

USERGUIDE

S900-II

Programming - Level 2

Version 2.1



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This manual is out-of-date and is provided only for its technical information, data and capacities.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containing warnings, information, precautions, and procedures that may be more adequate than those contained in this out-of-date manual.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



Innovation or information
concerning a particular
software version

CONTENTS

I – MAIN PROGRAM INSTRUCTIONS	1
I – 1. The instruction	1
I – 1. 1. The operands	1
I – 1. 2. The operators	1
I – 2. Allocation, test and counting instructions	2
I – 2. 1. SET instruction – Allocation –	2
I – 2. 2. RESET instruction – Setting to zero –	3
I – 2. 3. IF instruction – Conditions and testing –	3
I – 3. The bits – BIT –	4
I – 3. 1. System bits accessible in read only	4
I – 3. 2. System bits accessible in read and write	4
I – 3. 3. User bits accessible in read and write	4
I – 4. 16 bits words – WRD –	5
I – 4. 1. List of WORDs	5
I – 4. 2. 16 bit system words – WRD 32 -> 63	6
I – 4. 3. Details of the 16 bit system words	7
I – 5. The 32 bit words – WWRD –	18
I – 5. 1. The 32 bit user words	18
I – 5. 2. The 32 bit system words WWRD 56 -> 127	18
I – 5. 3. Details of the 32 bit system words	19
I – 6. CNT instruction – Counter handling –	20
I – 6. 1. Standard counters – CNT 00 to 15 –	21
I – 6. 2. Stacking counters – CNT 0041 to 9980 –	21
I – 7. The time delays – TIME –	22
I – 8. Summary of instructions	23
I – 9. Motorized movement codes	24
I – 9. 1. Movement code	24
I – 9. 2. Operand	25
I – 10. Preparatory functions "FUNC" of the numeric axes	26
I – 10. 1. VEL : Speed (Maintained)	26
I – 10. 2. SLA : Slow approach (Temporary)	27
II – REGULAR STACKING SUBROUTINES	28
III – GENERAL STACKING SUBROUTINES	35
III – 1. Editing a subroutine – SP61 to 80 –	35
III – 2. Teaching the general stacking sequences in adjust mode	36
III – 2. 1. Entering the parameters	36
III – 2. 2. Teaching the gaps	36
III – 2. 3. Creating the program	36
III – 3. Entering a general stacking subroutine	37
III – 4. Description and use of the rotation word	38

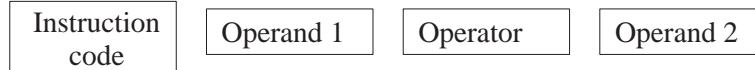
III – 5. Example 1	40
III – 6. Example 2 – Alternate layers	41
IV – PARALLEL SUBROUTINES – SP81 TO 99 –	42
V – BUILT IN PLC	44
V – 1. General information	44
V – 2. The instructions	45
V – 2. 1. The boolean operands	45
V – 2. 2. Structured operands	46
V – 2. 3. The instruction codes	48
V – 3. Application examples	55
V – 3. 1. Combination (relay drawings)	55
V – 3. 2. Sequential (Grafcet drawing)	58
V – 4. Monitoring PLC : PLC 99	62
INDEX	63

This manual is a complement to the Programming Level 1 manual.

I – MAIN PROGRAM INSTRUCTIONS

I – 1. The instruction

Syntax :



I – 1. 1. The operands

Name	Mnemonic	Keys	Number	Functions
Output	OUT *		000 -> 255	Boolean image of an action (output) to be performed externally (transmitted by the local or remote Output boards). Boolean image of an information (input) coming from outside (received by the local or remote Input boards)
Input	IN *		000 -> 255	
Counter	CNT	SHIFT	00 -> 15	Structures reserved for increments and decrements.
Bit	BIT *	SHIFT	0041 -> 9980 000 -> 127	
Timer	TIM	SHIFT	000 -> 15	Internal boolean variable resulting from the PLC timer.
Word	WRD	SHIFT	000 -> 4095	General data in 16 bit memory.
Double Word	WWRD	SHIFT	000 -> 127	General data in 32 bit memory.

* *The mnemonic code of these structures is also an instruction code.*

I – 1. 2. The operators

These operators are required to perform allocations, operations or tests in SET and IF type instructions (see following paragraph). They cannot be used alone as instructions in the main program.

The evaluation result is positioned in operand 1. Operand 2 can be a constant :

- ▶ decimal, Operand 2_ **D**
- ▶ hexadecimal, Operand 2_ **H**

▪ Arithmetic operators (only in SET instruction)

=	Allocation of a value	} <i>Capacity overrun in WORD or WWORD is not tested.</i>
+	Addition	
-	Subtraction	
*	Multiplication	
/	Whole number division (remainder not kept)	

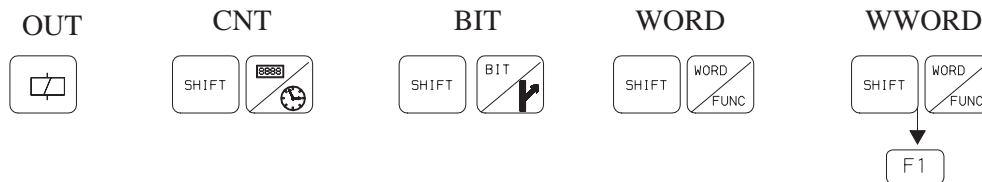
▪ Logic operators

=	2 operands are equal	} <i>Comparison (IF instruction)</i>
>=	Greater than or equal to	
<=	Less than or equal to	
AND	logic AND	} <i>For masking</i>
OR	logic OR	

I – 2. Allocation, test and counting instructions

I – 2. 1. SET instruction – Allocation –

The following variables can be set :



Addition : . **SET.WRD 0000 + 0012_D**

The contents of WRD 0000 is increased by 12 (decimal)

Division : . **SET.WWRD 005 / 03_D**

Before, WWRD 5 = 1200 (decimal).

After, WWRD 5 = 400 (decimal).

Multiplication : . **SET.WWRD 120 * A_H**

Before, WWRD 120 = 42 (decimal).

After, WWRD 120 = 420 (decimal).

Exclusive mask : . **SET.WWRD 0002 AND WWRD 005**

Before, WWRD 002 = 8AB74B7A (Hexadecimal)

WWRD 005 = 00008888 (Hexadecimal)

After, WWRD 002 = 00000808 (Hexadecimal)

Inclusive mask : . **SET.CNT 0010 OR WRD 0000**

Before, CNT 0010 = 0001 0000 0000 1111 = 000F (Hexa)

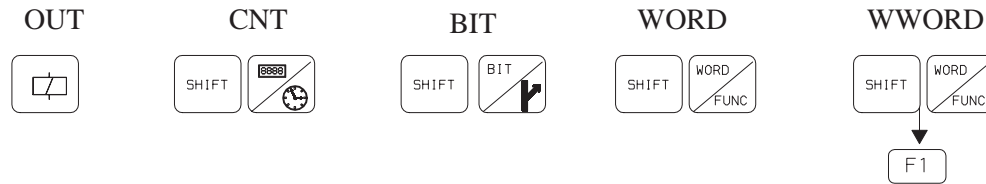
WRD 0000 = 0000 0001 0000 1111 = 010F (Hexa)

After, CNT 0010 = 0001 0001 0000 1111 = 010F (Hexa)

Reset to zero : . **SET.CNT 0141 = 0000_D**

I – 2. 2. RESET instruction – Setting to zero –

The following variables can be set :



. RST.WRD 0000 -> the contents of Word 0000 equals 0.

. The following can also be written : SET.WRD 0000 = 0000_D

I – 2. 3. IF instruction – Conditions and testing –

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

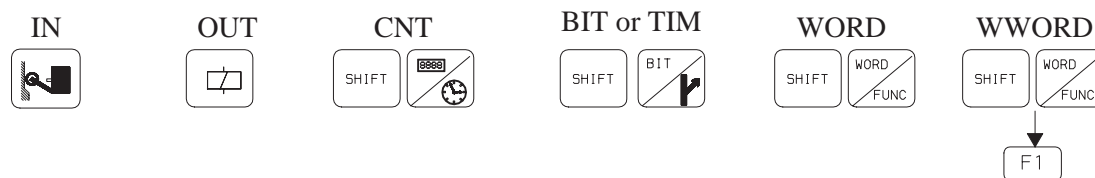
After an IF, all the instructions are valid APART FROM : L, R, MASTER and SLA.

You are offered two choices, once you have selected a variable :

- **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 the 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.

The following variables can be checked :



. IF CNT 07 \leq 5
if CNT 07 = 2
so if the condition is true, the following instruction will be executed.

. IF CNT 0008 \geq 02300_D
SP 61 L 15

If the contents of counter 0008 is greater than or equal to 2300 in decimal then the condition is confirmed and the following SP 61 will be executed.

. IF / WWRD 012 = 000B_H
OUT 012

If the contents of WWRD 0012 = B_H (11 in decimal)
then the condition is confirmed but the opposite of IF implies that OUT 012 will not be executed.

I – 3. The bits – BIT –

I – 3. 1. System bits accessible in read only

BIT NUMBER	FUNCTION
0	Reverse execution of stacking sequences
1	End of layer or stack
2	End of pallet
3	Odd row in progress
4	Odd column in progress
5	Odd layer in progress
6	Odd part number in progress
7	Stacking SP running
8	Reserved
9	TOTAL home return running
10	Parallel subroutine running
11	End of cycle triggered by Without Robot mode
12	Commands by external inputs
13	Reserved
14	Following axis attached
15	Reserved
16	Bit 0 of rotation word of general stackings.
17	Bit 1 of rotation word of general stackings.
18	Bit 2 of rotation word of general stackings.
19	Bit 3 of rotation word of general stackings.
20	Bit 4 of rotation word of general stackings.
21	Bit 5 of rotation word of general stackings.
22	Bit 6 of rotation word of general stackings.
23	Bit 7 of rotation word of general stackings.
24 -> 31	Reserved

I – 3. 2. System bits accessible in read and write

32	Bit resulting from part grips (replaces E_PP)
33	Bit validating automatic program change

I – 3. 3. User bits accessible in read and write

34 -> 127	User bits accessible in read and write by the main program PRG or PLC
-----------	---

I – 4. 16 bits words – WRD –

I – 4. 1. List of WORDs

WORD NUMBER	FUNCTION
WRD 0 -> WRD 31	32 user Words with no predefined functions, accessible in read and write.
WRD 32 -> WRD 63	32 system Words described in the following chapter.
WRD 64 -> WRD 79	16 Words supporting the PLC timers (TIM 00 to TIM 15).
WRD 80 -> WRD 95	16 Words supporting the standard counters (CNT 00 to CNT 15).
WRD 96 -> WRD 4095	4000 Words supporting the stacking subroutine counters (CNT 0041 to CNT 9980).

I – 4. 2. 16 bit system words – WRD 32 ->63

▶ Accessible in read only

WORD number	Name	Abbreviation	Function
32	MotMode	MD	Robot operating mode (Adjust, Stop, Step by Step or Automatic).
33	MotArrêt	AR	Type of stop (Stop or Safety).
34	MotStat1	E1	Mode running.
35	MotStat2	E2	System status.
36	MotStat3	E3	System status.
37 *	MotSec0	S0	Movements locked by the monitoring PLC.
38	MotSec1	S1	Safety devices' status.
39	MotSec2	S2	Safety devices' status.
40	MotSec3	S3	Safety devices' status.
41	MotExe1	X1	Execution being requested.
42	MotExe2	X2	Execution in progress.
43	MotDial1	M1	Inputs needed for dialogue with IMM 1.
44	MotDial2	M2	Inputs needed for dialogue with IMM 2.
45 *	MotKlax	XL	Pendant alarm.
46 *	MotRegl	RG	Movement orders in adjust mode.
47	ProgExec	PP	Number of program (PRG) running.
48	Sp_Exec	SP	Number of subroutine (SP or SR) running.
49	Plc_Exec	PL	PLC number running.
50	Pas_SPP	SL	Number of active step in parallel subroutine (SPP) running.
51	Pas_Exec	PA	Number of step running (in the PRG, SP or SR).
52	Version	VE	Type and version of Sepro software.
53	Options	CG	Options installed on the robot.
54	Cfg_I_O	IO	Number of robot Input/Output and Axes boards.
55	NumRob	SR	Robot serial number.
56	TypRob	TY	Robot type.
57	MotCod2	C2	not used.

* WRDs 37, 45 and 46 are accessible in read and write.

► Accessible in read and write

WORD number	Name	Abbreviation	Function
58	MotCod1	C1	Number of program to be executed (requested by encoding or JBUS).
59	MotDem1	D1	Action requests such as START, Stop at end of cycle,...
60	MotDem2	D2	Action requests continued.
61	DemMod	DM	Mode requests (Adjust, Automatic...).
62	Défaut	DF	Number of fault running or to be taken into account.
63	PiècFab	MP	Part made memory.

I – 4. 3. Details of the 16 bit system words

You can use the instructions IF and SET ... WRD (in 1st operand).

The values marked on the first line are in decimal, those on the second line in hexadecimal. They correspond to the weight of the significant bit.

– WRD 32 – MotMode (MD)

Robot operating mode

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
		ROBOT–OFF	EN–PROGR	EN–REGL	EN–SEMI	EN–AUTO	EN–ARRET

EN–ARRET : In Stop mode.

EN–AUTO : In Automatic mode.

EN–SEMI : In Step by Step mode.

EN–REGL : In Adjust mode.

EN–PROGR : In Programming mode.

ROBOT–OFF : In Robot Off mode

– WRD 33 – MotArret (AR)

Used for unconditional stop requests.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
						SECU	STOP

STOP : General request for an immediate stop. (At 1 for example if the DPCY button is released in SEMI–AUTO mode whilst the step is being executed).

SECU : General request for a safety stop (E.g. : no power).

I – Main program instructions

– WRD 34 – MotSta1 (E1)

Indicates the actual mode being used.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
		ROBOT- OFF	EN- PROGR	EN-REGL	EN-SEMI	EN-AUTO	EN- ARRET

EN-ARRET : Robot in stop mode (Auto, Semi-Auto, Adjust = 0)

EN-AUTO : Same for AUTO.

EN-SEMI : Same for SEMI-AUTO.

EN-REGL : Same for ADJUST.

EN-PROGR : In Programming mode.

ROBOT-OFF : In Robot Off mode.

Note : Even though it reflects the MotMode word, this word is updated by operating modes, taking into account the actual mode being used.



Step by Step mode selected but robot is carrying out a Home Return or a Tool Change Position which have special operating modes.

– WRD 35 – MotSta2 (E2)

Set and Reset using "Operating modes".

Each bit of this word is set to 1 after the procedure start request has been reset or the task (where used) has been launched.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
EN-DEF	EN-WARN	WWRD63	-	EN- ETEND	EN-VISU	EN-PROG	EN-EXEC

32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_ H	800_H	400_H	200_H	100_H
PROG-SUS	APPR- REGL		EN- TEST	EN-HIST	EN-PROD	EN- MAINT	EN-HELP

EN-EXEC : Indicator for operating mode being used.

EN-PROG : Indicator for programming mode being used.

EN-VISU : Indicator for I/O display monitor being used.

EN-ETEND : Indicator for extended monitor mode being used.

WWRD63 : Indicator for entering delay value for current anticipated restart.

EN-WARN : Indicator for warning being used.

EN-DEF : Indicator for robot fault.

EN-HELP : Indicator for help mode (fault commentaries).

EN-MAINT : Indicator for maintenance work in progress.

- EN-PROD : Indicator for production work.
- EN-HIST : Indicator for fault historic work.
- EN-TEST : Indicator for input or waiting bit test function.
- APPR-REGL : Indicator for teaching work in adjust mode running.
- PROG-SUS : Indicator for programming mode momentarily suspended.

– WRD 36 – MotSta3 (E3)

Set and Reset using "Operating modes".

Each bit of this word is set to 1 after the procedure start request has been reset, and remains at 0 until the requested procedure is completed.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
	RO-AUTO		RO-TOTAL	RO-SIM- PLE	EN-PCO	EN-ARFCY	EN-CYCLE
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
							EN-SUSP

- EN-CYCLE : Indicator for robot in cycle (once START has been pressed).
- EN-ARFCY : Indicator for end of auto cycle running.
- EN-PCO : Indicator for PCO procedure running.
- RO-SIMPL : Indicator for simple HR running.
- RO-TOTAL : Indicator for total home reutrn running.
- RO-AUTO : Indicator for automatic simple return (part grip fault in the mould).
- EN-SUSP : Indicator for robot cycle suspended.



– WRD 37 – MotSec0 (S0)

Each bit of this WORD will enable you to validate (or not) the robot's standard movements. They are all reset to 1 when the robot is powered up.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
ACT 4	ACT 3	ACT 2	AXE C	AXE B	AXE Z	AXE Y	AXE X
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
ACT 16	ACT 14	ACT 13	ACT 10	ACT 9	ACT 8	ACT 7	ACT 5

See example for the use of WORD 37 page 62.

I – Main program instructions

– WRD 38 – MotSec1 (S1)

Contains the status of the inputs used for the safeties.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
VAR_C_OK	PUISS_OK	BAUR_OK	SURC_X	SURC_Y	SURC_Z	SURC_B	SURC_C
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
CTL_PCO_1	CTL_PCO_2	CTL_SBD_1	CTL_SBD_2	VAR_X_OK	VAR_Y_OK	VAR_Z_OK	VAR_B_OK

SURC_? : Status of the electric overtravels of the numeric axes.

BAUR_OK : Status of the Emergency Stop Buttons.

PUISS_OK : Information that robot is powered up (at 0 if not powered up).

VAR_?_OK : Status of Speed Driver OK information (input).

CTL_SBD_? : Status of the SBD relay control.

