CONAIR
www.conairnet.com

USERGUIDE

Programming CNS900++ Level I

Version 0.6

⚠ **WARNING - Reliance on this Manual Could Result in Severe Bodily Injury or Death!**

This manual is out-of-date and is provided only for its technical information, data and capacities. Portions of this manual detailing procedures or precautions in the operation, inspection, maintenance and repair of the product forming the subject matter of this manual may be inadequate, inaccurate, and/or incomplete and cannot be used, followed, or relied upon. Contact Conair at info@conairgroup.com or 1-800-654-6661 for more current information, warnings, and materials about more recent product manuals containing warnings, information, precautions, and procedures that may be more adequate than those contained in this out-of-date manual.

- CONTENTS -

I - SOFTWARE ARCHITECTURE

I - 1. PART PROGRAM - MP00 to 99

One hundred 1,000-step main routines (N° 0 to N° 99) can be run and stored in memory. Simultaneous storage depends on available EEPROM capacity and the size of the routines.

Programs can be "named" (maximum 30 characters) in order to identify them with the product being handled:

Example: "MP 59 Shield X57"

It is possible to modify this name and display it during program search procedures. Characters can be entered via the keypad (or PC option). The alphabet is marked on the programming keys.

Structure is sequential, i.e. a step is not considered finished, and therefore the subsequent step cannot be run, until all the commands it contains have been executed.

The main programs will be selected from the keyboard when the robot is stopped.

I - 2. SUBROUTINE SEQUENCES - SP01 to 40 -

Standard subroutines are SP01 to SP40. These are a series of instructions which are grouped together in independant stuctures and run sequentially.

Like the main programs, they can be "named" (maximum 30 characters) in order to identify them with the function in the program:

Example : "SP 01 : Gripping part in the mould"

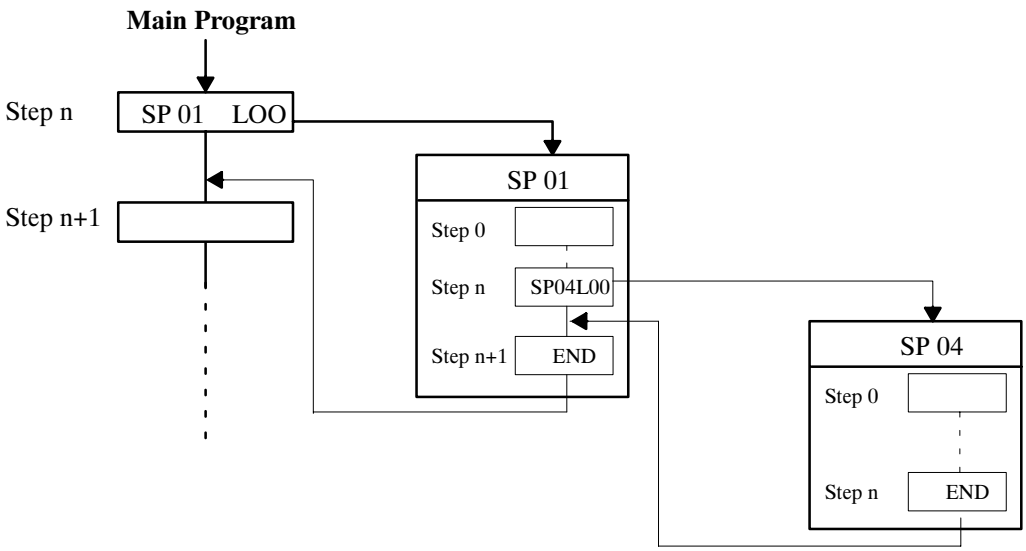
They can be run conditionally or unconditionally, depending on the declaration made by the programmer.

The address to which they return when they have been run is declared in the main program by programming a LABEL number. If the declared label is 00, when completed, the subroutine will return to the step in the routine after the call step.

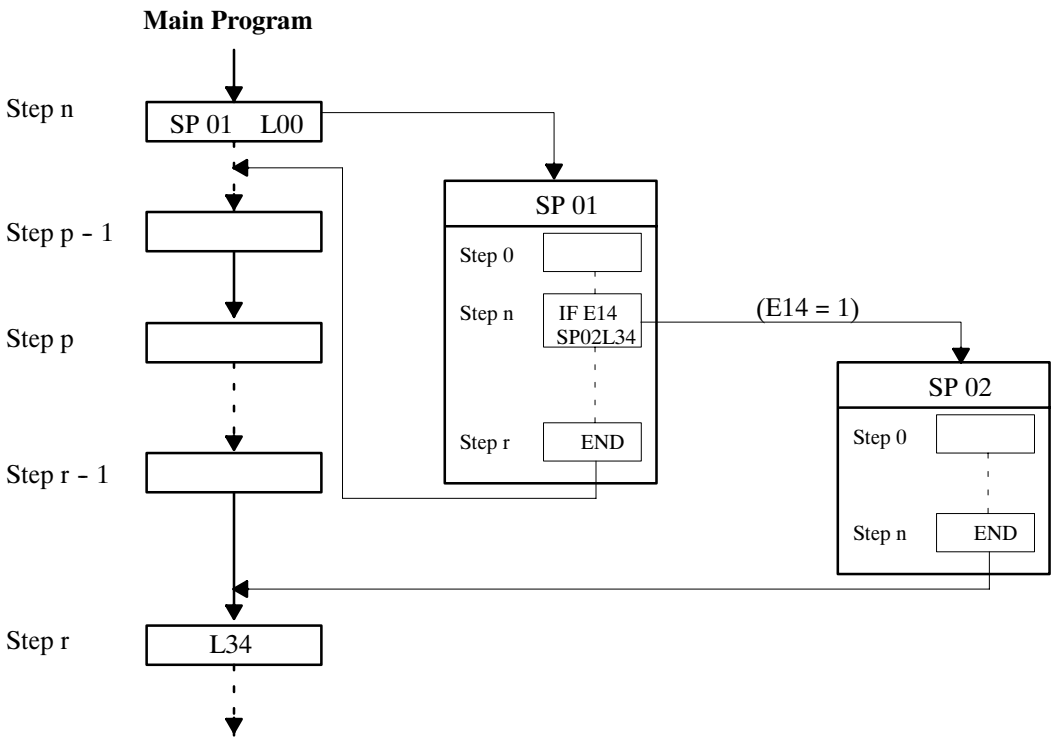
A subroutine may launch another subroutine, up to a limit of 3 levels of cascading.

If the label is not 00, it must be explicitly declared in the main routine.

Example No 1 :



Example No 2 :

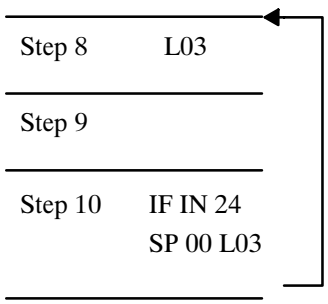


Note that this structure allows to use a same subroutine with different return addresses: the return address is an integral part of the call instruction and not of the subroutine. **It always goes to a main program step irrespective of the starting point (MP or SP)**.

Conditional running involves use of the "IF" instruction which is described in the section on instructions for the part routine. (Chapter II-7).

I - 2. 1.Subroutine SP00 used as go to instruction

Some programs require jump instructions to return to, or to jump to, a given STEP.



When using the instruction SP00 Lnn (01 to 99), it is not necessary to "create" the SP being used as required with SP 01, 2, 3, and so on. This takes up less memory space and reduces running time.

Note : The label L 00, for example SP00, is invalid.

I - 2. 2.Stacking subroutines - SP 41 to SP 80

SP 41 to SP 60 : These subroutines are used to simplify the declaration of pallets where a stack/column organization already exists.

SP 61 to SP 80 : These subroutines are used to describe an irregular stacking of parts which is repeated over several layers or several times in a cycle.

See Programming Manual Level II.

I - 2. 3.Personalized message subroutines - SP 100

The subroutine SP 100 is a structure that activates personalized messages which have been pre-defined with the PC program editor.

The message displayed following input errors will be that entered in the SP 100. In the case of absence of SP 100 or personalized message for the input error, the system message will be displayed.

To create an SP 100, consult the user manual of the PC program editior.

I - 3. HOME RETURN -SR00 to 99-

I - 3. 1.Home return subroutine - SR00 to 98 -

"Home Return" is a subroutine that takes the robot to a safe position before launching the "Automatic" cycle.

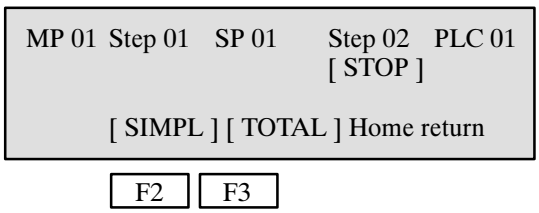
These subroutines are series of sequential instructions called up by the home return procedure *(see operating manual)*. The number of the return subroutine activated depends on the active step in the main routine (or subroutine) at the moment the request is made. The usual abbreviation for home return is "RO".

Subroutines SR00 to SR98 are valid for each main routine.

Like the main programs, it can be "named" (maximum 30 characters) in order to identify it with the function in the program:

Example: "SR 35: CLEAR SCALE"

Display during selection:



I - 3. 1.a)[Simple] home return

This procedure is used by default after execution of a subroutine (implicit: SR00 / explicit: SR01 to 98), and returns to the main program:

. if the Address is implicit, at step 00 of the current program,

. if the Address is explicit, at the step marked by a Label R bearing the same number as the return subroutine requested.

See example on page 6.

I - 3. 1.b)[Total] home return

After selecting key [F3], this procedure executes the subroutine: implicit SR00 / explicit: SR01 to 98.

It will always return to step 00 of the main program.

Caution: The stacking subroutines (See Programming Manual Level II) will be set at zero.

See example on page 6.

I - 3. 2.Tool change subroutine - SR 99 -

The usual abbreviation for a Tool Change is "PCO". This procedure is used to disengage the robot in order to change the gripping tool or to transfer the robot to a non-operational area in order to operate the host machine without the robot. This position usually corresponds to the robot location where the end of the PCO stroke is on the PCO cam.

