OWNER'S MANUAL

MESURE
MCC
CONTROLE
COMMANDE

PROGRAMMABLE INDICATOR

ICP 300

Ref : NE136A-02/95

MESURE CONTROLE COMMANDE
74, allée Helsinki
Z.E. Jean Monnet Nord
83500 La Seyne sur Mer - France
Tél : +33 (0)4 94 22 00 24
Fax : +33 (0)4 94 22 10 82
Email : info@mcc-instrumentation.com
Web : www.mcc-instrumentation.com

1 GENERAL FEATURES

1.1 Functions

Digital indicator, it allows 3 inputs, displays up to 7 variables, controls 7 alarms setpoints with additional output relays (4), 2 analogue retransmission signals, 2 transducer power supplies and 1 digital RS output.

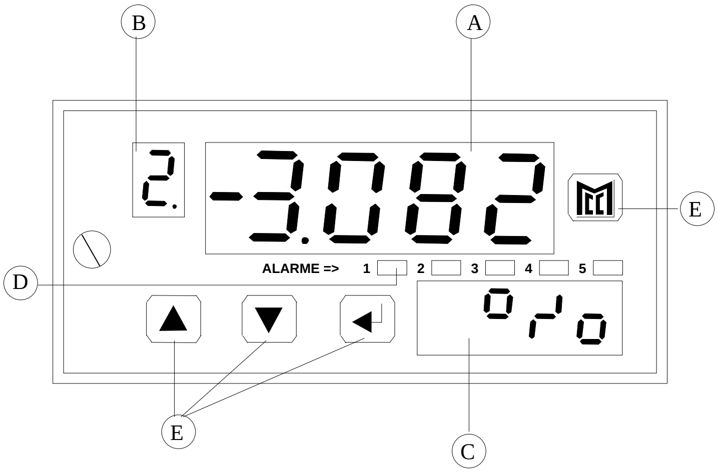
**** Standard :*** ***3 process variable inputs***
2 logic inputs
1 relay output

Programmable calculator, it makes various calculations on the 3 analogue inputs, displays and outputs the results. You can use the basic operation (+,-,x,/, etc...) and mix them to suit your needs.

Programmer on its 5 logic inputs, it performs simple automating functions created by the user according to its applications.

Entirely programmable on the front panel, its modular construction (by option boards) makes it a ***universal apparatus, adaptable to many diverse applications.***

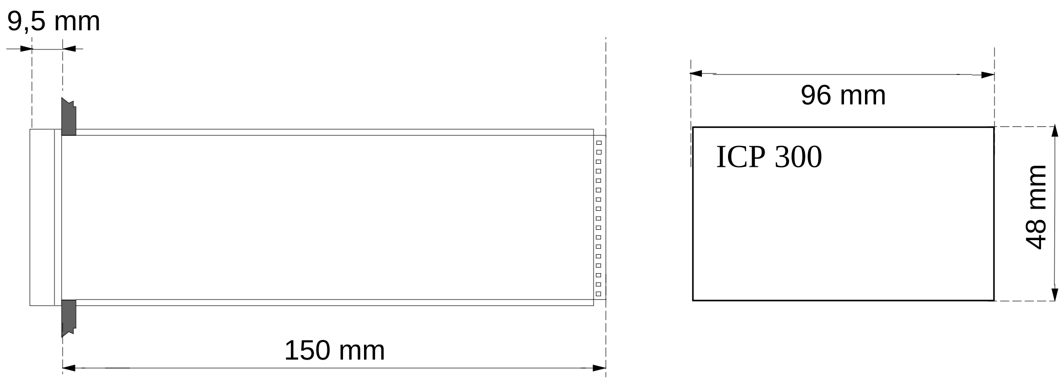
1.2 Front panel description

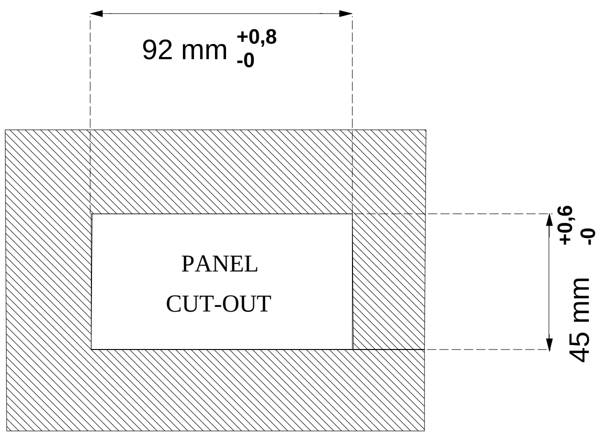


	DISPLAY	
A	**Process variable**	4 digits red LED , 7 segments of 13,6mm
B	**Number of the displayed channels**	1 digit green LED , 7 segments of 7,6mm
C	**Variable units and mnemonics**	4 digits green LED, 7 segments of 7,6mm
D	**Alarm status signal lights**	5 red signal lights
E	**Functions keys**	4 keys for the apparatus configuration or use.

1.3 Mechanical specifications

Dimensions : 96 x 48 x 150 mm behind the collar
Cut-out : 92 x 45 mm
Weight : around 0,6 Kg
Blue front panel, non flammable plastic NORYL grey case.
A secured "plug-in" arrangement for easy removal (screw on front panel).
Watertighness IP 65 in front panel.
Connection terminals : 2 x 1,5 mm2





1.4 Technical specifications

1.4.1 GENERALS

POWER SUPPLY 230 V -115 V /50-60 Hz, +10 % /-15 %, on 4 wires
consumption under 10 VA.
CMR : 120 dB to 250 V / 50 Hz
SMR : according to the response time
12, 24, 48 VDC as options.

ISOLATION Inputs/Outputs 500 V
I/O/Power supply 1500 V
Digital RS/µP board 500 V
Common point between logic and analogue inputs

ACCURACY 0,1 % on linear inputs
0,1 % on TC and RTD inputs according to the standard span
Internal cold junction compensation (0,5 ˚C at 25 ˚C, +0,5˚C/10˚C)

SAMPLING around 250 ms

TEMPERATURE LIMITS Working : 0-50 ˚C, variation : $150 ppm/°C$
Storage : -20 to 70 ˚C

1.4.2 ANALOGUE INPUTS

ICP 300 admits 3 analogue inputs entirely configurable.

N.B. : They can be use as logic inputs as well.

	STANDARD SPANS
Thermocouple K	-50-1373 ˚C / -50-500 ˚C
Thermocouple J	-50-1200 ˚C / -50-370 ˚C
Thermocouple T	-50-400 ˚C
Thermocouple S	-50-1769 ˚C
Thermocouple R	-50-1769 ˚C
Thermocouple N	-50-1300 ˚C / -50-600 ˚C
Thermocouple B	0-1820 ˚C
RTD	-200 - 650 ˚C / -50 - 200 ˚C
Voltage (square root extraction)	0-5 V / 1-5 V / 0-1 V / 0,2-1 V / 0-125 mV / 0-65 mV / 0-20 mV / ± 25 mV
Current (square root extraction)	4-20 mA, 0-20 mA on 250 or 50 Ohms at 0,1 % resistive load
Potentiometer	100 Ohms to 10 Kohms

LINEARISATION	2 tables of 6 segments.
SENSOR WIRE	20 Ohms maximum
RESISTIVE LOAD	variation : $0{,}5.10^{-4}/Ohm$
SENSOR FAILURE	Indication for the 3 channels : Flashing display Logic signal usable within a program.

1.4.3 LOGIC INPUTS

2 logic inputs are available (terminals 4 and 5)

Dry contact	Level 1 = closed contact Level 0 = opened contact
0 - 10 V max	Level 1 = lower than 2 V Level 0 = upper to 4,5 V

N.B.: Non used analogue inputs can be used as logic inputs.

Dry contact	Level 1 = closed contact Level 0 = opened contact
0 - 10 V max	Level 1 = lower than 2 V Level 0 = upper to 3,5 V

1 logic input can be assigned to the ↵ key in front panel by program .

1.4.4 IMPULSE INPUTS

The two logic inputs can be used as impulse inputs limited to 300 Hz for a period equal to 1.

The time between two status changes must be upper to 1,5 ms.

Time resolution : 25 ms.

If between 2 impulses, the time is lower than 0,237 ms, the accuracy is equal to ± 25 ms.