CTL_PCO_? : Status of the PCO End of Stroke control.

– WRD 39 – MotSec2 (S2)

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
BRAS2_HM	BRAS1_HAUT	BRAS2_HAUT	SECU_ROT1	SECU_ROT2	PROTECT	HOMME_OK	TRANSIST
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
SUSPEND	CTL_MAINT	MODE_MAINT	PRESSION	CTP_FDL	ZBD	LATERAL	BRAS1_HM

TRANSIST : Status of the Z brake transistor control.

HOMME_OK : Status of the Dead Man push-buttons.

PROTECT : Status of the operator protections.

SECU_ROT? : Safety for Horizontal/Vertical rotation.

BRAS?_HAUT: Status of the Arm Up informations.

BRAS?_HM : Status of the Arm Outside Mould informations.

LATERAL : Status of the side-entry selection input.

ZBD : Status of the robot on ZBD cam information.

CTP_FDL : Status of the CTP_FDL information (changeable parallelogram).

PRESSION : Status of the air pressure OK information.

MOD_MAINT : Status of the maintenance mode selected information.

CTL_MAINT : Status of the maintenance mode control information.

SUSPEND : Status of the cycle suspension information.

– WRD 40 – MotSec3 (S3)

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
GV_ACT8	GV_ACT7	–	GV_ACT5	GV_ACT4	GV_ACT3	GV_ACT2	DEF_ROT
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
VCM_OK	–	–	–	–	–	–	–

- DEF_ROT : Rotation 1 test OK.
- GV_ACT2 : High speed pneumatic Ascent 1 enabled
- GV_ACT3 : High speed pneumatic Ascent 2 enabled
- GV_ACT4 : High speed pneumatic Descent 1 enabled
- GV_ACT5 : High speed pneumatic Descent 2 enabled
- GV_ACT7 : High speed pneumatic arm Advance enabled
- GV_ACT8 : High speed pneumatic arm Retreat enabled
- VCM_OK : Reflection of the Arm Free Safety (SBD) that authorises the Machine Cycle Validation (VCM)

V 1.3 Robot
V 0.5 PC Editor

– WRD 41 – MotExe1 (X1)

Transmission of the executive program data to the operating mode management.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
APPR-OK	RO-AUTO	–	RO-TOTAL	RO-SIMPL	DEM-PCO	DEM-AFCY	DEM-START

- DEM-START : Request for start of robot cycle execution. (In Step by Step mode, this request is repeated if the START key or START input is validated – Release and press).
- DEM-AFCY : Request for robot stop at end of cycle.
- DEM-PCO : Request for execution of the PCO procedure.
- RO-SIMPL : Request for a simple home return.
- RO-TOTAL : Request for a total home return.
- RO-AUTO : Request for an automatic simple return.
- APPR-OK : Request for information showing that the teaching requested has been recorded and execution can continue.

– 42 – MotExe2 (X2)

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
FIN2PAS	–	APPRGT	APPRENT	FIN_PCO	FIN_RO	FIN_PAS	FCYCLE

- FCYCLE : Return showing that the cycle is finished. (Reply given after end of cycle stop request).
- FIN_PAS : Return showing the end of step execution. Used in Step by Step mode.

I – Main program instructions

- FIN_RO : Return showing the end of execution of the home return requested (simple return, total, point or auto).
- FIN_PCO : Same for PCO.
- APPRENT : Request for move to teaching mode.
- APPRGT : Request for move to teaching mode in the GEN STK header.
- FIN2PAS : Return showing the end of SPP step execution. Used in Step by Step mode.

– WRD 43 – MotDial1 (M1)

– WRD 44 – MotDial2 (M2)

Status of data required for dialogue with IMM 1 and 2.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
FIN_NOY1	EJECT_OUT	EJECT_IN	MACH_OUV	MOUL_FERM	PORTE_FERM	OUV_PART	AUT_SAUT
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
VCM_OK	HORS_MACH	AXE_AM_Y	AXE_AM_X	PAS_DECAL	VAL_COD	PRES_PP	FIN_NOY2

- AUT_SAUT : IMM in Auto or Semi–Auto.
- OUV_PART : Partial opening reached.
- PORTE_FERM: IMM gate closed.
- MOUL_FERM : Mould closed.
- MACH_OUV : Mould open.
- EJECT_IN : Ejector in.
- EJECT_OUT : Ejector out.
- FIN_NOY1 : End of core puller movement 1.
- FIN_NOY2 : End of core puller movement 2.
- PRESS_PP : Part gripped by the robot.
- VAL_COD : Program code validation in automatic program change.
- PAS_DECAL : Delayed wait not valid.
- AXE_AM_X : X machine axis.
- AXE_AM_Y : Y machine axis.
- HORS_MACH : Robot outside machine.
- VCM_OK : Conditions for VCM OK.

– WRD 45 – MotKlax (KL)

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
						KLAX–ON	KLAX–OFF

- KLAX–OFF : Set to 1 to stop the audible alarm.
- KLAX–ON : Set to 1 to activate the audible alarm.

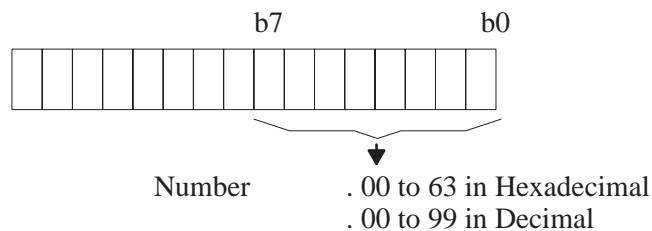
– WRD 46 – MotRegl (RG)

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
DEPL_ZM	DEPL_ZP	DEPL_YM	DEPL_YP	DEPL_XM	DEPL_XP	DEC_VIT	INC_VIT
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
–	–	–	–	DEPL_CM	DEPL_CP	DEPL_BM	DEPL_BP

- INC_VIT : Reflects the “Hare” key or the speed increase input.
- DEC_VIT : Reflects the “Tortoise” key or the speed decrease input.
- DEP_XP : Reflects the X+ axis movement key or input.
- DEP_XM : Same for X-.
- DEP_YP : Same for Y+.
- DEP_YM : Same for Y-.
- DEP_ZP : Same for Z+.
- DEP_ZM : Same for Z-.
- DEP_BP : Same for B+.
- DEP_BM : Same for B-.
- DEP_CP : Same for C+.
- DEP_CM : Same for C-.

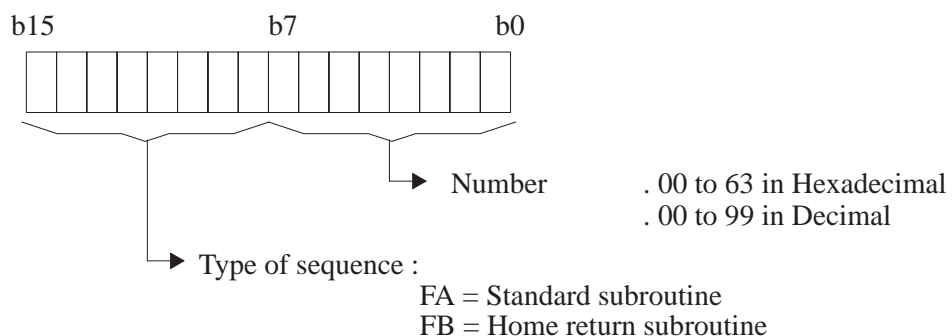
– WRD 47 – ProgExec (PP)

Indicates the number of the main program running.



– WRD 48 – SP_Exec (SP)

Gives the code of the SP or SR running.



– **WRD 53 – Config (CG)**

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
JBUS	ANTICIP	2_IMP	TAIL_ME M	E17	PC	TAIL_MO D	MOD
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
PAS_DIAL		EXT_ME M				TELE- MAINT	M_MODU LE

- MOD : Module present (0=present, 1=absent).
 TAIL_MOD : Size of memory module (0=16Kb, 1=64Kb).
 PC : PC option present (0=no, 1=yes).
 E17 : Euromap 17 option present (0=no, 1=yes).
 TAIL_MEM : Size of internal memory (0=32Kb, 1=128Kb).
 2_IMP : Second printer option (0=absent, 1=present).
 ANTICIP : Anticipated restart option (0=absent, 1=present).
 JBUS : JBUS option (0=absent, 1=present).
 M_MODULE : M_MODULE board present.
 TELEMAINT : Reserved.
 EXT_MEM : Memory extension board (512Kb) board present.
 PAS_DIAL : No DIAL board present.

– **WRD 54 – Cfg_I_O (IO)**

- Configuration of the Inputs/Outputs installed.
 b0 to b3 = Number of 16 output group installed.
 b4 to b7 = Number of 16 input group installed.
 b8 = 1 -> Axis board 1 present
 b9 = 1 -> Axis board 2 present

– **WRD 55 – MotSeri (SR)**

- Serial number (robot fabrication) in hexadecimal.
 Number 4000 (in decimal) -> 0FA0 (in hexadecimal).



– **WRD 56 – MotType (TY) encoded in hexadecimal**

b3	b2	b1	b0	Type
0	0	0	0	BX
0	0	0	1	BY
0	0	1	0	BZ
0	0	1	1	BB
0	1	0	0	BC
0	1	0	1	AX
0	1	1	0	AY
0	1	1	1	AZ
				Reserved

b4 to b15 -> Models 3020, 3021, 3030, ...



3020 BZ -> 30202 (in decimal) -> 75FA (in hexadecimal)

– **WRD 57 – MotCod2 (C2)**

Reserved.

– **WRD 58 – MotCod1 (C1)**

Encoding of requested program number.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
ssp	ss64	ss32	ss16	ss8	ss4	ss2	ss1

ss1, ss2, ss4, ss8, ss16, ss32 : Binary encoding of program number.

ssP : Parity bit (there should always be an even number of bits set to 1).

Note : In “short” encoding, only bits ss1 to ss8 and ssP are used. Bit ssP is not required if the word is written by the Host.

– **WRD 59 – MotDem1 (D1)**

Requests to operating modes manager.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
DEM-ARRET	DEM-MONIT	DEM-PROG	DEM-RO	PLU-DPCY	DEM-PCO	DEM-AFCY	DEM-DPCY
32768	16384	8192	4096	2048	1024	512	256
8000_H	4000_H	2000_H	1000_H	800_H	400_H	200_H	100_H
-	DEM-EDIT-EXT	DEM-EDIT	DEM-TEST	DEM-HIST	DEM-PROD	DEM-MAINT	DEM2DPCY

DEM-DPCY : Cycle Start request.

DEM-AFCY : Same for stop at end of cycle request.

- DEM-PCO : Same for move to PCO mode request.
 PLU-DPCY : Indicator for release of Cycle Start key.
 DEM-RO : Request for HR procedure execution.
 DEM-PROG : Request for Programming procedure execution.
 DEM-MONIT : Request for Monitor (Visu) procedure execution.
 DEM-ARRET : Request for stop at end of step.
 DEM2DPCY : Cycle Start request (for // SP).
 DEM-MAINT : Same for maintenance management request.
 DEM-PROD : Same for production management request.
 DEM-HIST : Same for faults historic request.
 DEM-TEST : Same for test request.
 DEM-EDIT : Same for program editing request.
 DEM-EDIT-EXT: Same for edition in extended monitor request.

V 2.0 Robot

– **WRD 60 – MotDem2 (D2)**

Indicates a home return to be performed in relation to the DEM-RO request of the previous word for external input commands.

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
DEM_RO_ TOTAL	DEM_RO_ SIMPL	DEM_SS_ ROB2	DEM_SS_ ROB1	-	DEM_RST 2	DEM_RST 1	DEM_STA RT

- DEM_START : Press START BRP 1 or 2.
 DEM_RST? : BRP Reset request.
 DEM_SS_ROB?: Request for Without Robot mode on BRP.
 DEM_RO_SIMPL: Simple HR request by the inputs.
 DEM_RO_TOTAL: Total HR request by the inputs.

Note : Word 60 "MotDem2" must be positionned before the HR is requested by Word 59 "MotDem1".

– **WRD 61 – DemMod (DM)**

Reserved.

– **WRD 62 – Défait (DF)**

Number of the fault running or the one you wish to provoke (in decimal).

Note : Displays the message associated with the fault.

V 1.3 Robot

V 0.5 PC Editor

– **WRD 63 – PiècFab (MP)**

WRD 63 = FF if the Part Made Memory (IMM 1) is at 1.

I – 5. The 32 bit words – WWRD –

I – 5. 1. The 32 bit user words

▶ Accessible in read and write

Number (logic address)	Structures / Functions
WWRD 0 -> WWRD 55	56 user words with no predefined function.

I – 5. 2. The 32 bit system words WWRD 56 -> 127

The WWRDs not described are reserved.

* **type of access** : R = read only

R/W = read and write

Number (logic address)	Abbreviation	Structures / Functions	*
		Duration of high speed pulses (in milliseconds)	
WWRD 56	–	Pneumatic ascent 1.	R/W
WWRD 57	–	Pneumatic ascent 2.	R/W
WWRD 58	–	Pneumatic descent 1.	R/W
WWRD 59	–	Pneumatic descent 2.	R/W
WWRD 60	–	Pneumatic arm advance.	R/W
WWRD 61	–	Pneumatic arm retreat.	R/W
WWRD 63	–	Value of programmed delay for anticipated restart (in tenths of seconds).	R/W
WWRD 64	AC	Mould position (Euromap 17).	R
WWRD 65	AE	Ejector position (Euromap 17).	R
WWRD 68	NP	Number of parts to be made.	R/W
WWRD 76	NG	Number of good parts treated by the robot.	R/W
WWRD 77	PX	Current position of the X axis in 1/10 mm or 1/10°.	R
WWRD 78	PY	Current position of the Y axis in 1/10 mm or 1/10°.	R
WWRD 79	PZ	Current position of the Z axis in 1/10 mm or 1/10°.	R
WWRD 81	PB	Current position of the B axis in 1/10 mm or 1/10°.	R
WWRD 82	PC	Current position of the C axis in 1/10 mm or 1/10°.	R
WWRD 83	PD	Current position of the N1 axis in 1/10 mm.	R
WWRD 84	PE	Current position of the N2 axis in 1/10 mm.	R
WWRD 85	PF	Current position of the A1 axis in 1/10 mm.	R
WWRD 86	PG	Current position of the A2 axis in 1/10 mm.	R
WWRD 87 -> WWRD 101		Reserved for Euromap 17.	R
WWRD 102	AN	Number of program to be executed (Euromap 17).	R
WWRD 103	SW	Exchange word for Euromap 17.	R

v 1.3 Robot

v 0.5 PC Editor

v 2.0 Robot

WWRD 104	MW	Exchange word for Euomap 17.	R
WWRD 106	KV	Current value of the general speed coefficient (Kv).	R
WWRD 116	–	Robot freeing time (in 1/10 s) when functioning with the IMM cycle anticipated restart.	R
WWRD 117	–	Time for IMM movement to start (in 1/10 s) when functioning with anticipated restart.	R
WWRD 120	–	Code of the last customized key pressed. (see encoding in chapter I – 5. 3.).	R
WWRD 121	–	Type of unloading and second arm.	R

The WWORDS used for Euomap 17 are described in the “Using Euomap 17 on Sepro S900–II robots” manual.

I – 5. 3. Details of the 32 bit system words

You can use the instructions IF and SET ... WWRD (in 1st operand).

The values marked on the first line are in decimal, those on the second line in hexadecimal. They correspond to the weight of the significant bit.

– WWRD 120 – Customised key code

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
TOUCH_Z	TOUCH_Y	TOUCH_X	TOUCH_W	TOUCH_P	TOUCH_O	TOUCH_H	TOUCH_G

– WWRD 121 – Type of unloading and second arm

128	64	32	16	8	4	2	1
80_H	40_H	20_H	10_H	8_H	4_H	2_H	1_H
–	–	TYP_BB	TYP_BA	MIXTE	VERT- AXIAL	LATERAL	VERTI- CAL

Type of access to IMM 1: (copy of parameter 170)

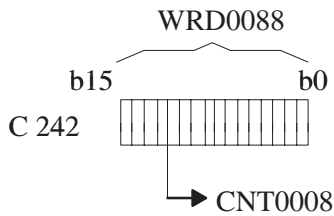
Vertical, Side–entry, Vertical – Axial or Mixed (Vertical–Side–entry)

Type of second arm: (encoding of parameter 37)

TYP_BB	TYP_BA	Type of second arm
0	0	No second arm
0	1	Electrical arm
1	0	Pneumatic arm with stop
1	1	Pneumatic tandem arm

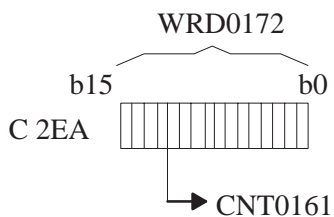
I – 6. 1. Standard counters – CNT 00 to 15 –

▶ Accessible in read and write

Number (logic address)	Indirection or logic address (on a Word)	Structures / Functions
CNT00	WRD 0080	 <p>. from 0000 to 9999 in decimal . from 0000 to FFFF in hexadecimal</p>
CNT01	WRD 0081	
CNT02	WRD 0082	
CNT03	WRD 0083	
CNT04	WRD 0084	
CNT05	WRD 0085	
CNT06	WRD 0086	
CNT07	WRD 0087	
CNT08	WRD 0088	
CNT09	WRD 0089	
CNT10	WRD 0090	
CNT11	WRD 0091	
CNT12	WRD 0092	
CNT13	WRD 0093	
CNT14	WRD 0094	
CNT15	WRD 0095	

I – 6. 2. Stacking counters – CNT 0041 to 9980 –

▶ Accessible in read and write

Number (logic address)	Indirection or logic address (on a Word)	Structures / Functions
CNT 0041 -> CNT 9980 See table for valid numbers.	WRD 0096 -> WRD 4095 The corresponding 16 bits words and the addresses are continuous. There are no gaps in the numbering : : CNT0080 = WRD0135 -> C2A0 CNT0141 = WRD0136 -> C2A2 : CNT0180 = WRD0175 -> C2F0 CNT0241 = WRD0176 -> C2F2 :	 <p>. from 0000 to 9999 (in decimal) in the general subroutine No. 61 of the main program No. 01.</p>

I – 7. The time delays – TIME –

Apart from a numeric value between 001 and 99 in 1/10s, you can also give the time delay the following value :

WORD : Word No. 00 to 15 containing a numerical value between 001 and 999 (in decimal).

