LinMot®

EtherCAT®

Documentation of the EtherCAT Interface of the following Controller:

- **E1250-EC-UC**
- **E1450-EC-QN**

EtherCAT Interface 4.2

User Manual

Document version 4.2 / Whp, July 2011

1 System overview

EtherCAT is the open real-time Ethernet network originally developed by Beckhoff. The LinMot act as Slave in this Network and is implemented with the standard ASIC ET1100 from Beckhoff.

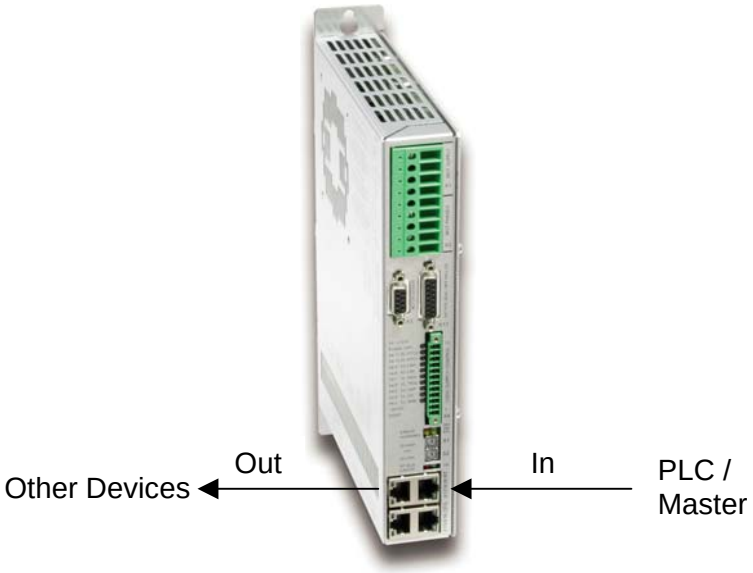
For further information on the EtherCAT fieldbus please visit:
http://www.ethercat.org/

1.1 References

All User Manuals are distributed with the LinMot-Talk software the newest version could be downloaded from the LinMot Homepage in the Download section.

Ref	Title	Source
1	User Manual Motion Control SW	www.linmot.com
2	LinMot Servo Controller Configuration over Fieldbus Interfaces SG5	www.linmot.com

1.2 Connecting In and Out



In the EtherCAT the Cabling is directed, so In and Out is different! The right realtime RJ45 (X17) Connector is the input and the left realtime RJ45 (X18) connector is the output.

2 Setup in the PLC

In the following the steps the integration of a LinMot EtherCAT servo controller in the PLC is described. In the example a Beckhoff master PLC is used.

2.1 Copy Device Description File

The LinMot servo controller is described with *.xml device description file distributed with the LinMot-Talk software.

Copy this file to PLC so it can access it.

Example Source path of EtherCAT Device description file:

C:\Programme\LinMot\LinTalk1100 3.11 Build 20091204\Firmware\Interfaces\EtherCAT\XML\LinMot_ECAT_Servos_V1_4.xml

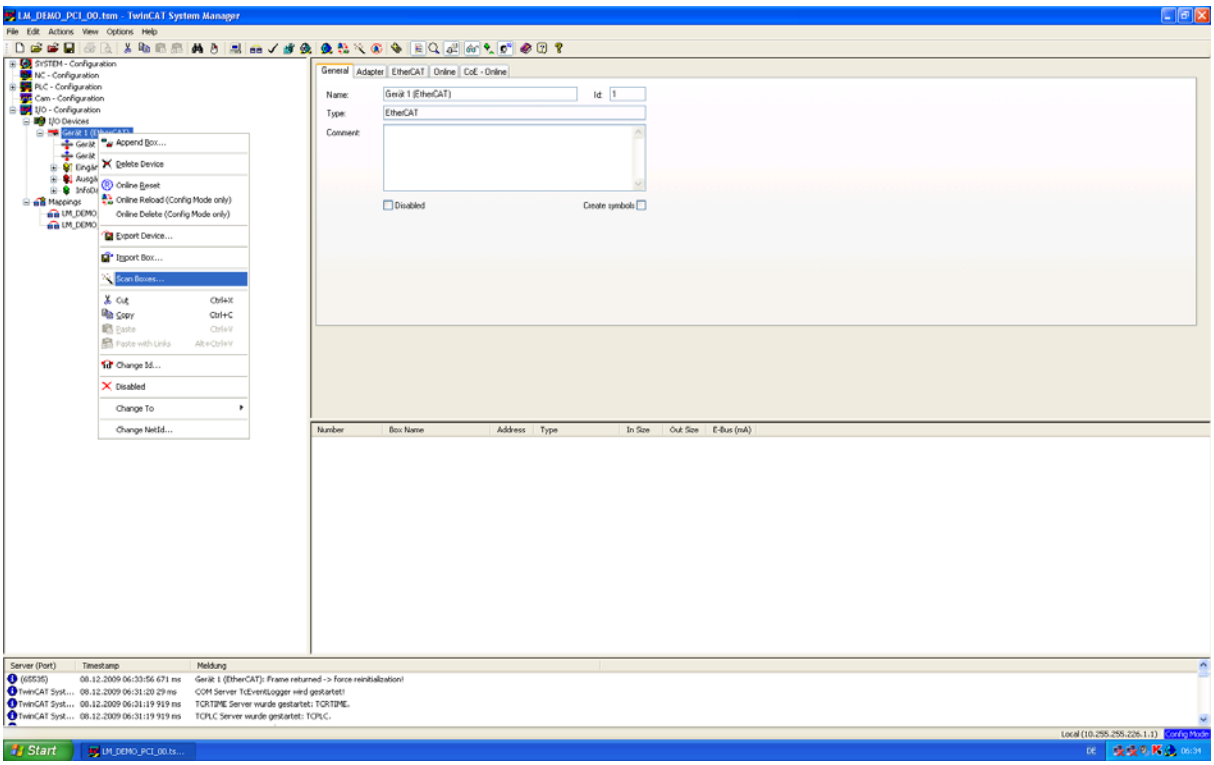
Example Destination path of EtherCAT Device description file:

C:\TwinCAT\Io\EtherCAT\LinMot_ECAT_Servos_V1_4.xml

If this is done the PLC should recognize all LinMot Servo controller on the EtherCAT fieldbus automatically.

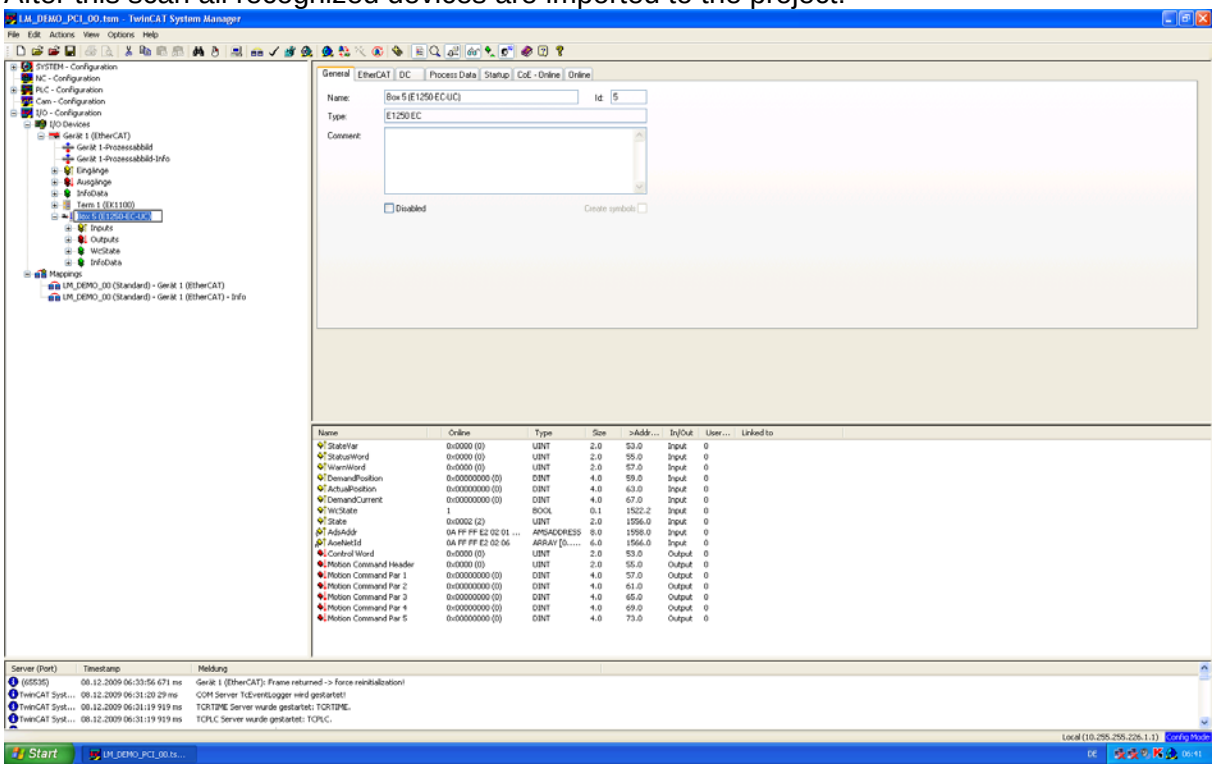
2.2 Scan the EtherCAT slave devices

Connect the EtherCAT LinMot Servo controller to the EtherCAT-Master and power on the Signal Supply. Then Scan for the connected devices in the System manager:



Scan for EtherCAT slave devices

After this scan all recognized devices are imported to the project:



In this example the scan found an EK1100 and the LinMot Servo controller E1250-EC-UC and are imported to the project.

3 Process Data Object (PDO) Configuration

The cyclic Process data is configured in the Master, and transmitted to the slave during startup. The default Mapping is documented in the Tables below input and output correspond to the PLC point of view. For a detailed description of the exchanged data and its meaning refer to [1].

For a detailed description of the PDO data refer to [1] or have a look at the TwinCAT Demo program, which is included with the LinMot-Talk software.

3.1 Input PDO Modules

3.1.1 Default Inputs

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1B00	***18***	***-***	***Variables***	***RECORD***
0x1B62:00	2	0	StateVar	Uint16
0x1D51:00	2	2	StatusWord	Uint16
0x1D8E:00	2	4	WarnWord	Uint16
0x1B8A:00	4	6	DemandPosition	Int32
0x1B8D:00	4	10	ActualPosition	Int32
0x1B93:00	4	14	DemandCurrent	Int32

Default input PDO mapping of 18 Bytes the index is the UPID value of the variable

3.1.2 Configuration Module Input

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1B08	***8***	***-***	***Variables***	***RECORD***
0x1D52:00	2	0	Config Status Word	Uint16
0x1DB0:00	2	2	Config Index/..	Uint16
0x1E40:00	4	4	Conig Value	Word32

This module is the feedback data module of the module 1708 "Configuration Module Output".

3.1.3 Input Demvel

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1B10	***4***	***-***	***Variables***	***RECORD***
0x1B8B:00	4	0	DemandVelocity	Int32

3.1.4 Input ActVel

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1B11	***4***	***-***	***Variables***	***RECORD***
0x1B8E:00	4	0	ActualVelocity	Int32

3.1.5 Input ActVelFilt

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1B12	***4***	***-***	***Variables***	***RECORD***
0x1BAF:00	4	0	ActualVelocityFilt	Int32

3.1.6 Input CAM Counts

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1B18	***4***	***-***	***Variables***	***RECORD***
0x1C50:00	4	0	CAM Counts	Int32

3.1.7 MC-Link A1 Inputs

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1B00	***18***	***-***	***Variables***	***RECORD***
0x1B62:00	2	0	StateVar	Uint16
0x1D51:00	2	2	StatusWord	Uint16
0x1D8E:00	2	4	WarnWord	Uint16
0x1B8A:00	4	6	DemandPosition	Int32
0x1B8D:00	4	10	ActualPosition	Int32
0x1B93:00	4	14	DemandCurrent	Int32
0x1B09:00	2	18	Config Status Word	Uint16
0x1B0A:00	2	20	Config Index/..	Uint16
0x1B0B:00	4	22	Conig Value	Word32

MC-Link A1 PDO mapping of 26 Bytes the index is the UPID value of the variable. With this module a B1150-ML-XX connected on X11 could be monitored.

3.2 Output PDO Modules

3.2.1 Default Outputs

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1700	***24***	***-***	***Variables***	***RECORD***
0x1D52:00	2	0	ControlWord	Uint16
0x1DB0:00	2	2	MotionCommandHeader	Uint16
0x1E40:00	4	4	MotionCommand Par 1	Word32
0x1E41:00	4	8	MotionCommand Par 2	Word32
0x1E42:00	4	12	MotionCommand Par 3	Word32
0x1E43:00	4	16	MotionCommand Par 4	Word32
0x1E44:00	4	20	MotionCommand Par 5	Word32

Default output PDO mapping of 24 Bytes the index is the UPID value of the variable

The default PDO output data contains the control word and the generic Motion Command Interface over which all types of motion commands could be accessed.

3.2.2 Configuration Module Output

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1708	***8***	***-***	***Variables***	***RECORD***
0x1D52:00	2	0	Config Control Word	Uint16
0x1DB0:00	2	2	Config Index/..	Uint16
0x1E40:00	4	4	Conig Value	Word32

With this module the LinMot Servo Controller could be completely configured. So it gives access over parameters curves command table entries and the error log. The detail functionality is described in the Document [2] "Configuration over fieldbus SG5". As response Module "Configuration Module Input" (0x1B08) has to be configured.

