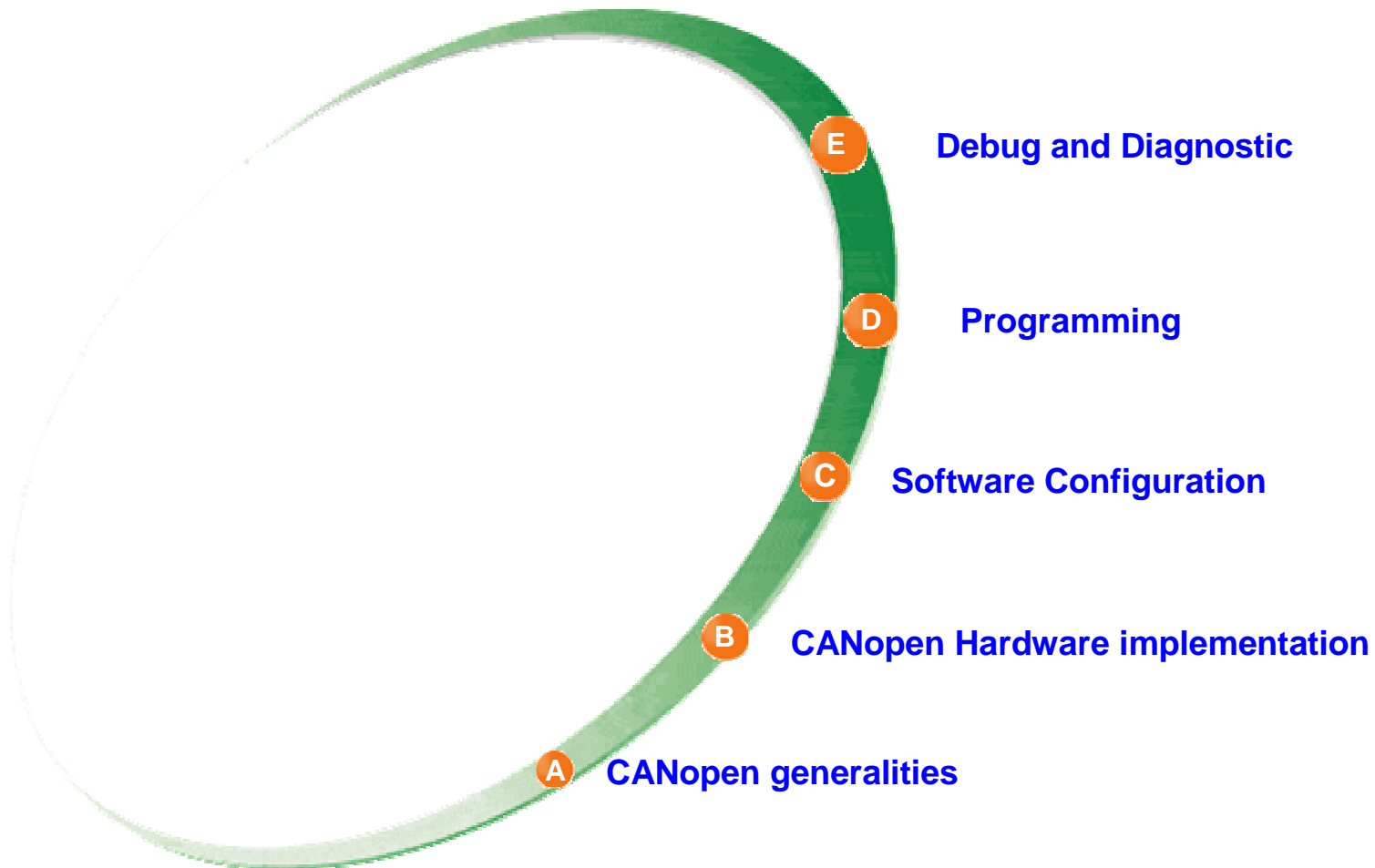
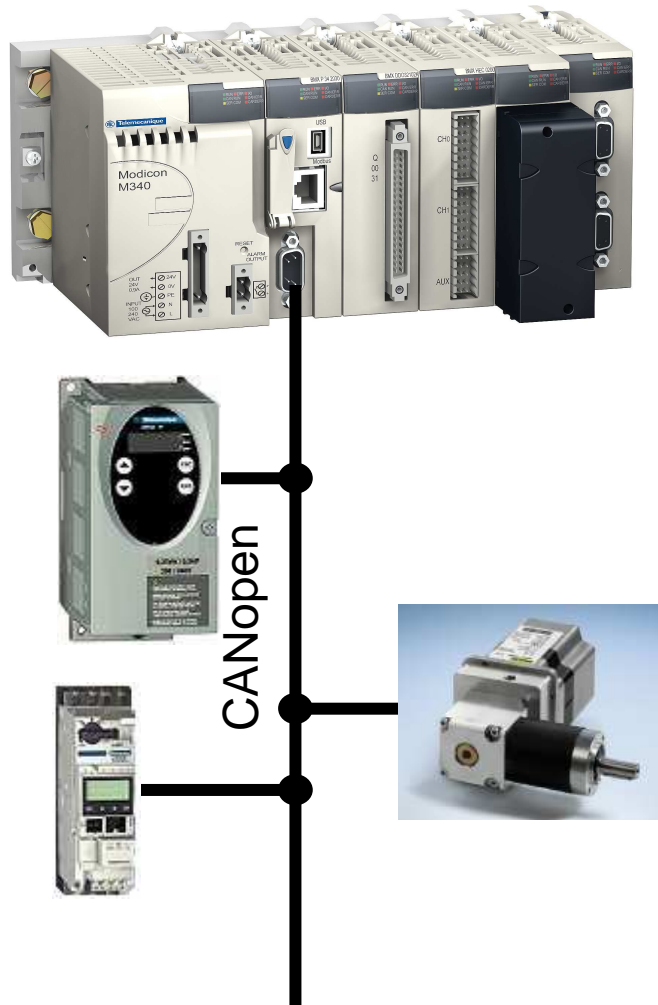


M7 - Modicon M340 : CANopen field bus



A – CANOpen generalities

CANopen architecture overview

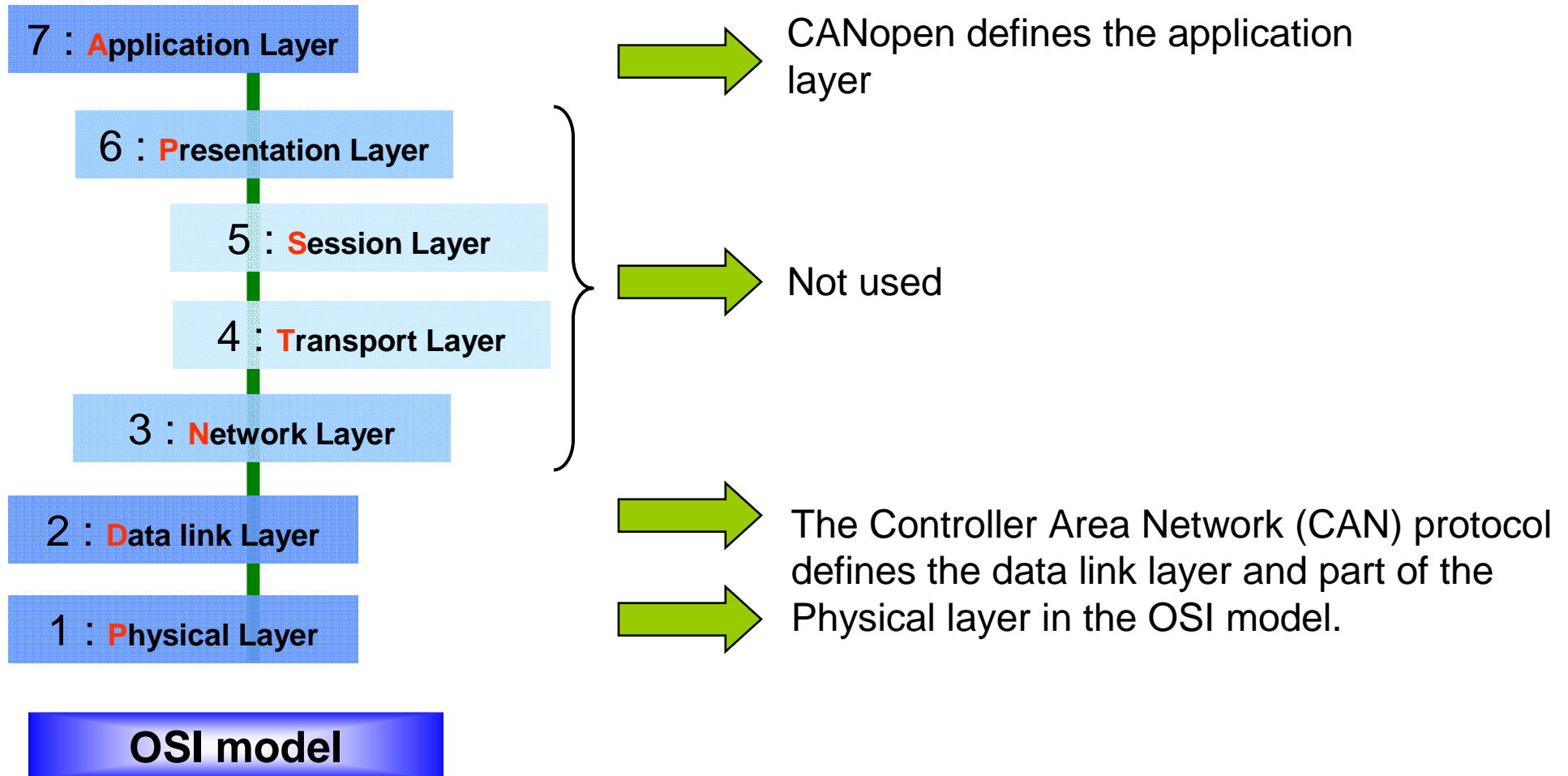


- Canopen field bus is used to exchange data with equipment connected to the bus.
 - Implicit exchanges (through **%QW**, **%IW**)
 - Explicit exchanges (through **write/read var**)
- CANopen field bus is available on some processors equipped with an integrated CANopen port.
 - BMX P34 2010 / 2030 processors
 - SubD 9 points connector as physical interface

Characteristic of integrated channel

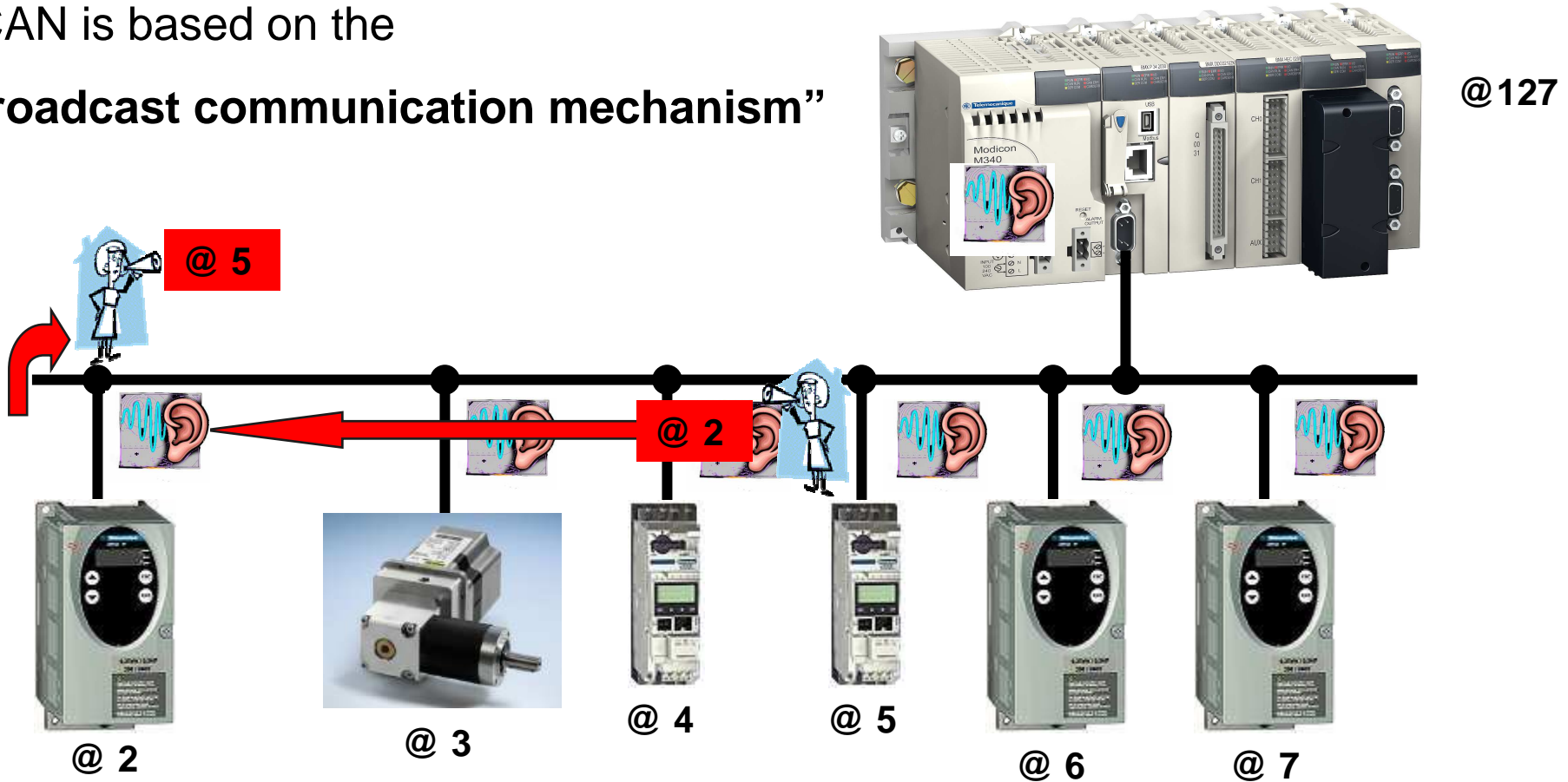
	Character mode												
Baud rate	20, 50, 125, 250, 500, 1000 kbds (according with the lenght)												
Equipments	63 Equipments max												
Bus length	0 to 2500 meters												
Services	NMT, PDO, SDO, SYNC, EMCY												
Speed / Lenght (the speed is directly linked with the bus's length)	<table> <tr> <td>1 Mbits</td><td>20 m</td></tr> <tr> <td>500 Kbits</td><td>100 m</td></tr> <tr> <td>250 Kbits</td><td>250 m</td></tr> <tr> <td>125 Kbits</td><td>500 m</td></tr> <tr> <td>50 Kbits</td><td>1000 m</td></tr> <tr> <td>20 Kbits</td><td>2500 m</td></tr> </table>	1 Mbits	20 m	500 Kbits	100 m	250 Kbits	250 m	125 Kbits	500 m	50 Kbits	1000 m	20 Kbits	2500 m
1 Mbits	20 m												
500 Kbits	100 m												
250 Kbits	250 m												
125 Kbits	500 m												
50 Kbits	1000 m												
20 Kbits	2500 m												

OSI model dedicated for the CAN bus



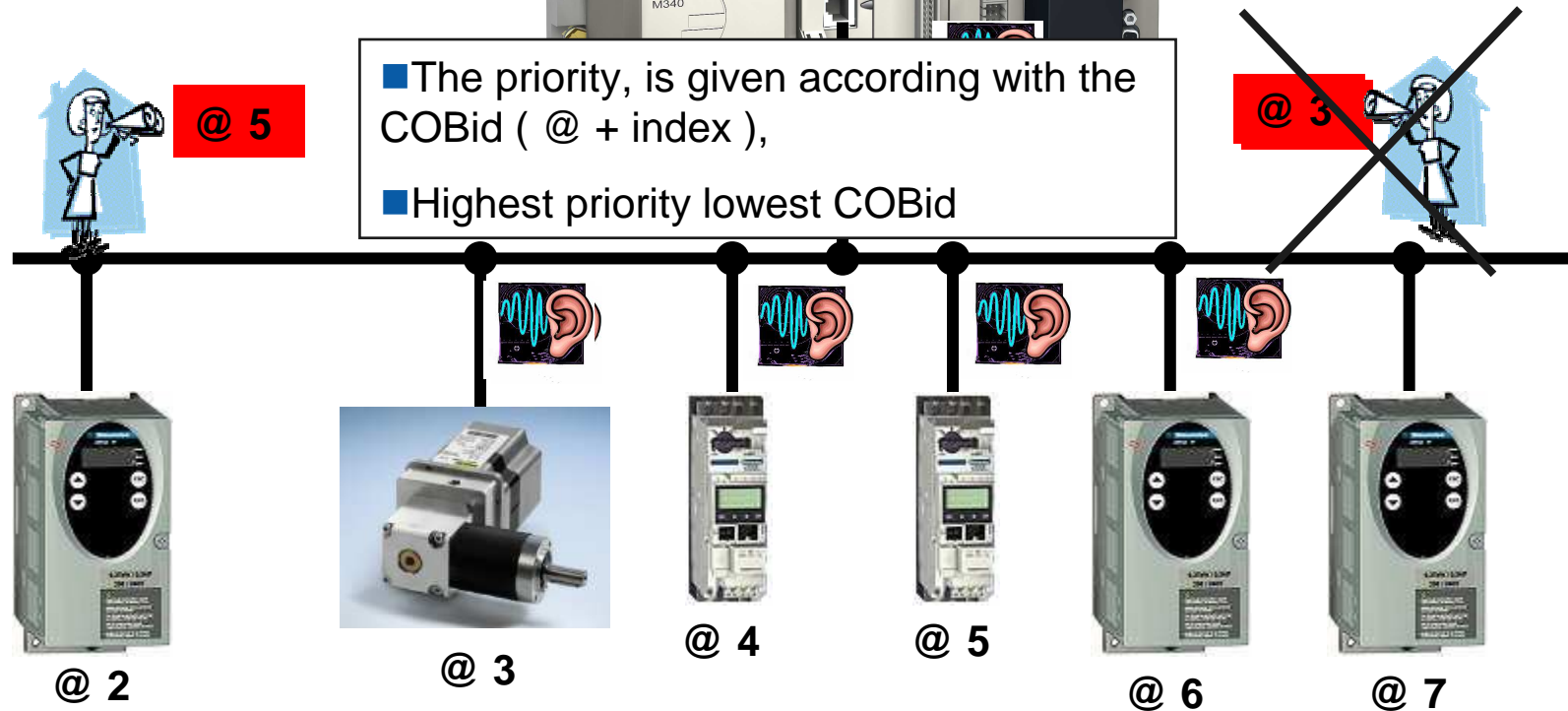
Communication protocol principle

■ CAN is based on the
“broadcast communication mechanism”



Communication protocol principle

Priority : The identifier with the lowest COBid number has the highest priority.



