www.conairnet.com

USERGUIDE

Programming

MPA-II v2 XN, XS or YN,YS Robots

Software version 1.0

⚠ **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This manual is out-of-date and is provided only for its technical information, data and capacities.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containing warnings, information, precautions, and procedures that may be more adequate than those contained in this out-of-date manual.

Logo definitions :

⚠	Warning, risk
🖉	Sepro robotique inventions
?	What to do ?
📜	Document evolutions
💡	Handy hints
🔍	Example

~> Key pulses.
Example : press [WRITE 1 DIRECT] ~> [TEACH 2 APPRCH]
means : press [WRITE 1 DIRECT] then [TEACH 2 APPRCH]

☞ Vxx Innovation or information concerning a particular software version

– CONTENTS –

This document is for users of the MPA–II v2 robots :

▶ with a numeric axis on X (XN, XS),

▶ with two numeric axes on X and Y (YN, YS).

In both cases, the Z vertical movement is always pneumatic.

You are advised to read at least the first two chapters of the MPA–II User Manual.

I – STUDYING AN APPLICATION EXAMPLE

This chapter describes an unloading application from an injection moulding machine (IMM). The example starts with the need analysis and goes as far as entering the program on the MPA–II terminal (chapter II). This example can be used as a basis for all new users of the Sepro MPA–II control unit who wish to create programs.

I – 1. Description of the robot cycle

The cycle described in the example is an IMM unloading application with a single part release on a conveyor belt.

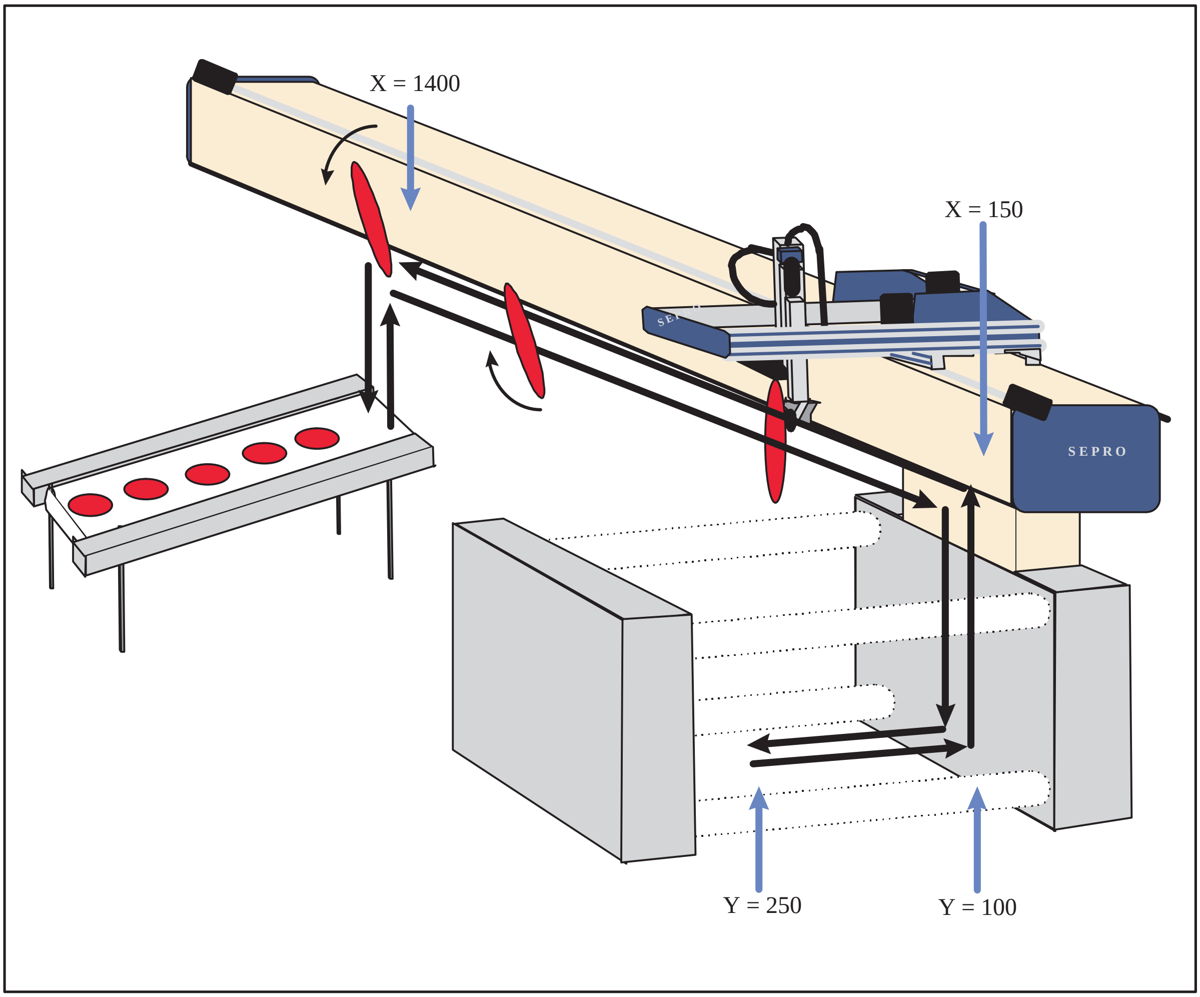


Figure 1 : Cycle movements

THE CYCLE	
The main sequences	**The actions and movements**
Disengaging sequence	– Release part. – Y back. – Z up.
IMM cycle start and await opening	– X positioned above the IMM, gripper head vertical, IMM cycle started and ejectors back authorised.
Part grip in the mould sequence	– Z descent into the mould. – Y forward towards the part. – Ejectors forward and await ejectors completely forward. – Grip part. – Y back for part demoulding, ejectors back and await ejectors completely back. – Z up to exit mould.
IMM cycle restart	– Restart IMM cycle.
Part release on belt sequence	– X positioned above the belt. – Gripper head horizontal. – Z down. – Release part. – Z up again.
Belt indexing	– Belt indexed one step for 5 seconds.

▶ Starting conditions

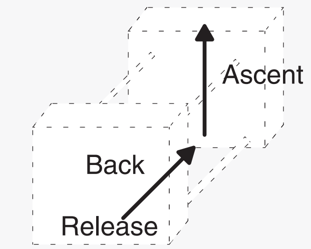
The positions needed to start up the robot cycle are programmed in Step 1. If this is not necessary, it is better not to write anything in Step 1 and start the program in Step 2.

In our example, you must consider that it is not necessary to define a position to start the robot cycle. Do not write anything in Step 1 and go on to Step 2.

▶ The disengaging sequence

This is used to free the robot after it has been stopped to position it above the machine. To ensure that the mould is not damaged, it is preferable to imagine that the robot could be in the IMM. The disengaging sequence is therefore as follows :

Disengaging sequence



Each action or movement will be programmed in a Step.

▶ Positioning the robot above the IMM and starting the IMM cycle :

This sequence is to prepare the robot for its descent into the mould. You must therefore position it in the machine axis and orientate the gripper head correctly.

You must also send a part to be produced. For this, the robot must validate the IMM cycle.

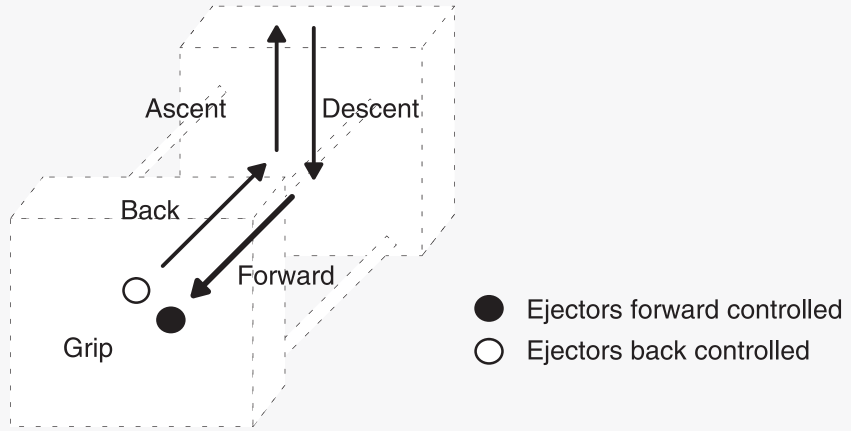
▶ The part grip sequence :

The part grip sequence can vary from one application to another depending on the type of part to be unloaded. The questions to be considered are, for example :

- Is the part held at the end of the ejectors when they are fully forward ?
- Is the part freed before the end of ejection ?
- Is it possible to put the ejectors forward before the Y axis has completed its forward movement ?
- Is it necessary to wait for the ejectors to be completely back before raising arm Z ?
- Etc...

For our example, the answer to these 4 questions is YES for the cycle which follows :

Part grip sequence



Restarting the machine cycle :

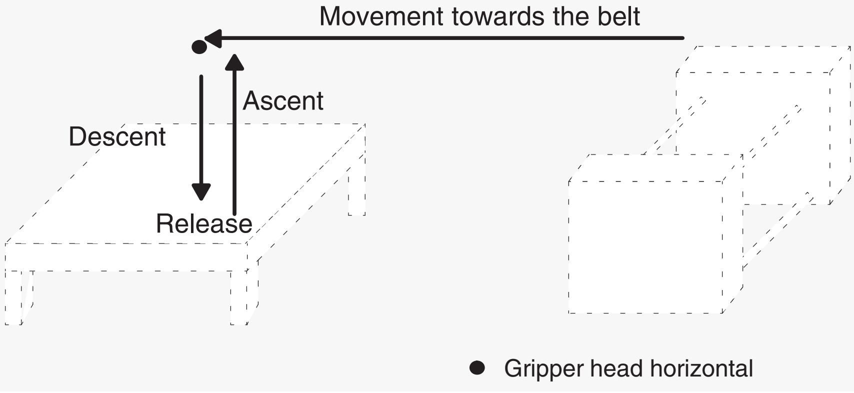
Now that the part grip sequence has been completed, we must program the restart of the machine cycle so that a new part can be produced.

Releasing the part :

Final sequence to program : releasing the part on the belt.

The release cycle in our example is as follows :

Release sequence

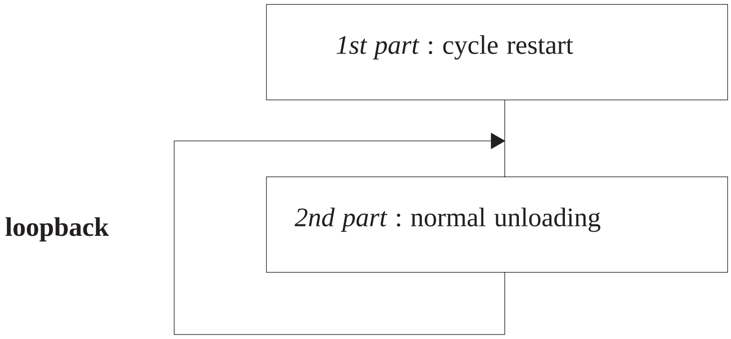


Belt indexing, the program end and loopback :

The belt indexing is controlled by an auxiliary output whose activation time is entered in the program options.
See chapter IV – 5. page 23 for the description of all the program options.

The program end is programmed by pressing ! END in the step containing the last instruction.

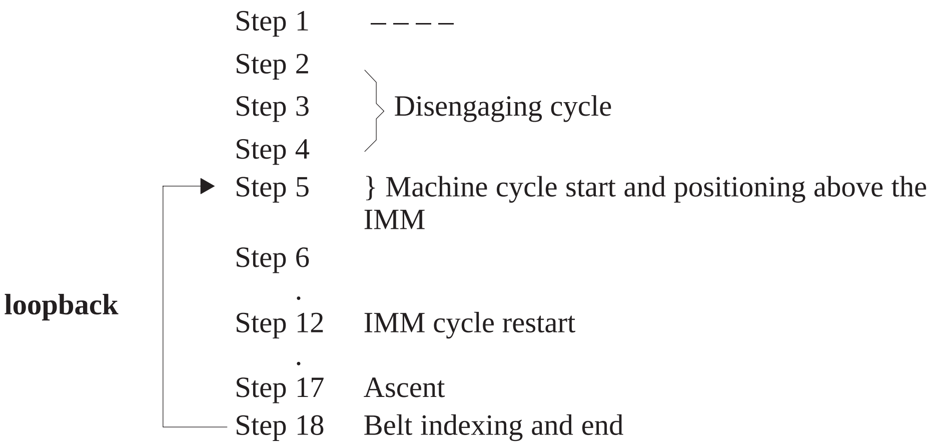
The following question appears when the last step of the program is validated : which step should you loopback to when the program is finished ?
You do not have to loopback to Step 1 ; the aim is to separate the program into two parts. The first part is only carried out when a cycle is restarted and the second is the normal unloading cycle. You will loopback to the beginning of this 2nd part.



When you exit Auto, Step by Step or Stop mode or following a part grip fault, the cycle starts again at the beginning of the first part and therefore carries out the disengaging sequences and the machine cycle start.

If we look at our program more closely, we can see that steps 1, 2, 3 and 4 are only carried out in the 1st cycle.

The program can therefore be instructed to loopback to Step 5.


Step 1 – – – –
Step 2
Step 3 Disengaging cycle
Step 4
Step 5 } Machine cycle start and positioning above the IMM
Step 6
.
Step 12 IMM cycle restart
.
Step 17 Ascent
Step 18 Belt indexing and end
loopback

Looping back to Step No. 5 triggers the movements that are programmed in this step apart from the IMM cycle which has already been programmed in Step No. 12. As long as the part has not been taken, the IMM cycle command is ignored.

II – PROGRAMMING A CYCLE

II – 1. Selecting a program

The internal memory of a SEPRO MPA–II robot can contain up to 15 program numbered from 0 to 14.

See chapter VI – 2. page 44 “Memory card”, for the program numbers contained in the external memory.

When you create a program, you can :

- ▶ either, create a new program,
- ▶ or, rewrite a program that already exists.

The list of programs that already exist can be consulted in COPY mode (see chapter VI – page 42 “Memory management”).

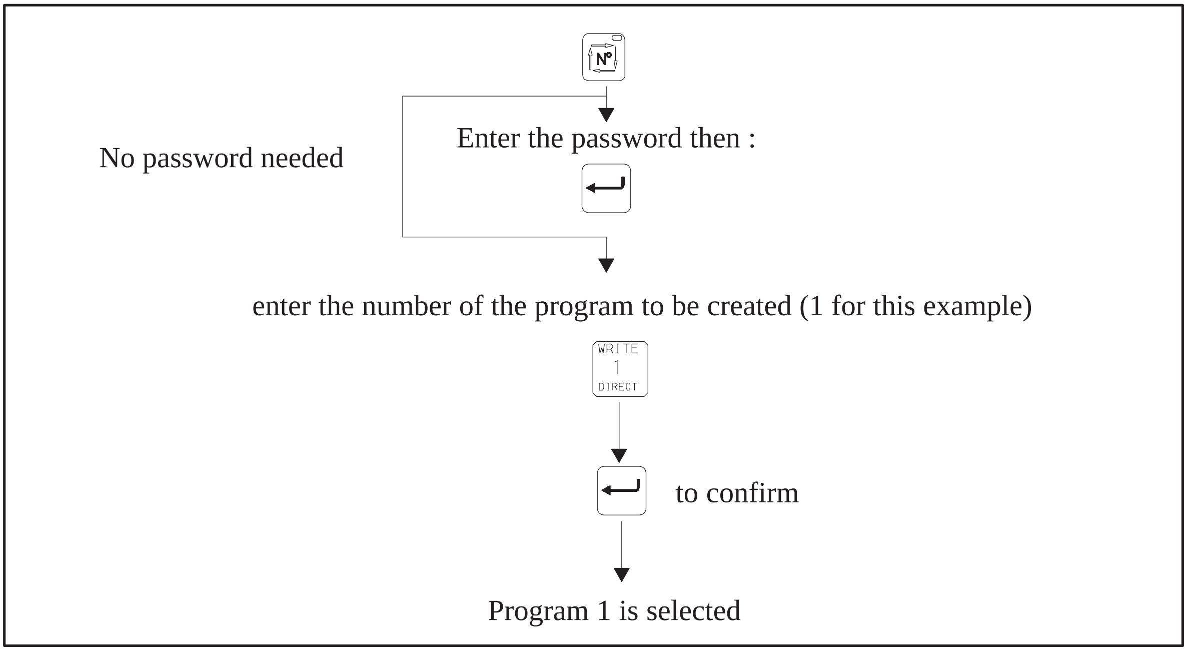
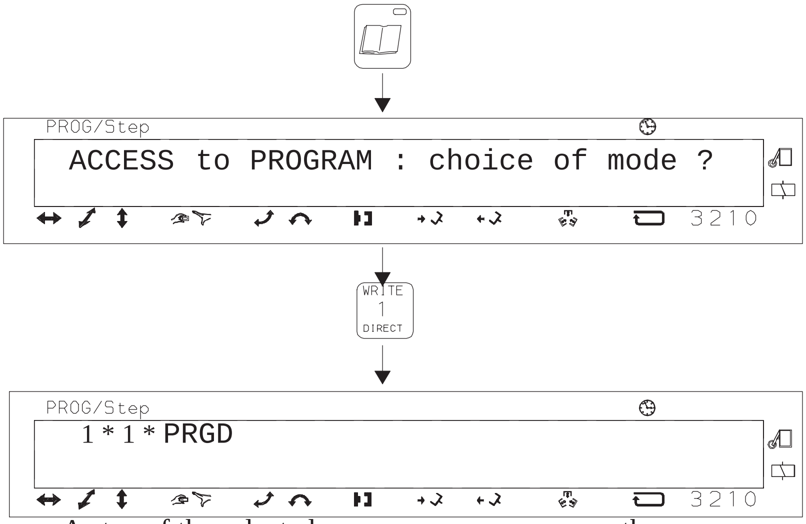

