MODBUS for *E*CH 400
Serial Communication Protocol

1 CONTENTS

2 HOW TO USE THIS MANUAL

This manual is designed to permit quick, easy reference with the following features:

References

References column**:**
A column to the left of the text contains *references* to subjects discussed in the text to help you locate the information you need quickly and easily.

Cross references

Cross references:
All words written in *italics* are referenced in the subject index to help you find the page containing details on this subject; supposing you read the following text:
”when the alarm is triggered, the compressors will be shut down”
The italics mean that you will find a reference to the page on the topic of compressors listed under the item compressors in the index.
If you are consulting the manual “on-line” (using a computer), words which appear in italics are hyperlinks: just click on a word in italics with the mouse to go directly to the part of the manual that discusses this topic.

Icons for emphasis

Some segments of text are marked by icons appearing in the *references* column with the meanings specified below:

☞ **Take note**: information on the topic under discussion which the user ought to keep in mind

💡 **Tip**: a recommendation which may help the user to understand and make use of the information supplied on the topic under discussion.

⚠ **Warning**! : **information which is essential for preventing negative consequences for the system or a hazard to personnel, instruments, data, etc., and which users MUST read with care.**

3 IMPLEMENTED FUNCTIONS AND AREAS

This section describes the Energy 400 communication and interface modes, using Modbus protocol.
Knowledge of specific protocol principles is required in order to be able to read and understand this section, therefore the reader is advised to refer to the "Modbus Protocol" general manual.

3.1 Logic areas

Energy 400 provides 10 *logic areas* where Modbus *functions* can be used:

Area no.	Area
1	*PARAMETERS*
2	*PROBES*
4	*DIGITAL INPUTS*
5	*DIGITAL OUTPUTS*
6	*EEPROM*
7	*ANALOGUE OUTPUTS*
8	*RAM*
9	*VER*
17	Visibility of *parameters* and submenus

3.1.1 Parameters

Area Index: 1
In this area the Energy 400 characteristic *parameters* can be read and written, using Modbus *functions* 03 (hex) and 10 (hex).

The first parameter (index 0 in the Energy 400 manual and Param manager) has a modbus index of 1; the various *parameters* will be addressed in the following manner (refer to the "data field" in the Modbus general manual):

Index	Parameter	Address
0	Pa G01	0000 1000 0000 0001 (08 01 hex)
1	Pa G02	0000 1000 0000 0010 (08 02 hex)
2	Pa H01	0000 1000 0000 0011 (08 03 hex)
.....	.....	.....
70	Pa H69	0000 1000 0100 0111 (08 47 hex)
71	Pa A01	0000 1000 0100 1000 (08 48 hex)
....	....	....
96	Pa A26	0000 1000 0110 0001 (08 61 hex)
97	Pa C01	0000 1000 0110 0010 (08 62 hex)
....	....	....
104	Pa C08	0000 1000 0110 1001 (08 69 hex)
105	Pa F01	0000 1000 0110 1010 (08 70 hex)
....	....	....
129	Pa F25	0000 1000 1000 0010 (08 82 hex)
130	Pa P01	0000 1000 1000 0011 (08 83 hex)
....	....	....
132	Pa P03	0000 1000 1000 0101 (08 85 hex)
133	Pa R01	0000 1000 1000 0110 (08 86 hex)
....	....	....
149	Pa R17	0000 1000 1001 0110 (08 96 hex)
150	Pa d01	0000 1000 1001 0111 (08 97 hex)
.....	.....	....
160	Pa d11	0000 1000 1010 0001 (08 A1 hex)
161	Pa N01	0000 1000 1010 0010 (08 A2 hex)
....	....	....
167	Pa N07	0000 1000 1010 1000 (08 A8 hex)
168	PA L01	0000 1000 1010 1001 (08 A9 hex)
….	….	....
175	PA L08	0000 1000 1011 0000 (08 B0 hex)
176	*LABEL* SeT	0000 1000 1011 0001 (08 B1 hex)
177	*LABEL* TP	(08 B2 hex)
178	*LABEL* ERR	(08 B3 hex)
179	*LABEL* ID	(08 B4 hex)
180	*LABEL* PAr	(08 B5 hex)
181	*LABEL* PSS	(08 B6 hex)
182	*LABEL*	(08 B7 hex)