Communication protocol principle



Priority : The identifier with the lowest COBId number has the highest priority.

■ The priority, is given according with the

@ 5

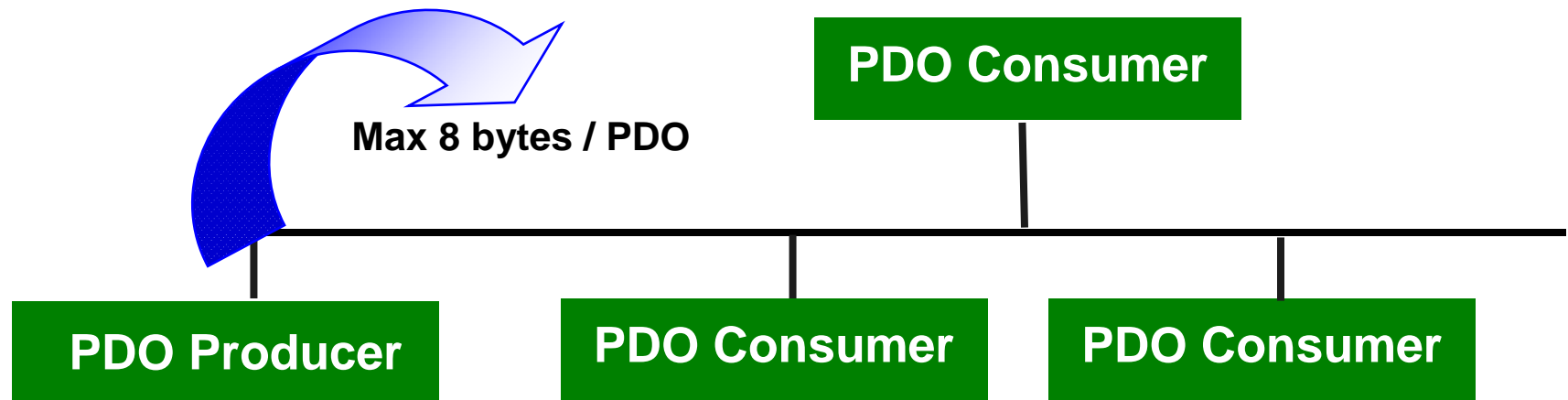
Communications object	Function code	Node address-node-Id [1...127]	COB-IDdecimal (hexadecimal)
NMT Start/Stop Service	0 0 0 0	0 0 0 0 0 0 0	0 (0 _h)
SYNC object	0 0 0 1	0 0 0 0 0 0 0	128 (80 _h)
EMCY object	0 0 0 1	x x x x x x x	128 (80 _h) + node-Id
T_PDO1	0 0 1 1	x x x x x x x	384 (180 _h) + node-Id
R_PDO1	0 1 0 0	x x x x x x x	512 (200 _h) + node-Id
T_PDO2	0 1 0 1	x x x x x x x	640 (280 _h) + node-Id
R_PDO2	0 1 1 0	x x x x x x x	768 (300 _h) + node-Id
T_PDO3	0 1 1 1	x x x x x x x	896 (380 _h) + node-Id
R_PDO3	1 0 0 0	x x x x x x x	1024 (400 _h) + node-Id
T_PDO4	1 0 0 1	x x x x x x x	1152 (480 _h) + node-Id
R_PDO4	1 0 1 0	x x x x x x x	1280 (500 _h) + node-Id
T_SDO	1 0 1 1	x x x x x x x	1408 (580 _h) + node-Id
NMT error control	1 1 1 0	x x x x x x x	1792 (700 _h) + node-Id
R_SDO	1 1 0 0	x x x x x x x	1536 (600 _h) + node-Id

@ 2

PDO (**P**rocess **D**ata **O**bjects)

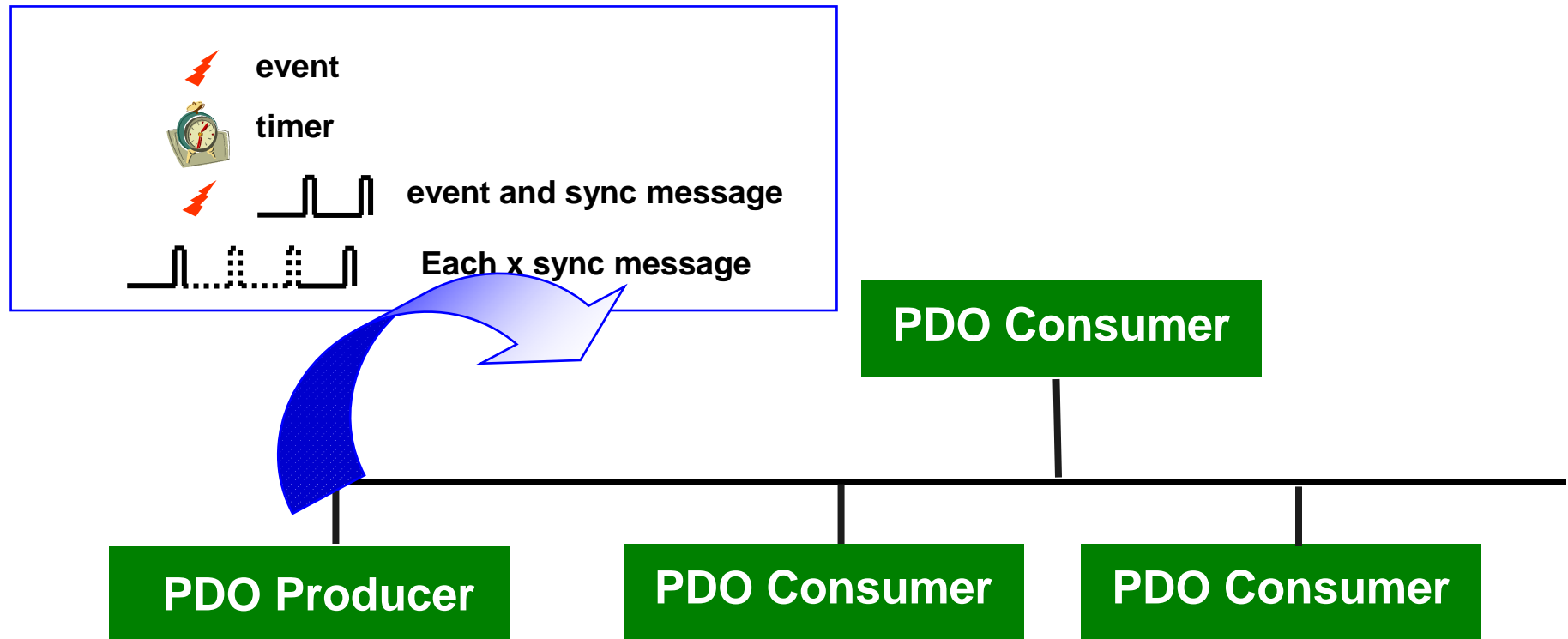
■ PDOs are implicit exchanges between a Producer and one or more Consumer (For this exchange M340 uses the %QW and %IW objects)

■ PDOs (Process Data Objects) are mapped to a single CAN frame using data 8 bytes max.



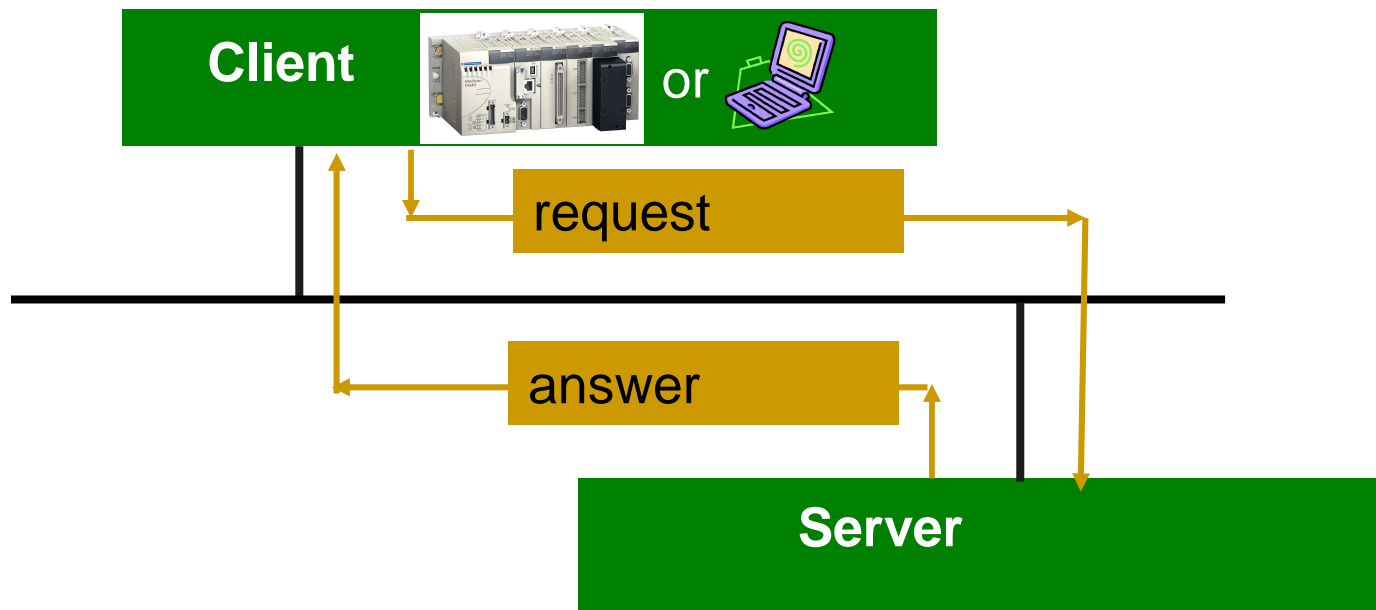
PDO (**P**rocess **D**ata **O**bjects)

■ PDOs exchange may be driven by an internal event, by an internal timer, by remote request or by the Sync message .



SDO (Service Data Object)

- SDOs are explicit exchanges and use the READ_VAR and WRITE_VAR commands
- SDO (Service Data Object) reads or writes the Objects predefined in the Dictionary (EDS file).
- The SDO transport protocol allows transmitting objects of any size.



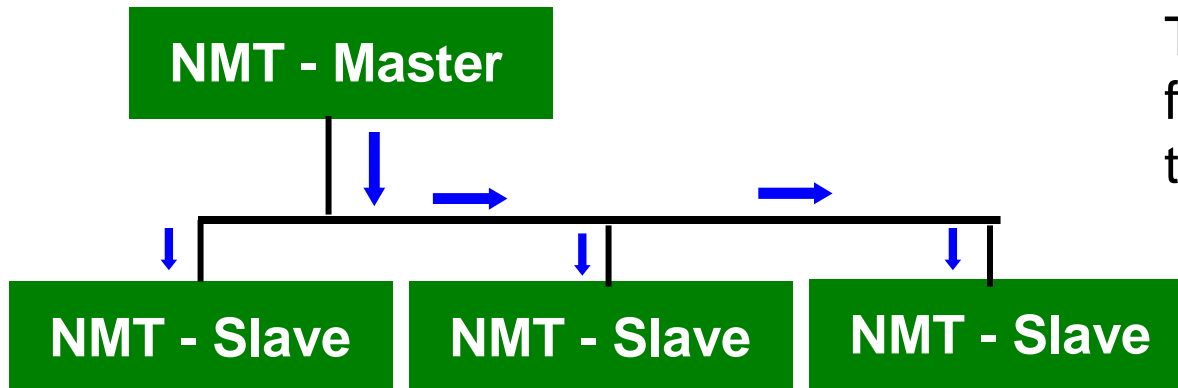
NMT (Network Management)

- The NMT (Network management) include three services:

1. NMT message
2. Boot up message
3. The Error Control service

1. The NMT message :

Transmitted by the NMT master forces one node or all the nodes to transit to another NMT state.



NMT (Network Management)

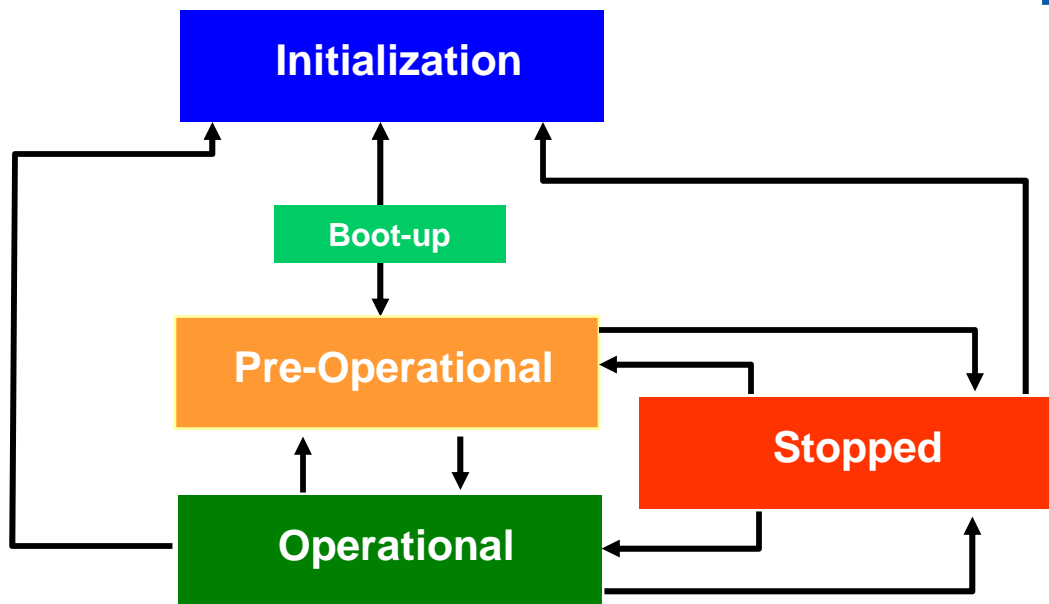
- The NMT (Network management) include three services :

1. NMT message
2. Boot up message
3. The Error Control service

1. The NMT message :

The CANopen state machine (NMT state) specifies the states :

- Initialization
- Pre-Operational SDO allowed
- Operational SDO and PDO allowed
- Stopped. SDO and PDO are not allowed



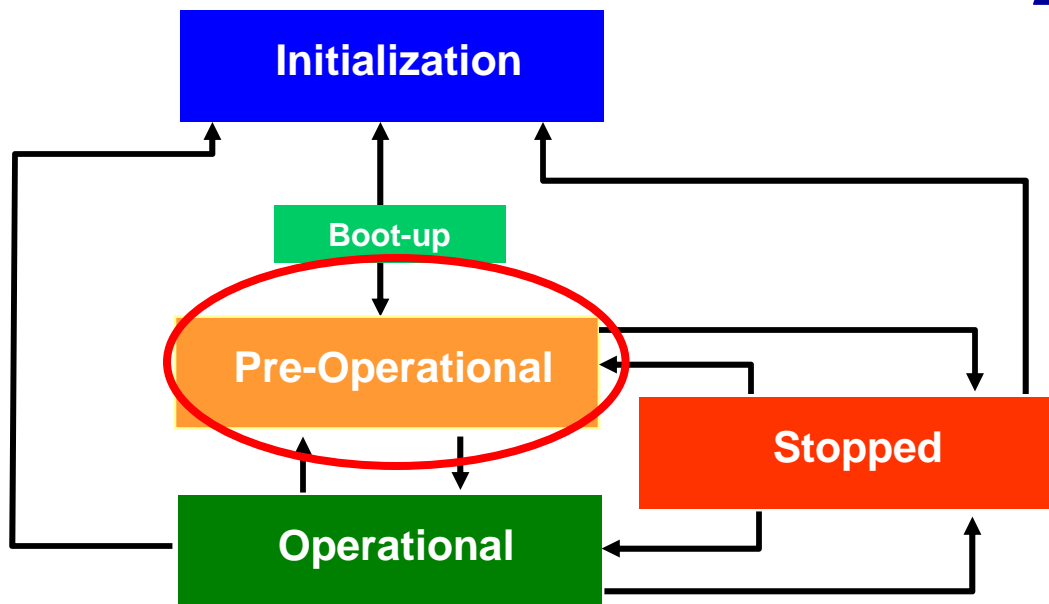
NMT (Network Management)

- The NMT (Network management) include three services :

1. NMT message
2. Boot up message
3. The Error Control service

2. The BOOT UP message

A device sends the **Boot-up message** to indicate to the NMT master that it has reached the state Pre-operational.



NMT (Network Management)

■ The NMT (Network management) include three services :

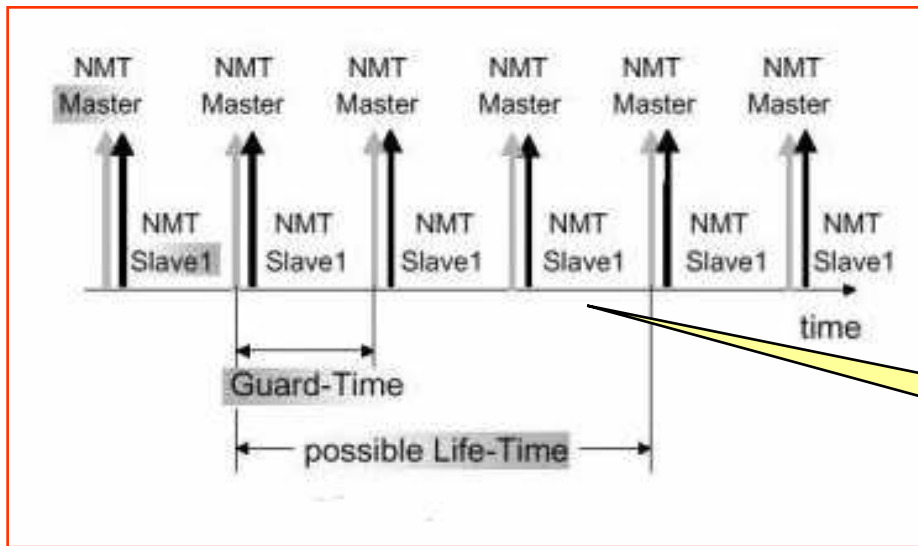
1. NMT message
2. Boot up message
3. The Error Control service

3. The Error Control service

Through this service the NMT detect the error on the field bus
The Error control service can manage :

The Heartbeating protocol

Periodically the master send a request to the slave in order to know its state



EDS file (**E**lectrical **D**ata **S**heets)

■ **Goal :** The EDS file describes the capabilities of CANopen nodes:

- General information about the node.
- Informations of all dictionary object of the node.
- Configuration of the PDO communication.

