


## How To

### Verwendung der Command Table

Entry ID:   Auto execute new command on next cycle  
 Entry Name:  ID of Sequenced Entry:  (=> Move to A)  
 Motion Command Category:   
 Motion Command Type:    
 Set Value Of Var 1:

Upload from Drive     Download to Drive

ID	Name	Type	Par 1	Par 2	Par 3
1	Init Var 1	Set Cmd Table Var 1 To	Set Val: 0		
2	Move to A	VAl Go To Pos	Pos: 5 mm	Vel: 1 m/s	Acc: 10 m/s^2
3	Pos A reached?	Wait until Motion Finished			
4	Move to B	VAl Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s^2
5	B reached?	Wait until Motion Finished			
6	Increment +1	Add To Cmd Table Var 1	Add Val: 1		
7	Loops done?	IF Cmd Table Var 1 Less T...	Val: 10	True Cmd ID: 2 (Move to ...	False Cmd ID: 8 (Finish...

**Klassifizierung:**     Öffentlich  
                                LinMot intern

<b>Information zur Command Table .....</b>	<b>3</b>
<i>Allgemeine Information zur Command Table.....</i>	<i>3</i>
<b>1 Startquellen für die Command Table Ausführung.....</b>	<b>4</b>
1.1 <i>Fehlersuche in der Command Table.....</i>	<i>4</i>
1.2 <i>Betriebsart: Triggered Command Table.....</i>	<i>5</i>
1.3 <i>Betriebsart: Command Table .....</i>	<i>6</i>
1.4 <i>Start der Command Table durch SPS .....</i>	<i>7</i>
1.5 <i>Start der Command Table mit EasySteps.....</i>	<i>8</i>
1.6 <i>Verwendung des Command Table I/O Interface (nur auf E1100-GP Drives).....</i>	<i>9</i>
<b>2 Bearbeiten der Command Table .....</b>	<b>10</b>
2.1 <i>Einen Befehl eingeben .....</i>	<i>11</i>
2.2 <i>Löschen von Kommandos.....</i>	<i>12</i>
2.3 <i>Zeilen verschieben / neue Zeile einfügen .....</i>	<i>12</i>
2.4 <i>Speichern der Command Table .....</i>	<i>12</i>
2.5 <i>Laden der Command Table .....</i>	<i>12</i>
<b>3 Command Table Programmierung .....</b>	<b>13</b>
3.1 <i>Bedingungen .....</i>	<i>14</i>
3.1.1 <i>Wartende Bedingungen .....</i>	<i>14</i>
3.1.2 <i>If Bedingungen .....</i>	<i>15</i>
3.2 <i>Variable / Parameter-Zugriff.....</i>	<i>16</i>
3.3 <i>Zugriff auf I/O .....</i>	<i>17</i>
3.3.1 <i>Erstellen der Bit-Maske .....</i>	<i>18</i>
3.3.2 <i>Abfragen eines digitalen Eingangs .....</i>	<i>19</i>
3.3.3 <i>Setzen eines Ausgangs .....</i>	<i>20</i>
<b>4 Beispiele.....</b>	<b>22</b>
4.1 <i>Beispiel 1: Gesteuertes Ausschalten .....</i>	<i>23</i>
4.2 <i>Einfacher Schleifenzähler .....</i>	<i>24</i>
4.3 <i>Stromausfallsicheres Speichern einer Command Table Variable .....</i>	<i>25</i>
4.4 <i>Applikation Schaumgummi Test.....</i>	<i>26</i>
<b>5 Dokumentversion .....</b>	<b>28</b>

## Information zur Command Table

LinMot Drives bieten eine Zusatzfunktionalität, welche als «**Command Table**» bezeichnet wird. Diese Funktionalität erlaubt es dem Benutzer einfache Ablaufsequenzen zu definieren - ähnlich wie auf einer SPS. Die **Command Table** ist kein Ersatz für eine SPS. Der Funktionsumfang ist limitiert. Typische Einsatzfälle können einfache Bewegungsabläufe sein, die ohne SPS ablaufen sollen. Beispielsweise eine Testanwendung, wo einfache Bewegungsabläufe über lange Zeit wiederholt werden sollen. Die Command Table stellt eine Simulation für das Interface dar. Ein Interface auf einem LinMot Drive ist typischerweise ein Feldbus, der die Abläufe steuert. Anstelle des Feldbusses als Datenquelle dient hier die Command Table.



**Achtung:** Die Command Table kann mittels Fahrbefehl über den Feldbus gestartet werden! Wenn die Command Table arbeitet, muss sichergestellt sein, dass keine weiteren Kommandos über den Feldbus erfolgen, da diese den Ablauf der Command Table überlagern und zu unvorhersehbaren Verhalten führt!  
Eine arbeitende Command Table wird im "Status Word Bit 8: Event Handler Active" signalisiert. Dieses Bit kann von der SPS überwacht werden, um den Betriebszustand der Command Table zu überwachen.



**Achtung:** LinMot Firmware bietet verschiedene Steuerquellen, die über die Firmware konfiguriert werden können. Daher muss eine doppelte Belegung vermieden werden, um Fehlverhalten zu vermeiden. Verwenden Sie beispielsweise niemals einen digitalen Eingang den Sie in der I/O Konfiguration zugewiesen haben, gleichzeitig in EasySteps!  
Verwenden Sie jede Signalquelle nur einmal in Ihrer Anwendung!

## Allgemeine Information zur Command Table

Übersicht Ausführungszeit, Befehlskapazität und I/O:

Drive	Ausführungszeit [us]	Kapazität [Zeilen]	I/O Linien
A1100	500	255	6 In / 2 Out
B1100	400	31	6 In / 6 Out
C1100	500	255	4 In / 2 Out
C1200	250	255	4 In / 2 Out
E1100	625	255	8 I/O (GP + 16 In / 8 Out)
E1200	200	255	8 I/O
C1400	250	255	4 In / 2 Out
E1400	250	255	8 I/O

Ausführungszeit bedeutet die Zykluszeit, die benötigt wird, um eine Zeile der Command Table aus zu führen. Die Firmware-Erweiterung „EasySteps“ wird in der gleichen Zykluszeit wie die Command Table ausgeführt. Detaillierte Informationen über I/O und deren Verdrahtung entnehmen Sie dem jeweiligen Drive Installation Guide.

Die I/O Abtastzeit ist unabhängig von der Command Table Zykluszeit und kann dem jeweiligen Drive Installation Guide entnommen werden.

## 1 Startquellen für die Command Table Ausführung

Das Starten einer Command Table kann von verschiedenen Signalquellen ausgelöst werden.

Für den Start der Tabelle wird die eindeutige Command Table ID (Zeilennummer) verwendet, um in der gewünschten Zeile zu beginnen.

In Abhängigkeit der gewählten Betriebsart wird diese Zeilennummer für die Ausführung angegeben.

Je nach verwendeter Betriebsart muss die Command Table als selbstaufrufende Schleife, oder mit Endpunkt angelegt werden.

Die Command Table endet, wenn in der auszuführenden Befehlszeile keine neue «Link ID» angegeben wird.

Eine laufende Command Table kann nicht direkt gestoppt werden!

Wird der Antrieb über das Control Word Bit 0 oder eine andere Quelle (Quickstop, STO) ausgeschaltet, während die Command Table noch läuft, kann dies zum Fehler "User Err: Motion Cmd In Wrong St" führen, falls die Command Table einen Fahrbefehl abarbeiten will. Dies sollte beim Erstellen der Command Table beachtet werden.

Eine laufende Command Table kann mit dem Fahrbefehl "Clear Event Evaluation" abgebrochen werden.

Die nachfolgenden Kapitel behandeln die verschiedenen Steuermöglichkeiten der Command Table und deren Programmierung.

### 1.1 Fehlersuche in der Command Table

Für den Zweck der Fehlersuche bei der Programmerstellung kann man den Bereich «Variables» in LinMot Talk benutzen:

The screenshot shows the LinMot Talk software interface. On the left, a project tree is visible with the 'Variables' folder expanded. A red box highlights the 'Variables' folder, and another red box highlights the 'MC SW Command Table' entry. A red arrow points from the 'MC SW Command Table' entry to the 'Variables' table on the right. The table has the following data:

Name	Value	RawData	UPID	Type
Last Evaluated Command Table ID	2	0002h	1E70h	UInt16
Actual Linked Command Table ID	3	0003h	1E71h	UInt16
Command Table Variable 1	0	00000000h	1E72h	SInt32
Command Table Variable 2	0	00000000h	1E73h	SInt32

Wenn die Command Table arbeitet, kann man die Variablen 1 & 2 sowie die Anzeige der aktuellen Zeile und der nächsten Zeile beobachten.

