USERGUIDE

Encoding and Addressing
CN900++
Version 0.4

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This manual is out-of-date and is provided only for its technical information, data and capacities.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containing warnings, information, precautions, and procedures that may be more adequate than those contained in this out-of-date manual.

– CONTENTS –

I – MEMORY

I – 1. ACCESSING THE MEMORY

After accessing ”Memory Mangement” by pressing [Memo_M] (programming menu), pressing the [MRead] key gives access to the read (or modification) function for the user and system RAM or EEPROM memory (at the address of the memory box in fault if necessary).

The address of the area at which reading is to begin is given in hexadecimal (0 to F) using the numerical keypad and first row of alphanumerical keys of the keyboard.

Certain areas are directly accessible by keyboard:

AND R : start of the RAM programming of MPs.

TRAJ L : start of the RAM programming of PLCs.

BIT P : start of saving EEPROM memory.

P+1 M : start of the MODULE in which the programs are stored.

For example: to access the start of EEPROM memory, the procedure is as follows:

[Memo_M] –> [MRead] –> [Address] –> BIT P

* **The keys**:

– [+] or [–] change addresses 2 by 2.
– [↑] or [↓] change addresses 10 by 10 (hexadecimal).
– [P+1] or [P–1] change addresses 100 by 100 (hexadecimal).

* **The function keys**:

– **[Address]** gives a new address.

– **[Modif]** modifies contents of memory area displayed (word).

– **[Search]** searches for a particular word (e.g.: FA1B).

– **[Print]** prints memory contents starting from the displayed address (in order to obtain the incorrect instructions which will be printed as ????).

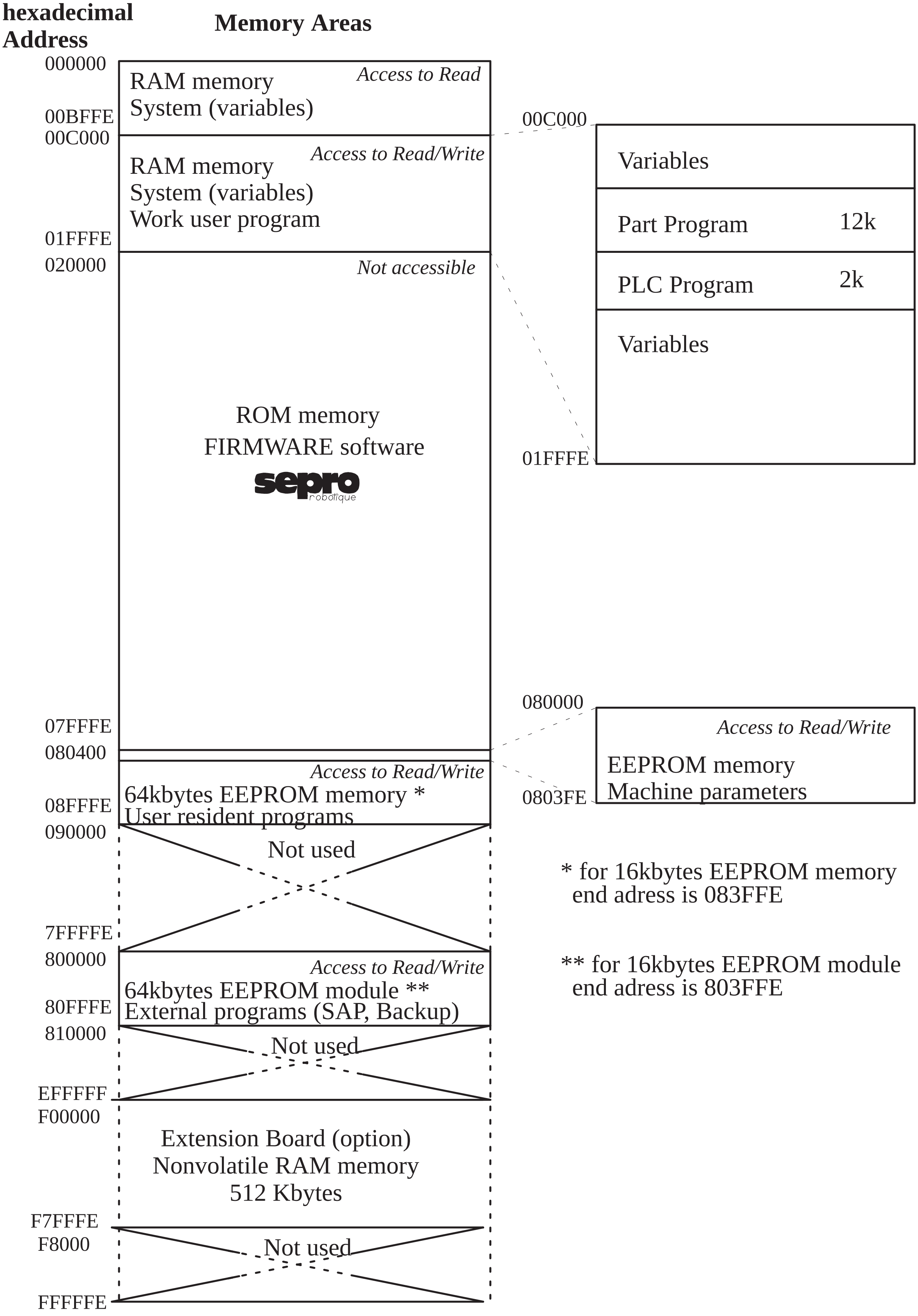
– **[StopP]** stops the sending of the memory contents to the printer.

Notes: To access the modification function, a password is necessary which remains valid as long as the user does not exit the “MRead” procedure. Certain critical system areas cannot be read and all requests to modify them will be rejected.

By default, the value given after modification request is 0 x FFFF (useful to delete words in EEPROM).

Like the other functions, the EXIT key is used to abandon a request or to exit the procedure.

I – 2. MEMORY AREAS



* for 16kbytes EEPROM memory
end adress is 083FFE

** for 16kbytes EEPROM module
end adress is 803FFE

II – INSTRUCTIONS CODES

II – 1. PART PROGRAMS

Type of Instruction	Display	Codop (hexadecimal)	Examples
ACTION	ACT 00 (to 32 or 90 or 99)	A000 [oper. 16 bits] ↓ Action No.	A000000C = ACT12
OUTPUT	OUT 000 (to 127)	A001 [oper. 16 bits] ↓ Output No.	A0010050 = OUT080
INPUT **Normal**	IN 000 (to 127)	A002 [oper. 16 bits] ↓ Input No.	A002000A = IN010
INPUT **Reverse**	IN/000 (to 127)	A003 [oper. 16 bits] ↓ Input No.	A0030020 = IN/032

TIMER	TIME 001 to 999	A004[oper.4bits]0[oper.11bits]
		↓ ↓
		SAP Marker No. Value in 1/10s
			A004000A = TIME010
			A004300A = TIME010
			Marker P03

	TIME W_00 to 15	A004 0000 1 [oper.11bits]
		↓
		Word No.
			A004080A = TIMEW10
			A004080F = TIMEW15

BIT	BIT 000 (to 127)	A005 [oper. 16 bits] ↓ Bit No.	A0050063 = BIT 99
	/ BIT 000 (to 127)	A006 [oper. 16 bits]	A006007D = BIT 127

Type of Instruction	Display	Codop (hexadecimal)	Examples
FUNCTION (FUN)			
SPEED	VEL.X 001 to 100	B000[oper.4bits][oper.12bits]	B0000062 = VEL.X 098
in % of parametered	VEL.Y 001 to 100	B001[oper.4bits][oper.12bits]	B001000A = VEL.Y 010
speed	VEL.Z 001 to 100	B002[oper.4bits][oper.12bits]	B0020012 = VEL.Z 018
	VEL.B 001 to 100	B003[oper.4bits][oper.12bits]	B0030064 = VEL.B 100
	VEL.C 001 to 100	B004[oper.4bits][oper.12bits]	B004A032 = VEL.C 050
		↓ ↓	Marker P10
		SAP Marker No. / Value in 1/10s	
	VEL.X WW_066 0r 067	B050 0000 [oper.12bits]	B0500042 = VEL.X WW066
	VEL.Y WW_066 0r 067	B051 0000 [oper.12bits]	B0510043 = VEL.Y WW067
	VEL.Z WW_066 0r 067	B052 0000 [oper.12bits]	B0520042 = VEL.Z WW066
	VEL.B WW_066 Or 067	B053 0000 [oper.12bits]	B0530042 = VEL.B WW066
	VEL.C WW_066 0r 067	B054 0000 [oper.12bits]	B0540043 = VEL.C WW067
		↓ Word No	
ACCELERATION	ACC.X 001 to 100	B010 [oper. 16 bits]	B010000F = ACC.X 015
in % of parametered	ACC.Y 001 to 100	B011 [oper. 16 bits]	B0110064 = ACC.Y 100
acceleration	ACC.Z 001 to 100	B012 [oper. 16 bits]	B0120044 = ACC.Z 068
	ACC.B 001 to 100	B013 [oper. 16 bits]	B0130005 = ACC.B 005
	ACC.C 001 to 100	B014 [oper. 16 bits]	B0140032 = ACC.C 050
		↓ Value in %	
Master MOVEMENT	MASTER.X	B030	
	MASTER.Y	B031	
	MASTER.Z	B032	
	MASTER.B	B033	
	MASTER.C	B034	
IMPRECISION	IMP.X	B040	
	IMP.Y	B041	
	IMP.Z	B042	
	IMP.B	B043	
	IMP.C	B044	