The EDS file is an ASCII file, and can be acquired from the manufacturer of the device.

■ **Ex :** Part of the MFBLXM15.eds

```
[FileInfo]
FileName=MFBLXM15L1.eds
FileVersion=1
FileRevision=5
EDSVersion=4.0
Description=LEXIUM15LP1 servodrive on MFB library
CreationTime=9:00AM
CreationDate=06-21-2005
CreatedBy=V&C
ModificationTime=00:40AM
ModificationDate=02-14-2006
ModifiedBy=PhB
; révision 14/02/2006 default values for MFB
; *****
; *****

[DeviceInfo]
VendorName=Telemecanique
VendorNumber=0x02000005A
ProductName=MFB_LEXIUM15LP
ProductNumber=0
RevisionNumber=1
```

```
[1A00]
SubNumber=0x3
ParameterName=1st transmit PDO-Mapping
ObjectType=0x9

[1A00sub0]
ParameterName=Number of Entries
ObjectType=0x7
DataType=0x0005
AccessType=ro
DefaultValue=0x2
PDOMapping=0

[1A00sub1]
ParameterName=Position actual value in user units
ObjectType=0x7
DataType=0x0007
AccessType=ro
DefaultValue=0x60640020
PDOMapping=0
```

Mapping of
the
Transmit
PDO1

B – Hardware implementation

Processors with an integrated CANopen

■ Modules **BMX P34 2010 / 2030**



Value	Designation
1	Display
2	USB port
3	Eth port
4	Serial port
5	CANopen port

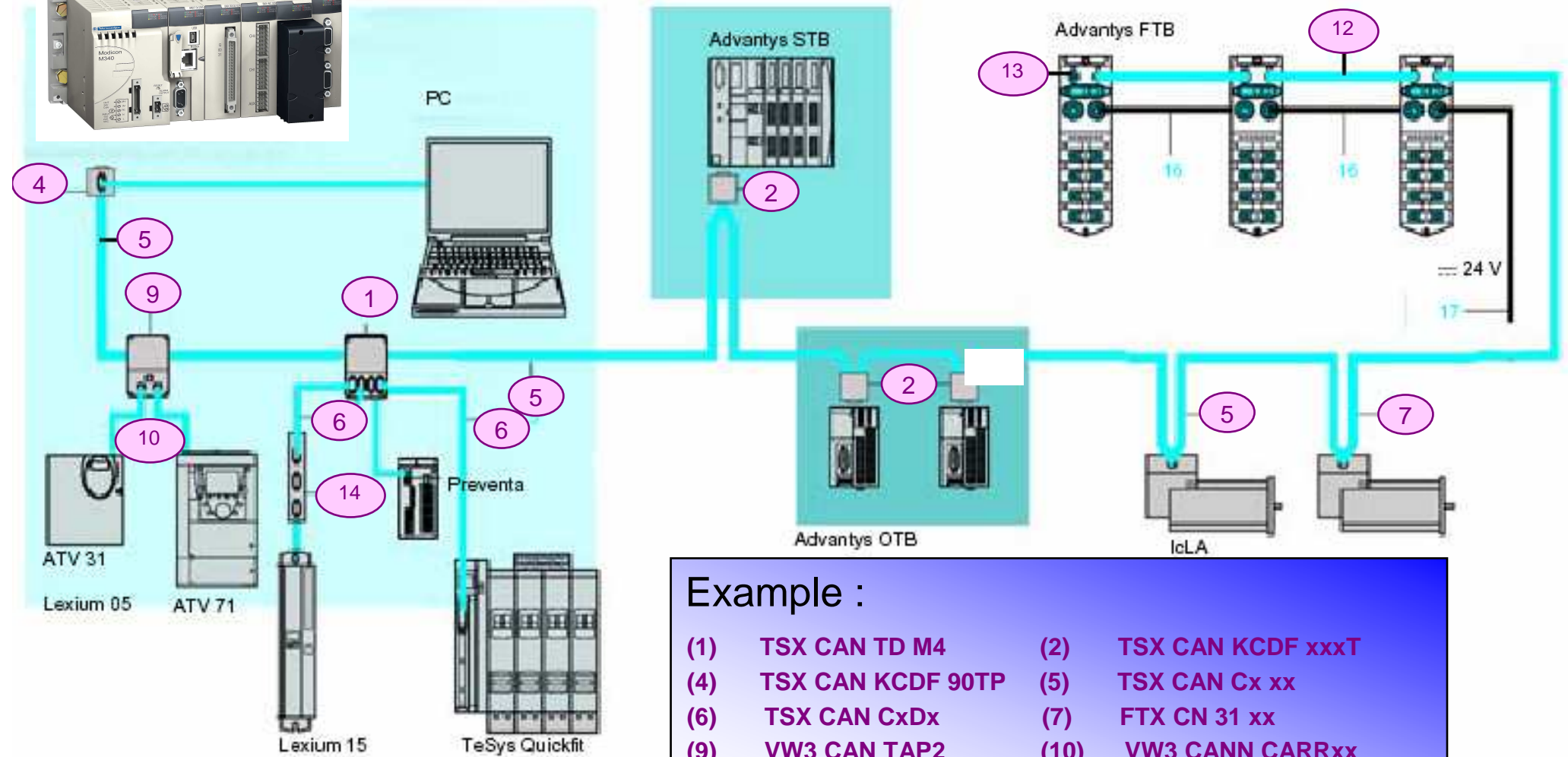
Wiring on the processor

- The connection is done on a SUBD 9 pts pin male connector of the processor



PIN	SIGNAL	
1		
2	CAN-L	
3	CAN-GND	
4		
5		
6	GND	
7	CAN-H	
8		
9		

Wiring (a global offer)

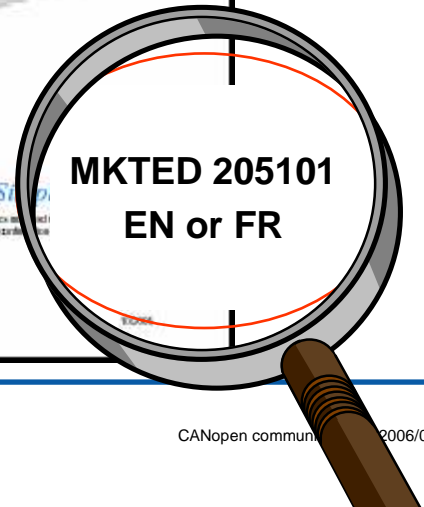
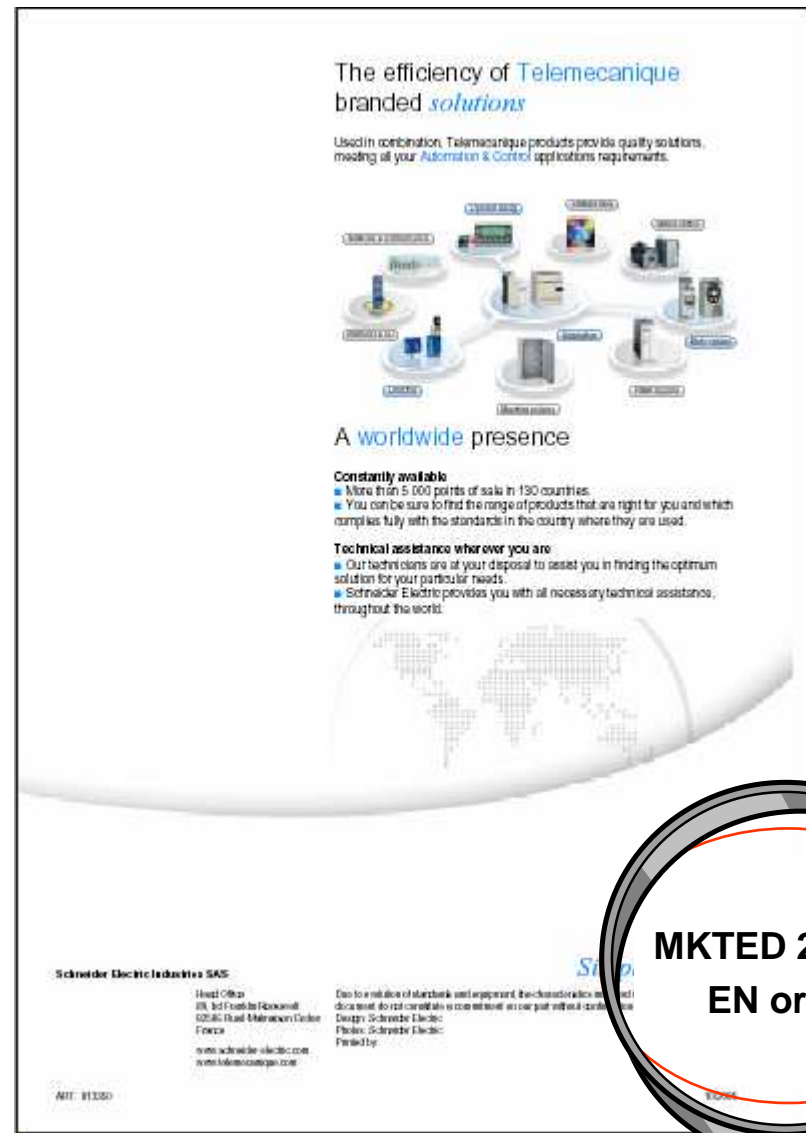
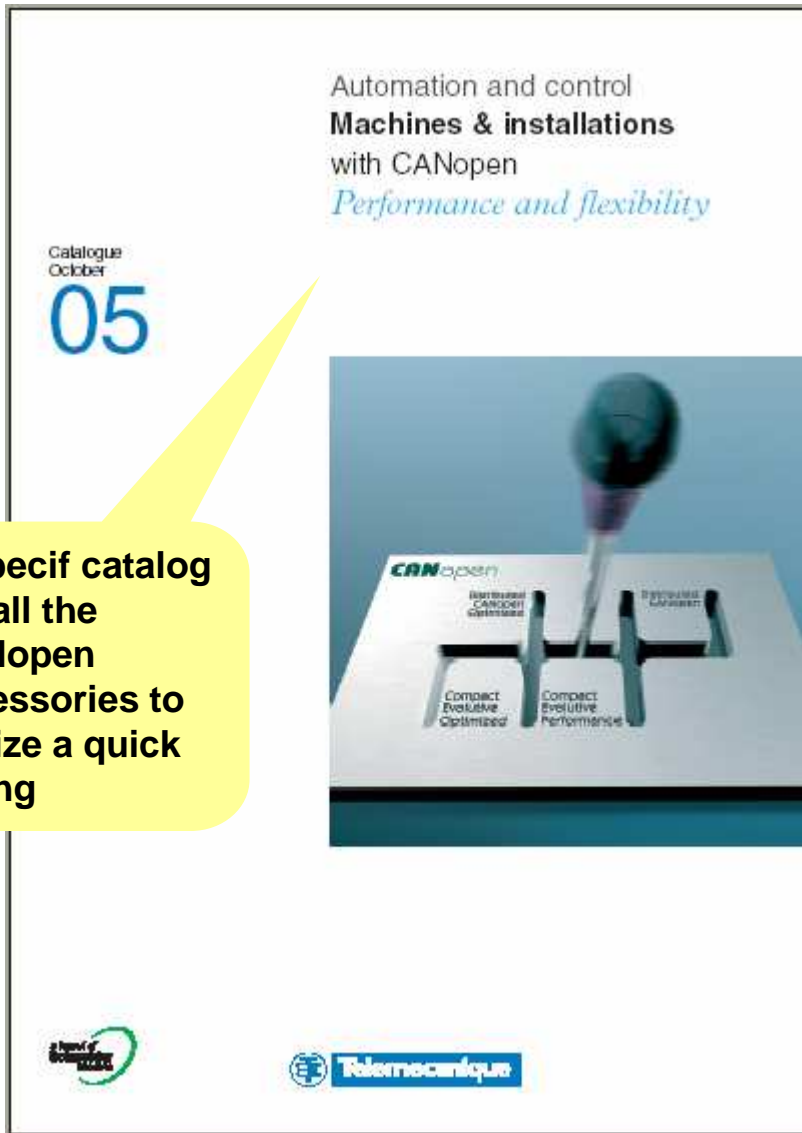


Example :

- | | |
|-----------------------|-----------------------|
| (1) TSX CAN TD M4 | (2) TSX CAN KCDF xxxT |
| (4) TSX CAN KCDF 90TP | (5) TSX CAN Cx xx |
| (6) TSX CAN CxDx | (7) FTX CN 31 xx |
| (9) VW3 CAN TAP2 | (10) VW3 CANN CARRxx |
| (12) FTX CN 32 xx | (13) FTX CNTL 12 |
| (14) AMO 2CA 001V000 | |

Wiring (a global offer)

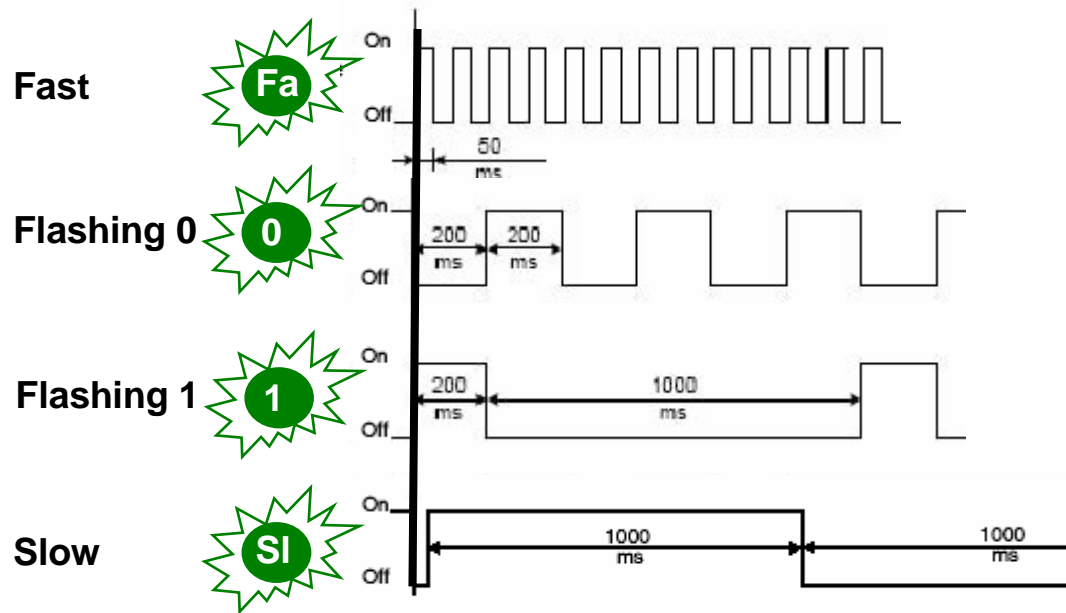
A specif catalog list all the CANopen accessories to realize a quick wiring






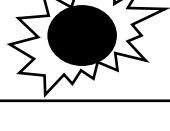


Visual diagnostic of CANopen

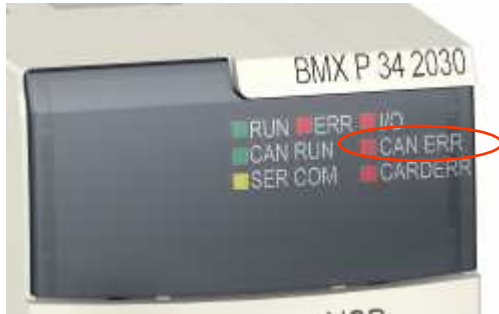


The information provided by the the **CAN RUN** lamp

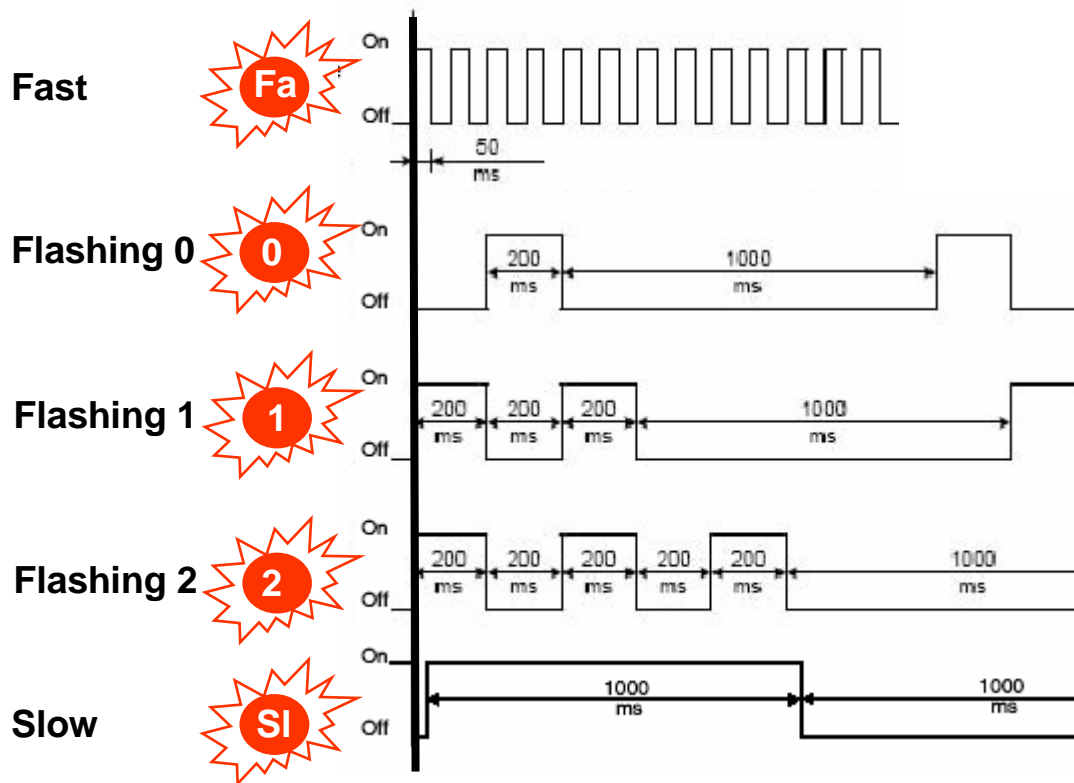


State	Comment
	Operationnal state
	Speed automatic detection
	Pre Operationnal state
	Stopped state
	Check & init the slave
	-

Visual diagnostic of CANopen



The information provide by the The **CAN ERR** lamp



State	Comment
	Bus stopped
	Speed automatic detection
	Error counter overflow
	Heartbeat fault
	No Sync message
	Canopen can't start
	No ERROR

C – Software configuration

Software Configuration

- The configuration for CANopen is fully integrated in UNITY (no external configurator need)
 - Here is the main steps to respect in order to configure the CANopen bus.

