

Programmable Controllers

LEVEL 1

Operating Modes and using the programming unit PCA2.P10

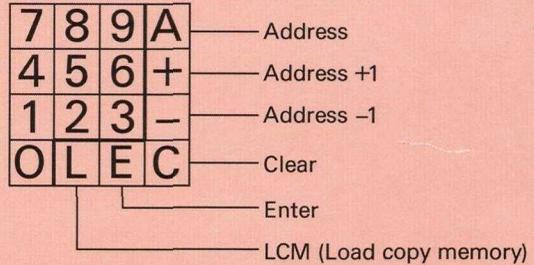
Operating mode selector switch

RUN
BREAK
STEP
MAN
LCM
PROG

Inputs and outputs are active

only PCA2

Keyboard of the programming unit PCA2.P10



RUN

Normal program execution

Irrespective of the mode selector switch position, the PCA2 is always on RUN with the programming unit being disconnected (contrary to the PCA1).

BREAK

Interruption of the program execution with subsequent step-by-step operation

- [+] The last processed instruction line is displayed
- [+] [+] ... Processing of the following instruction lines in step-by-step operation

The whole instruction line as well as the state of the ACCU is displayed.

Program execution in the BREAK-mode is effected in all parts like in the RUN-mode i. e. in the case of parallel programs all programs are executed.

Setting a «Breakpoint»

- [A] 820 [+] Up to step address 820 the program is executed automatically (10 times slower than in RUN-mode) and then stops
- [+] [+] ... Processing of the following instruction lines in step-by-step operation

If the operating mode selector is switched to RUN, execution of the program continues from where it was stopped in step-by-step operation.

STEP

Jump to the preselected step address

- [+] The last processed instruction line is displayed
- [A] 139 [+] Jump to step address 139
- [+] [+] ... Processing of the following instruction lines in step-by-step operation

The whole instruction line as well as the state of the ACCU is displayed.

A jump to the start step address of a selected parallel program allows execution of just this parallel program.

If the operating mode selector is switched to RUN, execution of the program continues from where it was stopped in step-by-step operation.

MAN

Manual interrogation and setting / resetting of the logic state of elements

(elements = inputs, outputs, flags, counters and timers)

Interrogation:

- [A] 32 Interrogation of the element 32. The last digit of the operand is 1 or 0 (High or Low)

Subsequently setting / resetting:

- [E] Releases input for the logic state
- [1]/[0] Sets the element to High or Low

LCM

Load Copy Memory (This operation is only possible with the series PCA2.)

- [A] STEP XXXX Introduction of the start step address of the program to be reloaded
- [E] CODE OPERAND 00 XXXX Introduction of the final step address of the program
- [L] Start of the reloading procedure

Reloading i. e. copying of a program from RAM to EPROM, EPROM to EPROM, EPROM to RAM, RAM to RAM (copy memory on the lower memory socket).

Reloading of the program on EPROM is effected at a speed of 1K/100s and can be watched on the display.

PROG

Programming

- [A] STEP XXXX The key [A] sets the step register to 0. Then introduction of the start step address.
- [E] CODE OPERAND XX XXXX The key [E] increments the step address by 1 and sets the code and operand-register to 0. Any memory contents will be cleared. Introduction of the instruction in numerical code as well as the operand.

- [C] Clears a wrong input
- [+] [+] resp. Examination of the program
- [-] [-]

A program can only be entered or corrected on a RAM-user memory.

Survey of the instructions level 1

Step address
STEP

Instruction in
numerical code
CODE

Element or jump address
OPERAND

Accumulator
ACCU = 1

0 1 3 9

0 1

0 3 5 6



Display on
programming
unit
PCA2.P10

Address range:
EA = Element address
S = Step address
PP = Parallel
program
number
OP = Operand
number
I = Index value
(i) = indexable

