



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Description of the program example “M340_LXM32M_PLCopenMapping_Vxx”

Version V001

This document describes a small application using a M340-PLC and one LXM32M drive using the PLCopen profile on CANopen.

It is not intended to replace any specific product documentation. On the contrary, it offers additional information to the product documentation, for installing, configuring and starting up the system. A detailed functional description or the specification for a specific user application is not part of this document. Nevertheless, the document outlines some typical applications where the system might be implemented.

The program example includes how to activate the operation modes:

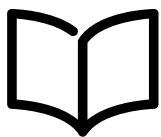
- Homing
- Profile Velocity mode
- Profile Torque mode
- Point to point
- Manual mode
- How to write and read parameters



CANopen

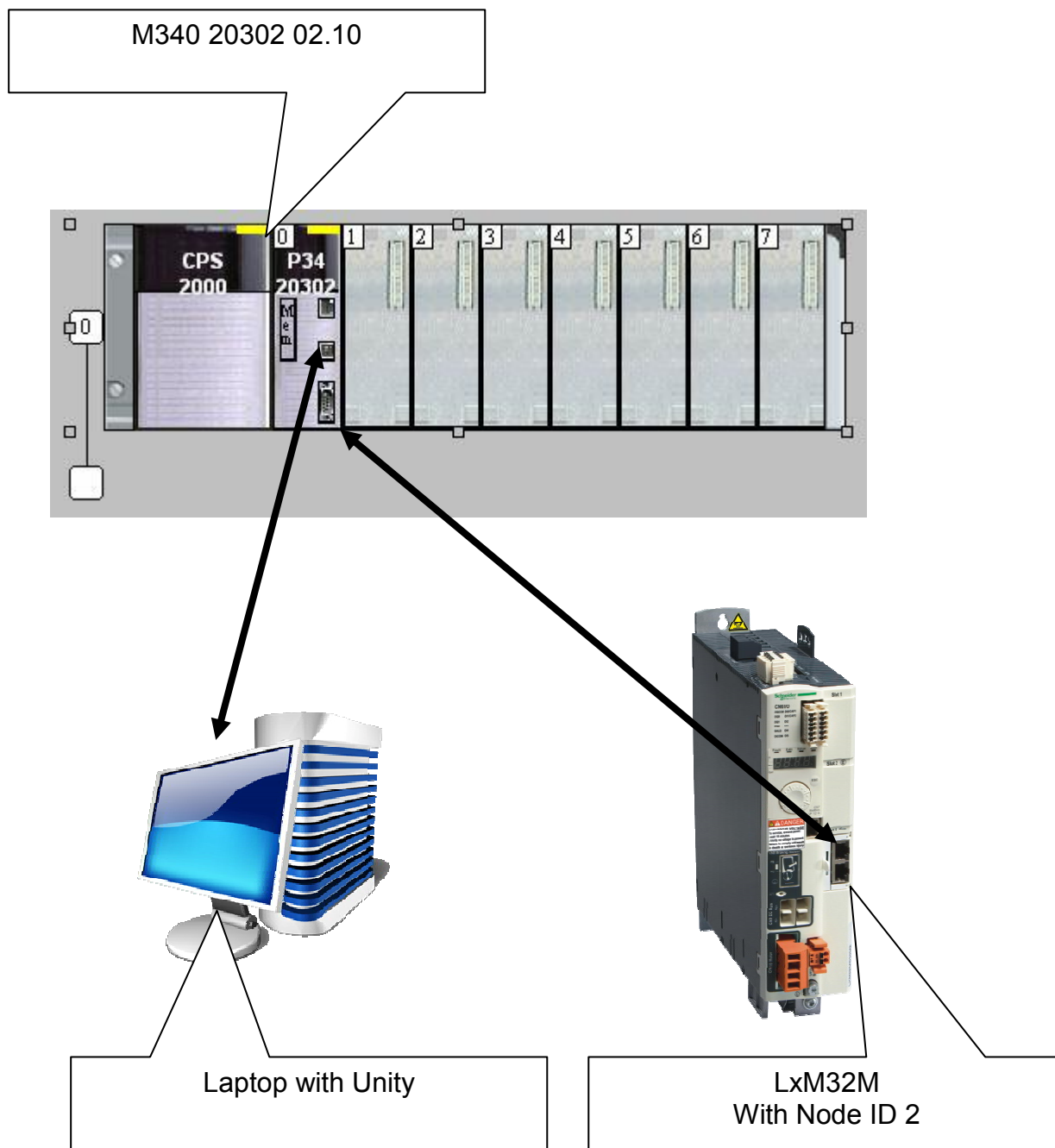


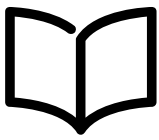
Please contact your local Schneider Electric partner.
www.schneider-electric.com



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

System configuration





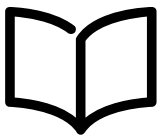
Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Configuration of Lexium 32 M

To configure LxM32M the commissioning tool “Lexium Control Tool” can be used.

For this program example the CAN address must be 2, the baud rate is 250 kBaud.

Name	Wert	Einheit	Beschreibung	Bereich	Modbus
DEVcmdIntert	Fieldbus Control Mode		Festlegung der Steuerungsart	0..3	1282
PTI_signal_type	A-B Signals		Auswahl der Signalart für die PTI-Schnittstelle	0..2	1284
PTO_mode	Off		Verwendungsart der PTO-Schnittstelle	0..4	1342
ESIM_scale	4096	EncInc	Auflösung der Encoder-Simulation	8..65535	1322
CTRL_v_max	13200	usr_v	Geschwindigkeitsbegrenzung	1..2147483647	4384
CTRL_l_max	18,00	Arms	Strombegrenzung	0..300	4376
LIM_l_maxQSTP	18,00	Arms	Strombegrenzung für Quick Stop	0,01..300	4378
LIM_l_maxHalt	18,00	Arms	Strombegrenzung für Halt	0,01..300	4380
_lmax_system	11,80	Arms	Strombegrenzung des Systems	..	7246
MOD_Enable	Modulo Off		Activation of Modulo	0..1	1648
InvertDirOfMove	Inversion Off		Umkehr der Bewegungsrichtung	0..1	1560
SimAbsolutePos	Simulation Off		Simulation of absolute position at power cycling	0..1	1350
ENC_abs_source	Encoder 1		Source for setting absolute encoder position	0..1	1354
Mains_reactor	No		Netzdrossel	0..1	1344
ShiftEncWorkRang	Off		Arbeitsbereich des Encoders verschieben	0..1	1346
CANaddress	2		CANopen Adresse (Knotennummer)	1..127	16644
CANbaud	250 kBaud		CANopen Baudrate	50..1000	16646
MOD_Min	0	usr_p	Minimum position of modulo range	..	1650
MOD_Max	3600	usr_p	Maximum position of modulo range	..	1652
MOD_AbsDirection	Shortest Distance		Direction of absolute movement with Modulo	0..2	1654
MOD_AbsMultiRng	Multiple Ranges Off		Multiple ranges for absolut movement with Modulo	0..1	1656



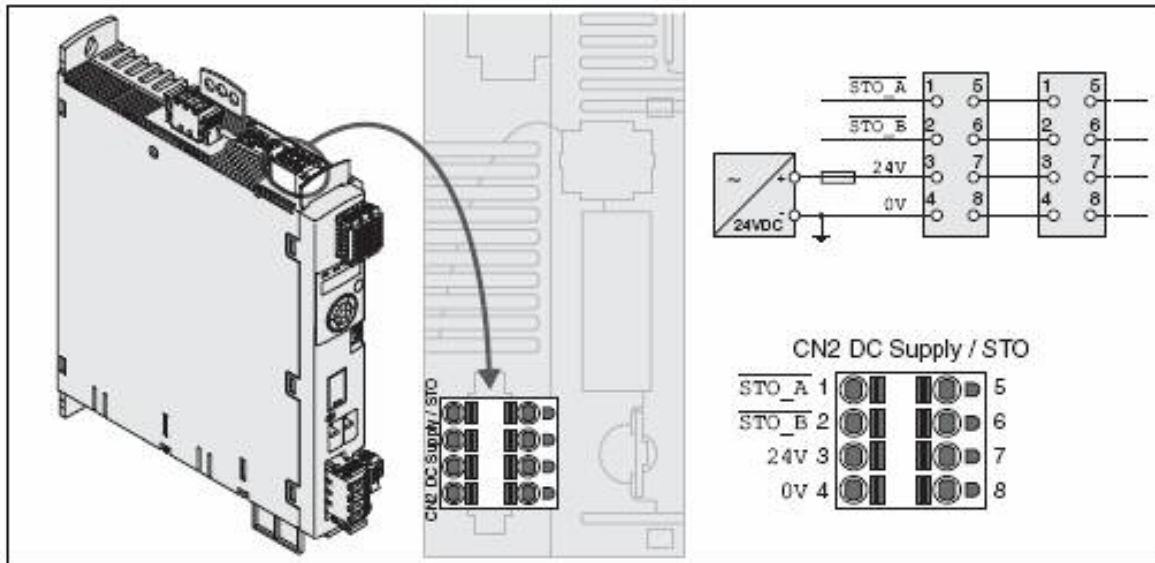
Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Wiring of Lexium 32 M

