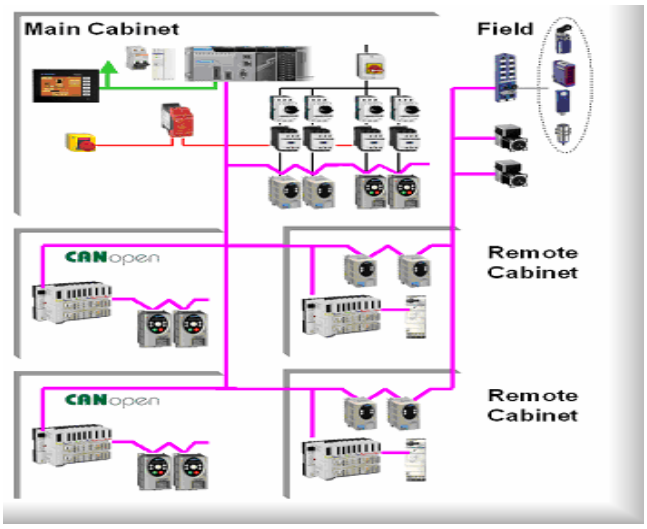
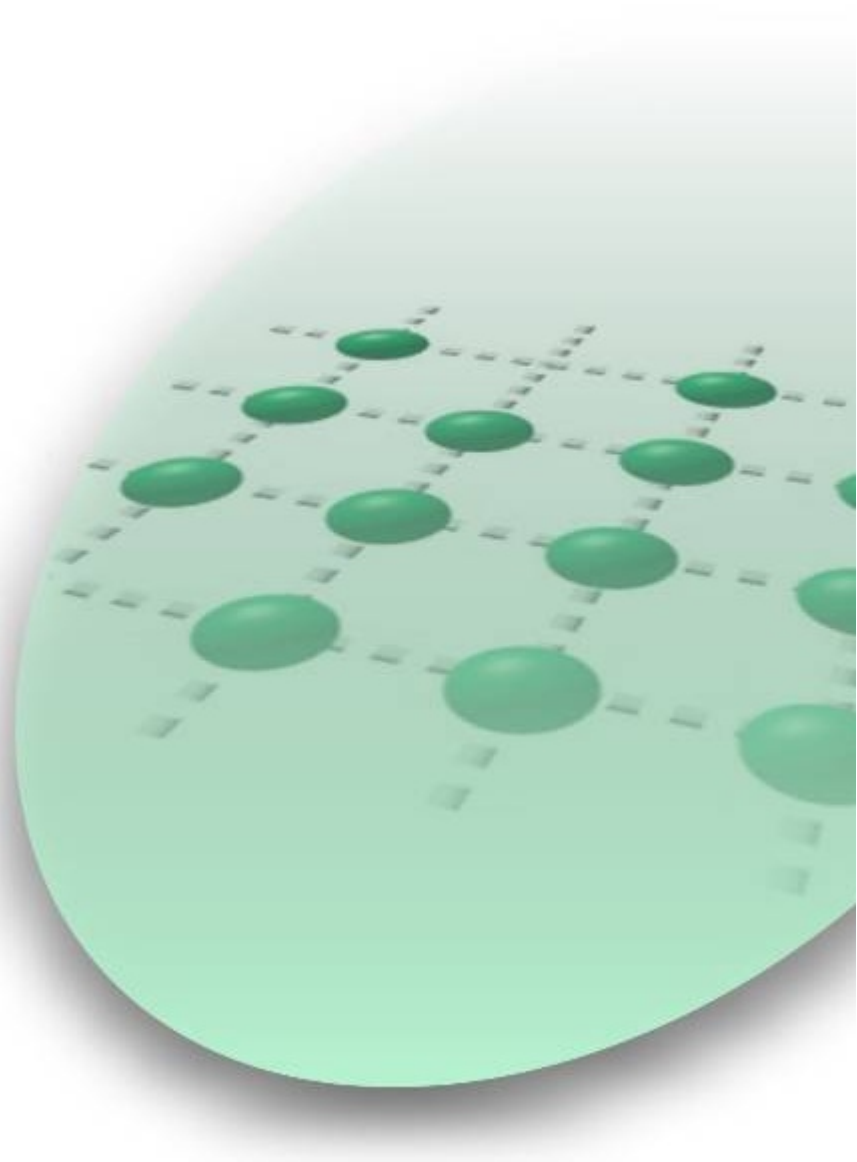
Modicon M340, CANopen, Altivar, Lexium, IclA,TeSysU and Advantys

System User Guide

[source code]





Preferred Implementation:
Distributed CANopen Performance

33004183.00

Merlin Gerin
Square D
Telemecanique

Schneider Electric
Building a New Electric World

JUL 2007

Contents

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.

Abbreviations

Word / Expression	Signification
AC	Alternating Current
Advantys	SE product name for a family of I/O modules
Altivar (ATV)	SE product name for a family of VSDs
CANopen	Name for a communications machine bus system
CB	Circuit Breaker
CoDeSys	Hardware-independent IEC 61131-3 programming software
ConneXium	SE product name for a Family of Transparent Factory devices
DC	Direct Current
EDS	Electronic Data Sheet
E-STOP	Emergency STOP button
Harmony	SE product name for a family of switches and indicators
HMI	Human Machine Interface
I/O	Input/Output
IclA (ICLA)	SE product name for a compact drive
Lexium/Lexium05/LXM	SE product name for a family of servo-drives
M340 / Modicon M340	SE product name for a mid range PLC family
Magelis	SE product name for a family of HMI-Devices
MFB	PLCopen Motion Function Block
MB - SL	SE name for a serial Modbus communications protocol
NIM	SE product name for a Network Interface Module
Osiswitch	SE product name for a family of position switches
PC	Personal Computer
PDO	Process Data Object (CANopen)
Phaseo	SE product name for a family of power supplies
PLC	Programmable Logic Computer
PowerSuite	An SE software product for configuring drives
Premium	SE product name for a middle range family of PLCs
Preventa	SE product name for a family of safety devices
PS1131 (CoDeSys)	SE Product name for PLC programming software with CoDeSys
PS	Power Supply
RPDO	Receive Process Data Object (CANopen)
SE	Schneider Electric
SDO	Service Data Object
SyCon	SE product name of a Field bus programming software

Word / Expression	Signification
Telefast	SE product name for a series of distributed I/O devices
TesysU	SE product name for a decentralized I/O System
TPDO	Transmit Process Data Object (CANopen)
Twido	SE product name of a basic range family of PLCs
TwidoSoft	SE product name for a PLC programming software
TwidoSuite	SE product name for a PLC programming software
Unity (Pro)	SE product name for a PLC programming software
Vijeo Designer	An SE software product for programming Magelis HMI devices
VSD	Variable Speed Drive
WxHxD	Dimensions : Width, Height and Depth
XBT-L1000	An SE software product for programming Magelis HMI devices
Zelio	SE product name for a low range PLC family
ZelioSoft	SE product name for a PLC programming software

Application Source Code

Introduction

Examples of the source code and wiring diagrams used to attain the system function as described in this document can be downloaded from our website under **this** link.

The example source code is in the form of configuration, application and import files. Use the appropriate software tool to either open or import the files.

Extension	File Type	Software Tool Required
AIW	Configuration file	Advantys Configuration software
CNF	**C**onfiguratio**n** **F**ile	SyCon
CO	**C**AN**o**pen definitions file	SyCon
CSV	**C**omma **S**eparated **V**alues, Spreadsheet	Twidosoft
CTX		UnityPro
DCF	**D**evice **C**onfiguration **F**ile	Advantys
DIB	**D**evice **I**ndependent **B**itmap	SyCon
DOC	**Doc**ument file	Microsoft Word
DOP	Project File	Magelis XBTL 1000
EDS	**E**lectronic **D**ata **S**heet – Device Definition	Industrial standard
FEF	Export file	PL7
GSD	EDS file (**G**eraete **S**tamm **D**atei)	Profibus
IFX	Project file	IclA Easy
ISL	Island file, Project file	Advantys Configuration Software
PB	**P**rofi**b**us definitions file	SyCon
PDF	**P**ortable **D**ocument **F**ormat - document	Adobe Acrobat
PRO	Project file	PS1131 - CoDeSys
PS2	Export file	PowerSuite
RTF	**R**ich **T**ext **F**ile - document	Microsoft Word
SPA	Schneider Product Archive	TwidoSuite
STA	Project Archive	UnityPro
STU	Project file	UnityPro
STX	Project file	PL7
TLX	Project file	Twinline control tool
TWD	Project file	TwidoSoft
VDZ	Project file	Vijeo Designer
XEF	Export file	UnityPro
XPR	Project file	TwidoSuite
ZM2	Project file	Zeliosoft

Typical Applications

Introduction

Here you will find a list of the typical applications, and their market segments, where this system or subsystem can be applied:

Industry

- Large automated machines or plants with remote modular devices
- De-centralised automatiion systems as components to larger machines

Machines

- Packaging systems
- Textile and printing machines
- Wood and metal processing

Food/Pharmaceuticals

- Filling machines

Application	Description	Image
Metal processing	Machine for forming metal sheet.	
Woodwork	Finishing the edges of wood sheeting.	
Filling machine	As part of a larger system, this machine sorts, places and fills bottles.	

System

Introduction

The system chapter describes the architecture, the dimensions, the quantities and different types of components used within this system.

Architecture

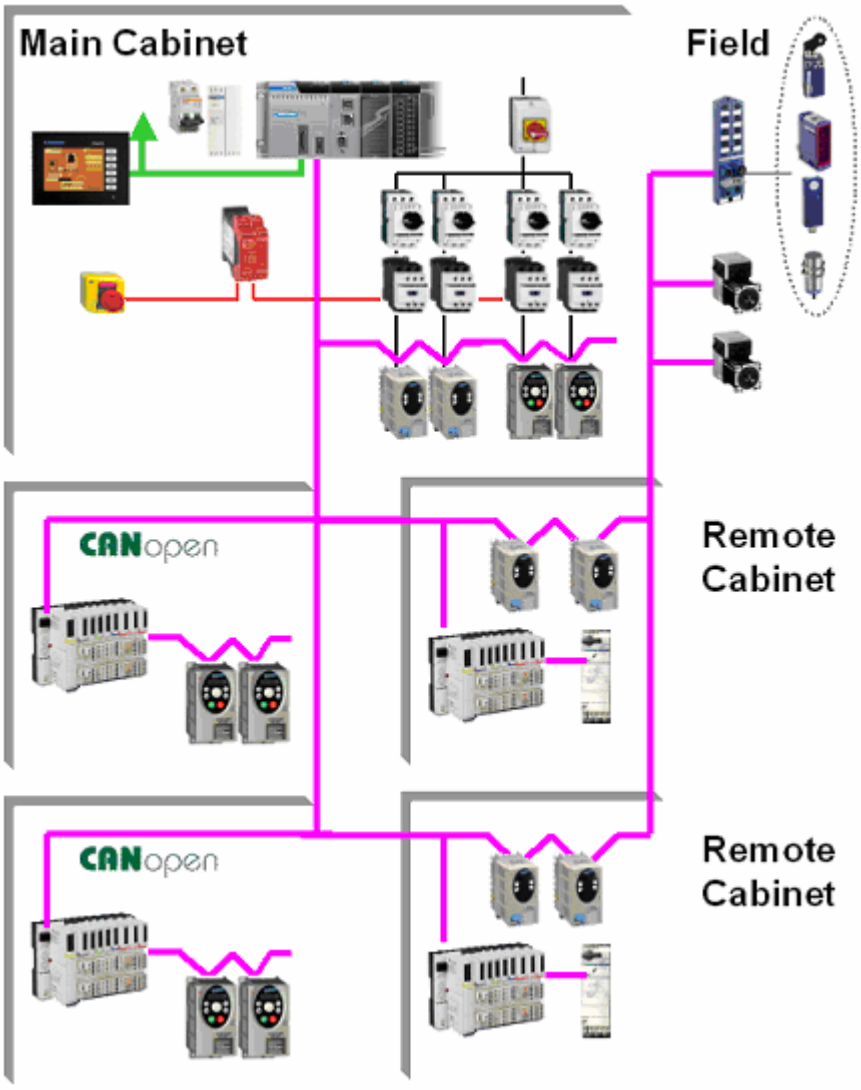
General

The control section of this application consists of a Modicon M340 PLC with remote Advantys STB I/O islands, which can be operated via a connected Magelis HMI panel at user level. The device section is implemented using Lexium 05, IcLA IFS, Altivar 31 and TeSysU, which are connected to the PLC via the CANopen bus system.

The Advantys FTB remote I/O Islands and IcLA are used in the field.

The solution illustrated below includes Preventa safety components featuring tamper-proof emergency off switches.

Layout



Components

Hardware:

- Compact master switch (NS100)
- GV2-L motor circuit breaker (short-circuit protection)
- Modicon M340 PLC with CANopen and Ethernet interface
- Magelis XBTGT HMI panel
- Preventa XPS safety module
- Lexium 05 servo drive
- Altivar ATV31 variable speed drive
- TeSysU motor starter
- TeSysK and TeSysD (LP1K and LC1D) motor contactors
- Remote I/O-Island Advantys STB
- Remote I/O-Island Advantys FTB
- Intelligent compact drive IclA IFS

Software:

- UnityPro V3.0
- Vijeo Designer V4.5
- Advantys Configuration Tool V2.50
- PowerSuite V2.40
- IclA Easy V1.104

Quantities of Components

For a complete and detailed list of components, the quantities required and the order numbers, please refer to the components list at the rear of this document.

Degree of Protection

Not all the components in this configuration are designed to withstand the same environmental conditions. Some components may need additional protection, in the form of housings, depending on the environment in which you intend to use them. For environmental details of the individual components please refer to the list in the appendix of this document and the appropriate user manual.

Technical Data

Mains voltage	400 V AC
Power requirement	~ 8 kW
Drive power rating	6x 0.37 kW , 8x 0.75kW
Motor brake	None
Connection	5x 2.5mm² (L1, L2, L3, N, PE)
Safety level	Cat. 3

Safety Notice

The standard and level of safety you apply to your application is determined by your system design and the overall extent to which your system may be a hazard to people and machinery.

As there are no moving mechanical parts in this application example, category 3 (according to EN954-1) has been selected as an optional safety level.

Whether or not the above safety category should be applied to your system should be ascertained with a proper risk analysis.

This document is not comprehensive for any systems using the given architecture and does not absolve users of their duty to uphold the safety requirements with respect to the equipment used in their systems or of compliance with either national or international safety laws and regulations

Dimensions

The dimensions of the individual devices used; PLC, Servodrives, Drives, Power supply, etc. require a housing cabinet (using double sided mounting) size of at least 800x1800x800mm (WxHxD) and four remote cabinets of size 600x100x400mm (BxHxT).

The IP67 de-centralised I/O islands and IcLA are installed on site, in the field.

The illuminated indicators/switches such as "24 V OK", "SYSTEM OFF" or "ACKNOWLEDGE EMERGENCY STOP", as well as the Emergency Stop button itself, can be built into the door of the cabinet.

Installation

Introduction

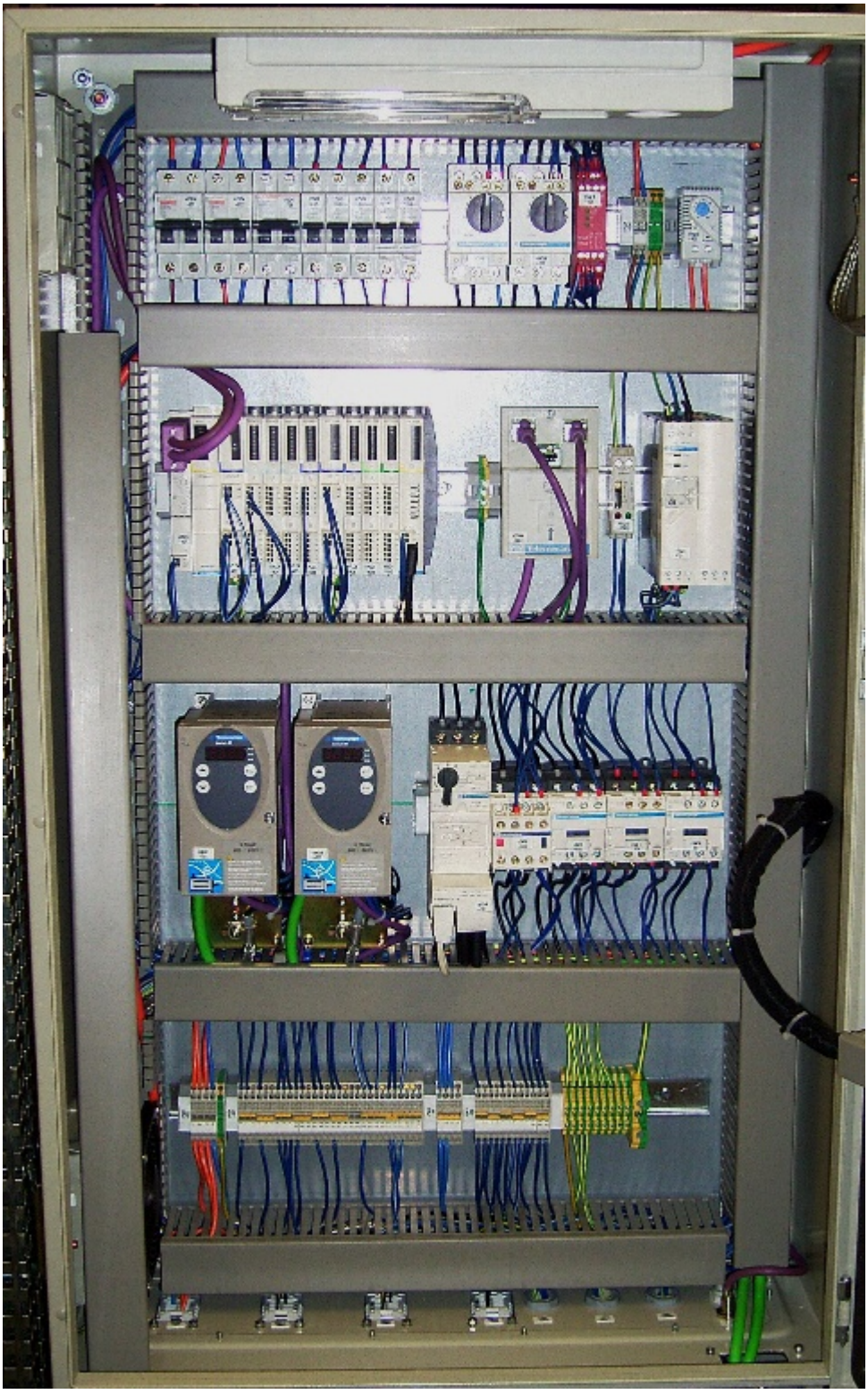
This chapter describes the steps necessary to set up the hardware and configure the software required to fulfill the described function of the application.

Assembly

Main Cabinet M340 + Magelis XBTGT + Lexium 05 + Altivar 31

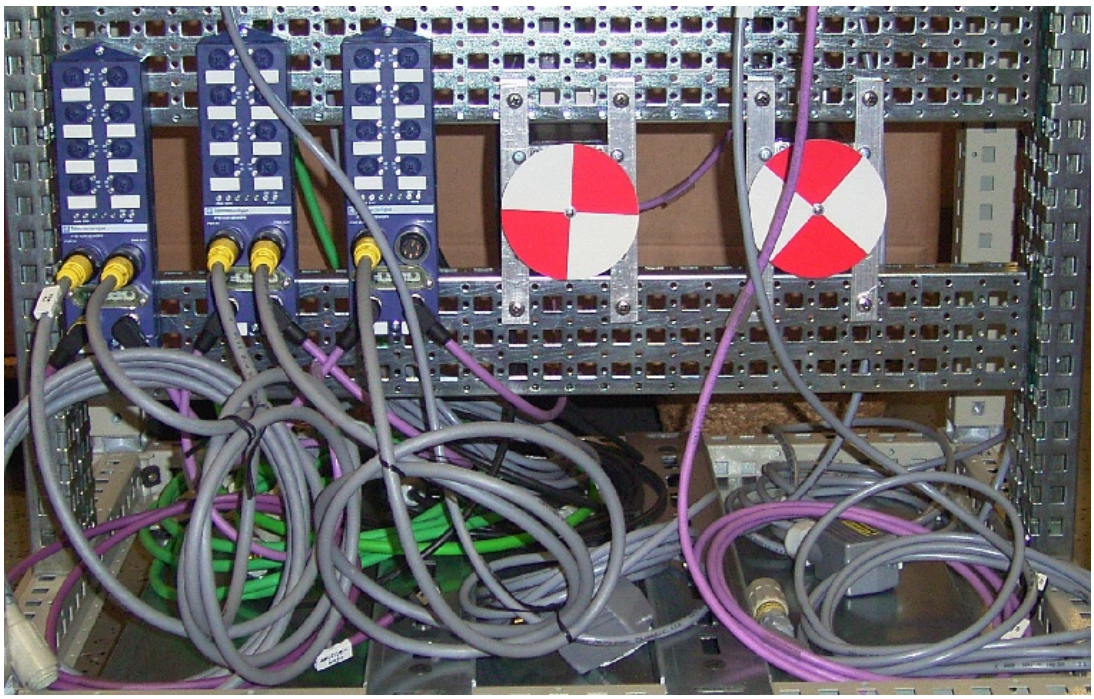
Assembly
contd.

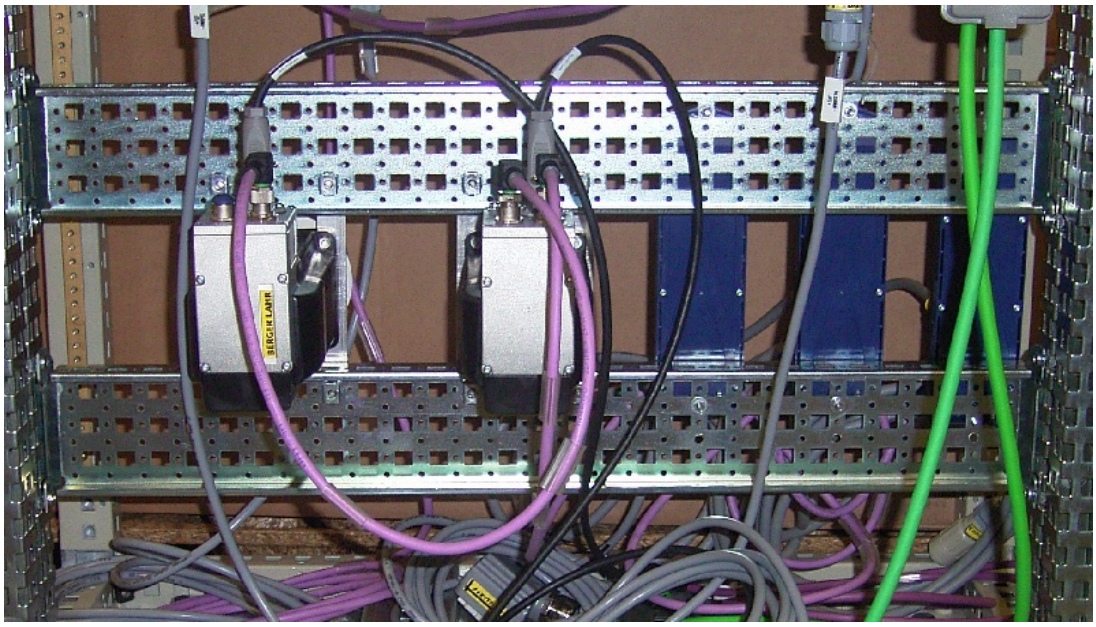
Remote Cabinet – Advantys STB, Lexium05 or Altivar 31 + TeSysU



Assembly
contd.

In Field – Advantys FTB + IclA IFS





Hardware

General

The components designed for installation in a control cabinet, i.e., the safety modules, line circuit breakers, contactors, motor circuit breakers and Advantys STB I/O modules can be mounted on a 35 mm top-hat rail.

Master switches, Modicon M340 PLC, Indicator lamps, the Phaseo power supply unit, Altivar variable speed drives, Lexium 05 servo drives and the IcLA IFS intelligent compact drives are screwed directly onto the mounting plate. Alternavtively the Altivar 31 and Lexium05 can be mounted on a top-hat rail if an adapter is used.

The Emergency Stop button, door safety switches and the pushbutton housing for display and acknowledgement indicators are designed for backplane assembly in the field. All switches (except the door safety switch) can also be installed directly in a control cabinet (e.g., in a cabinet door) without the need for their enclosing housings

There are two options for installing XB5 pushbuttons or indicator lamps: These pushbuttons or switches can be installed either in a 22 mm hole, e.g., drilled into the front door of the control cabinet, or in an XALD-type housing suitable for up to 5 pushbuttons or indicator lamps. The XALD pushbutton housing is designed for backplane assembly or direct wall mounting

The Magelis operator and display terminals require a cut-out in the front of the housing so that they can be secured to the housing wall using brackets/spring clamps

The IP67 I/O modules are mounted outside the cabinet.

400 V/3-phase AC wiring for the drive circuitry (LXM05, ATV31, TeSysU).

240 V AC wiring for the circuitry of the IcLA supplies.

240V~ or 400V /3~ wiring for the power supply.

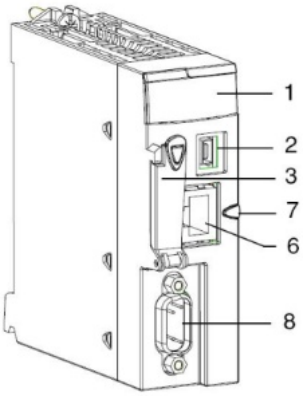
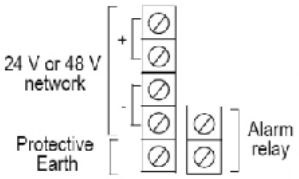
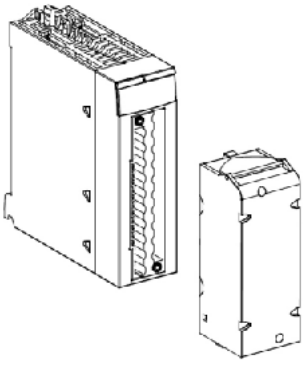
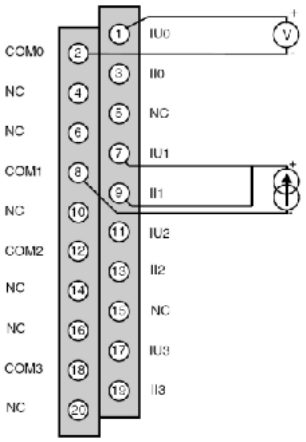
24 V DC wiring for control circuits and the PLC power supply, operator and display terminals, I/O modules, the HMI and the compact drives.

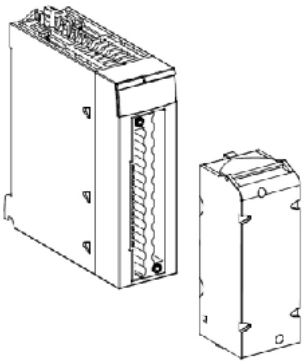
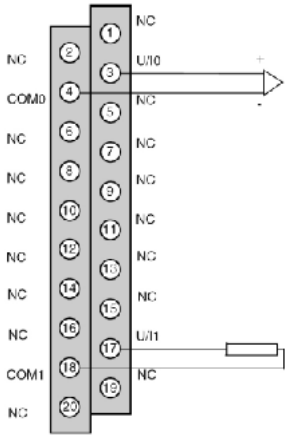
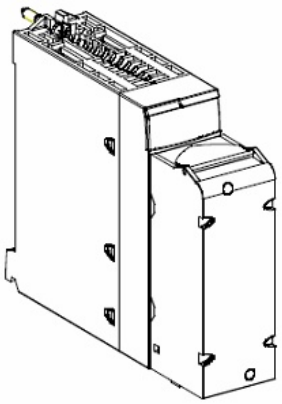
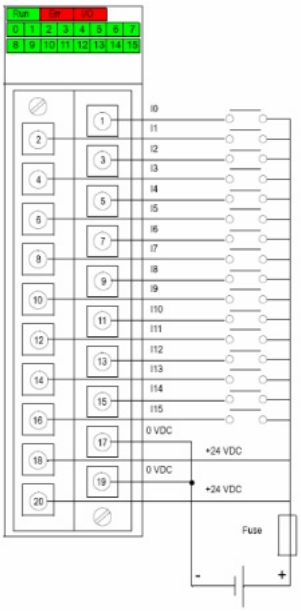
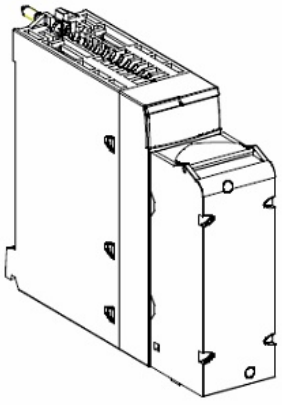
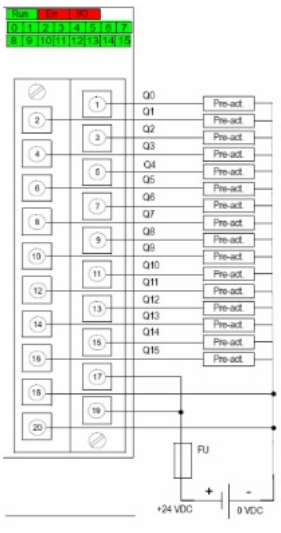
The individual components must be interconnected in accordance with the detailed circuit diagram in order to ensure that they function correctly.

CANopen cables are installed for the communication link between the PLC and the devices inside the control cabinet.

Further CANopen bus cabling is required between the PLC and each remote STB I/O island, the IP67 I/O modules,the Lexium05 servo drives and the ATV31 drives.

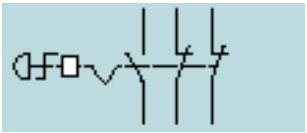
Hardware

Modicon M340 **CPU includiing CANopen and Ethernet** **BMXP342030**		
	1 Display panel **2 USB port** **3 Memory card protective cap** **6 Ethernet connection** **7 Ethernet identification ring (green)** **8 CANopen connection**	
Modicon M340 **Power supply** **BMXCPS3020**		
Modicon M340 **Analog I/O modules 4 inputs** **BMXAMI0410**		

Modicon M340 **Analog** **I/O modules** **2 outputs** **BMXAMO0210**		
Modicon M340 **Digital** **I/O Modules** **16 Inputs** **BMXDDI1602**		
Modicon M340 **Digital** **I/O Modules** **16 Outputs** **BMXDDO1602**		

EMERGENCY STOP button (tamper-proof)

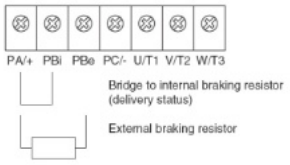
XALK178G



Lexium05

Single-phase servo drive

LXM05AD10M2



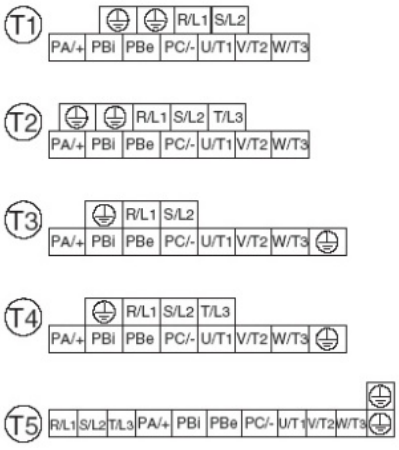
Power connections	Description
PE	Earth connection (protective earth)
R/L1, S/L2/N	Mains connection, single phase devices
R/L1, S/L2, T/L3	Mains connection, 3-phase devices
PA/+	DC bus
PBi	Braking resistor internal
PBe	Braking resistor external
PC/-	DC bus
U/T1, V/T2, W/T3	Motor connections

Lexium05

Single-phase servo drive

LXM05AD10M2

Power Terminals see T4



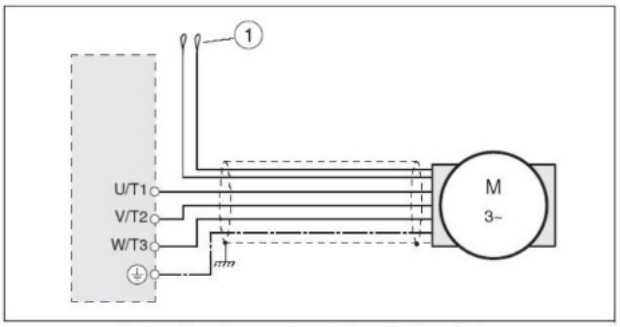
LXM05•...	
D10F1	(T1)
D10M2	(T1)
D10M3X	(T2)
D14N4	(T4)
D17F1	(T3)
D17M2	(T4)
D17M3X	(T4)
D22N4	(T4)
D28F1	(T3)
D28M2	(T4)
D34N4	(T4)
D42M3X	(T4)
D57N4	(T5)

Lexium05

Single-phase servo drive

Motor cable connection terminal
(cable length: 3 m)

VW3M5101R30



Motor wiring diagram, here without holding brake

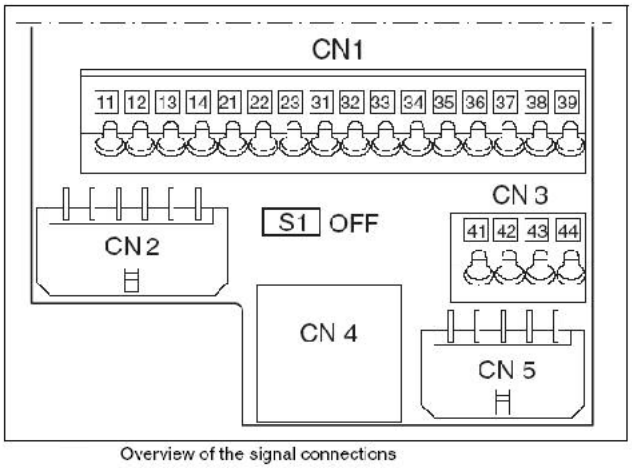
Terminal	Description	Colour
U/T1	Motor lead	black L1 (BK)
V/T2	Motor lead	black L2 (BK)
W/T3	Motor lead	black L3 (BK)
PE	Protective conductor	green/yellow (GN/YE)
(1)	Holding brake connection cable For motors with holding brake	white (WH), grey (GR)

Lexium05

Single-phase servo drive

LXM05AD10M2

Signal Terminals



Overview of the signal connections

Connection/ switch	Assignments
CN1	Analogue inputs ±10V, pin 11 to 14
	CANopen, pin 21-23
	Digital inputs/outputs, pin 31-39
CN2	Motor encoder (Hiperface Sensor)
CN3	24V PELV controller supply voltage
CN4	PC, peripheral operating terminal, Modbus, CANopen (RJ45)
CN5	ESIM (A/B/I out), PULSE/DIR in, encoder signals A/B/I in [1]
S1	Switch for fieldbus terminating resistor

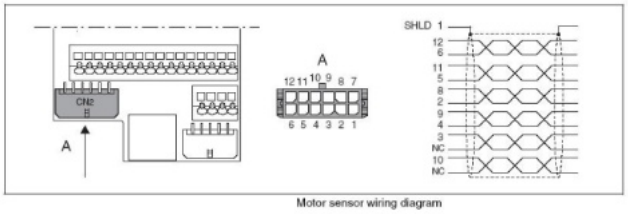
1) depending on the "First Setup"

Lexium05

Single-phase servo drive

Encoder cable connection terminal (cable length: 3 m)

VW3M8101R30



Motor sensor wiring diagram

Pin	Signal	Motor, pin	Colour [1]	Pair	Description	I/O
1	SHLD				Shielding braid	
12	SIN	8	white	1	Sine signal	E
6	REFSIN	4	brown	1	Reference for sine signal, 2.5 V	A
11	COS	9	green	2	Cosine signal	E
5	REFCOS	5	yellow	2	Reference for cosine signal, 2.5V	A
8	Data	6	grey	3	Receive and transmit data	I/O
2	Data	7	pink	3	Receive and transmit data, inverted	I/O
10	ENC_0V	11	blue	4	sensor reference potential (encoder) (0.5mm²)	A
			red	4	not assigned (0.5mm²)	
3	TMOT_0V	1	black	5	Reference potential for T_MOT	-
			purple	5	not assigned	
9	T_MOT	2	grey/pink	6	temperature sensor PTC	E
4	ENC+10V_OUT	10	red/blue	6	10 V_{DC} power supply for sensor, max. 150 mA	A
7	n.c.				not assigned	

1) Colour data is based on the prefabricated cables

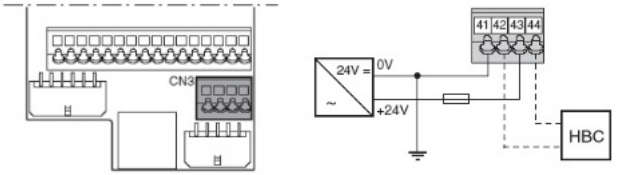
Lexium05

Single-phase servo drive

LXM05AD10M2

Control power supply

HBC = Holding brake control



Pin	Signal	Description
41	0VDC	Reference potential for 24V voltage
42	0VDC	Reference potential for 24V voltage
43	+24 VDC	24V controller supply voltage
44	+24 VDC	24V controller supply voltage

Lexium05

Single-phase servo drive

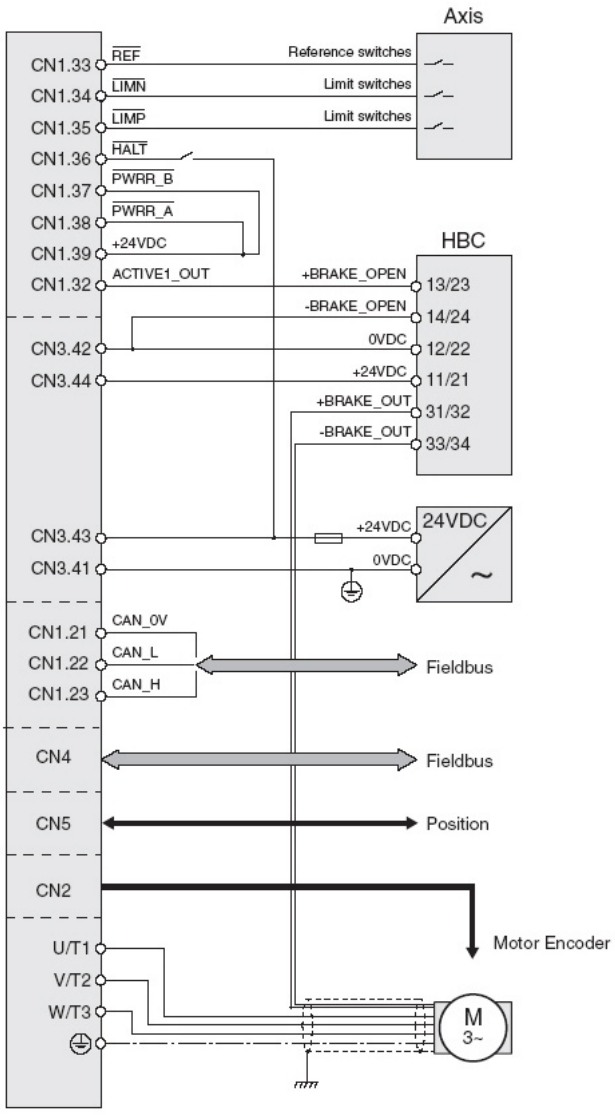
LXM05AD10M2

Terminals 33-39 must be connected.

The motor brake (if present) must be connected via a holding brake control (HBC)

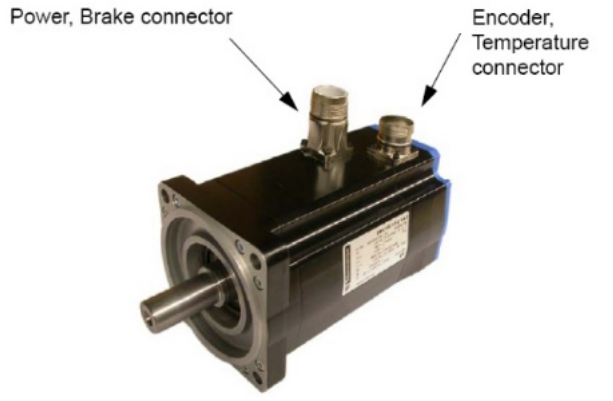
Fieldbus wiring:

The RJ45 connector (CN4) is used for CANopen



Servo motor

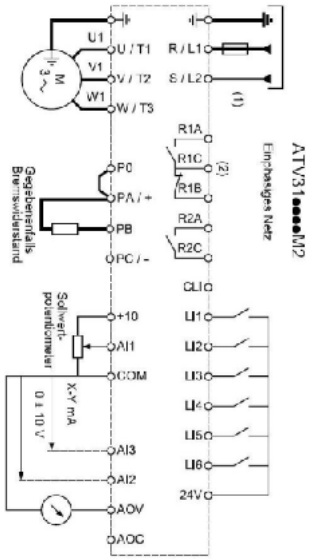
BSH0702P02A2A



Altivar 31

Variable Speed Drive

ATV31H018M2



Terminal	Function
⏚	Protective ground connection terminal
R/L1 S/L2 T/L3	Power supply
PO	DC bus + polarity
PA/+	Output to braking resistor (+ polarity)
PB	Output to braking resistor
PC/-	DC bus - polarity
U/T1 V/T2 W/T3	Outputs to the motor

Berger Lahr IcLA

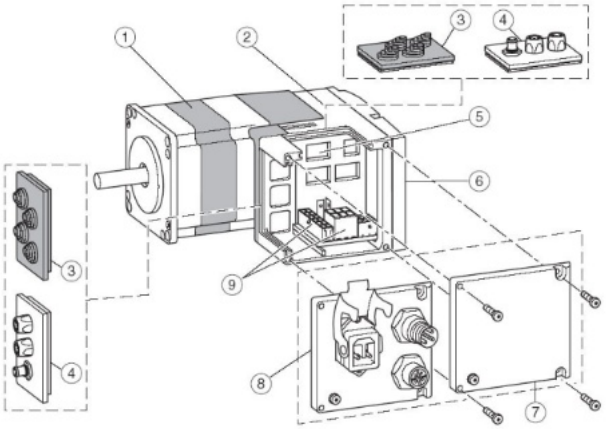
Intelligent Compact Drive

IFS93/2 CANISDS/3D-I54/O-001RPP41

Berger Lahr IcLA

Intelligent Compact Drive

Assembly

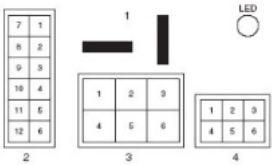


1 EC-Motor
2 Electronics housing
3 Socket cabling
4 I/O socket with Industrial connector
5 Adjustment DIP-Switch
6 Electronics housing cover, do not remove
7 socket cover – remove for installation
8 cover for Industrial connector for DC supply and Field bus IN/OUT
9 Electrical Interface

Berger Lahr IcLA

Intelligent Compact Drive

Connections



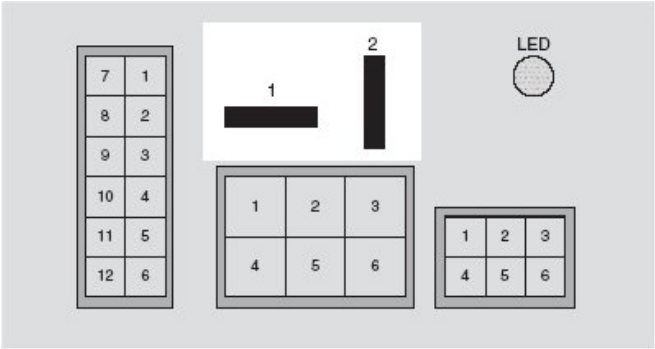
1 Power Supply
2 Field bus interface Profibus-DP
3 Field bus interface CAN or RS485
4 24V-Signal interface

Fieldbus interface
upper rotary switch
lower rotary switch
Supply interface
LEDs
fixing screws
Card slot for memory module
Cover for config.Port

Berger Lahr IcLA

Intelligent Compact Drive

Power supply

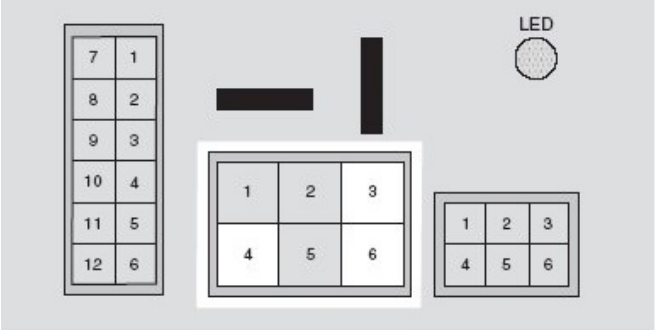


PIN	Signal	Use
1	VDC	Power supply 24/35 VDC
2	GND	GND for power supply, internal connaction GND fron CAN, RS485 and 24V-Signal interface

Berger Lahr IcLA

Intelligent Compact Drive

Connection CANopen Bus

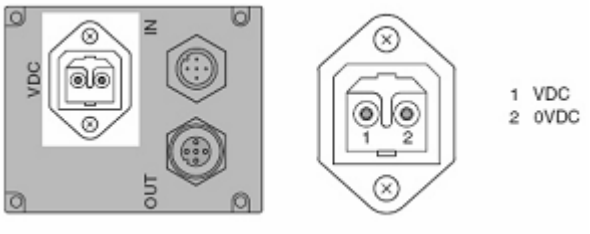


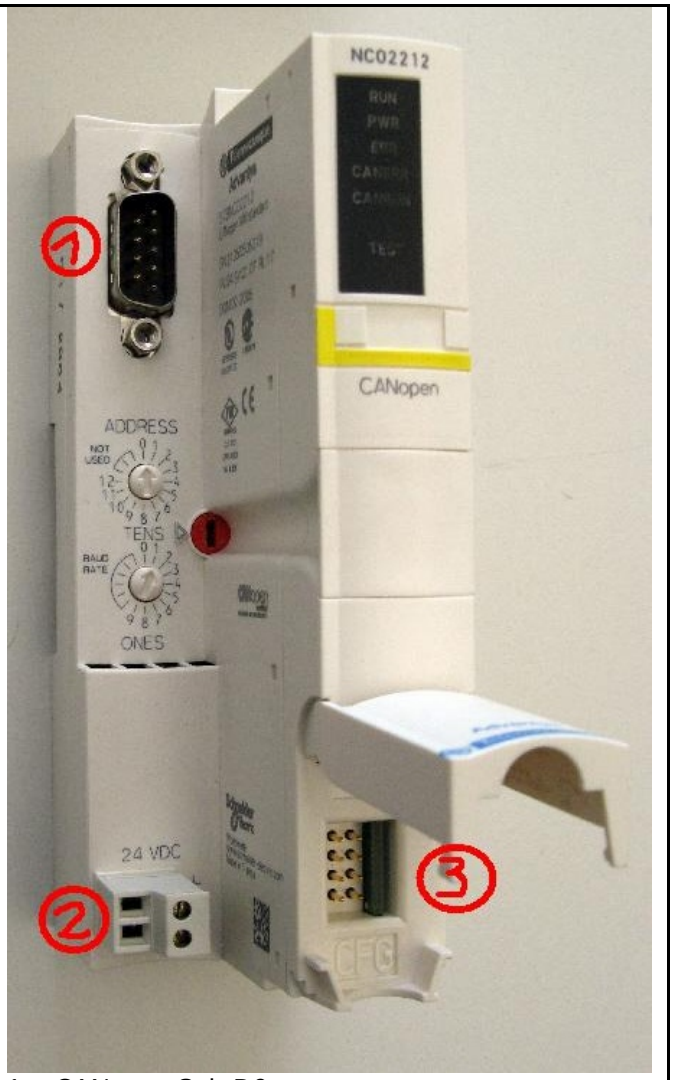
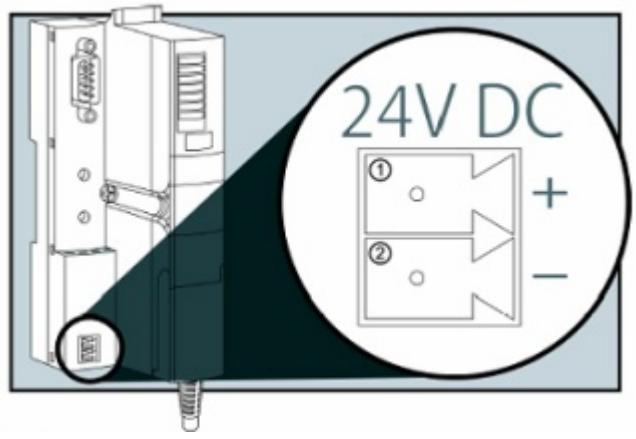
PIN	Signal	Bedeutung
3.	CAN_H	CAN-Signal interface
6.	CAN_L	CAN-Signal interface
4.	GND	internal GND from power supply

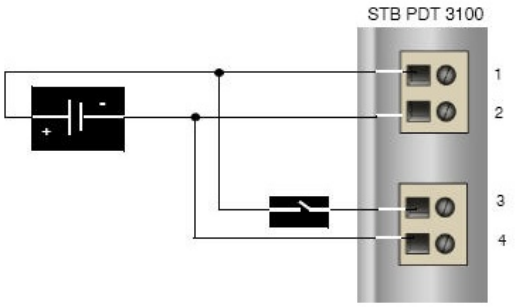
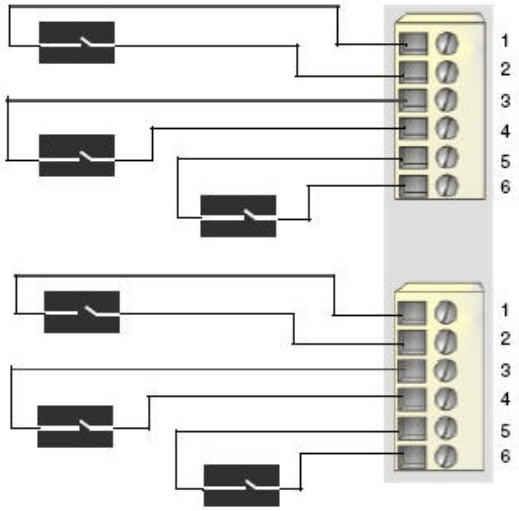
Berger Lahr IclA

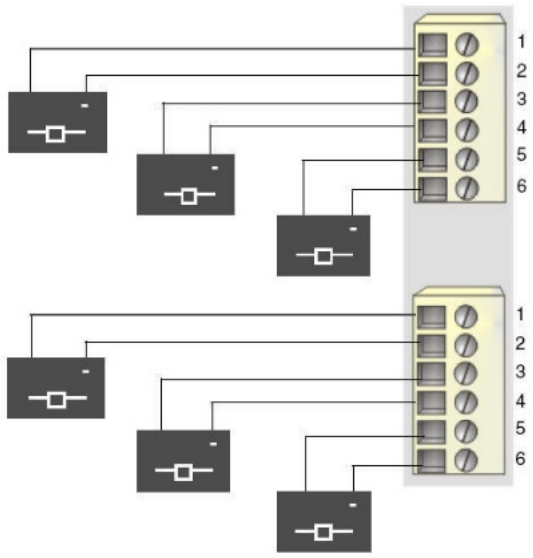
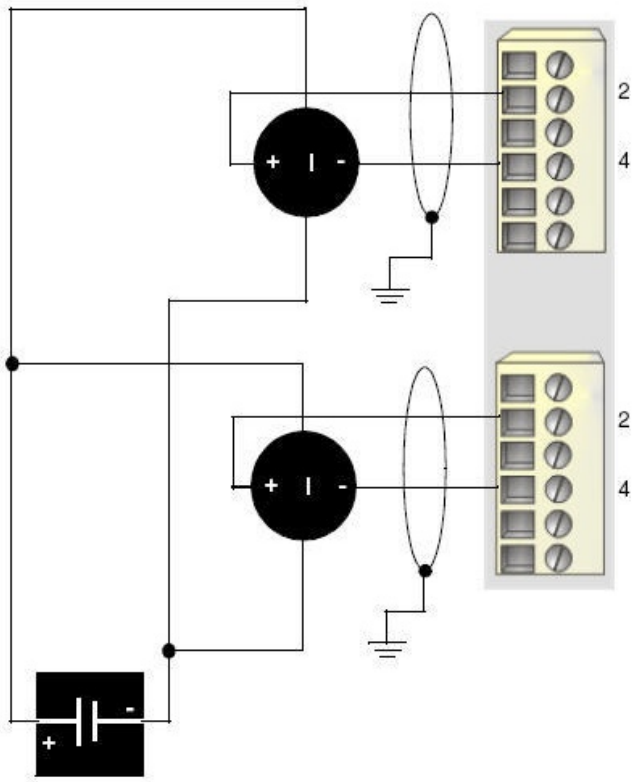
Intelligent Compact Drive

