# Applikation
# Adaption und Inbetriebnahme
# von Antrieben „BG...CI“ von Dunkermotoren über CANopen an TwinCAT

**Kurzfassung:** Beschreibung und Hinweise zu:

- CAN-Adresse, Baudrate
- Systemkonfiguration
- Achsparameter, Antriebsparameter
- Ansteuerung über TwinCAT

Warenzeichen und Warennamen sind ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Erstellung der Texte und Beispiele wurde mit großer Sorgfalt vorgegangen. Trotzdem können Fehler nicht ausgeschlossen werden. Die IEF Werner GmbH kann für fehlende oder fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Die IEF Werner GmbH behält sich das Recht vor, ohne Ankündigung die Software oder Hardware oder Teile davon, sowie die mitgelieferten Druckschriften oder Teile davon zu verändern oder zu verbessern.

Alle Rechte der Vervielfältigung, der fotomechanischen Wiedergabe, auch auszugsweise sind ausdrücklich der IEF Werner GmbH vorbehalten.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind wir jederzeit dankbar.

© Mai 2010, IEF Werner GmbH

## Inhaltsverzeichnis


**1 Änderungen 5**

**2 Allgemein 6**

**3 Vorabeinstellungen am dunMOT-Antrieb 7**

**4 SYSTEM-Konfiguration erstellen 8**

4.1 Baudrate des CAN-Gerätes „CIA“ einstellen 8

4.2 Anfügen von Geräten in der E/A-Konfiguration an den CAN-Bus „CIA“ (Dunkerantrieb BG...CI) 9

4.2.1 Einstellen der Adresse des Antriebes am CAN-Bus 11

4.3 Anfügen von Achsen in der NC-Konfiguration 12

4.4 Verknüpfen der NC-Achse mit dem Antrieb 14

4.5 Skalierungen und Faktoren 15

4.5.1 Encoder Skalierungsfaktor 15

4.5.2 Achse Drive Parameter 16

**5 Ansteuerung aus TwinCAT SPS 17**

5.1 ADS Parameter 17

5.2 Achsparameter 18

5.2.1 Geschwindigkeiten – Velocity 18

5.2.2 Beschleunigungen – Dynamics 19

5.2.3 Auflösung des Encoders – Encoder Scaling Factor 20

5.2.4 Anpassung der Ausgangswerte – Drive Output Scaling 21

5.2.5 Einstellungen für den Achsregler – Position Control Loop 22

5.3 TwinCAT SPS 23

5.3.1 Eingänge FB_dunMOT 24

5.3.2 Ausgänge FB_dunMOT 25

**6 Antriebsparameter 26**

6.1 Liste der Antriebsparameter 26

6.1.1 BG45x15CI – 24VDC 26

6.1.2 BG65x25CI – 24VDC 27

**7 Anhang 28**

7.1 CANopen-Mapping des dunMOT-Antriebes 28

# 1 Änderungen

Dokumentenänderungen und Lebenslauf

| Dokumentencode | Datum | Erstellung und Änderung |
|---|---|---|
| APP5024_DE_1092606_TwinCAT_dunMOT_R1a.doc | Mai 2010 | Neuerstellung dieses deutschen Dokuments. |

# 2 Allgemein

BG45 ... CI:

BG65 ... CI:

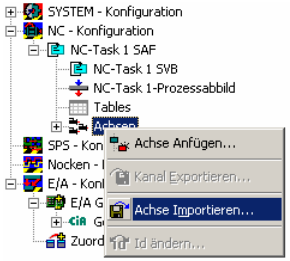
Die Antriebe „BG...CI“ sind mit einem CANopen-Interface ausgestattet. Über dieses Interface werden die Antriebe an das „Beckhoff-TwinCAT-Softwaresystem” adaptiert.

Dabei wird der Dunkerantrieb im Drehzahlreglermode (Velocity-Mode) unter einer „Kontinuierlichen Achse“ als NC-Achse in TwinCAT betrieben.

Zur einfachen Implementierung wurden Import-Dateien erstellt.

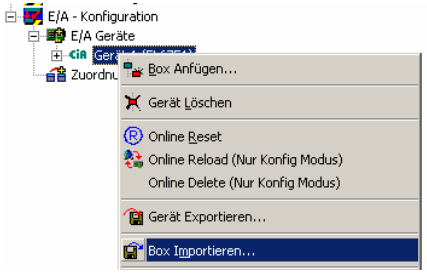
**Achse_dunMOT.tce:**

Wird im TwinCAT-System-Manager bei der „NC-Konfiguration“ beim Anfügen bzw. Importieren einer Achse verwendet.




**Drive_dunMOT.tce:**

Wird im TwinCAT-System-Manager bei der „E/A-Konfiguration“ beim Anfügen bzw. Importieren eines Gerätes an den CANopen verwendet.



# 3 Vorabeinstellungen am dunMOT-Antrieb

| Defaulteinstelllungen bei Auslieferung von dunkermotoren | | Bemerkung |
|---|---|---|
| Baudrate | 125 k Baud | Einstellung kann nur über den CAN-Bus verändert werden |
| CAN-ID (Adresse) | 127 | Einstellung kann nur über den CAN-Bus verändert werden |

Um die Baudrate zu ändern, setzen Sie bitte folgende Parameter:

| Index | Subindex | Wert | Beschreibung |
|---|---|---|---|
| 0x2000 | 1 | 0x6E657277 | Schreiben aktivieren |
| 0x2000 | 3 | Index der Baudrate | Neue Baudrate setzen (mit einer Zeitüberwachung von 2000ms) |

Den Index der Baudrate entnehmen sie bitte folgender Tabelle:

| Index | Baudrate |
|---|---|
| 0 | 1M Baud |
| 1 | 800k Baud |
| 2 | 500k Baud |
| 3 | 250k Baud |
| 4 | 125k Baud |
| 5 | 100k Baud |
| 6 | 50k Baud |
| 7 | 20k Baud |

Um eine Knotenadresse zu ändern, setzen Sie bitte folgende Parameter:

| Index | Subindex | Wert | Beschreibung |
|---|---|---|---|
| 0x2000 | 1 | 0x6E657277 | Schreiben aktivieren |
| 0x2000 | 2 | Knotenadresse | Neue Knotenadresse setzen (mit einer Zeitüberwachung von 2000ms) |

Siehe auch: „DunkerInformationsdatenbank.chm“, Beispiele - Kommunikation, AN002 - Node ID ändern, AN003 - Baudrate ändern

# 4 SYSTEM-Konfiguration erstellen

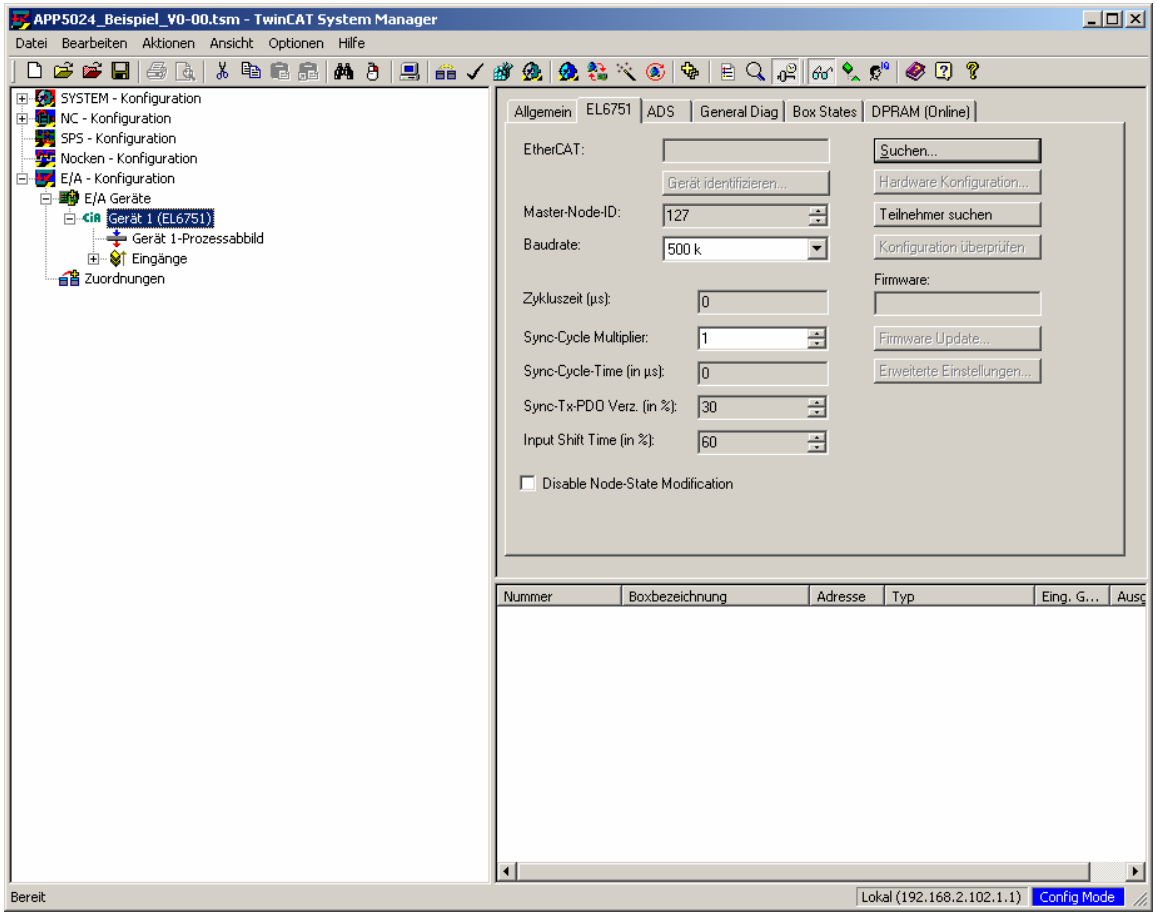
## 4.1 Baudrate des CAN-Gerätes „CIA“ einstellen

In diesem Beispiel wird als Baudrate für den CAN-BUS auf 500k eingestellt.

