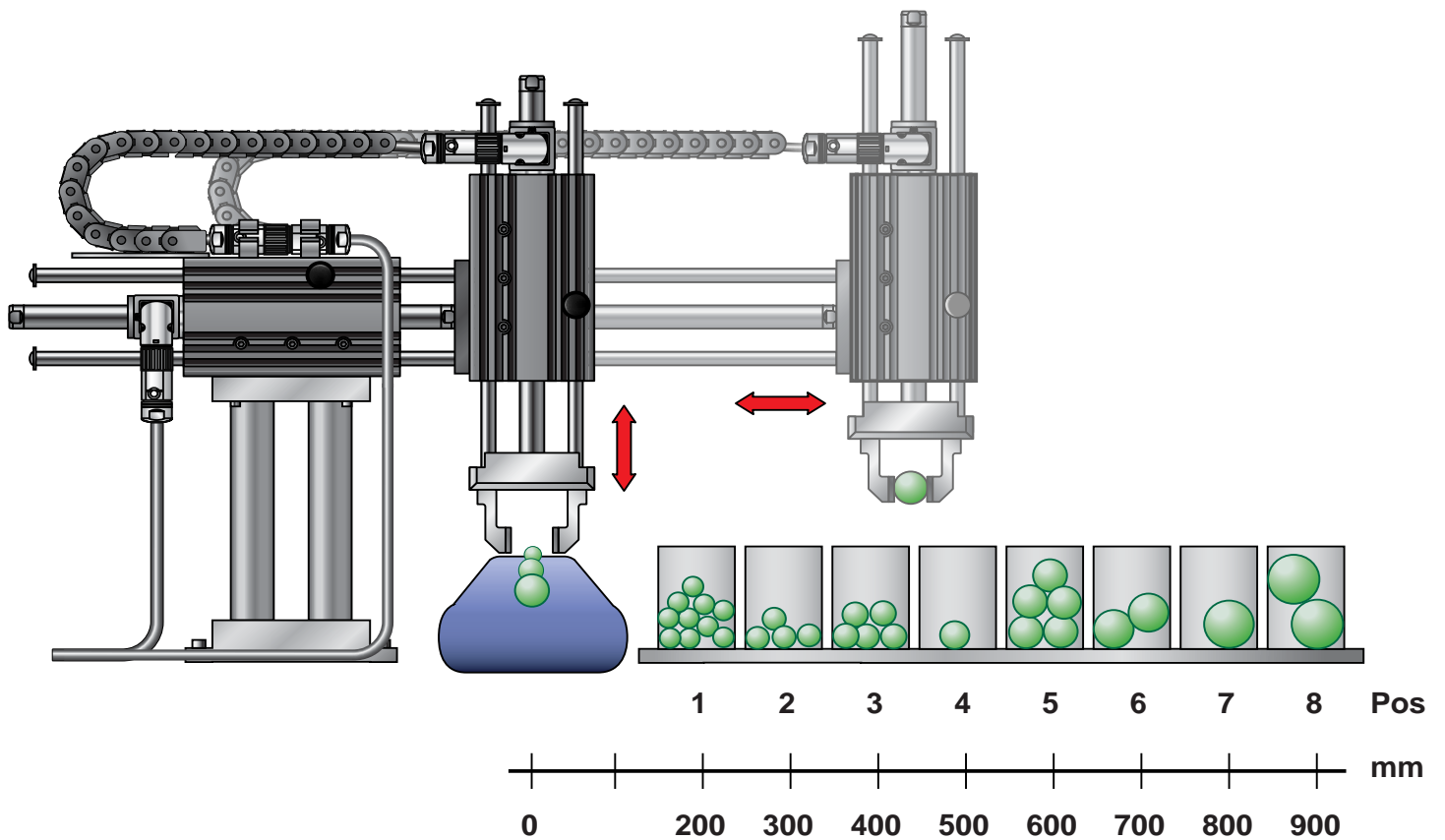


# LinMot®



## Anwendungsbeispiele

## **Impressum**

Produktion: Ernst Blumer  
Redaktion: Florian Jörg  
Gestaltung: Matthias Furler  
Druck: NTI AG  
Erstellt: Juni 2007  
Rechte: NTI AG

## Einführung

a. Parametrierung mit LinMot-Talk1100.....	2
c. Steuerung der Zustandsmaschine mittels digitaler Ein- und Ausgänge an X4 .....	3
b. Control- und StatusWord .....	3

## Applikationsbeispiele

1. Schieber mit zwei Endpositionen.....	5
2. Verfahren mit Bewegungsprofilen .....	7
3. Analog Position Mode .....	9
4. Indexing Mode (Step/Direction/Zero) .....	10
5. Anfahren von beliebigen Endpositionen über serielle Schnittstellen oder Feldbusse.....	11
6. Anfahren von 8 Positionen via digitaler I/O's.....	15
7. Ablaufsteuerung .....	16
8. Hochpräzises Positionieren.....	18
9. Betrieb mit externem absolut Sensor .....	19
10. Ansteuerung einer Bremse.....	20
11. Sichere Impulssperre .....	21
12. Parallelbetrieb von zwei Motoren.....	22
13. Kraftverdoppelung durch Master Booster .....	23
14. Auswertung von Endlagen und Referenzschaltern.....	24
15. Pressen mit definierter Kraft .....	25
16. Kraftregelung mit 0.1 N Auflösung .....	27
17. Einbindung von rotativen Motoren .....	28
18. Synchronisation zur Königswelle: Master Encoder 0° bis 360° .....	29
19. Synchronisation zur Königswelle: CAM1/CAM2.....	30

## a. Parametrierung mit LinMot-Talk1100

### Motorkonfiguration mittels Motor Wizard

Wird ein neuer Controller in Betrieb genommen, muss zuerst der angeschlossene Motor konfiguriert werden. LinMot-Talk1100 bietet hierzu den

Motor Wizard an, welcher den Benutzer Schritt für Schritt durch die Grundeinstellungen des Motors führt.



Der Motor Wizard ist über das Zauberstab-Symbol aufrufbar

### Konfiguration der Applikationsparameter

Sämtliche Parameter der LinMot Firmware können mit der LinMot-Talk1100 Software je nach Anforderungen der Anwendung angepasst werden. Jedem Parameter ist eine eindeutige Identifikationsnummer zugeordnet. Diese Nummer,

UPID (Unique Parameter ID) genannt, ist eine 16 Bit Integer Zahl. Sie wird im hexadezimal Format angegeben.

Über die Suchfunktion "Find UPID" (Menüleiste "Search -> Find UPID" oder Tastenkombination

"Ctrl + F") kann der betreffende Parameter anhand der UPID Nummer gesucht werden. Die UPID wird bei jedem Parameter in der Spalte "UPID" angezeigt (Siehe Abbildung).

Name	Value	Raw Data	UPID	Type	Scale
FF Constant Force	0 A	0000h	139Ch	SInt16	0.001 A
FF Friction	0 A	0000h	139Dh	SInt16	0.001 A
FF Spring Compensation	0 A/m	0000h	139Eh	SInt16	1 A/m
FF Damping	0 A/(m/s)	0000h	139Fh	SInt16	0.01 A/(m/s)
FF Acceleration	0 A/(m/s^2)	0000h	13A0h	UInt16	0.001 A/(m/...
Spring Zero Position	0 mm	00000000h	13A1h	SInt32	0.0001 mm
P Gain	1.5 A/mm	000Fh	13A2h	UInt16	0.1 A/mm
D Gain	3 A/(m/s)	001Eh	13A3h	UInt16	0.1 A/(m/s)
I Gain	0 A/(mm*s)	0000h	13A4h	UInt16	0.1 A/(mm*s)
Integrator Limit	0 A	0000h	13A5h	SInt16	0.001 A
Maximal Current	15 A	3A98h	13A6h	SInt16	0.001 A
Noise Deadband Width	0 mm	0000h	13A7h	UInt16	0.0001 mm

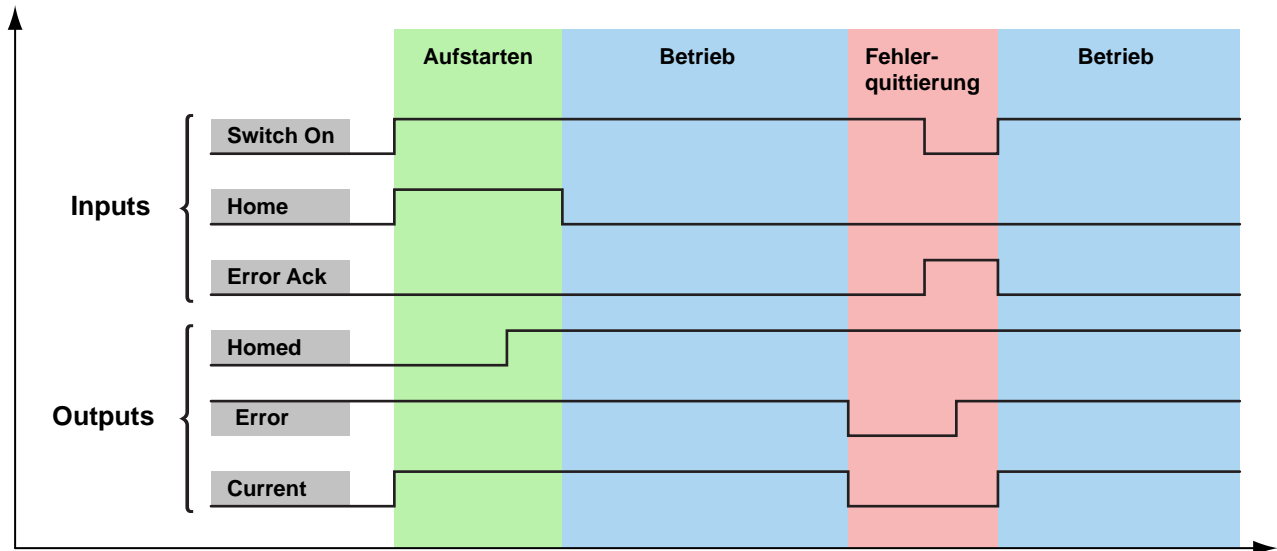
UPID Anzeige in LinMot-Talk1100

## b. Control- und StatusWord

Die Abbildung zeigt den Signalverlauf zum Bestromen und Referenzieren (Homing) des Motors, im normalen Betrieb sowie bei der

Quittierung eines aufgetretenen Fehlers. Sie zeigt ausserdem die wichtigsten Signale (Zustand der relevanten Bits im Control und Status

Word) zur Steuerung der Zustandsmaschine. Die Zustandsmaschine wird in Kapitel 3 des "Motion Control SW" Handbuches beschrieben.



Signale zur Steuerung der Zustandsmaschine

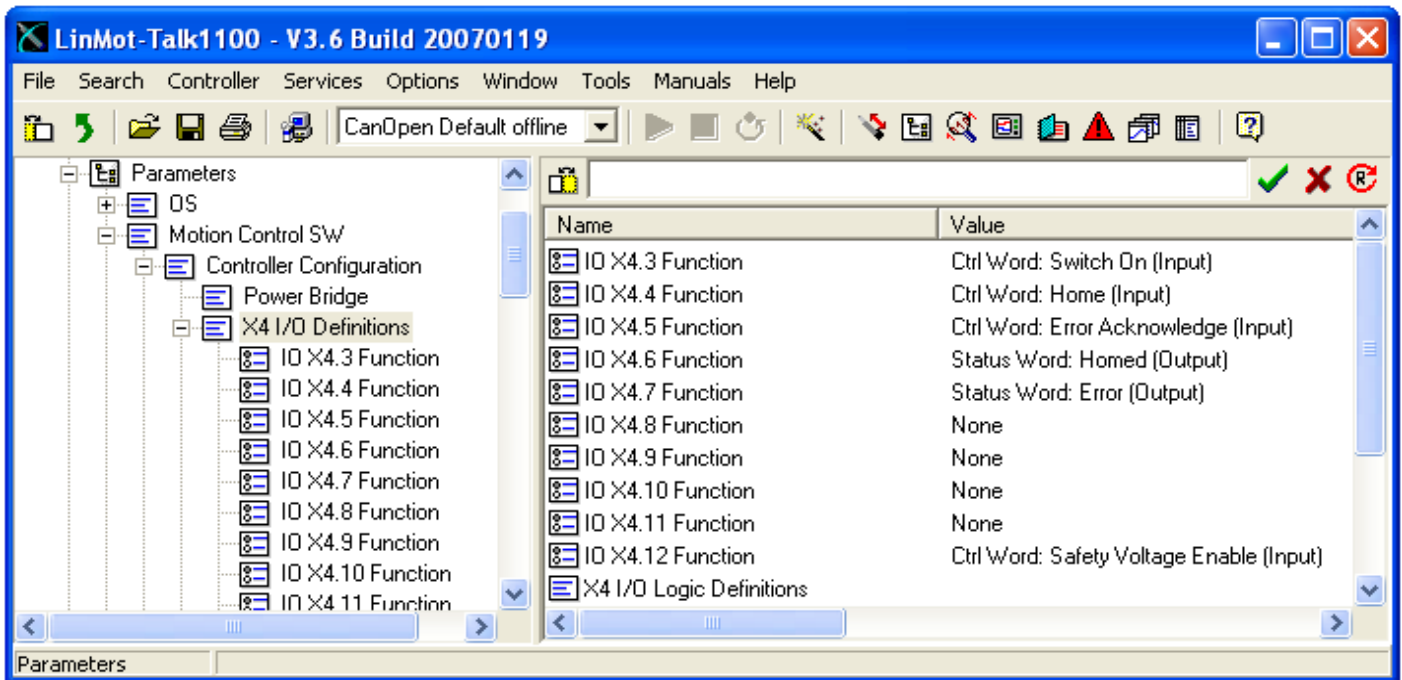
## c. Steuerung der Zustandsmaschine mittels digitaler Ein- und Ausgänge an X4

Die Steuersignale für die Zustandsmaschine können auf den Stecker X4 konfiguriert werden, um die Zustandsmaschine des Controllers über

digitale Signale zu steuern. Sämtliche Signale können beliebig auf X4.3 bis X4.11 gelegt werden. Nachfolgend eine Beispielkonfiguration.

