

Typ3 osa

CPL-Debugger Bedienungsanleitung



Ausgabe

101



BOSCH
Automation

Typ3 osa

CPL-Debugger Bedienungsanleitung

1070 073 872-101 (01.10) D



© 2001

Alle Rechte bei Robert Bosch GmbH,
auch für den Fall von Schutzrechtsanmeldungen.
Jede Verfügungsbefugnis, wie Kopier- und Weitergaberecht, bei uns.

Schutzgebühr 6.– EUR

Inhaltsverzeichnis

	Seite
1	Sicherheitshinweise 1–1
1.1	Bestimmungsgemäßer Gebrauch 1–1
1.2	Qualifiziertes Personal 1–2
1.3	Sicherheitshinweise an den Produkten 1–3
1.4	Sicherheitshinweise in diesem Handbuch 1–4
1.5	Sicherheitsanweisungen für das beschriebene Produkt 1–5
1.6	Dokumentation, Version und Warenzeichen 1–7
2	Übersicht 2–1
3	Installation und Start 3–1
3.1	CPL-Debugger installieren 3–1
3.2	CPL-Debugger starten 3–2
3.3	Konfigurationsdatei 3–4
3.3.1	Konfigurationsdatei erstellen 3–4
3.3.2	Konfigurationsdatei laden 3–5
4	Bedienoberfläche 4–1
4.1	Übersicht 4–1
4.2	Arbeitsbereich 4–3
4.3	Softkey-Leiste 4–4
4.4	Symbole (Images) 4–6
4.5	Text suchen 4–8
4.6	Fenstereigenschaften zurücksetzen 4–8
4.7	Hilfe 4–8
5	Arbeitsweise CPL-Debugger 5–1
5.1	Voraussetzungen 5–1
5.1.1	Steuerung und Kanal auswählen 5–2
5.1.2	Gesamtsperre 5–2
5.1.3	Programm anwählen 5–5
5.2	NC-Programm im Debug-Mode starten 5–6
5.3	NC-Programm anzeigen 5–9
5.3.1	Breakpoints setzen und editieren 5–12
5.3.2	Variablen beobachten 5–15
5.3.3	Meldungen 5–21
5.4	Aktives Programm nach Abbruch bearbeiten 5–22
6	Meldungen im CPL-Debugger 6–1
6.1	Fehler-Klassen CPL-Debugger 6–1
6.2	Popup-Fenster der Fehler-Klassen (Beispiele) 6–2
6.3	Benutzer-Reaktion auf Fehler und Meldungen des Systems .. 6–3
6.4	Fehlermeldungen 6–3
A	Anhang A–1
A.1	Abkürzungen A–1
A.2	Stichwortregister A–2

1 Sicherheitshinweise

Lesen Sie dieses Handbuch, bevor Sie mit dem CPL-Debugger arbeiten. Bewahren Sie dieses Handbuch an einem für alle Benutzer jederzeit zugänglichen Platz auf.

1.1 Bestimmungsgemäßer Gebrauch

Dieses Handbuch enthält Angaben zum bestimmungsgemäßen Gebrauch. Aus Gründen der Übersichtlichkeit enthält es jedoch nicht sämtliche Details zu allen möglichen Funktionskombinationen. Ebenso ist es nicht möglich, jeden denkbaren Fall der Integration oder des Betriebes zu berücksichtigen.

Die Typ3 osa dient zum

- Ansteuern von Vorschubantrieben, Spindeln und Hilfsachsen einer Werkzeugmaschine über SERCOS interface, zum Führen eines Bearbeitungswerkzeuges entlang einer programmierten Bahn, zur Bearbeitung eines Werkstückes (CNC). Zusätzlich wird eine SPS mit entsprechenden E/A-Komponenten benötigt, die – in Kommunikation mit der eigentlichen CNC – den Bearbeitungsprozess an der Maschine ganzheitlich steuert und auch in sicherheitstechnischer Hinsicht überwacht.
- Programmieren von Kontur und Bearbeitungs-Technologie (Bahnvor-schub, Spindeldrehzahl, Werkzeugwechsel) eines Werkstückes.

Jeder darüber hinausgehende Gebrauch ist nicht bestimmungsgemäß!

Die beschriebenen Produkte

- wurden unter Beachtung der Sicherheitsnormen entwickelt, gefertigt, geprüft und dokumentiert. Bei Beachtung der für Projektierung, Montage und bestimmungsgemäßen Betrieb beschriebenen Handhabungsvorschriften und sicherheitstechnischen Anweisungen gehen vom Produkt im Normalfall keine Gefahren für Personen oder Sachen aus.
- erfüllen die Anforderungen
 - der EMV-Richtlinien (89/336/EWG, 93/68/EWG und 93/44/EWG)
 - der Niederspannungsrichtlinie (73/23/EWG)
 - der harmonisierten Normen EN 50081-2 und EN 50082-2
- sind für den Betrieb in industrieller Umgebung vorgesehen, d. h.
 - kein direkter Anschluss an die öffentliche Niederspannungs-Stromversorgung,
 - Anschluss über einen Transformator an das Mittel- bzw. Hochspannungsnetz.

Im Wohnbereich, in Geschäfts- und Gewerbebereichen sowie in Kleinbetrieben dürfen Klasse-A-Geräte mit folgendem Hinweis eingesetzt werden:

 **Dies ist eine Einrichtung der Klasse A. Diese Einrichtung kann im Wohnbereich Funkstörungen verursachen; in diesem Fall kann vom Betreiber verlangt werden, angemessene Maßnahmen durchzuführen und dafür aufzukommen.**

Der einwandfreie und sichere Betrieb des Produktes setzt einen sachgemäßen Transport, sachgerechte Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung voraus.

1.2 Qualifiziertes Personal

Die Anforderungen an qualifiziertes Personal richten sich nach den von ZVEI und VDMA beschriebenen Anforderungsprofilen, siehe:

Weiterbildung in der Automatisierungstechnik

Hrsg.: ZVEI und VDMA

MaschinenbauVerlag

Postfach 71 08 64

60498 Frankfurt

Dieses Handbuch wendet sich an **NC-Programmierer und NC-Projek-teure**. Sie benötigen spezielle Kenntnisse über Wirkungsweise, Syntax und Befehlsvorrat der CPL- und DIN-Programmiersprache.

Programmierung, Start und Bedienung sowie das Verändern von Programmparametern darf nur durch entsprechend geschultes Fachpersonal erfolgen! Dieses Personal muss in der Lage sein, mögliche Gefahren zu erkennen, die durch Programmierung, Programmänderungen und allgemein durch die mechanische, elektrische oder elektronische Ausrüstung verursacht werden können.

Eingriffe in die Hard- und Software unserer Produkte, die nicht in diesem Handbuch beschrieben sind, dürfen nur durch Bosch-Fachpersonal vorgenommen werden.

Bei unqualifizierten Eingriffen in die Hard- oder Software oder bei Nichtbeachten der in diesem Handbuch gegebenen oder am Produkt angebrachten Warnhinweise können schwere Personen- oder Sachschäden eintreten.

Nur Elektrofachkräfte nach IEV 826-09-01 (modifiziert), die den Inhalt dieses Handbuches kennen, dürfen die beschriebenen Produkte installieren und warten.

Dies sind Personen, die

- aufgrund ihrer fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie aufgrund ihrer Kenntnis der einschlägigen Normen die auszuführenden Arbeiten beurteilen und mögliche Gefahren erkennen können.
- aufgrund einer mehrjährigen Tätigkeit auf vergleichbarem Gebiet den gleichen Kenntnisstand wie nach einer fachlichen Ausbildung haben.

Beachten Sie diesbezüglich unser umfangreiches Schulungsangebot. Die aktuellsten Informationen, Web-Shop für Teachware und Online-Seminarbuchung finden Sie unter <http://www.bosch.de/at/didactic>. Auskünfte erteilt Ihnen auch unser Schulungszentrum, Telefon: (+49) (0 60 62) 78-258.

1.3 Sicherheitshinweise an den Produkten



Warnung vor gefährlicher elektrischer Spannung!



Warnung vor Gefahren durch Batterien!



Elektrostatisch gefährdete Bauelemente!



Warnung vor schädlicher Lichtstrahlung (LWL-Sender)



Vor dem Öffnen Netzstecker ziehen!



Schutzleiter PE



Erde allgemein

1.4 Sicherheitshinweise in diesem Handbuch



GEFÄHRLICHE ELEKTRISCHE SPANNUNG

Dieses Symbol warnt vor einer **gefährlichen elektrischen Spannung**. Durch ungenaues Befolgen oder Nichtbefolgen dieser Anweisung kann es zu **Personenschäden** kommen.




GEFAHR

Dieses Symbol wird benutzt, wenn es durch ungenaues Befolgen oder Nichtbefolgen von Anweisungen zu **Personenschäden** kommen kann.



ACHTUNG

Dieses Symbol wird benutzt, wenn es durch ungenaues Befolgen oder Nichtbefolgen von Anweisungen zu **Beschädigungen an Geräten oder Dateien** kommen kann.

 Dieses Symbol wird benutzt, wenn Sie auf etwas Besonderes aufmerksam gemacht werden sollen.

★ Dieses Zeichen zeigt an, dass eine von Ihnen auszuführende Tätigkeit beschrieben wird.

1.5 Sicherheitsanweisungen für das beschriebene Produkt

**GEFAHR**

Lebensgefahr durch unzureichende NOT-AUS-Einrichtungen!
NOT-AUS-Einrichtungen müssen in allen Betriebsarten der Anlage wirksam und erreichbar bleiben. Ein Entriegeln der NOT-AUS-Einrichtung darf keinen unkontrollierten Wiederanlauf der Anlage bewirken!

Erst NOT-AUS-Kette prüfen, dann einschalten!

**GEFAHR**

Falsche oder ungewollte Achsbewegungen!
Testen Sie neue Programme zunächst ohne Achsbewegung sorgfältig aus! Die Steuerung bietet dazu in der Gruppenbetriebsart 'Abarbeiten' die Möglichkeit, über Softkeys Achsbewegungen und/oder Hilfsfunktionsausgaben zu blockieren.

**GEFAHR**

Falsche oder ungewollte Reaktionen der Steuerung!
Bosch haftet nicht für Folgeschäden, die aus der Abarbeitung eines NC-Programmes, eines einzelnen NC-Satzes oder durch manuelles Verfahren der Achsen resultieren.

Bosch haftet ebenfalls nicht für Folgeschäden, die durch entsprechende SPS-Programmierung hätten vermieden werden können!

**GEFAHR**

Werkzeug- und Achsbewegungen!
Vorschub- und Spindel motore bringen sehr hohe mechanische Kräfte auf und können durch die große Dynamik sehr schnell beschleunigen.

- Nie bei eingeschalteter Anlage im Gefahrenbereich der Maschine sein!
 - Nie sicherheitsrelevante Anlagenfunktionen außer Kraft setzen!
 - Sofort Störungen an der Anlage Ihrer Instandhaltungs- bzw. Reparaturabteilung melden!
-

**ACHTUNG**

Beim Umgang mit Baugruppen und Bauelementen alle Vorkehrungen zum ESD-Schutz einhalten! Elektrostatische Entladungen vermeiden!

Folgende Schutzmaßnahmen für elektrostatisch gefährdete Baugruppen und Bauelemente (EGB) beachten!

- Das für die Lagerung, den Transport und die Handhabung verantwortliche Personal muss im ESD-Schutz ausgebildet sein.
- EGB müssen in den vorgeschriebenen Schutzverpackungen gelagert und transportiert werden.
- EGB dürfen grundsätzlich nur an dafür eingerichteten ESD-Arbeitsplätzen gehandhabt werden.
- Personal, Arbeitsplatten und alle Geräte und Werkzeuge, die mit EGB in Berührung kommen können, müssen auf gleichem Potential (z. B. geerdet) sein.
- Ein zugelassenes Erdungsarmband anlegen. Das Erdungsarmband muss über ein Kabel mit integriertem 1-M Ω -Widerstand mit der Arbeitsplatte verbunden sein.
- EGB dürfen auf keinen Fall mit aufladbaren Gegenständen in Berührung kommen, dazu gehören die meisten Kunststoffe.
- Beim Einsetzen von EGB in Geräte und beim Herausnehmen muss das Gerät spannungsfrei sein.

1.6 Dokumentation, Version und Warenzeichen

Dokumentation

Das vorliegende Handbuch informiert über die Bedienung des CPL-Debuggers.

Übersicht der Gesamtdokumentation	Bestell-Nr.	
	Deutsch	Englisch
Anschlussbedingungen zur Projektierung und Wartung	1070 073 704	1070 073 736
Software-Installation	1070 073 796	1070 073 797
Funktionsbeschreibung	1070 073 870	–
MACODA Bedienung und Konfiguration der Maschinenparameter	1070 073 705	1070 073 742
Bedienungsanleitung Standard-Bedienoberfläche	1070 073 726	1070 073 739
Bedienungsanleitung Diagnose-Tools	1070 073 779	1070 073 780
Fehlermeldungen	1070 073 798	1070 073 799
SPS-Projektierung Software-Schnittstellen der integrierten SPS	1070 073 728	1070 073 741
ICL700 Systembeschreibung Programmstruktur der integrierten SPS ICL700	1070 073 706	1070 073 737
DIN Programmieranleitung zur Programmierung nach DIN 66025	1070 073 725	1070 073 738
CPL Programmieranleitung	1070 073 727	1070 073 740
CPL-Debugger Bedienungsanleitung	1070 073 872	–
Parametrierung Werkzeugverwaltung	1070 073 782	1070 073 793
Software-SPS Entwicklungsumgebung für Windows NT	1070 073 783	1070 073 792
Messzyklen für schaltende Messtaster	1070 073 788	1070 073 789
Universal-Fräszyklen	–	1070 073 795

 **In diesem Handbuch ist das Diskettenlaufwerk immer das Laufwerk A: und die Festplatte immer das Laufwerk C:.**

Spezielle Tasten oder Tastenkombinationen werden mit spitzer Klammer dargestellt:

- Spezielle Tasten: z.B. <enter>, <pgup>,
- Tastenkombination (gleichzeitiges Drücken): z.B. <ctrl> + <pgup>

Version

 **Dieses Handbuch gilt für folgende Versionen:**

Software: Typ3 osa V6.x
CPL-Debugger V1.5

Angaben über aktuelle Versionsnummern der einzelnen Software-Module finden Sie in der Gruppenbetriebsart 'Diagnose' unter Softkey 'Steuerungs-Diagnose'.

Angaben über die Software-Version für Windows 95 oder Windows NT erhalten Sie so:

1. Klick mit rechter Maustaste auf das Icon 'My Computer' ('Arbeitsplatz') auf dem Desktop
2. Menüpunkt 'Properties' ('Eigenschaften') auswählen

Warenzeichen

Alle Warenzeichen der Software, die mit der Auslieferung auf Bosch-Produkten installiert ist, sind Eigentum der entsprechenden Hersteller.

Bei Auslieferung besteht für jede installierte Software Copyright. Sie darf nur mit Einverständnis von Bosch beziehungsweise entsprechend den Lizenzvereinbarungen des jeweiligen Herstellers vervielfältigt werden.

MS-DOS® und Windows™ sind eingetragene Warenzeichen der Microsoft Corp.

PROFIBUS® ist ein eingetragenes Warenzeichen der PROFIBUS Nutzerorganisation e. V.

SERCOS interface™ ist ein eingetragenes Warenzeichen der Interessengemeinschaft SERCOS interface e. V.

2 Übersicht

Mit dem CPL-Debugger können NC-Programme getestet und eventuell vorhandene Fehler in der Programmierung komfortabel gefunden werden.

