



Stögra Antriebstechnik GmbH

Machtlfinger Straße 24
D-81379 München

Tel.: (089)15904000
Fax.: (089)15904009
Email : info@stoegra.de
Internet URL : www.stoegra.de

SERS Programmier 2

Handprogrammiergerät für SERS-Steuerungen

Handbuch

Ausgabe November 2011

Änderungen, die der Verbesserung dienen, bleiben vorbehalten.

Bei der Erstellung von Texten und Bildern wurde mit höchster Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Für fehlerhafte Angaben und deren Folgen können wir keine Haftung übernehmen.

Inhaltsverzeichnis

	Seite
1. Allgemeines.....	4
1.1 SERS Programmer 2 und Vorgängerversion SERS Programmer.....	4
1.2 Standardmodus und erweiterter Programm-Modus.....	4
1.3 Tastatur im erweiterten Programm-Modus.....	4
1.4 Tastatur im Standardmodus.....	5
1.5 Anschluß SERS-Programmer an SERS-Steuerung.....	6
1.6 Anschluß SERS-Programmer an Windows-PC im Programmiermodus.....	6
1.7 Tastenbelegung im Standardmodus.....	7
2. Betrieb und Programmierung im Standardmodus.....	8
2.1 Nach dem Einschalten.....	8
2.2 Allgemeines zu den Menüs.....	9
2.3 Hauptmenü.....	9
2.4 Adressen / Sprachmenü.....	9
2.5 START Programm.....	10
2.6 START Alle.....	10
2.7 MANUELL Mode Menü.....	11
2.8 PARAMETER Menü.....	11
2.9 PARAMETER Maske.....	12
2.10 RESET Menü.....	14
2.11 HANDVERFAHREN Menü.....	14
2.12 REFERENZFAHREN Menü.....	15
2.13 TERMINAL Mode.....	15
2.14 PROGRAM Mode Menü.....	16
2.15 PROGRAM EDIT Maske.....	17
2.16 PROGRAM NEU Maske.....	18
2.17 PROGRAM TEST Maske.....	19
2.18 Zusätzliche Zeichen im Terminalmodus und bei EDIT/NEU Programm.....	19
2.19 Menü Baumstruktur im Standardmodus.....	20
3. Betrieb und Programmierung im erweiterten Programm-Modus.....	21
3.1 Programmiermodus.....	21
3.2 Betrieb im erweiterten Programmmodus.....	22
3.3 Syntax erweiterter Programm-Modus.....	23
3.3.1 Syntax Programmierung Anzeigemenü.....	23
3.3.2 Syntax Programmierung Ablaufprogramm.....	24
3.3.3 Zieladresse bei Steuerzeichen, Parameter und Abfragen.....	27
3.4 Beschränkungen für E ² Prom-Programme.....	27
3.5 Funktionen für die Programmierung von Ablaufprogrammen.....	29
3.5.1 IF : Bedingte Ausführung.....	29
3.5.2 Label : Programm-Marken.....	31

Inhaltsverzeichnis

	Seite
3.5.3 GOTO, GT : Verzweigung.....	31
3.5.4 GOSUB, GS: Unterprogrammaufruf.....	31
3.5.5 RETURN, RT : Unterprogrammabschluss.....	31
3.5.6 Programmierung von Positioniervorgängen.....	32
3.5.7 Positionieren mit Geschwindigkeitsprofilen, Polynomfahren.....	33
3.5.8 WAIT Anweisung.....	35
3.5.9 “ : “ Anweisung - Handfahren bis STOP durch Eingang.....	35
3.5.10 Arithmetik / Rechnen mit der SERS.....	36
3.6 Parameter für erweiterten Programm-Modus.....	37
3.7 Tastenwerte bei Tastaturabfrage.....	41
3.8 Vordefinierte Menüs.....	42
3.9 Hinweise zur Programmierung im erweiterten Programm-Modus.....	45
3.10 Programmbeispiele für erweiterten Programm-Modus.....	46
4. Abmaße.....	47
5. Technische Spezifikationen.....	47

1. Allgemeines

1.1 SERS Programmer 2 und Vorgängerversion SERS Programmer

Der SERS Programmer 2 beinhaltet im Vergleich zur Vorgängerversion SERS Programmer (und SERS Programmer Version R) zusätzlich einen Programmiermodus, und einen erweiterten Programm-Modus. Damit können beliebige Anzeigemenüs programmiert werden, und ein komplettes Ablaufprogramm (zur Ansteuerung von SERS-Steuerungen) in den SERS Programmer 2 abgespeichert werden.

Der SERS Programmer 2 kann dadurch als vollständiger Master in einem Verbund mit SERS-Steuerungen eingesetzt werden.

Im Programmier-Modus wird für den erweiterten Programm-Modus ein Ablaufprogramm (über einen angeschlossenen PC) in den SERS Programmer 2 abgespeichert. Außerdem können hier Parametereinstellungen im SERS Programmer 2 vorgenommen werden.

Die Tastatur mit Display ist bei beiden Versionen (SERS Programmer 2 und Vorgängerversion SERS Programmer) identisch.

Der SERS Programmer 2 (und SERS Programmer 2-R) kann von der Vorgängerversion SERS Programmer optisch durch einen zusätzlich vorhandenen 2-poligen Stecker (neben dem 9-poligen D-Sub-Stecker), für einen 24V-Anschluss, unterschieden werden.

Ab Kapitel 1.3 dieses Handbuchs wird der SERS Programmer 2 nur noch als SERS-Programmer bezeichnet.

1.2 Standardmodus und erweiterter Programm-Modus

Der SERS Programmer 2 kann im Standardmodus oder im erweiterten Programm-Modus betrieben werden.

Der Standardmodus beinhaltet eine vorgegebene, fixe Menüstruktur (siehe Kapitel 2), und ist mit dem SERS-Programmer (Vorgängerversion des SERS Programmer 2) identisch.

Solange kein Ablaufprogramm im SERS-Programmer 2 existiert (d.h. solange der SERS-Programmer V2 noch nicht im Programmier-Modus programmiert wurde), wird der SERS-Programmer 2 beim Einschalten automatisch in den Standardmodus versetzt.

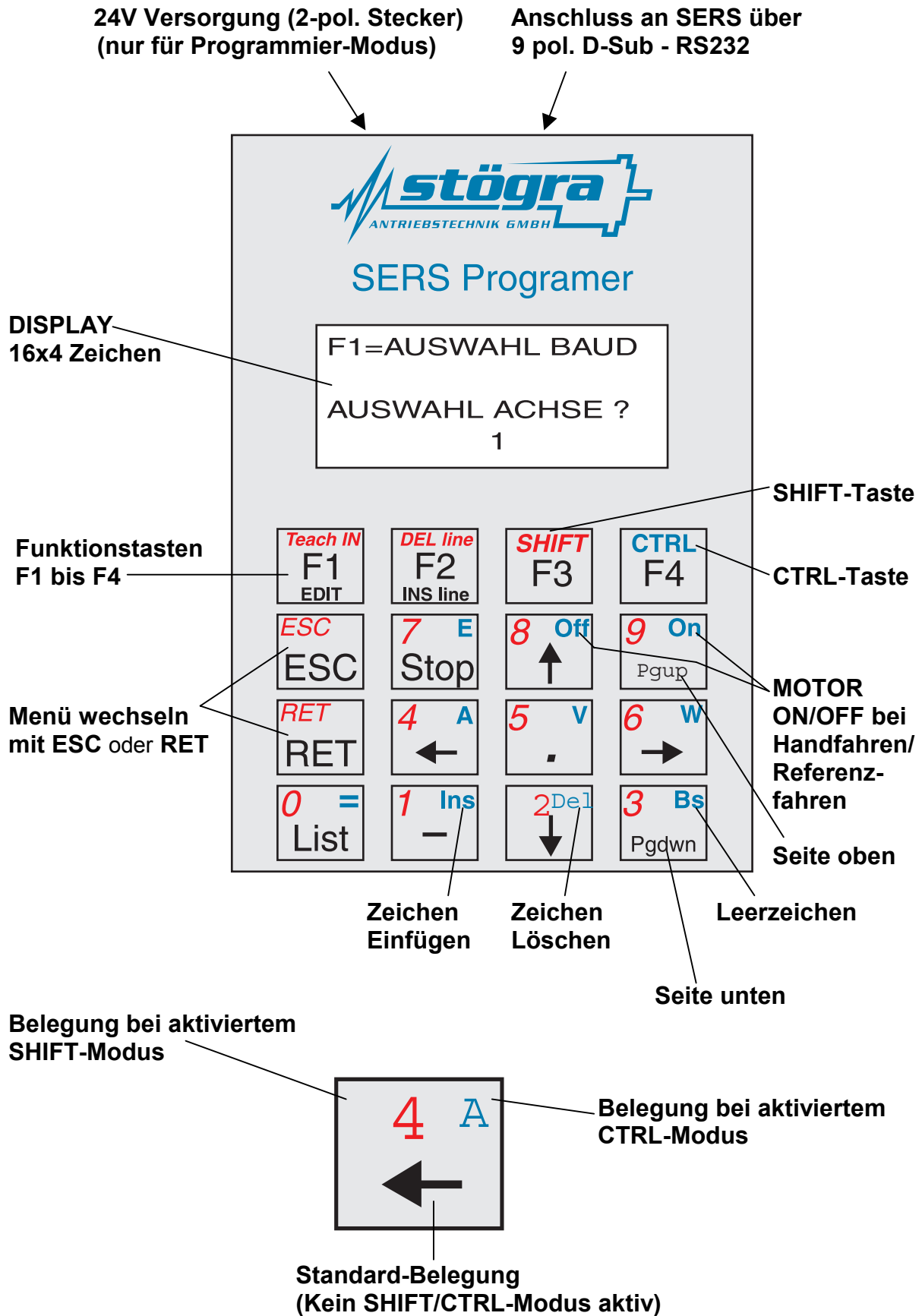
Wenn ein Ablaufprogramm im SERS Programmer 2 existiert, dann wird dieses automatisch beim Einschalten (Power ON) verwendet, und der SERS Programmer 2 dadurch im erweiterten Programm-Modus betrieben.

