LinMot®

Ethernet POWERLINK Interface

Manual

This document applies to the following devices:

- **E1250-PL-xx-xx-xxx (SG5)**
- **E1450-PL-xx-xx-xxx (SG6)**
- **C1450-PL-xx-xx-xxx (SG6)**
- **C1250-PL-xx-xx-xxx (SG6)**
- **C1250-MI-xx-xx-xxx (SG6)**

(with POWERLINK Interface SW installed)

May 2022

Doc.: 0185-1088-E_1V21_MA_POWERLINK-SG5-SG6

© 2022 NTI AG
This work is protected by copyright.
Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, microfilm, storing in an information retrieval system, not even for didactical use, or translating, in whole or in part, without the prior written consent of NTI AG.

LinMot® is a registered trademark of NTI AG.

Note
The information in this documentation reflects the stage of development at the time of press and is therefore without obligation. NTI AG reserves itself the right to make changes at any time and without notice to reflect further technical advance or product improvement.

NTI AG
LinMot
Bodenaeckerstrasse 2
CH-8957 Spreitenbach

Tel.: +41 56 419 91 91
Fax.: +41 56 419 91 92
Email: office@LinMot.com
Homepage: www.LinMot.com

Table of Contents

1 SYSTEM OVERVIEW

The LinMot POWERLINK devices are Ethernet POWERLINK CNs (Controlled Nodes) with the following parameters (according to EPSG-DS-301-V1.1.0):

Device Property	Value
Minimal POWERLINK cycle time	200 µs
Size of isochronous transmit buffer (maximal size of isochronous frames)	64 Byte
Size of isochronous receive buffer (maximal size of isochronous frames)	64 Byte
Overall buffer size available for isochronous data	128 Byte
PReq to PRes latency (CN isochronous reaction time)	~ 1 µs
SoA to ASnd latency (CN asynchronous reaction time)	~ 1 µs
Maximum asynchronous MTU	300 Byte
Ability to support multiplexed isochronous access	No
Asynchronous SDO transfer method (UDP/IP and/or POWERLINK ASnd)	POWERLINK ASnd only on C1250-PL, UDP/IP supported only on C1250-MI drives

For further information on Ethernet POWERLINK please visit:
http://www.ethernet-powerlink.org

2 CONNECTING TO THE POWERLINK NETWORK

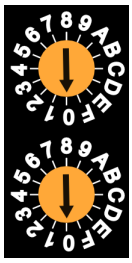
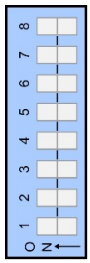
2.1 PIN ASSIGNMENT OF THE CONNECTORS X17-X18

The POWERLINK 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 2 3 4 5 6 7 8 case	WHT/ORG ORG WHT/GRN BLU WHT/BLU GRN WHT/BRN BRN -	Rx+ Rx- Tx+ - - Tx- - - -
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”.		

2.2 SETTING THE NODEID

The NodeID is set via the two Hex-Switches S1 and S2, where S1 sets the high digit and S2 the low digit. The NodeID can have a value between 1 (01h) and 239 (EFh).

S1, S2		NodeID Selectors	
E12x0	**C12x0 / E14x0**		
		S1 (5..8) S2 (1..4)	Bus ID High (0 … F). Bit 5 is the LSB, bit 8 the MSB. Bus ID Low (0 … F). Bit 1 is the LSB, bit 4 the MSB. **Setting the ID high & low to 0xFF resets the drive to manufacturer settings!**

3 POWERLINK PARAMETERS AND VARIABLES

3.1 PARAMETERS

The POWERLINK Interface has an additional parameter tree branch (Parameters → POWERLINK), which can be configured with the distributed LinMot-Talk software.
With these parameters, the POWERLINK behaviour can be configured.

The LinMot-Talk software can be downloaded from http://www.linmot.com under the section download, software & manuals.

3.1.1 Dis-/Enable

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

POWERLINK\ Dis-/Enable	
Disable	Device runs without POWERLINK.
Enable	Device runs with POWERLINK.

Important: If the POWERLINK Interface is disabled, the integrated POWERLINK-Hub is not powered! No messages will be sent to other devices connected to the POWERLINK-Network via the LinMot device.

3.1.2 NodeID

In this section the NodeID of the POWERLINK CN can be configured.

POWERLINK\ NodeID	
NodeID Source Select	Shows which source is selected to provide the NodeID.
NodeID Parameter Value	Value of the NodeID if 'By Parameter' is selected. The default value of this parameter is 63 (3Fh).

3.1.3 NodeID\ NodeID Source Select

In this section the source of the NodeID of the POWERLINK CN can be configured.

POWERLINK\ NodeID\ NodeID Source Select	
By Hex Switches S1 and S2	The NodeID is determined by the Switches S1 (ID High) and S2 (ID Low).
By Parameter	The NodeID is determined by the Parameter 'NodeID Parameter Value' in the LinTalk1100 Software.

3.2 VARIABLES

The POWERLINK Interface has an additional parameter tree branch (Parameters → POWERLINK), which can be configured with the distributed LinMot-Talk software.

Name	Type	Definition
Node-ID	UInt16	Current Node-ID
NMT State	UInt16 Enumerator	Current NMT State

NMT_CS_NOT_ACTIVE
NMT_CS_NOT_ACTIVE is a non-permanent state which allows a starting node to recognize the current network state.

NMT_CS_PRE_OPERATIONAL_1
In NMT_CS_PRE_OPERATIONAL_1 the node is identified by the master via IdentRequest. The transition from NMT_CS_PRE_OPERATIONAL_1 to the following state is triggered by a SoC frame being received.

NMT_CS_PRE_OPERATIONAL_2

In the state NMT_CS_PRE_OPERATIONAL_2, the drive waits for the configuration to be completed. Precondition for the transition from NMT_CS_PRE_OPERATIONAL_2 to NMT_CS_READY_TO_OPERATE is the reception of an NMTEnableReadyToOperate command.

NMT_CS_READY_TO_OPERATE

With the state NMT_CS_READY_TO_OPERATE, the drive shall signal its readiness to operation to the master.

NMT_CS_OPERATIONAL

NMT_CS_OPERATIONAL is the normal operating state of the drive.

NMT_CS_STOPPED

In the NMT_CS_STOPPED state, the node shall be largely passive. NMT_CS_STOPPED shall be used for controlled shutdown of a selected CN while the system is still running.

4 MAPPING OF PRES AND PREQ

The LinMot drive is a POWERLINK CN (Controlled Node). To configure it with a POWERLINK master, the corresponding XDD file is used. The XDD File can be found in the installation directory.
(typically C:\Program Files\LinMot\LinTalk x.x\Firmware\Interfaces\POWERLINK\XDD)

4.1 PDO MAPPING

Index	Sub-Index	Name	Object Type	Access Type	PDO Mapping
4F00h	**-**	**MCSW_Variables[3]**	**RECORD**	**const**	**-**
	01h	MCSW_StateVar	UInt16	ro	optional
	02h	MCSW_StatusWord	UInt16	ro	optional
	03h	MCSW_WarnWord	UInt16	ro	optional
	04h	MCSW_ActualPosition32Bit	Int32	ro	optional
	05h	MCSW_DemandPosition32Bit	Int32	ro	optional
	06h	MCSW_DemandCurrent	Int32	ro	optional
	07h	MCSW_ControlWord	UInt16	wo	optional
	08h	MCSW_MotionCommandHeader	UInt16	wo	optional
	09h	MCSW_MotionCommandByte_00_03	UInt32	wo	optional
	0Ah	MCSW_MotionCommandByte_04_07	UInt32	wo	optional
	0Bh	MCSW_MotionCommandByte_08_11	UInt32	wo	optional
	0Ch	MCSW_MotionCommandByte_12_15	UInt32	wo	optional
	0Dh	MCSW_MotionCommandByte_16_19	UInt32	wo	optional
	0Eh	MCSW_MotionCommandByte_20_23	UInt32	wo	optional
	0Fh	MCSW_MotionCommandByte_24_27	UInt32	wo	optional
	10h	MCSW_MotionCommandByte_00_01	UInt16	wo	optional
	11h	MCSW_MotionCommandByte_02_03	UInt16	wo	optional
	12h	MCSW_MotionCommandByte_04_05	UInt16	wo	optional
	13h	MCSW_MotionCommandByte_06_07	UInt16	wo	optional
	14h	MCSW_MotionCommandByte_08_09	UInt16	wo	optional
	15h	MCSW_MotionCommandByte_10_11	UInt16	wo	optional
	16h	MCSW_MotionCommandByte_12_13	UInt16	wo	optional
	17h	MCSW_MotionCommandByte_14_15	UInt16	wo	optional
	18h	MCSW_MotionCommandByte_16_17	UInt16	wo	optional
	19h	MCSW_MotionCommandByte_18_19	UInt16	wo	optional
4F02h	**-**	**UPID_Variables**	**RECORD**	**const**	**-**
	01h	Parameter_by_UPID_01	UInt32	rw	optional
	02h	Parameter_by_UPID_02	UInt32	rw	optional
	03h	Parameter_by_UPID_03	UInt32	rw	optional
	04h	Parameter_by_UPID_04	UInt32	rw	optional
4F03h	**-**	**Configuration_Module[1]**	**RECORD**	**const**	**-**
	01h	TX_Cfg_Module_Control	UInt16	wo	optional