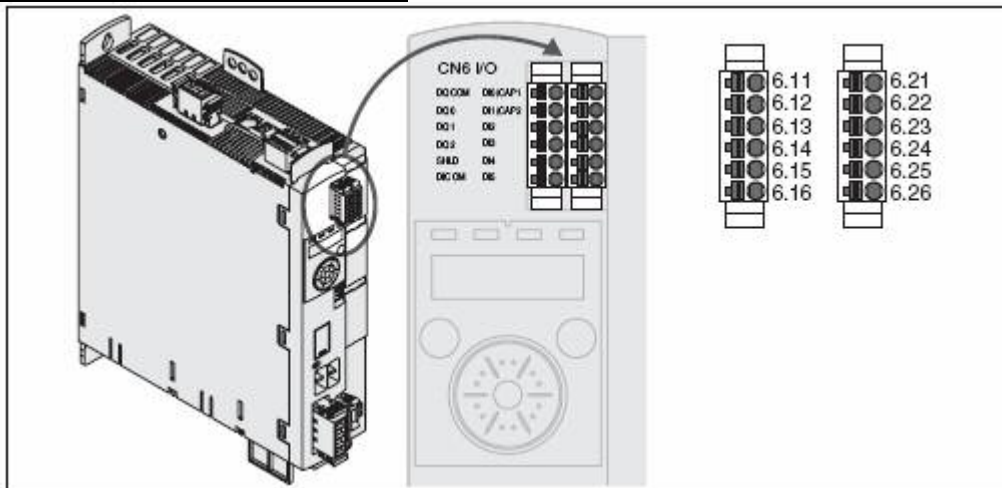
For the minimum configuration the power supply and the fieldbus must be connected. Very important are the inputs for safety torque off STO_A and STO_B. If the limit switches are enabled they also need to be connected, otherwise they could be disabled with the commissioning tool.

If the wiring is ok the drive will go automatically to state 4 (rdy).

24VDC and STO:



Digital 24V-inputs/outputs:



Inputs DI2 and DI3 are configured as LIMN and LIMP. They have connected to Limit switches, or 24VDC or disabled.

IOfuncn_DI2	Positive Limit Switch (LIMP)	Function Input DI2
IOfuncn_DI3	Negative Limit Switch (LIMN)	Function Input DI3



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

PLCopen interface of Lexium 32M

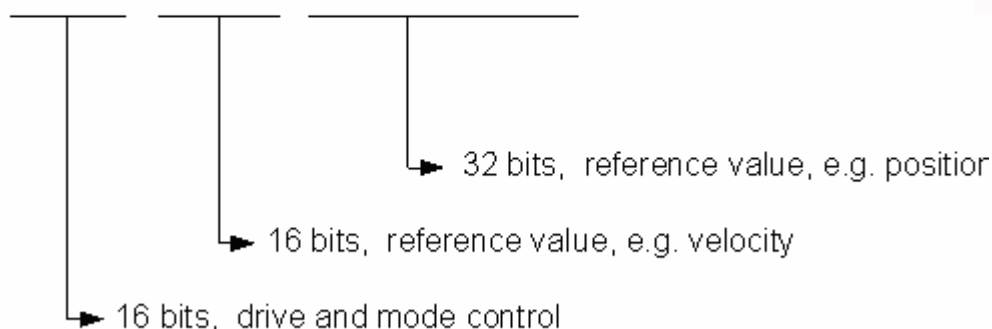
There are two possibilities to control the drive via CANopen. First possibility is to use it according DSP402. In this case the standard eds file must be used. The different modes of operations must be activated via the object 6060:0h and it will take several plc cycles.

Second possibility is to control the drive with the PLCopen profile. Therefore a special eds is needed. The different modes of operations can be started within one plc cycle.

From plc to drive

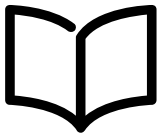
Byte

1	2	3	4	5	6	7	8
dmCtrl		refA16		refB32			



The word dmControl (drive and mode control) is a 16 bit value which consists of driveCtrl and modeCtrl.

dmControl																
driveCtrl								modeCtrl								
CU	CH	SH	0	FR	QS	EN	DS	MT	action	operation mode (int5)						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

DriveCtrl

driveCtrl controls the state machine of the drive. All bits are edge-sensitive. The relevant function is triggered with a rising edge. The modification is processed immediately.

Bit							
CU	CH	SH	0	FR	QS	EN	DS
15	14	13	12	11	10	9	8

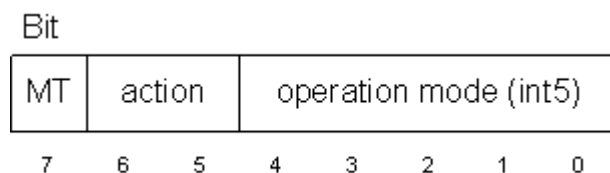
DS (8)	Disable
EN (9)	Enable
QS (10)	Quick Stop
FR (11)	Fault Reset
SH (13)	Set HALT
CH (14)	Clear HALT
CU (15)	Continue



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

ModeCtrl

Control of the operating modes is via modeCtrl.



CIA402 operating modes:

- 1 Profile position
- 2 not supported
- 3 Profile velocity
- 4 Torque profile
- 6 Homing

Manufacturer operating modes:

- 1 Jog
- 2 Electronic Gear Modes
- 3 Motion Sequence Mode

action: operating mode specific action

MT: bit mode toggle
(switch operating mode and operating mode specific reference values refA16 and refB32)



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

CIA402 operating modes:

mode	action	modeCtrl *(1) Bit 0..6	Description	reference value refA16, refA32	reference value refB32
1 (Profile Position)	0	0x01	Absolute positioning	Reference speed *(4)	Reference position [usr]
	1	0x21	Relative positioning with reference to current set target position	Reference speed *(4)	Reference position [usr]
	2	0x41	Relative positioning with reference to current motor position	Reference speed *(4)	Reference position [usr]
3 (Profile Velocity)	0	0x03	+10V analogue	*(3)	*(3)
	1	0x23	Reference value	Reference speed *(4)	*(3)
4 (Profile Torque)	0	0x04	+10V analogue	*(3)	*(3)
	1	0x24	Reference value	Reference current [0.01A]	*(3)
6 (Homing Mode)	0	0x06	Position setting	*(3)	Dimension setting position [usr]
	1	0x26	Reference movement	Type	*(3)

Manufacturer operating modes:

mode	action	modeCtrl *(1) Bit 0..6	Description	reference value refA16, refA32	reference value refB32
-1 / 0x1F (Jog Mode)	0	0x1F	classic inching	Operation (direction and rotation speed selection)	*(3)
-2 / 0x1E (Electronic Gear Mode)	0	0x1E	Position gear mode: Immediate synchronisation	GFAC denominator *(2)	GFAC numerator
	1	0x3E	Position gear mode: Compensation movement	GFAC denominator *(2)	GFAC numerator
	2	0x5E	Speed gear mode	GFAC denominator *(2)	GFAC numerator
-3 / 0x1D (Motion Sequence Mode)	0	0x1D	Sequential mode: Select data set	Data set number	Bit 0: 1 = takeover data set number
	2	0x3D	Direct mode: Start data set	Data set number	*(3)



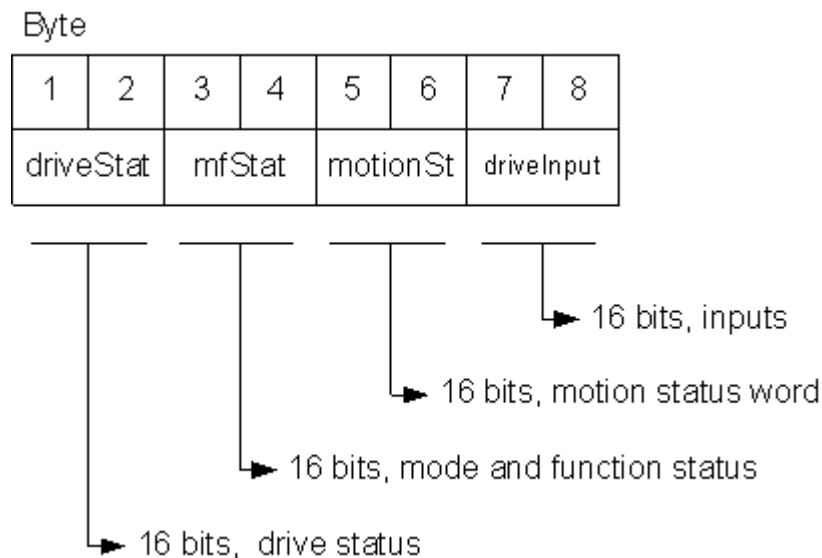
Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

- *(1) Column corresponds to a value to be entered in byte 'modeCtrl', but without bit MT.
- *(2) Special treatment for gear factor (GFAC) when denominator is 0:
Only the gear numerator will be changed.
The current active gear denominator value will be used for defining the new gear ratio.
- *(3) This value is not relevant for the operating mode. The drive ignores this value.
- *(4) The unit of the reference speed depends on the user normalisation. If maximal value of the reference speed cannot be coded within the available number range of refA16, the drive calculates a shift factor. With this additional shift factor the maximal value of the reference speed can be coded within the available number range of refA16. In this case, the reference and the actual speed will be transferred with the additional shift factor. After enabling the drive, the master has to read the actual shift factor from the drive.