N°
No password needed
Enter the password then :
enter the number of the program to be created (1 for this example)
WRITE
1
DIRECT
to confirm
Program 1 is selected

Figure 2 : Selecting a program

II – 2. Entering a program

II – 2. 1. Accessing the editor



```
PROG/Step
ACCESS to PROGRAM : choice of mode ?
3210
```

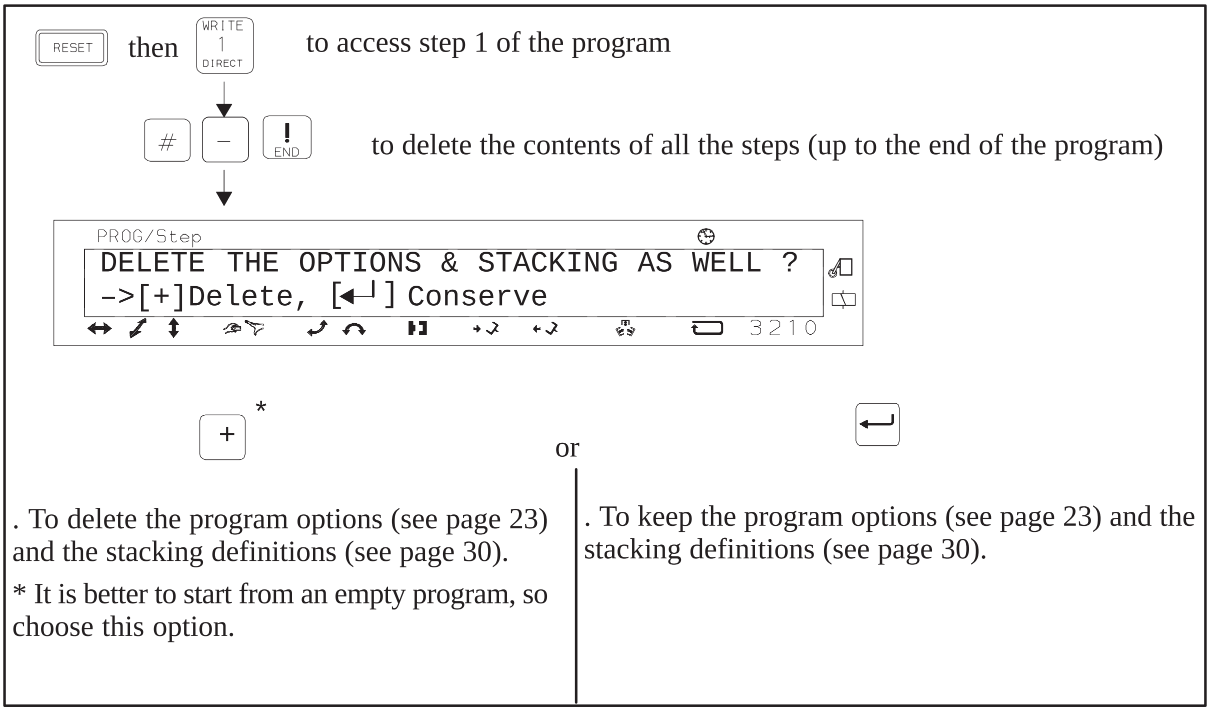
```
PROG/Step
1 * 1 * PRGD
3210
```

A step of the selected program now appears on the screen.

Figure 3 : Accessing the editor

II – 2. 2. Deleting program steps

If the program steps already contain instructions that you do not want to keep, it is better to delete the entire contents of the program at this stage.



RESET then WRITE 1 DIRECT to access step 1 of the program

– ! END to delete the contents of all the steps (up to the end of the program)

```
PROG/Step
DELETE THE OPTIONS & STACKING AS WELL ?
->[+]Delete, [◄┘] Conserve
3210
```

+ * or ◄┘

. To delete the program options (see page 23) and the stacking definitions (see page 30).

* It is better to start from an empty program, so choose this option.

. To keep the program options (see page 23) and the stacking definitions (see page 30).

Figure 4 : Deleting the program steps

Note : Pressing # – ! END deletes the program steps from the step being edited up to the end of the program.

Start–up conditions	STEP 1	↵	Page 2
Disengaging sequence	STEP 2	✂ ⇝ ↵	Page 2
	STEP 3 MPA–II XN	↗ ⇝ ↵	
	STEP 3 MPA–II YN	↗ ⇝ WRITE 1 DIRECT ⇝ 0 ⇝ 0 ⇝ 0 ⇝ ↵ ⇝ ↵	
	STEP 4	↑ ⇝ ↵	
IMM cycle start and positioning of the robot above the IMM	STEP 5	⊡ ⇝ ⊞ ← ⇝ WRITE 1 DIRECT ⇝ 5 ⇝ 0 ⇝ 0 ⇝ ↵ ⇝ ⤶ ⇝ ↵	Page 3
Part grip sequence	STEP 6	↓ ⇝ ↵	Page 3
	STEP 7 MPA–II XN	↙ ⇝ ↵	
	STEP 7 MPA–II YN	↙ ⇝ TEACH 2 APPREN ⇝ 5 ⇝ 0 ⇝ 0 ⇝ ↵ ⇝ ↵	
	STEP 8	⊞ ⇝ ⊞ ⇝ ↵	
	STEP 9	☞ ⇝ ↵	
	STEP 10	⊞ ⇝ ⊞ ⇝ : MPA–II XN ↗ ⇝ ↵ or MPA–II YN ↗ ⇝ WRITE 1 DIRECT ⇝ 0 ⇝ 0 ⇝ 0 ⇝ ↵ ⇝ ↵	
	STEP 11	↑ ⇝ ↵	
IMM cycle restart	STEP 12	⊡ ⇝ ↵	Page 3
Release sequence	STEP 13	→ ⇝ WRITE 1 DIRECT ⇝ 4 ⇝ 0 ⇝ 0 ⇝ 0 ⇝ ↵ ⇝ ↵	Page 3
	STEP 14	⤴ ⇝ ↵	
	STEP 15	↓ ⇝ ↵	
	STEP 16	✂ ⇝ ↵	
	STEP 17	↑ ⇝ ↵	
Belt indexing and loopback	STEP 18	⊟ 0 ⇝ ↵ RESET ⇝ 3 ⇝ ⊟ 0 ⇝ 5 ⇝ 0 ⇝ ↵ ⇝ EXIT ! END ⇝ ↵ 5 ⇝ ↵	Page 4

II – 3. Writing the cycle in SEPRO MPA–II v2 language

```
*****************************************
***  SEPRO MPA–II.v2 YN number 3000   ***
*****************************************
***  PROGRAM No.01                    ***
***  date:                            ***
***  by :                             ***
*****************************************
*****************************************
***  Main program                     ***
*****************************************
Step 1
Step 2
     Release part 1
Step 3
     Y ABSOLUTE 100.0(YN,YS)or Arm 1 back(XN,XS)
Step 4
     Arm 1 up
Step 5
     X ABSOLUTE 150.0
     Gripper vertical
     Ejectors in validation
     machine cycle
Step 6
     Arm 1 total descent
Step 7
     Y ABSOLUTE 250.0 (YN, YS) or Arm 1 forward (XN, XS)
Step 8
     Ejectors out validation
     Ejectors out control
Step 9
     Grip part 1
Step 10
     Y ABSOLUTE 100.0(YN,YS)or Arm 1 back(XN,XS)
     Ejectors in validation
     Ejectors in control
Step 11
     Arm 1 up
Step 12
     machine cycle
Step 13
     X ABSOLUTE 1400.0
Step 14
     Gripper horizontal
Step 15
     Arm 1 total descent
Step 16
     Release part 1
Step 17
     Arm 1 up
Step 18
     Auxiliary Output No.0 maintained for: 5.0 sec
     Program loopback to Step : 5
```

II – 4. The program editor functions

Changes are always made in “Direct Programing” mode.

`PRGD` must be marked on the display.

To select this mode, see “Accessing the editor” chapter II – 2. 1. page 6.

The step marked is the one where the robot was stopped.

The tools used for modifying a program are given below :

II – 4. 1. Moving about within a program

- Direct access to Step 1 : RESET then WRITE 1 DIRECT

- Direct access to the last step of the program : RESET then ! END

- Direct access to the first step of the subroutine : RESET then ↱

- Moving between steps : This is done using + and −

V1.0

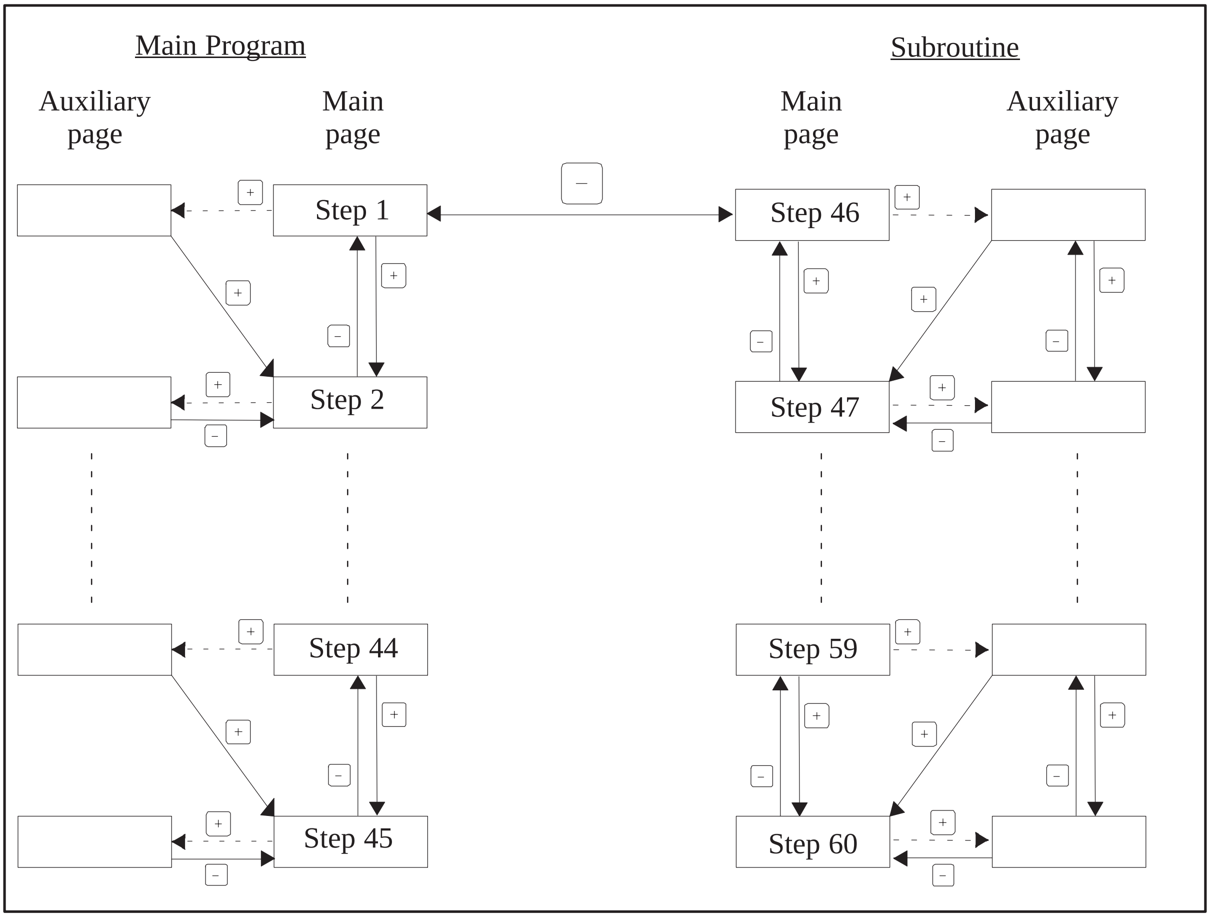


Figure 5 : Moving between program steps

Note : Press ../.. to move between the main page and the auxiliary page (or for the double arm robots).

II – 4. 2. Deleting an instruction

Press the instruction key and # simultaneously and confirm with ↵ or press the key of the instruction you want to delete for a second time.

II – 4. 3. Deleting a step

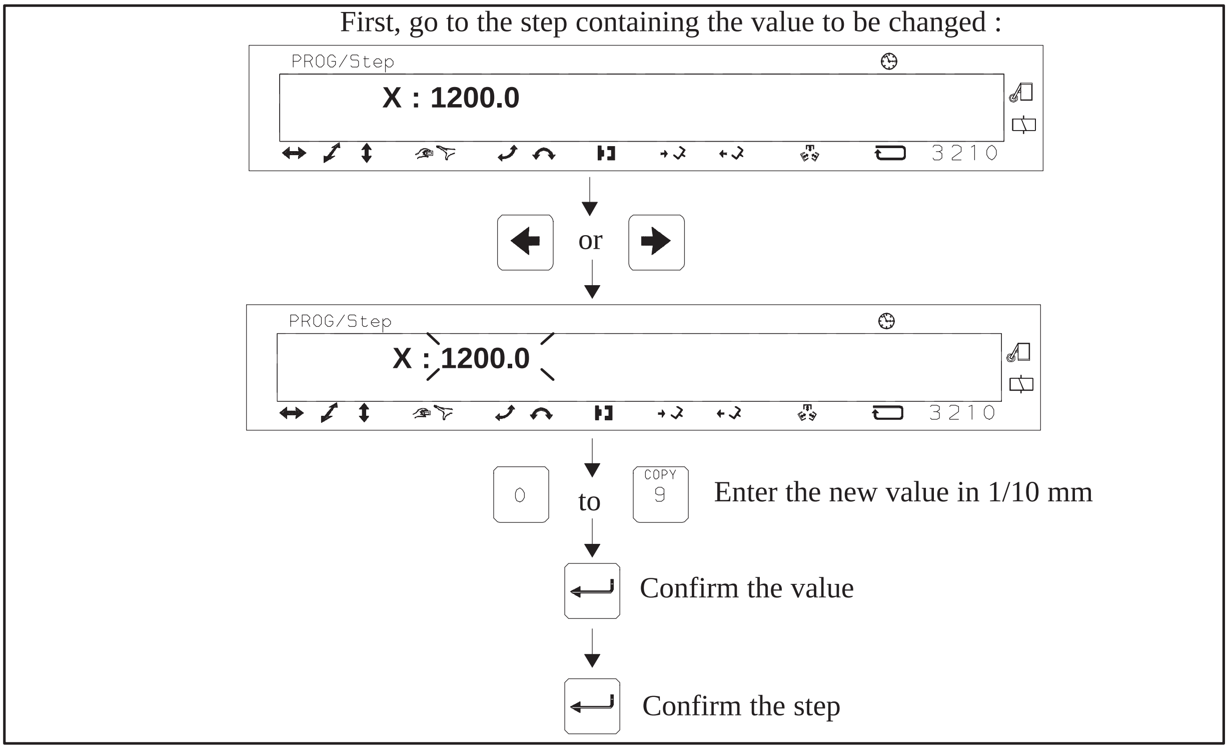
− simultaneously : Deletes the step marked.

Before : step marked –> STEP 3 `Release part 1`
STEP 4 `Sprue–picker arm up`
STEP 5 `Sprue–picker arm back`

After : –> STEP 3 `Sprue–picker arm up`
STEP 4 `Sprue–picker arm back`

II – 4. 4. Changing a value

The following example explains how to change an X position (← or →). Use the same procedure to change a Y position (↙ or ↗).