1.3 Tastatur im erweiterten Programm-Modus

Im Gegensatz zum Standardmodus kann im erweiterten Programm-Modus die Tastenbelegung frei definiert werden.

Die Definition der Tasten erfolgt im Ablaufprogramm des SERS-Programmer.

1.4 Tastatur im Standardmodus



1.5 Anschluss SERS-Programmer an SERS-Steuerung

SERS-Programmer und eine angeschlossene SERS-Steuerung:

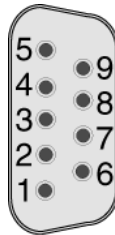


Die Pins 2,3,5 und 9 am 9-poligen Stecker des **SERS-Programmers** müssen **1:1** mit den Pins 2,3,5 und 9 am 9-poligen Stecker der **SERS-Steuerung** verbunden werden. (**Achtung: kein Null-Modem Kabel**)

Pinbelegung 9-poliger D-Sub-Stecker:

Pin

1 : nicht belegt
2 : RXD
3 : TXD
4 : nicht belegt
5 : GND

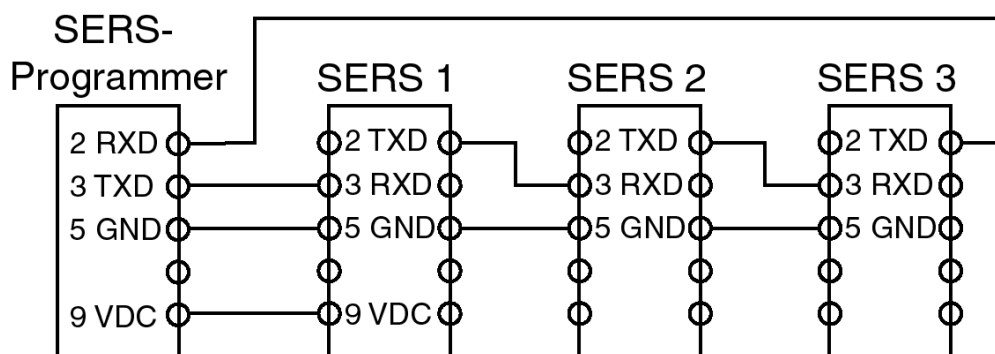


6 : nicht belegt
7 : HSI (Hand Shake IN)
8 : HSO (Hand Shake Out)
9 : 9VDC (Versorgung SERS Programmer)

Die Handshakeleitungen (Pin 7 und 8) sind im SERS-Programmer belegt, werden standardmäßig aber nicht ausgewertet.

SERS-Programmer und mehrere angeschlossene SERS-Steuerungen:

Anschluss-Schema:



ACHTUNG:

Der SERS-Programmer wird über den Pin 9 am 9-poligen D-Sub-Stecker mit 8VDC von einer SERS-Steuerung versorgt.

Beim Anschluß mehrerer SERS-Steuerungen an den SERS-Programmer sollte nur eine SERS-Steuerung den SERS-Programmer über den Pin 9 versorgen !

1.6 Anschluss SERS-Programmer an einen Windows-PC im Programmier-Modus

Im Programmier-Modus (Abspeichern eines Ablaufprogramms für den erweiterten Programm-Modus) wird der SERS-Programmer an einen PC angeschlossen. Der SERS Programmer muss dabei, zusätzlich zum Anschluss an die RS232-Schnittstelle des PCs (Anschluss über 1:1 Kabel, wie in Kapitel 1.5), extern mit 24VDC (über den 2-poligen Stecker) versorgt werden.

1.7 Tastenbelegung im Standardmodus

Die Tasten des SERS-Programmers sind abhängig vom aktuellen Menü im Display einfach, zweifach oder dreifach belegt !

Für viele Menü-Funktionen werden nur die Tasten F1, F2, F3, F4, ESC und RET mit der Standard-Belegung (schwarzen Beschriftung) benötigt.

Die jeweilige Funktion ist dabei aktuell im Displayfenster beschrieben.

Bsp.:

F 1	=	M	A	N	U	E	L	L		M	O	D	E	
F 2	=	P	R	O	G	R	A	M		M	O	D	E	
F 3	=	S	T	A	R	T		P	R	O	G	R	A	M
F 4	=	S	T	A	R	T		A	L	L	E			

Mit der ESC-Taste oder RET-Taste kann man in das vorherige übergeordnete Menü zurückkehren.

In einigen Menü-Funktionen sind die Tasten mehrfach belegt.

Die unterschiedlichen Belegungen der Tasten sind farblich gekennzeichnet (schwarz, rot und blau).

Die Tasten sind standardmäßig mit den SCHWARZEN Tastenbeschriftungen belegt.

Ein Einmaliges Betätigen der ROTEN SHIFT-Taste (F3-Taste) aktiviert den SHIFT-MODUS und damit die ROTE Beschriftung der Tasten.

Der Aktivierte SHIFT-Modus wird durch das Zeichen "▲" rechts unten im Display des SERS-Programmers angezeigt.

Solange dieses Zeichen erscheint, ist die rote Beschriftung der Tasten aktiv.

Beispiel:



Im SHIFT-Modus kann beim Editieren eines Programms oder beim Ändern von Parametern im Parametermodus die Zahl "4" eingegeben werden.

Durch nochmaliges Betätigen der SHIFT-Taste wird der SHIFT-Modus deaktiviert und das Zeichen "▲" verschwindet wieder aus dem Display.

Ein Einmaliges Betätigen der blauen CTRL-Taste (F4-Taste) aktiviert den CTRL-MODUS und damit die BLAUE Beschriftung der Tasten.

Der Aktivierte CTRL-Modus wird durch das Zeichen "■" rechts unten im Display des SERS-Programmers angezeigt.

Solange dieses Zeichen erscheint, ist die blaue Beschriftung der Tasten aktiv.



SHIFT-Modus und CTRL-Modus können nur beim Editieren von Parametern, beim Editieren des Programms und im Terminal-Modus aktiviert werden.

2. Betrieb und Programmierung im Standardmodus

2.1 Nach dem Einschalten

Der SERS-Programmer startet seinen Betrieb sofort nach Anlegen der Versorgungsspannung am 9-poligen D-Sub-Stecker.

Nach einer 2 Sekunden Initialisierungsphase sucht der SERS-Programmer nach angeschlossenen SERS-Steuerungen.

Die an der SERS-Steuerung eingestellte Baudrate und Adresse wird automatisch erkannt (es werden nur die Antriebsadressen 1 bis 3 automatisch überprüft).

Nach Erkennen einer angeschlossenen SERS-Steuerung erscheint folgende Anzeige auf dem Display:

F	1	=	A	U	S	W	A	H	L		B	A	U	D	
A	U	S	W	A	H	L		A	C	H	S	E		?	
								1							

Falls nur eine SERS-Steuerung angeschlossen ist, dann muß die angezeigte Antriebsadresse (entspricht der auf der SERS-Steuerung über die Mini-Schalter eingestellten Adresse) nur mit der "RET"-Taste bestätigt werden.

Falls mehrere SERS-Steuerungen am SERS-Programmer angeschlossen sind, dann kann mit den Pfeil-Tasten "↑" und "↓" eine andere Antriebsadresse angewählt werden.

Alle folgenden Operationen mit dem SERS-Programmer gelten nur für den Antrieb, mit der hier gewählten Adresse (Ausnahme Funktion START ALLE).

Soll später ein anderer Antrieb angesprochen werden, so muß zuerst in obigem Menü mit den Pfeiltasten die entsprechende Adresse eingestellt werden.

Wie man wieder in das Menü zum Anwählen einer Antriebsadresse kommt, wird im Kapitel "Adressen/Sprachmenü" beschrieben.

Nach Betätigen der RET-Taste nach der Auswahl bzw. dem Bestätigen einer Antriebsadresse kommt man in das Hauptmenü.

FEHLERMELDUNGEN:

Falls nach dem Einschalten die Meldung "**** **kein Echo**" erscheint, dann überprüfen Sie die Anschlüsse (Pins 2 und 3 müssen 1:1 verdrahtet sein - siehe Kapitel 1.2).

Falls nach dem Betätigen der Taste RET die Meldung "**** **kein Antrieb adressiert**" erscheint, dann überprüfen Sie die an der SERS-Steuerung über den Adressenschalter eingestellte Adresse und die angewählte Adresse bei "AUSWAHL Adresse ?"