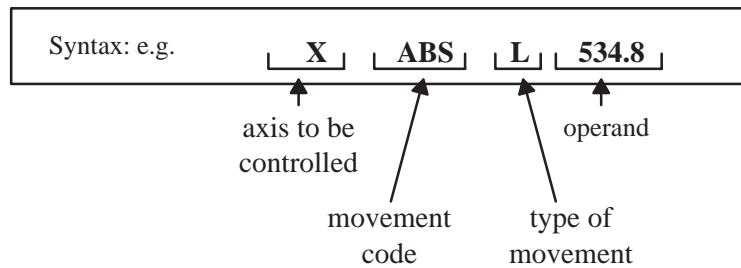


. Numeric value : TIME 0005 – time delay of 5 tenths of a second.

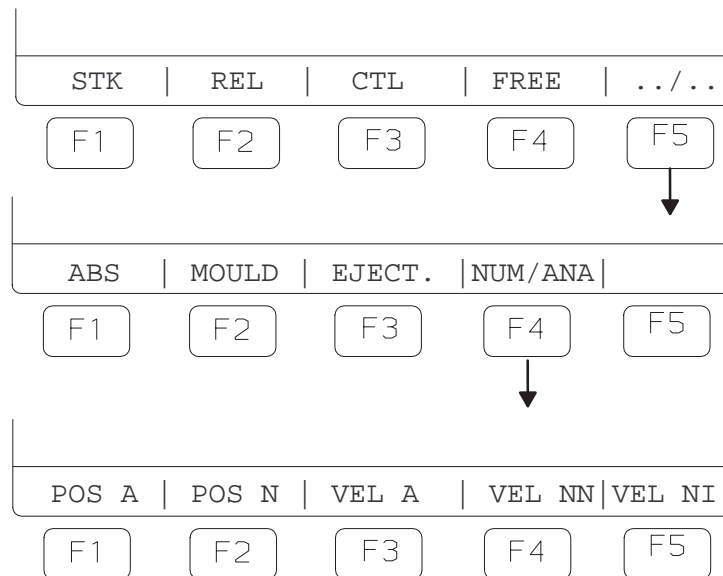
. Word : Step n SET WRD 0005 = 0010_D
 Step k TIME WRD 0005 – time delay of 10 tenths of a second.

I – 9. Motorized movement codes

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

**I – 9. 1. Movement code**

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



Details of these movement codes are given in the Programming S900–II Level 1 Manual.

► **STK : Stacking movement**

Use of this stacking movement is described in chapters II – page 28 and III – page 35.

► **MOULD / EJECT. : Movements associated with the Euromap 17 option**

Use of these instructions is described in the “Using Euromap 17” manual.

► **POSA / POSN / VELA / VELNN : Servo–controlled movements**

Use of these instructions is described in the “Mould chasing” manual.

I – 9. 2. Operand

After selecting the movement code, the operand can be given a numerical value (in 1/10 mm or in 1/10 degrees). One of the following functions can be used :



WWord : 32 bit Word

Teach : Teaching (described in the Programming S900–II Level 1 manual).

Offset : Use of this function is described in the “Mould chasing” manual.

ValStk : Position of the first part in a general stacking sequence (see chapter III – page 35).

▶ Linear movement

X.ABS L WW 00 (to 55)

X.STK L WW 00 (to 55)

X.REL L WW 00 (to 55)

X.CTL L WW 00 (to 55)

▶ Rotating movement

B.ABS R WW 00 (to 55)

B.REL R WW 00 (to 55)

B.CTL R WW 00 (to 55)

▶ Actuating movements of the B and C axes in stacking

B.STK value in degrees

C.STK value in degrees

(Codes only validated in the general stacking subroutines, SP 61 to 80).

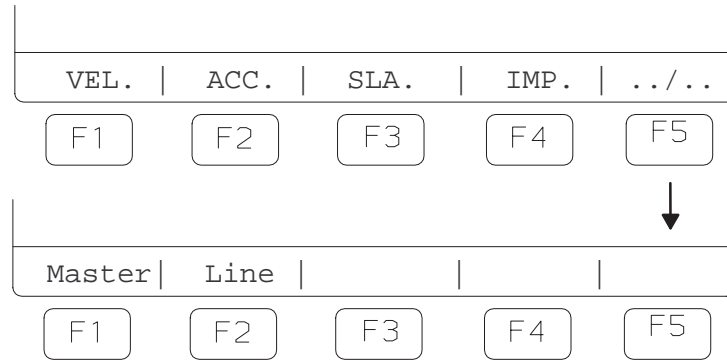
▶ Examples of movement programs

X.ABS L WW 06	Absolute movement of X in relation to double word No. 6
Y.STK L WW 02	Stacking movement in relation to double word No. 2.

I – 10. Preparatory functions "FUNC" of the numeric axes

These are preparatory functions concerning the numeric movements. There are basically two types :

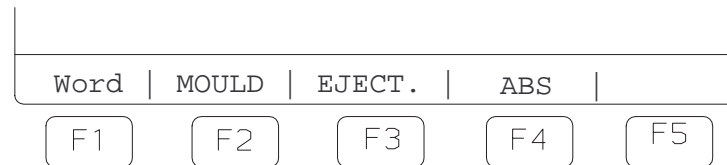
- ▶ Temporary effect functions (only valid for the current step)
- ▶ Maintained effect functions, maintained until a new function appears.



Note : The Acc, Imp, Master and Line instructions are described in the Programming Level 1 manual.

**I – 10. 1. VEL : Speed (Maintained)**

The VEL instruction can be programmed in 5 different ways :



- ▶ Axis' speed as a percentage of the maximum speed



VEL.X 50 means the X axis' speed passes to 50% of its maximum speed.

- ▶ Axis' speed in mm/s

- speed value in mm/s, programmed directly

The value of the speed must be between the minimum speed set in the parameters and the maximum speed allowed in adjust mode.



VEL.Y ABS 100.0 means the Y axis' speed passes to 100.0 mm/s.

- speed value in mm/s, programmed in a WWord

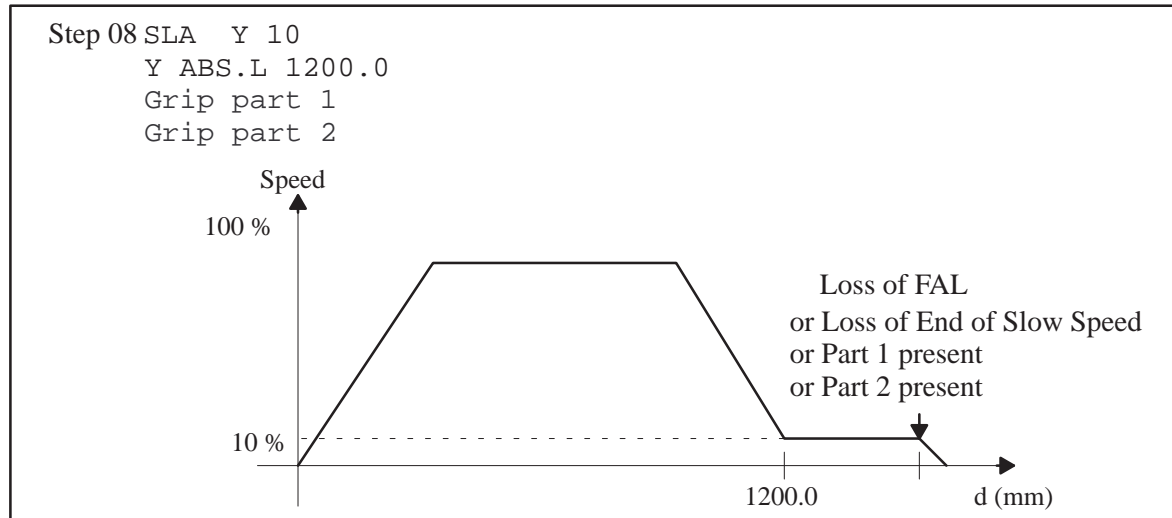
In this case, the speed must be less than the limit speed set in the parameters.



VEL.Z WW_00 means the Z axis' speed passes to the value contained in WWord 0, given in mm/s.

- ▶ The MOULE and EJECT speeds are reserved for Euromap 17 evolutions.

I – 10. 2. SLA : Slow approach (Temporary)



Setting the parameters :

- ▶ Loss of FAL : Parameter 443.
- ▶ Loss of End of Slow Speed : Parameter 629 for X, 765 for Y, 901 for Z, 1037 for B, 1173 for C.
- ▶ Part Presence : Parameter 389
- ▶ Part Presence : Parameter 407

In addition the so-called "slow" input, which causes a controlled reduction in speed when it disappears, is contained in the following parameters :

Parameters 628 for X, 764 for Y, 900 for Z, 1036 for B and 1172 for C.

II – REGULAR STACKING SUBROUTINES

The regular stacking subroutines are subroutines 41 to 60. They are used to simplify the organization of pallets or an existing column/layer organization.

They can be named (maximum 30 characters) in order to denote the palletization being done.



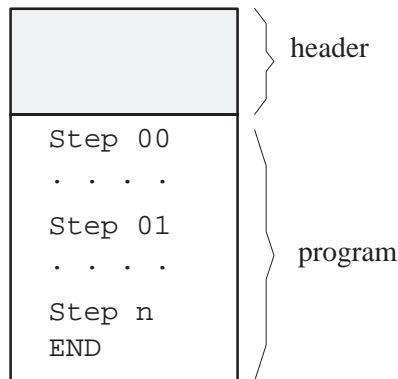
SP 42 Cover on pallet 2

Structure :

The stacking subroutines are comprised of 2 parts :

- ▶ A header : States the pallet organization.
- ▶ A program : Cycle describing the picking or the placing of the first part on the pallet.

SP41 to 60



Stacking counter :

A usable counter with a logic address that can be displayed is allocated to each stacking program.

CNT xxnn xx : Main program number

 nn : Number of subroutine concerned

The stacking counter is reset to 0 when a TOTAL home return is carried out. A simple home return has no effect on the stacking unless otherwise programmed.

The stacking counter is increased at each decoding of the stacking subroutine header. When it is equal to the number of parts to be stacked, the counter is set to 1 at the following decoding.

Header :

The header is completed using a dialogue box. This takes the form of a set of questions which vary according to whether the stacking is carried out with or without staggering.



```
Staggered no (0) or yes (1).....
By layers (0) or stacks (1).....
Stacking (0) or unstacking (1).....
Storage in XY (0) or YX (1).....

Number of parts in X row 1.....
* Number of parts in X row 2.....
Gap X between parts row 1.....
* Gap X between parts row 2.....
* Gap X between columns 1 / 2.....

Number of parts in Y column 1.....
* Number of parts in Y column 2.....
Gap Y between parts column 1.....
* Gap Y between parts column 2.....
* Gap Y between rows 1 / 2 .....

Number of layers.....
Gap between layers.....

Command at layer end or stack end.....
Command at pallet end.....

* These questions are only asked if the stacking includes staggering
```

Definitions :

- ▶ Row : Alignment of parts parallel to X-axis.
- ▶ Column : Alignment of parts parallel to Y-axis.
- ▶ Layer : Stacking of parts on Z-axis.

Part No. 1 is located in row 1, column 1, layer 1.

Command at end of layer, stack, pallet :

The possible commands are : OUT ___ (from 0 to 127)

BIT ___ (from 34 to 127)

The command is activated by the system if the following conditions are satisfied :

- ▶ The stacking counter has as a value the number of parts in a layer or stack (end of layer or stack) or the number of parts on the pallet (end of pallet).
- ▶ The system decodes the relevant stacking subroutine header.

The command activated by the system is set to 1. The following commands are then available :

SET OUT ___

or SET BIT ___

Do not forget, therefore, to program RST OUT ___ or RST BIT ___ in the program sequence.



Header statements :

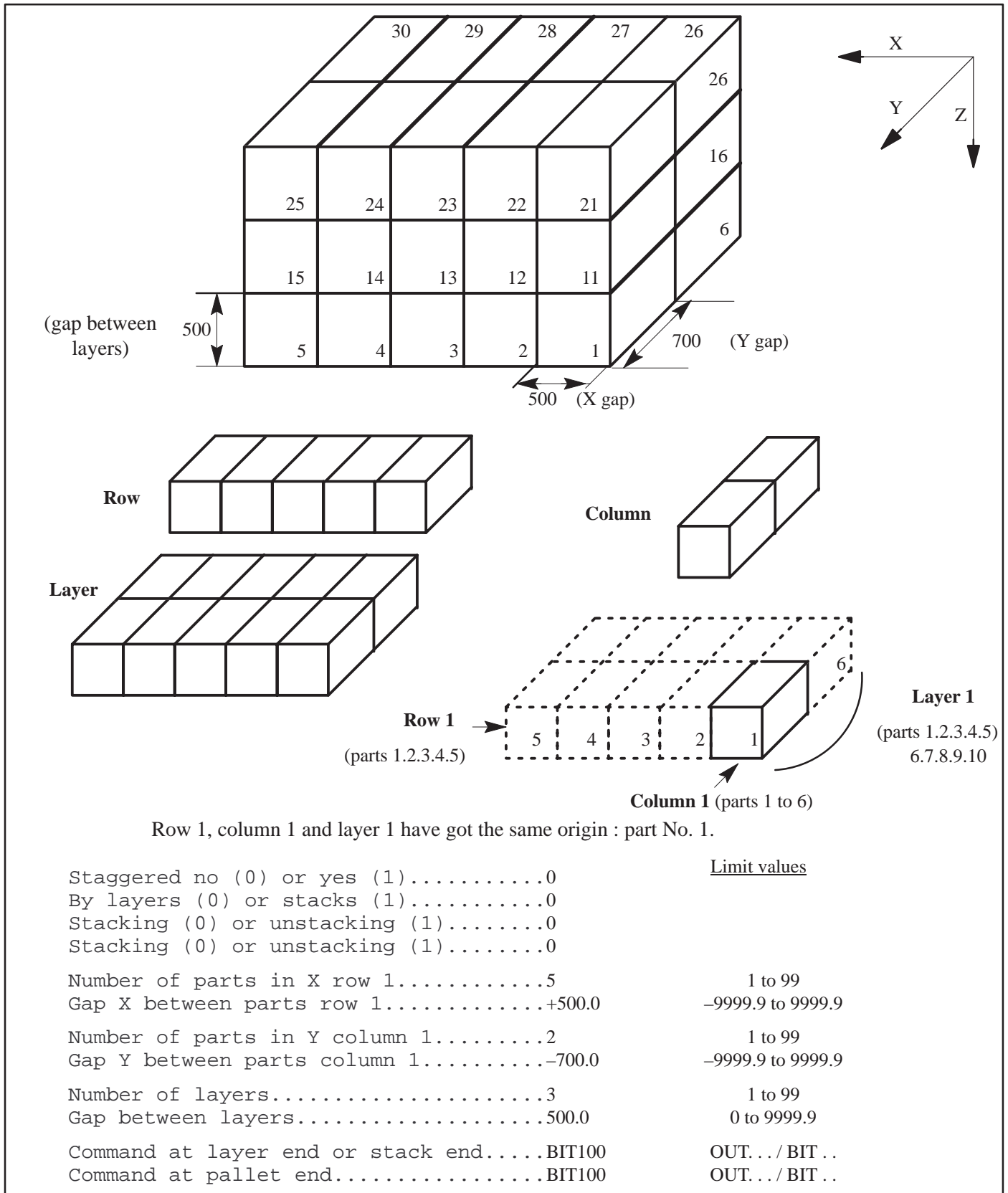
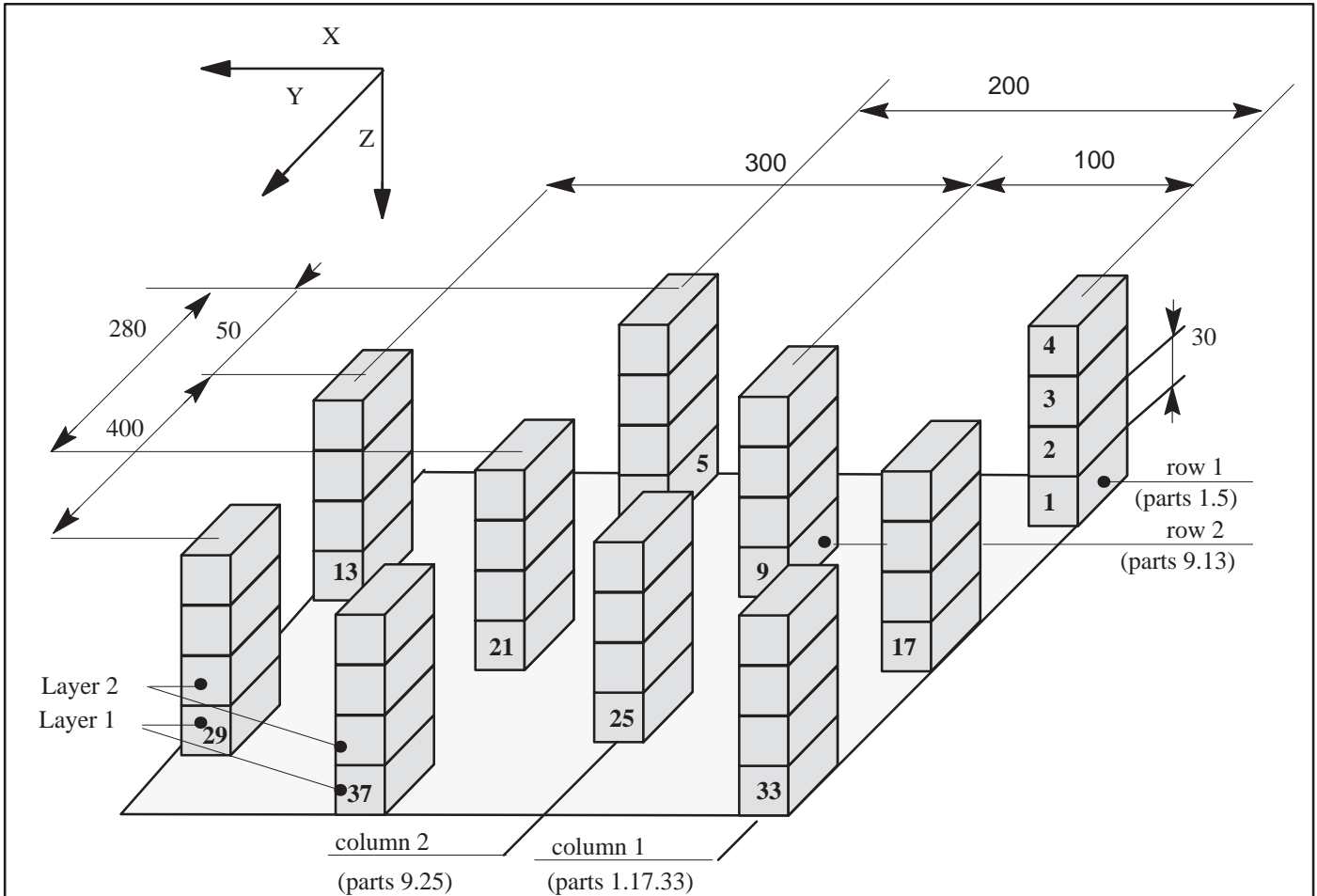