Index	Sub-Index	Name	Object Type	Access Type	PDO Mapping
	02h	TX_Cfg_Module_Index_Out	UInt16	wo	optional
	03h	TX_Cfg_Module_Value_Out	UInt32	wo	optional
	04h	RX_Cfg_Module_Status	UInt16	ro	optional
	05h	RX_Cfg_Module_Index_In	UInt16	ro	optional
	06h	RX_Cfg_Module_Value_In	UInt32	ro	optional
4F10h	**-**	**Process Monitoring Readout[2]**	**RECORD**	**const**	**-**
	01h	PM Readout Control	UInt32	rw	optional
	02h	PM Readout Status	UInt32	ro	optional
	03h	PM Readout X-Value 01	UInt32	ro	optional
	04h	PM Readout X-Value 02	UInt32	ro	optional
	05h	PM Readout X-Value 03	UInt32	ro	optional
	06h	PM Readout X-Value 04	UInt32	ro	optional
	07h	PM Readout X-Value 05	UInt32	ro	optional
	08h	PM Readout X-Value 06	UInt32	ro	optional
	09h	PM Readout X-Value 07	UInt32	ro	optional
	0Ah	PM Readout X-Value 08	UInt32	ro	optional
	0Bh	PM Readout X-Value 09	UInt32	ro	optional
	0Ch	PM Readout X-Value 10	UInt32	ro	optional
	0Dh	PM Readout X-Value 11	UInt32	ro	optional
	0Eh	PM Readout X-Value 12	UInt32	ro	optional
	0Fh	PM Readout X-Value 13	UInt32	ro	optional
	10h	PM Readout X-Value 14	UInt32	ro	optional
	11h	PM Readout X-Value 15	UInt32	ro	optional
	12h	PM Readout X-Value 16	UInt32	ro	optional
	13h	PM Readout Y-Value 01	UInt32	ro	optional
	14h	PM Readout Y-Value 02	UInt32	ro	optional
	15h	PM Readout Y-Value 03	UInt32	ro	optional
	16h	PM Readout Y-Value 04	UInt32	ro	optional
	17h	PM Readout Y-Value 05	UInt32	ro	optional
	18h	PM Readout Y-Value 06	UInt32	ro	optional
	19h	PM Readout Y-Value 07	UInt32	ro	optional
	1Ah	PM Readout Y-Value 08	UInt32	ro	optional
	1Bh	PM Readout Y-Value 09	UInt32	ro	optional
	1Ch	PM Readout Y-Value 10	UInt32	ro	optional
	1Dh	PM Readout Y-Value 11	UInt32	ro	optional
	1Eh	PM Readout Y-Value 12	UInt32	ro	optional
	1Fh	PM Readout Y-Value 13	UInt32	ro	optional
	20h	PM Readout Y-Value 14	UInt32	ro	optional

Index	Sub-Index	Name	Object Type	Access Type	PDO Mapping
	21h	PM Readout Y-Value 15	UInt32	ro	optional
	22h	PM Readout Y-Value 16	UInt32	ro	optional

Parameters 4F00hSub10h-19h are mappable for easy interface compatibility with legacy drives. Use the 32 Bit Versions if possible. The MCSW_MotionCommandByte_xx_xx UInt32 parameters shall not be configured together with their UInt16 counterparts.

1 See chapter „5.1 LinMot Configuration Module via PDO" 12 for details
2 These optionally mappable parameters can only be used in conjunction with the Process Monitoring application. This application is an optionally installable software package on LinMot SG6 devices. For a detailed description please refer to the user manual "0185-1172-E_xVx_MA_ProcessMonitoring-Application".
3 If the drive is used with the DS402 profile, do not configure the ControlWord, MotionCommandHeader and MotionCommandBytes as these variables are directly controlled and written in the DS402 profile functionality.

4.2 ADDITIONAL PARAMETERS

These Parameters can be set to configure additional services.
They are written once during the initialisation phase by the PLC to the drive.

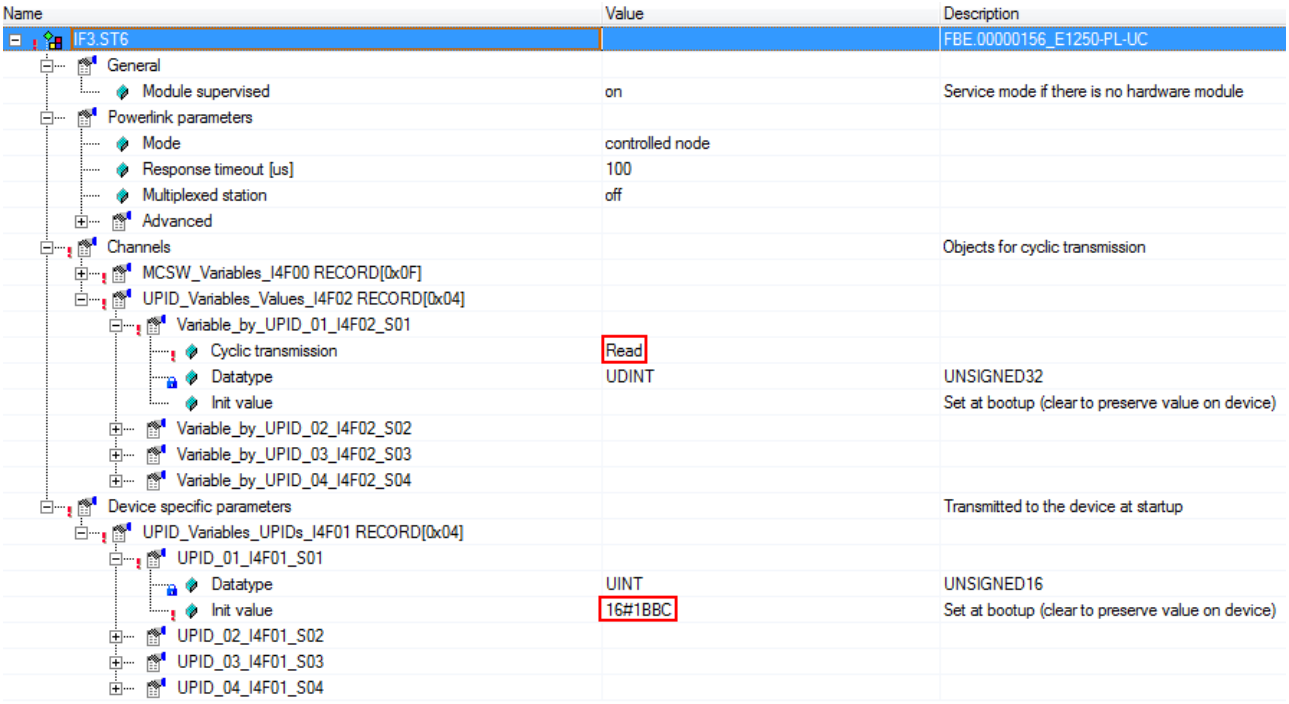
Index	Sub-Index	Name	Object Type	Access Type	PDO Mapping
4F01h	01h	UPID_01	UInt16	wo	no
4F01h	02h	UPID_02	UInt16	wo	no
4F01h	03h	UPID_03	UInt16	wo	no
4F01h	04h	UPID_04	UInt16	wo	no

4.3 MAPPING OF PARAMETERS VIA THEIR UPID

A total of four parameters are mappable to the poll response or the poll request for the drive by their UPIDs.
Any parameter of the drive which has its own UPID is mappable this way in order to achieve a high degree of flexibility for the user.

4.3.1 Example how to read a parameter via its UPID

In this example, the UPID 1BBCh (Logic Supply Voltage) is configured so that its value is sent with the poll response of the drive.



Settings for UPID mapping in the B&R Automation Studio

5 CONFIGURATION VIA POWERLINK INTERFACE

To configure the drive via the POWERLINK interface one has the following possibilities:

- Using the Configuration Module which is embedded in the PDO data.
- Using specific SDO-Commands

Both methods can be used independently from one another. One has to be aware that those methods are different from each other, although their functionality is largely the same.

5.1 LINMOT CONFIGURATION MODULE VIA PDO

This type of configuration access is supported on all type of ethernet fieldbus interfaces of the LinMot drive family.
For a detailed description on how to use this module please consult the manual “LinMot drive Configuration over Fieldbus Interfaces SG5”.

5.2 LINMOT PARAMETER COMMANDS VIA SDO

All LinMot-specific Services like the Curve-Service, the Command-Table-Service etc. are accessible over the POWERLINK Interface via SDO-Commands.

Index	Sub-Index	Description	Access Type
2000h - 5FFFh		LinMot Parameter Commands	-
UPID Commands			
	01h	RAM Value of UPID	rw
	02h	ROM Value of UPID	rw
	03h	Min Value of UPID	ro
	04h	Max Value of UPID	ro
	05h	Default Value of UPID	ro
	06h	RAM / ROM Write of UPID	wo
	20h	Start getting UPID List	wo
	21h	Get Next UPID List item	ro
	22h	Start Getting Modified UPID List	wo
	23h	Get Next Modified UPID List item	ro
System Commands			
	07h	Set ROM to default (OS)	wo
	08h	Set ROM to default (MC)	wo
	09h	Set ROM to default (Interface)	wo
	0Ah	Set ROM to default (Application)	wo
	0Bh	Reset drive	wo
	35h	Stop MC and APPL Software	wo
	36h	Start MC and APPL Software	wo
Curve Service Commands			
	40h	Curve Service: Save to Flash	wo

Index	Sub-Index	Description	Access Type
	41h	Curve Service: Delete all Curves (RAM)	wo
	42h	Curve Service: Poll Flash	ro
	50h	Curve Service: Add Curve	wo
	53h	Curve Service: Add Curve Data (32 Bit)	wo
	54h	Curve Service: Add Curve Info Block (32 Bit)	wo
	60h	Curve Service: Get Curve	ro
	61h	Curve Service: Get Curve Info Block	ro
	62h	Curve Service: Get Curve Data	ro
Error Log Commands			
	70h	Get Error Log Entry Counter	ro
	71h	Get Error Log Entry Error Code	ro
	72h	Get Error Log Entry Time low	ro
	73h	Get Error Log Entry Time high	ro
	74h + (Stringlet Nr. 0..7)	Get Error Code Text Stringlet	ro
Command Table Commands			
	80h	CT: Save to Flash	wo
	80h	CT: Poll Flash	ro
	81h	CT: Delete all Entries (RAM)	wo
	82h	CT: Delete Entry (Entry Nr.)	wo
	83h	CT: Write Entry (Entry Nr.)	wo
	84h	CT: Write Entry Data	wo
	85h	CT: Get Entry (Entry Nr.)	ro
	86h	CT: Get Entry Data	ro
	87h	CT: Get Entry List (Entry 0..31)	ro
	88h	CT: Get Entry List (Entry 32..63)	ro
	89h	CT: Get Entry List (Entry 64..95)	ro
	8Ah	CT: Get Entry List (Entry 96..127)	ro
	8Bh	CT: Get Entry List (Entry 128..159)	ro
	8Ch	CT: Get Entry List (Entry 160..191)	ro
	8Dh	CT: Get Entry List (Entry 192..223)	ro
	8Eh	CT: Get Entry List (Entry 224..255)	ro

For a general description and a Overview of the concepts used in fieldbus configuration, please refer to the document “LinMot Drive Configuration over Fieldbus Interfaces SG5”.

