

# EPSON

*SRC5\*\*/SPEL 95 Ver. 2.3*

## **Benutzerhandbuch**

Rev. 1

EM004S766F

# EPSON

Benutzerhandbuch

SRC-5\*\*/SPEL 95 Ver. 2.3

Rev. 1

EPSON Deutschland GmbH  
Industrial Robots-Division

SEIKO EPSON CORPORATION

# GARANTIE

Die Roboter und die dazugehörigen Optionen werden erst nach strengsten Qualitätskontrollen, Tests und Prüfungen gemäß unserer hohen Leistungsstandards an unsere Kunden ausgeliefert.

Treten bei normalem Betrieb, innerhalb von 12 Monaten nach Lieferung, Störungen auf, leistet EPSON folgende Garantie:

- Ersatzteile
- Arbeitszeit vor Ort
- Reisekosten und Reisezeit innerhalb der ersten 6 Monate

Davon ausgenommen sind jedoch die folgenden Fälle:

1. Schäden oder Fehlfunktionen, verursacht durch falsche Bedienung.
2. Schäden oder Fehlfunktionen, verursacht durch Demontagen.
3. Schäden oder Fehlfunktionen, verursacht durch unerlaubte Modifikationen oder Reparaturen.
4. Schäden oder Fehlfunktionen, die durch extreme, roboterunabhängige Ursachen entstanden sind, wie z.B. Brand- oder Wasserschäden.

Die Garantie beginnt ab Auslieferung, also ab Lager Düsseldorf.

Die vorliegenden Garantiebestimmungen gelten jedoch nicht in allen Ländern in vollem Umfang. Zusätzliche Informationen zum Haftungsumfang erhalten Sie bei Ihrem SEIKO EPSON Partner, bei dem Sie das Robotersystem erworben haben.

# WARENZEICHEN

Pentium ist ein Warenzeichen der Intel Corporation.

Microsoft, Windows und das Windows Logo sind eingetragene Warenzeichen der Microsoft Corporation.

Andere Warenbezeichnungen und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Hersteller.

## Service Center

Bei Reparaturen, Inspektionen oder Justageeinstellungen nehmen Sie bitte Kontakt zum nächsten Roboter Service Center auf.

Halten Sie folgende Informationen bereit:

- Roboter Typ,
- Seriennummer,
- M.Code,
- Software Version und
- eine Beschreibung des Roboterproblems.

EPSON Deutschland GmbH  
Industrial Robots-Division  
Zülpicher Straße 6  
D-40549 Düsseldorf

Tel: 0211 5603-204

Fax: 0211 5603-319

## SERVICE CENTER USA

Seiko Instruments Inc USA  
Factory Automation Division  
2990 West Lomita Blvd  
Torrance, CA 90505

Tel: (310) 517-7842

Fax: (310) 517-8158

## HERSTELLER

SEIKO EPSON CORPORATION

Robots & FA. Systems Department

Okaya Plant No. 2

1-16-15, Daiei-cho

Okaya-shi, Nagano-ken, 394

Japan

TEL: 81-266-23-0020

81-266-24-2001

FAX: 81-266-24-2017

## Bemerkung

- Dieses Handbuch darf ohne Genehmigung weder kopiert noch vervielfältigt werden.
- Änderungen am Handbuch bleiben jederzeit und ohne vorherige Ankündigung vorbehalten.
- Wenn Sie Bemerkungen zum Inhalt haben oder Fehler in diesem Handbuch finden sollten, wird Ihnen Seiko Epson gerne zur Verfügung stehen.

# Sicherheitshinweis

Vor Inbetriebnahme des Roboters lesen Sie bitte die folgenden Sicherheitsmaßnahmen aus diesem und allen weiteren Handbüchern. Bewahren Sie diese Handbücher anschließend an einem für alle Betreiber zugänglichen Ort auf.



## WARNUNG

Dieses Zeichen zeigt, dass der Bediener sich tödlich oder schwer verletzen kann, wenn er die Anzeige ignoriert oder die Maschine falsch bedient wird.



## ACHTUNG

Dieses Zeichen zeigt, dass der Bediener einen Defekt verursacht oder das Material beschädigt wird, wenn die Anzeige ignoriert oder die Maschine falsch bedient wird.



## WARNUNG

- Bitte lesen Sie das Benutzerhandbuch sorgfältig, bevor Sie das Robotersystem in Betrieb nehmen. Hersteller/Lieferanten müssen sich auf die Abschnitte „Sicherheitstechnische Grundlagen“ und „Sicherheitstechnische Anforderungen“ im ersten Kapitel beziehen.
- Das Robotersystem wurde ausschließlich für den Gebrauch innerhalb eines Gebäudes hergestellt. Der Roboter muss in einem geeigneten Umfeld verwendet werden, damit er sicher und wie geplant funktionieren kann. Installieren Sie ihn in einer Umgebung, in der die in diesem Handbuch vermerkten Anforderungen berücksichtigt werden. Verwenden Sie den Roboter in keiner anderen Umgebung.
- Stellen Sie während des Normalbetriebes sicher, dass nur unterwiesene Personen Zugriff auf die Eingabekonsolle erhalten. Falsche Handhabung oder Bedienung kann zu Fehlfunktionen, Zerstörungen oder gefährlichen Unfällen führen. Unterwiesene Personen sind diejenigen, die an einer Schulung des Herstellers teilgenommen haben oder solche, die bereits erfahren sind im Umgang mit Robotern und alle dazugehörigen Handbücher genau gelesen haben.

# VORWORT

Das SRC500/SPEL 95-Benutzerhandbuch wurde für diejenigen Benutzer erstellt, die Anwendungsprogramme erstellen oder die den Roboter mittels manueller Einrichtung „Teachen“ programmieren. Bitte arbeiten Sie das Handbuch, wie auch die folgenden Handbücher, gründlich durch, bevor Sie die Komponenten benutzen.

## Handbücher

### **1. SRC5\*\*/SPEL 95-Einführung**

Dieses Handbuch gibt einen Überblick über die Steuerungen SRC500 und SRC520CE, den Manipulator und die Programmiersoftware „SPEL 95“. Beschrieben werden Themen wie: Sicherheit, Einstellung und Bedienmethoden. Bitte arbeiten Sie das Handbuch, wie auch die folgenden Handbücher, gründlich durch, bevor Sie die Komponenten benutzen.

### **2. SRC5\*\*/SPEL 95-Benutzerhandbuch**

Dieses Handbuch beschreibt die Funktionen der Programmiersoftware „SPEL 95“. Beschrieben werden Themen wie: Einricht-, Programmier- und Fehlerbeseitigungsmethoden sowie Dateiverwaltung. Bitte arbeiten Sie das Handbuch, wie auch die folgenden Handbücher, gründlich durch, bevor Sie die Komponenten benutzen.

### **3. SRC5\*\*/SPEL 95-Anhang**

Im Anhang werden die Inhalte der Handbücher SRC5\*\*/SPEL 95-Benutzerhandbuch und SRC5\*\*/SPEL 95-Einführung ergänzt. Sie erhalten hier einen Überblick über alle SPEL 95-Befehle und eine Fehlerliste sowie über die Installation der Software. Bitte arbeiten Sie auch den Anhang durch, bevor Sie die Komponenten benutzen.

### **4. Manipulator-Handbuch**

Handbuch zum Manipulator. Hier werden die Installation und der Anschluss von Peripheriegeräten erläutert.

### **5. Handbuch zur Robotersteuerung**

Handbuch zur Steuerung des Roboters. Hier werden Installation, Schaltereinstellungen und der Anschluss an Peripheriegeräte beschrieben. Außerdem enthält es Informationen über eine sichere Installation und eine sichere Benutzung.

Optionale Platinenhandbücher:	Handbuch zur Ethernet-Platine Handbuch zur E/A-Remote-Platine Handbuch zur Puls-Erzeugungs-Platine
Verwandte Handbücher:	Multi-Manipulator-Handbuch Vision-Calibration-Anleitung M300 Micro-Image-Checker-Benutzerhandbuch Anwender-Manipulator-Tuning Handbuch zum Handprogrammiergerät OP500

# Eingabeformate der Befehle

## Formatierungsregeln für die Eingabebefehle

Einfache Zeichenfolge:	Geben Sie die Zeichenfolge wie dargestellt ein. Beispiel: MOTOR EIN
[ ] (eckige Klammern):	Benutzen Sie dies zur Dateneingabe. Beispiel: EIN [Ausgabe Bitnummer]
(vertikale Striche):	Benutzen Sie dies zur Eingabe der Einstellungsmöglichkeiten. Beispiel: MOTOR  EIN   AUS
{ } (Klammern):	Benutzen Sie dies zur Eingabe nicht benötigter Einstellungen (Schalter etc.). Die Funktion variiert je nachdem, ob die Einstellung genutzt wird oder nicht. Beispiel: DIR {/W}
{ }n (Klammern + n):	Benutzen Sie dies um einzugeben, dass die in den Klammern stehenden Werte n-Mal wiederholt werden sollen (geben Sie eine Zahl für n ein). Geben Sie nur „n“ ein, heißt das, dass die Werte wiederholt werden können.
~ (Tilde):	Hiermit zeigen Sie, dass ein Befehl noch in die nächste Zeile weiterführt oder dass er noch zur vorhergehenden Zeile gehört.

# Standardspezifikationen der Robotersteuerungen

Modell	SRC5**		
Konfiguration	Steuereinheit	PC-kompatibler Computer optimiert für SRC5**: CPU: Pentium 100 MHz oder höher HD/RAM: 500 MB/32 MB oder mehr BS: Windows 95	
	Antriebseinheit	Bis zu 4 AC Servomotoren anschließbar	
Steuerbare Achsen	max. 16		
Manipulatorsteuerung	Programmiersprache und Robotersteuerungssoftware	SPEL 95 (multi-task-fähige Robotersprache)	
	Gelenksteuerung	Gleichzeitige Steuerung aller 6 Gelenke Software Servo-Steuerung Gelenk/Manipulator	
	Geschwindigkeitssteuerung	PTP-Bewegung	Programmierbar im Bereich von 1 bis 100%
		CP-Bewegung	Programmierbar (Werte können manuell eingegeben werden)
	Beschleunigungs-/ Verzögerungssteuerung	PTP-Bewegung	Programmierbar im Bereich von 1 bis 100%
CP-Bewegung		Programmierbar (Werte können manuell eingegeben werden)	
Positionierungssteuerung	PTP (Punkt-zu-Punkt-Steuerung) / CP (continuous path control – Pfadverlaufssteuerung)		
Speicherkapazität	Programmierbarer Bereich	640 KB	
	Positionsdatenbereich	max. 2.000 programmierbare Punkte	
Einrichtmethoden	Remote		
	Direkt		
	MDI (Manuelle Dateneingabe)		
Externe Eingangs-/ Ausgangssignale	Standard E/A	Eingang :16 Ausgang: 16	Remote-Funktionen zuteilbar
	Erweiterungs-E/A	Eingang: 32 Ausgang: 32 (pro Platine)	
	Remote E/A	Bis zu 128 (pro Platine) für Ein- und Ausgang	
	Gesamt E/A	Eingang :512 Ausgang: 512	
Kommunikationsschnittstelle	RS-232C (x2) (Standard)		
	Ethernet (Optional)		



Sicherheitsmerkmale	<ul style="list-style-type: none"> <li>• NOT-AUS-Schalter</li> <li>• Unterbrechung bei geöffneter Sicherheitstür</li> <li>• Stromsparmodus</li> <li>• Dynamische Bremse</li> <li>• Encoder-Kabel-Fehlererkennung</li> <li>• Überlastungs-Erkennung</li> <li>• Motordrehmoment-Fehlererkennung</li> <li>• Erkennung unregelmäßiger Geschwindigkeit</li> <li>• Servo Überlastung - Positions-Fehlererkennung</li> <li>• Servo Überlastung - Geschwindigkeits-Fehlererkennung</li> <li>• Erkennung von Unregelmäßigkeiten der CPU</li> <li>• Erkennung von Unregelmäßigkeiten des Speichers</li> <li>• NORMAL/ATTEND-Modus mittels verriegelbarem Schalter</li> <li>• Überhitzungs-Erkennung</li> </ul>	
Optionale Platine	<ul style="list-style-type: none"> <li>• Erweiterungs-E-/A-Platine (Eingang: 32; Ausgang: 32)</li> <li>• Puls-Erzeugungs-Platine (4 zusätzliche steuerbare Achsen/ Platinen)</li> <li>• Remote-E-/A-Platine (128 für beide Ein- und Ausgänge)</li> <li>• Ethernet-Platine</li> <li>• Vision Option</li> </ul>	
Optionale externe Geräte	<ul style="list-style-type: none"> <li>• Jog Pad JP500</li> <li>• Handprogrammiergerät OP500</li> <li>• Farb-LCD mit Einrichtmöglichkeit (in Kürze erhältlich)</li> </ul>	
Spannungsquelle	AC200 bis 220V ± 10%, Einzelphase 50/60Hz	
Stromverbrauch	Max. 2000 W (abhängig vom Manipulator-Modell)	
Isolierungswiderstand	100MΩ oder mehr	
Betriebstemperatur	5 bis 40°C	
Relative Luftfeuchtigkeit	10 bis 80%	
Gewicht	Steuereinheit	18 kg
	Antriebseinheit	14 kg
Größe	Bitte lesen Sie die Angaben zur Größe im Kapitel 11.2 „Abmessungen“ im Handbuch zur Steuerung.	

# SPEL 95 Ausführungsumgebung

## Hardware

SRC5\*\* Steuereinheit

### HINWEIS

SPEL 95 wurde entwickelt, basierend auf der Annahme, dass die Systemhardware zusammen mit der SRC5\*\*-Steuereinheit eingesetzt wird. Das bedeutet: Die Hardware kann nur mit der Steuereinheit zusammen betrieben werden. Die Steuereinheit arbeitet basierend auf der Architektur eines Personal Computers (PC). Sie sollten jedoch keine kommerziellen SCSI-Adapter oder andere Peripheriegeräte selbst anschließen. Dies kann funktionale Probleme verursachen, z. B. könnte der Roboter nicht wie geplant arbeiten, und die Echtzeit-Leistung der Steuerung könnte nicht erreicht werden.

## Monitor

SVGA (800 x 600) oder besser /XGA (1024 x 764) (empfohlen)

Siehe auch Anhang G für ausführliche Informationen zum Ändern des Monitors oder der Auflösung.

## Freier Festplattenspeicher

20 MB oder mehr.

## Speicher

32 MB oder mehr (nur zur Ausführung von SPEL 95, Optionen nicht eingeschlossen).

Sollen andere Applikationen parallel ablaufen, ist zusätzlicher Speicher dringend erforderlich.

## Einschränkungen von Windows95-Funktionen

Verwenden Sie auf gar keinen Fall einen Windows95-Bildschirmschoner! Diese Funktion ist werksseitig ausgeschaltet, wenn die Steuerung SRC5\*\* ausgeliefert wird.

# SRC500/SPEL 95-Vorsichtsmaßnahmen

## **SRC500/SPEL 95-Leistung**

Die SRC5\*\*-Steuereinheit arbeitet basierend auf der Architektur eines Personal Computers (PC). Im PC-Bereich wurden aufsehenerregende Verbesserungen bei CPUs und Motherboards gemacht. Aufgrund dieser PC-Fortschritte wurden und werden Upgrades bei der Steuereinheit ohne besondere Ankündigung vorgenommen.

Wenn also Ihre Maschine durch ein gleiches, neues System ersetzt wird, kann es möglich sein, dass sich die Leistung von altem und neuem System unterscheidet.

## **Roboterzelle gesteuert durch SRC500/SPEL 95**

Das SRC500/SPEL 95-System basiert auf einer Architektur, die durch eine einzelne Roboterzelle, die sich aus mehreren Manipulatoren zusammensetzt, eine Gesamtsteuerung bereitstellt.

Es ist möglich, das gleiche Projekt zu benutzen, um mehrere Manipulatoren zu steuern und eine Verzahnung zwischen den Manipulatoren zu erzeugen, ohne externe Hardware einzusetzen.

Denken Sie auf jeden Fall daran, dass die Konfiguration auf „mehreren Manipulatoren innerhalb einer Roboterzelle“ basiert und nicht auf „mehreren Manipulatoren innerhalb mehrerer Roboterzellen“. In Normalfall sind das NOT-AUS-Signal und das Sicherheitstürsignal der Roboterzellen in einem System. Auch in dem SRC500/SPEL 95-System werden diese Signale betreffend einer einzelnen Roboterzelle gehandhabt.

## **Gleichzeitig ausgeführte Anwendungen**

Die SRC5\*\*-Steuereinheit arbeitet basierend auf der Architektur eines Personal Computers (PC), und SPEL 95 läuft unter Windows 95.

Um sicherzustellen, dass der Roboter wie geplant arbeitet und die Echtzeit-Leistung der Steuerung erreicht wird, unterliegen Anwendungen, die gleichzeitig mit SPEL 95 unter Windows 95 arbeiten, bestimmten Bedingungen. Anwendungen, die den unten aufgeführten Bedingungen nicht unterliegen, können Maschinen- oder Leistungsprobleme bei SPEL 95 zur Folge haben. Installieren Sie solche Programme nicht, und lassen Sie sie nicht parallel zu SPEL 95 laufen.

- Verarbeitung nach Prioritätsklassen ist NORMAL\_PRIORITY\_CLASS oder darunter.
- Unabhängige Gerätetreiber sind nicht installiert.
- Dateidaten werden nicht periodisch abgefragt und upgedated.

Alle Rechte vorbehalten. Kein Teil dieses Handbuchs darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne die schriftliche Genehmigung der SEIKO EPSON Corporation reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Im Hinblick auf die Nutzung der im Handbuch enthaltenen Informationen wird keinerlei Patenthaftung übernommen.

Das Handbuch wurde mit der gebotenen Sorgfalt erarbeitet, SEIKO EPSON übernimmt jedoch keinerlei Patenthaftung für etwaige Fehler oder Auslassungen. Außerdem wird keine Haftung übernommen für Schäden, die sich durch Verwendung der im Handbuch enthaltenen Informationen ergeben.

Weder SEIKO EPSON Corporation noch ihre Tochtergesellschaften haften gegenüber dem Käufer dieses Produkts oder Dritten für Schäden, Verluste, Kosten oder Ausgaben, die von dem Käufer oder Dritten aufgrund von Unfall, Missbrauch des Produkts oder unerlaubter Änderungen, Reparaturen oder Neuerungen verursacht wurden.

SEIKO EPSON haftet nicht für Schäden oder Störungen, die sich durch Einsatz von Optionen oder Fremdzubehör ergeben, die keine originalen EPSON-Produkte sind, oder keine ausdrückliche Zulassung der Firma SEIKO EPSON als „EPSON Approved Products“ haben.

Warenzeichen

Copyright © 2000 by EPSON Deutschland GmbH, Düsseldorf.

# INHALTSVERZEICHNIS

## Benutzerhandbuch

### Kapitel 1 ..... 1

Roboter Manipulator Betrieb.....	1
1.1 Koordinatensystem.....	1
1.1.1 Positionsdaten.....	1
1.1.2 Roboter- und Lokales Koordinatensystem.....	2
1.1.3 Basiskoordinatensystem.....	5
1.2 SPEL 95 Beschreibung der Positionsdaten.....	6
1.2.1 Spezifikationsmethoden.....	6
1.3 Koordinatensystem und JOG-Modus.....	10

### Kapitel 2 ..... 11

Einrichten.....	11
2.1 Positionsdaten-Datei und Einrichten.....	11
2.1.1 Laden einer Positionsdaten-Dateien.....	13
2.1.2 Speichern der Positionsdaten (in eine Positionsdaten-Datei) nach dem Einrichten.....	14
2.2 Bedienung per JOG & Teach-Menü.....	16
2.2.1 Allgemeine Anzeige im [Pose Data Manager]-Fenster.....	16
2.2.2 Bedienung per JOG&Teach-Menü.....	20
2.3 Einrichten von Positionsdaten in Manipulatoren.....	24
2.3.1 Remote Teaching (Einrichten per Fernsteuerung).....	25
2.3.2 Direct Teaching (Direktes Einrichten).....	26
2.3.3 MDI Teaching (MDI Einrichten).....	27
2.3.4 Editieren der Positionsdaten.....	28
2.3.5 Löschen von Positionsdaten.....	28
2.4 Bedienung per Pose List-Menü.....	29
2.4.1 Änderungen an den maximalen Positionsdateneinstellungen.....	29
2.4.2 Editieren der Koordinaten.....	30
2.4.3 Anzeigen von ausschließlich existierenden Daten.....	31
2.4.4 Ändern der Reihenfolge der Daten.....	31
2.4.5 Löschen von Positionsdaten.....	32
2.4.6 Speichern als Anfangs- oder Standardpositionsdaten-Datei.....	32
2.4.7 Speichern als Positionsdaten-Datei.....	33
2.4.8 Laden einer Positionsdaten-Datei.....	33
2.5 Positionsdaten-Datei.....	34
2.6 Anfangs-Positionsdaten-Datei.....	36
2.6.1 Registrieren einer Anfangs-Positionsdaten-Datei.....	36
2.6.2 Freigeben der Anfangs-Positionsdaten-Datei.....	38
2.6.3 Updaten der Anfangs-Positionsdaten-Datei.....	38
2.7 Standard-Positionsdaten-Datei.....	39
2.7.1 Ändern der Standard-Positionsdaten-Datei.....	39
2.7.2 Updaten der Standard-Positionsdaten-Datei.....	40
2.8 Bedienung per JOG-Pad.....	42
2.8.1 Bedienung.....	42

2.8.2 NOT-AUS-Schalter.....	44
2.8.3 Bedienung per [Jog&Teach]-Menü .....	45
2.8.4 [Robot Control]-Panel (Menü: Robotersteuerung) .....	51
2.8.5 Bedienung per JOG-Pad .....	53
2.9 Bedienung mit dem Handprogrammiergerät .....	56
2.9.1 SPEL 95 - Einstellungen.....	56
2.9.2 JOG-Bedienung per OP500 .....	57
2.9.3 Einrichten.....	58
2.9.4 Direktes Einrichten .....	59
2.10 Speichern der Positionsdaten am Ende der Arbeit.....	61
2.11 Sicherungskopie der Positionsdaten-Dateien.....	62
2.12 Definition einer Palette.....	63

## **Kapitel 3..... 65**

Überblick über die Programmierung.....	65
3.1 Befehle und Anweisungen .....	65
3.2 Programmier-Regeln .....	66
3.2.1 Grundlegende Bestandteile eines Programms .....	66
3.2.2 Funktionsname .....	66
3.2.3 Mehrfachanweisung .....	66
3.2.4 Kommentare.....	66
3.2.5 Sprungmarken.....	67
3.2.6 Zeilennummern .....	67
3.3 E/A-Sprungmarken und Positionsmarken .....	68
3.3.1 E/A-Sprungmarke .....	68
3.3.2 Positionsmarke .....	69
3.4 Variablen.....	70
3.4.1 Variablenarten.....	70
3.4.2 Bereichsaufteilung für lokale Variablen .....	70
3.4.3 Numerische Variablen .....	70
3.4.4 Zeichenketten-Variablen.....	72
3.4.5 Bereichsvariablen.....	72
3.5 Operatoren .....	73
3.5.1 Mathematische Operatoren .....	73
3.5.2 Logische Operatoren.....	73
3.5.3 Relationale Operatoren .....	74
3.5.4 Zeichenketten-Operatoren.....	74
3.5.5 Rangfolge der Operatoren.....	74
3.5.6 Operationen mit ganzen und reellen Zahlen.....	74
3.6 Betriebsausdrücke.....	76
3.7 Zeichenketten-Ausdrücke.....	78
3.8 Bedingte Ausdrücke .....	79
3.9 Ereignisbedingte Ausdrücke.....	81
3.10 Funktionen .....	83
3.10.1 Funktionsstruktur .....	83
3.10.2 Funktionsaufrufe .....	84
3.10.3 Festlegen des Argumenttyps .....	87
3.10.4 Festlegen des Referenz-Arguments.....	88
3.10.5 Systemfunktionen.....	89

3.11 Programmsteuerungsanweisungen.....	90
3.12 Interrupt (Unterbrechung) .....	101

## **Kapitel 4..... 102**

Projektverwaltung .....	102
4.1 Projekte .....	102
4.1.1 Gruppen .....	103
4.1.2 Beziehung zwischen Projekten und Gruppen.....	104
4.1.3 Standard-Positionsdaten-Datei.....	104
4.1.4 Anfangs-Positionsdaten-Datei.....	105
4.1.5 Projekt-Manager-Anzeige .....	106
4.2 Erzeugen von Projekten .....	107
4.2.2 Erzeugen einer Programmdatei .....	108
4.2.3 Registrieren von Positionsdaten-Dateien .....	110
4.2.4 Erzeugen einer E/A-Sprungmarkendatei .....	113
4.2.5 Erzeugen einer INCLUDE-Datei (enthaltene Datei).....	114
4.2.6 Einschränkungen bei Dateinamen.....	115
4.3 Im- und Export von Dateien.....	116
4.3.1 Import von Dateien .....	116
4.3.2 Exportieren von Dateien in ein anderes Verzeichnis .....	123
4.4 Dateiverwaltung .....	124
4.4.1 Hinzufügen einer Programm- oder Anfangs-Positionsdaten-Datei zu einer Gruppe.....	124
4.4.2 Freigeben einer Programm- oder Anfangs-Positionsdaten-Datei von einer Gruppe.....	125
4.4.3 Ändern von Standard-Positionsdaten-Dateien .....	126
4.4.4 Ändern von E/A-Sprungmarken-Dateien .....	128
4.4.5 Ändern von Include-Dateien .....	130
4.4.6 Umbenennen einer Datei.....	132
4.4.7 Löschen von Dateien.....	133
4.5 Gruppenverwaltung .....	135
4.5.1 Gruppenstruktur .....	135
4.5.2 Ändern von Gruppen.....	139
4.6 Editieren von Projekten .....	141
4.6.1 Editieren von Dateien .....	141
4.6.2 Öffnen von Projekten.....	141
4.6.3 Schließen von Projekten .....	142
4.6.4 Einrichten von Projekten.....	143
4.7 Übertragen von Projekten.....	145
4.7.1 Projekte sichern .....	145
4.7.2 Projekte laden .....	146
4.8 Löschen von Projekten .....	147

## **Kapitel 5..... 148**

Editor .....	148
5.1 Erzeugen von Programmen und Include-Dateien .....	148
5.1.1 Erzeugen neuer Dateien .....	148
5.1.2 Bearbeitungsfunktionen .....	152
5.1.3 Pseudobefehle .....	158
5.2 Editieren von Positionsdaten-Dateien.....	162
5.2.1 Erzeugen einer neuen Datei .....	162
5.2.2 Bearbeiten bereits existierender Dateien.....	165
5.2.3 Bearbeiten .....	167
5.2.4 Ändern der Positionsdaten-Anzahl .....	172

5.3 Editieren von E/A-Sprungmarken-Dateien .....	173
5.3.1 E/A-Sprungmarken .....	173
5.3.2 Erzeugen neuer E/A-Sprungmarken-Dateien .....	175
5.3.3. Bearbeiten existierender Dateien .....	178

## **Kapitel 6..... 181**

Fehlerbehebung .....	181
6.1 Erzeugen von Tasks (Ausführen von Programmen) .....	182
6.1.1 Wechseln der Gruppe.....	182
6.1.2 Erzeugen von Tasks (Ausführen von Programmen).....	183
6.1.3 Anzeigen des Taskzustands .....	184
6.1.4 Ausführen von Funktionen nach Hinzufügen von Argumenten .....	185
6.1.5 Kontrollbericht.....	186
6.1.6 Bearbeiten von Programmen.....	187
6.1.7 Fehler 2524 .....	188
6.1.8 Schalterbedienung zum Ausführen von Programmen .....	189
6.1.9 Fortsetzen der angehaltenen Tasks.....	189
6.2 Hauptfunktion .....	190
6.2.1 Festlegen der Hauptfunktion .....	190
6.3 Hintergrundfunktion .....	191
6.4 Debuggen einzelner Tasks .....	192
6.4.1 Debuggen.....	192
6.4.2 Debuggen einer Task, die programmiert wurde nicht anzuhalten .....	195
6.4.3 Linien in dem Quellprogramm-Fenster .....	195
6.4.4 Zuordnen und Updaten lokaler Variablen.....	196
6.4.5 Bezug zum Aufruf-Protokoll.....	196
6.5 Variableneinstellungen .....	198
6.5.1 Lokale Variablen.....	198
6.5.2 Globale und Backup-Variablen.....	201
6.5.3 Backup und Wiederherstellen .....	203
6.6 Unterbrechungspunkte .....	207
6.7 Zuordnen und Ändern mit dem System-Monitor-Fenster .....	208
6.7.1 [Globale Variablen]-Menü.....	208
6.7.2 [Backup-Variablen]-Menü .....	210
6.7.3 [Anwender E/A]-Menü .....	212
6.7.4 [Speicher E/A]-Menü.....	213
6.8 Fehlerprotokoll.....	214
6.8.1 Das Fenster [Fehlerprotokoll] .....	214
6.9 Bedienung direkt aus dem Befehlsausführungsfenster.....	217
6.10 Bedienung aus dem Betriebsfenster .....	218
6.11 Testen von Programmen (Trockenlauf) .....	221
6.12 Anzeigen im AUTO-Modus .....	223



# BENUTZERHANDBUCH

In diesem Abschnitt werden die einzelnen Funktionen von SRC5\*\*/SPEL 95 detailliert beschrieben.

# Kapitel 1

## Roboter Manipulator Betrieb

### 1.1 Koordinatensystem

#### 1.1.1 Positionsdaten

Damit ein Roboter arbeitet, ist es notwendig, ein Programm zu erstellen, das dem Wesen der Arbeit entspricht und Positionsdaten einzurichten, die von diesem Programm benötigt werden. Positionsdaten enthalten Informationen, die die Lage und die Position eines einzelnen Punktes in einem einzelnen Koordinatensystem kennzeichnen. Der Roboter benutzt diese Informationen, um die an jedem Gelenk angebrachten Motoren zu steuern.

Positionsdaten enthalten die folgenden Informationen:

- Daten des Koordinatensystems (X, Y, Z, U)

- Merkmale des Koordinatensystems (lokal)

- Merkmale des Manipulators (Ausrichtung)

Die „Daten des Koordinatensystems“ beinhalten Daten, die die Position innerhalb eines rechteckigen Koordinatensystems (X, Y, Z) kennzeichnen und Daten, die die Drehung der rechtwinkligen Koordinaten um die Z-Achse (U) darstellen.

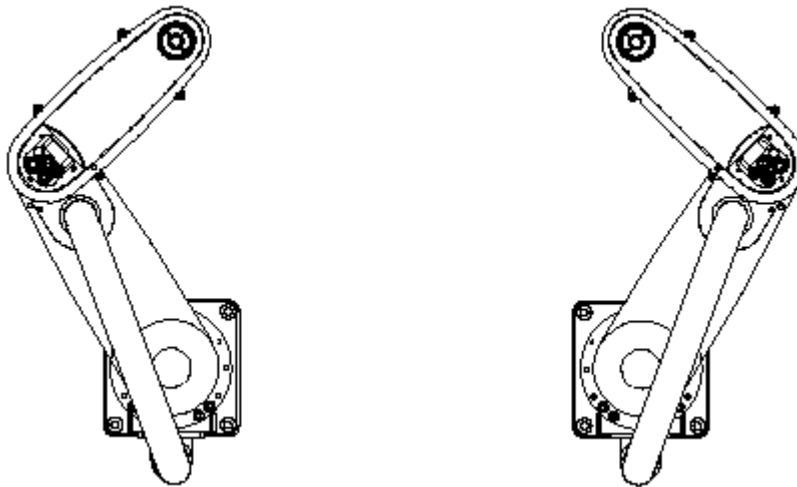
Die „Merkmale des Koordinatensystems (lokal)“ kennzeichnen die Anzahl der rechtwinkligen Koordinatensysteme, die als Standard für die Koordinatensystemdaten zugrunde liegt (Anzahl der lokalen Koordinatensysteme).

In Fällen, in denen der Roboter mit einer kleinen Anzahl von Positionsdaten eingerichtet wird, um eine einfache Task auszuführen, besteht keine Notwendigkeit, die Merkmale des Koordinatensystems (lokal) anzuwenden. Im Allgemeinen ist es jedoch so: Je komplizierter die einzelnen Tasks werden, je mehr Positionsdaten werden benötigt. Aber auch wenn eine große Anzahl von Positionsdaten benötigt wird, können alle Positionsdaten ausgeführt werden, indem man vorher nur einige von ihnen einrichtet. Zusätzlich gibt es Fälle, in denen die Beziehung zwischen verschiedenen Gruppen von Positionsdaten bereits durch Zeichnungen festgelegt ist.

Um einem Roboter die einfache Ausführung solcher Tasks zu ermöglichen, stellt SPEL 95 die Merkmale des Koordinatensystems (lokal) für Positionierungsdaten ein und ermöglicht, das Gesamtkoordinatensystem (welches als Standard für Positionierungsdaten zugrunde liegt) unabhängig zu steuern.

Mit zwei verschiedenen Armformen kann sich ein horizontaler Mehrfach-Gelenk-Manipulator (Typ: SCARA) zu nahezu jedem Punkt innerhalb eines vorgegebenen Arbeitsbereiches bewegen. Beispiele sehen Sie in Abb. 1. In Abb. 1 (a) ist die Reichweite des Bewegungsmechanismus nahezu genauso groß wie der linke menschliche Arm. Dieses System wird auch „Linker-Arm-System“ genannt. In Abb. 1 (b) ist die Reichweite des Bewegungsmechanismus nahezu genauso groß wie der rechte menschliche Arm. Dieses System wird auch „Rechter-Arm-System“ genannt.

Bei der Zuweisung einer Roboter-Task ist es notwendig, die für die jeweilige Armform eingerichteten Positionsdaten festzulegen. Sollte die Zuweisung misslingen, kann die Lage, abhängig von der Armform, leicht abweichen, was dazu führen kann, dass der Arm einen unerwarteten Weg einnimmt und so zu einer Störung mit den Peripheriegeräten führen kann. Das kann gefährlich sein! Um dies zu verhindern, wird die Form, die der Arm erreicht, wenn er sich in die angegebene Position bewegt, vorher in den Positionsdaten gespeichert. Daten dieses Typs werden „Merkmale des Manipulators (Ausrichtung)“ genannt. Diese Daten können auch vom Programm geändert werden.



(a) Armform „Linker-Arm-System“

(b) Armform „Rechter-Arm-System“

**Abb. 1: Beispiel: Bewegen zur gleichen Position mit unterschiedlichen Armformen**

### 1.1.2 Roboter- und Lokales Koordinatensystem

Wie bereits erwähnt, werden die Eigenschaften der Koordinatensysteme von den Positionsdaten zur Verfügung gestellt. Mit diesen Angaben können Anwender ihre eigenen Koordinatensysteme erstellen und alle Punkte der Positionsdaten, die die gleichen Koordinatensystemeigenschaften haben, bewegen. Ein vom Anwender unabhängig definiertes Koordinatensystem wird „lokales Koordinatensystem“ genannt.

Um ein lokales Koordinatensystem zu definieren, ist ein im Manipulator festgelegtes Referenzkoordinatensystem erforderlich.

Wie in Abb. 2 und 3 dargestellt, sind rechtwinklige Koordinatensysteme im Vorfeld für jedes Manipulatormodell festgelegt. Diese werden „Roboterkoordinatensysteme“ genannt. Wie in Abb. 4 dargestellt, wird die Z-Achse im Roboterkoordinatensystem der meisten Manipulatoren für die Bewegung in einer einzigen Richtung (abwärts) eingerichtet, wobei die Z-Koordinate am höchsten Punkt des 3. Gelenks den Wert 0 hat. Außerdem wird die U-Achse so eingerichtet, dass ihre +-Richtung, in Bezug auf die +-Richtung der Z-Koordinatenachse des Roboterkoordinatensystems, der Bewegung einer rechtsherum eingedrehten Schraube entspricht. Dies wird in Abb. 5 dargestellt. Das Roboterkoordinatensystem bezieht sich auf den Rotationswinkel jedes Manipulatormotors und kann nicht verändert werden.

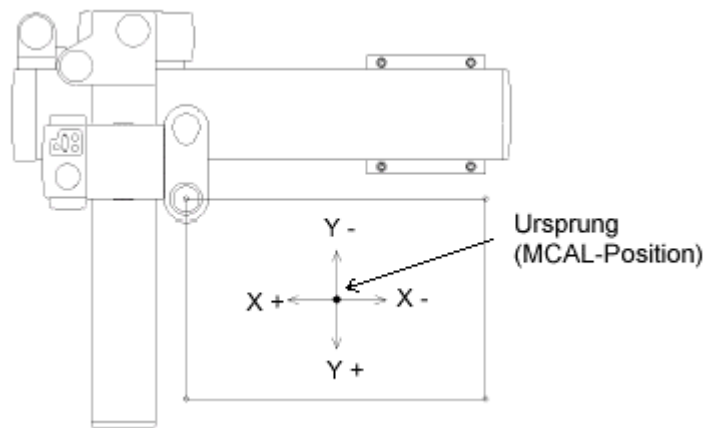


Abb. 2: Roboterkoordinatensystem eines rechtwinkligen Manipulators (Kartesischer Typ)

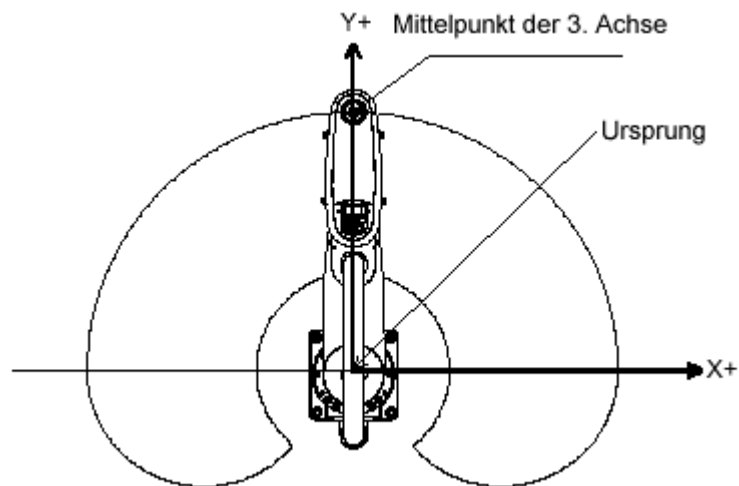


Abb. 3: Roboterkoordinatensystem eines horizontalen Knickarm-Manipulators (SCARA)

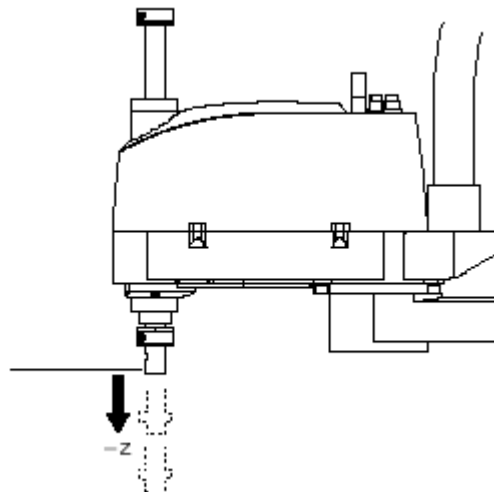
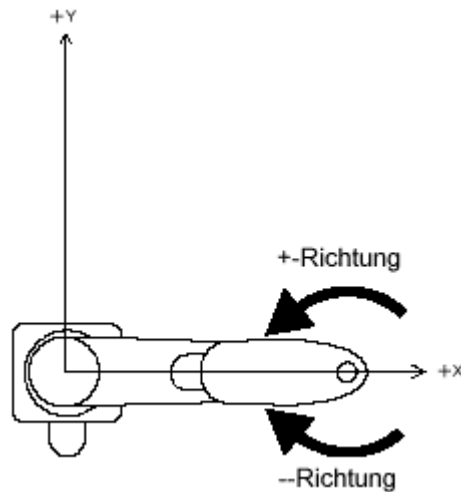


Abb. 4: Z-Achse des Roboterkoordinatensystems



**Abb. 5: U-Achse des Roboterkoordinatensystems**

Mit Hilfe von SPEL 95 können maximal 15 lokale Koordinatensysteme definiert werden. SPEL 95 koordiniert Roboter- und lokale Koordinatensysteme im Vorfeld durch Definieren der relativen Positionsbeziehung der lokalen Koordinatensysteme aus den Roboterkoordinatensystemen. Hierfür werden die ID-Nummern (1 bis 15) zugewiesen, die dann wiederum den Merkmalen der lokalen Koordinatensysteme zugewiesen werden.

Lokale Koordinatensysteme können auf zwei verschiedene Arten definiert werden. Die erste Möglichkeit ist das direkte Festlegen des Offset-Vektors des lokalen Koordinatensystems aus dem Referenzkoordinatensystem. Dies wird die „lokale Koordinatensystem-Spezifikationsmethode“ genannt. Das ist sehr bequem für alle Fälle, in denen die relative Position zwischen verschiedenen Koordinatensystemen im Vorfeld bekannt ist, und in Fällen, in denen die relative Position durch die Anwendermethode aufgefunden werden kann.

Die zweite Möglichkeit ist die Überprüfung der Positionsdaten zweier Punkte im Koordinatensystem und die Berechnung deren relativer Positionen in den Koordinatensystemen durch SPEL 95. Mit Hilfe von SPEL 95 kann jede der Methoden unter Verwendung der LOCAL-Befehle definiert werden. Die Offset-Vektoren werden mit der entsprechenden ID-Nummer des Koordinatensystems in SPEL 95 gespeichert.

### 1.1.3 Basiskoordinatensystem

Dies beschließt die Basiserklärungen der Roboter- und lokalen Koordinatensysteme, die bis hierher erklärt worden sind. Es gibt jedoch ein weiteres zu definierendes Koordinatensystem, welches für Spezialanwendungen mit Hilfe von SPEL 95 eingesetzt wird. Es wird „Basiskoordinatensystem“ genannt.

Unter normalen Bedingungen ist das Basiskoordinatensystem identisch mit dem Roboterkoordinatensystem, in dem alle Daten für die Offset-Vektoren im Roboterkoordinatensystem den Wert 0 haben. Mit SPEL 95 werden JOG-Vorschub und die Positionsdaten, für die keine Koordinatensystemmerkmale definiert sind, als Basiskoordinatensystemdaten behandelt. Kurz gesagt, wenn ein Basiskoordinatensystem definiert werden soll, ändern sich Punkte für alle Positionsdaten (und auch die Richtung des JOG-Einzugs), die bis zu diesem Punkt eingerichtet worden sind.

Das heißt, ein Basiskoordinatensystem sollte nur dann verwendet werden, wenn (1) die Befestigungsposition des Manipulators falsch ist und es notwendig ist, alle Positionsdaten (abhängig vom Grad der Abweichung), die bis zu diesem Punkt eingerichtet worden sind, neu einzustellen. Oder (2) wenn der Anwender ein Koordinatensystem frei definieren will, um dadurch das Roboterkoordinatensystem zu ersetzen, da er dieses nicht verändern kann. Und wenn der Anwender ein freies Koordinatensystem definiert, muss gewährleistet sein, dass er dieses korrekt steuern kann.

## 1.2 SPEL 95 Beschreibung der Positionsdaten

Mit SPEL 95 können Positionsdaten eingerichtet und registriert werden. Dazu wird die unten beschriebene Schreibweise verwendet.

- Festlegen einer Zielkoordinate für Anwendungsbefehle

Beispiel:           GO P10  
                   GO 100,-100,-  
                   50,45  
                   GO .PICK                   `Pick ist die Positionsmarke

- Speichern der Positionsdaten

Beispiel:           P1=P10  
                   P1=100,-100,-50,45  
                   .PICK=P10  
                   .PICK=100,-100,-50,45

### 1.2.1 Spezifikationsmethoden

Wie in den vorangehenden Beispielen dargestellt, gibt es zwei verschiedene Methoden zum Festlegen der Positionsdaten:

- Direkte numerische Spezifikation
- Positionsdaten-Ausdruck

Beschreibungen der Spezifikationsmethoden werden in einer Schreibweisen-Übersicht dargestellt.

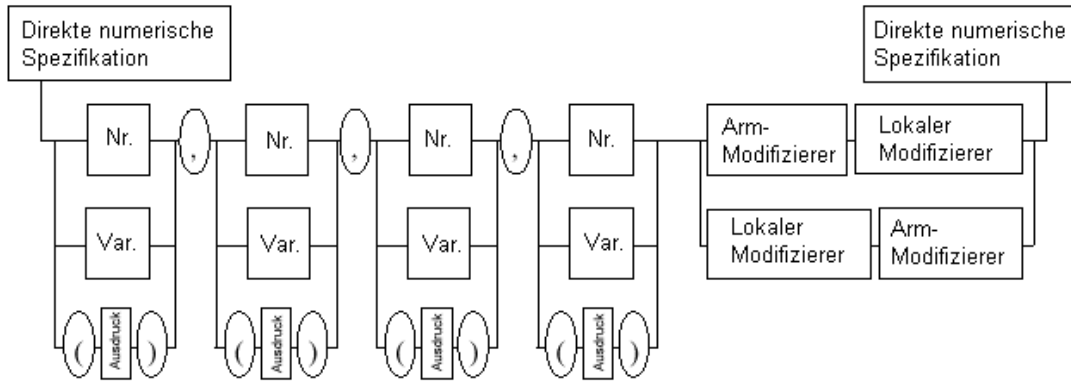
Eine Schreibweisen-Übersicht beinhaltet Diagramme, um die Verbindung zwischen Ausdrücken darzustellen.

Rechteckige Kästchen werden an anderer Stelle definiert, und ovale Rahmen beinhalten SPEL 95-Schlüsselwörter, -Zeichen und -Teile, die nicht abgetrennt werden können.

Pfeile treten in Bereichen auf, in denen Wiederholungen erfordern, dass sich entgegen der normalen Richtung bewegt wird.

▪ **Direkte numerische Spezifikation**

Beispiel: GO 100,-100,-50,45 @L /3  
 P10=100,-100,-50,45 /L @3



(Var. = Variable, Nr. = Nummer)

Ein lokaler Modifizierer legt die Nummer des lokalen Koordinatensystems fest.

Änderungen am festgelegten lokalen Koordinatensystem, wenn @ definiert ist.

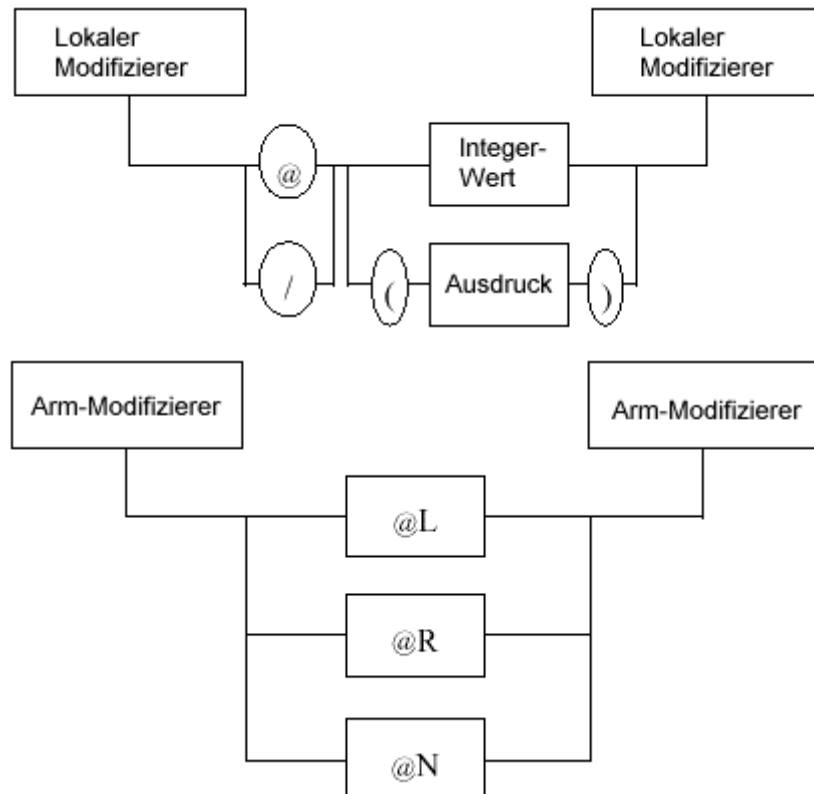
Ersetzt nur Merkmale der Daten, wenn / definiert ist.

Ein Armmodifizierer definiert die Richtung

L: Linker Arm

R: Rechter Arm

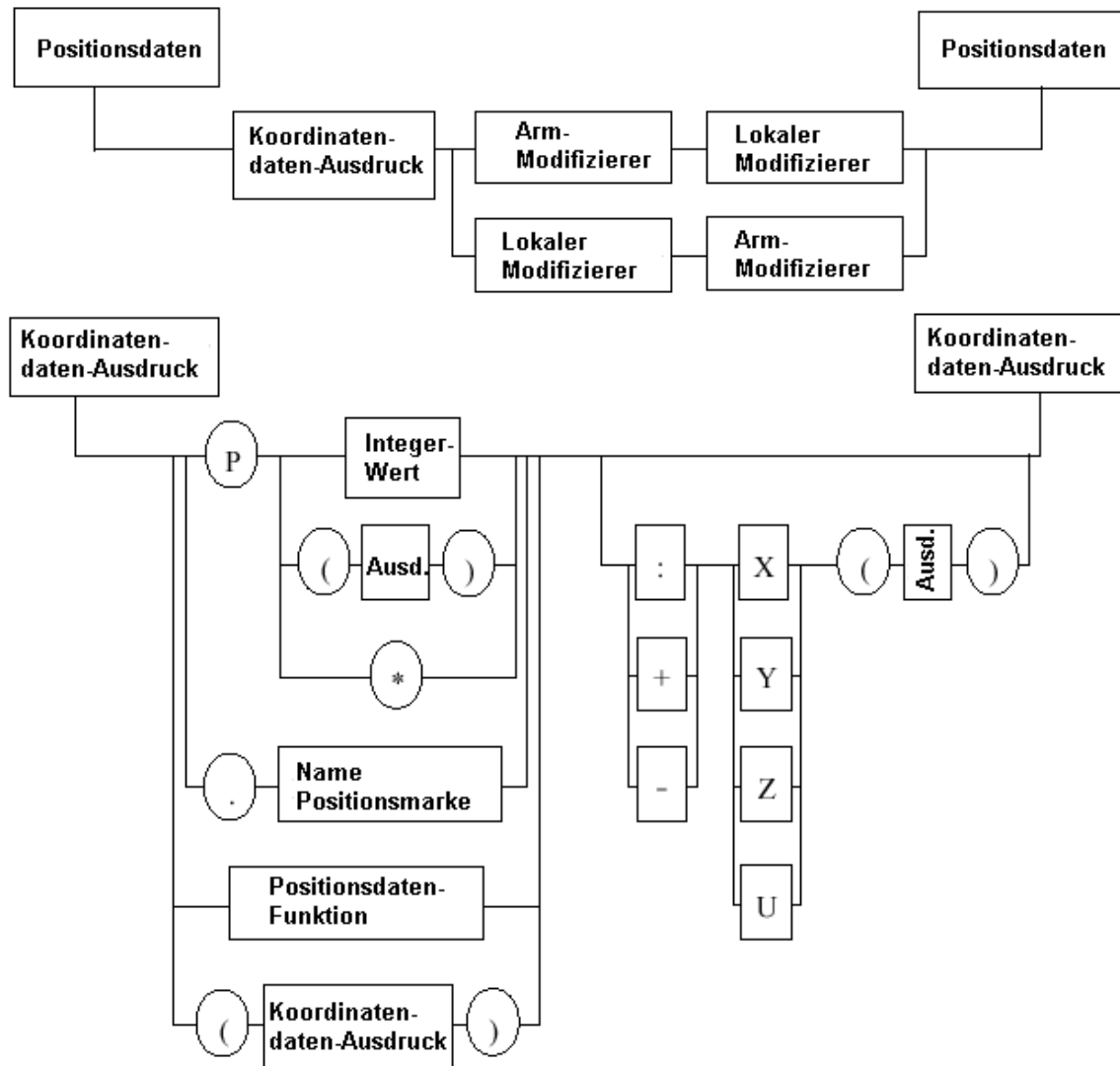
N: Standardausrichtung





### ▪ Positionsdaten-Ausdrücke

Beispiel:      GO P1,P10,@3  
                   GO PULSE (a,b,c,d)  
                   JUMP .PICK,.PLACE  
                   P10=P1+P2  
                   P10=P\*+X(100) :Z(-50)



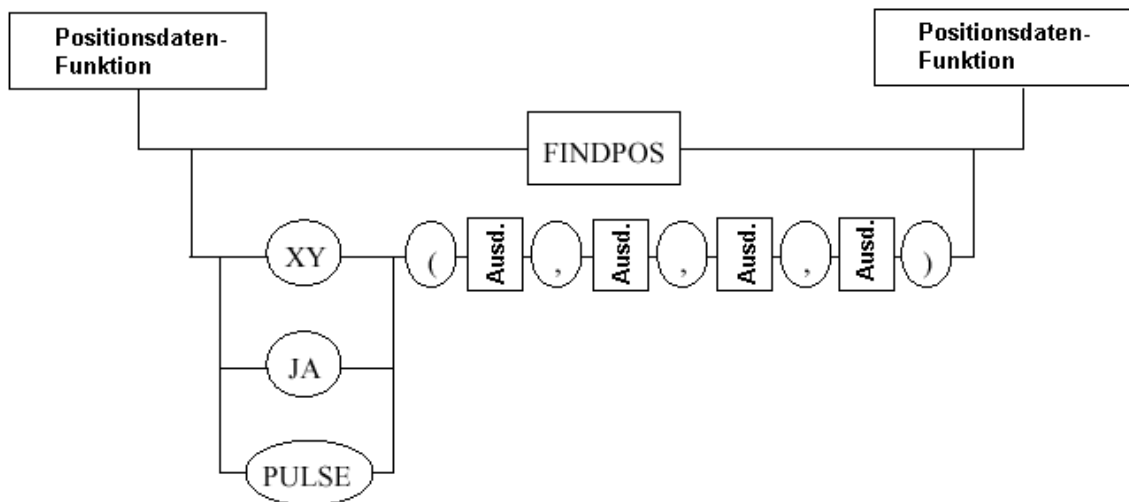
(Ausd. = Ausdruck)

P\* verweist auf die aktuelle Position.

:X(), :Y(), :Z() und :U() sind definierte Werte. Jedes Element wird ersetzt.

Die definierten Werte für +X(), +Y(), +Z() und +U() werden zu jedem aktuellen Element addiert.

Die definierten Werte für -X(), -Y(), -Z() und -U() werden von jedem aktuellen Element subtrahiert.



(Ausd. = Ausdruck)

XY() stellt den definierten Wert als Koordinate ein und wiederholt ihn.

JA() stellt den definierten Wert als Gelenkpunkt ein (Winkel oder Distanz), konvertiert ihn zur Koordinate und wiederholt ihn.

PULSE() stellt den definierten Wert als Puls ein, konvertiert ihn zur Koordinate und wiederholt ihn. (Falls jedoch PULSE() direkt als Betriebsbefehlparameter definiert wird, arbeitet das System mit dem Pulswert als Ziel, ohne ihn zu konvertieren.)

FINDPOS wiederholt die gespeicherte Koordinate in dem Moment, wo die FIND-Bedingungen zutreffen.

## 1.3 Koordinatensystem und JOG-Modus

Die drei nachfolgend aufgeführten JOG-Modi stehen nur zum Einrichten der Roboter zur Verfügung:

JOG-Verfahren im Basis-Koordinatensystem

JOG-Verfahren im Tool-Koordinatensystem

JOG-Verfahren im Joint-Koordinatensystem

Jede dieser Methoden kann nur zum Einrichten benutzt werden, und jede zeigt die Beziehung zwischen der Richtung der JOG-Taste und der Manipulator-Betriebsrichtung auf. Das System ist für jede der Achsen (X, Y, Z und U) mit +- und --Tasten ausgerüstet. Der ausgewählte JOG-Modus bestimmt, wie sich der Manipulator bewegen wird, wenn diese Tasten gedrückt werden.

Im JOG-Verfahren des Basis-Koordinatensystems verfährt der Manipulator basierend auf den Koordinaten, die durch die parallele Übersetzung des Basiskoordinatensystems (im Normalfall das Roboter-Koordinatensystem) ankommen, die ihren Ursprung am Ende des Greifers haben. Die JOG-Tasten entsprechen den Richtungen der Achsen im Basiskoordinatensystem. Wenn Sie, z. B., die X+-JOG-Taste drücken, verfährt der Manipulator in Richtung der X-Achse des Basiskoordinatensystems. Wenn Sie die U+-JOG-Taste drücken, dreht sich der Manipulator um die U-Achse. (Rechtsdrehende Schraubbewegung in Bezug auf die +-Richtung der Z-Achse in dem Koordinatensystem, dessen Ursprung am Ende des Greifers ist.)

Im JOG-Verfahren des Tool-Koordinatensystems verfährt der Manipulator basierend auf dem Koordinatensystem, das seinen Ursprung an der Spitze des Greifers hat. Wenn Sie das Tool-Koordinatensystem ausgewählt haben, wird die Richtung jeder Bewegung, die durch die JOG-Taste ausgeführt wird, durch die Ausrichtung des Greifers festgelegt. Auch wenn der Greifer rotiert, wird das JOG-Verfahren durch feste Koordinaten für die Ausrichtung des Greifers ausgeführt. Es kann daher für Düsen oder ähnliches, mit gleicher Richtung, eingesetzt werden. Wenn Sie, z. B., die X+-JOG-Taste drücken, verfährt der Manipulator in +-Richtung der X-Achse des Tool-Koordinatensystems. Also ändert sich die Richtung abhängig von der Greiferposition. Diese JOG-Methode eignet sich für einrichtungsrelevante Aufgaben, wenn die relativen Positionen des einzurichtenden Objekts und des Greifers wichtig sind.

Das JOG-Verfahren im JOINT-Koordinatensystem unterscheidet sich von den oben beschriebenen Methoden. Es wird nicht für den Betrieb in rechtwinkligen Koordinatensystemen eingesetzt. In diesem Modus wird jedes Gelenk einzeln betrieben, wobei jede JOG-Taste jeweils einem Gelenk entspricht.

Die Beziehung zwischen Gelenk und JOG-Taste wird in nachstehender Tabelle verdeutlicht:

JOG-Taste	entsprechendes Gelenk
X	Gelenk 1 (J1)
Y	Gelenk 2 (J2)
Z	Gelenk 3 (J3)
U	Gelenk 4 (J4)

Die Gelenke 1-4 unterscheiden sich in Abhängigkeit vom Manipulatormodell. Weitere Informationen entnehmen Sie bitte dem entsprechenden Manipulator-Handbuch.

Die +- und -Richtungen der JOG-Tasten entsprechen den +- und -Richtungen der Gelenke.

# Kapitel 2

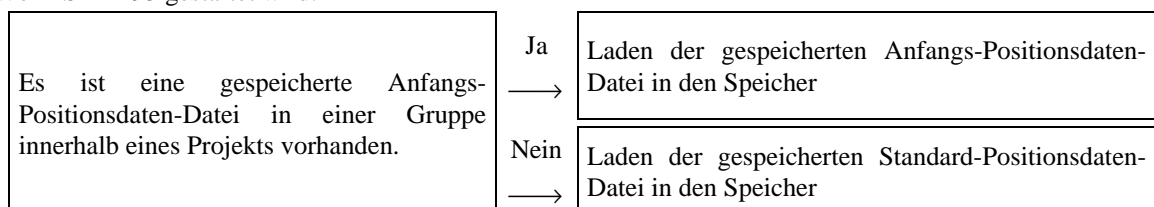
## Einrichten

Stellungen oder Positionen, die Ziel für die Manipulatorbewegung sein sollen, werden „Positionsdaten“ genannt. Das Schreiben bzw. Ablegen dieser Positionsdaten in den Speicher der Steuerung wird „Einrichten“ genannt. Die Positionsdaten werden in einer Datei gespeichert. Diese wird „Positionsdaten-Datei“ genannt. Zum Betrieb eines Manipulators wird zumindest eine gespeicherte Positionsdaten-Datei benötigt. Sind mehrere Positionsdaten-Dateien für einen einzelnen Manipulator vorhanden, werden die Positionsdaten-Dateien in verschiedene „Gruppen“ unterteilt.

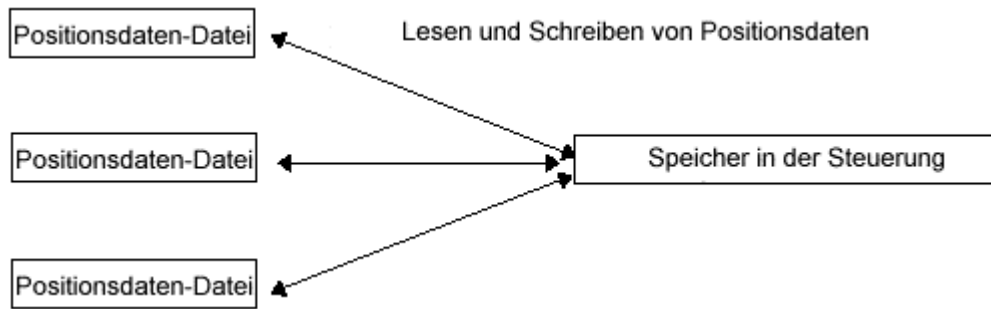
### 2.1 Positionsdaten-Datei und Einrichten

Die Standard-Positionsdaten-Datei wird automatisch in den Speicher der Steuerung geschrieben, wenn SPEL 95 gestartet und ein Projekt geöffnet wird. Sollte jedoch eine gespeicherte Anfangs-Positionsdaten-Datei in einer Gruppe innerhalb eines Projekts vorhanden sein, wenn SPEL 95 gestartet wird, ist das die Datei, die in den Speicher geschrieben wird.

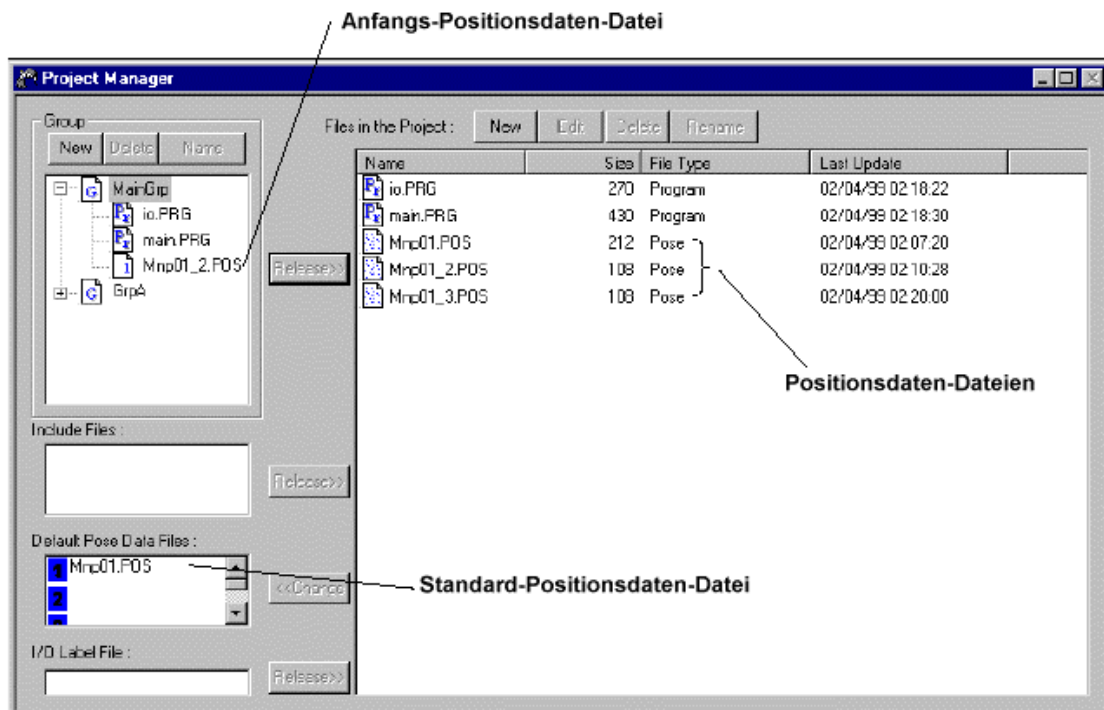
Wenn SPEL 95 gestartet wird:



Das Einrichten entspricht dem Schreiben der Positionsdaten aus der Positionsdaten-Datei in den Speicher der Steuerung. Es ist wichtig, die Spur zu behalten, um genau zu wissen, welche Positionsdaten aus welcher Datei in den Speicher geladen wird, wenn mehr als eine Positionsdaten-Datei in einem Projekt gespeichert ist.



Die Positionsdaten-Dateien enthalten folgendes:



**Anfangs-Positionsdaten-Datei:** Die Positionsdaten-Datei ist in einer Gruppe gespeichert. Eine Anfangs-Positionsdaten-Datei wird vom Anwender gespeichert.

**Standard-Positionsdaten-Datei:** Die Positionsdaten-Datei wird automatisch erzeugt, wenn ein Projekt generiert wird.

**[Dateien im Projekt]-Verzeichnis:** Alle Positionsdaten-Dateien, die in einem Projekt gespeichert sind, dessen [Dateityp] als „POSE“ definiert ist. Damit eine der Positionsdaten-Dateien aktiviert wird, muss sie entweder als Anfangs-Positionsdaten-Datei oder als Standard-Positionsdaten-Datei gespeichert sein.

## 2.1.1 Laden einer Positionsdaten-Datei

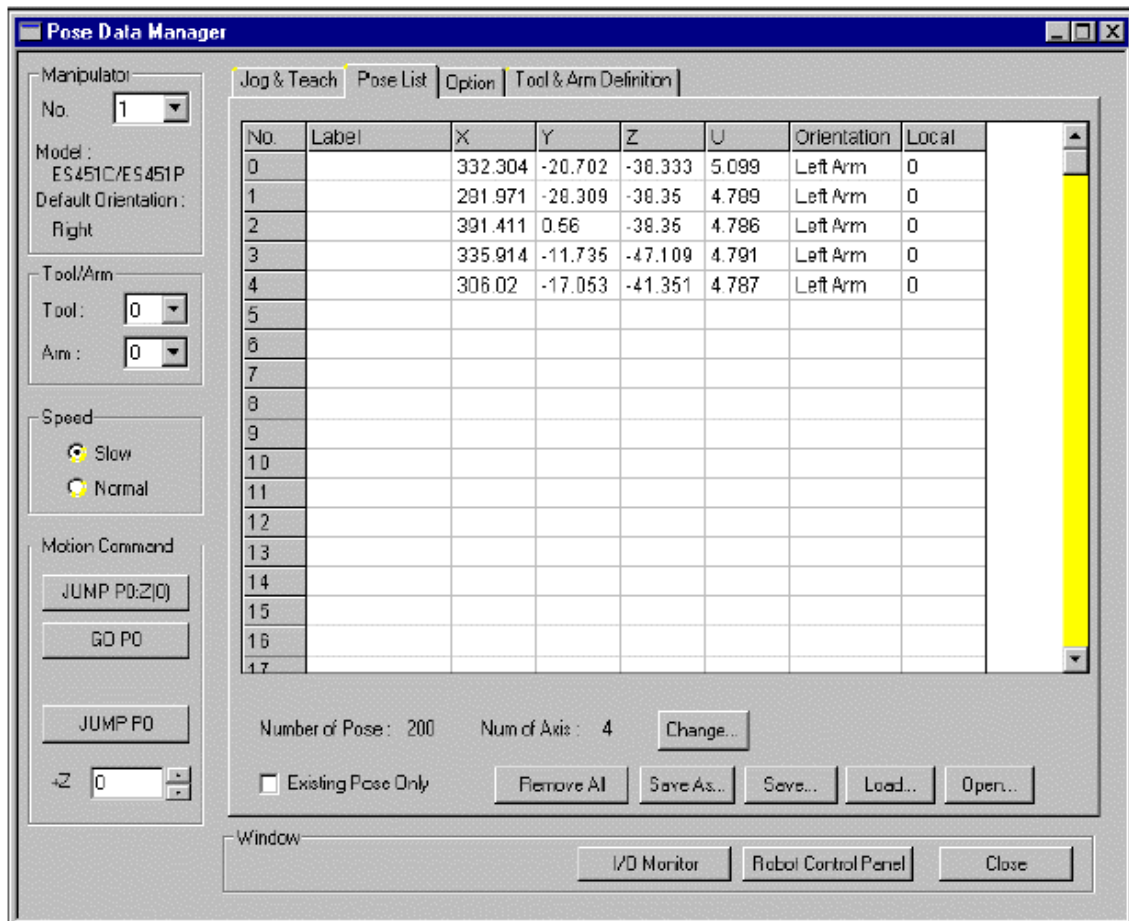
Das Einrichten kann mit Hilfe des [Pose Data Manager]- (Positionsdaten-Manager)-Fensters ausgeführt werden. Im [Pose Data Manager]-Fenster können die im Systemspeicher befindlichen Positionsdaten bearbeitet werden. Sollten zwei oder mehr Positionsdaten-Dateien gespeichert sein, müssen Sie eine definieren, die eingerichtet werden soll. Das Definieren einer Datei zum Einrichten muss dann nicht erfolgen, wenn nur eine Datei vorhanden ist oder wenn eine Datei bereits in den Speicher geladen wurde.

Klicken Sie auf den Knopf <Pose List> (Positionsdatenliste) in der Werkzeugleiste des [Pose Data Manager]-Fensters, um sich die [Pose List], die Liste aller verfügbaren Positionsdaten, anzeigen zu lassen.

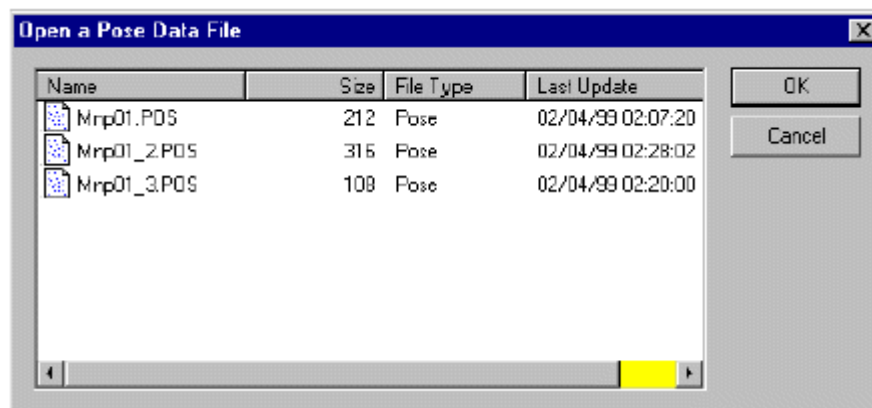
Klicken Sie auf den Knopf <Load...> (Laden) im [Pose List]-Menü.

PD

<Pose List> Knopf



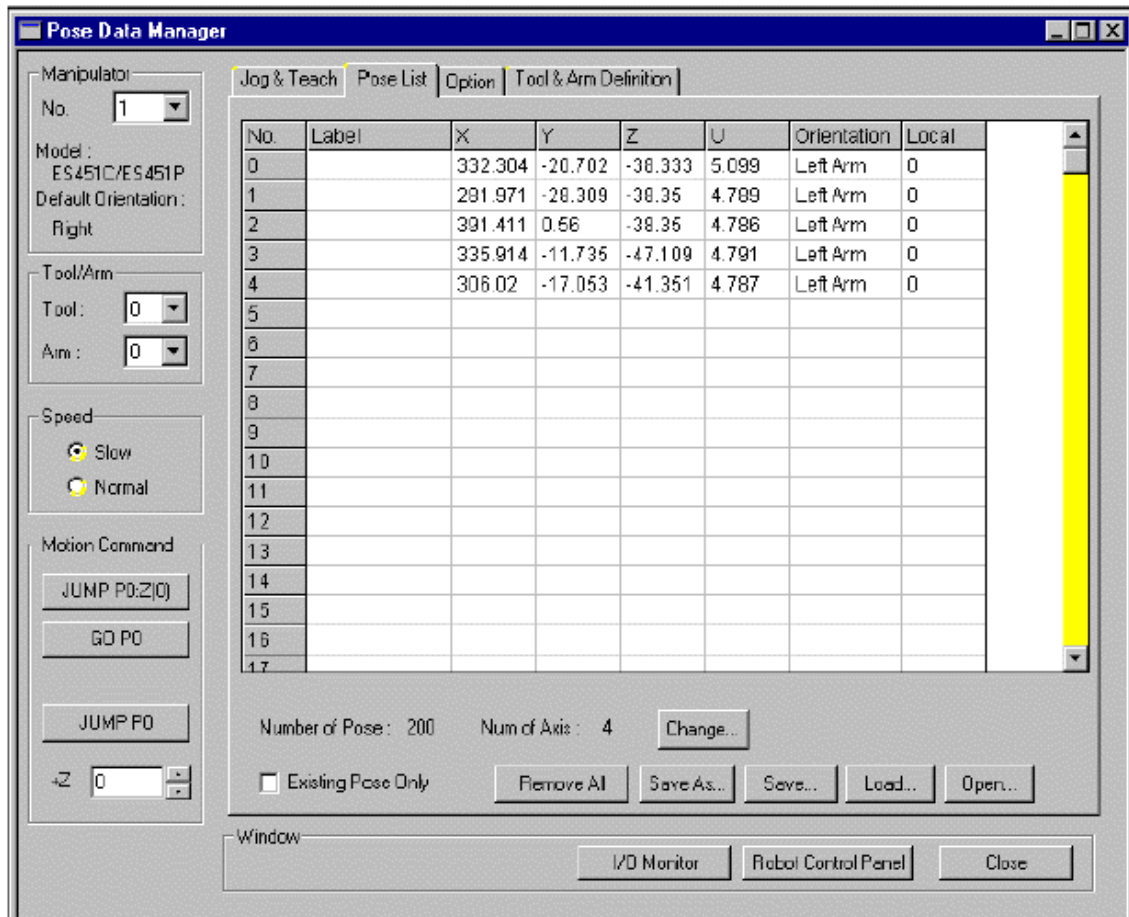
Wählen Sie eine Datei und klicken Sie auf <OK>.



Die Positionsdaten der ausgewählten Positionsdaten-Datei werden in den Systemspeicher geschrieben. Jetzt können die Positionsdaten im [Pose Data Manager]-Fenster bearbeitet werden.

## 2.1.2 Speichern der Positionsdaten (in eine Positionsdaten-Datei) nach dem Einrichten

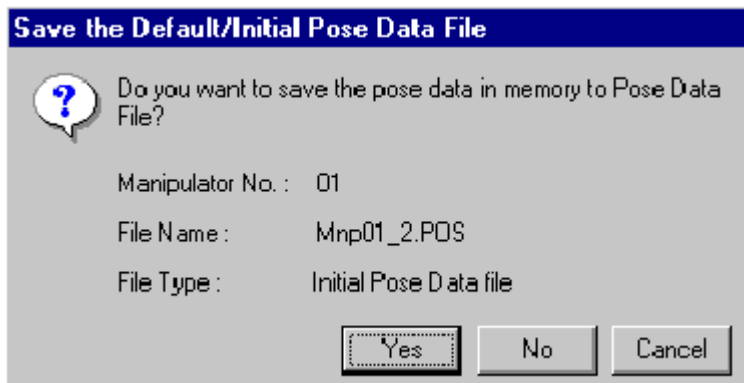
Die Positionsdaten, die während des Einrichtens in den Speicher geschrieben wurden, verschwinden sofort, wenn SPEL 95 beendet wird. Daher müssen Sie sicherstellen, dass Sie Ihre Daten nach dem Einrichten speichern.



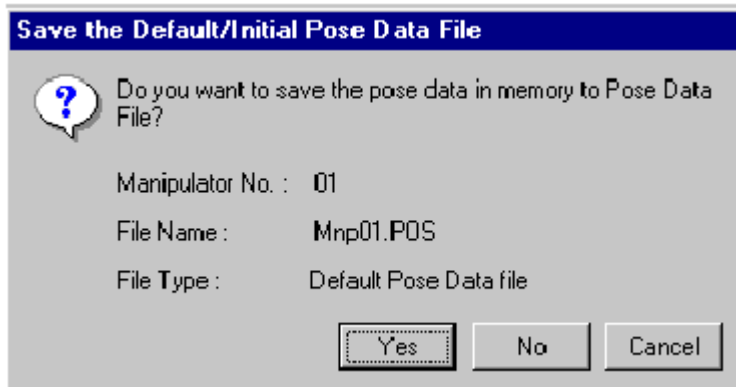
### ▪ Speichern der Positionsdaten in die Anfangs- oder Standard-Positionsdaten-Datei

Speichern Sie die im Speicher befindlichen Positionsdaten als Anfangs- oder Standard-Positionsdaten-Datei. Klicken Sie entweder auf den <Save>- (Speichern)-Knopf im [Pose List]-Menü oder auf den <Close>- (Schließen)-Knopf im [Pose Data Manager]-Fenster.

Eine Meldung, wie unten dargestellt, erscheint, wenn Sie die Positionsdaten als Anfangs-Positionsdaten-Datei speichern. Klicken Sie auf <Yes> (Ja).

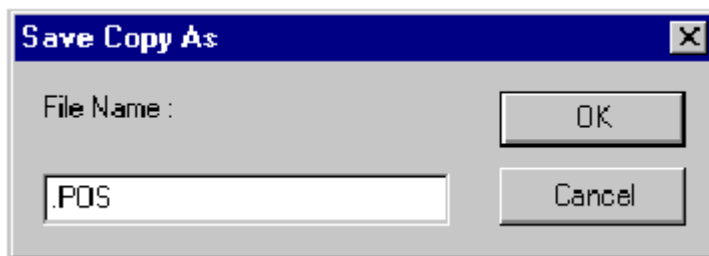


Eine Meldung, wie unten dargestellt, erscheint, wenn Sie die Positionsdaten als Standard-Positionsdaten-Datei speichern. Klicken Sie auf <Yes> (Ja).



- **Speichern der Positionsdaten in eine eigene Positionsdaten-Datei**

Sie können die Positionsdaten auch in eine eigene Positionsdaten-Datei schreiben. Klicken Sie dafür auf den <Save As...>-(Speichern unter...)-Knopf im [Pose List]-Menü. Geben Sie im erscheinenden Meldfenster einen entsprechenden Namen für die Datei ein. Die Endung muss .POS lauten. Klicken Sie dann auf den <OK>-Knopf.



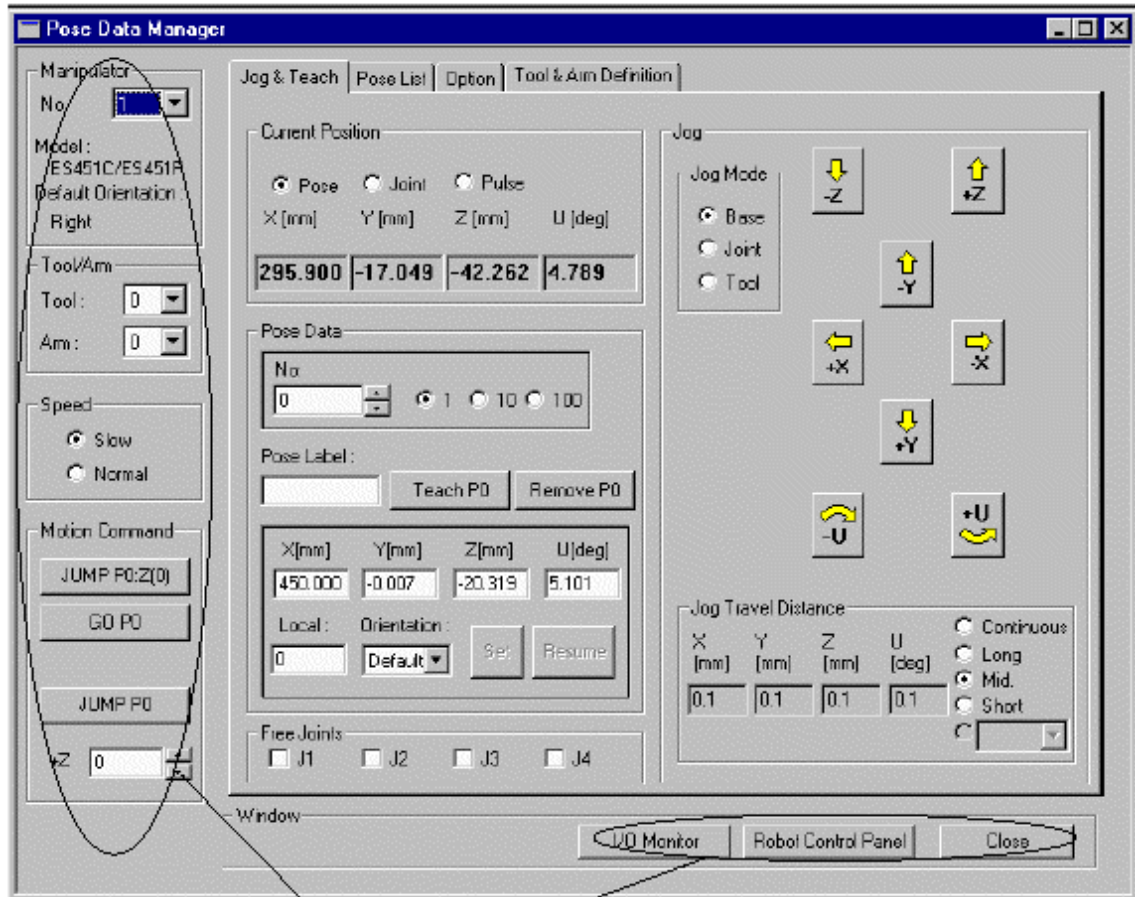
Die Positionsdaten aus dem Speicher werden dann in die von Ihnen angegebene Datei geschrieben.



## 2.2 Bedienung per JOG & Teach-Menü

Wenn Sie die [Tool]- (Werkzeug)-[JOG&Teach]- (JOG&Einrichten)-Befehle ausführen oder auf den <JOG&Teach>-Knopf in der Taskleiste klicken, erscheint das [JOG&Teach]-Menü im [Pose Data Manager]-Fenster.

JT  
<JOG&Teach>-  
(JOG&Einrichten)-  
Knopf



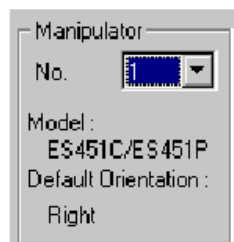
Allgemeine Anzeige im [Pose Data Manager]-Fenster

### 2.2.1 Allgemeine Anzeige im [Pose Data Manager]-Fenster

Allgemeine Bedienungen für jedes Menü im [Pose Data Manager]-Fenster werden nachstehend erklärt.

- **[Manipulator]-Group Box ([Manipulator Gruppenauswahl])**

Benutzen Sie diese Gruppenauswahl, um aus den angeschlossenen Manipulatoren denjenigen auszuwählen, mit dem Sie arbeiten wollen. Selektieren Sie dafür einen angeschlossenen Manipulator aus dem Auswahlfeld in der Kombobox [No.] (Nr.).

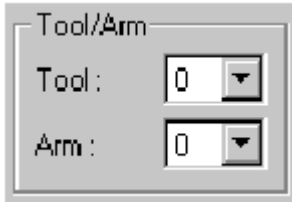


Um eine Manipulatornummer zu vergeben, benutzen Sie bitte die Befehle [Setup]-[Robot Manipulator] (Einrichten –Manipulator).

### ▪ [Tool/Arm]-Group Box ([Werkzeug/Arm Gruppenauswahl])

Diese Gruppenauswahl wird benutzt, um Werkzeuge und Arme auszuwählen, die vom Anwender eingerichtet worden sind. Funktionen dieser Werkzeuge und Arme können über das Menü [Tool/Arm] eingerichtet werden.

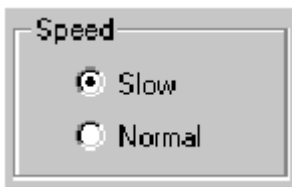
In der Online-Hilfe erhalten Sie weitere Informationen über die Einstellungen der Tool/Arm-Auswahl.



### ▪ [Speed]-Group Box ([Geschwindigkeit Gruppenauswahl])

In dieser Gruppenauswahl können Sie die Geschwindigkeit während der JOG&Teach-Einstellungen einrichten. Es gibt zwei verschiedene Geschwindigkeitsstufen: <Slow> (Langsam) und <Normal> (Normal). Klicken Sie auf den entsprechenden Auswahlknopf, um die entsprechende Geschwindigkeit einzustellen.

Die Geschwindigkeit beträgt im langsamen Modus 1% und im normalen Modus 5% der maximalen Betriebsgeschwindigkeit des Manipulators, wobei der eingestellte Wert abhängig vom verwendeten Manipulator differieren kann. Die Standardeinstellung ist <Slow>.



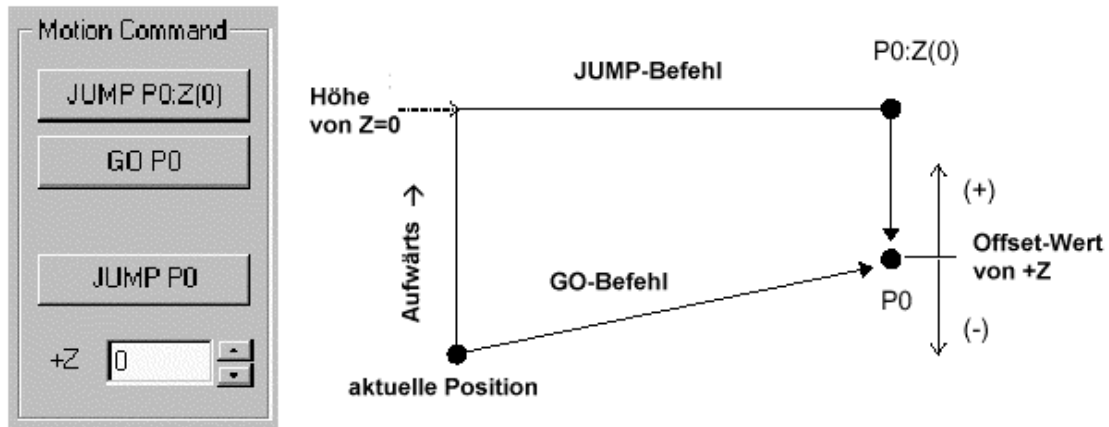
#### WARNUNG

Während der JOG&Teach-Einstellungen kann sich der Manipulator unter Umständen in unerwartete Richtungen bewegen und dabei Peripheriegeräte beschädigen, wenn ein Fehler in den Positionsdaten vorhanden ist. Im Anfangsstadium des Betriebes sollte daher der Geschwindigkeitsmodus auf <Slow> eingestellt sein, und die Bewegung des Manipulators sollte genauestens überwacht werden.

### ▪ [Motion Command]-Group Box ([Bewegungsbefehl Gruppenauswahl])

<JUMP P#:Z(0)>Knopf	Bewegt sich über P# (Z(0) Position) mit dem JUMP-Befehl
<GO P#>Knopf:	Bewegt sich zu den Koordinaten, die per GO-Befehl eingegeben werden.
<JUMP P#>Knopf:	Bewegt sich zu den Koordinaten, die per JUMP-Befehl eingegeben werden.
<+Z>Auswahlfenster:	Einrichten des Offset-Wertes für Z, der in JUMP P#-Befehlen benötigt wird. Der Wert ist der Offset von P# mit Richtung Z. Ist der Wert positiv (+), wird zu einer höheren als der Zielposition gesprungen. Ist der Wert negativ (-), wird zu einer niedrigeren Position gesprungen.

Bewegungen der Manipulorränder mit Hilfe von Befehlen



▪ **[Window]-Group Box ([Fenster Gruppenauswahl])**

Knöpfe in diesem Menü werden benötigt, um andere Menüs zu öffnen und um das [Pose Data Manager]-Fenster zu schließen.



<I/O-Monitor>(E/A-Monitor)-  
Knopf

Öffnet das [E/A-Monitor]-Menü.

<Robot Control Panel>-  
(Roboter-Steuerungs-Menü)-  
Knopf

Öffnet das [Roboter-Steuerungs]-Menü.

<Close>-(Schließen)-Knopf

Schließt das [Pose Data Manager]-Fenster.

Erscheint die folgende Meldung, wenn Sie ein Projekt durch Drücken auf den <No>-(Nein)-Knopf im vorausgegangenen Menü schließen oder ändern, bedeutet dies, dass die Werte der Positionsdaten im Speicher von den Anfangs- oder Standard-Positionsdaten abweichen.

Erscheint die folgende Meldung, bedeutet dies, dass die Werte der Positionsdaten im Speicher von den Anfangs- oder Standard-Positionsdaten abweichen. Klicken Sie auf den <Save>-(Speichern)-Knopf, um die Werte der Positionsdaten im Speicher zu speichern. Klicken auf den <Purge>-(Verwerfen)-Knopf verwirft die Daten.

Die eingerichtete Position bleibt in den Positionsdaten im Speicher und kann auch dann noch benutzt werden, wenn der <Purge>-Knopf gedrückt wurde. Die Position geht jedoch dann verloren, wenn SPEL 95 neu gestartet oder ein anderes Projekt geöffnet wird. In diesen Fällen werden die Anfangs- oder Standard-Positionsdaten in den Speicher geladen.

## Anfangs-Positionsdaten

**Save the Default/Initial Pose Data File**

The pose data in memory is not same as Pose Data File.  
Do you want to save this pose data in memory to Pose Data File?

Manipulator No. : 01

File Name : Mnp01\_2.POS

File Type : Initial Pose Data file

## Standard-Positionsdaten

**Save the Default/Initial Pose Data File**

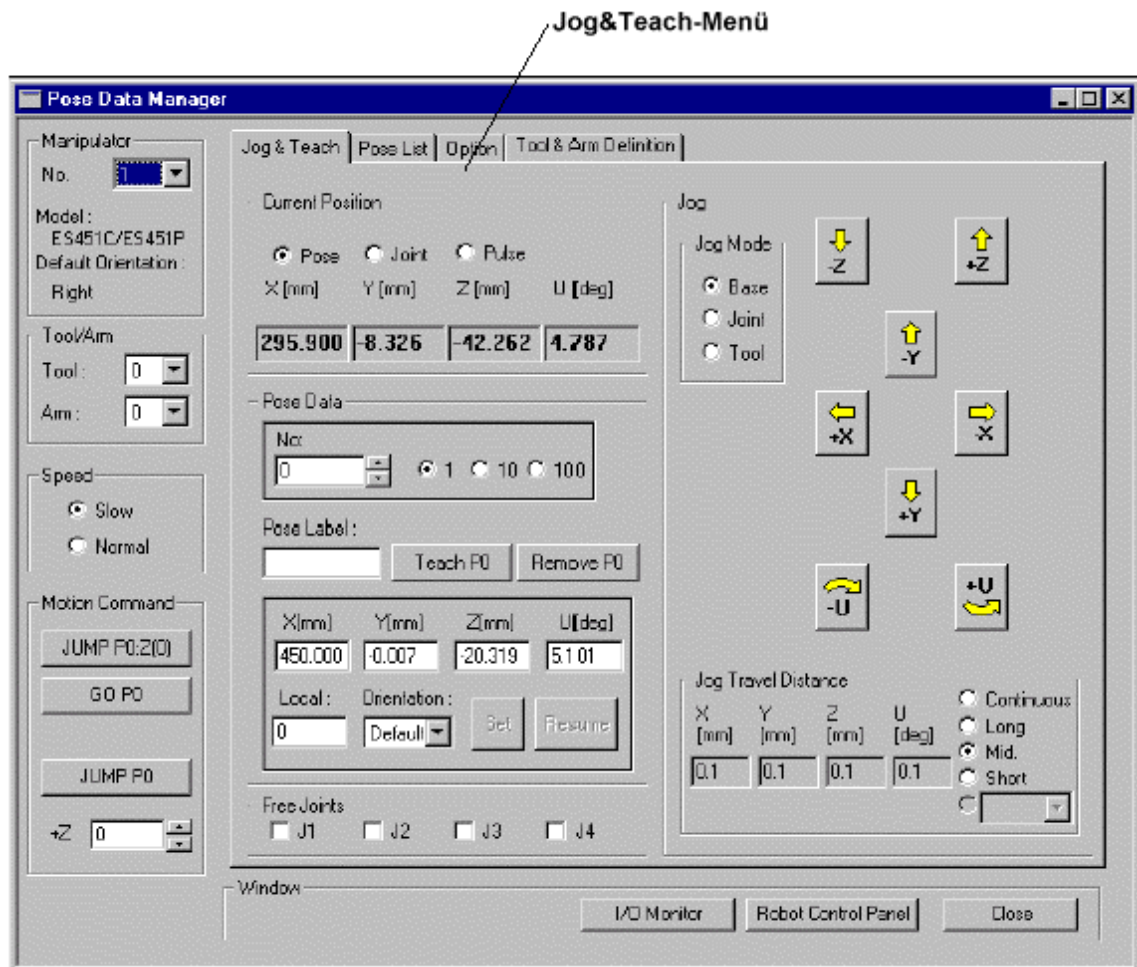
The pose data in memory is not same as Pose Data File.  
Do you want to save this pose data in memory to Pose Data File?