Eingänge:				
Ctrl Word:	Switch On (Input)	On	UPID 1036h	Switch On auf X4.3
Ctrl Word:	Home (Input)	On	UPID 1037h	Home auf X4.4
Ctrl Word:	Error Acknowledge (Input)	On	UPID 1038h	Error Acknowledge auf X4.5

Ausgänge:				
Status Word:	Homed (Output)	On	UPID 1039h	Homed auf X4.6
Status Word:	Error (Output)	On	UPID 103Ah	Error auf X4.7



Steuersignale konfiguriert auf Stecker X4



### Hinweis

Die Controller der E1100 Serie (ausgenommen E1100-GP(-HC)) haben auf X4.12 den Safety Voltage Enable Eingang. An diesem muss für den Betrieb eine Spannung von +24 V anliegen.

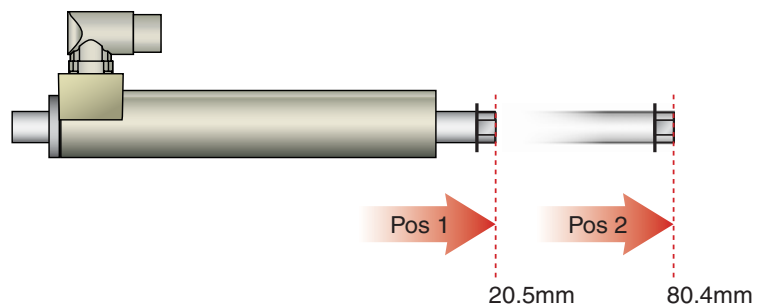
Ansonsten wird die PWM-Generierung der Leistungselektronik hardwaremässig nicht freigeschaltet. Die Logik (active High / active Low) der digitalen Ein- und Ausgänge wird unter "X4 I/O

Definitions (UPID 104Bh bis 1053h) konfiguriert. Damit kann der Errorausgang so konfiguriert werden, dass er im Betrieb High (+24V) und im Fehlerzustand Low (0V) ist.

## 1. Schieber mit zwei Endpositionen

### Aufgabe

Ein Produkt muss von Punkt A (Pos 1, 20.5mm) nach Punkt B (Pos 2, 80.4mm) transportiert werden. Die Ansteuerung der beiden Positionen wird über ein digitales Signal realisiert. Wenn der Motor eine der beiden Positionen erreicht, soll dies an einem digitalen Ausgang des Controllers signalisiert werden.



### Lösung

Für diese Aufgabe stellt LinMot den Run Mode "Triggered VA Interpolator" zur Verfügung. In diesem Modus können auf eine steigende bzw. fallende Flanke eines digitalen Triggersignals zwei Positionen angefahren werden. Sowohl

Geschwindigkeit wie auch die Beschleunigung können frei programmiert werden (VA Interpolator). Der VA Interpolator berechnet zur Laufzeit ein trapezförmiges Geschwindigkeits-Profil. Die Position 1 wird auf die fallende, Position 2 auf die

steigende Flanke des Triggersignals an X4.6 angefahren. Befindet sich der Antrieb an einer der beiden Positionen, wird dies an X4.5 signalisiert (als digitalen Ausgang konfigurieren).

#### Hardware Schnittstelle:

Das Triggersignal wird an X4.6 angeschlossen

#### RunMode konfigurieren:

	Triggered VA-Interpolator	On	UPID 1450h	Run Modus Triggered VA-Interpolator
--	---------------------------	----	------------	-------------------------------------

#### Konfiguration des Triggereingangs:

	Trigger (Input)	On	UPID 1039h	X4.6 als Trigger Eingang
	Direct	On	UPID 170Ch	Trigger Modus direkt

#### Konfiguration des Ausgangs

Status Word:	In Target Position	On	UPID 1038h	X4.5 als Ausgang für "In Target Position"
--------------	--------------------	----	------------	---

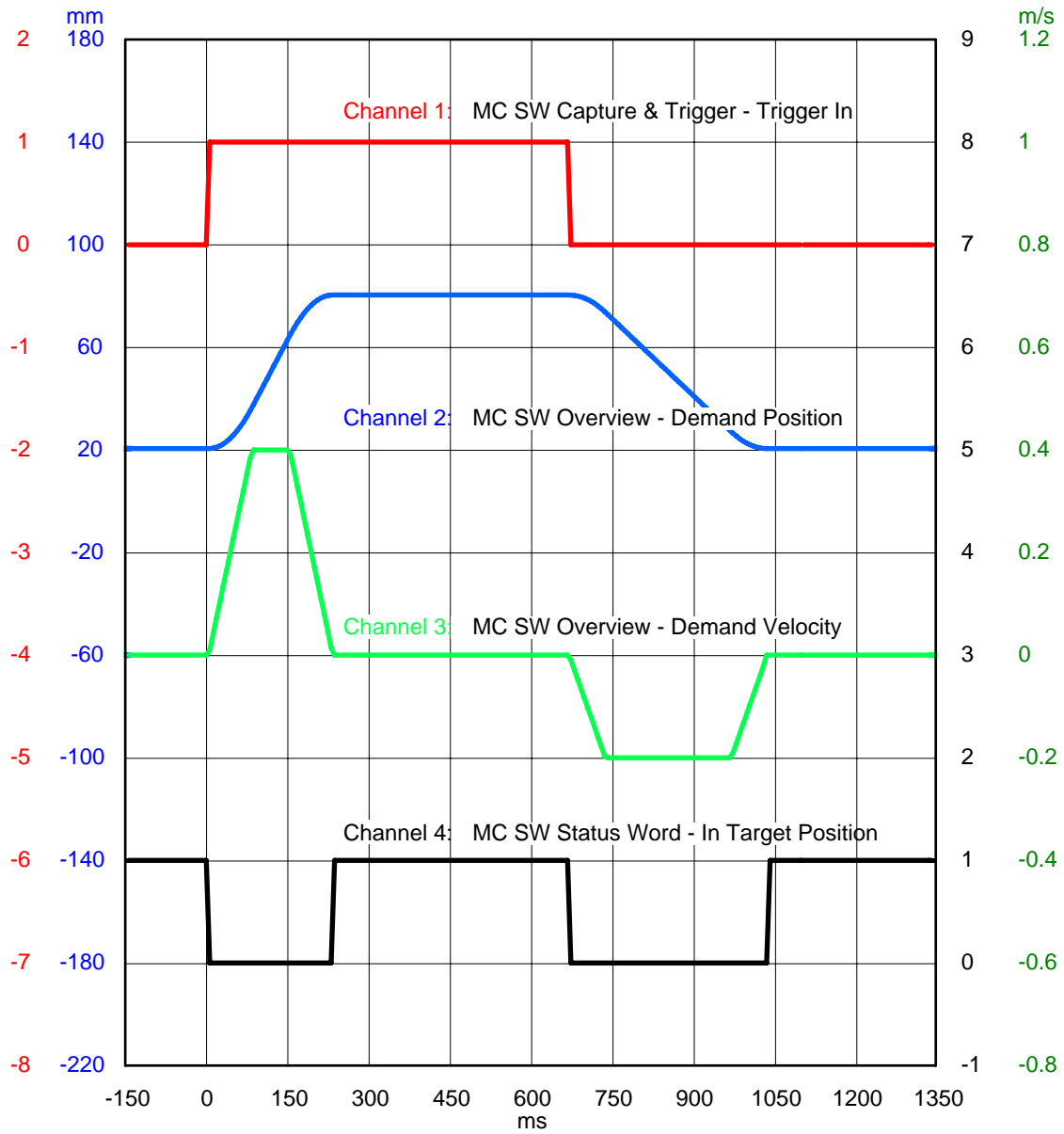
#### Konfiguration der Position 1:

	Position	20.5mm	UPID 145Ah	Position 1
	Max. Speed	0.2m/s	UPID 145Bh	Maximale Geschwindigkeit
	Acceleration	3m/s <sup>2</sup>	UPID 145Ch	Beschleunigung
	Deceleration	3m/s <sup>2</sup>	UPID 145Dh	Verzögerung

#### Konfiguration der Position 2:

	Position	80.4mm	UPID 145Fh	Position 2
	Max. Speed	0.2m/s	UPID 1460h	Maximale Geschwindigkeit
	Acceleration	5m/s <sup>2</sup>	UPID 1461h	Beschleunigung
	Deceleration	5m/s <sup>2</sup>	UPID 1462h	Verzögerung

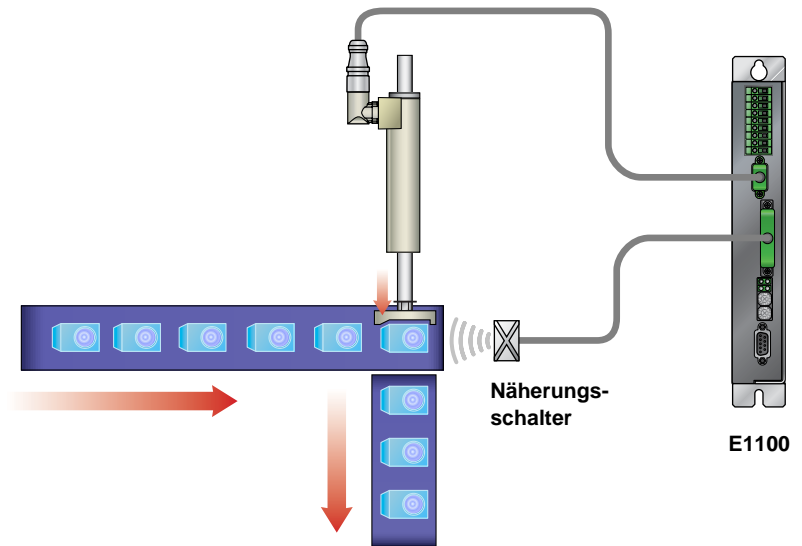
## Oszilloskop



## 2. Verfahren mit Bewegungsprofilen

### Aufgabe

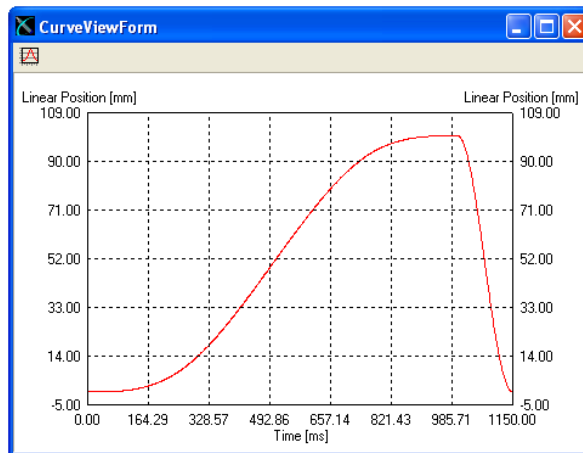
In einer Anlage werden Kontaktlinsen verpackt. Damit die Flüssigkeit nicht verschüttet wird, müssen sie ruckfrei (ruckminimiert) transportiert werden. Die Positionierung der Verpackung an ihren Zielpunkt (100mm) muss ruckfrei ausgeführt werden, wogegen die Rückfahrt zur Ausgangsposition (0mm) des Linearantriebs möglichst schnell erfolgen muss. Die Bewegung wird durch das Triggersignal eines Näherungsschalters ausgelöst.



### Lösung

Für diese Anwendung bietet sich der Run Mode "Triggered Time Curve" an. Dabei wird durch die steigende Flanke am Triggereingang ein im Controller gespeichertes Bewegungsprofil aufgerufen. Die Ausführung des Bewegungsprofils kann verzögert zum Triggersignal ausgelöst werden (Delayed). Dies erleichtert die Abstimmung zwischen der Auslösung des Näherungsschalters und dem Start der Bewegung.

Generieren des gewünschten Bewegungsprofils mithilfe des Kurvenservices (Handbuch LinMotTalk1100). Als ID der erstellten Kurve 1 wählen und sie auf den Controller runterladen.



Bewegungsprofil

#### Hardware Schnittstelle:

Das Triggersignal wird an X4.6 angeschlossen

#### RunMode konfigurieren:

Triggered Time Curve	On	UPID 1450h	Run Modus Triggered Time Curve
----------------------	----	------------	--------------------------------

#### Konfiguration des Triggereingangs:

Trigger (Input)	On	UPID 1039h	X4.6 als Trigger Eingang
Delayed	On	UPID 170Ch	Trigger Modus verzögert
Rise Delay Time	50ms	UPID 170Fh	Verzögerungszeit einstellen

#### Konfiguration des auszuführenden Bewegungsprofils:

Rise Curve ID	1	UPID 1482h	ID der Kurve einstellen
---------------	---	------------	-------------------------



## Hinweis

Die LinMot-Talk1100 Software unterstützt den Import von CSV Dateien (Excel). Damit können

mit Excel erstellte Bewegungsprofile in den Controller importiert werden (Siehe Abbildung).