Eigenschaften des CPL-Debuggers:

- anwendbar auf Programme im CPL-Format (Customer Programming Language) und DIN 66025-Format
- berücksichtigt Unterprogramm-Programmierung
- beinhaltet Editor-Funktionen für ausführbare Programme
- unterstützt alle CPL-Variablenformate
- unterstützt mehrdimensionale CPL-Variablen (Arrays)
- unterstützt lokale, globale und permanente Variable
- Breakpoints (Haltepunkte) setzen (max. 20 gleichzeitig aktiv)
- Variablenwerte zur Programmlaufzeit verändern
- Automatisches Beobachten (Watch) von Variablen.

Der CPL-Debugger läuft unter Windows NT auf einem Bosch Bedienfeld oder einem vergleichbaren PC.

Fehlersuche im Programm

- Wo im Programm trat der Fehler zum ersten Mal auf?
- Welche Auswirkungen hat der Fehler auf das Programm?
- Wie sind die Auswirkungen auf andere Variablen bzw. Programmteile?
- Welche Wichtigkeit hat der Fehler für das Programm?

Der CPL-Debugger bietet hier die Möglichkeit, systematisch alle relevanten Variablen auszuwählen und deren Inhalte während der Laufzeit eines Programms Schritt für Schritt zu verfolgen.

Durch Setzen von Breakpoints kann das Programm während der Laufzeit an benutzerdefinierten Haltepunkten angehalten werden, um dort die Werte einzelner Variablen zu erfragen und um beobachtete Variablen auf ihre Inhalte hin zu prüfen.

An einem Breakpoint kann jede Variable mit anderen Werten belegt werden. Dadurch kann ein anderes Ablaufszenario im Programm getestet werden, bei fehlerhafter Zuweisung einer Variablen durch das Programm kann diese versuchsweise mit dem "richtigen" Wert versehen werden.

Die Schachtelungsstruktur von Haupt-/und Unterprogrammen lässt sich im Fenster "Call Stack" nachvollziehen und beobachten.

Anhand benutzerspezifischer Meldungen (MSG, GMSG,..) im Fenster "Output" kann man den Programmablauf testen und überprüfen.

Der CPL-Debugger arbeitet:

- bei aktiver Gesamtsperre, d.h. ohne Achsbewegungen, oder
- mit Achsbewegungen bei eingeschränkter Funktionalität.



Der CPL-Debugger kann auch in Verbindung mit der "Typ3 osa Emu" (Steuerungs-Emulation) eingesetzt werden. In diesem Fall wird automatisch die Gesamtsperre als aktiviert angenommen.

Notizen:

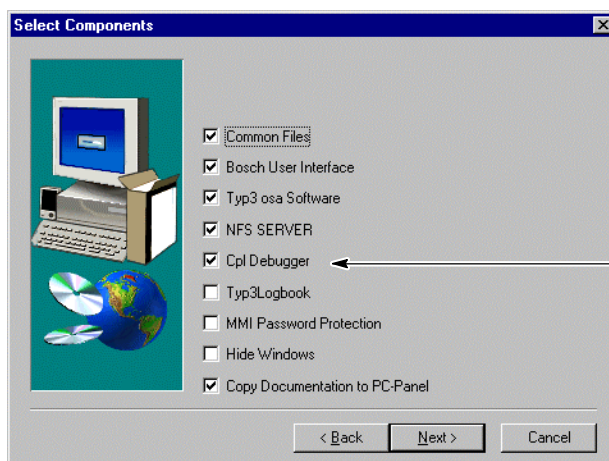
3 Installation und Start

Der CPL-Debugger ist eine optionale PC-Software, die auf der Datenträger-CD der aktuellen Steuerungssoftware enthalten ist.

3.1 CPL-Debugger installieren

Beim Installieren der Typ3 osa-Software auf einem über Ethernet angeschlossenen PC-Bedienfeld oder einem PC wird der CPL-Debugger mitinstalliert. Er kann auf diesem Weg auch nachinstalliert werden.

- ★ Wählen Sie *Cpl Debugger* im Customer Setup:



CPL-Debugger
explizit
markieren

- ☞ Weitere Hinweise zur Installation entnehmen Sie bitte dem Handbuch "Typ3 osa Software Installation".

3.2 CPL-Debugger starten

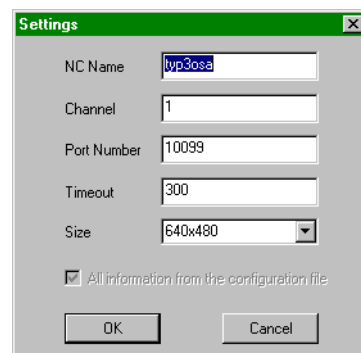
Der CPL-Debugger kann in Verbindung mit einer Konfigurationsdatei *.cld gestartet werden. Beim ersten Öffnen ist es die Konfigurationsdatei "Default.cld", im weiteren Verlauf können zusätzliche Konfigurationsdateien erzeugt (siehe Abschnitt 3.3) und beim Starten verwendet werden.

Sie starten den CPL-Debugger mit einer der folgenden Alternativen:



- Durch Doppelklicken auf das Desktop-Symbol CPL Debugger
Hier wird die Konfigurationsdatei "Default.cld" verwendet, wenn nicht vorher im Eigenschaftfenster der Verknüpfungspfad auf eine andere Konfigurationsdatei geändert wird. Mögliche Aufrufparameter siehe Seite 3-3)
- Durch Doppelklicken auf eine beliebige Anwender-Konfigurationsdatei (*.cld)
- Auf dem Windows Desktop mit *Start* ► *Ausführen*
 - Geben Sie im Fenster "Ausführen" eine beliebige Konfigurationsdatei (*.cld) an, oder
 - Suchen Sie die Datei "cpldebug.exe" im Verzeichnis TYP3PCP\BIN. In diesem Fall können nach einem Leerzeichen weitere Aufrufparameter angehängt werden (siehe Seite 3-3).

In jedem Fall erscheint das Dialogfenster "Settings", in dem Sie die erforderlichen Angaben für *NC Name* und *Channel* eintragen, optional unter *Size* die gewünschte Fenstergröße:

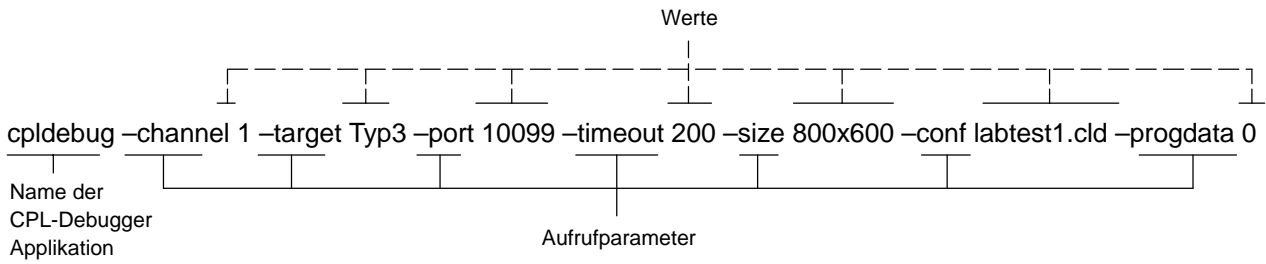


Zum Navigieren im Dialogfenster benutzen Sie <Tab> oder <Shift+Tab>.

Mit der Option *All information from the configuration file* werden auch die in der angewählten Konfigurationsdatei gespeicherten Breakpoints und die zur Beobachtung ausgewählten Variablen geladen, sofern sie zum angewählten Programm passen.

Kommt keine Verbindung zustande, wird der Vorgang nach Ablauf der Timeout-Zeit unterbrochen und eine Fehlermeldung ausgegeben.

Aufrufparameter für .exe-Datei



Die Parameter sind jeweils durch ein Leerzeichen und einen Bindestrich voneinander getrennt.

Notwendige Parameter:

Aufrufparameter	Bedeutung	Wertebereich/ Einheit	Voreinstellung
channel	Kanalnummer	1..12	1
target	Name der Typ3 osa-Steuerung im Netzwerk	siehe Host TCP/IP-Adresse	typ3osa

Optionale Parameter:

Aufrufparameter	Bedeutung	Wertebereich/ Einheit	Voreinstellung
port	TCP/IP Port-Nummer	10099	10099
timeout	Zeitspanne, innerhalb der die Verbindung zwischen CPL-Debugger und NC hergestellt sein muss.	> 0 sec	300
size	Größe des CPL-Debugger Fensters	640x480 oder 800x600 Pixel	640x480
conf	Name der verwendeten Konfigurationsdatei	.cld	default.cld
progdata	Startkonfiguration: 1: mit Breakpoints und zur Beobachtung ausgewählter Variablen 0: ohne Breakpoints und zur Beobachtung ausgewählter Variablen.	0 oder 1	1

3.3 Konfigurationsdatei

Die Konfigurationsdatei enthält alle relevante Einstellungsdaten einer CPL-Debugger-Sitzung:

- Name des aktiven CPL- oder DIN-Programms
- Einstellungen des CPL-Debugger-Fensters (Größe, Breite der Spalten)
- aktive Breakpoints in Hauptprogramm und Unterprogrammen (nur aktive Breakpoints können gespeichert werden)
- Liste der beobachteten Variablen

3.3.1 Konfigurationsdatei erstellen

... beim Beenden der Sitzung

Beim Verlassen des CPL-Debugger über den Menüpunkt *File* ► *Exit* werden alle Änderungen der aktuellen Sitzung, insbesondere auch Breakpoints und beobachtete Variablen, in der Konfigurationsdatei gespeichert, mit der der CPL-Debugger gestartet wurde.

Beim ersten Öffnen des CPL-Debugger ist dies die Datei "Default.cld".

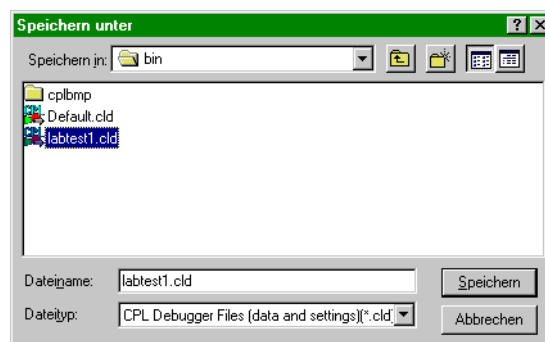
- ☞ **Speichern Sie die Änderungen der Sitzung vorzugsweise in einer neuen Konfigurationsdatei. So können Sie über "Default.cld" jederzeit auf die Grundeinstellung des Debugger zurückgreifen.**

... während der Sitzung

Crtl + F4

Save
Workspace

- ★ Wählen Sie den Menüpunkt *File* ► *Save Workspace* –oder– im Fenster "Program" den Softkey *Save Workspace*
- ★ Wählen Sie im Dialogfenster "Speichern unter" ein beliebiges Verzeichnis und geben Sie einen neuen Dateinamen ein (z.B. labtest1.cld).



Mit *Speichern* wird aus der aktuellen CPL-Debugger-Sitzung eine neue Konfigurationsdatei mit aktiven Breakpoints und beobachteten Variablen erstellt. So können auch mehrere Zustände einer Sitzung gespeichert werden.

3.3.2 Konfigurationsdatei laden

Enthält die Konfigurationsdatei beobachtete Variablen, werden diese geladen, wenn:

- das in der NC angewählte Programm mit dem in der Konfigurationsdatei abgespeicherten Programm-Namen übereinstimmt und
- die Variablen noch im Programm enthalten sind.

Enthält die Konfigurationsdatei Breakpoints, werden diese geladen, wenn:

- das in der NC angewählte Programm mit dem in der Konfigurationsdatei abgespeicherten Programm-Namen übereinstimmt und
- das Programm zwischenzeitlich nicht editiert wurde.
- der Debug-Mode aktiviert ist (siehe Kap. 5.2).

... beim Starten

Wie in Abschnitt 3.2 beschrieben, kann beim Starten jede Konfigurationsdatei verwendet werden, am einfachsten durch Doppelklicken auf die gewünschte Datei.

... während der Sitzung

Bei laufender Sitzung kann jederzeit eine vorhandene Konfigurationsdatei aufgerufen und deren Einstellungen übernommen werden, wenn das NC-Programm, auf das sich die Konfigurationsdatei bezieht, geladen ist. Andernfalls erscheint die Meldung, dass alle in der Konfigurationsdatei gespeicherten Breakpoints nicht geladen werden.



ACHTUNG

Datenverlust!

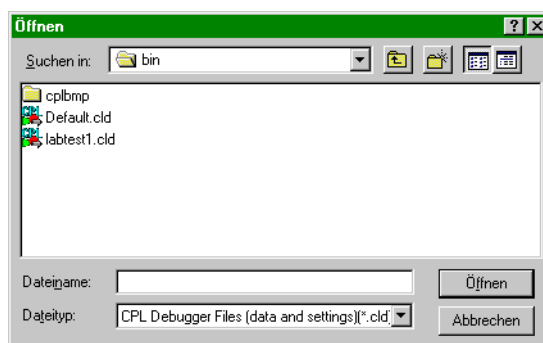
Beim Laden einer anderen Konfigurationsdatei gehen bisher vorgenommene Änderungen im CPL-Debugger verloren.

Sie können vorher mit *Save Workspace* gespeichert werden.

Ctrl + F4

Load
Workspace

- ★ Wählen Sie den Menüpunkt *File* ► *Load Workspace* –oder– im Fenster "Program" den Softkey *Load Workspace*.
- ★ Wählen Sie im Dialogfenster "Öffnen" eine Konfigurationsdatei aus (z.B. labtest1.cld).



- ☞ **Alle im weiteren Verlauf vorgenommenen Änderungen werden über den Menüpunkt *File* ► *Exit* in der zuvor geladenen Datei gespeichert.**

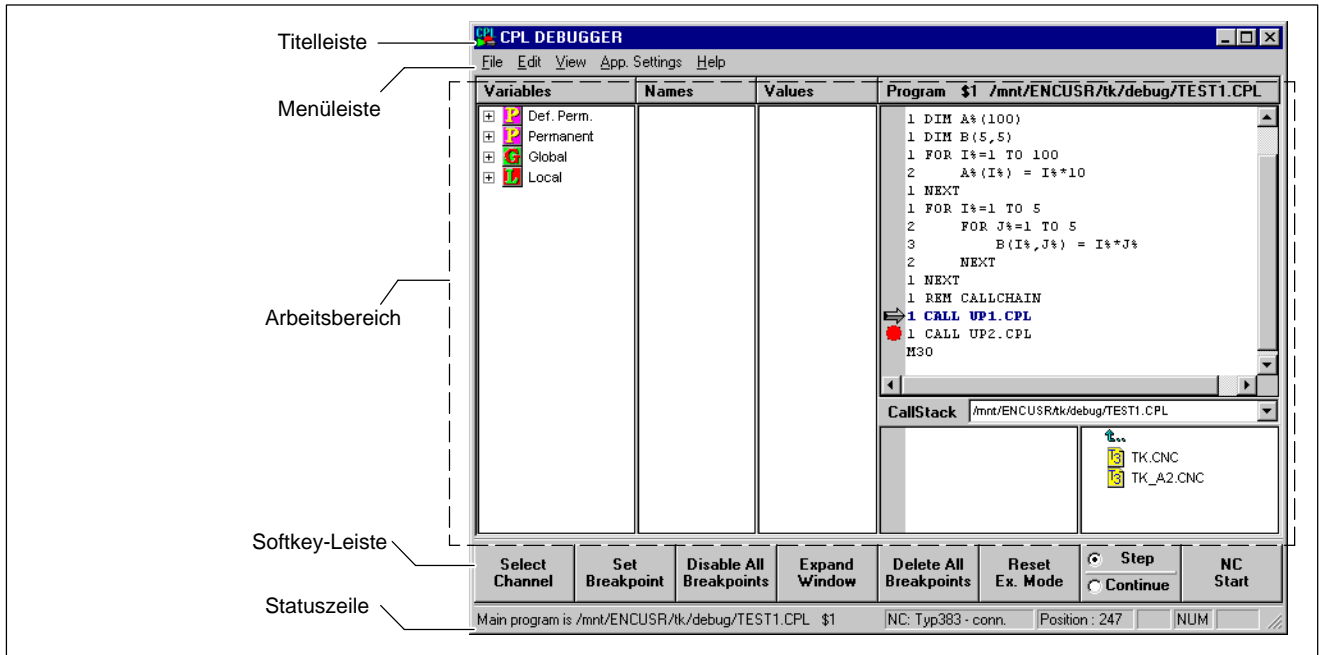
Notizen:

4 Bedienoberfläche

4.1 Übersicht

Das CPL-Debugger-Fenster enthält folgende unterschiedliche Bereiche:

- Titelleiste, Menüleiste und Statuszeile
- Arbeitsbereich, zum Visualisieren der Programmdaten und eventueller Eingaben
- Softkey-Leiste, abweichend vom MS-Windows-Standard



Menüleiste

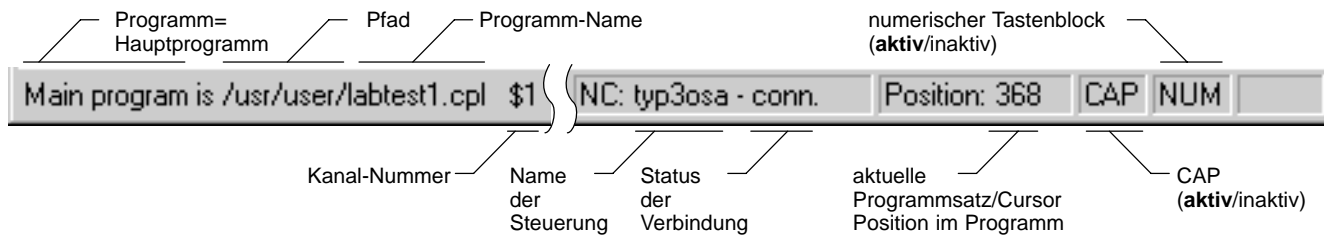
Die Hauptmenüs erreichen Sie mit der Maus oder mit <Alt>. Die Pfeiltasten steuern die Untermenüs an.