If between 2 impulses, the time is upper than 0,237 ms, the accuracy is given by :

$$Accuracy = \frac{T_m^2}{31068}$$

Intrinsic error is equal to : Tm - accuracy ± 25 ms.
with Tm = Time between 2 impulses.

1.4.5 OUTPUTS

1.4.5.1 Relay output

If those relays are used to command inductive loads, it could be better to add capacitor. For a good electromagnetic compensation, you may connect the spare part (referenced H10316) on the load terminal. You can also add it to the relay terminal but it would not be so good.

1.4.5.1.1 1 relay on basic apparatus

Relay type : normally open (relays are opened when they are de-energized.)

Cut off : 3 A , 250 Vac or 30 Vdc.

1.4.5.1.2 1 relay output board

Relay type : change over

Cut off : 3 A , 250 Vac or 30 Vdc.

1.4.5.1.3 2 relays output board

Relays type : normally open (de-energized) with common connection.

Cut off : 3 A , 250 Vac or 30 Vdc.

1.4.5.1.4 2 relays S output board

It is the same board as the previous one but a safety device avoids the 2 relays to be simultaneously closed even if they are both energized.

1.4.5.2 Logic output board

The output signal : 0 V to 10 V ± 10% with a current limit at 10 mA.
It can also be used as an open collector output with a current limit 10 mA.

The logic output is isolated from the rest of the apparatus at 500 Vac.

1.4.5.3 Current output board

Type : 4-20mA or 0-20mA.
Resolution : 12 bits.
Resistive load : 500 ohms max.
Isolated from the rest of the apparatus at 500 Vac.
Output accuracy : ± 0,1% (the board is calibrated in our factory fitted in the apparatus)

1.4.5.4 Voltage output board

Type : 0-5 V, 1-5 V, 0-10 V or 2-10 V.
Resolution : 12 bits.
Protected against short circuirt : 40 mA max.
Isolated from the rest of the apparatus at 500 Vac.
Output accuracy : ± 0,1% (the board is calibrated in our factory fitted in the apparatus)

1.4.5.5 Auxiliary power board

This board provides a power supply of about 22 Vdc with a current limitation at 28 mA.

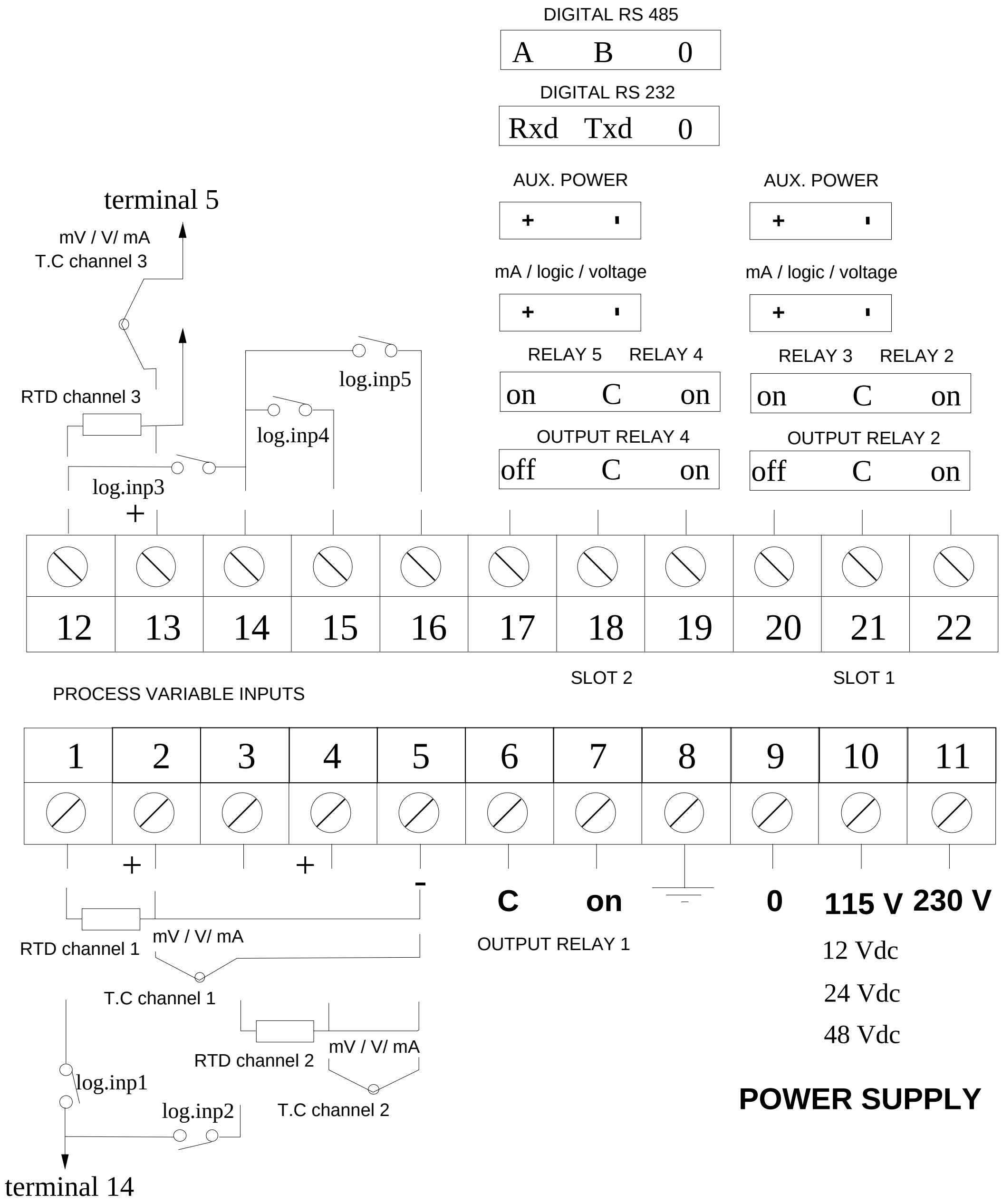
1.4.5.6 RS485 board

This digital RS allows long distance communications (<1 Km) and is multidrop (< 32 apparatus).
Protocol MODBUS ASCII or RTU, 300 to 9600 Baud
For further information a digital manual is provided with the board.
Isolated from the rest of the apparatus at 500 Vac.

1.4.5.7 RS232 board

The digital RS232 is limited in distance (<30 m) and is monodrop.
Protocol MODBUS, ASCII or RTU, 300 to 9600 Baud
This board can also be used as a printer output.
For further information a digital manual is provided with the board.
Isolated from the rest of the apparatus at 500 Vac.

1.5 Connections



2 HARDWARE CONFIGURATION

2.1 Principle

2.1.1 INPUTS

ICP 300 is entirely configurable by software on its inputs. No modification must be done on the hardware (see in § 3)

2.1.2 OUTPUTS

The basic apparatus is provided with 1 relay linked to alarm 1.

2 optional slots are available and all the following boards can be connected. Some of them can be added unimportantly on slot n°1 or on slot n°2.

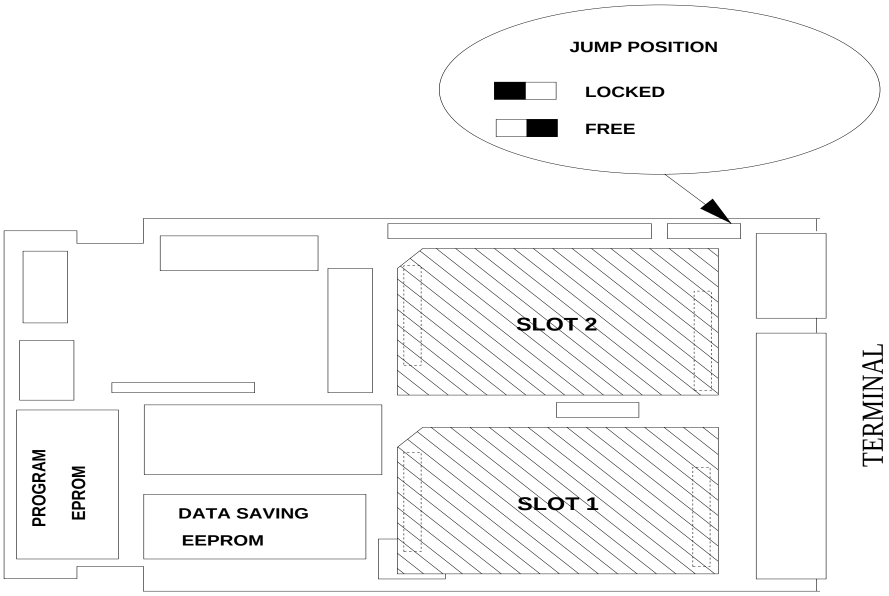
OPTIONAL BOARD	SLOT n° 1	SLOT n° 2	Code
LOGIC	Alarm 2	Alarm 4	1
1 RELAY	Alarm 2	Alarm 4	2
2 RELAYS	Alarm 2 and 3	Alarm 4 and 5	3
CURRENT	Retransmission PV or calculation result	Retransmission PV or calculation result	4
2 RELAYS S	Alarm 2 and 3	Alarm 4 and 5	5
RS232 or 485	FORBIDDEN	Calculator link	6
VOLTAGE	Retransmission PV or calculation result	Retransmission PV or calculation result	7
AUXILIARY POWER	22 V, 22mA	22 V, 22mA	8

At switch ON, these boards are recognized by the microprocessor and a two numbers code is shortly displayed. The left one gives the number of the board connected in slot n° 1, the right one the board's number in slot n° 2.

