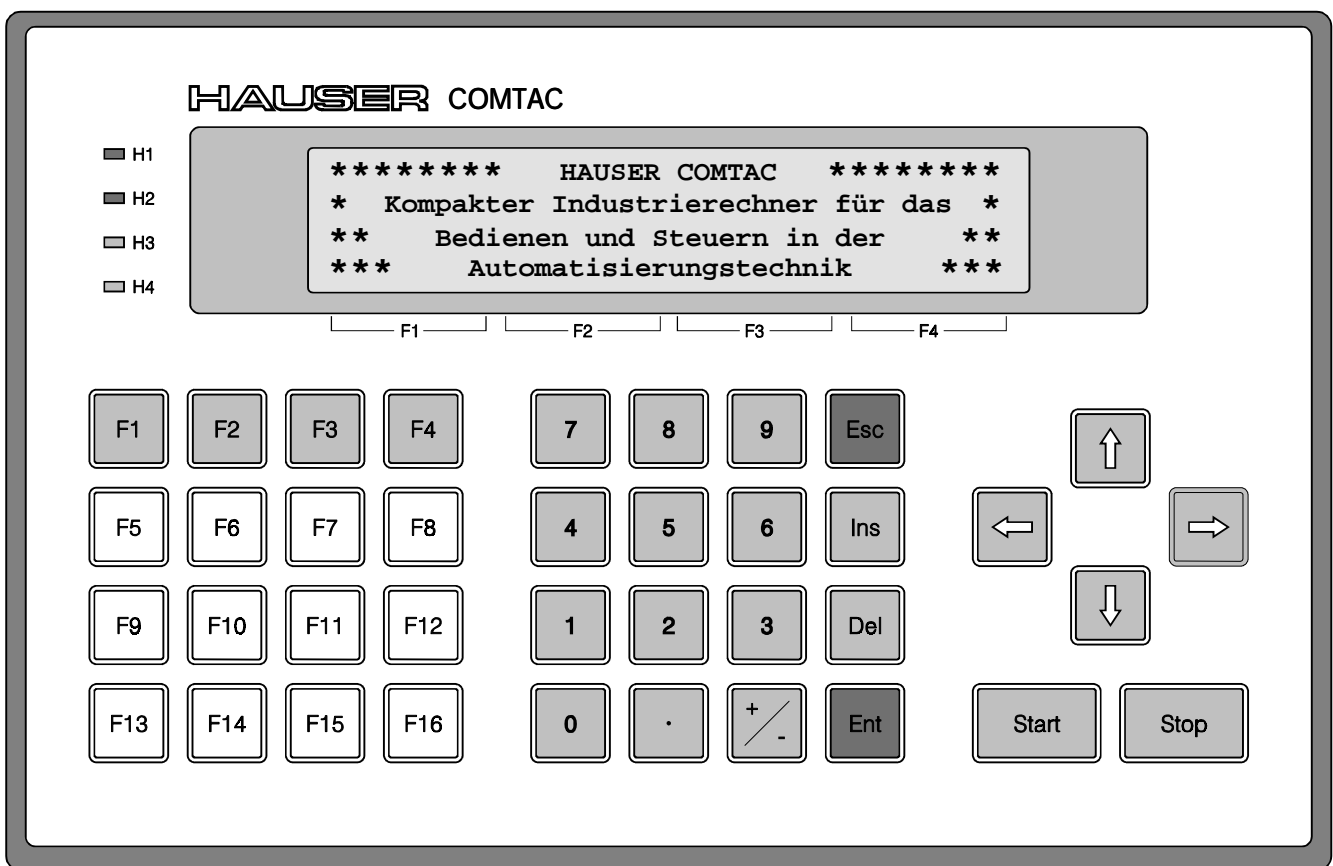


COMTAC-Software für die Automation - Funktionsbeschreibung -



Ab COMTAC - Softwareversion 2.00
Ab CTC-Pro-Version V1.00

Juli 2000

HAUSER
Wir automatisieren Bewegung



Parker Hannifin GmbH
EMD HAUSER
Postfach: 77607-1720
Robert-Bosch-Str. 22
D-77656 Offenburg, Germany
Tel.: +49 (0)781 509-0
Fax: +49 (0)781 509-176
<http://www.parker-emd.com>

1. Inhalt

1. Inhalt	2	9.3 Betriebsarten umschalten	35
2. Gerätezuordnung	3	9.4 COMTAC als Achssteuerung	36
3. Einsatzgebiete/Zielgruppen	4	9.4.1 Automatik	36
4. Entwicklungsschritte:	6	9.4.2 Hand	37
5. Voraussetzungen	8	9.4.3 Maschinennull suchen	37
5.1 Installation der COMTAC-Software für		9.5 COMTAC als E/A - Textanzeigeeinheit	38
die Automation "CTC_Pro"	8	10. Automatisierungstechniker	39
5.2 COMTAC-Konfiguration	10	10.1 Menü: [F5] Automatikbetrieb	39
5.3 Anlagenautomatisierung mit COMTAC		10.2 Zugriff auf Parameter und Positionen	39
und COMPAX.	11	10.3 Automatik-Betrieb / Achssteuerung	40
5.3.1 Aufbau des COMTAC - COMPAX -		10.3.1 Modul: Streckenpositionierung	40
Verbund mit dem Ablaufprogramm in		10.3.2 Modul: Bahnpositionierung	41
COMTAC	11	10.3.3 DB [F5] Automatik-Betrieb	42
5.3.2 Aufbau des COMTAC - COMPAX -		10.3.4 DH Eingabe Bearbeitungspro-	
Verbunds mit dem Ablaufprogramm in		gramm-Nummer	44
der SPS	12	10.3.5 DD Bearbeitungsprogramm laden	45
5.3.3 Notaus-Funktion	13	10.4 COMTAC / COMPAX Ausgänge setzen	45
5.3.4 Stop-Funktion	13	10.4.1 AF Ausgang als Impuls setzen	45
5.3.5 Konfiguration von COMPAX mit dem		10.4.2 AG Ausgang setzen	46
Servo-Manager	14	10.4.3 AH Ausgang löschen	46
5.4 COMTAC als Textanzeige	14	10.5 COMTAC / COMPAX Eingänge lesen	47
6. Texte anpassen	15	10.5.1 AI Eingang einlesen	47
7. Hauptprogramm	20	10.5.2 AJ Eingang einlesen mit Zeitabfrage	48
8. Die Betriebsarten	21	10.6 COMTAC Ein- und Ausgäbe	49
8.1 Einschalten von COMTAC und		10.6.1 AK Display anzeigen mit \$(11)..\$(14)	49
COMPAX	21	10.6.2 AY INPUT	49
8.1.1 Nullpositionen, Verfahrrichtungen der		11. Basic - Programmierer	51
Achsen	22	11.1 Übersicht	51
8.1.2 Ausschalten der Anlage	22	11.2 Variablen und Tabellen	51
8.2 [F6] Parametrierbetrieb	23	12. Anhang	53
8.3 [F7] Handbetrieb	26	12.1 A Grundfunktionen	53
8.4 [F9] M0 manuell	27	12.1.1 AA RS485-OUTPUT	53
8.5 [F10] Test E/A's	28	12.1.2 AB RS485-OUTPUT/ENTER	53
8.6 [F11] Korrektur Maßbezug	28	12.1.3 AC Achs-Positionierung starten	54
8.7 [F14] Parameter-Manager	29	12.1.4 AD Geschwindigkeit und Rampen	
8.8 [F13] CTC-Initialisierung	32	setzen	54
8.9 [F16] Anlagen-Info	32	12.1.5 AE POS-Feld Reset	55
8.10 [F15] Programm-Ende	32	12.1.6 AL Zeilen löschen	56
8.11 [F8] Einrichtbetrieb	32	12.1.7 AM Display Initialisierung	56
9. SPS-Programmierer	33	12.1.8 AN Fehler-Routine	57
9.1 Eingangsliste	33	12.1.9 AO COMPAX-Fehler-Routine	58
9.2 Ausgangsliste	34	12.1.10 AP NOTAUS-Routine	60
		12.1.11 AQ STOP-Routine	61
		12.1.12 AR Passwort Freigabe in Achse A	62
		12.1.13 AS Passwort Sperren in Achse A	62
		12.1.14 AT CPX-Parameter auslesen	63
		12.1.15 AU CPX-Status auslesen	63
		12.1.16 AV CPX-Parameter beschreiben	64
		12.1.17 AW Fehleranzeige und weiter mit	
		ENTER	64
		12.1.18 AX Dateibefehle	65
		12.1.19 AZ Achseingabe	65
		12.1.20 BA Weiter mit ENTER	66

12.2 C Hauptprogramm und Initialisierung	67
12.2.1 CA Hauptprogramm.....	67
12.2.2 CB String-Dimensionierung	69
12.2.3 CC String-Initialisierung	69
12.2.4 CD Variablen-Initialisierung	70
12.2.5 CE Funktionstasten sperren	70
12.2.6 CF Funktionstasten freigeben	71
12.2.7 CG Funktionstasten zuweisen	71
12.3 D Betriebsarten	73
12.3.1 D0 Grundsätzliches	73
12.3.2 DA Hauptmenü	73
12.3.2.1 DB [F5] Automatik-Betrieb.....	74
12.3.3 DC [F6] Parametrier-Betrieb	77
12.3.4 DD Parameter laden: Modul, Positi- onen laden.....	77
12.3.5 DE Maschinenparameter.....	78
12.3.6 DF Maschinenpositionen	79
12.3.7 DG Achsparameter.....	81
12.3.8 DI [F7] Hand-Betrieb.....	82
12.3.9 DJ [F9] Maschinen-Nullpunkt manu- ell	83
12.3.10 DK [F10] Test Eingänge/Ausgänge	84
12.3.11 DL [F11] Korrektur Maßbezug	85
12.3.12 DN [F13] COMTAC-Initialisierung.....	86
12.3.13 DO [F14] COMPAX-Parameter- Manager	86
12.3.14 DP [F16] Anlagen-Info	87
12.3.15 DQ Passwort-Eingabe	88
12.4 DR [F8] Einrichtbetrieb.....	90
12.5 DS [F15] Programmende.....	90
12.6 DT Anzeigesteuerung.....	91
12.7 E Steuerung SPS - COMTAC	91
12.7.1 EA Automatik über SPS.....	92
12.7.2 EB Handbetrieb über SPS.....	94
12.7.3 EC M0 anfahren über SPS	95
12.7.4 ED Bearbeitungsprogramm über SPS einlesen	96
12.7.5 EE Warten auf Start der SPS	97
12.7.6 EF Fehlerausgang setzen an SPS	98
12.7.7 F Optimierte 2-Achspositionierung	99
12.7.7.1 FA Bahnpositionierung mit 2 Achsen.....	99
12.7.8 FB Lade Konfiguration	99
13. Applikationsbeispiel:	101
14. Stichwortverzeichnis	104

2. Gerätezuordnung

Diese Dokumentation gilt für die Geräte:

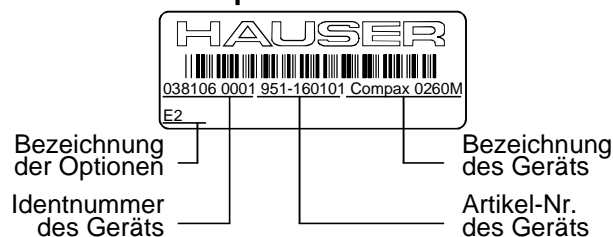
COMTAC 2000

COMTAC 3000

mit der COMTAC-Software für die Automation:

"CTC-Pro" (COMTAC - Professional)

**Grundsätzlicher Aufbau des HAUSER-Typen-
schilds am Beispiel:**



**Die COMTAC-Software für die Automation
"CTC-Pro" beinhaltet:**

- ◆ Diese Dokumentation und
- ◆ eine 3,5" - Diskette mit folgendem Inhalt:

• CTCT1S1.DAT	• CTCT1S2.DAT
• PART1S1.DAT	• PART1S2.DAT
• OUTT1S1.DAT	• OUTT1S2.DAT
• SPSTEXT.DAT	• MAP.DAT
• AXP.DAT	• CPXPAR34.DAT
• CPXPAR12.DAT	• CTCPARA.DAT
• GRENZEN.DAT	• CTCTEACH.BAS
• CTC_PRO.BAS	• CC_EDIT.BAS

Weitere Dokumentationen:

- ◆ COMTAC-Gerätebeschreibung
- ◆ Bedienungsanleitung zum COMTAC-Programmier-
tool.
- ◆ Befehlsbeschreibung zu COMTAC-Basic.



Für die Einhaltung der

◆ **EG - Maschinenrichtlinien und**

◆ **der harmonisierten Normen des jeweiligen Maschinentyps**

ist der Hersteller der Anlage verantwortlich!

Dazu kann es erforderlich sein, daß die Betriebsart "Einrichtbetrieb" (entspr. EG-Maschinenrichtlinien Anhang I Punkt 1.2.5) programmiert wird.

3. Einsatzgebiete/Zielgruppen

Die nachfolgend beschriebene COMTAC-Software für die Automation "CTC_Pro" entstand aus unserer langjährigen Erfahrung in der Automatisierungstechnik. Die Erkenntnisse aus einer Vielzahl von realisierten Projekten wurden hier in eine Software umgesetzt, die die Anlagenautomatisierung mit COMTAC und COMPAX wesentlich vereinfacht.

Einsatzgebiet:

- ◆ Anlagenautomatisierung mit COMTAC und COMPAX mit COMTAC oder SPS als Ablaufsteuerung.
- ◆ COMTAC als Textanzeige gesteuert von einer SPS über eine digitale E/A-Schnittstelle.

Zielgruppen: Die COMTAC-Software für die Automation "CTC_Pro" kann von mehreren Zielgruppen eingesetzt werden:

Automatisierungstechniker mit COMTAC - COMPAX - Verbund.

- Die Menüs der COMTAC-Software für die Automation "CTC_Pro" werden für die Anlagenbedienung eingesetzt.
- Im Menü "Parametrieren" werden Maschinenparameter, Positionen und Achsparameter programmiert (geteacht) und geräte-intern in Dateien gespeichert.
- Der Automatikbetrieb wird vom Automatisierungstechniker unter Einsatz vorhandener Module programmiert.
Vorhandene Module unterstützen: • Lesen von Eingängen • Definieren von Ausgängen • Positionieren der Achsen.
- Im Menü "Parameter-Manager" können Sie COMPAX Parameter editieren, in eine COMTAC - Datei schreiben oder von der Datei ins COMPAX übertragen.

➡ Für die Ablaufsteuerung sind COMTAC - Programmierkenntnisse notwendig - die Kommunikation mit COMPAX ist programmiert!

➡ Die Prozeß- und Ablaufsteuerung befindet sich im COMTAC.

SPS-Programmierer mit COMTAC - COMPAX - Verbund.

- Es steht ein lauffähiges COMTAC-Ablaufprogramm zur Verfügung, das über eine digitale E/A-Schnittstelle mit einer SPS kommuniziert.
- Im Menü "Parametrieren" werden Maschinenparameter, Positionen und Achsparameter programmiert (geteacht) und geräte-intern in Dateien gespeichert..
Über binär codierte Eingänge werden Programmnummer, Positionsnummer und Achsnummer von der SPS eingestellt. Das COMTAC übernimmt die eigentliche Positionieraufgabe und Achsüberwachung und meldet den Status der SPS zurück.
Werden zusätzliche E/A's zur Steuerung von Greifern, Sensoren, Zylindern, .. benötigt, so werden diese direkt von der SPS gesteuert.
- Betriebszustände, Fehler, etc. der SPS können über zusätzliche, binär codierte Eingänge angezeigt werden, was eine teure Anzeigeeinheit erspart.

➡ Es ist keine COMTAC - Programmierung notwendig!

➡ Die Prozeß- und Ablaufsteuerung befindet sich in der SPS.

SPS-Programmierer mit COMTAC als Textanzeige

- Über eine digitale E/A-Schnittstelle können Sie aus 100 Texten auswählen und COMTAC als Anzeigeeinheit nutzen.

➡ Es ist keine COMTAC - Programmierung notwendig!

Basic-Programmierer mit COMTAC - COMPAX - Verbund.

- Für spezielle Lösungen erhalten COMTAC-Basic-Programmierer eine hilfreiche Modulsammlung.
- Die Programme sind offen zugänglich und können vom Programmierer angepasst werden oder zeigen Lösungsbeispiele.
- Die Ablagestruktur für Daten und Text kann genutzt werden.

➡ Für die Ablaufsteuerung sind COMTAC - Programmierkenntnisse notwendig - die Kommunikation mit COMPAX ist programmiert!

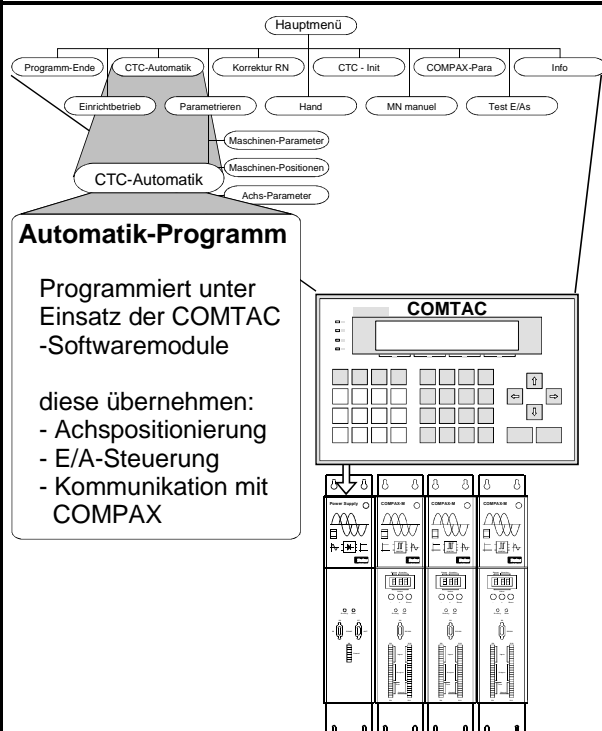
➡ Die Prozeß- und Ablaufsteuerung befindet sich im COMTAC.

Zielgruppen

COMTAC-Software für die Automation "CTC_Pro" bietet Einsatzmöglichkeiten für 4 Zielgruppen:

Automatisierungstechniker mit COMTAC - COMPAX - Verbund

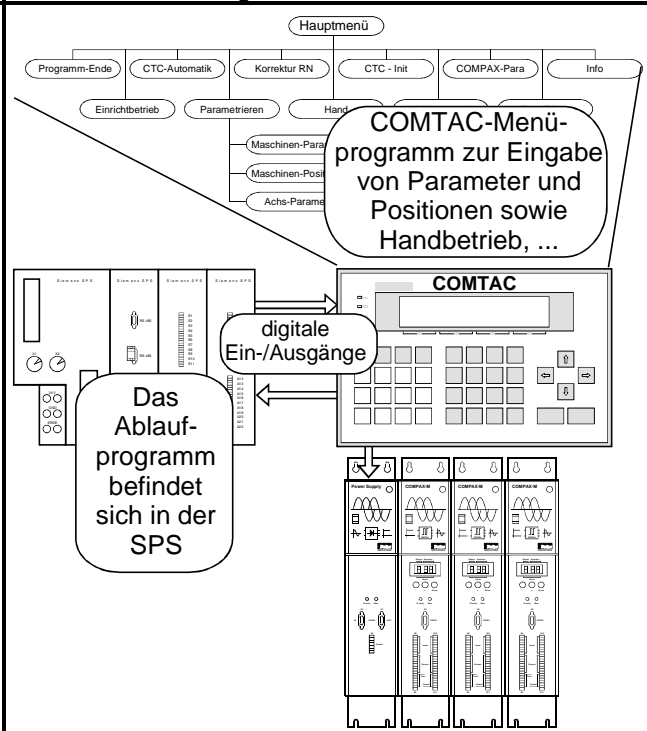
Sie erhalten eine komplett programmierte Oberfläche. Der Automatikbetrieb und evtl. den Einrichtbetrieb programmieren Sie mit Hilfe von Basic - Modulen für die Kommunikation mit COMPAX und mit COMTAC - Basic.



SPS-Programmierer mit COMTAC - COMPAX - Verbund

Sie verwenden die programmierte Oberfläche zur Eingabe von Positionswerten, Maschinen- und Achsparameter.

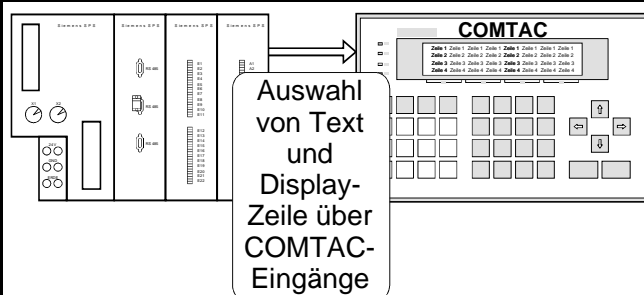
Der Automatikbetrieb wird in der SPS programmiert; das gesamte Ablaufprogramm steht damit in der SPS. Über E/A-Steuerung werden Achsbefehle aktiviert.



SPS-Programmierer mit COMTAC als Textanzeige

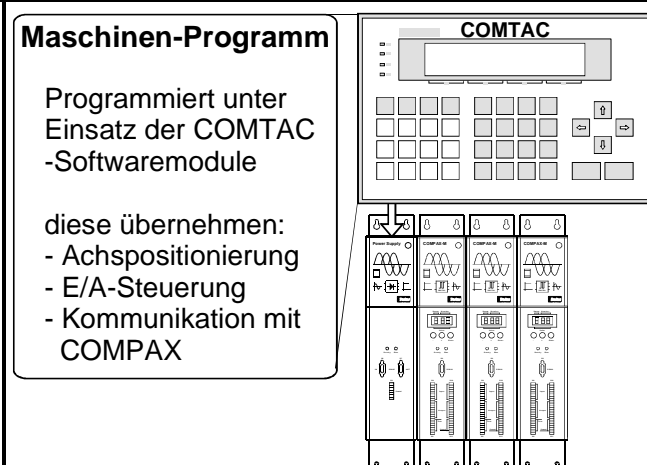
Über digitale COMTAC - Eingänge erfolgt die Text- und Zeilenauswahl.

Eine Programmierung von COMTAC ist nicht notwendig



Basic-Programmierer mit COMTAC - COMPAX - Verbund

Sie programmieren die Oberfläche und den Automatikbetrieb wie gewohnt und benutzen dazu die in COMTAC-Software für die Automation "CTC_Pro" vorhandenen Module für die Kommunikation mit COMPAX und mit COMTAC - Basic.



4. Entwicklungsschritte:

Automatisieren mit COMTAC:

Schritte zum Automatisieren einer Anlage mit der COMTAC-Software für die Automation "CTC_Pro"		
Hilfsmittel	Ablauf	Weitere Informationen
PC mit Windows und dem Servo - Manager	COMPAX konfigurieren.	Produkthandbuch COMPAX und Seite 14
	COMTAC mit COMPAX - Verbund und PC verbinden	Siehe Seite 11
PC und die Software: "COMTAC Programmiertool"	COMTAC Programmiertool installieren.	Bedienungsanleitung COMTAC Programmiertool
COMTAC Programmiertool	Software "CTC_Pro" installieren.	Siehe Seite 8
COMTAC Programmiertool	Programm laden: ♦ DIR R + [ENT] eingeben. ♦ F2 laden wählen. ♦ mit Cursortasten den Dateinamen "CTC_PRO.BAS" auswählen. ♦ [ENT] drücken. Programm starten: ♦ F3 drücken oder RUN + [ENT] eingeben.	Siehe Seite 8
	Beim 1. Starten des Programms werden Grundeinstellungen abgefragt.	Siehe Seite 8
	Nach dem 1. Starten Menü "COMTAC Init" [F13] aufrufen: es werden automatisch Grundeinstellungen vorgenommen.	Siehe Seite 10
Nun sind alle Betriebsarten außer dem Automatikbetrieb betriebsbereit! Sie können z. B. die COMPAX - Achsen per Hand verfahren. Voraussetzung ist selbstverständlich, daß die COMPAX - Achsen betriebsbereit sind (Versorgung angelegt, richtig Konfiguriert; siehe Proukthandbuch COMPAX)		
"CTC_Pro" mit RUN + [ENT] oder F3 starten.	Maschinenpositionen, Maschinenparameter und Achsdaten eingeben.	Siehe ab Seite 23



Bei Bedarf können Sie Maschinentexte Ihrer Anwendung anpassen (Siehe Seite 15).

An dieser Stelle hängt das weitere Vorgehen von der Art der Anwendung ab:

Bei Prozeß- und Ablaufprogrammierung in COMTAC gilt:

"CTC_Pro" mit [STRG c] oder F15 verlassen.	<p>Automatikbetrieb programmieren: Programm CTC_PRO.BAS ab Zeile 20300 auflisten (In den Zeilen 20300 ... 20550 stehen Beispiele für Automatikfunktionen).</p> <p>Von Zeile 20300 bis 24000 können Sie nun Ihr Automatikbetrieb programmieren. Dabei können Sie auf vorhandene Module zur COMTAC - COMPAX E/A-Steuerung und zur COMPAX - Positionierung verwenden.</p>	Siehe ab Seite 39
--	---	-------------------

 **Speichern Sie regelmäßig während der Programmerstellung!**

Bei Prozeß- und Ablaufprogrammierung in der SPS gilt:

	Verdrahten der E/As zwischen COMTAC und der SPS.	Siehe ab Seite 33
	Programmerstellung in der SPS.	Siehe ab Seite 33

COMTAC als Textanzeige

	Verdrahten der E/As zwischen COMTAC und der SPS.	Ein-/ Ausgangsliste siehe ab Seite 38
"CTC_Pro" mit [STRG + C] oder F15 verlassen. DATA-Text-Editor aufrufen.	Texteingabe in die Textdatei SPSTEXT.DAT	Siehe ab Seite 38 und im der Beschreibung zum COMTAC Programmiertool.
	Texte und Zeilen, in denen Sie angezeigt werden sollen, über die COMTAC - Eingänge auswählen	Siehe Seite 38

Autostart:

COMTAC-Software für die Automation "CTC_Pro" als Autostartprogramm definieren:

COMTAC Programmiertool	Das Programm "CTC_PRO.BAS" umbenennen in "CTC_PRO.ASP" (*.ASP definiert das Programm als Autostartprogramm) und die Autostartfunktion mit CT.(13) Bit 8 = "1" einschalten.	
------------------------	--	--

5. Voraussetzungen

5.1 Installation der COMTAC-Software für die Automation "CTC_Pro"

- Achtung:** Bevor die COMTAC-Software für die Automation "CTC_Pro" installiert wird, muß die Programmierumgebung "CTCPT" auf einen IBM-kompatiblen PC installiert werden.
- Installation:** Das Programm und die Dateien des CTC_PRO befinden sich auf einer Diskette. Alle Dateien müssen ins ZP-Ram des COMTAC kopiert werden. Dazu gibt es zwei Möglichkeiten (siehe CTC-Befehlsbeschreibung):
- Mit Diskettenlaufwerk am COMTAC**
- ◆ Mit Hilfe eines Diskettenlaufwerkes, das ans COMTAC angeschlossen ist, alle Dateien von Laufwerk A (B) nach R kopieren:
 - CTCPT aufrufen (evt. Baudrate mit [F10] einstellen)
 - DIR A + [ENT] eingeben
 - Datei mit den Cursortasten auswählen
 - F5 kopieren wählen
 - F1 für Datei auswählen
 - mit Cursortasten <-, -> R (ZP-Ram) anwählen
 - [ENT] drücken
 - Ablauf mit allen Dateien wiederholen
- Mit dem Programmier-PC**
- ◆ Per Download von dem Programmier-PC ins ZP-Ram des COMTAC. Dazu müssen zuerst alle Dateien von der Diskette auf die Festplatte des PC's kopiert werden.
 - CTCPT aufrufen (evt. Baudrate mit [F10] einstellen)
 - [ALT+F5] drücken für Download TOKEN für *.BAS und [ALT+F7] für *.DAT
 - Datei auswählen; [ENT] drücken; Y drücken zur Bestätigung
 - [ENT] drücken zum Start der Download-Funktion; Y drücken zur Bestätigung
 - Download wird ausgeführt (sichtbar an der hochlaufenden Byteanzahl)
 - Quittieren mit 2 mal [ESC]
 - DIR R + [ENT] eingeben oder [F5]
 - F3 speichern wählen
 - den Dateinamen eingeben (wie zuvor ausgewählt)
 - [ENT] drücken
 - Ablauf mit allen Dateien wiederholen
- Voraussetzung**
- ◆ Bevor das Programm gestartet werden kann, müssen Verbindungen vom COMTAC zum Programmier-PC und zu den COMPAX-Achsen hergestellt werden.
 - ◆ Ebenso müssen die COMPAX-Achsen zuvor auf die Applikation konfiguriert werden (siehe Produkthandbuch COMPAX). Um die COMPAX-Achsen über den Feldbus anzusprechen, sind Parameter im COMPAX einzustellen (siehe Seite 14).
 - ◆ Der Eingang 1 oder Eingang "RDY" des COMTAC ist mit dem Signal für 'STEUERUNG EIN' der Anlage zu verbinden.
- Laden:** DIR R + [ENT] eingeben
- ◆ F2 LOAD wählen
 - ◆ mit Cursortasten den Dateinamen "CTC_PRO.BAS" auswählen
 - ◆ [ENT] drücken

5.1 Installation der COMTAC-Software für die Automation "CTC_Pro"

◆ F3 drücken oder RUN + [ENT] eingeben

Starten:

Beim erstmaligen Start müssen die Grundeinstellungen eingegeben werden. Diese Daten werden im Programm "CTCTEACH.BAS" (startet automatisch) eingegeben. Die Daten werden in CONTROL-Worten des COMTAC ab CT.(130) gespeichert.

Parameter:	Bedeutung	Wertebereich
CT.(130)	Dialogsprache	deutsch (1) oder englisch (2)
CT.(131)	Anzahl der Achsen	1-4
CT.(132)	Anzahl der Maschinenpositionen	max. 31
CT.(133)	Anzahl der Bearbeitungsprogramme	max. 600/CT.(132) /CT.(131)
CT.(134)	Anzahl der Maschinenparameter	max. 44
CT.(135)	Anzahl der Achsparameter	max. 25 pro Achse
CT.(136)	Automatikmodus COMTAC: ="0" SPS: ="1" Anzeige: ="2"	Ablauf in der SPS oder in COMTAC
CT.(137)	Paßwort der Paßwortebene 0	302
CT.(138)	Paßwort der Paßwortebene 1	302
CT.(139)	Paßwort der Paßwortebene 2	302

Nach dem Starten des Programmes wartet das COMTAC im Hauptmenü auf die Wahl der Betriebsart. Hier sind vor dem ersten Starten des Automatikbetriebes die Maschinenparameter, Maschinenpositionen und Achsparameter einzugeben.

Dateistruktur:

CTCT1S1(2).DAT: Dialogtexte der Betriebsarten. ("1" für deutsch; "2" für englisch)
PART1S1(2).DAT: 1-25 Texte der Achsparameter; 26-69 Texte der Maschinenparameter; 70-100 Texte der Maschinenpositionen.

OUTT1S1(2).DAT: Texte der Ausgänge COMTAC (1-16/32) und COMPAX (4*16).
SPSTEXT.DAT: 100 Texte, die im SPS-Modus und im Anzeigemodus während des Automatikbetriebes angezeigt werden.

MAP.DAT: Feld, in dem max. 44 Maschinenparameter (Speicher 601-645) und max. 31 Maschinenpositionen je Achse pro Bearbeitungsprogramm (Speicher 1..600) gespeichert sind. Die Maschinenpositionen können auf max. 127 Bearbeitungsprogramme verteilt werden. Der Speicher von 600 Speicherstellen ist dabei zu beachten.

z.B.: 25 Positionen * 10 Bearbeitungsprogramme = 250 Speicherstellen bei einer Achse.

AXP.DAT: Feld, in dem die Achsparameter (Speicher 1..100) von max. 4 Achsen gespeichert werden.

CPXPAR34.DAT: Parameterdatei des Parameter-Manager (COMPAX -Achsen 3,4)

CPXPAR12.DAT: Parameterdatei des Parameter-Manager (COMPAX -Achsen 1,2)

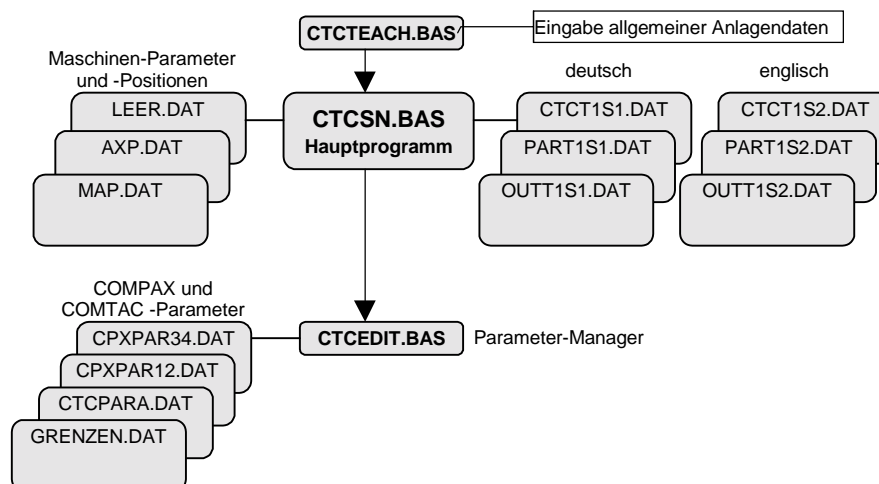
CTCPARA.DAT: Parameterdatei des Parameter-Manager

GRENZEN.DAT: Parameterdatei des Parameter-Manager

CTCTEACH.BAS: Eingabeprogramm zur Eingabe der allgemeinen Anlagendaten (CT.(130) ...CT.(139); siehe oben).

CTC_PRO.BAS: Programm COMTAC-Software für die Automation "CTC_Pro".

CC_EDIT.BAS: Programm "Parameter-Manager"; Untermenü der "CTC_Pro"



- Beachte:**
1. Die Textlänge der Texte in den Textfiles, die durch einen "\" gekennzeichnet ist, muß eingehalten werden.
 2. Werden die Grundeinstellungen verändert (Anzahl der Achsen, Anzahl Positionen, ...), so müssen wegen Speicherkollisionen die Datenfiles AXP.DAT und MAP.DAT gelöscht werden. Dies kann durch Kopieren der Datei LEER.DAT auf die Dateien AXP.DAT und MAP.DAT durchgeführt werden.
- Hinweis:**
- Dieses Softwaremodul entstand aus jahrelanger Erfahrung in der Automatisierungstechnik der Firma Parker Hannifin, Elektromechanik Division. Es deckt die Basis der Applikationen ab, schließt jedoch Anpassungen nicht aus. Grundsätzlich sind Schulungen in die Programmierung des COMTAC und die Funktionsweise der Feldbussysteme in Verbindung mit den digitalen Servoreglern immer zu empfehlen.

5.2 COMTAC-Konfiguration

Die nachfolgend beschriebenen Einstellungen sind Empfehlungen zum Arbeiten mit der COMTAC-Software für die Automation "CTC_Pro".
Die Einstellungen werden beim Aufruf der Betriebsart "COMTAC - Initialisierung" (F13) automatisch vorgenommen.

Systemparameter:

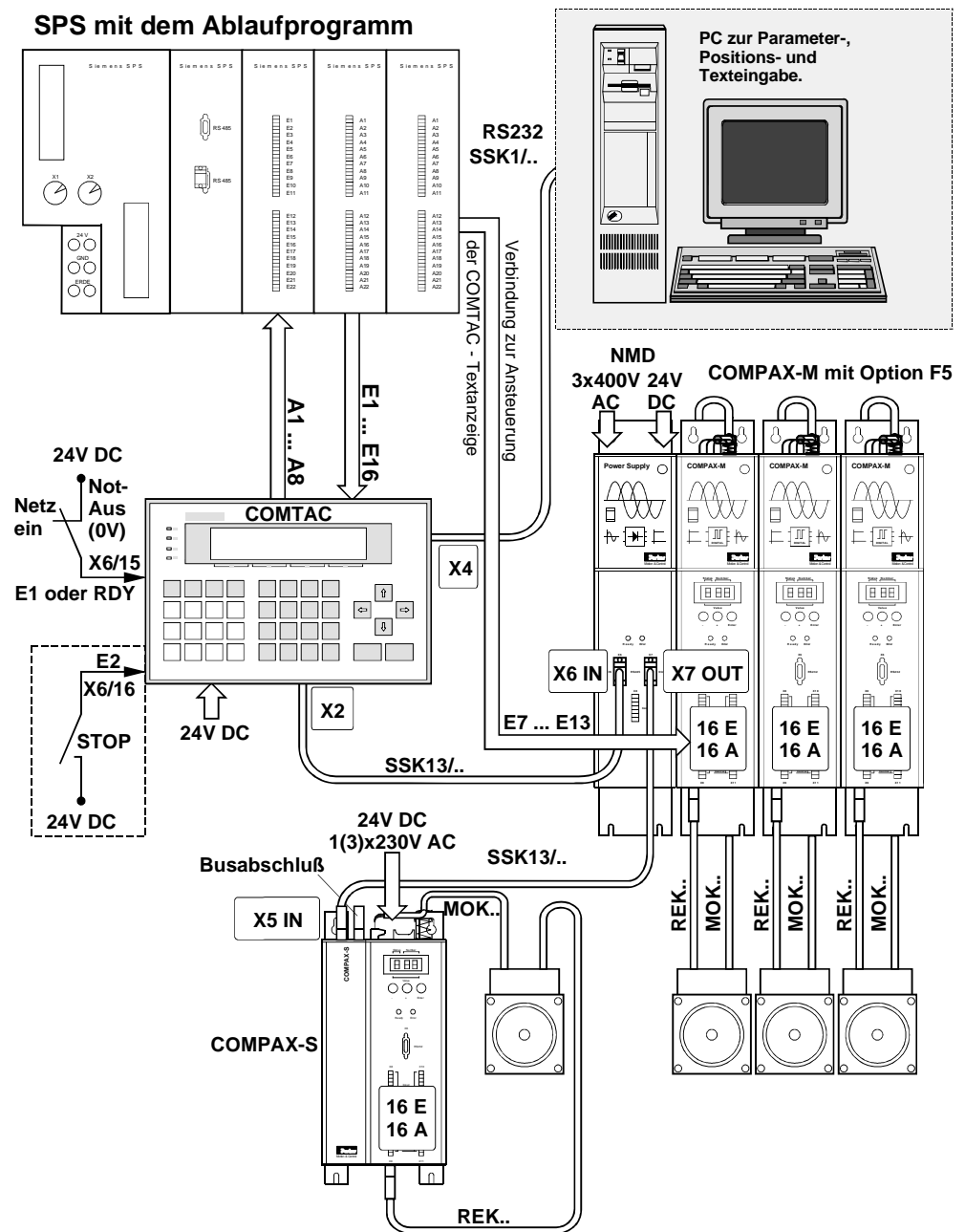
- | | |
|--------------------------------|--|
| ◆CONTROL(2)=0 | ! TV905 |
| ◆CONTROL(5)=0 | ! Power-On Zustand A1-16=0 |
| ◆CONTROL(6)= 0 | ! Power-On Zustand A17-32=0 |
| ◆CONTROL(7)=50 | ! Einschaltverzögerung RS485-Abfrage (1=100ms) |
| ◆CONTROL(8)=10 | ! Reqout-Timeout (A->E) (1=100ms) |
| ◆CONTROL(9)=3 | ! CTC-Repeat-Tastaturzeit (1=100ms) |
| ◆CONTROL(13)=0010000100000100B | ! Systemflags |
| ◆CONTROL(70)=2(3,4) | ! Teilnehmer Feldbus |

HAUSER

5.3.1 Aufbau des COMTAC - COMPAX - Verbund mit dem Ablaufprogramm in COMTAC



5.3.2 Aufbau des COMTAC - COMPAX - Verbunds mit dem Ablaufprogramm in der SPS

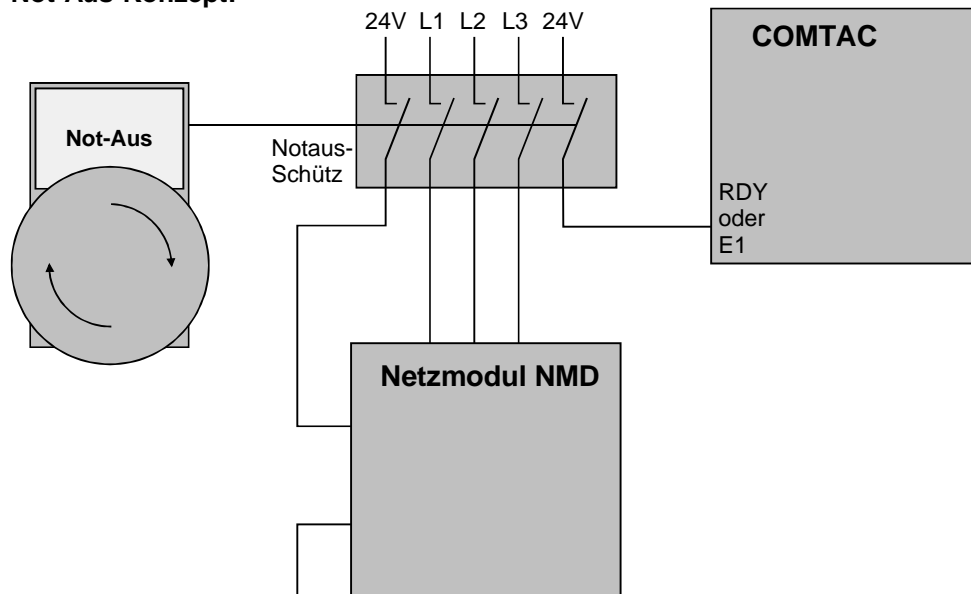


5.3.3 Notaus-Funktion

Voraussetzung

Verdrahten von E1 oder RDY mit Netz Ein / Not Aus.