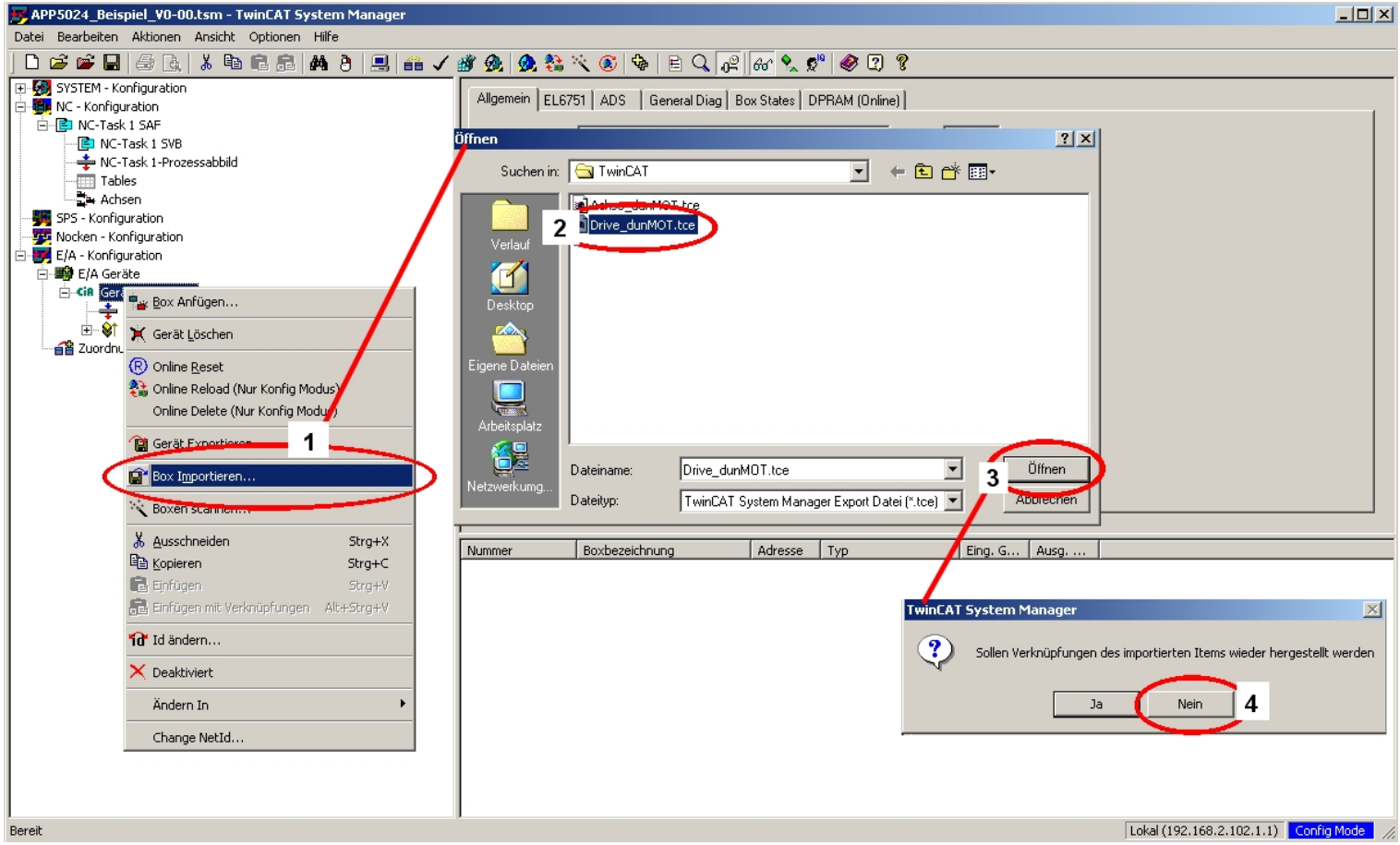
Alle Teilenehmer die an diesen CAN-Bus angeschlossen werden sollen, sind auf die gleiche Baudrate einzustellen.

Dies ist vor der Verkabelung mit den Inbetriebnahmewerkzeugen der CAN-Teilnehmer durchzuführen.

Dabei kann dann auch die CAN-Adresse der Teilnehmer eingestellt werden.



## 4.2 Anfügen von Geräten in der E/A-Konfiguration an den CAN-Bus „CIA“ (Dunkerantrieb BG...CI)




**HINWEIS** Die Verknüpfungen werden später von Hand hergestellt!

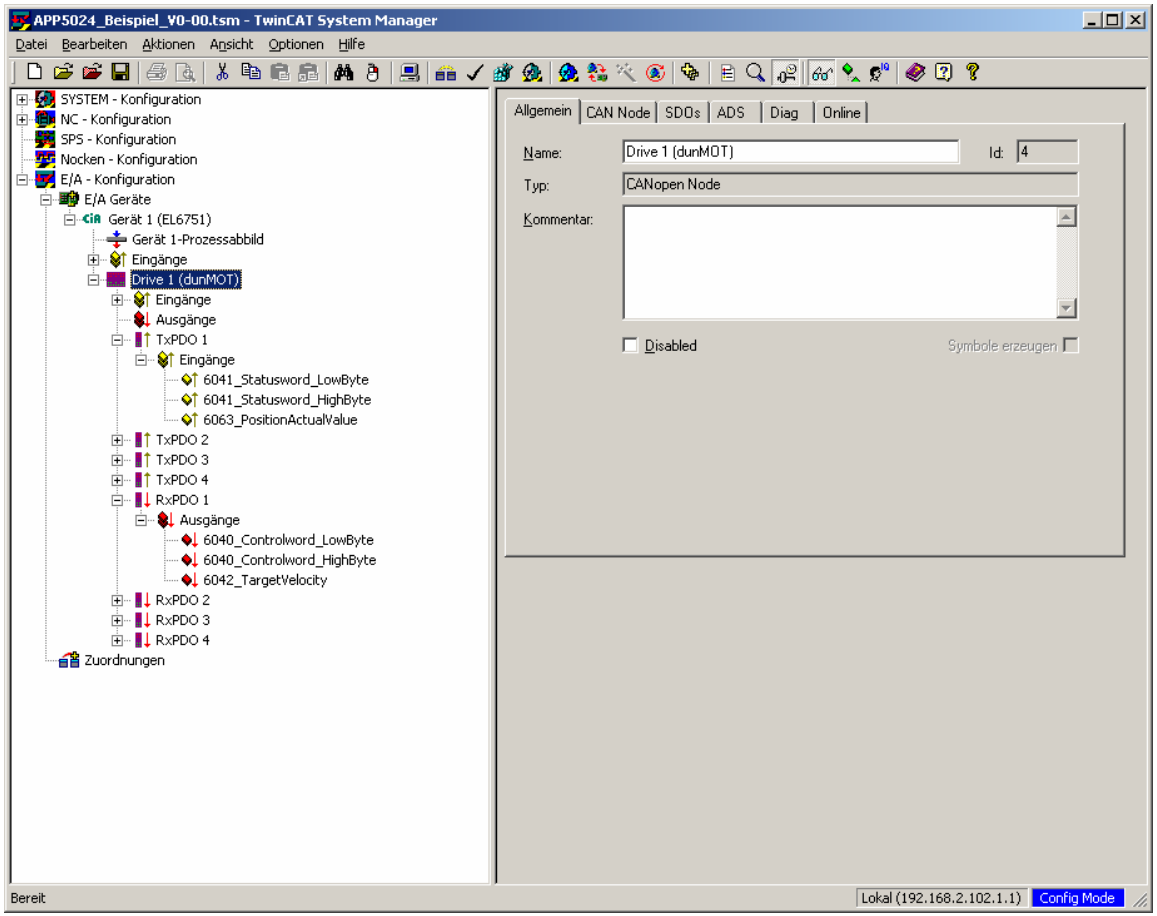
Nach dem Importieren der Datei „Drive_dunMOT.tce“ ergibt sich folgende Darstellung:

Der Antrieb „Drive 1 (dunMOT) hat nur die Variablen:

Eingänge (Antrieb → NC-Achse):
- 6041_Statusword_LowByte
- 6041_Statusword_HighByte
- 6063_PositionActualValue

Ausgänge (NC-Achse → Antrieb):
- 6040_Controleword_LowByte
- 6040_Controlword_HighByte
- 6042_TargetVelocity

