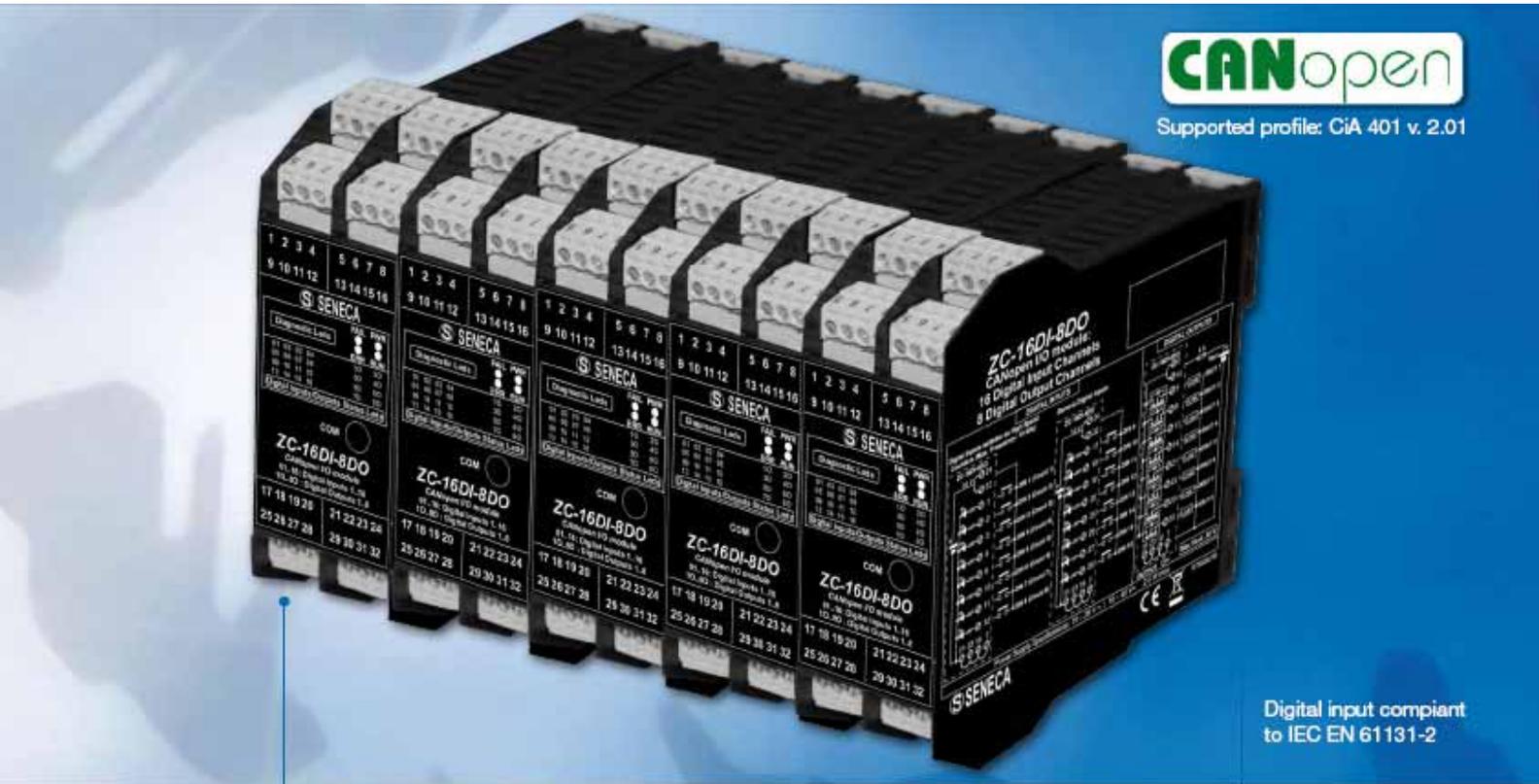


Z-PC LINE CANOPEN I/O SYSTEM

TECHNICAL OVERVIEW



1 IEC 61131 programming system

2 Very high accuracy: 0,01..0,05%

3 Isolation: 1,5 kVac (up tp 6 way)

4 Baud rate: up to 1 Mbps

5 Fast response time for digital channel ~ 1 ms

6 Fast response time for analog channel ~ 20 ms

7 Up to 8 powered sensors

8 Optimized distributed system (without multiple couples / controllers)

Index

1	The Z-PC CANopen System	3
1.1	Introduction	3
2	Basics on CANopen Networking	3
2.1	Introduction	3
2.2	Standards	4
2.3	PDO/SDO Messages	4
2.4	The Object Dictionary	4
2.5	PDO Mapping	4
2.6	PDO Linking	5
3	Technical Specifications	6
3.1	CANopen MASTER Stations	6
3.2	CANopen SLAVE Stations	6
3.2.1	SOFTWARE FEATURES	6
3.2.2	INPUT/OUPUT	7
3.3	CANopen ZC-107FO Repeater	7
4	CONNECTING THE Z-PC CANopen SERIES TO THE SENECA BUS	8
4.1	Bus Topology	8
4.2	Communication Speed (Baudrate) and BUS Length	8
4.3	Bus power supply	8
4.4	Connection of Z-PC-SLAVE CANopen and ZC-107FO stations to the SENECA BUS via the back IDC10 connector	9
4.5	Connection of the CANopen Master TWS3/TWS64 Station to the SENECA BUS via the back IDC10 connector	9
4.6	Connection of CANopen Master TWS3/ TWS64 station to theSENECA BUS via Side Clamp	10
4.7	Terminations Enable and Verification on SENECA BUS	11
5	CONNECTION OF Z-PC CANopen SERIES to a GENERIC BUS	12
5.1	Bus Topology	12
5.2	Communication Speed (Baudrate) and BUS Length	12
5.3	Connection of Z-PC-SLAVE CANopen and ZC-107FO stations to the SENECA BUS via the back IDC10 connector	12
5.4	Connection of TWS3/TWS64 stations to a generic CANopen BUS via the back IDC10 connector	13
6	SETTING THE DIP SWITCHES OF Z-PC STATIONS	14
6.1	Setting the station address by dip switches on CANopen slave devices	14
6.2	Setting the communication speed of the CANopen slave stations	14
7	EXTENDING A CANopen BUS with the ZC-107FO signal repeater	16
7.1	Setting the Buad rate BUS on ZC-107FO CANopen repeater	17
8	HOW IT WORKS	18
8.1	Start procedure	18
8.2	Default Set Connections	18
8.3	Node Guarding	19
8.4	Life Guarding	19
9	QUICK START: CANopen with CODESYS 2.3	20
9.1	Using the CodeSys CANopen integrated configurator	20
9.1.1	Loading of EDS files	20
9.1.2	Configuration of the TWS Master CANopen parameters	22
9.1.3	Parameters Settings of a ZC Slave	24
9.1.4	The CANopen library for CodeSys	27
9.1.5	EXOR_CAN_NMT Function	27
9.1.6	EXOR_CAN_SDO_RD Function	28
9.1.7	EXOR_CAN_SDO_WR Function	28
10	Quick Start: CANopen with ISAGRAF	30
10.1	Isagraf Library	30
10.2	IO Boards canocfg2	30
11	Appendix	32
11.1	EDS Files	32

1 The Z-PC CANopen System

1.1 Introduction

The Z-PC CANopen system allows the complete management of CANopen field bus.

It consist of:

- NR 2 CANopen master stations developed according to standard CiA DS-301 v4.02 (TWS3 and TWS64).
- NR 8 CANopen slave stations developed according to standard CiA DS-301 v4.02 and DS-401 v2.0 (ZC-24DI, ZC-16DI8DO, ZC-24DO, ZC-8TC, ZC-4RTD, ZC-8AI, ZC-3AO, ZC-SG).
- BUS SENECA that allows easy installation of CANopen SENECA stations, equipped with internal termination resistors.
- ZC-107FO repeater signal CANopen-based fiber optics

Thanks to Z-PC CANopen the station address and the baud rate is selectable through the configurator or by dip switches, making it even easier installation of the network. The stations performance combined with the baud rate up to 1Mbit / s leading the Z-CANopen PC at the top of the category

2 Basics on CANOpen Networking

2.1 Introduction

The CAN protocol (Controller Area Network) was developed in the mid-80s for applications related to motor vehicles from the German Robert Bosch. It describes the communication protocol at the physical layer and data layer (levels 1 and 2 of the OSI model). Not anything specific in relation to higher levels and in particular with regard to the Protocol relating to the application level (level 7 of the OSI model). The CANopen protocol was standardized by the International CiA and is a protocol-level application that is based on CAN protocol with respect to the lower levels. It defines what data and services must be transmitted and the significance of data for different categories of devices. CANopen Protocol is a standard application specific and is defined by the CIA DS301. The network management services defined in CANopen allow a simple initialization of the network. CAN is a communication system multi master. Unlike other bus systems, the connected modules are not identified, so are the messages sent on the bus. Participants in the network are authorized to send messages whenever the bus is free. Conflicts are resolved on the bus through a priority linked to the messages. CAN always sends broadcast messages that are divided into different levels of priority. All participants to the network have the same rights, then the communication is possible even without a master on the bus. The sending of data is decided independently by each station, but the data can be obtained from a remote station using a special message called "remote frame".

The CANopen specific (DS 301) indicates the technical and functional characteristics that each device must meet to be plugged into the network. CANopen The concepts underlying the CANopen are:

- The bus can contain up to 127 stations
- The device description is made by using a text file with .EDS extension. That file is supplied by the device manufacturer and is used to configure and use the device
- Communication is object-oriented through the use of messages PDO and SDO
- Complex services or low-priority messages are transmitted via SDO

- Data can be sent via PDO messages from all slaves on event or in response to the synchronization message.

2.2 Standards

The CiA DS-301 profile communication specifies mechanisms for configuration and communication between devices in real time environments. CANOpen is used for transmission at Level 2 specification ISO 11898 and CAN 2.0 A + B. The CANopen device TWS is based on communication profile CiA draft CANOpen 301 Standard Version 4.01. The ZC SLAVE CANopen devices meet the profile for modules I / O CiA 401 Draft Standard Version 1.4.