Figure 1 : Stacking without staggering



```

Staggered no (0) or yes (1).....1
By layers (0) or stacks (1).....1
Stacking (0) or unstacking (1).....0
Storage in XY (0) or YX (1).....0

Number of parts in X row 1.....2
Number of parts in X row 2.....2
Gap X between parts row 1.....200.0
Gap X between parts row 2.....300.0
Gap X between columns 1 / 2.....100.0

Number of parts in Y column 1.....3
Number of parts in Y column 2.....2
Gap Y between parts column 1.....280.0
Gap Y between parts column 2.....400.0
Gap Y between rows 1 / 2 .....50.0

Number of layers.....4
Gap between layers.....30.0

Command at layer end or stack end.....BIT50
Command at pallet end.....BIT51
    
```

Limit values

```

1 to 99
1 to 99
-9999.9 to 9999.9
-9999.9 to 9999.9
-9999.9 to 9999.9

0 to 99
0 to 99
-9999.9 to 0999.9
-9999.9 to 9999.9
-9999.9 to 9999.9

0 to 99
0 to 9999.9

OUT .../ BIT ...
OUT .../ BIT ...
    
```

Figure 2 : Stacking with staggering

Program :

▶ Calling a stacking subroutine :

The structure for calling a stacking subroutine is identical to those of standard subroutines. The only difference is the execution choice the system proposes

- Normal
- Reverse

The "Reverse" selection allows the execution of the program in the opposite way to that initially stated.

In "normal" execution mode, the stacking counter is increased before the distances are calculated. In "reverse" execution mode, the distances are calculated before the counter is decreased.

The "end of layer", "end of pallet" bits are also positioned in a coherent fashion. This function is mainly of interest in the management of stock reserves.

▶ Programming a stacking subroutine :

The movements for stacking calculations are programmed using the following codes :

```
X STK _____  
Y STK _____  
Z STK _____
```

When the program is running, the processor calculates position adjustments to be made in relation to the first pick-up/set-down programmed.

▶ System bits :

For the programmer use, the system sets the bits for choosing placement orientation, depending on whether the parts belong to odd or even lines.

- **Bit 0** : Reverse stacking operation bit.
- **Bit 1** : End of layer of stack bit.
- **Bit 2** : End of pallet bit.
- **Bit 3** : Odd row in progress bit.
- **Bit 4** : Odd column in progress bit.
- **Bit 5** : Odd layer in progress bit.
- **Bit 6** : Odd counter bit.
- **Bit 7** : Stacking in progress bit.

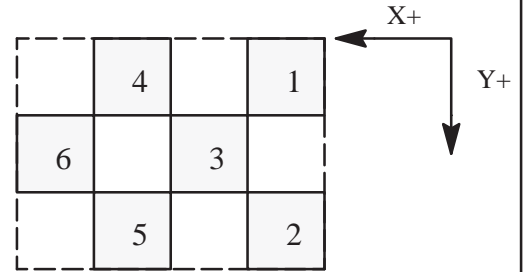
Bit 7 is set (if the conditions are fulfilled) from the moment the routine header is decoded until the word END is run or until a sequence returning to a label other than 00 is called.

The programmed bits or outputs must be reset to zero by the programmer once they have been used. Bits 0 to 5, at the end of the layer or pallet, are positioned when the header is decoded and are set to zero after a total home return.



Staggered no (0) or yes (1).....1
 By layers (0) or stacks (1).....0
 Stacking (0) or unstacking (1).....0
 Storage in XY (0) or YX (1).....1
 Number of parts in X row 1.....2
 Number of parts in X row 2.....2
 Gap X between parts row 1.....100.0
 Gap X between parts row 2.....100.0
 Gap X between columns 1 / 2.....50.0
 Number of parts in Y column 1.....2
 Number of parts in Y column 2.....1
 Gap Y between parts column 1.....100.0
 Gap Y between parts column 2.....100.0
 Gap Y between rows 1 / 250.0
 Number of layers.....1
 Gap between layers.....0
 Command at layer end or stack end.....
 Command at pallet end.....

YX



Staggered no (0) or yes (1).....1
 By layers (0) or stacks (1).....0
 Stacking (0) or unstacking (1).....0
 Storage in XY (0) or YX (1).....0
 Number of parts in X row 1.....2
 Number of parts in X row 2.....2
 Gap X between parts row 1.....100.0
 Gap X between parts row 2.....100.0
 Gap X between columns 1 / 2.....50.0
 Number of parts in Y column 1.....2
 Number of parts in Y column 2.....1
 Gap Y between parts column 1.....100.0
 Gap Y between parts column 2.....0
 Gap Y between rows 1 / 250.0
 Number of layers.....1
 Gap between layers.....0
 Command at layer end or stack end.....
 Command at pallet end.....

XY

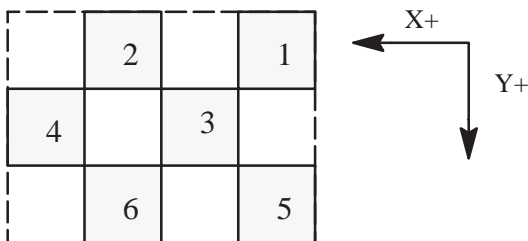
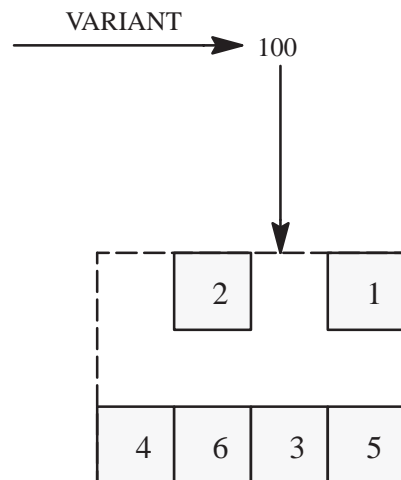
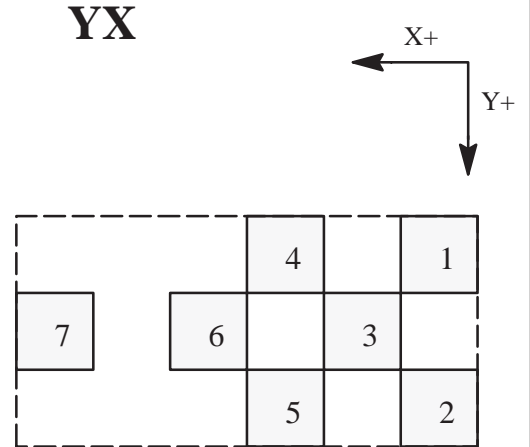


Figure 3 : Example No. 1 of special stacking sequence



Staggered no (0) or yes (1).....1
 By layers (0) or stacks (1).....0
 Stacking (0) or unstacking (1).....0
 Storage in XY (0) or YX (1).....1
 Number of parts in X row 1.....2
 Number of parts in X row 2.....3
 Gap X between parts row 1.....100.0
 Gap X between parts row 2.....100.0
 Gap X between columns 1 / 2.....50.0
 Number of parts in Y column 1.....2
 Number of parts in Y column 2.....1
 Gap Y between parts column 1.....100.0
 Gap Y between parts column 2.....0
 Gap Y between rows 1 / 250.0
 Number of layers.....1
 Gap between layers.....0
 Command at layer end or stack end.....
 Command at pallet end.....



Staggered no (0) or yes (1).....1
 By layers (0) or stacks (1).....0
 Stacking (0) or unstacking (1).....0
 Storage in XY (0) or YX (1).....0
 Number of parts in X row 1.....2
 Number of parts in X row 2.....3
 Gap X between parts row 1.....100.0
 Gap X between parts row 2.....100.0
 Gap X between columns 1 / 2.....50.0
 Number of parts in Y column 1.....2
 Number of parts in Y column 2.....1
 Gap Y between parts column 1.....100.0
 Gap Y between parts column 2.....0
 Gap Y between rows 1 / 250.0
 Number of layers.....1
 Gap between layers.....0
 Command at layer end or stack end.....
 Command at pallet end.....

VARIANT → 0

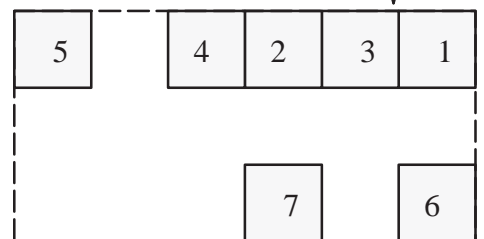
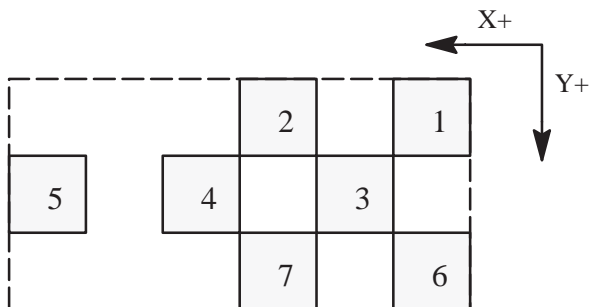


Figure 4 : Example No. 2 of special stacking sequence

III – GENERAL STACKING SUBROUTINES

The general stacking subroutines are subroutines 61 to 80. They are designed to describe stacking of parts that are not regular but that are repeated over several layers or several times in a cycle.

Each layer is described as follows :

- ▶ The first part is used as the absolute reference.
- ▶ The positions of subsequent parts are treated as relative (in relation to the first part) -> Gaps.
- ▶ A data item is allocated to each position, describing the position of the rotations (ROTWRD = Rotation word).

III – 1. Editing a subroutine – SP61 to 80 –

Subroutine		61	
Stacking to be taught ?			
YES		NO	



You will enter all the data concerning your stacking (predefinition).

You do not the position of your parts.

You want to teach them by moving the robot manually. Do not forget to put a part on the gripper.

III – 2. Teaching the general stacking sequences in adjust mode


III – 2. 1. Entering the parameters

```

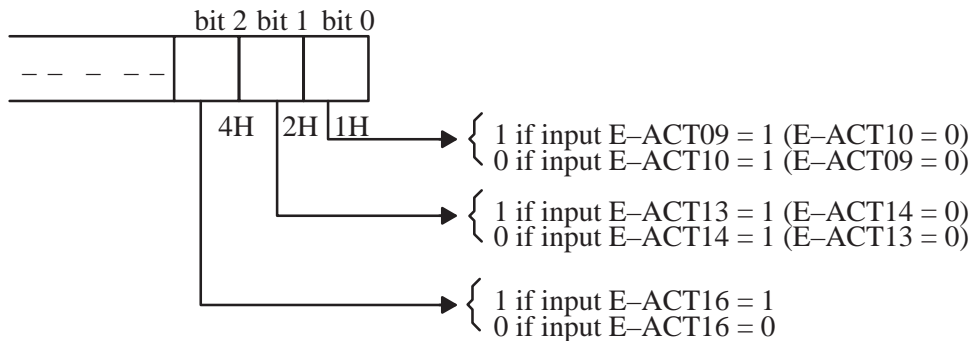
Alternated layers no (0) or yes (1)...1
Stacking (0) or unstacking (1).....0
Number of layers.....2
Gap between layers of same type.....150.0
Number of parts layer 1 "i".....3
Number of parts layer 2 "p".....5
Command at layer end or stack end....BIT61
Command at pallet end.....OUT12
    
```

III – 2. 2. Teaching the gaps

Motorised axes can be moved to bring the robot to the desired location. For part 1 layer 1, the X, Y, and Z gaps do not appear but will be stored in order to calculate the positions of subsequent parts relative to the position of part 1 layer 1.

The rotations should be positioned with the , , ,  and  keys before confirming the part position by pressing .

Depending on the rotation inputs present, and if parameter 5 = 1, the rotation Word, RotWrd is prepared (modification of bits weighted 1, 2, 4 by the rotations).



Pressing enables you to store the gaps of the declared axes and the rotation Word (RotWrd), if parameter 5 = 1, and go on to the next part, or the next layer, or the actual program.

III – 2. 3. Creating the program

After acquiring data on the last part of the last layer, the routine itself is edited.

Positions X, Y, Z, B, C, if the axes are declared, have been stored. They will be restored by keying in codes : **X.STK.L VAL Stk**. If the position is suitable, just press .

III – 3. Entering a general stacking subroutine

If all the positions and data of stackings are known, teaching can be refused.



Subroutine	61
Stacking to be taught ?	
YES	NO



These 2 headers correspond to the examples on the following pages :

Example 1 : (see chapter III – 5.)

```







Alternated layers no (0) or yes (1)...0
Stacking (0) or unstacking (1).....0
Number of layers.....1
Gap between successive layers.....2500
Number of parts layer 1 "i".....5
Command at layer end or stack end.....OUT10
Command at pallet end.....BIT107
    
```

Example 2 : (see chapter III – 6.)

```



Alternated layers no (0) or yes (1)...1
Stacking (0) or unstacking (1).....0
Number of layers.....2
Gap between layers of same type.....500.0
Number of parts layer 1 "i".....3
Command at layer end or stack end.....OUT10
Command at pallet end.....BIT107
    
```

* To enter and modify the first part of the header use :

- ▶ numerical keys for the numerical values,
- ▶ the ,  +  keys to choose the variable at the end of the layer of the pallet,
- ▶ the  or  keys to change line,
- ▶ the  key to confirm.

* For the gaps and the rotation words, use .

The value being processed flashes on the display (if the corresponding axis exists).

The  and  keys are used to increase the part number.

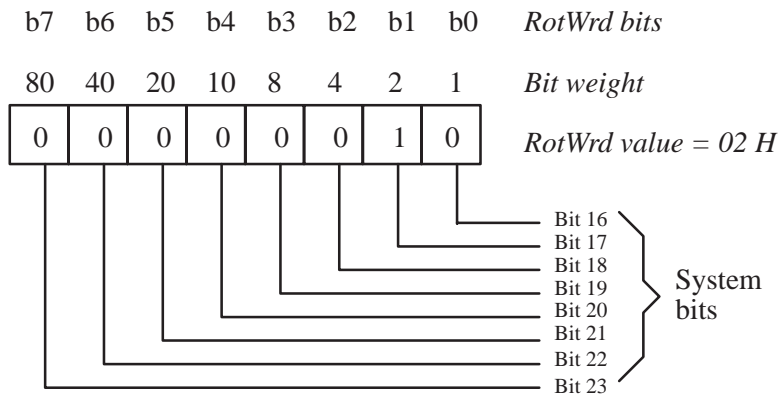
The keys : [layer -]  and [layer +]  are used to increase or decrease the layer number.

The  and  keys are used to move the axis to be processed forwards or backwards or RotWrd.

Note : Once you have entered a value followed by , you go on to the following value (following part, following layer).

The other bits are not changed.

