



ETHERNET   
**POWERLINK**

The Ethernet Powerlink logo graphic consists of a series of colored squares: black, blue, grey, light blue, black, green, blue, black, yellow, and grey.

**Ethernet POWERLINK Interface  
for SG5 and SG6**

User Manual

This document applies to the following devices:

E1250-PL-xx-xx-xxx (SG5)

E1450-PL-xx-xx-xxx (SG5)

C1250-PL-xx-xx-xxx (SG6)

(with POWERLINK Interface SW installed)

© 2014 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 1.4.3 / mk, January 2014

## Inhaltsverzeichnis

<b>1 SYSTEM OVERVIEW.....</b>	<b>4</b>
<b>2 CONNECTING TO THE POWERLINK NETWORK.....</b>	<b>5</b>
2.1 PIN ASSIGNMENT OF THE CONNECTORS X17-X18	5
2.2 SETTING THE NODEID	5
<b>3 POWERLINK PARAMETERS AND VARIABLES.....</b>	<b>6</b>
3.1 PARAMETERS	6
3.2 VARIABLES	7
<b>4 MAPPING OF PRES AND PREQ.....</b>	<b>8</b>
4.1 PDO MAPPING	8
4.2 ADDITIONAL PARAMETERS	9
4.3 MAPPING OF PARAMETERS VIA THEIR UPID	10
4.3.1 <i>Example how to read a parameter via its UPID.....</i>	<i>10</i>
<b>5 CONFIGURATION VIA POWERLINK INTERFACE.....</b>	<b>10</b>
5.1 LINMOT CONFIGURATION MODULE VIA PDO	10
5.2 LINMOT PARAMETER COMMANDS VIA SDO	11
5.2.1 <i>UPID Commands.....</i>	<i>12</i>
5.2.2 <i>System Commands.....</i>	<i>17</i>
5.2.3 <i>Curve Service Commands.....</i>	<i>20</i>
5.2.4 <i>Error Log Commands.....</i>	<i>27</i>
5.2.5 <i>Command Table Commands.....</i>	<i>31</i>
<b>6 CABLING.....</b>	<b>39</b>
6.1 TOPOLOGY RECOMMENDATIONS	39
6.2 CALCULATING THE MINIMAL POLL RESPONSE TIME (TPRES) OF AN ARBITRARY CN	40
6.3 MINIMAL POWERLINK CYCLE TIME FOR A PURE LINE STRUCTURE	41
<b>7 TROUBLESHOOTING.....</b>	<b>42</b>
7.1 ANALYZING TRAFFIC IN POWERLINK NETWORKS	42
7.2 FREQUENT PROBLEMS AND SOLUTIONS	42
<b>8 CONTACT ADDRESSES.....</b>	<b>43</b>

## 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):

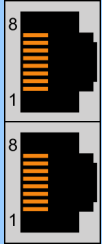
Device Property	Value
Minimal POWERLINK cycle time	200 $\mu$ s
Size of isochronous transmit buffer (maximal size of isochronous frames)	64 Byte
Size of isochronous receive buffer (maximal size of isochronous frames)	64 Byte
Overall buffer size available for isochronous data	128 Byte
PReq to PRes latency (CN isochronous reaction time)	$\sim$ 1 $\mu$ s
SoA to ASnd latency (CN asynchronous reaction time)	< 100 $\mu$ 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



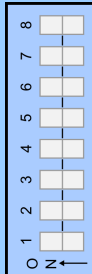
### 2.1 Pin Assignment of the Connectors X17-X18

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		
E12x0	C12x0, E14x0			
			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-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.