N.B The relay outputs uses are only given as examples. You can assigned them to other parameters (see in chapter 5 PROGRAM FUNCTION)

2.2 Board connections

1. Un-plug out the apparatus (screw on front panel)
2. Disconnect the front panel pulling the plastics clamps
3. Disconnect the 2 boards. Slots n° 1 and n° 2 are located on the microprocessor board.



4. Pull out the present board
5. Put the front panel down on a table and replace the display board
6. Re-connect microprocessor and power supply boards
7. Re-connect the font panel

N.B :

1. ***For any board change you must configure again the apparatus until you reach the working mode***
2. ***For any current or voltage board change you must calibrate again the apparatus***

2.3 Current output board calibration

When you connect a current output board we advise you to calibrate again to keep a ± 0,1 % accuracy.

To access to calibration of current output boards, the apparatus must be in CONFIGURATION mode.

Required means : Precision milliammeter

1. Press simultaneously ↓ and M.C.C keys at switch on.
2. "ETAL" message appears.
3. Press ↵ key, "CODE" message appears. If you don't want to calibrate again, press MCC key.
4. Otherwise enter 8031
5. According to the analogue board location,"ETAL OUT1" (slot1) or "ETAL OUT2 " (slot2) messages appears. If there are analogue boards on each slot, you will have to calibrate first OUT1 then OUT2.
6. Connect the milliammeter to OUT1 or OUT2 .
7. Press ↵ key.
8. ICP display indicates "3.300" mA .
9. Compare this previous value with the one you read on milliammeter .Use ↑ and ↓ keys to correct it in ICP.
10. Then ICP display indicates "17.90" mA
11. In the same way, use ↑ and ↓ keys to enter the real value read on milliammeter.
12. Finally, ICP display indicates "10.00" mA .
13. Verify that you read well the same value on the milliammeter . Otherwise, you must proceed again.
14. Press ↵ key.
15. At the calibration end,"ETAL STOC" message appears. To save your calibration press ↵ key otherwise press MCC.
"..." is displayed during saving.

2.4 Voltage output board calibration

When you connect a voltage output board we advise you to calibrate again to keep a ± 0,1 % accuracy.

To access to calibration of voltage output boards, the apparatus must be in CONFIGURATION mode.

Required means : Precision voltmeter

1. Press simultaneously ↓ and M.C.C keys at switch on.
2. "ETAL" message appears.
3. Press ↵ key, "CODE" message appears. If you don't want to calibrate again, press MCC key.
4. Otherwise enter 8031
5. According to the analogue board location,"ETAL OUT1" (slot1) or "ETAL OUT2 " (slot2) messages appears. If there are analogue boards on each slot, you will have to calibrate first OUT1 then OUT2.
6. Connect the voltmeter to OUT1 or OUT2 .
7. Press ↵ key.

8. ICP display indicates "1,660" V .

9. Compare this previous value with the one you read on voltmeter .Use ↑ and ↓ keys to correct it in ICP.

10. Then ICP display indicates "8,870 V"

11. In the same way, use ↑ and ↓ keys to enter the real value read on voltmeter.

12. Finally, ICP display indicates "5.00" V .

13. Verify that you read well the same value on the voltmeter. Otherwise, you must proceed again.

14. Press ↵ key.

15. At the calibration end,"ETAL STOC" message appears. To save your calibration press ↵ key otherwise press MCC.
"..." is displayed during saving.

3 SOFTWARE CONFIGURATION

3.1 Access to programmation mode

To go to configuration :

- The internal jump must be on "free" position (see in § 2.2)
- After switch on, press similtaneously ↑ and MCC.
  You will read "CONF"; press ↵.
  If you do not have set a locking level, you are in programmation mode.
- If a locking level has been set (see in § 4.2.5), you will read "CODE"; so enter 8031.

3.2 Key board use

MCC to change the digit.

↑ and ↓ to set the digit.

↵ to valid the configuration of each block and to move to the following one.

3.3 Mnemonics description

The green display gives the mnemonics, the red one the value of the parameter.

PARAMETERS TO SET	MNEMONICS
INPUT BLOCK N°1 - Channel 1 min scale - Channel 1 max scale - Message channel 1	**ENT.1** **MES._** **MES.-** **REP.1**
INPUT BLOCK N°2 - Channel 2 min scale - Channel 2 max scale - Message channel 2	**ENT.2** **MES._** **MES.-** **REP.2**
INPUT BLOCK N°3 - Channel 3 min scale - Channel 3 max scale - Message channel 3	**ENT.3** **MES._** **MES.-** **REP.3**

PARAMETERS TO SET	MNEMONICS
DISPLAY BLOCK I **DISPLAY BLOCK P** **message channel N°4** **message channel N°5** **message channel N°6** **message channel N°7**	**AFF.I** **AFF.P** **MES.4** **MES.5** **MES.6** **MES.7**
SPECIAL FUNCTIONS BLOCK	**SPEC**
ALARM BLOCK N°1 **ALARM BLOCK N°2** **ALARM BLOCK N°3** **ALARM BLOCK N°4** **ALARM BLOCK N°5** **ALARM BLOCK N°6** **ALARM BLOCK N°7**	**ALR.1** **ALR.2** **ALR.3** **ALR.4** **ALR.5** **ALR.6** **ALR.7**

Then according to the connected boards :

PARAMETERS TO SET	MNEMONICS
analog output block No 1 **analog output block** No 2	**OUT.1** **OUT.2**
DIGITAL RS BLOCK	**RS**

3.4 PROCESS VARIABLE input n° 1 [Ent.1]

3.4.1 THERMOCOUPLE INPUT

THERMOCOUPLE			
DIGIT N°1	**DIGIT N°2**	**DIGIT N°3**	**DIGIT N°4**
[1] With C.J.C [2] Without C.J.C	[0] Type K -50 to 1373 °C [1] Type K -50 to 500 °C [2] Type J -50 to 1200 °C [3] Type J -50 to 370°C [4] Type T -50 to 400 °C [5] Type S -50 to 1800 °C [6] Type R -50 to 1769 °C [7] Type N -50 to 1300 °C [8] Type N -50 to 600 °C [9] Type B 0 to 1820 °C	[0] displayed input [1] non displayed input*	[0] Without decimal point [1] 1 decimal place

To valid that code, press : [↵] key.

* This input will stay an internal value, you will never read it on display.

3.4.2 RTD INPUT

RTD			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
[3]	[0] -200 to 650 °C [1] -50 to 200 °C	[0] displayed input [1] non displayed input	[0] Without decimal point [1] 1 decimal place

To valid that code, press : [↵] key.

3.4.3 LINEAR INPUTS

LINEAR INPUTS			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
[4] Without square root extraction [5] With square root extraction	[0] 0 to 5 V [1] 1 to 5 V [2] 0 to 1 V [3] 0 to 125 mV [4] 0 to 65 mV [5] -25 to +25 mV [6] 0 to 20 mV [7] 0.2 to 1 V [8] potentiometer <1k [9] potentiometer <10k	[0] displayed input [1] non displayed input	[0] Without decimal point [1] 1 decimal place [2] 2 decimal [3] 3 decimal

To valid that code, press : [↵] key.