Connection 24VDC



Advantys STB **CANopen NIM** **STBNCO2212**	 1. CANopen Sub-D9 2. 24vDC +/- 3. CFG-Port HE10 

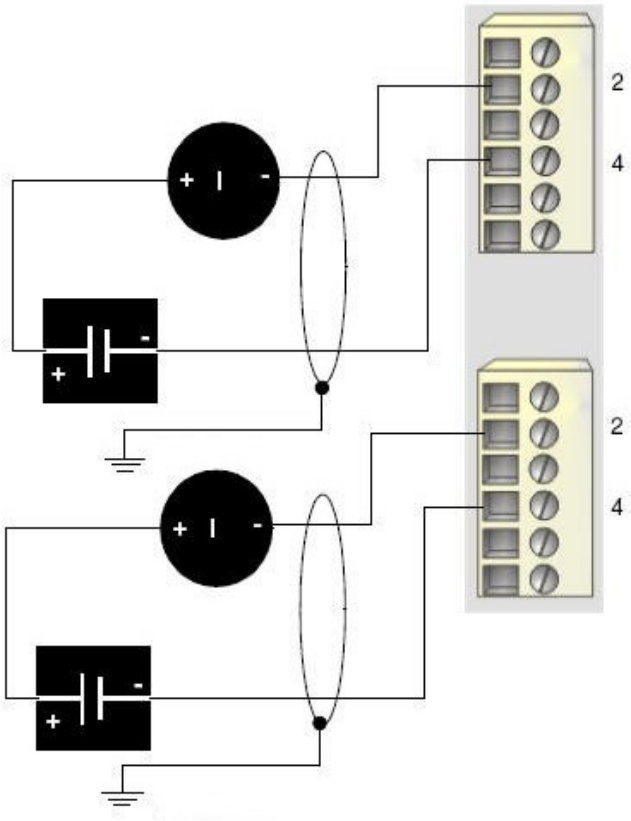
Advantys STB **Field power supply** **STBPDT3100**	 1. +24 VDC Sensor bus power 2. - 24 VDC return cable from power supply 3. +24 VDC Actuator bus power 4. –24 VDC return cable from actuator supply
Advantys STB **Digital 24VDC Input module** **STBDDI3610**	 1. +24 VDC to Sensor 1 (top) and Sensor 4 (bottom) 2. Input from Sensor 1 (top) and Sensor 4 (bottom) 3. +24 VDC to Sensor 2 (top) and Sensor 5 (bottom) 4. Input from Sensor 2 (top) and Sensor 5 (bottom) 5. +24 VDC to Sensor 3 (top) and Sensor 6 (bottom) 6. Input from Sensor 3 (top) and Sensor 6 (bottom)

Advantys STB **Digital 24VDC Output module** **STBDDO3600**	 1. Ouput to Actuator 1 (top) and Actuator 4 (bottom) 2. field supply return from Actuator 1 & 4 3. Ouput to Actuator 2 (top) and Actuator 5 (bottom) 4. field supply return from Actuator 2 & 5 5. Ouput to Actuator 3 (top) and Actuator 6 (bottom) 6. field supply return from Actuator 3 & 6
Advantys STB **Analog Power Input module** **STBACI1230**	 2. Inputs from Sensor 1 (top) and Sensor 2 (bottom) 4. return cable from input 1(top) and 2 (bottom)

Advantys STB

Analog Power Output module

STBACO1210



2. Ouputs to Actuator 1 (top) and Actuator 2 (bottom)
4. Field supply return from Actuator 1 (top) and Actuator 2 (bottom)

Advantys STB

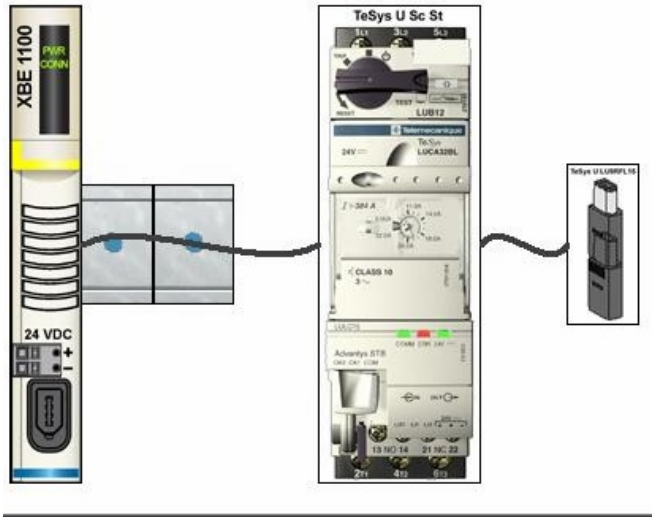
Segment terminal EOS

STBXBE1100

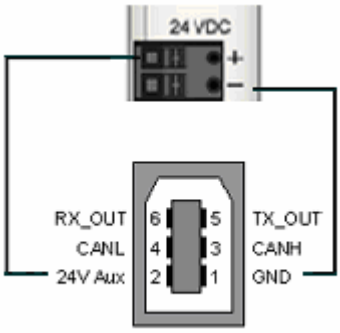
Advantys STB EOS/BOS to TeSysU LULC15 Cable LU9RDD10

TeSysU LULC15 to TeSysU LULC15 Cable LU9RCD10

TeSysU LU9RFL15 Terminal Switch



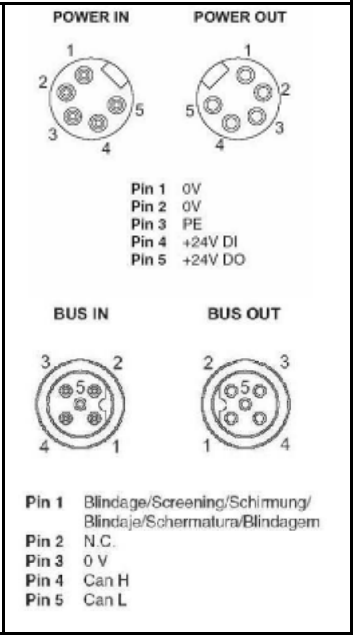
STBXBE1100 Interfaces 24VDC and CANopen

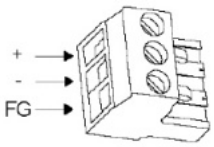
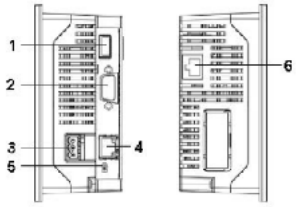
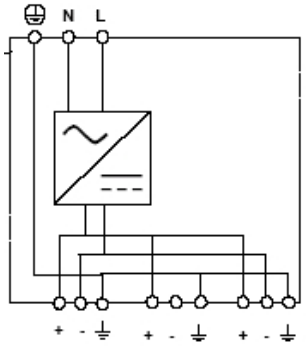
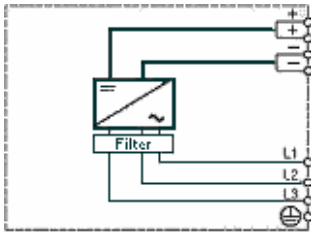


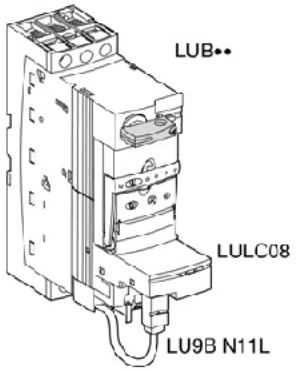
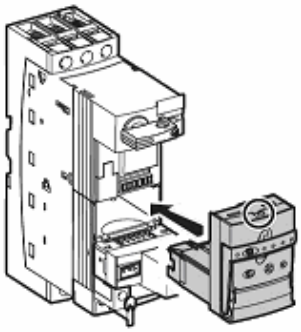
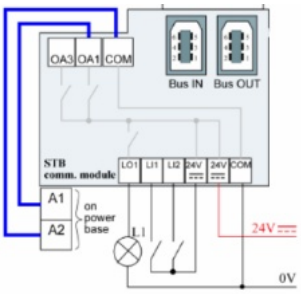
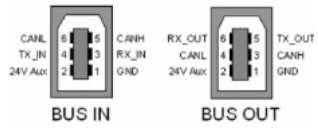
Advantys FTB

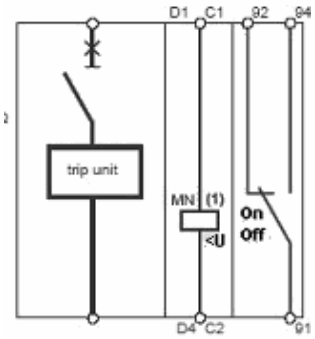
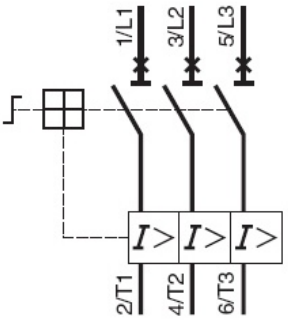
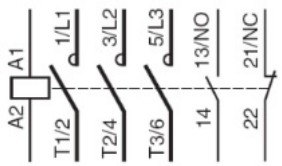
I/O-Module IP67

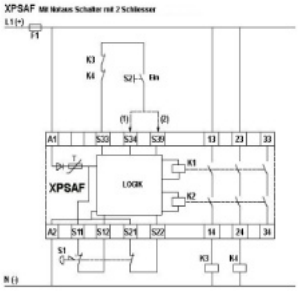
FTB1CN12E04SPO

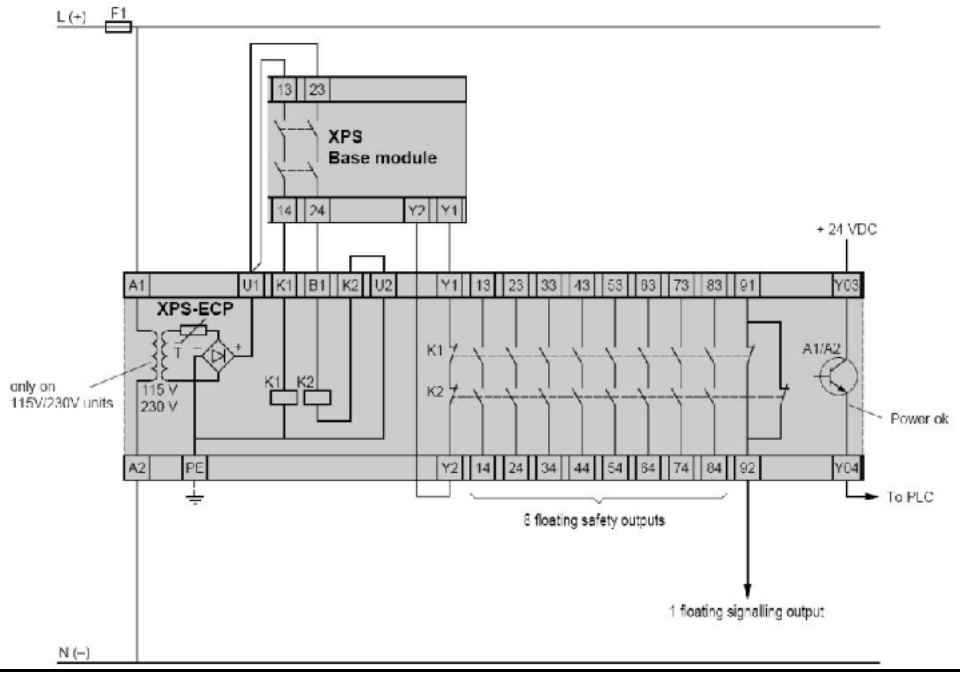


Magelis HMI **XBTGT2330**	  + 24V DC - 0 V FG Ground	 1 USB port (USB 1.1) 2 COM1 serial port (SubD, 9-pin) 3 Current input terminal block (see image on left) 4 COM2 serial port (RJ45) 5 Polarity selector switch 6 Ethernet interface
Phaseo power supply **ABL7RE2410**		ABL-7RE●●●● 
Phaseo power supply **ABL7UPS24200**		

TeSysU **motor starter** Power base **LUB32** Communication module CANopen **LULC15** Coil wiring kit **LU9B N11C**		
TeSysU **"Advanced" trip unit (0.35 A – 1.40 A)** **LUCB1XBL**		
TeSysU – AdvantysSTB **CANopen communication module** **LULC15** 24V DC Power supply CANopen Bus interface Bus terminal plug LU9RFL15		   

TeSysU **wiring set** Wiring kit for coil **LU9BN11C**		
Master Switch **Compact NSC100**		
Motor circuit breaker **GV2Lxx**		
Motor contactor **LC1Dxx**		

Preventa Safety module **XPSAF5130**		
Preventa Expansion module **XPSECP5131**		



Software

General

Software tools are used for two reasons: first, for programming the M340 PLC and configuring the CANopen communication, and second, for generating monitoring displays.

The PLC is programmed using the UnityPro programming tool.

The HMI application on the XBTGT 2320 Magelis display terminal is created using Vijeo Designer software.

The de-centralised Advantys STB I/O islands are configured using the Advantys Configuration Tool.

The Lexium 05 servo drives and Altivar 31 variable speed drives can be parameterized via the front operator panel. However, using the PowerSuite software is much easier.

The parameters can be saved and archived using UniLink and PowerSuite. This is extremely useful as it means that parameters can be restored rapidly whenever service tasks need to be performed. The software can also help you to optimize the parameters online.

The software Icla Easy is used to configure the IcLA IFS intelligent compact drives.

To use the software packages, your PC must have the appropriate Microsoft Windows operating system installed:

- Windows 2000 *or*
- Windows XP

Note: The description in this documentation is based on English-language versions of the operating systems and the installed software.

The software tools have the following default install paths:

- UnityPro

  C:\Program Files \Schneider Electric\Unity Pro

- Vijeo-Designer

  C:\Program Files \Schneider Electric\VijeoDesigner

- Advantys Configuration Tool

  C:\Program Files \Schneider Electric\Advantys

- PowerSuite

  C:\Program Files \Schneider Electric\PowerSuite

- IcLA Easy

  C:\Program Files \Berger Lahr\IclA Easy

Communication

General

Between devices, the methods of communication are :

- CANopen
- Ethernet

The machine bus enabling communication between the PLC and fieldbus devices is implemented in the form of **CANopen.** Devices linked via CANopen are the Advantys STB, Lexium, Altivar and TeSysU motor starter.

Ethernet is used for data exchange between the PLC (Modicon M340) and remote HMI (Magelis XBTGT). In addition, the applications are transferred from the PC to the PLC and HMI via Ethernet.

Further, connection cables are also required between the PC and the individual devices for programming and parameterization purposes.

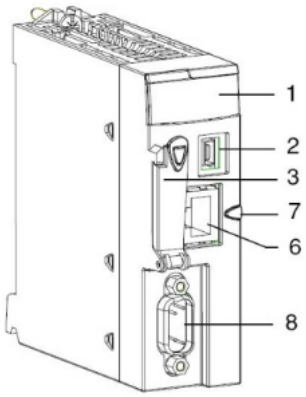
Modicon M340

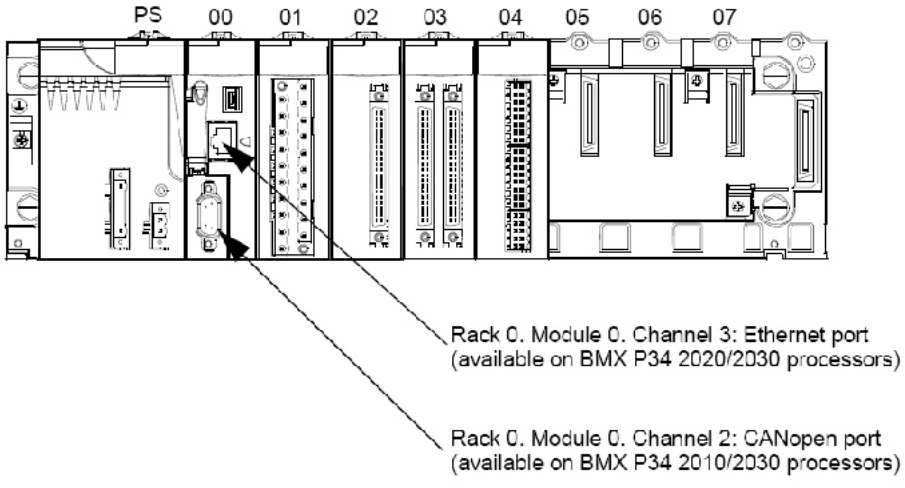
CPU including CANopen and Ethernet

BMX P34 2030

Ports

2 USB
6 Ethernet
8 CANopen





The **MAC address** is located on the front panel of the processor, below the display panel.

<table>
<tr>
<td>Modicon M340
CPU

USB PC connection cable

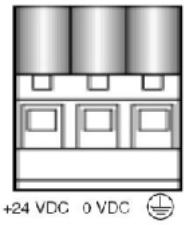
BMXXCAUSB018
(1.8 m)

BMXXCAUSB045
(4.5 m)</td>
<td> 
BMX XCA USB0●●

For transferring the Unity application from the PC to the PLC.

Alternatively, the Ethernet port can be used for connection purposes.</td>
</tr>
<tr>
<td>5-port ConneXium Ethernet switch

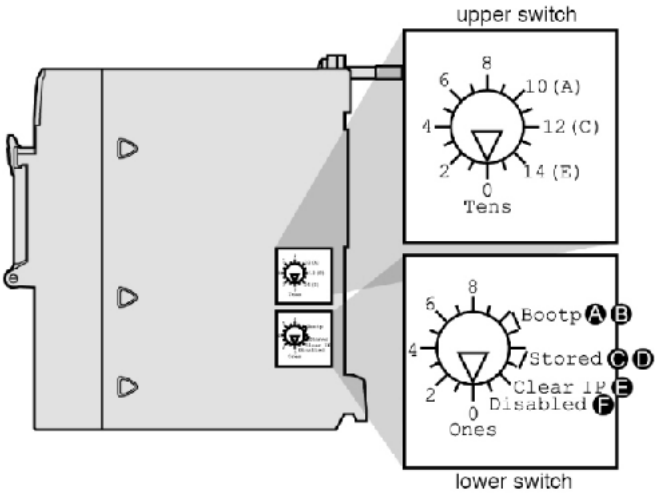
499NES25100</td>
<td> 



<table>
<tr><th>Pin Position</th><th>Description</th></tr>
<tr><td>Left</td><td>+24 VDC</td></tr>
<tr><td>Center</td><td>0 VDC</td></tr>
<tr><td>Right</td><td>Protective Earth (PE)</td></tr>
</table></td>
</tr>
<tr>
<td>Modicon M340

CPU including CANopen and Ethernet

BMXP342030

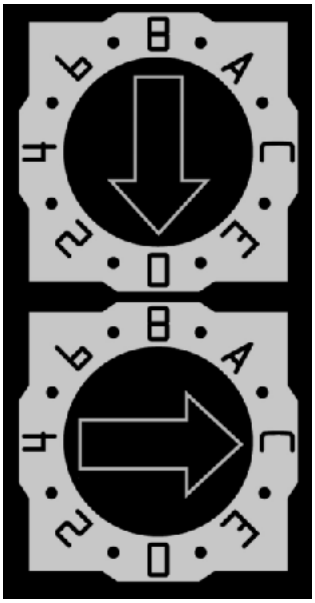
As both rotary switches are located on the rear of the module, assigning the IP address is really easy.</td>
<td> 
</td>
</tr>
</table>

For the purpose of this application, the IP address configured (**stored**) in the Unity project is used.

The following settings must also be made on the rotary switches:

Upper: 0
In this operating mode, the switch is not evaluated.

Lower: C or D
Use the configured (stored) IP address



Upper Switch
0 to 9: Tens value for the device name (0, 10, 20 . . . 90)
10(A) to 15(F): Tens value for the device name (100, 110, 120 . . . 150)

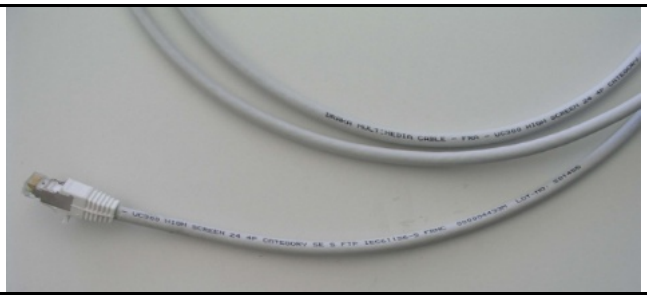
Lower Switch
0 to 9: Ones value for the device name (0, 1, 2 . . . 9)
Bootp: Set the switch to A or B to receive an IP address from a BOOTP server.
Stored: Set the switch to C or D to use the application's configured (stored) parameters.
Clear IP: Set the switch to E to use the default IP parameters.
Disabled: Set the switch to F to disable communications.

Magelis XBTGT2330

Ethernet port for data exchange with the PLC.

ConneXium Ethernet cable

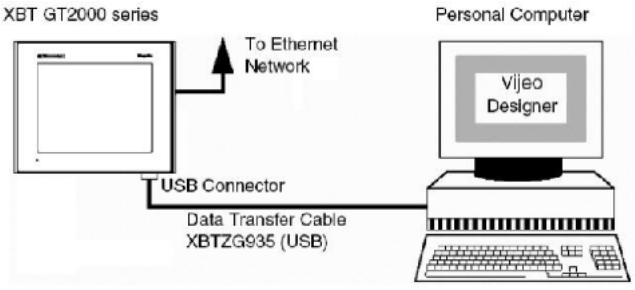
490NTW0000x



Magelis HMI

USB
PC connection cable

XBTZG935

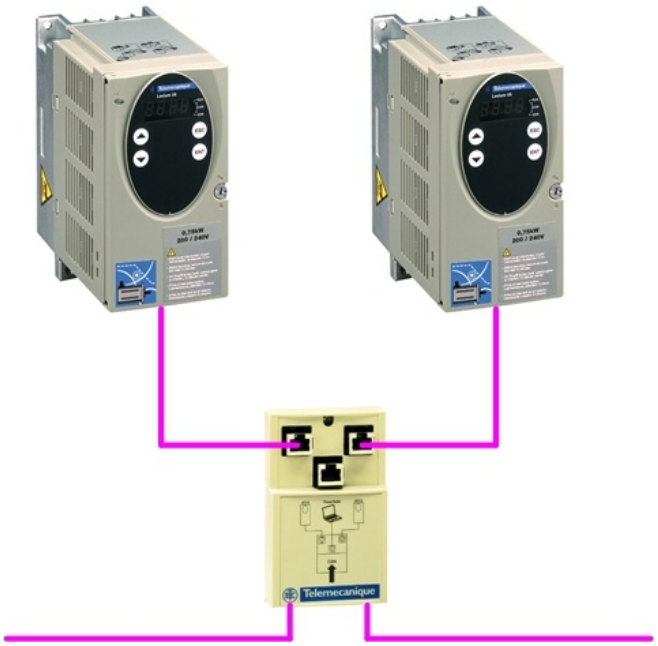


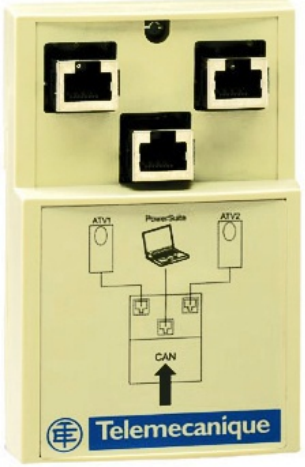
For transferring the Vijeo Designer configuration from the PC to the HMI.

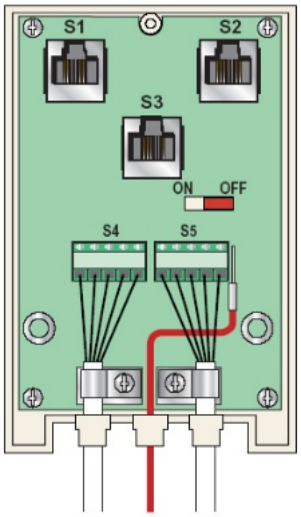
Alternatively, configuration can be performed via the Ethernet port.

CANopen junction box

VW3CANTAP2





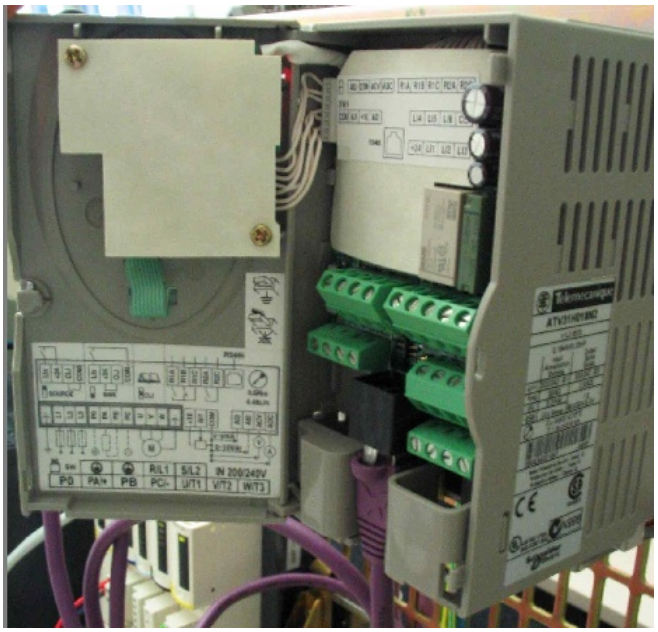


For the purpose of this application, the sliding switch must be set to **OFF**.

If, unlike in this application, there is no outgoing CANopen bus, the line terminator must be activated (i.e., the sliding switch must be set to ON).

Pin	Signal	Wire colour	Description
1	GND	Black	Ground
2	CAN_L	Blue	CAN_L bus line
3	SHLD	(bare cableshield)	Optional shield
4	CAN_H	White	CAN_H bus line
5	(V+)	Red	Optional supply

<table>
<tr><td>ATV31

Modbus- and CANopen Connection (RJ45)</td><td colspan="2">  </td></tr>
<tr><td>CANopen

RJ45 pre-assembled connection cable

VW3CANCARRxx

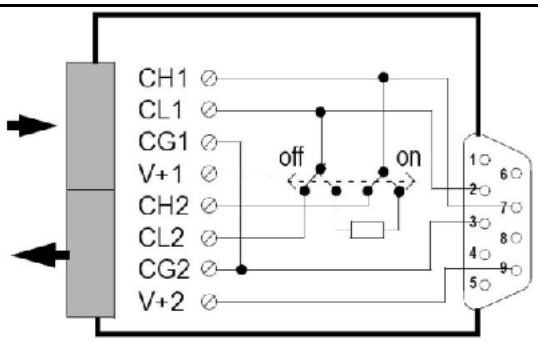
This cable is used to connect the junction box to the Lexium 05.</td><td> 
VW3CANCARR1
(Length: 1.0 m)</td><td> 
VW3CANCARR03
(Length: 0.3 m)</td></tr>
<tr><td>CANopen connector

VW3CANKCDF90T,
VW3CANKCDF90TP
or
VW3CANKCDF180T

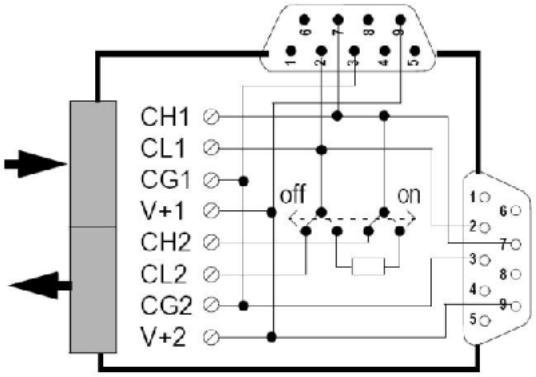
This connector is used for the link to the CANopen node.</td><td>  </td><td>  </td></tr>
</table>

The terminating resistor must be activated at the **end of the bus**. To do this, set the switch to **ON**.

The bus cable must be connected on the incoming side.



TSXCANKCDF90T
TSXCANKCDF180T



TSXCANKCDF90TP

Signal	Terminal block 1	Terminal block 2	Wire color
CAN_H	CH1	CH2	white
CAN_L	CL1	CL2	blue
CAN_GND	CG1	CG2	black
CAN_V+	V+1	V+2	red

CANopen cable

TCXCANCxyy

The cable is available in various versions (**x**):

Standard
No Flame
Heavy Duty

and various lengths (**yy**):

50, 100, 300 m.

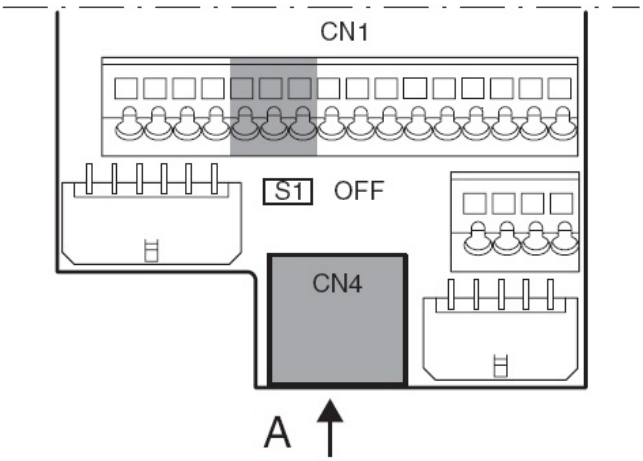
Lexium 05

Modbus and CANopen connector

The CANopen terminal resistor can be activated with switch S1.

However, this does not need to be carried out for the purpose of this application.

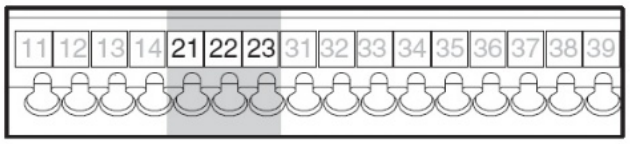
Therefore, ensure that the switch is set to **OFF**.



Lexium 05

CANopen connector

The servo drive is connected to the CANopen bus via terminal CN1



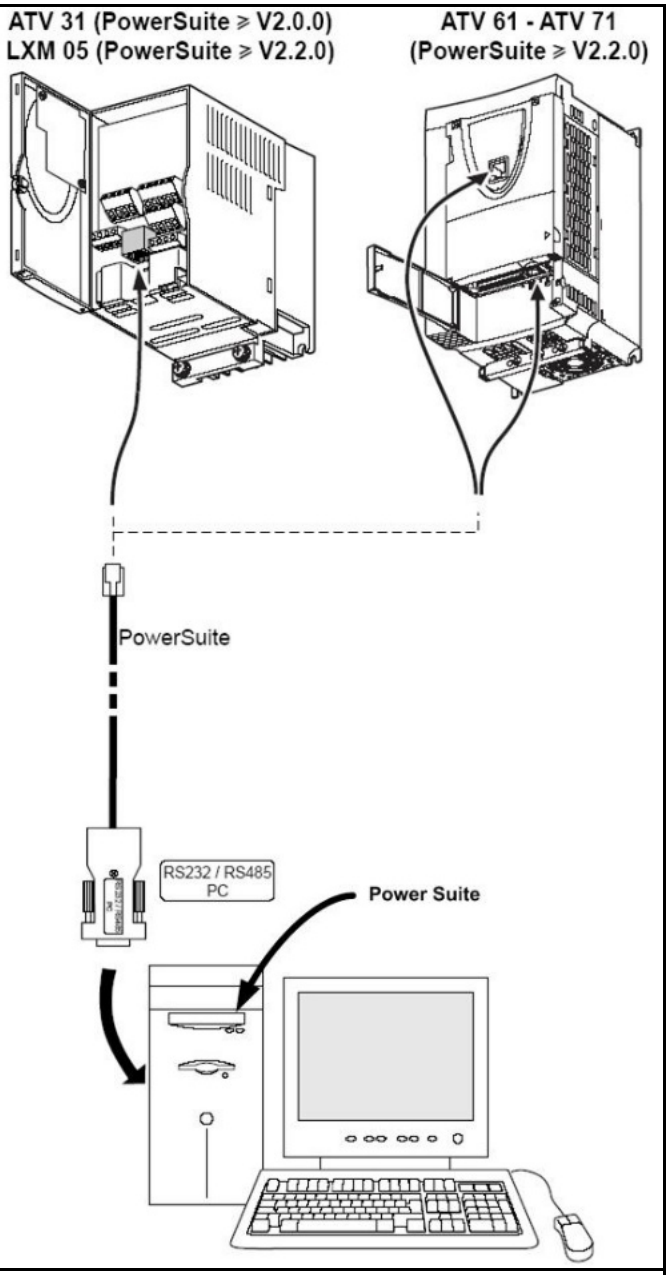
Pin	Signal	Description
21	CAN_0V	CAN reference potential
22	CAN_L	data wire, inverted
23	CAN_H	data wire

Lexium 05
Altivar 31

Serial
PC connection cable

VW3A8106

For the connection between the PC (with PowerSuite software) and the ATV31 VSDs/LXM05 servo drives.
.

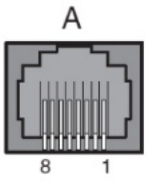


Lexium 05

Modbus and CANopen connector

In this application, the junction box is used to connect the servo drive to the **CANopen** bus via RJ45 socket **CN4**.

The same interface features a Modbus port for establishing a PC and PowerSuite software connection

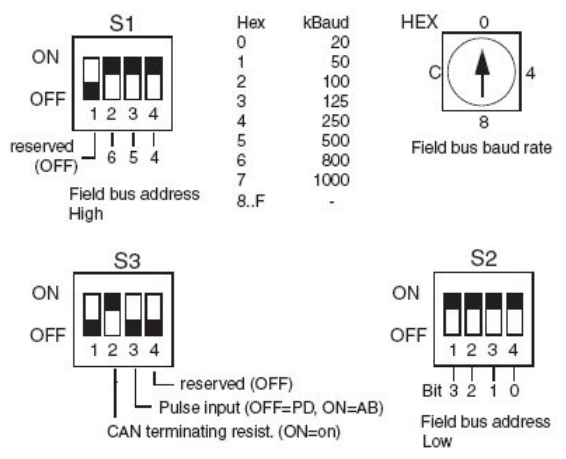


Pin	Signal	Description
1	CAN_H	data wire
2	CAN_L	data wire, inverted
7	MOD+10V_OUT	10V power supply
8	MOD_0V	Reference potential for MOD+10V_OUT

Pin	Signal	Description	
4	MOD_D1	Bidirectional transmit/receive signal	RS485 level
5	MOD_D0	Bidirectional transmit/receive signal, inverted	RS485 level
7	MOD+10V_OUT	10 V power supply, max. 150 mA	Output
8	MOD_0V	Reference potential for MOD+10V_OUT	Output

IcLA IFS93/2

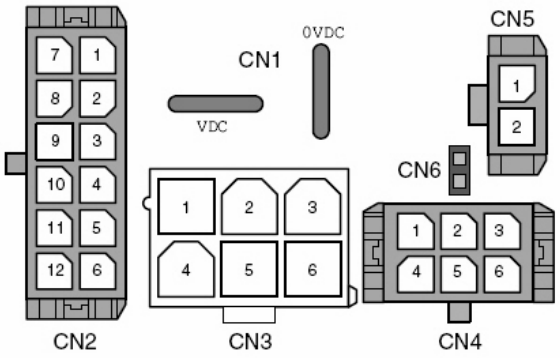
Setting the address and transmission rate

1	The CANopen address and transmission rate are set manually using the DIP switches and the rotary switch in the cable connector of the drive. The address can be varied between 1 and 127 using switches S1 and S2. S3 can be used in addition to define the drive as the last device on the bus by activating the CAN terminating resistor. Set the transfer rate using the rotary switch and select a value of 250 kbaud (switch setting 4) to enable operation with the Premium PLC.	
2	The table summarizes the switch settings for the desired CANopen address. Addresses 5 (000 0101b) and 6 (000 011 b) must be assigned to the drives.	

DIP switch:	S1.1	S1.2	S1.3	S1.4	S2.1	S2.2	S2.3	S2.4
Address bit:	-	6	5	4	3	2	1	0
Coding address 127 (default)	-	1	1	1	1	1	1	1
Coding address 25 (example)	-	0	0	1	1	0	0	1

IcIA IFS93/2

Connections

1	CN1 power supply VDC CN2 Multifunction interface CN3 Service Interface CN4 24-V-Signal Interface CN5 Interface for safety function „Power Removal" CN6 Bridge to de-activate the safety function "Power Removal"	

IcIA IFS93/2

Service interface CN3

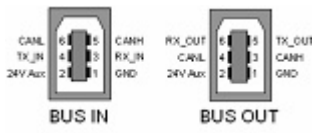
RS485 Cable 0062501463030 open ended

2	Die Service interface CN3 is used for the RS485-Bus connection for maintenance. A PC can be connected to the interface via a RS485-RS232-adapter. With the PC commissioning software "IcIA Easy" you can, for example, read the error buffer or monitor the operational temperature.	

TeSysU Communications module CANopen

LULC15

CANopen Bus Sockets STB zu TeSysU



The communication module LULC15 is connected to the Advantys STBXBE1100 using the cable **LU9RCDxx**.

Use the cable **LU9RDDxx** to connect the LULC15's.

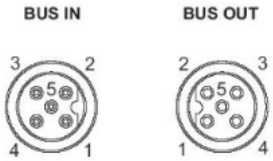
For the CANopen Bus termination use the plug **LU9RFL15**.

Advantys FTB

Bus Cable CANopen

FTXCN3210

Connect the FTB modules in serie starting with CANopen-Tap
VW3 CAN TDM4



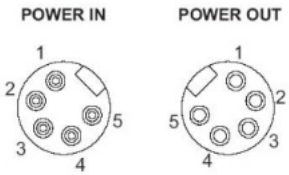
PIN	Signal	Colour
1	Shld	-
2	V+	Red
3	GND	black
4	CAN_H	White
5	CAN_L	Blue

Advantys FTB

Power Supply Cable

FTXDP2210

The FTB power supply cables are linked in serie from one module to the next.



PIN	Signal	Cable
1	0V	1
2	0V	2
3	PE	Green/Yellow
4	+24V DI	3
5	+24V DO	4

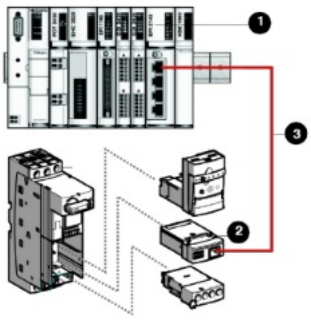
Advantys FTB
Terminal Resistor
CANopen

FTXCNTL12

Connect to the last FTBModule in the series on the BUS OUT-Socket

Advantys STB

Connecting cable LU9R10:
Parallel connection from communication module LUFC00 on the TeSysU to module EPI2145 in the STB-Platform.
2 plugs RJ54.



1 STB – Module EPI2145
2 Communication module LUFC00 in TeSysU
3 Connecting cable LU9R10

Implementation

Introduction

The implementation chapter describes all the steps necessary to initialise, to configure, to program and start-up the system to achieve the application functions as listed below.

Here is an overview of the individual sub-sections:

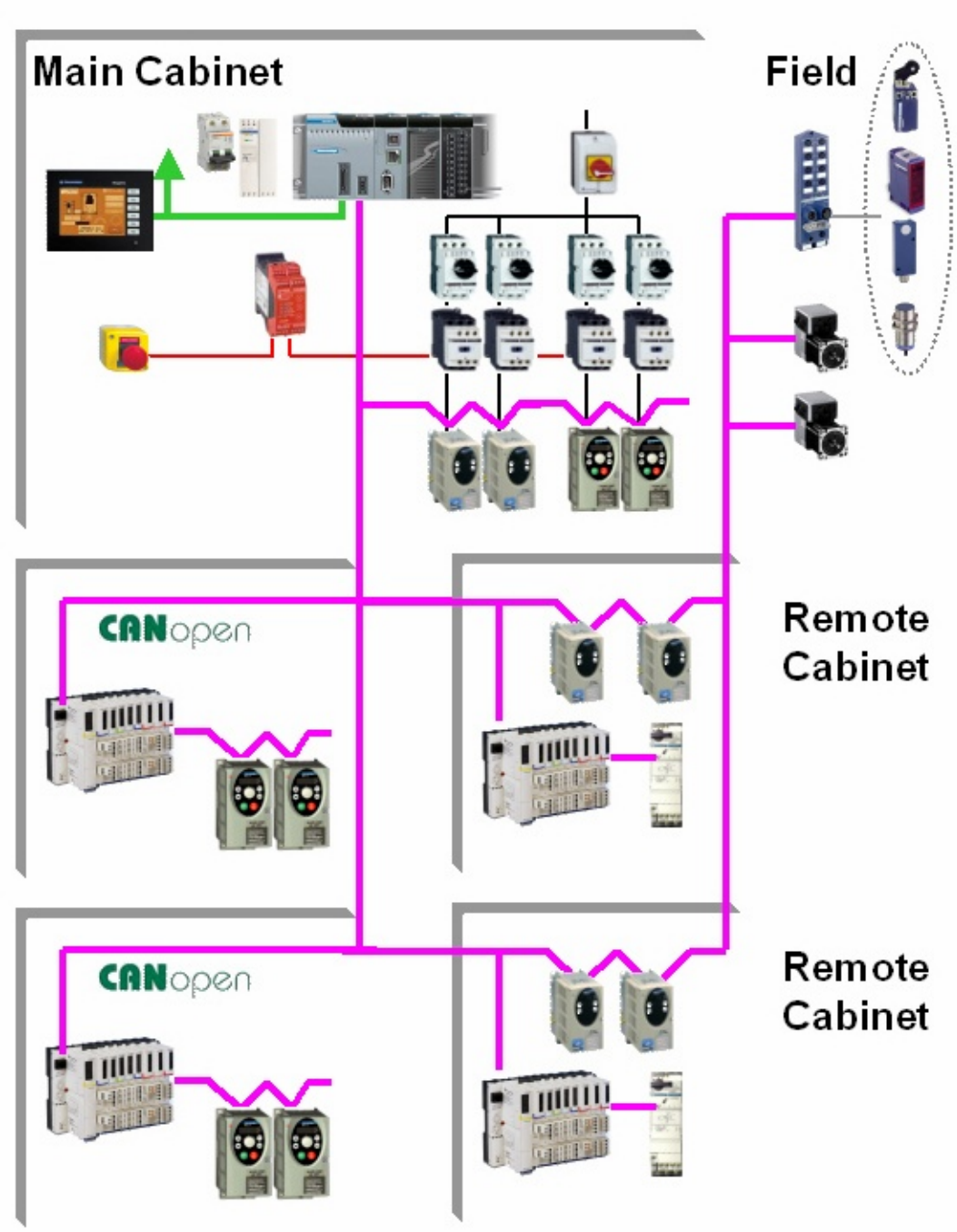
- **Function**
  A short description of the operating procedures
- **Communication**
  The settings, memory areas and variable names used for communication are described here.
- **PLC**
  Describes how to configure the PLC with UnityPro.
- **HMI**
  Instructions for creating the HMI application.
- **Devices**
  Procedure for parameterizing the devices used, such as the Advantys STB and FTB, Lexium05, IcLA, Altivar, and TeSysU.

Function

Instructions for switching on and functional description

1. Switch on the master switch.
2. Switch on all fuses and motor circuit breakers.
3. Acknowledge Emergency Off signals.
4. Wait until all CANopen nodes are on the network.
5. The relevant nodes can be selected and controlled on the HMI. This is only intended for manual operation.

Functional Layout



Communication

Introduction

This chapter describes the data passed via the communications bus (e.g. CANopen or TCP/IP) that is not bound directly with digital or analog hardware.

The list contains:

- The device links
- Direction of data flow
- symbolic name and
- Bus address of the device concerned.

Device Links

CANopen and TCP/IP bus systems are used in this application.

The devices below are networked via **CANopen**:

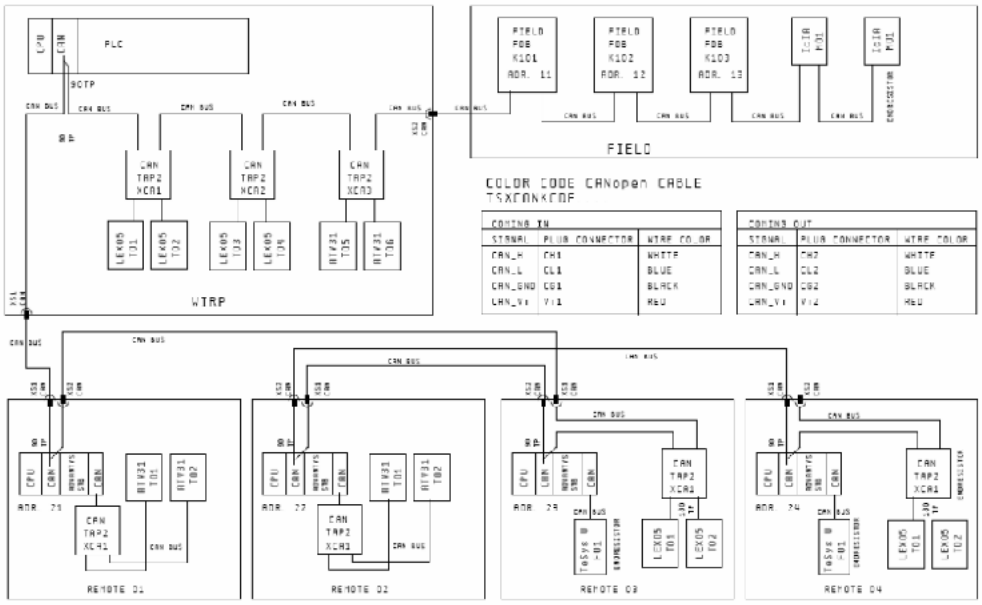
Main Cabinet:
- One Modicon M340 PLC as the bus master, bus address 127
- 2 local Altivar 31 drives, direct, bus addresses 2...3
- 4 local Lexium 05 servo drives, direct, bus addresses 4...7

Remote Cabinets:
- 4 remote STB I/O-Islands, bus addresses 21...24
- 4 remote Altivar 31 drives, attached to STB1,2, each with bus addresses 31 & 32
- 4 remote Lexium 05 servo-drives, direct in remote03 und 04, bus addresses 31, 32 and 41, 42
- 2 remote TeSysU motor-starters, attached to STB3,4, each with the bus address 8

In the field:
- 3 remote FTB I/O-Islands, bus addresses 11...13
- 2 IclA IFS intelligent compact drives, bus addresses 14,15

Two devices are interconnected via **TCP/IP**, along with a PC that has Unity and Vijeo Designer software installed on it for configuration purposes.

- Modicon M340-PLC, bus address 192.168.100.50
- Magelis XBTGT HMI, bus address 192.168.100.51



CANopen

On the CANopen network, you can connect up to **63 devices** and one bus master to the bus. Bus lengths, segments and junctions all have restrictions, which are outlined in the tables below.

The data throughput rate selected for the bus determines the maximum length of the entire network:

Baudrate	Maximale Länge
1 Mbit/s	4 m
500 Kbit/s	**100 m**
250 Kbit/s	250 m
125 Kbit/s	500 m
50 Kbit/s	1000 m
20 Kbit/s	2500 m

Note: Number of PDOs supported:

- 256 receiving (RxPDO)
- 256 transmitting (TxPDO)

The following **CANopen settings** are used in this application:

- A baud rate of **500 kbps** and
- A **200 ms heartbeat** monitoring the nodes

CANopen
Transmission Definition

Transmission type:

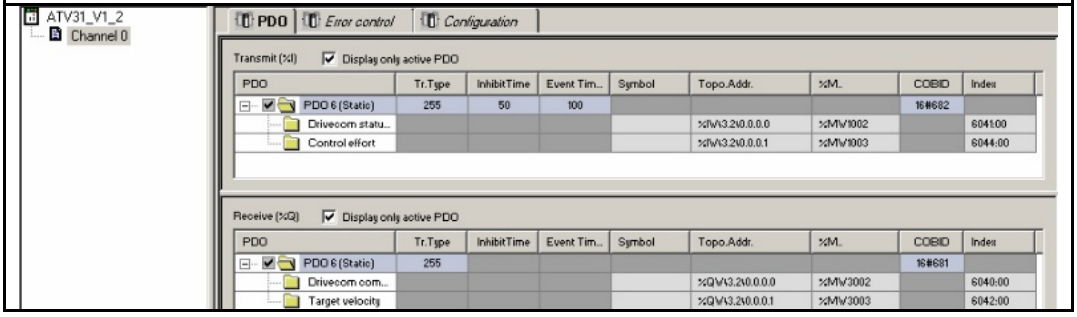
- Synchronous acyclic: The transmission type **0** means that the message is sent sent synchronously with the SYNC-signal, but not cyclically.
- Synchronous cyclic: A value between **1** and **240** means that the PDO is transmitted sychronously and cyclically; The value of the transmission type defines the number of SYNC signals between 2 PDO telegrams.
- Asynchronous PDO: The transmission type **254** means that the PDO is sent asynchronously. It depends on the implementation of the function in the device and is normally used for digital I/O.
- Synchronous PDO: The transmission type **255** means that the PDO is sent asynchronously, as soon as the value changes.

You must ensure that the the selected transmission type is supported by the selected device.

Inhibit time
- The time in which no PDO is sent. 0 means de-activated.

Event timer
- The time in which at least 1 PDO is sent. 0 means de-activated.