Not-Aus-Konzept:



Hardware-Funktion:

Die Leistungsspannung des Netzmoduls und somit der COMPAX-Module wird weggeschaltet. Die Achsen werden über eine einstellbare Rampenzeit abgebremst und kommen zum Stillstand. Bei vertikalen Achsen wird die Bremse zugeschaltet. Dabei muß P17 der COMPAX-Achsen mit ~ 300 msec programmiert werden. Die Spannung (24V) der Ausgänge bleibt erhalten. In der Anzeige erscheint die Meldung 'NOTAUS'.

Quittieren der NOTAUS-Funktion:

◆ NOTAUS Pilztaster verdrahtet mit E1 oder RDY (siehe Seite 11).

➡ Pilztaster lösen, Steuerung Ein drücken.

Bedienung der NOTAUS-Funktion:

Die erste NOTAUS-Meldung wird am COMTAC angezeigt. Zum Quittieren ist wie oben beschrieben zu verfahren. Anschließend wird in die Stop-Funktion verzweigt. Wurde NOTAUS im Automatikbetrieb ausgelöst, dann besteht nach dem Quittieren die Möglichkeit den Ablauf abubrechen oder fortzusetzen:

NOTAUS im Automatik:

◆ Nach Quittieren Taste 'Start' drücken:

➡ Automatikbetrieb wird fortgesetzt.

◆ Nach Quittieren Taste '[STOP]' drücken:

➡ Automatikbetrieb wird abgebrochen; Es wird ins Hauptmenü 'Anwahl der Betriebsarten mit FKT.-Tasten' verzweigt.

5.3.4 Stop-Funktion

Voraussetzung

Verdrahten von E2 mit einer externen [STOP]-Taste (siehe Seite 11) oder Sie verwenden die [STOP]-Taste des COMTAC Displays.

Hardware-Funktion:

Durch Drücken der Taste '[STOP]' wird ein Eingang am COMTAC aktiviert, der wiederum intern veranlaßt, daß die Achsen über eine programmierbare Rampenzeit abgebremst werden. Die Taste '[STOP]' ist für ca. 1 sec. zu drücken, um die Funktion auszulösen.

Quittieren und Bedienen der STOP-Funktion:

Am COMTAC erscheint die Meldung 'STOP' . Es besteht die Möglichkeit den Ablauf abubrechen oder fortzusetzen:

STOP im Automatik:

- ◆ Nach Quittieren Taste 'Start' drücken:



Automatikbetrieb wird fortgesetzt.

- ◆ Nach Quittieren Taste '[STOP]' drücken:



Automatikbetrieb wird abgebrochen; Es wird ins Hauptmenü 'Anwahl der Betriebsarten mit FKT.-Tasten' verzweigt.

Beachte:

Wird während der 'STOP'-Anzeige NOTAUS gedrückt, so wird 'NOTAUS' angezeigt. Zum Quittieren ist die Taste STEUERUNG EIN zu drücken und anschließend, wie oben beschrieben, zu verfahren.

5.3.5 Konfiguration von COMPAX mit dem Servo-Manager

Sämtliche COMPAX müssen mit dem Servo-Manager anlagenspezifisch konfiguriert werden.

Speziell zur Kommunikation mit COMTAC über den Feldbus müssen folgende COMPAX - Parameter definiert werden:

- ◆ P194: Adresse des Geräts

Die COMPAX - Adressen müssen von 1 fortlaufend bis zur Achsanzahl eingestellt werden. Bsp. bei 3 Achsen : Adresse 1,2,3

- ◆ P195: Feldbus-Baudrate (P195=345 600).

- ◆ P196: Betriebsart (P196=165).

Diese Einstellungen können Sie über die COMPAX - Frontplatte vornehmen. Eingestellt werden muß:

C01: Geräteadresse

C02: Baudrate; Einstellwert ist C02=345.

C03: Busprotokoll=165

Die Eingabe wird angewählt durch:

[+], [+], [Enter] nun können Sie C-Parameter mit [+] oder [-] anwählen und mit [Enter] auswählen.

Der gewünschte Wert wird mit [+] oder [-] eingestellt und mit [Enter] bestätigen.

mit [+] steht wieder "Run" in der Anzeigen (näheres siehe im Produkthandbuch COMPAX M/S).

5.4 COMTAC als Textanzeige

Neben einer E/A-Verbindung zu COMTAC und der Definition der Texte sind bei dieser Anwendung keine besonderen Voraussetzungen notwendig (siehe Seite 38).

6. Texte anpassen

Texte werden mit dem COMTAC Programmierwerkzeug eingegeben. Dazu müssen Sie die COMTAC-Software für die Automation "CTC_Pro" mit "[Strg c]" oder [F15] verlassen.

- ♦ Für Dialogtexte bestehen bereits 6 Textdateien: 3 Dateien jeweils in deutsch und englisch.
- ♦ Die Texte sind bereits belegt; lassen sich jedoch an die Automatisierungsaufgabe anpassen.

Die Textdateien (*.DAT) werden im DATA-Text-Editor bearbeitet. Der Editor wird mit dem Befehl "DTFEDIT" oder der Tastenkombination [Strg+Page up] aufgerufen. Danach können Sie die Texte Ihrer Applikation anpassen.

Dabei ist wichtig, daß Sie den Text immer mit dem Endezeichen "\" abschließen.

Außerdem sollten Sie in den Texten für die Maschinenparameter, Achsparameter (ab Achsparameter 11) und Maschinenpositionen jeweils die Nummer (Parameter- bzw. Positionsnummer) stehen lassen, da diese Größen im Programm über die Nummer angesprochen werden. Die Parameter- bzw. Positionsnummer stellt dann den Bezug her zwischen eingegebenem Wert und dem Ansprechen dieses Wertes im Programm.

Textdatei PARTS1T1.DAT

Diese Texte werden ausschließlich bei der Eingabe (Menü Parametrieren) der Maschinenparameter, Achsparameter und Maschinenpositionen verwendet.

Struktur:

PARTS1T1.DAT-Texte:	
Text-Nr.:	
1-25	25 Texte für die Achsparameter: davon sind die ersten 10 Texte den im Automatikprogramm verwendeten festen Achsparametern zugeordnet. Die Texte sind unabhängig von der Achsnummer für alle Achsen gleich.
26-69	44 Texte zu den Maschinenparametern.
70-100	31 Texte zu den Maschinenpositionen. Die Texte sind für alle Bearbeitungsprogramme gleich.

Textdatei OUTT1S1.DAT

Diese Texte werden ausschließlich im Menüpunkt "Test E/As" bei der Anzeige der E/A - Zustände verwendet.

Aufbau der Textdateien:

PART1S1.DAT
Texte für Maschinenparameter, Maschinenpositionen und Achspositionen.
Auto-Geschw. schnell [%]\
Auto-Beschl. schnell [ms]\
Auto-Geschw. langsam [%]\
Auto-Beschl. langsam [ms]\
Hand-Geschw. langsam [%]\
Hand-Geschw. schnell [%]\
Hand-Rampe langsam [ms]\
Hand-Rampe schnell [ms]\
Max. Geschwindigkeit [%]\
Min. Rampe [ms]\
Achsparameter 11 \
Achsparameter 12 \
Achsparameter 13 \
Achsparameter 14 \
Achsparameter 15 \

OUTT1S1.DAT
Texte zur Bezeichnung der digitalen Ausgänge
Ausgang Comtac 1\
Ausgang Comtac 2\
Ausgang Comtac 3\
Ausgang Comtac 4\
Ausgang Comtac 5\
Ausgang Comtac 6\
Ausgang Comtac 7\
Ausgang Comtac 8\
Ausgang Comtac 9\
Ausgang Comtac 10\
Ausgang Comtac 11\
Ausgang Comtac 12\
Ausgang Comtac 13\
Ausgang Comtac 14\
Ausgang Comtac 15\

PART1S1.DAT
Achsparemeter 16 \
Achsparemeter 17 \
Achsparemeter 18 \
Achsparemeter 19 \
Achsparemeter 20 \
Achsparemeter 21 \
Achsparemeter 22 \
Achsparemeter 23 \
Achsparemeter 24 \
Achsparemeter 25 \
Maschinenparam. 1 [Ug=0/Og=100]:\
Maschinenparam. 2 [Ug=0/Og=100]:\
Maschinenparam. 3 [Ug=0/Og=100]:\
Maschinenparam. 4 [Ug=0/Og=100]:\
Maschinenparam. 5 [Ug=0/Og=100]:\
Maschinenparam. 6 [Ug=0/Og=100]:\
Maschinenparam. 7 [Ug=0/Og=100]:\
Maschinenparam. 8 [Ug=0/Og=100]:\
Eingangs-Checkzeit [Ug=0/Og=100]:\
Bahngeschwindigk. [Ug=1/Og=100]:\
Bahnrampe [Ug=1/Og=5000]:\
Maschinenparam. 12 [Ug=0/Og=100]:\
Maschinenparam. 13 [Ug=0/Og=100]:\
Maschinenparam. 14 [Ug=0/Og=100]:\
Maschinenparam. 15 [Ug=0/Og=100]:\
Maschinenparam. 16 [Ug=0/Og=100]:\
Maschinenparam. 17 [Ug=0/Og=100]:\
Maschinenparam. 18 [Ug=0/Og=100]:\
Maschinenparam. 19 [Ug=0/Og=100]:\
Maschinenparam. 20 [Ug=0/Og=100]:\
Maschinenparam. 21 [Ug=0/Og=100]:\
Maschinenparam. 22 [Ug=0/Og=100]:\
Maschinenparam. 23 [Ug=0/Og=100]:\
Maschinenparam. 24 [Ug=0/Og=100]:\
Maschinenparam. 25 [Ug=0/Og=100]:\
Maschinenparam. 26 [Ug=0/Og=100]:\
Maschinenparam. 27 [Ug=0/Og=100]:\
Maschinenparam. 28 [Ug=0/Og=100]:\
Maschinenparam. 29 [Ug=0/Og=100]:\
Maschinenparam. 30 [Ug=0/Og=100]:\
Maschinenparam. 31 [Ug=0/Og=100]:\
Maschinenparam. 32 [Ug=0/Og=100]:\
Maschinenparam. 33 [Ug=0/Og=100]:\
Maschinenparam. 34 [Ug=0/Og=100]:\
Maschinenparam. 35 [Ug=0/Og=100]:\
Maschinenparam. 36 [Ug=0/Og=100]:\
Maschinenparam. 37 [Ug=0/Og=100]:\
Maschinenparam. 38 [Ug=0/Og=100]:\
Maschinenparam. 39 [Ug=0/Og=100]:\
Maschinenparam. 40 [Ug=0/Og=100]:\
Maschinenparam. 41 [Ug=0/Og=100]:\
Maschinenparam. 42 [Ug=0/Og=100]:\
Maschinenparam. 43 [Ug=0/Og=100]:\
Maschinenparam. 44 [Ug=0/Og=100]:\
Masch-Pos. 1:\
Masch-Pos. 2:\

OUTT1S1.DAT
Ausgang Comtac 16\
Ausgang Comtac 17\ ¹
Ausgang Comtac 18\
Ausgang Comtac 19\
Ausgang Comtac 20\
Ausgang Comtac 21\
Ausgang Comtac 22\
Ausgang Comtac 23\
Ausgang Comtac 24\
Ausgang Comtac 25\
Ausgang Comtac 26\
Ausgang Comtac 27\
Ausgang Comtac 28\
Ausgang Comtac 29\
Ausgang Comtac 30\
Ausgang Comtac 31\
Ausgang Comtac 32\
Ausgang Compax 1-1\
Ausgang Compax 1-2\
Ausgang Compax 1-3\
Ausgang Compax 1-4\
Ausgang Compax 1-5\
Ausgang Compax 1-6\
Ausgang Compax 1-7\
Ausgang Compax 1-8\
Ausgang Compax 1-9\
Ausgang Compax 1-10\
Ausgang Compax 1-11\
Ausgang Compax 1-12\
Ausgang Compax 1-13\
Ausgang Compax 1-14\
Ausgang Compax 1-15\
Ausgang Compax 1-16\
Ausgang Compax 2-1\
Ausgang Compax 2-2\
Ausgang Compax 2-3\
Ausgang Compax 2-4\
Ausgang Compax 2-5\
Ausgang Compax 2-6\
Ausgang Compax 2-7\
Ausgang Compax 2-8\
Ausgang Compax 2-9\
Ausgang Compax 2-10\
Ausgang Compax 2-11\
Ausgang Compax 2-12\
Ausgang Compax 2-13\
Ausgang Compax 2-14\
Ausgang Compax 2-15\
Ausgang Compax 2-16\
Ausgang Compax 3-1\
Ausgang Compax 3-2\
Ausgang Compax 3-3\
Ausgang Compax 3-4\
Ausgang Compax 3-5\
Ausgang Compax 3-6\
Ausgang Compax 3-7\

¹ Ausgänge 17-32 für COMTAC 3000.

PART1S1.DAT
Masch-Pos. 3:\
Masch-Pos. 4:\
Masch-Pos. 5:\
Masch-Pos. 6:\
Masch-Pos. 7:\
Masch-Pos. 8:\
Masch-Pos. 9:\
Masch-Pos.10:\
Masch-Pos.11:\
Masch-Pos.12:\
Masch-Pos.13:\
Masch-Pos.14:\
Masch-Pos.15:\
Masch-Pos.16:\
Masch-Pos.17:\
Masch-Pos.18:\
Masch-Pos.19:\
Masch-Pos.20:\
Masch-Pos.21:\
Masch-Pos.22:\
Masch-Pos.23:\
Masch-Pos.24:\
Masch-Pos.25:\
Masch-Pos.26:\
Masch-Pos.27:\
Masch-Pos.28:\
Masch-Pos.29:\
Masch-Pos.30:\
Masch-Pos.31:\

OUTT1S1.DAT
Ausgang Compax 3-8\
Ausgang Compax 3-9\
Ausgang Compax 3-10\
Ausgang Compax 3-11\
Ausgang Compax 3-12\
Ausgang Compax 3-13\
Ausgang Compax 3-14\
Ausgang Compax 3-15\
Ausgang Compax 3-16\
Ausgang Compax 4-1\
Ausgang Compax 4-2\
Ausgang Compax 4-3\
Ausgang Compax 4-4\
Ausgang Compax 4-5\
Ausgang Compax 4-6\
Ausgang Compax 4-7\
Ausgang Compax 4-8\
Ausgang Compax 4-9\
Ausgang Compax 4-10\
Ausgang Compax 4-11\
Ausgang Compax 4-12\
Ausgang Compax 4-13\
Ausgang Compax 4-14\
Ausgang Compax 4-15\
Ausgang Compax 4-16\

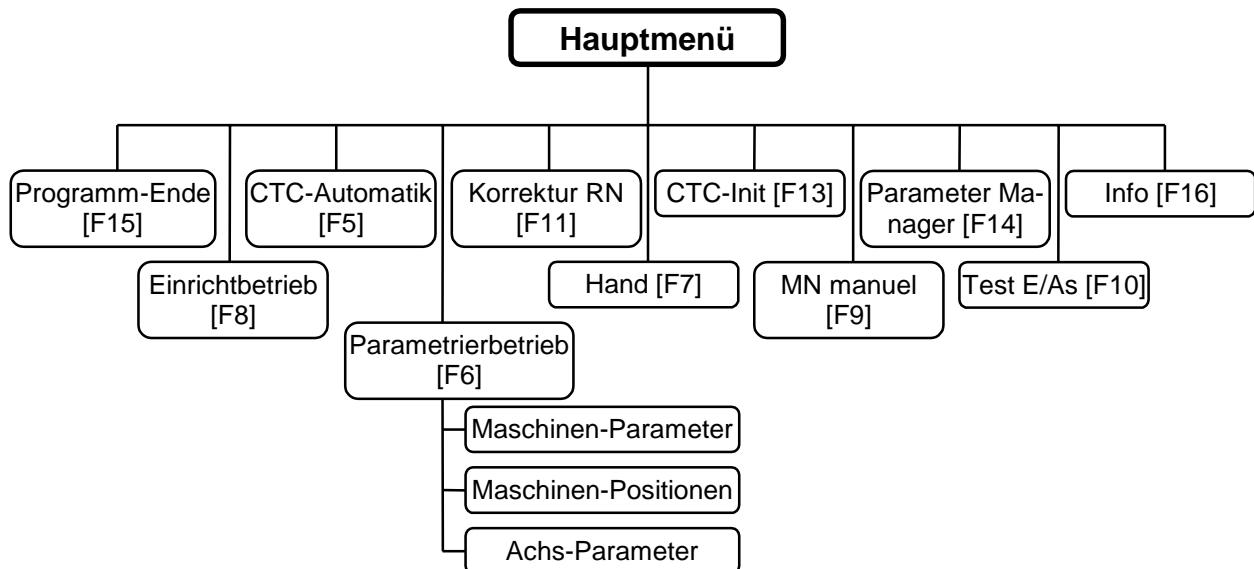
CTCT1S1.DAT
Dialogtexte für die COMTAC-Software für die Automation "CTC_Pro"
Anwahl der Betriebsarten mit FKT.TASTEN\ 20
HAUPTMENUE\ 21
Funktionstaste F1\ 22
Funktionstaste F2\ 23
Funktionstaste F3\ 24
Funktionstaste F4\ 25
Automatikbetrieb\ 26
Parametrierbetrieb\ 27
Handbetrieb\ 28
Einrichtbetrieb\ 29
Referenzfahrt manuell\ 30
Test Eingaenge/Ausgaenge\ 31
Korrektur Mass-System\ 32
Funktionstaste F12\ 33
COMTAC-Initialisierung\ 34
CPX-Parameter-Manager\ 35
Programm-Ende\ 36
Anlagen-Info\ 37
Maschinen-Parameter\ 38
Maschinen-Positionen\ 39
Achs-Parameter\ 40
Parametrierbetrieb beenden?\ 41
Bitte Wert eingeben !\ 42
X Y Z C\ 43
Achse: ACHSE X ACHSE Y ACHSE Z ACHSE C\44

Korrekturwert =\ 45
DIR LIST RUN PSTEP\ 46
Achsnummer eingeben: \ 47
Passwort eingeben: \ 48
Zugriff auf Funktion nicht erlaubt!\ 49
Antriebe Hand+/-\ 50
Achse->1,2,.. Verf->Cursor Geschw->. ESC\51
Achse->1,2,.. Para->LT/RT ändern->ENT ESC\52
Achse->1,2,.. ändern->ENT Ende->ESC\53
LANGSAM SCHNELL CPXPARA \ 54
Achse->1,2,.. Starten->START Abbruch->ESC\55
Weiter->ENT\ 56
Daten abspeichern ?\ 57
Test Eingaenge/Ausgaenge Ende->ESC\58
Achse->0,1,.. Ausg.->LT/RT 1/0->INS/DEL\59
Weiter->STEUERUNG EIN\ 60
Initialisierung...\ 61
Abbruch nach Stop\ 62
S T O P\ 63
N O T A U S\ 64
Warte auf Position erreicht...\ 65
Realnull anfahren\ 66
Weiter->START Abbruch->STOP\ 67
Weiter->ENT Abbruch->ESC\ 68
Ja->ENT Nein->ESC\ 69
Unterbrechen->STOP\ 70
Fehler Zeile Achse \ 71
COMTAC- COMPAX- ABLAUF-FEHLER \ 72
Fehler Feldbus-Kommunikation!\ 73
Endgrenzen sind gebrueckt!\ 74
Referenzfahrt nicht erfolgt!\ 75
Maschinen-Null anfahren !\ 76
Anlage Ausschalten !\ 77
Warten auf Eingang \ 78
Druckluft fehlt !\ 79
Start zum Anfahren des Maschinen-Null !\ 80
Fehler: Eingang 3 nicht ok !\ 81
Prog :\ 82
Eingabe Prog-Nr.:\ 83
Prog.Nr: Achse: Pos.Nr.: \84
\ 85
\ 86
\ 87
\ 88
\ 89
\ 90
\ 91
\ 92
\ 93
\ 94
\ 95
\ 96
\ 97
Projekt : \ 98
DUMMY-ANLAGE \ 99
AB-Nr. : 96/xxxx \100
Hersteller: Fa. Parker,Hauser Division \101
D-77656 Offenburg \ 102
Baujahr : 1996 \103

Inbetriebnahme: Dez. 1996 \104
Kunde : Firma XYZ aus ABC \105
Endkunde: Firma DEF aus GHI \106
Autor : AS, Hauser Softwareteam \107
Programm: CTC_PRO.BAS\108
Steuerung: COMTAC 2000 \109
\ 110
Achse : CPX \111
Geraete-Nr.: \112

7. Hauptprogramm

Sie erhalten eine komplette Programmstruktur mit allen wichtigen Funktionen einer Anlage:



Die COMTAC-Software für die Automation "CTC_Pro" beinhaltet dabei:

- ◆ Die Kommunikation mit COMPAX über den COMTAC - Feldbus:
 - Abfragen von Eingängen.
 - Definieren von Ausgängen.
 - Positionieren einzelner oder mehrerer Achsen.
 - Maschinennull suchen.
 - Handbetrieb der COMPAX - Achsen
- ◆ Eingabe der Maschinenpositionen.
- ◆ Eingabe der Maschinenparameter.
- ◆ Eingabe der Achsparameter (Geschwindigkeiten und Beschleunigungen für die Betriebsarten).
- ◆ Verwaltung der Maschinentexte.

Verlassen des Programms

Die COMTAC-Software für die Automation "CTC_Pro" wird mit [F15] (Programm-Ende) oder der Tastenkombination [STRG c] beendet.

Einlegestreifen für COMTAC

F5 Automatik- betrieb	F6 Parametrier- betrieb	F7 Handbetrieb	F8 Einricht- betrieb
F9 MO Manuell	F10 Test Eingänge/ Ausgänge	F11 Korrektur Maßbezug	F12
F13 COMTAC- Initialisierung	F14 Parameter- Manager	F15 Programm- Ende	F16 Anlagen- Info

8. Die Betriebsarten

Das Steuerprogramm enthält verschiedene Betriebsarten. Außer dem Automatikbetrieb gibt es einen Handbetrieb, einen Parametrierbetrieb und für Servicezwecke dienliche weitere Betriebsarten.

Die einzelnen Betriebsarten werden nachfolgend beschrieben.

Die Umschaltung der Betriebsarten erfolgt an der COMTAC-Tastatur.

Passworte: Der Zugriff auf die einzelnen Betriebsarten ist durch Passworte geschützt.

Die einzelnen Funktionen sind, wo dies notwendig ist, durch Paßwortabfragen vor unerlaubtem Zugriff geschützt. Es wird zwischen 2 Paßwortebenen unterschieden:

Passwort-Ebene	Zielgruppe	Paßwort	Abgelegt in:
Ebene 0	Maschinen-Einrichter	302	CT.(137)
Ebene 1	Servicepersonal	302	CT.(138)
Ebene 2	Programmierer	302	CT.(139)

Nachfolgend die Betriebsarten und ihre zugehörigen Paßwortebenen:

Fx	Betriebsarten	Paßwortebene
F5	Automatikbetrieb	0
F6	Parametrierbetrieb	0/1/1
F7	Handbetrieb	0
F8	Einrichtbetrieb	1
F9	Maschinen-Null Manuell	1
F10	Test Eingänge /Ausgänge	1
F11	Korrektur Maßbezug	2
F12		
F13	COMTAC-Initialisierung	1
F14	Parameter-Manager	2
F15	Programm-Ende	2
F16	Anlagen-Info	-

Jede Betriebsart wird von einem Hauptmenü aufgerufen. Nach Beenden der angewählten Betriebsart wird immer ins Hauptmenü zurückgegangen.

8.1 Einschalten von COMTAC und COMPAX

Stufe 1: Hauptschalter einschalten (Zentrale Versorgung)

Es gibt zwei Stufen zum Einschalten der Anlage:

➡ Spannungsversorgung des COMTAC wird geschaltet.

➡ COMTAC wird gebootet und Steuerprogramm wird automatisch gestartet.

Voraussetzung: die COMTAC-Software für die Automation "CTC_Pro" muß umbenannt sein: CTC_PRO.BAS in CTC_PRO.ASP (*.ASP definiert das Programm als Autostartprogramm) und die Autostartfunktion muß mit CT.(13) Bit 8 = "1" eingeschaltet sein.

➡ 24V der COMPAX-Achsen wird zugeschaltet.

➡ 24V für die digitalen Ein-/Ausgänge wird zugeschaltet.

Stufe 2: Taster 'Steuerung Ein' drücken

➡ Leistungsspannung (3*400V) der COMPAX-Achsen wird geschaltet.

➡ Die Anlage ist betriebsbereit.

Das weitere Vorgehen kann beispielsweise so aussehen:

Zum Einschalten der Anlage folgende Schritte nacheinander abarbeiten:

- ➡ Alle Türen am Schaltschrank schließen.
- ➡ Hauptschalter einschalten.
- ➡ Rechner bootet, Steuerprogramm wird gestartet.
- ➡ Warten, bis Meldung 'WEITER ➔ STEUERUNG EIN' am Bildschirm erscheint.
- ➡ Taste 'Steuerung Ein' drücken.
- ➡ nach ca. 3sec. ist die Anlage betriebsbereit.
- ➡ Zum Weiterarbeiten bitte nachfolgende Kapitel durcharbeiten.

8.1.1 Nullpositionen, Verfahrrichtungen der Achsen



Vorsicht:

Die Softwareendgrenzen aller Achsen werden erst überwacht, nachdem die Maschinennullpunkte angefahren worden sind. Als Sicherheit dienen die Hardware-Endschalter der Achsen (falls angeschlossen).

Nach dem Einschalten der Anlage müssen die Maschinennullpunkte der Achsen angefahren werden. Diese Punkte dienen als Bezugspunkt des gesamten Maßsystems.

Die Maschinennullpunkte müssen beim ersten Start des Automatikbetriebs angefahren werden. Dazu müssen Sie ab Zeile 20080 die einzelnen Achsen in der gewünschten Reihenfolge ansprechen (siehe ab Seite 42).

Die Achsen können im Handbetrieb verfahren werden, ohne zuvor die Maschinennullpunkte anzufahren.

Definition des Maschinennullpunktes:

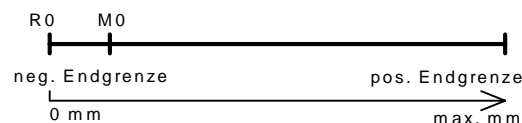
Der Maschinennullpunkt ist der Punkt, an dem der Maschinennullimpuls des Resolvers (1 pro Motorumdrehung) und das Signal des Maschinennullinitiators gemeinsam aktiv sind. Die Verundung dieser beiden Signale wird zum Definieren des Maschinennullpunktes herangezogen.

Finden des Maschinennullpunktes

a.) Die Maschinennullinitiatoren sind frei: Die Achsen fahren in Richtung der Initiatoren, bis diese bedämpft sind. Das Initiatorsignal und der Maschinennullimpuls werden verundet.

b.) Die Maschinennullinitiatoren sind bedämpft: Die Achsen fahren entgegengesetzt der Initiatoren, bis diese frei sind. Danach wird die Verfahrrichtung umgedreht. Die Achsen fahren in Richtung der Initiatoren, bis diese bedämpft sind. Das Initiatorsignal und der Maschinennullimpuls werden verundet.

Realnullpunkte:



Das Maßsystem kann vom Maschinennullpunkt aus verschoben werden (durch definieren eines Realnullpunkts - siehe im Produkthandbuch). Der Realnullpunkt hat das Maß 0mm. Alle Bezugsmaße beziehen sich auf den Realnullpunkt. Die Softwareendgrenzen beziehen sich weiter auf die Maschinennullpunkte.

8.1.2 Ausschalten der Anlage

Bedienung:

Zum Ausschalten der Anlage folgende Schritte nacheinander abarbeiten:

- ➡ Alle Betriebsarten beenden, am Bildschirm erscheint die Meldung 'Anwahl der Betriebsarten mit FKT.-Tasten'.
- ➡ Notaus drücken (wichtig, da sonst die vertikale Achse einige Zentimeter absinkt).
- ➡ Nach ca. 5 sec. Hauptschalter ausschalten.

8.2 [F6] Parametrierbetrieb

- ◆ Unter dem Menüpunkt " Parametrierbetrieb" geben Sie sämtliche Parameter Ihrer Anlage ein. Auf diese Parameter können Sie dann im Automatik-Programm zugreifen.
- ◆ Nach Anwahl der Funktion "Parametrierbetrieb" mit der Taste **[F6]** und Eingabe des entsprechenden Passworts wird der Parametrierbetrieb geöffnet:

Maschinen-Name Parametrierbetrieb

Maschinenparameter?

Ja-> ENT
Nein -> ESC

Im Parametrierbetrieb können Maschinenparameter, Maschinenpositionen und Achsparameter verändert werden. Normalerweise werden diese Parameter einmal bei der Inbetriebnahme der Anlage programmiert. Sie müssen dann nur noch in Ausnahmefällen verändert werden.

Der Parametrierbetrieb bietet nachfolgende Menüpunkte zur Auswahl, die mit [ENT] selektiert werden können:

- ◆ Maschinenparameter [Passwortebene 0].
- ◆ Maschinenpositionen [Passwortebene 1].
- ◆ Achsparameter [Passwortebene 1].
- ◆ Parametrierbetrieb beenden.

Maschinenparameter ändern

Nach Anwahl des Menüpunkts Maschinenparameter erscheint folgendes Menü zur Bearbeitung der Parameter:

Maschinenparameter sind achsunabhängig

Maschinen-Name Maschinen-Parameter

Maschinenparam. 1 [UG=0/Og=100]:

Achse->1,2,..
Para LT,RT
ändern->ENT
ESC

Dieser Name kommt aus der Datei PART1S1(2).DAT

Beschreibung der Steuertasten:

Achse->1,2,..:	Keine Funktion.
Para -> LT(🔍)/RT(🔍):	Mit Taste [LT(🔍)] bzw. [RT(🔍)] wird der vorherige, bzw. nächste Parameter angezeigt. Nach Erreichen des letzten Parameters wird wieder der erste Parameter angezeigt und umgekehrt.
ändern ->ENT:	Nach Drücken von [ENT] kann der selektierte Parameter verändert bzw. überschrieben werden. Die Eingabe wird dann mit [ENT] abgeschlossen. Wird bei der Eingabe ein Wert nicht akzeptiert, so lag der eingegebene Wert nicht innerhalb der zulässigen Grenzen.
ESC:	Die Funktion wird beendet, mit Abfrage ob die Daten gespeichert oder die ursprünglichen Daten beibehalten werden sollen. Danach kann man den Parametrierbetrieb beenden oder einen anderen Menüpunkt zur Bearbeitung selektieren.

Beschreibung der Maschinenparameter:

Parameter-Nummer	Wert	Beschreibung
Parameter 1		
Parameter 2		
Parameter 3		
Parameter 4		
Parameter 5		
Parameter 6		
Parameter 7		
Parameter 8		
Eingangs-Überwachungszeit (feste Belegung)	6s	
Bahngeschwindigkeit (feste Belegung)	70 %	Resultierende Geschwindigkeit aus den Komponentengeschwindigkeiten der 2 Achsen, die gemeinsam positioniert werden
Bahnrampenzeit (feste Belegung)	1200 ms	Resultierende Rampenzeit aus den Komponentenrampenzeiten der 2 Achsen, die gemeinsam positioniert werden
Parameter 12		
...		
Parameter [CT(134)]		max. 44

Die Bahnfunktion ist eine Linearinterpolation über 2 Achsen mit versetztem Start (um 1 Befehl)

Maschinenpositionen bearbeiten

Nach Anwahl des Menüpunkts Maschinenpositionen erscheint folgendes Menü zur Bearbeitung der Positionen. Zuvor muß die Nummer des Bearbeitungsprogrammes eingegeben werden, in welches die Positionen gespeichert werden sollen.

Maschinen-Name Masch-Pos. 1:				
Achse: X	Achse X	Achse Y	Achse Z	Achse C
	1000.00	2000.00	3000.00	4000.00
Achse->1,2,..	Para->LT/RT	ändern->ENT	ESC	

Dieser Name kommt aus der Datei PART1S1(2).DAT

➡ Die eingegebenen Positionen werden im Automatik-Programm mit den Positionsnummer angesprochen.

Beachte:

Die Positionen können geteacht werden. Vor Benutzen der Steuertaste [INS] zum Übernehmen von Positionen aus dem Zwischenspeicher, ist dafür zu sorgen, daß zuvor mit der Funktion [F7] Handbetrieb die zum aktuellen Parameter zugehörigen Positionen im Zwischenspeicher gespeichert wurden!

Beschreibung der Steuertasten:

Achse->1,2,..:	Mit Taste [1], [2], ... wird die Achse selektiert, deren Position geändert werden soll. Die selektierte Achse wird in Zeile 2 am Display angezeigt.
Para->LT(☐)/RT(☐):	Mit [LT(☐)] bzw. [RT(☐)] wird die vorherige, bzw. nächste Position angezeigt. Nach Erreichen der letzten Position wird wieder der erste Position angezeigt und umgekehrt.
ändern->ENT:	Nach Drücken von [ENT] kann die selektierte Position verändert bzw. überschrieben werden. Die Eingabe wird dann mit [ENT] abgeschlossen. Wird bei der Eingabe ein Wert nicht akzeptiert, so lag der eingegebene Wert außerhalb der zulässigen Grenzen.
[ESC]:	Die Funktion wird beendet, mit Abfrage ob die Daten gespeichert oder die ursprünglichen Daten beibehalten werden sollen. Danach kann man den Parametrierbetrieb beenden oder einen anderen Menüpunkt zur Bearbeitung selektieren.
INS:	Mit [INS] werden evtl. zuvor unter [F7] Handbetrieb zwischengespeicherte Istpositionen der Achsen eingelesen und als Positionen des aktuellen Parameters übernommen. Dadurch erspart man sich das Notieren und Eintippen von Positionen.

Beschreibung der Maschinenpositionen:

Positions-Nummer:	Wert	Beschreibung
Masch-Position 1	X=____;Y=____	
Masch-Position 2	X=____;Y=____	
Masch-Position 3	X=____;Y=____	
Masch-Position 4	X=____;Y=____	
Masch-Position 5	X=____;Y=____	
Masch-Position 6	X=____;Y=____	
.....	X=____;Y=____	
Masch-Position [CT(132)]	X=____;Y=____	max. 31

Achsparemeter bearbeiten

Nach Anwahl des Menüpunkts Achsparemeter erscheint folgendes Menü zur Bearbeitung der Parameter :

Achsparemeter werden für jede Achse getrennt eingegeben:

Maschinen-Name Achs-Parameter

Achse X Auto-Geschw. schnell [%]: 90

Achse->1,2,.. Para->LT/RT ändern->ENT ESC

Dieser Name kommt aus der Datei PART1S1(2).DAT

Beschreibung der Steuertasten:

Achse->1,2,..:	Mit Taste [1], [2], .. wird die Achse selektiert, deren Parameter geändert werden soll. Die selektierte Achse wird in Zeile 2 am Display angezeigt.
Para->LT(☐)/RT(☐):	Mit [LT(☐)] bzw. [RT(☐)] wird in der vorherige, bzw. nächste Parameter angezeigt. Nach Erreichen des letzten Parameters wird wieder der erste Parameter angezeigt und umgekehrt.
ändern->ENT:	Nach Drücken von [ENT] kann der Wert des selektierten Parameters verändert bzw. überschrieben werden. Die Eingabe wird dann mit [ENT] abgeschlossen. Wird bei der Eingabe ein Wert nicht akzeptiert, so lag der eingegebene Wert nicht innerhalb der zulässigen Grenzen.
[ESC]:	Die Funktion wird beendet, mit Abfrage ob die Daten gespeichert oder die ursprünglichen Daten beibehalten werden sollen.

Folgende Parameter können beeinflusst werden:

Parameter-Nummer:	Wert	Beschreibung
Auto-Geschw. schnell [%]: (feste Belegung)	X=____ % Y=____ %	Schnelle Automatik-Geschwindigkeit der Anlage (in % der Maximalgeschwindigkeit der selektierten Achse)
Auto-Beschl. schnell [ms]: (feste Belegung)	X=____ ms Y=____ ms	Automatik-Beschleunigung der Anlage schneller Geschwindigkeit (Beschl.Zeit der selektierten Achse)
Auto-Geschw. langsam [%]: (feste Belegung)	X=____ % Y=____ %	Langsame Automatik-Geschwindigkeit der Anlage (in % der Maximalgeschwindigkeit der selektierten Achse)
Auto-Beschl. langsam [ms]: (feste Belegung)	X=____ ms Y=____ ms	Automatik-Beschleunigung bei langsamer Geschwindigkeit der Anlage (Beschl.Zeit der selektierten Achse)

Hand-Geschw. langsam [%]: (feste Belegung)	X=___ % Y=___ %	Geschwindigkeit der Anlage beim Verfahren mit der Funktion [F7] Handbetrieb LANGSAM (in % der Maximalgeschwindigkeit der selektierten Achse)
Hand-Geschw. schnell [%]: (feste Belegung)	X=___ % Y=___ %	Geschwindigkeit der Anlage beim Verfahren mit der Funktion [F7] Handbetrieb SCHNELL (in % der Maximalgeschwindigkeit der selektierten Achse)
Hand-Beschl. langsam [ms]: (feste Belegung)	X=___ ms Y=___ ms	Beschleunigung der Anlage beim Verfahren mit der Funktion [F7] Handbetrieb LANGSAM (Beschl.Zeit der selektierten Achse)
Hand-Beschl. schnell [ms]: (feste Belegung)	X=___ ms Y=___ ms	Beschleunigung der Anlage beim Verfahren mit der Funktion [F7] Handbetrieb SCHNELL (Beschl.Zeit der selektierten Achse)
Maximale Geschwindigkeit [%]: (feste Belegung)	X=___ ms Y=___ ms	Die maximale Geschwindigkeit einer Einzelachse, die bei einer gemeinsamen Bahn-Positionierung von 2 Achsen nicht überschritten werden darf.
Minimale Rampenzeit [ms]: (feste Belegung)	X=___ ms Y=___ ms	Die minimale Rampenzeit einer Einzelachse, die bei einer gemeinsamen Bahn-Positionierung von 2 Achsen nicht unterschritten werden darf.
Weitere Achsparameter können für eigene Programmierung hinzugefügt werden.		

