-positions the cursor in the same position as for case 1.

3 - The size reserved exceeds the memory area available (EEPROM). The system displays an error message indicating the maximum size available in EEPROM.

Note :

- When an error message is displayed it is sufficient to key in any character to make it disappear and to return to the command mode.

- Sub-programs SP60 and SP61 are reserved for inter-PC exchanges (NUM BUS procedure).

6.3. - MODULE EDITING : <MODULE NAME>

Once the sector reservation is made, a sector can be edited. If constants (V or VV) are used, the table of constants must be programmed before writing the programs.

To return to editing of a sector, key in the name of the desired sector in the PC programming menu. Eg. : TSO (LF).

On calling the sector, two cases occur :

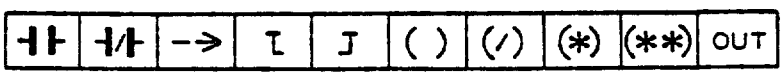
1 - The sector already contains sequences. In that case the system displays the first sequence of the sector.

2 - The sector contains no sequence. The cursor waits for a sequence number.

On creation of a sector, the system requests a sequence number.

The following page is displayed (at the bottom of the screen) :

SEQ : ■



The dynamic keys defined previously correspond to the first row of keys situated below the screen. Their significance is given in the table below

KEY	GRAPHIC	FUNCTION	ENABLED VARIABLES
⊣⊢	E12 ⊣⊢	Contact (normally open)	E0 to E17F B10 to B4FF A0 to A17F T0 to TF C0 to CF
⊣/⊢	E12 ⊣/⊢	Contact (normally closed)	ditto
→		Passage to the next relay Transfer-action	
⌈		Branch open	
⌋		Branch closure	
( )	A18 - ( ) -	Output	A0 to A17F B10 to B4FF
(/)	A26 -(/)-	Inverted output	ditto
(*)	B15 - (U) -	Unlatch	ditto
	B15 - (L) - ↓	Latch (held in memory)	
(**)	- (t) - 1234	Timer 1234 : threshold value the x are displayed by the system	T0 to TF 1234 timer value in 1/10th second
	/ E3 (C) 1234	Counter	C0 to CF /E3 : variable to count on the negative edge ⌙ E3 : variable to count on the positive edge 1234 : number of edges to count to set the counter flag to a logic 1
(OUT)		Enables exit from editing mode	

Note :

The threshold values of the timers and counters is expressed in four possible ways :

1 - immediate decimal : 256
2 - immediate hex : $FF
3 - EEPROM value : VVxx
4 - 2 byte memory : MMxx

The immediate values must be less than 32768 ($8000).

6.3.1. - Program input from the keyboard

Key in the number of the sequence desired, followed by (LF). If the sequence does not exist, the system asks for the type of sequence :

"TYPE" For a ladder sequence key in R (LF)
For a calculation sequence key in C (LF)

If (LF) is pressed directly, the system allocates the type of the last sequence edited.

If other characters are keyed in, the systems asks for the type again.

After type acquisition, the system asks for a message explaining the sequence. If there is no message, press (LF) directly.

The system goes directly to sequence editing mode with the cursor at the top left of the screen.

6.3.1.1. - Relay chart sequence

The system automatically displays the relay chart frame and places itself on the relay position on the 1st ladder line.

The dynamic keys defined at the bottom of the screen can be used to build the relay chart.

Keying in a contact or an output is purely sequential.

If -| |- is keyed in the cursor places itself above the contact and waits for the variable. For the first character, the system waits for the identifier (E, A, B, T, C) otherwise acquisition is not enabled. Then the number of the variable must be keyed in (3 digits maximum); if the number is less than 3, complete it with spaces. The system exits from writing a contact to place itself in the waiting position for a branch or line continued to be keyed in.

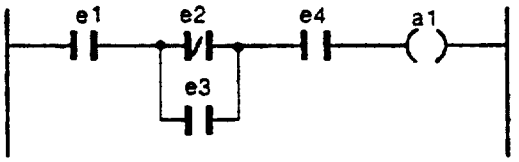
Note :
If the identification or the number are not correct, the system asks again for a new variable.

The following commands are available when editing.

Keys	Command
→	Moves the cursor 1 ladder character to the left
←	The same, but to the right
↓	Moves the cursor down 1 ladder line
↑	Moves the cursor up 1 ladder line (1st key to the left, on the 2nd line of the display control keypad)
//	Erases a ladder character (input, output, branch) and advances by one ladder character
← ←	Erases the complete ladder line where the cursor is situated
][	Inserts a ladder line before the line pointed to by the cursor if the insertion is impossible, the key is disabled
#	Erases the whole ladder sequence and places the cursor at the beginning of the chart

The construction of the relay chart is made easier by automatic guiding in normal mode.

Example : The programming of the following sequence is required :



The sequence will be created by the following sequence series :

$$\|E_1 \_\_ \lceil \rightarrow /E_2 \_\_ \rfloor \rightarrow \|E_3 \_\_ \rightarrow \|E_4 \_\_ (\ )A_1 \_\_$$

When "end of branch" key is pressed, the cursor places itself automatically on the 2nd line, at the beginning of the branch. If entry or another branch is desired, it is sufficient to key in "⌈" ... etc ...

At the end of the last line of a branch, pressing key "→" moves the cursor up on to the 1st line ready for another contact or output.

To exit from the sequence editing mode, the "service" key (OUT) is used. An analysis of the syntax of the chart is carried out, an error message appears to locate the origin of any fault. The system places itself in editing mode again to enable the ladder to be corrected.

When the ladder is valid it is transferred to the sequence management mode and a message appears above the sequence number "COMMAND (I, D, C) : "

I : Calls for the insertion of a new sequence in the sector before a given sequence

D : Enables the sector sequence to be erased

C : Enables the sequence number and message to be modified

(OUT) : Normal exit, the sequence is placed at the end of the sector (if it did not exist beforehand).

Insertion : On keying in "I", SEQ is displayed. The sequence number in front of which one wants to insert a new sequence must then be keyed in. The system searches for this sequence in the sector and sends an error message if the sequence does not exist in the sector.

Erasing : On keying in "D", the system destroys the edited sequence from the sector.