2.2 Allgemeines zu den Menüs

Die Menüs im SERS-Programmer sind nach einer Baumstruktur angeordnet. Durch Betätigen der in den Menüs beschriebenen Funktionstasten F1 bis F4 kommt man in die Untermenüs.

Durch Betätigen der "ESC"-Taste oder der "RET"-Taste springt man wieder in das übergeordnete Menü zurück

2.3 Hauptmenü

In das Hauptmenü kommt man entweder nach Bestätigen der Antriebsadresse mit RET oder ESC direkt nach dem Einschalten (siehe 2.1 "Nach dem Einschalten") oder aus allen anderen Menüs durch Zurückspringen in das jeweilige übergeordnete Menü mit ESC oder RET bis ins Hauptmenü.

F 1	=	M	A	N	U	E	L	L		M	O	D	E			
F 2	=	P	R	O	G	R	A	M		M	O	D	E			
F 3	=	S	T	A	R	T				P	R	O	G	R	A	M
F 4	=	S	T	A	R	T				A	L	L	E			

F1 → Sprung in das MANUELL MODE-Menü (siehe 2.7)

F2 → Sprung in das PROGRAM MODE-Menü (siehe 2.14)

F3 → Sprung zu START Programm (siehe 2.5)

F4 → Sprung zu START Alle (siehe 2.6)

ESC oder RET → Wechsel zum Adressen/Sprachmenü

2.4 Adressen/Sprachmenü

Durch Drücken von ESC oder RET im Hauptmenü erscheint das Adressen/Sprachmenü.

F 1	=	A	U	T	O		A	D	R	/	B	A	U	D
F 2	=	M	A	N			A	D	R	/	B	A	U	D
F 3	=	D	E	U	T	S	C	H						
F 4	=	E	N	G	L	I	S	H						

F1 → Es wird automatisch nach angeschlossenen SERS-Steuerungen gesucht
 - Baudrate und Adresse (1 bis 3) werden automatisch geprüft
 - Menü auf dem Display wie unter "2.1 Einschalten" beschrieben

F2 → Sprung in das Menü, wie unter "2.1 Einschalten" abgebildet
 - Baudrate und Adresse werden nicht automatisch geprüft (manuelle Einstellung der Adresse wie unter "2.1 Einschalten" und dort mit F1 Möglichkeit zur manuellen Einstellung der Baudrate - über Pfeiltasten)

F3 → Deutsche Sprachauswahl (für SERS-Programmer und SERS-Steuerung)

F4 → Englische Sprachauswahl (für SERS-Programmer und SERS-Steuerung)

ESC oder RET → Wechsel zum Hauptmenü

2.5 START Programm

Durch Betätigen der Taste F3 im Hauptmenü kommt man zur "START Programm" - Maske.

F 1 = S T A R T	F 4 = S T O P
I N : 0	O 1 : 0
W 2	O 3 : 0
0 . 0 0 0 0 - o k	O 4 : 0

Folgende SERS-Zustände werden hier angezeigt:

- Nach "IN" in der 2. Zeile im Display wird der Zustand der Eingänge angezeigt (digitale Eingänge I1 bis I8 binär codiert).
- Der Zustand der Ausgänge wird bei O1 bis O4 angezeigt.
- Am Platz in der 3. Zeile unter "IN" stehen Fehler bzw. Warnungsmeldungen.
 - Falls hier nichts steht, so existiert aktuell keine Warnung und kein Fehler.
 - Warnungen werden mit "W" und dem Inhalt von P12 (siehe Handbuch SERS) angezeigt. So bedeutet z.B. die Anzeige "W2" eine Temperaturwarnung.
 - Fehler werden mit "E" und dem Inhalt von P11 angezeigt (siehe Handbuch SERS)
- Die 4. Zeile zeigt die aktuelle Position an
- Ein "ok" nach der Positionsanzeige bedeutet, daß der Antrieb steht

Mit "F1" wird das aktuelle SERS-Programm (Ablaufprogramm im EEPROM der SERS-Steuerung) gestartet.

Mit "F4" oder der "STOP"-Taste wird das Programm unterbrochen bzw. abgebrochen (siehe Beschreibung Parameter P1033 im Handbuch SERS)

Durch "ESC" oder "RET" erfolgt der Rücksprung in Hauptmenü.

2.6 START Alle

Durch Betätigen der Taste F4 im Hauptmenü kommt man zur "START Alle" - Maske. Beim Anschluss mehrerer SERS-Steuerungen an den SERS-Programmer können hier alle Achsen gleichzeitig gestartet werden.

F 1 = S T A R T	F 4 = S T O P
A L L E	

F1 → Senden von Programm-START-Befehl an alle angeschlossenen SERS-Einheiten (Run-Befehl mit broadcast-Adresse "*" : #*RUN)

F4 → Senden von STOP-Befehl an alle angeschlossenen SERS-Einheiten ("S"-Befehl mit der broadcast-Adresse "*" : #*S)

ESC oder RET → Zurück zum Hauptmenü

2.7 MANUELL Mode - Menü

Durch Betätigen der Taste F1 im Hauptmenü kommt man zum "MANUELL Mode" - Menü.

F 1	=	P	A	R	A	M	E	T	E	R				
F 2	=	H	A	N	D	V	E	R	F	A	H	R	E	N
F 3	=	R	E	F	E	R	E	N	Z	F	A	H	R	T
F 4	=	T	E	R	M	I	N	A	L		M	O	D	E

F1 → Parameter-Menü (siehe 2.8)

F2 → Handverfahren-Menü (siehe 2.11)

F3 → Referenzfahren-Menü (siehe (2.12)

F4 → Terminal Mode (siehe 2.13)

ESC oder RET → Zurück zum Hauptmenü

2.8 PARAMETER - Menü

Durch Betätigen der Taste F1 im MANUELL Mode - Menü kommt man zum PARAMETER - Menü.

F 1	=	A	L	L	E										
		P	A	R	A	M	E	T	E	R					
F 2	=	R	E	S	E	T									
		W	A	R	N	U	N	G	/	F	E	H	L	E	R

F1 → Parameter Maske (siehe 2.9)

F2 → RESET Menü (siehe 2.10)

ESC oder RET → Zurück zum Hauptmenü

2.9 PARAMETER - Maske

Durch Betätigen der Taste F1 im PARAMETER - Menü kommt man zur PARAMETER - Maske.

Falls bis jetzt noch keine Paßwortabfrage erfolgt ist (z.B. vor dem PROGRAM Mode Menü) dann wird nun nach einem Paßwort gefragt.

Das Paßwort ist in der SERS-Steuerung definiert.

Jede angeschlossene SERS-Steuerung erfordert nach einem Power-On-Reset einmal die Eingabe eines Paßworts, bevor auf den Parameter und Programmbereich zugegriffen werden kann.

P	A	S	S	W	O	R	T	?									

Die Eingabe einer 4-stelligen Zahl wird hier erwartet (mit SHIFT-Taste umschalten auf rote Ziffernbelegung der Tasten).

Nach dem Drücken der RET-Taste erscheint die PARAMETER-Maske.

Falls die eingegebene Zahl mit dem in der SERS definierten Paßwort übereinstimmt, dann werden alle SERS-Parameter angezeigt.

Falls die eingegebene Zahl nicht mit dem in der SERS definierten Paßwort übereinstimmt, dann werden nur die Parameter angezeigt, die in der SERS für die Bedienung ohne Paßwort freigegeben wurden (Parameter P1060 bis P1065 - siehe Handbuch SERS).



HINWEIS: Falls kein Paßwort in der SERS definiert ist:

Bei der Paßwortabfrage muß nur die RET-Taste gedrückt werden (keine Paßwort-eingabe). Danach werden alle Parameter angezeigt.

Falls versehentlich ein falsches Paßwort eingegeben wurde:

Die SERS-Steuerung muß neu gestartet werden (Power-ON-Reset) um erneut ein Paßwort eingeben zu können

PARAMETER Maske:

F	1	=	E	D	I	T			F	2	=	S	A	V	E		
P	0	=	0														
Z	=	0															
P	2	=	0														

Mit den Pfeiltasten "↑" und "↓" und den Bildtasten "Pgup" und "Pgdn" kann der Cursor zu den einzelnen Parametern bewegt werden.

F1 → Der vom Cursor aktuell markierte Parameter (2.Zeile im Display) kann hier geändert werden.



ACHTUNG: Zur Eingabe von Zahlen muß zuerst der SHIFT-Modus mit der roten Tastenbelegung aktiviert werden (Einmaliges Drücken der SHIFT-Taste). Falls Zeichen gelöscht oder eingefügt werden sollen muß der CTRL-Modus mit der blauen Belegung aktiviert werden (Einmaliges Drücken der CTRL-Taste und "Ins" zum Einfügen von Zeichen und "Del" zum Löschen von Zeichen)

Es dürfen **keine Wichtungen** mit eingegeben werden (V=3000 und **nicht** V=3000U/min)

Mit **"RET"** wird der **geänderte Wert übernommen**, mit **"ESC"** wird der **alte Wert beibehalten**.