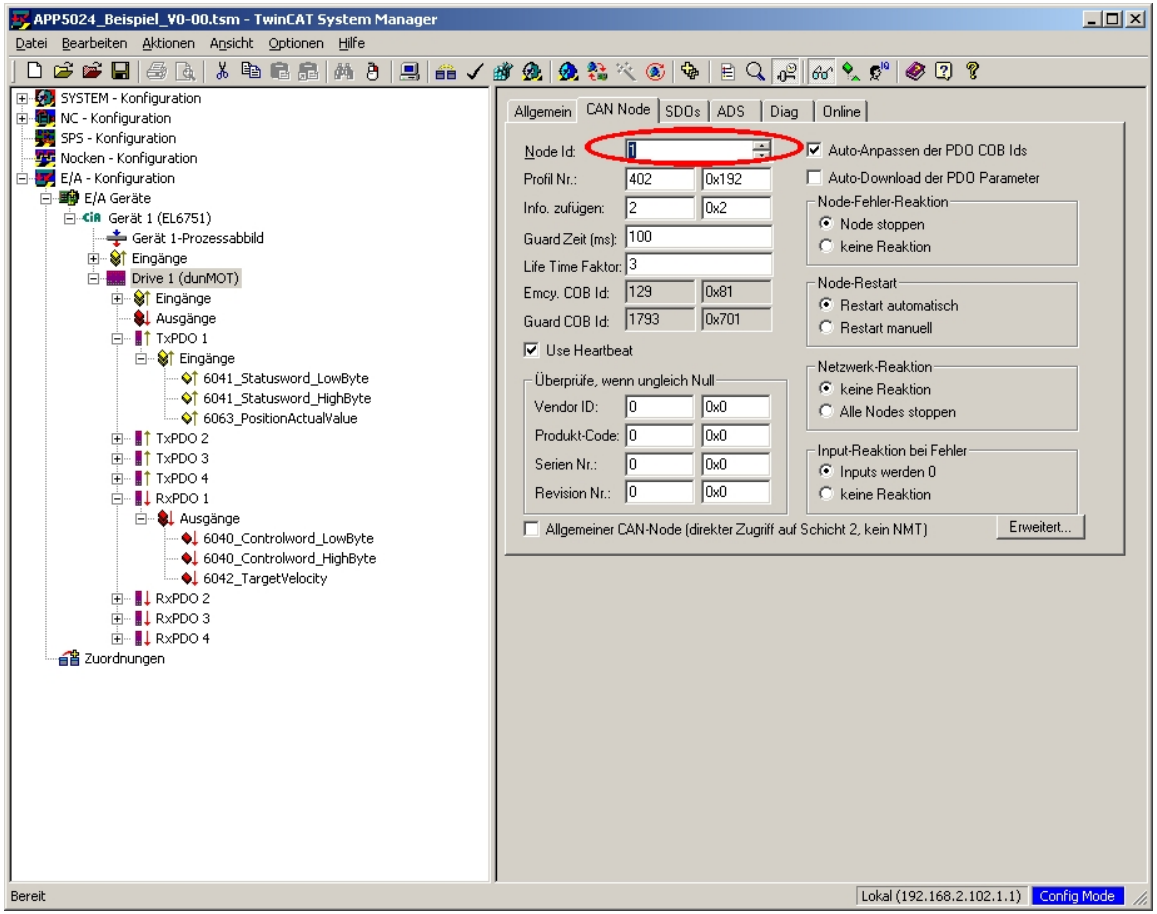
Weitere Variablen sind nicht vorhanden



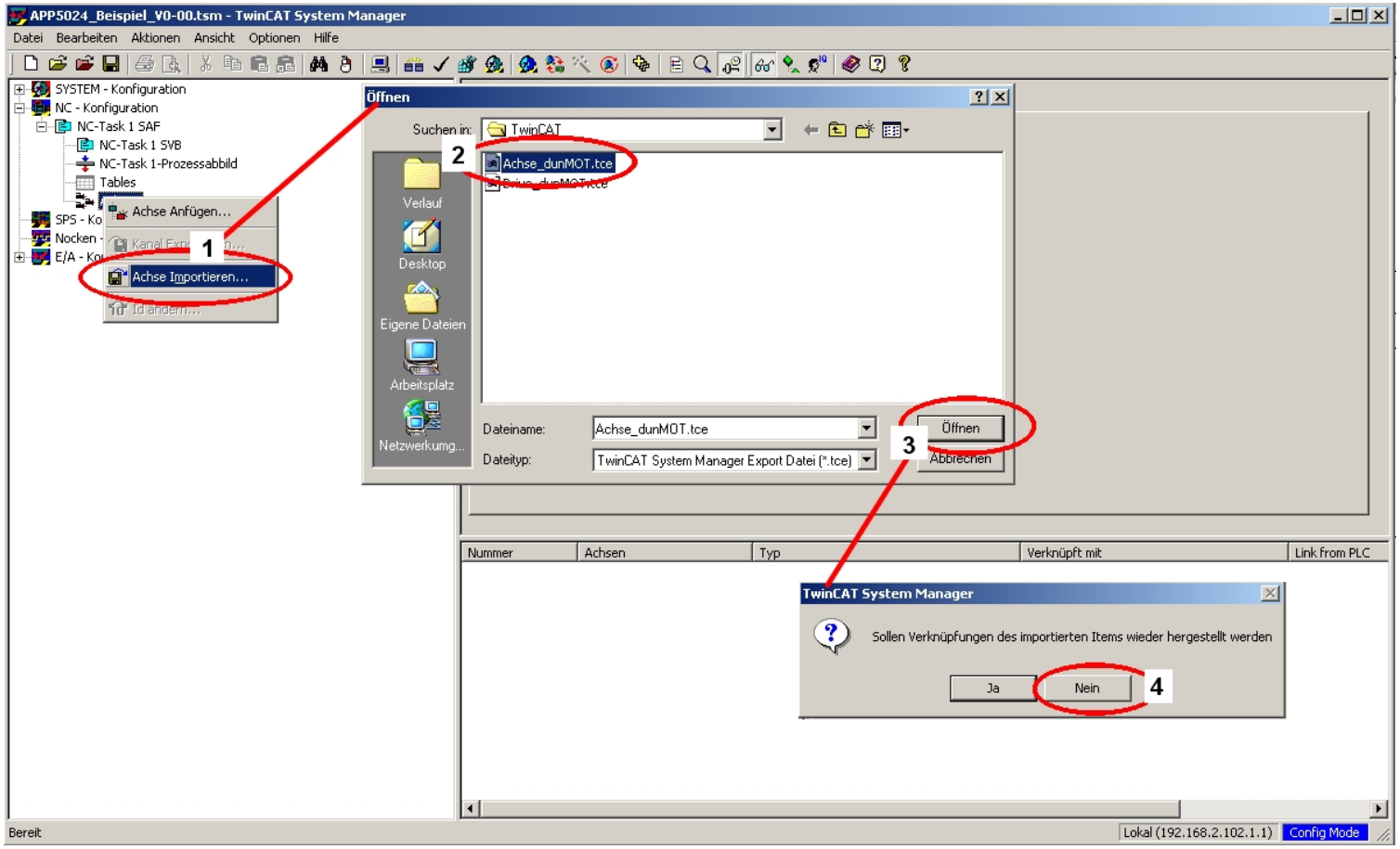
### 4.2.1 Einstellen der Adresse des Antriebes am CAN-Bus

Auf der Registerkarte „CAN Node“ ist für den Antrieb die CAN-Adresse einzustellen.

Alle anderen Einstellungen sollten nicht verändert werden.



## 4.3 Anfügen von Achsen in der NC-Konfiguration




**HINWEIS** Die Verknüpfungen werden später von Hand hergestellt!

Nach dem Importieren der Datei „Achse_dunMOT.tce“ ergibt sich folgende Darstellung:

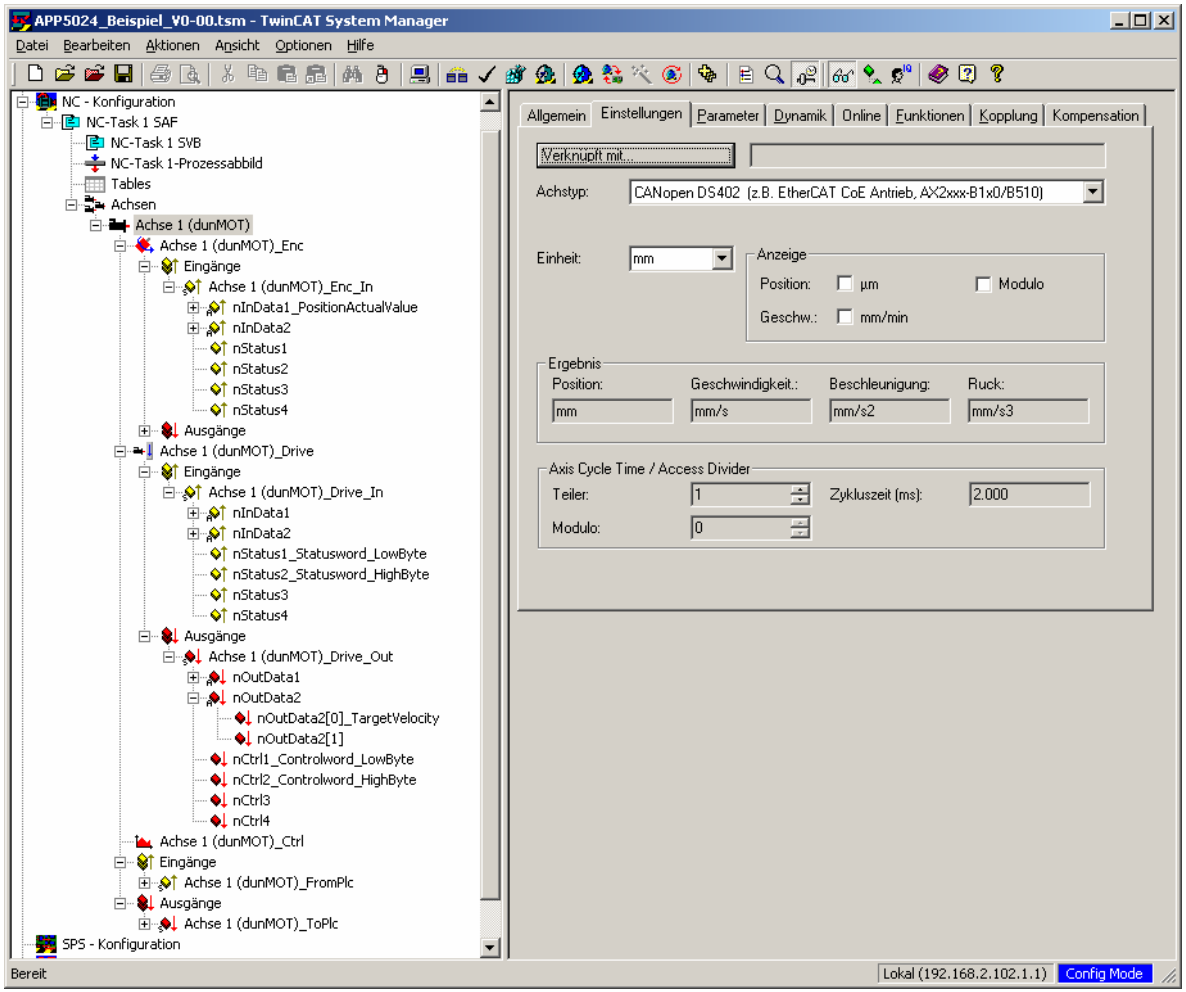
Das Kommunikationsprinzip beruht auf dem Achstyp „CANopen DS402“.

