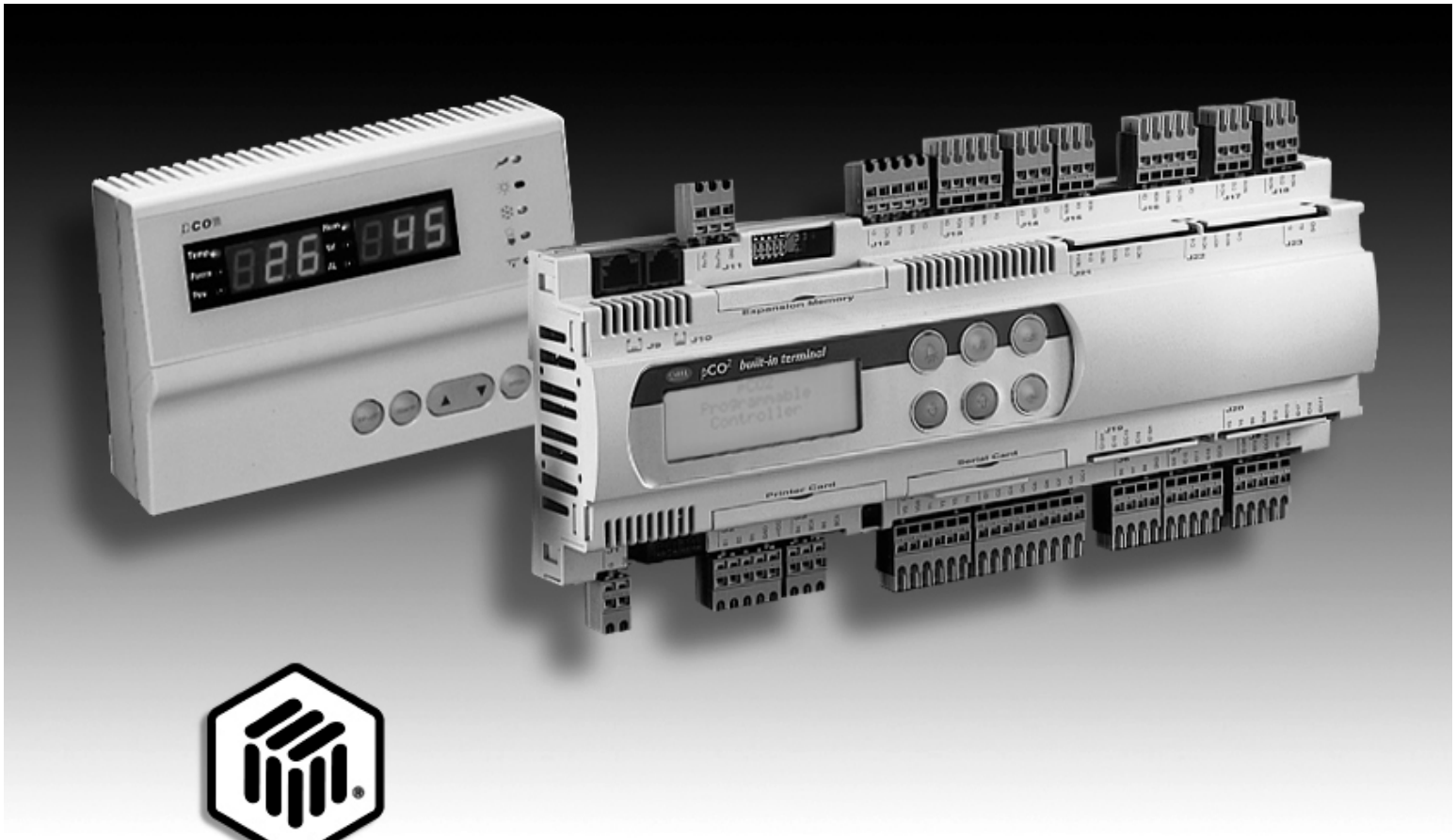
pCO/pCO² LonWorks®



LONMARK® PARTNER

Manuale d'installazione e uso

Installation and user manual

LEGGI E CONSERVA QUESTE ISTRUZIONI

READ AND SAVE THESE INSTRUCTIONS

CAREL

Technology & Evolution

INDEX

INTRODUCTION

The interface serial cards to a LonWorks® network consist in optional electronic cards of the pCO² and pCO controllers that allow you to interface directly the pCO or pCO² to a LonWorks® network.
The use of these cards requires knowledge and experience in the installation and maintenance tools of the LonWorks® networks .

1. General characteristics

***IMPORTANT WARNING:** To be ready for use, the card must be programmed by the user depending on the application program installed on the pCO or pCO².*

*In order to program the card the user must follow these steps (which are described in detail in **Programming the interface card**):*

1. *fill in the table which describes the exchange variables between pCO or pCO² and LonWorks® network according to the desired specifications;*
2. *give the table back to Carel. According to this table, Carel will produce the application file which will be returned to the user;*
3. *download the above-mentioned application in the interface memory by means of LonMaker™ or other LonWorks® network installation and maintenance tools.*

*Since it is not possible to know the application program to install on the pCO or pCO² in advance, in the factory the cards are programmed with a basic program that allows only to load the application and to handle three non standard variables, which are described in **Programming the interface card**.*
For applications or standard usages of pCO or pCO², the tables on point 1 may be already available; in this case, Carel will supply you directly with the application, which is to be downloaded in the interface memory.

The cards are available in two models both for pCO² and pCO and differ according to the type of interface on the LonWorks® network side.

2. Hardware architecture

2.1 Codes

for pCO	
PCOSERFTTL	*interface with FTT-10A 78kbs (TP/FT-10)*
PCOSER485L	*interface with RS485 39kbs (TP/485-39)*

for pCO²	
PCO20LFTT0	*interface with FTT-10A 78kbs (TP/FT-10)*
PCO20L4850	*interface with RS485 39kbs (TP/485-39)*

Tab. 2.1.1

2.2 Description

2.2.1 Physical channels

According to the models, the cards allow you to interface with two physical channels, TP/FT-10 and TP-RS485-39, which are described in the LonWorks® literature.