Performing this procedure in step-by-step mode, after pressing the TOOL CHANG. key, initiates the following sequences:

- Home return sequence programmed at the step at which the robot stopped. This is SR00 by default.

then:

- the Return to Tool Change Position sequence.

Note:

1) These two sequences follow one after the other without having to release the "Cycle Start" button.

V 0.6

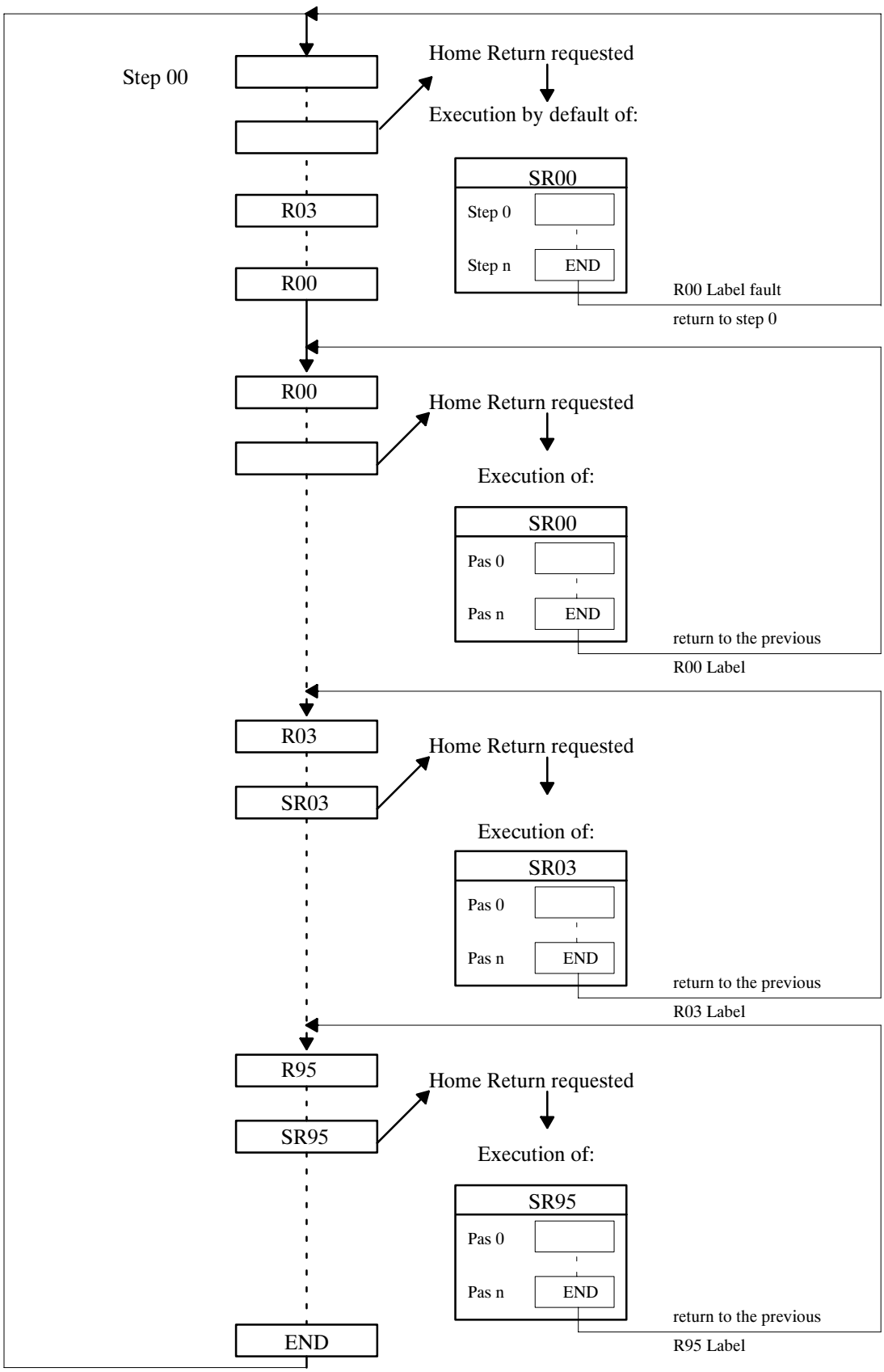
2) To simplify the NO ROBOT start, a parametered output can be programmed (PARAMETER 103) which can be used for operations without robot (in conformity with the interconnexion standard EUROMAP 12).

It is therefore unnecessary to position the mode selector to . This output cannot be activated in the main program.

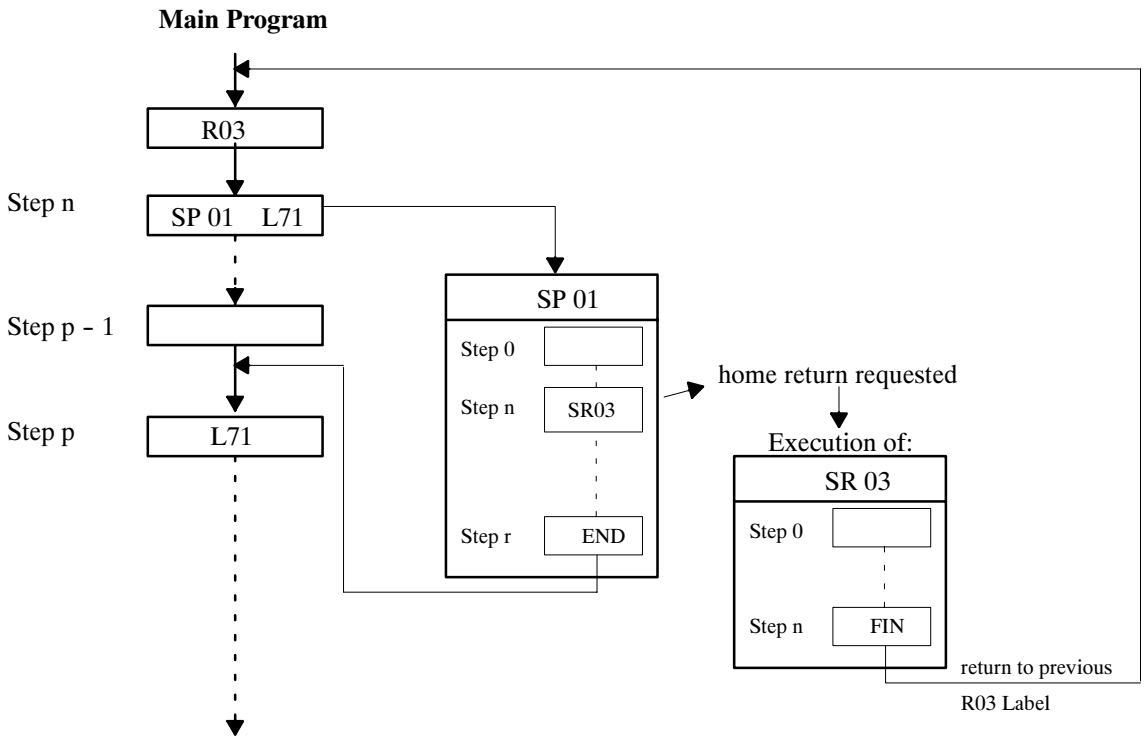
This function requires specific cabling. If your robot is not equipped for this, contact our After Sales Service.

I - 3. 3.Different Examples of Home return

Example No 1: Starting from the routine



Example No 2: Starting from the subroutine



Note that these structures allow a single subroutine to be used with different return addresses; they always go back to a main program step irrespective of the starting-point (MP or SP).

WARNING: Execution of Home Return subroutines is sequential and cannot be restarted from the beginning until they have been completed.

II - PART PROGRAM INSTRUCTIONS

There are basically four different types of instruction in the part program:

- ACTION (ACT) codes are preprogrammed functions giving the user easy command of movements without having to worry about activated physical outputs or their related checks.
- Operation concerning output actuation, bit or input test, binary state test counter handling, or delay times.
- Numerical axis movements.
- Calls, characterizations and declarations of subroutines.

II - 1. ACTION CODES

Syntax: > ACT.nn (nn = 00 to 32 or 90).

These codes are used for predefined bistable pneumatic command. Each one is associated with:

. An output set to 1 for actuation of the movement to be performed.

. A control input for its correct completion.

. An output set to 1 to stop the opposite movement.

. A control input for its correct completion.

Codes are associated as follows:

ACT.02 <-> ACT.04 => Rise 1 <-> Lowering 1 } Case of
ACT.03 <-> ACT.05 => Rise 2 <-> Lowering 2 } pneumatic Z

ACT.15 <-> ACT.02 => Slow lowering <-> Rise 1 } Case of pneumatic Z

ACT.06 <-> ACT.08 => Slow approach Y <-> Reverse Y } Case of
ACT.07 <-> ACT.08 => Normal approach Y <-> Reverse Y } pneumatic Y

ACT.09 <-> ACT.10 => Rotation 1 Horiz. <-> Vertical

ACT.11 <-> ACT.12 => Grip part 1 <-> Part 1 released
ACT.19 <-> ACT.20 => Grip part 2 <-> Part 2 released

ACT.21 <-> ACT.22 => Grip part 3 <-> Part 3 released
ACT.23 <-> ACT.24 => Grip part 4 <-> Part 4 released

ACT.25 <-> ACT.26 => Grip part 5 <-> Part 5 released
ACT.27 <-> ACT.28 => Grip part 6 <-> Part 6 released

ACT.29 <-> ACT.30 => Grip part 7 <-> Part 7 released
ACT.31 <-> ACT.32 => Grip part 8 <-> Part 8 released

ACT.13 <-> ACT.14 => Rotation 2 + <-> -

ACT.16 <-> ACT.13 => Rot 2: 90 degrees in + direction
ACT.16 <-> ACT.14 => Rot 2: 90 degrees in - direction

ACT.17 <-> ACT.18 => Free

Notes :

* Prior to being executeded, ACT.04 and ACT.05 are subject to safety tests, such as “machine open before lowering”.