	OHr	
183	*LABEL* COO	(08 B8 hex)
184	*LABEL* HEA	(08B9 hex)
185	*LABEL* CnF	(08BA hex)
186	*LABEL* CP	(08BB hex)
187	*LABEL* FAN	(08BC hex)
188	*LABEL* ALL	(08BD hex)
189	*LABEL* PUP	(08BE hex)
190	*LABEL* Fro	(08BF hex)
191	*LABEL* dFr	(08C0 hex)
192	*LABEL* ESP	(08C1 hex)
193	*LABEL* SPL	(08C2 hex)
194	*LABEL* OH1	(08C3 hex)
195	*LABEL* OH2	(08C4 hex)
196	*LABEL* OH3	(08C5 hex)
197	*LABEL* OH4	(08C6 hex)
198	*LABEL* OHP	(08C7 hex)

3.1.2 Probes

Area Index: 2
In this area the Energy 400 probe values can be read, using Modbus function 04 (hex).
The *probes* are addressed in the following manner (refer to the "data field" in the Modbus general manual):

Index	Probe	Address
1	ST1	0001 0000 0000 0001 (10 01 hex)
2	ST2	0001 0000 0000 0010 (10 02 hex)
3	ST3	0001 0000 0000 0011 (10 03 hex)
4	ST4	0001 0000 0000 0100 (10 04 hex)
5	ST5	0001 0000 0000 0101 (10 05 hex)
6	ST6 (if configured in digital mode)	0001 0000 0000 0110 (10 06 hex)

3.1.3 Digital inputs

Area Index: 4
In this area the Energy 400 digital input values can be read, using Modbus function 02 (hex).
The inputs are addressed in the following manner (refer to the "data field" in the Modbus general manual):

Index	Digital input	Address
1	ID1	0010 0000 0000 0001 (20 01 hex)
2	ID2	0010 0000 0000 0010 (20 02 hex)
....	....	....
15	ID15	0010 0000 0000 1111 (20 0F hex)
16	IDST4	0001 0000 0001 0000 (10 10 hex)

3.1.4 Digital outputs

Area Index: 5

In this area the Energy 400 digital input values can be read and written, using Modbus *functions* 01 (hex) and 0F (hex).
The outputs are addressed in the following manner (refer to the "data field" in the Modbus general manual):

Index	Digital input	Address
1	RL1	0010 1000 0000 0001 (28 01 hex)
2	RL2	0010 1000 0000 0010 (28 02 hex)
….	….	….
10	RL10	0010 1000 0000 1010 (28 0A hex)

3.1.5 Eeprom

Area Index: 6
The *eeprom* memory values can be read and written (organised by words), using Modbus *functions* 03 (hex) and 10 (hex).
The first word has an index of 1 and the memory is 256 words long, therefore the memory will be addressed in the following manner:

Index	Word	Address
1	1	0011 0000 0000 0001 (30 01 hex)
2	2	0011 0000 0000 0010 (30 02 hex)
….	….	….
256	255	0011 0000 1111 1111 (30 FF hex)

Followed by certain indexes of interest:

Eeprom indexes

Addr E2	Variable Name	Description	Size Bit	Location
0[1]	state_e2	machine state	8	8-15
1	hour_funct.[0]	hour funct. Comp. 1	16	0-15
2	hour_funct.[1]	hour funct. Comp. 2	16	0-15
3	hour_funct.[2]	hour funct. Comp. 3	16	0-15
4	hour_funct.[3]	hour funct. Comp. 4	16	0-15
5	hour_funct.[4]	hour funct. Pump	16	0-15
6	gain_in4_20[0]	ST3 current calibration gain	16	0-15
7	gain_in4_20[1]	ST6 current calibration gain	16	0-15
8	gain_in4_20[2]	Not Used	16	0-15
9	gain_in4_20[3]	Not Used	16	0-15
10	off_in4_20[0]	ST3 current calibration offset	16	0-15
11	off_in4_20[1]	ST6 current calibration offset	16	0-15
12	off_in4_20[2]	Not Used	16	0-15
13	off_in4_20[3]	Not Used	16	0-15
14	gain_out4_20[0]	gain output 4-20 mA1	8	8-15
14*	gain_out4_20[1]	gain output 4-20 mA2	8	0-7
15	gain_out4_20[2]	Not Used	8	8-15
15*	off_out4_20[0]	offset output 4-20 mA1	8	0-7
16	off_out4_20[1]	offset output 4-20 mA2	8	8-15
16*	off_out4_20[2]	Not Used	8	0-7