Dies kann bei einer Fehlersuche hilfreich sein, speziell wenn Bedingungen mittels «IF» Sprünge in der Ausführung erzeugen.

Diese Variablen können zur Beobachtung auch per Oszilloskop aufgezeichnet werden, da bestimmte Signalwechsel sehr schnell erfolgen können.

## 1.2 Betriebsart: Triggered Command Table

Die Command Table Ausführung kann per Trigger-Signal über einen digitalen Eingang gestartet werden.

Diese Betriebsart kann im Parameterbaum wie folgt konfiguriert werden:

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Betriebsart setzen	\Parameters\Motion Control SW\Motion Interface\Run Mode Settings\Run Mode Selection			
Triggered Command Table	On	62A8h	1450h	Betriebsart „Triggered Command Table“

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Setze ID steigende Flanke	\Parameters\Motion Control SW\Motion Interface\Run Mode Settings\Triggered Command Table Settings			
Rise Command Table Entry ID	ID Nummer	6302h	1486h	ID Nummer für Ausführung

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Setze ID fallende Flanke	\Parameters\Motion Control SW\Motion Interface\Run Mode Settings\Triggered Command Table Settings			
Fall Command Table Entry ID	ID Nummer	6303h	1487h	ID Nummer für Ausführung

In den Trigger-Einstellungen muss mindestens einer der beiden Trigger-Parameter auf eine gültige ID gesetzt werden. Je nach Anwendung können beide, oder nur steigende oder fallende Flanke Verwendung finden.



**Hinweis:** B1100 unterscheidet sich bei den UPID Adressen gegenüber den anderen Drive-Typen! Der Trigger Eingang ist immer auf einen spezifischen Hardware-Eingang definiert, der schneller abgetastet wird. Der Trigger Eingang kann daher nicht auf einen beliebigen Eingang konfiguriert werden!

Um diese Betriebsart zu nutzen, muss der Trigger wie folgt konfiguriert werden:

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Setze Trigger Input	\Parameters\Motion Control SW\Drive Configuration\X4/X14 IO Definitions\			
IO X4.6 (not B1100)	01 - Trigger	-	1039h	Setze Hardware Trigger
IO X14.15 (B1100)	01 - Trigger	62EAh	-	Setze Hardware Trigger

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Setze Trigger Modus	\Parameters\Motion Control SW\Drive Configuration\X4/X14 IO Definitions\Trigger\			
Trigger Mode	01 - Direct	62D8h	170Ch	Trigger-Auswertungsmodus

Abhängig vom gewählten Trigger-Modus kann eine Sperr- und Verzögerungszeit in den “Trigger Settings” definiert werden.



**Achtung:** Eine “Triggered Command Table” benötigt eine kurze, nicht blockierende Command Table!



**Achtung:** Die Command Table muss enden! Verwenden Sie keine kontinuierlichen Schleifen!



**Achtung:** Stellen Sie sicher, dass die Command Table fertig ist, bevor ein neuer Trigger ausgelöst wird, um undefiniertes Verhalten zu vermeiden!

## 1.3 Betriebsart: Command Table

Die Command Table kann direkt gestartet werden. Der Start erfolgt direkt nach dem Einschalten der Achse (Main State = 8), und nachdem die Achse referenziert ist (Status Word Homed = 1).  
Um diese Betriebsart zu nutzen, müssen folgende Einstellungen durchgeführt werden:

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Setze Betriebsart	\Parameters\Motion Control SW\Motion Interface\Run Mode Settings\Run Mode Selection			
Command Table Mode	On	62A8h	1450h	Betriebsart "Command Table Mode"

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Setze Start ID	\Parameters\Motion Control SW\Motion Interface\Run Mode Settings\Command Table Settings			
Command Table Entry ID	ID Nummer	6301h	1485h	ID Nummer für Start

Wenn in dieser Betriebsart die Command Table in einer Zeile ohne weitere Link-ID endet, so stoppt der Command Table Modus und startet erst wieder, wenn der Drive über das Control Word Switch On (Bit 0) aus- und wieder eingeschaltet wird.

Wenn in dieser Betriebsart der Ablauf der Command Table intern über die Link-ID wieder auf den Startpunkt geführt wird, endet die Ausführung nicht. Wird der Drive mit dem Control Word Bit 0 ausgeschaltet, stoppt die Command Table an der Zeile, wo die Abschaltung erfolgt und arbeitet nach dem Einschalten an dieser Stelle weiter.

In Abhängigkeit von der Anwendung kann dies ggf. zu ungewünschten Fehlern führen, da Fahrbefehle nur ausgeführt werden können, wenn der Drive betriebsbereit ist (Achse bestromt).

#### 1.4 Start der Command Table durch SPS

Die Command Table kann durch eine SPS per Feldbus gestartet werden. LinMot definiert als Kommandoschnittstelle ein «Interface». Dieses Interface kann Daten vom Feldbus oder auch von der Command Table (simulierter Feldbus) erhalten. Beide Kommandoquellen werdend verarbeitet, wobei der Feldbus eine höhere Priorität hat.

Daher: Beachten Sie, dass Sie immer nur eine Steuerquelle verwenden!

Wenn die Command Table gestartet wurde, darf die SPS keine Kommandos über den Feldbus senden, bis die Command Table beendet ist!

Um die Command Table zu starten, muss die SPS den Fahrbefehl "Start Command Table Command (200xh)" mit dem Parameterwort 1 senden, welches die Startzeile der Command Table enthalten muss.

Falls Sie LinMot Bibliotheken verwenden, finden Sie einen Funktionsbaustein, der diesen Aufruf erledigt.

Um eine laufende Command Table zu überwachen, muss die SPS das Bit 8 "Event Handler Active" im Status Word überwachen. Solange dieses Bit gesetzt ist, läuft die Command Table.

Wenn Sie eine laufende Command Table abbrechen wollen, senden Sie den Fahrbefehl "Clear Event Evaluation (008xh)", welcher ebenfalls als Funktionsbaustein in den meisten LinMot Bibliotheken verfügbar ist.

## 1.5 Start der Command Table mit EasySteps

Wenn die optionale Zusatzapplikation "EasySteps" auf dem Regler installiert ist, kann die Command Table auch damit gestartet werden.

Details zu dieser Applikation finden Sie im zugehörigen Handbuch  
<http://shop.linmot.com/E/product/0185-1037-E>

Mittels EasySteps können bestimmte digitale Eingänge zum Aufrufen einer Command Table Zeile konfiguriert werden (Eingänge sind abhängig vom Drive).  
 Dies kann an Stelle der Betriebsart "Triggered Command Table" verwendet werden.



**Achtung:** Die Verwendung von EasySteps erfordert eine kurze, nicht blockierende Command Table! Stellen Sie sicher, dass die Command Table beendet ist, bevor Sie ein neues Startsignal geben, um unvorhersehbares Verhalten zu vermeiden!



**Achtung:** Command Table muss enden! Bilden Sie keine dauerhaften Schleifen mittels „if“ oder „Wait“ Bedingungen!

### Beispiel: Drei kurze Bewegungsabläufe, startbar per EasySteps

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Start Seq 1	VAl Go To Pos	Pos: 20 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					2 (Unnamed)
2	Unnamed	Wait until Motion Finished									3 (End Seq 1)
3	End Seq 1	VAl Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					None
4											
5	Start Seq 2	VAl Go To Pos	Pos: 100 mm	Vel: 0.1 m/s	Acc: 1 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					6 (Unnamed)
6	Unnamed	Wait until Motion Finished									7 (End Seq 2)
7	End Seq 2	VAl Go To Pos	Pos: 50 mm	Vel: 1 m/s	Acc: 1 m/s <sup>2</sup>	Dec: 1 m/s <sup>2</sup>					None
8											
9											
10	Start Seq 3	VAl Go To Pos	Pos: 20 mm	Vel: 0.1 m/s	Acc: 1 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					11 (Unnamed)
11	Unnamed	Wait until Motion Finished									12 (End Seq 3)
12	End Seq 3	VAl Go To Pos	Pos: 50 mm	Vel: 1 m/s	Acc: 1 m/s <sup>2</sup>	Dec: 1 m/s <sup>2</sup>					None
13											
14											

### Konfiguration von EasySteps IO Interface (Einige Eingänge unterstützen diese Funktion, Abhängig vom Drive):

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Eingang Xxx Funktion	\Parameters\EasySteps\IO Motion\Input X4/X14 Config\			
X4.5 Rising Edge Function (nicht B1100)	0Ch – Eval Command Table Command	-	3600h	Starte Command Table Command
X14.2 Rising Edge Function (B1100)	0Ch – Eval Command Table Command	6418h	-	Starte Command Table Command

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Xxx IO Motion Konfiguration	\Parameters\EasySteps\IO Motion\Input X4/X14 Config\ IO Motion Config			
Curve/CmdID	Entry ID	6419h	3620h	Startzeilen-ID für Command Table Ausführung

Diese Einstellungen zeigen exemplarisch nur einen Eingang. Je nach verwendeter Drive-Hardware stehen weitere Eingänge zur Verfügung, die auf gleiche Weise konfiguriert werden können.

