

Funktionsbeschreibung | DE

## TF5200 | TwinCAT 3 CNC

Fehlermanagement





# Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

## Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

## Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

## Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

**EtherCAT** 

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

## Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zuwerhandlungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

# Allgemeine- und Sicherheitshinweise

## Verwendete Symbole und ihre Bedeutung

In der vorliegenden Dokumentation werden die folgenden Symbole mit nebenstehendem Sicherheitshinweis und Text verwendet. Die (Sicherheits-) Hinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

### Symbole im Erklärtext

1. Gibt eine Aktion an.
- ⇒ Gibt eine Handlungsanweisung an.

#### GEFAHR

##### Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

#### VORSICHT

##### Schädigung von Personen und Maschinen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen und Maschinen geschädigt werden!

#### HINWEIS

##### Einschränkung oder Fehler

Dieses Symbol beschreibt Einschränkungen oder warnt vor Fehlern.

### ● Tipps und weitere Hinweise



Dieses Symbol kennzeichnet Informationen, die zum grundsätzlichen Verständnis beitragen oder zusätzliche Hinweise geben.

### Allgemeines Beispiel

Beispiel zu einem erklärten Sachverhalt.

### NC-Programmierbeispiel

Programmierbeispiel (komplettes NC-Programm oder Programmsequenz) der beschriebenen Funktionalität bzw. des entsprechenden NC-Befehls.

### ● Spezifischer Versionshinweis



Optionale, ggf. auch eingeschränkte Funktionalität. Die Verfügbarkeit dieser Funktionalität ist von der Konfiguration und dem Versionsumfang abhängig.

# Inhaltsverzeichnis

<b>Hinweise zur Dokumentation</b> .....	<b>3</b>
<b>Allgemeine- und Sicherheitshinweise</b> .....	<b>4</b>
<b>1 Übersicht</b> .....	<b>8</b>
<b>2 Beschreibung</b> .....	<b>9</b>
2.1 Melden und Aufzeichnen der Fehler .....	11
2.1.1 Unterdrücken von Warnungen .....	13
2.1.2 Ausgabebeispiele .....	13
2.2 Filtern von Fehlermeldungen .....	14
2.2.1 Programmierbeispiel Filtern von Fehlermeldungen .....	17
2.3 Aktivieren von Filterregeln .....	18
2.4 Ausgeben eigener Fehlermeldungen .....	19
2.4.1 Benutzerdefinierte Fehlerausgabe (#ERROR) .....	20
2.4.2 Ausgabe eigendefinierter Fehlermeldungen an HMI .....	23
2.5 Fehlerausgabe TwinCAT3 .....	25
2.5.1 Ausgabe nur über SPS .....	26
2.5.2 Direkte Ausgabe an Eventlogger .....	27
2.5.3 Ausgabebeispiel an HMI .....	29
<b>3 Parameter</b> .....	<b>30</b>
3.1 Übersicht .....	30
3.2 Beschreibung .....	32
3.2.1 Hochlaufparameter .....	32
3.2.2 Kanalparameter .....	38
3.2.3 Achsparameter .....	42
3.3 CNC-Objekte .....	44
3.4 HLI-Parameter .....	47
<b>4 Support und Service</b> .....	<b>49</b>
<b>Stichwortverzeichnis</b> .....	<b>50</b>



# Abbildungsverzeichnis

Abb. 1	Übersicht Lesen und Auswerten der Fehlermeldung innerhalb der CNC .....	9
Abb. 2	Ablauf und Aufbereitung der Fehlermeldungen in der CNC .....	10
Abb. 3	Überblick der 3 Ausgabemöglichkeiten .....	11
Abb. 4	Überblick der Ausgabemöglichkeiten mit Anbindung des SPS-Bausteins .....	12
Abb. 5	Ausgabebeispiel - Protokolldatei .....	13
Abb. 6	Meldung aus dem Kanal .....	14
Abb. 7	Meldung einer Achse .....	15
Abb. 8	Meldung aus dem Hochlauf .....	16
Abb. 9	Aktivieren von Filterregeln .....	18
Abb. 10	Ausgabe eigener Fehlertexte in Protokolldatei .....	19
Abb. 11	Codeauszug – Ausgangslage .....	23
Abb. 12	Codeauszug mit integriertem Fehlertext .....	24
Abb. 13	Architektur- Übersicht TwinCAT-Eventlogger .....	25
Abb. 14	Übersicht mit TwinCAT-Eventlogger .....	25
Abb. 15	FB ChannelError()- Platzierung im System .....	27
Abb. 16	Eventlogger-Output .....	27
Abb. 17	Json-Attribute Output .....	28
Abb. 18	Bildschirmausgabe der Fehlermeldung an der Beckhoff-HMI .....	29
Abb. 19	Bildschirmausgabe nach Doppelklick .....	29

# 1 Übersicht

## Aufgabe

Die Funktionalität des Fehlermanagements umfasst folgende Bereiche:

- Ausgabeort und Umfang der Fehlerausgabe
- Filtern von Fehlermeldungen, sowohl global als auch kanal- oder achsspezifisch
- Zusätzliche Informationen in Fehlermeldungen hinzufügen
- Ausgabe eigener Fehlermeldungen durch den Anwender

---

**i** Diese Funktionalität ist ab der CNC-Version V3.00.xx verfügbar.

---

## Parametrierung

Die Parametrierung der Fehlerausgabe erfolgt in den Hochlaufparametern. Das Filtern und Ergänzen von Fehlermeldungen erfolgt abhängig von der Ebene, auf der der entsprechende Filter wirken soll.

## Programmierung

Eine komplette Übersicht findet sich im Kapitel [Parameter](#) [► 30].

Eigene Fehlermeldungen können vom Anwender über den [#ERROR](#) [► 20]- Befehl im NC-Programm erzwungen werden.

## Verweise auf andere Dokumente

Zwecks Übersichtlichkeit wird eine verkürzte Darstellung der Verweise (Links) auf andere Dokumente bzw. Parameter gewählt, z.B. [PROG] für Programmieranleitung oder P-AXIS-00001 für einen Achsparameter.

Technisch bedingt funktionieren diese Verweise nur in der Online-Hilfe (HTML5, CHM), allerdings nicht in PDF-Dateien, da PDF keine dokumentenübergreifende Verlinkungen unterstützt.



## 2 Beschreibung

Tritt ein Kanal-, Einzelachs- oder Plattformfehler auf, so wird eine entsprechende Meldung abgesetzt. Fehler können von einer CNC-Diagnoseinstanz aufbereitet und mit zusätzlicher Information ergänzt werden (Zugriff auf Datenbank / Dateisystem). Der Anwender kann das Verhalten (Filter) einzelner Fehlermeldungen parametrieren oder über die SPS einstellen. Die erweiterte Fehlermeldung wird über eine Diagnoseinstanz an das übergeordnete Diagnosesystem weitergemeldet.

An die steuerungsinternen weiteren Teilnehmer (z.B. SPS) wird das Auftreten eines Fehlers kompakt und zeitnah mit Fehler-ID, Fehlerklasse und Zeitpunkt gemeldet. Diese Teilnehmer können eine interne Fehlerreaktion unmittelbar ableiten.

Das Lesen und Aufbereiten der CNC-Fehlermeldungen wird standardmäßig durch die CNC durchgeführt.

Das Weitermelden der Fehler über das Kommunikationssystem an eine überlagerte Visualisierung wird spezifisch durch den Anwender eingebracht.

### Verwaltung von Fehlermeldungen

Die Funktionalität des Fehlermanagements ermöglicht dem Anwender die Ausgabe von Fehlermeldungen zu steuern. Die CNC fungiert hierfür als Sammelstelle aller zur Verfügung stehender Informationen:

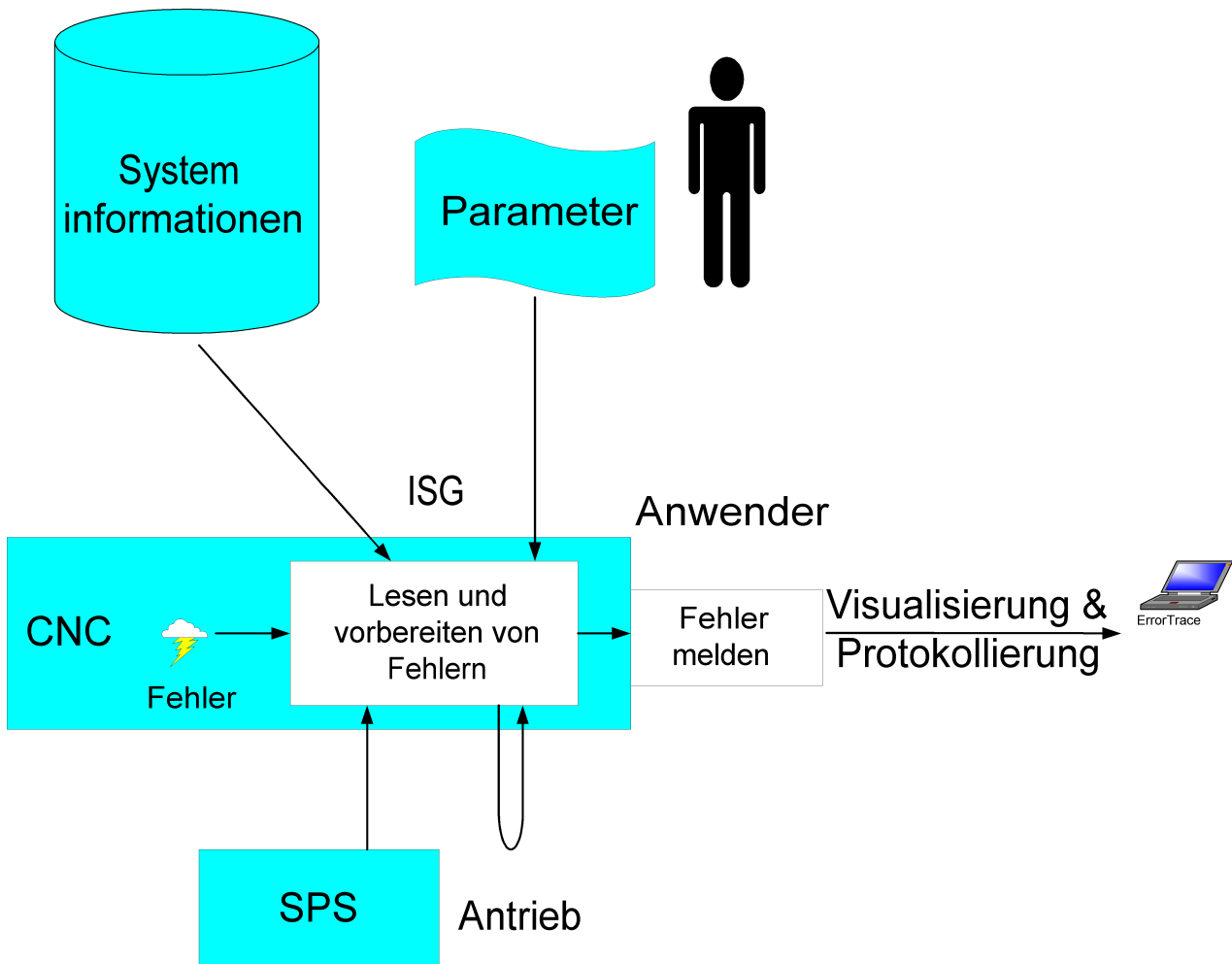


Abb. 1: Übersicht Lesen und Auswerten der Fehlermeldung innerhalb der CNC

**Ablauf**

- Erfassen und Sammeln aller Fehler der CNC
- Filter: Unterdrücken und Filtern der Fehler
- Add. Info: Hinzufügen von interner / externer Fehlerinformation
- Format: Aufbereitung und Formatierung des Fehlers
- Report & Log: Protokollierung der Fehler in Datei (LOG) -> Protokolldatei
- Report & Log: Melden der aufbereiteten Fehler (REPORT) -> Bildschirmausgabe

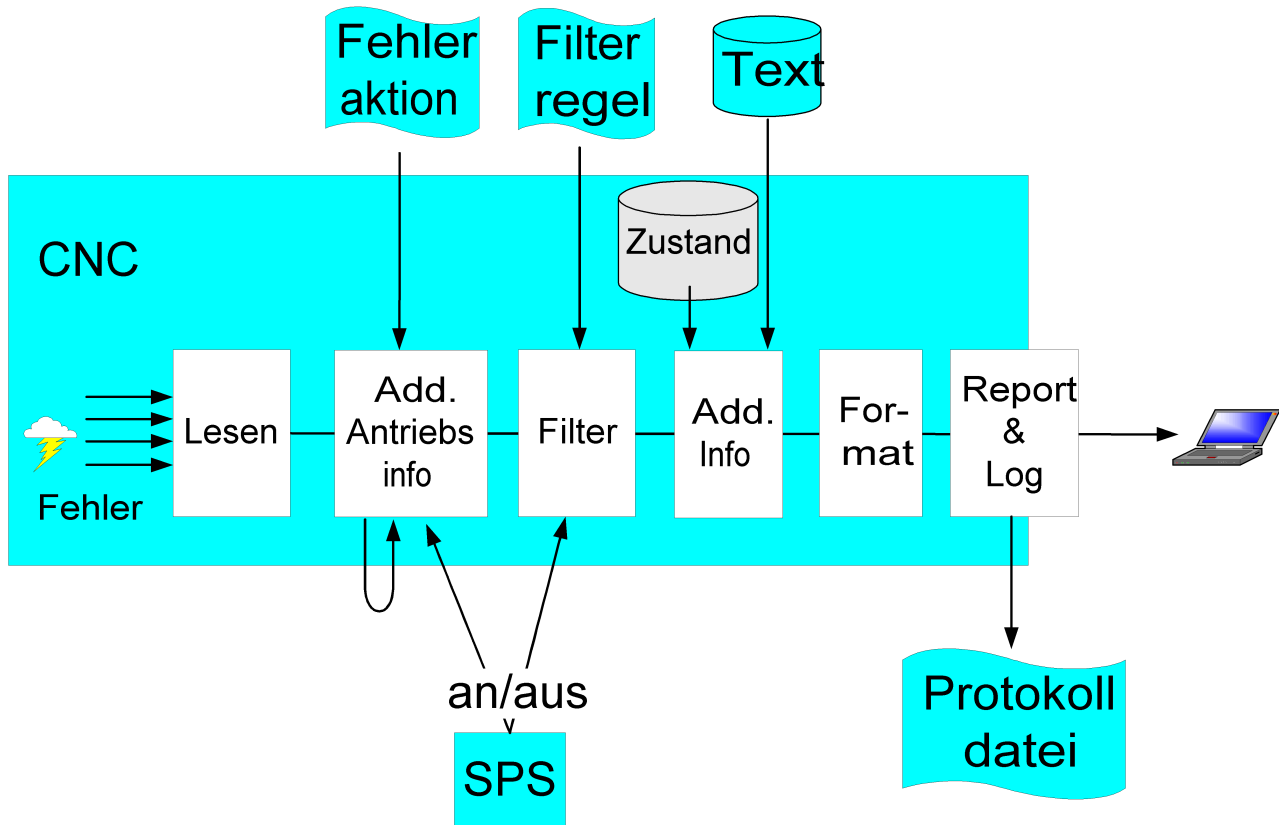


Abb. 2: Ablauf und Aufbereitung der Fehlermeldungen in der CNC

## 2.1 Melden und Aufzeichnen der Fehler

Die Ausgabe der Fehlermeldungen wird zentral über den Parameter `P-STUP-00167` [▶ 32] festgelegt.