3.2.3 CAM Start Enable

Index	Size [Byte]	Byte Offset	Name	Data Type
0x1718	***2***	***-***	***Variables***	***RECORD***
0x1528:00	2	0	CAM Start Enable	Uint16

Bit 0 of the 16 bit data field is mapped to CAM Start Enable Flag (UPID 1528). Write 0 to all other bits.

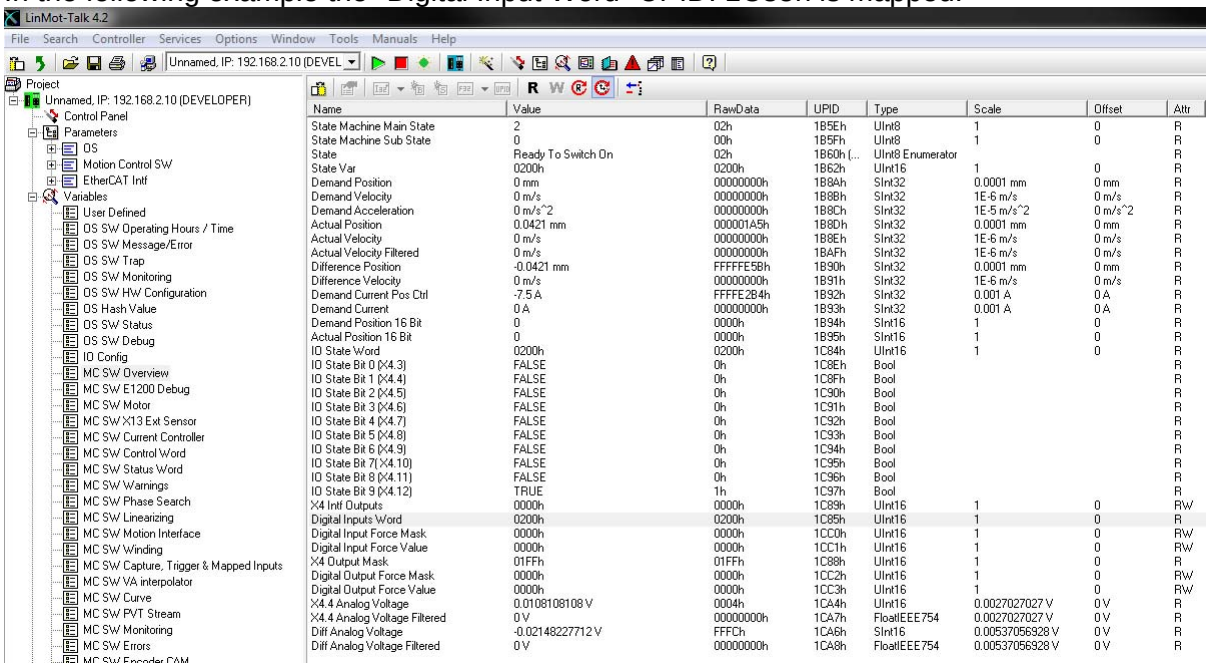
3.2.4 MC-Link A1 Outputs

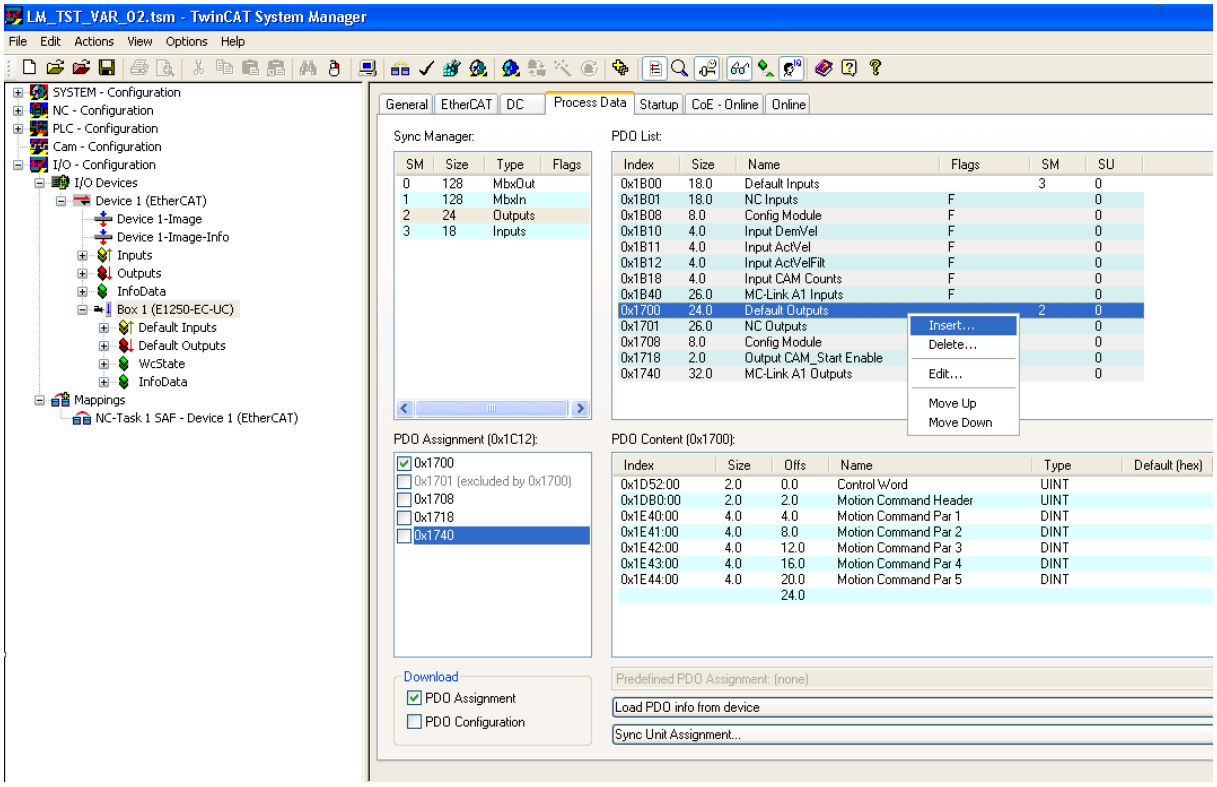
Index	Size [Byte]	Byte Offset	Name	Data Type
0x1740	***24***	***-***	***Variables***	***RECORD***
0x1D52:00	2	0	ControlWord	Uint16
0x1DB0:00	2	2	MotionCommandHeader	Uint16
0x1E40:00	4	4	MotionCommand Par 1	Word32
0x1E41:00	4	8	MotionCommand Par 2	Word32
0x1E42:00	4	12	MotionCommand Par 3	Word32
0x1E43:00	4	16	MotionCommand Par 4	Word32
0x1E44:00	4	20	MotionCommand Par 5	Word32
0x1709:00	2	24	Config Control Word	Uint16
0x170A:00	2	26	Config Index/..	Uint16
0x170B:00	4	28	Conig Value	Word32

MC-Link Outputs PDO mapping of 32 Bytes the index is the UPID value of the variable. With this module a B1150-ML-XX connected on X11 could be controlled.

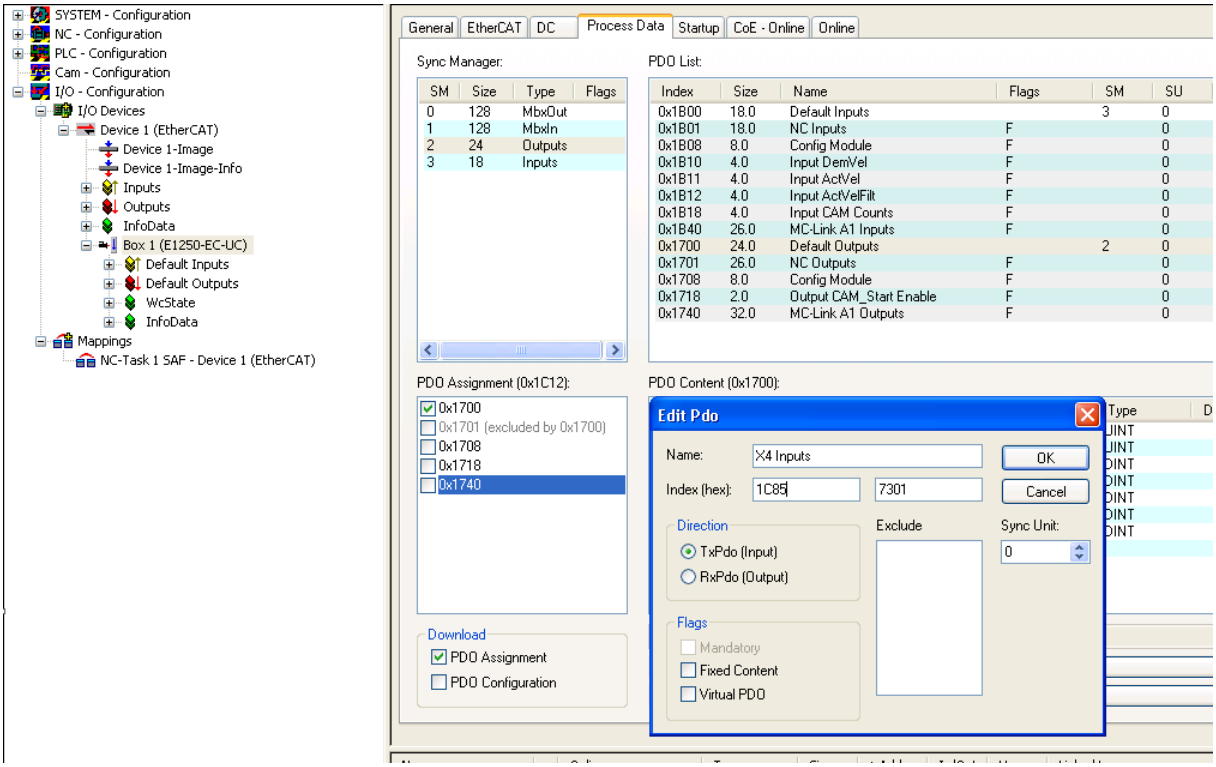
3.3 User defined Input PDO Modules

Within the Beckhoff system manager you could also define your individual IO process data. In the following example the "Digital Input Word" UPID: 1C85h is mapped.