Type of Instruction	Display	Codop (hexadecimal)	Examples
<u>MOTORIZED MOTIONS</u>			
SLOW APPROACH	SLA.X 001 to 100	B020 [oper. 16 bits]	B0200026 = SLA.X 026
in % of maximum	SLA.Y 001 to 100	B021 [oper. 16 bits]	B0210034 = SLA.Y 034
speed parametered	SLA.Z 001 to 100	B022 [oper. 16 bits]	B0220090 = SLA.Z 090
	SLA.B 001 to 100	B023 [oper. 16 bits]	B0230100 = SLA.B 100
	SLA.C 001 to 100	B024 [oper. 16 bits]	B0240010 = SLA.C 010
		↓ Value in %	
<u>LINEAR</u>			
ABSOLUTE	X.ABS_L distance	C000[oper.8bits][oper.24bits]	C00000000664=X.ABS.L00163.6
(Numerical operands)	Y.ABS_L distance	C001[oper.8bits][oper.24bits]	C001000F423F=Y.ABS.L9999.9
	Z.ABS_L distance	C002[oper.8bits][oper.24bits]	C00200000320=Z.ABS.L00080.0
	B.ABS_L distance	C003[oper.8bits][oper.24bits]	C0030000003F=B.ABS.L00006.3
	C.ABS_L distance	C004[oper.8bits][oper.24bits]	C0040000050C=C.ABS.L00150.0
STACKING	X.FIL_L distance	C010[oper.8bits][oper.24bits]	C01000008ACF=X.FIL.L03453.5
	Y.FIL_L distance	C011[oper.8bits][oper.24bits]	C01100030DE3=Y.FIL.L20016.3
	Z.FIL_L distance	C012[oper.8bits][oper.24bits]	C01200000159=Z.FIL.L00034.5
	B.FIL_L distance	C053	Reserved for general stacking
	C.FIL_L distance	C054	Absolute value from headline
RELATIVE	X.REL_L distance	C020[oper.8bits][oper.24bits]	C020800000A0=X.REL.L–0016.0
	Y.REL_L distance	C021[oper.8bits][oper.24bits]	C021000000A0=Y.REL.L–0016.0
	Z.REL_L distance	C022[oper.8bits][oper.24bits]	C0228001869F=Z.REL.L–9999.9
	B.REL_L distance	C023[oper.8bits][oper.24bits]	C02300002706=B.REL.L+0999.9
	C.REL_L distance	C024[oper.8bits][oper.24bits]	C0240000000A=C.REL.L+0001.0
CHECKING	X.CTL_L distance	C030[oper.8bits][oper.24bits]	C03000000664=X.CTL.L00163.6
	Y.CTL_L distance	C031[oper.8bits][oper.24bits]	C031000F423F=Y.CTL.L9999.9
	Z.CTL_L distance	C032[oper.8bits][oper.24bits]	C03200000320=Z.CTL.L00080.0
	B.CTL_L distance	C033[oper.8bits][oper.24bits]	C0330000003F=B.CTL.L00006.3
	C.CTL_L distance	C034[oper.8bits][oper.24bits]	C0340500050C=C.CTL.L00150.0
		↓ Marker No SAP ↓ Distance in 1/10 mm	Marker P05
<u>ROTATIVE</u>			
ABSOLUTE	X.ABS_R Angle	C100[oper.8bits][oper.24bits]	C10000000664=X.ABS.R00163.6
(Numerical operands)	Y.ABS_R Angle	C101[oper.8bits][oper.24bits]	C101000005DC=Y.ABS.R00150.0
	Z.ABS_R Angle	C102[oper.8bits][oper.24bits]	C10200000320=Z.ABS.R00080.0
	B.ABS_R Angle	C103[oper.8bits][oper.24bits]	C1030000003F=B.ABS.R00006.3
	C.ABS_R Angle	C104[oper.8bits][oper.24bits]	C10400000159=C.ABS.R00034.5
STACKING	X.FIL_R Angle	C110[oper.8bits][oper.24bits]	C11000008ACF=X.FIL.R03453.5
	Y.FIL_R Angle	C111[oper.8bits][oper.24bits]	C11100030DE3=Y.FIL.R20016.3
	Z.FIL_R Angle	C112[oper.8bits][oper.24bits]	C11200000159=Z.FIL.R00034.5
RELATIVE	X.REL_R Angle	C120[oper.8bits][oper.24bits]	C12000000384=X.REL.R+90.0
	Y.REL_R Angle	C121[oper.8bits][oper.24bits]	C12180000320=Y.REL.R–90.0
	Z.REL_R Angle	C122[oper.8bits][oper.24bits]	C12200000320=Z.REL.R+80.0
	B.REL_R Angle	C123[oper.8bits][oper.24bits]	C12380000159=B.REL.R–34.5
	C.REL_R Angle	C124[oper.8bits][oper.24bits]	C1240000003F=C.REL.R+06.3

Type of Instruction	Display	Codop (hexadecimal)	Examples
CHECKING	X.CTL_R Angle	C130[oper.8bits][oper.24bits]	C13000000664=X.CTL.R00163.6
	Y.CTL_R Angle	C131[oper.8bits][oper.24bits]	C131000F423F=Y.CTL.R9999.9
	Z.CTL_R Angle	C132[oper.8bits][oper.24bits]	C13200000320=Z.CTL.R00080.0
	B.CTL_R Angle	C133[oper.8bits][oper.24bits]	C1330000003F=B.CTL.R00006.3
	C.CTL_R Angle	C134[oper.8bits][oper.24bits]	C1340000050C=C.CTL.R00150.0
		Marker No SAP Angle in 1/10 deg.	
TEACHING	⊔ ⊔ ⊔ Teach	C 00AAAAAA	C01000AAAAAA=X.FIL.Learning
	Previous instruction	Instruction code	C10200AAAAAA=Z.ABS.RLearning
MOTORIZED MOTIONS (cont.)			
LINEAR			
ABSOLUTE	X.ABS_L WW00 (to 65)	C200 [oper. 16 bits]	C200000A = X.ABS.L WW10
(Words)	Y.ABS_L WW00 (to 65)	C201 [oper. 16 bits]	
	Z.ABS_L WW00 (to 65)	C202 [oper. 16 bits]	
	B.ABS_L WW00 (to 65)	C203 [oper. 16 bits]	
	C.ABS_L WW00 (to 65)	C204 [oper. 16 bits]	
STACKING	X.FIL_L WW00 (to 65)	C210 [oper. 16 bits]	C210000B = X.FIL.L WW11
	Y.FIL_L WW00 (to 65)	C211 [oper. 16 bits]	
	Z.FIL_L WW00 (to 65)	C212 [oper. 16 bits]	
RELATIVE	X.REL_L WW00 (to 65)	C220 [oper. 16 bits]	C2200041 = X.REL.L WW65
	Y.REL_L WW00 (to 65)	C221 [oper. 16 bits]	
	Z.REL_L WW00 (to 65)	C222 [oper. 16 bits]	
	B.REL_L WW00 (to 65)	C223 [oper. 16 bits]	
	C.REL_L WW00 (to 65)	C224 [oper. 16 bits]	
CHECKING	X.CTL_L WW00 (to 65)	C230 [oper. 16 bits]	C2300010 = X.CTL.L WW16
	Y.CTL_L WW00 (to 65)	C231 [oper. 16 bits]	
	Z.CTL_L WW00 (to 65)	C232 [oper. 16 bits]	
	B.CTL_L WW00 (to 65)	C233 [oper. 16 bits]	
	C.CTL_L WW00 (to 65)	C234 [oper. 16 bits]	
ROTATIVE			
ABSOLUTE	X.ABS_R WW00 (to 15)	C300 [oper. 16 bits]	C300000A = X.ABS.R WW10
(Words)	Y.ABS_R WW00 (to 15)	C301 [oper. 16 bits]	
	Z.ABS_R WW00 (to 15)	C302 [oper. 16 bits]	
	B.ABS_R WW00 (to 15)	C303 [oper. 16 bits]	
	C.ABS_R WW00 (to 15)	C304 [oper. 16 bits]	
STACKING	X.FIL_R WW00 (to 15)	C310 [oper. 16 bits]	C3100020 = X.FIL.R WW32
	Y.FIL_R WW00 (to 15)	C311 [oper. 16 bits]	
	Z.FIL_R WW00 (to 15)	C312 [oper. 16 bits]	

Type of Instruction	Display	Codop (hexadecimal)	Examples
RELATIVE	X.REL_R WW00 (to 15)	C320 [oper. 16 bits]	C3200001 = X.REL.R WW01
	Y.REL_R WW00 (to 15)	C321 [oper. 16 bits]	
	Z.REL_R WW00 (to 15)	C322 [oper. 16 bits]	
	B.REL_R WW00 (to 15)	C323 [oper. 16 bits]	
	C.REL_R WW00 (to 15)	C324 [oper. 16 bits]	
CHECKING	X.CTL_R WW00 (to 15)	C330 [oper. 16 bits]	C3300041 = X.CTL.R WW65
	Y.CTL_R WW00 (to 15)	C331 [oper. 16 bits]	
	Z.CTL_R WW00 (to 15)	C332 [oper. 16 bits]	
	B.CTL_R WW00 (to 15)	C333 [oper. 16 bits]	
	C.CTL_R WW00 (to 15)	C334 [oper. 16 bits]	
		WWORD No.	
FREE	X. FREE	C040	
	Y. FREE	C041	
	Z. FREE	C042	
	B. FREE	C043	
	C. FREE	C044	
LINE	LIN.	B046	