- **Alle gewünschten Optionen müssen bei P-STUP-00167 gesetzt werden.**  
Das ausschließliche setzen von `NO_WARNINGS` führt zu keiner Ausgabe!

Der Modus zur Steuerung der Ausgabe kann ebenfalls über CNC-Objekte gelesen und geschrieben werden.

- Lesen über `cnc_error_manager_mode_r` [▶ 45]
- Schreiben über `cnc_error_manager_mode_w` [▶ 45]

### Ausgabemöglichkeiten

Grundsätzlich stehen folgende 3 parallele Ausgabemöglichkeiten zur Verfügung:

1. Print: Ausgabe über Bildschirm/Shell
2. Log: Ausgabe in Protokolldatei
3. Report: Applikationsspezifische Ausgabe

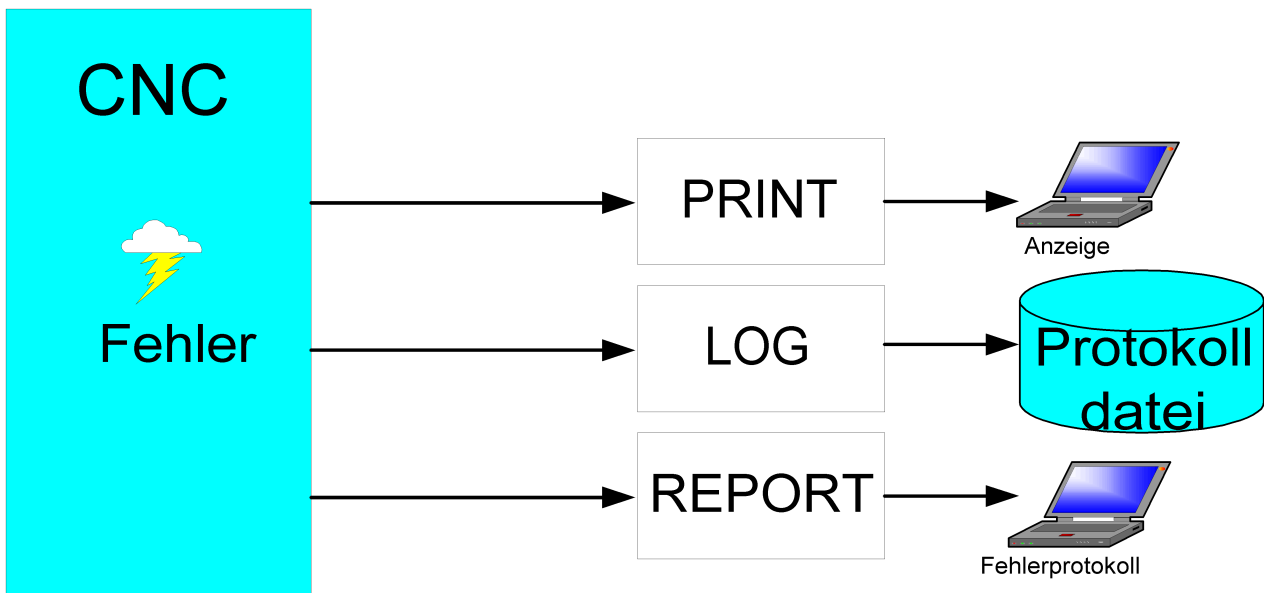


Abb. 3: Überblick der 3 Ausgabemöglichkeiten

### Print

Je nach Applikation kann die Fehlerausgabe direkt auf den Bildschirm bzw. über die Shell erfolgen.

### Protokollierung in Datei

Die Fehler werden in eine Protokolldatei protokolliert. Diese kann über den Parameter `P-STUP-00170` [▶ 33] namentlich festgelegt und über `P-STUP-00171` [▶ 33] die maximale Größe definiert werden. Sobald die Protokolldatei die definierte maximale Größe überschritten hat, wird die Originaldatei in eine Backupdatei kopiert und die Originaldatei gelöscht.

### Applikationsspezifischer Report

Zusätzlich kann eine applikationsspezifische Ausgabe erfolgen.

### „Unmittelbare“ Ausgabe an SPS

Neben der Ausgabe der aufbereiteten Fehlermeldung über PRINT/LOG/REPORT wird die Fehlermeldung "im Rohformat" unmittelbar nach Auftreten an die SPS gesendet. Diese Ausgabe kann über den Protokoll-Mode [P-STUP-00167](#) [► 32] des Fehlermanagers bei Bedarf komplett abgeschaltet werden.

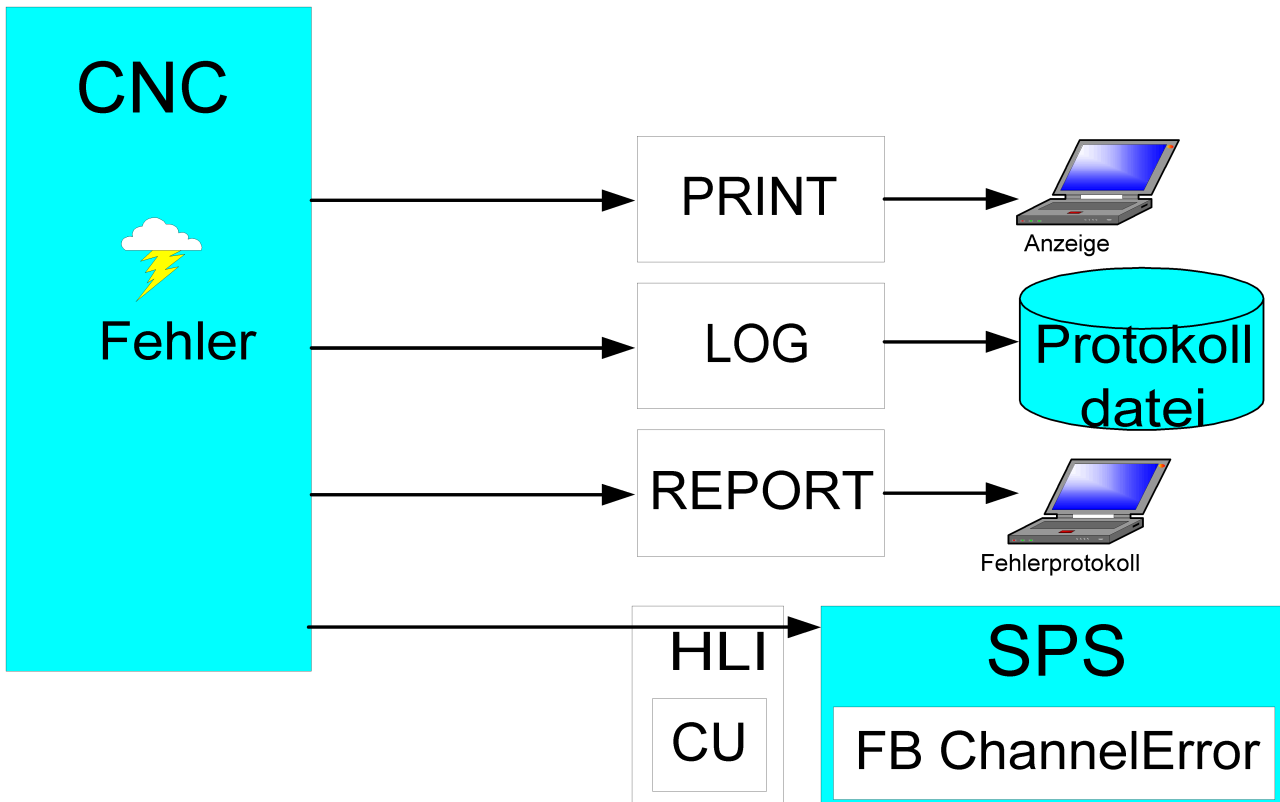


Abb. 4: Überblick der Ausgabemöglichkeiten mit Anbindung des SPS-Bausteins

Falls die SPS keine weiteren Fehlermeldungen der CNC mehr auswerten muss, kann die Ausgabe an die SPS über die ControlUnit (CU) auf dem HLI komplett entfallen.

```
error_protocol_mode PRINT | LOG | REPORT | SEND_NOT_TO_PLC
```

### Ausgaben unter TwinCAT 3

In TwinCAT 3 ist der PRINT-Kanal an die Ausgabe zum Event-Logger angebunden:

## 2.1.1 Unterdrücken von Warnungen

Um den Endanwender nicht mit häufigen Warnungen zu irritieren, kann es von Vorteil sein, diese für den Anwender in der Anzeige zu unterdrücken, aber in der Protokolldatei ([P-STUP-00170](#) [▶ 33]) zu dokumentieren.

Bei einem Reset der Steuerung wird standardmäßig eine Warnung erzeugt, diese kann aber über den Parameter [P-STUP-00166](#) [▶ 32] unterdrückt werden. Hierfür muss die Einstellung des Protokoll-Modus geändert werden:

### Protokoll-Mode: Filtern von Warnungen – TC3\_EventLogger

#### Warnungen nur in Protokolldatei

```
error_protocol_mode LOG | PRINT | REPORT | TC3_EVENT_LOGGER | PRINT_NO_WARNINGS | REPORT_NO_WARNINGS
```

#### Warnungen auch in Protokolldatei unterdrücken

```
error_protocol_mode LOG | PRINT | REPORT | TC3_EVENT_LOGGER | PRINT_NO_WARNINGS | REPORT_NO_WARNINGS | LOG_NO_WARNINGS
```

**Unterdrücken aller Warnungen:** Dies beinhaltet auch das Unterdrücken der Ausgabe von Warnungen zur SPS.

```
error_protocol_mode LOG | PRINT | REPORT | TC3_EVENT_LOGGER | NO_WARNINGS
```

Diese ist identisch zu:

```
error_protocol_mode LOG | PRINT | REPORT | TC3_EVENT_LOGGER | PRINT_NO_WARNINGS | REPORT_NO_WARNINGS | LOG_NO_WARNINGS | SEND_TO_PLC_NO_WARNINGS
```

## 2.1.2 Ausgabebeispiele

### Protokollierung in Datei

Die Ausgabe einer Fehlermeldung in der Protokolldatei sieht z.B. wie folgt aus:

```
<<-----
21.09.2018 16:03:13:571 (5616771)                               Version: V3.01.3061.3204
-----
Error   : 20232 - Unexpected 'ENDFOR'; it does not match the actual control block.
-----
Program : ..\prg\EventLogTest.nc
Path    : ..\prg\ (No: 65535)
File    : EventLogTest.nc
Block no: N10                                           Lineoffset Start/End: 5/13
Line    : N010 $ENDFOR
Position:          ^^^^^^^^^
-----
Channel : (No.: 1)
Class   : SYNTAX (2)                               Reaction : PROGRAM_ABORT (2)
=====
Config  : ZWEI_KANAL_KONFIGURIERUNG
Modul   : EW_FKT.C                               Line : 4017
BF-Type : DECODER (9)                             Commu: DEC_1 (42)           Multiple-ID: 0
Content : NC_PROGRAM (1)
----->>
```

Abb. 5: Ausgabebeispiel - Protokolldatei

## 2.2 Filtern von Fehlermeldungen

Die Ausgabe von Fehlermeldungen kann global, kanal- oder achsspezifisch gefiltert werden.

### Filterebene erkennen

Auf welcher Ebene die Fehlermeldung oder Warnung gefiltert werden muss, ist über die Art der Meldung in der Protokolldatei ersichtlich.