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

From drive to plc



Drive Stat

driveStat consists of the status of the state machine, warning and error bits and the status of the current axis operating mode.

Bit

X_ERR	X_END	X_ADD1	res.	res.	QS	RF	HALT
15	14	13	12	11	10	9	8

Bit

warn	fault	res.	res.	state			
7	6	5	4	3	2	1	0

state: State of state machine
 fault: Fault active (see StopFault)
 warn: Warning active (see LastWarning)
 QS: Quick Stop requested
 RF: REF-OK (drive has reference position set)
 HALT: HALT request active
 X_ADD1: Additional info of axis mode
 X_END: End bit of axis mode
 X_ERR: Fault bit of axis mode



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Mode and function status

The current processing status of the operating modes is shown in the field mfStat:

Bit

MT	ME	DE	operating mode				
7	6	5	4	3	2	1	0

mode: Active operating mode
(coding see modeCtrl)

DE: Data error
(error on writing toggle bit independent parameter)

ME: Mode error
(error on writing toggle bit dependent parameter e.g. acceleration
outside value range)

MT: Bit mode toggle
(mirror of bit MT)

Structure of mfStat (status of specific functions) – high word

Bit

res.	res.	res.	res.	CAP2 1	CAP2 0	CAP1 1	CAP1 0
15	14	13	12	11	10	9	8

CAP1_1/CAP1_0: counter of the position capture by CAP1

CAP2_1/CAP2_0: counter of the position capture by CAP2



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

MotionStat

The following relevant bits of the action word are displayed.

res.	CNST	ACC	DEC	TAR0	res.	PWIN	MOTN
15	14	13	12	11	10	9	8
MOTP	MOTZ	res.	res.	res.	res.	res.	res.
7	6	5	4	3	2	1	0

MOTZ: Motion zero: actual speed is zero
MOTP: Motion positive: motor turns in positive direction
MOTN: Motion negative: motor turns in negative direction
PWIN: Inside position window
TAR0: Profile generator: target speed is zero
DEC: Profile generator: Deceleration
ACC: Profile generator: Acceleration
CNST: Profile generator: constant movement

DriveInput

The current level of the inputs is displayed.

15	14	13	12	11	10	9	8
LI8	LI7	LI6	LI5	LI4	LI3	LI2	LI1
7	6	5	4	3	2	1	0

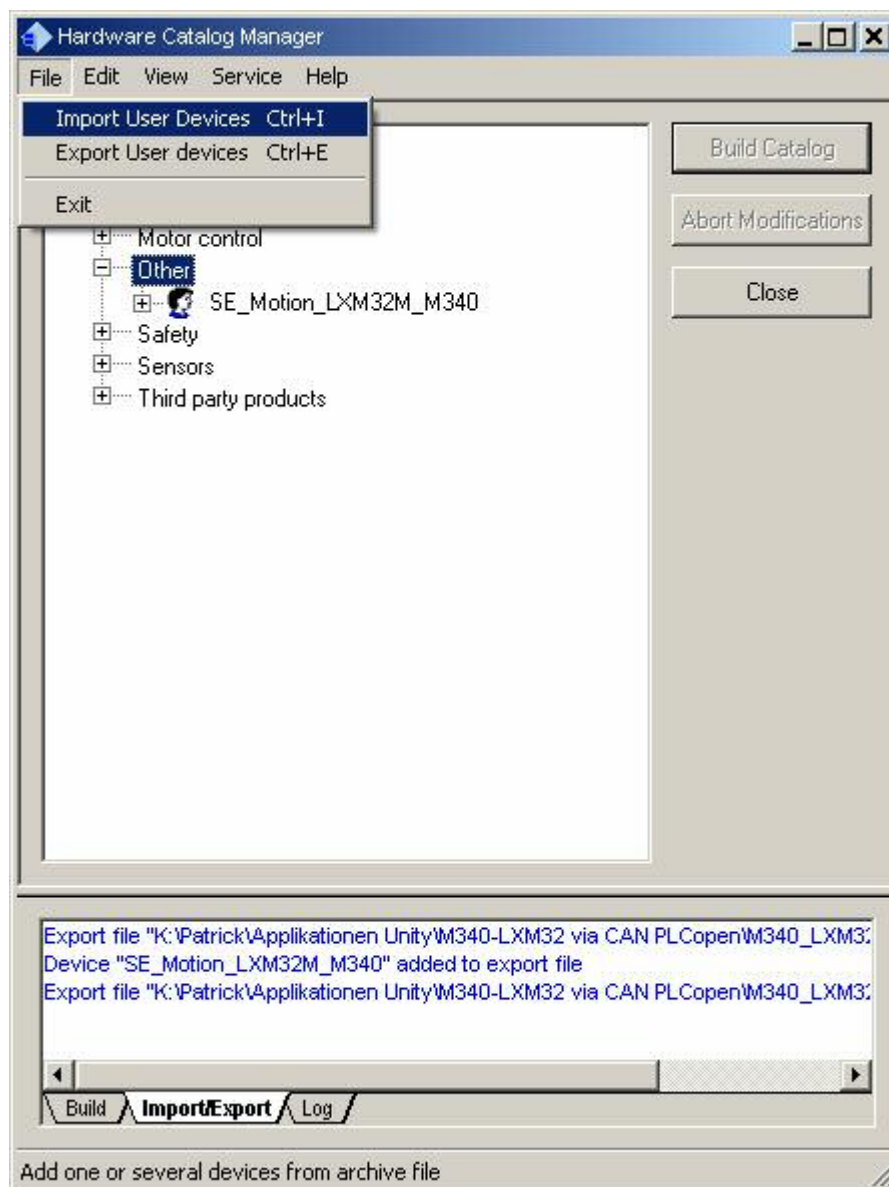
If the signals are forced, the levels of the corresponding force assignments will be stored here.



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

CANopen configuration in Unity

At first the user device *M340_LXM32M_PLCopen.cpx* must be imported into the Hardware Catalog Manager of Unity.





Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

After building the catalog the drive can be used in the can configuration.

The screenshot displays the Unity Pro XL software interface. The main window shows the 'Project Browser' on the left, listing the project structure: Configuration, Derived Data Types, Derived FB Types, Variables & FB instances, Motion, Communication, Program, Tasks, and MAST. The 'Configuration' section is expanded, showing the PLC bus configuration. The 'CANopen' window is open, showing the 'Bus' configuration. The 'Bus' is set to 3, and the 'CANopen comm head Expert' is selected. The 'Connections configured' count is 0. The 'New Device' dialog box is open, showing the 'Topological Address' [1..63] and 'Node-ID' 1. The 'Part Number' and 'Description' table is visible, listing various components including 'SE_Motion_LXM32M_M340' which is highlighted.

Part Number	Description
SE_Motion_LXM32M_M340	EDS for PLCopen library (SE_Motion_LXM32M_V1002.EDS)



Program example M340 Unit ↔ Lexium 32M with Canopen using the PLCopen profile

It is mandatory to map RxPDO1 and TxPDO1. The mapping of TxPDO2 is optional.