Bsp.: Ändern der Geschwindigkeit - Parameter "V" - Das Display muß dazu Folgendes anzeigen:

F 1	=	E	D	I	T		F 2	=	S	A	V	E	
V =	3	0	0	0	.	0	0	0	0	U /	m	i	n
P 1	0	0	=	0									
P 1	0	1	=	0									

Die Geschwindigkeit "V" soll auf 500U/min geändert werden. Zum Ändern von "V" muß jetzt die Taste F1 gedrückt werden. Danach erscheint folgende Anzeige:

V =													

Um Zahlen eingeben zu können muß die SHIFT-Taste gedrückt werden, dadurch erscheint rechts unten im Display das Zeichen "▲"

Jetzt kann die Zahl "500" eingegeben werden

V =	5	0	0										

und mit "RET" übernommen werden.



WICHTIGER HINWEIS:

Um die Parameteränderungen dauerhaft in der SERS abzuspeichern muß zusätzlich die Taste F2 (SAVE) gedrückt werden. Andernfalls sind die Änderungen nach dem nächsten Power-ON-Reset der SERS-Steuerung verloren.

Nach Drücken von F2 meldet die SERS-Steuerung ein "ok" zurück. Mit "ESC" oder "RET" wird wieder in die Parametermaske gewechselt.

ESC oder RET → Zurück zum PARAMETER Menü

2.10 RESET - Menü

Durch Betätigen der Taste F2 im PARAMETER - Menü kommt man in das RESET-Menü

F 1	=	R	E	S	E	T		F	E	H	L	E	R	
F 2	=	R	E	S	E	T		W	A	R	N	U	N	G

F1 → in der SERS werden vorhandene Fehlermeldungen zurückgesetzt (P11=0)

F2 → in der SERS werden vorhandene Warnungen zurückgesetzt (P12=0)

ESC oder RET → Zurück zum PARAMETER Menü

2.11 HANDVERFAHREN - Menü

Durch Betätigen der Taste F2 im "MANUELL Mode" - Menü kommt man in das HANDVERFAHREN - Menü

F 1	=	L	A	N	G	S	A	M		L	I	N	K	S
F 2	=	L	A	N	G	S	A	M		R	E	C	H	T
F 3	=	S	C	H	N	E	L	L		L	I	N	K	S
F 4	=	S	C	H	N	E	L	L		R	E	C	H	T

Bevor der Antrieb Verfahren werden kann, muß der Motor bestromt sein (freigegeben sein). Dabei zeigt die 7-Segment-Anzeige auf der SERS-Steuerung eine "5" an (wenn keine Fehlermeldung oder Warnung existiert).

Durch Drücken der "ON" Taste wird der Motor bestromt, durch Drücken der "OFF" Taste wird der Motorstrom abgeschaltet. (**Hinweis:** der CTRL-Modus muß hierzu nicht aktiviert werden)

F1 → Antrieb dreht langsam links

F2 → Antrieb dreht langsam rechts

F3 → Antrieb dreht schnell links

F4 → Antrieb dreht schnell rechts



HINWEIS:

Ob der Motor bei F1 bzw. F3 tatsächlich links dreht, hängt davon ab, wie die Phasen am Motor mit der SERS-Steuerung verdrahtet wurden (ein Vertauschen der beiden Anschlüsse innerhalb einer Motorphase bewirkt eine Drehrichtungsänderung) !

Die Geschwindigkeit und die Beschleunigung beim Handverfahren werden über die Parameter P1018 bis P1020 eingestellt (siehe Handbuch SERS)

ESC oder RET → Zurück zum MANUELL Mode - Menü

2.12 REFERENZFAHREN - Menü

Durch Betätigen der Taste F3 im "MANUELL Mode" - Menü kommt man in das REFERENZFAHREN-Menü

F 1	=	S	T	A	R	T									
		R	E	F	E	R	E	N	Z	F	A	H	R	T	
F 2	=	S	T	O	P										

Um Referenzfahren zu können muß der Motor bestromt sein (freigegeben sein). (Anzeige "4" auf der 7-Segment-Anzeige der SERS-Steuerung → Motor ist nicht bestromt, Anzeige "5" → Motor ist bestromt)

Durch Drücken der "ON" Taste wird der Motor bestromt, durch Drücken der "OFF" Taste wird der Motorstrom abgeschaltet. (**Hinweis:** der CTRL-Modus muß hierzu nicht aktiviert werden)



F1 → Das Referenzfahren des Antriebs wird gestartet
(Siehe auch Handbuch SERS - Kapitel Referenzfahren)

F2 → Der Antrieb wird gestoppt

ESC oder RET → Zurück zum MANUELL Mode - Menü

2.13 TERMINAL Mode

Durch Betätigen der Taste F4 im "MANUELL Mode" - Menü kommt man in den TERMINAL Mode.

Hier können alle auf der Tastatur verfügbaren Zeichen (und die über die Funktion "List" zusätzlich wählbaren Zeichen - siehe Kapitel 2.18) direkt an die SERS gesendet werden.



ACHTUNG: Jede neue Zeile an eine SERS-Steuerung muß mit dem Zeichen "#" beginnen. Außerdem muß einmal die Antriebs-Adresse der SERS-Steuerung nach dem "#"-Zeichen eingegeben werden.

Bsp: #1ON

Damit wird der Antrieb mit der Adresse 1 bestromt.



HINWEIS: Der Terminal Mode ist nur mit gültigem Paßwort zugänglich.

ESC oder RET → Zurück zum MANUELL Mode - Menü

2.14 PROGRAM Mode - Menü

Durch Betätigen der Taste F2 im Hauptmenü kommt man in das "PROGRAM Mode" - Menü.

Falls bis jetzt noch keine Passwortabfrage erfolgt ist (z.B. vor der PARAMETER Maske) dann wird nun nach einem Passwort gefragt.

Das Passwort ist in der SERS-Steuerung definiert.

Jede angeschlossene SERS-Steuerung erfordert nach einem Power-On-Reset einmal die Eingabe eines Passworts, bevor auf den Parameter und Programmbereich zugegriffen werden kann.

P	A	S	S	W	O	R	T	?									

Die Eingabe einer 4-stelligen Zahl wird hier erwartet (mit SHIFT-Taste umschalten auf rote Ziffernbelegung der Tasten).



HINWEIS: Falls kein Passwort in der SERS definiert ist:

→ nur die RET-Taste betätigen (keine Paßworteingabe).

Falls versehentlich ein falsches Passwort eingegeben wurde:

Die SERS-Steuerung muß neu gestartet werden (Power-ON-Reset) um erneut ein Paßwort eingeben zu können

Falls das jetzt eingegebene Passwort richtig ist oder kein Passwort definiert ist, dann wird nach Drücken der RET-Taste folgendes Menü angezeigt:

F	1	=	E	D	I	T		P	R	O	G	R	A	M			
F	2	=	N	E	U			P	R	O	G	R	A	M			
F	3	=	T	E	S	T		P	R	O	G	R	A	M			

Falls das eingegebene Passwort falsch ist bzw. falls ein Passwort definiert ist und nur die RET-Taste gedrückt wird (ohne Passworteingabe), dann wird folgendes Menu angezeigt:

F	1	=	E	D	I	T		P	R	O	G	R	A	M			

F1 → EDIT PROGRAM - Maske (siehe 2.15)

F2 → NEU PROGRAM - Maske (siehe 2.16)

F2 → TEST PROGRAM - Maske (siehe 2.17)

ESC oder RET → Zurück zum Hauptmenü

2.15 EDIT PROGRAM - Maske

Durch Betätigen der Taste F1 im PROGRAM Mode Menü kommt man zur EDIT PROGRAM - Maske.

Bei vorher falsch oder gar nicht eingegebenem Paßwort (falls ein Passwort in der SERS definiert ist) werden nur die in den Parameter P1070 bis P1074 freigegebenen Programmzeilen angezeigt.

Bsp.:

G	E	S	C	H	W	I	N	D	I	G	K	E	I	T	
V	=	1	0	0	0	.	0	0	0	0					
D	U	R	C	H	M	E	S	S	E	R					
W	=	1	0	.	0	0	0	0							

Alle angezeigten Wertzuweisungen können jetzt geändert werden. Dazu muß der Cursor mit den Pfeiltasten bzw. Seite oben/unten Tasten auf die zu ändernde Zuweisung gebracht werden.

Nach Drücken von F1 kann der Wert geändert werden, entweder durch Überschreiben des Werts oder durch Ändern der einzelnen Stellen mit Hilfe der Pfeiltasten "↑" und "↓".

Mit der RET-Taste wird der geänderte Wert in das Programm übernommen.
Mit der ESC-Taste wird der alte Wert beibehalten.

Falls das Paßwort vorher richtig eingegeben wurde wird das komplette Programm angezeigt.

Bsp.:

1	:	O	N												
2	:	L	1												
3	:	"	G	E	S	C	H	W	I	N	D	I	G	K	E
I	T	"													

Mit den Pfeiltasten "↑", "↓", "→" und "←" und den Bildtasten "Pgup" und "Pgdn" können alle Programmstellen erreicht werden.

Mit den Funktionstasten F1 und F2 können insgesamt 4 Funktionen aufgerufen werden.