2.3 PDO/SDO Messages

CANopen implements communications services differ depending on the different types of communications objects transmitted. The two basic types of objects are the Process Data Object (PDO) and Service Data Objects (SDO). The PDO contains information such as real-time message identified with high priority. The maximum number of data contained into a PDO is 8 Bytes. The SDO contains system settings with low priority identifiers and can send, thanks to the automatic message fragmentation, even large moles of data. The exchange of data can be controlled by events or by a synchronization message(SYNC). Sending data on events greatly reduces the load of the bus, **allowing a high-performance communication even with a reduced speed of the bus**. It's also possible to use the system with both modes simultaneously active. The Service Data Objects are transmitted using point to point mode. In addition to the SDO and PDO, CANopen offers other communication objects:

- Communication Objects for synchronize inputs and outputs
- Communication Objects for boot-up procedures (starting)
- Communication Objects for the "life guarding / node guarding"
- Communication Objects for emergency messages

2.4 The Object Dictionary

The Object dictionary of a device gives the items required for the configuration of the device. Access to the dictionary objects (read or write of parameters) is performed by SDO serices. The dictionary object is composed of several parts:

- Features that apply to all CANopen devices (DS 301)
- Features that are valid for Input / Output devices
- Features that are manufacturer-dependent

The index for access to objects is standardized within the CANopen profiles, except for items defined by the manufacturer. The communication profile supports the objects 0x1000 and 0x1018. There are also objects for the configuration of PDO communication (index 0x1400, 0x1600).

With regard to the profile DS401 items are placed in the 0x6000 area.

2.5 PDO Mapping

The mapping of PDO allows the customization of data to send / receive in order to optimize the employment of network bandwidth. Each PDO can contain up to 8 bytes of data. The PDO is divided into TPDO and RPDO. The TPDO are related to the transmission of output data from the station, while the PDO RPDO containing data transmitted to the station.

The types of possible transmission type for the PDO are illustrated in the following table:

Type Nr	Cyclic	Acyclic	Synchronous	Asynchronous	RTR Only
0		X	X		
1-240	X		X		
241-251			Reserved		
252			X		
253				X	X
254				X	X
255				X	

The type from 1 to 240 is the number of SYNC objects between two PDO transmissions PDO.

The type 252 updates values on SYNC reception but are not send. The type 253 updates values on RTR reception.

The type 254 provides an application-specific device.

The type 255 provides an application defined in the profile of the device.

2.6 PDO Linking

The Z-PC CANopen stations support the PDO Linking, this means that you can direct a TPDO from a CANopen slave station into a RPDO of another slave station without increase the master station CPU load.

3 Technical Specifications

3.1 CANopen MASTER Stations

The CANopen devices TWS3 and TWS64 have the capabilities of master CANopen and implement the following features:

CAN

- Managing CAN 2.0A network (that is, with a 11 bit identifier).
- Transmission rates supported: 10, 20, 50, 100, 125, 250, 500, 800.1000 Kbits/s)

CANopen

- Standard DS 301 V4.0
- Supported profile DS 401 IO modules (digital and analog)

NMT MASTER

- Single Master
- Management of 127 stations (from 1 to 127)
- Configuring stations through SDO messages
- NMT Start of single stations
- Monitoring through Node guarding
- Receiving emergency messages EMYC
- Generation of messages sync SYNC
- Master Heartbeat

Input / Output

- Management of synchronous and asynchronous transmission
- Auto Grouping of inputs and outputs on PLC memory
- 256 transmission PDO containing a maximum of 8 bytes (1024 bytes maximum output)
- 256 receive PDO containing a maximum of 8 bytes (1024 bytes maximum input)

CANopen Manager

- Read and import of EDS files
- Diagnostic data from devices
- Automatic generation of PDO messages

3.2 CANopen SLAVE Stations

The CANopen slave devices implement the following features:

3.3 SOFTWARE FEATURES

CAN

- CAN 2.0A network (that is, with a 11 bit identifier).
- Transmission rates supported: 20, 50, 125, 250, 500, 800.1000

CANOpen

- Standard DS 301 V4.0
- Supported profile DS 401 IO modules (digital and analog)

NMT SLAVE

- Address set (from 1 to 127) also from dip switches
- baudrate set also by dip switches
- Stations Configurations through SDO messages
- Node guarding
- Emergency messages EMYC

Input / Output Devices

- Send / Receive PDO synchronous and asynchronous
- Up to 5 PDO in transmission containing up to 8 bytes
- 1 PDO receive containing up to 8 bytes

3.4 INPUT/OUTPUT

Depending on station type they are equipped with the following inputs / outputs:

- up to 24 digital inputs
- up to 24 digital outputs
- up to 8 digital outputs and 16 digital inputs on the same station
- up to 8 thermocouple inputs
- up to 4 inputs RTD
- up to 8 inputs current / voltage DC
- up to 3 analog output current / voltage DC
- 1 strain gauge bridge input

3.5 CANopen ZC-107FO Repeater

The ZC-107FO is a CAN signal repeater through fiber optics. The repeater can also be used to expand the number of nodes within the same bus. By connecting two ZC-107FO with each other through the fiber optic the CAN communication can be extended to a maximum length of two kilometers for a maximum Baud rate of 1Mbit/s.

4 CONNECTING THE Z-PC CANopen SERIES TO THE SENECA BUS

4.1 Bus Topology

CAN is based on a linear topology type with a shielded cable with two wires and termination resistors on each end of the cable. The communication speed varies between 10 kbit / s (> 1000 m) and 1 Mbit / s (25 m), depending on the length of the network.

4.2 Communication Speed (Baudrate) and BUS Length

The maximum distance reached via the CANopen network depends on the speed of communication selected, the figure shows the different possibilities.

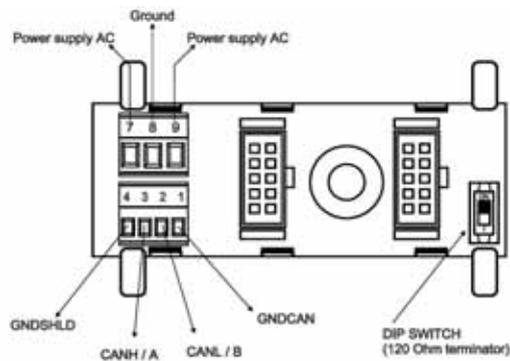
Velocità	10 kbit/s	20 kbit/s	50 kbit/s	125 kbit/s	250 kbit/s	500 kbit/s	800 kbit/s	1 Mbit/s
BUS Length	5000 m ⁽²⁾	2500 m ⁽²⁾	1000 m ⁽²⁾	500 m ⁽¹⁾	250 m ⁽¹⁾	100 m	50 m	25 m

⁽¹⁾ For distances over 200 meters is recommended the use of opto couplers.

⁽²⁾ For distances over 1,000 meters is recommended the use of repeater signal ZC-107FO.

4.3 The SENECA BUS

The Seneca bus has the following pinout:



For AC Supply:

Connect the AC supply to the Power Supply AC pins.

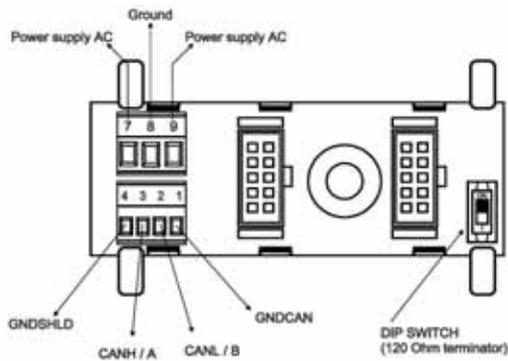
Pull pin 8 (Ground) to earth on the cabinet board.

For DC Supply:

Connect the DC supply to the Power Supply pins, it is not necessary to respect the supply polarity.

Pull pin 8 (Ground) to earth on the cabinet board.

4.4 Connection of Z-PC-SLAVE CANopen and ZC-107FO stations to the SENECA BUS via the back IDC10 connector.

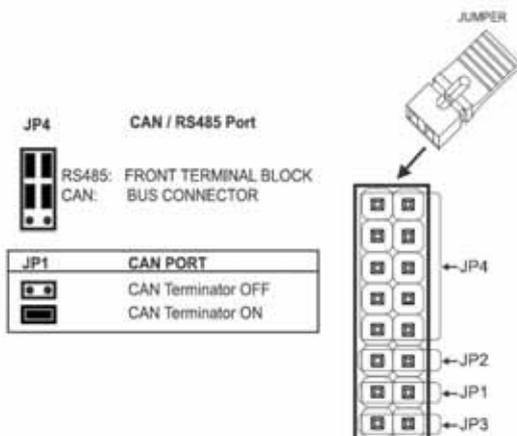


The connection to the bus of CANopen Slave station is done by inserting the back IDC10 plug in the bus Seneca. The pin 1 to 4 of the bus are used to extend the CANopen bus outside the SENECA bus, in this case the signal GNDCAN (pin 1) should be linked to obtain a robust communication. The pin 4 (GNDSHLD) should be connected to the bridging of the cable used to connect. If it is necessary to use a T connection from the main bus line, see the following table for their maximum length (from ISO / DIS 11898):

Speed	Max bus Length	Max T connection Length
20 kbit/s	1000 m	7,5 m
125 kbit/s	200 m	3,75 m
500 kbit/s	100 m	0,75 m
1000 Mbit/s	25 m	0,3 m

4.5 Connection of the CANopen Master TWS3/TWS64 Station to the SENECA BUS via the back IDC10 connector

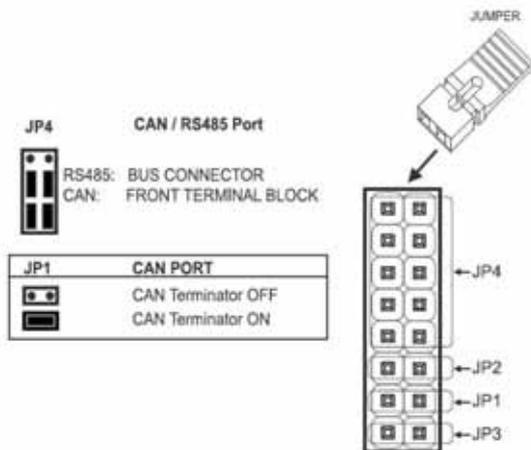
Set the jumpers as shown:



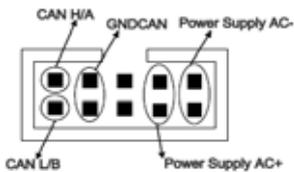
The connection is done by inserting the station back connector to the SENECA bus.