Changing : On keying in "C", sequentially the system first asks the new sequence number, then the message. If the new sequence number is already used in the sector, the system sends an error message and asks again for a new number.

Normal mode : Key in (OUT). If the sequence already existed in the sector, it is replaced in its initial position (in the sector). If the sequence is new, it is placed at the end of the sector. The sequence order in the sector does not depend upon the number, but on the input order.

Once this sequence mode is carried out, the starting mode is returned to, where a new sequence can to be edited "SEQ : " is displayed and the control awaits the sequence number.

Other commands allowed at that level are :

↑ allows editing the sequence placed before the current sequence

↓ allows editing the sequence placed after the current sequence

INIT (LF) erases all sequences present in the sector

(OUT) allows an exit from the sector and to write the modified sector in the EEPROM

QUIT (LF) allows an exit from the sector without writing the modifications carried out.

When (OUT) key is used, the system tests whether the sequence jumps are valid. It will display an error message following the 2 possible errors.

1 - jumping to a sequence which does not exist in the sector

2 - jumping to a sequence placed before the sequence containing the jump. (Not allowed, as the PC is in danger of looping indefinitely and thus tripping the watchdog timer, as the Inputs/Outputs are not refreshed).

An upstream jump is enabled in the sub-programs and background tasks.

The error message indicates the sequence number from which the wrong jump came, as well as the type of error.

The system then returns to sector editing to allow the errors to be modified.

The sector is not written into EEPROM while errors exist.

When the sector is writing into EEPROM, the message "GRAVURE EN COURS"

Note :
To exit from the sector, without accepting the modifications made, key in "QUIT". The system returns to the PC programming menu without rewriting the EEPROM.

6.3.1.2. - Calculation sequence

The access command to a calculation sequence is identical to that of a ladder sequence, except that "C" type must be placed after the sequence number. The system places itself automatically at the beginning of the first calculation line.

One calculation operation only per line must be written (8 lines maximum per sequence).

The list of the operands allowed, as well as the calculation operations, are explained in the setting up manual (PC programming).

Do not place outputs before the transfer arrow (as A.x AN.x ANN.x outputs cannot be read) and do not place inputs after the arrow (E.x EN.x ENN.x inputs cannot be written).

To exit from the editing mode, the same command is required as for the ladder sequence "OUT".

If a calculation line is incorrect, the system displays an error message indicating the type of error and the variable concerned in the line (each character group is considered as a variable).

The sequence of command is identical to that of the ladder sequences.

When editing, the following editor commands are enabled :

⇇	erasing one line
//	erasing one character
][	inserting one line
↑↓	step up and down by one line
→←	displacement of one character on the same line
(LF)	places the cursor at the beginning of the next line.

6.3.2. - Table of constants : TC

This table contains constants which are written on EEPROM and can be called in sequences.

These constants are of 1 or 2 bytes.

1 byte → V. xx order number in the table
2 bytes → VV. xx " " " " "

Note :
A constant is always written on 1 or 2 bytes. When a 1 byte constant is called, it is written in the least significant of the double byte constant (V.0 is the least significant byte of VV.0 ...).

If one wants to use an EEPROM constant in a sequence, the system only accepts constants which already exist in the table, consequently the table of constants must be reserved before editing sequences.

The table of constants is called by TC (LF) in the main menu. The table is displayed in the following way :

TABLE DES CONSTANTES NOMBRE : 20

N°	DECIMALE	HEXA	ASCII
0	16658	4142	AB
1	257	0101	■■
2	.		
3	.		
4	.		
5	.		
6	.		
7	.		

TABLE OF CONSTANTS : NUMBER : 20

ORDER	DECIMAL	HEXADECIMAL	ASCII
0	16658	4142	AB
1	257	0101	■■
2	.		
3	.		
4	.		
5	.		
6	.		
7	.		

One display line thus gives :
1) order number of the sequence in the table
2) decimal value of the constant ($0 \leq C \leq 65535$)
3) hexadecimal value of the constant ($0 \leq C \leq FFFF$)
4) ASCII representation of the constant if it exists
(if a byte has no ASCII correspondance, ■ is displayed).

Input of a constant can be done in three ways; in all cases the (LF) constant number must first be asked for (eg. : 10 (LF)).

The system asks the value of the constant "VAL (# , $) : "

1) in decimal code — # xxxx (LF)
   value in decimal code
2) in hexadecimal code — $ xxxx (LF)
   value in hexadecimal code
3) in ASCII — 1 - if two ASCIIs are keyed in AB (LF)
   2 - if one ASCII only is keyed in B (LF)

(the system places a 0 in the most significant byte).

Search for a constant :
If to the question "VAL (#, $) : " (LF) is keyed in, the content of the constant is not modified, but is displayed on the screen.

Note :
If a constant number is outside the table, the system refuses it. The same applies for values which are not between 0 and 65535.

↑ : gives the next 8 constants of the table
↓ : gives the 8 preceding constants of the table
INIT (LF) : initialises all constants of the table to 0
(OUT) : exits from the table of constants.

6.3.3. - Table of messages : TM

The table of messages is called by TM (LF) in the main menu. The table is displayed in the following way :

```
        T A B L E   D E S   M E S S A G E S  (N B R E : 10)

   O R D R E              M E S S A G E

      0   M A C H I N E   E N   E T A T   D E   M A R C H E
      1   A L A R M E
      '
      '
      '
      7   M A N Q U E   D ' H U I L E
```

```
      T A B L E   O F   M E S S A G E S   ( N U M B E R : 1 0 )

O R D E R          M E S S A G E

     0   M A C H I N E   I N   R U N N I N G   O R D E R
     1   A L A R M
     '
     '
     '
     7   S H O R T A G E   O F   O I L
```

One display line gives :
1- order number in the table
2- the message (32 characters)

To input a message, it is sufficient to key in the number of the message 10 (LF). The message places a question-mark "?", it waits for the characters of the message (LF) (if a message is less than 32 characters, the system completes with spaces).

As in the table of constants, keying in (LF) after "?" is equivalent to a message search (displays the no. called).

↑ : displays the next 8 messages

↓ : displays the preceding 8 messages

INIT : places spaces in all messages

(OUT) : enables to exit from the table of messages.