PCOSERFTTL *and* ***PCO20LFTT0*** *use an Echelon® FTT-10 transceiver, which is approved to be used on the TP/FT-10 channel.*
This channel is characterized by the following main features :

- *it consists of up to 64 nodes on a single network segment;*
- *the nodes are connected with free topology wiring; thus they will accommodate bus, star, loop, or any combination of these topologies;*
- *data rate: 78.125kbps;*
- *Maximum distance: 500m in case of free topology connection between the nodes; 2700m in case of bus connection with double line ending.*

For further details, refer to the official documentation LonWorks® FTT-10A Free Topology Transceiver User's Guide.

For more details you can refer to the official LonWorks® guidelines LonMark® Layers 1-6 Interoperability Guidelines, Version 3.0, pages. 2-18÷2-27, Physical Layer chapter, Free Topology Twisted Pair Transceiver Communication on TP/FT-10 Channels paragraph.

PCOSER485L *and* ***PCO20L4850*** *use an Echelon® TP/485-39 transceiver that supports the EIA RS-485 specification. In particular:*

- *the maximum number of nodes is 32;*
- *it is specified to support only bus wiring;*
- *the transceiver data rate is of 39kbps;*
- *Maximum distance: 1200m.*

For further details, refer to the document by the Electronic Industries Association (1983) EIA RS-485 Standard, and to the official documentation LonWorks® Twisted Pair Control Module User's Guide.

2.2.2 Interface card planimetry

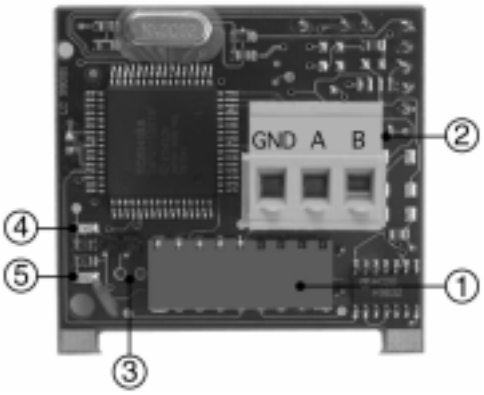


Fig. 2.3.1 *- PCOSERFTTL, PCOSER485L*

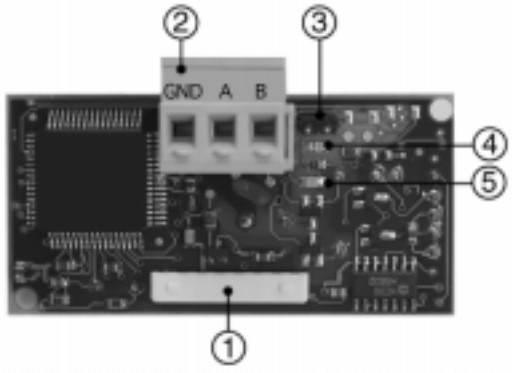


Fig. 2.3.2 *- PCO20LFTT0, PCO20L4850*

1. *connection to the pCO/pCO²*
2. *terminal block to the LonWorks® network (GND, A, B)*
3. *service pin*
4. *service green LED*
5. *anomaly red LED.*

To activate the service pin, simply short-circuit the two pins for a moment , using the tip of a screwdriver or similar.
The service pin is available only in the node installation phase. When the pin is activated, the node sends a broadcast message in the LonWorks® network containing the necessary information in order to be identified.

2.2.3 LED meaning and function

The service green LED:

- *signals the node status , as per the LonWorks® protocol:*
  broken hardware: always ON or always OFF;
  patterned node (normal operation): ½ second ON, then always OFF;
  NOT patterned node: flashing at ½Hz;
  node without application program: 1s ON, 2s OFF, then always OFF;
  node in continuous reset: flashing;
- *remains on during the activation of the service pin;*
- *remains on for a second when receiving a WINK command from the network (see* ***The WINK event****).*

The anomaly red LED:

- *signals the possible problems of connection between the pCO and pCO² cards.*

If the red LED switches on, be sure having followed all the instructions that are described in ***Installation*** *(in particular, check having set the communication baud rate of the pCO or pCO² at 4800).*

3. Installation

IMPORTANT WARNING. When handling the card, please follow the instructions below:
The electrical damages, which may occur to the electronic components, happen almost always as a result of electrostatic discharges that are caused by the operator. Therefore, suitable precautions must be taken when handling these components:

- *before using any electronic component or card, touch a grounding (not touching the card does not prevent a spike, as static electricity can produce a 10000V spike discharge, which can form an arc of about 1cm);*
- *all components must be kept inside their original packages as long as possible. If necessary, take the card from its package and place it into an antistatic package without touching the sides of the card on which the electrical components are placed;*
- *you must not use plastic bags, polystyrene or non-antistatic sponge;*
- *you must not pass the card directly to other operators (to avoid electrostatic induction and discharges).*

3.1 Connection to the LonWorks® network

The physical connection to the LonWorks® network is carried out by means of the plug-in terminal connector on the card and it must be made according to the indications and specifications supplied by Echelon.
For further installation and maintenance information, please refer to the LonWorks® literature.