Configuration du Bus CANopen :

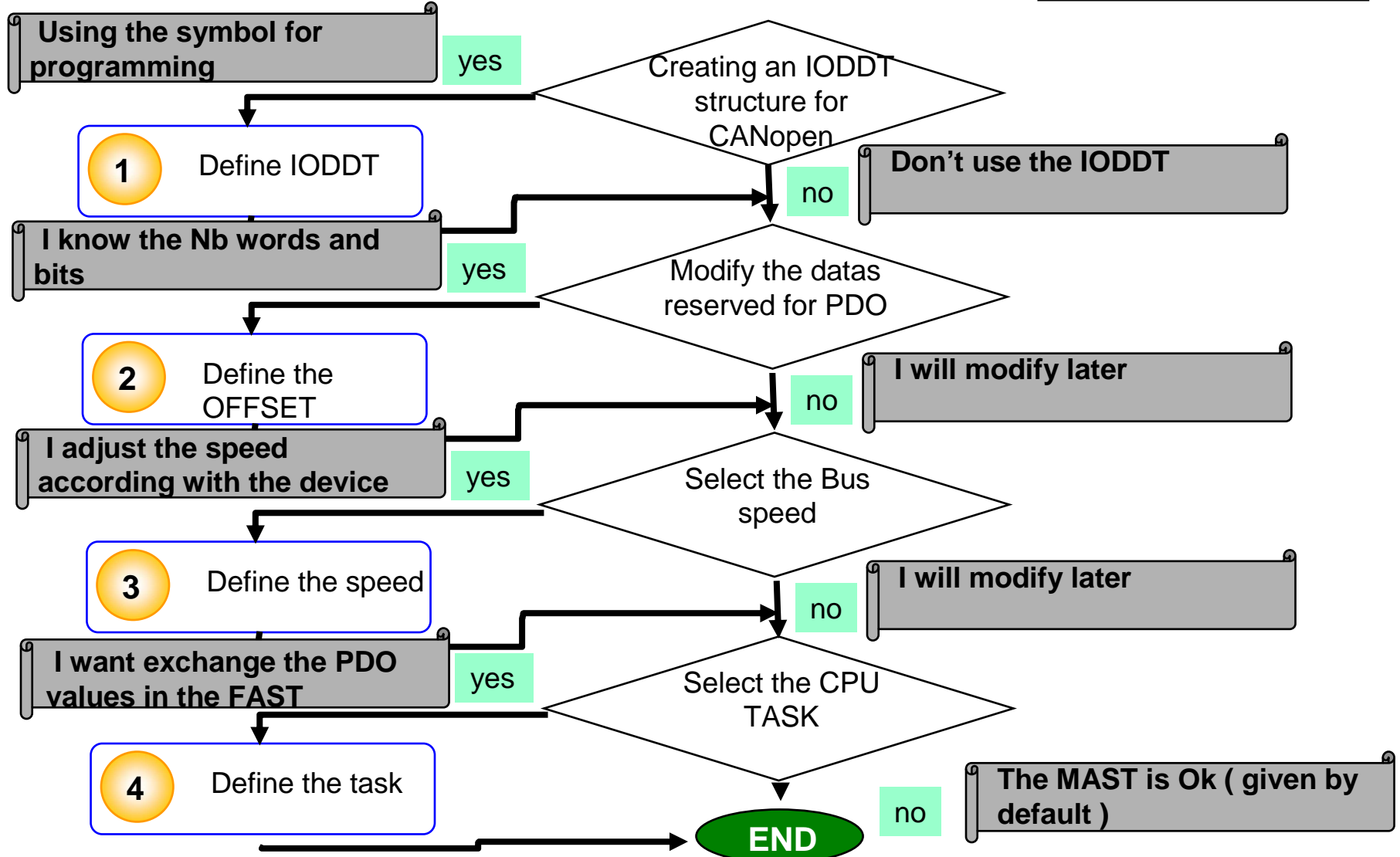
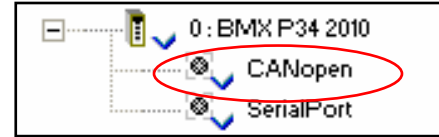
step 1 : Master Configuration (Speed, @, PDO Offset)

step 2 : Slave Configuration (PDO and exchanges)

- Standard
- Expert

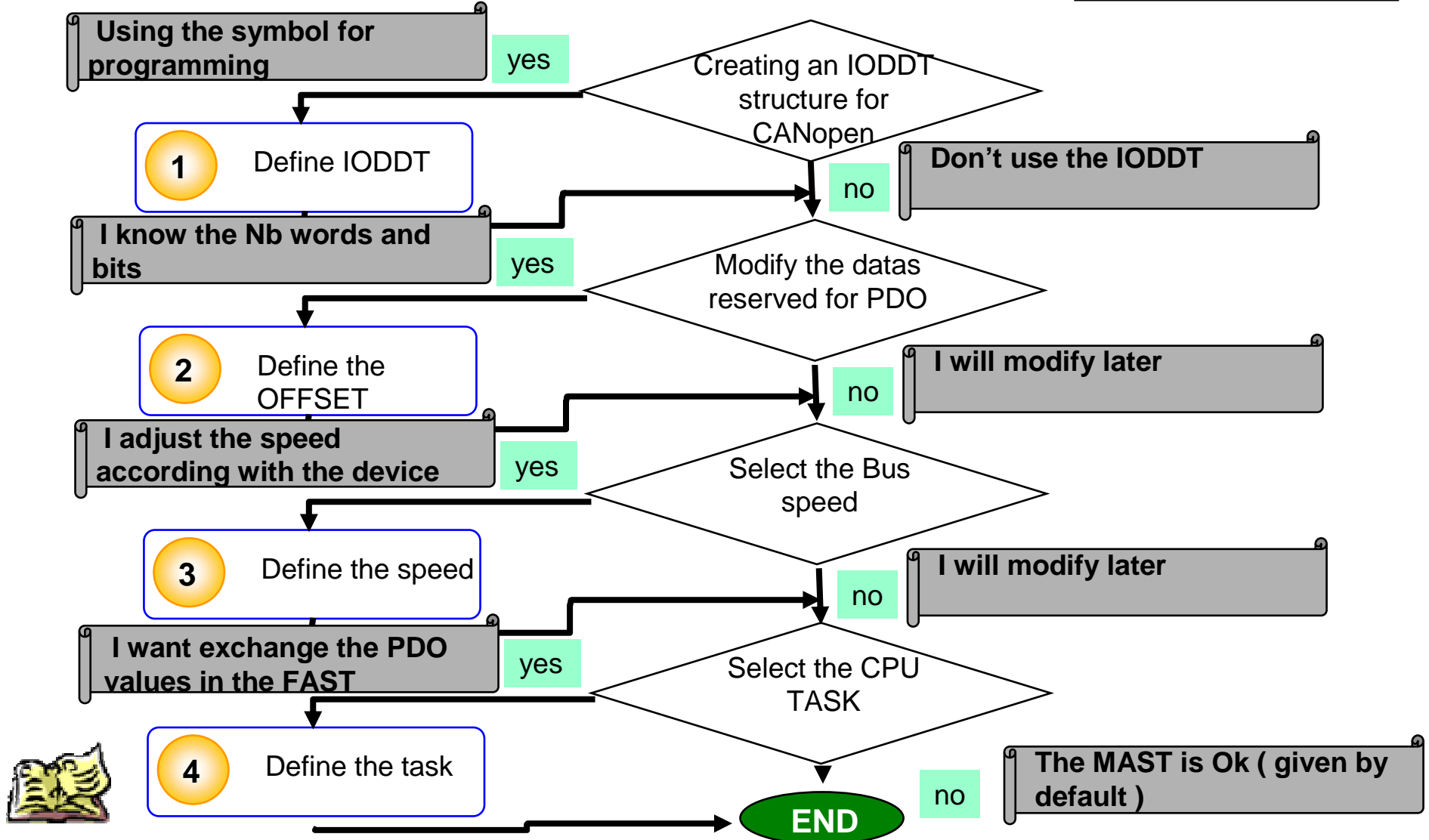
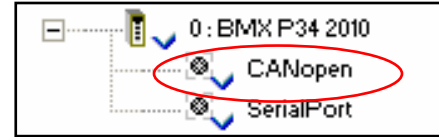
step 1 : Master Configuration

Processor configuration



step 1 : Master Configuration

Processor configuration



step 2 : Slave Configuration

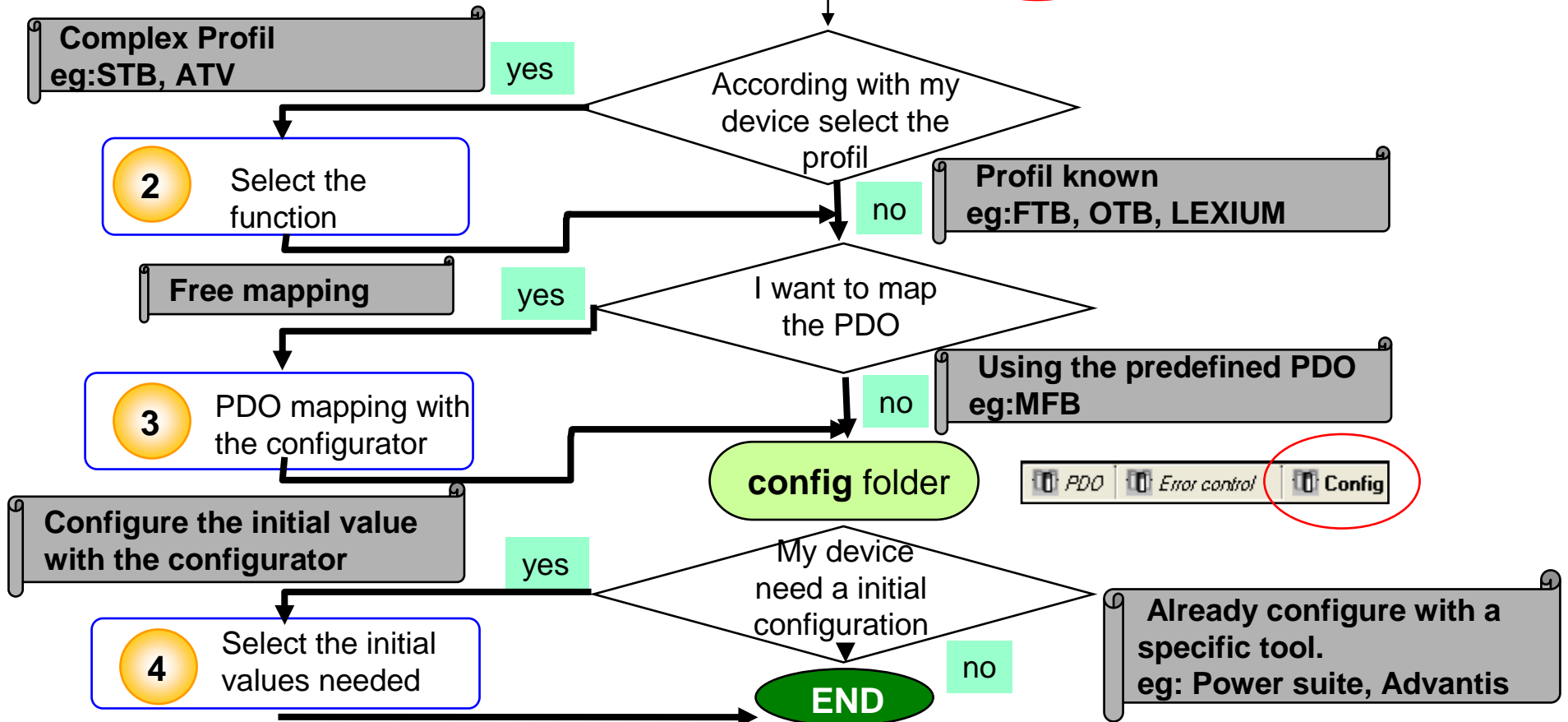
Logical node CANopen

3 : CANopen Micro

1 We must select the device

PDO folder

PDO Error control Config



step 2 : Slave Configuration

Logical node CANopen

3 : CANopen Micro

1 We must select the device

PDO folder

PDO Error control Config

Complex Profil
eg:STB, ATV

yes

2 Define a profil

Free mapping

yes

3 PDO mapping with
the configurator

Configure the initial value
with the configurator

yes

4 Select the initial
values needed

According with my
device select the
profil

no

Profil known
eg:FTB, OTB, LEXIUM

I want to map
the PDO

no

Using the predefined PDO
eg:MFB

config folder

PDO Error control Config

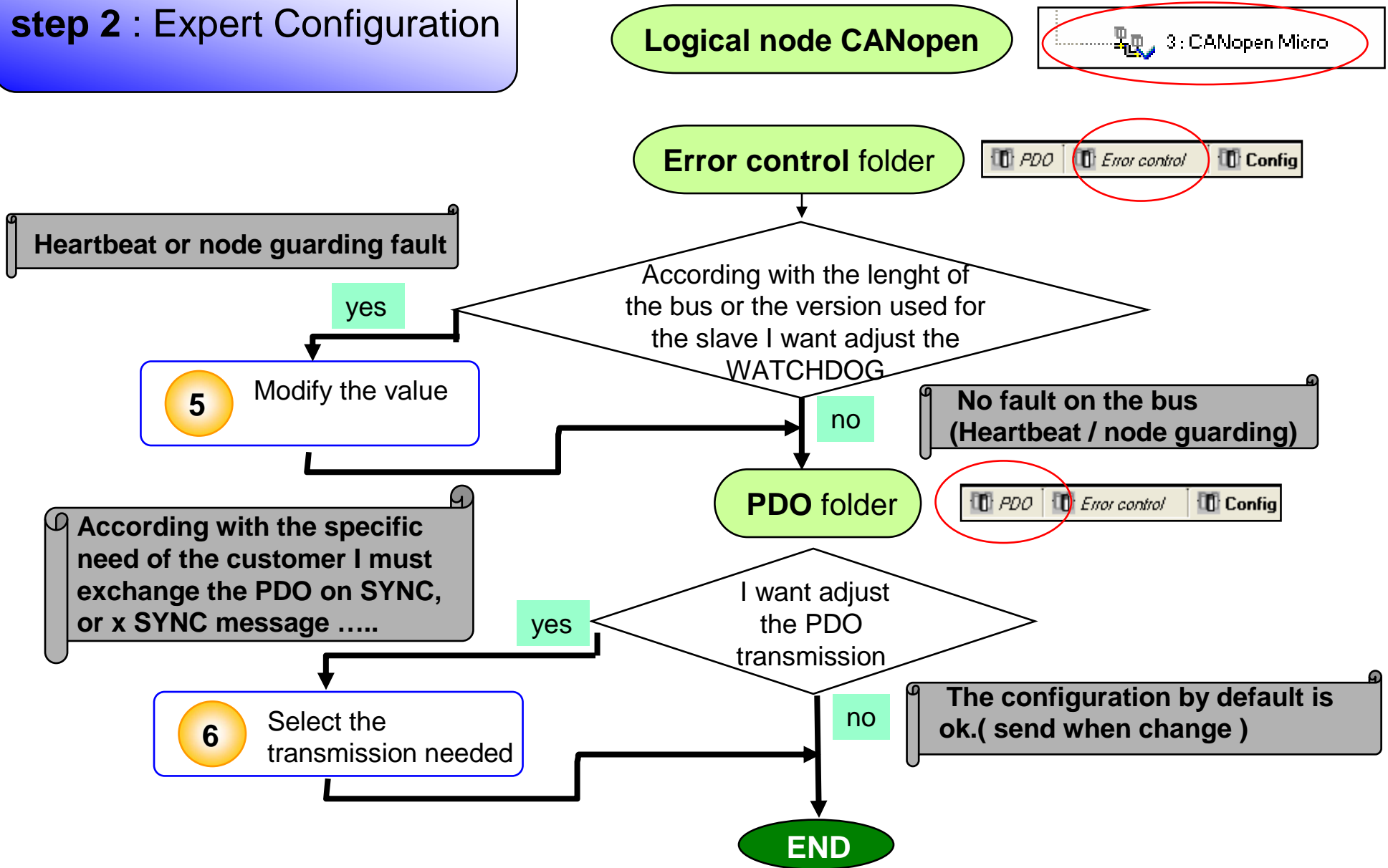
My device
need a initial
configuration

no

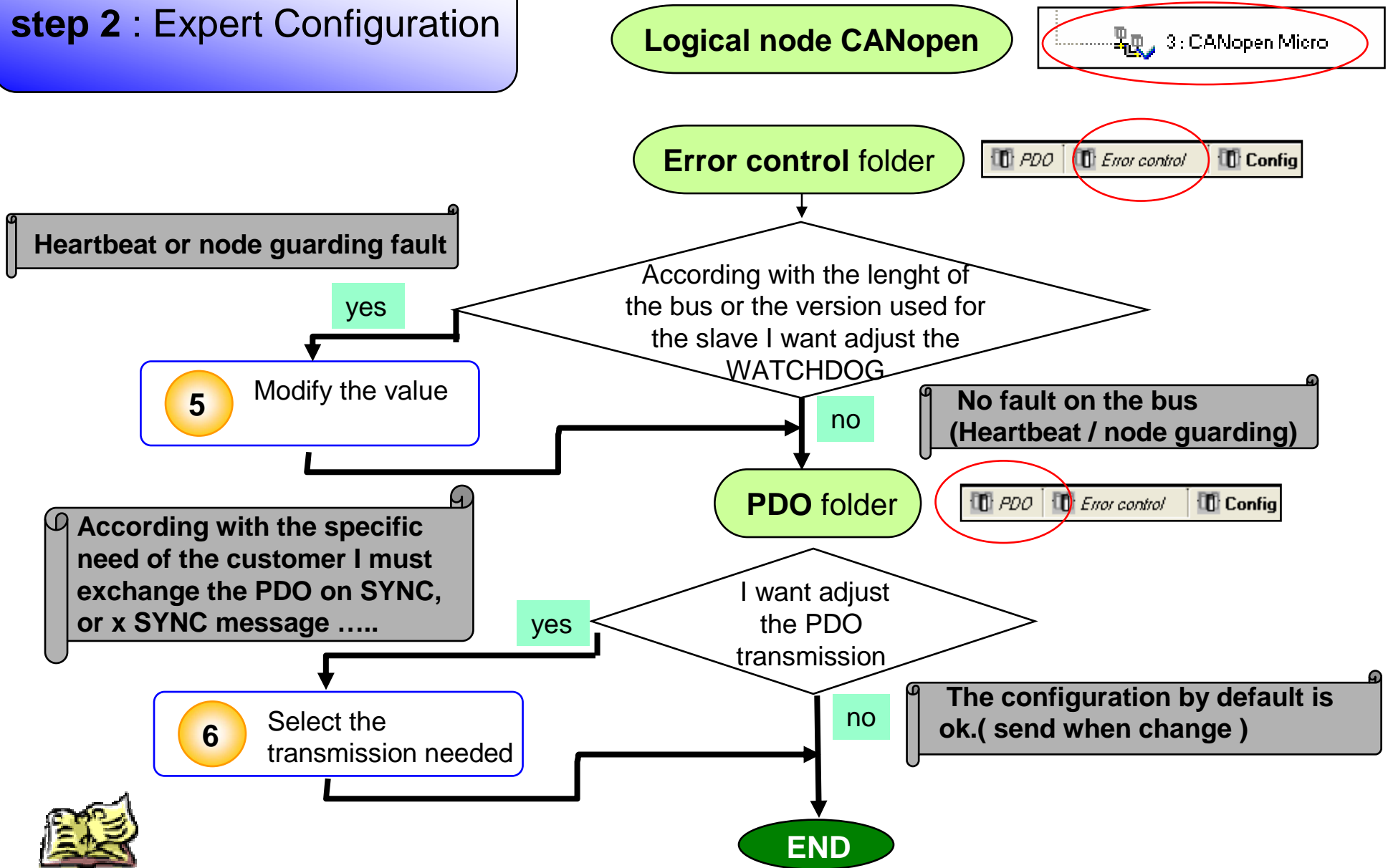
Already configure with a
specific tool.
eg: Power suite, Advantis