6.3.4. - Table of labels TE and TE2

Two tables are available to the programmer : one in EEPROM (TE) to retain the labels in the memory, the other in RAM (TE2) to output listings and cross-references. A table written in one module can be transferred to the other module by command MERG (the module area must have been reserved before carrying out the command).

For example, labels for inputs (E), outputs (A), variables (B), etc... written in module TE2, will be transferred by selecting TE in the main menu, then writing MERG. The table will be transferred into TE but not destroyed in TE2.

All operands allowed in program writing can be input to the table of labels (with the exception of I. and II).

The label comprises a maximum of 8 characters and the operands are classified in increasing order of address.

These messages will appear when the ladder sequences are output on a printer.

The table of labels is called by TE or TE2 (LF) in the main menu. The table is displayed in the following way :

```
T A B L E   D E S   E T I Q U E T T E S   ( N B R E : 4)

  E T I Q U E T T E S              M E S S A G E S
  E . 1                              D E G   U R G
  E . 4                              A L A R M E
  A . 0                              B R O C H E
  A . B

 ┌T A B L E   O F   L A B E L S    N U M B E R : 4                  ┐
 │                                                                 │
 │  L A B E L S              M E S S A G E S                       │
 │  E . 1                    E M E R G E N C Y   R E T R E A T     │
 │  E . 4                    A L A R M                             │
 │  A . 0                    S P I N D L E                         │
 │  A . B                    S T O P                               │
 └                                                                 ┘
```

Two commands are possible :
1- Erasing of a label + message
2- Creation of a label + message or modification of a label.

For the first case, key in "D", the system asks which label one wants to erase. If the label corresponds, the system carries out the erasure and awaits further command.

For the second case, key in "E" (LF not required). The system requests the label input in symbolic language in the shape of E.xx (LF). The point between E and the value is mandatory (if the label is incorrect, the system asks again).

A label search can be carried out by keying in "E" then the label searched for (LF). When the message is called, it is sufficient to key in (LF) directly.

↓ : displays the 7 labels placed before or after the current page.
↑

INIT (LF) : initialises the table of labels to 0 (erases all labels).

(OUT) : exit from the label table.

6.4. - SOFTWARE IDENTIFICATION : IDEN

When the PC program has been input, it can be referenced and identified with command IDEN (LF).

The following page is displayed :

```
        N O   L O G I C I E L  /  I D E N T I F I C A T E U R

    N O   L O G ( % ) :
    I D E N :
    Z O N E   M E M O I R E   O C C U P E E : 5 8 1

┌           N O   S O F T W A R E / I D E N T I F I E R ┐
│                                                       │
│S O F T   N O ( % ) :                                  │
│I D E N :                                              │
│M E M O R Y   A R E A   O C C U P I E D : 5 8 1        │
└                                                       ┘
```

- Software number is entered by writing on the dialog line : % 8 characters maximum (LF).

- Identification with : IDEN ␣ 32 characters maximum (LF).

- The memory area occupied indicates the total size occupied by the PC program to the user, in the knowledge that this size cannot exceed :
  8191 basic bytes
  or 40959 bytes with the 40K PC option.

  If no program is in the memory, the area occupied is 581 bytes (value pertaining to the PC).

- (OUT) : allows an exit from identifier mode.

6.5. - BYTE OUTPUT CONFIGURATION : OCT (LF)

The PC outputs to the machine interface via the 32 output PCB, can be configured in bits or in bytes.

Outputs declared in bits are refreshed every PC cycle, whereas the numerical byte outputs are not.

The page is displayed as follows :

```
          C O N F I G U R A T I O N S   S O R T I E / O C T E T S

A N       0     1     2     3     4     5     6     7
          0     0     0     1     0     0     0     1
A N       8     9     A     B     C     D     E     F
          0     0     0     0     0     0     0     0
A N    1 0   1 1   1 2   1 3   1 4   1 5   1 6   1 7
          0     0     0     0     0     0     0     0
A N    1 8   1 9   1 A   1 B   1 C   1 D   1 E   1 F
          0     0     0     0     0     0     0     0

          [O U T P U T / B Y T E   C O N F I G U R A T I O N S]
```

(see above)

AN.0 = 0 means output 0 to 7 of the PCB addressed in 0-1 are configured in bits
AN.3 = 1 means outputs 18 to 1F of the PCB addressed in 0-1 are considered as a 1 byte numerical output.

To modify the configuration, it is sufficient to key in the output concerned. Example : AN.7 (LF).

The system then asks the status of this output (0 or 1). 0 or 1 must be keyed in followed by (LF). The system immediately outputs the modification on the display.

INIT (LF) reconfigures all outputs in bits (0 in the whole table)

(OUT) allows an exit from the output configuration mode.

6.6. - HARDWARE TASK DEFINITION : THD

The hardware tasks process the priority interrupts; hardware task 0 is the highest priority and hardware task 7 the least priority. An interrupt input is associated to each task :

- EN.3E Input status for serial port 1
- EN.40 Input status for serial port 2
- EN.42 to EN.45 Status word of the interrupts

One of these inputs can be assigned to each hardware task.

The page is displayed in the following way :

HARDWARE TASK CONFIGURATION

TASK	ADDRESSES
TH0	EN.3E
TH1	*****
.	.
.	.
TH7	*****

To assign a hardware task, its name TH1 (LF) must be keyed in, then the system asks for the symbolic address : EN.42 (LF).

If a false address is keyed in, the system sends an error message and initialises the hardware task with ****.

Modification display is immediate.

INIT (LF) : reinitialises the hardware tasks
(OUT) : enables an exit from the mode

6.7. - ACCESS TO THE PERIPHERALS : PERI

When the PC program has been entered in the memory, it is useful to file it on tape. This allows the program to be reloaded should there be incidents, or an identical machine to be programmed.

Likewise, it is worthwhile outputting the program on a printer to constitute the machine dossier.

