KOSMOS SERIES

MODBUS RTU
COMMUNICATIONS PROTOCOL

CE

Y2K OK

PI-MBUS-300 Rev.D

INSTRUCTIONS MANUAL

EDITION: MARCH 2000
CODE: 30727077

MODBUS RTU
English

INDEX

1. INTRODUCTION

1.1. DESCRIPTION

The MODBUS RTU protocol can be used indiferently with RS232C and RS485 serial interfaces, allowing in mode RS485, a master device to address up to 99 instruments.
The link is half-duplex type with selectable baud rates from 1200 to 19200 bits/s.

Communications are based in a master-slave technique in which the master dispositive requests data from the addressed slave which only responds, never initiates the transaction. The master can address individual slaves to request or to send data or it can request an action to be taken by one or all the slaves in the network.

1.2. MODBUS RTU PROTOCOL

The modbus RTU protocol is a serial data transmission format widely used in communications with PLC's but easy adaptable to other types of remote units thanks to its particular message structure (it doesn't operate with variables but with memory addresses).

The implementation of a universal standard protocol such as modbus allows the instrument be connected to existing systems without need for creating specific communication logic programs.
Moreover, the quantity and variety of processable data can be infinite since it is no more necessary to especify the desired parameter or parameters but only the first memory address and the quantity to transmit.

The following definitions of modbus protocol are given adapted to the KOSMOS SERIES instruments.

The MODBUS RTU protocol have no delimiter character at the beginning and the end of a message.
Each frame must be preceeded by a silent interval of at least 3,5 character times and must be finished with a silent of the same duration.

The first character of a frame is the slave address, followed by the function number, the information field and two bytes of error check code (CRC).

Ad	Fn	information	CRC

Ad : slave address
Fn : modbus function
CRC : error check

One character is composed of 10 bits : 1 start bit, 8 data bits and 1 stop bit.

CRC GENERATION *(according to modbus RTU)*

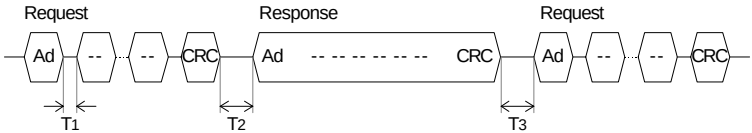
1. Load a 16-bit register with H'FFFF (all '1'). Call it CRC.
2. Exclusive OR the first message byte with the low-order byte of the CRC register and put the result in CRC.
3. Shift the CRC register 1 bit towards the right (to the LSB) writing a '0' in the MSB.
4. If the new LSB is '0' repeat from point 3. If the LSB is '1', exclusive OR the CRC register with the 16-bit value H'A001 (1010 0000 0000 0001).
5. Repeat points 3 and 4 until completing a total of 8 shifts, after what the first byte has been processed.
6. Repeat points 2 to 5 with the second byte and continue these operations until all bytes on the message are processed.
7. Append the obtained CRC to the message so that the low order byte is transmitted first.

CRC Polynome : $2^{15} + 2^{13} + 2^{0}$
CRC Initial value : H'FFFF

1.3. TIMING

The instrument detects the start of a message when receives a valid character (containing its address or the address 00) after a silent interval of at least 3,5 character times.

The end of a message is acknowledged by a silent interval of the same duration.



T1 : time between two characters (min 0, max 3,5CT)
T2 : time between query and response (min 3,5CT)
T3 : time between response and next query (min 3,5CT)

CT = 1 character time

baud (bits/s)	3,5CT
1200	30ms
2400	15ms
4800	8ms
9600	4ms
19200	2ms

1.4. MODBUS FUNCTIONS

The modbus functions supported by the instrument are the following :

CODE	FUNCTION
03	read n words
05	force state
01	read n bits
0F	write n bits

Function 03 Used to read dynamic variables such as display value, peak, valley, tare..., depending on model.

Function 05 Used to force the instrument to take an action such as 'make tare', 'reset tare', 'reset peak', 'reset latched setpoints'..., depending on model.

Function 01 Used to read, in binary format, the instrument programming variables (data contained in the memory) or dynamic variables type ON/OFF (setpoint outputs status).

Function 0F Used to write, in binary format, programming variables in the instrument memory.

1.5. DATA TYPE AND MEMORY LOCATIONS

The instrument manage different types of data which are accessible to the user by programming, display or through the serial channel RS232C or RS485.

The data is located depending on type, in specific memory areas with unique addresses incrementing in 1 per byte from the zero address.

The figure on next page shows the user accessible memory areas, with the data types and modbus functions to be used each zone.

PROGRAMMING DATA IN TABLE 4 *(READ AND WRITE)*	These are the data saved, in binary format in the instrument e2prom memory. The modbus function used to read is 01 and to write is 0F.
RESERVED	
FLOATING POINT VARIABLES IN TABLE 1 *(READ ONLY)*	These are measurement variables such as display, peak..., in floating point format (IEEE single precision) or integer with sign. The modbus function used to read these data is 03.
ON/OFF TYPE VARIABLES IN TABLE 2 *(READ ONLY)*	These are the setpoint outputs and the remote inputs status in binary format. The modbus function used to read these information is 01.
RESERVED	

CONTROL VARIABLES are not located to specific memory addresses but they consist of internal flags that the instrument interprets to take an action, then deactivate the flag (see page 12).

2. FUNCTION FORMATS

2.1. READING DYNAMIC VARIABLES

Dynamic variables are those that vary as a function of the process without direct intervention of the user. These variables normally are the display value, peak, valley..., depending on model. Their address locations are specified in table 1 of the annexe according to model.

These kind of variables are read, by using modbus function 03, in floating point format represented by 4 bytes (2 words) or in integer represented by 2 bytes (1 word).
In this format, the overrange condition is transmitted in form of error response with code 03. (see page 17).

FUNCTION 03 *(READ N WORDS)*

Request frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
slave address	modbus function	1st byte address according to tables 1	number of words (nº bytes / 2)	CRC

Response frame

1 byte	1 byte	1 byte	n bytes	2 bytes
address	function	nº bytes	information read	CRC

NOTE : In general these variables are transmitted without the decimal point of the display.
See indications on table 1 in annexe according to model.

Examples (referenced to model ALPHA-P, see addresses in tables 1a and 1b, annexe1)

Request of the display value (net) to the instrument with address of 01

H'01	H'03	H'01	H'00	H'00	H'02	H'81	H'F7
slave address	modbus function	address 1st byte (d'256 in table1)		number of words =2		CRC	

Response (supposing display = +3210.4)

H'01	H'03	H'04	H'45	H'48	H'A6	H'66	H'94	H'A3
address	function	nº bytes	data (display net value, floating point)				CRC	

Request of the peak and valley values to the instrument with address of 01

H'01	H'03	H'00	H'F0	H'00	H'04	H'44	H'3A
slave address	modbus function	address 1st byte (d'240 in table1)		number of words =4		CRC	

Response (supposing peak=99999, valley=-99999)

H'01	H'03	H'08	H'47	H'C3	H'4F	H'80
address	function	nº bytes	data (peak value)			

H'C7	H'C3	H'4F	H'80	H'D4	H'34
data (valley value)				CRC	

Request of the display value (net) to the instrument with address of 01

H'01	H'03	H'01	H'0E	H'00	H'01	H'E4	H'35
slave address	modbus function	Address 1st byte (d'270 in table1b)		number of words =1		CRC	

Response (supposing display = +1256.3)

H'01	H'03	H'02	H'31	H'13	H'EC	H'19
address	function	nº bytes	data		CRC	

2.2. READING ON/OFF TYPE DYNAMIC VARIABLES

These variables include the state of the setpoint outputs and the remote inputs. The response to a frame pointing to these addresses is one byte which four low order bits represent each one a setpoint output and the four high order bits correspond each one to a remote input pin (see table 2 in annexe).
The bits are at level '1' if the corresponding variable is activated and at level '0' if deactivated.
The modbus function used to read these variables is 01.