III – 4. Description and use of the rotation word



The following should therefore be programmed in this particular stacking sequence :



```

- IF BIT 016 }
  ACT.09
- IF/BIT 016 }
  ACT.10
- IF BIT 017 }
  ACT.13
- IF/BIT 017 }
  ACT.14

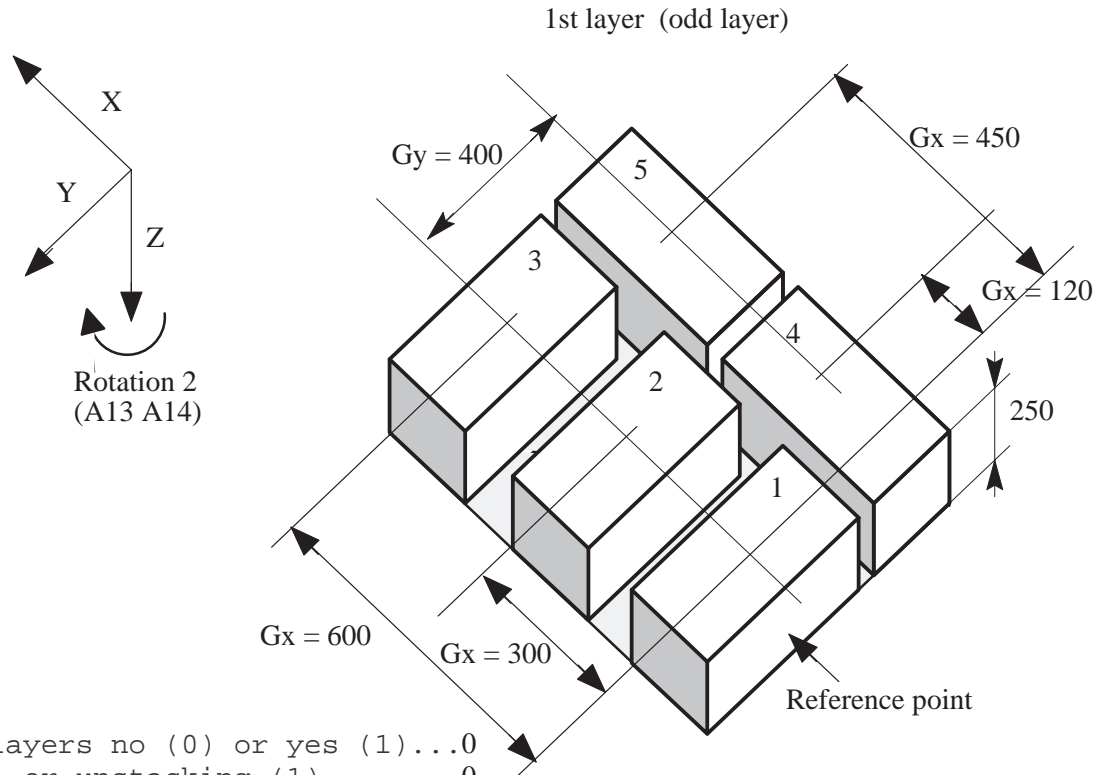
```

Example for a robot equipped with 2 standard pneumatic rotations.

When the stacking subroutine header is executed, the following bits are set :

- ▶ Bit 0 : Reverse stacking operation bit
- ▶ Bit 1 : End of layer bit
- ▶ Bit 2 : End of pallet bit
- ▶ Bit 3 : } Not significant (used in Regular stacking)
- ▶ Bit 4 : } Not significant (used in Regular stacking)
- ▶ Bit 5 : Odd layer in progress bit
- ▶ Bit 6 : Odd counter bit
- ▶ Bit 7 : Stacking in progress bit
- ▶ BIT 16 at 1 indicates that a rotation ACT 09 is to be carried out
- ▶ BIT 16 at 0 indicates that a rotation ACT 10 is to be carried out
- ▶ BIT 17 at 1 indicates that a rotation ACT 13 is to be carried out
- ▶ BIT 17 at 0 indicates that a rotation ACT 14 is to be carried out
- ▶ BIT 18 at 1 indicates a rotation with a stopper ACT.16
- ▶ BIT 18 at 0 indicates a rotation without a stopper ACT.16.

III – 5. Example 1



```

Alternated layers no (0) or yes (1)...0
Stacking (0) or unstacking (1).....0
Number of layers.....1
Gap between successive layers.....250,0
Number of parts layer 1 "i".....5
Command at layer end or stack end....OUT10
Command at pallet end.....BIT107
    
```

← this parameter should not be given a value if the number of layers is 1, which is the case in the example.

GAPS :

Part 1 Layer 1

```

Gx = 0      Gy = 0      Gz = 0
B = 0      C = 0      RotWrd = 3
    
```

Part 2 Layer 1

```

Gx = 300.0  Gy = 0      Gz = 0
B = 0      C = 0      RotWrd = 3
    
```

Part 3 Layer 1

```

Gx = 600.0  Gy = 0      Gz = 0
B = 0      C = 0      RotWrd = 3
    
```

Part 4 Layer 1

```

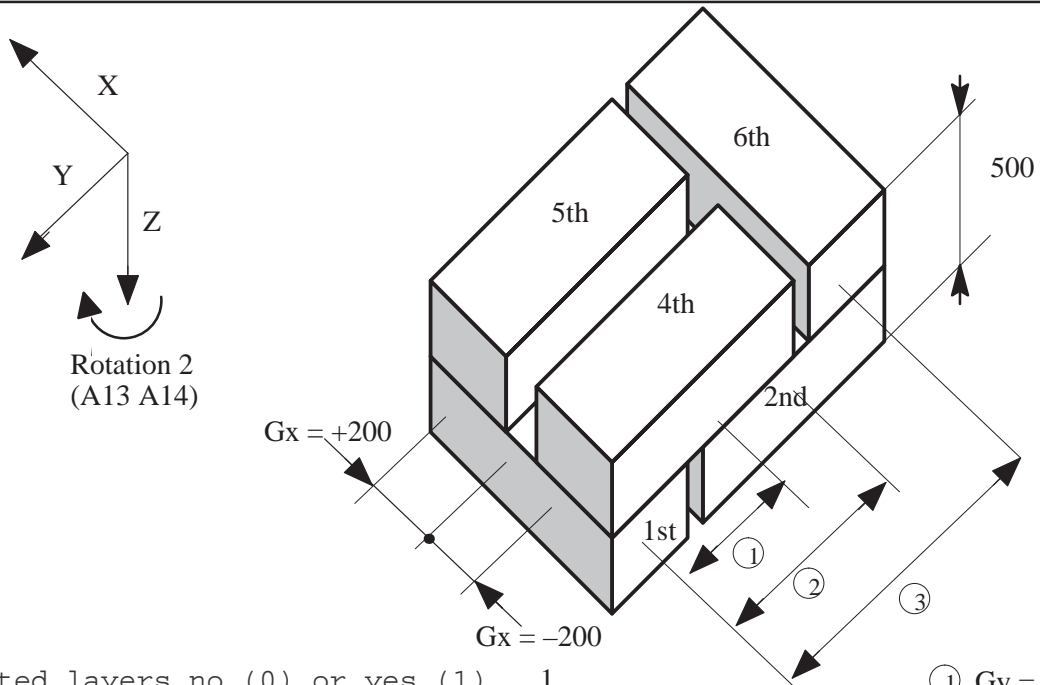
Gx = 120.0  Gy = 400.0  Gz = 0
B = 0      C = 0      RotWrd = 1
    
```

Part 5 Layer 1

```

Gx = 450.0  Gy = 400.0  Gz = 0
B = 0      C = 0      RotWrd = 1
    
```

III – 6. Example 2 – Alternate layers



Alternated layers no (0) or yes (1)...1
 Stacking (0) or unstacking (1).....0
 Number of layers.....2
 Gap between layers of same type.....500,0
 Number of parts layer 1 "i".....3
 Number of parts layer 2 "p".....3
 Command at layer end or stack end....OUT10
 Command at pallet end.....BIT107

- ① Gy = -200
- ② Gy = -400
- ③ Gy = -600

GAPS :

Part 1 Layer 1 (1st Part)

Gx = 0 Gy = 0 Gz = 0
 B = 0 C = 0 RotWrd = 0

Part 2 Layer 1 (2nd Part)

Gx = -200.0 Gy = -400.0 Gz = 0
 B = 0 C = 0 RotWrd = 1

Part 3 Layer 1 (3rd Part)

Gx = 200.0 Gy = -400.0 Gz = 0
 B = 0 C = 0 RotWrd = 1

Part 1 Layer 2 (4th Part in drawing)

Gx = -200.0 Gy = -200.0 Gz = 250.0
 B = 0 C = 0 RotWrd = 1

Part 2 Layer 2 (5th Part in drawing)

Gx = 200.0 Gy = -200.0 Gz = 250.0
 B = 0 C = 0 RotWrd = 1

Part 3 Layer 2 (6th Part in drawing)

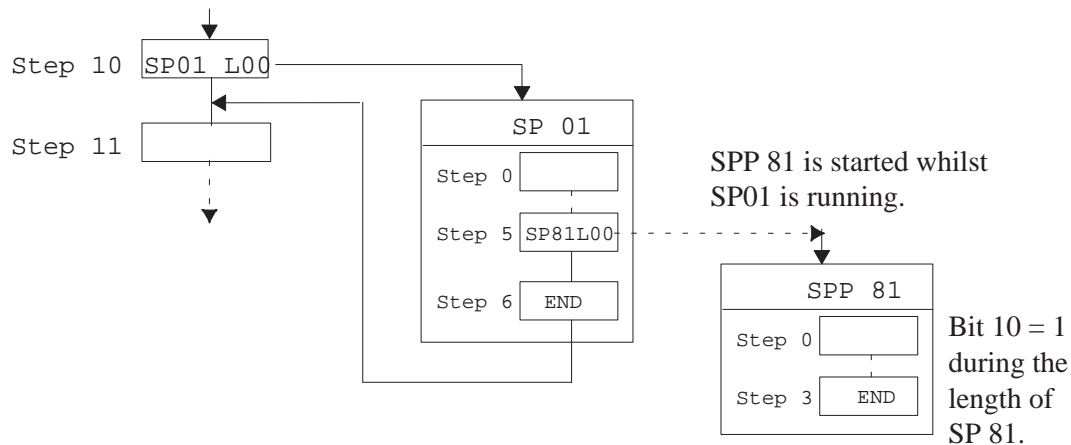
Gx = 0 Gy = -600.0 Gz = 250.0
 B = 0 C = 0 RotWrd = 0

IV – PARALLEL SUBROUTINES – SP81 TO 99 –

The parallel subroutines are subroutines 81 to 99. They are also groups of instructions, structured in a sequential manner.

They are executed in parallel to the normal running of the cycle.

There is no return address for the Parallel subroutines.



The system bit number 10 is set to 1 when the SPP is started and goes back to 0 at the end of the SPP.

A few restrictions :

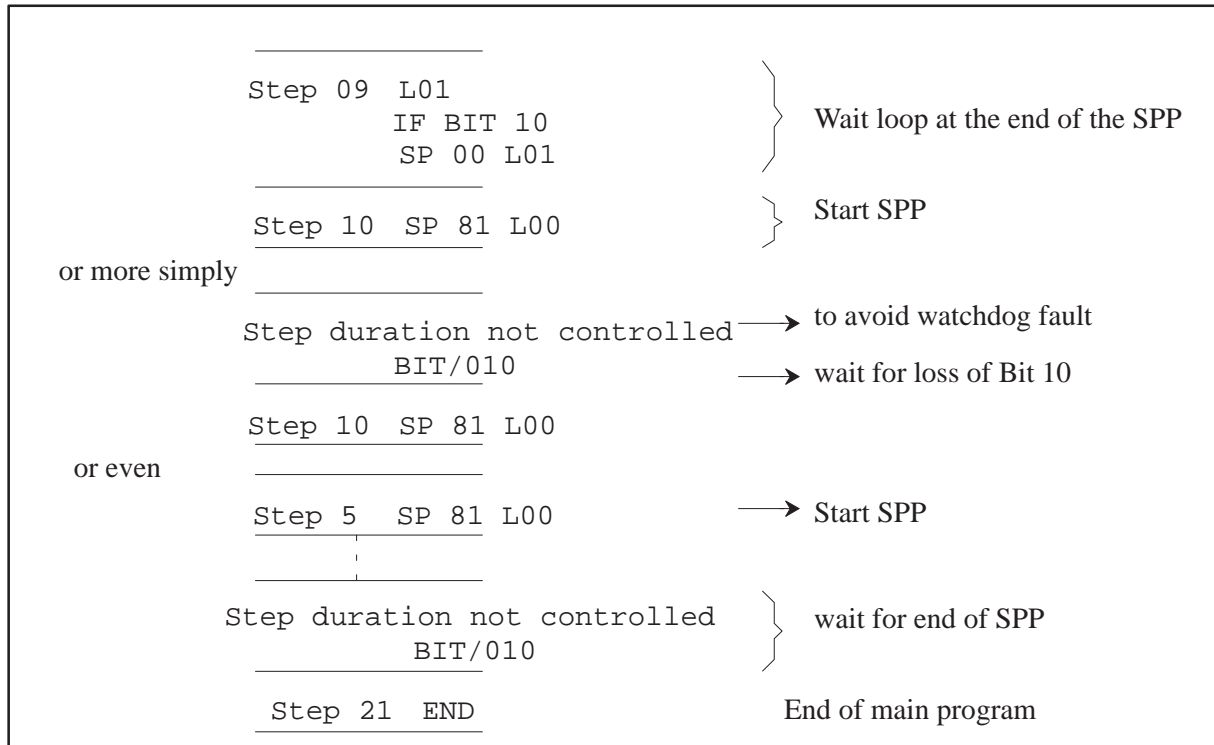
- ▶ Only one Parallel Subroutine can be active at a certain time. The calling-up of a second one before the first one is finished, triggers the fault `D_73: PARALLEL SP ALREADY RUNNING`.
- ▶ In an SPP, all the instructions of the main program can be used, apart from :
 - the MASTER preparatory function and consequently all triggered movements and CTL controls,
 - the special instructions `Await machine cycle` and `Await validation PRG change`.
 - Calling-up of other SPs or SRs.

Moreover, for robots using PIP parameters, the following instructions are taken into account during machine dialogue in the part program only. They are ignored in the SPP :

- ▶ `Validation machine cycle`
- ▶ `part grips in the IMM`.

It is not possible to activate two SPPs at the same time or launch the execution of an SPP that is already running. Therefore, using this bit 10, you can check whether an SPP has finished before starting another one.

System bit number 10, which is automatically set by the system to 1 when the SPP starts and to 0 when it is finished, may be used.



As the ordinary subroutines, the SPPs can be dependant on certain conditions.



```
Step 8  IF BIT 100
          SP 81 L00
          IF /BIT 100
          SP82 L00
```

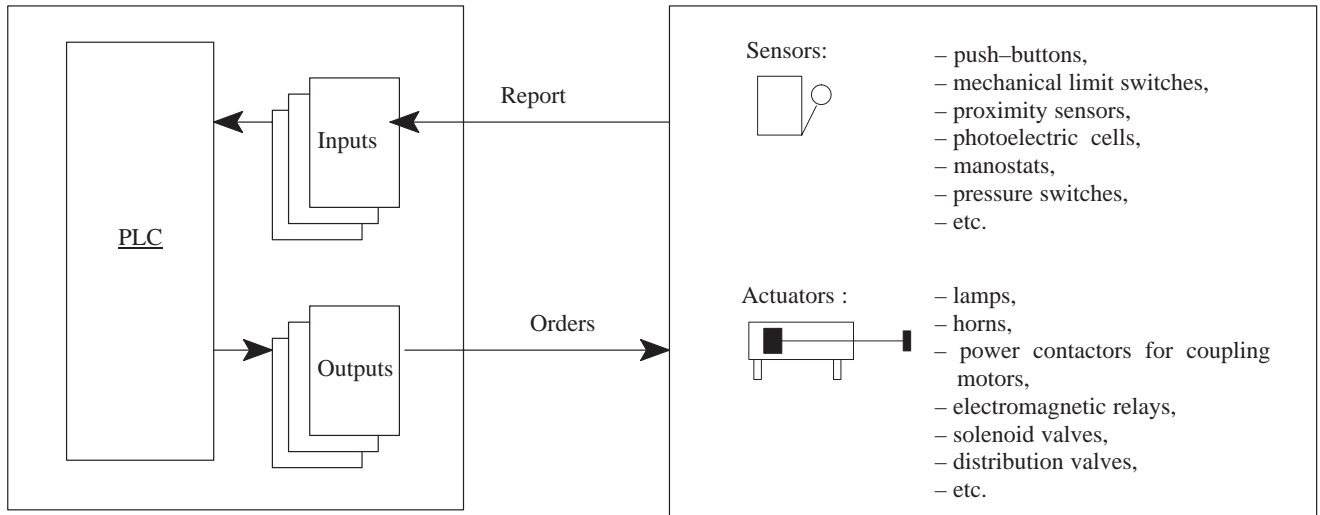


SIMPLE or TOTAL home return or PCO requests CANCEL the SPP running and reset the system bit 10 to 0.

V – BUILT IN PLC

V – 1. General information

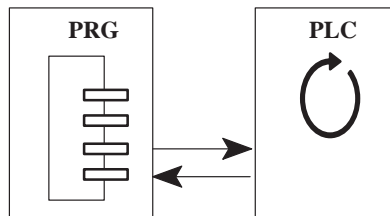
SEPRO NUMERICAL CONTROL SYSTEM



The command part of a simple automated installation can be managed by the PLC which is an additional function in the main program PRG of the robot. It is distinguished by :

- ▶ cyclic scanning every tenth of a second in **AUTO** and **STEP BY STEP** modes.
- ▶ the possible use of sequential or combination programming.

Each PLC program must be associated with a main program (PRG). Operands are common to the PLC and the PRG.



Program memories :

An instruction in the Sepro PLC can describe a contact, a coil, a counter, etc ...

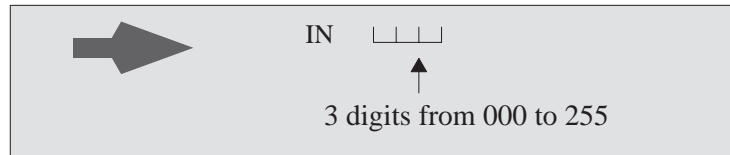
The last instruction of the program is always END OF PLC PROGRAM. This instruction resets the central processing unit to line 0 of the PLC program.