To do this, the user can request the peripheral menu from the main menu, which displays the following sub-menu :

A C C E S A U X P E R I P H E R I Q U E S

- D E C H A R G E M E N T B I N A I R E S U R R U B A N . P E R F
- V E R I F. P E R F O R A T I O N B I N A I R E . . . . . V E R I
- C H A R G E M E N T B I N A I R E P A R R U B A N . . . . L E C T
- P E R F O. B I N A I R E D ' U N S E C T E U R . . . . . P E R F S
- C H A R G. B I N A I R E D ' U N S E C T E U R . . . . . L E C T S
- I M P R I M E R T O U S L E S S E C T E U R S . . . . . L I S T
- I M P R I M E R C R O S S - R E F E R E N C E . . . . . C R E F

A C C E S S T O T H E P E R I P H E R A L S

- B I N A R Y P U N C H O U T P U T T O T A P E ..... P E R F
- B I N A R Y P U N C H V E R I F I C A T I O N ..... V E R I
- B I N A R Y L O A D I N G B Y T A P E ..... L E C T
- B I N A R Y P U N C H I N G O F A S E C T O R ... P E R F S
- B I N A R Y L O A D I N G O F A S E C T O R ..... L E C T S
- P R I N T A L L S E C T O R S ..... L I S T
- P R I N T C R O S S - R E F E R E N C E ..... C R E F

To leave this menu, press "service" key which returns to the main menu.

6.7.1. - Binary output to tape : PERF

The system transfers the whole content of the EEPROM on the tape in binary. During the whole of the transfer, the message "PERFO EN COURS" ..... ["PUNCHING IN PROGRESS"] is displayed.

1 m trailer and header with sprocket is automatically punched before and after the program on the tape. Output can be interrupted at any time by pressing "service" key, but it cannot be resumed at the location of the stop.

6.7.2. - Binary punch verification : VERI

When the program has just been output in binary to tape, its contents must be checked with respect to the content of the EEPROM.

With the tape in position on the tape-reader, key in VERI (LF). Tape reading proceeds and "VERIFICATION EN COURS" ["VERIFICATION IN PROGRESS"] is displayed.

If a punch error is detected, the tape-reader stops and "MAUVAISE PERFORATION DU RUBAN" ..... ["INCORRECT TAPE PERFORATION"] is displayed.

The program must be punched again, or there is a tape-reader fault.

If no error is detected, "RUBAN BINAIRE CONFORME" ..... ["BINARY TAPE CONFORMS"] is displayed at the end of read mode. The contents of the tape and of the EEPROM memory are identical.

6.7.3. - Binary-loading by tape : LECT

Command LECT (LF) allows the whole PC memory to be loaded directly in binary. In this case, memory reservations for the various tasks or tables need not be entered (automatically done by the tape).

Command LECT (LF) reads the tape and displays "GRAVURE EN COURS" ..... ["WRITING IN PROGRESS"] on the dialog line. At the end of read mode, the dialog line is erased and the menu returned to.

Should there be a transfer fault or incorrect tape reading, the systems sends an error message "ERREUR CHECKSUM" ..... ["CHECKSUM ERROR"] and the EEPROM must be reinitialized (INIT).

6.7.4. - Punching a sector in binary : PERFS

The principle is different from the punching of a complete program. In fact the sector is punched from the intermediate editing buffer. When the system acquires command PERFS (LF), it requests the sector "SECTOR". All sectors are accessible : TS0 .. TS4 , TF0 .. TF15 , SP0 .. SP32 , TH0 .. TH5 , TC , TE , TH.

When the system starts to punch, it places a character representing the type of sector at the head of the sector.

```
TS0 .. 4      → type = 'S'
TF0 .. 15     → type = 'F'
SP0 .. SP32   → type = 'S'
TH0 .. TH7    → type = 'S'
TC            → type = 'C'
TM            → type = 'M'
TE            → type = 'E'
```

then a checksum, followed by a checksum length and the sector. During punching "PERFO EN COURS" ..... ["PUNCHING IN PROGRESS"] is displayed on the dialog line. The system punches 1 m leader and trailer upstream and downstream from the program on the tape.

6.7.5. - Sector loading in binary : LECTS

The same principle as for punch output, the system asks for the sector in which to place the program located on tape.

The system reads the sector name on the tape, to make sure that the sector program and the designated sector are indeed of the same type.

If the types are incompatible (trying to load a TC in TM, TF in SP, etc ...) or if the transfer is incorrect, the system sends a "ERREUR DE CHECKSUM" .. ["CHECKSUM ERROR"]
During loading of the tape in the RAM, the system displays "CHARGEMENT EN COURS" ..... ["LOADING IN PROGRESS"]. When loading is finished, the system places itself in the sector in edit mode. One is then in the same context as the editing of a sector. It is sufficient to key in (OUT) to then transfer this program into the EEPROM.

6.7.6. - Sector output on a printer : LIST

This command allows outputting all sectors to a printer (TSx, THx, SPx,

Writing LIST (LF) displays on the dialog line message :

"TOUT OU PARTIE" ..... ["ALL OR PART (ALL/SELECT) ?"]

- Output of all sectors present by answering "A (LF)". "LISTING EN COURS" ..... | "LISTING IN PROGRESS" | is displayed on the dialog line until printing is ended.

- To output one or several sectors, answer S (LF). The system will automatically present the sectors used one after the other.
  The first question after answer S is always :

  "LISTING IDENTIFICATEUR" ..... ["IDENTIFIER LISTING ?"]

  If a listing is requested, answer (LF). "LISTING EN COURS"["LISTING IN PROGRESS"]is displayed until printing is ended, then return to the following listing request with :

  LISTING SECTEUR () ? ..... [SECTOR LISTING () ?]

  () name of the first sector which is present (TSO for example). If the sector is not to be printed, pressing ← key causes a skip to the following sector. Answer (LF) prints the sector.

  Sector printing or searching can be interrupted at any time by pressing "service" key.