Manipulator No. : 01

File Name : Mnp01.POS

File Type : Default Pose Data file

## 2.2.2 Bedienung per JOG&amp;Teach-Menü



### ▪ [Current Position]-Group Box ([Aktuelle Position Gruppenauswahl])

Sie wechseln zwischen den einzelnen JOG-Modi, indem Sie die entsprechenden Optionsknöpfe anklicken. Dadurch wird die aktuelle Position in jedem Koordinatensystem angezeigt.

- Pose (Position): Die aktuelle Position im Basis-Koordinatensystem (XY-Koordinatensystem) wird in [mm] angezeigt. Die Einheit für die U-Achse wird in [deg] (Grad) dargestellt.
- Joint (Gelenk): Die aktuelle Position (JOINT-Koordinatensystem) jedes Gelenks wird für die Rotationsachse in [deg] oder für die lineare Achse in [mm] dargestellt.
- Pulse (Puls): Die Pulslage jedes Gelenks wird angezeigt.

### ▪ [Pose Data]-Group Box ([Positionsdaten Gruppenauswahl])

Verschiedene Funktionen, wie z.B. das Einrichten (Registrieren) und Löschen von Positionsdaten, können von hier aus vorgenommen werden.

- [No.]-[Nr.]-Auswahloption Hier wählen Sie eine Positionsdaten-Nummer. Die drei Optionsknöpfe auf der rechten Seite zeigen die Größe der Nummer, die durch einen einzigen Klick verändert wird. Sie können auch direkt eine Nummer eingeben.
- [Pose Label]-[Positionsmarke]-Option Wurde ein Name für die definierte Positionsdaten-Datei vergeben, erscheint dieser hier. Sie können auch direkt einen Namen in das Feld eingeben. Klicken Sie nach Eingabe einer Positionsmarke auf den <Set>-[Einstellen]-Knopf.
- <Teach P#>-[P#-Einrichten]-Knopf Richtet eine Positionsdaten-Nummer, die in das Auswahlfeld [No.] eingegeben wurde, an der aktuellen Position ein.

<Remove P#>- (P# Entfernen)-Knopf	Löscht die aus dem Auswahlfeld [No.] gewählten Positionsdaten.
[MDI]-Auswahloption	Zeigt die Koordinaten der per [No.]-Auswahlfeld ausgewählten Positionsdaten an. Ein Wert kann auch direkt eingegeben werden. Klicken Sie nach Eingabe eines Wertes auf den <Set>- (Einstellen)-Knopf. Das Feld bleibt leer, wenn keine Daten eingegeben werden.
[Orientation]- (Ausrichtungs)-Auswahlfeld	Auswahl des rechten oder des linken Arms oder des Standardwerts. Die [Orientation] (Ausrichtung) ist nur bei horizontalen Knickarm-Robotern (vom Typ SCARA) wirksam.
<Set>- (Einstellen)-Knopf	Legt die Daten fest, die in der MDI-Auswahloption eingestellt wurden. Der Knopf ist nur dann aktiv, wenn Daten eingegeben oder geändert worden sind.
<Resume>- (Fortsetzen)-Knopf	Stellt die ursprünglichen Daten wieder her. Der Knopf ist nur dann aktiv, wenn Daten eingegeben oder geändert worden sind.

#### ▪ [Free Joints]- (Freie Gelenke)-Gruppenauswahl

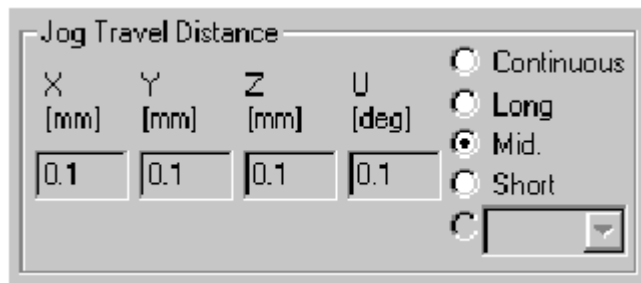
Die Gelenke des Manipulators können in den Status „servo-frei“ geschaltet werden.

Markieren Sie das Optionsfeld des Gelenks, welches den Status „servo-frei“ erhalten soll. Das Gelenk mit Status „servo-frei“ kann manuell bewegt werden. Diese Funktion wird für das direkte Einrichten (Teachen) verwendet.



#### ▪ [JOG Travel Distance]- (JOG-Bewegungsstrecke)-Gruppenauswahl

Diese Gruppenauswahl wird benutzt, um die Strecke einzustellen, die verfahren werden soll, wenn der JOG-Knopf angeklickt wird.



<Continuous>- (Ununterbrochen)-Optionsfeld	Während der JOG-Button gedrückt wird, bewegt sich das Gelenk ununterbrochen.
<Long>- (Lang)-Optionsfeld	Wird der JOG-Knopf einmal angeklickt, bewegt sich das Gelenk um 1mm*.
<Mid>- (Medium)-Optionsfeld	Wird der JOG-Knopf einmal angeklickt, bewegt sich das Gelenk um 0,1mm*.
<Short>- (Kurz)-Optionsfeld	Wird der JOG-Knopf einmal angeklickt, bewegt sich das Gelenk um 0,03mm*.

\*: Das sind Standardwerte. Wenn Sie die Standardwerte ändern wollen, können Sie dies mit Hilfe des Menüs [Option] im [Pose Data Manager].

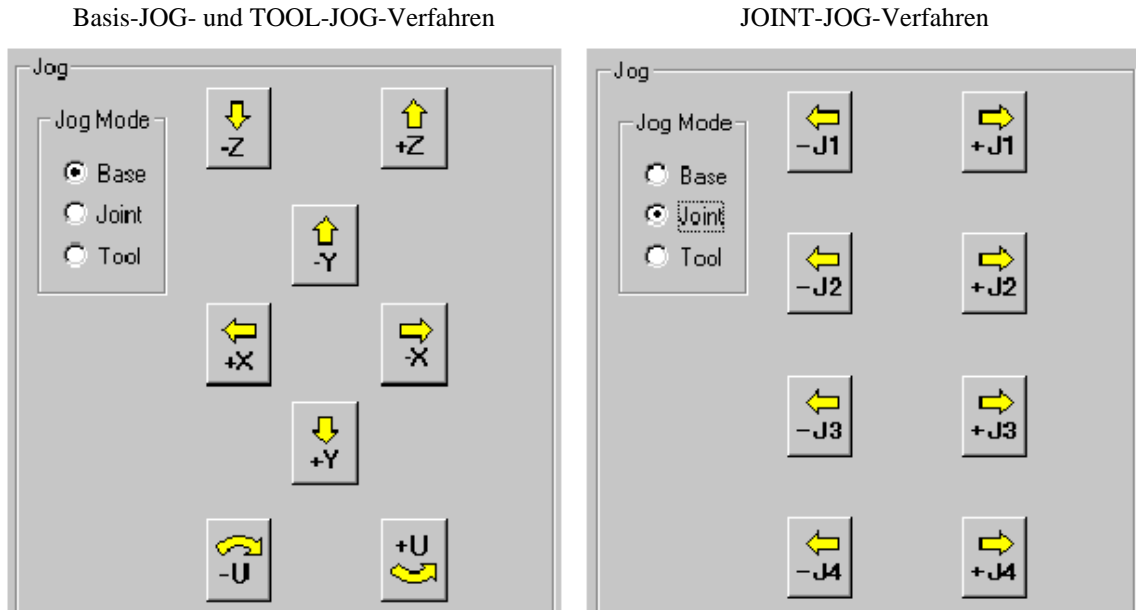
Sie können optionale Bewegungsstrecken mit Hilfe des Menüs [Option] im [Pose Data Manager] einrichten. Wird die Bewegungsstrecke eingerichtet, werden die Namen der Marken in der darunter stehenden Kombobox angezeigt. Wenn Sie aus dem Auswahlfeld eine gewünschte Entfernung auswählen, bewegt sich das Gelenk jedesmal, wenn Sie einmal auf den JOG-Knopf klicken, in der gewünschten Entfernung.

### ▪ [JOG]-Gruppenauswahl

Wird der JOG-Knopf gedrückt, wird vom Manipulator eine JOG-Tätigkeit ausgeführt. Ein Klick ist ausreichend, damit das Gelenk die in der [JOG Travel Distance]-[JOG-Bewegungsstrecke]-Gruppenauswahl festgelegte Entfernung zurücklegt.

Das Koordinatensystem im JOG-Verfahren wird über die Optionsknöpfe in der [JOG Mode]-[JOG Modus]-Gruppenauswahl eingestellt. Werden die Einstellungen geändert, ändern sich die JOG-Knöpfe automatisch.

Die nachstehenden Abbildungen zeigen die Monitore, wenn SCARA-Roboter eingesetzt werden.



- <Base>-[Basis]-Optionsschalter      Aktiviert die JOG-Tätigkeit im Basis-Koordinatensystem des Manipulators.
- <Joint>-[Gelenk]-Optionsschalter      Aktiviert eine JOG-Tätigkeit für das entsprechende Gelenk.
- <Tool>-[Werkzeug]-Optionsschalter      Aktiviert eine JOG-Tätigkeit im Tool-Koordinatensystem.

Die Anzeigen sind so konzipiert, dass sie für alle Manipulator-Typen zutreffen. Eine aktuelle Anzeige kann daher, abhängig vom eingesetzten Manipulator, von den oben gezeigten Darstellungen abweichen.



## 2.3 Einrichten von Positionsdaten in Manipulatoren

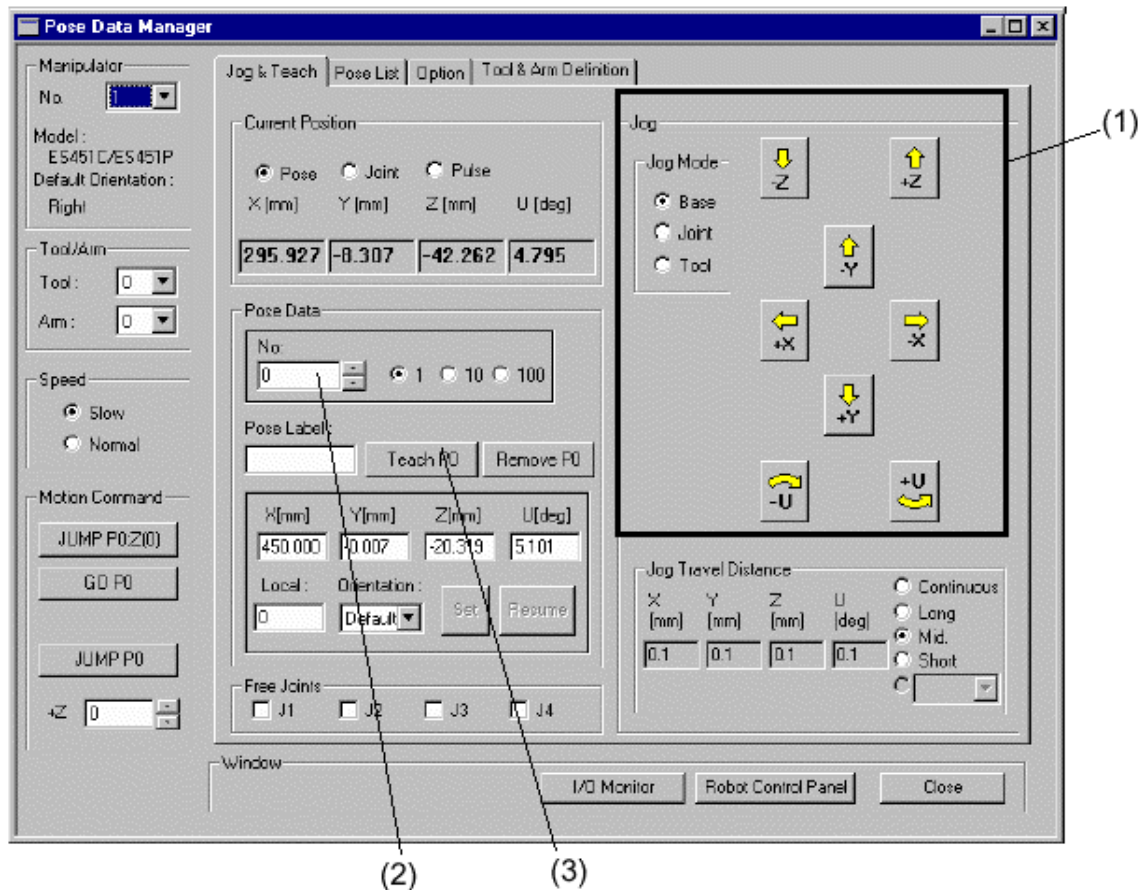
Nachfolgend werden die verschiedenen Methoden zum Einrichten der Positionsdaten für die Manipulatoren aufgelistet:

Remote Teaching (Einrichten per Fernsteuerung)	Der Manipulator wird per JOG-Tasten im Menü [JOG&Teach] in die einzurichtende Position gebracht.
Direct Teaching (Direktes Einrichten)	Stellt die Gelenke des Manipulators in den Zustand servo-frei. Bewegen Sie dann den Arm manuell in die einzurichtende Position.
MDI Teaching (MDI-Einrichten)	Geben Sie die Koordinaten der Zielposition über das Menü [JOG&Teach] oder über [Pose List] direkt ein.
Einrichten per JOG-Pad	Das JOG-Pad ist optional erhältlich. Weitere Informationen hierzu erhalten Sie in Kapitel 2.8 „Bedienung per JOG-Pad“.
Einrichten per Handprogrammiergerät	Handprogrammiergeräte sind optional erhältlich. Weitere Informationen hierzu erhalten Sie in Kapitel 2.9 „Bedienung per Handprogrammiergerät“.

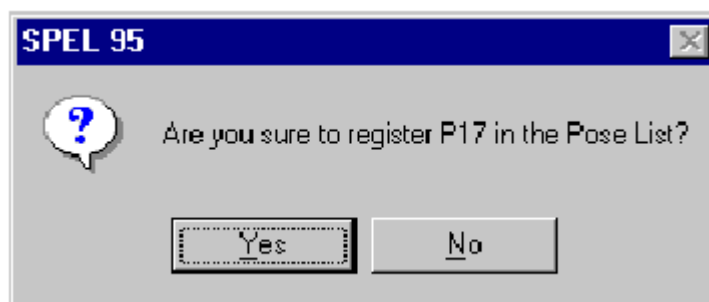
Weitere Informationen zum Einrichten eines Manipulators durch direkte Eingabe der Koordinaten erhalten Sie in Kapitel 2.4 „Bedienung aus dem Menü [Pose List]“.

### 2.3.1 Remote Teaching (Einrichten per Fernsteuerung)

Hierbei wird der Manipulator per JOG-Tasten im Menü [JOG&Teach] in die einzurichtende Position gebracht. Der Einrichtungsvorgang wird nachfolgend erklärt: Stellen Sie sicher, dass sich der Manipulator im MOTOR ON-Status befindet.



- (1) Benutzen Sie die JOG-Knöpfe, um den Manipulator in die einzurichtende Position zu verfahren.
- (2) Stellen Sie die Pose No. (Positionsdaten-Nummer) in der Kombobox [No.] im [JOG&Teach]-Menü ein.
- (3) Klicken Sie auf den Knopf <Teach P#>. Folgende Meldung erscheint:

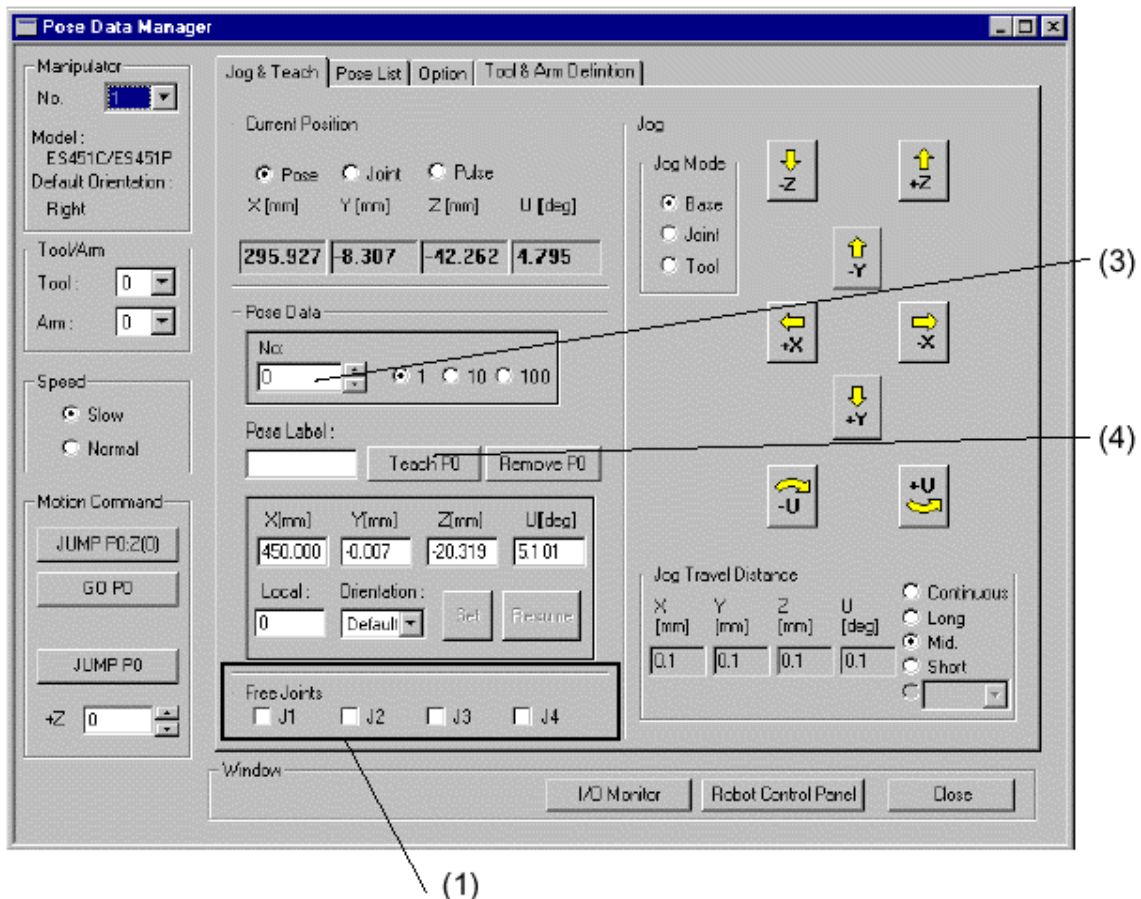


- (4) Klicken auf <Yes> (Ja), richtet sich die aktuelle Manipulatorposition unter der festgelegten Positionsnummer ein.

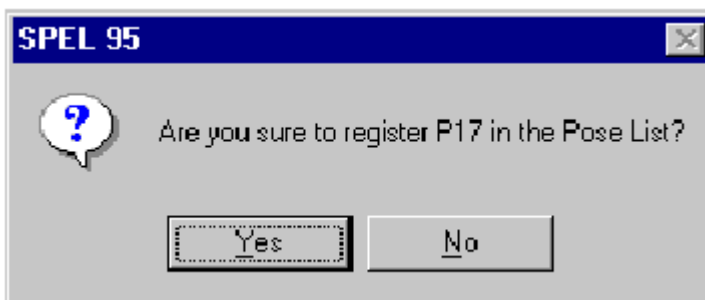
Wiederholen Sie die Schritte (1) – (4), um das Einrichten weiterzuführen.

### 2.3.2 Direct Teaching (Direktes Einrichten)

Für das direkte Einrichten des Roboters muss das Gelenk des Manipulators servo-frei geschaltet sein, damit der Arm manuell bewegt werden kann. Der Einrichtungsvorgang wird nachfolgend erklärt: Stellen Sie sicher, dass der Knopf [MOTOR] im [Robot Control]-(Robotersteuerung)-Menü auf ON steht.



- (1) Klicken Sie für jedes Gelenk das entsprechende Optionsfeld in der [Free Joints]-Gruppenauswahl im Menü [JOG&Teach] an, um dieses servo-frei zu schalten. Dadurch können Sie die Arme des Manipulators frei bewegen.
- (2) Bewegen Sie den Manipulator manuell in die Zielposition.
- (3) Stellen Sie die Pose No. (Positionsdaten-Nummer) in der Kombobox [No.] im [Jog&Teach]-Menü ein.
- (4) Klicken Sie auf den Knopf <Teach P#>. Folgende Meldung erscheint:

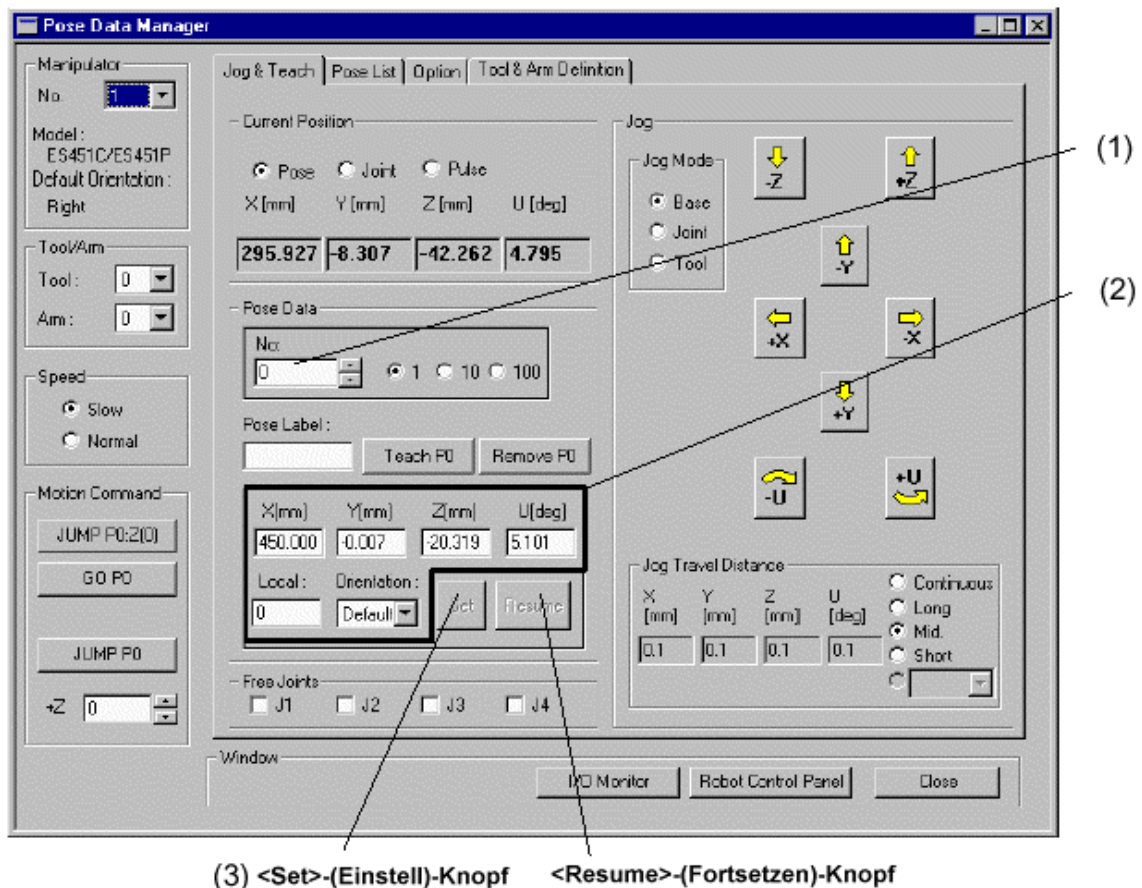


- (5) Klicken auf <Yes> (Ja), richtet sich die aktuelle Manipulatorposition unter der festgelegten Positionsnummer ein.

Wiederholen Sie die Schritte (2) – (5), um das Einrichten weiterzuführen.

### 2.3.3 MDI Teaching (MDI Einrichten)

Die Zielposition kann durch Eingabe der Koordinaten direkt am Bildschirm eingerichtet werden.



(3) <Set>-<Einstell>-Knopf <Resume>-<Fortsetzen>-Knopf

- (1) Stellen Sie die Pose No. (Positionsdaten-Nummer) in der Kombobox [No.] im [Jog&Teach]-Menü ein.
- (2) Geben Sie die Werte in die Felder für [X], [Y], [Z] und [U] ein. Sie können gleichzeitig den Wert für die [Ausrichtung] eingeben.
- (3) Klicken Sie auf den Knopf <Set>, wenn Sie alle Daten eingegeben haben.

Wiederholen Sie die Schritte (1) – (3), um das Einrichten weiterzuführen.

### 2.3.4 Editieren der Positionsdaten

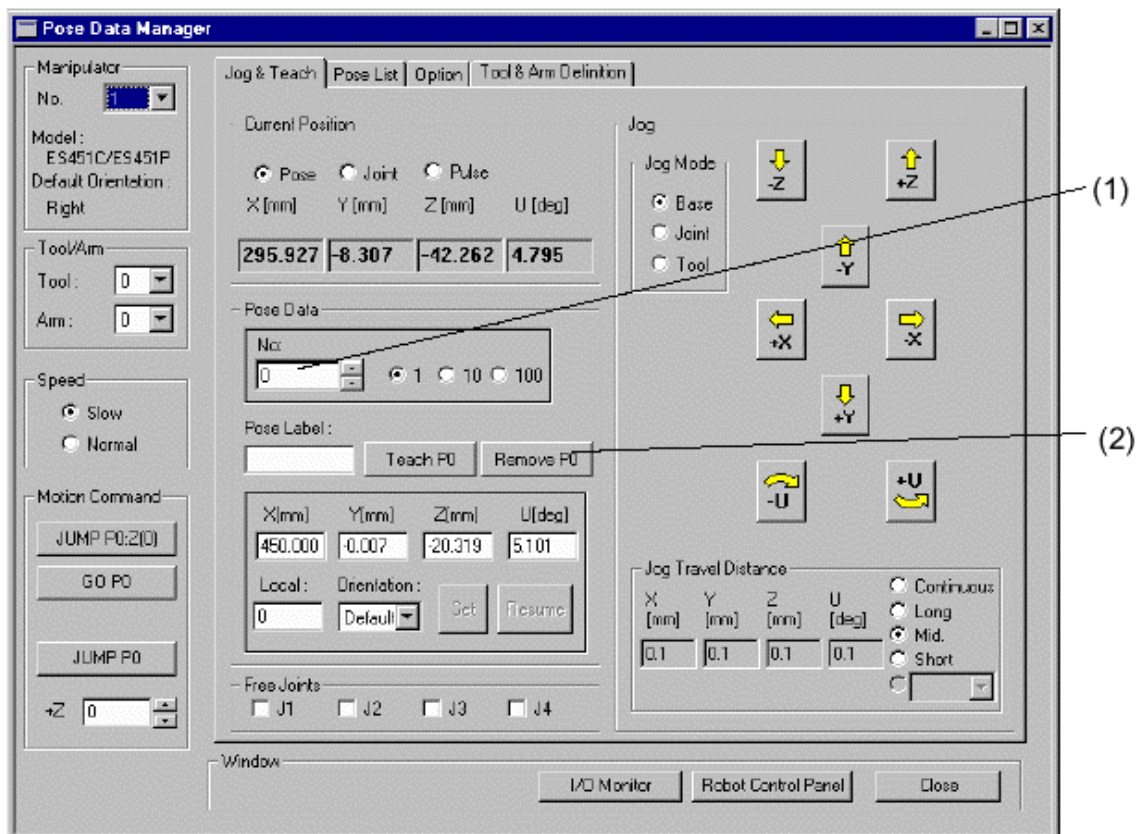
Die eingerichteten Positionsdaten lassen sich durch wiederholtes Ausführen der Einrichtungsprozedur ändern.

Durch Remote Teaching eingerichtete Positionsdaten können z.B. durch MDI-Einrichten geändert werden. Weitere Information erhalten Sie in Kapitel 5.2 „Editor – Editieren der Positionsdaten-Datei“.

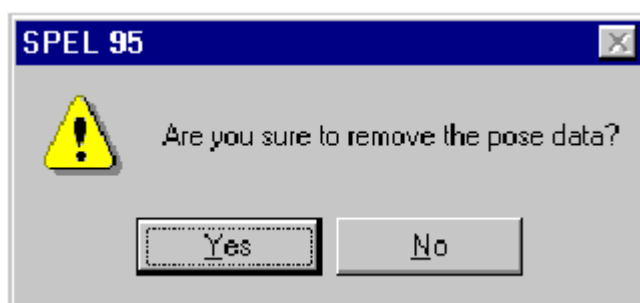
### 2.3.5 Löschen von Positionsdaten

Gehen Sie wie folgt vor, um einmal eingerichtete Positionsdaten zu löschen:

Entweder benutzen Sie das Menü [JOG&Teach], das Menü [Pose List], oder Sie editieren die Daten der Positionsdaten-Datei.



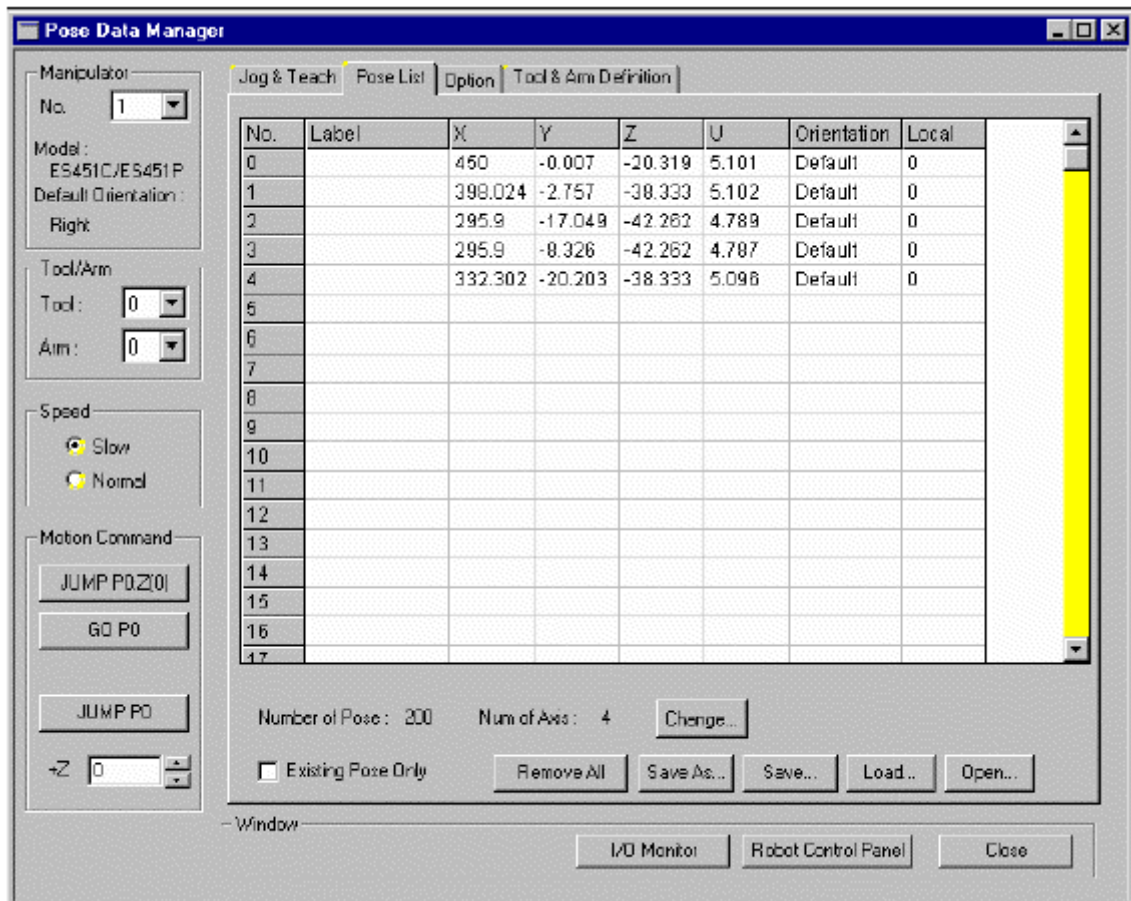
- (1) Stellen Sie die Pose No. (Positionsdaten-Nummer) in der Kombobox [No.] im [JOG&Teach]-Menü ein.
- (2) Klicken Sie auf den Knopf <Remove P#>(Entferne P#).
- (3) Klicken Sie auf den Knopf <Yes>, wenn folgende Meldung erscheint: Die eingerichteten Positionsdaten werden gelöscht.



## 2.4 Bedienung per Pose List-Menü

Das [JOG&Teach]-Menü dient hauptsächlich dem Einrichten von Positionsdaten. Es gibt einige Anlässe, in denen Sie möglicherweise die Positionsdaten oder die Positionsdaten-Nummer, Bezug nehmend auf die Liste der Positionsdaten, ändern möchten. In diesen Fällen benutzen Sie das Pose List-Menü.

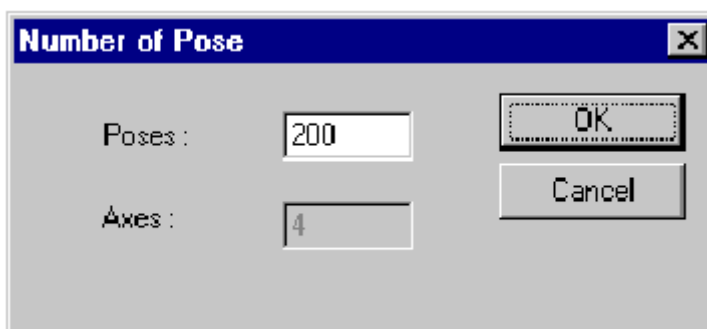
Wenn Sie im Pose List-Menü die Tool-Befehle ausführen oder auf den <Pose List>-Knopf in der Werkzeugleiste klicken, ändert sich der Bildschirm:



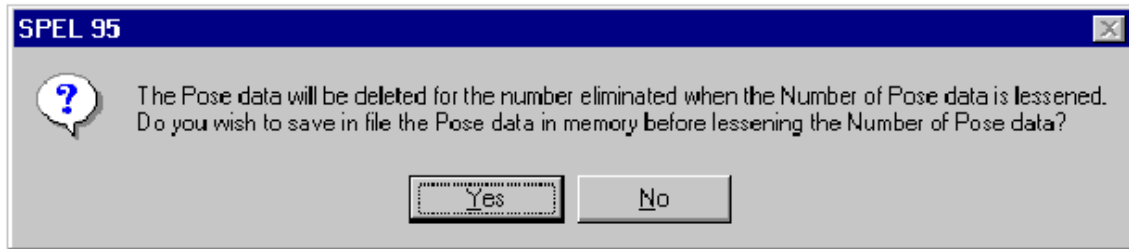
Der Bildschirm zeigt die Positionsdaten, die Nummer der Positionsdaten und die Anzahl der eingerichteten Achsen. Die Standardeinstellung für die Anzahl der Positionsdaten ist 200.

### 2.4.1 Änderungen an den maximalen Positionsdateneinstellungen

Klicken Sie auf den Knopf <Change...> (Ändern) im Menü [Pose List], um die maximale Einstellung für die Anzahl der Positionsdaten zu ändern. Die nachfolgende Dialogbox erscheint. Geben Sie eine maximale Anzahl für die Positionsdaten in das Feld ein, und klicken Sie dann auf <OK>.



Ist die neue Eingabe niedriger als die alte Einstellung, erscheint die nachfolgende Meldung. Speichern Sie die Daten, falls notwendig.



## 2.4.2 Editieren der Koordinaten

Die Daten für die Koordinaten können im Menü [Pose List] am Bildschirm geändert werden. Weitere Information erhalten Sie in Kapitel 5.2 „Editor – Editieren der Positionsdaten-Datei“.

### ▪ Direktes Eingeben der Koordinaten (MDI-Einrichten)

Wenn Sie den Mauszeiger auf die zu ändernden Positionsdaten bewegen und dann klicken, erscheint ein Cursor. Geben Sie eine Zahl ein, sie wird an der Position des Cursors eingetragen.

### ▪ Ändern der Koordinaten

Wenn Sie den Mauszeiger auf die zu ändernden Positionsdaten bewegen und dann klicken, erscheint ein Cursor. Geben Sie eine Zahl ein, sie wird an der Position des Cursors eingetragen. Um die Daten durch neue zu ersetzen, müssen Sie auf die zu ändernden Daten doppelklicken, um diese hervorzuheben. Geben Sie dann einfach Ihre neuen Daten ein.

### ▪ Ändern der Ausrichtung

Klicken Sie auf die [Orientation] der Daten, die Sie ändern möchten. Eine Kombobox erscheint. Klicken Sie auf den Pfeil, und eine Auswahlliste erscheint. Wählen Sie den gewünschten Armmodus aus der Liste.

### ▪ Ändern der Positionsmarken

Wenn Sie den Mauszeiger auf die zu ändernden Positionsdaten bewegen und dann klicken, erscheint ein Cursor. Um die Daten durch neue zu ersetzen, müssen Sie auf die zu ändernden Daten doppelklicken. Nachdem die Marke hervorgehoben ist, geben Sie einen neuen Namen ein. Er wird automatisch upgedatet.

### ▪ Verschieben oder Kopieren von Positionsdaten

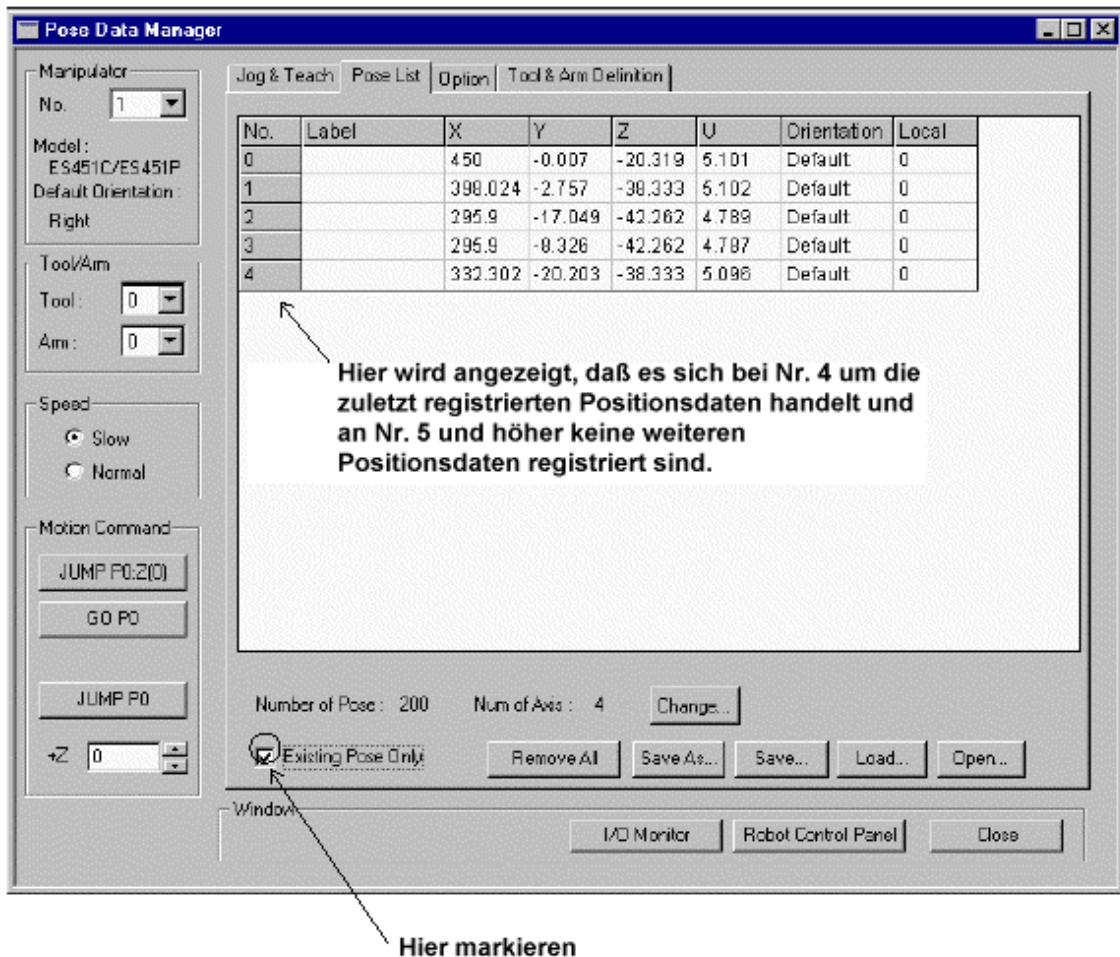
1. Klicken Sie auf die Nummer der Positionsdaten, die Sie verschieben oder kopieren möchten. Die gewählten Positionsdaten werden hervorgehoben.
2. Folgen Sie einem der nächsten Schritte:
  - Zum Verschieben führen Sie den Befehl [Edit]-[Cut] (Bearbeiten-Ausschneiden) aus.
  - Zum Kopieren führen Sie den Befehl [Edit]-[Copy] (Bearbeiten-Kopieren) aus.
3. Um die Positionsdaten zu verschieben oder zu kopieren, platzieren Sie den Cursor an einer Stelle in der Zeile der gewünschten Positionsnummer, oder klicken Sie auf die Positionsdaten-Nummer, um diese hervorzuheben.
4. Führen Sie dann den Befehl [Edit]-[Paste] (Bearbeiten-Einfügen) aus.

**HINWEIS**

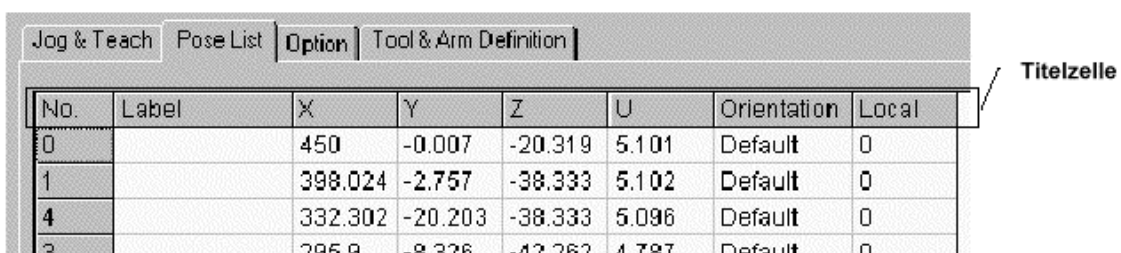
Um aufeinander folgende Positionsdaten zu markieren, klicken Sie auf die erste und letzte gewünschte Positionsdaten-Nummer. Halten Sie dabei die Umschalttaste gedrückt.

**2.4.3 Anzeigen von ausschließlich existierenden Daten**

Wenn der Optionsschalter [Existing Pose Only] (Nur existierende Positionsdaten) markiert ist, werden nur die Zeilen mit den aktuellen Daten angezeigt.

**2.4.4 Ändern der Reihenfolge der Daten**

Die Reihenfolge der Positionsdaten kann der Anwender selbst festlegen. Einmaliges Klicken auf die Titelzelle bewirkt, dass die Daten in aufsteigender Reihenfolge gemäß der Werte in den Zeilen sortiert werden. Erneutes Klicken auf die Titelzelle bewirkt, dass die Daten in absteigender Reihenfolge sortiert werden.





### 2.4.5 Löschen von Positionsdaten

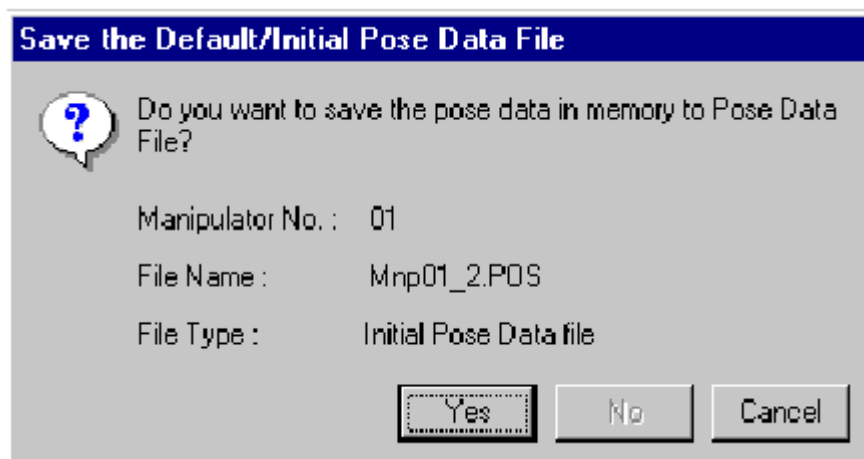
Um spezielle Positionsdaten zu löschen, klicken Sie die dazugehörige Zeilennummer an. Dann wählen Sie entweder [Cut] (Ausschneiden) aus dem Menü, oder Sie drücken die Taste [X], während Sie die Taste [STRG] gedrückt halten.

Um alle Positionsdaten auf einmal zu löschen, klicken Sie auf den Schalter [Remove all] (Alles löschen) im Menü [Pose List].

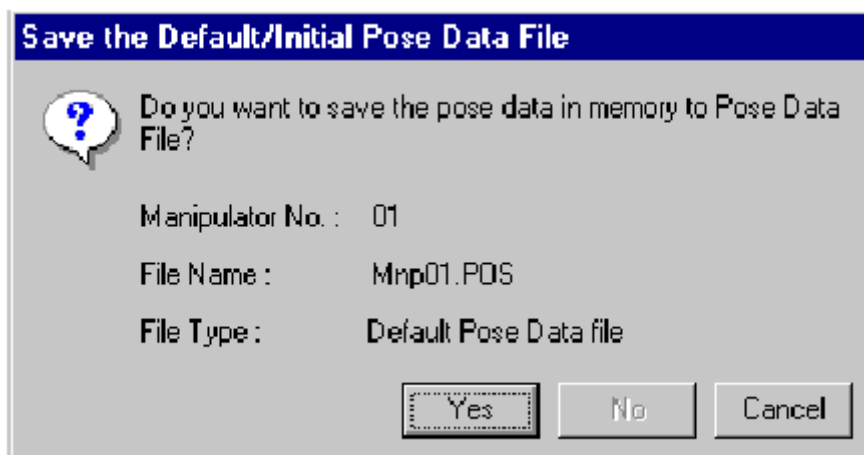
### 2.4.6 Speichern als Anfangs- oder Standardpositionsdaten-Datei

Um die Positionsdaten als Anfangs- oder Standardpositionsdaten-Datei zu speichern (hierbei wird die aktuelle Datei mit den neuen Daten überschrieben), klicken Sie auf <Save...> (Speichern...) im Menü [Pose List].

Die folgende Meldung erscheint, wenn Sie eine Anfangspositionsdaten-Datei speichern. Klicken Sie auf <Yes>.

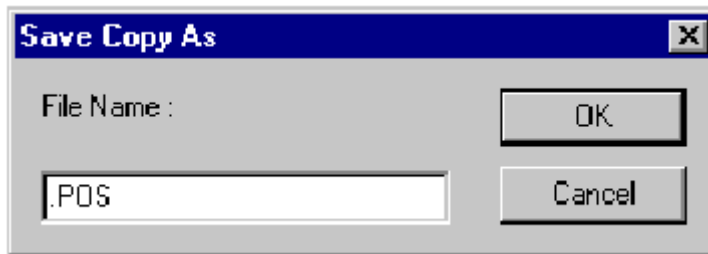


Die folgende Meldung erscheint, wenn Sie eine Standardpositionsdaten-Datei speichern. Klicken Sie auf <Yes>.



### 2.4.7 Speichern als Positionsdaten-Datei

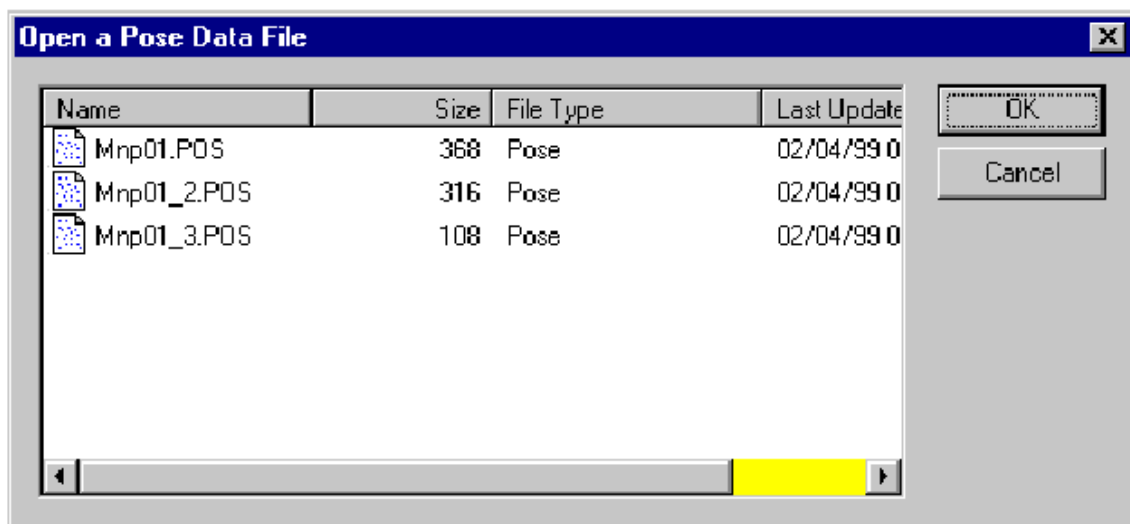
Um die Positionsdaten als Positionsdaten-Datei zu speichern, klicken Sie auf <Save as...> (Speichern unter...) im Menü [Pose List]. Vergeben Sie in der erscheinenden Dialogbox einen Namen für die Datei, und klicken Sie auf <Yes>.



Die Positionsdaten im Speicher sind jetzt als besondere Positionsdaten-Datei gespeichert.

### 2.4.8 Laden einer Positionsdaten-Datei

Klicken Sie auf <Load...> (Laden...) im Menü [Pose List]. Wählen Sie die zu ladende Datei aus, und klicken Sie dann auf <OK>.



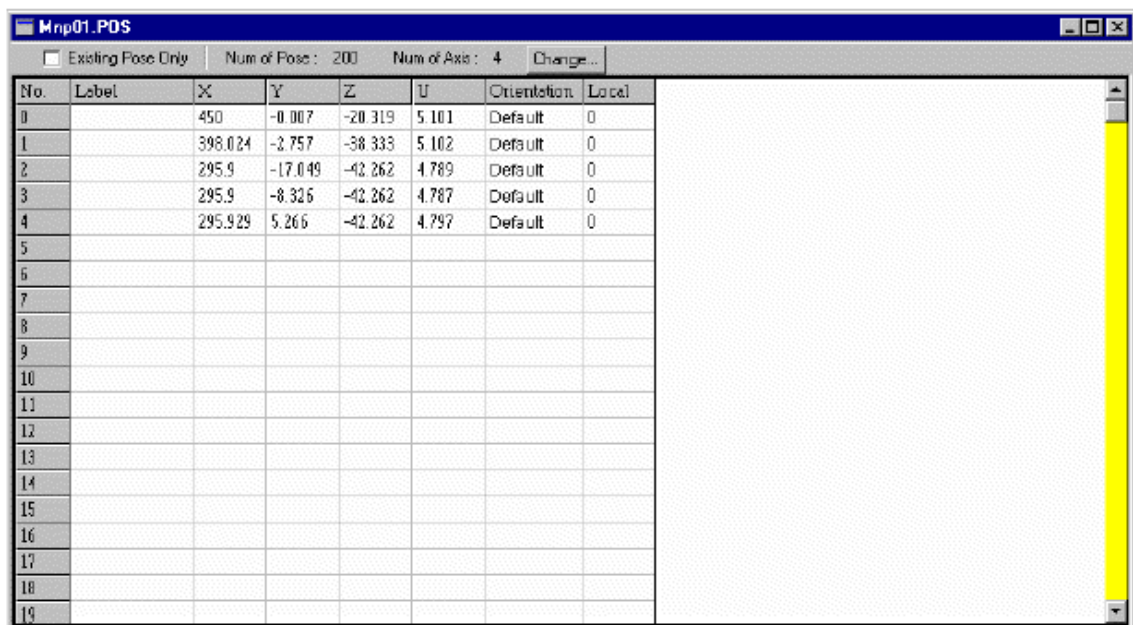
Die Daten aus der gewählten Positionsdaten-Datei werden in den Systempeicher geladen, und die Datei wird geöffnet.

## 2.5 Positionsdaten-Datei

Die Positionsdaten können in den Speicher der Steuerung geladen/geschrieben werden. Dann können Sie im Menü [Pose Data Manager] bearbeitet werden. Solche im Speicher befindlichen Daten können als Datei gespeichert werden. Die Werte der Positionsdaten sind als Positionsdaten-Datei direkt editierbar. Eine Positionsdaten-Datei lässt sich auf verschiedenen Wegen öffnen:

- Führen Sie den Befehl [File]-[Open] (Datei-Öffnen) aus, und wählen Sie die zu ladende Datei.
- Wechseln Sie in das Menü [Project Manager]. Wählen Sie aus der Auswahlliste [Files in the project] (Projektdateien) die entsprechende Datei.
- Wechseln Sie in das Menü [Pose List] im [Pose Data Manager]-Menü. Klicken Sie auf <Open> (Öffnen), und wählen Sie die Positionsdaten-Datei.

Die festgelegten Daten werden angezeigt.

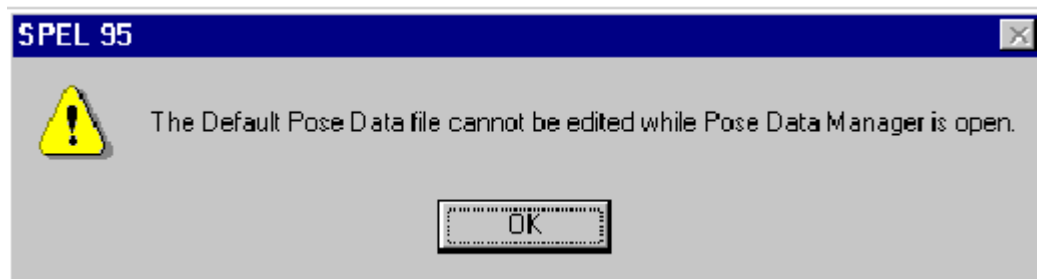


The screenshot shows a window titled "Mnp01.POS" with a toolbar containing "Existing Pose Only", "Num of Pose : 200", "Num of Axis : 4", and "Change...". Below the toolbar is a table with 19 rows and 8 columns. The columns are labeled "No.", "Label", "X", "Y", "Z", "U", "Orientation", and "Local". The first four rows contain numerical data, while the remaining rows are empty.

No.	Label	X	Y	Z	U	Orientation	Local
0		450	-0.007	-20.319	5.101	Default	0
1		398.024	-2.757	-38.333	5.102	Default	0
2		295.9	-17.049	-42.262	4.789	Default	0
3		295.9	-8.326	-42.262	4.787	Default	0
4		295.929	5.266	-42.262	4.797	Default	0
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

Ist die Datei bereits als Anfangs- oder Standardpositionsdaten-Datei im Speicher, kann sie nicht geöffnet werden.

Die Datei muss im [Pose Data Manager] bearbeitet werden.



- **Editieren einer Positionsdaten-Datei**

Jede Positionsdaten-Datei im geöffneten Projekt kann im [Pose Data Manager] bearbeitet werden. Nähere Informationen hierzu erhalten Sie in Kapitel 5 "Editor".

- **Anwenden einer Gesamtbearbeitung von vielfachen Positionsdaten-Dateien**

Es könnte möglich sein, dass eine bestimmte Koordinate geändert werden muss, unabhängig von einem festgelegten Winkel in vielfachen Dateien. In diesen Fällen benutzen Sie die Funktion [Paste with Offset] (Einfügen mit Offset), mit deren Hilfe ein spezieller Offset-Wert an vielfache Dateien vergeben werden kann. Nähere Informationen hierzu erhalten Sie in Kapitel 5 "Editor".

- **Registrieren einer Positionsdaten-Datei**

Eine Positionsdaten-Datei kann tatsächlich solange nicht benutzt werden, bis Sie als Anfangs- oder Standard-Positionsdaten-Datei registriert worden ist.

Nähere Informationen hierzu erhalten Sie in den Kapiteln 2.6 "Anfangs-Positionsdaten-Datei" und 2.7 "Standard-Positionsdaten-Datei".

## 2.6 Anfangs-Positionsdaten-Datei

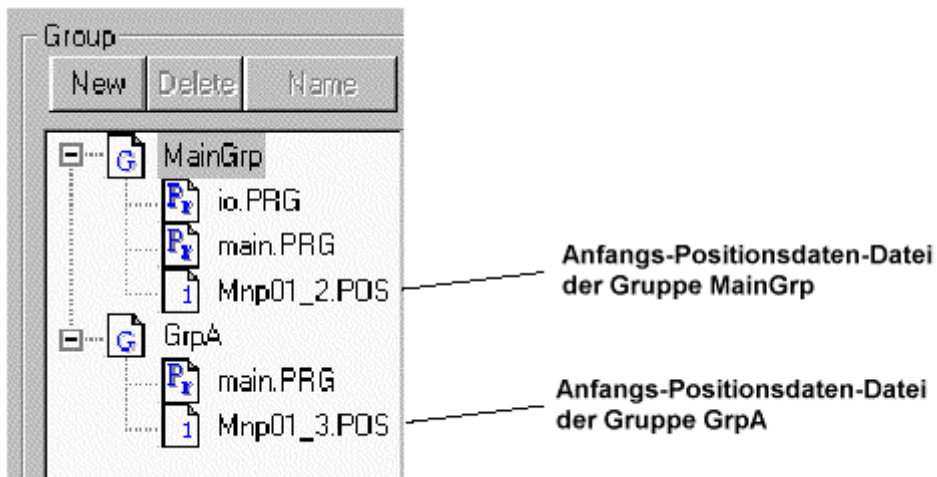
Damit ein Projekt korrekt funktioniert, muss jeder Manipulator zumindest über eine registrierte Positionsdaten-Datei verfügen.

Die Anfangs-Positionsdaten-Datei ist definiert als Positionsdaten-Datei, die der Anwender selbst in der "Gruppe" registriert.

Wenn eine Anfangs-Positionsdaten-Datei in der Projekt-Startgruppe registriert ist (die geladen wird, wenn SPEL 95 startet), werden automatisch die Positionsdaten aus dieser Datei geladen. Ist keine Anfangs-Positionsdaten-Datei in der Projekt-Startgruppe registriert, lädt das System automatisch die Daten aus der Standard-Positionsdaten-Datei.

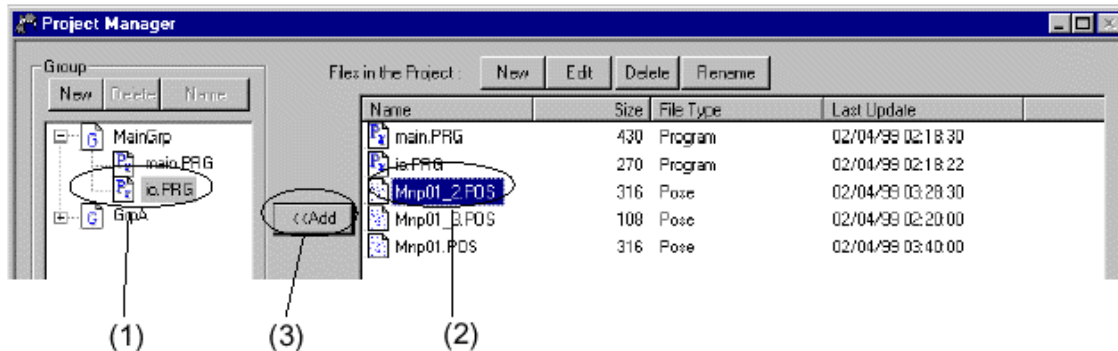
Eine neue Anwendung kann, wie nachfolgend dargestellt, durch Nutzung einer existierenden Anwendung erzeugt werden:

Erzeugen Sie eine neue "Gruppe", und registrieren Sie eine neue Positionsdaten-Datei als Anfangs-Positionsdaten-Datei. Die neue Anwendung beinhaltet dasselbe Programm mit unterschiedlichen Positionsdaten.

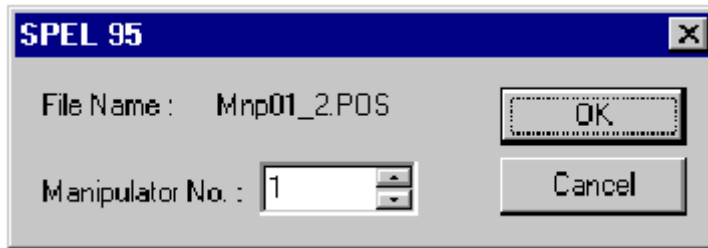


### 2.6.1 Registrieren einer Anfangs-Positionsdaten-Datei

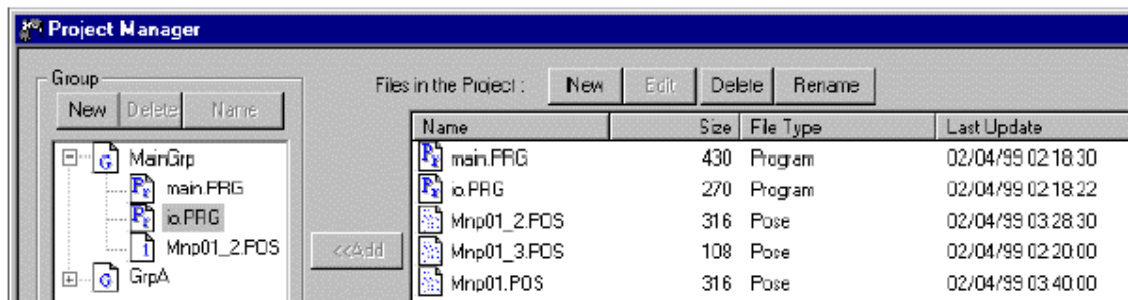
- (1) Öffnen Sie das Menü [Projekt Manager], und wechseln Sie zur [Group]-Auswahlliste. Wählen Sie die Gruppe aus, deren Anfangs-Positionsdaten-Datei registriert werden soll.
- (2) Wählen Sie die zu registrierende Positionsdaten-Datei in der Auswahlliste [Files in the project] in der rechten Seite des Fensters.
- (3) Klicken Sie auf den <Add>- (Hinzufügen)-Schalter.



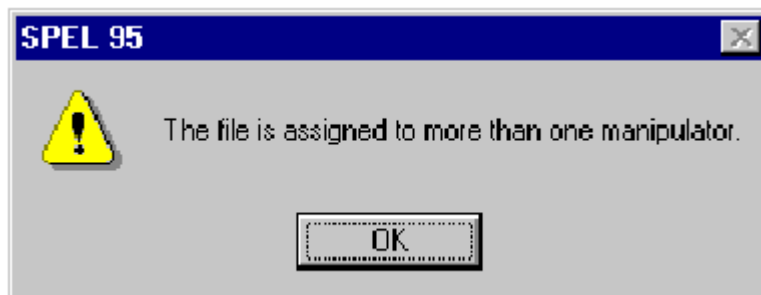
Wählen Sie den zu registrierenden Manipulator aus. Wählen Sie die Nummer des Manipulators aus der Auswahlliste [Manipulator No.] (Manipulator Nr.), und klicken Sie dann auf <OK>.



Die festgelegten Positionsdaten werden in der „Gruppe“ registriert und als Anfangs-Positionsdaten-Datei gespeichert.

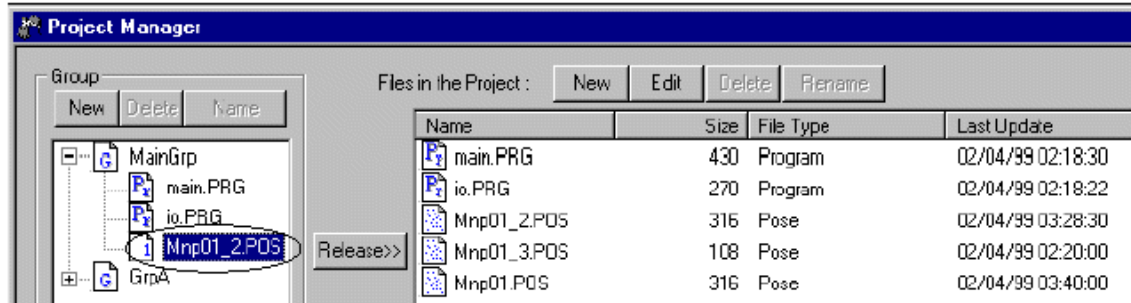


Eine Positionsdaten-Datei, die bereits in einer anderen Gruppe registriert worden ist, oder eine Standard-Positionsdaten-Datei wird nicht als Anfangs-Positionsdaten-Datei registriert. Folgende Meldung erscheint:



## 2.6.2 Freigeben der Anfangs-Positionsdaten-Datei

Wählen Sie die zu löschende Datei in der Auswahlliste [Group]. Klicken Sie dann auf den Schalter <Release>(Freigeben).

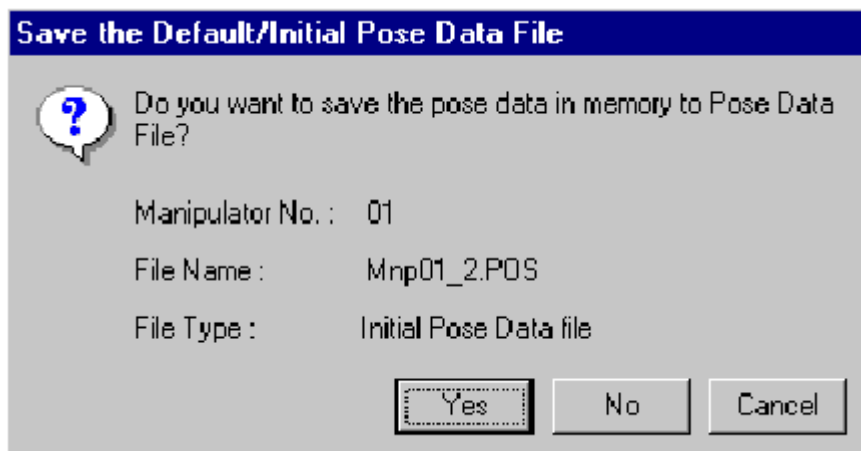


Ist die Datei einmal aus der Gruppe „freigegeben“, kann sie nicht mehr vom Manipulator verwendet werden. Die Datei ist jedoch nicht gelöscht! Sie bleibt in der Auswahlliste [Files in the project].

## 2.6.3 Updaten der Anfangs-Positionsdaten-Datei

Das Updaten der Positionsdaten in dem Speicher der Steuerung, die im [Pose Data Manager] angezeigt werden, bewirkt nicht automatisch ein Update der Anfangs-Positionsdaten-Datei. Um die Anfangs-Positionsdaten-Datei upzudaten, klicken Sie auf den Knopf <Save...> im Menü [Pose List].

Wenn Sie auf den Knopf <Close> (Schließen) klicken, erscheint folgende Meldung:



Klicken Sie auf <Yes>, um die Daten in der Anfangs-Positionsdaten-Datei zu speichern. Das Fenster [Pose Data Manager] wird geschlossen. Die Daten werden in die Anfangs-Positionsdaten-Datei gespeichert.

Klicken Sie auf <No>, wenn Sie die Daten nicht in der Anfangs-Positionsdaten-Datei speichern wollen. Das Fenster [Pose Data Manager] wird geschlossen. Die Daten werden im Speicher der Steuerung gespeichert, aber die Anfangs-Positionsdaten-Datei wird nicht upgedatet.

## 2.7 Standard-Positionsdaten-Datei

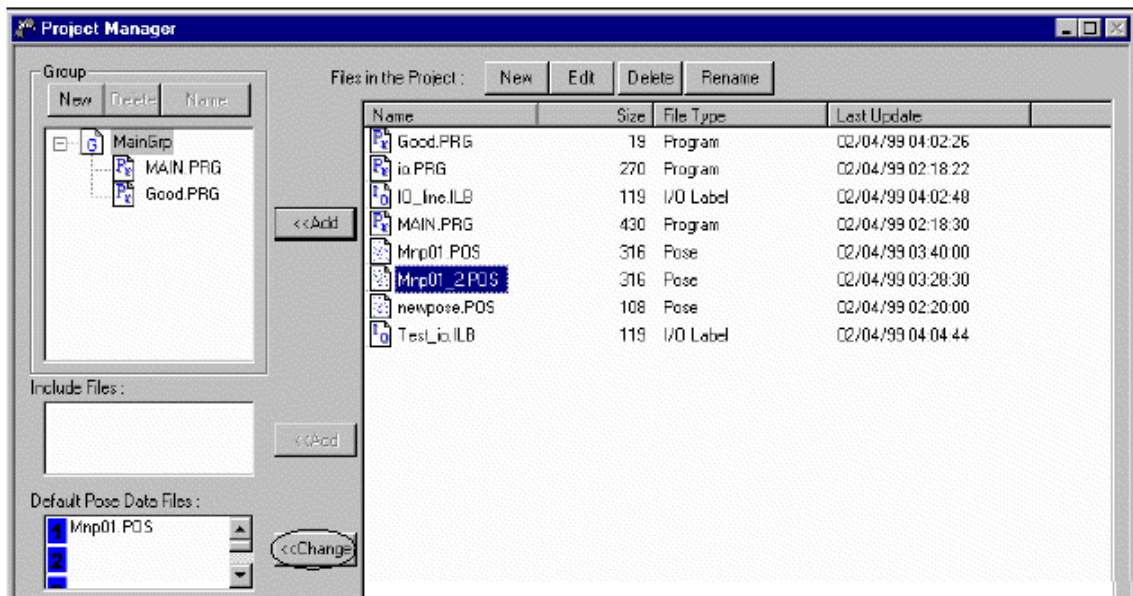
Ein Projekt benötigt eine Standard-Positionsdaten-Datei für jeden Manipulator. Wenn ein neues Projekt erzeugt wird, wird automatisch eine Standard-Positionsdaten-Datei registriert. Es wird auch dann eine neue Standard-Positionsdaten-Datei registriert, wenn ein Manipulator neu registriert wird.

Eine registrierte Standard-Positionsdaten-Datei bleibt auch dann unverändert, wenn eine andere Datei als Anfangs-Positionsdaten-Datei gespeichert wird.

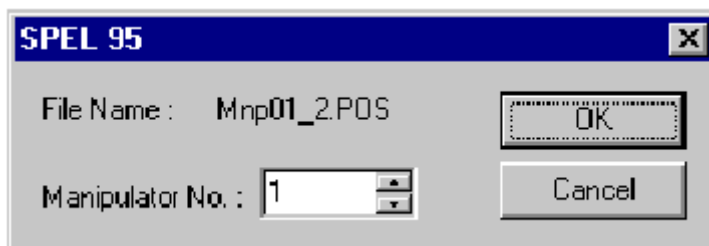
Ist keine Datei als Anfangs-Positionsdaten-Datei in der Projekt-Startgruppe registriert, wird die Standard-Positionsdaten-Datei in den Speicher geladen.

### 2.7.1 Ändern der Standard-Positionsdaten-Datei

Wählen Sie die zu registrierende Positionsdaten-Datei in der Auswahlliste [Files in the project] in der rechten Seite des Fensters, und klicken Sie auf den Knopf <Change> (Ändern).

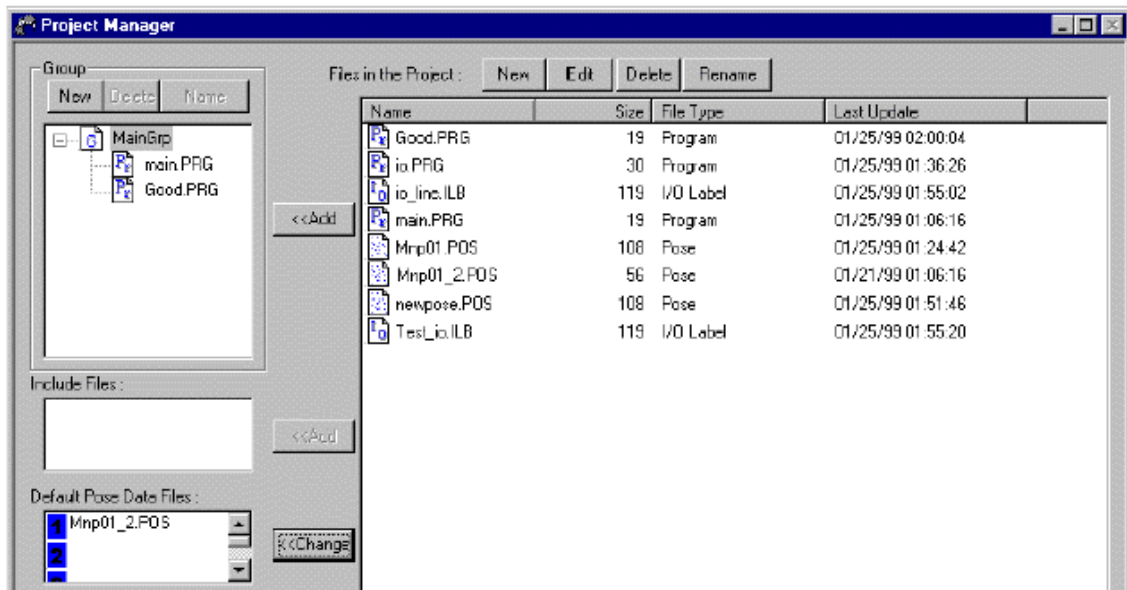


Wählen Sie den Manipulator, für den die Änderung durchgeführt werden soll, anhand seiner Nummer. Klicken Sie auf <OK>.





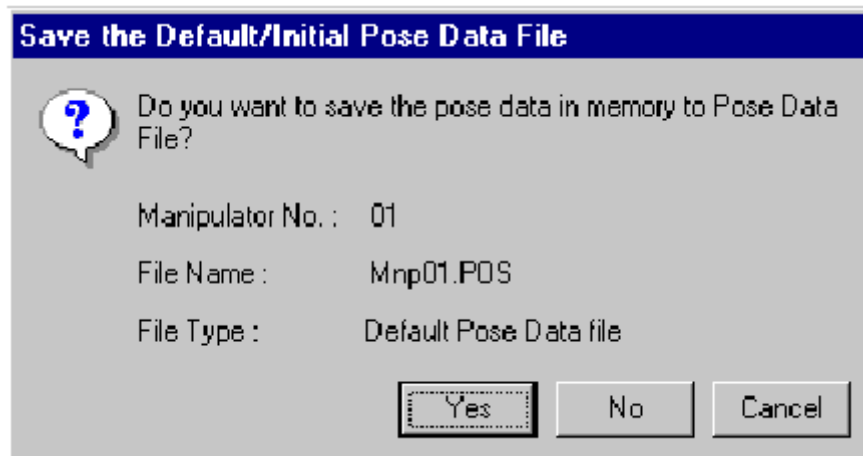
Die Standard-Positionsdaten-Datei wurde geändert und als neue Standard-Positionsdaten-Datei festgelegt.



## 2.7.2 Updaten der Standard-Positionsdaten-Datei

Auch wenn die Positionsdaten im Speicher durch den [Pose Data Manager] upgedated werden, wird die Standard-Positionsdaten-Datei nicht automatisch upgedated. Um die Standard-Positionsdaten-Datei upzudaten, klicken Sie auf den Knopf <Save...> im Menü [Pose List].

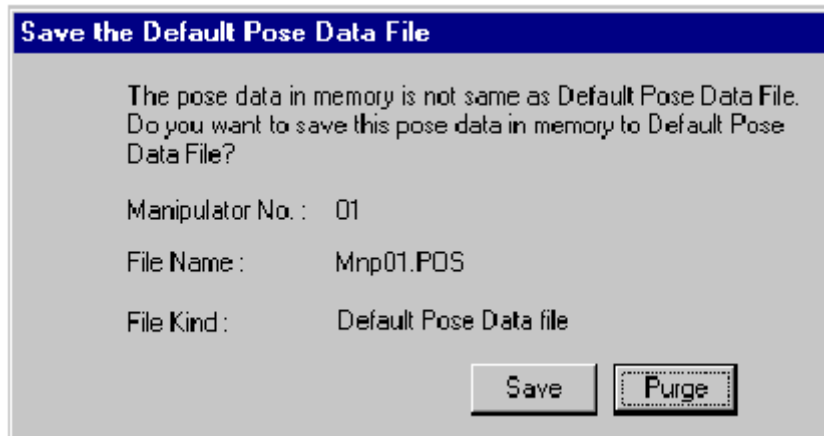
Wenn Sie auf den Knopf <Close> (Schließen) klicken, erscheint folgende Meldung:



Klicken Sie auf <Yes>, um die Daten in der Standard-Positionsdaten-Datei zu speichern. Das Fenster [Pose Data Manager] wird geschlossen. Die Daten werden in die Standard-Positionsdaten-Datei gespeichert.

Klicken Sie auf <No>, wenn Sie die Daten nicht in der Standard-Positionsdaten-Datei speichern wollen. Das Fenster [Pose Data Manager] wird geschlossen. Die Daten werden im Speicher der Steuerung gespeichert, aber die Standard-Positionsdaten-Datei wird nicht upgedated.

Die nachfolgende Meldung erscheint, wenn die Positionsdaten in dem Speicher der Steuerung nicht zu den Standard-Positionsdaten passen, wenn das Projekt im Programmiermodus geschlossen wird.



Klicken Sie auf <Save>, um die Daten im Speicher in der Standard-Positionsdaten-Datei zu speichern.

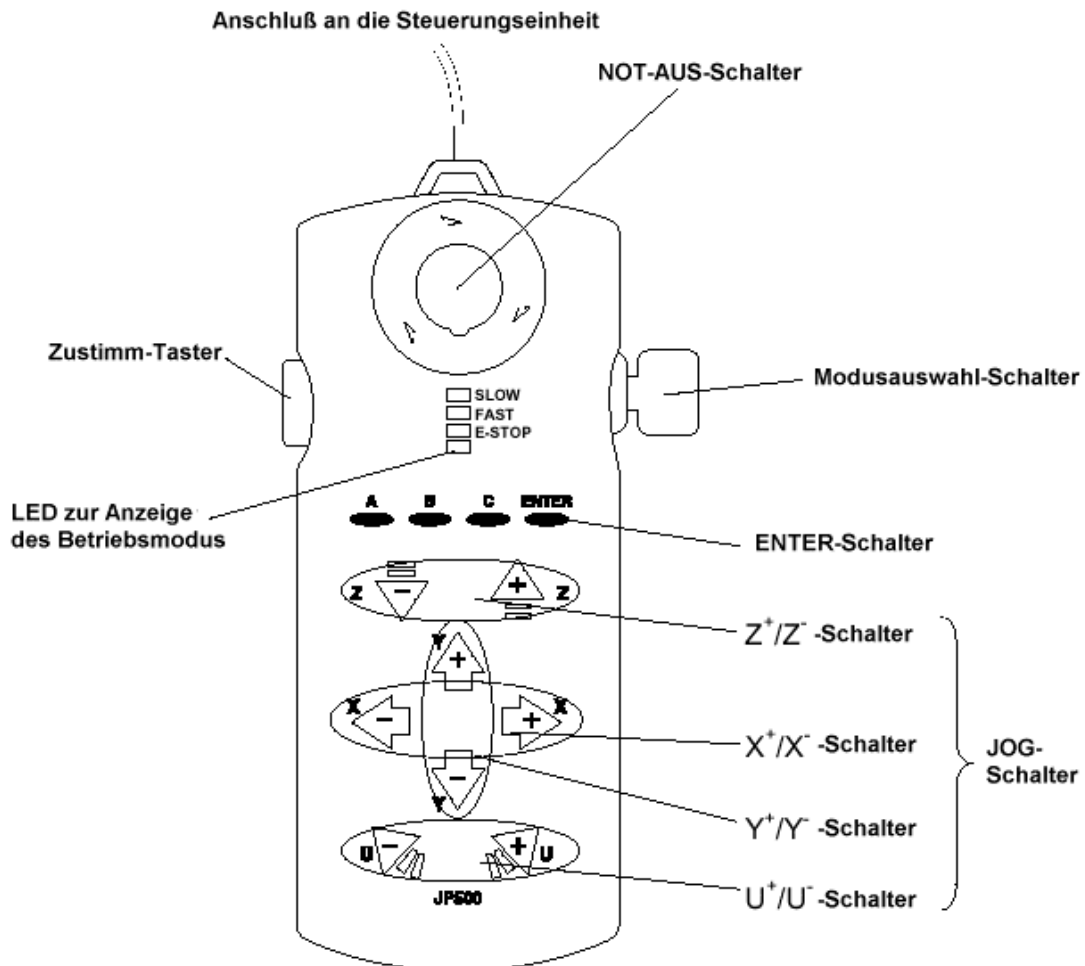
Klicken Sie auf <Purge> (Verwerfen), um die Daten im Speicher zu verwerfen.

Die Positionsdaten werden im AUTO-Modus automatisch verworfen.

## 2.8 Bedienung per JOG-Pad

Das JOG-Pad wird für ein effektives Einrichten eingesetzt. Es ist optional erhältlich. Das JOG-Pad selbst verfügt über keine Anzeige. Daher ist es daher erforderlich, eine andereanzeigeeinheit (z.B. die Anzeige der Steuerungseinheit) zu verwenden.

### 2.8.1 Bedienung



- NOT-AUS-Schalter:** Drücken Sie diesen Schalter, um das Gerät anzuhalten, wenn es einen Notfall gibt.
- Modusauswahl-Schalter:** Mit diesem Schalter können Sie die Bedienmodi wechseln. Drehen Sie den Schalter auf ATTEND, um in den Wartungs-Modus zu gelangen, von dem aus die Bedienung per JOG-Pad möglich ist. Drehen des Schalters auf NORMAL wechselt in den NORMAL-Modus.
- Zustimm-Taster:** Drücken Sie diesen Taster, damit der Manipulator sich mit den JOG-Schaltern des JOG-Pads bewegen lässt.
- Enter-Schalter:** Dieser Schalter wird benutzt, um die Befehle auf der Anzeige mit dem JOG-Pad auszuführen.
- JOG-Schalter:** Während der Zustimmtaster gedrückt wird, lässt sich der Manipulator mit Hilfe der JOG-Schalter bewegen. Wird der Zustimmtaster nicht gedrückt, ändert sich mit den JOG-Schaltern der Fokus auf dem Monitor.

LED zur Anzeige des Betriebsmodus: Hier wird der Modus des Spannungszustandes angezeigt:

Langsame Betriebsgeschwindigkeit:	Grün
Normale Betriebsgeschwindigkeit:	Orange
NOT-AUS-Status:	Rot

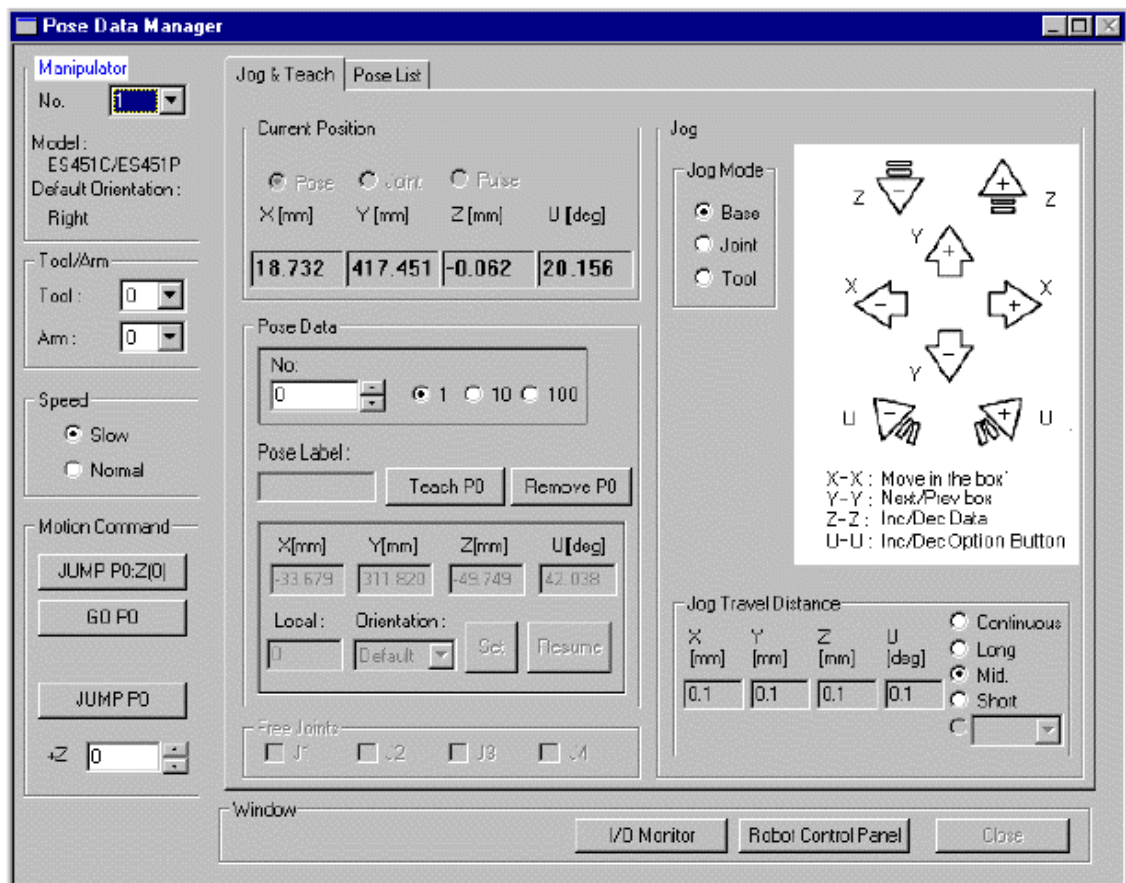
**HINWEIS** Wenn das JOG-Pad eingesetzt wird, kann nicht in den uneingeschränkten Modus (HP\_Status) gewechselt werden.

#### ▪ Wechseln der Bedienmodi

Drehen Sie den Schlüsselschalter am JOG-Pad, um in den Wartungs-Modus zu wechseln. Zu diesem Zeitpunkt werden alle Ausführungen angehalten. Der arbeitende Manipulator hält unverzüglich an und pausiert vorübergehend aufgrund des anliegenden PAUSE-Befehls.

Das unten stehende [Jog&Teach]-Menü wird angezeigt.

**HINWEIS** Im Wartungs-Modus wird das [Jog&Teach]-Menü nicht angezeigt, wenn das Projekt geschlossen wird!



## 2.8.2 NOT-AUS-Schalter

Wird der NOT-AUS-Schalter am JOG-Pad gedrückt, wird die Spannungsversorgung des Manipulators abgestellt, die dynamische Bremse hält den Manipulator an, und das System wechselt in den NOT-AUS-Zustand. Der Manipulator benötigt, abhängig von seinem Ladestatus und der Betriebsgeschwindigkeit, etwas Zeit und Strecke, bevor er hält.

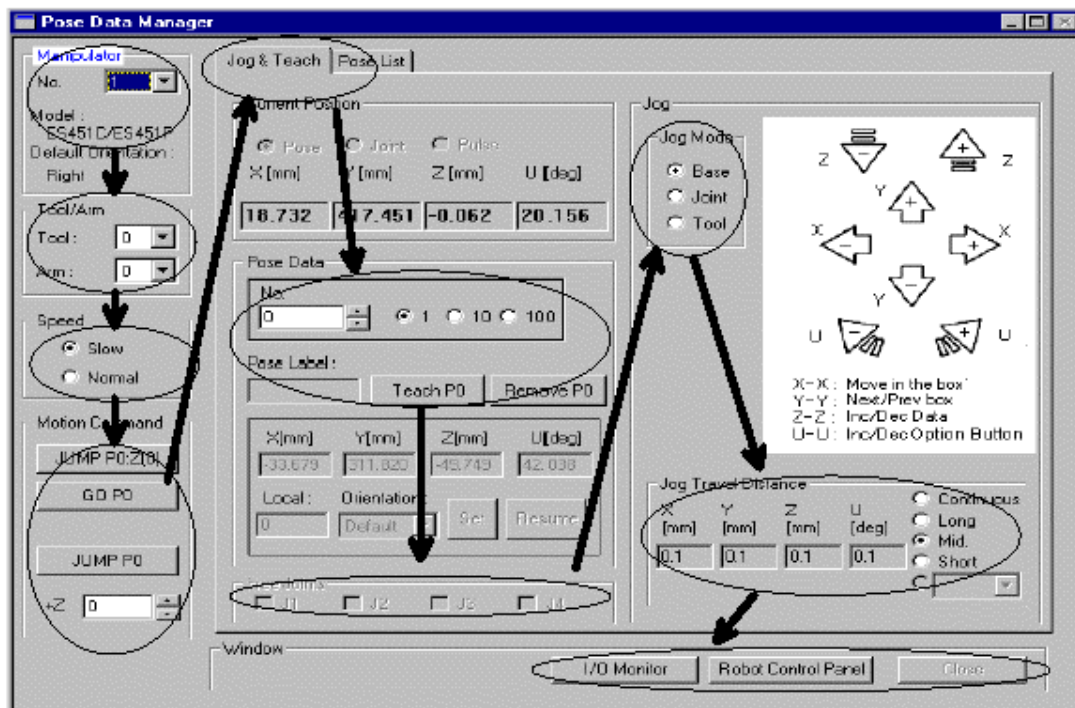
Die LED zur Anzeige des Betriebsmodus schaltet auf rot.

### ▪ Zurücksetzen des NOT-AUS-Zustandes

Nach einem NOT-AUS wird der NOT-AUS-Status folgendermaßen zurückgesetzt:

- (1) Drehen Sie den NOT-AUS-Schalter im Uhrzeigersinn, um den NOT-AUS-Status zurückzusetzen. Beachten Sie bitte, dass sich der Zustand nur durch Drehen des NOT-AUS-Schalters noch nicht zurückgesetzt hat!
- (2) Klicken Sie auf den Schalter <RESET/READY> (Reset/Fertig) im [Robot Control]-Menü. Die LED zur Anzeige des Betriebsmodus wechselt von rot auf grün oder orange, und der NOT-AUS-Status ist zurückgesetzt.

## 2.8.3 Bedienung per [Jog&amp;Teach]-Menü



→ Reihenfolge in der der Fokus der Gruppenbox mit Hilfe des Y<sup>+</sup>/Y<sup>-</sup>-Schalters geändert wird

Der Zustimmungstaster wird nicht benötigt, um den Bildschirm zu bedienen.

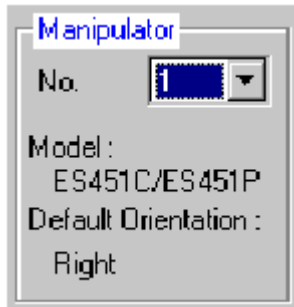
Bei jedem Drücken des Y<sup>+</sup>/Y<sup>-</sup>-Schalters am JOG-Pad ändert sich der Fokus im Menü wie folgt:

- [Manipulator]-Gruppe
- ↓
- [Tool/Arm]-Gruppe
- ↓
- [Speed]-Gruppe
- ↓
- [Motion Command]-Gruppe
- ↓
- Schalten zwischen den Menüs [JOG&Teach] und [Pose List]
- ↓
- [Pose Data]-Gruppe
- ↓
- [Free Joints]-Gruppe
- ↓
- [Jog Mode]-Gruppe
- ↓
- [Jog Travel Distance]-Gruppe
- ↓
- [Window]-Gruppe
- ↓
- [Manipulator]-Gruppe

- **[Manipulator]-Gruppe**

Diese Gruppe wird benötigt, um aus allen angeschlossenen Manipulatoren den auszuwählen, mit dem gearbeitet werden soll.

Benutzen Sie den  $Z^+/Z^-$  Schalter, um die Nummer des gewünschten Manipulators aus der Auswahlliste der angeschlossenen Manipulatoren auszuwählen. Drücken Sie dann den ENTER-Schalter. Die Schalter  $X^+/X^-$  und  $U^+/U^-$  haben hier keinerlei Funktion.

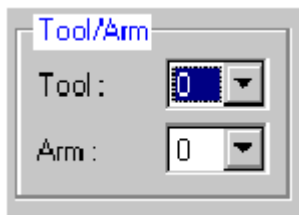


- **[Tool/Arm]-Gruppe (Werkzeug/Arm)**

Diese Gruppe wird benötigt, um das Werkzeug und den Arm auszuwählen.

Benutzen Sie den Schalter  $X^+/X^-$ , um zwischen Tool- und Arm-Auswahlliste zu wechseln und den Schalter  $Z^+/Z^-$ , um den Arm bzw. das Werkzeug entsprechend der jeweiligen Nummer auszuwählen.

