



ETHERNET POWERLINK

**Documentation of the Ethernet POWERLINK Interface for
the E1200 Series**

Ethernet POWERLINK Interface for E1200 1.0 User Manual

This document applies to the following controllers:
E1250-PL-xx
(with POWERLINK Interface SW installed)

© 2009 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.

Document version 1.0 / mk, December 2009

Inhaltsverzeichnis

1SYSTEM OVERVIEW.....	4
2CONNECTING TO THE POWERLINK NETWORK.....	5
2.1PIN ASSIGNMENT OF THE CONNECTORS X17-X18.....	5
2.2SETTING THE NODEID.....	5
3POWERLINK PARAMETERS AND VARIABLES.....	6
3.1PARAMETERS	6
3.2VARIABLES	7
4MAPPING OF PRES AND PREQ.....	8
4.1PDO MAPPING.....	8
4.2ADDITIONAL PARAMETERS.....	8
4.3MAPPING OF PARAMETERS VIA THEIR UPID.....	9
4.3.1Example how to read a parameter via its UPID.....	9
5LINMOT PARAMETER COMMANDS.....	10
5.1UPID COMMANDS.....	11
5.2SYSTEM COMMANDS.....	16
5.3CURVE SERVICE COMMANDS.....	20
5.4ERROR LOG COMMANDS.....	26
5.5COMMAND TABLE COMMANDS.....	30
6CABLING.....	38
6.1TOPOLOGY RECOMMENDATIONS.....	38
6.2CALCULATING THE MINIMAL POLL RESPONSE TIME (TPRES) OF AN ARBITRARY CN.....	39
6.3MINIMAL POWERLINK CYCLE TIME FOR A PURE LINE STRUCTURE.....	40
7CONTACT ADDRESSES.....	41

1 System overview

The LinMot POWERLINK controllers E1250-PL-UC 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 μ s
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)	< 50 μ s
SoA to ASnd latency (CN asynchronous reaction time)	< 50 μ s
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

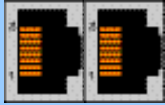
For further information on Ethernet POWERLINK please visit:

<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		POWERLINK 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".	

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	
	S1	Bus ID High (0h...Fh)
	S2	Bus ID Low (0h...Fh)

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-Talk1100 software. With these parameters, the POWERLINK behaviour can be configured.

The LinMot-Talk1100 software can be downloaded from <http://www.linmot.com> under the section download, software & manuals.

POWERLINK\ Dis-/Enable

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

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



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 E1250 controller.

POWERLINK\ 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).

POWERLINK\ 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.

POWERLINK\ CN Cross Traffic monitoring

In this section the parameters for cross traffic monitoring of other POWERLINK CN's can be configured. Two words (32Bit each) can be configured separately for monitoring. This way data can be taken directly from the PRes of another CN without the delay of a cycle if this data is i.e. fed back to the controller via a PLC.

POWERLINK\ CN Cross Traffic monitoring	
Generic CN 01	Parameters for cross traffic monitoring of a word of the PRes of an arbitrary CN can be configured here
Generic CN 02	Parameters for cross traffic monitoring of a word of the PRes of an arbitrary CN can be configured here

POWERLINK\ CN Cross Traffic monitoring\ Generic CN 01

In this section the NodeID of the monitored POWERLINK CN 01 and its data offset can be configured.

POWERLINK\ CN Cross Traffic monitoring\ Generic CN 01	
CN Node-ID	Node-ID of the CN from which data is monitored
PRes offset	Offset in bytes from the start of the sources PRes from which a word (32 Bit) is read

POWERLINK\ CN Cross Traffic monitoring\ Generic CN 02

In this section the NodeID of the monitored POWERLINK CN 02 and its data offset can be configured.

POWERLINK\ CN Cross Traffic monitoring\ Generic CN 02	
CN Node-ID	Node-ID of the CN from which data is monitored
PRes offset	Offset in bytes from the start of the sources PRes from which a word (32 Bit) is read

3.2 Variables

The POWERLINK Interface has an additional parameter tree branch (Parameters → POWERLINK), which can be configured with the distributed LinMot-Talk1100 software.

