

Modbus® protocol for pCO controllers



Modbus®



User manual

➔ **READ AND SAVE
THESE INSTRUCTIONS** ←

Integrated Control Solutions & Energy Savings

CONTENTS:

1	INTRODUCTION.....	3
2	CONNECTION MODES.....	3
2.1	Connection in RS485 mode	3
2.2	Connection in RS232 mode	3
3	SERIAL INTERFACES THE PROTOCOL IS AVAILABLE ON.....	4
4	PROTOCOL ACTIVATION.....	4
4.1	Selecting the supervisor address	4
4.2	Selecting the communication baud rate	5
4.3	Selecting the communication frame	5
5	MAPPING THE CAREL DATABASE TO MODBUS	6
5.1	Threshold.....	6
5.2	Relationship between Carel variable index and Modbus data address	6
5.3	Correspondence between Carel database and Modbus database.....	6
6	MODBUS COMMANDS.....	9
6.1	Error codes (Modbus exceptions)	9
6.2	Response to the Report Slave ID message	9
7	SUPERVISOR TIMEOUT SETTING.....	10

1 INTRODUCTION

The Modbus® Slave protocol can be used to directly connect pCO controllers to a Modbus® supervisory system. The protocol implemented is RTU (Remote Terminal Unit). Communication with the supervisor is managed by 8-bit data encoding and 16-bit CRC error control. Communication is asynchronous serial, 8 data bits, parity and settable stop bits; the baud rate and supervisor address can also be set.

2 CONNECTION MODES

Connection can be managed over both RS485 and RS232 (half duplex point-to-point), the latter only for serial communication using the corresponding optional card (see pCO sistema manual)

2.1 Connection in RS485 mode

A direct connection is established to the Modbus® network via the serial port on the controller or the optional card. The maximum baud rate depends on the type of serial interface used for connection and the type of controller (see par. 4.2). For the technical specifications, meanings of the connections (pins) and details on activation, see the instructions shown on the instruction sheet included in the packaging on the card itself, or in the controller instruction manual.

2.2 Connection in RS232 mode

Connection (point to point) is established by connecting the Modbus® supervisor directly to the controller, via the connector on the optional card. The maximum baud rate is 38400 bps. The type of cable used depends on the RS232 connector on the master. If, as in the majority of cases, the RS232 connector on the master is DTE⁴ (Fig.1), the cable used for connection must be a NULL MODEM cable DB9F – DB9F (Fig.2), in which pin 2 (RxD) at one end of the cable is connected to pin 3 (TxD) on the other end, and vice-versa.



Fig. 1



Fig. 2

For the technical specifications, meanings of the connections (pins) and details on the activation procedure, see the instructions shown on the instruction sheet included in the packaging on the optional card.

⁴ Data Terminal Equipment, DB9 pin, male

3 SERIAL INTERFACES THE PROTOCOL IS AVAILABLE ON

The serial interfaces that the protocol can be selected on depend on the type of controller used. Tab. 1 summarises the options.

Controller	pLAN	BMS	FIELDBUS	BMS_2	FIELDBUS_2
pCO2		<input checked="" type="checkbox"/>			
pCO1 ⁽¹⁾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
pCOXS ⁽¹⁾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
pCO3 ⁽¹⁾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
pCOCOMPACT ⁽¹⁾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
SuperNode ⁽²⁾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
pCO5 ⁽²⁾	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Tab. 1

⁽¹⁾ The protocol is not available on different serial interfaces at the same time.

⁽²⁾ The protocol is available on BMS_2 and either pLAN, BMS or FIELDBUS at the same time

4 PROTOCOL ACTIVATION

The protocol is activated by setting the system variables used as communication protocol selector for the required serial interface. The size of the variable database (registers and coils) will vary depending on the value used to activate communication. The system variable values required to activate Modbus® protocol are shown in Tab.2.

System variable	Value
COM_PROTOCOL_PLAN ⁽¹⁾	3, 30
COM_PROTOCOL_BMS ⁽¹⁾⁽²⁾⁽⁵⁾	3, 30, 33
COM_PROTOCOL_FIELDBUS ⁽¹⁾	5, 50
COM_PROTOCOL_BMS2 ⁽³⁾⁽⁴⁾	33

Tab. 2

⁽¹⁾ Values 30 and 50 are only allowed on controllers with 512Kbyte RAM.

