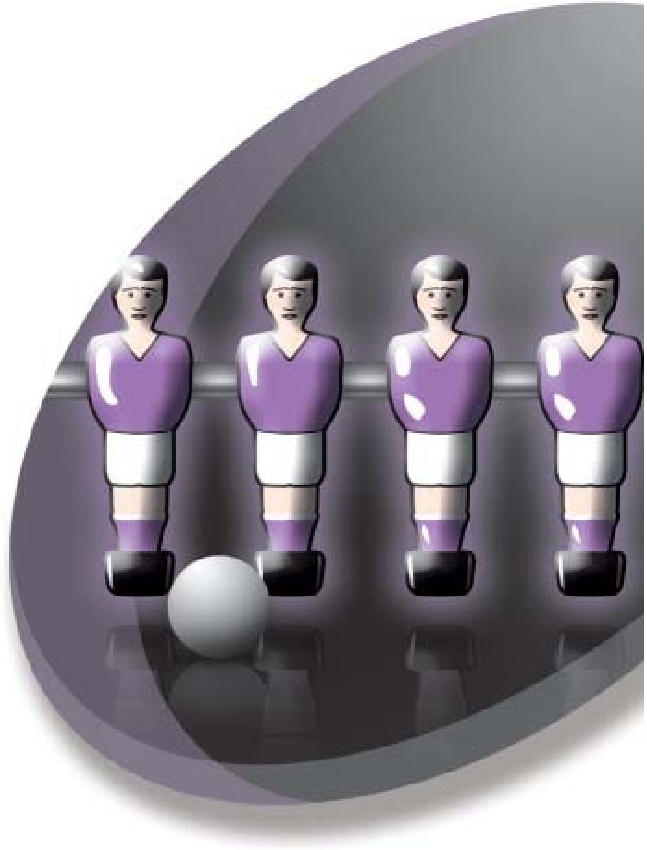
Automation solution for simple machines
Startup Guide

07/2007



a brand of Schneider Electric

Telemecanique

www.telemecanique.com

Table of Contents

Safety Information 7

Part I Presentation of the automation solution for simple machines. 9

At a Glance 9

Chapter 1 Introduction. 11

At a Glance 11
General 12
Technical specifications 14
Description 17
Required hardware 18

Part II Main System 21

At a Glance 21

Chapter 2 Hardware implementation of the main system 23

At a Glance 23
Main cabling 24
HMI cabling 25
CANopen network cabling. 26
I/O cabling 29
Power supply. 31

Chapter 3 Software implementation of the main system 35

At a Glance 35
Installing software and loading applications 36
ATV31 communications configuration. 40

Chapter 4 Presentation of the application 45

At a Glance 45
Operating mode. 46
HMI 47
Grafcet of the ATV31 application 49

Part III Add a Lexium 05 servo drive 51
At a Glance 51

Chapter 5 Hardware implementation of a Lexium 05 servo drive 53
At a Glance 53
Wiring for a Lexium 05 servo drive 54
Cabling of the CANopen network. 55
Power supply 56

Chapter 6 Software implementation of a Lexium 05 servo drive 59
At a Glance 59
Lexium 05 communications configuration 60
Add a Lexium 05 in the Twido controller application 62

Chapter 7 Presentation of the Lexium 05 application 73
At a Glance 73
Operating mode 74
Grafcet of the Lexium 05 application 75

Part IV Add an Advantys OTB I/O module 77
At a Glance 77

Chapter 8 Hardware implementation of an Advantys OTB module 79
At a Glance 79
Advantys OTB outputs cabling. 80
Cabling of the CANopen network. 81
Advantys OTB I/O cabling 82
Power supply 82

Chapter 9 Software implementation of an Advantys OTB module. 83
At a Glance 83
Configuration of Advantys OTB communications 84
Add an Advantys OTB module in the Twido controller application 86

Chapter 10 Add expansion modules to the Advantys OTB. 95
Principle 95
Mount expansion modules. 96
Add expansion modules to the Advantys OTB in the Twido controller application.. 96

Part V Add an Advantys FTB I/O splitter 103

At a Glance 103

Chapter 11 Hardware implementation of an Advantys FTB splitter 105

At a Glance 105

Cabling of the Advantys FTB I/O splitter 106

Cabling of the CANopen network 107

Power supply. 108

Chapter 12 Software implementation of an Advantys FTB splitter 109

At a Glance 109

Configuration of Advantys FTB communications 110

Add an Advantys FTB I/O splitter to the Twido controller application. 112

Appendices 119

At a Glance 119

Appendix A List of symbols used in the application 121

List of symbols used in the application 121

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

⚡ The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

⚠ This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER
DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death or serious injury.

⚠ WARNING
WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

⚠ CAUTION
CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

User Comments

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

Presentation of the automation solution for simple machines

I

At a Glance

Overview

This document presents installation of an automation solution for simple machines.

What's in this Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
1	Introduction	11

Introduction

1

At a Glance

Overview

This chapter presents the automation system described in this document.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
General	12
Technical specifications	14
Description	17
Required hardware	18

General

Introduction

Dedicated to simple installations and small compact machines, the programmable controller Twido, the Altivar 31 variable speed controller, the Lexium 05 servo drive, the Magelis XBTN displayer and the new Advantys OTB and FTB distributed I/O are already renowned for their ability to benefit you in terms of capacity, simplicity and competitiveness.

Today their association is a real solution, that guarantees the ease and speed of implementation as well as the prefect functioning of the whole, thanks to a package complete with cabling and well-integrated software (functional blocks integrated in TwidoSuite).

⚠ CAUTION

This document does not replace in any manner the individual documentation of each product.

It describes in simplified manner the installation, setup and implementation presented.

Descriptions and functional specifications of a specific application are not part of this document.

Nonetheless, this document presents a typical automated solution that can be used.

To facilitate system implementation, the setup files and applications necessary for the presented solution are delivered with the products.

Failure to follow these instructions can result in injury or equipment damage.

Abbreviations / Terminology

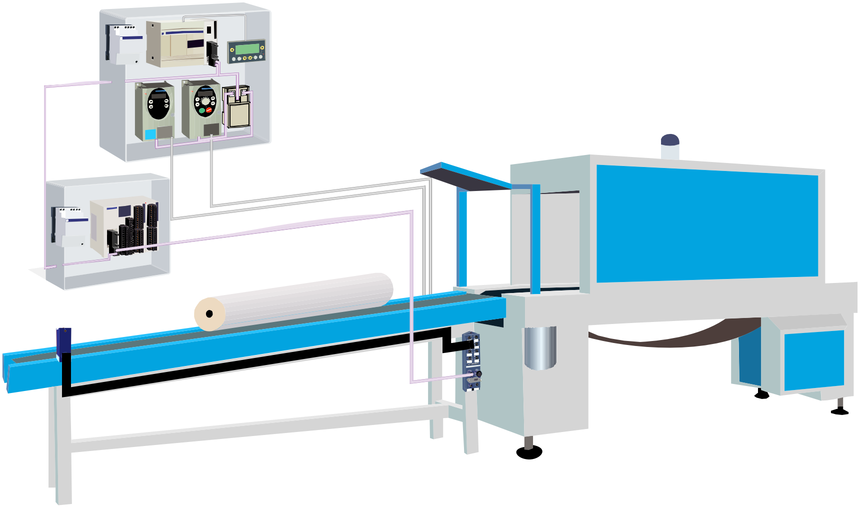
Here are the abbreviations and terms used in this document:

Abbreviation	Description
PC	Personal
XBTN	Displayer with alphanumeric screen
ATV31	Altivar 31 family variable speed controller
Lexium 05	Lexium 05 family servo drive
TAP	Shunt box
OTB	Advantys OTB IP 20 I/O module
FTB	Advantys FTB IP67 I/O splitters
HMI	Human/Machine Interface
AC	AC power
DC	DC power
I/O	Input/Output.
Altivar	Name of the family that includes the Schneider variable speed controllers
Lexium	Name of the family that includes part of the Schneider servo drive
Magelis	Name of the family that includes part of the Schneider HMI
Twido	Name of the family that includes part of the Schneider controllers

Technical specifications

Example of an automation solution

The diagram below shows an example of an automation solution:



The procedure for automating a simple machine of this type is shown in the following table:

For	Use
Monitor the operation, coordinate, configure and control the different sensors actuators	Twido controller
View and/or configure the different parameters of the application	A Magelis Human/Machine Interface
Operate the belt	An Altivar variable speed controller
Position the roller	A Lexium servo drive
Detect and control the products	Advantys remote inputs/outputs linked to sensors/actuators

To respond to these simplified technical specifications, we propose a complete solution outlined in this guide.

Objective

The automation solution presented in this document shows how to control one or several motors. To do this, we propose to link an ATV31 variable speed controller and a Lexium 05 servo drive with a Twido controller on the CANopen bus. The Twido controller includes the Macro Drives developed for the ATV31 controllers and the Lexium 05 drives (Macro: ready-to-use function blocks integrated in the TwidoSuite library). These Macro Drives include micro applications that simplify the development of a control application for an ATV31 or a Lexium 05 connected to the CANopen open fieldbus network.

It is possible to assign a name to each of the variables used in the Twido application, called SYMBOL.

To display and possibly modify the parameters of the system (example: motor speed), you add an XBTN operator display. The XBTN and the TwidoSuite application can use the same list of symbols (exported from TwidoSuite to XBT-L1000). This link facilitates the development of the operator display application.

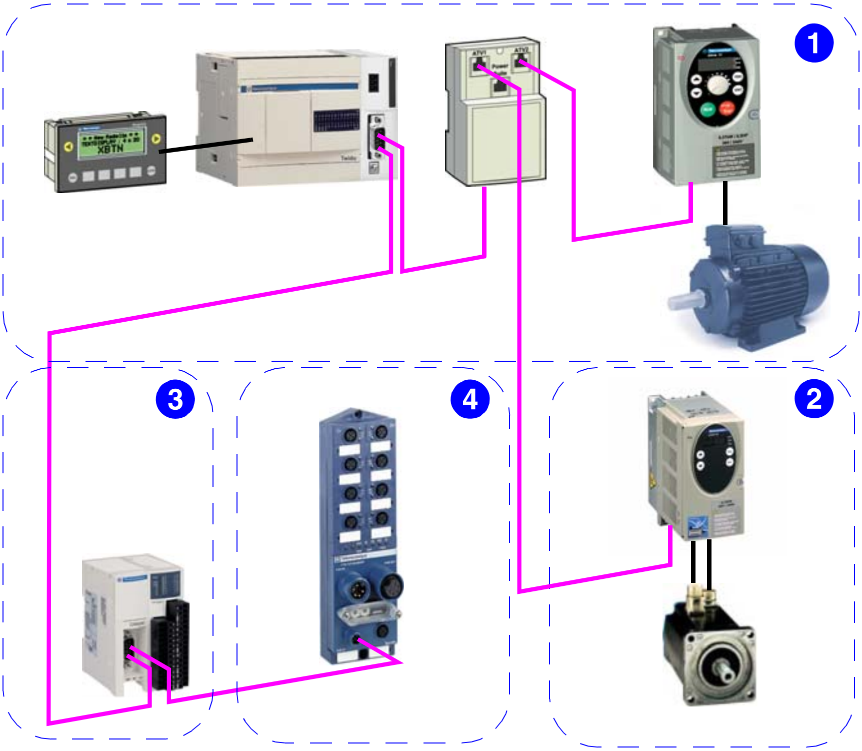
To control the system and give different kinds of information, we shall next add an Advantys OTB IP20 I/O module to the system as well as an Advantys FTB IP67 I/O splitter.

The automation solution includes the following elements:

Element	Description
Twido Controller	The Twido controller associated with the CANopen master supports the following: ● management and communication of the CANopen slaves: the ATV31 variable speed controller, the Lexium 05 servo drive, and the I/S modules, ● diagnostics for the ATV31 variable speed controller, the Lexium 05 servo drive, and the I/S modules, ● data management for the XBTN400 operator display.
HMI	The XBTN 400 is used for the following: ● to display the system parameters: motor speed, information on the I/O, system diagnostics, etc. ● to change the system parameters: motor speed.
Variable controller	The ATV31 variable speed controller is used to control its asynchronous motor according to the commands received.
Servo drive	The Lexium 05 servo drive is used to control the speed, position, or voltage of the BSH servo motor:
I/O module and splitter box	The I/O module and splitter box provide: ● information from the sensors to the controller, ● the transmission of commands to the actuators,

Diagram of the Principal

Here is the general diagram of the automation solution presented:



The present guide is divided into several sections depending on the modifications in the automation solution:

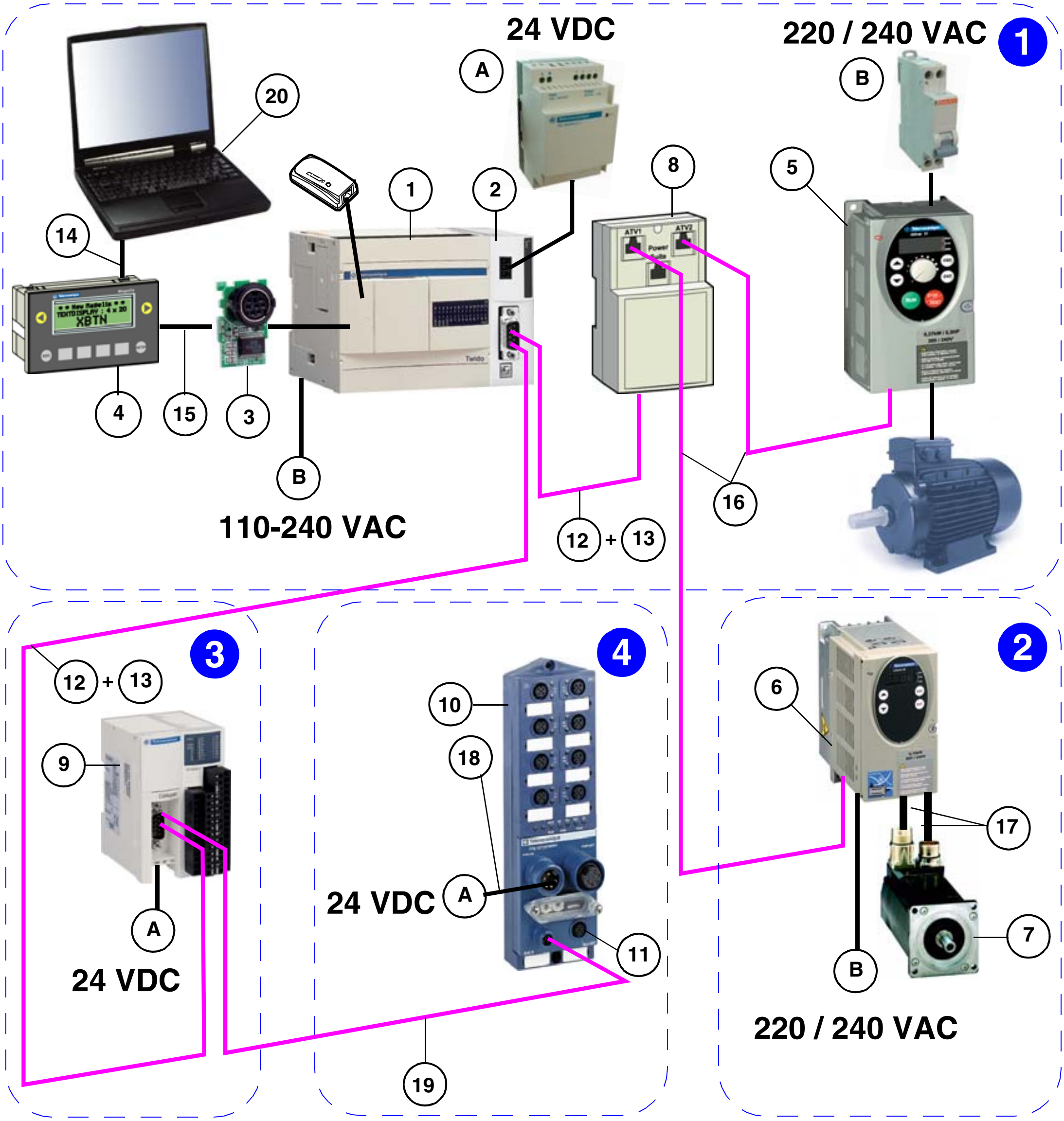
Steps	Description
1	Main system
2	Adding a servo drive: Lexium 05 and a BSH servo motor
3	Addition of a remote IP20 I/O module: Advantys OTB
4	Addition of a remote IP67 I/O splitter: Advantys FTB

Note: If an application requires more products, check the corresponding user guides to see if they are compatible (for example: maximum number of elements managed by the Twido controller).

Description

Installation diagram

The diagram below shows the automation solution:



Required hardware

Products

List of products:

TYPE	Reference	Description	Quantity	No.
Controller	TWDLCAA24DRF	Compact Twido	1	1
Controller	TWDNCO1M	Twido CANopen master	1	2
Controller	TWDNAC485D	Optional mini DIN RS485	1	3
HMI	XBTN400	Compact display, 4 line, 20 character	1	4
Variable controller	ATV31H018Mxx	0.18 kW/0.5 HP one-phase 200-240 V ATV31	1	5
Servo drive	LXM05AD10M2	0.75 kW/0.5 HP one-phase 200-240 V Lexium 05	1	6
Servo motor	BSH0551T11A2A	BSH 05 6000 rpm 1.4 Nm	1	7
Accessories	VW3CANTAP2	Derivation housing (TAP)	1	8
I/O	OTB1C0DM9LP	Advantys OTB CANopen 12E 8S remote IP 20 I/O module	1	9
I/O	TWDDDO8TT	8 static output Analog Expansion Module for OTB	1	9
I/O	TWDDRA8RT	8 relay outputs for OTB	1	9
I/O	FTB1CN08E08SP0	Advantys FTB CANopen 08E 08S remote IP67 I/O splitter	1	10
I/O	FTXCNTL12	CANopen M12 line end	1	11

Power supply and protection

List of recommended power supplies and protections

Reference	Description	Quantity	Num.
ABL8MEM24012	Supply 24VDC	1	A
MULTI 9 -C10	20725 power breaker	1	B

Note: Calibrate the power supply based on the application.

Cables

List of cables:

TYPE	Reference	Description	Quantity	No.
CANopen	TSXCANCA50	CANopen cable - bare wire 50 m	1	12
CANopen	TSXCANKCDF90T	CANopen SUB-D 9 female connector with line end terminal	2	13
HMI	XBTZ945	Configuration cable (PC - HMI)	1	14
HMI	TSXCUSB485	USB configuration cable (PC - HMI)	1	14
HMI	XBTZ9780	Link cable (HMI - Twido controller)	1	15
Variable controller	VW3CANCARR03	1 m CAN RJ45 cable for ATV31 and Lexium 05, 0.3 m	1	16
Variable controller	VW3CANCARR1	1 m CAN RJ45 cable for ATV31 and Lexium 05, 1 m	1	16
Servo drive	VW3M5101R50	Lexium 05 / BSH 05 servo drive power cable, 5 m	1	17
Servo drive	VW3M8101R50	Lexium 05 / BSH 05 servo drive encoder return cable, 5 m	1	17
I/O	FTXDP2115	1 m Advantys FTB power cable	1	18
I/O	FTXCN3230	3 m M12 - free wire Advantys FTB cable	1	19
I/O	FTXCN3250	5 m M12 - free wire Advantys FTB cable	1	19

Software

List of programming and configuration software:

TYPE	Reference	Description	No.
Controller	TWDSPU1004V10M	TwidoSuite ≥ V1.0 including a BlueTooth link*	20
HMI	XBTL1001M	XBTL1000 light ≥ V4.4	
I/O	FTXES00 (≥ V3.1)	Advantys Configuration Tool ≥ V1.4	

* : If the PC is not Bluetooth equipped, obtain the USB key adaptor reference VW3A8115.

Configuration

The present guide describes installation and implementation of the system, focusing on the main elements:

- A Twido TWDLCAA24DRF controller,
- An XBTN400 display,
- An ATV31H018Mxx variable speed controller,
- An LXM05AD10M2 Servo drive,
- A BSH0551T11A2A servo motor,
- An Advantys OTB1C0DM9LP I/O module,
- An Advantys FTB1CN08E08SP0 I/O splitter.

Application projects

This Quick start guide is divided into several sections. The sections describing the installation of the main system, then the integration of other products.

Application projects for the controller are also provided with this guide. Each application project corresponds to a configuration described in a section of this guide.

The table below shows, for each section, the initial application project and the project resulting from the actions described in the section:

Sections in this guide	Initial project	Description	Resulting project
II	-	Main system	Section II
III	Section II	Adding a Lexium 05 and a BSH05	Section III
IV	Section III	Adding an OTB	Section IV
V	Section IV	Adding an FTB	Section V

Main System

II

At a Glance

Overview

This section of the document presents the installation of the main system. In this chapter, references to the Lexium 05 servo drive and the BSH servo motor do not apply unless you want to add a Lexium 05 servo drive (see *Add a Lexium 05 servo drive, p. 51*).

Objective

The objective is to control an ATV31 variable speed controller via the CANopen bus using a Twido CANopen master. The XTBN allows to display / change the speed instruction of the controller and the I/O status information.

Application files

On the CD-ROM BUNDLE (DIA3CD3050101F), there are files for this configuration in the "Applicative files\Partie_II" folder for the Twido controller application and in the "Applicative files\XBTN_XBTR" folder for the XBTN operator display application.

What's in this Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
2	Hardware implementation of the main system	23
3	Software implementation of the main system	35
4	Presentation of the application	45

Hardware implementation of the main system

2

At a Glance

Overview

This chapter describes hardware implementation of the main system of the automation solution.

What's in this Chapter?

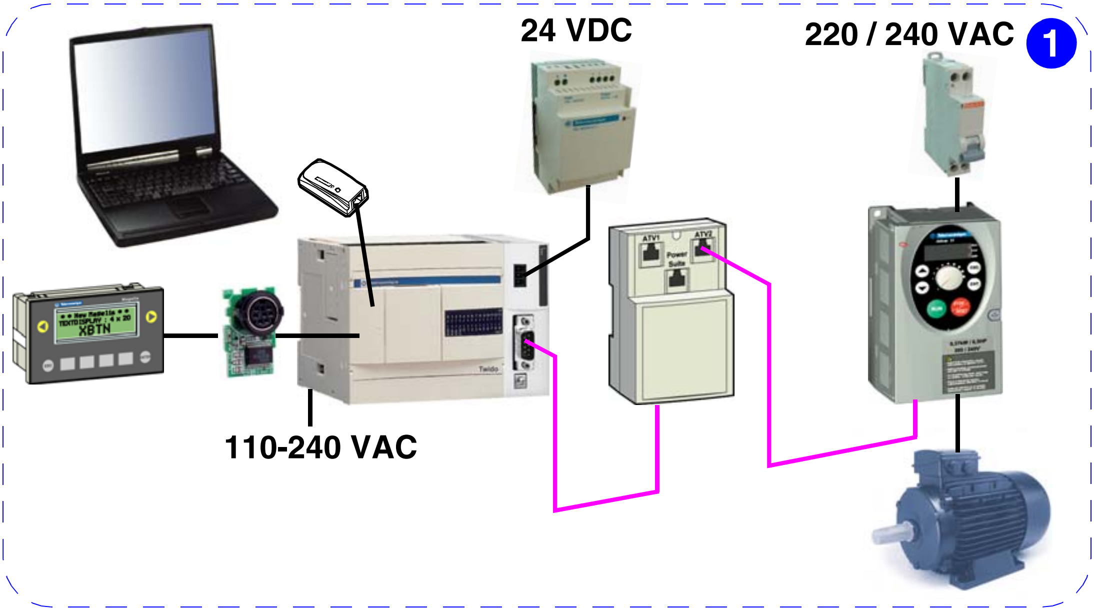
This chapter contains the following topics:

Topic	Page
Main cabling	24
HMI cabling	25
CANopen network cabling	26
I/O cabling	29
Power supply	31

Main cabling

Installation diagram

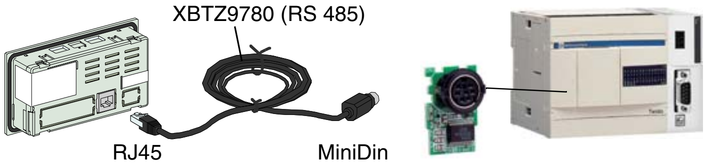
This part of the document covers the following mounting:



HMI cabling

XBTN-Twido cabling

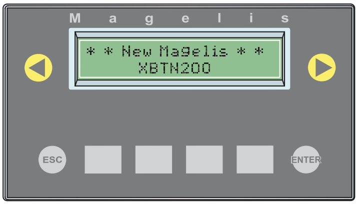
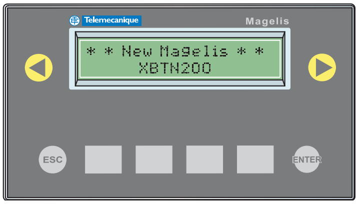
Connect the XBTN to the Twido controller using the XBTZ9780 cable:



Note: Use the additional TWDNAC485D interface mounted on the Twido controller. Twido port 1 is reserved for communications between the Twido and the PC.

According to the protocol used, the display of "?????" instead of values or the persistence of the connection popup indicates a communication problem. This can be caused by the cable used.