3.2 Connection to the pCO board

To connect the interface card to the pCO board:

- *power off the pCO board;*
- *insert the interface card in the correspondent connector; the card must be slotted into the guides (two holes) located in the sides of the contacts;*
- *when mounting, check that the interface card is perfectly vertical in respect to the pCO to avoid damaging the contacts;*
- *the pins of the connector on the pCO board must be properly inserted into the connector on the interface card.*

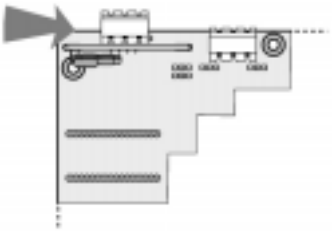


Fig. 3.2.1

3.3 Connection to the pCO² board

To install the card in the pCO² unit respect the following instructions (Fig.3.3.1÷3.3.4):

1. *power off the pCO² board;*
2. *remove the serial card placement cover with a screwdriver (Fig. 3.3.1);*
3. *remove the pre-punctured plastic part from the cover and you get the hole that corresponds to the outlet of the three-way-connector with a wire cutter (Fig. 3.3.2);*
4. *insert the optional card into the corresponding connector, initially inserting the card obliquely and then taking care that the card is firmly placed on both plastic supports on the pCO² case (Fig. 3.3.3);*
5. *close the cover making the outside card terminal fit with the punched hole made on the cover (Fig. 3.3.4).*

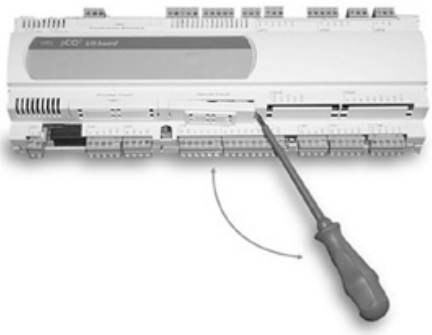


Fig. 3.3.1

Fig. 3.3.2

Fig. 3.3.3

Fig. 3.3.4

3.4 pCO and pCO² settings

On the pCO and pCO² only the baud rate of the serial communication must be set to 4800.
The address of the pCO/pCO² is not important, since it is automatically recognized by the interface card.

3.4.1 Adaptation of the application software (referred to EasyTools developers)

The application software, which is installed in the pCO/pCO², must have the following characteristics:

- *it has to manage the system variable INI_BAUD_SPV for the setting of the communication speed with the interface board at 4800 baud;*
- *it has to transfer the pCO/pCO² variables, which are to be available in LonWorks® network, to the supervision of the R*IN/R*OUT atoms;*
- *it has to grant the agreement between the pCO/pCO² variables range, which are to be available in LonWorks® network, and the SNVT type range, which is associated with the variables themselves (see* ***Construction of the customized program for the pCO/pCO² applications****).*

EXAMPLE

Measured size	***pCO/pCO² variable type***	***pCO/pCO² resolution***	***pCO/pCO² value***	***associated SNVT***	***LonWorks resolution***	***LonWorks network value***
Ambient temperature	Analogic	*0,1°C*	*10,2*	*SNVT_temp_p*	*0,01°C*	*1020*
			25,1			*2510*
Ambient humidity	Analogic	*0,1%*	*50,4*	*SNVT_lev_percent*	*0,005%*	*1080*
			62,7			*12540*
Alarm delay time	Entire	*1s*	*600*	*SNVT_time_sec*	*0,1s*	*6000*
			15			*150*

Table 3.4.1.1

4. Programming the interface card

The card must be programmed by the user according to the application program installed on the pCO/pCO².
The program resides in the flash memory housed on the Neuron® Chip, the electronic chip that handles the protocol of the LonWorks® networks, the LonTalk™. It is possible to program the flash memory directly via the LonWorks® network, using the LonMaker™ or NodeBuilder® installation and maintenance systems.

4.1 Network variables

The pCO/pCO² LonWorks® interface can manage up to 62 network variables.
Of these 62 variables:

- *59 can be directly associated to the same number of variables defined in the pCO/pCO² application program.*
  *The user must specify the desired association, as well as the type of variable and the direction (see **Construction of the customized program for the pCO/pCO² applications**).*
  On the basis of the provided specifications, Carel will create the file that the user will download to the interface card.
- *The remaining 3 variables are for the management of the interface itself. These are non-standard variables that allow the visibility of all the variables that are defined in the pCO/pCO² application program (see **Description of the non-standard variables**).*

4.2 Construction of the customized program for the pCO/pCO² applications

The first step is the creation by the user of a 5-column-table, like the example here below, which defines the exchange variables between pCO/pCO² and the LonWorks® network according to the desired specifications.

Type	pCO index	Name NV	Type NV	Direction
ANL	1	nva_1_out	30	output
ANL	2	nva_2_in	30	input
ANL	3			
...	...	...	...	...
ANL	207	nva_207_in	39	input
INT	1	nvi_1_in	9	input
...	...	...	...	...
INT	207	nvi_207_in	9	input
DGT	1	nvd_1_out	95	output
...	...	...	...	...
DGT	207	nvd_207_out	95	output

Tab. 4.2.1

The empty table NV_TABLE.XLS for Microsoft Excel for Windows 95, Version 7.0 is available at the WEB address www.attiva.it/carel/aggiorna/systems.htm

*The **Type** and **pCO index** columns are fixed.*

- *The Type column indicates the type of variable of the pCO/pCO² (ANL for analogue, INT for integer and DGT for digital);*
- *the pCO index column is the Carel's index corresponding to such variable.*

***WARNING**: the meaning of the variables, the corresponding index and their network availability depend on the application program installed on the pCO/pCO².*
*The pCO index may range from 1 to 207 per type. In the generic table, therefore, there are 207*3 = 621 rows, corresponding to the addressable space of the pCO/pCO²; only the variables, which are transmitted in the network (with the R*IN/R*OUT atoms), are available to the user for customization purposes.*
*You can find the table describing the meaning, type and index (pCO index) of the network variables in the manual of the application program at **Supervisor database** entry.*

*The fields relating to the **Name NV**, **Type NV** and **Direction** columns must be filled in (in Tab. 4.3.1 you can find some examples in italics of how they can be filled).*

- *in Name NV you must specify the name you want(MAX. 16 types) to associate with the network variable, which will be used as an external LonWorks® interface;*
- *in Type NV you must specify the type that is the number specified by the SNVT (Standard Network Variables Types), which identifies the physical quantity and the format associated with the network variable. The Standard Network Variable Types (SNVTs) facilitate interoperability by providing a well-defined interface for communication between nodes made by*

different manufacturers. A node may be installed in a network and logically connected to other nodes through the network variables provided that the data types match.