5.2.1 UPID Commands

Parameters can be modified via their UPIDs (Unique Parameter ID). To use a UPID command, an SDO read or write has to be performed on the index “2000h + UPID”.
The sub-index specifies the command which is performed.

Index	Sub-Index	Description	Access Type	Data Type
2000h + UPID	01h	RAM Value	rw	Bool - Unsigned32

Read / Write the RAM Value of a UPID. Any UPID from a Boolean type up to an unsigned32 type can be read or written.

Read RAM Value by UPID
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	01h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	01h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Read Value of the UPID

Write RAM Value by UPID
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	01h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Value of the UPID to write

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	01h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + UPID	02h	ROM Value	rw	Bool - Unsigned32

Read / Write the ROM Value of a UPID. Any UPID from a Boolean type up to an unsigned32 type can be read or written. If a value in the ROM is changed, it is not immediately reflected in the RAM. Values are read from the ROM to the RAM on startup of the drive. Use the “RAM/ROM Write” command (sub-index 06h) to changes both values at the same time.

Read RAM Value by UPID
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h +	02h		Payload Data	-	-	-	-

Index	Sub-Index	Description	Access Type	Data Type			
UPID			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	02h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Read Value of the UPID

Write RAM Value by UPID
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	02h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Value of the UPID to write

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	02h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + UPID	04h	Max Value	ro	Bool - Unsigned32

The maximal possible value of the UPID is returned.

Read Rax Value by UPID
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	04h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	04h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Max Value of the UPID

Index	Sub-Index	Description	Access Type	Data Type
2000h + UPID	05h	Default Value	ro	Bool - Unsigned32

The default value of the UPID is returned.

Read Default Value by UPID
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	05h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	05h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Default Value of the UPID

Index	Sub-Index	Description	Access Type	Data Type
2000h + UPID	06h	RAM/ROM Write	wo	Bool - Unsigned32

Write the RAM and ROM Value of a UPID. Any UPID from a Boolean type up to an unsigned32 type can be written.

Write RAM/ROM Value by UPID
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	06h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Value of the UPID to write

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	06h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	20h	Start Getting UPID List	wo	Unsigned16

With this command, the starting UPID can be set from which the command “Get Next UPID List item” begins returning info when called. This command has to be sent at least once before information on UPIDs can be retrieved via the “Get Next UPID List item” command.

Index	Sub-Index	Description	Access Type	Data Type

Write RAM/ROM Value by UPID
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	20h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	20h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	21h	Get Next UPID List item	ro	Unsigned32

With this command, the starting UPID can be set from which the command “Get Next UPID List item” begins returning info when called. This command has to be sent at least once before information on UPIDs can be retrieved via the “Get Next UPID List item” command.

Get Next UPID List item
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	21h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	21h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

yy yy: Address Usage
xx xx: UPID which was found

Address Usage is interpreted as follows:

Index	Sub-Index	Description	Access Type	Data Type

	RAM Read allowed	RAM Write allowed	ROM Read allowed	ROM Write allowed				Live Parameter				Parameter is included when calculating the hash table.	This Parameter is used for security Features.		
Bit Nr.:	0 (LSB)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 (MSB)

Index	Sub-Index	Description	Access Type	Data Type
2000h	**22h**	**Start Getting Modified UPID List**	**wo**	**Unsigned16**
This command is used in the same way as the “Start Getting UPID List” command (2000h sub 20h). Only UPIDs with values that differ from their default values are returned.				

Index	Sub-Index	Description	Access Type	Data Type
2000h	**23h**	**Get Next Modified UPID List item**	**ro**	**Unsigned32**
This command is used the in same way as the “Get Next UPID List item” command (2000h sub 21h). Only UPIDs with values that differ from their default values are returned.				

5.2.2 System Commands

Commands like resetting values of parameters, resetting the device etc. can be found here.

Index	Sub-Index	Description	Access Type	Data Type
2000h	07h	Default Value	wo	Unsigned8 - Unsigned32

Set all parameters of the OS to default values. This command needs about 0.5s to finish. Any data can be written for the command to be executed.

Set ROM to default (OS)
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	07h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	07h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	08h	Set ROM to default (MC SW)	wo	Unsigned8 - Unsigned32

Set all parameters of the MC SW to default values. This command needs about 2s to finish. Any data can be written for the command to be executed.

Set ROM to default (MC SW)
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + UPID	08h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	08h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	09h	Set ROM to default (Interface)	wo	Unsigned8 - Unsigned32

Set all parameters of the interface to default values. This command needs about 0.5s to finish. Any data can be written for the command to be executed.

Set ROM to default (Interface)
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	Description	Access Type	Data Type

Index	Sub-Index	T		LSB			MSB
2000h	09h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	09h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	09h	Set ROM to default (Interface)	wo	Unsigned8 - Unsigned32

Set all parameters of the interface to default values. This command needs about 0.5s to finish. Any data can be written for the command to be executed.

Set ROM to default (Interface)
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	09h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	09h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	0Ah	Set ROM to default (Application)	wo	Unsigned8 - Unsigned32

Set all parameters of the Application to default values. This command needs about 0.5s to finish. Any data can be written for the command to be executed.

Set ROM to default (Application)
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	0Ah		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Index	Sub-Index	Description	Access Type	Data Type

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	0Ah		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	0Bh	Reset Drive	wo	Unsigned8 - Unsigned32

Initiates a software reset of the drive. Any data can be written for the command to be executed.

Reset Drive
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	0Bh		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	0Bh		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	35h	Stop MC and APPL SW	wo	Unsigned8 - Unsigned32

MC SW and Application SW are stopped. Any data can be written for the command to be executed.

Stop MC and APPL Software
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	35h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	35h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	36h	Start MC and APPL SW	wo	Unsigned8 - Unsigned32

Index	Sub-Index	Description	Access Type	Data Type

MC SW and Application SW are started. Any data can be written for the command to be executed.

Stop MC and APPL Software
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	**36h**		**Payload Data**	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	**36h**		**Payload Data**	-	-	-	-
			Byte	01	02	03	04

5.2.3 Curve Service Commands

See the “LinMot 1100 Drive Configuration over Fieldbus Interfaces” for additional detail on the use of curve commands and a description of the content of the curve info and data blocks.

Index	Sub-Index	Description	Access Type	Data Type
2000h	40h	Curve Service: Save to Flash	wo	Unsigned8 - Unsigned32

All curves are saved from the RAM to the flash and are thus permanently saved. MC SW and application have to be stopped in order for this command to work (see command 2000h sub 35: Stop MC and Application Software). Any data can be written for the command to be executed.

Curve Service: Save to Flash
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	40h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	40h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	41h	Curve Service: Delete all Curves (RAM)	wo	Unsigned8 - Unsigned32

All curves in the RAM are deleted. This does NOT delete curves from the flash. After a system reset, the curves are loaded again from the flash to the RAM. Any data can be written for the command to be executed.

Curve Service: Delete all Curves (RAM)
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	41h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	41h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	42h	Curve Service: Poll Flash	ro	Unsigned8

Index	Sub-Index	Description	Access Type	Data Type

Read Parameter to get the status of a flash operation:
Result = 00h : State = Idle
Result = 04h : State = Busy

Curve Service: Poll Flash
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	**42h**		**Payload Data**	-	-	-	-
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	**42h**		**Payload Data**	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + CurveID	**50h**	**Curve Service: Add Curve**	**wo**	**Unsigned32**

With this command a curve with the ID “CurveID” will be created. Up to 100 curves can be programmed into the drive.

Curve Service: Add Curve
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	**50h**		**Payload Data**	yyh	yyh	yyh	yyh
			Byte	01	02	03	04

xx xx: Size of the curve info block in bytes, yy yy: Size of the curve data block in bytes

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	**50h**		**Payload Data**	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + CurveID	**53h**	**Curve Service: Add Curve Data (32 Bit)**	**wo**	**Unsigned32**

The Curve Data Block can be written in increments of 4 Bytes at a time. This way one setpoint (32Bit) can be written at a time. This command has to be repeatedly called, with each call containing the next setpoint of the Data Block.

Curve Service: Add Curve Data (32 Bit)

Index	Sub-Index	Description	Access Type	Data Type

SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	53h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Curve data block data: one setpoint as a 32Bit value

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	53h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + CurveID	54h	Curve Service: Add Curve Info Block (32 Bit)	wo	Unsigned32

The Curve Info Block can be written in increments of 4 bytes at a time. To write the info block, this command has to be repeatedly called, with each call containing the next 4 bytes of the info block.