FUNCTION 01 *(READ N BITS)*

Request frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
slave address	modbus function	1st byte address according to table2	number of bits (nº bytes x 8)	CRC

Response frame

1 byte	1 byte	1 byte	n bytes	2 bytes
address	function	nº bytes	data field (information)	CRC

Example (referenced to model ALPHA-P, see addresses in table 2, annexe 1)

Request of the four setpoint outputs state to the instrument of address 99

H'63	H'01	H'01	H'0C	H'00	H'08	H'F4	H'61
slave address	modbus function	1st byte address (d'268 as in table2)		number of bits (1byte x 8)		CRC	

Response (supposing SET1=ON, SET2=OFF, SET3=OFF, SET4=ON)

H'63	H'01	H'01	H'09	H'8E	H'36
address	function	nº bytes	data	CRC	

NOTE : These variables (and only these) can be requested in a number of bits from 1 to 8. The response frame will allways include the complete byte information.

2.3. COMMANDS (CONTROL VARIABLES)

Control variables are those that normally are at '0' state and leading them to '1' forces the instrument to take an action after what they return to the '0' state.
Only one command can be written at one time. The address of the variable is replaced by the command indicated in table 3 in annexe according to model.

FUNCTION 05 *(FORCE STATE)*

Request frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
slave address	modbus function	coil address (command)	put bit to '1' (fixed H'FF H'00)	CRC

Response frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
address	function	command	bit to '1' (H'FF H'00)	CRC

Example (referred to model ALPHA-P, see commands in table 3, annexe 1)

Command to the instrument with address of 01 of reseting the tare

H'01	H'05	H'00	H'72	H'FF	H'00	H'2C	H'21
slave address	modbus function	address (command '0r')		put bit to '1' (fixed H'FF H'00)		CRC	

Response

H'01	H'05	H'00	H'72	H'FF	H'00	H'2C	H'21
address	function	command '0r'		bit to '1'		CRC	

A command written to address 00 will be executed by all the units in the network and there won't be response.

2.4. READING AND WRITING PROGRAMMING DATA

Modbus functions 01 and 0F are used to read from and write to the instrument's configuration data zone (static variables) partially or totally. The number of configuration bytes and memory locations varies according to model (see table 4 of annexe depending on model).

FUNCTION 01 *(READ N BITS)*

Request frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
slave address	modbus function	1st byte address according to table2	number of bits (nº bytes x 8)	CRC

Response frame

1 byte	1 byte	1 byte	n bytes	2 bytes
address	function	nº bytes	data field (information)	CRC

Example of partial reading (referenced to model ALPHA-P, see addresses in table 4, annexe 1)

Request of the four logic functions programmed in the instrument number 99

H'63	H'01	H'00	H'E2	H'00	H'20	H'95	H'A6
slave address	modbus function	1st byte address (d'226 as in table2)		number of bits (4bytesx8=d'32)		CRC	

Response (supposing login1=01, login2=02, login4=10, login5=24)

H'63	H'01	H'04	H'01	H'02	H'0A	H'18	H'1E	H'81
address	function	nº bytes	data (value of the logic functions)				CRC	

NOTE : When reading or writing binary data to the instrument's configuration zone, the 'number of bits' field in the request frame must be a multiple of 8. Only complete bytes will be retransmitted from the instrument to the host.

When reading the complete programming data from an instrument, the 'first byte address' field must point to 00. The 'number of bits' and 'number of bytes' fields shall depend on model.
If the instrument's configuration requires more than 2000 bits (=250 bytes, limit of function 0F) the programming data must be requested in two frames at least.
This is not the case of the ALPHA-P model whose number of bits in the configuration area is 1848 (231 bytes x8, see table 4 - annexe 1).

Example of request of the complete configuration of the ALPHA-P with address of 01

H'01	H'01	H'00	H'00	H'07	H'38	H'3F	H'E8
slave address	modbus function	1st byte address (=0)		number of bits (231bytesx8=d'1848)		CRC	

The programming data field carries the contents of the instrument e2prom memory, arranged as indicated in table 4, annexe 1.
The positive sign is represented by the hexadecimal character H'0F and the negative sign by H'0A.

H'01	H'01	H'E7	- - - - - - - - - - -	-	-
address	function	nº bytes	programming data (231 bytes)	CRC	

If starting from address 00, the master requests for a 231, the data field in the response message will carry only 231 bytes without error.
If starting from any other valid address is tried to read a number of bytes that, even being less that 231, exceed from the last address in the valid memory area, the response will carry only the contents between the first requested address and the last valid address (d'230) without generating error.

Although the instrument generates no error when trying to read outside the valid memory range, it is advised, when using the function 01 with a standard program, to ensure this situation is not produced since most of these programs do not admit that the bit count field of the response differs from the bit count of the query.

FUNCTION 0F *(WRITE N BITS)*

Request frame

1 byte	1 byte	2 bytes	2 bytes	1 byte	n bytes	2 bytes
slave address	modbus function	1st byte address according to table4	number of bits (nº bytes x 8)	nº bytes to write	data to write	CRC

Response frame

1 byte	1 byte	2 bytes	2 bytes	2 bytes
address	function	address 1st byte	nº of bits writen	CRC

Examples of partial writing

Programming the setpoint 1 value with +12345 in the instrument of address 01

H'01	H'0F	H'00	H'90	H'00	H'30	H'06
slave address	modbus function	1st byte address (d'144 as in table4)		number of bits (6bytesx8=d'48)		nº bytes to write

H'05	H'04	H'03	H'02	H'01	H'0F	H'67	H'92
data (setpoint value)						CRC	

Response

H'01	H'0F	H'00	H'90	H'00	H'30	H'55	H'F2
address	function	address 1st byte		nº of bits written		CRC	

Programming the tare key lockout (locked=1) in the instrument of address 06

H'06	H'0F	H'00	H'E1	H'00	H'08	H'01	H'01	H'C2	H'A7
slave address	modbus function	1st byte address (d'225 as in table4)		number of bits (1bytesx8=d'8)		nº bytes to write	data	CRC	

Response

H'06	H'0F	H'00	H'E1	H'00	H'08	H'05	H'8C
address	function	address 1st byte		nº of bits written		CRC	

When sending the complete programming data to an instrument, the 'first byte address' fileld must point to 00. The 'number of bits' and 'number of bytes' fields shall depend on model.

If the instrument's configuration requires more than 2000 bits (=250 bytes, limit of function 0F) the programming data shall be sent in two frames at least.

This is not the case of the ALPHA-P model whose number of bits in the configuration area is 1848 (231 bytes x8, see table 4 - annexe 1).

Example of transmission of a complete configuration to the ALPHA-P with address of 01

H'01	H'0F	H'00	H'00	H'07	H'38	H'E7	- - - - - - - - - - -	-	-
slave address	modbus function	1st byte address (=0)		number of bits (231bytesx8=1848)		nº bytes (=231)	programming data field (231 bytes)	CRC	

The programming data field carries all bytes necessary to completely configure the instrument, arranged as indicated in table 4 in annexe 1.

The positive sign is represented by the hexadecimal character H'0F and the negative sign by H'0A.

Response

H'01	H'0F	H'00	H'00	H'07	H'38	H'56	H'29
address	function	1st byte address		nº of written bits		CRC	

If starting from a valid address (between 0 and 230) is tried to write a number of bytes that exceed from the last address in the valid memory area, the information outside this area will be ignored by the instrument and the 'nº of written bits' field of the response message will mark the effectively written bits (less than requested) without error.