Type of Instruction	Display	Codop (hexadecimal)	Examples
<u>**LINEAR**</u>			

POS_ANA	X = POS ANA + distance
	Y = POS ANA + distance
	Z = POS ANA + distance
	B = POS ANA + distance
	C = POS ANA + distance
		C060 [oper. 32 bits]
		C061 [oper. 32 bits]
		C062 [oper. 32 bits]
		C063 [oper. 32 bits]
		C064 [oper. 32 bits]	

POS_NUM	X = POS NUM + distance
	Y = POS NUM + distance
	Z = POS NUM + distance
	B = POS NUM + distance
	C = POS NUM + distance
		C070 [oper. 32 bits]
		C071 [oper. 32 bits]
		C072 [oper. 32 bits]
		C073 [oper. 32 bits]
		C074 [oper. 32 bits]	

VEL ANA NORMAL	X = VEL ANA_N + distance
	Y = VEL ANA_N + distance
	Z = VEL ANA_N + distance
	B = VEL ANA_N + distance
	C = VEL ANA_N + distance
		C080 [oper. 32 bits]
		C081 [oper. 32 bits]
		C082 [oper. 32 bits]
		C083 [oper. 32 bits]
		C084 [oper. 32 bits]	

VEL ANA INTEGRAL	X = VEL ANA_I + distance
	Y = VEL ANA_I + distance
	Z = VEL ANA_I + distance
	B = VEL ANA_I + distance
	C = VEL ANA_I + distance
		C090 [oper. 32 bits]
		C091 [oper. 32 bits]
		C092 [oper. 32 bits]
		C093 [oper. 32 bits]
		C094 [oper. 32 bits]	

VEL NUM NORMAL	X = VEL NUM_N + distance
	Y = VEL NUM_N + distance
	Z = VEL NUM_N + distance
	B = VEL NUM_N + distance
	C = VEL NUM_N + distance
		C0A0[oper. 32 bits]
		C0A1 [oper. 32 bits]
		C0A2 [oper. 32 bits]
		C0A3 [oper. 32 bits]
		C0A4 [oper. 32 bits]	

VEL NUM INTEGRAL	X = VEL NUM_I + distance
	Y = VEL NUM_I + distance
	Z = VEL NUM_I + distance
	B = VEL NUM_I + distance
	C = VEL NUM_I + distance
		C0B0[oper. 32 bits]
		C0B1 [oper. 32 bits]
		C0B2 [oper. 32 bits]
		C0B3 [oper. 32 bits]
		C0B4 [oper. 32 bits]	

Type of Instruction	Display	Codop (hexadecimal)	Examples
<u>**ROTATIVE**</u>			
POS_ANA	X = POS ANA + angle	C160 [oper. 32 bits]	
	Y = POS ANA + angle	C161 [oper. 32 bits]	
	Z = POS ANA + angle	C162 [oper. 32 bits]	
	B = POS ANA + angle	C163 [oper. 32 bits]	
	C = POS ANA + angle	C164 [oper. 32 bits]	
POS_NUM	X = POS NUM + angle	C170 [oper. 32 bits]	
	Y = POS NUM + angle	C171 [oper. 32 bits]	
	Z = POS NUM + angle	C172 [oper. 32 bits]	
	B = POS NUM + angle	C173 [oper. 32 bits]	
	C = POS NUM + angle	C174 [oper. 32 bits]	
VEL ANA NORMAL	X = VEL ANA_N + angle	C180 [oper. 32 bits]	
	Y = VEL ANA_N + angle	C181 [oper. 32 bits]	
	Z = VEL ANA_N + angle	C182 [oper. 32 bits]	
	B = VEL ANA_N + angle	C183 [oper. 32 bits]	
	C = VEL ANA_N + angle	C184 [oper. 32 bits]	
VEL ANA INTEGRAL	X = VEL ANA_I + angle	C190 [oper. 32 bits]	
	Y = VEL ANA_I + angle	C191 [oper. 32 bits]	
	Z = VEL ANA_I + angle	C192 [oper. 32 bits]	
	B = VEL ANA_I + angle	C193 [oper. 32 bits]	
	C = VEL ANA_I + angle	C194 [oper. 32 bits]	
VEL NUM NORMAL	X = VEL NUM_N + angle	C1A0[oper. 32 bits]	
	Y = VEL NUM_N + angle	C1A1 [oper. 32 bits]	
	Z = VEL NUM_N + angle	C1A2 [oper. 32 bits]	
	B = VEL NUM_N + angle	C1A3 [oper. 32 bits]	
	C = VEL NUM_N + angle	C1A4 [oper. 32 bits]	
VEL NUM INTEGRAL	X = VEL NUM_I + angle	C1B0[oper. 32 bits]	
	Y = VEL NUM_I + angle	C1B1 [oper. 32 bits]	
	Z = VEL NUM_I + angle	C1B2 [oper. 32 bits]	
	B = VEL NUM_I + angle	C1B3 [oper. 32 bits]	
	C = VEL NUM_I + angle	C1B4 [oper. 32 bits]	

Type of Instruction	Display	Codop (hexadecimal)	Examples
<u>CONDITIONS TEST</u>			
. 1 Operand			
on Bit	IF BIT 000 (to 127)	D000 [oper. 16 bits]	
	IF/BIT 000 (to 127)	D010 [oper. 16 bits]	
on Output	IF OUT 000 (to 127)	D001 [oper. 16 bits]	
	IF/OUT 000 (to 127)	D011 [oper. 16 bits]	
on Input	IF IN/000 (to 127)	D002 [oper. 16 bits]	
	IF IN 000 (to 127)	D003 [oper. 16 bits]	
	IF/IN 000 (to 127)	D012 [oper. 16 bits]	
on Timer	IF TIME 00 (to 15)	D004 [oper. 16 bits]	
	IF/TIME 00 (to 15)	D014 [oper. 16 bits]	
		↓ Operand No.	
. 2 Operands			
* on Word (16 bits)	IF WRD 000 (to 4095)	D300 [oper. 16 bits]	
–> 1st Operand	IF/WRD 000 (to 4095)	D310 [oper. 16 bits]	
with decimal value	= 0000 (to 9999)	D400 [oper. 16 bits]	*<u>Note</u>*: If the decimal value cannot exceed 9,999, the hexadecimal value goes up to 65,535.
	> = 0000 (to 9999)	D401 [oper. 16 bits]	
	< = 0000 (to 9999)	D402 [oper. 16 bits]	
	AND 0000 (to 9999)	D403 [oper. 16 bits]	
with hexadecimal value	= 0000 (to FFFF)	D410 [oper. 16 bits]	
	> = 0000 (to FFFF)	D411 [oper. 16 bits]	
	< = 0000 (to FFFF)	D412 [oper. 16 bits]	
	AND 0000 (to FFFF)	D413 [oper. 16 bits]	
with Counter	= 00 (to 15)	D420 [oper. 16 bits]	
	> = 00 (to 15)	D421 [oper. 16 bits]	
	< = 00 (to 15)	D422 [oper. 16 bits]	
	AND 00 (to 15)	D423 [oper. 16 bits]	
with Inputs (modulo 16)	= 000 (to 112)	D430 [oper. 16 bits]	
	> = 000 (to 112)	D431 [oper. 16 bits]	
	< = 000 (to 112)	D432 [oper. 16 bits]	
	AND 000 (to 112)	D433 [oper. 16 bits]	
with Word (16 bits)	= 0000 (to 4095)	D440 [oper. 16 bits]	
	> = 0000 (to 4095)	D441 [oper. 16 bits]	
	< = 0000 (to 4095)	D442 [oper. 16 bits]	
	AND 0000 (to 4095)	D443 [oper. 16 bits]	

