

BECKHOFF® TwinCAT LinMot Bibliothek

E1250-EC-UC, E1130-DP-xx, E1100-CO-xx, B1100-GP-xx

Achskontrolle, MC Befehle & Konfigurationsmodule

Version 1.0.3 (de) fj, 1. Februar 2012

© 2012 NTI AG

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdrucks und der Vervielfältigung des Handbuches oder Teilen daraus, sind vorbehalten. Kein Teil des Werks darf ohne schriftliche Genehmigung von NTI AG in irgendeiner Form (Fotokopie, Mikrofilm oder einem anderen Verfahren), auch nicht für Zwecke der Unterrichtsgestaltung, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

LinMot® ist ein registriertes Markenzeichen von NTI AG.

Hinweis

Die Angaben in dieser Dokumentation entsprechen dem Stand der Entwicklung zur Zeit der Drucklegung und sind daher unverbindlich. NTI AG behält sich vor, Änderungen, die dem technischen Fortschritt bzw. der Produktverbesserung dienen, jederzeit und ohne Angaben von Gründen vorzunehmen. Im Übrigen verweisen wir auf unsere "Allgemeinen Geschäftsbedingungen" in der jeweils gültigen Ausgabe

NTI AG
LinMot®
Haerdlistrasse 15
CH-8957 Spreitenbach

Tel.: +41 (0)56 419 91 91
Fax: +41 (0)56 419 91 92
Email: office@LinMot.com
Homepage: www.LinMot.com

Inhalt

Inhalt.....	3
Dokumentversion.....	4
Einsatz und Verwendung der Bibliothek.....	5
Empfohlene Dokumente.....	5
Allgemein.....	6
1. Hardware Konfiguration.....	7
1.1 <i>TwinCAT System Manager (E1250-EC-UC)</i>	7
1.1.1 LinMot Controller als Slave einfügen.....	7
1.1.2 Verknüpfungen zu PLC Control.....	8
1.1.3 Config Module.....	8
1.2 <i>TwinCAT System Manager (E1130-DP-xx)</i>	9
1.2.1 LinMot Controller als Slave einfügen.....	9
1.2.2 Verknüpfungen zu PLC Control.....	10
1.2.3 Config Module.....	10
1.3 <i>TwinCAT System Manager (E1100-CO-xx, B1100-GP-xx)</i>	11
1.3.1 LinMot Controller als Slave einfügen.....	11
1.3.2 Verknüpfungen zu PLC Control.....	12
1.3.3 Mapping der PDO's.....	13
1.4 <i>Konfiguration LinMot Controller</i>	14
1.4.1 E1250-EC-UC (EtherCAT).....	14
1.4.2 E1130-DP-xx & E1230-DP-UC (Profibus).....	14
1.4.3 E1100-CO-xx & B1100-GP-xx (CANopen).....	14
2. Datentypen.....	15
2.1 <i>Achskommunikation</i>	15
2.1.1 tstLM_Axis.....	15
2.1.2 tstLM_AxisComIn.....	16
2.1.3 tstLM_AxisComOut.....	17
2.1.4 tenLM_AxisState.....	17
2.2 <i>Datentypen der Konfigurationsbausteine</i>	18
2.2.1 tstLM_CfgCTEntry.....	18
2.2.2 tstLM_CfgUPIDListEntry.....	19
3. Bausteine.....	20
3.1 <i>Übersicht und Abhängigkeiten</i>	20
3.2 <i>IO und Achskontrolle</i>	21
3.2.1 LMct_RdAxisCom.....	21
3.2.2 LMct_WrAxisCom.....	21
3.2.3 LMct_AxisCtrlE12x0.....	22
3.2.4 LMct_AxisCtrlx11x0.....	22
3.2.5 Ein- und Ausgänge der Achskontrollbausteine (LMct_AxisCtrlE12x0 & LMct_AxisCtrlx11x0).....	23
3.3 <i>MC Bausteine (E12x0, E11x0, B1100-GP Controller)</i>	24
3.3.1 LMmt_MoveAbs.....	24
3.3.2 LMmt_MoveRel.....	25
3.3.3 LMmt_StartCTCommand.....	26
3.3.4 LMmt_Stop.....	27
3.3.5 LMmt_WriteLivePar.....	28
3.3.6 LMmt_GenericMC.....	29
3.4 <i>MC Bausteine (E12x0, E11x0 Controller)</i>	30
3.4.1 LMav_Mod16BitCTPar.....	30
3.4.2 LMav_Mod32BitCTPar.....	31
3.4.3 LMav_RunCurve.....	32
3.5 <i>MC Bausteine (E12x0 Controller)</i>	33
3.5.1 LMav_MoveBesthorn.....	33
3.5.2 LMav_MoveSin.....	34
3.6 <i>Config Bausteine (E12X0, E1130-DP)</i>	35
3.6.1 LMcf_StopStartDefault.....	36
3.6.2 LMcf_CTAccess.....	37
3.6.3 LMcf_CurveAccess.....	38
3.6.4 LMcf_ParaAccess.....	39

3.6.5 LMcf_GetModUPIDList.....	40
3.6.6 LMcf_WriteUPIDList.....	41
3.6.7 LMcf_GetErrorTxt.....	42
3.7 Config Bausteine CANopen (E11x0, B1100-GP Controller).....	43
3.7.1 LMcf_SDORead.....	43
3.7.2 LMcf_SDOWrite.....	44
4. Fehlerbeschreibungen.....	45
4.1 Fehler der Achskontrollbausteine (ErrorCode).....	45
4.2 Fehlernummern der MC Bausteine.....	45
4.3 Fehlernummern der Config Bausteine.....	45
Kontakt.....	46

Dokumentversion

Version	Datum	Autor	Bibliotheksversion	Beschreibung
1.0.0	15.08.2011	fj	1.0.2	Initialversion
1.0.3	01.02.2012	fj	1.0.3	Änderungen Dokument: <ul style="list-style-type: none"> Keine Änderung Änderungen Bibliothek: <ul style="list-style-type: none"> Baustein LMct_AxisCtrlE12x0 & LMct_AxisCtrlx11x0: Operation Enabled Ausgang direkt auf StatusWord Bit 0 gemappt.

Einsatz und Verwendung der Bibliothek

Die vorgestellte Bibliothek für BECKHOFF TwinCAT stellt Bausteine zur Ansteuerung von LinMot Controller mit EtherCAT, Profibus oder CANopen Schnittstelle zur Verfügung. Diese Bibliothek wird kostenfrei NTI AG / LinMot zur Verfügung gestellt. Für den Anwender besteht kein Anspruch auf Gewährleistung und Updates. Ebenso wird die Haftung für Schäden, die infolge Einsatz dieser Bibliothek auftreten, ausdrücklich ausgeschlossen.

Das vorliegende Dokument sowie die Bibliothek werden von NTI AG / LinMot kostenlos zur Verfügung gestellt.

Unterstützte Controller: E1250-EC-UC, E1230-DP-UC, E1130-DP-xx, E1100-CO-xx, B1100-GP-xx

Klassifizierung: ☐ LinMot intern
☒ Weitergabe an Kunden erlaubt

Freigabe: ☒ Bausteinbibliothek
☐ Einsatz in produktiver Umgebung

Empfohlene Dokumente

Die Lektüre der folgenden Handbücher ist Voraussetzung zum Verständnis der Kommunikation zwischen SPS und dem LinMot Controller. Die Handbücher sind in der LinMot-Talk Software enthalten, oder können von der Homepage heruntergeladen werden: <http://www.linmot.com/index.php?id=204>

E1250-EC-xx:

- LinMot-Talk 4 User Manual
- User Manual Motion Control Software E1200
- EtherCAT: User manual for E1250-EC-xx
- User manual configuration over fieldbus interface E1200 Series

E1130-DP-xx:

- LinMot-Talk 4 User Manual
- User Manual Motion Control Software E1100/B1100
- PROFIBUS: User manual for E1130-DP (-HC, -XC)
- User manual configuration over fieldbus interface E1100/B1100 Series

E1100-CO-xx und B1100-GP-xx:

- LinMot-Talk 4 User Manual
- User Manual Motion Control Software E1100/B1100
- CANopen: User Manual for E1100-CO (-HC, -XC), E1100-GP (-HC, -XC) and B1100-GP (-HC, -XC)
- User manual configuration over fieldbus interface E1100/B1100 Series

Allgemein

Die LinMot Controller können über verschiedene Feldbusse mit einer BECKHOFF Steuerung verbunden werden. Um das Einbinden der Controller zu vereinfachen und die generelle Ansteuerung aufzuzeigen, wird in diesem Dokument die vorliegende BECKHOFF TwinCAT Bibliothek vorgestellt.

Das Programmpaket besteht aus folgenden Bausteinen und Datentypen:

Achskontrolle:

- LMct_RdAxisCom
 - LMct_WrAxisCom
 - LMct_AxisCtrlx11x0
 - LMct_AxisCtrlE12x0
- E1130-DP-xx, E1100-CO-xx, B1100-GP-xx
E1250-EC-UC, E1230-DP-UC

MC Bausteine (Alle Controller):

- LMmt_MoveAbs
- LMmt_MoveRel
- LMmt_StartCTCommand
- LMmt_Stop
- LMmt_WriteLivePar
- LMmt_GenericMC

MC Bausteine (E1250-EC-UC, E1230-DP-UC, E1130-DP-xx, E1100-CO-xx):

- LMav_Mod16BitCTPar
- LMav_Mod32BitCTPar
- LMav_PStream
- LMav_PVStream
- LMav_RunCurve

MC Bausteine (Nur E1250-EC-UC):

- LMav_MoveBesthorn
- LMav_MoveSin

Config Bausteine (nur E1250-EC-UC, E1230-DP-UC & E1130-DP-xx):

- LMcf_CTAccess
- LMcf_CurveAccess
- LMcf_GetErrorTxt
- LMcf_GetModUPIDList
- LMcf_ParaAccess
- LMcf_StopStartDefault
- LMcf_WriteUPIDList

Config Bausteine CANopen (E12x0, E11x0, B1100-GP-xx)

Nicht Teil der LinMot.lib! SDOACCESS_X1X00.EXP ins Projekt importieren

- LMcf_SDORead
- LMcf_SDOWrite

Datentypen:

- tstLM_AxisRef
- tenLM_AxisState
- tstLM_AxisComIn
- tstLM_AxisComOut
- tstLM_CfgCTEntry
- tstLM_CfgUPIDListEntry

Von den Funktionsbausteinen können beliebig viele Instanzen angelegt werden.