Kanalfehlermeldungen oder -warnungen sind wie folgt zu erkennen:

```

<<-----
05.07.2021 08:14:56:273 (456)                               Version: V3.01.3077.6404
-----
Error   : 20418 - Kanalparameter: Spindelanzahl und Anzahl konf. Spindelachsen sti
-----
Channel : (No.: 1)
value   : 17
Class   : WARNING (0)                                     Reaction : NO_REACTION (1)
=====
Value 1 : Actual value           : 17
Value 2 : Expected value        : 0
Value 3 : Corrected value       : 0
-----
Config  : EIN_KANAL_KONFIGURIERUNG
Modul   : DEC_ABL5.C                               Line : 6002
BF-Type : DECODER (9)                               Commu: DEC_1 (42)           Multiple-ID: 0
Content : MACHINE-DATA_SET (2)
----->>

```

Abb. 6: Meldung aus dem Kanal

Achsfehlermeldungen oder -warnungen sind wie folgt zu erkennen:

```
<<-----
05.07.2021 08:22:36:805 (397)                               Version: V3.01.3077.6404
-----
Error   : 110112 - Geschwindigkeitsoverride ueberschreitet Grenzwert.
-----
Value   : 1.235e+004
Class   : WARNING (0)
Axis    : Achse_1 (Axis-no: 1)
Max     : 2000
Reaction : NO_REACTION (1)
=====
Value 1 : Logical axis-number : 1
Value 2 : Actual value       : 1.235e+004
Value 3 : Limiting value     : 2000
Value 4 : Corrected value    : 100
-----
Utility : Error 2045 - Fehlerhafter Index bei Arrayparameter.
Modul   : ISG_UTIL.C          Line : 3066
-----
Config  : EIN_KANAL_KONFIGURIERUNG
Modul   : MDS_UTIL.C          Line : 21987
BF-Type : AX_VERWALT (6)      Commu: AXV (2)           Multiple-ID: 0
Content : MACHINE-DATA_SET (2)
----->>
```

Abb. 7: Meldung einer Achse

Eine Meldung aus dem Hochlauf hat weder einen Achs- noch einen Kanaleintrag:

```
<<-----
05.07.2021 08:20:36:015 (0)                               Version: V3.01.3077.6404
-----
Error   : 1001117 - Unbekanntes Listenelement in Liste enthalten.
-----
File    : hochtwin.lis
                Fileoffset: 144
Line    : b_storage_size[0]                               0x200000
-----
Class   : WARNING (0)                                   Reaction : NO_REACTION (1)
=====
Utility : Error 1001117 - Unbekanntes Listenelement in Liste enthalten.
Modul   : INTPR.C                                       Line : 2096
-----
Config  : EIN_KANAL_KONFIGURIERUNG
Modul   : HOCH_UTI.C                                   Line : 582
BF-Type : SYS_ABL5 (18)                               Commu: (0)           Multiple-ID: 0
Content : INTERPRETE_FILE (6)
----->>
```

Abb. 8: Meldung aus dem Hochlauf



## 2.2.1 Programmierbeispiel Filtern von Fehlermeldungen

Die Warnung **ID 20048** soll im ersten Schritt exemplarisch gefiltert und im zweiten Schritt um einen Zusatztext erweitert werden.

Es handelt sich um eine Warnung im Kanal; folgendes ist zu parametrieren:

### Filtern einer einzelnen Fehlermeldung

```
error_filter[0].reason 20048 (P-CHAN-00378)
(HIDE - keine Ausgabe der Meldung)
error_filter[0].conditional_action HIDE (P-CHAN-00381)
```

Mit diesen Einstellungen wird die Ausgabe der Warnung ID 20048 unterbunden.

Um diesen Filter für mehrere Meldungen wie beispielsweise den IDs 20048, 20622 und 21691 zu verwenden, müsste die Zeile wie folgt erweitert werden:

```
error_filter[0].reason 20048, 20622 , 21691
```

Es können maximal 5 Fehler-IDs in einem Filter angegeben werden.

### Bestimmte Warnung zu Fehler verändern

```
error_filter[0].reason 20048 (P-CHAN-00378)
error_filter[0].action NONE (P-CHAN-00379 keine Aktion)

(Aktion: Ausgabe als Syntaxfehler)
error_filter[0].conditional_action F_SYNTAX (P-CHAN-00381)
```

Mit diesen Einstellungen wird aus der Warnung ID 20048 in der Ausgabe ein Fehler der Behebungsklasse 2.

## 2.3 Aktivieren von Filterregeln

Filterregeln können von der Oberfläche oder der SPS dynamisch aktiviert und deaktiviert werden. Dies ist auf Plattformebene, kanal- und achsspezifisch möglich.

Voraussetzung dafür ist, dass bei der Definition der Filterregel ein entsprechendes Bit im Datum „activation\_bit“ gesetzt wird.

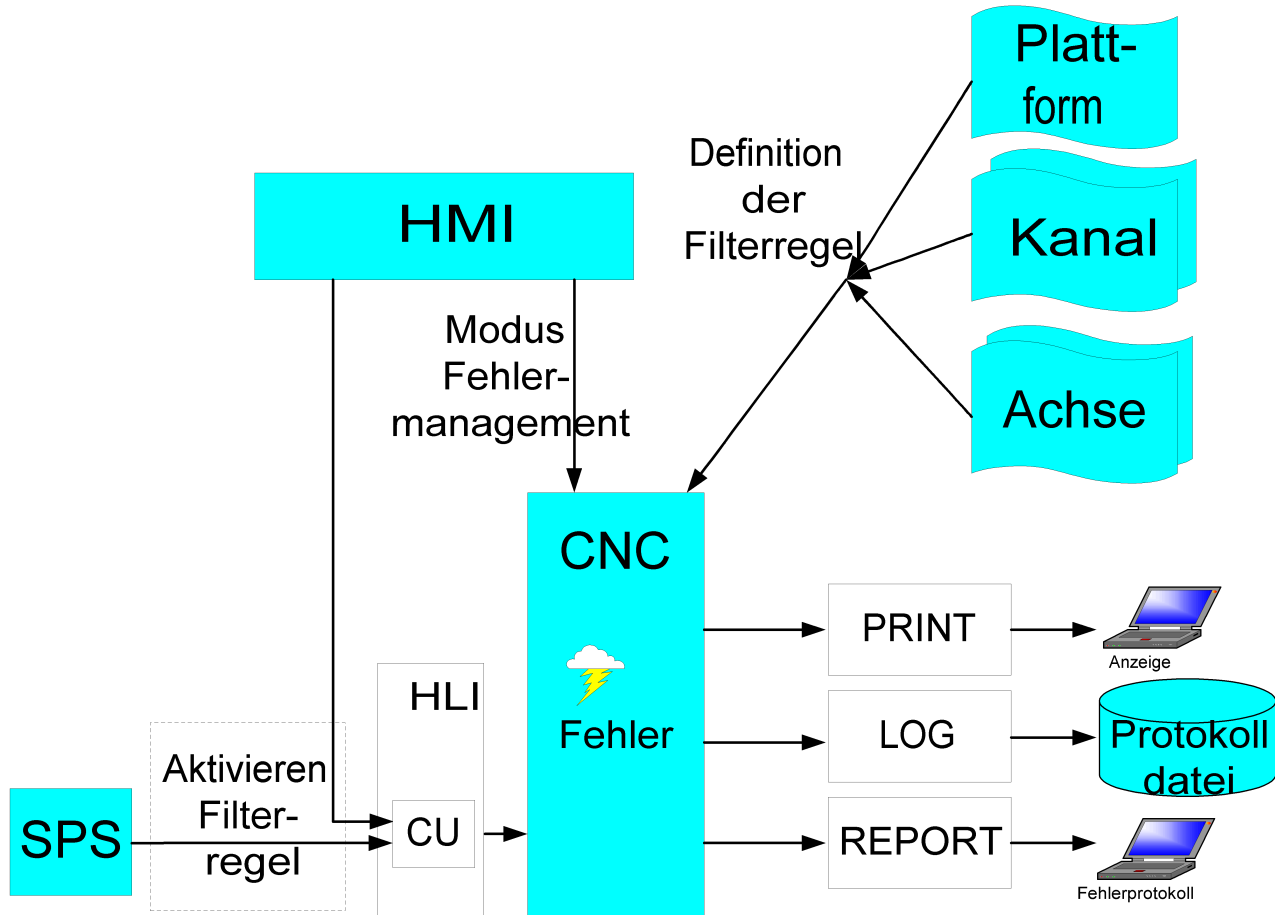


Abb. 9: Aktivieren von Filterregeln

### Aktivierung über die Oberfläche

Voraussetzung dafür ist das Zusammenspiel von Parameter und CNC-Objekt.

Für die Nutzung sind folgende Paare zu betrachten:

- Plattform: P-STUP-00188 und cnc\_error\_filter\_w [▶ 44]
- Kanalspezifisch: P-CHAN-00380 und mc\_error\_error\_filter\_w [▶ 45]
- Achsspezifisch: P-AXIS-00629 und ac\_<i>\_error\_filter\_w [▶ 45]

### Aktivierung über SPS

Voraussetzung dafür ist das Zusammenspiel von Parameter und entsprechender ControlUnit (CU).

Für die Nutzung sind folgende Paare zu betrachten:

- Plattform: : P-STUP-00188 und der CU error filter [▶ 47]
- Kanalspezifisch: P-CHAN-00380 und CU error filter [▶ 47]
- Achsspezifisch: P-AXIS-00629 und CU error filter [▶ 47]

## 2.4 Ausgeben eigener Fehlermeldungen

Der Anwender hat die Möglichkeit, im NC-Programm über den NC-Befehl `#ERROR` [► 20] eigene Fehlermeldungen auszugeben.

Die zugehörigen Fehlertexte werden in einer Textdatei hinterlegt, die über `P-STUP-00169` [► 33] parametrisiert wird. Die Ausgabe erfolgt in die Protokolldatei, die über `P-STUP-00170` [► 33] definiert wird.



**Die in der Textdatei hinterlegten Fehler-IDs mit den zugehörigen Fehlertexten werden ausschließlich in die Protokolldatei geschrieben.**

**Der Bereich für mögliche Fehler-IDs ist 1 - 1000.**

### Kundenspezifischer Fehlertext in Protokolldatei

Inhalt der kundenspezifischen Fehlertextdatei:

```
455 Fehlertext zu ID455
```

Der Aufruf im NC-Programm ist dann wie folgt:

```
#ERROR [ID455 RC2 PV1=5 PV2=4.999 PM1=2 PM2=3]
```

Weitere Informationen zum `#ERROR`-Befehl : [\[PROG://Benutzdefinierte Fehlerausgabe \[► 20\]\]](#)

Die Ausgabe in der Protokolldatei:

```
<<-----
09.07.2021 06:33:48:225 (73934)                               Version: V3.01.3077.6404
-----
Error   : 455 - Fehlertext zu ID 455
-----
Program : -
Block no: N0                                               Lineoffset Start/End: 0/46
Position: ^
-----
Channel : (No.: 1)
Value   : 5
Class   : SYNTAX (2)                                     Reaction : PROGRAM_ABORT (2)
=====
Value 1 : Actual value           : 5
Value 2 : Error value            : 4.999
-----
Config  : EIN_KANAL_KONFIGURIERUNG
Modul   : STR_ERR.C                                       Line : 766
BF-Type : DECODER (9)                                     Commu: DEC_1 (42)      Multiple-ID: 0
Content : NC_PROGRAM (1)
----->>
```

Abb. 10: Ausgabe eigener Fehlertexte in Protokolldatei

Eigendefinierte Fehlermeldungen unter TwinCAT3 siehe: [Ausgabe eigendefinierter Fehlermeldungen an HMI](#) [► 23]

### Sehen Sie dazu auch

- 📖 [Ausgabe eigendefinierter Fehlermeldungen an HMI \[► 23\]](#)

## 2.4.1 Benutzerdefinierte Fehlerausgabe (#ERROR)

Der NC-Befehl #ERROR erlaubt die Ausgabe benutzerdefinierter Fehlermeldungen, die von der übergeordneten Bedienung (GUI = Graphical User Interface) weiterverarbeitet werden. Zusätzliche Parameter bieten die Möglichkeit zur genaueren Spezifizierung des Fehlers.

<b>#ERROR</b> [ [ID<expr>] [RC<expr>] [MID<expr>] {PV<I><expr>} {PM<I><expr>} {PIV<I><expr>} ]	(nicht modal)
---	---------------

ID<expr>                      Fehlernummer:

Gültige Werte	Bedeutung
1...1000	Der Zahlenwert bestimmt die auszugebende kundenspezifische Fehlernummer.

RC<expr>                      Fehlerklasse:

Gültige Werte	Bedeutung
0	Warnung: Kein Übergang in Fehlerzustand. Fortsetzung der Programmbearbeitung.
2	Fehler: Übergang in Fehlerzustand. Per NC-RESET behebbar.
7	Fataler Fehler: Übergang in Zustand 'Systemfehler'. Neustart der Steuerung erforderlich.

MID<expr>                      Mehrfach-ID. Zähler dient als Unterscheidungsmerkmal, wenn in einem NC-Programm mehrmals der #ERROR-Befehl mit der gleichen Fehlernummer (ID) verwendet wird. MID muss eine positive Ganzzahl sein.

PV<i><expr>                      Es können maximal 5 (1 ≤ i ≤ 5) kundenspezifische Zahlenwerte (PV1...PV5) im Realformat in der Fehlermeldung mit ausgegeben werden.

PM<i><expr>                      Die maximal 5 (1 ≤ i ≤ 5) PM-Parameter (PM1...PM5) dienen dazu, die Bedeutung der PV-Parameter genauer zu spezifizieren.

Gültige Werte	Bedeutung
0	IGNORE, Wert ohne Bedeutung
1	Grenzwert
2	Aktueller Wert
3	Fehlerhafter Wert
4	Erwarteter Wert
5	Korrigierter Wert
6	Logische Achsnummer
7	Antriebstyp
8	Logische Bedienelementnummer
9	Zustand
10	Transition
11	Sender
12	Klasse
13	Instanz
14	Identifikationsnummer
15	Status
16	Ringnummer
17	Satznummer
18	Unterer Grenzwert

<b>19</b>	Oberer Grenzwert
<b>20</b>	Startwert
<b>21</b>	Endwert

PIV<*i*><*expr*> Die maximal 4 (1 <= i <= 4) PIV-Parameter (PIV1...PIV4) dienen zur Übermittlung zusätzlicher Informationen im Realformat.