***WARNING**: The list of all the available SNVT variables, which are managed by the interface, and details of their definitions is provided in **appendix A**. The number , which is associated to the desired variable type and is in the first column (SNVT number), is the one to be entered in the table Type NV .*

- *in Direction you must specify the direction in input or output of the variable:*
  from the point of view of the pCO/pCO², input defines those values that are acquired by the LonWorks® bus and then copied into the memory of the pCO/pCO²;
  on the other hand, output specifies the network variables that export the values, which the pCO/pCO² generate internally, to the LonWorks® bus.

EXAMPLE.
Suppose you want to make the room temperature measure, which is made by a pCO/pCO² (only reading variable), available on a LonWorks® network and in the specific application program installed in the pCO/pCO² that variable is represented by the analogical variable with address 1. The first raw of the table should be filled with the following values:

Name NV: insert the name you want to describe the variable to the other nodes of the network; ex. nva_1_out
Type_NV: 39 (SNVT_temp)
Direction: output

That is:

Type	***pCO index***	***Name NV***	***Type NV***	***Direction***
ANL	*1*	*nva_1_out*	*39*	*output*
…	…	…	…	…

Suppose then you want to make the room temperature set point (reading/writing variable) available on a LonWorks® network as well and this one is represented by the analogical variable with address 13. The table should be filled with the following values, creating two LonWorks® variables corresponding to the same pCO index:

…	…	…	…	…
ANL	*13*	*nva_13_in*	*39*	*input*
ANL	*13*	*nva_13_out*	*39*	*output*
…	…	…	…	…

Once the tables have been filled in, entering only the rows required (59 rows are available, since 3 are reserved, and 62 is the maximum number of rows that is expected in the LonTalk protocol). They are then saved in .TXT format with tab stops, using the name NV_TABLE.TXT.
The table must be given to Carel that will produce the application file to download in the memory of the interface card.
The application will be returned to the customer as a file having .NXE extension.
The .NXE file is then ready to be loaded by the user on the target node through the LonWorks® bus itself; a special tool, such as LonMaker® may be used for this purpose, allowing the node to operate.

Besides the .NXE file, Carel will return two more files (not essential to the interface programming).

- *A node's external interface file (.XIF extension) provides basic information regarding a node and can be used by a network management tool to allow configuration of the node.*
- *A Report File (.REP extension).*

To sum up briefly, the user should make the following steps:

1. *Fill in the table, which describes the exchange variables between pCO/pCO² and LonWorks® network according to the desired specifications, and save the table in text format.*
2. *Give back the table to Carel. According to this table, Carel will produce the application file. NXE that will be returned to the user.*
3. *Download the .NXE file in the interface memory by means of LonMaker™ or other LonWorks® network installation and maintenance tools.*

4.3 Description of the non-standard variables

The following three variables are non standard variables and are used to communicate particular drives to the pCO/pCO² and to check the results. These drives allow the visibility of all the pCO/pCO² variables in LonWorks® network, going beyond the physical limits of the Neuron® Chip memory, which allows the addressing of maximum 62 network variables.

4.3.1 xif_data

It allows displaying the status of the interface and checking the writing or reading operations that are carried out . It is transmitted only in the event of variation of one of its fields:

```
network output struct {
        unsigned long    sfw_idnt;
        unsigned short   pco_addr;
        unsigned short   xif_stat;
        unsigned short   xif_flag;
} xif_data;
```

***sfw_idnt**: software version.*
This is codified in hexadecimal form, with the first two figures representing the version number before the point, and the remaining two representing the number after the point. In the case of the 1.00 version, the identifier is thus 0x0100.

***pco_addr**: address of the pCO/pCO².*
On reset it is void, then it takes the value set on the pCO/pCO² and read in the initial phase.

***xif_stat**: status of the interface.*
This can take the following values:
0: connection to pCO/pCO² and acquisition of the address (immediately following the reset),
1: connection to pCO/pCO² activated,
2: initial acquisition of all the pCO/pCO² variables,
3: normal operation
The 0 and 1 statuses occur very quickly, while status 2 can last some seconds, according to the number of variables on the pCO/pCO² side. In normal operation the interface is in status 3.

***xif_flag**: acknowledge of write to pCO/pCO².*
This is set to 0 following the writing of a pCO/pCO² variable from the LonWorks® network. It is set to 1 when the pCO or pCO² confirms receipt of the write. As a matter of fact, the LonTalk protocol ensures the transfer of data from the node of origin to the node represented by the interface, while the transfer from the interface to the pCO/pCO² is managed by the interface itself.

4.3.2 wr_cmnd

It allows getting information on the status of the interface and writing or reading any pCO/pCO² variable. Its fields are collected in the following structure:

```
network input struct {
        unsigned short   cmnd;
        unsigned char    type;
        unsigned short   indx;
        signed long      data;
} wr_cmnd;
```

***cmnd**: requested command .*
The following commands are featured:
0: forced propagation of the xif_data variable (useful in examining the status of the interface). In this case the other fields have no meaning and are not used.
1: writing of any pCO/pCO² variable.
2: reading of any pCO/pCO² variable. In this case, the data field is ignored. After the receipt of this command, the current value of the variable and each of its successive variations will be mirrored in the variable rd_prmt.