3.4.4 LOGIC INPUT

INPUT LOGIC			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
[6] Logic input	[0]	[0] Display status 0 or 1 [1] No display	[0] Without decimal point [1] 1 decimal place [2] 2 decimal [3] 3 decimal

To valid that code, press : [↵] key.

3.4.5 PROCESS VARIABLE SPAN DEFINITION

3.4.5.1 Zero point [MES._]

Selectable on the whole scale according to the decimal point position you chose.

3.4.5.2 Full span [MES.-]

Selectable on the whole scale according to the decimal point position you chose.

3.4.6 TAG AND UNIT CONFIGURATION

REP

Set in the same way as codes.

For each digit, scroll the ASCII codes with ↑ and ↓.

If you press simultaneously ↑ and ↓ you will have a point on the digit required.

3.5 PROCESS VARIABLE input n° 2 and n° 3

Ditto process variable n° 1.

3.6 Display I

AFF.I

DISPLAY I							
DIGIT N°1		DIGIT N°2		DIGIT N°3*		DIGIT N°4	
1	cyclic display	0	1 channel displayed	0	start on channel 1	0	adjustable variables 1 to N
2	continuous display	1	2 channels displayed	1	start on channel 2	1	adjustable variables 2 to N
		2	3 channels displayed	2	start on channel 3	2	adjustable variables 3 to N
		3	4 channels displayed	3	start on channel 4	3	adjustable variables 4 to N
		4	5 channels displayed	4	start on channel 5	4	adjustable variables 5 to N
		5	6 channels displayed	5	start on channel 6	5	adjustable variables 6 and N
		6	7 channels displayed	6	start on channel 7	6	adjustable variable N
						7	None

N.B : N = the total number of channels you set with digit n°2

* At switch on, the diplay will be on the channel defined by that digit n°3.

Channels display :

The ICP 300 can display up to 7 process variables.
The 1st, 2nd and 3rd are generally coming from the 3 analogue inputs, the following ones from programs (see in chapter 5). If you do not use program, limits the number of displayed channels to your number of inputs (logic or analogue).

Start on channel n :

At switch on, the display will start on the channel you chose. If your display is not cyclic, it will stay on that channel.

Adjustable variables :

If you do not use programs, you must choose digit n°4 equal to the maximum number selected by you. (refer to the previous menu, if digit n°2 = 3 then digit n°4 = 3).

Adjustable variable definition :
Channels can be constants used by a program and adjustable by the user (directly with keyboard). These variable will be assigned to a memory n and will be displayed on the channel (n+1). Ex : memory n°6 value displayed on channel 7.

To valid press ↵.

3.7 Display P

AFF.P

This board define the decimal point position on channel 4, 5, 6 and 7.

DISPLAY P			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
1 channel 4 Display=xxxx 2 channel 4 Display=xxx.x 3 channel 4 Display=xx.xx 4 channel 4 Display=x.xxx	0 channel 5 Display=xxxx 1 channel 5 Display=xxx.x 2 channel 5 Display=xx.xx 3 channel 5 Display=x.xxx	0 channel 6 Display=xxxx 1 channel 6 Display=xxx.x 2 channel 6 Display=xx.xx 3 channel 6 Display=x.xxx	0 channel 7 Display=xxxx 1 channel 7 Display=xxx.x 2 channel 7 Display=xx.xx 3 channel 7 Display=x.xxx

To valid that code, press : ↵ key.

3.8 Special functions

SPEC

3.8.1 CONFIGURATION

SPEC			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
1 No linearisation N°1	0 No linearisation N°2	0 no alarm	0 display cycle 2 sec
2 1 segment	1 1 (2) segment	1 1 alarm	1 cycle 4 sec
3 2 segments	2 2 (4) segments	2 2 alarms	2 cycle 8 sec
4 3 segments	3 3 (6) segments	3 3 alarms	3 cycle 16 sec
5 4 segments	4 4 (8) segments	4 4 alarms	
6 5 segments	5 5 (10) segments	5 5 alarms	
7 6 segments	6 6 (12) segments	6 6 alarms	
		7 7 alarms	

N.B : If you do use the first linearisation board (digit n°1 = 1) then the second one can be composed of 6 segments more.

3.8.2 LINEARISATION N°1

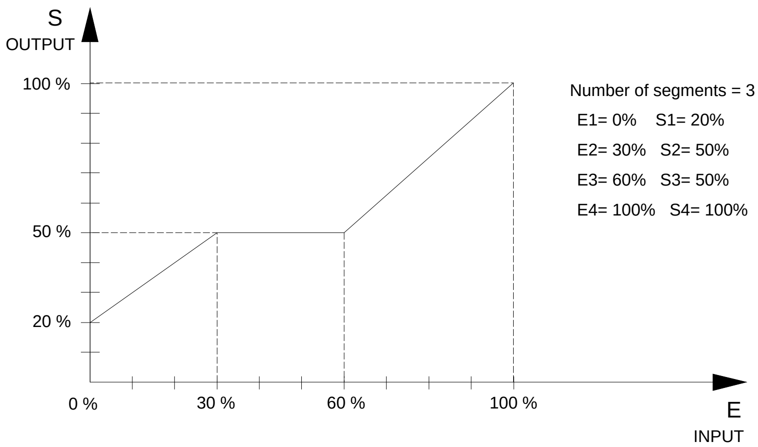
E1.n **and** S1.n

The couple number is equal to the nuber of segments + 1.

Each couple of points is defined by : E1.n = Input
S1.n = Linearised output

Values are in % and selectable between 0 and 100 %

For 3 segments :



Press ↵. The block configuration is saved.

Only with ***LIN.1*** instruction you will access to the linearisation function.

Example : Channel n°1 process variable (scale 0 to 100) is linearised then outputs through the retransmission output on slot 1 (scale 0 to 100).

0	CHA.A	MRE.1	put the relative value (0 < mes < 1) n° 1 in the calculation accumulator
1	LIN.1	ACC.A	linearise with the law N°1 the calculation accumulator value
2	MUL.A	ETE.1	bring back the value in the scale (0 to 100)
3	RAN.A	OUT.1	assigne the current output on slot 1
4	RAN.A	DISPLAY.3	display the result on display 3
5	JUMP	0	do it again at each cycle.

3.8.3 LINEARISATION N°2 E2.n and S2.n

Ditto linearisation n°1.

3.8.4 ALARMS NUMBER

According to the choosen code, the apparatus will propose you n alarms to configure.

3.9 Alarm n° 1 ALR.1

3.9.1 CONFIGURATION MENU

ALARM n° 1			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
1 high setpoint set	0 on channel 1	0 Action on signal lights and relays	0 relay 1 Positive logic
2 low setpoint set	1 on channel 2	1 action on signal lights	1 relay 1 negative logic
3 high setpoint calculated within a program AL0.A	2 on channel 3	2 action on relays	
4 low setpoint calculated within a program AL0.A	3 on ME0.A (program memory)	3 no action	

Alarm n° 1 is associated to relay n° 1 and signal lights n° 1 only.

If digit n°1 is equal to 3 or 4, the setpoint must be assigned to AL0.A memory. (AL1.A for alarm 2, etc...)

If the alarm is assigned to a program value, this value must be stored in ME0.A .(ME1.A for alarm 2, etc...)

To valid press ↵.

3.10 Alarms 2 to 5

ALR.n

Same configuration as alarm n° 1.

These alarms can be used even if no relay are fitted.

3.11 Alarms 6 and 7

ALR.n

Same configuration as alarm n° 1.

These alarms can not act directly on relays or signal lights. As the previous one they can be used within a program.

3.12 Analogue output n° 1

OUT.1

The following menu appears only if there is an analogue output connected in slot n° 1

3.12.1 CONFIGURATION

3.12.1.1 Current output menu example

CURRENT OUTPUT			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
1 retrans. channel 1 2 retrans. channel 2 3 retrans. channel 3 4 retrans. variable OUT.1	0 4-20 mA 1 0-20 mA	0	0

To valid this code press ↵ key.

3.12.1.2 Voltage output menu example

VOLTAGE OUTPUT			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
1 retrans. channel 1 2 retrans. channel 2 3 retrans. channel 3 4 retrans. variable OUT.1	0 2-10 V 1 0-10 V 2 1-5 V 3 0-5 V	0	0

To valid this code press ↵ key.