First, go to the step containing the value to be changed :

Enter the new value in 1/10 mm

Confirm the value

Confirm the step

Figure 6 : Changing a value

II – 4. 5. Inserting a step :

+ simultaneously : to insert an empty step before the step marked

Before : step marked –> STEP 3 `Release part 1`
STEP 4 `Sprue–picker arm up`
STEP 5 `Sprue–picker arm back`

After : –> STEP 3 *<– Empty*
STEP 4 `Release part 1`
STEP 5 `Sprue–picker arm up`
STEP 6 `Sprue–picker arm back`

III – PROGRAM STRUCTURE

III – 1. Main program

A program is made up of a series of steps corresponding to the movements a robot makes in a cycle.

A program can contain up to 45 Steps, numbered from 1 to 45.

III – 2. Subroutine

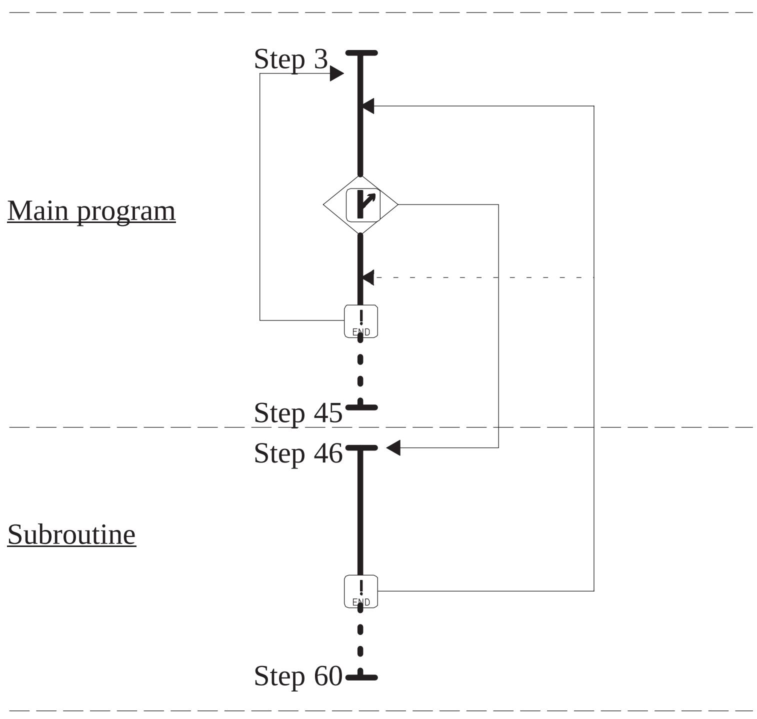
▶ A main program can be diverted to a subroutine upon the appearance of a data item.

▶ A subroutine consists of a series of Steps like the main program. It can contain up to 15 steps numbered from 46 to 60.

▶ At the end of the subroutine, the loopback step to the main program is entered.

▶ The data item that enables you to switch to the subroutine may be :

- The appearance of an AIG (Switching) input which may be memorised or not (Chapter IV – 5. 3. page 24)
- or the internal data item : Cmd at End of layer or Cmd at End of stacking if requested in the stacking header (see ”Stacking” chapter V – 1. 2. page 30).
- or the internal data item : End of cycle counting if requested (see Counters chapter IV – 5. 5. page 25).



III – 3. The program options

Each program can be characterised by the options described in chapter IV – 5. page 23

III – 4. Advice for cycle time optimization

The IMM cycle time is often greater than the robot’s one. Even so, it is always interesting to reduce it, especially the IMM immobilization time.

The IMM immobilization time is the length of time that separates :

- the mould access authorization that the IMM gives to the robot : Mould Open (MO) or Partial Opening Reached (OPA) signal

from

- the Machine Cycle Validation (VCM) that the robot gives to the IMM to authorize the fabrication of a new part.

To reduce the IMM immobilization time, you can use a certain number of instructions and options offered by Sepro, and several rules must be respected to guarantee the equipment safety and the reliability of the cycle.

▶ The cycle time reduction MUST NEVER increase the number of inicidents. In this case, the time lost restarting after an incident cancels out the few tenths of a second gained during optimization.

▶ The paths obtained after optimization must not lead to a risk of collision if, for one reason or another, a part of the movement slows down.

The robot arm must be stopped as close as possible to the mould to wait for the latter to open. The IMM cycle must be restarted as quickly as possible. For this to be possible, the Outside Mould Area (ZHM) cam must be adjusted correctly : the lowest position of the robot arm that enables the mould to open and close without colliding with the robot gripper head.

The “imprecise” instruction is used to optimize the cycle time. Its use is described in chapter IV – 2. 2. page 15.

Using this tool enables you :

- to round the robot’s path as the movements follow on from one another
- to mask the part demoulding movements (ejectors and/or core pullers) by doing them during the robot’s movements.

So that the cycle time optimization is efficient, some of the IMM adjustment parameters must be checked or modified :

- the Partial Opening position (OPA) must be adjusted to authorise the robot arm’s descent as soon as possible (avoiding any risks of collision)
- the IMM movement speeds and accelerations (mould, ejectors and core pullers) must be optimized to a maximum, whilst at the same time respecting the quality of the parts produced and the safety of the equipment.
- if possible, the fastest movements must be respected : for example, if the mould opening is quicker than the part ejection, the cycle time will be less penalised if the opening is large enough to mask the ejectors and/or core pullers back and forward time during the robot movements.

Reminder : the equipment’s productivity is calculated over long production periods. Consequently, stops due to incidents are taken into account.

▶ An efficient cycle time optimization MUST :

- Reduce the part fabrication time

▶ An efficient cycle time optimization MUST NOT :

- Increase the number of reject parts
- Increase the number of incidents

Sepro proposes, as an option, an additional means of cycle time optimization : anticipated restart that enables you to mask the IMM reaction time (see chapter V – 4. page 38).

IV – PROGRAMMING INSTRUCTIONS

IV – 1. The robot’s pneumatic movements

Y demoulding axis (if the Y axis is pneumatic)

↙	Arm 1 forward (XN, XS)
↗	Arm 1 back(XN,XS)

The demoulding movement can be controlled either physically or by a timer (see Configuration mode in the Configuration Manual).

V1.0 If the robot is equipped with a 3–state pneumatic circuit on the Y axis, it is possible to program a “Y free” command in the same way as for a numeric axis : see chapter IV – 2. 4. page 18.

This instruction leads to the purging of the cyclinder’s 2 chambers which enables the ejectors to push the robot’s arm back, for example.

Z vertical axis

↑	Arm 1 up
↓	Arm 1 total descent
⤓	Arm 1 partial descent
⇣	Arm 1 down slowly

- ▶The ascents are always controlled physically, but the descents can be controlled either physically or by a time delay (see Configuration mode in the Configuration Manual).
- ▶The speed of this movement is controlled by high speed pulsed commands whose length is adjusted in the program options (see chapter IV – 5. page 23) or changed when the robot is operating in Automatic (see User Manual).
- ▶When a program is empty, reference values are placed in it. These “default” values are entered in configuration mode (see Configuration manual).

IV – 1. 1. Wrist rotations

↲	Gripper vertical
↱	Gripper horizontal
↻	Rotation 2 + direction
↺	Rotation 2 – direction

These rotations can either be controlled physically or by time delays (see configuration mode in the Configuration manual).

IV – 1. 2. Part grips

If the robot has one part grip circuit			Grip part 1
			Release part 1
If the robot has 2 part grip circuits		WRITE 1 DIRECT	Grip part 1
		WRITE 1 DIRECT	Release part 1
		TEACH 2 APPREN	Grip part 2
		TEACH 2 APPREN	Release part 2

The part presence control can either be permanent or only active in the machine (see programming options page 23)

The number of part grip circuits to be controlled is also entered in the program options (see chapter IV – 5. 4. page 24).

IV – 1. 3. The Sprue–picker arm (Option)

⇧	Sprue–picker arm up
⇩	Sprue–picker arm down
↙	Sprue–picker arm forward
↗	Sprue–picker arm back
	release 3 (sprue)
	grip 3 (sprue)

IV – 2. The robot’s numeric movements

The X axis of an MPA–II robot is always numeric. The Y axis is numeric for the YN or YS robots.

The special commands for the numeric axes are accessible behind the keys. The left one is for the X axis and the right one for the Y axis.

IV – 2. 1. Programming a speed

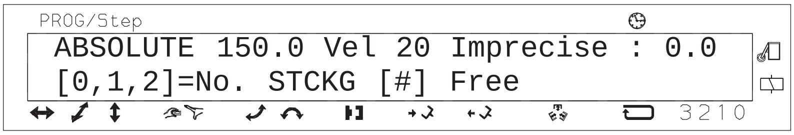
- ▶ The speed of a numeric axis can be changed from 15 to 100% of its maximum value.
- ▶ The change made to the speed in a given step is maintained until it is changed later on in the program.

Programming :

– Using the − and + keys, move to the step where the numeric movement whose speed you want to change is programmed

– Press (left key for X, right key for Y).

The robot displays :



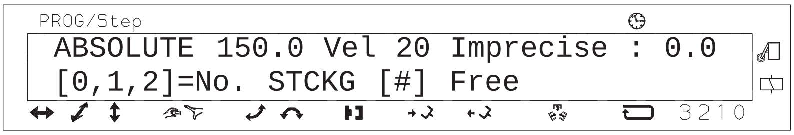
```
PROG/Step
ABSOLUTE 150.0 Vel 20 Imprecise : 0.0
[0,1,2]=No. STCKG [#] Free
                                    3210
```

– Press + to change the active field. The `Vel` message should blink.

– Enter the value of the speed percentage, for example 20.

– Confirm with ↵.

The robot displays :



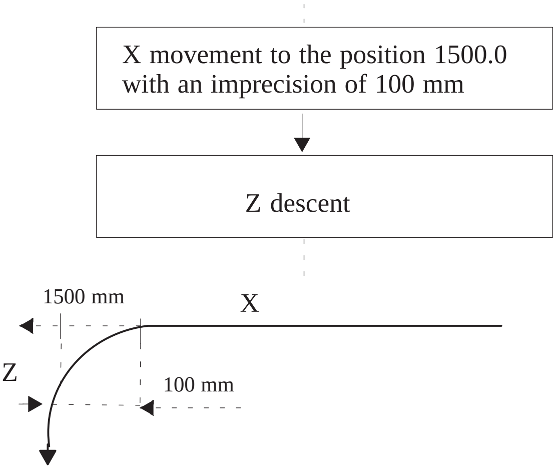
```
PROG/Step
ABSOLUTE 150.0 Vel 20 Imprecise : 0.0
[0,1,2]=No. STCKG [#] Free
                                    3210
```

– Press EXIT.

IV – 2. 2. Programming an imprecision

Using this function, you can anticipate moving on to the next step in order to gain time. However, the final programmed value is still reached.

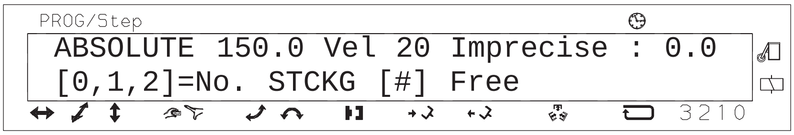
Programming :


X movement to the position 1500.0 with an imprecision of 100 mm
Z descent
1500 mm
X
Z
100 mm

– Using the [−] and [+] keys, go to the step where the numeric movement that you want to make imprecise is programmed.

– Press [⁓] *(*left key for X / right key for Y*)*

The robot displays :



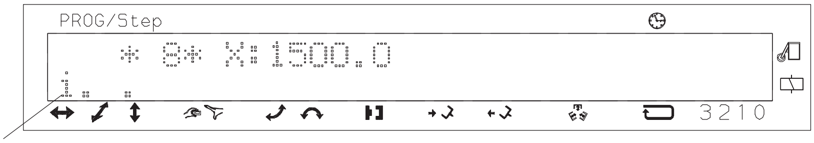
```
ABSOLUTE 150.0 Vel 20 Imprecise : 0.0
[0,1,2]=No. STCKG [#] Free
```

– Press [+] several times to make Imprecise the active field. `Imprecise` should flash.

– Enter the imprecision value, for example *:* 100.0

– Confirm with [↵].

The robot displays :



```
    * 8* X:1500.0
i. .
```

i : indicates an imprecise movement

<u>Display in automatic mode</u> :

– The value of an imprecision can be seen using the [⁓] key when the robot is carrying out this movement.

IV – 2. 3. Programming a slow approach

(Only possible on the X and Y numeric axes)

A movement whose final position is not known. The end of the movement is carried out at slow speed as from a value given in the program step. The movement is stopped by :

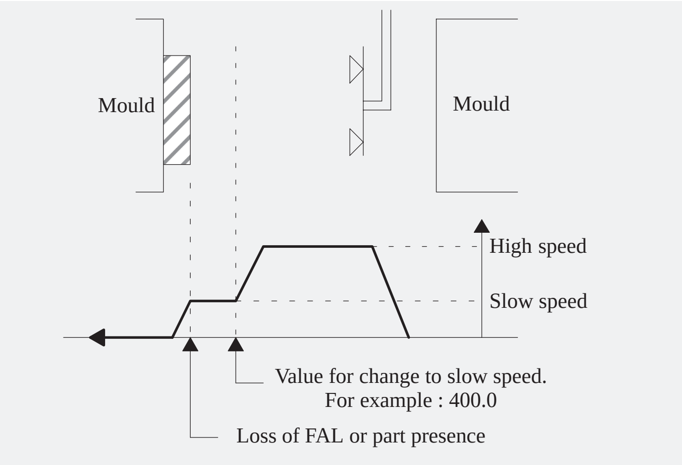
- ▶ the disappearance of the "End of Slow Approach" FAL input (sensor placed on the wrist when the elastic accompaniment option is present).

or

- ▶ the appearance of the part presence control input if the part grip function is programmed in the same step as the slow approach.

Programming :

Slow approach in the mould from 400.0 until contact is made with the part.

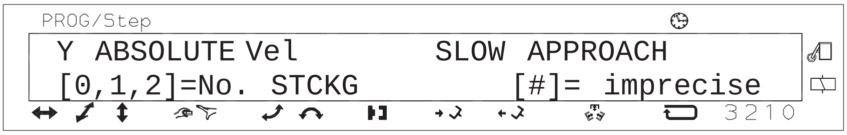


– In the step you have chosen, press the ⁔ key (left for X / right for Y).

– Press + several times to move the active field to Imprecise. ”`Imprecise`” should flash.

– Press # to select "`SLOW APPROACH`".

The robot displays :



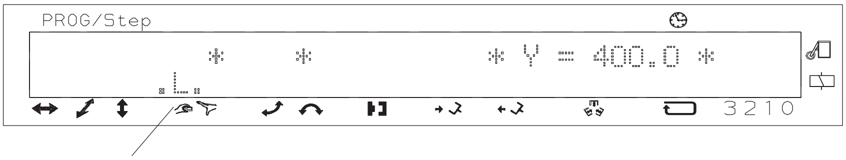
```
PROG/Step
 Y ABSOLUTE Vel        SLOW APPROACH
 [0,1,2]=No. STCKG         [#]= imprecise
                                        3210
```

▶ Confirm with ↵ .

– Press ✓ then 4 0 0 0 .

▶ Confirm with ↵ .

The robot displays :



```
PROG/Step
            *     *           * Y = 400.0 *
         .L.
                                        3210
```

L : indicates a slow approach movement

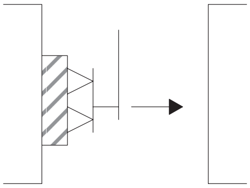
IV – 2. 4. Programming a free axis

This command enable you :

▶to free an axis' brake without controlling its motor. This is only possible for the X and Y axes.

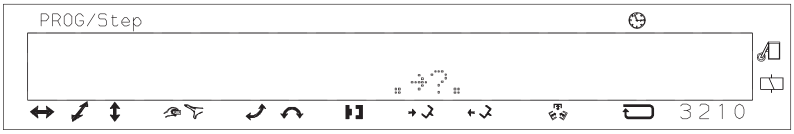
When the Y axis must accompany the part ejection movement.

Programming :



Y axis free during ejectors forward until the ejectors forward control

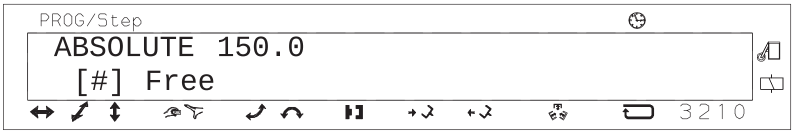
– At the step where the ejectors forward validation is programmed, the robot displays :



```
PROG/Step
3210
```

– Press ▭ (left key for X / right key for Y).