Although the instrument generates no error when trying to write outside the valid memory range, it is advised, when using the function 0F with a standard program, to ensure this situation is not produced since most of these programs do not admit that the bit count field of the response differs from the bit count of the query.

2.5. ERROR CONDITIONS AND FRAMES

CODE	ERROR TYPE
01	Illegal function or incompatible with data
02	Illegal data or invalid CRC
03	Instrument overrange

Error 01 :

- Error 01 is produced when the referenced address is outside the valid range or the allowable range for the requested function.
- Error 01 is produced when the function is not supported by the instrument (01, 0F, 03 or 05).

Error 02 :

- Error 02 is produced when the number of bytes in a 'write n bits' request frame exceeds from the maximum number of writable bytes or from the limit of 250.
- Error 02 is produced when the calculated CRC doesn't match the received message CRC.

Error 03 :

- Error 03 indicates overrange condition when reading dynamic variables using function 03.

ERROR MESSAGES FORMAT

Response frame

1 byte	1 byte	1 byte	2 bytes
slave address	function +H'80	error code	CRC

Example

Illegal data error (code 02) message from slave 01 in response to a 03 function

H'01	H'83	H'02	H'00	H'2C
address	H'03+H'80	code	CRC	

3. USING STANDARD COMMUNICATIONS PROGRAMS

3.1. INTRODUCTION

It is found in the market a wide variety of standard programs that allow to collect data through the serial port of a computer to be used in a virtual control panel on the screen of the PC. The modbus communications protocol is used as a universal tool for connecting these programs with all types of remote instrumentation capable of managing this protocol.
In general, such programs continuously scan, at a fixed rate, the setup addresses according to the type of data inside. The type of the data depends on the modbus function used to request it.
The information is continuously updated in the data bus according to the frame format described in section 2. The program user only has to extract each variable from the frame to be represented on the PC screen in the desired format.
Due to that not always the standard modbus data formats match with those of most instrumentation units, in the case of KOSMOS instruments, the data extraction must be made under the following considerations.

3.2. COLLECTING FLOATING POINT VARIABLES USING FUNCTION 03

In modbus format, the variables by function 03 are represented by words (1 word=2 bytes) and they are contained in addresses that increment in 1 per word, that is, each address contains 2 bytes.
In the instrument, the addresses are incremented in 1 per byte that means that each address contains one byte.
The result of this is that a floating point variable takes 4 addresses in the instrument and 2 addresses in modbus representation. An integer variable takes 2 addresses in the instrument and 1 address in modbus representation.

When requesting a single variable, the address pointed by the ‘first byte address’ field is the same where the logic program user can find the response to be represented on the screen afterwards.

Address of floating point variables in the instrument		Address of the variables in standard modbus format	
256	NET	256	NET
257		257	
258			
259			

When applying for more than one variable in the same message, the first one is found in the address pointed by the ‘first byte address’ in the query message, but the following are contained in addresses different from those given in table 1 of the annexe.

Address of floating point variables in the instrument		Address of the variables in standard modbus format	
252	GROSS	252	GROSS
253		253	
254		254	NET
255		255	
256	NET		
257			
258			
259			

This happens because a 4-byte floating point variable is represented internally with 4 addresses (1 per byte) in the instrument memory and with 2 addresses (1 per word) in the screen of the PC.

3.3. COLLECTING DATA IN BINARY FORMAT USING FUNCTIONS 01 & 0F

In general, functions 01 and 0F are not with a standard program since they are used to manage static variables contained in the instrument's memory (except ON/OFF status variables) and these kind of programs continually scan requested addresses to detect changes.

Nevertheless, in some cases it may be interesting to have some specific parameter accessible to be changed on the screen of the PC. For example, a user may find useful to be able to modify one setpoint value, or to activate/inhibit one alarm indication, or to have two different combinations of logic functions to alternate according to process needs, etc.

Since standard modbus functions 01 and 0F are to process bit informations, generally a standard logic program will represent data in binary format giving a different address to each bit, while in the instrument each consecutive address is referred to one byte.
This is not a problem as for the data frames since the transmission and the reception of data is made in blocks of 8 bits, that is, at least 1-byte data is carried out in the information field.
To view such data in the PC screen, the following considerations must be kept in mind.

Examples :

Suppose that you want to have a virtual switch selector on the PC screen to be able to change the filter-P level according to the process evolution :
In case of an ALPHA-P model, the filter-P level information is contained in the memory address d'222 (H'DE).

When using functions 01 or 0F from a host computer, each bit is placed in a different address so that the address d'222 contains the least significant bit of the requested 8-bit byte.
To represent the filter-P value in hexadecimal format, the user should call up 8 addresses starting from d'222 to the PC screen.

Address of binary variables in the instrument		Address of the variables in modbus standard format	
222	FILTER-P	222	BIT 0 FILTER-P
		223	BIT 1 FILTER-P
		224	BIT 2 FILTER-P
		225	BIT 3 FILTER-P
		226	BIT 4 FILTER-P
		227	BIT 5 FILTER-P
		228	BIT 6 FILTER-P
		229	BIT 7 FILTER-P
223	FILTER-E	230	BIT 0 FILTER-E
		231	BIT 1 FILTER-E

The logic program may have the ability to allow combination of binary data contained in 8 consecutive adresses (d'222 to d'229) so that a 1-byte variable could be viewed like one decimal number.

When reading ON/OFF variables (state of setpoint outputs and remote inputs), the requested information is in each bit of a single data byte in the response frame. Suppose that the response to a query about the state of the four setpoints is the following frame as indicated in page 11 (the byte address is d'268) :

H'63 H'01 H'01 H'09 H'8E H'36

To view the required information on the screen (for example, in form of colour indicators), it would be necessary to extract the bit values from address d'268 (set1), d'269 (set2), d'270 (set3) and d'271 (set4).
Only in case of ON/OFF type variables will be possible to ask for a number of bits less than 8.

ANNEXE 1 ALPHA-P

MEMORY LOCATIONS FOR MODEL ALPHA-P

READ ONLY DYNAMIC VARIABLES

TABLE 1a IN FLOATING POINT FORMAT

BYTE Nº	VARIABLE
240	PEAK
244	VALLEY
248	TARE
252	GROSS
256	NET
260	INPUT
264	DISPLAY

Variables in floating point format are transmitted without the decimal point of the display. When using a standard logic program it may be possible to manipulate these variables to read them in the same display format.

TABLE 1b IN INTEGER FORMAT

BYTE Nº	VARIABLE
270	DISPLAY
272	SETPOINT 1
274	SETPOINT 2
276	SETPOINT 3
278	SETPOINT 4

In models Alpha-P, **from versión P9**, it is possible to use function 03 to request the display and the setpoints values with sign in 'integer' format (2 bytes= 1 word). The addresses are specified in table 1b.