4.6 Connection of CANopen Master TWS3/ TWS64 station to the SENECA BUS via Side Clamp

Set the jumpers as shown:



The pinout of IDC10 connector is :



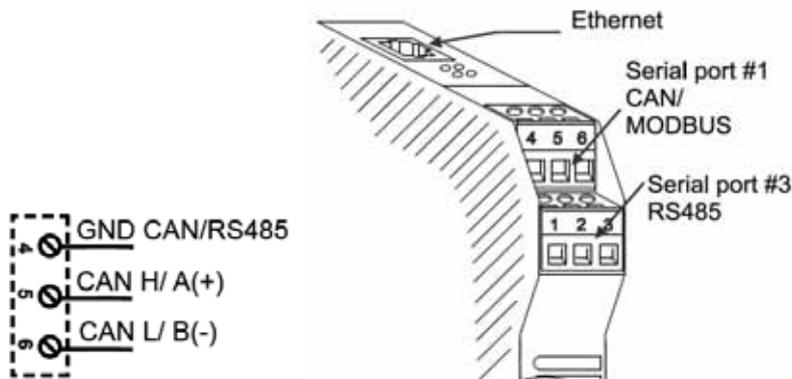
For AC Supply:

Connect the AC supply to the IDC10 connector Power Supply AC pins.

Per DC Supply:

Connect the DC supply to the IDC10 connector Power Supply pins it is not necessary to respect the supply polarity.

BUS Signals:

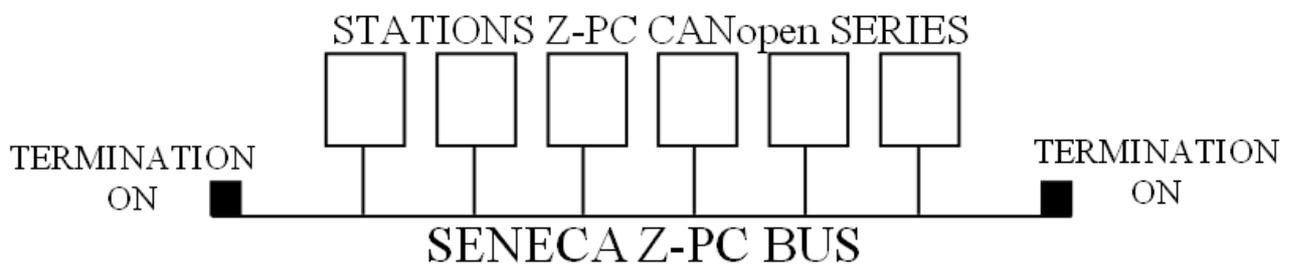


The pins CANH, CANL and GND CAN are used for CANopen bus signals, the GND CAN signal must be connected to obtain a robust communication. If it is necessary to use a T connection from the main bus line, see the following table for their maximum length (from ISO / DIS 11898):

Speed	Max BUS Length	Max T connection length
20 kbit/s	1000 m	7,5 m
125 kbit/s	200 m	3,75 m
500 kbit/s	100 m	0,75 m
1000 Mbit/s	25 m	0,3 m

4.7 Terminations Enable and Verification on SENECA BUS

The 120Ω termination between signals CAN_L and CAN_H is already included in the SENECA bus and must be enabled using the on board dip switch. The termination should be enabled both at the beginning (typically before the CANopen master) and end of the BUS (typically after the last CANopen slave station):



Note that a termination may be enabled directly to the TWS3/TWS64 stations through JP1 jumper.

After the enable operation, using a tester to verify the various impedances indicated:

Measure	Value	Meaning
Between GND and CAN_L	Infinity	OK
	0	CAUTION: Short circuit between GND and CAN_L
Between GND and CAN_H	Infinity	OK
	0	CAUTION: Short circuit between GND and CAN_H
Between CAN_L and CAN_H	About 60 Ω	OK Both terminations enabled
	About 120 Ω	CAUTION: Only one termination enabled
	< 50 Ω	CAUTION: More than two termination enabled

5 CONNECTION OF Z-PC CANopen SERIES to a GENERIC BUS

5.1 Bus Topology

CAN is based on a linear topology type with a shielded cable with two wires and termination resistors on each end of the cable. The communication speed varies between 10 kbit / s (> 1000 m) and 1 Mbit / s (25 m), depending on the length of the network.

5.2 Communication Speed (Baudrate) and BUS Length

The maximum distance reached via the CANopen network depends on the speed of communication selected, the figure shows the different possibilities.

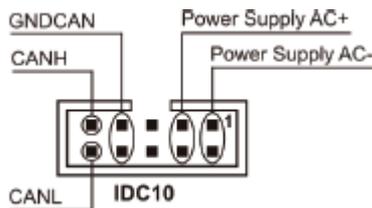
Velocità	10 kbit/s	20 kbit/s	50 kbit/s	125 kbit/s	250 kbit/s	500 kbit/s	800 kbit/s	1 Mbit/s
BUS Length	5000 m ⁽²⁾	2500 m ⁽²⁾	1000 m ⁽²⁾	500 m ⁽¹⁾	250 m ⁽¹⁾	100 m	50 m	25 m

⁽¹⁾ For distances over 200 meters is recommended the use of opto couplers.

⁽²⁾ For distances over 1,000 meters is recommended the use of repeater signal ZC-107FO.

5.3 Connection of Z-PC-SLAVE CANopen and ZC-107FO stations to the SENECA BUS via the back IDC10 connector.

The pinout of IDC10 connector is :



For AC Supply:

Connect the AC supply to the Power Supply AC pins.

For DC Supply:

Connect the DC supply to the Power Supply pins, it is not necessary to respect the supply polarity.

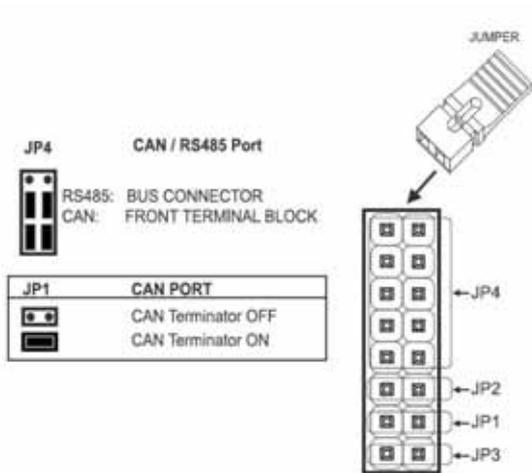
BUS Signals:

The pins CANH, CANL and GNDCAN are used for CANopen bus signals, the GNDCAN signal must be connected to obtain a robust communication. If it is necessary to use a T connection from the main bus line, see the following table for their maximum length (from ISO / DIS 11898):

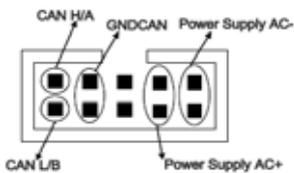
Speed	Max BUS Length	Max T connection length
20 kbit/s	1000 m	7,5 m
125 kbit/s	200 m	3,75 m
500 kbit/s	100 m	0,75 m
1000 Mbit/s	25 m	0,3 m

5.4 Connection of TWS3/TWS64 stations to a generic CANopen BUS via the back IDC10 connector

Set the jumpers as shown:



The pinout of IDC10 connector is :



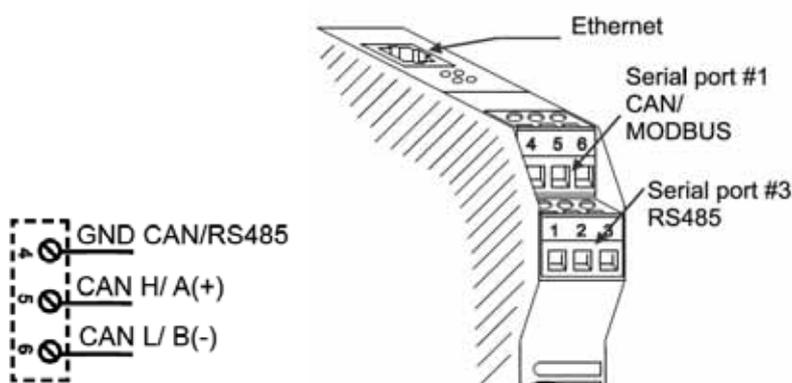
For AC Supply:

Connect the AC supply to the Power Supply AC pins.

For DC Supply:

Connect the DC supply to the Power Supply pins, it is not necessary to respect the supply polarity.

BUS Signals:



The pins CANH, CANL and GND/CAN are used for CANopen bus signals, the GND/CAN signal must be connected to obtain a robust communication. If it is necessary to use a T connection from the main bus line, see the following table for their maximum length (from ISO / DIS 11898):

Speed	Max BUS Length	Max T connection length
20 kbit/s	1000 m	7,5 m
125 kbit/s	200 m	3,75 m
500 kbit/s	100 m	0,75 m
1000 Mbit/s	25 m	0,3 m

6 SETTING THE DIP SWITCHES OF Z-PC STATIONS

6.1 Setting the station address by dip switches on CANopen slave devices

The Z-PC CANopen SLAVE devices have a series of DIP switches accessible from the hole on the container. The DIP switches are used for both the setting of the station number (ID) and for setting the speed of communication. The ID station is used to calculate the COB-ID of PDO, the SDO and Emergency objects. The binary weight of each DIP switch increases by the number of the switch, for example if the ID 1 is obtained by DIP4 = ON, the ID 8 is obtained by DIP8 = ON, etc.. The nodes can have values from 1 to 127. The COB ID allocation is made according to the Default Set Connection (CiA DS 301, 8.4.1).