Type of Instruction	Display	Codop (hexadecimal)	Examples
* on Word (32 bits)	IF WWRD 000 (to 127)	D320 [oper. 16 bits]	
–> 1st Operand	IF/WWRD 000 (to 127)	D330 [oper. 16 bits]	
with decimal value	= 00000000 (to 09999999)	D500 [oper. 32 bits]	*Note*: If the decimal value cannot exceed 9,999,999, the hexadecimal value goes up to 4,294,967,295.
	> = 00000000 (to 09999999)	D501 [oper. 32 bits]	
	< = 00000000 (to 09999999)	D502 [oper. 32 bits]	
	AND 00000000 (to 09999999)	D503 [oper. 32 bits]	
with hexadecimal value	= 00000000 (to FFFFFFFF)	D510 [oper. 32 bits]	
	> = 00000000 (to FFFFFFFF)	D511 [oper. 32 bits]	
	< = 00000000 (to FFFFFFFF)	D512 [oper. 32 bits]	
	AND00000000 (to FFFFFFFF)	D513 [oper. 32 bits]	
with Counter	= 00 (to 15)	D520 [oper. 16 bits]	
	> = 00 (to 15)	D521 [oper. 16 bits]	
	< = 00 (to 15)	D522 [oper. 16 bits]	
	AND 00 (to 15)	D523 [oper. 16 bits]	
with Inputs (modulo 16)	= 000 (to 112)	D530 [oper. 16 bits]	
	> = 000 (to 112)	D531 [oper. 16 bits]	
	< = 000 (to 112)	D532 [oper. 16 bits]	
	AND 000 (to 112)	D533 [oper. 16 bits]	
with Word (16 bits)	= 0000 (to 4095)	D540 [oper. 16 bits]	
	> = 0000 (to 4095)	D541 [oper. 16 bits]	
	< = 0000 (to 4095)	D542 [oper. 16 bits]	
	AND 0000 (to 4095)	D543 [oper. 16 bits]	
with WWord (32 bits)	= 000 (to 127)	D550 [oper. 16 bits]	
	> = 000 (to 127)	D551 [oper. 16 bits]	
	< = 000 (to 127)	D552 [oper. 16 bits]	
	AND 000 (to 127)	D553 [oper. 16 bits]	
* on Counter	IF CPT 00 (to 15)	D340 [oper. 16 bits]	
–> 1st Operand	IF/CPT 00 (to 15)	D350 [oper. 16 bits]	
with decimal value	= 0000 (to 9999)	D900 [oper. 16 bits]	
	> = 0000 (to 9999)	D901 [oper. 16 bits]	
	< = 0000 (to 9999)	D902 [oper. 16 bits]	
	AND 0000 (to 9999)	D903 [oper. 16 bits]	
with hexadecimal value	= 0000 (to FFFF)	D910 [oper. 16 bits]	
	> = 0000 (to FFFF)	D911 [oper. 16 bits]	
	< = 0000 (to FFFF)	D912 [oper. 16 bits]	
	AND 0000 (to FFFF)	D913 [oper. 16 bits]	
with Counter	= 00 (to 15)	D920 [oper. 16 bits]	
	> = 00 (to 15)	D921 [oper. 16 bits]	
	< = 00 (to 15)	D922 [oper. 16 bits]	
	AND 00 (to 15)	D923 [oper. 16 bits]	

Type of Instruction	Display	Codop (hexadecimal)	Examples
with Inputs (modulo 16)	= 000 (to 112) > = 000 (to 112) < = 000 (to 112) AND 000 (to 112)	D930 [oper. 16 bits] D931 [oper. 16 bits] D932 [oper. 16 bits] D933 [oper. 16 bits]	
with Word (16 bits)	= 0000 (to 4095) > = 0000 (to 4095) < = 0000 (to 4095) AND 0000 (to 4095)	D940 [oper. 16 bits] D941 [oper. 16 bits] D942 [oper. 16 bits] D943 [oper. 16 bits]	
INITIALIZATION			
. 1 Operand			
* on Bit –> 1 on Bit –> 0	SET.BIT 032 (to 127) RST.BIT 032 (to 127)	D015 [oper. 16 bits] D017 [oper. 16 bits]	
* on Output –> 1 on Output –> 0	SET.OUT 000 (to 127) RST.OUT 000 (to 127)	D016 [oper. 16 bits] D018 [oper. 16 bits]	
* on Word –> 0	RST.WRD0000 (to 4095)	D019 [oper. 16 bits] ↓ Variable No.	
* on Counter –> 0	RST.CPT 0000 (to 0015)	D01A 00 [oper. 8 bits] ↓ Counter No.	
	RST.CPT 0041 (to 9980)	D01A[oper. 8 bits] [oper. 8 bits] ↓ MP No. ↓ SP No.	
. 2 Operands			
* on Word (16 bits) –> 1st Operand	SET.WRD 0000 (to 4095	)D600 [oper. 16 bits]	
with numeric value	= 0000 (to 9999) + 0000 (to 9999) – 0000 (to 9999) x 0000 (to 9999) / 0000 (to 9999) AND 0000 (to 9999) OR 0000 (to 9999)	D700 [oper. 16 bits] D701 [oper. 16 bits] D702 [oper. 16 bits] D703 [oper. 16 bits] D704 [oper. 16 bits] D705 [oper. 16 bits] D706 [oper. 16 bits]	
with hexadecimal value	= 0000 (to FFFF) + 0000 (to FFFF) – 0000 (to FFFF) x 0000 (to FFFF) / 0000 (to FFFF) AND 0000 (to FFFF) OR 0000 (to FFFF)	D710 [oper. 16 bits] D711 [oper. 16 bits] D712 [oper. 16 bits] D713 [oper. 16 bits] D714 [oper. 16 bits] D715 [oper. 16 bits] D716 [oper. 16 bits]	

Type of Instruction	Display	Codop (hexadecimal)	Examples
with Counter	= 00 (to 15)	D720 [oper. 16 bits]	
	+ 00 (to 15)	D721 [oper. 16 bits]	
	– 00 (to 15)	D722 [oper. 16 bits]	
	x 00 (to 15)	D723 [oper. 16 bits]	
	/ 00 (to 15)	D724 [oper. 16 bits]	
	AND 00 (to 15)	D725 [oper. 16 bits]	
	OR 00 (to 15)	D726 [oper. 16 bits]	
with Inputs (modulo 16)	= 000 (to 112)	D730 [oper. 16 bits]	
	+ 000 (to 112)	D731 [oper. 16 bits]	
	– 000 (to 112)	D732 [oper. 16 bits]	
	x 000 (to 112)	D733 [oper. 16 bits]	
	/ 000 (to 112)	D734 [oper. 16 bits]	
	AND 000 (to 112)	D735 [oper. 16 bits]	
	OR 000 (to 112)	D736 [oper. 16 bits]	
with Word (16 bits)	= 0000 (to 4095)	D740 [oper. 16 bits]	
	+ 0000 (to 4095)	D741 [oper. 16 bits]	
	– 0000 (to 4095)	D742 [oper. 16 bits]	
	x 0000 (to 4095)	D743 [oper. 16 bits]	
	/ 0000 (to 4095)	D744 [oper. 16 bits]	
	AND 0000 (to 4095)	D745 [oper. 16 bits]	
	OR 0000 (to 4095)	D746 [oper. 16 bits]	
* on Word (32 bits) –> 1st Operand	SET.WWRD 000 (to 127)	D620 [oper. 16 bits]	
with decimal value	= 00000000 (to 09999999)	D800 [oper. 32 bits]	
	+ 00000000 (to 09999999)	D801 [oper. 32 bits]	
	– 00000000 (to 09999999)	D802 [oper. 32 bits]	
	x 00000000 (to 09999999)	D803 [oper. 32 bits]	
	/ 00000000 (to 09999999)	D804 [oper. 32 bits]	
	AND 00000000 (to 09999999)	D805 [oper. 32 bits]	
	OR 00000000 (to 09999999)	D806 [oper. 32 bits]	
with hexadecimal value	= 00000000 (to FFFFFFFF)	D810 [oper. 32 bits]	
	+ 00000000 (to FFFFFFFF)	D811 [oper. 32 bits]	
	– 00000000 (to FFFFFFFF)	D812 [oper. 32 bits]	
	x 00000000 (to FFFFFFFF)	D813 [oper. 32 bits]	
	/ 00000000 (to FFFFFFFF)	D814 [oper. 32 bits]	
	AND 00000000 (to FFFFFFFF)	D815 [oper. 32 bits]	
	OR 00000000 (to FFFFFFFF)	D816 [oper. 32 bits]	
with Counter	= 00 (to 15)	D820 [oper. 16 bits]	
	+ 00 (to 15)	D821 [oper. 16 bits]	
	– 00 (to 15)	D822 [oper. 16 bits]	
	x 00 (to 15)	D823 [oper. 16 bits]	
	/ 00 (to 15)	D824 [oper. 16 bits]	
	AND 00 (to 15)	D825 [oper. 16 bits]	
	OR 00 (to 15)	D826 [oper. 16 bits]	