**Influence
regarding
ACCU**

Only performed
when ACCU =
Instruction
sets ACCU =

**Logic
Instruc-
tions**



**Switching
Instruc-
tions**



**Time
and
Counting
Instruc-
tions**

**Jump
Instruc-
tions**



**Wait
Instruc-
tions**

**Auxiliary
Instruc-
tions**



Numerical code	Mnemonic code 2nd line	Instruction	Description	EA =	Only performed when ACCU =	Instruction sets ACCU =
01 02	STH STL	Start High Start Low	{ Start of an operation: } High { Element interrogated for } Low	EA = 0...999 (i)		1/0
03 04	ANH ANL	And High And Low	{ And-operation of } High { ACCU with element interrogated for } Low	EA = 0...999 (i)		1/0
05 06	ORH ORL	Or High Or Low	{ Or-operation of } High { ACCU with element interrogated for } Low	EA = 0...999 (i)		1/0
07	XOR	Exclusive Or	Exclusive-or-operation of ACCU with addressed element	EA = 0...999 (i)		1/0
08	NEG	Negate ACCU	Invert state of the ACCU	0		0/1
09	DYN	Dynamic Control	Signal edge triggering or dynamic control of an operation	EA = 288...999 (i)		1/0
10	OUT	Set Output with Status of ACCU	Transfer the state of the ACCU to an output or a flag	EA = 0...255, 288...999 (i)		1/0
11 12	SEO REO	Set Output Reset Output	Set output or a flag and store Reset output or flag and store	EA = 0...255, 288...999 (i)		1
13	COO	Complement Output	Interrogate state of output or flag and set it to the opposite state	EA = 0...255, 288...999 (i)		1
14	STR (001)	Set Timer -	Set timer to preselected value and start it Time value in 1/10 s (resp. 1/100 s)	EA = 256...287 (i) 0...2047		1
15	SCR (001)	Set Counter -	Set counter to preselected value Count value 1) for values > 2047 see Basic-Manual C 7.3.	EA = 256...287 (i) 0...2047		1
16 17 18	- - -	- - -	Externe value input in BCD format: Read-in instruction for 2 x 4 bit BCD x1 Read-in instruction for 2 x 4 bit BCD x10 Read-in instruction for 2 x 4 bit BCD x100	EA = 7...255, 295...999		
17 18	INC DEC	Increment Counter Decrement Counter	Increments } contents of the counter by 1 Decrements }	EA = 256...287 (i)		1
20	JMP	Unconditional Jump	Unconditional jump to step address	S = 0...2047		1
21 22	JIO JIZ	Jump if ACCU is One Jump if ACCU is Zero	{ Jump to step address if } ACCU = 1 { } ACCU = 0	S = 0...2047		1 0 1 1
23 24	JMS RET	Jump to Subroutine Return from Subroutine	Jump to the subroutine Return from the subroutine	S = 0...2047 0		1 1
25 26	WIH WIL	Wait if High Wait if Low	{ Wait as long as element is } High { } Low	EA = 0...999 (i)		1 1
00	NOP	No Operation	No operation	0		
19	SEA	Set Accumulator = 1	Sets ACCU = 1	0		1
16	SEI	Set Index	Sets index register to initial value (I0)	I0 = 0...255		1
27 28	INI DEI	Increment Index Decrement Index	Increments } index register by 1 Decrements } up to the final value (I)	I > I0 I < I0		1/0 1/0
29	PAS (00)	Program Assignment -	Assignment of the parallel program Start step address of the parallel program	PP = 0...15 S = 0...2047		
30 31	DOP DTC	Display Operand Display Timer/Counter	Display of the number in the operand Display of timer or counter value	OP = 0...2047 EA = 256...287 (i)		0 1

Input of text into the text memory

1 Programming of the serial data interface as editor (with the programming unit P10 in PROG-mode)

4001	PAS	(29)	100	
4002		00	898	Parameter (depending on the peripheral unit)
4003		01	63	01 = activate editor for text input
4004		00	254	} An element which continuously remains on LOW
4005		00	254	
4006		00	0	
4007		00	0	
4008		00	0	
4009		00	0	
4010		00	0	
4011	PAS	(29)	23	} TXn = text-number for text output in the STEP-mode
4012		00	TXn	
4013	JMP	(20)	0	
4014		00	4011	

2 Activation of the editor

In operating mode STEP jump to the step address 4001 and execute up to 4011 by means of the programming unit P10 → editor will be ready

4 Input mode of the editor (with peripheral unit)

CTRL T Text input as from character-no. which is preselected via the cursor in the display mode
Input of the floating text, the control and the function characters

CTRL @* End of a text

^ CR Will be taken over into the text memory as <CR> (Carriage Return)