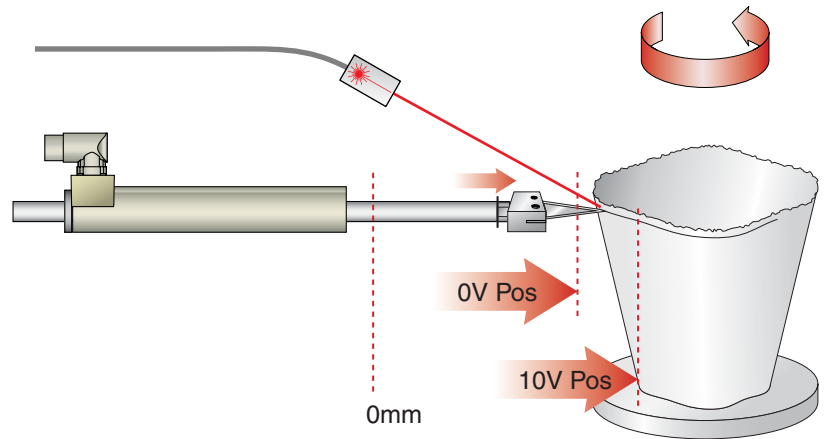
	A	B	C	D	E	F	G
1	0						
2	0.001						
3	0.0032						
4	0.0067						
5	0.0117						
6	0.0183						
7	0.0261						
8	0.0353						
9	0.0463						
10	0.0585						
11	0.072						
12	0.0872						
13	0.1038						
14	0.1216						
15	0.141						
16	0.162						
17	0.1842						
18	0.2077						
19	0.2331						
20	0.2598						
21	0.2876						
22	0.3171						
23	0.3481						
24	0.3804						
25	0.414						
26	0.4495						
27	0.4861						
28	0.5239						
29	0.5636						
30	0.6047						
31	0.6469						
32	0.6907						
33	0.7362						
34	0.7828						

In der Spalte A sind die Stützwerte eines Bewegungsprofils in mm angegeben. Die CSV-Datei kann im Kurvenservice von LinMot-Talk1100 importiert werden. Beim Erstellen einer neuen Kurve als "Setpoint Calculation Wizard" "FromFile..." wählen.

## 3. Analog Position Mode

### Aufgabe

Ein mittels Linearantrieb geführtes Messer muss Flaschen mit unregelmässiger Form aufschneiden. Das Messer soll dabei der Form der Flasche nachgeführt werden. Ein Distanzsensor misst den Abstand zur Oberfläche der Flasche und gibt dem Linearantrieb über ein analoges Signal von 0V (entspricht 20mm) bis 10V (entspricht 80mm) die Sollposition vor. Mittels eines zweiten, digitalen Signals soll der Antrieb an eine Warteposition (0mm) gefahren werden können.



### Lösung

Die Positionsvorgabe mittels analogem Eingangssignal wird vom Run Mode "Analog" unterstützt. Dabei positioniert sich der Linearmotor zwischen den frei konfigurierbaren 0V und 10V Positionen "proportional" zur Eingangsspannung

an X4.4. Um den Antrieb an eine Warteposition zu fahren, bietet sich die Funktion "Going To Position" an. Dazu wird ein digitales Signal an X4.3 angeschlossen. Erkennt der Controller an diesem Eingang einen hohen Pegel, wechselt er in

den Zustand 15: Going To Position und bewegt den Antrieb an die konfigurierte Position. Liegt an diesem Eingang ein tiefer Pegel an, ist der Controller im Zustand 8: Operation Enabled und folgt dem analogen Eingangssignal.

Hardware Schnittstelle:				
Das analoge Signal wird an X4.4 angeschlossen.				
Das digitale Signal wird an X4.3 angeschlossen.				
RunMode konfigurieren:				
	Analog	On	UPID 1450h	Run Modus Analog
Konfigurieren des analogen Eingangs:				
	Position	On	UPID 1790h	X4.4 als analoger Eingang
Konfigurieren des digitalen Eingangs:				
Ctrl Word:	Go To Position	On	UPID 1036h	X4.3 konfigurieren
Konfigurieren der Position für 0V:				
	0V Position	20mm	UPID 14D2h	Position für 0V konfigurieren
Konfigurieren der Position für 10V:				
	10V Position	80mm	UPID 14D3h	Position für 10V konfigurieren
Konfigurieren des vordefinierten VA-Interpolator (Dynamikbegrenzung des Analogmodus):				
	Max. Speed	1m/s	UPID 14Beh	Maximale Geschwindigkeit
	Acceleration	4m/s <sup>2</sup>	UPID 14BFh	Beschleunigung
	Deceleration	4m/s <sup>2</sup>	UPID 14C0h	Verzögerung
Konfiguration der Position und Dynamik der Warteposition (GoToPosition):				
	Position	0mm	UPID 1725h	Warteposition konfigurieren
	Max. Speed	0.1m/s	UPID 1726h	Maximale Geschwindigkeit
	Acceleration	1m/s <sup>2</sup>	UPID 1727h	Beschleunigung
	Deceleration	1m/s <sup>2</sup>	UPID 1728h	Verzögerung



### Hinweis

Ist das Signal an X4.3 hoch (24V), fährt der Linearmotor an die Position 0mm (Warteposition). Liegt keine Spannung an X4.3 an, folgt der Antrieb dem analogen Signal an X4.4. Für die ana-

loge Positionierung werden die Einstellungen des vordefinierten VA-Interpolators verwendet.

Eine Übersicht und Beschreibung der verschiedenen Zustände sind im Handbuch "Motion Control SW" zu finden.

## 4. Indexing Mode (Step/Direction/Zero)

### Aufgabe

In einer bestehenden Anwendung wurden bisher Schrittmotoren mit übergeordneter Schrittmotorsteuerung (Step/Direction/Zero) eingesetzt. Diese sollen nun aufgrund höherer Anforderungen

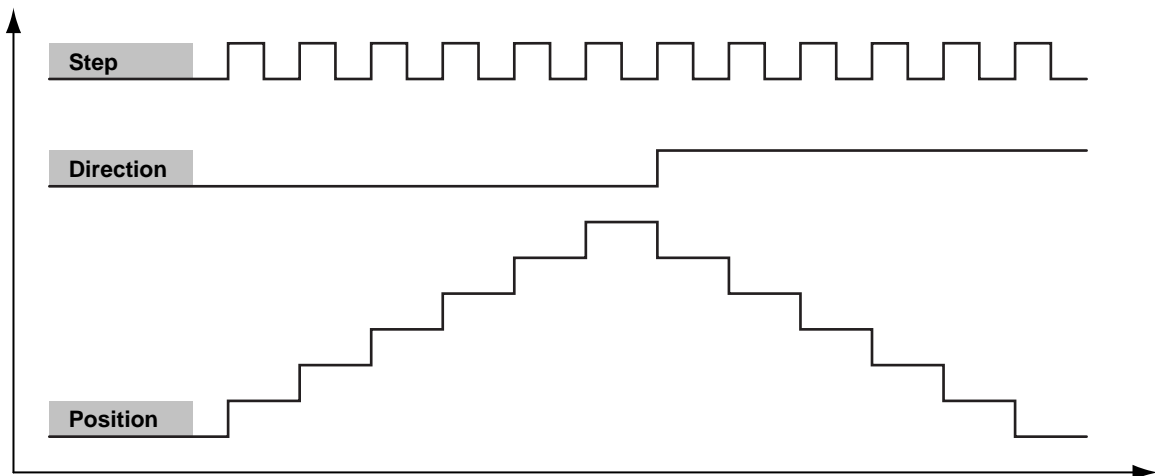
an Dynamik und Prozesssicherheit durch Linearmotoren ersetzt werden. Die Schrittweite beträgt  $0.1\mu\text{m}/\text{Schritt}$ .



### Lösung

Für diese Anwendung bietet sich der Run Mode "Position Indexing" an. In diesem Modus folgt der Motor dem Zählerwert des Indexer-Ein-

gangs. Das Indexer-Signal kann STEP/DIRECTION/ZERO (SDZ) oder ein Inkrementalsignal (ABZ) sein.



#### Hardware Schnittstelle:

Indexing-Signal an X12 anschliessen.

#### RunMode konfigurieren:

	Position Indexing	On	UPID 1450h	Run Modus Position Indexing
--	-------------------	----	------------	-----------------------------

#### Encoder Quelle definieren:

	Ext Sensor Input X12	On	UPID 172Ah	Anschluss des Indexing-Signals
--	----------------------	----	------------	--------------------------------

#### Encoder Typ definieren:

	Step Dir Zero (SDZ)	On	UPID 128Eh	Typ des Encoders festlegen
--	---------------------	----	------------	----------------------------

#### Decodierung einstellen

	1x	On	UPID 128F	Steigende Flanke zählen (Step)
--	----	----	-----------	--------------------------------

#### Positionsschritt pro Indexer-Inkrement einstellen:

	Resolution	$0.1\mu\text{m}$	UPID 14DDh	Schrittweite einstellen
--	------------	------------------	------------	-------------------------

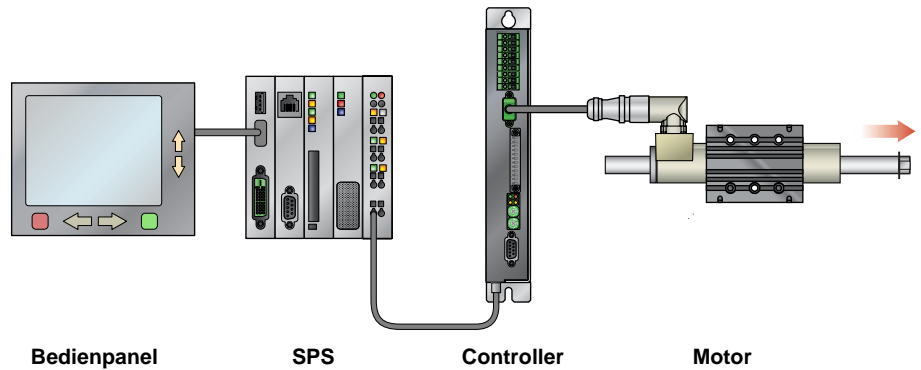
#### Dynamik begrenzen/konfigurieren:

	Max. Speed	1m/s	UPID 14BEh	Maximale Geschwindigkeit
	Acceleration	$5\text{m/s}^2$	UPID 14BFh	Beschleunigung
	Deceleration	$5\text{m/s}^2$	UPID 14C0h	Verzögerung

## 5. Anfahren von beliebigen Endpositionen über serielle Schnittstellen oder Feldbusse

### Aufgabe

Die Endpositionen des Linearantriebs sollen vom Bediener via Panel frei programmiert werden können. Er will in diesem Beispiel von der Ausgangsposition (0mm) auf 20mm ( $v=1\text{m/s}$ ,  $a=10\text{m/s}^2$ ) fahren, anschließend auf 80mm ( $0.2\text{m/s}$ ,  $1\text{m/s}^2$ ) und danach wieder zurück auf 0mm ( $2.5\text{m/s}$ ,  $30\text{m/s}^2$ ). Das Panel ist an einer SPS angeschlossen, welche über eine serielle Schnittstelle oder Feldbus mit dem LinMot Controller kommuniziert.



### Lösung

Die LinMot Controller lassen sich über verschiedene Schnittstellen wie Ethernet, Profibus DP, CANOpen, DeviceNet, RS232 und RS485 (LinRS Protokoll) mit einer übergeordneten Steuerung verbinden.

Dadurch lassen sich die gewünschten Endpositionen direkt über die Steuerung vorgeben.

Zur Kommunikation mit dem Controller werden die folgenden Ressourcen benötigt.

#### Control Word

Mit dem Control Word wird die Zustandsmaschine des Controllers gesteuert. Unter anderem wird über das Control Word der Controller gestartet und initialisiert, Fehler werden bestätigt, ein QuickStop ausgelöst usw.

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Switch On	Voltage Enable	/Quick Stop	Enable Operation	/Abort	/Freeze	Go To Position	Error Acknowledge	Jog Move +	Jog Move -	Reserved	Home	Clearance Check	Go To Initial Position	Reserved	Phase Search

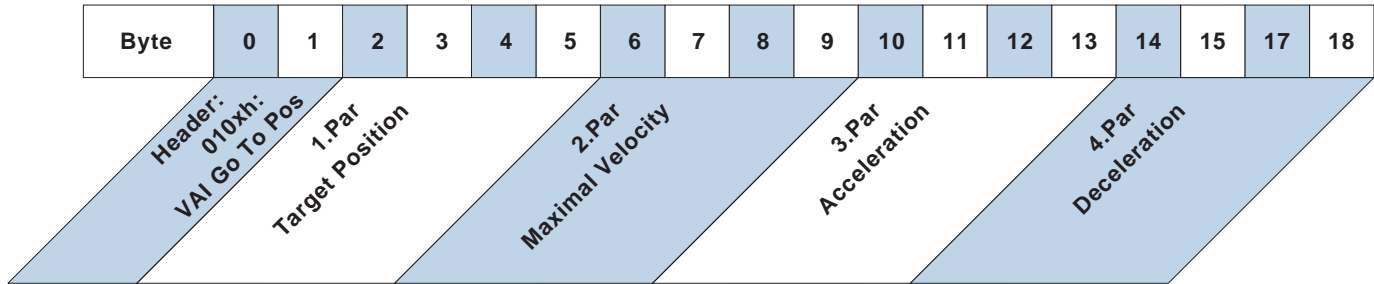
#### Status Word

Im Status Word werden Informationen zum Controller angezeigt. Ist der Antrieb initialisiert, liegt ein Fehler oder eine Warnung an, befindet sich der Antrieb an der Zielposition usw.