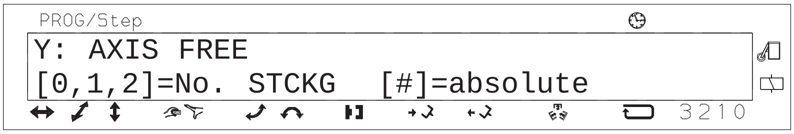
The robot displays :



```
PROG/Step
 ABSOLUTE 150.0
  [#] Free
3210
```

– Press # .

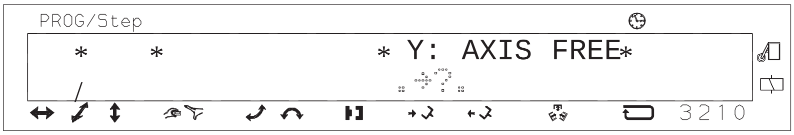
The robot displays :



```
PROG/Step
Y: AXIS FREE
[0,1,2]=No. STCKG  [#]=absolute
3210
```

– Press ↵ .

The robot displays :



```
PROG/Step
  *   *             * Y: AXIS FREE*
  /
3210
```

Note : When the X axis is free, the "–" symbol appears on the bottom left of the screen.

IV – 3. The commands concerning the IMM

This instruction validates the machine cycle.

- ▶ This validation is memorized until a part is produced. During this time, the robot can no longer enter the area occupied by the mould. This is a software safety.
- ▶ During normal operation, if the robot has not entered the mould, the IMM closing validation will be ignored. There is no risk that the IMM cycle will be started twice.
- ▶ Each time this key is pressed, the corresponding command's status will be reversed.

This instruction operates in different ways depending on the robot's position.

- ▶ Robot outside mould : Wait for mould open with a part.
- ▶ Robot in the mould : Validation and waiting for complete mould opening.

Ejector validation is checked or not, depending on the need.

- ▶ Press one of these 2 keys once to program the validation.
- ▶ Press twice to control this validation.
- ▶ Press three times to cancel the instruction.

WRITE 1 DIRCCT   TEACH 2 APPRCN Core puller validation is controlled or not, depending on the need.

- ▶ Press one of these 2 keys once to program the validation.
- ▶ Press twice to control this validation.
- ▶ Press three times to cancel the instruction.

IV – 4. The other instructions

IV – 4. 1. Programming an auxiliary input

There are 8 auxiliary inputs :

▶ Standard : E0 – E1 – E2 – E3

▶ Optional : E4 – E5 – E6 – E7

An input is programmed by pressing the following keys simultaneously :

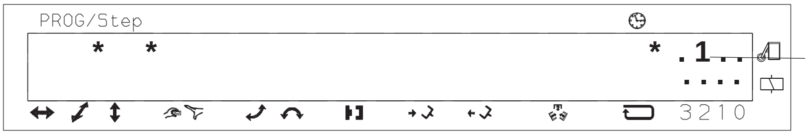
▶ 1 and 0 ... 7 to test a high level input (logic 1)

▶ 0 and 0 ... 7 to test a low level input (logic 0)

Programming a high level tested input.

– Press 1 and TEACH 2 APPREN simultaneously.

The robot displays :



Test symbol for the high level input No. 2

– Confirm by pressing ↵ .

IV – 4. 2. Programming an auxiliary output

There are 8 auxiliary outputs :

▶ Standard : S0 – S1 – S2 – S3

▶ Optional : S4 – S5 – S6 – S7

An output is programmed by pressing the following keys simultaneously :

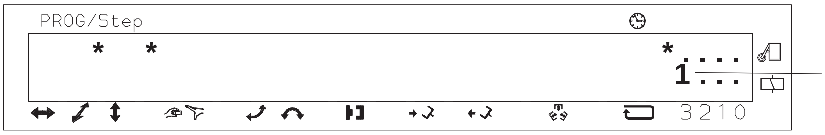
▶ and 0 ... 7

The length of time an auxiliary output is maintained is declared in the OPTION menu using the RESET key. This length of time does not affect the duration of the step in which the output is programmed.

Programming output 3.

– Press and SETUP 3 CONFIG simultaneously.

The robot displays :



Activation symbol for output No. 3

– Confirm by pressing ↵ .

IV – 4. 3. Switching to the subroutine

(see chapter III – 2. page 11).

IV – 4. 4. End and loopback instruction

(see “program end” page 4).

IV – 4. 5. Programming a time delay

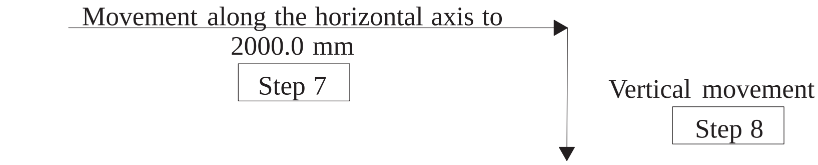
Three types of time delays can be programmed. The time delays to maintain the auxiliary outputs are described in chapter IV – 5. 1. page 23.

The step change time delay

This time delay only commences once the movements requested in the step have been completed. It delays the step change by the value programmed using the 🕒 key.

This value is given in 1/10 secs in 2 figures (9.9 secs max).

Note : The auxiliary commands programmed in the same step are maintained.

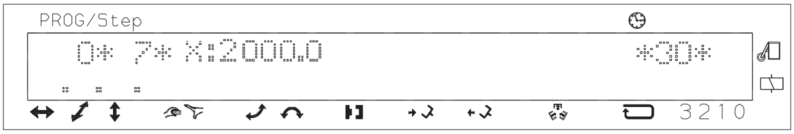


We want to delay the movement from Step 7 to Step 8 by 3 seconds (30 1/10 secs)

The time delay is entered as follows :

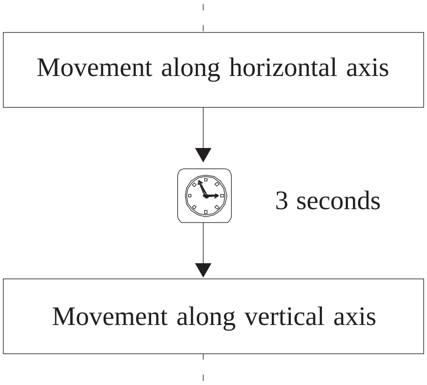
– In Step 7, press 🕒 then 3 then 0 then ↵ .

The robot displays :


```
PROG/Step
  0* 7* X:2000.0                    *30*
                                          3210
```

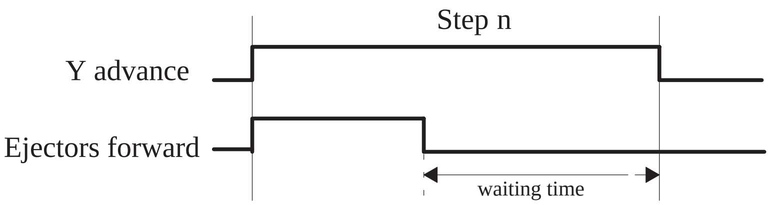
– Validate the step by pressing ↵ .

Result :



The ejection time delay

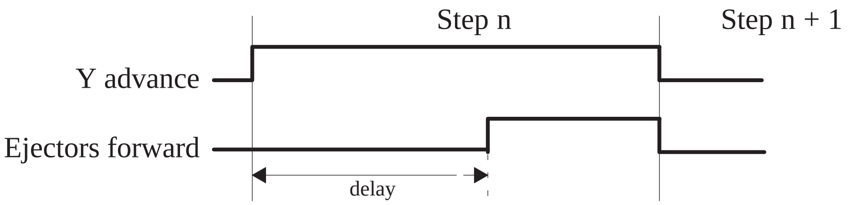
If the “Y advance” and “Ejectors forward” movements are programmed in the same step n, their execution speed can never be the same.



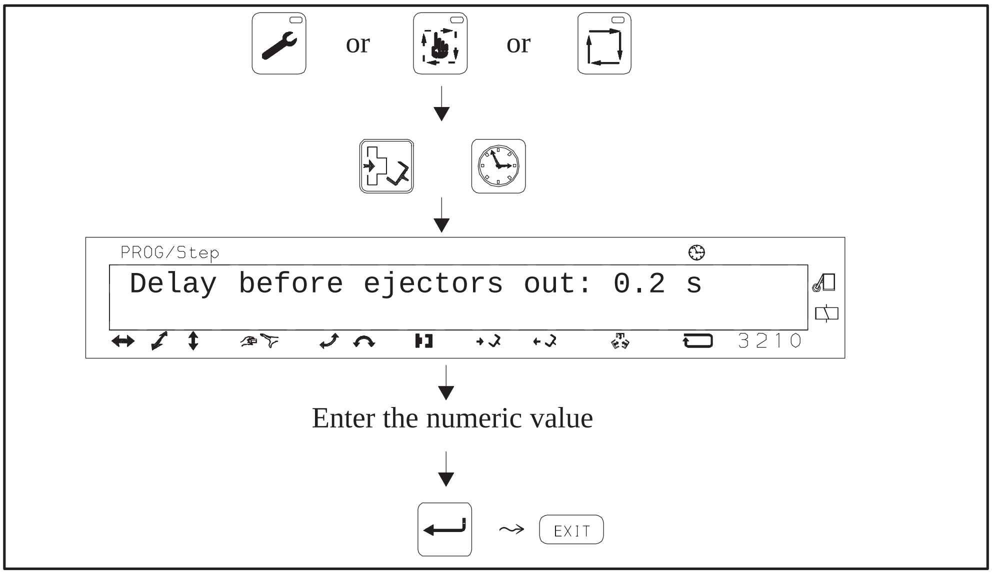
The waiting time marked on the diagram may lead to the following problems :

- ▶ the part is dropped,
- ▶ conflict between the robot arm and the part.

The special ejector time delay enables you to synchronise the end of the execution of the two movements.



The delay value is entered in or or mode by pressing the keys simultaneously. Only one value is possible per cycle and it can be changed during operation. It is given in 1/100 secs with 3 figures (9.99 secs max).



PROG/Step
Delay before ejectors out: 0.2 s
3210

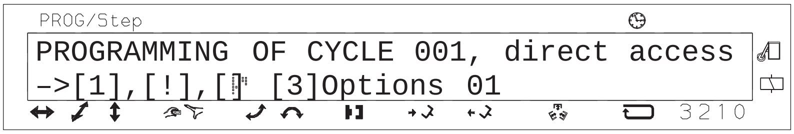
Figure 7 : Ejection delay

Note : If the ejection movement is very long, it can be programmed at the same time as the descent into the mould. The delay is then corrected to synchronise the part grip with the end of the ejection.

IV – 5. The program options

In direct programming mode (+ WRITE 1 DIRECT)

Press RESET and the following menu appears :

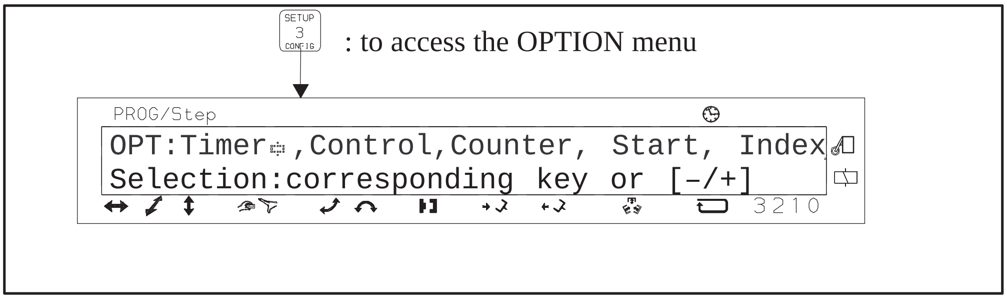


```
PROG/Step
PROGRAMMING OF CYCLE 001, direct access
->[1],[!],[  [3]Options 01
3210
```

WRITE 1 DIRECT : to go directly to the first step of the program.

! END : to go directly to the end of the program.

: to go directly to the first step of the subroutine (Step 46).



SETUP 3 CONFIG : to access the OPTION menu

```
PROG/Step
OPT:Timer ,Control,Counter, Start, Index,
Selection:corresponding key or [-/+]
3210
```

Figure 8 : Access to program options

This menu is used to define the operating conditions for a given program.

IV – 5. 1. Time delays associated with auxiliary outputs

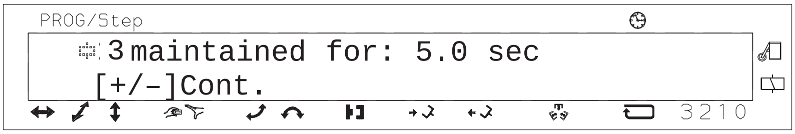
Used to assign a holding time to outputs S0 to S7 with a maximum value of 25 secs.

Access : + 0 ... 7

Use the + key to select the following auxiliary output.

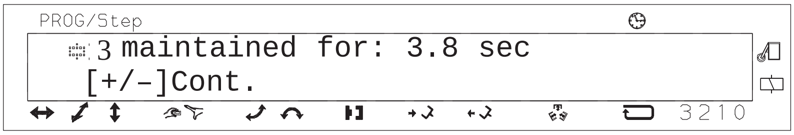
V0.3 The time delay operates in adjust mode , which means that you can index a belt for example.

Press SETUP 3 CONFIG –> to access output 3.



```
PROG/Step
   3 maintained for: 5.0 sec
   [+/-]Cont.
3210
```

Press SETUP 3 CONFIG 8 then .



```
PROG/Step
  3 maintained for: 3.8 sec
  [+/-]Cont.
3210
```

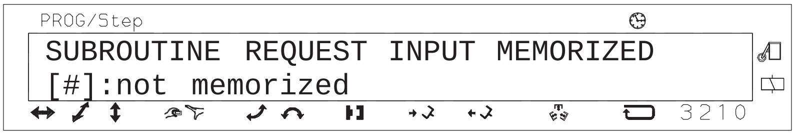
Output S3 will be maintained for 3.8 secs starting from the step where it was programmed, whatever the length of the step.

IV – 5. 2. IMM anticipated restart (option)

Access : (see description in chapter V – 4. page 38)

IV – 5. 3. Memorizing a switching input

Access :



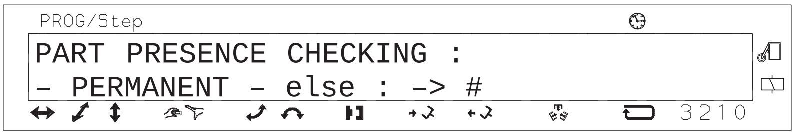
```
PROG/Step
SUBROUTINE REQUEST INPUT MEMORIZED
[#]:not memorized
3210
```

This enables the switching input to be memorized or not. Depending on the applications, it may be preferable to take the current status of the switching input into account or, on the contrary, to memorize the active status of this input before carrying out a particular function.

IV – 5. 4. Part grip control

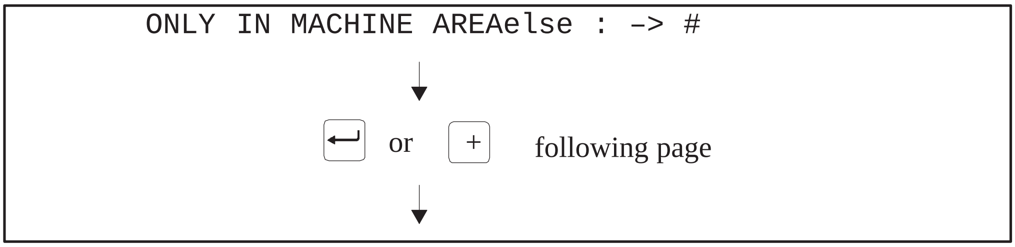
Access: or

V0.3 Used to define the areas where the part grip controls are actuated.



```
PROG/Step
PART PRESENCE CHECKING :
- PERMANENT - else : -> #
3210
```

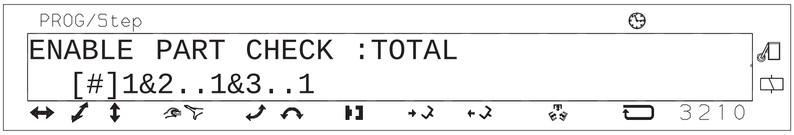
Using #, you can change the selection



```
ONLY IN MACHINE AREAelse : -> #
```

↵ or + following page

Used to define which part grip circuits are controlled when the part is gripped.



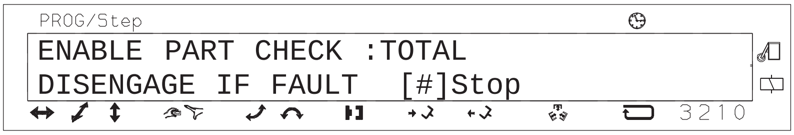
```
PROG/Step
ENABLE PART CHECK :TOTAL
  [#]1&2..1&3..1
3210
```

Using #, you can change the selection.