#### **POWERLINK\ 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.

#### **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.

### 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\POWERLINK\XDD)

### 4.1 PDO Mapping

Index	Sub-Index	Name	Object Type	Access Type	PDO Mapping
<b>4F00h</b>	-	<b>MCSW Variables</b>	<b>RECORD</b>	<b>const</b>	-
	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
	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
<b>4F02h</b>	-	<b>UPID Variables</b>	<b>RECORD</b>	<b>const</b>	-
	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
<b>4F03h</b>	-	<b>Configuration Module<sup>1</sup></b>	<b>RECORD</b>	<b>const</b>	-
	01h	TX_Cfg_Module_Control	UInt16	wo	optional
	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

<sup>1</sup>See chapter „5.1 LinMot Configuration Module via PDO,“ for details



## 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.

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 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	-
<b>UPID Commands</b>			
	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
<b>System Commands</b>			
	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
<b>Curve Service Commands</b>			
	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
<b>Error Log Commands</b>			
	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
<b>Command Table Commands</b>			
	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	LSB				MSB
2000h + UPID	01h	Payload Data	-	-	-	-
		Byte	01	02	03	04

Return value:

Index	Sub-Index	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	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	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 ROM Value by UPID**

SDO Read from PLC to LinMot Drive:

Index	Sub-Index
2000h + UPID	02h

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

Return value:

Index	Sub-Index
2000h + UPID	02h

	LSB			MSB
Payload Data	xxh	xxh	xxh	xxh
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 Drive:

Index	Sub-Index
2000h + UPID	02h

	LSB			MSB
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
2000h + UPID	02h

	LSB			MSB
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 Max Value by UPID**

SDO Read from PLC to LinMot Drive:

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

Return value:

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

Return value:

Index	Sub-Index	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	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

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.

**Start getting UPID List**

SDO Write from PLC to LinMot Drive:

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><b>Get Next UPID List item</b> SDO Read from PLC to LinMot Drive:</p>																																																
<table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> </tr> </thead> <tbody> <tr> <td>2000h + UPID</td> <td>21h</td> </tr> </tbody> </table>		Index	Sub-Index	2000h + UPID	21h	<table border="1"> <thead> <tr> <th></th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td>Payload Data</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table>				LSB	MSB			Payload Data	-	-	-	-	Byte	01	02	03	04																									
Index	Sub-Index																																															
2000h + UPID	21h																																															
	LSB	MSB																																														
Payload Data	-	-	-	-																																												
Byte	01	02	03	04																																												
<p>Return value:</p>																																																
<table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> </tr> </thead> <tbody> <tr> <td>2000h + UPID</td> <td>21h</td> </tr> </tbody> </table>		Index	Sub-Index	2000h + UPID	21h	<table border="1"> <thead> <tr> <th></th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td>Payload Data</td> <td>yyh</td> <td>yyh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table>				LSB	MSB			Payload Data	yyh	yyh	xxh	xxh	Byte	01	02	03	04																									
Index	Sub-Index																																															
2000h + UPID	21h																																															
	LSB	MSB																																														
Payload Data	yyh	yyh	xxh	xxh																																												
Byte	01	02	03	04																																												
<p>yy yy: Address Usage xx xx: UPID which was found</p> <p>Address Usage is interpreted as follows:</p>																																																
<table border="1"> <tbody> <tr> <td colspan="13">RAM Read allowed</td> <td colspan="2">RAM Write allowed</td> <td colspan="2">ROM Read allowed</td> <td colspan="2">ROM Write allowed</td> <td colspan="4">Live Parameter</td> <td colspan="4">This Parameter is used for security Features. Parameter is included when calculating the hash table.</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> </tbody> </table>					RAM Read allowed													RAM Write allowed		ROM Read allowed		ROM Write allowed		Live Parameter				This Parameter is used for security Features. Parameter is included when calculating the hash table.				Bit Nr.:	0 (LSB)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 (MSB)
RAM Read allowed													RAM Write allowed		ROM Read allowed		ROM Write allowed		Live Parameter				This Parameter is used for security Features. Parameter is included when calculating the hash table.																									
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
<p>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.</p>				