8.3 [F7] Handbetrieb



Achtung:

< Passwortebeine 0 >

Im Handbetrieb sind alle Funktionen durch den Bediener manuell zu kontrollieren. Erfolgte vor dem manuellen Verfahren noch keine M0-Fahrt, so ist die Überwachung der Software-Endgrenzen nicht aktiv. Deshalb darauf achten, daß (falls vorhanden) die mech. Endgrenzen (Schlüsselschalter Endbegrenzung überbrücken) aktiv sind.

Achsen manuell verfahren

Nach Anwahl der Funktion **Handbetrieb** mit der Taste **[F7]** und Eingabe des entsprechenden Paßwortes erscheint folgendes Menü am Display:

Maschinen-Name Antriebe Hand +/-			
Achse X	LANGSAM	V=xxx %	B=yyyy ms
A01: Istposition = nnnn			
Achse->1,2,..	Verf->Cursor	Geschw.->	ESC

Mit dieser Funktion können die Achsen manuell verfahren werden. Dabei werden die Istposition, die programmierte Geschwindigkeit (V=xx%) und die programmierte Beschleunigung (B=yyyy ms) am Display angezeigt.

Manuelles Verfahren der Achsen kann in folgenden Fällen erforderlich sein:

- ◆ Nach Anfahren der mechanischen Endgrenzen.
- ◆ Für Servicearbeiten innerhalb des Linienportals.
- ◆ Zum Ermitteln von Maschinenpositionen, die dann in der Funktion [F6] Parametrieren einzugeben sind.

**Beschreibung der
Steuertasten:**

Achse->1,2,..:	Mit Taste [1], [2], ... wird die Achse selektiert und in Zeile 2 am Display angezeigt.
Verf->Cursor:	Mit Taste [LT()], [RT()] für horizontale Achsen und [UP()], [DN()] für vertikale Achsen verfährt die selektierte Achse in ihre negative bzw. positive Richtung. Sie stoppt, wenn die gedrückte Taste losgelassen oder eine Software-Endgrenze der Achse erreicht wird.
Geschw->.::	Mit der Taste [.] kann zwischen einer langsamen und einer schnellen Handgeschwindigkeit umgeschaltet werden. Die aktuelle Einstellung wird am Display in Zeile 2 (LANGSAM/SCHNELL) angezeigt.
[ESC]:	Funktion wird beendet
INS:	Mit der Taste [INS] werden die Istpositionen aller Achsen eingelesen und in einem Zwischenspeicher abgelegt. Dieser Zwischenspeicher kann im Parametrierbetrieb [F6] dann als Maschinenposition übernommen werden. Dadurch erspart man sich das Notieren und Eintippen von Positionen. Der Inhalt dieses Zwischenspeichers geht nach Ausschalten der Anlage verloren.

8.4 [F9] M0 manuell

<Passwortebene 1>

Beachte:

- ◆ Als erstes die Referenzfahrt der vertikalen Achse durchführen, da eine Referenzfahrt der horizontalen Achse eine Kollision herbeiführen könnte.
- ◆ Mit [STOP] kann eine gestartete Referenzfahrt abgebrochen werden.
- ◆ Nach Finden des Referenzpunkts (Nachfolgender Nullimpuls des Resolvers nach Erreichen des Maschinennull-Initiators) erfolgt automatisch noch eine Positionierung zum Realnullpunkt der selektierten Achse.

**Referenzfahrt
durchzuführen**

Nach Anwahl der Funktion **M0 Manuell** mit der Taste [F9] und Eingabe des entsprechenden Paßwortes erscheint folgendes Menü am Display:

Maschinen-Name Referenzfahrt manuell

Achse X

Achse->1,2,.. Starten->START Abbruch->ESC

Immer nach dem Einschalten der Anlage ist einmalig eine Referenzfahrt mit jeder Achse durchzuführen, danach sind die programmierten Software-Endgrenzen der Positionierregler gültig und werden vom System überwacht.

Eine Referenzfahrt aller Achsen erfolgt normalerweise automatisch beim ersten Starten des Automatikbetriebs. Man hat mit dieser Funktion jedoch die Möglichkeit eine Referenzfahrt manuell durchzuführen (z.B. vor erstmaligem Anwenden der Funktion '[F7] Handbetrieb').

**Beschreibung der
Steuertasten:**

Achse->1,2,..:	Mit Taste [1], [2], .. wird die Achse selektiert, die eine Referenzfahrt machen soll. Die selektierte Achse wird in Zeile 2 am Display angezeigt
--------------------------	--

Starten->START:	Mit [START] wird die Referenzfahrt der selektierten Achse gestartet.
Stoppen->[STOP]:	Mit [STOP] wird die Referenzfahrt der selektierten Achse gestoppt.
Abbruch->[ESC]:	Funktion wird beendet

8.5 [F10] Test E/A's

<Passwortebene 1>

Beachte:

Die Funktion Test Eingänge/Ausgänge wurde für Servicezwecke integriert. Deshalb ist die Durchführung nur durch qualifiziertes und geschultes Personal erlaubt. Eine Falschbedienung kann zu Schäden führen !

Eingänge und Ausgänge anzeigen

Nach Anwahl der Funktion **Test E/A's** mit der Taste **[F10]** und Eingabe des entsprechenden Paßwortes wird ein Menü zum Ansehen der Eingänge bzw. Setzen/Rücksetzen der Ausgänge des COMTAC geöffnet :

Anzeige Eing./Ausg.		Ende -> ESC
Comtac E16-1=0000'0000'0000'0000B		
Ausgang: Ausgang Contac 1		
Achse->0,1,...	Ausg.->LT/RT	1/0->INS/DEL

Dieser Name kommt aus der Datei OUTT1S1(2).DAT

In der Display-Zeile 2 wird der Zustand der 16/32 Eingänge des COMTAC und die 16 Eingänge der COMPAX Achsen angezeigt. In der Zeile 3 wird der Ausgang angezeigt, der gesetzt bzw. rückgesetzt werden kann.

Beschreibung der Steuertasten:

Achse->0,1,...:	Mit der Taste [0] wird das COMTAC und mit [1], [2], .. wird die Achse angewählt, deren 16 Eingangszustände angezeigt werden sollen.
COMTAC 3000	mit der Taste [.] wird bei COMTAC 3000 zwischen den Eingängen 1-16 und 17-32 gewechselt
Ausg.->LT(☐)/RT(☐):	Mit der Taste [LT(☐)] bzw. [RT(☐)] wird der Ausgang angezeigt, der gesetzt bzw. rückgesetzt werden soll.
1/0->INS/DEL:	Mit der Taste [INS] wird der angewählte Ausgang gesetzt, mit [DEL] wird der angewählte Ausgang rückgesetzt.
Ende->[ESC]:	Die Funktion wird beendet. Alle Ausgänge werden zurückgesetzt.

Beachte:

Um alle Ausgänge der COMPAX-Achsen anzusprechen, ist eine Maskierung notwendig. Soll das Setzen/Rücksetzen der Standard-Ausgänge 1-6 verhindert werden, so muß die Maskierung aufgehoben werden. (Siehe auch Softwarebeschreibung).

8.6 [F11] Korrektur Maßbezug



Vorsicht:

<Passwortebene 2>

Diese Funktion beeinflusst das Maßsystem der Anlage. Deshalb ist sie durch ein spezielles Paßwort geschützt.



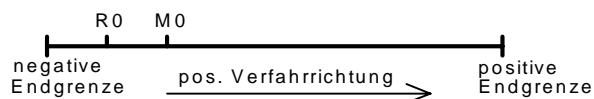
KOLLISIONSGEFAHR.

Maßsystem korrigieren

Nach Anwahl der Funktion **Korrektur Maßsystem** mit der Taste **[F11]** und Eingabe des entsprechenden Paßwortes wird ein Menü zur Korrektur des Maß-Systems geöffnet:

Falls nach der Referenzfahrt der Anlage die Referenzposition(en) einer (der) Achse(n) nicht mehr stimmt (z.B. wegen Zahnriemensprung durch Kollision, Maschinennull-Initiator verschoben, usw.), hat man mit dieser Funktion die Möglichkeit, das Maßsystem zu korrigieren.

Das Maßsystem jeder Achse der Anlage setzt sich zusammen aus Maschinennullpunkt (Resolver-Nullimpuls und Maschinennull-Initiator), Realnullpunkt und pos. und neg. Endgrenze, gemäß nachfolgender Skizze:



Der Realnullpunkt (R0) sowie die pos. und neg. Endgrenze beziehen sich auf den Maschinennullpunkt (M0). Daher würde durch eine Verschiebung des Maschinennullpunkts das gesamte Maßsystem verschoben, d.h. das Anfahren von Positionen würde eine Kollision zur Folge haben.

Für diesen Fall kann mit der Funktion [F11] das Maßsystem vom Maschinennullpunkt aus verschoben werden, indem man die Differenz zwischen aktueller und originaler Referenzposition jeder Achse als Korrekturwert eingibt.

Dabei gilt: Das Vorzeichen des Korrekturwerts entspricht der Verfahrrichtung (pos/neg), in welche die Referenzposition der zugehörigen Achse verschoben wurde.

Beschreibung der Steuertasten:

Para->1,2,...:	Mit Taste [1], [2], ... wird Korrekturwert der vorherigen, bzw. nächsten Achse angezeigt. Nach Erreichen des letzten Korrekturwerts wird wieder der Korrekturwert der ersten Achse angezeigt und umgekehrt.
ändern- >ENT:	Nach Drücken von [ENT] kann der selektierte Korrekturwert verändert bzw. überschrieben werden. Die Eingabe wird dann mit [ENT] abgeschlossen. Wird bei der Eingabe ein Wert nicht akzeptiert, so lag der eingegebene Wert nicht innerhalb der zulässigen Grenzen.
[ESC]:	Die Funktion wird beendet, mit Abfrage ob die Daten gespeichert oder die ursprünglichen Daten beibehalten werden sollen. Quittiert man mit [ENT], so korrigiert das Ablaufprogramm die Maßsysteme der COMPAX-Regler entsprechend den eingegebenen Korrekturwerten.

Nach der Korrektur sollte man eine manuelle Referenzfahrt durchführen, um die korrigierten Referenzpositionen zu überprüfen.

8.7 [F14] Parameter-Manager**< Passwortebeene 2>**

Das Einstellen und Ändern von COMPAX - Parameter sollte falls möglich über PC mit dem Servo-Manager vorgenommen werden.

**Vorsicht:**

Der integrierte Parametermanager erlaubt die Initialisierung von COMPAX-Geräten ohne Überwachung auf Gültigkeit, d.h. es liegt in der Verantwortung des Anwenders, die richtigen Parameter ins richtige COMPAX zu schreiben.



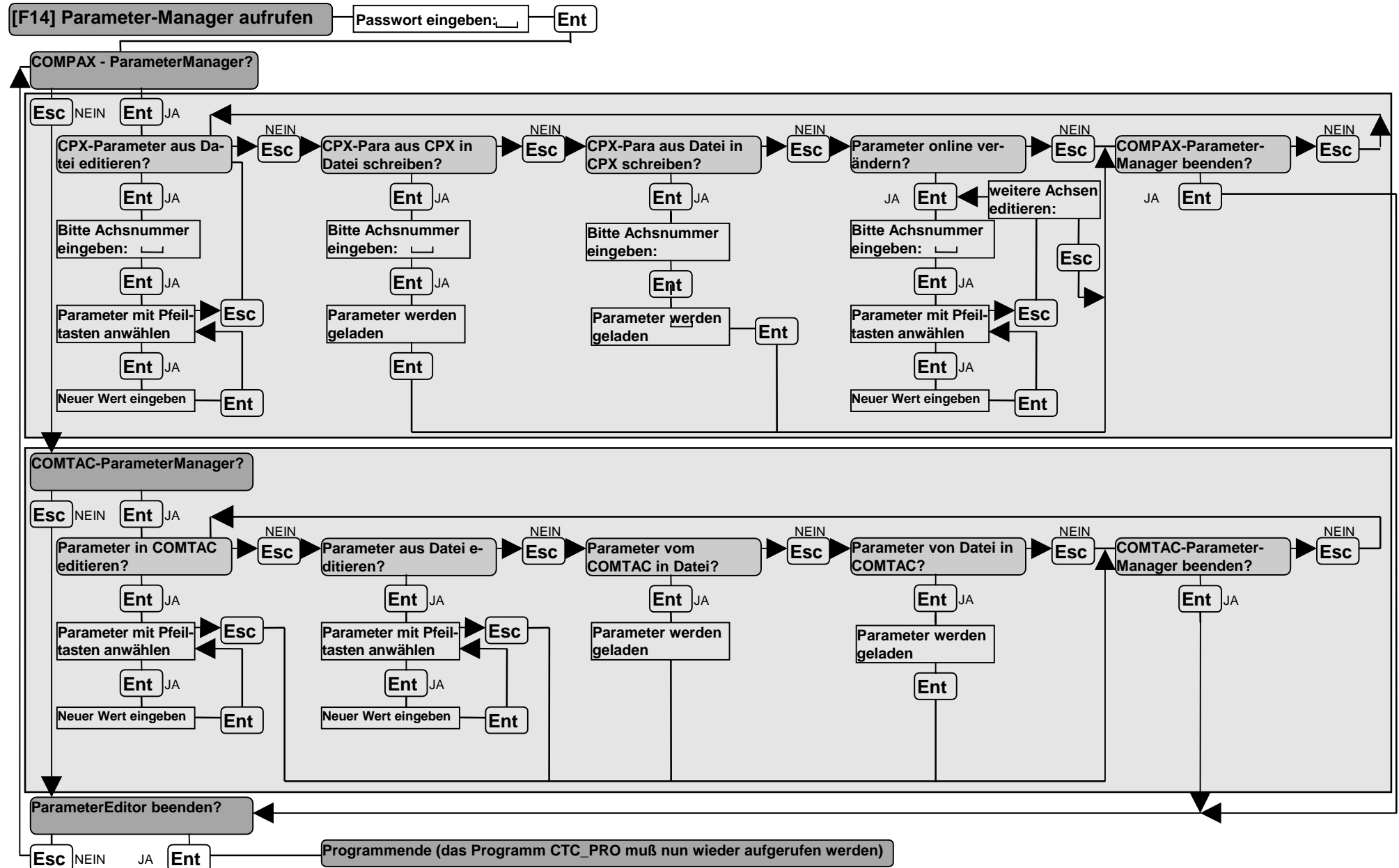
Unbefugtes oder inkorrektes Benutzen der Funktion kann zu Fehlfunktionen der Anlage führen !

Mit dem integrierten Parametermanager kann man die Parameter der COMPAX-Geräte sichern, ansehen und komplett in ein eventuell ausgetauschtes COMPAX schreiben.

Nach Anwahl der Funktion **Parameter-Manager** mit der Taste **[F14]** und Eingabe des entsprechenden Passworts wird der Parameter-Manager geöffnet:

The screenshot shows a menu box with a grey border. Inside, there is a title bar that says "Maschinen-Name CPX-Parameter-Manager". Below the title bar is a blank input field. Underneath that is a menu item "CPX-Parameter aus Datei anzeigen?". At the bottom of the menu is a line with two options: "Ja-> ENT" and "Nein -> ESC".

Menüstruktur des Parameter-Managers



Parameter mit Pfeiltasten auswählen:

LT(←)	Nächster Parameter auswählen
RT(→)	Vorhergehender Parameter auswählen
UP(↑)	Parameter in 10er - Schritten steigend auswählen
[DN(↓)]	Parameter in 10er - Schritten fallend auswählen

8.8 [F13] CTC-Initialisierung



Vorsicht:

<Passwortebene 1>

Mit dieser Funktion wird das COMTAC neu initialisiert.

➡ Unbefugtes oder inkorrektes Benutzen der Funktion kann zu Fehlfunktionen der Anlage führen!

Mit dieser Funktion kann, falls erforderlich, eine Neuinitialisierung der Steuerung COMTAC 2000/3000 vorgenommen werden, d.h. es werden spezielle Konfigurationsparameter (z.B. Schnittstellen, Display, Tastatur, usw.) mit voreingestellten Werten beschrieben.

Nach Anwahl der Funktion **COMTAC-Initialisierung** mit der Taste **[F13]** und Eingabe des entsprechenden Passworts wird die Initialisierung durchgeführt.

8.9 [F16] Anlagen-Info

Nach Anwahl der Funktion **Anlagen-Info** mit der Taste **[F16]** kann der Bediener durch mehrmaliges Drücken der Taste **[ENT]** allgemeine Anlagen-Informationen wie Hersteller, Baujahr, Software-Version des Ablaufprogramms, usw. einsehen. Es ist weiterhin sinnvoll, sich diese Anlagen-Info zu notieren und sie im Fehlerfall uns mitzuteilen.

Weiter->[ENT]	Anzeige der nächsten Information, nach der letzten anzuzeigenden Information wird die Funktion Anlagen-Info beendet.
Abbruch->ESC	Abbruch der Funktion Anlagen-Info

8.10 [F15] Programm-Ende

<Passwortebene 2>

Das Programm CTC_PRO.BAS wird abgebrochen und die Maskierung der E/As wird aufgehoben.

8.11 [F8] Einrichtbetrieb

<Passwortebene 1>

Hier wird ein Programm (Modul DR ab Zeile 47000) zum Einrichten der Anlage aufgerufen

Das Modul muß von Ihnen anlagenspezifisch programmiert werden.

9. SPS-Programmierer

SPS-Programmierer mit COMTAC - COMPAX - Verbund.

- Es steht ein lauffähiges COMTAC-Ablaufprogramm zur Verfügung, das über eine digitale E/A-Schnittstelle mit einer SPS kommuniziert.
- Über Menüs werden Maschinenparameter, Positionen und Achsparameter programmiert (oder geteacht) und in Dateien gespeichert.
Über binär codierte Eingänge werden Programmnummer, Positionsnummer und Achsnummer von der SPS eingestellt. Das COMTAC übernimmt die eigentliche Positionieraufgabe und Achsüberwachung und meldet den Status an die SPS zurück.
Werden zusätzliche E/A's zur Steuerung von Greifern, Sensoren, Zylindern, .. benötigt, so werden diese direkt von der SPS gesteuert.
- Betriebszustände, Fehler, etc. der SPS können über zusätzliche, binär codierte Eingänge am COMTAC angezeigt werden, was eine teure Anzeigeeinheit erspart.



Es ist keine COMTAC - Programmierung notwendig!



Die Prozeß- und Ablaufsteuerung befindet sich in der SPS.

Die Funktionsweise der E/A - Kommunikation zwischen SPS und COMTAC ist in Signaldiagrammen beschrieben.

9.1 Eingangsliste

E	Sender	Empfänger	Beschreibung mit Signalpegel
E1	SPS	Comtac 2000/3000	1 Netz Ein
E1			0 Notaus
E2	SPS	Comtac 2000/3000	1 Stop
E2			0
E3	SPS	Comtac 2000/3000	1 Umschalten in Automatik
E3			0
E4	SPS	Comtac 2000/3000	1 Start Automatikpositionierung
E4			0
E5	SPS	Comtac 2000/3000	1 Handfahren von der SPS aus
E5			0
E6	SPS	Comtac 2000/3000	1 M0 anfahren von der SPS aus
E6			0
E7	SPS	Comtac 2000/3000	1 Bearbeitungsprogramm einlesen
E7			0
E8	SPS	Comtac 2000/3000	1 Bearb.-Programmnr. Bit 0 / Achsnummer Bit 0 / Automatikgeschwindigkeit schnell / Handgeschwindigkeit schnell
E8			0 Automatikgeschwindigkeit langsam / Handgeschwindigkeit langsam
E9	SPS	Comtac 2000/3000	1 Bearb.-Programmnr. Bit 1 / Achsnummer Bit 1 / Start einer Bahnpositionierung mit 2 Achsen
E9			0
E10	SPS	Comtac 2000/3000	1 Bearb.-Programmnr. Bit 2 / Achsnummer Bit 2
E10			0
E11	SPS	Comtac 2000/3000	1 Bearb.-Programmnr. Bit 3 / Positionsnummer Bit 0
E11			0
E12	SPS	Comtac 2000/3000	1 Bearb.-Programmnr. Bit 4 / Positionsnummer Bit 1
E12			0
E13	SPS	Comtac 2000/3000	1 Bearb.-Programmnr. Bit 5 / Positionsnummer Bit 2
E13			0
E14	SPS	Comtac 2000/3000	1 Bearb.-Programmnr. Bit 6 / Positionsnummer Bit 3
E14			0
E15	SPS	Comtac 2000/3000	1 Positionsnummer Bit 4
E15			0
E16	SPS	Comtac 2000/3000	1 Positionsnummer / Achsnummer übernehmen / Teachen im Handbetrieb

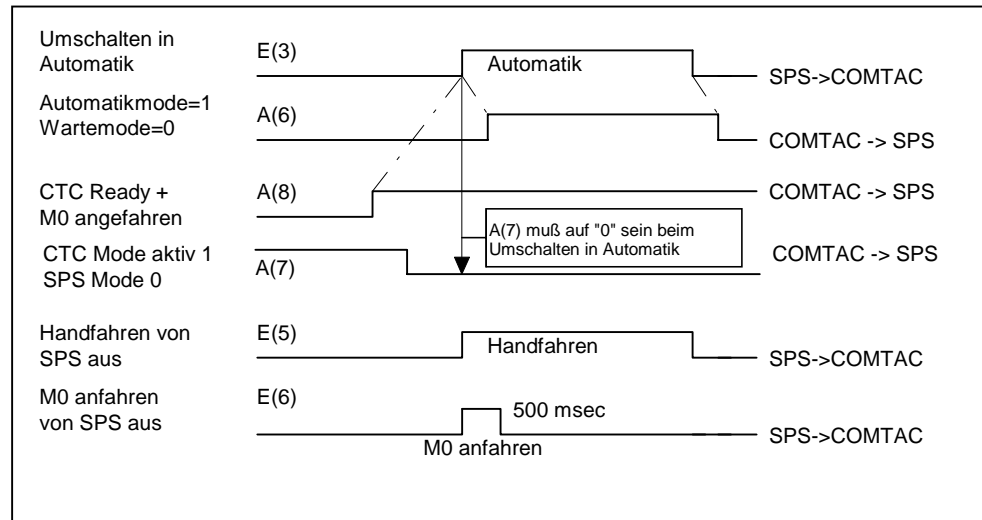
Über die Eingänge E7 ... E13 des 1. COMPAX können Sie die 100 Texte der Datei SPSTEXT.DAT in der 3. Zeile zur Anzeige bringen.

E	Sender	Empfänger	Beschreibung mit Signalpegel
E16			0
...			
E33			1
E33			0
E34			1
E34			0
E35			1
E35			0
E36			1
E36			0
E37			1
E37			0
E38			1
E38			0
E39	SPS	Comtac 2000/3000	1 SPS-Text Bit 0
E39			0
E40	SPS	Comtac 2000/3000	1 SPS-Text Bit 1
E40			0
E41	SPS	Comtac 2000/3000	1 SPS-Text Bit 2
E41			0
E42	SPS	Comtac 2000/3000	1 SPS-Text Bit 3
E42			0
E43	SPS	Comtac 2000/3000	1 SPS-Text Bit 4
E43			0
E44	SPS	Comtac 2000/3000	1 SPS-Text Bit 5
E44			0
E45	SPS	Comtac 2000/3000	1 SPS-Text Bit 6
E45			0
E46			1
E46			0
E47			1
E47			0
E48			1
E48			0

9.2 Ausgangsliste

A	Sender	Empfänger	Beschreibung mit Signalpegel
A1	Comtac 2000/3000	SPS	1 M0 angefahren pro Achse
A1			0
A2	Comtac 2000/3000	SPS	1 Automatikposition erreicht
A2			0
A3	Comtac 2000/3000	SPS	1 Fehler/Error
A3			0
A4	Comtac 2000/3000	SPS	1 Bearbeitungsprogrammnummer OK / Teachposition eingelesen
A4			0
A5	Comtac 2000/3000	SPS	1 Achsnummer und Positionsnummer OK
A5			0
A6	Comtac 2000/3000	SPS	1 Automatik-Mode
A6			0 Warte-Mode
A7	Comtac 2000/3000	SPS	1 Comtac-Mode (Menüs im COMTAC sind ausgewählt und aktiv)
A7			0
A8	Comtac 2000/3000	SPS	1 Comtac ok und M0 aller Achsen angefahren
A8			0
A9			1
A9			0
			1
...			0

9.3 Betriebarten umschalten



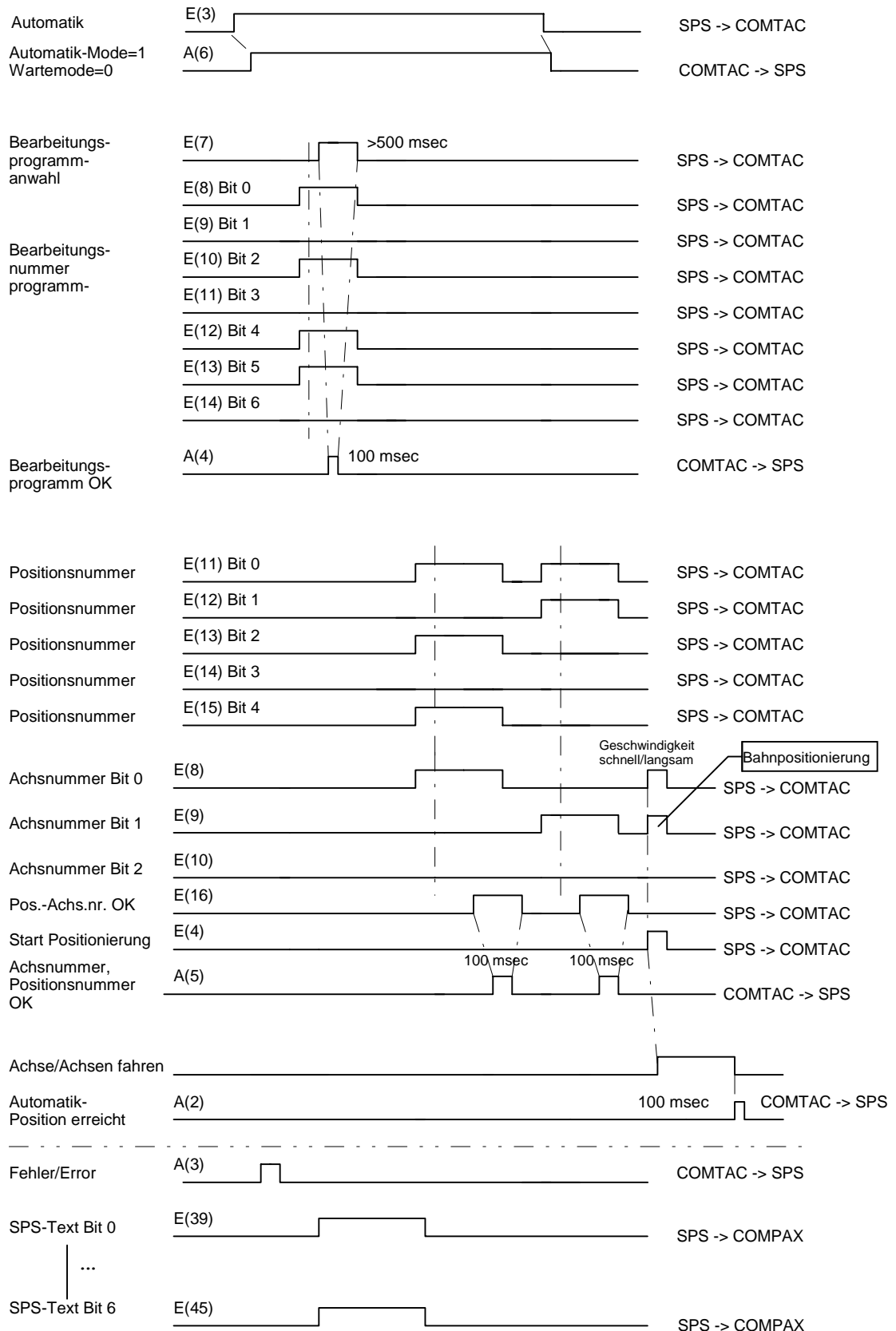
Zum Umschalten in den Automatikbetrieb muß die COMTAC-Software für die Automation "CTC_Pro" im Hauptmenü stehen, COMTAC muß bereit sein und der Maschinennull der Achsen muß angefahren worden sein..

OUT(7) = "1" zeigt an, daß sich COMTAC-Software für die Automation "CTC_Pro" in einer Betriebsart befindet.

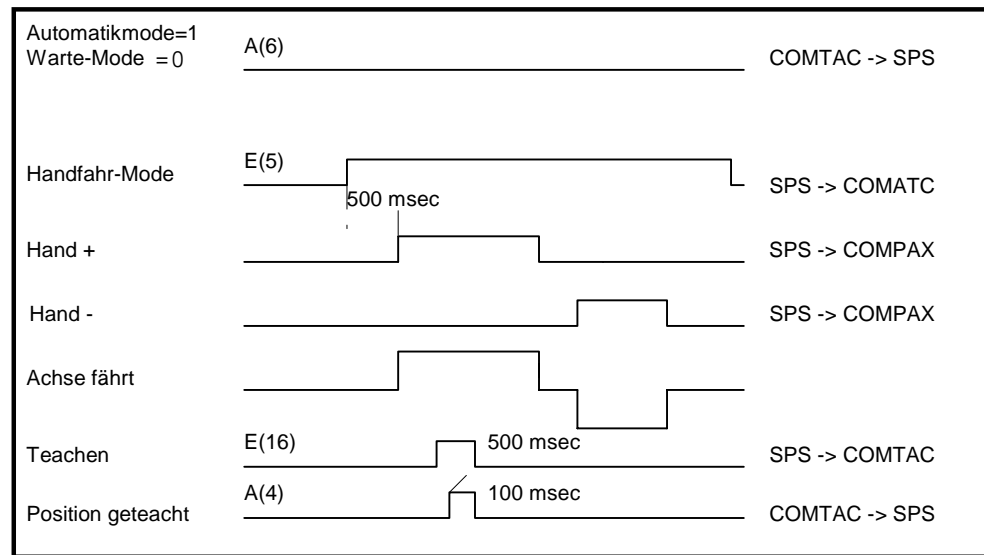
OUT(8) = "1" zeigt an, daß der Maschinennull der Achsen angefahren worden ist.

9.4 COMTAC als Achssteuerung

9.4.1 Automatik

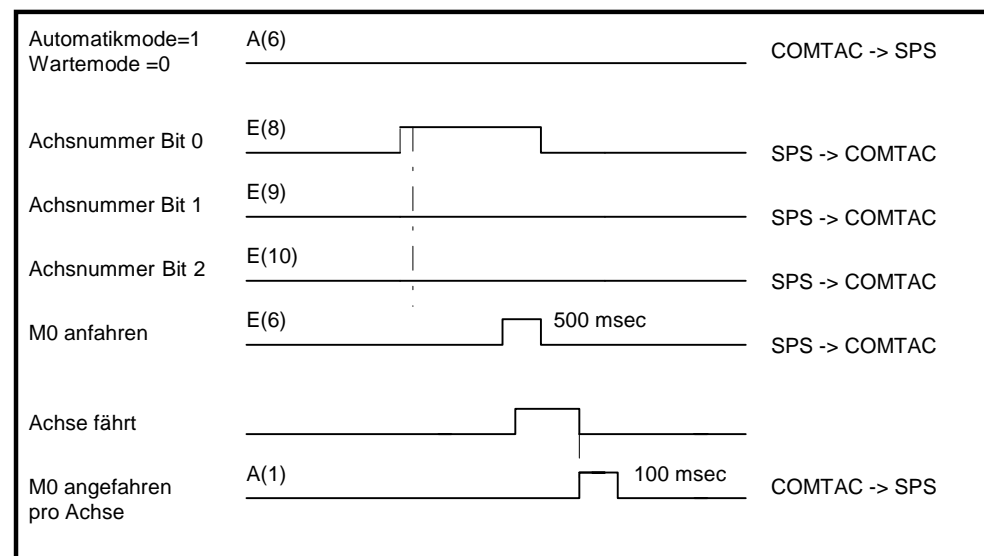


9.4.2 Hand



Hand+ und Hand- wird mit den entsprechenden COMPAX - Eingänge verdrahtet.

9.4.3 Maschinennull suchen



9.5 COMTAC als E/A - Textanzeigeeinheit

Betriebsart "COMTAC als Textanzeigeein- heit" einstellen:

CT.(136) = 2

Diese Einstellung kann:

♦ direkt durch Ändern des Parameters

oder

♦ über das Programm "CTCTEACH.BAS", welches beim erstmaligen Starten der COMTAC-Software für die Automation "CTC_Pro" automatisch abläuft.

Funktion:

Aus 100 Texten der Textdatei SPSTEXT.DAT können Sie über COMTAC - Eingänge Texte auswählen und in der angewählten Zeile anzeigen lassen.

♦ Textauswahl über E1 bis E7

♦ Zeilenauswahl: E8: 2. Zeile anwählen

E9: 3. Zeile anwählen

E10: 4. Zeile anwählen

Die 1. Zeile ist mit dem Text 99 aus der Textdatei CTCT1S1(2) belegt.

Textauswahl über die COMTAC - Ein- gänge

Eingangsbelegung							Ausgewählter Text (max. 40 Zeichen)
1	2	3	4	5	6	7	
0	0	0	0	0	0	0	keine Funktion
0	0	0	0	0	0	1	SPS - Text 1
0	0	0	0	0	1	0	SPS - Text 2
0	0	0	0	0	1	1	SPS - Text 3
0	0	0	0	1	0	0	SPS - Text 4
0	0	0	0	1	0	1	SPS - Text 5
0	0	0	0	1	1	0	SPS - Text 6
0	0	0	0	1	1	1	SPS - Text 7
0	0	0	1	0	0	0	SPS - Text 8
0	0	0	1	0	0	1	SPS - Text 9
0	0	0	1	0	1	0	SPS - Text 10
0	0	0	1	0	1	1	SPS - Text 11
0	0	0	1	1	0	0	SPS - Text 12
							SPS - Text 13
							...
1	1	0	0	1	0	0	SPS - Text 100

Texte eingeben:

Die Textdateien (*.DAT) werden im DATA-Text-Editor bearbeitet. Der Editor wird mit dem Befehl "DTFEDIT" oder der Tastenkombination [Strg+Page up] aufgerufen. Nun können Sie die Texte Ihrer Applikation anpassen.

Dabei ist wichtig, daß Sie den Text immer mit dem Endezeichen "\"" abschließen.

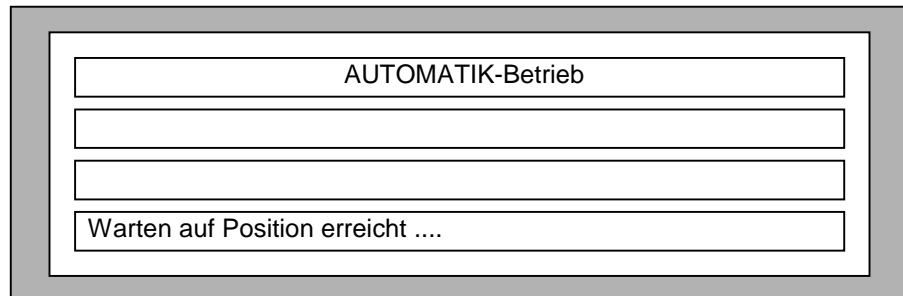
Eingangsliste:

E	Sender	Empfänger	Beschreibung mit Signalpegel
E1	SPS	Comtac 2000/3000	1 Text-Bit 0
E1			0
E2	SPS	Comtac 2000/3000	1 Text-Bit 1
E2			0
E3	SPS	Comtac 2000/3000	1 Text-Bit 2
E3			0
E4	SPS	Comtac 2000/3000	1 Text-Bit 3
E4			0
E5	SPS	Comtac 2000/3000	1 Text-Bit 4
E5			0
E6	SPS	Comtac 2000/3000	1 Text-Bit 5
E6			0
E7	SPS	Comtac 2000/3000	1 Text-Bit 6
E7			0
E8	SPS	Comtac 2000/3000	2. Zeile anwählen
E8			0
E9	SPS	Comtac 2000/3000	3. Zeile anwählen
E9			0
E10	SPS	Comtac 2000/3000	4. Zeile anwählen
E10			0

10. Automatisierungstechniker

10.1 Menü: [F5] Automatikbetrieb

Oberfläche:



Aktivieren des Automatikbetriebes

Drücken der Funktionstaste [F5] am COMTAC.

➡ Voraussetzung ist, daß ab Zeile 20300 der Automatikbetrieb programmiert ist.

Unter-/Abbrechen des Automatikbetriebes

Der Automatikbetrieb kann unterbrochen, wieder fortgesetzt oder abgebrochen werden.

Durch Drücken von '[STOP]' oder 'NOTAUS' wird der Automatikbetrieb unterbrochen.

Nach Quittieren Taste 'Start' drücken:

➡ Automatikbetrieb wird fortgesetzt.

Nach Quittieren Taste '[STOP]' drücken:

➡ Automatikbetrieb wird abgebrochen; Es wird ins Hauptmenü 'Anwahl der Betriebsarten mit FKT.-Tasten' verzweigt.

Die nötigen Parameter, wie Rampen und Geschwindigkeiten der Achsen, sind im 'Parametrierbetrieb [F6]' einzustellen.

10.2 Zugriff auf Parameter und Positionen

Maschinenparameter:

Nachdem die Parameter und Positionen im Parametrierbetrieb eingegeben wurden, können Sie auf diese im Programm folgendermaßen zugreifen:

Auf die Maschinenparameter haben Sie über die Matrix (Tabelle) KP(1...MPA) zugriff. MPA ist dabei die in CT.(132) angegebenen Anzahl der Maschinenparameter.

KP(1) = Maschinenparameter 1

KP(2) = Maschinenparameter 2

...

Achsparemeter:

Achsparemeter 1-8: Diese werden mit der Variablen SD für die Positioniermodule ausgewählt.

Bei SD=0 (schnell) gelten die Parameter für schnelle Positionierung (Parameter 1 und 2); bei SD=1 (langsam) gelten die Parameter 3 und 4.

Die Parameter 5 bis 8 werden beim Handbetrieb jeweils für "schnell" und "langsam" ausgewählt.

Achsparemeter 9-10: Auf diese Parameter benötigen Sie kein Zugriff. Diese Parameter werden vom Modul "Bahnpositionierung" zur Begrenzung der einzelnen Achsgeschwindigkeit und Rampenzeit verwendet. Geben Sie diese Grenzwerte für jede Achse ein.

Achsparemeter 11-25: Diese Parameter können Sie direkt für Ihr Automatikprogramm einsetzen. Sie sind in der Matrix (Tabelle):

AXP(1 ... AXPmax, 1 ... AA)

mit:

AXPmax: Anzahl der Achsparemeter (CT:(135))

AA: Anzahl der Achsen

In dieser Tabelle sind alle Achsparemeter abgespeichert.

Bsp.: AXP (2,3) = Die Beschleunigungszeit für schnelle Geschwindigkeit der 3. Achse

Achspositionen:

Auf die Achspositionen des aktuellen Bearbeitungsprogramms greifen Sie über die Positionsnummern zu. Dazu wird die Matrix (Tabelle) POSN(..) verwendet.