- **"EDIT"** → Parameterzuweisung ändern (editieren) (z.B: in der Zeile 4:V=1000.0000 kann nach Betätigen von F1 der Wert für V überschrieben oder mit den Pfeiltasten "↑" und "↓" ziffernweise nach oben bzw. unten geändert werden (nur Parameterzuweisungen können mit "EDIT" geändert werden). Mit der RET-Taste wird der geänderte Wert im Programm übernommen, mit der ESC-Taste bleibt der alte Wert erhalten.
- **"Teach IN"** (SHIFT-TASTE und danach F2-Taste) → Die Anzeige wechselt in das HANDVERFAHREN Menü und der Antrieb kann mit den Handverfahrfunktionen an eine gewünschte Position gefahren werden. Mit der RET-Taste wird diese Position dann in das Programm übernommen, indem eine neue Zeile hinter



der aktuell mit dem Cursor markierten Zeile eingefügt wird mit "W=' aktuelle Motorposition" (die aktuelle Motorposition ist die soeben angefahrne Position)

HINWEIS zu Teach IN: Beachten Sie dabei den richtigen Positioniermodus (absolut oder relativ Positionieren - Parameter P1014)

Mit der ESC wird die Funktion abgebrochen und keine Zeile eingefügt.

- **"INS line"** → Hinter der aktuell mit dem Cursor markierten Zeile wird eine neue Zeile eingefügt - hiermit kann eine beliebige neue Zuweisung oder Kommando in das Programm eingefügt werden - als Zeichensatz stehen alle Zeichen der dreifach belegten Tasten (über SHIFT und CTRL) und die zusätzlichen Zeichen über die "LIST"-Taste (siehe Kapitel 2.18) zur Verfügung.
Mit der RET-Taste wird die neue Zeile in das Programm übernommen, mit der ESC-Taste wird die Funktion abgebrochen.
HINWEISE: Um eine Zeile vor der ersten Zeile einzufügen muß der Cursor ganz links auf der Zeilennummerierung ("1") stehen.
Falls mehrere Kommandos in einer Zeile stehen, und eine Anweisung zwischen zwei Kommandos einer Zeile oder vor der nächsten Zeile plaziert werden soll, dann muß der Cursor mit den Pfeiltasten "→" und "←" auf die richtige Anweisung im Programm plaziert werden, und danach die neue Anweisung eingefügt werden (**Eingefügt wird immer nach dem Cursor !**)
- **"DEL line"** → Die aktuell mit dem Cursor markierte Zeile wird gelöscht. Vor dem Löschen wird aber nochmals gefragt, ob die Zeile tatsächlich gelöscht werden soll (ein RET löscht die Zeile, alle anderen Tasten brechen den Löschvorgang ab)



HINWEIS: Im Gegensatz zu Parameteränderungen werden Programmänderungen sofort dauerhaft im EEPROM der SERS-Steuerung abgespeichert (Bei Parameteränderungen erfolgt das Abspeichern im EEPROM erst durch Betätigen der Taste F2 - siehe Kapitel 2.9)

ESC oder RET → Zurück zum PROGRAM Mode Menü

2.16 NEU PROGRAM - Maske

Durch Betätigen der Taste F2 im PROGRAM Mode Menü kommt man zur NEU PROGRAM - Maske.

N	E	U	E	S		P	R	O	G	R	A	M	M		?
A	L	T	E	S		P	R	O	G	R	A	M	M		
W	I	R	D		G	E	L	O	E	S	C	H	T		!
R	E	T		=		J	A								



Durch Betätigen der RET-Taste wird das aktuelle Programm in der SERS gelöscht !

Danach kann ein neues Programm eingegeben werden.

Es stehen alle Funktionen wie unter 2.15 (mit richtig eingegebenem Passwort) beschrieben zur Verfügung.

ESC oder RET → Zurück zum PROGRAM Mode Menü

2.17 TEST PROGRAM - Maske

Durch Betätigen der Taste F3 im PROGRAM Mode Menü kommt man zur TEST PROGRAM - Maske.

Hier kann das Programm Schritt für Schritt ausgeführt werden.

Bsp.:

F	1	=	T	R	A	C	E		F	4	=	S	T	O	P
1	:	O	N												
2	:	W	=	1	0	0	0	.	0	0	0	0			
0	.	0	0	0	0	-	o	k							

- 2. Zeile : zuletzt ausgeführte Anweisung im Programm
- 3. Zeile : nächste auszuführende Anweisung
- 4. Zeile : aktuelle Antriebsposition und ein "-ok" falls der Antrieb steht

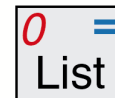
F1 → Die nächste Anweisung (im Display in 3. Zeile dargestellt) wird ausgeführt.

F2 → Die aktuelle Anweisung und der Antrieb werden gestoppt

Danach kann ein neues Programm eingegeben werden.

2.18 Zusätzliche Zeichen im Terminalmodus und bei EDIT Programm und NEU Program

Beim Editieren eines Programms und im Terminalmodus kann über die Taste "List" auf zusätzliche Zeichen zugegriffen werden.



Mit den Pfeiltasten kann ein Zeichen aus der angezeigten Liste ausgewählt werden und mit der Taste RET in das Programm oder in das Terminal übernommen werden.

Mit den Tasten "Pgup" und "Pgdown" können insgesamt drei verschiedene Seiten mit Zeichen angewählt werden.

1. Seite:

I	F		I	N		I		G	T		G	S		R	T
D		H		L		O		P		S		Z		X	
P	O	S		R	U	N		S	A	V	E				
!		*		?		=		#		/		+		&	

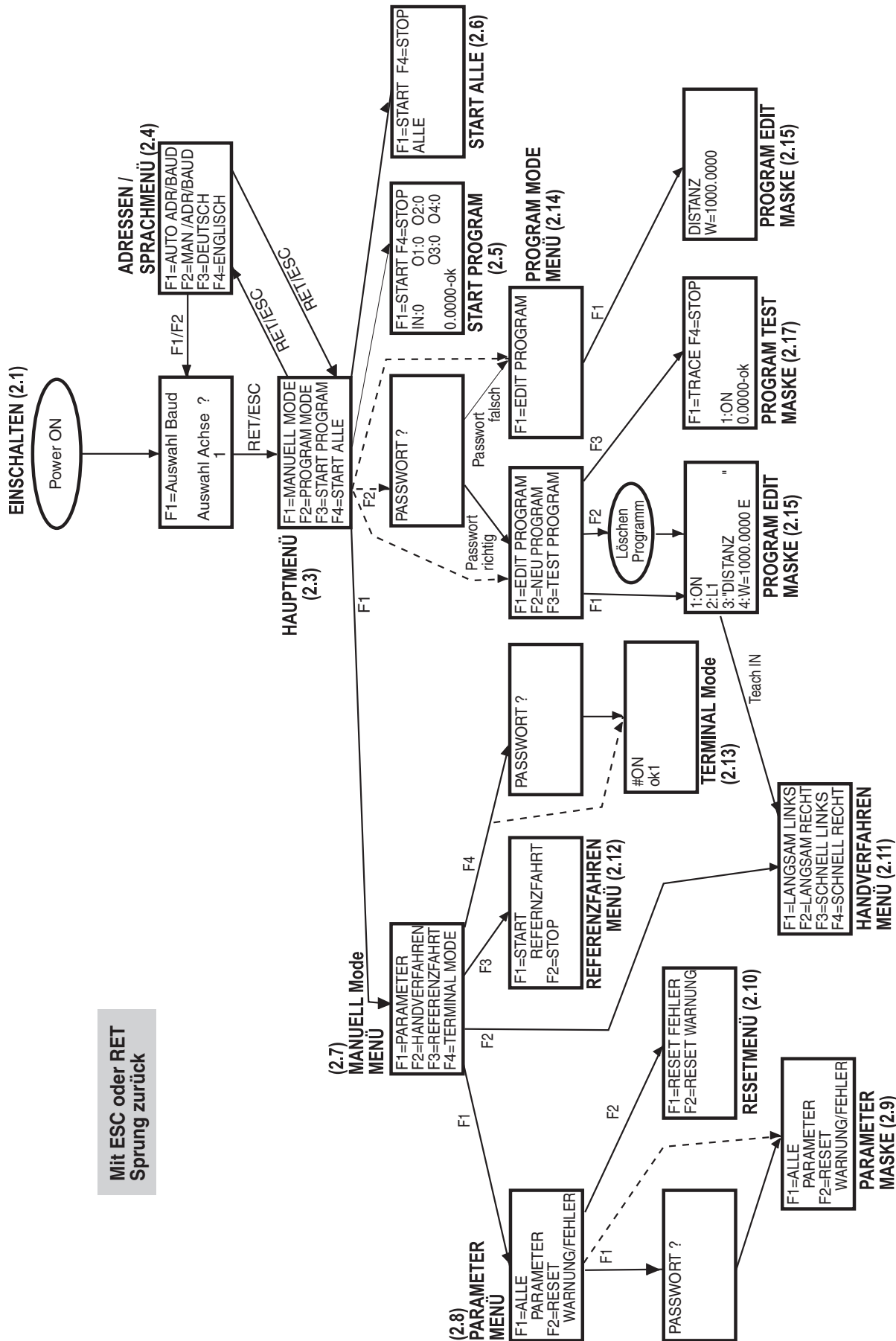
2. Seite:

"		A		B		C		D		E		F		G	
H		I		J		K		L		M		N		O	
P		Q		R		S		T		U		V		W	
X		Y		Z		^		<		>		[]	

3. Seite:

:		a		b		c		d		e		f		g	
h		i		j		k		l		m		n		o	
p		q		r		s		t		u		v		W	
x		y		z		%		@		\$		()	

2.19 Menü – Baumstruktur im Standardmodus



Mit ESC oder RET Sprung zurück

3. Betrieb und Programmierung im erweiterten Programm-Modus

3.1 Programmiermodus

Um ein Programm für den erweiterten Programm-Modus in den SERS-Programmer abspeichern zu können, oder um Parametereinstellungen vornehmen zu können, muss der SERS-Programmer über ein 1:1 Verbindungskabel mit der RS232-Schnittstelle eines PC verbunden werden.