Für jede Bewegungssequenz der oben sichtbaren Command Table muss ein Eingang in EasySteps konfiguriert werden. Diese Eingänge starten dann die jeweilige Sequenz.



## 1.6 Verwendung des Command Table I/O Interface (nur auf E1100-GP Drives)

Das Command Table I/O Interface ist eine spezielle Zusatzanwendung auf E1100-GP Drives. Diese Zusatzanwendung erlaubt es, jede einzelne Zeile der Command Table über ein Bitmuster auf der Erweiterungs-I/O X6 zu starten.

Details zu dieser Zusatzanwendung finden Sie im zugehörigen Handbuch

<http://shop.linmot.com/D/product/0185-1077-E>

## 2 Bearbeiten der Command Table

Die Bearbeitung der Command Table erfolgt tabellarisch.  
Die Command Table befindet sich jeweils unten unter dem Parameterbaum der Drive-Konfiguration.  
Jede Command Table Zeile entspricht einer Befehlszeile.  
Jede Zeile beginnt mit einer einmaligen ID, welche der Zeilennummer entspricht.  
Diese Zeilennummer dient als Adresse für das Verketteten der Befehle oder als Sprungziel von «IF» Bedingungen.

Ein Rechtsklick mit der Maus auf einer Command Table Zeile öffnet ein Menü mit zusätzlichen Bearbeitungsfunktionen.

The screenshot shows the LinMot software interface for editing the Command Table. The top section is a configuration panel for a specific entry (ID 1). It includes fields for Entry ID, Entry Name (Unnamed), Motion Command Category (Most Commonly Used), Motion Command Type (VAI Go To Pos), Target Position (0 mm), Maximal Velocity (1 m/s), Acceleration (10 m/s<sup>2</sup>), and Deceleration (10 m/s<sup>2</sup>). There is a checkbox for "Auto execute new command on next cycle" and a field for "ID of Sequenced Entry" (2). An "Apply" button is located at the bottom right of the configuration panel.

Below the configuration panel is a table with columns: ID, Name, Type, Par 1, Par 2, and Par 3. The table contains 11 rows. Row 1 is highlighted, and a context menu is open over it. The context menu options are: Delete (Del), Copy (Ctrl+C), Cut (Ctrl+X), Paste (Ctrl+V), Create Entry, Delete All, and Show Object Inspector F1.

ID	Name	Type	Par 1	Par 2	Par 3
1	Unnamed	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

Mehrfache, hintereinander liegende Zeilen können durch Drücken der Taste "Umschalten" während der Auswahl mit der Maus angewählt werden.  
Einzelne, nicht hintereinander liegende Zeilen können durch Drücken der Taste «Strg» bzw. «Ctrl» markiert werden.

## 2.1 Einen Befehl eingeben

Wählen Sie eine Zeile durch Doppelklick aus. Die gewählte Zeile wird markiert und im oberen Bereich der Command Table erscheint ein Eingabebereich (rot markiert).

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1											
2											
3											
4											
5											

Die Zeilennummer ist fix und kann nicht verändert werden. Sie können einen Namen oder eine Beschreibung im Feld «Entry Name» eintragen. Dies erleichtert die Orientierung bei grösseren Sequenzen.

In der Auswahl "Motion Command Category" können Sie die verfügbaren Befehle nach Kategorien vorsortieren lassen, um die Länge der Liste "Motion Command Type" zu reduzieren.

Abhängig von der gewählten Kategorie können Sie unter "Motion Command Type" einen verfügbaren Befehl auswählen.

Abhängig vom gewählten Befehl ändern sich dann die weiteren Eingabefelder, um die befehlspezifischen Parameter ein zu fügen.

Wenn die aktuelle Command Table Zeile die letzte Zeile der Sequenz sein soll (die Ausführung der Command Table soll hier enden), wählen Sie nicht die Funktion "Auto execute new command on next cycle" an.

Wenn nach dieser Zeile ein neues oder weiteres Kommando ausgeführt werden soll, aktivieren Sie "Auto execute new command on next cycle" und geben Sie dort die gewünschte Zielzeilennummer ein, die als nächstes ausgeführt werden soll.

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Unnamed	VAl Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					2 (Not exist...)
2											
3											

Leere, adressierte Zeilen werden automatisch markiert!

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Unnamed	VAl Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					2 (Not exist...)
2											
3											

## 2.2 Löschen von Kommandos

Wählen Sie die Zeile oder Zeilen, die Sie löschen möchten mittels Maus und öffnen mit der rechten Maustaste das Zusatzmenü.  
Wählen Sie dort "Delete" (oder drücken Sie die Taste "del").



**Achtung:** Wenn Sie "Delete All" im Menü wählen, wird die gesamte Command Table gelöscht!

## 2.3 Zeilen verschieben / neue Zeile einfügen

Ein Verschieben von Zeilen kann notwendig werden, wenn nachträglich Platz für zusätzliche Befehle benötigt wird. Es ist nicht möglich, direkt eine neue Zeile ein zu fügen.  
Um Platz zu schaffen, markieren Sie die gewünschten Zeilen, schneiden diese aus und fügen diese an anderer Stelle ein (Rechtsklick für Zusatzmenü, Cut / Paste).  
Mehrfache zusammenhängende Zeilen können als Block verarbeitet werden.  
Um die Zeilen einfügen zu können, muss der gewünschte Bereich ausreichende Leerzeilen aufweisen!  
Adressierte ID's speziell in "IF" Bedingungen werden nicht automatisch neu nummeriert! Dies muss manuell korrigiert werden!



**Achtung:** Eingelegte Zeilen (nach copy oder cut) werden nicht automatisch passend mit der Sprung-ID oder bei «IF» Bedingungen nummeriert!  
Dies muss manuell korrigiert werden!

## 2.4 Speichern der Command Table

Wenn die Command Table verändert wurde, muss diese neu auf den Drive gespeichert werden (Download).  
Nach einer Änderung erscheint ein Hinweis, dass sein Download nötig ist:

The screenshot shows the LinMot software interface. At the top, there are input fields for "Entry ID:" (value: 1), "Entry Name:" (value: StopCT), "Motion Command Category:" (value: All), and "Motion Command Type:" (value: Clear Event Evaluation). There is a checkbox for "Auto execute new command on next cycle" which is unchecked. Below these fields is an "Apply" button. At the bottom of the interface, there are two buttons: "Upload from Drive" and "Download to Drive". A red box highlights a warning message: "Command Table modified. Please download to drive." Below the warning is a table with the following structure:

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	F
1	StopCT	Clear Event Evaluation					
2							
3							
4							
5							

Während des Speicherns muss die Firmware gestoppt werden. Wenn das Speichern beendet ist, kann der Drive neu gestartet werden.

## 2.5 Laden der Command Table

Während dem Einloggen auf den Drive wird eine bestehende Command Table automatisch in LinMot Talk geladen. In manchen Fällen ist es hilfreich, die Command Table manuell in LinMot Talk zu laden.  
Per SPS können bei Bedarf beliebige Parameter einer Command Table Zeile verändert werden. Bei einer Fehlersuche kann das Laden der Command Table hilfreich sein, um die geänderten Werte zu kontrollieren.

### 3 Command Table Programmierung

Abhängig vom verwendeten Drive stehen verschiedene Befehle für die Verwendung in der Command Table zur Verfügung. Jeder vom Drive unterstützte Fahrbefehl kann verwendet werden.

Eine Übersicht der unterstützten Befehle finden Sie in folgenden Handbüchern:

Für B1100/E1100 Drives: <http://shop.linmot.com/E/product/0185-1092-E>

Für alle anderen Drives: <http://shop.linmot.com/E/product/0185-1093-E>

Die Drives bieten zusätzliche Befehle zur Verwendung in der Command Table. Diese Befehle sind ausschliesslich für die Command Table vorgesehen und dienen zur Ablaufsteuerung, Bedingungen, Parameterzugriff und I/O Zugriff.

## 3.1 Bedingungen

In der Command Table stehen verschiedene Bedingungen zur Ablaufsteuerung zur Verfügung.