* Completion of a slow approach motion, as in ACT.06, results from loss of contact “FAL” (End of slow-speed approach), part presence “MP”, or mechanical limit of the motion controlled by the associated input.5 (Identical principle to the SLA instruction on the numerical axis. See Chapter II - 4. 3. Page 17).

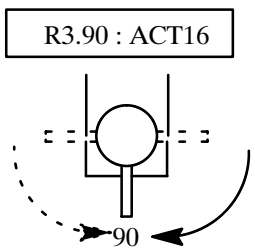
* An additional output is associated with outputs ACT.13, ACT.14 and ACT.16 to control the 90-degree rotation limit stop.

* When ACT.13 or ACT.14 are used, the output allocated to the retractable stop is systematically set at 1.

* When ACT.16 is used, there are 2 possibilities:

. Rotation is in the ACT.13 position; the positioning of the stop and the controlling of the movement will correspond to ACT.14.

. Rotation is in the ACT.14 position; the positioning of the stop and the controlling of the movement will correspond to ACT.13.



front view of the wrist

CAUTION: Do not program the following instructions:

Step n	ACT.13
Step n+1	ACT.16
Step n+2	ACT.14

Rotation after step n+1 is a backup to the intermediate stop in position ”ACT.14”. Requesting ACT.14 in step n+2 will reinforce this backup and obstruct the stop from reversing or if it does reverse, the rotating movement will be completed without any pneumatic speed limit.

Special cases:

ACT00 : This instruction indicates the step where the program stops after an end-of-cycle stop request. Moreover, if the robot parameters are for ”PIP”,the instruction allows the Z arm to lower into the press if the machine cycle has finished.

ACT01 : Instruction not employed.

ACT90 : Inhibits “inter-step watchdog” fault (Valid for the current step).

II - 2. INSTRUCTIONS

II - 2. 1.Variables

Name	Mnemonic	Number	Functions
Output	OUT *	000 to 127	Boolean image of an action to be performed externally (output) (transmitted by the 32S boards).
Input	IN *	000 to 127	Boolean image of an external data item (input) received via the 32EO boards.
Counter	CPT	00 to 15	Structures reserved for increments and decrements

** Caution: the mnemonic code of these structures is also an instruction code.*

II - 2. 1.a)Boolean instructions

* **Temporary output actuation**:

Syntax: > OUT... (000 to 127)

The output is actuated during the program step.Its status becomes 0 when next step is actuated.

Note : In "PIP" applications, the close mould output (parameter 97) is only at "1" if:

- Arm is up or in a free area.
- No part has been made,
- A part has been made + taken from the mould,
- Mode selector is not on "STOP" position,
  or mode selector is not on "ADJUST" position.

* **Status check of an input**:

Proposal: INVERT (otherwise NORMAL)

Syntax: > IN... (000 to 127) or IN/....

Status "1" (or "0") of the input is awaited for passage to next step. Several different inputs can be checked at 0 or 1 status in the same step.

II - 2. 2.Allocation, operation and test instructions

II - 2. 2.a)CNT instruction - Counter handling -

You have at your disposal:

- 16 standard counters,

3 operations can be performed on the counters:

- [RESET]	Counter reset.
- [INC]	Increment; counter(t) = counter(t-1) + 1
- [DEC]	Decrement; counter(t) = counter(t-1) - 1

Examples :

- RST.CNT 0001	Clearing of the counter No. 01.
- DEC.CNT 0015	Counter No. 15 is decremented.
- INC.CNT 0013	Counter No. 13 is incremented.

II - 2. 2.b)TIME instruction- Timer in part program -

The following values can be assigned to the timer:

- a numerical value from 0001 to 999 in 1/10s

Note: This instruction delays running the contents of the step in which it is programmed, or the running of the next step if it is the only instruction in the previous step. In the latter case, outputs programmed in the preceding step are maintained during the programmed time.

Example :

The following routine

Step n	TIME 010 OUT 032 IN 025

is the same as

Step n	TIME 010
Step n+1	OUT 032 IN 025

Instructions OUT 032 and IN 025 will be executed after a delay of 1 sec.

In the following case:

Step n	OUT 032
Step n+1	TIME 20

OUT 32 is maintained for 2 secs. after execution of step n+1.

II - 2. 2.c)<u>*SET instruction*</u>

The following variables can be set:

[OUT] [COUNT]

<u>*Examples*</u>:

Set at 1 : **SET OUT 20**

The output 20 is set at 1.

Affectation of a counter: SET CNT 0010 = 0010-D

The decimal value 10 is set in the counter 0010.

II - 2. 2.d)<u>*RESET instruction*</u>

The following variable can be set:

[OUT]

This instruction has been retained to ensure compatibility with programs written in CN900 softwares. It can easily be replaced by a RESET of the variables.

<u>*Examples*</u>:

Set at 0: **RST OUT 20**

The output 20 is set at 0.

II - 2. 2.e)<u>*IF instruction - Conditions test -*</u>

This instruction evaluates the variable it contains. Depending on the result, the instruction does or does not execute the next instruction (instruction IF must never be used alone).

. **[IF]**: if the condition evaluated is *TRUE*, the next instruction is executed.

. **[IF/]**: if the condition evaluated is *FALSE*, the next instruction is executed (or if the condition is true, the next instruction is not executed).

Note that these instructions will condition execution of any subroutine. Complex conditions can be calculated in the PLC and tested in the main program in IF Bit... or IF/Bit... form, followed by the call instruction of the desired subroutine. (See Programming Manual Level II)

The following variables can be checked:

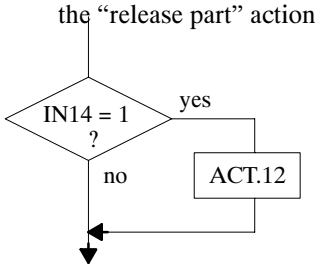
[IN] [OUT] [BIT]

The variables check others than [IN] and [OUT] is described in the Programming Manual Level II.

<u>*Example*</u> :

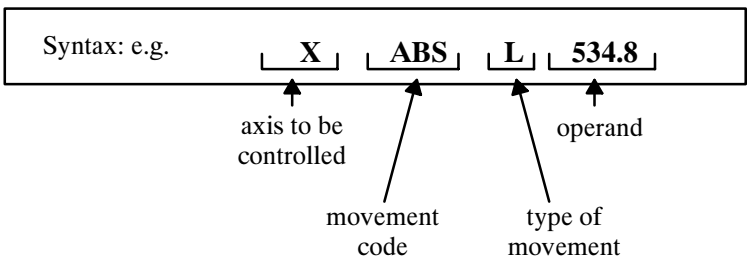
. IF IN 14 If Inpout No. 14 = 1 (True),

ACT 12 the “release part” action will be executed.



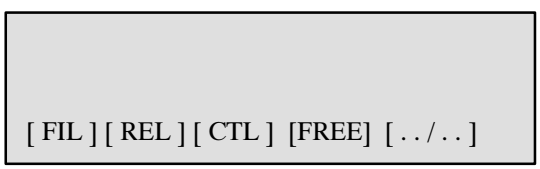
II - 3. MOTORIZED MOVEMENT CODES

These instruction codes are used to control a movement on a given axis.

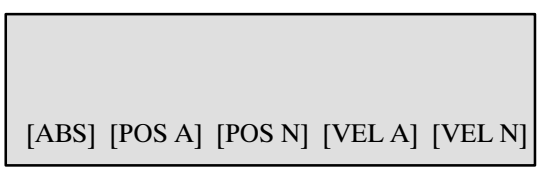


II - 3. 1.Movement Code

Having selected the axis to be controlled (if it is motorized) the following movements are proposed:



. The list is continued by pressing F5

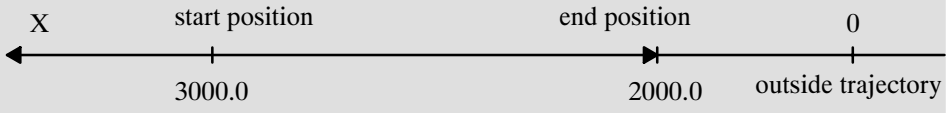


*** ABS: Absolute movement**

Absolute movement is a movement to a given position of the referential. This movement is selected by default. It is therefore possible to enter the operand value (numerical value) directly after selecting the axis. The value will always be positive by defining 0 outside the trajectory.

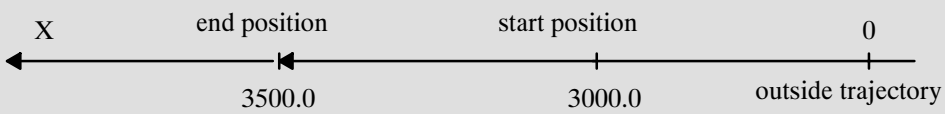
Syntax: > X.ABS L position in 1/10 mm (00000.0 to 99999.9)
> B. ABS R position in 1/10 degrees (000.0 to 360.0)

Example No 1: > X.ABS L 2000.0



Linear absolute movement to position 2000.0.

Example No 2: > X.ABS L 3500.0



Linear absolute movement to position 3500.0.

*** POSA / POSN / VELA / VELN : Slaved movements**

Use of this movement is described in the ”Monitoring an External Axis” Manual.

*** FIL: Stacking movement**

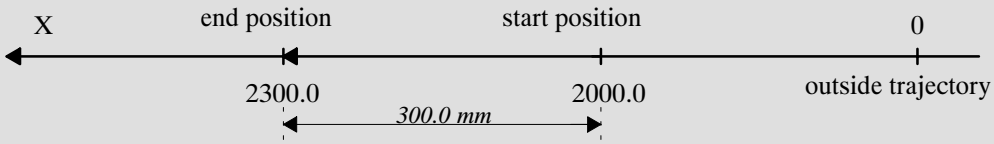
Use of this movement is described in the CN900++ Level II Programming Manual .