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Operation Enabled	Switch On Active	Enable Operation	Error	Voltage Enable	/Quick Stop	Switch On Locked	Warning	Event Handler Active	Special Motion Active	In Target Position	Homed	Fatal Error	Motion Active	Position Band 1	Position Band 2

## "VAI Go To Position (010xh)" Befehl

Der "VAI Go To Position" Befehl wird über das Motion Command Interface an den Controller gesendet. Als Parameter werden Zielposition, maximale Geschwindigkeit, Beschleunigung und Verzögerung mitgeliefert.



## StateVar

Die StateVar zeigt den Zustand, in dem sich der Controller befindet (Operation Enabled: 8, Homing: 9, Error: 4 usw).

Im Zustand 4 (Error) wird in den 8 niederwertigsten Bits der Errorcode angezeigt. Im Zustand 8 (Operation Enabled) zeigen die 4 niederwertigsten Bits den CommandCount. Die StateVar wird zur Synchronisation der Motion Commands über Feldbusse benötigt.

Main State								Sub State							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

## StateVar

Main State		Sub State
00	Not Ready To Switch On	0
01	Switch On Disabled	0
02	Ready To Switch On	0
03	Setup Error	Error Code which will be logged
04	Error	Logged Error Code
05	HW Tests	0 (Not yet defined)
06	Ready To Operate	0 (Not yet defined)
07		
08	Operation Enabled	Bits 0..3: Motion Command Count Bit 4: Event Handler Active Bit 5: Motion Active Bit 6: In Target Position Bit 7: Homed
09	Homing	0Fh: Homing Finished
10	Clearance Check	0Fh: Clearance Check Finished
11	Going To Initial Position	0Fh: Going To Initial Position Finished
12	Aborting	Not yet defined
13	Freezing	Not yet defined
14	Quick Stop (Error Behaviour)	Not yet defined
15	Going To Position	0Fh: Going To Position Finished
16	Jogging +	01h: Moving positive 0Fh: Jogging +Finished
17	Jogging -	01h: Moving negative 0Fh: Jogging -Finished
18	Linearizing	Not yet defined
19	Phase Search	Not yet defined

## Ausführung

Bevor ein Befehl gesendet wird, muss überprüft werden, dass sich der Controller im Zustand 8 (Operation Enabled) befindet (Highbyte der StateVar = 08h) und referenziert ist (Homed: Bit 11 im Status Word bzw. Bit 7 in der StateVar ge-

setzt). Ausserdem ist zu beachten, dass der Controller einen Befehl nur ausführt, wenn der Command Count des Command Headers (niederwertigste 4 Bits) nicht gleich dem Command Count in der StateVar (niederwertigste 4 Bits) ist.

Im einfachsten Fall setzt man von Befehl zu Befehl das Bit 0 des Command Headers abwechselungsweise 0 bzw. 1 (toggeln).

- Beispiel:
1. Motor Enabled? (Highbyte der StateVar = 08h)
  2. Motor referenziert? (Bit 11 des Status Word gesetzt)
  3. Positionsbefehl 1 senden (Command Header = 0101h, Command Count = 1)
  4. Motor in Position? (Bit 6 der StateVar gesetzt)
  5. Positionsbefehl 2 senden (Command Header = 0100h, Command Count = 0)
  6. Motor in Position? (Bit 6 der StateVar gesetzt)

### Zur Aufgabe:

Nachfolgend wird gezeigt, wie die Positionen in der Aufgabenstellung nacheinander abgefahren

werden können, und was vor und nach dem Senden eines Befehls überprüft werden muss.

1. Prüfen:	
	Ist der Controller in Zustand 8? (Highbyte der StateVar = 08h).
	Ist der Motor referenziert? (Bit 11 im Status Word).
	Liegt ein Fehler an? (Highbyte der StateVar = 04h bzw. Bit 3 im Status Word gesetzt).

2. Senden des Fahrbefehls auf 20mm mit $v = 1\text{m/s}$ und $a=10\text{m/s}^2$ . Als Command Count im Header 1h setzen:				
Word	Beschreibung	Beispiel (Werte hexadezimal)		Einheit
1	Command Header with ID	0101h	VAl Go To Position, 1 = Command Count	-
2-3	1. Command Parameter	00030D40h	Position, 50mm	0.1 $\mu\text{m}$
4-5	2. Command Parameter	000F4240h	Maximale Geschwindigkeit, 1m/s	1E-6 m/s
6-7	3. Command Parameter	000F4240h	Beschleunigung, 10m/s <sup>2</sup>	1E-5 m/s <sup>2</sup>
8-9	4. Command Parameter	000F4240h	Verzögerung, 10m/s <sup>2</sup>	1E-5 m/s <sup>2</sup>

3. Prüfen:	
	Ist der Befehl vom Controller ausgeführt worden? Bit 0-3 der StateVar (CommandCount) = 1h
	Ist der Motor an der Zielposition? Bit 6 (In Target Position) der StateVar gesetzt.
	Liegt ein Fehler an? (Highbyte der StateVar = 04h bzw. Bit 3 im Status Word gesetzt)

#### 4. Senden des Fahrbefehls auf 80mm mit $v = 0.2 \text{ m/s}$ und $a = 1 \text{ m/s}^2$ . Als Command Count im Header 2h setzen.

Word	Beschreibung	Beispiel (Werte hexadezimal)	Einheit
1	Command Header with ID	0102h VAI Go To Position, 2 = Command Count	-
2-3	1. Command Parameter	000C3500h Position, 80mm	0.1 $\mu\text{m}$
4-5	2. Command Parameter	00030D40h Maximale Geschwindigkeit, 0.2m/s	1E-6 m/s
6-7	3. Command Parameter	000186A0h Beschleunigung, $1 \text{ m/s}^2$	1E-5 $\text{m/s}^2$
8-9	4. Command Parameter	000186A0h Verzögerung, $1 \text{ m/s}^2$	1E-5 $\text{m/s}^2$

#### 5. Prüfen:

Ist der Befehl vom Controller ausgeführt worden? Bit 0-3 der StateVar = 2h

Ist der Motor an der Zielposition? Bit 6 (In Target Position) der StateVar gesetzt

Liegt ein Fehler an? (Highbyte der StateVar = 04h bzw. Bit 3 im Status Word gesetzt)

#### 6. Senden des Fahrbefehls auf 0mm mit $v = 2.5 \text{ m/s}$ und $a = 30 \text{ m/s}^2$ . Als Command Count im Header 3h setzen

Word	Beschreibung	Beispiel (Werte hexadezimal)	Einheit
1	Command Header with ID	0103h VAI Go To Position, 3 = Command Count	-
2-3	1. Command Parameter	00000000h Position, 0mm	0.1 $\mu\text{m}$
4-5	2. Command Parameter	002625A0h Maximale Geschwindigkeit, 2.5m/s	1E-6 m/s
6-7	3. Command Parameter	002DC6C0h Beschleunigung, $30 \text{ m/s}^2$	1E-5 $\text{m/s}^2$
8-9	4. Command Parameter	002DC6C0h Verzögerung, $30 \text{ m/s}^2$	1E-5 $\text{m/s}^2$

#### 7. Prüfen:

Ist der Befehl vom Controller ausgeführt worden? Bit 0-3 der StateVar = 3h

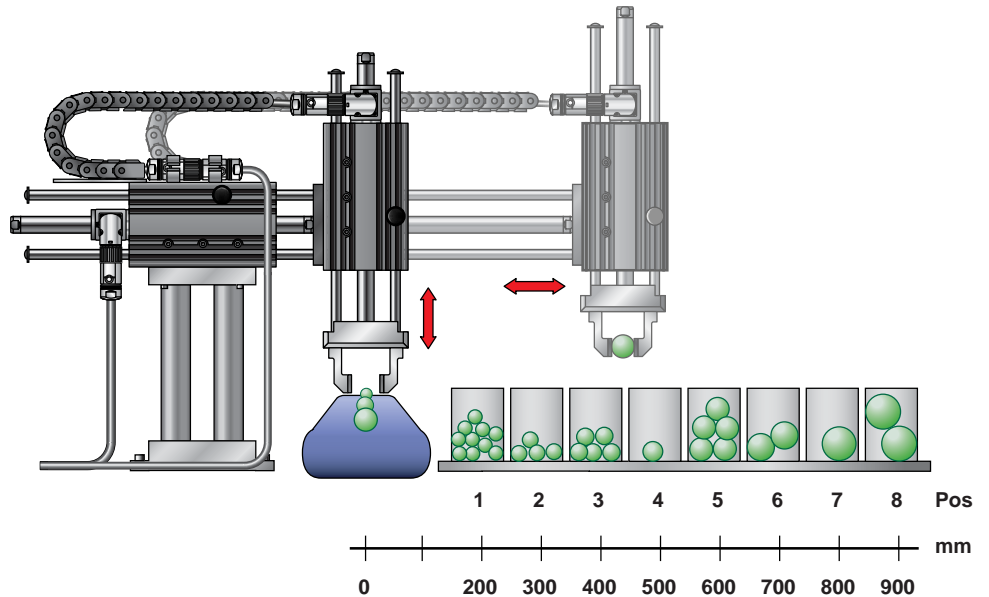
Ist der Motor an der Zielposition? Bit 6 (In Target Position) der StateVar gesetzt.

Liegt ein Fehler an? (Highbyte der StateVar = 04h bzw. Bit 3 im Status Word gesetzt)

## 6. Anfahren von 8 Positionen via digitaler I/O's

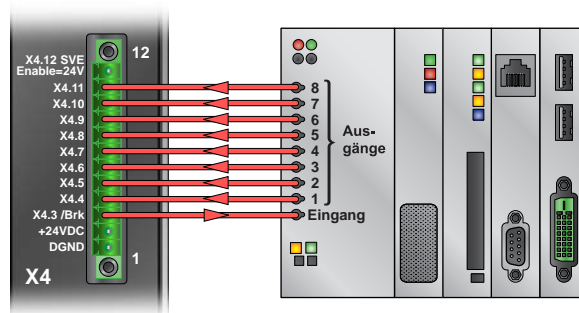
### Aufgabe

In einer In einer Anwendung soll ein Produkt nach Größe sortiert werden und entsprechend an maximal 7 Positionen abgelegt werden. Die eingesetzte SPS besitzt nur digitale I/O's. Es werden 8 Positionen benötigt. Eine Ausgangsposition, an der die Produkte aufgenommen werden und 7 Positionen für die Ablagen. Die 8 Positionen sollen über 8 einzelne digitale Eingangssignale aufgerufen werden. Sobald die betreffende Position erreicht ist, soll dies über einen InPosition-Ausgang angezeigt werden. Die Fahrt zu den Ablagen soll eher langsam (0.2 m/s, 2 m/s<sup>2</sup>), die Rückfahrt zur Ausgangsposition schnell (1 m/s, 10 m/s<sup>2</sup>) erfolgen.



### Lösung

Die Aufgabe wird mit Easy Steps gelöst. Easy Steps ist eine einfach zu bedienende Anwendung, bei der über eine steigende Flanke an den Eingängen von X4.4 bis X4.11 jeweils ein konfigurierbarer Fahrbefehl ausgelöst wird. Easy Steps ist eine Applikations-Software und muss via LinMot-Talk1100 installiert werden. ("File -> Install Firmware", ..\Applications\EasySteps)



#### Konfigurieren von X4.3 als InPosition-Ausgang:

Status Word:	In Target Position (Output)	On	UPID 1036h	X4.3 als InPosition-Ausgang
--------------	-----------------------------	----	------------	-----------------------------

#### Konfigurieren von X4.4 und Position 1:

Goto Abs Pos	On	UPID 3500h	Befehl bei steigender Flanke an X4.4 konfigurieren
Position	0mm	UPID 3510h	Position 1 festlegen
Max. Speed	1m/s	UPID 3511h	Maximale Geschwindigkeit
Acceleration	10m/s <sup>2</sup>	UPID 3512h	Beschleunigung
Deceleration	10m/s <sup>2</sup>	UPID 3513h	Verzögerung

#### Konfigurieren von X4.5 und Position 2:

Goto Abs Pos	On	UPID 3600h	Befehl bei steigender Flanke an X4.5 konfigurieren
Position	200mm	UPID 3610h	Position 2 festlegen
Max. Speed	0.2m/s	UPID 3611h	Maximale Geschwindigkeit
Acceleration	2m/s <sup>2</sup>	UPID 3612h	Beschleunigung
Deceleration	2m/s <sup>2</sup>	UPID 3613h	Verzögerung

#### Konfigurieren von X4.6 bis X4.11 und Position 3 bis Position 8

analog zu X4.4 und X4.5

## 7. Ablaufsteuerung

### Aufgabe

In einer Anlage werden Schaumstoffquader getestet. Ein Linearmotor soll den Quader mit einer Kraft von 40 N zusammendrücken. Nach 2 Sekunden Presszeit wird gemessen, ob der Quader innerhalb der Toleranzen liegt. Der ganze Ablauf soll mittels eines Triggersignals gestartet werden. Ist der Quader in Ordnung, soll dies an einem digitalen Ausgang signalisiert werden. Ist er fehlerhaft ebenfalls. Als Motor steht ein PL01-37x240 zur Verfügung.

Dies geschieht mit folgendem Ablauf:	
1.	Linearmotor auf Position 40 mm fahren, mit $v = 3 \text{ m/s}$ und $a = 5 \text{ m/s}^2$
2.	Kraft auf 40 N reduzieren und mit einer Geschwindigkeit von $0.05 \text{ m/s}$ den Quader zusammenpressen
3.	2 Sekunden pressen
4.	Überprüfen der Toleranzen: Wenn der Linearmotor sich an einer Position grösser als 65 mm und kleiner als 75mm befindet ist der Quader in Ordnung, ansonsten fehlerhaft.
5.	Rückfahrt zur Ausgangsposition 0mm, mit $v = 0.5 \text{ m/s}$ und $a = 5 \text{ m/s}^2$

### Lösung

Für diese Aufgabe bietet sich die Command Table an. Diese ermöglicht die Programmierung von einfachsten bis zu komplexen Abläufen mit Hilfe verschiedener Motion Commands, Bedingungen, Verzweigungen, Parameterzugriff ...