ADDRESS		
4 5 6 7 8 9 10	0000000	SOFTWARE PROGRAMMED
	0000001	ADD. 001
	0000010	ADD. 002
	0000011	ADD. 003
	0000100	ADD. 004
	0000101	ADD. 005
.....
	1111111	ADD. 127

The combination of the dip from 4 to 10 in OFF (Software Programmed) lets you configure the address from a CANopen configurator by the SDO protocol.

6.2 Setting the communication speed of the CANopen slave stations

The communication speed setting is done through the first three DIP switches. The figure shows the pattern of allocation of speeds through the DIP switches.

BAUD RATE			
1	2	3	SOFTWARE PROGRAMMED
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	20 kbps
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	50 kbps
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	125 kbps
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	250 kbps
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	500 kbps
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	800 kbps
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1 Mbps

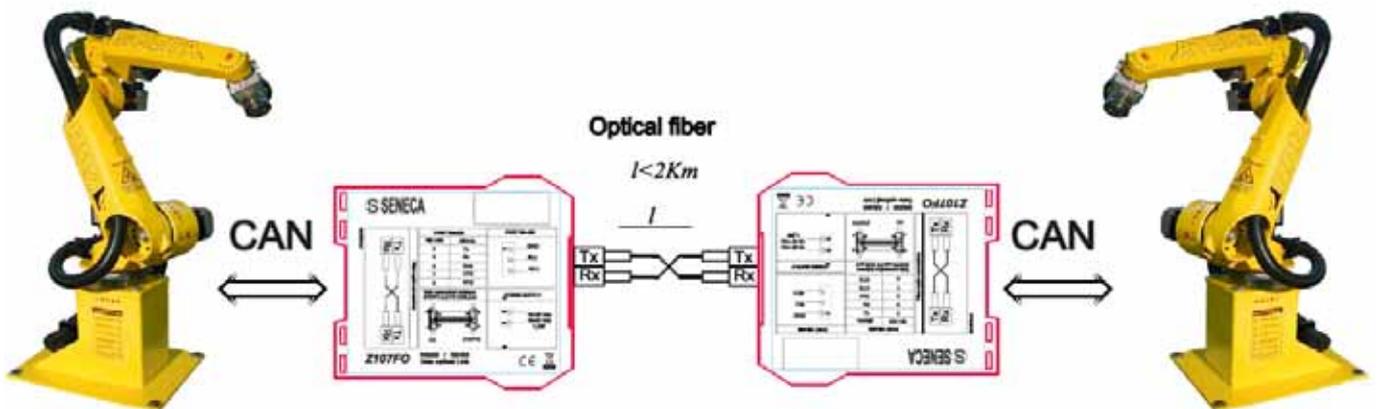
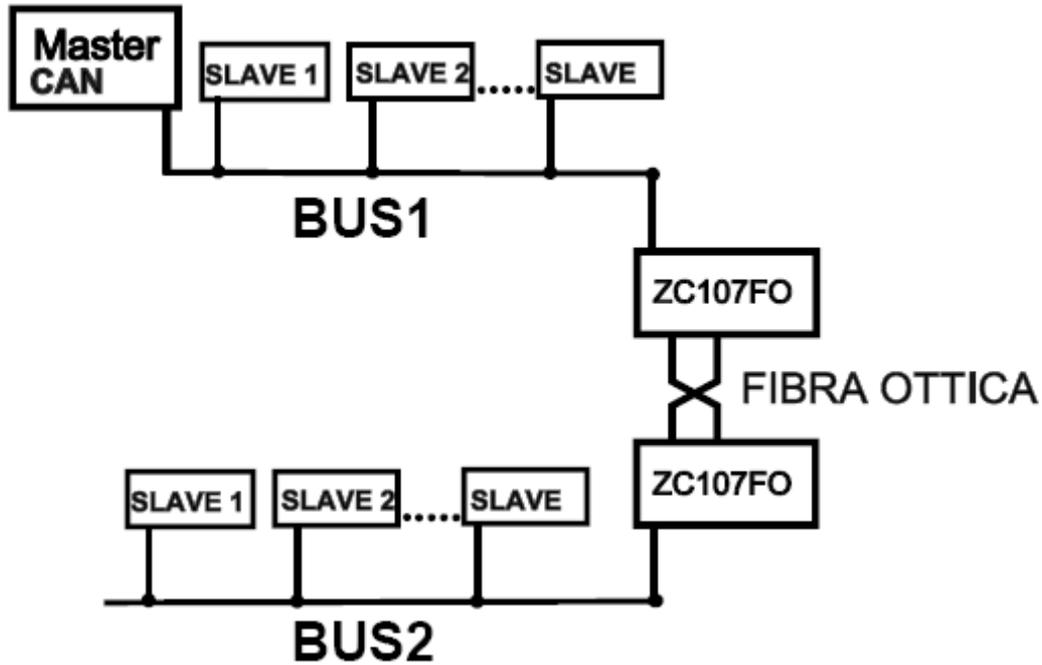
The figure is an example of a possible setting of DIP switches. The setting is the condition ID = 32 and communication speed = 500 kbits/s.

1	2	3	4	5	6	7	8	9	10
ON	OFF	ON	OFF	OFF	OFF	OFF	OFF	ON	OFF

The combination of dip 1 .. 3 to OFF (Software Programmed) lets you set the baud rate by using a CANopen configurator by the SDO protocol.

7 EXTENDING A CANopen BUS with the ZC-107FO signal repeater

A clear example of the use of the ZC-107FO repeater is:



7.1 Setting the Baud rate BUS on ZC-107FO CANopen repeater

7 8 9 10	Baud Rate	2	Terminator 120 ohm
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	10 kbps	<input type="checkbox"/>	Not Enable
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	20 kbps	<input type="checkbox"/>	Enable
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	50 kbps		
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	62,5 kbps		
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	100 kbps		
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	125 kbps		
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	250 kbps		
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	500 kbps		
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	800 kbps		
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	1000 kbps		

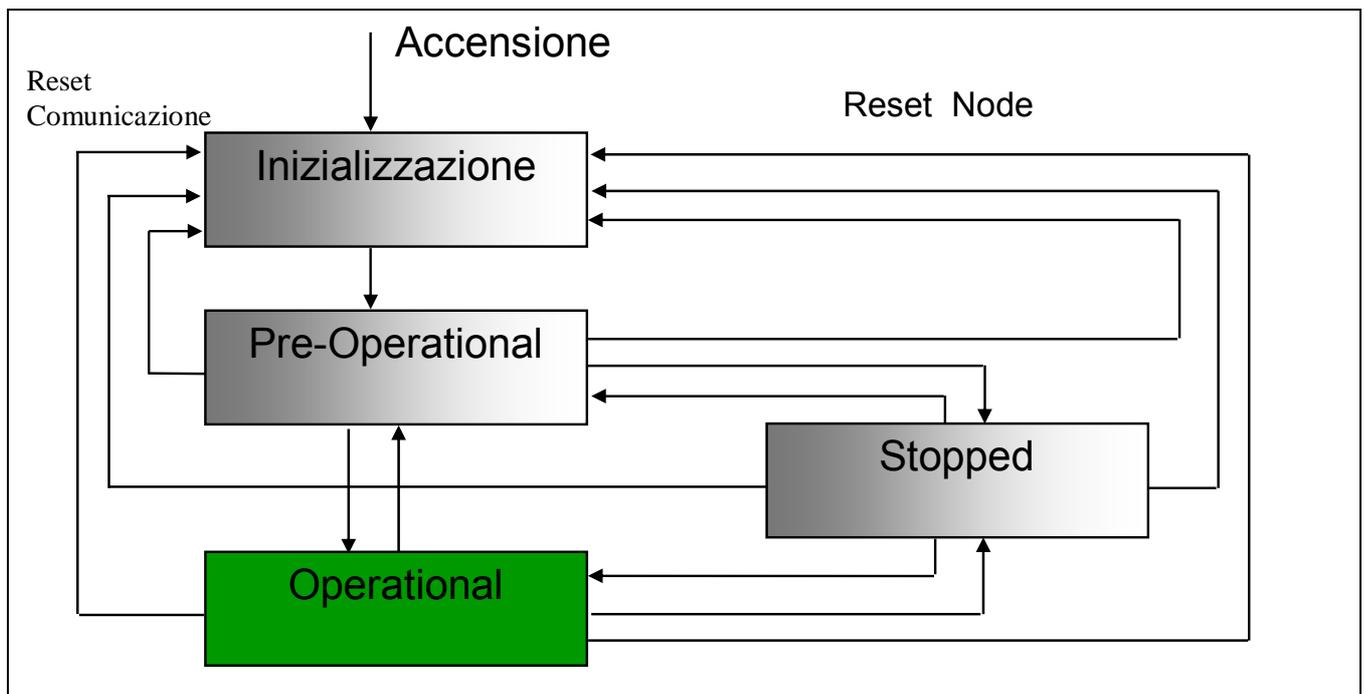
It is essential to set on the pair of ZC-107FO repeaters the same baud rate, the baud rate must coincide with the CANopen network that you want to extend. The baud rate is adjustable via dip switches 7 .. 10, you can also insert directly on the repeater the termination resistance.

8 HOW IT WORKS

The following chapter sets out the main aspects of the Z-PC CANopen SLAVE devices.

8.1 Start procedure

When a device is switched on is carried in the state **INITIALIZATION**, the application and communication objects are assigned at this stage. When this phase is successfully completed, the device is automatically carried in **PRE-OPERATIONAL** state.