The table below shows which cable to use according to the version of the XBTN (can be determined by the front panel) and the version of the XBT L1000 software:

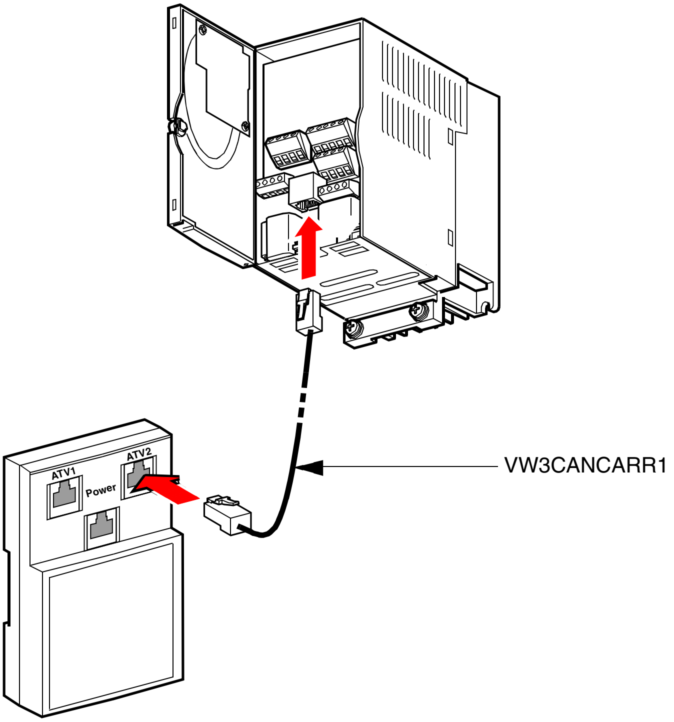
XBT Front panel	Cable
 M a g e l i s * * New Magelis * * XBTN200 ESC ENTER	Version of XBT L1000 ≤ V4.30 + Cable XBT Z978
	Version of XBT L1000 ≥ V4.40 + Cable XBT Z978 + XBT ZN999 Adapter
 Telemecanique Magelis * * New Magelis * * XBTN200 ESC ENTER	Version of XBT L1000 ≥ V4.40 + Cable XBT Z9780

For further information, see the Magelis XBTN / XBTR Operation Guide reference 1681028.

CANopen network cabling

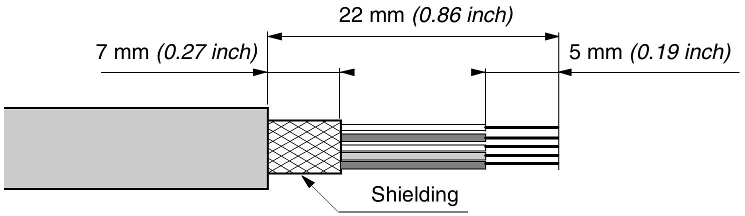
ATV31-TAP Cabling

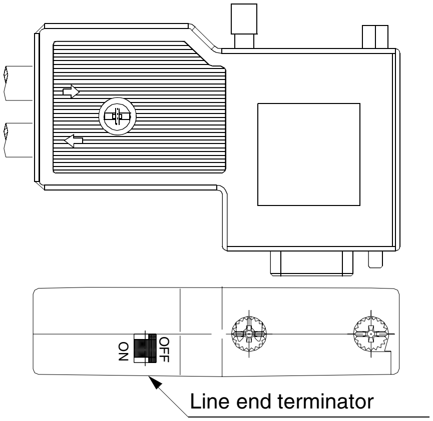
To connect the ATV31 to the TAP, use the type RJ45 - RJ45 cable so that:

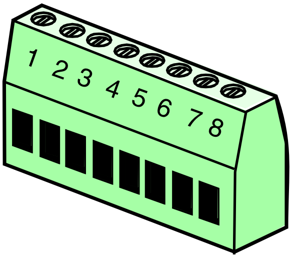


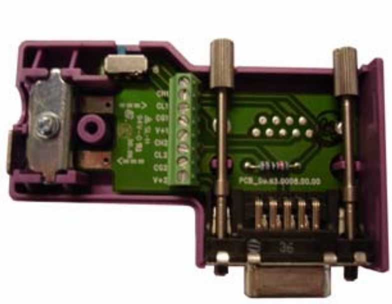
TAP-Twido Cable Preparation

To connect the TAP to the Twido controller, connect a SUB-D 9 TSXCANKCDF90T connector with the TSXCANCA50 cable in the following manner:









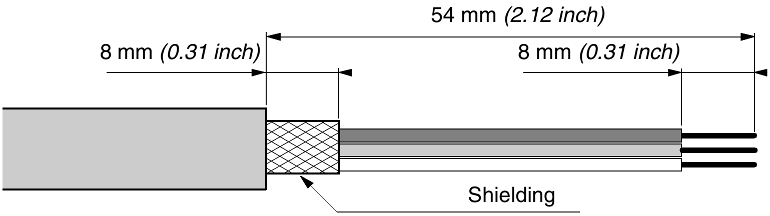
Cable pin assignment in the terminal of the SUB-D 9 connector:

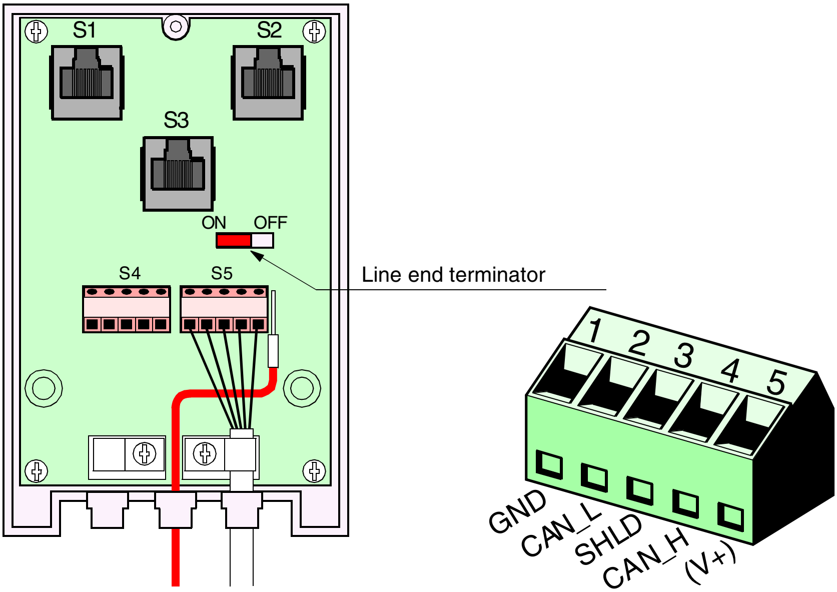
No.	Signal	Cable	Connector	Wire color
1	CAN_H	TAP/Twido	CH1	White
2	CAN_L	TAP/Twido	CL1	Blue
3	GND	TAP/Twido	CG1	Black
4	V+	TAP/Twido	V+1	Red

Note: Toggle the line end terminator of the connector to "ON".

TAP-Twido cabling

The SUB-D 9 end of the previously prepared cable is connected to the Twido controller. The "bare wire" end of the cable is connected in the TAP in the following manner:





Cable pin assignment in the TAP terminal:

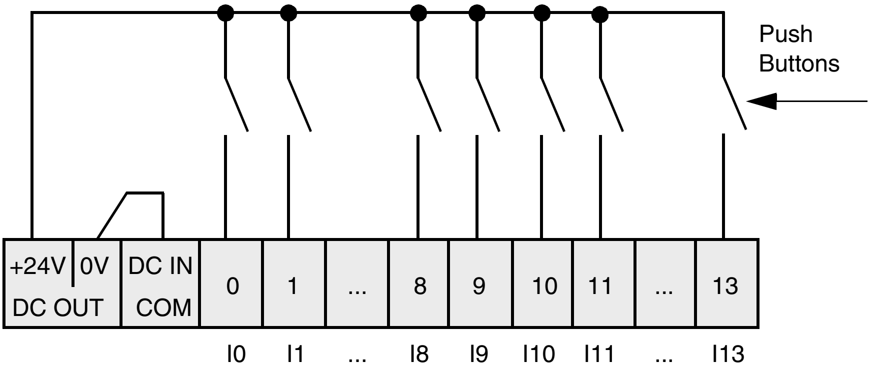
No.	Signal	Wire color	Description
1	GND	Black	Weight
2	CAN_L	Blue	CAN_L polarity
3	SHLD (CAN-GND)	(Bare cable shielding)	Optional shielding
4	CAN_H	White	CAN_H polarity
5	V+	Red	Optional power supply

Note: Toggle the line end terminator of the TAP to "On".

I/O cabling

Twido inputs cabling

Inputs cabling diagram:

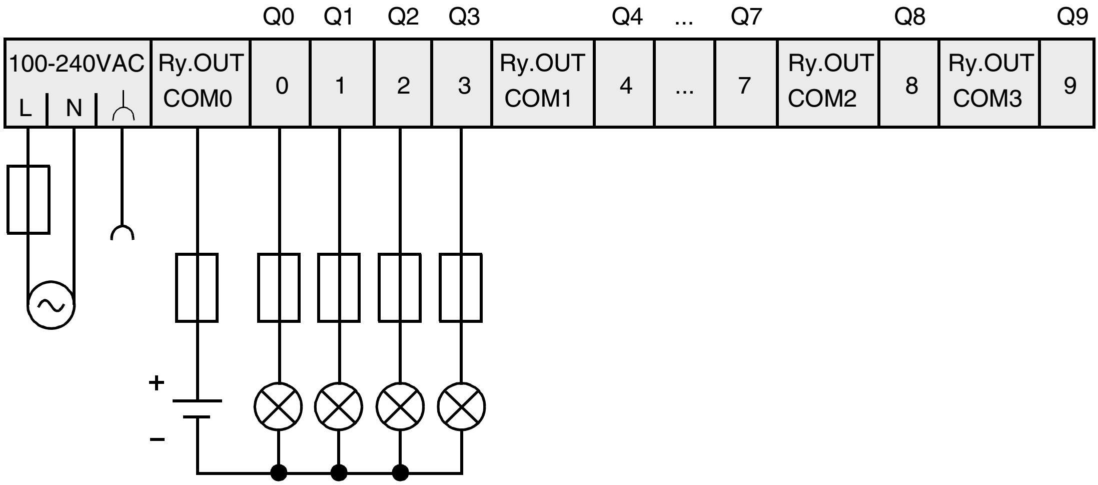


Additional information on inputs used in the application:

Input	Symbol used in the Twido application	Action
%I0.0	LXM POWER	**Lexium 05:** Switch the Lexium to the operational state and activate the BSH servo motor
%I0.1	START_POSITION_LXM	**Lexium 05:** Start the BSH servo motor rotation
%I0.8	PB_START_FORWARD	**ATV31:** Start forward motor rotation
%I0.9	PB_START_REVERSE	**ATV31:** Start reverse motor rotation
%I0.10	PB_STOP	**ATV31 and Lexium 05:** Stop the motor and the BSH servo motor
%I0.11	PB_SLOW_FAST	**ATV31:** Set speed fast or slow: ● Input = 0 for fast ● Input = 1 for slow
%I0.13	RESET_ERROR	**ATV31 and Lexium 05:** Acknowledge an error (the cause of the error must be eliminated):

Twido outputs cabling

Outputs cabling diagram:

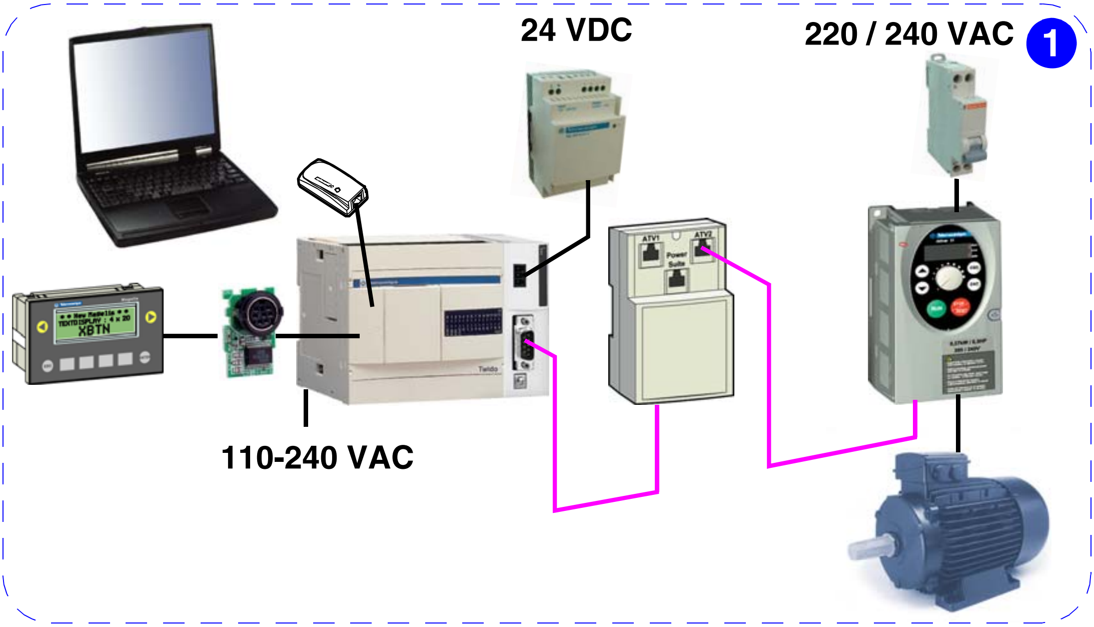


Additional information on outputs:

Output	Symbol used in the Twido application	Indication displayed
%Q0.0	SD_RUN_FORW	**ATV31:** Forward motor rotation
%Q0.1	SD_RUN_REV	**ATV31:** Reverse motor rotation
%Q0.2	SD_STOPPED	**ATV31:** Motor stopped
%Q0.3	MOTOR_IS_GOING_TO_POINT	**Lexium 05:** BSH servo motor in rotation

Power supply

Diagram



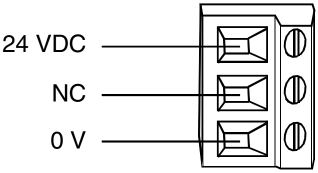
Power on the Twido

To power on the Twido controller, follow the recommendations provided in the "Hardware implementation guide."

Power on the CAN Master

To power on the CANopen master, follow the recommendations in the service instructions provided with the product.

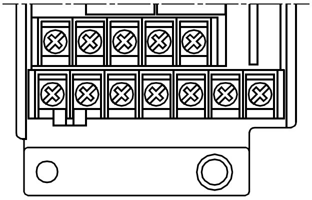
Cabling Diagram:



Power on the ATV31

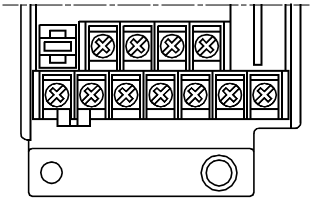
Connections:

ATV 31H018M3X , H037M3X, H055M3X, H075M3X



⏚	⏚	R/L1	S/L2	T/L3		
P0	PA/+	PB	PC/-	U/T1	V/T2	W/T3

ATV 31H018M2 , H037M2, H055M2



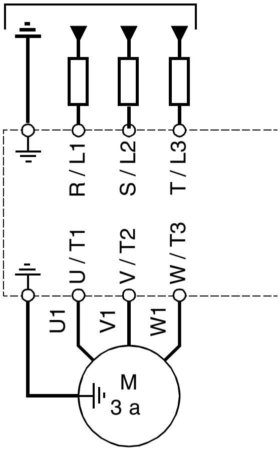
⏚	⏚	R/L1	S/L2			
P0	PA/+	PB	PC/-	U/T1	V/T2	W/T3

Note: Remove the self-adhesive labels to access the connectors underneath.

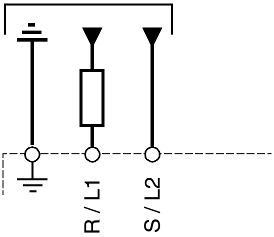
Note: R/L1 = Phase, R/L2 = Neutral.

Connection diagrams:

31xxxxM3X/N4/S6X Three-phase Network



ATV 31xxxxM2 One-phase Network



⚠ DANGER

RISK OF ELECTROCUTION.

To power on the ATV31 variable speed controller, follow the recommendations provided in the installation guide ref. VVDED303043.

Failure to follow these instructions will result in death or serious injury.

Software implementation of the main system

3

At a Glance

Overview

This chapter describes software implementation of the main system of the automation solution.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Installing software and loading applications	36
ATV31 communications configuration	40

Installing software and loading applications

Applications

The CD-ROM BUNDLE (DIA3CD3050101F) contains the applications files corresponding to the proposed automation solution. Loading these applications will allow a simple startup for the automation solution.

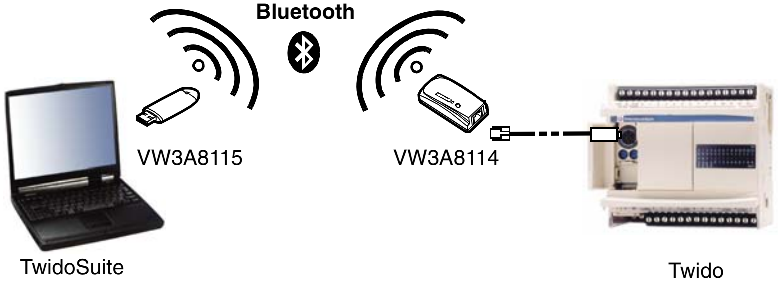
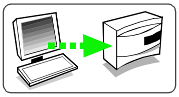
Software Installation

It is necessary, at the start, to install the following software:

- An XBT-L1000 for the XBTN operator display
- TwidoSuite for the Twido controller

Twido Application

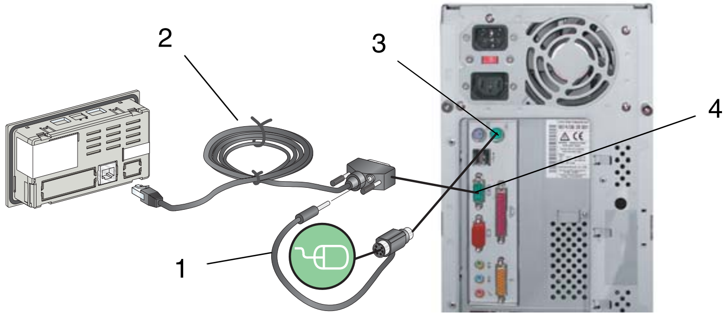
The following method allows to load the application provided for the Twido controller:

Step	Action
1	Connect the Twido controller to a PC in the following manner:  The controller must be powered on. Configure the Bluetooth key by following the guide provided with the key. Use port 1 of the Twido controller. The XBTN is connected to the RS485 additional interface of the Twido controller. **Advice:** Use the COM 4 port, which is the default port installed by the Bluetooth key.
2	From the CD-ROM BUNDLE (DIA3CD3050101F), **COPY** the folder "Applicative files\Partie II"
3	**PASTE** this folder into the directory "C:\Program Files\Schneider Electric\TwidoSuite\My projects"
4	Open the project "BUNDLE_CAN_Part_II.xpr" using the TwidoSuite software.
5	In the TwidoSuite main window, select the task **Program** → **Debug** → **Connector** and click **OK**. TwidoSuite tries to set up a connection with the controller and runs synchronization checks between the computer's applications and the controller. Once the connection has been established, select **Transfer PC => Controller**.  Click **OK**.
6	Wait until the program is finished loading. In the TwidoSuite main window, select the taskProgram → **Debut** → **Disconnect** and click **OK**.

Note: If the Bluetooth communication link is lost, disconnect the reconnect the Bluetooth VW3A8114 gateway (Twido side) to reset the communication.

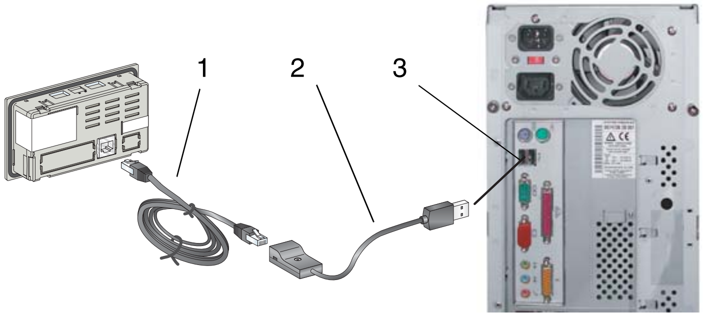
XBTN

XBTN / PC connection using a XBTZ945 cable:



No.	Description
1	XBTZ945 2/2 RJ45/MiniDIN cable
2	XBTZ945 1/2 RJ45/SUB-D 9 cable
3	Mouse Port MiniDIN
4	COM port

XBTN / PC connection using a TSXCUSV485 cable:



No.	Description
1	Cable XBTZ925(A) RJ45/RJ45
2	Adapter TSXCUSB485 RJ45/USB
3	USB port

For further information, see the Magelis XBTN / XBTR Operation Guide reference 1681028.

The following method allows to load the program provided for the XBTN:

Step	Action
1	From the CD-ROM BUNDLE (DIA3CD3050101F), **COPY** the file "XBTN_XBTR\QS_XBT_N400.DOP"
2	**PASTE** this file into the directory "C:\Program Files\Schneider Electric\XBT-L1000\appli"
3	Open the file using the XBT-L1000 software.
4	Connect the XBTN operator display to the PC (see above diagram). The XBTN operator display must be powered on (for more information, see the operation guide of the operator display. The XBTN displays "WAITING FOR TRANSFER".
5	From the XBT-L1000 software menu bar, select "Transfer/Export".
6	Wait until the program is finished loading to disconnect the XTBN from the PC.

ATV31 communications configuration

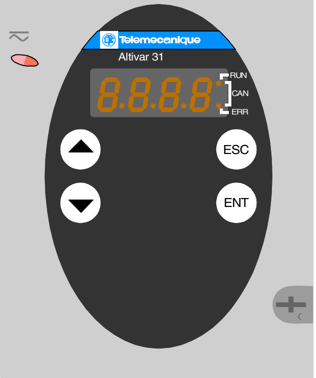
Principle

The system products must be configured according to the Twido application controller as follows:

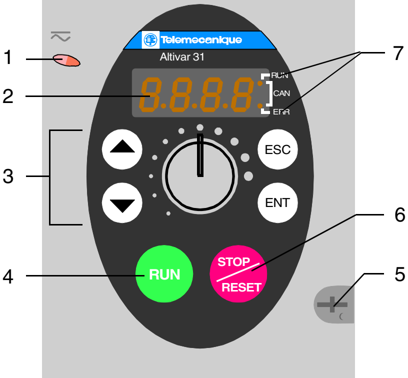
- Address 1: ATV31
- Transmission Speed: 125 kBits/s

Description

Front panel of the ATV31 variable speed controller:



ATV31xxxxx



ATV31xxxxxA

No.	Description
1	The red LED on indicates that the direct bus is powered on.
2	4-digit "7 segments" operations display.
3	Central programming terminal.
4	"RUN" key, for starting motor in forward mode.
5	To lock/unlock the front panel of the speed controller, a flat or cross-slot screwdriver is needed.
6	"STOP/RESET" key, for stopping the motor and resetting the current defaults.
7	These two LEDs indicate the status of communications ("RUN") and the presence of a possible fault ("ERR") on the CANopen bus.

Method

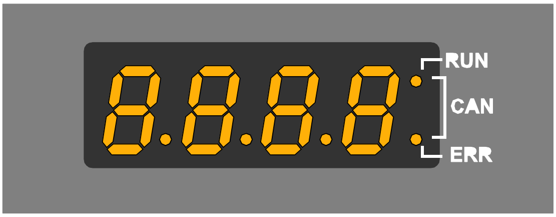
Configuration of ATV31communication parameters:

Step	Action
1	Press on the "ENT" key to enter the ATV31 configuration menu.
2	Use the "Arrows" keys to select the "COM" Communication menu then confirm using the "ENT" key.
3	Use the "Arrows" keys to select the "AdCO" menu then confirm using the "ENT" key. Enter the value "1" (Address on the CANopen bus). Confirm using the "ENT" key then exit the menu using the "ESC" key.
4	Use the "Arrows" keys to select the "bdCO" menu then confirm using the "ENT" key. Enter the value "125" (Speed on the CANopen bus). Confirm using the "ENT" key then exit the menu using the "ESC" key.
5	Press several times on the "ESC" key to exit the configuration menu.

Note: The configuration may be modified only when the motor is stopped and when the variable speed controller is locked (cover closed). Any modification entered will become effective after an "Off/On" cycle of the speed controller.
For more information, see the reference document VVDED303042.

Indicators

The two indicator LEDs, located on the right side of the 4-digit 7 segment display on the front panel of the Altivar 31, indicate the status of CANopen communications::

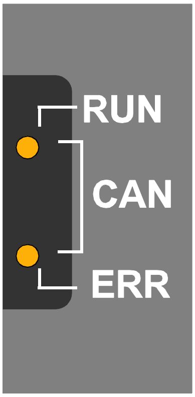


Normal display and not in installation:

- 43.0: Display of the parameter selected in the SUP menu (by default: motor frequency). In the event of current limitation, the display flashes.
- Init: Initialization sequence.
- rdY: Speed controller ready.
- dcb: Braking by injection of DC current in progress.
- nSt: Free wheel stop.
- FSt: Rapid stop.
- tUn: Self adjustment in progress.

Note: In the event of a fault the display will flash.

Description of the different states of the Altivar 31 / CANopen:

	LED indicator	LED Status	Altivar 31 status
	RUN	●	The CANopen controller is "OFF"
		◐	The Altivar 31 is "STOPPED"
		✹	The Altivar 31 is "PRE-OPERATIONAL"
		🔴	**The Altivar 31 is "OPERATIONAL"**
	ERR	🔴	**No error detected**
		◐	Alarm sent by the CANopen controller of the Altivar 31 (eg: too much error frame)
		◐ ◐	Error due to the appearance of a "Node-guarding" or "Heartbeat" event
		●	The CANopen controller is "bus off"

Description of different LED states:

LED Status	Visual description of the status of the LED
●	The indicator is **off**.
◐	The indicator is in **SINGLE FLASHING**. (On for 200 ms and off for 1 second.)
◐ ◐	The indicator is in **DOUBLE FLASHING**. (On for 200 ms, off for 200 ms, on for 200 ms and off for 1 second.)
✹	The indicator **FLASHES** at 2.5 Hz. (On for 200 ms and off for 200 ms.)
🔴	The indicator is **on**.

Special Note

In the event of use WITHOUT MOTOR (simulation mode), the ATV31 variable speed controller will display an error message: "OPF". Change the configuration in the "FLt / OPL" submenu and send the value "YES" to "OAC".

Communication Parameters

Use the "COM" communication menu to access configuration of CANopen communication functions of the ATV31 connected to a Twido controller:

Parameter	Possible Values	Display on terminal	Values to input for the application
CANopen AdC0 address	0 to 15	1 to 16	1
CANopen bdC0 speed	125 kBits/s	125.0	125 kBits/s
	250 kBits/s	250.0	
	500 kBits/s	500.0	

Presentation of the application

4

At a Glance

Overview

This chapter describes the operation of an automation solution.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Operating mode	46
HMI	47
Grafcet of the ATV31 application	49

Operating mode

Commands

When installation of the main system is complete, the system may be controlled using four push buttons and one two-position button.