▶`TOTAL`	: part grips 1, 2 and 3 (selected by default)
▶`1 & 2`	: part grips 1, 2 ; the presence of part 3 is not controlled
▶`1 & 3`	: part grips 1 and 3 ; the presence of part 2 is not controlled
▶`1 only`	: part grip 1 only ; the presence of parts 2 and 3 are not controlled

Confirm with ↵.

This window is used to define how the robot should react when it loses a part.



```
PROG/Step
ENABLE PART CHECK :TOTAL
DISENGAGE IF FAULT  [#]Stop
```

Using # , you can change the selection :

- ▶ Disengaging if part fault (selected by default) : if the part is not taken after 6 secs, the robot disengages automatically and stops as soon as the mould area is free.
- ▶ Stop if part fault : if the part is lost, the robot stops.

In both cases, an alarm is activated.

Confirm with ↵.

IV – 5. 5. Cycle counter

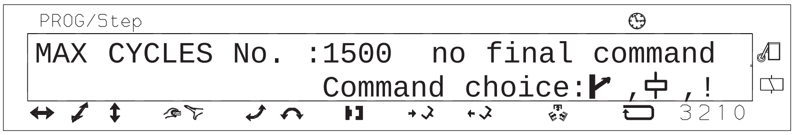
Access : 8888

This is used :

- ▶ to select a number of cycles to be completely carried out,
- ▶ to define an action to be carried out when the counter reaches the selected value.

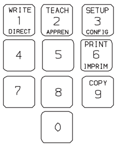
The counter increases at every loopback in the program or subroutine.

However, the loopback step must be inferior to the actual step.



```
PROG/Step
MAX CYCLES No. :1500  no final command
                 Command choice:Ⱶ ,□ ,!
```

Enter the cycle number (from 1 to 9999)



Confirm with ↵ .

Enter the type of action at the end of counting :

- ▶ Ⱶ to select switching ; in this case, you need the Ⱶ instruction in the program.
- ▶ ! END to select a stop at the end of the cycle,
- ▶ □ then the output No. then the Step No. where this auxiliary output is activated,
- ▶ # to select nothing.

Confirm with ↵ .

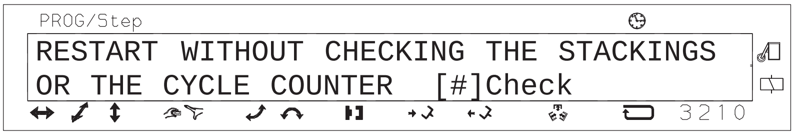
IV – 5. 6. Start–up conditions after having stopped a stacking sequence

Access :

Two possibilities :

▶ Following a stoppage, the stacking counters are not changed.

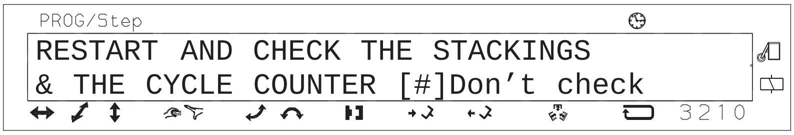
The selection is :



```
PROG/Step
RESTART WITHOUT CHECKING THE STACKINGS
OR THE CYCLE COUNTER  [#]Check
                                          3210
```

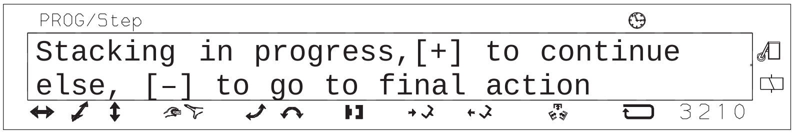
▶ Following a stoppage, the robot asks if it should set the stacking counters back to zero and carry out the end of stacking actions respectively.

The selection is :



```
PROG/Step
RESTART AND CHECK THE STACKINGS
& THE CYCLE COUNTER [#]Don’t check
                                          3210
```

If the counters are neither at zero or their maximum, the robot asks the following question each time you start it up again :

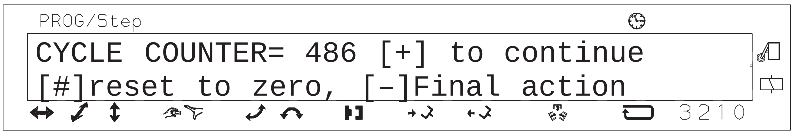


```
PROG/Step
Stacking in progress,[+] to continue
else, [–] to go to final action
                                          3210
```

▶ Press + to continue : the cycle is continued without changing the counters.

▶ Press – to go to the final phase. The end of stacking counters (and not end of layer or stack) are validated. The stacking counters are set to zero.

If the counter is neither at zero or at its maximum, the following question is asked :



```
PROG/Step
CYCLE COUNTER= 486 [+] to continue
[#]reset to zero, [–]Final action
                                          3210
```

The action is the same as for the stackings. # is used to reset the counter to zero.

IV – 5. 7. Offset wait position

Access: ← or →

This function described in chapter V – 3. page 36 depends on the robot’s configuration (see the Configuration manual, file R).

IV – 5. 8. Memorizing the index positions

Access : [|⇔|]

The indices are memorized positions that can be used in programming and adjust modes.

When a numeric axis key is pressed in Programming mode, the associated index is automatically displayed (providing its value has already been entered).

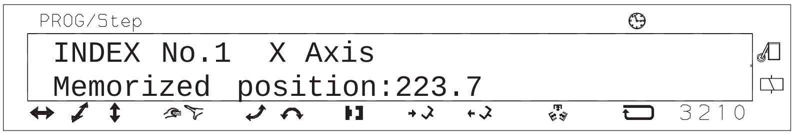
In Adjust mode, an index position acts as a stop as soon as the [|⇔|] key has been pressed.

There are 2 index positions for the X axis (indices 1 and 2), 2 for the Y axis when the robot is on the machine side (indices 3 and 4) and 2 for the Y axis when the robot is on the release side (indices 5 and 6).

Move the axis to the position to be memorized.

[📖] => [|⇔|]

The robot displays :



```
PROG/Step
 INDEX No.1  X Axis
 Memorized position:223.7
                                  3210
```

Press [+] or [−] until the index number that you want to use appears.

Press the 2 axis' keys simultaneously :

- ▶ for X : [←] and [→] simultaneously.
- ▶ for Y : [↙] and [↗] simultaneously.

=> The robot's actual position is taken for the index.

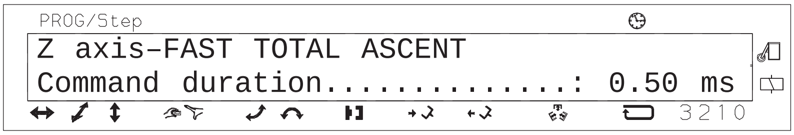
Confirm with [↵].

A different value can also be entered using the numeric keyboard.

Note : The indices are very useful as they make the Y axis manual movements in the IMM safer in particular and make it easier to enter values when programming. You are advised to enter the values before you start programming the cycles.

IV – 5. 9. Length of the high speed pulses for the vertical axes

Access : [↑] or [⭳] or [↓] for the main arm and the sprue–picker arm.



```
PROG/Step
Z axis–FAST TOTAL ASCENT
Command duration..............: 0.50 ms
                                  3210
```

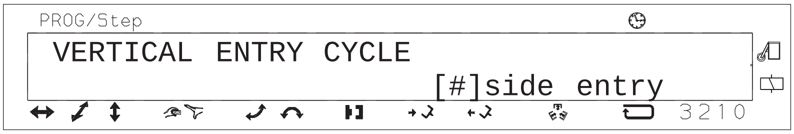
These values must be adjusted to obtain rapid movements with no shocks at the end of the stroke.

Note : The longer the length of the pulse, the more the axis is brisk and the more the shocks on the dampers are violent.

V1.0

IV – 5. 10. Type of IMM access

This OPTION is only available if side–entry access has been defined as possible in Context mode (see the Configuration manual in File R).



```
PROG/Step
 VERTICAL ENTRY CYCLE
                               [#]side entry
                                                  3210
```

Used to define the type of IMM access. Vertical access is when the robot enters the mould from above ; side–entry access is when the robot accesses the IMM from the side. For the latter, the robot arm is down and it is the X axis that moves.

See description in chapter V – 5. page 41.

V – SPECIAL PROGRAMMING FUNCTIONS

V – 1. Release and part stacking (palletization)

V – 1. 1. Introduction

Each cycle includes 3 stacking sequences numbered from 0 to 2.

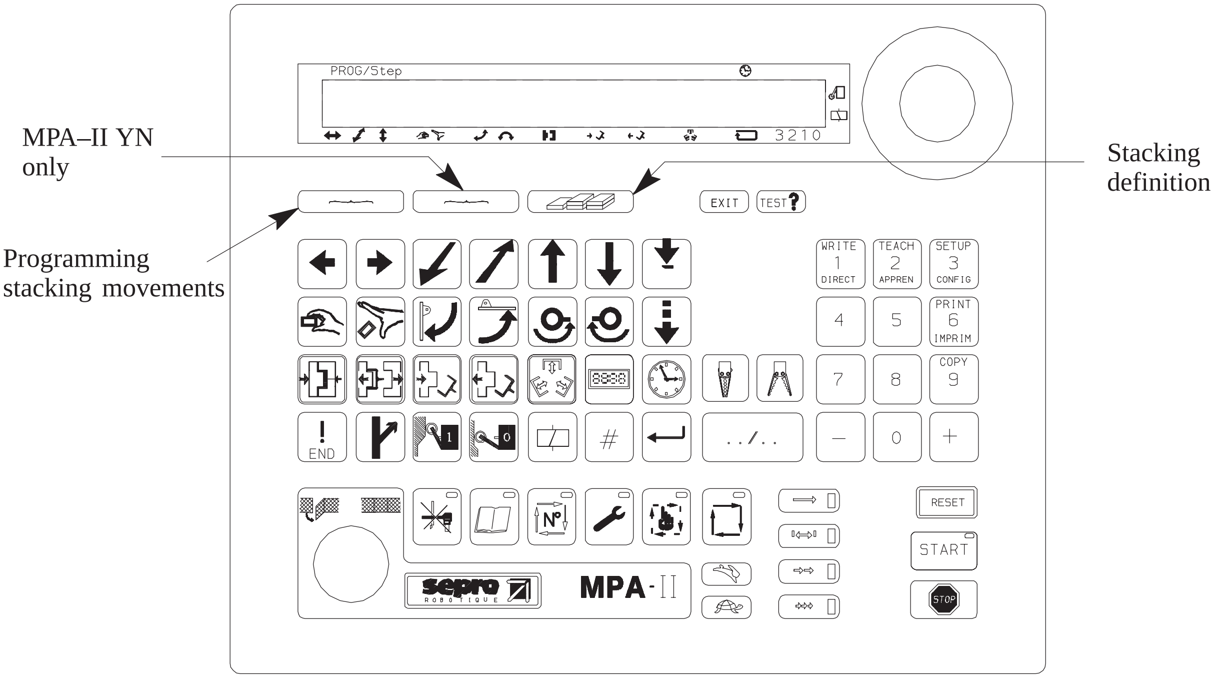
A stacking sequence is programmed in two stages :

▶ 1st Stage : Defining the stacking

Available using the ▭ key, a dialogue menu appears that enables the pallet organization to be easily defined.

▶ 2nd Stage : Programming the movements to stack the 1st part

Using the ▭ keys



Before programming a stacking sequence, you should become acquainted with the following keys:

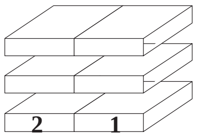
- ▭ : to access the menus for the definition.
- # : to select an option.
- ↵ : to validate and continue with the definition.
- + : to continue with the definition.
- EXIT : to return to step programming.

V – 1. 2. Defining a stacking sequence

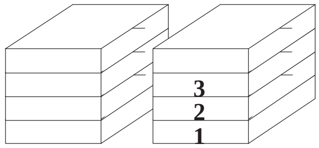
▶ Press to enter the stacking information :

- see the screens in the example on page 32

▶ Defining the stacking N° : 0, 1 or 2
and the type :

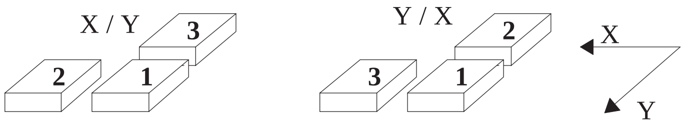


By layer



By stack

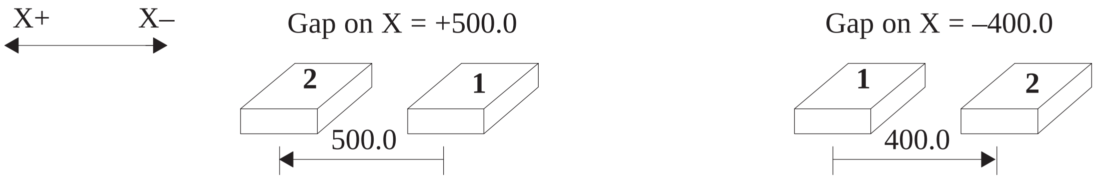
▶ Defining the order in which the robot will stack the parts (only for YN)



▶ Defining the number of parts on the X horizontal axis (max number of parts = 99)

▶ Defining the value of the X gap between parts (from 0 to ± 3200.0 mm) and the direction.

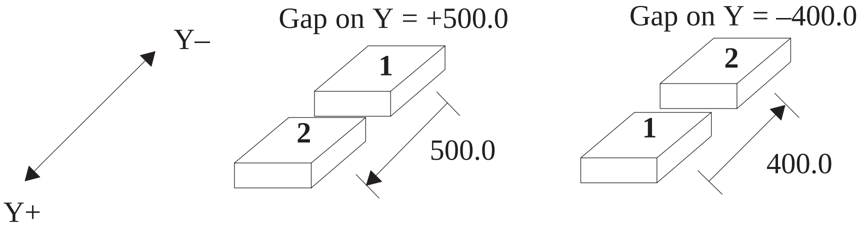
- enter the value using the numerical keys and the negative/positive sign using #.



▶ Defining the number of parts on the Y horizontal axis (max number of parts = 99)

▶ Defining the value of the Y gap between parts (from 0 to ± 3200.0 mm) and the direction.

- enter the value using the numerical keys and the negative/positive sign using #



▶ Defining the number of layers (Max number = 999).

▶ Defining the command at the end of the layer.

Action that the robot must carry out at the end of each stacking layer. Warning : the end of the last stacking layer is not considered as an end of layer, but as an end of stacking.

Command possibilities at end of layer :

- ⊢ : to execute a subroutine starting from the step where this instruction is programmed.
- ▭ : to activate an auxiliary output
  - ▭ ? : enter the output N° then ↲ .
  - ▭ `..at S` ?? : enter the Step N° from which the output will be controlled.

The text “at end of layer” corresponds to stacking in layers, otherwise you are asked “at end of stack”.

▶ Definition of the command at the end of stacking

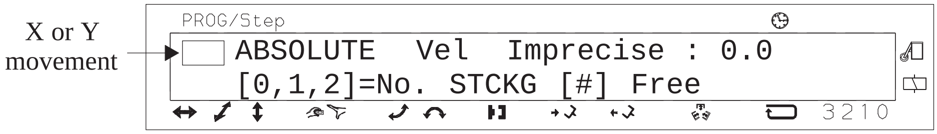
Action that the robot must carry out when the stacking has finished.

Command possibilities at end of stacking :

- ⊢ : to execute a subroutine
- ▭ : to activate an auxiliary output
  - ▭ ? : enter the output N° then ↲ .
  - ▭ `..at S` ?? : enter the Step N° from which the output will be controlled.

V – 1. 3. Programming the stacking movements

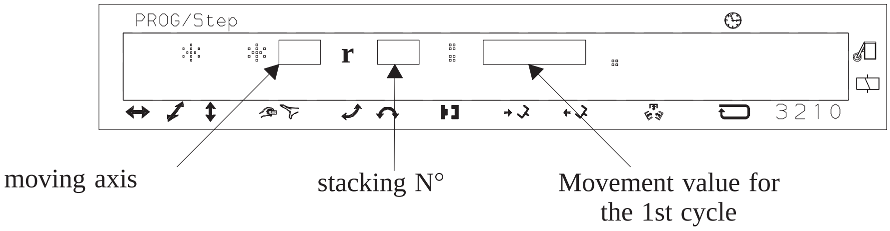
First of all a stacking number is allocated to a movement using the ⁀ key which displays the extended programming menu.



```
PROG/Step
□ ABSOLUTE  Vel  Imprecise : 0.0
  [0,1,2]=No. STCKG [#] Free
                          3210
```

Choose the stacking number using the 0 WRITE 1 DIRECT and TEACH 2 APPREN keys. The stacking number is validated with ↲.