TABLE 2 READ ONLY DYNAMIC VARIABLES TYPE ON/OFF

BYTE Nº	VARIABLE	
268	bit 0 (LSB)	Setpoint1 status (0=OFF, 1=ON)
	bit 1	Setpoint2 status (0=OFF, 1=ON)
	bit 2	Setpoint3 status (0=OFF, 1=ON)
	bit 3	Setpoint4 status (0=OFF, 1=ON)
	bit 4	Remote input PIN1 status (0=OFF, 1=ON)
	bit 5	Remote input PIN2 status (0=OFF, 1=ON)
	bit 6	Remote input PIN4 status (0=OFF, 1=ON)
	bit 7 (MSB)	Remote input PIN5 status (0=OFF, 1=ON)

TABLE 3 CONTROL VARIABLES

COMMAND		FUNCTION
0n =	110	reset setpoints latch
0p =	112	reset peak
0r =	114	reset tare
0t =	116	tare
0v =	118	reset valley

TABLE 4 STATIC VARIABLES

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
0	INPUT POINT 1	digit 0
1	[6]	digit 1
2		digit 2
3		digit 3
4		digit 4
5		sign
6	INPUT POINT 2	digit 0
7	[6]	digit 1
8		digit 2
9		digit 3
10		digit 4
11		sign
12	INPUT POINT 3	digit 0
13	[6]	digit 1
14		digit 2
15		digit 3
16		digit 4
17		sign
18	INPUT POINT 4	digit 0
19	[6]	digit 1
20		digit 2
21		digit 3
22		digit 4
23		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
24	INPUT POINT 5	digit 0
25	[6]	digit 1
26		digit 2
27		digit 3
28		digit 4
29		sign
30	INPUT POINT 6	digit 0
31	[6]	digit 1
32		digit 2
33		digit 3
34		digit 4
35		sign
36	INPUT POINT 7	digit 0
37	[6]	digit 1
38		digit 2
39		digit 3
40		digit 4
41		sign
42	INPUT POINT 8	digit 0
43	[6]	digit 1
44		digit 2
45		digit 3
46		digit 4
47		sign
48	INPUT POINT 9	digit 0
49	[6]	digit 1
50		digit 2
51		digit 3
52		digit 4
53		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
54	INPUT POINT 10	digit 0
55	[6]	digit 1
56		digit 2
57		digit 3
58		digit 4
59		sign
60	INPUT POINT 11	digit 0
61	[6]	digit 1
62		digit 2
63		digit 3
64		digit 4
65		sign
66	INPUT POINT 12	digit 0
67	[6]	digit 1
68		digit 2
69		digit 3
70		digit 4
71		sign
72	DISPLAY POINT 1	digit 0
73	[6]	digit 1
74		digit 2
75		digit 3
76		digit 4
77		sign
78	DISPLAY POINT 2	digit 0
79	[6]	digit 1
80		digit 2
81		digit 3
82		digit 4
83		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
84	DISPLAY POINT 3	digit 0
85	[6]	digit 1
86		digit 2
87		digit 3
88		digit 4
89		sign
90	DISPLAY POINT 4	digit 0
91	[6]	digit 1
92		digit 2
93		digit 3
94		digit 4
95		sign
96	DISPLAY POINT 5	digit 0
97	[6]	digit 1
98		digit 2
99		digit 3
100		digit 4
101		sign
102	DISPLAY POINT 6	digit 0
103	[6]	digit 1
104		digit 2
105		digit 3
106		digit 4
107		sign
108	DISPLAY POINT 7	digit 0
109	[6]	digit 1
110		digit 2
111		digit 3
112		digit 4
113		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
114	DISPLAY POINT 8	digit 0
115	[6]	digit 1
116		digit 2
117		digit 3
118		digit 4
119		sign
120	DISPLAY POINT 9	digit 0
121	[6]	digit 1
122		digit 2
123		digit 3
124		digit 4
125		sign
126	DISPLAY POINT 10	digit 0
127	[6]	digit 1
128		digit 2
129		digit 3
130		digit 4
131		sign
132	DISPLAY POINT 11	digit 0
133	[6]	digit 1
134		digit 2
135		digit 3
136		digit 4
137		sign
138	DISPLAY POINT 12	digit 0
139	[6]	digit 1
140		digit 2
141		digit 3
142		digit 4
143		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
144	SETPOINT 1	digit 0
145	[6]	digit 1
146		digit 2
147		digit 3
148		digit 4
149		sign
150	SETPOINT 2	digit 0
151	[6]	digit 1
152		digit 2
153		digit 3
154		digit 4
155		sign
156	SETPOINT 3	digit 0
157	[6]	digit 1
158		digit 2
159		digit 3
160		digit 4
161		sign
162	SETPOINT 4	digit 0
163	[6]	digit 1
164		digit 2
165		digit 3
166		digit 4
167		sign
168	MODE SETPOINT 1	0=LED indication, 1=LED indication + display flash
169	[5]	0=net, 2=gross, 3=peak, 4=valley
170		0=dly, 1=hys_1, 2=hys_2
171		0=hi, 1=lo
172		0=off, 1=on, 2=latch

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
173	MODE SETPOINT 2	0=LED indication, 1=LED indication + display flash
174	[5]	0=net, 1=track set, 2=gross, 3=peak, 4=valley, 5=track auto
175		0=dly, 1=hys_1, 2=hys_2
176		0=hi, 1=lo
177		0=off, 1=on, 2=latch
178	MODE SETPOINT 3	0=LED indication, 1=LED indication + display flash
179	[5]	0=net, 2=gross, 3=peak, 4=valley
180		0=dly, 1=hys_1, 2=hys_2
181		0=hi, 1=lo
182		0=off, 1=on, 2=latch
183	MODE SETPOINT 4	0=LED indication, 1=LED indication + display flash
184	[5]	0=net, 1=track set, 2=gross, 3=peak, 4=valley
185		0=dly, 1=hys_1, 2=hys_2
186		0=hi, 1=lo
187		0=off, 1=on, 2=latch
188	DELAY / HYSTERESIS	digit 0
189	SETPOINT 1	digit 1
190	[4]	digit 2
191		digit 3
192	DELAY / HYSTERESIS	digit 0
193	SETPOINT 2	digit 1
194	[4]	digit 2
195		digit 3
196	DELAY / HYSTERESIS	digit 0
197	SETPOINT 3	digit 1
198	[4]	digit 2
199		digit 3
200	DELAY / HYSTERESIS	digit 0
201	SETPOINT 4	digit 1
202	[4]	digit 2
203		digit 3

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
204	DISPLAY VALUE FOR	digit 0
205	ANALOG OUTPUT LO	digit 1
206	[6]	digit 2
207		digit 3
208		digit 4
209		sign
210	DISPLAY VALUE FOR	digit 0
211	ANALOG OUTPUT HI	digit 1
212	[6]	digit 2
213		digit 3
214		digit 4
215		sign
216	ANALOG OUTPUT TYPE [1]	0=0-10V DC, 1=4-20mA DC
217	ANALOG OUTPUT FILTER [1]	0=filter off, 1=filter on
218	INPUT TYPE [1]	0=volts, 1=milliamperes, 2=potentiometer
219	INPUT RANGE [1]	0=10V/20mA, 1=1V/1mA
220	EXCITATION [1]	0=24V, 2=10V
221	DECIMAL POINT [1]	position (0=88888, 1=8888.8, 2=888.88, 3=88.888)
222	FILTER-P [1]	level, 0 to 9
223	FILTER-E [1]	level, 0 to 9
224	ROUND [1]	0=01, 1=02, 2=05, 3=10
225	TARE FUNCTION LOCK-OUT [1]	0=tare function enabled, 1=tare function disabled
226	CN2 FUNCTION.pin1 [1]	see manual
227	CN2 FUNCTION.pin2 [1]	see manual
228	CN2 FUNCTION.pin4 [1]	see manual
229	CN2 FUNCTION.pin5 [1]	see manual
230	LINEARIZATION POINTS [1]	2 to 12

ANNEXE 2 ALPHA-C

MEMORY LOCATIONS FOR MODEL ALPHA-C

READ ONLY DYNAMIC VARIABLES

TABLE 1a IN FLOATING POINT FORMAT

BYTE Nº	VARIABLE
240	PEAK
244	VALLEY
248	TARE
252	GROSS
256	NET
260	INPUT
264	DISPLAY

Variables in floating point format are transmitted without the decimal point of the display. When using a standard logic program it may be possible to manipulate these variables to read them in the same display format.