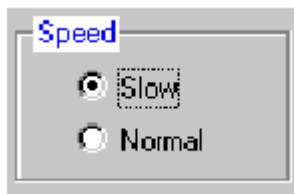
Der Schalter  $U^+/U^-$  hat hier keinerlei Funktion.



- **[Speed]-Gruppe (Geschwindigkeit)**

Hier wird die Bediengeschwindigkeit des Manipulators für Jog&Teach-Aufgaben eingestellt.

Um zwischen den beiden Modi <Slow> (Langsam) und <Normal> zu wechseln, benutzen Sie die Schalter  $Z^+/Z^-$  und  $U^+/U^-$ . Wenn sich der Fokus bewegt, ändert sich auch die Farbe der LED zur Anzeige des Betriebsmodus auf dem JOG-Pad.



- **[Motion Command]-Gruppe (Bewegungsbefehl)**

Diese Gruppe wird benötigt, damit sich der Manipulator anhand der Befehle JUMP oder GO zu den festgelegten Koordinaten der Positionsdaten bewegt.

Der Fokus kann mit Hilfe des Schalters  $X^+/X^-$  auf die Schalter <JUMP P#:Z0>, <GO P#>, <JUMP P#> oder auf das Eingabefeld [+Z] bewegt werden. Ausgeführt werden die einzelnen Befehle dann durch Drücken des ENTER-Schalters, während der Zustimmungstaster gedrückt gehalten wird.

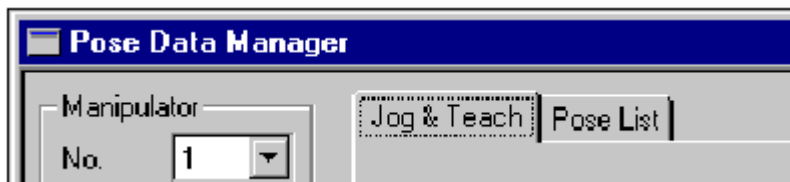
Der Schalter  $Z^+/Z^-$  wird verwendet, um die Positionsdaten-Nummer zu ändern. Befindet sich der Cursor im Eingabefeld [+Z], können Sie den Offset-Wert für +Z ändern.

Der Schalter  $U^+/U^-$  wird benutzt, um die Anzahl der Schritte (1, 10, 100) der Positionsdaten-Nummer zu ändern.



- **Schalten zwischen den Menüs [JOG&Teach] und [Pose List]**

Sie können zwischen den Menüs [JOG&Teach] und [Pose List] hin- und herschalten, wenn Sie den Schalter  $X^+/X^-$  betätigen. Die Positionsdaten können im Menü [Pose List] nicht bearbeitet werden!



- **[Pose Data]-Gruppe (Positionsdaten)**

In dieser Gruppe ist es möglich, Positionsdaten einzurichten (zu registrieren) oder zu löschen.

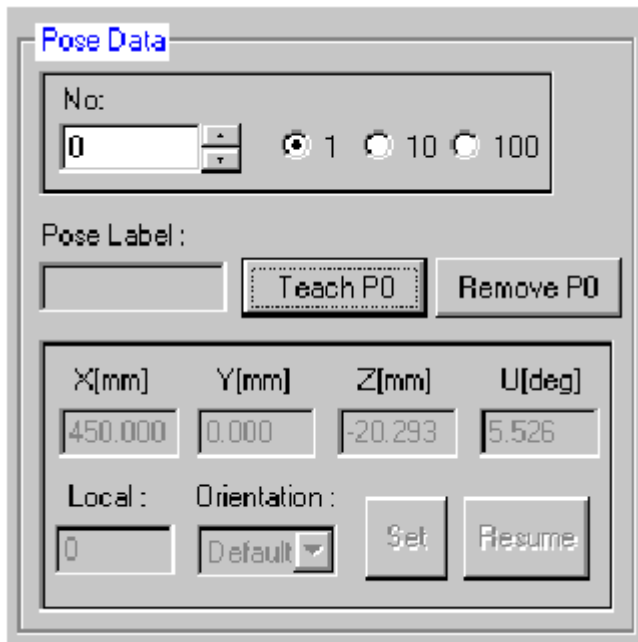
Der Fokus kann mit Hilfe des Schalters  $X^+/X^-$  auf die Schalter <TEACH P#>, <REMOVE P#> oder auf das Eingabefeld [No.] bewegt werden.

Einstellungen im Eingabefeld [No.]:

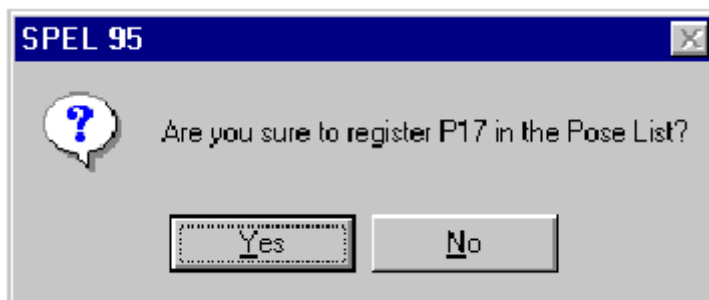
Verwenden Sie den Schalter  $Z^+/Z^-$ , um die Positionsdaten-Nummer zu ändern.

Verwenden Sie den Schalter  $U^+/U^-$ , um die Anzahl der Schritte (1, 10, 100) der Positionsdaten-Nummer zu ändern.

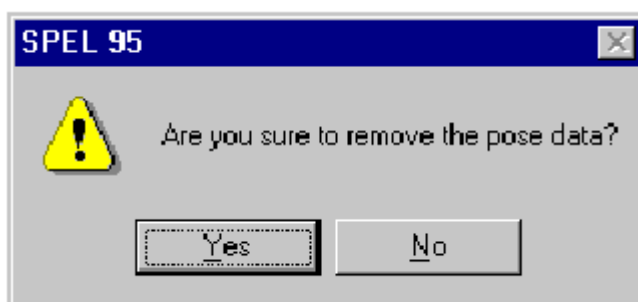




Wenn Sie den Schalter <TEACH P#> auswählen und dabei den ENTER-Schalter drücken, erscheint die nachstehende Meldung. Durch nochmaliges Drücken des ENTER-Schalters wird das Einrichten ausgeführt.



Wenn Sie den Schalter <REMOVE P#> auswählen und dabei den ENTER-Schalter drücken, erscheint die nachstehende Meldung. Durch nochmaliges Drücken des ENTER-Schalters wird das Löschen ausgeführt.



Verwenden Sie den Schalter Z<sup>+</sup>/Z<sup>-</sup>, um die Positionsdaten-Nummer zu ändern.

Verwenden Sie den Schalter U<sup>+</sup>/U<sup>-</sup>, um die Anzahl der Schritte (1, 10, 100) der Positionsdaten-Nummer zu ändern.

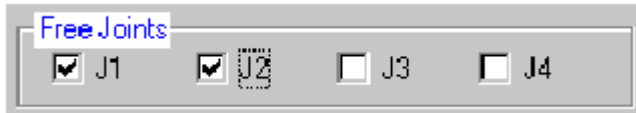
### ▪ [Free Joints]-Gruppe (Freie Gelenke)

Es ist möglich, den Status der Manipulatorgelenke auf servo-frei zu schalten.

Der Fokus kann mit Hilfe des Schalters  $X^+/X^-$  auf die Optionsfelder J1 bis J4 bewegt werden. Markiert werden die einzelnen Schalter dann durch Drücken des ENTER-Schalters, und der Status des entsprechenden Gelenks wechselt in servo-frei.

Wird der Fokus auf ein bereits markiertes Optionsfeld gesetzt und der ENTER-Schalter gedrückt, wird die Markierung entfernt, und der Status wechselt in den Servo-Status.

Die Schalter  $Z^+/Z^-$  und  $U^+/U^-$  haben hier keinerlei Funktion.



### ▪ [JOG Modus]-Gruppe

Diese Gruppe ermöglicht Ihnen, das zu verwendende Jog-Koordinatensystem einzustellen. Werden die Einstellungen verändert, ändert sich entsprechend die Gestalt des Schalters.

Um zwischen den Optionsfeldern zu wechseln, benutzen Sie die Schalter  $Z^+/Z^-$  und  $U^+/U^-$ .

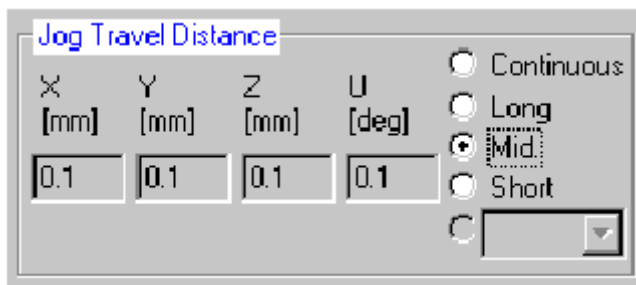
Der Schalter  $X^+/X^-$  hat hier keinerlei Funktion.



### ▪ [JOG Travel Distance]-Gruppe (JOG Bewegungsstrecke)

Diese Gruppe ermöglicht Ihnen, die zurückzulegende Strecke für JOG-Aufgaben einzurichten. Für sequenzielle JOG-Aufgaben wählen Sie <Continuous> (ununterbrochen). Für schrittweise JOG-Aufgaben wählen Sie abhängig von der Anzahl der Schritte <Long> (Lang), <Mid> (Mittel) oder <Short> (Kurz).

Um zwischen den Optionsfeldern zu wechseln, benutzen Sie die Schalter  $Z^+/Z^-$  und  $U^+/U^-$ .



- **[Windows]-Gruppe**

Diese Gruppe ermöglicht Ihnen, zwischen den Menüs [I/O-Monitor] (E/A-Monitor) und [Robot Control] (Robotersteuerung) zu wechseln.

Um zwischen den Schaltern zu wechseln, benutzen Sie den Schalter X<sup>+</sup>/X<sup>-</sup>. Zum Ausführen drücken Sie den ENTER-Schalter.

Die Schalter Z<sup>+</sup>/Z<sup>-</sup> und U<sup>+</sup>/U<sup>-</sup> haben hier keinerlei Funktion.



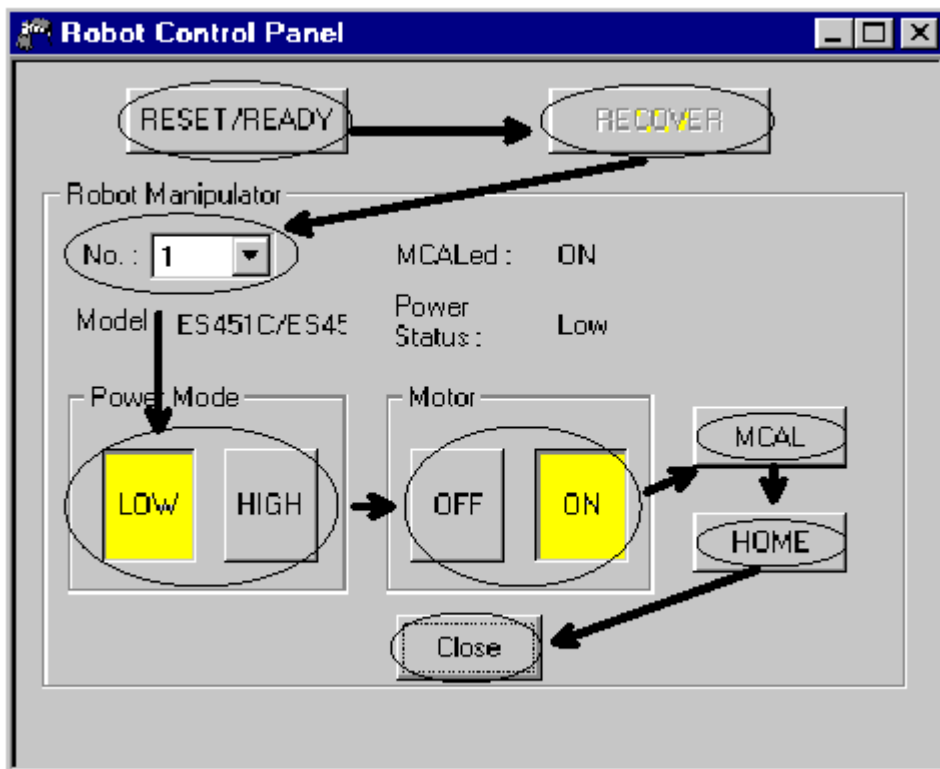
## 2.8.4 [Robot Control]-Panel (Menü: Robotersteuerung)

Verschiedene Funktionen, wie die Wahl des Manipulators und der Zustand der Spannungsversorgung, Ein- und Ausschalten der Motor-Servosteuerung und das Zurückkehren zur Ursprungsposition, werden von hier ausgeführt.

Um das Menü anzeigen zu lassen, während das Einrichten per JOG-Pad durchgeführt wird, gehen Sie wie folgt vor:

Drücken Sie den Schalter  $Y^+/Y^-$  im [Jog&Teach]-Menü, um zur Window-Gruppe zu gelangen. Dann drücken Sie den Schalter  $X^+/X^-$ , um den Fokus auf den Schalter <Robot Control Panel> zu setzen und drücken den ENTER-Schalter, um in das Menü zu gelangen, das Ihnen automatisch am Bildschirm angezeigt wird.

Benutzen Sie die Sie Schalter  $X^+/X^-$ , um den Fokus im Menü in der dargestellten Reihenfolge zu bewegen:



- **<RESET/READY>-Schalter (Reset/Fertig)**

Wurde der Manipulator durch einen NOT-AUS angehalten, wird der Zustand des Manipulators durch Auswahl dieses Schalters zurückgesetzt. Die Einstellungen für Geschwindigkeit und Beschleunigung werden auf die Standardwerte zurückgesetzt.

RESET/READY wird durch Drücken des ENTER-Schalters ausgeführt.

Die Schalter  $X^+/X^-$ ,  $Y^+/Y^-$  und  $U^+/U^-$  haben hier keinerlei Funktion.

- **<RECOVER >-Schalter (Wiederherstellen)**

Der <RECOVER>-Schalter wird benötigt, um den Manipulator zurückzusetzen, wenn er sich nach einem NOT-AUS in einem Bedienverbots-Zustand befindet. Der Schalter ist nur dann aktiv, wenn der Manipulator sich in einem Pausezustand nach einem NOT-AUS befindet.

RECOVER wird durch Drücken des ENTER-Schalters ausgeführt.

Die Schalter  $X^+/X^-$ ,  $Y^+/Y^-$  und  $U^+/U^-$  haben hier keinerlei Funktion.

- **[Manipulator]-Gruppe**

Diese Gruppe wird benötigt, um aus allen angeschlossenen Manipulatoren den auszuwählen, mit dem gearbeitet werden soll.

Benutzen Sie den  $Z^+/Z^-$  -Schalter, um die Nummer des gewünschten Manipulators aus der Auswahlliste der angeschlossenen Manipulatoren auszuwählen. Drücken Sie dann den ENTER-Schalter.

Die Schalter  $Y^+/Y^-$  und  $U^+/U^-$  haben hier keinerlei Funktion.

- **[Power Modus]-Gruppe (Zustand der Spannungsversorgung)**

Hier wechseln Sie den Zustand der Spannungsversorgung. Sie können mit dem Schalter  $U^+/U^-$  zwischen LOW (niedrig) und HIGH (hoch) wählen.

Die Schalter  $X^+/X^-$  und  $Z^+/Z^-$  haben hier keinerlei Funktion.

- **[Motor]-Gruppe**

Hier können Sie die Spannungsversorgung des Manipulators aus- und einschalten.

Sie können die Spannungsversorgung mit den Schaltern  $Z^+/Z^-$  und  $U^+/U^-$  ein- und ausschalten.

Der Schalter  $Y^+/Y^-$  hat hier keinerlei Funktion.

- **[MCAL]-Schalter**

Dieser Schalter initialisiert die mechanische Ursprungsposition (MCAL) des Manipulators.

Das „Rückkehren“ wird ausgeführt durch Drücken des ENTER-Schalters.

Sie können die Spannungsversorgung mit den Schaltern  $Z^+/Z^-$  und  $U^+/U^-$  ein- und ausschalten.

Die Schalter  $X^+/X^-$ ,  $Y^+/Y^-$  und  $U^+/U^-$  haben hier keinerlei Funktion.

- **[HOME]-Schalter**

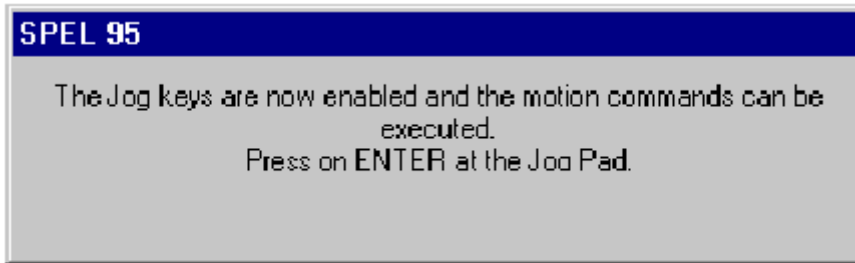
Dieser Schalter verfährt den Manipulator zu den Koordinaten, die mit dem Befehl HOMESSET eingerichtet wurden.

Das Verfahren wird ausgeführt durch Drücken des ENTER-Schalters.

Die Schalter  $X^+/X^-$ ,  $Y^+/Y^-$  und  $U^+/U^-$  haben hier keinerlei Funktion.

### 2.8.5 Bedienung per JOG-Pad

Wird der Zustimmungstaster gedrückt, erscheint die nachstehende Meldung. Drücken Sie den ENTER-Schalter, um die Bedienung per JOG-Pad auszuführen.



#### ▪ Ununterbrochene Bedienung per JOG-Pad

Wurde in der [Jog Travel Distance]-Gruppe <Continuous> ausgewählt, ist eine ununterbrochene Bedienung per JOG-Pad möglich.

Einmaliges schnelles Drücken und Loslassen der JOG-Schalter verfährt den Manipulator folgendermaßen laut der angegebenen Strecke:

- (1) Die Bediengeschwindigkeit ist eingestellt auf „normal“: Verfahrstrecke laut der Einstellung <Long>.
- (2) Die Bediengeschwindigkeit ist eingestellt auf „slow“: Verfahrstrecke laut der Einstellung <Short>.

Der Manipulator verfährt in der angegebenen Strecke, bevor er anfängt, ununterbrochen zu arbeiten, wenn der Schalter gedrückt gehalten wird, anstatt ihn wieder loszulassen.

Der Manipulator hält unverzüglich an, wenn der Zustimmungstaster oder der JOG-Schalter wieder losgelassen wird.

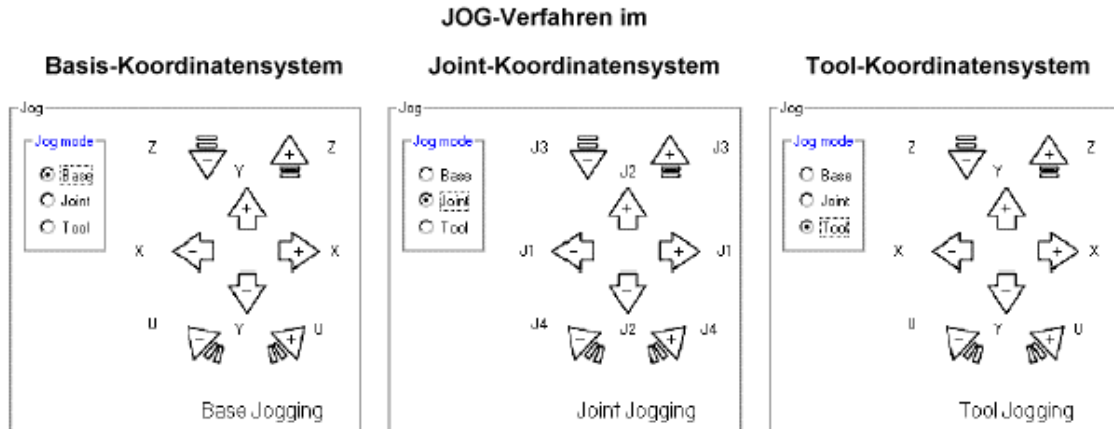
In der nachfolgenden Tabelle wird jeder Schalter für den entsprechenden JOG-Modus erläutert:

	Basis-Modus	Joint-Modus	Tool-Modus
Schalter X <sup>+</sup> /X <sup>-</sup>	X-Achse des Basis-Koordinatensystems	J1	X-Achse des Tool-Koordinatensystems
Schalter Y <sup>+</sup> /Y <sup>-</sup>	Y-Achse des Basis-Koordinatensystems	J2	Y-Achse des Tool-Koordinatensystems
Schalter Z <sup>+</sup> /Z <sup>-</sup>	Z-Achse des Basis-Koordinatensystems	J3	Z-Achse des Tool-Koordinatensystems
Schalter U <sup>+</sup> /U <sup>-</sup>	U-Achse des Basis-Koordinatensystems	J4	U-Achse des Tool-Koordinatensystems

### ▪ Schrittweise Bedienung per JOG-Pad

Um eine schrittweise Bedienung auszuführen, muss zuerst einer der folgenden Modi in der [Jog Travel Distance]-Gruppe ausgewählt werden: <Long>, <Mid> oder <Short>. Dann drücken Sie den JOG-Schalter, während Sie den Zustimmungstaster gedrückt halten.

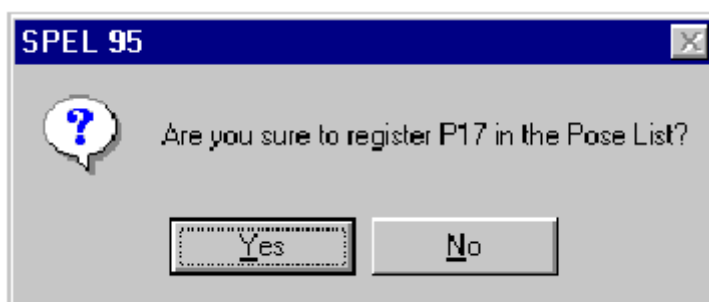
Darstellung des jeweiligen Jog-Modus am Bildschirm:



### ▪ Einrichten

Nachdem der Manipulator in die gewünschte Position verfahren wurde, richten Sie die Position folgendermaßen ein:

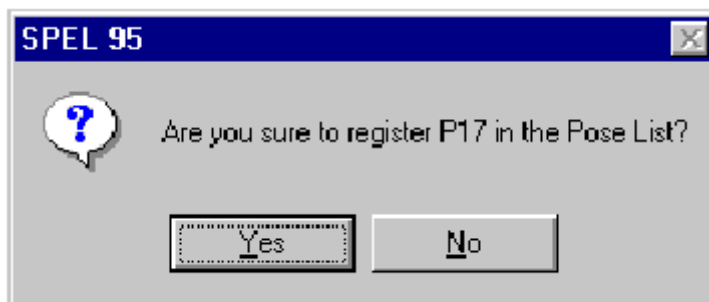
- (1) Lassen Sie den Zustimmungstaster los.
- (2) Bewegen Sie den Fokus mit Hilfe des Schalters Y<sup>+</sup>/Y<sup>-</sup> auf die [Pose Data]-Gruppe.
- (3) Bewegen Sie den Fokus mit Hilfe des Schalters X<sup>+</sup>/X<sup>-</sup> auf das Eingabefeld [No.], geben Sie eine Positionsdaten-Nummer ein, die Sie mit dem Schalter Z<sup>+</sup>/Z<sup>-</sup> einrichten wollen.
- (4) Bewegen Sie den Fokus mit Hilfe des Schalters X<sup>+</sup>/X<sup>-</sup> auf den Schalter [Teach P#].
- (5) Drücken Sie den ENTER-Schalter.
- (6) Die nachstehende Meldung erscheint. Drücken Sie dann noch einmal den ENTER-Schalter.



### ▪ Direktes Einrichten

Um direktes Einrichten durchführen zu können, müssen alle Gelenke des Manipulators im servo-freien Zustand sein.

- (1) Lassen Sie den Zustimmungstaster los.
- (2) Bewegen Sie den Fokus mit Hilfe des Schalters  $Y^+/Y^-$  auf die Gruppe [Free Joints].
- (3) Wählen Sie ein Gelenk, das mit Hilfe des Schalters  $X^+/X^-$  servo-frei geschaltet werden soll.
- (4) Führen Sie den servo-frei-Status durch, indem Sie den ENTER-Schalter betätigen.
- (5) Wiederholen Sie die Schritte (2) – (4) für alle Gelenke, die Sie in den servo-frei-Status versetzen wollen.
- (6) Bewegen Sie den Arm manuell in die gewünschte Position.
- (7) Bewegen Sie den Fokus mit Hilfe des Schalters  $Y^+/Y^-$  auf die Gruppe [Pose Data].
- (8) Bewegen Sie den Fokus mit Hilfe des Schalters  $X^+/X^-$  auf das Eingabefeld [No.], und stellen Sie dann eine Positionsdaten-Nummer ein, die mit Hilfe des Schalters  $Z^+/Z^-$  registriert werden soll.
- (9) Bewegen Sie den Fokus mit Hilfe des Schalters  $X^+/X^-$  auf den Schalter <TEACH P#>.
- (10) Drücken Sie den ENTER-Schalter.
- (11) Die nachfolgende Meldung erscheint. Drücken Sie den ENTER-Schalter noch einmal.





## 2.9 Bedienung mit dem Handprogrammiergerät

Das Handprogrammiergerät (OP500) ist ein mit einem LCD ausgestattetes Bediengerät mit Touchpanel.

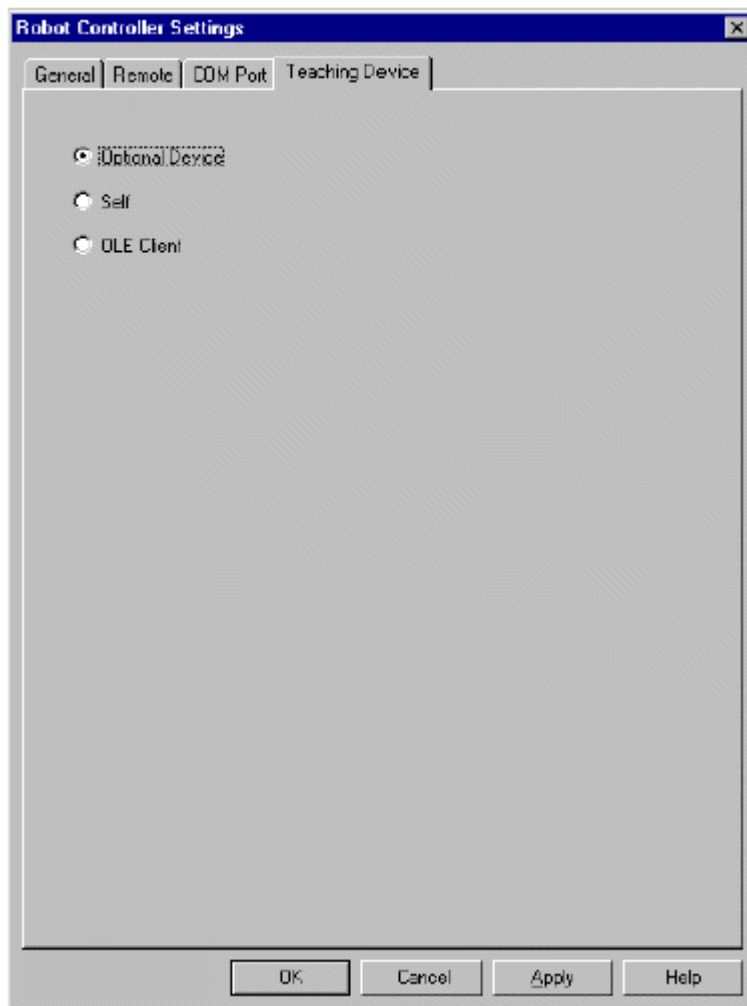
Mit dem OP500 lassen sich Einrichtoperationen effizienter durchführen, da auf der Anzeige mehrere Informationen gleichzeitig dargestellt werden.

Weitere Informationen erhalten Sie im Handbuch zum Handprogrammiergerät OP500.

In diesem Kapitel werden die generelle JOG-Bedienung und die Einrichtmethoden mit dem Handprogrammiergerät beschrieben.

### 2.9.1 SPEL 95 - Einstellungen

- (1) Stellen Sie sicher, dass der Stecker am Kabel des Handprogrammiergerätes sicher am Anschluss „Optional Device“ (optionales Gerät) der SRC5\*\*-Steuerungseinheit angeschlossen ist.
- (2) Wählen Sie das Optionsfeld <OPTIONAL DEVICE> im Menü [Teaching Device] (Einrichtgerät) im [Setup]-[Robot Controller Settings]-Dialog (Einrichtung –Einstellung Robotersteuerung).



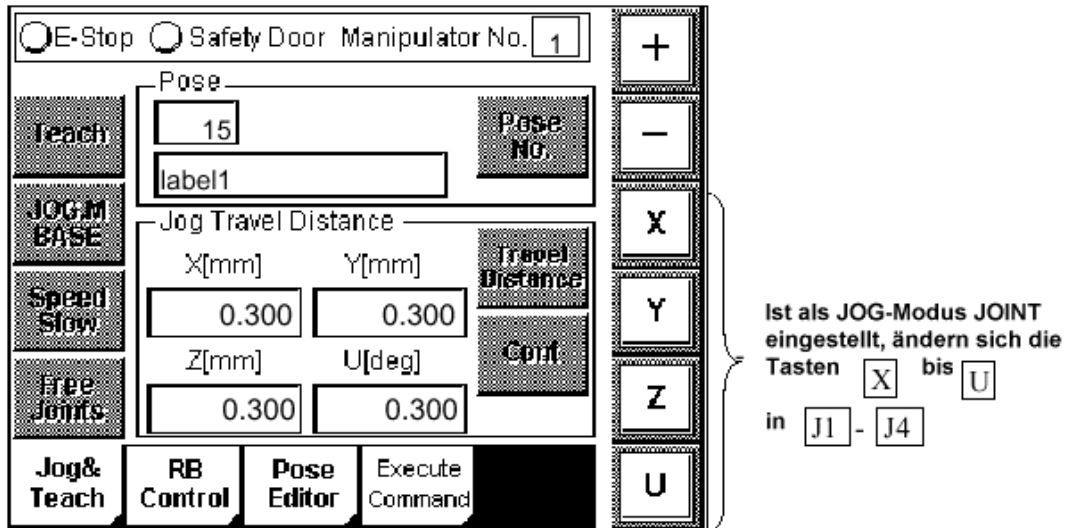
## 2.9.2 JOG-Bedienung per OP500

Drehen Sie den Modusauswahl-Schalter am Handprogrammiergerät auf ATTEND (Wartungsmodus).

### ▪ Schrittweise JOG-Bedienung

- (1) Um das Menü [Jog&Teach] anzeigen zu lassen, berühren Sie die Taste [Jog&Teach] auf dem OP500.

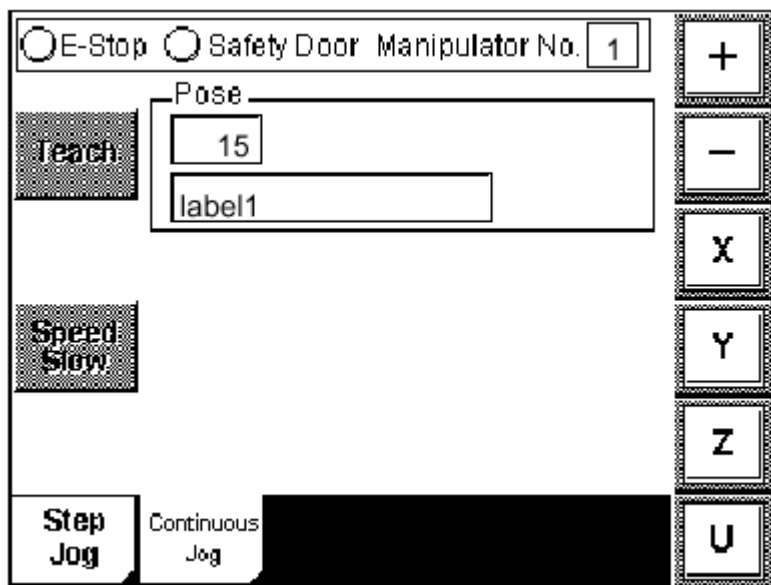
Operation  
Pendant



- (2) Um die Achse auszuwählen, an der die JOG-Bedienung ausgeführt werden soll, drücken Sie die Taste  $\odot\ominus\odot$  oder  $\odot$ . Drücken der Tasten  $\otimes$  oder  $\otimes$  verfährt den Manipulator in Abhängigkeit der Entfernungsstrecke, deren Wert angezeigt wird, wenn der Zustimmungstaster gedrückt wird.

### ▪ Ununterbrochene JOG-Bedienung

- (1) Um das Menü [Continuous Jog] anzeigen zu lassen, berühren Sie die Taste [Cont] im Menü [Jog&Teach].







- (2) Um die Achse auszuwählen, an der die ununterbrochene JOG-Bedienung ausgeführt werden soll, drücken Sie die Taste  $\odot\ominus\odot$  oder  $\odot$ . Drücken der Tasten  $\otimes$  oder  $\otimes$  verfährt den Manipulator in

Abhängigkeit der Entfernungsstrecke, deren Wert angezeigt wird, wenn der Zustimmungstaster gedrückt wird.

Die Entfernungsstrecke variiert in Abhängigkeit von der eingestellten Geschwindigkeit.

Niedrige Geschwindigkeit - Entfernungsstrecke = kurz

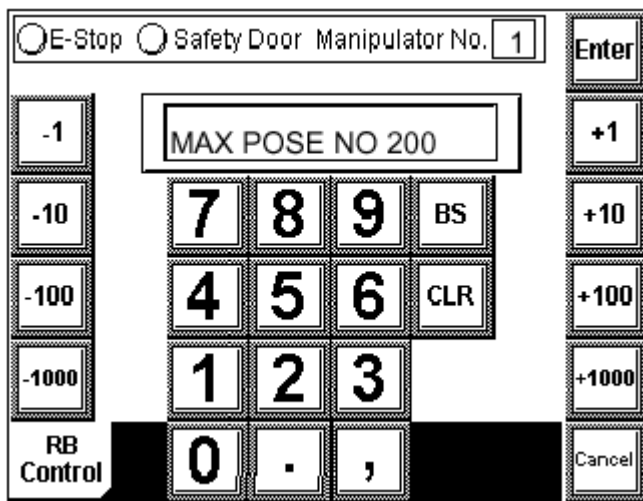
Normale Geschwindigkeit - Entfernungsstrecke = lang

- (3) Nachdem der Manipulator die eingestellte Strecke zurückgelegt hat, startet ein ununterbrochenes Drücken der   Tasten den ununterbrochenen Betrieb.
- (4) Wird der Zustimmungstaster losgelassen oder voll gedrückt, oder wenn die Tasten   freigegeben werden, hält der Manipulator sofort an.

### 2.9.3 Einrichten

Nachdem der Manipulator in Zielposition verfahren wurde, richten Sie die Positionsdaten für diese Position ein.

- (1) Berühren Sie die Taste <Pose No.> (Positionsnummer) im Menü [Jog&Teach].



- (2) Wählen Sie die zu speichernde Nummer für die Positionsdaten aus.
- (3) Berühren Sie die Taste <Enter>. Das [Jog&Teach]-Menü wird angezeigt.
- (4) Berühren Sie die Taste <Teach> (Einrichten) im Menü [Jog&Teach].

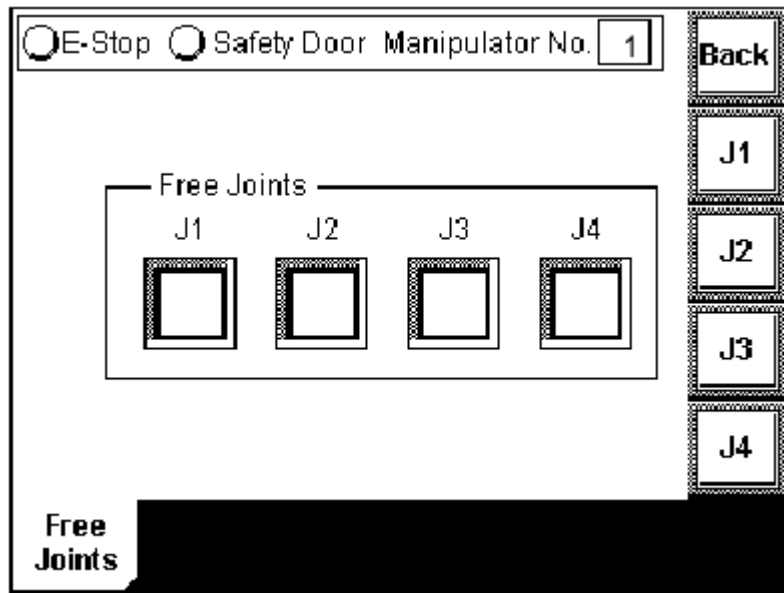


- (5) Wird die Taste <Yes> gedrückt, wird die aktuelle Position jedes Gelenks des Manipulators in der festgelegten Positionsnummer gespeichert.

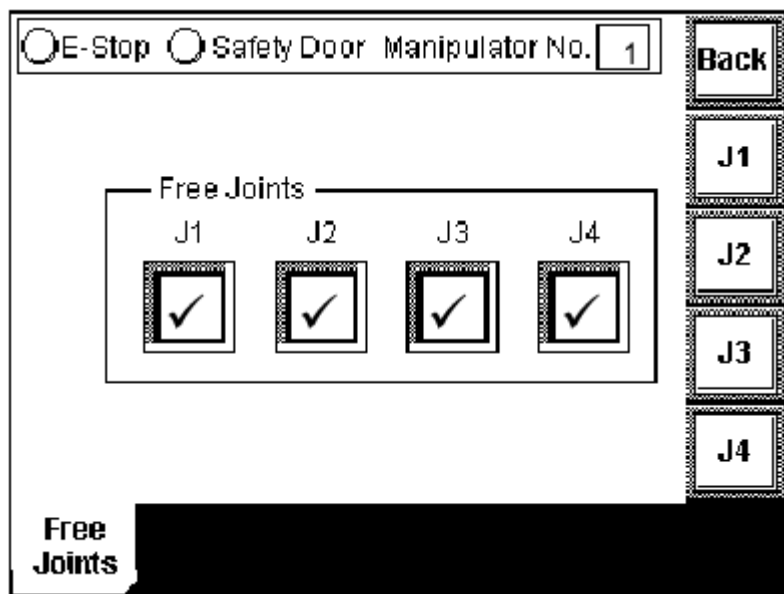
## 2.9.4 Direktes Einrichten

Um direktes Einrichten ausführen zu können, müssen Sie alle anliegenden Zustände an den Gelenken des Manipulators stornieren.

- (1) Berühren Sie die Taste <Free Joint> (freies Gelenk) im Menü [Jog&Teach]. Daraufhin wird das Menü [Free Joints] angezeigt.



- (2) Um die anliegenden Zustände an den einzurichtenden Gelenken außer Kraft zu setzen, berühren Sie die Tasten <J1> - <J4>.



Ein Häkchen wird in jedem Optionsfeld angezeigt. So wird deutlich, welche Gelenke frei sind. So freigegebene Gelenke können per Hand bewegt werden.

- (3) Um in das Menü [Jog&Teach] zurückzugelangen, drücken Sie die Taste <Back> (Zurück).
- (4) Bewegen Sie den Arm manuell in die Zielposition.

- (5) Berühren Sie die Taste <Teach> im Menü [Jog&Teach].

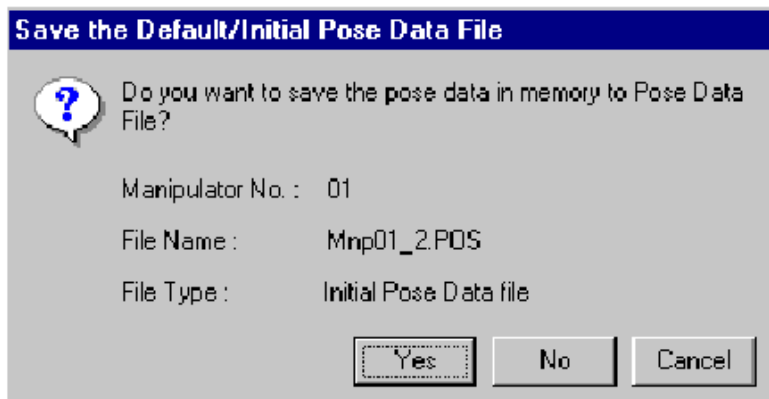


- (6) Wird die Taste <Yes> gedrückt, wird die aktuelle Position jedes Gelenks des Manipulators in der festgelegten Positionsnummer gespeichert.

## 2.10 Speichern der Positionsdaten am Ende der Arbeit

Wenn die Positionsdaten per [Pose Data Manager]-Menü oder Jog Pad verändert bzw. überarbeitet worden sind, erscheint unten dargestellte Meldung, wenn Sie am Ende der Arbeit im [Pose Data Manager]-Menü auf den Schalter <Close> (Schließen) geklickt oder am Jog Pad den Schlüsselschalter auf Normal-Position gestellt haben. In der Meldung wird abgefragt, ob Sie die bearbeiteten Positionsdaten als Anfangs- oder Standard-Positionsdaten-Datei speichern wollen.

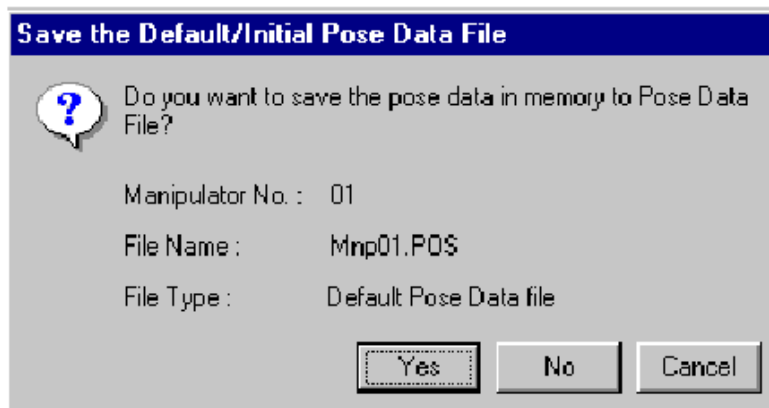
Haben Sie die Anfangs-Positionsdaten-Datei bearbeitet, erscheint die folgende Meldung:



Klicken Sie auf <Yes>, um die Daten in der Anfangs-Positionsdaten-Datei zu speichern. Das Fenster [Pose Data Manager] wird geschlossen. Die Daten werden in die Anfangs-Positionsdaten-Datei gespeichert.

Klicken Sie auf <No>, wenn Sie die Daten nicht in der Anfangs-Positionsdaten-Datei speichern wollen. Das Fenster [Pose Data Manager] wird geschlossen. Die Daten werden im Speicher der Steuerung gespeichert, aber die Anfangs-Positionsdaten-Datei wird nicht upgedatet.

Haben Sie die Standard-Positionsdaten-Datei bearbeitet, erscheint die folgende Meldung:



Klicken Sie auf <Yes>, um die Daten in der Standard-Positionsdaten-Datei zu speichern. Das Fenster [Pose Data Manager] wird geschlossen. Die Daten werden in die Standard-Positionsdaten-Datei gespeichert.

Klicken Sie auf <No>, wenn Sie die Daten nicht in der Standard-Positionsdaten-Datei speichern wollen. Das Fenster [Pose Data Manager] wird geschlossen. Die Daten werden im Speicher der Steuerung gespeichert, aber die Standard-Positionsdaten-Datei wird nicht upgedatet.

Der Manipulator verfährt jedoch zu der neuen Positionsdatenposition, wenn ein Bewegungsbefehl ausgeführt wurde, da die Positionsdaten auf jeden Fall im Speicher bleiben.

Beim erneuten Öffnen des Menüs [Pose Data Manager] werden die Daten angezeigt, die dort vorhanden waren, bevor das Menü geschlossen wurde!

## 2.11 Sicherungskopie der Positionsdaten-Dateien

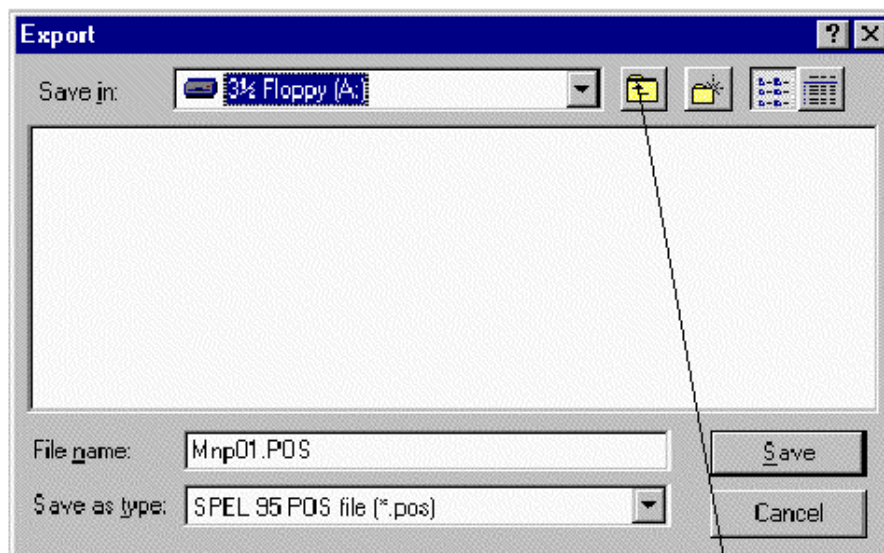
Das Speichern der eingerichteten Daten auf z.B. eine Diskette kann den Datenverlust minimieren, wenn z.B. die Festplatte zerstört wird.

Wählen Sie eine Positionsdaten-Datei aus der Auswahlliste [Files in the project] aus.

Wählen Sie aus dem Menü [File] (Datei) den Befehl [Export] (Exportieren) aus.



Wählen Sie dann aus, wohin die Daten gespeichert werden sollen. Legen Sie eine formatierte Diskette in das Laufwerk A ein. Wählen Sie 3½ Floppy (A) aus, um die Daten auf Diskette zu speichern, und vergeben Sie als Dateinamen einen zutreffenden Namen. Klicken Sie dann auf den Schalter <Save> (Speichern).



**Jedes Mal, wenn dieses Symbol angeklickt wird, wechselt man in den jeweils übergeordneten Ordner in der Verzeichnishierarchie. Klicken Sie solange auf dieses Symbol, bis Sie die 3,5"-Diskette auswählen können.**

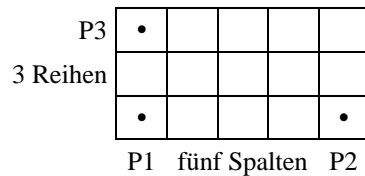
## 2.12 Definition einer Palette

Um die Paletten-Funktion nutzen zu können, ist die Einstellung der Positionsdaten notwendig. Für die Einstellungen richten Sie ein Rechteck mit vier Ecken ein und definieren die Anzahl der Teilungen (Reihen und Spalten) in vertikaler und horizontaler Ausrichtung.

Der Befehl PALET wird für Bewegungen zu mehreren, gleich weit voneinander entfernten Punkten verwendet.

### PALET

<b>Funktion</b>	Definiert bzw. zeigt Paletten an.
<b>Format</b>	PALET {[Paletten-Nr.] P[Punkt 1],P[Punkt 2],P[Punkt 3]{,P[Punkt 4]}-[Anzahl Reihen],[Anzahl Spalten]}
<b>Beschreibung</b>	<p>Die Paletten-Nr. muss eine ganze Zahl zwischen 0 und 15 sein.  [Anzahl Reihen] ist die Anzahl der Reihen von P[Punkt 1] bis P[Punkt 2]  [Anzahl Spalten] ist die Anzahl der Spalten von P[Punkt1] bis P[Punkt 3]</p> <p>Definiert eine Palette, indem die entsprechenden Punktdaten (für mindestens drei Punkte 1, 2 und 3) im TEACH-Modus eingerichtet werden. Dabei wird die Anzahl der Reihen und Spalten von Punkt 1 zu Punkt 2 und von Punkt 1 zu Punkt 3 angegeben.</p> <p>Hat die Palette eine regelmäßige rechteckige Form, brauchen nur drei der vier Eckpunkte definiert zu werden. Bei einer Palette mit weniger regelmäßiger Form müssen dementsprechend alle vier Eckpunkte angegeben werden.</p> <p>Zur Palettendefinition werden zuerst 3 oder 4 Eckpunkte im TEACH-Modus eingerichtet.</p>
<b>Beispiel</b>	PALET1 , P1 , P2 , P3 , 5 , 3





▪ Festlegen von Positionen innerhalb einer Palette

**PALET(n)**

**Funktion** Gibt die Position in der festgelegten Palette an. Wird immer mit einem Bewegungsbefehl kombiniert.

**Format** PALET [Paletten-Nr.] (|[Divisionspunktnr.] |)|  
 (|[Divisionspunktnr. Horizontale Koordinate], ? ?  
 [Divisionspunktnr. Vertikale Koordinate])|

Die Paletten-Nr. muss eine ganze Zahl zwischen 0 und 15 sein.  
 Die Divisionspunktnr. muss eine ganze Zahl zwischen 1 und 32767 sein.  
 Die Divisionspunktnr. für die horizontale und vertikale Koordinate werden festgelegt durch die Definition einer Palette.

**Beschreibung** Es gibt zwei Möglichkeiten, Divisionspunkte zu definieren:

Definition durch Divisionspunktnummern

P3	11	12	13	14	15
	6	7	8	9	10
	1	2	3	4	5
	P1			P2	

Definition durch Divisionskoordinaten

P3	1,3	2,3	3,3	4,3	5,3
	1,2	2,2	3,2	4,2	5,2
	1,1	2,1	3,1	4,1	5,1
	P1			P2	

**Beispiel** >JUMP PALET1 (5)  
 JUMP PALET1 (1,5)

**Übung** • • PALET

**Tätigkeit** Bewege das Werkstück von Palette 1 zu Palette2.

P3	P3	13	14	15	P13	11	12	13	14	15
		10	11	12		6	7	8	9	10
		7	8	9		1	2	3	4	5
		4	5	6		P11			P12	
P1	P1	1	2	3	P2					

**Programm:**

```

FUNCTION MAIN
    PALET1, P1,P2,P3,3,5
    PALET2, P11,P12,P13,5,3
    FOR I=1 TO 15
        JUMP PALET1(I)
        ON 1 ; WAIT 0,5
        JUMP PALET2(I)
        OFF 1 ; WAIT 0,5
    NEXT I
FEND
    
```

## Kapitel 3

---

# Überblick über die Programmierung

### 3.1 Befehle und Anweisungen

Die folgenden befehls- und anweisungsbezogenen Ausdrücke werden wie folgt definiert:

Befehl:	Ein Befehl, der unter Windows 95 ausführbar ist.
SPEL 95-Befehl:	Ein SPEL 95-spezifischer Befehl, der direkt aus dem [Command Execution]-Menü (Befehlsausführung) des Debug-Fensters ausführbar ist.
SPEL 95-Anweisung:	Eine Anweisung, die innerhalb eines SPEL 95-Programms aufgeführt ist. Wenn wir in diesem Handbuch über eine „Anweisung“ sprechen, ist immer von einer SPEL 95-Anweisung die Rede.
SPEL 95-Instruktion:	Kann als SPEL 95-Befehl oder Anweisung verstanden werden.

Beispiel: „JUMP P1“-Beschreibung

„JUMP P1“ ist ein SPEL 95-Befehl, der die Ausführung „JUMP P1“ aus dem [Command Execution]-Menü des Debug-Fensters ermöglicht. Es handelt sich gleichzeitig um eine Anweisung, die in einem Programm stehen kann.

Beispiel: „FOR i=0 TO 15“-Beschreibung

„FOR i=0 TO 15“ kann in einem Programm stehen, kann aber nicht aus dem [Command Execution]-Menü ausgeführt werden.

## 3.2 Programmier-Regeln

### 3.2.1 Grundlegende Bestandteile eines Programms

Die Grundeinheit eines Programms wird „Zeile“ genannt, und eine Anzahl von Zeilen macht ein Programm aus. Die Zeilen selbst bestehen aus Zeilennummern und Anweisungen.

Ein Programm beginnt immer mit der 1. Zeile „FUNCTION MAIN“ und endet mit der letzten Zeile „FEND“.

```
<Beispiel>          FUNCTION MAIN

                    FEND
```

Alles, was von FUNCTION bis FEND steht, wird eine Funktion genannt. Nach Eingabe von FUNCTION ist ein Name, im Beispiel MAIN, erforderlich.

### 3.2.2 Funktionsname

Im Folgenden werden die Beschränkungen bei der Verwendung von Funktionsnamen aufgeführt:

- Reservierte Worte (z. B. Befehle, Anweisungen und Funktionen) dürfen nicht verwendet werden. Auch, wenn Sie reservierte Worte durch Unterstriche oder Zahlen ergänzen oder verändern, werden sie trotzdem als reservierte Worte erkannt.
- Groß- und Kleinschreibung wird nicht unterschieden.
- Als verwendbare Zeichen sind alle alphanumerischen Zeichen und Unterstriche ( \_ ) zugelassen.
- Der Funktionsname darf aus max. 15 Zeichen bestehen.

### 3.2.3 Mehrfachanweisung

Wird das Semikolon (;) benutzt, können mehrere Befehle in eine Zeile geschrieben werden. Dies wird Mehrfachanweisung genannt.

Mit Hilfe der Mehrfachanweisungen können Programme leichter gelesen werden, und die Programmgröße wird verringert. Auch ist die Ausführzeit geringer. Werden jedoch unwichtige Mehrfachanweisungen eingegeben, kann das Programm unübersichtlich und schwer verständlich werden.

```
<Beispiel>
    JUMP P7 ; ON 1 ; WAIT 0,2
    JUMP P17 ; OFF 1 ; WAIT 0,2
```

### 3.2.4 Kommentare

Um ein Programm verständlicher zu machen, können Sie, wenn notwendig, Kommentare einfügen. Das Symbol, das einen Kommentar einleitet, ist das Apostroph (^). Alle Zeichen, die auf ein Apostroph folgen, werden als Kommentar berücksichtigt. Bei Ausführung des Programms werden alle Kommentare, die auf ein Apostroph folgen, außer Acht gelassen.

Als Kommentar können Sie jedes Zeichen benutzen, allerdings darf der Kommentar, einschließlich Zeilennummer, Anweisung und Text, 255 Zeichen nicht überschreiten.

<Beispiel>

```

FUNCTION MAIN
\ *****
\ *           HANDLER-1           *
\ *programmiert von SATO *
\ *****
\ GOSUB 1000
SPEED 100 ;ACCEL 50,100
JUMP P10 ;ON 1 ;WAIT 0,2 `Nimmt
Objekt von Einzug1 auf.

```

}      Kommentarzeilen  
←GOSUB wird nicht ausgeführt!  
←Kommentar

### 3.2.5 Sprungmarken

Wenn Sie die Programmausführungsreihenfolge mit Hilfe von GOTO, GOSUB oder IF...THEN...ELSE ändern, wird das Programm gemäß festgelegter sinnvoller Wörter zum Definieren von Zielorten ausgeführt. Die Zeilennummer ist nur eine Nummer und hat keinerlei weitere Bedeutung. Wenn Sie ein Wort benutzen können, um eine Bewegung anzuzeigen, ist dies für den Programmierer und den Benutzer viel angenehmer.

In SPEL 95 können Sie eine Sprungmarke für die GOTO, GOSUB oder IF...THEN...ELSE Anweisungen verwenden.

Im Folgenden werden die Beschränkungen bei der Verwendung von Sprungmarken aufgeführt:

- Reservierte Worte (z. B. Befehle, Anweisungen und Funktionen) dürfen nicht verwendet werden. Auch, wenn Sie reservierte Worte durch Unterstriche oder Zahlen ergänzen oder verändern, werden sie trotzdem als reservierte Worte erkannt.
- Groß- und Kleinschreibung wird nicht unterschieden.
- Der Name der Sprungmarke darf aus max. 32 Zeichen bestehen.

<Beispiel>

```

FUNCTION MAIN
LOOP1:                      ←Sprungmarke
    JUMP P1
    JUMP P2
GOTO LOOP1                  ←Verzweigung mit Hilfe der Sprungmarke zu LOOP 1
FEND0

```

Im Beispiel ist in der zweiten Zeile die Sprungmarke „LOOP“ aufgeführt. Erst der Doppelpunkt (:) hinter LOOP zeigt an, dass es sich tatsächlich um eine Sprungmarke handelt.

### 3.2.6 Zeilennummern

Zeilennummern werden nicht benötigt, können jedoch für GOSUB, GOTO oder andere Anweisungen als Sprungmarken verwendet werden:

<Beispiel>

```

FUNCTION MAIN
10 FOR I01 TO 3
    JUMP P(I)
NEXT I
GOTO 10
FEND

```

### 3.3 E/A-Sprungmarken und Positionsmarken

E/A-Sprungmarken und Positionsmarken können eingestellt werden, um vom Programm benutzt oder darauf bezogen zu werden.

#### 3.3.1 E/A-Sprungmarke

##### ▪ Einstellungen der E/A-Sprungmarke

Die E/A-Sprungmarke erzeugt eine E/A-Sprungmarken-Datei, die im Menü [I/O Label File] registriert wird. Nähere Informationen zum Erzeugen und Editieren einer E/A-Sprungmarken-Datei erhalten Sie in Kapitel 5.3 „Editieren von E/A-Sprungmarken-Dateien“.

Nähere Informationen zum Registrieren einer E/A-Sprungmarken-Datei erhalten Sie in Kapitel 4.4.4 „Ändern von E/A-Sprungmarken-Dateien“.

##### ▪ Bezug auf eine E/A-Sprungmarke

Um sich auf eine vom Anwender erstellte oder im Speicher befindliche E/A-Sprungmarke beziehen zu können, verwenden Sie bitte folgende Funktionen:

IONUM():	Diese Funktion gibt eine E/A-Nummer zurück, wenn es sich bei der E/A-Sprungmarke, nach der gesucht wird, um eine Zeichenfolge handelt.
IOLABEL\$( )::	Diese Funktion gibt eine E/A-Sprungmarke zurück, wenn der E/A-Typ oder die Nummer definiert ist.

#### IONUM()

<b>Funktion</b>	Gibt eine E/A-Nummer zurück, wenn es sich bei der E/A-Sprungmarke, nach der gesucht wird, um eine Zeichenfolge handelt.	
<b>Format</b>	IONUM(Name der Sprungmarke)	
	Name der Sprungmarke:	Die E/A-Sprungmarke unterscheidet nicht zwischen: Anwender E/A, Speicher E/A und Referenz Bit/Byte/Wort
<b>Beschreibung</b>	Gibt eines der folgenden zurück:	Bitnummer, auf die sich die E/A-Sprungmarke bezieht, Bytenummer oder Wortnummer
<b>Beispiel</b>	Angenommen, dass „CYLINDER1“ eingestellt ist für entweder die Anwender E/A, Speicher E/A oder Referenz Bit/Byte/Wort, ist die Einstellung für Ausgangsnummer „2“ „CYLINDER1“.	
	<pre>PRINT IONUM( „CYLINDER“ )</pre>	
	2	? Ergebnis

#### IOLABELS\$( )

<b>Funktion</b>	Gibt eine E/A-Nummer zurück, wenn der E/A-Typ oder die Nummer definiert ist.	
<b>Format</b>	IOLABELS(Typ, Bitlänge, Ausgangsnummer)	
	Typ	0: Anwender E/A (Eingang) 1: Anwender E/A (Ausgang) 2: Speicher E/A
	Bitlänge	1: Bit 8: Byte 16: Wort
	Ausgangsnummer	
<b>Beschreibung</b>	Gibt eine E/A-Sprungmarke zurück, die durch Typ, Bitlänge oder Ausgangsnummer gesucht wird.	
<b>Beispiel</b>	<pre>PRINT IOLABEL\$( 1 , 1 , 2 )</pre>	
	CYLINDER1	? Ergebnis



## 3.4 Variablen

Variablen werden als vorübergehende Ersatzdaten verwendet. Es gibt viele verschiedene Arten von Variablen, daher muss vor dem Benutzen einer Variable eine Variablendefinition eingegeben werden. Der Wert, durch den die Variable ersetzt wird, muss vom selben Typ sein, wie die Variable selbst.

### 3.4.1 Variablenarten

Lokale Variable:	Variablen, die in der gleichen Funktion benutzt werden können.
Globale Variable:	Variablen, die in der gleichen Gruppe benutzt werden können.
Backup-Variablen:	Variablen, die vom selben Projekt benutzt werden können und von denen eine Sicherungskopie erstellt werden kann.

Um diese Variablen verwenden zu können, müssen Sie in einer Anweisung angegeben werden. Z. B.:

Lokale Variable:	Der Bereich für alle diese Variablen zusammen darf eine Größe von 8 KB nicht überschreiten.
Globale Variable:	Bis zu 256 Variablen. Ihre Größe darf 8 KB nicht überschreiten.
Backup-Variablen:	Bis zu 32 Variablen. Ihre Größe darf 4 KB nicht überschreiten.

Die Variablendeklarationen können jedoch nicht endlos geschrieben werden. Maximale Anzahl für die Variablen:

Lokale Variable:	DIM [Variablenname].....AS [Variablenart]
Globale Variable:	GLOBAL [Variablenname].....AS [Variablenart]
Backup-Variablen:	SYS [Variablenname].....AS [Variablenart]

Nähere Informationen hierzu erhalten Sie in Kapitel 4.6.4 „Einrichten von Projekten – Einrichten von Variablen“.

### 3.4.2 Bereichsaufteilung für lokale Variablen

Der maximal zur Verfügung stehende Bereich für lokale Variablen beträgt standardmäßig 8 KB. Übersteigt der Gesamtbereich, der von den lokalen Variablen verwendet wird, während der Programmausführung den Standardwert, tritt ein Fehler auf. Daher muss der Bereich, der für die lokalen Variablen vonnöten ist, vorher bekannt sein. Dies wird in Kapitel 6.5 „Einstellung der Variablen – Überprüfen der Bereichsaufteilung für lokale Variablen“ näher erläutert.

### 3.4.3 Numerische Variablen

Es gibt verschiedene Arten von numerischen Variablen (siehe Tabelle):

	Beschreibung	Bytes	verfügbarer Wert/Gültige Ziffer
Ganze Zahlen	BYTE	1	-128 bis 127
	INTEGER	4	-2147483648 bis 2147483647*
Reelle Zahlen	REAL	4	7 Ziffern
	DOUBLE	8	14 Ziffern

Wenn Sie eine Variable definieren, müssen Sie zuerst die Definition und dann den Namen eingeben. Wenn verschiedene Variablen vom gleichen Typ definiert werden sollen, benutzen Sie bitte ein Komma „;“, um verschiedene Variablennamen zu vergeben (siehe Beispiel). Dahinter wird eine AS-Typdefinition (als Variable vom Typ) eingefügt. Die Variablendefinition für einen Typ muss in einer Zeile erfolgen. Um einen weiteren Typ zu definieren, benutzen Sie bitte eine weitere Zeile.

Wird keine Typendefinition durchgeführt, wird die Variable als reelle Zahl angenommen.

Format: [Variablendefinition] [Variablenname]{,[Variablenname]}n AS [Typdefinition]

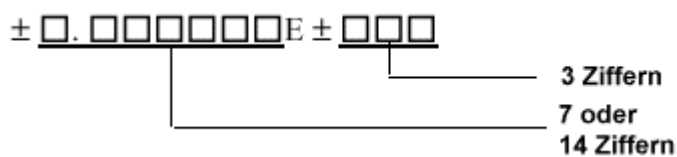
### ▪ Variablenname

Es gibt jeweils einen Variablennamen pro Variable. Variablennamen unterliegen den folgenden Beschränkungen:

- Reservierte Worte (z. B. Befehle, Anweisungen und Funktionen) dürfen nicht verwendet werden. Auch, wenn Sie reservierte Worte durch Unterstriche oder Zahlen ergänzen oder verändern, werden sie trotzdem als reservierte Worte erkannt.
- Das erste Zeichen muss ein Buchstabe sein.
- Groß- und Kleinschreibung wird nicht unterschieden.
- Der Variablenname darf aus max. 64 Zeichen bestehen.

### ▪ Ausdruck von reellen Variablen

Bei reellen Variablen wird der Wert durch Ziffern dargestellt.



### ▪ Schreiben eines aktuellen numerischen Wertes in ein Programm

Beschreibt den numerischen Wert durch einen aktuellen numerischen Wert in einem Programm. Der numerische Wert wird betrachtet als: INTEGER (wenn es sich um eine ganze Zahl handelt) oder als DOUBLE (wenn es sich um eine reelle Zahl handelt).

In dem folgenden Beispiel ist 0,1 in der Zeile „1=0,1“ der reellen Variablen „a“ zugewiesen. In der folgenden Anweisung „IF a = 0,1 THEN“ wurde jedoch der Variablentyp in eine Variable vom Typ DOUBLE umgewandelt, damit mit dem DOUBLE-Wert 0,1 verglichen werden kann. Der Wert des bedingten IF-Ausdruckes wird als FALSE (falsch) ausgegeben, und „NG;DOUBLE“ wird ausgedruckt.

```

FUNCTION Ua1_type
DIM a AS REAL
a = 0.1
IF a = 0.1 THEN
PRINT "OK;REAL"
ELSE
PRINT "NG;DOUBLE"
ENDIF
FEND

```

0,1 ist der reellen Variablen "a" zugewiesen

Die reelle Variable "a" wurde in eine Variable vom Typ DOUBLE umgewandelt, damit mit dem DOUBLE-Wert 0,1 verglichen werden kann.





## 3.5 Operatoren

### 3.5.1 Mathematische Operatoren

Die Mathematik wird für Operationen von numerischen Werten benutzt. Die mathematischen Operatoren sind wie nachfolgend aufgeführt:

Operator	Operation	Ausdruck	Mathematisches Zeichen
+	Addition	A+B	+
-	Subtraktion	A-B	-
*	Multiplikation	A*B	x
/	Division	A/B	÷
**	Exponential	A**B	A <sup>B</sup>
MOD	Module Mathematik	A MOD B	

(MOD kann nur für ganze Zahlen benutzt werden.)

### 3.5.2 Logische Operatoren

Die logischen Operatoren können für ganze Zahlen benutzt werden. Das Ergebnis ist auch eine ganze Zahl. Der numerische Wert wird innerhalb der Steuerung durch eine binäre Ziffer ausgedrückt. Die Operation wird in dieser Reihenfolge ausgeführt. Das heißt, dass jedes Bit des Ergebnisses von den entsprechenden Bits in den zwei Operanden bestimmt wird. Jede Operation ist unten aufgeführt, es werden die numerischen Werte A und B benutzt.

(1) AND (Logische Multiplikation)

Format: **A AND B**

BIT von A	BIT von B	Ergebnis
0	0	0
0	1	0
1	0	0
1	1	1

(2) OR (Logische Addition)

Format: **A OR B**

BIT von A	BIT von B	Ergebnis
0	0	0
0	1	1
1	0	1
1	1	1

(3) XOR (ausschließliches OR)

Format: **A XOR B**

BIT von A	BIT von B	Ergebnis
0	0	0
0	1	1
1	0	1
1	1	0

(4) NOT (logische Ergänzung)

Format: **NOT A**

BIT von A	Ergebnis
0	1
1	0

(5) Beispiel: Im Falle von A=1, B=0, C=1

A AND B OR C = 1

A AND B XOR C = 1

NOT A) AND C = 0

### 3.5.3 Relationale Operatoren

Operator	Ausdruck	Beschreibung
=	A=B	A und B sind gleich.
<> oder ><	A<>B	A und B sind nicht gleich.
>	A>B	A ist größer als B.
<	A<B	A ist kleiner als B.
>= oder =>	A>=B	A ist größer oder gleich B.
<= oder =<	A<=B	A ist kleiner oder gleich B.

### 3.5.4 Zeichenketten-Operatoren

Operator	Beschreibung
+	Verkettung der Zeichenkette
=	Wahr, wenn alle Eigenschaften der Zeichenkette dieselben sind.
<>	Wahr, wenn mindestens eine Eigenschaft unterschiedlich ist.

<Beispiel>

```
>PRINT "HE" + "LLO"
HELLO
```

### 3.5.5 Rangfolge der Operatoren

Es gibt eine Rangfolge für die Operatoren. Wird mehr als ein Operator benutzt, wird die Operation nach dieser Rangfolge ausgeführt. Operatoren, die in Klammern stehen, werden zuerst ausgeführt. Operationen mit gleichem Rang werden von links nach rechts ausgeführt.

Die Rangfolge der Operationen ist wie folgt:

- (1) ( )
- (2) \*\*
- (3) \*, /
- (4) MOD
- (5) +, -
- (6) =, >, <, >=, <=
- (7) AND, OR, XOR, NOT

Die Operationen werden nach dieser Rangfolge ausgeführt, das Benutzen von Klammern erleichtert das Lesen des Programms.

### 3.5.6 Operationen mit ganzen und reellen Zahlen

SPEL 95 unterscheidet Operationen mit ganzen Zahlen von denen mit reellen Zahlen, dadurch wird die Operation schneller. In Übereinstimmung mit dem Operandentyp unterscheidet sich die Operationsmethode und das Resultat wie folgt:

Operand	Ergebnis
Zwei ganze Zahlen	Ganze Zahl
Zwei reelle Zahlen	Reelle Zahl
Eine ganze und eine reelle Zahl	

Für die oben genannten Operationen beachten Sie folgende Punkte.

- (1) Der Bruch eines Quotienten, der durch zwei ganze Zahlen erreicht wird, wird verworfen.
- (2) Wenn das Ergebnis einer ganzzahligen Operation außerhalb des Bereichs der ganzen Zahlen (Überschreiten der 4 Byte-Grenze) liegt, wird ein Fehler angezeigt.
- (3) Wenn eine reelle Zahl einer Ganzzahlvariablen zugewiesen wird, wird der gebrochene Teil verworfen.

Beispiel	Operation mit ganzen Zahlen	Operation mit reellen Zahlen
(1)	<pre>&lt;PRINT 3 / 2 1 &gt;PRINT TAN(1 / 2) 0</pre>	<pre>&gt;PRINT 3 , 0 / 2 1 , 5 &gt;PRINT TAN(1 , 0 / 2) , 5463025</pre>
(2)	<pre>INTEGER A A=TAN(1 , 0 / 2) &gt;PRINT A 0</pre>	

## HINWEIS

Achten Sie auf folgendes, wenn Sie ein altes Programm importieren: In SPEL für die SRC-3XX (und frühere) Modelle wird eine reelle Zahl auf die nächstmögliche ganze Zahl gerundet, wenn sie einer INTEGER-Variablen zugewiesen wurde.

### 3.6 Betriebsausdrücke

Betriebsausdrücke können als Parameter für SPEL 95-Befehle und für die rechte Seite einer Substitutionsanweisung verwendet werden. Betriebselemente werden mit arithmetischen Operatoren verbunden. Es gibt die folgenden Betriebselemente (operationale Elemente):

- Variablen
- Konstanten
- Systemfunktionen (Rückgabewerte)
- Funktionen (Rückgabewerte)

Weitere Informationen zu den Rückgabewerten der Funktionen erhalten Sie in Kapitel 3.10.2 „Funktionsaufrufe“.

<Beispiel>

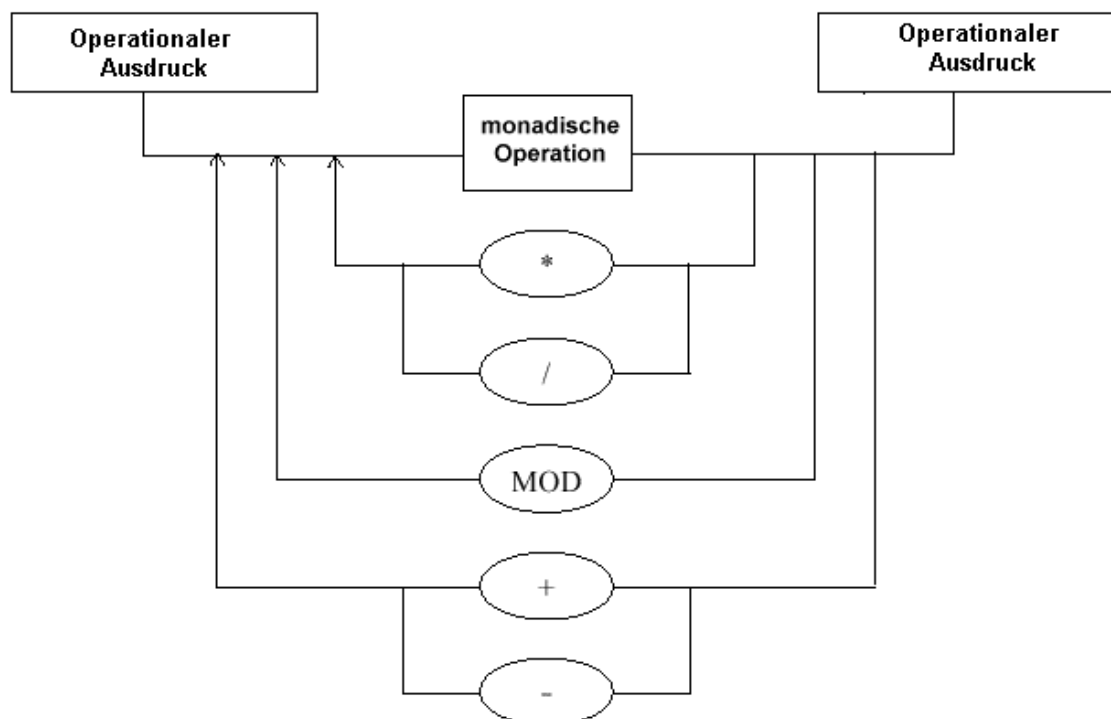
```
SPEED dat/2+10,Usrf(dat)
A=dat+prm*cos(dat)
```

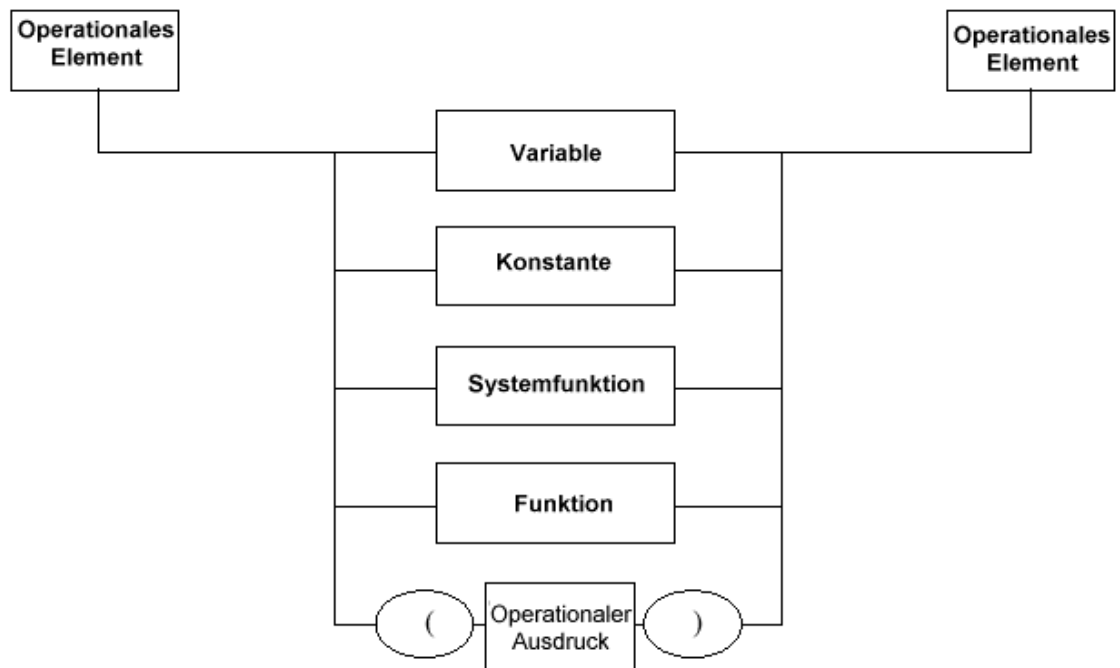
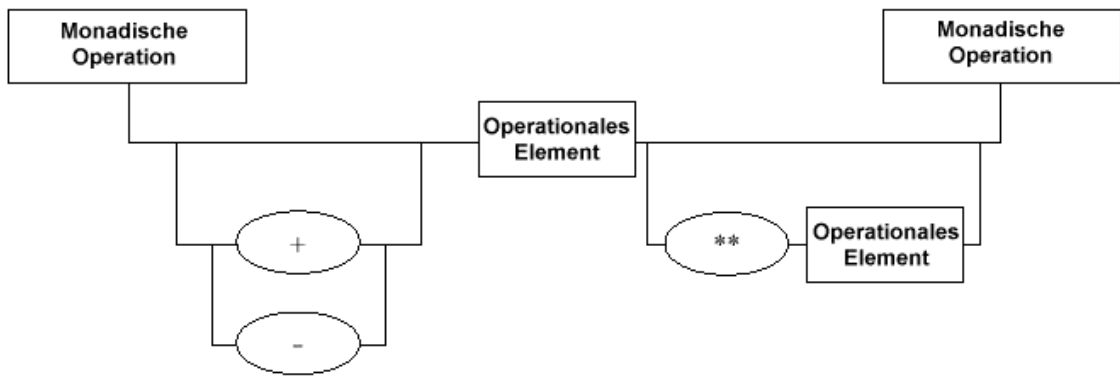
Man verwendet grafische Darstellungen, um die operationalen Ausdrücke zu beschreiben. Diagramme zeigen die Verbindung der operationalen Elemente und Operatoren von links nach rechts.

Rechteckige Kästchen werden anderswo in der Reihenfolge definiert, und ovale Rahmen enthalten SPEL 95-Schlüsselworte, -Zeichen und -Teile, die nicht getrennt werden können.

Pfeile treten in Bereichen auf, in denen Wiederholungen erfordern, dass sich entgegen der normalen Richtung bewegt wird.

Wie die Diagramme zeigen, können Betriebsausdrücke in Klammern eingeschlossen und als operationale Elemente geschrieben werden. Daraus resultierend können Teile eines operationalen Ausdrucks, die eingeklammert sind, früher als andere Teile ausgeführt werden.





### 3.7 Zeichenketten-Ausdrücke

Zeichenketten sind spezielle Anordnungen, die mehrere Zeichen wie die folgenden:

- mehrere Zeichen eingeschlossen durch Anführungszeichen („“)
- Zeichenketten-Variablen
- System-Zeichenketten-Funktionen

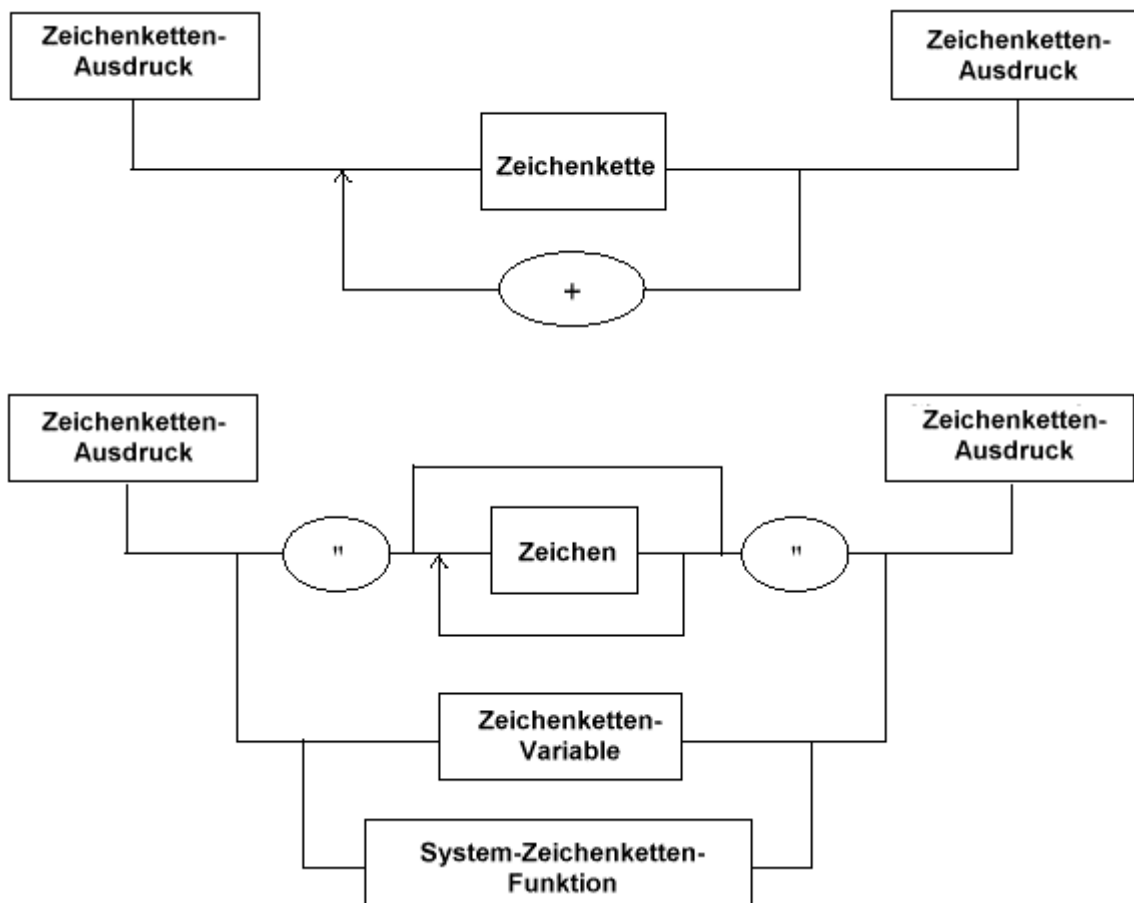
enthalten können.

Zeichenketten-Ausdrücke können als Parameter für SPEL 95 Befehle und für die rechte Seite einer Substitutionsanweisung für Zeichenketten-Variablen verwendet werden. Zeichenketten werden mit Zeichenketten-Operatoren verbunden.

<Beispiel>

```
PRINT A$+"Comment "  
B$=A$+CHR$( 20 )+"Comment "
```

Benutzen Sie die Grafik, um die Zeichenketten-Ausdrücke darzustellen:



### 3.8 Bedingte Ausdrücke

Bedingte Ausdrücke sind SPEL 95-Programmsteuerungs-Anweisungen, wie unten dargestellt. Die Task, die folgt, variiert in Abhängigkeit des Ergebnisses des Bedingungsausdrucks „true“ (wahr) oder „false“ (falsch).

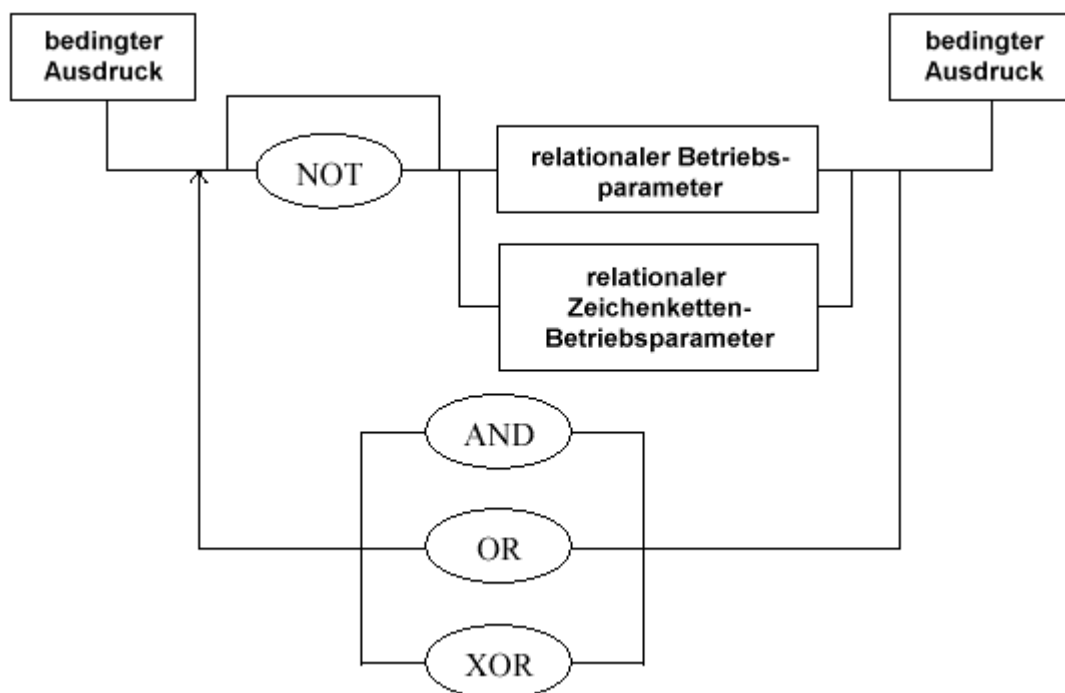
- IF THEN ELSE
- WHILE WHEN
- DO |WHILE| LOOP  
|UNTIL|
- DO LOOP |WHILE|  
|UNTIL|

Beim Verwenden eines Teils der logischen, relationalen und Zeichenketten-Operatoren (=, <>), verbinden sich bedingte Ausdrücke mit Zeichenketten-Ausdrücken.

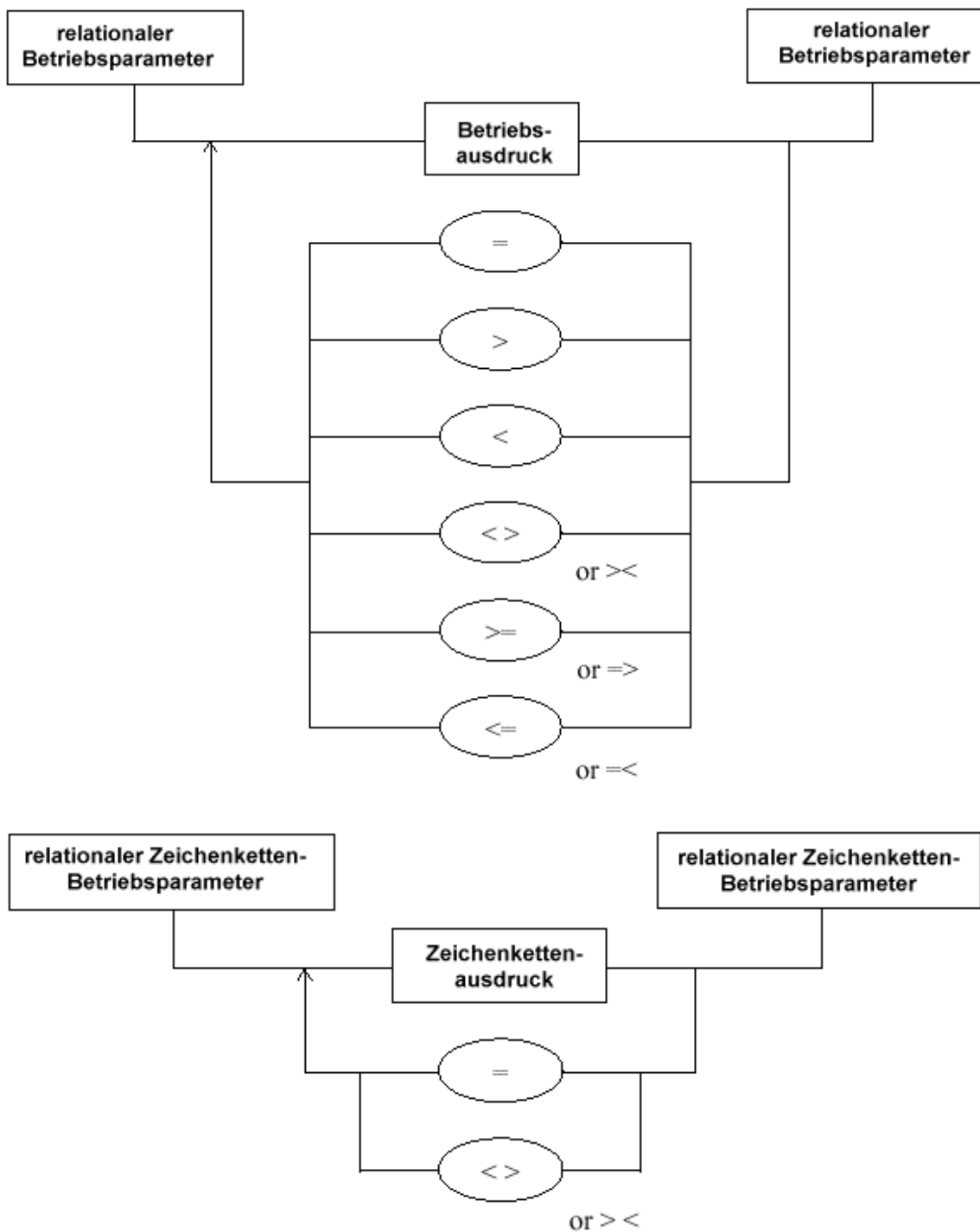
<Beispiel>

```
IF A>10 AND b<20 THEN
WHILE a$<>"GOOD"
```

Grafische Darstellungen werden verwendet, um die Zeichenketten-Ausdrücke darzustellen.







- **Genauere Bedeutung von „Ausdruck“**

Genau genommen kann SPEL 95 benutzt werden, um Befehlsparameter oder bedingte Ausdrücke auf der rechten Seite einer Substitutionsanweisung zu schreiben. Da bedingte Ausdrücke operationale Ausdrücke enthalten, ist es genauso möglich, Bitverarbeitungen an den arithmetischen Ergebnissen durchzuführen und durch Variablen zu ersetzen, als auch die Ergebnisse von relationalen Verarbeitungen (true/false) in Befehlsparameter zu wandeln.

### 3.9 Ereignisbedingte Ausdrücke

SPEL 95 beinhaltet Befehle, wie den WAIT-Befehl, der das Programm an der Ausführung hindert, bis bestimmte Bedingungen zutreffen oder ein bestimmtes Ereignis eintritt.

WAIT-Befehle beinhalten Zeit- oder ereignisbedingte Konditionen.

Bei den ereignisbedingten Konditionen werden mehrere ereignisbedingte Ausdrücke mit logischen Operatoren (AND/OR/XOR) verbunden. Ereignisbedingte Ausdrücke können weitgehend in zwei Klassen unterteilt werden: 1. Bit-ereignisbedingte Konditionen, die den Status von Bit-einheiten-Ereignissen aufdecken und 2. Daten-ereignisbedingte Konditionen, die Ereignisse durch Vergleich der Daten aufdecken. Bei beiden Typen beschreibt die linke Seite das aufzudeckende Ereignisobjekt, und die rechte Seite enthält eine Ziffer oder einen operationalen Ausdruck. Zusätzlich wird der Ausdruck auf der rechten Seite bewertet, bevor die Maschine in den Standby-Modus geht.

Das Format der Bit-ereignisbedingten Konditionen wird nachfolgend dargestellt:

```
[Bit-Ereignisobjekt]  {|=|[Ziffer oder operationaler Ausdruck]
                    |<>|
                    |><|
```

<Beispiel>

```
WAIT SW(5) = 1
WAIT$SW ( Bas + 1 ) <>res
```

Wenn das Bit-Ereignisobjekt den Status 1 annimmt, wird ein Ereignis aufgedeckt, wenn die linke Seite der Gleichung nicht 0 ist. Entsprechend wird ein Ereignis aufgedeckt, wenn der Status 0 annimmt, und die linke Seite der Gleichung gleich 0 ist. Die folgenden Objekte können allgemein als Bit-Ereignisse registriert werden:

SW([Nummer])	Änderung im festgelegten Eingangssignal wird als Ereignis registriert.
\$SW([Nummer])	Änderung im festgelegten E/A-Speicher wird als Ereignis registriert.
OPORT([Nummer])	Änderung im festgelegten Ausgangssignal wird als Ereignis registriert.
MOTOR([Nummer])	Änderung in der festgelegten Manipulatorenmotorspannung wird als Ereignis registriert.
MCALED([Nummer])	Änderung im Ergebnis der festgelegten Ursprungsrückgabe des Manipulators wird als Ereignis registriert.