▶ Next, the value of the movement for the 1st cycle is entered. You must return to the step programming with the EXIT key to do this.



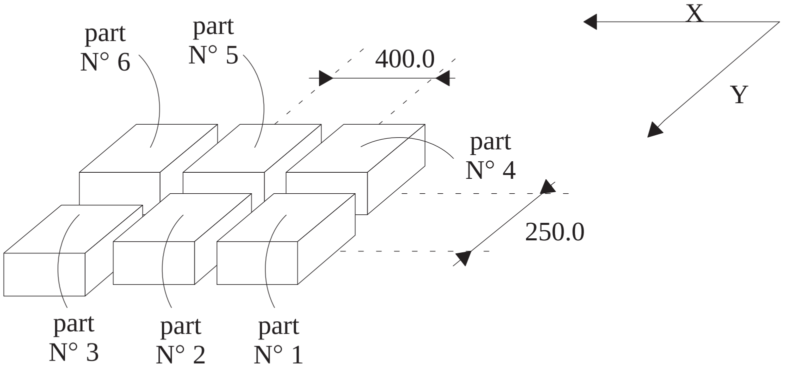
```
PROG/Step
        □ r □      □
                          3210
```

The letter of the stacking axis is followed by the letter r (from the French word for stacking, ’rangement’) and the stacking number.

The position is now entered from the keyboard. This position will be increased for each cyle by the value of the gap defined in the stacking using the ▱ key (see chapter V – 1. 1. page 29).

V – 1. 4. Examples of stacking sequences

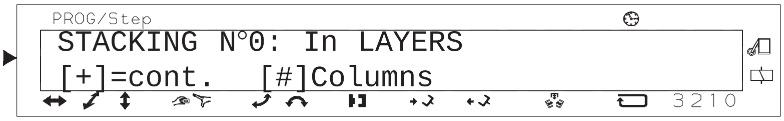
▶ Defining a stacking sequence



The stages marked with YN only apply to YN type robots (2 numeric axes).

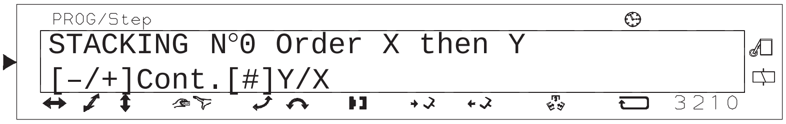
This is how you define this stacking sequence :

1. 0



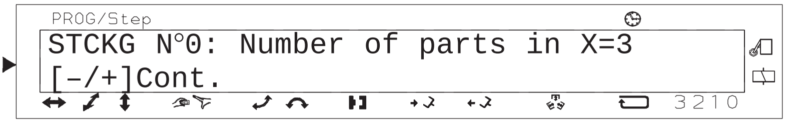
```
STACKING N°0: In LAYERS
[+]=cont.  [#]Columns
```

2. ↵



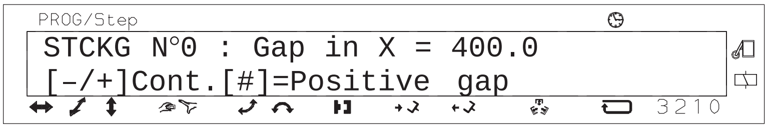
```
STACKING N°0 Order X then Y
[–/+]Cont.[#]Y/X
```

3. ↵ SETUP 3 CONFIG



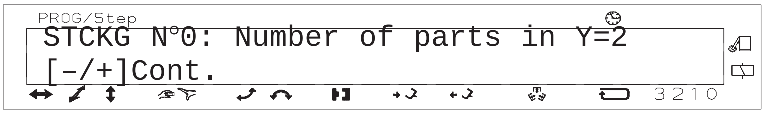
```
STCKG N°0: Number of parts in X=3
[–/+]Cont.
```

4. ↵ 4 0 0 0



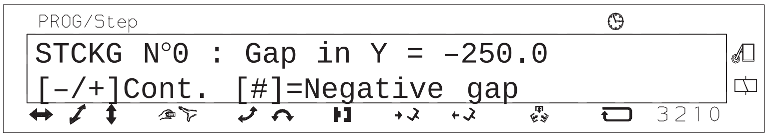
```
STCKG N°0 : Gap in X = 400.0
[–/+]Cont.[#]=Positive gap
```

YN 5. ↵ TEACH 2 APPRCN



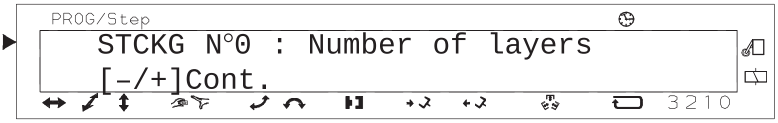
```
STCKG N°0: Number of parts in Y=2
[–/+]Cont.
```

YN 6. ↵ TEACH 2 APPRCN 5 0 0 #



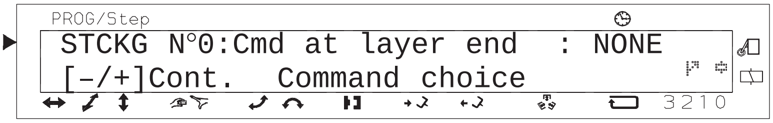
```
STCKG N°0 : Gap in Y = –250.0
[–/+]Cont. [#]=Negative gap
```

7. ↵



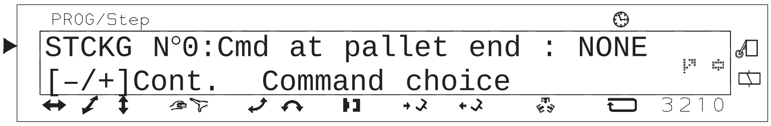
```
STCKG N°0 : Number of layers
[–/+]Cont.
```

8. ↵



```
STCKG N°0:Cmd at layer end  : NONE
[–/+]Cont.  Command choice
```

9. ↵

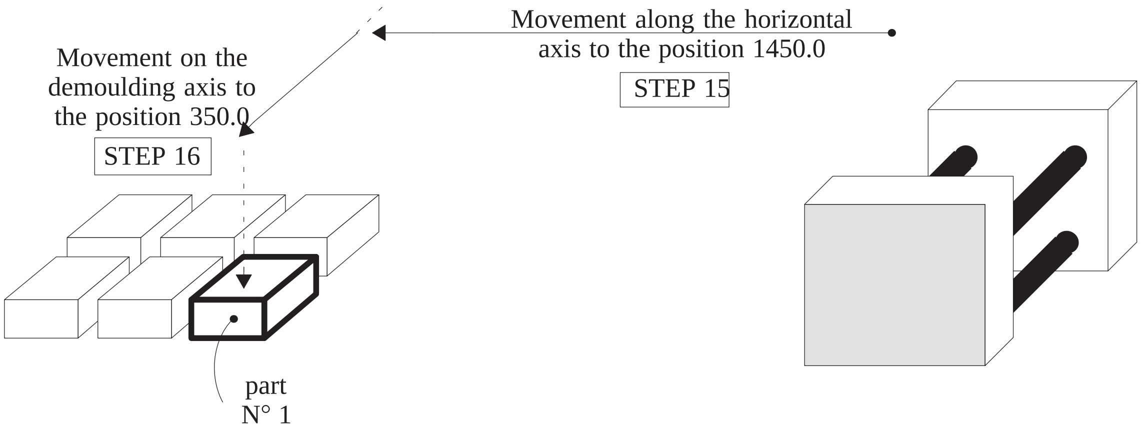


```
STCKG N°0:Cmd at pallet end : NONE
[–/+]Cont.  Command choice
```

10. ↵

Stacking sequence N° 0 has now been entered.

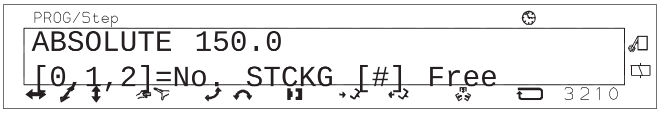
▶ Programming the movements


Movement along the horizontal axis to the position 1450.0
STEP 15
Movement on the demoulding axis to the position 350.0
STEP 16
part N° 1

The two numeric movements are programmed as follows :

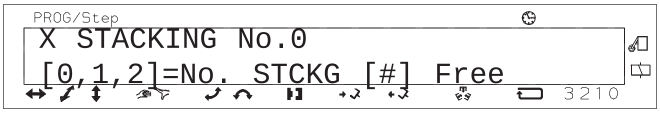
Let us suppose that we are going to program Step 15.

1. [⁀] located above the keys [←] [→]



```
PROG/Step
ABSOLUTE 150.0
[0,1,2]=No, STCKG [#] Free
3210
```

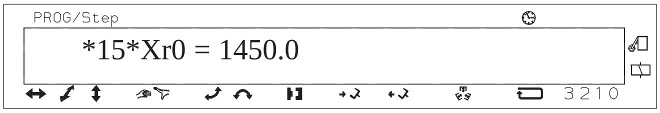
2. [0] [↵]



```
PROG/Step
 X STACKING No.0
 [0,1,2]=No. STCKG [#] Free
3210
```

3. [EXIT]

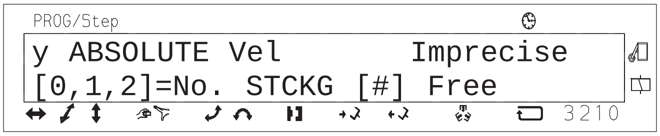
4. [←] [WRITE 1 DIRECT] [4] [5] [0] [0] [↵]



```
PROG/Step
     *15*Xr0 = 1450.0
3210
```

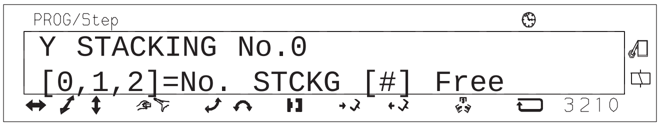
5. Press [↵] to continue onto the next step.

YN 6. [⁀] located above the keys [↙] [↗]



```
PROG/Step
y ABSOLUTE Vel       Imprecise
[0,1,2]=No. STCKG [#] Free
3210
```

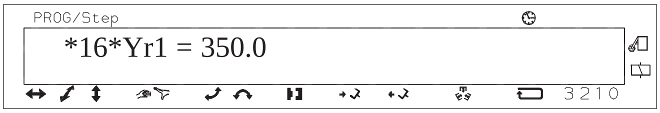
YN 7. [0] [↵]



```
PROG/Step
 Y STACKING No.0
 [0,1,2]=No. STCKG [#] Free
3210
```

YN 8. [EXIT]

YN 9. [↙] [SETUP 3 CONFIG] [5] [0] [0] [↵]

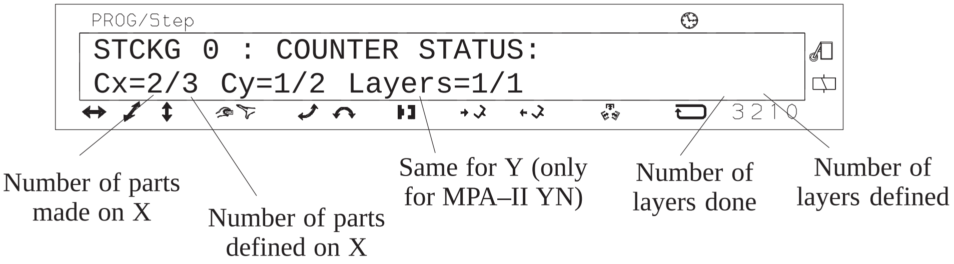


```
PROG/Step
    *16*Yr1 = 350.0
3210
```

10 Confirm with [↵]

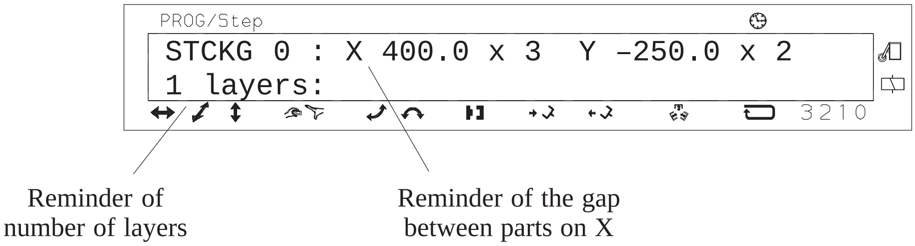
V – 1. 5. Viewing the stacking sequences :

key : 1st pulse : COUNTERS



```
PROG/Step
STCKG 0 : COUNTER STATUS:
Cx=2/3 Cy=1/2 Layers=1/1
3210
```

key : 2nd pulse : GAPS



```
PROG/Step
STCKG 0 : X 400.0 x 3  Y –250.0 x 2
1 layers:
3210
```

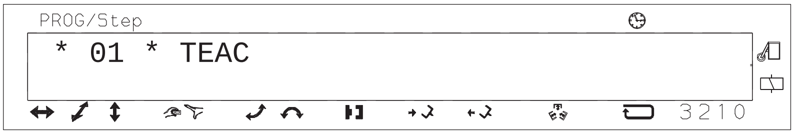
V – 2. Programming in Teach mode

In the teach programming mode ”TEAC”, the robot performs the cycle at the same time as you program it.

This mode is chosen after having selected the number of the cycle that you are going to create. The indices must already have been entered.

To access teach mode : 📖 –> TEACH 2 APPRCN

The robot displays :

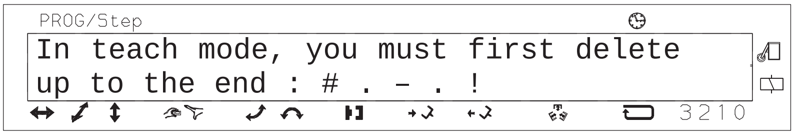


```
PROG/Step
 * 01 * TEAC

                                   3210
```

Press ↵

The robot displays :

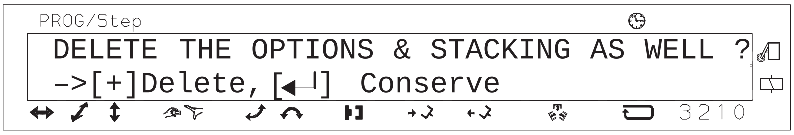


```
PROG/Step
In teach mode, you must first delete
up to the end : # . – . !
                                   3210
```

This is to ensure that the program is empty before you start programming.

Press # – ! END simultaneously.

The robot displays :

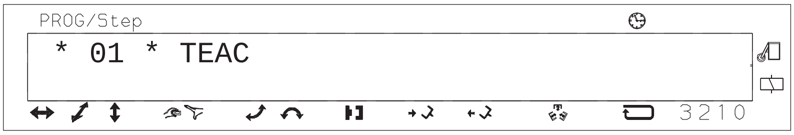


```
PROG/Step
 DELETE THE OPTIONS & STACKING AS WELL ?
 –>[+]Delete,[↵] Conserve
                                   3210
```

You must now decide whether to keep or delete the information associated with the program (information contained in Option (chapter IV – 5. page 23) and Stacking sequences (chapter V – 1. page 29)).

Press + or ↵ depending on your choice.

The robot once again displays :



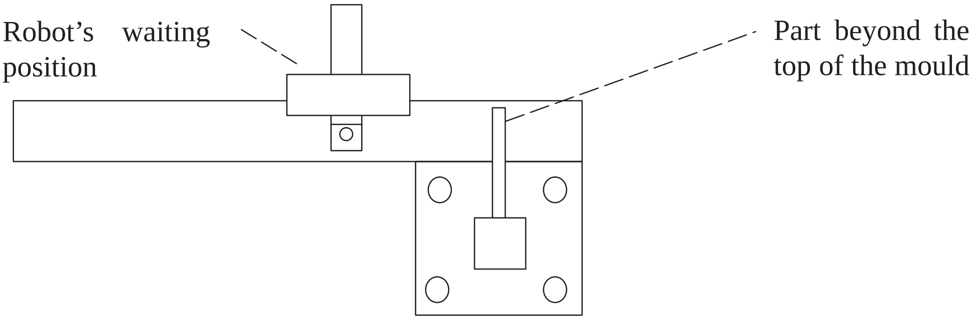
```
PROG/Step
 * 01 * TEAC

                                   3210
```

Enter the different cycle movements as for direct programming (see chapter II – “Programming a cycle” page 5), not forgetting that all the actions programmed in a same step are carried out simultaneously.

In Teach mode, the robot carries out the movements at the same time as they are programmed. The `MVts COMPLETED` message enables you to change step.

V – 3. Programming a cycle with offset wait compared to the IMM axis



If the robot is not able to approach the mould whilst it is not open because of the mould dimensions, we talk about a cycle with an offset wait compared to the IMM's axis.