Aufbau von POSN(AA) mit AA: Anzahl der Achsen

Bsp.: POSN(1)=12; Ausgewählt ist damit die 12. Position der Achse 1.



Das Bearbeitungsprogramm können Sie mit dem Modul "Bearbeitungsprogramm laden" wechseln.

Nach dem Starten des COMTAC-Software für die Automation "CTC_Pro" ist immer des 1. Bearbeitungsprogramm aktiv.

10.3 Automatik-Betrieb / Achssteuerung

10.3.1 Modul: Streckenpositionierung

Funktion: Positionieren einer COMPAX - Achse.

Variablen:

POSN(..): Positionsnummer, die angefahren werden soll.

POSN(x)=y; x: Achsnummer; y: Positionsnummer dieser Achse

POSN(A1)=1 mit A1=1 bedeutet, daß die 1. Position mit der 1. Achse angefahren wird.

Das Feld A(..) ist für jede Achse vorbelegt (siehe unten).

A(0) = Achsnummer, in einem 4 Bit Binärwert ist jeder Achse ein Bit zugeordnet.

Um eine Achse anzuwählen muß das entsprechende Bit gesetzt werden.

SD = Merker, für schnelles Fahren = 0 / langsames Fahren = 1

A = Achsnummer mit A1=1; A2=2; A3=3; A4=4

Strings:

\$(11)-\$(14) = Display-Zeilen 1-4

\$(10), \$(26), \$(80), \$(76), \$(66), \$(65) = Anzeigestrings

Felder:

POS(..) = Feld, in dem die Positionsvorgaben im Automatikbetrieb stehen.

POS(0)=zu positionierende Achse; POS(zu positionierende Achse)=Positionswert

PD(..) = Feld, in dem die Werte der Teachpunkte stehen.

A(..) = Feld, in dem die Binärwerte der Achsen stehen.

A(1) = 0001B

A(2) = 0010B

A(3) = 0100B

A(4) = 1000B

Code:

```
24000 LABEL "STRECKENPOSITIONIERUNG": !
```

```
24010 SPD(0)=A(0) : ACC(0)=A(0) : ! schnell|fast
```

```
24020 SPD(A)=AXP(1+SD*2,A) : ACC(A)=AXP(2+SD*2,A) : GOSUB 1300
```

```
24030 POS(0)=A(0) : POS(A)=PD(POSN(A),A) : GOSUB 1200
```

```
24040 DO : !
```



```
24050 $(14)=$(65) : DISP TABXY(31,4),"A",VAL$(A),"=",CPXPOS(A) : GOSUB 2000
24060 UNTIL WAITPOS (A) : POS(0)=0 : $(14)="..." : RETURN
24070 !
```

Beschreibung:

Zeile 24010: In die Tabellen SPD(0) und ACC(0) wird der Binärcode der ausgewählten Achse geschrieben.

Zeile 24020: In die Tabellen SPD(A) und ACC(A) wird der Wert der Achsparameter geschrieben.

AXP(1+SD*2,A) = AXP(Achsparameter-Nr.+SD*2,Achs-Nr.); SD=Schnell/Langsam.

Zeile 24030: In die Tabelle POS(0) wird der Binärwert der ausgewählten Achse geschrieben. In die Tabelle POS(A) wird der Positionswert aus der Tabelle PD(..) geschrieben. Die Positionierung wird gestartet. PD(1,A) = PD(Positionnummer,Achsnummer).

Zeile 24040-24060: Es wird auf Position Erreicht abgefragt. Während der Positionierung wird die Istposition der Achse angezeigt.

Achtung:

In der Zeile 24060 wird die Tabelle POS(0)=0 gesetzt. Dies ist wichtig um eine korrekte Stop/Weiter-Funktion zu realisieren.

10.3.2 Modul: Bahnpositionierung

Funktion:

Positionieren einer COMPAX - Achse.

Variablen:

POSN(..): Positionsnummer, die angefahren werden soll.

POSN(x)=y; x: Achsnummer; y: Positionsnummer dieser Achse

POSN(A1)=1 mit A1=1 bedeutet, daß die 1. Position mit der 1. Achse angefahren wird.

Die Positionsnummer wird zweimal - für jede Achse, die verfahren werden soll - angegeben

Das Feld A(..) ist für jede Achse und für jede Achskombination vorbelegt (siehe unten).

A(0) = Achsnummer der ausgewählten Achsen, als 4 Bit Binärwert in dem jeder Achse ein Bit zugeordnet ist.

Um eine Achse anzuwählen muß das entsprechende Bit gesetzt werden.

Das Feld A(..) ist für jede Achse und für jede Achskombination vorbelegt (siehe unten).

SD = Merker, für schnelles Fahren = 0 / langsames Fahren = 1

Strings:

\$(11)-\$(14) = Display-Zeilen 1-4

\$(10), \$(26), \$(80), \$(76), \$(66), \$(65) = Anzeigestrings

Felder:

POS(..) = Feld, in dem die Positionsvorgaben im Automatikbetrieb stehen.

POS(0)=zu positionierende Achse; POS(zu positionierende Achse)=Positionswert

PD(..) = Feld, in dem die Werte der Teachpunkte stehen.

A(..) = Feld, in dem die Binärwerte der Achsen stehen.

A(1) = 0001B A(12) = 0011B A(23) = 0110B A(34) = 1100B

A(2) = 0010B A(13) = 0101B A(24) = 1010B

A(3) = 0100B A(14) = 1001B

A(4) = 1000B

Code:

24500 LABEL "BAHNPOSITIONIERUNG": !

24510 SPD(0)=A(0) : ACC(0)=A(0) : POS(0)=A(0)

24520 FOR A=1 TO AA

24530 IF (A(0)@(A-1))=0 THEN GOTO 24550

24540 POS(A)=PD(POSN(A),A) : SPD(A)=AXP(1+SD*2,A) : ACC(A)=AXP(2+SD*2,A)

24550 NEXT A

24560 ! GOSUB 1300 ! Ist bei gemeinsamer Fahrt nicht notwendig!!!

```

24570 !
24580 GOSUB 26000 : GOSUB 1200 : ! Start
24590 !
24600 DO : !
24610 ON AA GOTO 24610,24620,24630,24640,24650
24620 UNTIL WAITPOS A1 : POS(0)=0 : $(14)="..." : GOTO 24660
24630 UNTIL WAITPOS A1,A2 : POS(0)=0 : $(14)="..." : GOTO 24660
24640 UNTIL WAITPOS A1,A2,A3 : POS(0)=0 : $(14)="..." : GOTO 24660
24650 UNTIL WAITPOS A1,A2,A3,A4 : POS(0)=0 : $(14)="..." : GOTO 24660
24660 RETURN

```

Beschreibung:

Zeile 24510: In die Tabellen SPD(0), ACC(0) und POS(0) wird der Binärcode der ausgewählten Achse geschrieben.

Zeile 24520-24550: In einer Schleife wird in die Tabellen SPD(A) und ACC(A) der Wert der Achsparameter der ausgewählten Achsen geschrieben. In die Tabelle POS(A) werden die Positionswerte aus der Tabelle PD(POSN(A),A) der ausgewählten Achsen geschrieben.

Zeile 24560: Die Achsparameter brauchen nicht in die Achsen geschrieben werden, da dies vom Unterprogramm 26000 erledigt wird.

Zeile 24580: Das Unterprogramm, das die Grenzwerte der zwei beteiligten Achsen überwacht, wird aufgerufen. Die Achsen werden gestartet.

Zeile 24600-24650: Es wird auf Position Erreicht abgefragt.

Achtung:

In den Zeilen 24620 bis 24650 wird die Tabelle POS(0)=0 gesetzt. Dies ist wichtig um eine korrekte Stop/Weiter-Funktion zu realisieren.

10.3.3 DB [F5] Automatik-Betrieb

Funktion:

Der Automatikbetrieb ist das Herzstück jeder Anlage. Hier finden in anderen Betriebsarten eingestellte Parameter Anwendung. Diese Betriebsart kann hier jedoch nur in einer minimalen Form dargestellt werden, da jede Anlage einen spezifischen und auf Ihre Ansprüche definierten Ablauf hat.

Variablen:

AB = Merker für den Automatikbetrieb = 1
 M0 = Merker für den Maschinen-Null-Punkt angefahren = 1
 SP = Konstante für die [STOP] -Taste am COMTAC
 A = Achsnummer
 A1 =1 = Achse 1
 A2 =2 = Achse 2
 A3 =3 = Achse 3
 A4= 4 = Achse 4
 AS = Ausgangsnummer
 EG = Eingangsnummer
 SN = Stringnummer
 SD = Merker, für schnelles Fahren =0 / langsames Fahren = 1
 RP = Variable für Ruheposition = z.B. 0 mm
 WT = Wartezeit nach Positionierbefehlen; WT = 80ms

Strings:

\$(11)-\$(14) = Display-Zeilen 1-4
 \$(10), \$(26), \$(80), \$(76), \$(66), \$(65) = Anzeigestrings

Felder:

POS(..) = Feld, in dem die Positionsvorgaben im Automatikbetrieb stehen.
 POS(0)=zu positionierende Achse; POS(zu positionierende Achse)=Positionswert
 A(..) = Feld, in dem die Binärwerte der Achsen stehen.

A(1) = 0001B	A(12) = 0011B	A(23) = 0110B	A(34) = 1100B
A(2) = 0010B	A(13) = 0101B	A(24) = 1010B	
A(3) = 0100B	A(14) = 1001B		
A(4) = 1000B			

Code:

20000 ! DB= AUTOMATIK|automatic mode =====

Maschinennull
anfahren falls
noch nicht
erfolgt.

Als Beispiel wird
der Maschinen-
null der Achsen 1
und 2 angefahren:
zuerst Achse
2 dann Achse 1.
Passen Sie die
Reihenfolge und
die Achsen an
Ihre Anwendung
an!

Ab Zeile 20300
programmieren
Sie Ihr Auto-
matikprogramm.

Als Bsp. sehen
Sie wie die
Module für
COMPAX -
Positionierungen
aufgerufen
werden.

```

20010 !
20020 AB=TRUE
20030 GOSUB 50000 : CLRLED 1 : CLRLED 2 : CLRLED 3 : CLRLED 4
20040 $(11)=$(26) : $(12)=$(10) : $(13)=$(10) : $(14)="" : GOSUB 2000
20050 ENABLE ONKEY SP : ENABLE ONINP 2
20060 GOSUB 1400 : ! Auto-Speed, Accel, POS() reset
20070 !
20080 IF M0=TRUE THEN GOTO 20300
20090 ! Mach-Null abfahren, wenn nötig|Drive to Zero-Point if necessary
20100 $(14)=$(80) : GOSUB 2000
20110 DO : !
20120 UNTIL KBDCODE =ST
20130 $(14)=$(76) : GOSUB 2000
20140 A=A2 : $(1)="PH" : GOSUB 1000 : WAIT WT
20150 DO : !
20160 UNTIL (CPXSTS(A)@2)=TRUE
20170 $(14)=$(66) : GOSUB 2000 : $(1)="PA0" : GOSUB 1000 : WAIT WT
20180 DO : !
20190 UNTIL (CPXSTS(A)@4)=TRUE
20200 !
20210 $(14)=$(76) : GOSUB 2000
20220 A=A1 : $(1)="PH" : GOSUB 1000 : WAIT WT
20230 DO : !
20240 UNTIL (CPXSTS(A)@2)=TRUE
20250 $(14)=$(66) : GOSUB 2000 : $(1)="PA0" : GOSUB 1000 : WAIT WT
20260 DO : !
20270 UNTIL (CPXSTS(A)@4)=TRUE
20280 M0=TRUE : FOR I=1 TO AA:M0=(M0'AND'CPXSTS(I)@2) : NEXT I
20290 !
20300 ! .....
20310 !
20320 !AS=2 : GOSUB 1700 : AS=3 : GOSUB 1700 : AS=3 : GOSUB 1500
20330 !EG=3 : GOSUB 1900 : IF EGS<>FALSE THEN GOTO 20360
20340 !SN=81 : GOSUB 50600 : GOTO 20310
20350 !
20360 ! Schnell : Achse 2 : PositionNr. 1|Fast : Axis 2 : PositionNo. 1
20370 SD=0 : A=A2: A(0)=A(2) : POSN(A)=1 : GOSUB "STRECKENPOSITIONIERUNG"
20380 !
20390 ! Schnell : Achse 1 : PositionNr. 1|Fast : Axis 1 : PositionNo. 1
20400 SD=0 : A=A1: A(0)=A(1) : POSN(A)=1 : GOSUB "STRECKENPOSITIONIERUNG"
20410 !
20420 ! Schnell : Achse 1 : PositionNr. 2|Fast : Axis 1 : PositionNo. 2
20430 SD=0 : A=A1: A(0)=A(1) : POSN(A)=2 : GOSUB "STRECKENPOSITIONIERUNG"
20440 !
20450 ! Langsam : Achse 2 : PositionNr. 2|Slow : Axis 2:PositionNo. 2
20460 SD=1 : A=A2: A(0)=A(2) : POSN(A)=2 : GOSUB "STRECKENPOSITIONIERUNG"
20470 !
20480 ! Langsam : Achse 2 : PositionNr. 2|Slow : Axis 2 : PositionNo. 2
20490 SD=1 : A=A2: A(0)=A(2) : POSN(A)=2 : GOSUB "STRECKENPOSITIONIERUNG"
20500 !
20510 ! Achse 1+2 : PositionNr. 3|Slow : Axis 1+2 : PositionNo. 3
20520 A(0)=A(12) : POSN(A1)=3 : POSN(A2)=3 : GOSUB "BAHNPOSITIONIERUNG"
20530 ! Es gilt die Bahngeschw. und Bahnbeschleunigung-; siehe Maschinenpara.
20540 GOTO 20350
20550 !

```

Bsp.:
MN-
Fahrt
der
Achse 2

Bsp.:
MN-
Fahrt
der
Achse 1

Beschreibung:

Zeile 20020: Der Merker für den Automatikbetrieb wird gesetzt.
 Zeile 20030: Die Funktionstasten werden gesperrt und die 4 LED's am COMTAC gelöscht
 Zeile 20040: Die \$(11)-\$(14) werden in den Display-Zeilen 1-4 dargestellt.
 Zeile 20050: Der [STOP]-Interrupt wird freigegeben (STOP-Taste und Eingang 2 des COMTAC)
 Zeile 20060: Die Felder POS(), SPD(), ACC() werden zurückgesetzt.
 Zeile 20080: Falls der Maschinennullpunkt schon angefahren wurde, wird die M0-Anfahroutine übersprungen.

Zeile 20100: Textanzeige zum Starten des M0-Punktes.
 Zeile 20110-20120: Warten auf Drücken der Start-Taste.
 Zeile 20130: Textanzeige M0-Punkt wird angefahren.
 Zeile 20140: Start der Achse A2 zum M0-Punkt anfahren; die programmierte Zeit WT wird gewartet, bis das Signal M0-Punkt erreicht abgefragt wird.
 Zeile 20150-20160: Abfrage auf M0-Punkt erreicht.
 Zeile 20170: Textanzeige Realnull anfahren und Starten des R0-Punktes; Zeit WT abwarten.
 Zeile 20180-20190: Abfrage auf R0-Punkt erreicht.
 Zeile 20210-20270: wie 20120-20190
 Zeile 20280: Nochmalige Abfrage, ob M0-Punkte angefahren wurden

Beispiel:

Zeile 20320: Zuweisung Ausgang AS=2, AS=3; Rücksetzen dieser Ausgänge; Setzen des Ausgangs A3 für einen Impuls.
 Zeile 20330-20340: Abfrage des Einganges 3; Wird dieser Eingang innerhalb einer programmierten Zeit nicht wahr, so wird ein Fehlerstring angezeigt. Es wird auf die Bestätigung mit **[ENT]** gewartet und der Eingang wieder abgefragt.
 Zeile 20370: Vorbereitung der Achse 2 zum Einzelstart: Setzen der Variable SD auf 0; d.h. schnelle Automatik-Geschwindigkeit. Die Achse A2 soll positioniert werden, d.h. in die Tabellen A(0) wird der Binärcode dieser Achse A2 geschrieben. Die Position, die angefahren werden soll, ist die Nummer 1.
 Zeile 20520: Vorbereitung der Achse 1+2 zum gemeinsamen Start: Als Geschwindigkeit und Rampe werden, falls physikalisch möglich, die entsprechenden Maschinenparameter eingesetzt (Berechnung ab Zeile 26000). Die Achsen A1+A2 sollen positioniert werden, d.h. in die Tabellen A(0) wird der Binärcode der Achse A12 geschrieben. Die Position, die angefahren werden soll, ist die Nummer 3.

Beispiel:

-

10.3.4 DH Eingabe Bearbeitungsprogramm-Nummer

Funktion:	Die Maschinenpositionen können in Bearbeitungsprogrammen gespeichert werden. Somit hat man z.B. die Möglichkeit jeweils 20 verschiedene Positionen in 3 Bearbeitungsprogrammen zu hinterlegen. Durch Aufruf eines Bearbeitungsprogrammes werden die Werte der darin enthaltenen Positionen geladen.
Variablen:	BP = Bearbeitungsprogramm-Nummer SA = Stellenanzahl zum Eingeben UG = Untergrenze zum Überwachen des Eingabewertes OG = Obergrenze zum Überwachen des Eingabewertes KK = Eingabewert
Strings:	\$(3), \$(83)
Felder:	-
Code:	35000 ! DH= Eingabe der BearbeitungsProgrammNummer Input Prog-Nr.= 35010 ! 35020 KK=BP : UG=1 : OG=BN : SA=2 : \$(3)=\$(83) : GOSUB 52000 : BP=KK 35030 RETURN 35040 !
Beschreibung:	Zeile 35020: Der Vorgabewert wird in die Variable KK geschrieben. Es werden die Untergrenze, Obergrenze, Stellenanzahl und der Eingabetext zugewiesen. Es wird ins Eingabeprogramm gesprungen. Der Eingabewert wird der Bearbeitungsprogramm-Nummer zugewiesen.
Beispiel:	Siehe Zeile: 35020

10.3.5 DD Bearbeitungsprogramm laden

Funktion:	Mit dieser Funktion ändern Sie das Bearbeitungsprogramm. Das Modul lädt die Positionen des angegebenen Bearbeitungsprogramms aus der Datei MAP.DAT in die entsprechende Tabelle. Danach können die Positionen über die Positionsnummer abrufen.
Variablen:	BP: Bearbeitungsprogrammnummer
Strings:	-
Felder:	-
Code:	<pre>30900 ! PD(..) laden load PD(..) 30910 FOR A=1 TO AA: ! Alle Achsen all axis 30920 FOR I=1 TO MPO : ! Position 1 bis MPO position 1 to max. position 30930 PD(I,A)= ;\$ (16) (((BP-1)*MPO*AA)+((A-1)*MPO)+I) : ! Position 30940 NEXT I 30950 NEXT A 30960 RETURN</pre>
Beschreibung:	In Zeile 30930 wird die Datei "MAP.DAT" in die Positionstabelle PD umgespeichert.
Beispiel:	BP =2 : GOSUB 30900

10.4 COMTAC / COMPAX Ausgänge setzen

10.4.1 AF Ausgang als Impuls setzen

Funktion:	Mit dieser Funktion können die Ausgänge vom COMTAC und den COMPAX-Achsen für eine programmierbare Zeit gesetzt werden. Das sind je Modul max. 16 Ausgänge (32 bei COMTAC 3000). Werden die unteren Ausgänge 1-6 des COMPAX verwendet, so müssen diese maskiert werden. (siehe CPXOMASK)
Variablen:	WI = Impulsdauer (fest belegt im Programmmodul CD-Variablen-Initialisierung mit 100ms)
Strings:	-
Felder:	-
Code:	<pre>1500 ! AF=Ausgang Impulsmaessig setzen set output for a impulse ===== 1510 ! 1520 GOSUB 1600 1530 WAIT WI 1540 GOSUB 1700 1550 RETURN 1560 !</pre>
Beschreibung:	<p>Zeile 1520: Ein zuvor zugewiesener Ausgang AS wird abhängig von seiner Nummer (1..96) an dem jeweiligen Modul (COMTAC, COMPAX 1..4) gesetzt (siehe Ausgang setzen).</p> <p>Zeile 1530: Die Wartezeit WI läuft ab.</p> <p>Zeile 1540: Der Ausgang wird wieder gelöscht.</p> <p>Zeile 1550: Rücksprung</p>

Beispiel: AS =3 : GOSUB 1500
Zeile:20320

10.4.2 AG Ausgang setzen

Funktion: Mit dieser Funktion können die Ausgänge vom COMTAC und den COMPAX-Achsen gesetzt werden. Das sind je Modul max. 16 Ausgänge (32 bei COMTAC 3000). Werden die unteren Ausgänge 1-6 des COMPAX verwendet, so müssen diese maskiert werden. (siehe CPXOMASK)

Variablen: AS = Ausgangsnummer
Comtac-AS = 1-16 (32)
A1 = Achse 1 = 1 AS = 33-48
A2 = Achse 2 = 2 AS = 49-64
A3 = Achse 3 = 3 AS = 65-80
A4 = Achse 4 = 4 AS = 81-96

Strings: -

Felder: -

Code: 1600 ! AG= Ausgang setzen|set output =====
1610 !
1620 IF AS<=16 THEN SETOUT AS : RETURN
1630 IF AS<=48 THEN CPXOUT(A1;(AS-32))=1 : RETURN
1640 IF AS<=64 THEN CPXOUT(A2;(AS-48))=1 : RETURN
1650 IF AS<=80 THEN CPXOUT(A3;(AS-64))=1 : RETURN
1660 IF AS<=96 THEN CPXOUT(A4;(AS-80))=1 : RETURN
1670 !

Beschreibung: Ein zuvor zugewiesener Ausgang AS wird abhängig von seiner Nummer (1..96) an dem jeweiligen Modul (COMTAC, COMPAX 1..4) gesetzt.

Beispiel: -

10.4.3 AH Ausgang löschen

Funktion: Mit dieser Funktion können die Ausgänge vom COMTAC und den COMPAX-Achsen gelöscht werden. Das sind je Modul max. 16 Ausgänge (32 bei COMTAC 3000). Werden die unteren Ausgänge 1-6 verwendet, so müssen diese maskiert werden. (siehe CPXOMASK)

Variablen: AS = Ausgangsnummer
Comtac-Ausgänge = 1-16 (32)
A1 = Achse 1 = 1 AS = 33-48
A2 = Achse 2 = 2 AS = 49-64
A3 = Achse 3 = 3 AS = 65-80
A4 = Achse 4 = 4 AS = 81-96

Strings: -

Felder: -

Code: 1700 ! AH= Ausgang loeschen|clear output =====
1710 !
1720 IF AS<=16 THEN CLROUT AS : RETURN

```

1730 IF AS<=48 THEN CPXOUT(A1;(AS-32))=0 : RETURN
1740 IF AS<=64 THEN CPXOUT(A2;(AS-48))=0 : RETURN
1750 IF AS<=80 THEN CPXOUT(A3;(AS-64))=0 : RETURN
1760 IF AS<=96 THEN CPXOUT(A4;(AS-80))=0 : RETURN
1770 !

```

Beschreibung: Ein zuvor zugewiesener Ausgang AS wird abhängig von seiner Nummer (1..96) an dem jeweiligen Modul (COMTAC, COMPAX 1..4) gelöscht.

Beispiel: AS =2 : GOSUB 1700
Siehe auch Zeile:20320

10.5 COMTAC / COMPAX Eingänge lesen

10.5.1 AI Eingang einlesen

Funktion: Mit dieser Funktion können alle Eingänge vom COMTAC und den COMPAX-Achsen eingelesen werden. Das sind je Modul max. 16 Eingänge (32 bei COMTAC 3000).

Variablen: EGS = Eingangsstatus
EG = Eingangsnummer
COMTAC = 1-16 (32)
A1 = Achse 1 = 1 EG = 33-48
A2 = Achse 2 = 2 EG = 49-64
A3 = Achse 3 = 3 EG = 65-80
A4 = Achse 4 = 4 EG = 81-96

Strings: -

Felder: -

Code:

```

1800 ! AI= Eingang einlesen|read input =====
1810
1820 EGS=FALSE
1830 IF EG<=16 THEN EGS=IN(EG) : RETURN
1840 IF EG<=48 THEN EGS=CPXINP(A1)@(EG-33) : RETURN
1850 IF EG<=64 THEN EGS=CPXINP(A2)@(EG-49) : RETURN
1860 IF EG<=80 THEN EGS=CPXINP(A3)@(EG-65) : RETURN
1870 IF EG<=96 THEN EGS=CPXINP(A4)@(EG-81) : RETURN
1880 !

```

Beschreibung: Ein zuvor zugewiesener Eingang EG wird abhängig von seiner Nummer (1..96) an dem jeweiligen Modul (COMTAC, COMPAX 1..4) gelesen. Dabei ist zu beachten, daß ein Eingang des COMTAC 0 oder 1 zurückliefert, ein Eingang des COMPAX jedoch 0 oder 65535. (@=Bitabfrage)

Beispiel: EG =1 : DO : GOSUB 1800
UNTIL EGS<>FALSE

Zeile:63430 - 63450

10.5.2 AJ Eingang einlesen mit Zeitabfrage

Funktion:	Mit dieser Funktion können alle Eingänge vom COMTAC und den COMPAX-Achsen eingelesen werden. Das sind je Modul max. 16 Eingänge (32 bei COMTAC 3000). Ist nach einer programmierbaren Zeit der Eingang nicht aktiv, so kann entsprechend eine Fehlermeldung ausgegeben werden. Beispiele hierzu sind pneumatische Drehgreifer, Backengreifer, usw..
Variablen:	EGS = Eingangsstatus EG = Eingangsnummer COMTAC = 1-16 (32) A1 = Achse 1 = 1 EG = 33-48 A2 = Achse 2 = 2 EG = 49-64 A3 = Achse 3 = 3 EG = 65-80 A4 = Achse 4 = 4 EG = 81-96 TRUE = Konstante = 65535 = Wahr FALSE = Konstante = 0
Strings:	\$(14) = Anzeigestring in Zeile 4 des Displays \$(78) = vorgelegter String
Felder:	KP(9) = Zeitdauer, in der der Eingang vorhanden sein muß. Wird im Parametrierbetrieb-Maschinenparameter- eingestellt.
Code:	<pre> 1900 ! AJ= Eingang einlesen mit Zeitabfrage read input for defined time ===== 1905 1910 EGS=FALSE : COUNT=0 : ENABLE ONCOUNT 1915 IF EG<=16 THEN EGS=IN(EG) : GOTO 1940 1920 IF EG<=48 THEN EGS=CPXINP(A1)@(EG-33) : GOTO 1940 1925 IF EG<=64 THEN EGS=CPXINP(A2)@(EG-49) : GOTO 1940 1930 IF EG<=80 THEN EGS=CPXINP(A3)@(EG-65) : GOTO 1940 1935 IF EG<=96 THEN EGS=CPXINP(A4)@(EG-81) : GOTO 1940 1940 IF EGS=TRUE'OR'EGS=1 THEN \$(14)=".." : DISABLE ONCOUNT : RETURN 1945 IF COUNT>=KP(9) THEN EGS=FALSE : DISABLE ONCOUNT : RETURN 1950 \$(14)=\$(78),VAL\$(EG) : GOSUB 2000 : GOTO 1915 1955 ! </pre>
Beschreibung:	<p>Zeile 1910: Die Variable EGS wird zurückgesetzt. Der Zähler COUNT wird 0 gesetzt und freigegeben.</p> <p>Zeile 1915-1935: Ein zuvor zugewiesener Eingang EG wird abhängig von seiner Nummer (1..96) an dem jeweiligen Modul (COMTAC, COMPAX 1..4) gelesen. Dabei ist zu beachten, daß ein Eingang des COMTAC 0 oder 1 zurückliefert, ein Eingang des COMPAX jedoch 0 oder 65535.</p> <p>Zeile 1940: Wenn der Eingangsstatus wahr ist, wird der Zähler gestoppt und zurückgesprungen.</p> <p>Zeile 1945-1950: In einer Schleife wird der Wert des Counters, der alle Sekunde erhöht wird mit dem Wert in KP(9) (programmierte Zeit) verglichen. Gleichzeitig wird die aktuelle Eingangsnummer angezeigt. Wird der Eingang vor dem Ablauf der Überwachungszeit aktiv, so wird das Unterprogramm mit EGS=TRUE oder 1 sofort beendet. Wird die Überwachungszeit überschritten, so wird das Unterprogramm mit EGS=False verlassen.</p>
Beispiel:	EG=3 : GOSUB 1900 siehe Zeile: 20330 - 20340

10.6 COMTAC Ein- und Ausgabe

10.6.1 AK Display anzeigen mit \$(11)..\$(14)

Funktion:	In einer Betriebsart werden bestimmte Zustände am Display angezeigt. Dies können Stückzahlen, Positionen, ... sein. Ist diese Betriebsart (wie der Automatikbetrieb) unterbrechbar, z.B. durch STOP, so wird ein neuer Text angezeigt ("STOP"). Soll nach dieser Unterbrechung der alte Zustand wieder angezeigt werden, so muß dieser zuvor zwischengespeichert werden. Alle Strings werden daher in die Strings \$(11) .. \$(14) geschrieben und angezeigt. \$(11) steht hierfür für Zeile 1 des Displays usw.
Variablen:	-
Strings:	\$(11) = Anzeigestring in Zeile 1 des Displays \$(12) = Anzeigestring in Zeile 2 des Displays \$(13) = Anzeigestring in Zeile 3 des Displays \$(14) = Anzeigestring in Zeile 4 des Displays \$(99) = vorgelegter String = Projektname
Felder:	-
Code:	<pre> 2000 ! AK= Display komplett anzeigen mit \$(11) bis \$(14) display line 1-4 ===== 2010 2020 CLEAR D,1 2030 DISP TABXY(INT((41-(LEN(\$(11))+LEN(\$(99))))/2),1),\$(99),\$(11) 2040 CLEAR D,2 : DISP TABXY(1,2),\$(12) 2050 CLEAR D,3 : DISP TABXY(1,3),\$(13) 2060 CLEAR D,4 : DISP TABXY(1,4),\$(14) 2070 RETURN 2080 ! </pre>
Beschreibung:	<p>Zeile 2020-2030: Zeile 1 wird gelöscht und der String 11 zusammen mit dem Projektnamen zentriert dargestellt.</p> <p>Zeile 2040-2060: Die Zeilen 2..4 werden gelöscht und die Strings 12..14 am Display linksbündig angezeigt.</p>
Beispiel:	<p>\$(14)=\$(65) : GOSUB 2000</p> <p>Zeile:24050</p>

10.6.2 AY INPUT

Funktion:	Diese Funktion wird für Zahleneingaben am COMTAC gebraucht. Es kann die Eingabe-position, die Anzahl der Stellen und die Unter-/Obergrenze eingegeben werden.
Variablen:	<p>SN = Stringnummer</p> <p>SA = Stellenanzahl zum Eingeben</p> <p>UG = Untergrenze zum Überwachen des Eingabewertes</p> <p>OG = Obergrenze zum Überwachen des Eingabewertes</p> <p>KK = Eingabewert</p>
Strings:	-
Felder:	-

Code:

```
52000 ! AY= INPUT|input =====
52010 !
52020 CURSOR_D 3 : CLEAR D,4
52030 SN=3 : GOSUB 2230
52040 INPKBD TABXY(1,4),SA,[UG,OG],KK : GOSUB 2134
52050 CURSOR_D 0
52060 RETURN
52070
```

Beschreibung:

Zeile 52020: Der Cursor wird als Unterstrich dargestellt und die Zeile 4 des Displays gelöscht.

Zeile 52030: Der String mit der Nummer 3 wird in Zeile 4 des Displays angezeigt.

Zeile 52040: Es wird die Eingabe verlangt. Der eingegebene Wert steht in der Variablen KK. Die Zeilen 3,4 des Displays werden gelöscht.

Zeile 52050: Der Cursor wird unsichtbar dargestellt.

Zeile 52060: Rücksprung

Beispiel:

```
KK=A:UG=1 : OG=AA:SA=1 : $(3)=$(47) : GOSUB 52000 : A=KK
```

Zeile: 52120

11. Basic - Programmierer

11.1 Übersicht

Basic-Programmierer mit COMTAC - COMPAX - Verbund.

- Für spezielle Lösungen erhalten Sie mit der COMTAC-Software für die Automation "CTC_Pro" eine hilfreiche Modulsammlung.
- Die Programme sind offen zugänglich und können von Ihnen angepasst werden oder zeigen Lösungsbeispiele.
- Die Ablagestruktur für Daten und Text kann genutzt werden.

⇒ Für die Ablaufsteuerung sind COMTAC - Programmierkenntnisse notwendig
- die Kommunikation mit COMPAX ist programmiert!

⇒ Die Prozeß- und Ablaufsteuerung befindet sich im COMTAC.

11.2 Variablen und Tabellen

A(1..34) : Tabelle enthält die binären Werte (siehe wie folgt) der an einer Positionierung beteiligten Achsen.

A(0) wird dieser Binärwert zugewiesen; dieser wird bei der Positionierung in die Tabelle POS(0) umgespeichert, und dient als Speicher, um nach einem STOP wieder die Achsen neu zu starten.

A(1)=1 :	Achse 1	0 0 0 1
A(2)=2 :	Achse 2	0 0 1 0
A(12)=3 :	Achse 1+2	0 0 1 1
A(3)=4 :	Achse 3	0 1 0 0
A(13)=5 :	Achsen 1+3	0 1 0 1
A(23)=6 :	Achsen 2+3	0 1 1 0
A(4)=8 :	Achse 4	1 0 0 0
A(14)=9 :	Achsen 1+4	1 0 0 1
A(24)=10 :	Achsen 2+4	1 0 1 0
A(34)=12 :	Achsen 3+4	1 1 0 0

POSN(1..MPO) : Tabelle enthält die Positionsnr. der gerade aktuellen Positionierung; MPO=max. Anzahl Positionen

FA(1..AA) : Tabelle für Fehlerachse Compax; AA=max. Achsanzahl (1..4)

FE(1..AA) : Tabelle fuer Fehler des jeweiligen Compax; AA=max. Achsanzahl (1..4)

NG(1..AA) : negative Endgrenzen; AA=max. Achsanzahl (1..4)

PG(1..AA) : Positive Endgrenzen; AA=max. Achsanzahl (1..4)

RN(1..AA) : Realnullpunkt; AA=max. Achsanzahl (1..4)

KO(1..AA) : fuer Realnullpunkt-Korrektur; AA=max. Achsanzahl (1..4)

KP(1..MPA) : Maschinenparameter; MPA=max. Anzahl Maschinenparameter

KPO(1..MPA) : Obergrenze Maschinenparameter; MPA=max. Anzahl Maschinenparameter

KPU(1..MPA) : Untergrenze Maschinenparameter; MPA=max. Anzahl Maschinenparameter

KPS(MPA) : Stellenanzahl Maschinenparameter; MPA=max. Anzahl Maschinenparameter

AXP(1..AXP,1..AA): Achsparameter; AXP=max. Anzahl Achsparameter, AA=max. Achsanzahl (1..4)

AXO(1..AXP,1..AA):	Obergrenze bei der Eingabe der Achsp.; AXP=max. Anzahl Achsp. parameter, AA=max. Achsanzahl (1..4)
AXU(1..AXP,1..AA):	Obergrenze bei der Eingabe der Achsp.; AXP=max. Anzahl Achsp. parameter, AA=max. Achsanzahl (1..4)
AXS(1..AXP,1..AA):	Stellenanzahl bei der Eingabe der Achsp.; AXP=max. Anzahl Achsp. parameter, AA=max. Achsanzahl (1..4)
PD(1..MPO,1..AA):	Maschinenpositionen (pro Bearbeitungsprogramm); MPO=max. Anzahl Masch-pos., AA=max. Achsanzahl
POS(0..AA) :	beinhaltet die aktuellen Werte der anzufahrenden Positionen; AA=max. Achsanzahl (1..4) POS(0) = Binärwert der beteiligten Achsen; POS(Achsnummer)=Wert der Position
SPD(0..AA) :	beinhaltet die aktuellen Geschwindigkeitswerte der anzufahrenden Position; AA=max. Achsanzahl (1..4) SPD(0) = Binärwert der beteiligten Achsen; SPD(Achsnummer)=Wert der Geschwindigkeit [%]
ACC(1..AA) :	beinhaltet die aktuellen Rampenwerte der anzufahrenden Position; AA=max. Achsanzahl (1..4) ACC(0) = Binärwert der beteiligten Achsen; ACC(Achsnummer)=Wert der Rampe [ms]
VB(1..AA) :	Bahngeschwindigkeit bei Start von 2 Achsen gemeinsam [%]; AA=max. Achsanzahl (1..4)
VBM(1..AA) :	Max. Achsgeschwindigkeit [m/s]; AA=max. Achsanzahl (1..4)
AL(1..AA) :	Bahnrampen bei Start von 2 Achsen gemeinsam [ms]; AA=max. Achsanzahl (1..4)
BS(1..AA) :	Beschleunigung bei Start von 2 Achsen gemeinsam [m/s ²]; AA=max. Achsanzahl (1..4)
BSM(1..AA) :	Min. Achsbeschleunigung bei Start von 2 Achsen gemeinsam [m/s ²]; AA=max. Achsanzahl (1..4)
LA(1..2) :	beinhaltet die 2 Achsnummern, der an der Bahnfahrt beteiligten Achsen (1..4)
WEG(1..AA) :	Bahnweg bei Start von 2 Achsen gemeinsam [mm]; AA=max. Achsanzahl (1..4)
P80(1..AA) :	Parameter 80 Compax-Achsen; AA=max. Achsanzahl (1..4)
P81(1..AA) :	Parameter 81 Compax-Achsen; AA=max. Achsanzahl (1..4)
P82(1..AA) :	Parameter 82 Compax-Achsen; AA=max. Achsanzahl (1..4)
P83(1..AA) :	Parameter 83 Compax-Achsen; AA=max. Achsanzahl (1..4)
P85(1..AA) :	Parameter 85 Compax-Achsen; AA=max. Achsanzahl (1..4)
P103(1..AA) :	Parameter 103 Compax-Achsen; AA=max. Achsanzahl (1..4)
P104(1..AA) :	Parameter 104 Compax-Achsen; AA=max. Achsanzahl (1..4)
P106(1..AA) :	Parameter 106 Compax-Achsen; AA=max. Achsanzahl (1..4)
VA(1..AA) :	Absolute Geschwindigkeit bei Start von 2 Achsen gemeinsam [m/s]; AA=max. Achsanzahl (1..4)
AS(1..96) :	Zum Ausgaenge setzen im Test E/A-Betrieb (16(32) CTC+max 4*16 CPX)

12. Anhang

Nachfolgend sind sämtliche Module erläutert, die in der COMTAC-Software für die Automation "CTC_Pro" enthalten sind.

Zur Automatisierung in Ihrer Anlage brauchen Sie diese Module in der Regel nicht; dazu reichen die unter "10.1 Menü: [F5] Automatikbetrieb" Seite 39 aufgeführten Module.

Das komplette Programmlisting finden Sie zusätzlich auf der beiliegenden Programmdiskette.