*** REL: Relative movement**

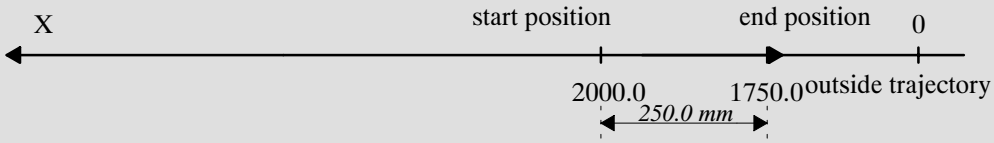
Relative movement is a movement over a given distance.

Syntax: > X.REL L distance in 1/10 mm (± 0000.0 to 9999.9)
> B.REL R distance in 1/10 degrees (± 000.0 to 90.0)

Example No 1: > X.REL L + 300.0
> X.REL - 250.0



Linear relative movement over a distance of 300 mm in the positive direction relative to the start position.



Linear relative movement over a distance of 250 mm in the negative direction relative to the start position.

*** CTL: Control movement**

Use of this movement is described in Chapter II - 4. 5.”Master Movement” Page 19.

*** FREE: Released movement**

This movement is used to release an axis brake without starting up the motor. Counting continues and provides the position of the robot even if this changes as the result of an outside action (ejector, mould, etc.).

This code is only valid if the required axis parameter is supplied correctly. (See Parameter Setting Manual).

The action of this code is continued during the steps following the programming step until the next movement request for this axis is sent.

Caution: Do not release vertical axes unless a balancing device has been installed.

Syntax: > Y.FREE

Example:

Step n	Y FREE
Step n+1	. . . . .
Step n+2	. . . . .
Step n+3	Y ABS . . . .

The Y axis is free during steps n, n+1 and n+2.

II - 3. 2.Type of Movement

L = Linear
R = Rotary

This information appears automatically when a motorized movement is programmed. It corresponds to the type of movement linked to the axis and it is provided in the axis definition parameters described in the Parameter Setting Manual.

II - 3. 3.Operand

After selecting the movement code:

1) The operand is provided immediately after the current axis position. (The axis must be initialized).

Example: X axis is in the position X = 2317.4

. Press [X] then [F5] [. . / . .] then [F1] [ABS]

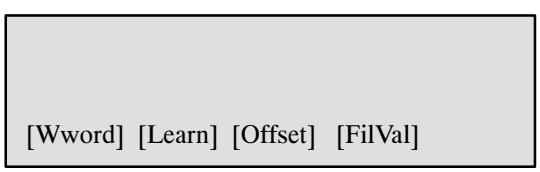
-> the robot displays > X ABS L 2317.4 (operand flashes).

. Press [ENR] .

-> the robot displays > X ABS L 2317.4

2) The operand can be a numerical value (in 1/10 mm or in 1/10 degrees).

3) One of the following functions can be used:



*** wword: 32-bit word**

Use of this operand is described in the CN900++ Programming Manual Level II.

*** Learn: Learning**

The operand will only be provided when the program is run in Step by Step mode.

When the robot carries out a Step containing a programmed movement whose destination was declared in "learning" mode, a message informs the operator that he must move the axis himself using the blue keys X, Y, Z, B or C. The final position of the axis is validated by the operator by pressing [ENR].

Syntax: > X.ABS L Learning
> B.ABS R Learning

Codes and movements for which learning is possible: ABS / FIL / REL / CTL.

*** Offset**

Use of this function is described in the "Monitoring an External Axis" manual.

*** FilVal: Position of the first part in a general stacking sequence**

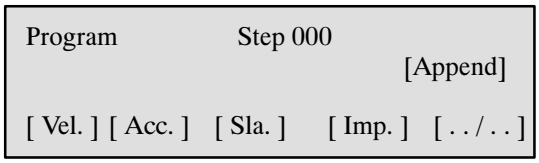
Use of this operand is described in the CN900++ Programming Manual Level II.

II - 4. PREPARATORY "FUN" FUNCTIONS OF NUMERICAL AXES

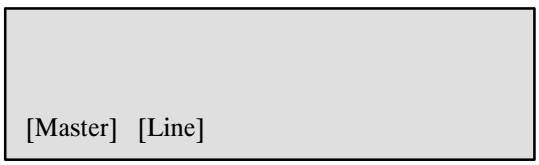
These are preparatory functions relating to numerical movements. There are basically two types:

. Temporary-effect functions (valid only for the current step),

. Maintained-effect functions until the appearance of a new function.



. The list is continued by pressing F5



II - 4. 1.[VEL]: Speed - axis value in % (Maintained)

Enables modification of the programmed axis-movement speed 1 to 100 (as a percentage of the maximum speed).

Note: speed changes can be programmed as release actions in a master movement.

II - 4. 2.[ACC]: Acceleration - axix value in % (Maintained)

Enables modification of the acceleration of the programmed axis-movement (as a percentage of the maximum acceleration - 1 to 100)).

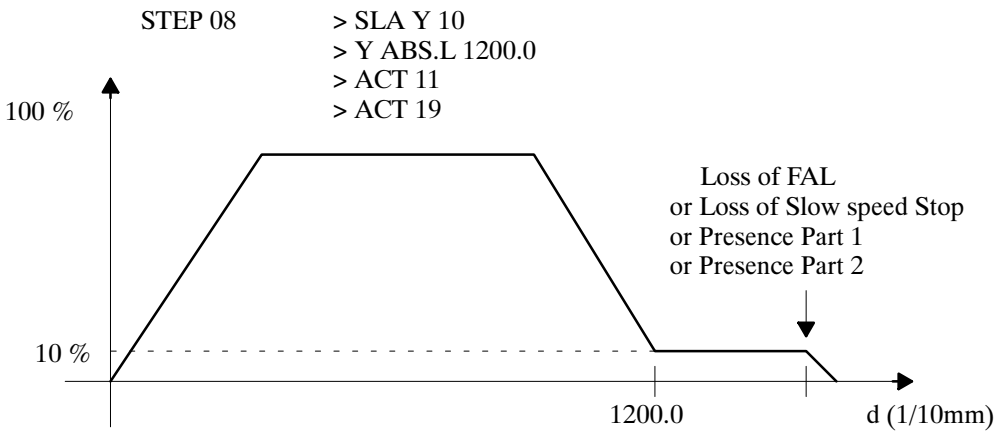
Note: acceleration changes can be programmed as release actions in a master movement.

II - 4. 3.[SLA]: Slow Approach (Temporary)

The following are proposed after the function has been called up: choice of axis, percentage of maximum speed at which the end of the movement must be carried out.

Functioning: this type of function is related to the execution of a numerical movement.

Example :



In the above example, the Y movement will normally be run to reach a speed of 10% of the value 1200,0. Subsequently, the speed will be maintained until:

- "End of Slow Approach" (FAL) input disappears. (Generally on input of the optional changeable parallelogram setting control on SEPRO wrists). Signal active at 0.
- or the disappearance of the "Slow speed stop" input. Signal active at 0.
- or the appearance of a Part Presence control input PP1 if the ACT.11 action code is programmed simultaneously with the slow approach (in Y or Z axes). Signal active at 1.
- or the appearance of a Part Presence PP2 control input if the ACT.19 action code is programmed simultaneously with the slow approach (in Y or Z axes), active signal at 1.
- or the appearance of all the Part Presence controls whose action codes are programmed.

  Example: If ACT21 and ACT23 are programmed, slow speed will be maintained until PP3 and PP4 appear.

Moreover, a so-called "slow" input is associated with each axis which causes a controlled reduction in speed when it disappears, so the movement ends as shown above.

Y and Z axes are usually controlled in relation to part contact inputs and PART PRESENCE if necessary, whereas the other numerical axes can be controlled by distinct inputs.

Possible uses:

Without an external speed-decrease sensor:
. gripping a part whose position is only vaguely known,
. lowering of a part on to a stack whose height is not exactly known.

With an external speed-decrease sensor (direct-reflection cell for example):
. gripping or lowering a part whose axial position is not known but has to be dealt with rapidly.

It will, however, be noted that solutions with the speed decrease sensor are harder to implement.

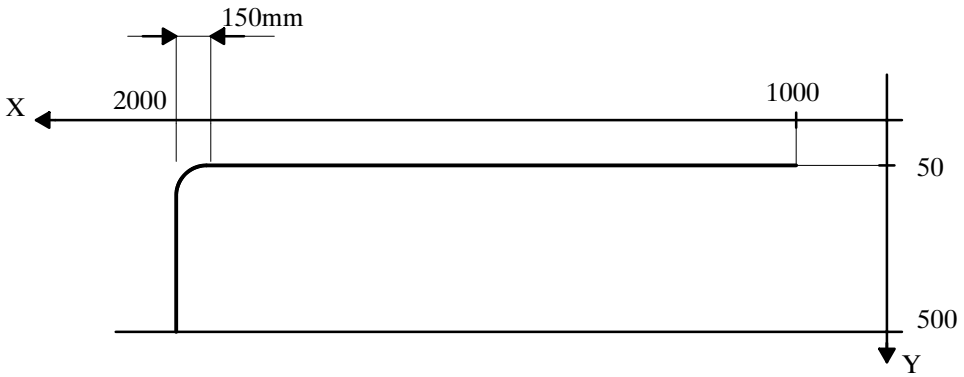
II - 4. 4.[IMP]: Imprecision - Axis value (Temporary)

This function is used to temporarily modify axis stopping tolerance.

The basic use of this type of function is to gain cycle time by moving more quickly form the step with the imprecise movement on to the next step.

The step-to-step transition is masked in the end of the declared unspecific movement.

In the same way, the programmed imprecision value (3.0 to 999.9mm) must not be greater than the programmed movement. If this happens, the movement will not be automatically performed without triggering a fault signal.



Example :

* Step 003

X.ABS.L 2000.0
IMP.X 150.0

* Step 004

Y.ABS.L 0500.0

II - 4. 5.[MASTER] = Master movement (Temporary)

This function enables a program step to be sub-divided into "sub-steps" separated by control points that can be controlled or released by numerical actions or movements.