Menü	Untermenü	Bedeutung
File	– Save Workspace – Load Workspace – Exit	Dateioperationen (Öffnen u. Speichern der Konfigurationsdatei); Beenden der CPL-Debugger Applikation
Edit	– Find	Text suchen
View	– Breakpoints..	Breakpoint-Liste ansehen; Breakpoints setzen, löschen, aktivieren/deaktivieren
App. Settings	– General Inhibit	Freigeben des CPL-Debuggers ohne Gesamtsperre, d.h. mit bewegten Achsen und eingeschränkter Funktionalität
Help	–	Hilfe anzeigen

Statuszeile

Die Statuszeile unterhalb der Softkeys enthält folgende Informationen:

- Name des Hauptprogramms mit Angabe des Pfads und des Kanals
- Netzwerk-Name der Steuerung und Status, ob eine Verbindung besteht (connected – not connected)
- Aktiver Programmsatz
- Anzeige CAP (Umschaltung Groß-oder Kleinschreibung)
- Anzeige NUM (numerisches Tastenfeld aktiv/inaktiv)



4.2 Arbeitsbereich

Im Arbeitsbereich stellt der CPL-Debugger die Programmdaten in verschiedenen Fenstern dar.

Die einzelnen Fenster des Arbeitsbereiches werden über Maus oder Softkeys aktiviert (siehe Seite 4-4). Durch die unterschiedlichen Funktionen der Fenster ändert sich auch die Softkey-Leiste unterhalb des Arbeitsbereiches.

Die Fensterbreite ist änderbar:
 – durch Mausbedienung
 – oder laden einer .cld-Datei (Load Workspace)
 Rücksetzen der Fensterbreite: <Ctrl>+<F2>, dann Softkey "Reset Properties" (F4) drücken

Nr.	Fenster im CPL-Debugger	Funktion	mögliche Eingaben
1	Variables	Variablen anzeigen: Auflistung aller im Programm verwendeten Variablen-Namen, nach folgenden Gruppen gegliedert: <ul style="list-style-type: none"> • Definierbar Permanent • Permanent • Global • Lokal. 	Zu beobachtende Variablen auswählen. Sie werden im Fenster "Names" angezeigt.
2	Names	Variablen beobachten: Auflistung der ausgewählten Variablen. Die aktuellen Werte werden im Fenster "Values" angezeigt.	Löschen der Variablen aus der Beobachtungsliste, die nicht mehr beobachtet werden sollen.
3	Values	Werte anzeigen: Anzeige der Variablen-Werte aus dem Fenster "Names" zur Laufzeit des Programms.	Editieren der Werte von Variablen (außer einfachen String-Variablen) zur Laufzeit des Programms.
4	Program	Programmfenster: Angezeigt werden: <ul style="list-style-type: none"> • Programmname mit Kanal (\$x) und Pfad-angabe • Programmcode • der zur Laufzeit abgearbeitete Satz • Breakpoints. 	<ul style="list-style-type: none"> • Text suchen • Breakpoints setzen, löschen, aktivieren, deaktivieren.

Nr.	Fenster im CPL-Debugger	Funktion	mögliche Eingaben
5	Call Stack	Unterprogramm anwählen: Zeigt die Unterprogramm-Schachtelung des angewählten Programms.	Auswahl eines Unterprogramms aus der gerade aktiven Unterprogrammstruktur.
6	File	Dateisystem: Anzeige des Dateisystems der Steuerung.	Auswahl eines Programms, das im Fenster "Program" angezeigt werden soll. (Dies ist noch keine Programmanwahl in der Steuerung.)
7	Output	Statusanzeige: Anzeige von Meldungen und Fehlern zum Programm und zum CPL-Debugger.	Löschen der Anzeigen. Genauere Informationen zu den angezeigten Meldungen und Fehlern finden sich im Info-dialog der Bedienoberfläche der Steuerung.

4.3 Softkey-Leiste

Unterhalb des Arbeitsbereiches sind 8 Softkeys angeordnet, die per Maus oder Funktionstasten F1...F8 bedient werden können:

- Aktive Tasten sind mit schwarzer Beschriftung dargestellt
- Inaktive Tasten, z.B. wegen einer gesetzten Option, sind gedimmt dargestellt.

Abhängig vom aktiven Fenster im Arbeitsbereich ändert sich die Belegung der Softkey-Leiste (siehe folgende Abbildung).

Ein Fenster wird ausgewählt durch:

- anklicken mit der Maus
- mit der Taste <Tab> oder <Shift>+<Tab> Fenster nacheinander durchlaufen oder
- durch Festhalten der <Ctrl>-Taste die Softkey-Leiste auf "Fensterauswahl" umschalten, dann mit Maus oder Funktionstaste (F1...F8) Fenster direkt anwählen (siehe Abbildung).

<F9> macht die letzte Aktion rückgängig (entspricht <Esc>).

Innerhalb der Fenster wird mit Pfeiltasten navigiert:

Pfeil-Taste \blacktriangleright öffnet, \blacktriangleleft schließt.

1 Variablen anzeigen
 2 Variablen beobachten
 3 Werte anzeigen
 4 Programm anzeigen
 5 Anwahl Unterprogramm
 6 Dateisystem (Typ3 osa)
 7 Statusanzeige (Meldungen,..)

Belegung der Softkey-Leiste bei Anwahl:

	<F1>	<F2>	<F3>	<F4>	<F5>	<F6>	<F7>	<F8>
1 Fenster "Variables"	Select Channel	Expand Branch	Collapse Branch	Expand Window	Quick Watch	Write Watch	Read Watch	Select NC
2 Fenster "Names"	Select Channel	Expand Branch	Collapse Branch	Expand Window	Delete Watch Variable	Delete All Watch Variables		Select NC
3 Fenster "Values"	Select Channel	Expand Branch	Collapse Branch	Expand Window	Edit Variable	Set Variable		Select NC
4 Fenster "Program"	Select Channel	Set Breakpoint		Expand Window	Delete All Breakpoints	Set Debug Mode	Step / Continue	NC Start
oder	Select Channel	Set Breakpoint		Expand Window	Delete All Breakpoints	Reset Ex. Mode	Step / Continue	NC Start
5 Fenster "Call Stack"	Select Channel			Expand Callstack	Collapse Callstack		Go To Call	Select NC
6 Fenster "Output"	Select Channel	Clear Output	Load Workspace	Save Workspace				Select NC
7 Fenster "File"	Select Channel	Sub Directory (+)	Parent Directory (-)	Reset Properties	Update Files	General Inhibit ON	Show File	Select NC
mit <Ctrl>-Taste	Program Window	File Window	Call Stack Window	Output Window	PH. Act / Ph. Prep	Variables Window	Var Name Window	Var Value Window

4.4 Symbole (Images)

Symbole im Fenster "Variables"

Die CPL-Sprache kennt lokale, globale, permanente und definierbare permanente Variablen.

Die verschiedenen Variablen des aktiven Programms werden in Gruppen zusammengefasst:

- Permanente Variablen
- Definierbare permanente Variablen
- Globale Variablen
- Lokale Variablen

Innerhalb der Gruppen werden die einzelnen Variablen nach Namen sortiert aufgelistet und mit einem Symbol mit kleinem Buchstaben gekennzeichnet.

Symbole im Fenster "Names"

Bei der Übernahme einer Variablen in das Fenster "Names" erhält diese Variable ein neues Symbol entsprechend der Beobachtungsvariante, mit der die Variable beobachtet werden soll. Bis zu 3 Beobachtungsvarianten können für eine Variable gleichzeitig festgelegt werden:

- Quick Watch
- Write Watch
- Read Watch

Beispiel: mit Quick Watch zu beobachtende Programmvariable "local1%"

- local1%

Symbole im Fenster "Program"

Der linke Rand des Fensters "Program" kann folgende Symbole enthalten:

Breakpoint-Symbole

- gesetzt, aber nicht aktiv (weiß),
- gesetzt und aktiv (rot)

Programmsatz-Symbole

- Markierung des Programmsatzes, der gerade abgearbeitet wird. Gleichzeitig wird die aktive Programmzeile farblich hervorgehoben. Nach Abarbeiten des Satzes wandert der Pfeil zur nächsten Programmzeile oder zur nächstmöglichen Programmzeile eines Unterprogramms.

```
16 DIM #D(30)
⇒ 17 #STRING1$ = "HREN"
18 #D(1) = 2
```

- Markierung des Programmsatzes mit einem Unterprogramm-Aufruf.

Symbole im Fenster "File"



Verzeichnis-Symbole

Die Verzeichnissymbole zeigen die vorhandenen Typ3 Verzeichnisse der angewählten Steuerung.

Symbol eines Verzeichnisses (z.B. usrfep)

Symbol "untergeordnetes Verzeichnis anwählen"

Symbol "übergeordnetes Verzeichnis anwählen"

Dateiattribut-Symbole

Jeder aufgelisteten Datei wird ein Symbol vorangestellt, das Auskunft über die gesetzten Eigenschaften der Datei gibt:



- rwx: lesbar, überschreibbar, ausführbar in der NC (gelb)
- rx: lesbar, ausführbar in der NC (grün)
- wx: überschreibbar, ausführbar in der NC (grau)
- x: ausführbar in der NC (grau/grün)

Die Dateieigenschaften können über die NC-Bedienoberfläche im Bedienbereich *VERWALTEN* ► *Datei Operationen* ► *Zugriffsrechte* eingestellt werden (siehe Bedienungsanleitung Standard-Bedienoberfläche).

☞ **Gegebenenfalls müssen Sie sich die Rechte zum Ändern der Dateieigenschaften beim System-Administrator zuweisen lassen.**

Symbole im Fenster "Output"



- Fehler (rot/blau)



- Warnungen (rot/blau)



- Hinweise (PRN#(0,...))

- MSG (grau) im NC Programm programmierte Meldungen



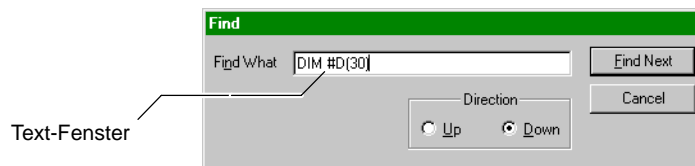
- GMSG (grün)

Bei Fehlern und Warnungen wird durch die Hintergrundfarbe der Symbole unterschieden, ob die Meldung:

- gerade ansteht —> Hintergrundfarbe: **rot**
- gelöscht ist bzw. bereits gelesen ist —> Hintergrundfarbe: **blau**

4.5 Text suchen

Mit *Edit* ► *Find..* kann eine alphanumerische Zeichenfolge im Fenster "Program" gesucht werden.



4.6 Fenstereigenschaften zurücksetzen

Die per Maus änderbare und ggf. in einer .cld-Datei abgespeicherte Fensterbreite kann jederzeit auf den fest hinterlegten Default-Wert zurückgesetzt werden:

Ctrl + F2

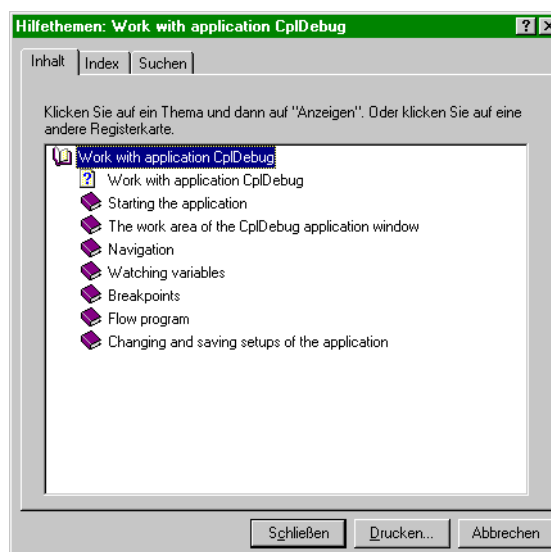


- ★ wählen Sie das Fenster "File" an und drücken den Softkey *Reset Properties*.

☞ **Das Zurücksetzen der Fenstereigenschaften mit *Reset Properties* hat keinen Einfluss auf Breakpoints oder ausgewählte Variablen.**

4.7 Hilfe

Mit *Help* ► *Topics* rufen Sie die Online-Hilfe des CPL-Debuggers auf. Über Suchfunktionen können ausführliche Informationen gefunden werden.



5 Arbeitsweise CPL-Debugger

5.1 Voraussetzungen

Die folgende Übersicht zeigt die Vorgehensweise zum Arbeiten mit dem CPL-Debugger:

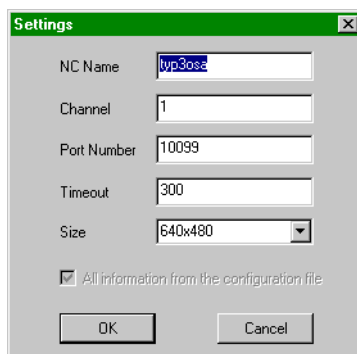
	Vorgehensweise	siehe Seite
1.	Steuerung und Kanal auswählen	5-2
2.	<ul style="list-style-type: none"> • Gesamtsperre setzen (in der NC-Bedienoberfläche im Bedienbereich "ABARBEITEN" oder im CPL-Debugger) oder • "Ignore General Inhibit" aktivieren (Arbeiten mit bewegten Achsen) 	5-3 5-2
3.	NC-Programm anwählen	5-5
4.	CPL-Debugger starten	3-2
5.	Konfigurationsdaten laden (optional)	3-5
6.	Navigation zum fehlerhaften Programmteil oder ggf. zum Unterprogramm im CPL-Debugger	5-6
7.	Breakpoints an vermuteten Fehlerstellen setzen	5-12
8.	Variablen zur Beobachtung auswählen	5-15
9.	NC-Programm starten	5-6
10.	Werte der ausgewählten Variablen beobachten und Fehleranalyse	5-15
11.	Fehlerbereinigung der betroffenen Programmteile im Editor der Steuerung	
12.	Testen des Programms (ab Pkt.6), ob Fehler beseitigt ist und ob weitere Fehler vorhanden sind	
13.	Konfiguration (Einstellungen) für evtl. weitere CPL-Debugger-Sitzungen sichern	3-4

5.1.1 Steuerung und Kanal auswählen



- ★ Wählen Sie den Softkey *Select NC*

Tragen Sie im Dialogfenster "Settings" den Namen der Steuerung (NC Name) und die Kanalnummer (Channel) ein. Die übrigen Voreinstellungen können Sie übernehmen. Verlassen Sie den Dialog mit *OK*.