Das Projekt wurde mit BECKHOFF TwinCAT in der Version v2.11.0 (Build 1555) erstellt

PLC Control Version v2.11.0 (Build 1016)

System Manager Version v2.11.0 (Build 1569)

1. Hardware Konfiguration

1.1 TwinCAT System Manager (E1250-EC-UC)

1.1.1 LinMot Controller als Slave einfügen

Mittels Rechtsklick auf den EtherCAT-Master (Abbildung 1) wird eine neuer Controller eingefügt.
Die benötigte XML-Datei befindet sich standardmässig in folgendem Verzeichnis:

C:\Programme\LinMot\LinMot-Talk X.X Build XXXXXXXX\Firmware\Interfaces\EtherCAT\XML

Diese muss ins IO Verzeichnis von TwinCAT kopiert werden:

C:\TwinCAT\Io\EtherCAT

Hinweis: Der Pfad hängt von der installierten Version von LinMot-Talk ab.

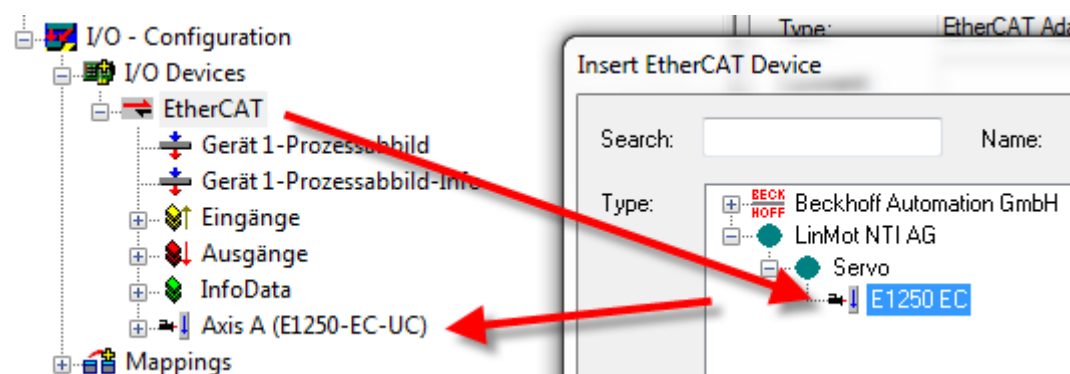


Abbildung 1: LinMot EtherCAT Slave einfügen

1.1.2 Verknüpfungen zu PLC Control

Die Ein- und Ausgänge der verschiedenen Module werden wie in Abbildung 2 zu sehen verknüpft. Die verknüpften Variablen sind in den globalen Variablen, Abschnitt „Axis_A Bus Communication“, im PLC Programm definiert.

Name	Linked to
StatusWord	StatusWord . Axis_A_ComIn . Inputs . Standard .
WarnWord	
DemandPosition	
ActualPosition	ComActualPosition . Axis_A_ComIn . Inputs . Sta
DemandCurrent	ComActualCurrent32 . Axis_A_ComIn . Inputs . S
Config Status Word	Status . Axis_A_ComIn . Inputs . Standard . LinM
Config Index In	Index_In . Axis_A_ComIn . Inputs . Standard . LinI
Config Value In	Value_In . Axis_A_ComIn . Inputs . Standard . LinI
Control Word	ControlWord . Axis_A_ComOut . Outputs . Stand
Motion Command Header	MCHeader . Axis_A_ComOut . Outputs . Standar
Motion Command Par 1	MCPParaDWord_00_03 . Axis_A_ComOut . Output
Motion Command Par 2	MCPParaDWord_04_07 . Axis_A_ComOut . Output
Motion Command Par 3	MCPParaDWord_08_11 . Axis_A_ComOut . Output
Motion Command Par 4	MCPParaDWord_12_15 . Axis_A_ComOut . Output
Motion Command Par 5	MCPParaDWord_16_19 . Axis_A_ComOut . Output
Config Control Word	Control . Axis_A_ComOut . Outputs . Standard . I
Config Index Out	Index_Out . Axis_A_ComOut . Outputs . Standar
Config Value Out	Value_Out . Axis_A_ComOut . Outputs . Standar

Abbildung 2: Ein- und Ausgangsverknüpfungen

1.1.3 Config Module

Wenn die Config Bausteine verwendet werden, muss zusätzlich das „Config Module“ aktiviert (Abbildung 3) und verknüpft werden (Abbildung 2).

The screenshot shows the 'I/O - Configuration' window. On the left, the 'I/O Devices' tree is expanded, showing 'EtherCAT' and 'Gerät 1-Prozessabbild-Info'. Under 'Gerät 1-Prozessabbild-Info', the 'Axis A (E1250-EC-UC)' module is highlighted. A red arrow points from this module to the 'Process Data' tab on the right. In the 'Process Data' tab, the 'PDO Assignment (0x1C12)' section shows a list of PDOs with checkboxes. The checkboxes for '0x1700' and '0x1708' are checked and highlighted with a red box. The 'PDO List' table on the right shows the following data:

Index
0x1B00
0x1B08
0x1B10
0x1B11
0x1700

The 'PDO Content' section shows the 'Index' field with the value '0x1B62:00' and a 'Predefined P' button.

Abbildung 3: Config Module aktivieren

1.2 TwinCAT System Manager (E1130-DP-xx)

1.2.1 LinMot Controller als Slave einfügen

Mittels Rechtsklick auf den Profibus-Master (Hier FC31XX, Abbildung 4) wird eine neue „Allgemeine Profibus Box“ eingefügt. Die benötigte GSD-Datei befindet sich standardmässig in folgendem Verzeichnis:

C:\Programme\LinMot\LinMot-Talk X.X Build XXXXXXXXX\Firmware\Interfaces\Profibus\GSD\

Hinweis: Der Pfad hängt von der installierten Version von LinMot-Talk ab.

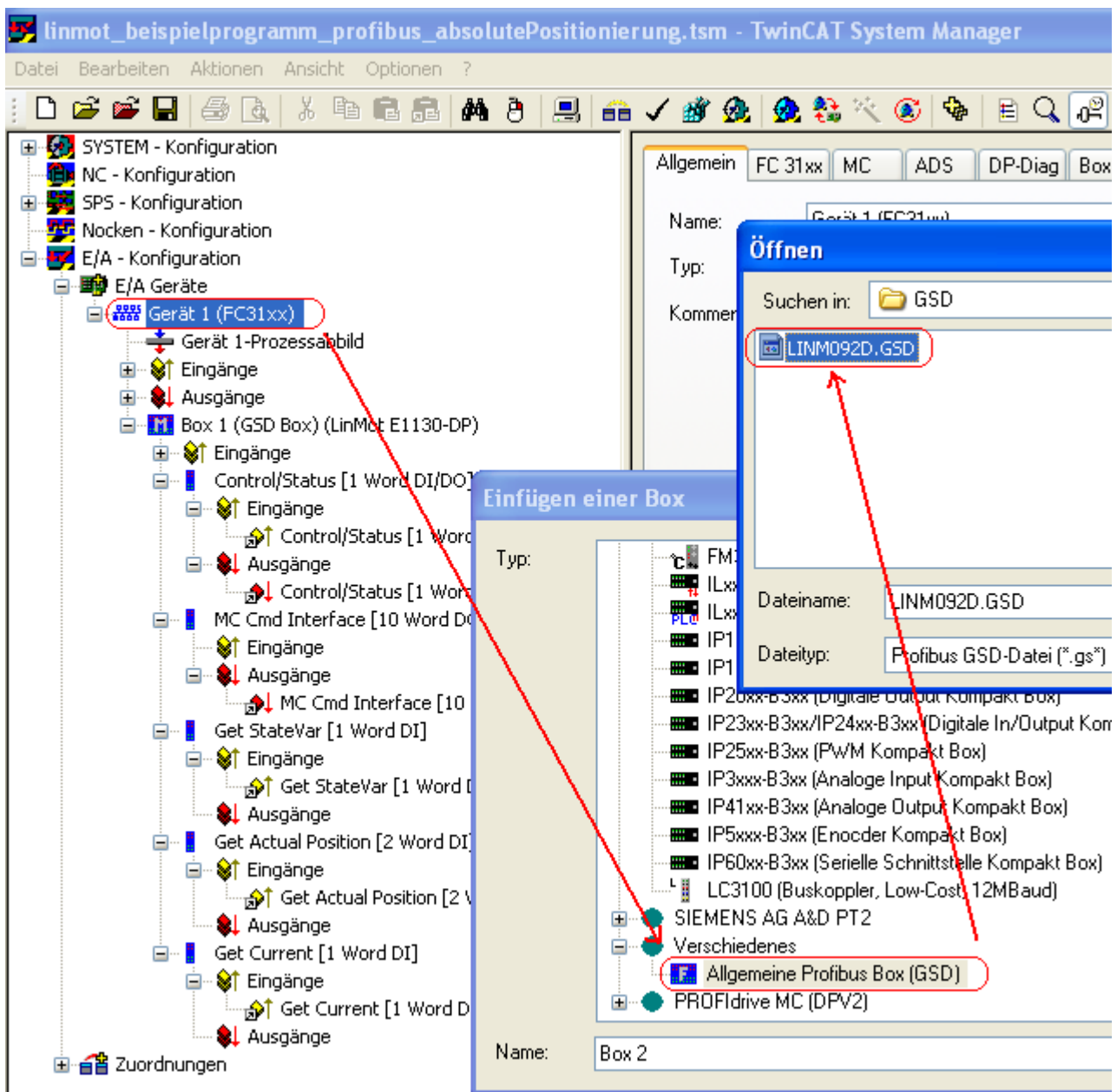


Abbildung 4: Installieren des LinMot Controllers mittels GSD-Datei

Anschliessend werden die benötigten Module (Control/Status, MC Cmd Interface, Get StateVar, Get Actual Position und Get Current) eingefügt. Rechtsklick auf Box 1 -> Modul anfügen.