⁽²⁾ The value 33 is only allowed on Supernode, pCOCOMPACT and pCO5 controllers.

⁽³⁾ The value 33 is only allowed on Supernode and pCO5 controllers

⁽⁴⁾ The value 33 can only be selected starting from BIOS 5.04 and higher.

⁽⁵⁾ The value 33 can only be selected starting from BIOS 5.17 and higher.

4.1 Selecting the supervisor address

The supervisor address can be set using the BMS_ADDRESS and BMS2_ADDRESS system variables. If the protocol is activated on the pLAN, BMS or FIELDBUS serial interface, the supervisor address is set using BMS_ADDRESS. If, on the other hand, the protocol is activated on BMS2, it's set using BMS2_ADDRESS (Tab. 3)

Serial	Supervisor address
pLAN, BMS, FIELDBUS	BMS_ADDRESS
BMS2	BMS2_ADDRESS

Tab. 3

4.2 Selecting the communication baud rate

The communication baud rate is selected based on the value assigned to the corresponding system variables for the serial interface the protocol has been activated on. The possible values of these variables are shown in Tab. 4.

System variable	Baud rate					
	1200	2400	4800	9600	19200	38400
COM_BAUDRATE_PLAN	0	1	2	3	4	
COM_BAUDRATE_BMS ⁽¹⁾⁽³⁾	0	1	2	3	4	5
COM_BAUDRATE_FIELDBUS	0	1	2	3	4	
COM_BAUDRATE_BMS2 ⁽²⁾⁽³⁾	0	1	2	3	4	5

Tab. 17

⁽¹⁾ The value 5 (38400 bps) can only be selected on Supernode, pCOCOMPACT and pCO5.

⁽²⁾ The value 5 (38400 bps) can only be selected on Supernode and pCO5.

⁽³⁾ The value 5 (38400 bps) can only be selected starting from BIOS 5.17 and higher.

4.3 Selecting the communication frame

The format of the communication frame (parity, number of stop bits) is set by assigning appropriate values to the COM_CONFIG system variables⁵. These variables are bit fields. The meaning of the bits for each of these variables is described in Tab. 5

Bit15 – Bit6	Bit5 – Bit4	Bit3 – Bit1	Bit0
---	Parity	---	Stop

Tab 5

where

Bit	Nome	Descrizione
0	Stop	Number of stop bits 0: 2 stop bits 1: 1 stop bits
1÷3	Not used	
4÷5	Party	Sets the communication parity Bit4 = 0, Bit5 = 0: no parity Bit4 = 1, Bit5 = 0: even parity Bit4 = 0, Bit5 = 1: odd parity
6÷15	Not used	

As an example, to transmit with 1 stop bit and odd parity, the variables must be assigned the value 0x21 (d33).

⁵COM_CONFIG_PLAN, COM_CONFIG_BMS, COM_CONFIG_FIELDBUS, COM_CONFIG_BMS2, COM_CONFIG_FIELDBUS2

5 MAPPING THE CAREL DATABASE TO MODBUS®

The digital variables in the Carel database are mapped to Modbus coil variables, while both analogue and integer variables are mapped to Modbus register variables. The values of registers are represented as WORDS (16 bits). The values of registers corresponding to Carel analogue variables correspond to the value of the variables multiplied by 10.

Carel type	Modbus® type
Digital	Coil
Analogue	Register
Integer	Register

The distinction between Modbus® registers corresponding to Carel analogue and integer variables is made by using an offset (threshold), the value of which depends on the value used to activate the protocol and the value of the BMS_EXTENSION system variable (see par. 5.2)

5.1 Threshold

The threshold, used to distinguish Modbus registers reserved for Carel analogue variables from those reserved for Carel integer variables, depends on the protocol activation value (Tab. 2) and the value of the BMS_EXTENSION system variable. The frame address (PDU) of threshold Th is described in Tab. 6.

MODBUS® SELECTION NUMBER	BMS_EXTENSION	
	0	1
3, 5	Th = 128	Th = 208
30, 50	Th = 5001	Th = 5001
33	Th = 128	Th = 5001

Tab. 6

5.2 Relationship between Carel variable index and Modbus data address

The threshold described in the previous paragraph is used to calculate, based on the supervisor index of the Carel variable (**Analogue, Integer, Digital**), the corresponding Modbus data (Register, Coil) index (see par. 5.1). Tab. 7 shows the correspondence, where Th refers to the threshold described in Tab. 6.