***type**: type of pCO/pCO² variable in writing or reading (A, I, or D),*
***indx**: index of the pCO/pCO² variable in writing or reading (from 1 to 207),*
***data**: value of the pCO/pCO² variable in writing (from–32767 to 32767).*

4.3.3 rd_prmt

It allows reading the current value of any pCO/pCO² variable (selected by wr_cmnd) and each of its successive variations to be read. It is transmitted only in the event of variation of one of its fields:

```
network output struct {
        unsigned short    stat;
        unsigned char     type;
        unsigned short    indx;
        signed long       data;
} rd_prmt;
```

***stat**: status relating to the reading of the variable from the pCO/pCO².*
The possible values are:
0: reading in progress, thus the data field has no meaning (immediately following the receipt of a reading command using wr_cmnd)
1: reading available; the data field contains the current value of the variable.
2: reading not completed (the pCO or pCO² does not respond to the request). This condition can occur if the selected variable has not been defined in the pCO/pCO² application program.

***type**: type of pCO/pCO² variable in reading (A, I, or D),*
***indx**: index of the pCO/pCO² variable in reading (from 1 to 207),*
***data**: value of the pCO/pCO² variable (from–32767 to 32767).*
The value is updated upon each variation of the variable.

5. Program operation

When the pCO/pCO² is switched on, power is also supplied to the interface card.
Immediately after the initial reset procedure, the interface card sends a service message through the network, which is received, if required, by the supervisor. It then begins to send inquiries to the pCO/pCO², in order to acquire first its address, and then the values of its available variables, using the force command and subsequent inquiries,.
The entire phase lasts some seconds, according with the number of variables.
Once it reached the normal operating status, the interface card send continuous inquiries to the pCO/pCO², in order to monitor the situation.
After the initial loading, the network variables, which are defined as outputs by the user, will have values reflecting those of the pCO/pCO², and thus will be available to any external device which wishes to make use of the interface, communicating with it through the LonWorks® bus.
The network variables, which, on the other hand, have been defined as inputs by the user, are at this point connected to the pCO/pCO², and any value written to these forces the application to initiate a session of writing to the pCO/pCO², which then copies the values internally.
For the complete management of the interface, a reserved network variable is available called xif_data, which consists in a structure with information on the status of both the interface itself and the application. These information are useful for having a required feedback on the reading or writing operations performed.
There are two reserved network variables as well: one is called rd_prmt and the other wr_cmnd. They allow the visibility of all the pCO/pCO² variables (see ***Description of the non standard variables****).*

5.1 The WINK event

On the LonWorks® network a generic supervisor may send the WINK command to a specific node. In this way an event is generated, to which the application at that node may respond with any action that the programmer decides to set.
In this specific case, the interface turns on the service LED for one second, thus allowing the identification of the correct operation of the connection between the interface and the LonWorks® bus.

6. Test application

A test application is available for pCO and pCO². Via LonWorks® network, it allows:

- *showing the status of all inputs,*
- *setting the status of all outputs*
- *setting some other variable values.*

WARNING: *All the following files are available in the PCO_TEST.ZIP file in the WEB page www.attiva.it/carel/aggiorna/systems.htm at the entry Test application programs ....*

6.1 Attached files

The following files make up the test program and allow its use:

For pCO

TESTEN.BIN	*Binary file to download in the pCO Eprom (1MBit).*

For pCO²

WINLOAD32.EXE:	*executable file for the loading of the program in the pCO²;*
SERIAL.VXD:	*32bit driver of the serial door for Windows 95/98;*
WINLOAD32.DOC:	*WinLOAD documentation in Microsoft Word 97 format (quick reference);*
TESTEN.IUP:	*application test file (user interface);*
TEST.BLB:	*application test file (algorithm).*

For pCO, pCO²/LonWorks® interface

PL10_001.NXE:	*Application file to be downloaded in the memory of the interface card using LonMaker™ or other network installing and managing tools;*
PL10_001.XIF:	*External Interface File, providing basic information regarding a node;*
PL10_001.REP:	*Report file.*

6.2 How to download the test program on pCO²

WinLOAD is the software that allows downloading the software to pCO².

6.2.1 How to install WinLOAD (Windows™ 95/98)

- *Be sure that that the executable file WINLOAD32.EXE and the application files (TESTEN.IUP e TEST.BLB) are in the same directory.*
- *Copy the SERIAL.VXD driver in the \WINDOWS\SYSTEM directory, making a backup of the preexistent SERIAL.VXD file .*
- *Restart Windows.*

6.2.2 How to use WinLOAD

Refer to the attached documentation WINLOAD32.DOC.

6.3 How to use pCO and pCO² test application

Press MENU button on the pCO/pCO² terminal to change the loop of masks on the display. You can choose among:

- *The loop of the information masks,*
- *The loop of the I/O masks,*
- *The loop of the configuration masks.*

When the cursor is in the top/left position of the display, press UP/DOWN to change the masks in the current loop.
When the cursor is NOT in the top/left position, press UP/DOWN to change the value of the current field.
Press ENTER to confirm the value and move the cursor.