Curve Service: Add Curve Info Block (32 Bit)
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	54h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Curve info block data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	54h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + CurveID	60h	Curve Service: Get Curve	ro	Unsigned32

The “Get Curve” command has to executed first in order to read a curve from the drive via SDO. With the commands “Get Curve Info Block” and “Get Curve Data Block” the corresponding blocks of the curve can be read afterwards.

Curve Service: Get Curve
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	Description	Access Type	Data Type

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	60h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	60h		Payload Data	xxh	yyh	zzh	zzh
			Byte	01	02	03	04

x Re 00h = Curve exists
xsult
: :
D4h = Curve does not exist

yy:Curve Info Block Size in Bytes

zz zz:Curve Data Block Size in Bytes

Index	Sub-Index	Description	Access Type	Data Type
2000h + CurveID	61h	Curve Service: Get Curve Info Block	ro	Unsigned32

The Curve Info Block can be read in increments of 4 Bytes. To read the Info Block, this command has to be repeatedly called, with each call one can read the next 4 Bytes of the Info Block.

Curve Service: Get Curve
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	61h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	61h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Info Block Data

Index	Sub-Index	Description	Access Type	Data Type
2000h + CurveID	62h	Curve Service: Get Curve Data	ro	Unsigned32

The Curve Data Block can be read in increments of 4 Bytes. To read the Data Block, this command has to be repeatedly called, with each call one can read the next 4 Bytes of the Data Block.

Index	Sub-Index	Description	Access Type	Data Type

Curve Service: Get Curve
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	**62h**		**Payload Data**	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + CurveID	**62h**		**Payload Data**	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Info Block Data

5.2.4 Error Log Commands

With these commands the error log of a drive can be read.

Index	Sub-Index	Description	Access Type	Data Type
2000h	70h	Get Error Log Entry Counter	ro	Unsigned32

This command returns the number of logged errors as well as the total number of occurred errors.

Get Error Log Entry Counter
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	70h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	70h		Payload Data	xxh	xxh	yyh	yyh
			Byte	01	02	03	04

xx xx: Number of logged errors
yy yy: Number of occurred errors

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	71h	Get Error Log Entry Error Code	ro	Unsigned32

This command returns the corresponding error code to the entry number.

Get Error Log Entry Error Code
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	71h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	71h		Payload Data	xxh	xxh	yyh	yyh
			Byte	01	02	03	04

xx xx: Source ID: of the part of the firmware which triggered the error:
1: OS
2: Motion Control Software
3: Interface (e.g. POWERLINK)
4: Application (e.g. EasySteps)

yy yy: Error Code: Further Information on the meaning of the error codes can be found in the manuals of the respective firmware parts.

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	72h	Get Error Log Entry Time low	ro	Unsigned32

This command returns the lower 32 bits of the drive's system time when the error has occurred.

Get Error Log Entry Time low
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	72h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	72h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Lower 32 Bits of the system time the error occurred.

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	73h	Get Error Log Entry Time high	ro	Unsigned32

This command returns the higher 32 bits of the drive's system time when the error happened.

Get Error Log Entry Time low
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	73h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	73h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Higher 32 Bits of the system time the error occurred.

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	74h + (Stringlet Nr. 0..7)	Get Error Code Text Stringlet	ro	Unsigned32

This command returns an error stringlet to a the corresponding error code. A stringlet is made up of four ASCII characters. Error code texts can have a maximum of 32 characters.

Get Error Code Text Stringlet
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	Description	Access Type	Data Type

Index	Sub-Index	T		LSB			MSB
2000h	74h + (Stringlet Nr. 0..7)		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	74h + (Stringlet Nr. 0..7)		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Stringlet (each byte represents one character in ASCII format)

5.2.5 Command Table Commands

See the “LinMot 1100 Drive Configuration over Fieldbus Interfaces” for additional detail on the use of the command table and a description of the CT entry format.

Index	Sub-Index	Description	Access Type	Data Type
2000h	80h	CT: Save to Flash	wo	Unsigned8 - Unsigned32

Write any data with this command to save the command table which is in the RAM to the ROM. The command table is loaded on startup of the drive from the ROM to the RAM. Any data can be written for the command to be executed.

CT: Save to Flash
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	80h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	80h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h	80h	CT: Poll Flash	ro	Unsigned8

Read Parameter to get the status of a flash operation:
Result = 00h : State = Idle
Result = 04h : State = Busy
This command can be used to check if a flash operation is still ongoing (e.g. command 2000h sub 80: CT:save to flash)

CT: Poll Flash
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	80h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	80h		Payload Data	xxh	-	-	-
			Byte	01	02	03	04

xx: Result

Index	Sub-Index	Description	Access Type	Data Type
2000h	81h	CT: Delete all Entries (RAM)	wo	Unsigned32

Write anything to delete the complete Command Table in the RAM.

CT: Delete all Entries (RAM)
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h	81h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	81h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	82h	CT: Delete Entry (Entry Nr.)	wo	Unsigned32

Write anything to delete the CT entry with the corresponding number in the RAM. The ROM entry of the CT entry is not deleted this way.

CT: Delete Entry (Entry Nr.)
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	82h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	82h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	83h	CT: Write Entry (Entry Nr.)	wo	Unsigned32

This command has to be executed first if one wants to write write a CT entry to the RAM. This command writes the block size of the CT entry to the RAM. Afterwards the data for the entry can be written with the command “CT: Write Entry Data”.

CT: Write Entry (Entry Nr.)

Index	Sub-Index	Description	Access Type	Data Type

SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	83h		Payload Data	xxh	xxh	-	-
			Byte	01	02	03	04

xx xx: Block size of CT entry

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	83h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	84h	CT: Write Entry Data	wo	Unsigned32

The CT entry data can be written in increments of 4 Bytes. To write the entry data, this command has to be repeatedly called, while each call contains the next 4 bytes of data. The entry will be activated when the last byte of the entry data has been written.

CT: Write Entry Data
SDO Write from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	84h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: CT entry Data

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	84h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	85h	CT: Get Entry (Entry Nr.)	ro	Unsigned32

Read the block size of a CT Entry.

CT: Get Entry (Entry Nr.)
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h +	85h		Payload Data	-	-	-	-

Index	Sub-Index	Description	Access Type	Data Type			
EntryNr			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	85h		Payload Data	xxh	xxh	-	-
			Byte	01	02	03	04

Index	Sub-Index	Description	Access Type	Data Type
2000h + EntryNr	86h	CT: Get Entry (Entry Nr.)	ro	Unsigned32

The CT entry data can be read in increments of 4 Bytes. To read the entry data, this command has to be repeatedly called, while the response to each call contains the next 4 bytes of data.

CT: Get Entry (Entry Nr.)
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	86h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	86h		Payload Data	xxh	xxh	xxh	xxh
			Byte	01	02	03	04

xx xx xx xx: Entry data

Index	Sub-Index	Description	Access Type	Data Type
2000h	87h	CT: Get Entry List (Entry 0..31)	ro	Unsigned32

With this command a bitfield is read, which indicates the presence of a CT entry (0 = CT entry present, 1 = No CT entry present).

CT: Get Entry List (Entry 0..31)
SDO Read from PLC to LinMot Drive:

Index	Sub-Index	T		LSB			MSB
2000h + EntryNr	87h		Payload Data	-	-	-	-
			Byte	01	02	03	04

Return value:

Index	Sub-Index	T		LSB			MSB
2000h	87h		Payload Data	xxh	xxh	xxh	xxh

Index	Sub-Index	Description	Access Type	Data Type			
+ EntryNr			Byte	01	02	03	04

xx xx xx xx: Entry presence bitfield

Index	Sub-Index	Description	Access Type	Data Type
2000h	88h	CT: Get Entry List (Entry 32..63)	ro	Unsigned32
		See command 2000h sub 87h for details.		
2000h	89h	CT: Get Entry List (Entry 64..95)	ro	Unsigned32
		See command 2000h sub 87h for details.		
2000h	8Ah	CT: Get Entry List (Entry 96..127)	ro	Unsigned32
		See command 2000h sub 87h for details.		
2000h	8Bh	CT: Get Entry List (Entry 128..159)	ro	Unsigned32
		See command 2000h sub 87h for details.		
2000h	8Ch	CT: Get Entry List (Entry 160..191)	ro	Unsigned32
		See command 2000h sub 87h for details.		
2000h	8Dh	CT: Get Entry List (Entry 192..223)	ro	Unsigned32
		See command 2000h sub 87h for details.		
2000h	8Eh	CT: Get Entry List (Entry 224..255)	ro	Unsigned32
		See command 2000h sub 87h for details.		

6 CiA402 drive profile (DS402)

Certain LinMot POWERLINK devices support the the CiA402 drive profile (DS402).
The implementation is based on the standards IEC-61800-7-201 and IEC 61800-7-301 .
Detailed Information on the POWERLINK-specific implementation can be found in the EPSG document "Implementation Directive for CiA402 Drive Profile".

If a PLC supports the Integration of drives according to this standard, the LinMot drives can be used as DS402 compatible axes.
Please refer to the manual of the corresponding PLC manufacturer on how to configure and use this functionality.

6.1 Preconditions to use DS402

- only SG6 devices are supported.
- Drive-FW version >= 6.10 b01 installed.
- Use of "00000156_xxxxx_DS402.xdd" as a device description file when configuring the drive in the PLC.