Type of Instruction	Display	Codop (hexadecimal)	Examples
with Inputs (modulo 16)	= 000 (to 112)	D830 [oper. 16 bits]	
	+ 000 (to 112)	D831 [oper. 16 bits]	
	– 000 (to 112)	D832 [oper. 16 bits]	
	x 000 (to 112)	D833 [oper. 16 bits]	
	/ 000 (to 112)	D834 [oper. 16 bits]	
	AND 000 (to 112)	D835 [oper. 16 bits]	
	OR 000 (to 112)	D836 [oper. 16 bits]	
with Word (16 bits)	= 0000 (to 4095)	D840 [oper. 16 bits]	
	+ 0000 (to 4095)	D841 [oper. 16 bits]	
	– 0000 (to 4095)	D842 [oper. 16 bits]	
	x 0000 (to 4095)	D843 [oper. 16 bits]	
	/ 0000 (to 4095)	D844 [oper. 16 bits]	
	AND 0000 (to 4095)	D845 [oper. 16 bits]	
	OR 0000 (to 4095)	D846 [oper. 16 bits]	
with Word (32 bits)	= 000 (to 112)	D850 [oper. 16 bits]	
	+ 000 (to 112)	D851 [oper. 16 bits]	
	– 000 (to 112)	D852 [oper. 16 bits]	
	x 000 (to 112)	D853 [oper. 16 bits]	
	/ 000 (to 112)	D854 [oper. 16 bits]	
	AND 000 (to 112)	D855 [oper. 16 bits]	
	OR 000 (to 112)	D856 [oper. 16 bits]	
* on Counter	SET.CPT 0000 (to 0015)	D640 [oper. 8 bits]	Standard counter
–> 1st Operand	SET.CPT 0041 (to 9980)	D640[oper. 8 bits] [oper. 8 bits]	Stacking counter
		MP No. SP No.	
with decimal value	= 0000 (to 9999)	DA00 [oper. 16 bits]	
	+ 0000 (to 9999)	DA01 [oper. 16 bits]	
	– 0000 (to 9999)	DA02 [oper. 16 bits]	
	x 0000 (to 9999)	DA03 [oper. 16 bits]	
	/ 0000 (to 9999)	DA04 [oper. 16 bits]	
	AND 0000 (to 9999)	DA05 [oper. 16 bits]	
	OR 0000 (to 9999)	DA06 [oper. 16 bits]	
with hexadecimal value	= 0000 (to FFFF)	DA10 [oper. 16 bits]	
	+ 0000 (to FFFF)	DA11 [oper. 16 bits]	
	– 0000 (to FFFF)	DA12 [oper. 16 bits]	
	x 0000 (to FFFF)	DA13 [oper. 16 bits]	
	/ 0000 (to FFFF)	DA14 [oper. 16 bits]	
	AND 0000 (to FFFF)	DA15 [oper. 16 bits]	
	OR 0000 (to FFFF)	DA16 [oper. 16 bits]	
with Counter	= 00 (to 15)	D920 [oper. 16 bits]	
	+ 00 (to 15)	D921 [oper. 16 bits]	
	– 00 (to 15)	D922 [oper. 16 bits]	
	x 00 (to 15)	D922 [oper. 16 bits]	
	/ 00 (to 15)	D922 [oper. 16 bits]	
	AND 00 (to 15)	D923 [oper. 16 bits]	
	OR 00 (to 15)	D923 [oper. 16 bits]	

Type of Instruction	Display	Codop (hexadecimal)	Examples
with Inputs (modulo 16)	= 000 (to 112)	DA30 [oper. 16 bits]	
	+ 000 (to 112)	DA31 [oper. 16 bits]	
	– 000 (to 112)	DA32 [oper. 16 bits]	
	x 000 (to 112)	DA33 [oper. 16 bits]	
	/ 000 (to 112)	DA34 [oper. 16 bits]	
	AND 000 (to 112)	DA35 [oper. 16 bits]	
	OR 000 (to 112)	DA36 [oper. 16 bits]	
with Word (16 bits)	= 0000 (to 4095)	DA40 [oper. 16 bits]	
	+ 0000 (to 4095)	DA41 [oper. 16 bits]	
	– 0000 (to 4095)	DA42 [oper. 16 bits]	
	x 0000 (to 4095)	DA43 [oper. 16 bits]	
	/ 0000 (to 4095)	DA44 [oper. 16 bits]	
	AND 0000 (to 4095)	DA45 [oper. 16 bits]	
	OR 0000 (to 4095)	DA46 [oper. 16 bits]	
–> + 1	INC.CPT 0000 (to 0015)	D01B 00 [oper. 8 bits]	
		Standard No.	
	INC.CPT 0041 (to 9980)	D01B[oper. 8 bits] [oper. 8 bits]	
		MP No. SP No.	
–> – 1	DEC.CPT 0000 (to 0015)	D01C 00 [oper. 8 bits]	
		Standard No.	
	DEC.CPT 0041 (to 9980)	D01C[oper. 8 bits] [oper. 8 bits]	
		MP No. SP No.	

II – 2. PLC PROGRAMS

Type of Instruction	Display	Codop (hexadecimal)
Header PROG.PLC xx (num)	PLC xx	FC [oper. 16 bits] PLC No.
CONDITION TEST	IF ...	See part programs
INITIALIZATION	SET ... RST ... INC ... DEC ...	See part programs
COMPARISON xxxx > = xxxx	CMP 0000 (to 0015) VAL 0000 (to FFFF) 0000 (to 0015)	D020 [oper. 16 bits] [oper. 16 bits] Counter No. Value
TIMER xx VALUE xxxx	TIMER 00 (to 15) VAL 0000 (to 9999)	D021 [oper. 16 bits] [oper. 16 bits] Timer No. Preselection value
AND FUNCTION on BIT	AND BIT 000 (to 127)	D022 [oper. 16 bits]
AND FUNCTION on OUTPUT	AND OUT 000 (to 127)	D023 [oper. 16 bits]
OR FUNCTION on BIT	OR BIT 000 (to 127)	D024 [oper. 16 bits]
OR FUNCTION on OUTPUT	OR OUT 000 (to 127)	D025 [oper. 16 bits] Variables No.
END OF PROGRAM	END	F5 [oper. 16 bits] PLC No.

III – PROGRAM CODES

III – 1. DECLARATION OF PROGRAMS, SUBROUTINES AND PLC

* Header codes of MP, SP,..., SR, PLC

F9nn	= Main program
FAnn	= Subroutine STD, FIL... // ... (See stacking header)
FBnn	= Return subroutine (See Home Return header)
FCnn	= PLC program
FEnn	= Free

* STEP TRANSITION codes

EC00 + Step 0 to 999 number

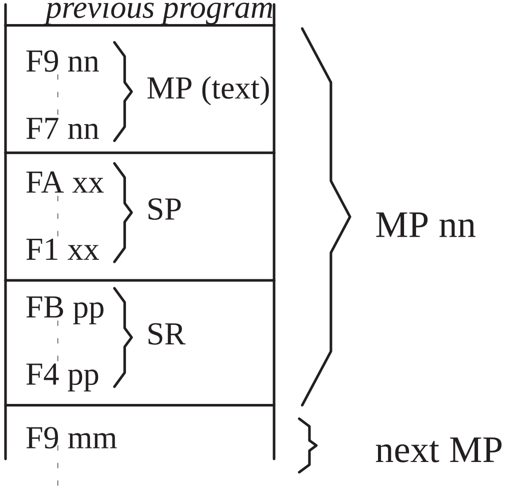
E.g. : EC12 => Step Number 18 (decimal)

E.g. : ED00 => Step Number 256 (decimal)

* END codes of MP, SP..., SR, PLC

F0nn	= End of SP: ”standard” nn.
F1nn	= End of SP: ”standard” stacking nn.
F2nn	= End of SP: ”general” stacking nn.
F3nn	= End of SP: // nn.
F4nn	= End of SR: simple or total nn.
F8nn	= End of SR: simple or total with return to step 0 of PP 00.
F5nn	= End of PLC program nn.
F7nn	= End of Main Program (MP) nn.

* Main Program architecture in the memory area



III – 2. SUBROUTINE AND PROGRAM CALLS

* SPECIFIC codes for SP, SR, PLC as instruction

– E000 [oper. 16 bits] :

. *Standard SP:* SP nn Lmm (nn = 01 to 40) (mm = 00 to 99)

. *Regular stacking SP:* SP nn D Lmm (or I Lmm) (nn = 41 to 60) (mm = 00 to 99)

. *General stacking SP:* SP nn D Lmm (or I Lmm) (nn = 61 to 80) (mm = 00 to 99)

. *Parallel SP:* SP nn L00 (nn = 81 to 99)

The operand contains:

. high order word–> LABEL number
–> bit 0 x 8000 at 0 indicates DIRECT
–> bit 0 x 8000 at 1 indicates REVERSE

. low order word–> SP number.

E.g.: E000 0103 –> SP 03 L01

E.g.: E000 8229 –> SP 41 I L02

– E000 [oper. 16 bits]: PLC prog. – Display: PLC 00 (to 99)

– E500 [oper. 16 bits]: Home Return – Display: SR 01 (to 99)

* Return Label

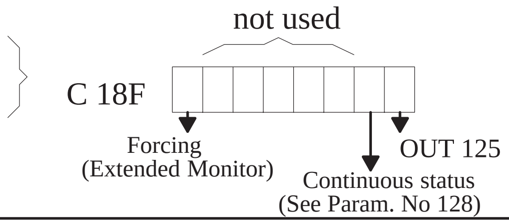
– E600 [oper. 16 bits]: Labels ”L” for SP – Display: L00 to L99

– E700 [oper. 16 bits]: Labels ”R” for SR – Display: R00 to R99

IV – VARIABLE ADDRESSING

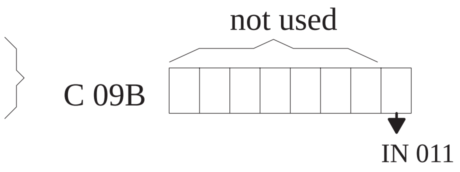
IV – 1. <u>OUTPUT – OUT –</u>

Accessible in read and write.

Number (logical address)	Physical address	Structures / Functions
OUT 000 ↓ OUT 127	C 111 ↓ C 191	

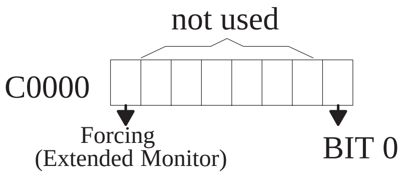
IV – 2. <u>INPUT – IN –</u>

Accessible in read.

Number (logical address)	Physical address	Structures / Functions
IN 000 ↓ IN 127	C 090 ↓ C 110	

IV – 3. <u>USER AND SYSTEM BITS – BIT –</u>

Each address corresponds to an 8 bit structure in memory.



Only the low order word is used.

– System bits accessible in read.