CR Change to the display mode

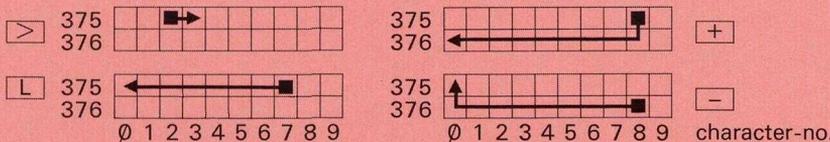
* ASCII NUL is depending on the peripheral unit

3 Display mode of the editor (display of the text memory contents on the peripheral unit)

(display of the text memory contents on the peripheral unit)

TXn **CR** Text-line with text-number TXn is displayed

Cursor control



Function characters in the text memory

Alphanumerical input	Input via P10	Description
\$C Cn*	36, 67, Cn	Output of the contents of timer/counter register Cn 5 decimal digits, leading zeros <u>are not suppressed</u>
\$c Cn*	36, 99, Cn	5 decimal digits, leading zeros <u>are suppressed</u>
\$E En*	36, 69, En	Output of the state of the 8 elements En-7...En (8 characters 1 or 0)
\$e En*	36, 101, En	Output of the ASCII character which is composed of the 8 elements En-7...En
\$T	36, 84	Output of the date-time contents: :14 : 04 / 18 : 50 : 42 week day / hour minute second (14.) (Thur.)
\$H	36, 72	84 - 04 - 05 / 18 : 50 : 42 year month day / hour minute second
\$D	36, 68	84 - 04 - 05 year month day
\$ _ xxx*	36, _ , xxx	Repeated output of a character immediately following the character \$, as often as specified in xxx: xxx < 127 Repetition according to xxx xxx = 256...319 Repetition corresponding to the contents of the addressed timer/counter (max. 127) All ASCII characters (incl. NUL) may be used except for the following characters which initiate functions: \$, A, C, D, E, H, L, R, T, U, a, c, e, r
\$L xxx*	36, 76, xxx	Jump to text no xxx and subsequent storage of the jump address (character-number)
\$U	36, 85	Jump back to the starting address +1
\$	36, 36	\$ (Output of the Dollar symbol)

* Three digits must always be entered for Cn, En, xxx, e.g. 027.

5 Text output for checking

Depress the **[+]**-key on the programming unit P10 twice in STEP-mode: every time the text will be displayed for checking purposes

ASCII CODE-Chart

B7	B6	0	0	0	0	1	1	1	1		
BITS	B5	0	1	0	1	0	1	0	1		
B4B3B2B1		CONTROL				SYMBOLS		UPPERCASE		LOWERCASE	
0 0 0 0		NUL	DLE	SP	@	P					
0 0 0 1		SOH	DC1	!	A	Q	a	q			
0 0 1 0		STX	DC2	"	B	R	b	r			
0 0 1 1		ETX	DC3	#	C	S	c	s			
0 1 0 0		EOT	DC4	\$	D	T	d	t			
0 1 0 1		ENQ	NAK	%	E	U	e	u			
0 1 1 0		ACK	SYN	&	F	V	f	v			
0 1 1 1		BEL	ETB	'	G	W	g	w			
1 0 0 0		BS	CAN	(H	X	h	x			
1 0 0 1		HT	EM)	I	Y	i	y			
1 0 1 0		LF	SUB	*	J	Z	j	z			
1 0 1 1		VT	ESC	+	K	[k	{			
1 1 0 0		FF	FS	,	L	\	l				
1 1 0 1		CR	GS	-	M]	m	}			
1 1 1 0		SO	RS	.	N	^	n	~			
1 1 1 1		SI	US	/	O	_	o	~			

* With most of the peripheral units the <control case> characters are obtained by simultaneously depressing the **CTRL** -key and the corresponding character of the <upper case>. For example <FF> (Form Feed, no. 12) is obtained by simultaneously depressing the keys **CTRL** and **[L]** (no. 76).