> MASTER ⌊____⌋ <- master movement axis

> ⌊____⌋ ABS ⌊ operand ⌋ <- master movement destination

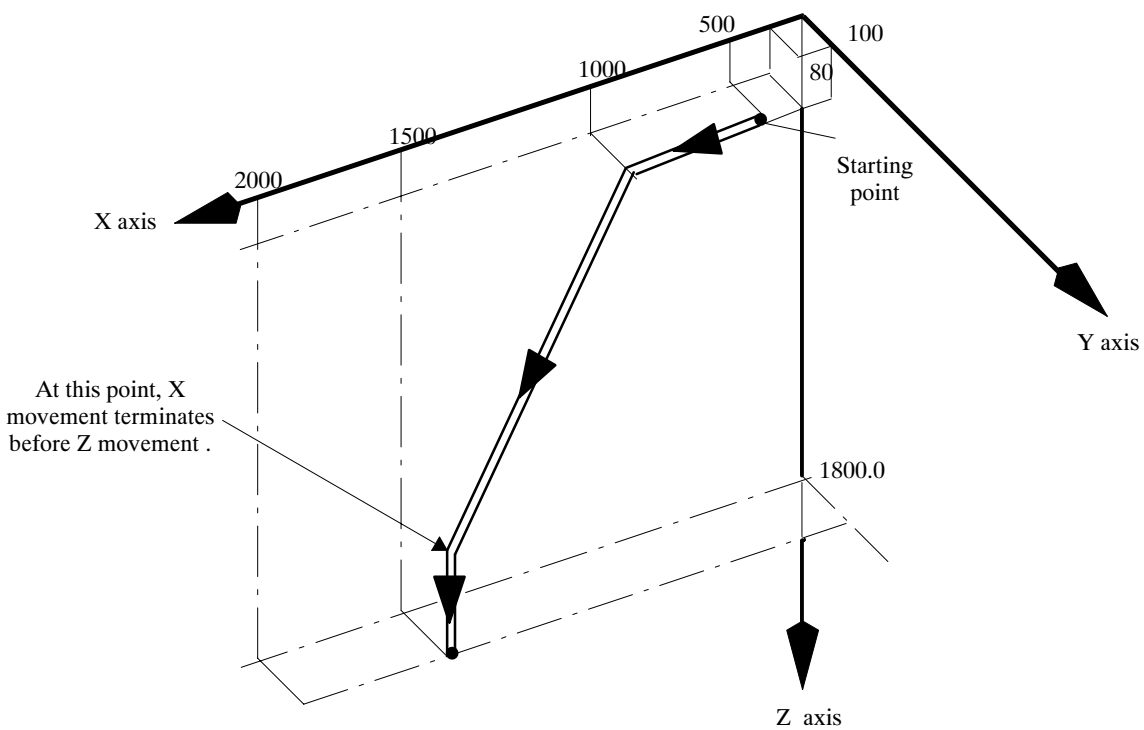
> ⌊____⌋ CTL ⌊ operand ⌋ <- triggering point on master movement

> ⌊_____⌋ <- instruction triggered

Example:

- The positions are as follows: X = 500
  Y = 100
  Z = 80

- An X master movement is to be actuated terminating at point 1500. From the point where X = 1000, the Z movement is to be actuated to move to point 1800.0.



The step will be written as follows:

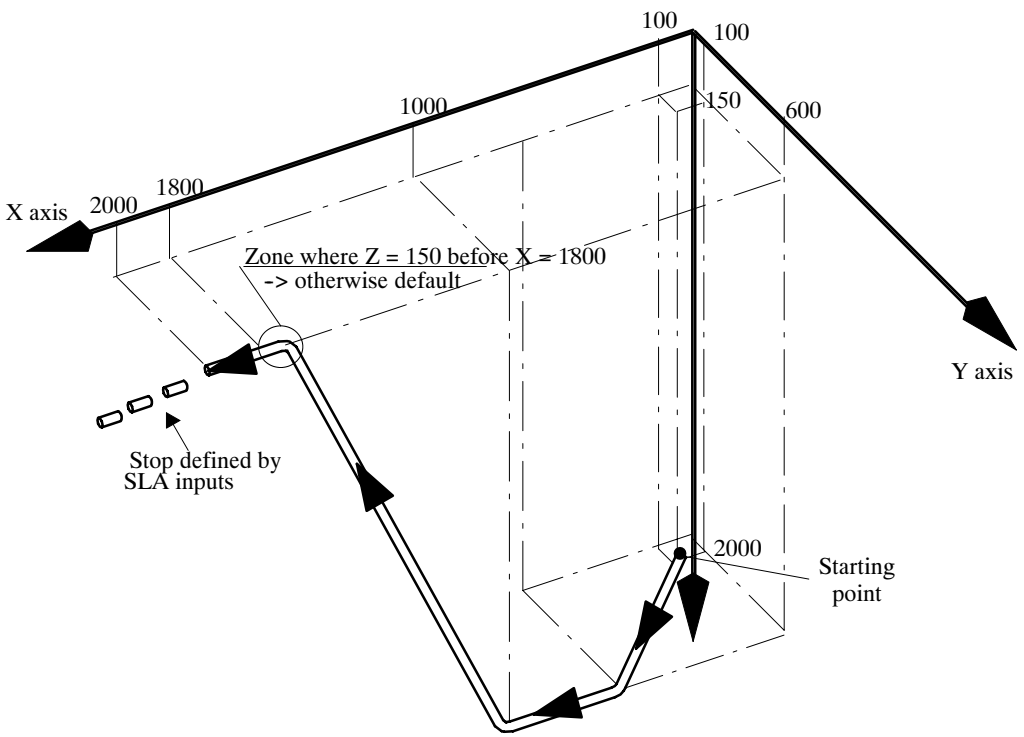
Step n	> MASTER X	* Master word statement
	> X.ABS 1500.0	* Destination of X master movement
	> X.CTL 1000.0	* Triggering point on X
	> Z.ABS 1800.0	* Z triggered movement

Example of a more complex combination

- The positions are as follows: X = 100
  Y = 100
  Z = 2000

- An X master movement is to be actuated ending in a slow approach as of the 2000 mark. From the point where X = 1000, the Z movement should be actuated to move to position Z = 150, ending BEFORE X = 1800.

- In parallel, Y will be actuated AS SOON AS movements start, to go to 600. (Without checking in relation to X).



The step will be written as follows:

```
Step n  > Y.ABS 600.0
        > MASTER X          * Master movement declaration
        > SLA.X 10          * Slow approach declaration
        > X.ABS 2000.0      * Destination of X master movement
        > X.CTL 1000.0      * Triggering point
        > Z.ABS 150.0       * Movement triggered
        > X.CTL 1800.0      * Movement Z control point
```

The only constraint is that movement Y (or other instructions WITHOUT control) must be declared BEFORE X master movement ---> Y.ABS 600.0 - X.ABS 2000.0

If any Y movements appear after X.ABS 2000.0 in the program, or at the time syntax is checked (when going from programming to actuation mode) a fault will be signalled to the operator.

Regarding "preprogrammed" actions: ACT number, defined between two control points (n.CTL...) is first actuated and then checked --->

X.CTL 2000.0
ACT.13
X.CTL 2200.0

To actuate an "ACTnn" during an "n.CTL" operation and to check at a given moment, use the corresponding OUTnn output declaration. The control will only be carried out if it encounters the control input test or the corresponding "ACTnn".

Note:

Control point "sub-steps" have the same effect as program steps in relation to temporarily-actuated outputs or Home Return orders.

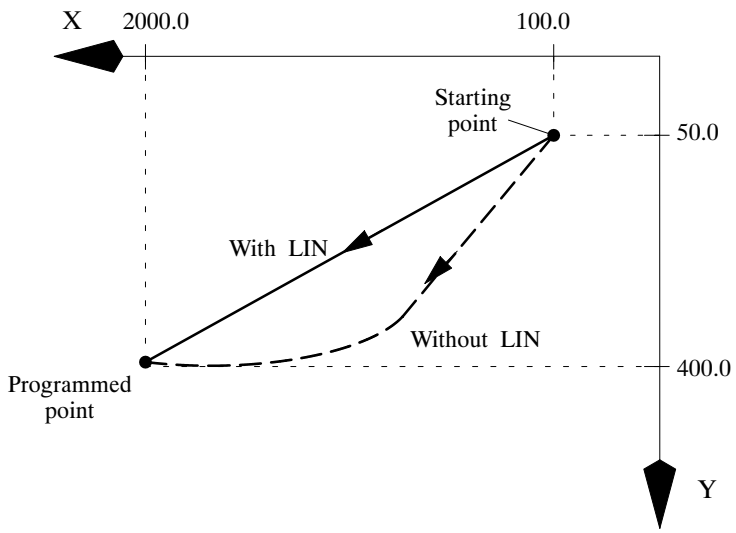
In order to actuate an output during the entire master movement, it should be actuated after each programmed control point.

The same applies to Home Return order numbers.

II - 4. 6.[LINE] = Linearization (Temporary)

This function is used to program a movement linearization instruction into a Step; i.e, all the axes programmed in the Step containing the LIN programming will terminate their movements together.

Example: X and Y movement are to be started simultaneously starting at points X=100, Y=50.0 to go to points X=2000.0, Y =400.0.



. Programming

Step 004 > LIN
> X.ABS 2000.0
> Y.ABS 400.0

II - 5. SPECIFIC CODES

II - 5. 1.SP code as instruction

For its use, see Chapter I - 2.

Syntax: > SP nn L00 (to 99)

After SP request (instruction), the operands requested are:

* Number from 00 to 99

* An Lmm label number, corresponding to a return step indicator and a suffix indicating the order in which the stackings are performed.

Number 00: "Go to" instruction (this is not a subroutine).

Number 01 to 40: Ordinary subroutines.

Note:

* When one subroutine calls another subroutine, only 3 overlapping levels are possible.

II - 5. 2.PLC code as instruction

The PLC (programmable logic controller) is a program that can be used in conjunction with a part program if required. Its functions are described in the Programming Manual Level II.