Die Command Table für den geforderten Ablauf wird mit der Software LinMot-Talk1100 erstellt

und auf den Controller geladen. Das Triggersignal wird an X4.6 angeschlossen. Ist der Quader innerhalb der Toleranz, wird dies an X4.8 (OK) signalisiert, ist er ausserhalb der Toleranz an X4.7 (Fehler).

Um die Kraft des Linearmotors auf 40 N zu beschränken, muss der maximale Strom begrenzt

werden. Der Motor vom Typ PL01-37x240 hat eine Kraftkonstante von  $23.8 \text{ N/A}$ , woraus für 40 N ein Strom von  $1.68 \text{ A}$  folgt ( $40/23.8$ ).

Die Schleppfehlerüberwachung des Controllers muss deaktiviert werden, da der Motor die Sollposition beim Pressen nicht erreichen wird. Was in dieser Aufgabe bewusst gewollt ist.

#### Hardware Schnittstellen:

Triggersignal an X4.6 (Input)

Error Signal an X4.7 (Output)

OK Signal an X4.8 (Output)

Konfiguration des Controllers:							
1. RunMode einstellen:							
Command Table Mode	On	UPID 1450h	Run Modus Command Table Mode				
2. Command Table Entry ID einstellen							
Command Table Entry ID	1	UPID 1485h	Festlegen der Startzeile				
3. Konfigurieren des Triggereingangs:							
Trigger (Input)	On	UPID 1039h	X4.6 als Trigger Eingang				
Direct	On	UPID 170Ch	Trigger Modus direkt				
4. Konfigurieren des Error Ausgangs:							
Interface Output	On	UPID 103Ah	X4.7 als Interface Ausgang				
5. Konfigurieren des OK Ausgangs:							
Interface Output	On	UPID 103Bh	X4.8 als Interface Ausgang				
6. Deaktivieren der Schleppfehlerüberwachung:							
Position Lag Always	False	UPID 1587h	allgemeiner Schleppfehler ausschalten				
Position Lag Standing	False	UPID 1588h	Schleppfehler im Stillstand ausschalten				
7. Erstellen der Command Table							
<input type="button" value="Upload from Controller"/> <input type="button" value="Download to Controller"/>							
ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Sequenced Entry
1	Warte Trigger	Wait until Rising Trigger Edge					2 (Fahre Pos 40mm)
2	Fahre Pos 40mm	VAI Go To Pos	Pos: 40 mm	Vel: 3 m/s	Acc: 5 m/s <sup>2</sup>	Dec: 5 m/s <sup>2</sup>	3 (Warte InPosition)
3	Warte InPosition	Wait until In Target Position					4 (Fahre Pos 90mm)
4	Fahre Pos 90mm	VAI Go To Pos	Pos: 100 mm	Vel: 0.05 m/s	Acc: 2 m/s <sup>2</sup>	Dec: 2 m/s <sup>2</sup>	5 (Kraft reduzieren)
5	Kraft reduzieren	Write Live Parameter	UPID: 13A6h (Maximal Current)	Value: 1.68 A			6 (Kraft erreicht?)
6	Kraft erreicht?	IF Current Greater Than	Val: 1.67 A	True Cmd ID: 10 (Presse 2s)	False Cmd ID: 7 (Dummy)		None
7	Dummy	No Operation					8 (Pos > 99 mm? )
8	Pos > 99 mm?	IF Actual Position Greater Than	Val: 99 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 6 (Kraft erreicht?)		None
9							
10	Presse 2s	Wait Time	Time: 2000 ms				11 (Pos kleiner 65?)
11	Pos kleiner 65?	IF Actual Position Less Than	Val: 65 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 12 (Pos grösser 75?)		None
12	Pos grösser 75?	IF Actual Position Greater Than	Val: 75 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 15 (Set OK X4.8)		None
13							
14	Set Error X4.7	Write Live Parameter	UPID: 1C89h [X4 Intf Outputs]	Value: 00000010h			17 (Fahre Pos 0mm)
15	Set OK X4.8	Write Live Parameter	UPID: 1C89h [X4 Intf Outputs]	Value: 00000020h			17 (Fahre Pos 0mm)
16							
17	Fahre Pos 0mm	VAI Go To Pos From Act Pos And Act Vel	Pos: 0 mm	Vel: 0.5 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>	18 (Normale Kraft)
18	Normale Kraft	Write Live Parameter	UPID: 13A6h (Maximal Current)	Value: 8 A			19 (Warte InPosition)
19	Warte InPosition	Wait until In Target Position					20 (Reset Ausgänge)
20	Reset Ausgänge	Write Live Parameter	UPID: 1C89h [X4 Intf Outputs]	Value: 00000000h			1 (Warte Trigger)
21							
8. Command Table auf dem Controller speichern -> Download to Controller							



## Hinweis

In Zeile 4 der Command Table wird ein VAI Go To Pos Befehl mit der Zielposition 100mm gesendet. Diese Position soll vom Linearmotor nie erreicht werden, er soll auf den Quader pressen.

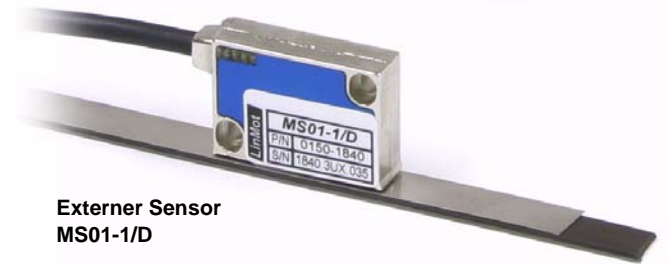
Wenn der Linearmotor auf kein Hindernis trifft, wird er, sobald er die Position 99mm erreicht, eine Fehlermeldung an X4.7 ausgeben.

Die Zeile 7 mit dem Befehl No Operation ist nötig, um die beiden IF-Anweisungen in Zeile 6 und 8 zu entkoppeln.

## 8. Hochpräzises Positionieren

### Aufgabe

Für eine hochpräzise Positionieraufgabe soll zur Verbesserung der Positioniergenauigkeit, die Position des Linearmotors durch einen hochauflösenden externen Sensor gemessen werden.



Externer Sensor MS01-1/D

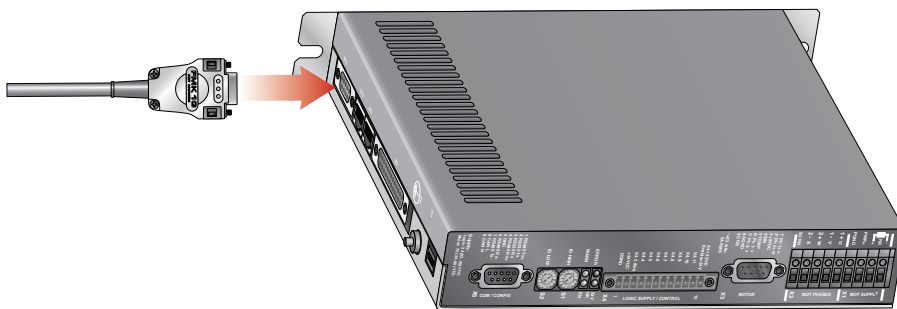
### Lösung

Die Controller der E1100 Serie unterstützen die Einbindung eines externen Positionssensors an X12. In diesem Beispiel wird ein AB Linear Enco-

der vom Typ MS01-1/D (LinMot Art.Nr. 0150-1840) mit einer Auflösung von 1µm verwendet.

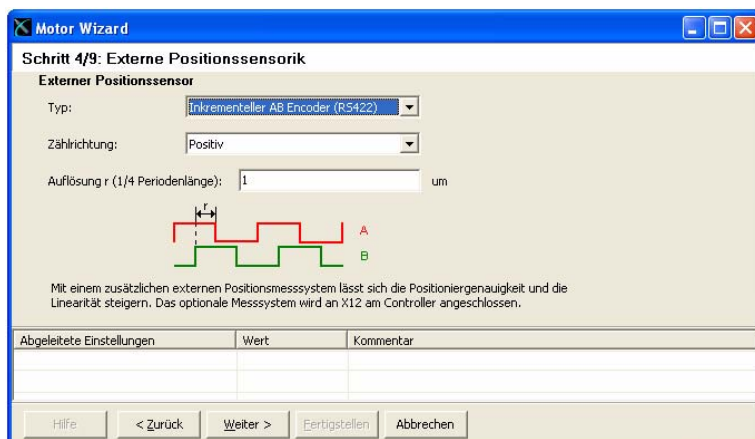
### Hardware Schnittstelle:

Anschliessen des externen Sensors an X12 am Controller.



### Konfiguration des Sensors:

Öffnen des Motor Wizard -> Schritt 4 "Externe Positionssensorik"



### Einstellungen

Auswahl des Typs:	Der MS01-1/D ist ein AB Sensor
Einstellen der Zählrichtung:	Positive oder Negativ
Konfigurieren der Auflösung:	1 µm
Weitere Einstellmöglichkeiten unter "Parameters -> Motor Configuration -> Position Feedback -> Feedback on X12/X10"	

### Konfigurieren eines externen Positionssensors mithilfe des MotorWizard



### Hinweis

Nach der Konfiguration des externen Sensors sollte die Zählrichtung folgendermassen überprüft werden. Zuerst die Firmware starten und

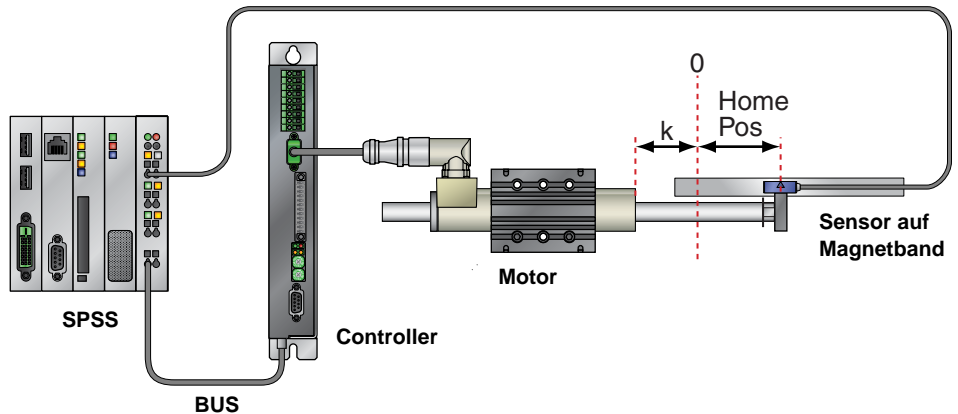
anschliessend den Läufer manuell mit der Hand verschieben. Dabei die Position im Control Panel von LinMot-Talk beobachten. Wird der Läufer

vorne aus dem Stator gezogen, muss die aktuelle Position in positiver Richtung zählen.

## 9. Betrieb mit externem absolut Sensor

### Aufgabe

In einer komplexen Anwendung ist es nicht möglich, eine Fahrt zur Referenzierung des Linearmotors durchzuführen. Aus diesem Grund wird ein absolut Sensor zu Positionserfassung des Linearantriebs verwendet.



### Lösung

Das Signal des absolut Sensors wird auf die SPS geführt. Damit der Linearmotor Randeffekte kompensieren kann, und eine optimale Positionsregelung erreicht wird, muss die Lage des

Läufers zum Stator zum Zeitpunkt der Initialisierung bekannt sein. Da die aktuelle Position des absolut Sensors von der SPS an den Controller

gesendet werden muss, ist eine Verbindung mittels serieller Schnittstelle oder Feldbus nötig.

#### Konfiguration des Controllers:

Actual Position	On	UPID 13C4h	Homing-Modus einstellen.
Disable	On	UPID 13D8h	Auto on Homing deaktivieren

#### Ablauf zur Initialisierung des Antriebs:

1.	Controller einschalten
2.	Motor Enable
3.	SPS liest IstPosition des absolut Sensors ein
4.	Home Position schreiben (IstPosition des absolut Sensors) UPID 13C7h
5.	Slider Home Position schreiben (Siehe unten) UPID 13CAh
6.	Homing auf Actual Position (Bit 11 im Control Word setzen)
7.	Sobald im Status Word Bit 11 (Homed) gesetzt ist, Bit 11 im Control Word wieder löschen
8.	Antrieb bereit

#### Bestimmung der Slider Home Position:

Um eine korrekte Initialisierung des Antriebs zu gewährleisten, muss zuerst bestimmt werden, wie weit der Läufer aus dem Stator hinausragt (=Distanz k), wenn der absolut Sensor sich an der Nullposition befindet. (Siehe Abbildung)

Die Slider Home Position, die in Schritt 5 geschrieben werden muss, ist die IstPosition des absolut Sensors plus die Distanz k.



### Hinweis

Werden Änderungen an der Mechanik vorgenommen, muss möglicherweise k neu bestimmt werden.

## 10. Ansteuerung einer Bremse

### Aufgabe

Ein Linearantrieb ist vertikal eingebaut. Um das Herunterfallen der Achse bei einem Fehler oder Stromausfall zu verhindern, soll eine mechanische Bremse eingesetzt werden, welche die Achse bei ausgeschaltetem Motor in Position hält.