END

step 2 : Expert Configuration



step 2 : Expert Configuration



Step 1: Master configuration

Step 1 : Master configuration

In the CAN screen we have :

1. **A overview of the bus specifications**
2. The I/O configuration
3. The master configuration

Project Browser

Structural view

Station

Configuration

- 0: PLC bus
 - 0: BMX XBP 0800
 - (P)(P) BMX CPS 2000
 - 0: BMX P34 2030
 - CANopen**
 - Ethernet

Communicator head CANopen

CANopen comm head

Channel 2

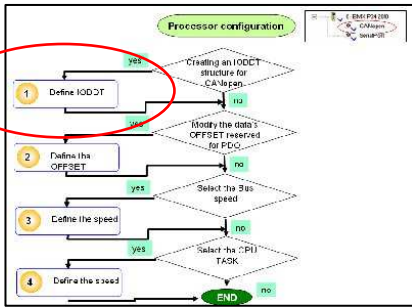
Overview I/O objects

Integrated CANopen

SPECIFICATIONS

Bus type	CANopen
Structure	
Physical interface	ISO 11898
Data rate	10 Kbps to 1 Mbps
Services	<p>CANopen:</p> <ul style="list-style-type: none">- implicit exchange of Process Data Object via %MW words- explicit exchange of Service Data Object by READ_VAR/WRITE_VAR function block <p>CAN</p> <ul style="list-style-type: none">- explicit exchange of CAN PDU by SEND_REQ generic function block- explicit transmission of CAN PDU by SEND_REQ generic function block

Step 1 : Master configuration 'IODDT'



- In the CAN screen we have :
1. A overview of the bus specifications
 2. **The I/O configuration**
 3. The master configuration

Project Browser

Structural view

Station

Configuration

0: PLC bus

Communicator head CANopen

CANopen comm head

Overview I/O objects

I/O variable creation

Prefix for name:

Type: T_COM_CO_BMX

Comment:

Create

I/O Objects

Channel: ☒ %CH

Configuration: ☒ %Kw ☒ %KD ☒ %KF

System: ☒ %Mw

Status: ☒ %Mw

Parameter: ☒ %Mw ☒ %MD ☒ %MF

Command: ☒ %Mw ☒ %MD ☒ %MF

Implicits: ☒ %I ☒ %Iw ☒ %ID ☒ %IF ☒ %IERR

☒ %Q ☒ %Qw ☒ %QD ☒ %QF

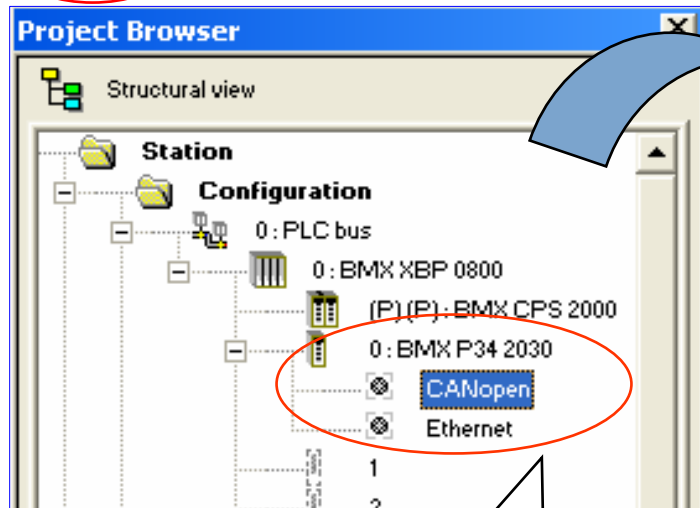
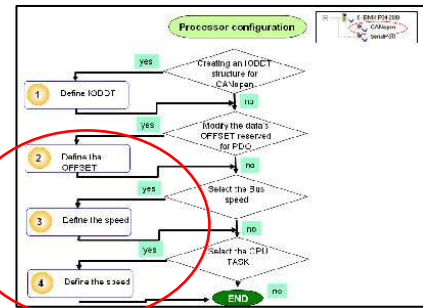
Update

	Address	Name	Type	Comment
1	%CH0.0.2			
2	%Kw0.0.2		INT	
3	%Kw0.0.2.1		INT	
4	%Kw0.0.2.2		INT	
5	%Kw0.0.2.3		INT	
6	%Kw0.0.2.4		INT	
7	%Kw0.0.2.5		INT	
8	%Kw0.0.2.6		INT	
9	%Kw0.0.2.7		INT	
10	%Kw0.0.2.8		INT	
11	%Kw0.0.2.9		INT	
12	%Kw0.0.2.10		INT	
13	%Kw0.0.2.11		INT	
14	%IO.0.2.ERR		BOOL	
15	%Iw0.0.2		INT	
16	%Iw0.0.2.1		INT	
17	%Iw0.0.2.2		INT	
18	%Iw0.0.2.3		INT	
19	%Iw0.0.2.4		INT	
20	%Iw0.0.2.5		INT	
21	%Iw0.0.2.6		INT	
22	%Iw0.0.2.7		INT	
23	%Iw0.0.2.8		INT	
24	%Iw0.0.2.9		INT	
25	%Iw0.0.2.10		INT	

Step 1 : Master configuration

In the CAN screen we have :

1. A overview of the bus specifications
2. The I/O configuration
3. **The Master configuration**



According with the processor it's possible to configure the bus.

Here : CANopen

communicator head CANopen

CANopen comm head

Channel 2

2

Config

Inputs

Nb. of words (%MW) 32

Index of 1st %MW 100

Nb. of bits (%M) 32

Index of 1st %M 0

Outputs

Nb. of words (%MW) 32

Index of 1st %MW 200

Nb. of bits (%M) 32

Index of 1st %M 32

Bus parameters

Transmission speed 500 kBaud

SYNC Message COB-ID 128

SYNC Message Period 100 ms

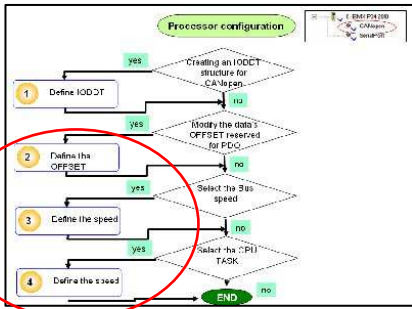
3

Function: CANopen

Task: MAST

4

Step 1 : Master configuration



Communicator head CANopen

CANopen comm head

Channel 2

2

Config

Inputs

Nb. of words (%Mw) 32

Index of 1st %Mw 100

Nb. of bits (%M) 32

Index of 1st %M 0

Outputs

Nb. of words (%Mw) 32

Index of 1st %Mw 200

Nb. of bits (%M) 32

Index of 1st %M 32

Bus parameters

Transmission speed 500 kBaud

SYNC Message COB-ID 128

SYNC Message Period 100 ms

3

Function: CAN

Task: MAST

■ Map the The PDO's to memory word (%mw) and bit (%m) location (mainly dedicated for the MODBUS communication).

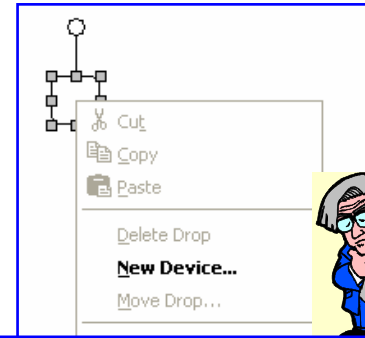
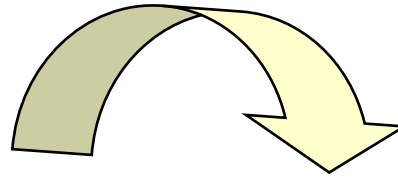
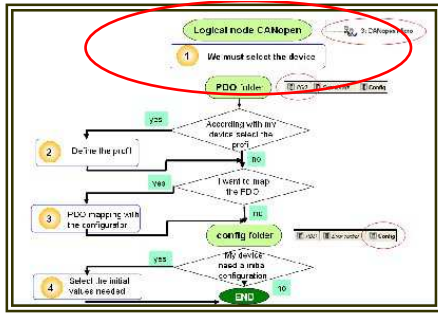
■ Bus Parameters :

1. Set speed according to the length of the bus.
2. Period of the Sync Message

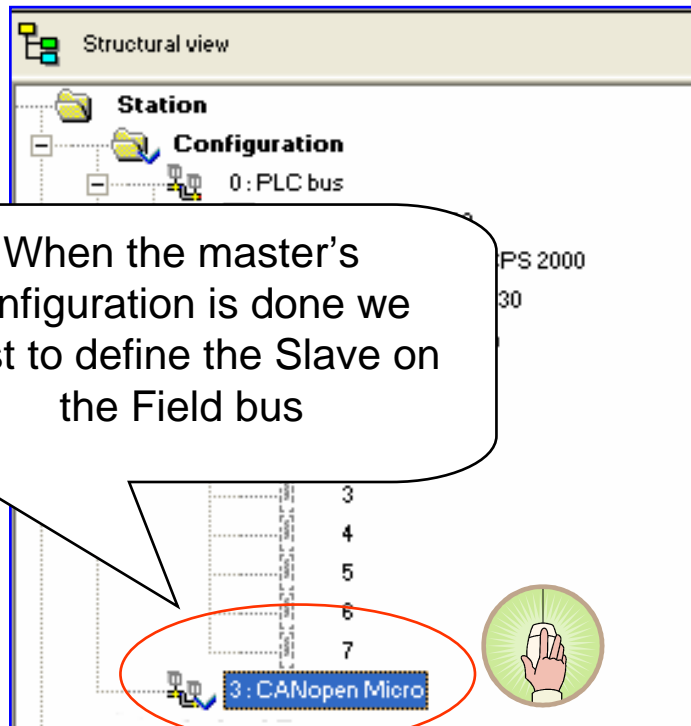
■ Task assigned to the PDOs

Step 2: Slave configuration

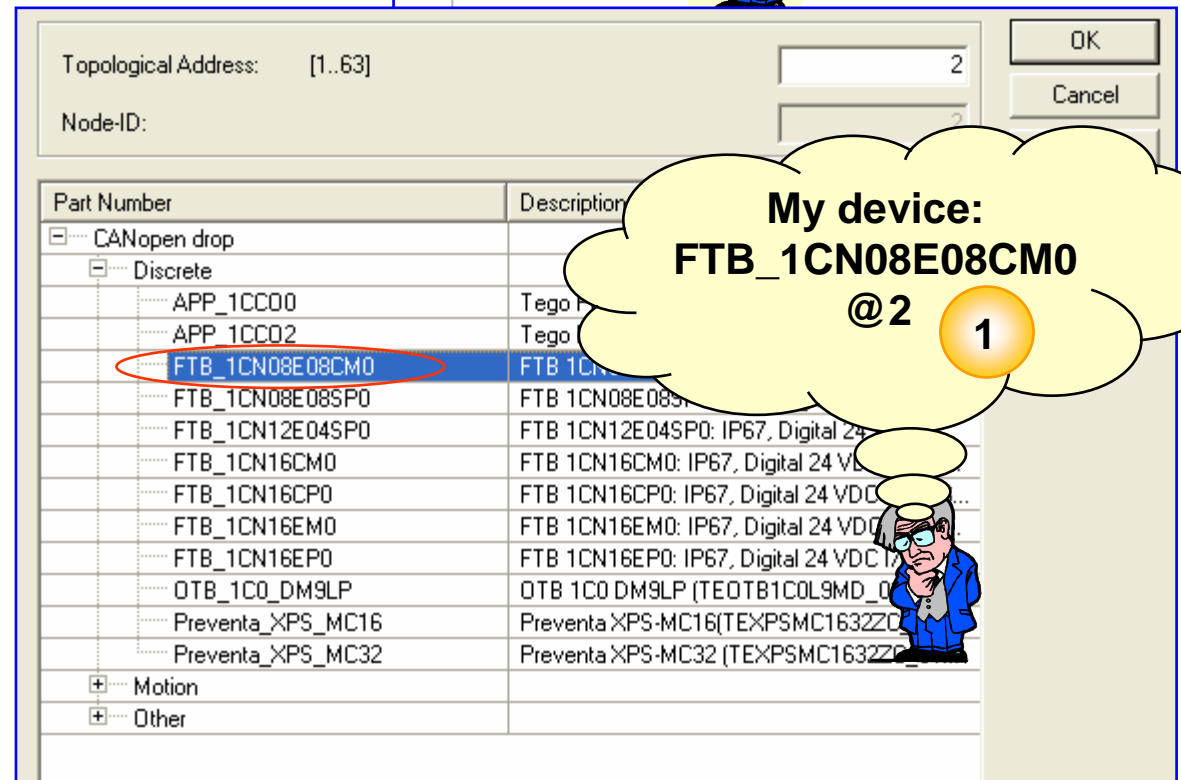
Step 2 : Slave Configuration



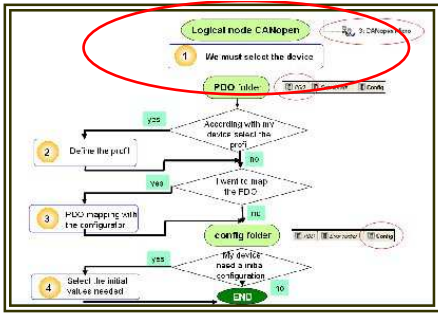
New device: yes



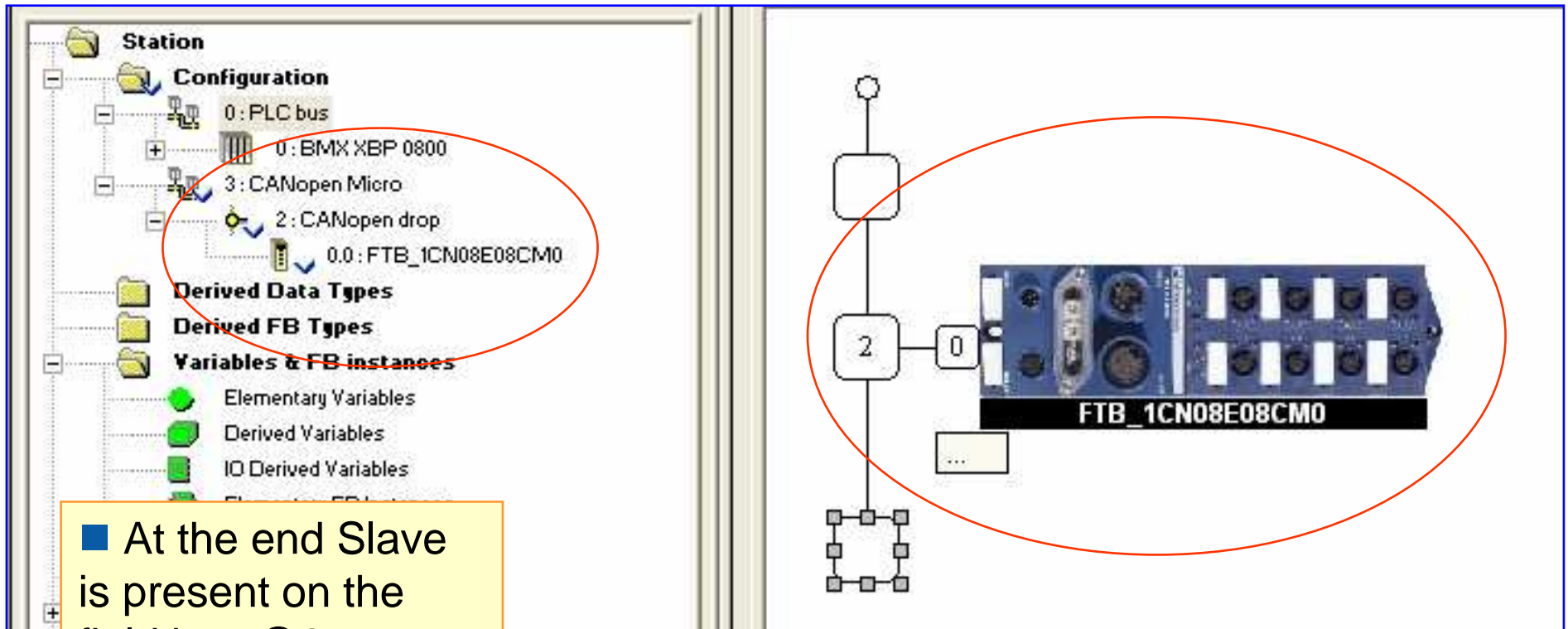
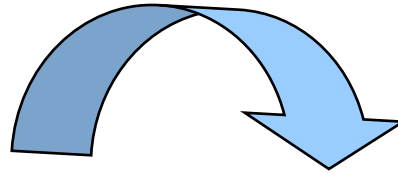
When the master's configuration is done we must to define the Slave on the Field bus



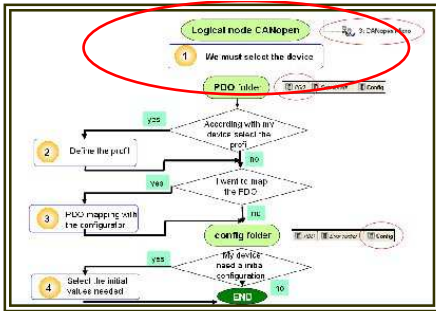
My device:
FTB_1CN08E08CM0
@2 1



Step 2 : Slave Configuration



■ At the end Slave is present on the field bus @2



Step 2 : Slave Configuration

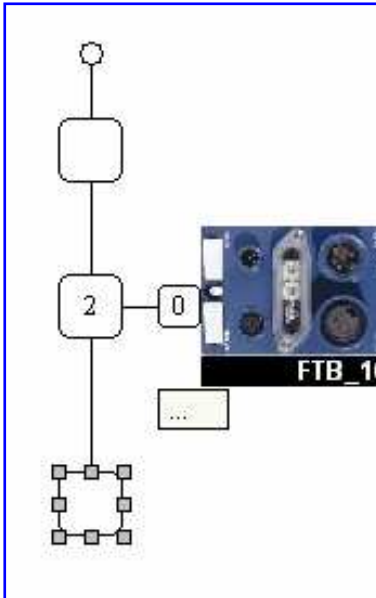
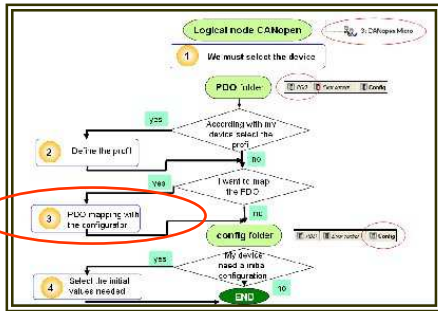
In L11 Unity includes following Schneider devices.