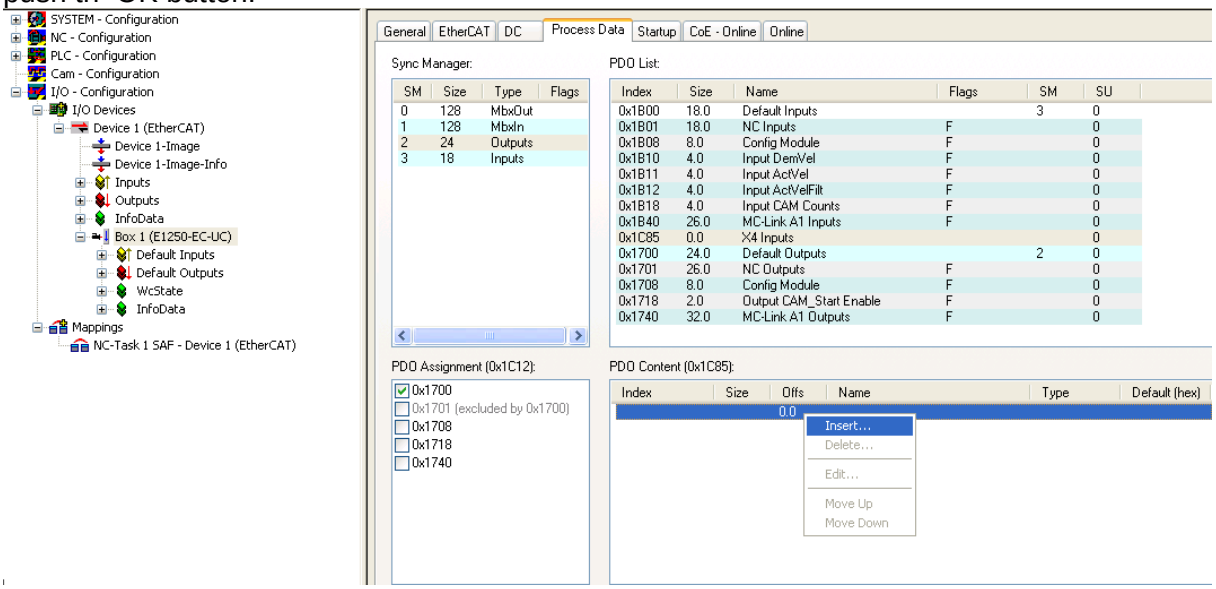




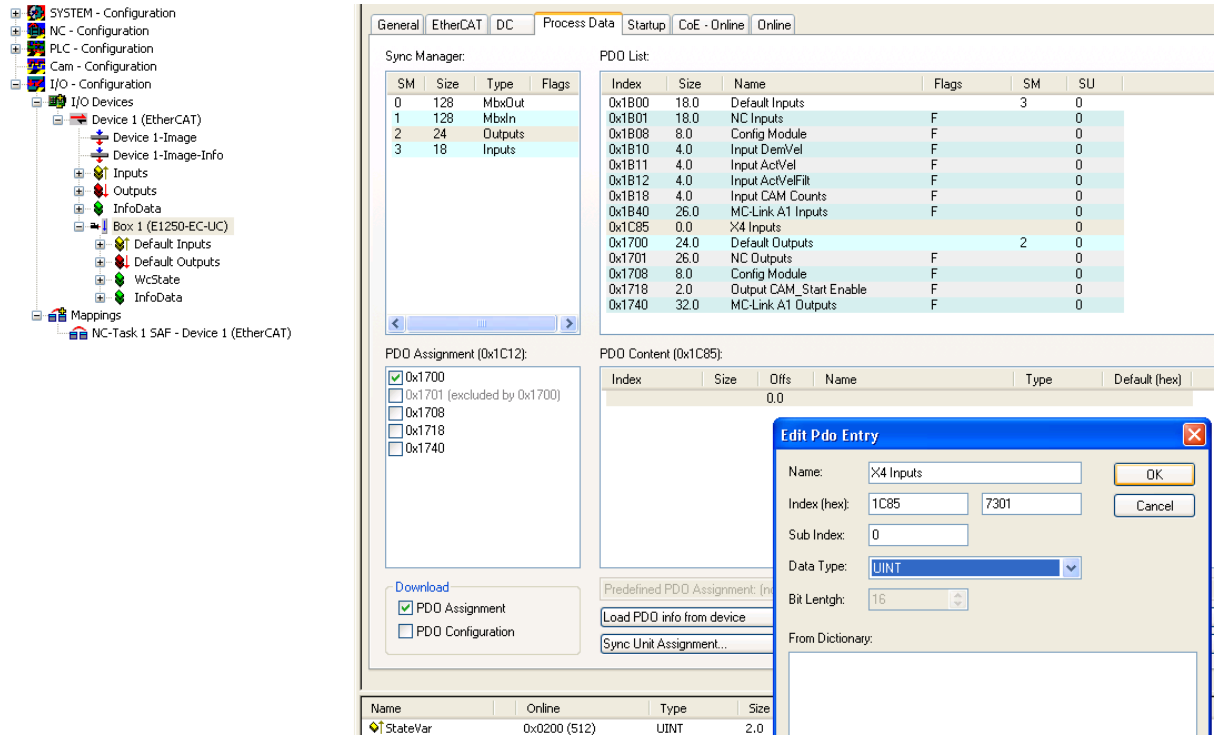
First define a new process data group by inserting into the PDO List.



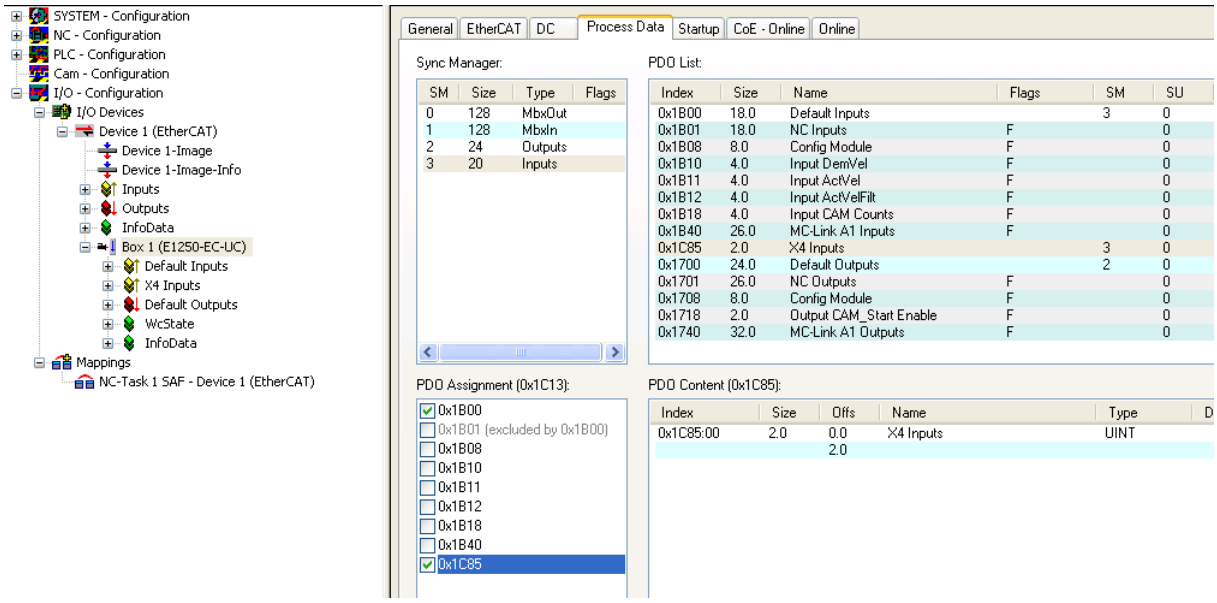
Then give a Name and choose the correct UPID as Index of the data you want to map. Then push th OK button.



Then select the new defined process data and define the data format by inserting PDO Content.



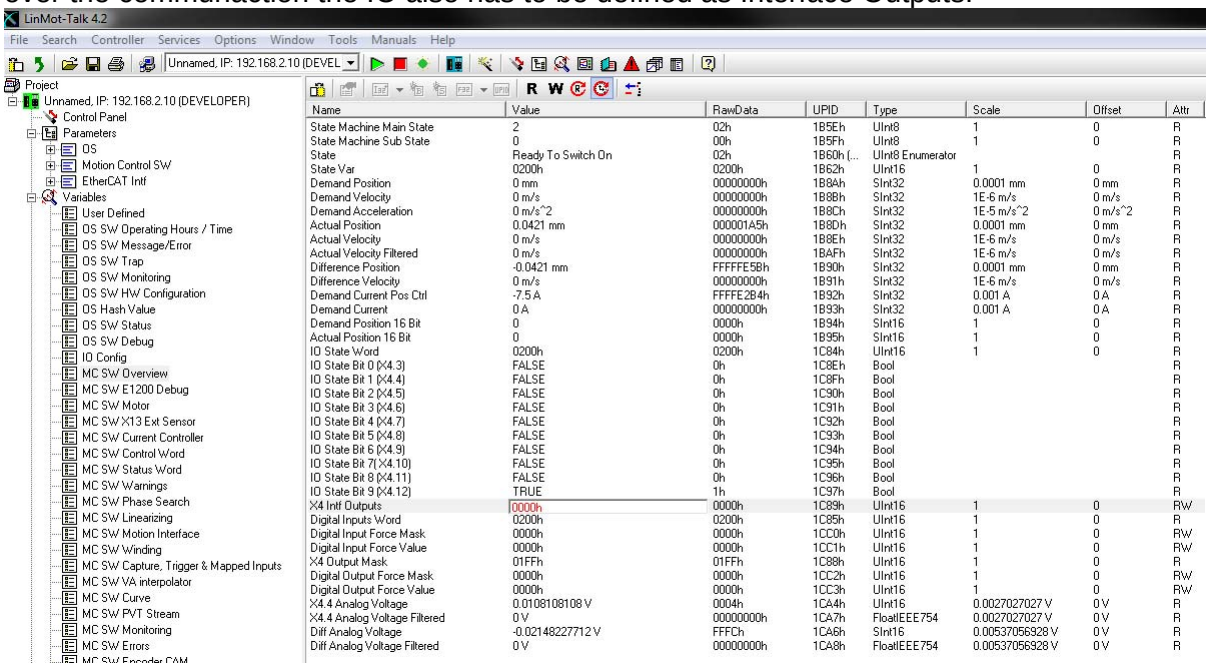
Here you could again define a name and Index (choose the UPID again) and choose the correct data format UINT16 in this case. Then push the OK button again. At the moment only 2 bytes and 4 bytes data size is supported!

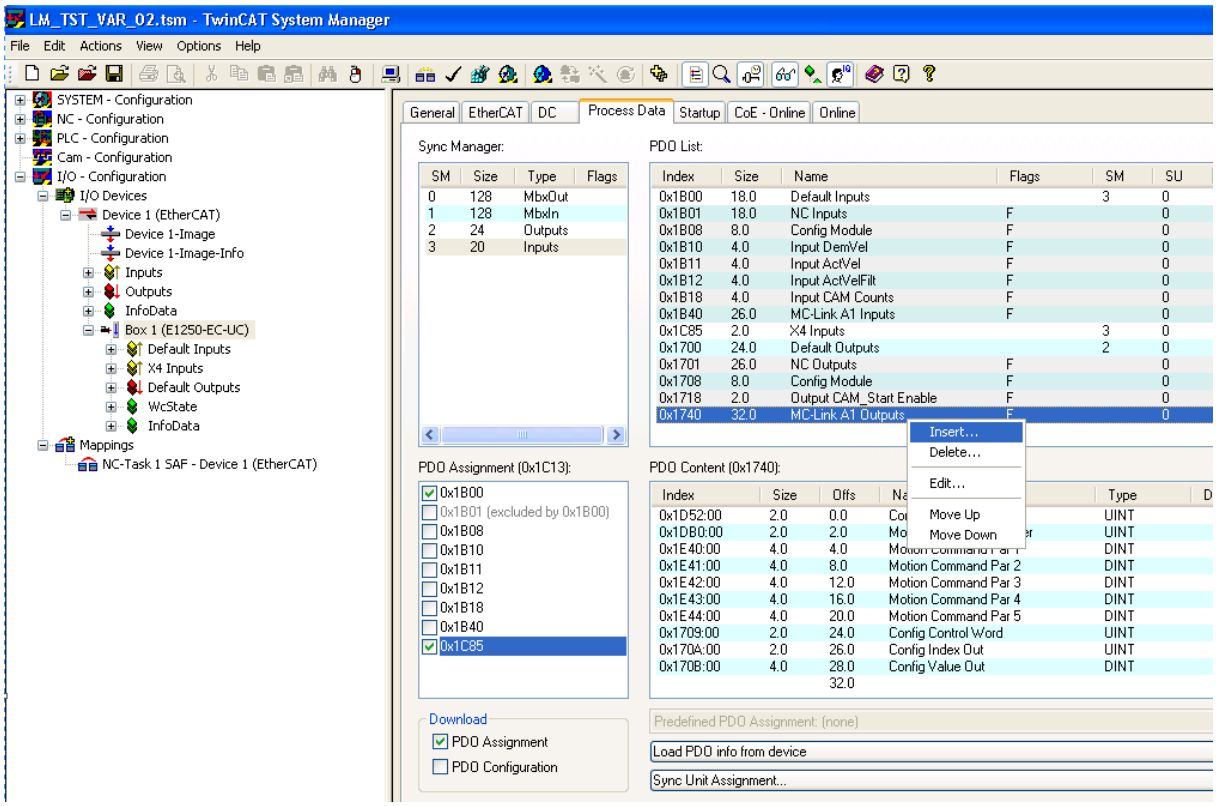


Now you can choose the new generated PDO Data.

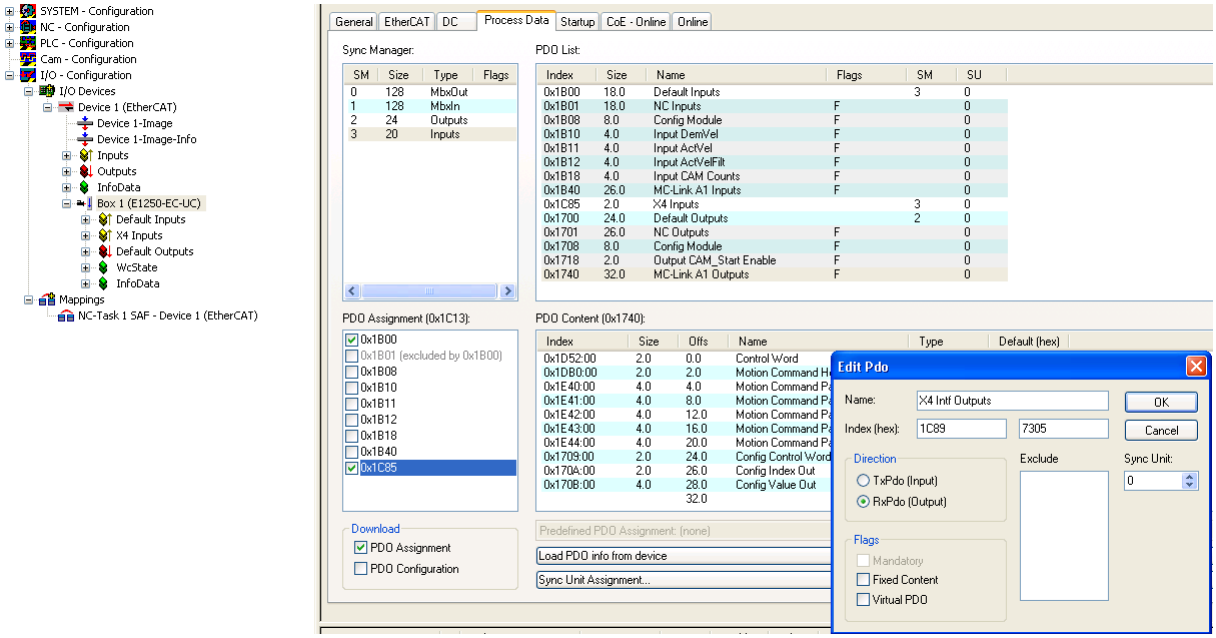
3.4 User defined Output PDO Modules

Within the Beckhoff system manager you could also define your individual IO process data. In the following example the "X4 Intf Outputs" UPID: 1C89h is mapped. To write the outputs over the communaction the IO also has to be defined as Interface Outputs!