EDS for PLCopen library (SE_Motion_LXM32M_V1002.EDS)

SE_Motion_LXM32M_M340
Channel 0

Transmit (%I) ☐ Display only active PDO

PDO	Tr. Type	InhibitTime	Event Tim...	Symbol	Topo.Addr.	%M...	COBID	Index
<input checked="" type="checkbox"/> PDO 1	255	0	0				16#182	
driveStat				%dV3.2v0.0.0.21	%Mv21			301B:25
mfStat				%dV3.2v0.0.0.22	%Mv22			301B:26
motionStat				%dV3.2v0.0.0.23	%Mv23			301B:27
driveInput				%dV3.2v0.0.0.24	%Mv24			301B:28
<input checked="" type="checkbox"/> PDO 2	255	0	100				16#282	
Statusword				%dV3.2v0.0.0.27	%Mv27			6041:00
Position actual...				%dV3.2v0.0.0.12	%Mv12			6064:00
<input checked="" type="checkbox"/> PDO 3	255	0	100				-	
Statusword				%dV3.2v0.0.0.27	%Mv27			6041:00
Velocity actual...				%dV3.2v0.0.0.14	%Mv14			606C:00
<input checked="" type="checkbox"/> PDO 4	254	0	0				-	

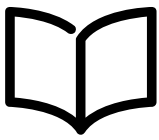
Receive (%Q) ☐ Display only active PDO

PDO	Tr. Type	InhibitTime	Event Tim...	Symbol	Topo.Addr.	%M...	COBID	Index
<input checked="" type="checkbox"/> PDO 1	255						16#202	
driveModeCtrl				%QV3.2v0.0.0.22	%Mv54			301B:1F
refA16				%QV3.2v0.0.0.23	%Mv55			301B:22
refB32				%QD3.2v0.0.0.0	%Mv32			301B:21
<input checked="" type="checkbox"/> PDO 2	255						-	
Controlword				%QV3.2v0.0.0.24	%Mv56			6040:00
Target position				%QD3.2v0.0.0.6	%Mv38			607A:00
<input checked="" type="checkbox"/> PDO 3	255						-	
Controlword				%QV3.2v0.0.0.24	%Mv56			6040:00
Target velocity				%QD3.2v0.0.0.16	%Mv48			60FF:00
<input checked="" type="checkbox"/> PDO 4	254						-	

Function: Default

Variable: ☐ Dis

Param
RAM
_IO_a
_IO_C
_IO_C
_IO_DC
_Cap1
_Cap1
_Cap1
_IOGa
MSM
driveh
refB3
refA16
driveS
mfSta
motio
driveI
_actic
_I_act
_p_ac
_p_ac
_RAN
GEAF
GEAF
Contr
Status
Mode
Mode
positi
Positi
Veloc
Targe
Torqu
Targe
Profil



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

The bootup configuration is different from the standard settings.

EDS for PLCopen library (SE_Motion_LXM32M_V1002.EDS)

SE_Motion_LXM32M_M340
Channel 0

PDO Error control **Bootup** Object Dictionary Configuration

Bootup procedure configuration

Power ON

Restore

- ☒ No Restore
- ☐ Restore communication parameters (1011:02)
- ☐ Restore application parameters (1011:03)
- ☐ Restore all parameters (1011:01)

Reset option is used to define the allowed restore procedure for a CANopen device during startup.
Communication parameters: 0x1000 -> 0x1FFF
Application parameters: 0x6000 -> 0x9FFF

Reset

- ☐ Reset communication parameters
- ☒ Reset node

Reset option is used to configure the type of reset sent to the device.
Communication parameters: 0x1000 -> 0x1FFF
Application parameters: 0x6000 -> 0x9FFF

Check node

- ☒ Device type (1000:00)
- ☒ Device identity (1018:xx)

Check node option is used to bypass the device type (0x1000) or the device identity (0x1018) test.

Download Configuration

- ☒ Force communication parameters
- ☐ Force application parameters

Download configuration option is used to force the parameters to be sent even if they are equal to the default value.

Start

- ☒ Start node

Start option is used to configure whether the device shall be started automatically by the master or not.

PDO transfer

Not necessary

Must be set!

Function: Default



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Finally, to activate the PLC communication profile, the parameter 301B:8h must be written to 1 and 3041:Bh must be written to 15. This is done automatically at the end of section *main*. This program part is executed automatically with the rising edge of system bit %IW0.0.2.35.1. This bit indicates that the drive is in operational, that means the configuration is finished.

Additionally it can be executed from the operator screen.

The table below presents the word %IW0.0.2.35:

Standard symbol	Type	Access	Description	Address
SLAVE_ACTIV_1	BOOL	R	Slave operational on the bus: device 1	%IW0.0.2.35.0
SLAVE_ACTIV_2	BOOL	R	Slave operational on the bus: device 2	%IW0.0.2.35.1
SLAVE_ACTIV_3	BOOL	R	Slave operational on the bus: device 3	%IW0.0.2.35.2
SLAVE_ACTIV_4	BOOL	R	Slave operational on the bus: device 4	%IW0.0.2.35.3



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

```
(*Initialization of PLCopen profile*)
Hmi_re_Start_init:= re(Hmi_Start_Init);

SLAVE_ACTIVE_2 := %iw0.0.2.35.1;
SLAVE_ACTIVE_2_RE := re(SLAVE_ACTIVE_2);

Case uistate of
00:   if Hmi_re_Start_init or SLAVE_ACTIVE_2_RE then
        uistate:=10;
        Init_done:=false;
    end_if;

10:   Write_var_data_1_2[0]:=1;
        Write_var_exchange_parameter_1_4[3] := 2 ;(*Length of data*)
        WRITE_VAR (ADR := Adress,
                    OBJ := 'SDO',
                    NUM := 16#0008301B,
                    NB := 0,
                    EMIS := Write_var_data_1_2,
                    GEST := Write_var_exchange_parameter_1_4);
        if Write_var_exchange_parameter_1_4[0].0 then
            uistate:=20;
        end_if;

20:   if not Write_var_exchange_parameter_1_4[0].0 then
            uistate:=30;
        end_if;

30:   Write_var_data_1_2[0]:=15;
        Write_var_exchange_parameter_1_4[3] := 2 ;(*Length of data*)
        WRITE_VAR (ADR := Adress,
                    OBJ := 'SDO',
                    NUM := 16#000B3041,
                    NB := 0,
                    EMIS := Write_var_data_1_2,
                    GEST := Write_var_exchange_parameter_1_4);
        if Write_var_exchange_parameter_1_4[0].0 then
            uistate:=40;
        end_if;

40:   if not Write_var_exchange_parameter_1_4[0].0 then
            uistate:=50;
        end_if;

50:   Init_done:=true;
        uistate:=0;

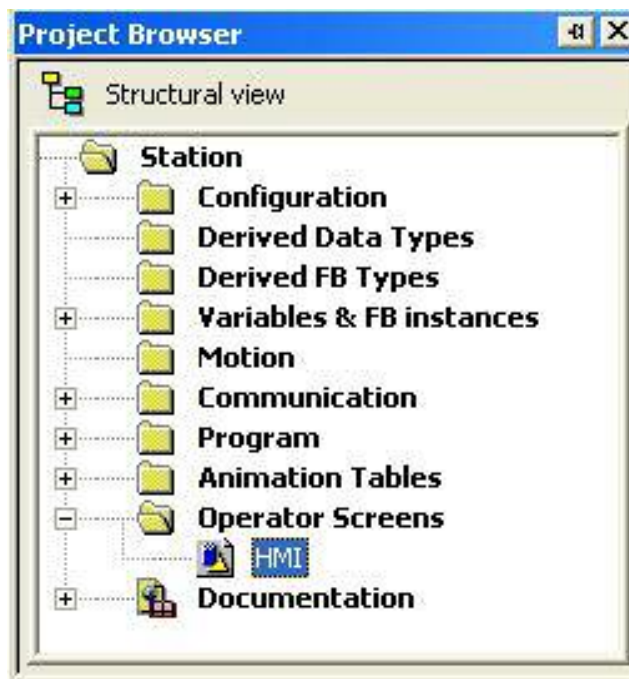
end_case;
```

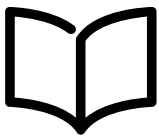


Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Operator screen

This program example contains an operator screen that allows to control the drive.





Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

M340-Unity - LXM32 via CANopen PLCopen profile

Status signals

X_END

X_ADD1

REF_OK

Actual position	Actual operation mode	Actual drive status
101206977	16#0001	6

Motion Status

TAR0

PWIN

MOTZ

Mode and Function Status

Drive input

DI3

DI2

DI1

Drive control

Fault reset

Quickstop

Enable

Disable

Homing

Homing

Type

0

Point to point

Start abs.

Start rel.

Set speed

0

Set position

0

Profile Velocity

Move_Vel

Set speed

0

Profile Torque

Start torque

Set torque

0

Manual

<<

<

>

>>

Parameter control via SDO

	Index	Subindex	Data	Data length in byte
Read SDO				
Write SDO				

Init communication

Start_Init

Init_Done

When Unity is connected with the PLC the first section *Status signals* shows the actual position, the actual operation mode and the actual drive status of the drive, also some important status bits from the status word “Motion status” “mf status” and the drive input.

In section *drive control* the drive could be switched on/off using the buttons *Enable* and *Disable*. The button *Fault reset* sends a fault reset. *Quickstop* is to set a quickstop.

To start a homing movement the type of homing, e.g. 17 for homing to LIMN, must be adjusted. A click on the button *Homing* starts the movement.

For a point to point movement *Set speed* and *Set position* must be adjusted, a click on *Start abs* starts an absolute movement, a click on *Start rel* a relative movement.