**Für nicht programmierte Parameter sind folgende Standardwerte gültig:**

ID	1
RC	0
MID	0
PV1...PV5	0.0
PM1...PM5	1
PIV1...PIV4	0.0

**Benutzerdefinierte Fehlerausgabe**

```
:  
#ERROR [ID100 RC0 MID10] ;Warning 100, Mehrfachkennung 10  
:  
#ERROR [ID455 RC2 PV1=5 PV2=4.999 PM1=2 PM2=3] ;Fehler 455 mit  
;Parametern  
:  
#ERROR [ID455 RC2 MID2 PV1 5 PV2 4.999 PM1 2 PM2 3] ;Fehler 455  
;Mehrfachkennung 2  
;mit Parametern  
:  
#ERROR [ID999 RC7] ;Fataler Fehler 999  
:
```

## 2.4.2 Ausgabe eigendefinierter Fehlermeldungen an HMI

Die Ausgabe von eigendefinierten Fehlermeldungen über den #ERROR [► 20] Befehl kann an der Oberfläche des TwinCAT3-Systems angezeigt werden. Der Bereich für die möglichen Fehler-IDs ist 1 - 1000.

Die Fehlertexte können mehrsprachig angelegt werden.



**Eine Verknüpfung der angegebenen Fehlertexte aus der Datei P-STUP-00169 mit der Ausgabe an die HMI findet nicht statt.**

Die erforderlichen Fehlertexte müssen in beiden Dateien integriert werden.

Die Fehlertexte für die Ausgabe an HMI müssen in der Datei

`C:\TwinCAT3.1\Target\Resource\TcCncUserEvents.xml`

integriert werden.

### Integration einer Meldung in TcCncUserEvents.xml

```
<Source>
<GUID>{3FD56AAE-0711-4359-89A5-3E1ECCCC634E}</GUID>
  <Id>650</Id>
  <Name>TwinCAT CNC User Errors</Name>
  <Event>
    <!-- User specific error messages: ID range 1 - 1000 -->
    <Id>1</Id>
      <Message LcId="1033">Error message ID1 (#ERROR[ID1])</Message>
      <Message LcId="1031">Fehlermeldung ID1 (#ERROR[ID1])</Message>
    </Event>
  <Event>
    <Id>2</Id>
      <Message LcId="1033">Error message ID2 (#ERROR[ID2])</Message>
      <Message LcId="1031">Fehlermeldung ID2 (#ERROR[ID2])</Message>
    </Event>
  <Event>
    <Id>1000</Id>
      <Message LcId="1033">Error message ID1000 (#ERROR[ID1000])</Message>
      <Message LcId="1031">Fehlermeldung ID1000 (#ERROR[ID1000])</Message>
    </Event>
</Source>
```

Abb. 11: Codeauszug – Ausgangslage

Exemplarisch wird in der Datei der Fehler mit der ID 455 integriert:

```

<Source>
<GUID>{3FD56AAE-0711-4359-89A5-3E1ECCCC634E}</GUID>
  <Id>650</Id>
  <Name>TwinCAT CNC User Errors</Name>
  <Event>
    <!-- User specific error messages: ID range 1 - 1000 -->
    <Id>1</Id>
      <Message LcId="1033">Error message ID1 (#ERROR[ID1])</Message>
      <Message LcId="1031">Fehlermeldung ID1 (#ERROR[ID1])</Message>
    </Event>
  <Event>
    <Id>2</Id>
      <Message LcId="1033">Error message ID2 (#ERROR[ID2])</Message>
      <Message LcId="1031">Fehlermeldung ID2 (#ERROR[ID2])</Message>
    </Event>
  <Event>
    <Id>455</Id>
      <Message LcId="1033">Error message ID455 (#ERROR[ID455])</Message>
      <Message LcId="1031">Fehlermeldung ID455 (#ERROR[ID455])</Message>
    </Event>
  <Event>
    <Id>1000</Id>
      <Message LcId="1033">Error message ID1000 (#ERROR[ID1000])</Message>
      <Message LcId="1031">Fehlermeldung ID1000 (#ERROR[ID1000])</Message>
    </Event>
</Source>

```

Abb. 12: Codeauszug mit integriertem Fehlertext



Ab TwinCAT-Version 4022.23 ist nach Integration des eigendefinierten Fehlertextes ein erneutes Aktivieren der Konfiguration erforderlich.

Bei TwinCAT-Versionen < 4022.23 ist ein Neustart des Rechners erforderlich



## 2.5 Fehlerausgabe TwinCAT3

Die Ausgabe von Fehlern unter TwinCAT3 erfolgt

- über die SPS oder
- direkt über den TwinCAT-Eventlogger.

Das nachfolgende Übersichtsbild verdeutlicht den Zusammenhang des Eventloggers:

### Architecture: Sending / Receiving BECKHOFF

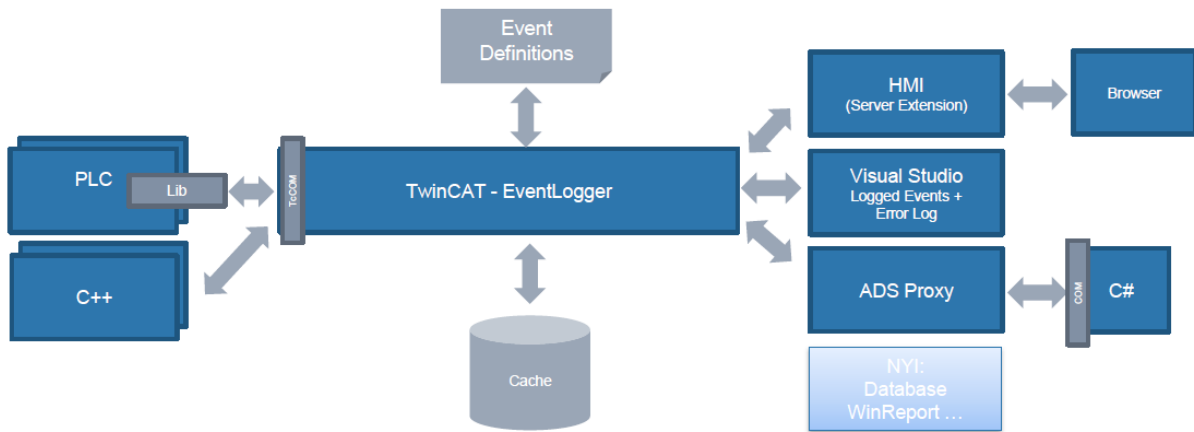


Abb. 13: Architektur- Übersicht TwinCAT-Eventlogger

### Ausgaben unter TwinCAT 3

In TwinCAT 3 ist der PRINT-Kanal an die Ausgabe zum Event-Logger angebunden:

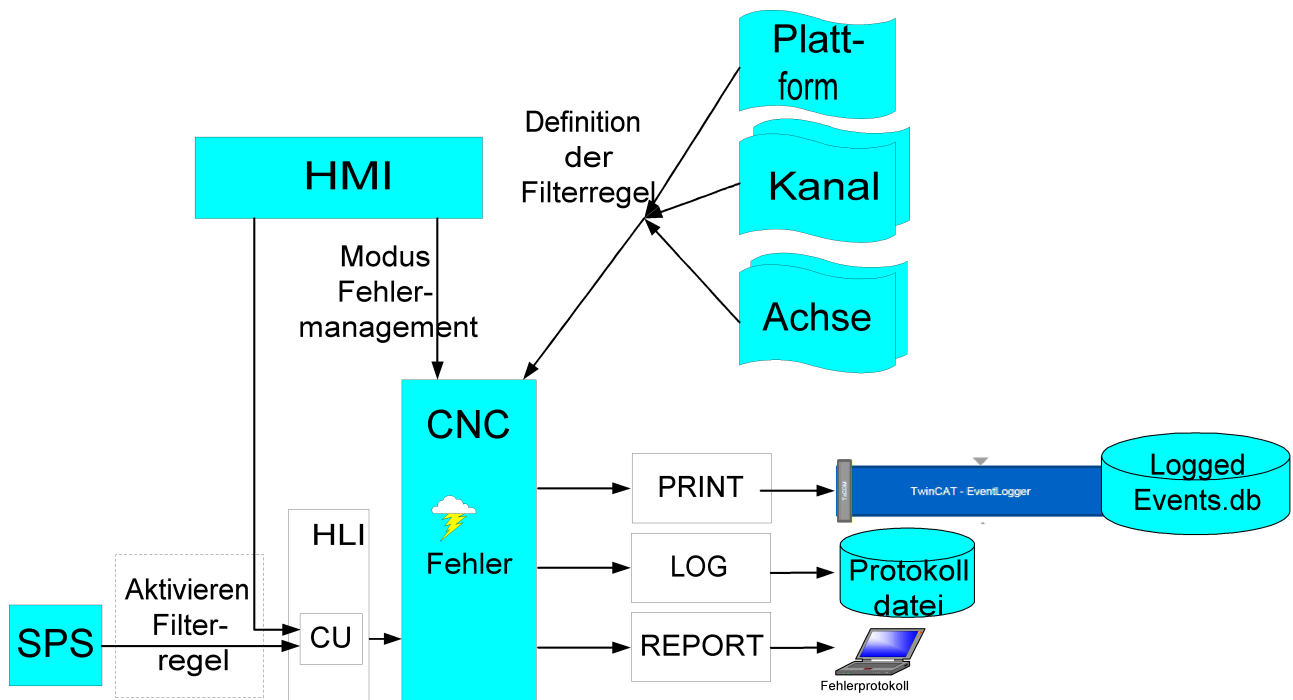


Abb. 14: Übersicht mit TwinCAT-Eventlogger

## Standardmäßige Parametrierung

Standardmäßig erfolgt die Ausgabe der Fehler an die SPS mit dem Loggen über den ursprünglichen Funktionsbaustein ChannelError an den Eventlogger.

Bei einer Migration einer bestehenden CNC Konfiguration von TwinCAT2 nach TwinCAT3 ist hierfür in den Hochlaufparametern bezüglich des Fehlermanagements keine Anpassung in [P-STUP-00167 \[► 32\]](#) (error\_protocol\_mode) erforderlich.

Parameter	Parametername	Einstellung
P-STUP-00167	error_protocol_mode	PRINT   LOG   REPORT
P-STUP-00168	error_text_of_id	<TC3-Install>\components\mc\cnc\diagnostics\err_text_version_eng.txt
P-STUP-00169	error_text_user_of_id	nicht belegt
P-STUP-00170	error_log_file_name	<TC3-Install> \components\mc\cnc\diagnostics\error.log
P-STUP-00171	error_log_file_max_size	100000 # maximum length in bytes
P-STUP-00172	error_plc_wait_cycles	5 # Wartezyklen [IPO ticks]
P-STUP-00173	error_ao_name	(nicht belegt)

## Fehlertexte

Die sprachspezifischen Fehlertexte werden bei einem Release der CNC über die Datei TcCncErrors.xml automatisch installiert.

## Sehen Sie dazu auch

 [CNC-Objekte \[► 45\]](#)

## 2.5.1 Ausgabe nur über SPS

Soll die Ausgabe an den Event-Logger nur über den ChannelError() Funktionsbaustein (FB) der SPS erfolgen, so kann dies durch Schreiben des CNC-Objektes „[log errors \[► 46\]](#)“ angewählt werden. Die Aktivierung findet normalerweise beim Start der SPS statt, wenn diese das Loggen der Fehlermeldungen übernimmt.

### Ausgabe über den FB ChannelError()

Werden die Fehler über die SPS-Schnittstelle an die SPS gemeldet, so können Fehler innerhalb der SPS ausgewertet werden und über einen Baustein FB ChannelError() an den Event-Logger gemeldet werden. Neben der Ausgabe des Fehlers kann die SPS hier evtl. unmittelbare Reaktionen auf einen individuellen Fehler ausführen.