3.12.2 PROCESS VARIABLES SPAN DEFINITION

3.12.2.1 output n° 1 zero — OUT._

Selectable over the whole display range.

The decimal point position can be changed by pressing simultaneously ↑ and ↓ key.

3.12.2.2 output n° 1 full span — OUT.-

Selectable over the whole display range.

The decimal point position can be changed by pressing simultaneously ↑ and ↓ key.

3.13 Analogue output n° 2 — OUT.2

This menu appears only if an analogue output menu is connected in slot n° 2
Same configuration as the previous one

3.14 Digital RS

RS

DIGITAL RS			
DIGIT N°1	DIGIT N°2	DIGIT N°3	DIGIT N°4
1 300 b	0 Protocol MODBUS	0 ASCII 7 bits without parity 2 stop bits *or 7 bits 2 stop bits in printer mode*	0 relative value between 0 and 65535 *1 sec*
2 600 b	1 Printer mode	1 ASCII 7 bits even parity 2 stop or *8 bits 1 stop without parity in printer mode*	1 relative value between 0 and 32767 *2 sec*
3 1200 b		2 ASCII 7 bits odd parity 2 stop	2 relative value between 0 and 16383 *3 sec*
4 2400 b		3 ASCII 7 bits even parity 1 stop	3 relative value between 0 and 4095 *0,1 sec*
5 4800 b		4 ASCII 7 bits odd parity 1 stop	
6 9600 b		5 RTU 8 bits without parity 1 stop	
		6 RTU 8 bits even parity 1 stop	
		7 RTU 8 bits odd parity 1 stop	
		8 RTU 8 bits without parity 2 stop	

Used with a printer with direct digital communication : DIGIT n° 2 = 1

According to your printer type, you must set the digit n° 3 at 0 or 1 :

0 7 data bits, 2 stop bits without parity
1 8 data bits, 1 stop bit without parity

According to your printer type, digit n° 4 will define the time between two printing.

Using IMPR AFFI instruction within a program (see in § 5.5), you will print all the channels displayed with according units and tags :

	Unit	**Tag**
V1 =	40.5	o C
V2 =	1042	mbAr
V3 =	604	TEMP

You must also enter the number (the position) of the ICP 300 inside the digital link.

For any other RS use, DIGIT n° 2 = 0, see in the specific manual.

4 ADJUSTING AND WORKING MODE

You access to the working mode at the end of configuration.

When "**MODE UTIL**" message appears, press [↵] key.
The apparatus needs few seconds to go to the working mode.

4.1 Working mode

It allows :

Process variable display on the upper indicator, unit or tag on the lower one.

Direct modification of some calculating parameter.

4.1.1 DISPLAYED CHANNELS CHOICE

[MCC] key allows to hold the display on one choosen channel selected with [↑] and [↓] keys.

4.1.2 ADJUSTABLE VARIABLE MODIFICATION

Adjustable variables are constant datas you can modify on front face (without having to enter in constant block configuration) as following :

Lock the display with [MCC]

Select the channel with [↑] and [↓].

Press [↵] and set the parameter with [↑] and [↓].

Valid with [↵]. The new value you have set is stored in the corresponding memory.

4.2 Ajusting mode

4.2.1 ACCESS TO ADJUSTING MODE

To access to adjusting mode, press simultaneously [↓] and [MCC] key.

4.2.2 KEYBOARD USE

In adjusting mode, the dialogue is composed of different menu. The transition from one to another is made with [↑] and [↓].

You enter in a block with [↵] keys.

The corresponding mnemonic appears on the lower indicator and the value on the upper one. To modify these parameters, use [↑] and [↓] and valid with [↵]. The following parameter will appear automatically.

4.2.3 ALARM MENU

ALRM code appears on the lower display.

Pressing on ↵ :

ALR.1	This is the setpoint value of alarm n°1. It can be adjustable if it has been set like this, during configuration (see § 3.9, digit n° 1 = 2). If it is calculated (from a program), it is not adjustable and is equal to AL0.A value.
HYS.1	HYS.1 selectable on the whole process variable range.

The ICP 300 can manage up to 7 alarms. (You have to set the same number of ALR.n and HYS.n)

Setpoints used for programmed variable are limited by the display. ***In that case modify the decimal point position using ↑ and ↓.***

4.2.4 SPECIAL FUNCTIONS

F_Sp appears on the lower display.

Pressing ↵ :

input filtering time	filter from 0.25 to 10 sec. With a step on the process variable the new value will be reached (at 1%) at the end of the time you have set previously.
filtering band	selectable from 0 to 100% . The filter will only be effective inside the defined band. This band is defined around the process variable value.

Example : process variable = 20 %, filter = 5 s, band = 5 %
The filter will only be effective with a process variable variation between 15 and 25 %.

4.2.5 LOCKING MENU

SECU code appears on the lower display.

Pressing ↵ :

Code	To modify the locking level enter : 369
Level 0	Nothing locked.
Level 1	You will have to enter "8031" to return to CONFIGURATION.
Level 2	Ditto level 1 + program locked
Level 3	Ditto level 2 + constant datas and timers locking.
Level 4	Ditto level 3 + setpoints and alarm hysteresis locking.

4.2.6 TIMER MENU

TIME appears on the lower display.

Pressing ↵ :

TIME.1	Minutes or secondes selectable from 0 to 9999

4 timers are available : TIM.1, TIM.2, TIM.3, TIM.4 usable by program

4.2.7 CONSTANTS MENU

CST appears on the lower display.

Pressing ↵ :

COn with n from 0 to 9	Each constant data is limited by the display range.

If needed, modify the decimal point position pressing simultaneously ↑ and ↓.

The ICP 300 can use up to 10 constant datas.

4.2.8 PROGRAM n° 1

PRG.1 appears on the lower display.

To enter in this menu press ↵

4.2.8.1 Program start and stop

PROG appears on the lower display. ↑ and ↓ keys are used to "ON" or to "OFF".

Valid with ↵.

If a program is running (RUN) you can hold it (HOLD) pressing ↑ or ↓ but you can not modify it.You can only read it.

Valid with ↵

4.2.8.2 Programmation

See in chapter 5 PROGRAM FUNCTION

4.2.9 PROGRAM n° 2, 3 and 4

ditto program N°1

5 PROGRAM FUNCTION

5.1 Generals

ICP 300 presents 4 programs of 120 steps.

5.1.1 PROGRAM USE

Programs allows calculating operations on analogue or boolean variables, memorisation and the control of 5 relays, 2 current or voltage outputs and 7 displays.

5.1.2 CYCLE TIME

5.1.2.1 ICP 300 working cycle

ICP 300 processing can be devided into in 10 different cycles:

CYCLE NUMBER	***DURING CYCLE ACTION***
**** Cycle N°1***	***Acquisition and processing of process variable channel N°1***
**** Cycle N°2***	***Acquisition and processing of process variable channel N°2***
**** Cycle N°3***	***Acquisition and processing of process variable channel N°3***
**** Cycle N°4***	***Logic inputs acquisition Update of logic informations***
**** Cycle N°5***	***Setpoints and timers processing***
**** Cycle N°6***	***Outputs management***
**** Cycle N°7***	***Program N°1 in progress***
**** Cycle N°8***	***Program N°2 in progress***
**** Cycle N°9***	***Program N°3 in progress***
**** Cycle N°10***	***Program N°4 in progress***

For each cycle the dwell time is about 24 ms.

The total cycle is performed in about 237 ms.

5.1.2.2 Program cycle time

The dwell time for each program is 24 ms (divided into 60 time units , each instruction needs 1 or more unit). So, during the total cycle time of the apparatus, a limited number of instructions can be performed.

If a program contains too many instructions it will be performed during different cycles; you must be careful if your programs follow one another.

N.B : For each instruction, a processing time is given (in time unit). So you can estimate the time needed for your program processing.

5.2 Programming

You define your programs in adjusting mode when they are stopped (OFF).

"PAS" (step) appears on the lower display.

To change a program step use [↑] and [↓]

The step number is displayed on the upper indicator.

On the left digit, a code gives you the action type (see in the following menu).

You can modify it with [MCC]

LEFT DIGIT	**ACTION TYPE**	Action description
F	**Finish**	Pressing [↵] ICP returns to PRG menu beginning.
I	**Insertion**	Pressing [↵], This step is inserted instead of the step which number is displayed, all the next steps are shifted to the end of the program
M	**Modification**	Pressing [↵] , this step replaces the step which number is displayed, the following steps do not move.
E	**Effacement clear**	Pressing [↵] this step is cleared, all the following steps are shifted to the program beginning.Pressing [MCC], you go to the next step without clearing the step in progress.