Diese setzen sich im Wesentlichen aus zwei Gruppen zusammen:

- Wartende Bedingungen (Die Bedingung wartet auf ein Ereignis, bevor die angegebene Zeile aufgerufen wird)
- "If" Bedingungen (Sprung in entsprechende Zeilen, je nach Ergebnis Wahr oder Falsch, nicht wartend)

### 3.1.1 Wartende Bedingungen

Für einfache Bewegungsabläufe sind wartende Bedingungen zur Ablaufsteuerung ausreichend. Diese wartenden Bedingungen starten die angegebene Zeile, wenn der Bedingungszustand erreicht ist.

Zum Beispiel: Fahre zu Position A, wenn die Bewegung abgeschlossen ist, fahre zu Position B:

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Position A	VAl Go To Pos	Pos: 10 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					2 (Motion Finished?)
2	Motion Finished?	Wait until Motion Finished									3 (Position B)
3	Position B	VAl Go To Pos	Pos: 35 mm	Vel: 0.1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					None
4											

Die Ausführung beginnt in Zeile 1, wodurch der Motor auf Position 10mm fährt. Die Command Table fährt direkt in Zeile 2 fort.

Während der Motor fährt, wartet die Bedingung in Zeile 2, bis der VA Interpolator abgeschlossen ist (Bewegungskurve beendet). Dann wird Zeile 3 aufgerufen, in welcher der Motor auf die nächste Position fährt. Die Ausführung endet in dieser Zeile während der Motor noch fährt, bis das Ziel erreicht ist.



**Hinweis:** Für einfache Bewegungsabläufe verwenden Sie wartende Bedingungen. Ohne wartende Bedingungen startet die Command Table nach Aufruf des angegebenen Befehls sofort die nächste angegebene Zeile, welche den aktuell ablaufenden Befehl überschreibt!  
Im oberen Beispiel hätte das Entfernen der wartenden Bedingung zur Folge, dass zuerst Position A gestartet wird und sofort durch den Fahrbefehl auf Position B überschrieben wird. Man beobachtet als Resultat lediglich die Fahrt auf Position B.



**Hinweis:** Unterschied zwischen: Wait until motion finished – Wait until in target position  
Die erste Bedingung wartet, bis die berechnete Sollkurve abgefahren ist, die zweite Bedingung wartet, bis die angegebene Endposition im Zielfenster von "Status In Target Position" angekommen ist.

Liste der wartenden Bedingungen:

- Wait Time (210xh)
- Wait Until Motion Finished (211xh)
- Wait Until In Target Position (212xh)
- Wait Until Rising Trigger Event (213xh)
- Wait Until Falling Trigger Event (214xh)
- Wait Until Demand Position Greater Than (220xh)
- Wait Until Demand Position Less Than (221xh)
- Wait Until Actual Position Greater Than (222xh)
- Wait Until Actual Position Less Than (223xh)
- Wait Until Difference Position Greater Than (224xh)
- Wait Until Difference Position Less Than (225xh)
- Wait Until Difference Position Unsigned Greater Than (226xh)
- Wait Until Difference Position Unsigned Less Than (227xh)
- Wait Until Demand Velocity Greater Than (228xh)
- Wait Until Demand Velocity Less Than (229xh)
- Wait Until Actual Velocity Greater Than (22Axh)
- Wait Until Actual Velocity Less Than (22Bxh)
- Wait Until Current Greater Than (22Exh)
- Wait Until Current Less Than (22Fhx)



**Achtung:** Prüfen Sie mit dem Handbuch Motion Control Software, ob der gewünschte Befehl auf dem Drive unterstützt wird!

## 3.1.2 If Bedingungen



**Hinweis:** Wenn IF Bedingungen verwendet werden, setzen Sie die Sprungadressen niemals auf die Zeile der If Bedingung. Dies führt zu einem «Trap Class A» Fehler!  
 Alle Command Table Zeilen werden in einem Interface-Zyklus ausgeführt. Dies sichert eine konstante Ausführungszeit, erlaubt aber nicht, den direkten Aufruf der If-Befehlszeile aus dieser If-Bedingung heraus. Schleifen mit If Bedingungen müssen entkoppelt werden!

Für bestimmte Anwendungen ist es hilfreich, IF-Bedingungen in der Command Table zu verwenden. Zum Beispiel soll eine IF-Bedingung während einer Positionierfahrt den Motorstrom überwachen, um das Anfahren eines Zielobjekts durch den Motorstromanstieg zu detektieren.

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Goto A	VAI Go To Pos	Pos: 100 mm	Vel: 0.1 m/s	Acc: 1 m/s <sup>2</sup>	Dec: 1 m/s <sup>2</sup>					2 (Check Current)
2	Check Current	IF Current Greater Than	Val: 5 A	True Cmd ID: 10 (Current >)	False Cmd ID: 3 (Loop, current < )						None
3	Loop, current <	No Operation									2 (Check Current)
4											
5											
6											
7											
8											
9											
10	Current >	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					None

Zeile 1 startet eine Fahrt auf Position 100mm und springt in Zeile 2.

Zeile 2 prüft, ob der Motorstrom grösser als 5A ist (wenn das Auftreffen auf das Objekt auftritt). Ist der Strom kleiner 5A, so wird in Zeile 3 gesprungen, ansonsten in Zeile 10.

Zeile 3 dient als Rückkopplung auf Zeile 2, da kein direkter Sprung innerhalb der If-Bedingung auf sich selbst erlaubt ist.

Zeile 3 entkoppelt die If-Bedingung.

Zeile 10 führt dann weitere gewünschte Funktionen aus, hier im Beispiel fährt der Motor wieder zurück auf 0mm und die Command Table endet.

Liste der IF Bedingungen:

- IF Cmd Table Var 1 Less Than (250xh)
- IF Cmd Table Var 1 Greater Than (251xh)
- IF Cmd Table Var 2 Less Than (252xh)
- IF Cmd Table Var 2 Greater Than (253xh)
- IF Cmd Table Var 1 Less Than UPID Value (256xh)
- IF Cmd Table Var 2 Less Than UPID Value (257xh)
- IF Demand Position Less Than (258xh)
- IF Demand Position Greater Than (259xh)
- IF Actual Position Less Than (25Axh)
- IF Actual Position Greater Than (25Bxh)
- IF Difference Position Less Than (25Cxh)
- IF Difference Position Greater Than (25Dxh)
- IF Current Less Than (25Exh)
- IF Current Greater Than (25Fhx)
- IF Analog Val On X4.4 Less Than (260xh)
- IF Masked X4 Input Value Equal Than (262xh) (E1100, E1200, C1100, C1200, C1400, E1400)
- IF Masked X14 Input Value Equal Than (262xh) (B1100)
- IF Masked X6 Input Value Equal Than (263xh) (only E1100-GP)
- IF Masked Status Word Equal Than (264xh)
- IF Masked Warn Word Equal Than (265xh)
- IF CAM Counts Less Than (266xh)



**Achtung:** Prüfen Sie mit dem Handbuch Motion Control Software, ob der gewünschte Befehl auf dem Drive unterstützt wird!

## 3.2 Variable / Parameter-Zugriff

Die Command Table bietet zwei Variablen zur temporären Datenspeicherung. Der Datentyp wird automatisch von der verwendeten Datenquelle übernommen.

Diese beiden Variablen können mit Werten gesetzt werden, sie können Inkrementiert und auch als Bedingungen von If-Bedingungen verwendet werden.

Falls ein Schleifenzähler benötigt wird, kann dies mit einer der beiden Variablen realisiert werden.

Die Variable kann auch über die UPID-Adresse mit einem RAM-Wert geladen werden. Umgekehrt kann man den Wert der Variablen auf eine UPID-Adresse schreiben.

Die beiden Variablen können mittels LinMot Talk in der Kategorie "Variables – Command Table" beobachtet werden.

Dieses Beispiel zeigt einen Schleifenzähler, der die Anzahl zu absolvierender Bewegungszyklen zählt:

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Init Var 1	Set Cmd Table Var 1 To	Set Val: 0								2 (Move to A)
2	Move to A	VAl Go To Pos	Pos: 5 mm	Vel: 1 m/s	Acc: 10 m/s^2	Dec: 10 m/s^2					3 (Pos A reached?)
3	Pos A reached?	Wait until Motion Finished									4 (Move to B)
4	Move to B	VAl Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s^2	Dec: 10 m/s^2					5 (B reached?)
5	B reached?	Wait until Motion Finished									6 (Increment +1)
6	Increment +1	Add To Cmd Table Var 1	Add Val: 1								7 (Loops done?)
7	Loops done?	IF Cmd Table Var 1 Less T...	Val: 10	True Cmd ID: 2 (Move to ...	False Cmd ID: 8 (Finished)						None
8	Finished	No Operation									None

In diesem Beispiel wird die Command Table in Zeile 1 gestartet. Dort wird der Zähler mit Variable 1 auf „0“ gesetzt.