The possibility to add a third party product will be added in L12.

Part Number	Description
CANopen drop	
Discrete	
APP_1CC00	Tego Power CANopen (APP1CC00.eds)
APP_1CC02	Tego Power CANopen (APP1CC02.eds)
FTB_1CN08E08CM0	FTB 1CN08E08CM0: IP67, Digital 24 VDC I/O, 8 In...
FTB_1CN08E08SP0	FTB 1CN08E08SP0: IP67, Digital 24 VDC I/O, 8 Inp...
FTB_1CN12E04SP0	FTB 1CN12E04SP0: IP67, Digital 24 VDC I/O, 12 In...
FTB_1CN16CM0	FTB 1CN16CM0: IP67, Digital 24 VDC I/O, 16 Input ...
FTB_1CN16CP0	FTB 1CN16CP0: IP67, Digital 24 VDC I/O, 16 Input ...
FTB_1CN16EM0	FTB 1CN16EM0: IP67, Digital 24 VDC I/O, 16 Input ...
FTB_1CN16EP0	FTB 1CN16EP0: IP67, Digital 24 VDC I/O, 16 Input ...
OTB_1C0_DM9LP	OTB 1C0 DM9LP (TEOTB1C0L9MD_0100E.eds)
Preventa_XPS_MC16	Preventa XPS-MC16 (TEXPSMC1632ZC_0105E.eds)
Preventa_XPS_MC32	Preventa XPS-MC32 (TEXPSMC1632ZC_0105E.eds)
Motion	
ATV31_V1_1	Altivar 31 CANopen Slave DSP402 (TEATV3111E.e...
ATV31_V1_2	Altivar 31 CANopen Slave DSP402 (TEATV3112E.e...
ATV31T_V1_3	Altivar 31 CANopen Slave DSP402 (TEATV31T13E....
ATV61_V1_1	ATV61 (TEATV6111E.eds)
ATV71_V1_1	ATV71 (TEATV7111E.eds)
IcIA_IFA	IcIA-IFA CANopen (IcIA-IFA.eds)
IcIA_IFE	IcIA-IFE CANopen (IcIA-IFE.eds)
IcIA_IFS	IcIA-IFS CANopen (IcIA-IFS.eds)
IcIA_N065	IcIAN065 based on profiles DS301V4.01 and DSP4...
Lexium05	DCX170 CANopen (TEDCX170_0100E.eds)
Lexium05_MFB	LXM05A PLCopen (LEXIUM05_MFB.EDS)
Lexium15_HP	LEXIUM 15 HP servodrive (Lexium 15 MP HP.eds)
Lexium15_MP	LEXIUM 15 MP servodrive (Lexium 15 MP HP.eds)
Osicoder	Osicoder - absolute rotary multi-turn encoders based ...
Other	
STB_NCO_1010	STB NCO 1010 CANopen Network Interface Modul...
STB_NCO_2212	STB NCO 2212 CANopen Network Interface Modul...



Step 2 : Slave Configuration



FTB 1CN08E08CM0: IP67, Digital 24 VDC I/O, 8 Input Points, 8 Input Points or Output Points (Configurable), 1.6 A (TEFTB05M01E.eds)

Config | **PDO** | **Error control**

PDO	Tr.Type	InhibitTime	Event Ti...	Symbol	Topo.Addr.	%M...	COBID	Index
<input checked="" type="checkbox"/> PDO 1	255	0	0				-	
+ Digital Input 8 Bit...					%I...			6000:01
+ Digital Input 8 Bit...					%I...			6000:02
<input type="checkbox"/> PDO 6	255	0	0				-	
+ Actuator warning...					%I...			3000:05
+ Actuator shutdo...					%I...			3000:03
+ Common diagno...					%Iw13.210.0.0.0			3000:01
+ Sensor short cir...					%I...			3000:02

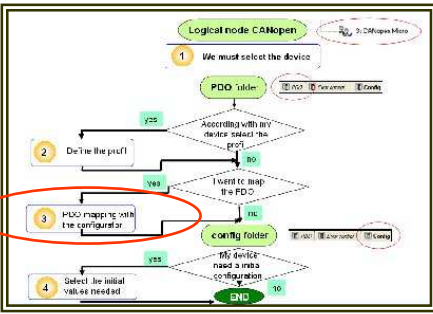
Receive (%Q)

PDO	Tr.Type	InhibitTime	Event Tim...	Symbol	Topo.Addr.	%M...	COBID
<input type="checkbox"/> PDO 1	255						-
- Write Outputs 1 to 8					%Q...		
Write Outputs 1 to 8 Ch:0					%Q13.210.0.0.0		
Write Outputs 1 to 8 Ch:1					%Q13.210.0.1.0		
Write Outputs 1 to 8 Ch:2					%Q13.210.0.2.0		
Write Outputs 1 to 8 Ch:3					%Q13.210.0.3.0		
Write Outputs 1 to 8 Ch:4					%Q13.210.0.4.0		
Write Outputs 1 to 8 Ch:5					%Q13.210.0.5.0		
Write Outputs 1 to 8 Ch:6					%Q13.210.0.6.0		
Write Outputs 1 to 8 Ch:7					%Q13.210.0.7.0		
<input type="checkbox"/> PDO 6	255						-

Step 2 : Slave Configuration

■ **PDO** : In this screen it is possible to map the PDO communication.

■ This screen is divided in 3 parts :



FTB 1CN08E08CM0: IP67, Digital 24 VDC I/O, 8 Input Points, 8 Input Points or Output Points (Configurable), 1.6 A (TEFTB05M01E.ed5)

PDO | Error control | Config

PDO	Tr.Type	Inhibit TI...	Event TI...	Symbol	Topo.Addr.	%M...	COBID	Index
PDO 1	255	0	0				-	
Digital Input 8 ...				%I...				6000:01
Digital Input 8 ...				%I...				6000:02
PDO 6	255	0	0				-	
Actuator warn...				%I...				3000:05
Actuator shut...				%I...				3000:03
Common diag...				%IwV3.2v0.0.0.0				3000:01
Sensor short ...				%I...				3000:02

Receive (%Q)

PDO	Tr.Type	Inhibit TI...	Event TI...	Symbol	Topo.Addr.	%M...	COBID	Index
PDO 1	255						-	
Write Outputs 1 ...				%Q...				6200:01
Write Outp...				%Qw3.2v0.0.0.0				
Write Outp...				%Qw3.2v0.0.1.0				
Write Outp...				%Qw3.2v0.0.2.0				
Write Outp...				%Qw3.2v0.0.3.0				
Write Outp...				%Qw3.2v0.0.4.0				
Write Outp...				%Qw3.2v0.0.5.0				
Write Outp...				%Qw3.2v0.0.6.0				
Write Outp...				%Qw3.2v0.0.7.0				
PDO 6	255						-	

Variables
☐ Display only unmapped variables

Parameter Name	Ind...	Topo.Addr.
Common diagnosis	3000:01	%IwV3.2v0.0.0.0
Sensor short circuit	3000:02	%I...
Actuator shutdown pin 4	3000:03	%I...
Actuator warning pin 4	3000:05	%I...
Desina inputs	3000:07	%IwV3.2v0.0.0.1
Digital Input 8 Bits Pin4	6000:01	%I...
Digital Input 8 Bits Pin2	6000:02	%I...
Write Outputs 1 to 8	6200:01	%Q...

Transmits PDOs
(slave transmits an information to the master)

Receives PDOs
(Slave receives an information from the Master)

A list of the variables that can be mapped in the PDOs

Step 2 : Slave Configuration

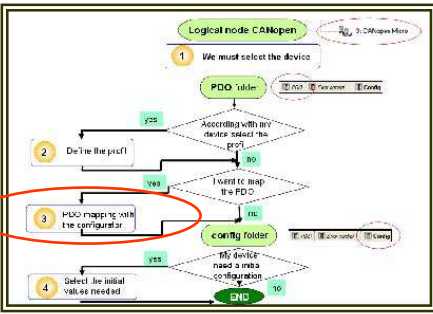
■ **PDO** : In this screen it is possible to map the PDO communication.

■ This screen is divided in 3 parts :

■ According to the EDS file some PDOs are already mapped

■ The PDOs used Topologies addresses %Ix or %Qx.

and
Internals variables %MW (these variables are always present in the product for the MODBUS communication)



FTB 1CN08E08CM0: IP67, Digital 24 VDC I/O, 8 Input Points, 8 Input Points or Output Points (Configurable), 1.6 A (TEFTB05M01E.eds)

PDO | Error control | Config

PDO	Tr.Type	Inhibit TI...	Event TI...	Symbol	Topo.Addr.	%M...	COBID	Index
PDO 1	255	0	0				-	
Digital Input 8 ...					%I...			6000:01
Digital Input 8 ...					%I...			6000:02
PDO 6	255	0	0				-	
Actuator warn...					%I...			3000:05
Actuator shut...					%I...			3000:03
Common diag...					%IwV3.2v0.0.0.0			3000:01
Sensor short ...					%I...			3000:02

Receive (%Q)

PDO	Tr.Type	Inhibit TI...	Event TI...	Symbol	Topo.Addr.	%M...	COBID	Index
PDO 1	255						-	
Write Outputs 1 ...					%Q...			6200:01
Write Outp...					%Qw3.2v0.0.0.0			
Write Outp...					%Qw3.2v0.0.1.0			
Write Outp...					%Qw3.2v0.0.2.0			
Write Outp...					%Qw3.2v0.0.3.0			

Variables
☐ Display only unmapped variables

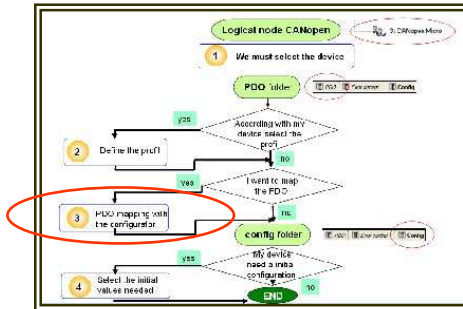
Parameter Name	Ind...	Topo.Addr.
Common diagnosis	3000:01	%IwV3.2v0.0.0.0
Sensor short circuit	3000:02	%I...
Actuator shutdown pin 4	3000:03	%I...
Actuator warning pin 4	3000:05	%I...
Digital Input 8 Bits Pin4	6000:01	%I...
Digital Input 8 Bits Pin2	6000:02	%I...
Write Outputs 1 to 8	6200:01	%Q...



With the CANopen configurator if I add a PDO or if I modify a PDO, the mapping for the %Q/I and %MW will never change, so it's not necessary to adjust your application.

Step 2: Slave configuration

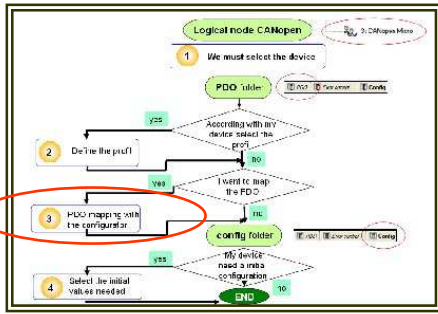
How to do the PDO configuration





Step 2 : Slave Configuration

- Particularity : For the STB devices.



STB NCO 2212 CANopen Network Interface Module (STBNCO2212.ed5)

Transmit (%I)

PDO	Tr.Type	InhibitT...	Event ...	Symbol	Topo.Addr.	%M...	COBID	Index
<input checked="" type="checkbox"/> PDO 1	255	0					16#182	
<input checked="" type="checkbox"/> Digital 8-bit...					%Iw13.210.0.0.35	%Mw535		6000:01
<input checked="" type="checkbox"/> PDO 2	255	0					-	
<input checked="" type="checkbox"/> PDO 3	255	0					-	
<input checked="" type="checkbox"/> PDO 4	255	0					-	
<input checked="" type="checkbox"/> PDO 5	255	0					-	
<input checked="" type="checkbox"/> PDO 6	255	0					-	
<input checked="" type="checkbox"/> PDO 7	255	0					-	
<input checked="" type="checkbox"/> PDO 8	255	0					-	
<input checked="" type="checkbox"/> PDO 9	255	0					-	
<input checked="" type="checkbox"/> PDO 10	255	0					-	
<input checked="" type="checkbox"/> PDO 11	255	0					-	
<input checked="" type="checkbox"/> PDO 12	255	0					-	
<input checked="" type="checkbox"/> PDO 13	255	0					-	

Receive (%Q)

PDO	Tr.Type	InhibitT...	Event ...	Symbol	Topo.Addr.	%M...	COBID	Index
<input checked="" type="checkbox"/> PDO 1	255						16#202	
<input checked="" type="checkbox"/> Digital 8-bit...					%Qw13.210.0.0.0	%Mw649		6200:01
<input checked="" type="checkbox"/> PDO 2	255						-	
<input checked="" type="checkbox"/> PDO 3	255						-	
<input checked="" type="checkbox"/> PDO 4	255						-	
<input checked="" type="checkbox"/> PDO 5	255						-	

Import DCF...

Variables

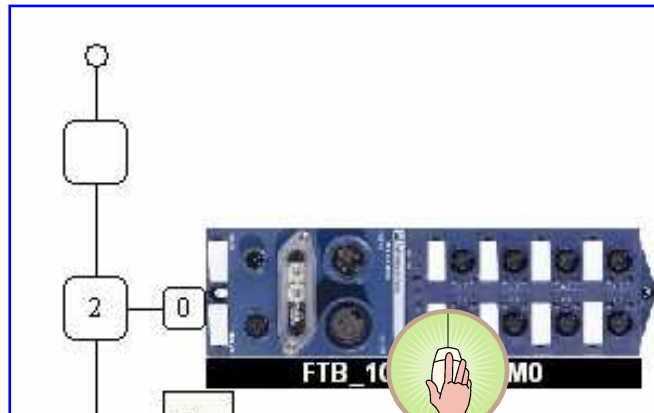
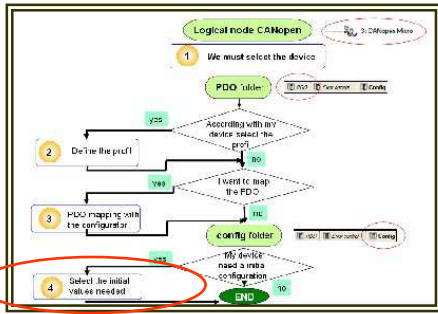
☐ Display only unmapped variables

Parameter Na...	Ind...
Island Diagnostics: ...	4000:0
Island Diagnostics: L...	4001:0
Configured Nodes 1...	4002:0
Configured Nodes 3...	4002:0
Configured Nodes 4...	4002:0
Configured Nodes 6...	4002:0
Configured Nodes 8...	4002:0
Configured Nodes 9...	4002:0
Configured Nodes 11...	4002:0
Configured Nodes 1...	4002:0
Operational Nodes 1...	4003:0
Operational Nodes 3...	4003:0
Operational Nodes 4...	4003:0
Operational Nodes 6...	4003:0
Operational Nodes 8...	4003:0
Operational Nodes 9...	4003:06
Operational Nodes 1...	4003:07
Operational Nodes 1...	4003:08
Nodes with Error 16...	4004:01
Nodes with Error 32...	4004:02
Nodes with Error 48...	4004:03
Nodes with Error 64...	4004:04
Nodes with Error 80...	4004:05
Nodes with Error 96...	4004:06

For the **STB** configuration it's possible to associate a specif file created with ADVANTYS (version 2.3 mini) in order to design exactly the configuration done on STB, this file is called **xxx.DCF**.

If we don't used this file it's always possible to manage the communication manually

Step 2 : Slave Configuration



FTB 1CN08E08CM0: IP67, Digital 24 VDC I/O, 8 Input Points, 8 Input Points or Output Points (Configurable), 1.6 A (TEFTB05M01E.eds)

FTB_1CN08E08CM0
Channel 0

Config

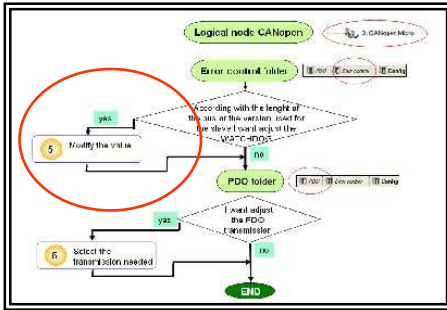
	Index	Label	Value
0	2001:01	Input/Output Ch:0	0
1	2001:01	Input/Output Ch:1	0
2	2001:01	Input/Output Ch:2	0
3	2001:01	Input/Output Ch:3	0
4	2001:01	Input/Output Ch:4	0
5	2001:01	Input/Output Ch:5	0
6	2001:01	Input/Output Ch:6	0
7	2001:01	Input/Output Ch:7	0
8	6103:01	Filter Input Ch:0	0
9	6103:01	Filter Input Ch:1	0
10	6103:01	Filter Input Ch:2	0
11	6103:01	Filter Input Ch:3	0
12	6103:01	Filter Input Ch:4	0
13	6103:01	Filter Input Ch:5	0

■ The goal here is to provide a method of configuring the node without an external configuration software (here ADVANTIS)

■ if the value is different to 0 the value will be send during the initialisation phase

Step 2: Slave configuration 'EXPERT'

Step 2 : Slave Configuration 'EXPERT'



FTB 1CN08E08CM0: IP67, Digital 24 VDC I/O, 8 Input Points, 8 Input Points or Output Points (Configurable), 1.6 A (TEFTB05M01E.eds)

FTB_1CN08E08CM0
Channel 0

Error control | Config

Error Control

☐ Use Node Guarding Protocol
Guard Time: 0 msec. Life Time Factor: 2

☒ Use Heartbeat Protocol
Node Heartbeat Producer time: 300 msec.