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

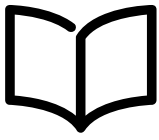
For the Profile velocity mode the *Set speed* must be adjusted, button *Move_Vel* starts the movement. To modify the speed the button must be pushed again.

For the Profile Torque mode the *Set torque* and set torque ramp must be adjusted, button *Start_torque* starts the Torque mode. To modify the current the button must be pushed again.

A manual movement could be done pushing the buttons in its green field.

The next section could be used to read and write data from the drive using *WRITE_VAR* and *READ_VAR* commands. For the parameter reading the Can address of the parameter must be edited with Index and Sub-Index, after pushing the *Read* button the data value is displayed. For parameter writing allboth fields must be edited.

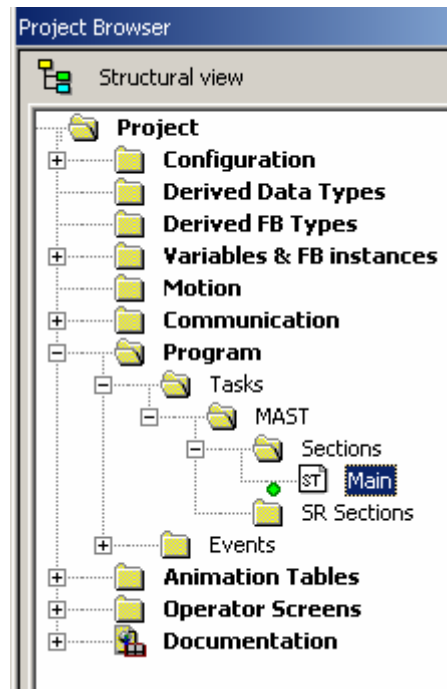
With *Start_init* the initialization of the PLCopen profile could be started manually. *Init_Done* indicates the end it.

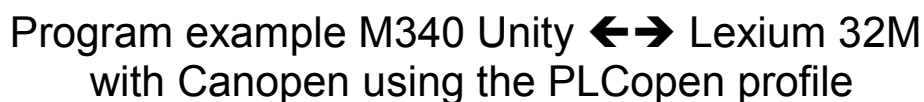


Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

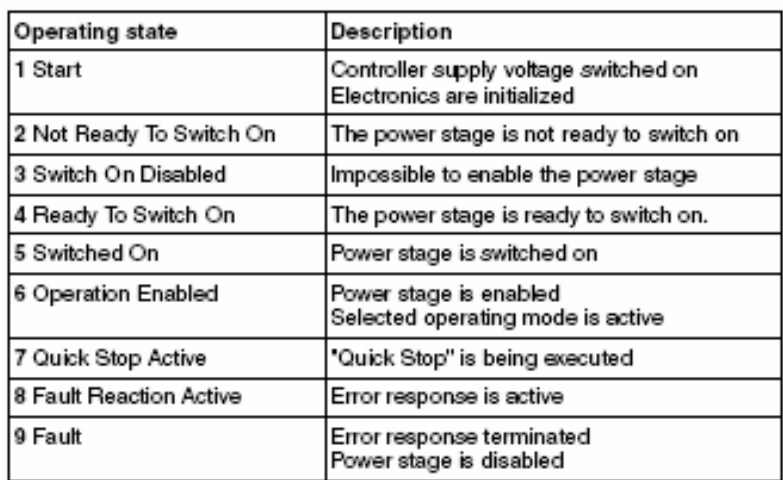
Explanation of the source code

This program example has only one section in the mast task.





After switching on the drive a sequence of operating states is progressed through. If the drive is ok it goes automatically to state 4. When it is enabled and there is no error it changes to state 6.





Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

E.g. to enable the drive the bit 9 of the drive control byte must be true. The bits *Hmi_disable*, *Hmi_enable*, *HMI_fault_reset* and *Hmi_quickstop* are connected with the push buttons *Disable*, *Enable*, *Quickstop* and *Fault reset* from the operator screen.

```
(*Disable the drive-----*)
IF Hmi_disable THEN
    driveCtrl.8 := TRUE ;
    driveCtrl.9 := FALSE ;
ELSE
    driveCtrl.8 := FALSE ;
END_IF;

(*Enable the drive-----*)
IF Hmi_enable AND NOT Hmi_disable THEN
    driveCtrl.9 := TRUE ;
ELSE
    driveCtrl.9 := FALSE ;
END_IF;

(*Quickstop-----*)
IF Hmi_quickstop THEN
    driveCtrl.10 := TRUE ;
ELSE
    driveCtrl.10 := FALSE ;
END_IF;

(*Fault reset-----*)
IF Hmi_fault_reset THEN
    driveCtrl.11 := TRUE ;
ELSE
    driveCtrl.11 := FALSE ;
END_IF;
```

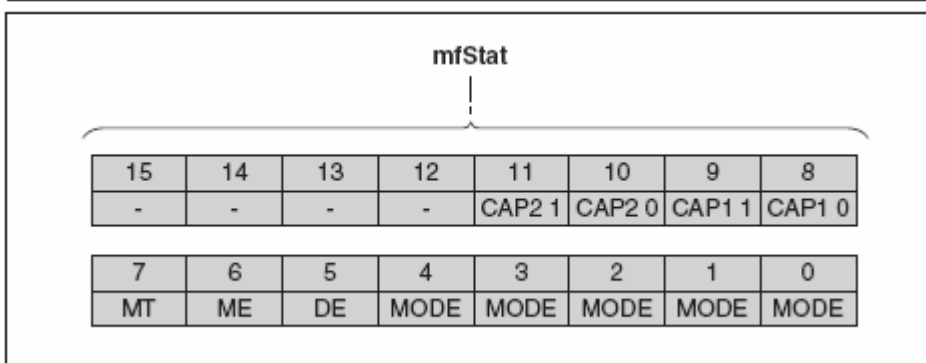
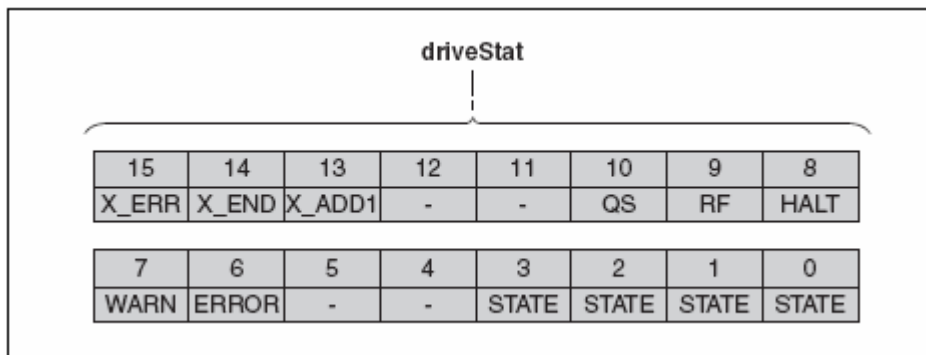
The result is stored in the variable *driveCtrl*. The reason is that drive control is only the high byte of %MW54 (*RPDO1_dmCtrl*), the low byte contains the mode control. Therefore the drive control is stored in *driveCtrl* and the mode control in *modeCtrl*. Some lines below both memory words are connected:

```
(*Copy DriveCtrl and ModeCtrl to output-----*)
RPDO1_dmCtrl := driveCtrl or modeCtrl;
```




Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

The drive status %MW21 (*TPDO1_driveStat*), mf status %MW22 (*TPDO1_mfStat*) and the motion status %MW23 (*TPDO1_motionSt*) contain the information of the actual state of the drive.



Bit	Meaning
0 ... 5	Reserved
6	MOTZ: Motor at a standstill
7	MOTP: Motor movement in positive direction
8	MOTN: Motor movement in negative direction
9 ... 10	Reserved
11	TAR0: Profile generator at standstill, reference velocity = 0
12	DEC: Profile generator decelerates
13	ACC: Profile generator accelerates
14	CNST: Profile generator moves at constant velocity
15	Reserved



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

This variables are directly displayed on the HMI, e.g the field *X_ERR* is directly connected with *TPDO1_driveStat.15*

The bit 0 ..3 (state) in the drive status contains the value of the actual status, the bits 0 .. 5 (mode) in the *mfstat* contains the number of the actual operation mode. For the HMI only the data in those fields are needed.

```
(*Display of actual operation mode-----*)  
HMI_Actual_operation_mode := TPDO1_mfStat AND 2#00000000000011111 ;  
HMI_Actual_drive_status := TPDO1_driveStat AND 16#000F ;
```



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Operation modes:

Bit	Name	Description
0 ... 4	MODE	Operating mode Value 01 _h : Profile Position Value 03 _h : Profile Velocity Value 04 _h : Profile Torque Value 06 _h : Homing Value 1F _h : Jog Value 1E _h : Electronic Gear
5 ... 6	ACTION	Operating mode-dependent
7	MT	Handshake via Mode Toggle

The operating modes can be changed during operation. For this purpose, the current process must be completed or explicitly canceled. The motor must be at a standstill.