Das Format der Daten-ereignisbedingten Konditionen wird nachfolgend dargestellt:

[Data-Ereignisobjekt]	{ = [Ziffer oder operationaler Ausdruck]
	<>
	<
	>
	<=
	>=
	=<
	=>

<Beispiel>

```
WAIT IN(5) = &B1111
WAIT$IN ( Bas + 1 ) <=res
```

Wenn sich das Daten-Ereignisobjekt ändert und das Ergebnis der verglichenen Daten - festgelegt durch die linke Seite - den Konditionen entspricht, wird ein Ereignis aufgedeckt. Die folgenden Objekte können allgemein als Daten-Ereignisse registriert werden:

IN([Nummer])	Änderung im Status des festgelegten Eingangs (8 Bit) wird als Ereignis registriert.
INW([Nummer])	Änderung im Status des festgelegten Eingangs (16 Bit) wird als Ereignis registriert.
\$IN([Nummer])	Änderung im Status des festgelegten E/A-Speichers (8 Bit) wird als Ereignis registriert.
\$INW([Nummer])	Änderung im Status des festgelegten E/A-Speichers (16 Bit) wird als Ereignis registriert.
OUT([Nummer])	Änderung im Status des festgelegten Ausgangs (8 Bit) wird als Ereignis registriert.
OUTW([Nummer])	Änderung im Status des festgelegten Zählers wird als Ereignis registriert.
LOF([Nummer])	Änderung im Status des festgelegten Kommunikationsspeichers wird als Ereignis registriert.

Bei Verwendung des Schlüsselwortes MASK können Sie die für das Datenereignisobjekt benötigten Bits als Objekt zur Aufdeckung des Ereignisses festlegen.

<Beispiel>

```
WAIT IN(5) MASK7 <>0
```

In diesem Beispiel wird ein Ereignis aufgedeckt, wenn eines der niedrigen drei Bits von IN(5) gleich 1 ist.

## 3.10 Funktionen

### 3.10.1 Funktionsstruktur

Eine Funktion ist die kleinste Einheit zum Ausführen eines SPEL 95-Programms (Erzeugen einer Task). Mehrere Funktionen können innerhalb einer Programmdatei existieren.

<Beispiel>

Programmdatei Test1

```

FUNCTION fun1
:
FEND
FUNCTION fun2
:
FEND

```

} Funktion fun1  
 } Funktion fun2  
 } 2 Funktionen können in einer Programmdatei Test1 geschrieben werden.

Programmdatei Test2

```

FUNCTION fun3
:
FEND

```

} Funktion fun3

Durch Registrieren einer Programmdatei als eine aktive Datei können Sie allen Funktionen, die im Programm enthalten sind, einen ausführbaren Status zuweisen.

Stellen Sie sicher, dass der Name der Funktion im Bereich [Function to execute] (ausführbare Funktion] des [Operating Window] angezeigt wird.

Markieren Sie den Namen der auszuführenden Funktion, und klicken Sie dann auf den <Start>-Schalter. Die Funktion wird ausgeführt, und eine Task wird erzeugt. Wenn die Task abgeschlossen ist, wird sie gecancelt.

Bis zu 32 Tasks können gleichzeitig ausgeführt werden.

### 3.10.2 Funktionsaufrufe

Eine Funktion kann als sogenannte Unteroutine von einer anderen Funktion, die augenblicklich ausgeführt wird, aufgerufen werden.

- |   |  |
|---|--|
| (1) Aufruf durch CALL:                  | Ruft eine andere Funktion von innerhalb der gleichen Task auf.   |
| (2) Aufruf durch XQT:                   | Erzeugt eine andere Funktion als separate Task.  |
| (3) Aufruf durch eine Anwenderfunktion: | Wird verwendet, um eine Funktion innerhalb der gleichen Task aufzurufen und um einen Rückgabewert zu erhalten. |

#### (1) Aufruf einer Funktion durch CALL

Wird eine Unterfunktion durch einen CALL-Befehl festgelegt, wird die Unterfunktion mittels der Hauptfunktion-Task ausgeführt. Wenn die Unteroutine ausgeführt ist, wird die Hauptfunktions-Task weiter ausgeführt.

Beschreibungen:

```
(Main function)
CALL subfunction {(Argument1...,Argument2,.....)}
:
(Seperate function)
FUNCTION subfunction {(Argument1...,Argument2,.....)}
```

<Beispiel1>

```
FUNCTION main
:
CALL sub           Unterfunktion wird ausgeführt
PRINT „ABC“       Nachdem die Unterfunktion ausgeführt ist, wird die Ausführung
:                 weitergeführt.
FEND

FUNCTION sub       Unterfunktion wird von der Haupttask ausgeführt.
:
FEND
```

Ein Argument kann von der Haupt- an die Unteroutine übergeben werden.

<Beispiel2>

```
FUNCTION main
:
X1=10
X2=20
CALL sub (X1,X2)  Die Unterfunktion wird ausgeführt. Zwei Argumente (X1, X2) werden an
:                 die Unterfunktion übergeben.
FEND

FUNCTION sub      Der Wert von X1 wird als arg1 akzeptiert. Der Wert von X2 wird als arg2
(arg1, arg2)      akzeptiert. Die Unteroutine wird ausgeführt.
PRINT arg1;arg2   Druckt die Argumente arg1 und arg2. In diesem Beispiel 10 und 20.
:
FEND
```

Bis zu 16 Level von CALL-Aufrufen (Verschachtelungen) können ausgeführt werden. (Verschachtelungen beinhalten hierbei auch die Anwenderfunktionen.)

Aufrufargumente können den direkten Gebrauch von Nummern einschließen.

Die Übergabe von Werten wird in der Reihenfolge der Argumente vollzogen. Daher ist die Reihenfolge, in der sie geschrieben sind, sehr wichtig. Ein Argument legt die Variablenart fest, und alle Konvertierungen der Argumente können durchgeführt werden. (Ist nichts angegeben, werden die Arten als reell betrachtet.)

Bis zu 32 Argumente können übertragen werden.

## (2) Aufruf einer Funktion durch XQT

Ein XQT-Befehl wird verwendet, um eine Funktion als einzelne Task aufzurufen. Wenn der XQT-Befehl innerhalb der Hauptfunktion geschrieben ist, wird die Unterfunktion ausgeführt, und Tasks werden erzeugt. Weitere Informationen erhalten Sie in Kapitel 8 „Multitasking“.

Beschreibungen:

(Main function)

```
XQT subfunction {(Argument1...,Argument2,.....)}
```

:

(Seperate function)

```
FUNCTION subfunction {(Argument1...,Argument2,.....)}
```

<Beispiel1>

```
FUNCTION main
XQT sub
PRINT „ABC“
:
FEND
```

Unterfunktion wird erzeugt.

Wird ausgeführt mit nachfolgendem XQT-Befehl.

```
FUNCTION sub
:
FEND
```

Unterfunktion wird ausgeführt. Nachdem der XQT-Befehl ausgeführt ist, wird eine neue Task erzeugt, die mit FEND endet.

<Beispiel2>

```
FUNCTION main
:
X1=10
X2=20
XQT sub (X1,X2)
:
FEND
```

Die Unterfunktion wird ausgeführt. Zwei Argumente (X1, X2) werden an die Unterfunktion übergeben.

```
FUNCTION sub
(arg1,arg2)
PRINT arg1;arg2
:
FEND
```

Der Wert von X1 wird als arg1 akzeptiert. Der Wert von X2 wird als arg2 akzeptiert. Die Unterroutine wird ausgeführt. Druckt die Argumente arg1 und arg2. In diesem Beispiel 10,000000 und 20,000000.

Die Übergabe von Werten wird in der Reihenfolge der Argumente vollzogen. Daher ist die Reihenfolge, in der sie geschrieben sind, sehr wichtig.

Ein Argument legt die Variablenart fest, und alle Konvertierungen der Argumente können durchgeführt werden. (Ist nichts angegeben, werden die Arten als reell betrachtet.)

Bis zu 32 Argumente können übertragen werden.

### (3) Aufruf einer Anwenderfunktion

Durch Verwendung des Unterfunktionsnamens als Variable erhalten Sie den Rückgabewert von der Unterfunktion.

Beschreibungen:

[Variablenname] = Subfunktion (Argument1 ..., Argument2 .....,)

:

(Subfunktion)

Function = Subfunktion (Argument1 ..., Argument2 .....,)

Variable mit gleichem Namen, wie die Unterfunktion = (Ausdruck)

<Beispiel1>

```
FUNCTION main
```

```
  :
```

```
  X1=10
```

```
  X2=20
```

```
  a=sub(X1, X2)
```

```
  PRINT a
```

```
  :
```

```
  FEND
```

```
FUNCTION sub (arg1, arg2)
```

```
  Das Ergebnis sub=arg1*arg2
```

```
  der Variable :
```

```
  sub wird :
```

```
  ersetzt für
```

```
  Variable a. FEND
```

Zwei Argumente (X1, X2) werden an die Unterfunktion übergeben. Die Variable a wird durch den Rückgabewert ersetzt.

Zeigt Variable a an, die durch den Rückgabewert ersetzt wurde. Es wird also 200 angezeigt.

Der Wert von X1 wird als arg1 akzeptiert. Der Wert von X2 wird als arg2 akzeptiert. Die Unteroutine wird ausgeführt. Die Variable sub wird durch das Ergebnis von arg1\*arg2 ersetzt.

Die Namen der Unterfunktionsvariablen sind die gleichen, wie die Namen der Unterfunktion. Ein Argument legt den Typ der Variablen fest, und Konvertierungen können durchgeführt werden. (Ist nichts festgelegt, wird der Typ als reell angenommen.)

Bis zu 32 Argumente können übertragen werden.

### 3.10.3 Festlegen des Argumenttyps

Sie können die Art des Arguments und den Rückgabewert selber festlegen. Ein automatisches Festlegen des Typs konvertiert den Typ des aus der höheren Funktion übergebenen Wertes. Ist der Typ nicht festgelegt, wird er automatisch als reeller Wert angenommen. Weitere Informationen erhalten Sie in Kapitel 3.4.3 „Numerische Variablen“.

Beschreibungen:

```
FUNCTION subfunction (Argument1 AS Variablenart, Argument2 AS Variablenart,...)
  AS Art des Rückgabewertes
```

<Beispiel>

```
FUNCTION main
  :
  X1=10
  X2=20
  a=sub(X1,X2)
  PRINT a
  :
  FEND
```

```
Variable sub wird zum Typ INTEGER
FUNCTION sub (arg1 AS INTEGER, arg2
  AS DOUBLE) AS INTEGER
  sub=arg1*arg2
  :
  FEND
```

Legt arg1 als INTEGER und arg2 als DOUBLE fest. Der Rückgabewert ist vom Typ INTEGER.



### 3.10.4 Festlegen des Referenz-Arguments

Durch Festlegen eines Referenzarguments vor einem Argument können Sie den Wert des Unterfunktions-Arguments in eine Hauptfunktions-Variable projizieren.

Beschreibungen:

(Main function)

CALL Subfunction {(BYREF Argument1 ..., BYREF Argument2 .....,...)}  
 :

(Seperate function)

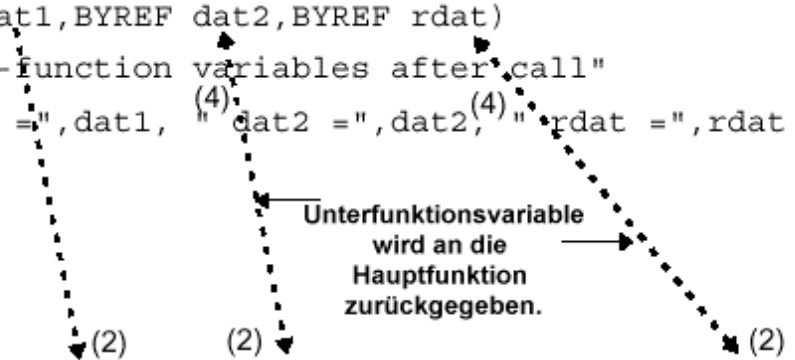
FUNCTION Subfunction {(BYREF Argument1 ..., BYREF Argument2 .....,...)}  
 :

<Beispiel1>

```

FUNCTION main
DIM dat1,dat2 AS INTEGER
DIM rdat AS REAL
CLS
dat1 = 123; dat2 = 456; rdat = 123.456789
PRINT "Main-function variables before call"
PRINT "dat1 =",dat1, " dat2 =",dat2, " rdat =",rdat
(1) CALL subf(dat1,BYREF dat2,BYREF rdat)
PRINT "Main-function variables after call"
PRINT "dat1 =",dat1, " dat2 =",dat2, " rdat =",rdat
FEND

```



```

(1)FUNCTION subf(arg1,BYREF arg2 AS INTEGER,BYREF arg3)
PRINT "Variables within function"
PRINT "arg1 =",arg1, " arg2 =",arg2, " arg3 =",arg3
(3) prml = 321; arg2 = 654; arg3 = 987.654321
PRINT "Variables within function after change"
PRINT "arg1 =",arg1, " arg2 =",arg2, " arg3 =",arg3
FEND

```

Wird das Programm ausgeführt, erscheint es wie unten dargestellt:

```

Main-function variables before call      (Hauptfunktionsvariablen vor Aufruf)
dat1=123 dat2=456 rdat=123,456789
Variables within function              (Variablen innerhalb der Funktion)
arg1=123 arg2=456 arg3=123,456789
Variables within function after change  (Variablen innerhalb der Funktion
nach Änderung)

arg1=321 arg2=654 arg3=987,654321
Main-function variables after call      (Hauptfunktionsvariablen nach Aufruf)
Dat1=123 dat2=654 rdat=987,654321
Die Werte der Variablen dat2 und rdat wurden innerhalb der Unterfunktion verändert, aber der
Wert von dat1 wurde davon nicht betroffen.

```

Bemerkung:

- (1) BYREF wird in den Hauptfunktionsvariablen dat2 und rdat festgelegt. BYREF wird auch in den Unterfunktionsvariablen arg2 und arg3 festgelegt.
- (2) Die Werte der Variablen dat1, dat2 und rdat werden an prm1, prm2 und prm3 der Unterfunktion subf übergeben.
- (3) Die Werte von arg1, arg2 und arg3 werden geändert.
- (4) Rückkehrend zur Hauptfunktion werden die Werte von arg2 und arg3 an dat2 und rdat übertragen. Da BYREF nicht für dat1 oder arg1 festgelegt wurde, werden die Werte hierfür nicht projiziert.

## HINWEIS

Vorsichtsmaßnahmen beim Gebrauch von Referenzargumenten:  
 Wenn Sie ein Referenzargument festlegen, müssen Sie die Referenz BYREF auf beiden Seiten (Aufrufseite und Funktionsseite) festlegen. Wurde nur eine Seite festgelegt, wird während der Ausführung ein Fehler auftreten. Globale und Backup-Variablen können auch als Referenzargumente übergeben werden.

### 3.10.5 Systemfunktionen

Systemfunktionen werden standardmäßig von SPEL 95 unterstützt. Sie können als Befehle von SPEL95-Parametern oder als rechte Seite von Substitutionsanweisungen eingesetzt werden.

Beispiel, in dem ein Argument vorhanden ist:

```
A=SIN(45*PI/180)
```

Beispiel, in dem kein Argument vorhanden ist:

```
A=MCALED
IF MCALED=1 THEN
```

### 3.11 Programmsteuerungsanweisungen

Dieser Abschnitt erläutert die Programmsteuerungsanweisungen und die entsprechende Grammatik. Beispiele werden dargestellt.

FOR...NEXT	(Führt eine Reihe von Anweisungen in einer bestimmten Zeit aus.)
GOTO	(Geht zur gewünschten Anweisung über/ unbedingte Verzweigung.)
IF...THEN...ELSE	(Führt Anweisungen auf der Grundlage von spezifizierten Konditionen aus/ bedingte Verzweigung.)
GOSUB...RETURN	(Verzweigt zum Unterprogramm, führt es aus und verlässt es wieder.)
SELECT...CASE...END    SELECT	(Spezifiziert die Verzweigungsvorschrift und die entsprechende Verzweigungsanweisungssequenz.)
WHILE...WEND	(Führt eine Reihe von Anweisungen aus, während der festgelegte Zustand erhalten bleibt/ bedingte Wiederholung.)
DO{WHILE UNTIL}...LOOP	(vorbedingte Überprüfung/ Wiederholung)
DO...LOOP {WHILE UNTIL}	(nachbedingte Überprüfung/ Wiederholung)
TRAP	(Definiert einen Unterbrechungsprozess.)

#### FOR...NEXT

<b>Funktion</b>	Führt eine Reihe von Anweisungen mit der angegebenen Anzahl von Wiederholungen aus.	
<b>Format</b>	FOR [Variablenname]=[Anfangswert] TO [Endwert] {STEP [Steigerungswert]} : NEXT {[Variablenname]}	
<b>Beschreibung</b>	Für den Anfangs-, End- und Steigerungswert sind Eingaben von $\pm 0,000001$ bis $\pm 9999999$ erlaubt. Es sind maximal 16 Verschachtelungen möglich. Führt eine Reihe von Anweisungen mit der angegebenen Anzahl von Wiederholungen aus. Die Anzahl der wiederholten Schleifen (Angabe, wie oft die Anweisungen zwischen FOR und NEXT wiederholt werden) wird durch die Angaben für den Anfangs-, End- und Steigerungswert bestimmt.	
<b>Beispiel</b>	<pre>FOR SP=10 TO 100 STEP 10   SPEED SP   FOR I=1 TO 5     JUMP P(I)   NEXT I NEXT SP</pre>	Ändert die Geschwindigkeit von 10 auf 100 in 10er-Intervallen, springt bei jedem Geschwindigkeitswechsel von P1 nach P5.

**GOTO**

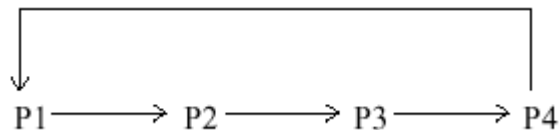
<b>Funktion</b>	Verzweigt zur angegebenen Zeilennummer.	
<b>Format</b>	GOTO  Zeilennummer   Sprungmarke	
<b>Beschreibung</b>	Verzweigt ohne Bedingung zur angegebenen Zeile bzw. zur angegebenen Sprungmarke. Existiert die angegebene Zeile bzw. die Sprungmarke nicht, erfolgt ein Fehler.	
<b>Beispiel</b>	<pre>GOTO TIMER : TIMER</pre>	Verzweigt zur mit TIMER gekennzeichneten Zeile, setzt die Ausführung fort.
	<pre>FUNCTION main 100 PRINT „Das ist mein erstes Programm.“       JUMP P0       JUMP P1  GOTO 100 FEND</pre>	Verzweigt zu Zeile 100, setzt die Ausführung fort.

**Übung**

- ★ FOR...NEXT
- ★ GOTO

**Tätigkeit**

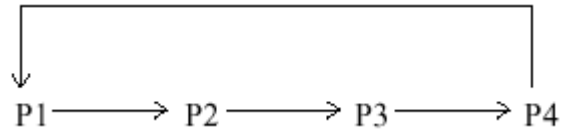
Laufend fortbewegte Aktion von P1 nach P4, und deren Wiederholung.

**Programm:**

```
FUNCTION MAIN
LOOP1:
  FOR I=1 TO 4
    JUMP P(I)
  NEXT I
GOTO Loop1
FEND
```

<b>Übung</b>	★ FOR...NEXT ★ GOTO
--------------	------------------------

**Tätigkeit** Laufend fortbewegte Aktion von P1 nach P4 und deren Wiederholung. Zusätzlich wird bei jeder Schleife die Geschwindigkeit (Speed) erhöht. Der ganze Vorgang endet nach 10 Schleifen.

**Programm:**

```
FUNCTION MAIN
FOR I=10 TO 100 STEP 10
  SPEED I
  FOR J=1 TO 4
    JUMP P(J)
  NEXT J
NEXT I
FEND
```

**IF...THEN...ELSE**

**Funktion** Ausführung einer Anweisung mit Bedingung.

**Format**

(1) IF [Bedingung] THEN

⋮

{ELSE}

⋮

ENDIF

(2) IF [Bedingung]

THEN { |Zeilennummer| } ELSE { |Zeilennummer| }  
       { |Sprungmarke| }        { |Sprungmarke| }

Es sind bis zu 16 Verschachtelungen erlaubt. (Einschließlich SELECT CASE-Anweisungen)

**Beschreibung**

- (1) Definiert eine IF-Bedingung. Ist die IF-Bedingung erfüllt, werden die Anweisungen zwischen IF und ELSE ausgeführt, bei Nichterfüllen der IF-Bedingung werden die Anweisungen zwischen ELSE und ENDIF ausgeführt. Die Zeilen von ELSE bis ENDIF können weggelassen werden. In diesem Fall wird die Steuerung bei Nichterfüllen der IF-Bedingung an die Zeile übergeben, die unmittelbar auf die Zeile ENDIF folgt.
- (2) Wird die IF-Bedingung der angegebenen Zeile erfüllt, werden die Anweisungen zwischen THEN und ELSE ausgeführt. Bei Nichterfüllen der Bedingung werden alle auf ELSE folgenden Anweisungen ausgeführt. Die Eingaben ab ELSE sind optional und können weggelassen werden. In diesem Fall wird die Steuerung bei Nichterfüllen der IF-Bedingung an die nächste Zeile übergeben.

Übung	★ IF...THEN...ELSE
<b>Tätigkeit</b>	Wenn Eingang 1 auf ON steht, werden Tätigkeiten zwischen P1 und P2 ausgeführt. Steht er auf OFF, blitzen die Ausgänge 1 bis 15 in 0,5-Sekunden-Intervallen auf.
<b>Programm:</b>	<pre>FUNCTION MAIN Loop1 :   IF SW(1)=1 THEN     JUMP P1; JUMP P2   ELSE     FOR I=1 TO 15       ON I;SLEEP 0,5; OFF I; SLEEP 0,5     NEXT I   ENDIF GOTO Loop1 FEND</pre>

Übung	★ IF...THEN...ELSE
<b>Tätigkeit</b>	Während der Statusprüfung von Eingangsbit 0, bewegt sich die Z-Achse an Punkt P1 hoch und runter. Erfolgt eine Eingabe, wird zwischen P2, P3 und P4 gesprungen.
<b>Programm:</b>	<pre>FUNCTION MAIN Loop1 : JUMP P1 IF SW(0)0= THEN Loop1 ELSE 10 10 FOR I=2 TO 4   JUMP P(I) NEXT I GOTO Loop1 FEND</pre>

**GOSUB...RETURN**

**Funktion** Verzweigt in ein Unterprogramm, führt es aus und verlässt es danach wieder.

**Format** GOSUB |Zeilennummer|  
|Sprungmarke|

```

:
|Zeilennummer|
|Sprungmarke|
:
RETURN

```

Es sind bis zu 16 Verschachtelungen erlaubt.

**Beschreibung** Verzweigt zur angegebenen Zeilennummer bzw. zur angegebenen Sprungmarke, führt das danach folgende Unterprogramm aus und kehrt anschließend zum Hauptprogramm zurück. Achten Sie darauf, dass jedes Unterprogramm mit RETURN beendet wird. Dadurch kehrt die Programmausführung zu der Stelle im Programm zurück, die auf GOSUB folgt.

**Übung** ★ GOSUB

**Tätigkeit** Wiederholt die Zyklen P1-P2, P2-P3, etc. dreimal in Folge. Macht eine Unterfunktion der Zwei-Punkt-Bewegung.

**Programm:**

```

FUNCTION MAIN
    I=1; J=2
    GOSUB I_J
    I=2, J=3
    GOSUB I_J
    I=3, J=4
    GOSUB I_J
    I=4, J=1
    GOSUB I_J
END

```

```

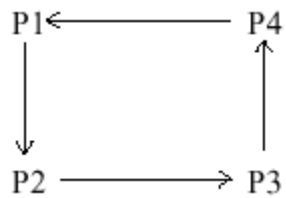
I_J:                                     Sprungmarke Unterroutine
    FOR LOOP1=1 TO 3
        JUMP P(I);JUMP P(J)
    NEXT
    RETURN
FEND

```



<b>Übung</b>	<ul style="list-style-type: none"> <li>★ GOTO</li> <li>★ FOR...NEXT</li> <li>★ GOSUB...RETURN</li> <li>★ IF...THEN...ELSE</li> </ul>
--------------	--

**Tätigkeit** Erhöht die Geschwindigkeit jedesmal, wenn eine Bewegung zum nächsten Punkt ausgeführt wird. Nach der Ausführung von SPEED100 wird zum SPEED10 zurückgekehrt. An jedem Punkt bewegt sich die Z-Achse dreimal hoch und runter.



**Programm:**

```

FUNCTION MAIN
  SPEED 10;A=10
10 FOR I=1 TO 4
  JUMP P(I)
  GOSUB H_L
  A=A+10
  IF A>100 THEN
    A=10
  ENDIF
  SPEED A
NEXT I
GOTO 10
'
H_L:
  FOR J=1 TO 3
    JUMP P(I)
  NEXT J
RETURN
'FEND
  
```

**SELECT CASE...CASE...END SELECT**

<b>Funktion</b>	Bewegt die Steuerung zu einem der verschiedenen Sätze entsprechend dem Wert des Ausdrucks.
<b>Format</b>	<pre> SELECT CASE  [Ausdruck] [Zeichenketten-Ausdruck]       CASE [Feld];[Anweisung]       :       CASE [Feld];[Anweisung]       CASE {ELSE};[Anweisung]}       END SELECT </pre>
<b>Beschreibung</b>	<p>CASE: Bis zu 32.</p> <p>Es sind bis zu 16 Verschachtelungen erlaubt (einschließlich IF-Anweisungen).</p> <p>Entspricht eines der CASE-Felder dem Ergebnis des SELECT-Ausdrucks, wird die entsprechende CASE-Anweisung ausgeführt. Danach wird die Programmsteuerung mit dem Befehl fortgesetzt, der dem Befehl END SELECT folgt. Entspricht keines der CASE-Felder dem Ergebnis des SELECT-Ausdrucks, wird der Befehl, der der CASE ELSE-Anweisung folgt, ausgeführt, und die Programmsteuerung wird mit dem Befehl fortgesetzt, der dem Befehl END SELECT folgt. Existiert keine CASE ELSE-Anweisung, geht das Programm zu dem Befehl, der der END SELECT-Anweisung folgt, ohne irgend etwas auszuführen.</p>
<b>Beispiel</b>	<pre> FUNCTION main DIM I AS INTEGER FOR I=0 TO 10       SELECT CASE I       CASE 0; OFF 1; ON 2; JUMP P1       CASE 3; ON 1; OFF 2       JUMP P2; MOVE P3; ON 3       CASE 7; ON 4       CASE ELSE; ON 7       END SELECT NEXT I FEND </pre>

**WHILE...WEND**

---

**Funktion** Führt die angegebenen Anweisungen aus, während die festgelegte Bedingung erfüllt ist.

**Format** WHILE [Bedingung]

Es sind bis zu 16 Verschachtelungen erlaubt (einschließlich anderer Wiederholungsanweisungen).

**Beschreibung** Definiert die WHILE-Bedingung. Wenn diese erfüllt wird, werden die Anweisungen zwischen WHILE und WEND ausgeführt, und anschließend wird die WHILE-Bedingung erneut geprüft. Dieser Vorgang (Ausführung der WHILE...WEND-Anweisungen, Prüfen der WHILE-Bedingung) wird solange wiederholt, wie die WHILE-Bedingung erfüllt ist. Wird die WHILE-Bedingung nicht erfüllt, wird die Programmsteuerung mit dem auf WEND folgenden Befehl fortgesetzt. Wird die WHILE-Bedingung bei der ersten Abfrage nicht erfüllt, werden die Anweisungen zwischen WHILE und WEND nie ausgeführt.

**Beispiel** I=1

```
    WHILE I<60
        :
        I=I+2
    WEND
```

**DO { |WHILE|UNTIL| }...LOOP**

<b>Funktion</b>	Führt die Anweisungen in den Zeilen zwischen DO und LOOP aus, solange oder bis die Bedingungen zutreffen.
<b>Format</b>	DO [{  WHILE  } Bedingung]  UNTIL  : LOOP WHILE: Während die Bedingungen zutreffen. UNTIL: Bis die Bedingungen zutreffen. Es sind bis zu 16 Verschachtelungen erlaubt (einschließlich anderer
<b>Beschreibung</b>	Wenn WHILE verwendet wird, wird die Bedingung überprüft. Ist das Ergebnis zutreffend, werden die Zeilen zwischen DO und LOOP ausgeführt. Nach der Ausführung wird die Bedingung wieder überprüft. Solange, wie die Bedingungen zutreffen, wird der Vorgang wiederholt ausgeführt. Ist die Bedingung nicht zutreffend, wird die auf die LOOP-Anweisung folgende Zeile ausgeführt. Wird die Bedingung von Anfang an nicht erfüllt, werden die Anweisungen zwischen DO und LOOP nie ausgeführt. Wenn UNTIL verwendet wird, wird die Bedingung überprüft. Ist das Ergebnis nicht zutreffend, werden die Zeilen zwischen DO und LOOP ausgeführt. Nach der Ausführung wird die Bedingung wieder überprüft. Solange, wie die Bedingungen nicht zutreffen, wird der Vorgang wiederholt ausgeführt. Ist die Bedingung zutreffend, wird die auf die LOOP-Anweisung folgende Zeile ausgeführt. Wird die Bedingung von Anfang an erfüllt, werden die Anweisungen zwischen DO und LOOP nie ausgeführt.
<b>Beispiel</b>	DO WHILE SW(0)=1 : LOOP  DO UNTIL SW(0)=0 : LOOP  Das Ergebnis ist in beiden Beispielen dasselbe.

**DO ...LOOP { [WHILE|UNTIL] }**

<b>Funktion</b>	Führt die Anweisungen in den Zeilen zwischen DO und LOOP aus, solange oder bis die Bedingungen zutreffen.
<b>Format</b>	<pre>DO : LOOP[{  [WHILE  ] Bedingung]         [UNTIL]</pre> <p>WHILE: Während die Bedingungen zutreffen.  UNTIL: Bis die Bedingungen zutreffen.  Es sind bis zu 16 Verschachtelungen erlaubt (einschließlich anderer Wiederholungsanweisungen).</p>
<b>Beschreibung</b>	<p>Wenn WHILE verwendet wird, werden die Zeilen zwischen DO und LOOP ausgeführt, und die Bedingung wird überprüft. Ist das Ergebnis zutreffend, werden die Zeilen zwischen DO und LOOP wiederholt ausgeführt. Solange, wie die Bedingungen zutreffen, wird der Vorgang wiederholt ausgeführt. Ist die Bedingung nicht zutreffend, wird die auf die LOOP-Anweisung folgende Zeile ausgeführt. Wird die Bedingung von Anfang an nicht erfüllt, werden die Anweisungen zwischen DO und LOOP trotzdem einmal ausgeführt.</p> <p>Wenn UNTIL verwendet wird, werden die Zeilen zwischen DO und LOOP ausgeführt, und die Bedingung wird überprüft. Ist das Ergebnis nicht zutreffend, werden die Zeilen zwischen DO und LOOP wiederholt ausgeführt. Solange, wie die Bedingungen nicht zutreffen, wird der Vorgang wiederholt ausgeführt. Ist die Bedingung zutreffend, wird die auf die LOOP-Anweisung folgende Zeile ausgeführt. Wird die Bedingung von Anfang an erfüllt, werden die Anweisungen zwischen DO und LOOP trotzdem einmal ausgeführt.</p>
<b>Beispiel</b>	<pre>DO : LOOP WHILE SW(0)=1  DO : LOOP UNTIL SW(0)=0</pre> <p>Das Ergebnis ist in beiden Beispielen dasselbe.</p>

## 3.12 Interrupt (Unterbrechung)

### TRAP

<b>Funktion</b>	Definiert einen Unterbrechungsprozess.
<b>Format</b>	
(1)	<pre>TRAP[Trap-Nummer]  {[Eingangsbedingung] (ereignisbedingter Ausdruck)                      GOTO   [Sprungmarke]        }                      [Zeilennummer]        }                      GOSUB  [Sprungmarke]                              [Zeilennummer]        }                      CALL  [Funktionsname]  </pre> <p>Trap-Nummer: ganze Zahl zwischen 1 und 4.</p>
(2)	<pre>TRAP   EMERGENCY {CALL[Funktionsname]}        ERROR               PAUSE               SGOPEN      </pre>
<b>Beschreibung</b>	Wir geben hier nur einen Überblick. Detaillierte Informationen erhalten Sie in der Online-Hilfe:
(1)	Führt den durch einen der Befehle GOTO, GOSUB bzw. CALL definierten Unterbrechungsvorgang aus, wenn die Eingangsbedingung erfüllt ist.
(2)	Ist eine der Eingangsbedingungen ein NOT-AUS, ein Fehler, der Befehl PAUSE oder ist die Sicherheitsabschränkung geöffnet, wird eine innerhalb des TRAP-Vorgangs mit CALL definierte Funktion vorrangig ausgeführt.
<b>Beispiel</b>	<p>Ausführen eines durch den Anwender definierten Fehlers.</p> <pre>FUNCTION main TRAP 1 SW(0)=1 CALL AFTERSW  `Trap ist eingerichtet : FUNCTION AFTERSW ON 31                                `Lampen gehen an PRINT #20, „Fehler aufgetreten“ FEND</pre>

## Kapitel 4

# Projektverwaltung

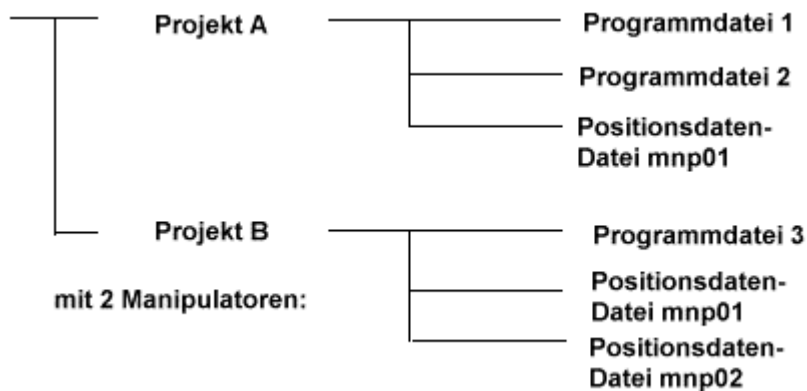
### 4.1 Projekte

Die Anwendungen, die mit SPEL 95 erstellt werden, werden als Projekte verwaltet. Ein Projekt beinhaltet zumindest eine Programmdatei, eine Positionsdaten-Datei und anwendungsbezogene Einstellungen (Konfigurationsdatei).

Sie können sich ein Projekt als Container vorstellen, in dem alle Informationen zum Ausführen der Anwendung zum Betrieb des Roboters gesammelt werden.

Zum Beispiel, wenn mehrere Programme oder Positionsdaten-Dateien zum Ausführen einer Anwendung benötigt werden, können die benötigten Dateien in einem Projekt angeordnet, registriert und verwaltet werden. Dadurch muss der Anwender mehrere Dateien nicht selbst verwalten.

Durch das Erzeugen mehrerer Projekte können Sie ein Generationsmanagement ausführen, gleiche Anwendungen veranlassen, verschiedene Ziele anzustreben und diese als Projekte verwalten.



Auch wenn es sich um ein einfaches Programm handelt, erzeugen Sie bitte ein Projekt und schreiben die Befehle in eine Programmdatei. Lassen Sie dann das Programm als Projekteinheit ablaufen.

Durch Ausführen mehrerer Programme innerhalb eines Projektes können Sie mehrere Tasks generieren. Ferner können Sie individuelle Tasks anhalten.

Sie können jedoch nicht mehrere Projekte zur selben Zeit öffnen.

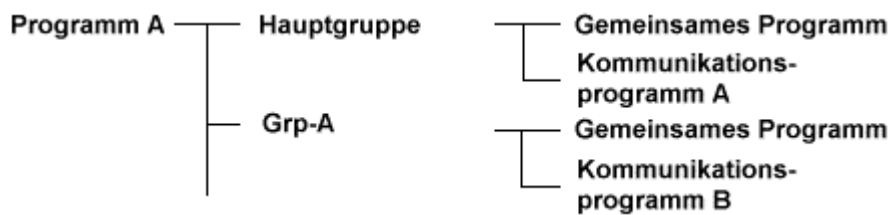
### 4.1.1 Gruppen

Bei einer Gruppe handelt es sich um ein Konzept zum Verwalten von Programmdateien. Sie können sich eine Gruppe als Container vorstellen, in dem Programme für einzelne Anwendungen gesammelt werden. Eine Programmdatei wird ausführbar, wenn Sie in einer Gruppe registriert wurde.

Eine Anwendung kann zum Ausführen einer speziellen Funktion erzeugt werden, indem Sie in einem gemeinsamen Programm innerhalb verschiedener Gruppen oder als besondere Positionsdaten-Datei registriert wurde.

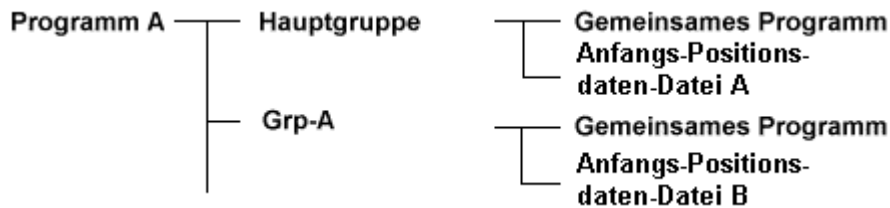
<Beispiel A>

Eine Programmdatei kann in mehr als einer Gruppe gespeichert werden. Durch das Erzeugen eines unterschiedlichen Kommunikationsprogramms und das Speichern in einer einzelnen Gruppe mit einem gemeinsamen Programm kann eine Anwendung mit unterschiedlichem Kommunikationsverfahren entwickelt werden.



<Beispiel B>

Um eine andere Anwendung mit einem gemeinsamen Programm mit unterschiedlicher Positionsdaten-Einstellung zu erzeugen, müssen Sie eine andere Anfangs-Positionsdaten-Datei innerhalb der Gruppe registrieren.

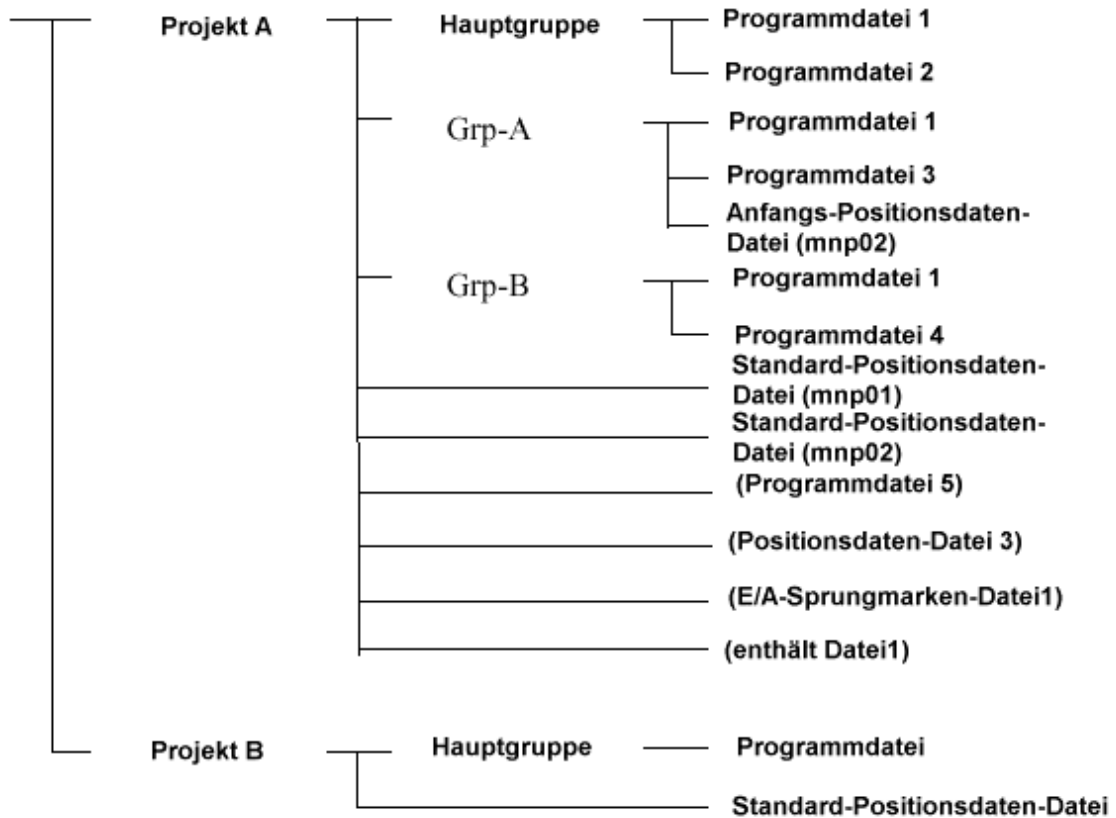




### 4.1.2 Beziehung zwischen Projekten und Gruppen

Für jedes Projekt können Sie wenigstens eine Gruppe einrichten. Jede Gruppe enthält mindestens eine Programmdatei.

Das nachfolgende Baummodell veranschaulicht eine Projektstruktur:



Das Projekt A steuert 2 Manipulatoren: mnp01 und mnp02.

Grp-A benutzt die Standard-Positionsdaten-Datei für mnp01 und die Anfangs-Positionsdaten-Datei für mnp02.

( ) weist auf Dateien hin, die.-falls notwendig - hinzugefügt werden können.

Das Projekt B ist typisch strukturiert mit der minimalsten Anzahl an Dateien.

### HINWEIS

Wenn Sie planen, mehrere Dateien zu verwenden, um einen einzelnen Manipulator zu betreiben, stellen Sie bitte sicher, dass das Programm über einen Schließmechanismus verfügt.

Wenn verschiedene Betriebsbefehle an den Manipulator gesendet werden, während er betrieben wird, kann es eine Fehlfunktion beim Manipulator geben, und es kann passieren, dass er sich in unerwarteter Weise bewegt. Das kann gefährlich sein!

### 4.1.3 Standard-Positionsdaten-Datei

Ein Projekt benötigt eine Standard-Positionsdaten-Datei für jeden Manipulator. Wenn ein neues Projekt erzeugt wird, wird automatisch eine Standard-Positionsdaten-Datei registriert. Es wird auch dann eine neue Standard-Positionsdaten-Datei registriert, wenn ein Manipulator neu registriert wird.

Die Standard-Positionsdaten-Datei ist die Datei, die automatisch in den Speicher der Steuerung geladen wird, wenn keine Anfangs-Positionsdaten-Datei registriert ist.

#### 4.1.4 Anfangs-Positionsdaten-Datei

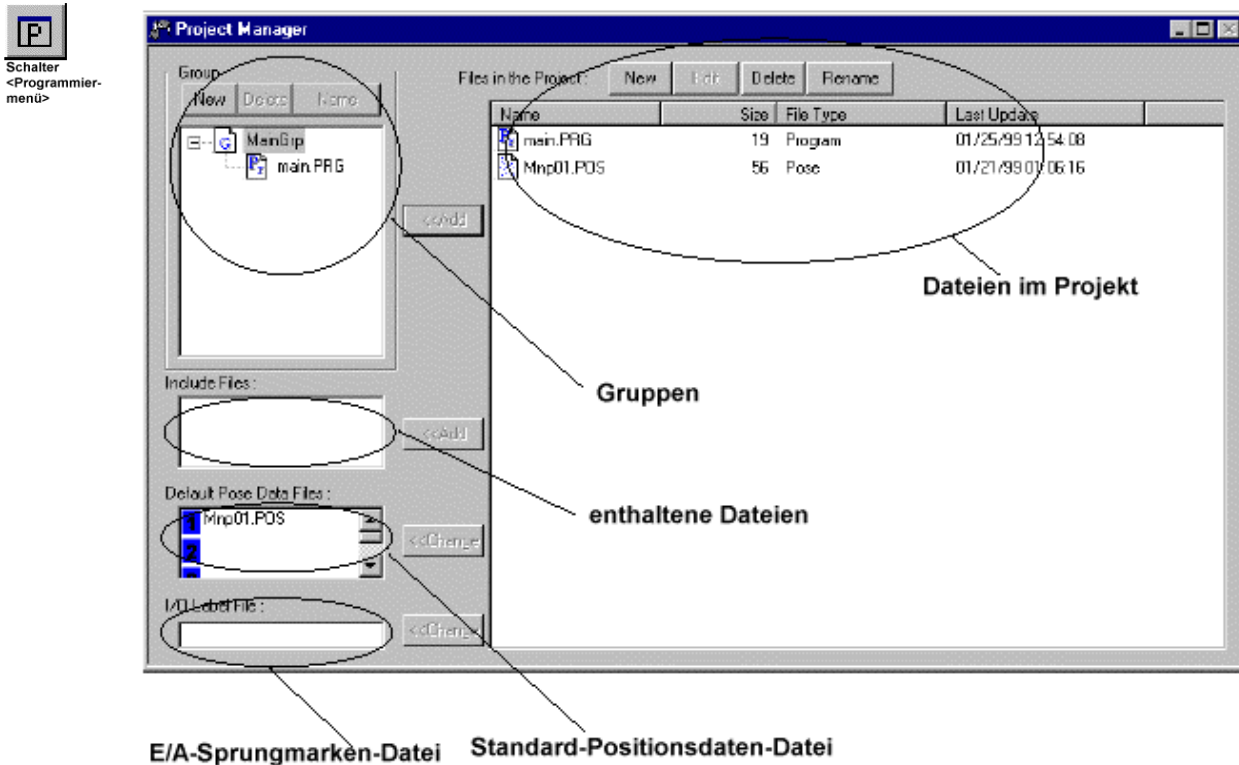
Damit ein Projekt korrekt funktioniert, muss jeder Manipulator zumindest über eine registrierte Positionsdaten-Datei verfügen.

Die Anfangs-Positionsdaten-Datei ist definiert als Positionsdaten-Datei, die der Anwender selbst in der "Gruppe" registriert.

Wenn eine Anfangs-Positionsdaten-Datei in der Projekt-Startgruppe registriert ist (die geladen wird, wenn SPEL 95 startet), werden automatisch die Positionsdaten aus dieser Datei geladen. Ist keine Anfangs-Positionsdaten-Datei in der Projekt-Startgruppe registriert, lädt das System automatisch die Daten aus der Standard-Positionsdaten-Datei.

### 4.1.5 Projekt-Manager-Anzeige

Klicken Sie in der Menüleiste auf [Project] (Projekt) und dann auf [Project Manager] (Projektmanager). Oder klicken Sie auf den Schalter <Programming Pane> (Programmiermenü) in der Werkzeugleiste.



[Files in the Project] (Dateien im Projekt)

In diesem Feld werden alle im Projekt vorhandenen Dateien angezeigt und verwaltet. Es gibt verschiedene Funktionen, wie z.B. Erzeugen, Löschen und Editieren.

[Group box] (Gruppenauswahl)

Gruppen, ausführbare Programmdateien und Anfangs-Positionsdaten-Dateien innerhalb des Projekts werden innerhalb der Gruppenauswahl verwaltet. Es gibt verschiedene Funktionen, wie Erzeugen und Löschen von Gruppen und Anzeigen der Dateistruktur, etc.

[Include Files] (enthaltene Dateien)

Die hier angezeigten Dateien beziehen sich auf Programme, die einen INCLUDE-Befehl enthalten. Informationen, wie im Programm verwendete Variablen und durch Makroanweisungen oder define-Pseudo-Befehle angelegte Definitionen von Identifizierern, sind in diesen Dateien eingeschlossen.

[Default Pose data Files] (Standard-Positionsdaten-Dateien)

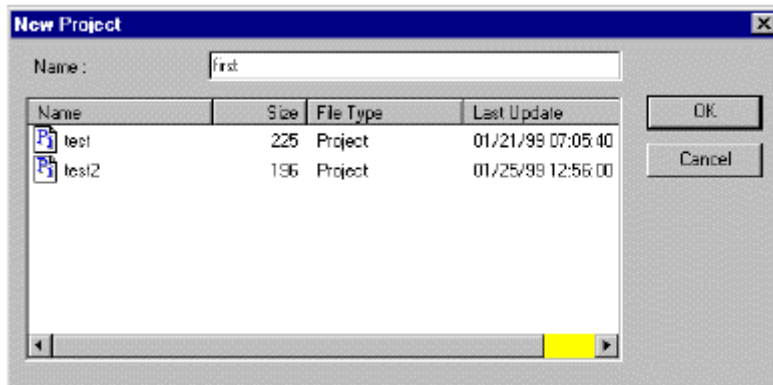
Hier werden die Standard-Positionsdaten-Dateien verwaltet. Sie können die Standard-Positionsdaten-Dateien ändern, indem Sie auf den Schalter <change> (ändern) klicken.

[I/O Label File] (E/A-Sprungmarken-Datei)

Hier werden alle für die Ein-/Ausgänge der Anwender- und Speicher-E/As verwendeten Sprungmarken verwaltet.

## 4.2 Erzeugen von Projekten

Klicken Sie in der Menüleiste auf [Project] (Projekt) und dann auf [New] (Neu).



Geben Sie einen Namen für das neu zu erzeugende Projekt ein.

Ein Projektname wird wie folgt definiert:

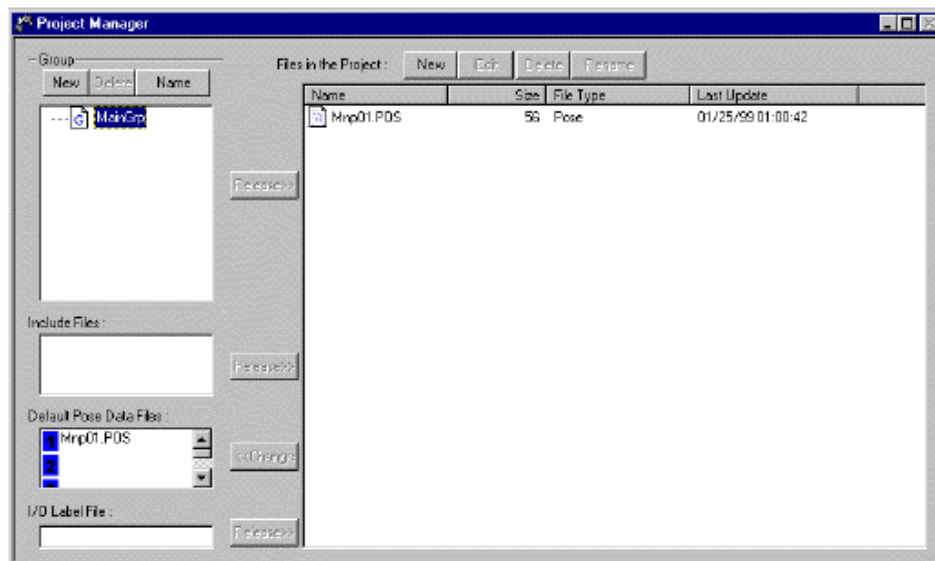
Laufwerksbezeichnung:\SPEL Verzeichnisname\Projekt\Projektverzeichnisname\Projektdateiname.prj

Beispiel: C:\PSPPELWORK\PROJECT\first\first.prj

- Wird der Projektname eingegeben, wird ein Verzeichnis mit dem entsprechenden Namen und eine Datei mit der Endung .prj erzeugt.
- Projektverzeichnisname und Projektdateiname sind identisch. Im o.g. Beispiel wurde als Projektname „first“ eingegeben.
- Wurde SPEL 95 im Standardverzeichnis C:\PSPPELWORK installiert, sollte der Projektname nicht mehr als 116 halb breite Zeichen enthalten. Ein voll breites Zeichen zählt als zwei Zeichen. Leerzeichen können auch verwendet werden. Die folgenden Zeichen können nicht verwendet werden: \,/,\*,?,",<,> und |.
- Es wird nicht zwischen Groß- und Kleinschreibung unterschieden.
- Ist SPEL 95 in einem anderen als dem PSPPELWORK-Verzeichnis installiert, können 259 halb breite Zeichen für die Eingabe von der Laufwerksbezeichnung bis zur Endung .prj benutzt werden.

Wählen und geben Sie einen Projektnamen ein, der den o.g. Bedingungen entspricht.

Das neue Projekt wird, wie unten dargestellt, erzeugt.

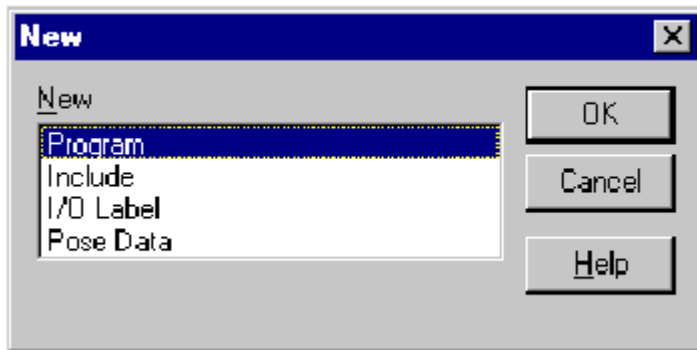


Die Gruppe (MainGrp) und die Standard-Positionsdaten-Datei (Mnp01.POS) wurden automatisch erzeugt. Wenn Sie mehr als einen Manipulator verwenden, müssen Sie zuerst den entsprechenden Manipulator auswählen. Weitere Informationen hierzu erhalten Sie im Handbuch: „SRC5\*\* Optionaler Multi-Manipulator“.

#### 4.2.2 Erzeugen einer Programmdatei

Klicken Sie auf den Schalter <New> im Bereich [Files in the Project] oder wählen Sie [File]-[New] (Datei-Neu) aus der Menüleiste.

Um eine neue Datei zu erzeugen, wählen Sie „Programm“ und klicken dann auf <OK>.



Das Fenster zum Editieren von Programmen wird geöffnet. Der Cursor befindet sich in der linken oberen Ecke des Fensters.

Hier können Sie Ihr Programm eingeben.



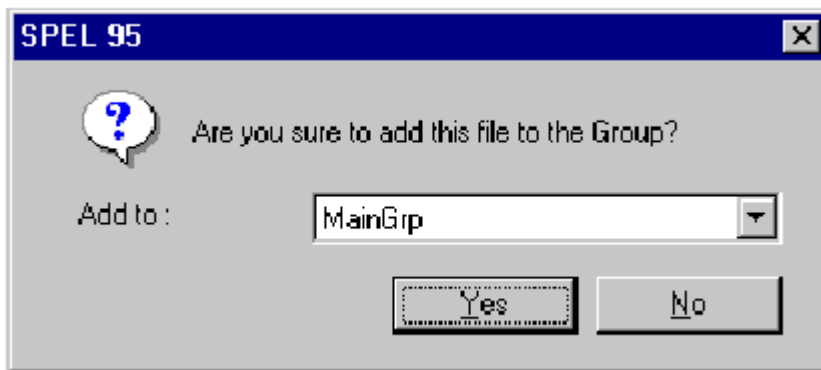
Die maximale Programmgröße beträgt 64 KB.

Geben Sie Ihr Programm ein, und sichern Sie es. Wählen Sie [File]-[Save As] (Datei – Sichern als).

Das nachstehende Dialogfenster erscheint. Vergeben Sie hier einen Dateinamen, und klicken Sie dann auf <OK>. Ein Programmdateiname (.PRG-Datei) kann 31 oder weniger normal breite Zeichen (einschließlich Endung) enthalten.

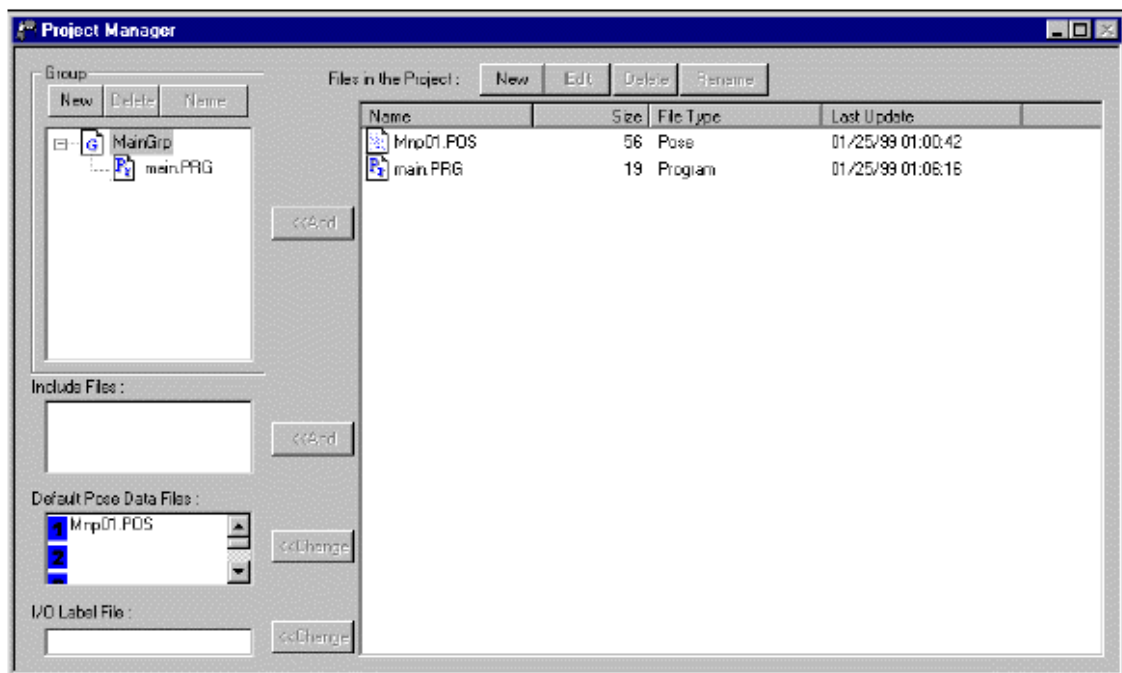


Sie werden gefragt, ob Sie die Datei einer Gruppe hinzufügen möchten. Wollen Sie dies tun, wählen Sie aus der Auswahlliste [Add to] (Hinzufügen) die gewünschte Gruppe, und klicken dann auf <Yes> (Ja).



MAIN.PRG wurde in der [Files in the Project]-Liste registriert. Außerdem wurde die Datei in der Gruppe MainGrp registriert und hat jetzt einen ausführbaren Status.

Um das Programm auszuführen, fügen Sie die Programmdateien der [Group] (Gruppe) zu. Werden die Programmdateien keiner Gruppe hinzugefügt, lassen sich die Programme nicht ausführen!

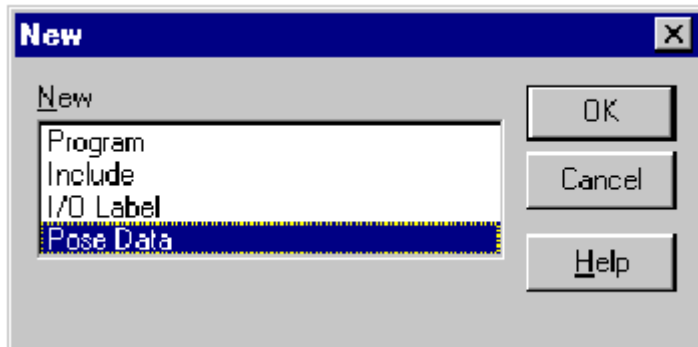


### 4.2.3 Registrieren von Positionsdaten-Dateien

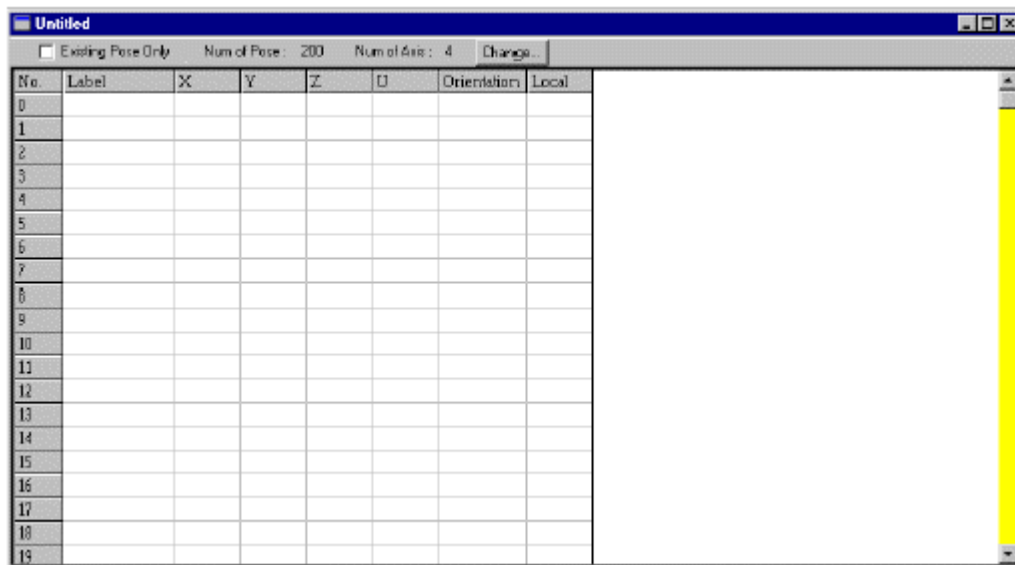
In der Regel werden Positionsdaten mittels Einrichten im Speicher der Steuerung als eine Positionsdaten-Datei gespeichert. Die Positionsdaten-Datei kann bearbeitet werden. Weitere Informationen zum Einrichten erhalten Sie in Kapitel 2 „Einrichten“.

- **Erzeugen neuer Positionsdaten-Dateien**

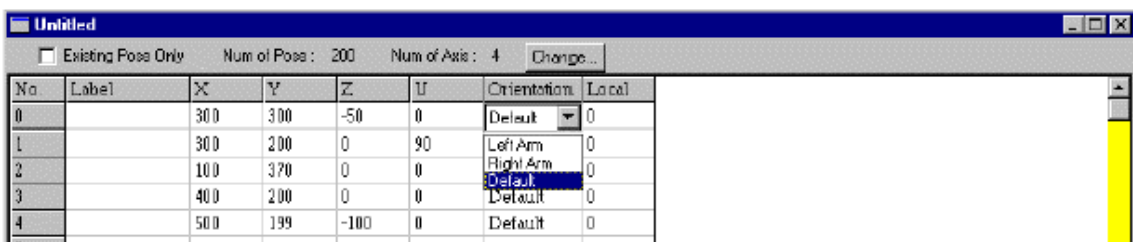
Klicken Sie auf <New> (Neu) im Bereich [Files in the Project], oder wählen Sie [File]-[New] (Datei – Neu) aus der Menüleiste.



Wählen Sie [Pose Data] (Positionsdaten) aus der Liste aus, und klicken Sie auf <OK>. Das Fenster zum Bearbeiten der Koordinaten wird angezeigt. Bewegen Sie den Mauscursor auf die zu editierende Zelle, und klicken Sie in sie hinein. Jetzt können Sie Daten eingeben.



In der Spalte [Orientation] (Ausrichtung) gibt es Auswahllisten. Die Auswahlliste öffnet sich, wenn Sie auf den Pfeil klicken. Dann können Sie eine Ausrichtung durch Anklicken auswählen.



Wählen Sie [File]-[Save As] (Datei – Sicher als).

Das nachstehende Dialogfenster erscheint. Vergeben Sie hier einen Dateinamen, und klicken Sie dann auf <OK>. Es gibt Einschränkungen bei der Anzahl der Zeichen für einen Positionsdatei-Namen (.POS-Datei). Weitere Informationen erhalten Sie in Kapitel 4.2.6 „Einschränkungen bei Dateinamen“.



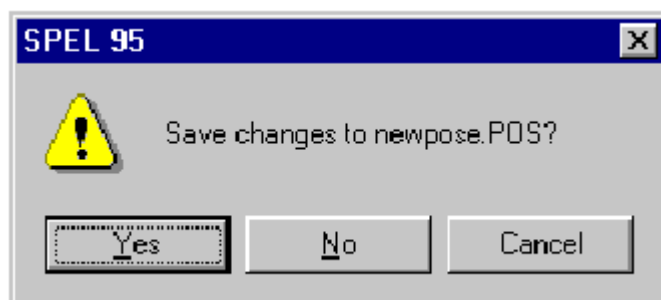
- **Editieren einer Positionsdaten-Datei**

Wählen Sie die Positionsdaten-Datei aus der [Files in the Project]-Liste aus, die Sie bearbeiten möchten, und klicken Sie auf <Edit> (Bearbeiten). Oder Doppelklicken Sie auf die entsprechende Datei im Fenster [Files in the Project] oder auf die Standard-Positionsdaten-Datei.

No.	Label	X	Y	Z	U	Orientation	Local
0		300	300	-50	0	Default	0
1		300	200	0	90	Default	0
2		100	370	0	0	Default	0
3		400	200	-534	0	Default	0
4		500	199	-100	0	Default	0
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							

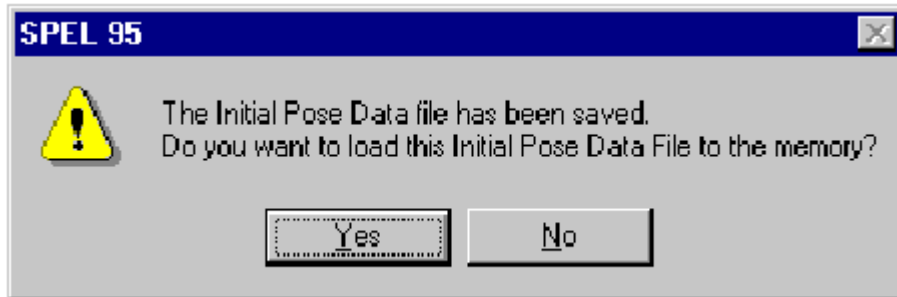
Die vom [Pose Data Manager] gelesene Datei kann nicht geöffnet werden, während das [Pose Data Manager]-Fenster geöffnet ist.

Klicken Sie nach dem Bearbeiten der Daten auf <Close> (Schließen) . Folgende Abfrage wird angezeigt. Klicken Sie auf <Yes> (Ja), wenn Sie die Änderungen an der Datei speichern möchten.

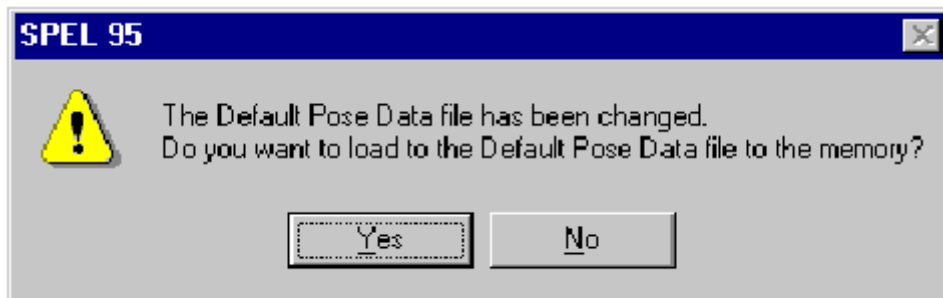




Wenn eine Anfangs-Positionsdaten-Datei bearbeitet wurde, erscheint eine Abfrage, ob sie in den Speicher geschrieben werden soll. Klicken Sie auf <Yes>, um die Daten sofort zur Verfügung zu haben. Wenn Sie auf <No> klicken, werden die Dateidaten neu geschrieben, aber die Daten im Speicher der Steuerung bleiben unverändert.

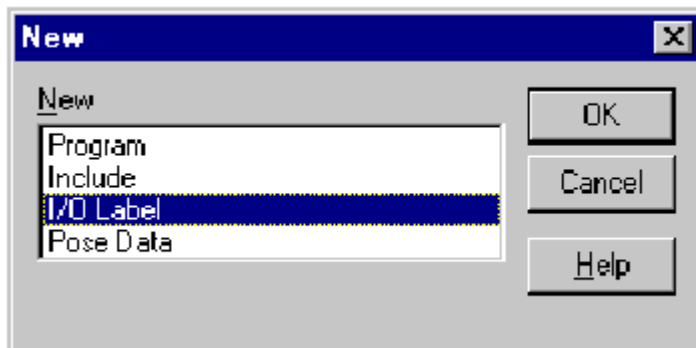


Wenn eine Standard-Positionsdaten-Datei bearbeitet wurde, erscheint eine Abfrage, ob sie in den Speicher geschrieben werden soll. Klicken Sie auf <Yes>, um die Daten sofort zur Verfügung zu haben. Wenn Sie auf <No> klicken, werden die Dateidaten neu geschrieben, aber die Daten im Speicher der Steuerung bleiben unverändert.

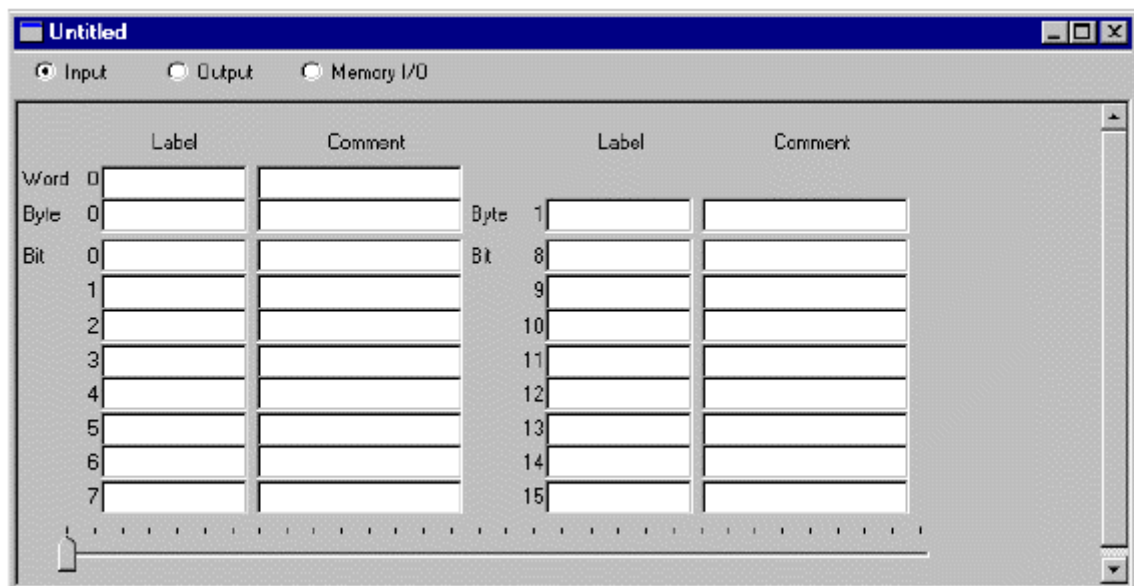


#### 4.2.4 Erzeugen einer E/A-Sprungmarkendatei

Klicken Sie auf <New> (Neu) im Bereich [Files in the Project], oder wählen Sie [File]-[New] (Datei – Neu) aus der Menüleiste.



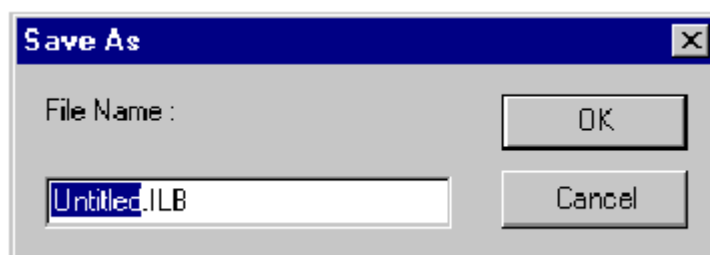
Wählen Sie I/O Label (E/A Sprungmarke) aus der Liste aus, und klicken Sie auf <OK>. Das Fenster zum Bearbeiten der E/A-Sprungmarkendatei wird angezeigt.



Sie können bis zu 16 Zeichen für eine E/A-Sprungmarke eingeben.

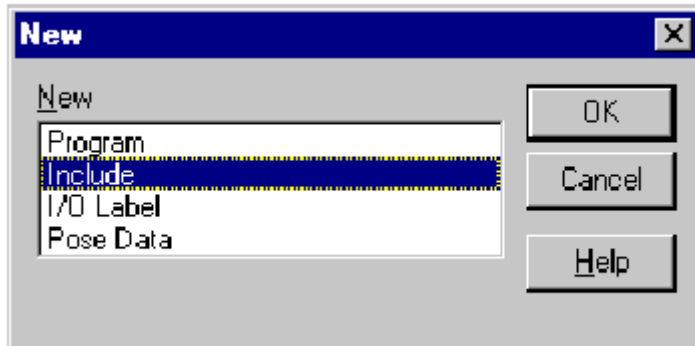
Nach dem Eingeben der Daten wählen Sie [File]-[Save As] (Datei – Sicher als).

Das nachstehende Dialogfenster erscheint. Vergeben Sie hier einen Dateinamen, und klicken Sie dann auf <OK>. Es gibt Einschränkungen bei der Anzahl der Zeichen für einen E/A-Sprungmarkendatei-Namen (.ILB-Datei). Weitere Informationen erhalten Sie in Kapitel 4.2.6 „Einschränkungen bei Dateinamen“.



#### 4.2.5 Erzeugen einer INCLUDE-Datei (enthaltene Datei)

Klicken Sie auf <New> (Neu) im Bereich [Files in the Project], oder wählen Sie [File]-[New] (Datei – Neu) aus der Menüleiste.

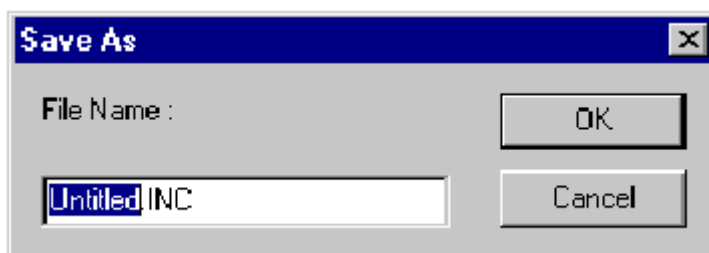


Wählen Sie Include (Enthalten) aus der Liste aus, und klicken Sie auf <OK>. Das Fenster zum Bearbeiten der Include-Datei wird angezeigt.



Geben Sie Ihr Programm in diesem Fenster ein, und sichern Sie es nach der Eingabe. Dazu wählen Sie [File]-[Save As] (Datei – Sichern als).

Das nachstehende Dialogfenster erscheint. Vergeben Sie hier einen Dateinamen, und klicken Sie dann auf <OK>. Es gibt Einschränkungen bei der Anzahl der Zeichen für einen Include-Datei-Namen (enthaltene Dateien .INC-Datei). Weitere Informationen erhalten Sie in Kapitel 4.2.6 „Einschränkungen bei Dateinamen“.



## 4.2.6 Einschränkungen bei Dateinamen

Bei der Eingabe von Dateinamen müssen Sie nachfolgende Regeln beachten:

### ▪ Programmdateinamen

Die Programmdateinamen unterliegen folgenden Beschränkungen:

- Die Länge des Namens darf 31 Zeichen (inklusive Erweiterung) nicht überschreiten. Leerzeichen können auch verwendet werden.
- Ein voll breites Zeichen wird als Zwei-Byte-Zeichen gezählt. Die folgenden Zeichen dürfen nicht verwendet werden: \, /, :, \*, ?, ,, <, > und |..
- Dateinamen dürfen nicht kasus-sensitiv sein.

### ▪ Include-Datei/E/A-Sprungmarkendatei/Positionsdaten-Datei

- Ein voll breites Zeichen wird als Zwei-Byte-Zeichen gezählt. Die folgenden Zeichen dürfen nicht verwendet werden: \, /, :, \*, ?, ,, <, > und |..
- Dateinamen dürfen nicht kasus-sensitiv sein.

Die Anzahl der Zeichen im Namen einer Include-Datei (.INC-Datei), einer E/A-Sprungmarkendatei (.ILB-Datei) oder einer Positionsdaten-Datei (.POS-Datei) wird wie folgt definiert:

Beispiel für eine Positionsdaten-Datei:

Laufwerksbezeichnung:\SPEL-Verzeichnisname\Projekt\Projektname\Dateiname.pos

Beispiel: C:\PSPELWORK\PROJECT\first\MyPose1.pos

- Im o.g. Beispiel wurde als Projektname „first“ und für die Positionsdaten-Datei der Name „MyPose1“ eingegeben.
- Die Länge des Projektnamens wirkt sich auf den Dateinamen aus. Der Projektname steht ein Level höher in der Verzeichnisstruktur. Daher müssen Sie folgendes beachten:
  - Es können maximal 259 halb breite Zeichen von der Laufwerksbezeichnung bis zur Erweiterung (.pos) eingegeben werden.
  - $259 \text{ Zeichen} - (\text{Laufwerksbezeichnung}:\backslash\text{SPEL-Verzeichnisname}\backslash\text{Projekt}\backslash\text{Projektname})\text{- (Erweiterung)} = \text{Anzahl der Zeichen im Namen}$   
 Beispiel:  $259 \text{ Zeichen} - (\text{C}:\backslash\text{PSPELWORK}\backslash\text{PROEJCT}\backslash\text{first}) - (\text{.pos}) = \text{Anzahl der Zeichen im Positionsdaten-Dateinamen}$

Wählen und geben Sie einen Dateinamen ein, der den o.g. Bedingungen entspricht. Die Vergabe von Include- bzw. E/A-Sprungmarken-Dateinamen unterliegt den gleichen Bedingungen/ Einschränkungen.

## 4.3 Im- und Export von Dateien

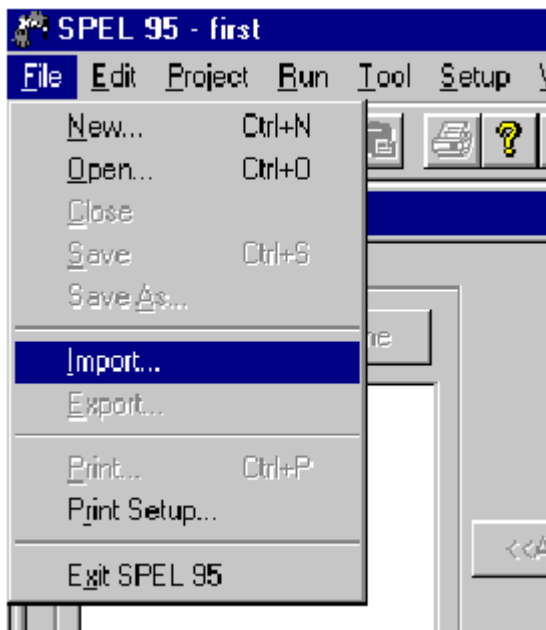
Ein Projekt setzt sich aus zumindest einer Programmdatei, einer Positionsdaten-Datei und einer anwendungsbezogenen Einstellungsdatei zusammen. Die zu verwendenden Dateien müssen nicht notwendigerweise neu erzeugt werden. Es kann sich um Dateien handeln, die aus anderen Projekten stammen oder um Dateien, die sich auf einer anderen Diskette oder einem anderen Laufwerk befinden.

### 4.3.1 Import von Dateien

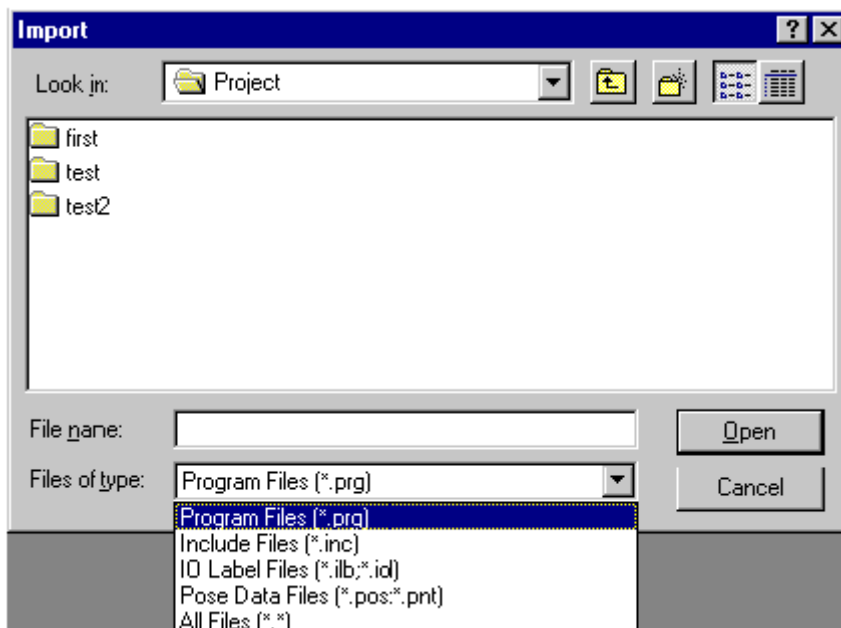
Jede, auch in einer älteren SPEL-Version erzeugte, Datei kann in SPEL 95 importiert werden:

- **Import von Programmdateien**

Wählen Sie [File]-[Import] (Datei-Import) aus der Menüleiste.



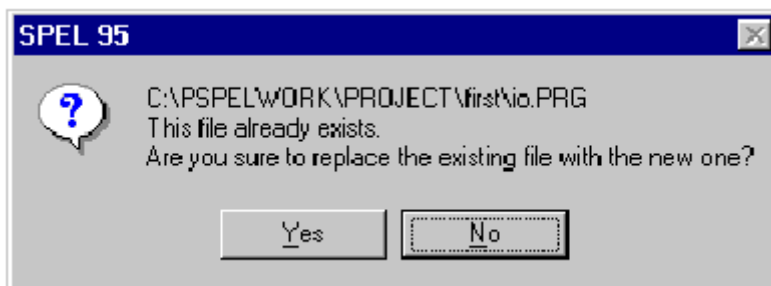
Das Import-Fenster erscheint. Wählen Sie als erstes den Dateityp der zu importierenden Datei aus. In diesem Fall wählen Sie „Programm File“ (Programmdatei) aus der Auswahlliste aus.



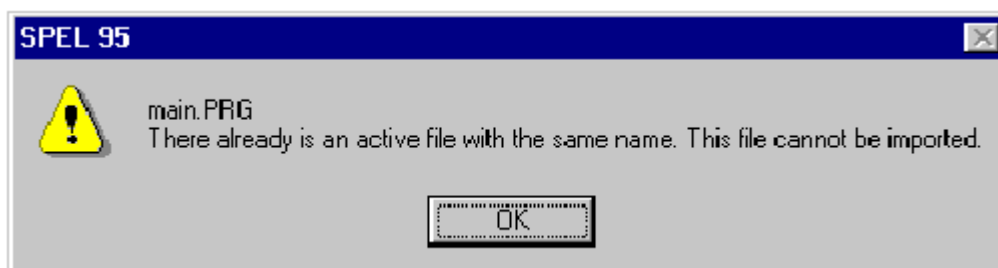
Wählen Sie dann die Zieldatei. Geben Sie den Namen der Datei in das Textfeld „File name“ (Dateiname) ein. Klicken Sie danach auf <Open> (Öffnen).



Die folgende Meldung erscheint, wenn in dem Projekt bereits eine Datei mit dem gleichen Namen vorhanden ist. Um die bestehende Datei durch die neue zu ersetzen, klicken Sie auf <Yes>.

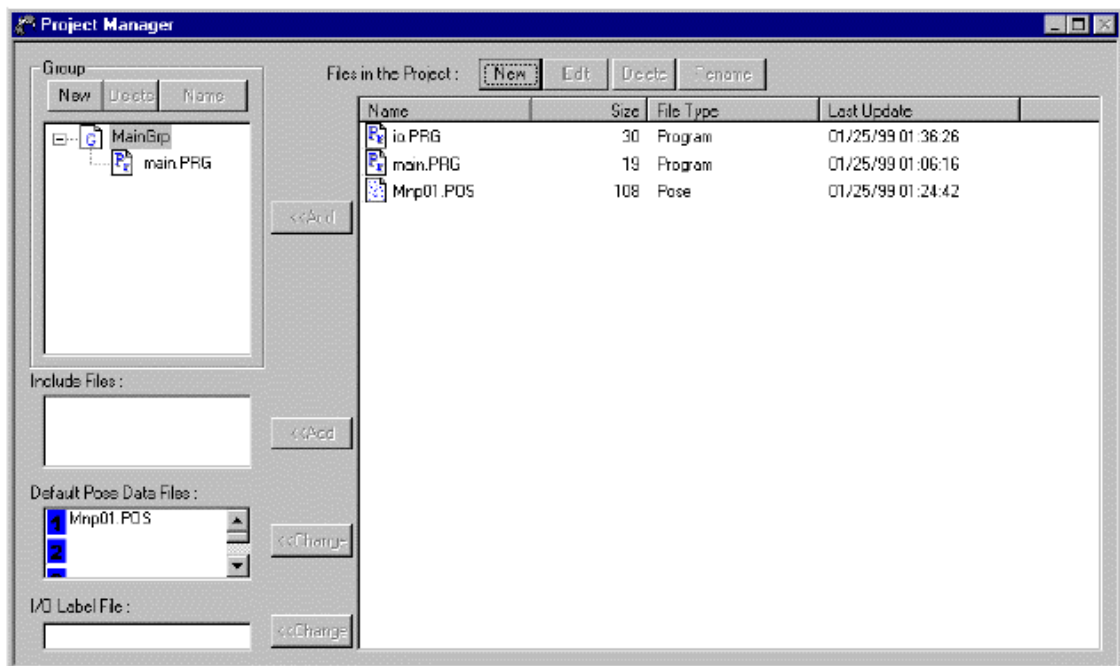


Eine Programmdatei mit dem gleichen Namen einer bereits in der Gruppe registrierten Datei kann nicht importiert werden. Ein Hinweis erscheint:



Um eine Programmdatei mit dem gleichen Namen einer bereits in der Gruppe registrierten Datei erfolgreich importieren zu können, gehen Sie wie folgt vor: (1) geben Sie die Programmdatei aus der Gruppe frei, (2) importieren, und (3) sichern Sie die Datei, indem Sie die alte durch die neue Datei ersetzen.

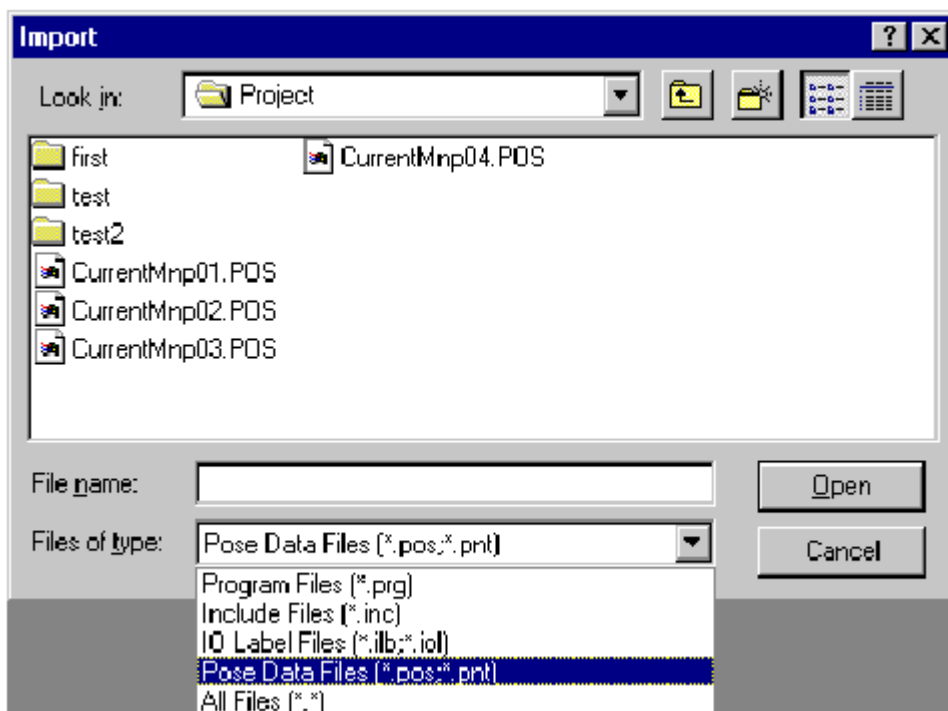
Die importierte Datei wird in der [Files in the Project]-Liste registriert.



Die Programmdatei ist in diesem Stadium nicht in der Gruppe registriert und somit nicht ausführbar. Um Sie in der Gruppe zu registrieren, markieren Sie die zu registrierende Datei in der Anzeige [Files in the Project] und klicken dann auf <Add> (Hinzufügen).

#### ▪ Import von Programmdateien

Wählen Sie [File]-[Import] (Datei-Import) aus der Menüleiste. Das Import-Fenster erscheint. Wählen Sie als erstes den Dateityp der zu importierenden Datei aus. Wählen Sie „Pose Data File“ (Positionsdaten-Datei) aus der Auswahlliste aus.

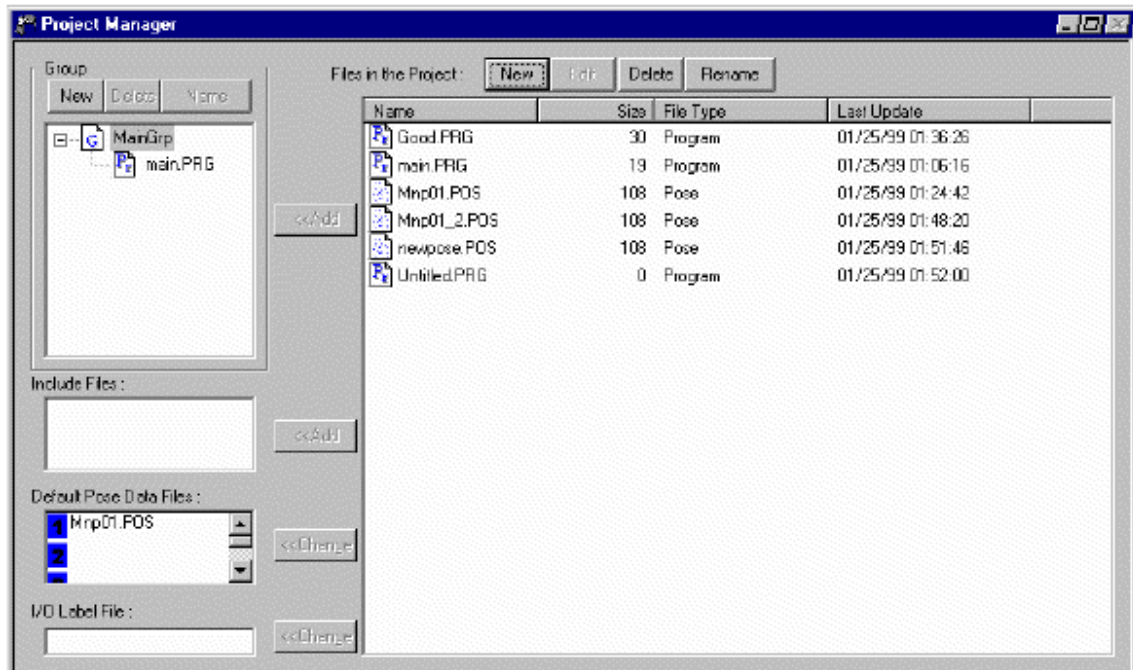


Wählen Sie das Verzeichnis, aus dem Sie die Daten laden möchten, und geben Sie den Dateinamen in dem Textfeld „Filename“ ein. Klicken Sie dann auf <Open> (Öffnen).



SPEL 95 kann herkömmliche SPEL-Positionsdaten-Dateien laden (Die Erweiterung der herkömmlichen Positionsdaten-Dateien ist \*.pnt). In diesen Fällen wird der Dateityp automatisch konvertiert, und die Erweiterung ändert sich zu \*.pos.

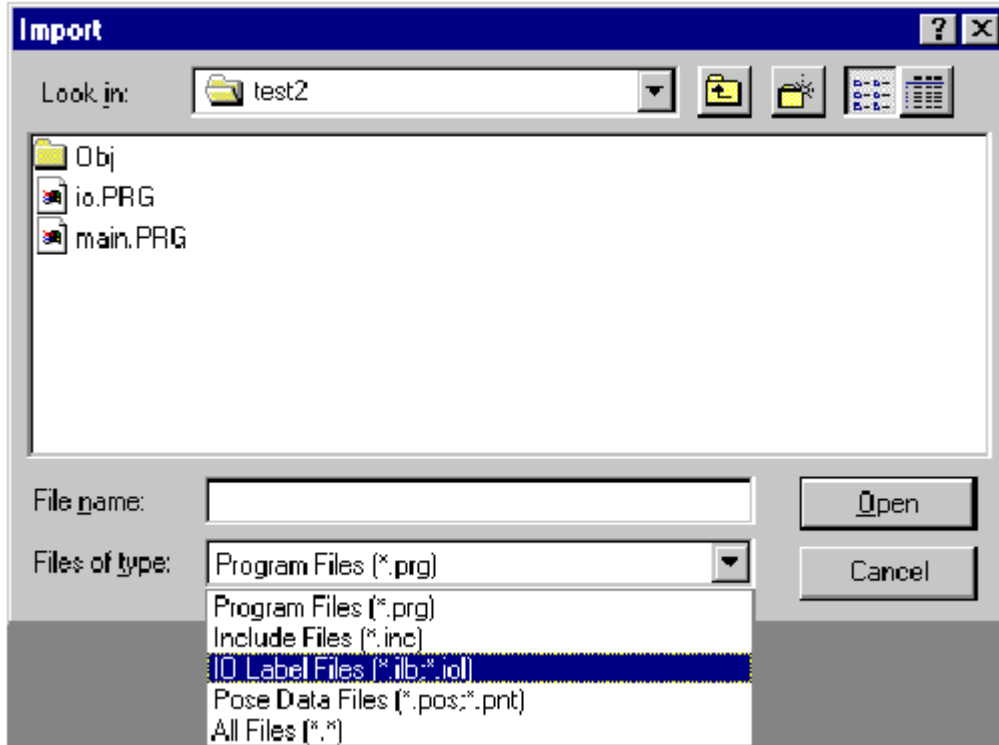
Importierte Dateien werden in der Liste [Files in the project] registriert.