[1]: The byte describing the state of the device is organised as follows:
Bit 8 –9:
- 00 = standby
- 01 = cooling
- 10 = heating

Bit 10 – 14:
- Not used

Bit 15:
- 1 = ON
- 0 = OFF

*It should be remembered that the *Eeprom* memory is organised by words: in certain cases the data is divided between Lo and Hi bytes within the same word

The *Eeprom* writing of the variables does not involve the writing in *ram* memory of the same ones. Updating happens if the instrument is turn off and then turn on. To ensure the immediate modification you must write in the copy in *ram* memory the saved variable in *Eeprom*,. Have a look to the relative paragraph regarding writing in *ram* memory
Followed by certain indexes of interest:

Eeprom indexes

Addr E2	Variable Name	Description	Size Bit	Location

0[1]	state_e2	machine state	8	8-15
1	hour_funct.[0]	hour funct. Comp. 1	16	0-15
2	hour_funct.[1]	hour funct. Comp. 2	16	0-15
3	hour_funct.[2]	hour funct. Comp. 3	16	0-15
4	hour_funct.[3]	hour funct. Comp. 4	16	0-15
5	hour_funct.[4]	hour funct. Pump	16	0-15

3.1.6 Analogue outputs

Area Index: 7
In this area the Energy 400 analogue output values can be read and written, using Modbus *functions* 03 (hex) and 100 (hex).
The outputs are addressed in the following manner (refer to the "data field" in the Modbus general manual):

Index	Digital input	Address
1	% fan circuit 1	0010 1000 0000 0001 (28 01 hex)
2	% fan circuit 1	0010 1000 0000 0010 (28 02 hex)
3	% freecooling output	0010 1000 0000 0011 (28 03 hex)

3.1.7 Ram

Area Index: 8
In this area the *ram* memory values (organised by words) can be read and written, using Modbus *functions* 03 (hex) and 10 (hex).
The *ram* memory starts at the FB7F physical address, therefore the index of a word is obtained with the following equation:

index = relative address – FB7F (hex)

the memory is 65407 words long (FF7F hex).

The following tables show the relative addresses of the *Ram* memory principal areas of interest:

3.1.8 Eeprom variaibles copy

Eeprom indexes

Addr E2	Variable Name	Description	Size Bit	Location
FBEE	state_e2	machine state	8	8-15
FBF0	hour_funct.[0]	hour funct. Comp. 1	16	0-15
FBF1	hour_funct.[1]	hour funct. Comp. 2	16	0-15
FBF2	hour_funct.[2]	hour funct. Comp. 3	16	0-15
FBF3	hour_funct.[3]	hour funct. Comp. 4	16	0-15
FBF4	hour_funct.[4]	hour funct. Pump	16	0-15

3.1.9 Device state and keyboard

Test Bit

Add. : **0xFF0F**		**States**	
n_bit	***Label***	***meaning***	***active at***
Bit_0	COLL	Puts the machine in test mode and stops all regulators	1
Bit_1			
Bit_2			
Bit_3			
Bit_4			
Bit_5			
Bit_6			
Bit_7			

To modify the output state (digital or analogue) and the state of the display, locate this bit at 1; in this event all regulators are disabled.

Wall keyboard: units led

Add. : **0xFBA4**	**Wall keyboard unit Digit 1 (units)**

Wall keyboard: tenths led

Add. : **0xFBA5**	**Wall keyboard unit Digit 2 (tenths)**

Wall keyboard: hundredths led

Add. : **0xFBA6**	**Wall keyboard unit Digit 3 (hundredths)**

Led

Add. : **0xFBA7**	**LED**

Keyboard 32 x 74

Add. : **0xFBA8**	**Keyboard unit 32x74**

3.1.10 Digital inputs

Lo byte

Add. : **0xFBD8**		***Digital inputs*** **(Lo byte)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0	ID11	Digital input 6 on base	1
Bit_1	ID13	Digital input 13 on expansion	1
Bit_2	IDST4	Digital input on analogue input ST4	1
Bit_3	ID3	Digital input 3 on base	1
Bit_4	ID2	Digital input 2 on base	1
Bit_5	ID1	Digital input 1 on base	1
Bit_6	ID14	Digital input 14 on expansion	1
Bit_7	ID15	Digital input 15 on expansion	1