TABLE 1b IN INTEGER FORMAT

BYTE Nº	VARIABLE
270	DISPLAY
272	SETPOINT 1
274	SETPOINT 2
276	SETPOINT 3
278	SETPOINT 4

In models Alpha-C, **from version CJ**, it is possible to use function 03 to request the display and the setpoints values with sign in 'integer' format (2 bytes= 1 word). The addresses are specified in table 1b.

TABLE 2 READ ONLY DYNAMIC VARIABLES TYPE ON/OFF

BYTE Nº	VARIABLE	
266	bit 0 (LSB)	Setpoint1 status (0=OFF, 1=ON)
	bit 1	Setpoint2 status (0=OFF, 1=ON)
	bit 2	Setpoint3 status (0=OFF, 1=ON)
	bit 3	Setpoint4 status (0=OFF, 1=ON)
	bit 4	Remote input PIN1 status (0=OFF, 1=ON)
	bit 5	Remote input PIN2 status (0=OFF, 1=ON)
	bit 6	Remote input PIN4 status (0=OFF, 1=ON)
	bit 7 (MSB)	Remote input PIN5 status (0=OFF, 1=ON)

TABLE 3 CONTROL VARIABLES

COMMAND		FUNCTION
0n =	110	reset setpoints latch
0p =	112	reset peak
0r =	114	reset tare
0t =	116	tare
0v =	118	reset valley

TABLE 4 STATIC VARIABLES

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
0	INPUT POINT 1	digit 0
1	[6]	digit 1
2		digit 2
3		digit 3
4		digit 4
5		sign
6	INPUT POINT 2	digit 0
7	[6]	digit 1
8		digit 2
9		digit 3
10		digit 4
11		sign
12	INPUT POINT 3	digit 0
13	[6]	digit 1
14		digit 2
15		digit 3
16		digit 4
17		sign
18	INPUT POINT 4	digit 0
19	[6]	digit 1
20		digit 2
21		digit 3
22		digit 4
23		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
24	INPUT POINT 5	digit 0
25	[6]	digit 1
26		digit 2
27		digit 3
28		digit 4
29		sign
30	INPUT POINT 6	digit 0
31	[6]	digit 1
32		digit 2
33		digit 3
34		digit 4
35		sign
36	INPUT POINT 7	digit 0
37	[6]	digit 1
38		digit 2
39		digit 3
40		digit 4
41		sign
42	INPUT POINT 8	digit 0
43	[6]	digit 1
44		digit 2
45		digit 3
46		digit 4
47		sign
48	INPUT POINT 9	digit 0
49	[6]	digit 1
50		digit 2
51		digit 3
52		digit 4
53		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
54	INPUT POINT 10	digit 0
55	[6]	digit 1
56		digit 2
57		digit 3
58		digit 4
59		sign
60	INPUT POINT 11	digit 0
61	[6]	digit 1
62		digit 2
63		digit 3
64		digit 4
65		sign
66	INPUT POINT 12	digit 0
67	[6]	digit 1
68		digit 2
69		digit 3
70		digit 4
71		sign
72	DISPLAY POINT 1	digit 0
73	[6]	digit 1
74		digit 2
75		digit 3
76		digit 4
77		sign
78	DISPLAY POINT 2	digit 0
79	[6]	digit 1
80		digit 2
81		digit 3
82		digit 4
83		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
84	DISPLAY POINT 3	digit 0
85	[6]	digit 1
86		digit 2
87		digit 3
88		digit 4
89		sign
90	DISPLAY POINT 4	digit 0
91	[6]	digit 1
92		digit 2
93		digit 3
94		digit 4
95		sign
96	DISPLAY POINT 5	digit 0
97	[6]	digit 1
98		digit 2
99		digit 3
100		digit 4
101		sign
102	DISPLAY POINT 6	digit 0
103	[6]	digit 1
104		digit 2
105		digit 3
106		digit 4
107		sign
108	DISPLAY POINT 7	digit 0
109	[6]	digit 1
110		digit 2
111		digit 3
112		digit 4
113		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
114	DISPLAY POINT 8	digit 0
115	[6]	digit 1
116		digit 2
117		digit 3
118		digit 4
119		sign
120	DISPLAY POINT 9	digit 0
121	[6]	digit 1
122		digit 2
123		digit 3
124		digit 4
125		sign
126	DISPLAY POINT 10	digit 0
127	[6]	digit 1
128		digit 2
129		digit 3
130		digit 4
131		sign
132	DISPLAY POINT 11	digit 0
133	[6]	digit 1
134		digit 2
135		digit 3
136		digit 4
137		sign
138	DISPLAY POINT 12	digit 0
139	[6]	digit 1
140		digit 2
141		digit 3
142		digit 4
143		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
144	SETPOINT 1	digit 0
145	[6]	digit 1
146		digit 2
147		digit 3
148		digit 4
149		sign
150	SETPOINT 2	digit 0
151	[6]	digit 1
152		digit 2
153		digit 3
154		digit 4
155		sign
156	SETPOINT 3	digit 0
157	[6]	digit 1
158		digit 2
159		digit 3
160		digit 4
161		sign
162	SETPOINT 4	digit 0
163	[6]	digit 1
164		digit 2
165		digit 3
166		digit 4
167		sign
168	MODE SETPOINT 1	0=LED indication, 1=LED indication + display flash
169	[5]	0=net, 2=gross, 3=peak, 4=valley
170		0=dly, 1=hys_1, 2=hys_2
171		0=hi, 1=lo
172		0=off, 1=on, 2=latch

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
173	MODE SETPOINT 2	0=LED indication, 1=LED indication + display flash
174	[5]	0=net, 1=track set, 2=gross, 3=peak, 4=valley, 5=track auto
175		0=dly, 1=hys_1, 2=hys_2
176		0=hi, 1=lo
177		0=off, 1=on, 2=latch
178	MODE SETPOINT 3	0=LED indication, 1=LED indication + display flash
179	[5]	0=net, 2=gross, 3=peak, 4=valley
180		0=dly, 1=hys_1, 2=hys_2
181		0=hi, 1=lo
182		0=off, 1=on, 2=latch
183	MODE SETPOINT 4	0=LED indication, 1=LED indication + display flash
184	[5]	0=net, 1=track set, 2=gross, 3=peak, 4=valley
185		0=dly, 1=hys_1, 2=hys_2
186		0=hi, 1=lo
187		0=off, 1=on, 2=latch
188	DELAY / HYSTERESIS	digit 0
189	SETPOINT 1	digit 1
190	[4]	digit 2
191		digit 3
192	DELAY / HYSTERESIS	digit 0
193	SETPOINT 2	digit 1
194	[4]	digit 2
195		digit 3
196	DELAY / HYSTERESIS	digit 0
197	SETPOINT 3	digit 1
198	[4]	digit 2
199		digit 3
200	DELAY / HYSTERESIS	digit 0
201	SETPOINT 4	digit 1
202	[4]	digit 2
203		digit 3

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
204	DISPLAY VALUE FOR	digit 0
205	ANALOG OUTPUT LO	digit 1
206	[6]	digit 2
207		digit 3
208		digit 4
209		sign
210	DISPLAY VALUE FOR	digit 0
211	ANALOG OUTPUT HI	digit 1
212	[6]	digit 2
213		digit 3
214		digit 4
215		sign
216	ANALOG OUTPUT TYPE [1]	0=0-10V DC, 1=4-20mA DC
217	ANALOG OUTPUT FILTER [1]	0=filter off, 1=filter on
218	INPUT RANGE [1]	0=15mV, 1=30mV, 2=60mV, 3=300mV
219	DECIMAL POINT [1]	position (0=88888, 1=8888.8, 2=888.88, 3=88.888)
220	FILTER-P [1]	level, 0 to 9
221	FILTER-E [1]	level, 0 to 9
222	ROUND [1]	0=01, 1=02, 2=05, 3=10
223	TARE FUNCTION LOCK-OUT [1]	0=tare function enabled, 1=tare function disabled
224	CN2 FUNCTION.pin1 [1]	see manual
225	CN2 FUNCTION.pin2 [1]	see manual
226	CN2 FUNCTION.pin4 [1]	see manual
227	CN2 FUNCTION.pin5 [1]	see manual
228	LINEARIZATION POINTS [1]	2 to 12