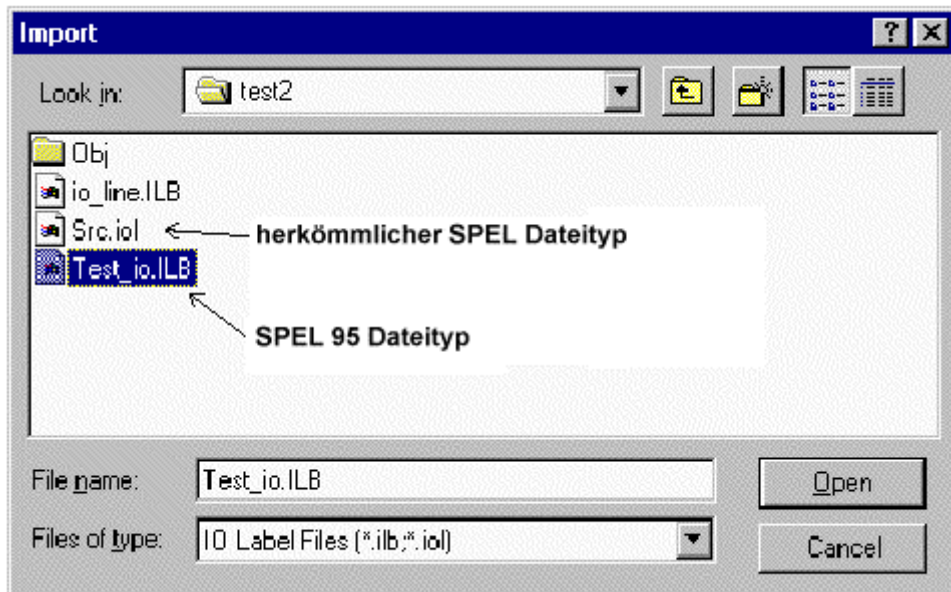


- **Laden von E/A-Sprungmarken-Dateien**

Wählen Sie [File]-[Import] (Datei-Import) aus der Menüleiste. Das Import-Fenster erscheint. Wählen Sie als erstes den Dateityp der zu importierenden Datei aus. Wählen Sie „I/O Label Files“ (E/A-Sprungmarken-Dateien) aus der Auswahlliste aus.

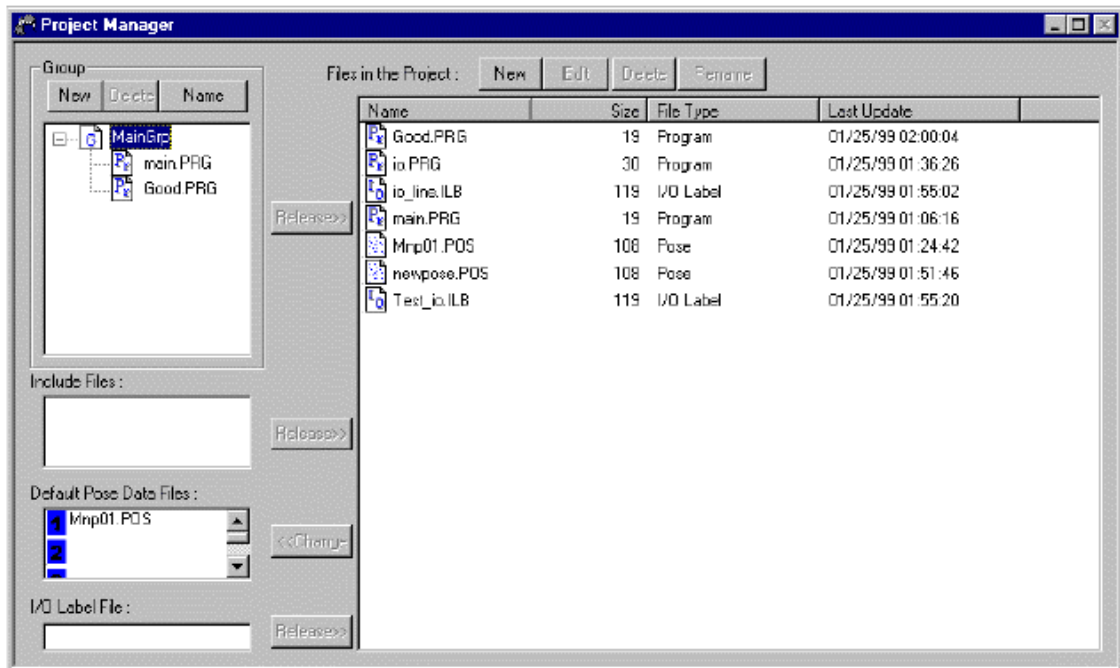


Wählen Sie das Verzeichnis, aus dem Sie die Daten laden möchten, und geben Sie den Dateinamen in dem Textfeld „Filename“ ein. Klicken Sie dann auf <Open> (Öffnen).



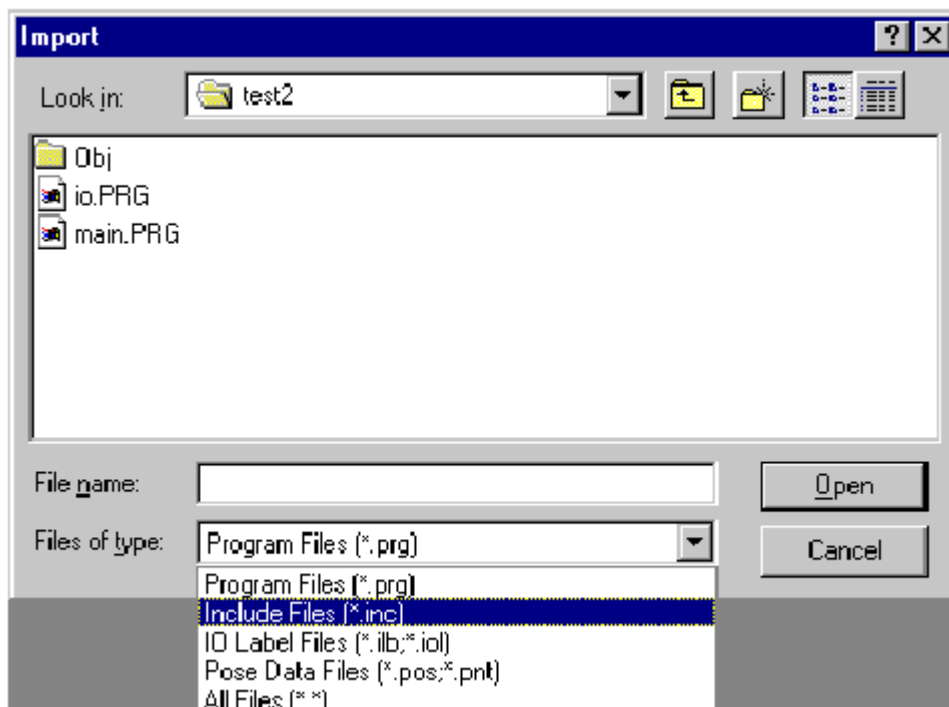
SPEL 95 kann herkömmliche SPEL-E/A-Sprungmarken-Dateien laden (Die Erweiterung der herkömmlichen E/A-Sprungmarken-Dateien ist \*.iol). In diesen Fällen wird der Dateityp automatisch konvertiert, und die Erweiterung ändert sich zu \*.ilb.

Importierte Dateien werden in der Liste [Files in the project] registriert.



#### ▪ Laden von Include-Dateien

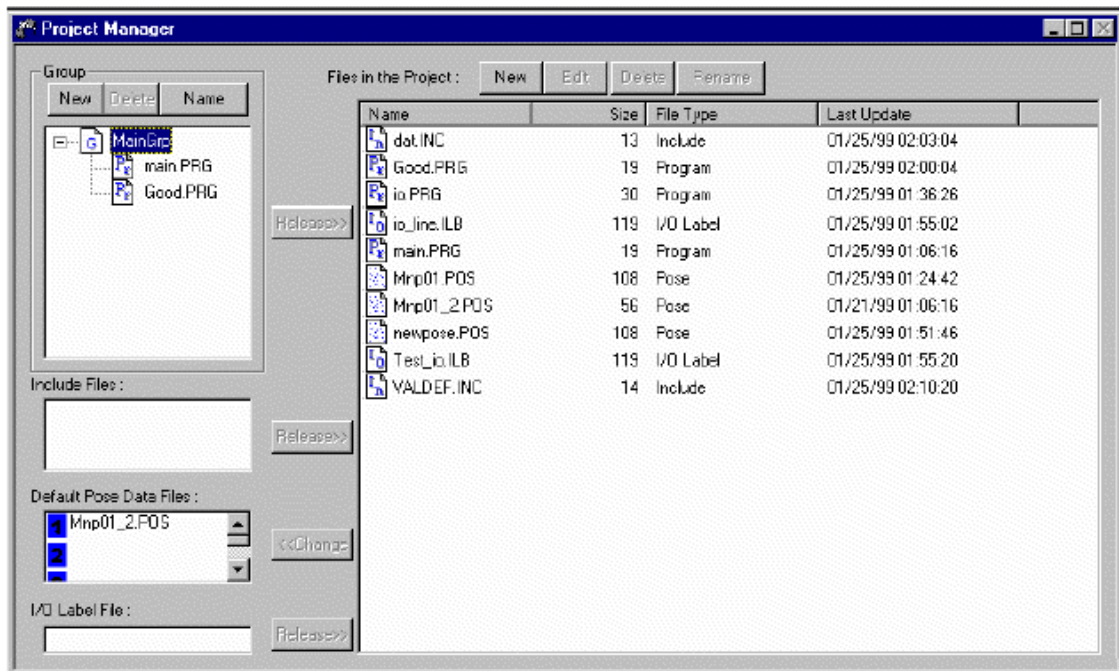
Wählen Sie [File]-[Import] (Datei-Import) aus der Menüleiste. Das Import-Fenster erscheint. Wählen Sie als erstes den Dateityp der zu importierenden Datei aus. Wählen Sie „Include Files“ (Include-Dateien) aus der Auswahlliste aus.



Wählen Sie das Verzeichnis, aus dem Sie die Daten laden möchten, und geben Sie den Dateinamen in dem Textfeld „Filename“ ein. Klicken Sie dann auf <Open> (Öffnen).



Importierte Dateien werden in der Liste [Files in the project] registriert.

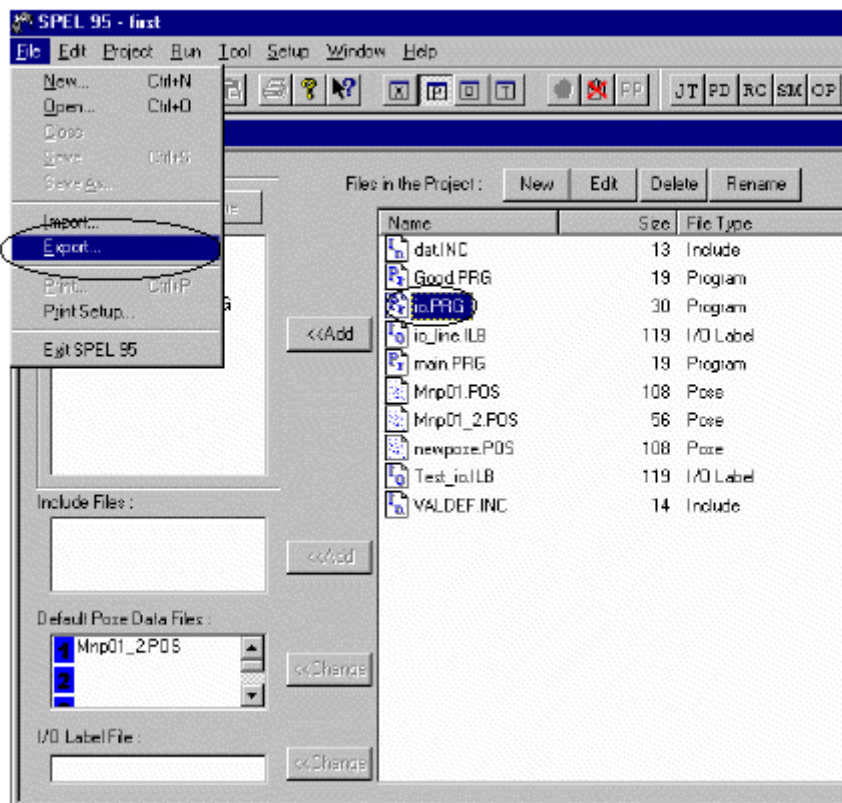


### 4.3.2 Exportieren von Dateien in ein anderes Verzeichnis

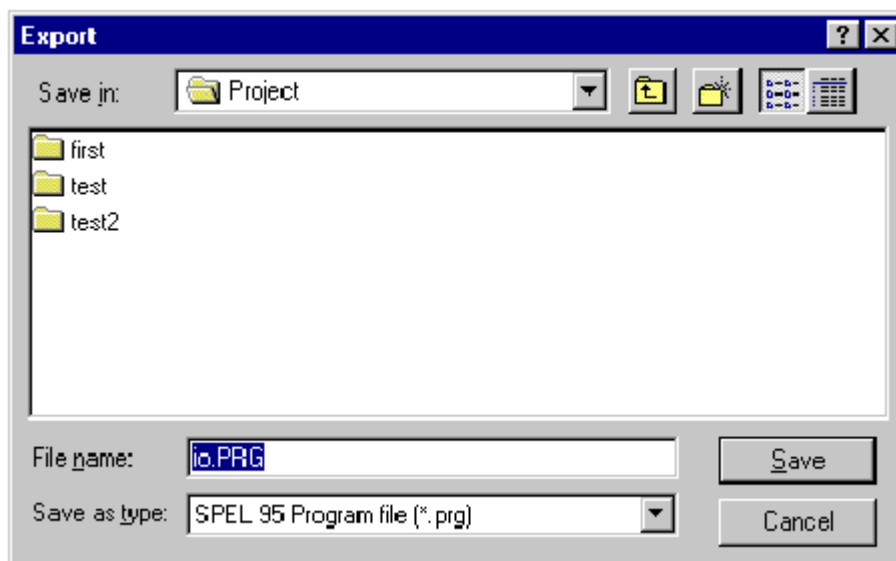
Eine Datei in einem Projekt kann von dem Verzeichnis, in dem sie sich auf der Festplatte befindet, in ein anderes Verzeichnis auf der Festplatte oder auch auf Diskette exportiert bzw. gespeichert werden. Sie lässt sich jedoch nicht in ein anderes Projekt exportieren!

Klicken Sie auf und markieren Sie die zu exportierende Datei in der Liste [Files in the Project].

Wählen Sie [File]-[Export] (Datei-Export) aus der Menüleiste.



Das Export-Fenster erscheint. Wählen Sie das Verzeichnis aus, in dem Sie die Daten speichern möchten, und klicken Sie dann auf <Save> (Speichern).



## 4.4 Dateiverwaltung

Dateien, die in einem Projekt registriert sind, können nicht ausgeführt werden.

Dateien, die in einer Gruppe registriert sind, können nicht gelöscht oder umbenannt werden.

Auf den nachfolgenden Seiten wird erläutert, wie Dateien in SPEL 95 verwaltet werden können.

### 4.4.1 Hinzufügen einer Programm- oder Anfangs-Positionsdaten-Datei zu einer Gruppe

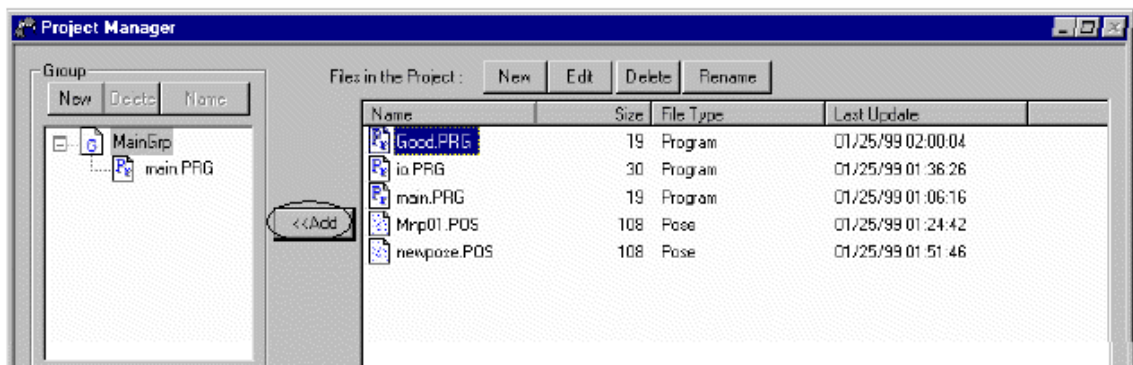
Eine Programmdatei wird ausführbar, wenn sie einer Gruppe angehört.

Eine Positionsdaten-Datei kann entweder als Standard- oder als Anfangs-Positionsdaten-Datei in einer Gruppe registriert werden.

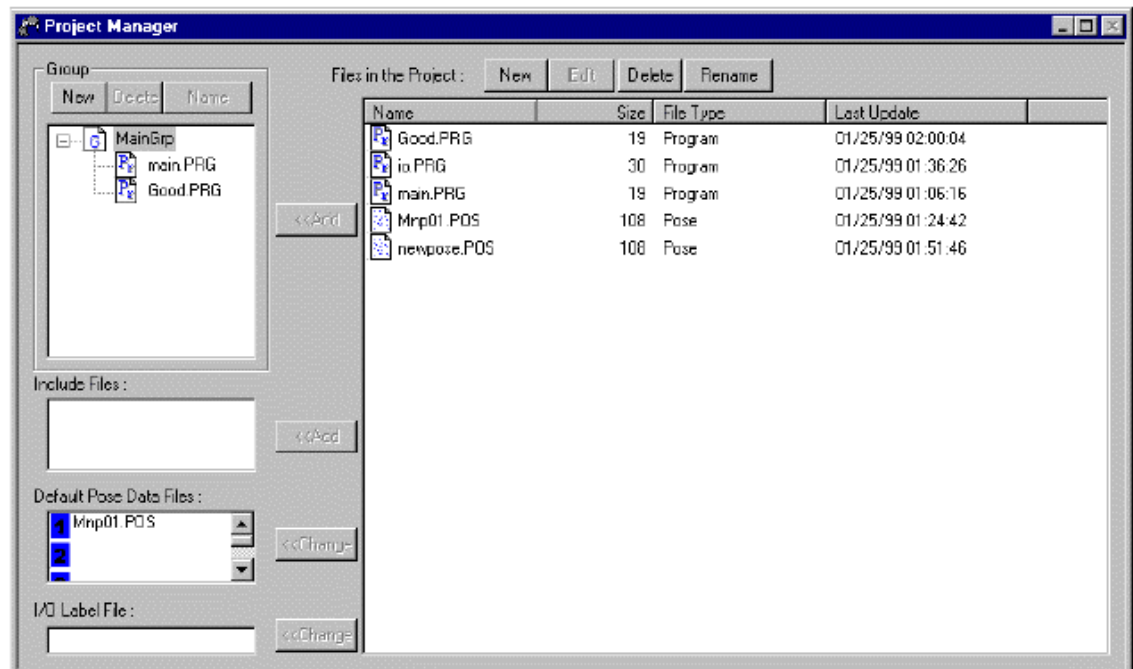
Auf jeden Fall kann die Datei nicht verwendet werden, wenn sie nur in einem Projekt registriert wurde.

Fügen Sie die Programmdatei „Good.PRG“ der Main Group (Hauptgruppe) zu.

Wählen Sie dazu die Datei „Good.PRG“ aus, und klicken Sie auf <<Add> (Hinzufügen).

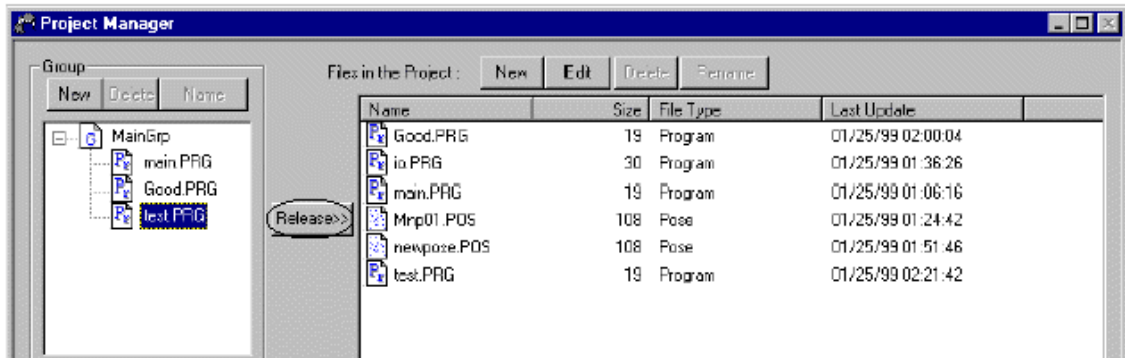


Die Programmdatei „Good.PRG“ wurde in der Main Group registriert.

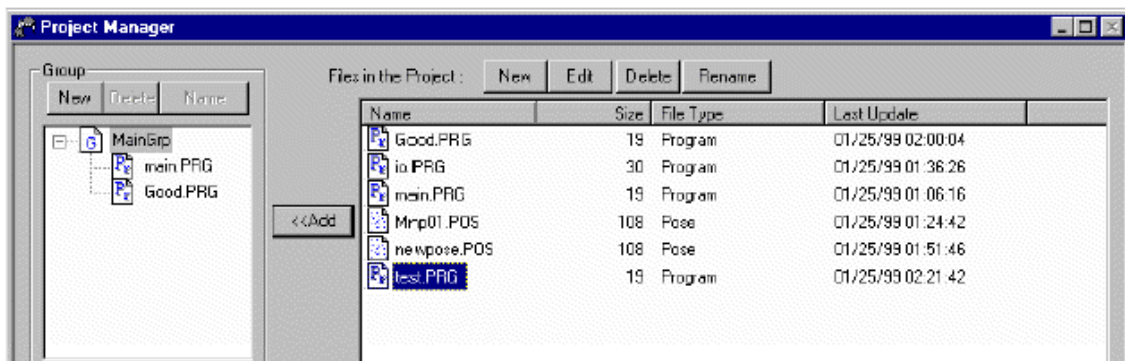


## 4.4.2 Freigeben einer Programm- oder Anfangs-Positionsdaten-Datei von einer Gruppe

Wählen Sie die freizugebende Datei aus der Gruppe, und klicken Sie auf <Release> (Freigeben).



Die Datei wurde von der Gruppe freigegeben.



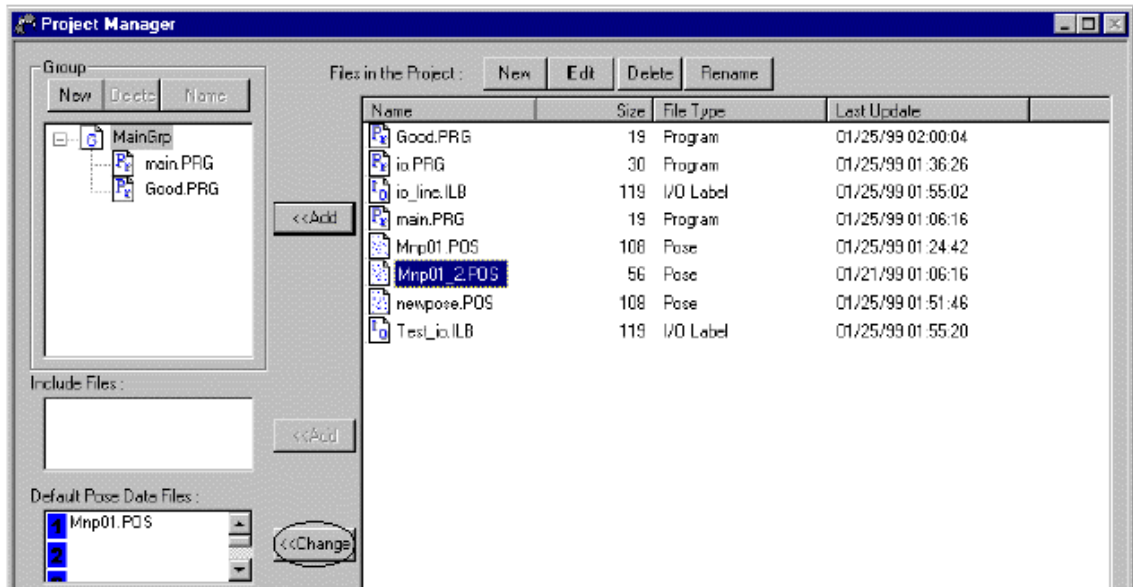
Wenn die Datei von der Gruppe freigegeben wurde, kann sie gelöscht oder umbenannt werden.

## 4.4.3 Ändern von Standard-Positionsdaten-Dateien

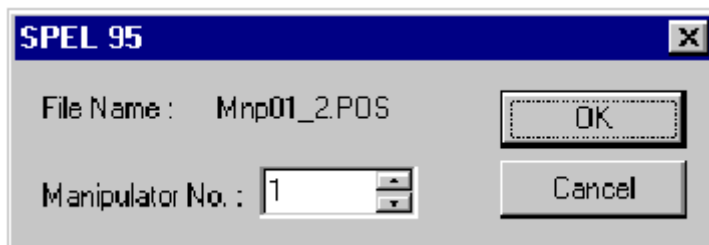
**ACHTUNG**

Wird die Standard-Positionsdaten-Datei geändert, sind davon alle Funktionen, die den Manipulator steuern, betroffen! Unternehmen Sie Schritte, um Störungen an den Peripheriegeräten zu vermeiden, wenn Sie die Standard-Positionsdaten-Datei ändern.

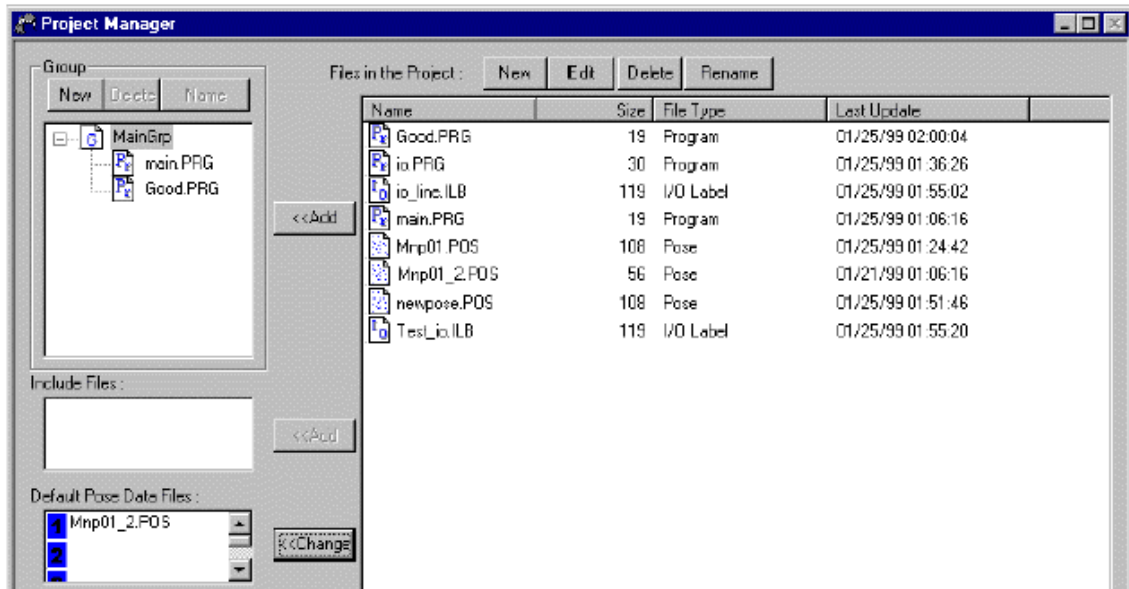
Wählen Sie die Datei, die Sie als Standard-Positionsdaten-Datei registrieren möchten, und klicken Sie auf <Change> (Ändern).



Legen Sie die entsprechende Manipulatornummer fest, und klicken Sie dann auf <OK>.



Die Standard-Positionsdaten-Datei wurde geändert.



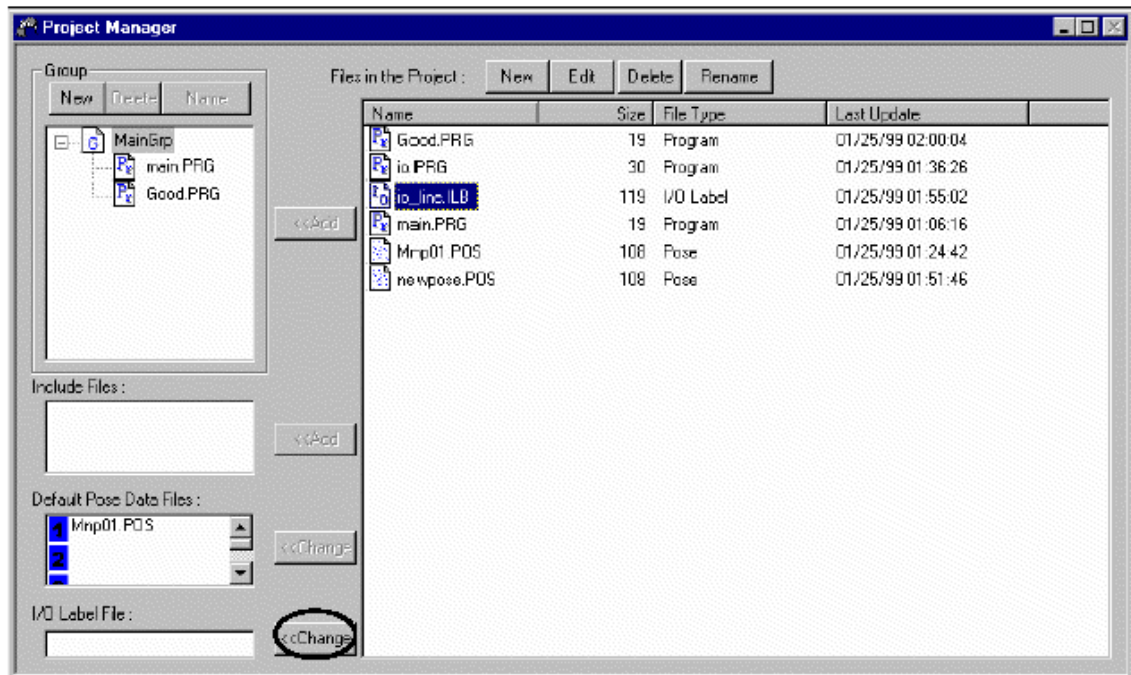


#### 4.4.4 Ändern von E/A-Sprungmarken-Dateien

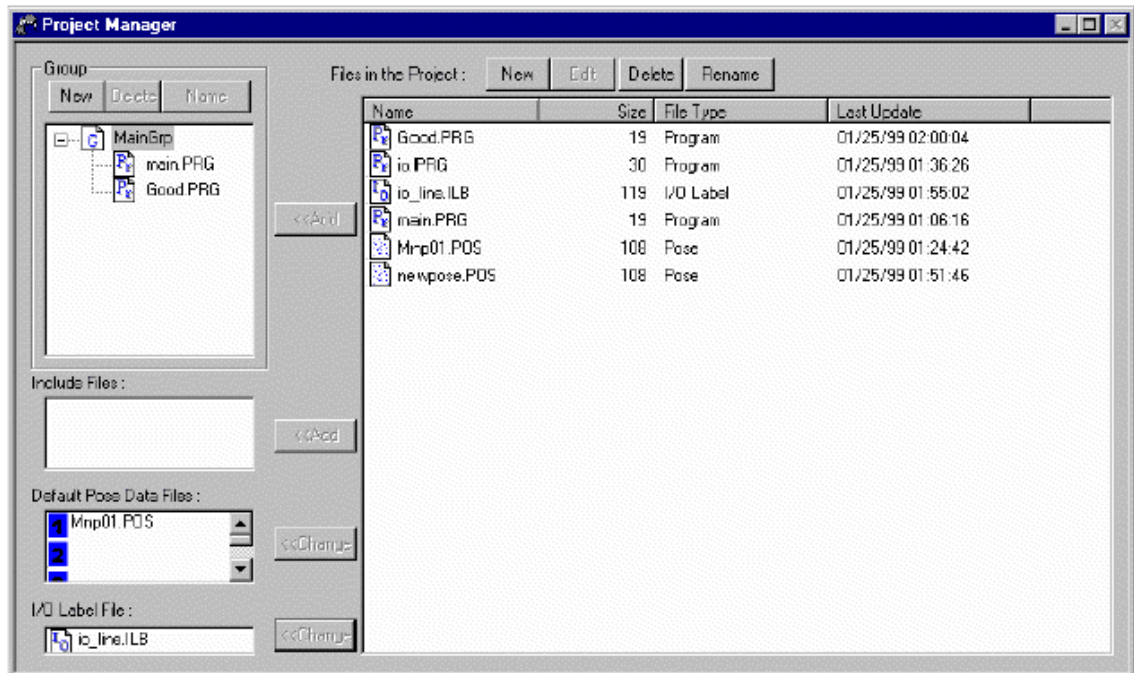
Um den Status einer E/A-Sprungmarken-Datei in einen ausführbaren Zustand zu ändern, müssen Sie sie in der Gruppenbox [I/O Label File] registrieren.

- **Nutzbar machen einer E/A-Sprungmarken-Datei**

Klicken Sie auf und markieren Sie in der Liste [Files in the Project] die Datei, die Sie als E/A-Sprungmarken-Datei registrieren möchten, und klicken Sie dann auf <Change> (Ändern).



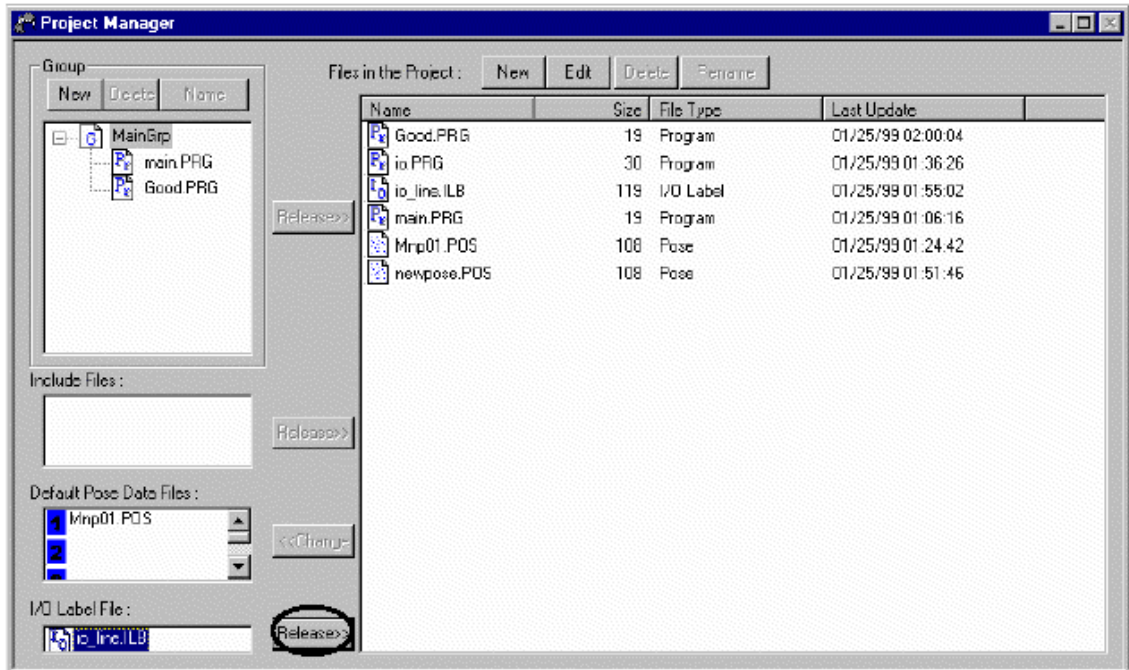
Die Datei wird nutzbar und kann in einem Programm verwendet werden.



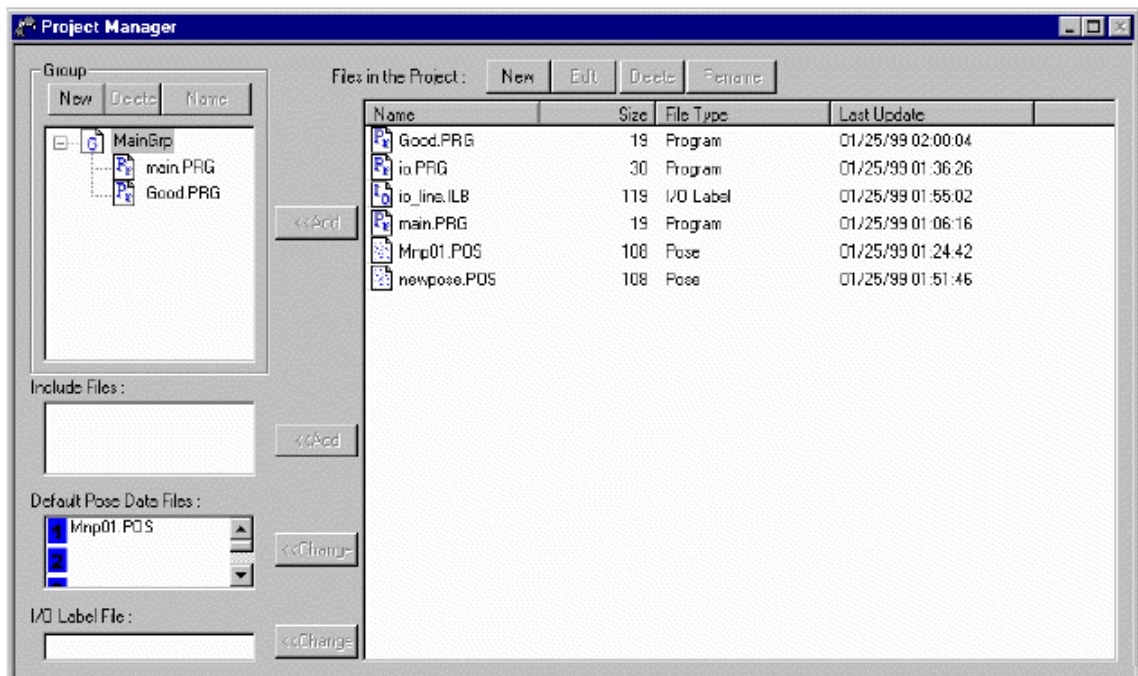
- **Ändern einer aktiven E/A-Sprungmarken-Datei**

Es kann immer nur eine E/A-Sprungmarken-Datei aktiv sein. Wenn Sie eine andere Datei nutzbar machen wollen, muss zuerst die aktuelle Datei freigegeben werden.

Klicken Sie auf die aktuelle E/A-Sprungmarken-Datei, um sie zu markieren und dann auf <Release> (Freigeben).



Die Datei wurde freigegeben.

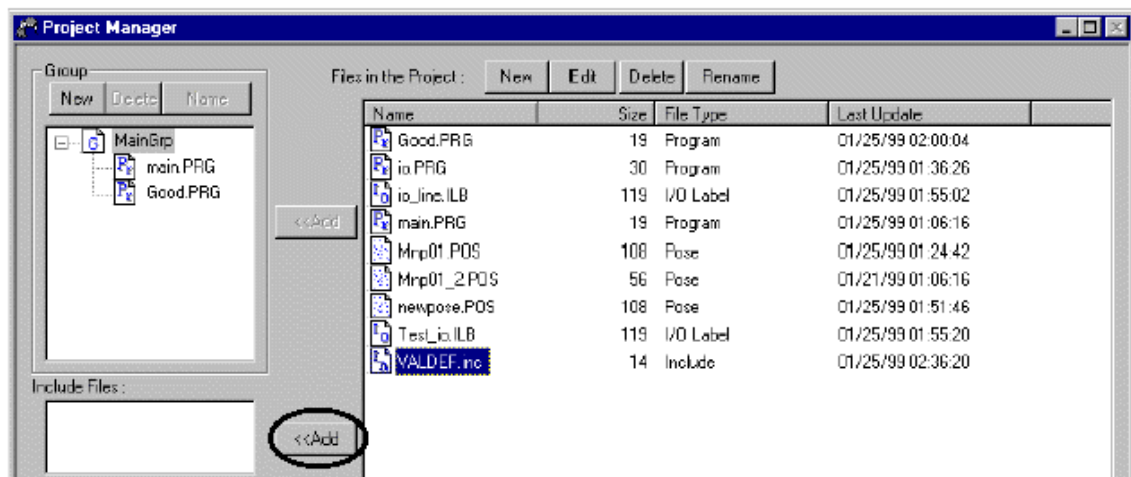


#### 4.4.5 Ändern von Include-Dateien

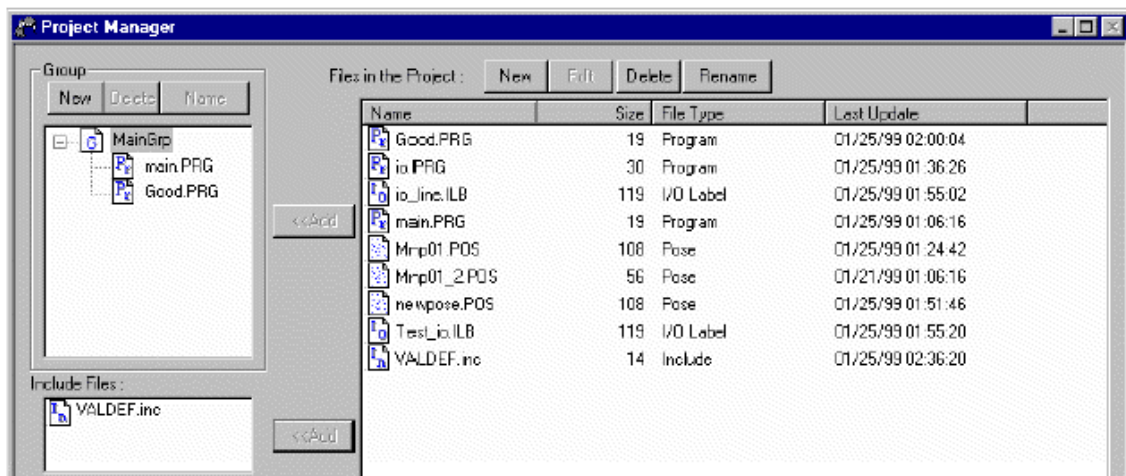
Wenn Sie Include-Dateien verwenden wollen, müssen Sie sie in der Gruppenbox [Include Files] registrieren.

- **Nutzbar machen einer Include-Datei**

Klicken Sie auf und markieren Sie in der Liste [Files in the Project] die Datei, die Sie als Include-Datei registrieren möchten, und klicken Sie dann auf <<Add> (Hinzufügen).

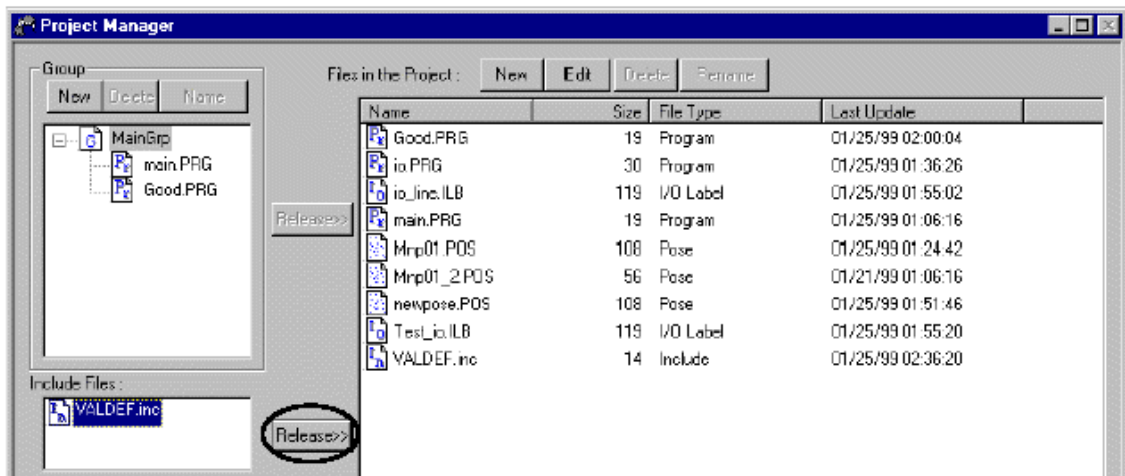


Die Datei wird nutzbar, und Sie können sich in einem Programm auf sie beziehen.

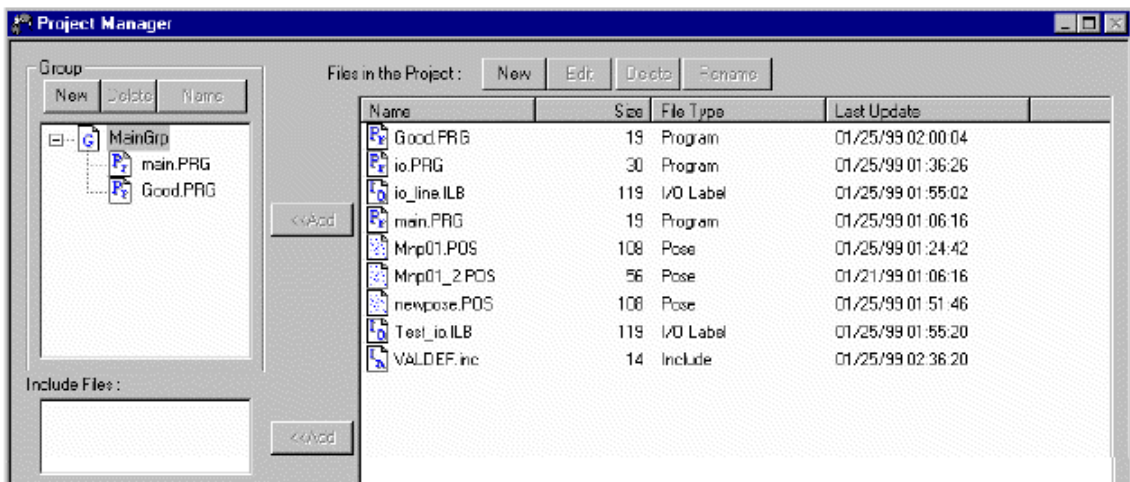


- **Freigeben einer aktiven Include-Datei**

Klicken Sie auf die aktive Include-Datei, um sie zu markieren und dann auf <Release> (Freigeben).

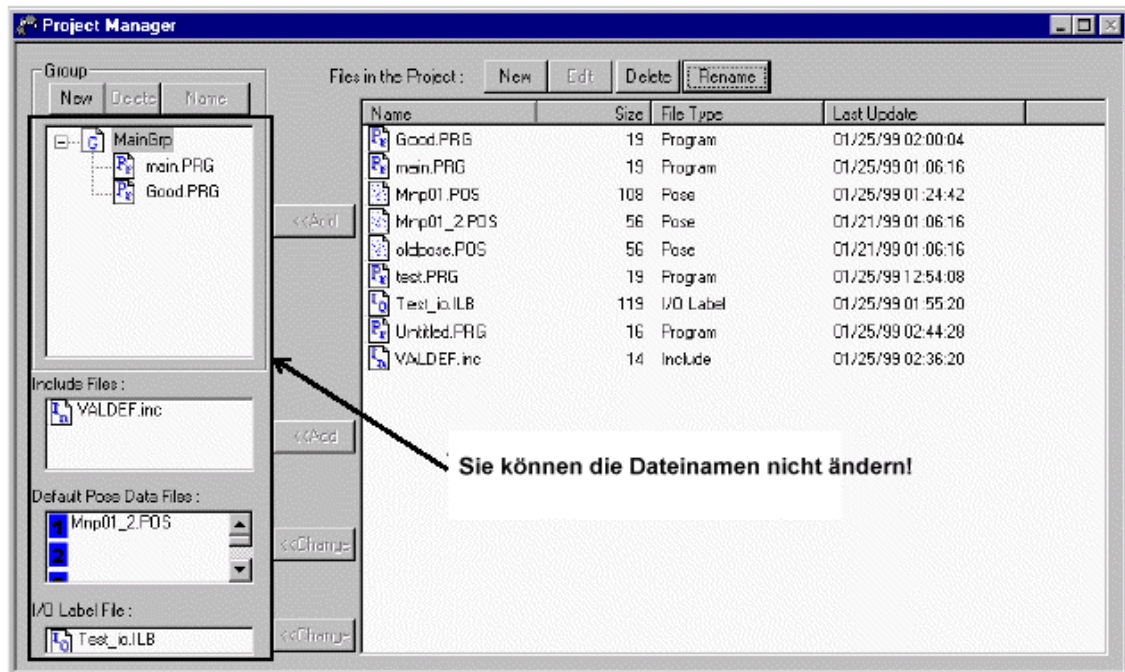


Die Gruppenbox [Include File] wird geleert. In diesem Zustand kann keine Include-Datei verwendet werden.

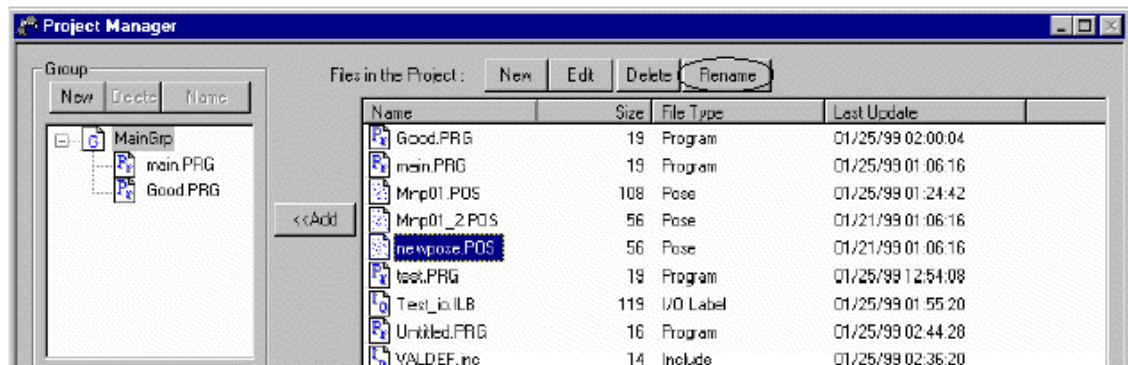


#### 4.4.6 Umbenennen einer Datei

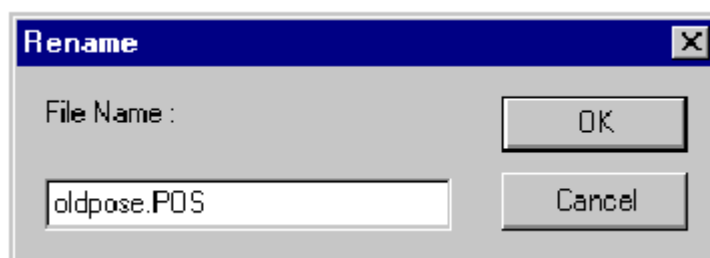
Solange sich die Dateien in den Gruppenboxen auf der linken Seite des Fensters befinden, können sie nicht umbenannt werden. Um sie umzubenennen, müssen sie zuvor freigegeben werden. Fügen Sie die Dateien nach dem Umbenennen wieder den entsprechenden Gruppenboxen zu.



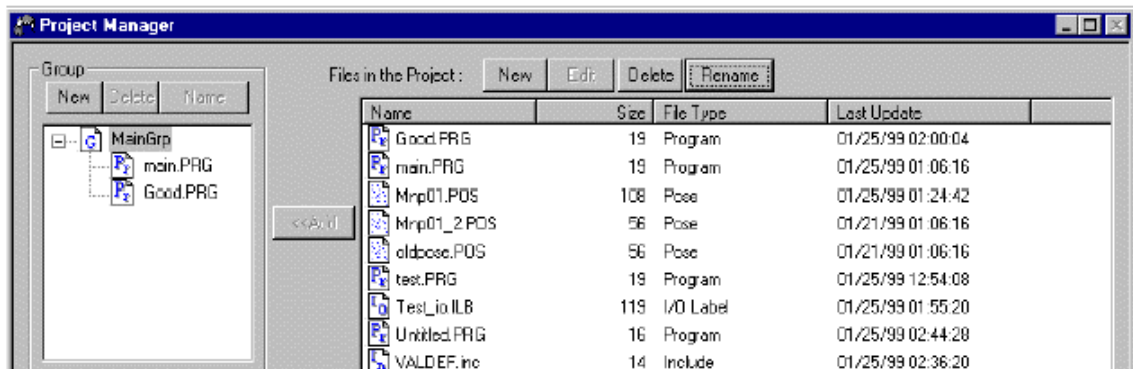
Klicken Sie auf die Datei, die Sie umbenennen möchten und dann auf <Rename> (Umbenennen).



Der folgende Dialog erscheint. Geben Sie in dem Textfeld (File name) einen neuen Namen ein, und klicken Sie dann auf <OK>.



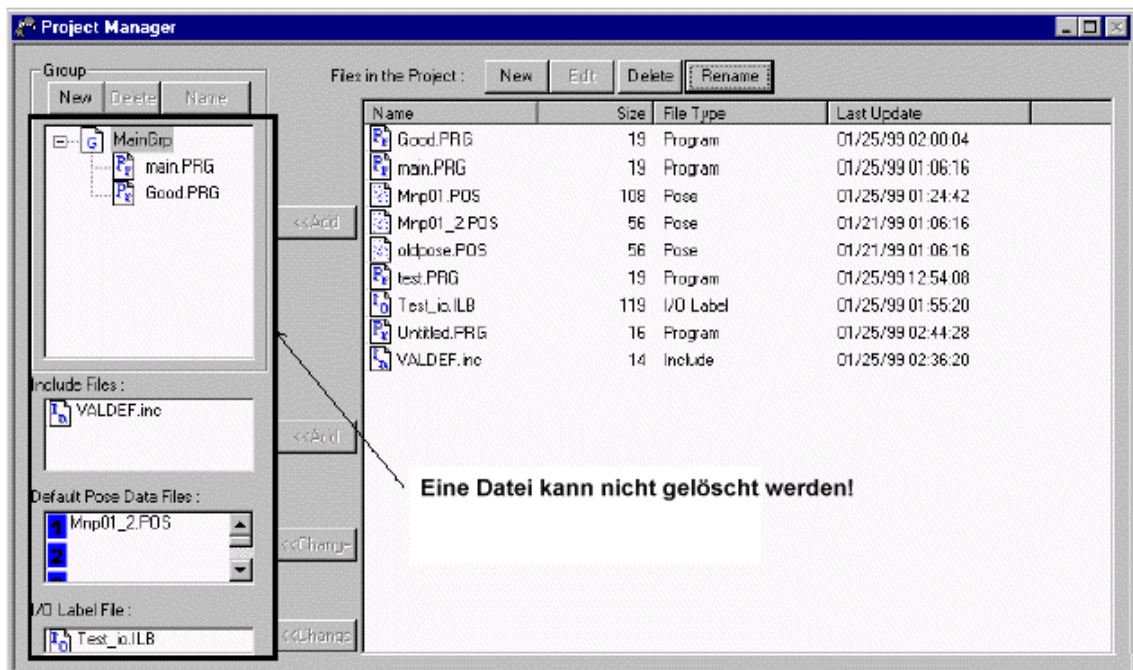
Der Dateiname wurde geändert.



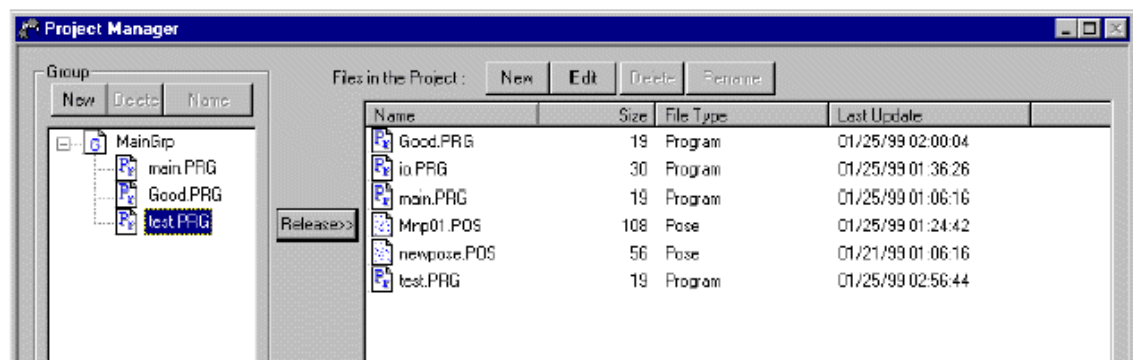
Die Architektur der Dateinamen unter SPEL 95 entspricht der von Windows 95. Es werden keine Fälle unterschieden.

#### 4.4.7 Löschen von Dateien

Solange sich die Dateien in den Gruppenboxen auf der linken Seite des Fensters befinden, können sie nicht gelöscht werden. Um sie zu löschen, müssen sie zuvor freigegeben werden.

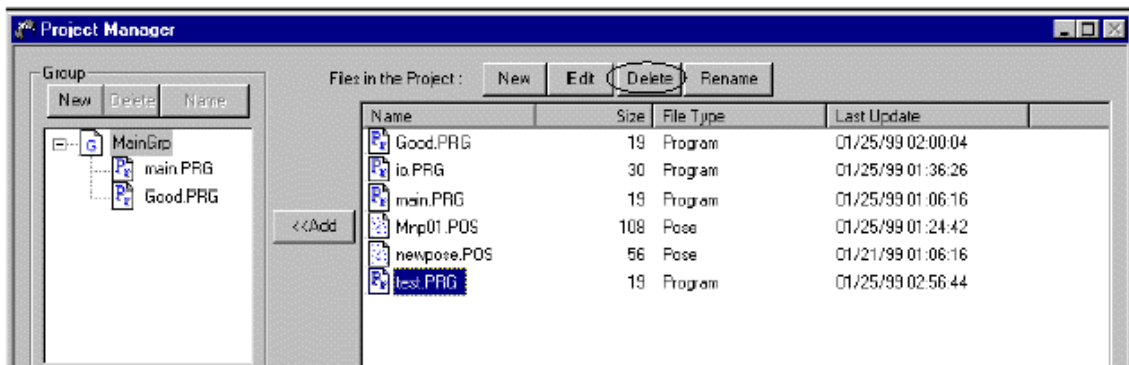


Das folgende Beispiel zeigt, wie Sie eine in einer Gruppe registrierte Programmdatei löschen können. Klicken Sie in der Gruppe auf die Datei, die Sie löschen möchten, und klicken Sie dann auf <Release> (Freigeben).

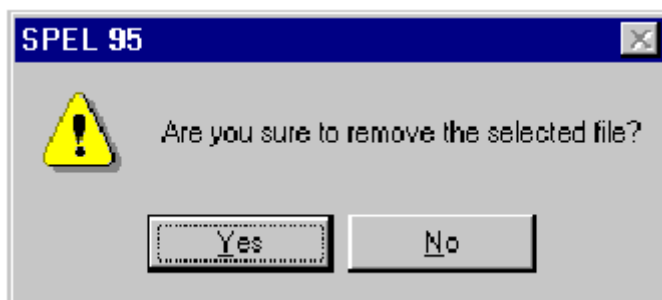


Die Datei wurde freigegeben.

Klicken Sie auf und markieren Sie dann in der Liste [Files in the Project] die Datei, die Sie löschen wollen, und klicken Sie dann auf <Delete> (Löschen).



Der folgende Dialog erscheint. Klicken Sie auf <Yes>, um die Datei endgültig zu löschen.



Die Datei wurde gelöscht.

## 4.5 Gruppenverwaltung

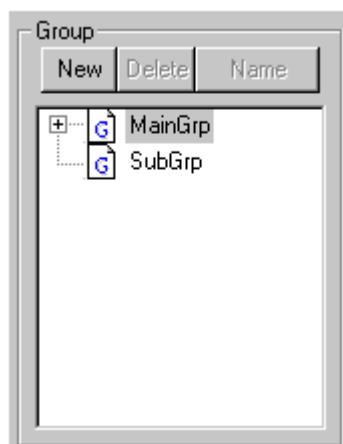
### 4.5.1 Gruppenstruktur

Erst, wenn Programmdateien einer Gruppe angehören, werden Sie aktiv und nutzbar. Es kann mehr als ein Programm in einer Gruppe registriert werden. Durch das Registrieren einer Anfangs-Positionsdaten-Datei können Sie unabhängige Positionsdaten für jede Gruppe verwenden.

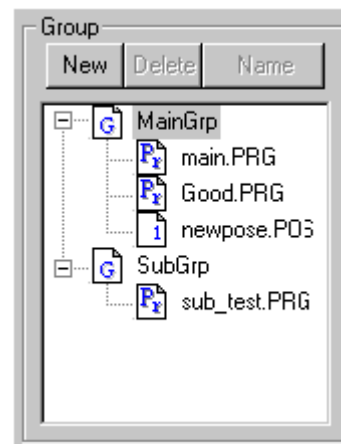
#### ▪ Wie Sie die Gruppenboxen lesen

Die Gruppenboxen zeigen Verzeichnisse in einer Baumstruktur. Wenn Sie auf <+> oder <-> (links angeordnet) klicken, wechselt die Ansicht. So können Sie entweder nur die einzelnen Gruppen oder auch die Gruppen und die darin enthaltenen Dateien anzeigen lassen.

Es werden nur Gruppen angezeigt.



Es werden Gruppen und Dateien angezeigt.



Werden Gruppen und Dateien angezeigt, erscheinen die Dateinamen der Programmdateien und der Anfangs-Positionsdaten-Dateien, die in der Gruppe registriert sind, eine Ebene tiefer.

Die Endung der Programmdateien lautet „.PRG“.

Die Endung der Anfangs-Positionsdaten-Dateien lautet „.POS“.

#### ▪ Erzeugen neuer Gruppen und Gruppennamen

Klicken Sie auf <New> (Neu) in der Gruppenbox.

Geben Sie den neuen Namen ein, den Sie erzeugen wollen.

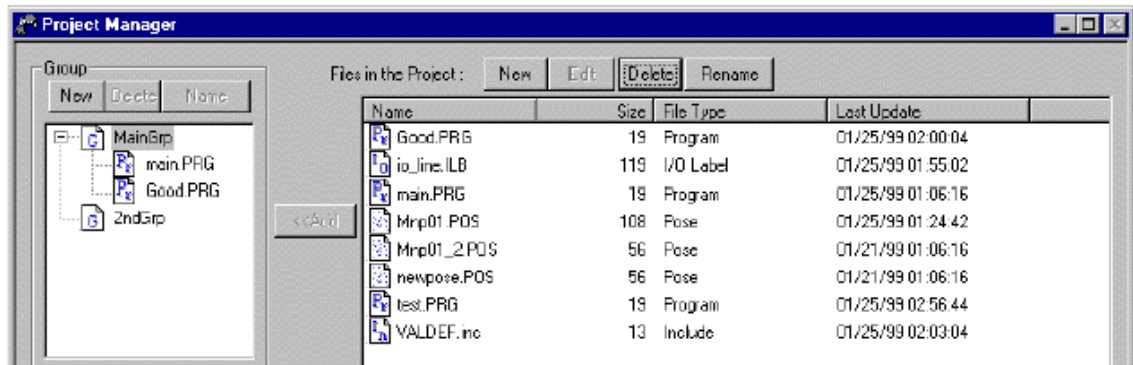
Die Gruppennamen unterliegen folgenden Beschränkungen:

- Die Länge des Namens darf 15 Zeichen (inklusive Erweiterung) nicht überschreiten. Ein voll breites Zeichen wird als Zwei-Byte-Zeichen gezählt.
- Reservierte Worte (inkl. Befehle, Anweisungen und Funktionen) dürfen nicht verwendet werden.
- Gruppennamen dürfen nicht kasus-sensitiv sein.



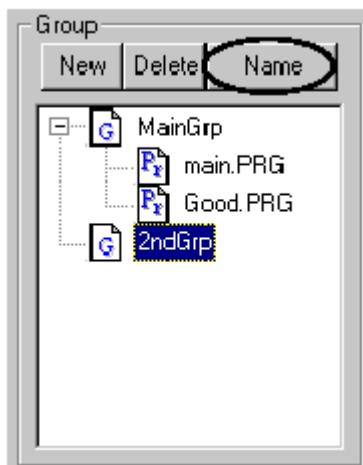


Die neue Gruppe wurde erzeugt.

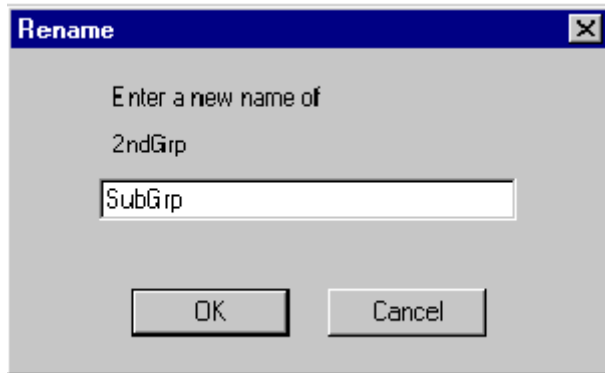


#### ▪ Umbenennen einer Gruppe

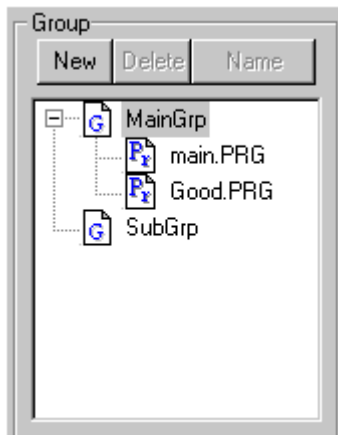
Wählen Sie die Gruppe, deren Namen Sie ändern möchten, aus der Liste der Gruppenbox. Klicken sie dann in der Gruppenbox auf <Name>.



Der nachstehende Dialog erscheint. Geben Sie in das Textfeld einen neuen Namen ein, und klicken Sie dann auf <OK>.

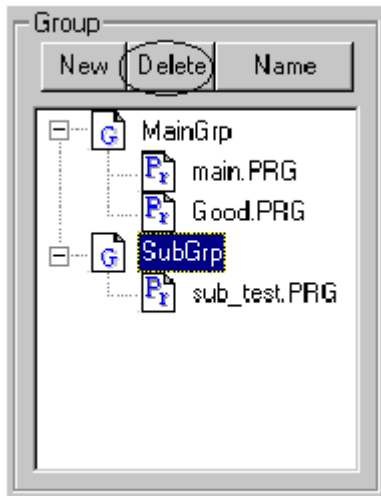


Der Name der Gruppe wurde geändert



### ▪ Löschen von Gruppen

Wählen Sie aus der Gruppenbox die Gruppe, die Sie löschen möchten, und klicken Sie dann auf <Delete> (Löschen) in der Gruppenbox.



Der nachstehende Dialog erscheint. Klicken Sie auf <OK>, um die Gruppe zu löschen.



Wenn in der Gruppe Dateien existieren, müssen diese zuerst freigegeben werden. Erst wenn die Gruppe leer ist, lässt sie sich löschen.



## 4.5.2 Ändern von Gruppen

Um Gruppen zu ändern, wählen Sie eine der drei nachfolgenden Methoden:

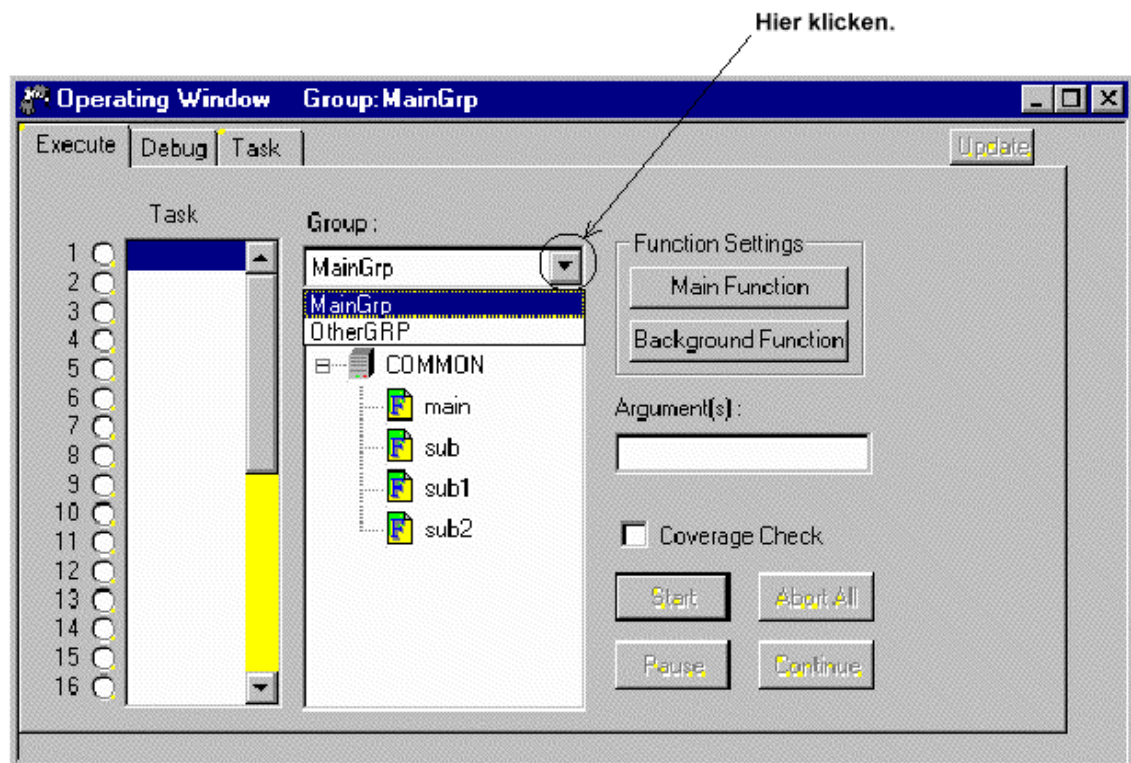
- Debug-Menü
- Operator-Menü
- Ändern mittels der Remote-Funktion

Wenn Sie Gruppen ändern, wird automatisch die Anfangs-Positionsdaten-Datei in den Speicher geladen. Falls keine Anfangs-Positionsdaten-Datei existiert, wird automatisch die Standard-Positionsdaten-Datei in den Speicher geladen.

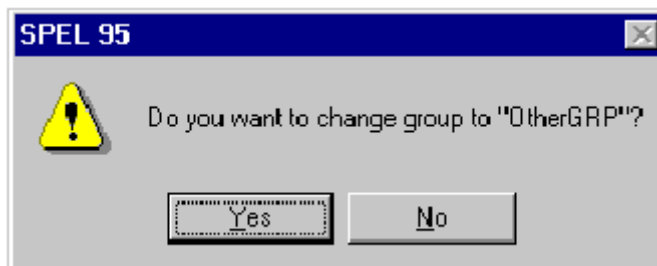
### ▪ Änderungen im Debug-Menü

Klicken Sie auf <Debug> im [Operating Window].

Gruppen können in der Gruppenauswahlliste geändert werden.



Der nachstehende Dialog erscheint.



Klicken Sie auf <Yes>, und die Gruppe wird gewechselt.

Weitere Informationen erhalten Sie in Kapitel 6 „Fehlerbeseitigung“.

### ▪ Änderungen im Operator-Menü

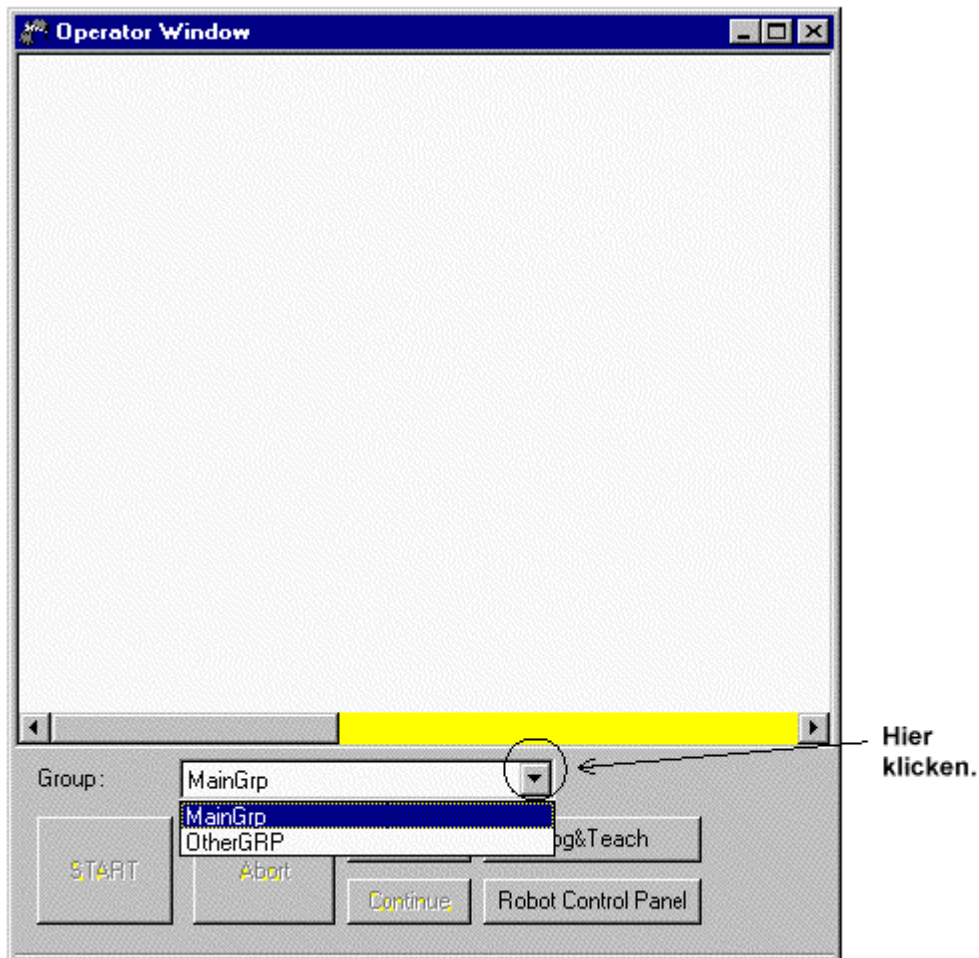
Klicken Sie auf <Operator Windows> in der Werkzeugleiste.

Gruppen können in der Gruppenauswahlliste geändert werden.

Der gleiche Bestätigungsdialog wie im Debug-Menü erscheint.

Um Funktionen vom Operator-Fenster ausführen zu können, muss im Vorfeld eine Hauptfunktion festgelegt worden sein.

Wie Sie diese Hauptfunktion festlegen, erfahren Sie in Kapitel 6 „Fehlerbeseitigung“.



### ▪ Änderungen mittels der Remote-Funktion

Im Auto-Modus ermöglicht die Remote-Funktion Ihnen den Wechsel zwischen Gruppen. Dann kann die Hauptfunktion ausgeführt werden.

Um den Remote-Eingang zu aktivieren, muss „Remote“ als Konsole im [Setup]-[Robot Controller Settings – [General Panel] (Einstellung - Robotersteuerung - Allgemeines Fenster) eingestellt sein.

Legen Sie Gruppennummern zwischen 0 und 3 in 4 Bits fest. Durch Eingabe des START-Befehls können Sie Gruppen ändern und diese starten.

Um dies zu tun, müssen im Vorfeld Gruppennummern für einzelne Gruppen eingestellt werden.

Nähere Informationen erhalten Sie in Kapitel 4.6.4 „Einrichten von Projekten – Einrichten von Gruppennummern“.

## 4.6 Editieren von Projekten

Im [Project Manager] lassen sich Gruppen und verschiedene Dateitypen bearbeiten.

### 4.6.1 Editieren von Dateien

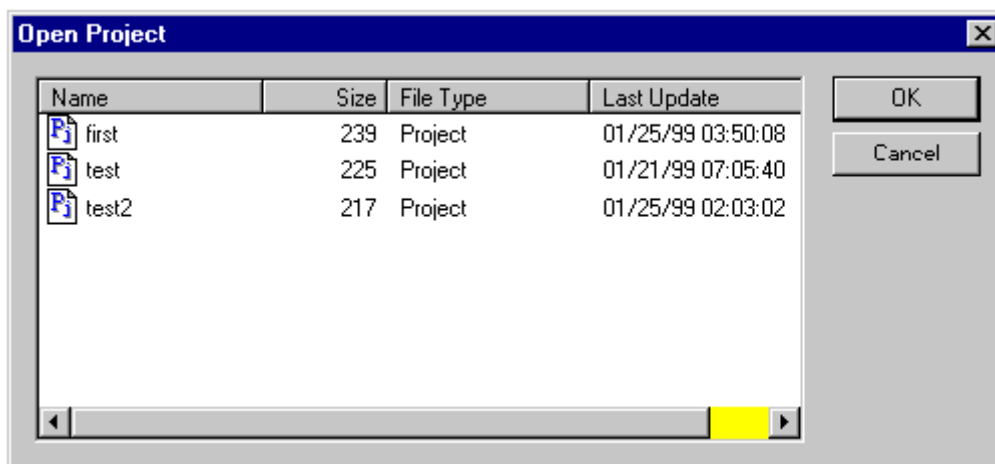
Wählen Sie die zu bearbeitende Datei aus der Liste [Files in the Project] aus, und klicken Sie auf <Edit> (Editieren/ Bearbeiten). Oder Doppelklicken Sie auf die entsprechende Datei.

Bewegen Sie den Cursor zu den gewünschten Daten. Editieren Sie dann die Daten, und speichern Sie ab.

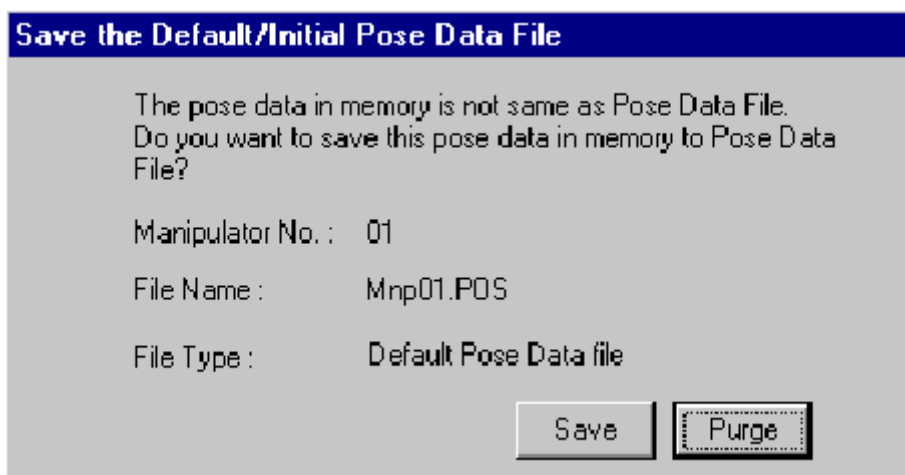
### 4.6.2 Öffnen von Projekten

Führen sie den Befehl [Project – Open] (Projekt – Öffnen) aus.

Wählen Sie das zu öffnende Projekt aus, und klicken Sie auf <OK>.

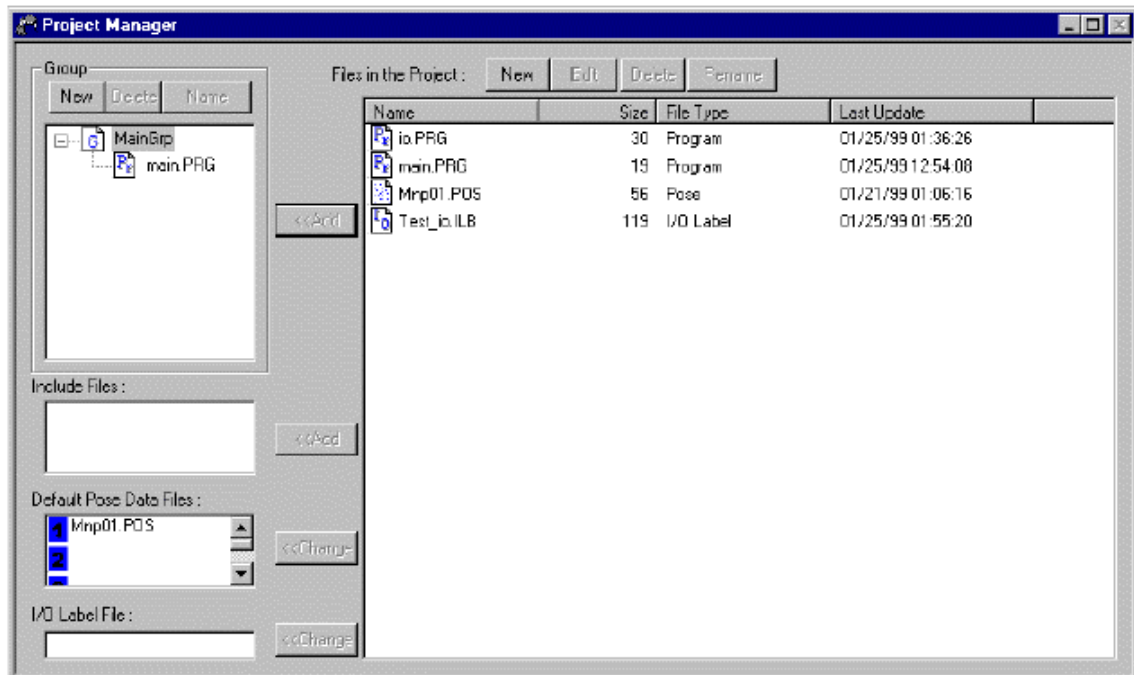


Stimmen die Daten der Standard-Positionsdaten-Datei des existierenden Projekts und die Daten im Speicher nicht überein, erscheint die folgende Meldung.



Wenn Sie die im Speicher befindlichen Daten als Standard-Positionsdaten speichern wollen, klicken Sie auf <Save> (Speichern) - ansonsten auf <Purge> (Verwerfen).

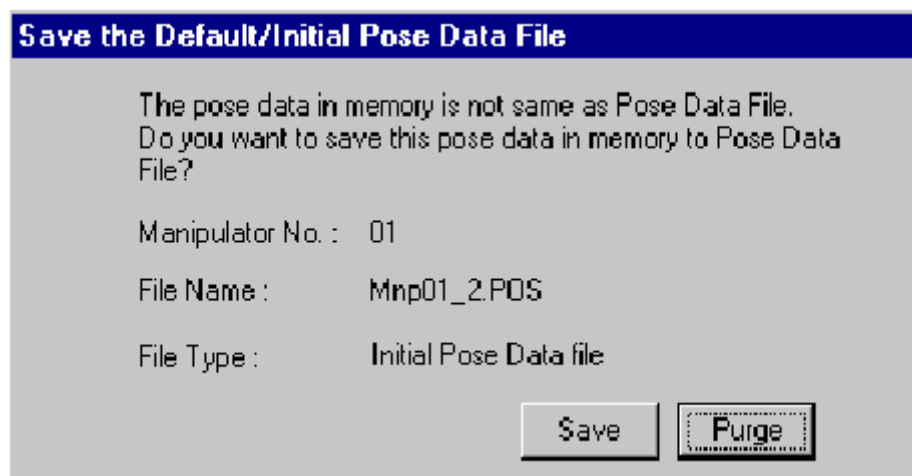
Die bisher entstandenen Projekte werden gespeichert, und ein neues Projekt wird geöffnet.



#### 4.6.3 Schließen von Projekten

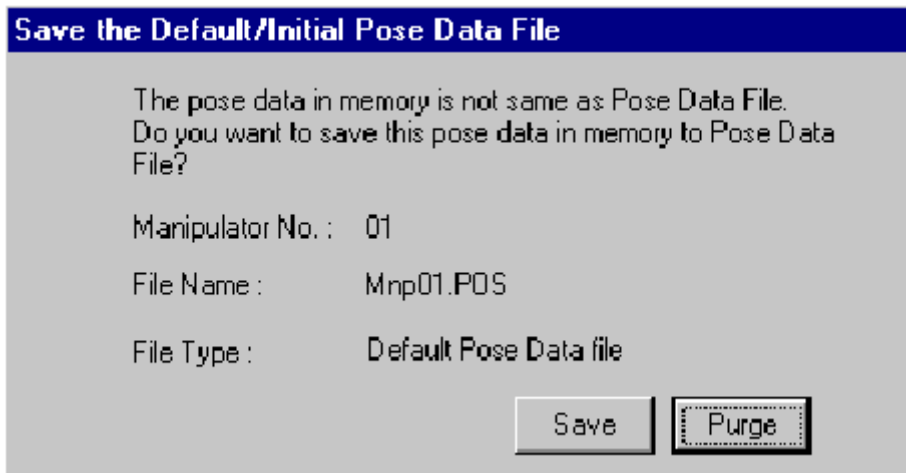
Führen Sie den Befehl [Project] – [Close] (Projekt – Schließen) aus.

Stimmen die Daten der Anfangs-Positionsdaten-Datei des existierenden Projekts und die Daten im Speicher nicht überein, erscheint die folgende Meldung.



Wenn Sie die im Speicher befindlichen Daten als Anfangs-Positionsdaten-Datei speichern wollen, klicken Sie auf <Save> (Speichern) - ansonsten auf <Purge> (Verwerfen).

Stimmen die Daten der Standard-Positionsdaten-Datei des existierenden Projekts und die Daten im Speicher nicht überein, erscheint die folgende Meldung.



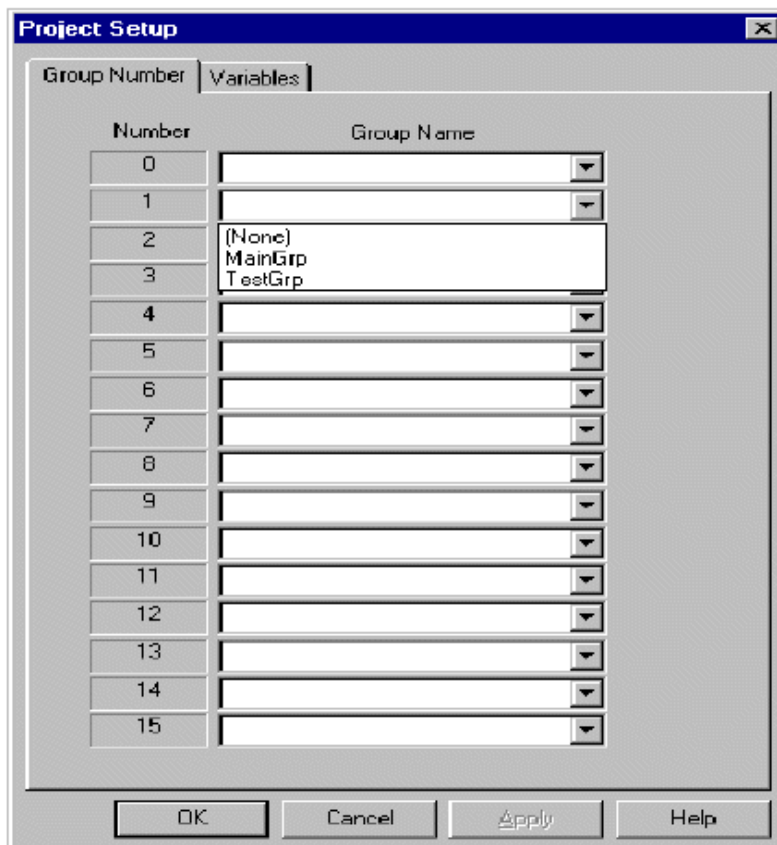
Wenn Sie die im Speicher befindlichen Daten als Standard-Positionsdaten speichern wollen, klicken Sie auf <Save> (Speichern) - ansonsten auf <Purge> (Verwerfen). Der Bildschirm verschwindet um anzuzeigen, dass keine Projekte ausgewählt sind. Im AUTO-Modus werden die Positionsdaten im Speicher gelöscht, ohne dass eine Meldung erscheint.

#### 4.6.4 Einrichten von Projekten

- **Einrichten der Gruppennummern**

Im AUTO-Modus ermöglicht Ihnen die Remote-Funktion Gruppennummern einzurichten, die beim Hochfahren der Anwendungen automatisch ausgewählt werden. Siehe auch Kapitel 7 „Remote-Einstellungen – Kommunikation mit externen Geräten“.

Führen Sie den Befehl [Project] [Group Number] (Projekt – Gruppennummer) aus.



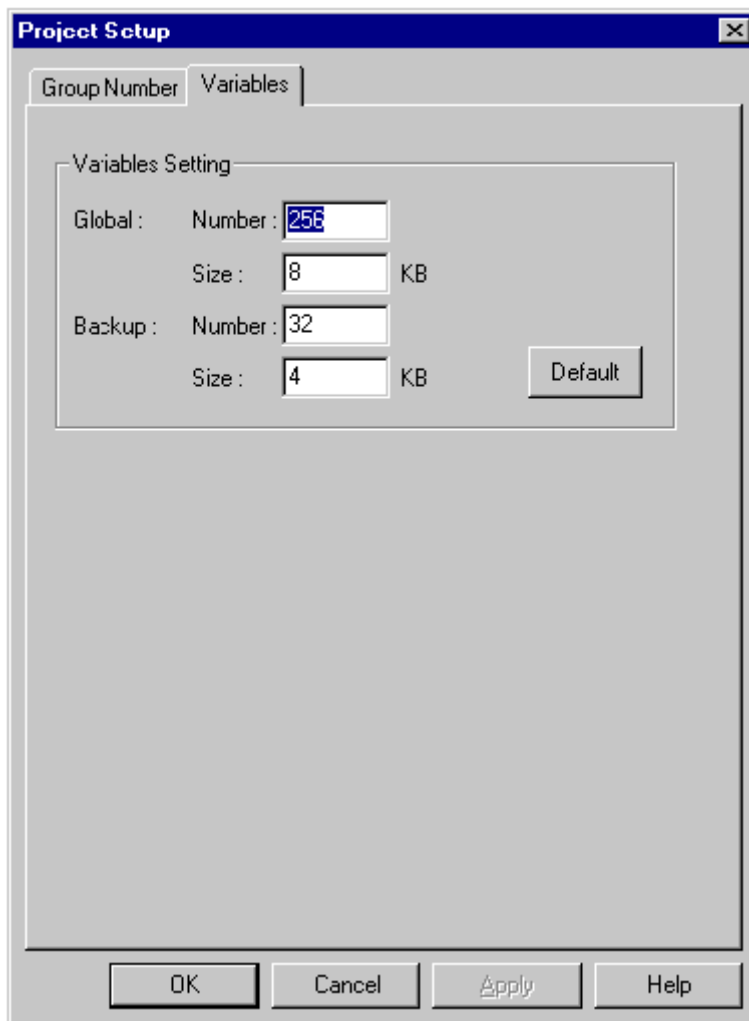


Wählen Sie einen Gruppennamen aus der Auswahlliste aus, oder geben Sie einen Namen ein. Wenn Sie alle benötigten Namen eingegeben oder ausgewählt haben, klicken Sie auf <OK>.

#### ▪ Einrichten von Variablen

Für globale und Backup-Variablen können Sie Nummern und deren entsprechende Speicherkapazität vergeben.

Führen Sie den Befehl [Project] [Group Number] (Projekt – Gruppennummer) aus. Klicken Sie dann auf [Variables] (Variablen).



Geben Sie die Werte in den entsprechenden Textfeldern ein. Wenn Sie auf <Default> (Standard) klicken, erscheinen die Standardwerte für die Variablen.

Nachfolgende Tabelle beinhaltet die Standard- und die maximalen Werte der Variablen:

		Standard	Maximum
Globale Variablen	Anzahl	256	32767
	Größe	8 KB	4096 KB
Backup-Variablen	Anzahl	32	32767
	Größe	4 KB	4096 KB

Nach vollständiger Eingabe der Daten klicken Sie zur Bestätigung auf <OK>. Zu diesem Zeitpunkt werden alle vorhandenen globalen und Backup-Variablen in allen Projekten gelöscht. Folglich müssen Sie anschließend das Wiederregistrierungsprogramm ausführen!