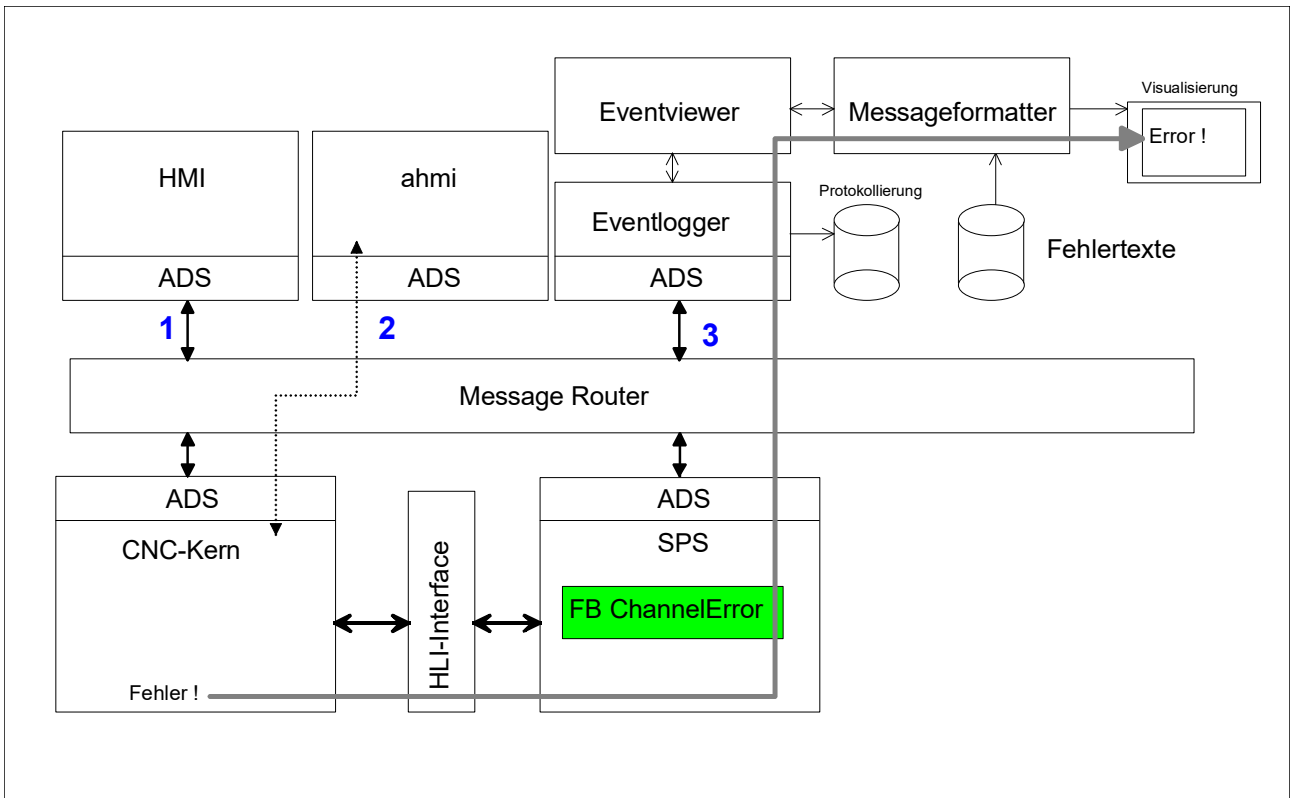


Abb. 15: FB ChannelError()- Platzierung im System

Diese Ausgabe ist identisch mit der von TwinCAT2.

## 2.5.2 Direkte Ausgabe an Eventlogger

Um die Ausgabe direkt an den „neuen“ Eventlogger zu nutzen muss in [P-STUP-00167 \[► 32\]](#) das Schlüsselwort `TC3_EVENT_LOGGER` angegeben werden. Die Fehler werden dann direkt von der CNC im neuen Format geloggt und nicht mehr über den SPS-Baustein `ChannelError()`. Eine Ausgabe der Fehler an die SPS erfolgt weiterhin.

```
error_protocol_mode PRINT | LOG | REPORT | TC3_EVENT_LOGGER
```

Falls die SPS keine Fehlermeldungen der CNC mehr auswerten muss kann die Ausgabe der Fehler an die SPS zu unterbunden werden durch:

```
error_protocol_mode PRINT | LOG | REPORT | TC3_EVENT_LOGGER | SEND_NOT_TO_PLC
```

Alternativ kann der Modus auch über das CNC-Objekt `cnc_error_manager_mode w` [\[► 45\]](#) entsprechend gesetzt werden

### 2.5.2.1 Ausgabeformat-Eventlogger

Im Visualstudio kann die Ausgabe des Eventloggers in einem separaten Fenster dargestellt werden:

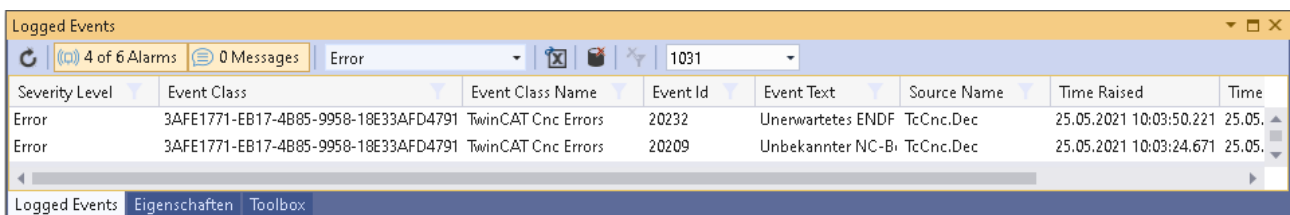


Abb. 16: Eventlogger-Output

## Übersicht-Ausgabeformat des Eventloggers

Reitername		Bedeutung
Severity Level	Klassifizierung	Warning (0) / Error (1-6) / Critical (7)
EventClassName	Name der Fehlerklasse	TwinCAT Cnc Errors
SourceName	Fehlerquelle	TcCNC Config / Diag / PosCtrl / PathPrep / AxesMgr / Commu / DrvMgr / Dec / TRC / Filter / FileMgr / SAI / IPO / Manual / SysMgr / Application
EventId	ID des Events	CNC-spezifische ID der aufgetretenen Meldung
Text	Fehlertext	Multilingualer Fehlertext
Time Raised	Auslösezeitpunkt	Datum und Uhrzeit
Json Attributes	Json Attribute	

### Json Attribute

Zusätzliche Parameter einer Fehlermeldung werden als Key/Value-Paare im Json-Format ausgegeben. Insbesondere stehen somit detailliertere Fehlerinformationen zum NC-Kanal oder Bezug zum NC-Programm zur Verfügung.

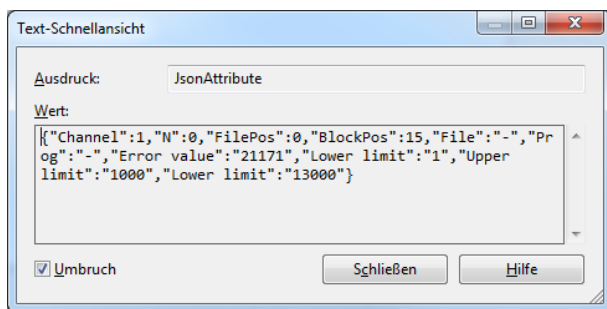


Abb. 17: Json-Attribute Output

### 2.5.3 Ausgabebeispiel an HMI

Die im Event-Logger aufgezeichneten Fehlermeldungen werden in der Oberfläche entsprechend dargestellt:

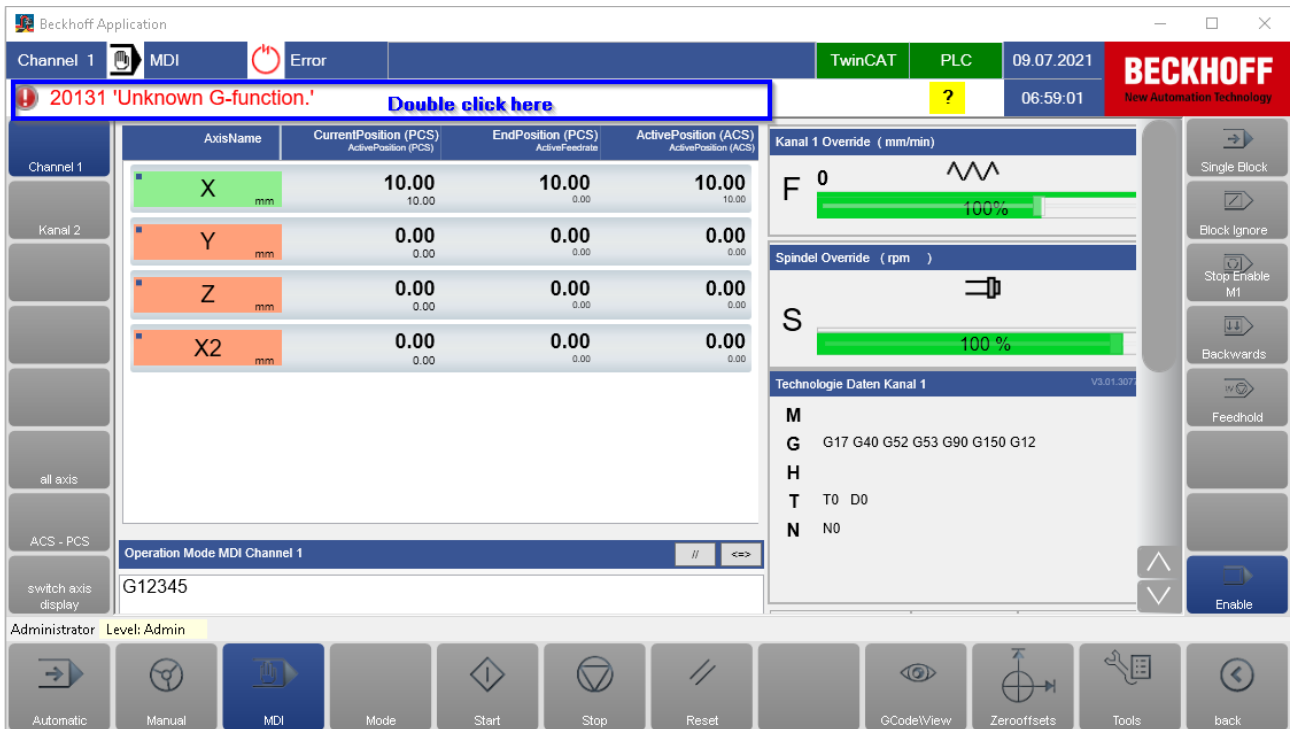


Abb. 18: Bildschirmausgabe der Fehlermeldung an der Beckhoff-HMI

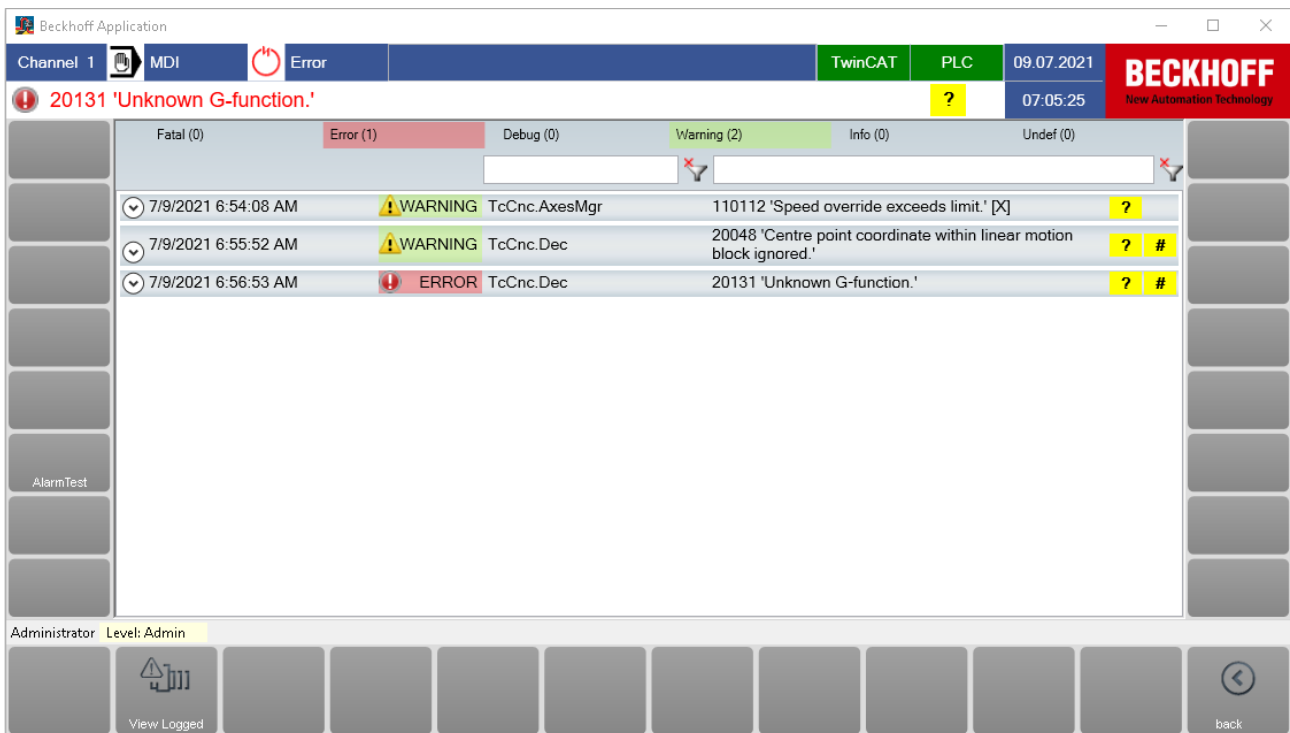


Abb. 19: Bildschirmausgabe nach Doppelklick

## 3 Parameter

### 3.1 Übersicht

#### Hochlaufparameter

ID	Parameter	Beschreibung
P-STUP-00166	no_error_message_at_reset	Protokollierung eines CNC-Resets
P-STUP-00167	error_protocol_mode	Logging Modus
P-STUP-00168	error_text_of_id	Name der Datei für Fehlermeldungstexte
P-STUP-00169	error_text_user_of_id	Name der Datei für anwenderspezifische Fehlermeldungstexte
P-STUP-00170	error_log_file_name	Name der Error-Logdatei
P-STUP-00171	error_log_file_max_size	Maximale Größe der Error-Logdatei
P-STUP-00172	error_plc_wait_cycles	Wartezyklen vor Auswertung der SPS Aktivierung
P-STUP-00173	error_ao_name	Zusätzlicher Beschreibungstext (AO-Name)
P-STUP-00186	error_filter[i].reason	Identifikation des Fehlers (Ursache)
P-STUP-00187	error_filter[i].action	Fehleraktion, die ausgeführt werden soll
P-STUP-00188	error_filter[i].conditional_activation	Aktivierung über SPS / HMI
P-STUP-00189	error_filter[i].conditional_action	Aktion die erst nach Freigabe durch SPS / HMI ausgeführt werden soll
P-STUP-00190	error_filter[i].conditional_param	Zusätzlicher Vergleichsparameter im Falle einer Antriebsfehlermeldung
P-STUP-00191	error_filter[i].conditional_output	Individuelle zusätzliche Fehlerinformation zur Ausgabe