Note :

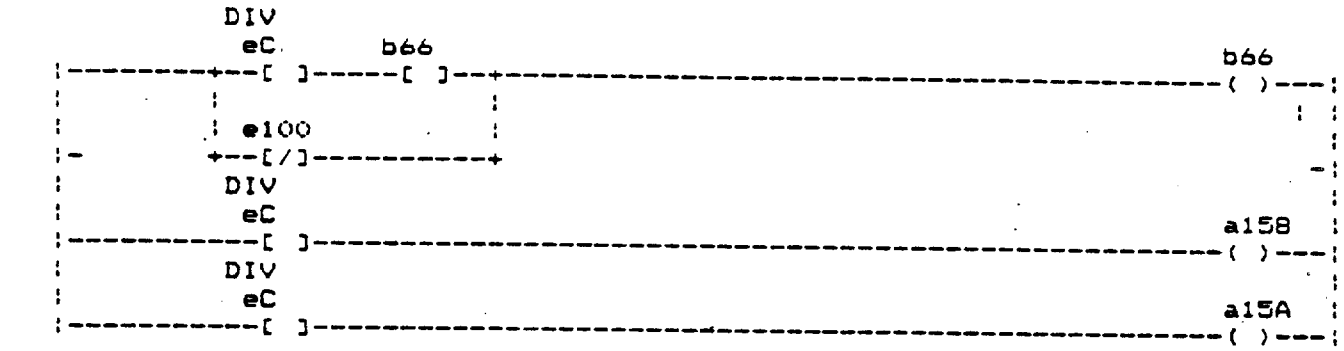
When utilities 6 and 7 are used simultaneously, only sectors written in utility 6 can be printed (except TE, TE2, TC, TM).

NUM 760

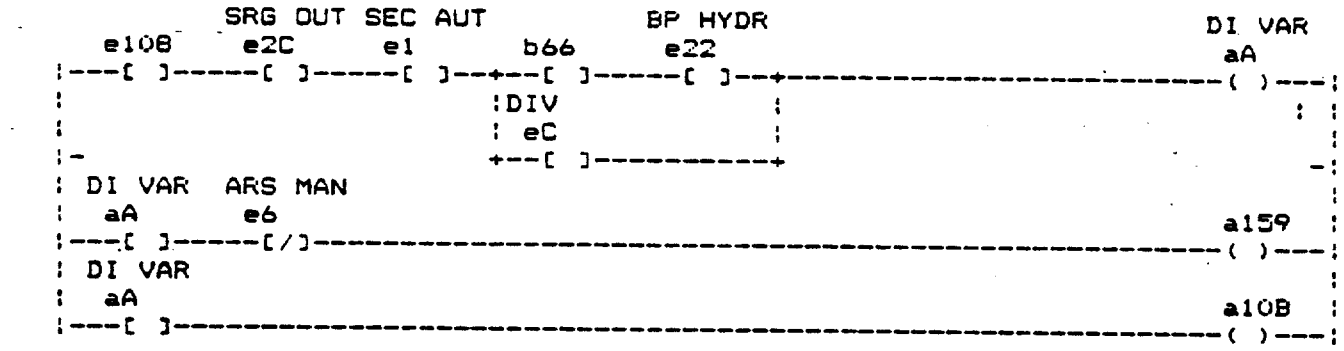
Listing example :

TSO SECTOR

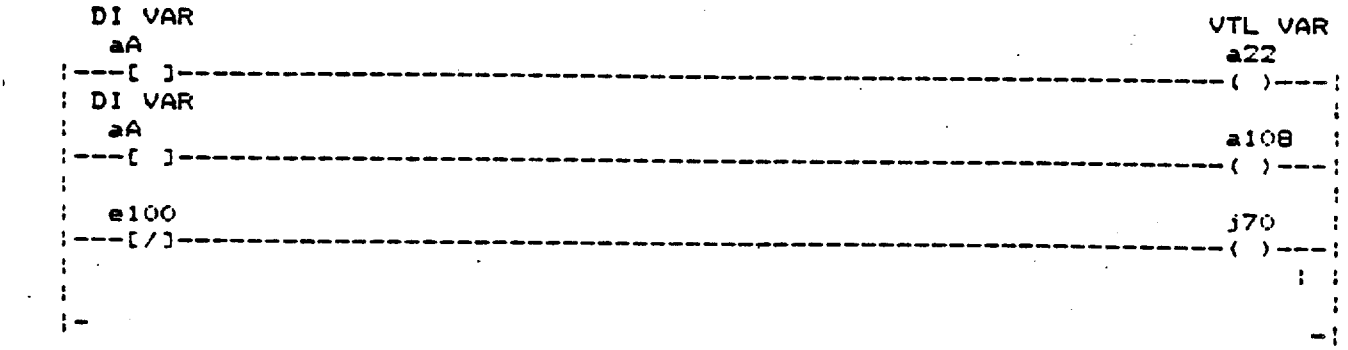
SEQ:30 TYPE:R MESS:RZ MACH



SEQ:40 TYPE:R MESS:DI VARS



SEQ:50 TYPE:R MESS:

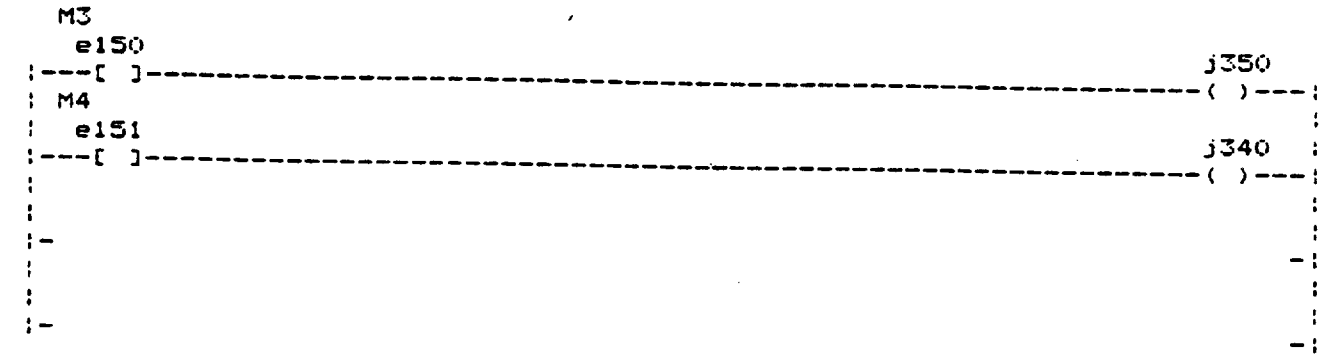


SEQ:60 TYPE:C MESS:RZ CNA B

$.0 ==> AN.30

NUM 760

SEQ:330 TYPE:R MESS:M3/M4



SEQ:340 TYPE:C MESS:SIGN M4

$.0 ==> AN.32
==> J.360

SEQ:350 TYPE:C MESS:SIGN M3

$.1 ==> AN.32

SEQ:360 TYPE:C MESS:TENS BRO (SPINDLE)