Hi byte

Add. : **0xFBD9**		***Digital inputs*** **(Hi byte)**	
n_bit	***Label***	***meaning***	***active at***
Bit_0	ID6	Digital input 6 on base	1
Bit_1	ID4	Digital input 4 on base	1
Bit_2	ID5	Digital input 5 on base	1
Bit_3	ID7	Digital input 7 on base	1
Bit_4	ID12	Digital input 12 on expansion	1
Bit_5	ID10	Digital input 10 on base	1
Bit_6	ID8	Digital input 8 on base	1
Bit_7	ID9	Digital input 6 on base	1

☞ A copy of the input values is generated in the *RAM* memory; in this event, reading the *RAM* memory or directly reading the *digital inputs* (logic area 4) is the same thing.

3.1.11 Digital outputs

Lo byte

Add. : **0xFCE8**		***Digital outputs*** **(low byte)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0	RL8	Output relay 8	1
Bit_1	RL1	Output relay 1	1
Bit_2	RL2	Output relay 2	1
Bit_3	RL3	Output relay 3	1
Bit_4	RL4	Output relay 4	1
Bit_5	RL5	Output relay 5	1
Bit_6	RL6	Output relay 6	1
Bit_7	RL7	Output relay 7	1

Hi byte

Add. : **0xFCE9**		***Digital outputs*** **(Hi byte)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0			1
Bit_1			1
Bit_2			1
Bit_3			1
Bit_4			1
Bit_5			1
Bit_6	RL10	Output relay 10 on expansion	1
Bit_7	RL9	Output relay 9 on expansion	1

☞ A copy of the output values is generated in the *RAM* memory; in this event, reading the *RAM* memory or directly reading the *digital outputs* (logic area 5) is the same thing

3.1.12 Alarms and resource blocks

AUTOMATICALLY RESET ALARMS

Auto reset: Lo byte – Word 1

Add. : **0xFD86**		**Automatically reset alarms (low byte – Word 1)**	
n_bit	***Label***	***Meaning***	***active at***

Bit_0		ON-OFF remote	1
Bit_1		Maximum pressure circuit 1	1
Bit_2		Low pressure circuit 1	1
Bit_3		Thermal switch compressor 1	1
Bit_4		Fan thermal switch 1	1
Bit_5		Antifreeze circuit 1	1
Bit_6		Error probe ST2	1
Bit_7		Error probe ST3	1

Auto reset.:
Hi byte – Word 1

Add. :	**0xFD87**	**Automatically reset alarms (Hi byte – Word 1)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0		Maximum pressure compressor 1	1
Bit_1		Maximum pressure circuit 1 analogue input	1
Bit_2		Low pressure circuit 1 analogue input	1
Bit_3		Thermal switch compressor 2	1
Bit_4		Maximum pressure compressor 2	1
Bit_5		Maximum pressure circuit 2	1
Bit_6		Low pressure circuit 2	1
Bit_7		Thermal switch compressor 3	1

Auto reset:
Lo byte – Word 2

Add. :	**0xFD88**	**Automatically reset alarms (low byte – Word 1)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0		Thermal switch fan 2	1
Bit_1		Antifreeze circuit 2	1
Bit_2		Error probe ST5	1
Bit_3		Error probe ST6	1
Bit_4		Maximum pressure compressor 3	1
Bit_5		Maximum pressure circuit 2 analogue input	1
Bit_6		Low pressure circuit 2 analogue input	1
Bit_7		Thermal switch compressor 4	1

Auto reset:
Hi byte – Word 2

Add. :	**0xFD89**	**Automatically reset alarms (Hi byte – Word 1)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0		Maximum pressure compressor 4	1
Bit_1		Error probe ST1	1
Bit_2		Flow meter alarm	1
Bit_3		Error probe ST4	1
Bit_4		Antifreeze external circuits	1
Bit_5		Machine discharged	1
Bit_6		Configuration error	1
Bit_7		Control temperature too high error	1