6.2 Supported operating modes

The following operation modes are supported:

Mode 1: Profile Position Mode	(pp)
Mode 3: Profile Velocity Mode	(pv)
Mode 6: Homing Mode	(hm)
Mode 7: Interpolated Position Mode	(ip)
Mode 8: Cyclic synchronous position mode	(csp)
Mode 9: Cyclic synchronous velocity mode	(csv)

6.3 Supported objects

For a detailed explanation of all supported objects, please refer to the standard IEC 61800-7-201.

The following objects are supported:

Index	Sub-Index	Name	PDO mappable
6040h		ControlWord	X
6041h		StatusWord	X
6060h		Modes of Operation	X
6061h		Modes of Operation Display	X
6064h		Position ActualValue	X
606Ch		Velocity ActualValue	X
607Ah		Target Position	X
6081h		Profile Velocity	X
6083h		Profile Acceleration	X
6084h		Profile Deceleration	X
60C1h	01h	InterpolationData Record 1stSetPoint	X
60FFh		Target Velocity	X
6007h		AbortConnectionOptionCode	
603Fh		ErrorCode[1]	X

Index	Sub-Index	Name	PDO mappable
605Ah		QuickStopOptionCode	
605Bh		ShutdownOptionCode	
605Ch		DisableOperationOptionCode	
605Dh		HaltOptionCode	
605Eh		FaultReactionOptionCode	
6062h		PositionDemandValue	X
6075h		Motor Rated Current (Rated Current is 20% of maximum current)	
6076h		Motor Rated Torque (Rated Torque is 20% of maximum torque)	
6077h		TorqueActualValue	X
607Bh	01h	MinPositionRangeLimit	
607Bh	02h	MaxPositionRangeLimit	
607Ch		HomeOffset	
607Dh	01h	MinPositionLimit	
607Dh	02h	MaxPositionLimit	
607Eh		Polarity	
6072h		Bipolar Torque/Force Limit (%-scaling w ith LSB = 0.1% in relation to Motor rated current)	X
607Fh		MaxProfileVelocity	
6085h		QuickStopDeceleration	
608Fh	01h	PositionEncoderResolution_EncoderIncrements	
608Fh	02h	PositionEncoderResolution_MotorRevolutions	
6090h	01h	VelocityEncoderResolution_EncoderIncrementsPerSecond	
6090h	02h	VelocityEncoderResolution_MotorRevolutionsPerSecond	
6091h	01h	GearRatio_MotorShaftRevolutions	
6091h	02h	GearRatio_DrivingShaftRevolutions	
6092h	01h	FeedConstant_Feed	
6092h	02h	FeedConstant_ShaftRevolutions	
6096h	01h	VelocityFactor_Numerator	
6096h	02h	VelocityFactor_Divisor	
6097h	01h	AccelerationFactor_Numerator	
6097h	02h	AccelerationFactor_Divisor	
6098h		HomingMethod	
6099h	01h	HomingSpeed_SpeedDuringSearchForSwitch	
6099h	02h	HomingSpeed_SpeedDuringSearchForZero	
609Ah		HomingAcceleration	
60A2h	01h	JerkFactor_Numerator	
60A2h	02h	JerkFactor_Divisor	
60B0h		PositionOffset	

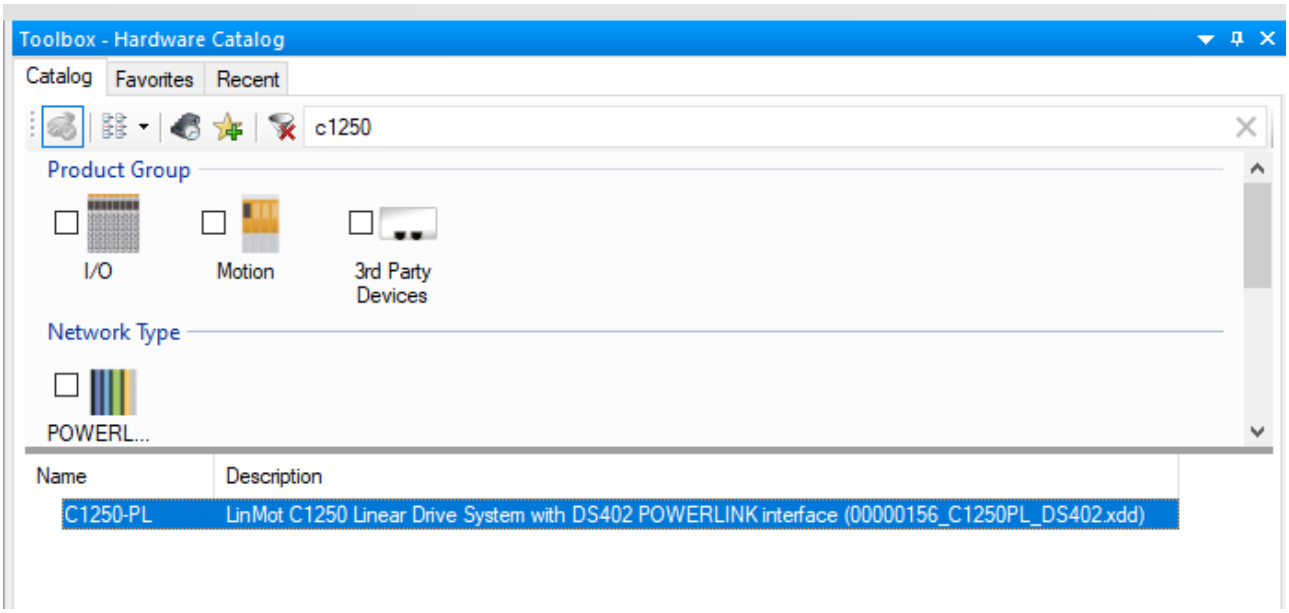
Index	Sub-Index	Name	PDO mappable
60B1h		VelocityOffset	
60B2h		TorqueOffset	
60C0h		InterpolationSubModeSelect	
60C2h	01h	InterpolationTimePeriod_IntepolationTimePeriodValue	
60C2h	02h	InterpolationTimePeriod_InterpolationTimeIndex	
60A4h	01h	InterpolationDataConfiguration_MaximumBufferSize	
60A4h	02h	InterpolationDataConfiguration_ActualBufferSize	
60C5h		MaxAcceleration	
60C6h		MaxDeceleration	
60E0h		Positive Torque/Force Limit (%-scaling with LSB = 0.1% in relation to Motor rated current)	X
60E1h		Negative Torque/Force Limit (%-scaling with LSB = 0.1% in relation to Motor rated current)	X
60E3h	01h	SupportedHomingMethods_1stSupportedHomingMethod	
60F2h		PositioningOptionCode	
6402h		MotorType	
6502h		SupportedDriveModes	

[1] The Error code is comprised as follows: (0x1000 (Generic Error) + 0x80 (Manufacturer Specific) + Logged Error Code (UPID 1D96)). E.g. the LinMot error "0003h: Err: X1 Pwr Voltage Too Low" will have the value 0x1083 in the object 603Fh.