EN.2A LSL ==> M.10
M.10 LSL ==> M.10
EN.2B LSR ==> M.11
M.11 LSR ==> M.11
M.11 LSR ==> M.11
M.11 LSR ==> M.11
M.11 LSR ==> M.11
M.11 LSR ==> M.11

NUM 760

SEQ:410 TYPE:C MESS:

==> J.440

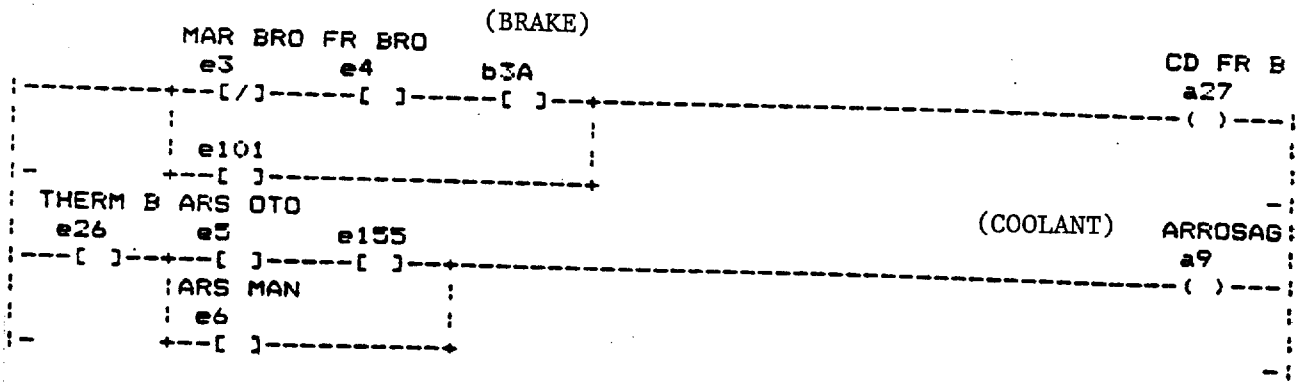
SEQ:420 TYPE:C MESS:

$.0 ==> M.16
$.0 ==> AN.30
$.8 ==> M.15

SEQ:425 TYPE:C MESS:ROT 100T

$.7 ==> AN.30
1 ==> B.39

SEQ:440 TYPE:R MESS:CD FREIN



6.7.7. - Output of cross-references on the printer : CREF

For the programmer or the user it is useful to know in which module and sequence the various variables, memories, inputs, outputs ... are created and used, as well as the mnemonics associated with them. To do this, CREF (LF) command allows printer output of these variables.

Variables with mnemonics are classified alphabetically and then the variables without mnemonics are classified by type and increasing number.

For cross-references to be output with their mnemonics, the latter must have been programmed in the table of labels TE2.

Note :

- To obtain the list of variables with or without mnemonics, classified by type of variable and increasing number, module TE2 must be printed by LIST (LF).

- In the case of a program written in utility 6 and 7, cross-references will only concern modules written in utility 7.

- Most printers use the 8th bit (parity bit) of the characters received for selection of another set of characters.

  Sending this parity bit limits printer possibilities; to avoid this the tape-punch and listing outputs are henceforth transmitted without parity bit.

Example :

MNEMONIQUE MNEMONIC	ETIQUETTE LABEL	GENERATION GENERATION	UTILISATION USE
AC AAX+	E.22		TS0 S9006
AC AAX-	E.23		TS0 S9006
AC AAZ+	E.24		TS0 S9006
AC AAZ-	E.25		TS0 S9006
AOB VAR	A.1	TS0 S20	TS0 S20 S60 S9000 S9001 TS0 S9002 S9005 TS1 S1
BCD VAR	E.28		TS0 S3800
BC ROT	E.29		TS0 S100 S3805 S3835 TS0 S3850 TS1 S1100 S1130
BLD VAR	A.F	TS4 S105 S110	
BO MOT	A.2	TS0 S55 S100 S3825 S3830 S3840 S3845	TS0 S15 S3805 S3835
CY ADEP	A.14	TS0 S55 S100 S3520 S3574 S3580	TS0 S3566 S3568 S3580
EC MOT	E.11		TS1 S3610
EO AV	A.A	TS0 S55 S100 TS1 S3605 S3625 S3630	

MNEMONIQUE MNEMONIC	ETIQUETTE LABEL	GENERATION GENERATION	UTILISATION USE
VAR INT	B.12	TS0 S250 S3725 TS4 S1250	TS0 S250 TS1 S1225 S4310
	E.13		TS0 S3028 S3036 S3056
	E.1E		TS3 S2015 S2130
	E.26		TS3 S2040 S2102
	A.9	TS0 S55 S100	
	A.D	TS0 S55 S100	
	A.E	TS0 S55 S100	
	A.15	TS0 S55 S100 S8011 S8013	
	A.16	TS0 S152 S3236 S3250	TS0 S3212
	B.10	TS0 S3805	TS0 S1105 S3003 S3104 TS0 S3212 S3502
	B.11	TS0 S1000	TS0 S3024 S3044 S3162
	B.27	TS0 S155 S3113 S3114 S3115 S3124	TS0 S3168 S3174 S3176
	B.28	TS0 S155	
	M.12	TS0 S3210	TS0 S3210
	M.14	TS0 S8001 S8005	TS0 S8002 TS1 S1005 S1010
	M.15	TS1 S1010	TS0 S8002
	EN.20		TS0 S3560 S9002
	EN.23		TS0 S3005 S3112 S3510 TS0 S4010 S8010
	AN.22	TS4 S105 S120	
	AN.23	TS4 S105 S120	
	AN.24	TS0 S9050	
	C.2	TS0 S8002 TS1 S1131 S1132	TS1 S1132
	HC.2	TS1 S3902	TS1 S3902
	QC.2	TS0 S55 TS1 S3900 S3903	TS1 S3902 S3904
	EN.30		TS0 S9050
	EN.31		TS0 S9040
	V.1		TS0 S8000 TS1 S1000 S1016
	V.3		TS0 S3200 S3210 S3230
	V.4		TS0 S3000 S3004 S3028

6.8. - EQUIVALENCE TABLE : EQUI

When one or several modules have been written with certain functions having different variables, it is possible to modify these to make a program compatible with different types of machines (E.100 is transformed into E.23 for example).