Altivar 31
Node 02



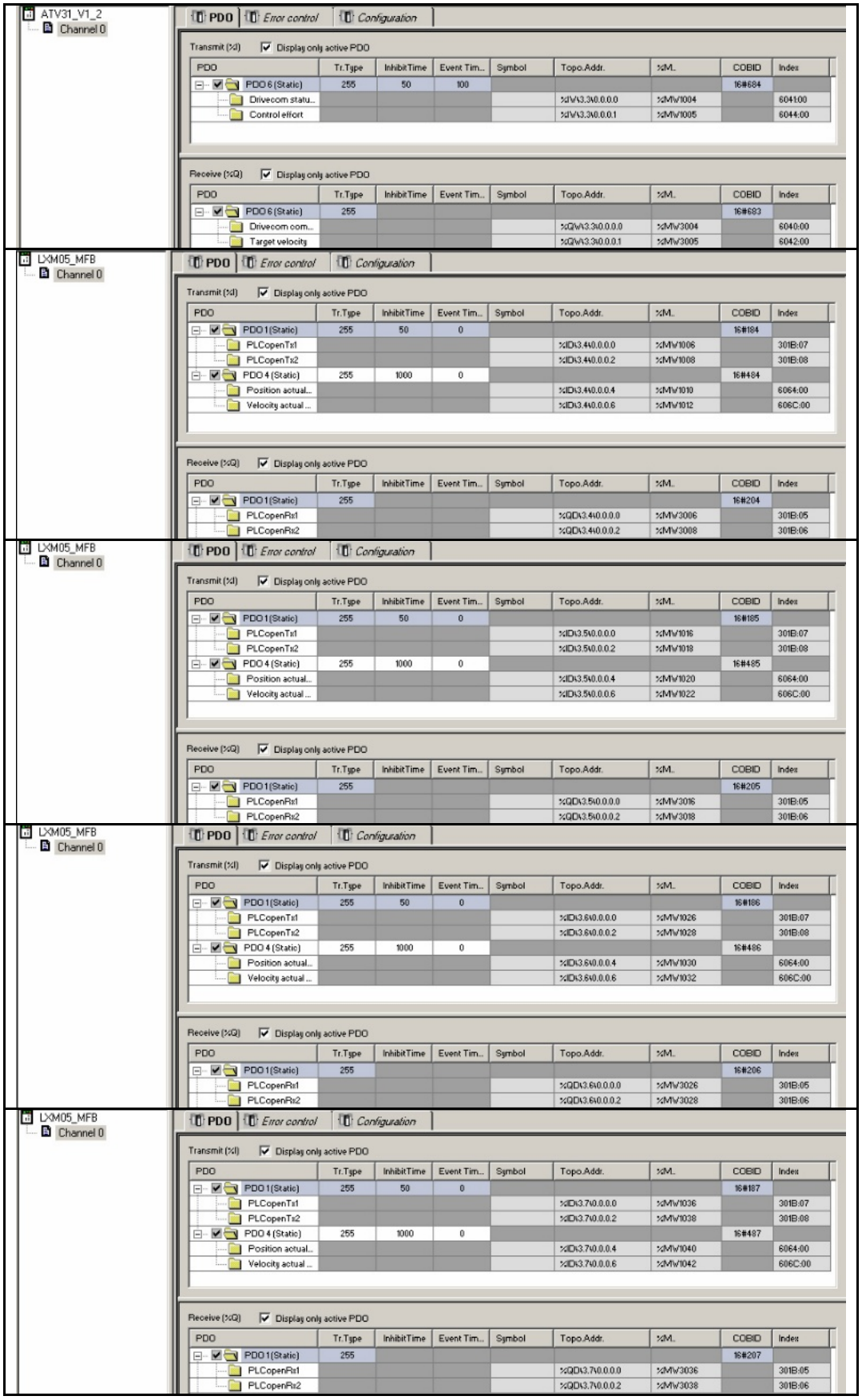
Altivar 31
Node 03

Lexium 05
Node 04

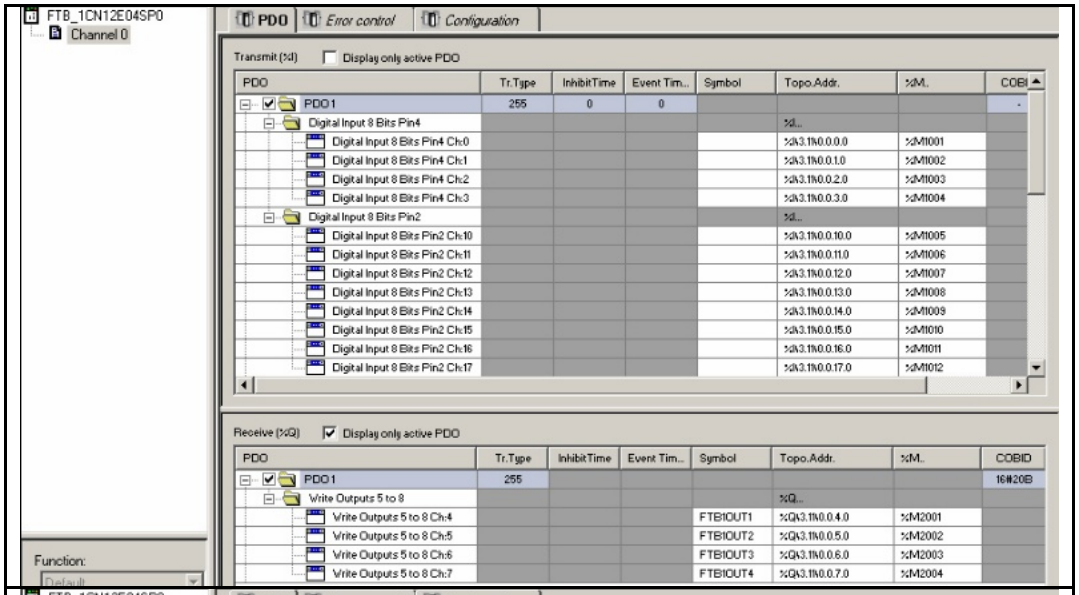
Lexium 05
Node 05

Lexium 05
Node 06

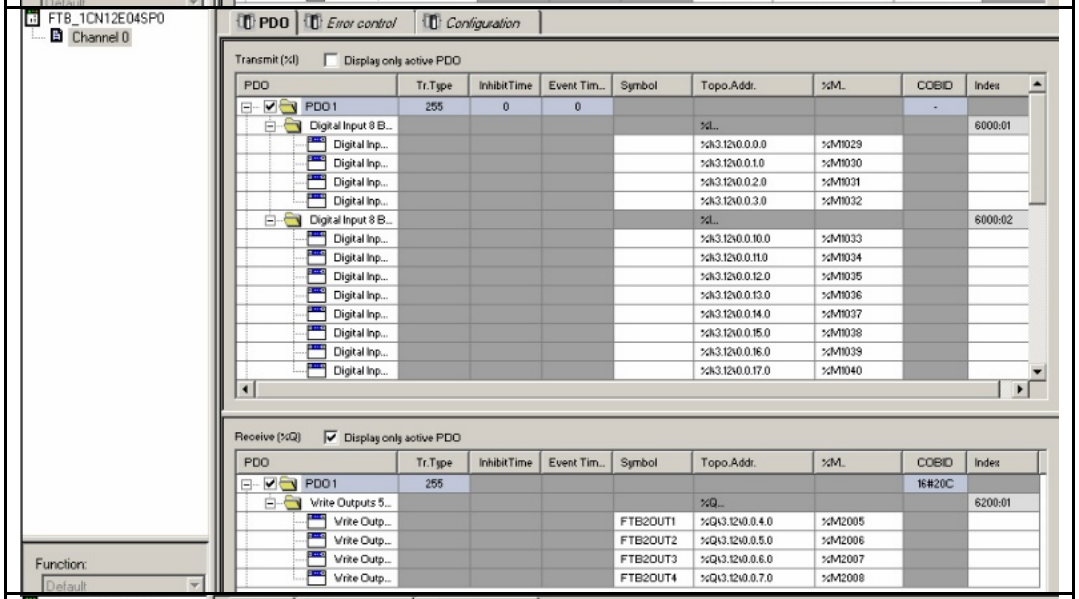
Lexium 05
Node 07



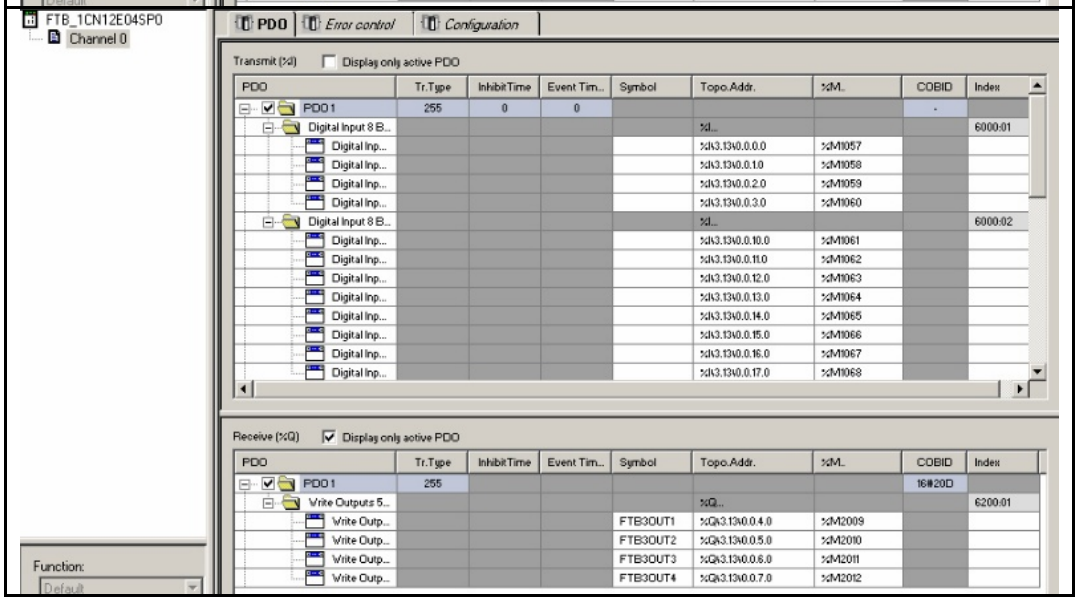
FTB
Node 11



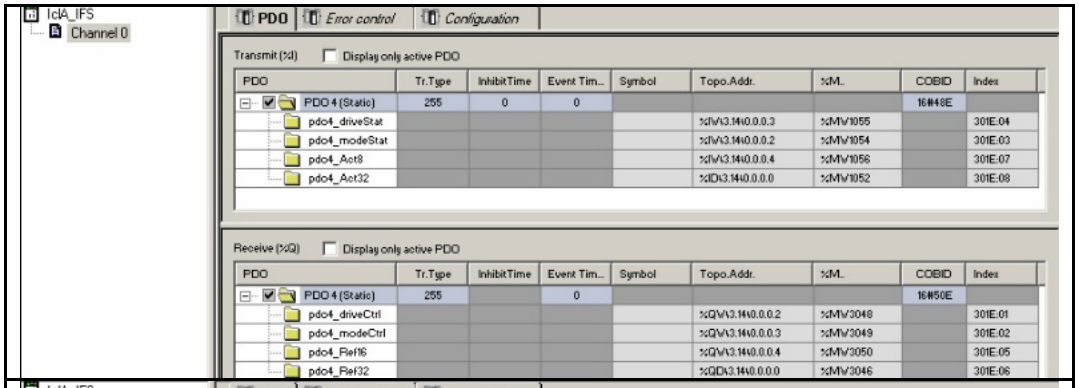
FTB
Node 12



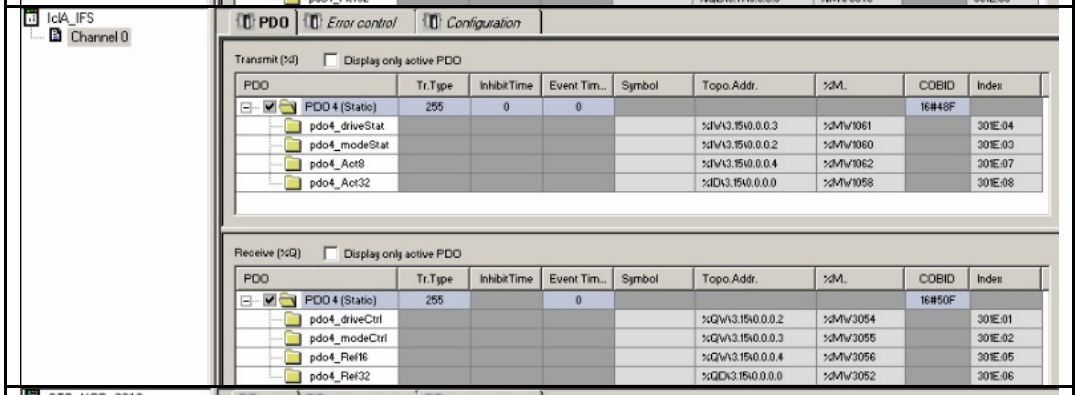
FTB
Node 13



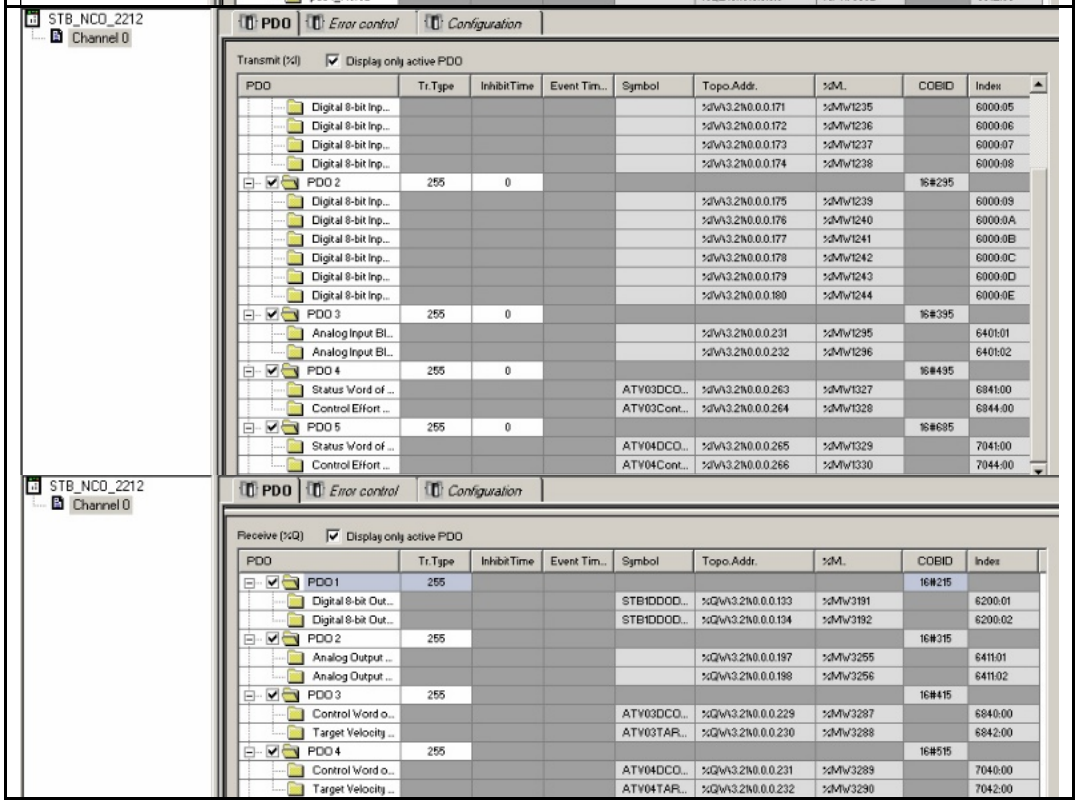
IclA IFS
Node 14



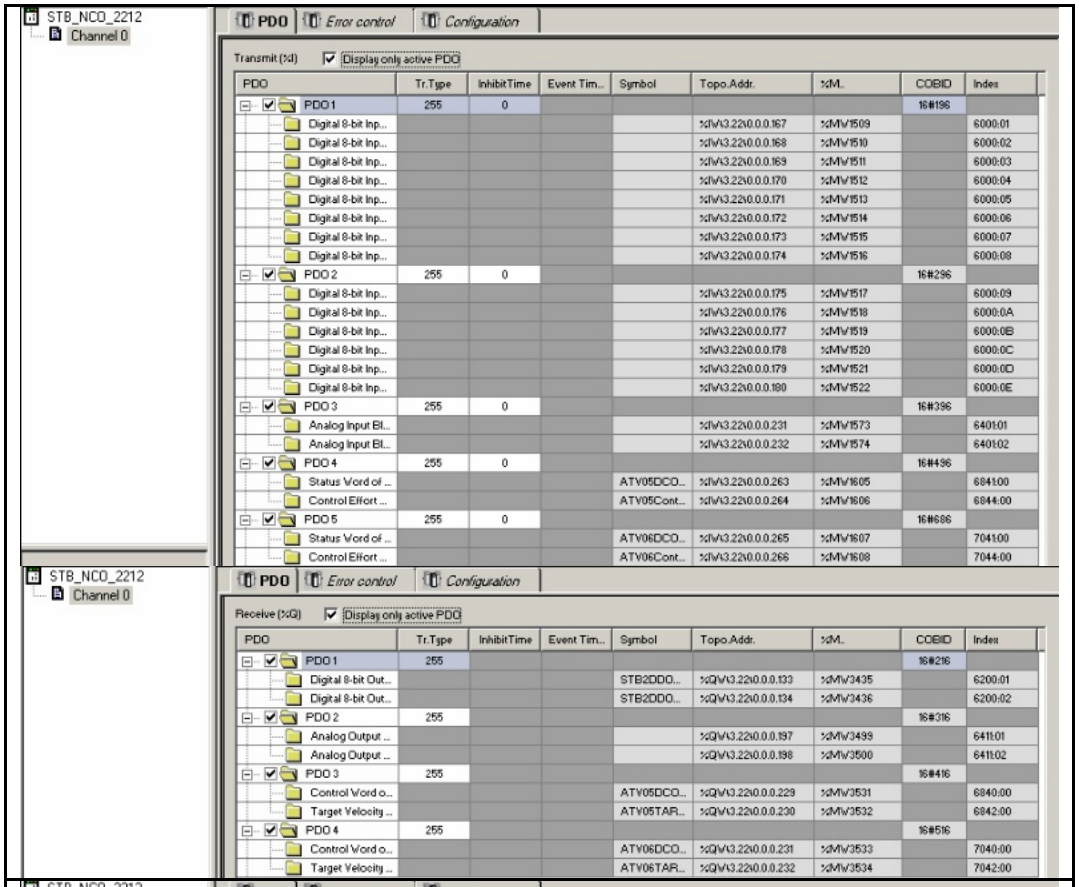
IclA IFS
Node 15



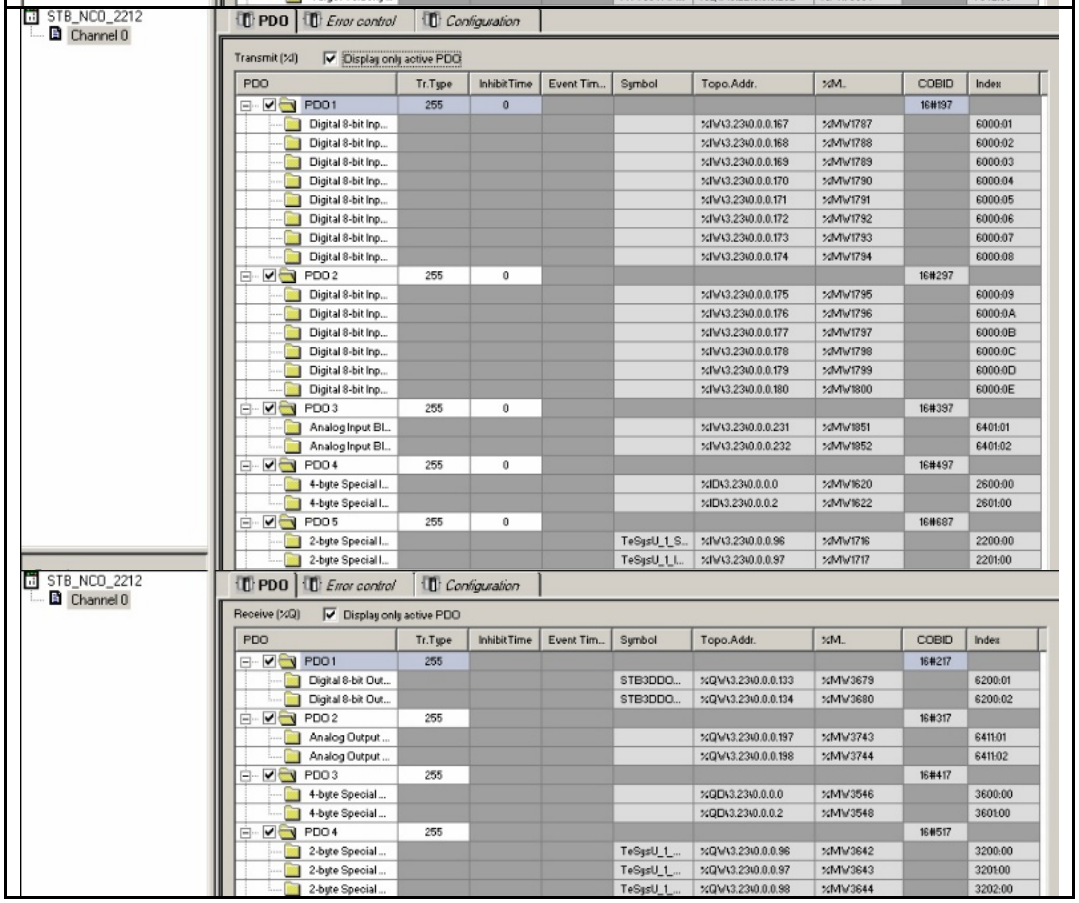
STB
Node 21



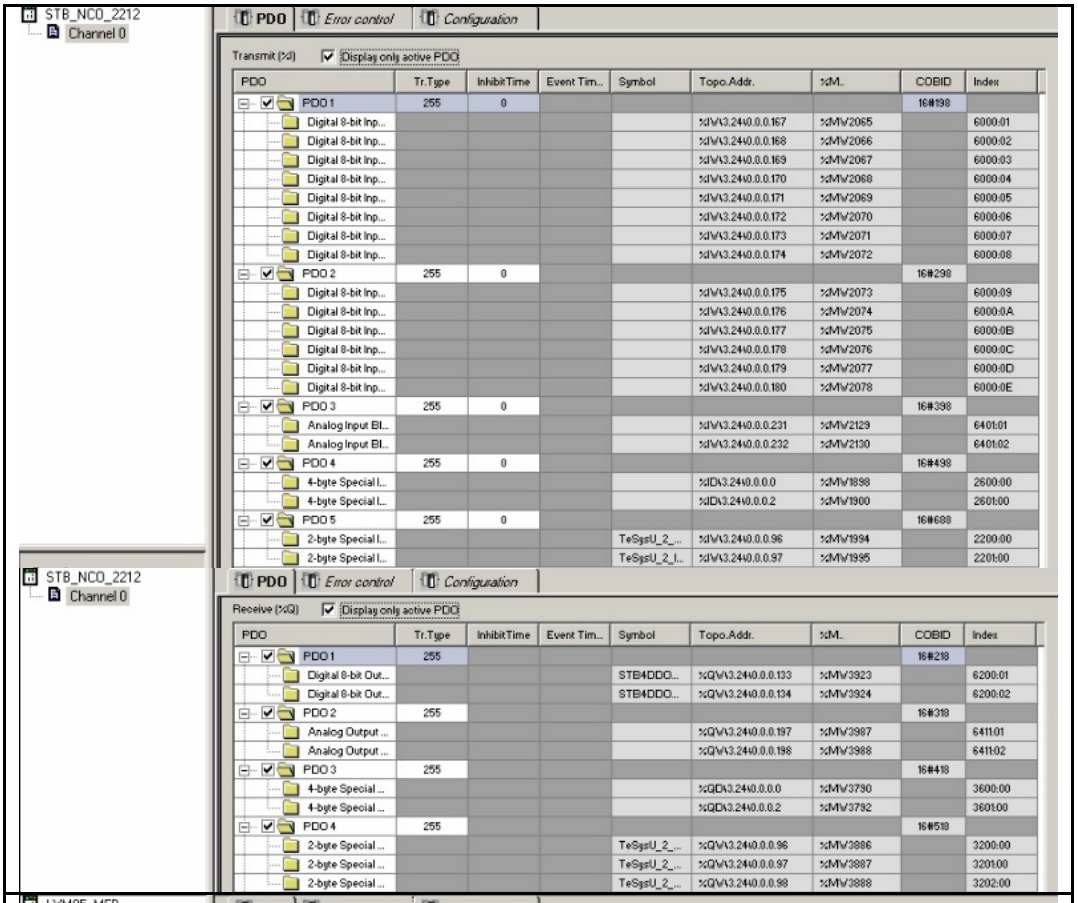
STB
Node 22



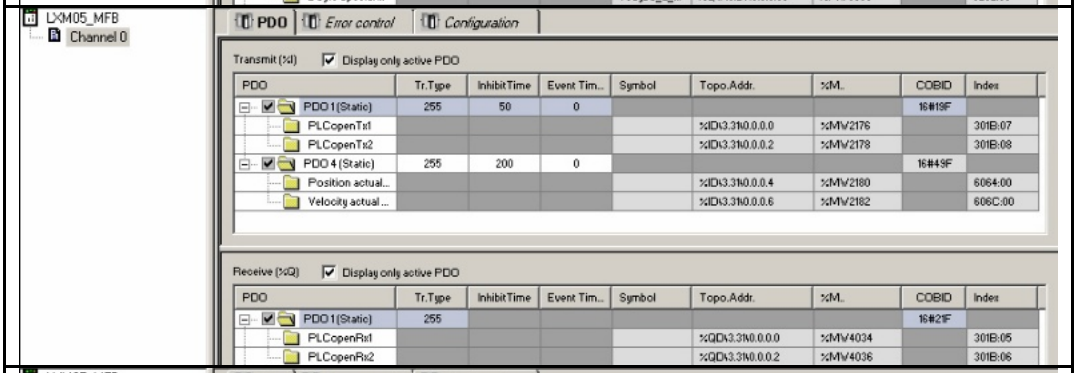
STB
Node 23



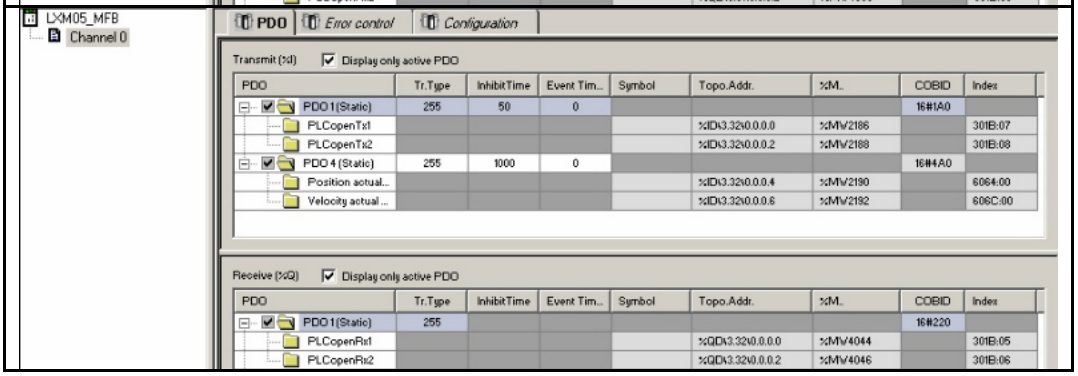
STB
Node 24



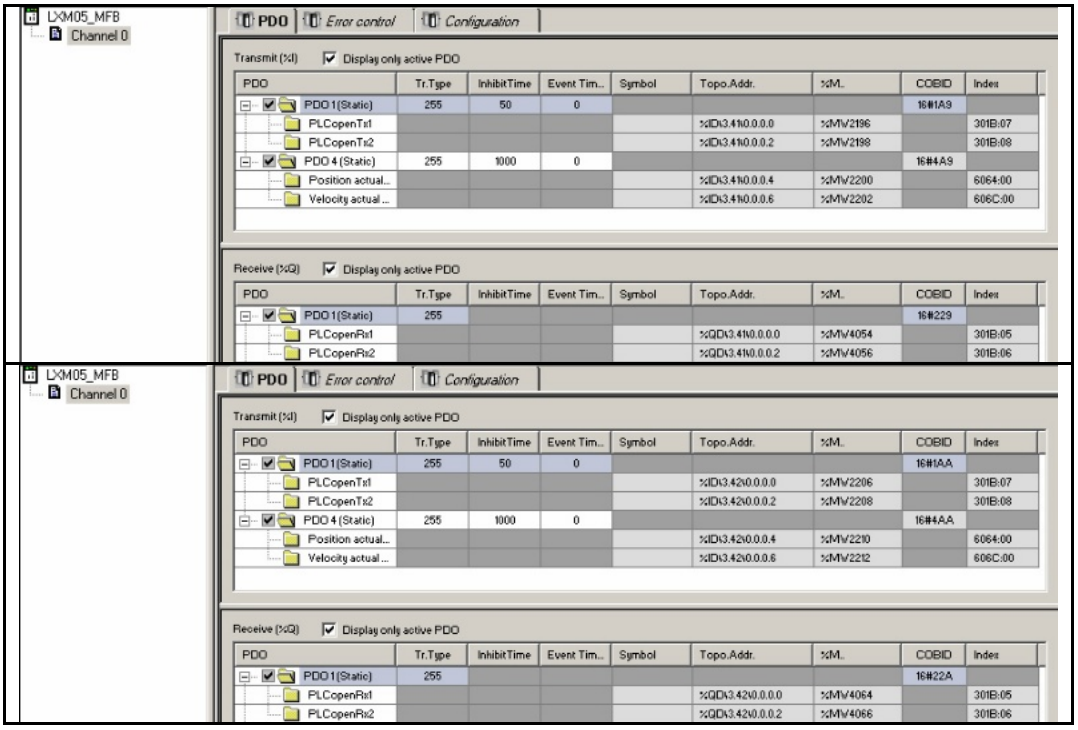
Lexium 05
Node 31



Lexium 05
Node 32



Lexium 05
Node 41



Lexium 05
Node 41

Ethernet

HMI <> PLC
Address
Summary

Data direction HMI ←→ PLC		
Device	**Start address**	**Reserved Area**
General		%M501…650
CANopen		%MW401…444
1. LXM05	%MW500	%MW501…520
2. LXM05	%MW520	%MW521…540
3. LXM05	%MW540	%MW541…560
4. LXM05	%MW560	%MW561…580
5. LXM05	%MW580	%MW581…600
6. LXM05	%MW600	%MW601…620
7. LXM05	%MW620	%MW621…640
8. LXM05	%MW640	%MW641…660
1. ATV31	%MW660	%MW661…680
2. ATV31	%MW680	%MW681…700
3. ATV31	%MW30	%MW30…34
4. ATV31	%MW35	%MW35…39
5. ATV31	%MW40	%MW40…44
6. ATV31	%MW45	%MW45…49
1. STB	%MW800	%MW801…819
2. STB	%MW820	%MW821…839
3. STB	%MW840	%MW841…859
4. STB	%MW860	%MW861…899
1. TeSysU	%MW900	%MW781…799
2. TeSysU	%MW910	%MW801…819
1. FTB	%MW920	%MW901…919
2. FTB	%MW930	%MW931…939
3. FTB	%MW940	%MW941…949
1. IclA	%MW920	%MW821…839
2. IclA	%MW930	%MW841…859

Ethernet
HMI <> PLC
Lexium and Altivar

Data Direction HMI ←→ PLC (Lexium and Altivar)						
Name	**%MW**	**Bit**	**Typ**	**LXM**	**ATV**	**Designation**
YY_X_Ready	+1	0	BOOL	X	x	Drive is ready
YY_X_Power	+1	1	BOOL	X	x	Drive power on
YY_X_Start	+1	2	BOOL	X	x	Start drive
YY_X_Dir	+1	3	BOOL	X	x	Direction
YY_X_Mode_VE	+1	4	BOOL	X		Set velocity mode
YY_X_Mode_AB	+1	5	BOOL	X		Set absolute pos. mode
YY_X_Mode_RE	+1	6	BOOL	X		Set relative pos. mode
YY_X_Reset	+1	7	BOOL	X	x	Reset error
YY_X_Velocity	+2		DINT	X	x	Target velocity
YY_X_Position	+4		DINT	X		Target position
YY_X_ACC	+6		UDINT	X		Acceleration
YY_X_DCC	+8		UDINT	X		Deceleration
YY_X_Active	+11	0	BOOL	X	x	Drive is active
YY_X_Disable	+11	1	BOOL	X	x	Drive is disabled
YY_X_Standstill	+11	2	BOOL	X	x	Drive in standstill
YY_X_Stopping	+11	3	BOOL	X	x	Drive in stopping
YY_X_IN_VE	+11	4	BOOL	X	x	Drive in velocity mode
YY_X_IN_AB	+11	5	BOOL	X		Drive in absolute pos mode
YY_X_IN_RE	+11	6	BOOL	X		Drive in relative pos mode
YY_X_in_Velocity	+11	7	BOOL	X	x	Drive has reached velocity
YY_X_in_Position	+11	8	BOOL	X		Drive is in position
YY_X_Error	+11	9	BOOL	X	x	Error
YY_X_Act_Position	+12		DINT	X		Position actual value
YY_X_Act_Velocity	+14		DINT	X	x	Velocity actual value
YY_X_ErrorID	+16		UDINT	X	x	Error ID code
YY_X_ErrorMA	+18		INT	X	x	Error message code

YY - **YY** stands for the drive type.
YY can be either **LXM05** or **ATV31**.
_X_ - **X** represents the specific drive number for a particular type.
X can range from **1 to 6.**
The address is made up of the start address (mentioned above) + %MW + bit. In the case of the third Lexium 05 for the direction, the address is:
%MW540 + 1 + bit = %MW541.3

Ethernet
HMI <> PLC
for TeSysU

Data Direction HMI ←→ PLC (for TeSysU)				
Name	**1. TeSysU**	**2. TeSysU**	**Typ**	**Designation**
YY_X_HMI_Ready	%MW781.0	%MW801.0	BOOL	Power is ON
YY_X_HMI_Run	%MW781.1	%MW801.1	BOOL	Pole status is closed
YY_X_HMI_Trip	%MW781.2	%MW801.2	BOOL	Tripped position
YY_X_HMI_Error	%MW781.3	%MW801.3	BOOL	Fault or warning
YY_X_HMI_Start	%MW782.0	%MW802.0	BOOL	Run forward
YY_X_HMI_Reset	%MW782.1	%MW802.1	BOOL	Reset fault and warning

YY - **YY** stands for the drive type.
YY is **TeSysU.**
_X_ - **X** represents the specific drive number.
X can be either **1 or 2.**

Ethernet
HMI <> PLC for STB

Data Direction HMI ←→ PLC (for STB)						
Name	**1. STB**	**2. STB**	**3. STB**	**4. STB**	**Typ**	**Designation**
YY_X_HMI_Input1	%MW801	%MW821	%MW841	%MW861	BOOL	Input
YY_X_HMI_Input2	%MW802	%MW822	%MW842	%MW862	BOOL	Input
..	..	..	..	..	..	
YY_X_HMI_Input10	%MW809	%MW829	%MW849	%MW869	BOOL	Input
YY_X_HMI_Output1	%MW811	%MW831	%MW851	%MW871	BOOL	Output
YY_X_HMI_Output2	%MW812	%MW832	%MW852	%MW872	BOOL	Output
..	..	..	..	..	..	
YY_X_HMI_Output10	%MW819	%MW839	%MW859	%MW879	BOOL	Output

YY - **YY** stands for the drive type.
YY is **STB**.
_X_ - **X** represents the specific drive number .
X can range from **1 to 2.**

Ethernet
HMI <> PLC for FTB

Data Direction HMI ←→ PLC (for FTB)					
Name	**1. FTB**	**2. FTB**	**3. FTB**	**Typ**	**Designation**
YY_X_HMI_Input1	%MW921	%MW931	%MW941	BOOL	Input
YY_X_HMI_Input2	%MW922	%MW932	%MW942	BOOL	Input
YY_X_HMI_Outpit	%MW925	%MW935	%MW945	BOOL	Output

YY - **YY** stands for the drive type.
YY is **FTB**.
_X_ - **X** represents the specific drive number .
X can range from **1 to 2.**

General Addressing

PLC and HMI

Various hardware addresses, as well as flags and flag words, are used in the PLC/HMI application. An overview of the addresses used is provided below. The "Address" column shows how the address is written and the potential ranges within the example application.

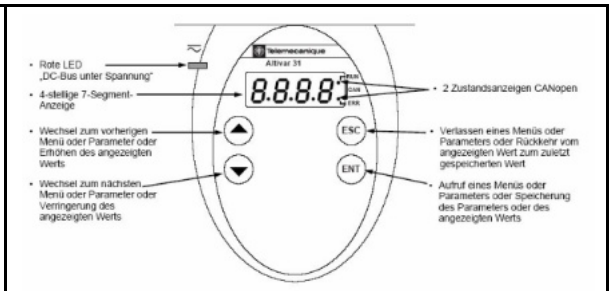
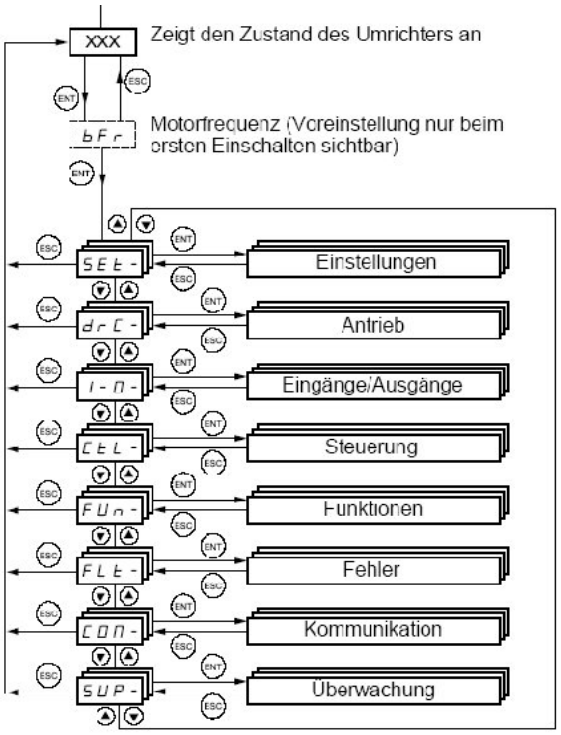
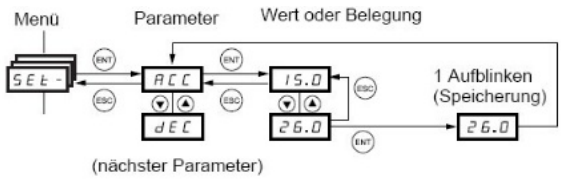
Type	Address	Comment
Digital inputs	%Ir.m.x -r: 0 -m: 1 - 3 -x: 0…31	**PLC:** Digital inputs are specified on a hardware basis: r indicates the rack number, m the slot and x the input number.
Digital outputs	%Qr.m.x -r: 0 -m: 3 - 4 -x: 0…31	**PLC:** Digital outputs are specified on a hardware basis: r indicates the rack number, m the slot and x the output number.
Analog inputs	%IWr.m.c -r: 0 -m: 5 -c: 0…3	**PLC:** Analog inputs are specified on a hardware basis: r indicates the rack number, m the slot and c the channel number.
Analog outputs	%QWr.m.c -r: 0 -m: 6 -c: 0…2	**PLC:** Analog outputs are specified on a hardware basis: r indicates the rack number, m the slot and c the channel number.
Flag words	%MWx -x Word	**PLC and HMI:** Flag words are used for data exchange between the PLC and HMI. The range depends on the settings in the PLC. Maximum: 32463; 0 - 9999 are used
Flags	%Mx -x Word	**PLC and HMI:** Flags are used for data exchange between the PLC and HMI. The range depends on the settings in the PLC. Maximum: 32633; 0 - 9999 are used
Derived flags	%MWx.y %MWx:Xy -x Word -y Bit	**PLC and HMI:** The elements (bits) from the flag words are used for data exchange between the PLC and HMI. The range depends on the settings in the PLC. Maximum: 32633; 0 – 9999 used; Bits 0 - 15. **Various PLC notations.** %MW100.1 Bit 1 from MW100 **HMI** %MW102:X1 Bit 1 from MW100
CANopen status	%CHr.m.c -r: 0 -m: 0 -c: 2	**PLC:** Status data for CANopen is read via data structure T_COM_CO_BMX (IODDT). Channel address: r indicates the rack number, m the slot and c the channel number. CANopen status %CH0.0.2

CANopen Bus Addresses

The device address and baud rate must be defined for all devices on the CANopen bus.

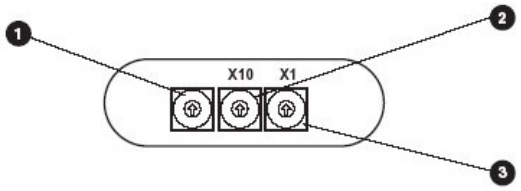
Altivar 31

Defining Address and Baud rate (manual)

1	The CANopen-Address and Baudrate can be input using the buttons on the front panel of the Altivar.	
2	Using the buttons on the front panel, select the sub-menu **Communication** In the **Communication** sub-menu input the CANopen address in the parameter **AdC0.** In the example application the adresses for the six controlers are 2,3 and twice 31,32. Confirm the changes by pressing **ENT**. Also in the **Communication** sub-menu, in the parameter **BdC0,** set the baudrate to **500.0** (kBits). Alternatively you can use the PowerSuite software to configure the CANopen addresses and baudrates. (see chapter : Devices - Altivar 31) To set the parameters you must reset each controller individually by switching the current off and on.	 
3	**To activate the downloaded bus parameters (address and baudrate) you must now switch off the drive (display goes off). On switching back on, the new parameters are ready.**	

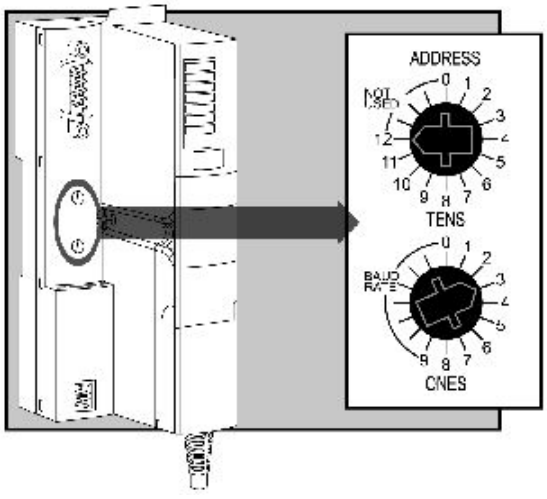
Advantys FTB

Setting the address and transmission rate

1	The CANopen address and transmission rate are set manually on the modules via rotary switches. There are two rotary switches for the address and one for the transmission rate. In the example software, both FTB modules are configured with the **addresses 11** to **13** and the transmission rate is set to **500.0 kbaud**. Alternatively, you can set the system to detect the transmission rate automatically.	 **Element** / **Function** 1 – Transmission speed (kBit/s) 2 – Node-ID x 10 switch 3 – Node-ID x 1 switch **Position of the encoder wheel** / **Transmission speed** O – Automatic recognition 1 – 10 kBits/s 2 – 20 kBits/s 3 – 50 kBits/s 4 – 100 kBits/s 5 – 125 kBits/s 6 – 250 kBits/s 7 – 500 kBits/s 8 – 800 kBits/s 9 – 1 Mbits/s

Advantys STB

Setting the address and transmission rate

1	To set the CANopen address and transmission rate, you will need a screwdriver to turn the rotary switches on the Advantys island head.	

<table>
<tr>
<td>2</td>
<td>Setting the baud rate:
1. Disconnect island voltage.

2. Set bottom rotary switch (ONES) to any of the positions after the number 9 (baud rate).

3. Set the baud rate on the top rotary switch (TENS). Select position 4 for a transmission rate of 500 kbaud (position 5).

4. Re-connect island voltage.</td>
<td>
<table>
<tr><th>Position (Upper Switch)</th><th>Baudrate</th></tr>
<tr><td>0</td><td>10.000 Bit/s</td></tr>
<tr><td>1</td><td>20.000 Bit/s</td></tr>
<tr><td>2</td><td>50.000 Bit/s</td></tr>
<tr><td>3</td><td>125.000 Bit/s</td></tr>
<tr><td>4</td><td>250.000 Bit/s</td></tr>
<tr><td>5</td><td>500.000 Bit/s</td></tr>
<tr><td>6</td><td>800.000 Bit/s</td></tr>
<tr><td>7</td><td>1 Mbit/s</td></tr>
</table>
</td>
</tr>
<tr>
<td>3</td>
<td>Setting the CANopen address:

1. Disconnect island voltage.

3. Set the bottom rotary switch (ONES) to the position that corresponds to the desired address in terms of the ONES place value. If the desired address is 010, the correct position is 0.

4. Set the top rotary switch (TENS) to the position that corresponds to the desired address in terms of the TENS and HUNDREDS place values. If the desired address is 010, the correct position is 1.

4. Re-connect the island voltage.</td>
<td>Notes:

Although addresses can be set manually, addresses 128 and 129 are not available for selection, as CANopen only supports addresses from 0 to 127.

After configuring the CANopen address, it is recommended that you leave the rotary switches in this address position so that, when the system is switched on, the island is always detected at the same address.</td>
</tr>
</table>

Modicon M340

Introduction

The PLC chapter describes the steps required for the initialization and configuration and the source program required to fulfill the functions.

Pre-Conditions

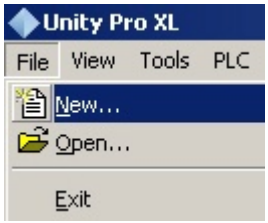
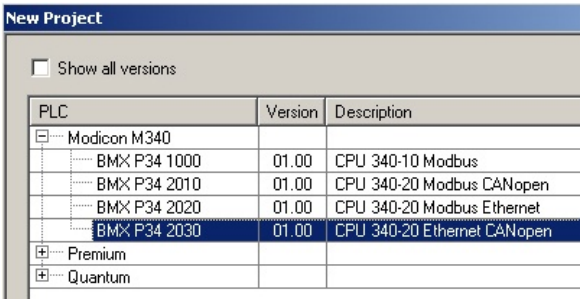
Before carrying out the steps described below, you must ensure the following:

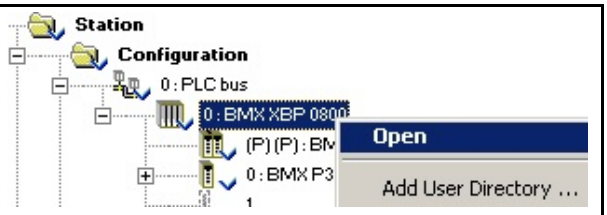
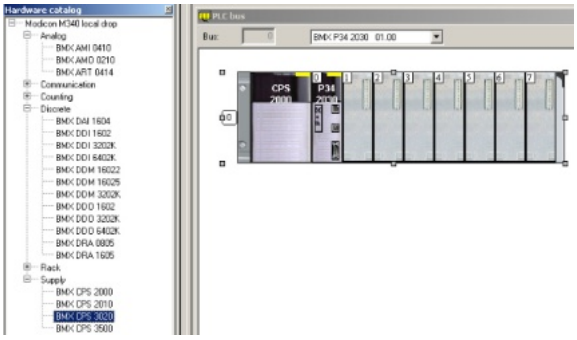
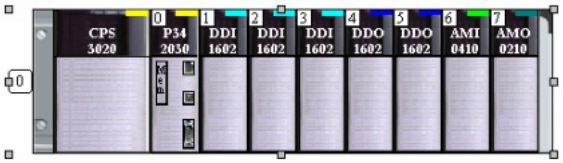
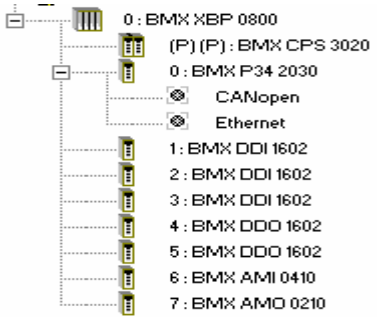
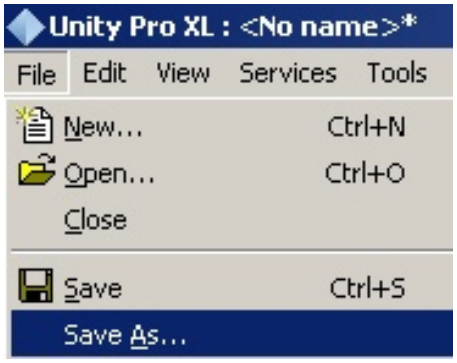
- The UnityPro programming software is installed on your PC.
- The Modicon M340 PLC is connected to the power supply.
- The PLC and the PC are connected to one another via the programming cable (**BMXXCAUSB0xx**) or Ethernet (with a known IP address).

Setting up the PLC is done as follows:

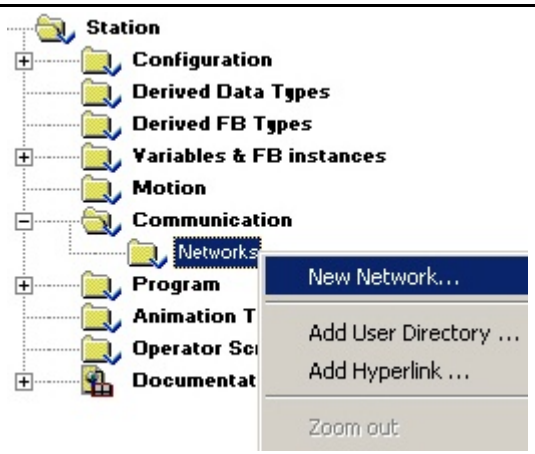
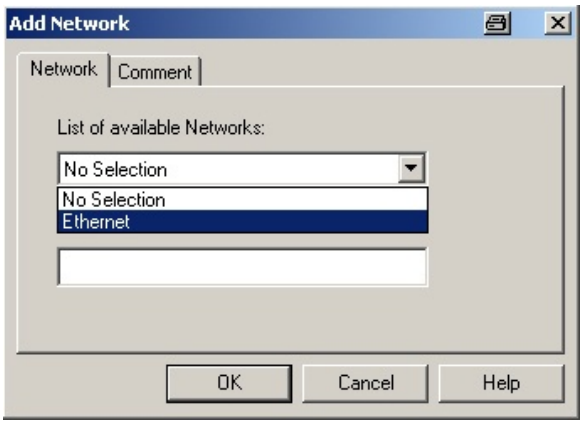
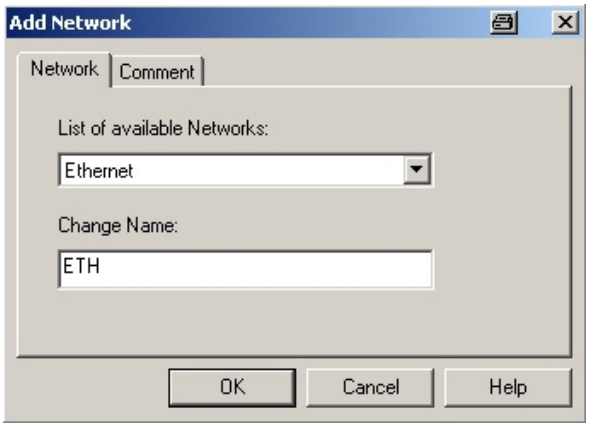
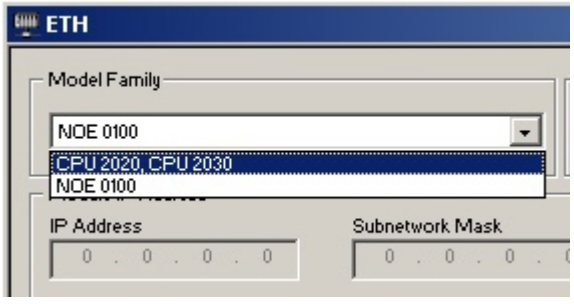
- Create a new program and select hardware.
- Parameterize the communication.
- Create new variables.
- Add CANopen nodes.
- Parameterize CANopen PDO.
- Set up axes for the drives.
- Program assignment.
- MFB - Motion Function Block.
- Create and use DFB.
- Required blocks.
- Build project.
- Connect PC to PLC and transfer project.
- Export and archive project.