#### Kanalparameter

ID	Parameter	Beschreibung
P-CHAN-00338	mdi_log_file	Name der Handsatz-Logdatei
P-CHAN-00339	mdi_log_file_max_size	Maximale Größe der Handsatz-Logdatei
P-CHAN-00378	error_filter[i].reason	Identifikation des Fehlers (Ursache)
P-CHAN-00379	error_filter[i].action	Fehleraktion, die ausgeführt werden soll
P-CHAN-00380	error_filter[i].conditional_activation	Aktivierung über SPS / HMI
P-CHAN-00381	error_filter[i].conditional_action	Aktion die erst nach Freigabe durch SPS / HMI ausgeführt werden soll
P-CHAN-00382	error_filter[i].conditional_param	Zusätzlicher Vergleichsparameter im Falle einer Antriebsfehlermeldung
P-CHAN-00383	error_filter[i].conditional_output	Individuelle zusätzliche Fehlerinformation zur Ausgabe

#### Achsparameter

ID	Parameter	Beschreibung
P-AXIS-00627	error_filter[i].reason	Identifikation des Fehlers (Ursache)
P-AXIS-00628	error_filter[i].action	Fehleraktion, die ausgeführt werden soll
P-AXIS-00629	error_filter[i].conditional_activation	Aktivierung über SPS / HMI
P-AXIS-00630	error_filter[i].conditional_action	Aktion die erst nach Freigabe durch SPS / HMI ausgeführt werden soll

<b>P-AXIS-00631</b>	error_filter[j].conditional_param	Zusätzlicher Vergleichsparameter im Falle einer Antriebsfehlermeldung
<b>P-AXIS-00632</b>	error_filter[j].conditional_output	Individuelle zusätzliche Fehlerinformation zur Ausgabe

## 3.2 Beschreibung

### 3.2.1 Hochlaufparameter

<b>P-STUP-00166</b>	<b>Protokollierung eines CNC-Resets als Events in der Fehlermeldungs Ausgabe</b>
Beschreibung	Der Parameter definiert, ob der durch den Anwender ausgelöste CNC-Reset als Ereignis in das Protokoll der Fehlermeldungen mit aufgenommen wird. Bei einem CNC-Reset werden die bisher aufgetretenen Fehlermeldungen quittiert. Dies findet immer statt, unabhängig von der Einstellung von P-STUP-00166.
Parameter	no_error_message_at_reset
Datentyp	BOOLEAN
Datenbereich	0: ein CNC-Reset wird als Warnung 270076 in der Fehlermeldungs Ausgabe protokolliert 1 ein CNC-Reset wird nicht protokolliert
Dimension	----
Standardwert	0
Anmerkungen	Parameter nur wirksam in TwinCAT-Systemen.

<b>P-STUP-00167</b>	<b>Protokollmode der Fehlerausgabe</b>	
Beschreibung	Mit diesem Parameter kann die Ausgabe und der Umfang der Fehlerausgabe gesteuert werden.	
Parameter	error_protocol_mode	
Datentyp	STRING	
Datenbereich	<b>Kennung</b>	<b>Bedeutung</b>
	FILTER_OFF	Es werden keine Filter ausgewertet
	VERBOSE	Erweiterte interne Diagnose
	WITHOUT_ERROR_MANAGER	Direkte Ausgabe ohne Fehlermanagement
	PRINT	Ausgabe als Print ausführen
	LOG	Ausgabe auf Log-Datei protokollieren
	REPORT	Ausgabe auf Log-Datei protokollieren
	SEND_TO_PLC_NONE	Ausgabe an die SPS unterdrücken
	PRINT_EXTENDED	Erweiterte Print-Ausgabe
	LOG_EXTENDED	Erweiterte Log-Ausgabe
	REPORT_EXTENDED	Erweiterte applikationsspezifische Ausgabe
	PRINT_NO_WARNINGS	Warnungen werden in der Print-Ausgabe unterdrückt
	LOG_NO_WARNINGS	Warnungen werden in der Log-Ausgabe unterdrückt
	REPORT_NO_WARNINGS	Warnungen werden in der Report-Ausgabe unterdrückt
	SEND_TO_PLC_NO_WARNINGS	Warnungen an SPS unterdrücken
STARTUP_NO_WARNINGS	Warnungen während Steuerungshochlauf unterdrücken	
NO_WARNINGS	Unterdrücken aller Warnungen	
TC3_EVENT_LOGGER	Ausgabe an TC3-Event-Logger	
Dimension	----	
Standardwert	LOG   PRINT   REPORT	
Anmerkungen	<b>Hinweis:</b> Um beispielsweise Warnungen bei der Print-Ausgabe zu unterdrücken muss der gesamte Mode entsprechend gesetzt werden. error_protocol_mode LOG   PRINT   REPORT   PRINT_NO_WARNINGS	



<b>P-STUP-00168</b>	<b>Name der Datei für Fehlermeldungstexte</b>
Beschreibung	Name der Datei, welche die zur ID (Fehlernummer) gehörigen Fehlermeldungstexte enthält. Diese können zur Ausgabe in die Log-Datei verwendet werden. Anhand dieser Datei erfolgt die Zuordnung einer Fehlernummer zum zugehörigen Fehlermeldungstext.  Die Datei enthält für jede Fehler-ID eine Zeile nach folgendem Format : <Error-ID> TABULATOR <Error-Text>  Wird keine Datei angegeben, so wird die Standarddatei 'err_text_version_eng.txt' angenommen.
Parameter	error_text_of_id
Datentyp	STRING
Datenbereich	Maximal 256 Zeichen
Dimension	----
Standardwert	-
Anmerkungen	

<b>P-STUP-00169</b>	<b>Name der Datei für anwenderspezifische Fehlermeldungstexte</b>
Beschreibung	Vergleichbar zu Standardfehlertexten (siehe P-STUP-00168) kann der Anwender in dieser Datei auch anwenderspezifische Texte angeben. Diese Texte werden für Fehler-IDs im Bereich [1;1000] und für Fehler der McCOM-Schnittstellen verwendet. Anhand dieser Datei erfolgt die Zuordnung einer Fehlernummer zum zugehörigen anwenderspezifischen Fehlermeldungstext.  (ERR_KIN_TRAFO_CONFIG/-INITIALIZE/-FORWARD/-BACKWARD = 292030-292033). D.h. in diesem Falle werden die zurückgegebenen Fehler-IDs der McCOM-Methoden aufgelöst.
Parameter	error_text_user_of_id
Datentyp	STRING
Datenbereich	Maximal 256 Zeichen
Dimension	----
Standardwert	-
Anmerkungen	

<b>P-STUP-00170</b>	<b>Name der Error-Logdatei</b>
Beschreibung	Name der Error-Logdatei (inklusive Verzeichnis und Pfadangabe). Wird kein Name angegeben, so wird keine Logdatei erzeugt.  Wenn der Parameter nicht konfiguriert ist, dann wird die Error-Logdatei mit dem Standarddateinamen erzeugt.
Parameter	error_log_file_name
Datentyp	STRING
Datenbereich	Maximal 256 Zeichen
Dimension	----
Standardwert	error.log
Anmerkungen	

<b>P-STUP-00171</b>	<b>Maximale Größe der Error-Logdatei in Byte</b>
Beschreibung	Der Parameter definiert die maximale Größe der Error-Logdatei.
Parameter	error_log_file_max_size
Datentyp	SGN32
Datenbereich	> 0 :Maximale Größe der Error-Logdatei. Wird diese Größe überschritten, so wird die Originaldatei in eine Backupdatei kopiert (Erweiterung <name>.bak) und der Inhalt der Originaldatei gelöscht.  == 0 : Es wird keine Backupdatei angelegt.

Dimension	----
Standardwert	100000
Anmerkungen	

<b>P-STUP-00172</b>	<b>Wartezyklen vor Auswertung der PLC Aktivierung</b>
Beschreibung	Wartezyklen in CNC-Ticks nach Auftreten eines Fehlers, bis die Aktivierungsmaske der SPS für die Filterregeln ausgewertet wird.
Parameter	error_plc_wait_cycles
Datentyp	UNS32
Datenbereich	0 ... MAX(UNS32)
Dimension	----
Standardwert	-
Anmerkungen	

<b>P-STUP-00173</b>	<b>Zusätzlicher Beschreibungstext (AO-Name)</b>
Beschreibung	Beschreibungstext (Architecture Object), welcher bei einer Fehlermeldung zusätzlich angefügt wird.
Parameter	error_ao_name
Datentyp	STRING
Datenbereich	Maximal 83 Zeichen
Dimension	----
Standardwert	-
Anmerkungen	

<b>P-STUP-00186</b>	<b>Fehlerursache</b>
Beschreibung	<p>Die einzelnen Fehlerkennungen können als Nummern oder Texte aufgelistet werden, wobei die gesamte Zeile folgender Syntax entsprechen muss: ( number   text ) { , ( number   text ) }</p> <p>mit: number := CNC-Fehlernummer text := " fehlerspezifischer Text "</p> <p>Beispiel: error_filter[0].reason := "D012:", 123000, 123001</p> <p>Wird ein Fehler gemeldet, so wird in den definierten Plattform-/ Kanal-/Achsfilttern nachgesehen, ob hierfür eine benutzerspezifische Filterregel definiert ist.</p>
Parameter	error_filter[i].reason mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)
Datentyp	STRING
Datenbereich	Maximal 96 Zeichen
Dimension	----
Standardwert	-
Anmerkungen	

<b>P-STUP-00187</b>	<b>Fehleraktion</b>
Beschreibung	Aktion, die bei Auftreten des entsprechenden Fehlers durchgeführt werden soll.
Parameter	error_filter[i].action mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)
Datentyp	STRING

Datenbereich	ACTION = NONE   DRIVE_STATE_REQ   PRE_RUN_STATE_REQ   RUN_STATE_REQ  NONE: Keine Aktion DRIVE_STATE_REQ: Auslesen des Antriebsstatus PRE_RUN_STATE_REQ: Fehler während Hochlauf des Steuerungsbusses in PRE-Run-Zustand RUN_STATE_REQ: Fehler während Hochlauf des Steuerungsbusses in Run-Zustand
Dimension	----
Standardwert	-
Anmerkungen	Für SERCOS-Antriebs-Profile: DRIVE_STATE_REQ: S-0-0095 diagnostic PRE_RUN_STATE_REQ: S-0-0021: list of unknown operation data in CP2 -> CP3, command 127 RUN_STATE_REQ: S-0-0022: list of unknown operation data in CP3 -> CP4, command 128  Für ProfiDrive-Profile: <all actions> Parameter 945  Für CANopen-Profile <all actions> Parameter ID603F

<b>P-STUP-00188</b>	<b>Bedingte Aktivierung</b>
Beschreibung	Beim Setzen des entsprechenden Bits über die Oberfläche oder die SPS (HLI::Kontrollunit) wird diese Filterregel aktiviert.
Parameter	error_filter[i].conditional_activation mit i = 0 ... 3 (Maximale. Anzahl der Filter, applikationsspezifisch)
Datentyp	UNS32
Datenbereich	32 Bit
Dimension	----
Standardwert	0
Anmerkungen	Parametrierbeispiel: <i>error_filter[0].conditional.action_activation 0x2</i>  Ein Aktivierungsbit = 0 bedeutet, dass die Aktion immer durchgeführt wird.

<b>P-STUP-00189</b>	<b>Bedingte Aktion</b>
Beschreibung	Aktion, die bei Auftreten des entsprechenden Fehlers und beim Zutreffen der Bedingung durchgeführt werden soll.
Parameter	error_filter[i].conditional_action mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)
Datentyp	STRING

Datenbereich	<p>ACTION = NONE   ( [ HIDE ] [ FORCE ] )</p> <p>FORCE = F_WARNING   F_SYNTAX   F_ERROR   F_SEVERE   F_FATAL</p> <p>HIDE = [ HIDE ] [ HIDE_LOG ] [ HIDE_PRINT ] [ HIDE_REPORT ]</p> <p>NONE: keine Aktion</p> <p>HIDE: Jede Fehlerausgabe unterdrücken</p> <p>HIDE_LOG: Fehlerausgabe in Error-Log-Datei wird unterdrückt</p> <p>HIDE_DISPLAY: Fehlerausgabe wird unterdrückt</p> <p>HIDE_REPORT: Applikationsspezifische Fehlerausgabe wird unterdrückt</p> <p>F_WARNING: Fehler wird als WARNING ausgegeben (Behebungsklasse = 0)</p> <p>F_SYNTAX: Fehler wird als Syntaxfehler ausgegeben (Behebungsklasse = 2)</p> <p>F_ERROR: Fehler durch NC-Programm oder andere Bedienaktion (Fehlerbehebungsklasse = 5)</p> <p>F_SEVERE: Schwere Fehler, erfordert Warmstart, (Behebungsklasse = 6)</p> <p>F_FATAL: Schwere Fehler, erfordert kompletten Kaltstart (Behebungsklasse = 7)</p>
Dimension	----
Standardwert	-
Anmerkungen	

<b>P-STUP-00190</b>	<b>Bedingter Filteraktivierung</b>
Beschreibung	<p>Die einzelnen Fehlerkennungen können als Nummern oder Texte aufgelistet werden, wobei die gesamte Zeile folgender Syntax entsprechen muss:</p> <p>( number   text ) { , ( number   text ) }</p> <p>mit:</p> <p>number := CNC-Fehlernummer</p> <p>text := " fehlerspezifischer Text "</p>
Parameter	error_filter[i].conditional_param mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)
Datentyp	STRING
Datenbereich	Maximal 96 Zeichen
Dimension	----
Standardwert	-
Anmerkungen	<p>Parametrierbeispiel:</p> <p><i>error_filter[0].conditional.param "D012:", 123, 1001</i></p> <p>Individuelle Filtertexte werden nur beim Auslesen des SERCOS-Antriebsfehlers S95 geprüft.</p> <p>Fehlernummer werden nur bei SERCOS-Antriebsfehler (S21 und S22) und bei ProfiDrive-Antriebsfehler (Parameter 945) geprüft.</p>

<b>P-STUP-00191</b>	<b>Ausgabe einer zusätzlichen Fehlerinformation</b>
Beschreibung	Dieser Text wird im Falle des Zutreffens der Filterbedingung transparent über die CNC_ERROR_INFO-Datenstruktur weitergereicht. D.h. hierüber hat der Anwender die Möglichkeit, bedingt eine zusätzliche Fehlerinformation mit auszugeben.
Parameter	error_filter[i].conditional_output mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)

Datentyp	STRING
Datenbereich	Maximal 32 Zeichen
Dimension	----
Standardwert	-
Anmerkungen	

### 3.2.2 Kanalparameter

<b>P-CHAN-00338</b>	<b>Name der Handsatz-Logdatei</b>
Beschreibung	Wird der Name angegeben, so wird jedes Handsatzkommando des NC-Kanals in diese Datei protokolliert. Neben späterer Diagnosefähigkeit wird diese Datei auch für die Fehleranzeige genutzt. D.h. kommt es innerhalb des Handsatzes zu einem CNC-Fehler, so wird in der Fehlermeldung der kommandierte Handsatz gleich angezeigt.
Parameter	mdi_log_file
Datentyp	STRING
Datenbereich	Maximal 256 Zeichen
Dimension	----
Standardwert	*
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring.