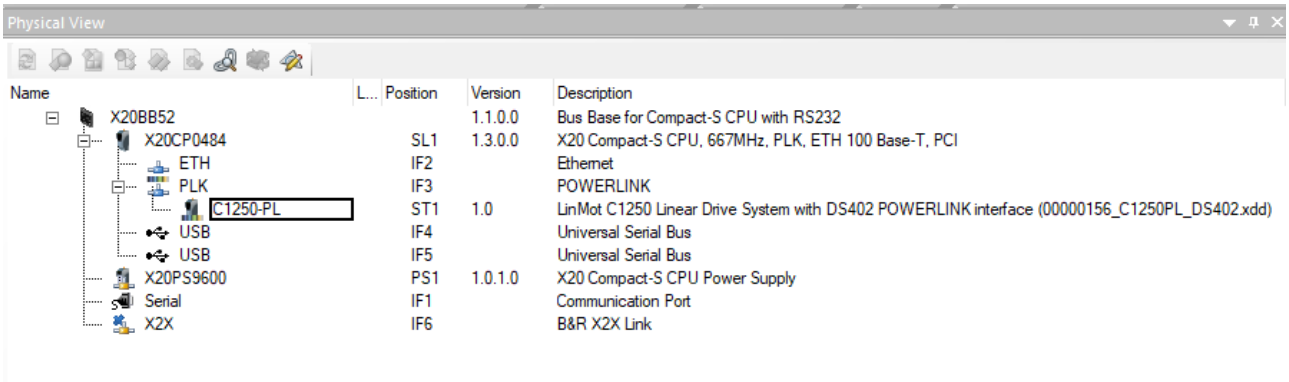
6.4 Commissioning with a B&R PLC

6.4.1 Configuring the LinMot drive as a DS402 axis

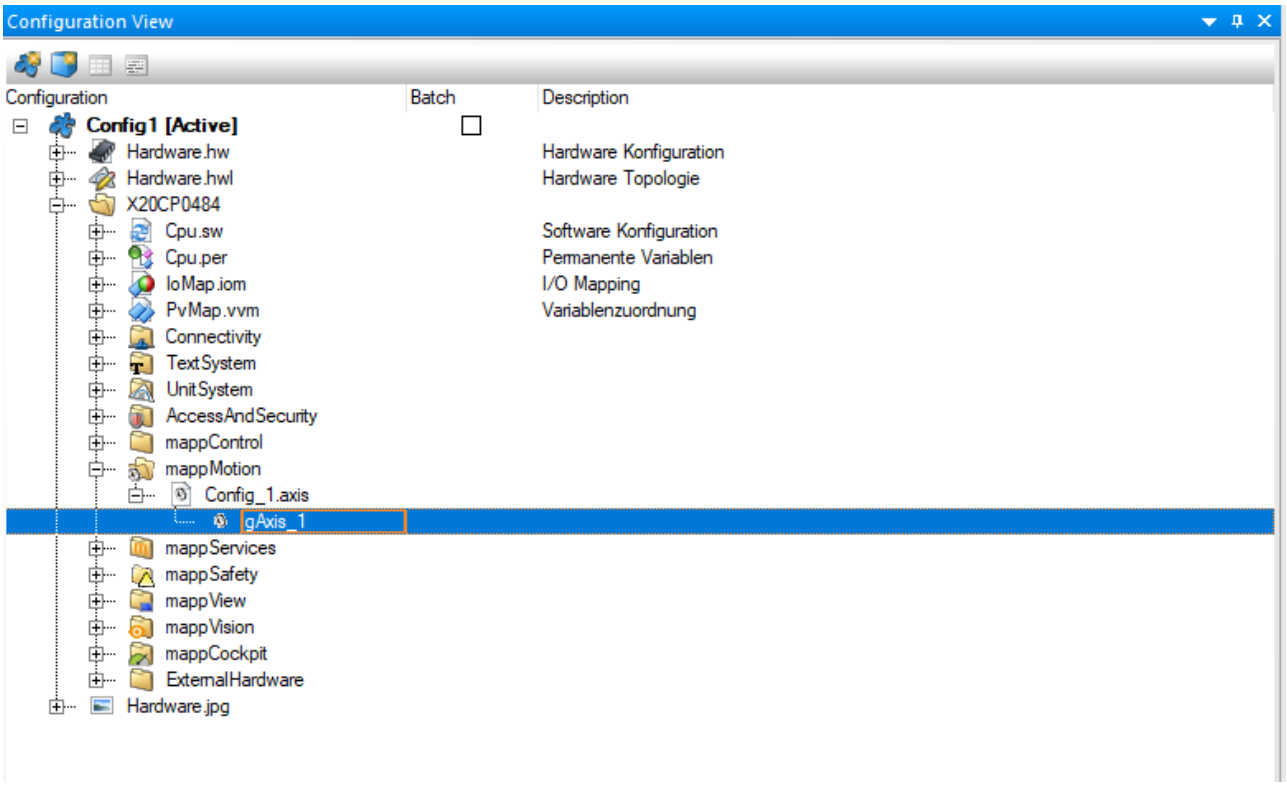
1. Import the device description file "00000156_xxxxxxx_DS402.xdd" in Automation Studio:



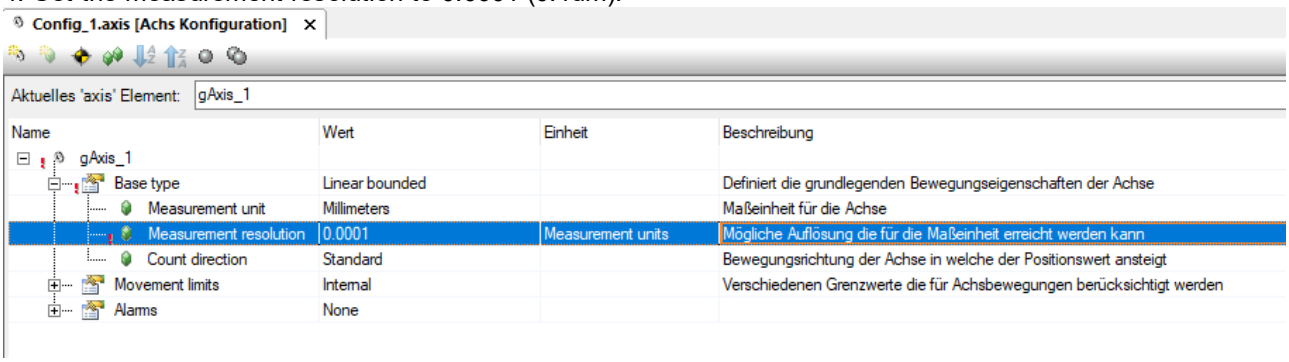
2. Add the device to the HW configuration:



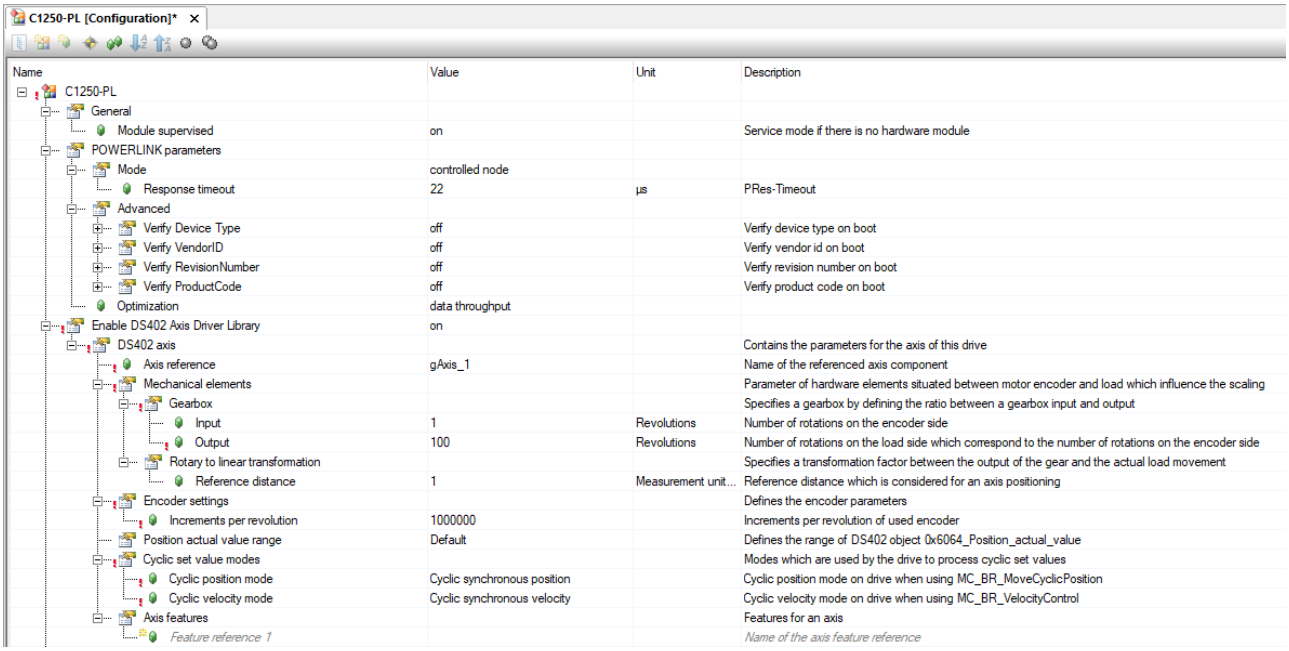
3. Add an axis object to the mappMotion config:



4. Set the measurement resolution to 0.0001 (0.1um):



5. Configure the device and enable the use of the DS402 Axis Driver Library and set the axis reference:



If a LinMot linear motor is used, set the Increments per revolution to 1'000'000 and the Gearbox output to 100.

6.4.2 Homing

Only manufacturer specific homing modes are supported.

When using the MC_BR_InitHome_DS402Ax function block, set "DriveSpecificHoming" to "mcSWITCH_ON". "DriveSpecificHomingModeMode" configures the HomingMode like the UPID 13C4h does:

Name	Value	Raw Data
Actual Position	Off	0000h
Mechanical Stop Negative Search	On	0001h
Mechanical Stop Positive Search	Off	0002h
Limit Switch Negative	Off	000Eh
Limit Switch Positive	Off	000Fh
Home Switch Negative	Off	0003h
Home Switch Positive	Off	0010h
Home Switch And Mechanical Stop Negative	Off	0004h
Home Switch And Mechanical Stop Positive	Off	0005h
Home Switch And Limit Switch Negative	Off	0006h
Home Switch And Limit Switch Positive	Off	0007h
Index On X13 And Mechanical Stop Negative	Off	0008h
Index On X13 And Mechanical Stop Positive	Off	0009h
Index On X13 And Limit Switch Negative	Off	000Ah
Index On X13 And Limit Switch Positive	Off	000Bh
Index On X13 Negative Search	Off	000Ch
Index On X13 Positive Search	Off	000Dh
No Drive Homing	Off	0011h

Position offset velocity and acceleration should also be configured to the users needs.

6.4.3 Configuring Modulo axes

A LinMot EC02 motor can be configured as a modulo axis with the following steps:

1. PLC configuration

Name	Wert	Einheit	Beschreibung
gAxis_1			
Base type	Rotary periodic		Defines the basic movement possibilities of the axis
Measurement unit	Degrees		Measurement unit for the axis
Measurement resolution	0.0006866455078125	Measurement units	Possible resolution of measurement unit that can be achieved
Count direction	Standard		Direction of the axis in which the position value is increasing
Period settings			Possible position value range of a periodic axis
Period	360.0	Measurement units	The value range for axis positions is [0 , Period[
Movement limits	Internal		Various limit values that will be considered for axis movements
Alarms	None		

Axis object configuration in the PLC for an EC02 as a modulo axis

Verwendung der DS402 Achstreiber-Bibliothek	Ein		
DS402 axis			Contains the parameters for the axis of this drive
Axis reference	gAxis_1		Name of the referenced axis component
Mechanical elements			Parameter of hardware elements situated between motor encoder and load which influence the scaling
Gearbox			Specifies a gearbox by defining the ratio between a gearbox input and output
Input	1	Revolutions	Number of rotations on the encoder side
Output	1	Revolutions	Number of rotations on the load side which correspond to the number of rotations on the encoder side
Rotary to linear transformation			Specifies a transformation factor between the output of the gear and the actual load movement
Reference distance	1	Measurement...	Reference distance which is considered for an axis positioning
Encoder settings			Defines the encoder parameters
Increments per revolution	524288		Increments per revolution of used encoder
Position actual value range	Default		Defines the range of DS402 object 0x6064_Position_actual_value
Cyclic set value modes			Modes which are used by the drive to process cyclic set values
Cyclic position mode	Interpolated position		Cyclic position mode on drive when using MC_BR_MoveCyclicPosition
Cyclic velocity mode	Profile velocity		Cyclic velocity mode on drive when using MC_BR_VelocityControl
Axis features			Features for an axis
Feature reference 1			Name of the axis feature reference
Feature reference 2			*Name of the axis feature reference*

Drive configuration in the PLC for an EC02 as a modulo axis

2. LinMot Talk configuration

   Modulo Mode has to be set to "On" (UPID 0x11C5)

3. Setting the Positioning Mode

   Possible movements:
   - normal positioning
   - positioning only in negative direction
   - positioning only in positive direction
   - positioning with the shortest way

   In order to set the desired positioning mode object 0x60F2 (PositioningOptionCode) has to be written by the PLC.