5.3 Program step modification

Choose the action type M(modification), I(insertion).

Press [↵]. The step n is displayed.

A step is composed of an operating code (green display) and a parameter (red display).

*The **OPERATING CODE** corresponds to the **ACTION** performed*

*The **PARAMETER** is the **VARIABLE CONCERNED** by the action*

You modify the operating code or the parameter (flashing) using [MCC].

[↑] and [↓] displays the different choices for both of them.

Valid the step N with [↵] .

5.4 Parameters

5.4.1 CONTINUOUS PARAMETERS

5.4.1.1 Analogue inputs

The 3 process variable channels are concerned.

code	***description***
MAB.n	Value in physical unit of channel n Example : If channel 1 is a temperature input and the process variable is equal to 320°C, MAB.1 contains 320.
MRE.n	Channel n value replaced in the scale. (between 0 and 1). Example : Channel 3 is a temperature input with a zero point equal to 0 °C and span of 1000°C. For a process variable of 320°C, MRE.3 will contain 0,32 .
EC_.n	Zero point value for channel n
EC-.n	Span value for channel n
ETE.n	Range for channel n

5.4.1.2 Registers (accumulator), memories, constants

code	***description***
ACC.A	Accumulator. All the operation results are stored in that accumulator
MEn.A, **with n from 0 to 9**	This ***memory is saved in case of power supply failure***. It stores the intermediate results. You can set the alarm n+1 on the value in that memory (see in § 3.9)
ALn.A, **with n from 0 to 4**	This analogue variable is ***saved in case of power supply failure***. It stores the intermediate results. Its value can be used as the setpoint value of alarm n+1 . Example : AL3.A is the setpoint of alarm 4. See in § 3.9 and § 3.10
ALn.A, **with n = 5 or 6**	This ***memory is saved in case of power supply failure***. It stores the intermediate results.
COn.A **with n from 0 to 9**	Constants adjustable during the CST menu configuration. See in 4.2.7. It can only be read by program.
C.TO4	This analogue value contains the number of pulses receipted on the logic input n°4 from its last reset.
C.TO5	This analogue value contains the number of pulses receipted on the logic input n°5 from its last reset.

T.TO4	This analogue value contains the time between 2 rising edges detected on the logic input n°4. This function is started by : *ON.b TRIG* instruction.
T.TO5	This analogue value contains the time between 2 rising edges detected on the logic input n°5. This function is started by : *ON.b TRIG* instruction.
T.T54	This analogue value contains the time between the rising edge detected on the logic input n°5 and the first one detected on logic input n°4.
T.CYC	This analogue value is a constant which contains the ICP300 cycle time, TE = 0,237 second. It is used to make an action at a precised time.

5.4.1.3 Analogue outputs

OUT.1	It is the value of the analogue output connected in slot 1. Example : In slot 1 the current output range is 0 to 100,0 and its value is a 4-20mA signal (see in § 3.12). If you set by program **OUT.1** = 50.0, the output will take the value 12 mA. This ***memory is saved in case of power supply failure***.
OUT.2	It is the value of the analogue output connected in slot 2.

5.4.1.4 Display

On AF1.A, AF2.A and AF3.A (display 1, 2 and 3) you can assigned direct inputs or calculated values.
Be careful : You must specify if your direct inputs are diplayed or not (input block, digit n° 3 =1). If you do not, you may have 2 different values on the same display (1 calculated and 1 direct).

AFn.A **with n from 1 to 7**	It is the value on the display n. It can be printed by program if the input n is not displayed.
AFFI	Can only be used with IMPR instruction. Allow the printing of all the channels configured to be displayed.

5.4.2 DISCRETE PARAMETERS

All the logic informations are transformed in 4 boolean values equal to 1 if they are right, 0 if they are wrong.

The update rate is 250ms.

5.4.2.1 Logic inputs

code	***description***
TOn.H **with n from 1 to 5**	Logic input n is energized .
TOn.b	Logic input n is de-energized.
TOn.M	Logic input n has just been energized . This variable is ON for 250 ms only.
TOn.d	Logic input n has just been de-energized . This variable is ON for 250 ms only.

Warning : ***The logic inputs 1, 2 and 3 are available only if they are not used as analogue inputs. If you use 3 analogue inputs and 2 logic inputs, it will be the inputs 4 and 5.***

5.4.2.2 Use Of ↵

You can define a function of that key by program (launch a program for example just by pressing it). This function is valid only when ICP 300 is in working mode.

code	***description***
CLA.H	↵ pressed. Available only in working mode
CLA.b	↵ not pressed. Available only in working mode
CLA.M	↵ has just been pressed. Available only in working mode This variable is ON for 250 ms only.
CLA.d	↵ has just been depressed. Available only in working mode This variable is ON for 250 ms only.

5.4.2.3 Alarms status

code	***description***
ALn.H **with n from 1 to 7**	The variable value is higher than the setpoint.
ALn.b	The variable value is lower than the setpoint.
ALn.M	The variable value just becomes higher than the setpoint. This variable is ON for 250 ms only.
ALn.d	The variable value has just become lower than the setpoint. This variable is ON for 250 ms only.

5.4.2.4 Timers status

code	***description***
TMn.H **with n from 1 to 4**	Timer is started.
TMn.b	Timer is stopped.
TMn.M	Timer has just been started. This variable is ON for 250 ms only.
TMn.d	Timer has just been stopped This variable is ON for 250 ms only.

5.4.2.5 Programs status

code	*description*
PGn.H **with n from 1 to 4**	Program is started.
PGn.b	Program is stopped.
PGn.M	Program has just been started. This variable is ON for 250 ms only.
PGn.d	Program has just been stopped. This variable is ON for 250 ms only.

5.4.2.6 Sensor failure status

A sensor failure is detected when the process variable is out of its set range by ± 3 % .

code	*description*
RPn.H **with n from 1 to 3**	Sensor failure on channel (n)
RPn.b	No failure on channel (n)
RPn.M	Immediate sensor failure on channel (n) This variable is ON for 250 ms only.
RPn.d	channel (n) has just come back from a sensor failure status This variable is ON for 250 ms only.

5.4.2.7 Display status

code	*description*
bLO.b	The display is fixed on a channel (not in cyclic mode). To return to cyclic mode, put this parameter equal to 0.
AFn.b **n from 1 to 7**	The display scan channel n if this parameter is at 1. Automatically, the display is locked and bLO.b is at 1.

5.4.2.8 Registers (accumulator), memories

code	*description*
ACC.b	Logic accumulator Used for boolean calculations
MEn.b **with n from 1 to 3**	Logic memory saved in case of power supply failure. Store intermediate results.
ME4.b	ME4.b is not saved in case of power supply failure. It is a logic memory and furthermore it can indicate out a power supply failure. ***ME4.b is set to 1 each time the power is switch on again.*** So you can launch a cycle as soon as the power is on without doing anything else.
MEn.b **with n = 5**	Binary memory saved in case of power failure. It allows intermediate results storage.
MEn.b **with n from 6 to 8**	Binary memory **not** saved in case of power failure. It allows intermediate results storage.

5.4.2.9 Action on relays

Relays can be energized by alarms or by program. If your relay is energized by program, you must configure your alarm "without action on relays" (see in chapter 3.9.1, digit n˚3 = 1 or 3)

Code	*description*
REL.1	standard relay on main board. Example : Relay n ˚1 supplied **ON.b REL.1** Example : Relay n ˚1 none-supplied **OFF.b REL.1**
REL.2	SLOT 1 1st relay on option board "2 relays" or relay on option board "1 relay" .
REL.3	SLOT 1 2nd relay on option board "2 relays"
REL.4	SLOT 2 1st relay on option board "2 relays" or relay on option board "1 relay".
REL.5	SLOT 2 2nd relay on option board "2 relays"

5.4.2.10 Action on signal lights

Signal lights can be turned on by program or by alarms.
If your light is energized by program, you must configure your alarm "without action on lights" (see in chapter 3.9.1, digit n˚3 = 2 or 3)

code	*description*
LED.1	signal light n˚ 1 EX: signal light n˚ 1 : ON **ON.b LED.1** EX: signal light n˚ 1 : OFF **OFF.b LED.1**
LED.2	ditto signal light n˚ 1
LED.3	ditto signal light n˚ 1
LED.4	ditto signal light n˚ 1
LED.5	ditto signal light n˚ 1

5.4.2.11 Action on timers

code	*description*
TIM.n with n from 1 to 4	If you set your timer in secondes, the accuracy will be equal ± 0,25 sec, in other case at 10 sec. Example : timer 1 started when pressing the key on front panel. Led n° 1 will be ON as long as the timer goes on. **0 CHA.b CLA.M** **1 J0 3** **2 ON.b TIM.1** **3 CHA.b TM1.H** **4 EGA.b LED.1** **5 JUMP 0** The timer will stop when its time is spent or when a **OFF.b TIM.1** instruction is started. If a **ON.b TIM.1** instruction is started before the time is elapsed, the timer will be started again at its beginning. Do not use **EGA.b** instruction to launch a timer.