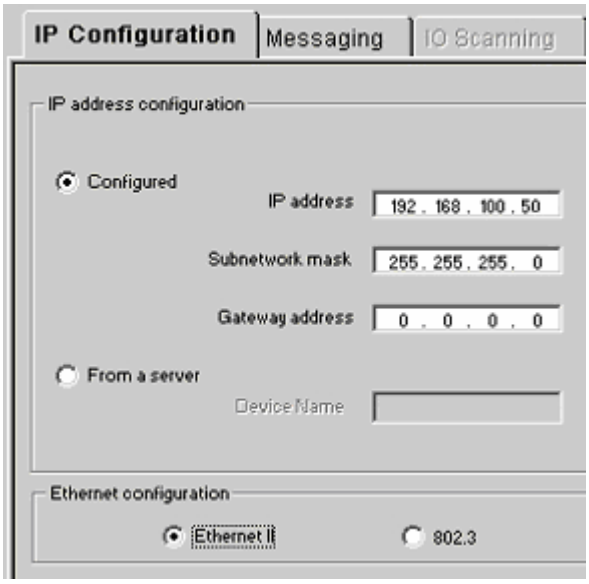
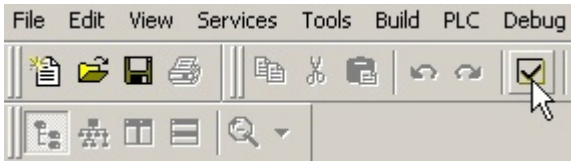
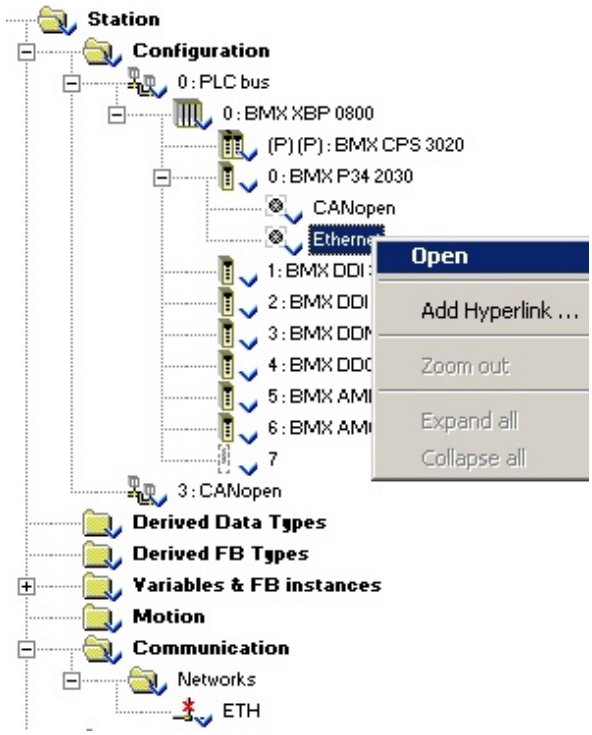
Creating a New Program and Selecting Hardware

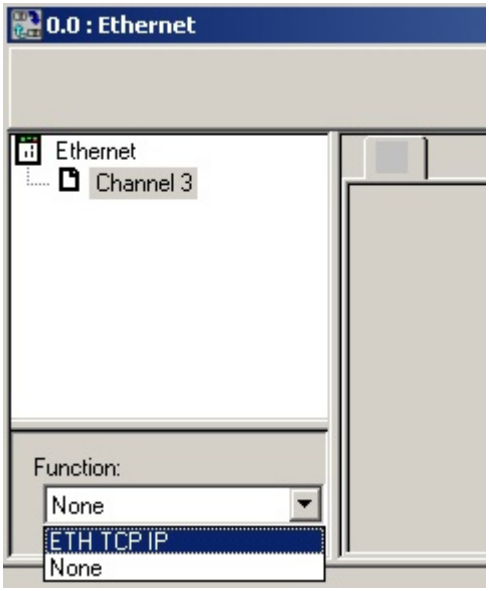
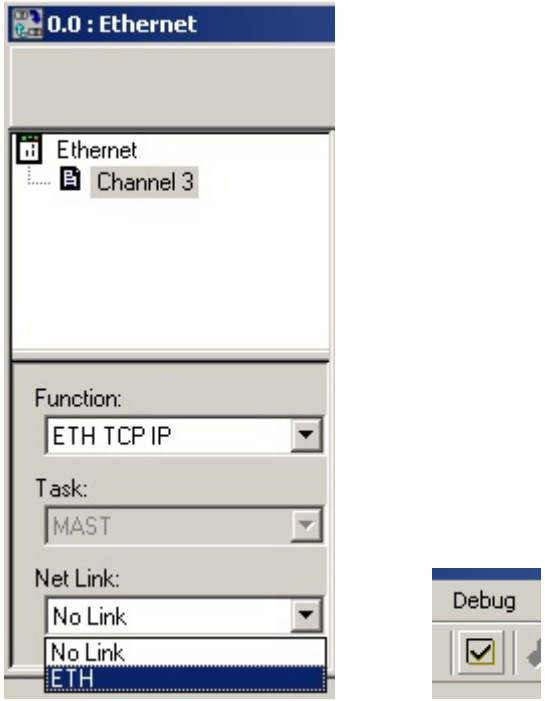
1	To create a new program, select **New** from the **File** menu.	
2	A window opens where you can select the CPU to be used. For this application, select the Modicon M340 CPU **BMX P34 2030** and click **OK** to confirm. This will load the default settings.	

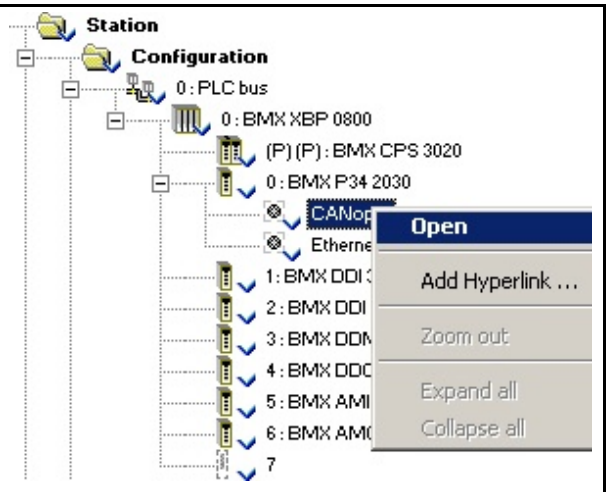
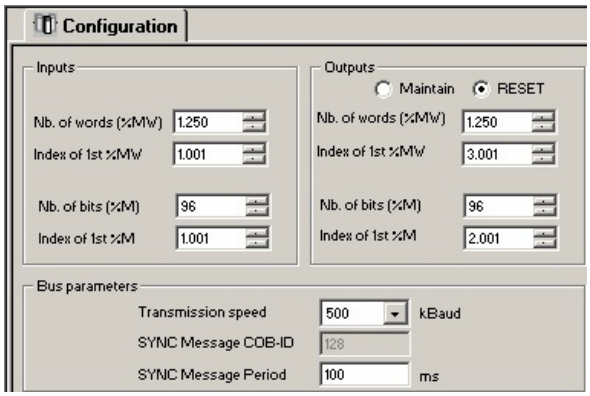
3	Double-click the rack in the project browser or right-click and select **Open**.	
4	This will open the rack and the **Hardware catalog**. To equip the rack, simply select the individual components and drag and drop them to the empty slots. The following hardware is used: Rack **BMX XBP 0800** Power **BMX CPS 3020** CPU **BMX P34 2030** 16DI **BMX DDI 1602** 16DI **BMX DDI 1602** 16DI **BMX DDI 1602** 16DO **BMX DDO 1602** 16DO **BMX DDO 1602** 4AI **BMX AMI 0410** 2AO **BMX AMO 0210**	
5	The display shown opposite will appear.	
6	This is what the display looks like as a tree structure in the project browser	
7	At this point, it is recommended that you save the project. To do this, select **Save As...** in the **File** menu. You can then select the **File name** (<File name>.**stu**) and the location where the file is to be saved under **Save in**. Click **OK** to exit.	

Parameterizing the Communication

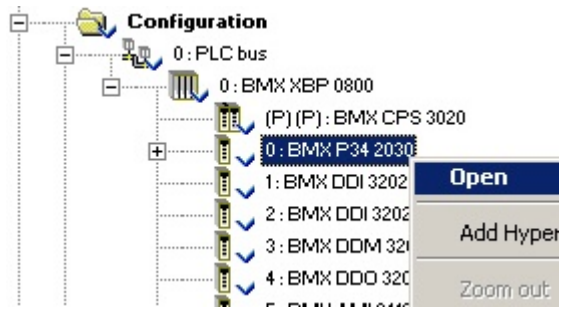
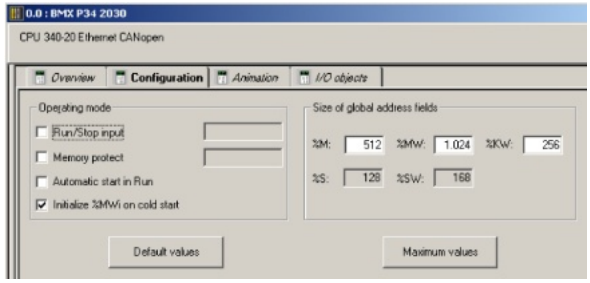
1	Ethernet and CANopen interfaces are used in this application. For Ethernet, the first thing you need to do is create a new network. To do this, right-click **Networks** in the **Communication** directory and select **New Network...**.	
2	Select **Ethernet** from the list of networks in the window that appears.	
3	A name must also be entered. You are free to choose any name, but in this example, **ETH** is used. Click **OK** to confirm.	
4	Open the parameterization window by right-clicking **ETH** and selecting **Open**.	
5	First, select **CPU 2030** under **Model Family**.	

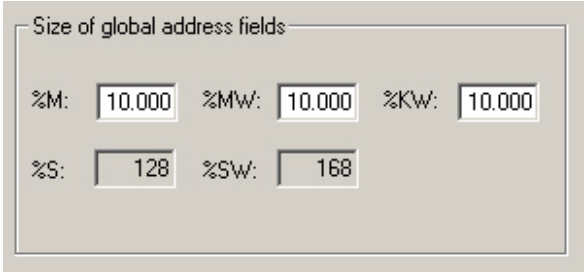
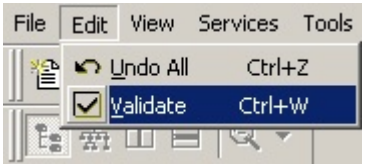
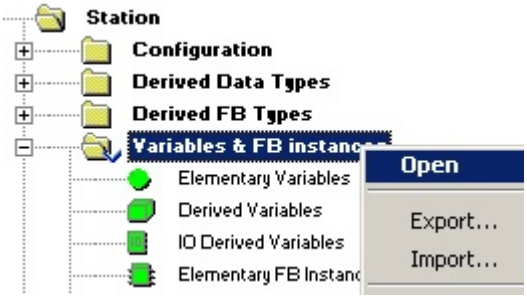
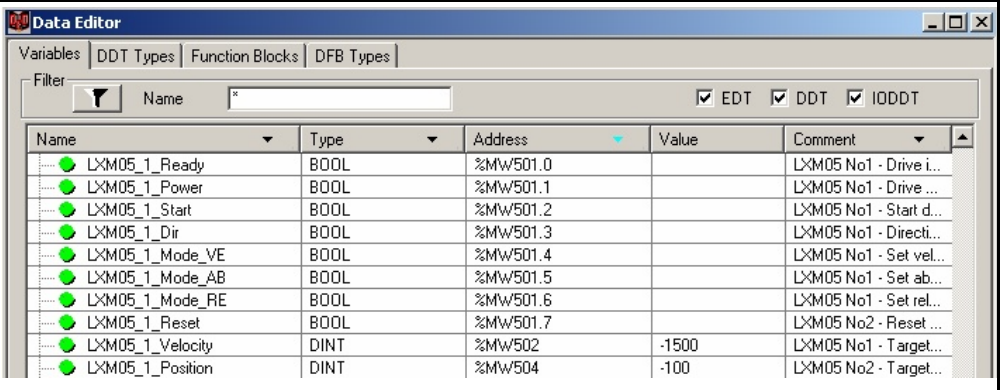
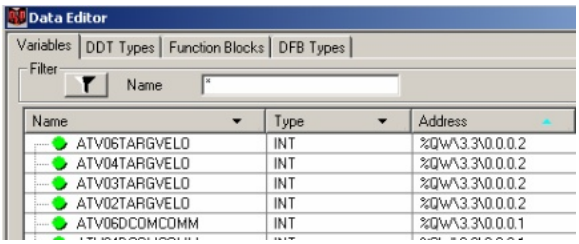
6	Click **Yes** to confirm the prompt that appears.	
7	Enter the **IP address** used on the **IP Configuration** tab. In this application, the following address is used: **192.168.100.50** **255.255.255.0** The HMI uses this address for data exchange, and Unity Pro uses it to connect to the PLC. **Note:** To be able to use this IP address, the rotary switch on the rear of the CPU must be set to the stored IP address. See Communication for further details.	
8	The entries must then be validated. To do this, click the **Tick** icon in the toolbar.	
9	Under **Communication** and **Networks**, a **red cross** indicates that the network is not assigned to any hardware. The Ethernet interface is available on the CPU being used here. Right-click on **Ethernet** and select **Open** to assign the CPU.	

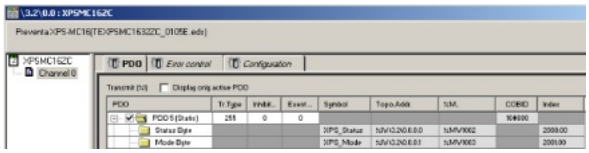
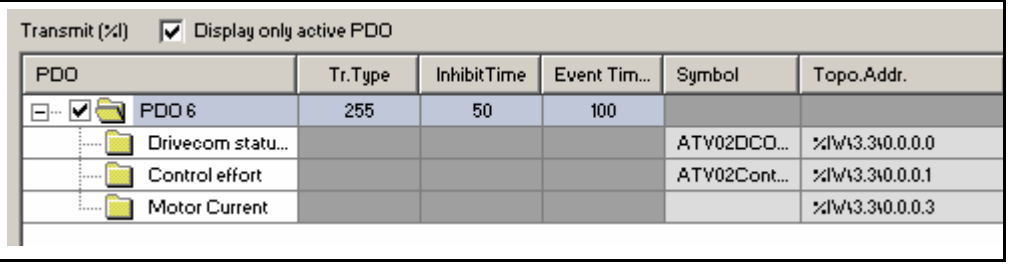
10	Under **Function**, select: **ETH_TCP_IP**.	
11	Then, under **Net Link**, assign the communication network **ETH** that was created previously. Finally, **validate** these entries as well.	
12	The red cross under **Networks** has now disappeared.	

13	To access the CANopen configuration, right-click **CANopen** in the project browser and select: **Open**.	
14	A **Transmission speed** (baud rate) of **500 kBaud** is used. Additionally, 1**250 words** are reserved for both **Inputs** and **Outputs**. The indices of the 1st %MWs are **1001** (Input) and **3001** (Output) respectively. **32** bits are reserved for each of the flags. Also 96 bits (%M) are reserved for input and output. the start addresses are 1001 (inputs) and 2001 (outputs)	
15	Once the application is closed, selecting **Build** will display the number of flags and words that are actually required.	**3. The configuration needs 1206 %MW IN.** **er 3. The configuration needs 1064 %MW OUT.** **The configuration needs 84 %M IN.** **3. The configuration needs 12 %M OUT.**

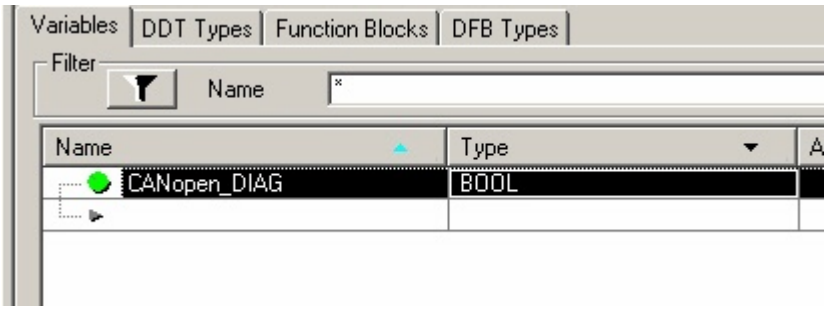
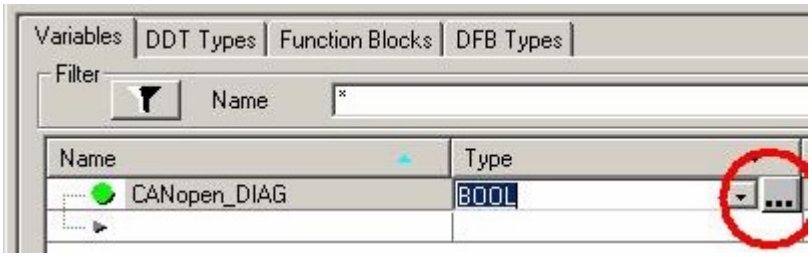
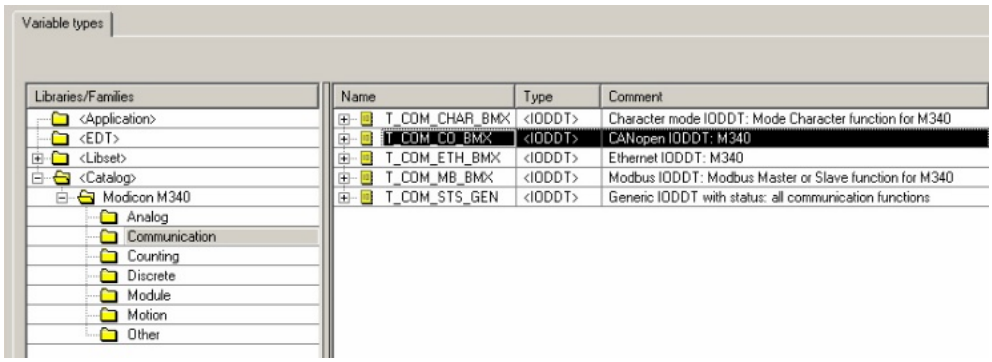
Creating New Variables

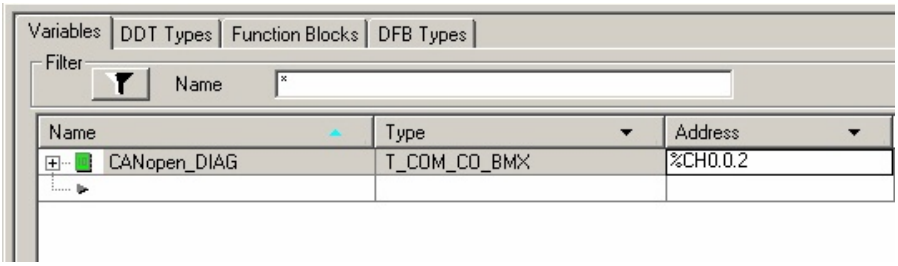
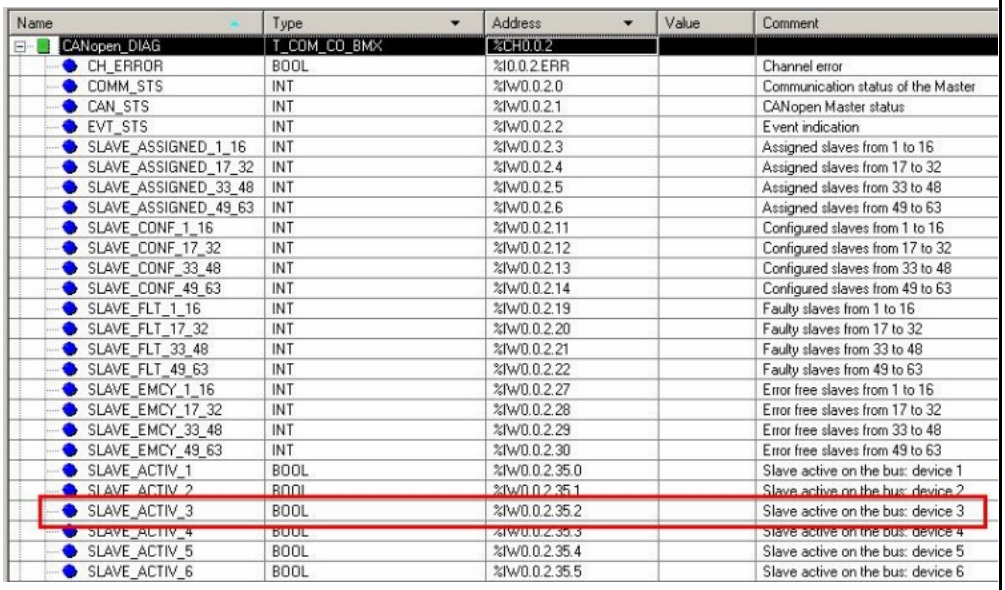
1	Addresses must be assigned to the variables for the purpose of data exchange with the HMI. The size of the addresses can be adjusted. To do this, right-click the **CPU** and select **Open**.	
2	The CPU properties appear.	

3	For this application, enter the following sizes for the individual **global address fields**: **%M 10000** **%MW 10000** **%KW 10000**	
4	Select **Validate** under **Edit** to validate the entries. Alternatively, you can click the icon on the toolbar	
5	Open the **Data Editor** by right-clicking **Variables & FB instances** and selecting **Open**.	
6	You can enter all variables in the **Data Editor**. To do this, enter the **variable name** in the **Name** column and the **variable type** in the **Type** column. An initial value can be set in the **Value** column. **To address the variables (located variables), an** address **must be entered in the** Address **column. The following addresses appear on the partial screenshot below:** %MW501.1 Bit 2 in word 501 %MW502 word 502 %I0.1.1 Digital input from rack 0; card 1 of input 1.	
7		
8	Here is an address for a **CANopen** node: %IW\3.3\0.0.1 or %IW\3.3\0.0.0.2 %IW Input word \3.3\ Card 3 (CANopen) CANopen address 3 0.0.0 1st word (also 0.0.0.0) 0.0.0.1 2nd word	

9	Once entered in the Data Editor, the variable name relating to the CANopen node is displayed in the **Symbol** column of the **PDO** tab.	
10		

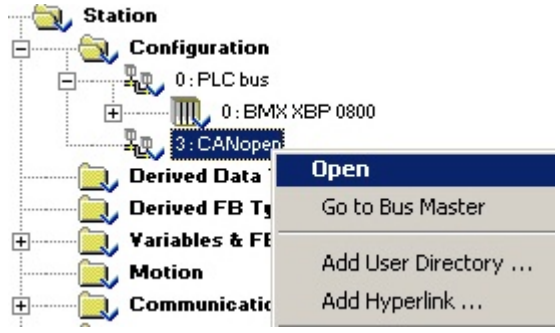
Create IODDT CANopen as Variable CANopen_DIAG

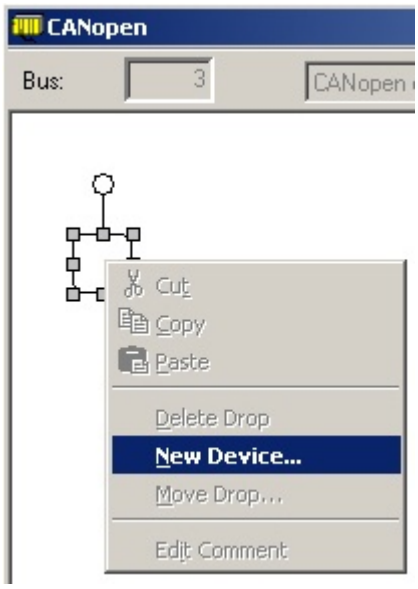
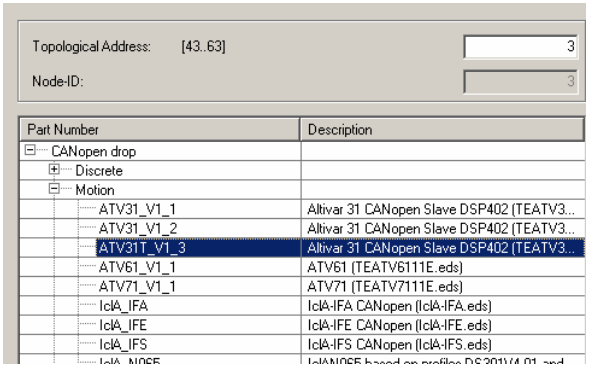
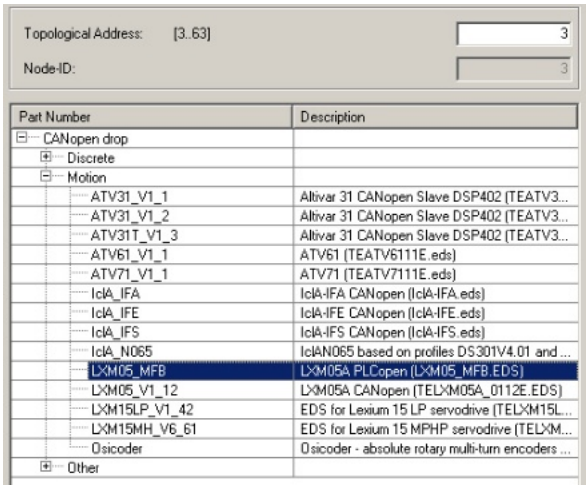
1	In the variable editor, create a CANopen diagnostic variable **CANopen_DIAG**. 
2	Use the icon (**...**) on the right of the **Type** field to select ther data type **IODDT**. 
3	Now select the variable **T_COM_CO_BMX** in the Library **Modicon M340**. Hit Enter to take on the variable for **CANopen_DIAG**. 

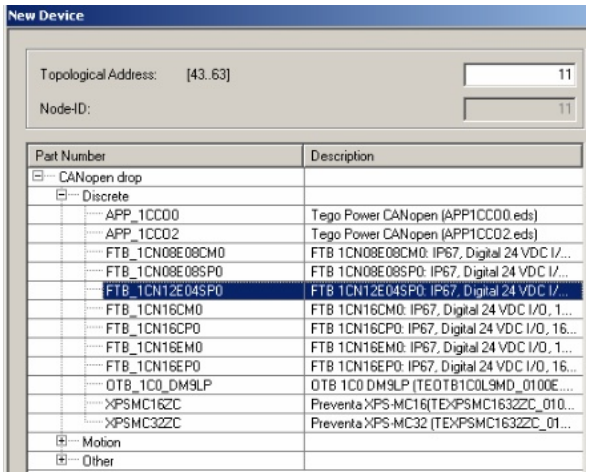
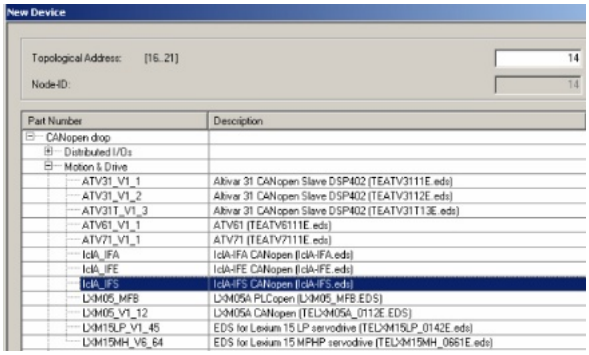
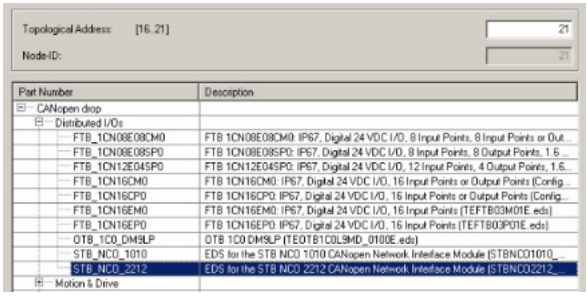
4	Now give the physical address for the **CANopen Master Channel %CH0.0.2**. 
5	Now you can access the variables of the CANopen Master, such as **SLAVE_ACTIV_3** and can use them in Unity. 
6	**Please consult the documentation for other address types.**

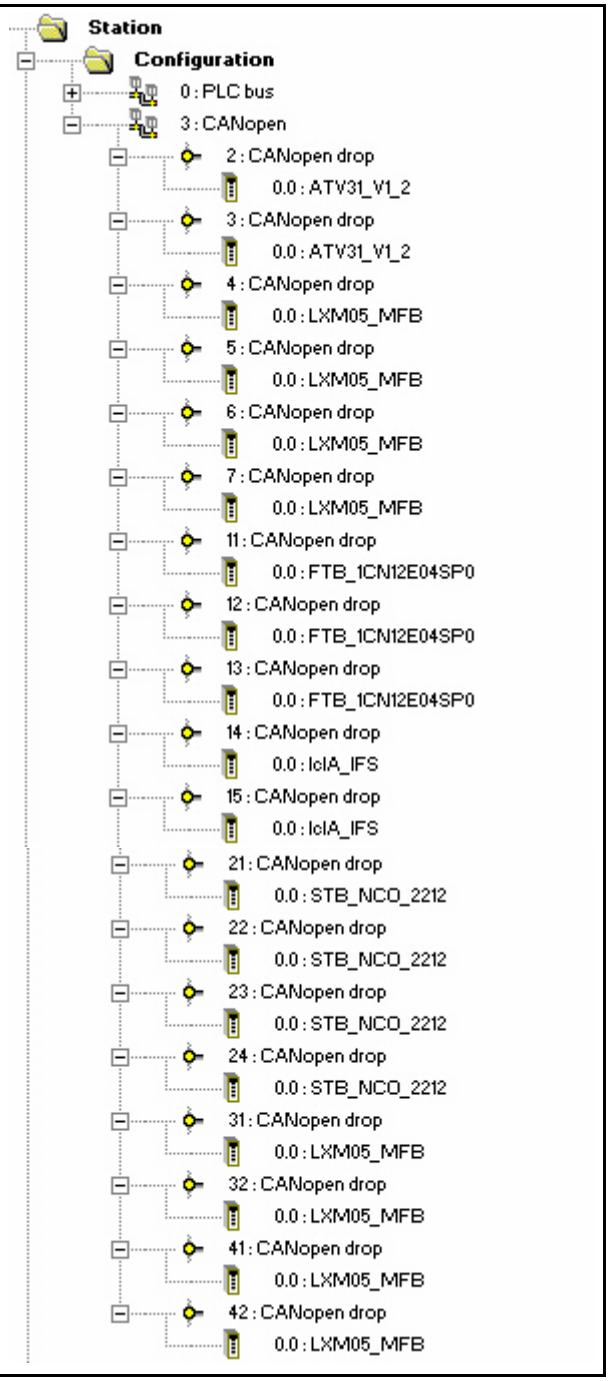
Name	Type	Address	Value	Comment
CANopen_DIAG	T_COM_CO_BMX	%CH0.0.2		
CH_ERROR	BOOL	%I0.0.2.ERR		Channel error
COMM_STS	INT	%IW0.0.2.0		Communication status of the Master
CAN_STS	INT	%IW0.0.2.1		CANopen Master status
EVT_STS	INT	%IW0.0.2.2		Event indication
SLAVE_ASSIGNED_1_16	INT	%IW0.0.2.3		Assigned slaves from 1 to 16
SLAVE_ASSIGNED_17_32	INT	%IW0.0.2.4		Assigned slaves from 17 to 32
SLAVE_ASSIGNED_33_48	INT	%IW0.0.2.5		Assigned slaves from 33 to 48
SLAVE_ASSIGNED_49_63	INT	%IW0.0.2.6		Assigned slaves from 49 to 63
SLAVE_CONF_1_16	INT	%IW0.0.2.11		Configured slaves from 1 to 16
SLAVE_CONF_17_32	INT	%IW0.0.2.12		Configured slaves from 17 to 32
SLAVE_CONF_33_48	INT	%IW0.0.2.13		Configured slaves from 33 to 48
SLAVE_CONF_49_63	INT	%IW0.0.2.14		Configured slaves from 49 to 63
SLAVE_FLT_1_16	INT	%IW0.0.2.19		Faulty slaves from 1 to 16
SLAVE_FLT_17_32	INT	%IW0.0.2.20		Faulty slaves from 17 to 32
SLAVE_FLT_33_48	INT	%IW0.0.2.21		Faulty slaves from 33 to 48
SLAVE_FLT_49_63	INT	%IW0.0.2.22		Faulty slaves from 49 to 63
SLAVE_EMCY_1_16	INT	%IW0.0.2.27		Error free slaves from 1 to 16
SLAVE_EMCY_17_32	INT	%IW0.0.2.28		Error free slaves from 17 to 32
SLAVE_EMCY_33_48	INT	%IW0.0.2.29		Error free slaves from 33 to 48
SLAVE_EMCY_49_63	INT	%IW0.0.2.30		Error free slaves from 49 to 63
SLAVE_ACTIV_1	BOOL	%IW0.0.2.35.0		Slave active on the bus: device 1
SLAVE_ACTIV_2	BOOL	%IW0.0.2.35.1		Slave active on the bus: device 2
SLAVE_ACTIV_3	BOOL	%IW0.0.2.35.2		Slave active on the bus: device 3
SLAVE_ACTIV_4	BOOL	%IW0.0.2.35.3		Slave active on the bus: device 4
SLAVE_ACTIV_5	BOOL	%IW0.0.2.35.4		Slave active on the bus: device 5
SLAVE_ACTIV_6	BOOL	%IW0.0.2.35.5		Slave active on the bus: device 6

Adding CANopen Nodes

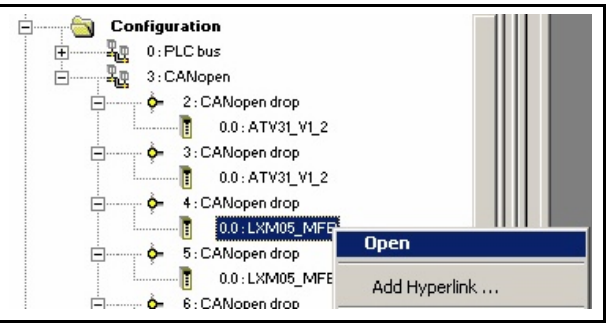
1	The CANopen bus window can be used to add up to 63 CANopen nodes. To do this, select **CANopen** in the project browser and **Open** from the pop-up menu.	

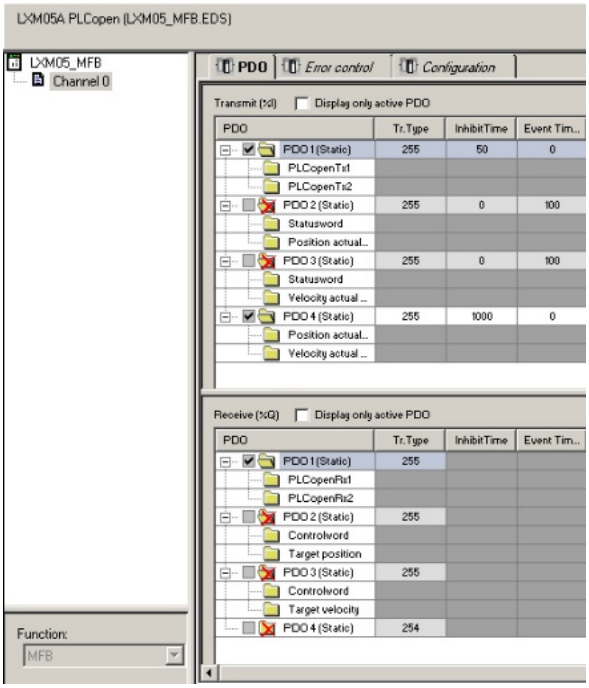
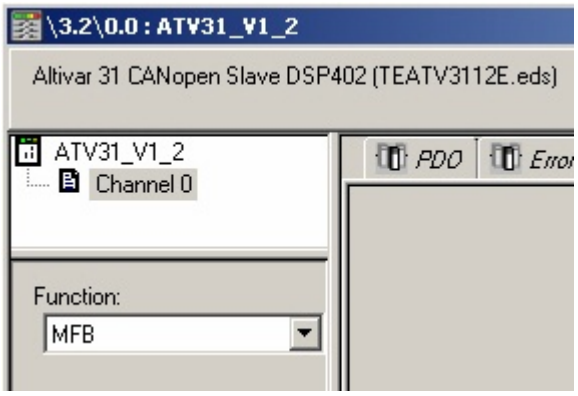
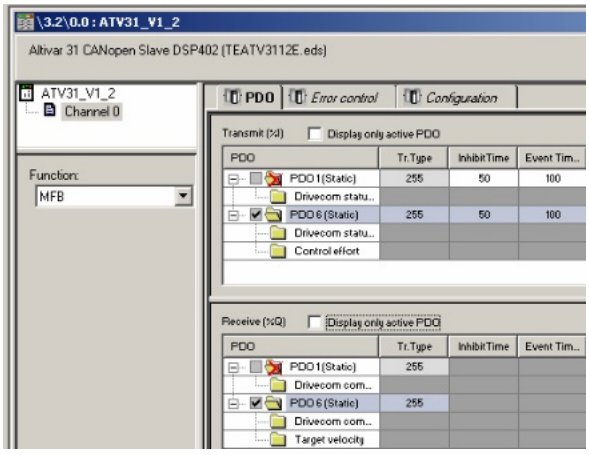
2	The CANopen window appears. Click the empty field and select **New Device** from the menu.	
3	For the two directly connected Altivar 31 drives, select **ATV31_V1_2** under **Motion**. Enter the values **2** and **3** for the **Address.** **Note:** The ATV31s in the remote cabinets are in the configuration of the STB I/O islands	
4	Since the eight Lexium 05 CANopen nodes are being controlled by MFB (Motion Function Block), you must select **LXM05_MFB** under **Motion**. As **Topological Address** insert: **3...7,31,32,41,42**	

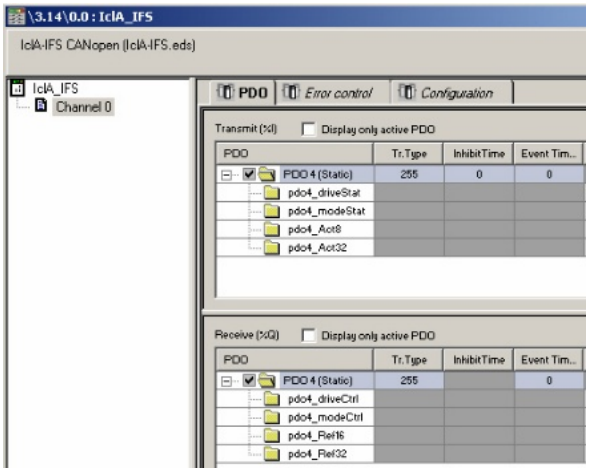
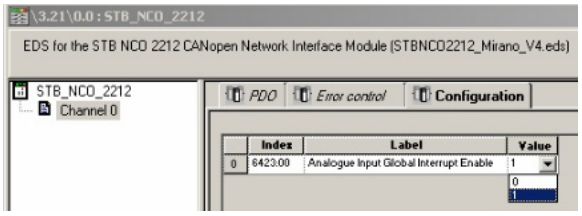
5	The 3 remote FTB Islands in the field can be selected via **Discrete** with **FTB_1CN12E04SP0** As **Topological Address** insert **11…13**	
6	The two IclA IFS drives in the field can be defined via **Motion&Drives** and **IclA_IFS**. As **Topological Address** insert **14…15**	
7	the four remote STB islands in the field can be selected with **Discrete** and **STB_NCO_2212**. As **Topological Address** input: **21…24**	

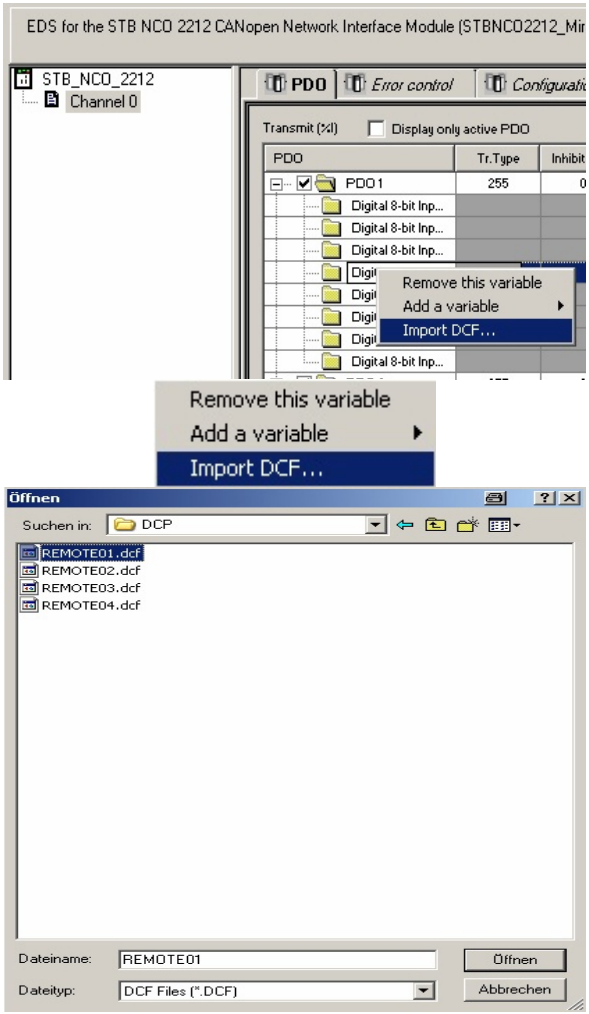
8	All CANopen nodes should now be displayed in the project browser..	

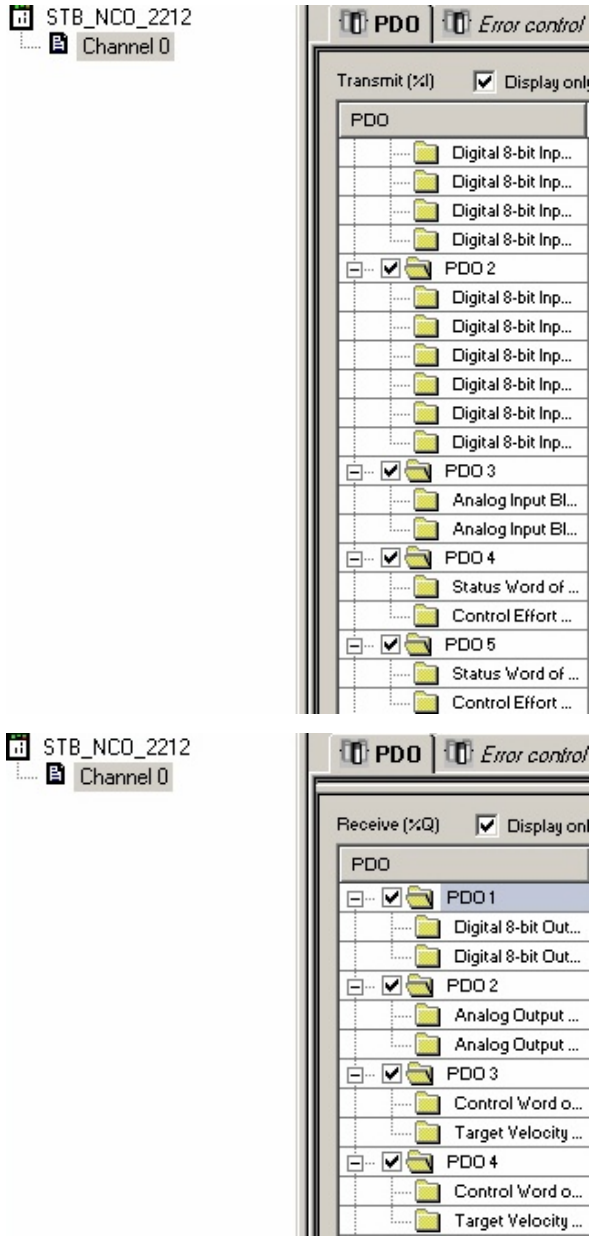
CANopen PDO Parametrierung

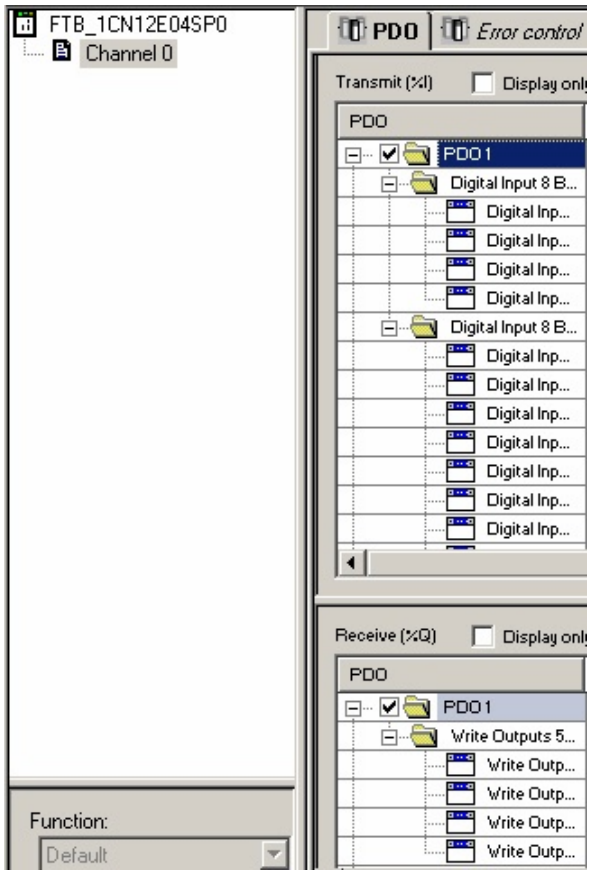
1	You must now parameterize the cyclic data exchange that takes place via the PDOs. To do this, select **Open** from the pop-up menu for each node and go to the **PDO** tab in the window that appears.	

2	Since the **Lexium 05** servo drive is controlled via MFB, no changes can be made to the PDOs. These are permanently pre-set as follows: **PDO 1 transmit** **PDO 4 transmit** **PDO 1 receive**	
3	MFB is also used with the **Altivar 31** variable speed drive. To activate this, you must select **MFB** in the **Function** drop-down list.	
4	The PDOs will then be set as follows: **PDO 6 transmit** **PDO 6 receive**	

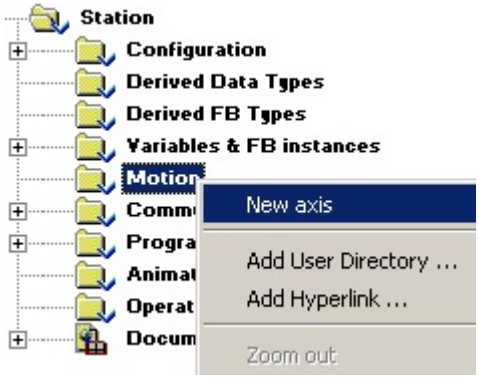
5	MFB is also used with the **IcLA IFS compact drive**. To activate this, you must select **MFB** in the **Function** drop-down list. The PDOs are fixed and have their own default values: **PDO 4 transmit** **PDO 4 receive**	
6	For the remote STBs in the field, select **Advanced** as **Function**. For example, here in the tab **Configuration** the **CANopen Object Index 6423:00 is set to „1“** to activate the analog value.	

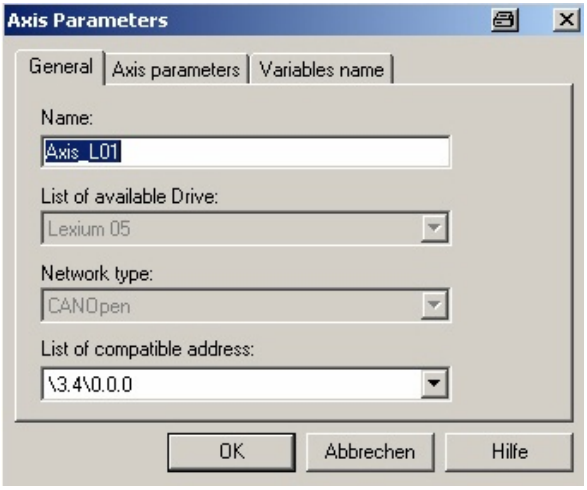
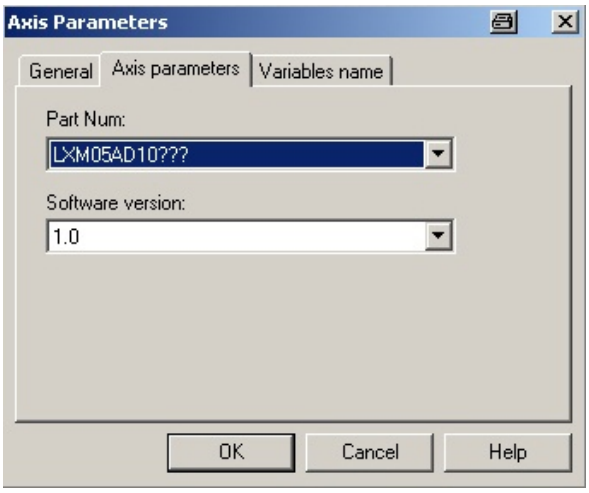
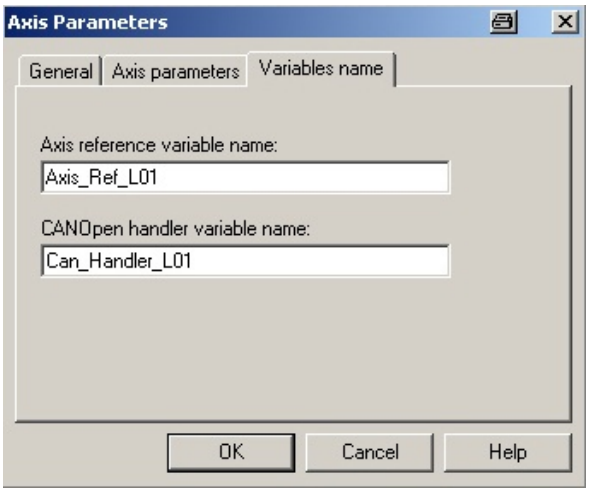
7	On the tab **PDO** right mouse click on **Import DCF…** opens the file open dialog. Select the first DCF file of the remote station : **REMOTE01**. Note: See the chapter on the Advantys configuration software	

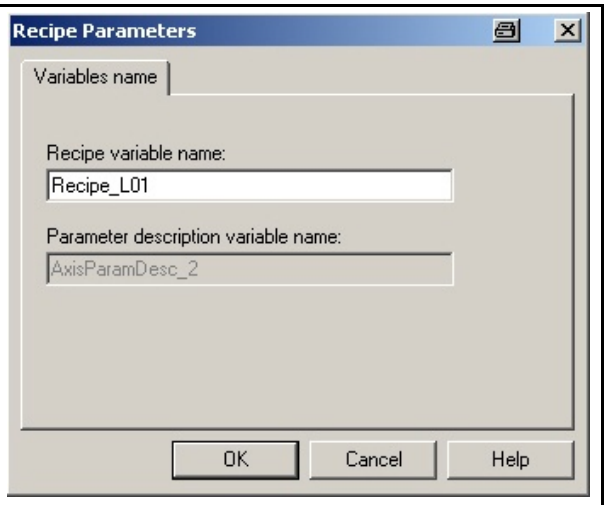
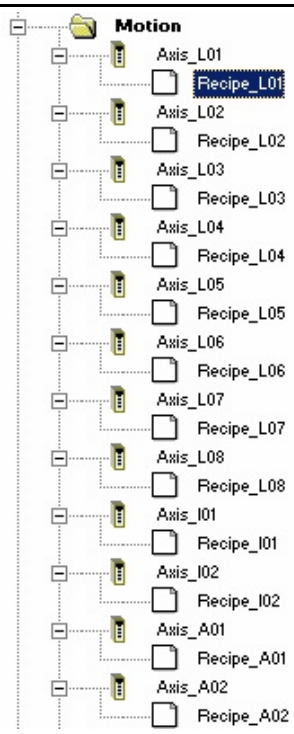
8	The PDOs are fixed: **PDO 1..5 transmit** **PDO 1..4 receive**	
9	For the remote **FTB** Islands the **function** is set to **Default** and cannot be changed.	

10	The PDOs for the FTB are fixed at: **PDO 1 transmit** **PDO 1 receive**	

Setting Up Axes for the Drives

1	To use the Lexium05 drives, the IcLA and the ATV31 with MFB (Motion Function Block) you must set up the the axis. To do this, select **Motion** in the project browser followed by **New axis** from the pop-up menu.	

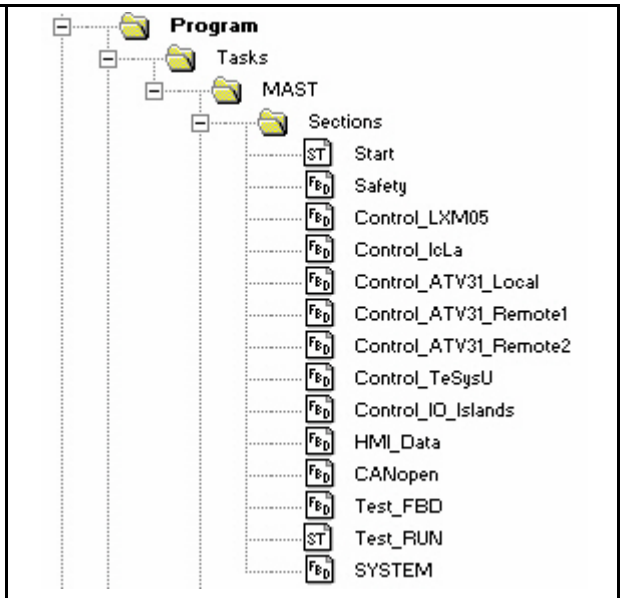
2	Note: An axis consists of the following configuration steps: **1. General Inputs** **2. Axis parameters** **3. Variables Name** **4. Recepe parameters** For the eight **Lexium 05** servo drives, first assign an axis name and a drive type on the **General** tab: Name: **AXIS_L01 … 08** Drive Type: **Lexium 05** Address: **\3.4\ 0.0.x … \3.7\ 0.0.x** **3.31\ 0.0.x … \3.32\ 0.0.x** **3.41\ 0.0.x … \3.42\ 0.0.x**	
3	On the Tab **Axis parameters** insert: **Part Num**: **LXM05AD10???** **Software Version**: **1.0**	
4	On the tab **Variable name** insert: Axis_Ref_ L0 **… 7** Can_Handler_ L**0 … 7**	

5	In the **Variables name** tab, Insert a **Recipe variable name** Each recipe contains a set of parameters for an axis. In this example there is one set for each axis. The xis recipes are named as follows: **Recipe_A01 .. L02** for the ATV31 **Recipe_L01 .. L08** for the LXM05 **Recipe_I01 .. I02** for the IcLa	
6	Listed under **Motion** you can now see the axis.	