### 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	Set ROM to default (OS)	wo	Unsigned8 - Unsigned32																																						
<p>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.</p> <p><b>Set ROM to default (OS)</b> SDO Write from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th colspan="4">LSB</th> <th>MSB</th> </tr> <tr> <td rowspan="2">2000h</td> <td rowspan="2">07h</td> <th>Payload Data</th> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <th>Byte</th> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </thead></table> <p>xx xx xx xx: Any Data</p> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th colspan="4">LSB</th> <th>MSB</th> </tr> <tr> <td rowspan="2">2000h</td> <td rowspan="2">07h</td> <th>Payload Data</th> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <th>Byte</th> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </thead></table>					Index	Sub-Index	LSB				MSB	2000h	07h	Payload Data	xxh	xxh	xxh	xxh	Byte	01	02	03	04	Index	Sub-Index	LSB				MSB	2000h	07h	Payload Data	-	-	-	-	Byte	01	02	03	04
Index	Sub-Index	LSB				MSB																																				
2000h	07h	Payload Data	xxh	xxh	xxh	xxh																																				
		Byte	01	02	03	04																																				
Index	Sub-Index	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	LSB				MSB
2000h	08h	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	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	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	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	LSB				MSB
2000h	0Ah	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	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	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	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																																				
<p>MC SW and Application SW are stopped. Any data can be written for the command to be executed.</p> <p><b>Stop MC and APPL Software</b> SDO Write from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h</td> <td rowspan="2">35h</td> <td>Payload Data</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table> <p>xx xx xx xx: Any Data</p> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h</td> <td rowspan="2">35h</td> <td>Payload Data</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table>					Index	Sub-Index	LSB	MSB			2000h	35h	Payload Data	xxh	xxh	xxh	xxh	Byte	01	02	03	04	Index	Sub-Index	LSB	MSB			2000h	35h	Payload Data	-	-	-	-	Byte	01	02	03	04
Index	Sub-Index	LSB	MSB																																					
2000h	35h	Payload Data	xxh	xxh	xxh	xxh																																		
		Byte	01	02	03	04																																		
Index	Sub-Index	LSB	MSB																																					
2000h	35h	Payload Data	-	-	-	-																																		
		Byte	01	02	03	04																																		
2000h	36h	Start MC and APPL SW	wo	Unsigned8 - Unsigned32																																				
<p>MC SW and Application SW are started. Any data can be written for the command to be executed.</p> <p><b>Start MC and APPL Software</b> SDO Write from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h</td> <td rowspan="2">36h</td> <td>Payload Data</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table> <p>xx xx xx xx: Any Data</p> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h</td> <td rowspan="2">36h</td> <td>Payload Data</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table>					Index	Sub-Index	LSB	MSB			2000h	36h	Payload Data	xxh	xxh	xxh	xxh	Byte	01	02	03	04	Index	Sub-Index	LSB	MSB			2000h	36h	Payload Data	-	-	-	-	Byte	01	02	03	04
Index	Sub-Index	LSB	MSB																																					
2000h	36h	Payload Data	xxh	xxh	xxh	xxh																																		
		Byte	01	02	03	04																																		
Index	Sub-Index	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
2000h	40h

	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	40h

	LSB			MSB
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	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	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

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	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 drive.

**Curve Service: Add Curve**

SDO Write from PLC to LinMot Drive:

Index	Sub-Index	Payload Data				
		LSB			MSB	
2000h + CurveID	50h	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	Payload Data				
		LSB			MSB	
2000h + CurveID	50h	-	-	-	-	
		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)**

SDO Write from PLC to LinMot Drive:

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

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 Drive:

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

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.

**Curve Service: Get Curve**

SDO Read from PLC to LinMot Drive:

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

**5.2.4 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	LSB				MSB			
2000h	70h	Payload Data	-	-	-	-	-	-	-
		Byte	01	02	03	04			

Return value:

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

Return value:

Index	Sub-Index	LSB				MSB
2000h	71h	Payload Data	xxh	xxh	yyh	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. 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																																																								
<p>This command returns the lower 32 bits of the drive's system time when the error has occurred.</p> <p><b>Get Error Log Entry Time low</b> SDO Read from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th colspan="4">LSB</th> <th colspan="4">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h</td> <td rowspan="2">72h</td> <td>Payload Data</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th colspan="4">LSB</th> <th colspan="4">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h</td> <td rowspan="2">72h</td> <td>Payload Data</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>xx xx xx xx: Lower 32 Bits of the system time the error occurred.</p>					Index	Sub-Index	LSB				MSB				2000h	72h	Payload Data	-	-	-	-	-	-	-	Byte	01	02	03	04				Index	Sub-Index	LSB				MSB				2000h	72h	Payload Data	xxh	xxh	xxh	xxh	xxh	xxh	xxh	Byte	01	02	03	04			
Index	Sub-Index	LSB				MSB																																																						
2000h	72h	Payload Data	-	-	-	-	-	-	-																																																			
		Byte	01	02	03	04																																																						
Index	Sub-Index	LSB				MSB																																																						
2000h	72h	Payload Data	xxh	xxh	xxh	xxh	xxh	xxh	xxh																																																			
		Byte	01	02	03	04																																																						
2000h + EntryNr	73h	Get Error Log Entry Time high	ro	Unsigned32																																																								
<p>This command returns the higher 32 bits of the drive's system time when the error happened.</p> <p><b>Get Error Log Entry Time low</b> SDO Read from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th colspan="4">LSB</th> <th colspan="4">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h</td> <td rowspan="2">73h</td> <td>Payload Data</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th colspan="4">LSB</th> <th colspan="4">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h</td> <td rowspan="2">73h</td> <td>Payload Data</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>xx xx xx xx: Higher 32 Bits of the system time the error occurred.</p>					Index	Sub-Index	LSB				MSB				2000h	73h	Payload Data	-	-	-	-	-	-	-	Byte	01	02	03	04				Index	Sub-Index	LSB				MSB				2000h	73h	Payload Data	xxh	xxh	xxh	xxh	xxh	xxh	xxh	Byte	01	02	03	04			
Index	Sub-Index	LSB				MSB																																																						
2000h	73h	Payload Data	-	-	-	-	-	-	-																																																			
		Byte	01	02	03	04																																																						
Index	Sub-Index	LSB				MSB																																																						
2000h	73h	Payload Data	xxh	xxh	xxh	xxh	xxh	xxh	xxh																																																			
		Byte	01	02	03	04																																																						

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 Drive:

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)

**5.2.5 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
2000h	80h

	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	80h

	LSB			MSB
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
2000h	80h

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

Return value:

Index	Sub-Index
2000h	80h

	LSB		MSB	
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
2000h	81h

	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	81h

	LSB		MSB	
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																																						
<p>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.</p> <p><b>CT: Delete Entry (Entry Nr.)</b> SDO Write from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th colspan="4">LSB</th> <th>MSB</th> </tr> <tr> <td rowspan="2">2000h + EntryNr</td> <td rowspan="2">82h</td> <th>Payload Data</th> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <th>Byte</th> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </thead></table> <p>xx xx xx xx: Any data</p> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th colspan="4">LSB</th> <th>MSB</th> </tr> <tr> <td rowspan="2">2000h + EntryNr</td> <td rowspan="2">82h</td> <th>Payload Data</th> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <th>Byte</th> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </thead></table>					Index	Sub-Index	LSB				MSB	2000h + EntryNr	82h	Payload Data	xxh	xxh	xxh	xxh	Byte	01	02	03	04	Index	Sub-Index	LSB				MSB	2000h + EntryNr	82h	Payload Data	-	-	-	-	Byte	01	02	03	04
Index	Sub-Index	LSB				MSB																																				
2000h + EntryNr	82h	Payload Data	xxh	xxh	xxh	xxh																																				
		Byte	01	02	03	04																																				
Index	Sub-Index	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 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 Drive:

Index	Sub-Index
2000h + EntryNr	83h

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

xx xx: Block size of CT entry

Return value:

Index	Sub-Index
2000h + EntryNr	83h

	LSB		MSB	
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
2000h + EntryNr	84h

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

xx xx xx xx: CT entry Data

Return value:

Index	Sub-Index
2000h + EntryNr	84h

	LSB			MSB
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																																				
<p>Read the block size of a CT Entry.</p> <p><b>CT: Get Entry (Entry Nr.)</b> SDO Read from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h + EntryNr</td> <td rowspan="2">85h</td> <td>Payload Data</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h + EntryNr</td> <td rowspan="2">85h</td> <td>Payload Data</td> <td>xxh</td> <td>xxh</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table> <p>xx xx: Block size</p>					Index	Sub-Index	LSB	MSB			2000h + EntryNr	85h	Payload Data	-	-	-	-	Byte	01	02	03	04	Index	Sub-Index	LSB	MSB			2000h + EntryNr	85h	Payload Data	xxh	xxh	-	-	Byte	01	02	03	04
Index	Sub-Index	LSB	MSB																																					
2000h + EntryNr	85h	Payload Data	-	-	-	-																																		
		Byte	01	02	03	04																																		
Index	Sub-Index	LSB	MSB																																					
2000h + EntryNr	85h	Payload Data	xxh	xxh	-	-																																		
		Byte	01	02	03	04																																		
2000h + EntryNr	86h	CT: Get Entry Data	ro	Unsigned32																																				
<p>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.</p> <p><b>CT: Get Entry Data</b> SDO Read from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h + EntryNr</td> <td rowspan="2">86h</td> <td>Payload Data</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h + EntryNr</td> <td rowspan="2">86h</td> <td>Payload Data</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table> <p>xx xx xx xx: Entry data</p>					Index	Sub-Index	LSB	MSB			2000h + EntryNr	86h	Payload Data	-	-	-	-	Byte	01	02	03	04	Index	Sub-Index	LSB	MSB			2000h + EntryNr	86h	Payload Data	xxh	xxh	xxh	xxh	Byte	01	02	03	04
Index	Sub-Index	LSB	MSB																																					
2000h + EntryNr	86h	Payload Data	-	-	-	-																																		
		Byte	01	02	03	04																																		
Index	Sub-Index	LSB	MSB																																					
2000h + EntryNr	86h	Payload Data	xxh	xxh	xxh	xxh																																		
		Byte	01	02	03	04																																		

Index	Sub-Index	Description	Access Type	Data Type																																				
2000h	87h	CT: Get Entry List (Entry 0..31)	ro	Unsigned32																																				
<p>With this command a bitfield is read, which indicates the presence of a CT entry (0 = CT entry present, 1 = No CT entry present).</p> <p><b>CT: Get Entry List (Entry 0..31)</b> SDO Read from PLC to LinMot Drive:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h + EntryNr</td> <td rowspan="2">87h</td> <td>Payload Data</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table> <p>Return value:</p> <table border="1"> <thead> <tr> <th>Index</th> <th>Sub-Index</th> <th>LSB</th> <th colspan="3">MSB</th> </tr> </thead> <tbody> <tr> <td rowspan="2">2000h + EntryNr</td> <td rowspan="2">87h</td> <td>Payload Data</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> <td>xxh</td> </tr> <tr> <td>Byte</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> </tr> </tbody> </table> <p>xx xx xx xx: Entry presence bitfield</p>					Index	Sub-Index	LSB	MSB			2000h + EntryNr	87h	Payload Data	-	-	-	-	Byte	01	02	03	04	Index	Sub-Index	LSB	MSB			2000h + EntryNr	87h	Payload Data	xxh	xxh	xxh	xxh	Byte	01	02	03	04
Index	Sub-Index	LSB	MSB																																					
2000h + EntryNr	87h	Payload Data	-	-	-	-																																		
		Byte	01	02	03	04																																		
Index	Sub-Index	LSB	MSB																																					
2000h + EntryNr	87h	Payload Data	xxh	xxh	xxh	xxh																																		
		Byte	01	02	03	04																																		
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:

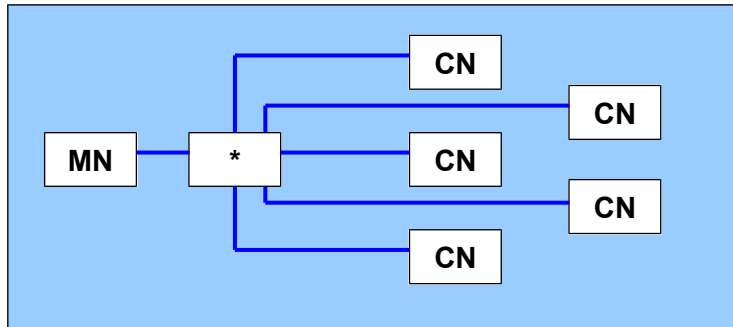


Figure 1: POWERLINK network organised as a tree structure

- Line structure (Daisy Chain):