Numerical code	Mnemonic code 2nd line	Description	Operand (i) = indexable	Numerical code	Mnemonic code 2nd line	Operand	Description
14	STR 00...31 ¹⁾	TIME AND COUNTING INSTRUCTIONS Set addressed timer to preselected time value and start it. Time value in 1/10 s (resp. 1/100 s)	256...287 (i) 256...319 ³⁾ (i) + Value in the operand 0...2047	29	PAS	16	PAS INSTRUCTIONS Exclusive operation of a parallel program (with blocked time base, software date-time and serial data line) 2nd line is always 0 End of the exclusive operation 2nd line is always 0
15	SCR 00...31 ¹⁾	Set addressed counter to preselected count value. Count value	7...999 (i) Highest address of the sequence of elements (LSB = Least Significant Bit)	29	PAS	17 0	Defines the number of the active, assigned parallel programs 2nd line determines the active parallel programs
	00 ... 15	¹⁾ Values 0 ... 32767 00 (≙ 0) 01 (≙ 2048) 02 (≙ 4096) 03 (≙ 6144) 04 (≙ 8192) 05 (≙ 10240) 06 (≙ 12288) 07 (≙ 14336) 08 (≙ 16384) 09 (≙ 18432) 10 (≙ 20480) 11 (≙ 22528) 12 (≙ 24576) 13 (≙ 26624) 14 (≙ 28672) 15 (≙ 30720)		29	PAS	18	
	16 17 18 19 24 25 26	¹⁾ Read-in Instructions for 2 x 4 bit BCD x1 for 2 x 4 bit BCD x10 for 2 x 4 bit BCD x100 for 5 x 4 bit BCD x1 for 8 bit binary for 12 bit binary for 16 bit binary		29	PAS	23 0...818	Instruction for text output In the 2nd line the starting text number is determined. Output is effected up to the character (NUL) in the text
	27 28 29 30	¹⁾ Arithmetic Operations Addition Subtraction Multiplication Division	1...255 256...319 ³⁾ (i)	29	PAS 00 PAS	30 0 31...38 xxxx	Checking the system program via (check sum) 2nd line is always 0 Checking the user program via (check sum) (31...38 ≙ 1.K...8.K) 2nd line contains the check sum
	31	¹⁾ Data exchange between counter register and index register INDEX REGISTER 8 BIT PP15 PP7 PP0 2nd line 31/0 COUNTER REGISTER 16 BIT C319 C295 Cn C280 1st line C256 1st line The value of the index register is loaded into the counter The value of the addressed timer/counter (2nd line) is loaded into the timer/counter (1st line) The value of the addressed counter (2nd line) is loaded into the counter (1st line)	0 256...287 (i) 288...319 ³⁾ (i)	29	PAS xy	50 Cn	Data transfer between date-time and counter register xy determines the type of function: x: 0 ≙ write into the register of the date-time 1 ≙ read out the register of the date-time y: 0 ≙ week of the year (value: 1...53) 1 ≙ day of the week (value: 1...7) 2 ≙ year (value: 0...99) 3 ≙ month (value: 1...12) 4 ≙ day of the month (value: 1...31) 5 ≙ hour (value: 0...23) 6 ≙ minute (value: 0...59) 7 ≙ second (value: 0...59) Cn determines the timer/counter from which the value to write into the register of the date-time is taken resp. where the value read out of the register of the date-time is stored.
20/21 22/23	JMP/JIO JIZ/JMS 00 ²⁾	JUMP INSTRUCTIONS Step addresses ≤ 2047 may be introduced using one or two lines Step addresses > 2047 have to be introduced using two lines	0 0 0...8191 ²⁾	29	PAS	100	10-line instruction to determine the transmission parameters of the serial data line and the mode T, E, C, N or P
16	SEI	AUXILIARY INSTRUCTIONS The value of the operand (constant) is loaded into the index register The value of the addressed timer/counter is loaded into the index register The final value for the index register is in the operand (constant) The final value for the index register is in the addressed timer/counter	0...255 256...319 ³⁾	29	PAS 00	190 19 0	10-line instruction for the interrupt-management (fast reaction due to an exclusive execution according to Interrupt-Service-Routine) End of the interrupt-management 2nd line is always 0
27 28	INI DEI	0...255 of the operand (constant)	0...255 256...319 ³⁾	29	PAS	200 201 202 250 251	10-line instruction for the parameter of a maximum of 32 PID-control loops 10-line instruction to define and activate a rotation or a shift register 10-line instruction to define and activate a FIFO-stack.

³⁾ can be extended to 383 when using PAS 200...202