1.2.2 Verknüpfungen zu PLC Control

Die Ein- und Ausgänge der verschiedenen Module werden wie in Abbildung 5 zu sehen verknüpft. Die verknüpften Variablen sind in den globalen Variablen, Abschnitt „Axis A Bus Communication“, im PLC Programm definiert.

Name	Linked to
Control/Status [1 Word DI/DO]_0_0	StatusWord . Axis_A_ComIn . Inputs . Standard . LinMot_Libra...
Get StateVar [1 Word DI]_0_0	StateVar . Axis_A_ComIn . Inputs . Standard . LinMot_Library_I...
Get Actual Position [2 Word DI]_0_0	ComActualPosition . Axis_A_ComIn . Inputs . Standard . LinM...
Get Current [1 Word DI]_0_0	ComActualCurrent16 . Axis_A_ComIn . Inputs . Standard . Lin...
Parameter Channel [4 Word DI/DO]_0_0	Status, Index_In, Value_In
Control/Status [1 Word DI/DO]_1_0	ControlWord . Axis_A_ComOut . Outputs . Standard . LinMot_...
MC Cmd Interface [10 Word DO]_1_0	MCHeader, MCPParaWord0, MCPParaWord1, MCPParaWord2, M...
Parameter Channel [4 Word DI/DO]_1_0	Control, Index_Out, Value_Out

Abbildung 5: Ein- und Ausgangsverknüpfungen

1.2.3 Config Module

Wenn die Config Bausteine verwendet werden, muss zusätzlich das Modul „Parameter Channel“ hinzugefügt und verknüpft (Abbildung 5) werden.

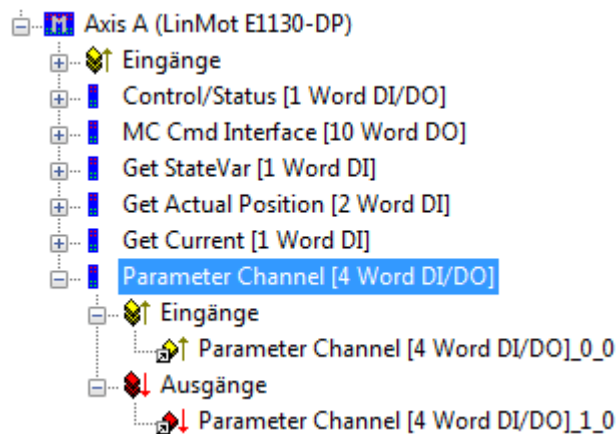


Abbildung 6: Modul "Parameter Channel"

1.3 TwinCAT System Manager (E1100-CO-xx, B1100-GP-xx)**1.3.1 LinMot Controller als Slave einfügen**

Mittels Rechtsklick auf den CANopen-Master (Hier EL6751, Abbildung 7) und „Gerät Importieren...“ wird die diesem Dokument beiliegende „Axis A (LinMot X11X0).tce“ eingefügt.

Das importierte Geräte verfügt bereits über sämtliche Verknüpfungen zum PLC-Programm. Die MACID ist 2 und kann entsprechend angepasst werden.

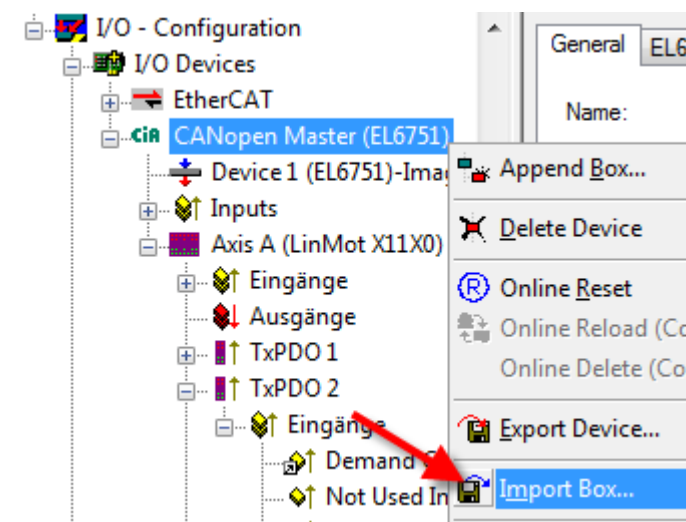


Abbildung 7: Gerät importieren

Eine passende EDS Datei für die Bibliothek ist ebenfalls beigelegt (X11X0_9W_MCI.EDS), wobei der Transmission Type der PDO's manuell auf 1 (cyc, sync) gesetzt werden muss (Siehe Abbildung 8).

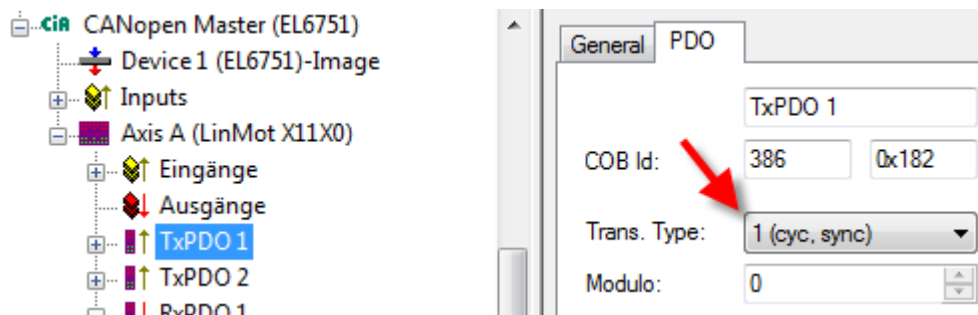


Abbildung 8: Transmission Type 1 (Cyc, Sync)

**Hinweis**

Wird wie in Abbildung 7 das Gerät importiert, wird die EDS Datei nicht benötigt.

1.3.2 Verknüpfungen zu PLC Control

Die Ein- und Ausgänge der in den PDO's definierten Variablen werden wie in Abbildung 9 zu sehen verknüpft. Die verknüpften Variablen sind in den globalen Variablen, Abschnitt „Axis_A Bus Communication“, im PLC Programm definiert.
















Name	Linked to
 ↑ Status Word [1 W]	StatusWord . Axis_A_ComIn . Inputs . Standard . LinMot
 ↑ State Var [1 W]	StateVar . Axis_A_ComIn . Inputs . Standard . LinMot_Lil
 ↑ Actual Position [2 W]	ComActualPosition . Axis_A_ComIn . Inputs . Standard
 ↑ Demand Current [1 W]	ComActualCurrent16 . Axis_A_ComIn . Inputs . Standar
 ↓ Control Word [1 W]	ControlWord . Axis_A_ComOut . Outputs . Standard . Li
 ↓ MCHeader [1 W]	MCHeader . Axis_A_ComOut . Outputs . Standard . Lin
 ↓ MCPParameterWord0 [1 W]	MCPParaWord0 . Axis_A_ComOut . Outputs . Standard .
 ↓ MCPParameterWord1 [1 W]	MCPParaWord1 . Axis_A_ComOut . Outputs . Standard .
 ↓ MCPParameterWord2 [1 W]	MCPParaWord2 . Axis_A_ComOut . Outputs . Standard .
 ↓ MCPParameterWord3 [1 W]	MCPParaWord3 . Axis_A_ComOut . Outputs . Standard .
 ↓ MCPParameterWord4 [1 W]	MCPParaWord4 . Axis_A_ComOut . Outputs . Standard .
 ↓ MCPParameterWord5 [1 W]	MCPParaWord5 . Axis_A_ComOut . Outputs . Standard .
 ↓ MCPParameterWord6 [1 W]	MCPParaWord6 . Axis_A_ComOut . Outputs . Standard .
 ↓ MCPParameterWord7 [1 W]	MCPParaWord7 . Axis_A_ComOut . Outputs . Standard .
 ↓ MCPParameterWord8 [1 W]	MCPParaWord8 . Axis_A_ComOut . Outputs . Standard .

Abbildung 9: Ein- und Ausgangsverknüpfungen zum PLC Programm

1.3.3 Mapping der PDO's

Abbildung 10 zeigt das Mapping der einzelnen PDO's

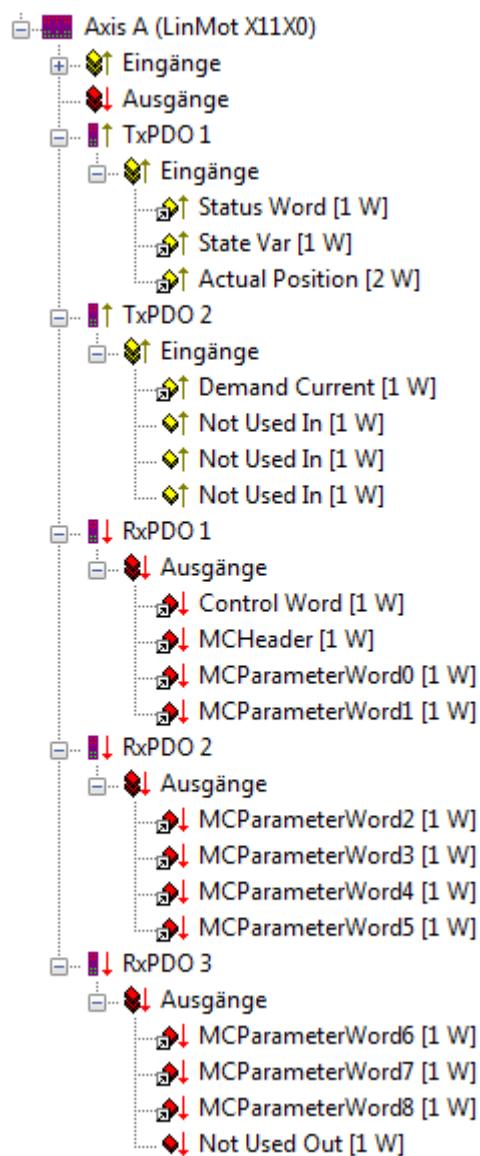


Abbildung 10: PDO Mapping

1.4 Konfiguration LinMot Controller

Der LinMot Controller wird mit der Software LinMot-Talk konfiguriert. <http://www.linmot.com/index.php?id=204>

Es wird vorausgesetzt, dass der am Controller angeschlossene Motor bereits konfiguriert wurde.