ANNEXE 4 ALPHA-T

MEMORY LOCATIONS FOR MODEL ALPHA-T

READ ONLY DYNAMIC VARIABLES

TABLE 1a IN FLOATING POINT FORMAT

BYTE Nº	VARIABLE
96	PEAK
100	VALLEY
104	TEMPERATURE
108	DISPLAY
112	OFFSET

Variables in floating point format are transmitted without the decimal point of the display. When using a standard logic program it may be possible to manipulate these variables to read them in the same display format.

TABLE 1b IN INTEGER FORMAT

BYTE Nº	VARIABLE
118	DISPLAY
120	SETPOINT 1
122	SETPOINT 2
124	SETPOINT 3
126	SETPOINT 4

In models Alpha-T, **from versión T8**, it is possible to use function 03 to request the display and the setpoints values with sign in 'integer' format (2 bytes= 1 word). The addresses are specified in table 1b.

TABLE 2 READ ONLY DYNAMIC VARIABLES TYPE ON/OFF

BYTE Nº	VARIABLE	
116	bit 0 (LSB)	Setpoint1 status (0=OFF, 1=ON)
	bit 1	Setpoint2 status (0=OFF, 1=ON)
	bit 2	Setpoint3 status (0=OFF, 1=ON)
	bit 3	Setpoint4 status (0=OFF, 1=ON)
	bit 4	Remote input PIN1 status (0=OFF, 1=ON)
	bit 5	Remote input PIN2 status (0=OFF, 1=ON)
	bit 6	Remote input PIN4 status (0=OFF, 1=ON)
	bit 7 (MSB)	Remote input PIN5 status (0=OFF, 1=ON)

PROGRAMMING DATA - STATIC VARIABLES

TABLE 4

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
0	SETPOINT 1	digit 0
1	[6]	digit 1
2		digit 2
3		digit 3
4		digit 4
5		sign
6	SETPOINT 2	digit 0
7	[6]	digit 1
8		digit 2
9		digit 3
10		digit 4
11		sign
12	SETPOINT 3	digit 0
13	[6]	digit 1
14		digit 2
15		digit 3
16		digit 4
17		sign
18	SETPOINT 4	digit 0
19	[6]	digit 1
20		digit 2
21		digit 3
22		digit 4
23		sign
24	MODE SETPOINT 1	0=LED indication, 1=LED indication + display flash
25	[5]	0=net, 2=gross, 3=peak, 4=valley
26		0=dly, 1=hys_1, 2=hys_2
27		0=hi, 1=lo
28		0=off, 1=on, 2=latch

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
24	MODE SETPOINT 1	0=LED indication, 1=LED indication + display flash
25	[5]	0=net, 2=gross, 3=peak, 4=valley
26		0=dly, 1=hys_1, 2=hys_2
27		0=hi, 1=lo
28		0=off, 1=on, 2=latch
29	MODE SETPOINT 2	0=LED indication, 1=LED indication + display flash
30	[5]	0=net, 1=track set, 2=gross, 3=peak, 4=valley, 5=track auto
31		0=dly, 1=hys_1, 2=hys_2
32		0=hi, 1=lo
33		0=off, 1=on, 2=latch
34	MODE SETPOINT 3	0=LED indication, 1=LED indication + display flash
35	[5]	0=net, 2=gross, 3=peak, 4=valley
36		0=dly, 1=hys_1, 2=hys_2
37		0=hi, 1=lo
38		0=off, 1=on, 2=latch
39	MODE SETPOINT 4	0=LED indication, 1=LED indication + display flash
40	[5]	0=net, 1=track set, 2=gross, 3=peak, 4=valley
41		0=dly, 1=hys_1, 2=hys_2
42		0=hi, 1=lo
43		0=off, 1=on, 2=latch
44	DELAY / HYSTERESIS	digit 0
45	SETPOINT 1	digit 1
46	[4]	digit 2
47		digit 3
48	DELAY / HYSTERESIS	digit 0
49	SETPOINT 2	digit 1
50	[4]	digit 2
51		digit 3
52	DELAY / HYSTERESIS	digit 0
53	SETPOINT 3	digit 1
54	[4]	digit 2
55		digit 3

BYTE Nº	VARIABLE [nº de bytes]	BYTE DATA
56	DELAY / HYSTERESIS	digit 0
57	SETPOINT 4	digit 1
58	[4]	digit 2
59		digit 3
60	DISPLAY VALUE FOR	digit 0
61	ANALOG OUTPUT LO	digit 1
62	[6]	digit 2
63		digit 3
64		digit 4
65		sign
66	DISPLAY VALUE FOR	digit 0
67	ANALOG OUTPUT HI	digit 1
68	[6]	digit 2
69		digit 3
70		digit 4
71		sign
72	ANALOG OUTPUT TYPE [1]	0=0-10V DC, 1=4-20mA DC
73	ANALOG OUTPUT FILTER [1]	0=filter off, 1=filter on
74	INPUT TYPE [1]	0=Pt100, 1=TCJ, 2=TCK, 3=TCT, 4=TCR, 5=TCS, 6=TCE
75	SCALE/RESOLUTION [1]	0=1°C, 1=0.1°C, 2=1°F, 3=0.1°F
76	TEMPERATURE OFFSET	digit 0
77	[3]	digit 1
78		sign
79	FILTER-P [1]	level, 0 to 9
80	CN2 FUNCION.pin1 [1]	see manual
81	CN2 FUNCION.pin2 [1]	see manual
82	CN2 FUNCION.pin4 [1]	see manual
83	CN2 FUNCION.pin5 [1]	see manual

ANNEXE 4 ALPHA-D

MEMORY LOCATIONS FOR MODEL ALPHA-D

TABLE 1 READ ONLY DYNAMIC VARIABLES IN FLOATING POINT FORMAT

BYTE Nº	VARIABLE
166	NºCOUNTS (batch counter)
170	NºBATCH (batch counter)
174	DISPLAY
178	PEAK
182	VALLEY

Variables in floating point format are transmitted with the decimal point in the position shown on the display except the variable DISPLAY in chronometer mode.