Ausserdem muss der SERS-Programmer mit 24VDC versorgt werden (über den 2-poligen Stecker, der neben dem 9-poligen RS232 D-Sub Stecker platziert ist).

Sofort nach (oder bereits vor) dem Anlegen der 24VDC Spannung muss die Taste "F1" am SERS-Programmer V2 gedrückt werden, bis im Display folgende Anzeige erscheint:

M	A	S	T	E	R	M	O	D	E						
1	9	2	0	0	B	A	U	D							

Jetzt kann die STÖGRA Programmiersoftware auf dem PC gestartet werden, und mit dem SERS-Programmer kommunizieren. Die Software zeigt rechts oben die gefundene Baudrate (19200 Baud) und die gefundene Adresse (200) an.

Der SERS-Programmer hat die Adresse "0". Um ihn programmieren zu können, muss er sich aber wie ein Slave verhalten. Und das tut er, wenn er mit der Adresse "200" angesprochen wird.

Falls die Software "kein Antrieb" anzeigt, schließen Sie die Software wieder und überprüfen Sie die Verbindung (korrektes 1:1 Kabel, SERS-Programmer bereit mit obiger Anzeige) und starten Sie die Software erneut.

Falls das Problem immer noch besteht (kein Antrieb), dann versuchen Sie, falls vorhanden, einen anderen PC.

Mit der STÖGRA-Programmiersoftware können Sie nun ein Ablaufprogramm mit eigenem Anzeigemenu erzeugen, und in den SERS-Programmer speichern. Es stehen 8Kbyte Programmspeicher zur Verfügung, was ca. 1200 Anweisungen entspricht.

Mit dem Button "Programm" in der PC-Programmiersoftware wird das Programm-Fenster geöffnet, in dem das Ablaufprogramm erzeugt werden kann.

Mit dem Button "Programm übertragen" wird das Ablaufprogramm in den SERS-Programmer übertragen.

Die Syntax für das Ablaufprogramm bzw. für das Anzeigemenu wird in Kapitel 3.3 beschrieben.

Mit dem Button "Parameter" in der PC-Programmiersoftware werden die aktuellen Parametereinstellungen im SERS-Programmer angezeigt.

Die Parameter können jetzt nach Bedarf geändert werden.

Mit dem Button "Parameter in E²Prom Speichern" werden die Änderungen dauerhaft im E²Prom des SERS-Programmer abgespeichert.

Erläuterungen zu den relevanten Parametern finden Sie in Kapitel 3.6.

Nach Abspeichern eines Ablaufprogramms in den SERS-Programmer wird dieser, nach einem Neustart (Versorgungsspannung AUS/EIN bzw. Abschalten der 24VDC und Verbinden mit einer SERS-Steuerung), automatisch im erweiterten Programm-Modus, mit dem hier eigens erstellten Anzeigemenu, gestartet.

3.2 Betrieb im erweiterten Programm-Modus

Wenn ein gültiges Ablaufprogramm im SERS-Programmer existiert (d.h. im Programmiermodus erstellt wurde), dann startet der SERS-Programmer nach einem Power-ON-Reset automatisch im erweiterten Programm-Modus. Damit wird das Ablaufprogramm automatisch gestartet.

Vor dem Start des Ablaufprogramms erfolgt ein automatischer Adressenscan (d.h. es wird nach angeschlossenen SERS/WSERS-Antrieben mit Adressen 1 bis 9 gesucht). Pro vorhandenem Antrieb wird ein entsprechendes Bit in Parameter P1188 gesetzt (siehe Beschreibung P1188). Die niedrigste vorhandene Adresse wird in Parameter P1185 eingetragen.

Falls $P1186=0$ oder $P1186>7$, dann wird auch die Baudrate der angeschlossenen SERS/WSERS-Antriebe automatisch ermittelt (bei mehreren angeschlossenen Antrieben muss die Baudrate aber bei allen Antrieben identisch sein). Andernfalls ($0 < P1186 < 7$) wird die entsprechend in P1186 definierte Baudrate verwendet.

Der SERS-Programmer hat immer die Adresse 0 (nur im Programmiermodus wird er mit Adresse "200" angesprochen).

3.3 Syntax erweiterter Programm-Modus

3.3.1 Syntax Programmierung Anzeigemenü

Die Programmierung des Anzeigemenüs erfolgt durch Steuerzeichen die zwischen Anführungszeichen " " gesetzt werden.

D.h. die Anführungszeichen " " müssen explizit programmiert werden.

Folgende Steuerzeichen sind definiert:

"\a": Anfang des Menü, für Menüs mit max. 4 Zeilen

"\b": Anfang des Menü, für Menüs mit mehr als 4 Zeilen (gescrollte Menüs)

"\e": Ende des Menü (hier wird die Tastatur ausgewertet)

"\c": LCD-Anzeige löschen und Ausgabe links oben beginnen

"\n" oder "\r": neue Zeile beginnen (Zeilenumbruch)

Ein Zeilenumbruch wird aber auch automatisch nach jeder Zeile erzeugt, wenn die Zeile nicht mit „\%“ abgeschlossen wurde.

Bei "\n" oder "\r" werden außerdem alle Zeichen bis zum Zeilenende, die nach "\n" oder "\r" kommen, gelöscht

"\%": keine neue Zeile beginnen (kein automatischer Zeilenumbruch), sondern Ausgabe der nächsten Zeichen direkt hinter den zuletzt definierten Zeichen

"\1": Ausgabe in der ersten Zeile beginnen

"\2": Ausgabe in der zweiten Zeile beginnen

"\3": Ausgabe in der dritten Zeile beginnen

"\4": Ausgabe in der vierten Zeile beginnen

"#x": "x" kann den Buchstaben "a" oder eine Ziffer (SERS-Slave Adresse) enthalten

"#a" (x = "a"): der folgende Parameter bezieht sich auf die aktuelle SERS-Adresse (in P1185 gesetzte Adresse – siehe P1185 Seite 40)

"#1" (x = "1"): der folgende Parameter bezieht sich auf die SERS mit Adresse "1" Alle nachfolgenden Zeichen bis "?" oder "!" sind Parameter, die nur angezeigt werden, oder angezeigt werden und editierbar sind (abhängig davon ob "?" oder "!" folgt – siehe unten).

"?": Parameter (siehe oben "#x") in LCD-Anzeige schreiben
Der Parameter wird nur angezeigt und ist nicht editierbar

"!": Parameter (siehe oben "#x") in LCD-Anzeige schreiben
Der angezeigte Wert (Parameter) kann mit der Taste "F1" editiert werden.

"text": *text* können beliebige Zeichen bzw. ein beliebiger Ausdruck sein, mit Ausnahme der oben definierten Steuerzeichen



ACHTUNG:

Wenn im Anzeigemenü Parameter vorhanden sind, die editiert werden sollen (Steuerzeichen "!"), dann muss das Menü wiederholt aufgerufen werden (z.B. in einer Wiederhol- bzw. Endloschleife).

Nur bei einem Aufruf der entsprechenden Zeile wird geprüft, ob die Taste "F1" betätigt wurde. Und nur dann kann der entsprechende Parameter editiert werden.

3.3.2 Syntax Programmierung Ablaufprogramm

Im folgendem wird die allgemeine Syntax Definition für ein Ablaufprogramm beschrieben. Die Syntax ist identisch mit der Syntax für SERS-Antriebe (Ablaufprogramme, die in SERS-Antrieben abgespeichert werden). Alle nicht weiter beschriebene Ausdrücke und Zeichen werden direkt eingegeben. *Kursiv* geschriebene Ausdrücke werden weiter beschrieben. Ausdrücke in [] sind optional. Alle Schlüsselwörter können groß oder klein geschrieben werden. Erläuterungen zur Syntax befinden sich hinter dem Kommentarzeichen // .