Hinweis

Im Zweifelsfall den Controller auf Werkseinstellungen zurücksetzen und anschliessend den Antrieb mit dem Motoren-Wizard neu konfigurieren. Vorher alte Konfiguration sichern!

Rücksetzen auf Werkseinstellungen (Nur mit E1xx0 Controllern möglich):

1. Controller ausschalten (24V).
2. Beide ID Drehschalter (S1 und S2) auf F setzen.
3. Controller einschalten (24V). Die ERROR und WARN LED blinken abwechselungsweise.
4. Beide ID Drehschalter auf 0 setzen.
5. Warten bis EN und WARN LED gleichzeitig blinken.
6. Controller aus- und wieder einschalten.

Alternativ für alle Controllertypen kann in LinMot-Talk über „Datei -> Offline erzeugen“ eine leere Konfigurationsdatei erstellt werden (Gewünschtes Interface und Applikation auswählen!).

Diese wird über „Datei -> Export“ gespeichert und kann anschliessen auf den Controller importiert werden.

1.4.1 E1250-EC-UC (EtherCAT)

Keine Einstellungen auf Controllerseite nötig.

1.4.2 E1130-DP-xx & E1230-DP-UC (Profibus)

Die einzige Einstellung, die am Controller bezüglich Profibus durchgeführt werden muss, ist das Setzen der "Node Adress". Standardmässig wird sie über die beiden Hex-Schalter S1 (ID High) und S2 (ID Low) am Gehäuse eingestellt. Alternativ kann sie auch über den Parameter „Node Adress Parameter Value“ (UPID 2076h) gesetzt werden.

Dazu muss der Parameter „Node Adress Selection“ (UPID 206Ch) auf „On“ gesetzt sein.

Alle anderen Profibusparameter werden auf ihren Standardwerten belassen.

1.4.3 E1100-CO-xx & B1100-GP-xx (CANopen)

E11x0 Serie Controller:

- Importieren der CANopen Interface Konfiguration (Liegt der Bibliothek bei) (Datei -> Import, E11x0_CANopen_Interface_Config.lmc)
- Setzen der MACID über den hexadezimalen Drehschalter S2 am Controller

B1100-GP Controller:

- Importieren der CANopen Interface Konfiguration (Liegt der Bibliothek bei) (Datei -> Import, B1100-GP_CANopen_Interface_Config.lmc)
- Konfigurieren der MACID und Baudrate
 - MACID: UPID 6081h auf die gewünschte MACID setzen
 - Baudrate: UPID 60B3h auf 500 kBit/s setzen

2. Datentypen

2.1 Achskommunikation

2.1.1 tstLM_Axis

Die Datenstruktur tstLM_Axis enthält sämtliche Daten für die Kommunikation zwischen den Bausteinen der Bibliothek und dem LinMot Controller.

TYPE tstLM_Axis :

STRUCT

(*This struct includes all required data for a proper use of the LinMot Function Blocks*)

(*General outputs to fieldbus*)

```
ControlWord      :UINT;
MCHeader         :UINT;
MCParaWord0      :UINT;
MCParaWord1      :UINT;
MCParaWord2      :UINT;
MCParaWord3      :UINT;
MCParaWord4      :UINT;
MCParaWord5      :UINT;
MCParaWord6      :UINT;
MCParaWord7      :UINT;
MCParaWord8      :UINT;
MCParaWord9      :UINT;
MCParaWord10     :UINT;
MCParaWord11     :UINT;
MCParaWord12     :UINT;
MCParaWord13     :UINT;
```

(*General inputs from fieldbus*)

```
StatusWord      :UINT;
StateVar         :UINT;
ComActualPosition :DINT;
ComActualCurrent32 :DINT; (*32Bit actual current For E1200 series controllers*)
ComActualCurrent16 :INT;  (*16Bit actual current For x11x0 series controller*)
```

(*Config channel outputs to fieldbus*)

```
Control          :UINT; (*Config Channel Control*)
Index_Out         :UINT; (*Argument (meaning depends on Cmd ID*)
Value_Out         :UDINT; (*Argument (meaning depends on Cmd ID*)
```

(*Config channel inputs from fieldbus*)

```
Status           :UINT; (*Config Channel Status*)
Index_In         :UINT; (*Argument (meaning depends on Cmd ID*)
Value_In         :UDINT; (*Argument (meaning depends on Cmd ID*)
```

(*Status*)

```
AxisState        :tenLM_AxisState;
CommandRunning   :BOOL;
CommandAborted   :BOOL;
ConfigChannelBusy :BOOL;
```

(*Information (Not required for proper work of the library function blocks)*)

```
AxisName         :STRING;
AxisNr           :UINT;
AxisCtrlType     :STRING;(*Eg. 'E1200-EC-UC', 'E1130-DP', 'B1100-GP', ....*)
```

END_STRUCT

END_TYPE

Abbildung 11: tstLM_Axis

**Hinweis**

Zusätzliche Informationen bezüglich der Bedeutung der einzelnen Daten sind im Handbuch "Motion Control Software" zu finden. (Empfohlene Dokumente)

2.1.2 tstLM_AxisComIn

Die Datenstruktur tstLM_AxisComIn enthält die vom Controller gesendeten Daten und kann direkt mit dem Systemmanager verknüpft werden.

```
TYPE tstLM_AxisComIn :  
STRUCT  
    StatusWord : UINT; (*Statusword of the controller*)  
    StateVar : UINT; (*StateVar of the controller*)  
    ComActualPosition : DINT; (*32Bit actual position*)  
    ComActualCurrent32 : DINT; (*32Bit actual current For E1200 series controllers*)  
    ComActualCurrent16 : INT; (*16Bit actual current For x11x0 series controller*)  
  
    Status : UINT; (*Config Channel Status*)  
    Index_In : UINT; (*Argument (meaning depends on Cmd ID*)  
    Value_In : UDINT; (*Argument (meaning depends on Cmd ID*)  
END_STRUCT  
END_TYPE
```

Abbildung 12: tstLM_AxisComIn

2.1.3 tstLM_AxisComOut

Die Datenstruktur tstLM_AxisComIn enthält die an den Controller gesendeten Daten und kann direkt mit dem Systemmanager verknüpft werden.

TYPE tstLM_AxisComOut :

STRUCT

```

ControlWord : UINT; (*ControlWord of the controller*)
MCHheader : UINT; (*Motion command header*)
MCParaWord0 : UINT; (*Motion command parameter word 0*)
MCParaWord1 : UINT; (*Motion command parameter word 1*)
MCParaWord2 : UINT; (*Motion command parameter word 2*)
MCParaWord3 : UINT; (*Motion command parameter word 3*)
MCParaWord4 : UINT; (*Motion command parameter word 4*)
MCParaWord5 : UINT; (*Motion command parameter word 5*)
MCParaWord6 : UINT; (*Motion command parameter word 6*)
MCParaWord7 : UINT; (*Motion command parameter word 7*)
MCParaWord8 : UINT; (*Motion command parameter word 8*)
MCParaWord9 : UINT; (*Motion command parameter word 9*)
MCParaWord10 : UINT; (*Motion command parameter word 10*)
MCParaWord11 : UINT; (*Motion command parameter word 11*)
MCParaWord12 : UINT; (*Motion command parameter word 12*)
MCParaWord13 : UINT; (*Motion command parameter word 13*)
MCParaDWord_00_03 : UDINT; (*Motion command parameter bytes 0-3 (used for E12X0 s
MCParaDWord_04_07 : UDINT; (*Motion command parameter bytes 4-7 (used for E12X0 s
MCParaDWord_08_11 : UDINT; (*Motion command parameter bytes 8-11 (used for E12X0
MCParaDWord_12_15 : UDINT; (*Motion command parameter bytes 12-15 (used for E12XI
MCParaDWord_16_19 : UDINT; (*Motion command parameter bytes 16-19 (used for E12XI
MCParaDWord_20_23 : UDINT; (*Motion command parameter bytes 20-23 (used for E12XI
MCParaDWord_24_27 : UDINT; (*Motion command parameter bytes 24-27 (used for E12XI

Control : UINT; (*Config Channel Control*)
Index_Out : UINT; (*Argument (meaning depends on Cmd ID*)
Value_Out : UDINT; (*Argument (meaning depends on Cmd ID*)

```

END_STRUCT

END_TYPE

Abbildung 13: tstLM_AxisComOut

2.1.4 tenLM_AxisState

An PLCopen angelehnte Zustandsanzeige der Achse. Hat derzeit keine funktionelle Bedeutung.

TYPE tenLM_AxisState :

(*PLC Open like naming of the controllers state - Not proper implemented yet*)

```

(
    Disabled_LM,
    StandStill_LM,
    Homing_LM,
    ErrorStop_LM,
    Stopping_LM,
    DiscreteMotion_LM,
    ContinuousMotion_LM,
    SynchronizedMotion_LM,
    Jogging_LM

```

);

END_TYPE

Abbildung 14: tenLM_AxisState

2.2 Datentypen der Konfigurationsbausteine

2.2.1 tstLM_CfgCTEntry

Die Datenstruktur tstLM_CfgCTEntry enthält die Daten einer Zeile der Command Table.
Wird in folgendem Konfigurationsbaustein verwendet:

- LMcf_CTAccess

Nur für E1250-EC-UC, E1230-DP-UC & E1130-DP-xx Controller!