Name	Type	Definition
Node-ID	UInt16	Current Node-ID
NMT State	UInt16 Enumerator	Current NMT State
Generic CN 01 cross-traffic monitored PRes value	UInt32	Value of the configured word for cross-traffic monitoring of CN 01
Generic CN 02 cross-traffic monitored PRes value	UInt32	Value of the configured word for cross-traffic monitoring of CN 02

4 Mapping of PRes and PReq

The LinMot Controller 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\LinTalk1100 3.x\Firmware\POWERLINK\XDD)

4.1 PDO Mapping

Index	Sub-Index	Name	Object Type	Access Type	PDO Mapping
4F00h	-	MCSW Variables	RECORD	const	-
	01h	MCSW_StateVar	UInt32	ro	optional
	02h	MCSW_StatusWord	UInt32	ro	optional
	03h	MCSW_WarnWord	UInt32	ro	optional
	04h	MCSW_ActualPosition32Bit	Int32	ro	optional
	05h	MCSW_DemandPosition32Bit	Int32	ro	optional
	06h	MCSW_DemandCurrent	Int32	ro	optional
	07h	MCSW_ControlWord	UInt32	wo	optional
	08h	MCSW_MotionCommandHeader	UInt32	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
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

4.2 Additional Parameters

These Parameters can be set to configure additional services on the E1250.

They are written once during the initialising phase by the PLC to the controller.

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 controller by their UPIDs.

Any parameter of the controller 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 controller.

Name	Value	Description
IF3.ST6		FBE.00000156_E1250-PL-UC
General		
Module supervised	on	Service mode if there is no hardware module
Powerlink parameters		
Mode	controlled node	
Response timeout [us]	100	
Multiplexed station	off	
Advanced		
Channels		Objects for cyclic transmission
MCSW_Variables_I4F00 RECORD[0x0F]		
UPID_Variables_Values_I4F02 RECORD[0x04]		
Variable_by_UPID_01_I4F02_S01		
Cyclic transmission	Read	
Datatype	UDINT	UNSIGNED32
Init value		Set at bootup (clear to preserve value on device)
Variable_by_UPID_02_I4F02_S02		
Variable_by_UPID_03_I4F02_S03		
Variable_by_UPID_04_I4F02_S04		
Device specific parameters		Transmitted to the device at startup
UPID_Variables_UPIDs_I4F01 RECORD[0x04]		
UPID_01_I4F01_S01		
Datatype	UINT	UNSIGNED16
Init value	16#1BBC	Set at bootup (clear to preserve value on device)
UPID_02_I4F01_S02		
UPID_03_I4F01_S03		
UPID_04_I4F01_S04		

Settings for UPID mapping in the B&R Automation Studio

5 LinMot Parameter Commands

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

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 Controller	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
	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 1100 Servo Controller Configuration over Fieldbus Interfaces*”.

5.1 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 Controller:

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

Return value:

Index	Sub-Index	LSB	MSB
2000h + UPID	01h	Payload Data	Payload Data
		xxh	xxh
		Byte	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 Controller:

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

xx xx xx xx: Value of the UPID to write

Return value:

Index	Sub-Index	LSB	MSB
2000h + UPID	01h	Payload Data	Payload Data
		-	-
		Byte	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 controller. Use the "RAM/ROM Write" command (sub-index 06h) to changes both values at the same time.

Read ROM Value by UPID

SDO Read from PLC to LinMot Controller:

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

Return value:

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

xx xx xx xx: Read Value of the UPID

Write ROM Value by UPID

SDO Write from PLC to LinMot Controller:

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

xx xx xx xx: Value of the UPID to write

Return value:

Index	Sub-Index	LSB	MSB
2000h + UPID	02h	Payload Data	Payload Data
		-	-
		Byte	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 Max Value by UPID
SDO Read from PLC to LinMot Controller:

Index	Sub-Index
2000h + UPID	04h

	LSB				MSB			
Payload Data	-	-	-	-				
Byte	01	02	03	04				

Return value:

Index	Sub-Index
2000h + UPID	04h

	LSB				MSB			
Payload Data	xxh	xxh	xxh	xxh				
Byte	01	02	03	04				

xx xx xx xx: Max Value of the UPID

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 Controller:

Index	Sub-Index
2000h + UPID	05h

	LSB				MSB			
Payload Data	-	-	-	-				
Byte	01	02	03	04				

Return value:

Index	Sub-Index
2000h + UPID	05h

	LSB				MSB			
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 Controller:

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

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.