7 The table below lists a summary of the drive entries.

drive	CANopen Address	Variable names of Axis for				
		Axis Name	Axis_Ref	CAN_Handler	Recipe	AxisParam
1. LXM05	\3.4\	_L01	_L01	_L01	_L01	Desc_1
2. LXM05	\3.5\	_L02	_L02	_L02	_L02	Desc_2
3. LXM05	\3.6\	_L03	_L03	_L03	_L03	Desc_3
4. LXM05	\3.7\	_L04	_L04	_L04	_L04	Desc_4
5. LXM05	\3.31\	_L05	_L05	_L05	_L05	Desc_5
6. LXM05	\3.32\	_L06	_L06	_L06	_L06	Desc_6
7. LXM05	\3.41\	_L07	_L07	_L07	_L07	Desc_7
8. LXM05	\3.42\	_L08	_L08	_L08	_L08	Desc_8
1. ATV31	\3.2\	_A01	_A01	_A01	_A01	Desc_1
2. ATV31	\3.3\	_A02	_A02	_A02	_A02	Desc_2
1. ICLA	\3.14\	_I01	_I01	_I01	_I01	Desc_1
2. ICLA	\3.15\	_I02	_I02	_I02	_I02	Desc_2

Program Assignment

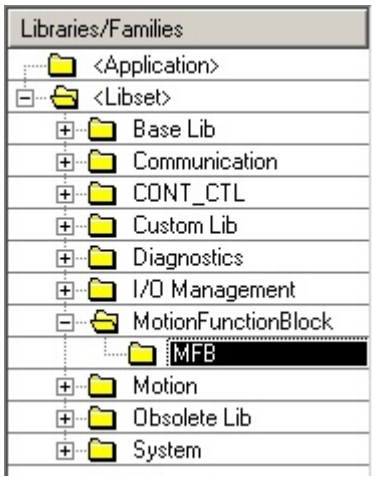
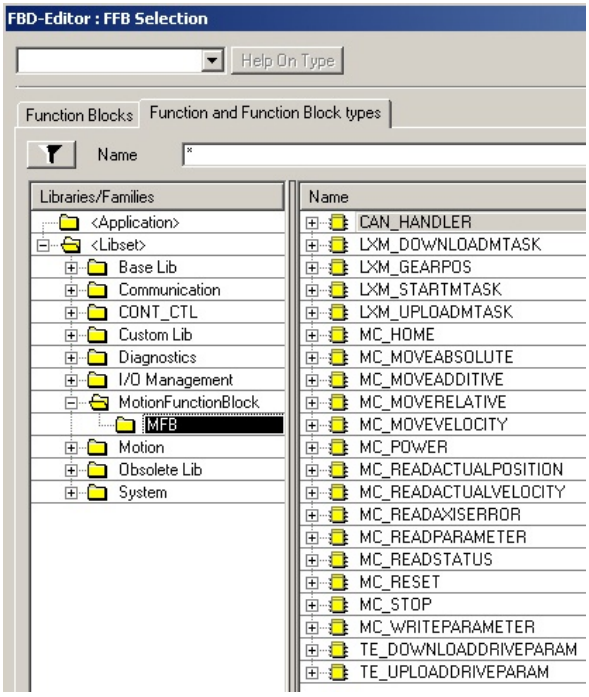
1	The individual program sections are displayed under **Program** in the project browser as shown here.	

2 Here is a brief summary:

Start	Consists of the functions that must be carried out when the application program is started (e.g., initialize variables).
Safety	Analyzes the safety information
Control_LXM05 Control_IcLa Control_ATV31	These sections are responsible for controlling the Lexium 05 drives , the IcLA and Altivar 31 with the Motion Function Block.
Control_ATV31_Remote1 und 2	These sections inlcude the function block (ATV) for the Altivar 31 in the remote STB islands.
Control_TeSysU	Conventional control of two TeSysU motor starters.
Control_IO_Islands	I/O Output for the STB and FTB islands.
HMI_Datas	Manages the communication and data exchange with the HMI.
CANopen	Summarizes the CANopen information of each node.

MFB

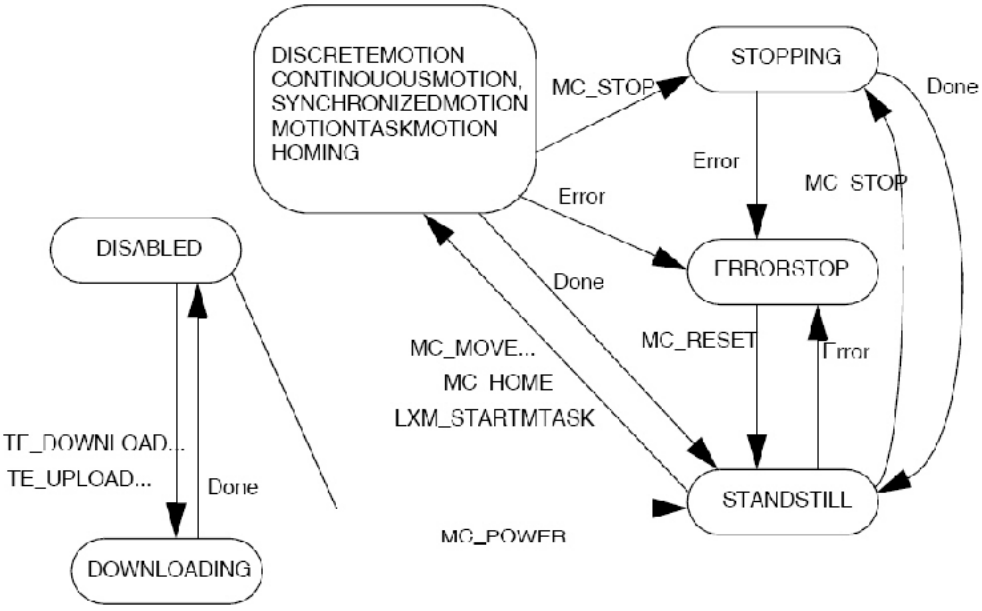
Motion Function Block

1	The Motion Function Block library contains function blocks for the straightforward control of servo drives and variable speed drives.	
2	These are listed in the **FBD-Editor** under **MotionFunctionBlock** and **MFB**.	

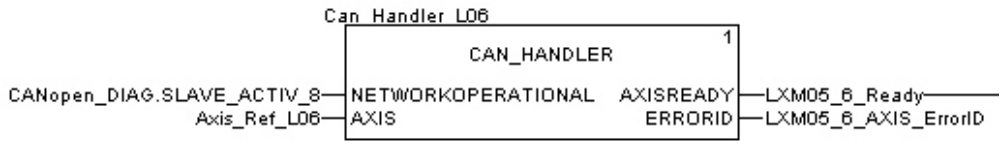
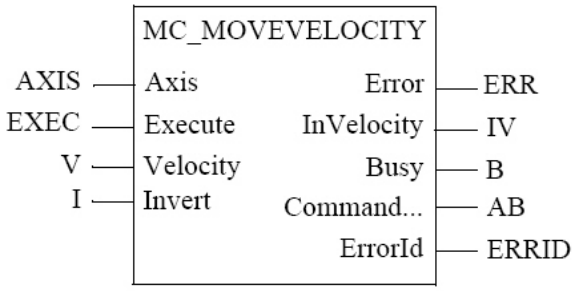
3 The following assignment table shows which blocks are available for which drive types:

Type	Block name	Lexium15 HP, MP, LP	Icla IFA, IFE, IFX	ATV31	ATV71	Lexium05
PLCopen	MC_ReadParameter	X	X	X	X	X
	MC_WriteParameter	X	X	X	X	X
	MC_ReadActualPosition	X	X			X
	MC_ReadActualVelocity	X	X	X	X	X
	MC_Reset	X	X	X	X	X
	MC_Stop	X	X	X	X	X
	MC_Power	X	X	X	X	X
	MC_MoveAbsolute	X	X			X
	MC_MoveRelative	X				X
	MC_MoveAdditive		X			X
	MC_MoveVelocity	X	X	X	X	X
	MC_ReadAxisError	X	X	X	X	X
	MC_ReadStatus	X	X	X	X	X
	MC_Home	X	X			X
Parameter set save and restore functions for management of recipes or replacement of faulty servodrives	TE_UploadDriveParam	X	X	X	X	X
	TE_DownloadDriveParam	X	X	X	X	X
Advanced functions for the Lexium 15	Lxm_GearPos	X				
	Lxm_DownloadMTask	X				
	Lxm_UploadMTask	X				
	Lxm_StartMTask	X				
System function	CAN_Handler	X	X	X	X	X

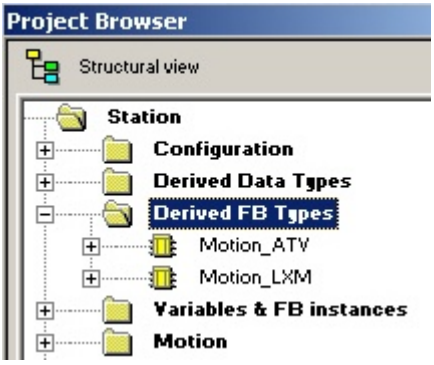
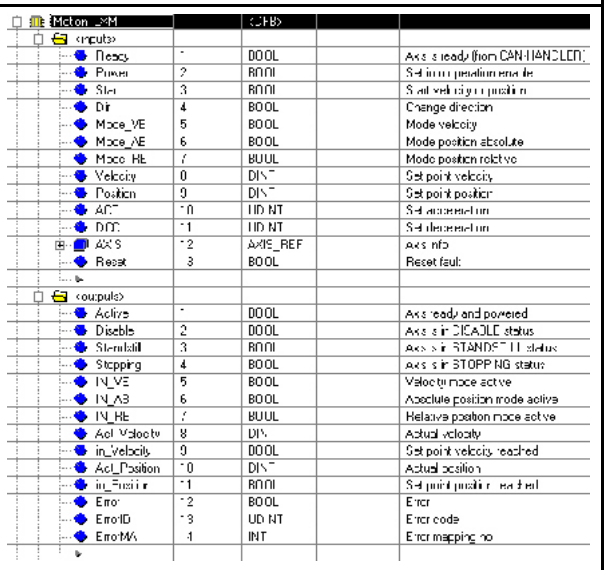
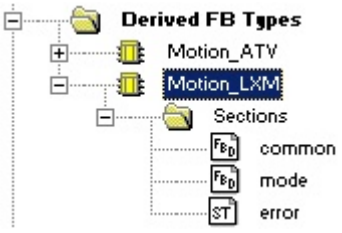
4 You can use the blocks to switch between particular operating states (see image below). After a drive is switched on, its status is normally **Disabled**.

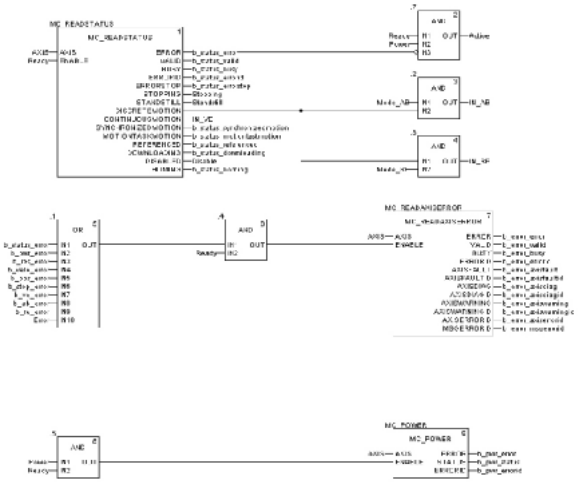
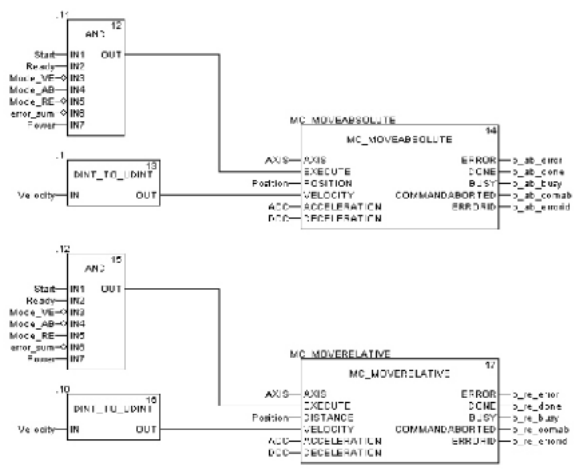


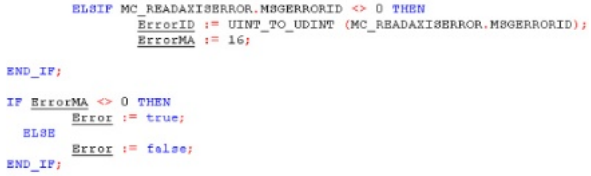
5 A **CAN_HANDLER** is absolutely essential for each drive and must be addressed in each PLC cycle. The block uses the **AXISREADY** output to indicate whether the axis/drive is available for control. The axes described above are used as parameters.

6		
7	The image opposite shows another block which, with the assistance of the axis parameters, can control a drive using CANopen. Please refer to the block and startup documentation for an exact description.	

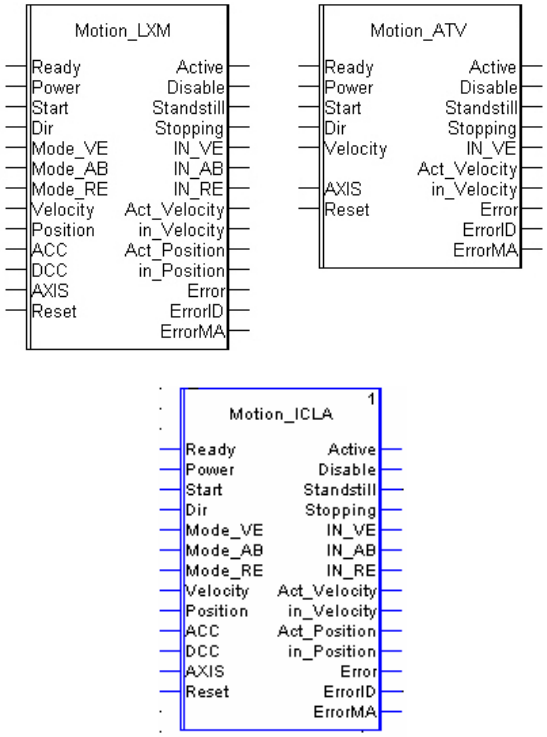
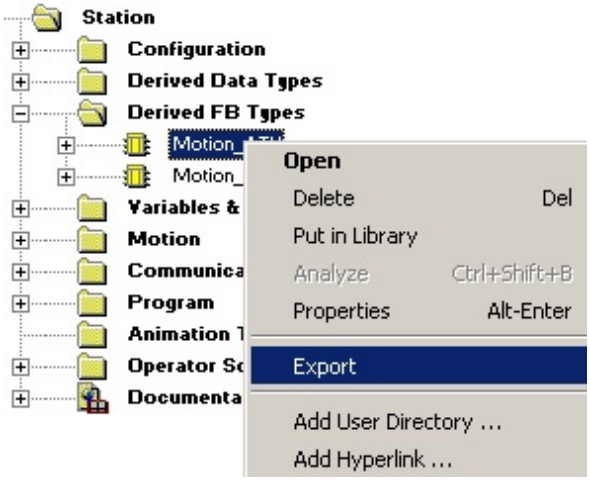
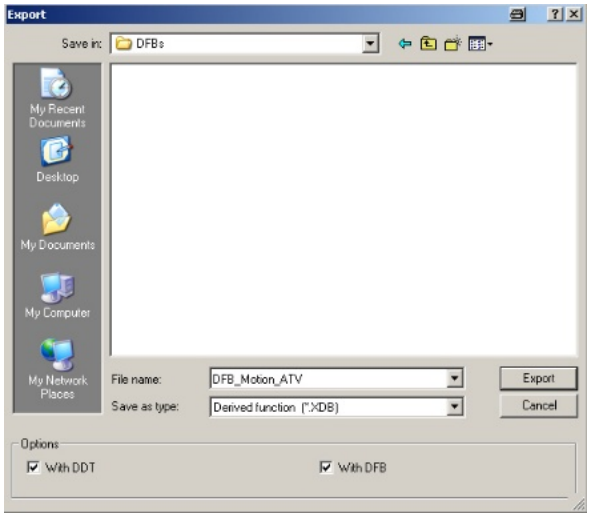
DFB erstellen und nutzen

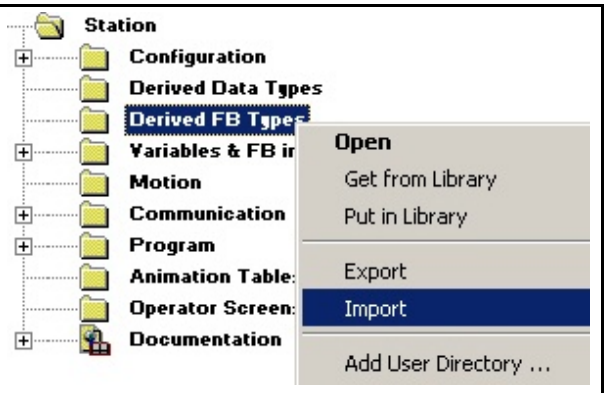
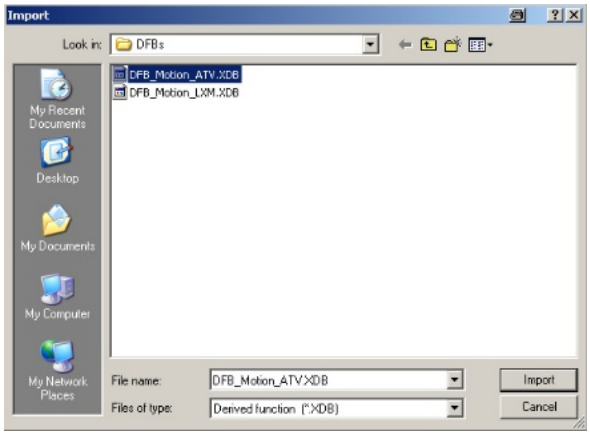
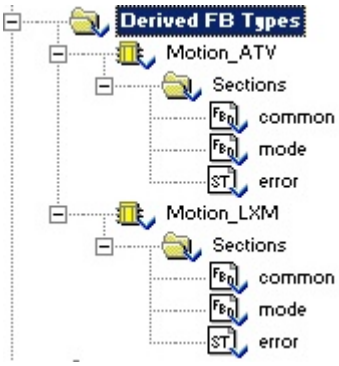
1	To get a compact and clear overview of the configuration, it is possible to group entire functions in a DFB. Two DFBs have been created in this application, each of which contains the Altivar, IcLA and Lexium MFBs mentioned above.	
2	First, specify the block inputs and outputs. You can determine the position on the block using the number entered.	
3	In the **Motion_LXM** block, for instance, there are three sections available.	

4	Common control commands are processed in the **common** section. These are: • Block status • Axis error message • Power connection • Error acknowledgement • Actual velocity and • Actual position	
5	The **mode** section consists of: • Stop drive • Velocity mode • Absolute positioning mode • Relative positioning mode	

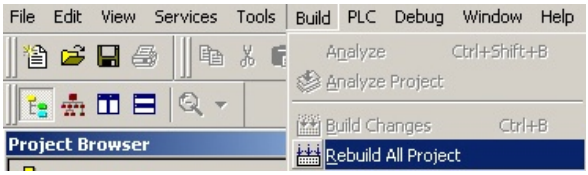
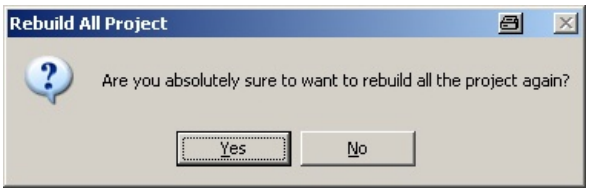
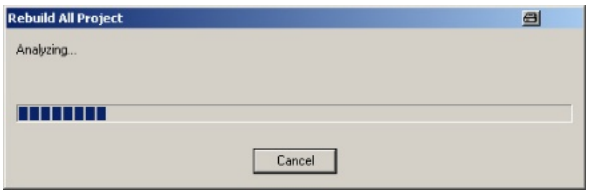
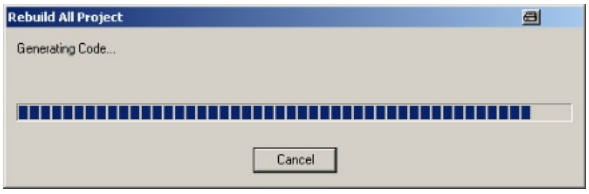
6	The **error** section provides a summary of the error messages	
		ELSIF MC_READAXISERROR.MBGERRORID <> 0 THEN
		ErrorID := UINT_TO_UDINT (MC_READAXISERROR.MBGERRORID);
		ErrorMA := 16;
		END_IF;
		
		IF ErrorMA <> 0 THEN
		Error := true;
		ELSE
		Error := false;
		END_IF;

Required Blocks

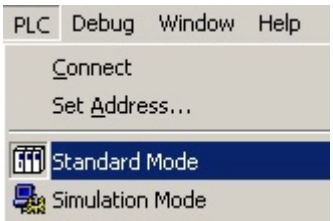
1	As well as the standard blocks, three DFBs are used in the application. These are: • **Motion_LXM** for Lexium05 • **Motion_ICLA** for IcLa IFS • **Motion_ATV** for Altivar 31. In the case of Motion_LXM and Motion_ICLA, positioning is also possible.	
2	These can be exported separately by right-clicking on the corresponding DFB and selecting **Export**.	
3	You can select any **directory** and **file name** here. The file name extension is **.XDB**.	

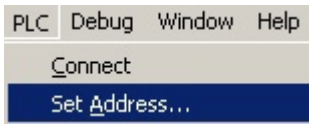
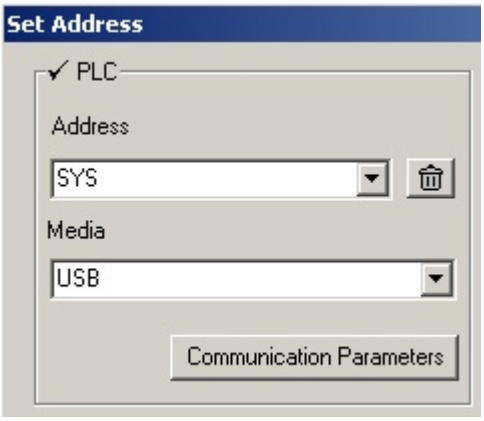
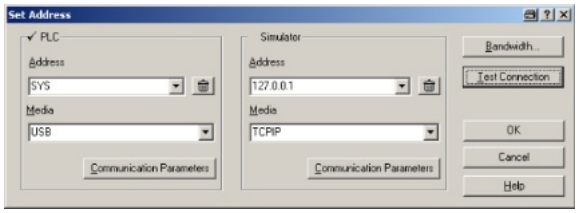
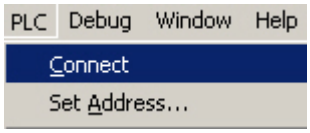
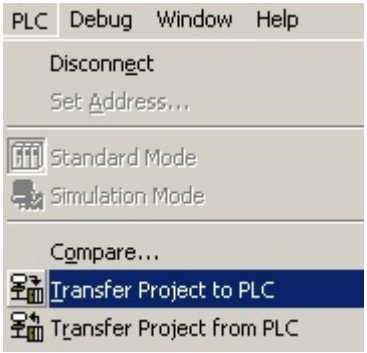
4	These can be imported into a new project at any time. To do this, select **Import** from the menu.	
5	Click **Yes** to confirm the **modification** and **save project** messages.	
6	Select the relevant file and click **Import**	
7	The DFBs are displayed in the **Derived FB Types** directory	

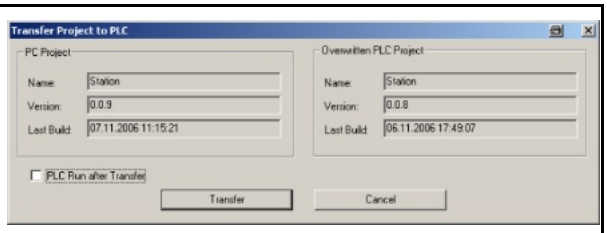
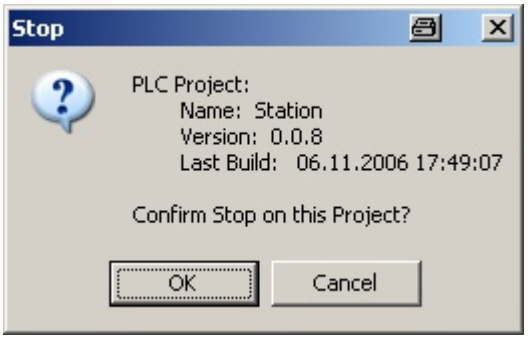
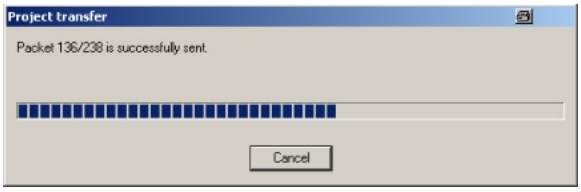
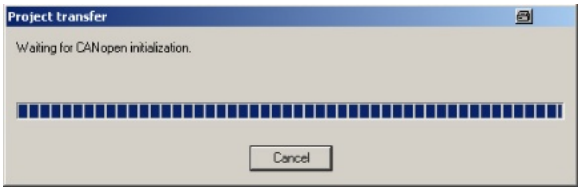
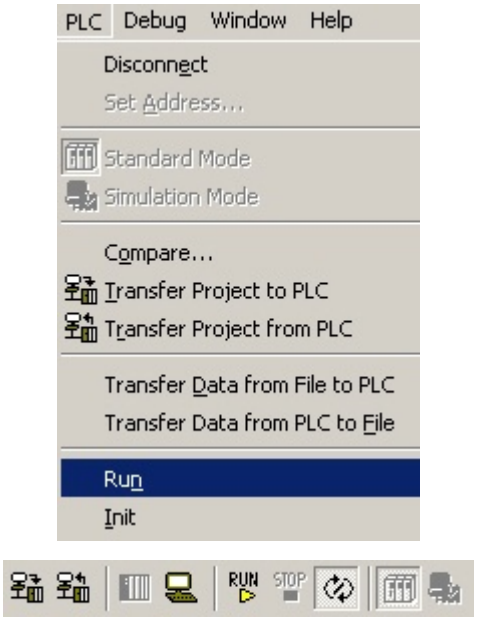
Building a Project

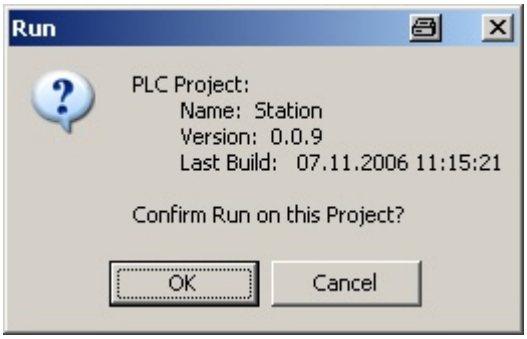
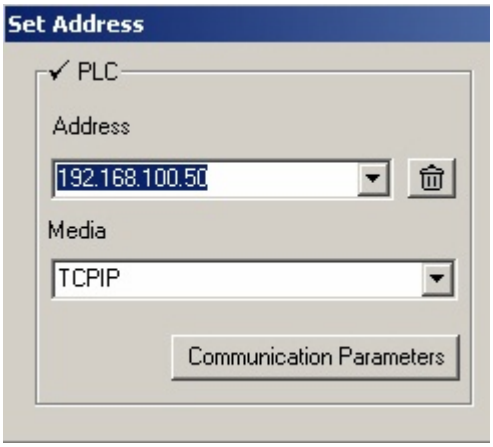
1	A project must be analyzed and compiled before it can be transferred to the PLC. To do this, select **Build** and **Rebuild All Project** in the menu bar. Alternatively, click the corresponding icon in the toolbar.	 
2	Click **Yes** to confirm the message that follows.	
3	The project is analyzed and the code generated.	 
4	Once this is complete, the number of errors and warnings is displayed. A box displaying **Built** can also be seen in the bottom right-hand corner of the Unity window.	 

Connecting the PC to the PLC and Transferring a Project

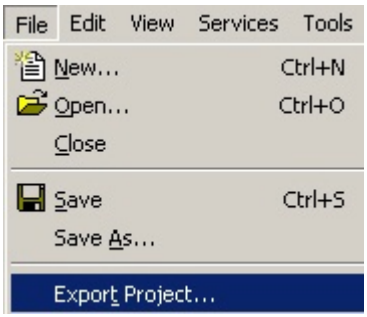
1	To establish a connection to the PLC, **Standard Mode** must first be activated.	
2	If the PLC is connected to the PC via the USB cable, an icon indicating this will be displayed in the PC status bar. The Modicon M340 – **BMX CPU** is displayed in the Windows screen.	 

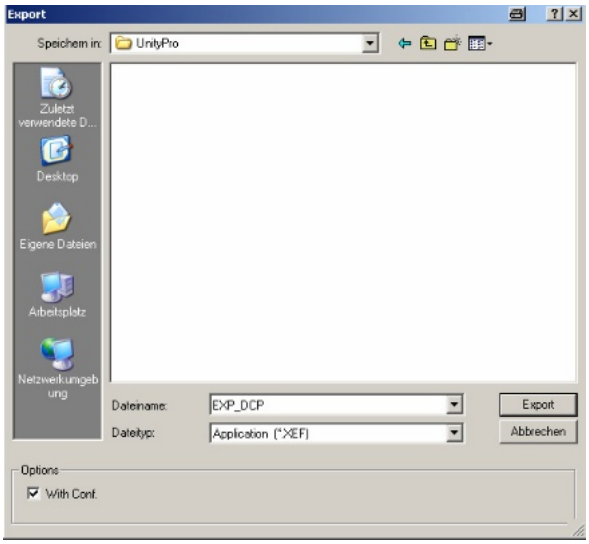
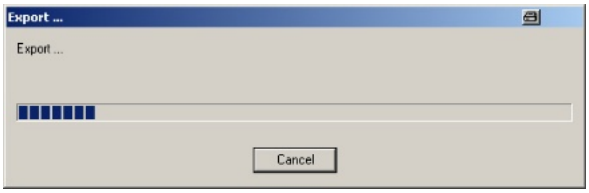
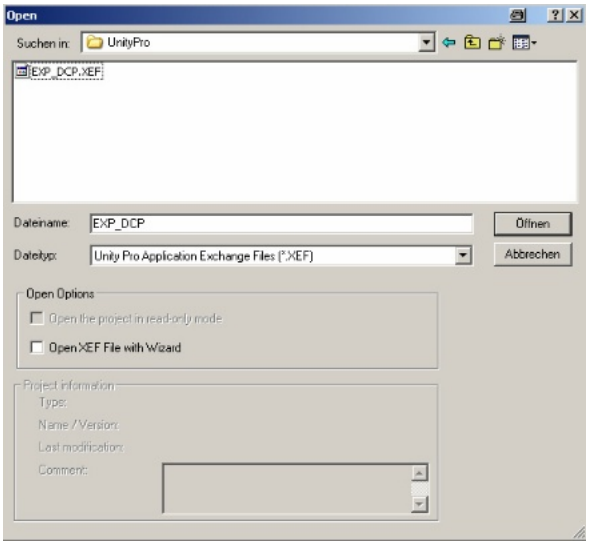
3	Select **PLC->Set Address** to set the address.	
4	The following parameters are set for a USB connection: Address: **SYS** Media: **USB**	
5	These entries can be tested directly. To do this, click **Test Connection** on the right-hand side. A message window will appear to indicate that connection has been successful. Click **OK** to confirm. Close the **Set Address** window by clicking **OK**.	 
6	In Unity Pro, the mode of connection that has been selected is displayed in the status bar at the bottom.	
7	Select **PLC->Connect** to connect to the PLC.	
8	The status bar shows that the PLC status is set to **RUN** and that the current program is not the same as the one in the PLC **(DIFFERENT)**.	
9	Select: **PLC->Transfer Project to PLC** to download the project.	

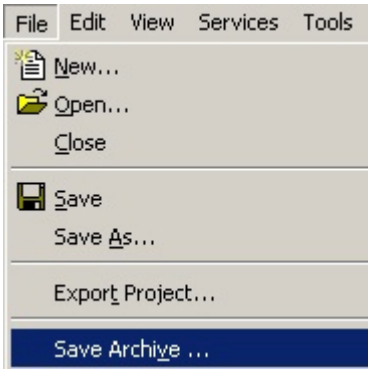
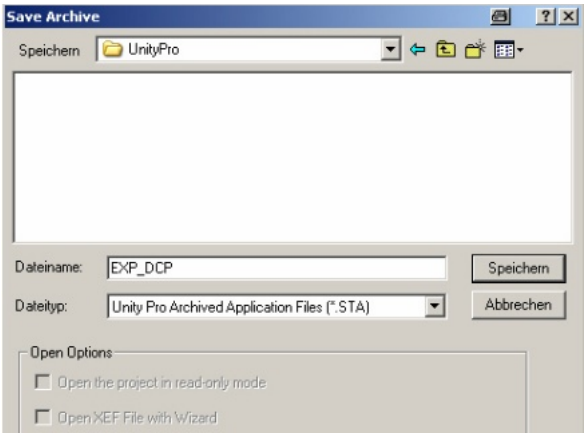
10	Both the PC and PLC projects, along with their version and date, are displayed in the window that opens next. Click **Transfer** to start the download process.	
11	The project running on the PLC must be stopped. Click **OK** to continue.	
12	The project is transferred and the CANopen bus initialized.	 
13	The status bar shows that the project is the same **(EQUAL**), but that it is still in **STOP** status.	
14	Select **PLC->Run** to start the program.	

15	Click **OK** to confirm.	
16	**The project begins to run.**	
17	If an IP address has been configured, it can be used to establish a connection between the PC and PLC. To do this, enter the **IP address** in the Address field in the Set Address window, and select **TCPIP** under Media..	
18	The IP address is displayed in the status bar.	

Exporting and Archiving a Project

1	The following are exported as part of a project export: • Input/output configuration • Sections • SR program modules • Event processing • Unprotected DFB types • DDTs • Variables • Animation tables • References to protected DFB types To perform an export, select **File->Export Project** via the menu bar.	

2	When a project is exported, the software generates a ***.XEF** file. You can select any location in which to save the file and any file name. Click **Export** to begin exporting.	
3	The project is exported. The progress bar is displayed.	
4	An exported project can be opened directly with UnityPro.	
5	As well as the XEF export file and the STU project file, there is an STA project archive. The properties of the STA file are as follows: • The STA file is highly compressed (around 50 times more than the STU file). It is used to transfer projects to networks (e.g, local or Internet networks). • The STA file can be used to transfer projects between different versions of the Unity Pro software. • The STA file contains the entire project: - The PLC binary files - The read-out information Comments and animation tables - The operator screen	

6	If an STA file is selected, the software offers a certain amount of information: • Project name • Accompanying comment • Version and date of project generation • The project's target PLC • The date when the source code was last changed. • The version of Unity Pro used to generate this archive. Select **File->Save Archive** via the menu bar.	
7	Select the location for saving the file and the file name. Click **Save** to begin archiving.	

Magelis XBTGT

Introduction

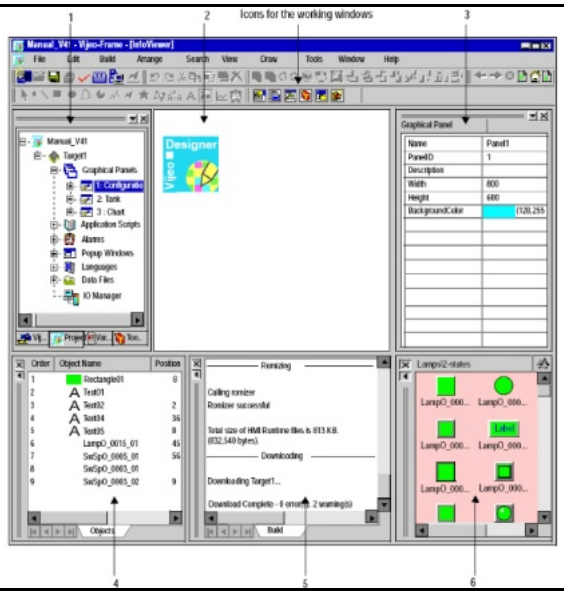
This application features a Magelis XBT-GT 2330 HMI, which is connected to the PLC via the Modbus TCP/IP protocol.

Vijeo Designer software is used to program and configure the terminal. The steps to be taken in order to create and download a program are described on the pages that follow.

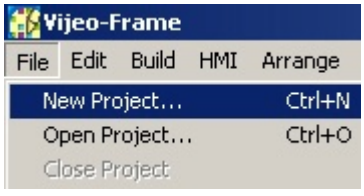
Setting up the HMI is done as follows:

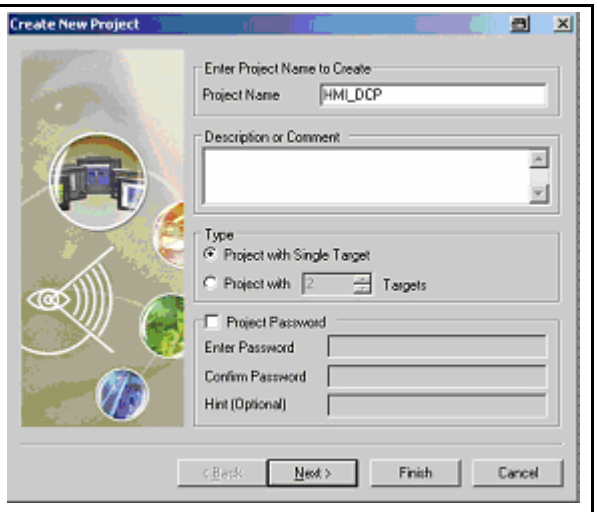
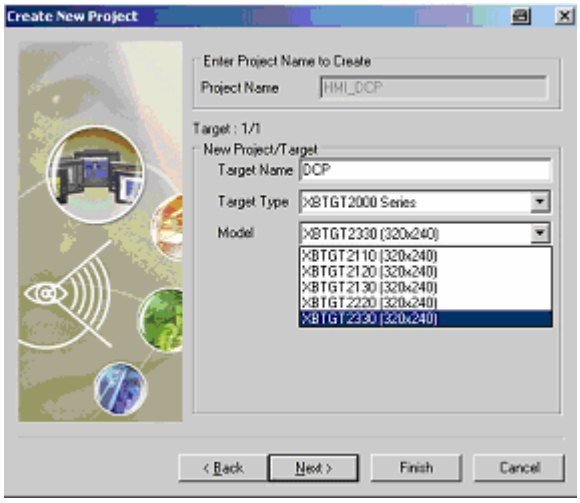
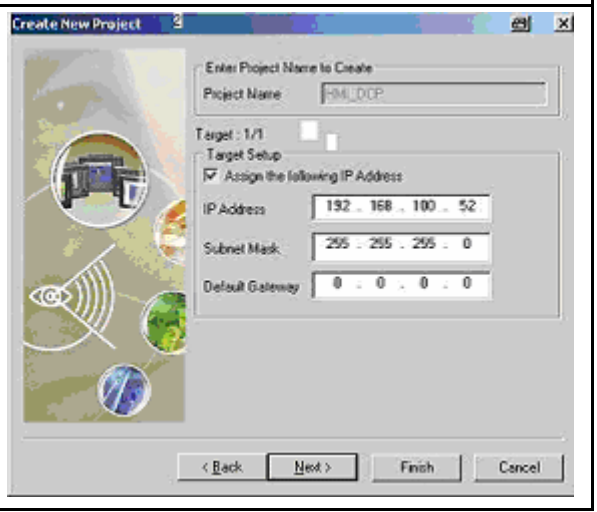
- Vijeo Designer function overview
- Create new project (specify platform, hardware, communication).
- Communication settings
- Create new variables
- Create screens
- Display error messages
- Check the project and download it
- Application overview

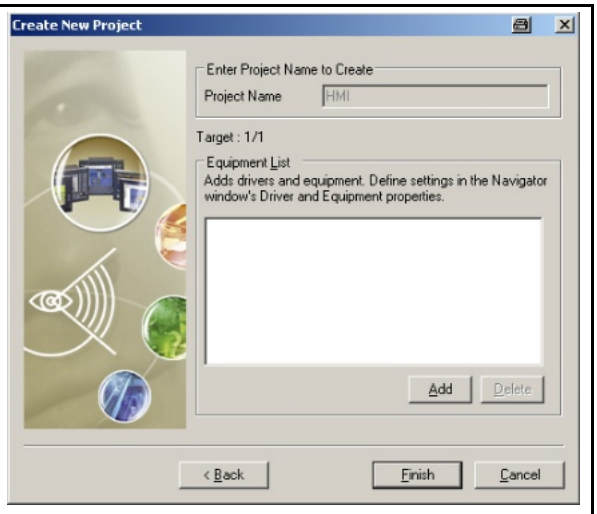
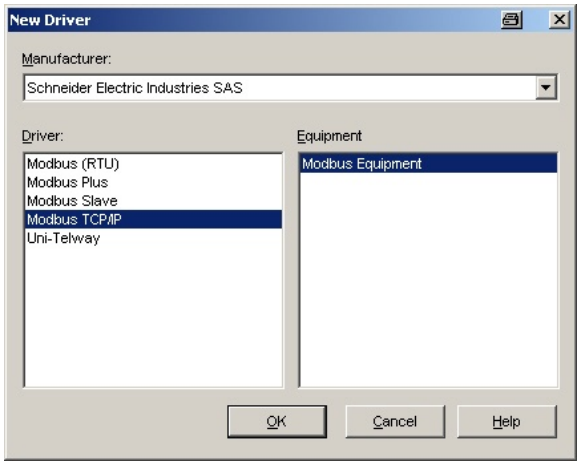
Function Overview

1	The Vijeo Designer environment consists of the following elements: 1 Navigator 2 Information display 3 Inspector 4 Data list 5 Feedback area 6 Toolbox	

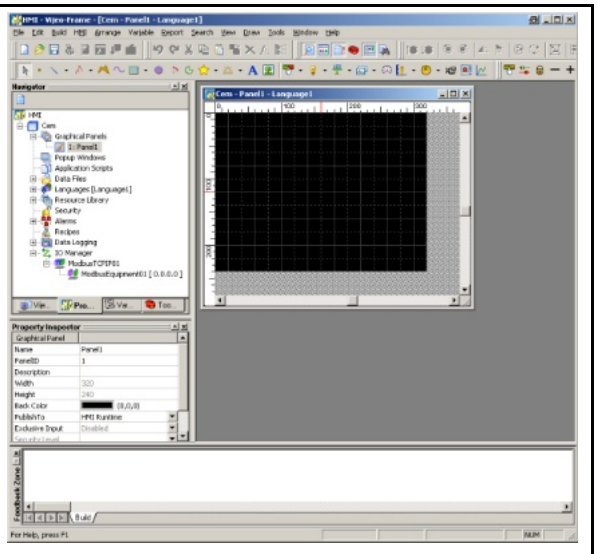
Creating a New Project

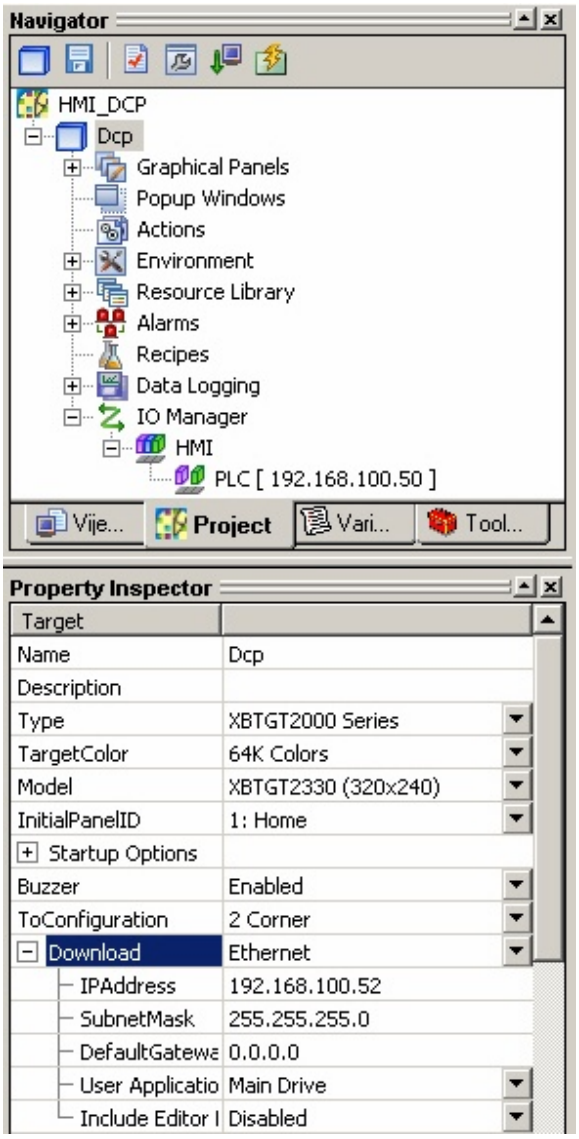
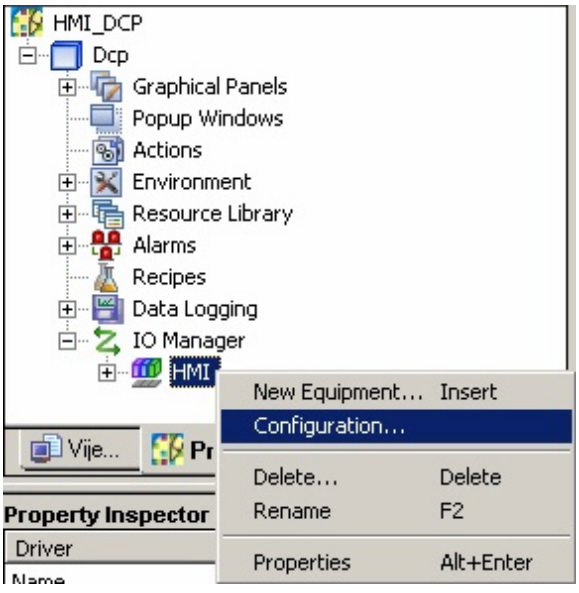
1	After starting Vijeo Designer, a new project can be created. To do this, select, **File->New Project** in the menu bar.	

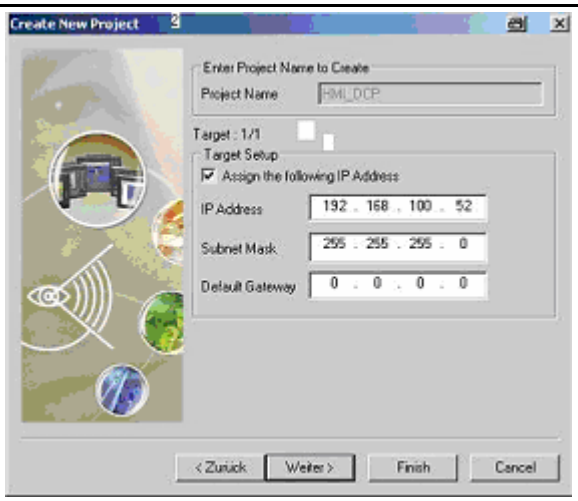
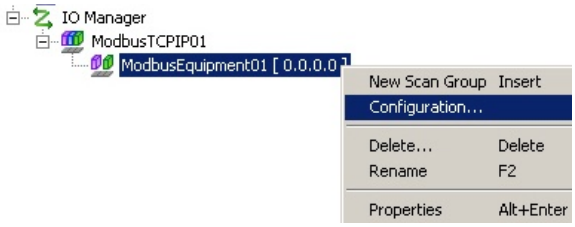
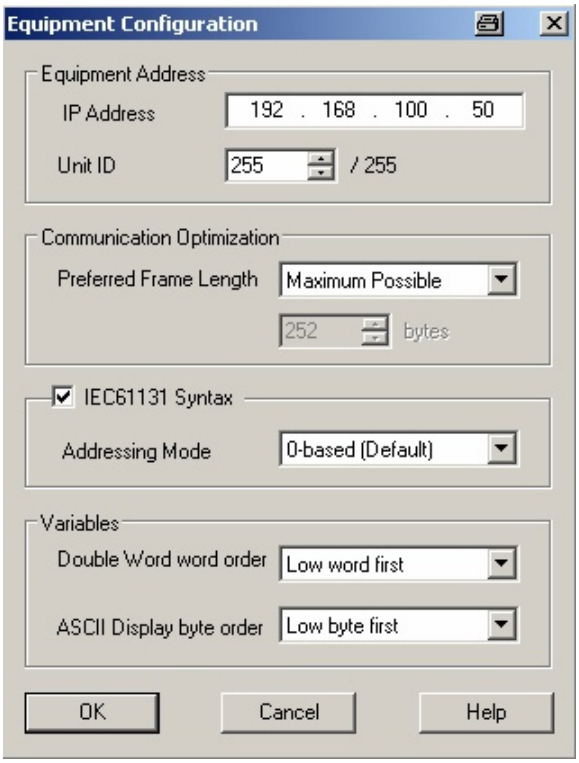
2	Enter a **Project Name** for the application and a comment (optional).	
3	Next, select the target device used and enter a logical name. Example project: Target Name: D**CP** Target Type **XBTGT 2000** Model: **XBTGT2330**	
4	In order to use the device's Ethernet interface, you need to enter the **IP Address**, **Subnet Mask** and, if applicable, the **Default Gateway**.	

5	In order to be able to exchange data with other devices, the Magelis HMI requires a communication driver. To set one up, click **Add**.	
6	Start by selecting **Schneider Electric Industries SAS** from **Manufacturer** list. For communication with the PLC, select **Modbus TCP/IP** in the **Driver** list and **Modbus Equipment** under **Equipment**. Once you have selected a communication driver, you can complete the creation of the new project by clicking **OK** followed by **Finish**.	