All variables can be modified : E, A, M, T, C, ... The maximum number of variables which can be modified is 512.

When the program has been input to the memory by loading a tape in binary, key in command EQUI (LF) in the main menu.

The following page is displayed :

TABLE D'EQUIVALENCE NOMBRE :

[EQUIVALENCE TABLE NUMBER :]

E.100 / E.23
A.120 / A.112
M.25 / M.33

The editing commands available are :

- "E"dit : equivalence input
- "D"elete : equivalence destruction
- "I"nit : table initialization

To input an equivalence, the operator keys in E (LF). "ADR" is then displayed on the dialog line. The operator keys in the variable followed by (LF), the radical of the same type of variable is displayed on the dialog line. The operator then types the new indice followed by (LF).

Example :

E (LF)		keyed in by the operator
E ADR:		displayed on the dialog line
E ADR:E.100 (LF)		E.100 (LF) keyed in by the operator
E ADR:E.100	E.	displayed on the dialog line
E ADR:E.100	E.112 (LF)	112 (LF) keyed in by the operator

Equivalences are classified in increasing address order.
Deletion of an equivalence is done by means of command "D" (LF) "ADR" : is then displayed on the dialog line. The operator indicates after that the variable he wishes to delete.

To search for an equivalence, the operator proceeds in the same way as for an input but after having indicated the variable required, key in (LF) (LF).

When the equivalence of all the variables to be modified have been input into the table, select the first module to be modified from the main menu. When it is displayed, key in EQUI (LF). The variables within that module will then be modified.

This operation can only be carried out module by module.

With display of the equivalence table page :

- command INIT (LF) erases the whole table, command ↓ causes advancement of one page,

- the equivalence table can be output on tape in binary by means of command PERFS in "PERI" menu, or loaded by a binary tape by means of command LECTs,

Section 6.9 - Program debug

To be inserted later.

7. - UTILITY 10 : LPC PROGRAMMING

Loading, display, modification or debugging of the LPC PCB program requires utility 10 to be stored in the memory. To load the utility see paragraph 1.2.

Implementation of this utility is strictly identical to that of utility 6, but calling utility 10 is done by means of 10 (LF) in the utility menu.

For hardware tasks, assignment of the first four tasks is fixed (TH0 to TH3), and it is not possible to modify them.

TH0	EN.40	Line TTL
TH1	EN.41	Serial line 1
TH2	EN.42	Serial line 2
TH3	EN.43	Serial line 3

Assignment of other hardware tasks is defined by the programmer.

8 - UTILITY 11 : EEPROM FILING

It allows safeguard of binary information (data and software) stored in EEPROM memory.

Punching tapes is mandatory at the end of machine integration or after each parameter modification. Punched tapes must remain permanently with the machine.

Loading operation is carried out when PCBs are replaced : personalization, PC memory extension, PC processor or NC software change.

The business number is specified on all tapes in order to be able to check conformity when loading.

Tapes are classified according to two types :

- information concerning the NC :
  . Software number, option grid and key
  . Parameter type and table
  . Measurement compensation
  . Resident sub-programs

- Information concerning the PC :
  . PC program binary.

8.1. - IMPLEMENTATION

After selecting 11 (LF) in the menu of the utilities, the following menu is displayed :

```
ARCHIVAGE EEPROM  .....                 .  [ EEPROM FILING                  ]

0 CHARGEMENT INFORMATIONS CN  .....        | NC INFORMATION TAPE READ
1 CHARGEMENT INFORMATIONS AUTOMATE .....   | PC INFORMATION TAPE READ
2 DECHARGEMENT INFORMATIONS CN       ..... | NC INFORMATION TAPE PUNCH
3 DECHARGEMENT INFORMATIONS AUTOMATE ..... | PC INFORMATION TAPE PUNCH
4 VERIFICATION RUBAN CN .....              | NC TAPE CHECK
5 VERIFICATION RUBAN AUTOMATE .....        | PC TAPE CHECK               ]
```

The heading is chosen by keying in the corresponding number followed by (LF).

A heading is output by pressing "service" key, which returns to the utility menu.

8.2. - LOADING NC INFORMATION

To load the NC tape it must be specified whether this is a defective EEPROM or an update of the NC software, Indeed, information is not stored in the same way in both cases :

- Faulty replacement (Exhchanging the faulty EEPROM of the personalization PCB with a standard EEPROM); tape information is stored in the EEPROM.

  The machine is operational with the maximum option grid, but whenever it is switched on the following message is displayed on the screen :

  PERSONNALISATION NON DEFINITIVE .....
  SYSTEME HORS GARANTIE .....
  CONTINUATION (O/N)

  [PERSONNALIZATION NOT FINAL
  SYSTEM OUTSIDE GUARANTEE
  CONTINUE (Y/N)]

  Answer Y (LF) displays page "current point".

  The aim of this message is to indicate to the user that he is working under exceptional conditions.

  Complete personnalization is required and a new EEPROM will be sent to the user.

  This must be placed in the location of the previous EEPROM and new copies of punched tape must be made.

  The standard EEPROM must be sent back to NUM.

  - Updationg the NC software can bring about restructuring of the EEPROM. New information on the tape is stored in the memory instead of the older version.

  If the parameter table type is different on the tape to the EEPROM being read, the operator must command either erasing or safeguard of the EEPROM values.

  In that case, the following message is displayed : "TYPE TABLE PARAMETRES RUBAN DIFFERENT DE CELUI EEPROM - CHARGEMENT ANCIENS PARAMETRES ?" ..... ["TAPE PARAMETER TABLE TYPE DIFFERENT FROM THAT OF THE EEPROM - LOADING OLD PARAMETERS ?"].