12.1 A Grundfunktionen

12.1.1 AA RS485-OUTPUT

Funktion:	Über den Feldbus wird ein Befehl vom COMTAC an ein COMPAX gesendet.
Variablen:	A = Achsnummer
Strings:	\$(1) = Ausgabestring
Felder:	-
Code:	<pre> 1000 ! AA= RS485-OUTPUT ===== 1010 ! 1020 OUTPUT 1,A;\$(1) 1030 RETURN 1040 ! </pre>
Beschreibung:	Über den Feldbus wird an das COMPAX mit der Achsnummer A der Ausgabestring \$(1) gesendet. Die Variable A und der String \$(1) müssen zuvor mit gültigen Werten beschrieben worden sein.
Beispiel:	<pre> A=1 : \$(1)="P11=1234.56" : GOSUB 1000 OUTPUT 1,1;\$(1) </pre> <p>Zeile: 63520, 50320</p>

12.1.2 AB RS485-OUTPUT/ENTER

Funktion:	Über den Feldbus wird ein Befehl vom COMTAC an ein COMPAX gesendet, der eine Antwort des COMPAX zur Folge hat.
Variablen:	A = Achsnummer
Strings:	\$(1) = Ausgabestring
Felder:	-
Code:	<pre> 1100 ! AB= RS485-OUTPUT/ENTER ===== 1110 ! \ = mit ENTER with ENTER 1120 OUTPUT 1,A\\$(1) 1130 RETURN 1140 </pre>

Beschreibung: Über den Feldbus wird an das COMPAX mit der Achsnummer A der Ausgabestring \$(1) gesendet. Die Variable A und der String \$(1) müssen zuvor mit gültigen Werten beschrieben worden sein. Im Rückgabestring \$(#1) steht die Rückantwort vom COMPAX.

Beispiel: A=1 : \$(1)="P11" : GOSUB 1100

⇒ \$(#1)=1234.56

Zeile: 50420, 50470

12.1.3 AC Achs-Positionierung starten

Funktion: Jede Positionierung im Automatikbetrieb wird durch dieses Unterprogramm gestartet. Dabei wird das Feld POS(..) benutzt. Binär codiert wird die gewünschte Achsnummer ins Feld POS(0) geschrieben.
Bit 0 = Achse 1; Bit 1 = Achse 2; Bit 2 = Achse 3; Bit 3 = Achse 4; (Wert 1;2;4;8). Ins Feld POS(I) wird die anzufahrende Position geschrieben.

Variablen: I = Schleifenzähler
AA = max. Achsanzahl
WT = Wartezeit nach der Ausgabe der Positionierung, sodaß das Signal 'POSITION ERREICHT' 0 wird
TRUE = Konstante = 65535 = Wahr

Strings: -

Felder: POS(0) = In dieses Feld wird bei Einzelpositionierungen die zu positionierende Achsnummer geschrieben (binär codiert)
POS(I) = In dieses Feld wird die anzufahrende Position geschrieben

Code:

```

1200 ! AC=Achs-Positionierung starten|start positioning =====
1210 !
1220 FOR I=1 TO AA
1230 IF (POS(0)@(I-1)=TRUE THEN OUTPUT 1,I;"PA",VAL$(POS(I))
1240 NEXT I
1250 !
1260 WAIT WT : RETURN
1270 !

```

Beschreibung: Zeile 1220-1240: In einer Schleife werden die angewählten Achsen mit den vorgegebenen Positionen gestartet. z.B.: POS(0)=A(A1), A(A2), A(A3), A(A4); POS(A1)=Position, POS(A2)=Position, POS(A3)=Position, POS(A4)=Position.
POS(0)@(I-1) = Bitabfrage, welche Achse angewählt wurde.
Zeile 1260: Nach dem Befehl der Positionierung wird eine Wartezeit WT gewartet, sodaß das Signal 'POSITION ERREICHT' auf 0 gehen kann. Dieses Signal wird dann beim Erreichen der Position wieder aktiv.

Beispiel: Siehe Zeile 20370, 24000-24030

12.1.4 AD Geschwindigkeit und Rampen setzen

Funktion: Vor den Positionierungen kann jeder Achse eine Geschwindigkeit und Rampe mitgeteilt werden. Sonst werden die Standardwerte aus den Parametern der COMPAX verwendet. Dazu werden die Felder SPD(..) und ACC(..) verwendet. Dabei wird binär codiert die gewünschte Achsnummer ins Feld SPD(0) und ACC(0) geschrieben. In die Felder SPD(I) und ACC(I) werden die Werte eingetragen.
Bit 0 = Achse 1; Bit 1 = Achse 2; Bit 2 = Achse 3; Bit 3 = Achse 4; (Wert 1;2;4;8)

Variablen:	<p>I = Schleifenzähler AA = max. Achsanzahl A1 = Achse 1 = 1 A2 = Achse 2 = 2 A3 = Achse 3 = 3 A4 = Achse 4 = 4 TRUE = Konstante = 65355 = Wahr</p>
Strings:	-
Felder:	<p>SPD(0) = In dieses Feld wird die Achsnummer eingetragen, welcher die Geschwindigkeit gesendet wird. SPD(I) = In dieses Feld wird der Wert der Geschwindigkeit eingetragen. ACC(0) = In dieses Feld wird die Achsnummer eingetragen, welcher die Rampenzeit gesendet wird. ACC(I) = In dieses Feld wird der Wert der Rampe eingetragen.</p>
Code:	<pre> 1300 ! AD= Speed und Rampen setzen set speed/accel ==== 1310 ! 1320 FOR I=1 TO AA 1330 IF (SPD(0)@(I-1)=TRUE THEN OUTPUT 1,I;"SD",VAL\$(SPD(I)) 1340 IF (ACC(0)@(I-1)=TRUE THEN OUTPUT 1,I;"AL",VAL\$(ACC(I)) 1350 NEXT I 1360 RETURN 1370 ! </pre>
Beschreibung:	<p>Zeile 1320-1350: In einer Schleife werden den angewählten Achsen die vorgegebene Geschwindigkeiten und Rampen vorgegeben. z.B.: SPD(0)=A(A1), A(A2), A(A3), A(A4), ACC(0)=A(A1), A(A2), A(A3), A(A4) $SPD(I) = AXP(1+SD*2,I)$ = Achsparameterwert der Achse I für Automatikgeschwindigkeit $AXP(1+SD*2,I) = AXP(\text{Achsparameter-Nr.}+SD*2,\text{Achs-Nr.})$; SD=schnell/langsam</p>
Beispiel:	Siehe Zeile 20370, 24000-24020

12.1.5 AE POS-Feld Reset

Funktion:	Löschen der Tabellen POS(..), SPD(..), ACC(..). Diese Tabellen werden im Automatikbetrieb zum Positionieren benutzt.
Variablen:	I = Schleifenvariable
Strings:	-
Felder:	<p>POS(0) = In dieses Feld wird bei Einzelpositionierungen die zu positionierende Achsnummer geschrieben (binär codiert) POS(I) = In dieses Feld wird die anzufahrende Position geschrieben SPD(0) = In dieses Feld wird die Achsnummer eingetragen, welcher die Geschwindigkeit gesendet wird. SPD(I) = In dieses Feld wird der Wert der Geschwindigkeit eingetragen. ACC(0) = In dieses Feld wird die Achsnummer eingetragen, welcher die Rampenzeit gesendet wird. ACC(I) = In dieses Feld wird der Wert der Rampe eingetragen.</p>
Code:	<pre> 1400 ! AE= POS-Feld Reset ===== 1410 ! 1420 FOR I=0 TO AA: POS(I)=0 : SPD(I)=0 : ACC(I)=0 : NEXT I : RETURN 1430 ! </pre>

Beschreibung: Zeile 1420: Alle Felder werden mit dem Wert 0 beschrieben.

Beispiel: Siehe Zeile: 20060

12.1.6 AL Zeilen löschen

Funktion: Mit dieser Funktion werden Display-Zeilen gelöscht und Strings angezeigt.

Variablen: SN = Stringnummer

Strings: -

Felder: -

```
Code: 2100 ! Al= Zeilen loeschen|clear lines =====
2101 CLEAR D,1 : RETURN : ! 1
2102 CLEAR D,2 : RETURN : ! 2
2103 CLEAR D,3 : RETURN : ! 3
2104 CLEAR D,4 : RETURN : ! 4
2114 CLEAR D : RETURN : ! alle Zeilen|all lines
2123 CLEAR D,2 : CLEAR D,3 : RETURN : ! 2-3
2124 CLEAR D,2 : CLEAR D,3 : CLEAR D,4 : RETURN : ! 2-4
2134 CLEAR D,3 : CLEAR D,4 : RETURN : ! 3-4
2140 !
2200 ! String SN in Zeile 1/2/3/4  ausgeben|display string SN in line 1/2/3/4
2210 CLEAR D,1 : DISP TABXY(1,1),$(SN) : RETURN
2220 CLEAR D,2 : DISP TABXY(1,2),$(SN) : RETURN
2230 CLEAR D,3 : DISP TABXY(1,3),$(SN) : RETURN
2240 CLEAR D,4 : DISP TABXY(1,4),$(SN) : RETURN
2250 !
2600 CLEAR D,1
2610 DISP TABXY(INT((41-(LEN$(SN))+LEN$(99)))/2,1),$(99),$(SN)
2620 RETURN
```

Beschreibung: Zeile 2101-2104: Die Zeilen 1-4 werden gelöscht.
 Zeile 2114: Alle Zeilen werden gelöscht.
 Zeile 2123: Zeile 2,3 werden gelöscht.
 Zeile 2124: Die Zeilen 2,3,4 werden gelöscht.
 Zeile 2134: Die Zeilen 3,4 werden gelöscht.
 Zeile 2210-2240: Die Zeilen 1-4 werden gelöscht und der Textstring mit der Nummer SN angezeigt.
 Zeile 2600-2620: Die Zeile 1 wird gelöscht und der Textstring mit der Nummer SN zusammen mit dem String \$(99) "Projektname" zentriert angezeigt. Die beiden Strings dürfen zusammen nicht länger als 40 Zeichen sein.

Beispiel: SN=27 : GOSUB 2600

Zeile 30020

12.1.7 AM Display Initialisierung

Funktion: Diese Funktion wird benötigt, um einen Grundzustand des Displays herbeizuführen. Wird während der Softwareerstellung im Editor das Programm mit Ctrl+C abgebrochen, so sind eventuell noch zyklische Anzeigen aktiv. Beim Durchlauf dieses Unterprogrammes werden diese ausgeschaltet.

Variablen: -

Strings: -

Felder: -

Code: 2700 ! AM= DISPLAY-Initialisierung|initialize display =====
 2710 !
 2720 ! Alle zyklischen Anzeigen ausschalten !!!|reset display !!!
 2730 CLEAR D : CURSOR_D 0 : DISP USING(0)
 2740 RETURN
 2750 !

Beschreibung: Zeile 2730: Das Display wird gelöscht, der Cursor wird unsichtbar geschaltet und die Formatierung wird ausgeschaltet.
 Zeile 2740 : Rücksprung

Beispiel: GOSUB 2700

Zeile:63040

12.1.8 AN Fehler-Routine

Funktion: Das COMTAC selbst generiert Fehler. Dies können z. B. Dateibefehle sein, wenn eine Diskette beschrieben werden soll und sie schreibgeschützt ist. Diese Fehler müssen "abgefangen" werden, sonst würde das Programm in der Fehlerzeile stehen bleiben. Durch diese Fehler-Routine werden die Fehler abgefangen, angezeigt und quittiert. Zuvor muß die Sprunganweisung und die Freigabe programmiert werden. (ONERR GOSUB 3000 : ENABLE ONERR)

Variablen: FZ = Fehlerzeile
 FE = Fehler
 SN = Stringnummer
 KC = Keyboard-Code
 ER = Konstante ENTER-Taste

Strings: ERRSTS\$ = Errorstatusstring
 \$(4) = Dummystring
 \$(71) = vorbelegter String
 \$(72) = vorbelegter String

Felder: -

Code: 3000 ! AN= ONERROR-ROUTINEN =====
 3010
 3020 DISABLE ONERR : DISABLE ONCPXERR : DISABLE ONKEY SP
 3030 GOSUB 50000
 3040
 3050 FZ=ERRSTSL
 3060 FE=ERRSTS
 3070 !
 3500 ! --- Reaktion auf COMTAC-Fehler|reaction of COMTAC faults -----

 3510 !
 3520 \$(4)=\$(72)[1,10],\$(72)[31,40] : SN=4 : GOSUB 2600
 3530 \$(4)=\$(71)[1,10],VAL\$(FE)," " : SN=4 : GOSUB 2220 : DISP ERRSTS\$
 3540 \$(4)=\$(71)[11,20],VAL\$(FZ) : SN=4 : GOSUB 2230
 3550 SN=56 : GOSUB 2240
 3560 DO
 3570 WHILE KBDCODE
 3580 DO : KC= KBDCODE
 3590 WHILE KC<>ER
 3600 GOSUB 2124
 3610 CLEAR : GOTO 140

Beschreibung: Zeile 3020 : Der COMTAC-Fehler, der COMPAX-Fehler und der STOP-Interrupt werden gesperrt, um in diesem Unterprogramm keinen dieser Fehler mehr zu erhalten.
 Zeile 3030: Die Funktionstasten werden gesperrt.
 Zeile 3050: Die Fehlerzeile wird in die Variable FZ eingelesen
 Zeile 3060: Die Fehlernummer wird in die Variable FE eingelesen
 Zeile 3520-3540: Die Fehlerzeile, die Fehlernummer und der Fehlertext werden angezeigt
 Zeile 3550: Textanzeige, daß es mit ENTER weiter geht
 Zeile 3560-3570: Warten bis Keyboard-Code 0 wird (keine Tasten mehr gedrückt)
 Zeile 3580-3590: Warten bis die Taste ENTER gedrückt wird
 Zeile 3600: Zeile 2-4 wird gelöscht
 Zeile 3610: Die Variablen, Strings und Interrupts werden gelöscht. Sprung zum Programm-anfang.

Beispiel: ONERR GOSUB 3000 : ENABLE ONERR

Zeile: 63160

12.1.9 AO COMPAX-Fehler-Routine

Funktion: Alle COMPAX-Fehler (auch NOT AUS) werden über den Feldbus ans COMPAX gemeldet. Im COMPAX wird ein Interrupt generiert. Dazu muß das Sprungziel und die Freigabe programmiert werden (ONCPXERR GOSUB 4000 : DISABLE ONCPXERR). Die Fehler der Achsen werden angezeigt und können quittiert werden. Diese Funktion ist sehr wichtig, da das COMTAC auf jeden Fall sofort auf NOT AUS und andere wichtige COMPAX-Ereignisse reagieren muß.

Variablen: A = Achsnummer
 AAA = Aktuelle Achsnummer
 FA = Fehlerachsnummer
 PN = Parameternummer
 NA = Notaus 0/1
 SN = Stringnummer
 AB = Automatikbetrieb
 STM = Stopmarker

Strings: \$(4) = Dummystring
 \$(63) = vorbelegter String
 \$(67) = vorbelegter String
 \$(71) = vorbelegter String
 \$(72) = vorbelegter String

Felder: FA(A) = Die Achsnummer der Fehlerachsen werden in das jeweilige Feld geschrieben
 FE(A) = Die Fehlernummer der Fehlerachse wird in das jeweilige Feld geschrieben

```
Code: 4000 ! AO= ONCPXERROR-ROUTINEN =====
      4010 !
      4020 STOP ONCPXERR : DISABLE ONINP 2 : DISABLE ONKEY SP : GOSUB 50000
      4021 AAA=A
      4030 FOR A=1 TO AA: DISPCPXZ A,0,0,3 : FA(A)=0 : NEXT A
      4040 DO : ! Fehlerachsen pollen|polling axis
      4050 FA= CPXERRADR : IF FA<=AA THEN FA(FA)=FA
      4060 UNTIL FA=0
      4070 ! --- Fehlernummern aus den Achsen auslesen|read errors from the axis -----
      -----
      4080 FOR A=1 TO AA:FE(A)=0
      4090 PN=30 : IF FA(A)<>0 THEN GOSUB 50450 : FE(A)=PW
      4100 NEXT A
      4110 !
      4120 GOSUB 2124
```

```

4130 $(4)=$(72)[11,20],$(72)[31,40] : SN=4 : GOSUB 2600
4140 $(4)=$(71)[21,30] : SN=4 : GOSUB 2220
4150 $(4)=$(71)[1,10] : SN=4 : GOSUB 2230
4160 FOR A=1 TO AA
4170 DISP TABXY(10+(A-1)*3,2),FA(A)
4180 DISP TABXY(10+(A-1)*3,3),FE(A)
4190 NEXT A
4200 !
4210 NA=FALSE : FOR A=1 TO AA
4220 IF FE(A)=55 THEN NA=TRUE
4230 NEXT A
4240 IF NA=TRUE THEN GOTO "NOTAUS" : !-> NOTAUS|EMERGENCY
4250 !
4260 OUTPUT 1,255;"BK"
4270 FOR A=1 TO AA
4280 IF FE(A)=0 THEN GOTO 4370
4290 IF FE(A)=36'OR'FE(A)=37 THEN SN=77 : GOSUB 2240 : DO : WHILE (1)
4300 IF FE(A)>=42'AND'FE(A)<=48 THEN SN=77 : GOSUB 2240 : DO : WHILE (1)
4310 SN=56 : GOSUB 2240
4320 DO : !
4330 WHILE KBDCODE
4340 DO : KC= KBDCODE : ! Warten auf ENTER|wait for ENTER
4350 WHILE KC<>ER:KC=0
4360 OUTPUT 1,FA(A);"QT" : WAIT 500
4370 NEXT A : A=AAA
4380 IF AB THEN GOTO 4470
4390 !
4400 CLEAR ONCPXERR : ENABLE ONCPXERR : CLEAR S : GOTO 10000
4410 !
4420 ! Weiter, wenn Fehler aus Automatik|continue, if fault from automatic
4430 ! Wenn Fehler nicht aus STOP, dann mit GOTO ins STOP-UP
4440 ! If fault not from STOP-routine then GOTO to STOP-routine
4450 ! ansonsten mit RETI zurueck ins STOP-UP
4460 ! If fault from STOP then RETI
4470 IF STM=FALSE THEN CONT ONCPXERR : GOTO "STOP"
4480 SN=63 : GOSUB 2600
4490 SN=67 : GOSUB 2240
4500 CONT ONCPXERR : RETI
4510 !

```

- Beschreibung:
- Zeile 4020: Bei Auftreten eines COMPAX-Fehlers wird in dieses Unterprogramm gesprungen. Der STOP-Interrupt wird gesperrt und der eigene Interrupt wird gestoppt. Treten weitere COMPAX-Fehler während des Abarbeitens dieses UP auf, so werden diese gespeichert und anschließend behandelt.
 - Zeile 4030: Die zyklische Anzeige eines COMPAX-Statuswertes wird ausgeschaltet, das Feld FA(A) wird zurückgesetzt.
 - Zeile 4040-4060: Die Fehleradressen werden solange ausgelesen, bis alle Achsen ihr Fehler mitgeteilt haben. Dies ist Voraussetzung dafür, daß die Fehlernummer aus der entsprechenden Fehlerachse ausgelesen werden kann.
 - Zeile 4080-4100: Die Fehlernummer wird mittels des Status 30 ausgelesen und in das Feld FE(A) geschrieben.
 - Zeile 4120-4200: Die Fehler der Achsen mit der Achsnummer werden am Display angezeigt.
 - Zeile 4210-4250: Prüfung, ob Fehler 55=NOTAUS ansteht.
 - Zeile 4260: Bei Fehler <> 55 wird ein BREAK=BK an alle COMPAX gesendet.
 - Zeile 4270-4370: Die Fehler werden selektiert. Bei nicht quittierbaren wird ein Auschalten verlangt. Bei Quittierbaren Fehlern wird jede Achse nacheinander quittiert.
 - Zeile 4380: Wenn Notaus aus dem Automatikbetrieb kam, dann gehts mit 4470 weiter
 - Zeile 4400: Display wird mit den aktuellen \$(11)..\$(14) beschrieben. Die Funktionstasten werden freigegeben und der Stack wird gelöscht. Es wird ins Hauptmenü gesprungen.
 - Zeile 4470: Wenn Fehler nicht aus STOP-UP kam, dann wird jetzt ins STOP-UP gesprungen. Dies ist notwendig, um den Ablauf fortzusetzen oder abubrechen. (siehe STOP-UP)
 - Zeile 4480-4500: Es wird ein Text angezeigt (Ablauf abbrechen oder fortsetzen), der COMPAX-Fehler wieder freigegeben und ins STOP-UP zurückgesprungen.

Beispiel: ONCPXERR GOSUB 4000 : DISABLE ONCPXERR

Zeile: 63160

12.1.10 AP NOT AUS-Routine

Funktion: Diese Routine wird von der COMPAX-Fehler-Routine aufgerufen. Sie behandelt die gesamte NOT AUS-Funktion. Diese Funktion ist auf jeden Fall in jeder Applikation zu implementieren, da sonst eventuell Schäden an Mensch und Material auftreten könnten.

Variablen: A = Achsnummer
AAA = Aktuelle Achsnummer
AB = Automatikbetrieb
M0 = Maschinennullmerker

Strings: \$(63) = vorbelegter String
\$(67) = vorbelegter String

Felder: -

Code:

```

NOT AUS:
4690 !
4700 ! AP= NOT AUS|emergency =====
4710 !
4720 SN=64 : GOSUB 2600 : SN=60 : GOSUB 2240
4730 !
4740 DO : !
4750 UNTIL ((IN(1)=1)'OR'( RDY =TRUE)) : ! Steuerung Ein|Power On
4760 WAIT 300
4770 IF ((IN(1)=0)'AND'( RDY =0)) THEN GOTO 4740
4780 OUTPUT 1,255;"QT" : WAIT 500
4790 A=AAA: ! akt. Achse
4800 IF AB'AND'M0=0 THEN GOTO 5220 : ! Abbruchstop bei M0 in Auto|Break
4810 IF AB THEN GOTO 4830
4820 GOSUB 2000 : GOSUB 50100 : CONT ONCPXERR : CLEAR S : GOTO 10000
4830 IF STM=FALSE THEN CONT ONCPXERR : GOTO 5000
4840 SN=63 : GOSUB 2600
4850 SN=67 : GOSUB 2240
4860 CONT ONCPXERR : RETI
4870 !

```

Beschreibung: Zeile 4720: Notaus wird angezeigt.
 Zeile 4740-4750: Es wird auf Steuerung Ein (Eingang 1) gewartet.
 Zeile 4760-4770: Nach einer Wartezeit von 300 msec. wird der Eingang 1 nochmals eingelesen, um ein Flackern des Eingangs mitzubekommen.
 Zeile 4780: Ein QUIT wird an alle Achsen gesendet.
 Zeile 4790: Die aktuelle Achse wird der Variablen A zurückgewiesen
 Zeile 4800: Falls im Automatikbetrieb noch kein M0 angefahren wurde, dann kann der Automatikt nicht fortgesetzt werden.
 Zeile 4820: Alle Betriebsarten außer dem Automatik werden abgebrochen. Die Anzeige mit den \$(11)..\$(14) werden angezeigt, die Funktionstasten werden freigegeben und der COMPAX-Fehler wird wieder freigegeben. Anschließend wird der Stack gelöscht und ins Hauptmenü gesprungen.
 Zeile 4830: Wenn Notaus nicht aus STOP-UP kam, dann wird jetzt ins STOP-UP gesprungen. Dies ist notwendig, um den Ablauf fortzusetzen oder abubrechen. (siehe STOP-UP)
 Zeile 4840-4860: Es wird ein Text angezeigt (Ablauf abbrechen oder fortsetzen), der COMPAX-Fehler wieder freigegeben und ins STOP-UP zurückgesprungen.

Beispiel: If NA=TRUE Then GOTO "NOT AUS"

Zeile: 4240

12.1.11 AQ STOP-Routine

Funktion:	Im Automatikbetrieb kann es notwendig sein, den Ablauf zu unterbrechen. Durch Drücken der STOP-Taste am COMTAC wird ein Interrupt am COMTAC ausgelöst. Dazu muß das Sprungziel und die Interrupt-Freigabe programmiert sein (ONKEY SP GOTO 5000 : ENABLE ONKEY SP). Es gibt anschließend die Möglichkeit den Ablauf fortzusetzen oder abzubrechen.
Variablen:	A = Achsnummer AB = Automatikbetrieb M0 = Maschinennullmerker STM = Stop-Merker I = Laufvariable, Keyboard-Code
Strings:	\$(63) = vorbelegter String \$(67) = vorbelegter String
Felder:	POS(0) = 0 -> Positionierung einer/mehreren Achsen ist beendet POS(1) = 1 -> Positionierung einer/mehreren Achsen wurde gestoppt
Code:	<pre> STOP: 4990 ! 5000 ! AQ= STOP ===== 5010 ! 5020 STM=TRUE : DISABLE ONKEY SP : DISABLE ONINP 2 5030 ! 5040 OUTPUT 1,255;"BK" 5050 FOR I=1 TO AA 5060 DO : ! 5070 UNTIL (WAITPOS I)'OR'(CPXSTS(I)@5) 5080 NEXT I 5090 IF M0=TRUE THEN OUTPUT 1,255;"PR0" 5100 GOSUB 2124 5110 SN=63 : GOSUB 2600 5120 SN=67 : GOSUB 2240 5130 IF AB'AND'M0=0 THEN GOTO 5220 : ! M0-Fahrt im AB->Abbruchstop Break 5140 DO : ! 5150 WHILE KBDCODE 5160 DO : ! warte auf Start-Taste oder Stop-Taste wait of Start or Stop 5170 I= KBDCODE 5180 UNTIL I=ST'OR'I=193 5190 IF I=193 THEN GOTO 5220 : ! Abbruchstop Break 5200 GOSUB 2000 : IF POS(0)<>0 THEN GOSUB 1200 : AB=TRUE 5210 STM=FALSE : ENABLE ONKEY SP : ENABLE ONINP 2 : RETI 5220 ! Abbruchstop Break 5230 STM=FALSE : CLEAR S : GOTO 10000 5240 ! </pre>
Beschreibung:	<p>Zeile 5020: Der Stopmerker STM wird gesetzt. Der eigene STOP-Interrupt wird gesperrt.</p> <p>Zeile 5040: Es wird ein BREAK =BK an alle COMPAX gesendet, um die Achsen zu stoppen.</p> <p>Zeile 5050-5080: Es wird auf Position Erreicht aller Achsen gewartet.</p> <p>Zeile 5090: Falls M0 angefahren wurde, wird eine Relativbewegung mit 0 mm durchgeführt. Dies ist notwendig, um auch wirklich ein Position Erreicht zu erhalten.</p> <p>Zeile 5100-5120: Die Zeilen 2-4 werden gelöscht und es wird ein Text am Display angezeigt. (Ablauf fortsetzen oder abbrechen)</p> <p>Zeile 5130: Falls STOP aus Automatikbetrieb und noch kein Maschinennullpunkt angefahren wurde, dann Ablauf abbrechen; Stack löschen und zurück ins Hauptmenü.</p> <p>Zeile 5140-5150: Warten bis Keyboardcode 0 ist.</p> <p>Zeile 5160-5180: Warten auf Tastendruck START-Taste oder STOP-Taste</p> <p>Zeile 5190: Falls STOP-Taste, dann Ablauf abbrechen; Stack löschen und zurück ins Hauptmenü.</p>

Zeile 5200-5210: Anzeige mit den aktuellen \$(1)..\$(14); Falls Positionierung unterbrochen wurde, dann Positionierung wiederholen; STOP-Interrupt wieder freigeben und zurück in den Automatikablauf.

Zeile 5230: Zurücksetzen der Variablen und Rücksprung

Beispiel: ONKEY SP GOSUB 5000

Zeile: 50210

12.1.12 AR Passwort Freigabe in Achse A

Funktion: Um bestimmte Befehle oder Parameter ans COMPAX zu senden, muß die Zugriffsberechtigung freigegeben werden. Dazu muß der Befehl 'GO320' ans COMPAX gesendet werden. (siehe auch COMPAX-Anleitungen)

Variablen: -

Strings: \$(1)

Felder: -

Code: 50300 ! AR= Passwort Freigabe in Achse A|enable password =====
50310
50320 \$(1)="GO320" : GOSUB 1000 : RETURN
50330 !

Beschreibung: Einer zuvor zugewiesenen Achse A wird der Befehl "GO320" gesendet. Dazu wird das Unterprogramm 1000 verwendet.

Beispiel: FOR A=1 TO AA : GOSUB 50300 : NEXT A

Zeile: 63520

12.1.13 AS Passwort Sperren in Achse A

Funktion: Um die Zugriffsberechtigung fürs COMPAX zu sperren, muß der Befehl 'GO270' ans COMPAX gesendet werden. (siehe auch COMPAX-Anleitungen)

Variablen: -

Strings: \$(1)

Felder: -

Code: 50350 ! AS= Passwort Sperren in Achse A|disable password =====
50360
50370 \$(1)="GO270" : GOSUB 1000 : RETURN
50380 !

Beschreibung: Einer zuvor zugewiesenen Achse A wird der Befehl "GO270" gesendet. Dazu wird das Unterprogramm 1000 verwendet.

Beispiel: -

12.1.14 AT CPX-Parameter auslesen

Funktion:	Um Werte von COMPAX-Parametern auszulesen, muß z.B. der Befehl 'P1' ans COMPAX gesendet werden. Das COMPAX wird dann veranlaßt den Wert z.B. 'P1=0.00' ans COMTAC zurückzusenden. Diese Funktion wird gebraucht, um z.B. die Endgrenzen der Achsen einzulesen. Sie dienen als Ober-/Unter-grenze der Positionseingaben.
Variablen:	PN = Parameternummer PW = Parameterwert
Strings:	\$(1) = Sendestring \$(#1) = Rückantwort
Felder:	-
Code:	50400 ! AT= CPX-Parameter auslesen aus Achse A read Compax parameter =====
	50410 ! 50420 \$(1)="P",VAL\$(PN) : GOSUB 1100 : PW=VAL\$(#1)[6] 50430 RETURN 50440 !
Beschreibung:	Einer zuvor zugewiesenen Achse A wird der Befehl "Px" gesendet. Als Rückantwort steht im String \$(#1) der Wert des angewählten Parameters. Dieser Wert wird in die Variable PW als Zahl gewandelt. Die Einheit wird dabei nicht betrachtet.
Beispiel:	A=1 : PN=1 : GOSUB 50400 : RN(A)=PW Zeile: 43820 - 43840

12.1.15 AU CPX-Status auslesen

Funktion:	Um Statuswerte vom COMPAX auszulesen, muß z.B. der Befehl 'S1' ans COMPAX gesendet werden. Das COMPAX wird dann veranlaßt den Wert z.B. 'S1=123.00' ans COMTAC zurückzusenden. Diese Funktion wird gebraucht, um z.B. die aktuellen Position der Achsen einzulesen. Sie sind die Teachpositionen für den Parametrierbetrieb. Ebenso für das Abfragen der Fehlernummer im COMPAX.
Variablen:	PN = Parameternummer PW = Parameterwert
Strings:	\$(1) = Sendestring \$(#1) = Rückantwort
Felder:	-
Code:	50450 ! AU= CPX-Status auslesen aus Achse A read Compax state 50460 ! 50470 \$(1)="S",VAL\$(PN) : GOSUB 1100 : PW=VAL\$(#1)[7] 50480 RETURN 50490 !
Beschreibung:	Einer zuvor zugewiesenen Achse A wird der Befehl "Sx" gesendet. Als Rückantwort steht im String \$(#1) der Wert des angewählten Status. Dieser Wert wird in die Variable PW als Zahl gewandelt. Die Einheit wird dabei nicht betrachtet.
Beispiel:	FOR A=1 TO AA : FE(A)=0 PN=30 : GOSUB 50450 : FE(A)=PW

NEXT A

Zeile: 4080 - 4100

12.1.16 AV CPX-Parameter beschreiben

Funktion: Um Parameter vom COMTAC aus über den Feldbus ans COMPAX zu beschreiben, muß z.B. der Befehl 'P2=10' gesendet werden. Zuvor muß jedoch die Zugriffsberechtigung 'GO320' ans COMPAX geschrieben werden.

Variablen: PN = Parameternummer
PW = Parameterwert

Strings: \$(1) = Sendestring

Felder: -

Code: 50500 ! AV= CPX-Parameter beschreiben in Achse A|write Compax parameter
=====

```
50510 !
50520 $(1)="P",VAL$(PN),"=",VAL$(PW) : GOSUB 1000
50530 RETURN
50540 !
```

Beschreibung: Einer zuvor zugewiesenen Achse A wird der Befehl "P_{PN}=PW" gesendet.

Beispiel: A=1 : PN=1 : PW=0 : GOSUB 50500

Zeile: 43230-43250

12.1.17 AW Fehleranzeige und weiter mit ENTER

Funktion: Beim Auftreten eines Fehlers wird durch Zuweisung eines Strings dieser Fehler angezeigt. Durch Drücken von ENTER wird diese Fehlermeldung bestätigt und aus dieser Funktion zurückgesprungen. Gleichzeitig leuchtet die LED 2 während der Fehleranzeige auf.

Variablen: ER = Code der ENTER-Taste

Strings: \$(14) = Anzeigestring der Zeile 4 am Display
\$(SN) = Stringnummer der Fehlermeldung

Felder: -

Code: 50600 ! AW= Fehleranzeige und weiter mit ENTER|display fault, cont. with ENTER
=====

```
50610
50620 $(14)=$(SN)," --> ENTER" : GOSUB 2060 : SETLED 2
50630 DO : !
50640 WHILE KBDCODE
50650 DO : !
50660 UNTIL KBDCODE =ER
50670 $(14)="" : GOSUB 2060 : CLRLED 2 : RETURN
50680 !
```

Beschreibung: Zeile 50620: Der String \$(SN) wird dem String \$(14) zugewiesen und angezeigt. Die Leuchtdiode 2 leuchtet auf.
Zeile 50630-50640: Warten, bis alle Tasten losgelassen wurden.
Zeile 50650-50660: Warten, bis Taste ENTER gedrückt wird.

Zeile 50670: Die Zeile 4 und die LED 2 werden gelöscht.

Beispiel: SN=81 : GOSUB 50600 : GOTO 20310

Zeile: 20340

12.1.18 AX Dateibefehle

Funktion: In dieser Funktion werden Dateibefehle ausgeführt. Bei angeschlossenem Diskettenlaufwerk können außer dem ZP-RAM R auch Laufwerk A und B angesprochen werden. Es gibt Befehle zum kopieren, laden, löschen, formatieren und umbenennen.

Variablen: DL = Datenlaufwerk
DB = Dateibefehl

Strings: \$(5) = Dateiname

Felder: -

Code:

```

51000 ! AX= Dateibefehle|Filecommands =====
51010
51020 ON DL GOTO 51100,51200,51300
51030 !
51100 ON DB GOTO 51110,51120,51130,51140,51150,51160,51170 : ! Zp Ram
51105 !
51110 LOAD R:$(5) : GOTO 51400
51120 STORE R:$(5) : GOTO 51400
51130 DEL R:$(5) : GOTO 51400
51140 FORMAT R: GOTO 51400
51150 COPY R:$(5) TO A:$(5) : GOTO 51400
51160 COPY R:$(5) TO B:$(5) : GOTO 51400
51170 RENAME R:$(5) TO R:$(5) : GOTO 51400

```

Beschreibung: Zeile 51020: Abhängig vom Datenlaufwerk R=0, A=1, B=2 wird verzweigt.
Zeile 51100: Abhängig vom Dateibefehl LOAD=0, STORE=1, DEL=2, FORMAT=3, COPY R->A=4, COPY R->B=5, RENAME=6 wird verzweigt.
Zeile 51110-51170: Die angewählten Befehle werden ausgeführt. Der Dateiname steht im \$(5).

Beispiel: Siehe Zeile 330

12.1.19 AZ Achseingabe

Funktion: In dieser Funktion wird die Achsnummer eingegeben und in die Variable A gespeichert.

Variablen: A = Achsnummer
KK = Zwischenspeicher
UG = Untergrenze
OG = Obergrenze
SA = Stellenanzahl
AA = max. Achsanzahl

Strings: \$(3) = Eingabestring
\$(47) = Anzeigestring

Felder: -

Code:	<pre> 52100 ! AZ= AchsEingabe (AchsNummer) input number of the axis 52110 ! 52120 KK=A:UG=1 : OG=AA:SA=1 : \$(3)=\$(47) : GOSUB 52000 : A=KK 52130 RETURN </pre>
Beschreibung:	Zeile 52120: Der Wert der akt. Achsnummer, die Unter-/Obergrenze, die Stellenanzahl und der Text werden zugewiesen. Es wird in das Eingabeprogramm verzweigt. Die eingegebene Nummer wird der Achse A zugewiesen.
Beispiel:	-

12.1.20 BA Weiter mit ENTER

Funktion:	Diese Funktion wird gebraucht, um z.B. Meldungen am COMTAC mit ENTER zu quittieren. Dies können Fehlermeldungen, Bestätigungen, ... sein.
Variablen:	SN = Stringnummer ER = Konstante für ENTER-Taste
Strings:	\$(14) = Anzeigestring für Zeile 4 des Displays
Felder:	-
Code:	<pre> 52700 ! BA= Weiter mit RETURN continue with ENTER ===== 52710 ! 52720 SN=56 : GOSUB 2240 : \$(14)=\$(SN) 52730 DO : ! 52740 UNTIL KBDCODE =ER 52750 DO : ! 52760 WHILE KBDCODE 52770 CLEAR D,4 52780 RETURN </pre>
Beschreibung:	Zeile 52720: Die Stringnummer 56 wird in Zeile 4 des Displays angezeigt. Zur Speicherung wird der String in den Anzeigestring 14 geschrieben. Zeile 52730-52740: Es wird auf den Keyboard-Code der ENTER-Taste gewartet. Zeile 52750: Nach dem Drücken von ENTER wird gewartet bis der Keyboard-Code wieder 0 wird, d.h. die Tasten losgelassen wurden. Zeile 52770: Die Zeile 4 wird gelöscht. Zeile 52780: Rücksprung
Beispiel:	<pre> CLEAR D,3 DISP TABXY(1,3),"Info: COMTAC initialisiert" GOSUB 52700 GOTO 10000 Zeile: 44350 </pre>

12.2 C Hauptprogramm und Initialisierung

12.2.1 CA Hauptprogramm

Funktion:	Der Kern des Programmes stellt das Hauptprogramm dar. Hier werden die verschiedenen Unterprogramme für das Initialisieren und der Grundeinstellungen aufgerufen. Es ist der erste Programmteil der durchlaufen wird. Vom Hauptprogramm aus wird das Hauptmenü aufgerufen.
Variablen:	I = Schleifenzähler AA = max. Achsanzahl A1 = Achse 1 = 1 A2 = Achse 2 = 2 A3 = Achse 3 = 3 A4 = Achse 4 = 4
Strings:	-
Felder:	AS(..) = Stringfeld zum Ausgangsbezeichnungen anzuzeigen
Code:	<pre> 63000 ! CA= HAUPT-PROGRAMM MAIN ===== 63010 ! 63020 !Initialisierung initialize 63030 ! 63040 GOSUB 2700 : ! DISP-Init 63050 DIM AS(96) : ! Zum Ausgaenge anzeigen (CTC+max 4 CPX) display outputs 63060 GOSUB 60000 : ! String-Dimensionierung string dimensioning 63070 GOSUB 61000 : ! String-Zuweisungen string assingment 63080 SN=61 : GOSUB 2600 63090 DISP TABXY(1,4),"V",USING(##),PV," / ",USING(#####),PD,USING(0) 63100 GOSUB 62000 : ! allg. Variablen-Initialisierung global variable init 63110 GOSUB 50200 : ! ONKEY/ONINP-Initialisierung ONKEY/ONINP init 63115 IF CONTROL(136)=2 THEN GOTO 49000 : ! Anzeigesteuerung display control 63120 ! 63130 ! Interrupt-Zuweisung assign interrupts 63140 ! 63150 ONERR GOSUB 3000 : ENABLE ONERR : ! Priority =9 63160 ONCPXERR GOSUB 4000 : DISABLE ONCPXERR : PRIORITY ONCPXERR =8 63170 ONINP 2 GOSUB 5000 : DISABLE ONINP 2 : ! PRIORITY ONINP (2)=1 63180 GOSUB 50000 63190 ! 63200 ! 63210 RESET #1 : WAIT 400 63220 ! 63230 ! Ausgaenge,Eingaenge maskieren und Zukl. Anzeigen abschalten 63240 ! mask outputs,inputs and disable display 63250 CPXOMASK(255)=0FFFFH : ! Alle 16 benutzen 63260 CPXIMASK(255)=0FFFFH : ! Alle 16 benutzen 63270 FOR A=1 TO AA: DISPCPXZ A,0,0,3 : NEXT A 63280 CLRLED 1 : CLRLED 2 : CLRLED 3 : CLRLED 4 63290 FOR AS=1 TO MA : GOSUB 1700 : NEXT AS 63300 FOR AS=33 TO 48 : GOSUB 1700 : NEXT AS : IF AA=1 THEN GOTO 63340 63310 FOR AS=49 TO 64 : GOSUB 1700 : NEXT AS : IF AA=2 THEN GOTO 63340 63320 FOR AS=65 TO 80 : GOSUB 1700 : NEXT AS : IF AA=3 THEN GOTO 63340 63330 FOR AS=81 TO 96 : GOSUB 1700 : NEXT AS : IF AA=4 THEN GOTO 63340 63340 ! 63350 M0=TRUE : FOR I=1 TO AA:M0=(M0'AND'CPXSTS(I)@2) : NEXT I 63360 ! 63370 GOSUB 30500 : ! Parameter und Positionen laden/load parameters 63380 GOSUB 43800 : ! Endgrenzen der Achsen ermitteln/load limits 63390 GOSUB 26800 : ! Parameter lesen load parameters 63400 ENABLE ONKEY 14: ENABLE ONCPXERR 63410 ! Warten auf Steuerung Ein wait for Power On </pre>

```

63420 SN=60 : GOSUB 2240
63430 EG=1 : DO : GOSUB 1800
63440 UNTIL ((EGS<>FALSE)'OR'( RDY <>FALSE)) : CLEAR D,4 : WAIT 300
63450 EG=1 : GOSUB 1800 : IF (EGS=FALSE)'AND'( RDY =FALSE) THEN GOTO 63420
63460 OUTPUT 1,255;"QT" : WAIT 500
63470 DISABLE ONKEY 19
63480 !
63490 !
63500 !
63510 ! CPX-Passworte freigeben|enable passwords
63520 FOR A=1 TO AA: GOSUB 50300 : NEXT A
63530 !
63540 OUTPUT 1,255;"N001:END"
63550 !
63560 GOSUB 2124 : ! Zeile 2 - 4
63570 !
63580 ENABLE ONCPXERR : GOTO 10000 : ! Hauptmenue|mainmenu
63590 !