Anweisungszeile

[*Achsnummer*] [*Anweisungsliste*] *Zeilenende*

Achsnummer

Dezimalkonstante

*

// Adressiert alle Achsen – z.B. für Synchronstart aller Achsen – Broadcast Adresse

Bei Verwendung der STÖGRA Programmiersoftware wird " # [*Achsnummer*] " bei jeder Zeile automatisch ergänzt. Relevant für die Programmierung ist in diesem Fall somit nur noch:

[*Anweisungsliste*] *Zeilenende*

Anweisungsliste

Anweisung

Anweisung *Anweisungsliste*

Anweisung

" *zeichenliste* " // Text zur Anzeige im SERS-Programmer

Zuweisung

Kommando

operator operand // Rechnen

unary_operator // Rechnen

IF [!] *adresse* [*bedingung dezimalkonstante*] // Abfrage siehe Kapitel 3.5.1

WAIT [!] *adresse* [*bedingung dezimalkonstante*] // "Warte" abhängig von Ereignis

: [!] *adresse* [*bedingung dezimalkonstante*] // Handfahren bis zu aktivem Eingang

GOTO *dezimalkonstante* // Sprung zu Labelnummer *dezimalkonstante*

GOSUB *dezimalkonstante* // Sprung zu Unterprogramm bei Labelnummer *dezimalkonstante*

GT *dezimalkonstante* // äquivalent zu GOTO

GS *dezimalkonstante* // äquivalent zu GOSUB

RETURN, RT // Rücksprung aus Unterprogramm – Sprungziel ist die Zeile nach letztem GOSUB-Sprung

RS // Handfahren rechts langsam (**R**ight **S**low)

RF // Handfahren rechts schnell (**R**ight **F**ast)

LS // Handfahren links langsam (**L**eft **S**low)

LF // Handfahren links schnell (**L**eft **F**ast)

L *dezimalkonstante* // Labelnummer (L1 bis L128)

PE // Programmende

PSAVE // Parameter in der SERS dauerhaft abspeichern – Sichern innerhalb der SERS vom RAM in das E²PROM

POSSAVE	// P51 (Ist-Position) im E ² Prom abspeichern
POS0	// Elektrische "0"-Position (alle 7,2°) anfahren
POSR	// Fahrt zur Sollposition bei Encoderfehler (siehe P1043)
RUN	// Entspricht P0=1 – Programm wird gestartet
VER	// Version ausgeben
ON	// Entspricht P134=7 – Motor bestromen
OFF	// Entspricht P134=0 – Motorstrom abschalten

Bedingung

< <= = > >= <>

Zeichenliste

Zeichen

Zeichenliste Zeichen

Zeichen

// beliebige Zeichen außer Anführungszeichen " und 0x00 (=ASCII-Zeichen mit Code 00)

Zuweisung

Adresse = datum

X = operand

// Akku zum Rechnen

Adresse ?

// Parameter Datum (Wert bzw. Inhalt) ausgeben

Adresse ??

// Parameterbezeichnung ausgeben

operator

+

// zum Akkuinhalt wird addiert

-

// vom Akkuinhalt wird subtrahiert

*

// Multiplikation mit Akkuinhalt

/

// der Akkuinhalt wird dividiert

&

// zum Akkuinhalt wird logisch "und" verknüpft

|

// zum Akkuinhalt wird logisch "oder" verknüpft

^

// zum Akkuinhalt wird logisch "exkl-oder" verknüpft

unary operator

NOT

// der Akkuinhalt wird bitweise invertiert

NEG

// der Akkuinhalt wird negiert

operand

adresse

dezimalkonstante

Adresse

A

// äquivalent P138 – Beschleunigung

B

// äquivalent P1096 – Bremsrampe beim Polynomfahren

ADC

// äquivalent P1046 – Analogeingang

C1, C2, C3

// äquivalent P100, P101, P102 - Zähler

D

// äquivalent P1100 – Zeitverzögerung in $1/10$ Sekunden

D2

// äquivalent P1164 – Wartezeit in $1/10$ Sekunden

DB

// äquivalent P1166 – Wartezeitbit

M1bis M20

// Merker M1 bis M20

I1 bis I8

// Eingänge I1 bis I8 aus P1300

IN

// äquivalent P1300 (für Syntax mit IF und WAIT)

I9 bis I16

// äquivalent P1301 – Eingänge I9 bis I16

J1

// Referenzschalter bei Abfragen mit IF, WAIT und ":"

J2

// Endschalter Links bei Abfragen mit IF, WAIT und ":"

J3	// Endschalter Rechts bei Abfragen mit IF, WAIT und “:”
J4	// STOP-Schalter bei Abfragen mit IF, WAIT und “:”
J5	// Serviceschalter bei Abfragen mit IF, WAIT und “:”
J6	// Nullpunkt, bei Drehgebereingängen und Drehgebern mit vorhandenem Nullpunkt
O1 bis O16	// Ausgänge 01 bis O16
P <i>dezimalkonstante</i>	// Parameter – siehe Parameterbeschreibung im SERS/WSERS-Handbuch, und hier in Kapitel 3.4
POS	// äquivalent P336 – Status In-Position
R0 bis R99	// Register zur freien Verwendung
V	// äquivalent P91 - Geschwindigkeit
W	// äquivalent P47 – Positionierweg bzw. Position
WA	// wie W aber zusätzlich P1014=2 (absolute Position)
WR	// wie W aber zusätzlich P1014=0 (relative Position)
WP	// erster Positionierabschnitt im Modus Polynomfahren
WAP	// wie WP aber zusätzlich P1014=2 (absolute Position)
WRP	// wie WP aber zusätzlich P1014=0 (relative Position)
WPT	// letzter Positionierabschnitt im Modus Polynomfahren
WAPT	// wie WPT aber zusätzlich P1014=2 (absolute Position)
WRPT	// wie WPT aber zusätzlich P1014=0 (relative Position)
X	// äquivalent P1047 - Akku zum Rechnen
Z	// äquivalent P1 – Zieladresse für Masterbetrieb

Kommando

E	// Start aktuellen Positionierjob – mit den aktuellen Werten von W (P47) , V (P91) und A (P138)
H	// Start Referenzfahren
S	// Stop – der Motor wird angehalten siehe auch Parameter P1033 – Fortsetzen nach Stop

Datum

[-] *dezimalkonstante*

Zeilenende

Carriage Return	// RETURN-Taste der Tastatur
/r	// Return Zeichen – ASCII-Code 13 (dezimal)
/n	// Linefeed Zeichen – ASCII-Code 10 (dezimal)

Dezimalkonstante

Dezimalziffer	
. Dezimalziffer	// z.B. . 5 (= 0.5)
Dezimalkonstante Dezimalziffer	// z.B. 10.75

Dezimalziffer

0,1,2,3,4,5,6,7,8,9

Trennzeichen

Leerzeichen	
,	// Komma
;	// Strichkomma verhindert einen Zeilenumbruch bei Programmeingabe in der aktuellen Zeile
“Tab“	// Tabulatorzeichen

3.3.3 Zieladresse bei Steuerzeichen, Parameter und Abfragen

Steuerzeichen für die Programmierung der Anzeige beziehen sich immer automatisch auf den SERS-Programmer (Adresse "0").

Anweisungen, Kommandos, Parameter und Abfragen in einem Ablaufprogramm müssen durch "Z=..." explizit auf einen SERS-Antrieb oder den SERS-Programmer zugewiesen werden.

Beispiel:

Z=0 bedeutet, Anweisungen sind für den SERS-Programmer bestimmt

Z=2 bedeutet, Anweisungen sind für den SERS-Antrieb mit Adresse 2 bestimmt

Aber es gibt verschiedene Anweisungen / Kommandos / Parameter, die bei Bezug auf den SERS-Programmer keinen Sinn machen würden (z.B. Start Positionierung oder Start Referenzfahren – diese Kommandos machen nur Sinn für einen SERS-Antrieb).

Folgende Anweisungen / Kommandos / Parameter werden daher immer automatisch der SERS-Antriebs-Adresse zugewiesen, die in P1185 definiert ist (also die niedrigste beim automatischen Adressenscan, nach dem Power-ON, gefundene Antriebsadresse). D.h. hier ist keine "Z=..." Anweisung notwendig:

"E", "S", "H", "RS", "LS", "RF", "LF", "RUN", "ON", "OFF"

"W", "V", "A", "WR", "WA", "B", "WP", "WRP", "WAP", "WPT", "WRPT", "WAPT"

"ADC", "I1".."I16", "J1".."J6", "POS", "LP", "O1".."O16"

Referenzfahr-Geschwindigkeit: P41, Referenzfahr-Beschleunigung: P42

Lage-Istwert 1 (Motorgeber): P51

Feedrate Override: P108

Referenzfahr-Parameter: P147, Referenzfahr-Geschwindigkeit langsam: P1003

Kommutierungstabelle: P1009

Treiberstatus und Endschalter: P1013

Positioniermodus: P1014

Beschleunigungsphase: P1015, Konstantphase: P1016

Handfahren -Beschleunigung: P1018

Handfahr-Geschwindigkeit langsam: P1019, Handfahr-Geschwindigkeit schnell: P1020

Rampenform: P1032

Endschalter und Digin: P1056

Nächsten Polynomabschnitt laden: P1123

Digin: P1300, P1301

3.4 Beschränkungen für E²Prom-Programme

Folgende Beschränkungen für ein E²Prom-Programm gelten :

- Programmspeicher: Anzahl der Programmzeilen hängt von der Programmart ab - 8KByte stehen zur Verfügung, je länger die einzelnen Zeilen sind, desto weniger Zeilen können programmiert werden. Bei Verwendung von nur einer Anweisung pro Zeile können ca. 1200 Zeilen programmiert werden.
- Maximal 60 Zeichen pro Zeile
- Maximal 128 Label – L1 bis L128 - (mit "speziellem" Label L65, der nach einem Programmstop angesprungen wird)
- Verschachtelungstiefe für Unterprogramme : max. 4 (ineinander verschachtelte Programmschleifen)

Syntax-Beispiele:**Beispiel 1:**

```
"\a"  
"#0R2?"  
"\e"
```

Registerwert R2 von Achse 0 (SERS-Programmer) wird angezeigt.