V – 2. The instructions



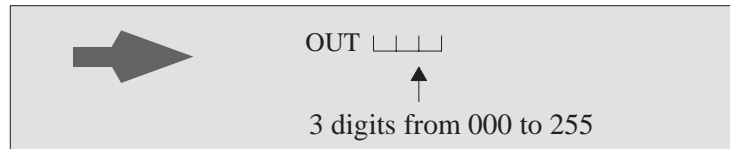
V – 2. 1. The boolean operands

► Inputs :



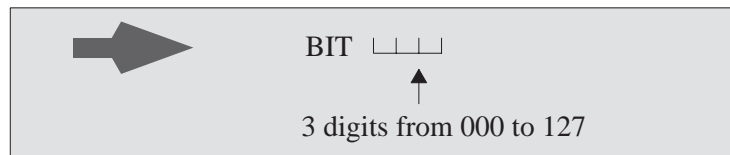
=> the inputs shown are common to the main program (PRG) and the PLC.

► Outputs :

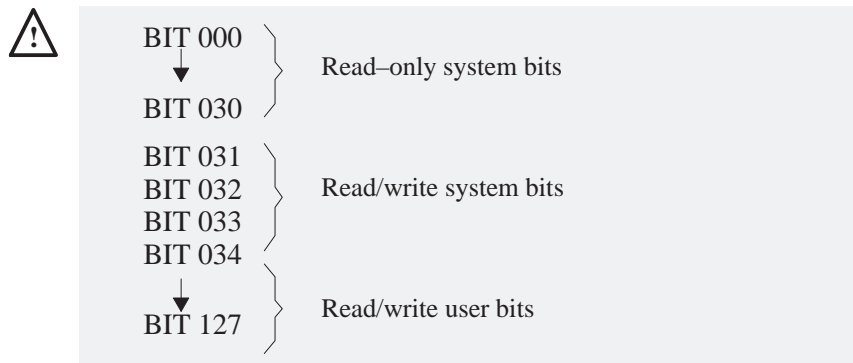


=> the outputs shown are common to the main program (PRG) and the PLC.

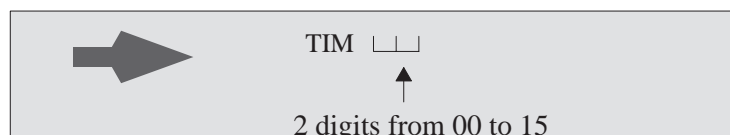
► Bits :



=> the bits shown are common to the main program (PRG) and the PLC.



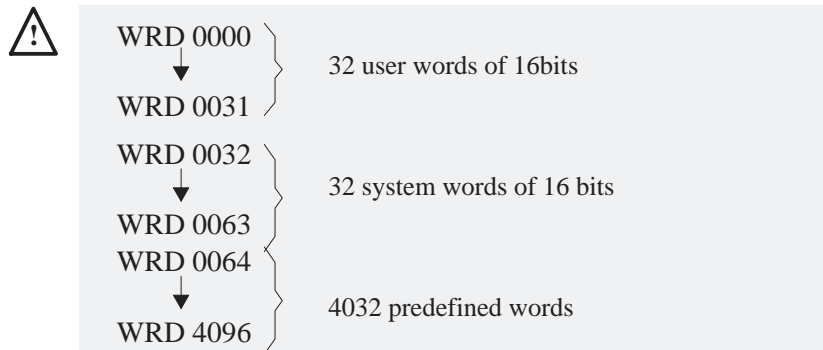
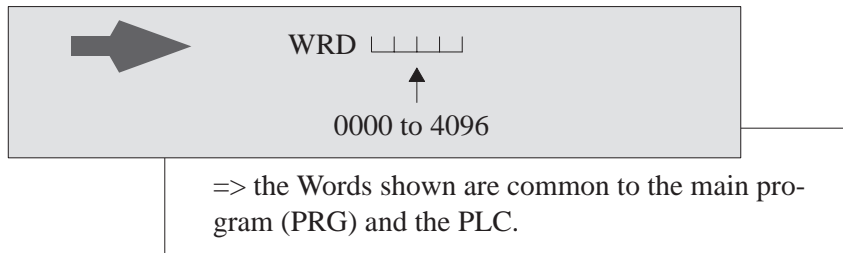
► TIM :



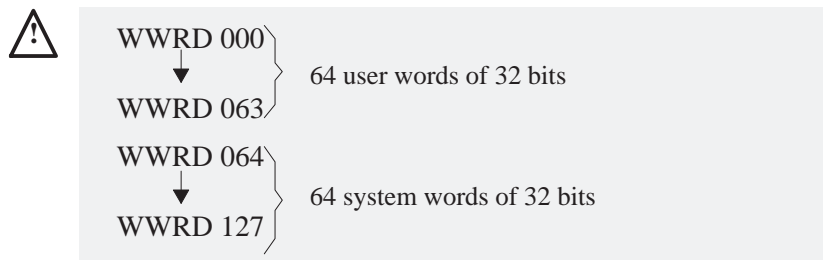
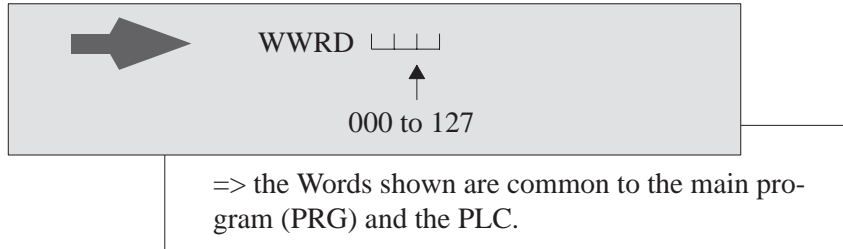
=> bits resulting from associated TIMERS (TIMER 00 to TIMER 15)

V – 2. 2. Structured operands

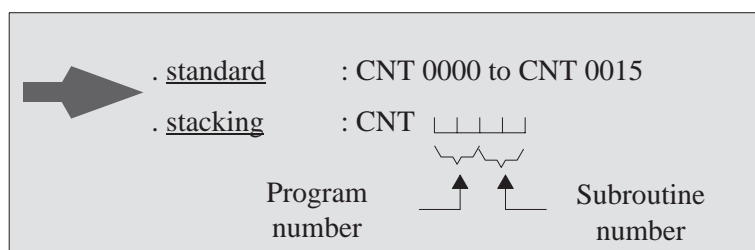
► Words :



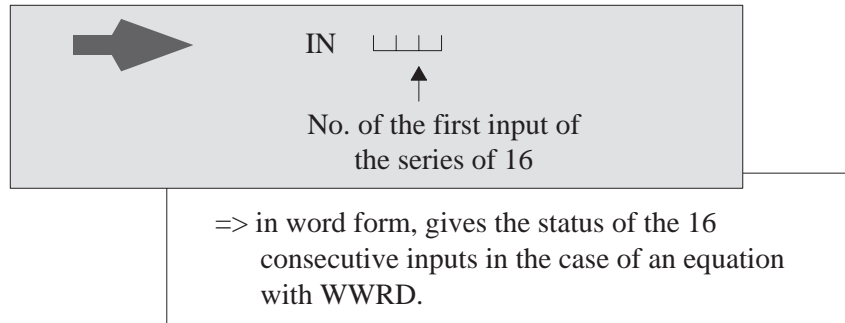
► Double words :



► Counters :



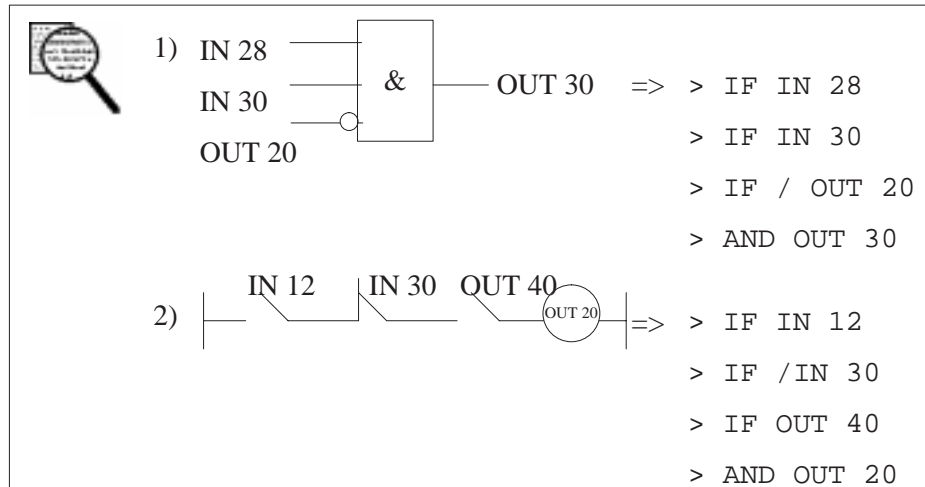
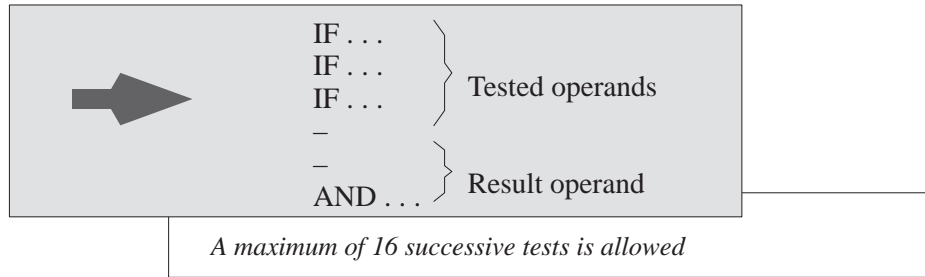
► Series of 16 consecutive inputs :



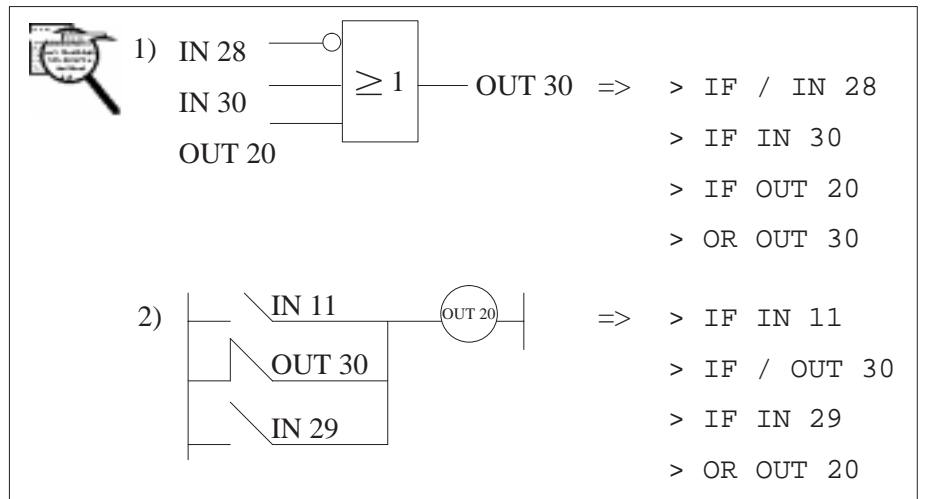
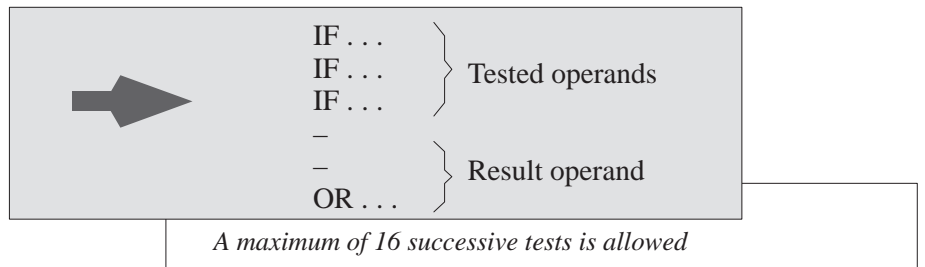
V – 2. 3. The instruction codes

► Boolean operators

▪ AND : Logic "AND"

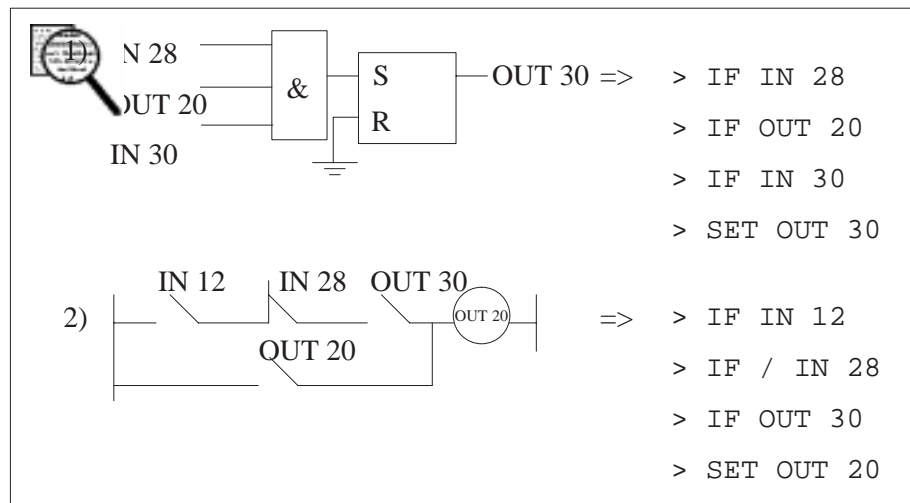
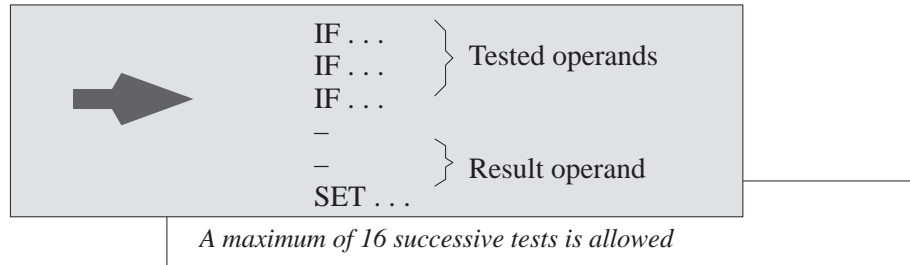


▪ OR : Logic "OR"

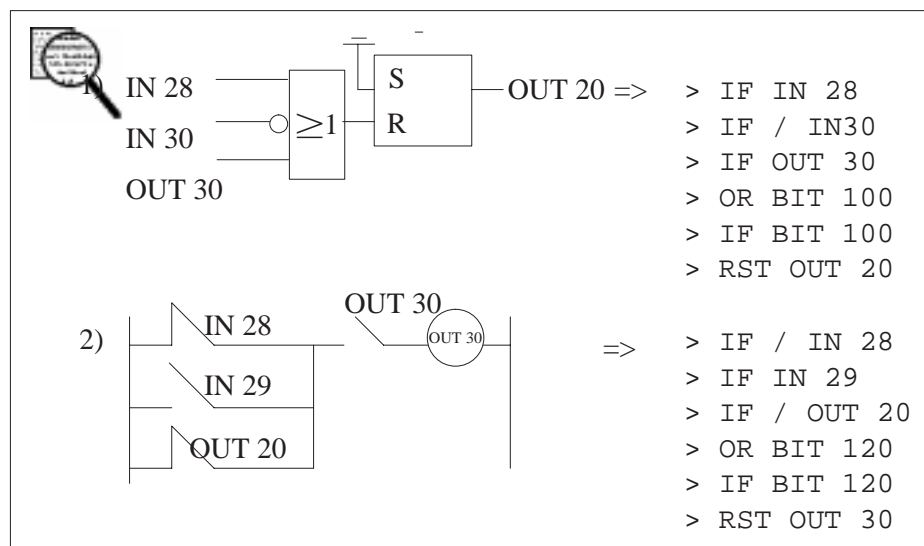
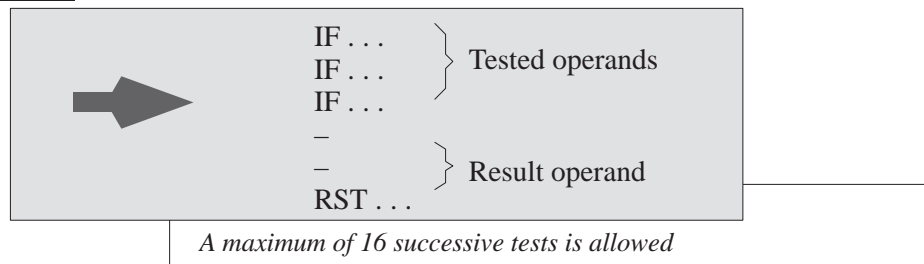


► Allocation

▪ Set to 1 :



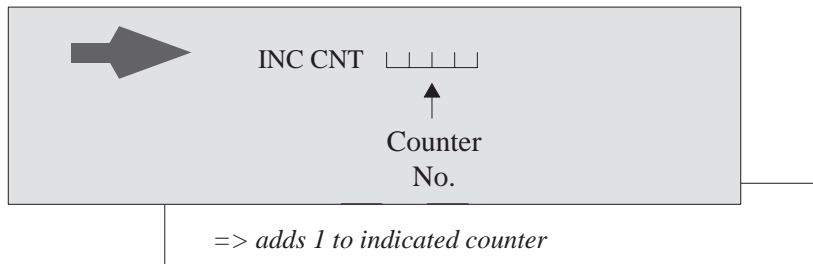
▪ * Set to 0 :