```

Beschreibung:

Zeile 63040: Das Display wird initialisiert

Zeile 63050: Das Feld AS() wird dimensioniert. Es wird benötigt, um die Ausgänge in der Betriebsart Test E/A's anzusehen. (80 = 'CTC' 1*16 + 'CPX' 4*16)

Zeile 63060: Die benötigten Strings werden dimensioniert.

Zeile 63070: Den Strings werden Texte zugewiesen.

Zeile 63080: Der Projektname wird in Zeile 1 zentriert dargestellt.

Zeile 63090: Die Programmversion, -datum werden in Zeile 4 dargestellt.

Zeile 63100: Die Variablen werden initialisiert und Grundwerte zugewiesen.

Zeile 63110: Die Sprunganweisungen der Funktionstasten und Interrupte werden zugewiesen.

Zeile 63150: Der COMTAC-Fehler wird initialisiert und freigegeben.

Zeile 63160: Der COMPAX-Fehler wird initialisiert und gesperrt, die Funktionstasten werden gesperrt.

Zeile 63170: Der STOP-Eingang 2 wird initialisiert und gesperrt.

Zeile 63180: Die Funktionstasten werden gesperrt.

Zeile 63210: Der Feldbus wird zurückgesetzt.

Zeile 63250-63260: Die Ausgänge und Eingänge werden maskiert, so daß die Ausgänge 1-6 und Eingänge 1-6 verwendet werden können.

Zeile 63270: Die zyklische Anzeige vom COMPAX am COMTAC-Display wird ausgeschaltet.

Zeile 63280: Die 4 Leuchtdioden werden gelöscht.

Zeile 63290-63330: Alle Ausgänge vom COMTAC und den COMPAX werden gelöscht.

Zeile 63350: Es wird gepüft, ob der Maschinen-Nullpunkt aller Achsen schon angefahren wurde.

Zeile 63370: Die Maschinenparameter und die Maschinenpositionen werden aus den Dateien geladen.

Zeile 63380: Die Endgrenzen der Achsen werden ermittelt. Sie dienen als Unter-/Obergrenze für Eingaben.

Zeile 63390: Parameter der Achsen werden gelesen, um die Absolutgeschwindigkeit im m/s zu bestimmen.

Zeile 63420-63450: Es wird der Eingang 1 des COMTAC abgefragt, der 'STEUERUNG EIN' mitteilt.

Zeile 63460: Es wird ein Quit an alle COMPAX gesendet, um den Fehler E55 'NOTAUS<->STEUERUNG EIN' zu quittieren. Die Achsen gehen anschließend in den RUN-Zustand.

Zeile 63520: Die Zugriffsberechtigung der COMPAX-Achsen wird freigegeben.

Zeile 63540: Es wird ein 'END' in den Satz 1 an alle COMPAX-Achsen gesendet, um ein im Satzspeicher der COMPAX-Achsen stehendes Programm nicht starten zu können.

Zeile 63560: Die Zeilen 2-4 des Displays werden gelöscht.

Zeile 63580: Der COMPAX-Fehler wird freigegeben und es wird ins Hauptmenü verzweigt.

12.2.2 CB String-Dimensionierung

Funktion:	Alle Strings, die länger als 16 Zeichen haben, sind zu dimensionieren. Grundsätzlich sollten aus Übersichtsgründen alle Strings dimensioniert werden.
Variablen:	I= Schleifenvariable
Strings:	\$ (0)-\$ (19) = vorbelegte Strings \$ (20)-\$ (112) = Texte aus den Dateien
Felder:	-
Code:	<pre> 60000 !=== String-Dimensionierung === 60010 ! 60020 FOR I=0 TO 10 : DIM \$(I)[80] : NEXT I 60030 FOR I=11 TO 255 : DIM \$(I)[40] : NEXT I 60040 ! \$(1)..\$(19) vorbelegt 60050 \$(17)="CTCT1S",VAL\$(CONTROL(33)) 60060 FOR I=20 TO 112 : \$(I)=\$(\$ (17)(I-19)[1,,92] : NEXT I 60070 ! \$(113)..\$(255) leer, zur freien Verfügung 60080 RETURN 60090 ! </pre>
Beschreibung:	<p>Zeile 60020: Die \$ (0)-\$ (10) werden mit einer Länge von 80 Zeichen dimensioniert.</p> <p>Zeile 60030: Die \$ (11)-\$ (255) werden mit einer Länge von 40 Zeichen dimensioniert.</p> <p>Zeile 60050: Um die Texte sprachabhängig aus der Datei "CTCT1S_" zuzuweisen, wird in den \$ (17) der Dateiname geschrieben. Die Sprache steht im Controlwort 101 des COMTAC.</p> <p>Zeile 60060: Die Texte der \$ (20)-\$ (112) werden aus der Datei "CTCT1S_" zugewiesen. _ steht für die Sprache: 0=Deutsch, 1=Englisch. Jedem String wird dabei ein Text aus der Datei UMEM\$ "CTCT1S_" zugewiesen. Um die Länge der Strings variabel zu halten, enden alle Texte mit dem Zeichen "\"=ASCII-Code 92.</p>
Beispiel:	-

12.2.3 CC String-Initialisierung

Funktion:	Nachdem die Strings dimensioniert wurden, werden sie mit Texten beschrieben. Die String \$ (113) bis \$ (250) können frei verwendet werden.
Variablen:	-
Strings:	\$ (0)-\$ (112)
Felder:	-
Code:	<pre> 61000 ! CC= String-Initialisierung ===== 61010 ! 61020 \$(#0)=" " : ! RS232/1-ENTER-String 61030 \$(1)=" " : ! RS485-OUTPUT-String 61040 \$(#1)=" " : ! RS485/1-ENTER-String 61050 \$(2)=" " : ! HOST-Sendestring transmit string 61060 \$(0)=" " : ! HOST-Empfangsstring receive string 61070 \$(#2)=" " : ! RS232/2-ENTER-String 61080 \$(3)=" " : ! Input String 61090 \$(4)=" " : ! Dummy zum zerstückeln von Strings dummy string 61100 \$(5)=" " : ! Dateiname String filename string 61110 \$(6)=" " : ! Empfangsstring beim Senden an HOST receive string 61120 \$(7)=" " : ! Dummy-String dummy string 61130 \$(8)=" " : ! reservierter String reserved string </pre>

```

61140 $(9)="" : ! reservierter String|reserved string
61150 $(10)="" : ! reservierter String
61160 !
61170 $(11)="" : ! Display Zeile 1|display line 1
61180 $(12)="" : ! Display Zeile 2|display line 2
61190 $(13)="" : ! Display Zeile 3|display line 3
61200 $(14)="" : ! Display Zeile 4|display line 4
61210 $(15)="AXP" : ! Datei mit Achs-Parametern|file of the axis-parameter
61220 $(16)="MAP" : ! Datei mit Maschinen-Parametern|file of the machine-param
61230 !$(17)="CTCT1S",VAL$(CONTROL(101)) :! in 60000 ( standard text )
61240 $(18)="PART1S",VAL$(CONTROL(101)) : ! textfile for parameters
61250 $(19)="OUTT1S",VAL$(CONTROL(101)) : ! textfile for outputs
61260 ! ab $(20)..$(112) Standard-Texte aus CTCT1S1/2 : ! standard text
61270 RETURN

```

Beschreibung: Zeile 61000-61150: Die \$(0)-\$(10) werden hier vorbelegt.
 Zeile 61160-61200: Die \$(11)-\$(14) werden zur Darstellung der Zeile 1-4 des Displays verwendet.
 Zeile 61210: Der Dateiname der Achsparameter wird in \$(15) zugewiesen.
 Zeile 61220: Der Dateiname der Maschinenparameter und Maschinenpositionen wird in \$(16) zugewiesen.
 Zeile 61230: Der Dateiname der Standardtexte wird in \$(17) zugewiesen. Dies wird programmtechnisch jedoch im UP Stringdimensionierung vorgenommen.
 Zeile 61240: Der Dateiname der Texte der Achsparameter, Maschinenparameter und Maschinenpositionen wird in \$(18) zugewiesen.
 Zeile 61250: Der Dateiname der Texte der Achsparameter wird in \$(19) zugewiesen.
 Zeile 61270: Rücksprung

Beispiel: -

12.2.4 CD Variablen-Initialisierung

Funktion: Alle Variablen sollten aus Übersichtsgründen an dieser zentralen Stelle aufgeführt und beschrieben werden. Da es im COMTAC-Basic keine generelle Unterscheidung von Variablen und Konstanten gibt, ist es besonders bei den "Konstanten" wichtig diese zentral zu definieren.

Variablen: siehe Programm-Code (ab Zeile 62000)

Strings: -

Felder: siehe ab Seite 51.

Code: siehe Programm-Code

Beschreibung: Allen Variablen werden Anfangswerte zugewiesen. Die genaue Beschreibung der einzelnen Variablen ist im Programm-Code zu sehen.

Beispiel: A1 = 1 : ! Achse 1

12.2.5 CE Funktionstasten sperren

Funktion: Um in einer Betriebsart, wie z.B. dem Automatikbetrieb, das Auslösen der Funktionstasten zu verhindern, müssen diese gesperrt werden.

Variablen: -

Strings:	-
Felder:	-
Code:	<pre> 50000 ! CE= Funktionstasten sperren disable onkeys ===== 50010 ! 50020 DISABLE ONKEY 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 50030 ! ONKEY 8 GOSUB 6000 : ENABLE ONKEY 8 50040 RETURN 50050 ! </pre>
Beschreibung:	<p>Alle 16 Funktionstasten werden gesperrt.</p> <p>Das Sperren der Funktionstasten ist notwendig, wenn eine Betriebsart aufgerufen wurde und diese nicht unterbrochen werden soll.</p>
Beispiel:	<p>GOSUB 50000</p> <p>Zeile: 20030</p>

12.2.6 CF Funktionstasten freigeben

Funktion:	Um im Hauptmenü das Auslösen der Funktionstasten zu gewährleisten, müssen diese freigegeben sein. Zuvor muß ein Sprungziel (siehe Funktionstasten zuweisen) definiert werden.
Variablen:	-
Strings:	-
Felder:	-
Code:	<pre> 50100 ! CF= Funktionstasten freigeben enable onkeys ===== 50110 ! 50120 ! ONKEY 8 GOSUB 6000 50130 ENABLE ONKEY 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 50140 RETURN 50150 ! </pre>
Beschreibung:	<p>Alle 16 Funktionstasten werden freigegeben. Zuvor muß jedoch die Zuweisung der Sprungziele vorgenommen werden. Die Freigabe der Funktionstasten ist im Hauptmenü sinnvoll, wenn die verschiedenen Betriebsarten aufgerufen werden können.</p>
Beispiel:	<p>GOSUB 50100</p> <p>Zeile: 10060</p>

12.2.7 CG Funktionstasten zuweisen

Funktion:	Vom Hauptmenü aus können verschiedene Betriebsarten durch Drücken einer Funktionstaste aufgerufen werden. Diese Sprungziele der Funktionstasten müssen einmal definiert werden.
Variablen:	SP = STOP-Taste
Strings:	-
Felder:	-

Code: 50200 ! CG= Funktionstasten Zuweisen|onkey assignment =====
50205 !
50210 ONKEY SP GOSUB 5000 : ! Stop-Taste
50215 ONKEY F1 GOSUB 6000 : ! F1
50220 ONKEY F2 GOSUB 6000 : ! F2
50225 ONKEY F3 GOSUB 6000 : ! F3
50230 ONKEY F4 GOSUB 6000 : ! F4
50235 ONKEY F5 GOTO 20000 : ! F5 Automatik
50240 ONKEY F6 GOTO 30000 : ! F6 Parametrierbetrieb
50245 ONKEY F7 GOTO 40000 : ! F7 Handbetrieb
50250 ONKEY 8 GOTO 47000 : ! F8 Einrichtbetrieb
50255 ONKEY F9 GOTO 41000 : ! F9 M0 manuell
50260 ONKEY F10 GOTO 42000 : ! F10 Test Ein-/Ausgänge
50265 ONKEY F11 GOTO 43000 : ! F11 Korrektur Maßbezug
50270 ONKEY F12 GOSUB 6000 : ! F12
50275 ONKEY F13 GOTO 44000 : ! F13 CTC-Initialisierung
50280 ONKEY F14 GOTO 45000 : ! F14 CPX-Parameter-Manager
50285 ONKEY F15 GOTO 48000 : ! F15 Programm-Ende
50290 ONKEY F16 GOTO 46000 : ! F16 Anlageninfo
50295 !
50296 RETURN

Beschreibung: Die Sprungziele aller 16 Funktionstasten werden zugewiesen. Dabei sind GOSUB- und GOTO-Sprünge möglich. Die Betriebsarten werden mit GOTO angesprungen. Nicht belegte Funktionstasten wird das Pseudoziel 6000 zugewiesen. Von dort aus wird ohne Ausführen einer Funktion zurückgesprungen.

Beispiel: GOSUB 50200

Zeile: 63110

12.3 D Betriebsarten

12.3.1 D0 Grundsätzliches

Dokumentation: Es kann im folgenden nicht Programmzeile für Programmzeile beschrieben werden. Dies würde den Rahmen sprengen. Es wird jedoch versucht wichtige Programmteile hervorzuheben und zu erläutern.

Aufbau: Die Softwarestruktur ist derart gegliedert, daß nach einer Initialisierungsphase gewartet wird, bis 'STEUERUNG EIN' gedrückt wird. Danach wird in das Hauptmenü gesprungen, aus dem die verschiedenen Betriebsarten aufgerufen werden können. Jede Betriebsart ist durch ein Passwort geschützt. Beim 1. Start des Programmes können folgende Einstellungen programmiert werden.

CT.(130) = 0=Deutsch/1=Englisch
 CT.(131) = 1..4 Achsen
 CT.(132) = Anzahl Maschinenpositionen (1..31)
 CT.(133) = Anzahl Bearbeitungsprogramme (1..(600/CT.(102)/CT.(103))
 CT.(134) = Anzahl Maschinenparameter (1..44)
 CT.(135) = Anzahl Achsparameter (1.. 25/Achse)
 CT.(136) = COMTAC-Automatik=0 / SPS-Automatik=1
 CT.(137) = Passwort der Ebene 0
 CT.(138) = Passwort der Ebene 1
 CT.(139) = Passwort der Ebene 2

12.3.2 DA Hauptmenü

Funktion: Alle Betriebsarten werden vom Hauptmenü aus aufgerufen. Nach dem Beenden der Betriebsarten wird wieder ins Hauptmenü zurückgesprungen.

Variablen: AB = Merker für Automatikbetrieb
 A = Achsnummer
 AA = max. Achsanzahl

Strings: \$(11)-\$(14) = Display-Zeilen 1-4
 \$(21),\$(20),\$(10) = Anzeigestrings

Felder: -

Code:

```

10000 ! DA= HAUPTMENUE|MAINMENUE =====
10010 !
10020 AB=FALSE : CONT ONCPXERR : DISABLE ONINP 2 : DISABLE ONKEY SP
10030 GOSUB 2700 : ! Display initialisieren\initialize display
10040 FOR A=1 TO AA: DISPCPXZ A,0,0,3 : NEXT A
10050 $(11)=$(21) : $(14)=$(20) : $(12)=$(10) : $(13)=$(10) : GOSUB 2000
10060 GOSUB 50100 : IF CONTROL(107)=1 THEN AS=7 : GOSUB 1700
10070 !
10080 ! Warten Funktionstaste|wait for functionkey pressed
10090 IF CONTROL(107)=0 THEN GOTO 10080 ELSE DISABLE ONKEY 5
10100 M0=TRUE : FOR I=1 TO AA:M0=(M0'AND'(CPXSTS(I)@2)) : NEXT I
10110 AS=8 : IF M0=TRUE THEN GOSUB 1600 ELSE GOSUB 1700 : ! CTC-Ready
10120 AS=6 : GOSUB 1700 : ! Wartemode|Idlemode
10130 EG=3 : GOSUB 1800 : IF EGS<>FALSE THEN GOTO 25000 : ! Automatik
10140 EG=5 : GOSUB 1800 : IF EGS<>FALSE THEN GOTO 27000 : ! Handverfahren
10150 EG=6 : GOSUB 1800 : IF EGS<>FALSE THEN GOTO 28000 : ! M0 anfahren
10160 GOTO 10080
10170 !
  
```

Beschreibung: Zeile 10020: Der Merker für den Automatikbetrieb wird zurückgesetzt. Der Interrupt für den COMPAX-Error wird freigegeben. Der Interrupt des Eingang 2 wird gesperrt.
 Zeile 10030: Das Display wird initialisiert.
 Zeile 10040: In einer Schleife werden die zyklischen Anzeigen vom COMPAX am COMTAC-Display ausgeschaltet.
 Zeile 10050: Den \$(11)-\$(14) werden Texte zugewiesen und am Display in Zeile 1-4 angezeigt.
 Zeile 10060: Die Funktionstasten werden freigegeben. Bei SPS-Automatik wird Ausgang 7 gelöscht.
 Zeile 10090: Falls CTC-Automatikbetrieb angewählt, dann wird auf den Druck einer Funktionstaste gewartet.
 Zeile 10100: Abfrage, ob die M0-Punkte aller COMPAX-Achsen angefahren wurden.
 Zeile 10110: Ausgang 8 wird in Abhängigkeit der Variable M0 gesetzt oder gelöscht.
 Zeile 10120: Ausgang 6 wird gelöscht.
 Zeile 10130: Falls Eingang 3 gesetzt, dann wird in den SPS-Automatikbetrieb verweigert.
 Zeile 10140: Falls Eingang 5 gesetzt, dann wird in den SPS-Handbetrieb verweigert.
 Zeile 10150: Falls Eingang 6 gesetzt, dann wird in den SPS-M0-anfahren verweigert.
 Zeile 10160: Rücksprung zur Anfangsschleife

Beispiel: -

12.3.2.1 DB [F5] Automatik-Betrieb

Funktion: Der Automatikbetrieb ist das Herzstück jeder Anlage. Hier finden in anderen Betriebsarten eingestellte Parameter Anwendung. Diese Betriebsart kann hier jedoch nur in einer minimalen Form dargestellt werden, da jede Anlage einen spezifischen und auf Ihre Ansprüche definierten Ablauf hat.

Variablen: AB = Merker für den Automatikbetrieb = 1
 M0 = Merker für den Maschinen-Null-Punkt angefahren = 1
 SP = Konstante für die STOP -Taste am COMTAC
 A = Achsnummer
 A1 =1 = Achse 1
 A2 =2 = Achse 2
 A3 =3 = Achse 3
 A4= 4 = Achse 4
 AS = Ausgangsnummer
 EG = Eingangsnummer
 SN = Stringnummer
 SD = Merker, für schnelles Fahren =0 / langsames Fahren = 1
 RP = Variable für Ruheposition = z.B. 0 mm

Strings: \$(11)-\$(14) = Display-Zeilen 1-4
 \$(10), \$(26), \$(80), \$(76), \$(66), \$(65) = Anzeigestrings

Felder: POS(..) = Feld, in dem die Positionsvorgaben im Automatikbetrieb stehen. POS(0)=zu positionierende Achse; POS(zu positionierende Achse)=Positionswert
 PD(..) = Feld, in dem die Werte der Teachpunkte stehen.
 A(..) = Feld, in dem die Binärwerte der Achsen stehen.

Code: 20000 ! DB= AUTOMATIK|automatic mode =====
 20010 !
 20020 AB=TRUE
 20030 GOSUB 50000 : CLRLED 1 : CLRLED 2 : CLRLED 3 : CLRLED 4
 20040 \$(11)=\$(26) : \$(12)=\$(10) : \$(13)=\$(10) : \$(14)="" : GOSUB 2000
 20050 ENABLE ONKEY SP : ENABLE ONINP 2
 20060 GOSUB 1400 : ! Auto-Speed, Accel, POS() reset
 20070 !
 20080 IF M0=TRUE THEN GOTO 20300
 20090 ! Mach-Null abfahren, wenn nötig|Drive to Zero-Point if necessary
 20100 \$(14)=\$(80) : GOSUB 2000

```

20110 DO : !
20120 UNTIL KBDCODE =ST
20130 $(14)=$(76) : GOSUB 2000
20140 A=A2 : $(1)="PH" : GOSUB 1000 : WAIT WT
20150 DO : !
20160 UNTIL (CPXSTS(A)@2)=TRUE
20170 $(14)=$(66) : GOSUB 2000 : $(1)="PA0" : GOSUB 1000 : WAIT WT
20180 DO : !
20190 UNTIL (CPXSTS(A)@4)=TRUE
20200 !
20210 $(14)=$(76) : GOSUB 2000
20220 A=A1 : $(1)="PH" : GOSUB 1000 : WAIT WT
20230 DO : !
20240 UNTIL (CPXSTS(A)@2)=TRUE
20250 $(14)=$(66) : GOSUB 2000 : $(1)="PA0" : GOSUB 1000 : WAIT WT
20260 DO : !
20270 UNTIL (CPXSTS(A)@4)=TRUE
20280 M0=TRUE : FOR I=1 TO AA:M0=(M0'AND'CPXSTS(I)@2) : NEXT I
20290 !
20300 ! .....
20310 !
20320 !AS=2 : GOSUB 1700 : AS=3 : GOSUB 1700 : AS=3 : GOSUB 1500
20330 !EG=3 : GOSUB 1900 : IF EGS<>FALSE THEN GOTO 20360
20340 !SN=81 : GOSUB 50600 : GOTO 20310
20350 !
20355 BP=1 : GOSUB 30900
20360 ! Schnell : Achse 2 : PositionNr. 1|Fast : Axis 2 : PositionNo. 1
20370 SD=0 : A(0)=A(2) : POSN(A(2))=1 : GOSUB "STRECKENPOSITIONIERUNG"
20380 !
20390 ! Schnell : Achse 1 : PositionNr. 1|Fast : Axis 1 : PositionNo. 1
20400 SD=0 : A(0)=A(1) : POSN(A(1))=1 : GOSUB "STRECKENPOSITIONIERUNG"
20410 !
20420 ! Schnell : Achse 1 : PositionNr. 2|Fast : Axis 1 : PositionNo. 2
20430 SD=0 : A(0)=A(1) : POSN(A(1))=2 : GOSUB "STRECKENPOSITIONIERUNG"
20440 !
20450 ! Langsam : Achse 2 : PositionNr. 2|Slow : Axis 2:PositionNo. 2
20460 SD=1 : A(0)=A(2) : POSN(A(2))=2 : GOSUB "STRECKENPOSITIONIERUNG"
20470 !
20480 ! Langsam : Achse 2 : PositionNr. 2|Slow : Axis 2 : PositionNo. 2
20490 SD=1 : A(0)=A(2) : POSN(A(2))=2 : GOSUB "STRECKENPOSITIONIERUNG"
20500 !
20510 ! Achse 1+2 : PositionNr. 3|Slow : Axis 1+2 : PositionNo. 3
20520 A(0)=A(12) : POSN(A1)=3 : POSN(A2)=3 : GOSUB "BAHNPOSITIONIERUNG"
20530 !
20540 GOTO 20350
20550 !
24000 LABEL "STRECKENPOSITIONIERUNG": !
24010 A=A(0) : SPD(0)=A(0) : ACC(0)=A(0) : ! schnell|fast
24020 SPD(A)=AXP(1+SD*2,A) : ACC(A)=AXP(2+SD*2,A) : GOSUB 1300
24030 POS(0)=A(0) : POS(A)=PD(POSN(A),A) : GOSUB 1200
24040 DO : !
24050 $(14)=$(65) : DISP TABXY(31,4),"A",VAL$(A),"=",CPXPOS(A) : GOSUB 2000
24060 UNTIL WAITPOS (A) : POS(0)=0 : $(14)="..." : RETURN
24070 !
24500 LABEL "BAHNPOSITIONIERUNG": !
24510 SPD(0)=A(0) : ACC(0)=A(0) : POS(0)=A(0)
24520 FOR A=1 TO AA
24530 IF (A(0)@(A-1))=0 THEN GOTO 24550
24540 POS(A)=PD(POSN(A),A) : SPD(A)=AXP(1+SD*2,A) : ACC(A)=AXP(2+SD*2,A)
24550 NEXT A
24560 ! GOSUB 1300 ! Ist bei gemeinsamer Fahrt nicht notwendig!!!
24570 !
24580 GOSUB 26000 : GOSUB 1200 : ! Start
24590 !
24600 DO : !
24610 ON AA GOTO 24610,24620,24630,24640,24650
24620 UNTIL WAITPOS A1 : POS(0)=0 : $(14)="..." : GOTO 24660
24630 UNTIL WAITPOS A1,A2 : POS(0)=0 : $(14)="..." : GOTO 24660
24640 UNTIL WAITPOS A1,A2,A3 : POS(0)=0 : $(14)="..." : GOTO 24660

```

24650 UNTIL WAITPOS A1,A2,A3,A4 : POS(0)=0 : \$(14)="..." : GOTO 24660
 24660 RETURN

- Beschreibung:**
- Zeile 20020: Der Merker für den Automatikbetrieb wird gesetzt.
 - Zeile 20030: Die Funktionstasten werden gesperrt und die 4 LED's am COMTAC gelöscht
 - Zeile 20040: Die \$(11)-\$(14) werden in den Display-Zeilen 1-4 dargestellt.
 - Zeile 20050: Der STOP-Interrupt wird freigegeben (STOP-Taste und Eingang 2 des COMTAC)
 - Zeile 20060: Die Felder POS(), SPD(), ACC() werden zurückgesetzt.
 - Zeile 20080: Falls der Maschinennullpunkt schon angefahren wurde, wird die M0-Anfahroutine übersprungen.
 - Zeile 20100: Textanzeige zum Starten des M0-Punktes.
 - Zeile 20110-20120: Warten auf Drücken der Start-Taste.
 - Zeile 20130: Textanzeige M0-Punkt wird angefahren.
 - Zeile 20140: Start der Achse A2 zum M0-Punkt anfahren; die programmierte Zeit WT wird gewartet, bis das Signal M0-Punkt erreicht abgefragt wird.
 - Zeile 20150-20160: Abfrage auf M0-Punkt erreicht.
 - Zeile 20170: Textanzeige Realnull anfahren und Starten des R0-Punktes; Zeit WT abwarten.
 - Zeile 20180-20190: Abfrage auf R0-Punkt erreicht.
 - Zeile 20210-20270: wie 20120-20190
 - Zeile 20280: Nochmalige Abfrage, ob M0-Punkte angefahren wurden
- Beispiel:**
- Zeile 20320: Zuweisung Ausgang AS=2, AS=3; Rücksetzen dieser Ausgänge; Setzen des Ausganges A3 für einen Impuls.
 - Zeile 20330-20340: Abfrage des Einganges 3; Wird dieser Eingang innerhalb einer programmierten Zeit nicht wahr, so wird ein Fehlerstring angezeigt. Es wird auf die Bestätigung mit ENTER gewartet und der Eingang wieder abgefragt.
 - Zeile 20370: Vorbereitung der Achse 2 zum Einzelstart: Setzen der Variable SD auf 0; d.h. schnelle Automatik-Geschwindigkeit. Die Achse A2 soll positioniert werden, d.h. in die Tabellen A(0) wird der Binärcode dieser Achse A2 geschrieben. Die Position, die angefahren werden soll, ist die Nummer 1.
 - Zeile 20520: Vorbereitung der Achse 1+2 zum gemeinsamen Start: Setzen der Variable SD auf 1; d.h. langsame Automatik-Geschwindigkeit. Die Achsen A1+A2 sollen positioniert werden, d.h. in die Tabellen A(0) wird der Binärcode der Achse A12 geschrieben. Die Position, die angefahren werden soll, ist die Nummer 3.
 - Zeile 24010: In die Tabellen SPD(0) und ACC(0) wird der Binärcode der ausgewählten Achse geschrieben.
 - Zeile 24020: In die Tabellen SPD(A) und ACC(A) wird der Wert der Achsparameter geschrieben.
 - $AXP(1+SD*2,A) = AXP(\text{Achsparameter-Nr.}+SD*2,\text{Achs-Nr.})$; SD=Schnell/Langsam.
 - Zeile 24030: In die Tabelle POS(0) wird der Binärwert der ausgewählten Achse geschrieben. In die Tabelle POS(A) wird der Positionswert aus der Tabelle PD(..) geschrieben. Die Positionierung wird gestartet. $PD(1,A) = PD(\text{Positionsnummer},\text{Achsnummer})$
 - Zeile 24040-24060: Es wird auf Position Erreicht abgefragt. Während der Positionierung wird die Istposition der Achse angezeigt.
 - Achtung:** In der gleichen Zeile wird die Tabelle POS(0)=0 gesetzt. Dies ist wichtig um eine korrekte Stop/Weiter-Funktion zu realisieren.
 - Zeile 24510: In die Tabellen SPD(0), ACC(0) und POS(0) wird der Binärcode der ausgewählten Achse geschrieben.
 - Zeile 24520-24550: In einer Schleife wird in die Tabellen SPD(A) und ACC(A) der Wert der Achsparameter der ausgewählten Achsen geschrieben. In die Tabelle POS(A) werden die Positionswerte aus der Tabelle PD(POSN(A),A) der ausgewählten Achsen geschrieben.
 - Zeile 24560: Die Achsparameter brauchen nicht in die Achsen geschrieben werden, da dies vom Unterprogramm 26000 erledigt wird.
 - Zeile 24580: Das Unterprogramm, das die Grenzwerte der zwei beteiligten Achsen überwacht, wird aufgerufen. Die Achsen werden gestartet.
 - Zeile 24600-24650: Es wird auf Position Erreicht abgefragt.
 - Achtung:** In der gleichen Zeile wird die Tabelle POS(0)=0 gesetzt. Dies ist wichtig um eine korrekte Stop/Weiter-Funktion zu realisieren.

Beispiel: -

12.3.3 DC [F6] Parametrier-Betrieb

Funktion:	<p>Im Parametrierbetrieb werden Parameter und Positionen programmiert und spannungsfest abgespeichert. Sie werden im Automatikablauf benötigt, um den Ablauf korrekt zu steuern.</p> <p>Im Dialog wird ausgewählt, welche Parameter programmiert werden. Erst nach Bestätigung wird ein Passwort verlangt, da die Programmierung der Maschinenparameter, Maschinenpositionen, Achsparameter unterschiedliche Zugangsberechtigungen haben.</p>
Variablen:	<p>ERG = Ergebnis eines Tastendrucks SN = Stringnummer</p>
Strings:	\$(69), \$(38)-\$(41) = Anzeigestrings
Felder:	-
Code:	<p>z.B.:</p> <pre>30080 ERG= ASKKB [13,26]TABXY(1,3),\$(38),"?" : ! Maschinenparam.[mach.-para. 30090 IF ERG THEN GOSUB 31000 : GOTO 30160</pre>
Beschreibung:	<p>Zeile: 30080: In der Zeile 3 des Displays wird eine Ja=ENT/Nein=ESC verlangt. [13,26] : 13 = Code der ENTER-Taste; 26 = Code der ESC-Taste Zeile 30090: Das Ergebnis des Tastendrucks steht in der Variablen ERG. Ja = ENT -> ERG=TRUE Nein = ESC -> ERG = FALSE</p>
Beispiel:	-

12.3.4 DD Parameter laden: Modul, Positionen laden

Funktion:	<p>Die Tabellen AXP(..) Achsparameter, KP(..) Maschinenparameter, PD(..) Maschinenpositionen werden mit den Werten aus den spannungsfest abgelegten Datei geladen.</p>
Variablen:	<p>A= Achsnummer AA = max. Achsanzahl I = Laufvariable AXP = max. Anzahl der Achsparameter MAP = max. Anzahl der Maschinenparameter MPO = max. Anzahl der Maschinenpositionen BP = Bearbeitungsprogrammnummer</p>
Strings:	<p>\$(15) = "AXP" = Datei der Achsparameter \$(16) = "MAP" = Datei der Maschinenparameter und -positionen</p>
Felder:	<p>AXP(I,A) = Wert der Achsparameter(Achsparameter-Nr.,Achs-Nr.) AXU(..) = Untere Eingabegrenze der Achsparameter AXO(..) = Obere Eingabegrenze der Achsparameter AXS(..) = Stellenanzahl bei der Eingabe der Achsparameter</p> <p>PD(I,A) = Wert der Maschinenposition(Positions-Nr.,Achs-Nr.) aus Datei "MAP"; Speicherplatz 1..600 KP(I) = Wert der Maschinenparameter aus Datei "MAP"; Speicherplatz 601..645</p>
Code:	<pre>30500 ! DD= Achs-Parameter, Maschinenparameter und -Positionen laden ===== 30510 !=== load axis-parameter, machine-parameter, machine positions</pre>

```

=====
30520 ! Achsparameter|axisparameter
30530 FOR A=1 TO AA: ! Alle Achsen|all axis
30540 FOR I=1 TO AXP: ! Parameter 1 bis AXP|parameter 1 to max. parameter
30550 AXP(I,A)=|$(15)((A-1)*AXP+I): ! Achsparameter|axisparameter
30560 AXU(I,A)=|$(15)((A-1)*AXP+100+I): ! Untergrenze der Eingabe|low limit
30570 AXO(I,A)=|$(15)((A-1)*AXP+200+I): ! Obergrenze der Eingabe|up limit
30580 AXS(I,A)=|$(15)((A-1)*AXP+300+I): ! Stellenanzahl der Eingabe|digits
30590 IF AXO(I,A)=0 THEN AXO(I,A)=60000: ! Fuellen, wegen Eingabe|default
30600 IF AXS(I,A)=0 THEN AXS(I,A)=8: ! Fuellen, wegen Eingabe|default
30605 IF AXP(I,A)=0 THEN AXP(I,A)=1: ! Fuellen, wegen SD=0,AL=0|default
30610 NEXT I
30620 NEXT A
30630 ! Maschinen-Positionen|machine positions
30640 BP=1: ! Bearbeitungsprogramm 1|program No 1
30650 GOSUB 30900
30660 !
30800 ! Maschinen-Parameter|machine parameters
30802 FOR I=1 TO MPA: ! Paramter 1 bis MPA|parameter 1 to max. parameter
30804 KP(I)=|$(16)(600+I)
30806 NEXT I
30807 IF KP(10)=0 THEN KP(10)=1: |$(16)(600+10)=KP(10)
30808 IF KP(11)=0 THEN KP(11)=1000: |$(18)(800+11)=KP(11)
30810 KPU(2)=0: KPO(2)=100: KPS(2)=5: IF MPA<=2 THEN RETURN
30900 ! PD(..) laden|load PD(..)
30910 FOR A=1 TO AA: ! Alle Achsen|all axis
30920 FOR I=1 TO MPO: ! Position 1 bis MPO|position 1 to max. position
30930 PD(I,A)=|$(16)(((BP-1)*MPO*AA)+((A-1)*MPO)+I): ! Position
30940 NEXT I
30950 NEXT A
30960 RETURN
30970 !

```

Beschreibung: Zeile 30530-30620: Die Tabelle AXP() wird geladen.
 Zeile 30640-30680: Die Tabelle PD() wird geladen.
 Zeile 30802-30806: Die Tabelle KP() wird geladen.
 Zeile 30608-30894: Jedem Maschinenparameter wird eine Unter-/Obergrenze und eine Stellenanzahl für die Eingabe zugeteilt.

Beispiel: -

12.3.5 DE Maschinenparameter




Funktion: Es können maximal 44 Maschinenparameter programmiert werden. Die Werte werden in der Datei "MAP" gespeichert. Die Werte der Maschinenparameter stehen in der Tabelle KP(..).
 Speicherplatz in der Datei "MAP": 601-645.