Zur Kommunikation mit dem Antrieb hat die Achse folgende Variablen.

Eingänge (Antrieb → NC-Achse):
- nInData1_PositionActualValue
- nStatus1_Statusword_LowByte
- nStatus2_Statusword_HighByte

Ausgänge (NC-Achse → Antrieb):
- nOutData2[0]_TargetVelocity
- nCtrl1_Controlword_LowByte
- nCtrl2_Controlword_HighByte

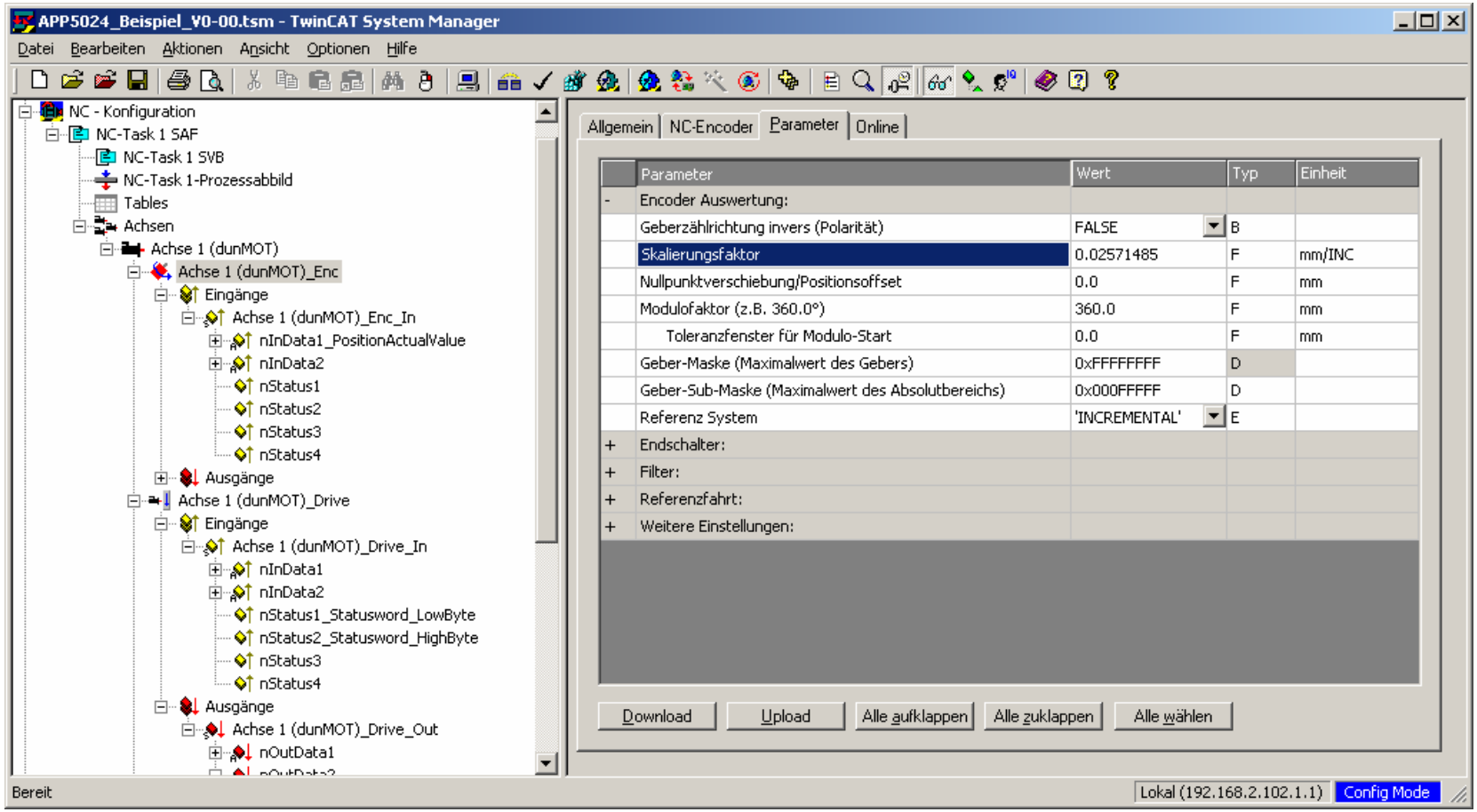
Zur Kommunikation mit der SPS sind die NC-Standard-Strukturen vorhanden



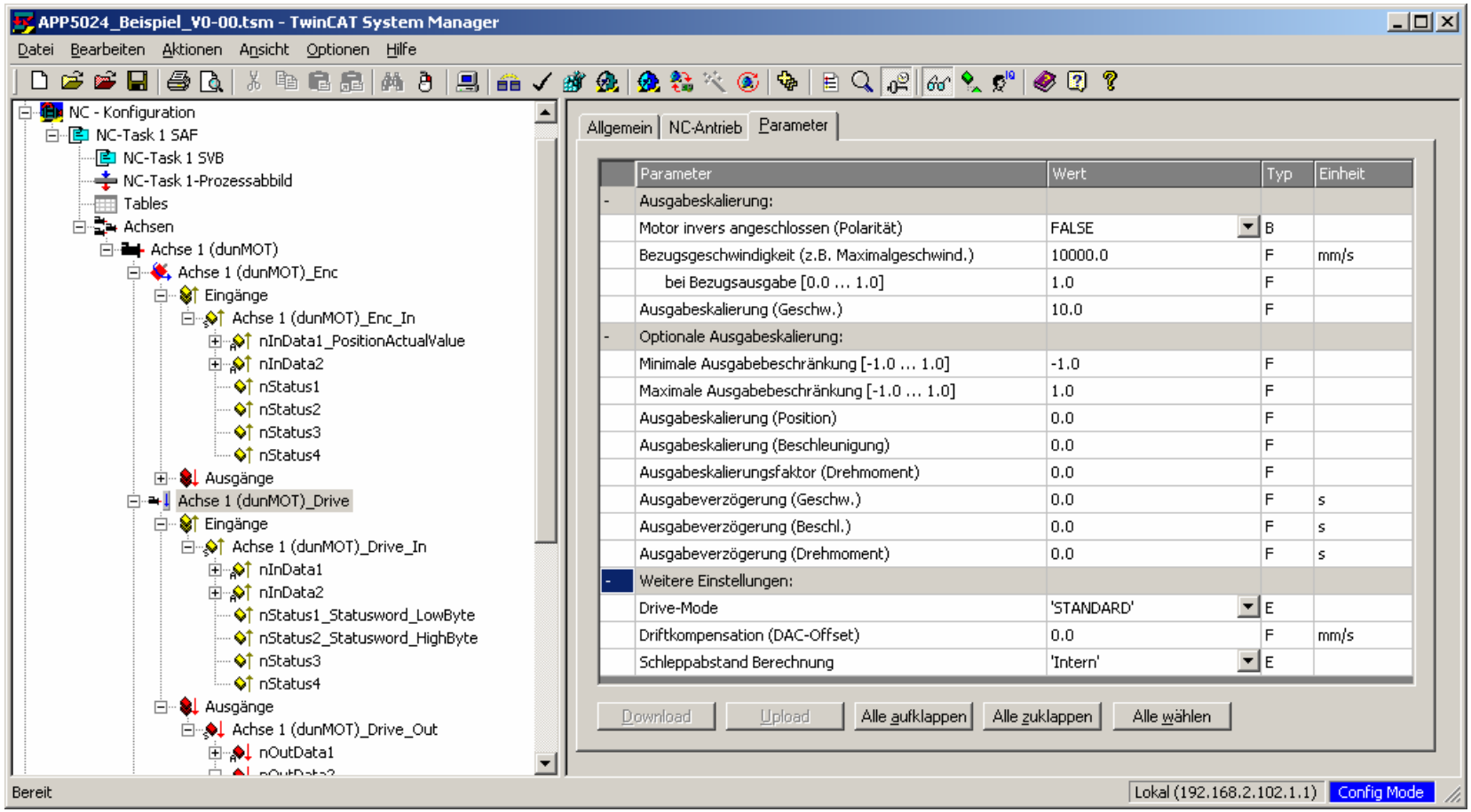
## 4.4 Verknüpfen der NC-Achse mit dem Antrieb