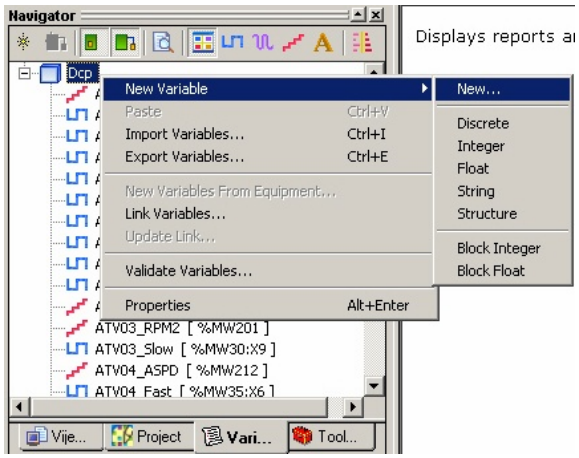
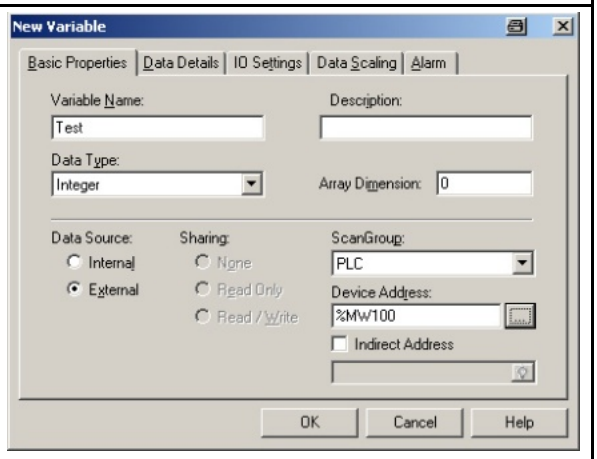
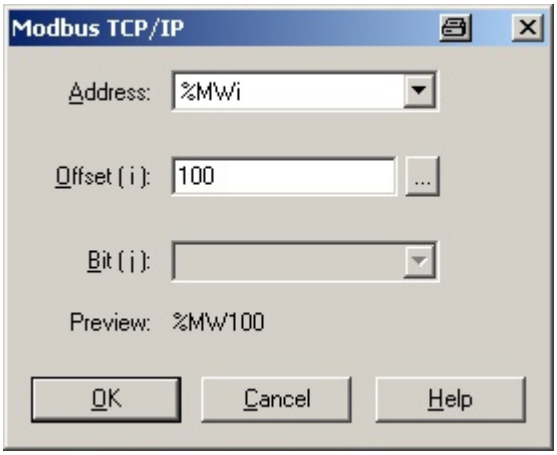
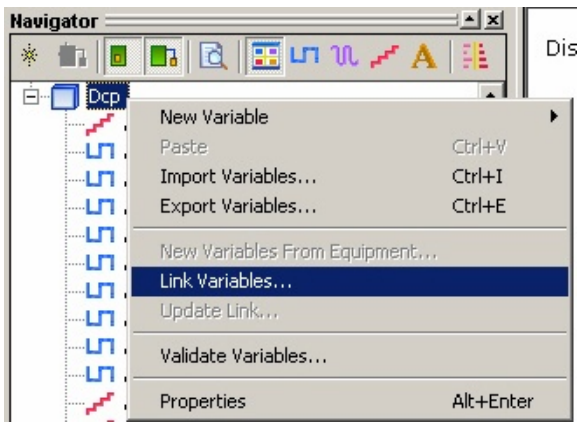
Communi-cation Settings

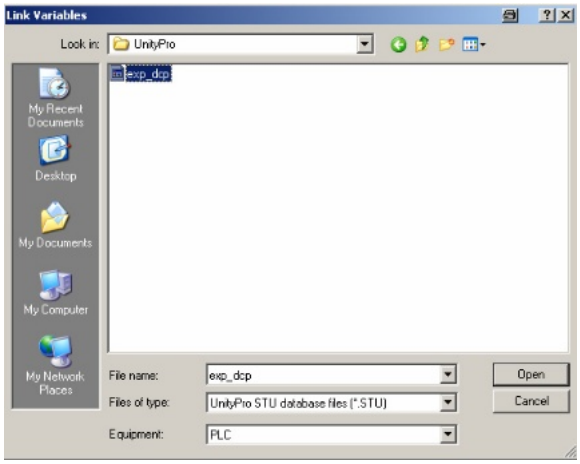
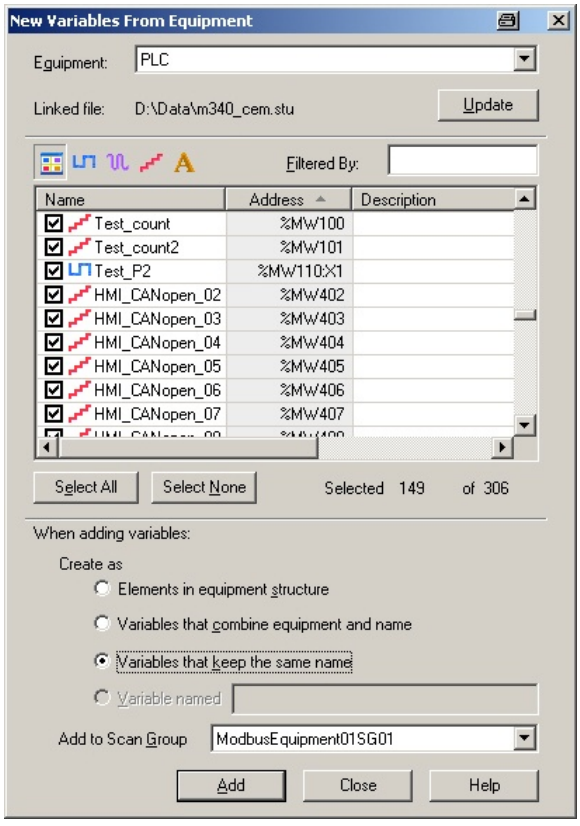
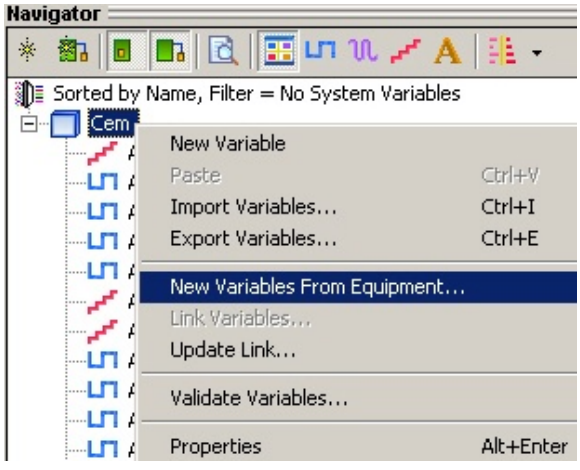
1	Once you have created the project, Vijeo Designer will display the workspace described above with an empty edit screen on the right-hand side.	

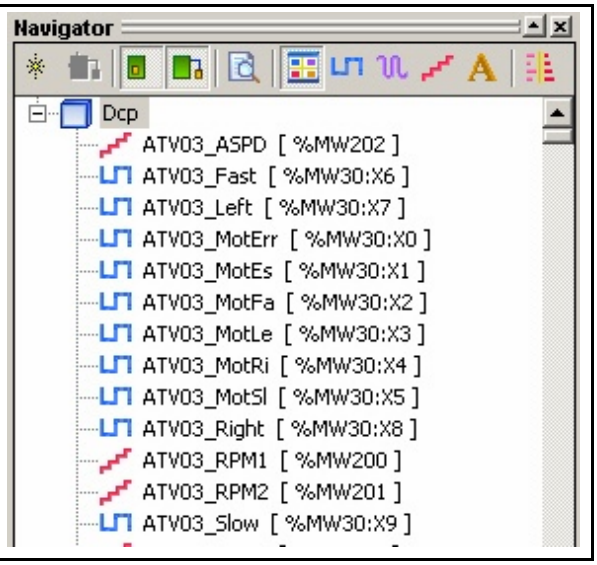
2	It is possible to change the settings for downloading the project to the HMI. To do this, click the target in the Navigator (in this case, **Dcp**) and select **Download** in the **Property Inspector**. In order that the project can be transferred to the Magelis HMI, you will need to select **Ethernet** as well as the **IPAddress** and the **SubnetMask** of the HMI.	
3	The interface parameters must be declared to the Modbus TCP/IP driver for communication with the PLC. Right-click **HMI** and select **Configuration...**.	

4	The IP address of the HMI is displayed here.	
5	For the **equipment configuration**, right-click **ModbusEquipment01** and select **Configuration...**.	
6	Enter the **IP Address** of the PLC here. Under **Communication Optimization**, select **Maximum Possible**. Following this, activate the **IEC Syntax** and set the addressing mode to **0-based (Default)**. This means that the same addressing is used as in the PLC (%MWxxx). Click **Yes** to confirm the message that appears.	
7	Right-click and select **Rename** to change the default names.	

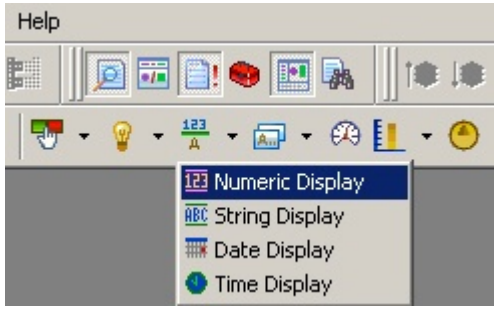
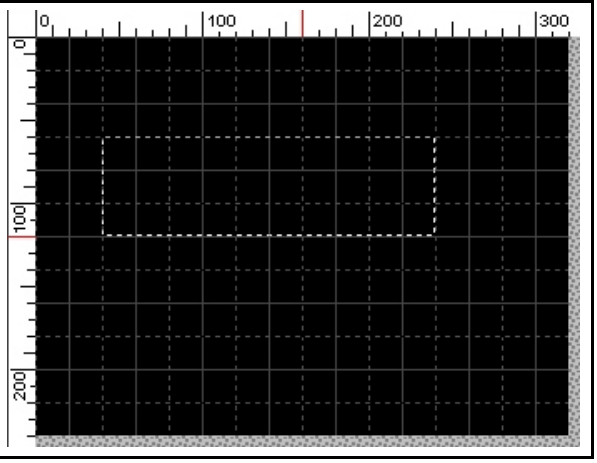
Creating Variables

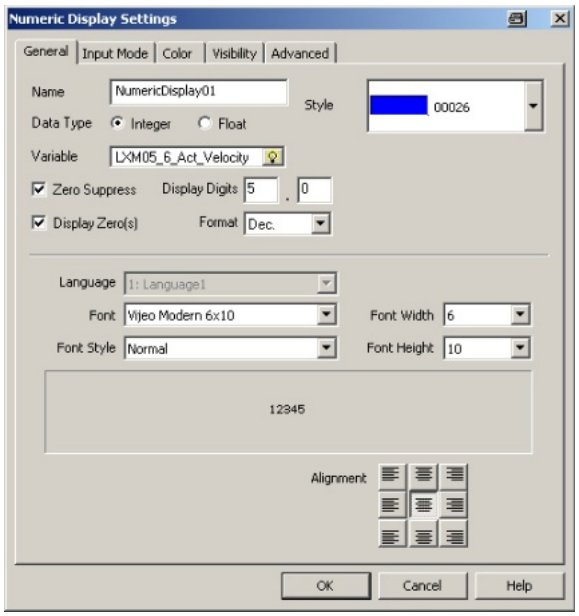
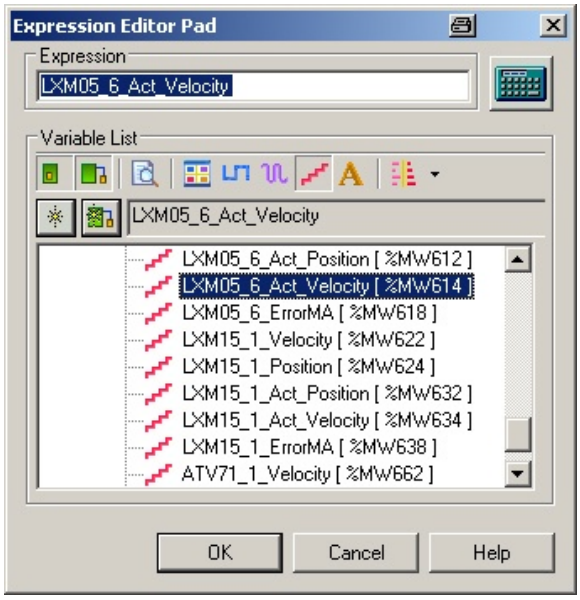
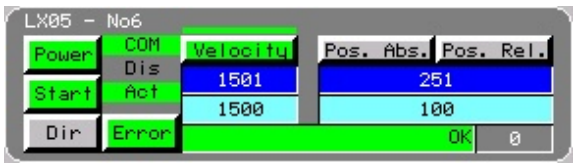
1	To create new variables in the Navigator, select the **Variables** tab at the bottom of the screen. Right-click the **project name** to access a popup menu and select **New Variable→New...**.	
2	To create variables, the following information must be entered: • **Variable Name** • **Data Type** • **Data Source (Extern)** • **Device Address in der SPS**	
3	All PLC flags (located variables) can be addressed. Types that can be defined include flags (%M), words (%MW), double words (%MD) and floating points (%MF). All data to be displayed on the Viewer must be transferred to one of these types.	
4	It is also possible to both import and export variables. Another extremely convenient way of importing the PLC variables is to establish a direct connection to the PLC project. To do this, select the **Link Variables** option from the **project name** menu on the **Variables** tab	

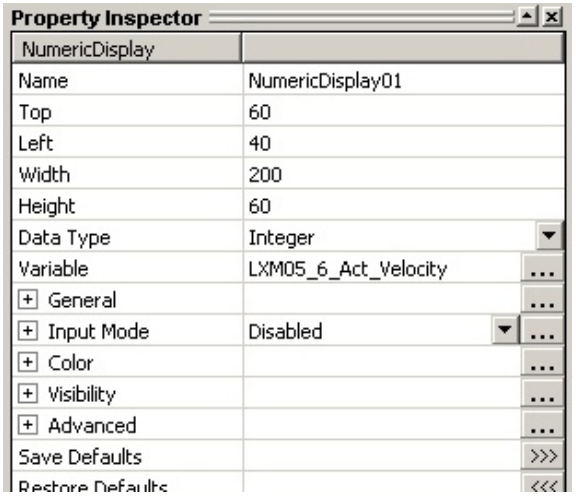
5	Next select the **File name.** Files of type: **Unity Pro (*.stu)** Equipment: **PLC** Click **Open** to continue.	
6	All variables from the PLC project are displayed in this window with a name and address. The required variables can be selected by clicking the **checkbox** on the left-hand side. To make the connection between the PLC and HMI clear, the same variable names are used here. This option is set by selecting **Variables that keep the same name**. The selected variables are then transferred by clicking **Add**. Select **Close** to close the window.	
7	If other variables are required at a later point, it is possible to recall the window described above by selecting **New Variables From Equipment**. **Update Link...** can be selected to update the link in relation to the PLC file.	

8	The variables created are displayed in the **Navigator**, along with their names and addresses.	

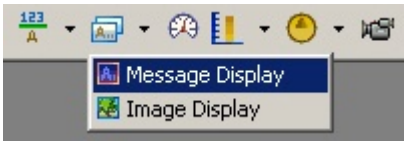
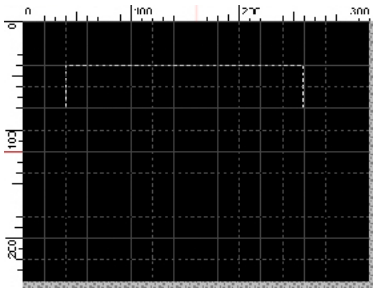
Creating Screens

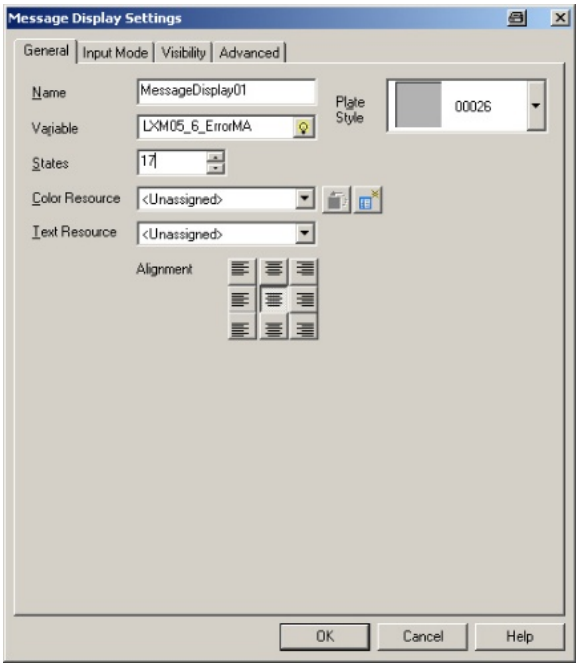
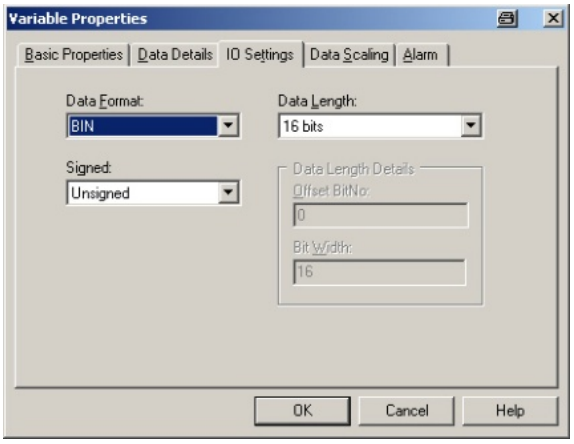
The process for creating animations on screens will now be described using a numerical display. The functions are similar for other animation elements.		
1	Selection from the menu bar. Various icons and elements are available in the menu bar and the toolbox. Select **Numeric Display**	
2	First, define the **position** and **size** of the display area.	

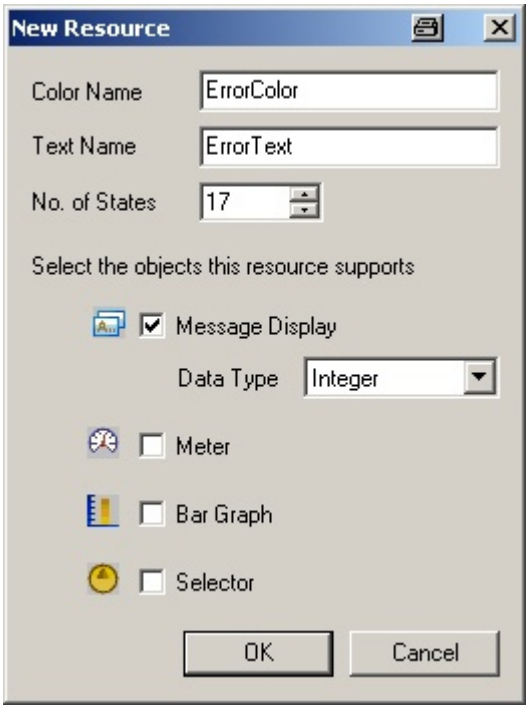
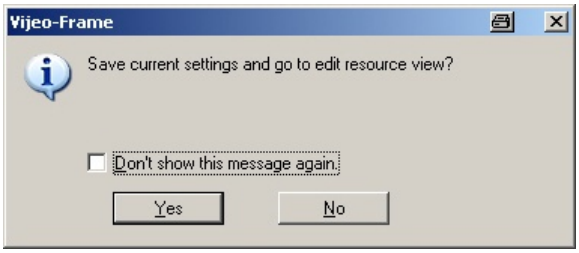
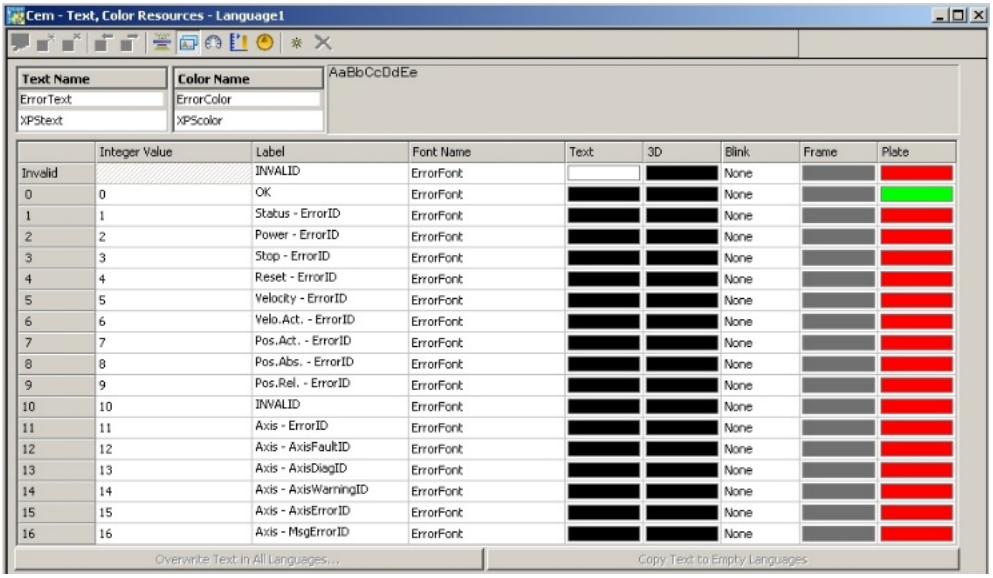
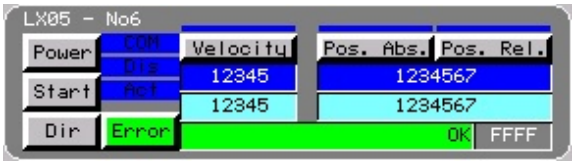
3	**Numeric Display Setting:** • **Name** • **Data Tpye** • **Variable** • **Display Form** • **Font** The variable can be entered directly or can be selected by means of the icon to the right of the field (**light bulb**). **Note:** A variable name that has been entered but not recognized appears in red.	
4	The variable to be animated can be transferred from the list by double-clicking it. Additional functions, e.g., value inversion, can be executed by clicking on the **calculator icon**.	
5	The screenshot opposite (showing part of a complete screen) displays various animation elements.	

6	**Property Inspector** Each animation element on the screen has its own properties which can be viewed in the Property Inspector (right-click on the object). The property Inspector lists all settings associated with the element and they can be modified.	

Displaying an Error Message

1	In the PLC, servo drive error messages can be displayed as a number from 0 to 16. However, this needs to be displayed as text on the HMI. For this purpose, it is possible to select and position the **Message Display**.	 

2	To begin making the settings, select the **Variable**. Also enter: States: **17** Then click the **New Resource** icon (to the right of the **Color Resource** field). Note: On the **I/O Settings** tab in **Variable Properties**, **BIN** must be selected as **Data Format** **16 bits** as Data Length.	 

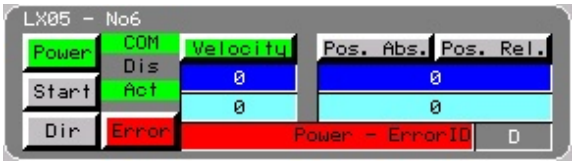
3	In the New Resource window, enter the following: Color Name: **ErrorColor** Text Name: **ErrorText** No. of States **17** Select **Message Display**. Data Type: **Integer** Finally, click **OK** and **Yes**.	 
4	In the following table, a **Label** and **Font Name** can be entered for each **Integer Value** (0 – 16). 	
5	The message display appears on the screen, e.g. on the screen for displaying a Lexium 05 error.	

Cem - Text, Color Resources - Language1

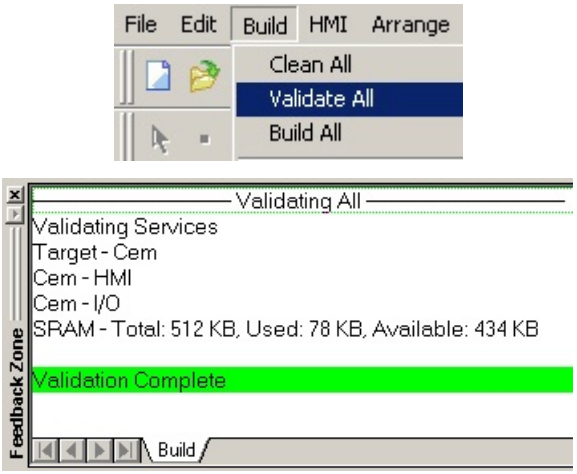
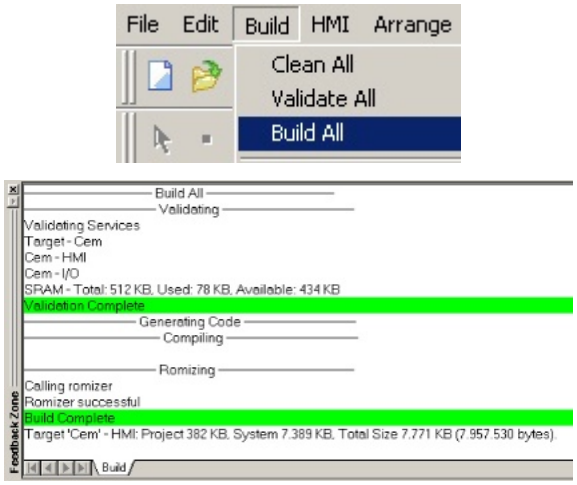
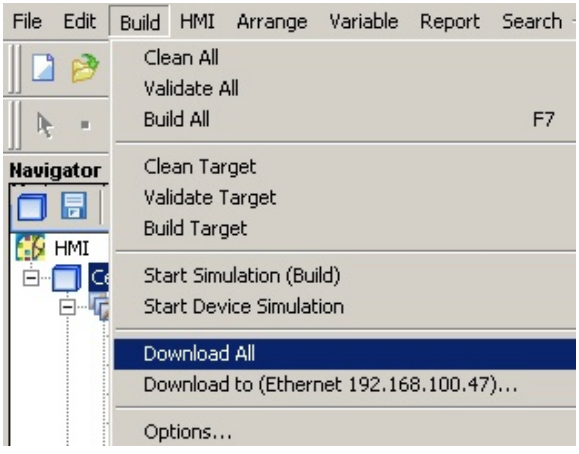
Text Name	Color Name	AaBbCcDdEe
ErrorText	ErrorColor	
XPStext	XPScolor	

	Integer Value	Label	Font Name	Text	3D	Blink	Frame	Plate
Invalid		INVALID	ErrorFont			None		
0	0	OK	ErrorFont			None		
1	1	Status - ErrorID	ErrorFont			None		
2	2	Power - ErrorID	ErrorFont			None		
3	3	Stop - ErrorID	ErrorFont			None		
4	4	Reset - ErrorID	ErrorFont			None		
5	5	Velocity - ErrorID	ErrorFont			None		
6	6	Velo.Act. - ErrorID	ErrorFont			None		
7	7	Pos.Act. - ErrorID	ErrorFont			None		
8	8	Pos.Abs. - ErrorID	ErrorFont			None		
9	9	Pos.Rel. - ErrorID	ErrorFont			None		
10	10	INVALID	ErrorFont			None		
11	11	Axis - ErrorID	ErrorFont			None		
12	12	Axis - AxisFaultID	ErrorFont			None		
13	13	Axis - AxisDiagID	ErrorFont			None		
14	14	Axis - AxisWarningID	ErrorFont			None		
15	15	Axis - AxisErrorID	ErrorFont			None		
16	16	Axis - MsgErrorID	ErrorFont			None		

Overwrite Text in All Languages... Copy Text to Empty Languages

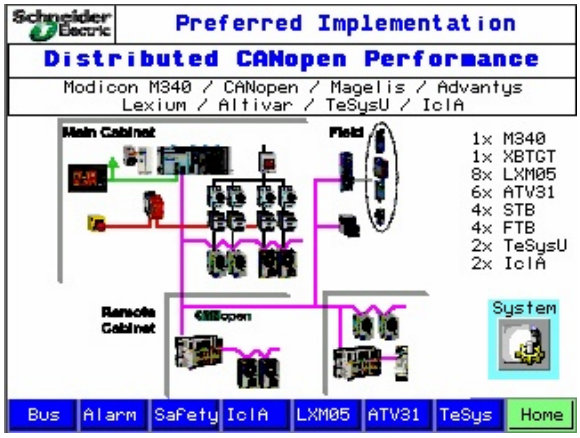
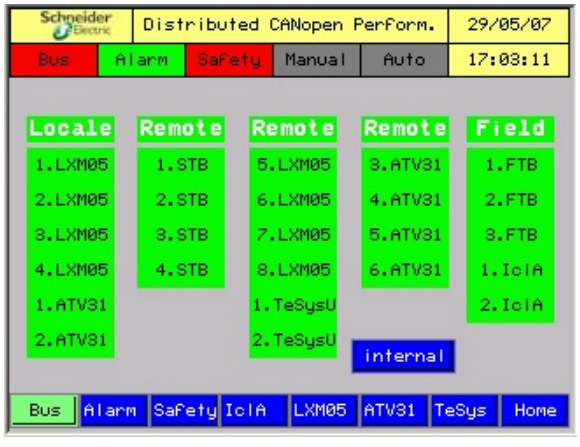
6	The appropriate output text will then appear during operation according to the error number.	

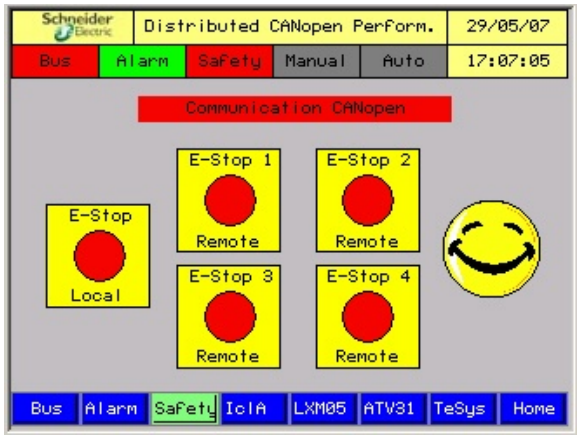
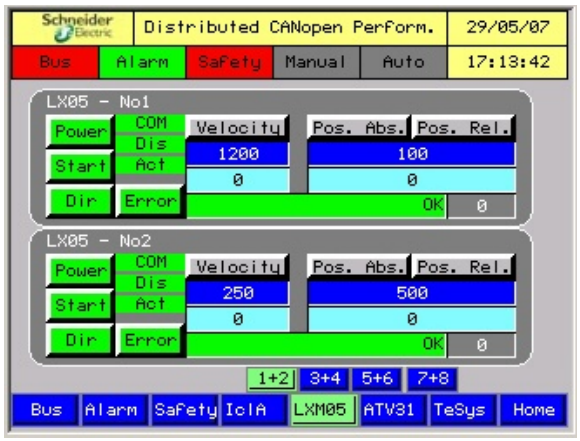
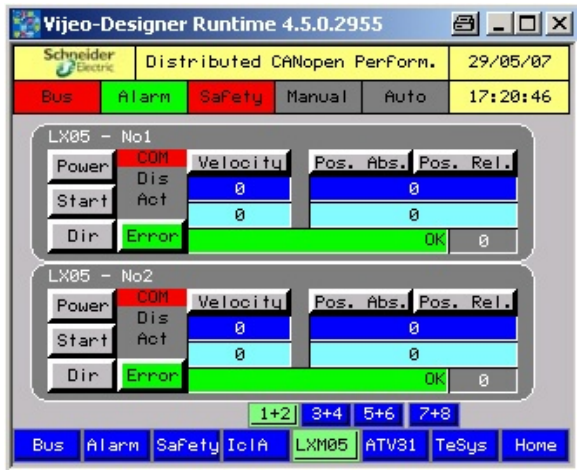
Downloading the Project

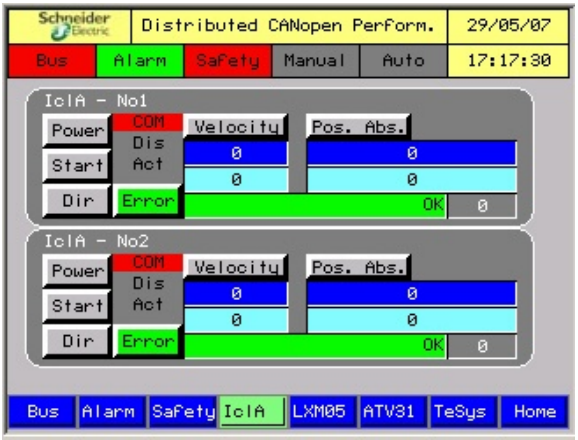
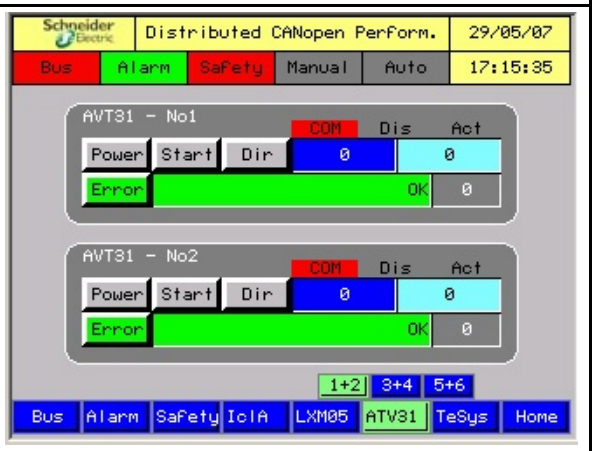
1	Before being downloaded to the HMI, the project must first be analyzed. To do this, select **Validate All** from the **Build** menu. The results are listed in the **Feedback Zone**.	
2	If **Build All** is selected instead, the messages are still listed in the **Feedback Zone**.	
3	Select **Download All** under **Build** to transfer the application to the connected Magelis terminal. The configured method of communication (in this case, Ethernet) is used.	

4	**Assigning the Ethernet IP Address** Unless the project has already been transferred using a USB cable, the HMI will not have the correct IP address. For this reason, the IP address must be entered via the offline setting mode before downloading takes place. This is called up as follows: • **On powering up, touch the top left-hand corner of the screen.** • Alternatively, while the application is being executed, touch three corners of the screen at the same time. (In the platform properties of the Vijeo Designer Editor, you can select the procedure to be followed by your application.) • Next, enter the IP address. • Switch back to online mode.

Übersicht Applikation

1	The example application features a number of displays that can be selected by the user. The structure is shown on the welcome screen. **Manual** operation mode is set by default. There are no logic configuration settings in the PLC for automatic mode. All drives can run in manual mode, controlled directly via the display. To do this, you must switch to the relevant screen. The HMI configuration screen can be reached via **System** (in the lower right-hand corner of the screen).	
2	The header on subsequent screens is identical and provides information about the status of the machine. If a CANopen bus node is faulty, this will be indicated in the header under **Bus**. Switch to the Bus screen to identify the node. More information can be accessed by pressing **Detail**.	

3	A summary of the individual alarms is provided on the **Alarm** screens. In the header, the Alarm field is a group message.	
4	Safety controller messages are displayed on the **Safety** screen. The two Emergency Stop buttons are displayed along with their inputs and the two outputs. The details provide information about the status, mode, outputs, inputs and diagnosis.	
5	The screenshot opposite shows two Lexium 05 drives. Each operation mode (velocity, absolute positioning and relative positioning) has a selection button. The drive is activated by means of **Power**. The operation mode is triggered using **Start**. The direction or rotation is set using **Dir** (in Velocity mode only). An error message is acknowledged by means of **Error**. Both the setpoint speed and setpoint position can be set using a virtual keypad. The status message (COM = Communication, Dis = Disabled and Act = Active), the display of the actual speed and actual position, and the error message all act as feedback	 

6	Control of the two IcLAs is the same as for the Lexium 05 but has no positioning function. The other control elements remain the same.	
7	The control functions for the six Altivar 31 variable speed drives is the same as for the IcLAs i.e. no positioning. The other control elements remain the same.	
8	The two TeSysU motor starters can be switched on and off using Start. The status is displayed by means of the status elements.	

Devices

Introduction

This chapter describes the steps required to initialize and configure the devices to attain the described system function.

General

The following devices are used:

- **Advantys STB**
- **Advantys FTB**
- **Lexium 05 und Altivar 31**

  The Lexium 05 servo drives and the Altivar 31 variable speed drive can be set up via the front operator panel. You also have the option of using the **PowerSuite** software. The advantages of using PowerSuite are that you

  - Can save the data on your PC and copy it as you wish
  - Can print out the documentation *and*
  - Can be assisted in optimizing the parameters online.

- **IclA**

  The software **IclA Easy** is used to commision the IclA IFS compact drivesa. The software enbles you to:

  - Save data on your PC to be duplicated at a future date
  - Print documentation
  - Optimise parameters online.

- **TeSysU**

  The TeSysU motor starter consists of a power base, control unit and communication module. No software is required for parameterization.

Advantys STB

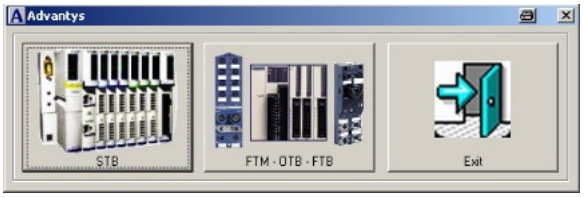
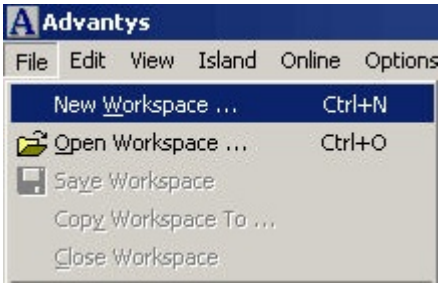
Einleitung

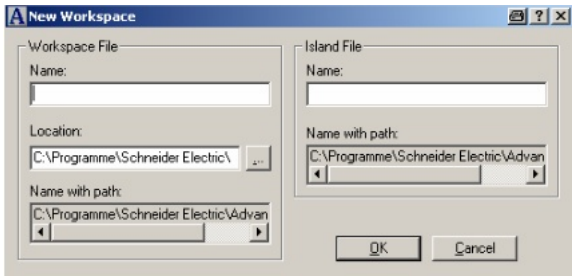
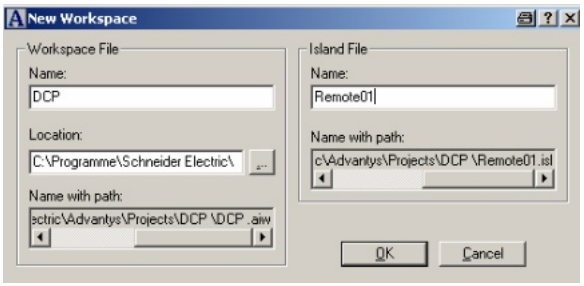
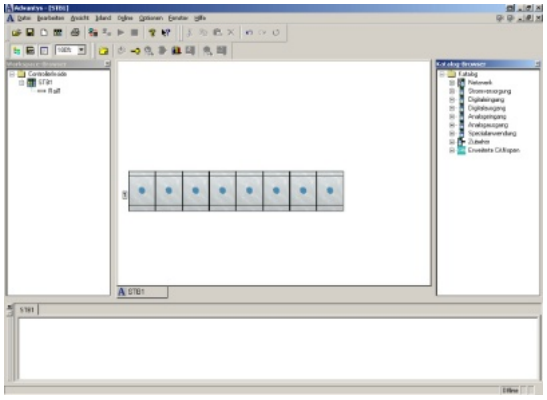
This chapter describes how to configure the Advantys I/O island. For this we use the Advantys configuration software.

The following steps are required:

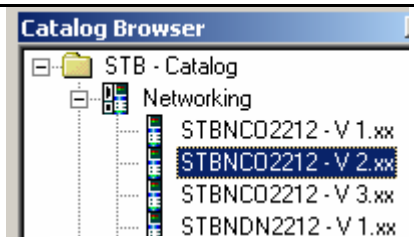
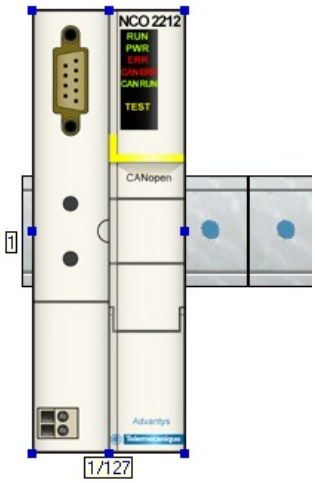
- Create a new Project (Workspace)
- Configure the Hardware (Network interface, Power Supply and I/O Modules)
- Configure the CANopen Expansion bus (Baudrate)
- Download the configuration to the Island
- Create a DCF file

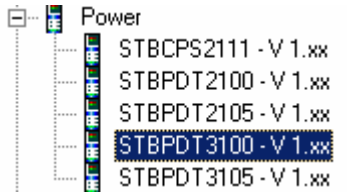
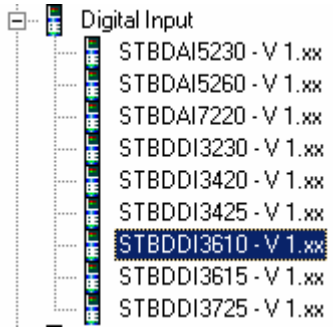
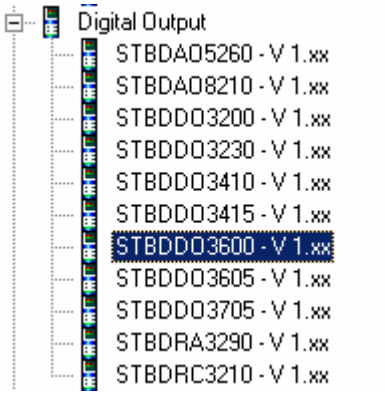
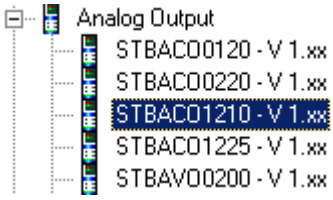
Create a new Project

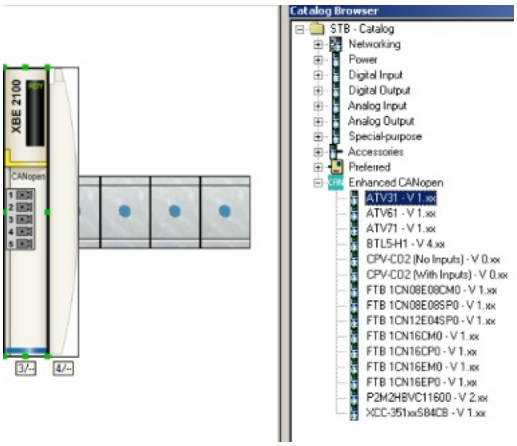
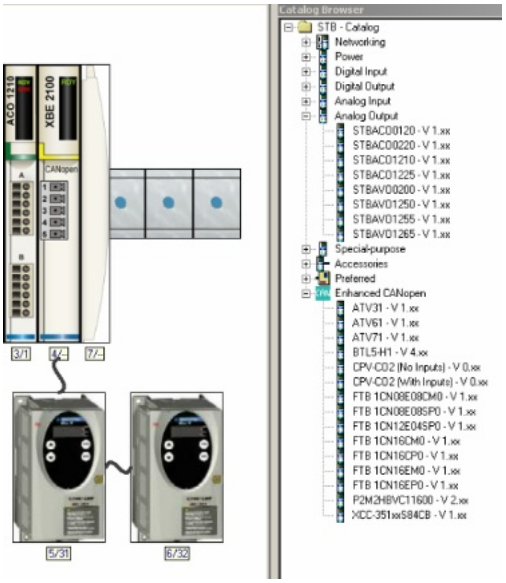
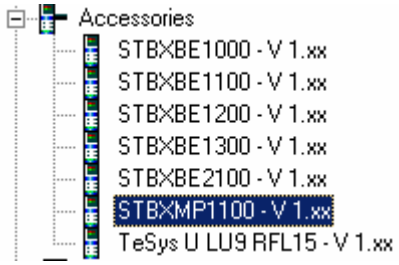
1	After installing and starting the Advantys Configuration Software you are offered a choice of A**dvantys STB,** Advantys FTB, FTM und OTB. Select **STB**.	
2	Next, select the language.	
3	After starting the Advantys software, you must create a new workspace.	

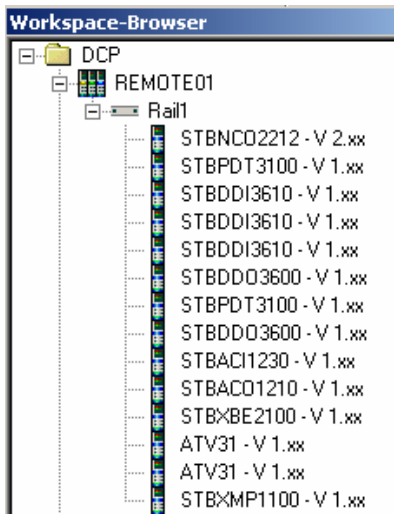
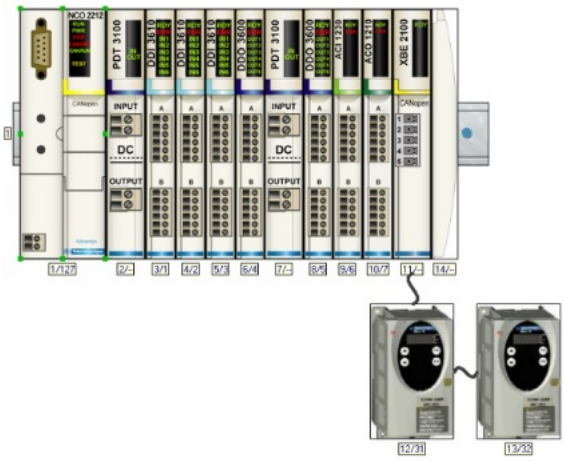
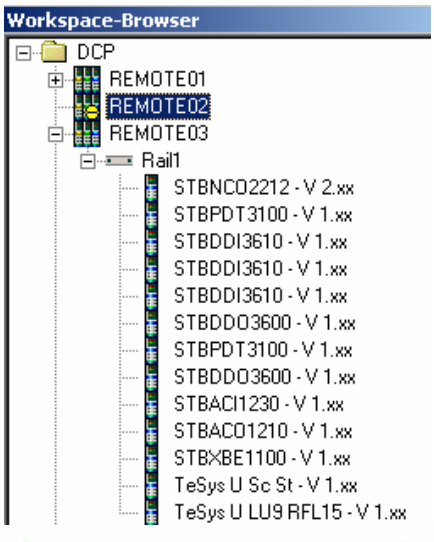
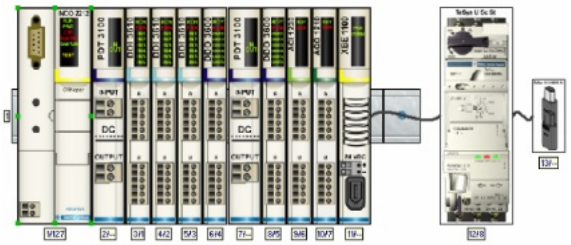
4	To do this, you must specify the path, the workspace name(DCP) and the name of the first island (Remote01..04).	 
5	An empty railing is displayed.	

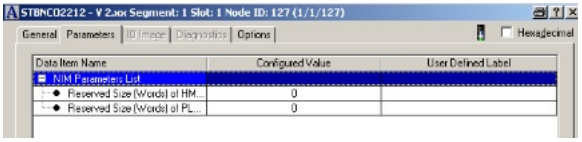
Configure the Hardware

1	Select the CANopen network interface: **STB NCO 2212 V2.xx**	 

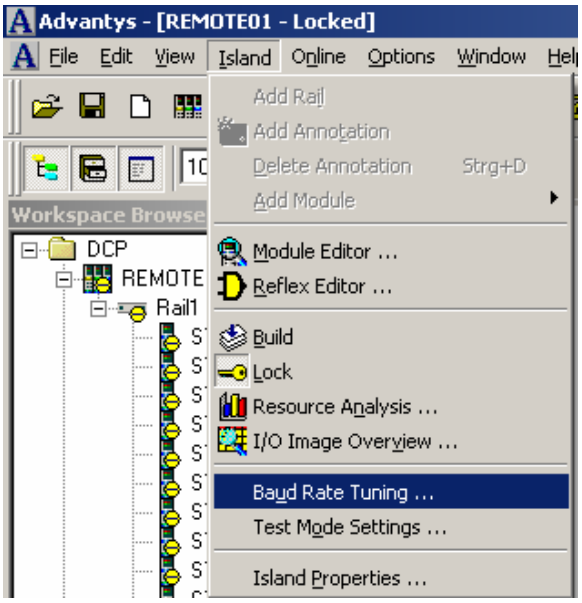
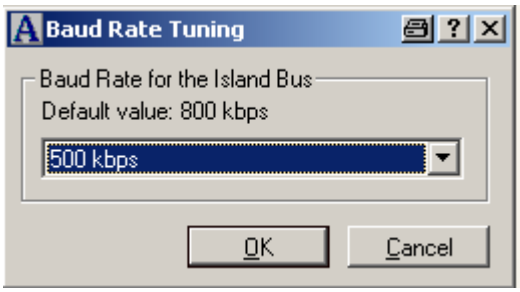
2	The power supply: **STB PDT 3100** (2x)	
3	the digital input modules: **STB DDI 3610** (3x),	
4	the digital output modules: **STB DDO 3600** (2x)	
5	the analog input modules: **STB ACI 1230** (1x)	
6	the analog output modules: **STB ACO 1210** (1x)	

7	The devices on the CANopen extenssion module: **STBXBE2100**	
8	For example, the 2 Altivar 31‘s with the node addresses 31 und 32	
9	As terminal resistor: **STB XMP 1100**.	

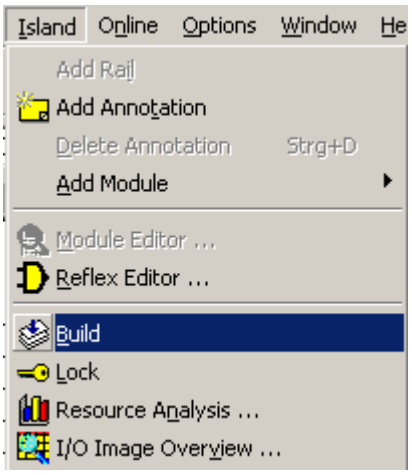
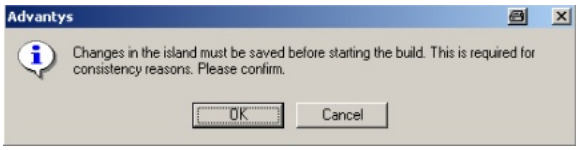
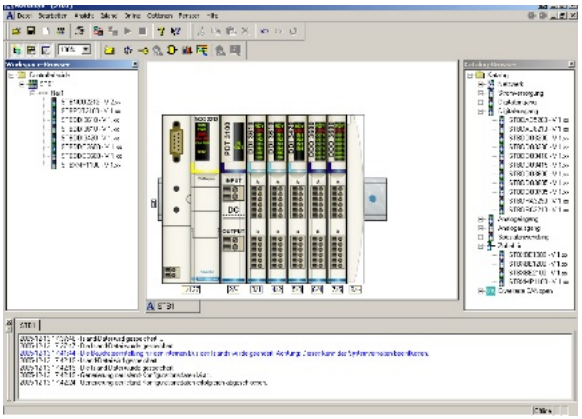
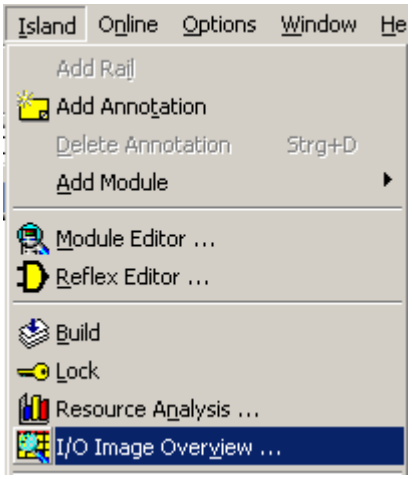
10	You should now see the setup on the right. For the STB-Island: Remote01 and Remote02 For the STB-Island: Remote03 and Remote04	   

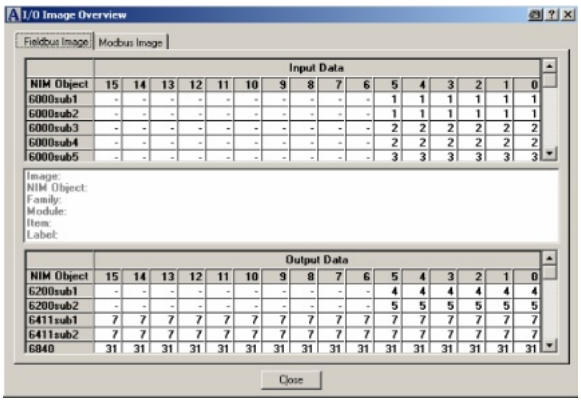
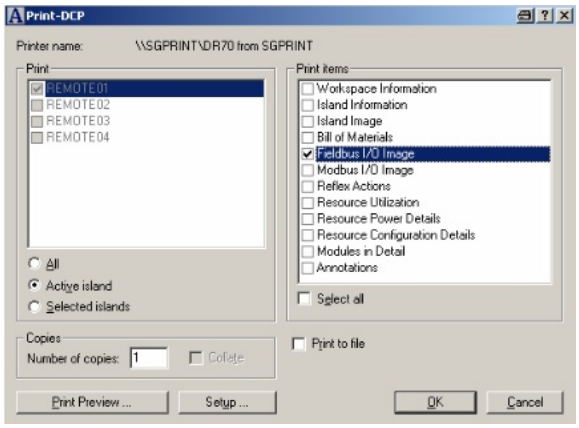
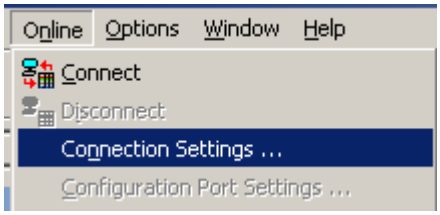
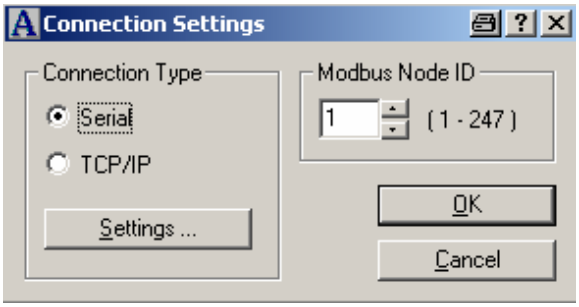
11	a double clcick on the CANopen interface (**STBNCO2212)** brings up the attributes. On the **Parameters** tab you can define the length of the exchange buffer between the PLC and the HMI. This is not used in our example and is set to **0**.	