The master must enter the following values to activate an operating mode or to change reference values:

- Reference values, depending on desired operating mode
- Operating mode in "dmControl", bits 0 ... 4 (MODE).
- Action for this operating mode in bit 5 and bit 6 (ACTION)
- Toggle bit 7 (MT)

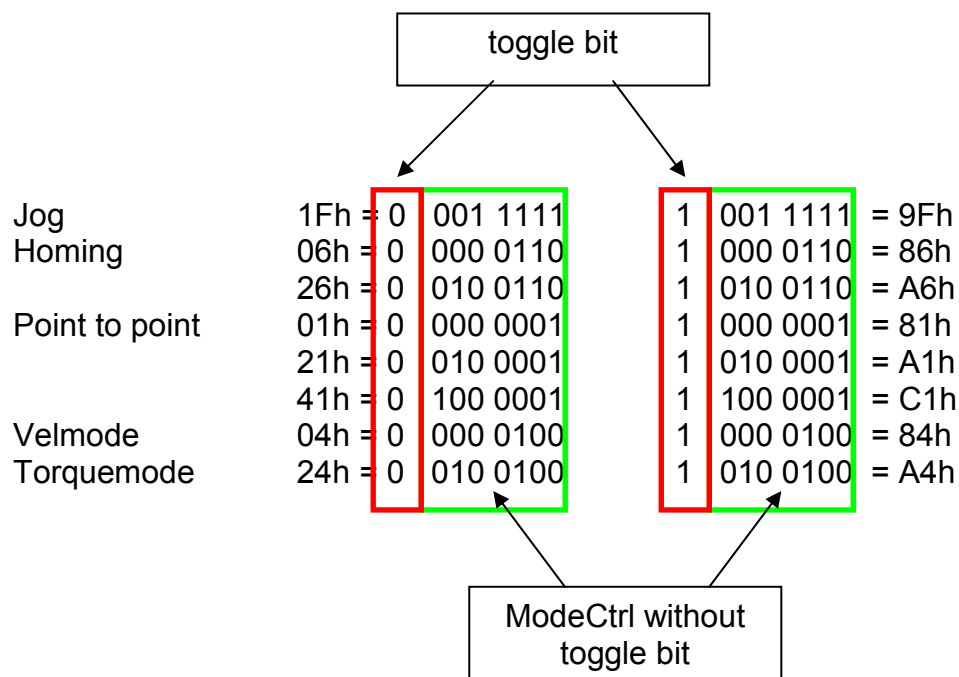
The following chapters describe the possible operating modes, functions and the corresponding reference values.



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

The left column in the table below shows the operation mode, the next column the entry in the mode control without the first bit, the toggle bit. This bit appears twice.

Once in the *dmcontrol* and once in the *mfstat*, where it is mirrored. To start an operation mode its state has to change. When the toggle bit is false in the mode status it must be set to true in the mode control. After receiving the command it changes to true in the mode status. To activate the next operation mode it must be set to false. The hexadecimal values are:



Depending on the operation mode the meaning of the fields RefA16 and RefB32 is different. For example in velocity mode RefA32 contains the set speed, in homing it is the homing method.



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Homing

The product manual contains detailed information about the operation mode.

Description In the operating mode Homing, a movement is performed to a defined position. This position is defined as the reference point.

The reference point also defines the zero point. The zero point is the point of reference for absolute movements in the operating mode Profile Position.

A movement can be made using one of 4 methods:

- Reference movement to a limit switch

In the case of a reference movement to a limit switch, a movement to the negative or positive limit switch is performed. After that, a movement is made to the switching point of the limit switch. From the switching point of the limit switch, a movement is made to the index pulse or to a parameterizable distance from the switching point. The position of the index pulse or the parameterizable distance from the switching point is the reference point.

- Reference movement to the reference switch

In the case of a reference movement to the reference switch, a movement to the reference switch is performed. After that, a movement is made to the switching point of the reference switch. From the switching point of the reference switch, a movement is made to the index pulse or to a parameterizable distance from the switching point. The position of the index pulse or the parameterizable distance from the switching point is the reference point.

- Reference movement to the index pulse

In the case of a reference movement to the index pulse, a movement is made directly to the closest index pulse. The position of the index pulse is the reference point.

- Position setting

In the case of position setting, the current motor position is set to a desired position value. The zero point is defined by the position value.

A reference movement must be completed for the new zero point to be valid. If the reference movement is interrupted, it must be started again.



Motors with multiturn encoder deliver a valid zero point after they are switched on.



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
HMmethod	<p>Homing method</p> <p>1: LIMN with index pulse 2: LIMP with index pulse 7: REF+ with index pulse, inv., outside 8: REF+ with index pulse, inv., inside 9: REF+ with index pulse, not inv., inside 10: REF+ with index pulse, not inv., outside 11: REF- with index pulse, inv., outside 12: REF- with index pulse, inv., inside 13: REF- with index pulse, not inv., inside 14: REF- with index pulse, not inv., outside 17: LIMN 18: LIMP 23: REF+, inv., outside 24: REF+, inv., inside 25: REF+, not inv., inside 26: REF+, not inv., outside 27: REF-, inv., outside 28: REF-, inv., inside 29: REF-, not inv., inside 30: REF-, not inv., outside 33: Index pulse neg. direction 34: Index pulse pos. direction 35: Position setting</p> <p>Abbreviations: REF+: Search movement in pos. direction REF-: Search movement in pos. direction inv.: Invert direction in switch not inv.: Direction not inverted in switch outside: Index pulse / distance outside switch inside: Index pulse / distance inside switch</p> <p>Changed settings become active immediately.</p>	- 1 18 35	INT8 INT16 INT16 INT16 R/W - -	CANopen 6098:0 _h Modbus 6936 Profibus 6936 CIP 127.1.12



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Starting the operating mode The operating mode is started in the process data channel with the transmit data (master to slave) and "Mode Toggle". The value in the word "dmControl" is without set "Mode Toggle" bit.

Method	dmControl MODE+ACTION	Reference value RefA32	Reference value RefB32
Position setting	06 _h	-	As Hmp_setP
Reference movement	26 _h	As HMmethod	-

Status information The word "driveStat" provides information on the operating mode.

Bit	Name	Meaning
13	X_INFO	Reserved
14	X_END	0: Operating mode active 1: Operating mode terminated
15	X_ERR	0: No error 1: Error

Terminating the operating mode The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Homing successful
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

The source code is very short. *Hmi_homing* is the bit from the button in the HMI. *Hmi_re_homing* is a help flag to detect the rising edge of the button. The type data from the HMI *Homing_type* goes directly to the RefA16 in the PDO data (%MW55). Depending on the toggle bit in *mfstat* (MW22:X7) the toggle bit in *modeCtrl* is set or reset.

```
(*Homing mode-----*)
Hmi_re_homing := Hmi_homing ;

IF RE(Hmi_re_homing) THEN
    RPDO1_refA16 := HMI_Homing_type ;
    IF TPDO1_mfStat.7 THEN
        modeCtrl := 16#0026 ;
    ELSE
        modeCtrl := 16#00A6 ;
    END_IF;
END_IF;
```



Program example M340 Unit ↔ Lexium 32M with Canopen using the PLCopen profile

Profile velocity

The product manual contains detailed information.

8.3.5 Operating mode Profile Velocity

Description In the operating mode Profile Velocity, a movement is made with a desired target velocity.

8.3.5.1 Parameterization

Overview The illustration below provides an overview of the adjustable parameters.

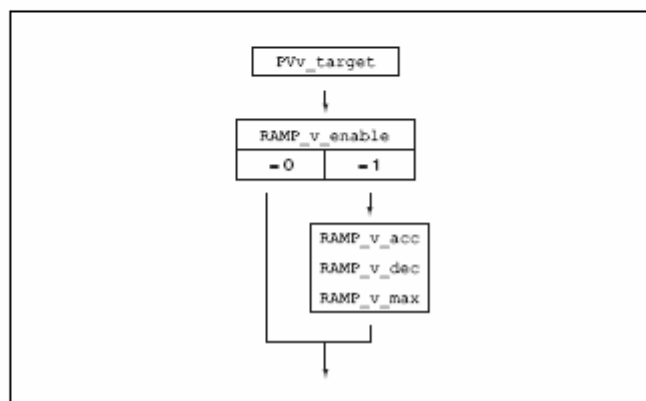


Figure 8.8 Overview of adjustable parameters

Setting the target velocity The target velocity is set by means of the parameter PVv_target.