TABLA 2 READ ONLY DYNAMIC VARIABLES TYPE ON/OFF

BYTE Nº	VARIABLE	
186	bit 0 (LSB)	Setpoint1 status (0=OFF, 1=ON)
	bit 1	Setpoint2 status (0=OFF, 1=ON)
	bit 2	Setpoint3 status (0=OFF, 1=ON)
	bit 3	Setpoint4 status (0=OFF, 1=ON)

TABLA 3 CONTROL VARIABLES

COMMAND		FUNCTION
0p =	112	reset peak
0r =	114	reset offset
0t =	116	offset
0v =	118	reset valley
0x =	120	reset batch count
0z =	122	reset process count

PROGRAMMING DATA - STATIC VARIABLES

TABLE 4

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
0	SETPOINT 1	digit 0
1	[6]	digit 1
2		digit 2
3		digit 3
4		digit 4
5		sign
6	SETPOINT 2	digit 0
7	[6]	digit 1
8		digit 2
9		digit 3
10		digit 4
11		sign
12	SETPOINT 3	digit 0
13	[6]	digit 1
14		digit 2
15		digit 3
16		digit 4
17		sign
18	SETPOINT 4	digit 0
19	[6]	digit 1
20		digit 2
21		digit 3
22		digit 4
23		sign
24	MODE SETPOINT 1 [1]	0=off, 1=pulse, 2=latch, (3=btch-1, 4=btch-2, for batch counter)
25	MODE SETPOINT 2 [1]	0=off, 1=pulse, 2=latch
26	MODE SETPOINT 3 [1]	0=off, 1=pulse, 2=latch
27	MODE SETPOINT 4 [1]	0=off, 1=pulse, 2=latch

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
28	CONTROL MODE [1]	1=independent, 2=stop on set2, 3=reset on set2, 4=deactivation on set2
29	MODE SETPOINT 1	0
30	[5]	0
31		0=dly, 1=hys_1, 2=hys_2
32		0=hi, 1=lo
33		0=off, 1=on
34	MODE SETPOINT 2	0
35	[5]	0=set, 1=track
36		0=dly, 1=hys_1, 2=hys_2
37		0=hi, 1=lo
38		0=off, 1=on
39	MODE SETPOINT 3	0
40	[5]	0
41		0=dly, 1=hys_1, 2=hys_2
42		0=hi, 1=lo
43		0=off, 1=on
44	MODE SETPOINT 4	0
45	[5]	0=set, 1=track
46		0=dly, 1=hys_1, 2=hys_2
47		0=hi, 1=lo
48		0=off, 1=on
49	DELAY / HYSTERESIS (FREQ&TACH)	digit 0
50	or TIME (COUNT&CHRONO) FOR	digit 1
51	OUTPUT SETPOINT 1	digit 2
52	[5]	digit 3 (=0 for COUNT&CHRONO)
53		0
54	DELAY / HYSTERESIS (FREQ&TACH)	digit 0
55	or TIME (COUNT&CHRONO) FOR	digit 1
56	OUTPUT SETPOINT 2	digit 2
57	[5]	digit 3 (=0 for COUNT&CHRONO)
58		0

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
59	DELAY / HYSTERESIS (FREQ&TACH)	digit 0
60	or TIME (COUNT&CHRONO) FOR	digit 1
61	OUTPUT SETPOINT 3	digit 2
62	[5]	digit 3 (=0 for COUNT&CHRONO)
63		0
64	DELAY / HYSTERESIS (FREQ&TACH)	digit 0
65	or TIME (COUNT&CHRONO) FOR	digit 1
66	OUTPUT SETPOINT 3	digit 2
67	[5]	digit 3 (=0 for COUNT&CHRONO)
68		0
69	DISPLAY VALUE FOR	digit 0
70	ANALOG OUTPUT LO	digit 1
71	[6]	digit 2
72		digit 3
73		digit 4
74		sign
75	DISPLAY VALUE FOR	digit 0
76	ANALOG OUTPUT HI	digit 1
77	[6]	digit 2
78		digit 3
79		digit 4
80		sign
81	ANALOG OUTPUT TYPE [1]	0=0-10V DC, 1=4-20mA DC
82	TIMING DIRECTION [1]	0=up, 1=down
83	RATE MODE [1]	0=direct (dIrEC), 1=reversed (InvEr)
84	DECIMAL POINT [1]	position (0=88888, 1=8888.8, 2=888.88, 3=88.888, 4=8.8888)
85	COUNTER OFFSET	digit 0
86	[6]	digit 1
87		digit 2
88		digit 3
89		digit 4
90		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
91	Nº OF COUNTS PER BATCH	digit 0
92	[5]	digit 1
93		digit 2
94		digit 3
95		0
96	CHRONOMETER OFFSET	digit 0
97	[5]	digit 1
98		digit 2
99		digit 3
100		digit 4
101	MULTIPLIER FACTOR	digit 0
102	[5]	digit 1
103		digit 2
104		digit 3
105		0
106	PULSES PER CYCLE	digit 0
107	[5]	digit 1
108		digit 2
109		digit 3
110		0
111	AVERAGE TIME	digit 0
112	[5]	digit 1
113		digit 2
114		0
115		0
116	DESIRED DISPLAY	digit 0
117	(RATE MODE TACHOMETER)	digit 1
118	[5]	digit 2
119		digit 3
120		digit 4

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
121	INPUT FREQUENCY (RATE)	digit 0
122	or	digit 1
123	PULSES/REVOLUTION (RPM)	digit 2
124	[5]	digit 3
125		digit 4 (=0 for RPM)
126	LIMIT TIME	digit 0
127	[5]	digit 1
128		0
129		0
130		0
131	DISPLAY UPDATE TIME	digit 0
132	[5]	digit 1
133		0
134		0
135		0
136	INPUT TYPE [1]	0=counter, 1=chronometer, 2=frequency meter, 3=tachometer
137	COUNTER MODE [1]	0=batch, 1=up, 2=down, 3=up/down
138	CHRONOMETER MODE [1]	0=A↑ start A↓ stop (In-A), 1=A↑ start B↑ stop (In-Ab), 2=A↑ start A↑ stop (In-AA)
139	CHRONOMETER UNITS [1]	0=9999.9h (H.H), 1=999h59min (H.MM), 2=999min59s (M.SS), 3=9999.9s (0.1-S), 4=999.99s (0.01-S)
140	COUNT INHIBIT INPUT [1]	0=no (In-A), 1=input B (In-Ab)
141	FACTOR DECIMAL POINT [1]	position (0=88888, 1=8888.8, 2=888.88, 3=88.888)
142	TACHOMETER MODE [1]	0=rpm, 1=rate
143	UP/DOWN COUNTER MODE [1]	0=independent, 1=directionnal, 2=phase
144	DE-BOUNCE FILTER [1]	0=no, 1=yes
145	FREQUENCY DECIMAL POINT [1]	position (0=88888, 1=8888.8, 2=888.88)
146	LOCK-OUT/MODE RESET KEY (PROCESS) [1]	0=reset key disabled, 1=momentary reset, 2=maintained reset
147	LOCK-OUT/MODE RESET KEY (BATCH) [1]	0=reset key disabled, 1=momentary reset, 2=maintained reset
148	LOCK-OUT OFFSET KEY [1]	0=offset key disabled, 1=offset key enabled

ANNEXE 5 BETA-M

MEMORY LOCATIONS FOR MODEL BETA-M

TABLE 1 READ ONLY DYNAMIC VARIABLES IN FLOATING POINT FORMAT

BYTE Nº	VARIABLE
162	PEAK
166	VALLEY
170	TARE
174	GROSS
178	NET
182	INPUT
186	DISPLAY
190	GROSS without FILTERS
194	TOTAL
198	Nº BATCH

Variables in floating point format are transmitted without the decimal point of the display. When using a standard logic program it may be possible to manipulate these variables to read them in the same format as the display.