Variablen: CN = Code für Passwort
 KC = Variable für Keyboard-Code
 YY = Nummer des Maschinenparameters
 SN = Stringnummer
 RT(☞) = Keyboard-Code für Pfeiltaste rechts
 LT(☞) = Keyboard-Code für Pfeiltaste links
 ER = Keyboard-Code für Enter-Taste
 EC = Keyboard-Code für Escape-Taste
 MPA = max. Anzahl der Maschinenparameter
 I = Laufvariable

Strings: \$(38), \$(53), \$(18), \$(57), \$(16) = Anzeigestrings
 \$(16) = Datei "MAP"

Felder:	KP(..) = Werte der Maschinenparameter KPO(..) = Obergrenze KPU(..) = Untergrenze KPS(..) = Stellenanzahl
Code:	z.B.: 31020 CN=Control (137) : GOSUB 52200 : IF NOT(CN) THEN GOTO 31260 31140 IF KC=ER THEN GOSUB 31600: ! Eingabe input 31220 IF ASKKB [ER,EC]TABXY(1,3)=0 THEN 31250 : ! alte wieder laden break 31230 FOR I=1 TO MPA: ;\$ (16)(600+I)=KP(I) : NEXT I 31240 GOTO 31260 31250 FOR I=1 TO MPA:KP(I)= ;\$ (16)(600+I) : NEXT I 31260 RETURN : ! Zur Programmierauswahl 31600 ! ===== Parameter-Werte eingeben ===== 31610 ! 31620 KK=KP(YY) : OG=KPO(YY) : UG=KPU(YY) : SA=KPS(YY) 31630 \$(3)=\$(42) : GOSUB 52000 : KP(YY)=KK 31640 RETURN
Beschreibung:	Zeile 31020: Das Passwort Control (137) wird in die Variable CN geschrieben. Das Passwort wird im UP 52200 eingegeben. Falls das Passwort nicht übereinstimmt wird die Funktion verlassen. Zeile 31240: Beim Druck der ENTER-Taste wird in die Eingaberoutine verzweigt. Zeile 31220: Es wird eine Ja/Nein-Abfrage verlangt, ob die eingegebenen Werte gespeichert werden sollen. Zeile 31230: Die Werte werden in die Datei "MPA" gespeichert, ab Speicherplatz 601. Zeile 31250: Die alten Werte werden aus der Datei "MAP" in die Tabelle KP() zurückgeschrieben. Zeile 31620: Zuweisung des Wertes, Obergrenze, Untergrenze, Stellenanzahl aus der Tabelle KP(..), Zeile 31630: Sprung ins Eingabeprogramm und zuweisen des eingegebenen Wertes in die Tabelle KP(..).
Beispiel:	-

12.3.6 DF Maschinenpositionen

Funktion:	Es können maximal 31 Maschinenpositionen pro Bearbeitungsprogramm programmiert werden. Die Werte werden in der Datei "MAP" gespeichert. Die Werte der Maschinenpositionen stehen in der Tabelle PD(..). Speicherplatz in der Datei "MAP": 1-600. Jede Maschinenposition besteht aus Positionswerten aller Achsen. Wurde zuvor im Handbetrieb eine Position geteacht, so wird sie durch Drücken der INS-Taste in die angewählte Maschinenposition geschrieben.
Variablen:	CN = Code für Passwort KC = Variable für Keyboard-Code ZZ = Nummer der Maschinenposition SN = Stringnummer A = Achsnummer AA = max. Achsanzahl BP = Nummer des Bearbeitungsprogramms RT() = Keyboard-Code für Pfeiltaste rechts LT() = Keyboard-Code für Pfeiltaste links UP() = Keyboard-Code für Pfeiltaste hoch

DN(↓) = Keyboard-Code für Pfeiltaste tief
 IS = Keyboard-Code für Inserttaste Insert
 ER = Keyboard-Code für Enter-Taste
 EC = Keyboard-Code für Escape-Taste
 MPO = max. Anzahl der Maschinenpositionen
 I, N = Laufvariable

Strings: \$(39), \$(82), \$(43), \$(44), \$(29), \$(69), \$(57) = Anzeigestrings
 \$(16) = Datei "MAP"

Felder: PD(..) = Werte der Maschinenpositionen
 NG(..) = Negative Endgrenze aller COMPAX Achsen = P12
 PG(..) = Positive Endgrenze aller COMPAX Achsen = P11

Code: z.B.:
 32070 GOSUB 35000 : ! Eingabe Bearbeitungsprog.-Nr
 32080 IF BP=0 THEN GOTO 32070
 32090 FOR N=1 TO AA
 32100 FOR I=1 TO MPO
 32110 PD(I,N)=|;\$(16)(((BP-1)*MPO*AA)+((N-1)*MPO)+I)
 32120 NEXT I
 32130 NEXT N

 32190 DISP TABXY(1,3),\$(82),VAL\$(BP)
 32200 DISP TABXY(1,2),\$(44)[1,8+AA*8] : DISP USING(#####.##)
 32210 DISP TABXY(7,2),\$(43)[(A*2)-1,A*2]
 32220 \$(4)=|;\$(18)(69+ZZ)[1,,92] : SN=4 : GOSUB 2600
 32230 FOR N=1 TO AA
 32240 DISP TABXY(9+(N-1)*8,3),PD(ZZ,N) : NEXT N
 32250 DISP USING(0)

 32290 IF KC>=49'AND'KC<=48+AA THEN A=KC-48

 32600 ! ----- Daten eingeben-----
 32610 !
 32620 KK=PD(ZZ,A) : UG=NG(A) : OG=PG(A) : SA=8
 32630 !
 32640 \$(4)=\$(44)[(A-1)*8+10,A*8+8], " , ",|;\$(18)(69+ZZ)[1,,92],VAL\$(PD(ZZ,A))
 32650 SN=4 : GOSUB 2220
 32660 \$(3)=\$(42) : GOSUB 52000 : PD(ZZ,A)=KK : GOSUB 2124
 32670 RETURN

Beschreibung: Zeile 32070: Sprung ins Eingabeprogramm für die Bearbeitungsprogrammnummer
 Zeile 32080: Die Bearbeitungsprogrammnummer BP muß >0 sein, sonst Neueingabe
 Zeile 32090-32130: Schleife, in der die Werte aus der Datei "MAP" in die Tabelle PD(..) geschrieben werden.

Zeile 32190: Anzeige der Bearbeitungsprogrammnummer in Display-Zeile 3
 Zeile 32200: Anzeige der Achsbezeichnung abhängig von der Achsanzahl, einschalten der formatierten Anzeige für die Position
 Zeile 32210: Anzeige der aktuell angewählten Achse
 Zeile 32220: Anzeige der Bezeichnung der Maschinenposition
 Zeile 32230-32240: Anzeige der Maschinenposition
 Zeile 32250: Ausschalten der formatierten Anzeige

Zeile 32290: Die Achsnummer wird durch Drücken der Zahlen 1..4 ausgewählt.

Zeile 32620: Zuweisung des Wertes aus der Tabelle PD(..), der Unter-/Obergrenze aus der Tabelle der Endgrenzen und setzen der Stellenanzahl.

Zeile 32640-32650: Anzeige der Achsnummer, Bezeichnung der Maschinenposition und des Wertes der Maschinenposition. ([1,,92] = String einlesen von Stelle 1 - "\" aus der Datei)

Zeile 32660: Sprung ins Eingabeprogramm und zuweisen des eingegebenen Wertes in die Tabelle PD(..).

Beispiel: -

12.3.7 DG Achsparameter

Funktion: Es können maximal 25 Achsparameter/Achse programmiert werden. Die Werte werden in der Datei "AXP" gespeichert. Die Werte der Achsparameter stehen in der Tabelle AXP(..). Speicherplatz in der Datei "AXP": 1-100.

Variablen: CN = Code für Passwort
 KC = Variable für Keyboard-Code
 PN = Nummer des Achsparameters
 SN = Stringnummer
 A = Achsnummer
 AA = max. Achsanzahl
 RT(☞) = Keyboard-Code für Pfeiltaste rechts
 LT(☞) = Keyboard-Code für Pfeiltaste links
 ER = Keyboard-Code für Enter-Taste
 EC = Keyboard-Code für Escape-Taste
 AXP = max. Anzahl der Achsparameter
 KK = Zuweisung des Wertes für die Eingabe
 I, N = Laufvariable

Strings: \$(40), \$(52), \$(43), \$(18), \$(42), \$(69), \$(57),
 \$(15 = Datei "AXP")

Felder: AXP(..) = Wert des Achsparameters
 AXO(..) = Obergrenze
 AXU(..) = Untergrenze
 AXS(..) = Stellenanzahl

Code: z.B.:
 33220 KK=AXP(PN,A) : OG=AXO(PN,A) : UG=AXU(PN,A) : SA=AXS(PN,A)
 33230 GOSUB 52000 : AXP(PN,A)=KK

 33300 IF ASKKB [ER,EC]TABXY(1,3)=0 THEN 33350 : ! alte wieder laden|break
 33310 FOR A=1 TO AA: FOR I=1 TO AXP
 33320 |;\$(15)((A-1)*AXP+I)=AXP(I,A) : |;\$(15)((A-1)*AXP+100+I)=0 : !AXU(I,A)
 33330 |;\$(15)((A-1)*AXP+200+I)=60000 : |;\$(15)((A-1)*AXP+300+I)=8
 33340 NEXT I : NEXT A: GOTO 33390
 33350 FOR A=1 TO AA: FOR I=1 TO AXP
 33360 AXP(I,A)=|;\$(15)((A-1)*AXP+I) : AXU(I,A)=|;\$(15)((A-1)*AXP+100+I)
 33370 AXO(I,A)=|;\$(15)((A-1)*AXP+200+I) : AXS(I,A)=|;\$(15)((A-1)*AXP+300+I)
 33380 NEXT I : NEXT A
 33390 RETURN

Beschreibung: Zeile 33220: Zuweisung des Wertes des Achsparameters, der Ober-/Untergrenze, der Stellenanzahl aus der Tabelle AXP(), AXO(), AXU(), AXS().
 Zeile 33230: Aufruf des Eingabeprogrammes und Zuweisung des eingegebenen Wertes in die Tabelle AXP().

Zeile 33300: Frage, ob die Werte gespeichert werden sollen. ER,EC sind Keyboard-Codes von ENTER/ESC.

Zeile 33310-33340: Zuweisung der Werte in die Datei "AXP".

Zeile 33350-33380: Zurückschreiben der alten Werte aus der Datei "AXP" in die Tabellen AXP(), ...

Beispiel: -

12.3.8 DI [F7] Hand-Betrieb

Funktion:	<p>Der Handbetrieb dient dazu, mittels der Cursortasten alle Achsen mit einer schnellen und langsamen Handgeschwindigkeit zu verfahren.</p> <p>Teach-In-Funktion: Durch Drücken der INS-Taste wird die aktuelle Position in das Feld POS(Achsnummer) eingelesen. Diese Position kann dann in Parametrierbetrieb Maschinenpositionen weiterverwendet werden.</p>
Variablen:	<p>CN = Code für Passwort</p> <p>KC = Variable für Keyboard-Code</p> <p>NN = Zwischenspeicher der Hand-Achsnummer</p> <p>NNN =Merker für langsam/schnell Umschaltung</p> <p>N_ = Merker, ob Taste gedrückt wurde</p> <p>PN = COMPAX-Parameter-Nummer 5 und 9 (Handgeschwindigkeit-/rampe)</p> <p>SN = Stringnummer</p> <p>A = Achsnummer</p> <p>AA = max. Achsanzahl</p> <p>RT(→) = Keyboard-Code für Pfeiltaste rechts</p> <p>LT(←) = Keyboard-Code für Pfeiltaste links</p> <p>UP(↑) = Keyboard-Code für Pfeiltaste hoch</p> <p>DN(↓) = Keyboard-Code für Pfeiltaste tief</p> <p>IS = Keyboard-Code für Pfeiltaste Insert</p> <p>DP = Keyboard-Code für Punkt</p> <p>ER = Keyboard-Code für Enter-Taste</p> <p>EC = Keyboard-Code für Escape-Taste</p> <p>M0 = Merker, ob M0-Punkt angefahren wurde</p> <p>I, N = Laufvariable</p>
Strings:	\$ (50), \$ (51), \$ (44), \$ (54), \$ (75) = Anzeigestrings
Felder:	POS(Achsnummer) = Speicher der geteachten Position aller Achsen
Code:	<p>z.B.:</p> <pre> 40100 FOR A=1 TO AA:CPXIMASK(A)=63 : NEXT A: ! Maskierung mask inputs 40110 A=NN : PN=5 : IF NNN THEN PW=AXP(6,A) ELSE PW=AXP(5,A) 40120 GOSUB 50500 : DISP TABXY(21,2),"V=",VAL\$(PW),"% " 40130 PN=9 : IF NNN THEN PW=AXP(8,A) ELSE PW=AXP(7,A) 40140 GOSUB 50500 : DISP TABXY(31,2),"B=",VAL\$(PW),"ms " 40180 DISPCPXZ A,1,0,3 40200 DO : UNTIL (CPXSTS(A)@3) : ! Warten bis Achsen auch stehen wait 40280 IF NN=A1'AND'KC=LT THEN CPXCTR(A)=2 40290 IF NN=A1'AND'KC=RT THEN CPXCTR(A)=4 40300 IF NN=A2'AND'KC=LT THEN CPXCTR(A)=2 40310 IF NN=A2'AND'KC=RT THEN CPXCTR(A)=4 40320 IF NN=A3'AND'KC=DN THEN CPXCTR(A)=2 40330 IF NN=A3'AND'KC=UP THEN CPXCTR(A)=4 40340 IF NN=A4'AND'KC=LT THEN CPXCTR(A)=2 40350 IF NN=A4'AND'KC=RT THEN CPXCTR(A)=4 40480 FOR N=1 TO AA: OUTPUT 1,N,"S1" 40490 POS(N)=VAL\$(#1)[7]) : NEXT N </pre>
Beschreibung:	<p>Zeile 40100: Die Standardeingänge 1-6 werden maskiert, d.h. sie sind von außen nicht mehr zugänglich. Die Eingänge 2 (Hand+) und 3 (Hand-) können über die Software gesetzt werden.</p> <p>Zeile 40110-40140: Abhängig von der schnell/langsam Umschaltung werden die COMPAX-Parameter 5,9 mit den Werten aus der Tabelle AXP(Achsparemeter-Nr.,Achs-Nr.) beschrieben.</p>

Zeile 40180: Anzeige des Status S1=Istposition der angewählten Achse in der Display-Zeile 3

Zeile 40200: In einer Schleife wird gewartet, bis die akt. Achse still steht; erst dann darf wieder positioniert werden.

Zeile 40280-40350: Abhängig von der Achse wird der Eingang Hand +/- über die Software gesetzt, und so das Handverfahren eingeleitet. KC ist der Keyboard-Code der Cursor-Tasten.

CPXCTR(A)=2: Hand + mit Achse A; CPXCTR(A)=4: Hand- mit Achse A

Zeile 40480-40490: Durch Drücken von INS wird die aktuelle Position aller Achsen in das Feld POS(Achs-Nr.) geschrieben.

Beispiel:

-

12.3.9 DJ [F9] Maschinen-Nullpunkt manuell

Funktion: Diese Funktion dient dazu, den Maschinennullpunkt einer ausgewählten Achse anzufahren.

Variablen: CN = Code für Passwort
KC = Variable für Keyboard-Code
SN = Stringnummer
A = Achsnummer
AA = max. Achsanzahl
ST = Keyboard-Code für START
WT = Wartezeit nach M0 Start
EC = Keyboard-Code für Escape-Taste
M0 = Merker, ob M0-Punkt angefahren wurde
I = Laufvariable

Strings: \$(30), \$(55), \$(70), \$(44), \$(66) = Anzeigestrings

Felder: -

Code: z.B.:
41080 M0=TRUE : FOR I=1 TO AA:M0=(M0'AND'(CPXSTS(I)@2)) : NEXT I
41090 DISP TABXY(1,2),\$(44)[(A-1)*8+10,((A-1)+1)*8+8]

41230 \$(1)="PH" : GOSUB 1000 : WAIT WT : KC=0
41240 DO : KC= KBDCODE
41250 UNTIL (CPXSTS(A)@2)'OR'(KC=193)
41260 IF KC=193 THEN M0=FALSE : \$(1)="BK" : GOSUB 1000 : GOTO 41320
41270 SN=66 : GOSUB 2230 : DISP "..."
41280 \$(1)="PA0" : GOSUB 1000 : WAIT WT : KC=0
41290 DO : KC= KBDCODE
41300 UNTIL (CPXSTS(A)@4)'OR'(KC=193)
41310 IF KC=193 THEN \$(1)="BK" : GOSUB 1000

Beschreibung: Zeile 41080: Abfrage des Ausganges 3 jeder COMPAX-Achse, ob M0 angefahren wurde.
Zeile 41090: Anzeige der ausgewählten Achsnummer

Zeile 41230: Start M0 anfahren und die Zeit WT abwarten, bevor M0 Erreicht abgefragt wird.

Zeile 41240-41250: Warten auf M0 Erreicht; mit Stop-Taste kann die Bewegung unterbrochen werden.

Zeile 41260: Wenn Stop gedrückt wurde, dann senden von "BK" an die ausgewählte Achse; die Verfahrbewegung dieser Achse wird gestoppt.

Zeile 41270: Anzeige von Realnull anfahren.

Zeile 41280: Start R0 anfahren und die Zeit WT abwarten, bevor R0 Erreicht abgefragt wird.

Zeile 41290-41300: Warten auf R0 Erreicht; mit Stop-Taste kann die Bewegung unterbrochen werden.

Zeile 41310: Wenn Stop gedrückt wurde, dann senden von "BK" an die ausgewählte Achse; die Verfahrbewegung dieser Achse wird gestoppt.

Beispiel:

-

12.3.10 DK [F10] Test Eingänge/Ausgänge

Funktion: In dieser Funktion wird der Zustand der Eingänge des COMTAC und aller COMPAX-Achsen angezeigt. Alle Ausgänge können gesetzt bzw. zurückgesetzt werden.

Variablen: CN = Code für Passwort
 KC = Variable für Keyboard-Code
 SN = Stringnummer
 A = Achsnummer
 AA = max. Achsanzahl
 EC = Keyboard-Code für Escape-Taste
 RT(␣) = Keyboard-Code für Cursor rechts
 IS = Keyboard-Code für INS-Taste
 DE = Keyboard-Code für DEL-Taste
 A1, A2, A3, A4 = Achsnummern
 IMIN = 1. Ausgang der gesetzt/zurückgesetzt werden kann
 IMAX = letzter Ausgang der gesetzt/zurückgesetzt werden kann
 I,N = Laufvariable

Strings: \$(31), \$(59), \$(19) = Anzeigestrings

Felder: AS(..) = Ausgangsfeld

Code: z.B.:
 42060 FOR I=1 TO MA:AS(I)=I : NEXT I
 42070 FOR I=33 TO 32+AA*16 : AS(I)=I
 42080 NEXT I
 42090 A=0 : IMIN=1 : IMAX=32+AA*16 : I=IMIN : GOSUB 2102

 42130 IF A=0 THEN DISP TABXY(1,2),"COMTAC: E16-1=" : DB. IN(16~1)
 42140 IF A=A1 THEN DISP TABXY(1,2),"CPX A1: E16-1=" : DB. CPXINP(A1)

 42260 IF AS(I)<=32 THEN SETOUT AS(I) : GOTO 42310
 42270 IF AS(I)<=48 THEN CPXOUT(A1;(AS(I)-32))=1 : GOTO 42310

 42410 IF CONTROL(0)=2000 THEN BINOUT(16~1)=0 ELSE BINOUT(32~1)=0
 42420 FOR N=1 TO AA:CPXOUT(N;16~1)=0 : NEXT N : ! Reset Ausgaenge|reset outp

Beschreibung: Zeile 41060: Die Ausgangs-Nummern 1-16 werden in die Tabelle AS(..) geschrieben.
 Zeile 42070-42080: Die Ausgangs-Nummern aller angeschlossenen COMPAX-Achsen werden in die Tabelle AS(..) geschrieben.
 Zeile 42090: Die Grenzen der Ausgänge, die gesetzt/zurückgesetzt werden können, wird festgelegt.

Zeile 42130: Der Status der COMTAC-Eingänge 1-16(32) wird in Display-Zeile 2 angezeigt.
 Zeile 42140: Der Status der COMPAX-Eingänge 1-16(32) der Achse 1 wird in Display-Zeile 2 angezeigt.

Zeile 42260: Der angewählte Ausgang des COMTAC wird gesetzt.

Zeile 42270: Der angewählte Ausgang der COMPAX Achse 1 wird gesetzt.

Zeile 42410: Alle Ausgänge des COMTAC werden zurückgesetzt.

Zeile 42420: Alle Ausgänge der COMPAX Achsen werden zurückgesetzt.

Beispiel:

-

12.3.11 DL [F11] Korrektur Maßbezug

Funktion: In dieser Funktion kann der Realnullpunkt (R0) aller COMPAX-Achsen verändert werden. Der R0 dient als Bezugspunkt aller Maschinenpositionen. Dies ist notwendig, wenn z.B. der M0-Punkt verschoben wurde.

Variablen: CN = Code für Passwort
KC = Variable für Keyboard-Code
SN = Stringnummer
A = Achsnummer
AA = max. Achsanzahl
ER = Keyboard-Code für Enter-Taste
EC = Keyboard-Code für Escape-Taste
PN = COMPAX Parameternummer (1,11,12)
YY = Zwischenspeicher der akt. Achsnummer

Strings: \$(32), \$(53), \$(45), \$(43), \$(57), \$(69) = Anzeigestrings

Felder: KO(..) = Korrekturwert
RN(..) = Realnullwert aller COMPAX-Achsen = P1
PG(..) = Positive Endgrenze aller COMPAX Achsen = P11
NG(..) = Negative Endgrenze aller COMPAX Achsen = P12

Code: z.B.:
43100 \$(45)[15,16]=\$\$(43)[(YY*2)-1,YY*2] : DISP TABXY(1,3),\$(45),KO(YY)

43230 PN=1 : GOSUB 50400 : PW=PW+KO(A) : GOSUB 50500 : RN(A)=PW
43240 PN=11 : GOSUB 50400 : PW=PW+KO(A) : GOSUB 50500 : PG(A)=PW-RN(A)
43250 PN=12 : GOSUB 50400 : PW=PW+KO(A) : GOSUB 50500 : NG(A)=PW-RN(A)

43270 OUTPUT 1,255;"VP"

43600 ! ----- R0-Werte eingeben|input -----
43610 !
43620 KK=RN(YY) : OG=1000 : UG=-1000 : SA=5
43630 \$(3)=\$(42) : GOSUB 52000 : KO(YY)=KK
43640 RETURN

43800 ! === Verfahrbereiche der Achsen ermitteln|read limits =====
43810 FOR A=1 TO AA
43820 PN=1 : GOSUB 50400 : RN(A)=PW : PRINT A,"P 1=",PW
43830 PN=11 : GOSUB 50400 : PG(A)=PW-RN(A) : PRINT A,"P11=",PW
43840 PN=12 : GOSUB 50400 : NG(A)=PW-RN(A) : PRINT A,"P12=",PW
43850 NEXT A
43860 RETURN

Beschreibung: Zeile 43100: Anzeige der Achsnummer und des Korrekturwertes

Zeile 43230: Einlesen des R0 = P1, Verrechnen des R0 mit dem Korrekturwert und Schreiben des neuen R0 in die COMPAX-Achse.

Zeile 43240: Einlesen der pos. Endgrenze = P11, Verrechnen der pos. Endgrenze mit dem Korrekturwert und Schreiben der neuen pos. Endgrenze in die COMPAX-Achse.

Zeile 43250: Einlesen der neg. Endgrenze = P12, Verrechnen der neg. Endgrenze mit dem Korrekturwert und Schreiben der neuen neg. Endgrenze in die COMPAX-Achse.

Zeile 43270: Senden des Befehls "VP" = Valid Parameter an alle COMPAX Achsen, sodaß

die Parameter in das COMPAX übernommen werden.

Zeile 43620: Zuweisung des R0-Wertes, der Unter-/Obergrenze und der Stellenanzahl.

Zeile 43630: Sprung ins Eingabeprogramm

Zeile 43810-43850: In einer Schleife werden die Verfahrbereiche der Achsen ermittelt. Die Parameternummer wird zugewiesen, das Einlesenprogramm wird aufgerufen, die Tabelle wird beschrieben.

Beispiel: -

12.3.12 DN [F13] COMTAC-Initialisierung

Funktion: In dieser Funktion werden die COMTAC-Controlworte mit Standardwerten beschrieben.

Variablen: CN = Code für Passwort
I = Schleifenzähler
ZZ = Zwischenmerker für den Wert der Controlworte

Strings: \$(34) = Anzeigestring

Felder: -

Code: z.B.:
44040 ! CTC-Parameter:
44050 ! D: 0,2,13,34,8,0,1,13,1,0 : ! P50 - P59 / RS232/1 - Terminal
44060 DATA 0,2,13,2,6,0,1,13,1,10 : ! P60 - P69 / RS485
44070 DATA AA,32,0,1,11,10,10,0,0,0 : ! P70 - P79 / RS485-Feldbus
44080 DATA 0,0,0,0,0,0,0,0,0,0 : ! P80 - P89 / frei
44090 DATA 0,3,62,34,7,0,1,13,1,0 : ! P90 - P99 / RS232/2 - Host/Floppy
44100 DATA 0 : ! P100 / Terminal
44110 RESTORE
44120 FOR I=60 TO 100
44130 READ ZZ : CONTROL(I)=ZZ
44140 NEXT I

Beschreibung: Zeile 44040-44100: Anlegen eines Datenfeldes im BASIC.
Zeile 44110: Zurücksetzen des READ-Zeigers auf die 1. DATA Variable.
Zeile 44120-44140: Zuweisung der DATA-Werte in die CONTROL-Worte 60-100.

Beispiel: -

12.3.13 DO [F14] COMPAX-Parameter-Manager

Funktion: Dieses Softwareprogramm wird abgebrochen. Über das COMTAC kann der Parameter-Manager geladen werden. Dies ist ein eigenständiges Programm, um Parameter des COMTAC und der COMPAX-Achsen zu editieren, abzuspeichern und zu laden. Dieses Programm wird hier nicht näher dokumentiert.

Variablen: CN = Code für Passwort
SN = Stringnummer

Strings: \$(33) = Anzeigestring

Felder: -

Code: 45000 ! CC_Edit aufrufen
 45010
 45020 STOP ONCPXERR : SN=35 : GOSUB 2600
 45030 GOSUB 50000 : IF CONTROL(136)=1 THEN AS=7 : GOSUB 1600
 45040 CN=CONTROL(139) : GOSUB 52200 : IF NOT(CN) THEN GOTO 45060
 45050 DL=0 : DB=0 : \$(5)="CC_EDIT.PRG" : GOSUB 51000
 45060 GOTO 10000

Beschreibung: Zeile 45020: Der COMPAX-Fehler-Interrupt wird gesperrt, Text wird angezeigt.
 Zeile 54030: Die Funktionstasten werden gesperrt.
 Zeile 45040: Aufforderung der Eingabe des Passwortes.
 Zeile 45050: Display wird gelöscht. Programm wird beendet.

Beispiel: -

12.3.14 DP [F16] Anlagen-Info

Funktion: In dieser Version werden die Projektbezeichnungen, Firmenname, ... sowie die Versionen der Software, des COMTAC und der COMPAX-Achsen angezeigt.

Variablen: SN = Stringnummer
 ERG = Ergebnis eines Tastendrucks (Ja/NEIN)
 PV = Programmversion
 PD = Programmdatum
 A = Achsnummer
 AA = max. Achsanzahl

Strings: \$(68), \$(89)-\$(112) = Anzeigestrings
 \$(1) = RS485-OUTPUT-String
 \$(4) = Dummystring

Felder: -

Code: 46000 !=== Anlagen-Infos / Aboutbox ===
 46010 !
 46020 DISABLE ONCPXERR : GOSUB 50000
 46030 SN=35 : GOSUB 2600
 46040 DISP TABXY(1,2),\$(98) : DISP TABXY(12,2),\$(99)
 46050 DISP TABXY(1,3),\$(100)
 46060 CLEAR D,4 : ERG= ASKKBD [13,26]TABXY(1,4),\$(68)
 46070 IF NOT(ERG) THEN GOTO 46480
 46080 !
 46090 DISP TABXY(1,2),\$(101) : DISP TABXY(1,3),\$(102)
 46100 CLEAR D,4 : ERG= ASKKBD [13,26]TABXY(1,4),\$(68)
 46110 IF NOT(ERG) THEN GOTO 46480
 46120 !
 46130 DISP TABXY(1,2),\$(103) : DISP TABXY(1,3),\$(104)
 46140 CLEAR D,4 : ERG= ASKKBD [13,26]TABXY(1,4),\$(68)
 46150 IF NOT(ERG) THEN GOTO 46480
 46160 !
 46170 DISP TABXY(1,2),\$(105) : DISP TABXY(1,3),\$(106)
 46180 CLEAR D,4 : ERG= ASKKBD [13,26]TABXY(1,4),\$(68)
 46190 IF NOT(ERG) THEN GOTO 46480
 46200 !
 46210 DISP TABXY(1,2),\$(107) : \$(4)=\$(108) : ! PD/PV
 46220 PRINT USING(###) : \$(4)[21]="V",VAL\$(PV)
 46230 PRINT USING(#####) : \$(4)[26]=" / ",VAL\$(PD)
 46240 DISP TABXY(1,3),\$(4) : PRINT USING(0)
 46250 CLEAR D,4 : ERG= ASKKBD [13,26]TABXY(1,4),\$(68)
 46260 IF NOT(ERG) THEN GOTO 46480
 46270 !

```

46280 $(4)=$(109) : ! CTC/V
46290 PRINT USING(#####) : $(4)[24]=", ", "V2.00/", VAL$(CONTROL(1))
46300 DISP TABXY(1,2),$(4) : $(4)=$(110)
46310 PRINT USING(#####) : $(4)[12]="SN", VAL$(CONTROL(3))
46320 DISP TABXY(1,3),$(4) : PRINT USING(0)
46330 CLEAR D,4 : ERG= ASKKBD [13,26]TABXY(1,4),$(68)
46340 IF NOT(ERG) THEN GOTO 46480
46350 !
46360 FOR A=1 TO AA
46370 $(4)=$(111) : $(4)[7,7]=VAL$(A) : ! CPX/Type
46380 $(1)="S031" : GOSUB 1100 : $(4)[14]=$(#1)[12]
46390 $(1)="S032" : GOSUB 1100 : $(4)[25]=" / ",$(#1)[10]
46400 GOSUB 2102 : DISP TABXY(1,2),$(4)
46410 $(4)=$(112) : ! CPX/SN
46420 $(1)="S033" : GOSUB 1100 : $(4)[23]=$(#1)[14]
46430 $(1)="S034" : GOSUB 1100 : $(4)[29]=$(#1)[13]
46440 GOSUB 2103 : DISP TABXY(1,3),$(4)
46450 CLEAR D,4 : ERG= ASKKBD [13,26]TABXY(1,4),$(68)
46460 NEXT A
46470 !
46480 GOTO 10000
46490 !

```

Beschreibung: Zeile 46020: Der COMPAX-Fehler und die Funktionstasten werden gesperrt.
 Zeile 46030: Ein Textstring wird in Zeile 1 zentriert angezeigt.
 Zeile 46040-46050: Texte bezüglich des Projekts werden in Zeile 2,3 angezeigt.
 Zeile 46060-46070: Frage, ob weitere Anzeiger oder Abbruch.
 Zeile 46090-46190: wie 46040-46070.
 Zeile 46210-46260: Texte bezüglich der Programmversion werden angezeigt und Frage, ob Abbruch.
 Zeile 46280-46350: Texte bezüglich der COMTAC-Version werden angezeigt und Frage, ob Abbruch.
 Zeile 46360-46460: Texte bezüglich der COMPAX-Version werden angezeigt und Frage, ob Abbruch.
 Zeile 46480: Rücksprung ins Hauptmenü.

Beispiel: -

12.3.15 DQ Passwort-Eingabe

Funktion: Der Zugriff auf Betriebsarten ist durch Passworte geschützt. Somit können verschiedene Benutzerebenen definiert werden. Als Passworte sind Zahlen einzugeben.

Variablen: CN = Passwort als Nummer
 SN = Stringnummer
 I = Schleifenzähler
 KC = Keyboard-Code
 TRUE = Konstante = 1 oder 65535 =wahr
 FALSE = Konstante = 0 = nicht wahr

Strings: \$(4) = Dummstring
 \$(49) = Anzeigestring

Felder: -

Code: 52200 !=== Passwort-Eingabe ===
 52210 !
 52220 IF DBG THEN CN=TRUE : RETURN
 52230 GOSUB 2134 : SN=48 : GOSUB 2230
 52240 \$(4)=VAL\$(CN) : \$(7)=VAL\$(CONTROL(139)) : CN=TRUE : CURSOR_D 3
 52250 FOR I=1 TO LEN\$(4)
 52260 DO : !


```

52270 WHILE KBDCODE
52280 DO : KC= KBDCODE
52290 UNTIL KC
52300 IF (CHR$(KC)<>$ (4)[I,I])'AND'(CHR$(KC)<>$ (7)[I,I]) THEN CN=FALSE
52310 DISP ""
52320 NEXT I
52330 IF CN=TRUE THEN GOTO 52360
52340 GOSUB 2134 : DISP TABXY(1,3),$ (49)
52350 WAIT 1200
52360 DO : !
52370 WHILE KBDCODE
52380 GOSUB 2134 : CURSOR_D 0 : RETURN
52390 !

```

Beschreibung: Zeile 52220: Wenn der Schalter DBG (Debug=Testmode) eingeschaltet ist, dann wird CN=TRUE gesetzt. Somit sind keine Eingaben zu tätigen.

Zeile 52230: Die Display-Zeilen 3,4 werden gelöscht. Der \$(48) wird angezeigt.

Zeile 52240: Der Code wird in den \$(4) umgeschrieben, CN wird TRUE gesetzt und der Cursor als Unterstrich dargestellt. Die höchste Priorität wird in \$(7) geschrieben.

Zeile 52250-52320: In einer Schleife wird jedes eingegebene Zeichen mit dem Codestring verglichen. Entspricht ein Zeichen nicht dem Codestring, so wird CN=FALSE gesetzt. Jedes eingegebene Zeichen wird als '*' dargestellt. Die höchste Priorität ist immer gültig.

Zeile 52330-52350: Wenn der Code richtig eingegeben wurde, dann Sprung. Wenn der Code falsch eingegeben wurde, dann wird Display-Zeile 3,4 gelöscht. Der Textstring \$(49) wird 1.2 sec. angezeigt.

Zeile 52360-52370: Es wird gewartet, bis keine Taste mehr gedrückt ist.

Zeile 52380: Die Display-Zeilen 3,4 werden gelöscht und die Cursordarstellung wird unsichtbar geschaltet

Beispiel: CN=620 : GOSUB 52200 : IF NOT(CN) THEN STOP

12.4 DR [F8] Einrichtbetrieb

Funktion:	Im Einrichtbetrieb können anlagenspezifische Funktionen realisiert werden. Dies könnte z.B. ein Hand-/Teachbetrieb sein. Um die notwendigen Sicherheitsmaßnahmen zu gewährleisten, sind Hardwaremaßnahmen (NOTSTOP der Achsen, Freischalten,...) erforderlich. ??
Variablen:	SN = Stringnummer CN = Code für Passwort AS = Ausgangsnummer
Strings:	\$(29) = Anzeigestring
Felder:	
Code:	<pre> 47000 ! DR= Einrichtbetrieb control mode ===== 47010 ! 47020 GOSUB 50000 : IF CONTROL(136)=1 THEN AS=7 : GOSUB 1600 47030 GOSUB 2124 : SN=29 : GOSUB 2600 47040 CN=CONTROL(138) : GOSUB 52200 : IF NOT(CN) THEN GOTO 47080 47050 ! Hier Platz fuer Maschinenspez. Funktionen 47060 ! place for specific functions 47070 GOSUB 52700 47080 GOTO 10000 47090 ! </pre>
Beschreibung:	<p>Zeile 47020: Sperren der Funktionstasten; Falls SPS-Betrieb dann wird Ausgang 7 'Comtac Mode' gesetzt</p> <p>Zeile 47030: Löschen der Zeilen 2-4; Anzeige des Einrichtbetriebes in Zeile 1</p> <p>Zeile 47040: Abfrage des Passwortes</p> <p>Zeile 47070: Weiter mit ENTER-Tastendruck</p>
Beispiel:	-

12.5 DS [F15] Programmende

Funktion:	Diese Funktion erlaubt den Abbruch des Programmes. Gleichzeitig werden die Maskierungen der Ein-/Ausgänge aufgehoben.
Variablen:	SN = Stringnummer CN = Code für Passwort AS = Ausgangsnummer
Strings:	\$(36), \$(46) = Anzeigestrings
Felder:	
Code:	<pre> 48000 ! DS= Programmende programm end ===== 48010 ! 48020 GOSUB 50000 : IF CONTROL(136)=1 THEN AS=7 : GOSUB 1600 48030 GOSUB 2124 : SN=36 : GOSUB 2600 48040 CN=CONTROL(139) : GOSUB 52200 : IF NOT(CN) THEN GOTO 10000 48050 CPXIMASK(255)=0 : CPXOMASK(255)=0 48060 CLEAR D : SN=46 : GOSUB 2240 : END 48070 ! </pre>

Beschreibung: Zeile 48020: Sperren der Funktionstasten; Falls SPS-Betrieb dann wird Ausgang 7 'Comtac Mode' gesetzt
 Zeile 48030: Löschen der Zeilen 2-4; Anzeige von Programmende in Zeile 1
 Zeile 48040: Abfrage des Passwortes
 Zeile 48050: Aufheben der Maskierungen der Ein-/Ausgänge
 Zeile 48060: Löschen des Displays, Anzeige der Standard-Funktionen der Tasten F1, F2, F3, F4

Beispiel: -

12.6 DT Anzeigesteuerung

Funktion: In dieser Funktion wird das COMTAC als reines Anzeigeelement betrieben. Die Steuerung wird durch digitale Eingänge vorgenommen. Es stehen 100 Texte zur Auswahl, die in der Datei 'SPSTEXT.DAT' hinterlegt sind. Mit Hilfe des DATA-Text-Editors können diese angepaßt werden.

Variablen: ZN = Zeilennummer 1..4
 SN = Stringnummer

Strings: \$(10) = Leerstring
 \$(12)-\$(14) = Strings, die in Zeile 2-4 angezeigt werden

Felder: -

Code:

```

49000 ! DT= Anzeigesteuerung|display control =====
49010 !
49020 GOSUB 50000 : ENABLE ONKEY 15
49030 $(12)=$(10) : $(13)=$(10) : $(14)=$(10) : GOSUB 2000
49040 DO : !
49050 ZN=IN(10~8) : SN=IN(7~1)
49060 UNTIL ZN>1'AND'ZN<5'AND'SN>0'AND'SN<101
49070 $(10+ZN)=$"SPSTEXT"(SN) : GOSUB 2000 : GOTO 49040
49080 !
  
```

Beschreibung: Zeile 49020: Sperren der Funktionstasten; Freigabe von F15 - Programmende
 Zeile 49030: Löschen der Zeilen 2-4
 Zeile 49040-49060: Schleife, daß Zeilennummern nur 2,3,4 und Strings von 1-100 sein dürfen
 Zeile 49070: Anzeige der ausgewählten Stings

Beispiel: -

12.7 E Steuerung SPS - COMTAC

Allgemeines: Über eine digitale E/A-Schnittstelle ist eine SPS mit dem COMTAC verbunden. Die SPS ist dabei Master und steuert das COMTAC. Die SPS gibt Betriebsarten sowie Programm-, Achs- und Positionsnummern vor. Die notwendigen Parameter und Positionen werden im COMTAC programmiert und gespeichert. Das E/A-Handling für Greiferfunktionen, etc. wird von der SPS ausgeführt.

Betriebsarten: Automatikbetrieb
 Handbetrieb

M0 anfahren

Textmeldungen anzeigen

Alle Betriebsarten werden von einem Hauptmenü aus aufgerufen. Das COMTAC gibt entsprechende Meldungen über die aktive Betriebsart aus.

Beschreibung der Betriebsarten:

Textmeldungen: Über eine 3-Bit-Leitung können binär codiert bis zu 10 Texte angewählt werden. Die Texte werden in der Datei CTCT1S1(2).DAT gespeichert und können mit dem DATATEXT-Editor verändert werden. Die Texte werden nur im Automatikmode in der 3. Zeile angezeigt. Somit kann ein Anzeigemodul für die SPS entfallen.

M0 anfahren: Von der SPS aus kann der M0-Punkt aller Achsen angefahren werden. Nach Erreichen des M0-Punktes wird eine Rückmeldung gegeben. Bevor der Start von der SPS ausgegeben wird, muß zuvor eine gültige Achsnummer angelegt werden. Wurde der M0-Punkt aller Achsen angefahren, so wird eine Meldung vom COMTAC an die SPS ausgegeben.

Handfahren: Von der SPS aus können alle Achsen über die Hand+/- Eingänge der COMPAX-Achsen (E2, E3) verfahren werden. Nach Auswahl dieser Betriebsart von der SPS werden diese Eingänge freigegeben. Zusätzlich kann die aktuelle Istposition aller Achsen eingeteacht werden. Diese Werte können im Parametrierbetrieb des COMTAC in eine Positionsnummer gespeichert werden.