- ★ Wollen Sie nur den Kanal ändern, wählen Sie den Softkey *Select Channel*.

Tragen Sie im Dialogfenster "Settings" die Kanalnummer der Steuerung (Channel) ein. Verlassen Sie den Dialog mit *OK*.

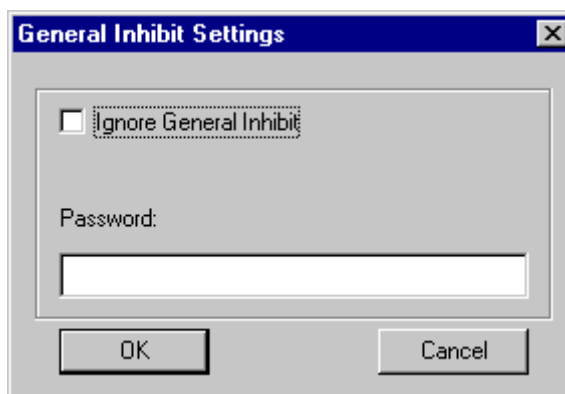
5.1.2 Gesamtsperre

In seiner Default-Einstellung arbeitet der CPL-Debugger nur dann, wenn in der Steuerung die Gesamtsperre gesetzt ist, so dass sich die Achsen nicht bewegen.

Wenn es jedoch notwendig ist, das Programm mit verfahrenenden Achsen zu untersuchen, kann über ein Passwort die Voraussetzung "Gesamtsperre" abgeschaltet werden.

Arbeiten ohne Gesamtsperre

Für die Freigabe zum Arbeiten ohne Gesamtsperre aktivieren Sie im Fenster "General Inhibit Settings" die Checkbox *Ignore General Inhibit*.



OK ist nur nach Eingeben des zugehörigen Passwortes möglich.

Nach dieser Freigabe kann der Debug-Mode auch ohne Gesamtsperre aktiviert werden.



GEFAHR

Personengefährdung durch bewegte Achsen!

Während der Programmprüfung können Achsen unvorhergesehen verfahren und Sach- oder Personenschäden verursachen.

Sichern Sie den Arbeitsbereich der Maschine ab. Prüfen Sie Programme möglichst mit Gesamtsperre.

- Haltepunkte können gesetzt werden. Das Programm kann im Einzelschritt abgearbeitet werden.
- Starten des Programms nur mit der START-Taste an der Maschine. Der Softkey *NC Start* ist ohne Funktion.

Einschränkungen

- Im Einzelschrittbetrieb (Step) existiert immer nur eine Verfahrbewegung.
- Eine aktive **Fräserradiuskorrektur** führt zu Fehlermeldungen der Satzverarbeitung, da durch den Einzelschrittbetrieb oder bei Erreichen eines Haltepunktes die notwendigen Geometrieinformationen nicht komplett vorliegen.
- Geänderte **Variablen** werden ggf. erst verzögert wirksam, da die Steuerung "vorbereitete" Verfahrinformationen bereits zwischenspeichert, bevor der Interpolator die Bewegungen ausführt (bis zu 30 Verfahrbewegungen).

Beispiel:

In vielen CPL-Programmen wird der anzufahrende Endpunkt berechnet und dann an die Achsen übergeben:

```
N10 G1 X[ACHSE] F100
```

Die Achse X wird an die Position gefahren, die durch den Inhalt der Variablen A festgelegt ist. Der Inhalt von A wird also vom CPL-Interpreter bestimmt. Die daraus ermittelte Endposition wird in die Reihe der abzuarbeitenden Verfahrbewegungen eingereiht und eine nachträgliche Änderung ist nach diesem Zeitpunkt nicht mehr möglich.

Deshalb kann in diesem Fall eine Änderung der Variablen A mit dem CPL-Debugger erst nach dem Abarbeiten der "vorbereiteten" Verfahrbewegungen wirksam werden.

Arbeiten mit Gesamtsperre (Default)

Wenn in der Steuerung ein Programm angewählt ist, führen Sie zunächst "Kanal grundstellen" aus. Danach kann die Gesamtsperre gesetzt werden.



Sind zur gleichen Zeit noch Programme in anderen Kanälen der ausgeschuchten Steuerung aktiv, kann Gesamtsperre nicht gesetzt werden. Eine entsprechende Meldung erscheint in der Steuerung und im CPL-Debugger.

Beenden Sie zunächst alle laufenden Programme. Danach kann die Gesamtsperre aktiviert werden.



- ★ Wählen Sie in der **Standard-Bedienoberfläche** im Bedienbereich "Abarbeiten" die Softkeys *Bearbeitung vorbereiten* ► *Gesamt-sperre*

\$1: <input type="text"/>		11.11.00	
NC: Typ3osa <input type="text"/>		12:15:22	
A: Abarbeiten/Vorbereitung			
Werkstückposition	Restweg	Datei:	
X 0.000	X 0.000 *		
Y 0.000	Y 0.000 *		
Z 0.000	Z 0.000 *		
B 0.0000	B 0.0000 *		
UA 0.000	UA 0.000 #		
VA 0.000	VA 0.000 #		
WA 0.0000	WA 0.0000 #		
Korrekturen + Technologie		Programm	
V: /database/V1		Verzeichnis:	
G53 G153 G253		Angew. Programm:	
K: /database/K4		Unterprogramm:	
D0: 0.000 H0: 0.000		Ausführung: Folgesatz	
		aktiver Satz:	
MSG			
F: 0.0	<input type="text"/>	0%	
S1: 0.0	<input type="text"/>	0%	
S2: 0.0	<input type="text"/>	0%	
Satz suchen	<input type="checkbox"/> Satz überlesen	<input checked="" type="radio"/> Auto	<input checked="" type="radio"/> Einzel-satz
		<input type="radio"/> Manuell	<input type="radio"/> Einzel-schritt
			<input type="radio"/> Progr.-satz
			<input type="checkbox"/> Wahlweise Halt
			<input checked="" type="checkbox"/> Gesamt-sperre

Ctrl + F2

 General
Inhibit
OFF

- ★ Oder wählen Sie im **CPL-Debugger** im Fenster "File" den Softkey *General Inhibit OFF*.

Nach dem Drücken des Toggle-Softkeys wechselt die Beschriftung nach *General Inhibit ON* und zeigt an, dass die Gesamtsperre nun aktiv ist.

Program \$1 Ausr/User/labtest1.opl	
<pre> 5 L JUST 11 DIM #STRING1\$(5) 12 DIM #DBLDIM(2,3) 13 #DBLDIM(2,1)=1 14 DIM #INTVAR%(4) 15 #INTVAR1%= 9 16 DIM #D(30) 17 #STRING1\$ = "HREN" 18 #D(1) = 2 19 @3 = @2 + @3 + 1 20 #INTVAR1% = #INTVAR1% + 5 25 #INTVAR2% = #INTVAR1% 26 LOCVAR1% = #INTVAR1% 28 PRN#(0,"LOCVAR1% = ",LOCVAR1%,"") N29 (MSG, labtest1.cpl) N5 G63 </pre>	
Call Stack	Ausr/User/labtest1.opl
	<ul style="list-style-type: none"> KG_A42 KG_A42_Kanal2 labtest1.cpl labtest2.cpl
Update Files	General Inhibit ON
Show File	Select NC
NC: typ3osa - conn. NUM	

Gesamt-sperre deaktivieren



- ★ Wählen Sie in der **Standard-Bedienoberfläche** im Bedienbereich "Abarbeiten" die Softkeys *Bearbeitung vorbereiten* ► *Gesamt-sperre*

Ctrl + F2

 General
Inhibit
ON

- ★ Oder wählen Sie im **CPL-Debugger** im Fenster "File" den Softkey *General Inhibit ON*.

5.1.3 Programm anwählen

- ★ Wählen Sie ein Programm aus über die Standard-Bedienoberfläche der Steuerung im Bedienbereich "Abarbeiten", Softkey *Programm auswählen*

\$1: 11.11.00
 NC: Typ3osa 12:15:22

A: Abarbeiten

Werkstückposition	Restweg	Datei: /usr/user/Teil-1234
X 0.000	X 0.000 *	N10 G0 X0 Y0 Z0 M3 S1000
Y 0.000	Y 0.000 *	> N20 G1 X40 F2000
Z 0.000	Z 0.000 *	N30 X40 Z 10.22
B 0.0000	B 0.0000 *	N40 G3 X21 Y5 R100
UA 0.000	UA 0.000 #	...
VA 0.000	VA 0.000 #	
WA 0.0000	WA 0.0000 #	

Korrekturen + Technologie

V: /database/V1
 G53 G153 G253

K: /database/K4
 D0: 0.000 H0: 0.000
 G145 D: 0.000 H: 0.000
 G147 H: 0.000

F: 0.0 80 %
 S1: 0.0 90 %
 S2: 0.0 0 %

Programm

Gruppe: Benutzer
 Angew. Programm: Teil-1234
 Unterprogramm:
 Ausführung: Folgesatz
 aktiver Satz: N20

MSG

Auto
 Manuell

Programm auswählen

- Es kann immer nur ein Programm bearbeitet werden.
- Um ein Programm bearbeiten zu können, müssen das **Dateiattribut** des Programms entsprechend gesetzt sein.
Im Fenster "File" ist jeder Datei ein Symbol für das Dateiattribut vorangestellt (siehe Seite 4–7).
- Das im CPL-Debugger angezeigte Programm kann dort nur bearbeitet werden, wenn es in der angewählten Steuerung und im eingestellten Kanal aktiv ist.
- Wird das angewählte Programm in der Steuerung editiert, sind alle Änderungen auch im CPL-Debugger sofort wirksam (incl. beobachteter Variablen und Unterprogrammen).



ACHTUNG Datenverlust!

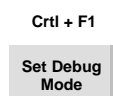
- Mit dem Anwählen eines anderen Programms zum Abarbeiten wird es gleichzeitig auch in den CPL-Debugger geladen. Änderungen im vorhergehenden Programm gehen verloren.
- Mit "Kanal grundstellen" in der Steuerung gehen Breakpoints und beobachtete Variablen verloren.

5.2 NC-Programm im Debug-Mode starten

- ★ Nach dem Anwählen des NC-Programms in der Steuerung starten Sie den **CPL-Debugger**, wie in Abschnitt 3.2 beschrieben, am besten durch Doppelklicken auf die dazugehörige Konfigurationsdatei. Damit werden bereits gesetzte Breakpoints und beobachtete Variablen berücksichtigt.
- ★ Wechseln Sie zum Fenster "Programm".

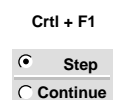
Debug-Mode starten

Das NC-Programm darf in der Steuerung noch nicht gestartet sein.



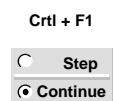
- ★ Wählen Sie den Softkey *Set Debug Mode*. Der **Debug-Mode** startet, der Softkey nimmt danach die Bezeichnung *Reset Ex. Mode* an. Zur Information erscheint auf der Standard-Bedienoberfläche der Steuerung im Fenster "Programm" im Bedienbereich "Abarbeiten" der Hinweis: "Debug-Mode aktiv!"

Debug-Einstellung wählen



Einstellung "Step":

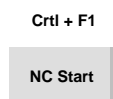
- Toggle-Softkey *Step / Continue*: Das Programm wird **schrittweise** ausgeführt, d.h. nach jedem Schritt muss der Softkey "NC Start" erneut gedrückt werden, um das Programm fortzusetzen.



Einstellung "Continue":

- Toggle-Softkey *Step / Continue*: Das Programm wird im **"Folgesatzbetrieb"** solange ausgeführt, bis es auf einen Breakpoint stößt und dort den Programmablauf unterbricht. Nach einer Unterbrechung kann das Programm mit dem Softkey "NC Start" erneut fortgesetzt werden, soweit dies möglich ist (z.B. kein Abarbeiten nach schweren Fehlern möglich).

NC-Programm starten



Arbeiten mit Gesamtsperre oder mit Steuerungs-Emulation

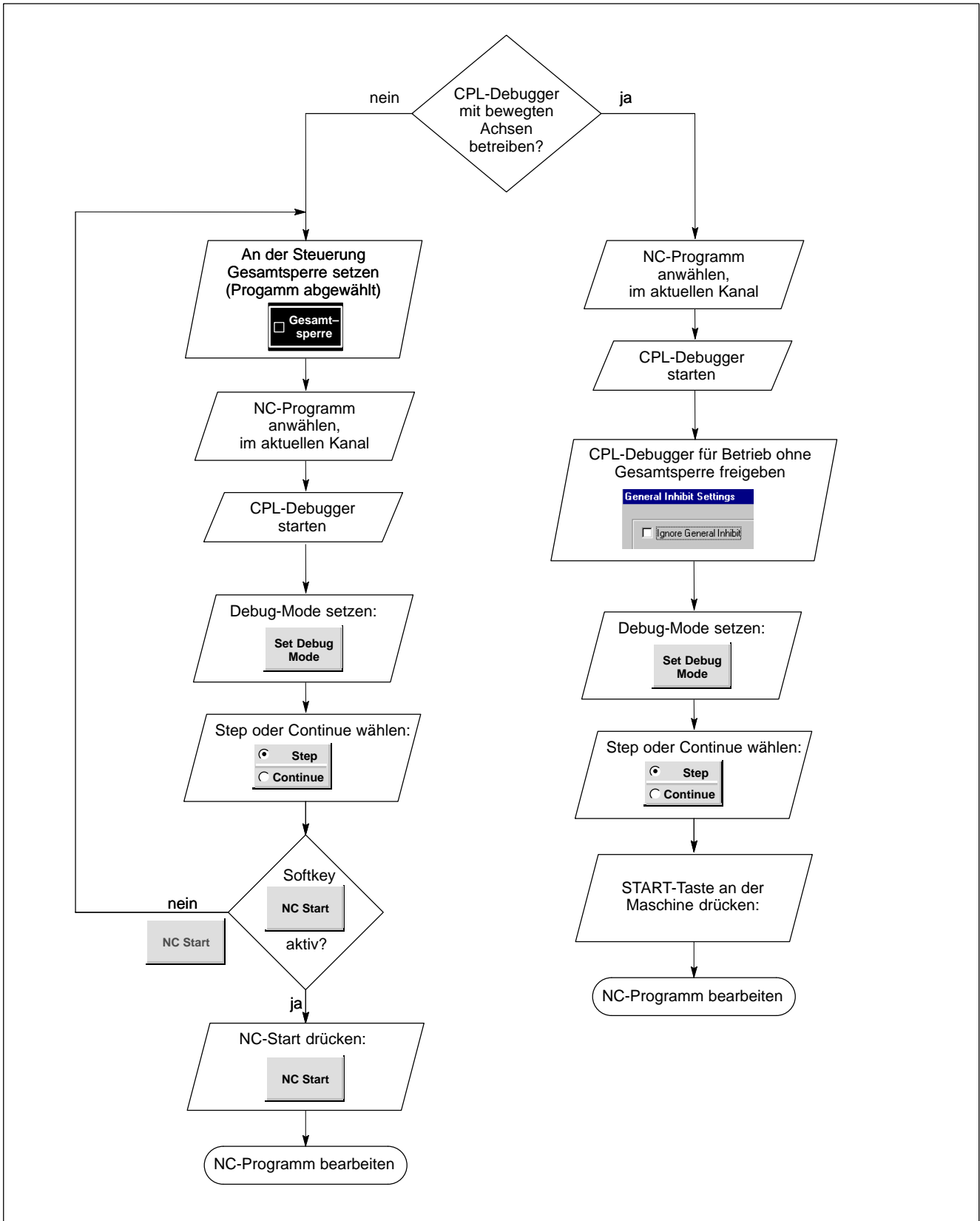
- Softkey *NC Start*: Im Debug-Mode wird das angewählte NC-Programm gestartet oder nach Halten des Programmablaufs fortgesetzt (entsprechend der Einstellung des Toggle-Softkeys "Step/Continue").