TABLE 2 READ ONLY DYNAMIC VARIABLES TYPE ON/OFF

BYTE Nº	VARIABLE	
202	bit 0 (LSB)	Setpoint1 status (0=OFF, 1=ON)
	bit 1	Setpoint2 status (0=OFF, 1=ON)
	bit 2	Setpoint3 status (0=OFF, 1=ON)
	bit 3	Setpoint4 status (0=OFF, 1=ON)
	bit 4	Remote input PIN1 status (0=OFF, 1=ON)
	bit 5	Remote input PIN2 status (0=OFF, 1=ON)
	bit 6	Remote input PIN4 status (0=OFF, 1=ON)
	bit 7 (MSB)	Remote input PIN5 status (0=OFF, 1=ON)

TABLE 3 CONTROL VARIABLES

COMMAND		FUNCTION
0n =	110	reset setpoints latch
0p =	112	reset peak
0r =	114	reset tare
0t =	116	tare
0v =	118	reset valley
0x =	122	reset total and nº of batches

PROGRAMMING DATA - STATIC VARIABLES

TABLE 4

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
0	INPUT POINT 1	digit 0
1	[6]	digit 1
2		digit 2
3		digit 3
4		digit 4
5		sign
6	INPUT POINT 2	digit 0
7	[6]	digit 1
8		digit 2
9		digit 3
10		digit 4
11		sign
12	DISPLAY POINT 1	digit 0
13	[6]	digit 1
14		digit 2
15		digit 3
16		digit 4
17		sign
18	DISPLAY POINT 2	digit 0
19	[6]	digit 1
20		digit 2
21		digit 3
22		digit 4
23		sign

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
24	SETPOINT 1	digit 0
25	[6]	digit 1
26		digit 2
27		digit 3
28		digit 4
29		sign
30	SETPOINT 2	digit 0
31	[6]	digit 1
32		digit 2
33		digit 3
34		digit 4
35		sign
36	SETPOINT 3	digit 0
37	[6]	digit 1
38		digit 2
39		digit 3
40		digit 4
41		sign
42	SETPOINT 4	digit 0
43	[6]	digit 1
44		digit 2
45		digit 3
46		digit 4
47		sign
48	DELAY/HYSTERESIS	digit 0
49	SETPOINT 1	digit 1
50	[5]	digit 2
51		digit 3
52		digit 4

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
53	DELAY/HYSTERESIS	digit 0
54	FOR OUTPUT SETPOINT 2	digit 1
55	[5]	digit 2
56		digit 3
57		digit 4
58	DELAY/HYSTERESIS	digit 0
59	FOR OUTPUT SETPOINT 3	digit 1
60	[5]	digit 2
61		digit 3
62		digit 4
63	DELAY/HYSTERESIS	digit 0
64	FOR OUTPUT SETPOINT 4	digit 1
65	[5]	digit 2
66		digit 3
67		digit 4
68	SET SETPOINT 1 [1]	0=off, 1=on
69	SET SETPOINT 2 [1]	0=off, 1=on, 2= track
70	SET SETPOINT 3 [1]	0=off, 1=on
71	SET SETPOINT 4 [1]	0=off, 1=on, 2=track
72	COMP SETPOINT 1 [1]	0=net, 1=gross, 1=peak, 2=valley
73	COMP SETPOINT 2 [1]	0=net, 1=gross, 1=peak, 2=valley
74	COMP SETPOINT 3 [1]	0=net, 1=gross, 1=peak, 2=valley
75	COMP SETPOINT 4 [1]	0=net, 1=gross, 1=peak, 2=valley
76	MODE SETPOINT 1 [1]	0=hi, 1=lo
77	MODE SETPOINT 2 [1]	0=hi, 1=lo
78	MODE SETPOINT 3 [1]	0=hi, 1=lo
79	MODE SETPOINT 4 [1]	0=hi, 1=lo
80	DEL/HYS SETPOINT 1 [1]	0=time delay, 1=hysteresis-1, 2=hysteresis-2
81	DEL/HYS SETPOINT 2 [1]	0=time delay, 1=hysteresis-1, 2=hysteresis-2
82	DEL/HYS SETPOINT 3 [1]	0=time delay, 1=hysteresis-1, 2=hysteresis-2
83	DEL/HYS SETPOINT 4 [1]	0=time delay, 1=hysteresis-1, 2=hysteresis-2

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
84	LATCH SETPOINT 1 [1]	0=no, 1=yes
85	LATCH SETPOINT 2 [1]	0=no, 1=yes
86	LATCH SETPOINT 3 [1]	0=no, 1=yes
87	LATCH SETPOINT 4 [1]	0=no, 1=yes
88	BLINK SETPOINT 1 [1]	0=no, 1=yes
89	BLINK SETPOINT 2 [1]	0=no, 1=yes
90	BLINK SETPOINT 3 [1]	0=no, 1=yes
91	BLINK SETPOINT 4 [1]	0=no, 1=yes
92	TRACK AUTO [1]	0=no, 1=yes
93	DISPLAY VALUE FOR	digit 0
94	ANALOG OUPUT HI	digit 1
95	[6]	digit 2
96		digit 3
97		digit 4
98		sign
99	DISPLAY VALUE FOR	digit 0
100	ANALOG OUPUT LO	digit 1
101	[6]	digit 2
102		digit 3
103		digit 4
104		sign
105	ANALOG OUTPUT TYPE [1]	0=0-10V DC, 1=4-20mA DC
106	ANALOG OUTPUT FILTER [1]	0=filter OFF, 1=filter ON
107	EXCITATION [1]	0=24V, 1=10V
108	INPUT [1]	0=process, 1=load cell, 2=Pt100, 3=thermocouple, 4=potentiometer
109	PROCESS INPUT TYPE [1]	0=volts, 1=milliamperes
110	THERMOCOUPLE INPUT TYPE [1]	0=TCJ, 1=TCK, 2=TCT, 3=TCR, 4=TCS, 5=TCE
111	PROCESS INPUT RANGE [1]	0=1V/1mA, 1=10V/20mA
112	LOAD CELL INPUT RANGE [1]	0=0=300mV, 1=60mV, 2=30mV, 3=15mV
113	TEMPERATURE UNITS [1]	0=Celsius. 1=Fahrenheit
114	RESOLUTIÓN THERMOMETERS [1]	0=0.1°, 1=1°

BYTE Nº	VARIABLE [nº of bytes]	BYTE DATA
115	TEMPERATURE OFFSET	digit 0
116	[3]	digit 1
117		sign
118	DECIMAL POINT [1]	position (0=±88888, 1=±8888.8, 2=±888.88, 3=±88.888, 4=±8.8888)
119	FILTER-P [1]	level, 0 to 9
120	FILTER-E [1]	level, 0 to 9
121	NUMBER OF READINGS	digit 0
122	TO AVERAGE (1 TO 200)	digit 1
123	[3]	digit 2
124	DISPLAY BRIGHTNESS [1]	0=hi, 1=lo
125	LEFT ZEROS [1]	0=no, 1=yes
126	DISPLAY UPDATE RATE [1]	0=16/s, 1=4/s, 2=1/s
127	ROUND [1]	0=001, 1=002, 2=005, 3=010, 4=020, 5=050, 6=100
128	LOCK-OUT TARE KEY [1]	0=unlocked, 1= locked
129	CN2 FUNCTION.pin 1 [1]	see manual
130	CN2 FUNCTION.pin 2 [1]	see manual
131	CN2 FUNCTION.pin 4 [1]	see manual
132	CN2 FUNCTION.pin 5 [1]	see manual
133	LOCK SET1 [1]	0=unlocked, 1= locked
134	LOCK SET2 [1]	0=unlocked, 1= locked
135	LOCK SET3 [1]	0=unlocked, 1= locked
136	LOCK SET4 [1]	0=unlocked, 1= locked
137	LOCK INPUT [1]	0=unlocked, 1= locked
138	LOCK SCAL [1]	0=unlocked, 1= locked
139	LOCK FILTERS [1]	0=unlocked, 1= locked
140	LOCK TARE KEY LOCK [1]	0=unlocked, 1= locked
141	LOCK ANALOG OUTPUT [1]	0=unlocked, 1= locked
142	LOCK RS OUTPUT [1]	0=unlocked, 1= locked
143	LOCK LOGIC FUNCTIONS [1]	0=unlocked, 1= locked
144	TOTAL LOCK [1]	0=unlocked, 1= locked