   Bits 6 and 7 are used to configure the desired mode:

Bit 7	Bit 6	Definition
0	0	Normal positioning similar to linear axis If reaching or exceeding the position range limits (607B) the input value shall wrap automatically to the other end of the range
0	1	Positioning only in negative direction; if target position is higher than actual position, axis moves over "Min position limit" to target position
1	0	Positioning only in positive direction; if target position is lower than actual position, axis moves over "Max position limit" to target position
1	1	Positioning with the shortest way to the target position. Special condition: If the difference between actual value and target position in a 360° system is 180°, the axis will move in positive direction.

Option Bits 6 and 7 of object 0x60F2 (PositioningOptionCode)

7 RT LEDS

C1250-PL

Error Codes		
RT BUS ERROR  OK		
OK	**RT Bus Error**	**Description**
Off	-	NMT State is „NMT_CS_Not_Active“
On	-	NMT State is „NMT_CS_Operational“
Flickering	-	NMT State is „NMT_CS_BasicEthernet“
Blinking	-	NMT State is „NMT_CS_Stopped“
Single Flash	-	NMT State is „NMT_CS_PreOperational_1“
Double Flash	-	NMT State is „NMT_CS_PreOperational_2“
Triple Flash	-	NMT State is „NMT_CS_ReadyToOperate“

C1250-MI

Error Codes		
L3 / BS (Bus status)	L3  L4	
Color	**State**	**Description**
 (green)	On	Slave is in **‘Operational’** state
 (green)	Triple flash	Slave is in **‘ReadyToOperate’** state
	Double flash	Slave is in **‘Pre-Operational 2’** state
	Single flash	Slave is in **‘Pre-Operational 1’** state
	Flickering (10 Hz)	Slave is in **‘Basic Ethernet’** state
	Blinking (2.5 Hz)	Slave is in **‘Stopped’** state
 (off)	Off	Slave initializing
L4 / BE (Bus error)		

Error Codes		
(off)	Off	Slave has no error
(red)	On	Slave has detected an error

Off: constantly off.
On: constantly on.
Flickering: equal on and off times with a frequency of approximately 10 Hz.
Blinking: equal on and off times with a frequency of approximately 2,5 Hz.
Single Flash: one short flash followed by a long off phase.
Double Flash: a sequence of two short flashes, separated by an off phase . The sequence is finished by a long off phase.
Triple Flash: a sequence of three short flashes, separated by an off phase . The sequence is finished by a long off phase.

8 INTERFACE ERROR CODES

Please refer to “Usermanual Motion Control Software” for the error codes of the MC software. The Powerlink interface has the following additional error codes:

Error Code	Error Description	Recommended Actions
C0h	Cfg Err: Invalid NodeID	The defined NodeID with S1 & S2 is not in the valid range of 0..239

9 CABLING

This chapter describes the possibilities and some considerations to take into account when wiring a POWERLINK network.

9.1 TOPOLOGY RECOMMENDATIONS

There are two basic wiring Topologies that can be used to build a POWERLINK network:

- Tree Structure

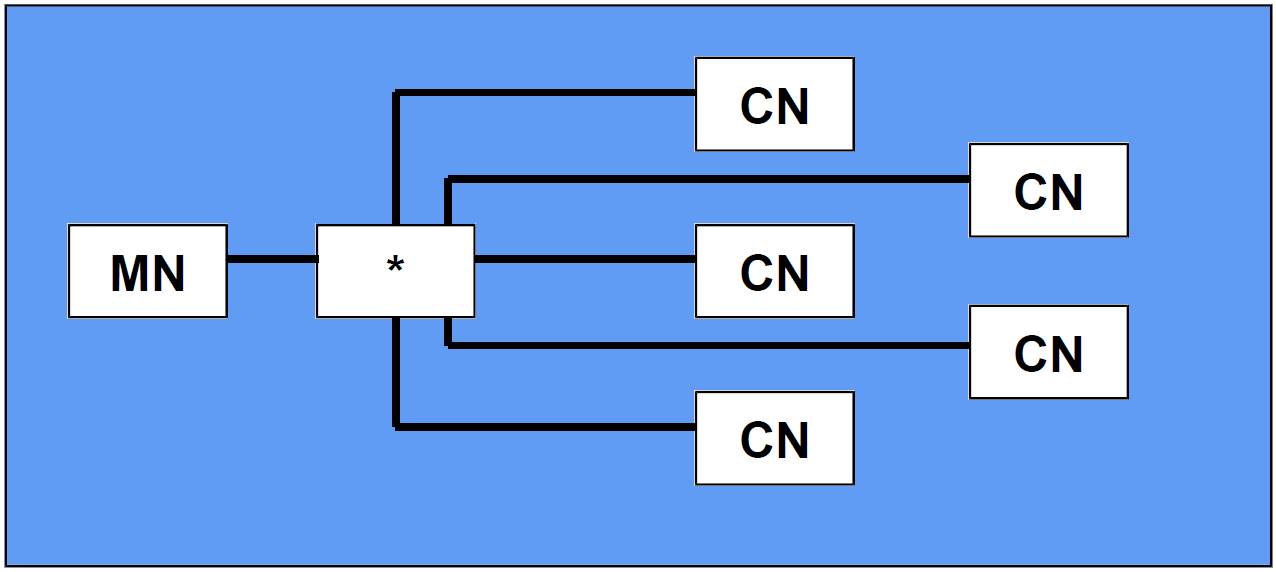


Figure 1: POWERLINK network organised as a tree structure

- Line Structure (Daisy Chain)

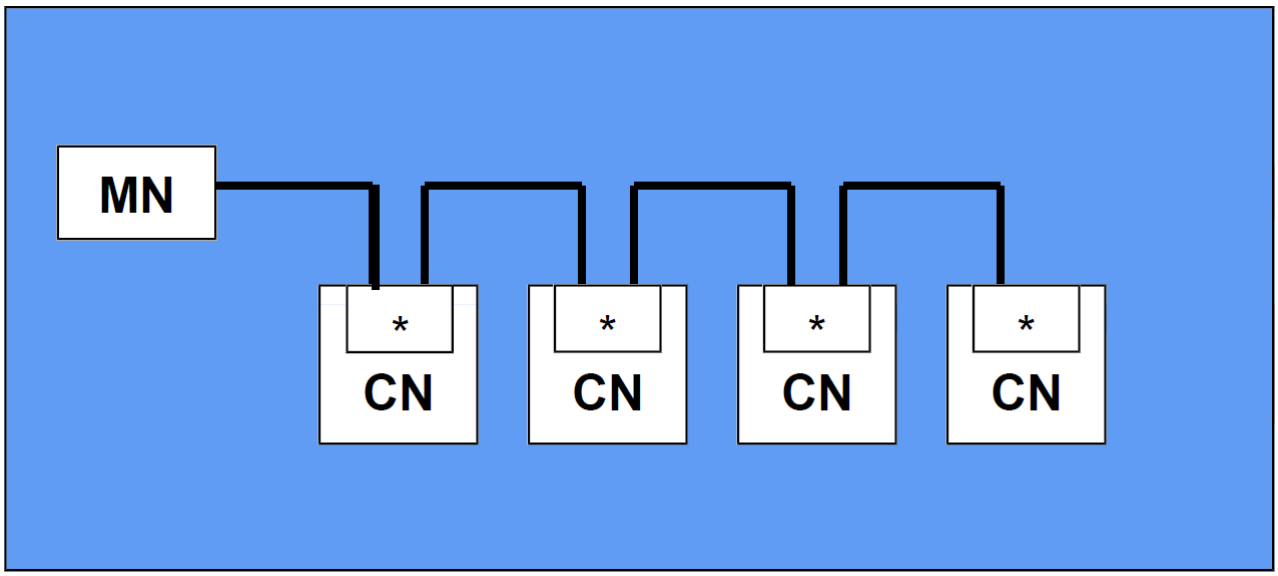


Figure 2: POWERLINK network organised as a line structure

Since every hub/switch introduces a certain delay, the hub/switch depth should be kept to a minimum. This is done by using external hubs/switches to implement a tree structure instead of daisy-chaining elements in a line.