CAREL	MODBUS®
A[N]	REGISTER[N]
I[N]	REGISTER[Th+N]
D[N]	COIL[N]

Tab. 7

Consequently, for example, the first analogue variable A[1] will correspond to register Reg[2], the second A[2] to register Reg[3], and so on. The same is true for digital variables, only these are mapped to Modbus coils rather than registers. Similarly, the first integer variable I[1], will be mapped to register Reg[129] if the threshold Th is equal to 128, to register Reg[209] if the threshold is 208, or to register Reg[5002] if the threshold is 5001.

5.3 Correspondence between Carel database and Modbus® database

The correspondence between the Carel variable database and the Modbus® protocol database depends on the number chosen to activate the protocol (Tab. 2) and the threshold Th, i.e. the value of the BMS_EXTENSION variable.

5.3.1 Carel – Modbus correspondence with Modbus® versions 3 and 5

The extension of the Carel database depends on the value of the BMS_EXTENSION variable, as shown in Tab. 8.

SUPERVISOR_EXTENSION = 0			SUPERVISOR_EXTENSION = 1		
127 A	127 I	199 D	207 A	207 I	207 D

Tab. 8

In this case the correspondence is represented in Tab. 9. The Modbus addresses are the physical addresses of the registers and coils (see par. 5.1).

BMS_EXTENSION = 0				BMS_EXTENSION = 1			
CAREL		MODBUS		CAREL		MODBUS	
Type	Address	Type	Address	Type	Address	Type	Address
Digital	1	Coil	1	Digital	1	Coil	1
Digital	2	Coil	2	Digital	2	Coil	2
Digital	...	Coil	...	Digital	...	Coil	...
Digital	198	Coil	198	Digital	206	Coil	206
Digital	199	Coil	199	Digital	207	Coil	207
Analogue	1	Register	1	Analogue	1	Register	1
Analogue	2	Register	2	Analogue	2	Register	2
Analogue	...	Register	...	Analogue	...	Register	...
Analogue	126	Register	126	Analogue	206	Register	206
Analogue	127	Register	127	Analogue	207	Register	207
Integer	1	Register	129	Integer	1	Register	209
Integer	2	Register	130	Integer	2	Register	210
Integer	...	Register	...	Integer	...	Register	...
Integer	126	Register	126	Integer	206	Register	414
Integer	127	Register	255	Integer	207	Register	415

Tab. 9

5.3.2 Carel - Modbus correspondence with Modbus® versions 30 and 50

The database of the three types of variables extends beyond the maximum index managed by the Carel protocol (207). The extension of the Carel variable database in this case is 5000A, 5000I, 2048D. The correspondence is shown in Tab. 10. The indices in bold are those managed by the Carel protocol, while those in italics correspond to the extension beyond the maximum index. In other words, with this type of configuration, the application software can manage addresses that go beyond the maximum envisaged by the Carel protocol for each type of variable. The Modbus® addresses are the physical addresses of the registers and coils (see par. 5.1). The BMS_EXTENSION variable does not modify the threshold Th that distinguishes the registers corresponding to analogue variables from those corresponding to integer variables (Tab. 6).

CAREL		MODBUS®	
Type	Address	Type	Address
Digital	1	Coil	1
Digital	2	Coil	2
Digital	...	Coil	...
Digital	207	Coil	207
<i>Digital</i>	<i>208</i>	<i>Coil</i>	<i>208</i>
<i>Digital</i>	<i>209</i>	<i>Coil</i>	<i>209</i>
<i>Digital</i>	<i>...</i>	<i>Coil</i>	<i>...</i>
<i>Digital</i>	<i>2047</i>	<i>Coil</i>	<i>2047</i>
<i>Digital</i>	<i>2048</i>	<i>Coil</i>	<i>2048</i>
Analogue	1	Register	1
Analogue	2	Register	2
Analogue	...	Register	...
Analogue	207	Register	207
<i>Analogue</i>	<i>208</i>	<i>Register</i>	<i>208</i>
<i>Analogue</i>	<i>...</i>	<i>Register</i>	<i>...</i>
<i>Analogue</i>	<i>4999</i>	<i>Register</i>	<i>4999</i>
<i>Analogue</i>	<i>5000</i>	<i>Register</i>	<i>5000</i>
Integer	1	Register	5002
Integer	2	Register	5003
Integer	...	Register	...
Integer	207	Register	5208
<i>Integer</i>	<i>208</i>	<i>Register</i>	<i>5209</i>
<i>Integer</i>	<i>...</i>	<i>Register</i>	<i>...</i>
<i>Integer</i>	<i>4999</i>	<i>Register</i>	<i>10000</i>
<i>Integer</i>	<i>5000</i>	<i>Register</i>	<i>10001</i>