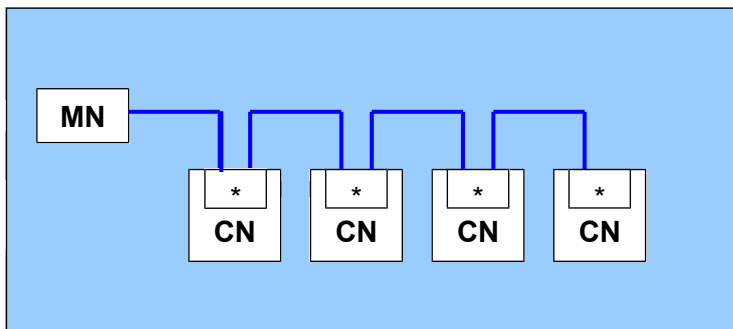


Figure 2: POWERLINK network organised as a line structure

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:

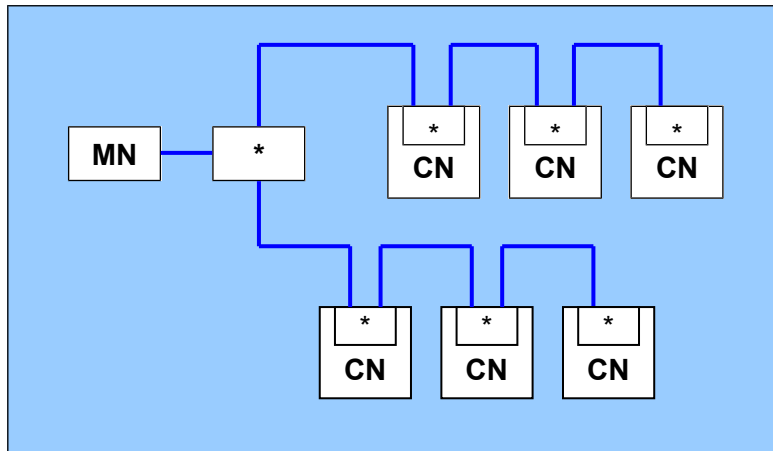


Figure 3: POWERLINK network organised as a line structure

## 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.

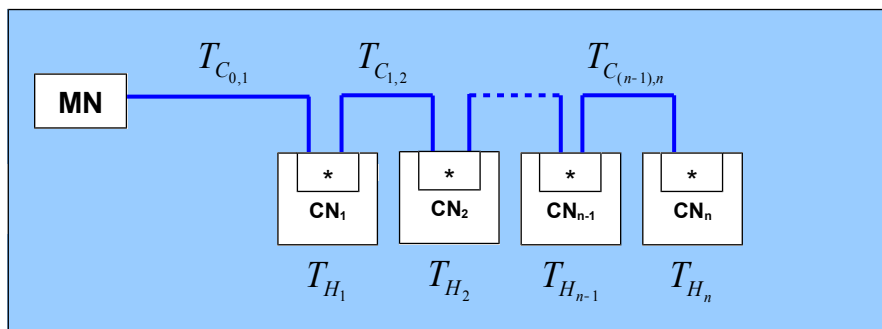


### 6.3 Minimal POWERLINK cycle time for a pure line structure

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 Troubleshooting

### 7.1 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>.



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.

### 7.2 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.

## 8 Contact Addresses

---

### SWITZERLAND

**NTI AG / LinMot**  
Haerdlistr. 15  
CH-8957 Spreitenbach

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

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

**Tech. Support (Skype) :** [skype:support.linmot](https://www.skype.com/contact/linmot)

**Fax:** +41-(0)56-419 91 92

**Web:** <http://www.linmot.com>

---

### USA

**LinMot, Inc.**  
204 E Morrissey Dr.  
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](mailto: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...