### Lösung

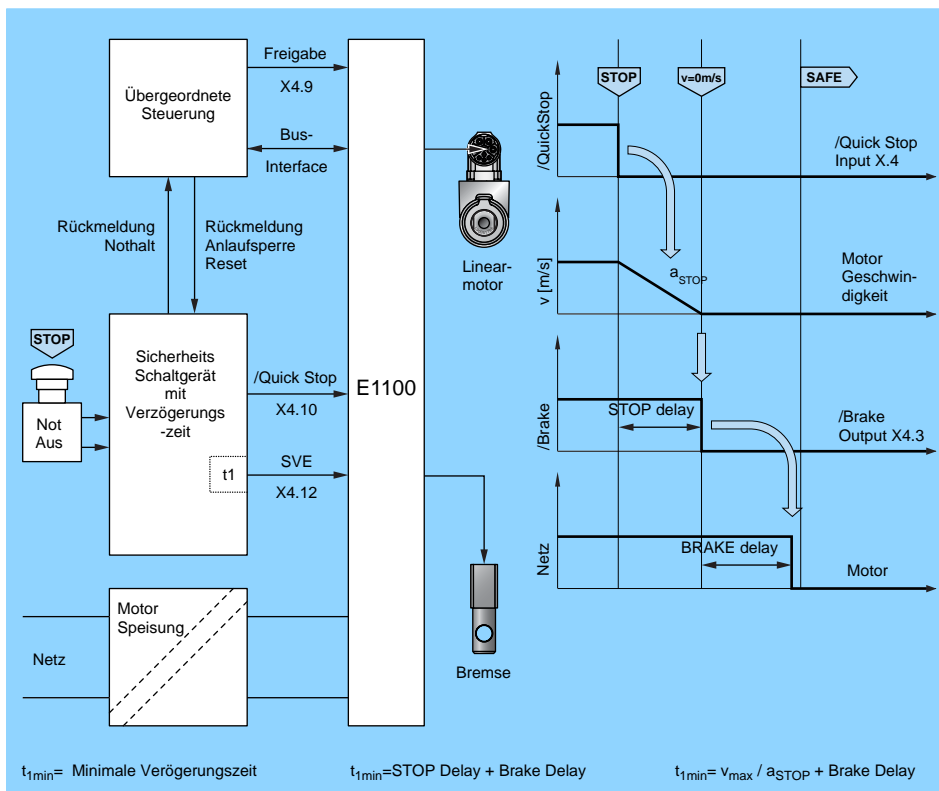
Die Controller der Serie E1100 unterstützen die Ansteuerung einer externen Bremse. X4.3 lässt sich als Bremsausgang mit einem maximalen Ausgangsstrom von 1A konfigurieren. Die Bremse arbeitet nach dem Ruhestromprinzip. D.h., bei anliegender Spannung wird sie gelüftet. Im Falle einer Störung geht der invertierte Bremsausgang auf OFF und die Bremse zieht an. Entsprechend ist der Bremsausgang des Controllers invertiert geschaltet (Betrieb: X4.3 = 1, Motor stromlos; X4.3 = 0). Fehler, die während einer Bewegung auftreten, lösen einen Quick-Stop aus, der den Motor sofort stoppt. Ist der QuickStop beendet, ist der Motor nicht mehr länger positionsgeregelt, d.h. der Motorstrom ist 0 (Null).

sprechend ist der Bremsausgang des Controllers invertiert geschaltet (Betrieb: X4.3 = 1, Motor stromlos; X4.3 = 0). Fehler, die während einer Bewegung auftreten, lösen einen Quick-Stop aus, der den Motor sofort stoppt. Ist der QuickStop beendet, ist der Motor nicht mehr länger positionsgeregelt, d.h. der Motorstrom ist 0 (Null).



Bremse

Konfigurieren des digitalen Ausgangs X4.3:				
	Brake (Output)	On	UPID 1036h	X4.3 als Bremsausgang
Einstellen des Bremsmodus:				
Status Word:	Operation Enabled	True	UPID 1717h	Bremsmodus einstellen
	Quick Stop	True	UPID 1716h	Bremsmodus einstellen
Verzögerungszeiten konfigurieren:				
	Apply Delay Time	100ms	UPID 171Bh	Nach Bedarf festlegen
	Release Delay Time	50ms	UPID 171Ch	Nach Bedarf festlegen



Apply Delay Time:
Ausschalten des Motors wird verzögert, bis die Bremse gegriffen hat.
Release Delay Time:
Das Lösen der Bremse wird verzögert, bis die aktive Positionsregelung arbeitet.



### Hinweis

Um den Quick Stop im Control Panel in LinMot-Talk auslösen zu können, muss die UPID 13EEh auf "False" gesetzt werden.

### Signalverlauf im Falle eines QuickStop

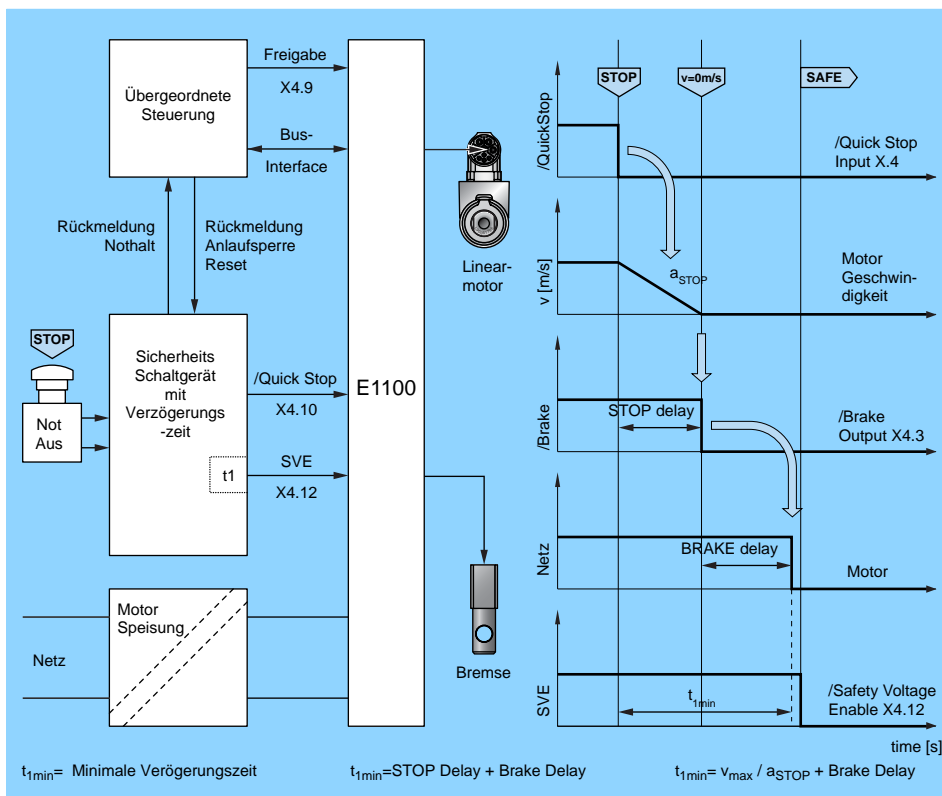
## 11. Sichere Impulssperre

Die sichere Impulssperre (Safety Voltage Enable) ist eine Sicherheitsfunktion. Die PWM-Generierung der Leistungselektronik wird hardwaremässig erst freigeschaltet, wenn der Eingang Safety Voltage Enable (X4.12) auf High (+24V) ist. Zur Realisierung des "Sicheren Halts" Kategorie 3 nach EN954-1 mit gesteuertem Stillsetzen

(Stopp 1 nach DIN EN60204-1) zeigt untenstehende Abbildung einen Schaltungsvorschlag.

Sobald die Impulsfreigabe über das Signal SVE gesperrt ist, wird die Leistungsendstufe ohne Verzögerung sicher ausgeschaltet. Die Impulsfreigabe ist extern als eine Klemme ausgeführt. Die Aufteilung in zwei unabhängige Signale geschieht intern im Servo Controller. Die externe

Beschaltung muss so ausgeführt werden, dass ein Kurzschluss mit andern spannungsführenden Teilen ausgeschlossen werden kann.

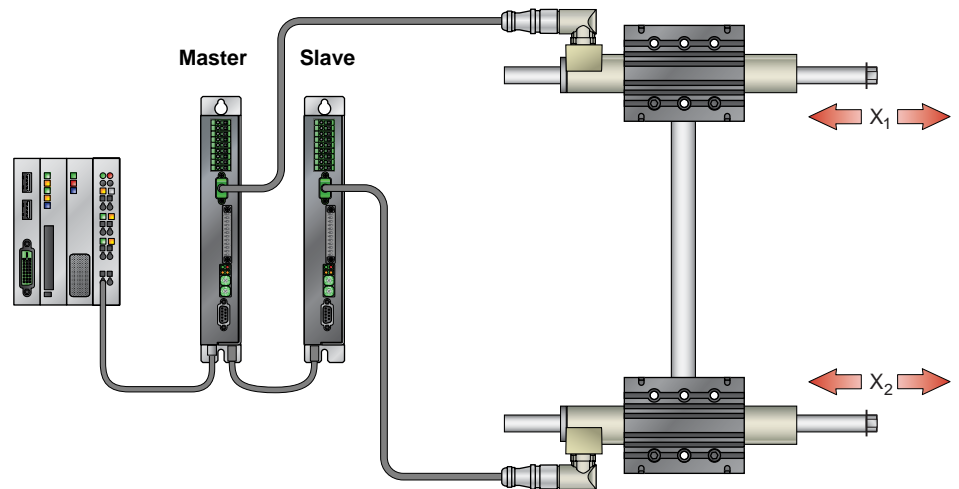


Schaltungsvorschlag "Sicherer Halt"

## 12. Parallelbetrieb von zwei Motoren

### Aufgabe

Es soll ein Portal mit zwei parallelen X-Achsen realisiert werden. Die beiden Motoren sollen sich parallel bewegen und von der übergeordneten Steuerung als eine Achse angesteuert werden können.



### Lösung

Für diese Anwendung bieten die LinMot Controller die Master-Slave Application-Software mit der Funktion "Master Gantry" an. Dabei werden zwei Controller der E1100 Serie über die Stecker X7/X8 bzw. X10/X11(E1100-GP(-HC)) miteinander verbunden. Der eine Controller wird als Master konfiguriert und durch die übergeordnete Steuerung angesprochen, der andere wird als

Slave konfiguriert und erhält vom Master die geforderte Sollposition. Beide Motoren sind unabhängig voneinander positionsgeregt. Die Initialisierung des Slave Controllers wird parallel zum Master ausgeführt.

Im Gantry-Betrieb liegen die Motoren eine gewisse Distanz auseinander und dürfen nicht 100% mechanisch steif gekoppelt sein.

#### Konfiguration der Controller:

Der Motor muss auf beiden Controllern konfiguriert werden (Motor Wizard)

Installation der Master-Slave Anwendung auf beiden Controllern: "File -> Install Firmware", \Firmware\Applications

Verbinden der beiden Controller via CAN-Bus (Stecker X7/X8) mit einem Ethernetkabel nach EIA / TIA 568A (Art. Nr. 0150-1853). CAN-Term S3.3 bei beiden Controllern auf ON setzen

#### Konfiguration des Masters:

CAN	On	UPID 3EF7h	Als Schnittstelle CAN wählen
Gantry Master	On	UPID 30D4h	Controller als Gantry Master konfigurieren

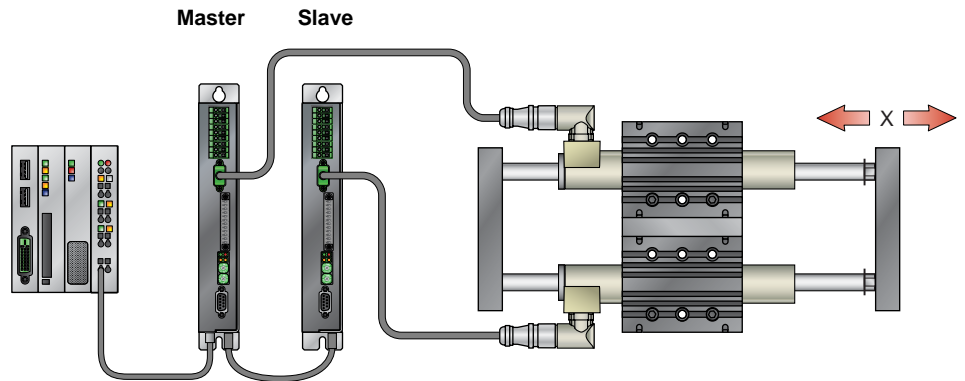
#### Konfiguration des Slaves:

CAN	On	UPID 3EF7h	Als Schnittstelle CAN wählen
Gantry Slave	On	UPID 30D4h	Controller als Gantry Slave konfigurieren
Normal*	On	UPID 30E2h	Richtung des Slave-Antriebs wählen
*(Normal = beide Statoren schauen in dieselbe Richtung)			

## 13. Kraftverdoppelung durch Master Booster

### Aufgabe

In einer Montageeinrichtung soll ein Werkzeug horizontal verschoben werden. Um die Dynamik und die Kraft zu erhöhen, sollen zwei Motoren parallel betrieben werden. Die beiden Antriebe sollen von der SPS als eine Achse angesteuert werden können.