5.4.2.12 Action on programs

code	*description*
PRG.n **with n from 1 to 4**	A program can be started or stopped by another program. Example : program n° 1 is started with an action on the keyboard **0 CHA.b CLA.M** **1 J0 3** **2 ON.B PRG.1** **3 JUMP 0**

5.4.2.13 Action on time measurement

code	*description*
TRIG	Allows the time measurement on logic inputs 4 and 5.

5.4.2.14 Action on display

code	*description*
AFF.N	Lock the display on the channel which number is in (analogue accumulator +1).

NB : Refer to 5.4.2.7. Status codes may also act on display.

5.5 Operating codes

Operating codes used with ***analogue*** parameters look like : **XXX.A**

Operating codes used with ***logic*** parameters look like : **XXX.b**

For each instruction (or code) you have the time needed (in time unit). If the addition of all those units is upper than 60, your progamm will run during more than one cycle.

5.5.1 ANALOGUE OPERATING CODES

Code	*Nb of time unit*	*description*
CHA.A	**2**	Load the analogue value in the accumulator Example : process variable n° 1 in the analogue accumulator. **CHA.A MAB.1**
ADD.A	**2**	Add the parameter to the accumulator value, and load the result in the accumulator. Example : add process variable n° 2 to process variable n° 1 **CHA.A MAB.1**o **ADD.A MAB.2**
SUB.A	**2**	Substract the parameter to the accumulator value, and load the result in the accumulator. Example : substract process variable n° 2 to process variable n° 1 **CHA.A MAB.1** **SUB.A MAB.2**
DIV.A	**4**	Divide the accumulator by the parameter and load the result in the accumulator. Example : Divide process variable n° 1 by process variable n° 2 **CHA.A MAB.1** **DIV.A MAB.2**
MUL.A	**2**	Multiply the accumulator value by the parameter and load the result in the accumulator. Example : Multiply process variable n° 1 by process variable n° 2 **CHA.A MAB.1** **MUL.A MAB.2**
CMP.A	**2**	Compare the accumulator value with the parameter. If **(ACC.A minus Parameter) higher or equal to 0** the logic accumulator is set to 1. Its value is not modified. Example : signal light n° 1 is ON if process variable n° 1 is equal or higher than process variable n° 2, it is off in the other case. **CHA.A MAB.1** **CMP.A MAB.2** **EGA.b LED.1**

IMPR	**2**	Send on digital printer download all the displayed values to digital printer. Example : **IMPR AFFI** ***The delay between 2 printing instructions must be higher than 10 secondes.*** You can also print programs : **PRG1, PRG2, PRG3 and PRG4.**
CUM.A	**8**	Precise addition used to add up to 1000 time the same value without error. This operation cumulates the accumulator value. The result is loaded in the defined memory. ***This operation uses 3 memories.*** ***The 1st one contains the sum of the others. To clear the result you must clear the 3 memories (set it to 0)*** ***If the memory instruction is used, the 3 memories can not be used by other programs*** Example : Cumulate process variable n° 1 in memory n° 0. **CHA.A MAB.1** **CUM.A ME0.A**
RAC.A	**15**	Square root extraction. The result replaces the previous value of the parameter. ***It is possible on relative value only (0 < n < 1).*** Ex: process variable square root **CHA.A MRE.1** **RAC.A ACC.A** Load the process variable n° 1 in relative value in the accumulator. Extract the square root from the accumulator value. The result is loaded in the accumulator.
CSG.A	**2**	Change the parameter sign. The result replaces the previous value of the parameter. Ex: **CSG.A ME2.A** The sign of the memory 2 is directly change.
ABS.A	**2**	Extract the absolute value. The result replaces the previous value of the parameter.
N-1.A	**5**	1/x. The result replaces the previous value of the parameter.
INC.A	**2**	Add 1 to the parameter. The result replaces the previous value of the parameter.
DEC.A	**2**	Substract 1 from the parameter. The result replaces the previous value of the parameter.
RA0.A	**2**	Set to 0.
LIN.1	**10**	Take the value after linearisation n°1. The parameter must be between 0 and 1. The result replaces the previous value of the parameter.
LIN.2	**10**	Take the value after linearisation n°2. The parameter must be between 0 and 1. The result replaces the previous value of the parameter.

RAN.A	**2**	The accumulator value replaces the previous value of the parameter.

The resolution for analogue calculation is 1 for 65535.
You can not add a number which is 65535 time bigger.
Ex: 70000 + 1= 70000 instead of 70001
Ex: 650,85 + 1,895= 652,734 instead of 652,745

5.5.2 LOGIC OPERATING CODES

Code	***Nb of time unit***	***description***
CHA.b	**1**	Load the parameter in the logic accumulator.
AND.b	**1**	Realise an AND function between the logic parameter and the accumulator. The result is load in the logic accumulator.
OR.b	**1**	Realise an OR function between the logic parameter and the accumulator. The result is load in the logic accumulator.
XOR.b	**1**	Realise an "exclusive" OR function between the logic parameter and the accumulator. The result is load in the logic accumulator.
RAN.b	**1**	Load the accumulator in the memory.
RA1.b	**1**	Set to 1.
RA0.b	**1**	Set to 0.
NOT.b	**1**	Realise a NOT Function.

5.5.3 OPERATING CODES FOR JUMP ACTION

Code	***Nb of time unit***	***description***
J0.b	**1**	Read the accumulator and if it is equal to 0, go to the defined step.
J1.b	**1**	Read the accumulator and if it is not equal to 0, go to the defined step.
JUMP	**1**	Jump directly to the step defined. If the step number is lower than the number of the step in progress then that step will be performed during the following cycle (250 ms later). ***This instruction must be at the end if you want your program to run completely during each cycle.***

5.5.4 DIRECT OPERATING CODES

These codes act directly on :

The 5 signal lights
The 5 relays
The 4 timers
The 4 programs

Available actions :

Code	*Nb of time unit*	*description*
ON.b	***1***	Relays, signal lights, timers or programs TURNED ON.
OFF.b	***1***	Relays, signal lights, timers or programs OFF.
HOLD.b	***1***	Relays, signal lights, timers or programs HELD.
RUN.b	***1***	Relays, signal lights, timers or programs RELEASED.
EGA.b	***1***	Relays, signal lights or programs assigned to the logic accumulator value.

5.6 Program examples

5.6.1 CALCULATING PROGRAM

Purpose : Average of process variables n° 1 and n° 2.

The result must be displayed on channel 4 and outputs on output n°1.

Step	***Operating code***	***parameter***	***description***
0	**CHA.A**	**MAB.1**	Load process variable n° 1 in the accumulator
1	**ADD.A**	**MAB.2**	Add process variable n° 2 to the accumulator value.
2	**DIV.A**	**CO0.1**	Divide the accumulator value by a constant set at 2,000 (via the key board see in § 4.2.7)
3	**RAN.A**	**AF4.A**	Display the result on channel n° 4.
5	**RAN.A**	**OUT.1**	Output the result on output n° 1.
6	**JUMP**	**0**	Return to the program beginning. It will start at step 0, 250 ms later (next cycle).

5.6.2 LOGIC PROGRAM

Purpose : Signal light 1 is ON if there is an alarm or a sensor failure on process variable n° 1.
The alarm setpoint is set during the alarm (ALRM) menu configuration.