## 4.5 Skalierungen und Faktoren

### 4.5.1 Encoder Skalierungsfaktor



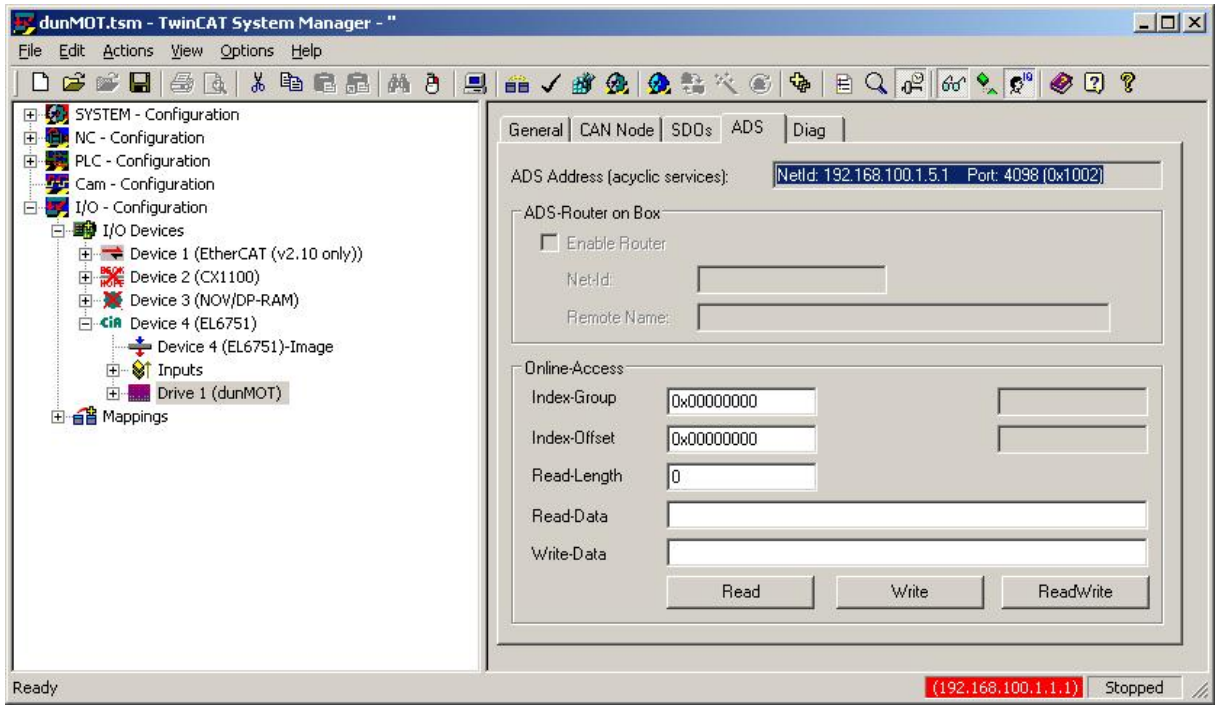
### 4.5.2 Achse Drive Parameter



# 5 Ansteuerung aus TwinCAT SPS

## 5.1 ADS Parameter

Um die Achse aus der TwinCAT SPS ansteuern zu können, braucht man die AmsNetId und den Port.

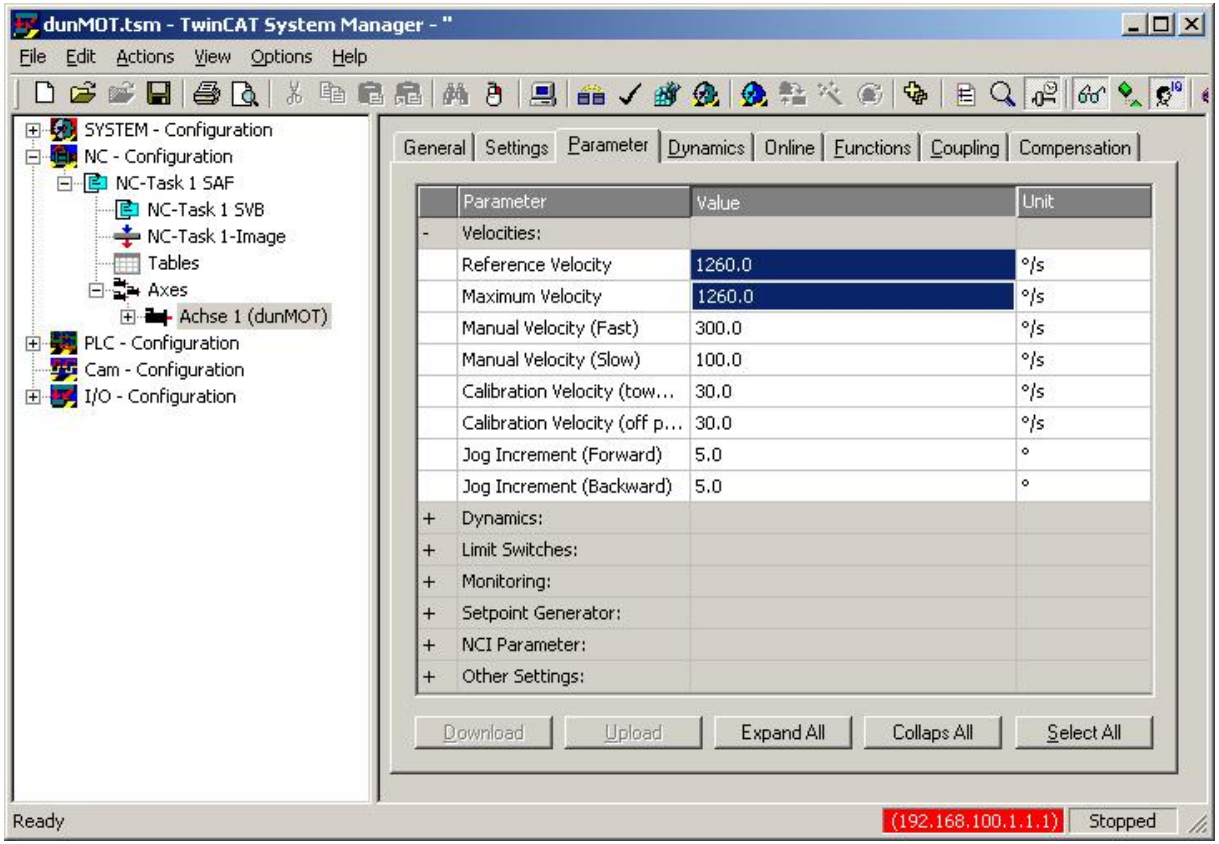


## 5.2 Achsparameter

### 5.2.1 Geschwindigkeiten – Velocity

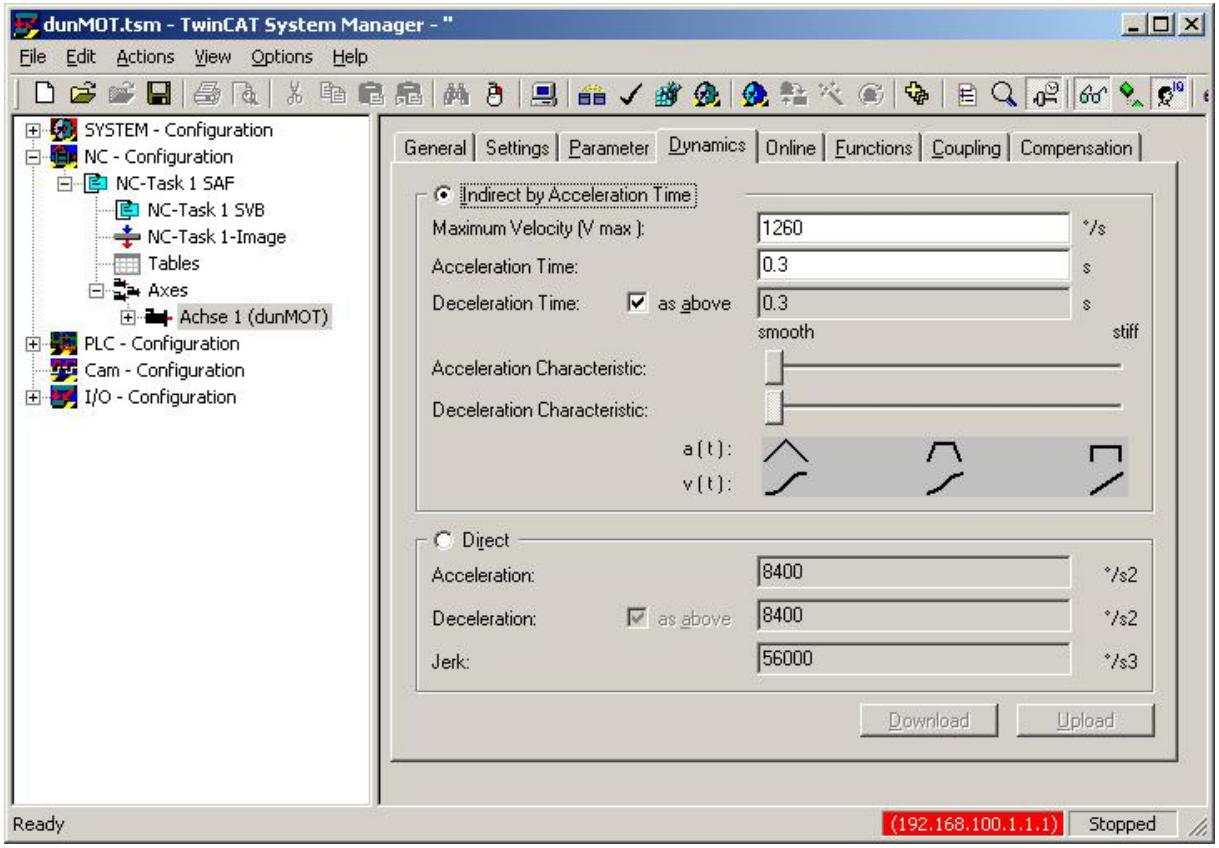
Beim Motortyp BG45x15CI sind folgende Achsparameter einzustellen:

3150 U/min ergeben 1260 °/s



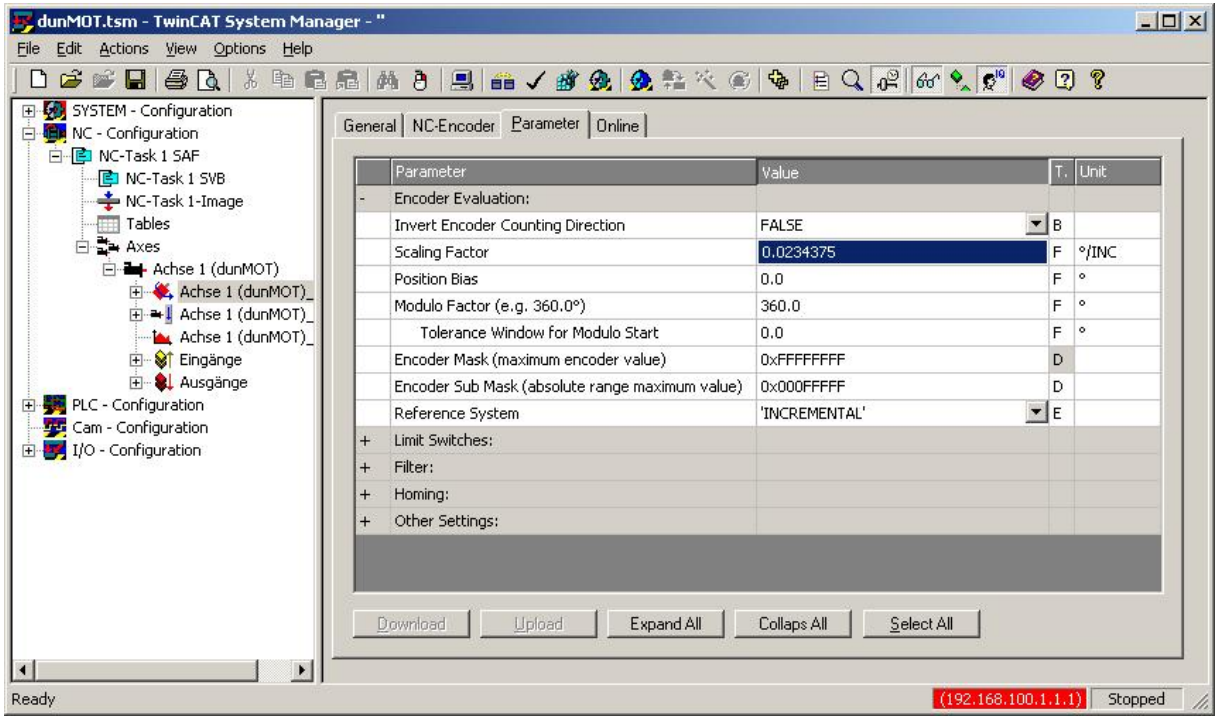
### 5.2.2 Beschleunigungen – Dynamics

Je nach Anwendung muss die Dynamik der Achse angepasst werden!



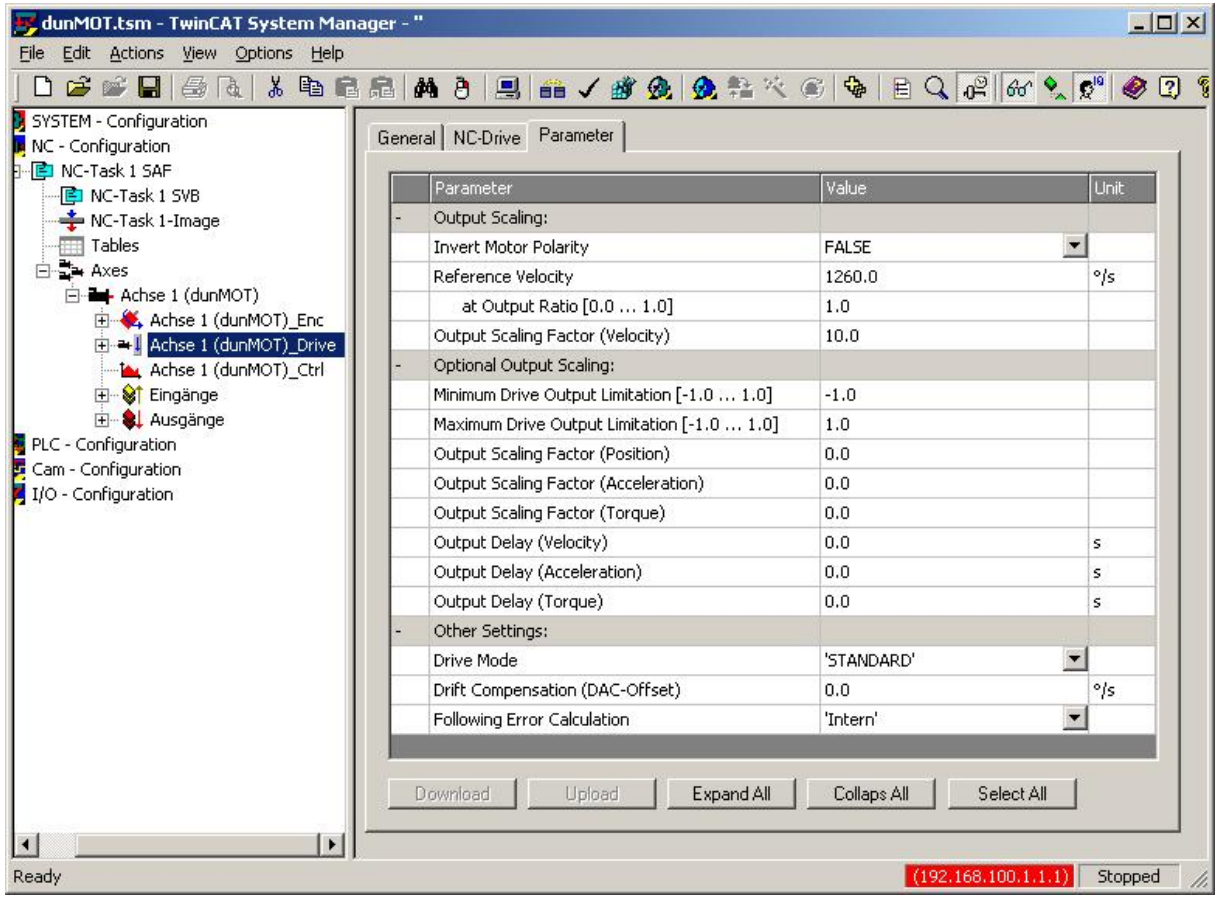
### 5.2.3 Auflösung des Encoders – Encoder Scaling Factor

Bei einer Auflösung von 1024 Inkrementen pro Umdrehung und einem Getriebe von 15:1 berechnet sich der Skalierungsfaktor mit 360° / (1024 * 15)

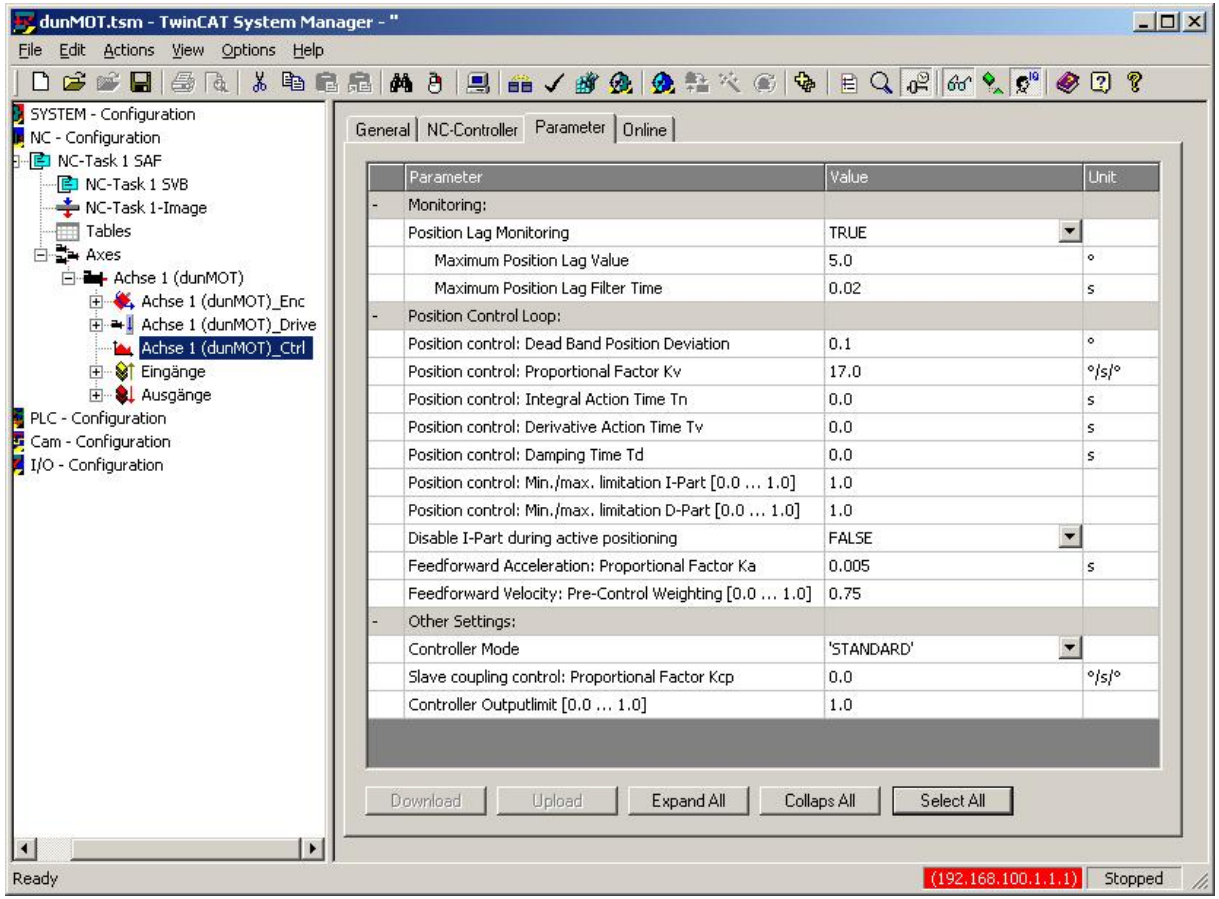


### 5.2.4 Anpassung der Ausgangswerte – Drive Output Scaling

Auch bei den folgenden beiden Einstellungen handelt es sich nur um ungefähre Werte.