Arbeiten mit bewegten Achsen

- Start-Taste an der Steuerung: Das angewählte NC-Programm wird gestartet oder nach Halten des Programmablaufs fortgesetzt (entsprechend der Einstellung des Toggle-Softkeys "Step/Continue").

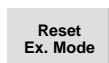
- ☞ **Nach Abwählen und Wiederanwählen eines NC-Programms muss der Debug-Mode erneut gesetzt werden.**

Die Vorgehensweise bis zum Starten des NC-Programms im Debug-Mode sind noch einmal vollständig im folgenden Flussdiagramm dargestellt.



Debug-Mode verlassen

Ctrl + F1



- ★ Wählen Sie den Softkey *Reset Ex. Mode*.

Der CPL-Debugger und die Steuerung verlassen den Debug-Mode. Der Softkey nimmt danach die Bezeichnung *Set Debug Mode* an.

Auf der Standard-Bedienoberfläche der Steuerung im Fenster "Programm" im Bedienbereich "Abarbeiten" erscheint unter Ausführung der Hinweis: "Folgesatz", "Einzelsatz" oder "Einzelschritt",...

- Wird ein NC-Programm gestartet und befindet sich der CPL-Debugger nicht mehr im Debug-Mode, werden alle vorher gesetzten Breakpoints beim Abarbeiten des NC-Programms ignoriert.
- Erst nach Ab- und Wiederanwahl des laufenden NC-Programms kann der Debug-Mode wieder gestartet werden.

5.3 NC-Programm anzeigen

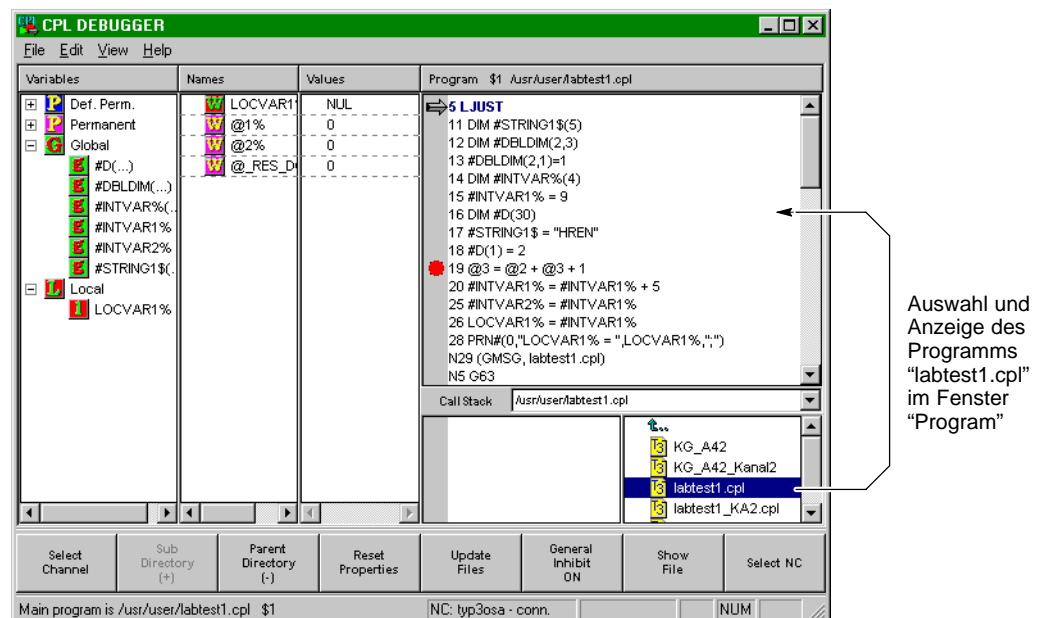
Fenster "Program"

Im Fenster "Program" des CPL-Debuggers wird immer das aktuell angeählte NC-Programm angezeigt, wenn keine andere Datei aus dem Verzeichnis im Fenster "File" zur Anzeige ausgewählt wurde. Ist kein NC-Programm in der NC angewählt, bleibt das Fenster "Program" leer.

Fenster "File"

Das Fenster "File" zeigt das Dateisystem der Steuerung an. Jede dort vorhandene Datei kann über die Dateiauswahl im Fenster "Program" angezeigt werden.

 **Nur aktive Programme können mit dem CPL-Debugger bearbeitet werden.**

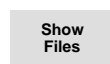


★ Gehen Sie im Fenster "File" zu dem gewünschten Programm:

- mit der Maus oder
- mit dem Softkey *Sub Directory*, alternativ <+> und dem Softkey *Parent Directory*, alternativ <->

 **Der Softkey *Update Files* aktualisiert den Verzeichnisinhalt.**


Crtl + F2



★ Wählen Sie die gewünschte Datei aus durch

- Doppelklicken oder
- Softkey *Show File*.

Danach wird das Programm im Fenster "Program" angezeigt.

 **Zum Anzeigen eines Programm im Fenster "Program" muss das Dateiattribut "Lesen" (r) gesetzt sein.**

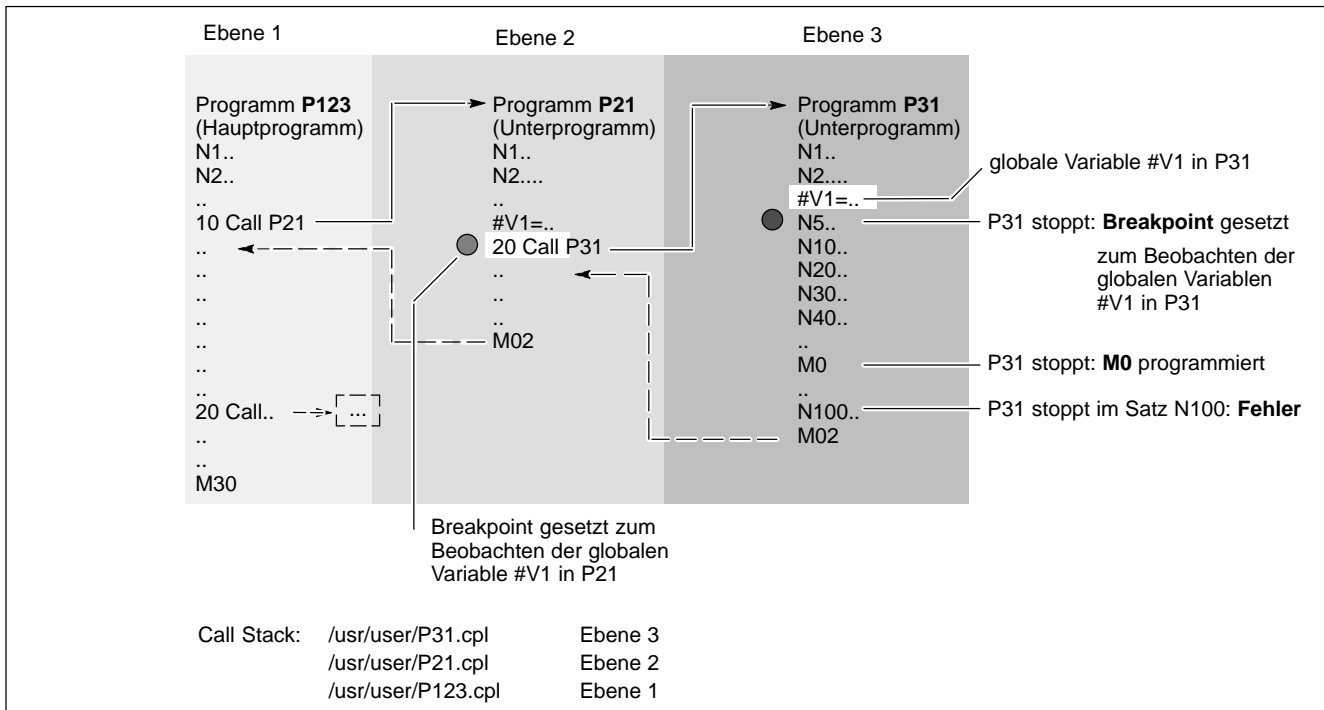
Programm im Call Stack auswählen

Über den Call Stack kann auf Unterprogramme zugegriffen werden, die bis zu einer Schachtelungstiefe von 7 bzgl. eines Hauptprogramms organisiert sind.

Das Unterprogramm kann in der gerade aktuellen Aufrufstruktur angezeigt werden, wenn es durch einen Fehler, durch "M0" oder durch einen aktiven Breakpoint zum Stehen gebracht wurde. In diesem Fall wird auch die Ausführung aller übrigen Programme, die das Unterprogramm aufgerufen haben, unterbrochen.

Die Namen aller in der Aufrufstruktur aktiven Programme werden im Fenster "Call Stack" angezeigt. Dort kann ein Programm ausgewählt und mit dem Softkey *Go to Call* im Fenster "Program" angezeigt werden.

Im ausgewählten Programm können nun lokale Variablen beobachtet und ggf. neue Breakpoints gesetzt werden. Die Aufrufstruktur im Call Stack gibt Auskunft, über welche Programmschachtelung das ausgewählte Programm aufgerufen wurde (siehe folgende Abb.):

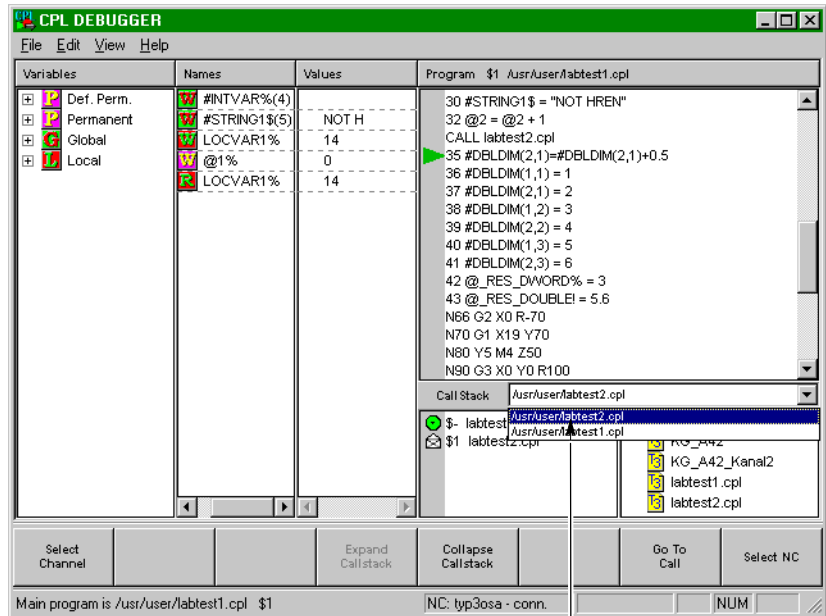


Ctrl + F3

★ Klicken Sie das Fenster "Call Stack" an.



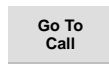
Mit den Softkeys *Expand Callstack* und *Collapse Callstack* öffnen/schließen sich die Verschachtelungsebenen, die durch Haupt- und Unterprogrammaufrufe verursacht sind.



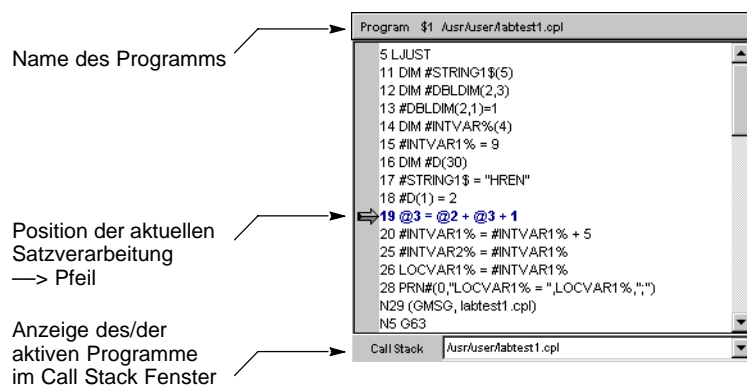
Auswahl eines Unterprogramms im Fenster "Call Stack"

Ctrl + F3

★ Wählen Sie eines der Programme (Unterprogramme) im Fenster "Call Stack" aus und drücken Sie den Softkey *Go to Call*.



Sie erkennen das zuletzt "rufende" aktive Programm an dem Pfeil im aktuell abzuarbeitenden Programmsatz, in dem die Satzverarbeitung steht.



Name des Programms

Position der aktuellen Satzverarbeitung
—> Pfeil

Anzeige des/der aktiven Programme im Call Stack Fenster

5.3.1 Breakpoints setzen und editieren

Breakpoints können im CPL-Debugger in DIN- und CPL-Programmen in verschiedenen Kanälen eingesetzt werden.

Der Breakpoint ist ein **Haltepunkt**, an dem die Satzverarbeitung im Folgesatz-Betrieb stoppt. An dieser Stelle können dann Variablenwerte überprüft und eventuell geändert werden.

Der Breakpoint erscheint als Kreis-Symbol an dem NC-Satz, **vor** dem der CPL-Debugger stehen bleiben soll.

Die Variablen-Informationen im Fenster "Values" beziehen sich auf alle bis zum Breakpoint bearbeiteten Sätze.

Alle folgenden Sätze, einschließlich des Satzes, vor dem der Breakpoint steht, sind noch **nicht** in die Programm-Verarbeitung eingegangen.

Sie nutzen Breakpoints:

- zum Testen von expliziten Stellen im Programm, wenn keine genaue Kenntnis der Fehlerursache vorhanden ist.
- zum Testen von verschachtelten Unterprogrammen, wenn die Fehlerursache dort zu suchen ist.