Number (logical address)	Physical address	Action
0	C000	< > 0 to indicate the reverse execution for stacking
1	C001	< > 0 to indicate the end of layer or column
2	C002	< > 0 to indicate the end of pallet
3	C003	< > 0 to indicate the odd row in progress
4	C004	< > 0 to indicate the odd column in progress
5	C005	< > 0 to indicate the odd layer in progress
6	C006	< > 0 Number of odd part in progress
7	C007	< > 0 Stacking (REG or GEN) in progress
8	C008	Reserved
9	C009	< > 0 TOTAL Home Return in progress
10	C00A	< > 0 Parallel subroutine in progress
11	C00B	Reserved
12	C00C	< > 0 Commands by external inputs (control board)
13	C00D	Reserved
14	C00E	“Following” axe attached
15	C00F	Reserved
16	C010	Bit 0 of rotation word of general stackings
17	C011	Bit 1 of rotation word of general stackings
18	C012	Bit 2 of rotation word of general stackings
19	C013	Bit 3 of rotation word of general stackings
20	C014	Bit 4 of rotation word of general stackings
21	C015	Bit 5 of rotation word of general stackings
22	C016	Bit 6 of rotation word of general stackings
23	C017	Bit 7 of rotation word of general stackings
24 to 30	C018 to C01E	Reserved

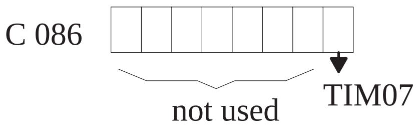
– System bits accessible in read and write.

Number (logical address)	Physical address	Action
31	C01F	External validation of an automatic program change can only be positioned by Host via protocol.
32	C020	Bit: part grips result (replaces E_PP)
33	C021	Bit: Validation of automatic change of program by Host
34 to 127	C022 to C07F	Bit: User bits accessible with the part program or PLC program

IV – 4. TIMERS

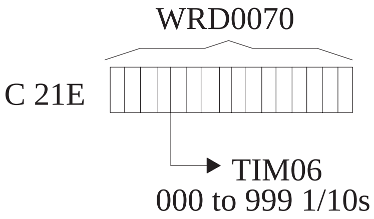
IV – 4. 1. End of timer for part program – TIM –

Accessible in read and write.

Nuumber (logical address)	Physical address	Structures / Functions
TIM00	C080	 Only the low order word is used.
TIM01	C081	
TIM02	C082	
TIM03	C083	
TIM04	C084	
TIM05	C085	
TIM06	C086	
TIM07	C087	
TIM08	C088	
TIM09	C089	
TIM10	C08A	
TIM11	C08B	
TIM12	C08C	
TIM13	C08D	
TIM14	C08E	
TIM15	C08F	

IV – 4. 2. PLC timer – TIMER –

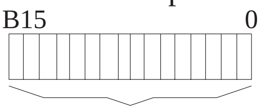
Accessible in read and write.

Number (logical address)	Indirection or logical address (on a Word)	Physical address	Structures / Functions
TIM00	WRD 0064	C230	
TIM01	WRD 0065	C214	
TIM02	WRD 0066	C216	
TIM03	WRD 0067	C218	
TIM04	WRD 0068	C21A	
TIM05	WRD 0069	C21C	
TIM06	WRD 0070	C21E	
TIM07	WRD 0071	C220	
TIM08	WRD 0072	C222	
TIM09	WRD 0073	C224	
TIM10	WRD 0074	C226	
TIM11	WRD 0075	C228	
TIM12	WRD 0076	C22A	
TIM13	WRD 0077	C22C	
TIM14	WRD 0078	C22E	
TIM15	WRD 0079	C230	

IV – 5. WORDS: 16 BIT USER AND SYSTEM– WRD –

IV – 5. 1. User words – WRD –

Accessible in read and write.

Number (logical address)	Physical address	Structures / Functions
WRD 0000 ↓ WRD 0031	C192 ↓ C1D0	32 Words: 16 user bits with no predefined functions.  16 bit structure available
WRD 0064 ↓ WRD 0079	C212 ↓ C230	16 Words: 16 bits supporting the PLC timers (TIM 00 to TIM 15).
WRD 0080 ↓ WRD 0095	C232 ↓ C250	16 Words: 16 bits supporting the standard counters (CPT 00 to CPT 15).
WRD 0096 ↓ WRD 4096	C252 ↓ E192	4000 Words: 16 bits supporting the stacking subroutines counters (CPT 0041 to CPT 9980).

IV – 5. 2. System words

Accessible in read.

No	Physical address	Name	Description
32	C1D2	MotMode	Contains the mode selector position
33	C1D4	MotArret	Contains the immediate stopping or safety request
34	C1D6	MotSta1	
35	C1D8	MotSta2	Contains the system status
36	C1DA	MotSta3	
37			Reserved
38	C1DE	MotSec1	Status of inputs such as ZBD, Power, BH, etc.
39	C1E0	MotSec1	Status of inputs for balance and coherence
40	C1E2	MotExe1	Used by the execution manager to synchronize
41	C1E4	MotExe2	the various tasks required.
42	C1E6	MotReg1	Motion keys status
43	C1E8	MotReg2	V+, V–, X+, X–, etc.

No	Physical address	Name	Description
44	C1EA	MotDio1	Used for managing the terminal LEDs.
45	C1EC	Klaxon	Alarm 1 = ON / 2 = OFF
46	C1EE	MotDial	Signals used for machine dialogue (MO, PF,...)
47	C1F0	PrgExec	Number of MP in progress
48	C1F2	SP_Exec	Code of SP or SR in progress (e.g.: FA01 = SP01)
49	C1F4	PlcExec	Number of PLC in progress
50	C1F6	TraExec	Reserved
51	C1F8	PasExec	Number of MP or SP or SR STEP in progress
52	C1FA	Version	Number of software version with or without FIL GEN
53	C1FC	Config	Contains the installed options (PAL of Dial board)
54	C1FE	Cfg_I_O	Number of modulo 16 inputs/outputs and axis mother board
55	C200		Not used
56	C202		Not used
57	C204	MotCod2	Not used

– Detail of system words accessible in reading:

The IF and SET ... WRD instructions can be used (in 1st Operand)

– 32 – MotMode (MD)

Contains mode selector–switch information.

80	40	20	10	8	4	2	1
–	–	–	–	EN–REGL	EN–SEMI	EN–AUTO	EN–ARRET

EN–ARRET : Selector switch set to Stop.

EN–AUTO : Same for AUTO.

EN–SEMI : Same for SEMI–AUTO.

EN–REGL : Same for ADJUST.

– 33 – MotArret (AR)

Used for the unconditional stop requests.

						2	1
–	–	–	–	–	–	SECU	STOP

STOP : General request for immediate stop. (At 1 for example if START key is released in SEMI–AUTO mode during execution of a step).

SECU : General request for safety stop. (E.g.: power OFF)

– 34 – MotSta1 (E1)

Indicates the real mode in progress.

80	40	20	10	8	4	2	1
–	–	–	–	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.

Notes:

Although it reflects the MotMode word, this word is updated by operating modes, taking into account the Real mode in progress.

e.g.: Selector switch set to SEMI–AUTO but robot is executing an RO or PCO that have special types of operation.

– 35 – MotSta2

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.

80	40	20	10	8	4	2	1
EN–DEF	EN–WARN	–	EN–POINT	EN–ETEND	EN–VISU	EN–PROG	EN–EXEC

EN–EXEC : Indicator of execution mode in progress.

EN–PROG : Same for programming in progress.

EN–VISU : Same for display mode.

EN–ETEND : Same for extended monitor.

EN–POINT : Same for display point of the position in progress.

EN–WARN : Same for warning in progress.

EN–DEF : Same for robot in error.

1000	800	400	200	100
EN–TEST	EN–HIST	EN–PROD	EN–MAINT	EN–HELP

EN–HELP : Indicator of help mode (fault comments).

EN–MAINT : Same for maintenance work in progress.

EN–PROD : Same for production work.

EN–HIST : Same for fault historic work.

EN–TEST : Same for output test or waiting bits function.

– 36 – MotSta3

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.

80	40	20	10	8	4	2	1
–	RO–AUTO	RO–POINT	RO–TOTAL	RO–SIMPL	RO–PCO	EN–ARFCY	EN–CYCLE

EN–CYCLE : Same for robot in cycle (after pressing START).

EN–ARFCY : Same for end of auto cycle in progress request.

EN–PCO : Same for PCO procedure in progress.

RO–SIMPL : Same for simple RO procedure in progress.

RO–TOTAL : Same for total return.

RO–POINT : Same for return to previous point.

RO–AUTO : Same for automatic simple return (part grip fault).

– 38 – MotSec1 (S1)

Contains the power states of the inputs dedicated to the safety devices.

80	40	20	10	8	4	2	1
SLA	BRASHAUT	ZBD	AM–MO	HOMME–OK	SUSPEND	VARS–OK	PUISS–OK

PUISS–OK : Set to 1 if power is rearmed.

VARS–OK : Set to 1 if the velocity adjustors are in working order

SUSPEND : Input image (or command) of cycle suspension (0 = suspend cycle / 1 = continue cycle).

HOMME–OK : Set to 1 if the ”Dead Man” button is pressed or if the terminal is in place.

AM–MO : Set to 1 if the AM and MO signals are present.

ZBD : Same as for ZBD cam.

BRASHAUT : Same if the arm is in the high, safety position.

SLA : Set to 0 if the SLA contact is pressed (normally set to 1).