### Lösung

Für diese Anwendung bieten die LinMot Controller die Master-Slave Application-Software mit der Funktion "Master Booster" an. Dabei werden zwei Controller der E1100 Serie über die Stecker X7/X8 bzw. X10/X11 (E1100-GP(-HC)) miteinander verbunden. Der eine Controller wird als Master konfiguriert und durch die übergeordnete

Steuerung angesprochen, der andere wird als Slave konfiguriert und erhält vom Master den berechneten Sollstrom (Slave nicht positionsgeregt). Die Initialisierung des Slave Controllers wird parallel zum Master ausgeführt. Im Booster Betrieb müssen die Motoren mechanisch steif gekoppelt sein.

#### Konfiguration der Controller:

Der Motor muss auf beiden Controllern konfiguriert werden (Motor Wizard)

Installation der Master-Slave Applikation auf beiden Controllern: "File -> Install Firmware", ..\Firmware\Applications

Verbinden der beiden Controller via CAN-Bus (Stecker X7/X8) mit einem Ethernetkabel nach EIA / TIA 568A (Art. Nr. 0150-1853). CAN-Term S3.3 bei beiden Controllern auf ON setzen.

#### Konfiguration des Masters:

	CAN	On	UPID 3EF7h	Als Schnittstelle CAN wählen
	Current Master	On	UPID 30D4h	Controller als Current Master konfigurieren

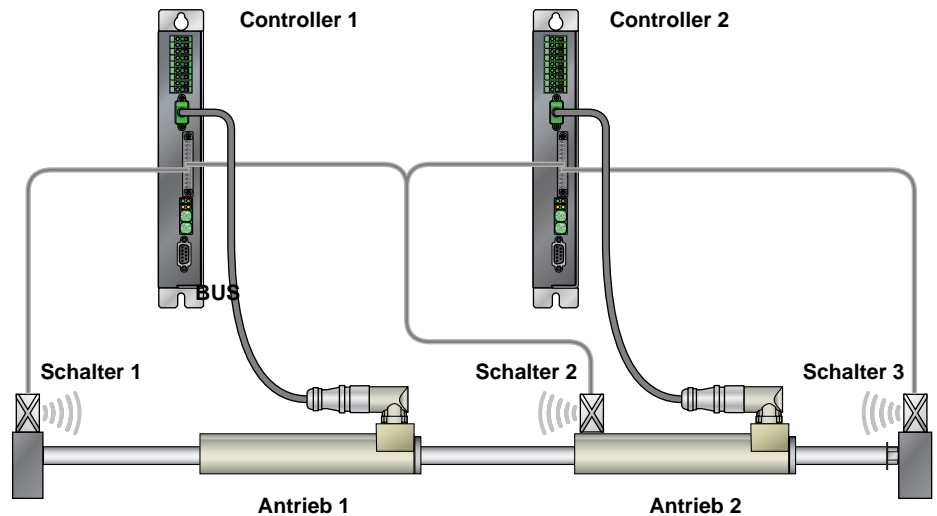
#### Konfiguration des Slaves:

	CAN	On	UPID 3EF7h	Als Schnittstelle CAN wählen
	Current Slave	On	UPID 30D4h	Controller als Current Slave konfigurieren
	Normal*	On	UPID 30E2h	Richtung des Slave-Antriebs wählen
*(Normal = beide Statoren schauen in dieselbe Richtung)				

## 14. Auswertung von Endlagen und Referenzschaltern

### Aufgabe

Zwei LinMot Antriebe bewegen sich unabhängig voneinander auf demselben Läufer. Um Kollisionen zu verhindern, wird an einem Stator ein Endlagenschalter (Schalter 2) befestigt. Wenn dieser Schalter auslöst, bedeutet das, dass die Distanz zwischen den beiden Statorn zu klein geworden ist und die Motoren müssen umgehend gestoppt werden. Zusätzlich werden an den beiden Endlagen zwei zusätzliche Endschalter montiert (Schalter 1 und 3).



### Lösung

An den Anschlüssen X4.8 und X4.9 der E1100 Serie Controller können Endschalter angeschlossen werden. Löst einer dieser Endschalter

aus, wird der Motor sofort durch einen Quick Stop angehalten.

#### Hardware Schnittstellen:

Endschalter 1 wird an X4.8 von Controller 1 angeschlossen
Endschalter 2 an X4.9 von Controller 1 und an X4.8 von Controller 2
Endschalter 3 an X4.9 von Controller 2

#### Konfiguration Controller 1:

Limit Switch Negative (Input)	On	UPID 103Bh	X4.8 als Endschalter negativ
Limit Switch Positive (Input)	On	UPID 103Ch	X4.9 als Endschalter positiv
Quick Stop	On	UPID 121Bh	Fehlverhalten einstellen
Deceleration	10 m/s <sup>2</sup>	UPID 1721h	Verzögerung bei Quick Stop

#### Konfiguration Controller 2:

Limit Switch Negative (Input)	On	UPID 103Bh	X4.8 als Endschalter negativ
Limit Switch Positive (Input)	On	UPID 103Ch	X4.9 als Endschalter positiv
Quick Stop	On	UPID 121Bh	Fehlverhalten einstellen
Deceleration	10 m/s <sup>2</sup>	UPID 1721h	Verzögerung bei Quick Stop

#### Gegebenenfalls die Logik der Endschalter anpassen (auf beiden Controllern):

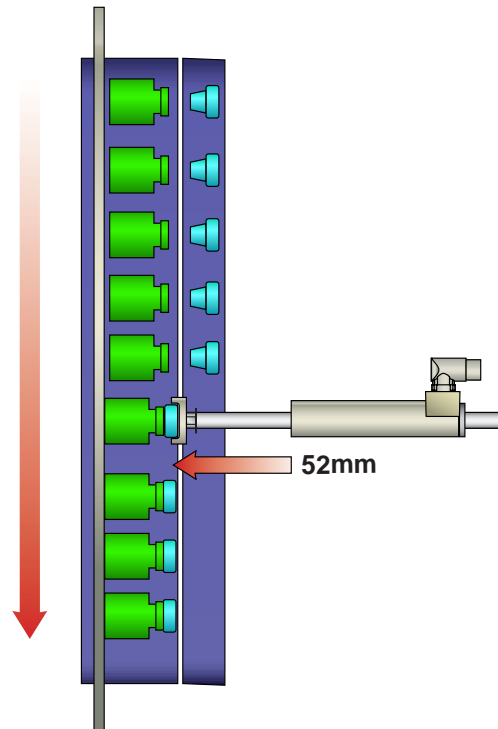
Invert I/O X4.8 (low active)	False	UPID 1050h	Logik von X4.8
Invert I/O X4.9 (low active)	False	UPID 1051h	Logik von X4.9

## 15. Pressen mit definierter Kraft

### Aufgabe

In einer Anwendung werden Verschlüsse eingeschoben. Um zu kontrollieren, dass der Verschluss korrekt sitzt, sollen die Endposition und die aufbrachte Kraft überprüft werden. Die geforderte Kraft beträgt 22 N und die Endposition 52mm mit einer Toleranz von +/- 1mm. Wird die Endposition erreicht, und die Kraft wurde nicht während 2s aufgebracht, oder liegt die Endposition ausserhalb der Toleranz, soll ein Fehler signalisiert werden. Andernfalls wird ein OK signalisiert. Die maximale Geschwindigkeit während des Einschubens darf 0.05 m/s nicht überschreiten.

Die Prüfung wird mittels eines Triggersignals gestartet. Als Motor steht ein PL01-23x160 zur Verfügung



### Lösung

Grundsätzlich ist die Kraftsteuerung sehr einfach und erfolgt über die Begrenzung des maximalen Motorstroms. Die resultierende Kraft wird über die Kraftkonstante [N/A] berechnet. Im Falle eines Motors vom Typ PL01-23x160 ist die Kraftkonstante 22.08 N/A. Es muss folglich ein

Maximalstrom von 0.996 A gesetzt werden, um mit 22 N zu pressen. Es ist zu beachten, dass der maximal zulässige Strom vom eingesetzten Controller und dem Motorentyp abhängig ist. Das Triggersignal und die beiden digitalen Aus-

gänge für das Fehler- und OK-Signal werden auf den Stecker X4 gelegt. Der geforderte Ablauf lässt sich mit kleinem Aufwand mit der Command Table realisieren.

#### Hardware Schnittstellen:

Triggersignal an X4.6 (Input)

Error Signal an X4.7 (Output)

OK Signal an X4.8 (Output)

## Konfiguration des Controllers:

### 1. RunMode einstellen:

Command Table Mode	On	UPID 1450h	Run Modus Command Table Mode
--------------------	----	------------	------------------------------

### 2. Command Table Entry ID einstellen

Command Table Entry ID	1	UPID 1485h	Festlegen der Startzeile
------------------------	---	------------	--------------------------

### 3. Konfigurieren des Triggereingangs:

Trigger (Input)	On	UPID 1039h	X4.6 als Trigger Eingang
Direct	On	UPID 170Ch	Trigger Modus direkt

### 4. Konfigurieren des Error Ausgangs:

Interface Output	On	UPID 103Ah	X4.7 als Interface Ausgang
------------------	----	------------	----------------------------

### 5. Konfigurieren des OK Ausgangs:

Interface Output	On	UPID 103Bh	X4.8 als Interface Ausgang
------------------	----	------------	----------------------------

### 6. Deaktivieren der Schleppfehlerüberwachung:

Position Lag Always	False	UPID 1587h	allgemeiner Schleppfehler ausschalten
Position Lag Standing	False	UPID 1588h	Schleppfehler im Stillstand ausschalten

### 7. Erstellen der Command Table

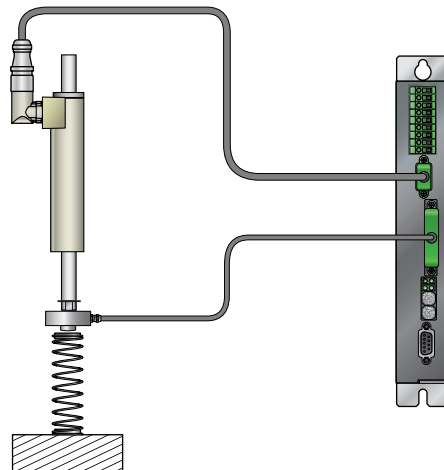
ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Sequenced Entry
1	Warte Trigger	Wait until Rising Trigger Edge					2 (Fahre Pos 40mm)
2	Fahre Pos 40mm	VAI Go To Pos	Pos: 40 mm	Vel: 3 m/s	Acc: 5 m/s <sup>2</sup>	Dec: 5 m/s <sup>2</sup>	3 (Warte InPosition)
3	Warte InPosition	Wait until In Target Position					4 (Fahre Pos 90mm)
4	Fahre Pos 90mm	VAI Go To Pos	Pos: 100 mm	Vel: 0.05 m/s	Acc: 2 m/s <sup>2</sup>	Dec: 2 m/s <sup>2</sup>	5 (Kraft reduzieren)
5	Kraft reduzieren	Write Live Parameter	UPID: 13A6h (Maximal Current)	Value: 1.68 A			6 (Kraft erreicht?)
6	Kraft erreicht?	IF Current Greater Than	Val: 1.67 A	True Cmd ID: 10 (Presse 2s)	False Cmd ID: 7 (Dummy)		None
7	Dummy	No Operation					8 (Pos > 99 mm? )
8	Pos > 99 mm?	IF Actual Position Greater Than	Val: 99 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 6 (Kraft erreicht?)		None
9							
10	Presse 2s	Wait Time	Time: 2000 ms				11 (Pos kleiner 65?)
11	Pos kleiner 65?	IF Actual Position Less Than	Val: 65 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 12 (Pos grösser 75?)		None
12	Pos grösser 75?	IF Actual Position Greater Than	Val: 75 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 15 (Set OK X4.8)		None
13							
14	Set Error X4.7	Write Live Parameter	UPID: 1C89h (X4 Intf Outputs)	Value: 00000010h			17 (Fahre Pos 0mm)
15	Set OK X4.8	Write Live Parameter	UPID: 1C89h (X4 Intf Outputs)	Value: 00000020h			17 (Fahre Pos 0mm)
16							
17	Fahre Pos 0mm	VAI Go To Pos From Act Pos And Act Vel	Pos: 0 mm	Vel: 0.5 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>	18 (Normale Kraft)
18	Normale Kraft	Write Live Parameter	UPID: 13A6h (Maximal Current)	Value: 8 A			19 (Warte InPosition)
19	Warte InPosition	Wait until In Target Position					20 (Reset Ausgänge)
20	Reset Ausgänge	Write Live Parameter	UPID: 1C89h (X4 Intf Outputs)	Value: 00000000h			1 (Warte Trigger)
21							

### 8. Command Table auf dem Controller speichern -> Download to Controller

## 16. Kraftregelung mit 0.1 N Auflösung

### Aufgabe

Zur Qualitätskontrolle sollen in einer Anlage Federn auf ihre Stärke geprüft werden. Dazu muss mit einer konstanten Kraft von 43.2 N vertikal auf die Feder gedrückt werden. Mithilfe der internen Positionsmessung wird ermittelt, wie stark die Feder zusammengepresst wird. Je nach gemessener Distanz erfüllen die Federn die Spezifikation. Die aufgebrachte Kraft wird von einer Kraftmessdose mit einem Messbereich von 0 bis 50 N gemessen.



### Lösung

LinMot bietet mit der Kraftregelung eine Technologiefunktion, welche eine präzise Regelung einer konstanten Kraft im ganzen Hubbereich, unabhängig von der aktuellen Position und mit einer Auflösung von bis zu 0.1 N ermöglicht. Da

die vom Linearmotor generierte Kraft von einer Kraftmessdose gemessen und direkt im Servo Controller geregelt wird, werden störende Effekte wie unterschiedliche Reibung, Verschmutzung, Slip-Stick-Effekte,

Temperaturschwankungen und andere Störgrößen ausgeregelt. Die von der Kraftmessdose ermittelte Kraft wird als ein analoges Signal (0 bis 10V) auf den Anschluss X4.4 geführt.