When the device is in **PRE-OPERATIONAL** state, communication via SDO is initiated. In this state, you can run the following settings via SDO:

- Setting the Guard Time and Life Time Factors.
- Setting the communication parameters of PDOs.
- Mapping of PDOs.
- Saving information.

In **PRE-OPERATIONAL** state the device is unable to perform either the PDO communication or transmit emergency messages.

In the **OPERATIONAL** stage the device can automatically send messages PDO and emergency messages.

In the **STOPPED** state (also indicated with **PREPARED**) the communication on the bus is turned off (SDO and PDO communication not possible), the only accepted command is a network command (for example a Start Remote Node).

8.2 Default Set Connections

At start up by default the PDO reception is available, for example RPDO1 and RPDO2, with the COB-ID:

RPDO1 = 200h + Node ID

RPDO2 = 300h + Node ID

At start up by default the PDO transmission is available, for example TPDO1 and TPDO2, with the COB-ID:

TPDO1 = 180h + Node ID

TPDO2 = 280h + Node ID

Through the CANOpen network configurator you can change the default setting of the connections.

8.3 Node Guarding

The Node Guarding enables the network administrator (typically a CANopen master station) to verify if a slave station has a fault. To detect what slave is in fault, the master sends the message to the Guard ID (100Eh) of a slave, every "Node Guard Time" through an RTR message for each node. The slaves reply with a Guard message, which contains a toggle bit, and the slave status. This reply message is used by the CANopen master to update the status of all slaves nodes connected and properly functioning.

8.4 Life Guarding

While node guarding is used by the network administrator to detect if a slave station is in fault, the slave use the guarding messages to see if the master is present. This slave monitoring function is called life guarding. **To detect a broken cable and force the outputs to the condition of fault with the CANOpen, you must use both the node and life guarding.** To activate the life guarding the station master configures the Guard Time (item 100Ch) and the Life Time Factor (item 100Dh). If time monitoring resulting from Life Time = Life Time Factor * Guard Time [ms] expire before that the slave has received a telegram of guarding, the device sets the exits/inputs in fault mode and considers the communication with the master interrupted.

9 QUICK START: CANopen with CODESYS 2.3

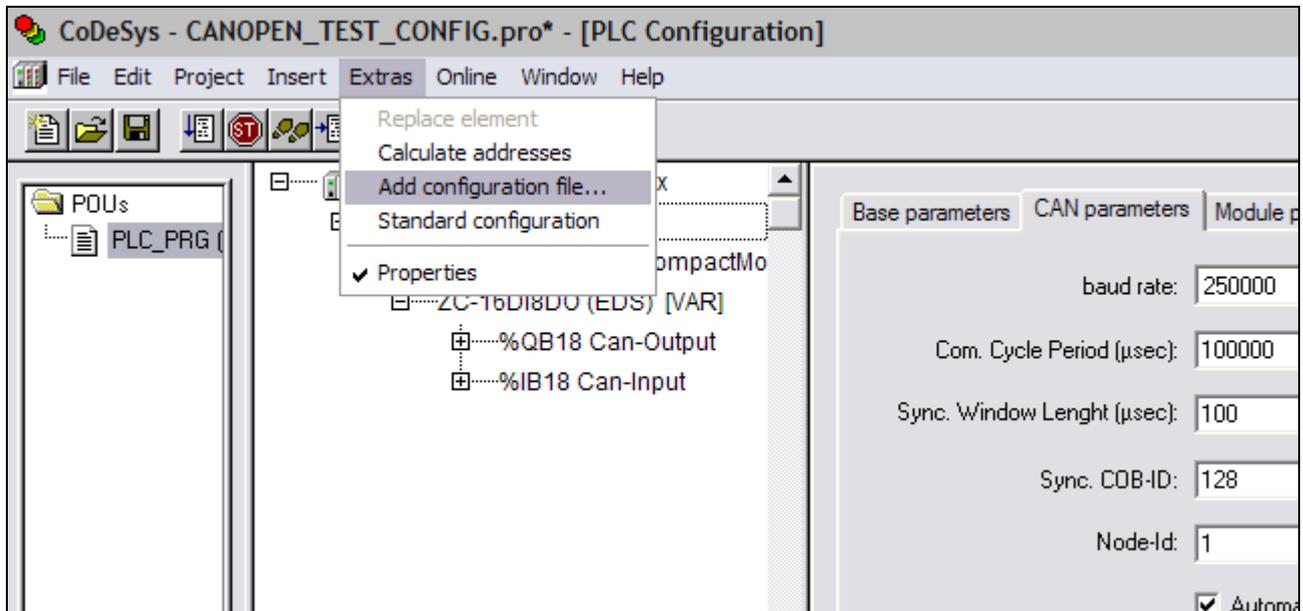
This chapter discusses the using and configuration process of the CANopen Master station into the CodeSys development environment used for the programming of PLC TWS3/TWS64. This environment uses an integrated CANOpen network configurator.

9.1 Using the CodeSys CANopen integrated configurator

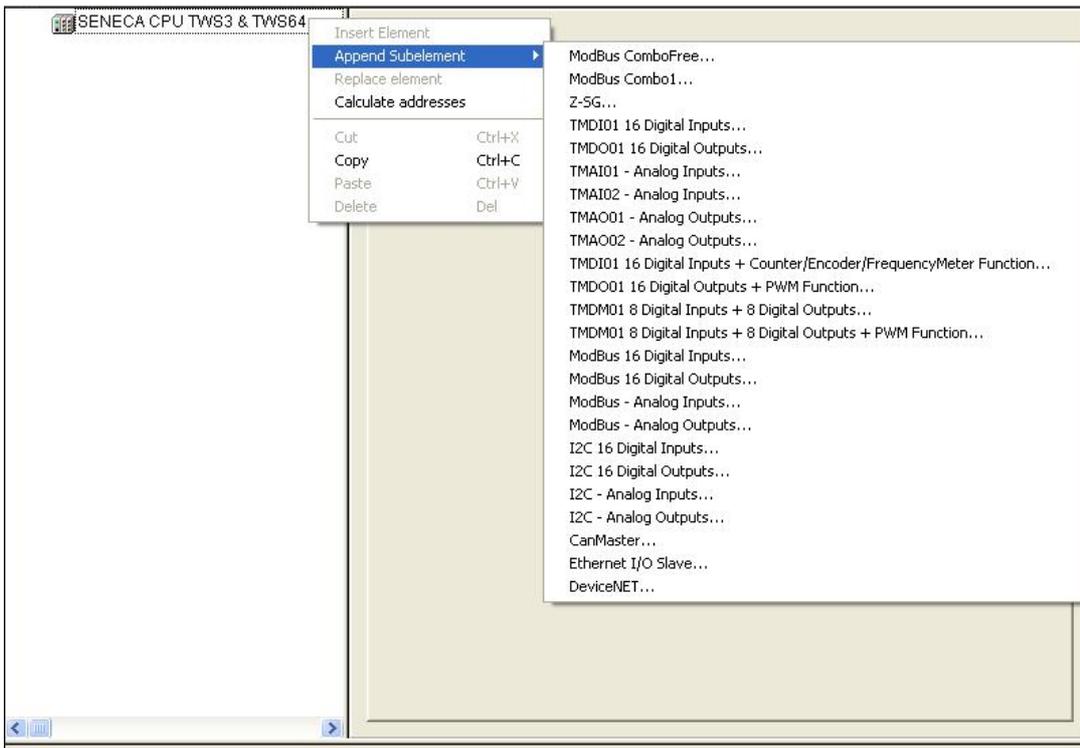
This section describes the use of the CodeSys CANopen integrated configurator to configure the network connected to the TWS3/TWS64 Master CANOpen. Please refer to the CodeSys software manuals for proper installation and a detailed description of its use.

9.2 Loading of EDS files

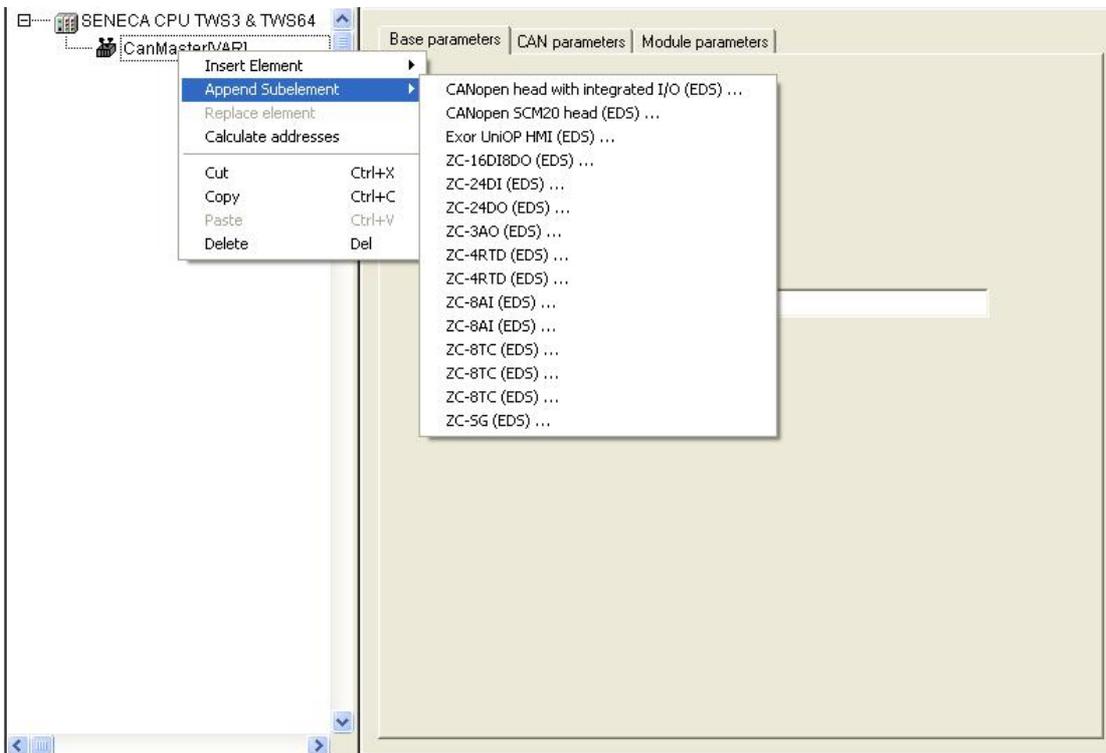
It's essential to use the menu *Extras | Add Configuration file* to install the EDS device file before inserting it into the CANOpen net:



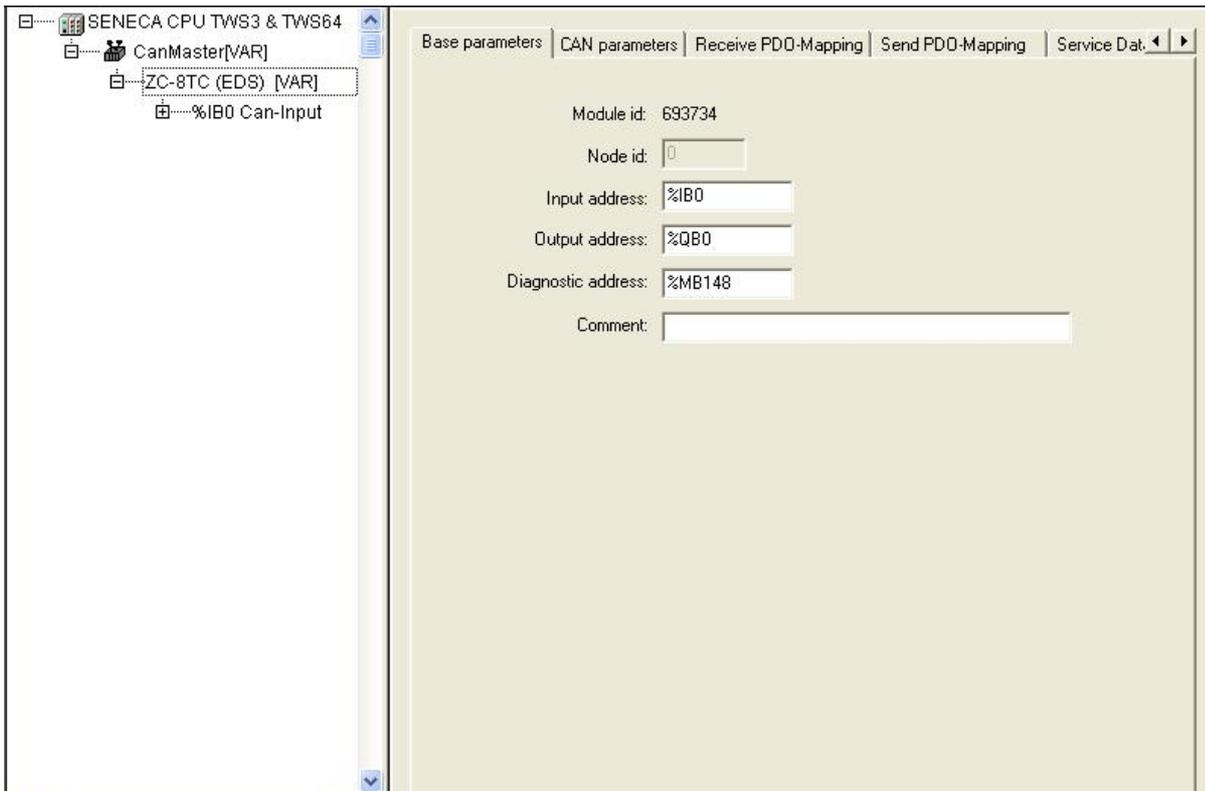
Once this operation was done by positioning itself in *Resources | PLC Configuration*, you can right click on the node "SENECA CPU TWS3 & TWS64" and add a "CanMaster":



Once inserted the CANMaster positioning on right click menu you can view the various slave available and include them in the configuration:

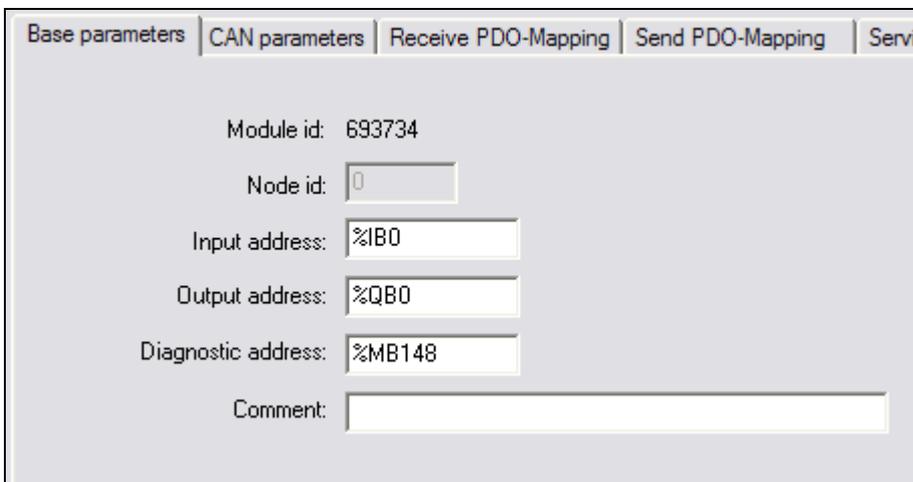


In the below figure is shown the result obtained by selecting the ZC-8TC station:



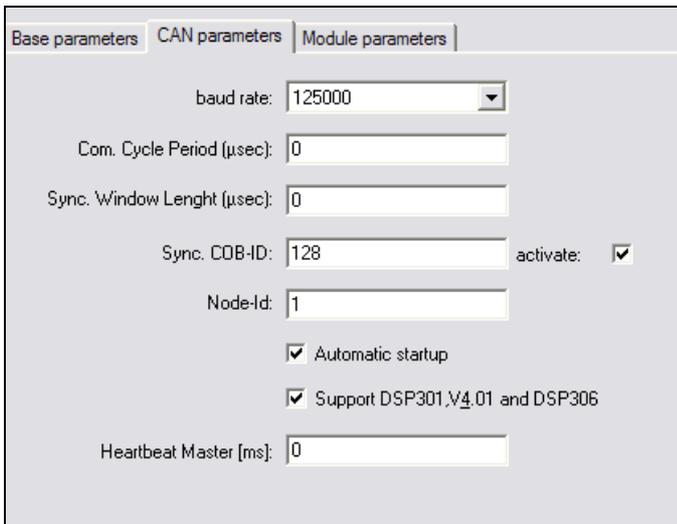
9.3 Configuration of the TWS Master CANOpen parameters

Positioning on CANMASTER[VAR], you can set its operating parameters.



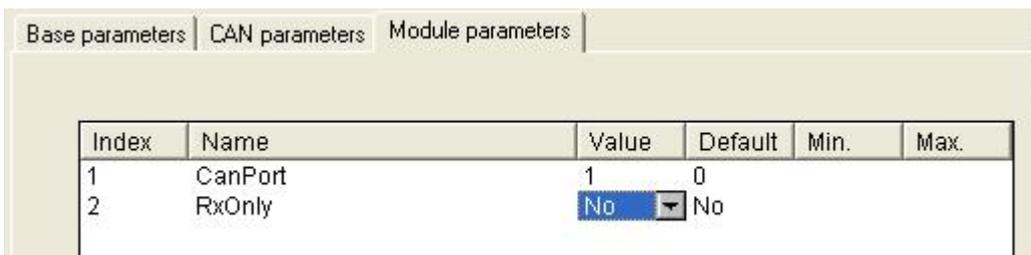
In the Basic Parameters window it's possible to insert the following parameters:

- Input Address: Shows the input memory location where the configurator starts to place the data of the Master CANOpen.
- Output Address: Shows the output memory location where the configurator starts to place the data of the Master CANOpen.
- Diagnostic address: indicates the location in memory where the CANopen Master diagnostics data are saved.



In the CAN parameters window you can configure the following parameters:

- Baud rate: Shows the speed of CANOpen network operations and must match the set of the DIP SWITCHES slaves stations
- Com Cycle Period: means the period between the release of two SYNC message
- Sync Window Length: temporal window within must be sent the synchronous PDOs
- Sync COB-ID: COB-ID assigned to the SYNC message sends from the master if you check the option "activate".
- Node-Id: The Node-id (1-127) is the node number which is used by the master for addressing the device in a CANOpen network.
- Automatic Startup: If this option is activated, at a download or at starting up the PLC the CAN bus will be initialized and started automatically: If the option is not activated, the CanDevice will wait for an appropriate command.
- Support DSP301, DSP306 and v4.01: active option extensions to the Protocol, in particular the functionality of Heartbeat Master (ways of monitoring the station presence where slave stations also act actively). When activated, you can indicate in the next fields the generation time of the Heartbeat message. It is recalled that the Heartbeat protocol is not supported by the Z-PC CANOpen slaves stations.