When the robot is in the IMM area, it must stop the mould closing whatever the operating mode (adjust, without robot, step by step or automatic).

When the mould is not open, the robot access must be prohibited whatever the operating mode (adjust, without robot, step by step or automatic).

Programming :

In this type of cycle, the first stage that describes the robot disengaging sequence, must contain the position of the X axis outside of the IMM.

Only then can you enter the machine cycle validation instruction (`machine cycle`) and `machine opening control` which is obligatory.

The return to the IMM axis, the descent, part grip, etc ... will be in the next step.

Configuration :

In the robot's configuration, you define whether it is obligatory or not to enter an offset wait position (see Configuration manual).

▶ Facultative data :

In this case, you can choose in each program whether the "Machine Area Free" is programmed or determined automatically :

- Programmed "Machine Area Free" : you must enter an X position as from which the robot is sufficiently free from the IMM to authorize the mould to close.

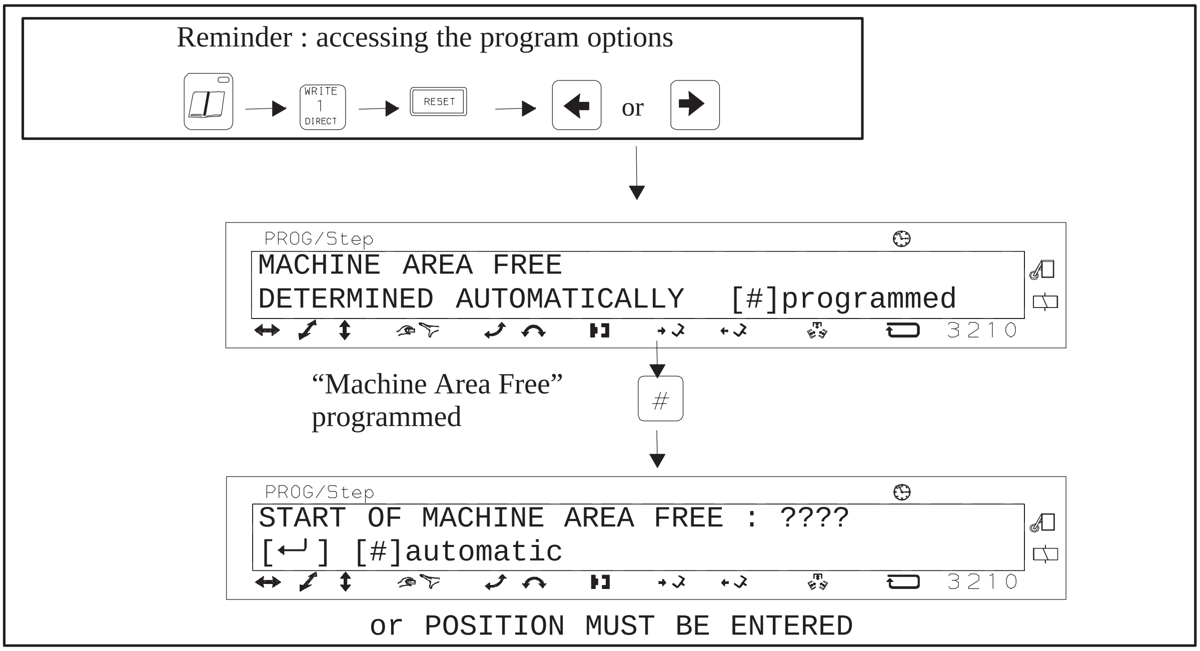
As long as the cycle has not been executed, the Machine Area Free corresponds to ZBD. The mould must be open to authorize the robot movements outside of ZBD.

- "Machine Area Free" determined automatically : the robot analyses the cycle and defines this area by itself. It corresponds to the ZBD cam if the cycle is programmed with an offset wait. It corresponds to the whole of the X stroke if the IMM cycle is restarted when the robot is on the Machine Axis (AM).

If the robot never interferes with the IMM when the Z arm is up, a value greater than the machine axis can be programmed.

▶ Obligatory data :

The cycle can only be used if a "Machine Area Free" has been entered in the program options (see figure 9 : page 37).



Reminder : accessing the program options

```
PROG/Step
MACHINE AREA FREE
DETERMINED AUTOMATICALLY   [#]programmed
3210
```

“Machine Area Free” programmed

```
PROG/Step
START OF MACHINE AREA FREE : ????
[↵] [#]automatic
3210
```

or POSITION MUST BE ENTERED

Figure 9 : Offset wait

V – 3. 1. Signalling

When you run cycles containing an offset wait, the following messages may appear on the screen.

▶ If you try to command the mould closing or select the Without robot mode whilst the latter is badly placed, you may get :

```
*ROBOT BADLY PLACED !*
```

and
```
MACHINE COMMAND: MOVE THE ROBOT
          BEYOND POSITION...
```

or

```
MACHINE COMMAND: MOVE THE ROBOT
  ONTO ZBD CAM
```

▶ If the mould is not open and if the robot is no longer in the Machine Free area :

```
ROBOT/MACHINE AREAS NOT INITIALIZED
MACHINE MUST BE OPEN OUTSIDE ZBD
```

or

```
ROBOT OUTSIDE MACHINE FREE AREA :
Must have MACHINE OPEN (or ADJ & START)
```

▶ If the robot configuration imperatively requires an“Machine Area Free” and this has not been entered in the program options for the current program :

- INCOMPLETE CYCLE : MACHINE FREE VALUE?
- Prog. then [RESET] then [↵]

V – 4. IMM anticipated restart (option)

Aim :

This shortens the cycle time by masking the IMM reaction time (time between the closing authorization from the robot and the actual mould movement).

Principle :

The machine cycle validation (VCM) and the arm free safety (SBD) are given when the robot is still inside the mould.

Type of anticipated restart :

There are two types of anticipated restart :

- anticipated restart with a programmed delay,
- auto–adaptative anticipated restart.

The type of restart is chosen in the options of each program.

Conditions :

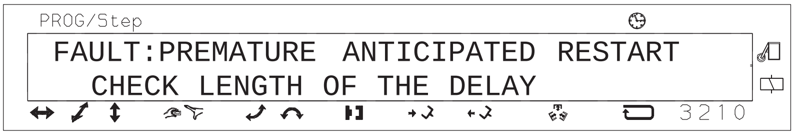
The IMM anticipated restart is only effective if :

- ▶ the robot is in automatic mode,
- ▶ the overall speed coefficient Kv = 100 %

Safety :

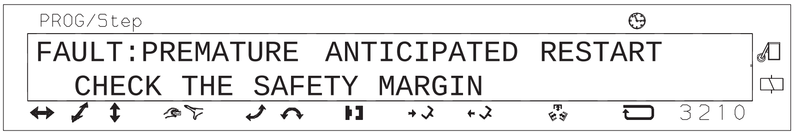
If one of the data items mould open (MO) or partial opening reached (OPA) disappears whilst the robot is still inside the mould, the robot goes into fault and immediately interrupts the mould closing authorization commands.

If Mould Open (MO) or Partial Opening Reached (OPA) are lost and the robot is not outside of the mould, the robot stops and displays :



```
FAULT:PREMATURE ANTICIPATED RESTART
  CHECK LENGTH OF THE DELAY
```

for anticipated restart with a programmed delay.

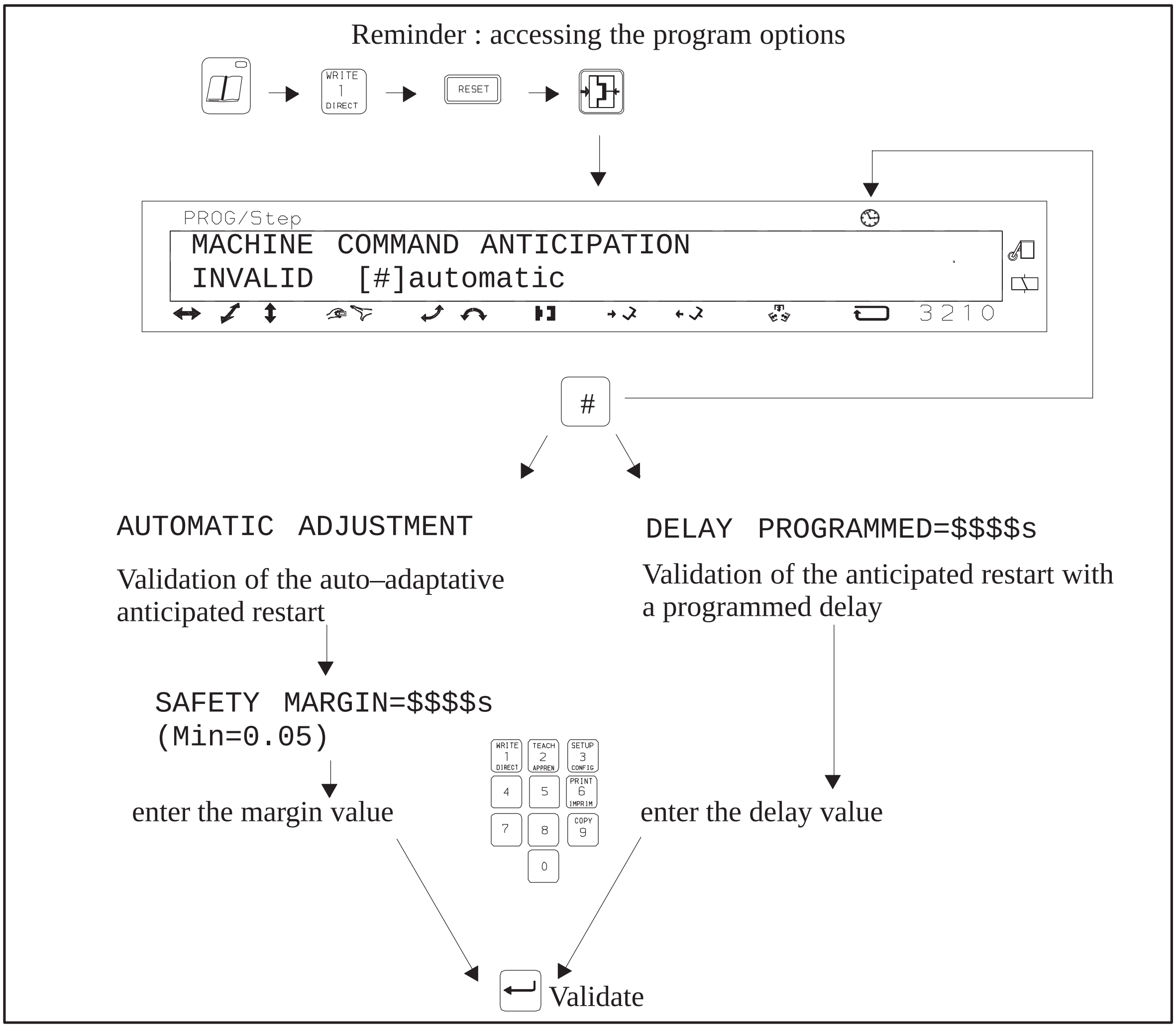


```
FAULT:PREMATURE ANTICIPATED RESTART
  CHECK THE SAFETY MARGIN
```

for an auto–adaptative anticipated restart.

This option is only available if the software that validates it is installed.

A special cabling must also be added and defined in the configuration.



Reminder : accessing the program options

Validation of the auto–adaptative anticipated restart

Validation of the anticipated restart with a programmed delay

enter the margin value

enter the delay value

Validate

Figure 10 : Anticipated restart

V – 4. 1. Anticipated restart with a programmed delay

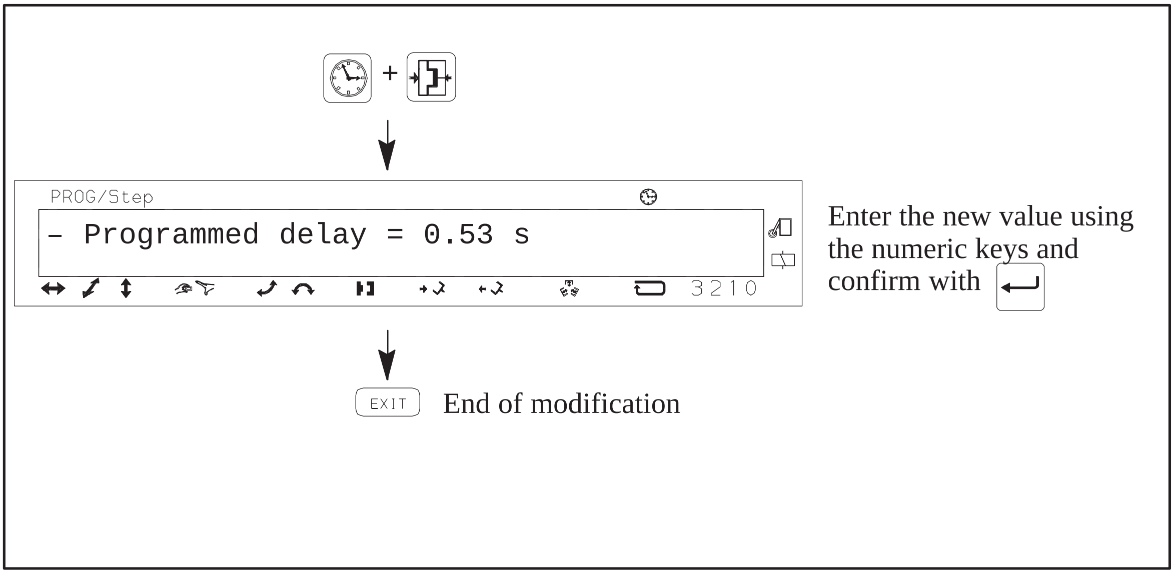
The programmed delay which enables you to optimize the IMM restart is applied between :

- ▶ the beginning of the movement which frees the robot from the mould,

and

- ▶ the activation of the commands that authorize the mould closing.

To optimize the anticipation, it is possible to change the length of the delay in automatic mode.



```
PROG/Step
– Programmed delay = 0.53 s
3210
```

Enter the new value using the numeric keys and confirm with ↵

End of modification

V – 4. 2. Auto–adaptative anticipated restart

In this case, the robot calculates by itself the delay applied between the beginning of the movement that frees the robot from the mould and the activation of the commands the authorize the mould to close.

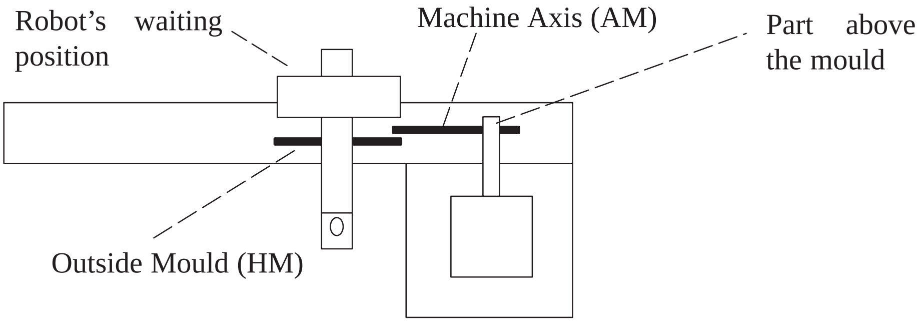
The margin represents the minimum tolerated duration between the arrival of the robot in an Out of Mould area and the loss of the Mould Open information. This is to avoid accidents if there are large differences in the IMM reaction times. Each time the robot stops, a delay of 5 seconds is applied. The calculation is made by successive tries. It will be optimal after several cycles in automatic mode.

V1.0

V – 5. Unloading tie–barless Injection Moulding Machines

When the IMM does not have tie–bars, the MPA–II robot can be programmed and configured for side–entry access to the mould. This configuration is useful when it is not possible to access the mould vertically, as the X axis' movement to reach and remove the part generally takes longer than the Z axis' movement for the same operation.