#### Hardware Schnittstelle:

Das analoge Signal der Kraftmessdose wird an X4.4 angeschlossen

#### Konfiguration von X4.4:

Analog Input	On	UPID 1037h	X4.4 als analogen Eingang konfigurieren
Force	On	UPID 1790h	Signaltyp Kraft

#### Konfiguration der Kraft bei 0V an X4.4

0V Force	0 N	UPID 1501h	Kraft bei 0V festlegen
----------	-----	------------	------------------------

#### Konfiguration der Kraft bei 10V an X4.4

10V Force	50	UPID 1502h	Kraft bei 10V festlegen
-----------	----	------------	-------------------------

#### Die Kraftregelung kann nun mit folgenden Befehlen genutzt werden:

VAI Go To Pos With Force Ctrl Limit (380xh):

Fahrt zu definierter Zielposition. Sobald die gemessene Kraft das Kraftlimit erreicht, wechselt der Controller in den Modus Kraftsteuerung mit Sollkraft = Kraftlimit. Um wieder mit Positionsregelung zu fahren, den Befehl "VAI Go To Pos From Act Pos And Reset Force Control (381xh)" benutzen!

VAI Go To Pos With Force Ctrl Limit And Target Force (383xh):

Fahrt zu definierter Zielposition. Sobald die gemessene Kraft das Kraftlimit erreicht, wechselt der Controller in den Modus Kraftsteuerung mit Sollkraft = Target Force. Um wieder mit Positionsregelung zu fahren, den Befehl "VAI Go To Pos From Act Pos And Reset Force Control (381xh)" benutzen!

Force Ctrl Change Target Force (382xh):

Mit diesem Befehl kann die Sollkraft im Kraftregelungsmodus geändert werden.

VAI Go To Pos From Act Pos And Reset Force Control (381xh)

Reaktiviert die Positionsregelung und fährt an die definierte Position.

Detaillierte Beschreibungen der Befehle sind im Motion Control SW Handbuch zu finden.



### Hinweis

Die Kraftregelung ist eine Technologiefunktion, die separat bestellt wird. (LinMot Art. Nr. 0150-2503)

## 17. Einbindung von rotativen Motoren

### Aufgabe

In einer Anlage mit Profibus soll ein bürstenloser Servomotor (EC Motor) mit Getriebe und Spindel mit einem LinMot Controller E1130-DP eingesetzt werden.

Der Antrieb ist ein Faulhaber Motor vom Typ 2036 U 036 B K1155. Das Getriebe hat ein Übersetzungsverhältnis von 3.71:1. Die Spindel besitzt eine Steigung von 1.5mm pro Umdrehung.



### Lösung

Die Controller der Serie E1100 unterstützen die Ansteuerung von 3-phasigen, rotativen EC Mo-

toren. Für einige Motorentypen stehen Actuator Definition Files (\*.adf) zur Verfügung. Sämtliche

bereits unterstützten Motoren sind im Handbuch "EC\_Motors\_with\_E1100" aufgelistet.

#### Anschliessen des Motors:

Motorphasen A, B, C an X2 (Alternativ X3)	(A->U, B->V, C->W)
Analoge Hall Sensoren A, B, C an X3	(A->X3.4, B->X3.9, C->X3.5)

#### Konfiguration des Controllers:

Motor Wizard öffnen
ADF Datei wählen: ..\Other Motors\EC Motors\Faulhaber\Faulhaber_EC.adf
Motorentyp wählen: 2036 U 036 B K1155
Bei Schritt 3 des Wizards wird das Übersetzungsverhältnis zwischen Motorwinkel und Position eingestellt. Wenn dem Controller eine Position in mm vorgegeben wird, soll sich der Linearschlitten der Spindel an diese Position bewegen. Deshalb muss berechnet werden, wie viele Millimeter Hub einer Umdrehung des Motors entsprechen.  Die Nominalübersetzung des Getriebes ist 3.71:1, die Steigung der Spindel ist 1.5 mm pro Umdrehung. Folglich ergibt eine Umdrehung des Motors $1.5/3.71 = 0.4043$ mm. Dieser Wert wird bei "1 Umdrehung = ... mm" eingetragen.
Einstellungen in Schritt 4 und 5 vornehmen und den Wizard abschliessen.



### Hinweis

Für die Ansteuerung von rotativen Motoren steht derselbe Funktionsumfang wie bei Linearmotoren zur Verfügung.

## 18. Synchronisation zur Königswelle: Master Encoder 0° bis 360°

### Aufgabe

In einer Anlage soll ein Linearmotor ein Bewegungsprofil synchron zur Königswelle abfahren. Der inkrementelle Master Encoder (ABZ) hat 512

Striche pro Umdrehung. Das gesamte Bewegungsprofil soll innerhalb einer Umdrehung der Königswelle (360 Maschinengrade) ablaufen.

### Lösung

Diese Aufgabe wird mit dem Run Mode "CAM Mode" gelöst. Dabei kann ein Bewegungsprofil erstellt und im Controller gespeichert werden.

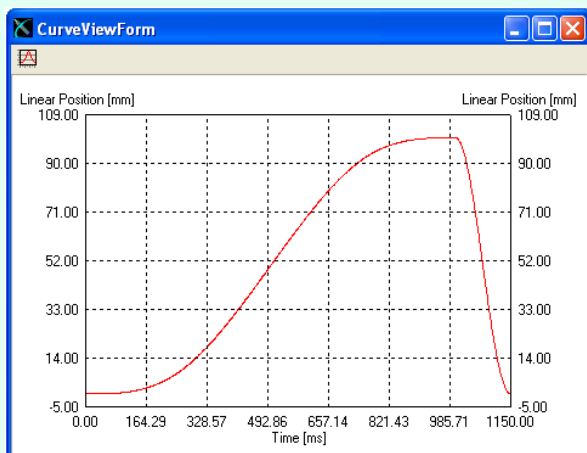
Das Bewegungsprofil wird nach dem Aufstarten des Controllers gestartet, wenn der Antrieb initialisiert wurde, der Controller sich im Zustand 8

"Operation Enabled" befindet und zum ersten Mal ein Z-Signal des Encoders detektiert wurde.

#### Erstellen des Bewegungsprofils:

Das Encodersignal der Königswelle wird 4-Fach ausgewertet. Daraus ergeben sich 2048 Inkremente pro Umdrehung des Master Encoders.

Im Curve Inspector in LinMot-Talk1100 wird das gewünschte Bewegungsprofil vom Typ "CAM (Pos. vs. Enc. Pos.)", mit einer Länge von 2048 Inkrementen erstellt. Eine Anleitung zur Erstellung von Kurven ist im Handbuch LinMot-Talk1100 zu finden.



Anschließend wird die erstellte Kurve mit ID 1 auf den Controller geladen.

#### Hardware Schnittstelle:

Der Master Encoder wird an X12 des Controllers angeschlossen.

#### RunMode konfigurieren:

CAM Mode	On	UPID 1450h	Run Modus CAM Mode
Start Counts	0	UPID 1523h	Startoffset einstellen
Curve ID	1	UPID 1524h	ID der Kurve konfigurieren

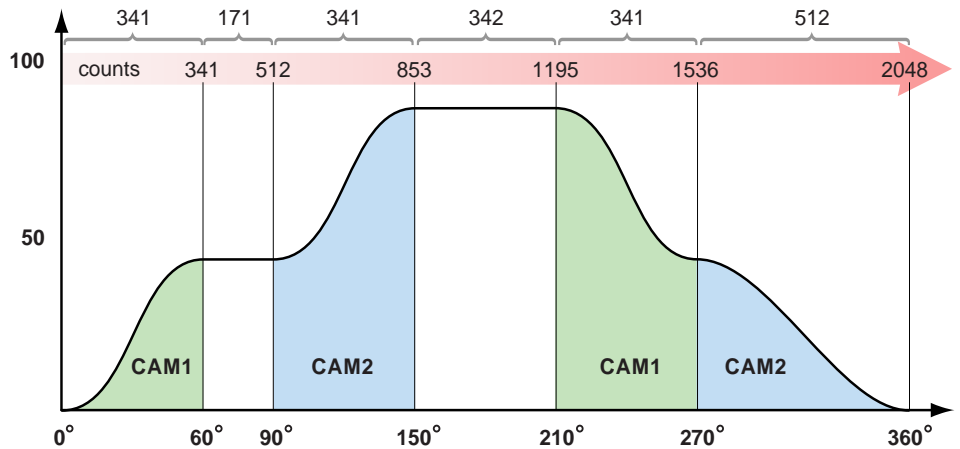
#### Master Encoder konfigurieren:

Ext Sensor Input X12	On	UPID 172Ah	Master Encoder an X12
ABZ	On	UPID 128Eh	Encoder vom Typ ABZ
4x	On	UPID 128Fh	Encodersignal 4fach decodieren
CAM length	2048	UPID 1525h	Länge der Kurvenscheibe

## 19. Synchronisation zur Königswelle: CAM1/CAM2

### Aufgabe

In einer Anwendung sollen abhängig vom Maschinenwinkel (Position Master Encoder / Königswelle) die Positionen 0mm, 50mm und 100mm angefahren werden. (Siehe Abbildung rechts).



### Lösung

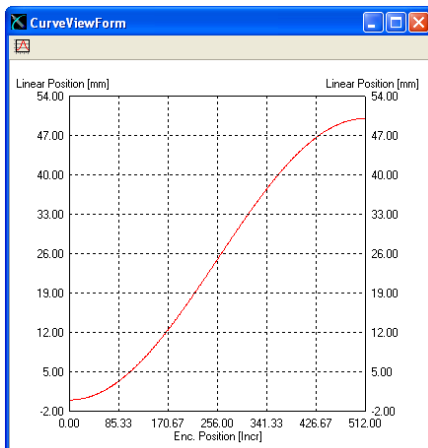
Die Controller der Serie E1100 unterstützen das Definieren von zwei CAM's. Dies ermöglicht das Laden eines Bewegungsprofils in ein CAM, wäh-

rend das Bewegungsprofil des anderen CAM abgefahren wird, und umgekehrt. Der Master Encoder hat 512 Striche pro Umdrehung. Der

Controller wird so konfiguriert, dass das Encodersignal 4x decodiert wird, woraus sich 2048 Inkremente pro Umdrehung des Master Encoders

### Erstellen des Bewegungsprofils:

Es wird ein zunächst ein einfaches sinusförmiges Bewegungsprofil benötigt. Dieses wird mit LinMot-Talk1100 erstellt und auf dem Controller gespeichert.



Typ: Cam (Pos. vs. Enc. Pos.), 0 bis 100mm, Länge: 512 Counts, ID 1

### Hardware Schnittstelle:

Die ABZ Signale des Master Encoders werden an X12 des Controllers angeschlossen.

### Master Encoder konfigurieren:

Ext Sensor Input X12	On	UPID 172Ah	Master Encoder an X12
ABZ	On	UPID 128Eh	Encoder vom Typ ABZ
4x	On	UPID 128Fh	Encodersignal 4fach decodieren
CAM length	2048	UPID 1525h	Länge der Kurvenscheibe

Ablauf:	
1.	Laden der Kurve in CAM1 mit dem Befehl:
	Encoder CAM 1 Define Curve To Pos In Counts (114xh)
	Curve ID: 1
	Curve Start Count 0 Incr
	Target Position 50 mm
	Curve Length 341
2.	Aktivieren des CAM-Modus mit dem Befehl:
	Encoder CAM Enable (100xh)
	Sobald nun das erste Z-Signal des Encoders detektiert wird, folgt der Motor dem Bewegungsprofil in CAM1.
3.	Während mit CAM1 die Kurve abgefahren wird, muss CAM2 geladen werden:
	Encoder CAM 2 Define Curve To Pos In Counts (124xh)
	Curve ID: 1
	Curve Start Count 512 Incr
	Target Position 100 mm
	Curve Length 341
4.	Während mit CAM2 die Kurve abgefahren wird, muss wieder CAM1 geladen werden:
	Encoder CAM 1 Define Curve To Pos In Counts (114xh)
	Curve ID: 1
	Curve Start Count 1195 Incr
	Target Position 50 mm
	Curve Length 341
5.	Während mit CAM1 die Kurve abgefahren wird, muss wieder CAM2 geladen werden:
	Encoder CAM 2 Define Curve To Pos In Counts (124xh)
	Curve ID: 1
	Curve Start Count 1536 Incr
	Target Position 0 mm
	Curve Length 512
6.	Während mit CAM2 die Kurve abgefahren wird, muss CAM1 wieder für die nächste Umdrehung des Encoders geladen werden:
	Encoder CAM 1 Define Curve To Pos In Counts (114xh)
	Curve ID: 1
	Curve Start Count 0 Incr
	Target Position 50 mm
	Curve Length 341
7.	Wiederholen der Schritte 3 bis 6



## Hinweis

Motion Commands können einerseits über serielle Schnittstellen, Feldbusse oder die Command Table aufgerufen werden. Zu Übungszwecken können sie auch über das Control Panel in LinMot-Talk 1100 abgesetzt werden (Motion Command Interface). Weitere Informati-

onen dazu sind in den Handbüchern "LinMot-Talk 1100" und "Motion Control SW" zu finden.

Die Synchronisierung und das Aufstarten bei stehendem Master Encoder bzw. die (Re)Synchronisierung zu einem laufenden Master Encoder sind im Benutzerhandbuch "Motion Control SW" in Kapitel 5.3 beschrieben.