6.4 Test program: exchange variables between pCO/pCO² and LonWorks® network

6.4.1 Standard variables

Type	Index pCO	Description	Name NV	Type NV	Direction
ANL	1	n.1 analog input value	anl_01_out	39 (SNVT_temp)	output
ANL	2	n.2 analog input value	anl_02_out	39 (SNVT_temp)	output
ANL	3	n.3 analog input value	anl_03_out	39 (SNVT_temp)	output
ANL	4	n.4 analog input value	anl_04_out	39 (SNVT_temp)	output
ANL	5	n.5 analog input value	anl_05_out	39 (SNVT_temp)	output
ANL	6	n.6 analog input value	anl_06_out	39 (SNVT_temp)	output
ANL	7	n.7 analog input value	anl_07_out	39 (SNVT_temp)	output
ANL	8	n.8 analog input value	anl_08_out	39 (SNVT_temp)	output
ANL	9	n.9 analog input value	anl_09_out	39 (SNVT_temp)	output
ANL	10	n.10 analog input value	anl_10_out	39 (SNVT_temp)	output
ANL	11	n.1 analog output value(0=0 Volt, 1000=10 Volt)	anl_11_in	8 (SNVT_count)	input
ANL	12	n.2 analog output value(0=0 Volt, 1000=10 Volt)	anl_12_in	8 (SNVT_count)	input
ANL	13	n.3 analog output value(0=0 Volt, 1000=10 Volt)	anl_13_in	8 (SNVT_count)	input
ANL	14	n.4 analog output value(0=0 Volt, 1000=10 Volt)	anl_14_in	8 (SNVT_count)	input
ANL	15	n.5 analog output value(0=0 Volt, 1000=10 Volt)	anl_15_in	8 (SNVT_count)	input
ANL	16	n.6 analog output value(0=0 Volt, 1000=10 Volt)	anl_16_in	8 (SNVT_count)	input
INT	1	Generic parameter n.1 (range -32768/+32767)	int_01_in	9 (SNVT_count_inc)	input
INT	2	Generic parameter n.2 (range -32768/+32767)	int_02_in	9 (SNVT_count_inc)	input
INT	3	Generic parameter n.3 (range -32768/+32767)	int_03_in	9 (SNVT_count_inc)	input
INT	4	Generic parameter n.4 (range -32768/+32767)	int_04_in	9 (SNVT_count_inc)	input
INT	5	Generic parameter n.5 (range -32768/+32767)	int_05_in	9 (SNVT_count_inc)	input
INT	6	Generic parameter n.6 (range -32768/+32767)	int_06_in	9 (SNVT_count_inc)	input
DGT	1	n. 1 digital input status (0: closed, 1: open)	dig_01_out	95 (SNVT_switch)	output
DGT	2	n. 2 digital input status (0: closed, 1: open)	dig_02_out	95 (SNVT_switch)	output
DGT	3	n. 3 digital input status (0: closed, 1: open)	dig_03_out	95 (SNVT_switch)	output
DGT	4	n. 4 digital input status (0: closed, 1: open)	dig_04_out	95 (SNVT_switch)	output
DGT	5	n. 5 digital input status (0: closed, 1: open)	dig_05_out	95 (SNVT_switch)	output
DGT	6	n. 6 digital input status (0: closed, 1: open)	dig_06_out	95 (SNVT_switch)	output
DGT	7	n. 7 digital input status (0: closed, 1: open)	dig_07_out	95 (SNVT_switch)	output
DGT	8	n. 8 digital input status (0: closed, 1: open)	dig_08_out	95 (SNVT_switch)	output
DGT	9	n. 9 digital input status (0: closed, 1: open)	dig_09_out	95 (SNVT_switch)	output
DGT	10	n. 10 digital input status (0: closed, 1: open)	dig_10_out	95 (SNVT_switch)	output
DGT	11	n. 11 digital input status (0: closed, 1: open)	dig_11_out	95 (SNVT_switch)	output
DGT	12	n. 12 digital input status (0: closed, 1: open)	dig_12_out	95 (SNVT_switch)	output
DGT	13	n. 13 digital input status (0: closed, 1: open)	dig_13_out	95 (SNVT_switch)	output
DGT	14	n. 14 digital input status (0: closed, 1: open)	dig_14_out	95 (SNVT_switch)	output
DGT	15	n. 15 digital input status (0: closed, 1: open)	dig_15_out	95 (SNVT_switch)	output
DGT	16	n. 16 digital input status (0: closed, 1: open)	dig_16_out	95 (SNVT_switch)	output
DGT	17	n. 17 digital input status (0: closed, 1: open)	dig_17_out	95 (SNVT_switch)	output
DGT	18	n. 18 digital input status (0: closed, 1: open)	dig_18_out	95 (SNVT_switch)	output
DGT	21	n. 1 digital output status (0: open/not energized, 1: closed/energized)	dig_21_in	95 (SNVT_switch)	input
DGT	22	n. 2 digital output status (0: open/not energized, 1: closed/energized)	dig_22_in	95 (SNVT_switch)	input
DGT	23	n. 3 digital output status (0: open/not energized, 1: closed/energized)	dig_23_in	95 (SNVT_switch)	input
DGT	24	n. 4 digital output status (0: open/not energized, 1: closed/energized)	dig_24_in	95 (SNVT_switch)	input
DGT	25	n. 5 digital output status (0: open/not energized, 1: closed/energized)	dig_25_in	95 (SNVT_switch)	input
DGT	26	n. 6 digital output status (0: open/not energized, 1:	dig_26_in	95 (SNVT_switch)	input

		closed/energized)			
DGT	*27*	*n. 7 digital output status (0: open/not energized, 1: closed/energized)*	*dig_27_in*	*95 (SNVT_switch)*	*input*
DGT	*28*	*n. 8 digital output status (0: open/not energized, 1: closed/energized)*	*dig_28_in*	*95 (SNVT_switch)*	*input*
DGT	*29*	*n. 9 digital output status (0: open/not energized, 1: closed/energized)*	*dig_29_in*	*95 (SNVT_switch)*	*input*
DGT	*30*	*n. 10 digital output status (0: open/not energized, 1: closed/energized)*	*dig_30_in*	*95 (SNVT_switch)*	*input*
DGT	*31*	*n. 11 digital output status (0: open/not energized, 1: closed/energized)*	*dig_31_in*	*95 (SNVT_switch)*	*input*
DGT	*32*	*n. 12 digital output status (0: open/not energized, 1: closed/energized)*	*dig_32_in*	*95 (SNVT_switch)*	*input*
DGT	*33*	*n. 13 digital output status (0: open/not energized, 1: closed/energized)*	*dig_33_in*	*95 (SNVT_switch)*	*input*
DGT	*34*	*n. 14 digital output status (0: open/not energized, 1: closed/energized)*	*dig_34_in*	*95 (SNVT_switch)*	*input*
DGT	*35*	*n. 15 digital output status (0: open/not energized, 1: closed/energized)*	*dig_35_in*	*95 (SNVT_switch)*	*input*
DGT	*36*	*n. 16 digital output status (0: open/not energized, 1: closed/energized)*	*dig_36_in*	*95 (SNVT_switch)*	*input*
DGT	*37*	*n. 17 digital output status (0: open/not energized, 1: closed/energized)*	*dig_37_in*	*95 (SNVT_switch)*	*input*
DGT	*38*	*n. 18 digital output status (0: open/not energized, 1: closed/energized)*	*dig_38_in*	*95 (SNVT_switch)*	*input*