Automatik: Von der SPS aus werden Bearbeitungsprogramme, Positions- und Achsnummern vorgegeben. Das COMTAC liest die zuvor programmierten Werte und führt die Positionierungen durch. Durch Rückmeldungen wird der Ablauf kontrolliert. Die Steuerung der digitalen Eingänge/Ausgänge übernimmt die SPS.

12.7.1 EA Automatik über SPS

Funktion: Der Automatikbetrieb wird über die SPS gesteuert. Über Eingänge werden Bearbeitungsprogrammnummern, Achsnummern, Positionsnummern angewählt. Das COMTAC liest die entsprechenden Daten aus den Tabellen und startet die Achsen. Ausgänge für Ventile etc. werden direkt von der SPS aus gesteuert. Das COMTAC überwacht die Positionierungen und meldet Statusinformationen an die SPS zurück.

Variablen:

- AS = Ausgangsnummer
- M0 = Merker Maschinennullpunkt angefahren
- AB = Merker Automatikbetrieb
- A, A1, A2, A3, A4 = Achsnummern
- AA = max. Achsanzahl
- SD = Variable schnelle/langsame Geschwindigkeit
- I = Schleifenzähler
- EG = Eingangsnummer
- FB = Fehlerbit (binär codiert)
- SN = Stringnummer

Strings:

- \$(11)-\$(14) = Anzeigestrings Displayzeile 1-4
- \$(65), \$(91)-\$(97) = Textstrings

Felder:

- PD(..) = Feld, in dem die Werte der Teachpunkte stehen.
- POS(..) = Feld, in dem die Positionsvorgaben im Automatikbetrieb stehen. POS(0)=zu posi-

tionierende Achse binär codiert; POS(zu positionierende Achse)=Positionswert
 SPD(..) = Feld, in dem die Geschwindigkeitsvorgaben im Automatikbetrieb stehen.
 SPD(0)=zu positionierende Achse binär codiert; SPD(zu positionierende Achse)=Geschwindigkeitswert
 ACC(..) = Feld, in dem die Beschleunigungsvorgaben im Automatikbetrieb stehen.
 ACC(0)=zu positionierende Achse binär codiert; ACC(zu positionierende Achse)=Beschleunigungswert
 A(..) = Feld, in dem die Binärwerte der zu positionierenden Achsen stehen
 POSN(..) = Feld, in dem die Positionsnummern für den SPS-Betrieb stehen
 AXP(..) = Feld, in dem die Achsparameter der COMPAX-Achsen stehen

```

Code: 25000 ! EA= AUTOMATIK SPS|automatic mode PLC =====
25010 !
25015 AS=6
25020 IF M0 THEN GOSUB 1600 ELSE GOSUB 1700 : GOSUB 29500 : GOTO 10000
25030 !
25040 AB=TRUE
25050 GOSUB 50000 : CLRLED 1 : CLRLED 2 : CLRLED 3 : CLRLED 4
25060 $(11)=$(26) : $(12)=$(10) : $(13)=$(10) : $(14)="" : GOSUB 2000
25070 ENABLE ONKEY SP : ENABLE ONINP 2
25080 GOSUB 1400 : ! Auto-Speed, Accel, POS() reset
25090 !
25100 FOR A=1 TO AA
25110 SD=0 : SPD(0)=A(A) : ACC(0)=A(A) : ! schnell|fast
25120 SPD(A)=AXP(1+SD*2,A) : ACC(A)=AXP(2+SD*2,A) : GOSUB 1300
25130 NEXT A
25140 !
25150 A=0 : POS(0)=0 : FOR I=1 TO AA:POS(I)=0 : POSN(I)=0 : NEXT I
25160 DO : !
25170 EG=3 : GOSUB 1800 : IF EGS=FALSE THEN GOTO 10000
25180 FB=0 : FB=(2**0)*((CPXINP(A1)@6)/65535)+(2**1)*((CPXINP(A1)@7)/65535)
25190 FB=FB+(2**2)*((CPXINP(A1)@8)/65535)
25200 IF FB=0 THEN GOSUB 2000 ELSE SN=(90+FB) : GOSUB 2230
25205 GOSUB 2000
25210 GOSUB 29000
25220 GOSUB 29200
25230 WHILE A=0'OR'POSN(A)=0'OR'BP=0'OR'IN(4)=0
25240 !
25250 FOR I=1 TO AA
25260 IF POSN(I)<>0 THEN POS(I)=PD(POSN(I),I)
25270 NEXT I
25280 !
25290 EG=9 : GOSUB 1800 : IF EGS<>FALSE THEN GOSUB 26000 : GOTO 25380
25300 !
25310 SD=IN(8) : IF SD=SD_ THEN GOTO 25380
25320 SD_=SD : SPD(0)=POS(0) : ACC(0)=POS(0) : ! schnell,langsam|fast,slow
25330 FOR I=1 TO AA
25340 IF POSN(I)<>0 THEN SPD(I)=AXP(1+SD*2,I) : ACC(I)=AXP(2+SD*2,I)
25350 NEXT I
25360 GOSUB 1300
25370 !
25380 GOSUB 1200 : ! Start
25390 DO : !
25400 $(14)=$(65)
25410 DISP TABXY(30,4),"A",VAL$(A),":",CPXPOS(A)
25420 GOSUB 2000
25430 ON AA GOTO 25430,25440,25450,25460,25470
25440 UNTIL WAITPOS A1 : POS(0)=0 : $(14)="..." : GOTO 25480
25450 UNTIL WAITPOS A1,A2 : POS(0)=0 : $(14)="..." : GOTO 25480
25460 UNTIL WAITPOS A1,A2,A3 : POS(0)=0 : $(14)="..." : GOTO 25480
25470 UNTIL WAITPOS A1,A2,A3,A4 : POS(0)=0 : $(14)="..." : GOTO 25480
25480 AS=2 : GOSUB 1500 : $(12)[36,39]="0000" : GOSUB 2000
25490 !
25500 GOTO 25150
25510 !

```

Beschreibung: Zeile 25015-25020: Ausgang 6 setzen falls Maschinennullpunkt angefahren wurde, ansonsten Ausgang 6 zurücksetzen und Fehlerausgang 3 setzen.

Zeile 25040: Automatikbetrieb wird TRUE gesetzt.
 Zeile 25050: Funktionstasten sperren, LED's löschen.
 Zeile 25060: Strings der Displayzeilen 1-4 zuweisen und anzeigen.
 Zeile 25070: STOP-Interrupt freigeben.
 Zeile 25080: Tabellen zurücksetzen.
 Zeile 25100-25130: Zuweisen der Geschwindigkeiten und Rampen aller Achsen.
 SD=0 : Schnelle Geschwindigkeit
 SPD(0)=A(A); ACC(0)=A(A): In einer Schleife wird das Bit jeder Achse in die Tabellen für Geschwindigkeit und Rampe geschrieben. Achse 1=2⁰; Achse 2=2¹
 SPD(A)=AXP(1+SD*2,A): Aus der Achsparametertabelle wird der Wert in die Geschwindigkeitstabelle geschrieben.
 ACC(A)=AXP(2+SD*2,A): Aus der Achsparametertabelle wird der Wert in die Rampentabelle geschrieben.
 Zeile 25150: In einer Schleife werden die Tabellen der Positionen und Positionsnummern gelöscht.
 Zeile 25160-25230: In einer Schleife wird solange gewartet, bis gültige Werte der Achsnummer, Positionsnummer, Bearbeitungsprogrammnummer vorliegen.
 Zeile 25170: Liegt Eingang 3 'Automatik' nicht mehr an, so wird ins Hauptmenü verzweigt.
 Zeile 25180-25200: Die Eingänge 7,8,9 der COMPAX-Achse 1 werden abgefragt. Binär codiert werden Fehlertexte der SPS am COMTAC-Display angezeigt.
 Zeile 25210: Im Unterprogramm 29000 wird die Nummer des Bearbeitungsprogrammes abgefragt.
 Zeile 25220: Im Unterprogramm 29200 wird die Achsnummer, Positionsnummer und der Start abgefragt.
 Zeile 25250-25270: In einer Schleife werden abhängig von der Positionsnummer die Positionsdaten aus der Tabelle PD(..) in die Tabelle POS(..) geschrieben.
 Zeile 25290: Ist Eingang 9 vorhanden, so werden 2 Achsen für einen gemeinsamen Start vorbereitet.
 Zeile 25310-25360: Über den Eingang 8 wird schnelle bzw. langsame Geschwindigkeit ausgewählt. Die notwendigen Tabellen werden beschrieben.
 Zeile 25380: Die ausgewählten Achsen werden gestartet.
 Zeile 25390-25470: Die Meldung 'Position Erreicht' der ausgewählten Achsen wird abgefragt. Während der Positionierung wird die Istposition angezeigt.
 Zeile 25480: Ausgang 2 für Automatikposition erreicht wird an die SPS gesendet.

Beispiel:

-

12.7.2 EB Handbetrieb über SPS

Funktion:	Von der SPS aus können die Achsen durch Anlegen der COMPAX-Eingänge Hand+/-manuell verfahren werden. Es ist eine schnelle Handgeschwindigkeit, -rampe oder eine langsame Handgeschwindigkeit, -rampe möglich. Zusätzlich kann die momentane Istposition geteacht werden.
Variablen:	EB = Merker für schnelle/langsame Handgeschwindigkeit, - rampe SN = Stringnummer A = Achsnummer PN = Parameternummer der COMPAX-Achsen N = Laufvariable AS = Ausgangsnummer EGS = Eingangsstatus
Strings:	\$(28) = Textstring
Felder:	AXP(..) = Feld, in dem die Achsparameter der COMPAX-Achsen stehen POS(..) = Feld, in dem die Positionsvorgaben im Automatikbetrieb stehen. POS(0)=zu positionierende Achse binär codiert; POS(zu positionierende Achse)=Positionswert; In dieser Funktion wird die Tabelle zum Zwischenspeichern der geteachten Position genutzt.

```

Code:      27000 ! EB= CPX-HandFunktionen SPS|manual +/- PLC =====
           27010 !
           27020 GOSUB 50000 : EB=-1
           27030 FOR A=1 TO AA:CPXIMASK(A)=111111111111001B:CPXOMASK(A)=0 : NEXT A
           27040 SN=28 : GOSUB 2600 : GOSUB 2104
           27050 DO : !
           27060 EG=5 : GOSUB 1800
           27070 !
           27080 IF EB=IN(8) THEN GOTO 27160
           27090 FOR A=1 TO AA
           27100 PN=5 : IF IN(8)=1 THEN PW=AXP(6,A) ELSE PW=AXP(5,A)
           27110 GOSUB 50500
           27120 PN=9 : IF IN(8)=1 THEN PW=AXP(8,A) ELSE PW=AXP(7,A)
           27130 GOSUB 50500
           27140 NEXT A:EB=IN(8)
           27150 !
           27160 IF IN(16)=0 THEN GOTO 27230 : ! Teachen
           27170 IF M0=FALSE THEN GOSUB 29500 : GOTO 27230
           27180 FOR N=1 TO AA: OUTPUT 1,N"S1"
           27190 POS(N)=VAL($(#1)[7]) : NEXT N : AS=4 : GOSUB 1600
           27200 !
           27210 IF IN(16)=1 THEN GOTO 27200 ELSE AS=4 : GOSUB 1700
           27220 !
           27230 WHILE EGS<>FALSE
           27240 FOR A=1 TO AA:CPXIMASK(A)=111111111111111B:CPXOMASK(A)=255 :
           NEXT A
           27250 GOTO 10000
           27260 !

```

Beschreibung:

Zeile 27020: Die Funktionstasten werden gesperrt; Merker für schnell/langsam Fahrt wird so gesetzt, daß zum ersten Mal jedenfalls die COMPAX-Parameter beschrieben werden.

Zeile 27030: Die Eingänge Hand+/- der COMPAX-Achsen werden maskiert, sodaß sie von der SPS aus gesetzt werden können.

Zeile 27040: Textanzeige

Zeile 27050-27230: Schleife, in der solange verweilt wird, bis der Handbetrieb von der SPS aus abgewählt wird.

Zeile 27060: Eingang Handbetrieb von der SPS aus wird eingelesen.

Zeile 27080-27140: In Abhängigkeit des Einganges 8 werden die COMAPX-Parameter für schnelle/langsame Handfahrt beschrieben.

Zeile 27160: Abfrage des Einganges zum Teachen der momentanen Istposition.

Zeile 27170: Abfrage des Maschinennull-Status. Nur bei angefahrenem M0-Punkt, kann geteacht werden.

Zeile 27180-27190: In einer Schleife werden die Istpositionen aller COMPAX-Achsen eingelesen. Die Istposition steht im Status 1. Es wird nur der Wert ohne Maßeinheit eingelesen.

Zeile 27210: Abfrage des Einganges zum Teachen, der nach der Bestätigung des Einlesens (Ausgang 4) wieder zurückgesetzt werden muß.

Zeile 27240: Alle Eingänge werden wieder maskiert, sodaß die SPS nicht mehr auf die Eingänge zugreifen kann.

Beispiel:

-

12.7.3 EC M0 anfahren über SPS

Funktion: Von der SPS aus können die Maschinennullpunkte aller Achsen angefahren werden. Durch Anlegen der entsprechenden binärcodierten Eingänge werden die Achsen selektiert. Das COMTAC meldet das Erreichen der M0-Punkte zurück.

Variablen:

- SN = Stringnummer
- A = Achsnummer
- AA = max. Achsanzahl
- AS = Ausgangsnummer

Strings:	\$(30), \$(70) = Anmzeigestrings
Felder:	-
Code:	<pre> 28000 ! EC= Referenzfahrt manuell SPS zeropoint manual PLC ===== 28010 ! 28020 SN=30 : GOSUB 2600 : ENABLE ONINP 2 28030 GOSUB 50000 28040 ! 28050 A=0 : A=IN(10~8) : IF A>AA'OR'A<1 THEN GOSUB 29500 : GOTO 28190 28060 ! 28070 ! M0 mit Achse A zeropoint with axis A 28080 AS=8 : GOSUB 1700 28090 SN=70 : GOSUB 2240 28100 SN=30 : GOSUB 2230 : DISP "..." 28110 \$(1)="PH" : GOSUB 1000 : WAIT WT 28120 DO : ! 28130 UNTIL (CPXSTS(A)@2) 28140 SN=66 : GOSUB 2230 : DISP "..." 28150 \$(1)="PA0" : GOSUB 1000 : WAIT WT 28160 DO : ! 28170 UNTIL (CPXSTS(A)@4) : AS=1 : GOSUB 1500 28180 ! 28190 GOTO 10000 28200 ! </pre>
Beschreibung:	<p>Zeile 28020: Textanzeige. der STOP-Eingang wird freigegeben.</p> <p>Zeile 28030: Die Funktionstasten werden gesperrt.</p> <p>Zeile 28050: Einlesen der Achsnummer</p> <p>Zeile 28080: M0-angefahren-Signal wird zurückgesetzt.</p> <p>Zeile 28090-28100: Textanzeige</p> <p>Zeile 28110: Start der Nullpunktfahrt</p> <p>Zeile 28120-28130: Warten auf M0-Punkt erreicht</p> <p>Zeile 28140: Textanzeige</p> <p>Zeile 28150: Start zum Realnullpunkt der Achsen</p> <p>Zeile 28160-28170: Warten auf R0-Punkt erreicht</p>
Beispiel:	-

12.7.4 ED Bearbeitungsprogramm über SPS einlesen

Funktion:	Von der SPS aus wird binär codiert die Nummer eines Bearbeitungsprogrammes angelegt. In jedem Bearbeitungsprogramm sind Maschinenpositionen abgelegt.
Variablen:	<p>EG = Eingangsnummer</p> <p>EGS = Eingangsstatus</p> <p>AS = Ausgangsnummer</p> <p>BP = Bearbeitungsprogrammnummer</p> <p>N, I = Laufvariable</p> <p>AA = max. Achsanzahl</p>
Strings:	\$(84) = Textstring
Felder:	PD(..) = Feld, in dem die Positionen abgelegt sind
Code:	<pre> 29000 ! ED= Bearbeitungsprogramm einlesen read program = 29010 ! 29020 EG=7 : GOSUB 1800 : IF EGS=FALSE THEN GOTO 29150 29030 AS=4 : GOSUB 1700 29040 BP=0 : BP=IN(14~8) : IF BP>127'OR'BP<1 THEN BP=0 </pre>


```

29050 IF BP=0 THEN GOSUB 29500 : GOTO 29140
29060 ! Maschinen-Positionen einlesen|read machine positions
29070 FOR N=1 TO AA: ! Alle Achsen
29080 FOR I=1 TO MPO : ! Position 1 bis MPO
29090 PD(I,N)=|;$ (16) (((BP-1)*MPO*AA)+((N-1)*MPO)+I) : ! Position lesen
29100 NEXT I
29110 NEXT N
29120 !
29130 IF BP<>0 THEN AS=4 : GOSUB 1500 : ! program OK
29140 $(12)=$(84) : $(12)[10,12]=VAL$(BP) : GOSUB 2000
29150 RETURN
29160 !

```

Beschreibung: Zeile 29020: Eingang Bearbeitungsprogramm einlesen wird abgefragt.
 Zeile 29030: Ausgang Bearbeitungsprogrammnummer OK wird zurückgesetzt.
 Zeile 29040: Einlesen der Bearbeitungsprogrammnummer
 Zeile 29050: Falls Bearbeitungsprogrammnummer falsch, Fehlerausgabe
 Zeile 29070-29110: Laden der Positionen aus der Datei PAR.DAT in das Feld PD(..).
 Zeile 29130: Ausgang Bearbeitungsprogrammnummer OK wird gesetzt.
 Zeile 29140: Anzeige der Bearbeitungsprogrammnummer.

Beispiel: -

12.7.5 EE Warten auf Start der SPS

Funktion: Von der SPS aus werden Achsnummern, Positionsnummern binärcodiert angelegt und vom COMTAC eingelesen. Wurden die Nummern bestätigt, kann die SPS den START für die Positionierung anlegen.

Variablen: EG = Eingangsnummer
 EGS = Eingangsstatus
 AS = Ausgangsnummer
 A = Achsnummer
 AA = max. Achsanzahl
 BP = Bearbeitungsprogrammnummer
 MPO = max. Maschinenposition

Strings: \$(84) = Textanzeige

Felder: POS(..) = Feld, in dem die Positionsvorgaben im Automatikbetrieb stehen. POS(0)=zu positionierende Achse binär codiert; POS(zu positionierende Achse)=Positionswert
 POSN(..) = Feld, in dem die Positionsnummern für den SPS-Betrieb stehen

Code: 29200 ! EE= Warten AchsNr,PositionsNr,START|wait for axisNo,positionNo,START==
 29210 !
 29220 EG=16 : GOSUB 1800 : IF EGS=FALSE THEN GOTO 29410
 29230 AS=5 : GOSUB 1700
 29240 !
 29250 DO : !
 29260 UNTIL IN(16)
 29270 A=IN(10~8) : IF A>AA'OR'A<1 THEN A=0 : GOSUB 29500 : GOTO 29410
 29280 POS(0)=POS(0)'OR'A: IF POS(0)>((2**AA)-1)'OR'POS(0)<1 THEN POS(0)=0
 29290 POSN(A)=IN(15~11) : IF POSN(A)>MPO'OR'POSN(A)<1 THEN POSN(A)=0
 29300 \$(12)=\$(84) : \$(12)[10,12]=VAL\$(BP) : \$(12)[21]=VAL\$(A)
 29310 \$(12)[32,34]=VAL\$(POSN(A))
 29320 IF (POS(0)@0)=TRUE THEN \$(12)[39]="1" ELSE \$(12)[39]="0"
 29330 IF (POS(0)@1)=TRUE THEN \$(12)[38]="1" ELSE \$(12)[38]="0"
 29340 IF (POS(0)@2)=TRUE THEN \$(12)[37]="1" ELSE \$(12)[37]="0"
 29350 IF (POS(0)@3)=TRUE THEN \$(12)[36]="1" ELSE \$(12)[36]="0"
 29360 GOSUB 2000
 29370 IF A<>0'AND'POSN(A)<>0 THEN AS=5 : GOSUB 1500
 29380 DO : !

```

29390 WHILE IN(16)
29400 !
29410 RETURN
29420 !

```

Beschreibung: Zeile 29220: Einlesen des Einganges Positions-/Achsnnummer übernehmen
 Zeile 29230: Zurücksetzen des Ausganges Positions-/Achsnnummer OK.
 Zeile 29250-29260: Warten bis Eingang Positions-/Achsnnummer übernehmen aktiv ist.
 Zeile 29270: Einlesen der Achsnnummer
 Zeile 29280: Schreiben des Bits der angewählten Achse in das Feld POS(0).
 Zeile 29290: Einlesen der Positionsnummer.
 Zeile 29300-29310: Anzeige des Bearbeitungsprogrammes, Achsnnummer, Positionsnummer.
 Zeile 29320-29360: Anzeige der Bits des Feldes POS(0), der angewählten Achsen.
 Zeile 29370: Setzen des Ausganges Positions-/Achsnnummer OK.
 Zeile 29380-29390: Warten bis der Eingang Positions-/Achsnnummer zurückgesetzt ist.

Beispiel: -

12.7.6 EF Fehlerausgang setzen an SPS

Funktion: Werden nicht korrekte Bearbeitungsprogramm-, Positions-, Achsnummern angewählt, so wird der Ausgang 3 zur SPS gesetzt, um den Fehlerzustand anzuzeigen.

Variablen: AS = Ausgangsnummer

Strings: -

Felder: -

```

Code: 29500 ! EF= Fehlerausgang setzen|set error-output =====
      29510 !
      29520 AS=3 : GOSUB 1500 : WAIT 2000 : RETURN
      29530 !

```

Beschreibung: Zeile 29520: Ausgang 3 wird als Impuls gesetzt.

Beispiel: -

12.7.7 F Optimierte 2-Achspositionierung

12.7.7.1 FA Bahnpositionierung mit 2 Achsen

Funktion:	Bei Verwendung der COMPAX X00-Positioniermodule sind keine gekoppelten Positionierungen (Linearinterpolationen) möglich. Diese Funktion leistet Abhilfe, indem Werte, die für eine Linearinterpolation benötigt werden, berechnet werden. Beim nachfolgenden Aufruf der Unterprogramme 'AC Achs-Positionierungen starten' und 'AD Speed und Rampen setzen', werden die berechneten Werte den angewählten Achsen zugewiesen und die Achsen nacheinander gestartet. Deshalb ist hier nur eine Quasi-Linearinterpolation zu realisieren.
Variablen:	I = Schleifenzähler AA = max. Achsanzahl
Strings:	-
Felder:	LA(0) == Anzahl Linearachsen LA(1), LA(2) = Feld, in dem die an der Positionierung beteiligten Achsnummern stehen VA(..) = Absolutgeschwindigkeiten der Achsen in m/s VB(..) = Bahngeschwindigkeiten der Achsen in m/s VBM(..) = max. Bahngeschwindigkeiten der Achsen in m/s BSM(..) = max. Beschleunigungen der Achsen in m/s ² BS(..) = Beschleunigungen der Achsen in m/s ² WEG(..) = realtiver Weg, den die Achsen positionieren in mm AL(..) = Bahnrampen der Achsen in ms
Code:	Math. Funktionen, die hier nicht näher erläutert werden.
Beschreibung:	-
Beispiel:	20520, 24500-24590

12.7.8 FB Lade Konfiguration

Funktion:	Um die physikalischen Werte der COMPAX-Achsen zu bestimmen, werden in dieser Funktion die Konfigurationsdaten eingelesen. Diese Daten werden verrechnet und daraus die Absolutgeschwindigkeiten berechnet. Diese werden benötigt, um Grenzwerte zu überprüfen.
Variablen:	A = Achsnummer AA = max. Achsanzahl PN = Parameternummer
Strings:	-
Felder:	P80(..)-P84(..), P85(..), P103(..), P104(..), P106(..) = Felder, in denen die Konfigurationsdaten der Achsen stehen.
Code:	26800 ! FB= Lade Konfiguration load configuration ===== 26805 ! 26810 FOR A=1 TO AA 26815 PN=80 : GOSUB 50400 : P80(A)=PW 26820 PN=81 : GOSUB 50400 : P81(A)=PW 26825 PN=82 : GOSUB 50400 : P82(A)=PW 26830 PN=83 : GOSUB 50400 : P83(A)=PW 26835 PN=85 : GOSUB 50400 : P85(A)=PW

```
26840 PN=103 : GOSUB 50400 : P103(A)=PW
26845 PN=104 : GOSUB 50400 : P104(A)=PW
26850 PN=106 : GOSUB 50400 : P106(A)=PW
26855 IF P80(A)=2 THEN VA(A)=P104(A)*P83(A)/(P85(A)*60*1000)
26860 IF P80(A)=4 THEN VA(A)=P104(A)*P82(A)*P83(A)/(P85(A)*60*1000)
26865 IF P80(A)=8 THEN VA(A)=P104(A)*P82(A)*P83(A)/(P85(A)*60*1000)
26870 IF P80(A)=16 THEN VA(A)=P104(A)*P83(A)/(60*1000*1000)
26875 NEXT A
26880 RETURN
26885 !
```

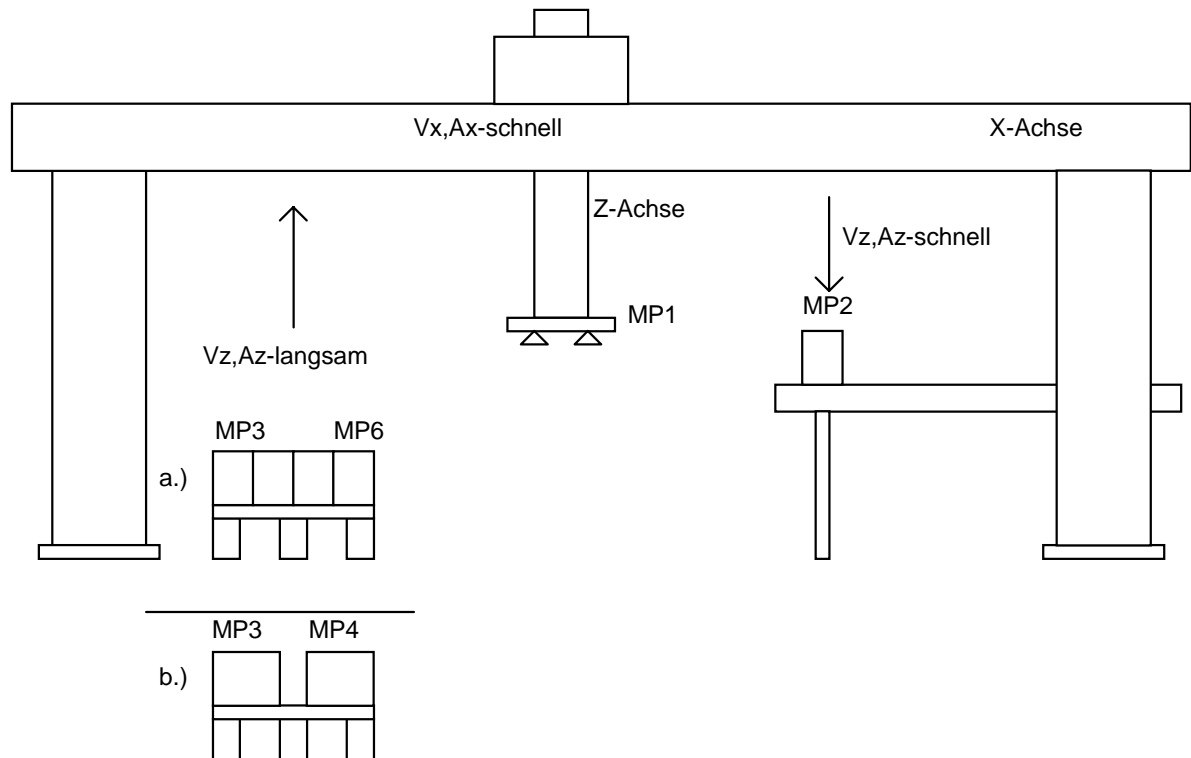
Beschreibung: Zeile 26810-26875: In einer Schleife werden alle Konfigurationsdaten der Achsen eingelesen.

Beispiel: Siehe Zeile 63390

13. Applikationsbeispiel:

2-achsige Entpalettieranlage

Aufbau:



Legende:

- a.) Bearbeitungsprogramm 1 (Produkt 1 wird entpalettiert)
- b.) Bearbeitungsprogramm 2 (Produkt 2 wird entpalettiert)

MPx = Maschinenpositionen getrennt für Bearbeitungsprogramm 1 und 2

Vx,Ax,Vz,Az = Achsparameter = Geschwindigkeiten und Rampen der Achsen X,Z

Aufgabe:

Mit einer Entpalettieranlage sollen Kartons von Euro-Paletten entnommen und auf einem Rollenband abgesetzt werden. Es werden 2 Kartontypen entpalettiert, weshalb auch 2 Bearbeitungsprogramme benötigt werden. Jedes Bearbeitungsprogramm enthält max. 10 Maschinenpositionen. Der Automatikablauf ist unabhängig von der Anzahl der Bearbeitungsprogramme. Es gibt 2 Automatikgeschwindigkeiten für schnelle und langsame Fahrt. Als Maschinenparameter werden eine Saugzeit zum Aufnehmen der Kartons und eine max. Drehzeit des Drehgreifers programmiert.

Realisierung:

1. Aufruf des Programmes "CTCTEACH.BAS" und eingeben der Randbedingungen:

- CT.(101) = Dialogsprache = Deutsch/Englisch
- CT.(102) = Anzahl der Achsen = 2
- CT.(103) = Anzahl der Maschinenpositionen = 10
- CT.(104) = Anzahl der Bearbeitungsprogramme = 2
- CT.(105) = Anzahl der Maschinenparameter = 10
- CT.(106) = Anzahl der Achsparameter = 10
- CT.(107) = Automatikmodus (CTC/SPS) = 0

2. Eingabe der maschinenabhängigen Texte in der Datei "PART1S1(2).DAT" mit dem DATA-Text-Editor:

- Maschinenpositionen Zeile 70.. 100
- Maschinenparameter Zeile 26..69

3. Eingabe der Ausgangsbezeichnungen in der Datei "OUTT1S1(2).DAT" mit dem DATA-Text-Editor:

- Ausgangsbezeichnungen ab Zeile 1

4. Umsetzen in Code: (dient nur als Beispiel)

```
20350 !
20360 ! Schnell : Achse 1 : PositionsNr. 1
20370 SD=0 : A(0)=A(2) : POSN(A(2))=1 : GOSUB "STRECKENPOSITIONIERUNG"
20380 !
```

Legende: SD = Variable für Automatikgeschwindigkeit schnell=0 / langsam=1
 A(0)= Binärwert der zu positionierenden Achsen
 A(2) = 0010B, damit auch A(0)=0010B
 POSN(A(2)) = Positionsnummer, die mit dieser Achse angefahren werden soll

```
20390 ! Schnell : Achse 2 : PositionsNr. 1
20400 SD=1 : A(0)=A(1) : POSN(A(2))=1 : GOSUB "STRECKENPOSITIONIERUNG"
20410 !
```

Legende: SD = Variable für Automatikgeschwindigkeit schnell=0 / langsam=1
 A(0)= Binärwert der zu positionierenden Achsen
 A(1) = 0001B, damit auch A(0)=0001B
 POSN(A(1)) = Positionsnummer, die mit dieser Achse angefahren werden soll

Tabelle A(..): A(1) = 1 = 0001B A1 = 1
 A(2) = 2 = 0010B A2 = 2
 A(3) = 4 = 0100B A3 = 3
 A(4) = 8 = 1000B A4 = 4
 A(12) = 3 = 0011B
 A(13) = 5 = 0101B
 A(14) = 9 = 1001B
 A(23) = 6 = 0110B
 A(24) = 10 = 1010B
 A(34) = 12 = 1100B

Dateiorganisation:

MAP.DAT	<p>Datei im ZP-Ram, in dem Maschinenparameter und -positionen gespeichert sind; zugehörige Tabelle im Arbeitspeicher: KP(I); I=Nummer des Maschinenparameters PD(I,A); I=Nummer der Maschinenposition, A= Achse Bei mehreren Bearbeitungsprogrammen muß jedesmal die Tabelle PD(..) geladen werden (siehe Seite 45) !</p>
AXP.DAT	<p>Datei im ZP-Ram, in dem Achsparameter gespeichert sind; zugehörige Tabelle im Code: AXP(I,A); I= Nummer der Achsparameter, A= Achse</p>

Dateistruktur: MAP.DAT (Daten)

PART1S1(2).DAT(Texte)

Speicherstelle 1			Zeile 70
..	Achse 1		
Speicherstelle 10		Bearbeitungsprogramm 1	Zeile 79
Speicherstelle 11	..		Zeile 70
	Achse 2		
Speicherstelle 20			Zeile 79
Speicherstelle 21			Zeile 70
..	Achse 1		
Speicherstelle 30		Bearbeitungsprogramm 2	Zeile 79
Speicherstelle 31	..		Zeile 70
	Achse 2		
Speicherstelle 40			Zeile 79
...			...
600			Zeile 100
Speicherstelle 601			Zeile 26
	Maschinenparameter		...
Speicherstelle 645			Zeile 69



Die Positionen werden in Tabellen umgespeichert und sind über Positionsnummern anwählbar (siehe Seite 39).

Dateistruktur: AXP.DAT (ZP-Ram)

PART1S1(2).DAT(Texte)

Speicherstelle 1		Zeile 1
	Achse 1 Achsparameter	..
Speicherstelle 25		Zeile 25
Speicherstelle 26		Zeile 1
	Achse 2 Achsparameter	..
Speicherstelle 50		Zeile 25
Speicherstelle 51		Zeile 1
	Achse 3 Achsparameter	..
Speicherstelle 75		Zeile 25
Speicherstelle 76		Zeile 1
	Achse 4 Achsparameter	..
Speicherstelle 100		Zeile 25



Die Positionen werden in Tabellen umgespeichert (siehe Seite 39).

14. Stichwortverzeichnis

2-achsige Entpalettieranlage	101	Display anzeigen mit		Maschinenpositionen bear-	
60000	69	\$(11)..\$(14	49	beiten	24
Achseingabe	65	Display Initialisierung	56	Maßsystem korrigieren	29
Achsparemeter		EG - Maschinenrichtlinien	3	Maximale Geschwindigkeit	26
Modul	81	Ein- und Ausgabe	49	Minimale Rampenzeit	26
Achsparemeter bearbeiten	25	Eingabe Bearbeitungspro-		Modul: Autostart	40
Achs-Positionierung starten	54	gramm-Nummer	44	Modul: Autostart_2Achsen	41
Anlagen-Info	32	Eingang einlesen	47	Notaus-Funktion	13
Modul	87	Eingang einlesen mit Zeitab-		NOTAUS-Routine	60
Anzeigesteuerung		frage	48	Nullpositionen	22
Modul	91	Einrichtbetrieb	32	Optimierte 2-	
Applikationsbeispiel	101	Modul	90	Achspositionierung	
Aufbau des COMTAC -		Einsatzgebiet	4	Module	99
COMPAX - Verbund	11	Einschalten von COMTAC		OUTT1S1	15
Ausgang als Impuls setzen	45	und COMPAX	21	Parameter	39
Ausgang löschen	46	Entwicklungsschritte	6	Parameter laden	
Ausgänge setzen	45	Fehleranzeige und weiter mit		Modul	77
Ausschalten der Anlage	22	ENTER	64	Parametrier-Betrieb	
Auto-Beschl. langsam	25	Fehlerausgang setzen an		Modul	77
Auto-Beschl. schnell	25	SPS		Parametrierbetrieb	23
Auto-Geschw. langsam	25	Modul	98	PARTS1T1	15
Auto-Geschw. schnell	25	Fehler-Routine	57	Passwort Freigabe in Achse	
Automatik über SPS		Funktionstasten freigeben	71	A	62
Modul	92	Funktionstasten sperren	70	Passwort Sperren in Achse A	62
Automatik-Betrieb		Funktionstasten zuweisen	71	Passwort-Ebene	21
Modul	74	Gerätezuordnung	3	Passwort-Eingabe	
Automatikbetrieb	39	Geschwindigkeit und Ram-		Modul	88
Automatik-Betrieb / Achs-		pen setzen	54	POS-Feld Reset	55
steuerung	40	Hand-Beschl. langsam	26	Positionen	39
Automatik-Betrieb Programm	42	Hand-Beschl. schnell	26	Positionen laden	
Bearbeitungsprogramm laden	45	Handbetrieb	26	Modul	77
Bearbeitungsprogramm über		Hand-Betrieb		Positionierung mit 2 Achsen	
SPS einlesen		Modul	82	Modul	99
Modul	96	Handbetrieb über SPS		Programmende	
Betriebsarten	21; 73	Modul	94	Modul	90
COMPAX - Adressen	14	Hand-Geschw. langsam	26	Programm-Ende	32
COMPAX-Fehler-Routine	58	Hand-Geschw. schnell	26	Programmzeile	
COMPAX-Parameter-		Hauptmenü		1000	53
Manager		Modul	73	10000	73
Modul	86	Hauptprogramm	20	1100	53
COMTAC / COMPAX Ein-		Modul	67	1200	54
gänge lesen	47	Hauptprogramm und Initiali-		1300	55
COMTAC als Achssteuerung		sierung	67	1400	55
über SPS	36	INPUT	49	1500	45
COMTAC als Textanzeige-		Installation	8	1600	46
einheit	38	Konfiguration von COMPAX	14	1700	46
COMTAC-Initialisierung		Korrektur Maßbezug	28	1800	47
Modul	86	Modul	85	1900	48
COMTAC-Konfiguration	10	Lade Konfiguration		2000	49
CPX-Parameter auslesen	63	Modul	99	20000	42; 74
CPX-Parameter beschreiben	64	M0 anfahren über SPS		2100	56
CPX-Parameter-Manager	29	Modul	95	24000	40
CPX-Status auslesen	63	M0 manuell	27	25000	93
CTC-Initialisierung	32	Maschinen-Nullpunkt manuell		26800	99
CTCT1S1	17	Modul	83	2700	57
Dateibefehle	65	Maschinenparameter	24	27000	95
Dateistruktur	9	Modul	78	28000	96
Dateistruktur: AXP.DAT	103	Maschinenparameter ändern	23	29000	96
Dateistruktur: MAP.DAT	103	Maschinenpositionen	25	29200	97
		Modul	79	29500	98
				3000	57
				30500	77
				30900	45
				35000	44
				4000	58
				45000	87

46000.....	87
4690.....	60
47000.....	90
48000.....	90
49000.....	91
4990.....	61
50000.....	71
50100.....	71
50200.....	72
50300.....	62
50350.....	62
50400.....	63
50450.....	63
50500.....	64
50600.....	64
51000.....	65
52000.....	49
52100.....	66
52200.....	88
52700.....	66
61000.....	69
63000.....	67
RS485-OUTPUT	53
RS485-OUTPUT/ENTER.....	53
Softwareversion	1
Betriebsarten umschalten	35
SPS-Programmierer	33
Steuerung SPS - COMTAC	
Modul	91
Stop-Funktion	13
STOP-Routine	61
String-Dimensionierung	69
String-Initialisierung	69
Systemparameter	10
Test E/A's.....	28
Test Eingänge/Ausgänge	
Modul	84
Textauswahl über die	
COMTAC - Eingänge.....	38
Texte anpassen	15
Variablen und Tabellen	51
Variablen-Initialisierung.....	70
Verfahrrichtungen	22
Voraussetzungen.....	8
Warten auf Start der SPS	
Modul	97
Weiter mit ENTER	66
Weitere Dokumentationen	3
Zeilen löschen.....	56
Zielgruppen.....	4
Zugriff auf Parameter und	
Positionen	39