Syntax: > PLC 00 (to 99)

Note: The number 00 means that there is no associated PLC.

II - 5. 3.SR code as instruction

For the using principle, see Chapter I - 3. page 4

Syntax: > SR 01 (to 99)

Notes:

* SR 00 is not valid, it is implicit (run by default).

* When an SR is executed, it returns to the last Rnn Label with the same number as itself.

e.g.: SR 05 returns to the last R05 encountered before its execution.

II - 5. 4."L" and "R" Labels

Syntax: > L01 to L99

Syntax: > R00 to R99

Notes:

* Label L00 is invalid.
* "L" labels are unique; "R" labels can be multiple.
* They can ONLY be used in the MAIN part program, (excluding SP, SR, etc.) (See Chapter I - 2.).

II - 5. 5.END of MP, SP..., SR, PLC codes

Syntax: > END

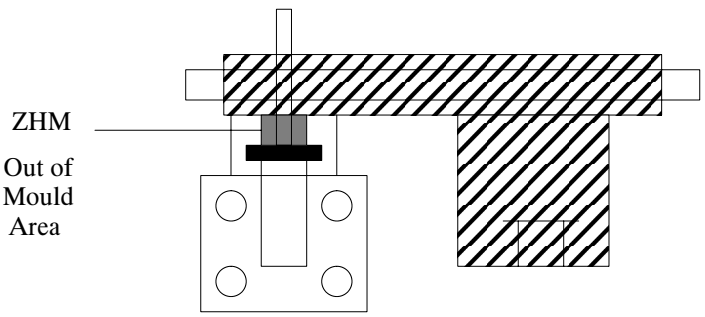
II - 6. PROGRAMMING ANTICIPATED RESTARTS

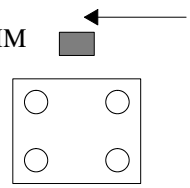
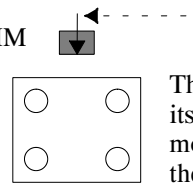
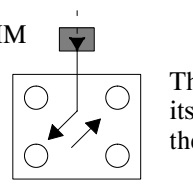
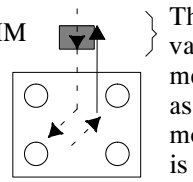
II - 6. 1.Using the ZHM cam

Principle :

This cam enables movement between the high arm position (BH) and the wait position nearest to the mould. Gain is possible in rising and lowering modes.

Example of application :



Step n	X_ABS_L 1000.0 ACT 10 SET OUT 28	ZHM 
Step n + 1	Z_ABS_L 180.0 ACT 00	ZHM  The robot positions itself in the out-of-mould area or close to the mould
Step n + 2 - - - Step m		ZHM  The robot carries out its grip part cycle in the mould.
Step m + 1	Z_ABS_L 100.0 SET OUT 28	ZHM  The machine cycle validation is programmed at the same time as the arm rise. The mould close command is sent as soon as the out-of-mould area is reached*.

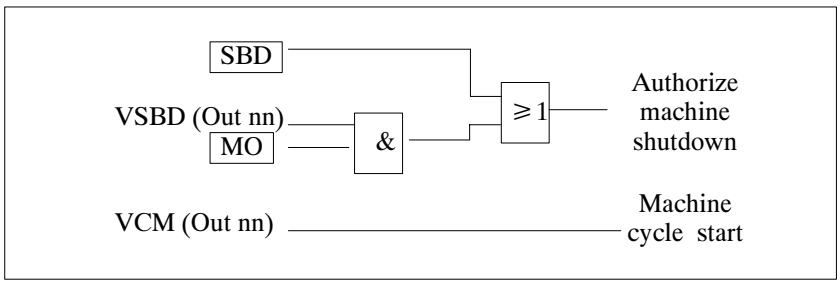
* except when using the authorization for anticipated machine shutdown
(see chapter II - 6. 2. page 26).

The cam defining the Out-of-Mould Area (ZHM) must be reset at each mould change in order to take into account the gripper + part space needed.

II - 6. 2.Authorization for anticipated machine shutdown

Principle : enables anticipation of machine reaction time.

Validation of mould closing (SBD) is given to the press during robot arm rising, before reaching the ZHM safety cam. The loss of the signal Machine Open inhibits the signal "authorization machine shutdown" and thus prevents machine shutdown. The signal "part made" (PFAB) resets the validations to zero.



- Electrical adaptation must be planned.

- Some parameters must be updated:

Parameter 102 : "Validation arm free safety"

-> contains the number of the VSBD output.

Parameter 157 : "Anticipation machine control"

-> 2 possibilities : . invalidated
. programmable delayed activation.

Parameter 156 : "Type of anticipated restart"

-> contains the duration of activation delay for the VCM and VSBD outputs.

- The program must be adapted to enable anticipated restart:

Example of application :

PP xx

Step 00	PLC 00 SET WW 063 = xxxx	Value of actual activation delay for the VCM and VSBD outputs . This value cannot be less than half the value of parameter 157. If this command is not programmed, the value of parameter 157 will be used.
Step n	ACT 11	Grip part
Step n +1	Y_ABS_L 60.0 OUT 29	
Step n + 2	Z_ABS_L 100.0 SET OUT 28 SET OUT nn	VCM VSBD : output parametered to parameter 102
Step n +3	. . . . .	

-> The timer of the parameter 157 and/or the WW063 is activated at the start of the step containing the parameter 102 output. When the delay value is reached, the VCM and VSBD validations are given to the machine.

Note :

When using the command SET WWORD 63 = xxx, an immediate modification of the activation delay value can be carried out by pressing key [T] of the control unit.

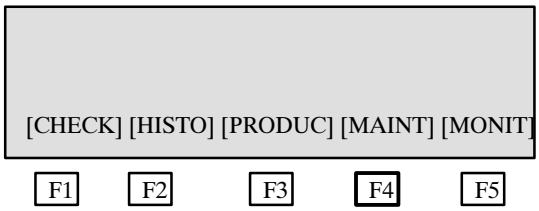
The new value can be typed in on the numerical keyboard.

This function is accessible only on VEL = 100 % and in automatic operating mode.

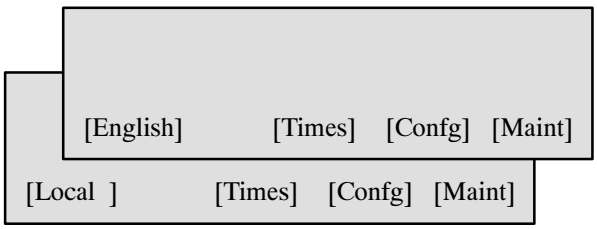
III - PROGRAMMING

LANGUAGE CHOICE:

The language used (local language or English) can be selected using the function [MAINT] [F4] available in the main menu.



The system then proposes:



By pressing [F1] [ENGLISH] English will be selected.

By pressing [F1] [LOCAL] the language of the country where the robot is delivered can be selected.

PART PROGRAM CREATION:

On CN900++ robots, an executable program can be created in various ways:

In standard :

Those with a thorough knowledge of programming Sepro robots can write the program directly into the robot using Edition mode.

SAP Option :

Those with no knowledge of programming can use the programming help system (SAP). However, predefined "Source" programs corresponding to the required cycles must be present on the machine.

PC Editor Option:

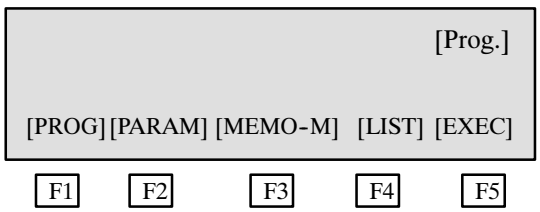
Programs can also be written using the program editor on PC or compatible. This requires a knowledge of programming but has write facilities that remind the operator of the various codes that can be used, together with their mnemonics and operands. A "help" function is constantly available.

III - 1. DIRECT PROGRAMMING ON THE ROBOT

The “PROG” mode is called by briefly positioning the programming key (on the offset box) in position 1.

Note: The mode selector must be on the ”STOP” or ”ADJUST” position before the call. If not, the ”MONITOR” function will be accessed. In this case, exit this function by pressing [EXIT] key, put the mode selector on ”STOP” position and turn the key again .

Entry into this mode is signalled by display of [Prog.] in the upper right-hand corner, and a menu of five possibilities selectable using the function keys (F1 to F5).



F1 -> [PROG] : Accesses a higher level at which a program can be written directly (MP or PLC). Also accesses functions such as Save, Restore, Delete, etc. (*Note*: If the SAP module (see SAP Manual) is present, the system requests a password before allowing access to edition).

F2 -> [PARAM] : To read and/or modify ”machine” and ”system” configuration parameters.

F3 -> [MEMO_M] Calls memory manager (user and system).

F4 -> [LIST] Used to list or print the various programs in the robot memories.

* Select the type of routine to be listed. (MP, PLC).

* The function looks for the routine type in the memory. The type of memory and routine are then displayed in the upper right-hard corner of the screen.

* When a routine is found, its number and size (Kbytes) are displayed.

* The keys F5 [->] or [P+1] or [↑] allows the function to look for the next routine.

* The [Print] key selects the displayed routine for printing. (10 successives routines may be selected).

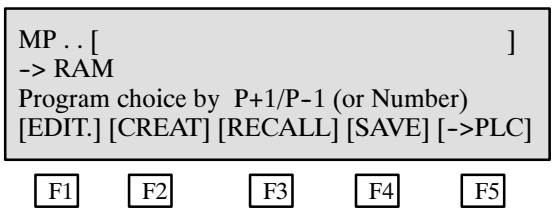
Example: [Print], F5, [->], [Print], F5, [->] [Print] ...

* If the program to be printed contains SAP ”point markers” and if the SAP module is present, the system asks if the operator wishes to print the list of points contained in the selected program. If the reply is NO, a normal listing will be printed.

If there are programs in the EEPROM memory, the number of bytes which are still available will be displayed once they have been listed.