► Set the target velocity with the parameter PVv_target.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
PVv_target	Target velocity for operating mode Profile Velocity The target velocity is limited to the setting in CTRL_v_max and RAMP_v_max. Changed settings become active immediately.	usr_v - 0 -	INT32 INT32 INT32 R/W - -	CANopen 60FF:0h Modbus 6938 Profibus 6938 CIP 127.1.13

Changing the motion profile for the velocity It is possible to change the settings of the motion profile for the velocity, see chapter 8.4.7.1 "Motion profile for the velocity".



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Starting the operating mode The operating mode is started in the process data channel with the transmit data (master to slave) and "Mode Toggle". The value in the word "dmControl" is without set "Mode Toggle" bit.

dmControl MODE+ACTION	Reference value RefA32	Reference value RefB32
23 _h	As PVv_target	-

Status information The word "driveStat" provides information on the operating mode.

Bit	Name	Meaning
13	X_INFO	0: Target velocity not reached 1: Target velocity reached
14	X_END	0: Operating mode active 1: Operating mode terminated
15	X_ERR	0: No error 1: Error

Terminating the operating mode The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

The velocity mode works in the same way. Now RefA16A contains the set speed.

```
(*Velocity mode-----*)
Hmi_re_velmode := Hmi_velmode ;

IF RE(Hmi_re_velmode) THEN
    RPDO1_refA16 := HMI_Vel_setspeed ;
    IF TPDO1_mfStat.7 THEN
        modeCtrl := 16#0023 ;
    ELSE
        modeCtrl := 16#00A3 ;
    END_IF;
END_IF;
```



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Profile position (point to point)

The product manual contains detailed information.

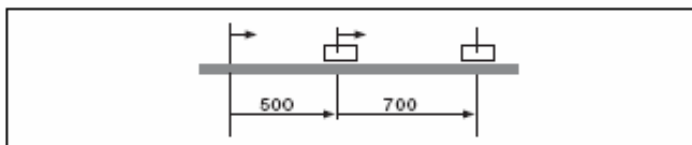
8.3.6 Operating mode Profile Position

Description In the operating mode Profile Position, a movement to a desired target position is performed.

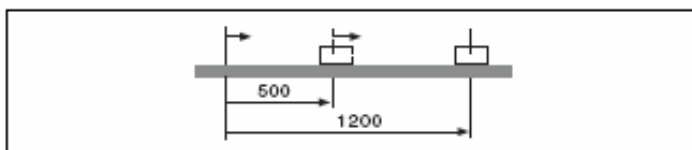
A movement can be made using one of 2 methods:

- Relative movement
- Absolute movement

Relative movement In the case of a relative movement, the movement is relative with reference to the previous target position or the current motor position.



Absolute movement In the case of an absolute movement, the movement is absolute with reference to the zero point.



A zero point must be defined with the operating mode Homing prior to the first absolute movement.



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Starting the operating mode The operating mode is started in the process data channel with the transmit data (master to slave) and "Mode Toggle". The value in the word "dmControl" is without set "Mode Toggle" bit.

Method	dmControl MODE+ACTION	Reference value RefA32	Reference value RefB32
absolute	01 _h	As PPv_target	As PPp_target
Relative with reference to the currently set target position	21 _h	As PPv_target	As PPp_target
Relative with reference to the current motor position	41 _h	As PPv_target	As PPp_target

Status information The word "driveStat" provides information on the operating mode.

Bit	Name	Meaning
13	X_INFO	0: Target position not reached 1: Target position reached
14	X_END	0: Operating mode active 1: Operating mode terminated
15	X_ERR	0: No error 1: Error

Terminating the operating mode The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Target position reached
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

The source code works also in the same way than the other operation modes. Now RefA16 contains the set speed and RefB32 the set position.

```
(*Point to Point mode relative-----*)
Hmi_re_ptpmode_rel := Hmi_ptpmode_rel ;

IF RE(Hmi_re_ptpmode_rel) THEN
    RPDO1_refA16 := HMI_setspeed ;
    RPDO1_refB32 := HMI_setposition ;
    IF TPDO1_mfStat.7 THEN
        modeCtrl := 16#0021 ;
    ELSE
        modeCtrl := 16#00A1 ;
    END_IF;
END_IF;
```



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Jog mode (manual mode)

The product manual contains detailed information.

8.3.2 Operating mode Jog

Description In the operating mode Jog, a movement is made from the current motor position in the desired direction.

A movement can be made using one of 2 methods:

- Continuous movement
- Step movement

In addition, the product features 2 parameterizable velocities.

Continuous movement As long as the signal for the direction ("Jog Positive" or "Jog Negative") is available, a continuous movement is made in the desired direction.

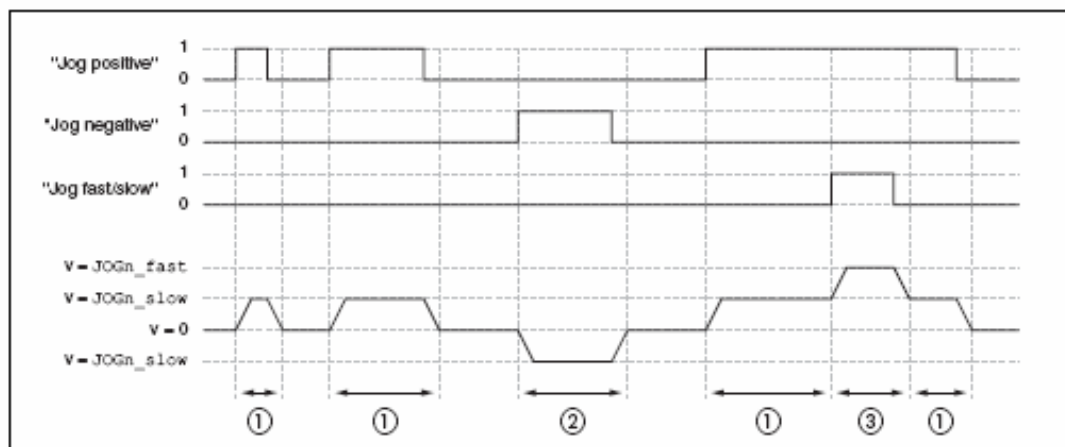


Figure 8.3 Continuous movement

- (1) Slow movement in positive direction
- (2) Slow movement in negative direction
- (3) Fast movement in positive direction

Step movement If the signal for the direction ("Jog Positive" or "Jog Negative") is available, a movement by a parameterizable number of user-defined units is made in the desired direction. After this movement, the motor stops for a defined time. Then a continuous movement is made in the desired direction.



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Availability The operating mode is only available with the drive profile "Drive Profile Lexium".

Starting the operating mode The operating mode is started in the process data channel with the transmit data (master to slave) and "Mode Toggle". The value in the word "dmControl" is without set "Mode Toggle" bit.

dmControl MODE+ACTION	Reference value RefA32	Reference value RefB32
1F _h	As JOGactivate	-





Status information The word "driveStat" provides information on the operating mode.

Bit	Name	Meaning
13	X_INFO	Reserved
14	X_END	0: Operating mode active 1: Operating mode terminated
15	X_ERR	0: No error 1: Error





Terminating the operating mode The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Value 0 in RefA32
- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

The bits

 Hmi_manual_neg_fast	EBOOL
 Hmi_manual_neg_slow	EBOOL
 Hmi_manual_pos_fast	EBOOL
 Hmi_manual_pos_slow	EBOOL

are the signals from the push buttons, the bits

 Hmi_re_manual_neg_fast	EBOOL
 Hmi_re_manual_neg_slow	EBOOL
 Hmi_re_manual_pos_fast	EBOOL
 Hmi_re_manual_pos_slow	EBOOL

are help flags to detect the rising and falling edges. For the manual mode RefA16 has the meaning of parameter *JOGactivate* (CANopen 301B:9h). The resulting bit combinations are:

no movement	000	=	0
move right slow	001	=	1
move left slow	010	=	2
move right fast	101	=	5
move left fast	110	=	6



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

A rising edge of a push button starts a movement. The falling edge sets RefA16 to zero, the movement stops.

```
(*Manual mode-----*)
Hmi_re_manual_pos_slow := Hmi_manual_pos_slow ;
Hmi_re_manual_pos_fast := Hmi_manual_pos_fast ;
Hmi_re_manual_neg_slow := Hmi_manual_neg_slow ;
Hmi_re_manual_neg_fast := Hmi_manual_neg_fast ;

IF RE(Hmi_re_manual_pos_slow) THEN RPDO1_refA16 := 1 ;END_IF;
IF FE(Hmi_re_manual_pos_slow) THEN RPDO1_refA16 := 0 ;END_IF;
IF RE(Hmi_re_manual_pos_fast) THEN RPDO1_refA16 := 5 ;END_IF;
IF FE(Hmi_re_manual_pos_fast) THEN RPDO1_refA16 := 0 ;END_IF;
IF RE(Hmi_re_manual_neg_slow) THEN RPDO1_refA16 := 2 ;END_IF;
IF FE(Hmi_re_manual_neg_slow) THEN RPDO1_refA16 := 0 ;END_IF;
IF RE(Hmi_re_manual_neg_fast) THEN RPDO1_refA16 := 6 ;END_IF;
IF FE(Hmi_re_manual_neg_fast) THEN RPDO1_refA16 := 0 ;END_IF;

IF RE(Hmi_re_manual_pos_slow) OR FE(Hmi_re_manual_pos_slow) OR
   RE(Hmi_re_manual_pos_fast) OR FE(Hmi_re_manual_pos_fast) OR
   RE(Hmi_re_manual_neg_slow) OR FE(Hmi_re_manual_neg_slow) OR
   RE(Hmi_re_manual_neg_fast) OR FE(Hmi_re_manual_neg_fast) THEN

   IF TPDO1_mfStat.7 THEN
       modeCtrl := 16#001F ;
   ELSE
       modeCtrl := 16#009F ;
   END_IF;
END_IF;
```



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Profile torque

Description In the operating mode Profile Torque, a movement is made with a desired target torque.