## 4.7 Übertragen von Projekten

### 4.7.1 Projekte sichern

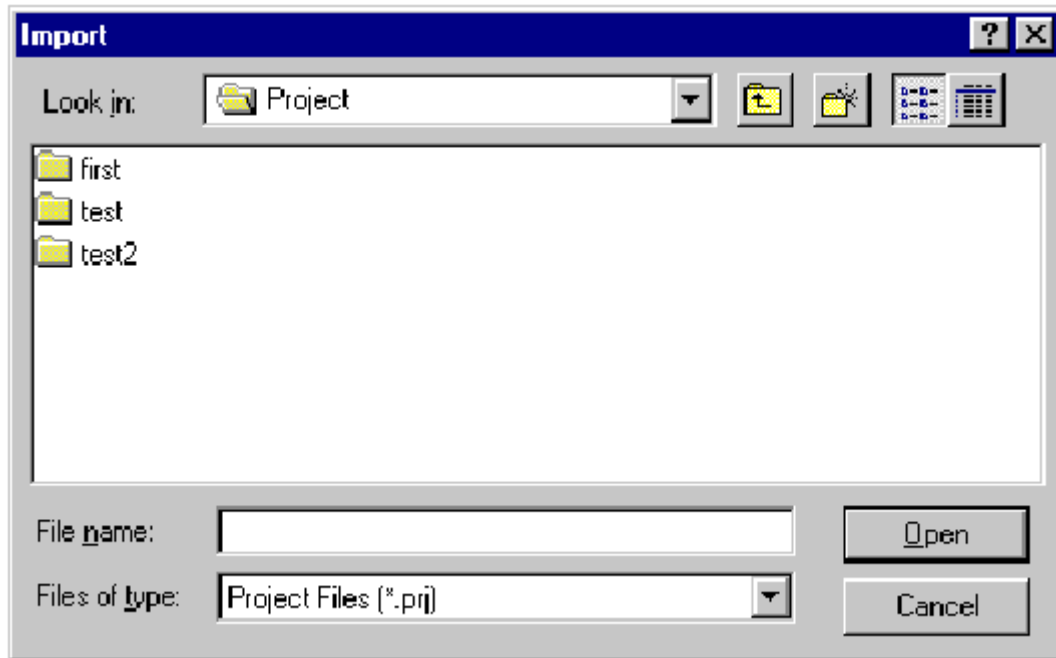
Um eine Sicherungskopie des Projektes zu erstellen, verwenden Sie den Befehl [Project] – [Transfer] – [Export] (Projekt – Übertragung - Export).



Sie können eine Sicherungskopie auf Diskette oder in einem anderen Verzeichnis auf der Festplatte erstellen. Geben Sie entsprechend das Laufwerk, das Verzeichnis und einen Projektnamen ein.

### 4.7.2 Projekte laden

Um ein Projekt zu laden, verwenden Sie den Befehl [Project] – [Transfer] – [Import] (Projekt –Übertragung - Import).



Wählen Sie die zu ladende Datei aus, und klicken Sie auf <Open> (Öffnen). Die nachfolgende Meldung erscheint, wenn die Datei geladen wurde.

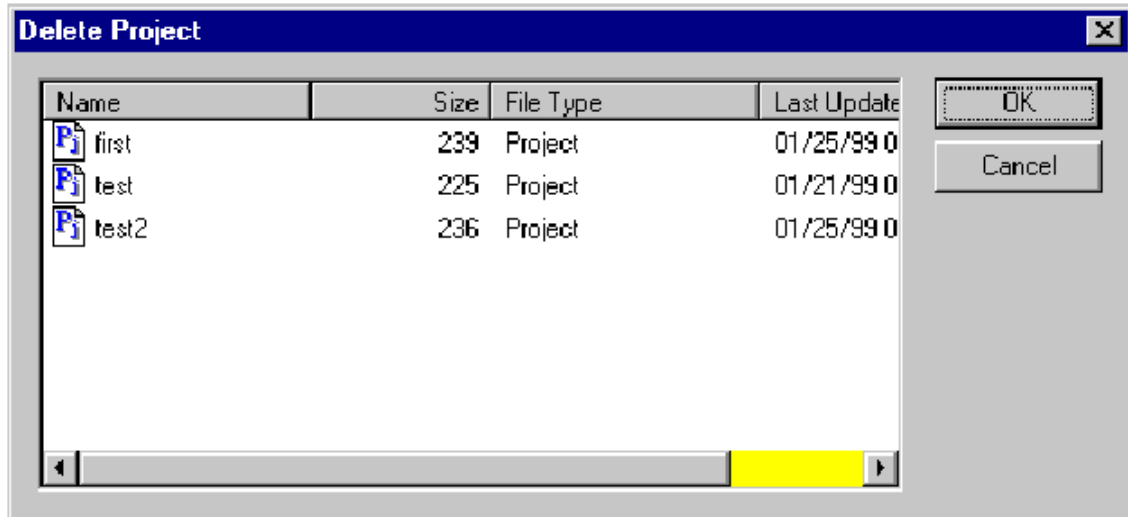


Wenn Sie das geladene Projekt als aktuelles Projekt laden wollen, klicken Sie auf <Yes>. Die geladene Datei wird geöffnet.

Klicken Sie auf <OK>, wenn Sie es in den Projektordner speichern wollen, ohne es sofort zu öffnen. Um die Projekte im Verzeichnis zu öffnen, verwenden Sie den Dialog [Open Project] (Projekt öffnen), der erscheint, wenn Sie den Befehl [Project] – [Open] ausgeführt haben.

## 4.8 Löschen von Projekten

Um ein Projekt zu löschen, verwenden Sie den Befehl [Project] – [Delete] (Projekt - Löschen). Wählen Sie das zu löschende Projekt, und klicken Sie auf <OK>.



Das aktuell geöffnete Projekt kann nicht gelöscht werden!

# Kapitel 5

## Editor

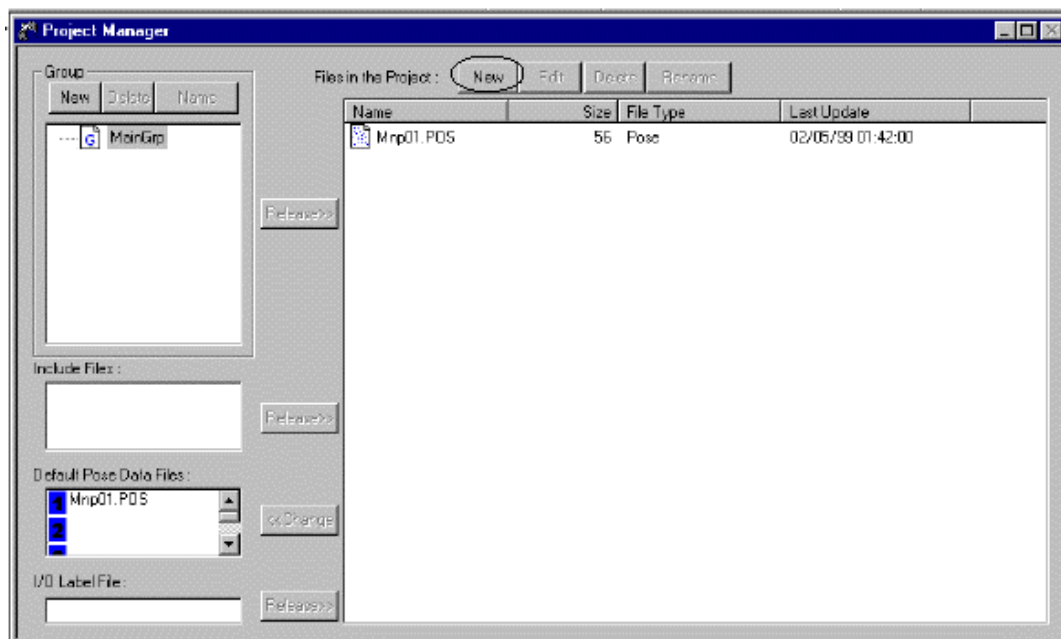
### 5.1 Erzeugen von Programmen und Include-Dateien

#### 5.1.1 Erzeugen neuer Dateien

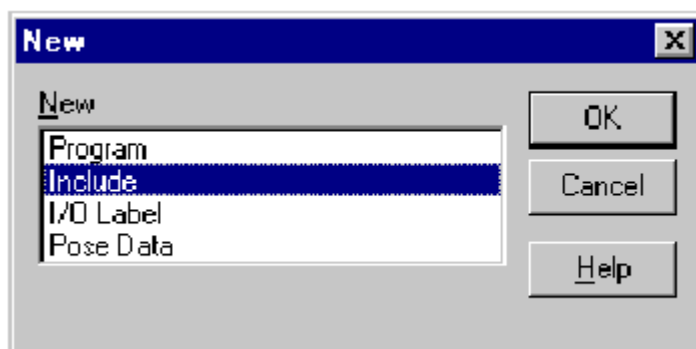
Klicken Sie auf <Programming Pane> (Programmiermenü) in der Werkzeugleiste. Klicken Sie auf <New> (Neu) in der Liste [Files in the Project].



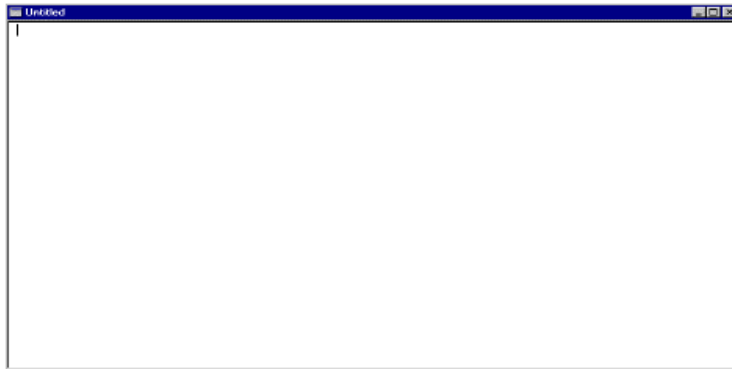
<Programming  
Pane>  
(Programmier-  
menü)



Um eine Programmdatei zu erzeugen, wählen Sie „Program“ (Programm), und um eine Include-Datei zu erzeugen, klicken Sie auf „Include“ und dann auf <Yes> (Ja).



Das Fenster zum Editieren von Programmen erscheint. Mit Hilfe des Cursors in der linken oberen Ecke des Fensters können Sie Ihr Programm eingeben. Die maximale Programmgröße beträgt 64 KB.

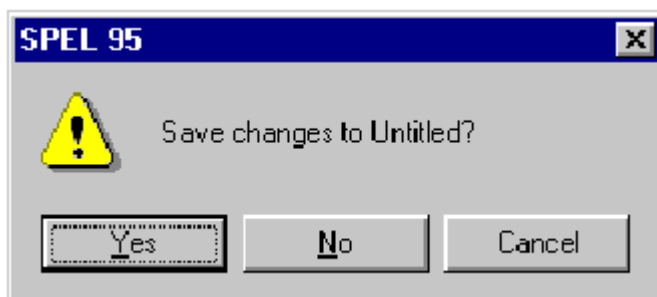


#### ▪ Speichern des Programms

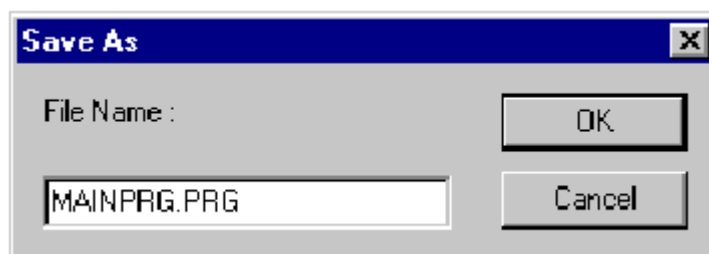
Klicken Sie auf [File] – [Save as] (Datei – Speichern unter), oder klicken Sie auf das Kreuz in der rechten oberen Ecke des Programmierfensters.



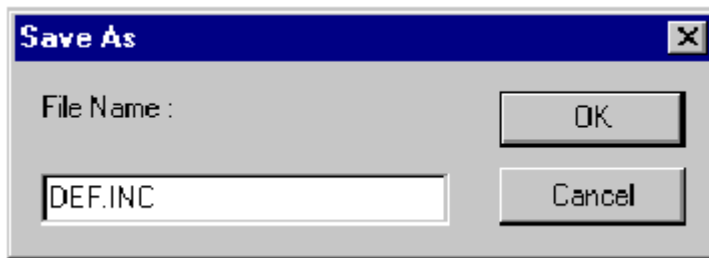
Die nachfolgende Abfrage erscheint. Wenn Sie die Datei speichern, wollen klicken Sie auf <Yes>.



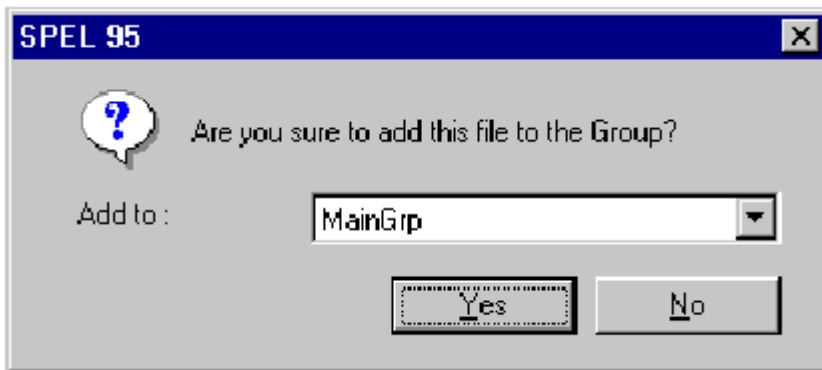
Geben Sie in der erscheinenden Meldung einen Namen für die Datei ein. In dem folgenden Beispiel vergeben wir „MAINPRG.PRG“ als Dateinamen. Ein Programmdateiname (.PRG-Datei) kann 31 oder weniger normal breite Zeichen inklusive Endung enthalten.



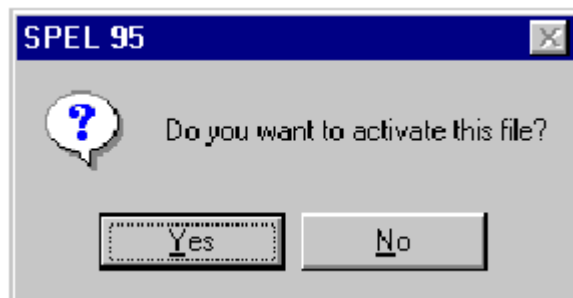
(Dialogfenster zum Speichern von Include-Dateien) Es gibt Beschränkungen bei der Anzahl der Zeichen für einen Include-Dateinamen. Nähere Informationen entnehmen Sie bitte Kapitel 4.2.6 „Einschränkungen bei Dateinamen“.



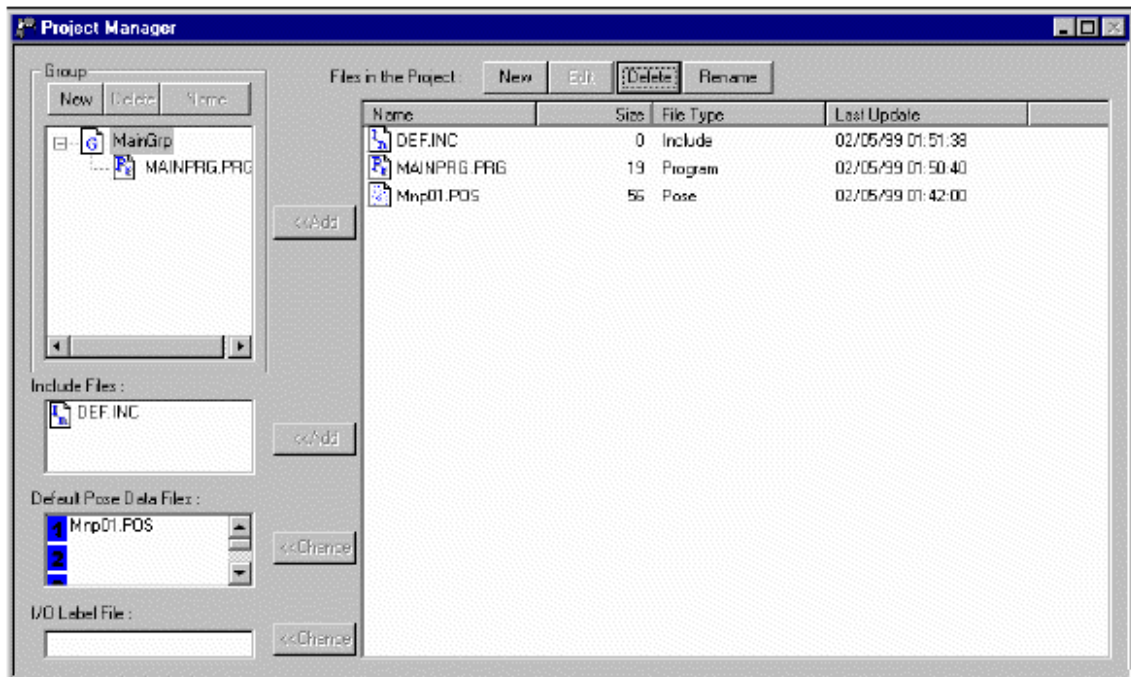
Sie werden dann gefragt, ob Sie die Datei einer Gruppe zuordnen möchten. Klicken Sie auf <Yes>.



Dann wird abgefragt, ob die Datei aktiviert werden soll. Klicken Sie auf <Yes>.

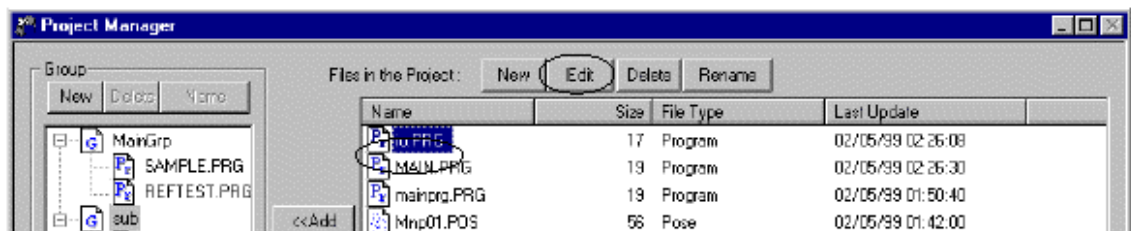


Die neuen Dateien (in diesem Fall MAINPRG.PRG und DEF.INC) wurden im Projekt registriert. Die Datei MAINPRG wird auch in der Gruppenauswahl registriert, damit sie ausführbar wird, und die Datei DEF.INC wird in den Include-Dateien registriert, damit man sich auf sie beziehen kann.



#### ▪ Bearbeiten existierender Dateien

Um eine existierende Datei bearbeiten zu können, wählen Sie die zu bearbeitende Datei aus, und klicken Sie dann auf <Edit> (Bearbeiten). Oder Doppelklicken Sie auf die zu bearbeitende Datei.





## 5.1.2 Bearbeitungsfunktionen

Die Funktion vom Bearbeiten von Texten arbeitet generell basierend auf Windows 95. Die folgenden Funktionen werden nachfolgend beschrieben:

Einfügen  
 Löschen  
 Aufheben  
 Wiederherstellen  
 Ausschneiden  
 Kopieren  
 Einfügen  
 Finden  
 Ersetzen


### ▪ Einfügen


Wird benötigt, um Zeichen in der Mitte von Texten einzufügen. Bewegen Sie den Cursor zur linken Seite der Stelle, an der Sie einfügen wollen. Wenn Sie dann das Zeichen eingeben, wird es an der entsprechenden Stelle eingefügt.

### ▪ Löschen

Benutzen Sie diese Funktion, um nicht benötigte Zeichen zu löschen. Zum Löschen von Zeichen gibt es mehrere Methoden:

- Backspace (Rücktaste)

Bewegen Sie den Cursor zur rechten Seite des Zeichens, das Sie löschen möchten. Wenn Sie jetzt die Rücktaste  drücken, wird jeweils ein Zeichen links vom Cursor gelöscht.

FUNCTION main| ← Bewegen Sie den Cursor zur rechten Seite des zu löschenden Zeichens. Drücken Sie dann die Rücktaste .

↓

FUNCTION mai|

- Entfernen-Taste (Entf--Taste)

Bewegen Sie den Cursor zur linken Seite des Zeichens, das Sie löschen möchten. Wenn Sie jetzt die Entfernen-Taste drücken, wird jeweils ein Zeichen rechts vom Cursor gelöscht.

FUNCTION |main ← Bewegen Sie den Cursor zur linken Seite des zu löschenden Zeichens. Drücken Sie dann die Entf-Taste.

↓

FUNCTION |ain

Sie können auch einen Textbereich markieren, der gelöscht werden soll. Wenn Sie dann die Entfernen-Taste drücken, wird der markierte Bereich gelöscht.

FUN~~CTION~~ main ← Markieren Sie den zu löschenden Bereich. Drücken Sie dann die Entf-Taste.

↓

FUON main

- **Aufheben (Undo)**

Benutzen Sie diese Funktion, um die letzte Aktion rückgängig zu machen. Klicken Sie in der Menüleiste auf [Bearbeiten] und dann auf [Undo]. Sie können auch die Tastenkombination <Strg>+<Z> verwenden.

- **Ausschneiden**

Benutzen Sie diese Funktion, um mit der Maus markierte Daten auszuschneiden. Die ausgeschnittenen Daten werden temporär in der Zwischenablage gelagert. Um sie auszuschneiden, klicken Sie auf [Edit] und dann auf [Cut]. Sie können auch die Tastenkombination <Strg>+<X> verwenden.



<Cut>  
(Ausschneiden)

```

MAINPRG.PRG
FUNCTION main
DIM dat1,dat2 AS INTEGER
DIM rdat AS REAL
CLS
at1 = 123; dat2 = 456; rdat = 123.456789
PRINT "The value before call sub"
PRINT "dat1=",dat1, "dat2=",dat2,TAB$(1), "rdat=",rdat
CALL subf(dat1,BYREF dat2,BYREF rdat)
PRINT "The value after call sub"
PRINT "dat1=",dat1,TAB$(1), "dat2=",dat2,TAB$(1), "rdat=",rdat
FEND

FUNCTION subf(prn1,BYREF prn2 AS INTEGER,BYREF prn3)
PRINT "The value inside the sub function"
PRINT "prn1=",prn1,TAB$(1), "prn2=",prn2,TAB$(1), "prn3=",prn3
prn1 = 321; prn2 = 654; prn3 = 987.654321
PRINT "The update value inside the sub funtion"
PRINT "prn1=",prn1,TAB$(1), "prn2=",prn2,TAB$(1), "prn3=",prn3
FEND

```

Die markierten Daten werden entfernt, stehen aber in der Zwischenablage zur Verfügung.:

```

MAINPRG.PRG
FUNCTION main
DIM dat1,dat2 AS INTEGER
DIM rdat AS REAL
CLS
at1 = 123; dat2 = 456; rdat = 123.456789
PRINT "The value before call sub"
PRINT "dat1=",dat1, "dat2=",dat2,TAB$(1), "rdat=",rdat
CALL subf(dat1,BYREF dat2,BYREF rdat)
FEND ← Der markierte Teil wurde entfernt.

FUNCTION subf(prn1,BYREF prn2 AS INTEGER,BYREF prn3)
PRINT "The value inside the sub function"
PRINT "prn1=",prn1,TAB$(1), "prn2=",prn2,TAB$(1), "prn3=",prn3
prn1 = 321; prn2 = 654; prn3 = 987.654321
PRINT "The update value inside the sub funtion"
PRINT "prn1=",prn1,TAB$(1), "prn2=",prn2,TAB$(1), "prn3=",prn3
FEND

```

### ▪ Kopieren

Benutzen Sie diese Funktion, um mit der Maus markierte Daten in die Zwischenablage zu kopieren. Die kopierten Daten bleiben in ihrer ursprünglichen Form erhalten, können aber aus dem Zwischenspeicher an jede andere Stelle kopiert werden. Um die Daten zu kopieren, klicken Sie auf [Edit] und dann auf [Copy]. Sie können auch die Tastenkombination <Strg>+<C> verwenden.



<Copy>  
(Kopieren)

```

MAINPRG.PRG
FUNCTION main
DIM dat1,dat2 AS INTEGER
DIM rdat AS REAL
CLS
at1 = 123; dat2 = 456; rdat = 123.456789
PRINT "The value before call sub"
PRINT "dat1-",dat1, "dat2-",dat2,TAB$(1), "rdat-",rdat
CALL subf(dat1,BYREF dat2,BYREF rdat)
FEND

FUNCTION subf(prm1,BYREF prm2 AS INTEGER,BYREF prm3)
PRINT "The value inside the sub function"
PRINT "prm1=",prm1,TAB$(1), "prm2=",prm2,TAB$(1), "prm3=",prm3
prm1 = 321; prm2 = 654; prm3 = 987.654321
PRINT "The update value inside the sub funtion"
PRINT "prm1=",prm1,TAB$(1), "prm2=",prm2,TAB$(1), "prm3=",prm3
FEND

```

### ▪ Einfügen

Benutzen Sie diese Funktion, um in der Zwischenablage befindliche Daten an der Position des Cursors einzufügen. Die Daten in der Zwischenablage bleiben solange erhalten, bis Sie neue Daten ausschneiden oder kopieren. Um die Daten einzufügen, klicken Sie auf [Edit] und dann auf [Paste]. Sie können auch die Tastenkombination <Strg>+<V> verwenden.



<Paste>  
(Einfügen)

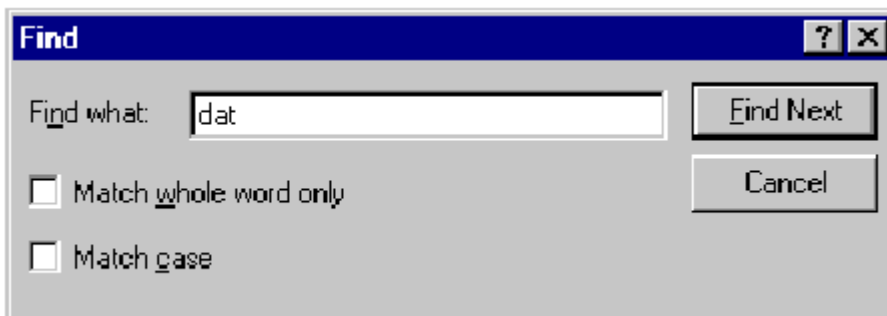
```

MAINPRG.PRG
FUNCTION main
DIM dat1,dat2 AS INTEGER
DIM rdat AS REAL
CLS
at1 = 123; dat2 = 456; rdat = 123.456789
PRINT "The value before call sub"
PRINT "dat1=",dat1, "dat2=",dat2,TAB$(1), "rdat=",rdat
CALL subf(dat1,BYREF dat2,BYREF rdat)
PRINT "The value before call sub"
PRINT "dat1-",dat1, "dat2-",dat2,TAB$(1), "rdat-",rdat
FEND

FUNCTION subf(prn1,BYREF prn2 AS INTEGER,BYREF prn3)
PRINT "The value inside the sub function"
PRINT "prn1=",prn1,TAB$(1), "prn2=",prn2,TAB$(1), "prn3=",prn3
prn1 = 321; prn2 = 654; prn3 = 987.654321
PRINT "The update value inside the sub function"
PRINT "prn1-",prn1,TAB$(1), "prn2-",prn2,TAB$(1), "prn3-",prn3
FEND
  
```

### ▪ Finden

Benutzen Sie diese Funktion, um einzelne Zeichen oder Zeichenkombinationen zu finden. Um die Daten zu suchen, klicken Sie auf [Edit] und dann auf [Find]. Der folgende Dialog erscheint:

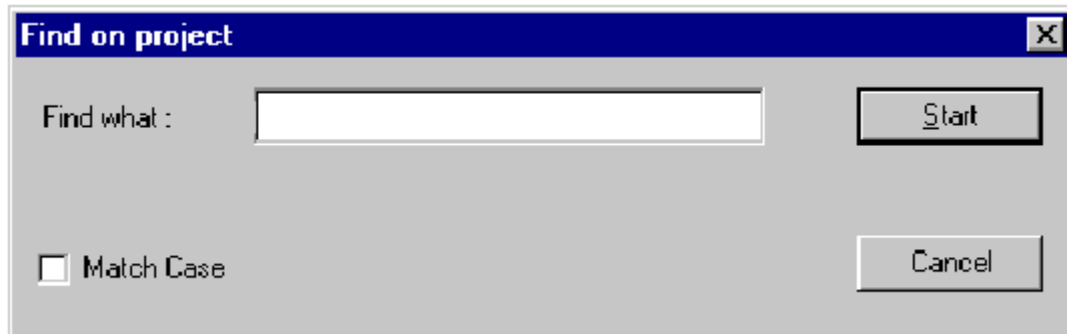


Geben Sie die Zeichenkombination, nach der gesucht werden soll, in das Textfeld ein. Klicken Sie dann auf <Find next> (Nächstes finden). Jetzt wird nach der Zeichenkombination gesucht.

Markieren Sie das Optionsfeld [Match whole word only], wenn nur nach ganzen Worten gesucht werden soll.

Markieren Sie das Optionsfeld [Match case], wenn entsprechend der eingegebenen Groß- und Kleinschreibweise gesucht werden soll.

Sie können die festgelegte Zeichenkombination auch direkt im Projekt suchen. Klicken Sie in der Menüleiste auf [Edit] und dann auf [Find On Project] (Finde im Projekt). Der folgende Dialog erscheint:



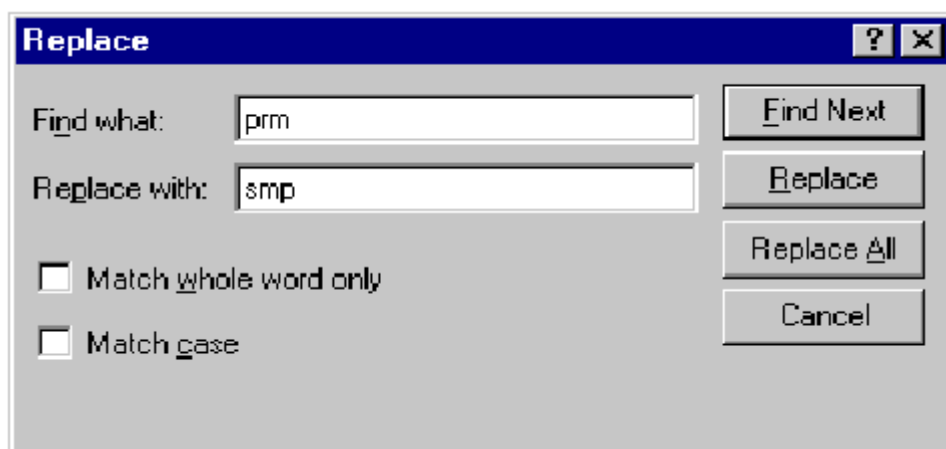
Geben Sie die Zeichenkombination, nach der gesucht werden soll, in das Textfeld ein. Klicken Sie dann auf <Start>. Jetzt wird nach der Zeichenkombination gesucht.

Markieren Sie das Optionsfeld [Match case], wenn entsprechend der eingegebenen Groß- und Kleinschreibweise gesucht werden soll.

Das Suchergebnis wird in einem speziellen Fenster dargestellt. Wenn Sie in diesem Fenster auf das Ergebnis doppelklicken, öffnet sich die Datei, in dem sich das Suchergebnis befindet, und der Cursor bewegt sich zu der Stelle in der Datei, wo das Ergebnis steht.

#### ▪ Ersetzen

Benutzen Sie diese Funktion, um einzelne Zeichen oder Zeichenkombinationen zu finden und durch andere zu ersetzen. Klicken Sie auf [Edit] und dann auf [Replace]. Sie können auch die Tastenkombination <Strg>+<H> verwenden. Der folgende Dialog erscheint:



Geben Sie die Zeichenkombination, nach der gesucht werden soll, in das erste Textfeld ein. Geben Sie dann die Zeichenkombination, durch die ersetzt werden soll, in das zweite Textfeld ein.

In dem obigen Beispiel wird nach „prm“ gesucht, und diese Zeichenkombination wird durch „smp“ ersetzt.

<Find next> (Nächstes finden): Sucht nach der nächsten Zeichenkombination, ohne diese zu ersetzen.

<Replace> (Ersetzen): Ersetzt die gefundene Zeichenkombination und sucht die nächste.

<Replace all> (Alles ersetzen): Ersetzt automatisch alle gefundenen Zeichenkombinationen.

[Match whole word only]: Markieren Sie dieses Optionsfeld, wenn nur nach ganzen Worten gesucht werden soll.

[Match case]: Markieren Sie dieses Optionsfeld, wenn entsprechend der eingegebenen Groß- und Kleinschreibweise gesucht werden soll.

Nachfolgend finden Sie das Ergebnis des zuvor genannten Beispiels. Die Zeichenkombinationen „prm“ in der Funktion subf wurden durch „smp“ ersetzt.

```

MAINPRG.PRG
FUNCTION main
DIM dat1,dat2 AS INTEGER
DIM rdat AS REAL
CLS
at1 = 123; dat2 = 456; rdat = 123.456789
PRINT "The value before call sub"
PRINT "dat1=",dat1, "dat2=",dat2,TAB$(1), "rdat=",rdat
CALL subf(dat1,BYREF dat2,BYREF rdat)
PRINT "The value after call sub"
PRINT "dat1=",dat1, "dat2=",dat2,TAB$(1), "rdat=",rdat
FEND

FUNCTION subf(smp1,BYREF smp2 AS INTEGER,BYREF smp3)
PRINT "The value inside sub function"
PRINT "smp1=",smp1,TAB$(1), "smp2=",smp2,TAB$(1), "smp3=",smp3
smp1 = 321; smp2 = 054; smp3 = 987.054321
PRINT "The value after modify inside sub funtion"
PRINT "smp1=",smp1,TAB$(1), "smp2=",smp2,TAB$(1), "smp3=",smp3
FEND

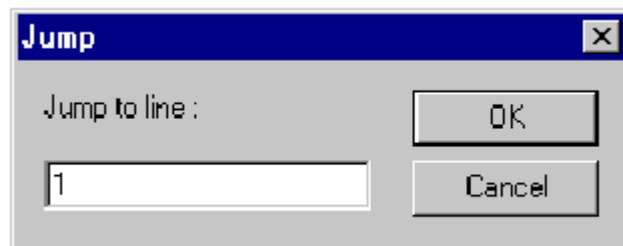
```

#### ▪ Cursorsprung

Sie können den Cursor an beliebige Zeilen oder Stellen im Programm springen lassen. Benutzen Sie folgende Tastenkombinationen zum Springen:

<Strg>+<G>

Geben Sie in den erscheinenden Dialog die gewünschte Zeilennummer ein, zu der gesprungen werden soll, und klicken Sie dann auf <OK>:



<HOME> (Pos1)

Cursor springt zum Zeilenanfang.

<END> (Ende)

Cursor springt zum Zeilenende.

<Strg>+<HOME>

Cursor springt zum Dateianfang.

<Strg>+<END>

Cursor springt zum Dateieende.

### 5.1.3 Pseudobefehle

Das Schreiben von Pseudobefehlen innerhalb eines Programmes kann Vorgangsinhalte vereinfachen und bei der Standardisierung von Programmen hilfreich sein. Darüber hinaus wird es möglich, komplexe Vorgänge in einem Quellprogramm kurz und bündig auszudrücken: Es wird leichter, Programme zu visualisieren und zu korrigieren. Pseudobefehle haben Funktionen, wie zum Beispiel:

#define:	Ersetzt bestimmte Identifizierer (Zeichenketten) und definiert Makroanweisungen.
#ifdef, #ifndef:	Verhindert die Ausführung spezieller Programmteile.
#include:	Verwendet für die Erklärung von #include-Dateien.

#### ▪ Vorverarbeitung von Pseudobefehlen

Der Pseudobefehl-Pre-Prozessor von SPEL 95 konvertiert Programmtext, bevor die Grammatikanalyse des Quellprogramms ausgeführt wird. Die Konvertierung durch den Pseudobefehl-Pre-Prozessor wird automatisch durchgeführt, wenn das Quellprogramm geladen oder gespeichert wird. Wenn Sie das Quellprogramm mit Hilfe des Editors bearbeiten und ein Identifizierer den #define-Pseudobefehl ersetzt, kann durch die Grammatikanalyse innerhalb der Zeile ein Fehler erkannt werden. (Die Zeile wird rot markiert.)

In solchen Fällen können Sie die Vorverarbeitung sofort ausführen lassen. Klicken Sie dafür auf <Preprocessor> in der Werkzeuggestreife. Daraus ergibt sich, dass keine Fehler mehr auftreten, wenn normale Ersetzungen ausgeführt werden.

Weiterhin werden, wenn die Vorverarbeitung ausgeführt wird, Programmbereiche, die nicht von der Vorverarbeitung betroffen sind, grau dargestellt.

PP

<Preprocessor>  
(Pre-Prozessor)

```

' #define settings
#define MAXSPD  speed 100;accel 100,100
#define MINSPD  speed 10;accel 10,10
#define ABC      "test.prg"          ' The file of "test.prg"

' Variable settings
GLOBAL read_data AS INTEGER

' Variable settings with condition branch
#ifdef value 'The case of VALUE defined
  GLOBAL work[100] AS REAL
  GLOBAL a_sin,a_cos AS DOUBLE
#else
  EXTERN GLOBAL work[100] AS REAL
  EXTERN GLOBAL a_sin,a_cos AS DOUBLE
#endif

```

**#define**

<b>Funktion</b>	Definiert Identifizierer, die durch eine festgelegte Ersatzzeichenkette ausgetauscht werden sollen.
<b>Format</b>	<code>#define [Identifizierer]{(Parameter){,[Parameter]}} [Ersatzzeichenkette]</code>
<b>Beschreibung</b>	<p>(1) Durchsucht ein Programm nach dem angegebenen Identifizierer, ersetzt diesen an jeder gefundenen Stelle durch die Ersatzzeichenkette und kompiliert.</p> <p>(2) Der festgelegte Identifizierer kann zum bedingten Kompilieren mit <code>#ifdef</code>- oder <code>#ifndef</code>-Befehlen verwendet werden.</p> <p>Zwischen Identifizierer und der Ersatzzeichenkette muss mindestens ein Leerzeichen stehen.</p> <p>Parameter müssen durch runde Klammern ( ) eingeschlossen werden. Jeder Parameter muss durch Kommas "," getrennt werden. Es dürfen maximal 8 Parameter angegeben werden. Wenn Parameter eingesetzt werden, kann dieser Befehl als Makro verwendet werden.</p> <p>[Identifizierer]</p> <p>Reservierte Worte können nicht benutzt werden.</p> <p>Leerzeichen oder Tabulatoren sind nicht erlaubt.</p> <p>Wahlweise können Groß- und Kleinbuchstaben verwendet werden; sie werden entsprechend als Groß- bzw. Kleinbuchstaben erkannt.</p> <p>Das bedeutet, die folgenden Zeichenketten werden als unterschiedliche Identifizierer erkannt und unterschieden:</p> <p style="padding-left: 40px;">XYZ, Xyz, xYZ</p> <p>Identifizierer werden außerdem nach der Anzahl der Zeichen erkannt und unterschieden. Beispielsweise wird "ABC" (5 Zeichen) als unterschiedlich zu ABC (3 Zeichen) erkannt und nicht ersetzt.</p> <p>[Ersatzzeichenketten]</p> <p>Leerzeichen und Tabulatoren sind erlaubt.</p> <p>Wird ein Kommentarzeichen ( ' ) eingebracht, werden die Zeichen nach dem ( ' ) als Kommentar interpretiert und nicht in die Ersatzzeichenkette einbezogen.</p> <p>Die Ersatzzeichenkette ist nicht unbedingt erforderlich. In diesem Fall wird der angegebene Identifizierer durch nichts ersetzt, d.h., de facto wird der Identifizierer aus dem Programm gelöscht.</p>



**#ifdef...#else...#endif, #ifndef...#else...#endif**

<b>Funktion</b>	Verhindert die Ausführung spezieller Programmteile.
<b>Format</b>	<pre> #ifdef [Identifizierer] : {#else} : #endif  #ifndef Identifizierer] : {#else} : #endif </pre>
<b>Beschreibung</b>	<p>* Bis zu 7 Verschachtelungen sind möglich.</p> <p>Werden diese Befehle zusammen mit Pseudo-Befehlen verwendet, führen #endif, #ifdef und #ifndef eine bedingte Kompilierung durch.</p> <p>#ifdef überprüft zuerst, ob der angegebene Identifizierer derzeit durch #define definiert ist. Ist dies der Fall, werden die Anweisungen zwischen #ifdef und #else kompiliert. Existiert kein #else, wird bis zu #endif kompiliert. Falls keine Definition durch #define gefunden wird, werden die Anweisungen zwischen #ifdef und #endif übersprungen und nicht kompiliert.</p> <p>#ifndef überprüft die gegenteilige Bedingung zu #ifdef.</p> <p>#ifndef überprüft zuerst, ob der angegebene Identifizierer derzeit durch #define definiert ist. Ist keine Definition durch #define vorhanden, werden die Anweisungen zwischen #ifndef und #else kompiliert. Existiert kein #else, wird bis zu #endif kompiliert. Wird eine Definition durch #define gefunden, werden die Anweisungen zwischen #ifndef und #endif übersprungen und nicht kompiliert.</p>

**#include**

<b>Funktion</b>	Fügt spezielle „Include“-Dateien ein.
<b>Format</b>	<pre>#include „Name der Include-Datei“</pre> <p>Name der Include-Datei: Fügen Sie den Namen in Anführungszeichen ein („“), und legen Sie ihn durch eine Zeichenkette fest. („.inc“) wird als Erweiterung benötigt.</p> <p>* Bis zu 7 Verschachtelungen sind möglich.</p>
<b>Beschreibung</b>	<p>Bezieht den Inhalt der angegebenen Quellprogrammdatei mit ein. #include erlaubt die Einbeziehung deklarierter globaler Variablen und von #define-Definitionen aus anderen Dateien.</p> <p>Eine Include-Datei kann eine sekundäre Include-Datei beinhalten. Beispielsweise kann FILE 2 in FILE 1 enthalten sein und FILE 3 in FILE 2. Dieser Vorgang wird als Verschachtelung bezeichnet. Es sind maximal 7 Schleifen (einschließlich der Originaldatei) erlaubt.</p>

**Beispiele**

```
<Dateiname: valdef.inc>
#ifdef VALUE
GLOBAL WORK[100] AS REAL
GLOBAL A_SIN,A_COS AS DOUBLE
#else
EXTERN GLOBAL WORK[100] AS REAL
EXTERN GLOBAL A_SIN,A_COS AS DOUBLE
#endif
```

```
<Dateiname: main.prg>
#define VALUE
#include „valdef.inc“
FUNCTION MAIN
XQT !2, INIT
:
FEND
```

```
<Dateiname: sub.prg>
#include „valdef.inc“
FUNCTION INIT
:
FEND
```

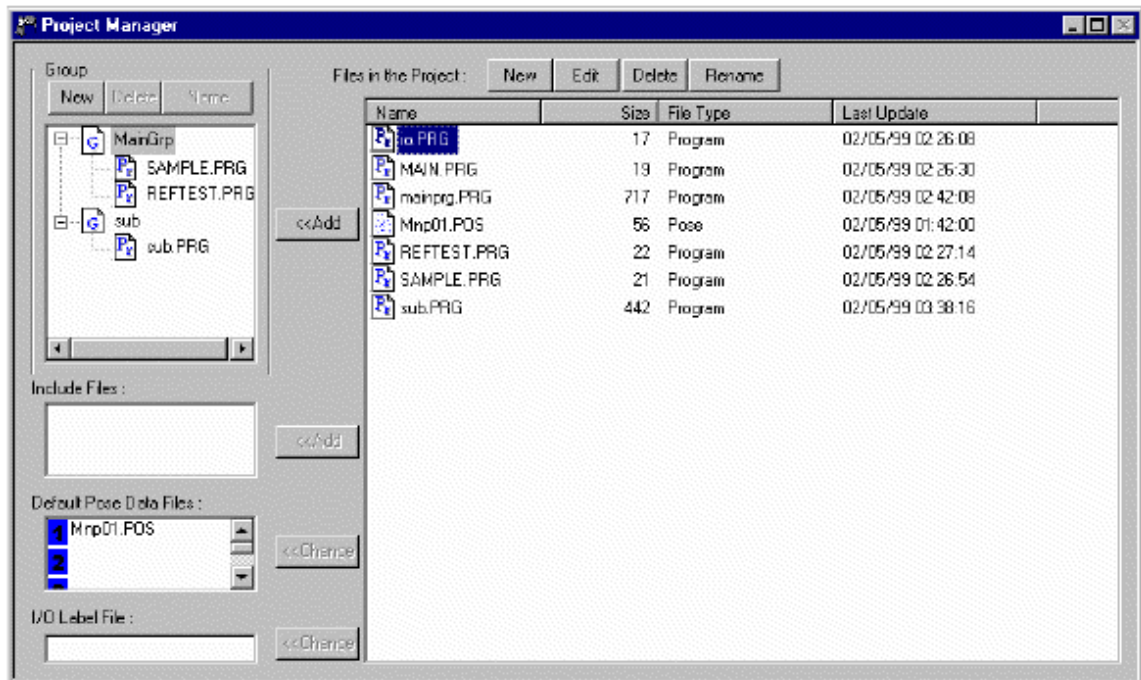
## 5.2 Editieren von Positionsdaten-Dateien

### 5.2.1 Erzeugen einer neuen Datei

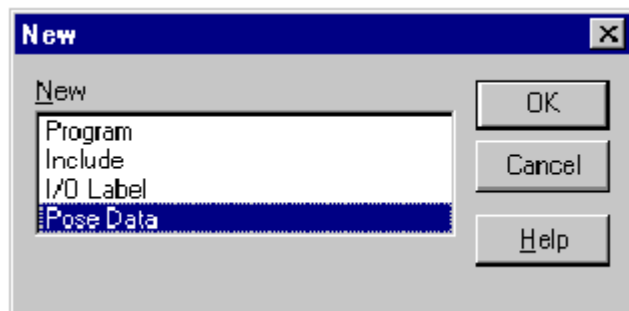
Klicken Sie auf <Programmier-Menü> in der Werkzeugleiste. Oder öffnen Sie den Projekt-Manager durch Ausführen der Befehle [Project - Project Manager]. Klicken Sie dann auf <New> (Neu) im [Files in the Project]-Fenster.



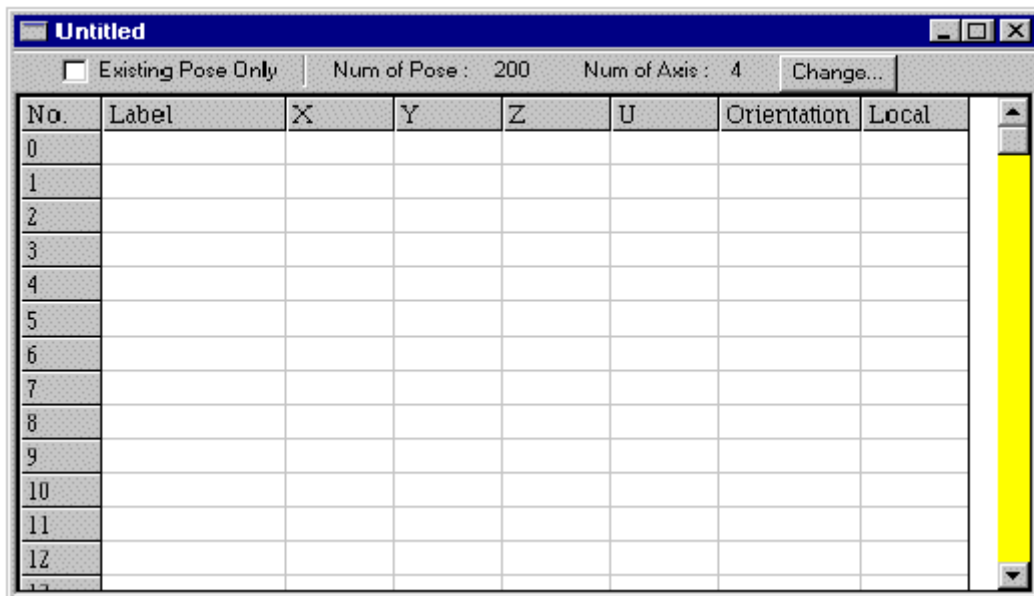
<Programming  
Pane>  
(Programmier-  
menü)



Um eine Positionsdaten-Datei zu erzeugen, wählen Sie „Pose Data“.



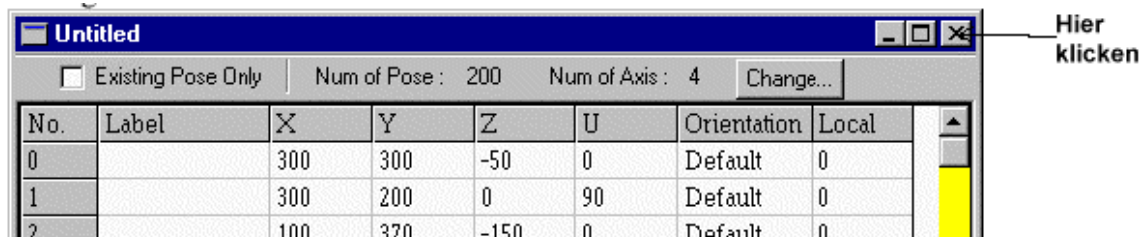
Das Fenster zum Bearbeiten der Positionsdaten-Datei wird geöffnet:



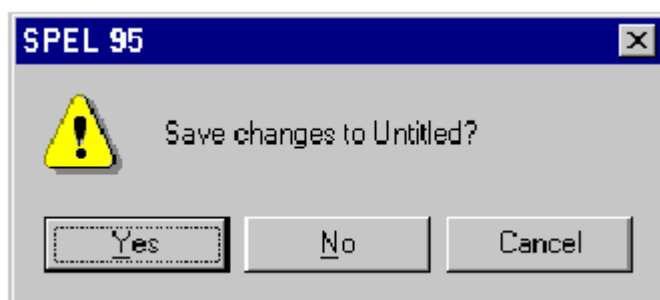
Klicken Sie in die Anzeige, um Daten einzugeben. Geben Sie dann Daten ein.

#### ▪ Speichern einer Datei

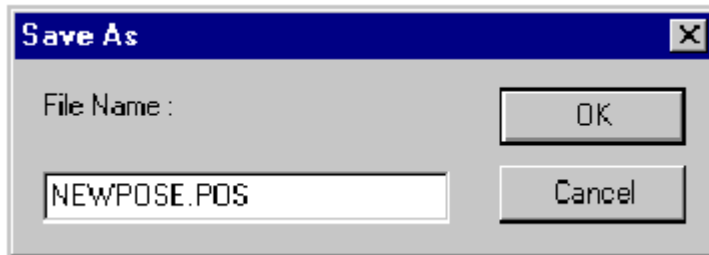
Klicken Sie auf [File] – [Save as] (Datei – Speichern unter), oder klicken Sie auf das Kreuz in der rechten oberen Ecke des Fensters.



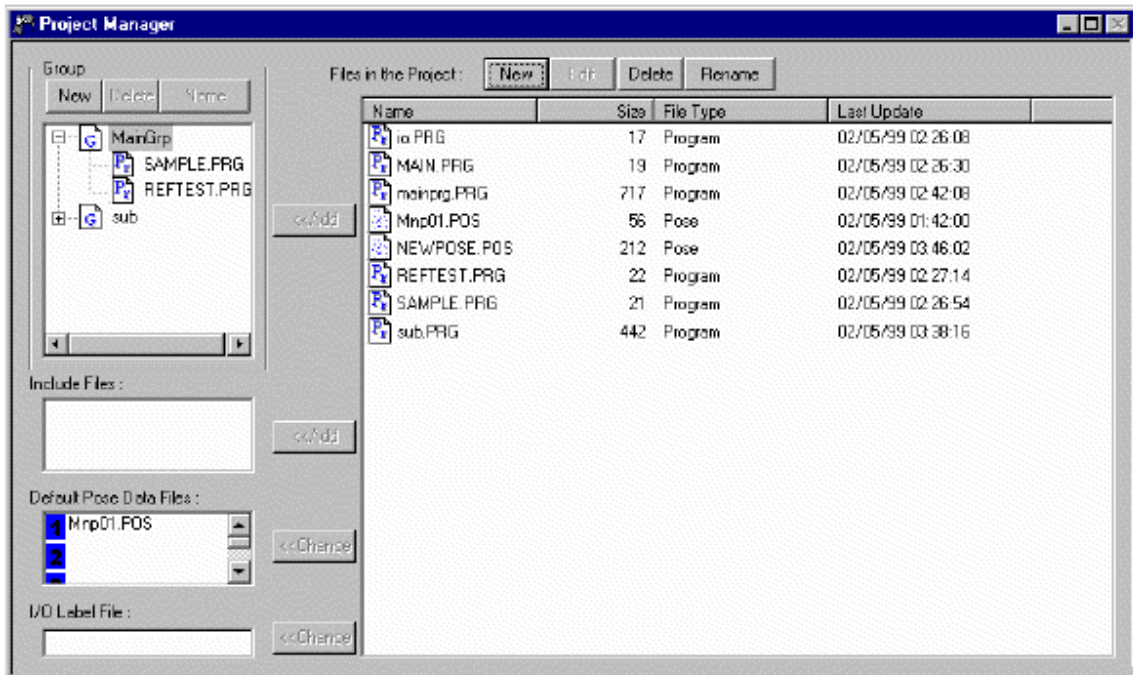
Die nachfolgende Meldung erscheint. Zum Speichern klicken Sie auf <Yes> (Ja).



Geben Sie dann einen treffenden Dateinamen in dem Textfeld des erscheinenden Dialogs ein.



Die neue Datei wurde im Projekt registriert. Die Datei „NEWPOSE.POS“ wurde der Liste hinzugefügt:

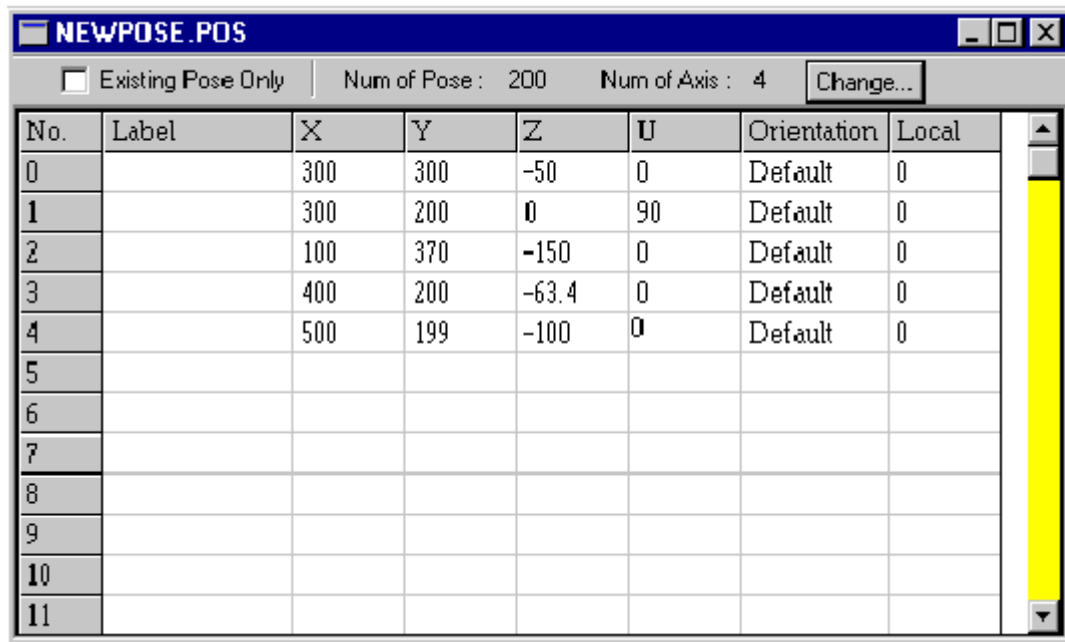


## 5.2.2 Bearbeiten bereits existierender Dateien

Um eine existierende Datei zu bearbeiten, wählen Sie die Datei aus, die Sie bearbeiten möchten, und klicken Sie dann auf <Edit> (Bearbeiten). Oder doppelklicken Sie einfach auf die zu bearbeitende Datei.

### ▪ Eingeben von Koordinaten

Benutzen Sie die Maus oder die Pfeiltasten, um den Cursor zum Eingeben der Daten hin- und herzubewegen. Geben Sie numerische Daten in den Spalten für X, Y, Z und U ein.

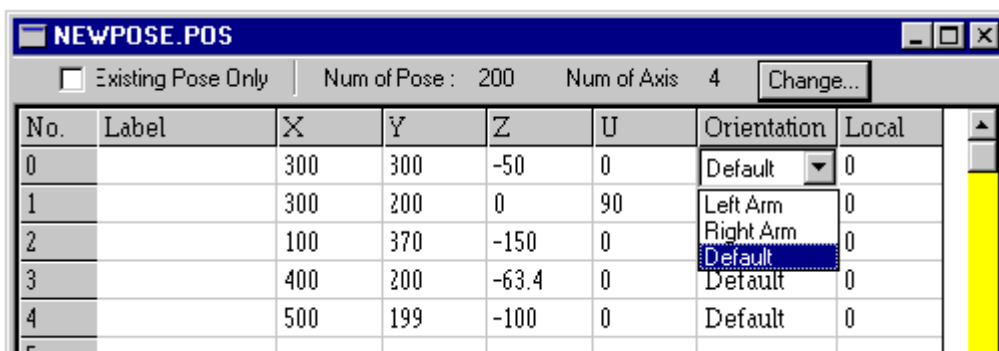


The screenshot shows the 'NEWPOSE.POS' dialog box with the 'Existing Pose Only' checkbox unchecked. The 'Num of Pose' is 200 and 'Num of Axis' is 4. A 'Change...' button is visible. The table below contains the following data:

No.	Label	X	Y	Z	U	Orientation	Local
0		300	300	-50	0	Default	0
1		300	200	0	90	Default	0
2		100	370	-150	0	Default	0
3		400	200	-63.4	0	Default	0
4		500	199	-100	0	Default	0
5							
6							
7							
8							
9							
10							
11							

Wenn Sie Daten in einen Bereich z.B. für X eingeben und dann in eine andere Zeile wechseln, obwohl Sie die Achsen X, Y, Z und U verwenden, werden für die anderen Werte jeweils „0“ und für die Spalte „Orientation“ (Ausrichtung) „Default“ (Standard) eingefügt.

Benutzen Sie die Auswahlliste, um eine gewünschte Ausrichtung auszuwählen. Klicken Sie hierfür auf den Pfeil rechts in der Auswahlliste.



The screenshot shows the 'NEWPOSE.POS' dialog box with the 'Orientation' dropdown menu open for row 0. The menu options are 'Default', 'Left Arm', 'Right Arm', and 'Default'. The 'Default' option is currently selected.

No.	Label	X	Y	Z	U	Orientation	Local
0		300	300	-50	0	Default	0
1		300	200	0	90	Left Arm	0
2		100	370	-150	0	Right Arm	0
3		400	200	-63.4	0	Default	0
4		500	199	-100	0	Default	0
5							

In der Spalte „Label“ (Sprungmarke) können Sie Sprungmarken auswählen, um Positionsdaten zu identifizieren. Maximal 15 Zeichen können für den Namen einer Sprungmarke verwendet werden.

Derselbe Name kann nicht noch einmal innerhalb der gleichen Datei verwendet werden. Versuchen Sie eine Sprungmarke zu verwenden, die bereits benutzt wird, erscheint folgende Meldung.



### 5.2.3 Bearbeiten

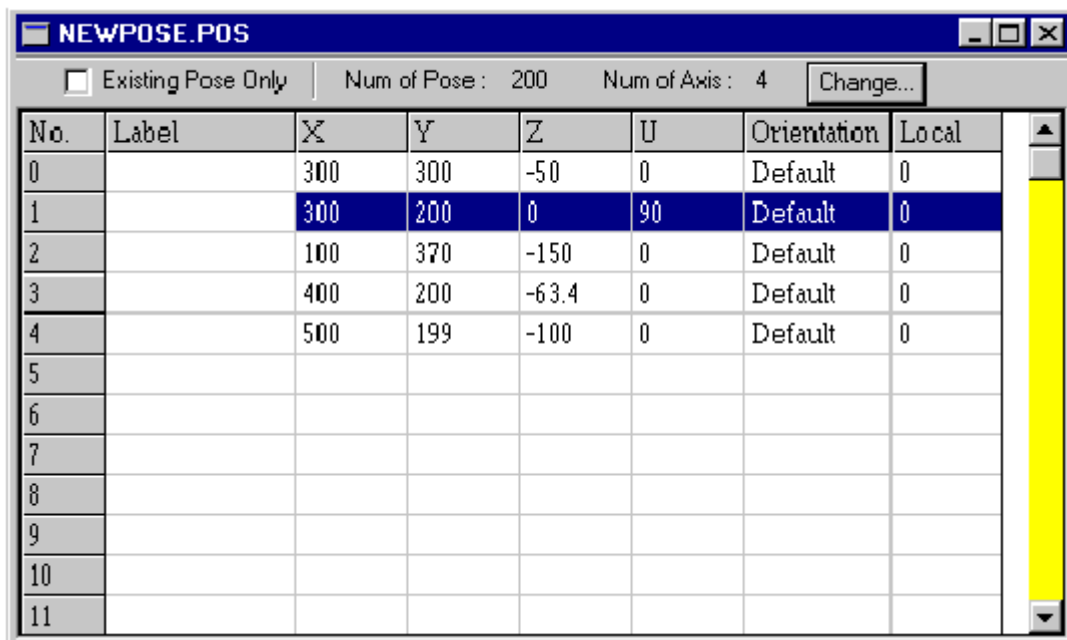
Sie können bestimmte Positionsdaten auswählen und daraus unterschiedliche Positionsdaten erzeugen, die auf den früheren basieren.

Die folgenden Funktionen werden nachfolgend beschrieben:

- Zeile auswählen
- Ausschneiden
- Kopieren
- Einfügen
- Einfügen mit Offset
- Löschen

#### ▪ Zeile auswählen

Klicken Sie auf die gewünschte Zeilennummer (im Beispiel Zeilennummer 1). Die entsprechende Zeile wird markiert.



The screenshot shows a window titled "NEWPOSE.POS" with a toolbar containing a checkbox for "Existing Pose Only", "Num of Pose : 200", "Num of Axis : 4", and a "Change..." button. Below the toolbar is a table with 8 columns: "No.", "Label", "X", "Y", "Z", "U", "Orientation", and "Local". Row 1 is highlighted in blue.

No.	Label	X	Y	Z	U	Orientation	Local
0		300	300	-50	0	Default	0
1		300	200	0	90	Default	0
2		100	370	-150	0	Default	0
3		400	200	-63.4	0	Default	0
4		500	199	-100	0	Default	0
5							
6							
7							
8							
9							
10							
11							

Durch Ziehen der Spalte „No.“ können Sie mehrere Zeilen markieren. Nur wenn mindestens eine Zeile markiert ist, kann ausgeschnitten oder kopiert werden!

Klicken Sie einmal auf das Feld „No.“, werden alle Daten innerhalb der Tabelle markiert.

Das Markieren einer Zeile gestattet die Auswahl der Koordinaten und der Ausrichtung, aber nicht die Auswahl der Sprungmarken!



### ▪ Ausschneiden

Benutzen Sie diese Funktion, um markierte Zeilen, einzelne Koordinaten oder Sprungmarken auszuschneiden.. Die ausgeschnittenen Daten werden temporär in der Zwischenablage gelagert. Um sie auszuschneiden, klicken Sie auf [Edit] und dann auf [Cut]. Sie können auch die Tastenkombination <Strg>+<X> verwenden.



<Cut>  
(Ausschneiden)

NEWPOSE.POS							
<input type="checkbox"/> Existing Pose Only							
		Num of Pose : 200		Num of Axis 4		Change	
No.	Label	X	Y	Z	U	Orientation	Local
0		300	300	-50	0	Default	0
1							
2							
3		400	200	-63.4	0	Default	0
4		500	199	-100	0	Default	0
5							
6							
7							
8							
9							
10							
11							

Die Daten werden ausgeschnitten. Die ausgeschnittenen Bereiche bleiben leer.

### ▪ Kopieren



<Copy>  
(Kopieren)

Benutzen Sie diese Funktion, um markierte Daten in die Zwischenablage zu kopieren. Die kopierten Daten bleiben in ihrer ursprünglichen Form erhalten, können aber aus dem Zwischenspeicher an jede andere Stelle kopiert werden. Um die Daten zu kopieren, klicken Sie auf [Edit] und dann auf [Copy]. Sie können auch die Tastenkombination <Strg>+<C> verwenden.

### ▪ Paste



<Paste>  
(Einfügen)

Benutzen Sie diese Funktion, um in der Zwischenablage befindliche Daten an der Position des Cursors einzufügen. Die Daten in der Zwischenablage bleiben solange erhalten, bis Sie neue Daten ausschneiden oder kopieren. Um die Daten einzufügen, klicken Sie auf [Edit] und dann auf [Paste]. Sie können auch die Tastenkombination <Strg>+<V> verwenden.

- **Paste mit Offset**

Wird benötigt, um die in der Zwischenablage befindlichen Daten mit einem bestimmten Offset einzufügen. Bewegen Sie den Cursor zu der Position, an der die Werte eingefügt werden sollen, und klicken Sie auf [Edit] und dann auf [Paste with Offset]. Ein Menü zum Eingeben der Offset-Werte erscheint. Geben Sie einen Wert für den Offset ein. Die eingefügten Werte werden zu Offset-Werten.

Wenn Sie Daten mit Offset einfügen, erscheint das folgende Fenster:

Axis	Current Value	Offset
X	300	0
Y	300	0
Z	-50	0
U	0	0

Die aus der Zwischenablage eingefügten Werte werden als aktuelle Werte „Current Value“ angezeigt. Nachdem die Zeilen 0 und 1 kopiert wurden und [Paste with Offset] ausgeführt wurde, wird folgendes angezeigt:

Axis	Current Value	Offset
X		0
Y		-50
Z		0
U		0

Da mehrere Zeilen aus der Zwischenablage eingefügt wurden, wird die aktuelle Position nicht angezeigt. Im Beispiel wird -50 als Wert für die Y-Achse eingestellt. Klicken Sie auf <OK>.

No.	Label	X	Y	Z	U	Orientation	Local
0		300	300	-50	0	Default	0
1		300	200	0	90	Default	0
2		100	370	-150	0	Default	0
3		400	200	-63.4	0	Default	0
4		500	199	-100	0	Default	0
5							
6		300	250	-50	0	Default	0
7		300	150	0	0	Default	0
8							
9							
10							
11							

Die in den Zeilen 6 und 7 dargestellten Werte für die Y-Achse sind die Werte für die Zeilen 1 und 0 mit einem Offset von  $-50$ .

- **Löschen**

Wird verwendet, um Daten in markierten Zeilen zu löschen. Um die Daten zu löschen, klicken Sie auf [Edit] und dann auf [Delete].

- **Anzeigen ausschließlich existierender Zeilen**

Wenn Sie nur die tatsächlich existierenden Daten anzeigen lassen wollen, markieren Sie das Optionsfeld [Existing Pose Only].

No.	Label	X	Y	Z	U	Orientation	Local
0		300	300	-50	0	Default	0
1		300	200	0	90	Default	0
2		100	370	-150	0	Default	0
3		400	200	-63.4	0	Default	0
4		500	199	-100	0	Default	0

**Ankreuzen**

### ▪ Ändern der Anzeigenreihenfolge

Die Reihenfolge der Positionsdaten kann der Anwender selbst festlegen. Einmaliges Klicken auf die Titelzelle, z.B. [No.], [Label], [X], etc., bewirkt, dass die Daten in aufsteigender Reihenfolge gemäß der Werte in den Zeilen sortiert werden. Erneutes Klicken auf die Titelzelle bewirkt, dass die Daten in absteigender Reihenfolge sortiert werden.

Hier klicken

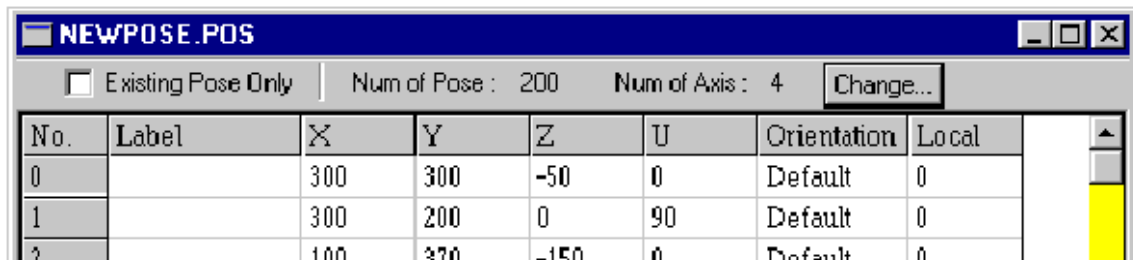
No.	Label	X	Y	Z	U	Orientation	Local
2		100	370	-150	0	Default	0
0		300	300	-50	0	Default	0
1		300	200	0	90	Default	0
3		400	200	-63.4	0	Default	0
4		500	199	-100	0	Default	0
5							
6							
7							
8							
9							
10							
11							

X Daten und Positionsnummern werden neu arrangiert

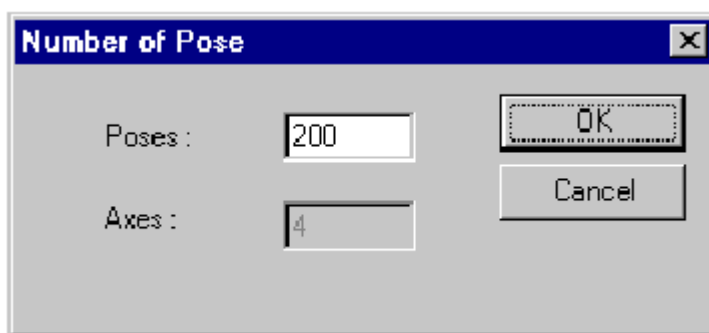
Wenn Sie, wie im oben dargestellten Beispiel, auf X klicken, erscheint die Reihenfolge der Daten in aufsteigender Reihenfolge. Die Bildschirmanzeige wird neu arrangiert. Aber die Zeilennummern bleiben erhalten.

## 5.2.4 Ändern der Positionsdaten-Anzahl

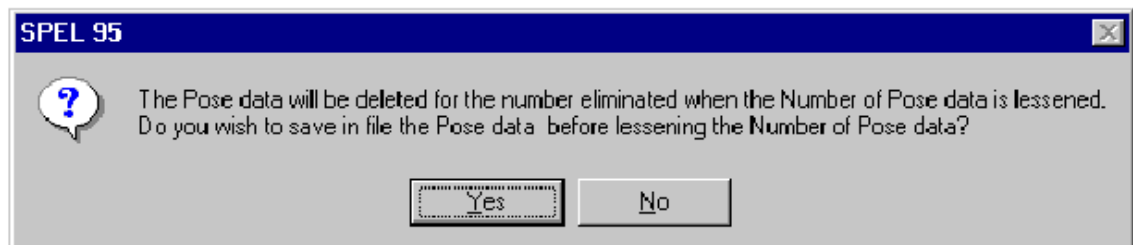
Um die maximale Positionsdaten-Einstellungen zu ändern, klicken Sie auf <Change> (Ändern).



Geben Sie den maximalen Wert für die Positionsdaten in dem Textfeld [Pose] ein, und klicken Sie dann auf <OK>.



Wenn sich durch Ihre Eingabe die Anzahl der Positionsdaten verringert hat, erscheint folgender Dialog. Sichern Sie, falls notwendig.



## 5.3 Editieren von E/A-Sprungmarken-Dateien

### 5.3.1 E/A-Sprungmarken

Um die Anwender-E/A und die Speicher-E/A zu steuern, müssen die E/A-Bitnummern, -Bytenummern und -Wortnummern festgelegt werden. Mit gewöhnlich automatisierter Ausrüstung werden jeder Anwender- und Speicher-E/A Ressourcen zugewiesen, die über jeweilig festgelegte Funktionen verfügen. In SPEL 95 werden einmalige Namen für jedes E/A-Bit, -Byte und -Wort der Anwender- und Speicher-E/A registriert. Somit wird eine Steuerung per Namen, nicht durch Zahlen, von den Programmen gestattet. Diese einmaligen Namen werden E/A-Sprungmarken genannt.

Dies vereinfacht Programme und macht die Visualisierung einfacher.

FUNCTION I/O CONTROL	Funktion: E/A-Steuerung
DO	
ON 1	^Startet Actuator-Bedienung
WAIT SW(1) = ON	^Wartet bis Sensor auf ON steht
OFF1	^Stoppt Actuator-Bedienung
IF INW(3) >= 10000 THEN	^Wenn die gemessenen Daten 10000 überschreiten
OUTW 3, 5000	^Ändert die Anweisungsdaten auf 5000
ELSEIF INW(3) < 500 THEN	^Wenn die gemessenen Daten 500 unterschreiten
OUTW 3, 1000	^Ändert die Anweisungsdaten auf 1000
ELSE	^Wenn die gemessenen Daten von den Werten abweichen
\$OUT 4, 100	^Ändert den Status auf 100
ENDIF	
WAIT \$IN(1) = 10	^Wartet bis Merkmal 10 erreicht ist.
LOOP WHILE \$IN(4) <> 100	^Wiederholt solange Status ist ungleich 100 .
FEND	

Anwender E/A- und Speicher E/A-Sprungmarken werden wie nachfolgend gezeigt registriert:

Eingangs Bitnummer 1:	Sensor
Ausgangs Bitnummer 1:	Actuator
Eingangs Wortnummer 3:	Gemessene Daten
Ausgangs Wortnummer 3:	Anweisungsdaten
Speicher E/A-Bytenummer 1:	Status
Speicher E/A-Bytenummer 4:	Merkmal (Flag)

Beispiel: Programm benutzt E/A-Sprungmarken:

```
FUNCTION I/O CONTROL
  DO
    ON Actuator
    WAIT SW(Sensor) = ON; OFF Actuator
    IF INW(gemessene Daten) >= 10000 THEN; OUTW Anweisungsdaten,
    5000
    ELSEIF INW(gemessene Daten) < 500 THEN; OUTW Anweisungsdaten, 1000
    ELSE;      $OUT Status, 100
    ENDIF
    WAIT $IN(Merkmal) = 10
    LOOP WHILE $IN(Status) <> 100
  FEND
```

Es ist möglich, die E/A-Steuerungsfunktion auf diese Weise zu beschreiben. Nicht benötigte Kommentare können unterlassen werden, damit das Programm leichter lesbar wird.

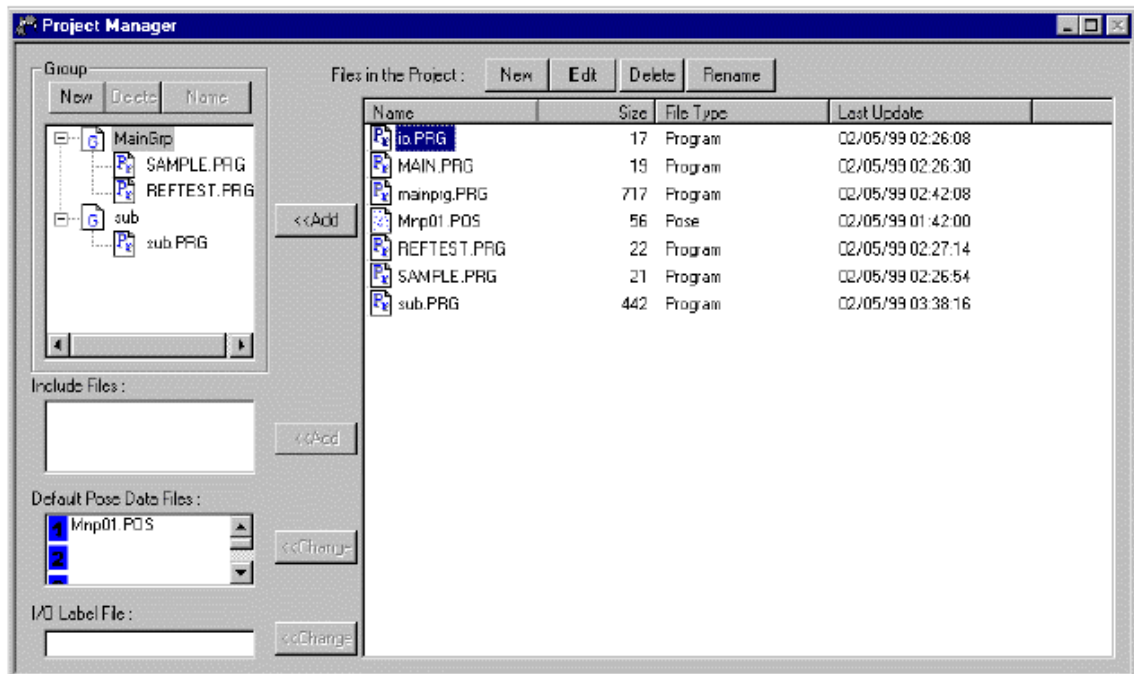
Projekte können mittels der E/A-Sprungmarkendateien, die die E/A-Sprungmarken definieren, registriert werden. Der Systemmonitor bezieht sich für die Anzeige ebenfalls auf die E/A-Sprungmarken.

### 5.3.2 Erzeugen neuer E/A-Sprungmarken-Dateien

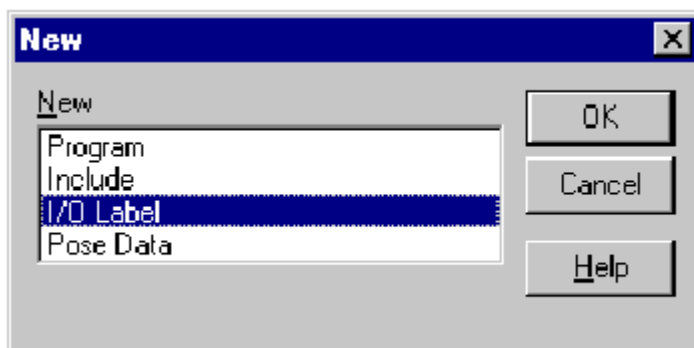
Klicken Sie auf <Programmiermenü> in der Werkzeugleiste. Klicken Sie dann auf <New> (Neu) in der Liste [Files in the Project].



<Programming  
Pane>  
(Programmier-  
menü)

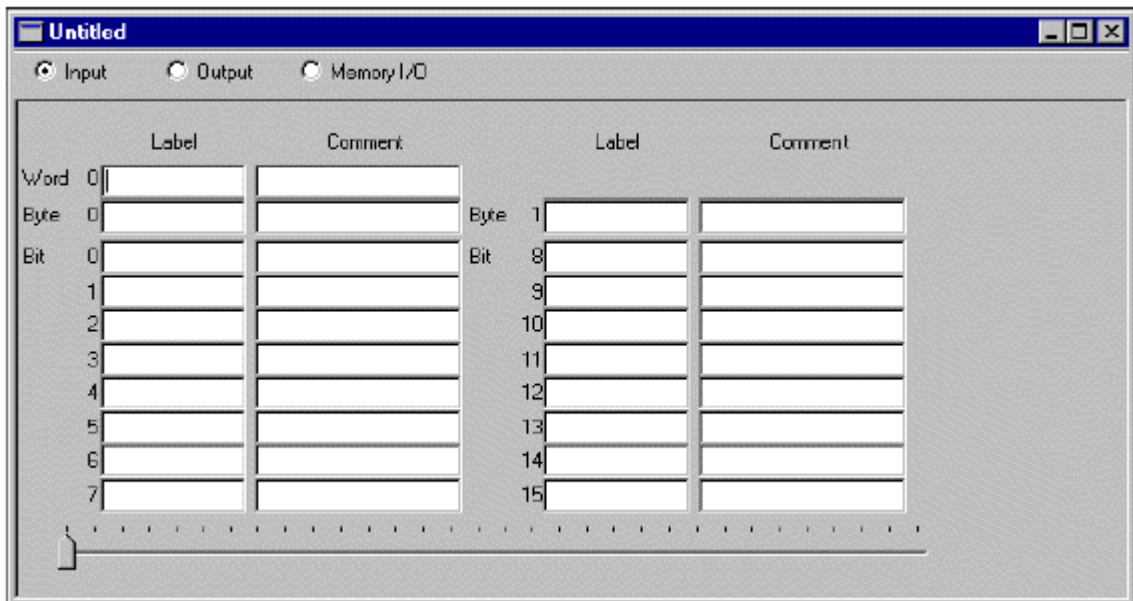


Um eine neue Datei zu erzeugen, wählen Sie „I/O-Label“ (E/A-Sprungmarke), und klicken Sie auf <OK>.



Das Fenster zum Bearbeiten von E/A-Sprungmarken-Dateien erscheint.

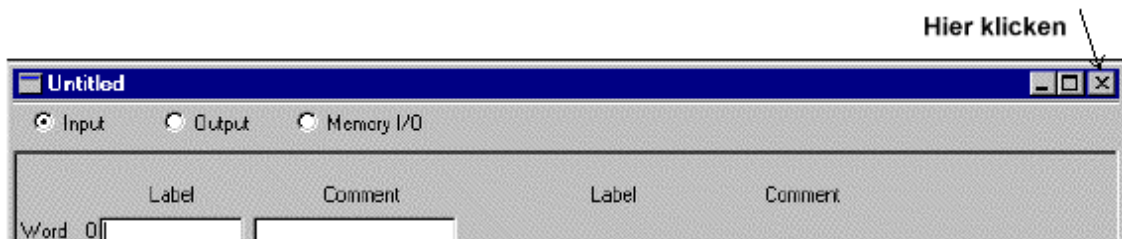




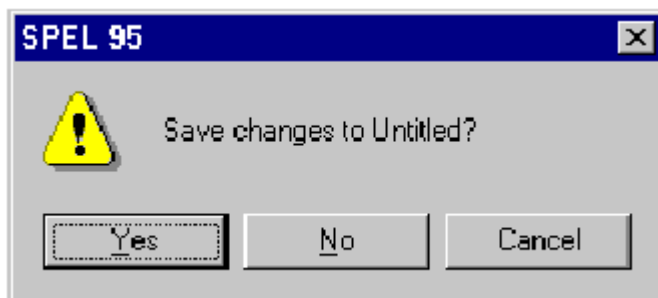
Geben Sie hier die E/A-Sprungmarken und die Kommentare ein.

- **Speichern einer Datei**

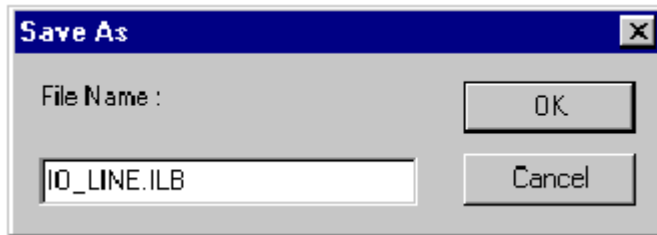
Klicken Sie auf [File] – [Save As] (Datei – Speichern unter), oder klicken Sie auf das Kreuz in der rechten oberen Ecke des Fensters.



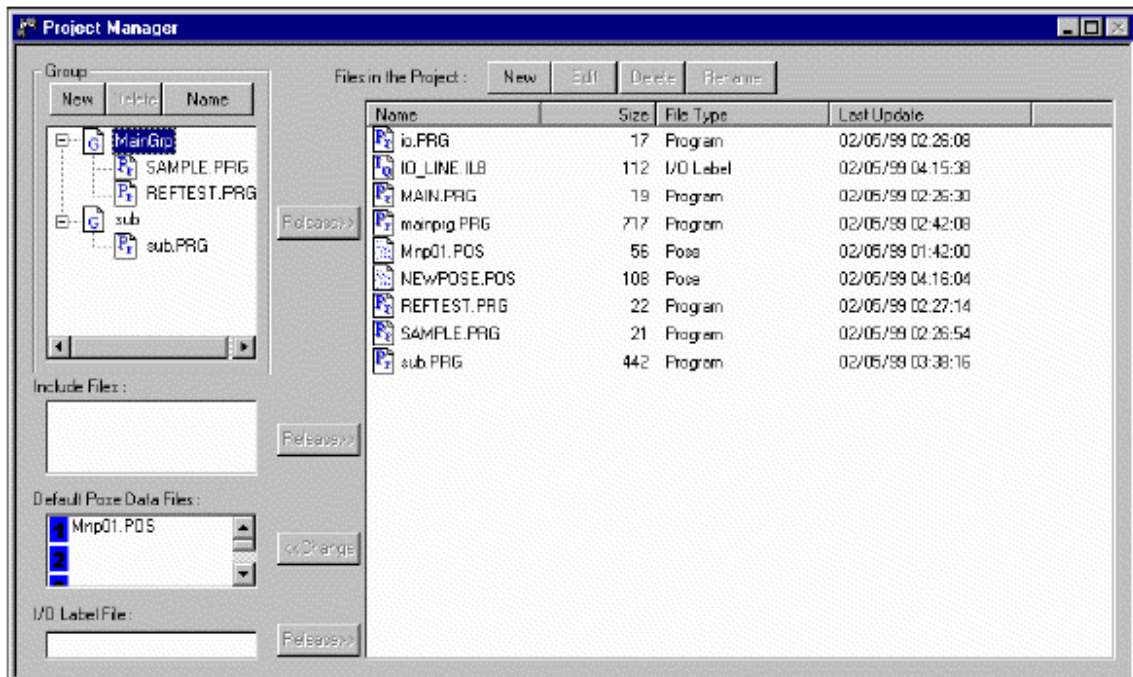
Die nachfolgende Meldung erscheint. Zum Speichern klicken Sie auf <Yes> (Ja).



Geben Sie dann einen treffenden Dateinamen in dem Textfeld des erscheinenden Dialogs ein. Geben Sie hier „IO\_LINE.LIB“ ein.



Die neue Datei wurde im Projekt registriert. Die Datei „IO\_LINE.LIB“ wurde der Liste hinzugefügt.



### 5.3.3. Bearbeiten existierender Dateien

Um eine existierende Datei bearbeiten zu können, wählen Sie die zu bearbeitende Datei aus, und klicken Sie dann auf <Edit> (Bearbeiten). Oder Doppelklicken Sie auf die zu bearbeitende Datei.

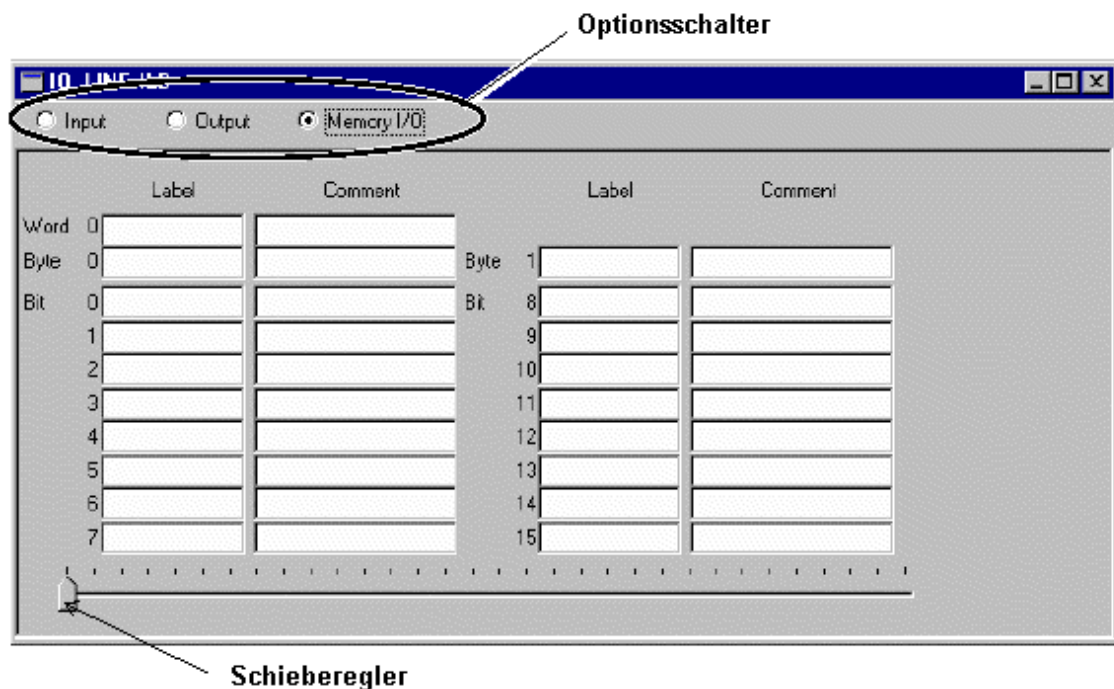
#### ▪ Bildschirmanzeige

Um den Eingang für jede E/A-Sprungmarke auszuwählen, benutzen Sie die Optionsschalter am oberen Ende des Editors.

Anwender-Eingangs-Sprungmarke: Eingangsschalter (Input)

Anwender-Ausgangs-Sprungmarke: Ausgangsschalter (Output)

Speicher-E/A-Sprungmarke: Speicher-E/A-Schalter



Wird die Anwender-E/A erweitert, können Sie die Adresse für die Speicher-E/A auswählen. Sie erreichen dies auch durch Ziehen des horizontalen Schiebereglers.

### ▪ Bearbeiten

Mit Hilfe der Maus oder der <TAB>-Taste bewegen Sie den Cursor; geben Sie dann Sprungmarkennamen und Kommentare ein. Sprungmarken können in Wort-, Byte- und Biteinheiten eingegeben werden.

The screenshot shows a software window titled "IO\_LINE\_ILB" with three radio buttons: "Input" (selected), "Output", and "Memory I/O". Below the buttons is a table with two columns for "Label" and "Comment". The table is organized into two sections: "Word" and "Bit".

	Label	Comment		Label	Comment		
Word	0						
Byte	0	Line 1	Line 1 input signal	Byte	1	Line 2	Line 2 input signal
Bit	0	Line Set	Line entrance sensor	Bit	8		
	1	Sensor A	A position sensor		9		
	2	Sensor B	B position sensor		10		
	3	Sensor C	C position sensor		11		
	4	Sensor_ACT	actuator sensor		12		
	5	Sensor_Check	Measure point sensor		13		
	6				14		
	7				15		

At the bottom of the window, there is a horizontal bar with a small icon on the left and a series of tick marks along its length.

Maximal 16 Zeichen können für einen Sprungmarkennamen verwendet werden.

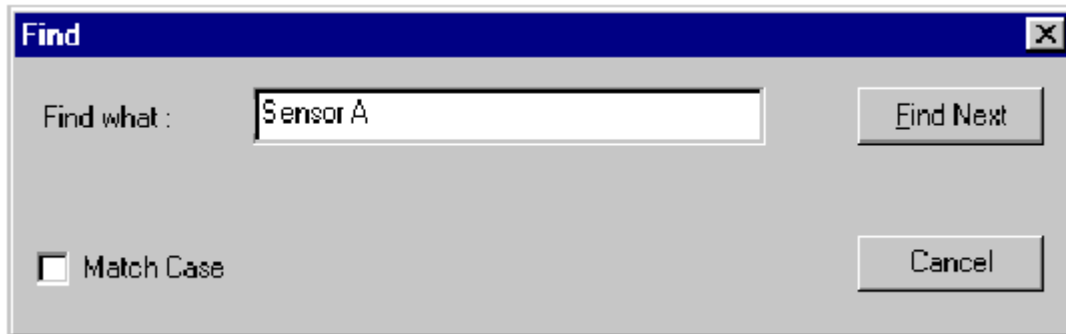
**HINWEIS**

Stellen Sie sicher, dass die Befehle mit den REMOTE-Einstellungen vereinbar sind, wenn die REMOTE-Funktionen (siehe auch "Automatische Bedienung" in Kapitel 10) eingestellt sind. Wenn die E/A-Sprungmarken-Einstellung und die REMOTE-Einstellung nicht im Einklang miteinander sind, haben die REMOTE-Einstellungen Vorrang!

Kommentare können Bemerkungen über Sprungmarken enthalten. Bis zu 32 Zeichen können benutzt werden.

- **Finden**

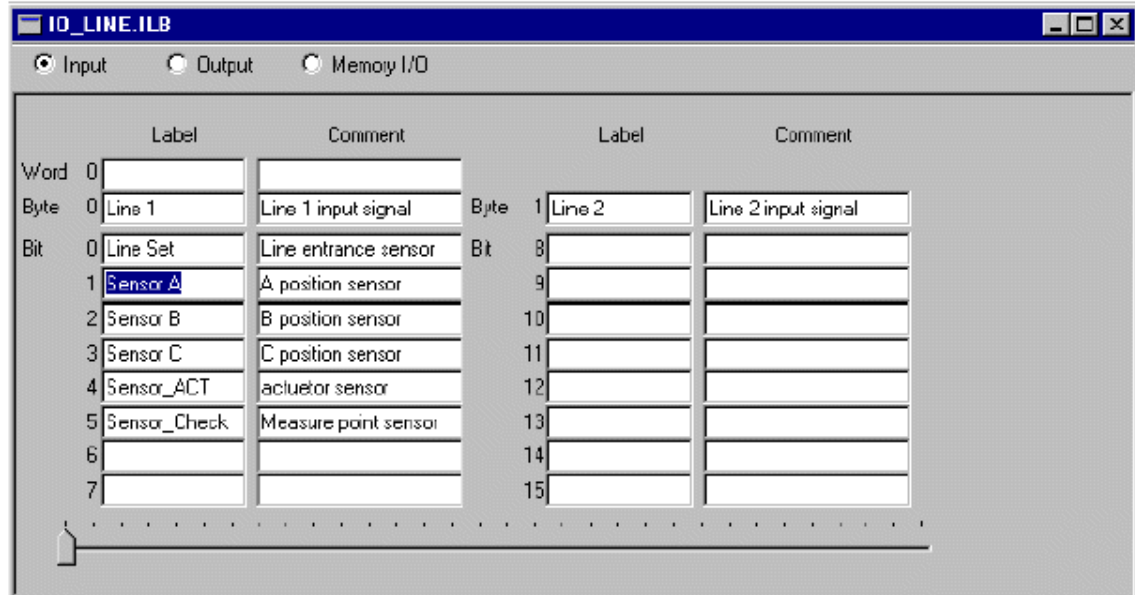
Klicken Sie auf [Edit] – [Find] (Bearbeiten – Finden), um nach E/A-Sprungmarken zu suchen.