F5 -> [EXEC] Used to execute an existing program or to use one program to create a new program.

After pressing F1, key, the display is:



F1 -> [EDIT] Calls the editor to read and/or write user program selected.

F2 -> [CREAT] Calls the editor to write a new user program.

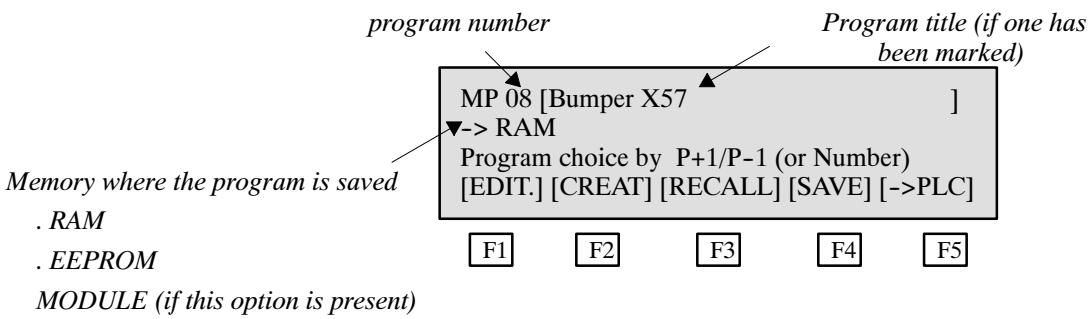
F3 -> [RECALL] Recalls program selected in EEPROM to RAM.

F4 -> [SAVE] Saves program being read and/or written and/or modified in RAM to EEPROM.

F5 -> [-> PLC] Selects program type to be read and /or modified or written : PLC or MP.

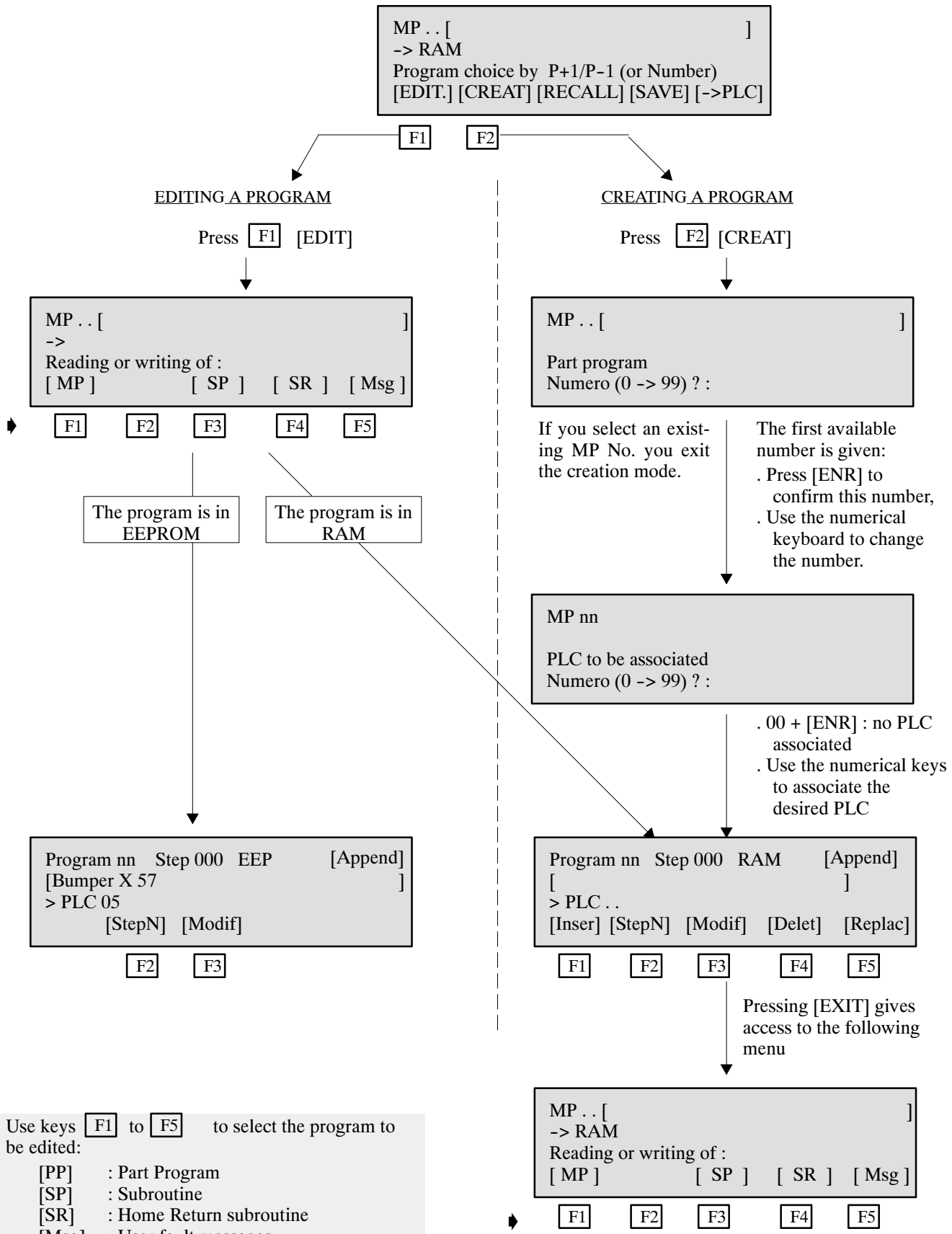
III - 1. 1.Choice of Program to Edit

The choice is made by pressing P+1/P-1 in order to list the existing programs or by directly entering the program number to be selected.



III - 1. 2.Editing or Creating a Program

III - 1. 2.a)Access Procedure



III - 1. 2.b)Editing commands:

All commands described below are performed on or in relation to the first instruction displayed in line 2 (Edit line).

Line 2

```
Program          Step 000        [Append]
> PLC 00
> X.ABS.L 01000.5
[Inser] [StepN]  [Modif]  [Delet]  [Replac]
  F1      F2       F3       F4       F5
```

Instructions given outside of these controls are ADDED AFTER this first instruction (Implicit mode).

F1 -> [Inser] Used to insert an instruction or a Step PRIOR to the displayed instruction. (It is impossible to insert an instruction or a step before the PLCnn instruction or the associated message).

F2 -> [Step-N] Used to obtain direct access to step n (0 to 999) of the program.

F3 -> [Modif] Used for simple modification of the operand NUMERICAL value or to access to the modification of MP, SP, SR or PLC associated message.

Note: This function cannot be used to replace the numerical operand by the "learning" option, or to correct modifications of instructions such as: SP 41 N L00 (multiple operands).

In this case, use the replace function and retype the desired instruction.

F4 -> [Delet] Deletes the displayed instruction. If it is the last one in the step, it is removed and the whole steps program is renumbered.

(PLC nn and END instructions cannot be removed !)

F5 -> [Replac] Used to completely replace the displayed instruction by any other (Except PLC nn, END and message).

[P+1] To go to the following step .

[P-1] To return to the previous step.

[↑] Used to return to the previous instruction contained in the editing step.

[↓] Used to read the contents of the step being edited, instruction by instruction.

Notes: The EXIT key is used to exit at any time from:

. instruction in progress (before validation)

. the current editing function,

. the "read/write" procedure if the program is ended with the word END.

. writing a new program in which nothing is yet written. If this is the case, the display is:

```
MP . . [                                    ]

Leave program created ?
[YES]  [NO]
  F1     F2
```

III - 1. 2.c)Insertion : (F1)

- Using the [↑] and [↓] keys,place on the edit line the instruction that will follow the one to be inserted.
- Press F1 [Inser].
- Choose the operation to be executed:

  . Insertion of an instruction (or set of instructions).

  . Insertion of a step.

Program	Step 125	[Insert]
> ACT 12		
> ACT 13		
[Insert] [StepN]		

* INSERTING AN INSTRUCTION :

The edit line is moved downwards and space is created for the new instruction. After writing and validating of the new instruction, the editor returns to ADD mode for the following instructions.

Program	Step 125	[Insert]
> ACT 12		
Instruction ?		

EXAMPLE :

Inserting instruction SLA.X.010 before X movement concerned.

- Initial display

> X.ABS.L 02000.0
> OUT.012

- Instruction insertion mode

>
> X.ABS.L 02000.0

- Entry and validation of instruction

> SLA.X 010
> X.ABS.L 02000.0

- If further instructions are written, they will ADDED after SLA.X 010.

e.g.: IN.075

> SLA.X 010
> IN.075

* INSERTING A STEP:

The two editing lines are removed and a new step is created BEFORE the BEGINNING of the one which was displayed.

e.g.: step 3 contains	> OUT.01
	> OUT.02
	> OUT.03
the display shows	> OUT.02
	> OUT.03
or	> OUT.03

The new step will be created BEFORE the instruction OUT. 01, no matter what the display.

Notes: It is not possible to insert:

. an instruction BEFORE the PLC nn or the associated message of the program,

. a step before step 0.

III - 1. 2.d)Jump to step N: (F2)

After selecting this command, type the number of the step to be edited and then validate.

. If the step exists, its content is displayed,

. If not, the following error message is displayed:

"This step does not exist !"

III - 1. 2.e)Modifications: (F3)

* MODIFICATION OF THE PROGRAM ASSOCIATED NAME

- As for the operands, transfer the message to the editing line.
- Press F3 [Modif].
- A flashing cursor is displayed: use the programming keys to enter the letters (using the letters shown on the light background of the labels), and the numerical keys for the digits. The [+] key acts as the space bar and apostrophes are obtained using the [END] key. To rectify errors or modify messages already written, use the [<-] (F4) and [->] (F5) keys to position the curser on the character to be modified.
- Validate the end of writing by ENT.

III - 1. 2.f)Deleting: (F4)

- Use the [↑] and [↓] keys place the instruction to be deleted on the editing line.
- Press F4 [Delet].
- Confirm by pressing ENT.