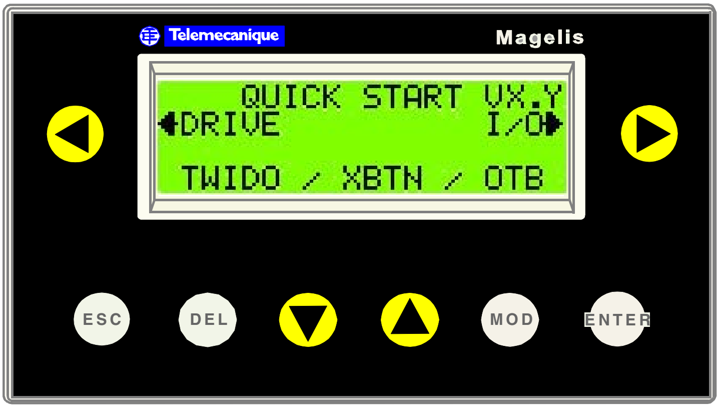
Input Corresponding symbol	Function	Description	Output Corresponding symbol
%I0.8 PB_START_FORWARD	Motor forward start	The corresponding LED turns on when the selected speed is reached.	%Q0.0 SD_RUN_FORW
%I0.9 PB_START_REVERSE	Motor reverse start	The corresponding LED turns on when the selected speed is reached.	%Q0.1 SD_RUN_REV
%I0.10 PB_STOP	Stop motor	The corresponding LED turns on when the motor stops.	%Q0.2 SD_STOPPED
%I0.11 PB_SLOW_FAST	Predefined motor speeds	Apply a predefined speed when changing position: ● %I0.11 is set to 0: 3,300 rpm ● %I0.11 is set to 1: 88 rpm	No output to match input
%I0.13 RESET_ERROR	Acknowledge an ATV31 error	Reset an error (the cause of the error must be eliminated):	No output to match input

Note: In our sample application, the push button for %I0.10 must be pushed (Stop motor) in order to change the direction of motor rotation.

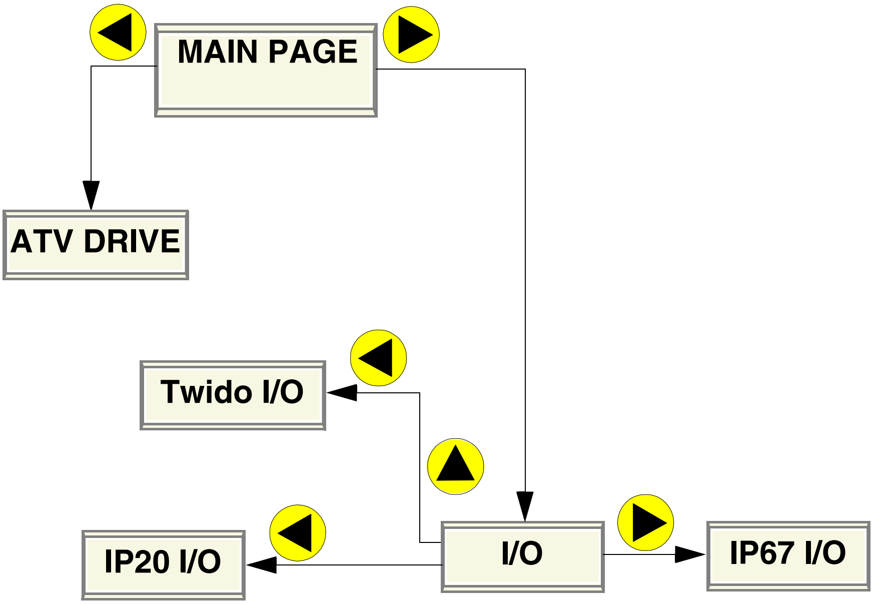
Note: In the sample program, the speed configured manually with the XBTN is not memorized after a motor stop. The speed of 3,300 rpm depends on the configuration of the variable speed controller.

HMI

Main page

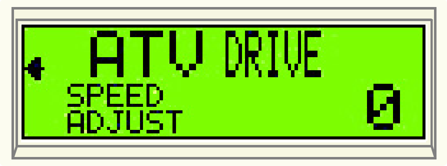


Organigram of pages



Note: Press on the ESC key to return to the preceding page.

"ATV DRIVE" page



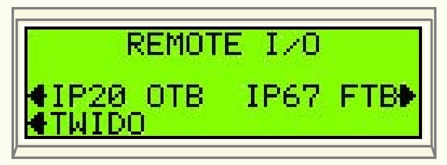
◀ ATV DRIVE
SPEED
ADJUST 0

The "ATV DRIVE" page displays the speed instruction sent to the ATV31 by the Twido controller.

To modify speed, do the following:

Step	Action
1	Press on the MOD key.
2	Use the ▲ and ▼ arrows to change the speed (immediate write).
3	Press on the ENTER key to confirm.

"I/O" page



REMOTE I/O
◀IP20 OTB IP67 FTB▶
◀TWIDO

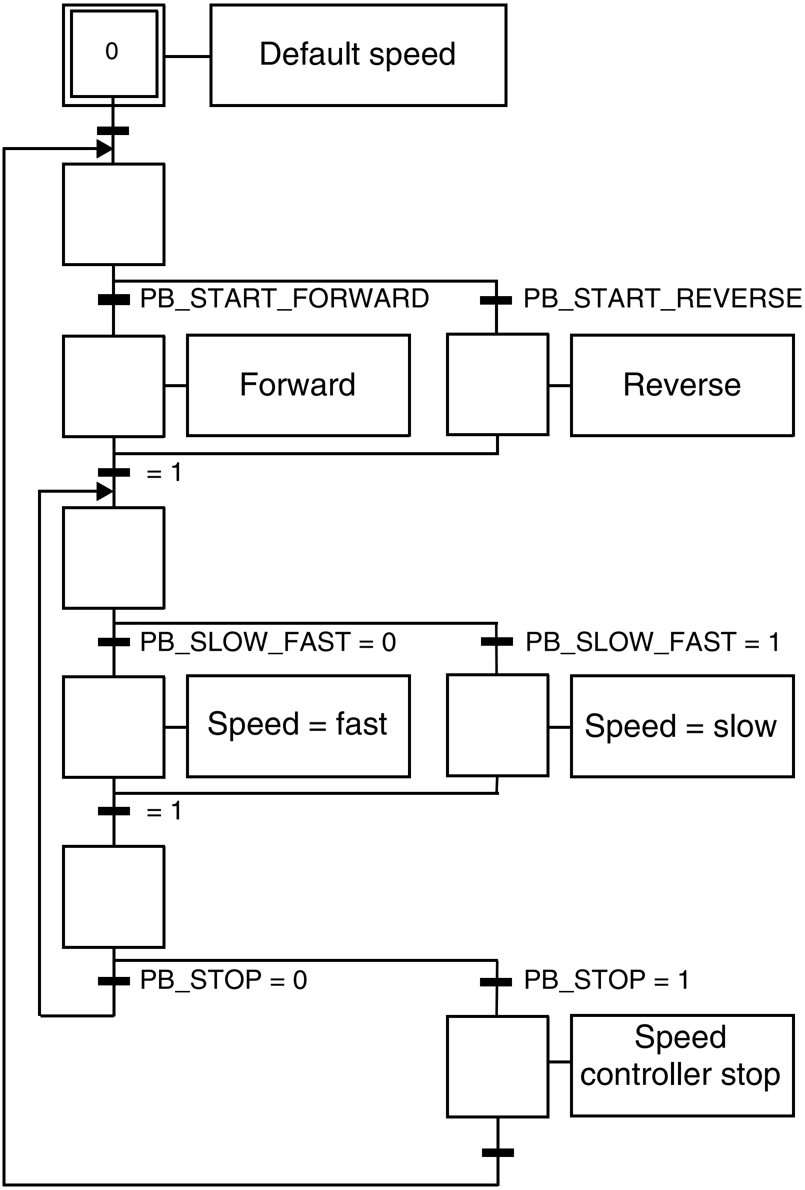
The "I/O" screen provides access to 3 screens that display the status of:

- I/O of the Twido controller,
- I/O of the Advantys OTB I/O module (see *Add an Advantys OTB I/O module, p. 77*),
- I/O of the Advantys FTB I/O splitter (see *Add an Advantys FTB I/O splitter, p. 103*).

Grafcet of the ATV31 application

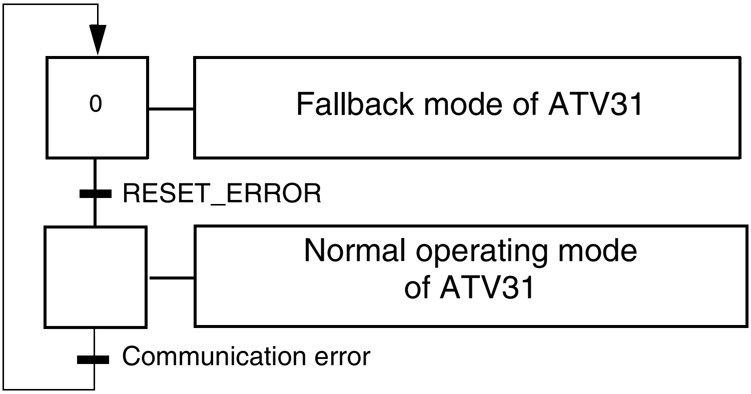
Simplified chart of ATV31 operation

The operation of the ATV31 may be represented in the following manner:



Communication error chart

Communication error chart for the ATV31:



Note: In the event of a communication error, the motor state depends on the ATV31 configuration (fallback modes)

Add a Lexium 05 servo drive

III

At a Glance

Overview

This section of the document presents a possible evolution of the main system, that is the addition of a Lexium 05 servo drive.

Application files

Files for this configuration may be found on the CD-ROM BUNDLE (DIA3CD3050101F) in the "Applicative files\Partie_III" folder for the Twido controller application.

What's in this Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
5	Hardware implementation of a Lexium 05 servo drive	53
6	Software implementation of a Lexium 05 servo drive	59
7	Presentation of the Lexium 05 application	73

Hardware implementation of a Lexium 05 servo drive

5

At a Glance

Overview

This chapter describes hardware implementation of a Lexium 05 servo drive.

What's in this Chapter?

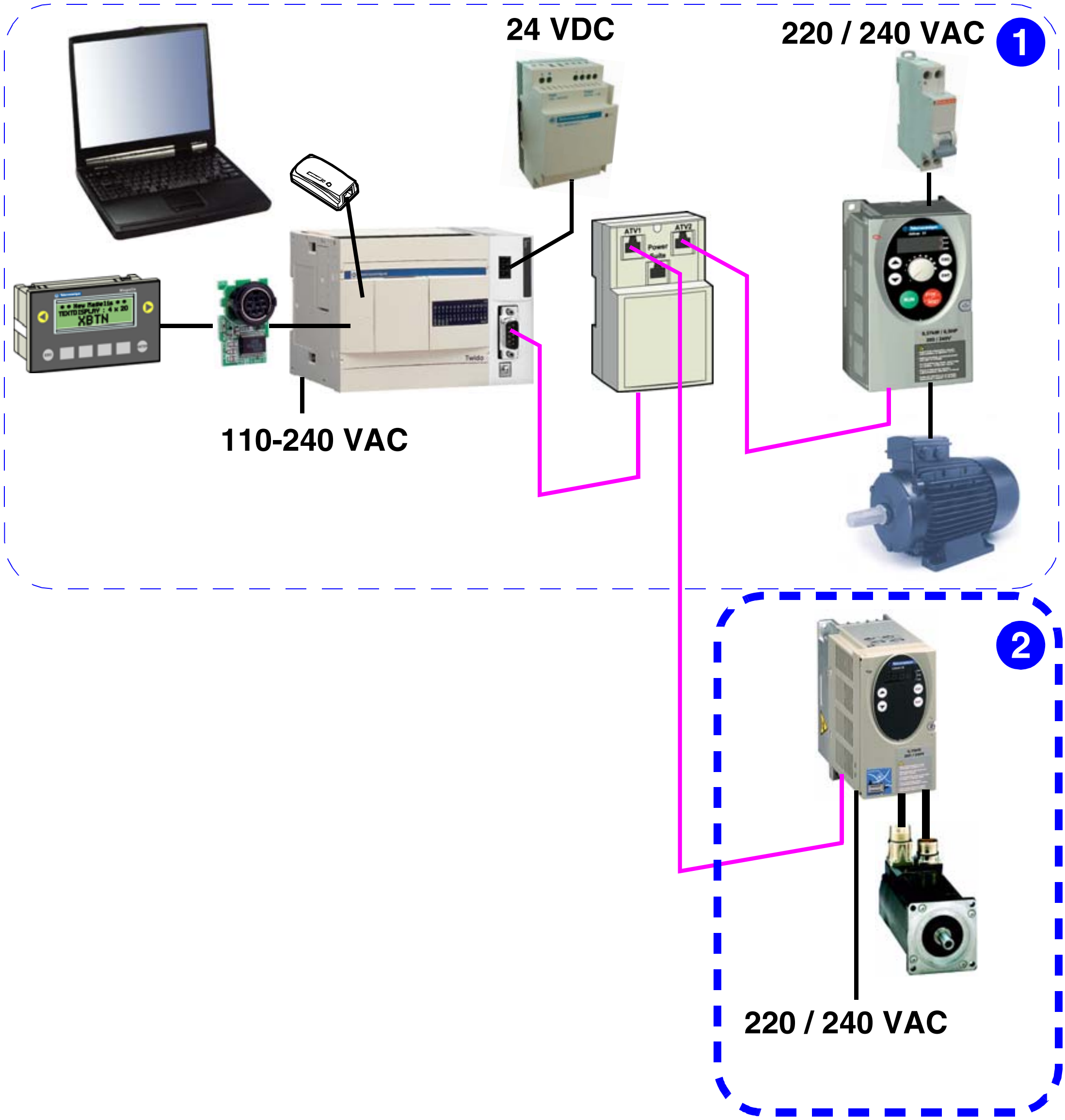
This chapter contains the following topics:

Topic	Page
Wiring for a Lexium 05 servo drive	54
Cabling of the CANopen network	55
Power supply	56

Wiring for a Lexium 05 servo drive

Installation diagram

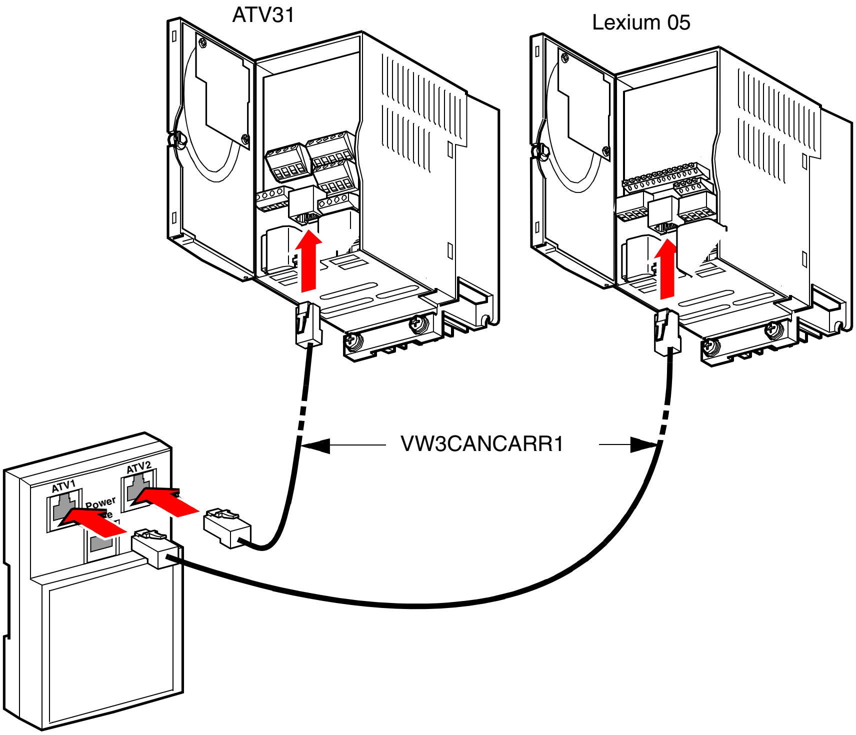
This part of the document covers the following mounting:



Cabling of the CANopen network

Wiring the Lexium 05-TAP

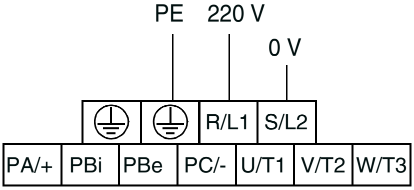
To connect the Lexium 05 to the TAP, use the type RJ45 - RJ45 cable so that:



Power supply

Lexium 05 Power Supply Ratings

Power supply of the LXM 05AD10M2:



⚠ DANGER

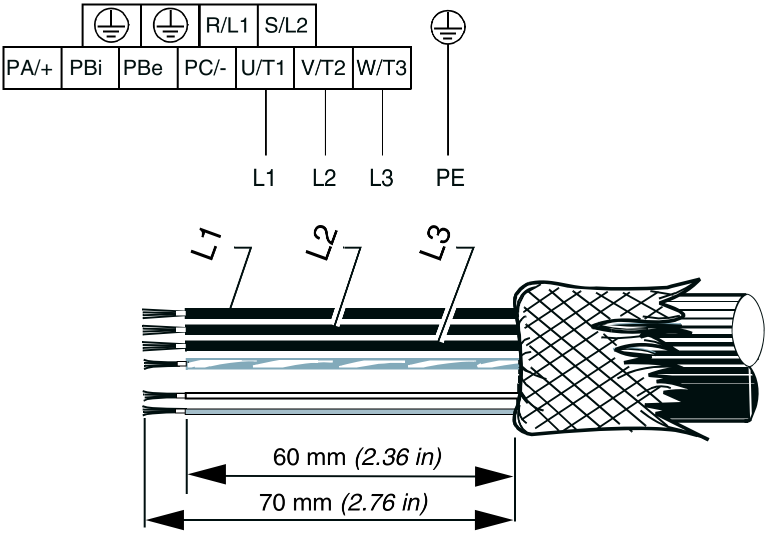
RISK OF ELECTROCUTION.

To power on the Lexium 05 servo drive, follow the recommendations provided in its installation guide reference 0198441113233.

Failure to follow these instructions will result in death or serious injury.

BSH servo motor power supply

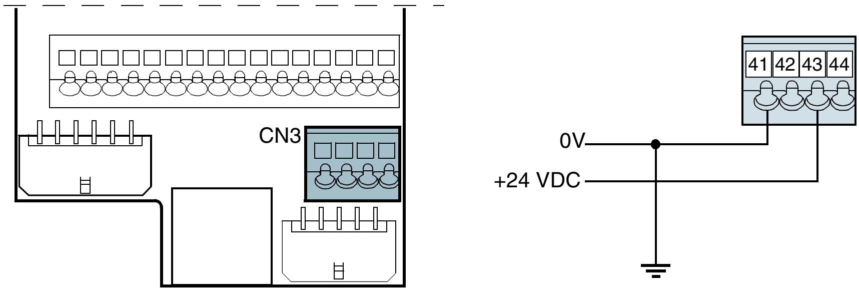
Connect the BSH 0551T11A2A servo motor to the Lexium 05 using cable VW3M5101R50:



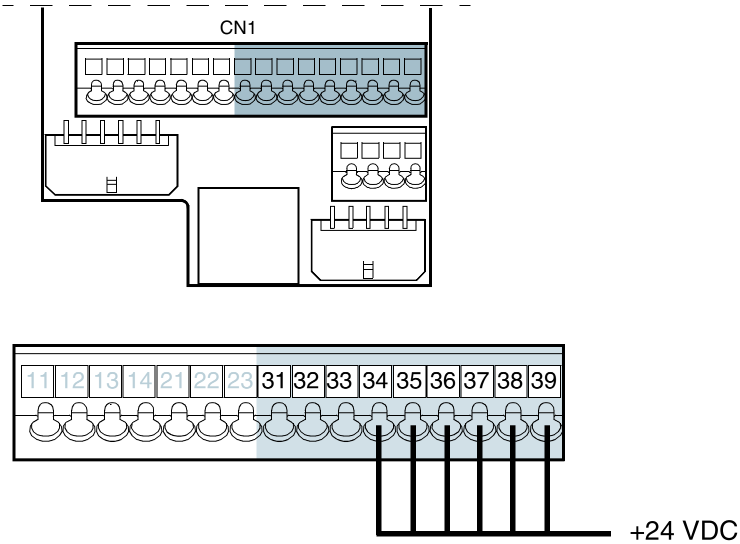
Note: The terms L1, L2 and L3 appear on each of the 3 black wires.

Lexium 05 command power supply

Lexium 05 command power supply wiring diagram

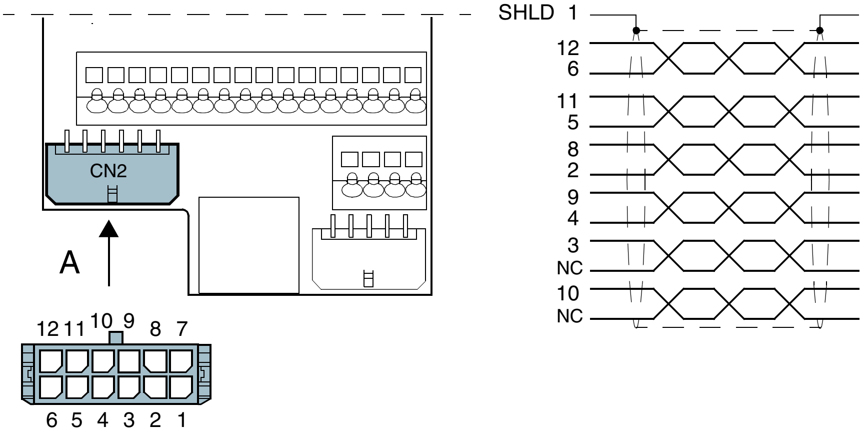


Lexium 05 digital I/Os wiring



Servo motor command

Connect the BSH servo motor to the Lexium 05 with cable VW3M8101R50 as follows:



Pin CN2	Signal	Color	Even	MotorP in	Meaning	I/O
1	SHLD				Shielding tracer	
12	SIN	White	1	8	Sine signal	E
6	REFSIN	Brown	1	4	Reference for sine signal, 2.5V	S
11	COS	Green	2	9	Cosine signal	E
5	REFCOS	Yellow	2	5	Reference for cosine signal, 2.5V	S
8	Data	Grey	3	6	Reception data, data sent	I/O
2	/Data	Pink	3	7	Reception data, data sent, inverted	I/O
10	ENC_0V	Blue	4	11	Encoder reference potential (0.5mm)	S
		Red	4		free (0.5 mm)	
3	T_MOT_0V	Black	5	1	Reference potential towards T_MOT	
		Violet	5		Free	
9	T_MOT	Grey/Pink	6	2	CTP temperature sensor	E
4	ENC+10V_OUT	Red/Blue	6	10	10 VDC power supply foe encoder, 150 mA max.	S
7	n.c.				Free	

Software implementation of a Lexium 05 servo drive

6

At a Glance

Overview

This chapter describes software implementation of a Lexium 05 servo drive.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Lexium 05 communications configuration	60
Add a Lexium 05 in the Twido controller application	62

Lexium 05 communications configuration

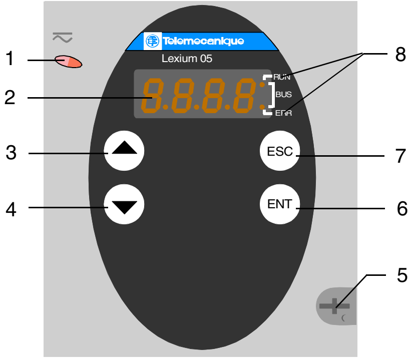
Principle

The system products must be configured according to the Twido application controller as follows:

- Address 1: ATV31
- Address 10: Lexium 05
- Transmission Speed: 125 kBits/s

Description

Lexium 05 servo drive front panel:



No.	Description
1	The red LED on indicates that the direct bus is powered on.
2	4-digit "7 segments" operations display.
3	Up arrow button: ● Return to the menu or the previous parameter ● Increase the value displayed
4	Down arrow button: ● Go to the menu or the next parameter ● Lower the value displayed
5	To lock/unlock the front panel of the speed controller, a flat or cross-slot screwdriver is needed.
6	ENT: ● Call a menu or a parameter ● Store the values displayed in the EEPROM
7	ESC: ● Exit a menu or a parameter ● Return to the last value stored.
8	These two LEDs indicate the status of communications ("RUN") and the presence of a possible fault ("ERR") on the CANopen bus.

Method

Configuration of Lexium 05 communication parameters:

Step	Action
1	Press on the "ENT" key to enter the configuration menu.
2	Use the "Arrows" keys to select the "COM" Communication menu then confirm using the "ENT" key.
3	Use the "Arrows" keys to select the "CoAd" menu then confirm using the "ENT" key. Enter the value "10" (Address on the CANopen bus). Confirm using the "ENT" key then exit the menu using the "ESC" key.
4	Use the "Arrows" keys to select the "Cobd" menu then confirm using the "ENT" key. Enter the value "125" (Speed on the CANopen bus). Confirm using the "ENT" key then exit the menu using the "ESC" key.
5	Press several times on the "ESC" key to exit the configuration menu.

Note: The configuration may be modified only when the motor is stopped and when the servo drive is locked (cover closed). Any modification entered will become effective after an "Off/On" cycle of the servo drive.
For more information, see the reference document 0198441113233.

Communication Parameters

Use the "COM" communication menu to access configuration of the CANopen communication functions of the Lexium 05 connected to a Twido controller:

Parameter	Possible values for the Lexium	Display on the Lexium terminal	Values to input for the application
CANopen CoAd address	1 to 127	1 to 127	10
CANopen Cobd speed	125 kBits/s	125.0	125 kBits/s
	250 kBits/s	250.0	
	500 kBits/s	500.0	

Note: The default network address of the Lexium is 127.
Twido only authorizes addresses from 1 to 16.