Zeilen 2 bis 5 stellen den Bewegungszyklus dar, welche direkt nach Zeile 1 gestartet werden.

In Zeile 6 wird Variable 1 um „+1“ inkrementiert.

Zeile 7 prüft, ob Variable 1 kleiner 10 ist, wenn ja, wird in Zeile 2 gesprungen wenn nein in Zeile 8.

Dadurch wird der Ablauf (Zeile 2-5) 10 mal ausgeführt.

Liste der Befehle für Variable / Parameter-Zugriff:

- Set Cmd Table Var 1 To (240xh)
- Add To Cmd Table Var 1 (241xh)
- Set Cmd Table Var 2 To (242xh)
- Add To Cmd Table Var 2 (243xh)
- Write Cmd Table Var 1 To UPID RAM value (248xh)
- Write Cmd Table Var 2 To UPID RAM value (249xh)
- Write UPID RAM Value To Cmd Table Var 1 (24Cxh)
- Write UPID RAM Value To Cmd Table Var 2 (24Dxh)
- Write UPID RAM value To UPID ROM value (24Exh)



**Hinweis:** Wenn eine Art „Count Down“ nötig ist, inkrementieren Sie mit einem negativen Wert (z.B. „-1“).



**Achtung:** Prüfen Sie mit dem Handbuch Motion Control Software, ob der gewünschte Befehl auf dem Drive unterstützt wird!



### 3.3 Zugriff auf I/O

Mittels Command Table kann auch auf die digitalen Ein- und Ausgänge des Drives zugegriffen werden. Um einen Eingang zu lesen, ist keine zusätzliche Konfiguration nötig. Wenn ein Ausgang gesteuert werden soll, so muss dieser Ausgangspin als „Interface Output“ konfiguriert werden, damit die Command Table diesen Ausgang steuern kann.

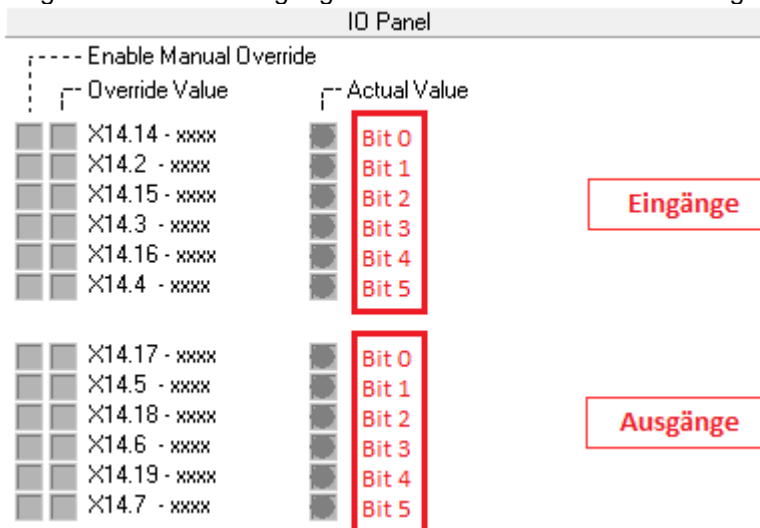
Konfiguration eines Ausgangs zur Verwendung mit der Command Table:

Funktion	Wert	UPID (B1100)	UPID (alle anderen)	Beschreibung
Definiere Ausgang	\Parameters\Motion Control SW\Drive Configuration\X4/X14 IO Definitions\			
IO X4.3 (not B1100)	0Eh – Interface Output	-	1036h	Setze X4.3 als Ausgang
IO X14.17 (B1100)	0Eh – Interface Output	62EEh	-	Setze X14.17 als Ausgang

Um einen Eingang mit einer IF Bedingung zu verwenden, oder einen Ausgang zu setzen, ist eine Bitmaske notwendig, um die gewünschte Ein/Ausgangsleitung zu adressieren.

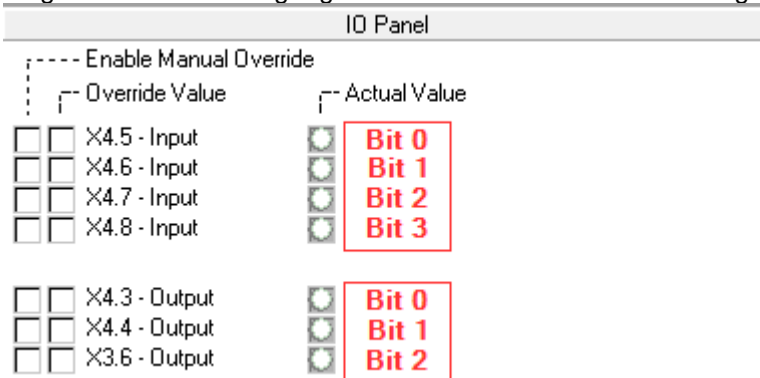
B1100 und C Serien Drives haben getrennte Ein- und Ausgänge, während die Drives der E-Serie Variable Ein-/oder Ausgänge (I/O) haben. Ohne Konfiguration an den I/O's kann die Signalleitung immer als Eingang gelesen werden.

Folgende Ein- und Ausgänge sind auf dem Drive B1100 verfügbar:



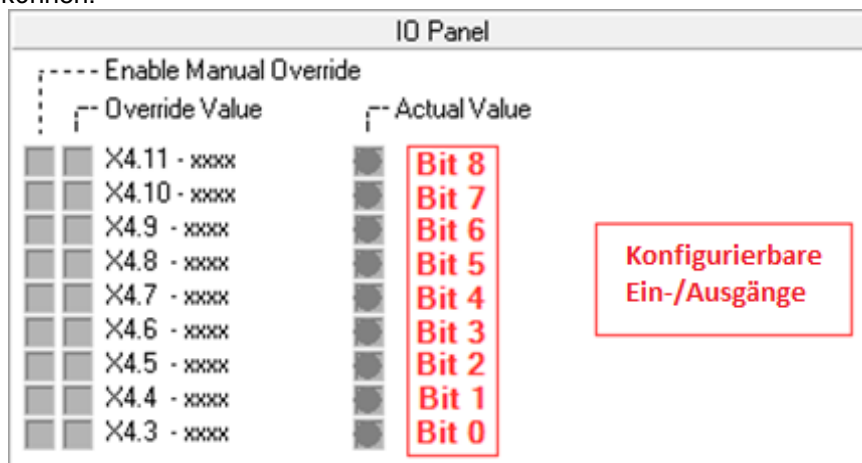
Bitpositionszählung erfolgt von oben nach unten.

Folgende Ein- und Ausgänge sind auf dem Drive C1X00 verfügbar:



Bitpositionszählung erfolgt von oben nach unten.

Alle E1XX0 Serien Drives bieten I/O Signale, welche wahlweise als Eingang oder Ausgang konfiguriert werden können:



Bitpositionsählung erfolgt von unten nach oben. Je nach Drive variiert die Anzahl der verfügbaren I/O bis X4.12 (Bit 9).

### 3.3.1 Erstellen der Bit-Maske

Erstellen einer Bit-Maske bedeutet, das gewünschte Bit im Maskier-Byte (Adresse) zu setzen. Dies kann mit einem einfachen mathematischen Ausdruck beschrieben werden:

$$\text{Bitmasken} - \text{Wert} = 2^{\text{Bitnummer}}$$

Beispiel Eingänge B1100:

$$\text{Bit-Maske für X14.3, Bit Nummer 4} \rightarrow 2^4 = 16 = 10h = 0001\ 0000$$

Beispiel Ausgänge B1100:

$$\text{Bit Maske für X14.17, Bit Nummer 0} \rightarrow 2^0 = 1 = 1h = 0000\ 0001$$

$$\text{Bit Maske für X14.6, Bit Nummer 4} \rightarrow 2^4 = 16 = 10h = 0001\ 0000$$

Eine Bitmaske erlaubt das Auswählen mehrerer Bits zur gleichen Zeit. Dazu errechnet man die einzelnen Bitmasken-Werte und addiert diese.

Beispiel E1100:

$$\text{Bit Maske für X4.4, Bit Nummer 1} \rightarrow 2^1 = 2 = 2h = 0000\ 0010$$

$$\text{Bit Maske für X4.9, Bit Nummer 6} \rightarrow 2^6 = 64 = 40h = 0100\ 0000$$

$$\text{Summe: Bit mask} = 2^1 + 2^6 = 2 + 64 = 66 = 42h = 0100\ 0010$$

Diese Bitmaske adressiert X4.4 und X4.9. Eine Bitmaske stellt auch eine logisch UND-Verknüpfung dar.