<b>P-CHAN-00339</b>	<b>Maximale Größe der Handsatz-Logdatei</b>
Beschreibung	Da die Protokolldatei bei jedem neuen Handsatz anwächst, kann diese über diese Angabe in der Größe begrenzt werden. Wird die Größe der Protokolldatei überschritten, so wird diese vor dem Protokollieren des aktuellen Handsatzes zunächst automatisch geleert.
Parameter	mdi_log_file_max_size
Datentyp	UNS32
Datenbereich	0 ... MAX(UNS32)
Dimension	----
Standardwert	0 *
Anmerkungen	* keine Größenbegrenzung

<b>P-CHAN-00378</b>	<b>Fehlerursache (Filterung von Fehlermeldungen im Kanal)</b>
Beschreibung	Die einzelnen Fehlerkennungen können als Nummern oder Texte aufgelistet werden, wobei die gesamte Zeile folgender Syntax entsprechen muss: ( number   text ) { , ( number   text ) }  mit: number := CNC-Fehlernummer text := " fehlerspezifischer Text "  Beispiel: error_filter[0].reason := "D012:", 123000, 123001  Wird ein Fehler gemeldet, so wird in den definierten Plattform-/ Kanal-/Achsfiltren nachgesehen, ob hierfür eine benutzerspezifische Filterregel definiert ist.
Parameter	error_filter[i].reason mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)
Datentyp	STRING
Datenbereich	Maximal 96 Zeichen
Dimension	----
Standardwert	*
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring.

<b>P-CHAN-00379</b>	<b>Fehleraktion (Filterung von Fehlermeldungen im Kanal)</b>
Beschreibung	Aktion, die bei Auftreten des entsprechenden Fehlers durchgeführt werden soll.

Parameter	error_filter[i].action mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)	
Datentyp	STRING	
Datenbereich	ACTION = NONE   DRIVE_STATE_REQ   PRE_RUN_STATE_REQ   RUN_STATE_REQ	
	<b>Schlüsselwort</b>	<b>Bedeutung</b>
	NONE	Keine Aktion
	DRIVE_STATE_REQ	Auslesen des Antriebsstatus
	PRE_RUN_STATE_REQ	Fehler während Hochlauf des Steuerungsbusses in PRE-Run-Zustand
	RUN_STATE_REQ	Fehler während Hochlauf des Steuerungsbusses in Run-Zustand
Dimension	----	
Standardwert	*	
Anmerkungen	<p>Für SERCOS-Antriebs-Profile:</p> <ul style="list-style-type: none"> <li>• DRIVE_STATE_REQ S-0-0095 diagnostic</li> <li>• PRE_RUN_STATE_REQ S-0-0021: list of unknown operation data in CP2 -&gt; CP3, command 127</li> <li>• RUN_STATE_REQ S-0-0022: list of unknown operation data in CP3 -&gt; CP4, command 128</li> </ul> <p>Für ProfiDrive-Profile:</p> <ul style="list-style-type: none"> <li>• &lt;all actions&gt; Parameter 945</li> </ul> <p>Für CANopen-Profile</p> <ul style="list-style-type: none"> <li>• &lt;all actions&gt; Parameter ID603F</li> </ul> <p>* Hinweis: Der Standardwert der Variablen ist ein Leerstring.</p>	

<b>P-CHAN-00380</b>	<b>Bedingte Aktivierung (Filterung von Fehlermeldungen im Kanal)</b>
Beschreibung	Beim Setzen des entsprechenden Bits über die Oberfläche oder die SPS (HLI::ControlUnit) wird diese Filterregel aktiviert.
Parameter	error_filter[i].conditional_activation mit i = 0 ... 3 (Maximale. Anzahl der Filter, applikationsspezifisch)
Datentyp	UNS32
Datenbereich	32 Bit
Dimension	----
Standardwert	0
Anmerkungen	<p>Parametrierbeispiel:</p> <p><i>error_filter[0].conditional_action_activation 0x2</i></p> <p>Ein Aktivierungsbit = 0 bedeutet, dass die Aktion immer durchgeführt wird.</p>

<b>P-CHAN-00381</b>	<b>Bedingte Aktion (Filterung von Fehlermeldungen im Kanal)</b>
Beschreibung	Aktion, die bei Auftreten des entsprechenden Fehlers und beim Zutreffen der Bedingung durchgeführt werden soll.
Parameter	error_filter[i].conditional_action mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)
Datentyp	STRING

Datenbereich	<p>ACTION = NONE   ( [ HIDE ] [ FORCE ] )</p> <p>FORCE = F_WARNING   F_SYNTAX   F_ERROR   F_SEVERE   F_FATAL</p> <p>HIDE = [ HIDE ] [ HIDE_LOG ] [ HIDE_PRINT ] [ HIDE_REPORT ]</p> <p>NONE: keine Aktion</p> <p>HIDE: Jede Fehlerausgabe unterdrücken</p> <p>HIDE_LOG: Fehlerausgabe in Error-Log-Datei wird unterdrückt</p> <p>HIDE_DISPLAY: Fehlerausgabe wird unterdrückt</p> <p>HIDE_REPORT: Applikationsspezifische Fehlerausgabe wird unterdrückt</p> <p>F_WARNING: Fehler wird als WARNING ausgegeben (Behebungsstufe = 0)</p> <p>F_SYNTAX: Fehler wird als Syntaxfehler ausgegeben (Behebungsstufe = 2)</p> <p>F_ERROR: Fehler durch NC-Programm oder andere Bedienaktion (Fehlerbehebungsstufe = 5)</p> <p>F_SEVERE: Schwere Fehler, erfordert Warmstart, (Behebungsstufe = 6)</p> <p>F_FATAL: Schwere Fehler, erfordert kompletten Kaltstart (Behebungsstufe = 7)</p>
Dimension	----
Standardwert	*
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring.

<b>P-CHAN-00382</b>	<b>Bedingter Filteraktivierung (Filterung von Fehlermeldungen im Kanal)</b>
Beschreibung	<p>Die einzelnen Fehlerkennungen können als Nummern oder Texte aufgelistet werden, wobei die gesamte Zeile folgender Syntax entsprechen muss:</p> <p>( number   text ) { , ( number   text ) }</p> <p>mit:</p> <p>number := CNC-Fehlernummer</p> <p>text := " fehlerspezifischer Text "</p>
Parameter	error_filter[i].conditional_param mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)
Datentyp	STRING
Datenbereich	Maximal 96 Zeichen
Dimension	----
Standardwert	*
Anmerkungen	<p>Parametrierbeispiel:</p> <p><i>error_filter[0].conditional_param "D012:", 123, 1001</i></p> <p>Individuelle Filtertexte werden aktuell nur beim Auslesen des SERCOS-Antriebsfehlers S95 geprüft.</p> <p>Fehlernummer werden nur bei SERCOS-Antriebsfehler (S21 und S22) und bei ProfiDrive-Antriebsfehler (Parameter 945) geprüft.</p> <p>* Hinweis: Der Standardwert der Variablen ist ein Leerstring.</p>

<b>P-CHAN-00383</b>	<b>Ausgabe einer zusätzlichen Fehlerinformation (Filterung von Fehlermeldungen im Kanal)</b>
---------------------	--



Beschreibung	Dieser Text wird bei zutreffender Filterbedingung transparent über die CNC_ERROR_INFO-Datenstruktur weitergereicht. D.h. hierüber hat der Anwender die Möglichkeit, bedingt eine zusätzliche Fehlerinformation mit auszugeben.
Parameter	error_filter[i].conditional_output mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)
Datentyp	STRING
Datenbereich	Maximal 32 Zeichen
Dimension	----
Standardwert	*
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring.

### 3.2.3 Achsparameter

<b>P-AXIS-00627</b>	<b>Fehlerursache (Filterung von Achsfehlermeldungen)</b>	
Beschreibung	<p>Die einzelnen Fehlerkennungen können als Nummern oder Texte aufgelistet werden, wobei die gesamte Zeile folgender Syntax entsprechen muss: ( number   text ) {, ( number   text ) }</p> <p>mit:</p> <p style="padding-left: 40px;">number := CNC-Fehlernummer</p> <p style="padding-left: 40px;">text := "fehlerspezifischer Text"</p> <p>Beispiel:</p> <p style="padding-left: 40px;">error_filter[0].reason := "D012:", 123000, 123001</p> <p>Wird ein Fehler gemeldet, so wird in den definierten Plattform-/ Kanal-/Achsfilttern nachgesehen, ob hierfür eine benutzerspezifische Filterregel definiert ist.</p>	
Parameter	error_filter[i].reason mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)	
Datentyp	STRING	
Datenbereich	Maximal 96 Zeichen	
Achstypen	T, R, S	
Dimension	T: ----	R,S: ----
Standardwert	*	
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring.	

<b>P-AXIS-00628</b>	<b>Fehleraktion (Filterung von Achsfehlermeldungen)</b>	
Beschreibung	Aktion, die bei Auftreten des entsprechenden Fehlers durchgeführt werden soll.	
Parameter	error_filter[i].action mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)	
Datentyp	STRING	
Datenbereich	<p>ACTION = NONE   DRIVE_STATE_REQ   PRE_RUN_STATE_REQ   RUN_STATE_REQ</p> <p>NONE: Keine Aktion</p> <p>DRIVE_STATE_REQ: Auslesen des Antriebsstatus</p> <p>PRE_RUN_STATE_REQ: Fehler während Hochlauf des Steuerungsbusses in PRE-Run-Zustand</p> <p>RUN_STATE_REQ: Fehler während Hochlauf des Steuerungsbusses in Run-Zustand</p>	
Achstypen	T, R, S	
Dimension	T: ----	R,S: ----
Standardwert	*	

Anmerkungen	<p>Für SERCOS-Antriebs-Profile:</p> <p>DRIVE_STATE_REQ      S-0-0095 diagnostic</p> <p>PRE_RUN_STATE_REQ    S-0-0021: list of unknown operation data in CP2 -&gt; CP3, command 127</p> <p>RUN_STATE_REQ        S-0-0022: list of unknown operation data in CP3 -&gt; CP4, command 128</p> <p>Für Profidrive-Profile:</p> <p>&lt;all actions&gt;            Parameter 945</p> <p>Für CANopen-Profile</p> <p>&lt;all actions&gt;            Parameter ID603F</p> <p>* Hinweis: Der Standardwert der Variablen ist ein Leerstring.</p>
-------------	---

<b>P-AXIS-00629</b>	<b>Bedingte Aktivierung (Filterung von Achsfehlermeldungen)</b>	
Beschreibung	Beim Setzen des entsprechenden Bits über die Oberfläche oder die SPS (HLI::Kontrollunit) wird diese Filterregel aktiviert.	
Parameter	error_filter[i].conditional_activation mit i = 0 ... 3 (Maximale. Anzahl der Filter, applikationsspezifisch)	
Datentyp	UNS32	
Datenbereich	32 Bit	
Achstypen	T, R, S	
Dimension	T: ----	R,S: ----
Standardwert	0	
Anmerkungen	<p>Parametrierbeispiel:</p> <p><i>error_filter[0].conditional.action_activation 0x2</i></p> <p>Ein Aktivierungsbit = 0 bedeutet, dass die Aktion immer durchgeführt wird.</p>	

<b>P-AXIS-00630</b>	<b>Bedingte Aktion (Filterung von Achsfehlermeldungen)</b>	
Beschreibung	Aktion, die bei Auftreten des entsprechenden Fehlers und beim Zutreffen der Bedingung durchgeführt werden soll.	
Parameter	error_filter[i].conditional_action mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)	
Datentyp	STRING	
Datenbereich	<p>ACTION = NONE   ( [ HIDE ] [ FORCE ] )</p> <p>FORCE = F_WARNING   F_SYNTAX   F_ERROR   F_SEVERE   F_FATAL</p> <p>HIDE = [ HIDE ] [ HIDE_LOG ] [ HIDE_PRINT ] [ HIDE_REPORT ]</p>	
	NONE:	keine Aktion
	HIDE:	Jede Fehlerausgabe unterdrücken
	HIDE_LOG:	Fehlerausgabe in Error-Log-Datei wird unterdrückt
	HIDE_DISPLAY:	Fehlerausgabe wird unterdrückt
	HIDE_REPORT:	Applikationsspezifische Fehlerausgabe wird unterdrückt
	F_WARNING:	Fehler wird als WARNING ausgegeben (Behungsklasse = 0)
	F_SYNTAX:	Fehler wird als Syntaxfehler ausgegeben (Behebungs-klasse = 2)
	F_ERROR:	Fehler durch NC-Programm oder andere Bedienaktion (Fehlerbehebungs-klasse = 5)
	F_SEVERE:	Schwerer Fehler, erfordert Warmstart, (Behebungs-klasse = 6)
	F_FATAL:	Schwerer Fehler, erfordert kompletten Kaltstart (Behebungs-klasse = 7)
Achstypen	T, R, S	

Dimension	T: ----	R,S: ----
Standardwert	*	
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring.	