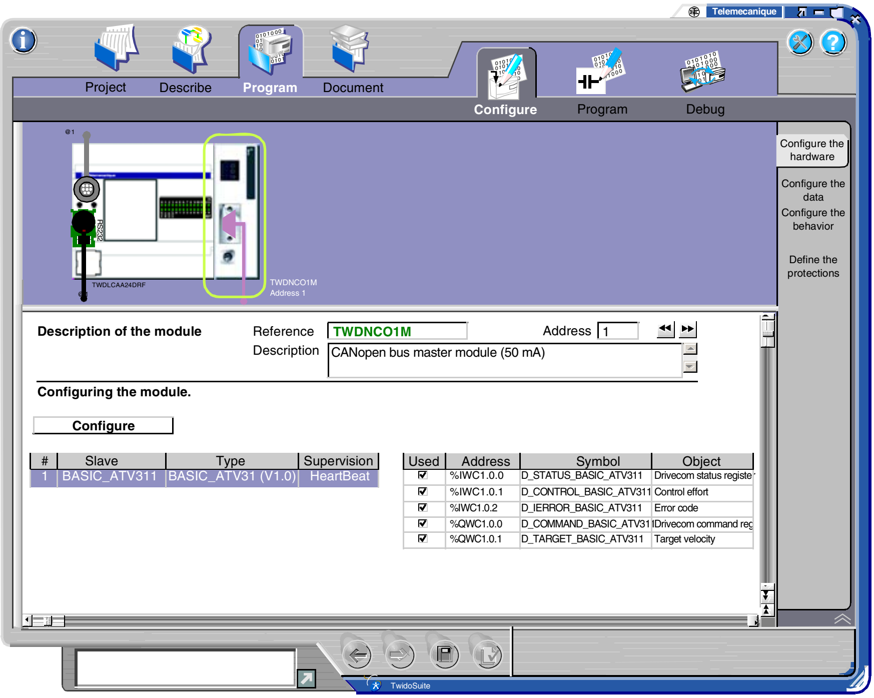
Add a Lexium 05 in the Twido controller application

Principle

The objective is to modify the existing application in order to be able to control the Lexium 05 servo drive.

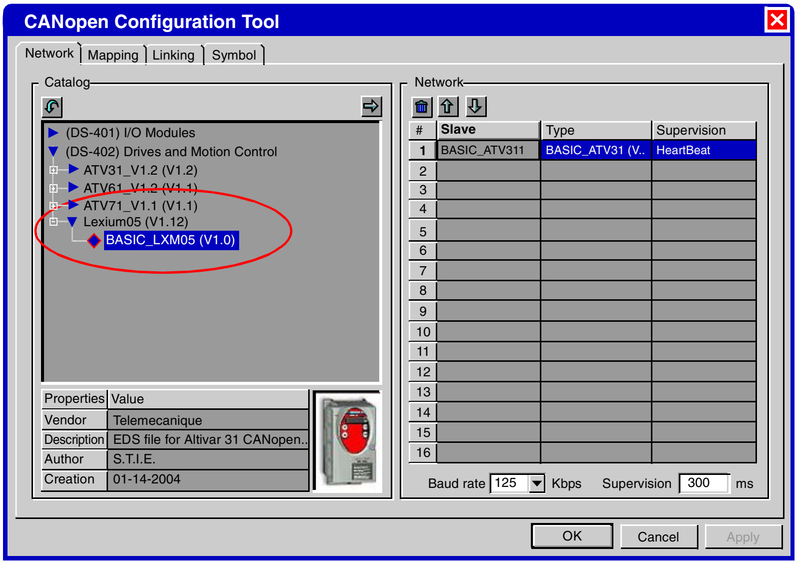
Declare the Lexium 05

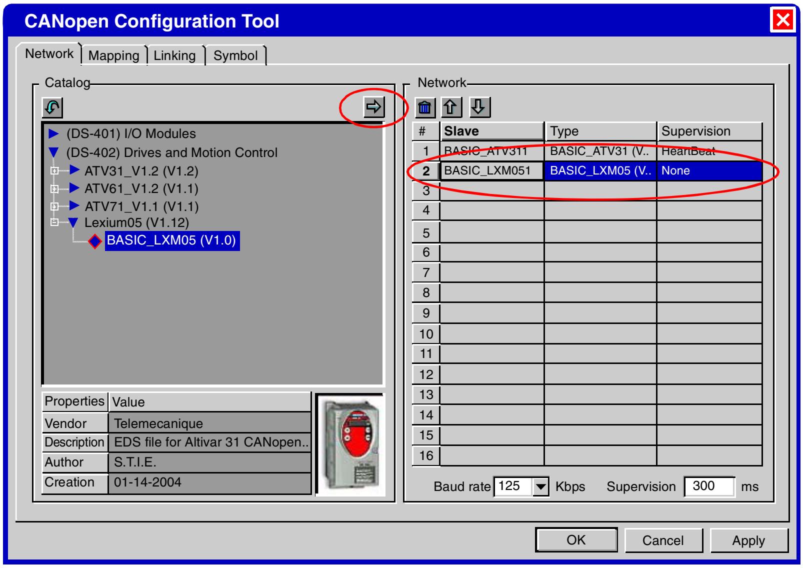
The following method allows declaration of the Lexium 05 in the Twido controller and in the CANopen master.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Open the project "BUNDLE\Partie_II\BUNDLE_CAN_Part_II.xpr" using the TwidoSuite software (TwidoSuite must be in disconnected mode).
In the TwidoSuite main window, select the task <b>Program</b> → <b>Configure</b> → <b>Configure hardware</b>.
Select the CANopen network TWDNCO1M:


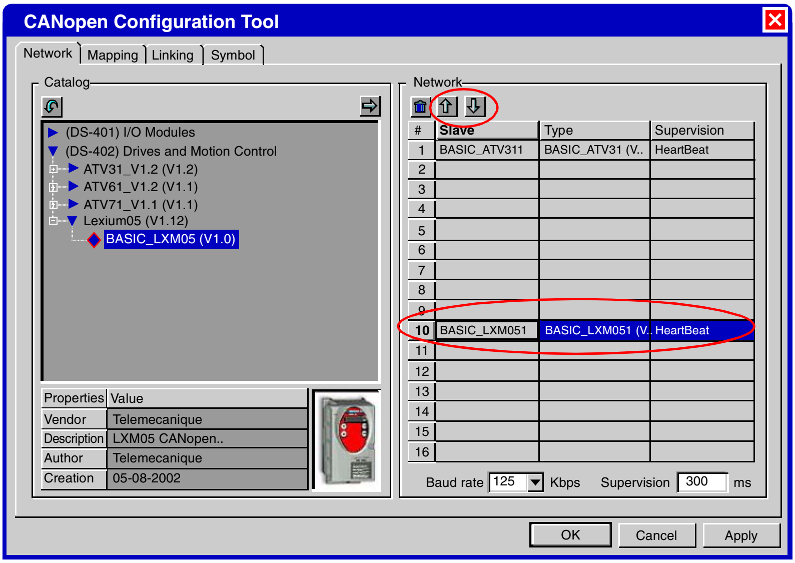
Telemecanique
Project
Describe
Program
Document
Configure
Program
Debug
Configure the hardware
Configure the data
Configure the behavior
Define the protections
TWDLCAA24DRF
RS232
TWDNCO1M
Address 1
TwidoSuite

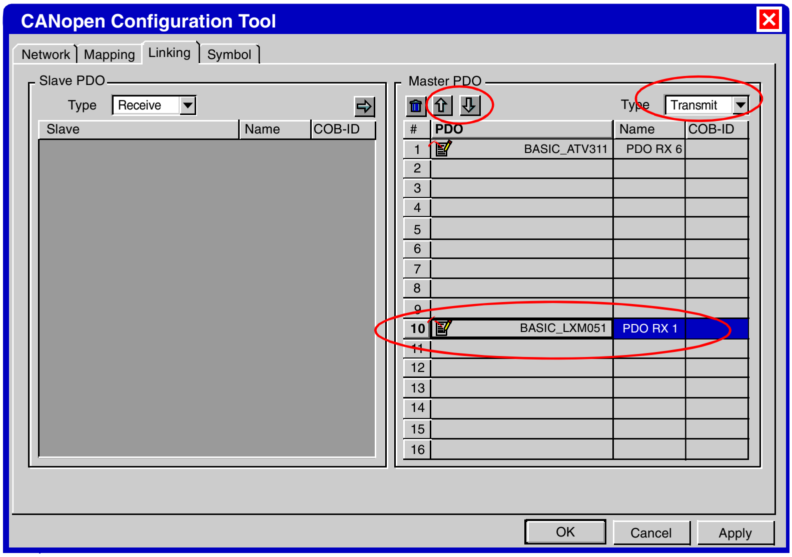
<b>Description of the module</b>
Reference TWDNCO1M
Address 1
Description CANopen bus master module (50 mA)
<b>Configuring the module.</b>
<b>Configure</b>
<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>BASIC_ATV31 (V1.0)</td><td>HeartBeat</td></tr>
</table>
<table>
<tr><th>Used</th><th>Address</th><th>Symbol</th><th>Object</th></tr>
<tr><td>☑</td><td>%IWC1.0.0</td><td>D_STATUS_BASIC_ATV311</td><td>Drivecom status registe</td></tr>
<tr><td>☑</td><td>%IWC1.0.1</td><td>D_CONTROL_BASIC_ATV311</td><td>Control effort</td></tr>
<tr><td>☑</td><td>%IWC1.0.2</td><td>D_IERROR_BASIC_ATV311</td><td>Error code</td></tr>
<tr><td>☑</td><td>%QWC1.0.0</td><td>D_COMMAND_BASIC_ATV31</td><td>Drivecom command reg</td></tr>
<tr><td>☑</td><td>%QWC1.0.1</td><td>D_TARGET_BASIC_ATV311</td><td>Target velocity</td></tr>
</table>
Click <b>Configure</b>.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>Expand the tree of the Drives and Motion Control (DS-402) profile:


▶ (DS-401) I/O Modules
▼ (DS-402) Drives and Motion Control
▶ ATV31_V1.2 (V1.2)
▶ ATV61_V1.2 (V1.1)
▶ ATV71_V1.1 (V1.1)
▼ Lexium05 (V1.12)
◆ BASIC_LXM05 (V1.0)
<table>
<tr><th>Properties</th><th>Value</th></tr>
<tr><td>Vendor</td><td>Telemecanique</td></tr>
<tr><td>Description</td><td>EDS file for Altivar 31 CANopen..</td></tr>
<tr><td>Author</td><td>S.T.I.E.</td></tr>
<tr><td>Creation</td><td>01-14-2004</td></tr>
</table>
<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>BASIC_ATV31 (V..</td><td>HeartBeat</td></tr>
<tr><td>2</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td></td><td></td><td></td></tr>
<tr><td>11</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td></td><td></td><td></td></tr>
</table>
Baud rate 125 Kbps Supervision 300 ms</td></tr>
<tr><td>3</td><td>Select "BASIC_LXM05 (V1.0)" in the "Catalog" area and click on and click ⇨ to add the Lexium 05 servo drive in the "Network" area:


▶ (DS-401) I/O Modules
▼ (DS-402) Drives and Motion Control
▶ ATV31_V1.2 (V1.2)
▶ ATV61_V1.2 (V1.1)
▶ ATV71_V1.1 (V1.1)
▼ Lexium05 (V1.12)
◆ BASIC_LXM05 (V1.0)
<table>
<tr><th>Properties</th><th>Value</th></tr>
<tr><td>Vendor</td><td>Telemecanique</td></tr>
<tr><td>Description</td><td>EDS file for Altivar 31 CANopen..</td></tr>
<tr><td>Author</td><td>S.T.I.E.</td></tr>
<tr><td>Creation</td><td>01-14-2004</td></tr>
</table>
<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>BASIC_ATV31 (V..</td><td>HeartBeat</td></tr>
<tr><td>2</td><td>BASIC_LXM051</td><td>BASIC_LXM05 (V..</td><td>None</td></tr>
<tr><td>3</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td></td><td></td><td></td></tr>
<tr><td>11</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td></td><td></td><td></td></tr>
</table>
Baud rate 125 Kbps Supervision 300 ms</td></tr>
</table>

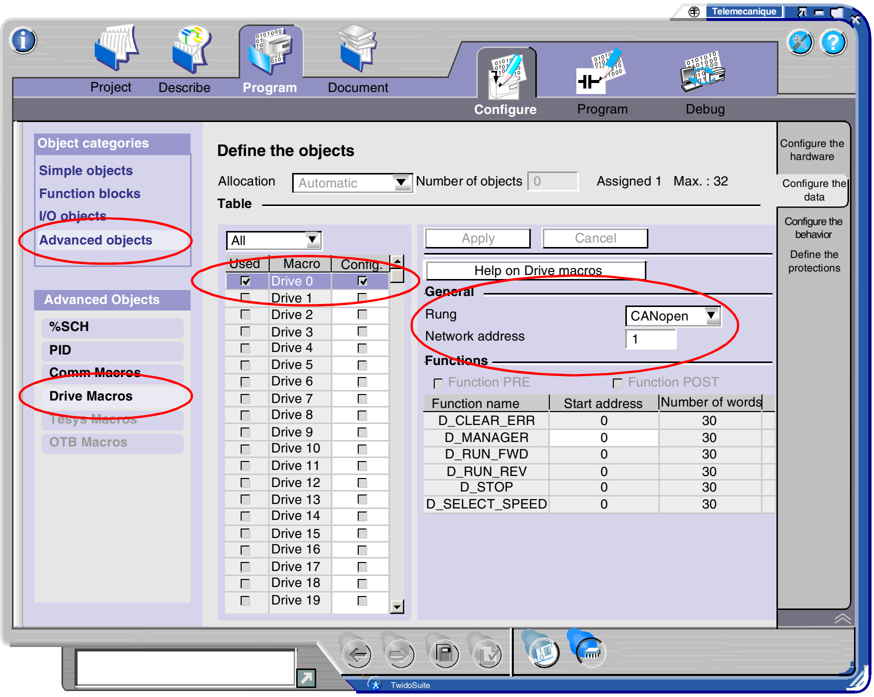
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>
<p>For better visibility, the network address of the servo drive will be address 10.</p>
<p>Click on ⇩ to declare the new speed controller at the network address 10:</p>

<p>CANopen Configuration Tool</p>
<p>Network | Mapping | Linking | Symbol</p>
<p>Catalog</p>
<ul>
<li>(DS-401) I/O Modules</li>
<li>(DS-402) Drives and Motion Control
<ul>
<li>ATV31_V1.2 (V1.2)</li>
<li>ATV61_V1.2 (V1.1)</li>
<li>ATV71_V1.1 (V1.1)</li>
<li>Lexium05 (V1.12)
<ul><li>BASIC_LXM05 (V1.0)</li></ul></li>
</ul></li>
</ul>
<table>
<tr><th>Properties</th><th>Value</th></tr>
<tr><td>Vendor</td><td>Telemecanique</td></tr>
<tr><td>Description</td><td>LXM05 CANopen..</td></tr>
<tr><td>Author</td><td>Telemecanique</td></tr>
<tr><td>Creation</td><td>05-08-2002</td></tr>
</table>
<p>Network</p>
<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>BASIC_ATV31 (V..</td><td>HeartBeat</td></tr>
<tr><td>2</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>BASIC_LXM051</td><td>BASIC_LXM051 (V..</td><td>HeartBeat</td></tr>
<tr><td>11</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td></td><td></td><td></td></tr>
</table>
<p>Baud rate 125 Kbps Supervision 300 ms</p>
<p>OK | Cancel | Apply</p>
<p><b>Important note</b>: When the Supervision type stays on "None", the motor will continue to turn even if the communication between the CANopen bus and the Lexium 05 is interrupted. <b>Change the Heartbeat value</b>.</p>
<p>Management of the error mode is as follows:</p>
<ul>
<li>Nodeguarding: The Master monitors the Slave.</li>
<li>Heartbeat: The Slave monitors the Master. If the RJ45 or the SUB-D 9 cable of the TWIDO is disconnected, the servo drive stops.</li>
<li>None: There is no monitoring, even if one of the cable is disconnected, the servo motor continues operating.</li>
</ul>
</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>5</td><td>Select the "Linking" tab.
Use ⇩ to assign the Lexium 05 to PDO number 10 in "Send" and to PDO 10 and 11 in "Receive":

CANopen Configuration Tool
Network | Mapping | Linking | Symbol
Slave PDO
Type: Receive
<table>
<tr><th>Slave</th><th>Name</th><th>COB-ID</th></tr>
</table>
Master PDO
Type: Transmit
<table>
<tr><th>#</th><th>PDO</th><th>Name</th><th>COB-ID</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>PDO RX 6</td><td></td></tr>
<tr><td>2</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>BASIC_LXM051</td><td>PDO RX 1</td><td></td></tr>
<tr><td>11</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td></td><td></td><td></td></tr>
</table>
OK | Cancel | Apply
It is recommended to maintain consistency with the CANopen Slave address and the associated PDOs. To do this, the devices added subsequently in our example will obey the following rule:
<b>slave address = number of the first PDO</b></td></tr>
<tr><td>6</td><td>Click on the "OK" button to confirm. The Lexium 05 is then declared in the Twido controller and in the CANopen master.</td></tr>
</table>

Macro Drive of Lexium 05

The following method allows configuration of the Lexium 05 in the Macro Drive of the application in order to be able to control it.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>
<p>In the TwidoSuite main window, select the task <b>Program</b> → <b>Configure</b> → <b>Configure the data</b>.
In the left side of the window, select <b>Advanced objects</b> → <b>Drive Macros</b>.
The first ATV31 speed controller is declared in the Macro Drive "Drive 0". It can be seen that its CANopen network address is indeed 1.</p>

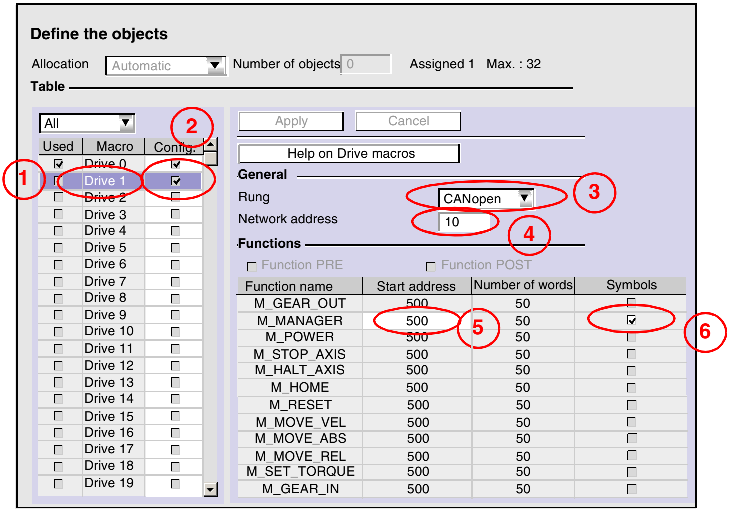


<p>Define the objects</p>
<p>Allocation: Automatic
Number of objects: 0
Assigned 1 Max. : 32</p>
<p>Table
All</p>

<table>
<tr><th>Used</th><th>Macro</th><th>Config.</th></tr>
<tr><td>☑</td><td>Drive 0</td><td>☑</td></tr>
<tr><td>☐</td><td>Drive 1</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 2</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 3</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 4</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 5</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 6</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 7</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 8</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 9</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 10</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 11</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 12</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 13</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 14</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 15</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 16</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 17</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 18</td><td>☐</td></tr>
<tr><td>☐</td><td>Drive 19</td><td>☐</td></tr>
</table>

<p>General
Rung: CANopen
Network address: 1</p>
<p>Functions
☐ Function PRE ☐ Function POST</p>

<table>
<tr><th>Function name</th><th>Start address</th><th>Number of words</th></tr>
<tr><td>D_CLEAR_ERR</td><td>0</td><td>30</td></tr>
<tr><td>D_MANAGER</td><td>0</td><td>30</td></tr>
<tr><td>D_RUN_FWD</td><td>0</td><td>30</td></tr>
<tr><td>D_RUN_REV</td><td>0</td><td>30</td></tr>
<tr><td>D_STOP</td><td>0</td><td>30</td></tr>
<tr><td>D_SELECT_SPEED</td><td>0</td><td>30</td></tr>
</table>
</td></tr>
</table>

Step	Action
2	1 – Select **Drive 1**. 2 - Check the **Configure** box. 3 - Select the **CANopen** network. 4 - Enter the network address **10** of the servo drive. 5 - Click in the white area (Beginning address) and enter the value **500**. 6 - Check the **Symbols** box.  **Note**: Macro Drive 0 integrated in part I (ATV31) uses 30 words starting at address 0; i.e. words %MW0 to %MW29. Macro Drive 1 (Lexium 05) must have a different address range. In our example, Macro Drive 1 will use words %MW500 to %MW549.
3	Click on the "Apply" button to confirm.

Define the objects

Allocation: Automatic   Number of objects: 0   Assigned 1   Max. : 32

Table

All

Used	Macro	Config.
☑	Drive 0	☑
□	Drive 1	☑
□	Drive 2	□
□	Drive 3	□
□	Drive 4	□
□	Drive 5	□
□	Drive 6	□
□	Drive 7	□
□	Drive 8	□
□	Drive 9	□
□	Drive 10	□
□	Drive 11	□
□	Drive 12	□
□	Drive 13	□
□	Drive 14	□
□	Drive 15	□
□	Drive 16	□
□	Drive 17	□
□	Drive 18	□
□	Drive 19	□

General

Rung: CANopen

Network address: 10

Functions

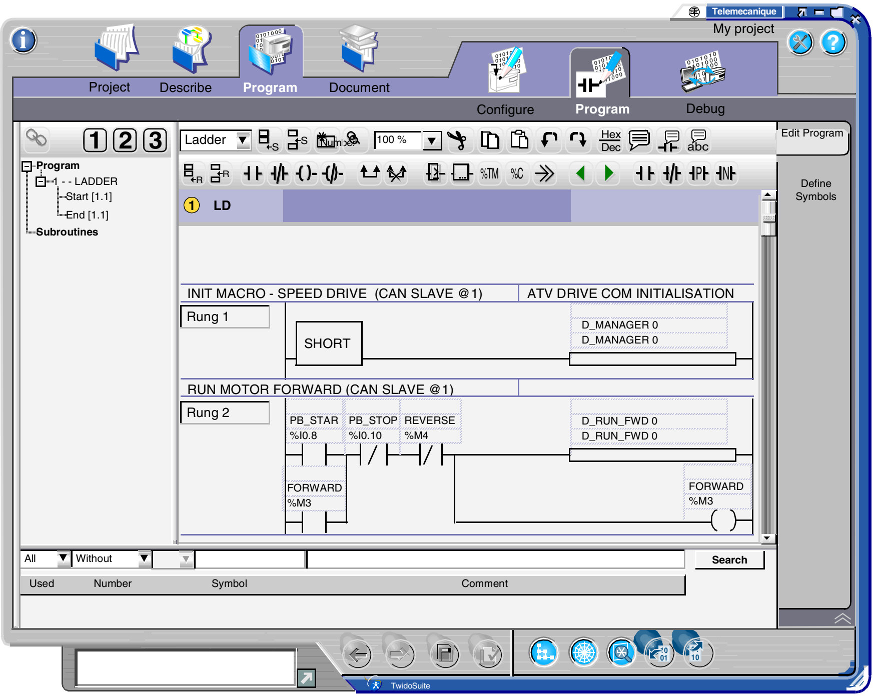
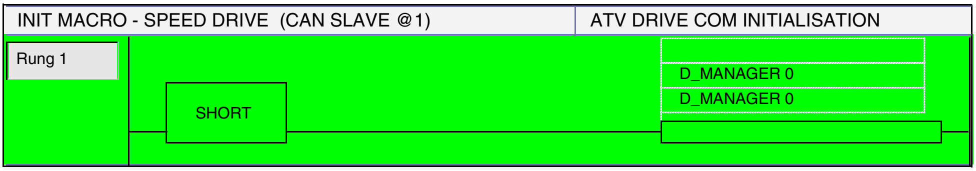
□ Function PRE   □ Function POST

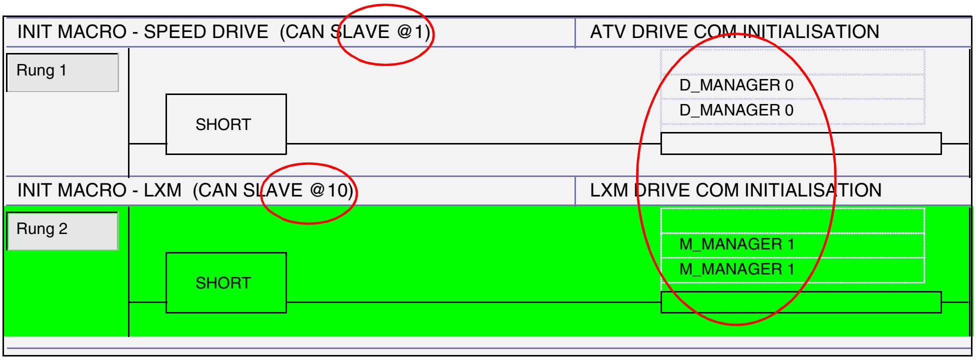
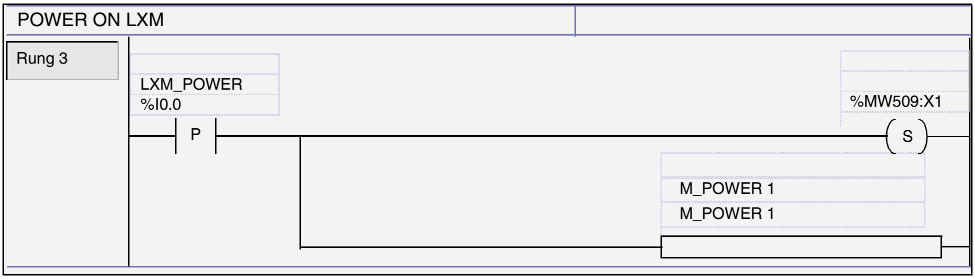
Function name	Start address	Number of words	Symbols
M_GEAR_OUT	500	50	□
M_MANAGER	500	50	☑
M_POWER	500	50	□
M_STOP_AXIS	500	50	□
M_HALT_AXIS	500	50	□
M_HOME	500	50	□
M_RESET	500	50	□
M_MOVE_VEL	500	50	□
M_MOVE_ABS	500	50	□
M_MOVE_REL	500	50	□
M_SET_TORQUE	500	50	□
M_GEAR_IN	500	50	□

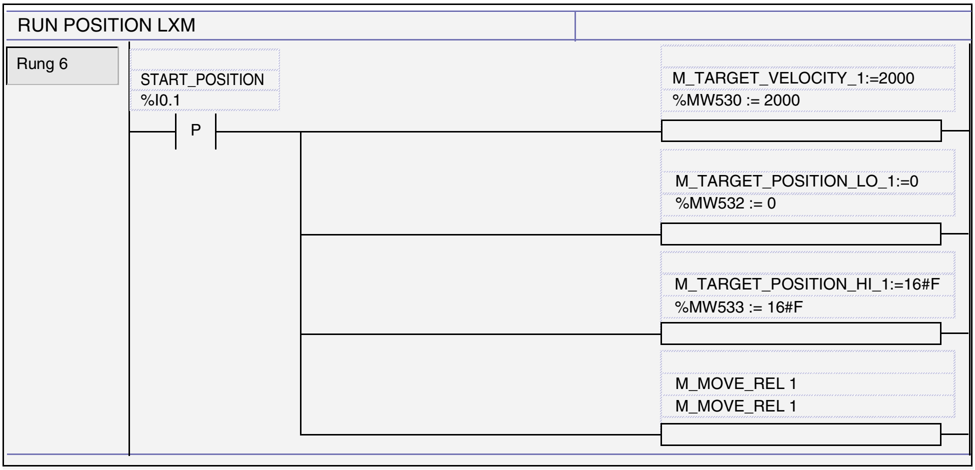
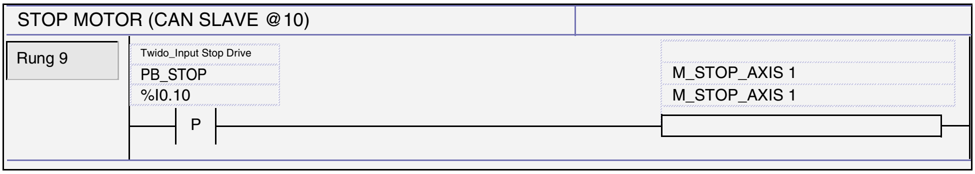
Programming

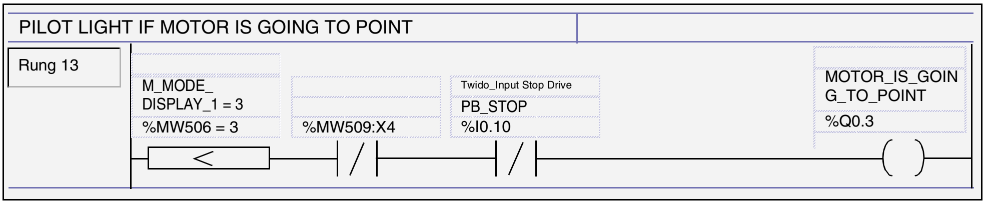
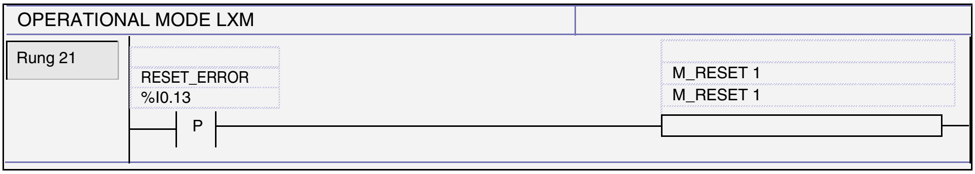
The following method allows modification of the application to control the Lexium 05.