Alle Bitnummern für I/O Signale starten mit der Bitnummer 0 für X4.3 und steigen bis zur letzten I/O an. Bei der B1100 startet das Zählen der Bits mit Nummer 0 im Control Panel von oben nach unten (siehe Screenshot).

Eingänge und Ausgänge werden als einzelne Blöcke dargestellt, I/O's werden gemeinsam angezeigt.

### 3.3.2 Abfragen eines digitalen Eingangs

Einen digitalen Eingang kann man z.B. mit der IF-Bedingung "IF Masked X4 Input Value Equal Than (262xh)" abfragen.

Hierzu wird die Bitmaske und ein Vergleichswert benötigt, um den Zustand des Signals aus zu werten.

Beispiel: Prüfe, ob X4.5 auf dem E1100 Drive gesetzt ist:

X4.5 Bitwert = 2 ->  $2^2 = 4 = 04h = 0000\ 0100$

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Start	IF Masked X4 Input Value ...	X4 Mask: 0004h	X4 Bit Values: 0004h	True Cmd ID: 10 (Do some...	False Cmd ID: 2...					None
2	loop	No Operation									1 (Start)
3											
4											
5											
6											
7											
8											
9											
10	Do something	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					None

Zeile 1 prüft, ob der maskierte Eingang mit dem Bit Wert übereinstimmt (der Bit Wert ist Gleich der Bitmaske), um in Zeile 10 zu springen, ansonsten wird über Zeile 2 wieder Zeile 1 gestartet.

Beispiel: Prüfe ob X4.5 gesetzt ist und X4.7 nicht gesetzt ist auf dem E1100 Drive:

Bit-Maske: =  $Bitmask = 2^2 + 2^4 = 20 = 14h = 0001\ 0100$

Vergleichswert:  $CompareValue = 1 * 2^2 + 0 * 2^4 = 4 = 04h = 0000\ 0100$

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Start	IF Masked X4 Input Value ...	X4 Mask: 0014h	X4 Bit Values: 0004h	True Cmd ID: 10 (Do some...	False Cmd ID: 2...					None
2	loop	No Operation									1 (Start)
3											
4											
5											
6											
7											
8											
9											
10	Do something	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					None
11											

Diese IF Bedingung wechselt zu Zeile 10, wenn X4.5 gesetzt ist und X4.7 nicht gesetzt ist.

### 3.3.3 Setzen eines Ausgangs

Folgende Befehle können zum Setzen eines Ausgangs verwendet werden:

- Write X4 Intf Outputs with Mask (003xh) (E1100, E1200, C1100, C1200, C1400, E1400)
- Write X14 Intf Outputs with Mask (003xh) (B1100)
- Write X6 Intf Outputs with Mask (004xh) (nur verfügbar auf E1100-GP Drives)



**Achtung:** Stellen Sie sicher, dass der gewünschte Ausgang als "Interface Output" konfiguriert ist!

Um ein Ausgang zu setzen, wird wieder eine Bitmaske und ein Bitwert benötigt, ähnlich wie beim Lesen eines Eingangs.

Beispiel: Setze X4.5 als Ausgang auf E1100 Drive:

X4.5 Bit Maske = 2 ->  $2^2 = 4 = 04h = 0000\ 0100$

X4.5 Bit Wert = 2 ->  $2^2 = 4 = 04h = 0000\ 0100$

Entry ID:   Auto execute new command on next cycle

Entry Name:

Motion Command Category:

Motion Command Type:

X4 Write Bit Mask:

X4 Write Bit Values:

---

Upload from Drive     Download to Drive

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	Set X4.5	Write X4 Intf Outputs with Mask	X4 Write Bit Mask: 0004h	X4 Write Bit Values: 0004h							2 (Wait)
2	Wait	Wait Time	Time: 1000 ms								3 (Reset)
3	Reset	Write X4 Intf Outputs with Mask	X4 Write Bit Mask: 0004h	X4 Write Bit Values: 0000h							None

Dieses Beispiel setzt X4.5 auf +24V, wartet 1s und setzt X4.5 wieder zurück auf Masse und beendet die Command Table.

Beispiel: Setzen der Signale X14.17, X14.5 und X14.18 auf dem Drive B1100 als Ausgänge:

Bit-Maske:  $Bitmask = 2^0 + 2^1 + 2^2 = 1 + 2 + 4 = 7 = 07h = 0000\ 0111$

Die Command Table setzt die Ausgänge in einer Ablaufsequenz.

Zuerst wird X14.17 gesetzt, nach einer Sekunde folgt X14.5, dann nach einer weiteren Sekunde wird X14.18 gesetzt und letztlich nach einer weiteren Sekunde wird X14.5 zurück gesetzt. Anschliessend werden alle Ausgänge zurück gesetzt.

Für diesen Ablauf werden verschiedene Bit-Werte benötigt:

Bit-Wert für X14.17:  $Bitvalue = 2^0 = 1 = 01h = 0000\ 0001$

Bit-Wert für X14.17 und X14.5:  $Bitvalue = 2^0 + 2^1 = 1 + 2 = 3 = 03h = 0000\ 0011$

Bit-Wert für X14.17 und X14.5 und X14.18:  $Bitvalue = 2^0 + 2^1 + 2^2 = 1 + 2 + 4 = 7 = 07h = 0000\ 0111$

Bit-Wert für X14.17 und nicht X14.5 und X14.18:  $Bitvalue = 1 * 2^0 + 0 * 2^1 + 1 * 2^2 = 1 + 0 + 4 = 5 = 05h = 0000\ 0101$

Bit-Wert alle Ausgänge aus: 0

Entry ID:   Auto execute new command on next cycle

Entry Name:  ID of Sequenced Entry:  (=> wait 1s)

Motion Command Category:

Motion Command Type:

X14 Write Bit Mask:

X14 Write Bit Values:

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	Par 5	Par 6	Par 7	Par 8	Sequenced Entry
1	X14.7	Write X14 Intf Outputs with Mask	X14 Write Bit Mask: 0007h	X14 Write Bit Values: 0001h							2 (wait 1s)
2	wait 1s	Wait Time	Time: 1000 ms								3 (X14.17 & X14...
3	X14.17 & X14.5	Write X14 Intf Outputs with Mask	X14 Write Bit Mask: 0007h	X14 Write Bit Values: 0003h							4 (wait 1s)
4	wait 1s	Wait Time	Time: 1000 ms								5 (all on)
5	all on	Write X14 Intf Outputs with Mask	X14 Write Bit Mask: 0007h	X14 Write Bit Values: 0007h							6 (wait 1s)
6	wait 1s	Wait Time	Time: 1000 ms								7 (X14.5 off)
7	X14.5 off	Write X14 Intf Outputs with Mask	X14 Write Bit Mask: 0007h	X14 Write Bit Values: 0005h							8 (wait 1s)
8	wait 1s	Wait Time	Time: 1000 ms								9 (all off)
9	all off	Write X14 Intf Outputs with Mask	X14 Write Bit Mask: 0007h	X14 Write Bit Values: 0000h							None

## 4 Beispiele

Dieses Kapitel zeigt einige einfache Beispiele zur Verwendung der Command Table. Alle hier gezeigten Beispiele sind für die E1100-GP Drive-Serie mit installierter EasySteps-Applikation erstellt.

Für jedes Beispiel wird vorausgesetzt, dass sich der Drive auf Werkseinstellungen befindet und der Motor-Wizard ausgeführt wurde.