MANUALLY RESET ALARMS

Manual reset:
Lo byte – Word 1

Add. :	**0xFD8A**	**Manually reset alarms (low byte – Word 1)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0		ON-OFF remote	1
Bit_1		Maximum pressure circuit 1	1
Bit_2		Low pressure circuit 1	1
Bit_3		Thermal switch compressor 1	1
Bit_4		Thermal switch fan 1	1
Bit_5		Antifreeze circuit 1	1
Bit_6		Error probe ST2	1
Bit_7		Error probe ST3	1

Manual reset:
Hi byte – Word 1

Add. :	**0xFD8B**	**Manually reset alarms (Hi byte – Word 1)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0		Maximum pressure compressor 1	1
Bit_1		Maximum pressure circuit 1 analogue input	1
Bit_2		Low pressure circuit 1 analogue input	1
Bit_3		Thermal switch compressor 2	1
Bit_4		Maximum pressure compressor 2	1
Bit_5		Maximum pressure circuit 2	1
Bit_6		Low pressure circuit 2	1
Bit_7		Thermal switch compressor 3	1

Manual reset:
Lo byte – Word 2

Add. :	**0xFD8c**	**Manually reset alarms (low byte – Word 1)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0		Thermal switch fan 2	1
Bit_1		Antifreeze circuit 2	1
Bit_2		Error probe ST5	1

Bit_3		Error probe ST6	1
Bit_4		Maximum pressure compressor 3	1
Bit_5		Maximum pressure circuit 2 analogue input	1
Bit_6		Low pressure circuit 2 analogue input	1
Bit_7		Thermal switch compressor 4	1

Manual reset: Hi byte – Word 2

Add. : **0xFD8d**		**Manually reset alarms (Hi byte – Word 1)**	
n_bit	***Label***	***Meaning***	***active at***
Bit_0		Maximum pressure compressor 4	1
Bit_1		Error probe ST1	1
Bit_2		Flow meter alarm	1
Bit_3		Error probe ST4	1
Bit_4		Antifreeze external circuits	1
Bit_5		Machine discharged	1
Bit_6		Configuration error	1
Bit_7		Control temperature too high error	1

To reset the manual alarms, reset the signalling bit in the previous 2 word and the alarm counters; refer to the following table:

Number operations per hour counter

Add. : **0xFD8E**	**Low pressure counter circuit 1**
Add. : **0xFD8F**	**Low pressure counter circuit 2**
Add. : **0xFD90**	**Thermal switch compressor 1**
Add. : **0xFD91**	**Thermal switch compressor 2**
Add. : **0xFD92**	**Thermal switch compressor 3**
Add. : **0xFD93**	**Thermal switch compressor 4**
Add. : **0xFD94**	**Thermal switch fan 1**
Add. : **0xFD95**	**Thermal switch fan 2**
Add. : **0xFD96**	**Low pressure circuit 1 analogue**
Add. : **0xFD97**	**Low pressure circuit 2 analogue**
Add. : **0xFD98**	**Antifreeze circuit 1**
Add. : **0xFD99**	**Antifreeze circuit 2**
Add. : **0xFD9A**	**Antifreeze external circuit**
Add. : **0xFD9B**	**Antifreeze external circuit**
Add. : **0xFD9C**	**Flow meter**

3.1.13 Ver

Area Index: 9
In this area the mask values and device version can be read, using Modbus function 04 (hex).
These values can be read using the following logic addresses (refer to the "data field" in the Modbus general manual):

Index	*VER*	Address
1	Mask and Version	1000 1000 0000 0001 (88 01 hex)

Word includes in LO byte the version and in HI byte the family.

3.1.14 Visibility of parameters and submenus

Area Index: 1
In this area, a “visibility value” may be assigned to each parameter or *label*, as described below:

label

Value	Meaning
0003	Parameter or *label* visible at all times
0258	Parameter or *label* visible if user password entered correctly (password = Pa H67)
0770	Parameter or *label* visible if user password entered correctly (password = Pa H67). Parameter cannot be modified.
0768	Parameter visible from PC only.

Some visibility settings are factory set.

3.2 Functions

The following table shows all available Modbus *functions* for reading and/or writing in the respective *logic areas*.