TYPE LM_stConfigCTEntry :

STRUCT

CTID	:UINT;	(*ID of command table entry*)
BlockSize	:UINT;	(*Size of EntryBlock in Bytes*)
DataBlock	:ARRAY[0..15] OF UDINT;	(*EntryBlock, 64Bytes*)

(* Example of a Command Table Entry Data Block:

0002A701h,	LowWord = Entry Version ID (fix A701h) / HighWord = Sequenced Entry ID
00000100h,	LowWord: Motion Cmd Header(010xh=VAI Go To Pos) / HighWord: Motion Cmd Par Byte 0..1
42400000h,	Motion Cmd Par Byte 2..5
4240000Fh,	Motion Cmd Par Byte 6..9
4240000Fh,	Motion Cmd Par Byte 10..13
0000000Fh,	Motion Cmd Par Byte 14..17
00000000h,	Motion Cmd Par Byte 18..21
00000000h,	Motion Cmd Par Byte 22..25
00000000h,	Motion Cmd Par Byte 26..29
6E550000h,	LowWord: Motion Cmd Par Byte 30..31 / HighWord: Entry Name Character 0..1
656D616Eh,	Entry Name Character 2..5
00000064h,	Entry Name Character 6..9
00000000h,	Entry Name Character 10..13
FFFF0000h,	LowWord: Entry Name Character 14..15 / HighWord: Reserved for future use
FFFFFFFFh,	Reserved for future use
FFFFFFFFh,	Reserved for future use

*)

END_STRUCT

END_TYPE

Abbildung 15: tstLM_CfgCTEntry



Hinweis

Zusätzliche Informationen bezüglich der Bedeutung der einzelnen Daten sind im Handbuch "LinMot Controller Configuration over Fieldbus Interfaces" zu finden. (Empfohlene Dokumente)

2.2.2 tstLM_CfgUPIDListEntry

Die Datenstruktur tstLM_CfgUPIDListEntry enthält die Nummer und den Wert eine LinMot Parameters.
Wird als Array in folgenden Konfigurationsbausteinen verwendet:

- LMcf_GetModUPIDList
- LMcf_WriteUPIDList

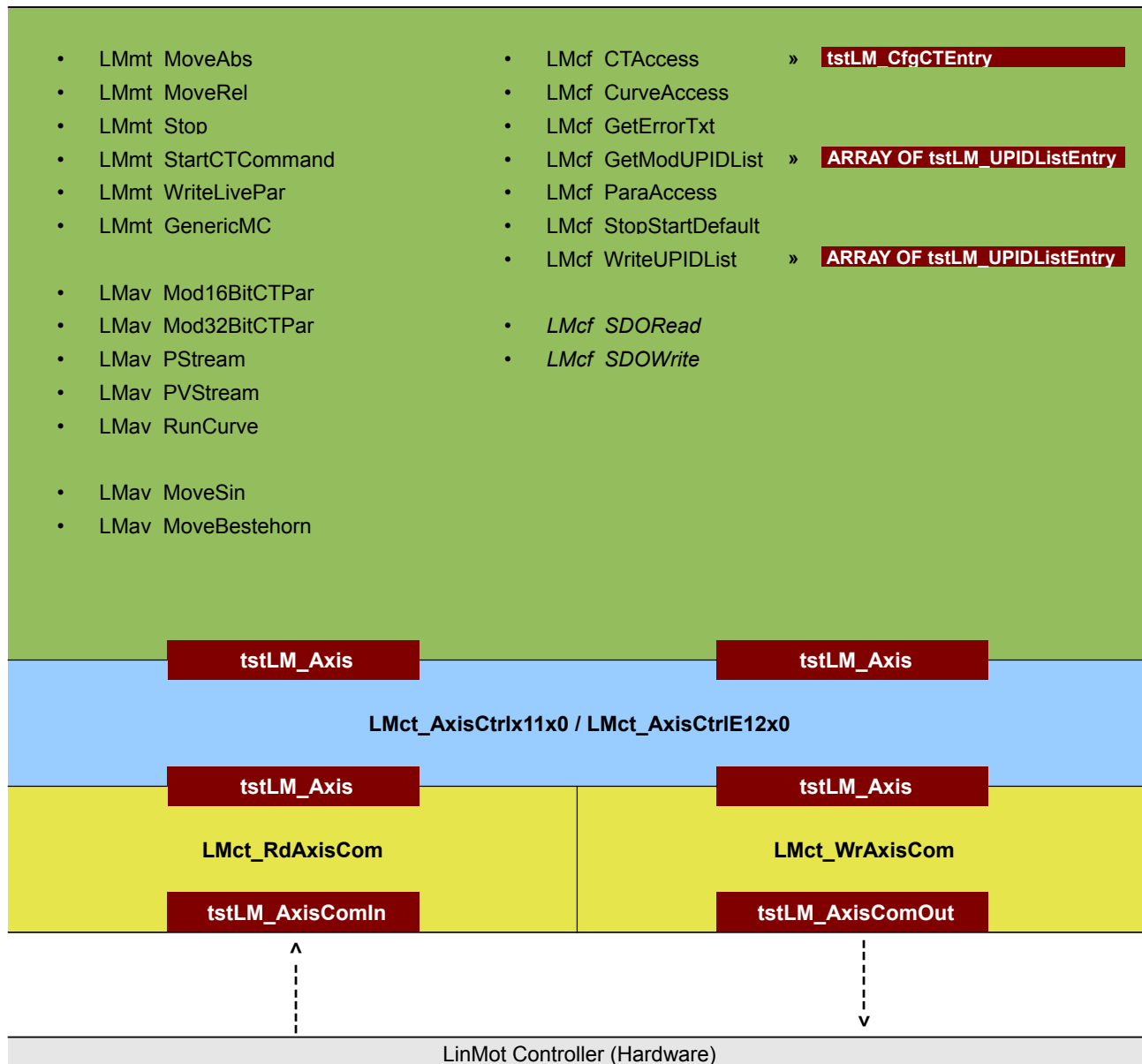
Nur für E1250-EC-UC & E1130-DP-xx Controller!

```
TYPE LM_stConfigUPIDList:
STRUCT
    (*Array size may have to be increased depending of the number of UPIDs used*)
    UPIDListEntries :ARRAY[0..2000] OF LM_stConfigUPIDListEntry;
END_STRUCT
END_TYPE
```

Abbildung 16: tstLM_CfgUPIDListEntry

3. Bausteine

3.1 Übersicht und Abhängigkeiten



3.2 IO und Achskontrolle

3.2.1 LMct_RdAxisCom

Der Baustein LMct_RdAxisCom liest die Eingangsdaten und befüllt die Achsstruktur. Sollte möglichst am Anfang des Programms aufgerufen werden.

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)
- B1100-GP-xx (CANopen)

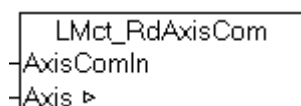


Abbildung 17: LMct_RdAxisCom

Eingänge		
Bezeichnung	Datentyp	Beschreibung
AxisComIn	tstLM_AxisComIn	Feldbus Eingangsdaten
Axis	tstLM_Axis	Achsreferenz (IN_OUT)

3.2.2 LMct_WrAxisCom

Der Baustein LMct_WrAxisCom liest die Ausgangsdaten der Achsstruktur (tstLM_Axis), bereitet sie auf und schreibt sie in die Ausgangsdatenstruktur (tstLM_AxisComOut). Sollte möglichst am Ende des Programms aufgerufen werden. Zumindest nach allen andere Bausteinen dieser Bibliothek.

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)
- B1100-GP-xx (CANopen)

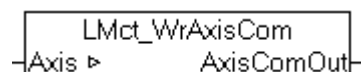


Abbildung 18: LMct_WrAxisCom

Eingänge		
Bezeichnung	Datentyp	Beschreibung
Axis	tstLM_Axis	Achsreferenz (IN_OUT)

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
AxisComOut	tstLM_AxisComOut	Feldbus Ausgangsdaten

3.2.3 LMct_AxisCtrlE12x0

Mit diesem Baustein lässt sich die Zustandsmaschine eines LinMot Controller steuern. Über die Ausgänge des Bausteins kann der Status des Controllers abgefragt werden.

Unterstützte Controller:

- Für E1250-EC-UC & E1230-DP-UC

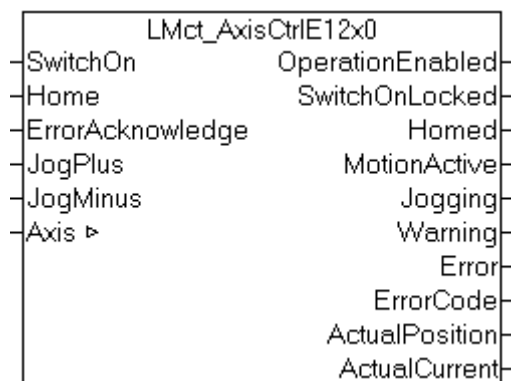


Abbildung 19: LMct_AxisCtrlE12x0

3.2.4 LMct_AxisCtrlx11x0

Mit diesem Baustein lässt sich die Zustandsmaschine eines LinMot Controller steuern. Über die Ausgänge des Bausteins kann der Status des Controllers abgefragt werden.

Unterstützte Controller:

- Für E1130-DP-xx, E1100-CO-xx, B1100-GP-xx

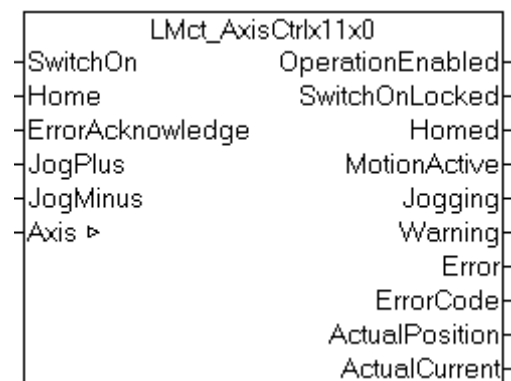


Abbildung 20: LMct_AxisCtrlx11x0

3.2.5 Ein- und Ausgänge der Achskontrollbausteine (LMct_AxisCtrlE12x0 & LMct_AxisCtrlx11x0)

Bei beiden Achskontrollbausteinen stehen die selben Ein- und Ausgänge zu Verfügung.
Deren Funktion ist identisch.

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
SwitchOn	Bool		Achse einschalten
Home	Bool		Achse referenzieren (muss anliegen bis Ausgang „Homed“ TRUE ist)
ErrorAcknowledge	Bool		Fehlerbestätigung auf steigende Flanke
JogPlus	Bool		Tippbetrieb positiv
JogMinus	Bool		Tippbetrieb negativ

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
OperationEnabled	Bool	Achse ist bestromt und bereit für Fahrbefehle
SwitchOnLocked	Bool	Anfahrsperr aktiv (-> SwitchOn auf FALSE setzen)
Homed	Bool	Achse hat Referenz
MotionActive	Bool	Sollwertgenerierung aktiv (VAI- / Kurven-Interpolator)
Jogging	Bool	Achse bewegt sich im Tippbetrieb
Warning	Bool	Eine Warnung liegt an
Error	Bool	Ein Fehler liegt an
ErrorCode	UINT	Ist ein Fehler aufgetreten, wird an diesem Ausgang der Fehler Code angezeigt. (Siehe Handbuch „Motion Control SW“)
ActualPosition	Real	Aktuelle Position der Achse in mm
ActualCurrent	Real	Aktueller Motorstrom in A (Ampère)

**Achtung**

Alle IO- und Achskontrollbausteine (Kapitel 3.2.1 - 3.2.4) müssen zyklisch aufgerufen werden!