Step	***Operating code***	***parameter***	***description***
0	**CHA.b**	**AL1.H**	Load flag of alarm n°1 (0 or 1) in the logic accumulator.
1	**OR.b**	**RP1.H**	OR with the flag of failure on channel n° 1
2	**EGA.b**	**LED.1**	The result is assigned to the led n° 1
3	**JUMP**	**0**	Return to the program beginning. It will start at step 0, 250 ms later (next cycle).

5.6.3 TIMER PROGRAM

Purpose : Pressing ↵ , the relays R1 is energized for 20 secondes.

Step	*Operating code*	*parameter*	*description*
0	**CHA.b**	**CLA.M**	The logic parameter loaded is : ↵ has just been pressed This information is valid for 250 ms only.
1	**J0**	**3**	If the key is not pressed, the program jumps to the step n° 3.
2	**ON.b**	**TIM.1**	Start the timer. Its time has been set during TIME menu configuration.
3	**CHA.b**	**TM1.H**	The logic parameter loaded is : timer is running.
4	**EGA.b**	**REL.1**	The result is assigned to relay 1.
5	**JUMP**	**0**	Return to the program beginning. It will start at step 0, 250 ms later (next cycle).

5.6.4 IMPULSE INPUTS PROGRAM

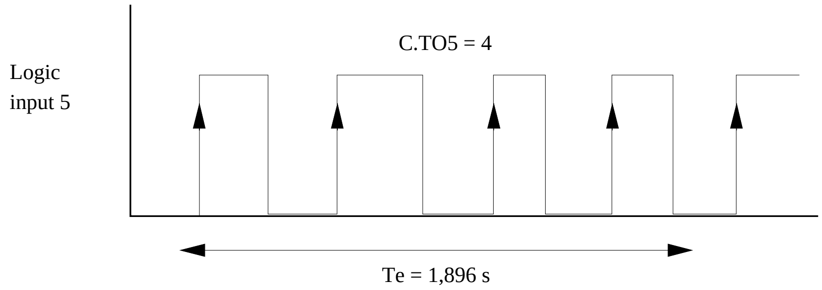
To define a frequency, you have to count the number of impulses during a time. Program n°1 define this basic time and program n°2 calculates the frequencies and displays them on channels 2 and 3.

Program N°1 : Definition of a basic time equal to 1,896 sec

Step	*Operating Code*	*Parameter*	*Description*
0	CHA.A	ME0.A	Load logic memory n°0 in accumulator
1	CMP.A	CO0.A	compare to CO0.A = 7 (1st running equal to 0)
2	J1.b	5	If ME0.A>CO0.A, jump to step 5
3	INC.A	ME0.A	increase memory 0 of 1 at each running
4	JUMP	0	Jump to step 0
5	RA0.A	ME0.A	The fist 8 passages has been run(0,237 x 8 = 1.896 sec)
6	ON.b	PRG.2	Launch the frequence calculation
7	JUMP	0	Return to step 0

Program N°2 : Frequence calculation

step	***Operating Code***	***Parameter***	***Description***
0	CHA.A	C.TO5	load the constant C.T05 (number of impulses on logic input 5)
1	DIV.A	CO1.A	divide by TE = 1.896 sec
2	RAN.A	AF2.A	display on channel 2 the frequency of logic input 5
3	CHA.A	C.TO4	load constant C.T04 (number of impulses on logic input 4)
4	DIV.A	CO1.A	divide by TE = 1.896sec
5	RAN.A	AF3.A	display on channel 3 the frequency of logic input 4
6	RA0.A	C.TO5	Reset of C.TO5
7	RA0.A	C.TO4	Reset of C.TO4
8	FIN	PROG	



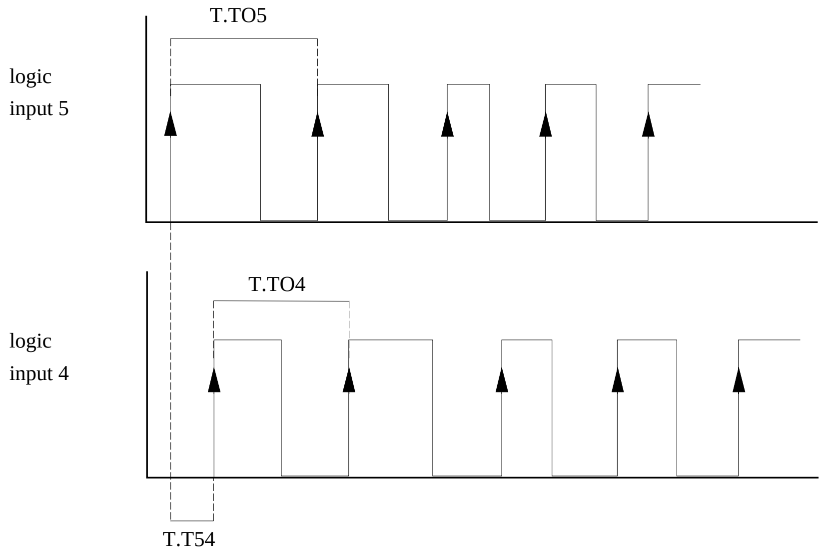
5.6.5 TIME BETWEEN TWO IMPULSES

To define the time between two impulses, we have to realise the measure then read the value inside T.TO5,T.TO4,T.T54

Program N°1

Step	***Operating Code***	***Parameter***	***Description***
0	CHA.b	CLA.M	test if Enter is on
1	J0.b	9	If not, jump to step 9
2	RA0.A	T.TO5	Reset of the time value on logic input 5 between 2 peaks
3	RA0.A	T.TO4	Reset of the time value on logic input 4 between 2 peaks
4	RA0.A	T.TO54	Reset of the time value between 1 peak on logic input 4 and one on logic input 5
5	ON.b	TRIG	measure. The first peak on input 5 will launch the counting.

9	CHA.A	T.TO5	load constant T.T05
10	RAN.A	AF2.A	display on channel 2 the time between 2 peaks on inputs 5.
11	CHA.A	T.TO4	load constant T.T04
12	RAN.A	AF3.A	display on channel 3 the time between 2 peaks on inputs 4.
13	CHA.A	T.T54	load constant T.T54
14	RAN.A	AF4.A	display the time value between 1 peak on logic input 4 and one on logic input 5
15	JUMP	0	jump to next cycle



6 MESSAGES

MESSAGE	DISPLAY	SIGNIFICATION	ACTION
U1.n	HIGH	Version 1.n	
F1.n	HIGH	Version 1.n Saving error on novram	Check CST, TIME, PRG1, PRG2, PRG3, PRG4 menus.
S1.n	HIGH	Version 1.n No novram	Check the component
FLASHING DISPLAY	HIGH	Sensor failure on displayed process variable (± 3 % out of range)	Check the sensor, its connection and your configuration
RAM ERR	HIGH LOW	Parameter not saved on novram.	Enter a new value.
ERR.1	LOW	Non adapted board in slot N°1	Change the board in slot N°1.
ERR.2	LOW	Non adapted board in slot N°2	Change the board in slot N°2.
ERR.3	LOW	Boards connected are different from these connected during configuration.	Change the boards or configure again.

7 SPARE PARTS

DESIGNATION	REFERENCE
1 relay board	H10246
2 relays board	H10243
2 relays S board	H10248
Current output board	H10310
Logic output board	H10257
Voltage output board	H10311
Auxiliary power board	H10312
Digital RS 232	H10250
Digital RS 485	H10249
Microprocessor board	H10293
Display panel	H10294
Complete case	H10295
Clamps	H20298
Capacitor device	H90064

8 NOMENCLATURE

SLOT 1 OUTPUT
0 None
1 1 relay *
2 2 relays **
3 Current
4 Logic
5 Voltage
6 Auxiliary power
7 2 relays S

SLOT 2 OUTPUT
0 None
1 1 relay *
2 2 relays **
3 Current
4 Logic
5 Voltage
6 Auxiliary power
7 Digital RS 232
8 Digital RS 485
9 2 relays S

POWER SUPPLY
0 115/230 Volts 50/60 Hz
1 12 VDC
2 24 VDC
3 48 VDC

SOFTWARE VERSION
1 standard

ICP 300 2 3 0 1

***** 1 changeover relay
****** 2 relays with common connection

Example : **ICP 300 2301**, 3 process variable inputs, 2 logic inputs and 1 relay. 2 relays in slot 1 and 1 current output in slot 2. Power supply 115/230 VOLTS 50/60 Hz, SOFTWARE version 1.

CONTENTS