The Macro Drive is used to control the servo drive transparently for the user using the Drive com status chart.

Step	Action
1	In the TwidoSuite main window, select the task **Program** → **Program** → **Edit Program**.  Telemecanique My project Project Describe Program Document Configure Program Debug Edit Program Define Symbols Program 1 - - LADDER Start [1.1] End [1.1] Subroutines Ladder 100 % Hex Dec abc LD INIT MACRO - SPEED DRIVE (CAN SLAVE @1) ATV DRIVE COM INITIALISATION Rung 1 SHORT D_MANAGER 0 D_MANAGER 0 RUN MOTOR FORWARD (CAN SLAVE @1) Rung 2 PB_STAR %I0.8 PB_STOP %I0.10 REVERSE %M4 D_RUN_FWD 0 D_RUN_FWD 0 FORWARD %M3 FORWARD %M3 All Without Search Used Number Symbol Comment TwidoSuite
2	Click on the line "RUNG 1" to select the whole line. The selected line turns green:  INIT MACRO - SPEED DRIVE (CAN SLAVE @1) ATV DRIVE COM INITIALISATION Rung 1 SHORT D_MANAGER 0 D_MANAGER 0
3	Copy the line: "Edit / Copy" or "Ctrl+C". Select an other line ("RUNG 2" for example). Paste the new line: "Edit / Paste" or "Ctrl+V".

Step	Action
4	Select the new line. Double click on the function "D_MANAGER_0" and change it to "M_MANAGER_1" in order to match the Macro no. of the Lexium 05. Confirm this modification with the "Enter" key on the keyboard. Click on the line header to change the comment. Press the "Enter" key of the keyboard to confirm.  INIT MACRO - SPEED DRIVE (CAN SLAVE @1) — ATV DRIVE COM INITIALISATION Rung 1 SHORT D_MANAGER 0 D_MANAGER 0 INIT MACRO - LXM (CAN SLAVE @10) — LXM DRIVE COM INITIALISATION Rung 2 SHORT M_MANAGER 1 M_MANAGER 1 The first line allowing to control the Lexium 05 is created.
5	Create the following line: Activate the servo motor and switch the state of the Lexium from READY to RUN:  POWER ON LXM Rung 3 LXM_POWER %I0.0 P %MW509:X1 S M_POWER 1 M_POWER 1

Step	Action
6	Create the following lines: Send a relative movement command:  RUN POSITION LXM Rung 6 START_POSITION %I0.1 P M_TARGET_VELOCITY_1:=2000 %MW530 := 2000 M_TARGET_POSITION_LO_1:=0 %MW532 := 0 M_TARGET_POSITION_HI_1:=16#F %MW533 := 16#F M_MOVE_REL 1 M_MOVE_REL 1 Send a command to stop the servo motor axis:  STOP MOTOR (CAN SLAVE @10) Rung 9 Twido_Input Stop Drive PB_STOP %I0.10 P M_STOP_AXIS 1 M_STOP_AXIS 1

Step	Action
7	Create the following lines: Turn on the indicator light when the servo motor is running:  PILOT LIGHT IF MOTOR IS GOING TO POINT Rung 13 M_MODE_ DISPLAY_1 = 3 %MW506 = 3 < %MW509:X4 Twido_Input Stop Drive PB_STOP %I0.10 MOTOR_IS_GOIN G_TO_POINT %Q0.3 Send a RESET command for an error:  OPERATIONAL MODE LXM Rung 21 RESET_ERROR %I0.13 P M_RESET 1 M_RESET 1 **Note:** While in progress, the variables are represented as addresses and symbols.
8	In the TwidoSuite main window, select the task **Project** → **Save the current project**.
9	Load the new application into the Twido controller (see *Installing software and loading applications, p. 36*).

Note: The command for switching the state from READY to RUN is programmed permanently: once you are in Power ON mode, you can no longer switch back to Power OFF (no reset command for the associated bit).

Note: For details on the Macro DRIVE used, refer to the TwidoSuite online help.

Presentation of the Lexium 05 application

7

At a Glance

Overview

This section describes the operation of the Lexium 05.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Operating mode	74
Grafcet of the Lexium 05 application	75

Operating mode

Commands

When system installation is complete, the system may be controlled using six push buttons and one two-position button.

Input **Corresponding symbol**	**Function**	**Description**	**Output** **Corresponding symbol**
%I0.0 LXM POWER	Switching the Lexium 05 from the READY to RUN	The Lexium 05 switches from READY to RUN.	No output to match input
%I0.1 START_POSITION_LXM	Starting the servo motor	The corresponding LED turns on when the servo motor is running.	%Q0.3 MOTOR_IS_GOING_TO_POINT
%I0.8 PB_START_FORWARD	Motor forward start	The corresponding LED turns on when the selected speed is reached.	%Q0.0 SD_RUN_FORW
%I0.9 PB_START_REVERSE	Motor reverse start	The corresponding LED turns on when the selected speed is reached.	%Q0.1 SD_RUN_REV
%I0.10 PB_STOP	Stopping the motor and the servo motor	The corresponding LED turns on when the motor stops.	%Q0.2 SD_STOPPED
%I0.11 PB_SLOW_FAST	Predefined motor speeds	Apply a predefined speed when changing position: ● %I0.11 is set to 0: 3,300 rpm ● %I0.11 is set to 1: 88 rpm	No output to match input
%I0.13 RESET_ERROR	Acknowledgement of an ATV31 and/or Lexium 05 error	Reset an error (the cause of the error must be eliminated):	No output to match input

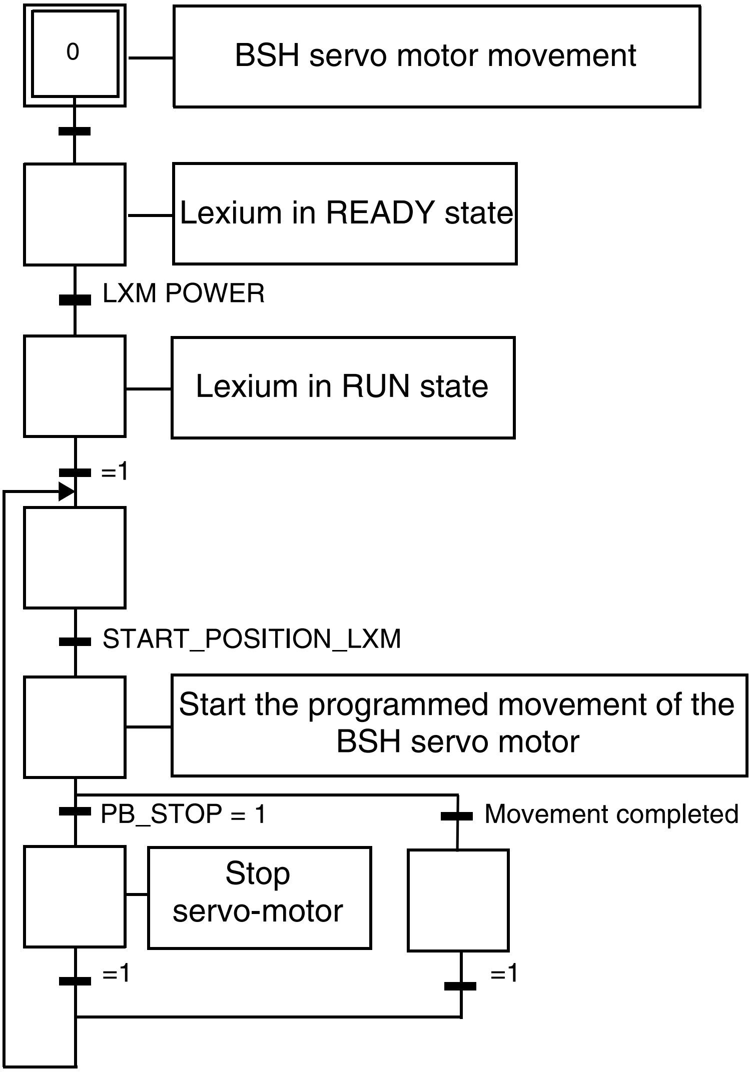
Note: In our sample application, the push button for %I0.10 must be pushed (Stop motor) in order to change the direction of motor rotation.

Note: In the sample program, the speed configured manually with the XBTN is not memorized after a motor stop. The speed of 3,300 rpm depends on the configuration of the variable speed controller.

Grafcet of the Lexium 05 application

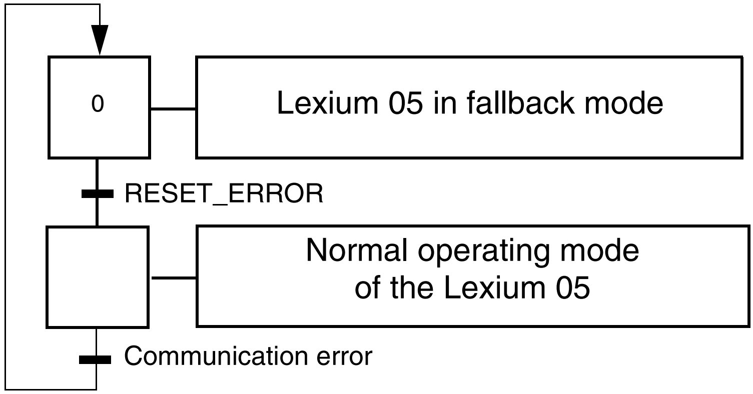
Simplified chart of Lexium 05 operation

The operation of the Lexium 05 may be represented in the following manner:



Communication error chart

Communication error chart for the Lexium 05:


0
Lexium 05 in fallback mode
RESET_ERROR
Normal operating mode of the Lexium 05
Communication error

Note: In the event of a communication error, the BSH servo motor state depends on the Lexium 05 configuration (fallback modes)

Add an Advantys OTB I/O module

IV

At a Glance

Overview

This section of the document presents a possible evolution of the main system, that is the addition of an Advantys OTB I/O module.

Note: If you are not using a speed controller in your application, erase the associated elements in the CANopen configuration as well as in the Twido controller application program.

Application principle

The following example will allow to control outputs of the Advantys OTB I/O module the same as those of the Twido controller.

Files

Files for this configuration may be found on the CD-ROM BUNDLE (DIA3CD3050101F) in the "Applicative files\Partie_IV" folder for the Twido controller application.

What's in this Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
8	Hardware implementation of an Advantys OTB module	79
9	Software implementation of an Advantys OTB module	83
10	Add expansion modules to the Advantys OTB	95

Hardware implementation of an Advantys OTB module

8

At a Glance

Overview

This chapter describes hardware implementation of an Advantys OTB I/O module added to the main system.

What's in this Chapter?

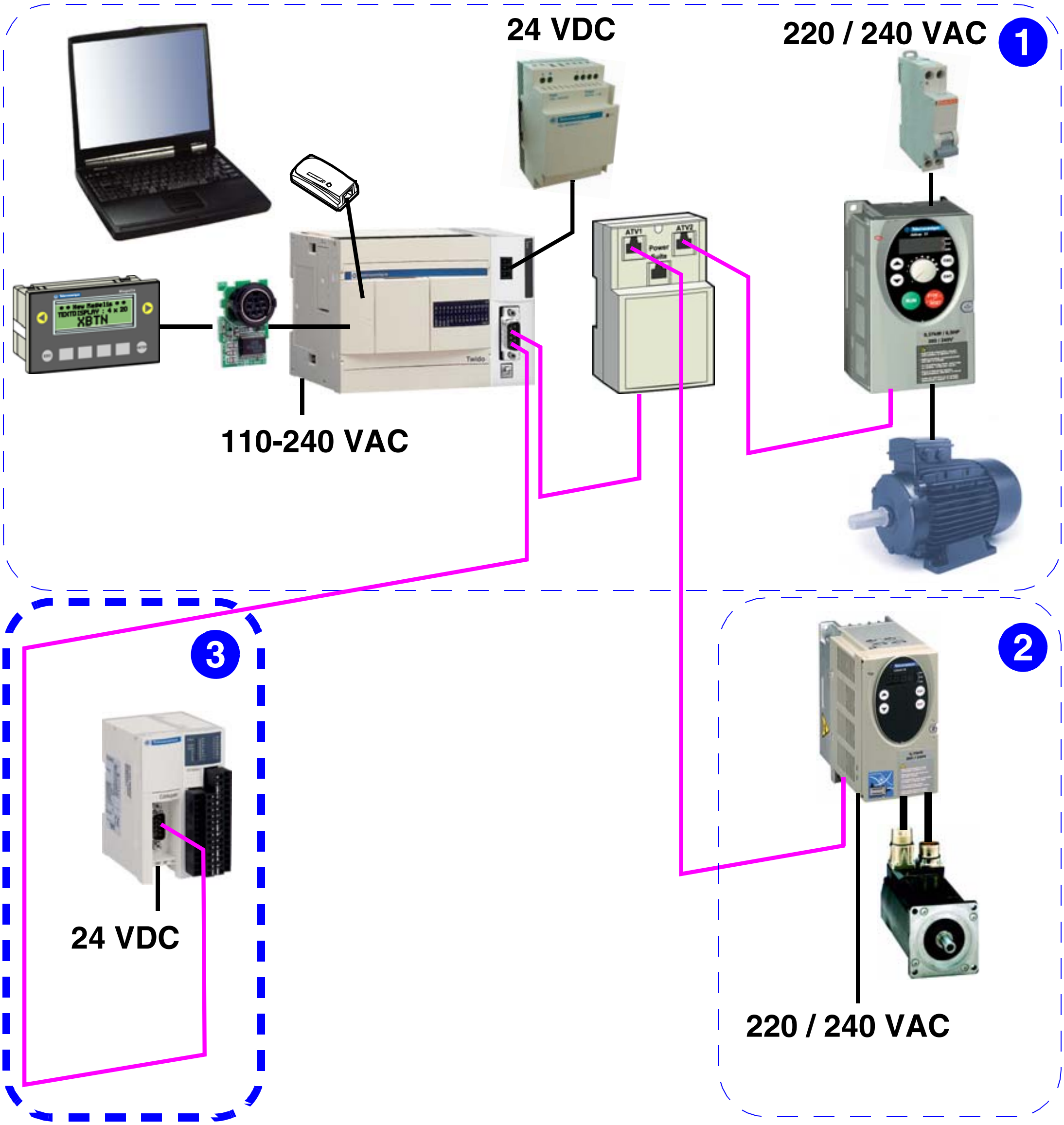
This chapter contains the following topics:

Topic	Page
Advantys OTB outputs cabling	80
Cabling of the CANopen network	81
Advantys OTB I/O cabling	82
Power supply	82

Advantys OTB outputs cabling

Installation diagram

This part of the document covers the following mounting:



Cabling of the CANopen network

Principle

The objective is to connect the Advantys OTB to the Twido controller using a cable that first must be prepared.

OTB-Twido cable preparation

Connect a SUB-D 9 TSXCANKCDF90T connector with the TSXCANCA50 cable (see *TAP-Twido Cable Preparation, p. 27*).

Cable pin assignment in the terminal of the SUB-D 9 connector:

No.	Signal	Cable	Connector	Wire color
1	CAN_H	OTB/Twido	CH1	White
2	CAN_L	OTB/Twido	CL1	Blue
3	GND	OTB/Twido	CG1	Black
4	V+	OTB/Twido	V+1	Red

Note: Toggle the line end terminator of the connector on the Advantys OTB side to "On".

OTB side cabling

The SUB-D 9 end of the previously prepared cable is connected to the Advantys OTB.

Twido side cabling

The "bare wire" end of the previously prepared cable is connected to the SUB-D 9 connector joined to the Twido controller. This mounting is performed in the same manner as previously with the following pin assignment:

No.	Signal	Cable	Connector	Wire color
5	CAN_H	OTB/Twido	CH2	White
6	CAN_L	OTB/Twido	CL2	Blue
7	GND	OTB/Twido	CG2	Black
8	V+	OTB/Twido	V+2	Red

Note: Toggle the line end terminator of the SUB-D 9 connector on the Twido controller side to "Off".

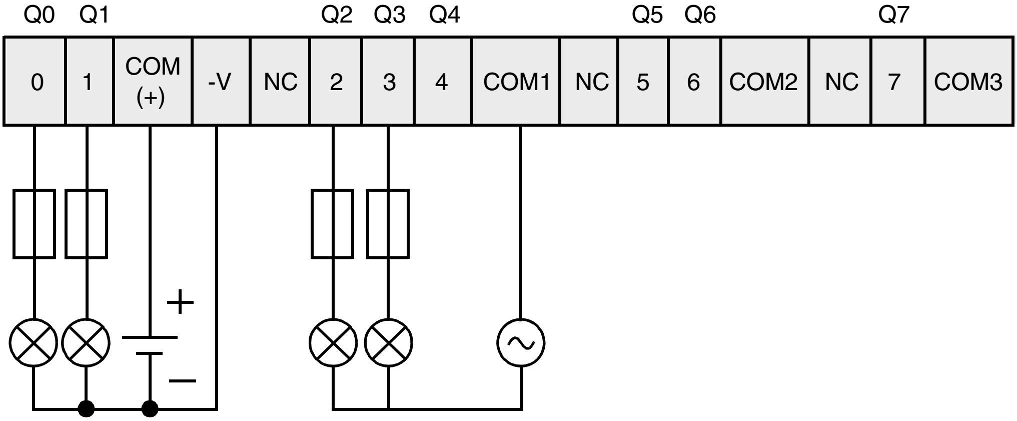
Advantys OTB I/O cabling

OTB inputs cabling

Inputs of the Advantys OTB I/O module are not used in this example. For more information, see the reference guide 160384.

OTB outputs cabling

Outputs cabling diagram:



Additional information on outputs:

Output	Indication displayed	Information
%QWC1.1.0:X0	Forward motor rotation	Source output
%QWC1.1.0:X1	Reverse motor rotation	Source output
%QWC1.1.0:X2	Motor stopped	Relay output
%QWC1.1.0:X3	Servo motor in rotation	Relay output

Power supply

Power on the Advantys OTB

To power on the Advantys OTB I/O module, follow the recommendations provided in reference guide 1606384.

Software implementation of an Advantys OTB module

9

At a Glance

Overview

This chapter describes software implementation of an Advantys OTB I/O module added to the main system.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Configuration of Advantys OTB communications	84
Add an Advantys OTB module in the Twido controller application	86

Configuration of Advantys OTB communications

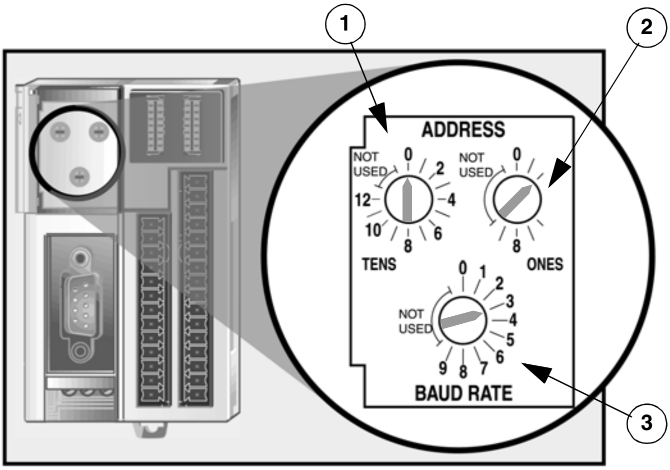
Principle

The system products must be configured according to the Twido application controller as follows:

- Address 1: ATV31
- Address 2: Advantys OTB
- Address 10: Lexium 05
- Transmission Speed: 125 kBits/s

Description

Front panel of the Advantys OTB I/O module:



No.	Function
1	Network address (Node-ID x10) encoder wheel
2	Network address (Node-ID x1) encoder wheel
3	Transmission speed encoder wheel

Method

Configuration of Advantys OTB communication parameters:

Step	Action
1	Cut off all power to the Advantys OTB.
2	Set the lower encoder wheel 3 to the position corresponding to the required baud rate. In the example: encoder wheel 3 is set to position 3 (125 kBits/s).
3	Set encoder wheels 1 and 2 to the position corresponding to the desired network address. In the example, the address is 2 (10x0 + 1x2): ● encoder wheel 1: position 0 (tens) ● encoder wheel 2: position 2 (units)
4	Reapply power to the OTB in order to implement the new configuration.

Note: After changing any features of the components of the CANopen network, the bus should be reinitialized by cycling the Twido controller On/Off.

Baud rate selection table

The possible speeds are as follows:

Position (lower encoder wheel)	Baud rate
0	10 Kbps
1	20 Kbps
2	50 Kbps
3	125 Kbps
4	250 Kbps
5	500 Kbps
6	800 Kbps
7	1 Mbps
8	Automatic
9	Default rate (250 Kbps)

Note: The value of 8 allows automatic search for the bus transmission speed. The search starts at a value of 1 Mbps then progressively lowers over successive searches until communication is established on the bus. The automatic search only works on an operational CANopen network.

Add an Advantys OTB module in the Twido controller application

Principle

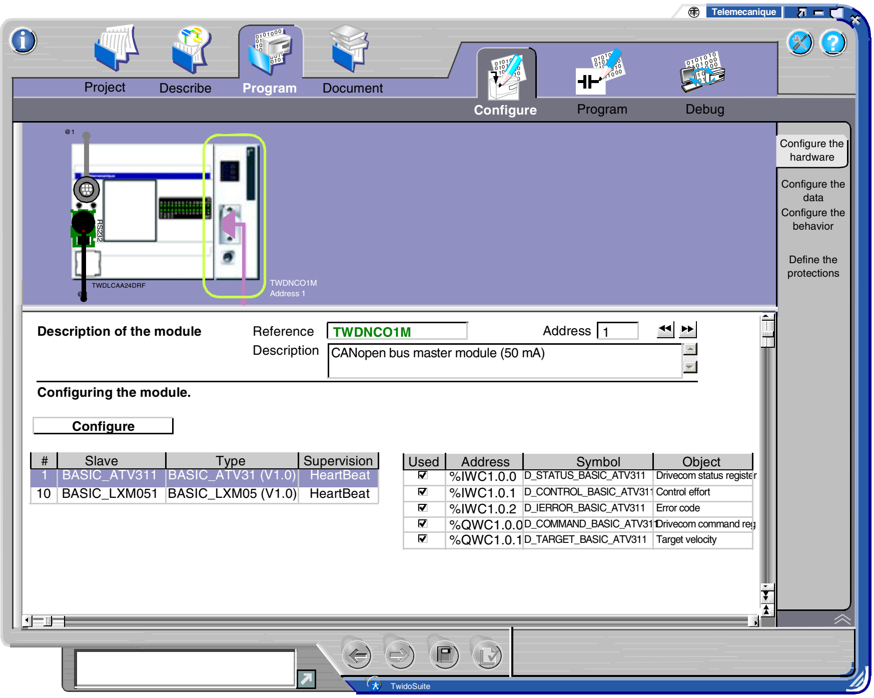
The objective is to modify the existing application in order to be able to manage the I/O of the Advantys OTB module.