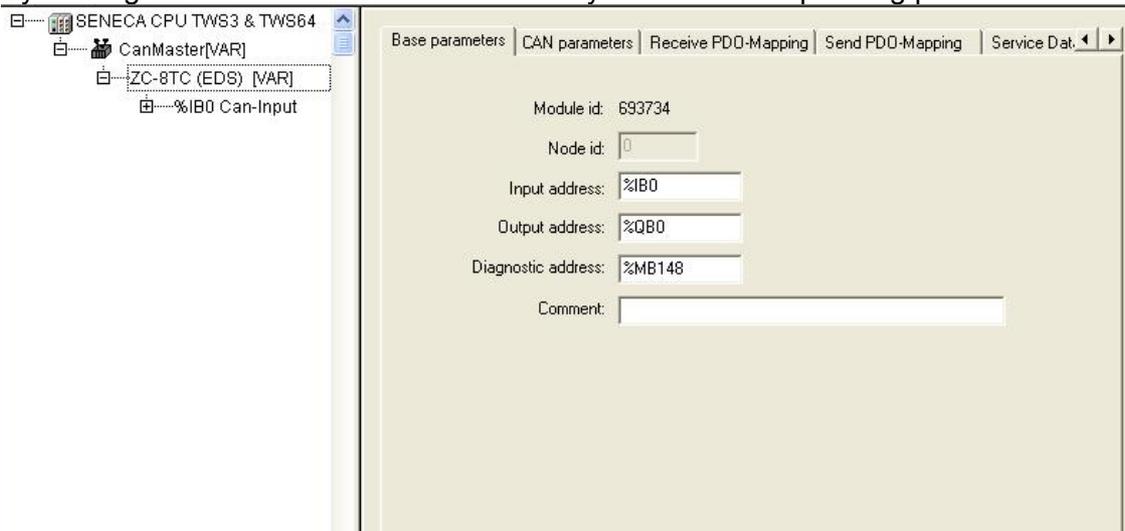
Index	Name	Value	Default	Min.	Max.
1	CanPort	1	0		
2	RxOnly	No	No		

In the MODULE parameters window you can configure the following parameters:

- CANPort: Shows the position of the CAN master port on TWS3/TWS64. Usually the first card has CanPort = 1, you can connect through TP-WIRE CAN additional ports on the same TWS3/TWS64 station.

9.3.1 Parameters Settings of a ZC Slave

By clicking on the inserted ZC Slave module you can set its operating parameters.



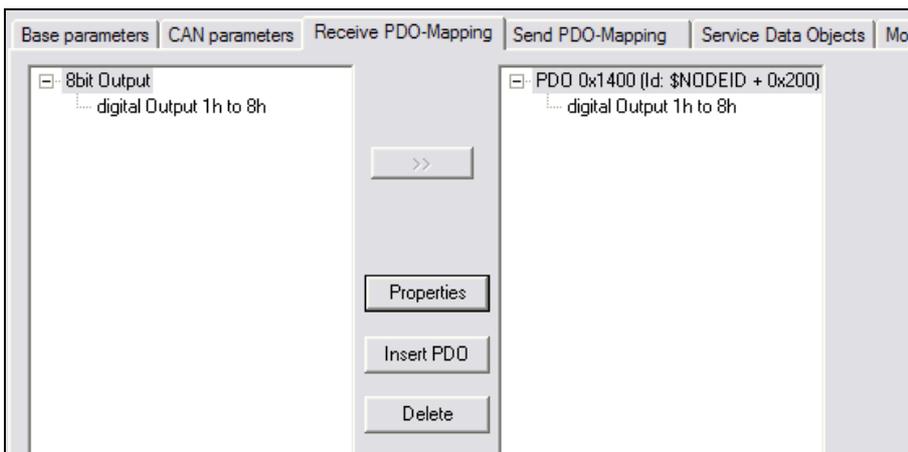
In the window Base Parameters it's possible to insert following parameters:

- Input Address: Shows the input memory location where the configurator starts to place the data of the CANOpen Slave.
- Output Address: Shows the output memory location where the configurator starts to place the data of the CANOpen Slave.
- Diagnostic address: indicates the location in memory where the CANOpen Slave diagnostics data are saved.

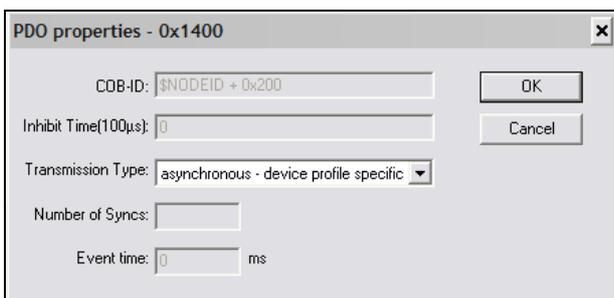


In the CAN parameters window you can configure the following parameters:

- Node ID: Node ID assigned to the slave CANOpen (valid only if the Z-PC CANOpen slave station dip switches has dips 4 .. 10 in OFF).
- Write DCF: creates a DCF file for the node in question (this is an EDS file with the values instantiated in the configurator)
- Create all SDO's: creates and initializes all SDO items indicated in the EDS file, not just those that are changed
- Optional Device: the master checks for the slave device, if there is not recognised continues in its normal operation
- No initialization, the master initiates communication with the node without initialize the objects of the dictionary via SDO.
- Node Guarding: activates or deactivates the NodeGuard protocol for verifying the presence of the station by the slave master.
- Guard time: the period to send the NodeGuard message in milliseconds.
- Life time factor: shows within time intervals the slave must receive a Guard message from the master, if not the slave pass in preoperational state.
- Activate Heartbeat generation: used to activate the Heartbeat protocol (Not supported by Z-PC CANOpen slaves stations).
- Activate Heartbeat consumer: Allows to consume heartbeat messages (Not supported by Z-PC CANOpen slaves stations).
- Emergency Telegram: The station can produce the emergency messages in case of errors or failures.
- Communication Cycle is the time between two SYNC messages



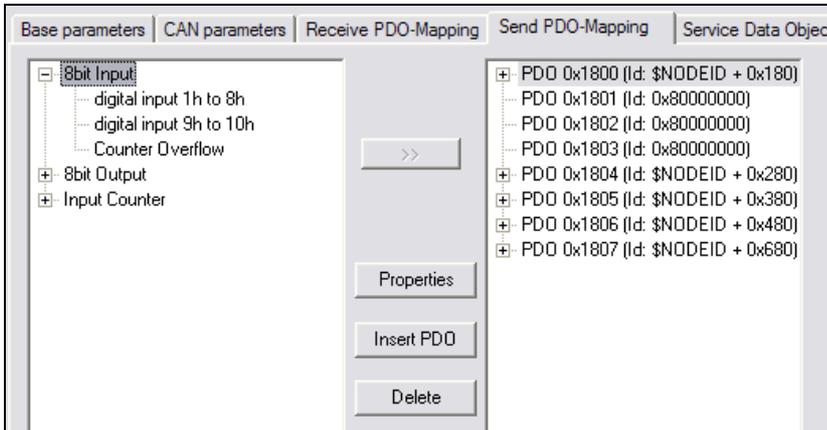
In the Receive PDO Mapping window, you can configure and view the PDO telegrams sent from the master to the slave. On the left you can see the data that the slave is able to receive through PDOs. In the figure case is an 8 digital outputs. On the right you see the link between the PDO and data output.



Clicking to the Properties button it's possible to configure the properties of the PDO. In the window that appears this is an asynchronous transmission type PDO (the PDO is sent only in a trigger

condition). If the type of transmission chosen is synchronous you can select after many SYNC the value must be updated.

In the Send SDO-Mapping window, you can configure and display the PDO sent by the slave master. On the left you can see the data that the slave is able to send via PDO. In the figure case is a 16 digital inputs and 8 digital outputs. On the right you see the link between the PDO and data entry. Similarly to Receive PDO you can set the properties of individual PDO through the appropriate keys.



Through the Service Data Objects, you can configure different values from the default items listed in the dictionary (EDS file). For the meaning of individual items you must refer to the slave documentation. Only the changed values are sent via SDO communication at the slave initialization.

Index	Name	Value	Type	Default
2001	Module Address	0x7F	Unsign...	0x7F
2002	Module Baud Rate	0x07	Unsign...	0x07
2051	Master Command	0	Unsign...	0
2052	Master Aux Command	0	Unsign...	0
2211su...	Preset Value Counter Nr. 1	0	Unsign...	0
2211su...	Preset Value Counter Nr. 2	0	Unsign...	0
2211su...	Preset Value Counter Nr. 3	0	Unsign...	0
2211su...	Preset Value Counter Nr. 4	0	Unsign...	0
2211su...	Preset Value Counter Nr. 5	0	Unsign...	0
2211su...	Preset Value Counter Nr. 6	0	Unsign...	0
2211su...	Preset Value Counter Nr. 7	0	Unsign...	0
2211su...	Preset Value Counter Nr. 8	0	Unsign...	0
6005	Global Interrupt enable	0x0	Boolean	0x0
6007su...	Mask interrupt input 1h to 8h	0xFF	Unsign...	0xFF
6007su...	Mask interrupt input 9h to 10h	0xFF	Unsign...	0xFF
6007su...	Mask interrupt Counter Overflow	0xFF	Unsign...	0xFF
6008su...	Mask interrupt input 1h to 8h	0xFF	Unsign...	0xFF
6008su...	Mask interrupt input 9h to 10h	0xFF	Unsign...	0xFF
6008su...	Mask interrupt Counter Overflow	0x0	Unsign...	0x0
6200su...	digital Output 1h to 8h	0x0	Unsign...	0x0
6205su...	Error Mode 1h to 8h	0xFF	Unsign...	0xFF
6207su...	Error Value 1h to 8h	0x00	Unsign...	0x00
6220su...	Bit Output 1	0x0	Boolean	0x0
6220su...	Bit Output 2	0x0	Boolean	0x0
6220su...	Bit Output 3	0x0	Boolean	0x0
6220su...	Bit Output 4	0x0	Boolean	0x0
6220su...	Bit Output 5	0x0	Boolean	0x0
6220su...	Bit Output 6	0x0	Boolean	0x0
6220su...	Bit Output 7	0x0	Boolean	0x0
6220su...	Bit Output 8	0x0	Boolean	0x0

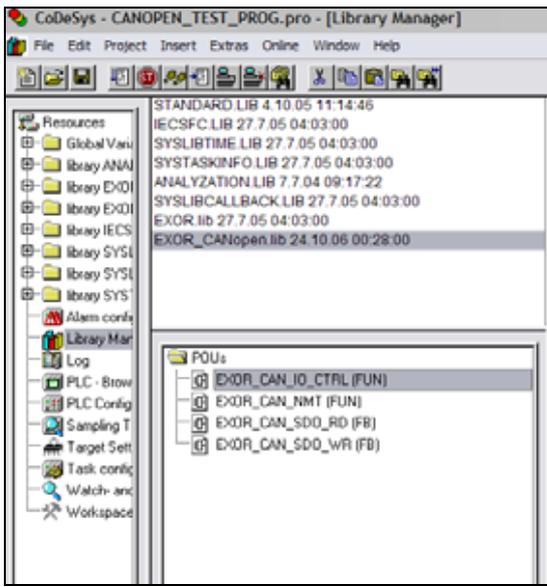
The parameter module window enables the specific slave property into the TWS PLC, in this case the diagnosis:

Index	Name	Value	Def...	Min.	Max.
1	EnableDiags	No	No		

9.4 The CANopen library for CodeSys

This section describes the use of the program library for exchanging data between the master TWS3/TWS64 and the CANopen slave. Please refer to the software manuals for a detailed description.

In order to manage the CANopen code communication it is necessary to include the library EXOR_CANOpen.lib into the project. This transaction is obtained through the window opened from the menu Window | Library Manager:



9.5 EXOR_CAN_NMT Function

Through the EXOR_CAN_NMT function it's possible to act selectively or generally. Selecting the zero address the command is sent to all stations. If the address is different from zero the command is sent only to the station with the same slave node number. For the value Command are possible the following values:

*Start Remote Node (CS=1),
Stop Remote Node (CS=2),
Enter Pre-Operational (CS=128),
Reset Node (CS=129) and
Reset Communication (CS=130).*

```
0009  
0010 R_TRIG1(CLK:=Start_NMT);  
0011 EXOR_CAN_NMT(R_TRIG1.q, 1, ucCmd, ucNodeId); (*Cmd= 1 start all node, 2 stop all node, 128 preoperational *)  
0012
```

9.6 EXOR_CAN_SDO_RD Function

Through the EXOR_CAN_SDO_RD it's possible to read objects from a slave dictionary object. The meaning of individual parameters are:

- bEnable: the reading occurs on FALSE-TRUE toggle
- wCanPort: CAN card port number
- wTXCOBID COB-ID used for the TX SDO message. In the figure case the node-id is 3 (0x600 + nodeid = 1539)
- wRXCOBID COB-ID used for the RX SDO message. In the figure case the node-id is 3 (0x580 + nodeid)
- windex: index of the object that you want to read
- ucSubindex: sub index of the object you want to read

In the next parameters the function returns the data type of reading and the actual value or the error if, for example, the object does not exist.

0013	
0014	EXOR_CAN_SDO_RD_1(
0015	bEnable:=Start_Read ,
0016	wCanPort:= 1,
0017	wTxCOBID:=1539 ,
0018	wRxCOBID:= 1411,
0019	wIndex:=Indice_R ,
0020	ucSubIndex:=Sub_Indice_R ,
0021	wDataType:= ,
0022	bDone=> ,
0023	diErrCod=> ,
0024	dwAbortCod=> ,
0025	diIntegerValue=> ,
0026	rFloatValue=>);

9.6.1 EXOR_CAN_SDO_WR Function

Through the EXOR_CAN_SDO_WR it's possible to write CAN objects from the slave dictionary object.

The meaning of individual parameters are:

- bEnable: the writing occurs on FALSE-TRUE toggle
- wCanPort: CAN card number through which you are writing
- wTXCOBID number the user's message SDO used for transmission. In the case of the node with id 3 (600hex + nodeid)
- wRXCOBID number the user's message SDO used to receive. In the case of the node with id 3 (580hex + nodeid)
- windex: index of the object that you want to write
- ucSubindex: sub index of the object you want to write
- diIntegerValue: value to be written in the object
- rFloatValue: value to be written in the object

In the next parameters the function returns the data type of writing and the actual value or the error if, for example, the object does not exist.

```

0030
0031 Indice_W:=16#6200;
0032 Sub_Indice_W:=16#1;
0033 wTxCOBID_W:=1539; (*SDO_TX_MOD3*)
0034 wRxCOBID_W:=1411; (*SDO_RX_MOD3*)
0035
0036
0037 EXOR_CAN_SDO_WR_1(
0038     bEnable:=Start_Write AND ENABLE_SCRITTURA_SDO ,
0039     wCanPort:=1,
0040     wTxCOBID:=wTxCOBID_W ,
0041     wRxCOBID:= wRxCOBID_W,
0042     wIndex:=Indice_W ,
0043     ucSubIndex:=Sub_Indice_W ,
0044     wDataType:= ,
0045     diIntegerValue:=DatoScrittura ,
0046     rFloatValue:= ,
0047     bDone=> ,
0048     diErrCod=> ,
0049     dwAbortCod=> );

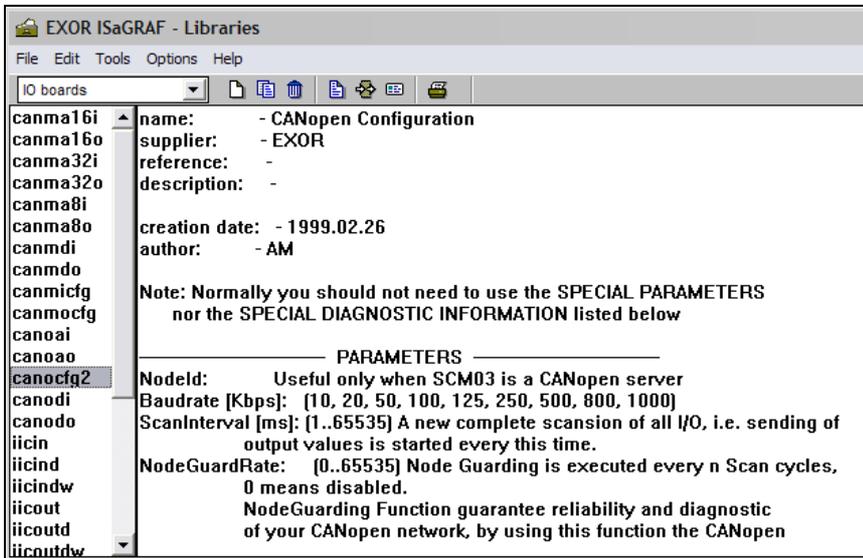
```

10 Quick Start: CANopen with ISAGRAF

This section describes the use of CANOpen in the ISaGRAF environment.

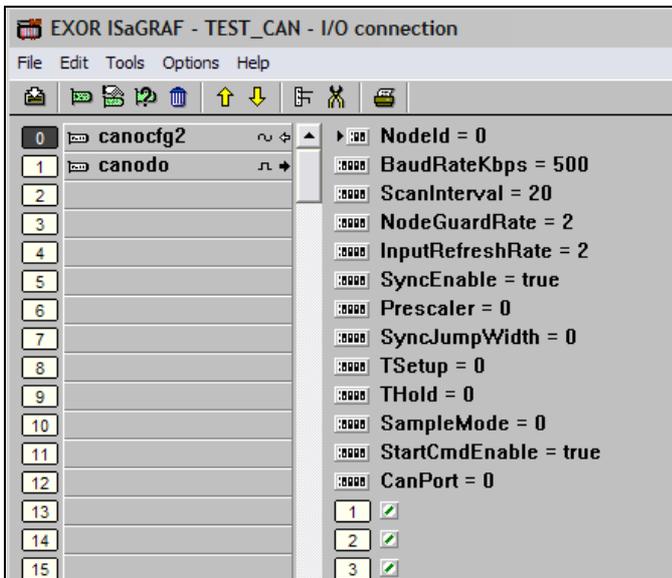
10.1 Isagraf Library

The use of CANOpen in ISAGRAF requires loading library for CANOpen. In particular, it is necessary to load the library for the IO boards canocfg2, canodi, canodo and canao:



10.2 IO Boards canocfg2

The board canocfg2 is used to configure the communication via the TWS CANopen master. The integration of the board is done through the menu Project | IO Connection.



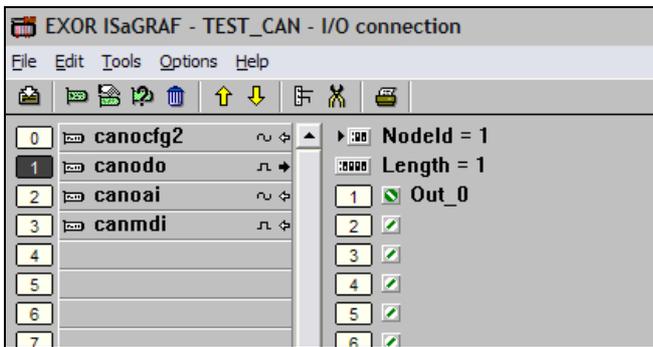
The fields of the IO Boards have the following meanings:

- Nodeid: identification of the CANOpen master node
- Baudrate: Select the speed of the network CANOpen. It must be the same as set by the DIP switch on the slave.
- ScanInterval: scan time of the inputs and outputs in milliseconds.

- NodeGuardRate: cycles number of to scan before sending the NodeGuard message. The value 0 means disabled.
- InputRefreshRate: request for inputs delivery through Remote Transmission Request (not supported by Z-PC CANopen slaves)
- SyncEnable: Enabling SYNC message from the master
- StartCmdEnable: If TRUE the NMT start network command is given at boot-up or in case of Node Guarding error .

The special parameters (prescaler (1 .. 64), SyncJumpWidth (1 .. 4), TSetup (1 .. 16) THold (1 .. 8), SampleMode (0 .. 1)) are used to select the speed of the network and have special meaning only if the parameter is Baudrate to zero. Normally should not be used.

Now you can point directly to the inputs and outputs of the slave stations using the appropriate boards. In the example below shows the use of a canodo for to point to the first digital output of the slave with station number 1.



11 Appendix

11.1 EDS Files

The EDS files for the Z-PC CANopen Slave stations can be downloaded directly from the site <http://www.seneca.it> on the CANopen section.