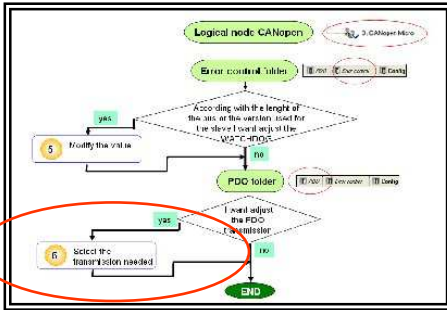
Through this service the NMT detect the error on the field bus, we can select Heartbeat or node guarding protocol, and adjust the Timeout

5

Step 2 : Slave Configuration 'EXPERT'

■ In this screen it is possible to map the PDO communication.

■ This screen is divided in 3 parts :



DCX170 CANopen (TEDCX170_0100E.eds)

PDO | Error control | Config

PDO	Tr.Type	InhibitTi...	Event Ti...	Symbol	Topo.Addr.	%M...	COBID	Index
<input checked="" type="checkbox"/> PDO 1(Static)	255	0	0				16#182	6041:00
Statusword	255		100	%Iw13.2v0.0.0.16				6041:00
<input checked="" type="checkbox"/> PDO 2(Static)	255		100	%Iw13.2v0.0.0.16				6041:00
Statusword	255		100	%IDv3.2v0.0.0.8				6064:00
<input checked="" type="checkbox"/> PDO 3(Static)	255		100	%Iw13.2v0.0.0.16				6041:00
Position actu...	255		100	%Iw13.2v0.0.0.16				6041:00
<input checked="" type="checkbox"/> PDO 4	254	0						
Statusword	254	0						
Velocity actua...	254	0						

Receive (%Q)

PDO	Tr.Type	InhibitTi...	Event Ti...	Symbol	Topo.Addr.	%M...	COBID	Index
<input checked="" type="checkbox"/> PDO 1(Static)	255						16#202	6040:00
Controlword	255			%Qw13.2v0.0.0.25				6040:00
<input checked="" type="checkbox"/> PDO 2(Static)	255			%Qw13.2v0.0.0.25				6040:00
Controlword	255			%Qw13.2v0.0.0.25				6040:00
Target position	255			%QDv3.2v0.0.0.14				607A:00
<input checked="" type="checkbox"/> PDO 3(Static)	255			%Qw13.2v0.0.0.25				6040:00
Controlword	255			%Qw13.2v0.0.0.25				6040:00
Target velocity	255			%QDv3.2v0.0.0.16				6081:00

Select Type, Inhibit or event on the PDO line in order to define the PDO communication :

Synchronous
Asynchronous

6

PDO 1 (Static)

Transmission Type

☐ Synchronous acyclic (0)

☒ Synchronous cyclic (1-240) * Sync period

☐ Asynchronous (Manuf. Event) (254)

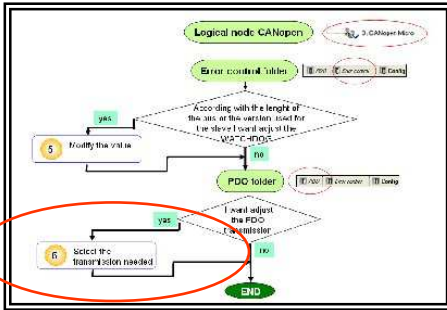
☐ Asynchronous (Profile Event) (255)

Properties

Inhibit time (0-65535) : * 100 us

Event timer (0-65535) : ms

OK Cancel



Step 2 : Slave Configuration 'EXPERT'

■ The PDO are exchanges according with the configuration done .(example transmission type)

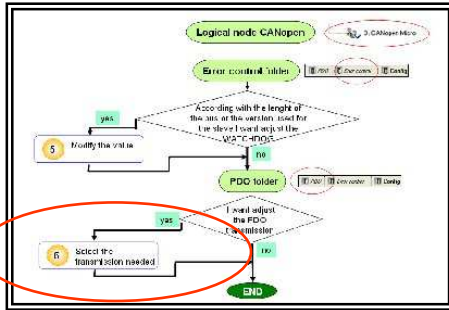
6

Synchronous acyclic PDO :

A transmission type of 0 means that the message shall be transmitted synchronously with the SYNC message but not periodically according to the value. Mainly used for drives.

Synchronous cyclic PDO :

A value between 1 and 240 means that the PDO is transmitted synchronously and cyclically, the transmission type value indicating the number of SYNC messages between two PDO transmissions.



Step 2 : Slave Configuration 'EXPERT'

■ The PDO are exchanges according with the configuration done .(example transmission type)

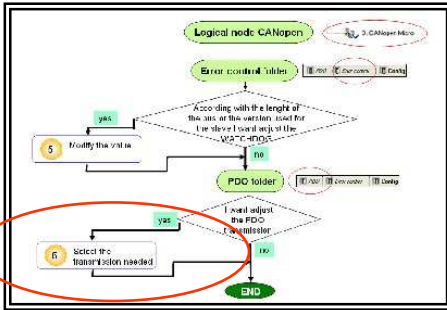
6

Asynchronous PDO:

The transmission type 254 means that the PDO is transmitted asynchronous. It is fully depending on the implementation in the device. Mainly used for digital I/O.

Asynchronous PDO:

The transmission type 255 means that the PDO is transmitted asynchronous when the value change.



Step 2 : Slave Configuration 'EXPERT'

■ The PDO are exchanges according to the configuration. (example transmission type)

6

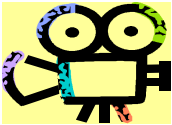
Inhibit time :

masks the communication during this time.

Nota: With the MFB we can only adjust this value according with the traffic and the length of the bus

Event timer :

Time to manage an event in order to start a PDO



Reminder : How to realize a complete configuration

- In this example we will configure one device :

LEXIUM 05

@2

Speed 500Kb

With

PDO4 In receive:

JOG-Activate

PDO4 in transmit

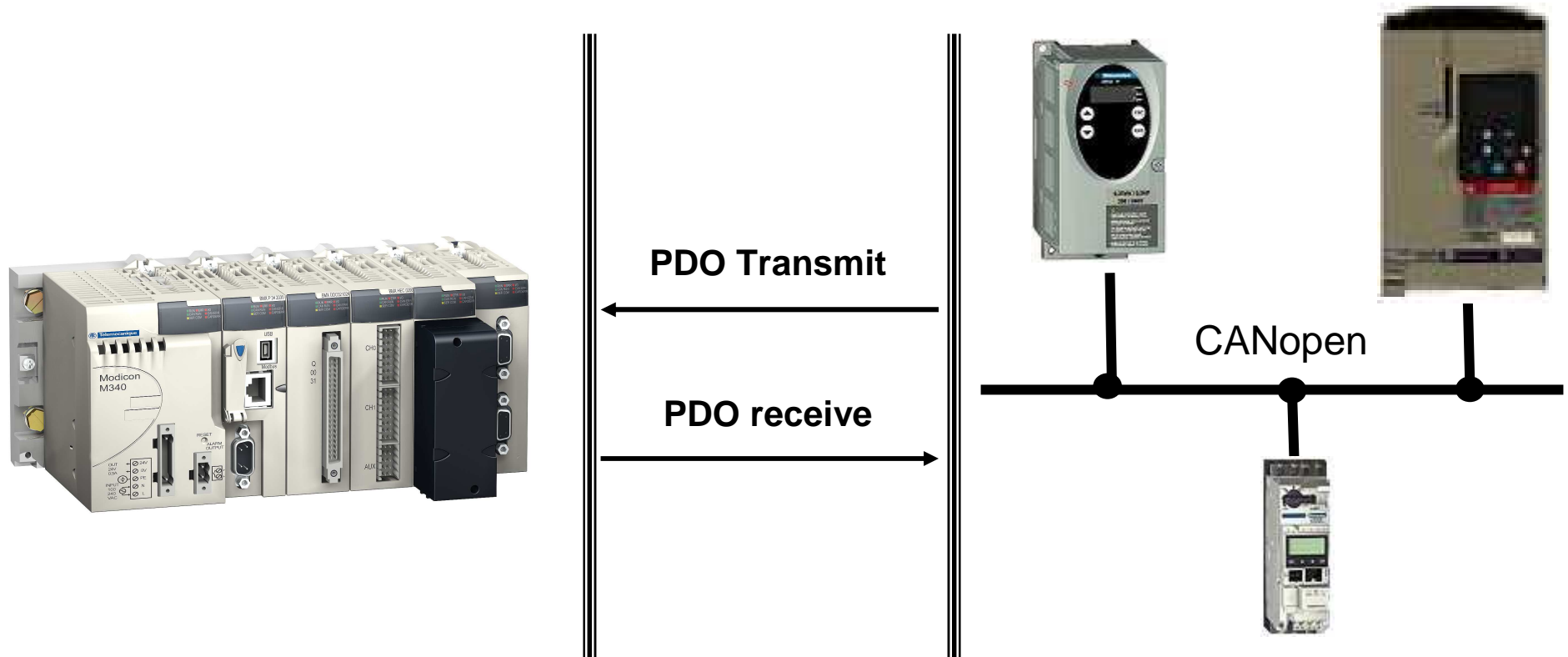
IO-act



D – Programming

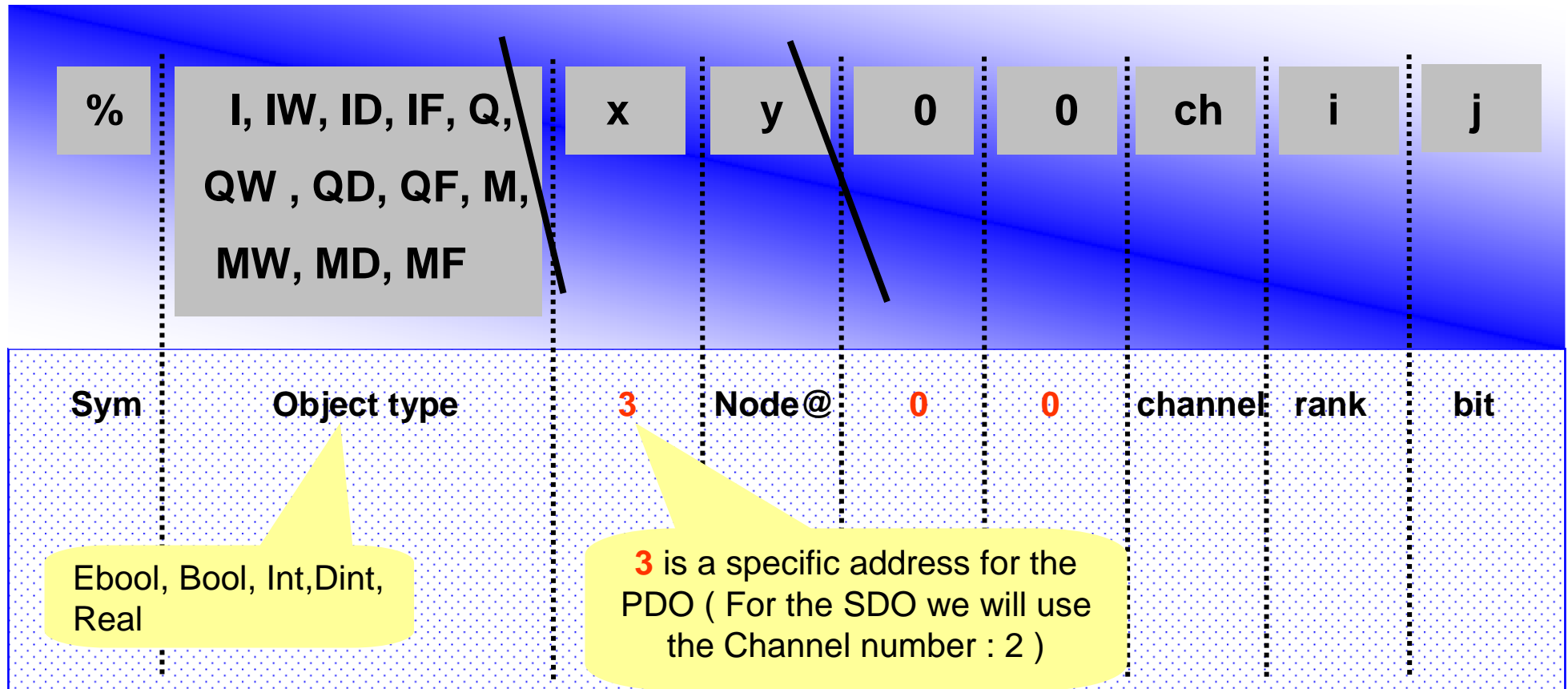
Management Implicit exchanges PDO

PDOs allow the implicit data management between the Master, in our case the M340 CPU and the slave by read/write words.



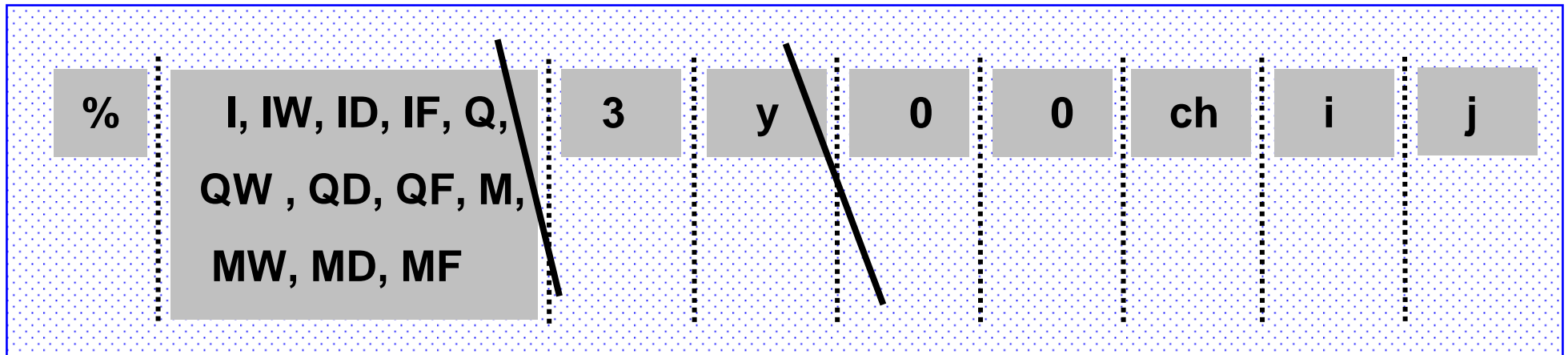
Addressing

- On Modicon M340, PDO variable addresses can be defined as word, double word and floating point types.



Addressing

■ Example.



1. Read an input on an ADVANTYS @ 4, channel 2, Word 3, bit2

%IW \ 3. 4 \ 0 . 0 . 2 . 3 . 2

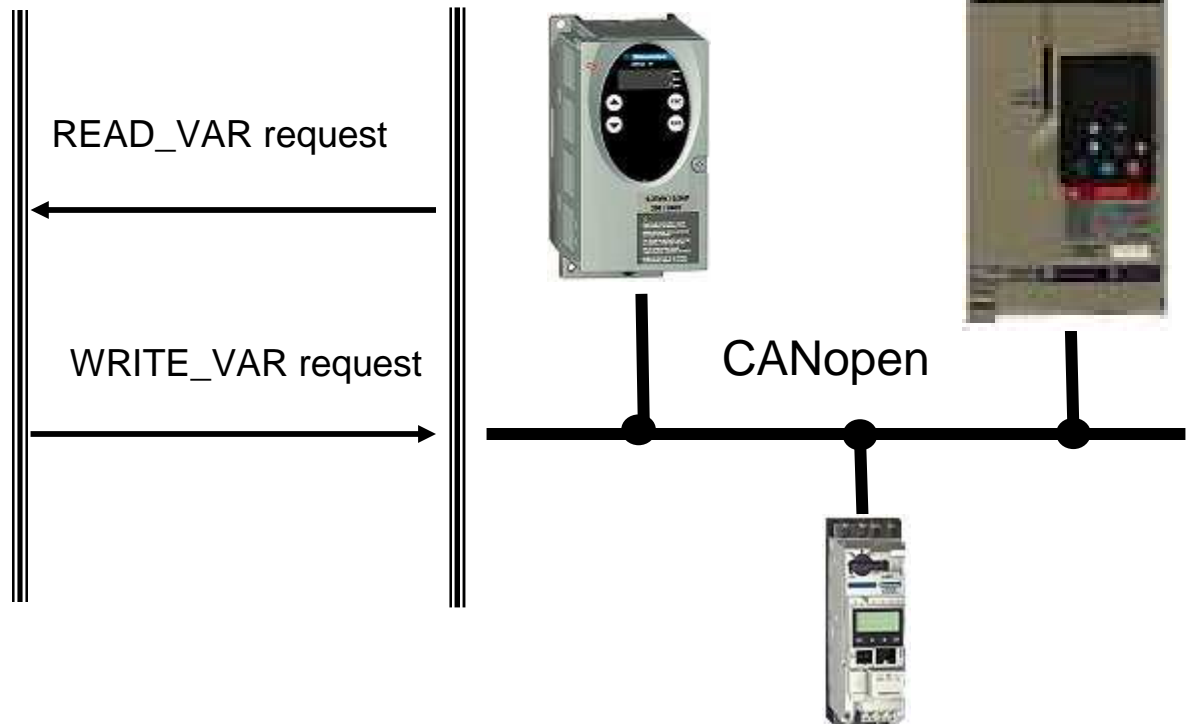
2. Write control word on an LEXIUM @ 9, channel 4, Word 3