Software Installation

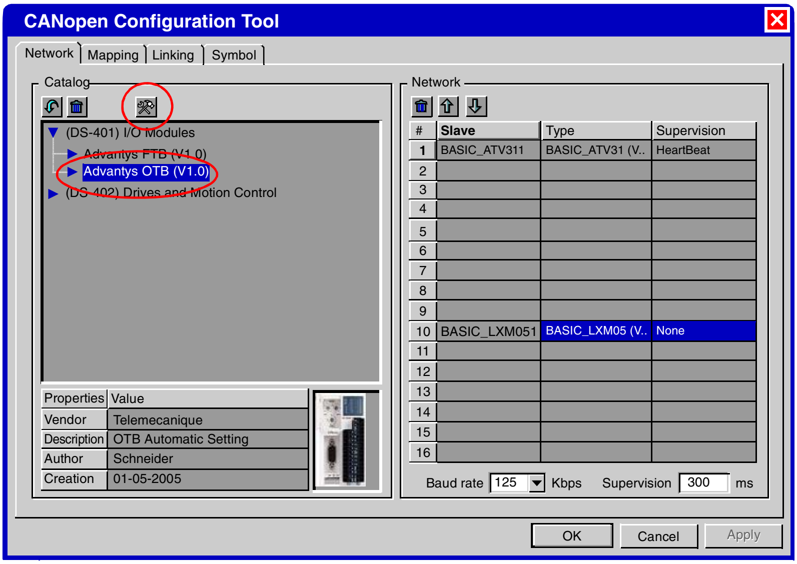
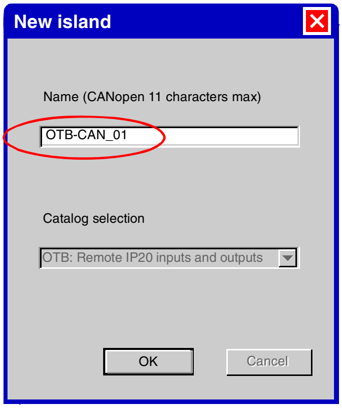
It is first necessary to install the Advantys Configuration Tool software that allows configuration of Advantys OTB and FTB I/O modules and splitters.

Declare the Advantys OTB module

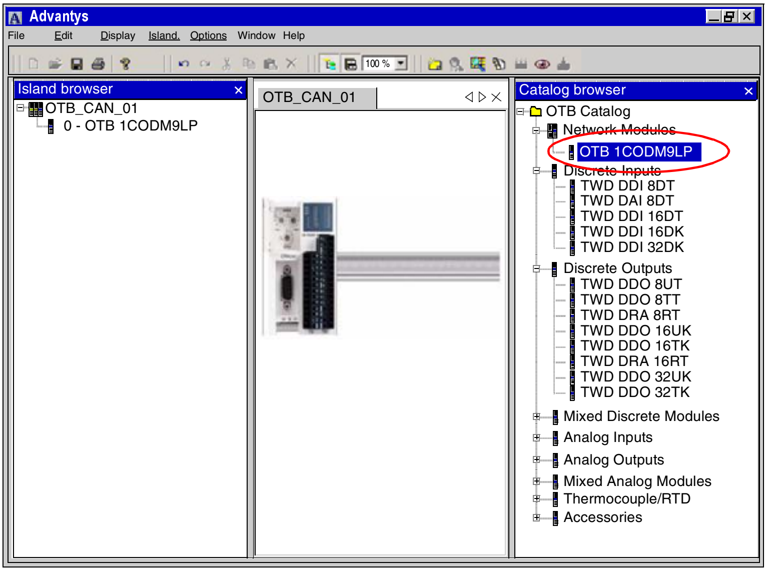
The following method allows declaration of the Advantys OTB I/O module in the Twido controller application.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>
Open the application "BUNDLE\Partie_III\BUNDLE_CAN_Part_III.xpr" using the TwidoSuite software (TwidoSuite must be in disconnected mode).
In the TwidoSuite main window, select the task <b>Program</b> → <b>Configure</b> → <b>Configure hardware</b>.
Select the CANopen network TWDNCO1M:


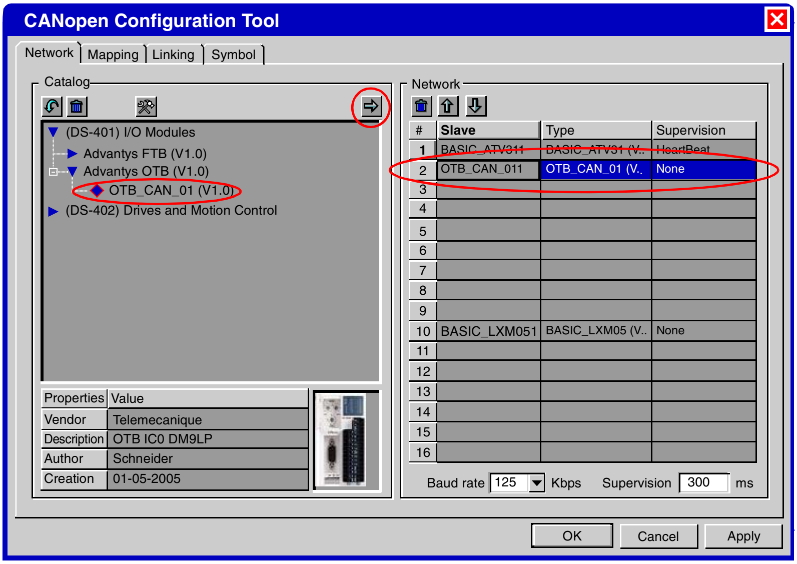
<b>Description of the module</b>
Reference TWDNCO1M
Address 1
Description CANopen bus master module (50 mA)
<b>Configuring the module.</b>
Configure
<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>BASIC_ATV31 (V1.0)</td><td>HeartBeat</td></tr>
<tr><td>10</td><td>BASIC_LXM051</td><td>BASIC_LXM05 (V1.0)</td><td>HeartBeat</td></tr>
</table>
<table>
<tr><th>Used</th><th>Address</th><th>Symbol</th><th>Object</th></tr>
<tr><td>☑</td><td>%IWC1.0.0</td><td>D_STATUS_BASIC_ATV311</td><td>Drivecom status registe</td></tr>
<tr><td>☑</td><td>%IWC1.0.1</td><td>D_CONTROL_BASIC_ATV311</td><td>Control effort</td></tr>
<tr><td>☑</td><td>%IWC1.0.2</td><td>D_IERROR_BASIC_ATV311</td><td>Error code</td></tr>
<tr><td>☑</td><td>%QWC1.0.0</td><td>D_COMMAND_BASIC_ATV311</td><td>Drivecom command reg</td></tr>
<tr><td>☑</td><td>%QWC1.0.1</td><td>D_TARGET_BASIC_ATV311</td><td>Target velocity</td></tr>
</table>
Click <b>Configure</b>.
</td></tr>
</table>

Step	Action
2	Expand the tree of the I/O Modules (DS-401) profile, select "Advantys OTB (V1.0)" and click on ⚒ to launch configuration and addition of a new element:  CANopen Configuration Tool Network / Mapping / Linking / Symbol Catalog (DS-401) I/O Modules Advantys FTB (V1.0) Advantys OTB (V1.0) (DS-402) Drives and Motion Control Properties: Value Vendor: Telemecanique Description: OTB Automatic Setting Author: Schneider Creation: 01-05-2005 Network # / Slave / Type / Supervision 1 / BASIC_ATV311 / BASIC_ATV31 (V.. / HeartBeat 2 3 4 5 6 7 8 9 10 / BASIC_LXM051 / BASIC_LXM05 (V.. / None 11 12 13 14 15 16 Baud rate 125 Kbps Supervision 300 ms OK Cancel Apply
3	In the "New Island" window, enter "OTB_CAN_01" matching the name of the equipment that will be used in TwidoSuite. Next click on the "OK" button to confirm and launch the Advantys Configuration Tool application:  New island Name (CANopen 11 characters max) OTB-CAN_01 Catalog selection OTB: Remote IP20 inputs and outputs OK Cancel

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>Double click on the module "OTB1C0DM9LP" to add it in the new island.



</td></tr>
<tr><td>5</td><td>Perform a "File / Save".
Perform a "File / Quit".
The new island containing the Advantys OTB I/O module is then added to the catalog under the name "OTB_CAN_01".</td></tr>
</table>

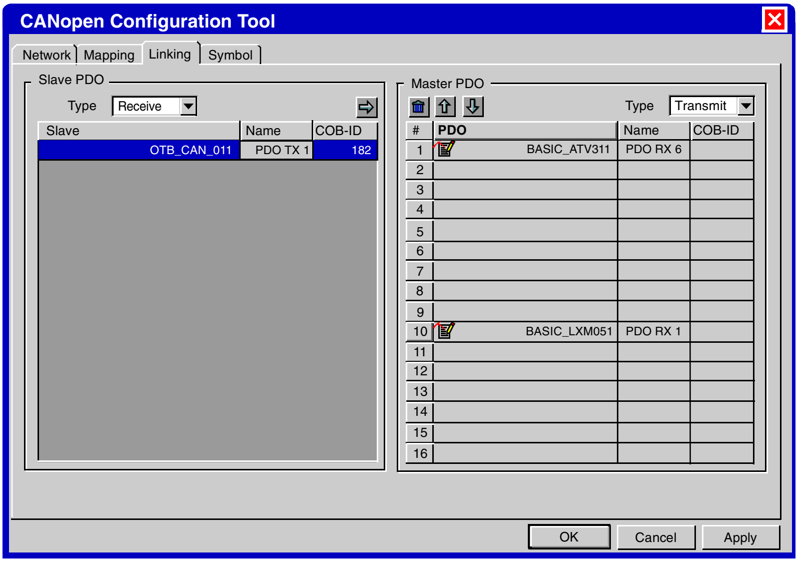
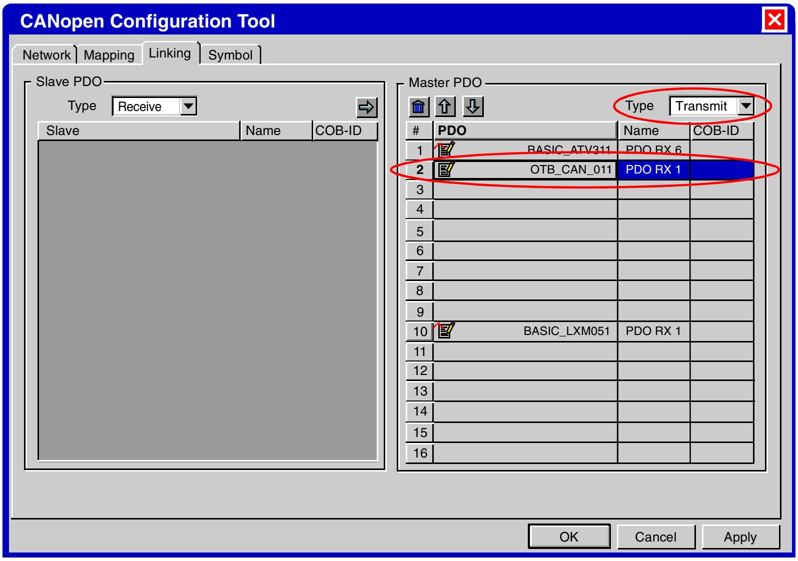
Step	Action
6	Select "OTB_CAN_01 (V1.0)" in the "Catalog" area and click on ⇨ to add the Advantys OTB in the "Network" area:  Note: The Advantys OTB is configured by default to the first free network address, that is the network address 2 in our example.

CANopen Configuration Tool

Properties	Value
Vendor	Telemecanique
Description	OTB IC0 DM9LP
Author	Schneider
Creation	01-05-2005

#	Slave	Type	Supervision
1	BASIC_ATV311	BASIC_ATV31 (V..	HeartBeat
2	OTB_CAN_011	OTB_CAN_01 (V..	None
3			
4			
5			
6			
7			
8			
9			
10	BASIC_LXM051	BASIC_LXM05 (V..	None
11			
12			
13			
14			
15			
16			

Baud rate 125 Kbps Supervision 300 ms

Step	Action
7	Select the "Linking" tab.  Use ⇨ to assign the new Advantys OTB I/O module to the PDO no. 2, in both "Receive" and "Send": 

CANopen Configuration Tool

Network | Mapping | Linking | Symbol

Slave PDO — Type: Receive

Slave	Name	COB-ID
OTB_CAN_011	PDO TX 1	182

Master PDO — Type: Transmit

#	PDO	Name	COB-ID
1	BASIC_ATV311	PDO RX 6	
2			
3			
4			
5			
6			
7			
8			
9			
10	BASIC_LXM051	PDO RX 1	
11			
12			
13			
14			
15			
16			

OK | Cancel | Apply

CANopen Configuration Tool

Network | Mapping | Linking | Symbol

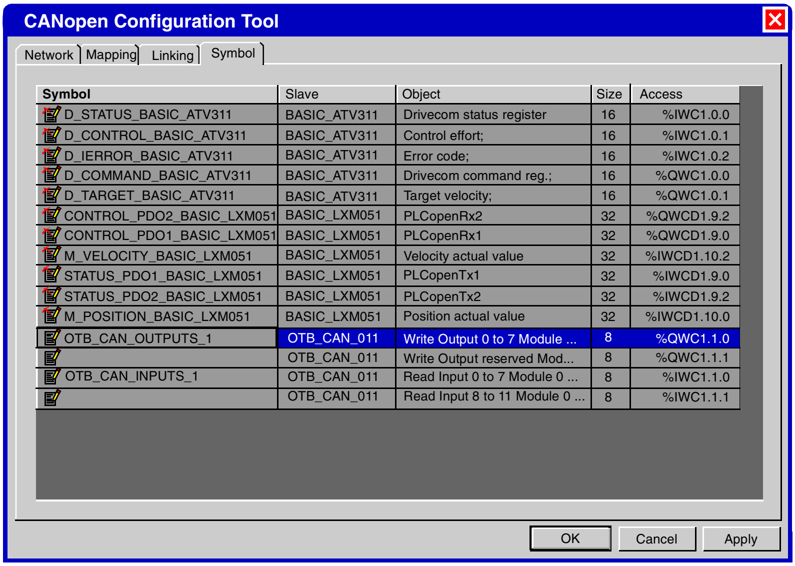
Slave PDO — Type: Receive

Slave	Name	COB-ID

Master PDO — Type: Transmit

#	PDO	Name	COB-ID
1	BASIC_ATV311	PDO RX 6	
2	OTB_CAN_011	PDO RX 1	
3			
4			
5			
6			
7			
8			
9			
10	BASIC_LXM051	PDO RX 1	
11			
12			
13			
14			
15			
16			

OK | Cancel | Apply

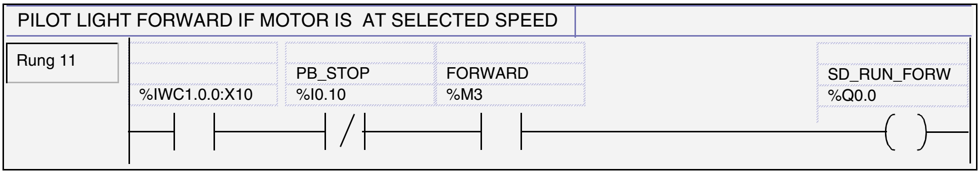
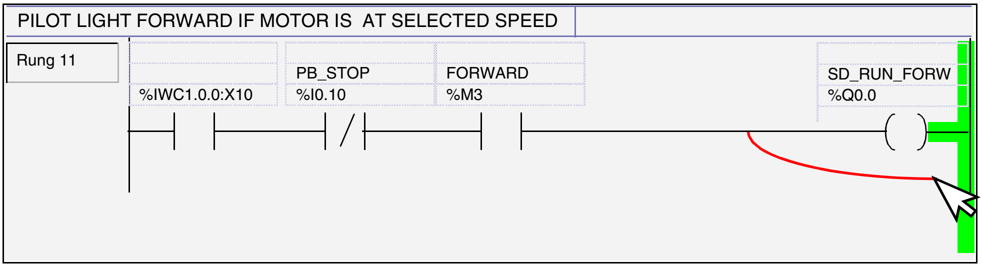
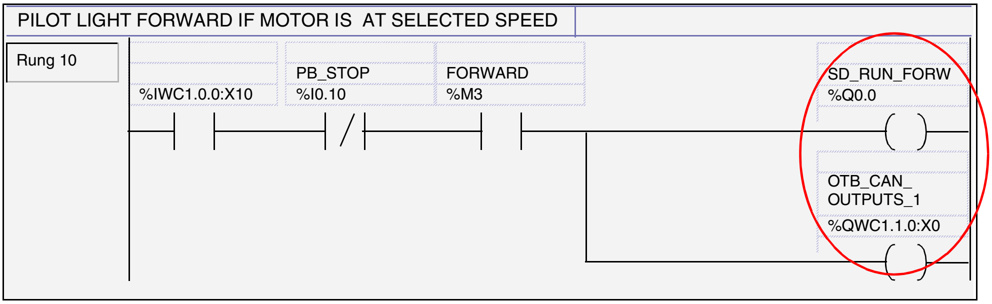
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>8</td><td>Select the "Symbol" tab.
Enter, if necessary a "symbol" for the objects associated with the Advantys OTB I/O module:
● Double click in the object's "Symbol" field,
● Type the name of the "Symbol",
● Confirm using the "ENTER" key of the keyboard.


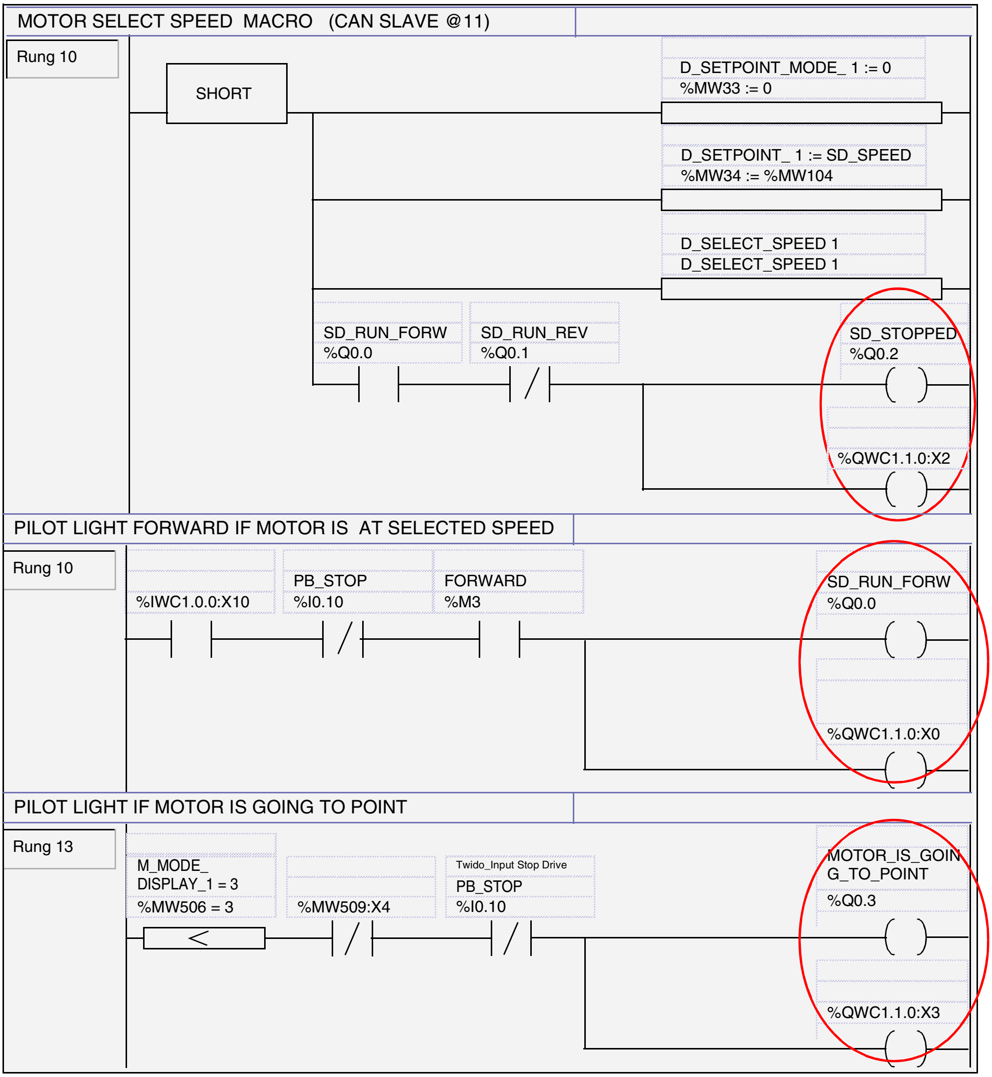
<table>
<tr><th>Symbol</th><th>Slave</th><th>Object</th><th>Size</th><th>Access</th></tr>
<tr><td>D_STATUS_BASIC_ATV311</td><td>BASIC_ATV311</td><td>Drivecom status register</td><td>16</td><td>%IWC1.0.0</td></tr>
<tr><td>D_CONTROL_BASIC_ATV311</td><td>BASIC_ATV311</td><td>Control effort;</td><td>16</td><td>%IWC1.0.1</td></tr>
<tr><td>D_IERROR_BASIC_ATV311</td><td>BASIC_ATV311</td><td>Error code;</td><td>16</td><td>%IWC1.0.2</td></tr>
<tr><td>D_COMMAND_BASIC_ATV311</td><td>BASIC_ATV311</td><td>Drivecom command reg.;</td><td>16</td><td>%QWC1.0.0</td></tr>
<tr><td>D_TARGET_BASIC_ATV311</td><td>BASIC_ATV311</td><td>Target velocity;</td><td>16</td><td>%QWC1.0.1</td></tr>
<tr><td>CONTROL_PDO2_BASIC_LXM051</td><td>BASIC_LXM051</td><td>PLCopenRx2</td><td>32</td><td>%QWCD1.9.2</td></tr>
<tr><td>CONTROL_PDO1_BASIC_LXM051</td><td>BASIC_LXM051</td><td>PLCopenRx1</td><td>32</td><td>%QWCD1.9.0</td></tr>
<tr><td>M_VELOCITY_BASIC_LXM051</td><td>BASIC_LXM051</td><td>Velocity actual value</td><td>32</td><td>%IWCD1.10.2</td></tr>
<tr><td>STATUS_PDO1_BASIC_LXM051</td><td>BASIC_LXM051</td><td>PLCopenTx1</td><td>32</td><td>%IWCD1.9.0</td></tr>
<tr><td>STATUS_PDO2_BASIC_LXM051</td><td>BASIC_LXM051</td><td>PLCopenTx2</td><td>32</td><td>%IWCD1.9.2</td></tr>
<tr><td>M_POSITION_BASIC_LXM051</td><td>BASIC_LXM051</td><td>Position actual value</td><td>32</td><td>%IWCD1.10.0</td></tr>
<tr><td>OTB_CAN_OUTPUTS_1</td><td>OTB_CAN_011</td><td>Write Output 0 to 7 Module ...</td><td>8</td><td>%QWC1.1.0</td></tr>
<tr><td></td><td>OTB_CAN_011</td><td>Write Output reserved Mod...</td><td>8</td><td>%QWC1.1.1</td></tr>
<tr><td>OTB_CAN_INPUTS_1</td><td>OTB_CAN_011</td><td>Read Input 0 to 7 Module 0 ...</td><td>8</td><td>%IWC1.1.0</td></tr>
<tr><td></td><td>OTB_CAN_011</td><td>Read Input 8 to 11 Module 0 ...</td><td>8</td><td>%IWC1.1.1</td></tr>
</table>
</td></tr>
<tr><td>9</td><td>Next click on the "OK" button to confirm the changes. The Advantys OTB I/O module is then declared in the Twido controller application.</td></tr>
</table>

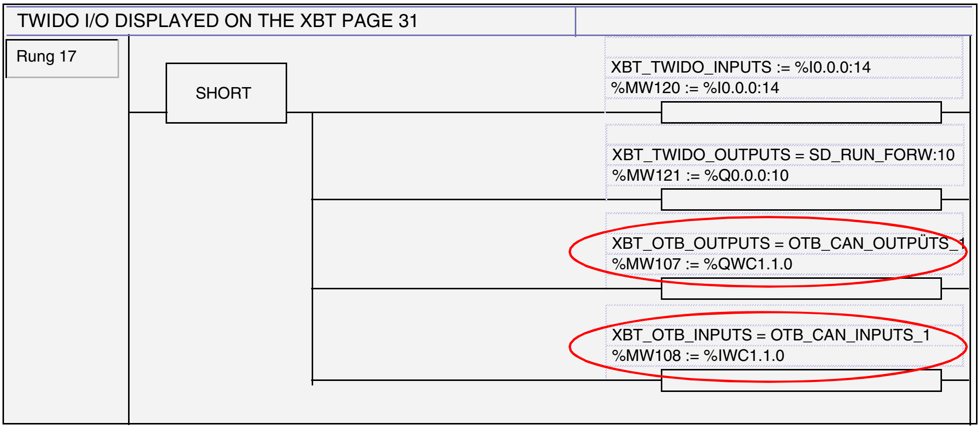
Non-reversible

The following method allows modification of the application to control outputs of the Advantys OTB I/O module the same as those of the Twido controller.

The following method consists in changing the lines that manage the outputs of the Twido controller by adding the outputs of the Advantys OTB.

