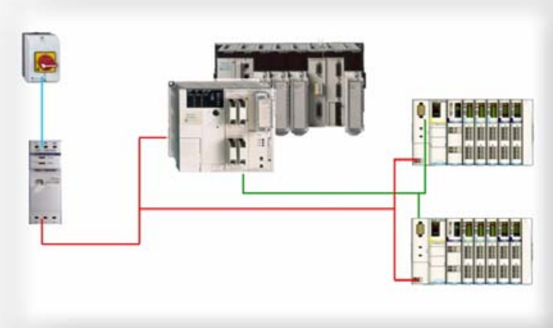
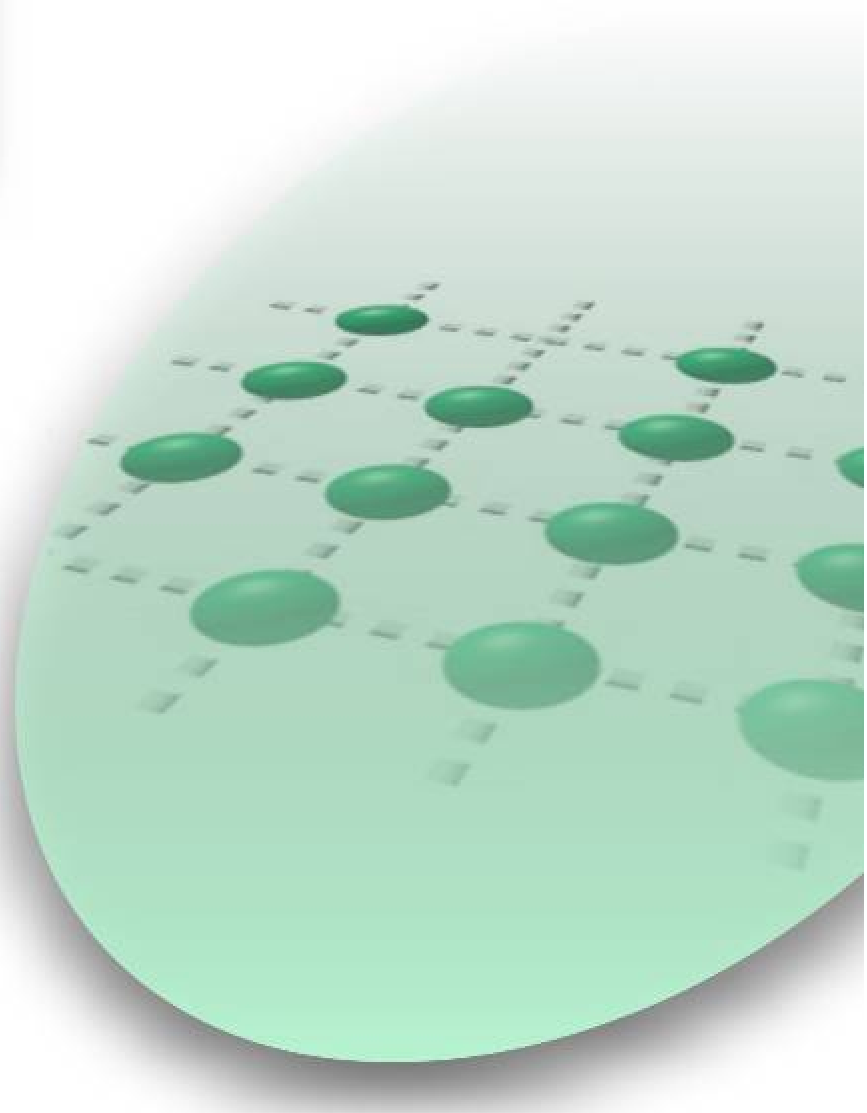
CANopen Basic
PL7, Premium, Micro, Advantys
System User Guide





33003453.01

Merlin Gerin
Square D
Telemecanique

Schneider Electric
Building a New Electric World

Table of Content

Introduction

This document is intended to provide a quick introduction to the described System.
It is not intended to replace any specific product documentation. On the contrary, it offers additional information to the product documentation, for installing, configuring and starting up the system.

A detailed functional description or the specification for a specific user application is not part of this document. Nevertheless, the document outlines some typical applications where the system might be implemented.

.

This document is intended to aid customers who are not familiar with CANopen in their first steps to set up CANopen devices on a Premium or Micro PLC.

It explains how to set up the hardware and which software tools must be used for which purpose during the process of software configuration. Default settings are retained wherever possible in order to facilitate the configuration process and to prevent the user from loosing his orientation.

Abbreviations

Word / Expression	Signification
PLC	Programmable Logic Controller
HMI	Human Machine Interface
VVD	Variable Velocity Drive
PC	Personal Computer
AC	Alternating current
DC	Direct current
PS	Power supply
I/O	Input / Output
CB	Circuit Breaker
ESTOP	Emergency Stop
Premium	A product name for a Schneider midrange PLC
Micro	A product name for a Schneider midrange PLC
Phaseo	A product name for Schneider power supply devices
Magelis	A product name for Schneider HMI devices
Altivar	A product name for Schneider VVD devices
Telefast	A product name for Schneider distributed I/O devices

System

Introduction

The system chapter describes the architecture, the components, the dimensions and the number of components used within this system.

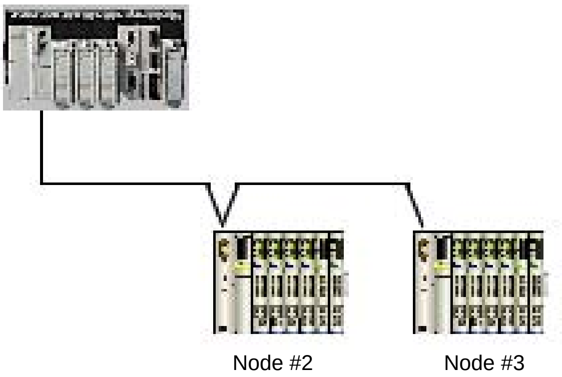
Architecture

Overview

We set up CANopen network as shown below with two Advantys STB on a Premium PLC.

Note: The Premium PLC has been chosen for this example, but everything in this guide also applies to the Micro.

Layout



Node #2 consists of a STBNCO2212 CANopen interface module, a STBPDT310 power supply module, a STBDDI3410 4 bit input module, and a STBDDO3410 4 bit output module.

Node #3 is similar to Node #2, but has additionally 2 analog modules, the STBAVI1270 two channels analog input module and the STBAVO1250 two analog channels output module.

For both nodes, the outputs are wired to the inputs,so that:
- the digital input goes on when the output is set, etc ...
- the analog inputs read the values from the analog outputs

Components

Hardware:
- CANopen master : TSX CPP 110 (PCMCIA card type III,
- DS 301 V4.01 standard)
- On Premium P572xxx to 574xxx: CPU V5.0
- On Premium P571xxx: CPU V5.6
- On Micro : CPU (TSX 372x) V6.0
- 2 CANopen STB Network Interface Modules : STB NCO 2212
- 2 STB power supply modules STB PDT 3100
- STB I/O modules as listed in the description of the configuration example
- 3 CANopen connectors and cable
- Programming cable for PLC

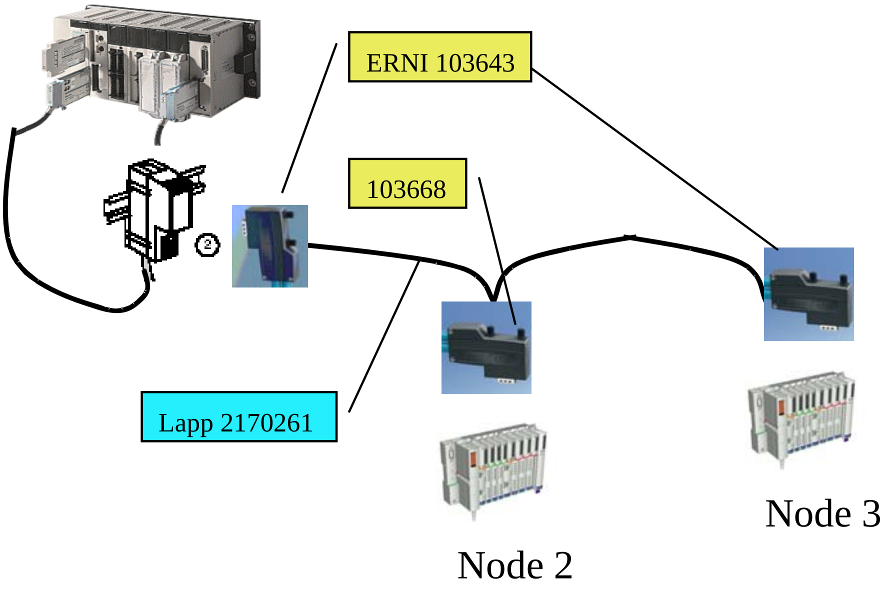
Software :
- Advantys : to configure the STB island
- SyCon V2.8 : to configure the CANopen bus
- PL7 V4.4 : to configure the PLC

Installation

Introduction

This chapter describes the steps necessary to install the hardware and to set up the software to attain the following application architecture.

Layout



Hardware

General

- Assemble the modules incl. wiring and hardware settings (baud rate, network address, ...)
- Assemble the Premium PLC incl. TSX CPP 110 (for this example)
- Prepare and install the CANopen cable

Assemble the STB devices

Connect the STB hot swap bases and mount the modules in the sequence listed below. Changing the sequence of the I/O modules has an impact on the I/O addresses in the state RAM of the PLC.

Step 1 Assemble the modules

Node #2:

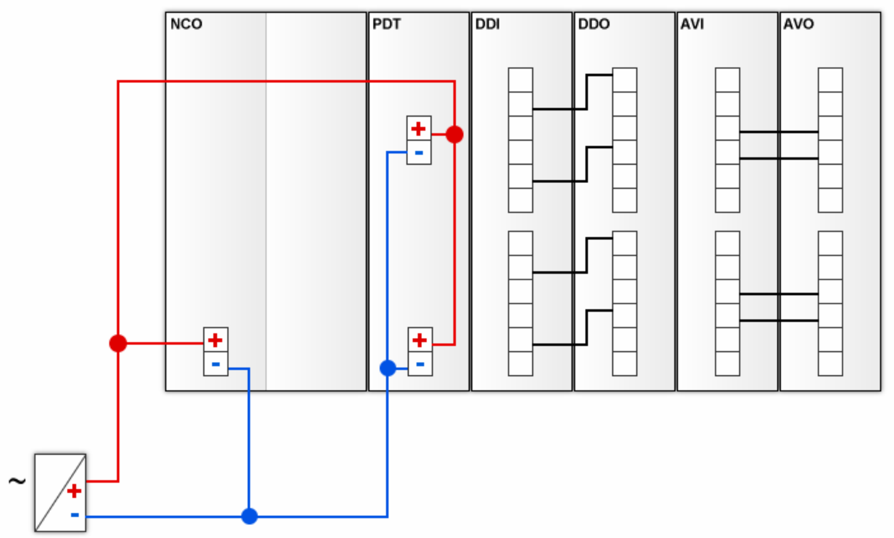
Network interface *STBNCO2212*
Power supply *STBPDT3100*
Digital input module *STBDDI3420*
Digital output module *STBDDO3410*
Termination plate *STBXMP1100*

Node #3:

Network interface *STBNCO2212*
Power supply *STBPDT3100*
Digital input module *STBDDI3420*
Digital output module *STBDDO3410*
Analog input module *STBAVI1270*
Analog output module *STBAVO1250*
Termination plate *STBXMP1100*

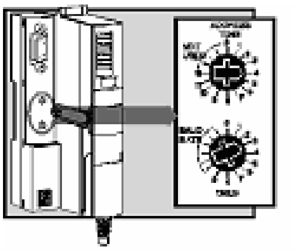
Step 2 Field wire the devices

Wire the Advantys STB. The following illustration shows the wiring of Node #3. The wiring of Node #2 is similar (only the last two modules are missing)



Note that we recommend having a separate power supply for the outputs. For testing purposes, however, you can have one common power supply for inputs, outputs and logic supply (as shown in the figure above).

Step 3
Set up the CANopen baud rate to 250 Kbit



Both the baud rate and the node address are set using the two rotary switches in the middle of the NCO module.

To set the baud rate, proceed as follows :

1. **Check that the power is off**
2. **Set the upper rotary switch (*TENS*) to position 4**
3. **Set the bottom rotary switch (*ONES*) to “Baud Rate” (= any position after 9)**
4. **Power on**

Note that the baud rate will be taken into account after power on and only when the bottom rotary switch is set to position "Baud Rate". The baud rate itself is selected using the upper rotary switch (0 = 10 Kbit, 1 = 20 Kbit, 2 = 50 Kbit, 3 = 125 Kbit, 4 = 250 Kbit, 5 = 500 Kbit, 6 = 800 Kbit, 7 = 1 Mbit).

Step 4
Set up the CANopen node address

The node address is set using the same two rotary switches:

1. Check that the power is off
2. Select 0 on the upper switch (***TENS***)
3. Select 2 on the lower switch (***ONES***) for the module with the CANopen address 2 and 3 for the module with the CANopen address 3
4. Power on

Note: The 2 switches represent the address value. For a CANopen address of 16: select 1 on the upper switch (TENS) and 6 on the lower switch (UNITS).

Note that the node address will be taken into account only after power on. When changing the address without a power cycle, the module will keep the old address until the next power cycle takes place.

Step 5
Load the Advantys STB configuration

Use the Auto-Configuration feature (no SIM card) of the NCO module as follows:

1. Check that the power is on and remove the SIM card if inserted
2. Press the reset button which is located under the door in the bottom of the NCO module for about 5 seconds

Now the Advantys STB boots. The hardware configuration is read from the backplane and stored in a flash memory.

Note that an Advantys STB always trys to load the configuration from the SIM card. When no SIM card is inserted, the configuration is taken from the flash. When the current configuration is different from the one in the flash and you wish to update the flash, push the reset button. Always push the reset button after a configuration change or when the flash configuration is unknown.

Step 6
Last check

Now the Advantys STB is properly set up and the devices are ready to communicate with the CPP110 CANopen master.

The LEDs must show the following status:

- NCO module: "RUN" and "PWR" are set to on, "CANRUN” blinks
- PDT module: "IN" and "OUT" are set to on
- I/O modules: "RDY" is set to on, on every I/O module

Possible Errors

Configuration mismatch

When the configuration in the flash is different from the actual configuration, the LED status is as follows:

- NCO module: "RUN" and "PWR" are set to on, "CANRUN" blinks green, "ERR" and "CANERR" blink red
- PDT module: "IN" and "OUT" are set to on
- I/O modules: "RDY" blinks on every module which does not match with the configuration in flash, "RDY" is on for every other I/O module

Module Error

Some modules can display an error condition (e.g. DDO3230, when output voltage supply is missing). In this case, "RDY" is on and "ERR" blinks on the module, while the NCO module is healthy ("RUN" and "PWR" are set to on, "CANRUN" blinks).

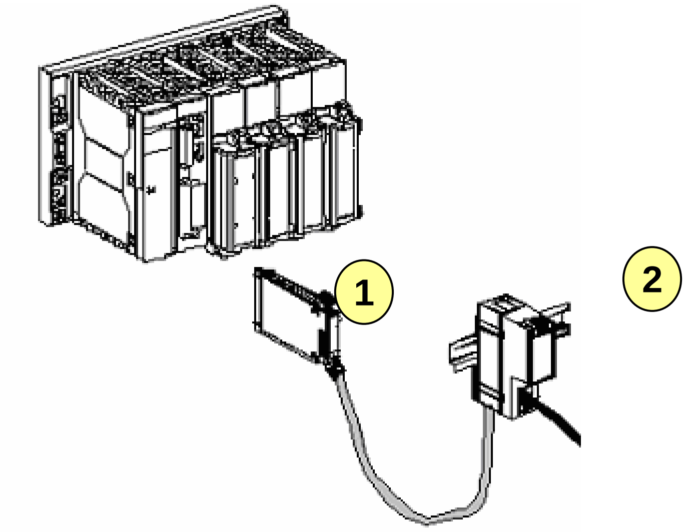
Assemble the Premium

Assemble the PLC as shown in the figure below.

- Ensure that the power supply module is powered off
- Mount all Premium modules in the backplane
- Insert the TSX CPP 110 card into the PCMCIA slot of the processor (1)
- Fix the TAP on a DIN rail (2)
- Wire the power supply module

It is mandatory for the PCMCIA card to be installed in the slot located in the processor module. As a result, only one CANopen bus is available for each PLC CPU.

Note that when the PCMCIA card is inserted the PLC must be powered off.

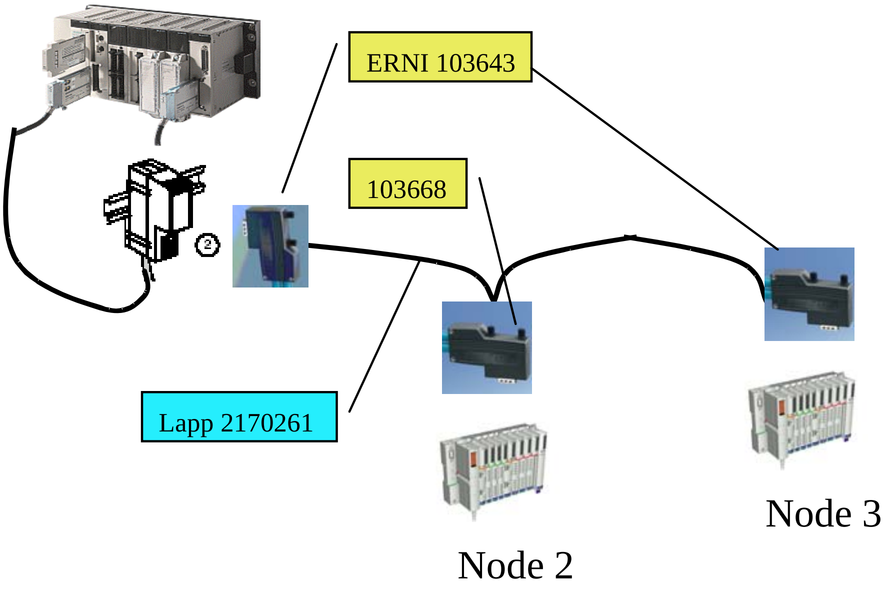


Now the Premium is properly set up, it can be turned on and the software configured.

Prepare and install the CANopen cable

For CANopen, special connectors and cable are available on the market (refer to CANopen cabling document).

For the present configuration, you need to prepare a cable with 3 female SUB D 9 connectors:



Example of CANopen cable is available by **Selectron** under the product ref:

- DCA 701 (article number 44170014)

For any additional information, consult http://www.selectron.ch/

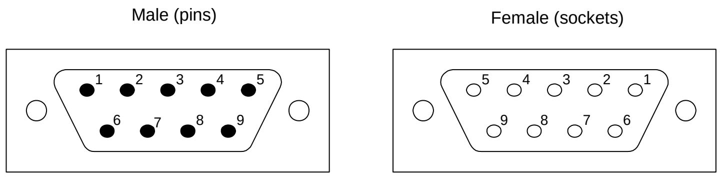
Example of cable from **Lapp**: http://www.lappcable.com/products/

- UNITRONIC BUS CAN 2170261: 120 Ohms shielded double twisted pair cable

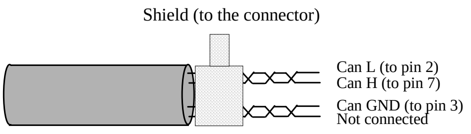
Example of connectors from **ERNI**: http://www.erni.com/

- 1 x ref 103668 for daisy chain (plugged on Node 2)
- 2 x ref 103643 for the end of the bus (includes the line termination; plugged on the TSXCPP110 tap and on Node 3)

CANopen connectors normally have screw type terminals and must be assembled manually, according to the following pin layout:



Pin N°	Signal	Description
2	CAN_L	CAN_L bus Line
3	CAN_GND	CAN ground
7	CAN_H	CAN_H bus Line



Pins 2, 3, and 7 must be connected.

Implementation

Introduction

The implementation chapter describes all the steps necessary to initialize, parameterize, program and to start-up the system.

Function

Functional description

We set up the following CANopen network with two Advantys STB on a Premium PLC.

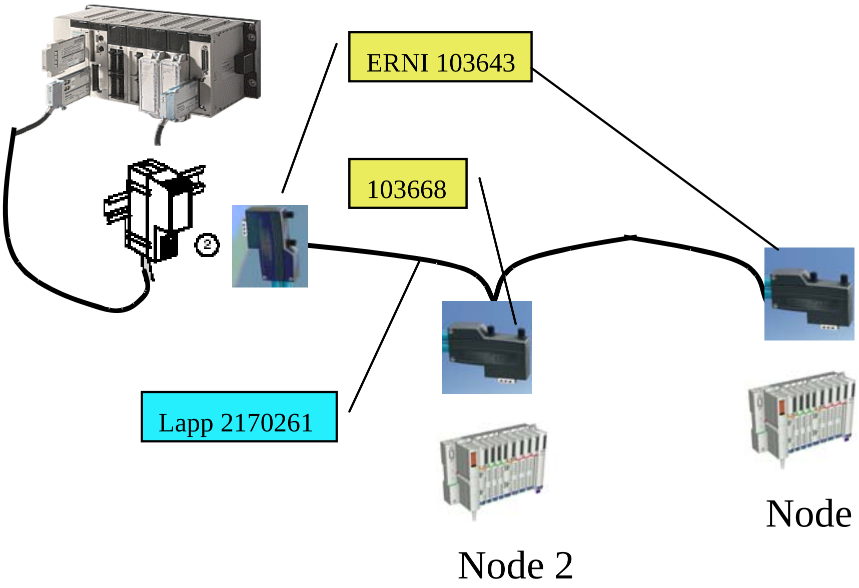
Note: The Premium was chosen for this example, but this guide also applies to the Micro.

Node #2 consists of a STBNCO2212 CANopen interface module, a STBPDT310 power supply module, a STBDDI3410 4 bit input module, and a STBDDO3410 4 bit output module.

Node #3 is similar Node #2, but has additionally 2 analog modules, the STBAVI1270 two channels analog input module and the STBAVO1250 two analog channels output module.

For both nodes, the outputs are wired to the inputs so that:

- the digital input goes on when the output is set, etc ...
- the analog inputs read the values from the analog outputs



Software Configuration

Introduction

The software configuration consists of three major steps:

1. Create the Advantys STB configuration and generate an EDS file for each node (Advantys software)
2. Create the CANopen configuration (SyCon software)
3. Create the PLC application (PL7 software) and transfer the project to the PLC.

Once the system is running and you can write outputs to / read inputs from the CANopen devices.

Create the Advantys STB configuration

The main purpose of the Advantys tool is:

- To modify the default parameters of the I/O modules (i.e. switching off behavior of outputs, ...)
- To load the Advantys configuration into the SIM card (if any)
- To generate the EDS files

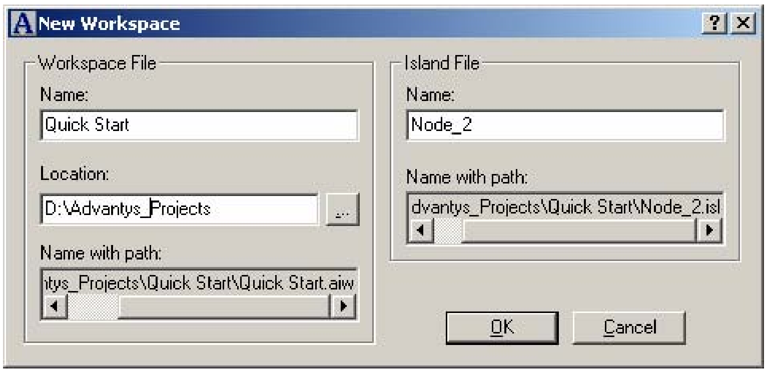
As we use in our example the module's default settings and we have already loaded a configuration from flash (refer to section Hardware configuration), we only use this tool to generate an EDS file for each of the nodes. These EDS files provide all the information on the nodes needed by SyCon to configure the bus.

Note that the Advantys tool is not mandatory. You can also use a generic EDS file, which is available in SyCon. This, however, requires deeper CANopen knowledge and advanced usage of the SyCon software. EDS files created by the Advantys software are dedicated to the individual configuration, of each node and keep the configuration work to a minimum.

Note:If by you want to load the configuration through the Advantys tool, use the menu Online/Connect then Online/Download into the I/O island (in this case, specific cable is required).

Steps to create the Advantys configuration Advantys tool - Step 1 Create a new workspace

Start the Advantys tool, create a new workspace and enter name and path.



New Workspace

Workspace File

Name: Quick Start

Location: D:\Advantys_Projects

Name with path: itys_Projects\Quick Start\Quick Start.aiw

Island File

Name: Node_2

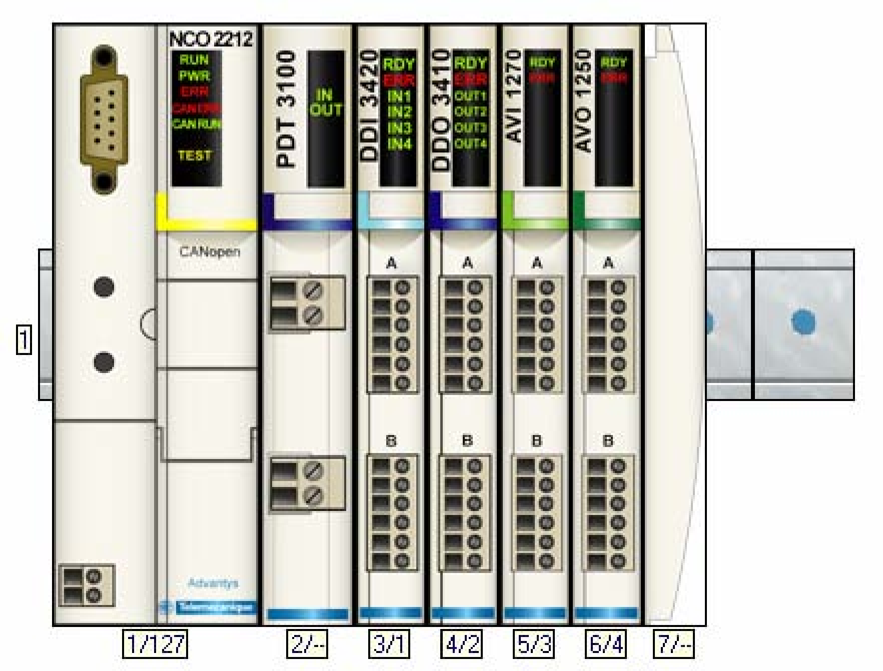
Name with path: dvantys_Projects\Quick Start\Node_2.isl

In our example, we have chosen the path D:\Advantys_Projects\Quick Start. The name of the project file is Quick Start.aiw and the name of the Advantys STB is Node_2 (referring to its CANopen node address). All the Advantys STB on the same bus must be declared in the same workspace.The Default workspace path is C:\program Files\Schneider Electric\Advantys\Project\

Advantys tool - Step 2 Configure the STB nodes

Now open a workspace with a DIN rail for Node #2 and configure Node #2 according to its hardware configuration using drag and drop and the modules from the hardware catalog on the right side of the screen. Do not forget the termination plate (Ref STB XMP 1100). Then, create a new node ("Add new Island" from the "File menu"), name it Node_3 and configure it according to the hardware configuration of Node #3.

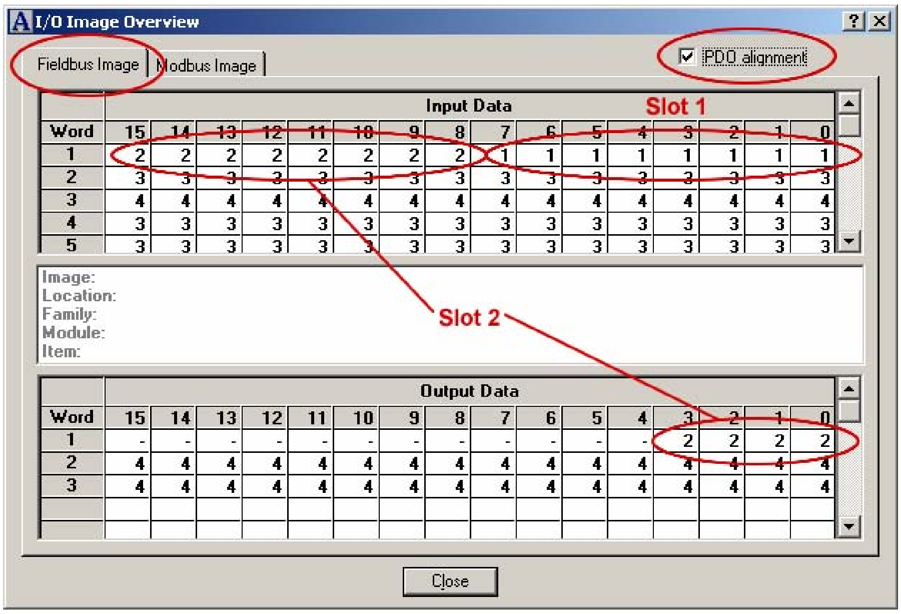
The following figure shows the Node #3 configured correctly.



Note: You can read out the configuration when you are in online mode. In this case, the power supply module and the termination plate will be missing as they cannot be detected on the island's backplane. You must add them manually.

Advantys tool - Step 3 Display the Fieldbus I/O image

Open the fieldbus I/O image using the menu I/O image overview. Select Node #3, click on the TAB “Fieldbus image” and select PDO alignment as shown in the screen below.



Word	Input Data															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Image:
Location:
Family:
Module:
Item:

Word	Output Data															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	-	-	-	-	-	-	-	-	-	-	-	-	2	2	2	2
2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Do the same for Node #2. Make a hardcopy of both screens as it helps in understanding the IO mapping.

In the PLC memory, Node #3 assigns 3 words of output data and 5 words of input data. The table is read as follows: You can find the input of slot 1 (DDI module) in the input word 1 (low byte), the I/O of slot 2 (DDO module) in input word 1 and output word 1, ...

General mapping rules are:

- First a block with discrete I/O, then the block with analog I/O
- Within the blocks, the I/O points are sorted by the physical sequence of the I/O modules.
- Discrete I/O points are mapped into the discretes block and are, sorted by number. First the I/O points, followed by, the echo (outputs only) and then the status. Analog channels are sorted by number. The input/output values are mapped into the analog input/output block, the status bytes are mapped into the discrete input block.

Detailed mapping interpretation of Node #3

Input Data				
Word	15..12	11..8	7..4	3..0
1	Status bits- slot 2	Echo bits- slot 2	Status bits- slot 1	Input bits- slot 1
2	Status bytes- slot 3		Status bytes- slot 3	
3	Status bytes- slot 4		Status bytes- slot 4	
4	Input channel - slot 3			
5	Input channel - slot 3			
Output Data				
Word	15..12	11..8	7..4	3..0
1				Output bits- slot 2
2	Output channel- slot 4			
3	Output channel- slot 4			

Advantys tool - Step 4
Create the EDS files

Select node 2 and create the EDS file using **File->Export....** Select "Node_2" as name for the EDS file.

Do the same for node 3.

In our example the files are exported to the following directory : D:\Advantys_Projects\Quick Start*.eds.

Now the Advantys STB configuration is complete. You have generated the EDS files as output and you are now ready to start the CANopen configuration with SyCon.

Create the CANopen configuration

With the CANopen configuration, we generate an electronic description of the CANopen fieldbus. This description contains all information that PL7 needs to configure the CPP110 CANopen master.

Perform the following steps:

SyCon tool - Step 1
Create a new SyCon project

Start the SyCon tool (it can be opened from the PL7 configuration screen, see: PL7 tool – Step 1) and open a new CANopen project. Save the empty project as ...\Demo_cfg.co. The default path is ...\SyCon\Project\.

You will need the path and the filename later, as PL7 needs it during the PLC configuration.

SyCon tool - Step 2
Import the EDS files

Using Menu **File** then **Copy EDS**, Copy the EDS files node_2.eds and node_3.eds that you have generated with the Advantys tool. Dont import the bitmap file (those files don't exist). Files to be imported in this example are in the following directory (refer to Advantys tool - Step 4) : D:\Advantys_Projects\Quick Start*.eds.

SyCon tool - Step 3
Insert the TSX CPP 110

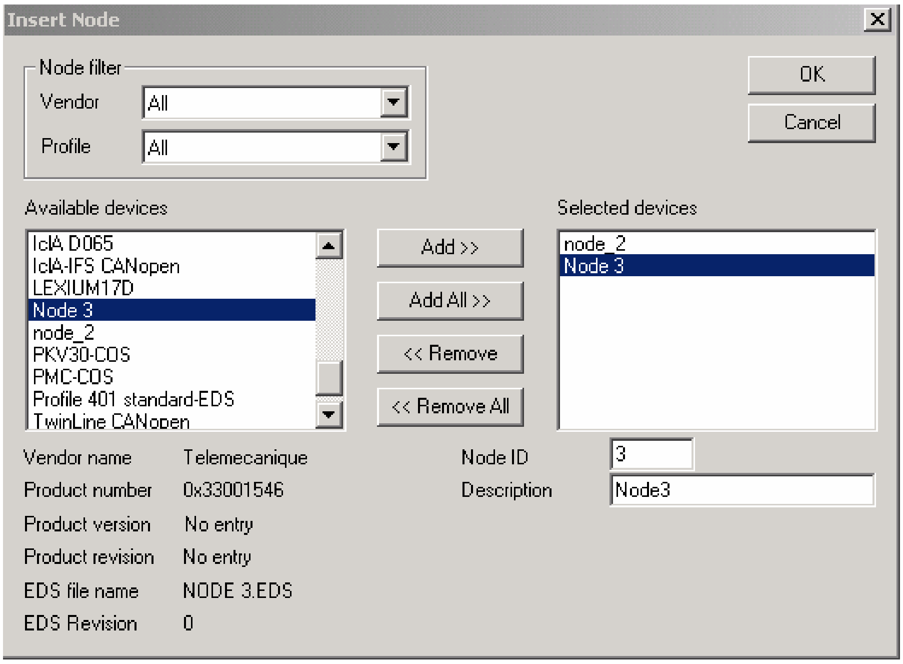
Insert the CANopen master TSX CPP 110 (**Insert->Master...).**

Keep the node address #1. SyCon offers it as a default value.

SyCon tool - Step 4 Insert the nodes "Node_2" and "Node_3"

Insert Node #2 (**Insert->Node...** and choose Node_2 from the list of available devices). Keep the node address #2 that SyCon offers as default value then click on the **Add>>** button and validate it with **OK**.

Do the same for Node #3.



Insert Node

Node filter
Vendor: All
Profile: All

Available devices:
IclA D065
IclA-IFS CANopen
LEXIUM17D
Node 3
node_2
PKV30-COS
PMC-COS
Profile 401 standard-EDS
TwinLine CANopen

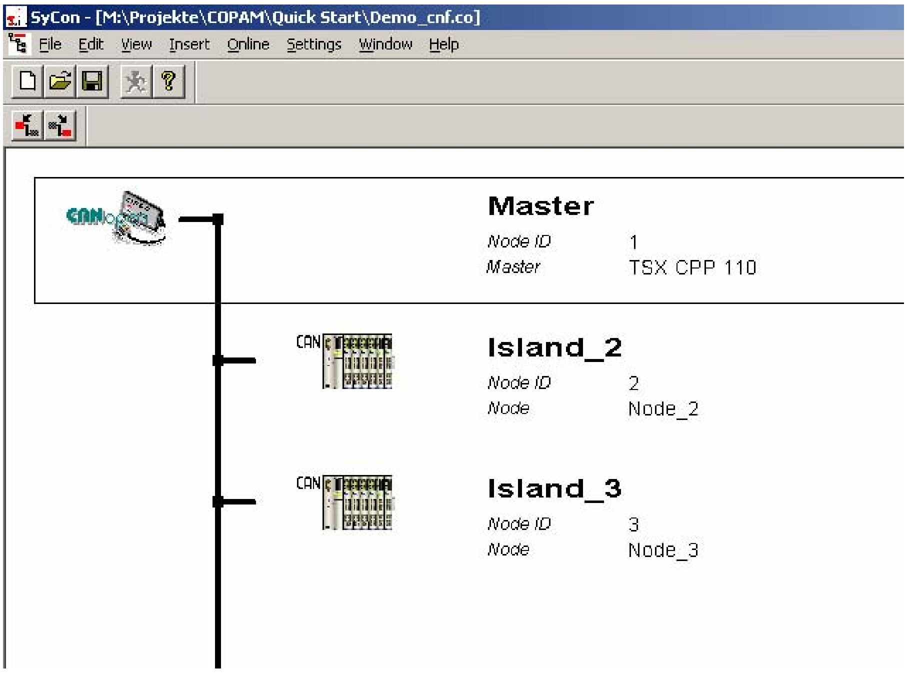
Selected devices:
node_2
Node 3

Vendor name	Telemecanique
Product number	0x33001546
Product version	No entry
Product revision	No entry
EDS file name	NODE 3.EDS
EDS Revision	0

Node ID: 3
Description: Node3

Configuration screen in SyCon

Now SyCon shows the following CANopen configuration screen. SyCon has taken the names of Node_2 and Node_3 from the EDS files.



Master
Node ID 1
Master TSX CPP 110

Island_2
Node ID 2
Node Node_2

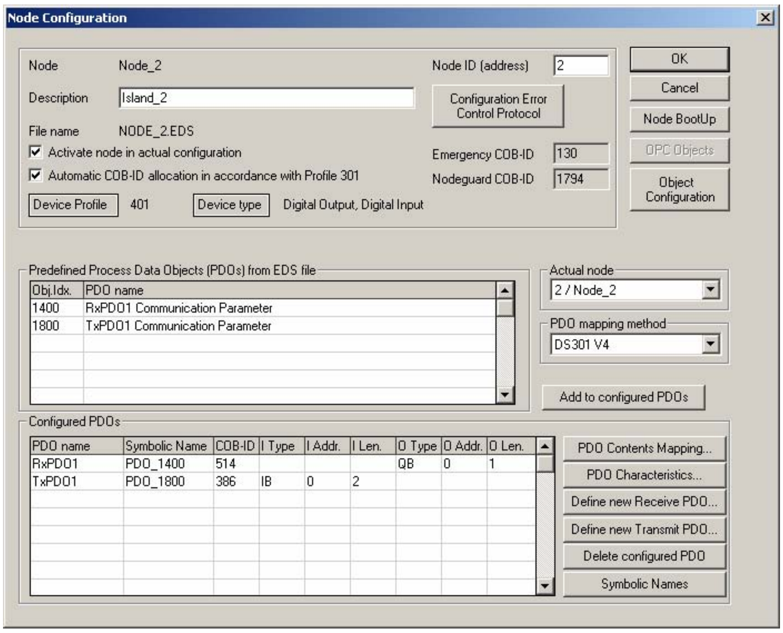
Island_3
Node ID 3
Node Node_3

SyCon tool - Step 5 Set the Baud rate to 250 kBit

Simply Click on the TSX CPP 110 and then select **Settings->Bus Prameter** in the menu. Change the Baud rate to 250 kBit/s, the value previously set on the Advantys STB hardware.

SyCon tool - Step 6 Configure the PDOs for Node #2

Double click on Node_2. The Node configuration screen opens an shows two PDOs in the“Predefined Process Data” grid. The first PDO is a Receive-PDO (RxPDO) to configure the output data for Node #2, the second is a Transmit-PDO (TxPDO) to configure the input data from Node #2. (The transmission direction is always seen from the node’s point of view.)
Double click on the first PDO and validate the transmission type window. (We use the predefined settings from this screen.) Now you have configured the first PDO SyCon has all the necessary information from the EDS file you created with the Advantys tool. Do the same for the second PDO and the PDO mapping for Node #2 is finished. Now the screen looks as follows:



Node Configuration

Node: Node_2
Node ID (address): 2
Description: Island_2
File name: NODE_2.EDS
☑ Activate node in actual configuration
☑ Automatic COB-ID allocation in accordance with Profile 301
Emergency COB-ID: 130
Nodeguard COB-ID: 1794
Device Profile: 401
Device type: Digital Output, Digital Input

Predefined Process Data Objects (PDOs) from EDS file

Obj.Idx.	PDO name
1400	RxPDO1 Communication Parameter
1800	TxPDO1 Communication Parameter

Actual node: 2 / Node_2
PDO mapping method: DS301 V4

Configured PDOs

PDO name	Symbolic Name	COB-ID	I Type	I Addr.	I Len.	O Type	O Addr.	O Len.
RxPDO1	PDO_1400	514				QB	0	1
TxPDO1	PDO_1800	386	IB	0	2			

Click on **OK** to validate and close the node configuration window.

SyCon tool - Step 7 Configure the PDOs for Node #3

Do the same with Node #3. For Node #3, SyCon offers four predefined PDOs, two Receive PDOs and two Transmit PDOs.

- RxPDO1 is defining the PLC digital output data
- RxPDO2 is defining the PLC analog output data
- TxPDO1 is defining the PLC digital input data
- TxPDO2 is defining the PLC analog input data

Configure all 4 PDOs in the same way as you did make with Node #2.

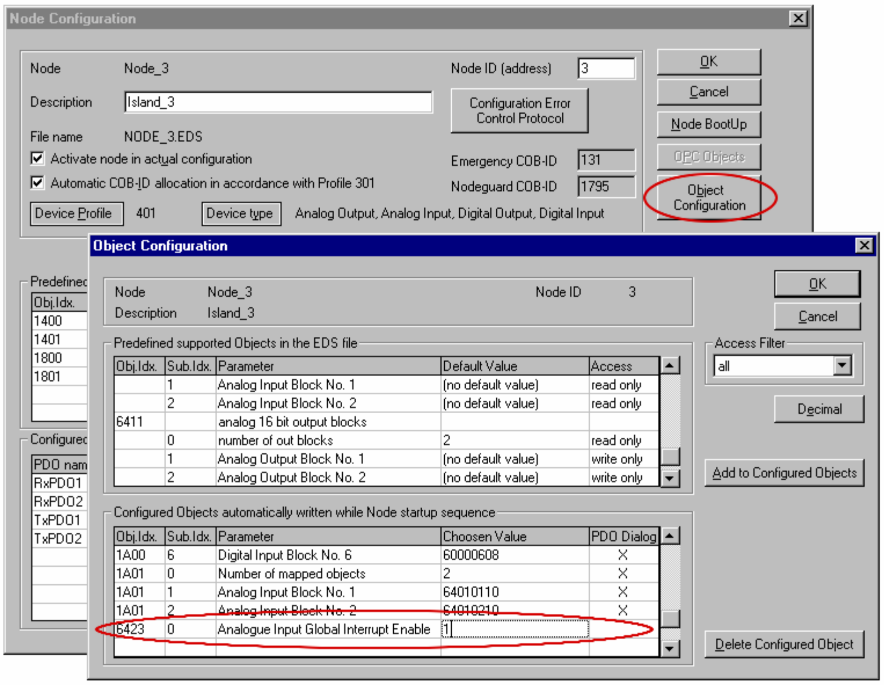
Warning: You have to map all offered PDOs and make no changes to these PDOs, otherwise your I/O mapping in PL7 does not correspond to the address table in the Advantys tool.

SyCon tool - Step 8 Enable analog input transmission for Node #3

By default, the transmission of analog input values is disabled on the modules. Perform the following steps to enable analog input transmission:

1. Open the *Node configuration* window for Node #3
2. Press on the *Object Configuration* button

3. Double click on the object *6423 : Analog Input Global Enable* in the list of Predefined supported Objects
4. Enter *1* in the *Chosen Value* to validate the analog input



Node Node_3

Node ID (address) 3

Description Island_3

File name NODE_3.EDS

☑ Activate node in actual configuration

☑ Automatic COB-ID allocation in accordance with Profile 301

Emergency COB-ID 131

Nodeguard COB-ID 1795

Device Profile 401 Device type Analog Output, Analog Input, Digital Output, Digital Input

Predefined

Obj.Idx.
1400
1401
1800
1801

Configured

PDO nam
RxPDO1
RxPDO2
TxPDO1
TxPDO2

Node Node_3 Node ID 3

Description Island_3

Access Filter all

Predefined supported Objects in the EDS file

Obj.Idx.	Sub.Idx.	Parameter	Default Value	Access
	1	Analog Input Block No. 1	(no default value)	read only
	2	Analog Input Block No. 2	(no default value)	read only
6411		analog 16 bit output blocks		
	0	number of out blocks	2	read only
	1	Analog Output Block No. 1	(no default value)	write only
	2	Analog Output Block No. 2	(no default value)	write only

Configured Objects automatically written while Node startup sequence

Obj.Idx.	Sub.Idx.	Parameter	Choosen Value	PDO Dialog
1A00	6	Digital Input Block No. 6	60000608	X
1A01	0	Number of mapped objects	2	X
1A01	1	Analog Input Block No. 1	64010110	X
1A01	2	Analog Input Block No. 2	64010210	X
6423	0	Analogue Input Global Interrupt Enable	1	

Close the window and save the project. You have now finished the CANopen configuration with SyCon and created the necessary data required by PL7 to configure the TSX CPP110 module.

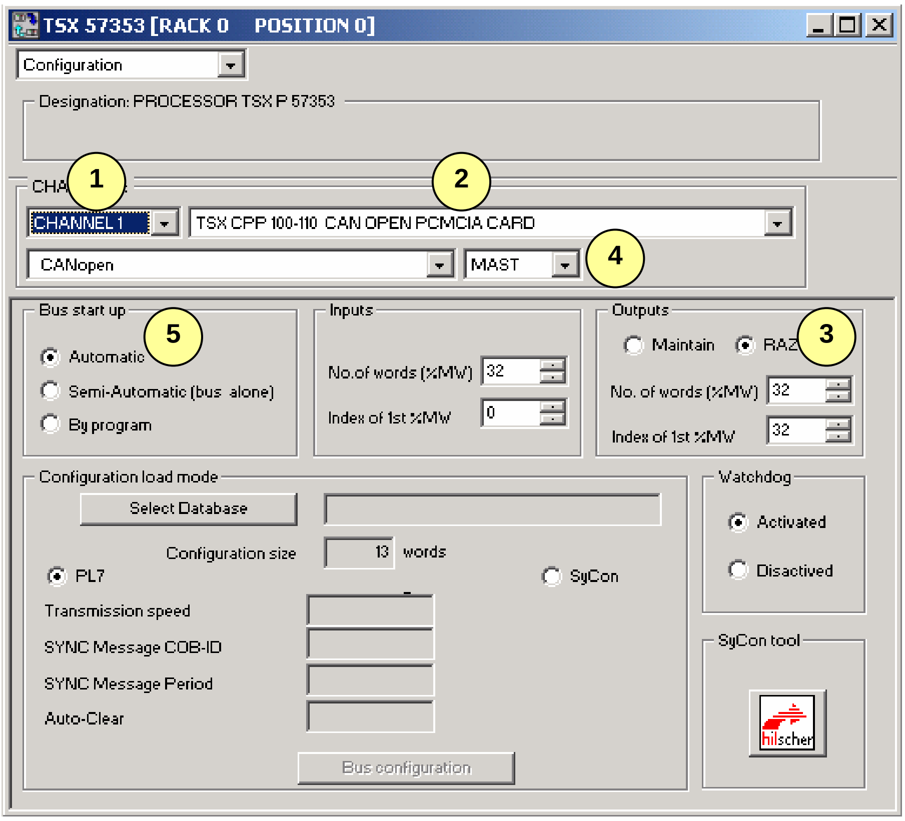
The information is available in a database, the default path for our example is ....\SyCon\Projects\demo_cfg.co.

You are now ready to start with the PL7 application.

Create the PL7 Application

The final phase is the programming of the application program for the PLC. For this you use PL7 with the imported CANopen project created above**.**

PL7 tool – Step 1



TSX 57353 [RACK 0 POSITION 0]

Configuration

Designation: PROCESSOR TSX P 57353

CHANNEL 1 | TSX CPP 100-110 CAN OPEN PCMCIA CARD

CANopen | MAST

Bus start up
- ● Automatic
- ○ Semi-Automatic (bus alone)
- ○ By program

Inputs
- No.of words (%MW) 32
- Index of 1st %MW 0

Outputs
- ○ Maintain ● RAZ
- No. of words (%MW) 32
- Index of 1st %MW 32

Configuration load mode
- Select Database
- Configuration size 13 words
- ● PL7 ○ SyCon
- Transmission speed
- SYNC Message COB-ID
- SYNC Message Period
- Auto-Clear
- Bus configuration

Watchdog
- ● Activated
- ○ Disactived

SyCon tool

Start PL7, configure the hardware and double click on the PCMCIA slot of the CPU

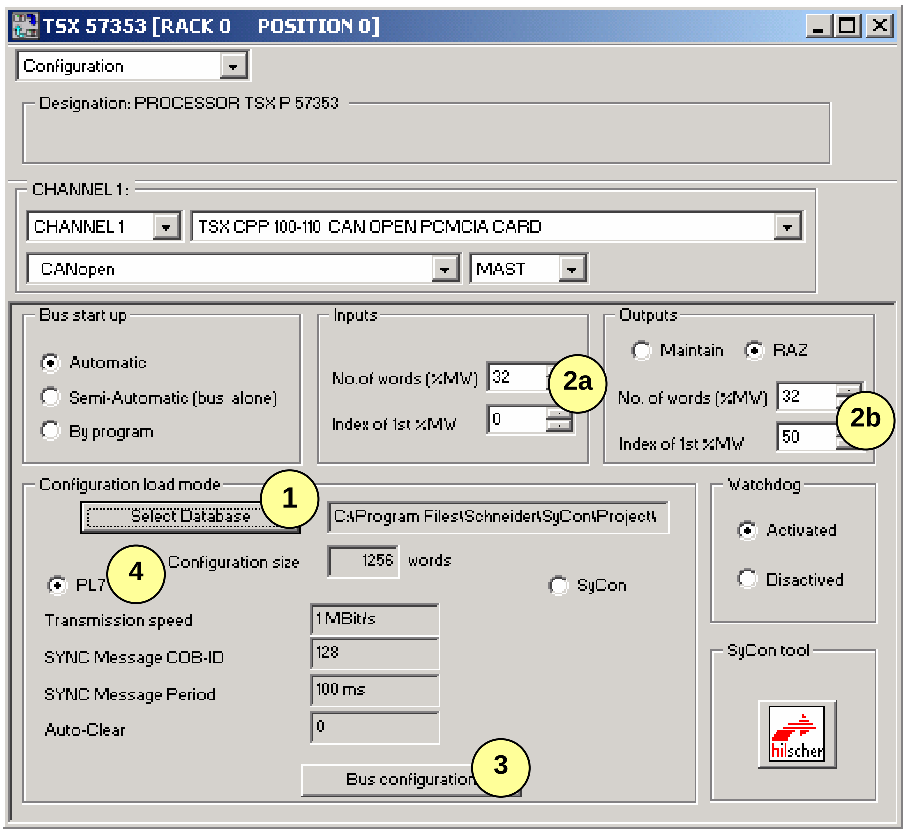
1. Select the *Channel 1*
2. Select the *TSX CPP100 - 110* card
3. Define the output behavior in case of PLC stop: Maintain or Reset

By default:

4. “Mast” task is selected for the rate of update of the storage area associated with the I/O
5. Select “Automatic” bus start up mode

Note that SyCon tool could be launched from this screen.

PL7 tool – Step 2



TSX 57353 [RACK 0 POSITION 0]

Configuration

Designation: PROCESSOR TSX P 57353

CHANNEL 1 :

CHANNEL 1 | TSX CPP 100-110 CAN OPEN PCMCIA CARD

CANopen | MAST

Bus start up
- ● Automatic
- ○ Semi-Automatic (bus alone)
- ○ By program

Inputs
- No.of words (%MW) 32
- Index of 1st %MW 0

Outputs
- ○ Maintain ● RAZ
- No. of words (%MW) 32
- Index of 1st %MW 50

Configuration load mode
- Select Database
- C:\Program Files\Schneider\SyCon\Project\
- Configuration size 1256 words
- ● PL7 ○ SyCon
- Transmission speed 1 MBit/s
- SYNC Message COB-ID 128
- SYNC Message Period 100 ms
- Auto-Clear 0
- Bus configuration

Watchdog
- ● Activated
- ○ Disactived

SyCon tool

1. Click on *Select Database* to import the *.co file into PL7 (example : Demo_cfg.co)
2. Define the addresses of the Inputs and Outputs:
   Example:
- 2a : Inputs: Array of 32 words (from %MW0 to %MW31) We keep default values.
- 2b : Outputs: Array of 32 words (from %MW50 to %MW81).
3. Press on *Bus configuration* to see the list of nodes configured on the bus.

By default:

4. “PL7” mode is selected to have the CANopen configuration loaded together with the PL7 application into the PLC

Close the window, confirm all changes and save the project. You have now finished the PLC application with PL7 and created all necessary data the PLC needs to start the communication with the CANopen nodes.

You have now reserved 32 words for inputs and for outputs. The input words start at %MW0, the output words at %MW50. As Node #2 is using one input word and one output word and Node #3 five input words and three output words, we have the following address assignment:

Inputs Node #2:	%MW0
Outputs Node #2:	%MW50
Inputs Node #3:	%MW1 to %MW5
Outputs Node #3:	%MW51 to %MW53

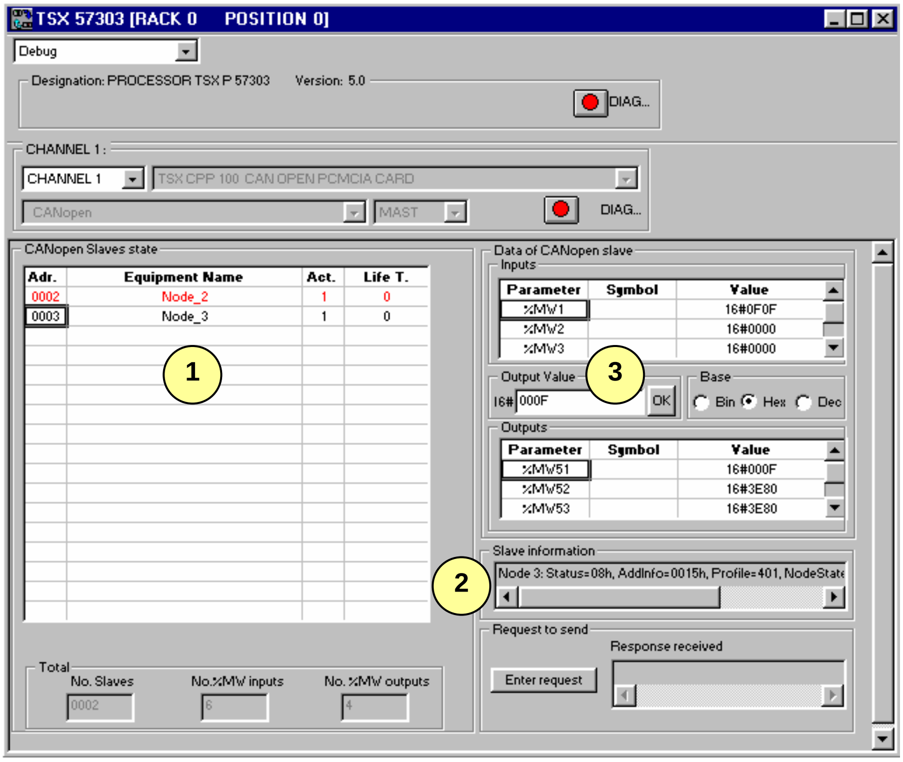
(Refer to Advantys tool – Step 3)

Transfer the application to the PLC and start the program.

Debugging

PL7 Debug Screen

For debugging use the CPP110 debug screen



TSX 57303 [RACK 0 POSITION 0]

Debug

Designation: PROCESSOR TSX P 57303 Version: 5.0

DIAG...

CHANNEL 1 :

CHANNEL 1 — TSX CPP 100 CAN OPEN PCMCIA CARD

CANopen — MAST — DIAG...

CANopen Slaves state

Adr.	Equipment Name	Act.	Life T.
0002	Node_2	1	0
0003	Node_3	1	0

Data of CANopen slave

Inputs

Parameter	Symbol	Value
%MW1		16#0F0F
%MW2		16#0000
%MW3		16#0000

Output Value: 16# 000F OK

Base: ○ Bin ● Hex ○ Dec

Outputs

Parameter	Symbol	Value
%MW51		16#000F
%MW52		16#3E80
%MW53		16#3E80

Slave information

Node 3: Status=08h, AddInfo=0015h, Profile=401, NodeState

Request to send — Enter request

Response received

Total

No. Slaves	No.%MW inputs	No. %MW outputs
0002	6	4

PL7 Debug - Step 1 Node overview

1. The node list provides an overview of all configured nodes. Node #2 is shown in red because it is not connected to the bus. Node #3 is working properly and therefore is shown in black.

If one or more nodes are red, the DIAG button (and the CPP error led) is red too. The Premium resets the error LED and the DIAG button automatically when the fault has been corrected. For the TSX Micro, it must be reset by a positive edge on bit %QW0.1:X2.

PL7 Debug - Step 2 Node diagnostic

2. Node diagnostic. To obtain more details, click on the red listed node. The diagnostic field shows the string : *Node 2: Status=01h, AddInfo=0000h, Profile=0, NodeState=127, Error= 34, EmcyEntries= 0*
   Status = 01 indicates that Node #2 is not answering.

   The diagnostic string of Node #3 shows *Node 3: Status=08h, AddInfo=0015h, Profile=401, NodeState=5, Error= 0, EmcyEntries= 0*
   Status=8 indicates that the node is controlled by the TSX CPP 110.

   For more details refer to the TSX CPP 100/110 user manual.

PL7 Debug - Step 3 Set I/O points

3. Set output word %MW51 to 16#000F, all four outputs will go on. You can see the echo of the outputs in the MSB (Most Significant Byte) of %MW1. As the outputs are also wired to the inputs, you also see the inputs in the LSB (Less Significant Byte) of %MW1 too.

   Set output word %MW52 and %MW53 to 16000 (=16#3E80). The output channels put out 5 VDC and feed them back to the analog input channels. You can read this value back in the input words %MW4 and %MW5 (by using the scroll bar).

   Input word %MW2 contains one Status byte for each of the two analog input channels, Input word %MW3, the Status bytes for the two analog output channels.

   Refer to Advantys tool-Step 4, where I/O Mapping is described.

More than 4 PDOs required for a node

More than 4 PDOs required for a node

You can configure up to 4 PDOs per direction per node. If you require more, some additional steps are required. This is due to the following:

How COB-Ids are assigned

COB-IDs for PDOs are in the range of 385 ... 1407 (hex 180 ... 57F). In general, the user is free in the choice of the COB-ID for a given PDO. However; the user has to take care to stay in this range and should not use a COB-ID twice. CAN open configuration tools normally provide an automatic COB-ID allocation which takes care of this.

Sycon uses the following algorithm in accordance to profile 301:

		Node #1 (decimal)	Node #2 (decimal)	.....	Node #127 (decimal)
1. TxPDO	16#180+Node-ID	385	386	.....	511
1. RxPDO	16#200+Node-ID	513	514	.....	639
2. TxPDO	16#280+Node-ID	641	642	.....	511
2. RxPDO	16#300+Node-ID	769	770	.....	895
3. TxPDO	16#380+Node-ID	897	898	.....	1023
3. RxPDO	16#400+Node-ID	1025	1026	.....	1151
4. TxPDO	16#480+Node-ID	1153	1154	.....	1279
4. RxPDO	16#500+Node-ID	1281	1282	.....	1407

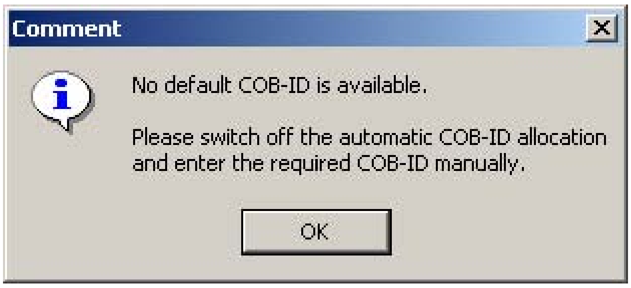
As the COB-ID determines the priority of a frame (the lower the ID is, the higher the priority), this has the following consequence:

The first PDO of a node has higher priority than the second or the third Transmit PDO1 has higher priority than Receive PDO1, Transmit PDO2 than Receive PDO2, etc.

The lower the node ID, the higher the priority of the PDO’s.

Steps to configure PDO5

Note that the range for COB-IDs allows each node to have 4 Transmit PDOs and 4 Receive PDOs. A 5th PDO is not given its COB ID automatically as there are no more free numbers left.
Configure PDO5 in the Node configuration screen of Sycon. The following window opens:



Comment

No default COB-ID is available.

Please switch off the automatic COB-ID allocation and enter the required COB-ID manually.

Disable the automatic COB-ID allocation (node configuration screen). Now you can manually overwrite the COB-ID for PDO5, PDO6, ...

Use the following COB-IDs:

- Transmit PDO 5, 6, ... in the range of 1664 ... 1759 (Hex 680 ... 6DF)
- Receive PDO 5, 6, ... in the range of 1761 ... 1792 (Hex 6E1 ... 700)

Now you can close the node configuration screen and save your project.

Basics of CANopen

CANopen specific terms

The following terms and abbreviations are helpful in understanding the communication principles in a CANopen network.

EDS files

EDS = Electronic Data Sheet. An EDS file describes the communication properties of a device (baudrates, transmission types, I/O offer, ...). It is used in the configuration tool to configure a node (like a driver in a Windows operating system).

CO files

*.CO files are configuration files generated by the SyCon tool. They are imported into PL7 and contain all necessary information the TSXCPP110 needs to configure the CANopen the nodes and to exchange I/O data.

PDO

PDO = Process Data Object. CANopen frame containing I/O data. We distinguish between:

- Transmit-PDOs (TxPDOs with data provided by a node) and
- Receive PDOs (RxPDOs with data to be consumed by a node).

The transmission direction is always viewed from a node point of view.A PDO does not necessarily contain the whole data image of a node (for both TxPDO and RxPDO). Normaly, analog input data and discrete input data are divided onto different TxPDOs.The same is true for outputs.

SDO

SDO = Service Data Object. CANopen frames containing parameters. As the data of PDOs is automatically handled by the CANopen nodes (according to the configuration in SyCon) SDOs must be launched by function blocks in the application. As our example does not require SDOs, for a further explanation refer to the TSX.CPP100/CPP110 user manual (reference TSX DM CPP100/110 CAN open, available on the PL7 documentation CD).

SDOs are typically used to read parameters from / write parameters to drives while the application is running.

Transmission Types

CAN open frames can be either sent **cyclically**, on **change of state**, or on **remote request**.

For each PDO you can define a transmission type (in SyCon). This reduces the network load. (In this guide we use the default settings and do not need to go deeper into this subject. For more information refer to the TSX CPP100/CPP110 user manual).

COB-ID

COB-ID = Communication Object Identifier. Each CANopen frame starts with a COB-ID and is the identifier in a CAN frame. During the configuration phase each node receives the COB-ID(s) for the frame(s) it provides and for the frames it uses.
In a CANopen PDO you don’t find the node ID of a provider or consumer as is common for other types of network. This role is taken over by the COB ID and this enables the spread of the I/O image of a node over more than one PDO.
Each of these PDOs can be sent with a different transmission type and different priority. This also allows for more than one consumer for a PDO (they only have to be sensitive to the same COB-ID).

For more details on COB-ID assignment, refer to “**more than 4 PDOs required for a node**” above.

Contact

Author	Telephone	e-mail
Schneider Electric GmbH Customer & Market System & Architecture Architecture Definition Support	+49 6182 81 2555	**cm.systems@de.schneider-electric.com**

Schneider Electric GmbH
Steinheimer Strasse 117
D - 63500 Seligenstadt
Germany

As standards, specifications and designs change from time to time, please ask for confirmation of the information given in this publication.