– 39 – MotSec2 (S2)

80	40	20	10	8	4	2	1
–	–	–	–	–	–	PRES–OK	COHERENCE

PRES–OK : Set to 1 if the air pressure is correct.

COHERENCE : Set to 1 if the signals that monitor the “out of mould” area position are coherent.

– 40 – MotExe1 (X1)

Transmission of operating mode management data to the executive program.

80	40	20	10	8	4	2	1
APPR–OK	RO–AUTO	RO–POINT	RO–TOTAL	RO–SIMPL	REQ–PCO	REQ–AFCY	REQ–DPCY

DEM–DPCY : Request start of robot cycle (if in Semi–Auto, this request is repeated if the START key or START input is validated – release and press again).

DEM–AFCY : Request end–of–cycle stop of robot.

DEM–PCO : Request execution of Tool Position Change procedure.

RO–SIMPL : Same for standard Home Return.

RO–TOTAL : Same for total Home Return.

RO–POINT : Same for return to previous point.

RO–AUTO : Same for standard automatic Home Return.

APPR–OK : Information return to indicate that learning requested by the execution has been recorded and the execution can continue.

– 41 – MotExe2 (X2)

Data return from the executive program to the operating modes manager.

80	40	20	10	8	4	2	1
FIN2PAS	–	APPFIL	APPRENT	FIN–PCO	FIN–RO	FIN–PAS	FCYCLE

FCYCLE : Return indicating that the cycle is complete. (Response after request for stop at end of cycle).

FIN–PAS : Return indicating end of execution of a step. Used in Semi–Auto mode.

FIN–RO : Return indicating end of execution of Home Return requested (simple, total, point or automatic return).

FIN–PCO : Same for PCO.

APPRENT : Request to go to teach mode.

APPFIL : Request to go to teach mode in the header of FIL GEN.

FIN2PAS : Return indicating end of execution of an SPP step. Used in Semi–Automatic mode.

– 42 – MotReg1 (R1)

Displacement commands in adjust mode: 1.

80	40	20	10	8	4	2	1
DEP–ZM	DEP–ZP	DEP–YM	DEP–YP	DEP–XM	DEP–XP	DEC–VIT	INC–VIT

INC–VIT : Reflects velocity incrementation key or input.

DEC–VIT : Reflects velocity decrementation key or input.

DEP–XP : Reflects X+axis displacement 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–.

– 43 – MotReg2 (R2)

Displacement commands in adjust mode: 2.

80	40	20	10	8	4	2	1
–	–	–	–	DEP–CM	DEP–CP	DEP–BM	DEP–BP

DEP–BP	: Reflects B+axis displacement key or input.
DEP–BM	: Same for B–.
DEP–CP	: Same for C+.
DEP–CM	: Same for C–.

– 44 – MotDio1 (DI)

Management word of terminal diodes.

80	40	20	10	8	4	2	1
– – – –	EXT–TOUT	CLIG–AUT	ALL–REGL	ALL–PCO	ALL–RO	ALL–SEMI	ALL–AUTO

ALL–AUTO	: Set to 1 to request illumination of ”Auto” diode
ALL–SEMI	: Same for ”Semi” diode
ALL–RO	: Same for ”RO” diode
ALL–PCO	: Same for ”PCO” diode
ALL–REGL	: Same for ”Adjust” diode
CLIG–AUT	: Set to 1 to request flashing of ”Auto” diode
EXT–TOUT	: Set to 1 to request extinction of **all** diodes.

– 45 – MotKlax (KL)

80	40	20	10	8	4	2	1
–	–	–	–	–	–	KLAX–OFF	KLAX–ON

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

– 46 – MotDial (DM)

Status of data required for machine dialogue.

80	40	20	10	8	4	2	1
VALID	AUX2	AUX1	PP	PIEC–FAB	PORT–FER	MACH–OUV	AUT–SAUT

AUT–SAUT : 0 = machine in Semi–Auto / 1 = in Auto

MACH–OUV : 0 = machine closed/ 1 = machine open

PORT–FER : 0 = door open/ 1 = door closed

PIEC–FAB : 1 = part made

PP : Reflects presence of a part in the gripper (or status of bit 32).

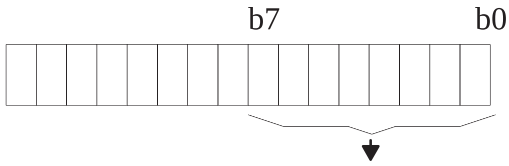
AUX1 : Reserved

AUX2 : Reserved

VALID : Changing program validation by external input.

– 47 – PrgExec (MP)

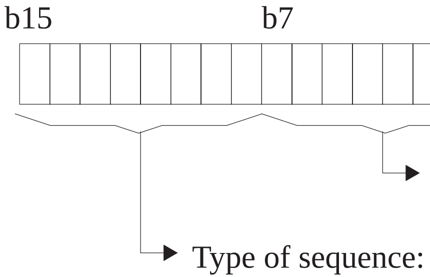
Indicates the program number in progress.



Number from 00 to 63 in Hexadecimal
99 in Decimal

– 48 – SPExec (SP)

Gives SP or SR code in progress.((See Chapter III – 2.)

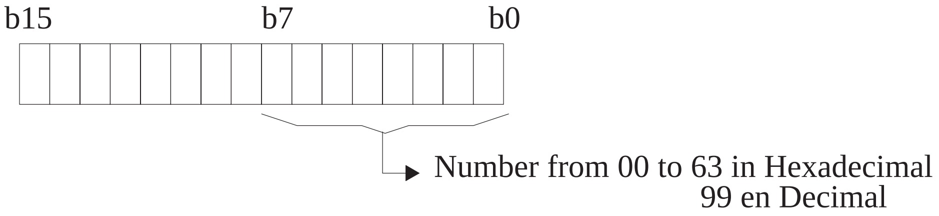


Number 00 to 63 in Hexadecimal
99 in Decimal

Type of sequence:
FA = Standard stacking subroutine
FB = Home Return subroutine

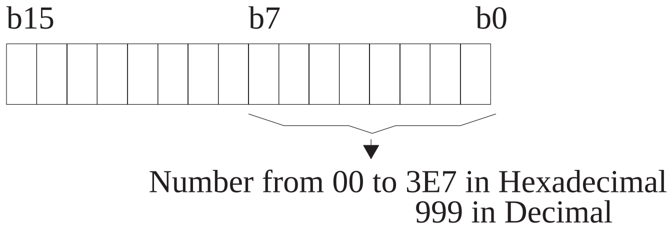
– 49 – PLcExec (PL)

Indicates the PLC program number currently being processed.



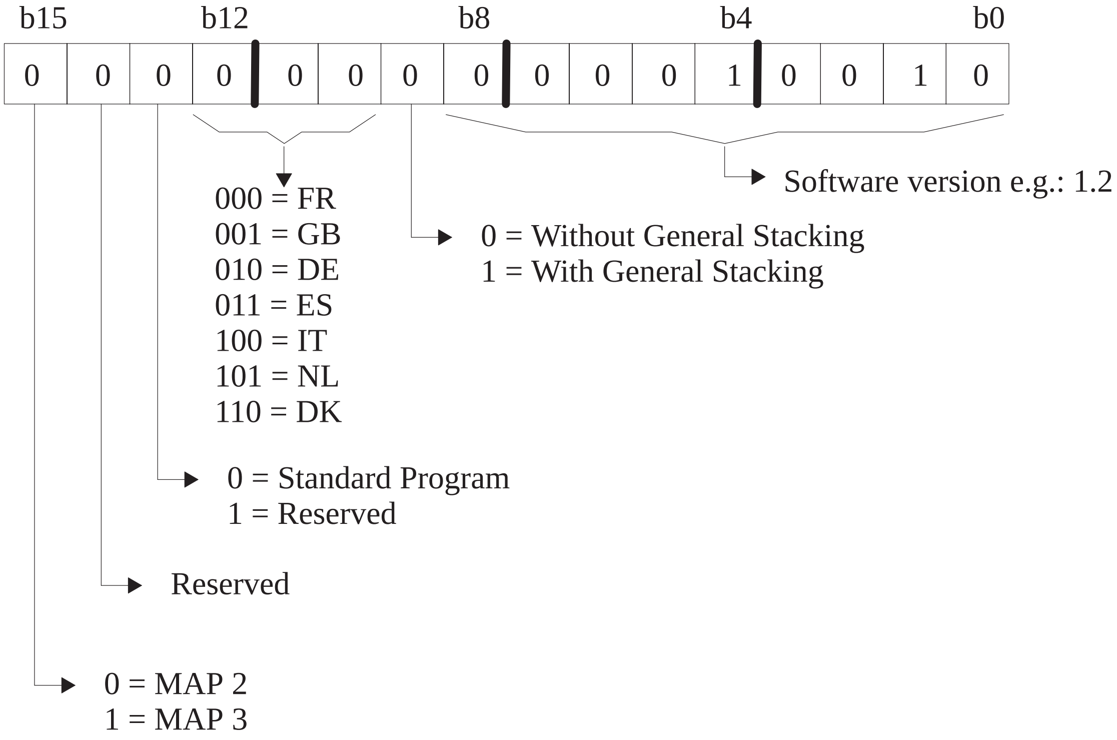
– 51 – PasExec (PA)

Indicates the current step number of the sequence MP, SP or SR.