<b>P-AXIS-00631</b>	<b>Bedingte Filteraktivierung (Filterung von Achsfehlermeldungen)</b>	
Beschreibung	Die einzelnen Fehlerkennungen können als Nummern oder Texte aufgelistet werden, wobei die gesamte Zeile folgender Syntax entsprechen muss: ( number   text ) { , ( number   text ) }  mit: number := CNC-Fehlernummer text := " fehlerspezifischer Text "	
Parameter	error_filter[i].conditional_param mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)	
Datentyp	STRING	
Datenbereich	Maximal 96 Zeichen	
Achstypen	T, R, S	
Dimension	T: ----	R,S: ----
Standardwert	*	
Anmerkungen	<p>Parametrierbeispiel: <i>error_filter[0].conditional.param "D012:", 123, 1001</i></p> <p>Individuelle Filtertexte werden aktuell nur beim Auslesen des SERCOS-Antriebsfehlers S95 geprüft.</p> <p>Fehlernummer werden nur bei SERCOS-Antriebsfehler (S21 und S22) und bei Profidrive-Antriebsfehler (Parameter 945) geprüft.</p> <p>* Hinweis: Der Standardwert der Variablen ist ein Leerstring.</p>	

<b>P-AXIS-00632</b>	<b>Ausgabe einer zusätzlichen Fehlerinformation (Filterung von Achsfehlermeldungen)</b>	
Beschreibung	Dieser Text wird im Falle des Zutreffens der Filterbedingung transparent über die CNC_ERROR_INFO-Datenstruktur weitergereicht. D.h. hierüber hat der Anwender die Möglichkeit, bedingt eine zusätzliche Fehlerinformation mit auszugeben.	
Parameter	error_filter[i].conditional_output mit i = 0 ... 3 (Maximale Anzahl der Filter, applikationsspezifisch)	
Datentyp	STRING	
Datenbereich	Maximal 32 Zeichen	
Achstypen	T, R, S	
Dimension	T: ----	R,S: ----
Standardwert	*	
Anmerkungen	* Hinweis: Der Standardwert der Variablen ist ein Leerstring.	

### 3.3 CNC-Objekte

#### Hinweise zur Adressierung

#### Aktivieren von Filterregeln

<b>Name</b>	cnc_error_filter_w
-------------	--------------------

<b>Beschreibung</b>	Die einzelnen Filterregeln der Plattform lassen sich über die Oberfläche durch Setzen der entsprechenden Bitmaske ein- und ausschalten. Voraussetzung hierfür ist die Parametrierung von P-STUP-00188. Es lassen sich somit maximal 32 verschiedene Filterregeln schalten. Beispiel der zugeordneten Filterdefinition: error_filter[0].activation_bit                    0x1		
<b>Task</b>	COM (Port 553)		
<b>Indexgruppe</b>	0x120100	<b>Indexoffset</b>	0x26C
<b>Datentyp</b>	UNS32	<b>Länge</b>	4
<b>Attribute</b>	write	<b>Einheit</b>	-
<b>Anmerkungen</b>	Siehe Fehlerfilterdefinition [► 18] in Hochlaufliste		

<b>Name</b>	mc_error_filter_w		
<b>Beschreibung</b>	Die einzelnen Filterregeln eines Kanals lassen sich über die Oberfläche durch Setzen der entsprechenden Bitmaske ein- und ausschalten. Voraussetzung hierfür ist die Parametrierung von P-CHAN-00380. Es lassen sich somit maximal 32 verschiedene Filterregeln schalten. Beispiel der zugeordneten Filterdefinition: error_filter[0].activation_bit                    0x1		
<b>Task</b>	COM (Port 553)		
<b>Indexgruppe</b>	0x12010<C <sub>ID</sub> >	<b>Indexoffset</b>	0x419
<b>Datentyp</b>	UNS32	<b>Länge</b>	4
<b>Attribute</b>	write	<b>Einheit</b>	-
<b>Anmerkungen</b>			

<b>Name</b>	ac_<A <sub>ID</sub> >_error_filter_w		
<b>Beschreibung</b>	Die einzelnen Filterregeln einer Achse lassen sich über die Oberfläche durch Setzen der entsprechenden Bitmaske ein- und ausschalten. Voraussetzung dafür ist Parametrierung von P-AXIS-00629. Es lassen sich somit maximal 32 verschiedene Filterregeln schalten. Beispiel der zugeordneten Filterdefinition: error_filter[0].activation_bit                    0x1		
<b>Task</b>	COM (Port 553)		
<b>Indexgruppe</b>	0x120200	<b>Indexoffset</b>	0x<A <sub>ID</sub> >0028
<b>Datentyp</b>	UNS32	<b>Länge</b>	4
<b>Attribute</b>	write	<b>Einheit</b>	-
<b>Anmerkungen</b>			

**Zugriff auf Protokollierungsmodus**

<b>Name</b>	cnc_error_manager_mode_r		
<b>Beschreibung</b>	CNC Objekt zum Lesen des Mode des Fehlermanagements. Siehe P-STUP-00167.		
<b>Task</b>	COM (Port 553)		
<b>Indexgruppe</b>	0x120100	<b>Indexoffset</b>	0x293
<b>Datentyp</b>	UN32	<b>Länge/Byte</b>	4
<b>Attribute</b>	read	<b>Einheit</b>	-
<b>Anmerkungen</b>	Die Zuordnung und Bedeutung der Bits sind in der Tabelle [► 46] beschrieben.		

<b>Name</b>	cnc_error_manager_mode_w		
-------------	--------------------------	--	--

<b>Beschreibung</b>	CNC Objekt zum Schreiben des Mode des Fehlermanagements. Siehe P-STUP-00167.		
<b>Task</b>	COM (Port 553)		
<b>Indexgruppe</b>	0x120100	<b>Indexoffset</b>	0x26D
<b>Datentyp</b>	UN32	<b>Länge/Byte</b>	4
<b>Attribute</b>	write	<b>Einheit</b>	-
<b>Anmerkungen</b>	Die Zuordnung und Bedeutung der Bits sind in der <a href="#">Tabelle [► 46]</a> beschrieben.		

<b>Kennung</b>	<b>Bitbelegung</b>	<b>Bedeutung</b>
FILTER_OFF	0x00004	Es werden keine Filter ausgewertet
VERBOSE	0x00008	Erweiterte interne Diagnose
WITHOUT_ERROR_MANAGEMENT	0x00001	Direkte Ausgabe ohne Fehlermanagement
PRINT	0x00010	Ausgabe als Print ausführen
LOG	0x00020	Ausgabe auf Log-Datei protokollieren
REPORT	0x00040	Ausgabe auf Log-Datei protokollieren
SEND_TO_PLC_NONE	0x20000	Ausgabe an die SPS unterdrücken
PRINT_EXTENDED	0x00100	Erweiterte Print-Ausgabe
LOG_EXTENDED	0x00200	Erweiterte Log-Ausgabe
REPORT_EXTENDED	0x00400	Erweiterte applikationsspezifische Ausgabe
PRINT_NO_WARNINGS	0x01000	Warnungen werden in der Print-Ausgabe unterdrückt
LOG_NO_WARNINGS	0x02000	Warnungen werden in der Log-Ausgabe unterdrückt
REPORT_NO_WARNINGS	0x04000	Warnungen werden in der Report-Ausgabe unterdrückt
SEND_TO_PLC_NO_WARNINGS	0x08000	Warnungen an SPS unterdrücken
STARTUP_NO_WARNINGS	0x10000	Warnungen während Steuerungshochlauf unterdrücken
NO_WARNINGS	0x1f000	Unterdrücken aller Warnungen
TC3_EVENT_LOGGER	0x80000	Ausgabe an TC3-Event-Logger

<b>Name</b>	log errors		
<b>Beschreibung</b>	Mit diesem Objekt kann festgelegt werden ob Fehlermeldungen der CNC über an die SPS ausgegeben werden. (Standard = TRUE)		
<b>Task</b>	GEO (Port 551)		
<b>Indexgruppe</b>	0x120300	<b>Indexoffset</b>	0x4
<b>Datentyp</b>	BOOLEAN	<b>Länge</b>	1
<b>Attribute</b>	read/ write	<b>Einheit</b>	[-]
<b>Anmerkungen</b>	Werte : TRUE/FALSE		

### 3.4 HLI-Parameter

Aktivieren der Fehlerfilterregeln - Plattform	
Beschreibung	Die einzelnen Fehlerfilter können hier entsprechend ihres Aktivierungsbits durch die 32-Bit-Maske ein-/ausgeschaltet werden. Z.B. Nachfolgende Regel wird durch Setzen des ersten Bits (command_w = 0x00000001) wirksam: <pre>error_filter[0].activation_bit      0x1</pre>
Datentyp	MC_CONTROL_UN32_UNIT, s. Beschreibung Control Unit
Einheit	[-]
Zugriff	PLC liest request_r + state_r und schreibt command_w + enable_w
ST-Pfad	gpPform^.error_filter
Kommandierter und angeforderter Wert	
ST-Element	<b>.command_w</b> <b>.request_r</b>
Datentyp	UDINT
Rückgabewert	
ST-Element	<b>.state_r</b>
Datentyp	UDINT
Umleitung	
ST-Element	<b>.enable_w</b>

Aktivieren der Fehlerfilterregeln - Kanal	
Beschreibung	Die einzelnen Fehlerfilter können hier entsprechend ihres Aktivierungsbits durch die 32-Bit-Maske ein-/ausgeschaltet werden. Z.B. Nachfolgende Regel wird durch Setzen des ersten Bits (command_w = 0x00000001) wirksam: <pre>error_filter[0].activation_bit      0x1</pre>
Datentyp	MC_CONTROL_UN32_UNIT, s. Beschreibung Control Unit
Einheit	[-]
Zugriff	PLC liest request_r + state_r und schreibt command_w + enable_w
ST-Pfad	gpCh[channel_idx]^head.error_filter
Kommandierter und angeforderter Wert	
ST-Element	<b>.command_w</b> <b>.request_r</b>
Datentyp	UDINT
Rückgabewert	
ST-Element	<b>.state_r</b>
Datentyp	UDINT
Umleitung	
ST-Element	<b>.enable_w</b>

Aktivieren der Fehlerfilterregeln - Achse	
---	--

Beschreibung	<p>Die einzelnen Fehlerfilter können hier entsprechend ihres Aktivierungsbits durch die 32-Bit-Maske ein-/ausgeschaltet werden.</p> <p>Z.B. Nachfolgende Regel wird durch Setzen des ersten Bits (command_w = 0x00000001) wirksam:</p> <pre>error_filter[0].activation_bit      0x1</pre>
Datentyp	MC_CONTROL_UN32_UNIT, s. Beschreibung Control Unit
Einheit	[-]
Zugriff	PLC liest request_r + state_r und schreibt command_w + enable_w
ST-Pfad	gpAx[axis_idx]^head.error_filter
Kommandierter und angeforderter Wert	
ST-Element	<b>.command_w</b> <b>.request_r</b>
Datentyp	UDINT
Rückgabewert	
ST-Element	<b>.state_r</b>
Datentyp	UDINT
Umleitung	
ST-Element	<b>.enable_w</b>



## 4 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

### Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den lokalen Support und Service zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unseren Internetseiten: <https://www.beckhoff.de>

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49(0)5246 963 157  
Fax: +49(0)5246 963 9157  
E-Mail: [support@beckhoff.com](mailto:support@beckhoff.com)

### Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49(0)5246 963 460  
Fax: +49(0)5246 963 479  
E-Mail: [service@beckhoff.com](mailto:service@beckhoff.com)

### Beckhoff Firmenzentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49(0)5246 963 0  
Fax: +49(0)5246 963 198  
E-Mail: [info@beckhoff.com](mailto:info@beckhoff.com)  
Internet: <https://www.beckhoff.de>

# Stichwortverzeichnis

## F

---

### Fehlerfilter

aktivieren-Achse	47
aktivieren-Kanal	47
aktivieren-Plattform	47

## P

---

P-AXIS-00627	42
P-AXIS-00628	42
P-AXIS-00629	43
P-AXIS-00630	43
P-AXIS-00631	44
P-AXIS-00632	44
P-CHAN-00338	38
P-CHAN-00339	38
P-CHAN-00378	38
P-CHAN-00379	38
P-CHAN-00380	39
P-CHAN-00381	39
P-CHAN-00382	40
P-CHAN-00383	40
P-STUP-00166	32
P-STUP-00167	32
P-STUP-00168	33
P-STUP-00169	33
P-STUP-00170	33
P-STUP-00171	33
P-STUP-00172	34
P-STUP-00173	34
P-STUP-00186	34
P-STUP-00187	34
P-STUP-00188	35
P-STUP-00189	35
P-STUP-00190	36
P-STUP-00191	36



Mehr Informationen:  
[www.beckhoff.de/TF5200](http://www.beckhoff.de/TF5200)

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