Start getting UPID List
SDO Write from PLC to LinMot Controller:

Index	Sub-Index	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	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																																																																																																																	
<p>With this command information on UPIDs can be read. After the initialisation with the command “Start getting UPID List”, information on UIPIDs can be read with this command. The command can be repeatedly issued. With each new command, the information on the next existing UPID is sent. When the end of the list is reached, the UPID FFFFh is sent.</p>																																																																																																																					
<p>Get Next UPID List item SDO Read from PLC to LinMot Controller:</p>																																																																																																																					
<table><tr><th>Index</th><th>Sub-Index</th></tr><tr><td>2000h + UPID</td><td>21h</td></tr></table>	Index	Sub-Index	2000h + UPID	21h	<table><tr><th>LSB</th><th colspan="3">MSB</th></tr><tr><td>Payload Data</td><td>-</td><td>-</td><td>-</td></tr><tr><td>Byte</td><td>01</td><td>02</td><td>03</td></tr></table>	LSB	MSB			Payload Data	-	-	-	Byte	01	02	03																																																																																																				
Index	Sub-Index																																																																																																																				
2000h + UPID	21h																																																																																																																				
LSB	MSB																																																																																																																				
Payload Data	-	-	-																																																																																																																		
Byte	01	02	03																																																																																																																		
<p>Return value:</p>																																																																																																																					
<table><tr><th>Index</th><th>Sub-Index</th></tr><tr><td>2000h + UPID</td><td>21h</td></tr></table>	Index	Sub-Index	2000h + UPID	21h	<table><tr><th>LSB</th><th colspan="3">MSB</th></tr><tr><td>Payload Data</td><td>yyh</td><td>yyh</td><td>xxh</td></tr><tr><td>Byte</td><td>01</td><td>02</td><td>03</td></tr></table>	LSB	MSB			Payload Data	yyh	yyh	xxh	Byte	01	02	03																																																																																																				
Index	Sub-Index																																																																																																																				
2000h + UPID	21h																																																																																																																				
LSB	MSB																																																																																																																				
Payload Data	yyh	yyh	xxh																																																																																																																		
Byte	01	02	03																																																																																																																		
<p>yy yy: Address Usage xx xx: UPID which was found</p>																																																																																																																					
<p>Address Usage is interpreted as follows:</p>																																																																																																																					
<table><tr><td colspan="16">This Parameter is used for security Features. Parameter is included when calculating the hash table.</td></tr><tr><td colspan="16">Live Parameter</td></tr><tr><td colspan="16">ROM Write allowed</td></tr><tr><td colspan="16">ROM Read allowed</td></tr><tr><td colspan="16">RAM Write allowed</td></tr><tr><td colspan="16">RAM Read allowed</td></tr><tr><td>Bit Nr.:</td><td>0 (LSB)</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15 (MSB)</td></tr></table>					This Parameter is used for security Features. Parameter is included when calculating the hash table.																Live Parameter																ROM Write allowed																ROM Read allowed																RAM Write allowed																RAM Read allowed																Bit Nr.:	0 (LSB)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 (MSB)
This Parameter is used for security Features. Parameter is included when calculating the hash table.																																																																																																																					
Live Parameter																																																																																																																					
ROM Write allowed																																																																																																																					
ROM Read allowed																																																																																																																					
RAM Write allowed																																																																																																																					
RAM Read allowed																																																																																																																					
Bit Nr.:	0 (LSB)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 (MSB)																																																																																																					
2000h	22h	Start Getting Modified UPID List	wo	Unsigned16																																																																																																																	
<p>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.</p>																																																																																																																					

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 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	Set ROM to default (OS)	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 Write from PLC to LinMot Controller:

Index	Sub-Index
2000h	07h

	LSB		MSB	
Payload Data	xxh	xxh	xxh	xxh
Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index
2000h	07h

	LSB		MSB	
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 Controller:

Index	Sub-Index
2000h	08h

	LSB			MSB
Payload Data	xxh	xxh	xxh	xxh
Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index
2000h	08h

	LSB			MSB
Payload Data	-	-	-	-
Byte	01	02	03	04

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 Controller:

Index	Sub-Index
2000h	09h

	LSB			MSB
Payload Data	xxh	xxh	xxh	xxh
Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index
2000h	09h

	LSB			MSB
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 Controller:

Index	Sub-Index
2000h	0Ah

	LSB		MSB	
Payload Data	xxh	xxh	xxh	xxh
Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index
2000h	0Ah