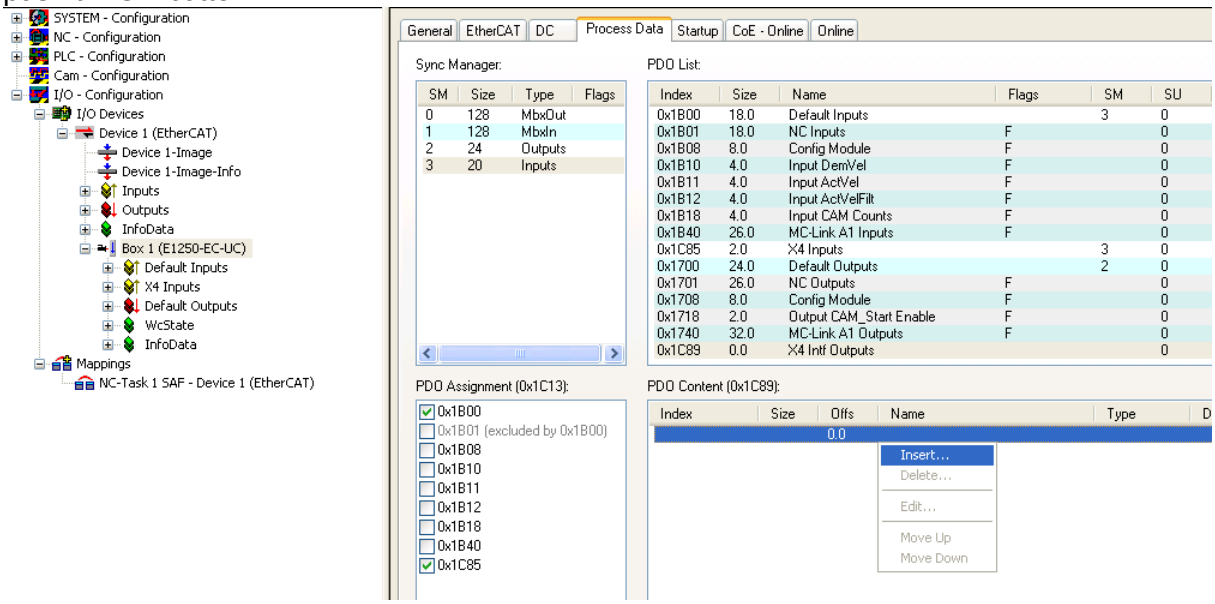




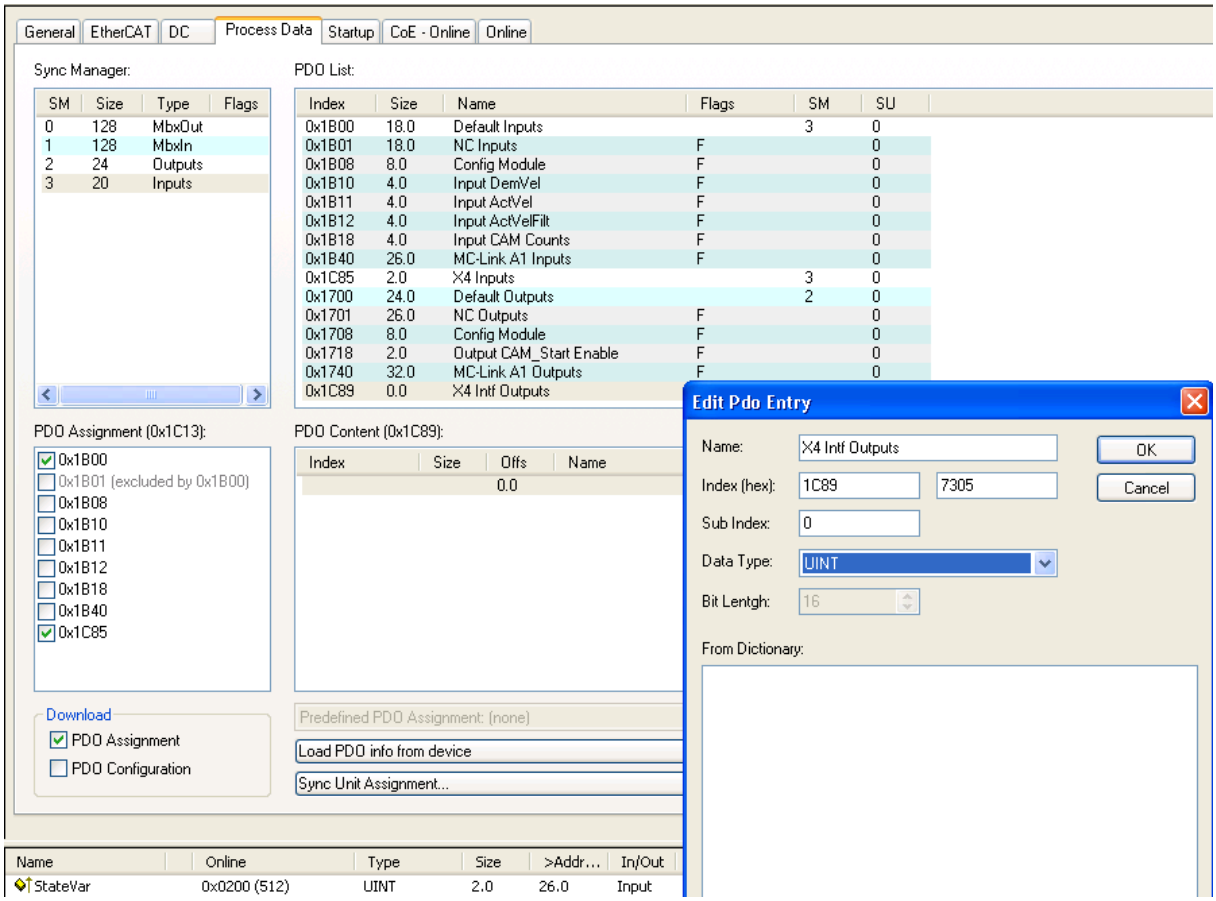
First define a new process data group by inserting into the PDO List.



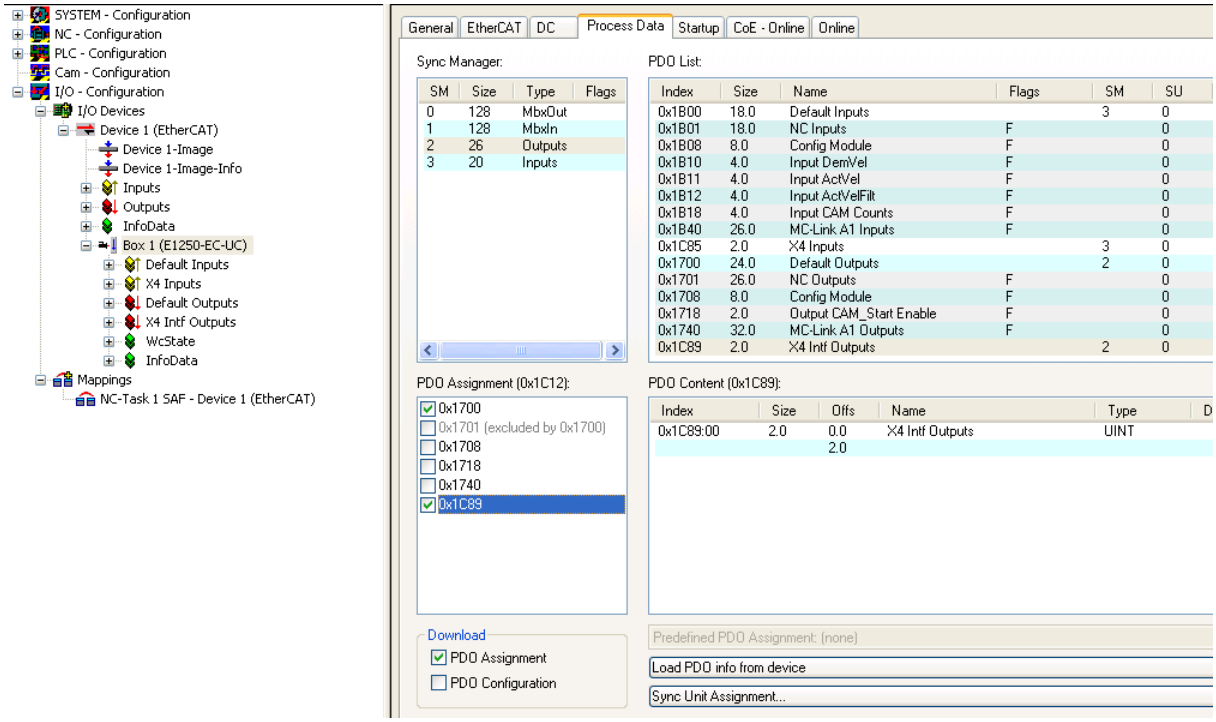
Then give a Name and choose the correct UPID as Index of the data you want to map. Then push th OK button.



Then select the new defined process data and define the data format by inserting PDO Content.



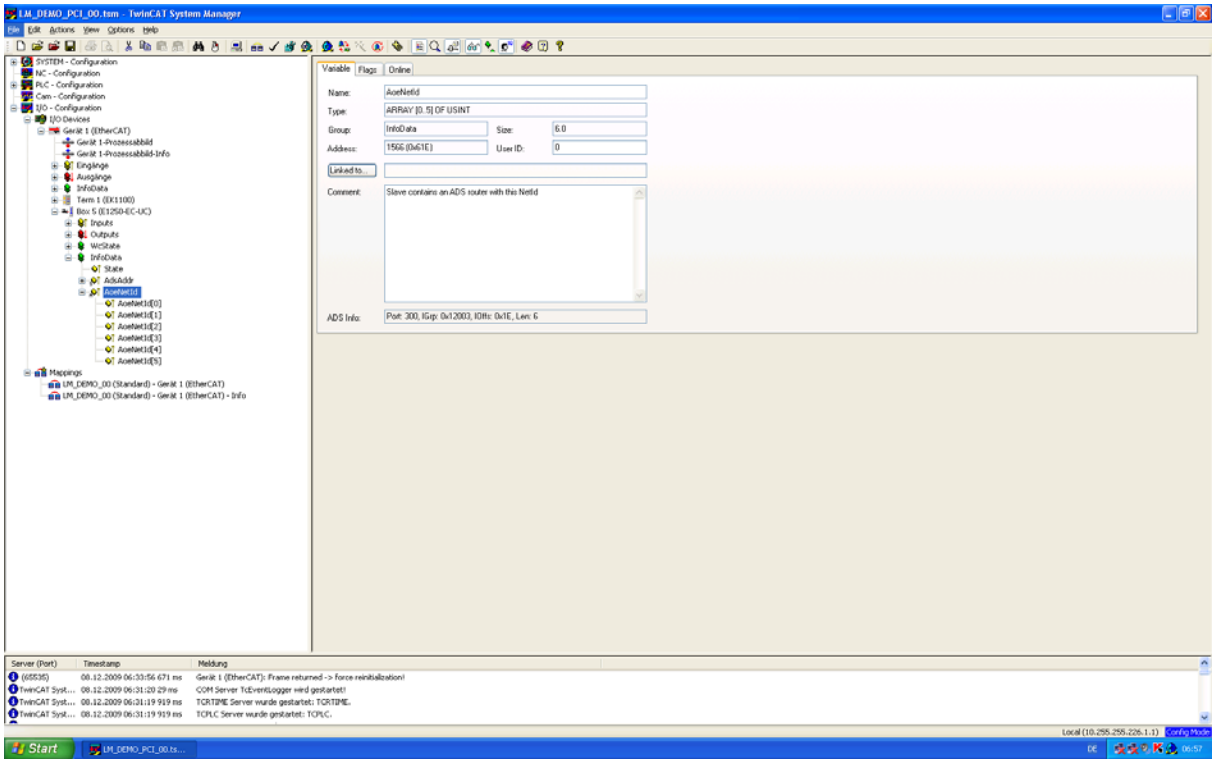
Here you could again define a name and Index (choose the UPID again) and choose the correct data format UINT16 in this case. Then push the OK button again. At the moment only 2 bytes and 4 bytes data size is supported!



Now you can choose the new generated PDO Data.

4 Asynchronous Configuration Protocol

For configuration needs (Parameter, Curves, Command Table and Programm Handling) the Beckhoff ADS-Protocol is used. Within EtherCAT this Protocol is integrated as AoE (ADS over EtherCAT). For this reason the AoeNetId has to be used for ADS over EtherCAT.



AoeNetId with Beckhoff

4.1 ADS Services Overview

For the LinMot servo controllers in the ADS index group, the service is coded. In the index offset normally the UPID for parameters or curve ID for curves is coded.