### 4.1 Beispiel 1: Gesteuertes Ausschalten

Dieses Beispiel verwendet EasySteps X4.4, um die Command Table in Zeile 10 zu starten.  
 (Die Verwendung von EasySteps ist in einem separaten Handbuch beschrieben:  
[http://shop.linmot.com/D/ag9200.sw.h/dokumentation/software/drive-sw-handb%  
 Das Ein- und Ausschalten des Motors wird über das Control Word Bit 0 realisiert \(z.B. per Control Panel oder per I/O\).](http://shop.linmot.com/D/ag9200.sw.h/dokumentation/software/drive-sw-handb%c3%bccher/handbuch-0185-1037-(en):.htm)

In diesem Beispiel fährt der Motor auf die Startposition, wenn X4.4 aktiviert wird.  
 Zeile 10 prüft, ob im Status Word Bit 0 gesetzt ist, also ob der Drive betriebsbereit ist.  
 Während der Bewegung zur Startposition wird nicht geprüft, ob der Drive ausgeschaltet wird!  
 Nach dieser Initialisierung wird Zeile 20 gestartet. Hier wird nochmals geprüft, ob der Drive betriebsbereit ist.  
 Wenn nicht, wird in Zeile 1 gewechselt, wo die Command Table beendet wird.  
 Ansonsten wird in Zeile 23 geprüft, ob der Motor in Bewegung ist. Solange der Motor noch fährt, wird zurück in Zeile 21 gewechselt, ansonsten wird in Zeile 25 für die nächste Bewegung gewechselt.  
 Für diese weitere Bewegung wird die gleiche Abfrage durchgeführt, wie bei der vorigen Bewegung, anschliessend wird wieder in Zeile 20 gewechselt.  
 Solange das Control Word Bit 0 «Switch On» aktiv ist, fährt der Motor zwischen diesen beiden Positionen hin und her. Wird das Control Word Bit 0 auf 0 gesetzt, beendet sich die Command Table selbstständig.  
 Wenn das Bit 0 im Control Word wieder auf 1 gesetzt wird, kann die Bewegung wieder mittels Trigger auf X4.4 gestartet werden.  
 In diesem Ablauf wird dafür gesorgt, dass beim Ausschalten des Drives kein neues Fahrkommando mehr aufgerufen wird, was sonst zu einem Fehler führen würde.

Entry ID:   Auto execute new command on next cycle

Entry Name: 

Motion Command Category:

Motion Command Type:

Status Word Bit Mask:

Status Word Compare Value:

Table Command ID IF TRUE:  (=> Init Axis)

Table Command ID IF FALSE:  (=> loop)

Upload from Drive     Download to Drive

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	P.	I	F	P.	Sequenced Entry
1	StopCT	Clear Event Evaluation									None
2											
3											
4											
5											
6											
7											
8											
9											
10	Operational?	IF Masked Status Word Equal Than	Status Mask: 0001h	Status Value: 0001h	True Cmd ID: 12 (Init Axis)	False Cmd ID: 11 (loop)					None
11	loop	No Operation									10 (Operational?)
12	Init Axis	VAI Go To Pos	Pos: 0 mm	Vel: 0.1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					13 (Init Done?)
13	Init Done?	Wait until Motion Finished									20 (Move A)
14											
15											
16											
17											
18											
19											
20	Move A	VAI Go To Pos	Pos: 50 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					21 (StillOperatio...
21	StillOperational	IF Masked Status Word Equal Than	Status Mask: 0001h	Status Value: 0000h	True Cmd ID: 1 (StopCT)	False Cmd ID: 22 (dummy)					None
22	dummy	No Operation									23 (StillMoving?)
23	StillMoving?	IF Masked Status Word Equal Than	Status Mask: 2000h	Status Value: 2000h	True Cmd ID: 24 (Loop)	False Cmd ID: 25 (Move B)					None
24	Loop	No Operation									21 (StillOperatio...
25	Move B	VAI Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s <sup>2</sup>	Dec: 10 m/s <sup>2</sup>					26 (StillOperatio...
26	StillOperational	IF Masked Status Word Equal Than	Status Mask: 0001h	Status Value: 0000h	True Cmd ID: 1 (StopCT)	False Cmd ID: 27 (dummy)					None
27	dummy	No Operation									28 (StillMoving?)
28	StillMoving?	IF Masked Status Word Equal Than	Status Mask: 2000h	Status Value: 2000h	True Cmd ID: 29 (Loop)	False Cmd ID: 20 (Move A)					None
29	Loop	No Operation									26 (StillOperatio...
--											

## 4.2 Einfacher Schleifenzähler

Das folgende Command Table Beispiel führt 10 Mal eine Ein-und Ausfahrbewegung aus.

Das Beispiel zeigt, wie eine Command Table Variable dekrementiert wird.

In Zeile 1 wird Variable 1 mit der Anzahl Schleifendurchläufe (10) initialisiert.

Dann fährt der Motor aus und wieder ein (Zeilen 2-5).

In Zeile 6 wird die Command Table Variable 1 um den Wert 1 dekrementiert, das bedeutet, es wird der Wert «-1» addiert.

In Zeile 7 wird mit der IF-Bedingung abgefragt, ob die Variable 1 grösser als «0» ist. Wenn ja, erfolgt ein Sprung in Zeile 2 wenn nein, wird die Command Table in Zeile 8 beendet.

Dieses Beispiel wiederholt den Bewegungsablauf 10 mal.

Entry ID:   Auto execute new command on next cycle

Entry Name:  ID of Sequenced Entry:  (=> MoveOut)

Motion Command Category:

Motion Command Type:  ?

Set Value Of Var 1:

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	P..	I	F	P..	Sequenced Entry
1	PresetCounter	Set Cmd Table Var 1 To	Set Val: 10								2 (MoveOut)
2	MoveOut	VAl Go To Pos	Pos: 50 mm	Vel: 1 m/s	Acc: 10 m/s^2	Dec: 10 m/s^2					3 (Out?)
3	Out?	Wait until In Target Position									4 (MoveIn)
4	MoveIn	VAl Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s^2	Dec: 10 m/s^2					5 (In?)
5	In?	Wait until In Target Position									6 (Decrement V...
6	Decrement Var1	Add To Cmd Table Var 1	Add Val: -1								7 (Loop?)
7	Loop?	IF Cmd Table Var 1 Greater Than	Val: 0	True Cmd ID: 2 (MoveOut)	False Cmd ID: 8 (End)						None
8	End	No Operation									None

Allgemeine Konfiguration:

Keine spezifische Konfiguration, Command Table kann als "Triggered Command Table" (siehe Kapitel 1.2) gestartet werden.



### 4.3 Stromausfallsicheres Speichern einer Command Table Variable

Dieses Beispiel zeigt eine Möglichkeit, wie ein Wert stromausfallsicher auf dem Drive gespeichert werden kann.

Das Beispiel verwendet zwei digitale Eingänge in der Command Table.

Eingang X4.3 wird zum Starten einer 2-Positionen-Bewegung verwendet, X4.4 wird zum „Einlernen“ von einer der beiden Positionen verwendet. Im „Lernbetrieb“ kann die Position mit dem Jog-Modus +/- verändert werden. Das Einlernen ist nur verfügbar, wenn der Bewegungsablauf aktiv ist.

ID	Name	Type	Par 1	Par 2	Par 3	Par 4	P	F	Sequenced Entry
1	Init	Write UPID RAM value To Cmd Table Var 1	UPID: 13B5h (Spring Zero Position)						2 (Start)
2	Start	No Operation							3 (Start 4.3?)
3	Start 4.3?	IF Masked X4 Input Value Equal Than	X4 Mask: 0001h	X4 Bit Values: 0001h	True Cmd ID: 4 (Dummy)	False Cmd ID: 2 (Start)			None
4	Dummy	No Operation							5 (Teach?)
5	Teach?	IF Masked X4 Input Value Equal Than	X4 Mask: 0002h	X4 Bit Values: 0002h	True Cmd ID: 20 (StillTea...)	False Cmd ID: 6 (GotoPosTeached)			None
6	GotoPosTeach...	VAl Go To Cmd Tab Var1 Pos	Vel: 1 m/s	Acc: 10 m/s^2	Dec: 10 m/s^2				7 (Reached?)
7	Reached?	Wait until In Target Position							8 (MoveBack)
8	MoveBack	VAl Go To Pos	Pos: 0 mm	Vel: 1 m/s	Acc: 10 m/s^2	Dec: 10 m/s^2			9 (IsBack?)
9	IsBack?	Wait until In Target Position							2 (Start)
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20	StillTeaching?	IF Masked X4 Input Value Equal Than	X4 Mask: 0002h	X4 Bit Values: 0002h	True Cmd ID: 21 (loop)	False Cmd ID: 22 (Teach)			None
21	loop	No Operation							20 (StillTeachin...)
22	Teach	Write UPID RAM value To Cmd Table Var 1	UPID: 1B8Dh (Actual Position)						23 (StoreRAM)
23	StoreRAM	Write Cmd Table Var 1 To UPID RAM value	UPID: 13B5h (Spring Zero Position)						24 (StoreROM)
24	StoreROM	Write UPID RAM value To UPID ROM value	UPID: 13B5h (Spring Zero Position)						2 (Start)
25									
26									