3.3 MC Bausteine (E12x0, E11x0, B1100-GP Controller)

3.3.1 LMmt_MoveAbs

Mit diesem Baustein kann eine absolute Position mit anpassbarer Geschwindigkeit, Beschleunigung und Verzögerung angefahren werden.

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)
- B1100-GP (CANopen)

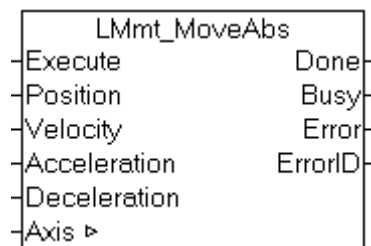


Abbildung 21: LMmt_MoveAbs

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
Position	Real		Zielposition in [mm]
Velocity	Real		Maximale Geschwindigkeit in [m/s]
Acceleration	Real		Beschleunigung in [m/s²]
Deceleration	Real		Verzögerung in [m/s²]

Tabelle 1: Eingänge LMmt_MoveAbs

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet und Achse an Zielposition
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 2: Ausgänge LMmt_MoveAbs

3.3.2 LMmt_MoveRel

Mit diesem Baustein kann die aktuelle Sollposition mit anpassbarer Geschwindigkeit, Beschleunigung und Verzögerung inkrementiert werden.

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)
- B1100-GP (CANopen)

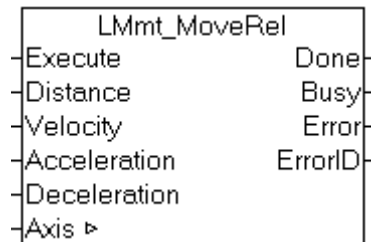


Abbildung 22: LMmt_MoveAbs

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
Distance	Real		Positionsinkrement in [mm]
Velocity	Real		Maximale Geschwindigkeit in [m/s]
Acceleration	Real		Beschleunigung in [m/s ²]
Deceleration	Real		Verzögerung in [m/s ²]

Tabelle 3: Eingänge LMmt_MoveAbs

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet und Achse an Zielposition
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 4: Ausgänge LMmt_MoveAbs

3.3.3 LMmt_StartCTCommand

Mit diesem Baustein kann eine Zeile der Command Table (Im Controller gespeichert) ausgeführt werden.

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)
- B1100-GP (CANopen)

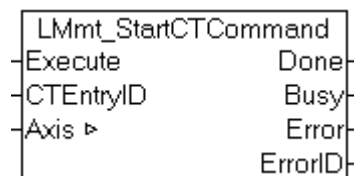


Abbildung 23: LMmt_StartCTCommand

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
CTEntryID	UINT	1...255 (31)	ID der Zeile der Command Table (Achtung: Auf B1100-GP-xx max. 31)

Tabelle 5: Eingänge LMmt_StartCTCommand

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 6: Ausgänge LMmt_StartCTCommand

3.3.4 LMmt_Stop

Dieser Baustein leitet einen sofortigen Stopp der Achse mit der angegebenen Verzögerung ein. Allenfalls durch aktive MC Bausteine laufende Fahrbefehle werden abgebrochen!

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)
- B1100-GP (CANopen)

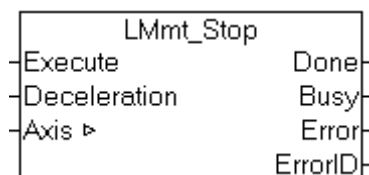


Abbildung 24: LMmt_Stop

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
Deceleration	Real		Verzögerung in [m/s ²]

Tabelle 7: Eingänge LMmt_Stop

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet und Achse steht
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 8: Ausgänge LMmt_Stop

3.3.5 LMmt_WriteLivePar

Mit diesem Baustein kann ein Live-Parameter des Controllers geschrieben werden.
(Live-Parameter sind während der Laufzeit veränderbare Parameter)

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)
- B1100-GP (CANopen)

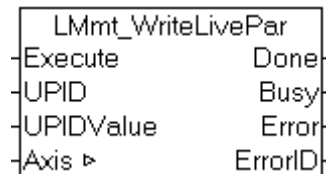


Abbildung 25: LMmt_WriteLivePar

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
UPID	UINT		Parameter Adresse (Unique Parameter ID)
UPIDValue	DINT		Parameterwert

Tabelle 9: Eingänge LMmt_WriteLivePar

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 10: Ausgänge LMmt_WriteLivePar

3.3.6 LMmt_GenericMC

Mit diesem Baustein lassen sich alle möglichen Befehle (des verwendeten Controllers) ausführen. Die Parameter müssen entsprechend des gewählten MCHeader skaliert werden! Eine Liste aller Befehle befindet sich im Handbuch "Motion Control Software".

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)
- B1100-GP (CANopen)

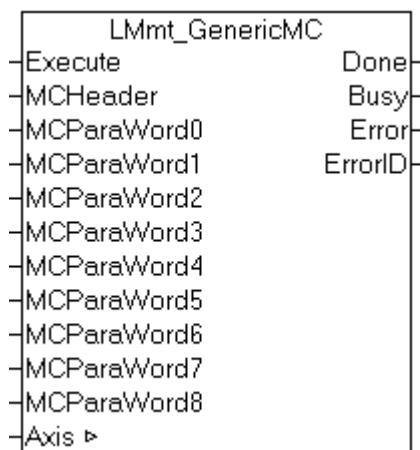


Abbildung 26: LMmt_GenericMC

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
MCHeader	UINT		Befehlsskopf (Befehlsnummer)
MCParaWord0	UINT		0. Parameterwort
MCParaWord1	UINT		1. Parameterwort
MCParaWord2	UINT		2. Parameterwort
MCParaWord3	UINT		3. Parameterwort
MCParaWord4	UINT		4. Parameterwort
MCParaWord5	UINT		5. Parameterwort
MCParaWord6	UINT		6. Parameterwort
MCParaWord7	UINT		7. Parameterwort
MCParaWord8	UINT		8. Parameterwort

Tabelle 11: Eingänge LMmt_GenericMC

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 12: Ausgänge LMmt_GenericMC

3.4 MC Bausteine (E12x0, E11x0 Controller)

3.4.1 LMav_Mod16BitCTPar

Mit diesem Baustein lässt sich der Wert eines Parameters (16Bit) in der Command Table verändern (Nur RAM).

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)

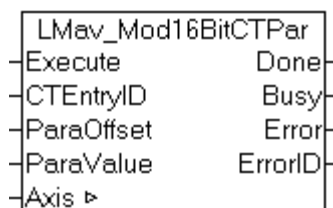


Abbildung 27: LMav_Mod16BitCTPar

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
CTEntryID	UINT		Zeilennummer der Command Table
ParaOffset	UINT		Offset des zu schreibenden Parameters
ParaValue	INT		Wert des zu schreibenden Parameters

Tabelle 13: Eingänge LMav_Mod16BitCTPar

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 14: Ausgänge LMav_Mod16BitCTPar



Hinweis

Weitere Informationen zum Befehl sind im Handbuch "Motion Control Software" zu finden. (Empfohlene Dokumente)

3.4.2 LMav_Mod32BitCTPar

Mit diesem Baustein lässt sich der Wert eines Parameters (32Bit) in der Command Table verändern (Nur RAM).

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen)

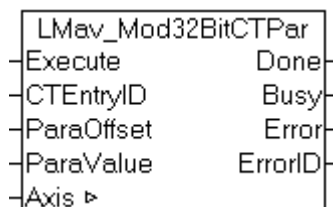


Abbildung 28: LMav_Mod32BitCTPar

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
CTEntryID	UINT		Zeilennummer der Command Table
ParaOffset	UINT		Offset des zu schreibenden Parameters
ParaValue	DINT		Wert des zu schreibenden Parameters

Tabelle 15: Eingänge LMav_Mod32BitCTPar

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 16: Ausgänge LMav_Mod32BitCTPar



Hinweis

Weitere Informationen zum Befehl sind im Handbuch "Motion Control Software" zu finden.
(Empfohlene Dokumente)

3.4.3 LMAV_RunCurve

Mit diesem Baustein lässt sich ein auf dem Controller gespeichertes Fahrprofil (Kurve) starten.

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- x11x0 Serie (Profibus oder CANopen)

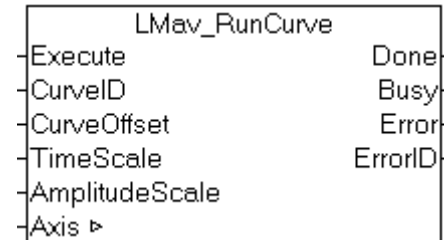


Abbildung 29: LMAV_RunCurve

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
CurveID	UINT	1...99	Kurvennummer (ID)
CurveOffset	Real		Offset der Kurve
TimeScale	Real	0...200	Zeitskalierung der Kurve in [%]
AmplitudeScale	Real	-2000...+2000	Amplitudenskalierung der Kurve in [%]

Tabelle 17: Eingänge LMAV_RunCurve

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet und Achse an Zielposition
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 18: Ausgänge LMAV_RunCurve



Hinweis

Weitere Informationen zum Befehl sind im Handbuch "Motion Control Software" zu finden. (Empfohlene Dokumente)

3.5 MC Bausteine (E12x0 Controller)

3.5.1 LMav_MoveBestehorn

Mit diesem Baustein lässt sich eine Zielposition mit Bestehorn-Profil anfahren.