Tab. 6.4.1.1

6.4.2 Non standard variables

For the description of the non-standard variables see ***Description of the non standard variables****.*

```
output struct {
        unsigned long int       sfw_idnt;
        unsigned char           pco_addr;
        unsigned char           xif_stat;
        unsigned char           xif_flag;
} xif_data;     // interface info

output struct {
        unsigned char           stat;
        char                    type;
        unsigned char           indx;
        signed long int         data;
} rd_prmt;      // pCO/pCO² variables reading

input struct {
        unsigned char           cmnd;
        char                    type;
        unsigned char           indx;
        signed long int         data;
} wr_cmnd;    // pCO/pCO² variables writing
```

7. Technical specifications

Power	from pCO/pCO²
Operating conditions	0T55°C; 20÷80 %UR not condensing
Storage conditions	-20T70°C; 20÷80 %UR not condensing
Environmental pollution	normal
Dimensions (mm):	PCO20L**** 60x30x20, PCOSER*** 47x44x21

Tab. 7.1

8. APPENDIX A

8.1 SNVT variables list handled by the interface

The variables marked with an X in the 16-bit range column are NOT handled by the interface.

SNVT number	SNVT name	16bit range	resolution unit	SNVT range	available range
1	SNVT_amp	-32768 .. 32767	0.1A	-3276.8 .. 3276.7A	-3276.8 .. 3276.7A
2	SNVT_amp_mil	-32768 .. 32767	0.1mA	-3276.8 .. 3276.7mA	-3276.8 .. 3276.7mA
3	SNVT_angle	0 .. 65535	0.00rad	0.000 .. 65.535rad	0.000 .. 32.767rad
4	SNVT_angle_vel	-32768 .. 32767	0.rad/s	-3276.8 .. 3276.7rad/s	-3276.8 .. 3276.7rad/s
5	SNVT_btu_kilo	0 .. 65535	1kBTU	0 .. 65535kBTU	0 .. 32767kBTU
6	SNVT_btu_mega	0 .. 65535	1MBTU	0 .. 65535MBTU	0 .. 32767MBTU
7	SNVT_char_ascii	0 .. 255	1	0 .. 255	0 .. 255
8	SNVT_count	0 .. 65535	1	0 .. 65535	0 .. 32767
9	SNVT_count_inc	-32768 .. 32767	1	-32768 .. 32767	-32768 .. 32767
10	SNVT_date_cal	X			
11	SNVT_date_day	X			
12	SNVT_date_time	X			
13	SNVT_elec_kwh	0 .. 65535	1kWH	0 .. 65535kWH	0 .. 3276kWH
14	SNVT_elec_whr	0 .. 65535	0.1WH	0.0 .. 6553.5WH	0.0 .. 3276.7WH
15	SNVT_flow	0 .. 65534	1l/s	0 .. 65534l/s	0 .. 32767l/s
16	SNVT_flow_mil	0 .. 65535	1ml/s	0 .. 65535ml/s	0 .. 32767ml/s
17	SNVT_length	0 .. 65535	0.1m	0.0 .. 6553.5m	0.0 .. 3276.7m
18	SNVT_length_kilo	0 .. 65535	0.1km	0.0 .. 6553.5km	0.0 .. 3276.7km
19	SNVT_length_micr	0 .. 65535	0.1um	0.0 .. 6553.5um	0.0 .. 3276.7um
20	SNVT_length_mil	0 .. 65535	0.1mm	0.0 .. 6553.5 mm	0.0 .. 3276.7mm
21	SNVT_lev_cont	0 .. 200	0.5%	0.0 .. 100.0%	0.0 .. 100.0%
22	SNVT_lev_disc	X			
23	SNVT_mass	0 .. 65535	0.1g	0.0 .. 6553.5g	0.0 .. 3276.7g
24	SNVT_mass_kilo	0 .. 65535	0.1kg	0.0 .. 6553.5kg	0.0 .. 3276.7kg
25	SNVT_mass_mega	0 .. 65535	0.1ton	0.0 .. 6553.5ton	0.0 .. 3276.7ton
26	SNVT_mass_mil	0 .. 65535	0.1mg	0.0 .. 6553.5mg	0.0 .. 3276.7mg
27	SNVT_power	0 .. 65535	0.1W	0.0 .. 6553.5W	0.0 .. 3276.7W
28	SNVT_power_kilo	0 .. 65535	0.1kW	0.0 .. 6553.5kW	0.0 .. 3276.7kW
29	SNVT_ppm	0 .. 65535	1ppm	0 .. 65535ppm	0 .. 32767ppm
30	SNVT_press	-32768 .. 32767	0.1kPa	-3276.8 .. 3276.7kPa	-3276.8 .. 3276.7kPa
31	SNVT_res	0 .. 65535	0.1ohm	0.0 .. 6553.5ohm	0.0 .. 3276.7ohm
32	SNVT_res_kilo	0 .. 65535	0.1kohm	0.0 .. 6553.5kohm	0.0 .. 3276.7kohm
33	SNVT_sound_db	-32768 .. 32767	0.01dB	-327.68 .. 327.67dB	-327.68 .. 327.67dB
34	SNVT_speed	0 .. 65535	0.1m/s	0.0 .. 6553.5m/s	0.0 .. 3276.7m/s
35	SNVT_speed_mil	0 .. 65535	0.001m/s	0.000 .. 65.535m/s	0.000 .. 32767m/s
36	SNVT_str_asc	X			
37	SNVT_str_int	X			
38	SNVT_telcom	X			
39	SNVT_temp	0 .. 65535	0.1°C	-274.0 .. 6279.5°C Note[1]	-274.0 .. 3276.7°C

[1] SNVT_temp represents the tenths of Celsius degrees beyond -274°C.