Step	Action
1	In the TwidoSuite main window, select the task **Program** → **Program** → **Edit Program**.
2	Select a program line corresponding to the control of outputs of the Twido controller:  PILOT LIGHT FORWARD IF MOTOR IS AT SELECTED SPEED Rung 11 %IWC1.0.0:X10 PB_STOP %I0.10 FORWARD %M3 SD_RUN_FORW %Q0.0 Draw a line parallel to the output coil of the Twido controller (output %Q0.0):  PILOT LIGHT FORWARD IF MOTOR IS AT SELECTED SPEED Rung 11 %IWC1.0.0:X10 PB_STOP %I0.10 FORWARD %M3 SD_RUN_FORW %Q0.0
3	Add a coil to which the OTB output is applied: "%QWC1.1.0:X0". The control program line that simultaneously maintains the two outputs:  PILOT LIGHT FORWARD IF MOTOR IS AT SELECTED SPEED Rung 10 %IWC1.0.0:X10 PB_STOP %I0.10 FORWARD %M3 SD_RUN_FORW %Q0.0 OTB_CAN_ OUTPUTS_1 %QWC1.1.0:X0

Step	Action
4	Redo the same changes to be able to control the other outputs X1, X2 and X3: 

Step	Action
5	Add OTB I/O management by the XBT:  TWIDO I/O DISPLAYED ON THE XBT PAGE 31 Rung 17 SHORT XBT_TWIDO_INPUTS := %I0.0.0:14 %MW120 := %I0.0.0:14 XBT_TWIDO_OUTPUTS = SD_RUN_FORW:10 %MW121 := %Q0.0.0:10 XBT_OTB_OUTPUTS = OTB_CAN_OUTPUTS_1 %MW107 := %QWC1.1.0 XBT_OTB_INPUTS = OTB_CAN_INPUTS_1 %MW108 := %IWC1.1.0
6	In the TwidoSuite main window, select the task **Project** → **Save the current project**.
7	Load the new application into the Twido controller (see *Installing software and loading applications, p. 36*).

Add expansion modules to the Advantys OTB

10

Principle

Overview

This section describes how to add I/O expansion modules to the Advantys OTB. These expansion modules allow to manage the Discrete and Analog I/O.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Mount expansion modules	96
Add expansion modules to the Advantys OTB in the Twido controller application.	96

Mount expansion modules

Principle

See reference guide no. 160384.

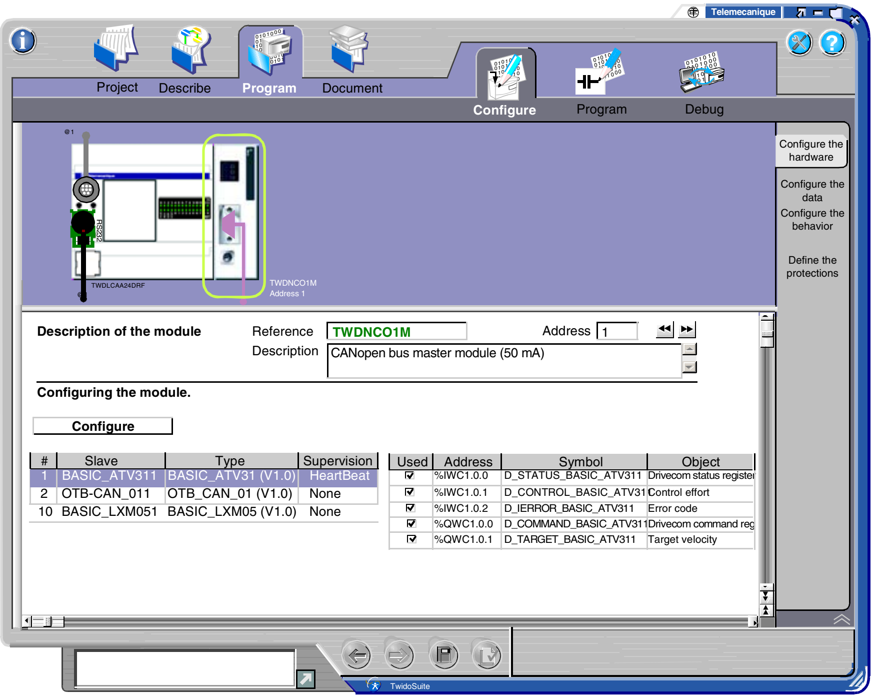
Add expansion modules to the Advantys OTB in the Twido controller application.

Principle

The objective is to modify the existing application in order to be able to manage the I/O of the Advantys OTB I/O module associated with two expansion modules.

Declare the OTB module

The following method allows declaration of the Advantys OTB I/O module in the Twido controller application.

Step	Action
1	Open the project "BUNDLE\Partie_III\BUNDLE_CAN_Part_III.xpr" using the TwidoSuite software (TwidoSuite must be in disconnected mode). In the TwidoSuite main window, select the task **Program** → **Configure** → **Configure hardware**. Select the CANopen network TWDNCO1M:  Click **Configure**.

Description of the module

Reference: TWDNCO1M

Address: 1

Description: CANopen bus master module (50 mA)

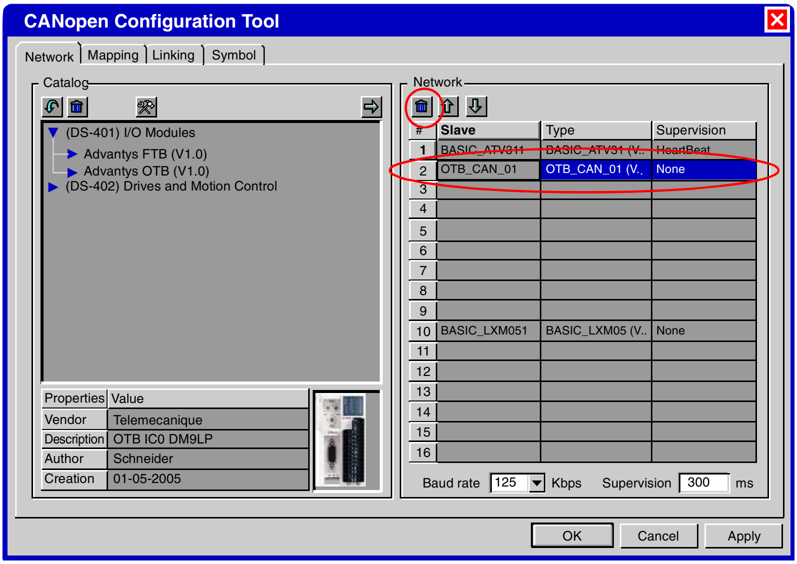
Configuring the module.

Configure

#	Slave	Type	Supervision
1	BASIC_ATV311	BASIC_ATV31 (V1.0)	HeartBeat
2	OTB-CAN_011	OTB_CAN_01 (V1.0)	None
10	BASIC_LXM051	BASIC_LXM05 (V1.0)	None

Used	Address	Symbol	Object
☑	%IWC1.0.0	D_STATUS_BASIC_ATV311	Drivecom status register
☑	%IWC1.0.1	D_CONTROL_BASIC_ATV311	Control effort
☑	%IWC1.0.2	D_IERROR_BASIC_ATV311	Error code
☑	%QWC1.0.0	D_COMMAND_BASIC_ATV311	Drivecom command reg
☑	%QWC1.0.1	D_TARGET_BASIC_ATV311	Target velocity

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>
Select the module "OTB_CAN_01" in the "Network" area and click on 🗑 to delete it:



CANopen Configuration Tool

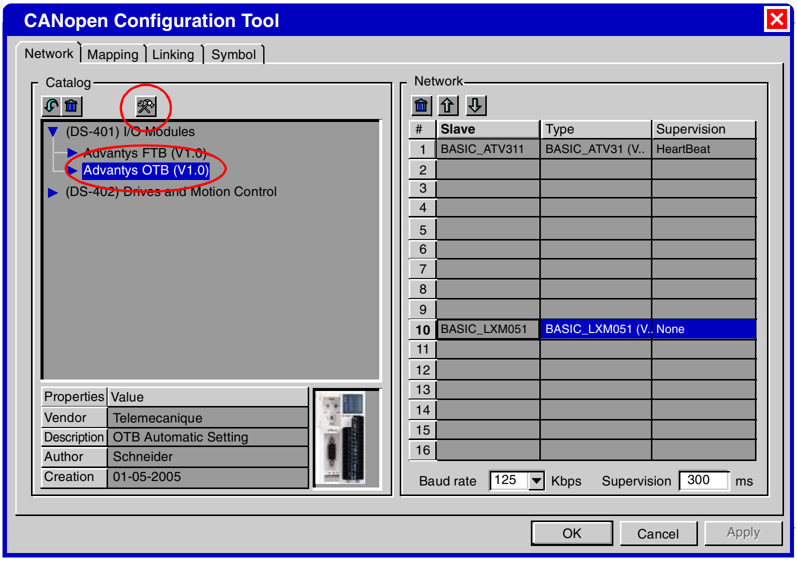
Catalog:
(DS-401) I/O Modules
Advantys FTB (V1.0)
Advantys OTB (V1.0)
(DS-402) Drives and Motion Control

<table>
<tr><th>Properties</th><th>Value</th></tr>
<tr><td>Vendor</td><td>Telemecanique</td></tr>
<tr><td>Description</td><td>OTB ICO DM9LP</td></tr>
<tr><td>Author</td><td>Schneider</td></tr>
<tr><td>Creation</td><td>01-05-2005</td></tr>
</table>

Network:

<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>BASIC_ATV31 (V..</td><td>HeartBeat</td></tr>
<tr><td>2</td><td>OTB_CAN_01</td><td>OTB_CAN_01 (V..</td><td>None</td></tr>
<tr><td>3</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>BASIC_LXM051</td><td>BASIC_LXM05 (V..</td><td>None</td></tr>
<tr><td>11</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td></td><td></td><td></td></tr>
</table>

Baud rate 125 Kbps Supervision 300 ms
</td></tr>
<tr><td>3</td><td>
Expand the tree of the I/O Modules (DS-401) profile, select "Advantys OTB (V1.0)" and click on 🛠 to launch configuration and addition of a new element:



CANopen Configuration Tool

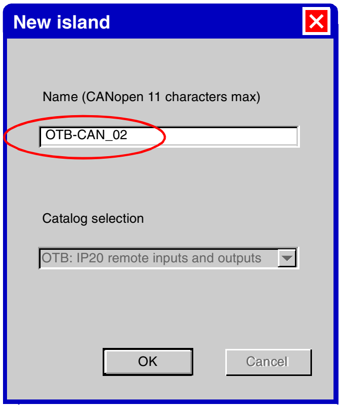
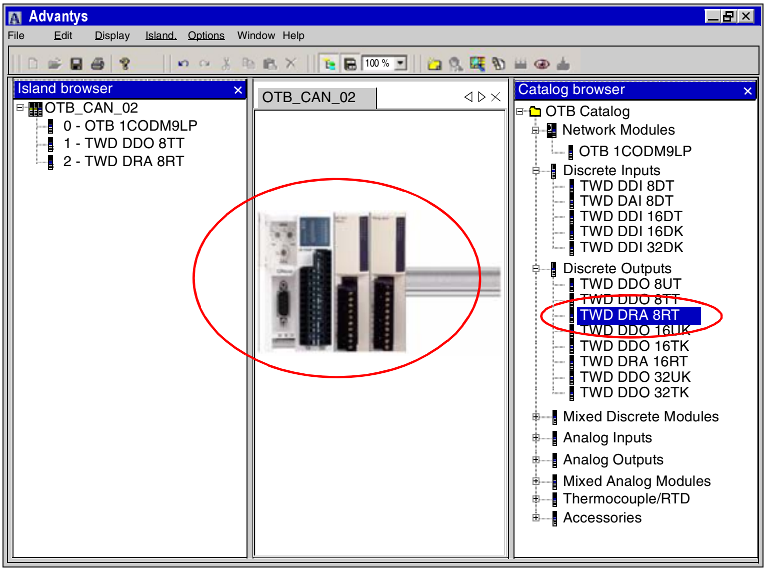
Catalog:
(DS-401) I/O Modules
Advantys FTB (V1.0)
Advantys OTB (V1.0)
(DS-402) Drives and Motion Control

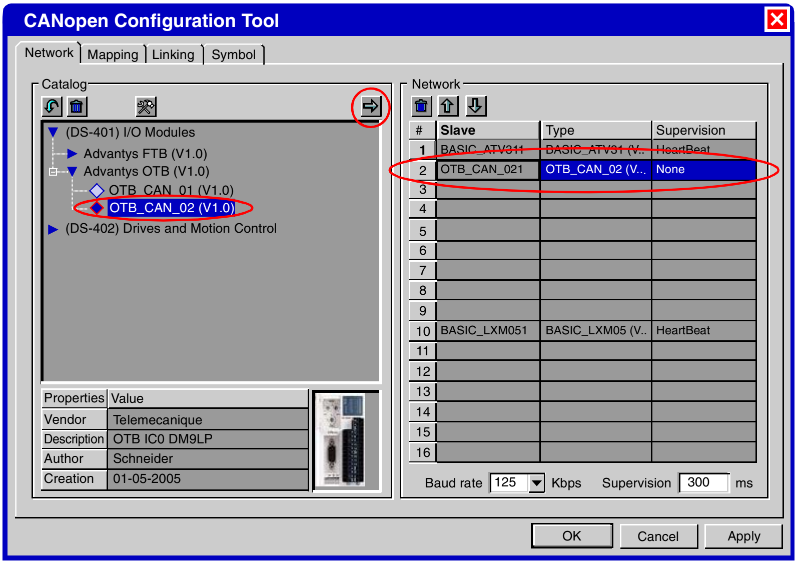
<table>
<tr><th>Properties</th><th>Value</th></tr>
<tr><td>Vendor</td><td>Telemecanique</td></tr>
<tr><td>Description</td><td>OTB Automatic Setting</td></tr>
<tr><td>Author</td><td>Schneider</td></tr>
<tr><td>Creation</td><td>01-05-2005</td></tr>
</table>

Network:

<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>BASIC_ATV31 (V..</td><td>HeartBeat</td></tr>
<tr><td>2</td><td></td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>BASIC_LXM051</td><td>BASIC_LXM051 (V..</td><td>None</td></tr>
<tr><td>11</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td></td><td></td><td></td></tr>
</table>

Baud rate 125 Kbps Supervision 300 ms
</td></tr>
</table>

Step	Action
4	In the "New Island" window, enter "OTB_CAN_02" matching the name of the equipment that will be used in TwidoSuite. Next click on the "OK" button to confirm and launch the Advantys Configuration Tool application:  New island Name (CANopen 11 characters max) OTB-CAN_02 Catalog selection OTB: IP20 remote inputs and outputs
5	Double click on the "OTB1C0DM9LP" I/O module to add it in the new island: Do the same with the "TWDDDO8TT" and "TWDDRA8RT" expansion modules to add them to the new island:  Advantys File Edit Display Island Options Window Help 100 % Island browser OTB_CAN_02 0 - OTB 1CODM9LP 1 - TWD DDO 8TT 2 - TWD DRA 8RT OTB_CAN_02 Catalog browser OTB Catalog Network Modules OTB 1CODM9LP Discrete Inputs TWD DDI 8DT TWD DAI 8DT TWD DDI 16DT TWD DDI 16DK TWD DDI 32DK Discrete Outputs TWD DDO 8UT TWD DDO 8TT TWD DRA 8RT TWD DDO 16UK TWD DDO 16TK TWD DRA 16RT TWD DDO 32UK TWD DDO 32TK Mixed Discrete Modules Analog Inputs Analog Outputs Mixed Analog Modules Thermocouple/RTD Accessories

Step	Action
6	It is possible to double click on the image of each module to set the parameters (Mask, polarity, management of fallback modes, etc.). Perform a "File / Save". Perform a "File / Quit". The new island containing the Advantys OTB I/O module and its expansion modules is then added to the CANopen catalog under the name "OTB_CAN_02".
7	Select "OTB_CAN_02 (V1.0)" in the "Catalog" area and click on ⇨ to add the Advantys OTB in the "Network" area:  Note: The Advantys OTB is configured by default to the first free network address, i.e., the network address 2 in our example.

CANopen Configuration Tool

Catalog

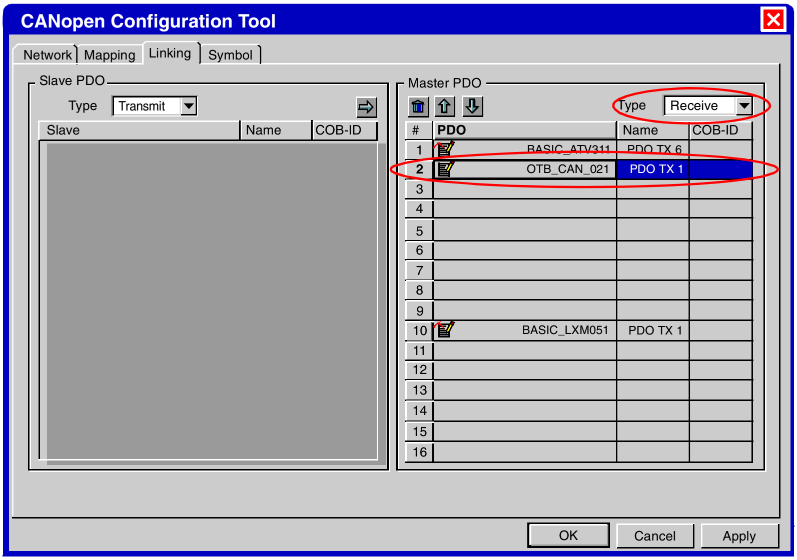
- (DS-401) I/O Modules
  - Advantys FTB (V1.0)
  - Advantys OTB (V1.0)
    - OTB_CAN_01 (V1.0)
    - OTB_CAN_02 (V1.0)
- (DS-402) Drives and Motion Control

Properties	Value
Vendor	Telemecanique
Description	OTB IC0 DM9LP
Author	Schneider
Creation	01-05-2005

Network

#	Slave	Type	Supervision
1	BASIC_ATV311	BASIC_ATV31 (V..	HeartBeat
2	OTB_CAN_021	OTB_CAN_02 (V..	None
3			
4			
5			
6			
7			
8			
9			
10	BASIC_LXM051	BASIC_LXM05 (V..	HeartBeat
11			
12			
13			
14			
15			
16			

Baud rate 125 Kbps Supervision 300 ms

Step	Action
8	Select the "Linking" tab. Use ⇨ to assign the new Advantys OTB I/O module and its two expansion modules to the PDO no. 2, in both "Receive" and "Send" modes:  CANopen Configuration Tool Slave PDO: Type Transmit Master PDO: Type Receive 1 BASIC_ATV311 PDO TX 6 2 OTB_CAN_021 PDO TX 1 10 BASIC_LXM051 PDO TX 1
9	Select the "Symbol" tab. Enter, if necessary a "symbol" for the objects associated with the Advantys OTB I/O module: ● Double click in the object's "Symbol" field, ● Type the name of the "Symbol", ● Confirm using the "ENTER" key of the keyboard.
10	Click on the "OK" button to confirm. The Advantys OTB I/O module and its expansion modules are then declared in the Twido controller application.

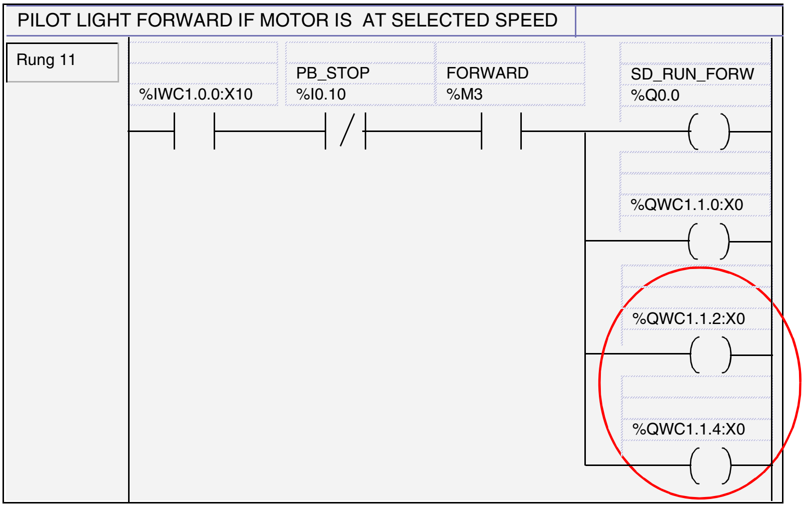
Non-reversible

The goal is to modify the application to manage the outputs of the Advantys OTB I/O module and of its expansion module the same as those of the Twido controller.

The method is identical to that described for the addition of an Advantys OTB by integrating the new outputs of the new island:

- %QWC1.1.0:X0 to X7 for the Advantys OTB "OTB1C0DEM9LP" module,
- %QWC1.1.2:X0 to X7 for the "TWDDO8TT" expansion module,
- %QWC1.1.4:X0 to X7 for the "TWDRA8RT" expansion module.

Here, for example, the line of the modified application managing the "FOWARD START" indicator LEDs:



In the TwidoSuite main window, select the task **Project** → **Save the current project**.

Load the new application into the Twido controller (see *Installing software and loading applications, p. 36*).

Add an Advantys FTB I/O splitter

V

At a Glance

Overview

This section of the document presents a possible evolution of the main system, that is the addition of an Advantys FTB I/O splitter.

Note: If you are not using a speed controller or Advantys OTB I/O module in your application, erase the associated elements in the CANopen configuration as well as in the Twido controller application program.

Application files

Files for this configuration may be found in CD-ROM BUNDLE (DIA3CD3050101F) in the "Applicative files\Partie_V" folder for the Twido controller application.

What's in this Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
11	Hardware implementation of an Advantys FTB splitter	105
12	Software implementation of an Advantys FTB splitter	109

Hardware implementation of an Advantys FTB splitter

11

At a Glance

Overview

This chapter describes hardware implementation of an Advantys FTB I/O splitter added to the main system.

What's in this Chapter?

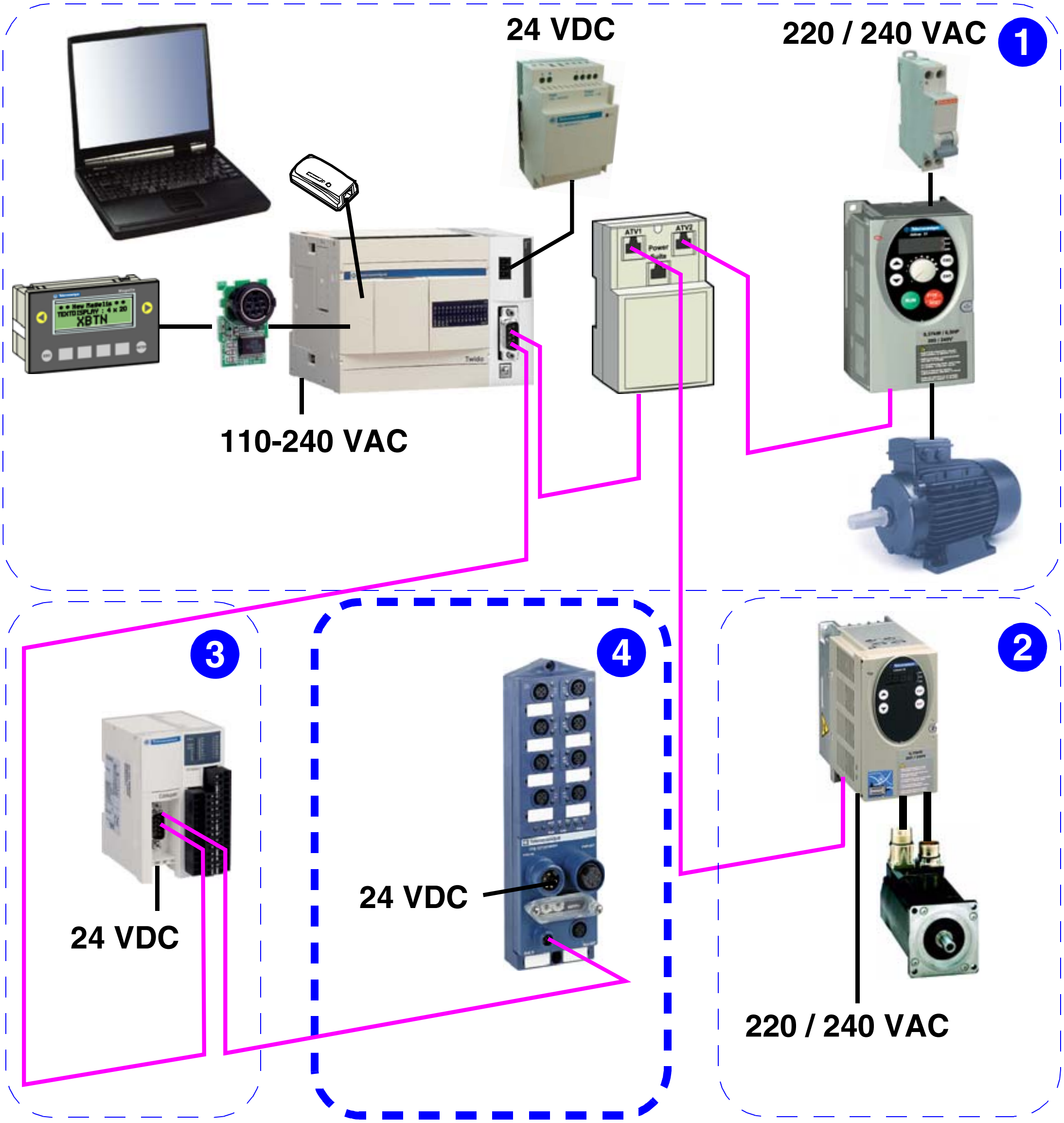
This chapter contains the following topics:

Topic	Page
Cabling of the Advantys FTB I/O splitter	106
Cabling of the CANopen network	107
Power supply	108

Cabling of the Advantys FTB I/O splitter

Installation diagram

This part of the document covers the following mounting:



Cabling of the CANopen network

Principle

The objective is to connect the Advantys FTB to the Advantys OTB using a cable.

Select a cable

To connect the Advantys FTB to the Advantys FTB, select one of the following cables:

Reference	Cable length (m)
FTXCN3230	3
FTXCN3250	5

Advantys FTB side cabling

The M12 end of the cable is connected to the Advantys FTB on the "BUS IN" base.

Advantys OTB side cabling

The "bare wire" end of cable is connected to the SUB-D 9 connector joined to the Advantys OTB (see *TAP-Twido Cable Preparation, p. 27*).