Geben Sie den Namen der E/A-Sprungmarke, nach der gesucht werden soll, in das Textfeld ein. Klicken Sie dann auf <Find next> (Nächstes finden). Jetzt wird nach der E/A-Sprungmarke gesucht.

Markieren Sie das Optionsfeld [Match case], wenn entsprechend der eingegebenen Groß- und Kleinschreibweise gesucht werden soll.

Das Ergebnis der Suche wird angezeigt:

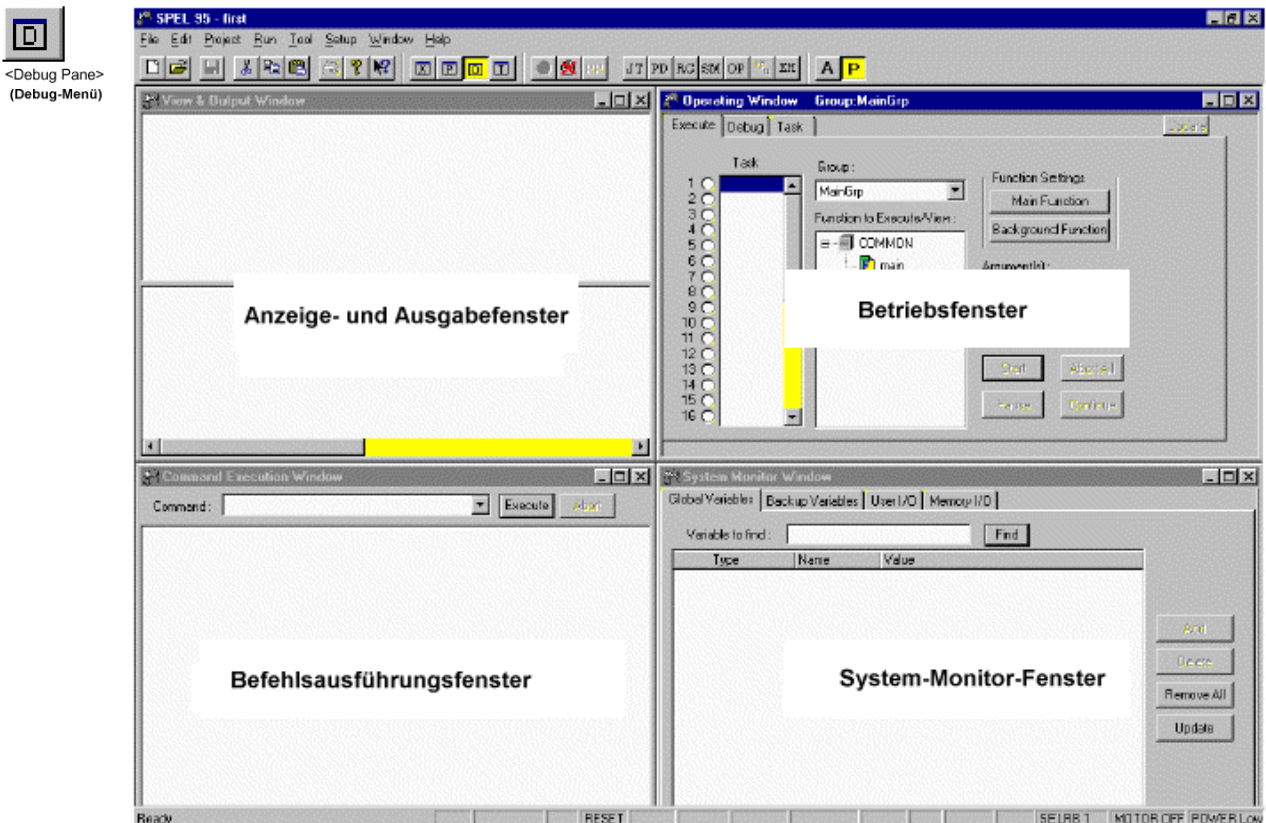


Es werden keine Ergebnisse angezeigt, wenn die Zeichenketten nicht exakt miteinander übereinstimmen.

# Kapitel 6

## Fehlerbehebung

Klicken Sie auf <Debug Pane> (Debug-Menü/Fehlerbeseitigungs-Menü) in der Werkzeugleiste.  
Das unten dargestellte Fenster erscheint.



- [Betriebsfenster]: Verrichtet größere Vorgänge, wie das Ausführen von Programmen.
- [Anzeig- und Ausgabefenster]: Zeigt eine Liste der Programme und Druckanweisungen, die in den erzeugten Programmen vorhanden sind.
- [Befehlsausführungsfenster]: Ändert Variablen oder E/A-Sprungmarken, und verfährt den Manipulator durch Eingabe von SPEL 95-Befehlen direkt.
- [System-Monitor-Fenster]: Bestätigt Variablen oder E/A-Zustände, und verändert deren Werte.

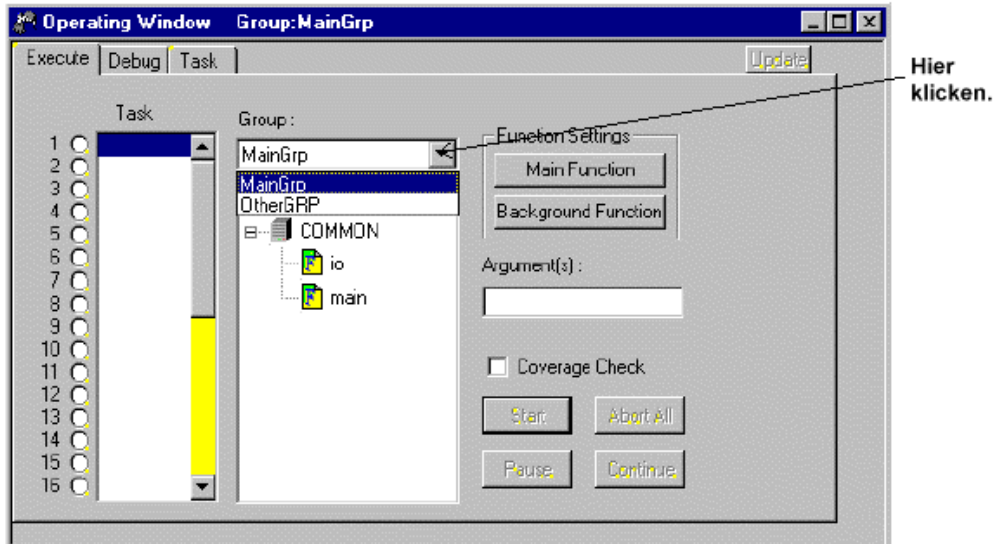
Die Fehlerliste wird in dem Fenster [Error History] angezeigt, wenn Sie auf <Error History> in der Werkzeugleiste klicken.

## 6.1 Erzeugen von Tasks (Ausführen von Programmen)

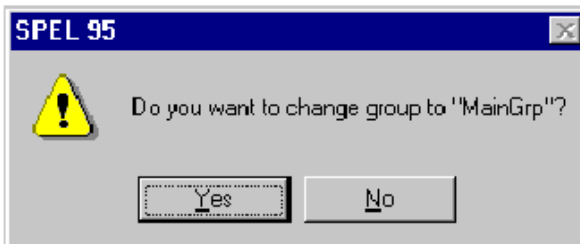
Wenn mehrere Gruppen in dem Projekt vorhanden sind, wählen Sie zuerst die Gruppe aus, mit der Sie arbeiten möchten.

### 6.1.1 Wechseln der Gruppe

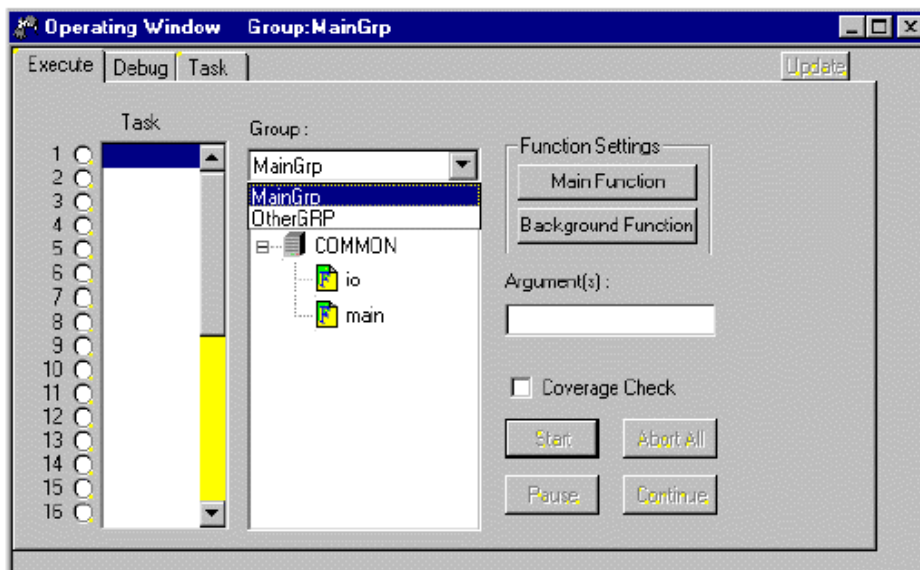
Klicken Sie auf die Auswahlliste, um die Gruppe zu wechseln.



Der nachfolgende Dialog erscheint.



Die Gruppe wird gewechselt, wenn Sie auf <Yes> klicken. Die Gruppe „MainGrp“ ist ausgewählt.



### 6.1.2 Erzeugen von Tasks (Ausführen von Programmen)

Klicken und markieren Sie die Funktionen, die Sie aus der Liste [Function to Execute/View] (Funktion zum Ausführen/ Ansehen) ausführen möchten.

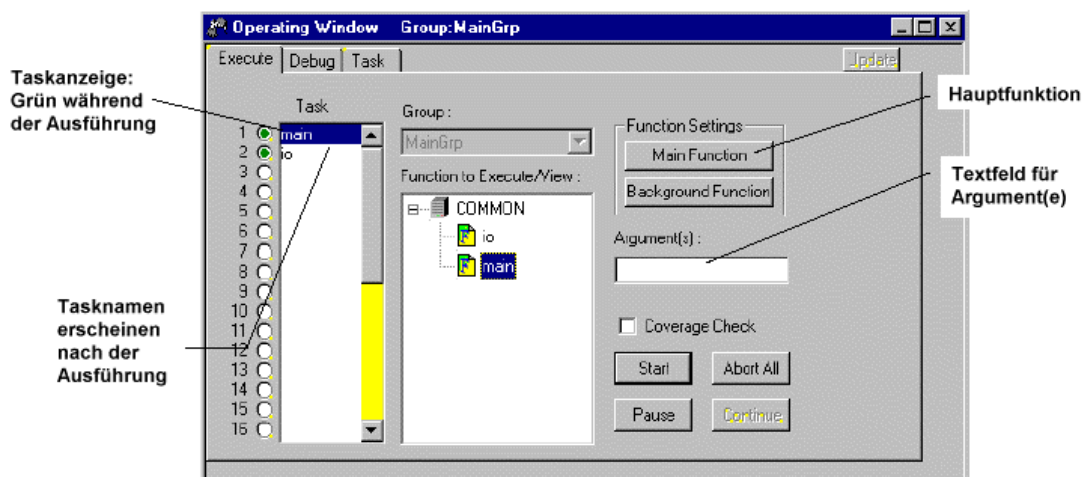
Eine Liste der Funktionen, die Sie ausführen möchten, wird im Programmanzeige-Fenster des [View&Output Window] (Anzeige- & Ausgabe-Fenster) angezeigt.

```

View & Output Window - io.PRG
FUNCTION io
loop2:
PRINT "This is IO control program."
FOR i = 8 TO 15
    ON i; WAIT 2
    OFF i; WAIT 2
NEXT i
GOTO loop2
FEND
  
```

Klicken Sie auf <Start>, oder doppelklicken Sie auf die Funktion, die Sie ausführen möchten.

Wenn Tasks erzeugt werden, erscheinen die Tasknamen in der Taskliste.



Die Taskanzeigen zeigen den jeweiligen Status der erzeugten Tasks. Der Zustand der Task und die Farbe der relevanten Taskanzeigen sind wie folgt definiert:

Taskzustand	Farbe der Taskanzeige
Während der Ausführung	Grün
Pause	Gelb
Fehler	Rot
Schnelle Pause	Dunkles Gelb



- **Anhalten aller Tasks (Pause)**

Klicken Sie auf <Pause>, um alle Tasks vorübergehend anzuhalten. Die Taskanzeigen ändern sich in gelb.

Wenn QP (Quick Pause – Schnelle Pause) eingeschaltet ist, wenn der Manipulator betrieben wird, stoppt der Manipulator unverzüglich, und die Taskanzeigen ändern sich in dunkel gelb. Die Tasks, die mit Hilfe des Befehls HTASKTYPE programmiert wurden, um während einer Pause nicht zu stoppen, halten nicht an! Auch die Hintergrundtask hält nicht an.

- **Fortsetzen aller Tasks**

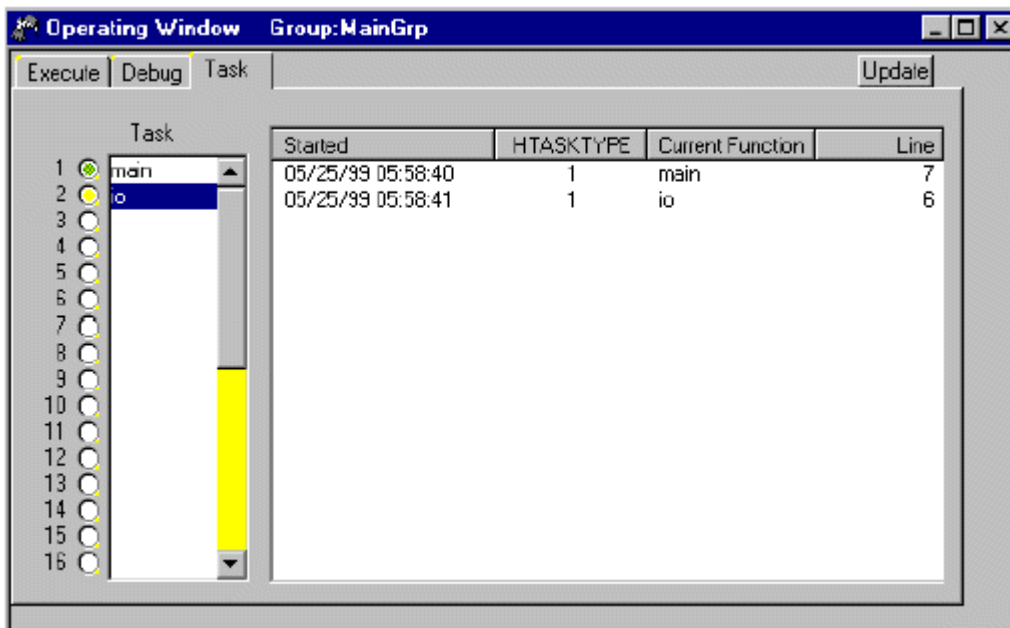
Klicken Sie auf <Continue> (Fortsetzen), um alle Tasks fortzusetzen. Die Taskanzeigen ändern sich in grün.

- **Beenden aller Tasks**

Klicken Sie auf <Abort all> (Alle abbrechen), um alle Tasks zu beenden. Die relevanten Tasknamen in der Taskliste und die Taskanzeigen verschwinden. Wird die privilegierte Task ausgeführt, erscheint ein Dialog, der den Abbruch der privilegierten Task bestätigt. Siehe auch Kapitel 12 „Roboterzelle – Steuerungsfunktionen“.

### 6.1.3 Anzeigen des Taskzustands

Das Anfangsdatum, die Anfangszeit, die Ausführungsanweisungen während einer Pause und die Stelle an der pausiert wird (Zeile, an der angehalten wird), erscheinen im Task-Menü.



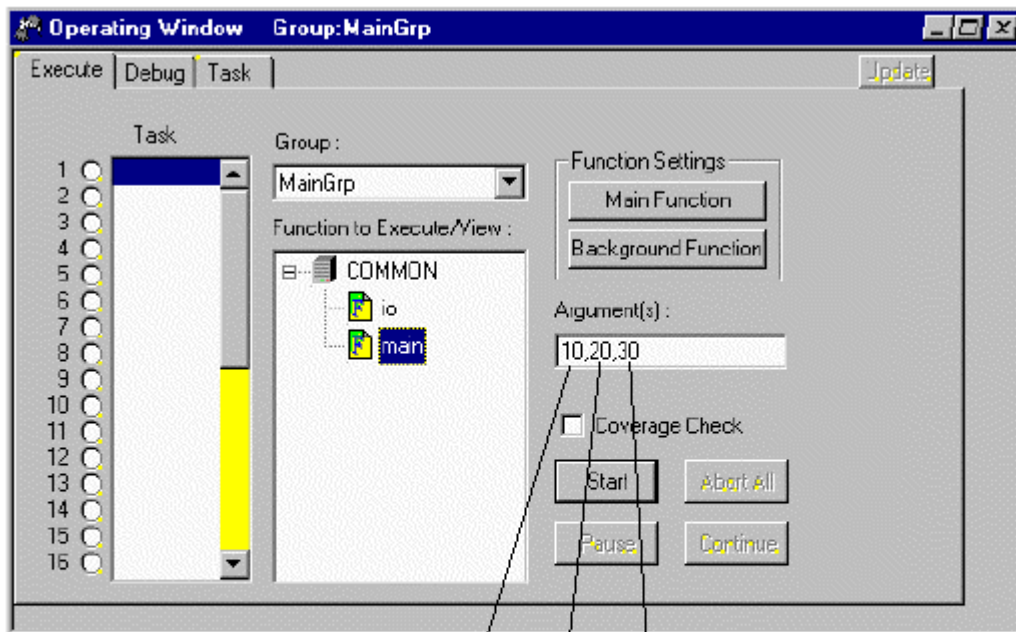
#### HINWEIS

Das Ausführen eines Programmes erzeugt eine Task. Wird ein Programm nicht ausgeführt und somit keine Task erzeugt, wird im Task- oder im Debug-Menü nichts angezeigt.

### 6.1.4 Ausführen von Funktionen nach Hinzufügen von Argumenten

Einige Funktionen, die als Unterfunktionen verwendet werden, erhalten Argumente von anderen Funktionen. Um solche Funktionen unabhängig auszuführen, fügen Sie zuerst die Argumente hinzu.

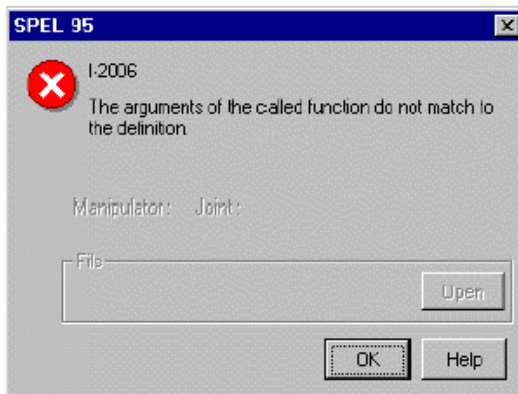
- (1) Geben Sie Parameter in das Textfeld [Argument(s)] (Argument/e) ein. Werte werden in der Reihenfolge der Parametereinträge übergeben. Daher ist es wichtig, bei der Eingabe auf die Reihenfolge zu achten. Um mehrere Argumente einzugeben, werden Kommas (,) zur Trennung benutzt.  
10, 20, 30
- (2) Klicken Sie auf und markieren Sie die Funktion, die Sie ausführen möchten, in der Liste [Function to Execute/View].
- (3) Klicken Sie auf <Start>.



```
FUNCTION main(arg1, arg2, arg3)
:
:
```

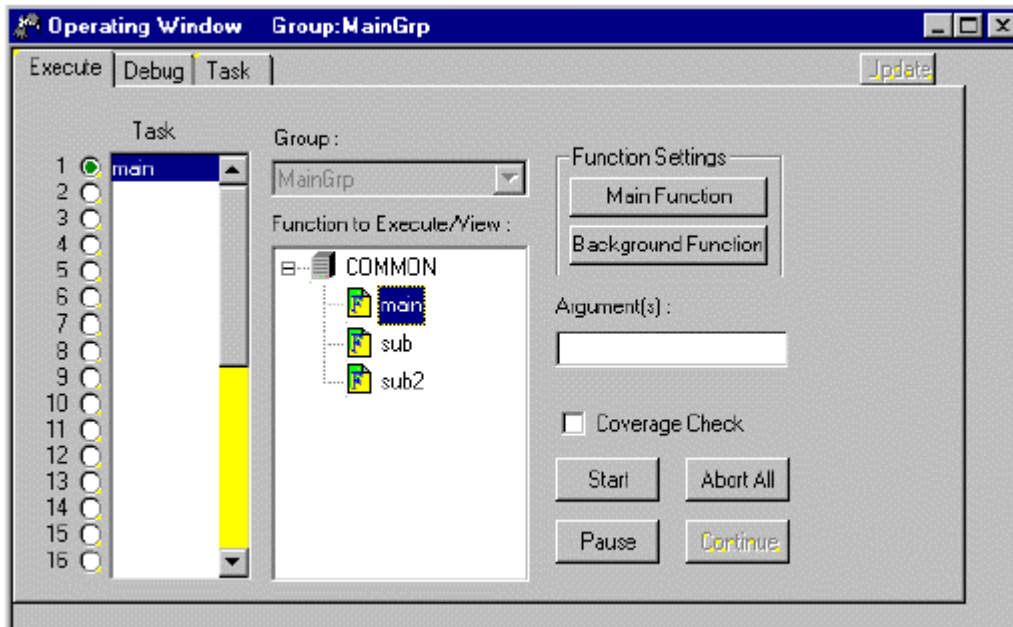
**Argumente werden, wie oben dargestellt, übertragen und Tasks werden erzeugt.**

Wenn Sie die Funktion ausführen, ohne Argumente einzugeben, erscheint nachfolgende Fehlermeldung, und es werden keine Tasks erzeugt.

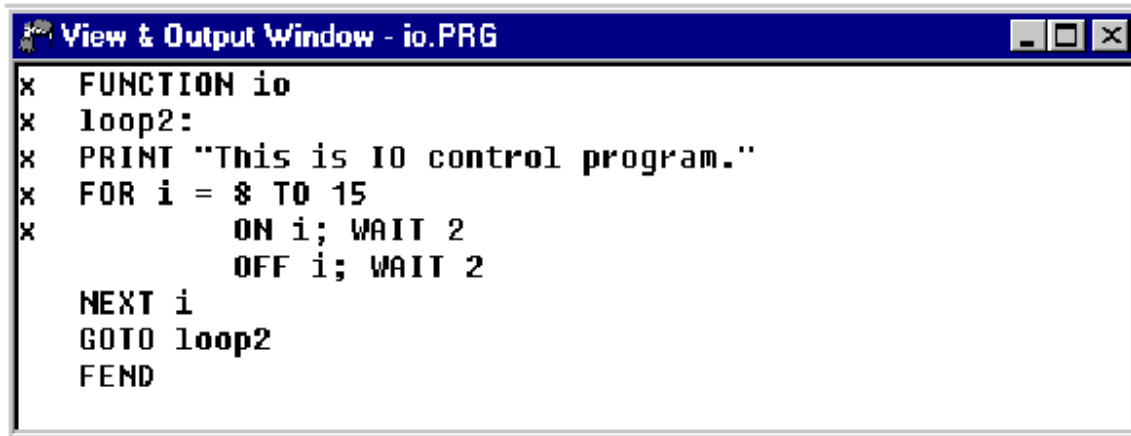


### 6.1.5 Kontrollbericht

Wenn eine Task abgebrochen wird, kann das Programm, soweit wie es ausgeführt wurde, zeilenweise dargestellt werden.

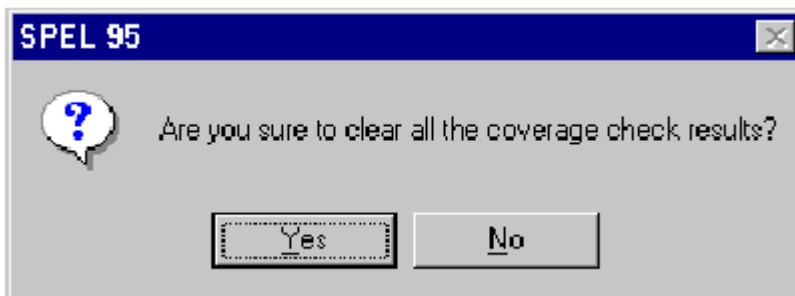


In der Liste, die im Quellprogramm-Fenster angezeigt wird, erscheint ein „X“ am Anfang der Zeilen, die vor dem Abbruch ausgeführt wurden.



Um das Ergebnis des Kontrollberichts zu löschen, entfernen Sie die Markierung in dem Optionsfeld [Coverage Check].

Wenn nachfolgende Meldung erscheint, klicken Sie auf <Yes>.



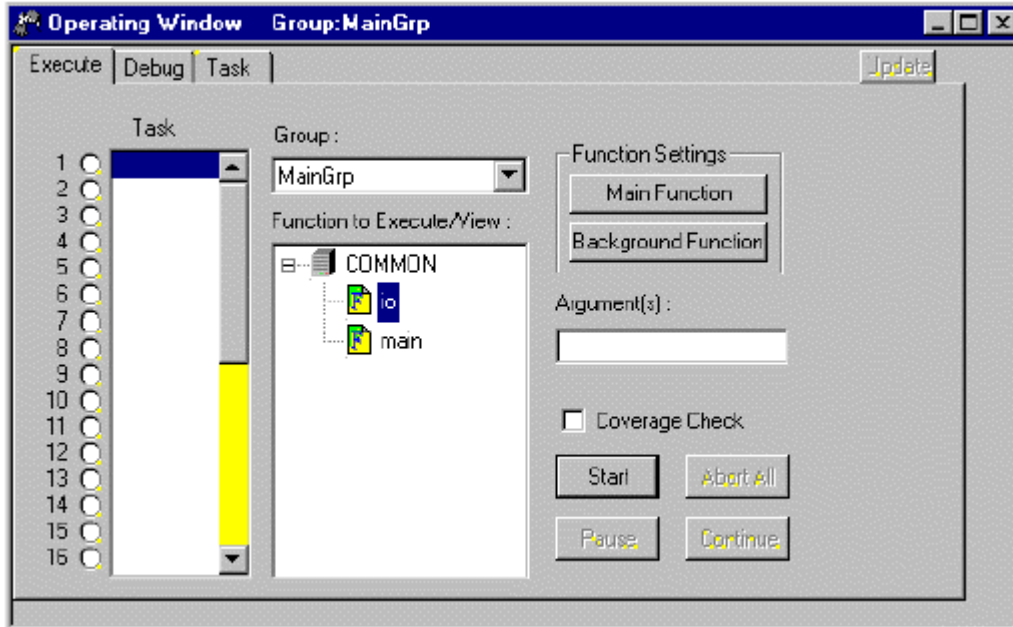
Jetzt sind alle „X“-Markierungen im Quellprogramm-Fenster verschwunden.

### 6.1.6 Bearbeiten von Programmen

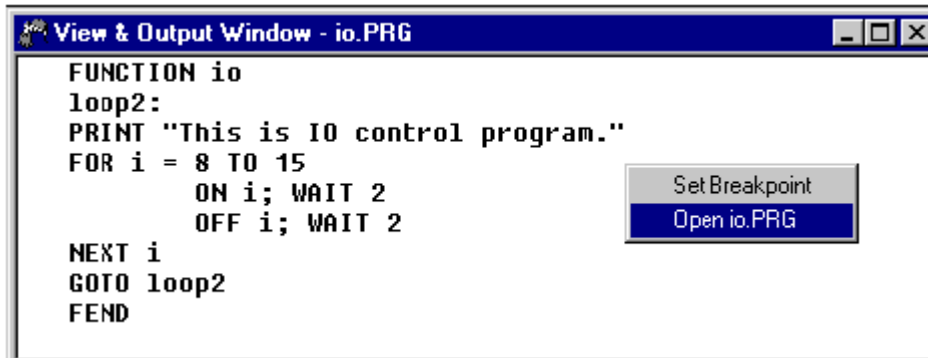
Wenn Sie Programme überarbeiten wollen, klicken Sie auf <Programmier-Menü>, um das Fenster [Project Manager] anzeigen zu lassen. Dann öffnen Sie die Programmdatei.

Ein Programm kann während des Debuggens wie folgt ausgeführt werden:

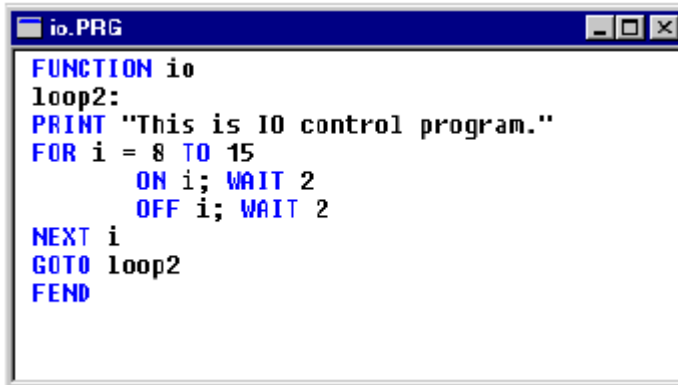
Wählen Sie die zu bearbeitende Funktion aus der Liste [Function to Execute/View] in dem Betriebsfenster aus. Um die zu bearbeitende Funktion auszuwählen, darf sie nicht ausgeführt werden.



Das Programm wird im Quellprogramm-Fenster angezeigt. Klicken Sie mit der rechten Maustaste in die Anzeige und dann mit der linken Maustaste auf „Open io.PRG“ (Öffne Programmname „io.PRG“).



Das Fenster zum Bearbeiten des Programms erscheint.



```

FUNCTION io
loop2:
PRINT "This is IO control program."
FOR i = 8 TO 15
    ON i; WAIT 2
    OFF i; WAIT 2
NEXT i
GOTO loop2
FEND

```

Klicken Sie auf <Close> (Schließen), wenn Sie mit dem Bearbeiten des Programms fertig sind. Speichern Sie das bearbeitete Programm.



```

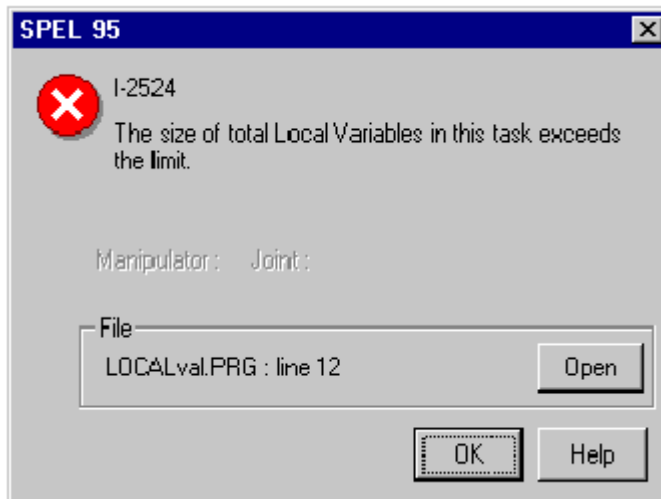
FUNCTION io
loop2:

```

Diese Methode zum Bearbeiten des Programms während des Debuggens wird mit der Zeit sehr nützlich. Das Programm darf zum Bearbeiten weder ausgeführt werden, noch sich im Pausezustand befinden.

### 6.1.7 Fehler 2524

Der Fehler 2524 tritt auf, wenn die Gesamtgröße der lokalen Variablen die zugewiesene Größe (8 KB Standard für jede Task) überschreitet.



Überprüfen Sie den von jeder Task verwendeten Bereich der lokalen Variablen, und sichern Sie dementsprechend ausreichend Platz für diesen Bereich. Weitere Informationen entnehmen Sie dem Kapitel „Variableneinstellungen“.

### 6.1.8 Schalterbedienung zum Ausführen von Programmen

Funktion	Ausführungs-Menü	Debug-Menü
Pause	Pause: Alle Tasks	Halt: ausgewählte Tasks
Abbruch	Alle abbrechen: Alle Tasks	Abbruch: ausgewählte Tasks
Fortsetzen	Fortsetzen: Alle Tasks	Fortsetzen: ausgewählte Tasks
Ausführen	Start nach Auswahl der Funktionen	

### 6.1.9 Fortsetzen der angehaltenen Tasks

#### HINWEIS

Der <Pause>-Schalter im [Execute]-Menü des Betriebsfensters und der <Pause>-Schalter im Bedienfenster werden genauso aktiviert, als wenn eine Eingabe vom REMOTE-Eingang anliegt oder wenn der <Pause>-Befehl in einer der Tasks ausgeführt wurde.

Der <Fortsetzen>-Schalter im [Execute]-Menü des Betriebsfensters und der <Fortsetzen>-Schalter im Bedienfenster setzen die Tätigkeit fort, die zeitweise durch die Betätigung des <Pause>-Schalters (wie oben beschrieben) gestoppt wurde.

Auf der anderen Seite wird der <Halt>-Schalter im [Debug]-Menü des Betriebsfensters so aktiviert, als wenn der HALT-Befehl in einer festgelegten Task ausgeführt wird. Um eine zeitweilig durch diesen Schalter angehaltene Task fortzusetzen, wählen Sie zuerst die Task und klicken dann auf <Resume> (Fortsetzen). Wurde mehr als eine Task angehalten, klicken Sie auf <Resume All>, um alle Tasks gleichzeitig fortzusetzen.

Beachten Sie, dass die Funktion der <Pause>-Schalter irrelevant ist zu <Halt> oder <Resume>. Daher können die Tasks nicht individuell durch den <Resume>-Schalter oder durch den Schalter <Resume All> fortgesetzt werden, falls mehrere Tasks angehalten worden sind.

Mit anderen Worten: Eine Task, die mit Hilfe des <Pause>-Schalters im [Debug]-Menü angehalten worden ist, kann nur mit dem <Continue>-Schalter im [Debug]-Menü fortgesetzt werden.

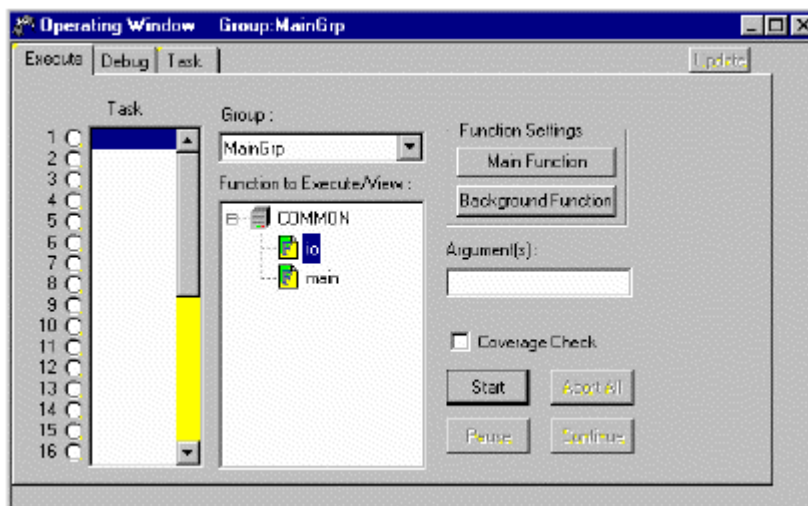
## 6.2 Hauptfunktion

Wird ein Programm aus dem Betriebsfenster ausgeführt, wird zuerst die Hauptfunktion gestartet. Sie wird auch dann zuerst gestartet, wenn das START-Signal vom REMOTE-Eingang anliegt. Es ist wichtig, die Hauptfunktion für jede einzelne Gruppe unabhängig einzurichten.

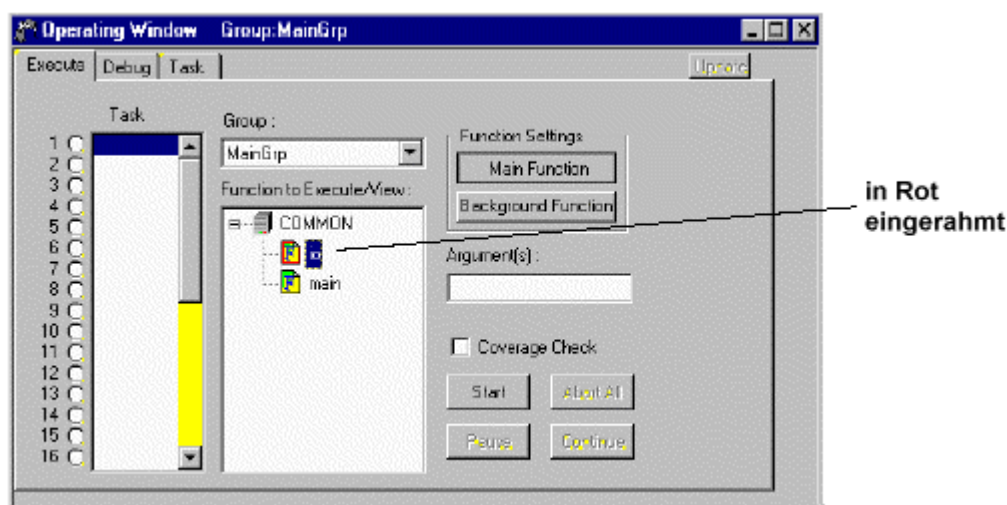
### 6.2.1 Festlegen der Hauptfunktion

Die Hauptfunktion ist in der Liste [Function to Execute/View] rot eingerahmt. Um die Hauptfunktion festzulegen, gehen Sie wie folgt vor:

- (1) Überprüfen Sie, ob die Funktion, die Sie als Hauptfunktion festlegen wollen, tatsächlich in der Liste [Function to Execute/View] vorhanden ist. Ist sie nicht in der Liste, fügen Sie die entsprechende Programmdatei der Gruppe hinzu. Markieren Sie die Datei, die der Liste hinzugefügt werden soll, in der Auswahlliste [Files in the Project] und klicken dann auf <Add>.
- (2) Markieren Sie die Funktion, die Sie als Hauptfunktion festlegen wollen, in der Liste [Function to Execute/View].



- (3) Klicken Sie auf <Main Function> (Hauptfunktion) in der Gruppenauswahl [Function Settings] (Funktionseinstellungen). Die Hauptfunktion wird rot eingerahmt.



## 6.3 Hintergrundfunktion

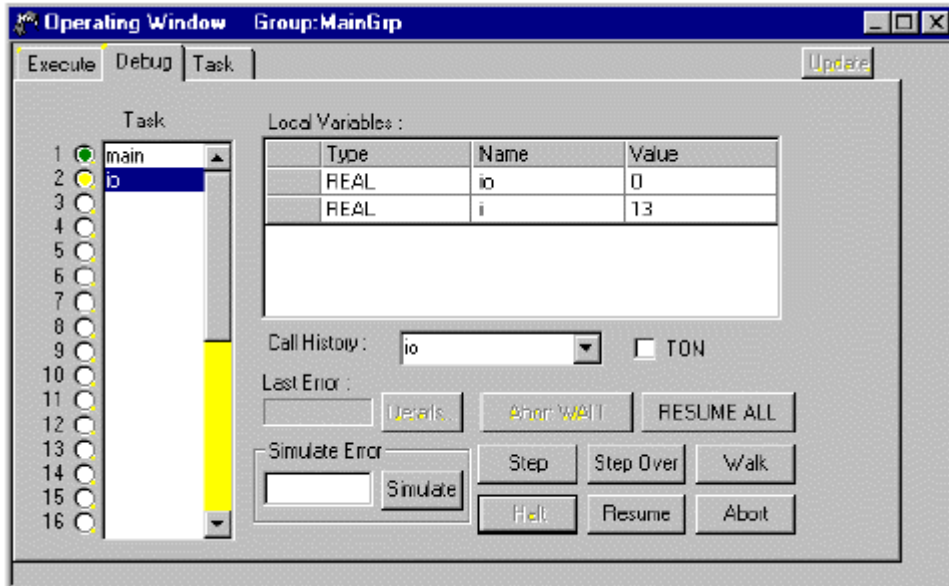
Eine Hintergrundfunktion ist eine Funktion, die dann startet, wenn eine Gruppe geladen wird. Weitere Informationen erhalten Sie in Kapitel 12, im Abschnitt „Hintergrundtask“.



## 6.4 Debuggen einzelner Tasks

### 6.4.1 Debuggen

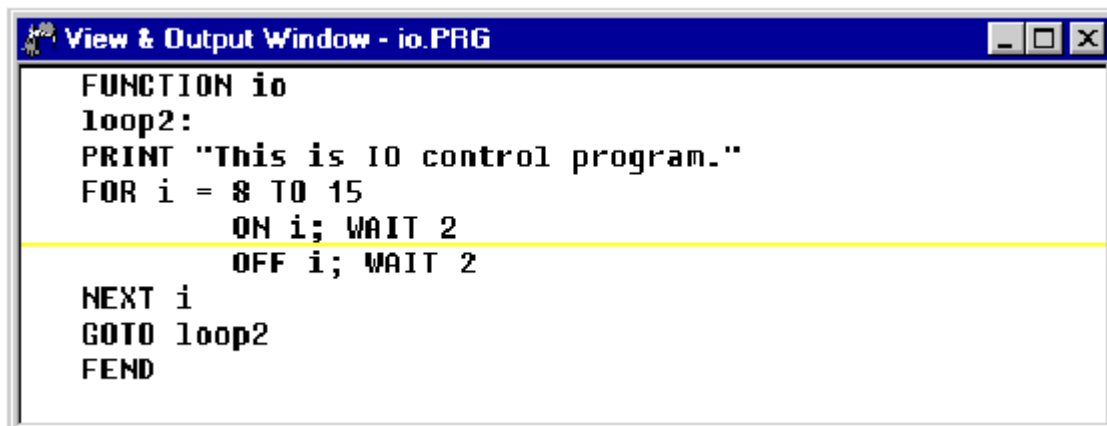
Verwenden Sie den Schalter <Start> im Ausführungsfenster, um Funktionen auszuführen und Tasks zu erzeugen. Benutzen Sie das Menü [Debug] im Betriebsfenster, um einzelne Tasks anzuhalten oder zu beenden.



#### ▪ Anhalten von Tasks

Wenn Sie eine Task vorübergehend anhalten wollen, markieren Sie die entsprechende Task in der Auswahlliste [Task], und klicken Sie auf <Halt>. Die Taskanzeige ändert sich in gelb.

Im Quellprogrammfenster erscheint eine gelbe Linie an der Stelle, wo die Task hält. (Die gelbe Linie erscheint in der Zeile, die als nächstes ausgeführt werden soll.)



#### ▪ Fortsetzen der Tasks

Wenn Sie eine Task weiter ausführen lassen wollen, markieren Sie die entsprechende Task, und klicken Sie auf <Resume> (Fortsetzen). Die Farbe der Taskanzeige ändert sich in grün. Wenn Sie mehr als eine Task fortsetzen wollen, klicken Sie auf <Resume All> (Alle fortsetzen).

- **Tasks abbrechen**

Wenn Sie eine Task abbrechen wollen, markieren Sie die entsprechende Task, und klicken Sie auf <Abort> (Abbrechen). Der Taskname und die Taskanzeige verschwinden aus der Taskliste.

Sie können auch die Hintergrundfunktion abbrechen.

- **Erzwungener Abbruch durch WAIT-Anweisung**

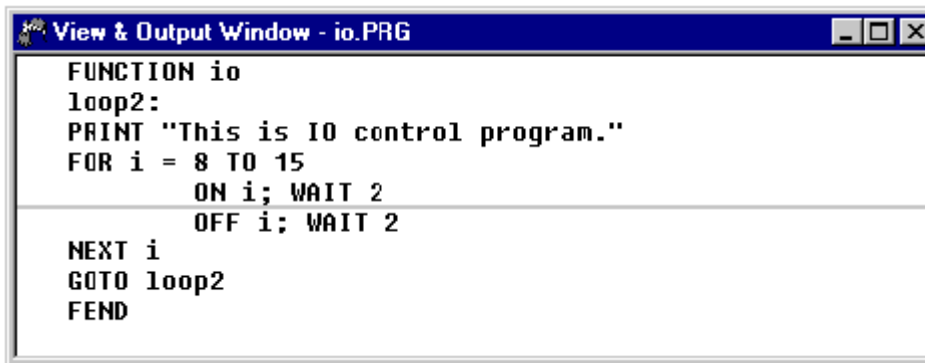
WAIT-Anweisungen ermöglichen den Tasks, die sich im Standby-Modus befinden, den WAIT-Modus gewaltsam zu verlassen und den nächsten Vorgang auszuführen. Markieren Sie die Task, die Sie verlassen möchten, und klicken Sie auf <Abort WAIT> (WAIT-Zustand abbrechen).

- **Laufende Vorgänge**

Tasks werden nach einer Pause ununterbrochen ausgeführt, bis der Manipulator den nächsten Vorgang beendet hat. Wenn Sie diesen Vorgang ausführen wollen, klicken Sie während des Pause-Status auf <Walk> (laufend).

- **Schrittweise Bedienung**

Tasks können zeilenweise ausgeführt werden, um die Bewegungen des Manipulators zu überprüfen. Das Programm pausiert. Dann erscheint eine gelbe Linie im Quellprogramm-Fenster.



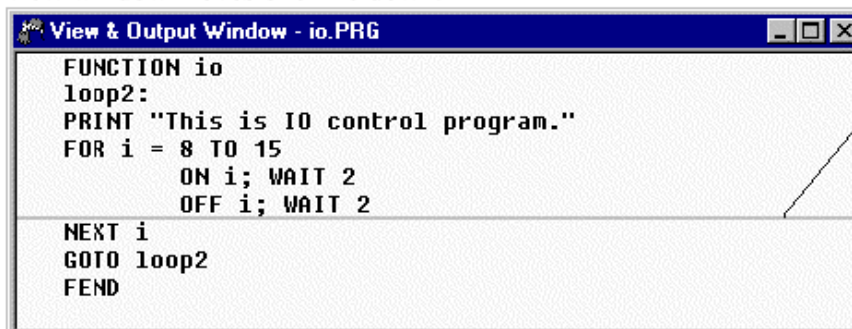
```

FUNCTION io
loop2:
PRINT "This is IO control program."
FOR i = 8 TO 15
    ON i; WAIT 2
    OFF i; WAIT 2
NEXT i
GOTO loop2
FEND

```

Klicken Sie auf <Step> (Schritt) im Betriebsfenster.

Eine Zeile der Anweisungen wird ausgeführt, dann hält das Programm wieder an. Die gelbe Linie im Quellprogramm-Fenster bewegt sich eine Zeile tiefer.



```

FUNCTION io
loop2:
PRINT "This is IO control program."
FOR i = 8 TO 15
    ON i; WAIT 2
    OFF i; WAIT 2
NEXT i
GOTO loop2
FEND

```

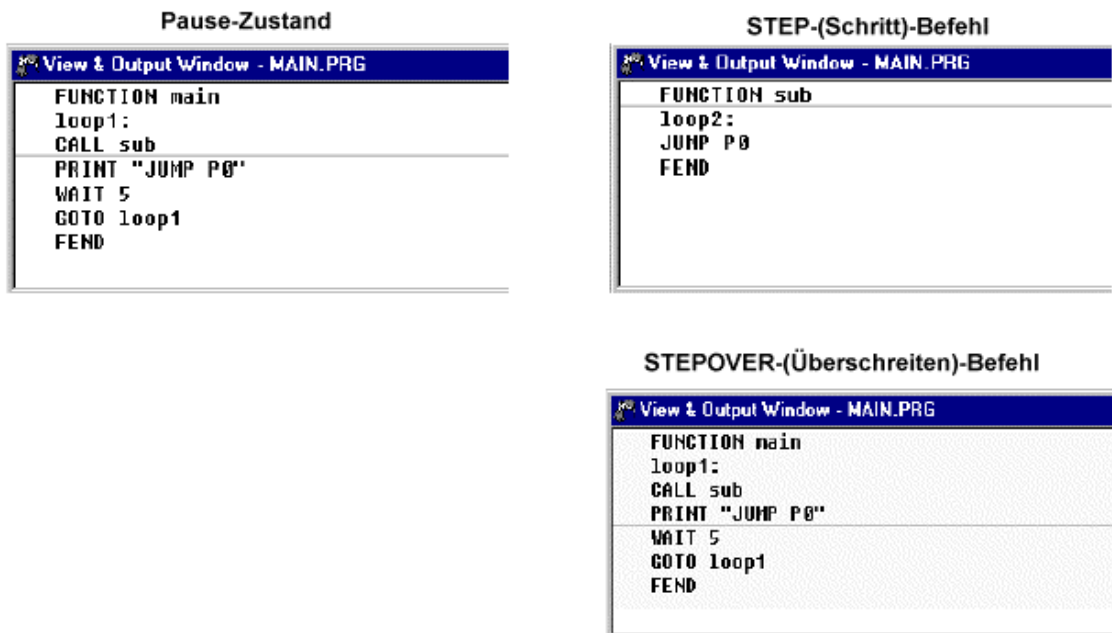
Hat sich eine  
Zeile nach  
unten bewegt.

Zu diesem Zeitpunkt ändern sich die Variablen im Debug-Menü und der Pause-Standort im Task-Menü.  
Wird eine Funktion mit dem CALL-Befehl aufgerufen, wird sie zeilenweise ausgeführt.

#### ▪ STEPOVER

Die Task wird zeilenweise ausgeführt, um jeden einzelnen Vorgang überprüfen zu können und sicherzustellen. Wenn eine andere Funktion durch die CALL-Anweisung festgelegt wurde, wird die aufgerufene Funktion ausgeführt, dann zur Originalroutine zurückgekehrt und danach pausiert.

Der Unterschied zwischen dem schrittweisen und dem Steptover-Verfahren ist folgendermaßen:



Der STEP-Befehl: Das Programm wird in den Pause-Zustand versetzt, nachdem eine Funktion durch die CALL-Anweisung aufgerufen wurde.

Der STEPOVER-Befehl: Das Programm wird in den Pause-Zustand versetzt, nachdem eine durch die CALL-Anweisung aufgerufene Funktion ausgeführt wurde und zur Originalroutine zurückgekehrt wurde.

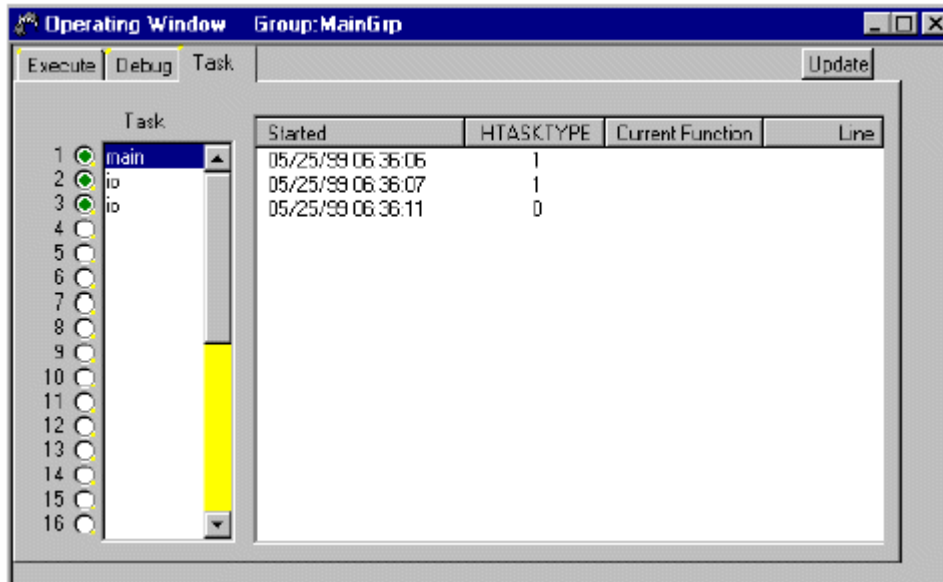
Das Debuggen wird effektiver, wenn Sie zuerst die Funktionen, die aufgerufen werden sollen, debuggen lassen.

### 6.4.2 Debuggen einer Task, die programmiert wurde nicht anzuhalten.

Eine Task kann auch dann angehalten werden, wenn sie per HTASKTYPE-Befehl so programmiert wurde, dass sie nicht anhalten soll. Auch eine so programmierte Hintergrund-Task kann angehalten werden. Dafür verwenden Sie das [Debug]-Menü-Verfahren.

Ob die Task so eingestellt ist, dass sie nicht pausieren soll, kann im [Task]-Menü bestätigt werden.

Der aktuelle Zustand wird in der Reihe HTASKTYPE angezeigt.



### 6.4.3 Linien in dem Quellprogramm-Fenster

Wenn Sie aus der [Task]-Liste des Betriebsfensters eine Task auswählen, erscheint eine Programmliste in dem Quellprogramm-Fenster. Wenn das Programm aus irgendeinem Grund anhält, erscheinen farbige Linien in der Liste - abhängig vom Zustand des Programms.

**Programm-Pause:** Eine gelbe Linie erscheint dort, wo das Programm pausiert.

**WAIT:** Eine grüne Linie erscheint dort, wo das Programm aufgrund eines WAIT-Befehls angehalten wurde.

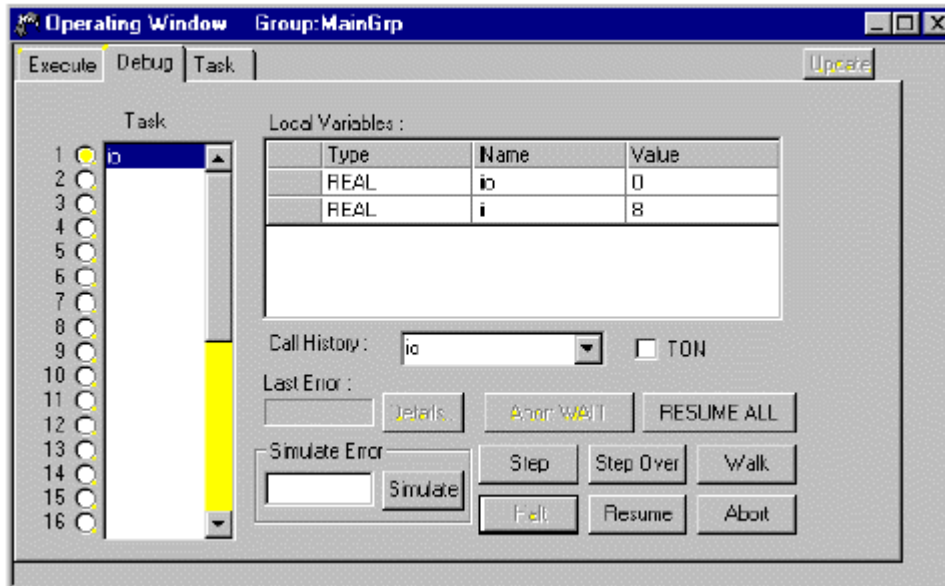
**Quick Pause:** Eine dunkelgelbe Linie erscheint, wenn das System auf eine Tätigkeit des Manipulators wartet.

Wurde ein Unterbrechungspunkt eingerichtet, erscheint eine dunkelbraune Linie.

Wenn ein Kontrollbericht erstellt werden soll, erscheint ein „X“ an der linken Seite der ausgeführten Zeile.

### 6.4.4 Zuordnen und Updaten lokaler Variablen

Die Typen, Namen und Werte der Variablen während einer Pause erscheinen in der Auswahl [Local Variables] (Lokale Variablen) im [Debug]-Menü.



Wenn Sie auf die Zeile doppelklicken, in der der Wert steht, den Sie ändern möchten, kann die Variable verändert werden. Während das Programm ausgeführt oder gestoppt wird, werden die Variablen nicht angezeigt.

### 6.4.5 Bezug zum Aufruf-Protokoll

Das Aufruf-Protokoll der angehaltenen Tasks wird angezeigt.

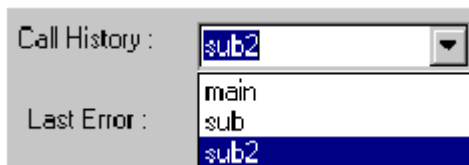
Wenn verschiedene Funktionen als Unterrouinen von Aufrufen, durch CALL-Anweisungen oder Anwenderfunktionen verwendet werden, werden die Funktionsebenen im PAUSE-Zustand angezeigt.

Wenn Sie auf den Pfeil in der Liste [Call History] (Aufruf-Protokoll) klicken, erscheinen die Funktionsebenen.

<Beispiel>

```
FUNCTION Main
  :
  CALL Sub
  .
FEND
FUNCTION Sub
  .
  CALL Sub2      } Funktion Sub
  :
FEND
FUNCTION Sub2    } Funktion Sub2
  :
FEND
```

Während einer Pause aufgrund der Sub2-Funktion wird der unten dargestellte Hinweis in der Liste [Call History] dargestellt.



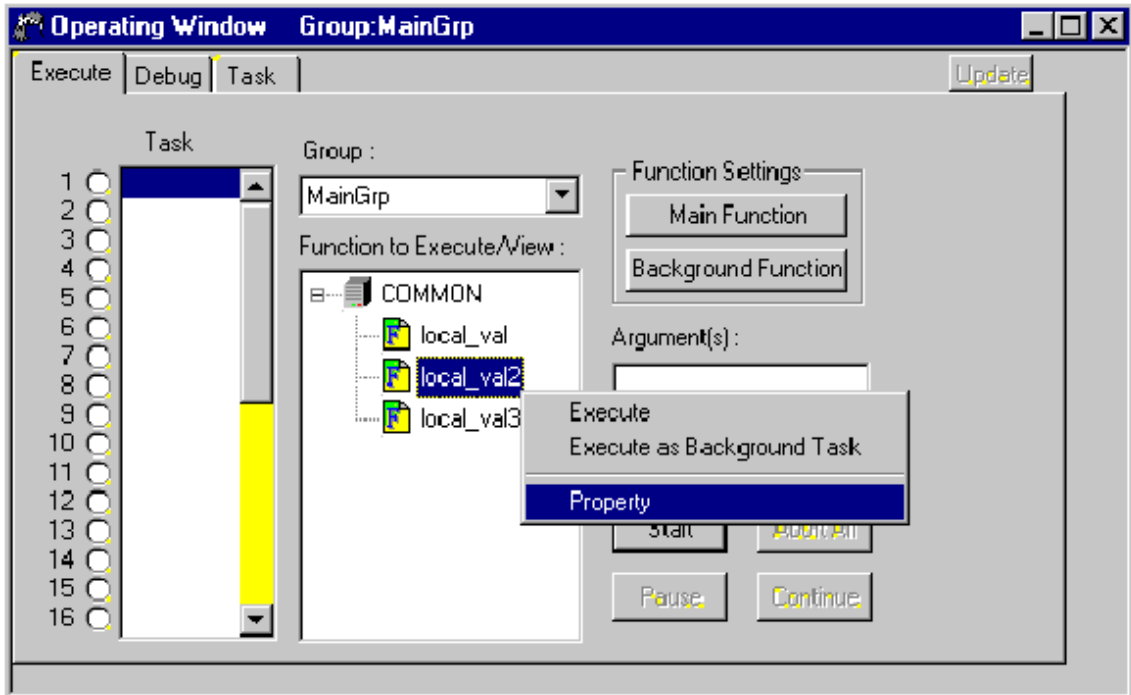
## 6.5 Variableneinstellungen

### 6.5.1 Lokale Variablen

- **Überprüfen des zugewiesenen Bereiches der lokalen Variablen**

Um den zugewiesenen Bereich für die lokalen Variablen der ausgewählten Funktion zu finden: Wählen Sie die Funktion aus der Liste [Function to Execute/View] im Ausführungs-Menü des Betriebsfensters.

Klicken Sie dann mit der rechten Maustaste auf die Funktion, um „Property“ (Eigenschaft) zu öffnen.



Der Dialog [Function Property] (Funktionseigenschaft) erscheint. Er zeigt die zugewiesene Größe für die lokalen Variablen der ausgewählten Funktion.



Jede Task hat einen zugewiesenen Bereich für die lokalen Variablen. Verwenden Sie den Befehl DIMAREA, um den zugewiesenen Bereich für die lokalen Variablen ausfindig zu machen.

Geben Sie „DIMAREA“ in das Textfeld [Command] (Befehl) ein, und klicken Sie auf <Execute> (Ausführen). Der aktuelle Bereich für die lokalen Variablen jeder einzelnen Task wird angezeigt.

TASK	PEAK	NOW(free)
01	7980	7980( 211)
02	7076	7076( 1115)
03	7332	7332( 859)
04	0	0( 8191)
05	0	0( 8191)
06	0	0( 8191)
07	0	0( 8191)
08	0	0( 8191)
09	0	0( 8191)
10	0	0( 8191)
11	0	0( 8191)
12	0	0( 8191)
13	0	0( 8191)
14	0	0( 8191)
15	0	0( 8191)
16	0	0( 8191)

In der dargestellten Liste bedeuten die Überschriften jeder Spalte folgendes:

TASK: Tasknummer

PEAK: der maximal zu verwendende Bereich

NOW: der aktuell von den lokalen Variablen verwendete Bereich

FREE: der verbleibende Platz

Eine CALL-Anweisung verwendet den Bereich der lokalen Variablen innerhalb der Task.

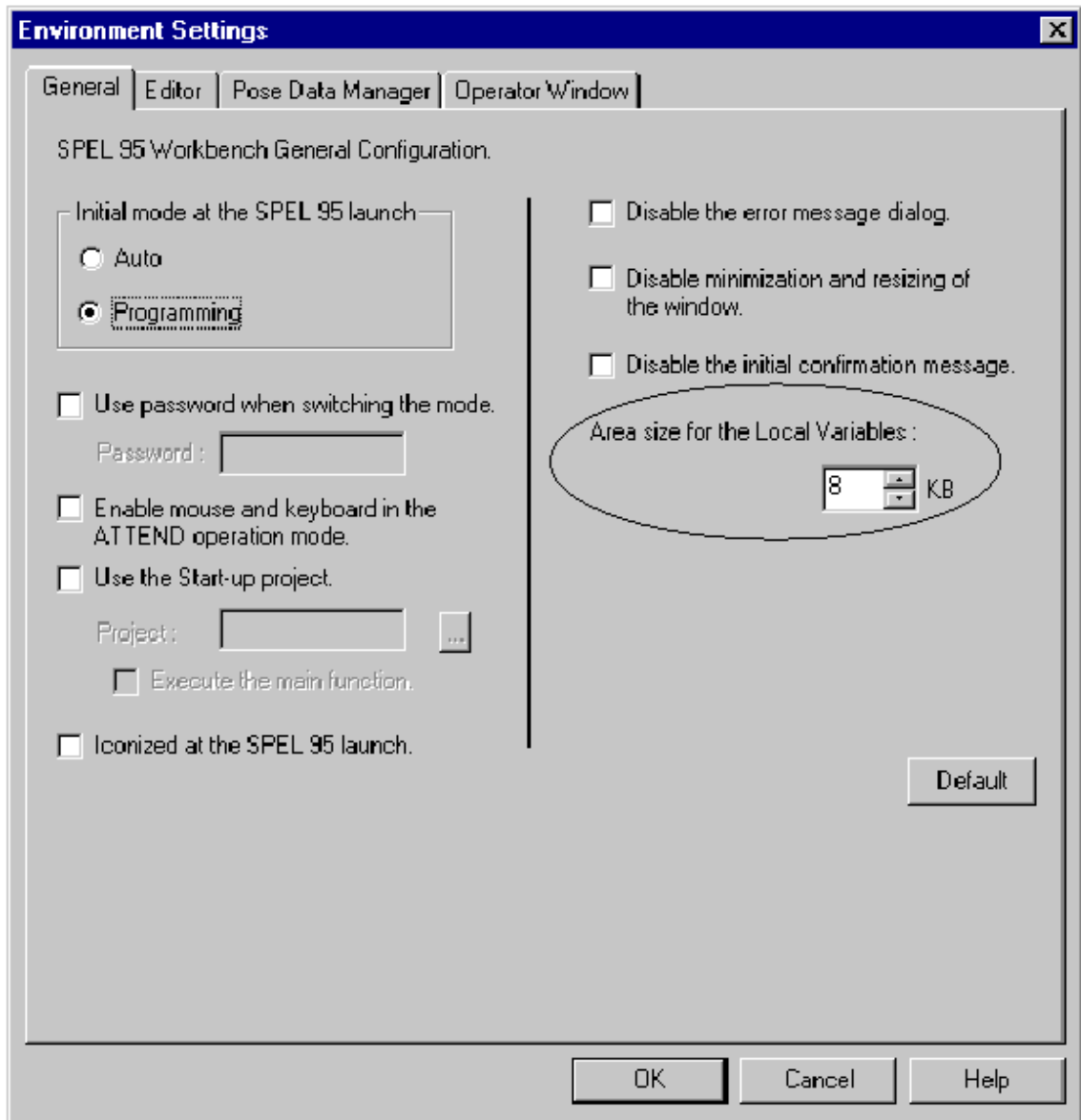
In dem oben dargestellten Beispiel ergibt sich ein Bereich für die lokalen Variablen von 15056 Byte innerhalb der TASK01, wenn eine Funktion aus der Task02 von der TASK01 aufgerufen wird. Das überschreitet den zugewiesenen Bereich von 8191 Byte und verursacht den Fehler 2524. In diesem Fall muss der Bereich der lokalen Variablen in TASK01 erhöht werden.

Wenn erwartet wird, dass der Bereich der lokalen Variablen durch die CALL-Anweisungen, die andere Funktionen aufrufen, verwendet werden soll, stellen Sie vorher sicher, dass genügend Platz zur Verfügung steht.



- **Einstellen des zugewiesenen Bereiches für die lokalen Variablen**

Führen Sie [Setup] – [Environment] (Einstellen – Umgebung) aus, und klicken Sie auf „General“ (Allgemein).



Der zugewiesene Bereich der lokalen Variablen, der jeder Task zugeteilt wird, kann in der Auswahlliste [Area size for the local variables] (Bereichsgröße für die lokalen Variablen) bearbeitet werden. Die Standardgröße beträgt 8 KB.

Klicken Sie auf <OK>, wenn Sie die gewünschten Änderungen vorgenommen haben. Dann werden die neuen Einstellungen wirksam.

Falls es notwendig sein sollte, erscheint ein Dialog, der Sie darauf hinweist, dass SPEL 95 neu gestartet werden muss, damit die neuen Einstellungen wirksam werden.

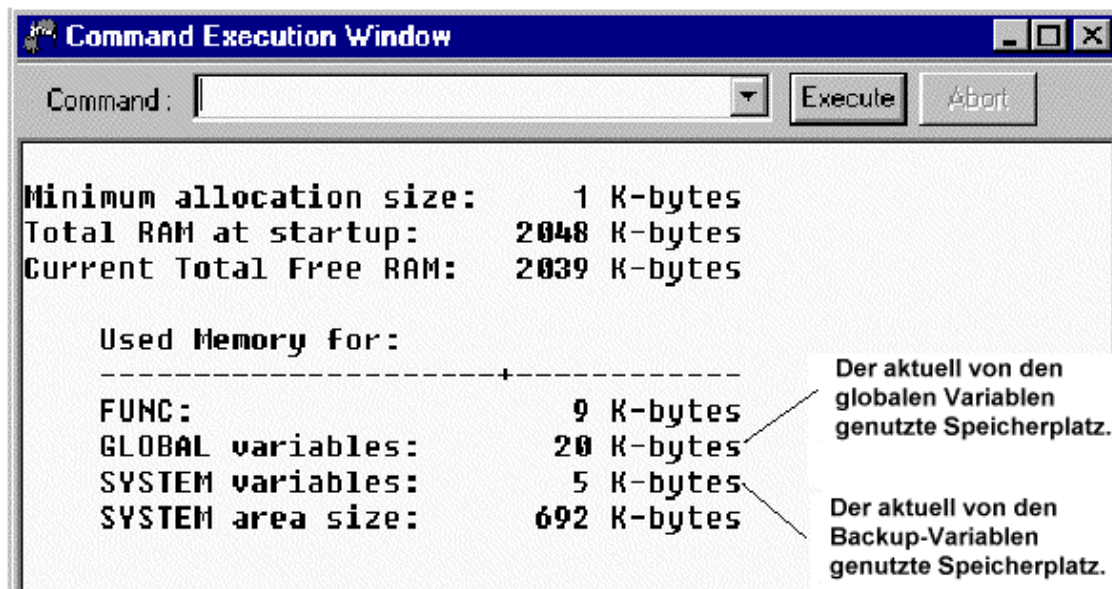
Die neuen Einstellungen werden dann erst nach einem Neustart von SPEL 95 wirksam.

## 6.5.2 Globale und Backup-Variablen

### ▪ Überprüfen des zugewiesenen Bereiches der globalen und Backup-Variablen

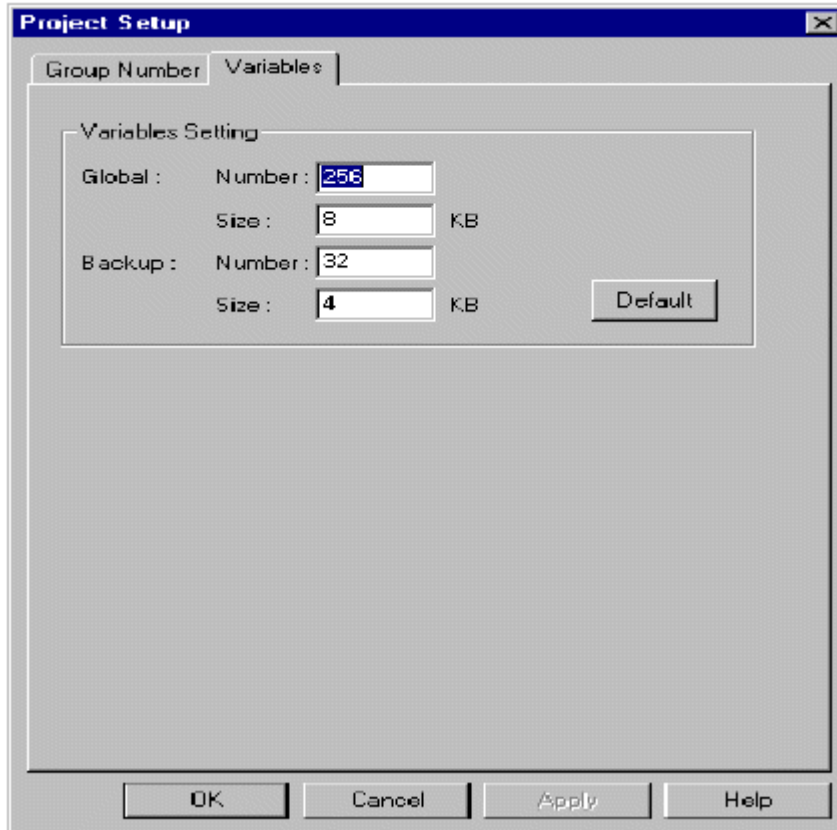
Mit Hilfe des Befehls FREE lässt sich herausfinden, wieviel Speicher der Steuerung gegenwärtig von den globalen und Backup-Variablen verwendet wird.

Geben Sie „FREE“ in das Textfeld [Command] (Befehl) ein, und klicken Sie auf <Execute> (Ausführen). Der aktuelle Bereich für die globalen und Backup-Variablen wird angezeigt.



- **Einstellen des zugewiesenen Bereiches für die globalen und Backup-Variablen**

Führen Sie [Project] – [Group Number] (Projekt – Gruppennummer) aus, und klicken Sie auf „Variables“ (Variablen), um die Anzahl oder den zugewiesenen Bereich für die globalen und Backup-Variablen zu bearbeiten.



In diesem Menü werden Ihnen Textfelder zur Verfügung gestellt, um die Anzahl und den zugewiesenen Bereich der globalen und Backup-Variablen einzustellen.

Klicken Sie auf <Default>, um die Standardeinstellungen wiederherzustellen.

Nachfolgende Tabelle beinhaltet die Standard- und die maximalen Werte der Variablen:

		Standard	Maximum
Globale Variablen	Anzahl	256	32767
	Größe	8 KB	4096 KB
Backup-Variablen	Anzahl	32	32767
	Größe	4 KB	4096 KB

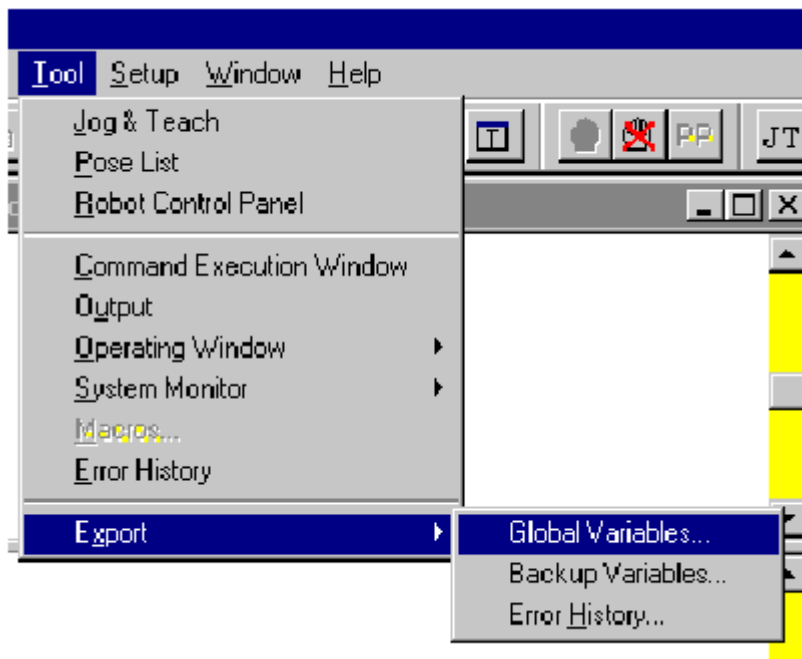
Nach vollständiger Eingabe der Daten klicken Sie zur Bestätigung auf <OK>. Zu diesem Zeitpunkt werden alle vorhandenen globalen und Backup-Variablen in allen Projekten gelöscht. Folglich müssen Sie anschließend das Wiederregistrierungsprogramm ausführen!

### 6.5.3 Backup und Wiederherstellen

Das Backup der globalen und Backup-Variablen wird im Programmformat gespeichert. Um die globalen und Backup-Variablen wiederherzustellen, importieren Sie das Programm, das als Backup gespeichert wurde, und führen Sie es aus.

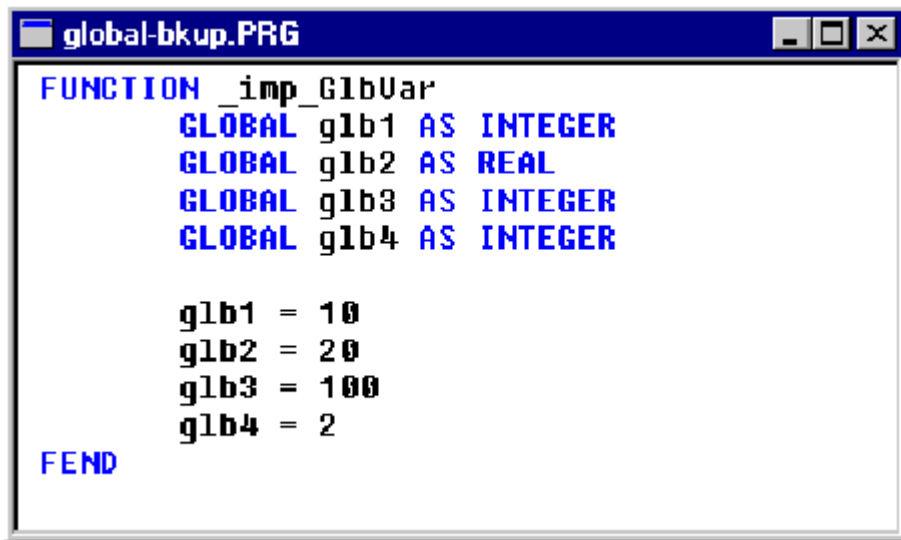
- **Speichern eines Backups der globalen und Backup-Variablen**

Führen Sie [Tool] – [Export] – [Global Variables] (Werkzeug – Export – Globale Variablen) aus.



Wählen Sie den Dateinamen und das Verzeichnis, um ein Backup-Programm für eine globale Variable zu erzeugen.

Ein Backup-Programm für eine globale Variable sollte wie folgt aussehen:

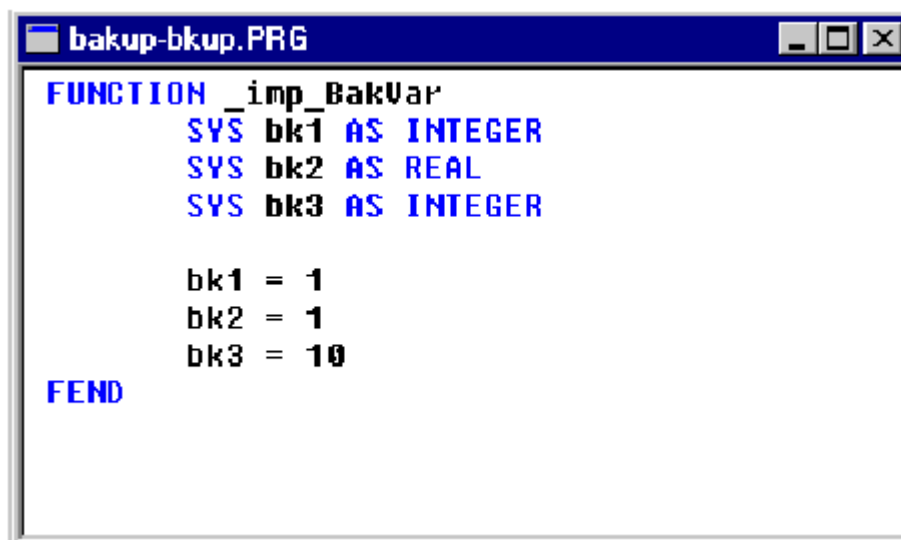


```
FUNCTION _imp_GlbVar
    GLOBAL glb1 AS INTEGER
    GLOBAL glb2 AS REAL
    GLOBAL glb3 AS INTEGER
    GLOBAL glb4 AS INTEGER

    glb1 = 10
    glb2 = 20
    glb3 = 100
    glb4 = 2

FEND
```

Um ein Backup-Programm für eine globale Variable zu erzeugen, führen Sie den Befehl [Tool] – [Export] – [Global Variables] (Werkzeug – Export – Globale Variablen) aus, und speichern Sie die erzeugte Datei backup-bkup.PRG.



```
FUNCTION _imp_BakVar
    SYS bk1 AS INTEGER
    SYS bk2 AS REAL
    SYS bk3 AS INTEGER

    bk1 = 1
    bk2 = 1
    bk3 = 10

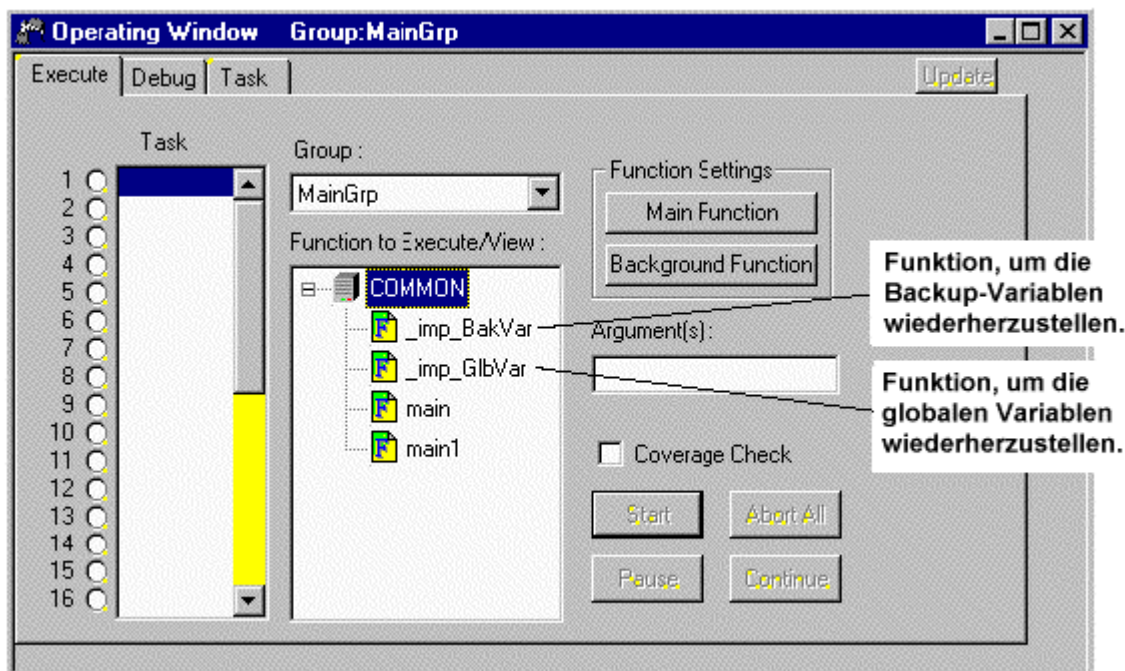
FEND
```

Der Typ der Variablen und deren Wert werden im Programmformat gespeichert.

### ▪ Wiederherstellen der globalen/Backup-Variablen

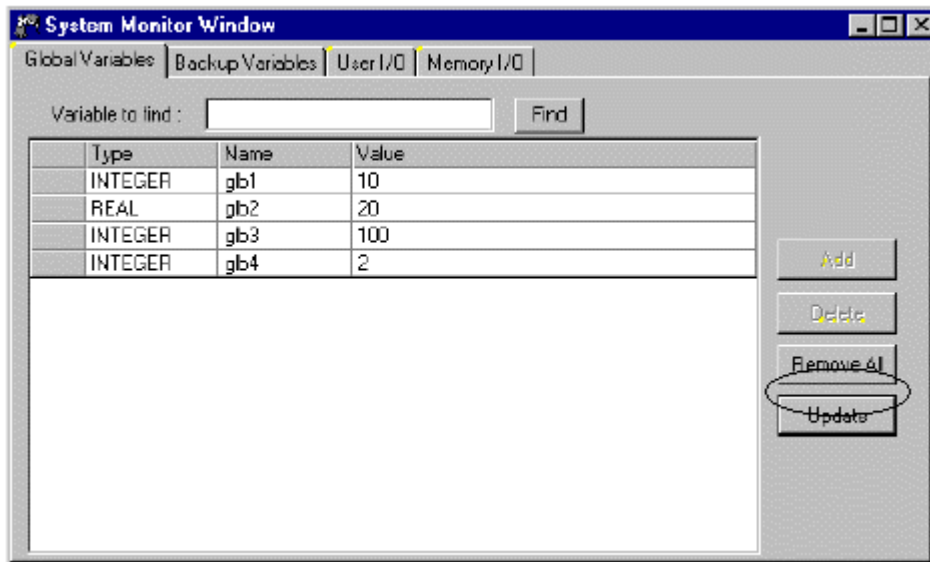
Die globalen/Backup-Variablen werden im Programmformat gespeichert. Um eine Backup-Programmdatei wiederherzustellen, gehen Sie wie folgt vor (genau der gleiche Weg wie beim Importieren anderer Programmdateien):

- (1) Öffnen Sie das wiederherzustellende Projekt.
- (2) Öffnen Sie den Programm-Manager.
- (3) Führen Sie den Befehl [File] – [Import] (Datei – Import) aus, um die Backup-Programmdatei der Variablen zu importieren.
- (4) Die importierte Programmdatei wird der Liste [Files in the project] hinzugefügt. Fügen Sie dann die Datei der Gruppe zu.
- (5) Klicken Sie auf <Debug>-Menü. Öffnen Sie das [Execute]-Menü (Ausführen-Menü) im Betriebsfenster.
- (6) Die Funktion zum Wiederherstellen wird in der Liste [Function to Execute/View] angezeigt.



Erscheint die Funktion nicht in dieser Liste, wählen Sie die hinzugefügte Gruppe aus.

- (7) Wählen Sie die Funktion, und klicken Sie auf <Start>. Das festgelegte Programm wird ausgeführt, und die enthaltenen Variablen werden in den Speicher der Steuerung geladen. Wählen Sie das Menü globale Variablen oder Backup-Variablen, und klicken Sie auf <Update>, um sicherzustellen, dass die Variablen der importierten Datei geladen werden.



## 6.6 Unterbrechungspunkte

Eine Task pausiert, nachdem das Programm ausgeführt worden ist, an einem Unterbrechungspunkt, der an irgendeiner Stelle des Programms eingefügt worden ist.

### ▪ Einrichten von Unterbrechungspunkten



<Set/Release a Breakpoint>  
(Setzen/Freigeben eines Unterbrechungspunktes)

Klicken Sie auf [View & Output Window] (Anzeige- und Ausgabefenster).

Bewegen Sie den Cursor an die Stellen, an denen Sie einen Unterbrechungspunkt einrichten möchten.

Klicken Sie auf <Set/Release a Breakpoint> (Einrichten/Freigeben eines Unterbrechungspunktes).

Oder klicken Sie mit der rechten Maustaste. Das Auswahlmenü wird geöffnet. Wählen Sie dann „Set Breakpoint“ (Einrichten Unterbrechungspunkt) aus.

Ein Unterbrechungspunkt wird in der Zeile eingerichtet, in der sich der Cursor befindet:

```

FUNCTION io
loop2:
PRINT "This is IO control program."
FOR i = 8 TO 15
    ON i; WAIT 2
    OFF i; WAIT 2
NEXT i
GOTO loop2
FEND
  
```

Wenn Sie die Task einmal unterbrechen und erneut ausführen, wird die Funktion von Anfang an ausgeführt und hält genau vor der Unterbrechungspunktlinie.

### ▪ Entfernen von Unterbrechungspunkten

Wenn Sie erneut auf <Set/Release a Breakpoint> (Einrichten/Freigeben eines Unterbrechungspunktes) klicken, wird der Unterbrechungspunkt entfernt.



<Clear All Breakpoints>  
(Entfernen aller Unterbrechungspunkte)

### ▪ Entfernen aller Unterbrechungspunkte

Klicken Sie auf <Clear all Breakpoints> (Entfernen aller Unterbrechungspunkte), werden alle Unterbrechungspunkte entfernt.

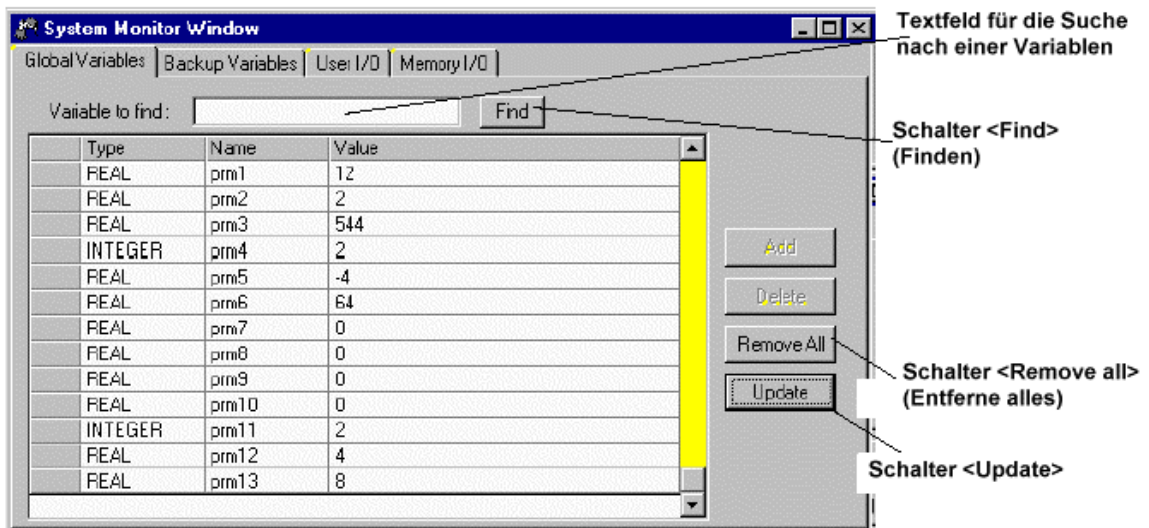


## 6.7 Zuordnen und Ändern mit dem System-Monitor-Fenster

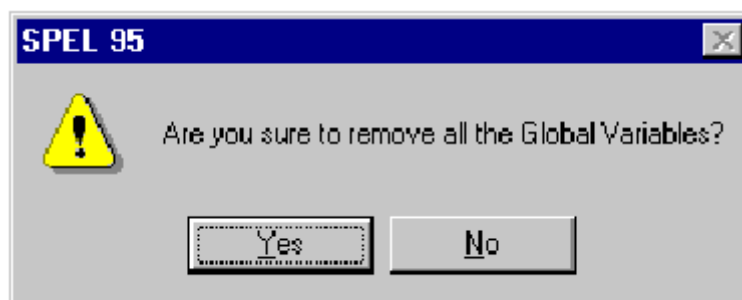
Das System-Monitor-Fenster enthält verschiedene Menüs. Sie können jedes Menü verwenden, indem Sie auf den entsprechenden Reiter klicken.

Menü	Beschreibung
[Globale Variablen]:	Variablen, die von verschiedenen Tasks gemeinsam benutzt werden können.
[Backup-Variablen]:	Variablen, die von verschiedenen Tasks gemeinsam benutzt werden können und nach dem Ausschalten der Spannungszufuhr beibehalten werden.
[Anwender E/A]:	E/As, die mit externen Geräten synchron laufen.
[Speicher E/A]:	E/As, die mit externen Geräten synchron laufen und die im Speicher der Steuerung zur Verfügung stehen.

### 6.7.1 [Globale Variablen]-Menü



- <Update>-Schalter: Verweist auf die aktuelle Variablen.
- <Remove all>-Schalter: Entfernt alle aktuell eingestellten Variablen. Wenn Sie auf diesen Schalter klicken, erscheint nachfolgende Meldung:



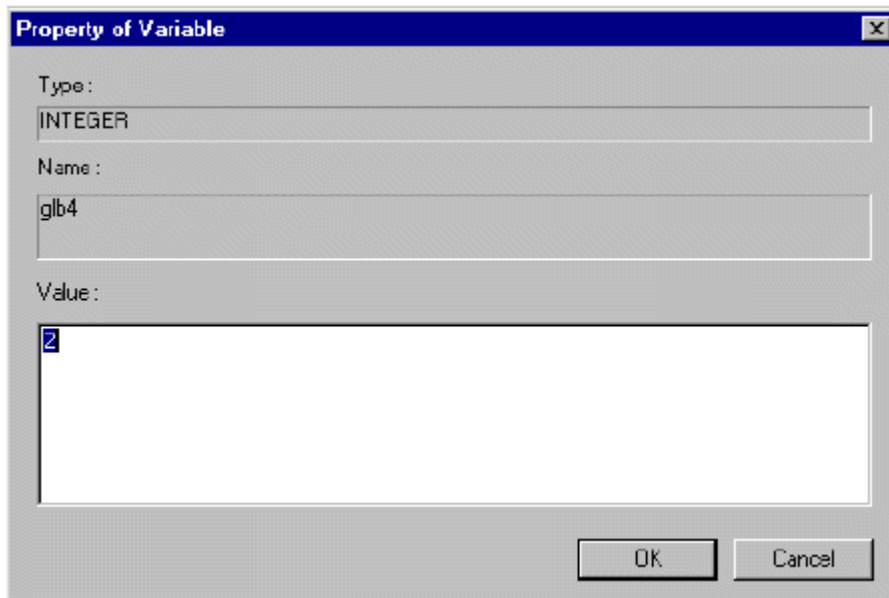
Klicken Sie auf <Yes>, um alle globalen Variablen zu entfernen.

[Variable to find]- Geben Sie in dem Textfeld den Namen der Variablen ein, nach der Sie suchen  
Textfeld und <Find>- möchten, und klicken Sie dann auf <Find>.

Schalter: Wird eine Variable gefunden, wird sie am Bildschirm markiert.

Doppelklicken Sie in der Spalte [Value] (Wert) auf eine Variable, die Sie ändern möchten.