	LSB		MSB	
Payload Data	-	-	-	-
Byte	01	02	03	04

2000h	0Bh	Reset Controller	wo	Unsigned8 - Unsigned32
-------	-----	------------------	----	------------------------

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

Reset Controller
SDO Write from PLC to LinMot Controller:

Index	Sub-Index
2000h	0Bh

	LSB		MSB	
Payload Data	xxh	xxh	xxh	xxh
Byte	01	02	03	04

xx xx xx xx: Any Data

Return value:

Index	Sub-Index
2000h	0Bh

	LSB		MSB	
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 Controller:

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

2000h	36h	Start MC and APPL SW	wo	Unsigned8 - Unsigned32
-------	-----	----------------------	----	------------------------

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

Start MC and APPL Software
SDO Write from PLC to LinMot Controller:

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

5.3 Curve Service Commands

See the “LinMot 1100 Servo Controller 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 Controller:

Index	Sub-Index	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	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 Controller:

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

xx xx xx xx: Any Data

Return value:

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

2000h	42h	Curve Service: Poll Flash	ro	Unsigned8
-------	-----	---------------------------	----	-----------

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 Controller:

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

Return value:

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

xx: Result

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 controller.

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

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

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)
SDO Write from PLC to LinMot Controller:

Index	Sub-Index	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	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 Controller:

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

xx xx xx xx: Curve info block data

Return value:

Index	Sub-Index	LSB	MSB
2000h + CurveID	53h	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 be executed first in order to read a curve from the controller 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 Controller:

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

Return value:

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

xx: Result: 00h = Curve exists
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 Controller:

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

Return value:

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

xx xx xx xx: Info Block Data

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.

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

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

Return value:

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

xx xx xx xx: Data Block Data

5.4 Error Log Commands

With these commands the error log of a controller 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 Controller:

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

Return value:

Index	Sub-Index	LSB	MSB
2000h	70h	Payload Data	xxh 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 Controller:

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

Return value:

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

xx xx: SourceID: ID of the part of the firmware which triggered the error:
1: OS
2: Motion Control Software
3: Interface (e.g. CANopen)
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 controllers system time when the error has occurred.

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

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

Return value:

Index	Sub-Index	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.

2000h + EntryNr	73h	Get Error Log Entry Time high	ro	Unsigned32
-----------------	-----	-------------------------------	----	------------

This command returns the higher 32 bits of the controllers system time when the error happened.

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

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

Return value:

Index	Sub-Index	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 + ErrCode	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 Controller:

Index	Sub-Index
2000h + ErrCode	74h + (Stringlet Nr. 0..7)

	LSB				MSB			
Payload Data	-	-	-	-				
Byte	01	02	03	04				

Return value:

Index	Sub-Index
2000h + ErrCode	74h + (Stringlet Nr. 0..7)

	LSB				MSB			
Payload Data	xxh	xxh	xxh	xxh				
Byte	01	02	03	04				

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

5.5 Command Table Commands

See the “LinMot 1100 Servo Controller 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 controller 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 Controller:

Index	Sub-Index	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	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																				
<div>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)</div>																								
<div>CT: Poll Flash</div> <div>SDO Read from PLC to LinMot Controller:</div> <table><tr><th>Index</th><th>Sub-Index</th><th>LSB</th><th>MSB</th></tr><tr><td rowspan="3">2000h</td><td rowspan="3">80h</td><td>Payload Data</td><td>- - - -</td></tr><tr><td>Byte</td><td>01 02 03 04</td></tr></table> <div>Return value:</div> <table><tr><th>Index</th><th>Sub-Index</th><th>LSB</th><th>MSB</th></tr><tr><td rowspan="3">2000h</td><td rowspan="3">80h</td><td>Payload Data</td><td>xxh - - -</td></tr><tr><td>Byte</td><td>01 02 03 04</td></tr></table> <div>xx: Result</div>					Index	Sub-Index	LSB	MSB	2000h	80h	Payload Data	- - - -	Byte	01 02 03 04	Index	Sub-Index	LSB	MSB	2000h	80h	Payload Data	xxh - - -	Byte	01 02 03 04
Index	Sub-Index	LSB	MSB																					
2000h	80h	Payload Data	- - - -																					
		Byte	01 02 03 04																					
		Index	Sub-Index	LSB	MSB																			
2000h	80h	Payload Data	xxh - - -																					
		Byte	01 02 03 04																					
		2000h	81h	CT: Delete all Entries (RAM)	wo	Unsigned32																		
<div>Write anything to delete the complete Command Table in the RAM.</div>																								
<div>CT: Delete all Entries (RAM)</div> <div>SDO Write from PLC to LinMot Controller:</div> <table><tr><th>Index</th><th>Sub-Index</th><th>LSB</th><th>MSB</th></tr><tr><td rowspan="3">2000h</td><td rowspan="3">81h</td><td>Payload Data</td><td>xxh xxh xxh xxh</td></tr><tr><td>Byte</td><td>01 02 03 04</td></tr></table> <div>xx xx xx xx: Any data</div> <div>Return value:</div> <table><tr><th>Index</th><th>Sub-Index</th><th>LSB</th><th>MSB</th></tr><tr><td rowspan="3">2000h</td><td rowspan="3">81h</td><td>Payload Data</td><td>- - - -</td></tr><tr><td>Byte</td><td>01 02 03 04</td></tr></table>					Index	Sub-Index	LSB	MSB	2000h	81h	Payload Data	xxh xxh xxh xxh	Byte	01 02 03 04	Index	Sub-Index	LSB	MSB	2000h	81h	Payload Data	- - - -	Byte	01 02 03 04
Index	Sub-Index	LSB	MSB																					
2000h	81h	Payload Data	xxh xxh xxh xxh																					
		Byte	01 02 03 04																					
		Index	Sub-Index	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 Controller:

Index	Sub-Index	LSB	MSB
2000h + EntryNr	82h	xxh	xxh
		01	02
		03	04

xx xx xx xx: Any data

Return value:

Index	Sub-Index	LSB	MSB
2000h + EntryNr	82h	-	-
		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 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.)
SDO Write from PLC to LinMot Controller:

Index	Sub-Index	LSB	MSB
2000h + EntryNr	83h	xxh	xxh
		-	-
		01	02
		03	04

xx xx: Block size of CT entry

Return value:

Index	Sub-Index	LSB	MSB
2000h + EntryNr	83h	-	-
		-	-
		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 Controller:

Index	Sub-Index	LSB	MSB
2000h + EntryNr	84h	xxh	xxh
		01	02
		03	04

xx xx xx xx: CT entry Data

Return value:

Index	Sub-Index	LSB	MSB
2000h + EntryNr	84h	-	-
		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 Controller:

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

Return value:

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

xx xx: Block size

2000h + EntryNr	86h	CT: Get Entry Data	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 Data
SDO Read from PLC to LinMot Controller:

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

Return value:

Index	Sub-Index	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 Controller:											
<table><tr><th>Index</th><th>Sub-Index</th></tr><tr><td>2000h + EntryNr</td><td>87h</td></tr></table>	Index	Sub-Index	2000h + EntryNr	87h	<table><tr><th>LSB</th><th>MSB</th></tr><tr><td>Payload Data</td><td>-</td></tr><tr><td>Byte</td><td>01 02 03 04</td></tr></table>	LSB	MSB	Payload Data	-	Byte	01 02 03 04
Index	Sub-Index										
2000h + EntryNr	87h										
LSB	MSB										
Payload Data	-										
Byte	01 02 03 04										
Return value:											
<table><tr><th>Index</th><th>Sub-Index</th></tr><tr><td>2000h + EntryNr</td><td>87h</td></tr></table>	Index	Sub-Index	2000h + EntryNr	87h	<table><tr><th>LSB</th><th>MSB</th></tr><tr><td>Payload Data</td><td>xxh</td></tr><tr><td>Byte</td><td>01 02 03 04</td></tr></table>	LSB	MSB	Payload Data	xxh	Byte	01 02 03 04
Index	Sub-Index										
2000h + EntryNr	87h										
LSB	MSB										
Payload Data	xxh										
Byte	01 02 03 04										
xx xx xx xx: Entry presence bitfield											
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.											