8.3.4.1 Parameterization

Overview The illustration below provides an overview of the adjustable parameters.

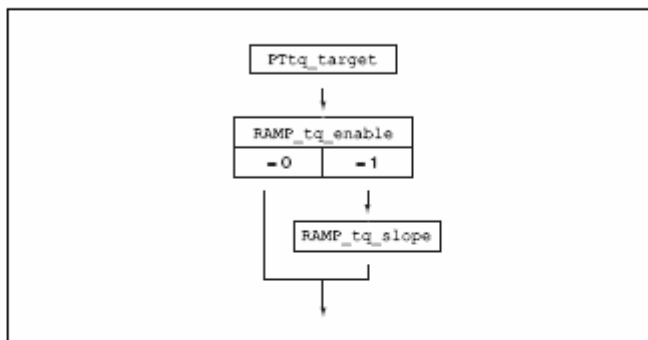


Figure 8.7 Overview of adjustable parameters

Setting the target torque The target torque is set by means of the parameter PTtq_target.

- Set the desired target torque with the parameter PTtq_target.

Parameter name HMI menu HMI name	Description	Unit Minimum value Factory setting Maximum value	Data type R/W Persistent Expert	Parameter address via fieldbus
PTtq_target	Target torque for operating mode Profile Torque 100.0 % correspond to the continuous stall torque _M_M_0. In increments of 0.1 %. Changed settings become active immediately.	% -3000.0 0.0 3000.0	INT16 INT16 INT16 INT16 R/W - -	CANopen 6071:0 _h Modbus 6944 Profibus 6944 CIP 127.1.16

Changing the motion profile for the torque It is possible to change the parameterization of the motion profile for the torque.

RAMP_tq_slope	Slope setting of the motion profile for torque 100.0 % correspond to the continuous stall torque _M_M_0. In increments of 0.1 %/s. Changed settings become active immediately.	%/s 0.1 10000.0 3000000.0	UINT32 UINT32 UINT32 UINT32 R/W per. -	CANopen 6087:0 _h Modbus 1620 Profibus 1620 CIP 106.1.42
---------------	---	------------------------------------	--	---



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Starting the operating mode The operating mode is started in the process data channel with the transmit data (master to slave) and "Mode Toggle". The value in the word "dmControl" is without set "Mode Toggle" bit.

dmControl MODE+ACTION	Reference value RefA32	Reference value RefB32
24 _h	As PTtq_target	As RAMP_tq_slope

Status information The word "driveStat" provides information on the operating mode.

Bit	Name	Meaning
13	X_INFO	0: Target torque not reached 1: Target torque reached
14	X_END	0: Operating mode active 1: Operating mode terminated
15	X_ERR	0: No error 1: Error

Terminating the operating mode The operating mode is terminated when the motor is at a standstill and one of the following conditions is met:

- Stop caused by "Halt" or "Quick Stop"
- Stop caused by an error

The torque mode works in the same way. Now RefA16 contains the target torque and RefB32 contains the setting of the motion profile for torque.

```
(*Profile torque mode-----*)
Hmi_re_torquemode := Hmi_torquemode ;

IF RE(Hmi_re_torquemode) THEN
    RPDO1_refA16 := HMI_setcurrent ;
    IF TPDO1_mfStat.7 THEN
        modeCtrl := 16#0024;
    ELSE
        modeCtrl := 16#00A4;
    END_IF;
END_IF;
```



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Reading and writing parameter via SDO

The reading and writing is done with READ_VAR and WRITE_VAR.

```
(*Adress conversion*)
Address:=ADDM ('0.0.2.2');
(*Read Parameter via SDO-----*)
Hmi_re_read_sdo := RE(Hmi_read_sdo);

IF Hmi_re_read_sdo THEN
    READ_VAR (ADR := Address,
              OBJ := 'SDO',
              NUM := Read_sdo_subindex_index,
              NB  := 0,
              GEST := Read_var_exchange_parameter_1_4,
              RECP => Read_var_data_1_2);

END_IF;

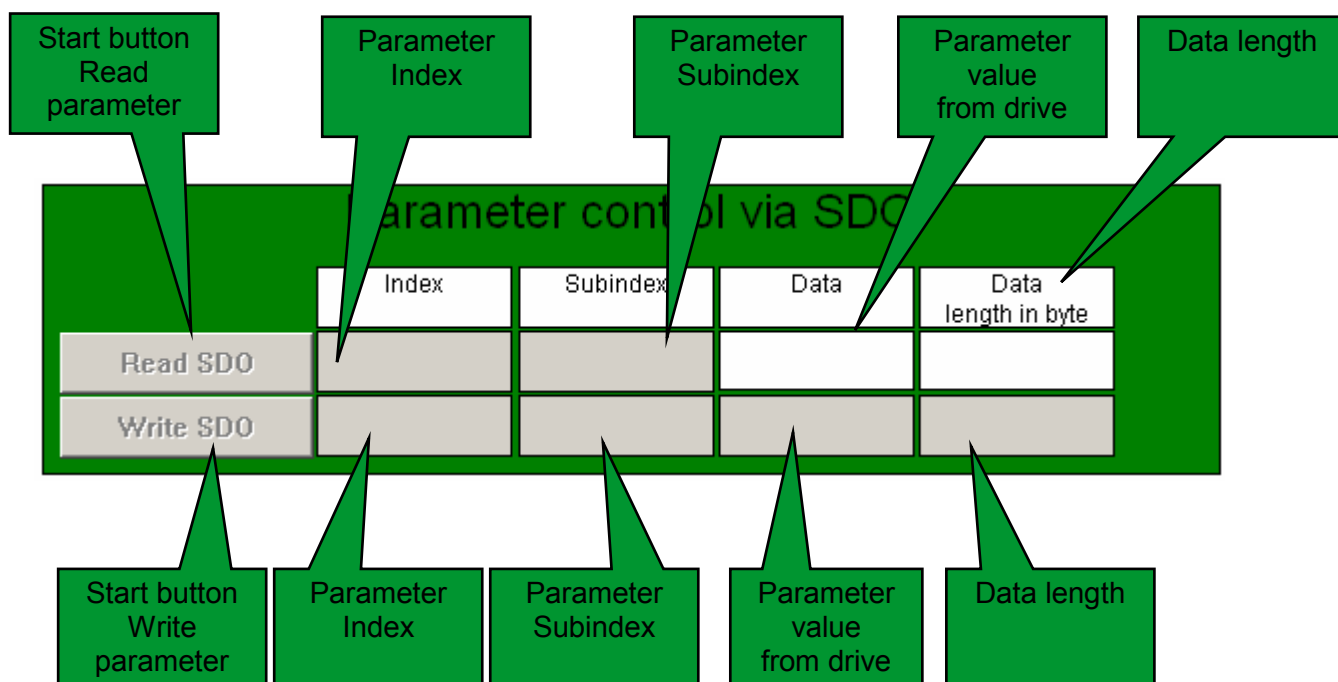
(*Write Parameter via SDO-----*)
Hmi_re_write_sdo := RE(Hmi_write_sdo);

IF Hmi_re_write_sdo THEN
    Write_var_exchange_parameter_1_4[3] := Write_length ;(*Length of data*)
    WRITE_VAR (ADR := Address,
              OBJ := 'SDO',
              NUM := Write_sdo_subindex_index,
              NB  := 0,
              EMIS := Write_var_data_1_2,
              GEST := Write_var_exchange_parameter_1_4);
END_IF;
```



Program example M340 Unity ↔ Lexium 32M with Canopen using the PLCopen profile

Operator screen:



Summary

This program example shows an easy way how to start the operation modes homing, profile velocity, profile torque, profile position and jog mode with the PLCopen Interface. It shows also how to read and write parameters out of the application.