Tab. 18

5.3.3 Carel - Modbus correspondence with Modbus® version 33

The size of the variable database and the corresponding extension beyond the maximum index envisaged by the Carel protocol depend on the value of the BMS_EXTENSION system variable, as shown in Tab. 11.

SUPERVISOR_EXTENSION = 0		SUPERVISOR_EXTENSION = 1	
127 A	10000 I	2048 D	5000 A
			10000 I
			2048 D

Tab. 11

When the value is 0 the analogue variable database has no extension, and the size coincides with the value envisaged for versions 3 and 5 of the Modbus protocol (see Tab. 8). When the value is 1, on the other hand, the extension is the same as envisaged by versions 30 and 50 (see Tab. 10). The integer and digital variables always extend beyond the maximum index managed by the Carel protocol (207). The correspondence is shown in Tab. 12, where the indices in bold are those managed by the Carel protocol, while those in italics correspond to the extension of the database if the BMS_EXTENSION system variable is 1. The Modbus addresses are the physical addresses of the registers and coils (see par. 5.1).

BMS_EXTENSION = 0				BMS_EXTENSION = 1			
CAREL		MODBUS®		CAREL		MODBUS®	
Type	Address	Type	Address	Type	Address	Type	Address
Digital	1	Coil	1	Digital	1	Coil	1
Digital	2	Coil	2	Digital	2	Coil	2
Digital	...	Coil	...	Digital	...	Coil	...
Digital	207	Coil	207	Digital	207	Coil	207
<i>Digital</i>	<i>208</i>	<i>Coil</i>	<i>208</i>	<i>Digital</i>	<i>208</i>	<i>Coil</i>	<i>208</i>
<i>Digital</i>	<i>209</i>	<i>Coil</i>	<i>209</i>	<i>Digital</i>	<i>209</i>	<i>Coil</i>	<i>209</i>
<i>Digital</i>	<i>...</i>	<i>Coil</i>	<i>...</i>	<i>Digital</i>	<i>...</i>	<i>Coil</i>	<i>...</i>
<i>Digital</i>	<i>2047</i>	<i>Coil</i>	<i>2047</i>	<i>Digital</i>	<i>2047</i>	<i>Coil</i>	<i>2047</i>
<i>Digital</i>	<i>2048</i>	<i>Coil</i>	<i>2048</i>	<i>Digital</i>	<i>2048</i>	<i>Coil</i>	<i>2048</i>
Analogue	1	Register	1	Analogue	1	Register	1
Analogue	2	Register	2	Analogue	2	Register	2
Analogue	...	Register	...	Analogue	...	Register	...
Analogue	126	Register	126	Analogue	206	Register	206
Analogue	127	Register	127	Analogue	207	Register	207
				<i>Analogue</i>	<i>208</i>	<i>Register</i>	<i>208</i>
				<i>Analogue</i>	<i>...</i>	<i>Register</i>	<i>...</i>
				<i>Analogue</i>	<i>4999</i>	<i>Register</i>	<i>4999</i>
				<i>Analogue</i>	<i>5000</i>	<i>Register</i>	<i>5000</i>
Integer	1	Register	129	Integer	1	Register	5002
Integer	2	Register	130	Integer	2	Register	5003
Integer	...	Register	...	Integer	...	Register	...
Integer	207	Register	335	Integer	207	Register	5208
<i>Integer</i>	<i>208</i>	<i>Register</i>	<i>336</i>	<i>Integer</i>	<i>208</i>	<i>Register</i>	<i>5209</i>
<i>Integer</i>	<i>...</i>	<i>Register</i>	<i>...</i>	<i>Integer</i>	<i>...</i>	<i>Register</i>	<i>...</i>
<i>Integer</i>	<i>9999</i>	<i>Register</i>	<i>126</i>	<i>Integer</i>	<i>9999</i>	<i>Register</i>	<i>15000</i>
<i>Integer</i>	<i>10000</i>	<i>Register</i>	<i>10128</i>	<i>Integer</i>	<i>10000</i>	<i>Register</i>	<i>15001</i>