Index	Sub-Index	Description	Access Type	Data Type
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 Cabling

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

6.1 Topology Recommendations

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

- Tree Structure:

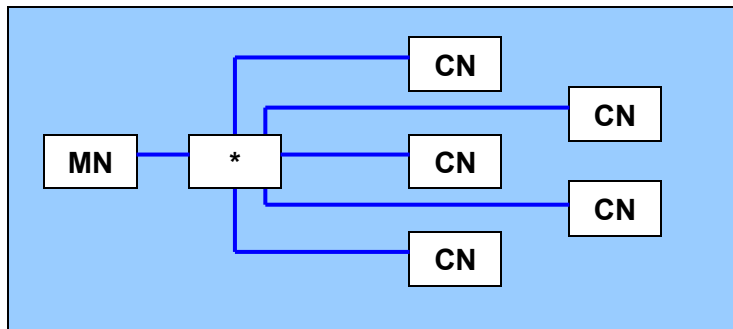


Figure 1: POWERLINK network organised as a tree structure

- Line structure (Daisy Chain):

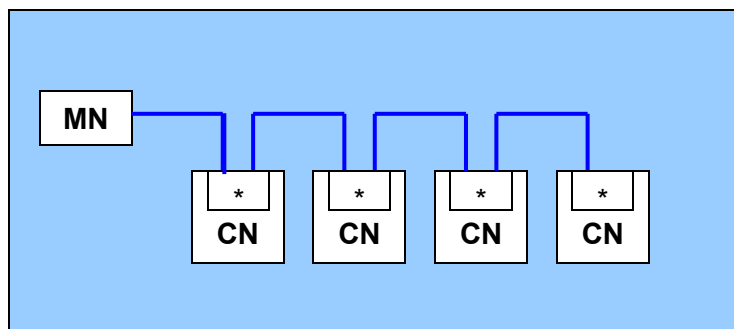


Figure 2: POWERLINK network organised as a line structure

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:

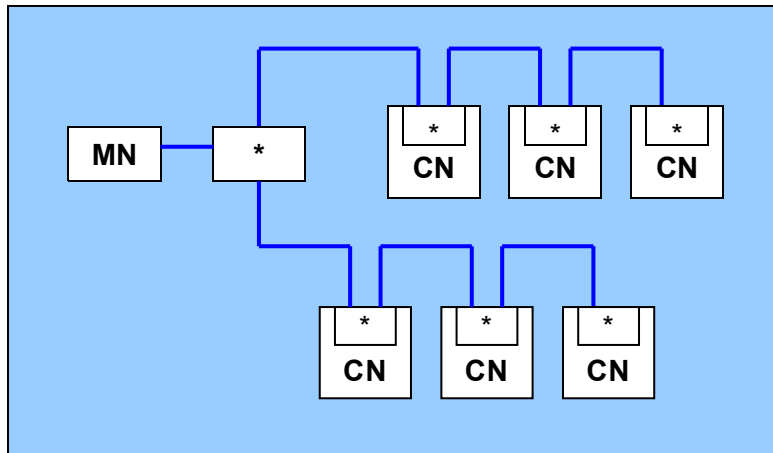


Figure 3: POWERLINK network organised as a line structure

6.2 Calculating the minimal poll response time (T_{PRes}) of an arbitrary CN

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

$$T_{PRes} = 2 \times (T_C + T_H) + T_{PReq_PRes}$$

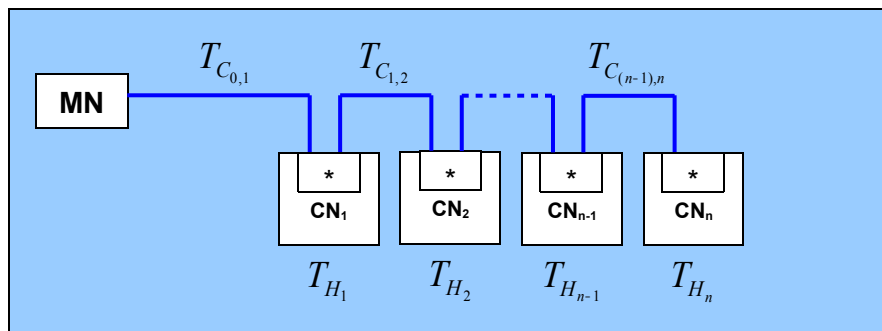
T_{PRes} : Poll Response Time	The total time it takes for a Poll Request message from the time it is being sent from the MN until the MN receives the response from the CN.
T_C : Cable Runtime	The signal runtime of the cabling causes delays. For each meter of cable a delay of approximately 5ns should 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.

6.3 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 cycle 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.



7 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](https://www.skype.com/user/linmot/support)

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 closest to you.

Smart solutions are...