– 52 – Version (VE)

Number (BCD) of the software version installed with its characteristics.



– 53 – Config (CG)

Options and declared axes.

b0 = 0 –> External EEPROM module: Present
1 –> External EEPROM module: Absent

b1 = 0 –> Module size = 16k x 8 bytes
1 –> Module size = 64k x 8 bytes

b2 = 1 –> File transfer authorised (DCN)

b3 = 1 –> JBus functions authorised

b4 = 1 –> Output on // printer authorised

b5 = 1 –> Trajectory follow–up installed

b6 = 0 –> JBus Protocol
1 –> Euromap 17 Protocol

b7 = 0 –> CPU memory parameters set in 16k x 8 bytes
1 –> CPU memory parameters set in 64k x 8 bytes

b8 = 1 –> X–axis declared

b9 = 1 –> Y– axis declared

b10 = 1 –> Z–axis declared

b11 = 1 –> B–axis declared

b12 = 1 –> C–axis declared

b13 to b15 : Not used

– 54 – Cfg_I_O (IO)

Configuration of installed Inputs/Outputs.

b0 to b4 = Number of 16 installed outputs modulo.

b5 to b7 = Number of 16 installed inputs modulo.

b8 = 1 –> Axis mother board 1 present

b9 = 1 –> Axis mother board 2 present.

– 55 – MotSeri (SR)

Serial number (robot production)

– 56 – MotType (TY)

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:

–> 100, 150, 200, ... models

IV – 5. 3. System words accessible in read/ write

No	Physical address	Name	Description
58	C206	MotCod1	Program number encoded by the inputs or the PLC
59	C208	MotDem1	DPCY, ARFCY, RO, PCO, requests
60	C20A	MotDem2	Complement for ROs and PCOs if commanded by external inputs
61	C20C		Not used
62	C20E		Fault requests
63	C210		Not used

– Detail of system words accessible in read / write:

IF, SET, RST ... WRD instructions can be use.

– 58 – MotCod1 (C1)

Encoding of requested program number.

80	40	20	10	8	4	2	1
ssP	ss64	ss32	ss16	ss8	ss4	ss2	ss1

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

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

Notes:

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.

– 59 – MotDem1 (D1)

Requests to the operating modes manager.

80	40	20	10	8	4	2	1
DEM–POIN	DEM–MONI	DEM–PROG	DEM–RO	PLU–DPCY	DEM–PCO	DEM–AFCY	DEM–DPCY

DEM–DPCY : Start request.

DEM–AFCY : Same for end of cycle stop request.

DEM–PCO : Same for request to go to PCO mode.

PLU–DPCY : Displays release of cycle start key.

DEM–RO : Request for execution of an RO procedure.

DEM–PROG: Request execution of a Programming procedure.

DEM–MONI : Request execution of a Monitor (Display) procedure.

DEM–POIN : Request execution of Position display.

1000	800	400	200	100
REQ–TEST	REQ–HIST	REQ–PROD	REQ–MAINT	REQ2DPCY

DEM2DPCY : Cycle start request (for SP // task).

DEM–MAINT: Same for maintenance management request.

DEM–PROD : Same for request of production management task.

DEM–HIST : Same for request of fault history task.

DEM–TEST ; Same for request of test task.

– 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.

80	40	20	10	8	4	2	1
–	–	RO–POINT	RO–TOTAL	RO–SIMPL	–	–	–

RO–SIMPL : Same for simple Home Return.

RO–TOTAL : Same for total Home Return.

RO–POINT : Same for previous point return.

Note: Word 60, ”MotDem2” should be positioned before the RO request by Word 59 ”MotDem1”.

– 61 –

Reserved.

– 62 – Error

Binary encoding of the error number to be produced.

200	100	80	40	20	10	8	4	2	1

Note : Also produces the message associated with the error.

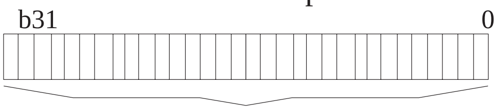
– 63 –

Reserved.

IV – 6. 32 BIT USER AND SYSTEM WORDS – WWRD –

IV – 6. 1. Words: 32 bit user

Accessible in read and write.

Number (logical address)	Physical address	Structures / Functions
WWRD 000 ↓ WWRD 063	F8EA ↓ F9E6	64 Words 32 user bits with no predefined functions.  structure: 32 available bits

IV – 6. 2. Words: 32 bit specific

Not accessible, reserved for future applications.

Number (logical address)	Physical address	Structures / Functions
WWRD 064 ↓ WWRD 127	F9EA ↓ FAE6	Reserved for future software versions.

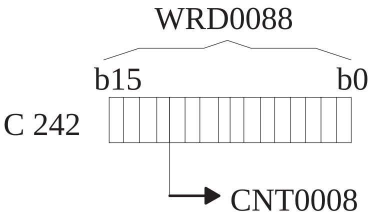
<u>Special case</u>: The following 32 bit words can be used with the 2.0 version:

Number (logical address)	Physical address	Structures / Functions
WWRD 068	F9FA	Number of parts to be made.
WWRD 076	F93A	Number of good parts handled by the robot.
WWRD 077	F93E	Current position of X–axis.
WWRD 078	F942	Current position of Y–axis.
WWRD 079	F946	Current position of Z–axis.
WWRD 081	F94E	Current position of B–axis.
WWRD 082	F952	Current position of C–axis.

IV – 7. COUNTERS

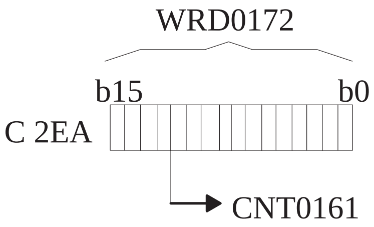
IV – 7. 1. Standard counters – CNT 00 to 15 –

Accessible in read and write.

Number (logical address)	Indirection or logical address (on a Word)	Physical address	Structures / Functions
CNT0000	WRD 0080	C232	 . from 0000 to 9999 in decimal . from 0000 to FFFF in hexadecimal
CNT0001	WRD 0081	C234	
CNT0002	WRD 0082	C236	
CNT0003	WRD 0083	C238	
CNT0004	WRD 0084	C23A	
CNT0005	WRD 0085	C23C	
CNT0006	WRD 0086	C23E	
CNT0007	WRD 0087	C240	
CNT0008	WRD 0088	C242	
CNT0009	WRD 0089	C244	
CNT0010	WRD 0090	C246	
CNT0011	WRD 0091	C248	
CNT0012	WRD 0092	C24A	
CNT0013	WRD 0093	C24C	
CNT0014	WRD 0094	C24E	
CNT0015	WRD 0095	C250	

IV – 7. 2. Stacking counters – CNT 0041 to 9980 –

Accessible in read and write.

Number (logical address)	Indirection or logical address (on a Word)	Physical address	Structures / Functions
CNT0041	WRD 0096	C252	
↓	↓		
CNT9980	WRD 4095	E192	
↓			

See table for valid numbers.	The corresponding 16–bit words and the addresses are continuous. There is no break in the numbering:
	:
	CPT0080 = WRD0135 –> C2A0
	CPT0141 = WRD0136 –> C2A2
	:
	CPT0180 = WRD0175 –> C2F0
	CPT0241 = WRD0176 –> C2F2
	:
			. from 0000 to 9999 (decimal) in general storage subroutine No.61 of main program No. 01.

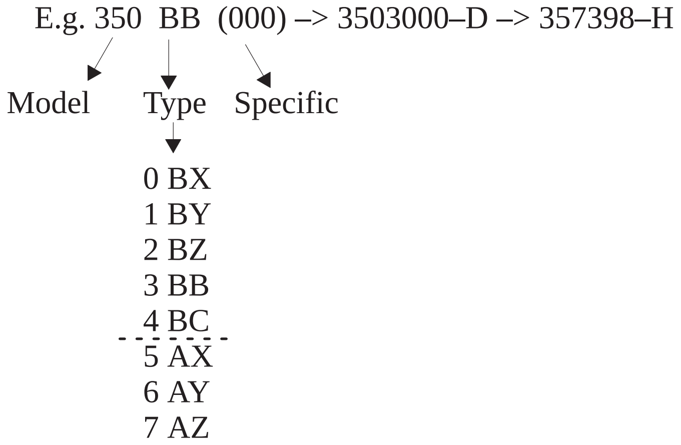
IV – 8. SPECIFIC INFORMATION

These are directly accessed using the Memory Read function followed by the request [Address] and a letter :

– MOT. Y W to access the memory area containing the passwords.

– FUN S to access the memory area containing the serial number and the type of robot.

	15	0	
809A0	00	00	Password to enter edition if SAP module is present (1.9.9.2)
809A2	07	C8	
809A4	00	00	Password to enter the [SYST] function of the Memory management procedure (3.1.4)
809A6	01	3A	
809A8	00	00	Password for modifying instructions in the [MRead] procedure (4.2.1)
809AA	01	A5	
809AC	00	00	Password to enter into the machine parameters or the Machine Origin Values (POM) (1.2.3.4)
809AE	04	D2	
809B0	00	00	Password to modify the procedure counters [Prod] (1.2.3)
809B4	00	7B	
809E8	00	00	Serial number of the robot: E.g. 1024
809EA	04	00	
809EC	00	35	Type of robot:
809EE	73	98	
809FF			


E.g. 350 BB (000) –> 3503000–D –> 357398–H

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.