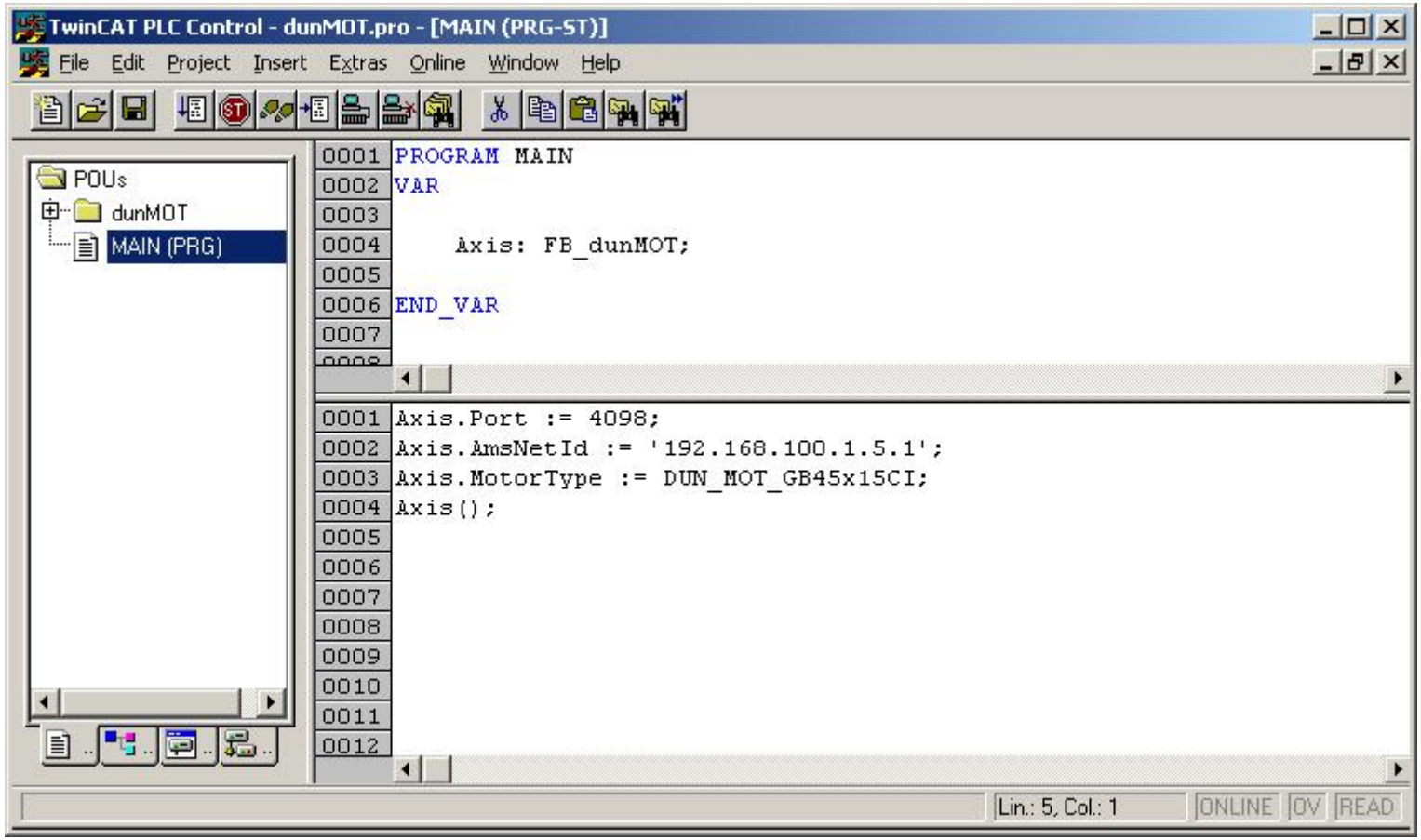


### 5.2.5 Einstellungen für den Achsregler – Position Control Loop



## 5.3 TwinCAT SPS

Um aus der Steuerung die Achse anzusprechen, muss die Bibliothek „IEFWernerDunMOT.lib“ in das Projekt eingebunden werden. Diese verwendet die neue „Motion Control 2“ Bibliothek von Beckhoff. Danach steht der Funktionsblock „FB_dunMOT“ zur Verfügung. Jede Instanz des FBs entspricht einer Achse. Jede Achse muss mit den folgenden absolut notwendigen Parametern versorgt werden (Port, AmsNetId, MotorType):



### 5.3.1 Eingänge FB_dunMOT

Die folgende Tabelle beschreibt alle Eingänge des FB_dunMOT:

| Name | Typ | Bemerkung |
|---|---|---|
| Enable | BOOL | Allgemeine Freigabe der Achse / Achse wird bestromt |
| Enable_Positive | BOOL | Freigabe für eine Fahrt in positiver Richtung |
| Enable_Negative | BOOL | Freigabe für eine Fahrt in negativer Richtung |
| Reset | BOOL | Quittierung eines Fehlers |
| Home | BOOL | Referenziert die Achse **ohne sie zu bewegen** |
| Absolute | BOOL | Startet eine Absolutfahrt mit „Target“ als Zielposition |
| Relative | BOOL | Startet eine Relativfahrt mit „Target“ als Distanz |
| JogForward | BOOL | Tippbetrieb (positive Richtung) |
| JogBackward | BOOL | Tippbetrieb (negative Richtung) |
| StopMovement | BOOL | Hält jegliche Bewegung sofort an |
| Target | LREAL | Zielposition (bei absoluter Positionierung) oder Distanz (bei relativer Positionierung) |
| Velocity | LREAL | Geschwindigkeit für jegliche Art von Bewegung |
| Override | LREAL | Override für jegliche Art von Bewegung |
| Acceleration | LREAL | Beschleunigung für jegliche Art von Bewegung |
| Deceleration | LREAL | Verzögerung für jegliche Art von Bewegung |
| Jerk | LREAL | Ruck für jegliche Art von Bewegung |

**HINWEIS** Falls die Parameter „Velocity”, „Acceleration”, „Deceleration” und „Jerk“ nicht mit einem Wert ungleich 0 initialisiert werden, dann werden die entsprechenden Werte aus der Systemkonfiguration (System Manager) verwendet. Der Standardwert für „Override“ ist 100%.

### 5.3.2 Ausgänge FB_dunMOT

Die folgende Tabelle beschreibt alle Ausgänge des FB_dunMOT:

| Name | Typ | Bemerkung |
|---|---|---|
| Busy | BOOL | True solange eine Aktion läuft (steigende Flanke von Eingängen „Reset“, „Home“, „Absolute“, „Relative“, „JogForward“, „JogBackward“ oder „StopMovement“) |
| Ready | BOOL | Sobald die Freigabe (Eingang „Enable“) gesetzt wird |
| Calibrated | BOOL | Nach der Referenzfahrt (Eingang „Home“) ist die Achse referenziert |
| Started | BOOL | True sobald eine Aktion gestartet wird (steigende Flanke von Eingängen „Home“, „Absolute“, „Relative“, „JogForward“, „JogBackward“ oder „StopMovement“) |
| Position | LREAL | Aktuelle Position der Achse |
| InPosition | BOOL | „In Position“ Flag der Achse |
| HasError | BOOL | Achsfehler vorhanden |
| ErrorId | UDINT | Fehlernummer (siehe Beckhoff Information System) |
| AxisParameterSet | ST_AxisParameterSet | Parameter aus Systemkonfiguration (System Manager) |

# 6 Antriebsparameter

## 6.1 Liste der Antriebsparameter

### 6.1.1 BG45x15CI – 24VDC

```
VAR CONSTANT

    dunMOTAnzahlInitData       : UINT:=13;

    dunMOTInitData     : ARRAY[1..13] OF CANopenSDOSendDataStruct :=
    (Index:=16#3350,   SubIndex:=16#0,    DataLength:=4, Data:=16#96A),      (* Encoder Rückführung für den Drehzahlregler                 *)
    (Index:=16#3550,   SubIndex:=16#0,    DataLength:=4, Data:=16#96A),      (* Encoder Rückführung für den unterlagerten Drehzahlregler   *)
    (Index:=16#3962,   SubIndex:=16#0,    DataLength:=4, Data:=1024),        (* Encoder Auflösung = 1024INC/U (4*256)                     *)

    (***** Regelparameter : Drehzahlregler *****)
    (Index:=16#3310,   SubIndex:=16#0,    DataLength:=4, Data:=100),         (* PID Vel Kp (Proportionalfaktor)                           *)
    (Index:=16#3314,   SubIndex:=16#0,    DataLength:=4, Data:=1000),        (* PID Vel Kvff (Geschwindigkeitsvorsteuerung)               *)

    (***** Unterlagereter Drehzahlregler *****)
    (Index:=16#3510,   SubIndex:=16#0,    DataLength:=4, Data:=100),         (* PI SVel Kp (Proportionalfaktor)                           *)
    (Index:=16#3511,   SubIndex:=16#0,    DataLength:=4, Data:=100),         (* PI SVel Ki (Integralfaktor)                               *)

    (***** Dynamische Strombegrenzung *****)
    (Index:=16#3224,   SubIndex:=16#0,    DataLength:=4, Data:=1),           (* Dynamische Strombegrenzung aktiviert                      *)
    (Index:=16#3224,   SubIndex:=16#1,    DataLength:=4, Data:=4650),        (* I*t- Spitzenstrom  (1,5*Nennstrom) in mA                  *)
    (Index:=16#3224,   SubIndex:=16#2,    DataLength:=4, Data:=3100),        (* I*t- Dauerstrom  (gleich Nennstrom) in mA                 *)
    (Index:=16#3224,   SubIndex:=16#3,    DataLength:=4, Data:=2000),        (* I*t- Zeit in ms                                           *)
    (***** Stromgrenzen *****)
    (Index:=16#3221,   SubIndex:=16#0,    DataLength:=4, Data:=4650),        (* Strombegrenzung - max. positiv in mA  (wie I*t-Spitzenstrom)   *)
    (Index:=16#3223,   SubIndex:=16#0,    DataLength:=4, Data:=4650);        (* Strombegrenzung - max. negativ in mA  (wie I*t-Spitzenstrom)   *)

END_VAR
```