40	SNVT_time_passed	X			
41	SNVT_vol	0 .. 65535	0.1l	0.0 .. 6553.5l	0.0 .. 3276.7l
42	SNVT_vol_kilo	0 .. 65535	0.1kl	0.0 .. 6553.5kl	0.0 .. 3276.7kl
43	SNVT_vol_mil	0 .. 65535	0.1ml	0.0 .. 6553.5ml	0.0 .. 3276.7ml
44	SNVT_volt	-32768 .. 32767	0.1V	-3276.8 .. 3276.7V	-3276.8 .. 3276.7V
45	SNVT_volt_dbmv	-32768 .. 32767	0.1dB uV	-327.68 .. 327.67dB uV	-327.68 .. 327.67 dBuV
46	SNVT_volt_kilo	-32768 .. 32767	0.1kV	-3276.8 .. 3276.7kV	-3276.8 .. 3276.7kV
47	SNVT_volt_mil	-32768 .. 32767	0.1mV	-3276.8 .. 3276.7mV	-3276.8 .. 3276.7mV
48	SNVT_amp_f	X			
49	SNVT_angle_f	X			
50	SNVT_angle_vel_f	X			
51	SNVT_count_f	X			
52	SNVT_count_inc_f	X			
53	SNVT_flow_f	X			
54	SNVT_length_f	X			
55	SNVT_lev_cont_f	X			
56	SNVT_mass_f	X			
57	SNVT_power_f	X			
58	SNVT_ppm_f	X			
59	SNVT_press_f	X			
60	SNVT_res_f	X			
61	SNVT_sound_db_f	X			
62	SNVT_speed_f	X			
63	SNVT_temp_f	X			
64	SNVT_time_f	X			
65	SNVT_vol_f	X			
66	SNVT_volt_f	X			
67	SNVT_btu_f	X			
68	SNVT_elec_whr_f	X			
69	SNVT_config_src	X			
70	SNVT_color	X			
71	SNVT_grammage	0 .. 65535	0.1gsm	0.0 .. 6553.5gsm	0.0 .. 3276.7gsm
72	SNVT_grammage_f	X			
73	SNVT_file_req	X			
74	SNVT_file_status	X			
75	SNVT_freq_f	X			
76	SNVT_freq_hz	0 .. 65535	0.1Hz	0.0 .. 6553.5Hz	0.0 .. 3276.7Hz
77	SNVT_freq_kilohz	0 .. 65535	0.1kHz	0.0 .. 6553.5kHz	0.0 .. 3276.7kHz
78	SNVT_freq_milhz	0 .. 65535	0.1MHz	0.0 .. 6553.5MHz	0.0 .. 3276.7MHz
79	SNVT_lux	0 .. 65535	1lux	0 .. 65535lux	0 .. 32767lux
80	SNVT_ISO_7811	X			
81	SNVT_lev_percent	-32768 .. 32766	0.005%	-163.840 .. 163.830%	-163.840 .. 163.830%
82	SNVT_multiplier	0 .. 65535	0.0005	0.0000 .. 32.7675	0.0000 .. 16.3835
83	SNVT_state	X			
84	SNVT_time_stamp	X			
85	SNVT_zerospan	X			
86	SNVT_magcard	X			
87	SNVT_elapsed_tm	X			
88	SNVT_alarm	X			
89	SNVT_currency	X			
90	SNVT_file_pos	X			
91	SNVT_muldiv	X			
92	SNVT_obj_request	X			
93	SNVT_obj_status	X			
94	SNVT_preset	X			
95	SNVT_switch	0 .. 1		Note[2]	Note[3]
96	SNVT_trans_table	X			
97	SNVT_override	X			

[2] SNVT_switch only the state field is used.
[3] SNVT_switch only the state field is used.

98	*SNVT_pwr_fact*	*-20000 .. 20000*	*0.00005*	*-1.00000 .. 1.00000*	*-1.00000 .. 1.00000*
99	*SNVT_pwr_fact_f*	*X*			
100	*SNVT_density*	*0 .. 65534*	*0.5kg/m3*	*0 .. 32767.5kg/m3*	*0 .. 1638.3kg/m3*
101	*SNVT_density_f*	*X*			
102	*SNVT_rpm*	*0 .. 65534*	*1rpm*	*0 .. 65534rpm*	*0 .. 32767rpm*
103	*SNVT_hvac_emerg*	*X*			
104	*SNVT_angle_deg*	*-32768 .. 32766*	*0.02deg*	*-359.98 .. 360.00deg*	*-359.98 .. 360.00deg*
105	*SNVT_temp_p*	*-27317 .. 32766*	*0.01°C*	*-273.17 .. 327.66°C*	*-273.17 .. 327.66°C*
106	*SNVT_temp_setpt*	*X*			
107	*SNVT_time_sec*	*0 .. 65534*	*0.1 s*	*0.0 .. 6553.4s*	*0.0 .. 3276.7s*
108	*SNVT_hvac_mode*	*X*			
109	*SNVT_occupancy*	*X*			
110	*SNVT_area*	*0 .. 65534*	*200mm^2*	*0 .. 13.1068m^2*	*0 .. 32767m^2*
111	*SNVT_hvac_overid*	*X*			
112	*SNVT_hvac_status*	*X*			
113	*SNVT_press_p*	*-32768 .. 32766*	*1Pa*	*-32768 .. 32766Pa*	*-32768 .. 32766Pa*

Tab. 8.1.1

Carel reserves the right to modify or change its products without prior notice.

CAREL

Technology & Evolution

CAREL srl
Via dell'Industria, 11 - 35020 Brugine - Padova (Italy)
Tel. (+39) 049.9716611 Fax (+39) 049.9716600
http://www.carel.com - e-mail: carel@carel.com

Agenzia / *Agency:*

Cod: +030221960 rel. 1.0 – 28/11/00