%QW \ 3. 9 \ 0 . 0 . 4 . 3

Management explicit exchanges SDO

SDOs allow the management of the explicit data between the Master, in our case the M340 and the slave using of request `READ_var` / `WRITE_var`

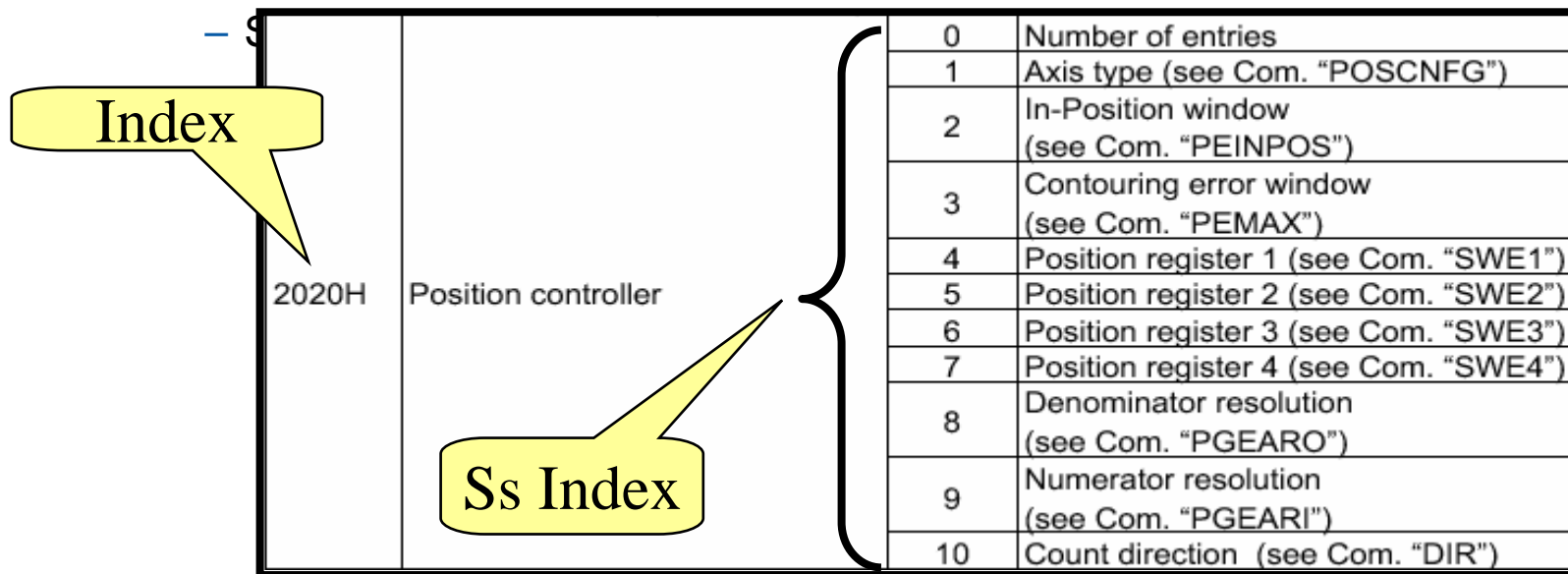
PLC Management of the request



Management explicit exchanges SDO

INDEX / Ss INDEX Concept .

- All requests on CANopen are defined with an INDEX / Ss INDEX .
- Here is an example on a CANopen device ' LEXIUM 15x '
 - INDEX : Define the function wished .
 - Ex : 2022 => Position data for the ' position ' mode .
 - 2020 => Position Controller



Management explicit exchanges SDO

- Syntax to program a SDO (Write var) :

```
If write_SDO then
    buffer_gest_write[2] := 50;    (*Time-out*)
    buffer_gest_write[3] := 4;    (*Longueur*)
    (*buffer_sent[0] := 200;*)

    WRITE_VAR (ADR := ADDM('0.0.2.2'),
               OBJ := 'SDO',
               NUM := 16#00001005,
               NB := 2,
               EMIS := buffer_sent,
               GEST := buffer_gest_write) ;

    write_SDO := false;
end_if;
```

Parameters	descriptions
ADDM('0.0.2.node')	Address of the exchange destination 0.0 : processor slot in the rack (0) 2 : channel (always 2 for CANopen) 2: Device adress.
'SDO'	SDO object type (always SDO in capitals)
Subindex / index:	Double word or immediate value identifying the CANopen SDO index or subindex:
NodeID	Word or value identifying the destination device on the CANopen bus
%MWi:L	Table of words containing the data to be sent (minimum length = 1)
%MWk:4	Exchange management parameters

Management explicit exchanges SDO

- Syntax to program a SDO (Read var) :

```

IF read_SDO then
    buffer_gest_read[2] := 50; (*Time-out*)
    buffer_rec[0] := 0;

    READ_VAR
        (ADR := ADDM('0.0.2.2'),
         OBJ := 'SDO',
         NUM := 16#00001005,
         NB := 2,
         GEST := buffer_gest_read,
         RECP => buffer_rec);

    read_SDO := false;
END_IF;
    
```

Parameters	descriptions
ADDM('0.0.2.node')	Address of the exchange destination 0.0 : processor slot in the rack (0) 2 : channel (always 2 for CANopen) 2: Device adress.
'SDO'	SDO object type (always SDO in capitals)
Subindex / index:	Double word or immediate value identifying the CANopen SDO index or subindex:
NodeID	Word or value identifying the destination device on the CANopen bus
%MWk:4	Exchange management parameters
%MWi:L	Table of words containing the data to be received (minimum length = 1)

Management explicit exchanges SDO

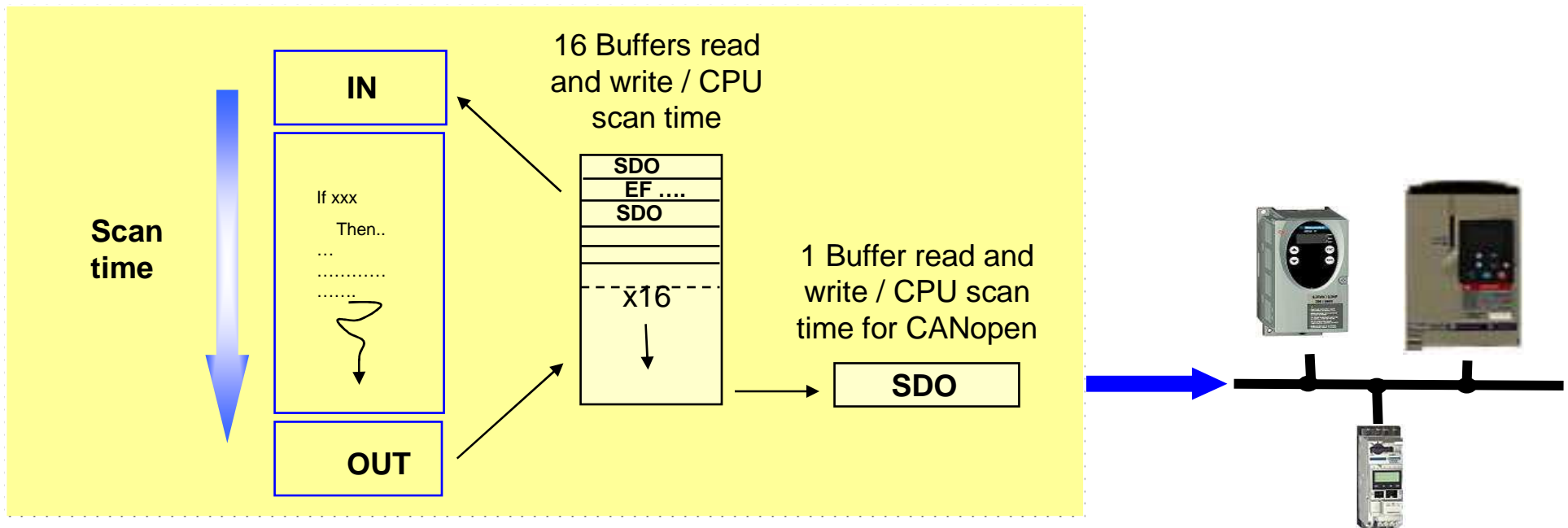
■ Performances :

■ For all the processors we can manage :

16 messages in Input / Output by scan time of the MAST task.

■ For CANopen message channel (SDO) we are limited to :

1 message in Input / Output by scan time of the MAST task.



E – Debug & Diagnostic

Debug and Diagnostic Screen

- On the processor it's possible to realize a quick diagnostic on the CANopen bus

Configuration Tree:

- Station
 - Configuration
 - 0: PLC bus
 - 0: BMX XBP 0800
 - (P) (P): BMX CPS 2000
 - 0: BMX P34 2030
 - CANopen**

CANopen comm head:

- Channel 2 (Fault)

Who provides the fault

CANopen Slaves status:

Adr.	Device Name	State	NMT	Emcy
2	STB_NCO_2212	Configured	Operational	
3	Lexium05	Failed	Unknown	

What's fault !

Fault Details:

- Internal fault
- External fault
 - Slave error (one or more slaves not running or in fault condition)
 - Card not present, not ready or fatal error on card
- Other fault
 - X0=1: channel error, OR logic between bits X8 to X15

Debug and Diagnostic Screen

The screenshot displays the 'Project Browser' window on the left and the 'CANopen Tree' diagnostic screen on the right.

Project Browser (Left):

- Structural view
- Station
 - Configuration
 - 0: PLC bus
 - 0: BMX XBP 0800
 - (P) (P): BMX CPS 2000
 - 0: BMX P34 2030
 - CANopen
 - Ethernet
 - 1: BMX NOE 0100
 - 2: BMX DDI 1602
 - 3: BMX DDO 3202K
 - 4: BMX AMI 0410
 - 5: BMX AMO 0210
 - 6: BMX EHC 0200
 - 7
 - 3: CANopen Micro
 - 2: CANopen drop
 - 0.0: STB_NCO_2212
 - 3: CANopen drop
 - Derived Data Types
 - GD
 - Derived FB Types
 - Variables & FB instances
 - Motion
 - Communication
 - Program
 - Animation Tables
 - Operator Screens
- Ex_3_IO exercise

CANopen Tree (Right):

- Bus: 3
- CANopen comm head: 01.00
- Connections configured: 2
- Diagram showing a CANopen network with nodes 2 and 3. Node 2 is connected to a rack of modules labeled 'STB NCO 2212'. Node 3 is connected to a unit labeled 'LEXIUM05'.

■ We can also obtain a quick diagnostic on the CANopen bus with the CANopen Tree.

Debug and Diagnostic Screen

■ On BUS editor It is possible to :

- Read and Write all the SDO through the INDEX / SUB INDEX of the device.
- Read the Emergency message (Information provides by the device)
- Read the I/O object Available on the device.
- Read and Write all the PDO used.

STB NCO 2212 CANopen Network Interface Module [STBNCO2212.eds]

Operational Emergency

STB_NCO_2212

Channel 0

Overview CANopen Fault I/O objects

STB_NCO_2212

CANopen features	Conformance class		S10
	Standard		DS 301 V4.02, DR 30
	Profile		-
	Specials		Supports CAN bus ex external CANopen pr Allows hot swapping modules
Structure	Physical interface		SUB-D 9-way male
	Data rate	Kbps	10, 20, 50, 125, 25
	Medium		Shielded dual twist
CANopen communication device	Operating temperature	°C	0...+ 60
	Degree of protection		IP 20
	LED indicators		Network state (CAN Module and islands ERROR and TEST)
	Product certification		CEI/EN 61131-2, UL FM Classe 1 Divisio
Max. number of I/O modules per island	Power supply	VDC	24
Number of segments	Principal		32
	Extension		1
I/O module	Discrete inputs		6
			24 VDC (2, 4, 6 or 115 VAC and 230 VAC

If you select
the NIM we
can

Debug and Diagnostic Screen

- On BUS editor It is possible to :
 - **Read and Write all the SDO through the INDEX / SUB INDEX of the device.**
 - Read the Emergency message (Information provides by the device)
 - Read the I/O object Available on the device.
 - Read and Write all the PDO used.

STB_NCO_2212
Channel 0

Request to send

Request to send: Read SDO

Index: 16# 100C

Subindex: 16# 100C

Value: (120 bytes max.) 16#

Parameter name: Guard Time

Parameter size (Byte): 2

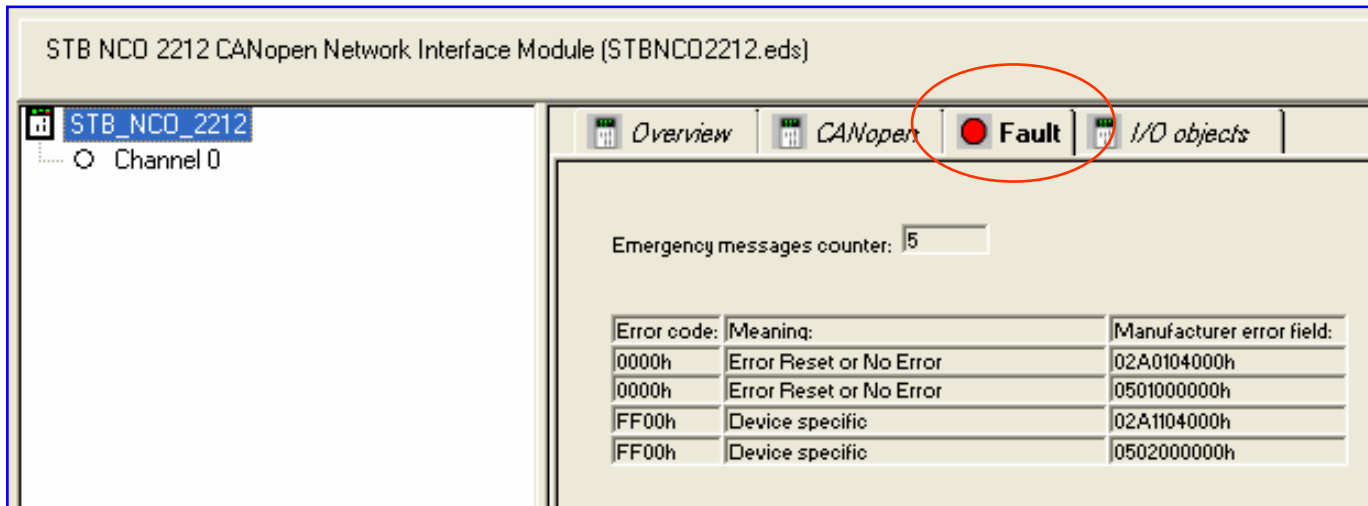
Status

Response received:

In this screen the Index and Sub index are provide by the system (according with the EDS file) with a comment associated

Debug and Diagnostic Screen

- On BUS editor Tree It is possible to :
 - Read and Write all the SDO through the INDEX / SUB INDEX of the device.
 - **Read the Emergency message (Information provides by the device)**
 - Read the I/O object Available on the device.
 - Read and Write all the PDO used.



Debug and Diagnostic Screen

■ On BUS editor It is possible to :

- Read and Write all the SDO through the INDEX / SUB INDEX of the device.
- Read the Emergency message (Information provides by the device)
- **Read the I/O object Available on the device.**
- Read and Write all the PDO used.

STB NCO 2212 CANopen Network Interface Module (STBNCO2212.eds)

Operational Emergency

STB_NCO_2212
Channel 0

Overview CANopen Fault **I/O objects**

I/O variable creation

Prefix for name:

Type:

Comment:

I/O Objects

Channel: ☒ %CH

Configuration: ☐ %KW ☐ %KD ☐ %KF

System: ☐ %MW

Status: ☐ %MW

Parameter: ☐ %MW ☐ %MD ☐ %MF

Command: ☐ %MW ☐ %MD ☐ %MF

Implicits: ☒ %I ☒ %W ☒ %ID ☒ %IF ☐ %ERR
☒ %Q ☒ %QW ☒ %QD ☒ %QF

Update

	Address	Name	Type	Comment	%M
1	%CH\3.2\0.0.0				
2	%Iw\3.2\0.0.0		INT		%Mw
3	%Iw\3.2\0.0.0.1		INT		%Mw
4	%Iw\3.2\0.0.0.2		INT		%Mw
5	%Iw\3.2\0.0.0.3		INT		%Mw
6	%Iw\3.2\0.0.0.4		INT		%Mw
7	%Iw\3.2\0.0.0.5		INT		%Mw
8	%Iw\3.2\0.0.0.6		INT		%Mw
9	%Iw\3.2\0.0.0.7		INT		%Mw
10	%Iw\3.2\0.0.0.8		INT		%Mw
11	%Iw\3.2\0.0.0.9		INT		%Mw
12	%Iw\3.2\0.0.0.10		INT		%Mw
13	%Iw\3.2\0.0.0.11		INT		%Mw
14	%Iw\3.2\0.0.0.12		INT		%Mw
15	%Iw\3.2\0.0.0.13		INT		%Mw
16	%Iw\3.2\0.0.0.14		INT		%Mw
17	%Iw\3.2\0.0.0.15		INT		%Mw
18	%Iw\3.2\0.0.0.16		INT		%Mw
19	%Iw\3.2\0.0.0.17		INT		%Mw
20	%Iw\3.2\0.0.0.18		INT		%Mw
21	%Iw\3.2\0.0.0.19		INT		%Mw
22	%Iw\3.2\0.0.0.20		INT		%Mw
23	%Iw\3.2\0.0.0.21		INT		%Mw
24	%Iw\3.2\0.0.0.22		INT		%Mw
25	%Iw\3.2\0.0.0.23		INT		%Mw
26	%Iw\3.2\0.0.0.24		INT		%Mw

Debug and Diagnostic Screen

- On BUS editor Tree It is possible to :
 - Read and Write all the SDO through the INDEX / SUB INDEX of the device.
 - Read the Emergency message (Information provides by the device)
 - Read the I/O object Available on the device.
 - **Read and Write all the PDO used.**

STB NCO 2212 CANopen Network Interface Module (STBNCO2212.eds)

STB_NCO_2212

- Channel 0

If you select the channel you can

PDO Error control Config **Debug**

	Reference	Symbol	Value
125	%IW43.2\0.0.0.125		0
126	%IW43.2\0.0.0.126		0
127	%IW43.2\0.0.0.127		0
128	%IW43.2\0.0.0.128		0
129	%IW43.2\0.0.0.129		0
130	%IW43.2\0.0.0.130		0
131	%QW43.2\0.0.0.0		1
132	%QW43.2\0.0.0.1		0
133	%QW43.2\0.0.0.2		0
134	%QW43.2\0.0.0.3		0
135	%QW43.2\0.0.0.4		0

CANOpen