### 6.1.2 BG65x25CI – 24VDC

```
VAR CONSTANT

    dunMOTAnzahlInitData       : UINT:=13;

    dunMOTInitData     : ARRAY[1..13] OF CANopenSDOSendDataStruct :=
    (Index:=16#3350,   SubIndex:=16#0,    DataLength:=4, Data:=16#96A),      (* Encoder Rückführung für den Drehzahlregler                *)
    (Index:=16#3550,   SubIndex:=16#0,    DataLength:=4, Data:=16#96A),      (* Encoder Rückführung für den unterlagerten Drehzahlregler  *)
    (Index:=16#3962,   SubIndex:=16#0,    DataLength:=4, Data:=2000),        (* Encoder Auflösung = 2000INC/U (4*500)                    *)

    (***** Regelparameter : Drehzahlregler *****)
    (Index:=16#3310,   SubIndex:=16#0,    DataLength:=4, Data:=100),         (* PID Vel Kp (Proportionalfaktor)                          *)
    (Index:=16#3314,   SubIndex:=16#0,    DataLength:=4, Data:=1000),        (* PID Vel Kvff (Geschwindigkeitsvorsteuerung)              *)

    (***** Unterlagereter Drehzahlregler *****)
    (Index:=16#3510,   SubIndex:=16#0,    DataLength:=4, Data:=100),         (* PI SVel Kp (Proportionalfaktor)                          *)
    (Index:=16#3511,   SubIndex:=16#0,    DataLength:=4, Data:=100),         (* PI SVel Ki (Integralfaktor)                              *)

    (***** Dynamische Strombegrenzung *****)
    (Index:=16#3224,   SubIndex:=16#0,    DataLength:=4, Data:=1),           (* Dynamische Strombegrenzung aktiviert                     *)
    (Index:=16#3224,   SubIndex:=16#1,    DataLength:=4, Data:=6000),        (* I*t- Spitzenstrom  (1,5*Nennstrom) in mA                 *)
    (Index:=16#3224,   SubIndex:=16#2,    DataLength:=4, Data:=4000),        (* I*t- Dauerstrom  (gleich Nennstrom) in mA                *)
    (Index:=16#3224,   SubIndex:=16#3,    DataLength:=4, Data:=2000),        (* I*t- Zeit in ms                                          *)

    (***** Stromgrenzen *****)
    (Index:=16#3221,   SubIndex:=16#0,    DataLength:=4, Data:=6000),        (* Strombegrenzung - max. positiv in mA  (wie I*t-Spitzenstrom)  *)
    (Index:=16#3223,   SubIndex:=16#0,    DataLength:=4, Data:=6000);        (* Strombegrenzung - max. negativ in mA  (wie I*t-Spitzenstrom)  *)

END_VAR
```

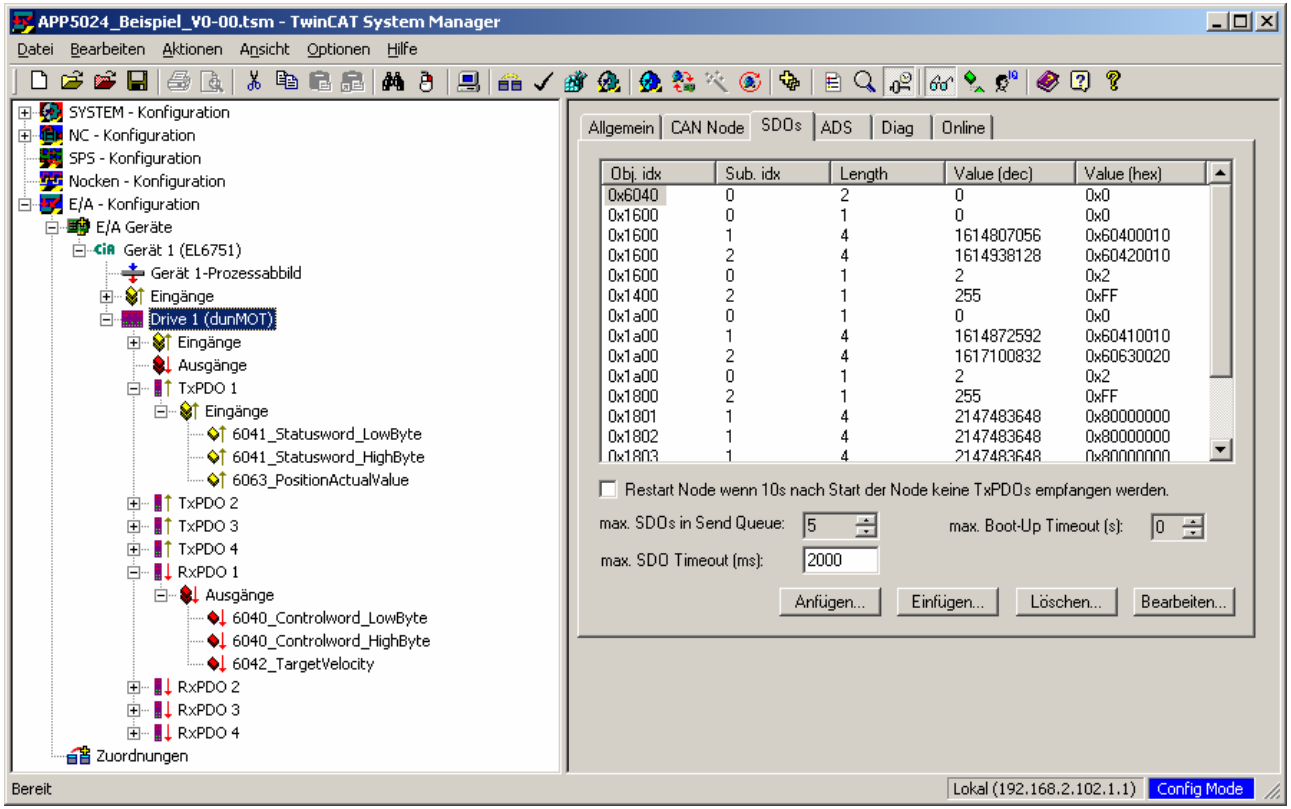
# 7 Anhang

## 7.1 CANopen-Mapping des dunMOT-Antriebes

Auf der Registerkarte „SDOs“ sind die Einträge zur Einstellung des PDO-Mappings und der Betriebsart „Drehzahlregler“.

Diese sind Bestand der Importdatei „Drive_dunMOT.tce“ und sollten nicht verändert werden.

Diese Liste wird vom TwinCAT-System bei einem Start abgearbeitet (übertragen)



| Liste der SDO-Einträge: | | | | |
|---|---|---|---|---|
| **Index** | **Sub-Index** | **Länge** | **Inhalt** | **Bemerkung** |
| 0x6040 | 0 | 2 | 0x0 | Controlword = 0 |
| 0x1600 | 0 | 1 | 0x0 | RxPDO1 : |
| 0x1600 | 1 | 4 | 0x60400010 | RxPDO1 : Controlword |
| 0x1600 | 2 | 4 | 0x60420010 | RxPDO1 : PositionActualValue |
| 0x1600 | 0 | 1 | 0x2 | RxPDO1 : 2 Elemente |
| 0x1400 | 2 | 1 | 0xFF | RxPDO1 : Receive asynchron |
| 0x1A00 | 0 | 1 | 0x0 | TxPDO1 : |
| 0x1A00 | 1 | 4 | 0x60410011 | TxPDO1 : Statusword |
| 0x1A00 | 2 | 4 | 0x60620020 | TxPDO1 : TargetVelocity |
| 0x1A00 | 0 | 1 | 0x2 | TxPDO1 : 2 Elemente |
| 0x1800 | 2 | 1 | 0xFF | TxPDO1 : Transmit asynchron |
| 0x1801 | 1 | 4 | 0x80000000 | TxPDO2 : deaktiviert |
| 0x1802 | 1 | 4 | 0x80000000 | TxPDO3 : deaktiviert |
| 0x1803 | 1 | 4 | 0x80000000 | TxPDO4 : deaktiviert |
| 0x6060 | 0 | 2 | 0x2 | Operation-Mode : Velocity-Mode |
| 0x6048 | 2 | 2 | 0x0 | „vl velocity acceleration“ = 0 → Beschleunigungsrampe deaktiviert, Regler folgt der TargetVelocity |
| 0x6049 | 2 | 2 | 0x0 | “vl velocity deceleration” = 0 → Bremsrampe deaktiviert, Regler folgt der TargetVelocity |