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen mit installierter „Sinoide“ Applikation)

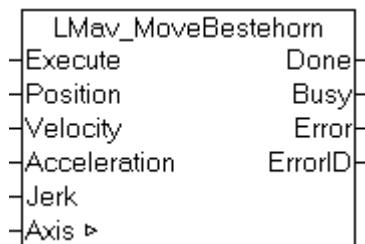


Abbildung 30: LMav_MoveBestehorn

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
Position	Real		Zielposition in [mm]
Velocity	Real		Maximale Geschwindigkeit in [m/s]
Acceleration	Real		Beschleunigung in [m/s^2]
Jerk	Real		Ruck in [m/s^3]

Tabelle 19: Eingänge LMav_MoveBestehorn

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet und Achse an Zielposition
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 20: Ausgänge LMav_MoveBestehorn



Hinweis

Weitere Informationen zum Befehl sind im Handbuch „Motion Control Software“ zu finden.
(Empfohlene Dokumente)

3.5.2 LMav_MoveSin

Mit diesem Baustein lässt sich eine Zielposition mit Sinus-Profil anfahren.

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E11x0 Serie (Profibus oder CANopen mit installierter „Sinoide“ Applikation)

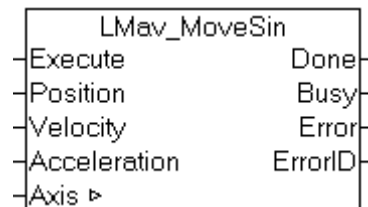


Abbildung 31: LMav_MoveSin

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
Position	Real		Zielposition in [mm]
Velocity	Real		Maximale Geschwindigkeit in [m/s]
Acceleration	Real		Beschleunigung in [m/s^2]

Tabelle 21: Eingänge LMav_MoveSin

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet und Achse an Zielposition
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 22: Ausgänge LMav_MoveSin



Hinweis

Weitere Informationen zum Befehl sind im Handbuch "Motion Control Software" zu finden. (Empfohlene Dokumente)

3.6 Config Bausteine (E12X0, E1130-DP)

Mit den Config Bausteinen ist der Zugriff auf Parameter, Kurven und die Command Table eines LinMot Controllers möglich. Ausserdem stellen Sie Funktionen wie Neustart oder Stop der Firmware oder Teilen davon, lesen von Fehlerbeschreibungen (STRING) und Rücksetzen von einzelnen Firmwareschichten auf Werkseinstellungen zur Verfügung.

Wichtig:

*Wenn mittels eines Bausteins Daten in den Flash-Speicher des Controller gesichert werden sollen, muss die Firmwareschicht MC_SW gestoppt sein. Dies kann mit dem Baustein **LMcf_StopStartDefault** mit **Modus 5** durchgeführt werden. Mit **Modus 6** kann sie anschliessend wieder gestartet werden.*

Für folgende Modi und Bausteine ist dies erforderlich:

- **LMcf_CTAccess** **Modus 0**
- **LMcf_CurveAccess** **Modus 0**

Die in diesem Kapitel vorgestellten Bausteine sind kompatibel mit folgenden Controllern und Schnittstellen:

- E1250-EC-UC EtherCAT
- E1230-DP-UC Profibus
- E1130-DP-xx Profibus

3.6.1 LMcf_StopStartDefault

Mit diesem Baustein lässt sich die Firmware eines LinMot Controllers bzw. einzelne Firmwareschichten starten oder stoppen. Zusätzlich können die einzelnen Firmwareschichten auf Werkseinstellungen zurückgesetzt werden.

Unterstützte Controller:

- E12x0 Serie (EtherCAT oder Profibus)
- E1130-DP-xx (Profibus)



Abbildung 32: LMcf_StopStartDefault

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
Mode	UINT	0...6	Modus

Tabelle 23: Eingänge LMcf_StopStartDefault

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl ausgeführt
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 24: Ausgänge LMcf_StopStartDefault

Modus		
Wert	Benutzte Eingänge	Beschreibung
0	-	Restart Controller
1	-	Set parameter ROM values to default (OS SW)
2	-	Set parameter ROM values to default (MC SW)
3	-	Set parameter ROM values to default (Interface SW)
4	-	Set parameter ROM values to default (Application SW)
5	-	Stop MC and Application Software (for Flash Access)
6	-	Start MC and Application Software

Tabelle 25: Modi von LMcf_StopStartDefault



Achtung

Modus 5&6 sind wichtig im Zusammenspiel mit den Bausteinen LMcf_CTAccess und LMcf_CurveAccess. Bevor Kurven oder Command Table Einträge vom RAM in den Flash-Speicher geschrieben werden, **muss die MC_SW gestoppt sein!**

3.6.2 LMcf_CTAccess

Dieser Baustein bietet Zugriff auf die Command Table von LinMot Controllern. Lesen und schreiben von einzelnen Zeilen. Löschen einzelner Zeilen oder der gesamten Command Table. Sichern der Command Table vom RAM in den Flash-Speicher.

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E1130-DP-xx (Profibus)

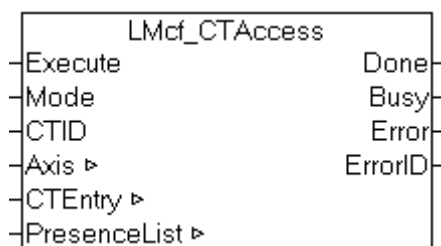


Abbildung 33: LMcf_CTAccess

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
Mode	UINT	0...5	Modus
CTID	UINT	1...255	Command Table ID (Zeilennummer)
CTEntry	tstLM_CfgCTEntry		Datenstruktur für eine CT Zeile (IN_OUT)
PresenceList	Array[0..7] of UDINT		Presence List (IN_OUT)

Tabelle 26: Eingänge LMcf_CTAccess

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl ausgeführt
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 27: Ausgänge LMcf_CTAccess

Modus		
Wert	Benutzte Eingänge	Beschreibung
0	-	Save to Flash: Sichert die Command Table im Flashspeicher MC_SW muss gestoppt sein!
1	-	Delete all Entries: Alle Zeilen der Command Table löschen (RAM)
2	CTID	Delete Entry: Einzelne Zeile der Command Table löschen (RAM)
3	CTEntry	Write Entry: Zeile in Command Table schreiben (ID ist in CTEntry) (RAM)
4	CTID, CTEntry	Get Entry: Zeile aus Command Table lesen (Wird in CTEntry gespeichert)
5	PresenceList	Die PresenceList wird aus dem Controller geladen

Tabelle 28: Modi von LMcf_CTAccess

3.6.3 LMcf_CurveAccess

Dieser Baustein bietet Zugriff auf die in LinMot Controllern hinterlegten Fahrprofile (Kurven). Lesen und schreiben von einzelnen Kurven. Löschen aller Kurven im RAM. Sichern aller Kurven vom RAM in den Flash-Speicher. Das Fahrprofil wird in einem ARRAY OF UDINT gespeichert.

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E1130-DP-xx (Profibus)

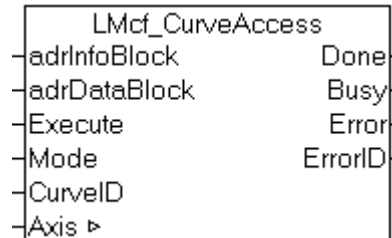


Abbildung 34: LMcf_CurveAccess

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
adrInfoBlock	POINTER TO ARRAY [0-18] OF UDINT		Zeiger auf ein Array von UDINT
adrDataBlock	POINTER TO ARRAY OF UDINT		Zeiger auf ein Array von UDINT
Execute	Bool		Befehlsauslösung (steigende Flanke)
Mode	UINT	0...4	Modus
CurveID	UINT	1...99	Kurvennummer

Tabelle 29: Eingänge LMcf_CurveAccess

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl ausgeführt
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 30: Ausgänge LMcf_CurveAccess

Modus		
Wert	Benutzte Eingänge	Beschreibung
0	-	Save all Curves from RAM to Flash: Sichert alle Kurven in den Flash-Speicher. MC_SW muss gestoppt sein!
1	-	Delete all Curves: Löscht alle Kurven (RAM)
2	adrInfoBlock, adrDataBlock	Add Curve: Einzelne Kurve schreiben (RAM)
3	adrInfoBlock, adrDataBlock	Modify Curve: Kurve auf dem Controller modifizieren (RAM)
4	CurveID, adrInfoBlock, adrDataBlock	Get Curve: Einzelne Kurve aus dem Controller lesen (Wird im Array an „adrInfoBlock“ bzw. „adrDataBlock“ gespeichert)

Tabelle 31: Modi von LMcf_CurveAccess



Achtung

Das ARRAY OF UDINT muss genügend gross sein, um die Kurve zu speichern!
 Aufrufbeispiel: LMcf_CurveAccess(adrInfoBlock:= ADR(Axis_A_Curve[0]),
 adrDataBlock:= ADR(Axis_A_Curve[20]),
 Axis:= Axis_A_Axis);

3.6.4 LMcf_ParaAccess

Dieser Baustein bietet Zugriff auf die Parameter eines LinMot Controllers. Lesen und schreiben von RAM und ROM Werten. Lesen von minimal, maximal und default Werten der Parameter.

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E1130-DP-xx (Profibus)

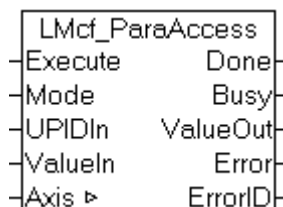


Abbildung 35: LMcf_ParaAccess

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
Axis	tstLM_Axis		Achsreferenz (IN_OUT)
Execute	Bool		Befehlsauslösung (steigende Flanke)
Mode	UINT	0...7	Modus
UPIDIn	UINT		Parameternummer (UPID)
ValueIn	DINT		Wert welcher geschrieben werden soll

Tabelle 32: Eingänge LMcf_ParaAccess

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl ausgeführt
Busy	Bool	Befehl aktiv
ValueOut	DINT	Gelesener Wert / Rückmeldung geschriebener Wert
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 33: Ausgänge LMcf_ParaAccess

Modus		
Wert	Benutzte Eingänge	Beschreibung
0	UPIDIn	Read ROM Value of Parameter by UPID
1	UPIDIn	Read RAM Value of Parameter by UPID
2	UPIDIn, ValueIn	Write ROM Value of Parameter by UPID
3	UPIDIn, ValueIn	Write RAM Value of Parameter by UPID
4	UPIDIn, ValueIn	Write RAM and ROM Value of Parameter by UPID
5	UPIDIn	Get minimal Value of Parameter by UPID
6	UPIDIn	Get maximal Value of Parameter by UPID
7	UPIDIn	Get default Value of Parameter by UPID

Tabelle 34: Modi von LMcf_ParaAccess

3.6.5 LMcf_GetModUPIDList

Dieser Baustein liest sämtliche Parameter und deren Werte welche im Vergleich zur Werkseinstellung verändert wurden aus dem Controller. Damit lässt sich der Parametersatz eines Controllers auf der SPS sichern und gegebenenfalls wieder runter schreiben.