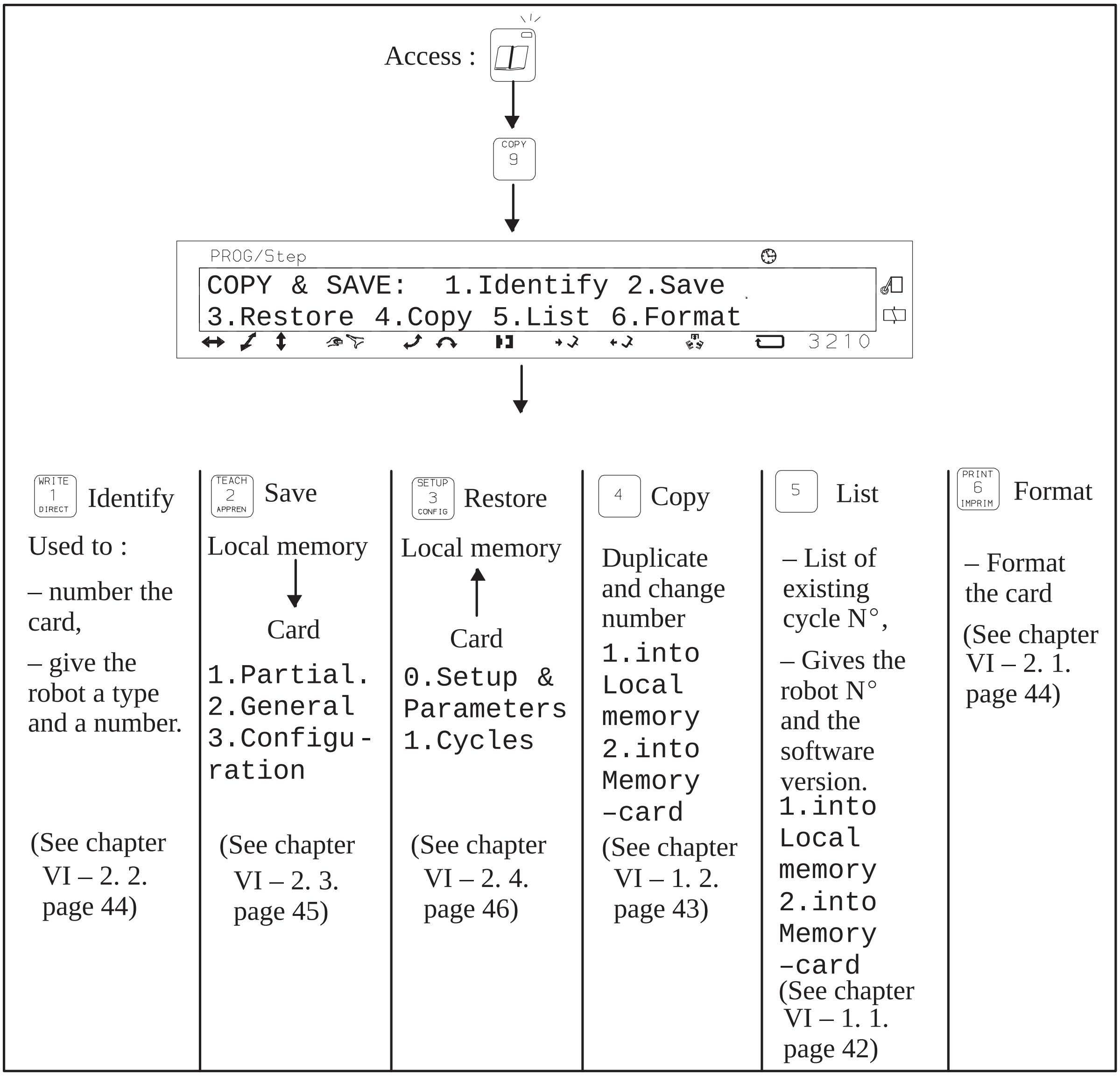
To ensure the safety of the machines, this type of access is only possible if additional cams and sensors have been installed on the robot beam.



Side–entry access must be defined as being possible in Context mode (see the Configuration manual in File R).

The cycle must be defined with side–entry access in the program options (see chapter IV – 5. 10. page 28).

VI – MEMORY MANAGEMENT : COPY MODE



Access :

COPY 9

```
PROG/Step
COPY & SAVE:  1.Identify 2.Save
3.Restore 4.Copy 5.List 6.Format
3210
```

WRITE 1 DIRECT Identify

Used to :

– number the card,

– give the robot a type and a number.

(See chapter VI – 2. 2. page 44)

TEACH 2 APPREN Save

Local memory

Card

```
1.Partial.
2.General
3.Configu-
ration
```

(See chapter VI – 2. 3. page 45)

SETUP 3 CONFIG Restore

Local memory

Card

```
0.Setup &
Parameters
1.Cycles
```

(See chapter VI – 2. 4. page 46)

4 Copy

Duplicate and change number

```
1.into
Local
memory
2.into
Memory
-card
```

(See chapter VI – 1. 2. page 43)

5 List

– List of existing cycle N°,

– Gives the robot N° and the software version.

```
1.into
Local
memory
2.into
Memory
-card
```

(See chapter VI – 1. 1. page 42)

PRINT 6 IMPRIM Format

– Format the card

(See chapter VI – 2. 1. page 44)

Figure 11 : Memory management (summary)

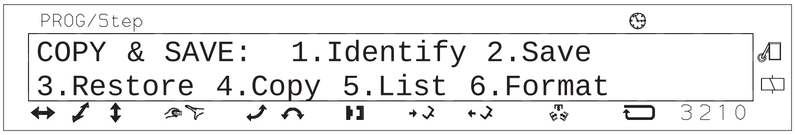
VI – 1. The local memory

VI – 1. 1. List of the programs

This function lists the programs stored in :

- ▶ either the internal local memory,
- ▶ or the external memory card.

Press COPY 9 to view the Copy mode main menu.



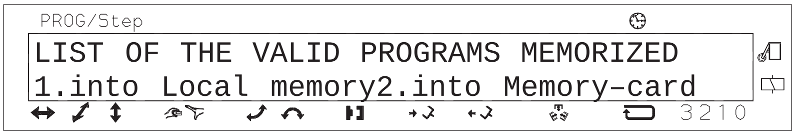
```
PROG/Step
COPY & SAVE:  1.Identify 2.Save
3.Restore 4.Copy 5.List 6.Format
                                   3210
```

Procedure for listing the programs in the local memory :

▶ => COPY 9

▶ Press 5

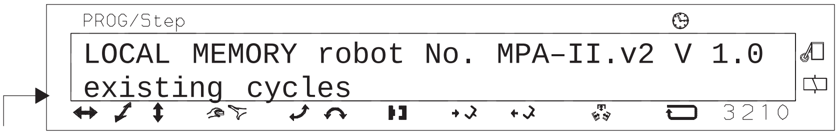
The robot displays :



```
PROG/Step
LIST OF THE VALID PROGRAMS MEMORIZED
1.into Local memory2.into Memory–card
                                   3210
```

Press WRITE 1 DIRECT (press TEACH 2 APPREN to consult the list of programs in the memory card).

The robot displays :



```
PROG/Step
LOCAL MEMORY robot No. MPA–II.v2 V 1.0
existing cycles
                                   3210
```

–> The robot displays the cycles stored in the relevant memory.

VI – 1. 2. Copying a program

This function is used to duplicate a program :

▶ either in the local memory (internal),

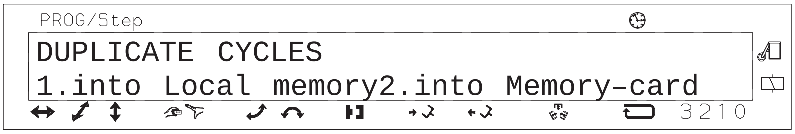
▶ or in the memory card (external).

Procedure for copying a program stored in the local memory :

▶ => COPY 9

▶ Press 4.

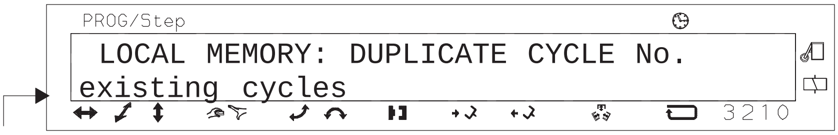
The robot displays :



```
PROG/Step
DUPLICATE CYCLES
1.into Local memory2.into Memory–card
                                   3210
```

Press WRITE 1 DIRECT (press TEACH 2 APPREN to duplicate a program in the card).

The robot displays :

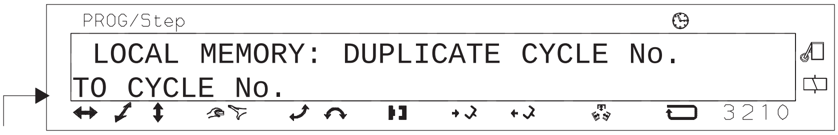


```
PROG/Step
 LOCAL MEMORY: DUPLICATE CYCLE No.
existing cycles
                                   3210
```

The robot displays the cycles stored in the relevant memory.

▶ Enter the number of the program to be duplicated then ↵.

The robot displays :



```
 LOCAL MEMORY: DUPLICATE CYCLE No.
TO CYCLE No.
```

▶ Enter the number of the new program then ↵.

VI – 2. The memory card (option)

This card is a PCMCIA type external memory in which can be saved :

▶ the programs and their options,

▶ the robot’s configuration (values entered in configuration, context and parameter modes).

It is possible to program a cycle directly into the card. It is also possible to execute this program.

VI – 2. 1. Formatting the card

This function enables the card to be initialised by :

▶ erasing the memory,

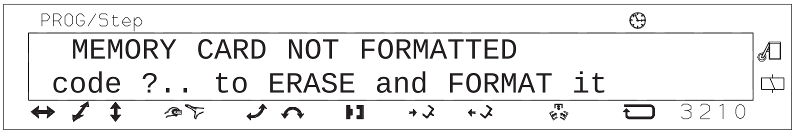
▶ assigning areas to the programs and the robot configuration.

Formatting procedure :

▶ 📖 => COPY 9

▶ Press PRINT 6 IMPRIM.

The robot displays :



```
  MEMORY CARD NOT FORMATTED
 code ?.. to ERASE and FORMAT it
```

▶ Enter the code WRITE 1 DIRECT – TEACH 2 APPREN – SETUP 3 CONFIG – 4 then ↵.

Formatting takes place immediately and you then return to Identification mode.

VI – 2. 2. Identifying the card

This function enables you to customize the card by giving it :

▶ an identification number,

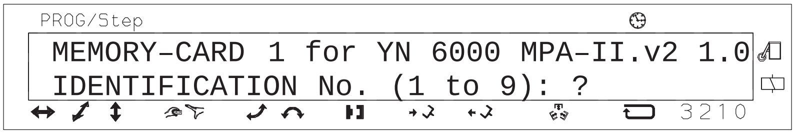
▶ the robot’s type and number.

Identification procedure :

▶ 📖 => COPY 9

▶ Press WRITE 1 DIRECT.

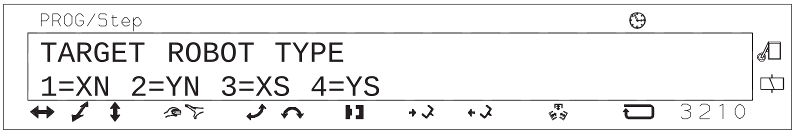
The robot displays :



```
 MEMORY–CARD 1 for YN 6000 MPA–II.v2 1.0
 IDENTIFICATION No. (1 to 9): ?
```

- ▶ Enter the identification number then ↵.

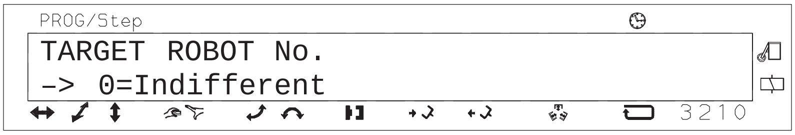
The robot displays :



```
PROG/Step
TARGET ROBOT TYPE
1=XN 2=YN 3=XS 4=YS
3210
```

- ▶ XN : numeric X / asynchronous motor
- ▶ YN : numeric XY / asynchronous motors
- ▶ XS : numeric X / Brushless servo–control motor
- ▶ YS : numeric XY / Brushless servo–control motors
- ▶ Enter the robot type then ↵.

The robot displays :



```
PROG/Step
TARGET ROBOT No.
-> 0=Indifferent
3210
```

- ▶ Enter the robot's serial N° then ↵.
- ▶ This number is preceded by PIP |______| which you can find on :
  - ▪ the end of the beam,
  - ▪ the identification plate (X carriage or cabinet),
  - ▪ the Customer File.

VI – 2. 3. Saving a program

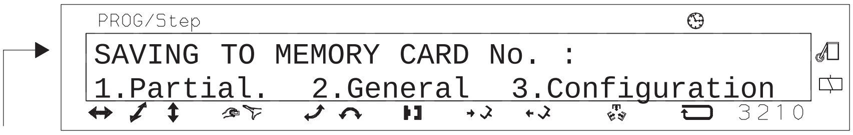
This function enables the following to be stored on a memory card :

- ▶ one or all of the programs with their options,
- ▶ the robot's configuration (values entered in configuration, context and parameter modes).

<u>Procedure for saving a program :</u>

- ▶ ▭ => COPY 9
- ▶ Install the memory card.
- ▶ Press TEACH 2 APPREN.

The robot displays :



```
PROG/Step
SAVING TO MEMORY CARD No. :
1.Partial.   2.General   3.Configuration
                                      3210
```

The robot displays the card number. See chapter on “Memory card”.

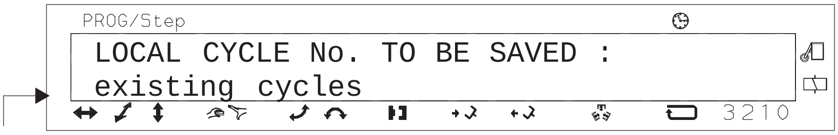
▶ Partial : for saving one program only.

▶ General : for saving all the programs and the configuration.

▶ Configuration : for saving the configuration.

▶ Press WRITE 1 DIRECT.

The robot displays :

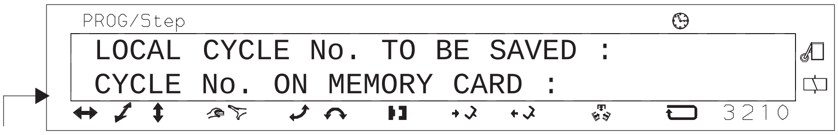


```
PROG/Step
LOCAL CYCLE No. TO BE SAVED :
existing cycles
                                      3210
```

The robot displays the cycles present in the local memory.

▶ Enter the number of the program to be saved then ↵.

The robot displays :



```
PROG/Step
LOCAL CYCLE No. TO BE SAVED :
CYCLE No. ON MEMORY CARD :
                                      3210
```

The number of a program stored on the card is always preceded by the card number.

Program 11 on card 8 will have the following number : 811.

▶ Enter the number that will be used for the program on the card, for example “11”, then ↵.

VI – 2. 4. Restoring a program

This function enables the following to be recalled from the memory card to the local memory :

▶ a program and its options,

▶ the robot’s configuration (values entered in configuration, context and parameter modes).

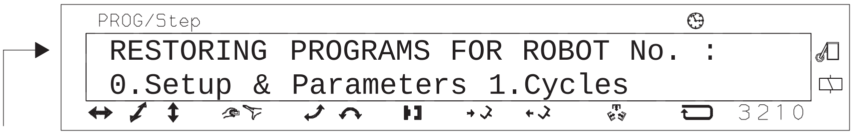
<u>Procedure for restoring a program :</u>

▶ => COPY 9

▶ Install the memory card.

▶ Press SETUP 3 CONFIG.

The robot displays :

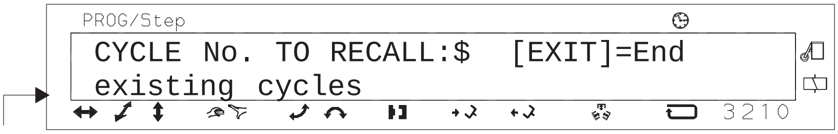


```
PROG/Step
RESTORING PROGRAMS FOR ROBOT No. :
0.Setup & Parameters 1.Cycles
3210
```

–> The robot displays the robot number saved in the memory.

- ▶ Setup & Parameters : to restore the configuration.
- ▶ Cycles : to restore a program.
- ▶ Press WRITE 1 DIRECT.

The robot displays :



```
PROG/Step
CYCLE No. TO RECALL:$  [EXIT]=End
existing cycles
3210
```

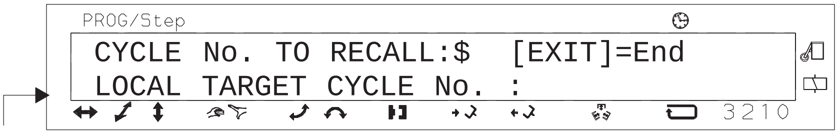
–> The robot displays the programs present on the card.

⚠ The number of a program stored on a card is always preceded by the number of the card.

Program 11 on card 8 will have the following number : 811.

- ▶ Enter the number of the program to be restored, for example “11”, then ↵.

The robot displays :



```
PROG/Step
CYCLE No. TO RECALL:$  [EXIT]=End
LOCAL TARGET CYCLE No. :
3210
```

- ▶ Enter the number that the program will have in the local memory then ↵.

– FIGURES –

– INDEX –

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:

PARTS & SERVICE 800 458 1960
Instant Access CONAIR™

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.