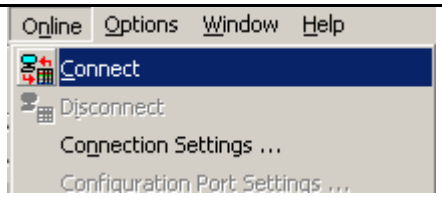
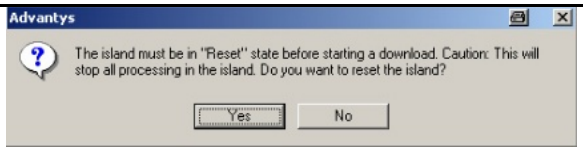
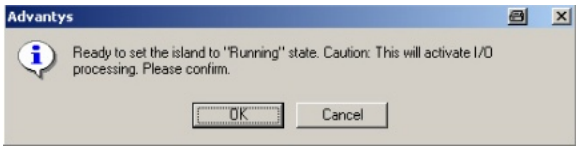
Configure CANopen-Expansion bus

1	The CANopen baud rate can be set via: **Island->Baud Rate Tuning…**	
2	Set the baud rate to **500 kbps**. **Hinweis:** Die Übertragungsrate zwischen NIM und SPS wird an den zwei Drehschalter auf der Frontseite vom NIM parametriert. Siehe hierzu das Kapitel Kommunikation.	 

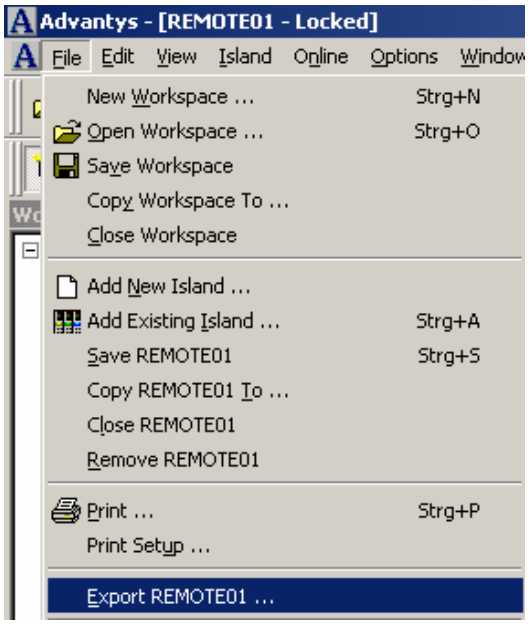
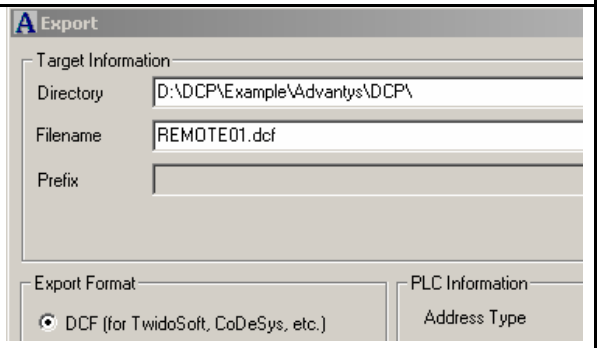
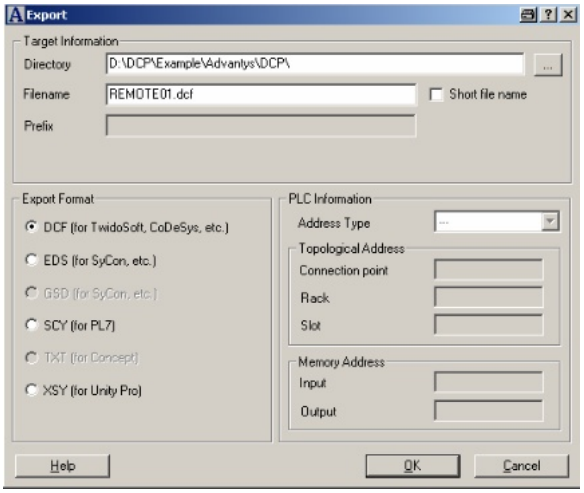
Download the Configuration to the Island

1	The configuration is now finished. Now you must build the project. select: **Island->Build**.	
2	You will be asked to save your configuration first. Continue with **OK**.	
3	The results of the build are shown in the lower frame.	
4	If all goes well you should now see **... Build completed successfully**	2006-10-10 11:24:42 - Island file has been saved. 2006-10-10 11:24:42 - Build of island data in progress ... 2006-10-10 11:24:42 - Note: Please check for bus termination on the last module in the island. 2006-10-10 11:24:47 - Build completed successfully
5	You can use the **I/O Image Overview...** menu item (in the **Island** menu) or the equivalent icon to call the function for assigning the I/O to the memory areas.	

6	The information concerning the selected data is displayed in the description field. Alternatively, the project can also be printed out. The printout will contain the same information. **Note:** To obtain a print out the data select: **Fieldbus-I/O-Image**.	 
7	To load the configuration you must first set up the connection to the device. select: **Online->Connection Settings...** The serial cable requires the following settings: **Serial** **Modbus Node ID: 1**	 

8	Use **Online->connect** to make the connection. The configuration version in the PC and NIM (network interface module) will be compared to see if they match. If they do not match you will receive a message to that effect. Continue with **Download**.	 
9	Confirm the following dialogs with **Yes** and **OK**.	  
10	The lower frame will eventually show:	**Island is healthy.**
11	You can now **Disconnect**.	

Create a DCF file

1	To create a DCF file select **File->Export Remote01...**	
2	Enter a **Directory** and a **Filename**. **Note: The** DCF file is required for the CANopen set up in UnityPro.	 

Advantys FTB

Introduction

This section describes the steps required to prepare the Advantys FTB I/O platform for the CANopen communication.

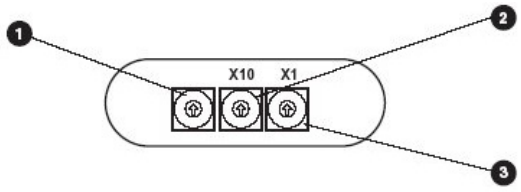
Inclusion of the FTB requires the designation of a CANopen bus address and the transmission baudrate.

All other configuration requirements are carried out using UnityPro

Advantys FTB CANopen Configuration

The CANopen address and transmission rate are set manually on the modules via rotary switches. There are two rotary switches for the address and one for the transmission rate.

In the example software, the 3 FTB modules are configured with the addresses 11,12, and 13 and the transmission rate is set to 500.0 kbaud (position 7 on the rotary switch).



Element	Function
1	Transmission speed (kBit/s)
2	Node-ID x 10 switch
3	Node-ID x 1 switch

Position of the encoder wheel	Transmission speed
O	Automatic recognition
1	10 kBits/s
2	20 kBits/s
3	50 kBits/s
4	100 kBits/s
5	125 kBits/s
6	250 kBits/s
7	500 kBits/s
8	800 kBits/s
9	1 Mbits/s

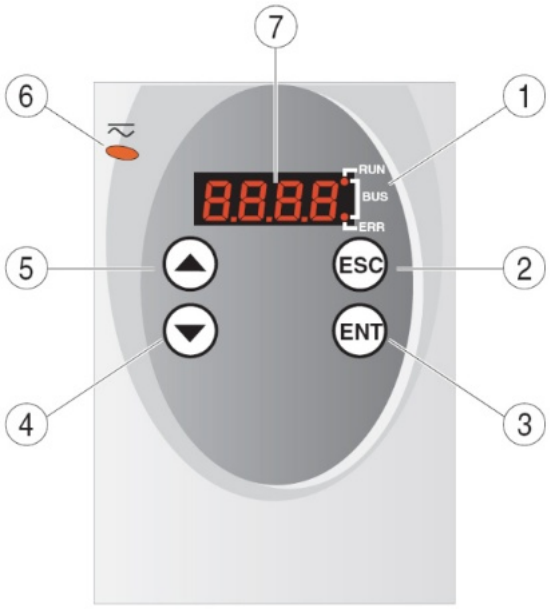
Lexium 05

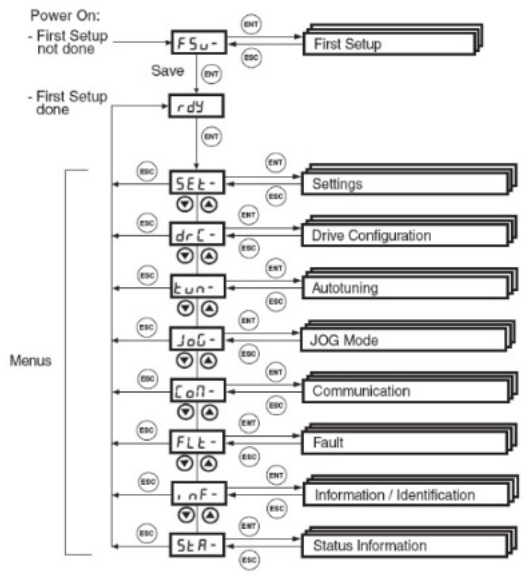
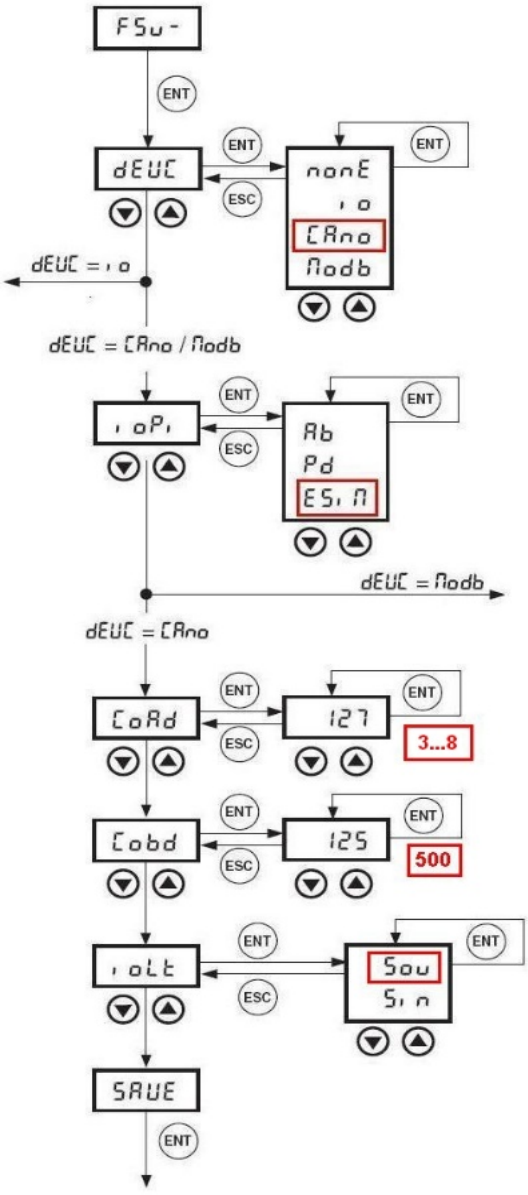
Introduction This chapter describes how to parameterize Lexium 05 servo drives.

Preconditions Before carrying out the steps described below, you must ensure that:

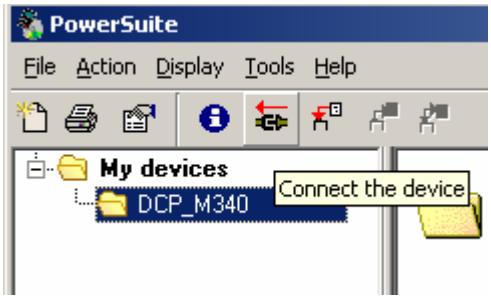
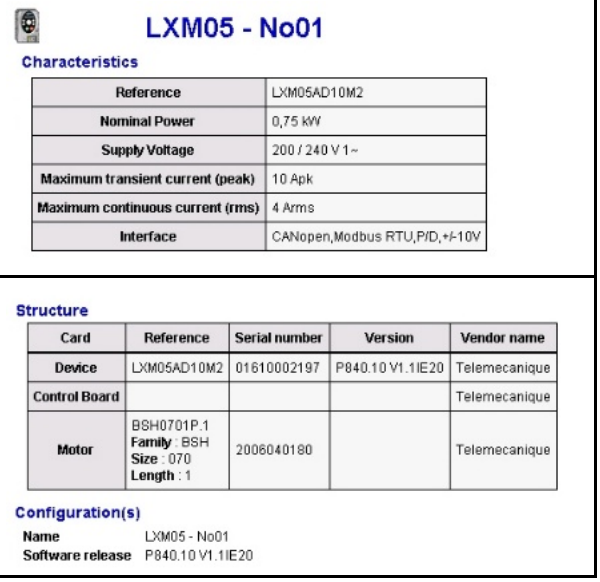
- The PowerSuite parameterization software is installed on your PC.
- The servo drive is connected to the power supply.
- The PC is connected to the servo drive via the communication cable.

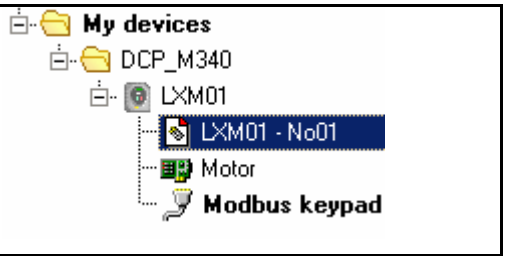
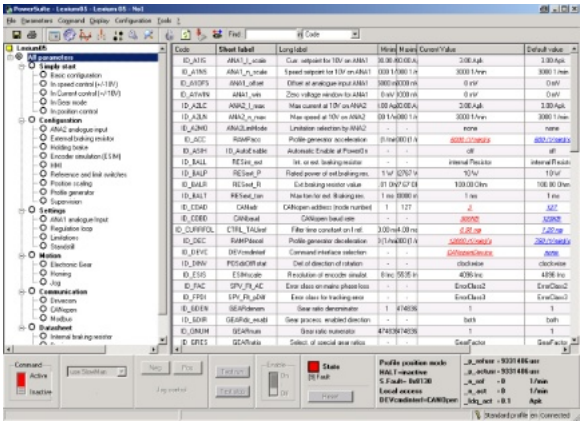
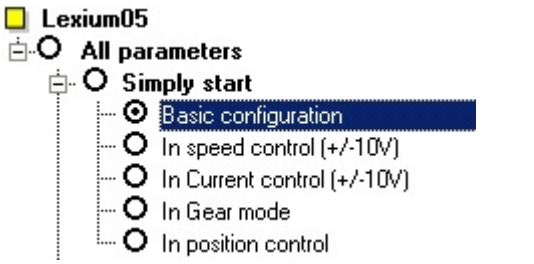
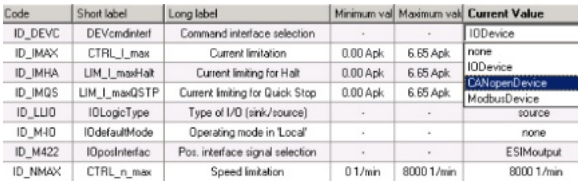
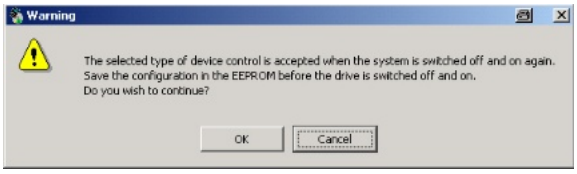
LXM05 Manual Setup

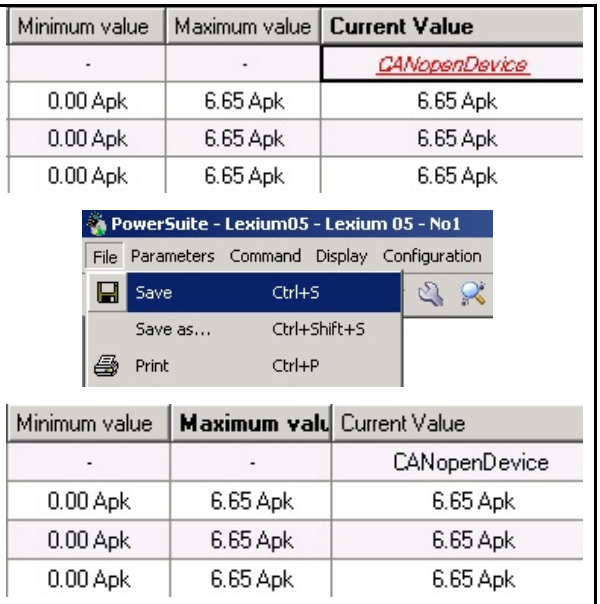
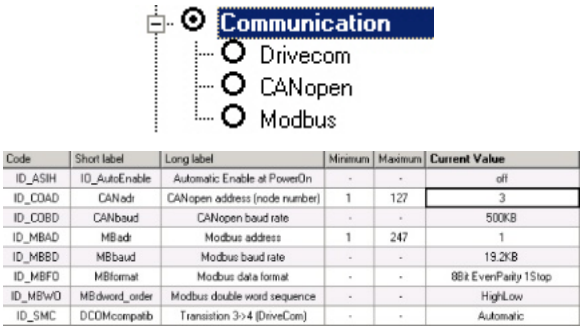
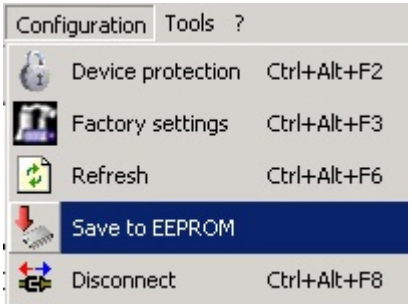
1	After wiring is complete, the drive control parameters must be set. Parameters can be edited via the integral operating panel (HMI).	 (1) LEDs for fieldbus (2) ESC: - exit a menu or parameter - return from the displayed to the last saved value (3) ENT: - call a menu or parameter - save the displayed value to EEPROM (4) Down arrow: - switch to next menu or parameter - reduce the displayed value (5) Up arrow: - switch to previous menu or parameter - increase the displayed value (6) Red LED on: DC bus under power (7) Status display

2	The HMI operates on the basis of menus. The screenshot to the right shows the top level of the menu structure. In order to gain access via the PowerSuite software, you will first need to check the Modbus parameters. Under **CoM**, set **MbAd = 1** and **Mbbd = 19.2**	
3	When the drive is supplied with 24V for the first time, or if the factory settings have previously been loaded with the `PARfactorySet` parameter, all the drive functions are still blocked. **You must carry out an initial setup procedure.** To establish the link to the CANopen master, you will need to make settings in respect of the following: - **Method of control** - **Signal selection position interface** - **CANopen parameter and** - **Logic type** On completion, the drive reports "**RDY**" (ready) in the status display.	

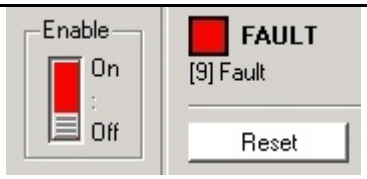
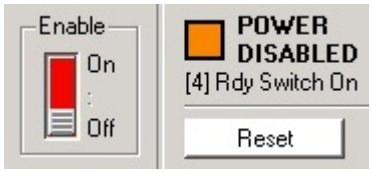
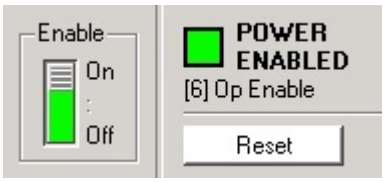
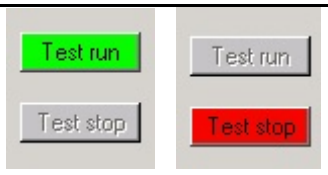
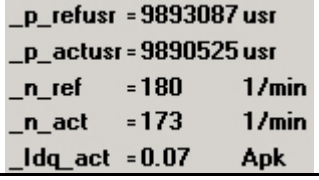
PowerSuite with LXM05

1	As well as making settings manually, it is also possible to use the PowerSuite configuration software.	
2	Following startup, a connection to the device is established via **Action->Connect** or by clicking on the corresponding icon.	
3	You will receive a warning that there is no record of the new device. Select **Create**.	
4	Next, you need to enter the name of the configuration or device.	
5	The data is read from the Lexium 05.	
6	Once the transfer is complete, the device data will be displayed.	

7	You can select the relevant drive by double-clicking it in the project browser on the left-hand side.	
8	The parameters can be displayed in list format or in page view. You can switch the view via **Display->List** or **Pages** from the menu bar.	
9	Select **Simply start->Basic configuration**. And in the **Command interface selection** field, you should select: **CANopenDevice**. The servo drive will now be enabled for control via CANopen. In order for this change to take effect on the Lexium 05, you will need to switch the device off and then back on again. Click **OK** to close the message window.	  

10	Initially, the change will be highlighted in red, but the display colour will change when you select: **File->Save.**	
11	Make the following settings under **Communication**: **CANopen address: 4..7,** **31 and 32 ,** **41 and 42** **CANopen baud rate: 500 kbs** **Modbus address: 1**	
12	To transfer the settings to the Lexium 05, select **Configuration->** **Save to EEPROM**	
13	Click **OK** to confirm the message windows that appear. The transfer is complete.	

Online Lexium 05 Control

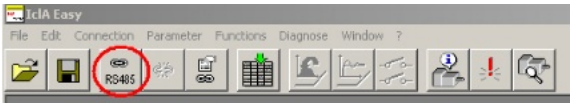
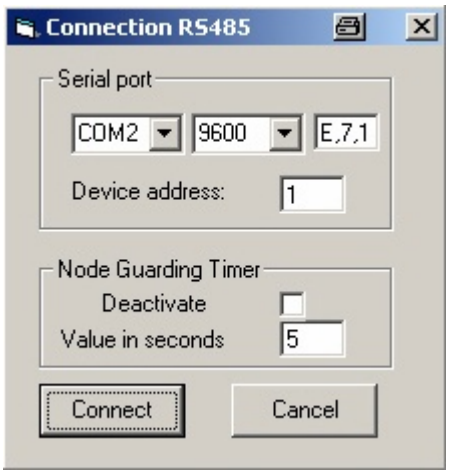
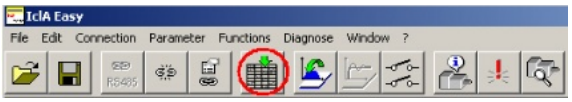
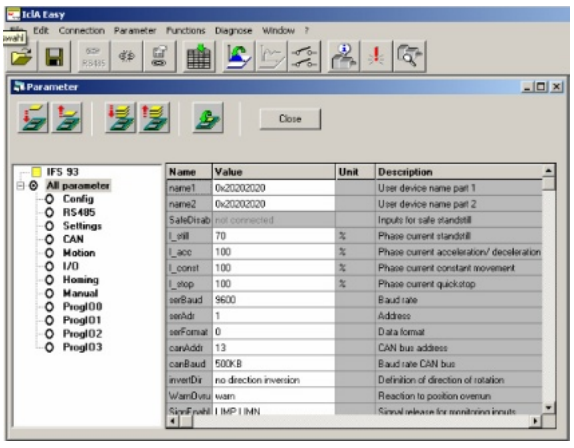
1	You have the option of controlling the servo drive via the PowerSuite software. To do this, you must first set the **Command** switch to **Active.** Press **Alt+F** to confirm the security warning once you have read it.	  
2	Then set the **Enable** switch to **On**. Faults can be acknowledged by clicking **Reset**.	  
3	**Test run** can be used to activate the servo drive. **Test stop** can be used to stop it again.	
4	**Neg.** and **Pos.** can be used to rotate the drive.	
5	Information about the speed and position is displayed on the bottom right.	

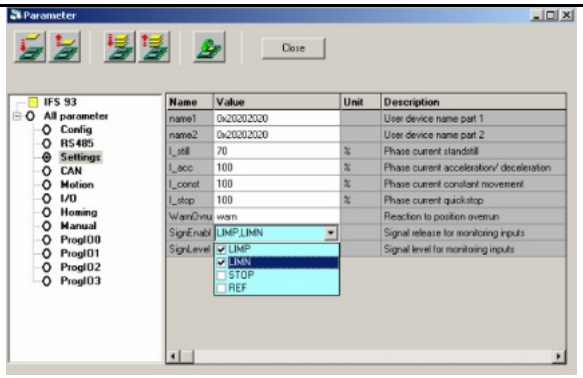
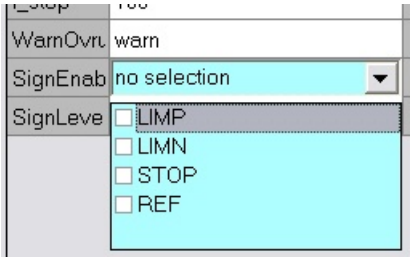
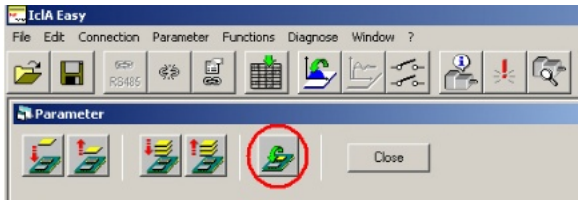
IcIA IFS93

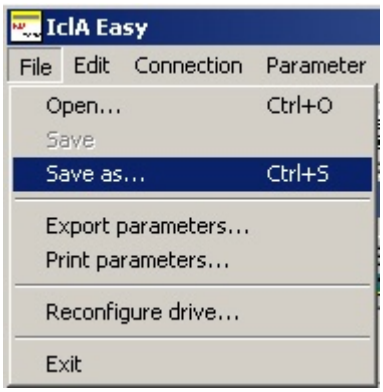
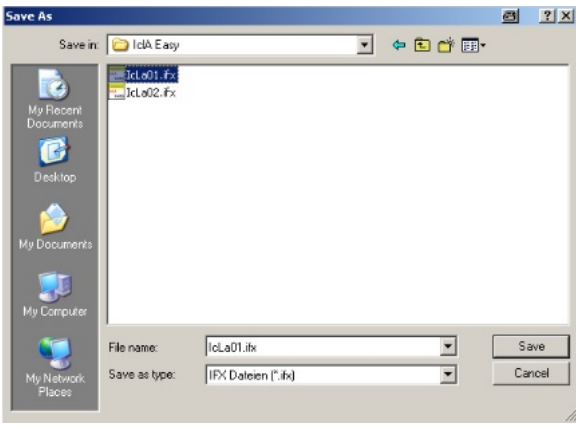
General

The IcIA IFS parameters can be modified using the IcIA EASY software.

The IcIA EASY software is available free on the Berger Lahr website. A programming cable for the RS485 interface is also required.

1	After starting up IcIA EASY, click the **Connect** icon...	
2	...to open the **Connection RS485** dialog box. In this dialog box, only the COM port used by the interface needs to be set. The other settings should be left unchanged. The COM port can be read from the computer's Hardware Manager.	
3	Once the connection has been established, click on the icon for **Open parameter window...** to display the parameter list.	
4	Here, all the IcIA IFS parameters are listed in their relevant groups. Under the **Settings** entry...	

5	...the limit switch should be disabled via **SignEnab**. To do this, remove the checks in the boxes next to **LIMP**, **LIMN** and **STOP**.	 
6	Transfer the parameters to the IcLa.	
7	The parameters must now be loaded into the drive's EEPROM. To do this, click the **Save device parameters in EEPROM** icon.	

8	Now save the parameters on your PC with **SAVE as....** The file extension for IclA Easy is *.ifx	 

Altivar 31

Introduction

This section describes how to configure the parameters for the Altivar VSD. You can input the Altivar parameters using the programming panel on the device itself (optional). However, the big advantage of using PowerSuite is that you can store the data on your PC and also have the possibility to print and document this information. The software is also an aid to invoking the Altivar and optimising the parameters in online mode.

NOTE: The factory settings in your version of PowerSuite may be different to those delivered on the actual device. If you wish to ensure that you start with the same base configuration on device and in PowerSuite, it is recommended that you upload the configuration from the Altivar before making any changes in the factory settings. If you change the settings on the Altivar, the factory settings can be restored by setting the FCS function in the control menu DRC using the front panel of the Altivar.

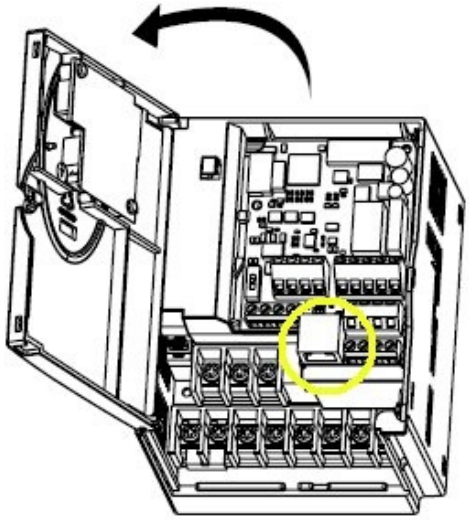
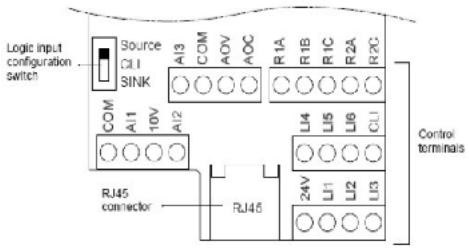
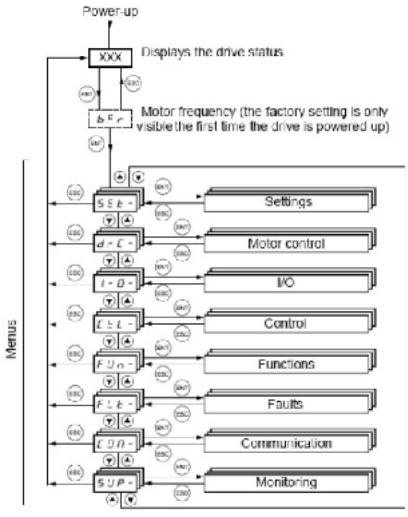
For the given example we use mainly the factory settings on the Altivar. The following parameters, however, have to be changed and you can view them with the following values on the front panel of the Altivar:

In menu I/O:	**TCT** = LEL	i.e. Two wire control uses values 0 or 1
In menu Functions: **STC->**	**NST** = No	i.e. Freewheel Stop is not assigned
In menu Faults:	**ATR** = Yes	i.e. Automatic Restart
	RSF = LI1	i.e. Fault Reset controlled by Logic Input 1
	ETF = No	i.e. No external Fault

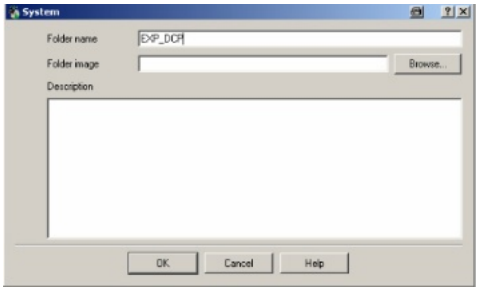
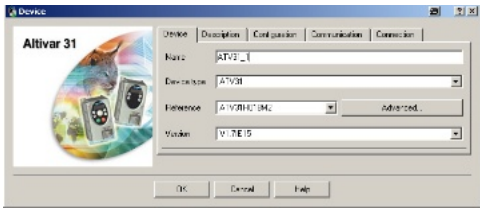
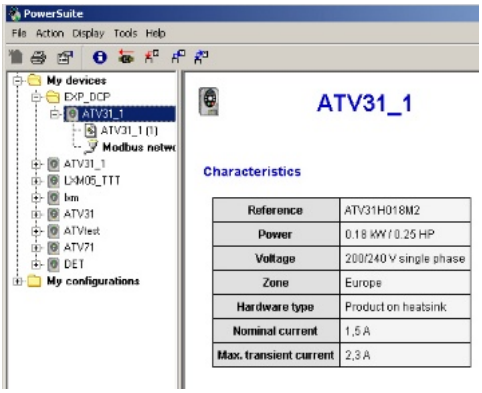
Configuring and setting up the parameters of the Altivar consists of:

- Altivar signal interface
- PowerSuite Start
- Create a device
- Create, Edit and Save a configuration for the device
- Download the configuration to the Altivar

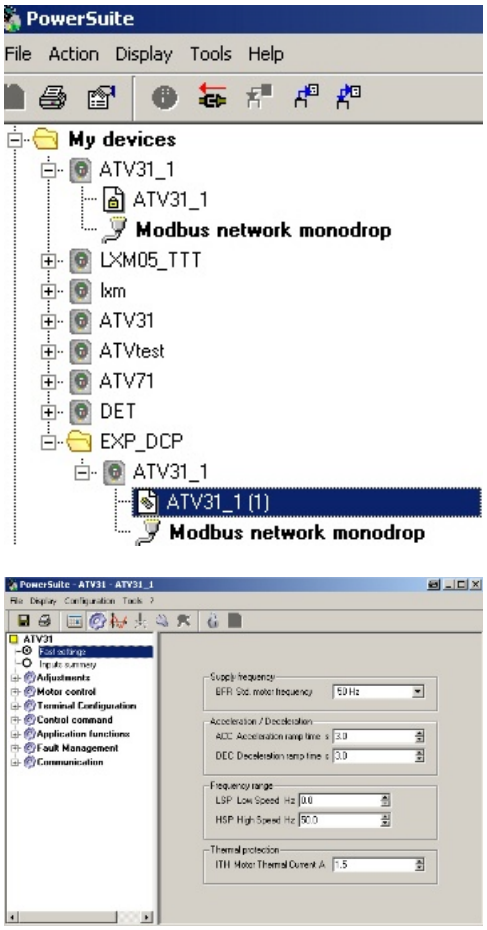
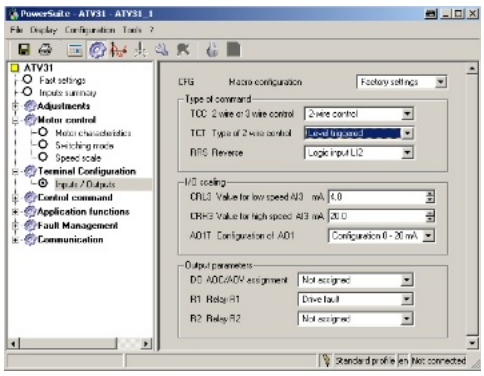
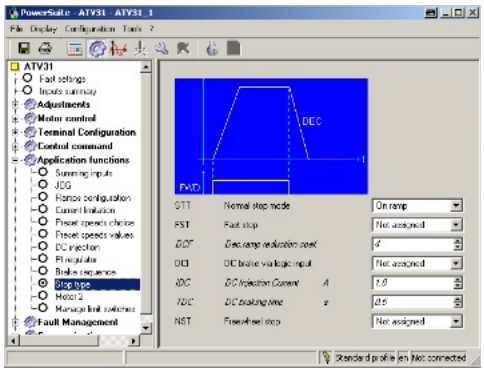
Altivar Signal Interface

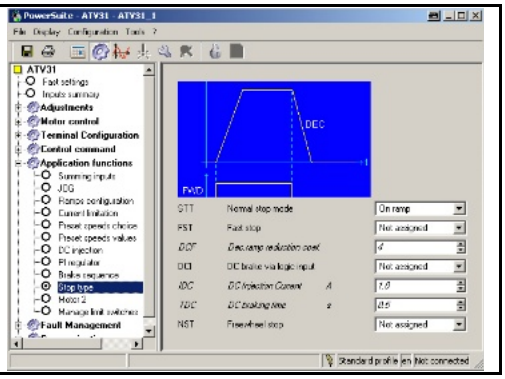
1	**Layout of Altivar 31** RJ45-Interface. The RJ45-Connection is found behind the front panel of the Altivar and is the communications interface of the device. It is used for connecting the Altivar to a PC (i.e. with PowerSuite) or a hand held HMI device. You can use this interface to run the PowerSuite simulator.	
2	**Altivar** **The Signal interface**	
3	**Altivar** **Front Panel Functions**	

Starting PowerSuite

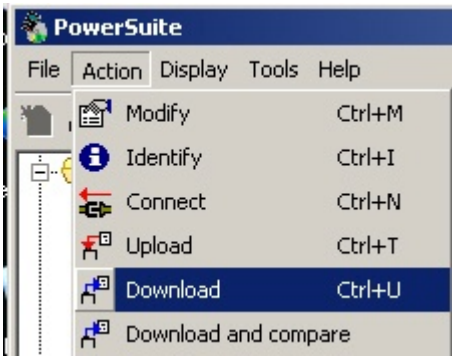
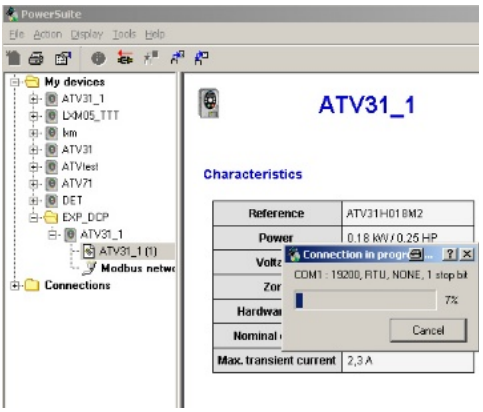
1	Use **File->New->Folder** to create a projekt folder.	
2	Give the folder a name. in our example: **EXP_DCP**	
3	select: **File->New->Device** to create a new device.	
4	In the configuration dialog give the configuration a name and select the reference device – **ATV31H018M2.** **Note:** **The new configuration is created using the factory settings as found in the PowerSuite tool. As software tools change independently of hardware, the settings in your PowerSuite version may be different to those actually delivered with the Altivar. If you wish to be sure of working with the factory settings of your device as it was delivered, you must upload them from the device first.**	
5	The configuration name **EXP_DCP** will appear in the project browser under **My Devices** after you exit the dialog with **OK**	

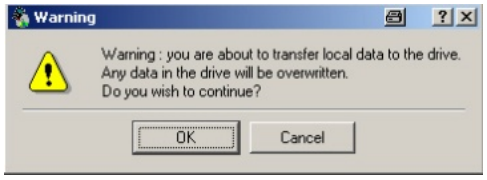
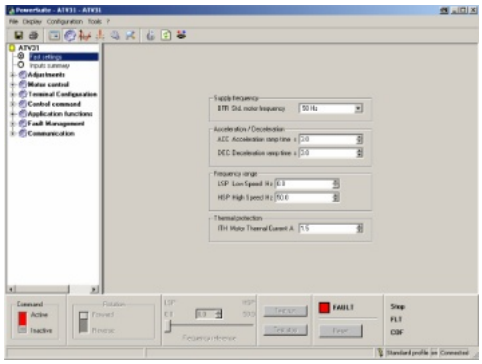
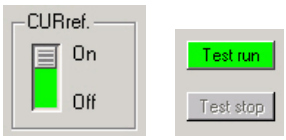
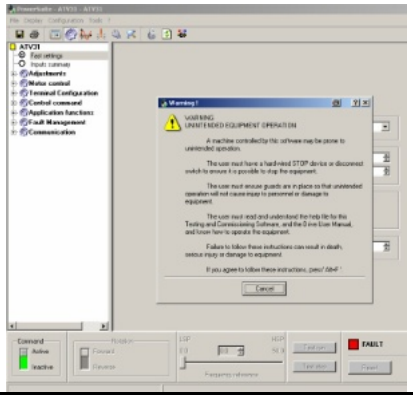
Create, Edit, & Save a Configuration

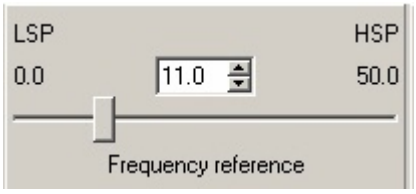
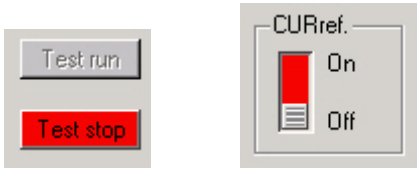
1	Double click on the Configuration name in the browser and the configuration window opens. Here you find the different parameter groups for the drive. When the configuration is newly created the parameters are set at the factory settings. Here we need to make a few changes. Note: you can input these changes on the front panel of the Altivar itself.	
2	In the **Inputs/Outputs** group Set **TCT = Level Triggered (LEL)**	
3	In the **Application Functions** group Select **Stop type** and set **NST = Not Assigned**	

4	In the group **Fault Management** Select **Fault behaviour** and set **ATR** = Yes **RSF** = LI1 **ETF** = Not assigned The adjustments to the factory settings are now finished.	

Download the Configuration to the Altivar

1	Now Load the configuration into the Altivar. Connect the PC (COMxx-Interface 9 Pole plug) to the Altivar controller (RJ45 socket) as described in the hardware section. Use the menu: **Action->Download** to make the connection	
2	A progress bar is displayed while the connection is being made.	

3	Pay careful attention to the warnings – Remember, motion in the system can kill and maim! – Refer to the safety manual for your system if you are not sure. Confirm your agreement with ALT-F. You will then be asked to confirm the over-writing of the contents of the drive. Continue with **OK**.	 
4	The online dialog with simulator starts up automatically. The test mode is started with **CURref. =ON** and **Test run** A warning appears.	  

3	Pay careful attention to the warnings – Remember, motion in the system can kill and maim! – Refer to the safety manual for your system if you are not sure. Confirm your agreement with ALT-F.	
4	With the **Frequency reference** slider you can alter the speed.	
5	End the Test mode with **Test stop** and **CURref. =ON**	

TeSysU

Introduction

This chapter describes how to configure the TesysU motor starter.. The motor starter consists of 3 components; a power base unit, a control unit and a communication module. You also need a wiring kit.

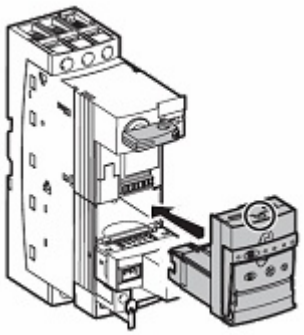
The first section describes the hardware assembly.

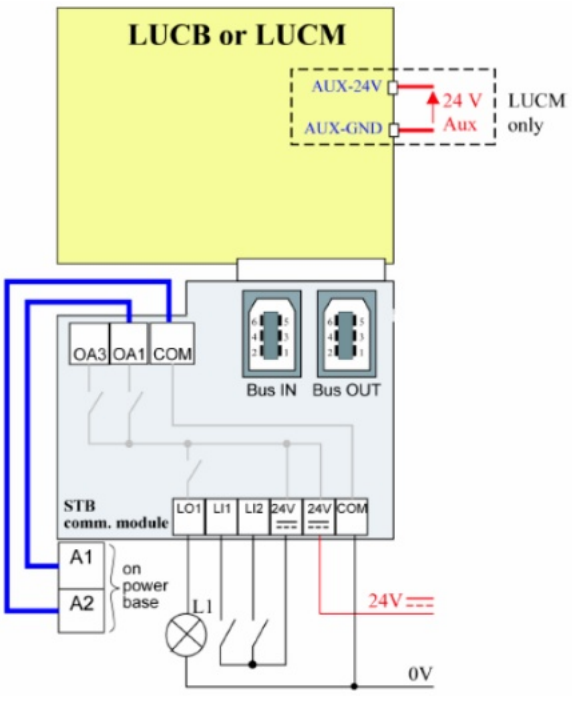
The Advanys configuration software is used to implement the TesysU in combination with the Advantys STB I/O island CANopen expansion.

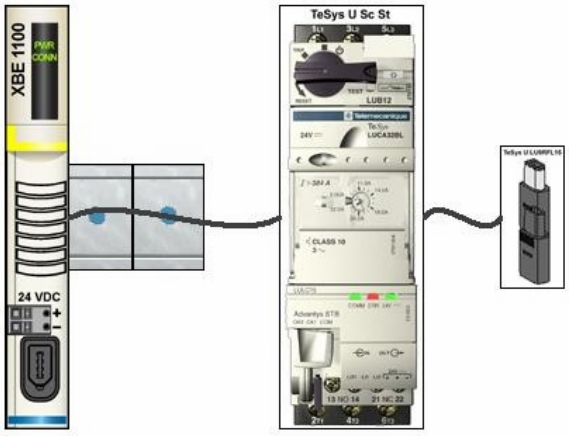
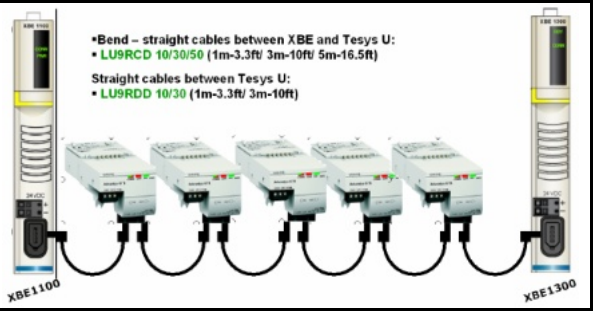
Implementation consists of the following steps:

- Create a new Project (Workspace)
- Configure the hardware (Network interface, Power and I/O module
- Configure CANopen extension communication (baud rate)
- Download the configuration to the Island
- Create a DCF-file

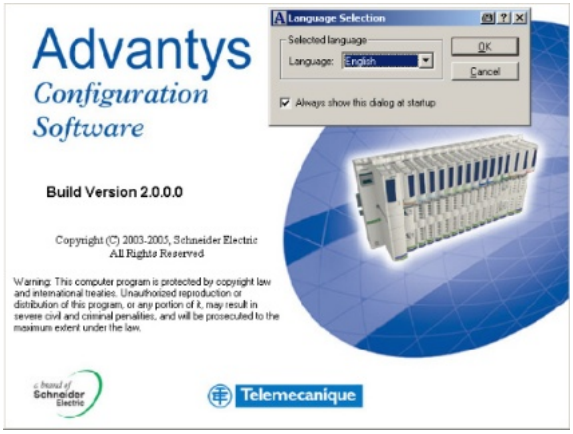
Layout

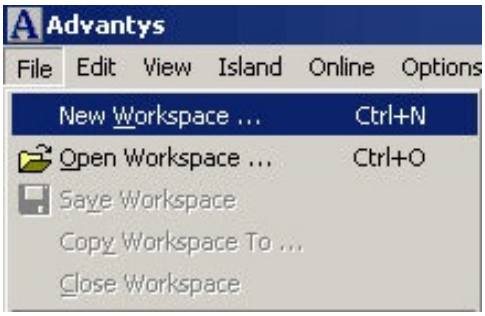
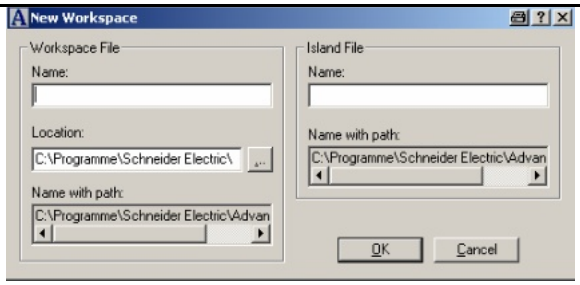
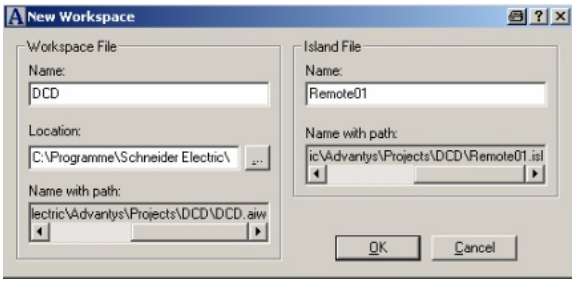
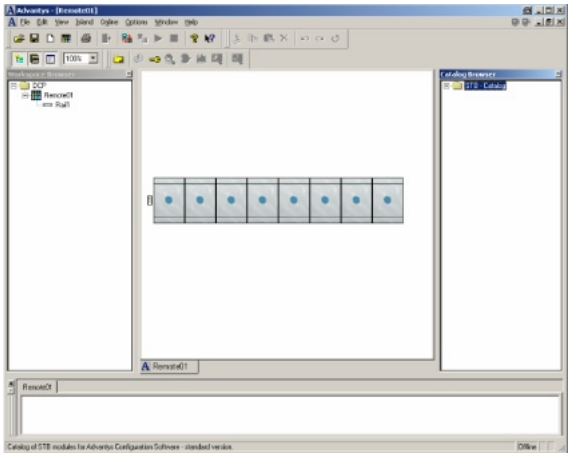
1	The TeSysU motor starter consists of a **Power base** **Control unit** **Wiring kit** and **Communication module**. The individual components can be assembled or exchanged without the need for tools.	

2	CANopen Communication module LULC15 for use with Advantys STB. BUS IN/OUT connections and 24VDC power supply	 
3	TeSys Modell U LULC15 Advantys STB CANopen Module wiring and 24VDC power supply	

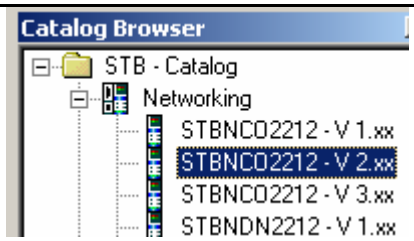
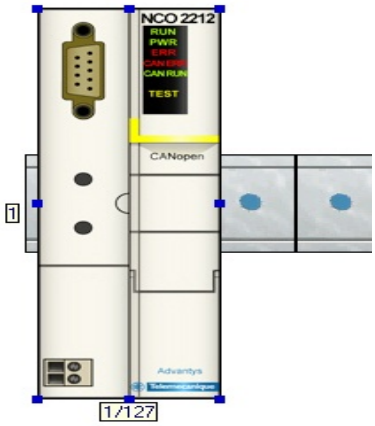
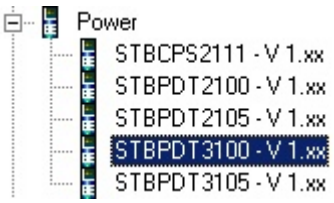
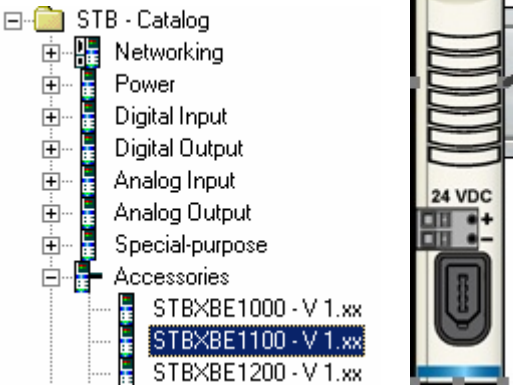
4	TeSysU with Advantys STB via **LULC15 / STBXBE1100** with bus terminal plug **LULRFL19**	
5	TeSysU with Advantys STB via **LULC15 / STBXBE1100** und **LU9RCD10 / LU9RDD10** cable	

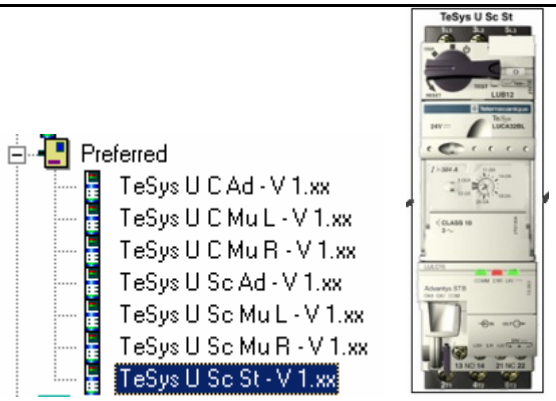
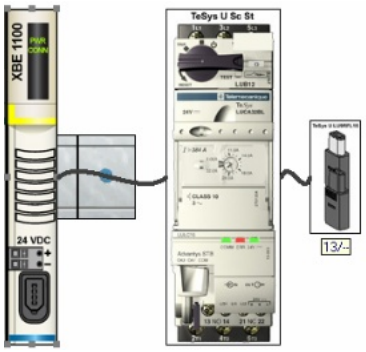
Creating a New Project (Workspace)

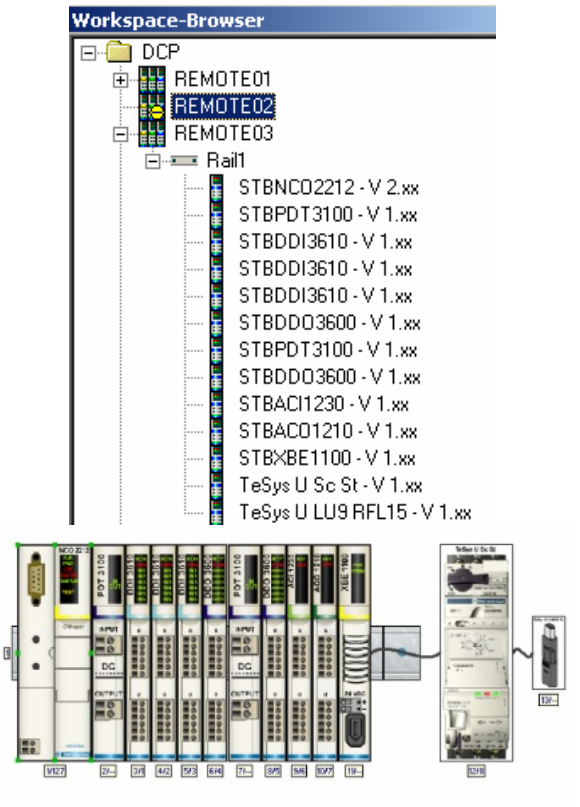
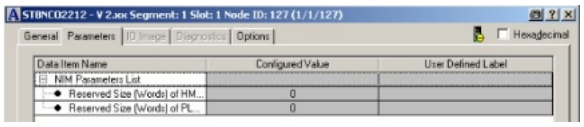
1	Once you have installed and started the Advantys Configuration Software, you will be presented with a choice between **Advantys STB** Advantys FTB, FTM, and OTB. Select the first of these options.	
2	Next select the language.	