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E1130-DP-xx (Profibus)



Abbildung 36: LMcf_GetModUPIDList

Eingänge		
Bezeichnung	Datentyp	Beschreibung
Axis	tstLM_Axis	Achsreferenz (IN_OUT)
Execute	Bool	Befehlsauslösung (steigende Flanke)
adrFirstEntry	POINTER TO tstLM_CfgUPIDListEntry	Zeiger auf den ersten Eintrag eines Arrays von tstLM_CfgUPIDListEntry
NrOfEntries	UINT	Anzahl der zu lesenden Parameter (>0)

Tabelle 35: Eingänge LMcf_GetModUPIDList

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet / UPID List gelesen
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 36: Ausgänge LMcf_GetModUPIDList



Achtung

Das Array von tstLM_CfgUPIDListEntry muss genügend gross sein, um die an „NrOfEntries“ gewählte Anzahl von Parametern zu sichern!

Aufrufbeispiel: LMcf_GetModUPIDList(
 adrFirstEntry:= ADR(Axis_A_ModUPIDList[1]),
 Axis:=Axis_A_Axis
);

3.6.6 LMcf_WriteUPIDList

Dieser Baustein schreibt ein Array von Parametern (UPID und Wert) auf den Controller.

Wichtig:

Der Baustein beendet das Schreiben, sobald er einen Eintrag mit UPID = 0 findet, **oder** nach der Anzahl an „NrOfEntries“. D.h nach dem letzten Parameter im Array muss noch ein weiterer Eintrag mit UPID = 0 folgen, oder die Anzahl an „NrOfEntries“ exakt sein!

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E1130-DP-xx (Profibus)

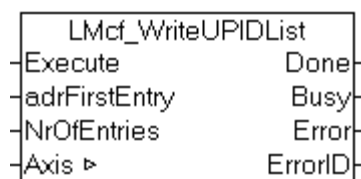


Abbildung 37: LMcf_WriteUPIDList

Eingänge		
Bezeichnung	Datentyp	Beschreibung
Axis	tstLM_Axis	Achsreferenz (IN_OUT)
Execute	Bool	Befehlsauslösung (steigende Flanke)
adrFirstEntry	POINTER TO tstLM_CfgUPIDListEntry	Zeiger auf den ersten Eintrag eines Arrays von tstLM_CfgUPIDListEntry
NrOfEntries	UINT	Anzahl der zu schreibenden Parameter (>0)

Tabelle 37: Eingänge LMcf_WriteUPIDList

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet / UPID List geschrieben
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 38: Ausgänge LMcf_WriteUPIDList

3.6.7 LMcf_GetErrorTxt

Dieser Baustein gibt anhand des anliegenden ErrorCode einen String mit dem Fehlertext zurück.

Unterstützte Controller:

- E12X0 Serie (EtherCAT oder Profibus)
- E1130-DP-xx (Profibus)

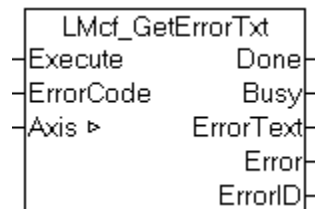


Abbildung 38: LMcf_GetErrorTxt

Eingänge		
Bezeichnung	Datentyp	Beschreibung
Axis	tstLM_Axis	Achsreferenz (IN_OUT)
Execute	Bool	Befehlsauslösung (steigende Flanke)
ErrorCode	UINT	Fehlernummer (Siehe Achskontrollbausteine)

Tabelle 39: Eingänge LMcf_GetErrorTxt

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet / Fehlertext gelesen
Busy	Bool	Befehl aktiv
ErrorText	String	Fehlertext als String
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Kapitel 4. Fehlerbeschreibungen)

Tabelle 40: Ausgänge LMcf_GetErrorTxt

3.7 Config Bausteine CANopen (E11x0, B1100-GP Controller)**3.7.1 LMcf_SDORead**

Importieren Sie SDOACCESS_X1X00.EXP ins Projekt, um die Bausteine zu benutzen.

Mit diesem Baustein lässt sich ein SDO read an einen LinMot Controller senden.

Unterstützte Controller:

- E11x0 Serie (CANopen)
- B1100-GP-xx (CANopen)

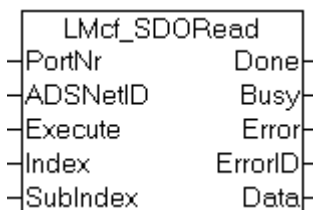


Abbildung 39: LMcf_SDORead

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
PortNr	UINT		AMS Portnummer (MACID des Controllers)
ADSNetID	String		ADSNetID des CANopen Masters
Execute	Bool		Befehlsauslösung (steigende Flanke)
Index	UINT		Object Dictionary Index
SubIndex	USINT		Object Dictionary SubIndex

Tabelle 41: Eingänge LMcf_SDORead

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet und Daten gelesen
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Beckhoff ADS Errors)
Data	DINT	Gelesene Daten

Tabelle 42: Ausgänge LMcf_SDORead

**Hinweis**

Zusätzliche Informationen sowie das Objektverzeichnis sind im Handbuch "CANopen Interface" zu finden. (Empfohlene Dokumente)

3.7.2 LMcf_SDOWrite

Importieren Sie SDOACCESS_X1X00.EXP ins Projekt, um die Bausteine zu benutzen.

Mit diesem Baustein lässt sich ein SDO write an einen LinMot Controller senden.

Unterstützte Controller:

- E11x0 Serie (CANopen)
- B1100-GP-xx (CANopen)

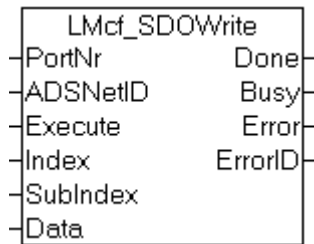


Abbildung 40: LMcf_SDOWrite

Eingänge			
Bezeichnung	Datentyp	Wertebereich	Beschreibung
PortNr	UINT		AMS Portnummer (MACID des Controllers)
ADSNetID	String		ADSNetID des CANopen Masters
Execute	Bool		Befehlsauslösung (steigende Flanke)
Index	UINT		Object Dictionary Index
SubIndex	USINT		Object Dictionary SubIndex
Data	UINT		Zu schreibende Daten

Tabelle 43: Eingänge LMcf_SDOWrite

Ausgänge		
Bezeichnung	Datentyp	Beschreibung
Done	Bool	Befehl gesendet und Daten geschrieben
Busy	Bool	Befehl aktiv
Error	Bool	Fehler in Baustein aufgetreten
ErrorID	UINT	Fehlernummer (Siehe Beckhoff ADS Errors)

Tabelle 44: Ausgänge LMcf_SDOWrite



Hinweis

Zusätzliche Informationen sowie das Objektverzeichnis sind im Handbuch "CANopen Interface" zu finden. (Empfohlene Dokumente)

4. Fehlerbeschreibungen

4.1 Fehler der Achskontrollbausteine (ErrorCode)

Eine Liste möglicher Fehler ist im Handbuch „Motion Control SW“ bzw. den entsprechenden Interface Handbüchern zu finden (Empfohlene Dokumente).

4.2 Fehlernummern der MC Bausteine

FehlerNr.	Fehlertext	Beschreibung
01h	Axis not ready	Die Achse ist nicht bereit für einen Fahrbefehl. Am Achskontrollbaustein prüfen, ob „Operation Enabled“-Ausgang TRUE ist.
02h	Axis already has command running	Die Achse hat bereits einen laufenden Befehl. Prüfen, ob ein anderer MC Baustein aktiv (busy) ist. Hinweis: Durch zurücksetzen des SwitchOn-Eingang am Achskontrollbaustein wird auch das CommandRunning Flag in der Achsreferenz zurückgesetzt.
03h	Axis has error	Die Achse hat einen Fehler. ErrorCode an Achskontrollbaustein prüfen
04h	Command interrupted	Der Befehl wurde unterbrochen (Die Achse ist nicht mehr „Operation Enabled“)
05h	Command aborted	Der Befehl wurde abgebrochen (z.B. Durch den Baustein LinMotFB_Stop)

Tabelle 45: Fehlernummern MC Bausteine

4.3 Fehlernummern der Config Bausteine

FehlerNr.	Fehlertext	Beschreibung
01h	TimeOut (No response from controller)	Der Controller antwortet nicht in der geforderten Zeit. Busverbindung prüfen.
02h	ConfigChannel already busy	Der Konfigurationskanal ist bereits von einem anderen Config-Baustein belegt. Prüfen, ob ein anderer Config-Baustein beschäftigt (busy) ist.
03h	Invalid Mode selected	Ungültiger Modus gewählt. Eingang Mode prüfen
04h	Address of adrDataBlock or adrInfoBlock is invalid (0), check inputs (CurveAccess)	Ungültige Adresse der Eingänge „adrDataBlock“ oder „adrInfoBlock“. Eingänge prüfen
05h	NrOfEntries must be greater than 0 (zero)! (UPID List)	Die Anzahl zu lesender/schreibender Einträge muss grösser als Null sein
06h	Address of UPIDListEntry array is zero! Check adrFirstEntry input! (UPID List)	Ungültige Adresse am Eingang „adrFirstEntry“. Eingang prüfen
C0h	UPID error	
C1h	Parameter Type Error	
C2h	Range Error	Der zu schreibende Werte ist ausserhalb des gültigen Bereichs für den gewählten Parameter
C3h	Address Usage Error	Es wird versucht auf einen „read only“ Parameter schreibend zuzugreifen.
C5h	Error: Command 21h	
D0h	Odd Address	
D1h	Size Error (CurveAccess)	
D4h	Curve already defined / Curve not present (CurveAccess)	
>D4h		Kontaktieren Sie den Support

Tabelle 46: Fehlernummern Config Bausteine

Kontakt**SWITZERLAND****NTI AG**

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)

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/>

Bitte besuchen Sie <http://www.linmot.com/> um einen Distributor in Ihrer Nähe zu finden.

Smart solutions are...