Folgende Tabelle zeigt die Steuerungsmöglichkeiten von Breakpoints:

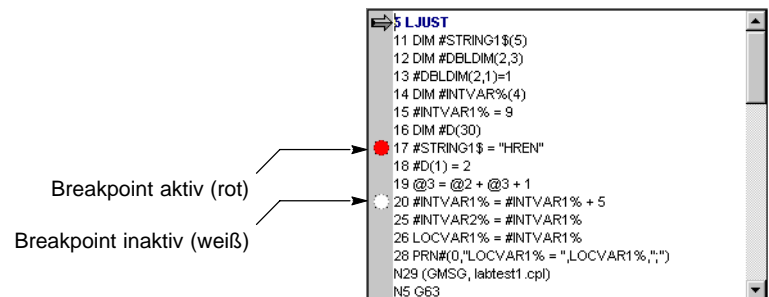
Breakpoint	Auswirkung	Softkey	Tastenaufruf im Menü "Breakpoints"	Fkt.-Taste
setzen	Breakpoint ist gesetzt und aktiv, der CPL-Debugger hält vor einem NC-Satz an, der durch einen gesetzten Breakpoint markiert ist.	Set Breakpoint	Enable	–
löschen	Breakpoint wird vollständig entfernt.	Delete Breakpoint	Delete	Shift + F2
deaktivieren	Der gesetzte Breakpoint ist ohne Funktion und hält das Programm nicht an. Die Markierung bleibt jedoch erhalten.	Disable Breakpoint (Toggle-Key: Enable Breakpoint)	Disable	Shift + F4
aktivieren	Inaktiver Breakpoint wird wieder aktiviert. Der CPL-Debugger hält vor einem NC-Satz an, der durch einen gesetzten Breakpoint markiert ist.	Breakpoint (Toggle-Key: Disable Breakpoint)	Enable	Shift + F3
löschen (alle Breakpoints)	Alle Breakpoint des im Fenster "Program" angezeigten Programms werden vollständig entfernt. Das Programm kann auch ohne Gesamtsperre abgearbeitet werden, wenn alle Breakpoints auch von ggf. vorhandenen Unterprogrammen deaktiviert sind.	Delete All Breakpoints	–	–
deaktivieren (alle Breakpoints)	Alle gesetzten Breakpoints des im Fenster "Program" angezeigten Programms sind ohne Funktion. Die Markierungen bleiben jedoch erhalten. Ein Programm kann auch ohne Gesamtsperre abgearbeitet werden, wenn alle Breakpoints auch von ggf. vorhandenen Unterprogrammen deaktiviert sind.	Disable All Breakpoints (Toggle-Key: Enable All Breakpoints)	–	–
aktivieren (alle Breakpoints)	Alle inaktiven Breakpoints werden wieder aktiviert. CPL-Debugger hält vor allen mit Breakpoint markierten NC-Sätzen	Enable All Breakpoints (Toggle-Key: Disable All Breakpoints)	–	–
Sichern	Alle Breakpoint-Positionen für späteres Bearbeiten mit entsprechender Reproduzierbarkeit sichern	–	Exit	Shift + F7

Breakpoint Setzen und Aktivieren

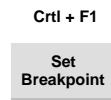
Die Anzahl der aktiven Breakpoints ist auf 20 beschränkt. Ein Breakpoint kann an einem beliebigen NC-Satz des Programms und/oder Unterprogramms in unterschiedlichen Kanälen gesetzt werden.

Wenn 20 Breakpoints gesetzt sind, werden die neu hinzukommenden Breakpoints inaktiv geschaltet. Zusätzlich wird eine Meldung ausgegeben. Erst nach Deaktivieren eines der 20 Breakpoints kann ein inaktiver Breakpoint aktiviert oder ein neuer aktiver Breakpoint gesetzt werden.

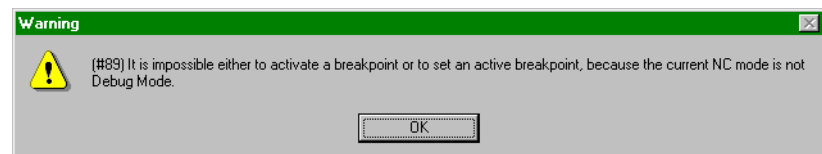
- ☞ **Die Farbe des Kreis-Symbols zeigt an, ob der Breakpoint aktiv (rot) oder inaktiv (weiß) ist.**



- ★ Wählen Sie das Fenster "Program" an.
- ★ Mit den Pfeiltasten oder per Maus-Cursor wählen Sie im Fenster "Program" den Programmsatz, an dem ein Breakpoint gesetzt werden soll.
- ★ Zum Setzen des Breakpoints drücken Sie den Softkey *Set Breakpoint* oder klicken Sie einmal mit dem Cursor auf den grauen senkrechten Balken. Es erscheint ein roter Kreis (Breakpoint gesetzt und aktiviert). Klicken Sie ein zweites Mal auf den roten Kreis, um den Breakpoint zu deaktivieren.



- ☞ **Breakpoints können nur im "Debug-Mode" aktiv gesetzt oder aktiviert werden. Andernfalls erscheint eine Fehlermeldung und der Breakpoint wird inaktiv (weiß) gesetzt.**



Breakpoint Deaktivieren und Löschen

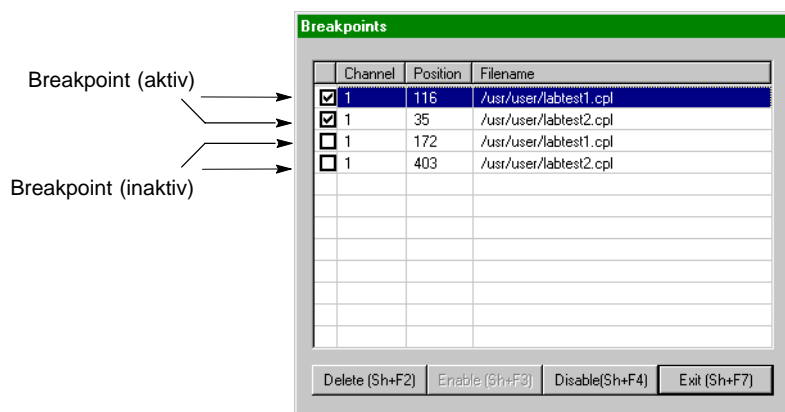
- ★ Wählen Sie das Fenster "Program" an.
- Crtl + F1 ★ Zum Deaktivieren der Breakpoints drücken Sie den Softkey *Disable Breakpoint* oder klicken Sie **einmal** mit dem Cursor auf den roten Kreis. Sind mehrere Breakpoints im Programm vorhanden, lassen sich alle gemeinsam mit dem Softkey *Disable All Breakpoints* deaktivieren.
- Crtl + F1 ★ Zum Löschen der Breakpoints drücken Sie den Softkey *Delete Breakpoint*, oder klicken Sie **zweimal** mit dem Cursor auf den roten Kreis. Sind mehrere Breakpoints im Programm vorhanden, lassen sie alle gemeinsam mit dem Softkey *Delete All Breakpoints* löschen.

☞ **Das Deaktivieren und Löschen aller Breakpoints erstreckt sich nur auf das im Fenster "Program" angezeigte Programm.**

Liste vorhandener Breakpoints anzeigen

Das Menü *View ► Breakpoints* enthält einen Überblick und den Status der gesetzten Breakpoints aller Programme und den zugehörigen Unterprogrammen:

- **Kanalzugehörigkeit** des/der angewählten Programme
- **Position** (Position der mit einem Breakpoint markierten Zeile bezogen auf den Anfang des Programms)
- **Name und Pfad** des Programms, das Breakpoints enthält



Zusätzlich können in diesem Menü Breakpoints aktiviert (Enable), deaktiviert (Disable) und gelöscht (Delete) werden.

Aktive Breakpoints aus der Konfigurationsdatei laden

Sind **aktive Breakpoints** in der CPL-Debugger-Konfigurationsdatei eines zugehörigen NC-Programms gespeichert und wollen Sie diese in den CPL-Debugger laden (mit "Load Workspace" im Fenster "File"), muss der Debug-Mode aktiv sein.

Andernfalls werden die gespeicherten Breakpoints inaktiv gesetzt.

5.3.2 Variablen beobachten

Sie können für einzelne Programmvariablen überprüfen, ob die logische Wertzuweisung durch das Programm korrekt durchgeführt wurden.

Variablengruppen:

- Permanente Variablen
- Definierbare permanente Variablen
- Globale Variablen
- Lokale Variablen

Die Variablengruppen sind im Fenster "Variables" aufgelistet und werden mit den Symbolen **P** (Permanent und Def.-Permanent), **G** (Global) und **L** (Local) gekennzeichnet.

Variables	Names	Values
Def. Perm.		
Permanent		
Global		
Local		

Variablengruppen

Einzelne Variablen anzeigen

Bei Anwahl eines Programms werden automatisch **alle** im Programm vorkommenden Variablen nach den genannten Variablengruppen klassifiziert und unter der jeweiligen Gruppe eingeordnet. Die mit "+" gekennzeichneten Variablengruppen enthalten jeweils die Namensliste der verwendeten Programmvariablen.

Durch Klick auf das "+" Zeichen oder mit den Pfeiltasten "⬇", "⬆" in Verbindung mit dem Softkey *Expand Branch* klappt die **Namensliste** der im Programm verwendeten Variablen heraus.

Variables	Names	Values
Def. Perm.		
Permanent		
Global	#D(30)	
	#DBLDIM(2,3)	
	#INTVAR%(4)	
	#INTVAR1%	
	#INTVAR2%	
	#STRING1\$(5)	
Local	LOCINT1%	
	LOCINT2%	

Globale Variablengruppe (aufgeklappt)

Namensliste der verwendeten Variablen

☞ **Steht der Cursor auf einer der aufgeklappten Variablen, sind die vorher gedimmten Softkey "Quick Watch", "Write Watch", "Read Watch" aktiv.**

Array-Variablen anzeigen

Bei mehrdimensionalen Variablen (Arrays) muss die Satzverarbeitung die Größe der entsprechenden Array-Variablen durch die "DIM"-Anweisung bearbeitet haben, bevor eine Array-Variable zur Beobachtung ausgewählt werden kann (zu erkennen an den nicht gedimten Softkeys F5, F6, F7, wenn die Array-Variable anwählt ist).

Anzeige globaler Array-Variablen vor der Dimensionierung im Watch Variables Fenster

Globale Array-Variable noch dimensionslos

Satzverarbeitung steht noch vor den Anweisungen zur Dimensionierung der Array-Variablen

Softkeys Quick-, Write-, Read-Watch noch gedimmt

The screenshot shows the 'CPL DEBUGGER' interface. On the left, the 'Variables' pane shows a tree structure with 'Global' expanded to show '#D(...)' selected. The main window displays assembly code with line 12 'DIM #DBLDIM(2,3)' highlighted. The bottom status bar shows 'Main program is /usr/user/labtest1.cpl \$1' and 'NC: typ3osa - conn.' with buttons for 'Quick Watch', 'Write Watch', 'Read Watch', and 'Select NC'.

Anzeige globaler Array-Variablen nach der Dimensionierung im Watch Variables Fenster

Globale Array-Variable ist dimensioniert

DIM-Anweisung

Satzverarbeitung hat Anweisungen zur Dimensionierung der Array-Variablen durchlaufen

Softkeys Quick-, Write-, Read-Watch nach der Dimensionierung der Array-Variablen verfügbar

The screenshot shows the 'CPL DEBUGGER' interface. In the 'Variables' pane, '#D(30)' is now selected. The main window shows the code with line 18 '#D(1) = 2' highlighted. The bottom status bar is the same as in the previous screenshot, but the 'Quick Watch', 'Write Watch', and 'Read Watch' buttons are now active.

**Beobachtungsliste,
Variablen hinzufügen**

Die Beobachtungsliste im Fenster "Names" setzt sich aus ausgewählten Variablen zusammen.

Die Variablen werden in der Reihenfolge angezeigt, in der sie in die Liste aufgenommen wurden. Die Liste kann jederzeit durch Hinzufügen oder Entfernen von Variablen erweitert oder verkleinert werden.

Variables	Names	Values
Def. Perm.	LOCVAR1	NUL
Permanent	@1%	0
Global	@2%	0
Local	@_RES_D	0
LOCVAR1%	LOCVAR1	NUL

Variablen übernehmen mit "Read Watch"

Beobachtungsliste

☞ Welche Variablen in das Fenster "Names" zur Beobachtung übernehmen und welche Informationen herausgefiltert werden sollten, hängt vom jeweiligen Problem im Programm ab (siehe Kap. 2, Fehlersuche im Programm).

★ Wählen Sie das Fenster "Variables" an.

Array-Variablen können erst in das Fenster "Names" übernommen werden, wenn die erforderlichen CPL-Programmsätze für das Dimensionieren (DIM...) der Array-Variablen abgearbeitet sind.

Setzen Sie dazu unmittelbar nach dem Dimensionieren der Variablen einen Breakpoint, damit der CPL-Debugger hält und Sie die erforderlichen Array-Variablen in das Fenster "Names" übernehmen können:

- entweder die gesamte dimensionierte Array-Variablen,
- oder nur den Teil, in dem das zu beobachtende Element steht.

Auswahl der gesamten Array-Variablen DBLDIM(2,3) für die Anzeige im Fenster "Names"

Auswahl eines Teils der Array-Variablen (nur DBLDIM(1,1)) für die Anzeige im Fenster "Names"

Crtl + F6

Expand Branch

Collapse Branch

Um alle Elemente der Array-Variablen anzusehen, benutzen Sie die Softkeys *Expand Branch* und *Collapse Branche*.

- ★ Wählen Sie eine Variable im Fenster “Variables” aus, um sie der Beobachtungsliste im Fenster “Names” hinzuzufügen. Dabei wird festgelegt und über ein Symbol gekennzeichnet, auf welche Weise die Variable beobachtet werden soll:

Ctrl + F6

Quick
Watch

- *Quick Watch* – synchrone Abfrage des Variablenwertes
Quick Watch zeigt den **augenblicklichen** Wert einer Variablen im Fenster “Names” an. Die Anzeige wird nach dem Fortsetzen oder Starten des Programms sofort aus der Beobachtungsliste gelöscht, da der Wert der Variablen nur an einer bestimmten “Stelle” im Programm interessiert. Dazu muss an der gewünschten Stelle ein Breakpoint gesetzt werden.

Ctrl + F6

Write
Watch

- *Write Watch* – asynchrone Beobachtung bei Schreibzugriffen
Mit *Write Watch* wird die Anzeige des Variablenwertes im Fenster “Values” jedesmal aktualisiert, wenn auf die Variable schreibend zugegriffen wird, d.h. ein neuer Wert zugeordnet wird. Auf diesem Weg kann während des gesamten Programmablaufs beobachtet werden, mit welchen neuen Werten die Variable versehen wird.




Ctrl + F6

Read
Watch

- *Read Watch* – asynchrone Beobachtung bei Lesezugriffen
mit *Read Watch* wird die Anzeige des Variablenwertes im Fenster “Value” immer dann aktualisiert, wenn auf die Variable lesend zugegriffen wird. Auf diesem Weg kann während des gesamten Programmablaufs beobachtet werden, welche Werte aus der Variablen gelesen werden.

Jede Beobachtungsvariante wird durch ein eigenes Symbol im Fenster “Names” gekennzeichnet.

Beispiel:

 LOCVAR1% → Quick Watch Beobachtung
 LOCVAR1% → Write Watch Beobachtung
 LOCVAR1% → Read Watch Beobachtung

Für die Variable LOCVAR1% wurden 3 Beobachtungsvarianten gewählt.

Einschränkung

Maximal 40 Variablen können gleichzeitig in die Beobachtungsliste aufgenommen werden.

Beim Versuch, mehr als 40 Variablen aufzunehmen, wird eine Warnung angezeigt. Bereits vorhandene Variablen müssen aus der Liste entfernt werden, um neue aufnehmen zu können.

Beispiel: Schreibender und lesender Zugriff auf die Variablen A, B, C

```
18  B=1
19  A=0
20  A=B+5
21  A=7
22  A=10
23  C=A
```

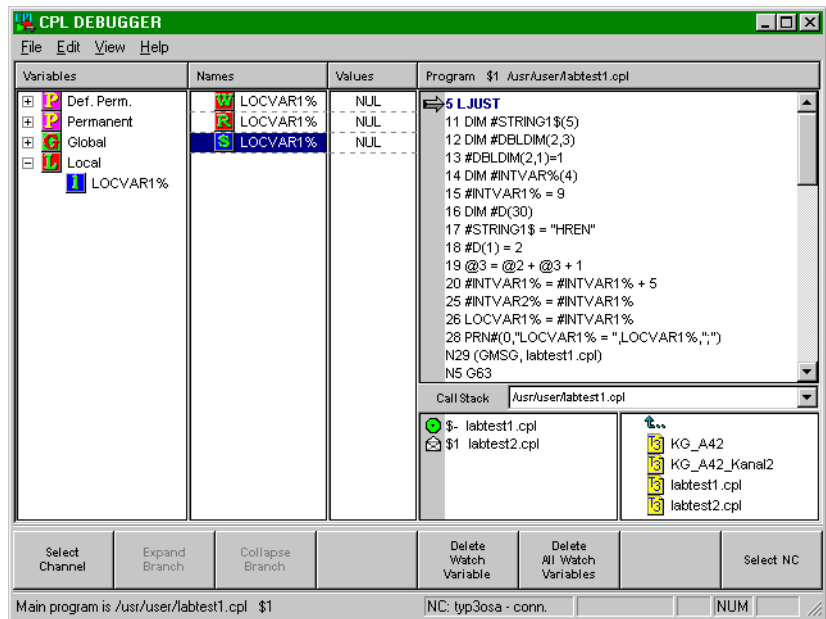
Im “Write Watch”-Mode wird der Wert der Variablen A jedesmal aktualisiert, da die Variable in jeder Zeile einen neuen Wert erhält.