Notes:

. If the instruction deleted was the only one in the step, this is also deleted and the entire program is re-numbered.

. It is not possible to delete the instruction PLC nn nor the word END.

III - 1. 2.g)Replacing: (F5)

- Use the [↑] and [↓] keys place the instruction to be replaced on the editing line.
- Press F5 [Replac].

. The instruction is deleted, leaving space for the new one. Write this in and then validate it.

Notes: It is not possible to replace:

. Instruction PLC nn,

. the word END.

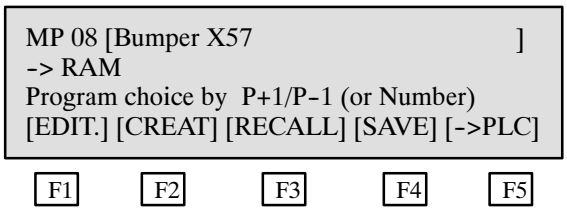
Note: Do not forget that for all editing commands, the EXIT key can be used to suppress the current command prior to validation.

Example: Request for insertion of a step:

-> the empty step is created.

-> EXIT -> return to previous status.

III - 1. 3.Recall and Saving Procedures



F3 -> [RECALL] Calls up user programs (MP, PLC) recall function from RAM.

F4 -> [SAVE] Calls up saving function for user programs (MP, PLC) from EEPROM (internal or module).

Note: When save or recall functions are called up and the EEPROM module is installed on the DIAL board, you will be asked if the operation to be carried out concerns the module.

III - 1. 3.a)SAVE function

* Select the type of program to be saved

MP -> main program

PLC -> programmable logic controller

* The procedure checks that the program to be saved exists in the RAM memory of the robot, if not, the message *"The requested program does not exist"* is displayed.

* You are asked:

"Program to be saved with the same number ?"

[YES] *[NO]*

NO, the system will ask for the new number.

* If YES, you are then asked:

"Program in RAM to be deleted after saving ?"

[YES] *[NO]*

* The system checks that there is not a program already in EEPROM (internal or module) with the same number. If this is the case, a warning is displayed with a request for confirmation of "overwriting" by the new program:

"Program already in memory"

"Continue"

[YES] *[NO]*

* If write in EEPROM is performed correctly, a series of dots appear on the third line (or a counter followed by a series of dots), if not the message *"EEPROM writing error"* or *"Program in memory incoherent"* is displayed.

If this occurs, refer to the "MEMORY MANAGEMENT" section for possible correction of memory.

III - 1. 3.b)<u>*RECALL function*</u>

* Select the type of program to be recalled from RAM (MP, PLC).

* Give the number of the program requested or use the key [?] to access the exiting programs list.

* The procedure checks:

  - that the program requested exists in EEPROM (internal or module depending on selection) if not the message *"The program does not exist!"* is displayed.

  - that there is not a program of the same type already in the RAM area concerned. If this is the case, a warning is displayed with a request for confirmation of "overwriting" by the new program.

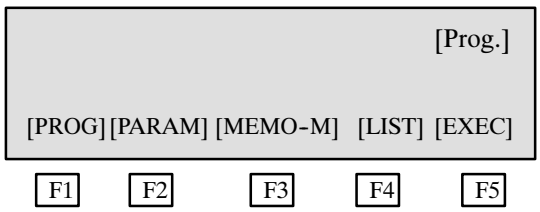
    "Program already in memory!"

    "To be saved ?"

    [YES] *[NO]*

* A series of dots is displayed on the third line, confirming that the command has been carried out.

III - 1. 4."Memory management" procedures



When the F3 [MEMO-M] key is pressed, the following choices are proposed:

F1 -> [Val+0] Procedure for reconfiguring of system and user variables.

F2 -> [Val+EFT] Procedure for reconfiguring of system and user variables.

F3 -> [Syst] Calls up procedure for primary modifications to memory.

F4 -> [Delet] Deletes part or all of the program.

F5 -> [MRead] Used to display and modify user-memory areas (and part of system area).

Note: A password and/or confirmation to continue may be requested for certain functions modifying the system or critical areas of memory (these are reserved for Sepro technicians, since incorrect use can lead to operating faults).

III - 1. 4.a)[Val+0] and [Val+EFT] functions

Confirmation for continuation is required for both functions.

III - 1. 4.a)a)[Val+0] function [F1]

- clearing and reconfiguration of system and user variables concerning the immediate environment for running a program.
- the RAM or EEPROM memory containing the programs is not affected.

Note: In this procedure the storage counter area is not affected. The program running or to be run will start or restart from the beginning. If the operator does not want to affect the counters, a simple home return is performed; otherwise carry out a TOTAL Home Return (recommended Home Return).

III - 1. 4.a)b)[Val+EFT] function [F2]

Same as preceding function, plus:

- resetting of all storage counters,
- resetting of user programs in RAM memory,
- resetting of various basic variables: fault stack / bit area / word area / etc...

III - 1. 4.b)SYSTEM procedure [F3]

Access reserved for Sepro technicians.

III - 1. 4.c)DELETE function [F4]

* Select the memory concerned RAM or EEPROM (internal) or Module (if the module is not installed pressing the module key is ignored).

* Select the type of program to be deleted. (MP, PLC, SP or SR).

* For the programs in EEPROM (internal or module), give the number of the program to be deleted.

* The procedure checks that the program exists, if not a message is displayed:

"This program does not exist!"

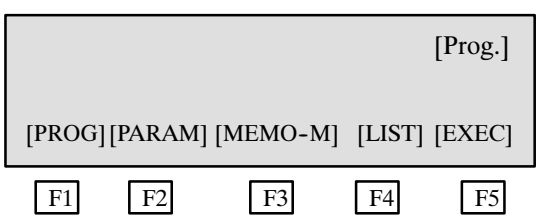
* A series of dots is displayed on the third line confirms that the command has been carried out.

Note: SPs and SRs cannot be deleted directly from EEPROM. The SR number 0 cannot be deleted by this function.

III - 1. 4.d)READ MEMORY function [F5]

Access reserved for Sepro technicians.

III - 1. 5.Parameter setting



The [Param] key gives access to the parametering mode via the following menu:

F5 -> [Print] : To list all parameters (machine and user) on printer (if option installed)

F1 -> [Mach.] : Accesses machine parameters after password has been entered. Access must be strictly reserved for authorized personnel.

F2 -> [User] : Allows reading of all parameters and writing of user parameters (No. 4 to 11) only.

(For further information, see "Parameters" Manual).

IV - BIBLIOGRAPHY

1 - Simple Pick and Place Applications

- CN900++ Operating Manual
- Using of the CNC 900++ printer link
- CN900++ Hardware architecture
- CN900++ Programming Level I

2 - Supplement for advanced applications (Stacking, peripheral units control, dialogue with other machines) (to be requested if necessary)

- CPU Board
- DIAL Board
- 32EO Board
- 32S Board
- 3 AXES Board
- Secu 94 Board
- 4 Brakes Board
- Z89 Brake Board
- CN900++ Programming Level II
- CN900++ Parameters
- CN900++ encoding and adressing of instructions
- Monitoring an external axis

3 - SAP Option

- Using the SAP assisted programming

4 - PC Option

- NEWAS900

We're Here to Help

Conair has made the largest investment in customer support in the plastics industry. Our service experts are available to help with any problem you might have installing and operating your equipment. Your Conair sales representative also can help analyze the nature of your problem, assuring that it did not result from misapplication or improper use.

How to Contact Customer Service

To contact Customer Service personnel, call:

From outside the United States, call: 814-437-6861

You can commission Conair service personnel to provide on-site service by contacting the Customer Service Department. Standard rates include an on-site hourly rate, with a one-day minimum plus expenses.

Before You Call ...

If you do have a problem, please complete the following checklist before calling Conair:

- ❒ Make sure you have all model, serial and parts list numbers for your particular equipment. Service personnel will need this information to assist you.
- ❒ Make sure power is supplied to the equipment.
- ❒ Make sure that all connectors and wires within and between loading control and related components have been installed correctly.
- ❒ Check the troubleshooting guide of this manual for a solution.
- ❒ Thoroughly examine the instruction manual(s) for associated equipment, especially controls. Each manual may have its own troubleshooting guide to help you.
- ❒ Check that the equipment has been operated as described in this manual.
- ❒ Check accompanying schematic drawings for information on special considerations.

Additional manuals and prints for your Conair equipment may be ordered through the Customer Service or Parts Departments for a nominal fee.

EQUIPMENT GUARANTEE

Conair guarantees the machinery and equipment on this order, for a period as defined in the quotation from date of shipment, against defects in material and workmanship under the normal use and service for which it was recommended (except for parts that are typically replaced after normal usage, such as filters, liner plates, etc.). Conair's guarantee is limited to replacing, at our option, the part or parts determined by us to be defective after examination. The customer assumes the cost of transportation of the part or parts to and from the factory.

PERFORMANCE WARRANTY

Conair warrants that this equipment will perform at or above the ratings stated in specific quotations covering the equipment or as detailed in engineering specifications, provided the equipment is applied, installed, operated and maintained in the recommended manner as outlined in our quotation or specifications.

Should performance not meet warranted levels, Conair at its discretion will exercise one of the following options:

- Inspect the equipment and perform alterations or adjustments to satisfy performance claims. (Charges for such inspections and corrections will be waived unless failure to meet warranty is due to misapplication, improper installation, poor maintenance practices or improper operation.)
- Replace the original equipment with other Conair equipment that will meet original performance claims at no extra cost to the customer.
- Refund the invoiced cost to the customer. Credit is subject to prior notice by the customer at which time a Return Goods Authorization Number (RGA) will be issued by Conair's Service Department. Returned equipment must be well crated and in proper operating condition, including all parts. Returns must be prepaid.

Purchaser must notify Conair in writing of any claim and provide a customer receipt and other evidence that a claim is being made.

WARRANTY LIMITATIONS

Except for the Equipment Guarantee and Performance Warranty stated above, Conair disclaims all other warranties with respect to the equipment, express or implied, arising by operation of law, course of dealing, usage of trade or otherwise, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.