Tab. 12

6 MODBUS COMMANDS

The Modbus commands implemented are described in Tab. 13. Digital variables can be read both with Read Coil Status (code 1) and Read Input Status (code 2). The controller manages these supervisor requests in the same way. Similarly, analogue and integer variables can be read with Read Holding Register (code 3) and Read Input Register (code 4).

Reading/writing of supervisor variables using the commands described in Tab. 13 during normal controller operation depends on the application software that manages these.

The controller does not verify the range of values received by writing analogue and/or integer supervisor variables using the Modbus Preset Single/Multiple Register commands.

Function code	Modbus® command	Effect
1	Read Coil Status	Read digital variables
2	Read Input Status	Read digital variables
3	Read Holding Register	Read analogue and/or integer variables
4	Read Input Register	Read analogue and/or integer variables
5	Force Single Coil	Write a digital variable
6	Preset Single Register	Write an analogue or integer variable
15	Force Multiple Coils	Write a consecutive series of digital variables
16	Preset Multiple Registers	Write a consecutive series of analogue and/or integer variables
17	Report Slave ID	Request information from the controller

Tab. 19

6.1 Error codes (Modbus® exceptions)

When the controller cannot perform requests from the supervisor, it returns error messages (Modbus® exceptions). The Modbus® exception codes managed by the controller are described in Tab. 14.

Type	Exception code	Description
0x01	ILLEGAL FUNCTION	The requested function is not implemented on the controller
0x02	ILLEGAL DATA ADDRESS	One or more addresses in the request are out of range

Tab. 14

6.2 Response to the Report Slave ID message

The Report Slave ID message is used to request specific information from the controller, such as the BIOS version and mapping of the Carel variable database to the Modbus® database. The structure of the response message is described in Tab. 16.

6.2.1 BIOS version representation

The BIOS version is represented respectively by byte 6 and 7. For test or unofficial BIOS versions the content of the two bytes will be the ASCII codes for letters 'T' and 'E'. Otherwise it will be the BIOS version number, in the format a.bb (see Tab. 15)

BIOS version	BYTE 6	BYTE 7
TE	'T' = 0x54	'E' = 0x45
A.BB	A	BB

Tab. 15

6.2.2 Example response

Fig. 4 shows an example of a response to the Report Slave ID message by the controller, assuming the following:

Address = 1

BIOS = 5.12

Modbus version = 33

BMS_EXTENSION = 1

01,11,16,C9,FF,05,0C,13,88,27,10,08,00,00,02,13,89,13,8B,3A,9A,00,02,08,01,64,08

Fig. 4

Byte	Contained	Description
4	Controller code	201 (type pCO)
5	0xFF	Run status: ON
6,7	BIOS version	BIOS version installed on the controller
8,9	Maximum number of analogue var.	See par. 5
10,11	Maximum number of integer var.	See par. 5
12,13	Maximum number of digital var.	See par. 5
14,15	Data address of the register corresponding to the first analogue variable	See par. 5
16,17	Data address of the register corresponding to the last analogue variable	See par. 5
18,19	Data address of the register corresponding to the first integer variable	See par. 5
20,21	Data address of the register corresponding to the last integer variable	See par. 5
22,23	Data address of the coil corresponding to the first digital variable	See par. 5
24,25	Data address of the coil corresponding to the last digital variable	See par. 5

Tab. 20

7 SUPERVISOR TIMEOUT SETTING

The value of the timeout to be set on the supervisor to await a response from the controller depends on the communication baud rate. Typical values when the baud rate is 19200 are:

timeout: ≥1000 ms

turnaround delay⁶: from 50 to 200 ms

⁶ Delay before a subsequent request after having received the response to the previous one.

CAREL

CAREL INDUSTRIES HeadQuarters

Via dell'Industria, 11 - 35020 Brugine - Padova (Italy)

Tel. (+39) 049.9716611 - Fax (+39) 049.9716600

e-mail: Carel@Carel.com - www.Carel.com

Agency: