

CANopen network Quick Start Guide on Premium and Micro PLCs

eng February 2005

Document Set

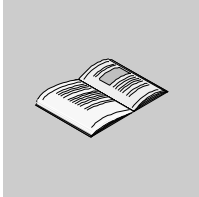
At a Glance

This document is a First-Time User Guide to the configuration of a CANopen network on a Premium or Micro PLC.

Its aim is to help clients with no experience of CANopen networks to carry out the first steps in a CANopen peripheral configuration process on a Premium or Micro PLC.

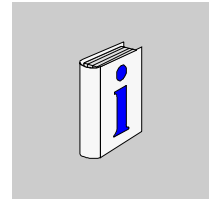
It describes how to install the hardware and which software tools to use for the various tasks to be carried out during the software configuration process. Default parameters are stored whenever possible to facilitate the configuration process and to avoid confusing the user.

Table of Contents



	About the Book	7
Chapter 1	General	9
	Terminology	9
Chapter 2	Configuration	11
	At a Glance	11
	Using the AMO2CA001V000 extension module	12
	Configuring a Lexium Drive	14
	Configuration procedures	15
	Configuring the Lexium	20

About the Book



At a Glance

Document Scope This document is a First-Time User Guide to the configuration of a CANopen network on a Premium or Micro PLC.

User Comments We welcome your comments about this document. You can reach us by e-mail at techpub@schneider-electric.com.

General



Terminology

Special CANopen terms

The following terms and abbreviations will help you better understand the basic principles of CANopen network communications.

EDS Files

EDS = Electronic Data Sheet. The EDS file describes the communication properties for a peripheral (transmission speed, transmission type, I/Os available, etc.). It is used in the configuration tool to configure a node (equivalent to a pilot in a Windows operating system).

CO Files

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

PDO

PDO = Process Data Object. CANopen frame containing I/O data. There are two types of PDOs:

- Output PDOs (TxPDO objects containing data from a node) and
- Reception PDOs (RxPDO objects containing data for use by a node)

Transmission direction is always determined from the point of view of the node. A PDO (whether a TxPDO or an RxPDO) does not necessarily contain a full description of node data. As a rule, analog input data and discrete input data are divided between different TxPDOs. The same applies to output data.

SDO

SDO = Service Data Object. CANopen frames containing parameters. As the PDO data are automatically managed by the CANopen nodes (according to the SyCon configuration), the SDOs must be activated by the application function blocks. As our configuration example can be described without using SDOs, see the TSX CPP100/110 operating guide (reference TSX DM CPP100/110 CAN open, available on the PL7 documentation CD) for further information on SDOs. SDOs are generally used for read/write operations on the drives when the application is running.

Transmission types

CANopen frames can be sent on a cyclical basis, at each status change or following a remote request. A transmission type can be defined (in SyCon) for each PDO. This reduces the network load. (In this guide, we will use the default parameters, and will not therefore describe parametering in detail. For further information on this subject, see the TSX CPP100/PPP110 User Guide).

COB-ID.

COB-ID = Communication Object Identifier. Each CANopen frame begins with a COB-ID which is the CAN frame identifier. During the configuration phase, each node receives the COB-IDs for the frames sent and the frames used by the node. In a CANopen PDO, there is no supplier node or consumer node identifier, as is the case in other networks. The COB ID identifier also serves as the node identifier, which means the I/O image for a node can be reproduced on several PDOs. Each PDO can be sent with its own transmission type and priority. This means there can be several PDO consumers (providing they are activated by the same COB-ID). For further details on COB-ID allocation, see the appendix at the end of the guide.

Configuration

2

At a Glance

Aim of This Chapter

This chapter describes CANopen peripheral configuration using a Premium or Microz PLC.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Using the AMO2CA001V000 extension module	12
Configuring a Lexium Drive	14
Configuration procedures	15
Configuring the Lexium	20

Using the AMO2CA001V000 extension module

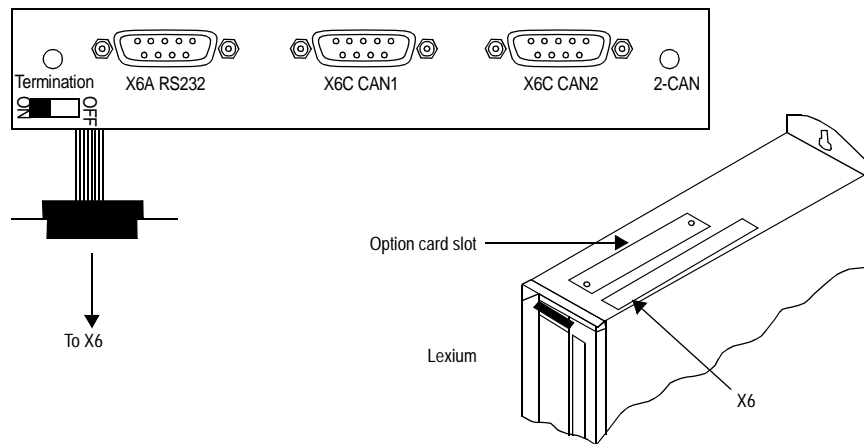
At a Glance

The X6 connector on Lexium drives carries the signals for an RS232 link and for a CAN bus. This means that a special cable is needed to ensure that both the RS232 link and the CAN bus remain immediately available on the drive. Moreover, pin assignment on this X6 connector is not standard compliant.

The AMO2CAN001V000 module can be used to solve the problems listed above, as it includes:

- a male SubD connector for the RS232 link,
- two SubD connectors for the CAN bus, both these connectors are cabled in parallel.

Pin assignment for the three connectors is standard compliant. The module also includes a switch to activate the end of line resistance (120 Ohm) for the CAN bus. The figure below shows the Lexium connector positions.



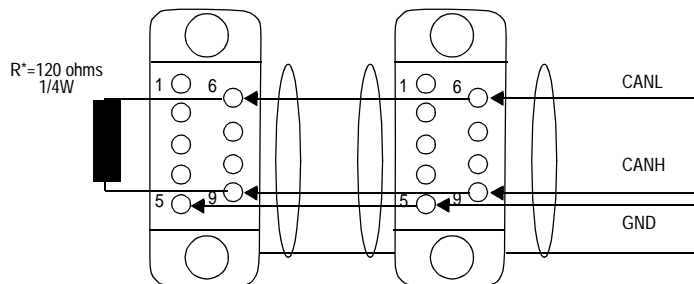
Standard cables with can be used, with shielding for the RS232 and CAN interfaces.

**Note: When the drive is the last item on the CAN bus, the end switch must be on ON.
Otherwise, the switch must remain on OFF (delivery state).**

Communications with the Lexium

If the AMO2CA001V000 extension module is not used, a SUB D 9 point female connector is required.

The figure below shows the SUB D 9 connector cabling.



Adapting the network:

- According to ISO 11898, the bus should have an impedance of 120 ohms. The line length usable for secure communications decreases as the transfer speed increases.

For any other type of cabling, see the Lexium CANopen documentation (available on the "Motion tools" CD).

Configuring a Lexium Drive

At a Glance

The PDOs allow implicit management of the data between a Micro or Premium PLC and the Lexium drive via PLC words.

The Lexium uses two types of PDOs:

- **Predefined PDOs:**
 These PDOs are pre-mapped in the Lexium via special commands.
 e.g.: PDO 22 for this appendix has the following mapping:
 - Request H6040 => Command word
 - Request H2060 => Current or setpoint speed.

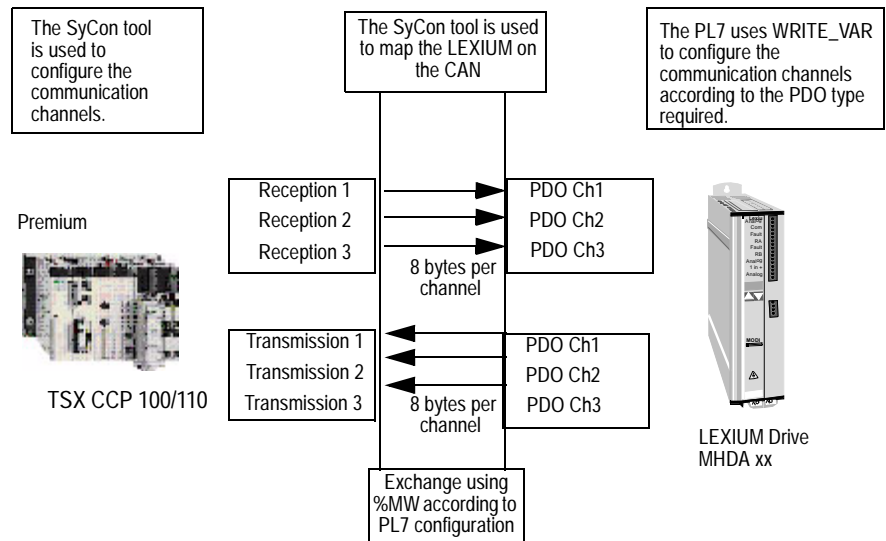
- **Free PDOs:**
 Special commands are not required to map these PDOs: the application will configure them. Free PDOs are not described in this guide.

PDOs are exchanged by using predefined channels (available on the >= SV5.51 version Lexium):

- 3 reception channels (H2600, H2601, H2602)
- 3 transmission channels (H2A00, H2A01, H2A02).

Maximum data size is 8 bytes per channel

The figure below illustrates the mapping principle



Configuration procedures

Communications with the Lexium - PDO22

PDO configuration is carried out in 3 different stages:

- PDO configuration using the SyCon tool: definition of channel, exchange, size and PDO used,
- CANopen configuration using the PL7 or Unity Pro tool: database selection and I/O exchange variable definition,
- Lexium Drive control using the PL7 or Unity Pro debug screen.

The example below shows the different steps for configuring reception PDO 22 on channel 1.

Once these steps have been carried out, the values for the servo-drive will be directly available in the CPO variables (%MW).

Note: This example will explain how to configure a receive PDO. To display drive status in the PLC, also configure a transmit PDO.

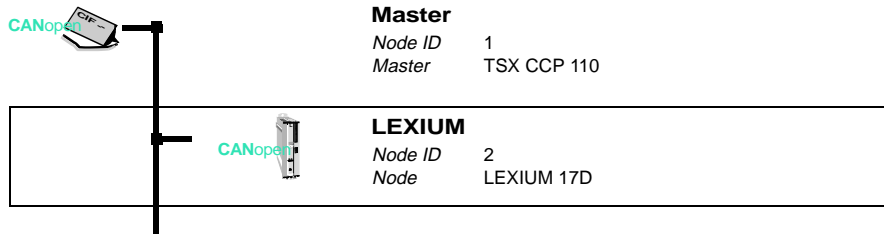
SyCon Tool - Step 1 Inserting the Lexium node

PDO 22 is used on reception channel 1.

Predefined PDO 22 configuration:

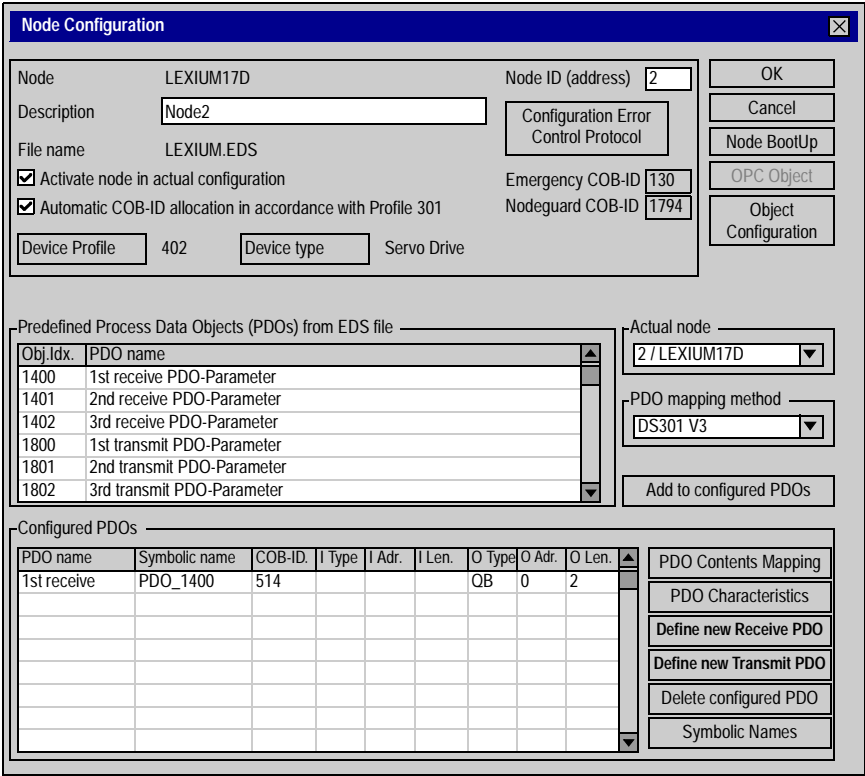
- Command word
- Setpoint current or speed.

Use SyCon to insert the Lexium in the CANopen configuration:

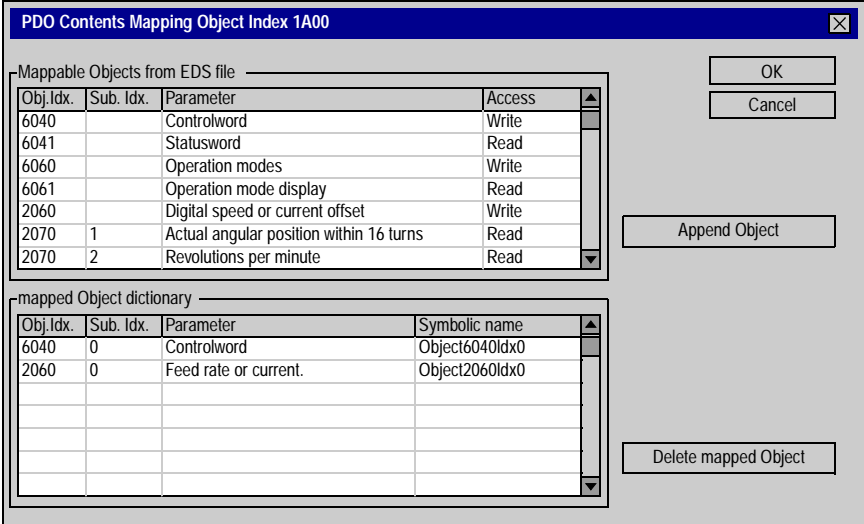


SyCon Tool - Step 2
Configuring the 1st reception PDO

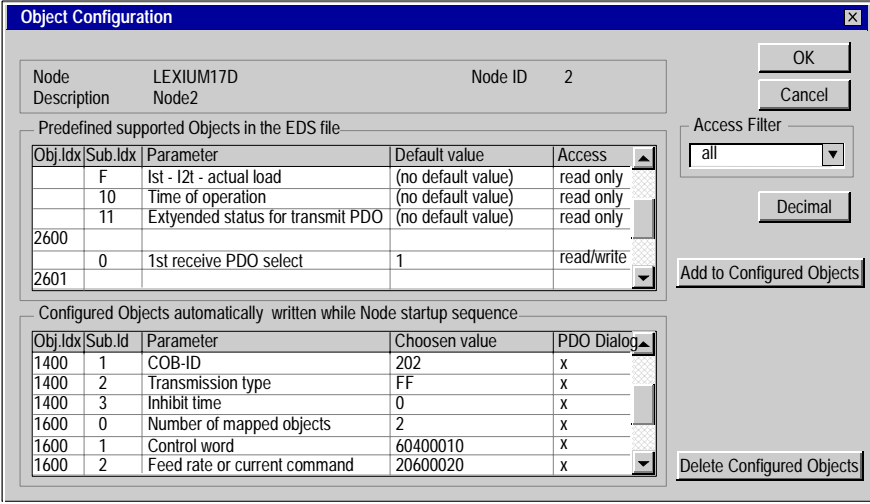
The table below shows the procedure for configuring the 1st reception PDO:

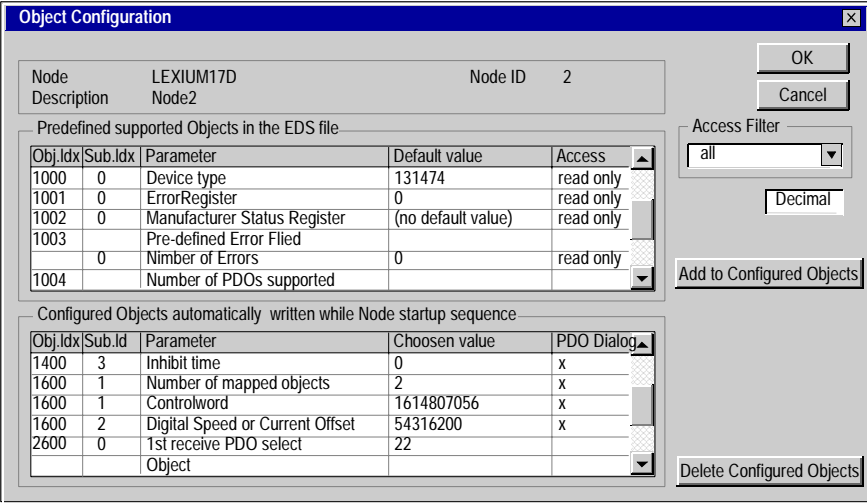
Step	Action
1	<p>In the Node Configuration window, select the 1st receive PDO in the PDOs predefined in the EDS file zone, in the PDO Name field.</p> 
2	Click on the Add to configured PDOs button.
3	In the Configured PDOs field, select the PDO 1s receive PDO name.
4	Click on the PDO Contents Mapping button.

SyCon Tool - Defining the length of the PDO (description of parameters to be exchanged)
Step 3
Configuring the
PDO parameters

Step	Action																																																								
1	<p>In the Mapping PDO Content Index Object 1600 window, in the EDS file objects zone, select the parameter to be exchanged.</p>  <p>The screenshot shows a dialog box titled "PDO Contents Mapping Object Index 1A00". It contains two tables and several buttons. The first table, "Mappable Objects from EDS file", has columns for Obj.Idx., Sub. Idx., Parameter, and Access. The second table, "mapped Object dictionary", has columns for Obj.Idx., Sub. Idx., Parameter, and Symbolic name. Buttons for "Append Object", "Delete mapped Object", "OK", and "Cancel" are located on the right side of the dialog.</p> <table border="1" data-bbox="294 487 884 690"> <thead> <tr> <th>Obj.Idx.</th> <th>Sub. Idx.</th> <th>Parameter</th> <th>Access</th> </tr> </thead> <tbody> <tr> <td>6040</td> <td></td> <td>Controlword</td> <td>Write</td> </tr> <tr> <td>6041</td> <td></td> <td>Statusword</td> <td>Read</td> </tr> <tr> <td>6060</td> <td></td> <td>Operation modes</td> <td>Write</td> </tr> <tr> <td>6061</td> <td></td> <td>Operation mode display</td> <td>Read</td> </tr> <tr> <td>2060</td> <td></td> <td>Digital speed or current offset</td> <td>Write</td> </tr> <tr> <td>2070</td> <td>1</td> <td>Actual angular position within 16 turns</td> <td>Read</td> </tr> <tr> <td>2070</td> <td>2</td> <td>Revolutions per minute</td> <td>Read</td> </tr> </tbody> </table> <table border="1" data-bbox="294 706 884 909"> <thead> <tr> <th>Obj.Idx.</th> <th>Sub. Idx.</th> <th>Parameter</th> <th>Symbolic name</th> </tr> </thead> <tbody> <tr> <td>6040</td> <td>0</td> <td>Controlword</td> <td>Object6040Idx0</td> </tr> <tr> <td>2060</td> <td>0</td> <td>Feed rate or current.</td> <td>Object2060Idx0</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Obj.Idx.	Sub. Idx.	Parameter	Access	6040		Controlword	Write	6041		Statusword	Read	6060		Operation modes	Write	6061		Operation mode display	Read	2060		Digital speed or current offset	Write	2070	1	Actual angular position within 16 turns	Read	2070	2	Revolutions per minute	Read	Obj.Idx.	Sub. Idx.	Parameter	Symbolic name	6040	0	Controlword	Object6040Idx0	2060	0	Feed rate or current.	Object2060Idx0												
Obj.Idx.	Sub. Idx.	Parameter	Access																																																						
6040		Controlword	Write																																																						
6041		Statusword	Read																																																						
6060		Operation modes	Write																																																						
6061		Operation mode display	Read																																																						
2060		Digital speed or current offset	Write																																																						
2070	1	Actual angular position within 16 turns	Read																																																						
2070	2	Revolutions per minute	Read																																																						
Obj.Idx.	Sub. Idx.	Parameter	Symbolic name																																																						
6040	0	Controlword	Object6040Idx0																																																						
2060	0	Feed rate or current.	Object2060Idx0																																																						
2	Click on the Append Object button.																																																								
3	Click on the OK button.																																																								

**SyCon Tool -
Step 4
Defining PDO
value (PDO 22)**

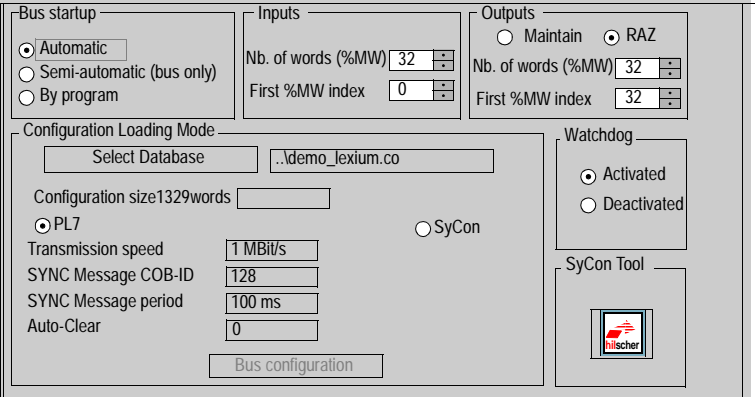
Step	Action
1	<p>In the Node Configuration screen, click on the Object Configuration button.</p> <p>Result: The Object Configuration window is displayed.</p> 
2	<p>In the Predefined supported Objects in the EDS file zone, select "1st receive PDO" for predefined object 2600 (see "General - Lexium") .</p>
3	<p>Click on the Add to Configured Objects button.</p>

Step	Action																																																																						
4	 <p>Object Configuration</p> <p>Node LEXIUM17D Node ID 2 Description Node2</p> <p>Predefined supported Objects in the EDS file</p> <table border="1" data-bbox="299 370 934 516"> <thead> <tr> <th>Obj.Idx</th> <th>Sub.Idx</th> <th>Parameter</th> <th>Default value</th> <th>Access</th> </tr> </thead> <tbody> <tr> <td>1000</td> <td>0</td> <td>Device type</td> <td>131474</td> <td>read only</td> </tr> <tr> <td>1001</td> <td>0</td> <td>ErrorRegister</td> <td>0</td> <td>read only</td> </tr> <tr> <td>1002</td> <td>0</td> <td>Manufacturer Status Register</td> <td>(no default value)</td> <td>read only</td> </tr> <tr> <td>1003</td> <td>0</td> <td>Pre-defined Error Filed</td> <td></td> <td></td> </tr> <tr> <td></td> <td>0</td> <td>Number of Errors</td> <td>0</td> <td>read only</td> </tr> <tr> <td>1004</td> <td></td> <td>Number of PDOs supported</td> <td></td> <td></td> </tr> </tbody> </table> <p>Configured Objects automatically written while Node startup sequence</p> <table border="1" data-bbox="299 553 934 699"> <thead> <tr> <th>Obj.Idx</th> <th>Sub.Id</th> <th>Parameter</th> <th>Chosen value</th> <th>PDO Dialog</th> </tr> </thead> <tbody> <tr> <td>1400</td> <td>3</td> <td>Inhibit time</td> <td>0</td> <td>x</td> </tr> <tr> <td>1600</td> <td>1</td> <td>Number of mapped objects</td> <td>2</td> <td>x</td> </tr> <tr> <td>1600</td> <td>1</td> <td>Controlword</td> <td>1614807056</td> <td>x</td> </tr> <tr> <td>1600</td> <td>2</td> <td>Digital Speed or Current Offset</td> <td>54316200</td> <td>x</td> </tr> <tr> <td>2600</td> <td>0</td> <td>1st receive PDO select</td> <td>22</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Object</td> <td></td> <td></td> </tr> </tbody> </table> <p>After clicking on the Decimal button, enter '22' in Chosen Value for the "1st receive PDO".</p>	Obj.Idx	Sub.Idx	Parameter	Default value	Access	1000	0	Device type	131474	read only	1001	0	ErrorRegister	0	read only	1002	0	Manufacturer Status Register	(no default value)	read only	1003	0	Pre-defined Error Filed				0	Number of Errors	0	read only	1004		Number of PDOs supported			Obj.Idx	Sub.Id	Parameter	Chosen value	PDO Dialog	1400	3	Inhibit time	0	x	1600	1	Number of mapped objects	2	x	1600	1	Controlword	1614807056	x	1600	2	Digital Speed or Current Offset	54316200	x	2600	0	1st receive PDO select	22				Object		
Obj.Idx	Sub.Idx	Parameter	Default value	Access																																																																			
1000	0	Device type	131474	read only																																																																			
1001	0	ErrorRegister	0	read only																																																																			
1002	0	Manufacturer Status Register	(no default value)	read only																																																																			
1003	0	Pre-defined Error Filed																																																																					
	0	Number of Errors	0	read only																																																																			
1004		Number of PDOs supported																																																																					
Obj.Idx	Sub.Id	Parameter	Chosen value	PDO Dialog																																																																			
1400	3	Inhibit time	0	x																																																																			
1600	1	Number of mapped objects	2	x																																																																			
1600	1	Controlword	1614807056	x																																																																			
1600	2	Digital Speed or Current Offset	54316200	x																																																																			
2600	0	1st receive PDO select	22																																																																				
		Object																																																																					
5	Click on the OK button to validate your selection.																																																																						
6	Save the configuration in the "..\SyCon\Project\demo_lexium.co" file. You can now start up the PL7 or Unity Pro application.																																																																						

Configuring the Lexium

Unity Pro or PL7 Tool - Step 1 Lexium Configuration

The following procedures apply if you are using Unity Pro or PL7.

Step	Action
1	<p>In the TSX CPP 100-110 module configuration screen</p>  <p>Click on the Select Database button to import the *.co file in the Unity Pro or PL7 (..\SyCon\Project\demo_lexium.co).</p>
2	<p>The default values are maintained, and the I/O variables are configured in the following words:</p> <ul style="list-style-type: none"> ● Inputs: from %MW0 to %MW31 (not in use as no transmit PDO is configured), ● Outputs: from %MW32 to %MW63.

**Unity Pro or PL7
Tool - Step 2
Drive control
using the debug
screen**

Use the debug screen to control the servo-drive.

The screenshot shows the following data:

CANopen slave status			
Addr.	Device ready.	Act.	Life T.
0002	LEXIUM17D	1	600

CANopen slave data		
Inputs		
Parameter	Symbol	Value

Output value: OK Base: Bin Hex Dec

Outputs		
Parameter	Symbol	Value
%MW32		0
%MW33		0
%MW34		0

Information on the slave
Node 2: Status=08h, AddInfo=0002h, Profile=402, ...

Request to transmit
Enter request: Response received:

Total	Nb. Input	Nb. Output
Slave number	%MWs	%MWs
0001	0	3

The debug screen shows that the CANopen communications with the Lexium servo-drive are active (ex. "Status=08h" in the **Slave data**) field.

Change the output variable values to control the drive:

- %MW32: direct access to the drive operating modes (control word),
- %MD33: direct access to current or setpoint speed.