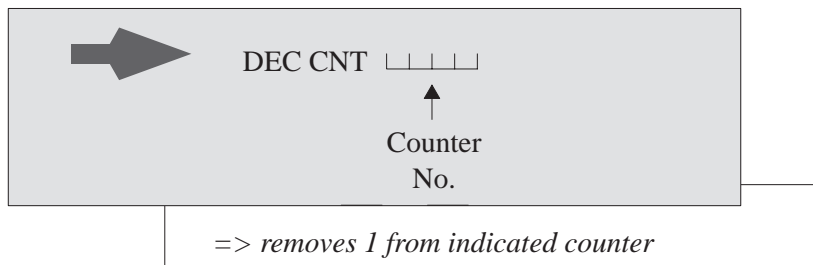
► Basic operators applied to counters

- Available counters : see chapter I – 6. page 20.
- Operations :

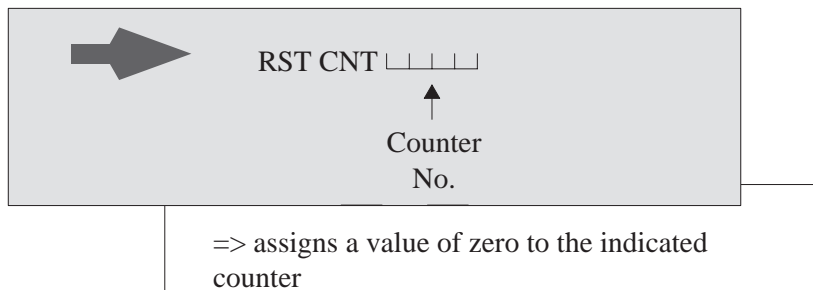
. Incrementation :



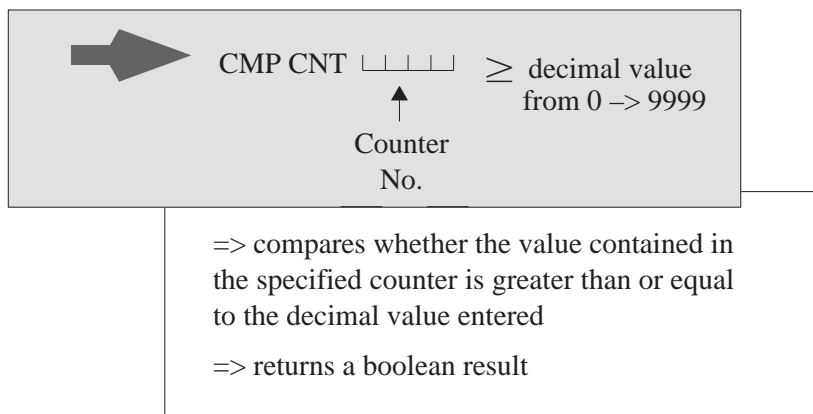
. Decrementation :



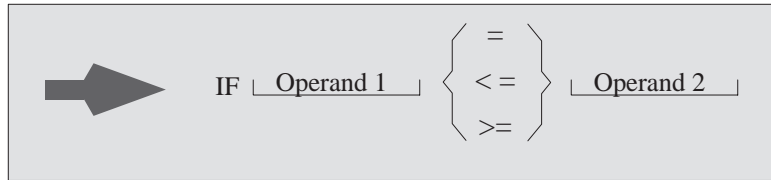
. Reset to zero :



. Comparing a counter to a decimal value :



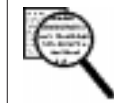
► Comparison



=> depending on operator, determines:

- . the equality of two operands,
- . whether Operand 1 is less than Operand 2,
- . whether Operand 1 is greater than Operand 2.

=> sends back a boolean result.



CNT 0003 = 10_D (Decimal)

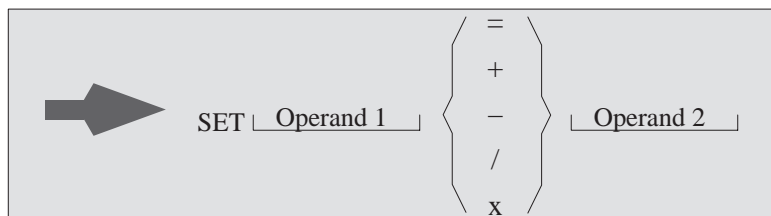
CNT 0007 = 19_D (Decimal)

> IF CNT 0003 <= CNT 0007

> SET OUT 20

Output No. 20 is set to 1 since the value of counter 0003 is lower than the value of counter 0007.

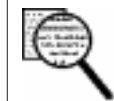
► Arithmetic operations



=> depending on the operator, performs the following:

- . assigns the value of Operand 2 to Operand 1,
- . adds the two operands,
- . subtracts Operand 2 from Operand 1,
- . divides Operand 1 by Operand 2,
- . multiplies the two operands.

=> the result overwrites the value contained in Operand 1.



Before:

CNT 0006 = 18_D (Decimal)


CNT 0012 = 6_D (Decimal)

> SET CNT 0006 / CNT 0012


After:

CNT 0006 = 3_D (Decimal)

- "OR" mask :

 SET Operand 1 OR Operand 2

=> performs a logic OR on bits forming the operands.
=> the result overwrites the value contained in Operand 1.

 Before:
WORD 40 = 0001 1001 0100 0111 (Binary); operand to evaluate
WORD 60 = 1011 0001 1110 1100 (Binary): mask operand

> SET WRD 40 OR WRD 60

After:
WORD 40 = 1011 1001 1110 1111 (Binary)

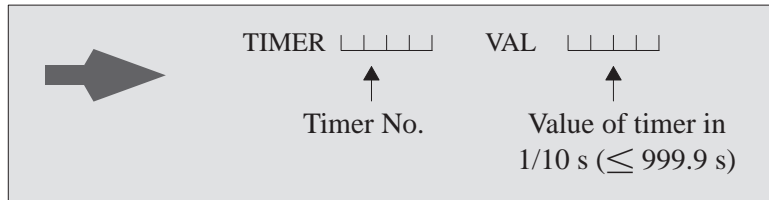
► TIMER (PLC time delays)

▪ Available TIMERS :

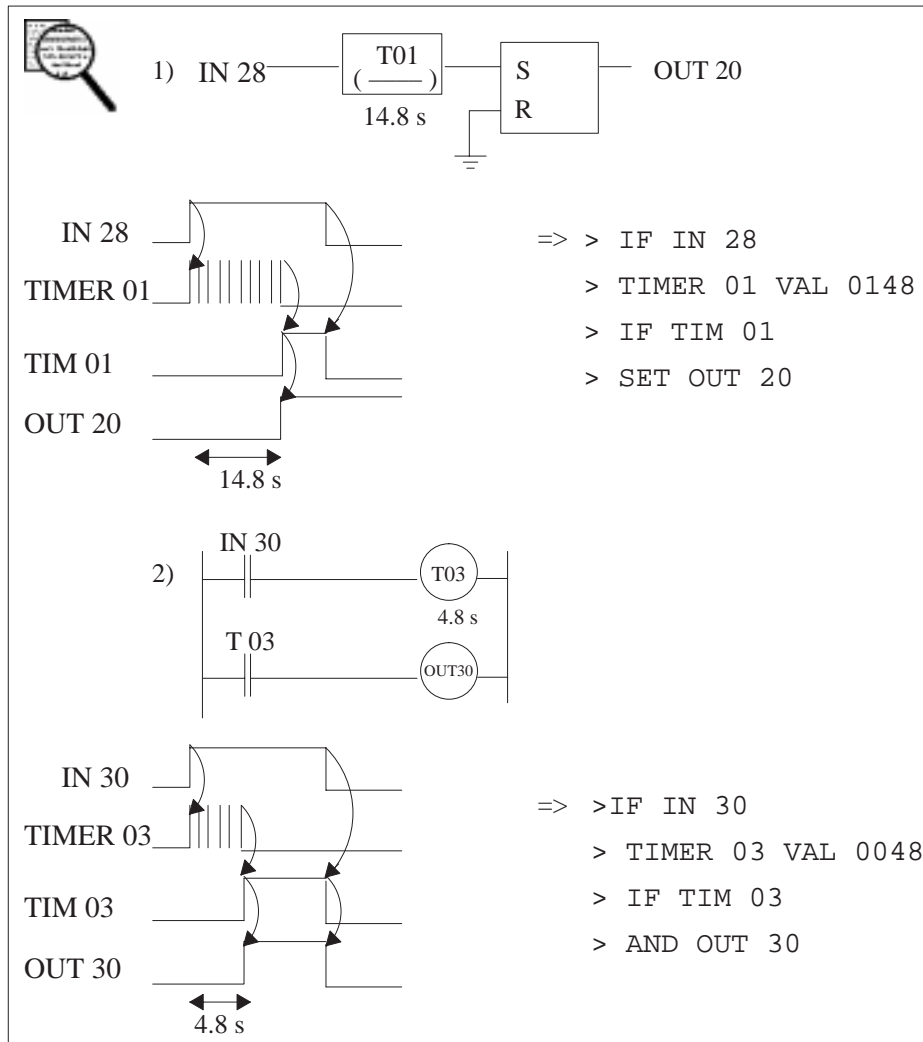
16 timers are available : TIMER 00 to TIMER 15

Each one is associated with a result bit : TIM 00 to TIM 15

▪ Use :



=> starts the indicated timer for the period specified after VAL.
=> the associated result bit TIM is set to 1 when the timer runs out.



Result bit TIM is set to 0 if the TIMER command operand is lost.
The command operand must therefore be present when the TIMER times out.
If it is lost before time is out, the timer is interrupted.

V – 3. Application examples

Since two different types of analysis may be applied, there are two possible types of encoding :

or

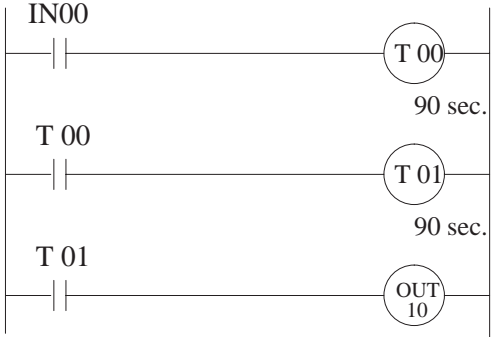
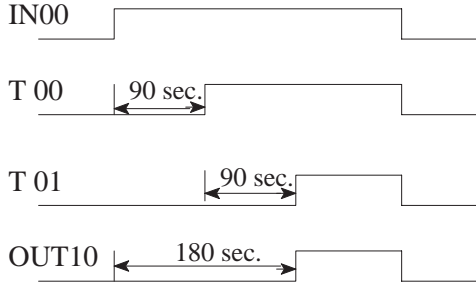
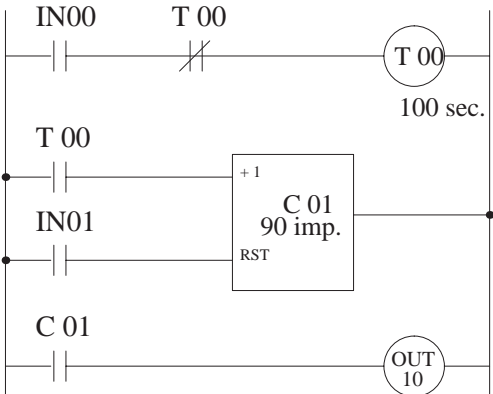
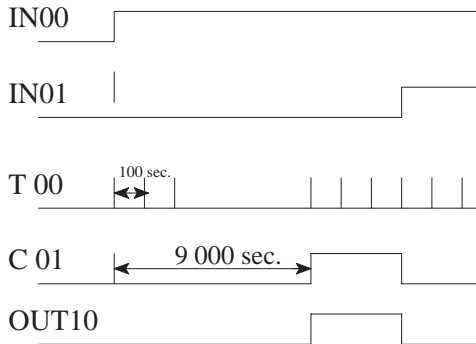
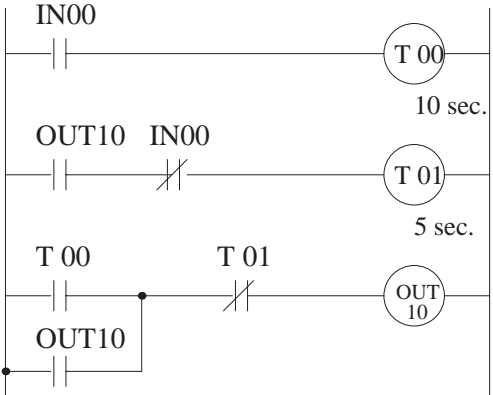
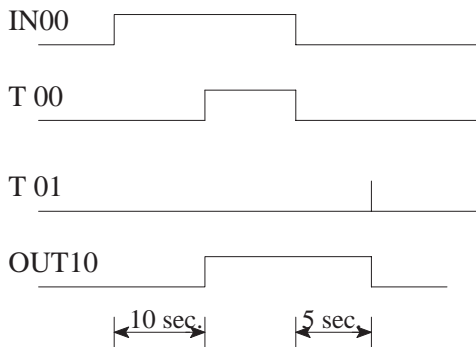
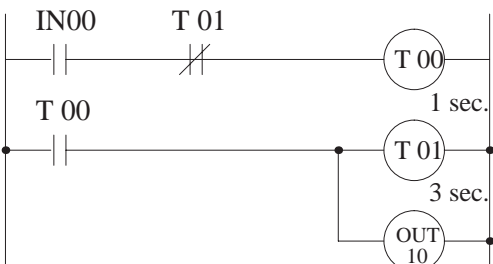
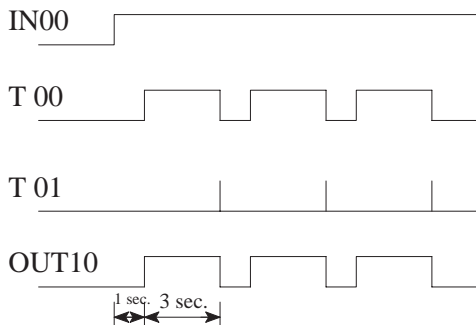
– <u>COMBINATION (Relay drawings):</u>	
<i>Advantages:</i>	. quick analysis . simplifies program writing
<i>Disadvantages:</i>	. debugging sometimes difficult . troubleshooting sometimes difficult . cycle safety functions difficult to set up
=> Suitable for simple applications.	
– <u>SEQUENTIAL (Grafcet drawing):</u>	
<i>Advantages:</i>	. easy debugging . easier to program startups and restarts . easier to program safety functions . easier to troubleshoot
<i>Disadvantages:</i>	. detailed analysis required . takes longer to write program . cycle safety functions difficult to set up
=> Suitable for sophisticated applications.	

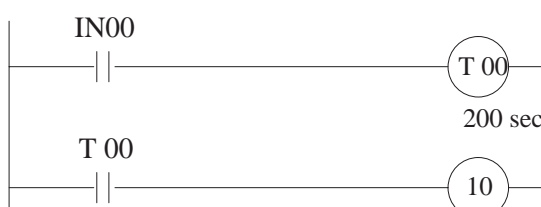
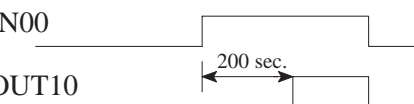
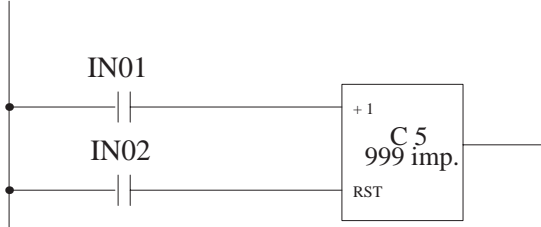
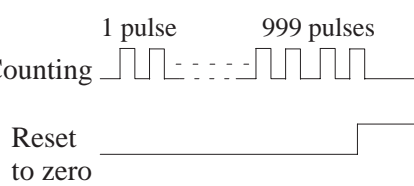
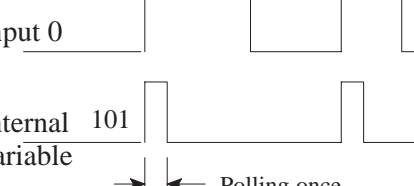
V – 3. 1. Combination (relay drawings)

► Combination of AND and OR operators

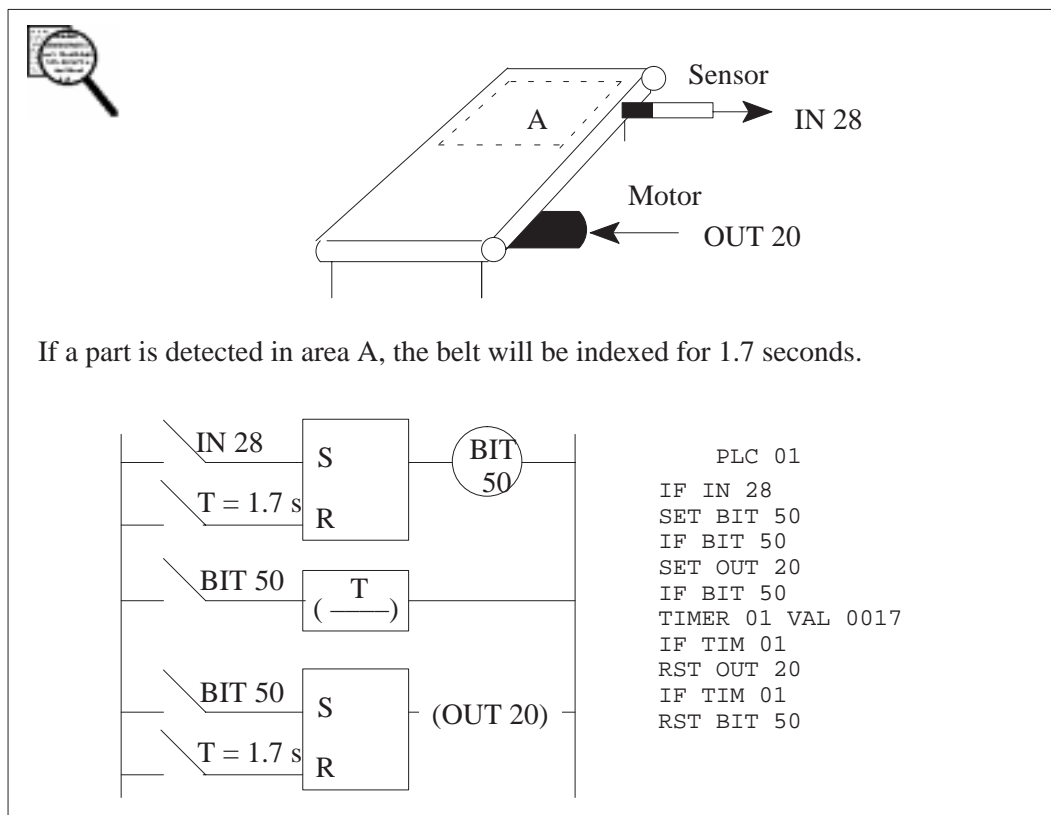
DRAWING	PROGRAM	DESCRIPTION
	<pre>IF IN 0 IF IN 1 AND BIT 100 IF IN 10 IF BIT 100 OR BIT 101 IF BIT 101 IF IN 2 IF/ IN 3 AND OUT 10</pre>	<p>Circuit a</p> <p>Circuit b</p>
	<pre>IF IN 2 IF IN 3 AND BIT 50 IF BIT 50 IF IN 10 IF IN 4 OR BIT 51 IF BIT 51 IF IN 0 IF/ IN 1 AND OUT 10</pre>	<p>Circuit b</p> <p>Circuits a and b series link</p>
	<pre>IF IN 2 IF / IN 3 AND BIT 100 IF / IN 4 IF IN 10 AND BIT 101 IF BIT 100 IF BIT 101 OR BIT 102 IF / IN 0 IF IN 1 IF BIT 102 AND OUT 10</pre>	<p>Circuit b1</p> <p>Circuit b2</p> <p>Circuit b1 + b2 = b</p> <p>Circuits a and b series link</p>