Function	Description	Application logic area
1 (01 hex)	Digital variables reading	5 : *Digital outputs*
2 (02 hex)	Digital variables reading	4 : *Digital inputs*
3 (03 hex)	Analogue variables reading	1–6-7-8 : *Parameters* – *Eeprom* – *Analogue outputs* – *Ram*
4 (04 hex)	Analogue variables reading	2 – 9 :*Probes* - *Ver*
15 (0F hex)	Digital variables writing	5 : *Digital outputs*
16 (10 hex)	Analogue variables writing	1–6-7-8 : *Parameters* – *Eeprom* – *Analogue outputs* – *Ram*

3.3 Example

Let us suppose that we want to write a value of 30 in the Energy 400 H52 parameter, with a network (slave) address of 1:

- First of all we must determine the parameter area index: 1
- This index is expressed with 5 bits: 00001
- Then we must determine the function code necessary to write in the index 1 (*parameters*) area: 10 (hex)
- The H52 parameter index: 54
- This index is expressed with 11 bits: 00000110110
- We combine the area index and the parameter index to form 2 bytes: 0000 1000 0011 0110 (08 36 hex)

The function frame is composed according to the modbus function (10 hex) (see the modbus manual):

Field	Query (hex)	RTU 8-bit (binary)	Description
Slave address	01	0000 0001	Address of the slave in the network
Function code	10	0000 1111	Function code
First bit address	08 35	0000 1000 0011 0101	Hexadecimal value to indicate the logic area 1 and the index 53
Number of variables	00 01	0000 0000 0000 0001	Hexadecimal value to indicate the number of variables to write
Byte count	02	0000 0010	No. bytes contained in the data field
Force data	00	0000 0000	If the area is a set of bytes this field is always on zero
Data	1E	0001 1110	Value to write: 30
CRC	xx xx	xxxx xxxx xxxx xxxx	CRC value calculated using appropriate algorithm (see general modbus manual)

Therefore the function to send will be (in hexadecimals):
01 10 08 35 00 01 02 00 1E xx xx

Whereas the response will be:

Field	Query (hex)	RTU 8-bit (binary)	Description
Slave address	01	0000 0001	Address of the slave in the network
Function code	10	0000 1111	Function code (echo)
First bit address	08 35	0000 1000 0011 0101	Hexadecimal value to indicate the logic area 1 and the index 53 (echo)
Number of variables written	00 01	0000 0000 0000 0001	Hexadecimal value to indicate the number of variables written
CRC	xx xx	xxxx xxxx xxxx xxxx	CRC codified by the device

With the following frame:
01 10 08 35 00 01 xx xx

3.3.1 Additional Examples

Other examples of function frames and response:
RL1 reading on the slave with address 1

- Query: 01 01 28 01 00 01
- Response: 01 01 01 xx

RL1 ON on the slave with address 1

- Query: 01 0F 28 01 00 01 01 01
- Response: 01 0F 28 01 00 01

RL1 OFF on the slave with address 1

- Query: 01 0F 28 01 00 01 01 00
- Response: 01 0F 28 01 00 01

ID2 - ID3 reading on the slave with address 1

- Query: 01 02 20 02 00 02
- Response: 01 02 01 xx xx

writing the vent. output 1 on the slave with address 1(xx=data)

- Query: 01 10 38 01 00 01 02 00 xx
- Response: 01 10 38 01 00 01

reading the vent. output 1 on the slave with address 1

- Query: 01 03 38 01 00 01
- Response: 01 03 02 00 xx

reading ST1 on the slave with address 1

- Query: 01 04 10 01 00 01
- Response: 01 04 02 xx xx

3.4 Error codes

If an error occurs, the Energy 400 device provides one of the following *error codes* in the data field:

error code	error	description
1	function error	The function in question has not been recognised
2	address error	The area does not correspond with the function, index 0, index non-existent in the requested area
3	data error	Too much data requested for the area in question, buffer length exceeded in the response

4 ANALITIC INDEX

Invensys Controls Italy S.r.L.
via dell'Industria, 15 Zona Industriale Paludi
32010 Pieve d'Alpago (BL) ITALY
Telephone +39 0437 986111
Facsimile +39 0437 989066
Internet http://www.climate-eu.invensys.com