To minimize the POWERLINK cycle time, it is recommended to use a tree structure whenever possible

In many cases a mixture of both structures is the most suitable solution, since sub-systems which functionally can be seen as one unit can be wired as line structures and joined together with a hub/switch.Mixed structure:

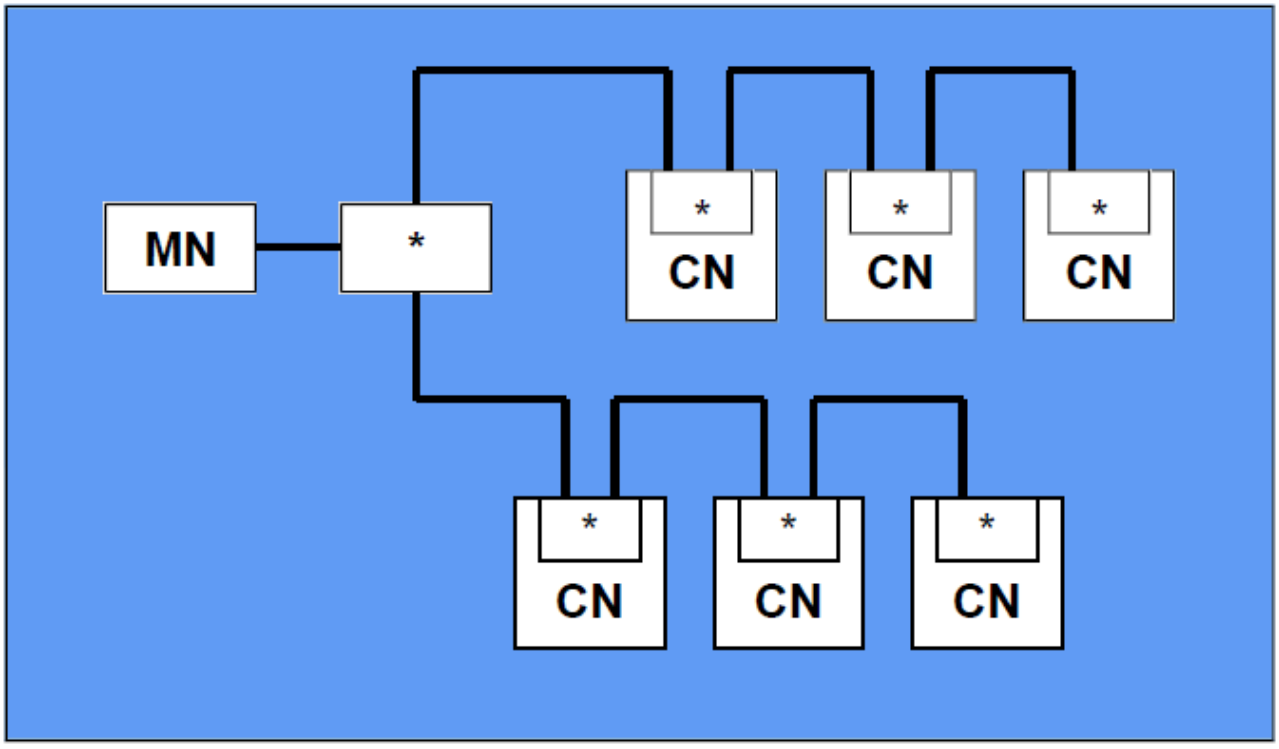


Figure 3: POWERLINK network organised as a line strucure

9.2 CALCULATING THE MINIMAL POLL RESPONSE TIME (TPRES) OF AN ARBITRARY CN

To estimate the poll response time of a specific slave one can use the following formula:

$$T_{PRes} = 2 \times \left(T_C + T_H \right) + T_{PReq\_PRes}$$

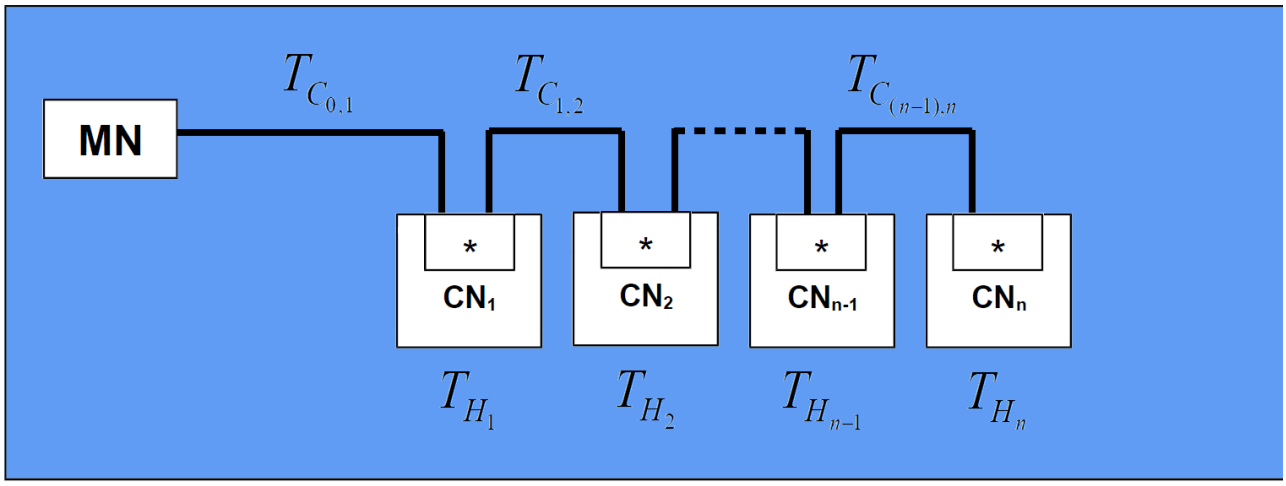
T_{PRes}	: Poll Response Time	The total time it takes for a Poll Request message from the time it is being sent from the MN until the MN receives the response from the CN.
T_C	: Cable Runtime	The signal runtime of the cabling causes delays. For each meter of cable a delay of approximately 5ns should be taken into consideration.
T_H	: Runtime for all infrastructure elements	This is the sum of all delays of all elements (Hubs, Routers, etc.) between the MN and the CN. If daisy-chaining is used to connect several CN's, the sum of the delays of the CN's in between the MN and the CN one is interested in comprise T_H.
T_{PReq_PRes}	: Response time of the CN	Time it takes the CN to process the poll request and to output its response.

9.3 MINIMAL POWERLINK CYCLE TIME FOR A PURE LINE STRUCTURE

Minimal cycle time if the network is laid out as a pure line structure:

$$T_{EPLCyc_{\min}} = \sum_{n=1}^{\#_{CN}} \left(2 \cdot \sum_{m=1}^{n} T_{C_{(m-1),m}} + T_{PReq\_PRes_n} + \sum_{m=1}^{n} T_{H_m} \right)$$

$T_{EPLCyc_{\min}}$	: Minimal POWERLINK cycle time	Minimal POWERLINK cyle time that is feasible
T_{H_m}	: Runtime for the m-th CN	This is the delay of the m-th specific CN caused by its hub/switch.
$\#_{CN}$	: Total number of CN's in line	Total number of slaves connected to each other via daisy-chaining
$T_{C_{(m-1),m}}$	: Cable Runtime for the cable from CN (m-1) to CN m.	The signal runtime of the cabling causes delays. For each meter of cable a delay of approximately 5ns should be taken into consideration.
$T_{PReq\_PRes_n}$	: Response time of the n-th CN	Time it takes the n-th CN to process the poll request and to output its response.



10 TROUBLESHOOTING

10.1 ANALYZING TRAFFIC IN POWERLINK NETWORKS

To analyze the data traffic in a POWERLINK network the use of a network protocol analyzer is strongly recommended.
Wireshark is one of the most used analyzers and can be downloaded free of charge from http://www.wireshark.org.

Attention: When using a network interface from a personal computer, make sure that any other protocols such as TCP/IP etc. are disabled for this interface. Transmission of any unwanted data frames from the personal computers operating system may lead to unpredictable behavior and/or errors in a connected POWERLINK node.

10.2 FREQUENT PROBLEMS AND SOLUTIONS

Problem:	**The node never gets to the NMT-State Operational**
Possible Solution:	Check if the Node-ID of the drive matches with the one configured in the PLC
Possible Solution:	Check if the Node-ID of the drive is unique, and no other node in the network has the same ID.

Problem:	**Warnings appear when importing an xdd-File in certain versions of commissioning tools**
	Mess... 20.09.2021 12:33:27.9281 Checking XDD schema compliance using version v0.16... Mess... 20.09.2021 12:33:27.9437 Checking Powerlink specification 'EPSG DS301 v1.3.0' compliance using XDD-Check v1.3.3... Warni... 20.09.2021 12:33:27.9437 Object 1300h: "defaultValue" value ('5000') does not match the defaultValue value in the POWERLINK-Specification. Warni... 20.09.2021 12:33:27.9437 Object 1C14h: "defaultValue" value ('300000') does not match the defaultValue value in the POWERLINK-Specification. Warni... 20.09.2021 12:33:27.9437 Object 1F8Ch: "PDOMapping" value ('optional') does not match the PDO-mapping value in the POWERLINK-Specification. Mess... 20.09.2021 12:33:27.9437 Powerlink specification compliance check finished with warnings... Mess... 20.09.2021 12:33:28.0218 File '00000156_C1250PL.xdd' imported with validation issues.
Possible Solution:	The warnings can be ignored. The default values are different from the default value in the specification due to compatibility reasons with different systems.

Contact & Support

SWITZERLAND

NTI AG
Bodenaeckerstrasse 2
CH-8957 Spreitenbach

Sales and Administration:	+41 56 419 91 91 office@linmot.com
Tech. Support:	+41 56 544 71 00 support@linmot.com http://www.linmot.com/support
Tech. Support (Skype):	support.linmot
Fax:	+41 56 419 91 92
Web:	http://www.linmot.com

USA

LinMot USA Inc.
N1922 State Road 120, Unit 1
Lake Geneva, WI 53147
USA

Phone:	262-743-2555
E-Mail:	usasales@linmot.com
Web:	http://www.linmot-usa.com/

Please visit http://www.linmot.com/contact to find the distribution close to you.

Smart solutions are...