ADS Service	Index Group	Index Offset	Description
Read	0040h	UPID	Parameter UPID read RAM value
Write	0041h	UPID	Parameter UPID write RAM value
Read	0042h	UPID	Parameter UPID read ROM value
Write	0043h	UPID	Parameter UPID write ROM value
Read	0044h	UPID	Parameter Start read UPID list
Read	0045h	-	Parameter Get next UPID list item
Read	0046h	UPID	Parameter Start read modified UPID list
Read	0047h	-	Parameter Get next modified UPID list item
Write	0048h	Inst	Parameter Default SW instance
Write	0050h	-	Curves delete all in RAM
Write	0051h	ID	Curves delete single curve in RAM
Read	0052h	ID	Curves read info and data size
Read	0053h	ID	Curves read info block data
Read	0054h	ID	Curves read setpoint data
Write	0055h	ID	Curves start write curve in RAM
Write	0056h	ID	Curves write curve info block data in RAM
Write	0057h	ID	Curves write curve set point data in RAM
Write	0058h	ID	Curves start modify curve in RAM
Write	0059h	ID	Curves modify curve info block data in RAM
Write	005Ah	ID	Curves modify curve set point data in RAM
Read	005Eh	ID Off	Curves read presence list
Write	005Fh	-	Curves start write curves from RAM to FLASH
Write	0060h	-	Command Table delete all entries in RAM
Write	0061h	ID	Command Table delete single entry in RAM
Read	0062h	ID	Command Table read data size
Read	0063h	ID	Command Table read data
Write	0065h	ID	Command Table start write entry in RAM
Write	0066h	ID	Command Table write entry data in RAM
Read	006Eh	ID Off	Command Table read presence list
Write	006Fh	-	Command Table start write entries from RAM to FLASH
Write	0070h	-	Reset device
Write	0071h	Inst	Stop SW instances
Write	0072h	Inst	Start SW instances

For a general description and an Overview of the concepts used in fieldbus configuration, please refer to the document *“LinMot 1100 Servo Controller Configuration over Fieldbus Interfaces”.*

4.2 ADS Parameter Services

The LinMot Servo controller supports a unique parameter access for all parameter data types (bit, byte, uint16, ..). The parameter data could always be mapped into a 4 bytes data field. The parameter itself is specified by its UPID (unique parameter ID).

4.2.1 Parameter UPID read RAM value

Read the RAM value of the parameter specified by its UPID.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0040h (LinMot Service ID)
Input	IDXOFFS	UDINT	UPID
Input	LEN	UDINT	always 4 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (4bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

Example:

```
VARIABLES:

        FbAdsRead:      ADSREAD;        (* instance of ADSREAD function block *)
        NetID:          T_AmsNetId;     (* AoE net ID of slave device *)
        Port:           T_AmsPort;      (* ADS port of slave device *
        TimeOut:        TIME;           (* time out value for AoE communication *)
        RdVal:          DWORD;          (* 4byte buffer for read response *)
        bErr:           BOOL;           (* flag for error return *)
        ErrId:          DUINT;          (* error ID *)

CODE:

(* starting parameter UPID read RAM value
(* should be called only once per reading
=======================================*)

FbAdsRead(READ:= FALSE);        (* reset READ input to assure rising edge for start reading *)
FbAdsRead(
        NETID   := NetID,       (* AoE NetId of slave device to read *)
        PORT    := Port,        (* ADS port of slave device to read *)
        IDXGRP  := 0x0040,      (* Parameter UPID read RAM value service ID *)
        IDXOFFS := 0x13A2,      (* UPID value of parameter P Gain of position controller *)
        LEN     := 4,           (* read length 4 bytes for all LinMot parameter types *)
        DESTADDR:= ADR(RdVal), (* pointer to read result data buffer *)
        READ    := TRUE,        (* generate rising edge on input for start reading *)
        TMOUT   := TimeOut,     (* timeout value for read command *)
        BUSY    := RdBusy,      (* busy output for detecting end of reading *)
        ERR     := bErr         (* error output for assure correct reading *)
        ERRID   := ErrId);      (* error id for debugging problem *)

(* polling for read response
(* should be called cyclic
============================*)
FbAdsRead(
        BUSY    := RdBusy,      (* busy output for detecting end of reading *)
        ERR     := bErr         (* error output for assure correct reading *)
        ERRID   := ErrId);      (* error id for debugging problem *)

IF(RdBusy = FALSE)
        IF(bErr = FALSE)
                PosCtrlPGain = DWORD_TO_UINT(RdVal)
        ELSE
                (* eval error condition *)
        END_IF
END_IF
```

4.2.2 Parameter UPID write RAM value

Write the RAM value of the parameter specified by its UPID.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0041h (LinMot Service ID)
Input	IDXOFFS	UDINT	UPID
Input	LEN	UDINT	always 4 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (4bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.2.3 Parameter UPID read ROM value

Read the ROM value of the parameter specified by its UPID.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0042h (LinMot Service ID)
Input	IDXOFFS	UDINT	UPID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (4bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.2.4 Parameter UPID write ROM value

Write the RAM value of the parameter specified by its UPID.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0043h (LinMot Service ID)
Input	IDXOFFS	UDINT	UPID
Input	LEN	UDINT	Always 4 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (4bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.2.5 Parameter start read UPID list

Specify the SW layer by its start UPID of which the UPID list will be read out. The list contains all parameters and variables of the SW layer.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0044h (LinMot Service ID)
Input	IDXOFFS	UDINT	Start UPID: 0000h: OS SW layer 1000h: MC SW layer 2000h: Intf SW layer 3000h: Appl SW layer
Input	LEN	UDINT	Always 8 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (8bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID: 00h: ok

4.2.6 Parameter get next UPID list item

Read the UPID, Address usage and ROM value of the parameter and the RAM value of the variables.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0045h (LinMot Service ID)
Input	IDXOFFS	-	Not evaluated
Input	LEN	UDINT	Always 8 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (8bytes)) Bytes 0..1 UPID Bytes 2..3 Address usage Bytes 4..7 ROM Value
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID: 00h: ok C6h: No new UPID found

4.2.7 Parameter start read modified UPID list

Specify the SW layer by its start UPID of which the UPID list will be read out. The list contains only the parameter that differs in the ROM value of its default value.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0046h (LinMot Service ID)
Input	IDXOFFS	UDINT	Start UPID: 0000h: OS SW layer 1000h: MC SW layer 2000h: Intf SW layer 3000h: Appl SW layer
Input	LEN	UDINT	0
Input	DESTADDR	DWORD	ADR(read_buffer (8bytes)) Bytes 0..1 UPID Bytes 2..3 Address usage Bytes 4..7 ROM Value
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID: 00h: ok

4.2.8 Parameter get next modified UPID list item

Read the UPID, Address usage and ROM value of the parameter of the defined SW layer specified by its start UPID.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0047h (LinMot Service ID)
Input	IDXOFFS	-	Not evaluated
Input	LEN	UDINT	Always 8 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (8bytes)) Bytes 0..1 UPID Bytes 2..3 Address usage Bytes 4..7 ROM Value
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID: 00h: ok C6h: No new UPID found

4.2.9 Parameter Default SW instance

With this command all parameters of one or more SW instances could be set to the default value.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0048h (LinMot Service ID)
Input	IDXOFFS	UDINT	SW instance selection Bit 0: OS SW layer Bit 1: MC SW layer Bit 2: Intferace SW layer Bit 3: application SW layer
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.3 ADS Curves Services

4.3.1 Curves delete all in RAM

Delete all curves in RAM.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0050h (LinMot Service ID)
Input	IDXOFFS	UDINT	Not evaluated
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.3.2 Curves delete single curve in RAM

Delete single curves in RAM.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0051h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID to delete
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.3.3 Curves read info and data size

Read the info block and curve setpoint data size of the specified curve.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0052h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (4bytes)) 0..1 : Info block size in bytes 0x0046 2..3 : setpoint data size in bytes (4 bytes per setpoint)
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.3.4 Curves read info block data

Read the info block data of the specified curve, this command has to be repeated 18 times.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0053h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (4bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.3.5 Curves read setpoint data

Read the setpoint data of the specified curve, this command has to be repeated until all data is read out.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0054h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (4bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.3.6 Curves start modify curve in RAM

Read the info block and curve setpoint data size of the specified curve.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0058h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(write _buffer (4bytes)) 0..1 : Info block size in bytes 0x0046 2..3 : setpoint data size in bytes (4 bytes per setpoint)
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.3.7 Curves modify curve info block data in RAM

Write the info block data of the specified curve, this command has to be repeated 18 times.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0059h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(write _buffer (4bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.3.8 Curves modify curve setpoint data in RAM

Write the setpoint data of the specified curve, this command has to be repeated until all data is written.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	005Ah (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(write _buffer (4bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.3.9 Curves start write curve to RAM

Initiate to write a curve to RAM. Then the “Curves write curve info block data to RAM” service has to be called several times. Only non existing curve ID’s could be written, if curve exists delete curve before (Command. “Curves delete single curve in RAM”). Repeating writing curves could lead to full curve memeory, in this case the curve memory has to be deleted completely with the “Delete all curves” command.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0055h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID to write
Input	LEN	UDINT	Always 4 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes)) Bytes 0..1 curve info block size Bytes 2..3 curve data block size
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.3.10 Curves write curve info block data to RAM

Write the curve header data to RAM. This command has to be repeated until the whole curve info block data is written. Then the “**Curves write curve info block data to RAM**” service has to be called several times.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0056h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID to write
Input	LEN	UDINT	Always 4 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (4bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.3.11 Curves write curve setpoint data to RAM

Write the curve setpoint data to RAM. This has to be repeated until the whole setpoint data is written

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0057h (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID to write
Input	LEN	UDINT	Always 4 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (4bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.3.12 Curves read presence list

Write the RAM value of the parameter specified by its UPID.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	005Eh (LinMot Service ID)
Input	IDXOFFS	UDINT	IDOff : 0: curves 1..32 32: curves 33..64 64: curves 65..96 96: curves 96..100
Input	LEN	UDINT	Always 4 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (4bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.3.13 Curves start write curves from RAM to FLASH

Copy the whole curve section from RAM to FLASH memory.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	005Fh (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID to write
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.4 ADS Command Table Services

4.4.1 Command Table delete all entries in RAM

Delete all command table entries in RAM.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0060h (LinMot Service ID)
Input	IDXOFFS	UDINT	Not evaluated
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.4.2 Command Table single curve in RAM

Delete single command table entry in RAM.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0061h (LinMot Service ID)
Input	IDXOFFS	UDINT	Command Table ID to delete
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.4.3 Command Table read data size

Read the info block and curve setpoint data size of the specified curve.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0062h (LinMot Service ID)
Input	IDXOFFS	UDINT	Command Table ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (4bytes)) 0..3 : Command Table size in bytes 0x0040
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.4.4 Command Table read data

Read the Command Table data this command has to be repeated until all 64 bytes are read out.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0063h (LinMot Service ID)
Input	IDXOFFS	UDINT	Command Table ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(read_buffer (4bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.4.5 Command Table start write entry in RAM

Start command to write a Command Table entry in the RAM.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0065h (LinMot Service ID)
Input	IDXOFFS	UDINT	Command Table ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(write_buffer (4bytes)) 0..3 : Command Table size in bytes 0x0040
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.4.6 Command Table write entry data in RAM

Write the Command Table data this command has to be repeated until all 64 bytes are written.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0066h (LinMot Service ID)
Input	IDXOFFS	UDINT	Command Table ID
Input	LEN	UDINT	Always 4 bytes
Input	DESTADDR	DWORD	ADR(write_buffer (4bytes))
Input	READ	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error id

4.4.7 Command table read presence list

Write the RAM value of the parameter specified by its UPID.

FUNCTION_BLOCK ADSREAD			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	006Eh (LinMot Service ID)
Input	IDXOFFS	UDINT	(Bit=0 means Entry exists) IDOff : 0: entries 1..31 32: entries 33..63 64: entries 65..95 96: entries 96..127 128: entries 128..159 160: entries 160..191 192: entries 192..223 224: entries 224..255
Input	LEN	UDINT	Always 4 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (4bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.4.8 Command Table start write entries from RAM to FLASH

Copy the whole Command Table section from RAM to FLASH memory.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	006Fh (LinMot Service ID)
Input	IDXOFFS	UDINT	Curve ID to write
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.5 Program handling

4.5.1 Reset device

Resets the device.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0070h (LinMot Service ID)
Input	IDXOFFS	UDINT	-
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value >6s
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

Typical restart time are 5 seconds, take this fact in account, when setting the ADS timeout value.

4.5.2 Stop SW instances

Stop the selected SW instances e.g. for flashing the curves.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0071h (LinMot Service ID)
Input	IDXOFFS	UDINT	SW instance selection Bit 0: MC-SW layer Bit 1: Intferace SW layer Bit 2: application SW layer
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

4.5.3 Start SW instances

Start the selected SW instances e.g. after flashing the curves.

FUNCTION_BLOCK ADSWRITE			
Direction	**Name**	**Type**	**Description**
Input	NETID	T_AmsNetId	AoE NetID
Input	PORT	T_AmsPort	ADS Port
Input	IDXGRP	UDINT	0072h (LinMot Service ID)
Input	IDXOFFS	UDINT	SW instance selection Bit 0: MC-SW layer Bit 1: Intferace SW layer Bit 2: application SW layer
Input	LEN	UDINT	Always 0 bytes
Input	SRCADDR	DWORD	ADR(write_buffer (0bytes))
Input	WRITE	BOOL	Rising edge starts reading
Input	TMOUT	TIME	Timeout value
Output	BUSY	BOOL	Indicates reading active/finished
Output	ERR	BOOL	Indicates reading error
Output	ERRID	UDINT	Error ID

5 EtherCAT Parameters and Variables

5.1 Parameters

The EtherCAT Interface has an additional parameter tree branch (Parameters → EtherCAT), which can be configured with the distributed LinMot-Talk software.
With these parameters, the EtherCAT interface can be enabled or disabled.
The LinMot-Talk software can be downloaded from http://www.linmot.com under the section download, software & manuals.

EtherCAT Dis-/Enable

With the Dis-/Enable parameter the LinMot servo controller can be run without the Ethernet EtherCAT Interface going online. So in a first step the system can be configured and run without any bus connection.