Zeile 1 lädt die letzte gespeicherte Position in die Variable 1 der Command Table und geht über Zeile 2 in Zeile 3. In Zeile 3 prüft eine IF Bedingung Eingang X4.3, ob eine Bewegung laufen soll, oder nicht. Wenn X4.3 wahr ist, wird über Zeile 4 in Zeile 5 gewechselt, wo geprüft wird, ob der Lernmodus benötigt wird. Ist der Lernmodus nicht angefordert, wird über Zeile 5 die Zeile 6 gestartet, in der die Position von Variable 1 verwendet wird. Wenn diese Position erreicht wurde, fährt der Motor zu Position 0mm zurück und die Command Table springt in Zeile 2. Wurde der Lernmodus angefragt, so wird von Zeile 5 aus in Zeile 20 gewechselt. Ab Zeile 20 wird mit einer weiteren IF Bedingung eine Schleife gebildet, die solange ausgeführt wird, bis der Lernmodus über X4.4 deaktiviert ist. Vor dem deaktivieren muss mit den Jog+/- Signalen der Motor auf die gewünschte Position gefahren werden. Wenn X4.4 dann deaktiviert ist, wird die aktuelle Position in der Command Table Zeile 22 in Variable 1 gespeichert. Diese Variable wird nun in ein RAM Register «Spring Zero Position» geschrieben, welche im Parameterbaum «Position Controller Set B» liegt. Dies wird normalerweise meist nicht verwendet. Diese Variable wird als Speicher verwendet. Zeile 24 schreibt schliesslich diesen RAM Wert in den Flash, sodass der Wert stromausfallsicher ist. Bei jedem Neustart des Drives wird dieser Wert nun in den RAM geladen und durch Zeile 1 beim ersten Start ausgelesen.

#### Allgemeine Einstellungen:

Funktion	Wert	UPID	Beschreibung
Starteingang "Move"	\Parameters\Motion Control SW\Drive Configuration\X4/ IO Definitions\		
IO X4.3	00h – None	1036h	Verwende X4.3 als Eingang

Funktion	Wert	UPID	Beschreibung
Definiere Eingang "Lerne"	\Parameters\Motion Control SW\Drive Configuration\X4/ IO Definitions\		
IO X4.4	01h – Trigger (Input)	1039h	Verwende X4.4 als Eingang

Konfiguration JOG Modus je nach Notwendigkeit: \Parameters\Motion Control SW\State Machine Setup\Jogging\

Die Command Table wird im Modus "Command Table" ausgeführt, siehe Kapitel 1.3.



**Achtung:** Stellen Sie sicher, dass der Reglerparametersatz "Control Parameter Set B" für dieses Beispiel nicht in Verwendung ist!  
(\Parameters\Motion Control SW\Position Controller\Ctrl Par Set Selection muss auf Set A stehen)

## 4.4 Applikation Schaumgummi Test

Eine Maschine testet Schaumgummistreifen. Ein Linearmotor wird verwendet, um den Streifen mit einer Kraft von 40N zusammen zu drücken. Nach 2 Sekunden Presszeit wird der Streifen auf seine Abmaße hin kontrolliert.

Der Prüfablauf wird mittels Trigger-Signal gestartet. Wenn der Streifen innerhalb seines Maßes ist, wird dies durch ein digitales Ausgangssignal gemeldet. Gleiches gilt, wenn es nicht ok ist. Als Motor in diesem Beispiel wird hier der PS01-37x240 verwendet.

### Ablaufbeschreibung:

1. Motor fährt auf 40mm Position mit  $v = 3 \text{ m/s}$  und  $a = 5 \text{ m/s}^2$
2. Kraftreduktion auf 40 N und zusammendrücken des Streifens mit einer Geschwindigkeit von 0.05 m/s
3. Pressen für 2 Sekunden
4. Maß prüfen:  
Wenn der Linearmotor in seiner Ist-Position weiter als 65mm und kürzer als 75mm meldet, ist das Teil ok, ansonsten nicht.
5. Zurück zur Startposition bei 0mm mit  $v = 0.5 \text{ m/s}$  and  $a = 5 \text{ m/s}^2$

Das Triggersignal ist auf X4.6 verdrahtet. Wenn der Streifen innerhalb des Sollmaßes liegt, wird X4.8 aktiviert (OK), falls nicht OK wird X4.7 aktiviert (Defekt).

Um die Motorkraft auf 40N zu begrenzen, muss der zugehörige Motorstrom berechnet werden. Bei dem verwendeten Modell PS01-37x240 beträgt die Motorkraftkonstante 23.8 N/A, was einen Strom von 1.68 A für 40 N (40N / 23.8N/A) bedeutet.

Die Schleppfehlerüberwachung des Drives muss ausgeschaltet werden, damit der Motor bei Nichterreichen der Endposition beim Pressen nicht in den Fehlerzustand wechselt.

Allgemeine Einstellungen:

Funktion	Wert	UPID	Beschreibung
Definition Trigger Eingang	\Parameters\Motion Control SW\Drive Configuration\X4/ IO Definitions\		
IO X4.6	01h – Trigger (Input)	1039h	Setze X4.6 als Trigger
Trigger Mode*	01h – Direct	170Ch	Trigger Modus "Direct"

\* Die Triggereinstellungen (Direct, inhibited and/or delayed) und die zugehörigen Parameter können im Baum unter:  
\Parameters\Motion Control SW\Drive Configuration\X4 I/O Definitions\Trigger gefunden werden

Funktion	Wert	UPID	Beschreibung
Run mode configuration	\Parameters\Motion Control SW\Motion Interface\RunMode Settings\Run Mode Selection		
Triggered command table	0Ch – Triggered Command Table	1450h	Run Mode: Triggered Command Table

Funktion	Wert	UPID	Beschreibung
Command Table Startzeile	\Parameters\Motion Control SW\Motion Interface\RunMode Settings\Command Table Settings		
Rise Command Table entry ID	1	1486h	Setze Command Table Start Zeile

Funktion	Wert	UPID	Beschreibung
Fehler-Ausgang	\Parameters\Motion Control SW\Drive Configuration\X4 I/O Definitions		
Interface Output	0Eh – Interface Output	103Ah	X4.7 als Interface Output

Funktion	Wert	UPID	Beschreibung
OK Ausgang	\Parameters\Motion Control SW\Drive Configuration\X4 I/O Definitions		
Interface Output	0Eh – Interface Output	103Bh	X4.8 als Interface Output

Funktion	Wert	UPID	Beschreibung
Schleppfehlerüberwachung	\Parameters\Motion Control SW\Errors & Warnings>Error Detection Mask		
Position Lag Always	false	1587h	Schleppfehlerüberwachung ausschalten

Upload from Drive		Download to Drive		⚠ Command Table modified. Please download to drive.				
ID	Name	Type	Par 1	Par 2	Par 3	Par 4	F	Sequenced Entry
1	WaitForTrigger	Wait until Rising Trigger Edge						2 (Move Pos 40mm)
2	Move Pos 40mm	VAl 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 (Wait InPosition)
3	Wait InPosition	Wait until In Target Position						4 (Move Pos 90mm)
4	Move Pos 90mm	VAl 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 (Limit Force)
5	Limit Force	Write Live Parameter	UPID: 13A6h (...)	Value: 1.68 A				6 (ForceReached?)
6	ForceReached?	IF Current Greater Than	Val: 1.67 A	True Cmd ID: 10 (Press 2s)	False Cmd ID: 7 (dummy)			None
7	dummy	No Operation						8 (Pos > 99mm?)
8	Pos > 99mm?	IF Actual Position Greater Than	Val: 99 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 9 (dummy)			None
9	dummy	No Operation						6 (ForceReached?)
10	Press 2s	Wait Time	Time: 2000 ms					11 (Pos < 65mm?)
11	Pos < 65mm?	IF Actual Position Less Than	Val: 65 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 12 (dummy)			None
12	dummy	No Operation						13 (Pos > 75mm?)
13	Pos > 75mm?	IF Actual Position Greater Than	Val: 75 mm	True Cmd ID: 14 (Set Error X4.7)	False Cmd ID: 15 (Set O...			None
14	Set Error X4.7	Write Live Parameter	UPID: 1C89h (X...	Value: 00000010h				15 (Set OK X4.8)
15	Set OK X4.8	Write Live Parameter	UPID: 1C89h (X...	Value: 00000020h				17 (Move Pos 0mm)
16								
17	Move Pos 0mm	VAl 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 (Restore Force)
18	Restore Force	Write Live Parameter	UPID: 13A6h (...)	Value: 8 A				19 (Wait InPos)
19	Wait InPos	Wait until In Target Position						20 (Reset Outputs)
20	Reset Outputs	Write Live Parameter	UPID: 1C89h (X...	Value: 00000000h				1 (WaitForTrigger)

## 5 Dokumentversion

Version	Datum	Autor	Beschreibung
1V0	16.05.2018	mr	Initialversion

© 2018 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®  
Bodenaeckerstrasse 2  
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