Im “Read Watch”-Mode wird dagegen die Anzeige der Variablen A nur in Zeile 23 aktualisiert, weil nur da der Wert ausgelesen wird.

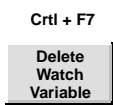
Werte-Verhalten siehe folgende Tabelle:

Zeile	Programm Code	Write Watch-Wert	Aktualisierung des Write Watch Wertes	Read Watch-Wert	Aktualisierung des Read Watch Wertes	Quick Watch-Wert
18	B=1	A= NUL B=1 C=NUL	A: nein B: ja C: nein	A=NUL B=NUL C=NUL	A: nein B: nein C: nein	A=NUL B=1 C=NUL
19	A=0	A=0 B=1 C=NUL	A: ja B: nein C: nein	A=NUL B=NUL C=NUL	A: nein B: nein C: nein	A=0 B=1 C=NUL
20	A=B+5	A=6 B=1 C=NUL	A: ja B: nein C: nein	A=NUL B=1 C=NUL	A: nein B: ja C: nein	A=6 B=1 C=NUL
21	A=7	A=7 B=1 C=NUL	A: ja B: nein C: nein	A=NUL B=NUL C=NUL	A: nein B: nein C: nein	A=7 B=1 C=NUL
22	A=10	A=10 B=1 C=NUL	A: ja B: nein C: nein	A=NUL B=NUL C=NUL	A: nein B: nein C: nein	A=10 B=1 C=NUL
23	C=A	A=10 B=1 C=10	A: nein B: nein C: ja	A=10 B=NUL C=NUL	A: ja B: nein C: nein	A=10 B=1 C=10

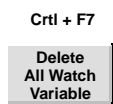
Variablen entfernen



★ Wählen Sie das Fenster "Names" an.



★ Markieren Sie (per Cursor-Taste oder Maus) die Variable, die Sie entfernen wollen und drücken Sie den Softkey *Delete Watch Variable*.



★ Zum Entfernen **aller** Variablen aus der Beobachtungsliste drücken Sie den Softkey *Delete All Watch Variables*.

Variablenwerte anzeigen

Im Fenster "Values" stehen die Werte der im Fenster "Names" aufgelisteten Variablen.

Names	Values
LOCVAR1%	14
LOCVAR1%	14
#INTVAR1%	14
#STRING1\$(5)	HREN
#STRING1\$(1)	H
#STRING1\$(2)	R
#STRING1\$(3)	E
#STRING1\$(4)	N
#STRING1\$(5)	NUL

Die Wertezuweisung und Aktualisierung der Variablen hängt vom Programmablauf ab:

- Nach Anwahl des Programms ermittelt der CPL-Debugger die Namen der Variablen und ordnet allen beobachteten Variablen (außer Array-Variablen) die Größe "NUL" zu (ohne Wert). Arrays erhalten "NUL", nachdem das Programm die Array-Dimension mit der Anweisung "DIM" eingelesen hat.
- Entsprechend des Programmablaufs aktualisiert der CPL-Debugger die Variablen bei jedem Schreib- oder Lesevorgang (Write/Read).

Variablenwerte editieren

Beim Programmtest können Sie einzelne Werte einer Variablen oder Zeichen eines Strings ändern, um den Programmablauf zu testen.

 **CPL-Stringkonstanten (z.B. A\$) können nicht geändert werden.**

- ★ Wählen Sie mit Softkey *Var Value Window* das Fenster "Values" an.
- ★ Suchen Sie die zu editierende Variable oder Array-Variable mit dem Softkey *Expand Branch* aus.
- ★ Danach drücken Sie den Softkey *Edit Variable*, um einen neuen Wert einzugeben.

Ctrl + F8

Edit Variable


Names	Values
LOCVAR1%	14
LOCVAR1%	14
#INTVAR1%	14
#STRING1\$(5)	HREN
#STRING1\$(1)	H
#STRING1\$(2)	R
#STRING1\$(3)	E
#STRING1\$(4)	N
#STRING1\$(5)	NUL

Ctrl + F8

Set Variable

- ★ Bestätigen Sie die Eingabe mit Softkey *Set Variable*.

Eine falsche Eingabe wird mit einer Warnmeldung quittiert.

 **Der Wert "NUL" wird einer Variablen zugewiesen, wenn der Inhalt mit der Taste <Backspace> oder <Delete> gelöscht wurde und keine neuen Zeichen oder Werte eingegeben wurden. Permanenten Variablen kann der Wert "NUL" nicht zugewiesen werden.**

5.3.3 Meldungen

Meldungen anzeigen

Beim Arbeiten mit dem CPL-Debugger werden Meldungen von der NC und vom CPL-Debugger im Fenster "Output" angezeigt. Dies ist wichtig, weil ein Fehler oft durch eine NC Laufzeitfehlermeldung angezeigt wird. Im Fenster "Output" wird ein NC-Fehler nur mit seiner Fehlernummer angezeigt. Nähere Informationen sind im Info-Dialog der Standard-Bedienoberfläche der Steuerung enthalten.

Im Fenster "Output" werden alle Meldungen der Steuerung mit Text und Symbol angezeigt:

- Fehler
- Warnungen
- Hinweise (PRN#(0,...), MSG , GMSG)

The screenshot shows the 'Output' window of the CPL-Debugger. The main area displays a list of G-code commands with their corresponding symbols:

- 26 LOCVAR1% = #INTVAR1%
- 28 PRN#(0,"LOCVAR1% = ",LOCVAR1%,",")
- N29 (GMSG, labtest1.cpl)
- ⇒ N5 G63 K5 (highlighted with a blue arrow)
- N10 G0 X0 Y0 Z0 M3 S1000
- N20 G1 X40 F20000
- N15 X40 Z22.11
- N30 G3 X21 Y5 R100
- N40 G1 Y70
- N50 X40 Y190 S500
- N60 G2 X0 R-70
- N65 G2 X40 R-70
- 30 #STRING1\$ = "NOT HREN"
- 32 @2 = @2 + 1
- CALL labtest2.cpl
- 35 #DBLDIM(2,1)=#DBLDIM(2,1)+0.5

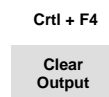
 Below the code list is a 'Call Stack' window showing the following entries:

- labtest1.cpl (green circle)
- labtest2.cpl (blue circle)
- labtest1.cpl (green circle)
- Error # 367 (red square)
- KG_A42 (blue square)
- KG_A42_Kanal2 (blue square)
- labtest1.cpl (blue square)
- labtest2.cpl (blue square)

Bei Fehlern und Warnungen unterscheidet die Farbe der Symbole, ob die Meldung:

- gerade ansteht : Hintergrundfarbe des Symbols: **rot**
- gelöscht ist bzw. bereits gelesen ist: Hintergrundfarbe des Symbols: **blau**

Meldungen im Fenster "Output" löschen



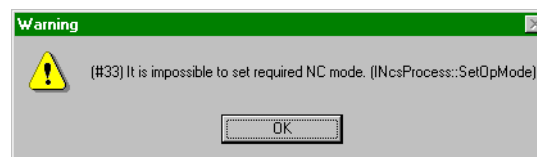
- ★ Wählen Sie mit dem Softkey *Output Window* das Fenster "Output" an.
- ★ Um alle Meldungen im Fenster "Output" zu löschen, drücken Sie den Softkey *Clear Output*.

5.4 Aktives Programm nach Abbruch bearbeiten

Wird ein aktives Programm aufgrund eines Fehlers abgebrochen, kann mit Hilfe des CPL-Debuggers eine **statische Bestandsaufnahme** der verwendeten Variablen für alle aktiven Programme durchgeführt werden. Durch den Call Stack können Zustände der lokalen Variablen in Unterprogrammen untersucht werden, wie sie zum Zeitpunkt des Unterprogramm-Aufrufs gültig waren.

Um diese Bestandsaufnahme durchzuführen, muss das abgebrochene Programm in der NC **noch angewählt** sein. Dann erst kann der CPL-Debugger beim Starten das aktive Programm laden, um alle relevanten Information für eine Fehlersuche bereitzustellen.

Breakpoints können in diesem Zustand nicht gesetzt werden, weil der Debug-Mode im CPL-Debugger nicht aktiv ist. Beim Versuch, in den Debug-Mode zu schalten, wird eine Fehlermeldung angezeigt:



CPL-Debugger starten

- ★ Starten Sie den CPL-Debugger.
- ★ Wählen Sie die Steuerung und den Kanal mit dem fehlerhaften Programm aus:
 - die Stelle, an der das NC-Programm aufgrund eines Fehlers angehalten hat, wird im Fenster "Program" des CPL-Debuggers angezeigt
 - die Aufrufstruktur (bei ggf. vorhandener Unterprogrammshachtelung im NC-Programm) wird im Fenster "Call Stack" angezeigt.
 - Unterprogramme können im Fenster "Call Stack" ausgewählt und über den Softkey *Go To Call* im Fenster "Program" angezeigt werden, um deren Programmvariablen zu untersuchen
 - aktivieren Sie im Fenster "General Inhibit Settings" die Checkbox *Ignore General Inhibit*. Danach sind im CPL-Debugger alle Programmvariablen und deren Inhalte zum Zeitpunkt des Abbruchs sichtbar.

Mit Hilfe dieser Informationen kann oft schon der Fehler gefunden werden. Mindestens ist aber der fehlerhafte Programmbereich bekannt, der dann im Debug-Mode untersucht werden kann.

- ★ Beseitigen Sie den Fehlerursache.
- ☞ **Zum Editieren eines Programm muss das Dateiattribut "Schreiben" (w) gesetzt sein.**

Programm testen

Vor der weiteren Maschinenbearbeitung sollten Sie das Programm nochmals auf Fehlerfreiheit testen.


Auch dabei kann der CPL-Debugger hilfreich sein, da die Informationen (Werte) zu verwendeten Variablen während des Programmablaufs jederzeit abrufbar sind.

6 Meldungen im CPL-Debugger

- System- und Benutzermeldungen des CPL-Debuggers werden in Popup-Fenstern mit Fehlertext und Klassifizierung dargestellt.

 Die Meldungen gliedern sich in die in Kap. 6.1 aufgeführten Fehler-Klassen und sind ab Kap. 6.4 tabellarisch aufgelistet.

- Fehler- und Warnmeldungen
 - des Steuerungssystems (zu erkennen an der Fehlernummer)
 - vom Programm, das mit dem CPL-Debugger bearbeitet wird (z.B. MSG., GMSG..)
 werden im Fenster "Output" des CPL-Debuggers und im Info-Dialog der Standard-Bedienoberfläche der Steuerung angezeigt.

 Nähere Erläuterungen finden Sie im Handbuch "Typ3 osa Fehlermeldungen".

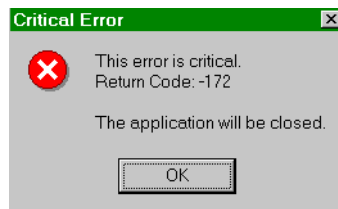
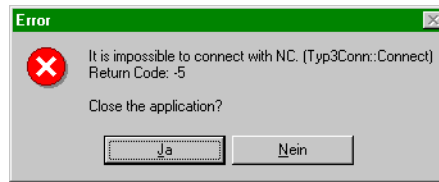
6.1 Fehler-Klassen CPL-Debugger

Typ	Klasse	Beschreibung, Beispiel
Systemfehler: – Error – Critical Error	schwerer Systemfehler*) → Critical Error	Interner Fehler, der dazu führt, dass kein weiterer Betrieb der Steuerung möglich ist. Der CPL-Debugger schließt nach Bestätigen automatisch.
	leichter Systemfehler*) → Error	Interner Fehler, der dazu führt, dass ein Teilbereich der Steuerung nicht mehr funktionsfähig ist. Der Benutzer kann entscheiden, ob er weiterarbeitet oder den CPL-Debugger schließt.
Warnung	System-Warnung	Warnung aus einem Teilbereich des CPL-Debuggers, z.B. Konfigurationsdaten konnten nicht geladen werden
Information: – allgemein – vom Benutzer-Programm	Systeminformation	z.B. Anzahl der Breakpoints überschritten
	programmierter Hinweis	Programmierter Hinweise vom NC-Programm: <ul style="list-style-type: none"> • (MSG,)... • PRN#(0,...
Eingabefehler	Bedienungsfehler	<ul style="list-style-type: none"> • Variablen-Wert ungültig • Kanalnummer inkorrekt (Kanal nicht vorhanden)

 *) Bei Systemfehlern muss ggf. der Service benachrichtigt werden

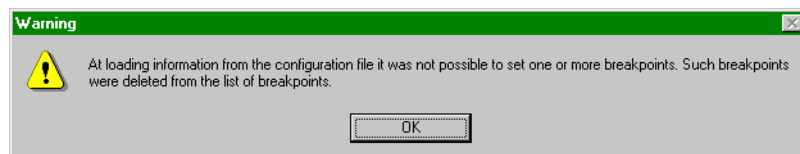
6.2 Popup-Fenster der Fehler-Klassen (Beispiele)

Systemfehler:

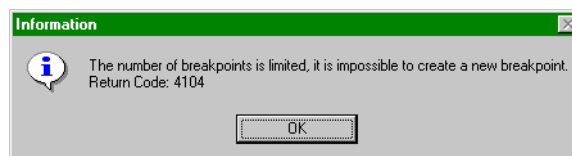


Die CPL-Debugger Meldungen "Information" und "Warning" werden in eigenen Dialogfenstern mit Fehlertext angezeigt:

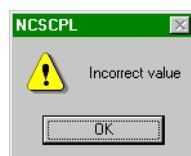
Warnung:



Information:



Eingabefehler werden vom CPL-Debugger als "Incorrect Value" in den Dialogfenstern "NcsCPL" und "CplDebug" angezeigt:



6.3 Benutzer-Reaktion auf Fehler und Meldungen des Systems

Bei "Informationen", "Warnungen" und bei "leichten Systemfehlern" können Sie je nachdem, wie stark das System gestört ist, folgende Maßnahmen durchzuführen:

- Meldungen oder Fehler **verwerfen**, solange man mit dem CPL-Debugger noch arbeiten kann bzw. die Informationen zur Fehlerbeseitigung geliefert werden.
- bei Problemen mit der Verbindung zur NC, nochmals versuchen die Verbindung mit dem Menüpunkt "SetTyp3" nochmals herzustellen
- Programm (Applikation) neu starten
- CPL-Debugger **und** Steuerung neu starten
- den CPL-Debugger mit anderen Steuerungen testen
- Service benachrichtigen


 Bei "schweren Systemfehlern" hilft ggf. nur noch:

1. Programm (Applikation) neu starten
2. CPL-Debugger **und** Steuerung neu starten
3. CPL-Debugger mit anderen Steuerungen testen
4. Service benachrichtigen

6.4 Fehlermeldungen

Die nachfolgende Fehler-Tabelle enthält neben angezeigten Fehlernummern und Fehlertexten folgende Informationen:

- Hilfe: Erklärung der Fehlerursache
Nicht jede Fehler-Hilfe zeigt die tatsächliche Fehlerursache an. Fehler können durch Kausalitäten mittelbar entstehen. Für eine endgültige Klärung eines zunächst nicht erklärbaren Fehlers wenden Sie sich bitte an den Kunden-Service.
- Fehler-Behebung und Klasse: Informationen und Maßnahmen zur Fehlerbeseitigung durch den Anwender
- Fehleranzeige löschen durch: Maßnahme zum Löschen der Fehlermeldung

 **Mit Löschen der Fehleranzeige im Fenster "Output" des CPL-Debuggers wird nicht die Fehleranzeige in der Standard-Bedienoberfläche der Steuerung gelöscht. Dies muss explizit in der NC geschehen.**

Wird die Fehleranzeige in der NC gelöscht, bleibt dennoch die Fehleranzeige im Fenster "Output" des CPL-Debuggers stehen.