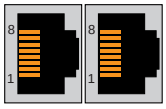
ETHERCAT\ Dis-/Enable	
Disable	Servo controller runs without ETHERCAT.
Enable	Servo controller runs with ETHERCAT connection.

IMPORTANT: If the ETHERCAT Interface is disabled, the integrated ETHERCAT-ASIC rests in reset state! No messages will be sent to other devices connected to the ETHERCAT-Network via the E1250 or E1450 controller.

6 Connecting to the EtherCAT Network

6.1 Pin Assignment of the Connectors X17-X18

The ETHERCAT connector is a standard RJ45 female connector with a pin assignment as defined by EIA/TIA T568B:

X17 – X18	ETHERCAT Connector		
	**Pin**	**Wire color code**	**Assignment 100BASE-TX**
	1	WHT/ORG	Rx+
	2	ORG	Rx-
	3	WHT/GRN	Tx+
	4	BLU	-
	5	WHT/BLU	-
	6	GRN	Tx-
	7	WHT/BRN	-
	8	BRN	-
	case	-	-
RJ-45	Use standard patch cables (twisted pair, S/UTP, AWG26) for wiring. This type of cable is usually referred to as a “Cat5e-Cable”.		

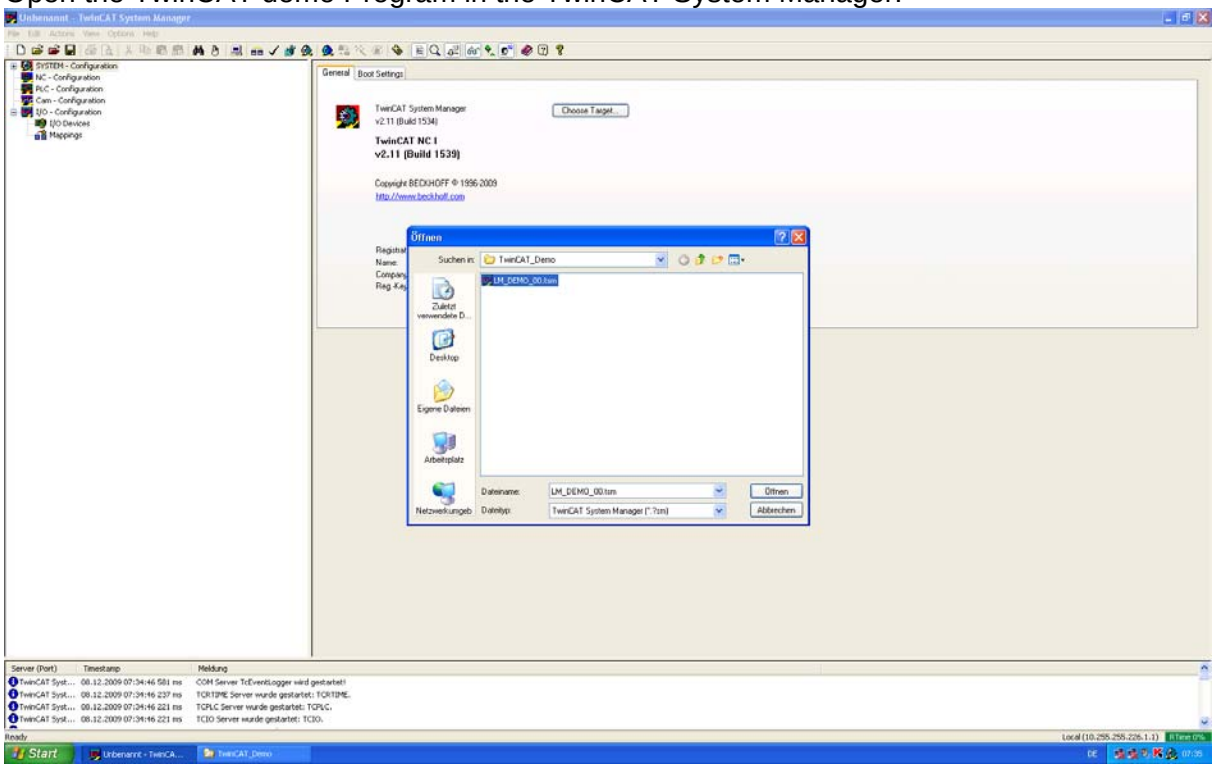
7 TwinCAT Demo Program

With the LinMot-Talk software a TwinCAT demo program is included as start point for new LinMot users, to get familiar with the concepts that stands behind LinMot motion SW. For getting started with this demo program Connect one LinMot EtherCAT Servo directly to the master Wire signal supply (24V Dc) and the motor power Supply. Wire also the “Safety Voltage Enable” on X4.12 to 24V. Then connect the motor to the servo power on the signal supply and configure the motor with the help of the LinMot-Talk software.

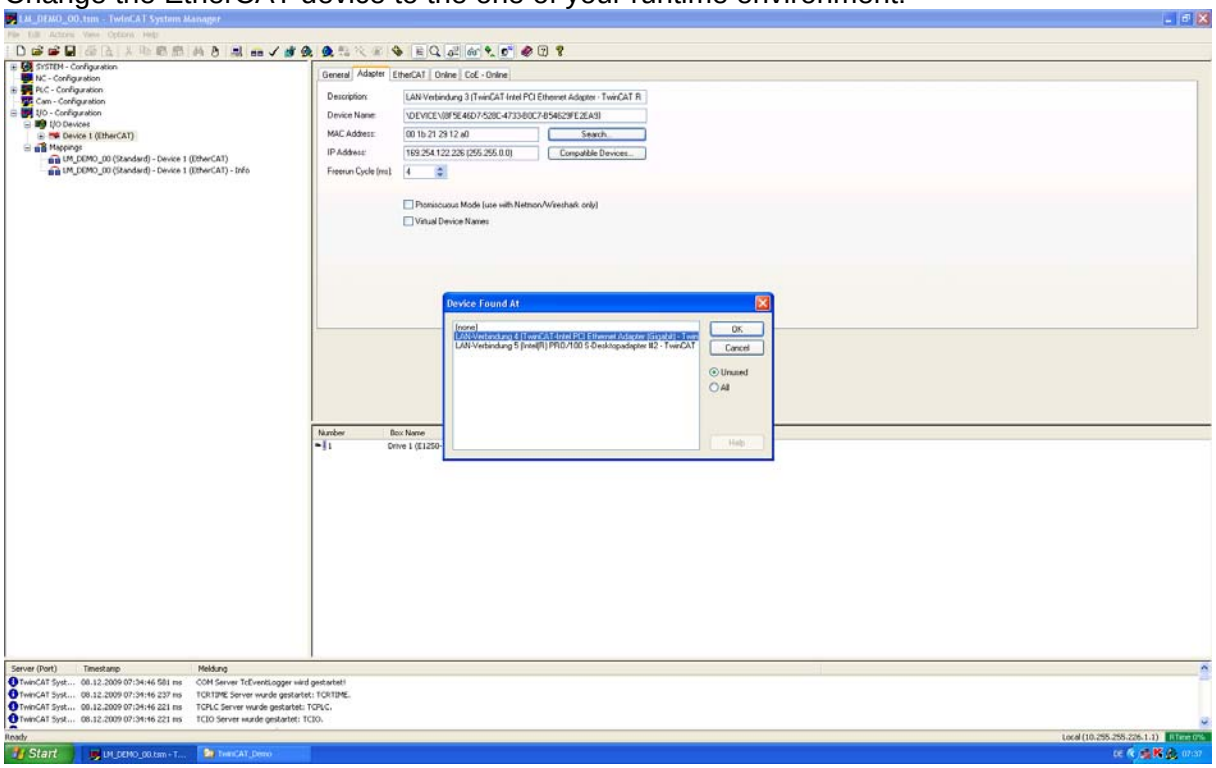
The demo program files are saved under the path:

C:\Program Files\LINMOT\LinMot-Talk 4 Build 20100616\Firmware\Interfaces\EtherCAT\TwinCAT_Demo\

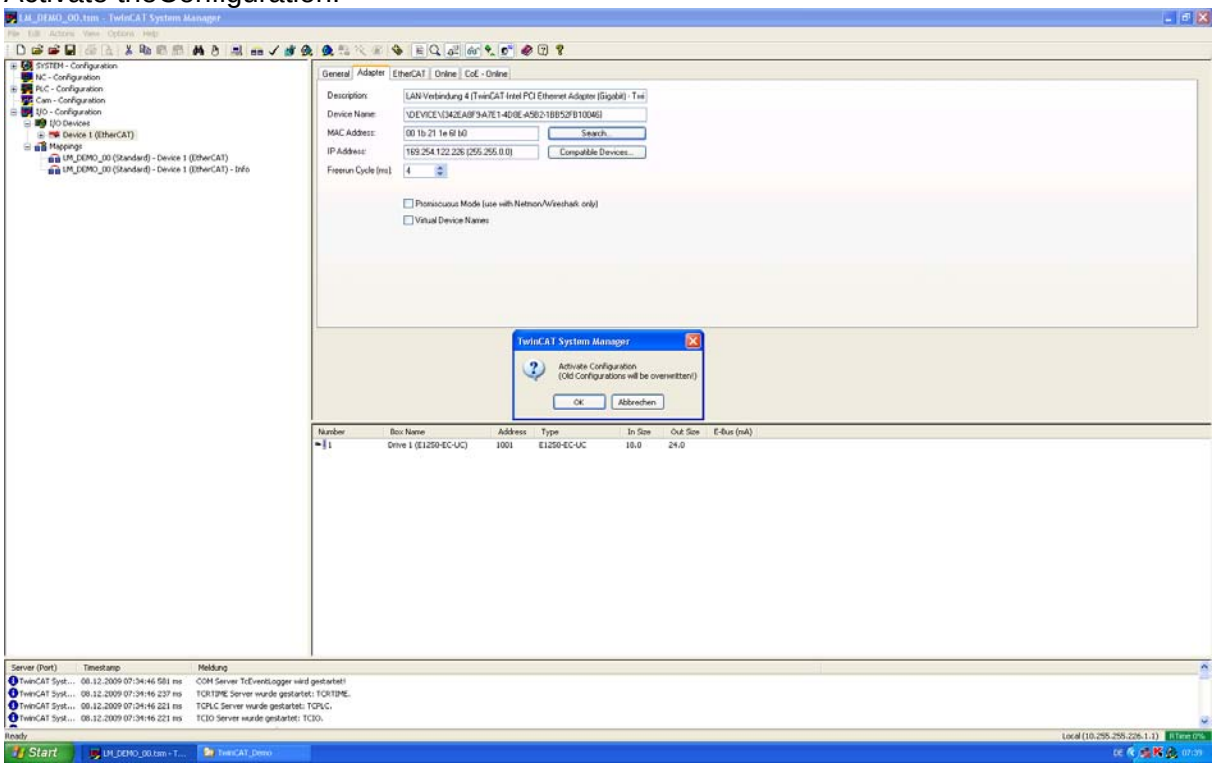
Open the TwinCAT demo Program in the TwinCAT System Manager:



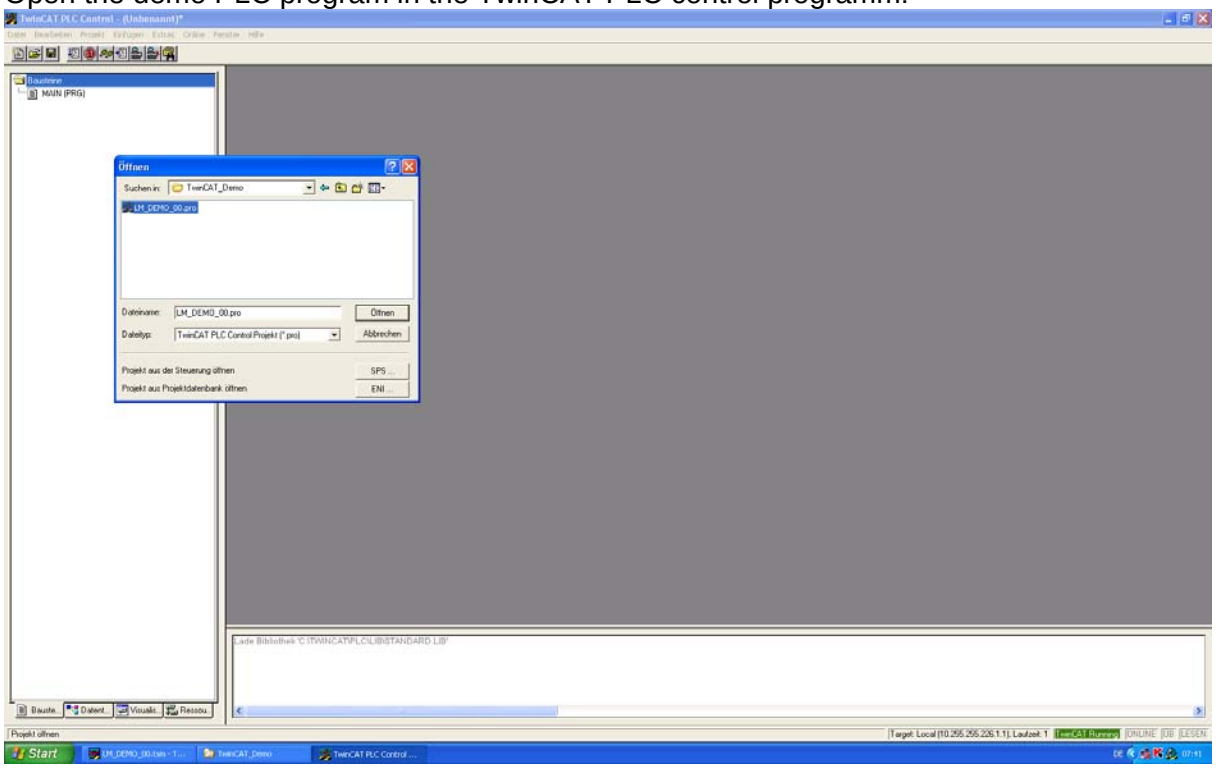
Change the EtherCAT device to the one of your runtime environment:



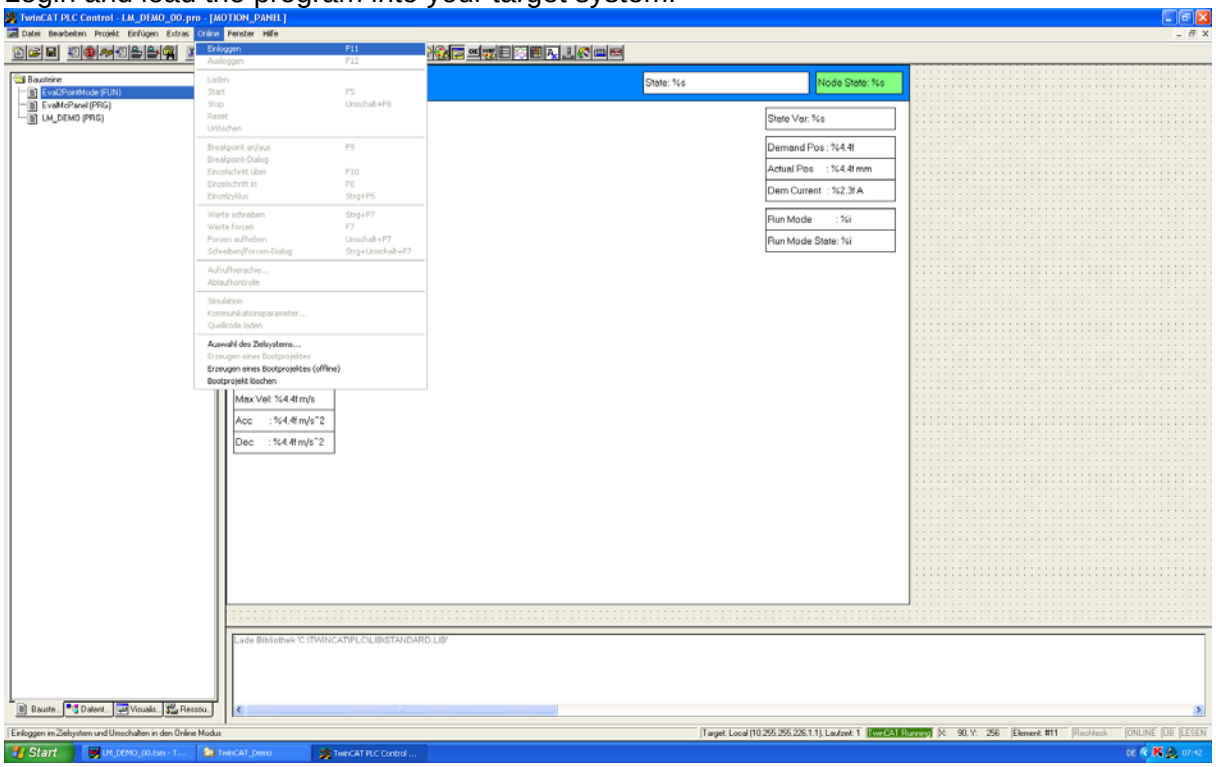
Activate theConfiguration:



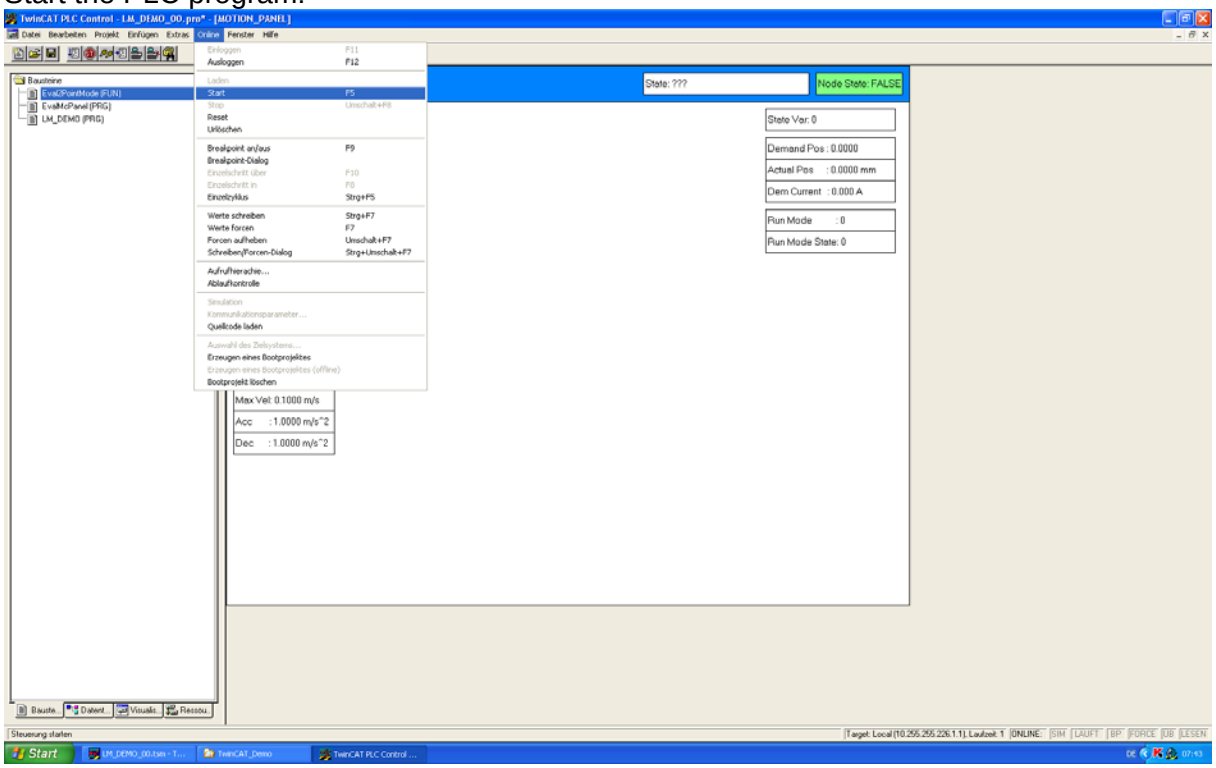
Open the demo PLC program in the TwinCAT PLC control programm:



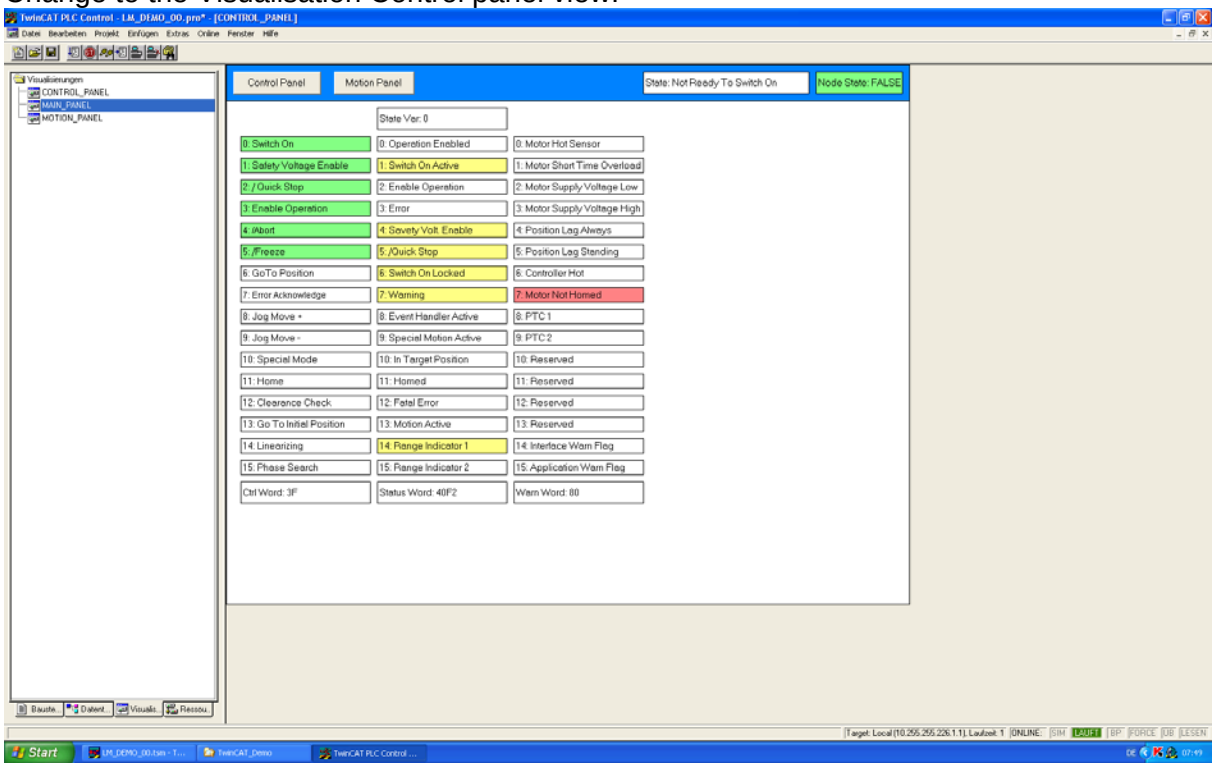
Login and load the program into your target system:



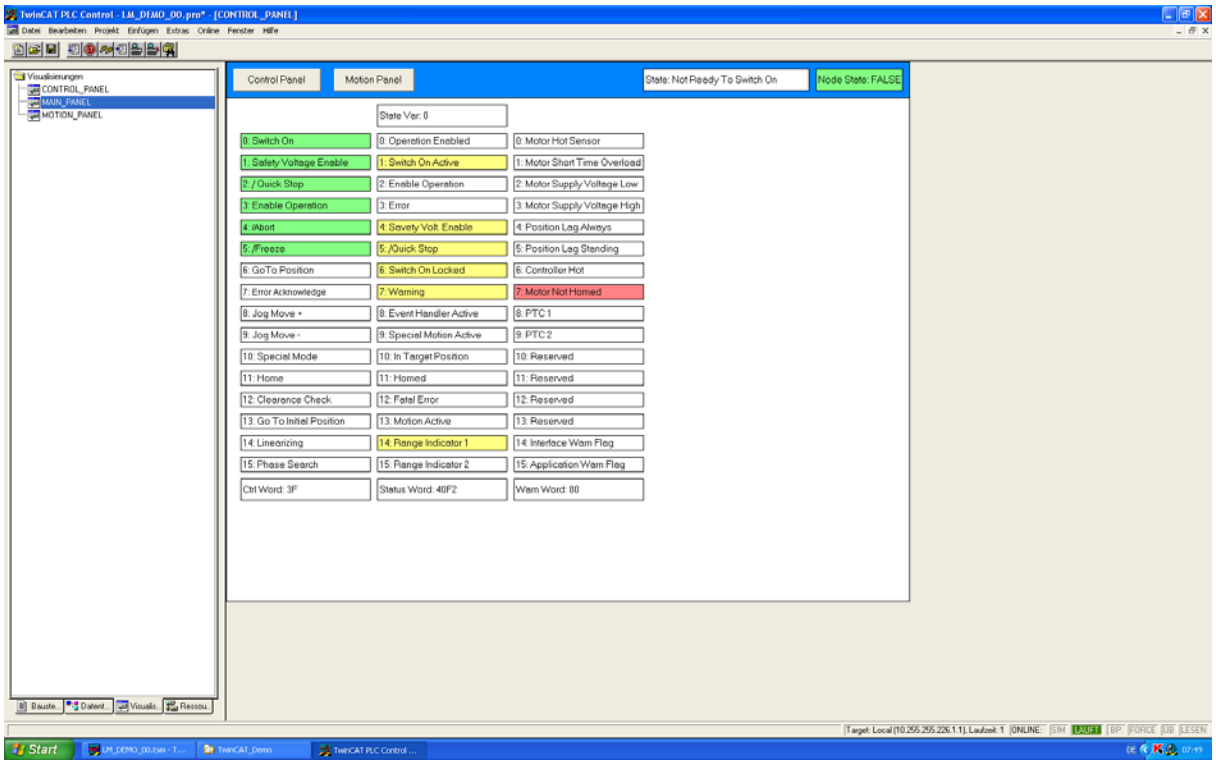
Start the PLC program:



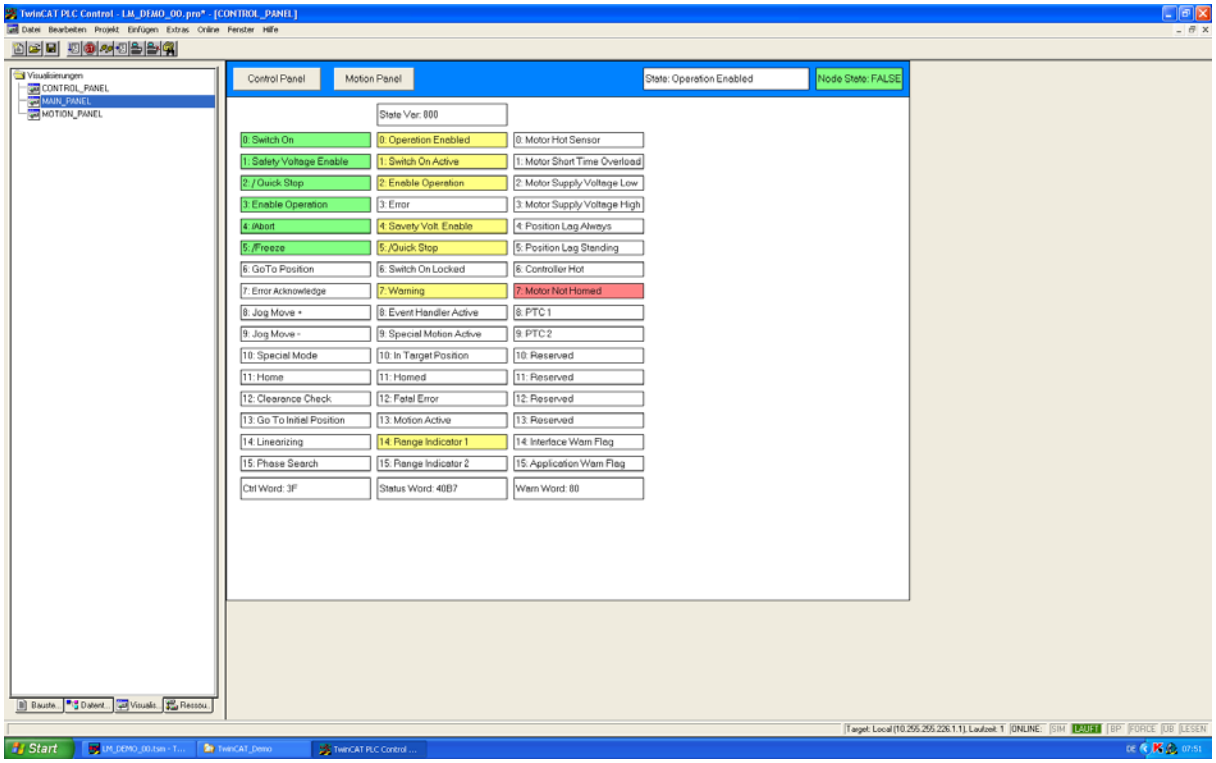
Change to the Visualisation Control panel view:



Release the Control Word bit 0 “Switch On”:

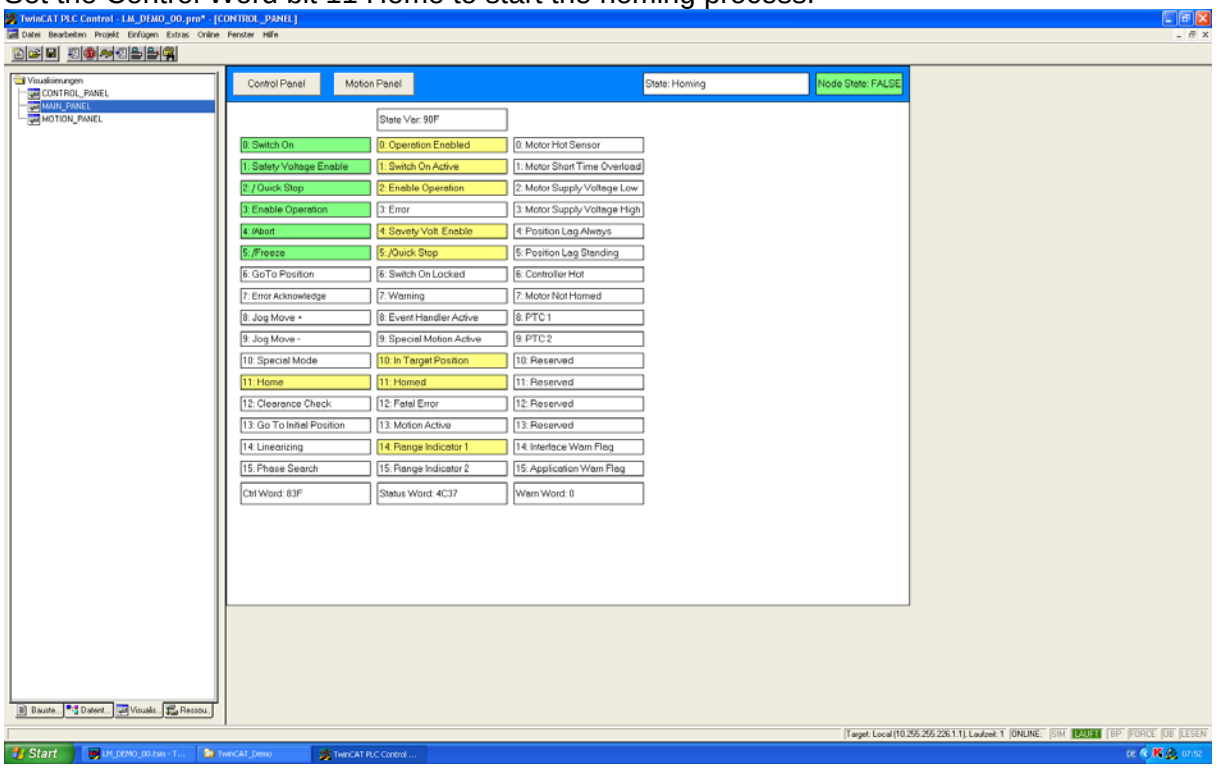


Set the Control Word bit 0 “Switch On”:



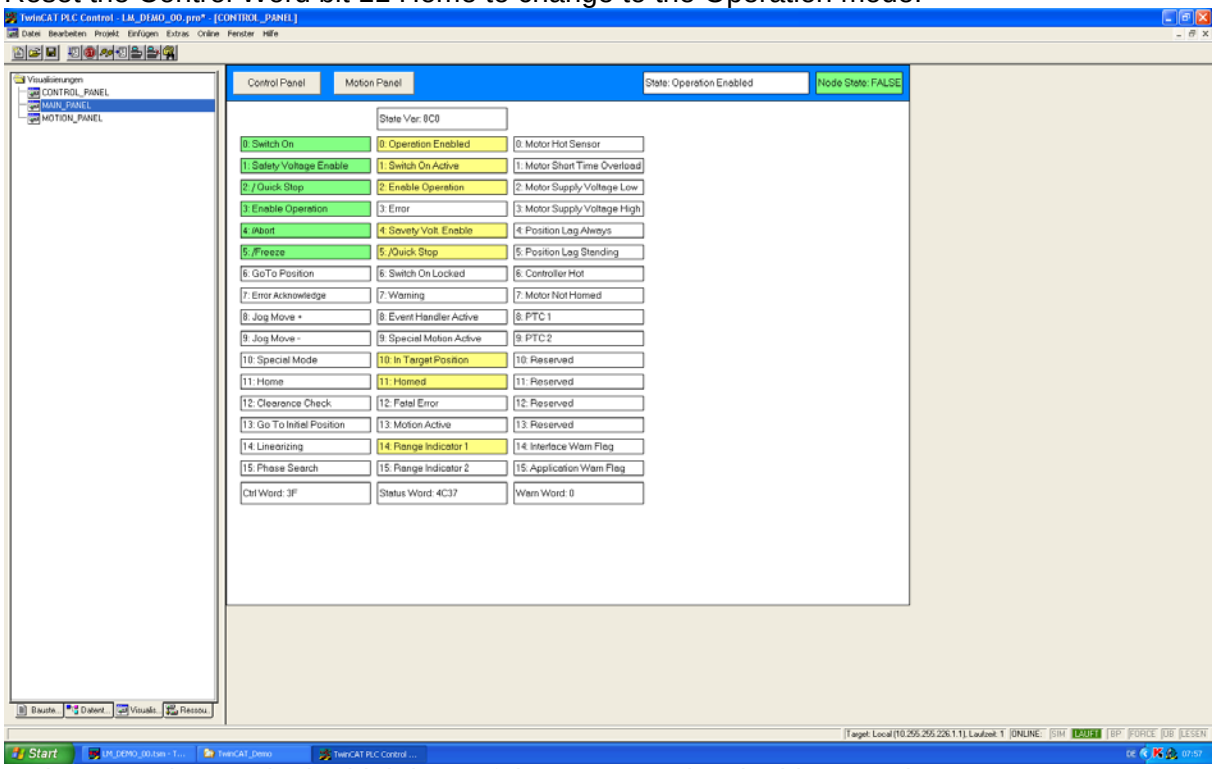
Now the motor is position controlled mode.

Set the Control Word bit 11 Home to start the homing process.



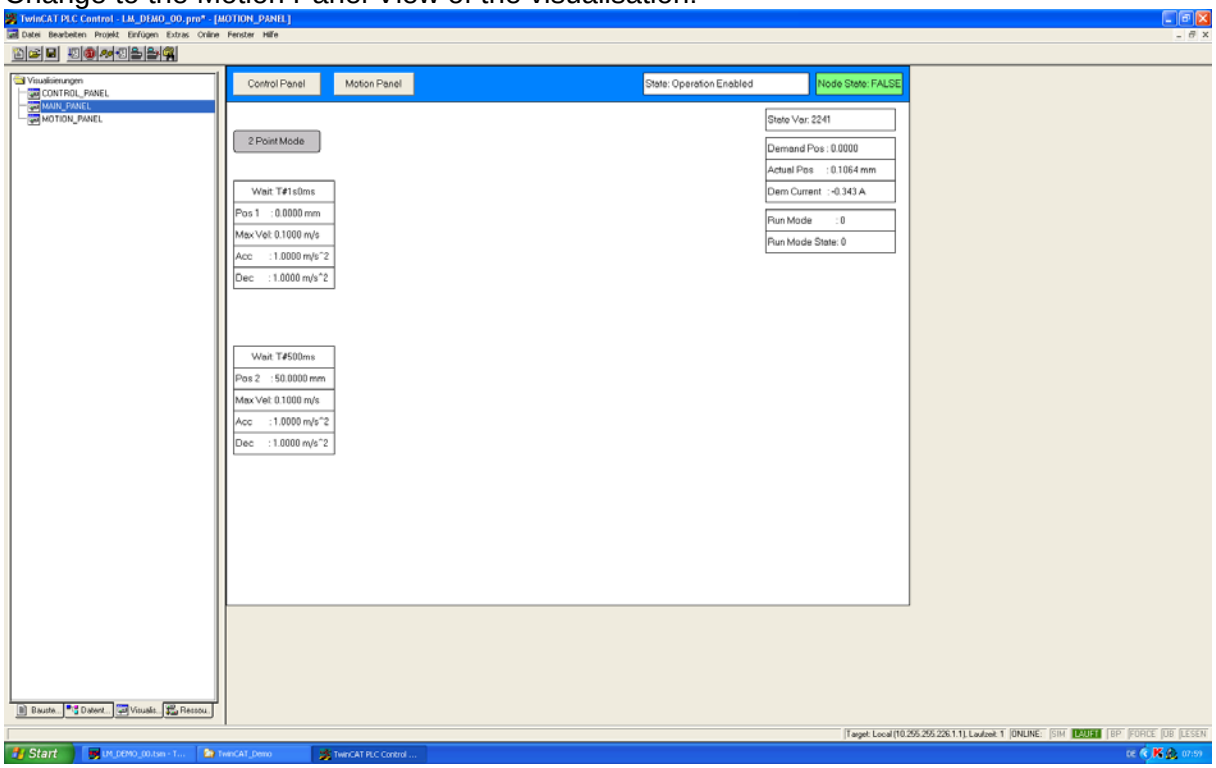
If the State Var changes to 0x90F the Homing is finished, which is also indicated by the vanished bit 7 in the Warn Word “Motor not Homed” and the set bit 11 in the Status Word “Homed”

Reset the Control Word bit 11 Home to change to the Operation mode:



In the operation mode the high byte of the State Var is 8 in this state the servo accepts motion commands.

Change to the Motion Panel View of the visualisation:



In the “2 Point Mode” the motor alternative goes from Pos 1 to Pos 2 this values maybe have to be adapted if your motor is limited to a smaller stroke. To activate the mode just press the “2 Point Mode” button. Now the motor should change between the two positions the motion parameters could be changed any time and are taken to account at the next start of the motion. The lowest nibble of the State Var contains Count nibble of the motion command header, every time this nibble changes, the motion control SW interprets the data of generic motion control interface which consists of the motion control header which selects the type of motion that is executed its count nibble to assure consistent data and the motion command parameters section which depends on the motion command.

8 Contact Addresses

SWITZERLAND

NTI AG / LinMot
Haerdlistr. 15
CH-8957 Spreitenbach

Sales and Administration: +41-(0)56-419 91 91
office@linmot.com

Tech. Support: +41-(0)56-544 71 00
support@linmot.com

Tech. Support (Skype) : skype:support.linmot

Fax: +41-(0)56-419 91 92
Web: http://www.linmot.com

USA

LinMot, Inc.
5750 Townline Road
Elkhorn, WI 53121

Sales and Administration: 877-546-3270
262-743-2555

Tech. Support: 877-804-0718
262-743-1284

Fax: 800-463-8708
262-723-6688

E-Mail: us-sales@linmot.com
Web: http://www.linmot-usa.com

Please visit http://www.linmot.com to find the distributor nearest to you.

Smart solutions are…