Cable pin assignment in the terminal of the SUB-D 9 connector:

No.	Signal	Cable	Connector	Wire color
5	CAN_H	FTB/OTB	CH2	White
6	CAN_L	FTB/OTB	CL2	Blue
7	GND	FTB/OTB	CG2	Black
8	V+	FTB/OTB	V+2	Red

Note: Toggle the line end terminator of the SUB-D 9 connector on the Advantys OTB to "OFF".
Connect the FTXCNTL12 line end terminator to the "BUS OUT" base of the Advantys FTB.

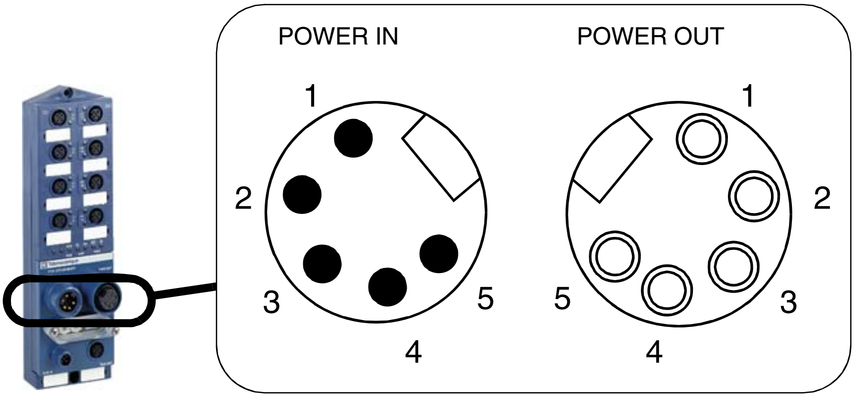
Power supply

Power on the Advantys FTB

Power on the Advantys FTB I/O splitter using one of the following cables:

Reference	Cable length (m)
FTXDP2115	1,5
FTXDP2130	3
FTXDP2150	5

Wiring Diagram:



Pin Assignment:

PIN No.	Description	Wire color	Printing on wire
1	0 V	Black	1
2	0 V	Black	2
3	PE	Green/Yellow	-
4	+24 V	Black	3
5	+24 V	Black	4

Software implementation of an Advantys FTB splitter

12

At a Glance

Overview

This chapter describes software implementation of an Advantys FTB I/O splitter added to the main system.

What's in this Chapter?

This chapter contains the following topics:

Topic	Page
Configuration of Advantys FTB communications	110
Add an Advantys FTB I/O splitter to the Twido controller application.	112

Configuration of Advantys FTB communications

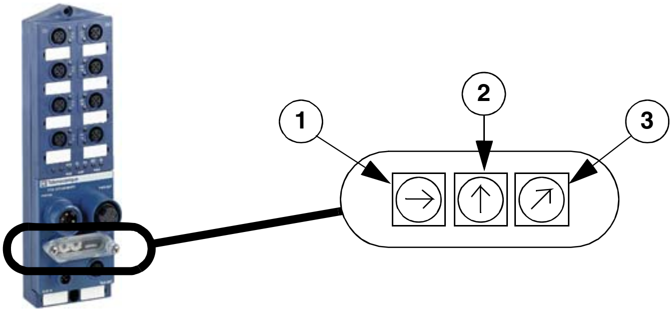
Principle

The system products must be configured according to the Twido application controller as follows:

- Address 1: ATV31
- Address 2: Advantys OTB
- Address 3: Advantys FTB
- Address 10: Lexium 05
- Transmission Speed: 125 kBits/s

Description

Front panel of the Advantys FTB I/O splitter:



No.	Function
1	Transmission speed encoder wheel
2	Network address (Node-ID x10) encoder wheel
3	Network address (Node-ID x1) encoder wheel

Method

Configuration of Advantys FTB communication parameters:

Step	Action
1	Cut off all power to the element.
2	Set encoder wheel 1 to the position corresponding to the required baud rate. In the example: encoder wheel 1 is set to position 5 (125 kBits/s)
3	Set encoder wheels 2 and 3 to the position corresponding to the desired network address. In the example, the address is 3 (10x0 + 1x3): ● encoder wheel 2: position 0 (tens) ● encoder wheel 3: position 3 (units)
4	Reapply power to the Advantys FTB to implement the new configuration.

Transmission speed configuration

Set the transmission speed using a turn switch.

The possible speeds are as follows:

Switch position	Transmission speed
0	Automatic
1	10 Kbps
2	20 Kbps
3	50 Kbps
4	100 Kbps
5	125 Kbps
6	250 Kbps
7	500 Kbps
8	800 Kbps
9	1000 Kbps

Add an Advantys FTB I/O splitter to the Twido controller application.

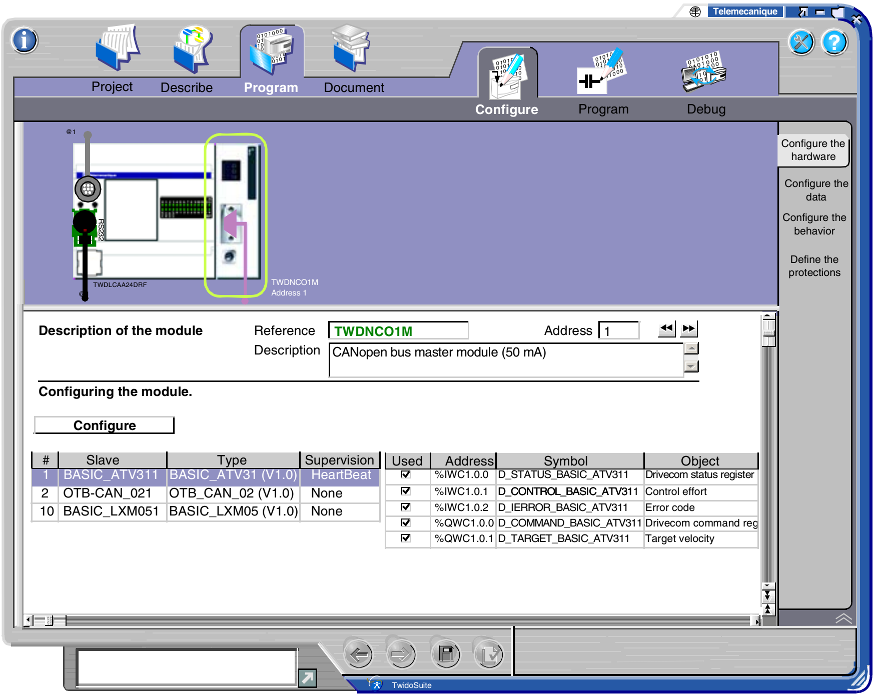
Principle

The objective is to modify the existing application in order to be able to manage the I/O of the Advantys FTB splitter.

Declare the Advantys FTB splitter

The following method allows declaration of the Advantys FTB I/O splitter in the Twido controller application.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Open the project corresponding to your hardware configuration using the TwidoSuite software (TwidoSuite must be in disconnected mode).
<ul>
<li>ATV31, LXM05, OTB without the expansion module: BUNDLE\Partie_IV\BUNDLE_CAN_Part_IV.xpr</li>
<li>ATV31, LXM05, OTB + expansion modules: BUNDLE\Partie_IV\BUNDLE_CAN_Part_IVb.xpr</li>
</ul>
In the TwidoSuite main window, select the task <b>Program</b> → <b>Configure</b> → <b>Configure hardware</b>.
Select the CANopen network TWDNCO1M:

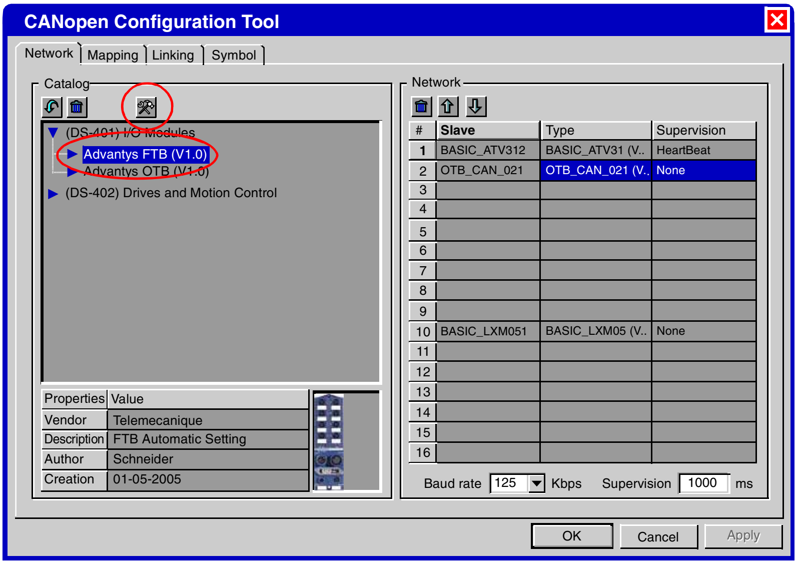


<b>Description of the module</b>
Reference TWDNCO1M
Address 1
Description CANopen bus master module (50 mA)
<b>Configuring the module.</b>
Configure

<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>BASIC_ATV31 (V1.0)</td><td>HeartBeat</td></tr>
<tr><td>2</td><td>OTB-CAN_021</td><td>OTB_CAN_02 (V1.0)</td><td>None</td></tr>
<tr><td>10</td><td>BASIC_LXM051</td><td>BASIC_LXM05 (V1.0)</td><td>None</td></tr>
</table>

<table>
<tr><th>Used</th><th>Address</th><th>Symbol</th><th>Object</th></tr>
<tr><td>☑</td><td>%IWC1.0.0</td><td>D_STATUS_BASIC_ATV311</td><td>Drivecom status register</td></tr>
<tr><td>☑</td><td>%IWC1.0.1</td><td>D_CONTROL_BASIC_ATV311</td><td>Control effort</td></tr>
<tr><td>☑</td><td>%IWC1.0.2</td><td>D_IERROR_BASIC_ATV311</td><td>Error code</td></tr>
<tr><td>☑</td><td>%QWC1.0.0</td><td>D_COMMAND_BASIC_ATV311</td><td>Drivecom command reg</td></tr>
<tr><td>☑</td><td>%QWC1.0.1</td><td>D_TARGET_BASIC_ATV311</td><td>Target velocity</td></tr>
</table>

Click <b>Configure</b>.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>
Expand the tree of the I/O Modules (DS-401) profile, select "Advantys FTB (V1.0)" and click on 🛠 to launch configuration and addition of a new element:


CANopen Configuration Tool

Network | Mapping | Linking | Symbol

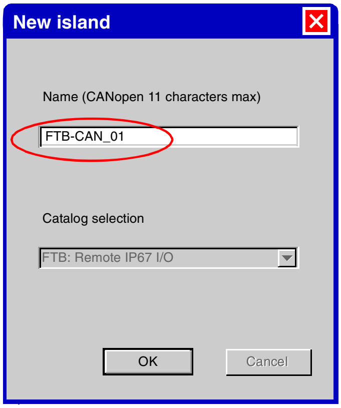
Catalog

▼ (DS-401) I/O Modules

▶ Advantys FTB (V1.0)

▶ Advantys OTB (V1.0)

▶ (DS-402) Drives and Motion Control
<table>
<tr><th>Properties</th><th>Value</th></tr>
<tr><td>Vendor</td><td>Telemecanique</td></tr>
<tr><td>Description</td><td>FTB Automatic Setting</td></tr>
<tr><td>Author</td><td>Schneider</td></tr>
<tr><td>Creation</td><td>01-05-2005</td></tr>
</table>
Network
<table>
<tr><th>#</th><th>Slave</th><th>Type</th><th>Supervision</th></tr>
<tr><td>1</td><td>BASIC_ATV312</td><td>BASIC_ATV31 (V..</td><td>HeartBeat</td></tr>
<tr><td>2</td><td>OTB_CAN_021</td><td>OTB_CAN_021 (V..</td><td>None</td></tr>
<tr><td>3</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>BASIC_LXM051</td><td>BASIC_LXM05 (V..</td><td>None</td></tr>
<tr><td>11</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td></td><td></td><td></td></tr>
</table>
Baud rate 125 Kbps Supervision 1000 ms

OK Cancel Apply
</td></tr>
<tr><td>3</td><td>
In the "New Island" window, enter "FTB_CAN_01" matching the name of the equipment that will be used in TwidoSuite. Next click on the "OK" button to confirm and launch the Advantys Configuration Tool application:


New island

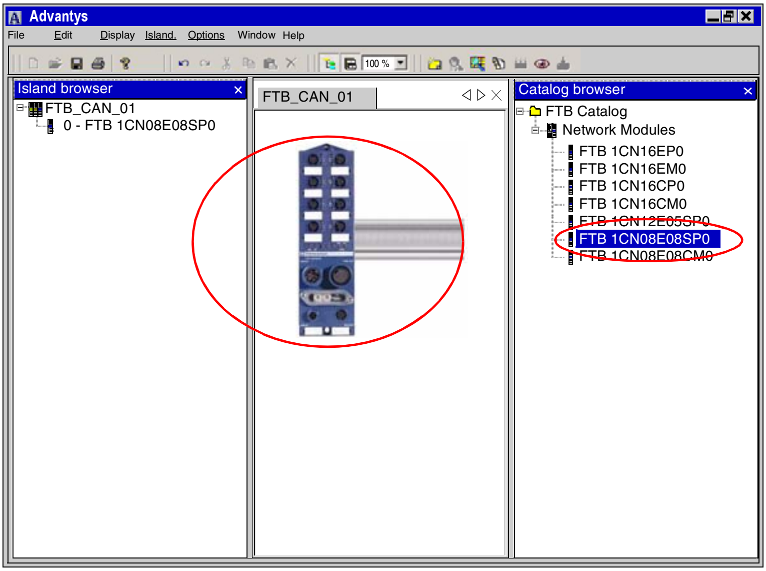
Name (CANopen 11 characters max)

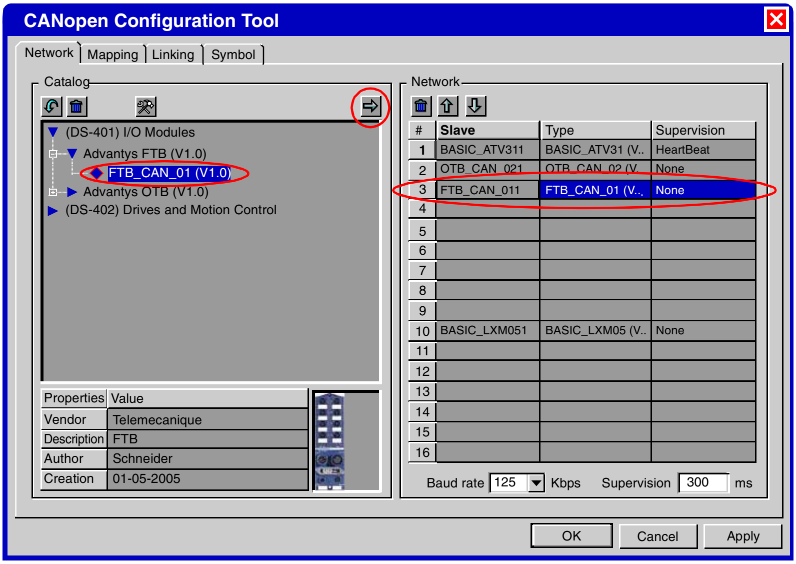
FTB-CAN_01

Catalog selection

FTB: Remote IP67 I/O

OK Cancel
</td></tr>
</table>

Step	Action
4	Double click on the Advantys FTB I/O "FTB1CN08E08SP0" splitter to add it in the new island:  Advantys File Edit Display Island. Options Window Help 100 % Island browser FTB_CAN_01 0 - FTB 1CN08E08SP0 FTB_CAN_01 Catalog browser FTB Catalog Network Modules FTB 1CN16EP0 FTB 1CN16EM0 FTB 1CN16CP0 FTB 1CN16CM0 FTB 1CN12E05SP0 FTB 1CN08E08SP0 FTB 1CN08E08CM0
5	It is possible to double click on the image of the splitter to set the parameters (Function, Mask, polarity, management of fallback modes, etc.). Perform a "File / Save". Perform a "File / Quit". The new island containing the Advantys FTB I/O splitter is then added to the CANopen catalog under the name "FTB_CAN_01".

Step	Action
6	Select "FTB_CAN_01 (V1.0)" in the "Catalog" area and click on ⇨ to add the Advantys FTB splitter in the "Network" area.  Note: The Advantys FTB is configured by default to the first free network address, i.e., the network address 3 in our example.

CANopen Configuration Tool

Catalog

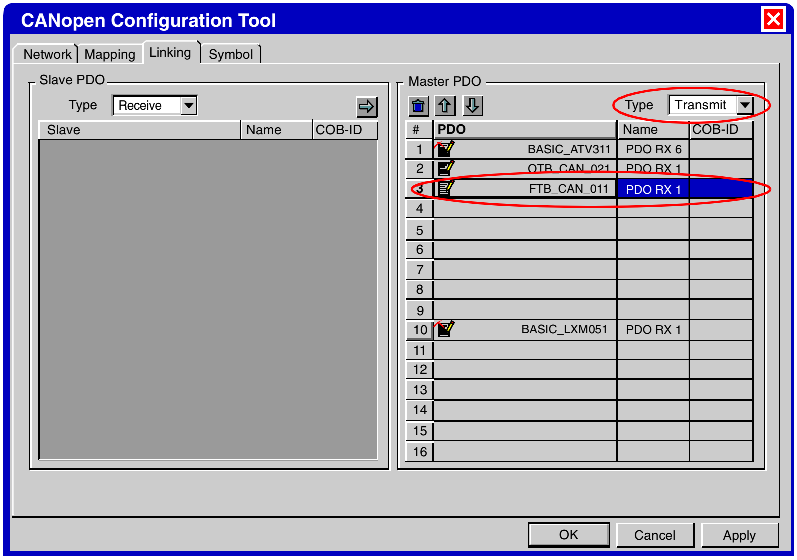
- (DS-401) I/O Modules
  - Advantys FTB (V1.0)
    - FTB_CAN_01 (V1.0)
  - Advantys OTB (V1.0)
- (DS-402) Drives and Motion Control

Properties	Value
Vendor	Telemecanique
Description	FTB
Author	Schneider
Creation	01-05-2005

Network

#	Slave	Type	Supervision
1	BASIC_ATV311	BASIC_ATV31 (V..	HeartBeat
2	OTB_CAN_021	OTB_CAN_02 (V..	None
3	FTB_CAN_011	FTB_CAN_01 (V..	None
4			
5			
6			
7			
8			
9			
10	BASIC_LXM051	BASIC_LXM05 (V..	None
11			
12			
13			
14			
15			
16			

Baud rate 125 Kbps Supervision 300 ms

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>7</td><td>Select the "Linking" tab.
Use ⇨ to assign the new Advantys FTB splitter to the PDO no. 3, in both "Receive" and "Send":


Slave PDO Type: Receive
Master PDO Type: Transmit
<table>
<tr><th>#</th><th>PDO</th><th>Name</th><th>COB-ID</th></tr>
<tr><td>1</td><td>BASIC_ATV311</td><td>PDO RX 6</td><td></td></tr>
<tr><td>2</td><td>OTB_CAN_021</td><td>PDO RX 1</td><td></td></tr>
<tr><td>3</td><td>FTB_CAN_011</td><td>PDO RX 1</td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>BASIC_LXM051</td><td>PDO RX 1</td><td></td></tr>
<tr><td>11</td><td></td><td></td><td></td></tr>
<tr><td>12</td><td></td><td></td><td></td></tr>
<tr><td>13</td><td></td><td></td><td></td></tr>
<tr><td>14</td><td></td><td></td><td></td></tr>
<tr><td>15</td><td></td><td></td><td></td></tr>
<tr><td>16</td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td>8</td><td>Select the "Symbol" tab.
Enter, if necessary, a "symbol" for the objects associated with the Advantys FTB I/O splitter:
● Double click in the object's "Symbol" field,
● Type the name of the "Symbol",
● Confirm using the "ENTER" key of the keyboard.</td></tr>
<tr><td>9</td><td>Click on the "OK" button to confirm. The Advantys FTB I/O splitter is then declared in the Twido controller application.</td></tr>
</table>

Non-reversible

The goal is to change the application to control the I/O of the Advantys FTB splitter the same as those of the Twido controller.

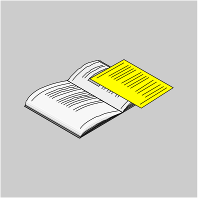
The method is identical to the one described for adding an Advantys OTB by integrating to the application the new I/O of the Advantys FTB splitter:

- %IWC1.2.0:X0 to X7 for inputs,
- %QWC1.2.0:X0 to X7 for the outputs.

In the TwidoSuite main window, select the task **Project** → **Save the current project**.

Load the new application into the Twido controller (see *Installing software and loading applications, p. 36*).

Appendices



At a Glance

List of symbols

The following pages contain the list of symbols used in the application described in part V of this document.

What's in this Appendix?

The appendix contains the following chapters:

Chapter	Chapter Name	Page
A	List of symbols used in the application	121

List of symbols used in the application

A

List of symbols used in the application

List of symbols

In the TwidoSuite main window, select the task **Program** → **Program** → **Define Symbols**.

Here, for example, is the list of symbols used in the application described in part V of this document:

Symbol	Number	Comment
LXM POWER	%I0.0	
START_POSITION	%I0.1	
PB_START_FORWARD	%I0.8	Twido_input: start drive Forward
PB_START_REVERSE	%I0.9	Twido_Input: Start drive Reverse
PB_STOP	%I0.10	Twido_Input: Stop drive
PB_SLOW_FAST	%I0.11	Twido_Input: select speed drive SLOW or FAST
RESET_ERROR	%I0.13	
D_STATUS_BASIC_ATV311	%IWC1.0.0	
D_CONTROL_BASIC_ATV311	%IWC1.0.1	
D_IERROR_BASIC_ATV311	%IWC1.0.2	
OTB_CAN_INPUTS_1	%IWC1.1.0	
OTB_CAN_INPUTS_2	%IWC1.1.1	
FTB_CAN_INPUTS_1	%IWC1.2.0	
STATUS_PDO1_BASIC_LXM051	%IWCD1.9.0	
STATUS_PDO2_BASIC_LXM051	%IWCD1.9.2	
M_POSITION_BASIC_LXM051	%IWCD1.10.0	
M_VELOCITY_BASIC_LXM051	%IWCD1.10.2	
FORWARD	%M3	
REVERSE	%M4	
PB_12_RF	%M12	

Symbol	Number	Comment
D_STATE_0	%MW0	Altivar state
D_CANSTATE_0	%MW1	Altivar CANopen state
D_ERROR_0	%MW2	Altivar error code
D_SETPOINT_MODE_0	%MW3	Altivar set-point mode
D_SETPOINT_0	%MW4	Altivar set-point
D_SELECT_SPEED_VAL_0	%MW17	Control effort on the Altivar
D_MODBUS_INIT_PHASE_0	%MW28	Modbus initialisation phase running Bit0
SD_SPEED	%MW104	Speed entered with the XBT or selected by OTB input %IWC1.1.1:X3
XBT_FTB_OUTPUTS	%MW105	Used to display information on the XBT
XBT_FTB_INPUTS	%MW106	Used to display information on the XBT
XBT_OTB_OUTPUTS	%MW107	Used to display information on the XBT
XBT_OTB_INPUTS	%MW108	Used to display information on the XBT
XBT_DIALOG_TABLE_ALARM	%MW110	Used to manage Alarm display on the XBTN400
XBT_TWIDO_INPUTS	%MW120	
XBT_TWIDO_OUTPUTS	%MW121	
CAN_SLAVE_STATUS_2_1	%MW200	Status for slave 2 (MSB) and 1 (LSB) / system words %SW20
CAN_SLAVE_STATUS_4_3	%MW201	Status for slave 4 (MSB) and 3 (LSB) / system words %SW21
CAN_OPEN_SLAVE_STATUS_XBT_1	%MW202	CAN Status for slave 1
CAN_OPEN_SLAVE_STATUS_XBT_2	%MW203	CAN Status for slave 2
CAN_OPEN_SLAVE_STATUS_XBT_3	%MW204	CAN Status for slave 3
M_AXIS_NB_1	%MW502	Lexium Address
M_AXIS_STATUS_1	%MW503	Active state of axis control graph
M_ERROR_LD_1	%MW504	Lexium error code
M_MVT_TYPE_1	%MW505	Actual mvt of servo drive
M_MODE_DISPLAY_1	%MW506	Actual mode of operation active
M_GEAR_REF_1	%MW507	Operating mode of electronic gear processing
M_STATE_1	%MW509	Actual mode of operation active
M_POSITION_REF_LO_1	%MW512	Reference position for homing method 35 low bits
M_POSITION_REF_HI_1	%MW513	Reference position for homing method 35 high bits
M_GEAR_NUM_LO_1	%MW514	Gear Numerator low bits
M_GEAR_NUM_HI_1	%MW515	Gear Numerator high bits
M_HOMING_METHOD_1	%MW529	Reference movement method

Symbol	Number	Comment
M_TARGET_VELOCITY_1	%MW530	Speed reached
M_GEAR_DENOM_1	%MW531	Gear Denominator low bits
M_TARGET_POSITION_LO_1	%MW532	position reached low bits
M_TARGET_POSITION_HI_1	%MW533	position reached high bits
M_TARGET_CURRENT_1	%MW534	Current reached
SD_RUN_FORW	%Q0.0	Twido_Output: speed drive is running forward
SD_RUN_REV	%Q0.1	Twido_Output: speed drive is running reverse
SD_STOPPED	%Q0.2	Twido_Output: speed drive is stopped
MOTOR_IS_GOING_TO_POINT	%Q0.3	
D_COMMAND_BASIC_ATV311	%QWC1.0.0	
D_TARGET_BASIC_ATV311	%QWC1.0.1	
OTB_CAN_OUTPUTS_1	%QWC1.1.0	
OTB_CAN_OUTPUTS_2	%QWC1.1.1	
OTB_CAN_OUTPUTS_3	%QWC1.1.2	
OTB_CAN_OUTPUTS_4	%QWC1.1.3	
OTB_CAN_OUTPUTS_5	%QWC1.1.4	
OTB_CAN_OUTPUTS_6	%QWC1.1.5	
FTB_CAN_OUTPUTS_1	%QWC1.2.0	
CONTROL_PDO1_BASIC_LXM051	%QWCD1.9.0	
CONTROL_PDO2_BASIC_LXM051	%QWCD1.9.2	