3	After starting the Advantys software, you must create a new Workspace**:** **File->New Workspace…**	
4	To do this, specify the **path**, the **workspace name** and the **name** of the first **island**.	 
5	An empty top-hat rail will be displayed.	

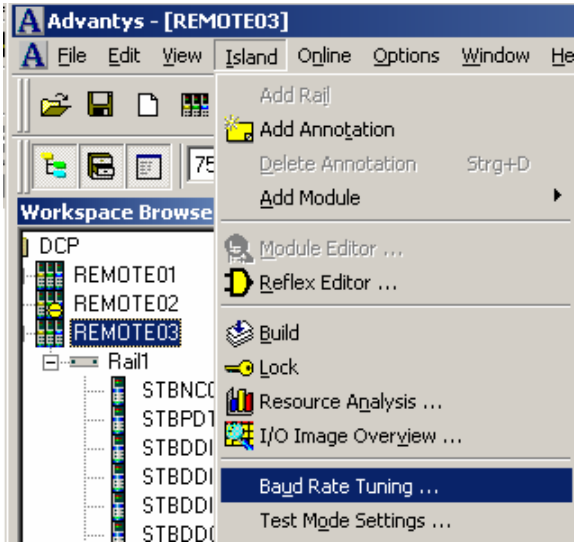
Configuring the hardware

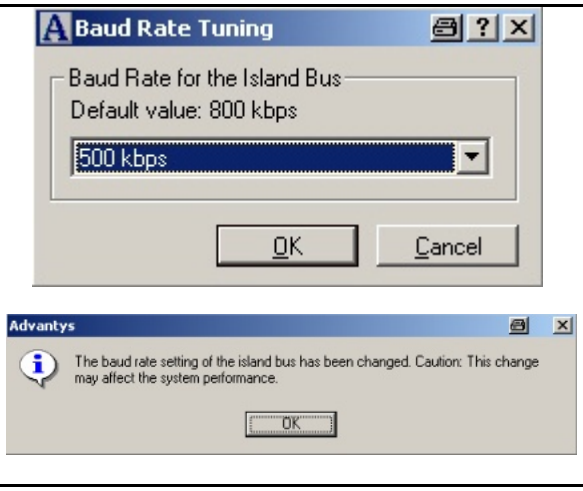
1	Then select the network interface for CANopen: **STB NCO 2212 V2.xx**	 
2	Next, select the power supply **STB PDT 3100** (2x)	
3	Adding the digital and analog I/O-modules has already been handled in the **Advantys STB** chapter and is not described here.	
4	Add the End of Segment Module for STB TesysU **STB XBE1100** (1x)	

5	Add the Standard Control Module STB: **TeSysU Sc ST – V1.xx**	
6	To terminate the internal CANopen add the terminal plug for the TesysU communication module **TeSysU LU9RFL15 – V1.xx**	
7	The TeSysU-STB connection should now look like this:	

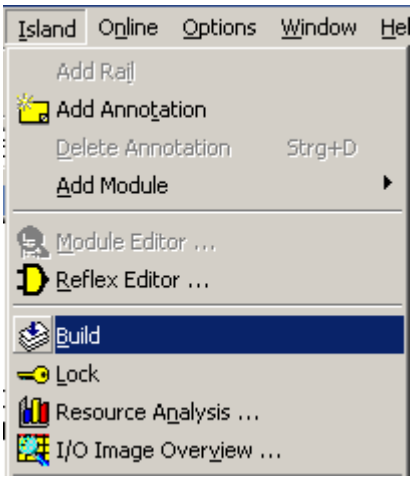
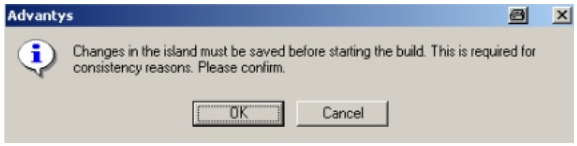
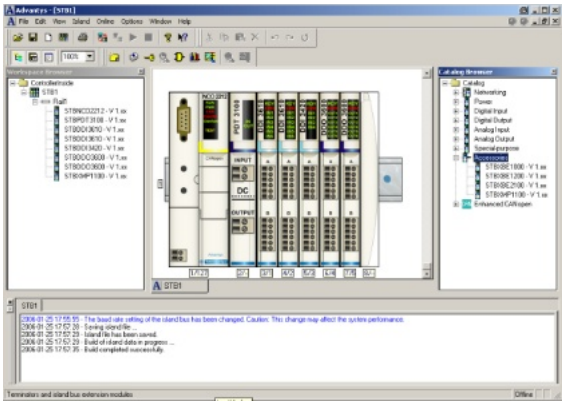
8	Here is the complete STB-I/O-Island **Remote03** and **Remote04**	
9	Double-click on the CANopen header to display the Properties page. Here you can enter a size for your exchange table on the **Parameters** tab. Our example does not involve the use of an exchange table so the value is set to **0**.	

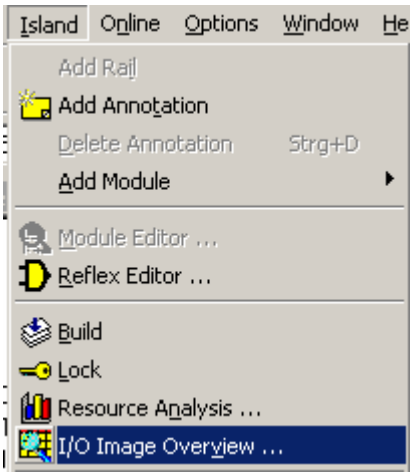
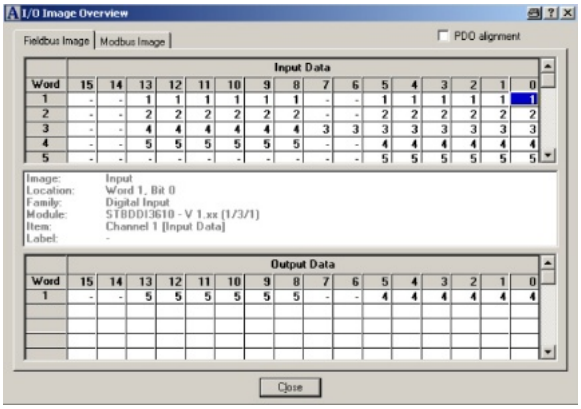
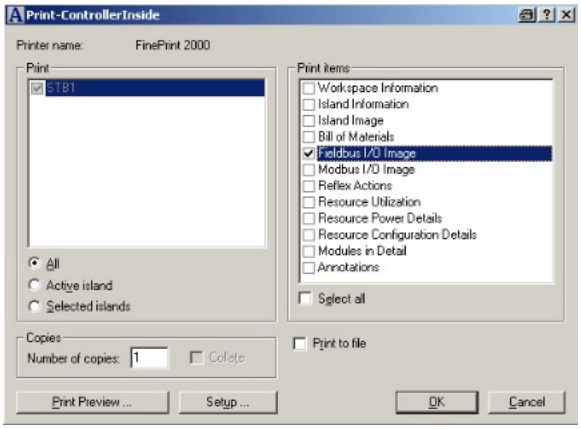
Configuring internal CANopen bus communi-cation (baud rate)

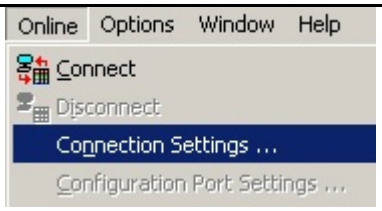
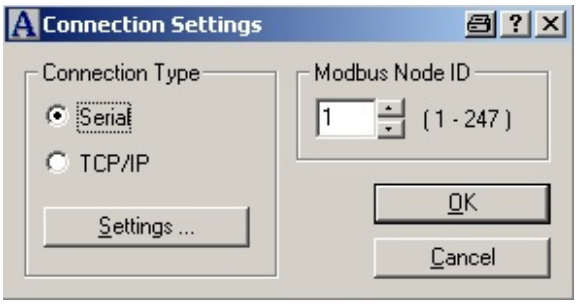
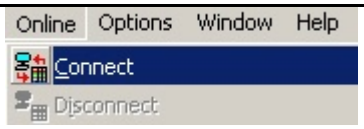
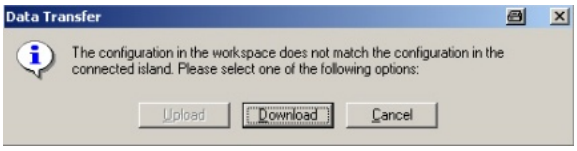
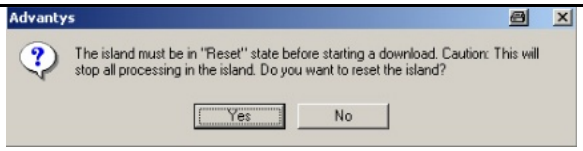
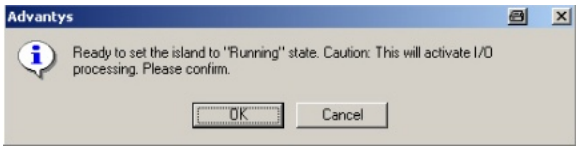
1	The internal baud rate can be set via menu by selecting **Island->Baud Rate Tuning...**.	

2	The rate used is **500 kbps.** **Note:** Set the parameter for the transfer rate between NIM and PLC with the two rotary switches on the front of the NIM. See Communication for further details.	

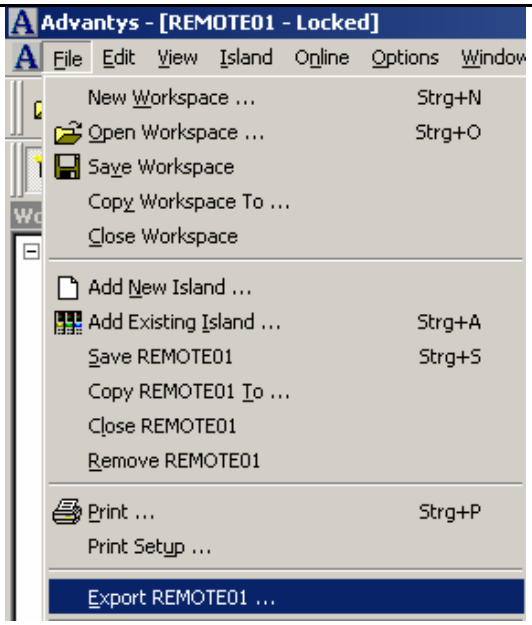
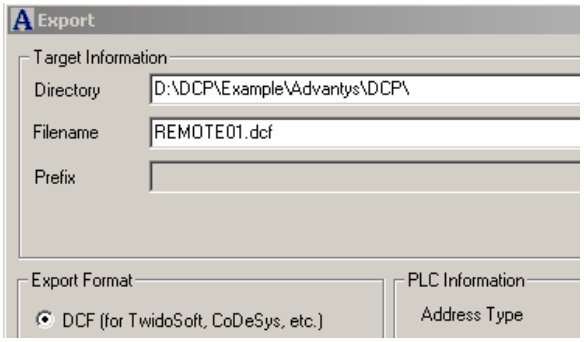
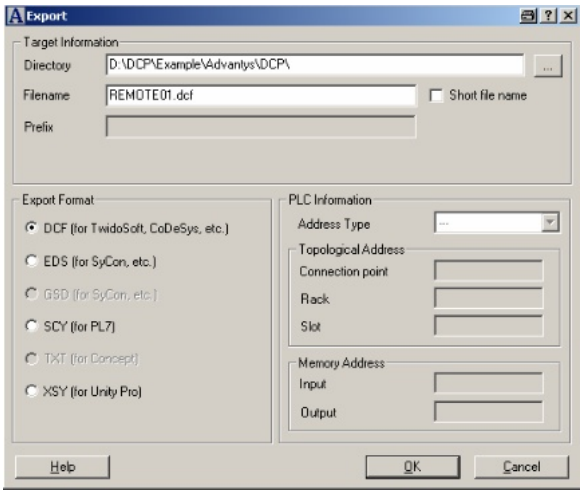
Downloading configuration to island

1	Parameterization is now complete. Now you need to generate the STB project. To do this select: **Island->Build**	
2	Unless you have already saved the changes, you should do this now by clicking **OK.**	
3	The bottom frame provides a log of the individual actions.	

4	The words **... Build completed successfully** should now be visible here.	2007-07-02 15:58:47 - Build started on 2007-07-02, 15:58:47, V2.5.0.1. 2007-07-02 15:58:47 - Source file : C:\DCP\Example\Advantys\REMOTE03.isl 2007-07-02 15:58:47 - Destination file : C:\DCP\Example\Advantys\REMOTE03.bin 2007-07-02 15:58:47 - Reading source file .... 2007-07-02 15:58:47 - Reading global information from core database ... 2007-07-02 15:58:47 - Creating module descriptions ..... 2007-07-02 15:58:49 - Creating Reflex Action descriptions ... 2007-07-02 15:58:49 - Configuring SYNC parameters ... 2007-07-02 15:58:49 - Configuring Heartbeat parameters ... 2007-07-02 15:58:49 - Configuring module identity parameters ... 2007-07-02 15:58:49 - Configuring PDO parameters ... 2007-07-02 15:58:49 - Configuring module parameters ... 2007-07-02 15:58:49 - Writing destination file .... 2007-07-02 15:58:49 - Build completed successfully. (Configuration size: 2834 bytes.)
5	To find out how the individual inputs and outputs fit into the data exchange process, you can call the **I/O Image Overview.**	
6	You must ensure that the **Fieldbus Image** tab is selected. Select an input or output word element here to display its content in the intermediate window. **Note:** Alternatively, this information can also be printed out. To do this, select **Fieldbus Image.**	 

7	Before you can load the configuration, you must first define the appropriate settings under: **Online->Connection Settings…** As the serial cable is being used, select the following: **Serial** **Modbus Node ID: 1**	 
8	Establish a connection via **Online->Connect** The NIM and PC configuration versions will now be compared. If they do not match, this window (see right) will appear. Select **Download.**	 
9	Then respond to the subsequent messages with **Yes** And **OK**.	  
10	The text shown on the right should now appear in the bottom window.	**Island is healthy.**
11	You can now terminate the connection with: **Online->Disconnect**	

Create a DCF-Datei

1	To create a DCF file select: **File->Export Remote01...**	
2	Enter a folder and file name. Click on **OK** to perform the export. Note: The DCF file is required by UnityPro V3.0 to process the CANopen connection.	 

PowerSuite Software

Introduction

PowerSuite is a tool for configuring and supervising motor controllers. Using PowerSuite, the user can define machines, their configuration and the communications parameters.

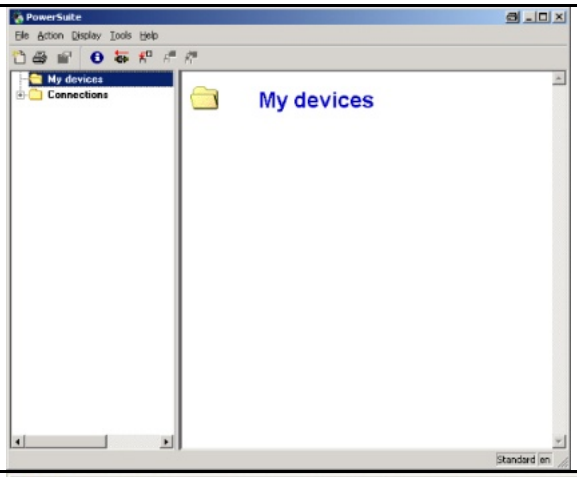
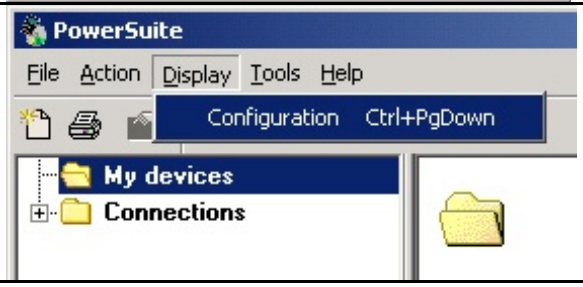
PowerSuite offers the advantage that you can

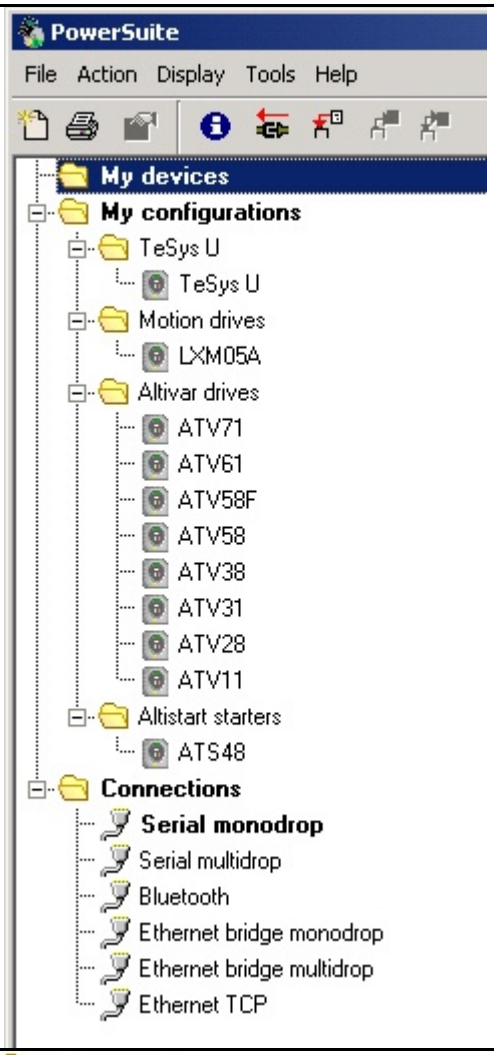
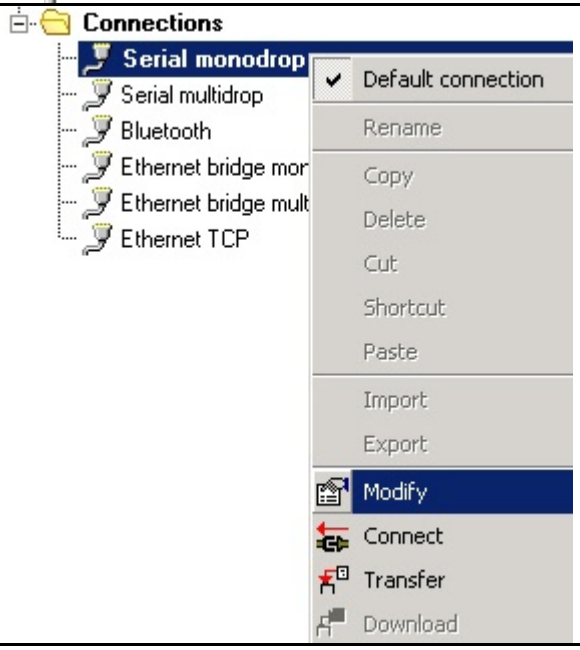
- Save data to your hard drive and duplicate it
- Print documentation for your project
- Optimise your parameters online

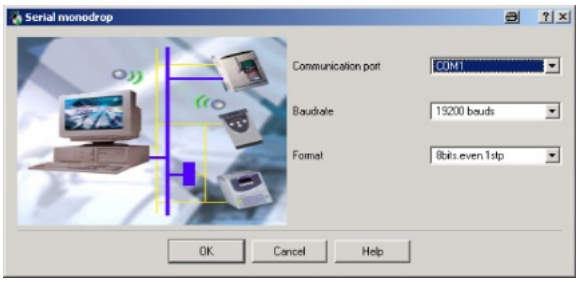
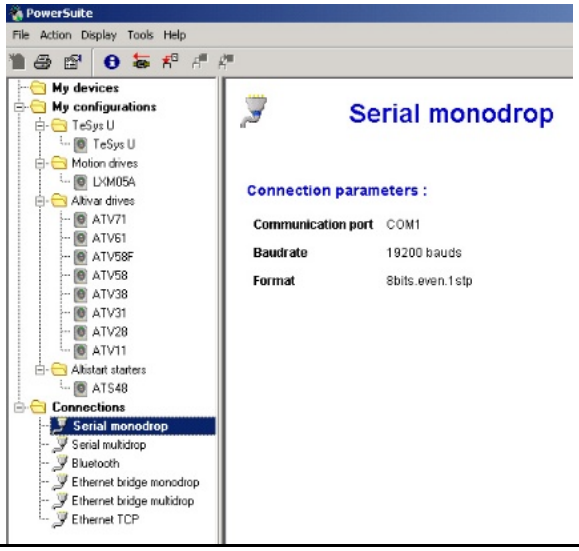
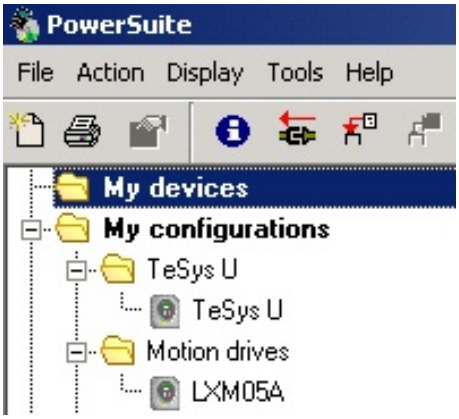
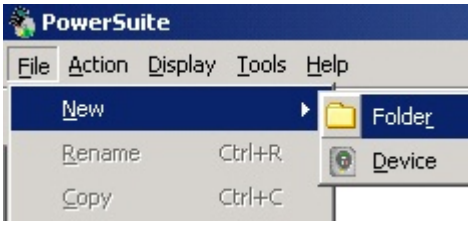
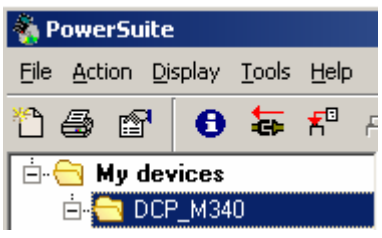
The version described here can be used for all controllers used in this configuration. (Lexium05 and Altivar 31).
.

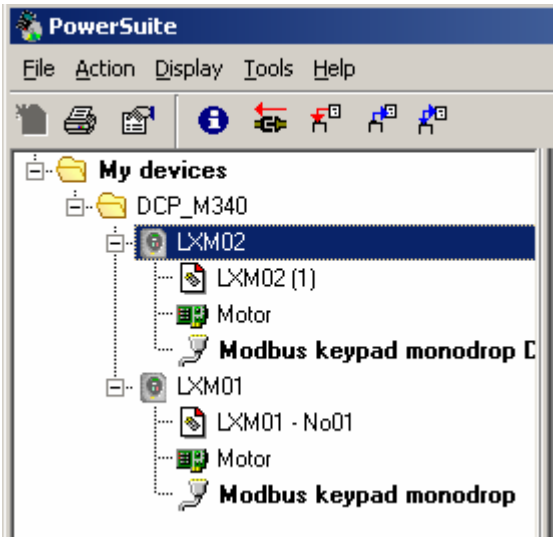
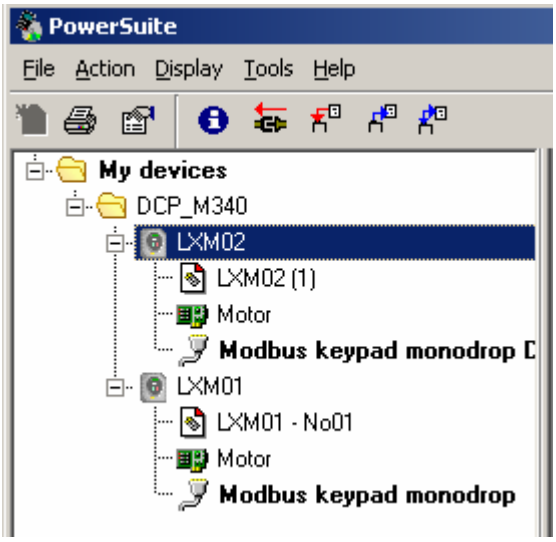
General Setup

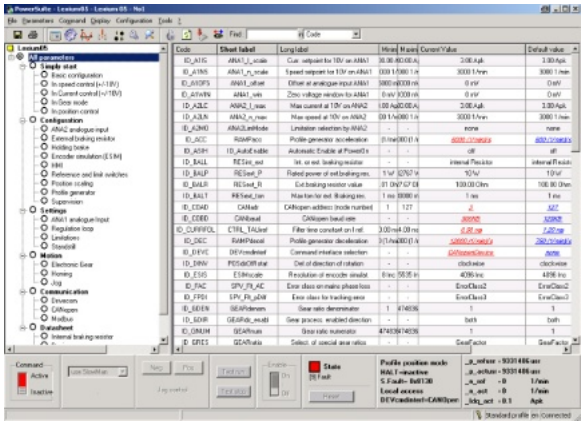
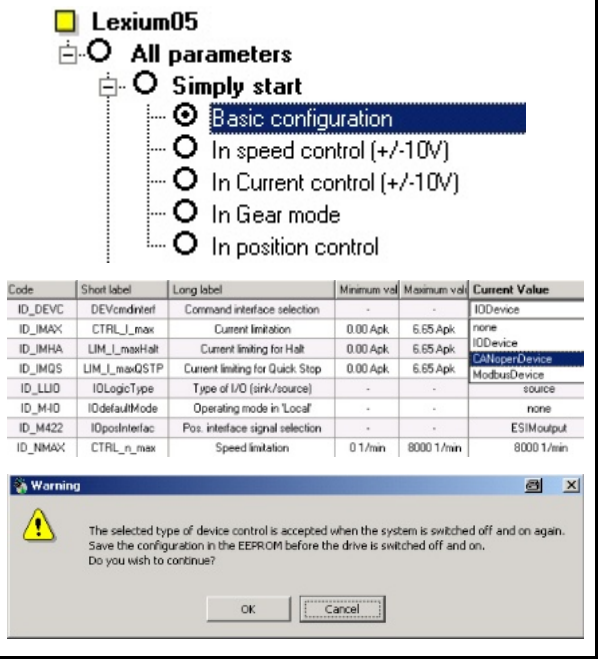
The following describes the basic setup of PowerSuite:

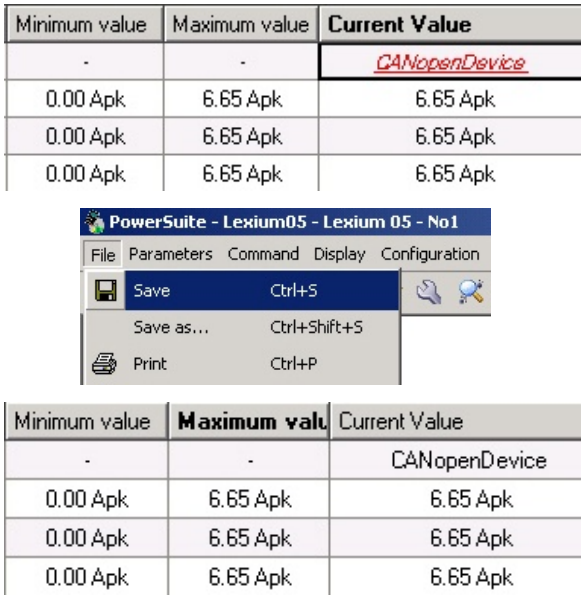
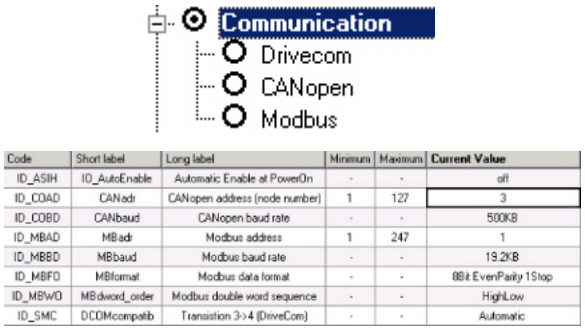
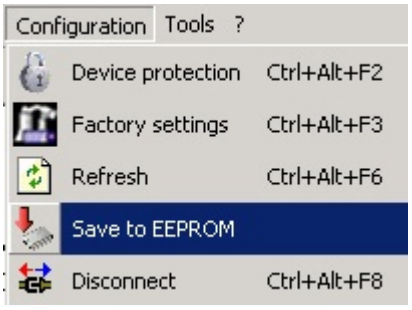
1	After starting PowerSuite you see the display on the right.	
2	Select **Display->Configuration** so that the configuration is added to the project browser list.	

3	The project browser aids you in managing your devices.	
4	In **Connections** you can view the communications setup and **Modify** it.	

5	In **Serial monodrop** Sslect the **COM** interface you wish to use.	
6	When you select the connection in the project browser, the description appears in the window on the right.	
7	Powersuite allows you to create separate folders for your drives. To do this click on the main folder: **My Devices**	
8	Then select: **File->New->Folder**	
9	In the dialog that opens, input the **Folder name**. You can also add a **Folder image** and a **Description**. Exit with **OK** to add the folder to the project browser	

10	You can group all the drives and export the data.	
11	Once PowerSuite is connected to the drive you can used the control panel to operate the drive	
12	The data is read from the Lexium05.	
13	In the project browser you can select a drive using double click.	

14	After the data transmission has finished the data is displayed.	**LXM05 - No01** **Characteristics** Reference: LXM05AD10M2 Nominal Power: 0,75 kW Supply Voltage: 200 / 240 V 1~ Maximum transient current (peak): 10 Apk Maximum continuous current (rms): 4 Arms Interface: CANopen,Modbus RTU,P/D,+/-10V **Structure** Device: LXM05AD10M2, 01610002197, P840.10 V1.1IE20, Telemecanique Control Board: Telemecanique Motor: BSH0701P.1 Family : BSH Size : 070 Length : 1, 2006040180, Telemecanique **Configuration(s)** Name: LXM05 - No01 Software release: P840.10 V1.1IE20
15	The parameters can be displayed and modified in list or page format. Use : **Display->List** oder **Pages**.	
16	Select: **Simply start->Basic configuration.** In the row for defining the **command interface selection,** you should select **CANopenDevice** as the **Current Value.** The servo drive will now be enabled for control via CANopen. In order for this change to take effect on the Lexium05, you will need to switch the device off and then back on again.	

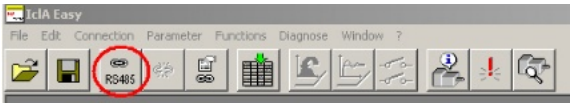
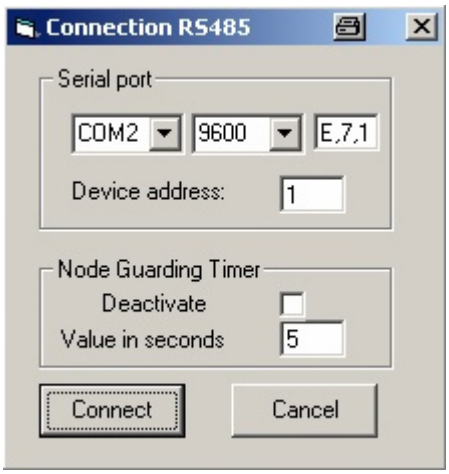
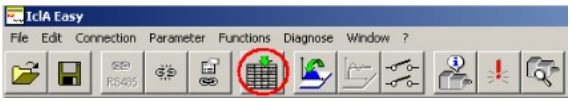
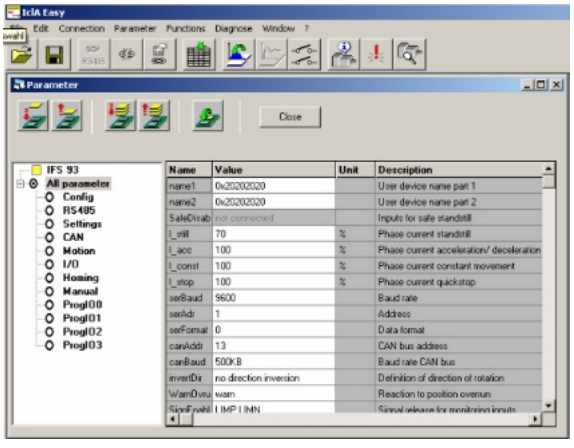
17	Initially, the change will be highlighted in red, but the display color will change when you select: **File->Save**.	
18	Make the following settings under **Communication**: **CANopen address: 4...7 31, 32, 41, 42** **CANopen baud rate: 500** **Modbus address: 1**	
19	To transfer the settings to the Lexium05, select: **Configuration->** **Save to EEPROM**	
20	The settings will now be transferred. To ensure that the settings are saved on the servo drive, you need to confirm the prompt by clicking **OK.** PowerSuite confirms the save when completed	

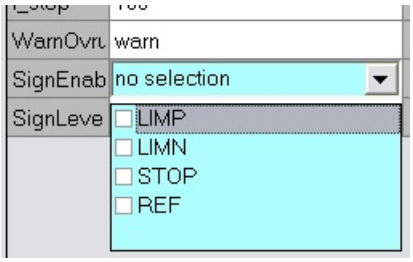
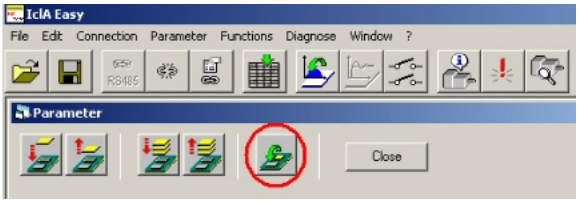
IcIA Easy Software

General

The IcIA IFS parameters can be modified using the IcIA EASY software.

The IcIA EASY software is available free on the Berger Lahr website. The RS485 interface programming cable is also required.

1	After starting up IcIA EASY, click the **Connect** icon...	
2	...to open the **Connection RS485** dialog box. In this dialog box, only the COM port used by the interface needs to be set.(COM 2 .... COM 6) The other settings should be left unchanged. The COM port can be read from the MS Windows' Hardware Manager.	
3	Once the connection has been established, click on the icon to display the parameter list.	
4	Here, all the IcIA IFS parameters are listed in their relevant groups. Under the **Settings** entry...	

5	...the limit switch should be disabled using **SignEnab**. To do this, remove the check marks next to **LIMP**, **LIMN** and **STOP**.	
6	... now the parameters are transferred to the IcLa**.**	
7	The parameters must now be loaded into the drive's EEPROM. To do this, click the **Save device parameters in EEPROM** icon.	

Appendix

Detailed Component List

Hardware Components	Pos.	Amt.	Description	Part-Number	Rev./ Vers.
Performance	1.01	1	3-pin master switch 36 kA NS100N	29003	
	1.02	1	Trip block	29035	
	1.03	1	Terminal cover	29321	
	1.04	1	Rotary drive	29340	
	1.05	1	230/24 V DC, 10 A power supply unit	ABL7RP2410	
	1.06	4	Power Supply 230/24VDC, 5A	ABL7RP2405	
	1.07	1	Power Supply 230/24VDC, 20A	ABL7UPS24200	
PLC	2.01	1	PLC CPU with CANopen and Ethernet	BMXP342030	V3.0IE28
	2.02	1	Rack with 8 slots	BMXXBP0800	
	2.03	1	Power supply	BMXCPS3020	
	2.04	3	M340 Digital Input card 16 Channels	BMXDDI1602	
	2.05	2	M340 Digital Ouput card 16 Channels	BMXDDO1602	
	2.06	1	Analog output card, 4 channels	BMXAMI0410	
	2.07	1	Analog output card, 2 channels	BMXAMO0210	
	2.08	2	20-pin terminal block	BMXFTB2020	
	2.09	1	ConneXium Ethernet Switch 5-Port	499NES25100	
HMI	3.01	1	Magelis 5.7" operator terminal	XBTGT2330	V4.5.0. 2955
	3.02	1	Ethernet Cable	490NTW000 0x	
Remote I/O	4.00	4	Advantys STB I/O Island		
	4.01	4	Fieldbus coupling module for CANopen	STBNCO2212	V2.2
	4.02	8	Power input module 24V	STBPDT3100	
	4.03	8	Module mounting for PDT3100	STBXBA2200	
	4.04	12	Digital Input module, 6 channels	STBDDI3610	
	4.05	28	Module mounting for DDI3610, DD03600, ACI1230, ACO1210	STBXBA1000	
	4.06	8	Digital Output module, 6 channels	STBDDO3600	
	4.07	4	Analog Input module I, 2 channels	STBACI1230	
	4.08	4	Analog Output module, 2 channels	STBACO1210	
	4.09	2	STB CANopen Extension	STBXBE2100	
	4.10	2	STB Segment Terminator	STBXBE1100	
	4.11	3	Advantys FTB CANopen I/O-Module, IP67, 12 In, 4 Out	FTB1CN12E04SPO	
	4.12	1	Cover 7/8-connection	FTXC78B	
	4.13	3	Cover M12-connection (10Stk.)	FTXCM12B	
	4.14	3	FTB CANopen connecting cable	FTXCN3220	
	4.15	3	FTB Power supply cable	FTXDP2206	

Servo Drives	5.01	8	Lexium05, 0,75kW, 230VAC single phase	LXM05AD10M2	V1.2
	5.02	8	Servo motor for Lexium05	BSH0702P02A2A	
	5.03	8	Encoder cable for LXM05, 3m	VW3M8101R30	
	5.04	8	Motor cable for LXM15 and LXM05, 3m	VW3M5101R30	
	5.05	6	Altivar31, 0,18kW, 230VAC single phase	ATV31H018M2	V1.7
Motor Starter	6.00	2	TeSys U Basis unit 12A 400V	LUB12	
	6.01	2	Control unit Standard	LUCB05BL	
	6.02	2	CANopen Adapter for TeSysU	LULC15	
	6.03	2	Coil wiring kit	LU9BN11C	
	6.04	2	Communication cable LU9RCDxx LULC15 to Advantys STBXBE1100.	LU9RCD030	
	6.05	2	TeSysU End switch for LULC15	LU9RFL15	
Safety	7.01	5	Preventa Safety module	XPSAF5130	
	7.02	1	Preventa Safety extension module	XPSECP5131	
	7.03	5	E-STOP, trigger action	XALK178G	
CANopen	8.01	5	CANopen TAP	VW3CANTAP2	
	8.02	10	CANopen connection cable RJ45	VW3CANCARR1	
	8.03	10	CANopen Plug 90° with extra Port	TSXCANKCDF90TP	
	8.04	8	CANopen Plug 90°	TSXCANKCDF90T	
	8.05	8	CANopen Plug 180°	TSXCANKCFD180T	
	8.06		CANopen Cable	TCXCANC x yy	
IclA IFS93	9.01	2	Step Motor with positioning, CANopen-interface, Industrial connectors	(BL) IFS93/2 CANISDS/3D-I54/O-001RPP41 (BL) 066109300098	Rev.1.107
	9.02	2	Power cable, Industial connector with open ended cable, 3m	(BL) 0062501470030	
	9.03	2	CANopen cable Industrial connector with M12, 3m	(BL) 62501526001	
	9.04	2	fuse motor current	(MG) 25085	
	9.05	1	Phaseo Power Supply 400V, 240W, 24VDC	ABL7UPS24200	
	9.06	2	motor cicuit breaker, fused Transformer	GV2-ME08	
	9.07	2	Auxiliary switch mounting for contactor.	GVAE11	
	9.08	2	contactor 9A, 24V-trigger, GL	LC1D09BL	

Software Components

Pos.	Amt.	Description	Part-Number	Rev./ Vers.
1.1	1	UnityPro		V3.0
1.2	1	VijeoDesigner		V4.5
1.3	1	Advantys Configuration Tool		V2.501
1.4	1	PowerSuite		V2.40
1.5	1	IclA Easy		V1.104

Component Protection Classes

Positioning	Component	In Field, On Site			Cabinet		
					Front		Inside
Protection Class		IP54	IP65	IP67	IP55	IP65	IP20
	Master Switch						X
	Maintenance Switch		X				
	E-STOP housing		X				
	contactor, 24VDC triggered, 3 pole AC3, 1NO+1NC						X
	Illuminated button, all forms and colours		X				
	auxiliary switch module with LED + 1 auxiliary switch (1x N/O), all colours						X
	Label holder 30x40, all Texts		X				
	LS-Switch all sizes and denominations						X
	motor circuit breaker, all sizes and denominationsd						X
	Phaseo Power supply 24 V DC/1,2 A						X
	PLC Modicon M340						X
	Magelis XBTGT Operators Terminal					X	X
	Remote I/O Island Advantys STB						X
	Remote I/O Island Advantys FTB			X			
	Servo drive Lexium05						X
	Intelligent Compact drive IclA IFS (Drive format IP41)	X					
	Variable speed Drive ATV31						X
	Motor starter TeSysU						X

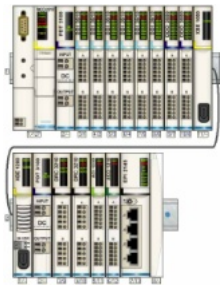
Component Features

Components

Programmable logic controller – Modicon M340

- For complex machines in medium-sized infrastructures
- CPU with up to two integrated bus interfaces (CANopen, TCP/IP and/or Modbus)
- Unity Pro software used for programming
- Flexible width with 4, 6, 8 or 12 slots
- 24 V DC or 100 – 240 V AC power supply
- Cards available for digital or analog I/O, counters, communication and networks
- Compact card dimensions: 32 x 100 x 93 mm (W x H x D)
- USB programming connection
- Large internal memory and slot for additional SD memory card

Modular I/O-Island Advantys STB



- open modular I/O System
- Easy commissioning using Advantys Configuration Software
- Power Supply, Data Capture and Power Control via Island bus
- Field bus coupling module for Ethernet TCP/IP, CANopen, Modbus Plus, Fipio, INTERBUS, Profibus DP and DeviceNet
- Gateway Function with CANopen expansion module
- EMC-Protected
- Certificates UL, CSA

Advantys FTB I/O module

- Suitable for use in harsh environments
- Metallic version available for extremely harsh environments
- Communication via: Profibus DP, CANopen, DeviceNet, INTERBUS
- Each channel has an LED status display
- Short-circuit-proof inputs and outputs
- Temperature range: 0 to +55°C
- Approval: UL

Lexium 05 servo drive

- Voltage range:
  Single-phase 100 – 120 V AC or 200 – 240 V AC
  Three-phase 200 – 240 V AC or 380 – 480 V AC
- Power: 0.4 to 6 kW
- Rated torque: 0.5 to 36 Nm
- Rated speed: 1500 to 8000 rpm
- The compact design allows for space-saving installation of the drive in control cabinets or machines.
- Features the "Power Removal" (Safe Stop) safety function, which prevents the motor from being started accidentally. Category 3 with machine standard EN 954-1
- Lexium 05 servo amplifiers are fitted with a brake resistor as standard (an external brake resistor is optional)
- Quick control loop scan time: 62.5 µs for current control loop, 250 µs for speed control loop and 250 µs for position control loop
- Operating modes: Point-to-point positioning (relative and absolute), electronic gears, speed profile, speed control and manual operation for straightforward setup.
- Control interfaces:
  CANopen, Modbus or Profibus DP
  Analog reference inputs with ± 10 V
  Logic inputs and outputs
- The PowerSuite dialog tool enables the Lexium 05 servo drive to be configured, set and tested.

Components
contd.

Altivar Variable Speed Drive

ATV31H018M2

The Altivar 31 drive is a frequency inverter for 3-phase squirrel cage asynchronous motors. The Altivar 31 is robust, compact, easy to use and conforms to EN 50190, IEC/EN 61800-2, IEC/EN 61800-3 standards UL/CSA certification and to CE marking.

It incorporates functions that are suitable for the most common applications, including:
- Materials handling (small conveyors, hoists, etc),
- Packing and packaging machines,
- Specialist machines (mixers, kneaders, textile machines, etc.),
- Pumps, compressors, fans.

Altivar 31 drives communicate on Modbus and CANopen industrial buses. These two
protocols are integrated as standard into the drive.
Altivar 31 drives are supplied with a heatsink for normal environments and ventilated enclosures. Multiple units can be mounted side by side to save space.

Drives are available for motor ratings between 0.18 kW and 15 kW, with four types of power supply:
- 200 V to 240 V single phase, 0.18 kW to 2.2 kW
- 200 V to 240 V 3-phase, 0.18 kW to 15 kW
- 380 V to 500 V 3-phase, 0.37 kW to 15 kW
- 525 V to 600 V 3-phase, 0.75 kW to 15 kW

Intelligent Compact Drive IclA IFS93

- Performance 74/117W (24/36 VDC Einspeisung)
- Maximum current 6A
- Torque 0,17Nm; 3,1 ... 11Nm (with gear box)
- High continuous static torque
- Revolutions bis 4800 1/min (ohne Getriebe)
- Temperature Range: 0..+ 65°C
- Compact - Motor and Elektronic in one unit
- simple installation and highl EMC
- Optional industrial sockets
- Communication via CANopen, Profibus DP oder RS485
- Modes: Referencing, Manual, Point to point, speed
- Certificates cRLus, CE

Components contd.

TeSys Model U

- One Base Unit
- Control Unit 0,15 - 32A
  - Only 6 settings up to 32A
  - Only 4 voltage ranges up to 240V AC/DC
  - 3 Variants: Standard, Extended, Multi-function
- Width 45 mm
- Complete reversing combination 0,15 - 32A
- Auxiliary switch & Function module
  - Integrated: Auxiliary contact motor contactor 1NC, wired
  - Integrated: Auxiliary contact contactor 1NO+1NC, available
  - Option: Auxiliary switch module with 2 contacts protected
  - Option: Message contact "Error" und "Switch Position"
  - Function module Alarm Over-heating
  - Function module display Motor power (0-10V, 4-20mA)
  - Function module Multiple error display (coming soon)
- Communication modules
  - Parallel wiring, for 8 motors on one sub-station module
  - Modbus Protocol RTU
  - AS-Interface
  - CANopen

Gateway: FIP IO/Modbus, DeviceNet/Modbus, Profibus DP/Modbus

Components
contd.

Magelis XBT GT 2xxx operator terminal

- High-definition display
  - Extremely sharp picture with 65,536 colours (TFT), 4,096 colours (STN) or 16 shades of gray, depending on the model
  - Adjustable contrast and brightness
  - QVGA, 320 x 240 pixel resolution
  - Analog touch screen so that objects can be positioned freely
- Compact
  - Small dimensions: Only 167 x 135 mm (W x H)
- Communicative
  - Two serial interfaces (RS232C & RS485);
  - One USB port
  - Optional 10/100 BaseT Ethernet port, depending on the model
- Can be used around the world
  - Multilingual applications – up to 10 languages can be used simultaneously
  - Numerous character sets available (Latin, Japanese, Chinese, Cyrillic, etc.)
- Save time when creating applications by using the Vijeo Designer configuration software
  - Library containing over 4,000 ready-to-use symbols, bitmaps and pictograms
  - Preconfigured objects for alarm lists, recipes and trends

Preventa safety relay: XPSAF5130

- Category 3 to EN 954 Part 1
- 24 V DC
- 3 safety-oriented switching contacts
- 1 semiconductor output for PLC
- Slimline design

Phaseo power supply units

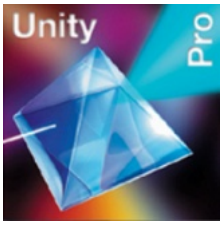
ABL7RE2403, ABL7RE2405 and ABL7UPS24100

- ABL7RE: 100 – 240 V AC/24 V DC
- ABL7UPS: 3-ph. 400 V AC/24 V DC
- 3 A, 5 A or 10 A, secondary
- Can be connected in parallel (ABL7RE)
- Short-circuit-proof and protected against overload Can be reset manually or automatically
- ABL7RE approvals: UL, CSA, TÜV, Ctick
- ABL7UPS approvals: cULus, cRLus

Components
contd.

Unity Pro

UNYSPUEFUCD30



Unity Pro is the combined programming, testing and operating software for the Premium, Modicon M340 and Quantum PLCs.

- Unity Pro supports all 5 IEC 61131-3 programming languages as standard with all test functions via PC simulation or online on the PLC directly.
- Thanks to the icon variables that are independent of the memory, the structured data and the user function blocks, the application objects are mapped directly from the special components of the automated process.
- The user configures the Unity Pro operator screens within the application using the graphical libraries. Operator access is simple and direct.
- The test and maintenance functions are simplified thanks to animated graphic objects.
- For diagnosis, all system and application errors are displayed in plain text and in chronological order (date and time is provided at the origin) in a visualization window. You can return to the source of the conditions that have caused the error using the navigation function for troubleshooting.
- XML format, a Web standard for data exchange, has been used as the source format for Unity applications. The simple import/export functions mean that the entire application or parts of it can be exchanged with other software in your project.
- The converters integrated in Unity Pro automatically convert PL7 and Concept programs into Unity Pro programs.

Vijeo Designer

VJDSPULFUCDV45M

Vijeo Designer configuration software has a number of parameterization windows that enable a project to be developed quickly and simply and are very user-friendly. Vijeo Designer uses Java scripts that allow process data to be further processed on the XBT-G touch panel.

These are some of its functions:

- Navigator
- Library of animated graphic objects
- Online help
- Display of error reports
- Display of object characteristics
- Display of the list of variables

Components
Contd.

Advantys Configuration Software: STB SPU 1000

Proceed as follows to configure an Advantys STB system:

- If applicable, parameterize all the I/O modules on the Advantys STB platform (digital, analog and intelligent modules) with standard functions.
- Generate the dedicated island EDS file, which will be used at a later stage
- Parameterize the reflex functions executed at island level. These parameters are set using the Advantys Configuration Software.

This software can also be used:

- To optimize island performance by specifying priorities to be applied when processing module data
- To add preferred modules or standard CANopen devices (such as FTB, OTB, ATV31, Lexium05, for example)
- To check that the configuration complies with the design guidelines and to check the current consumption
- To change the module's standard functions.

PowerSuite VW3A8104

- The PowerSuite dialog tool enables user-friendly operation of the following devices:
  - Altivar variable speed drive
  - Lexium05 servo drive
  - TeSys model U motor controls
  - Altistart soft starter
- A wide range of functions are integrated for the various application phases, e.g.:
  - Preparing the configurations (for PC),
  - Commissioning (for PC and Pocket PC)
  - Maintenance (for PC and Pocket PC)
- During the startup phase, the device is connected to a PC and can be used:
  - To transfer the configuration that has been made
  - To make settings
  - For monitoring. New functions have now been added for this option such as:
    - the Oscilloscope function.
    - monitoring displays
    - create user specific menus
    - search and sorting parameters
  - To save the final configuration
  - Online Context Help

Components contd.

IcLa Easy

The software *IclA Easy* is provided for working with the intelligent compact drives IclA and can be used to support commissioning, product presentation, diagnostics and testing.

Supported intelligent compact Drives

- IclA IFS
- IclA IFE
- IclA IFA
- IclA IDS (only for diagnostics and only via RS485 interface)

Functions

- Input and display of device parameters
- Archiving and duplication of device parameters
- display of stati and device information
- Positioning the motor with a PC
- Triggering of Reference patterns
- CLC optimisation (only with IclA IFA)
- Access to all documented parameters
- Diagnosis of operational failures
- Provides interfaces to the compact drive

IclA Easy communicates with the compact drive using a field bus converter.

Contact

Author	Telephone	E-Mail
Schneider Electric GmbH Machines and Process Architectures	+49 6182 81 2555	cm.systems@de.schneider-electric.com