Wenn Sie mit der rechten Maustaste klicken, erscheint ein Menü. Klicken Sie in diesem Menü auf  
„Property“ (Eigenschaft), um die Eigenschaft der Variablen anzeigen zu lassen.



The screenshot shows a dialog box titled "Property of Variable". It has a blue title bar with a close button (X). The dialog contains three input fields: "Type:" with the value "INTEGER", "Name:" with the value "glb4", and "Value:" with the value "2". At the bottom of the dialog are two buttons: "OK" and "Cancel".

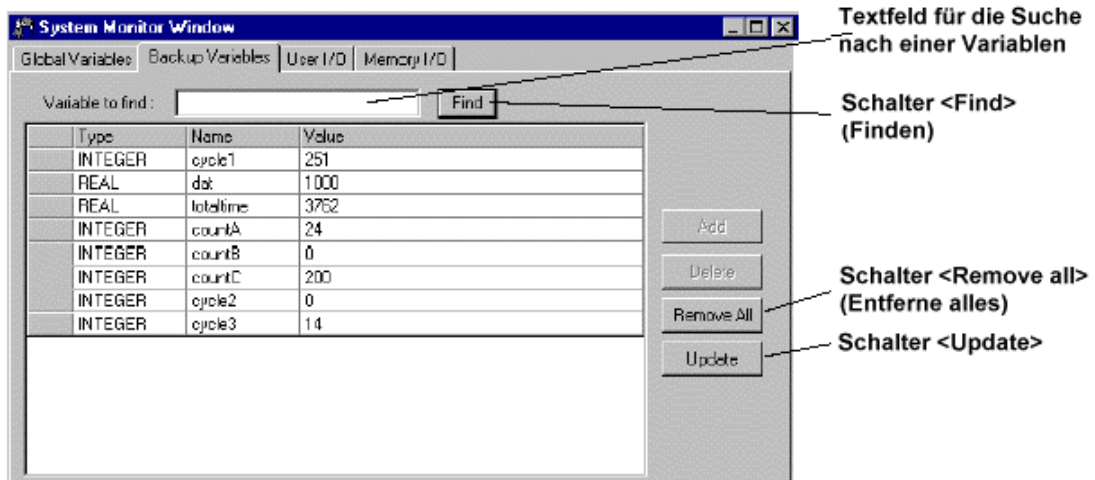
Jetzt kann die Variable geändert werden.



#### ACHTUNG

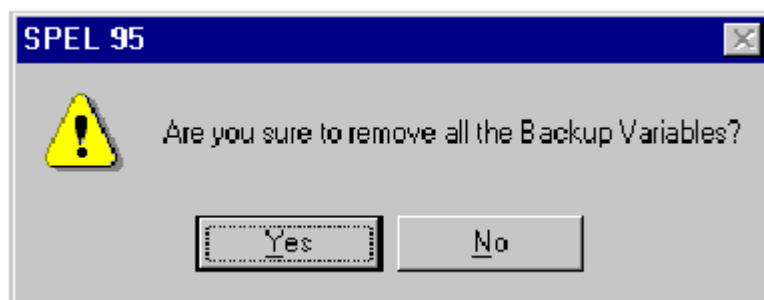
Wenn Sie eine globale Variable während der Ausführung einer Task ändern, kann sich die Task in unerwarteter Weise verhalten, und externe Geräte könnten beschädigt werden!

## 6.7.2 [Backup-Variablen]-Menü



<Update>-Schalter: Zeigt die Werte der aktuellen Variablen.

<Remove all>-Schalter: Entfernt alle aktuell eingestellten Variablen. Wenn Sie auf diesen Schalter klicken, erscheint nachfolgende Meldung:



Klicken Sie auf <Yes>, um alle Backup-Variablen zu entfernen.

[Variable to find]- Geben Sie in dem Textfeld den Namen der Variablen ein, nach der Sie suchen  
Textfeld und <Find>- möchten, und klicken Sie dann auf <Find>.

Schalter: Wird eine Variable gefunden, wird sie am Bildschirm markiert.

Doppelklicken Sie in der Spalte [Value] (Wert) auf eine Variable, die Sie ändern möchten.

Wenn Sie mit der rechten Maustaste klicken, erscheint ein Menü. Klicken Sie in diesem Menü auf  
„Property“ (Eigenschaft), um die Eigenschaft der Variablen anzeigen zu lassen.

Jetzt kann die Variable geändert werden.



**ACHTUNG**

Wenn Sie eine Backup-Variable während der Ausführung einer Task ändern, kann sich die Task in unerwarteter Weise verhalten, und externe Geräte könnten beschädigt werden!

### 6.7.3 [Anwender E/A]-Menü

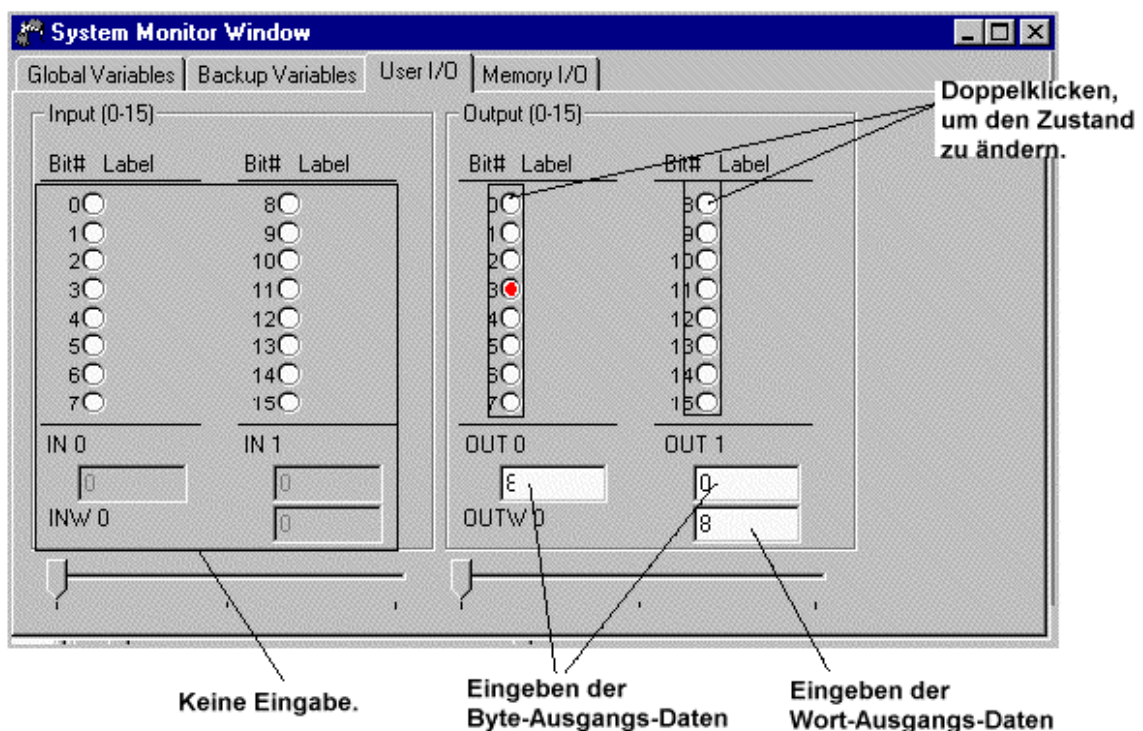
Bezug auf und Ändern von Ausgabeausgängen:

In der Auswahlbox [Output] können Sie sich auf den Zustand der Ausgabeausgänge beziehen und Sie am Bildschirm bearbeiten.

- Um den Zustand der Ausgabeausgänge zu steuern, geben Sie die Byte-Ausgangsdaten zwischen 0 und 255 in dem Feld <OUT#> ein.
- Wenn Sie den Zustand der Ausgabeausgänge in Biteinheiten eingeben wollen, bewegen Sie den Cursor zu dem Schalter der Nummer, und doppelklicken Sie dann auf den Schalter.

Bezug auf Eingabeeingänge:

In der Auswahlbox [Input] können Sie sich auf den Zustand der Eingabeeingänge beziehen. Deren Werte können geändert werden.



#### ACHTUNG

Werden Ausgabeausgänge geändert, können sich externe Geräte, die an diesen Ausgabeausgängen angeschlossen sind, unerwartet verhalten. Treffen Sie Vorsichtsmaßnahmen für einen sicheren Gebrauch!  
Auch eine unkorrekte Verdrahtung kann unnormales Verhalten verursachen.

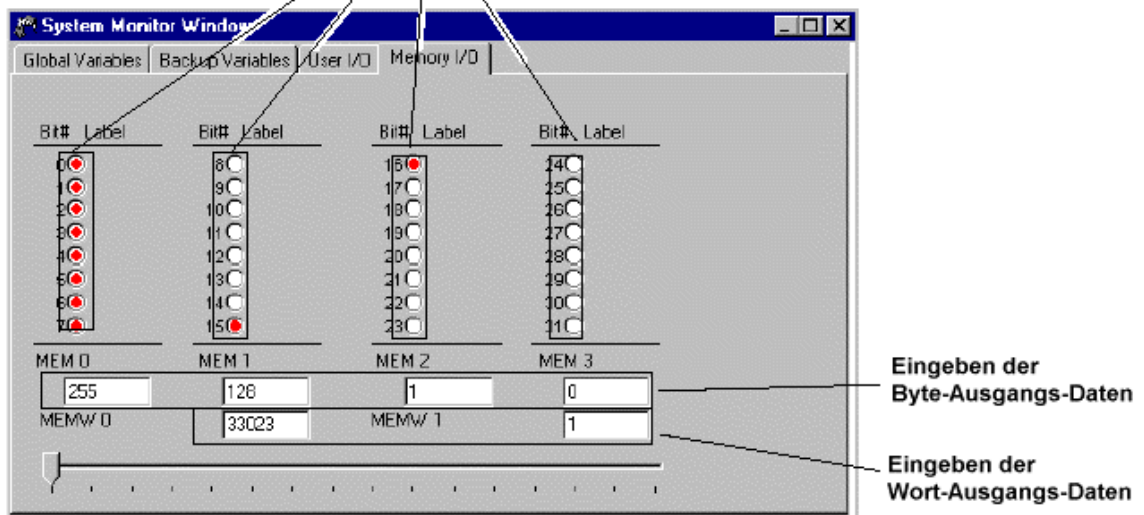
## 6.7.4 [Speicher E/A]-Menü

Sie können den Zustand der Speicher E/A am Bildschirm ändern.

- Um den Zustand der Ausgänge zu steuern, geben Sie die Byte-Ausgangsdaten zwischen 0 und 255 in dem Feld <MEM#> ein.
- Wenn Sie den Zustand der Ausgabeausgänge in Biteinheiten eingeben wollen, bewegen Sie den Cursor zu dem Schalter der Nummer, und doppelklicken Sie dann auf den Schalter.

Wenn die Speicher-E/A durch Tasks gesteuert werden, lassen sich die Werte möglicherweise nicht ändern.

**Doppelklicken, um den Zustand zu ändern.**



## 6.8 Fehlerprotokoll

### 6.8.1 Das Fenster [Fehlerprotokoll]

**EH**

In diesem Fenster werden detaillierte Informationen zum Fehlerprotokoll angezeigt. Klicken Sie in der Werkzeugleiste auf <Error History>.

<Error History>  
(Fehlerprotokoll)

Error Code	Message	Manipu.	Joint	Date/Time	Unit	Function
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:56:19	COMMON	localvat:12
T-4102	The Drive Unit power is not turned on.	1	0	06/04/99 04:56:06	MONITOR:0	
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:56:02	COMMON	localvat:8
T-4102	The Drive Unit power is not turned on.	1	0	06/04/99 04:55:50	MONITOR:0	
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:55:44	COMMON	localvat:2
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:55:31	COMMON	localvat:2
T-4102	The Drive Unit power is not turned on.	1	0	06/04/99 04:54:06	MONITOR:0	
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:53:56	COMMON	localvat:2
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:52:58	MONITOR:0	
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:52:39	MONITOR:0	
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:51:11	MONITOR:0	
I-2006	The arguments of the called function do not m...	0	0	06/04/99 04:26:50	MONITOR:0	
T-4102	The Drive Unit power is not turned on.	0	0	06/04/99 03:23:24	WB	InitIP:0
T-4102	The Drive Unit power is not turned on.	0	0	06/04/99 03:23:11	WB	InitIP:0
I-2323	The specified manipulator is executing a Paral...	0	1	06/04/99 02:52:06	COMMON	_main:2:8
U-1598	Construction Error: Syntax error.	0	0	06/03/99 02:29:35	MONITOR:0	
M-3201	The Manipulator is being used in another task.	0	0	06/03/99 11:13:22	COMMON	main:8
T-4001	The Arm reached the limit of motion range.	1	2	06/03/99 09:24:52	COMMON	main:7
T-4007	The Arm reached the limit of motion range.	1	0	06/02/99 07:26:56	COMMON	ba:3
T-4031	A motion command cannot be executed under...	1	0	06/02/99 07:26:41	COMMON	ba:2
T-4031	A motion command cannot be executed under...	1	0	06/02/99 07:25:59	COMMON	ba:2
T-4210	he RAS circuit detected the Control Unit malfu...	2	0	05/28/99 09:33:17	SYSTEMTASK	DAEMON:0
T-4210	he RAS circuit detected the Control Unit malfu...	2	0	05/28/99 09:33:16	SYSTEMTASK	DAEMON:0
E-9103	SRCLoad information error (Input port).	0	0	05/21/99 04:24:14	WB	InitIP:0

Klicken Sie auf <Update>, um die Informationen auf den aktuellsten Stand zu bringen.

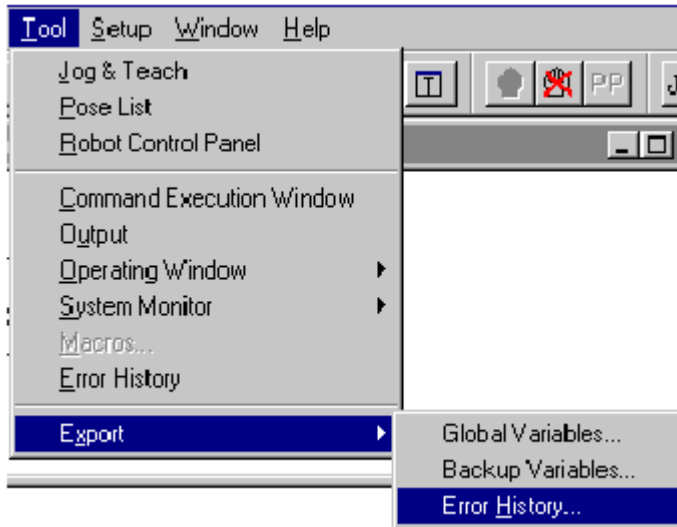
Wenn Sie auf einen Spaltennamen klicken, können Sie die Informationen nach der entsprechenden Spalte sortieren lassen.

Im unteren Beispiel wurde nach „Error Code“ (Fehlercode) sortiert.

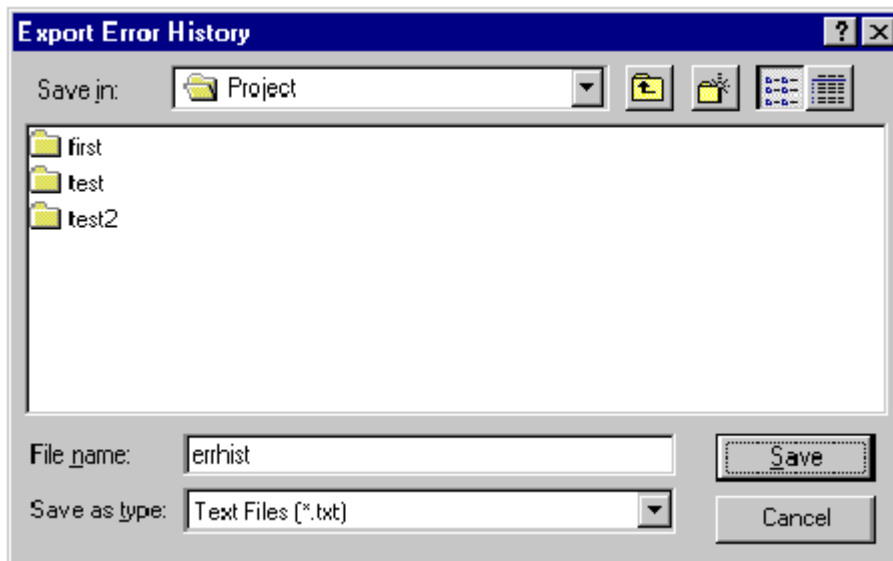
Error Code	Message	Manipu.	Joint	Date/Time	Unit	Function
E-9103	SRCLoad information error (Input port).	0	0	05/21/99 04:24:14	WB	InitIP:0
E-9103	SRCLoad information error (Input port).	0	0	05/21/99 04:17:49	WB	InitIP:0
E-9103	SRCLoad information error (Input port).	0	0	05/21/99 04:16:38	WB	InitIP:0
E-9103	SRCLoad information error (Input port).	0	0	05/21/99 04:11:08	WB	InitIP:0
E-9103	SRCLoad information error (Input port).	0	0	05/21/99 04:10:51	WB	InitIP:0
E-9103	SRCLoad information error (Input port).	0	0	05/21/99 04:09:32	WB	InitIP:0
I-2006	The arguments of the called function do not m...	0	0	06/04/99 04:26:50	MONITOR:0	
I-2323	The specified manipulator is executing a Paral...	0	1	06/04/99 02:52:06	COMMON	_main:2:8
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:56:19	COMMON	localvat:12
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:56:02	COMMON	localvat:8
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:55:44	COMMON	localvat:2
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:55:31	COMMON	localvat:2
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:53:56	COMMON	localvat:2
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:52:58	MONITOR:0	
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:52:39	MONITOR:0	
I-2524	The size of total Local Variables in this task ex...	0	0	06/04/99 04:51:11	MONITOR:0	
M-3020	HOME was executed before setting HOMESSET.	0	0	05/13/99 06:48:08	SYSTEMTASK	
M-3201	The Manipulator is being used in another task.	0	0	06/03/99 11:13:22	COMMON	main:8
T-4001	The Arm reached the limit of motion range.	1	2	06/03/99 09:24:52	COMMON	main:7
T-4007	The Arm reached the limit of motion range.	1	0	06/02/99 07:26:56	COMMON	ba:3
T-4031	A motion command cannot be executed under...	1	0	06/02/99 07:26:41	COMMON	ba:2
T-4031	A motion command cannot be executed under...	1	0	06/02/99 07:25:59	COMMON	ba:2
T-4031	A motion command cannot be executed under...	1	0	05/13/99 06:49:24	SYSTEMTASK	
T-4031	A motion command cannot be executed under...	1	0	05/13/99 06:49:20	SYSTEMTASK	

### ▪ Speichern des Fehlerprotokolls

Inhalte des Fehlerprotokolls können als Textdatei gespeichert werden. Führen Sie die Befehle [Tool] – [Export] – [Error History] (Werkzeug – Export – Fehlerprotokoll) aus.

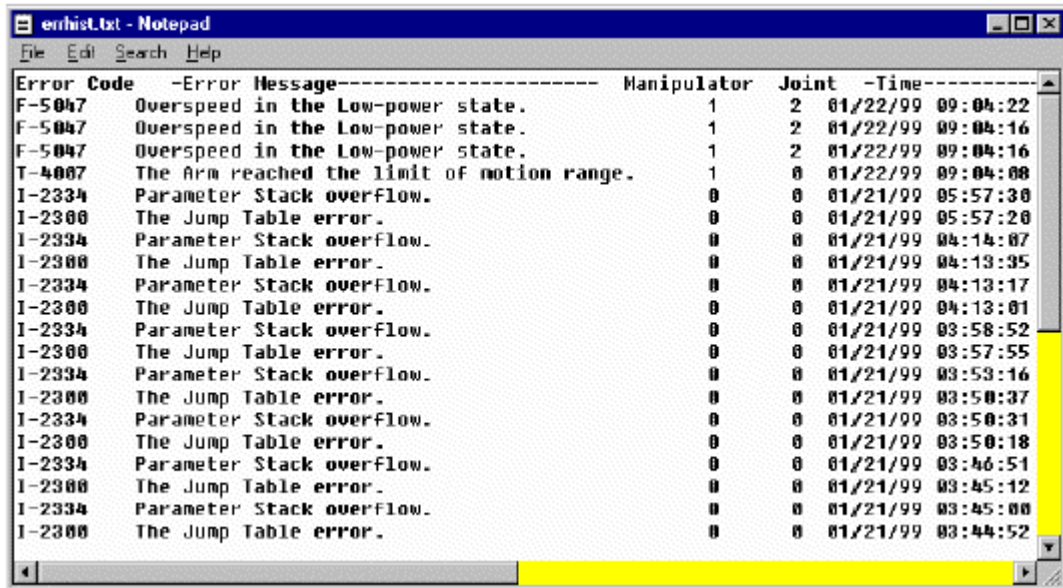


Legen Sie das Verzeichnis fest, in welches die Datei gespeichert werden soll, geben Sie einen Dateinamen ein, und klicken Sie auf <Save> (Speichern).





Diese Datei lässt sich mit einem normalen Texteditor öffnen (z.B.: Windows Notepad).

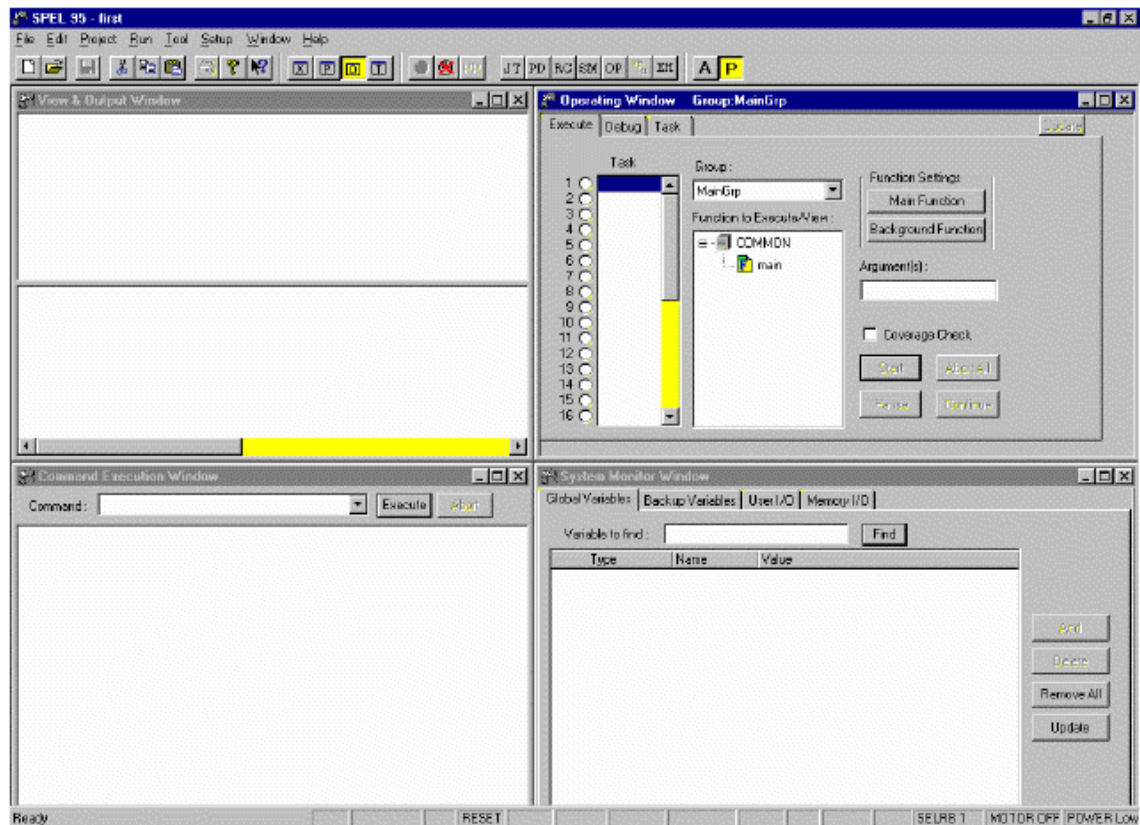


The screenshot shows a Notepad window titled 'erhist.txt - Notepad'. The window contains a log of error messages with the following columns: Error Code, Error Message, Manipulator, Joint, and Time. The errors are listed as follows:

Error Code	Error Message	Manipulator	Joint	Time
F-5047	Overspeed in the Low-power state.	1	2	01/22/99 09:04:22
F-5047	Overspeed in the Low-power state.	1	2	01/22/99 09:04:16
F-5047	Overspeed in the Low-power state.	1	2	01/22/99 09:04:16
T-4007	The Arm reached the limit of motion range.	1	0	01/22/99 09:04:00
I-2334	Parameter Stack overflow.	0	0	01/21/99 05:57:30
I-2300	The Jump Table error.	0	0	01/21/99 05:57:20
I-2334	Parameter Stack overflow.	0	0	01/21/99 04:14:07
I-2300	The Jump Table error.	0	0	01/21/99 04:13:35
I-2334	Parameter Stack overflow.	0	0	01/21/99 04:13:17
I-2300	The Jump Table error.	0	0	01/21/99 04:13:01
I-2334	Parameter Stack overflow.	0	0	01/21/99 03:58:52
I-2300	The Jump Table error.	0	0	01/21/99 03:57:55
I-2334	Parameter Stack overflow.	0	0	01/21/99 03:53:16
I-2300	The Jump Table error.	0	0	01/21/99 03:50:37
I-2334	Parameter Stack overflow.	0	0	01/21/99 03:50:31
I-2300	The Jump Table error.	0	0	01/21/99 03:50:18
I-2334	Parameter Stack overflow.	0	0	01/21/99 03:46:51
I-2300	The Jump Table error.	0	0	01/21/99 03:45:12
I-2334	Parameter Stack overflow.	0	0	01/21/99 03:45:00
I-2300	The Jump Table error.	0	0	01/21/99 03:44:52

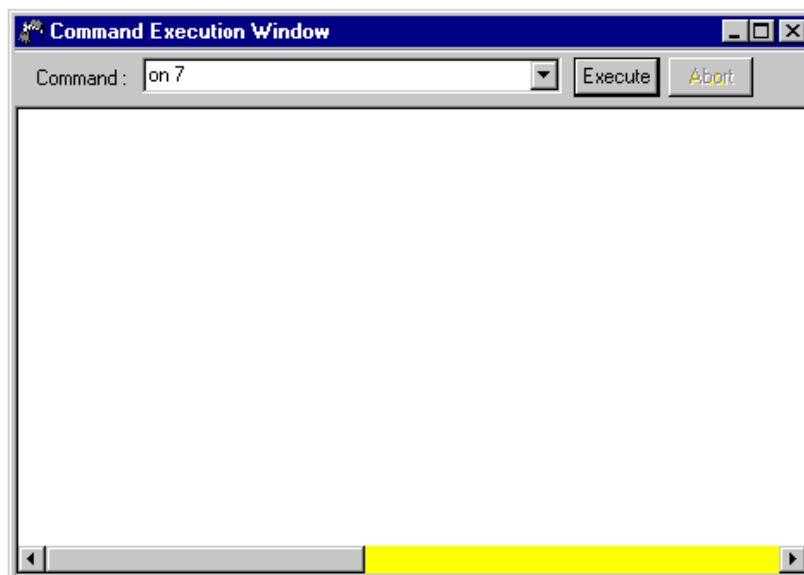
## 6.9 Bedienung direkt aus dem Befehlsausführungsfenster

Durch die Eingabe von SPEL 95-Befehlen können Sie den E/A-Ausgangszustand verändern oder den Manipulator verfahren. Weitere Informationen erhalten Sie in der SPEL 95-Onlinehilfe.



Geben Sie die Befehlszeile in dem Feld [Command] (Befehl) ein, und klicken Sie auf <Execute> (Ausführen).

Um den Betrieb während der Ausführung zu unterbrechen, klicken Sie auf <Abort> (Abbruch).



## 6.10 Bedienung aus dem Betriebsfenster

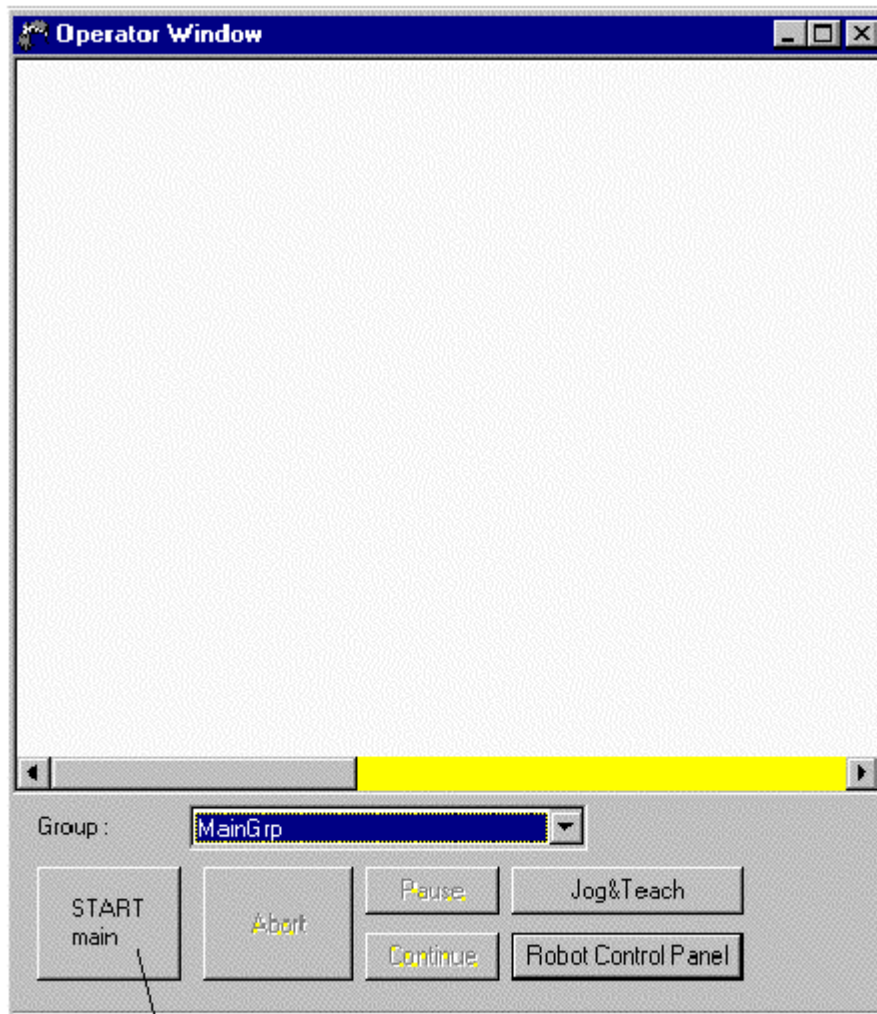
Aus dem Betriebsfenster lassen sich verschiedene Aufgaben wie z. B. Erzeugen von Tasks, (Ausführen von Programmen), Pausieren, Fortsetzen und Abbrechen aller Task, etc. steuern. Sie können sich auch zum Menü [[Jog&Teach] oder [Robot Control] bewegen.

Nachdem Sie die Entwicklung der Applikation beendet haben, wird empfohlen, Ihr System so einzustellen, dass Bediener aus dem Betriebsfenster arbeiten können. Das ermöglicht den Bedienern zu arbeiten, ohne unnötige Informationen zu sehen und demzufolge Bedienfehler zu vermeiden. Sie können die einzelnen Punkte kundenspezifisch ändern.



<Operator Window>  
(Anwender-Fenster)

Klicken Sie auf <Operator Window> (Betriebsfenster). Das Betriebsfenster erscheint am Bildschirm.



Der Name der Funktion main wird angezeigt.

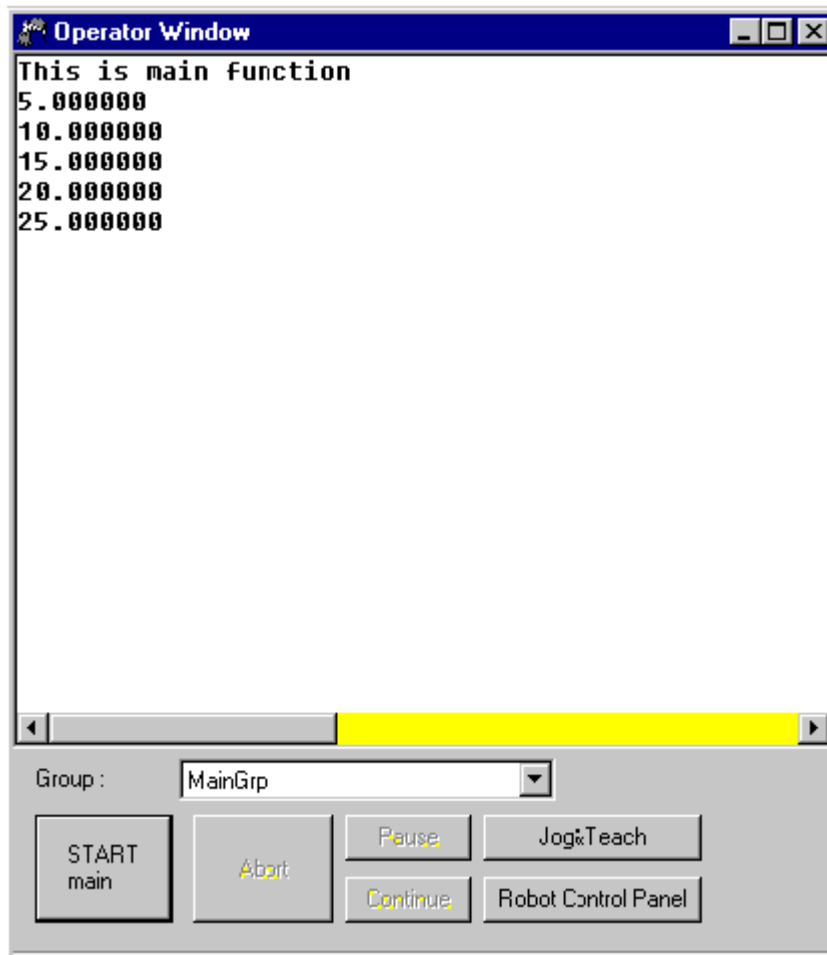
- **Wechseln der Gruppe**

Um zwischen den Gruppen zu wechseln, wählen Sie in der Auswahlliste [Group] (Gruppe) die entsprechende Gruppe.

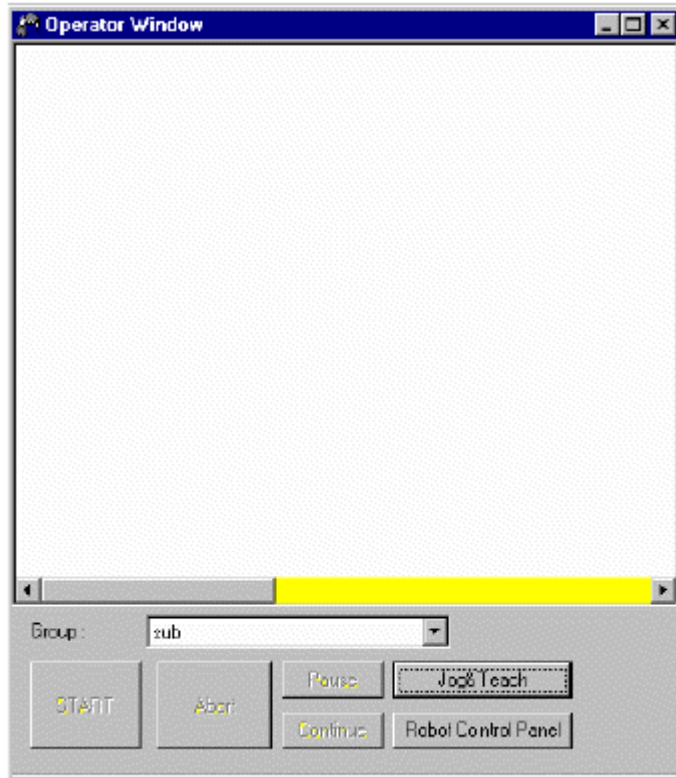
- **Erzeugen von Tasks**

Klicken Sie auf <Start>, um die Funktion main auszuführen.

Die PRINT-Anweisungen des Programms erscheinen in dem [View&Output]-Fenster (Ansicht- und Ausgabefenster).



Wurde die Funktion main keiner Gruppe hinzugefügt, ist der Schalter <Start> nicht aktiv. Das Programm kann solange nicht ausgeführt werden, bis der Schalter aktiv ist.



- **Anhalten aller Tasks**

Klicken Sie auf <Pause>, um alle Tasks vorübergehend anzuhalten. Die Tasks, die mit Hilfe des Befehls HTASKTYPE programmiert wurden, um nicht anzuhalten, halten nicht an. Auch die Hintergrundtask wird nicht gestoppt.

- **Fortsetzen aller Tasks**

Klicken Sie auf <Resume> (Fortsetzen), um alle Tasks fortzusetzen.

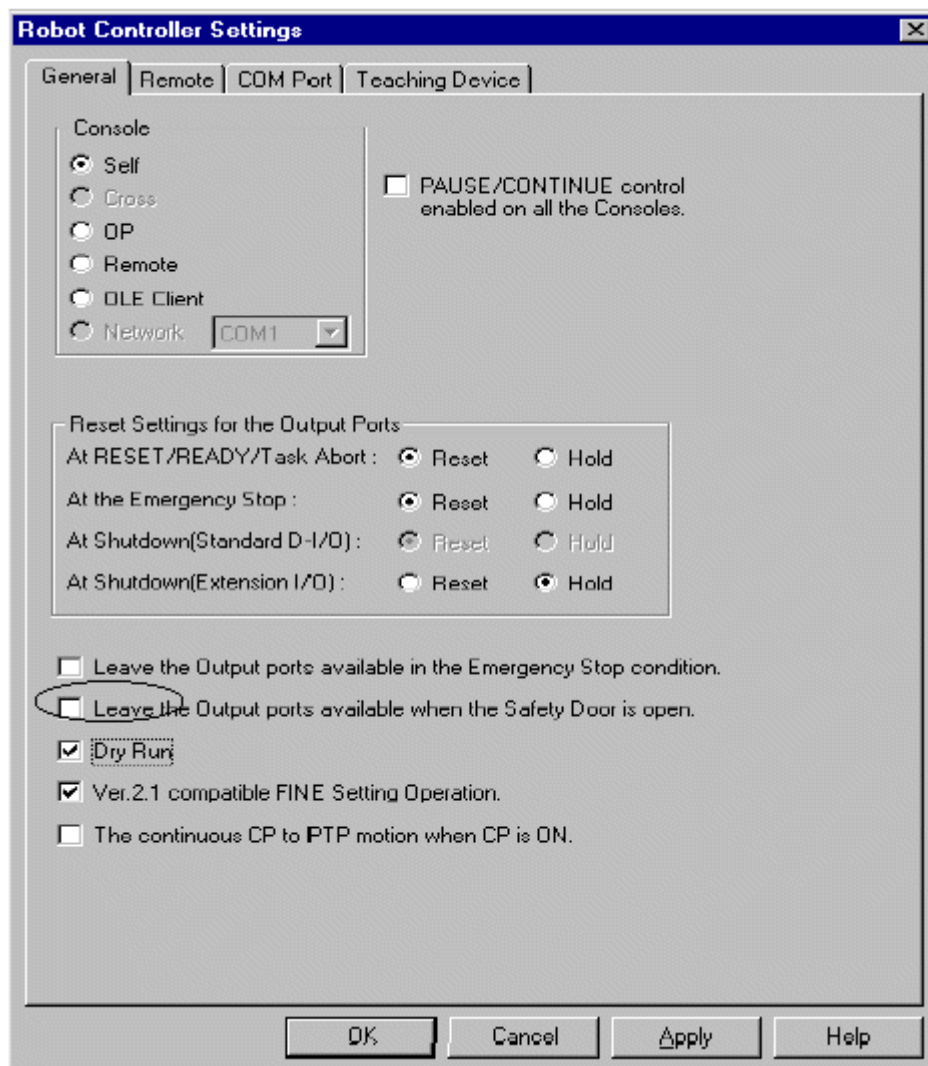
- **Abbrechen aller Tasks**

Klicken Sie auf <Abort> (Abbruch), um alle Tasks zu beenden. Die Hintergrundtask kann nicht abgebrochen werden. Nähere Informationen erhalten Sie in Kapitel 12 „Roboterzelle Steuerungsfunktionen“.

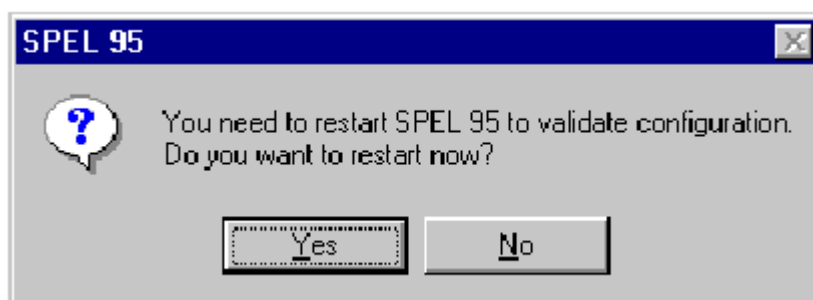
## 6.11 Testen von Programmen (Trockenlauf)

Wenn Sie in den allgemeinen Einstellungen das Optionsfeld „Dry Run“ (Trockenlauf) markiert haben, können alle Tasks ausgeführt werden, ohne dass der an der Antriebseinheit angeschlossene Manipulator betrieben wird. Das ist hilfreich, um Algorithmen oder Zykluszeiten zu testen.

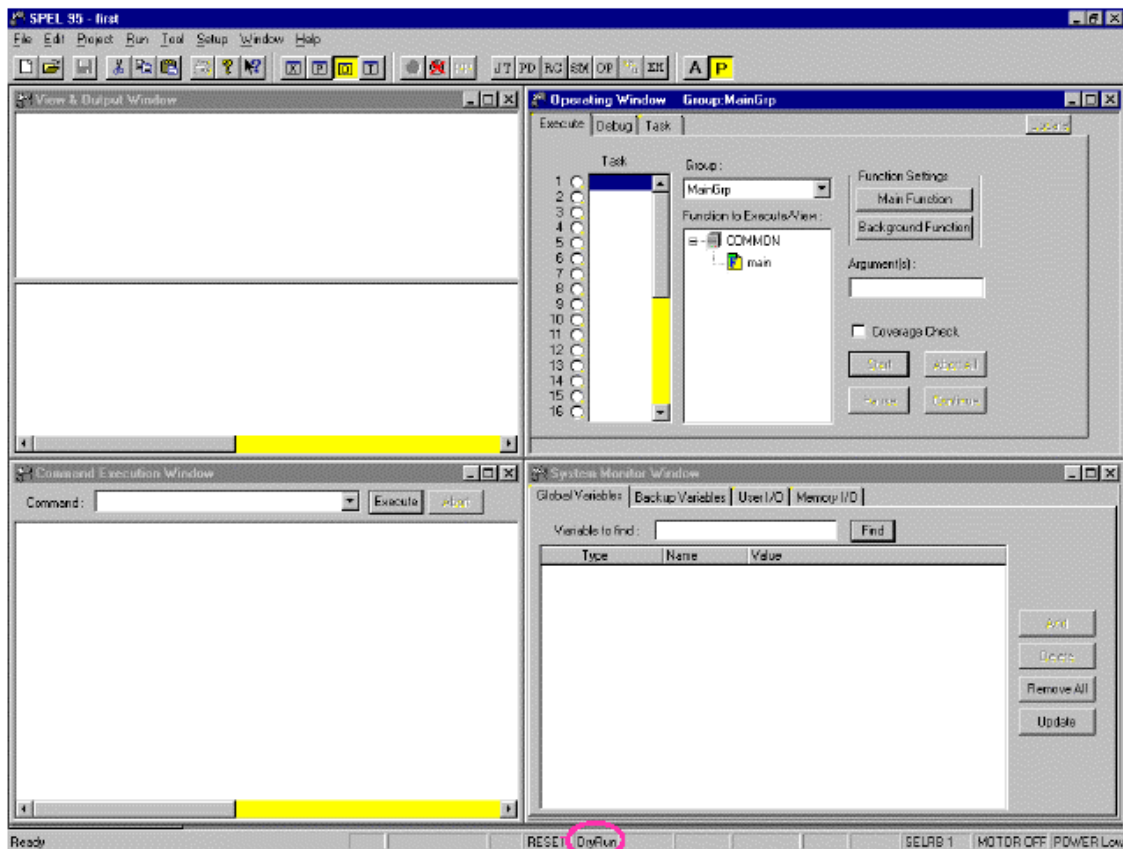
Wählen Sie [Setup] – [Robot Controller Settings] (Einrichten - Einstellungen der Robotersteuerung), und klicken Sie dann auf den Reiter [General] (Allgemein). Markieren Sie das Optionsfeld „Dry Run“, und klicken Sie dann auf <OK>.



SPEL 95 muss neu gestartet werden, damit die vorgenommenen Einstellungen wirksam werden. Klicken Sie auf <Yes>.



In der Statuszeile wird „Dry Run“ angezeigt, wenn SPEL 95 ohne MIB ausgeführt wird.



Ist die Funktion „Dry Run“ aktiviert, kann auch dann eine SPEL 95-Anwendung ausgeführt werden, wenn keine Antriebseinheit am MIB angeschlossen ist. Die Zeit, die benötigt wird, um den Manipulator zu bedienen, ist in diesem Zustand und zu diesem Zeitpunkt fast identisch mit der, die benötigt wird, wenn der Manipulator aktuell betrieben wird.



#### ACHTUNG

Die Funktionen der Erweiterungsplatinen (Anwender-E/A, Pulse-Ausgabe-Platine) sind auch im Dry-Run-Modus wirksam!

Jedoch werden die NOT-AUS-Funktionen der optionalen Geräte, wie z.B. des Jogpads, unwirksam.

Genauso wenig halten Manipulatoren an, die mit Pulse-Ausgabe-Platinen bestückt sind, und Ausgange der Anwender-E/A-Erweiterung werden nicht initialisiert.

#### HINWEIS

Im Dry-Run-Modus sind folgende Funktionen auch dann unwirksam, wenn eine Antriebseinheit angeschlossen ist:

- Spannungsversorgung zum Manipulator.
- LED-Anzeigen der Antriebseinheit.
- Eingang/Ausgang der Sicherheitsabschränkung.
- Eingang/Ausgang des NOT-AUS.
- Eingang/Ausgang der Standard-Anwender-E/A.

## 6.12 Anzeigen im AUTO-Modus

Der AUTO-Modus wurde für den automatischen Betrieb in Fabriken entwickelt. Die verfügbaren Funktionen im AUTO-Modus sind eingeschränkt für die Bedienungen.

Werkzeugleiste während der Programmausführung im AUTO-Modus:



Klicken Sie auf <System Monitor>, um sich das System-Monitor-Fenster anzeigen zu lassen.

Klicken Sie auf <Error History>, um sich das Fehlerprotokoll anzeigen zu lassen.

Benutzen Sie das Menü [Tool] (Werkzeug), um sich folgende Fenster anzeigen zu lassen:

- [Ansicht- und Ausgabefenster]: Der Ausführungszustand des Programmes kann im Quellprogrammfenster überprüft werden.
- [Betriebsfenster]: Nutzen Sie dieses Fenster, um eine Task während der Ausführung anzuhalten, zu unterbrechen oder abzuberechnen.
- [System-Monitor-Fenster]: Die Variablen können geprüft oder upgedatet, aber nicht bearbeitet werden. Die E/A kann von diesem Fenster aus überprüft, aber nicht betrieben werden. Klicken Sie auf <Update>, um die Informationen auf den neuesten Stand zu bringen.
- [Fehlerprotokoll]: Das Fehlerprotokoll kann geprüft werden. Klicken Sie auf <Update>, um das Fehlerprotokoll auf den neuesten Stand zu bringen.



## Kapitel 7

# Kommunikation mit externen Geräten

### 7.1 Eingangs-/Ausgangssteuerung

Die Daten- oder Ereignissynchronisierung wird von der externen Ausstattung empfangen und das externe Zubehör wird gesteuert.

#### 7.1.1 Anwender-E/A

Die Anwender-E/A-Signale werden zugewiesen, um mit dem externen Zubehör zu synchronisieren. Die Ein- und Ausgangssignale werden separat zugewiesen. Jedes Signal wird von „0“ an nummeriert in ansteigender Reihenfolge für die entsprechenden Ein- und Ausgangssignale.

Die Standard-Anwender-E/A ist 16 Bit für Ein- und Ausgang.

Es kann durch Hinzufügen einer Antriebseinheit auf weitere 16 Bit erweitert werden. Bis zu 3 Antriebseinheiten können hinzugefügt werden. Es kann auch auf 32 Bit erweitert werden durch Hinzufügen einer Erweiterungs-E/A. Durch die Kombination von Antriebseinheit und Erweiterungs-E/A ist eine Erweiterung auf bis zu 512 Bit möglich.

Beziehung zwischen Bitnummer und Peripheriegerät (Beispiele zeigen nur Fällen, in denen die Erweiterungs-E/A verwendet wird. Wenn die optionale REMOTE-E/A als Erweiterung verwendet wird, trifft das Beispiel nicht zu.)

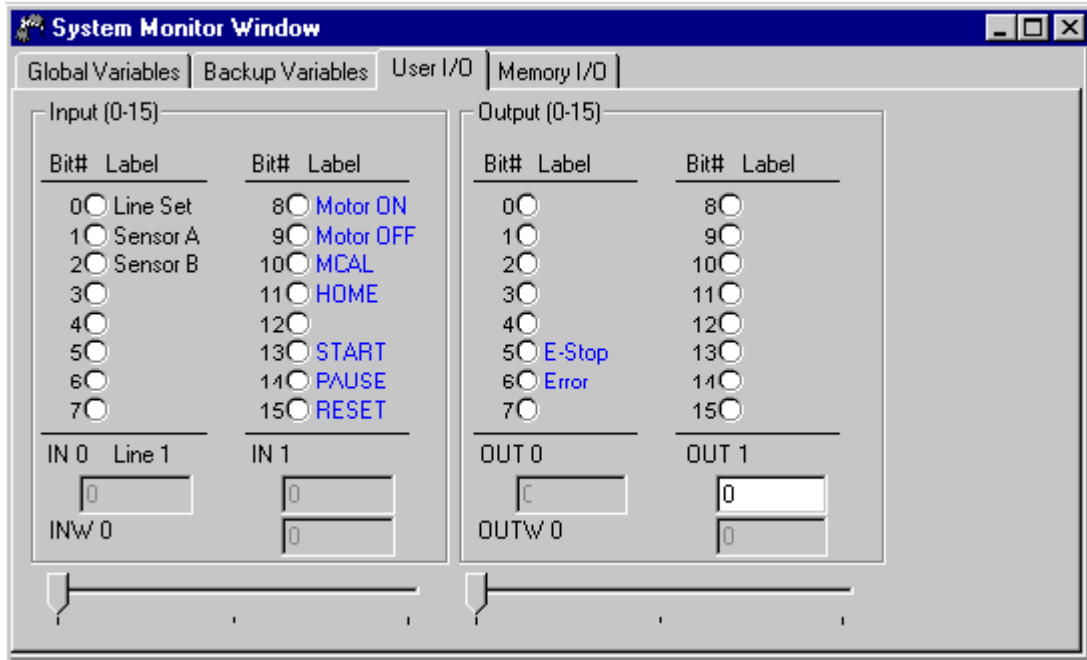
Eingangs-Bitnummer	Ausgangs-Bitnummer	anzuschließendes Gerät
0-15	0-15	Standard (Angeschlossen am D-E/A-Stecker der Antriebseinheit)
16-31	16-31	Erweiterung Antriebseinheit 1
32-47	32-47	Erweiterung Antriebseinheit 2
48-79	48-79	1. Platine der Erweiterungs-E/A
80-111	80-111	2. Platine der Erweiterungs-E/A
⋮	⋮	⋮
-495	-495	14. Platine der Erweiterungs-E/A

Weitere Informationen zu den Spezifikationen der Zusatzantriebseinheiten und der Erweiterungs-E/A erhalten Sie im Handbuch „Robotersteuerung SRC5\*\*\*“. Weitere Informationen zu den Einstellungen der Erweiterungs-E/A erhalten Sie im Handbuch „SRC5\*\* Handbuch zur E/A-REMOTE-Platine“.

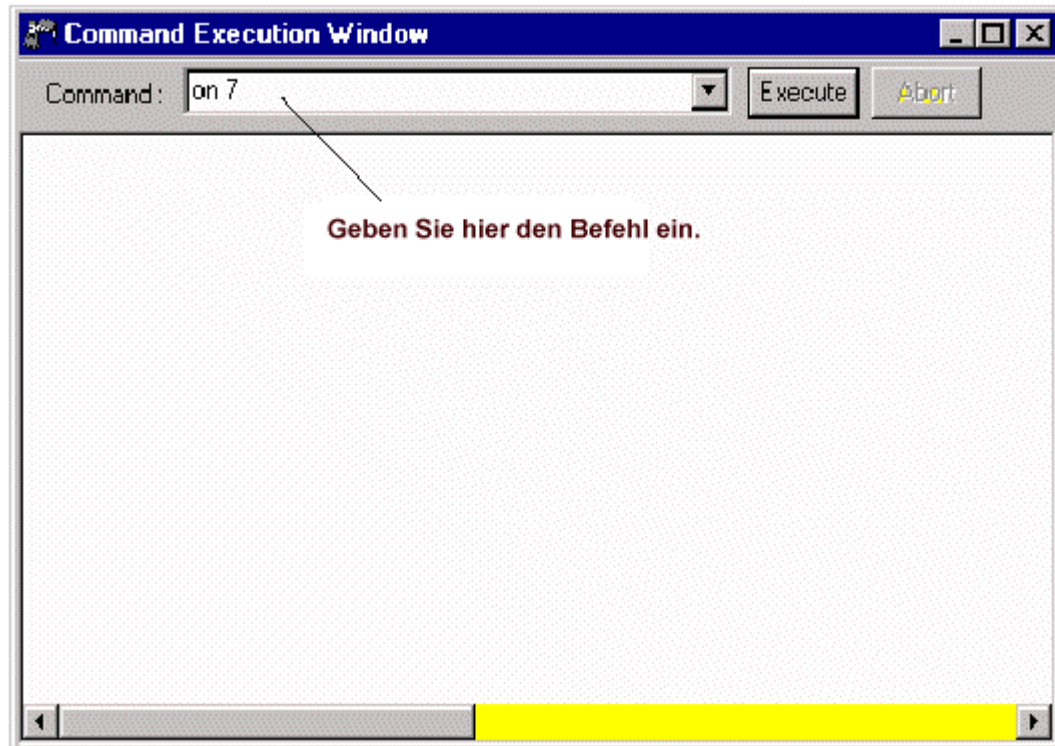
### 7.1.2 Synchronisierung mit der Anwender-E/A

Verwenden Sie das Menü [User I/O] (Anwender-E/A) im Fenster [System Monitor] um den Anwender-E/A-Status zu erhalten. Durch Eingabe eines Befehls in das Textfeld [Command] (Befehl) im Befehlsausführungsfenster können die Daten ausgegeben oder der aktuelle Eingangsstatus nachgeprüft werden.

Menü [User I/O] im Fenster [System Monitor]

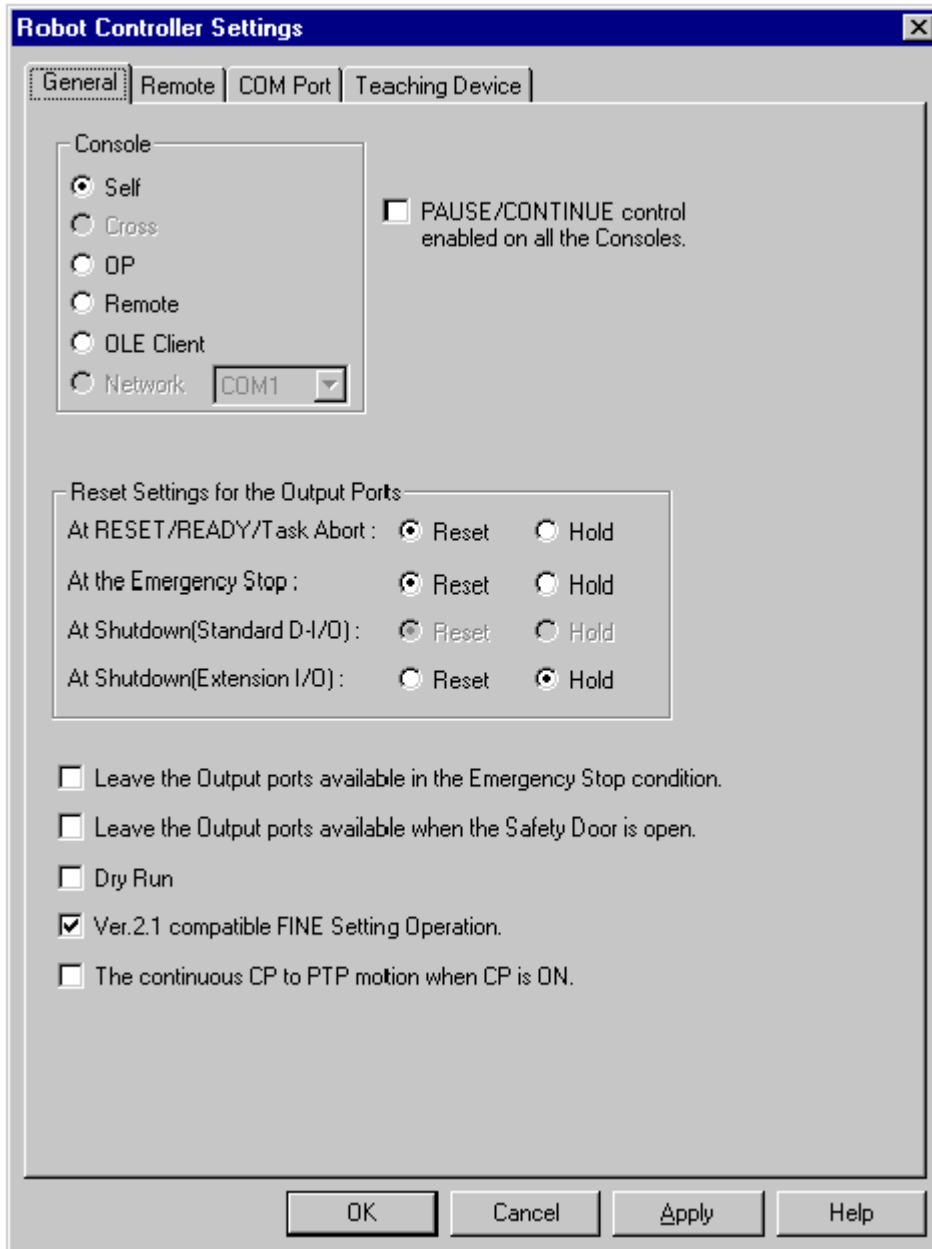


Befehlsausführungsfenster



**HINWEIS**

Wenn das an die Ausgabeausgänge angeschlossene Zubehör aktiviert es entweder eine Bewegung für sich selbst oder löst eine Bewegung an einem anderen Gerät aus. In jedem Fall ist es gefährlich die Ausgabeausgänge zu bedienen, ohne sich vorher Gedanken über mögliche Ergebnisse während eines NOT-AUS oder, wenn die Sicherheitsabschränkung geöffnet ist, zu machen. Öffnen Sie das Menü [General] (Allgemein) in den Einstellungen für die Robotersteuerung und entfernen Sie die Häkchen vor den Optionsfeldern [Leave the Output ports available in the Emergency Stop Conditon] (Ausgabeausgänge während eines NOT-AUS verfügbar lassen) und [Leave the Output ports available when the Safety Door is open] (Ausgabeausgänge während geöffneter Sicherheitsabschränkung verfügbar lassen)



**HINWEIS**

Im Menü [General] (Allgemein) in den Einstellungen für die Robotersteuerung wird in der Gruppenauswahl [Reset Settings for the Output Ports] (Reset-Einstellungen der Ausgabeausgänge) festgelegt, ob die Einstellungen für die Ausgabeausgänge im Falle eines NOT-AUS, beim Schließen von SPEL 95 oder, wenn ein Programm unterbrochen wird, zurückgesetzt oder aufrechterhalten werden.

<p>At RESET/READY/Task Abort  (Bei RESET/READY/Taskabbruch)</p>	<p>Wählen Sie entweder &lt;Reset&gt; (Zurücksetzen) oder &lt;Hold&gt;(Aufrechterhalten) für die Einstellungen der Ausgabeausgänge für jeden der folgenden Fälle:</p> <ul style="list-style-type: none"> <li>• (Beibehalten) für die Einstellungen der Ausgabeausgänge für jede der folgenden Fälle:</li> <li>• Wenn es ein gültiger RESET- oder READY-Eingang vom REMOTE anliegt.</li> <li>• Wenn ein RESET-Befehl ausgeführt wird.</li> <li>• Wenn im Menü [Ausführen] im Betriebsfenster &lt;Abort&gt; (Abbrechen) angeklickt wird.</li> <li>• Wenn im Betriebsfenster &lt;Abort&gt; (Abbrechen) angeklickt wird.</li> <li>• Wenn ein gültiger Abbruch vom REMOTE anliegt.</li> <li>• Wenn die Schalter Abort All oder ABORT AUF DEM Handprogrammiergerät gedrückt werden.</li> </ul>
<p>At the Emergency Stop (Bei einem NOT-AUS)</p>	<p>Wählen Sie entweder &lt;Reset&gt; (Zurücksetzen) oder &lt;Hold&gt; (Aufrechterhalten ) für die Einstellungen der Ausgabeausgänge wenn ein NOT-Aus anliegt und sich das System in einem NOT-AUS-Zustand befindet.</p>
<p>At Shutdown (Standard D-I/O) Beim Herunterfahren (Standard D-E/A)</p>	<p>Wählen Sie entweder &lt;Reset&gt; (Zurücksetzen) oder &lt;Hold&gt; (Aufrechterhalten ) für die Einstellungen der Ausgabeausgänge, die an der Antriebseinheit angeschlossen sind, wenn SPEL 95 geschlossen wird.</p>
<p>At Shutdown (Extension I/O) Beim Herunterfahren (Erweiterungs-E/A)</p>	<p>Wählen Sie entweder &lt;Reset&gt; (Zurücksetzen) oder &lt;Hold&gt; (Aufrechterhalten ) für die Einstellungen der Ausgabeausgänge, die an der Erweiterungs-E/A oder der REMOTE-E/A angeschlossen sind, wenn SPEL 95 geschlossen wird.</p>

## 7.2 Anwender-E/A-Steuerung Befehle und Beschreibungsmethode

### 7.2.1 Ausgang

Um den Zustand des Anwender-E/A-Ausgangs zu verändern gibt es zwei Methoden: 1. Die Änderung durch Biteinheiten und 2. die Methode durch Ausgabe von Byteeinheiten.

Biteinheiten:	ON
	OFF
Byteeinheiten:	OUT
Worteinheiten:	OUTW

**ON**

<b>Funktion</b>	Schaltet den angegebenen E/A-Ausgang des Robotersystems in Biteinheiten ein und nach einer bestimmten Zeit wieder aus.
<b>Format</b>	ON [Ausgangsbitnummer]{,[Zeit]{,[asynchrone Einstellung]}}{,FORCED}
<b>Ausgangsbitnummer</b>	Die Ausgangsbitnummer wird festgelegt durch einen Ausdruck, eine Zahl oder eine E/A-Sprungmarke. (Es handelt sich um eine ganze Zahl größer 0 und innerhalb der Anzahl der installierten E/A-Platinen.)
<b>Zeit</b>	Der Zeitpunkt an dem der Ausgang eingeschaltet wird wird durch einen Ausdruck oder eine Zahl festgelegt. (Es handelt sich um eine positive reelle Zahl in Sekunden. Kleinster einstellbarer Wert ist 0,01 Sekunden, größter einstellbarer Wert ist 10 Sekunden.)
<b>asynchrone Einstellung</b>	Für die asynchrone Einstellung kann 0 oder 1 (Standardwert ist 1) gewählt werden.
<b>Beschreibung</b>	<p>Wird nur die Ausgangsbitnummer angegeben, wird der entsprechende Ausgang eingeschaltet. Bei Angabe einer Zeit wird der Ausgang nach Ablauf des Intervalls wieder ausgeschaltet, nachdem er zuvor eingeschaltet worden war. In solchen Fällen wird der Ausgang nach der definierten Zeit ausgeschaltet, auch wenn der Ausgang zuvor durch einen Befehl in eingeschalteten Zustand versetzt worden ist. Asynchrone Einstellungen sind verwendbar, wenn die Zeit folgendermaßen definiert wird:</p> <p>1: Schaltet den Ausgang ein und führt nach Ablauf der Zeit den nächsten Befehl aus.  0: Schaltet den Ausgang ein und führt gleichzeitig den nächsten Befehl aus.  Wird keine Einstellung gewählt, ist dies identisch mit Einstellung 1.</p> <p>Wird der ON-Befehl mit dem Ausdruck „FORCED“ (Erzwungen) versehen, wird der Ausgang eingeschaltet ungeachtet davon, ob die Optionsfelder [Leave the Output ports available in the Emergency Stop Conditon] (Ausgabeausgänge während eines NOT-AUS verfügbar lassen) und [Leave the Output ports available when the Safety Door is open] (Ausgabeausgänge während geöffneter Sicherheitsabschränkung verfügbar lassen) im Menü [General] markiert sind. Schließen Sie keine Geräte, die sich bewegen oder Bewegungen auslösen an dem erzwungenen Ausgang an.</p> <p>Wird eine negative Zahl als Ausgangsbitnummer eingegeben oder, wenn der eingegebene Wert die Anzahl der installierten E/A-Platinen überschreitet, tritt ein Fehler auf.</p> <p>Wurde eine Ausgangsbitnummer eingegeben, die durch REMOTE festgelegt wurde, tritt ein Fehler auf. Das durch REMOTE eingestellte Ausgangsbit schaltet gemäß des Status der Robotersteuerung automatisch ein und aus. Dann wird der Befehl nicht funktionieren.</p>
<b>Beispiel</b>	<pre>ON 1 ON LED1 , 1 , FORCED</pre>

**OFF**

<b>Funktion</b>	Schaltet den angegebenen E/A-Ausgang des Robotersystems in Biteinheiten aus und nach einer bestimmten Zeit wieder ein.
<b>Format</b>	OFF [Ausgangsbitnummer]{,[Zeit]{,[asynchrone Einstellung]}}{,FORCED}
<b>Ausgangsbitnummer</b>	Die Ausgangsbitnummer wird festgelegt durch einen Ausdruck, eine Zahl oder eine E/A-Sprungmarke. (Es handelt sich um eine ganze Zahl größer 0 und innerhalb der Anzahl der installierten E/A-Platinen.)
<b>Zeit</b>	Der Zeitpunkt an dem der Ausgang ausgeschaltet wird wird durch einen Ausdruck oder eine Zahl festgelegt. (Es handelt sich um eine positive reelle Zahl in Sekunden. Kleinster einstellbarer Wert ist 0,01 Sekunden, größter einstellbarer Wert ist 10 Sekunden.)
<b>asynchrone Einstellung</b>	Für die asynchrone Einstellung kann 0 oder 1 (Standardwert ist 1) gewählt werden.
<b>Beschreibung</b>	<p>Wird nur die Ausgangsbitnummer angegeben, wird der entsprechende Ausgang ausgeschaltet. Bei Angabe einer Zeit wird der Ausgang nach Ablauf des Intervalls ausgeschaltet und danach wieder eingeschaltet. In solchen Fällen wird der Ausgang nach der definierten Zeit ausgeschaltet, auch wenn der Ausgang zuvor durch einen Befehl in eingeschalteten Zustand versetzt worden ist.</p> <p>Asynchrone Einstellungen sind verwendbar, wenn die Zeit folgendermaßen definiert wird:</p> <p>1: Schaltet den Ausgang ein und führt nach Ablauf der Zeit den nächsten Befehl aus.  0: Schaltet den Ausgang ein und führt gleichzeitig den nächsten Befehl aus.  Wird keine Einstellung gewählt, ist dies identisch mit Einstellung 1.</p> <p>Wird der OFF-Befehl mit dem Ausdruck „FORCED“ (Erzwingen) versehen, wird der Ausgang ausgeschaltet ungeachtet davon, ob die Optionsfelder [Leave the Output ports available in the Emergency Stop Conditon] (Ausgabeausgänge während eines NOT-AUS verfügbar lassen) und [Leave the Output ports available when the Safety Door is open] (Ausgabeausgänge während geöffneter Sicherheitsabschränkung verfügbar lassen) im Menü [General] markiert sind. Schließen Sie keine Geräte, die sich bewegen oder Bewegungen auslösen an dem erzwungenen Ausgang an.</p> <p>Wird eine negative Zahl als Ausgangsbitnummer eingegeben oder, wenn der eingegebene Wert die Anzahl der installierten E/A-Platinen überschreitet, tritt ein Fehler auf.</p> <p>Wurde eine Ausgangsbitnummer eingegeben, die durch REMOTE festgelegt wurde, tritt ein Fehler auf. Das durch REMOTE eingestellte Ausgangsbit schaltet gemäß des Status der Robotersteuerung automatisch ein und aus. Dann wird der Befehl nicht funktionieren.</p>
<b>Beispiel</b>	<pre>OFF 1,5 OFF LED1, FORCED</pre>

**OUT**

<b>Funktion</b>	Schaltet die Anwender-E/A-Ausgangsports des Robotersystems in Byteeinheiten ein und aus.
<b>Format</b>	OUT [Ausgangsbytenummer],[Byte-Ausgangsdaten]{,FORCED} Ausgangsbytenummer    Ausdruck, Zahl oder E/A-Sprungmarke zum Identifizieren des Ausgangsbytes.. Byte-Ausgangsdaten    Zweistellige hexadezimale Zahl (8Bit), Dezimalzahl oder numerische Formel.
<b>Beschreibung</b>	Das festgelegte Bit oder die numerische Formel der Anwender-E/A werden in Byteeinheiten ein- und ausgeschaltet. Für ON- und OFF-Befehle werden Bitausgangsnummern mittels absoluten Zahlen ausgedrückt. Für OUT-Befehle werden sie mittels Ausgangsbytenummer und Byte-Ausgangsdaten ausgedrückt. Ausgangsbytenummern werden nach jedem achten Bit um 1 erhöht. Also, wenn Sie die Bits 8 bis 15 steuern wollen, ist die Ausgangsbytenummer 1.

Ausgangsbitnummer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Ausgangsbytenummer	0								1								2
Byte-Ausgangsdaten	Hexadezimale Zahlen von 0-FF Dezimale Zahlen von 0-255								Hexadezimale Zahlen von 0-FF Dezimale Zahlen von 0-255								

Die Byte-Ausgangsdaten (für eine OUT-Befehl), innerhalb einer Byte-Ausgangsnummer wird als Hexadezimal- oder Dezimalzahl ausgedrückt.

Bitnummer	0	1	2	3	4	5	6	7	Hexadezimal	Dezimal
Bitstatus	0	1	1	0	0	0	0	0	&H06	6
	1	0	1	0	1	1	0	1	&HB5	181
	1	1	1	1	1	1	1	1	&HFF	255

Bitstatus: 0:OFF; 1:ON

Wird der OUT-Befehl mit dem Ausdruck „FORCED“ (Erzwungen) versehen, wird der Ausgang ein-/ausgeschaltet ungeachtet davon, ob die Optionsfelder [Leave the Output ports available in the Emergency Stop Conditon] (Ausgabeausgänge während eines NOT-AUS verfügbar lassen) und [Leave the Output ports available when the Safety Door is open] (Ausgabeausgänge während geöffneter Sicherheitsabschränkung verfügbar lassen) im Menü [General] markiert sind. Schließen Sie keine Geräte, die sich bewegen oder Bewegung auslösen, an dem erzwungenen Ausgang an.

**Beispiel**

```
OUT 2, &HE4           `Ausgabebits 16 bis 32 als „00100111“.
```

```
OUT 2, 228           `Ausgabebits 16 bis 32 als „00100111“.
```

```
OUT Port1, &HF0
```



**OUTW**

<b>Funktion</b>	Schaltet die Anwender-E/A-Ausgangsports des Robotersystems in Worteinheiten ein und aus.
<b>Format</b>	OUTW [Ausgangswortnummer],[Ausgangsdaten]{,FORCED} Ausgangswortnummer Wird festgelegt durch einen Ausdruck, eine Zahl oder eine E/A-Sprungmarke. Ausgangsdaten Ausdruck oder Zahl (Integer zwischen 0 und 65535)
<b>Beschreibung</b>	Die Anwender-E/A-Ausgabeausgänge sind in Gruppen zu je 16 Bits organisiert. Der Status dieser Gruppen, die durch Ausgangswortnummer festgelegt werden, wechselt entsprechend der festgelegten Ausgangsdaten. Die Gruppen beinhalten standardmäßig 16 Ausgangsbits. Ausgangsbits können bis 512 Bits einschließlich Optionen beinhalten. D. h. es können maximal 32 Gruppen vorhanden sein. (Sie unterscheiden sich in Abhängigkeit der installierten E/A-Platinen.)  Wird eine negative Zahl als Ausgangswortnummer eingegeben oder, wenn der eingegebene Wert die Anzahl der installierten E/A-Platinen überschreitet, tritt ein Fehler auf. In der nachfolgenden Tabelle wird die Beziehung zwischen Ausgangswortnummer, LSB (Least significant Bit) (niedrigste Bitstelle) / MSB(most significant Bit) (höchste Bitstelle) und Ausgangs-E/A-Bitnummer dargestellt.

Wortausgangsnummer	MSB								LSB
0	15	14	13	12	...	3	2	1	0
1	31	30	29	28	...	19	18	17	16
2	47	46	45	44	...	35	34	33	32
...	...	...	...	...	...	...	...	...	...
30	495	494	493	492	...	483	482	481	480
31	511	510	509	508	...	499	498	497	496

Wird der OUTW-Befehl mit dem Ausdruck „FORCED“ (Erzwungen) versehen, wird der Ausgang ein-/ausgeschaltet ungeachtet davon, ob die Optionsfelder [Leave the Output ports available in the Emergency Stop Conditon] (Ausgabeausgänge während eines NOT-AUS verfügbar lassen) und [Leave the Output ports available when the Safety Door is open] (Ausgabeausgänge während geöffneter Sicherheitsabschrankeung verfügbar lassen) im Menü [General] markiert sind. Schließen Sie keine Geräte, die sich bewegen oder Bewegungen auslösen, an dem erzwungenen Ausgang an.

## 7.2.2 Eingang

Um den Zustand des Anwender-E/A-Eingangs zu verändern gibt es zwei Methoden: 1. Die Überprüfung durch Biteinheiten und 2. die Überprüfung der Byteeinheiten.

Biteinheiten: SW( )  
 Byteeinheiten: IN( )  
 Worteinheiten: INW( )

### SW()

<b>Funktion</b>	Gibt den Status eines Anwender-E/A-Bits aus.	
<b>Format</b>	SW ([Eingangsbitnummer])	
<b>(1)</b>	Eingangsbitnummer	Wird festgelegt durch einen Ausdruck, eine Zahl oder eine E/A-Sprungmarke. (Es handelt sich um eine ganze Zahl, größer 0 und innerhalb der Anzahl der installierten E/A-Platinen.
<b>Beschreibung</b>	Gibt den Status des in der Anwender-E/A festgelegten Bits entweder mit 0 oder mit 1 aus. 0: AUS 1: EIN	
<b>Beispiel</b>		

```
>PRINT SW(1)
a=SW(Sensor1)

IF SW(1)=1 THEN
  ON 10
ELSE
  OFF 10
ENDIF

WAIT SW(2)=1
```

**IN()**

<b>Funktion</b>	Gibt den Status eines Anwender-E/A-Bits in Byte aus.
<b>Format</b>	IN [Eingangsbytenummer]
<b>(1)</b>	Eingangsbytenummer Wird festgelegt durch einen Ausdruck, eine Zahl oder eine E/A-Sprungmarke. (Es handelt sich um eine ganze Zahl, größer 0 und innerhalb der Anzahl der installierten E/A-Platinen.
<b>Beschreibung</b>	Gibt den Status eines Eingangsports mit der entsprechenden Eingangsbytenummer als Dezimalwert aus. Eingangsbytenummern werden nach jedem achten Bit um 1 erhöht. Also, wenn Sie die Bits 8 bis 15 steuern wollen, ist die Eingangsbytenummer 1.

Eingangsbitnummer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Eingangsbytenummer	0								1								2
Byte-Eingangsdaten	Dezimale Zahlen von 0-255								Dezimale Zahlen von 0-255								

**Beispiel**

```
PRINT IN(1)
```

Druckt den Status von Eingangsbytenummer 1  
(Eingangsbitnummern 8-15)

```
SELECT CASE IN(2)
```

```
CASE 0
```

```
CALL Sub1
```

```
CASE 128
```

```
CALL Sub2
```

```
CASE 255
```

```
CALL Sub3
```

```
END SELECT
```

```
dat=IN(Port1)
```

## INW()

<b>Funktion</b>	Gibt den Status eines Anwender-E/A-Bits in Worteinheiten aus.				
<b>Format</b>	INW [Eingangswortnummer]				
	<table border="0"> <tr> <td style="padding-right: 10px;">Eingangsbitnummer</td> <td>Wird festgelegt durch einen Ausdruck, eine Zahl oder eine E/A-Sprungmarke. (Es handelt sich um eine ganze Zahl, größer 0 und innerhalb der Anzahl der installierten E/A-Platinen.</td> </tr> <tr> <td>Rückgabewerte</td> <td>Der aktuelle Status (ganze Zahl zwischen 0 und 65535) der Speicher-E/A in Worteinheiten.</td> </tr> </table>	Eingangsbitnummer	Wird festgelegt durch einen Ausdruck, eine Zahl oder eine E/A-Sprungmarke. (Es handelt sich um eine ganze Zahl, größer 0 und innerhalb der Anzahl der installierten E/A-Platinen.	Rückgabewerte	Der aktuelle Status (ganze Zahl zwischen 0 und 65535) der Speicher-E/A in Worteinheiten.
Eingangsbitnummer	Wird festgelegt durch einen Ausdruck, eine Zahl oder eine E/A-Sprungmarke. (Es handelt sich um eine ganze Zahl, größer 0 und innerhalb der Anzahl der installierten E/A-Platinen.				
Rückgabewerte	Der aktuelle Status (ganze Zahl zwischen 0 und 65535) der Speicher-E/A in Worteinheiten.				
<b>Beschreibung</b>	<p>Die Anwender-E/A-Eingabeeingänge sind in Gruppen zu je 16 Bits organisiert. Der Status dieser Gruppen, die durch Eingangswortnummer festgelegt werden, wechselt entsprechend der festgelegten Eingangsdaten.</p> <p>Die Gruppen beinhalten standardmäßig 16 Eingangsbits. Eingangsbits können bis 512 Bits einschließlich Optionen beinhalten. D. h. es können maximal 32 Gruppen vorhanden sein. (Sie unterscheiden sich in Abhängigkeit der installierten E/A-Platinen.)</p> <p>Wird eine negative Zahl als Eingangswortnummer eingegeben oder, wenn der eingegebene Wert die Anzahl der installierten E/A-Platinen überschreitet, tritt ein Fehler auf.</p> <p>In der nachfolgenden Tabelle wird die Beziehung zwischen Eingangswortnummer, LSB (Least significant Bit) (niedrigste Bitstelle) / MSB (most significant Bit) (höchste Bitstelle) und Eingangs-E/A-Bitnummer dargestellt.</p>				

Worteingangsnummer	MSB								LSB
0	15	14	13	12	...	3	2	1	0
1	31	30	29	28	...	19	18	17	16
2	47	46	45	44	...	35	34	33	32
...	...	...	...	...	...	...	...	...	...
30	495	494	493	492	...	483	482	481	480
31	511	510	509	508	...	499	498	497	496

## 7.3 REMOTE-Einstellungen

Um den Roboter mit Hilfe eines Sequencers oder einer anderen Anwenderbedieneinheit zu steuern können die folgenden REMOTE-Funktionen an die Standard- oder Erweiterungs-E/A angelegt werden. Der Anwender kann jede E/A-Nummer einstellen, um die REMOTE-Funktionen anlegen zu können. Das Handbuch „SRC5\*\* Robotersteuerung“ beinhaltet Informationen über die Verdrahtung der Steuerung. Nähere Informationen zur Standard-E/A entnehmen Sie bitte Kapitel 5 und die Informationen zur Erweiterungs-E/A entnehmen Sie Kapitel 6 des Handbuchs. Zur Steuerung SRC5\*\*.

### 7.3.1 Ausgabe

Die REMOTE-Ausgabe ist eine Funktion um den aktuellen Status des Manipulators, der Steuerung und Bedienmethode, etc. extern an die Steuerung auszugeben.

Folgende Punkte können eingestellt werden:

Funktion	Beschreibung
Motor Power	Gibt MotorPower (MOTOR ON:ON) an den Monitor aus. Gibt den Status des Manipulators, wie durch Eingabe des Sel Mnp-Signals festgelegt, aus.
Motor Status	Gibt den Motorstatus (HP-Status: ON) aus. Gibt den Status des Manipulators, wie durch Eingabe des Sel Mnp-Signals festgelegt, aus.
MCAL-LED	Gibt den MCAL-Status aus (Initialisiert den Manipulator zur mechanischen Ursprungsposition)(Executed:ON). Gibt den Status des Manipulators, wie durch Eingabe des Sel Mnp-Signals festgelegt, aus.
HOME	Gibt den Status der Ursprungsposition wider. Gibt den Status des Manipulators, wie durch Eingabe des Sel Mnp-Signals festgelegt, aus.
Mnp.No.0	Gibt die Manipulatornummer aus (*1) Gibt den Status des Manipulators, wie durch Eingabe des Sel Mnp-Signals festgelegt, aus.
Mnp.No.1	
Mnp.No.2	
Mnp.No.3	
START-Monitor	Gibt den START-Status der Steuerung an den Monitor aus.
In PAUSE	Gibt den PAUSE-Status der Steuerung an den Monitor aus.
RESET	Gibt den RESET-Status der Steuerung an den Monitor aus.
Sicherheitstüre	Gibt den Status der Sicherheitstüre (sicherheitsabschränkung) an den Monitor aus. (Wenn die Sicherheitsabschränkung geöffnet ist: ON)
Enable	Gibt den Status des Zustimm-Tasters des Jog-Pads an den Monitor aus.
E-Stop	Gibt den NOT-AUS-Status der Steuerung an den Monitor aus.
Error	Gibt den Fehler-Status der Steuerung an den Monitor aus.
Grp.No.0	Gibt die Gruppennummer im Projekt aus (*2) Gibt die aktuelle Gruppennummer, wie durch Eingabe diese Bits festgelegt, aus.
Grp.No.1	
Grp.No.2	
Grp.No.3	
AUTO-Modus	Gibt den Betriebsmodus an den Monitor aus. (Im AUTO-Modus : ON; Im Programmiermodus: OFF)
ATTEND-Modus	Gibt den Betriebsmodus an den Monitor aus. (Im ATTEND-Modus : ON; Im Normalmodus: OFF)
READY	Gibt den READY-Status der Steuerung an den Monitor aus. (Im RESET-Zustand: ON)

So wie die Standard-E/A hat 16/16 Ausgänge, ist es wichtig für die Erweiterungs-E/A alle Funktionen auszugeben:

(\*1) Festlegen der Manipulatornummer mit Mnp.0-3 4 Bits

Mnp.No.0	Mnp.No.1	Mnp.No.2	Mnp.No.3	Festgelegter Manipulator
0	0	0	0	1
1	0	0	0	2
0	1	0	0	3
1	1	0	0	4
0	0	1	0	5
1	0	1	0	6
0	1	1	0	7
1	1	1	0	8
0	0	0	1	9
1	0	0	1	10
0	1	0	1	11
1	1	0	1	12
0	0	1	1	13
1	0	1	1	14
0	1	1	1	15
1	1	1	1	16

0=OFF; 1=ON

(\*1) Festlegen der Gruppennummer mit Mnp.0-3 4 Bits

Mnp.No.0	Mnp.No.1	Mnp.No.2	Mnp.No.3	Festgelegter Manipulator
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
1	1	0	0	3
0	0	1	0	4
1	0	1	0	5
0	1	1	0	6
1	1	1	0	7
0	0	0	1	8
1	0	0	1	9
0	1	0	1	10
1	1	0	1	11
0	0	1	1	12
1	0	1	1	13
0	1	1	1	14
1	1	1	1	15

0=OFF; 1=ON

### ▪ Zustand der Steuerung

Der REMOTE-Ausgang verfügt über eine Funktion, die den aktuellen Zustand der Steuerung ausgibt.fdgdfg

Der Zustand der Steuerung läßt sich in 11 Kategorien unterteilen:

- |   |  |
|---|--|
| 1. RESET (Zurücksetzen)                     | Der Zustand in dem keine Tasks ausgeführt werden. (Auch keine Hintergrundtask!)  |
| 2. READY (Fertig)                           | In diesem Zustand können nur Hintergrundtasks ausgeführt werden.   |
| 3. START                                    | In diesem Zustand können gewöhnliche Tasks ausgeführt werden. Dies beinhaltet die gleichzeitige Ausführung von Hintergrundtasks. |
| 4. PAUSE                                    | Der Zustand in dem eine Task pausiert.   |
| 5. Safety door open                         | In diesem Zustand ist eine Sicherheitsabschränkung geöffnet.   |
| 6. Emergency Stop                           | In diesem Zustand liegt ein NOT-AUS an.  |
| 7. Error                                    | In diesem Zustand ist zumindest in einer Task ein Fehler aufgetreten.  |
| 8. Safety door open, emergency Stop         |  |
| 9. Safety door open, error                  |  |
| 10. Emergency Stop, error                   |  |
| 11. Safety door open, Emergency Stop, error |  |

<Erklärung>

- |  |  |
|--|--|
| 1. RESET (Zurücksetzen)  | Der Zustand in dem keine Tasks ausgeführt werden. (Auch keine Hintergrundtask!)  |
| 2. READY (Fertig)  | In diesem Zustand können nur Hintergrundtasks ausgeführt werden.   |
| 3. START   | In diesem Zustand können gewöhnliche Tasks ausgeführt werden. Dies beinhaltet die gleichzeitige Ausführung von Hintergrundtasks. |
| • Die Ausführung von Befehlen aus dem Befehlsausführungsfenster, E/A-Operationen mit Hilfe des GUIs und so weiter, werden als gewöhnliche Tasks angesehen, wenn diese ausgeführt werden, während sich die Steuerung im Zustand START befindet. |  |
| • Der Zustand „Task im Prozeß“ beinhaltet die Zustände in der eine Task aufgrund des Befehls HALT pausiert und den Ereignis-Warte-Zustand.   |  |
| • Nähere Informationen über den Zustand in der eine Task aufgrund des Befehls HALT pausiert erhalten Sie in Kapitel 6.1.9 „Fortsetzen der angehaltenen Tasks“. Nähere Informationen zu den Hintergrundtasks erhalten Sie in Kapitel 12.1.      |  |
| 4. PAUSE   | Der Zustand in dem eine Task pausiert.   |

Die Ursachen für den Wechsel in den PAUSE-Zustand:

- ❖ Im START-Zustand tritt ein PAUSE-Ereignis ein (PAUSE-Befehl, <Pause>-Schalter, PAUSE-Eingang)
- ❖ Im NOT-AUS-Zustand liegt ein RECOVER-Eingang an. (Wiederherstellen)
- ❖ Im Zustand „Sicherheitsabschränkung geöffnet“ liegt ein Schloßfreigabe-Eingang an.
- ❖ Im Fehler-Zustand folgt dem Verlassen einer Task ein Fehler.

Wenn sich die Steuerung im PAUSE-Zustand befindet, ist es abhängig vom zugewiesenen Wert in den Einstellungen, die jede Task steuern, ob die Task tatsächlich angehalten wird oder nicht. Daher ist es durch Verwendung des Befehls HTASKTYPE, und Einrichten der Einstellungen so, daß die Task nicht angehalten wird, wenn eine PAUSE-Situation eintritt, möglich eine Situation zu erreichen, in der die Task tatsächlich ausgeführt wird, obwohl sich die Steuerung im PAUSE-Zustand befindet.

Jedoch wenn der Befehl zum Ausführen der Task ausgeführt wird gerät der Manipulator in einen QUICK-PAUSE-Zustand.

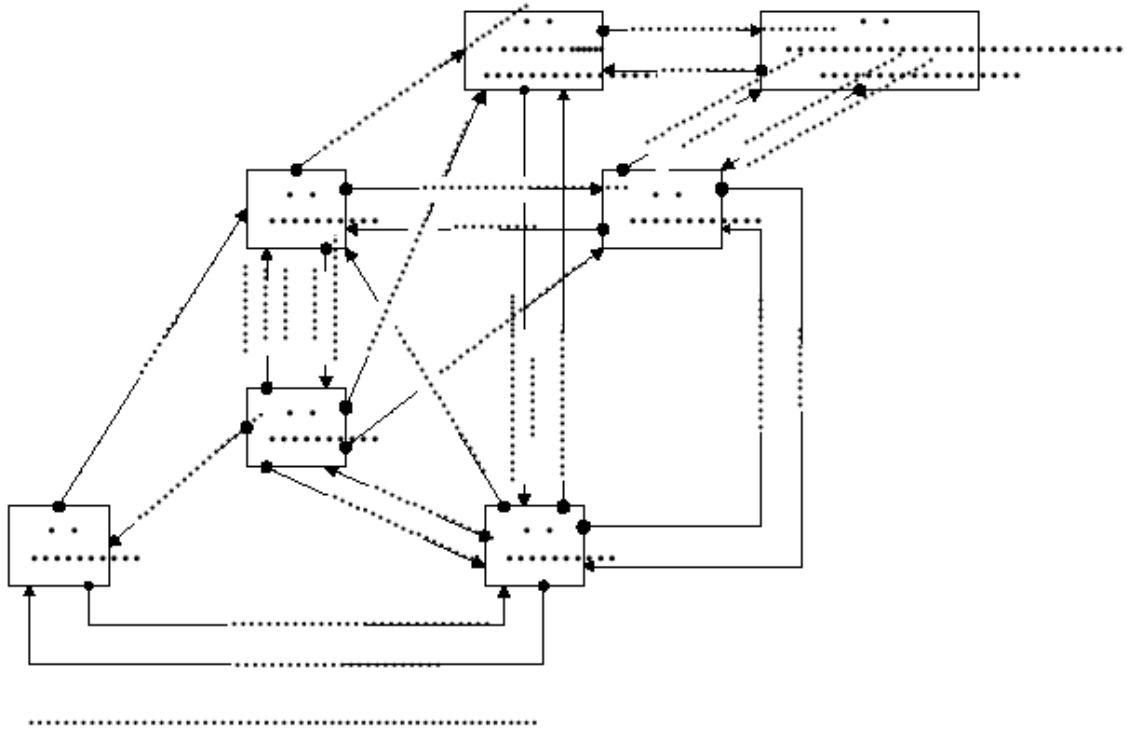
Die folgenden vier sind Kombinationen von 5., 6. und 7.

8. Safety door open, emergency Stop
9. Safety door open, error
10. Emergency Stop, error
11. Safety door open, Emergency Stop, error

#### ▪ Wechsel der Zustände der Steuerung

Nachfolgend wird der Wechsel zwischen den Zuständen der Steuerung aufgezeigt:

Der Unterschied zwischen dem READY- und dem RESET-Zustand liegt darin, ob Hintergrundtask ausgeführt werden oder nicht, wobei RESET dem READY untergeordnet ist. Weiterhin wird die Kombination der Fehlerzustände 9. bis 11. unterdrückt.





▪ **Beziehung zwischen dem Zustand der Steuerung und dem REMOTE-Ausgang**

RESET    READY    START    PAUSE    Sicherheitsabschränkung offen    NOT-AUS    Fehler    Fehler  
Fehler

Zustand der Steuerung

Zustand der Steuerung	REMOTE-Ausgang						
	RESET	READY	START	PAUSE	Sicherheitsab. offen	NOT-AUS	Fehler
1. RESET (Zurücksetzen)	ON	ON	-	-	-	-	-
2. READY (Fertig)	-	ON	-	-	-	-	-
3. START	-	-	ON	-	-	-	-
4. PAUSE	-	-	-	ON	-	-	-
5. Safety door open	-	-	-	-	ON	-	-
6. Emergency Stop	-	-	-	-	-	ON	-
7. Error	-	-	-	-	-	-	ON
8. Safety door open, emergency Stop	-	-	-	-	ON	ON	-
9. Safety door open, error	-	-	-	-	ON	-	ON
10. Emergency Stop, error	-	-	-	-	-	ON	ON
11. Safety door open, Emergency Stop, error	-	-	-	-	ON	ON	ON

„-“ entspricht OFF.