► Timer and counter applications

DRAWING	PROGRAM	DESCRIPTION
<p>. Timers in series :</p> 	<pre>IF IN 00 TIMER 00 VAL 0900 IF TIM 00 TIMER 01 VAL0900 IF TIM 01 AND OUT 10</pre>	
<p>. Timer > 999 sec. :</p> 	<pre>IF IN 00 IF / TIM 00 TIMER 00 VAL1000 IF TIM 00 INC.CNT 01 IF CNT 01 >= 90 AND OUT 10 IF IN 01 RST CNT 01</pre>	
<p>. Timers on rising and falling edge :</p> 	<pre>IF IN 00 TIMER 00 VAL 0100 IF OUT 10 IF / IN 00 TIMER 01 VAL 0050 IF TIM 00 IF OUT 10 OR BIT 100 IF BIT 100 IF / TIM 01 AND OUT 10</pre>	
<p>. Flashing light :</p> 	<pre>IF IN 00 IF / TIM 01 TIMER 00 VAL 0010 IF TIM 00 TIMER 01 VAL 0030 IF TIM 00 AND OUT 10</pre>	

DRAWING	PROGRAM	DESCRIPTION
<p>. Timer :</p> 	<pre>IF IN 00 TIMER 00 VAL 2000 IF TIM 00 AND OUT 10</pre>	
<p>. Counter :</p> 	<pre>IF IN 01 INC.CNT 5 IF CNT 5 >= 999 IF IN 02 RST.CNT 5</pre>	
<p>. Creation of a rising edge when an input appears :</p>	<pre>IF IN 00 IF / BIT 100 AND BIT 101 IF IN 00 AND BIT 100</pre>	

► Example : indexing belt



V – 3. 2. Sequential (Grafcet drawing)

► Programming the sequential instructions

Each stage of the Grafcet drawing is defined by a **counter value**.

▪ Activating the stage :

Activating the operand associated with the stage "E n".

➔ Two conditions must be met :

- 1) All the stages immediately preceding E (n-1) must be active (validation of transition).
- 2) Receptivity R (n-1) associated with the transition must be true.

▪ Deactivating the stage :

Deactivation of the variable associated with "E n".

➔ When the stage "E n" is activated, this immediately deactivates preceding stages E (n-1). Stage "E n" is therefore deactivated by activating those stages immediately following, i.e. E (n+1).

As soon as a transition occurs, all stages immediately following are activated, and all stages immediately preceding are deactivated.

1) General case

```


=> > -
      > -
      > IF CNT 01 = 7
      > IF IN 28
      > SET CNT 01 = 8
      > -
      > -
  
```

2) Example of divergence in AND

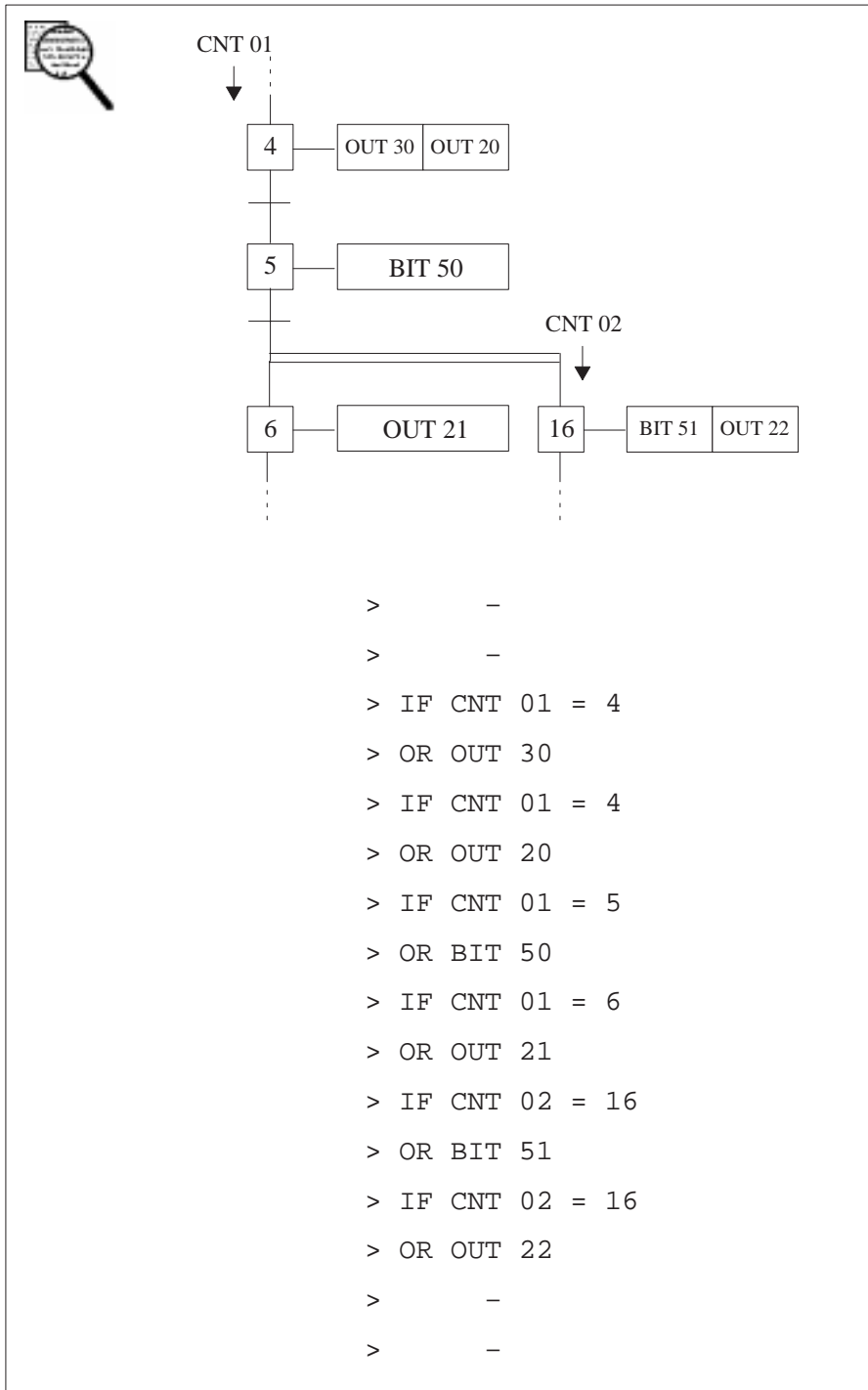
```

=> > -
      > -
      > IF CNT 01 = 7
      > IF IN 28
      > SET CNT 01 = 8
      > IF CNT 01 = 8
      > SET CNT 02 = 18
      > -
  
```

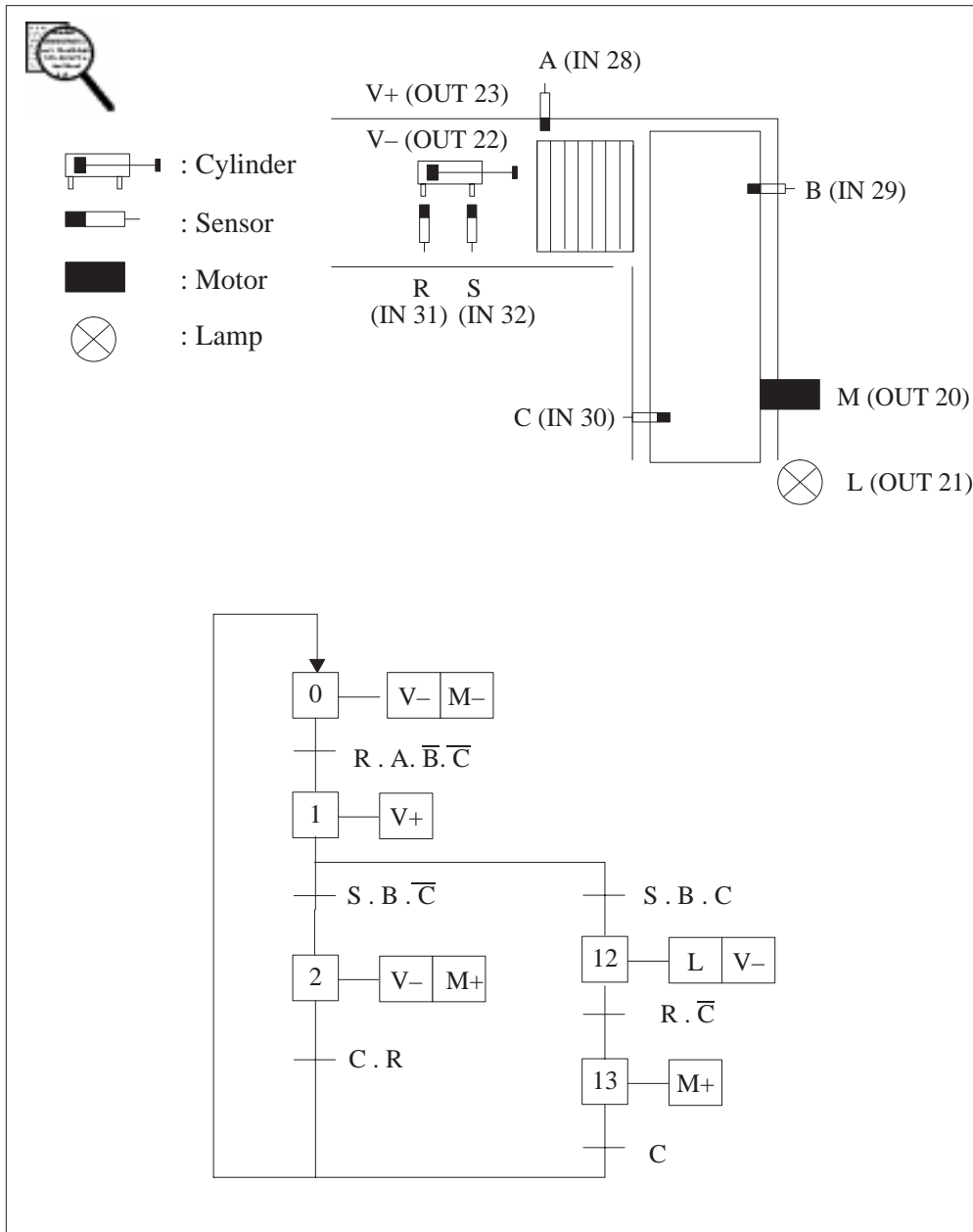
▪ Programming actuators

 Each stage can include the activation of one or more variables.

=> These variables may be bits / outputs / words / wwords / timers.



▪ Example : transferring onto conveyor belt



Sequential part

```

                                PLC 01
Stage 0      IF BIT 09          Total HR
              RST CNT 00

              IF BIT 51
              IF BIT 52
              OR BIT 53

              IF BIT 53
              RST CNT 00
-----
Stage 1      IF CNT 00 = 0000_D
              IF IN 31          Cylinder retracted
              IF IN 28          Part present
              IF / IN 30
              IF / IN 29
              SET CNT 00 = 1_D
-----
Stage 2      IF CNT 00 = 1_D
              IF IN 32          Cylinder extended
              IF IN 29          Part on belt
              IF / IN 30        Part present at end of belt
              SET CNT 00 = 2_D

              IF CNT 00 = 2_D
              IF IN 30          Part present at end of belt
              IF IN 31          Cylinder retracted
              AND BIT 51
-----
Stage 12     IF CNT 00 = 1_D
              IF IN 32          Cylinder extended
              IF IN 29          Part on belt
              IF IN 30          Part present at end of belt
              SET CNT 00 = 12_D
-----
Stage 13     IF CNT 00 = 12_D
              IF IN 31          Cylinder retracted
              IF / IN 30        Part not present at end of belt
              SET CNT 00 = 13_D

              IF CNT 00 = 13_D
              IF IN 30          Part present at end of belt
              AND BIT 52

```

Actuator part

```

              IF CNT 00 = 0_D
              IF CNT 00 = 2_D
              IF CNT 00 = 12_D
              OR OUT 22          Retract cylinder

              IF CNT 00 = 0_D
              RST OUT 20         Stop motor

              IF CNT 00 = 1_D
              OR OUT 23          Extend cylinder

              IF CNT 00 = 2_D
              SET OUT 20         Start motor

              IF CNT 00 = 12_D
              OR OUT 21          Switch on lamp

              IF CNT 00 = 13_D
              SET OUT 20         Start motor
-----
END

```

V – 4. Monitoring PLC : PLC 99

Unlike PLC 1 to 98 that only work in Step by Step and Automatic modes, PLC 99 also works in Adjust mode.

It is not linked to a particular program and works with any program (including those associated with a PLC).

The monitoring PLC is perfectly suitable for managing equations that are common to all the programs.

The refresh time is the same as for the standard PLCs (100 ms).

The monitoring PLC can write in the safety word S0 (Word 37).



As PLC 99 is active with all the programs, you are advised to use a locking / unlocking tool in the fault equations. This is because one particular program may not need the same monitoring as the other programs. This locking–unlocking may be done using an input positioned by an external system (switch, sensor....) or by the contents of a WORD (value of program encoding or any other external data item).



Monitoring PLC example : robot is POWERED DOWN when the X axis goes over a certain position.

```
IF    IN    100                monitoring is active for this program
IF    WWRD  77    >=    35000_D X position greater than or equal to 3500.0 mm
IF/   OUT    3                overtravel forcing inactive
SET   WORD  62    =    5_D    robot powered down
```

The robot will go into fault D_5 : MOVEMENT OUTSIDE CAMS as soon as the X axis goes over the position 3500 mm. This is the case in all modes. To cancel the fault and free the robot, press and hold down START in adjust mode and press the X movement key.



Monitoring PLC example : ROTATIONS BLOCKED when the Z axis goes over a certain position.

```
IF    IN    100                monitoring is active for this program
IF    WWRD  79    >=    8000_D Z position greater than or equal to 800.0 mm
IF/   OUT    3                overtravel forcing inactive
AND   BIT    100              conditions for rotations are lost
IF    BIT    100              conditions for rotations are lost
SET   WORD  37    AND    07FF_H rotations are invalidated
```

It will no longer be possible to control the robot rotations as soon as the Z axis goes over the position 800 mm. If one of the rotations is requested, it will not be carried out and the robot will display : D_7 : PROTECTED MOVEMENT. This is the case in all modes.

To cancel the blocking and to authorise the rotations again, you must write another equation that allocates a value to WORD 37.

```
IF/   BIT    100              conditions for rotations re-established
SET   WORD  37    OR    F800_H rotations are validated
```

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.

WE'RE HERE TO HELP

To contact Customer Service personnel, call:



HOW TO CONTACT CUSTOMER SERVICE

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.

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.

BEFORE YOU CALL ...

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.

– INDEX –

A

Addition, 2
Allocation, 2
AND, 2
Axes' position, 18
Axis movement, 24

B

BIT, 1, 2, 4
Bit
 . system, 4, 32, 42
 . user, 4

C

CNT, 1, 20
Condition, 3
Counter, 1, 20
Customised key, 19

D

Decrease, 20
Division, 2
Double Word, 1

E

Egality, 2
End of cycle, 9, 16
End of slow approach, 27
Euromap 17, 18

F

FAL, 27
Fault, 17, 42, 62

Fault number, 17

G

Greater than or equal to, 2

I

IF, 3
IN, 1
Increase, 20
Input, 1

L

Less than or equal to, 2

M

Multiplication, 2

O

Operand, 1, 25
Operator, 1
OR, 2
OUT, 1, 2
Output, 1

P

PLC, 44
 . monitoring, 62

R

RESET, 3
Robot number, 15
Robot option, 15
Robot type, 16

S

Setting to zero, 3

SLA, 27

Slow approach, 27

SPP, 42

Subroutine

. general stacking, 35

. parallel, 42

. regular stacking, 28

Subtraction, 2

T

Test, 3

TIME, 22

Timer, 1

W

WORD, 1, 2, 5

. user, 5

WORD, 1, 2, 18