Nr	Fehlertext	Hilfe	Fehler-Behebung	Klasse	Fehleranzeige löschen durch
	NcsCPL Incorrect value	Eingabe-Wert der editierten Variablen passt nicht zum Variablentyp.	Eingabe eines Wertes, der dem erforderlichen Variablentyp entspricht (s.a. CPL Programmierhandbuch).	Eingabefehler	
	NcsCPL The value NUL cannot be assigned to the permanent variable. The old value will be restored.	Es wurde versucht, einen NUL-Wert einer permanenten Variablen zuzuweisen.	Einen für eine permanente Variable geeigneten Wert eingeben, oder alten Wert akzeptieren.	Eingabefehler	
	The number of watched variables is limited, it is impossible to set for asynchronous watching a new variable.	Anzahl der asynchron beobachteten Variablen ist überschritten.	Löschen von asynchron beobachteten Variablen aus der Beobachtungsliste, bis weitere Variablen in die Liste aufgenommen werden können.	Information Variable	
	The number of breakpoints is limited, it is impossible to create or activate a new breakpoint.	1) Anzahl der gesetzten Breakpoints ist überschritten 2) Setzen eines Breakpoints in einem inaktiven Programm	1) Löschen eines oder mehrere vorhandener Breakpoints, um neue Breakpoints zu aktivieren. 2) Breakpoint bleibt inaktiv (auch für den Fall, dass das inaktive Programm angewählt wird.)	Information Breakpoint	
	At loading information from the configuration file it was not possible to set one or more breakpoints. Such breakpoints were deleted from the list of breakpoints.	1) Es ist kein angewähltes Programm, dessen Name im Konfigurationsfile gespeichert ist, beim Starten des CPL-Debuggers vorhanden. 2) Ein anderes Programm war beim Erstellen des Konfigurationsfiles aktiv. 3) Das Programm wurde nach dem Speichern der Konfigurationsdaten nochmals editiert (Debug-Informationen können ggf. verloren gehen)	1) CPL-Debugger erst starten, wenn entsprechendes Programm angewählt wurde. 2) Debuggen fortsetzen (Breakpoint-Informationen sind jedoch verloren) (falls erforderlich, speichern Sie die Konfigurationsdatei unter dem Namen des Programms ab, das bearbeitet wird: z.B.: TEST.CPL → TEST.cld) 3) Debuggen fortsetzen. Ggf. Breakpoints,.. neu setzen und im Konfigurationsfile speichern.	Warnung Breakpoint	

	<p>CplDebug It is impossible to set the channel <number>... (INcsChannel::SetChannel)</p>	<p>1) Timeout zu klein eingestellt.</p> <p>2) Anwahl eines nicht freigegebenen bzw. nicht vorhandenen Kanals.</p> <p>3) Anwahl einer nicht freigegebenen bzw. nicht vorhandenen Steuerung.</p>	<p>1) CPL-Debugger schließen; Timeout vergrößern und CPL-Debugger neu starten</p> <p>2) Kanalnummer prüfen (im Kanaldialog der NC). Anwahl der Kanalnummer im Dialog "Settings". Evtl. andere NC und dort gesuchten Kanal wählen.</p> <p>3) CPL-Debugger schließen; Starten des CPL-Debuggers mit Angabe des NC Namens. Anmerkung: über den Dialog "Settings" kann der NC Name bei laufendem CPL-Debugger geändert werden. Der CPL-Debugger wird danach mit der neuen NC verbunden.</p>	<p>Warnung Kanalnummer</p>	<p>1) CPL-Debugger neu starten</p>
	<p>It is impossible to connect with NC. (INcConnect::Connect)</p>	<p>1) Timeout zu klein eingestellt.</p> <p>2) Port Nummer (TCP/IP) ist falsch</p> <p>3) NC Name ist falsch</p> <p>4) NC ist nicht betriebsbereit oder Verbindung zur NC ist unterbrochen.</p>	<p>1) CPL-Debugger schließen; Timeout vergrößern</p> <p>2) CPL-Debugger schließen; CPL-Debugger neu starten mit korrekter Port Nummer.</p> <p>3) CPL-Debugger schließen; CPL-Debugger mit korrektem NC Namen wieder starten.</p> <p>Anmerkung: über den Dialog "Settings" kann der NC Name bei laufendem CPL-Debugger geändert werden. Der CPL-Debugger wird danach mit der neuen NC verbunden.</p> <p>4) CPL-Debugger schließen; NC bis zur Betriebsbereitschaft starten und ggf. Verbindung überprüfen; CPL-Debugger bei erfolgreichen NC Start und intakter Verbindung wieder starten.</p>	<p>Warnung Verbindung</p>	<p>1) CPL-Debugger neu starten</p> <p>2) CPL-Debugger neu starten mit korrekter Port Nummer.</p> <p>3) CPL-Debugger neu starten mit korrektem NC Namen.</p> <p>4) CPL-Debugger neu starten</p>
	<p>It is impossible to break the connection with NC. (INcConnect::Disconnect)</p>	<p>Kanalwahl mit zu kleinem Timeoutfenster abgebrochen</p>	<p>CPL-Debugger schließen; Timeout vergrößern.</p>	<p>Warnung Verbindung</p>	<p>CPL-Debugger neu starten</p>
	<p>It is impossible to get the quantity of variables. (IVariable::GetNumberOfVariable)</p>	<p>Timeout zu klein eingestellt</p>	<p>CPL-Debugger schließen; Timeout vergrößern.</p>	<p>Warnung Verbindung</p>	<p>CPL-Debugger neu starten</p>

	It is impossible to get the current value of the variable. (Variable::GetVarValue)	Timeout zu klein eingestellt	CPL-Debugger schließen; Timeout vergrößern.	Warnung Verbindung	CPL-Debugger neu starten
	It is impossible to get the current NC state. (INcsChannel::GetNcState)	Timeout zu klein eingestellt	CPL-Debugger schließen; Timeout vergrößern.	Warnung Verbindung	CPL-Debugger neu starten
	It is impossible to get the current NC mode. (INcsChannel::GetNcMode)	Timeout zu klein eingestellt	CPL-Debugger schließen; Timeout vergrößern.	Warnung Verbindung	CPL-Debugger neu starten
	It is impossible to read dry run state with synchronous request. (IDryRunMode::GetDryRunMode)	Timeout zu klein eingestellt	CPL-Debugger schließen; Timeout vergrößern	Warnung Verbindung	CPL-Debugger neu starten
	It is impossible to get the depth of the callstack. (IProgChain::GetProgChainDeep)	Timeout zu klein eingestellt	CPL-Debugger schließen; Timeout vergrößern	Warnung Verbindung	CPL-Debugger neu starten
	It is impossible to get the parameters of the subprogram. (IProgChain::GetSubProgLevel)	Timeout zu klein eingestellt	CPL-Debugger schließen; Timeout vergrößern.	Warnung Verbindung	CPL-Debugger neu starten
	It is impossible to stop watching the state of connection with NC.	OLE Fehler.	keine Angabe	Warnung Verbindung	
	(INcConnect::BreakWatchConnectState)				
	Connection with NC has been broken.	Verbindung zur NC unterbrochen	CPL-Debugger schließen; Verbindung zur NC überprüfen und wiederherstellen; CPL-Debugger wieder starten.	Error Verbindung	CPL-Debugger neu starten
	It is impossible to connect with NC. (INcConnect::Connect)	Verbindung zur NC nicht hergestellt	Verbindung zur NC über Softkey "Select NC" auswählen	Warnung Verbindung	
	It is impossible to set required NC mode. (INcsProcess::SetOpMode)	NC Mode (Betriebszustand) für den CPL-Debugger nicht geeignet	NC Zustand für das Debuggen einstellen. NC starten; CPL-Debugger starten	Warnung Verbindung	CPL-Debugger neu starten

A Anhang

A.1 Abkürzungen

Abkürzung	Bedeutung
BA	Betriebsart
BOF	Bedienoberfläche
C:	Laufwerksbezeichnung hier Laufwerk C (Festplattenlaufwerk)
EGB	Elektrostatisch gefährdete Baugruppen
ESD	electro static discharge Abkürzung für alle Bezeichnungen die elektrostatische Entladungen betreffen, z.B.ESD-Schutz, ESD-Gefährdung.
Fx	Funktionstaste mit der Nummer x
GBA	Gruppenbetriebsart
HP	Hauptprogramm
LSEC	Spindelsteigungsfehler-Kompensation (Lead Screw Error Compensation)
MDI	Betriebsart "Handeingabe" (manual data input)
MP	Maschinenparameter
MZA	Maschinenzustandsanzeige
NC, CNC	Numerische Steuerung
PE	Protective Earth, Schutzleiter
SK	Softkey
SPS	Speicherprogrammierbare Steuerung
UP	Unterprogramm
WMH	Werkzeugmaschinenhersteller
WZ	Werkzeug

A.2 Stichwortregister

A

Abbruch , aktives Programm, 5–22
 Aktive Breakpoints aus der Konfigurationsdatei laden, 5–14
 Aktives Programm im Call Stack auswählen, 5–10
 Aktives Programm nach Abbruch bearbeiten, 5–22
 Anzeige einzelner Variablen, 5–15
 Anzeige von Array-Variablen, 5–16
 Arbeitsbereich, Übersicht der Funktionen, 4–3
 Arbeitsweise des CPL-Debuggers, 5–1
 Auswahl, Steuerung und Kanal, 5–2

B

Bedienoberfläche, 4–1
 Behebung, 6–3
 Benutzer-Reaktion, auf Fehler, 6–3
 Beobachten von Variablen, 5–15
 Beobachtungsliste, 5–17
 asynchrone Beobachtung, 5–18
 synchrone Beobachtung, 5–18
 Bestimmungsgemäßer Gebrauch, 1–1
 Breakpoint, 5–12
 Aktive Breakpoints aus der Konfigurationsdatei laden,
 5–14
 Deaktivieren und Löschen , 5–14
 Liste anzeigen, 5–14
 Setzen und Aktivieren, 5–13

C

Call Stack, 5–10
 cld-Datei, 3–2
 Continue, 5–6
 Customer Programming Language, 2–1

D

Debug-Einstellung, wählen, 5–6
 Debug-Mode
 NC-Programm starten, 5–6
 setzen, 5–6
 verlassen, 5–8
 Desktop Symbol, 3–2
 Diskettenlaufwerk, 1–7
 Dokumentation, 1–7

E

EGB, 1–6
 Elektrostatisch gefährdete Baugruppe, 1–6
 EMV-Richtlinie, 1–1
 Erdungsarmband, 1–6
 ESD
 Arbeitsplatz, 1–6
 Schutz, 1–6

F

Fehler- und Warnmeldungen , 6–1
 Fehler-Klassen, 6–1
 Fenster, File (anzeigen), 5–9
 Fenstereigenschaften zurücksetzen , 4–8
 Festplatte, 1–7

G

Gesamtsperre
 Debug-Mode mit Gesamtsperre, 5–3
 Debug-Mode ohne Gesamtsperre, 5–2

H

Haltepunkte, 5–12
 Hilfe, 4–8, 6–3

I

Installation , 3–1

K

Konfigurationsdatei, erstellen, 3–4
 Konfigurationsdatei, 3–2
 laden, 3–5

L

Liste vorhandener Breakpoints anzeigen, 5–14
 Lösch-Mode, 6–3

M

Meldungen anzeigen, Ausgabe Fenster, 5–21
 Meldungen im CPL-Debugger, 6–1
 Meldungen im löschen, Fenster Output, 5–21
 Menüleiste, 4–1

N

NC-Programm im Debug-Mode starten, 5–6
 Niederspannungsrichtlinie, 1–1
 NOT-AUS-Einrichtungen, 1–5

P

Popup-Fenster, Fehlermeldungen, 6–2
 Programmanwahl, 5–5
 Programme anzeigen, 5–9

Q

Qualifiziertes Personal, 1–2

S

Settings, 5–2
 Sicherheitshinweise, 1–4

Softkey

- Clear Output, 5-21
- Collaps Branch, 5-17
- Collapse Callstack, 5-11
- Delete All Breakpoints, 5-14
- Delete All Watch Variables, 5-19
- Delete Breakpoint, 5-14
- Delete Watch Variable, 5-19
- Disable All Breakpoints, 5-14
- Disable Breakpoint, 5-14
- Edit Variable, 5-20
- Expand Branch, 5-17
- Expand Callstack, 5-11
- General Inhibit OFF, 5-4
- Go to Call, 5-11
- Load Workspace, 3-5
- NC Start, 5-6
- Quick Watch, 5-18
- Read Watch, 5-18
- Reset Ex. Mode, 5-8
- Reset Properties, 4-8
- Save Workspace, 3-4
- Select Channel, 5-2
- Select NC, 5-2
- Set Breakpoint, 5-13
- Set Debug Mode, 5-6
- Set Variable, 5-20
- Show File, 5-9
- Step/Continue, 5-6
- Write Watch, 5-18

Softkey-Leiste, 4-4

Startoptionen , 3-2, 3-5

Statuszeile, 4-2

Step, 5-6

Such-Funktion, 4-8

Symbole , 4-6

- File Fenster, 4-7

- Name Fenster, 4-6

- Output Fenster, 4-7

- Program Fenster, 4-6

- Variables Fenster, 4-6

System- und Benutzermeldungen, 6-1

T

Text suchen, 4-8

U

Übersicht, 2-1

V

Var Value Window, 5-20

Variablen entfernen, 5-19

Variablenwerte anzeigen, 5-20

Variablenwerte editieren, 5-20

Version, 1-8

Voraussetzungen, zum Arbeiten mit dem CPL-Debugger,
5-1

W

Warenzeichen, 1-8

Notizen:

Bosch-Automationsstechnik

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Industriehydraulik
Postfach 30 02 40
D-70442 Stuttgart
Fax (07 11) 8 11-18 57

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Mobilhydraulik
Postfach 30 02 40
D-70442 Stuttgart
Fax (07 11) 8 11-17 98

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Pneumatik
Postfach 30 02 40
D-70442 Stuttgart
Fax (07 11) 8 11-2 45 30

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Montagetechnik
Postfach 30 02 07
D-70442 Stuttgart
Fax (07 11) 8 11-77 77

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Fax (0 60 62) 78-4 28

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Schraub- und Einpreßsysteme
Postfach 11 61
D-71534 Murrhardt
Fax (0 71 92) 22-1 81

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Entgratetechnik
Postfach 30 02 07
D-70442 Stuttgart
Fax (07 11) 8 11-3 34 75

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
didactic
Postfach 11 62
D-64701 Erbach
Fax (0 60 62) 78-8 33

Österreich

Robert Bosch AG
Geschäftsbereich
Automationstechnik
Hüttenbrennergasse 5
A-1030 Wien
Fax (01) 7 97 22-60 96

Schweiz

Robert Bosch AG
Geschäftsbereich
Automationstechnik
Industriest. 31
CH-8112 Otelfingen
Fax (01) 8 47 14 99

Technische Änderungen vorbehalten

Ihr Ansprechpartner

BOSCH



Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Fax (0 60 62) 78-4 28