  If the answer is Y (LF), the EEPROM information are replaced by those of the tape.

  If the answer is N (LF), the parameters present in the EEPROM are unchanged.

  Measurement compensations are modified axis by axis in all cases. For each axis, the number of points corrected will be the minimum number of the tape or EEPROM. Values stored in the memory are those of the tape.

For example :

	TAPE	EEPROM		
Axis 0	5 points	EEPROM	3 points	the first 3 points on the tape are stored in the memory
Axis 1	3 points	EEPROM	5 points	the 3 points of the tape are stored in the memory, the other 2 are lost.

This can only be done if measurement compensation modifications have been made and no tape has been punched.

- Implementation : Place the tape on the tape-reader and select "READ".

Menu is selected, key in Y (LF) on the NC keyboard.
Message "MISE A JOUR (O/N) ?" ..... ["UPDATE (Y/N) ?"] is displayed.
Answer Oui or Non ..... Yes or No depending on the case.
Display of message "PRET ?" ..... ["READY ?"]
Answer Y (LF) which reads the tape. If it is an update, message "TYPE TABLE PARAMETRES RUBAN DIFFERENT DE CELUI EEPROM ..." ..... ["TAPE PARAMETER TABLE TYPE DIFFERENT OF THAT OF EEPROM... "] will be displayed if the content of the EEPROM is different from that of the tape.
Depending on whether loading is required (see above) answer Y or N (LF).

At the end of read-in, the menu is reselected.

If the business number written on the tape is different from that stored in the NC, message "NO AFFAIRE INCORRECT" ..... ["BUSINESS NO INCORRECT"] is displayed.

8.3. - PC INFORMATION LOADING

The PC programs are safeguarded in EEPROM. Restitution is automatic and complete.

Menu is selected, key in 1 (LF)
Display of "PRET ?" ..... ["READY ?"]
With the tape on the reader and the latter set to "READ", answer Y (LF) brings about read mode and writing in the PC program memory.

At the end of read-in, the menu is reselected.

When the PC information is processed, three faults may be displayed :

- "NO AFFAIRE INCORRECT" ..... ["INCORRECT BUSINESS NO".] The tape which is read does not correspond to the business number of that system.

- "ZONE MEMOIRE INSUFFISANTE" ..... ["MEMORY ZONE NOT SUFFICIENT"] : Information on the tape cannot be completely recorded in the place available on the EEPROM. PC hardware configuration has had to be modified : PC 40K extension absent.

- "DEFAUT CHARGEMENT AUTOMATE" ..... ["PC LOADING FAULT"] : One character at least was not correctly read. Resume the operation. If the fault persists, it may be the tape-reader, the NC-tape-reader link, or the punched tape faulty and not checked when punched.

8.4. - UNLOADING INFORMATION FROM THE PC

Having selected the type of tape information and the punch being ready, tape punching occurs as soon as the operator has keyed in Y (LF) in answer to question "PRET ?" ..... ["READY ?"].

No other command is necessary. At the end of punching, return to the utility menu.

One meter length of blank leader is automatially created before and one meter of blank trailer after punching the tape.

8.5. - CHECKING THE NC OR PC TAPE

After punching conformity of the punched tape with the content of the EEPROM memories must be checked before filing.

Place the tape on the tape-reader and select "READ".

Select the type of check (NC or PC). Answer Y (LF) to question "PRET ?" ..... ["READY ?"].

Tape read-in and display of "VERIFICATION EN COURS" ..... ["VERIFICATION IN PROGRESS"]:

- If the tape is correct, return to the menu at the end of read-in.
- If a fault is detected, display of "RUBAN DEFECTUEUX" ..... ["DEFECTIVE TAPE"]message.

Resume the operation from the beginning because the fault might have been generated by dust on the tape.

If the fault persists, check whether the tape is correct, if yes punch a new tape and check again.

9. - UTILITY 13 : UTILITY LOADER

This utility is designed to load utilities whether they are segmented or not. (Utility 7 more particularly).

A segmented utility will be in several tapes, each tape bearing readable information of the size of the segment and its number.

A utility supplied in the shape of three tapes will, for example, bear 1/3 for the first segment, 2/3 for the second segment and finally 3/3 for the third segment. Loading will be done in order of segmentation.

When the utility is input the following menu is displayed :

```
1  SORTIE CHARGEUR .....          [PUNCH OUTPUT]
2  CHARGEMENT RUBAN .....         [TAPE READ   ]
```

- When heading 1 is chosen, return to page current point without utility erasure.

- When heading 2 is chosen display on the screen of :

```
ATTENTION !  COUPURE DE LA PUISSANCE .....  [WARNING !  POWER OFF]
OK (O/N)D :                                 [OK (Y/N) :           ]
```

  This sentence is displayed to indicate to the operator that machine power off will be provoked when loading the segmented utility.

  Answer N (LF) returns display to the utility menu.

  Answer Y (LF) means that power off information is taken into account and brings about display of :

```
LECT PRET ? (O/N) .....        [READ READY ?  (Y/N)]
```

  If the power off has already occurred before loading, this sentence will not be displayed on the screen and display will step directly to :

```
LECT PRET ? (O/N)              [READ READY ?  (Y/N)]
```

  If the tape-reader is loaded with segment 1 of the utility, or the complete tape if the latter is not segmented, read mode will be started by answer Y (LF).

PLACER SEGMENT SUIVANT SUR LECTEUR .. [PLACE NEXT SEGMENT ON THE TAPE-READER
PRET (O/N) ? READY (Y/N) ?]

This question will not be displayed if the utility being loaded is not segmented.

It will not be possible to load utility 0 or any other utility tape containing Utility 0 by means of utility 13. In that case, the operator will carry out loading via the usual method.

Fault reading during loading will provoke the same error diagnostics for a loading carried out by the usual method. If the order of segment loading is not observed, diagnostic ERR SEQ appears.

Furthermore, after any error display during read mode, the operator will restart reading of the segment at fault after having answered 0 (LF) to the question "REPRISE SEGMENT COURANT ?" ..... [RESUME CURRENT SEGMENT].