Falls z.B. R2 = 100

SERS-Programmer Anzeige:

100

Beispiel 2:

```
"\a"  
"PROGRAMMBEISPIEL"  
"REG2: #0R2!"  
"\e"
```

In der ersten Zeile des SERS-Programmers wird der Text "PROGRAMMBEISPIEL" angezeigt.

In der nächsten Zeile erscheint der Text "REG2 " und danach (in der selben Zeile) der Wert aus Register R2 (von Achse 0 - SERS-Programmer).

R2 kann editiert werden (da "!" programmiert wurde).

Falls z.B. R2 = 100

SERS-Programmer Anzeige:

PROGRAMMBEISPIEL

REG2: 100

Beispiel 3:

```
"\a"  
"PROGRAMMBEISPIEL"  
"REG2: #0R2!"  
"POS: #1P51?"  
"\e"
```

In der ersten Zeile des SERS-Programmers wird der Text "PROGRAMMBEISPIEL" angezeigt.

In der nächsten Zeile erscheint der Text "REG2 " und danach (in der selben Zeile) der Wert aus Register R2 (von Achse 0 - SERS-Programmer).

R2 kann editiert werden (da "!" programmiert wurde).

In der dritten Zeile des SERS-Programmers wird die Motorposition von Achse 1 angezeigt, mit vorangestelltem Text "POS: "

Falls z.B. R2 = 100, und aktuelle Motorposition Achse 1 = 360.5000

SERS-Programmer Anzeige:

PROGRAMMBEISPIEL

REG2: 100

POS: 360.5000

3.5 Funktionen für die Programmierung von Ablaufprogrammen

3.5.1 IF : Bedingte Ausführung

Syntax : **IF** *expression*

oder **IF** *expression*
 THEN ... Anweisungsblock ...
 [ELSE] ... Anweisungsblock ...
 END

expression: *parameter vergleich parameter* | *festkommakonstante*

Beispiel: P11<P12 X>=V M1=O1 X<=-123.456 V>3000.0 O1<>O2

parameter: Es sind alle SERS-Parameter und digitalen Eingänge (I1 bis I16, und J1 bis J6) erlaubt. Es können die Parameternummern und die alternativen Namen verwendet werden (z.B. "V" oder "P91", was beides den selben Parameter bezeichnet).

vergleich: < <= = <> > >=

festkommakonstante: - 2147483.639 bis +2147483.639

IF-Struktur ohne "THEN ...":

Falls "**expression**" wahr ist, dann wird das nächste Kommando ausgeführt.

Falls "**expression**" nicht wahr ist, wird das nächste Kommando übersprungen.

Alle danach folgenden Kommandos werden unabhängig der IF-Abfrage ausgeführt.

IF-Struktur mit "THEN ...":

Falls "**expression**" wahr ist, dann werden alle Kommandos (Anweisungsblock)

zwischen "THEN" und ELSE ausgeführt. Falls kein "ELSE" vorhanden ist (also Struktur nur IF ... THEN ... END), dann werden alle Kommandos zwischen "THEN" und END ausgeführt.

Falls "ELSE" vorhanden ist, und "**expression**" nicht wahr ist, dann werden alle Kommandos zwischen "ELSE" und "END" ausgeführt.

Für **digitale Eingänge** gelten folgende Vereinbarungen:

- **I1 bis I8** : wenn einer der Eingänge gesetzt ist, dann ist das Ereignis wahr.
 z.B.: IF I5=1 Wenn Eingang I5 gesetzt ist, dann nächstes Kommando ausführen,
 sonst nächstes Kommando ignorieren / überspringen
 IF I5=0 Wenn Eingang I5 nicht gesetzt ist, dann nächstes Kommando
 ausführen sonst nächstes Kommando ignorieren / überspringen
- **IN0 bis IN255** : mit IN wird die anstehende Eingangskombination abgefragt,
 wobei die Eingänge binär codiert sind (I1=1, I2=2, I3=4, I4=8, I5=16, I6=32,
 I7=64, I8=128)
 z.B.: IF IN=5 Wenn I1 und I3 gesetzt sind (I1 + I3 = 1 + 4 = 5), dann nächstes
 Kommando ausführen
- **J1** - Referenzschalter (IF J1=1 → falls der Antrieb auf dem Referenzsch. steht)
- **J2** - Endschalter-Links (IF J2=0 → falls der Antrieb auf dem Endsch.-Links steht)
- **J3** - Endschalter-Rechts
- **J4** - Stopschalter
- **J5** - Serviceschalter-extern
- **J6** - Nullpunkt (bei Drehgebern mit Nullpunkt)

IF-Abfrage der Zähler C1, C2 und C3 (P100, P101 und P102):

Hier wird bei jeder IF-Abfrage der Zähler um 1 nach unten gezählt

IF C1>1 → C1=C1-1 und falls C1>1 führe nächste Anweisung aus

Mit dieser Struktur können Zählerschleifen generiert werden.

Typische Abfragen von SERS-Antrieben:

IF POS=0 ... Überprüfen, ob der Antrieb noch positioniert – nach Start einer Positionierung eines SERS-Antriebs mit “E”

POS=0 → Antrieb positioniert noch, POS=1 → Antrieb steht

IF RUN ... Überprüfen, ob ein im Antrieb mit “RUN“ gestartetes Programm noch läuft
 RUN → Programm läuft noch

Beispiele für IF-Abfragen:

Beispiel 1: IF P12<>0 O1=1
 GT 20

Falls P12 ungleich 0 ist (es liegt eine Warnung vor) dann wird zuerst Ausgang O1=1 gesetzt (O1=1 ist nächste Anweisung nach der IF-Abfrage), und danach zu Label 20 gesprungen (GT20).

Falls P12 gleich 0 ist (P12<>0 also falsch ist), dann wird die folgende Anweisung O1=1 nicht ausgeführt, sondern übersprungen, und nur die danach folgende Anweisung GT 20 ausgeführt.

Folgendes Programm ist funktionell identisch mit obigem Programm:

```
IF P12<>0
O1=1
GT20
```

Achtung: Ob die der IF-Abfrage folgende Anweisung in der nächsten Zeile oder in der selben Zeile wie die IF-Abfrage steht, ist unerheblich.

Beispiel 2: IF P12<>0 THEN
 O1=1 O2=0 O3=1
 ELSE
 O1=0 O2=1 O3=0
 END
 GT 20

Wenn P12 ungleich 0 ist, dann werden die Kommandos (nach THEN) O1=1, O2=0 und O3=1 ausgeführt. Falls P12 gleich 0 ist, dann werden die Kommandos (nach ELSE) O1=0 O2=1 O3=0 ausgeführt.

Beispiel 3: C1=10
 L1
 WAIT I1=1
 WR=180 E
 IF C1>1 GT1

Der Anweisungsblock mit “WAIT I1=1“ und “WR=180 E“ wird 10 mal ausgeführt.

Beispiel 4: IF ADC>R0 THEN
 X=ADC*1000 V=X WR=3600 E
 END

Wenn der Wert am analogen Eingang ADC größer als der in R0 gespeicherte Wert ist, dann führe die Anweisungen “X=ADC*1000 V=X WR=3600 E“ aus.

ACHTUNG: Es dürfen max. 64 Strukturen mit THEN .. ELSE .. END in einem Programm vorhanden sein!

3.5.2 Label : Programm-Marken

Syntax : L *dezimalkonstante* - dezimalkonstante ist ein Wert von 1 bis 128

z.B: L1 oder L64

Label werden verwendet als Sprungziele, um Programmschleifen oder Unterprogramme zu erzeugen. Jede Labelnummer darf nur einmal im Programm definiert sein !

Bei P1033=2: Der Label L65 wird angesprungen wenn ein E²PROM-Programm unvorhergesehen beendet wird – z.B. durch einen externen Stop-Befehl oder einen Antriebsfehler (z.B. Temperaturfehler). Dadurch können im Fehlerfall noch Aktionen im Programm ausgeführt werden.

Wenn der Label L65 nicht existiert wird das Programm bei einem Antriebsfehler beendet.

3.5.3 GOTO, GT : Verzweigung

Syntax : GOTO *Label* - Label muss irgendwo im Programm definiert sein

Alternative Syntax : GT *Label*

Programmsprung zu einem Label

```

L1
...
GOTO 1          // äquivalent ist : #GT 1
oder
GOTO 1
...
L1

```

3.5.4 GOSUB : Unterprogrammaufruf

Syntax : GOSUB *Label* - Label muss irgendwo im Programm definiert sein

Alternative Syntax : GS *Label*

Sprung zu einem Unterprogramm

– bei einem Unterprogramm steht am Beginn ein Label und am Ende ein 'RETURN'

z.B.:

```

GOSUB 12        // äquivalent ist #GS 1
...
L12             // Beginn Unterprogramm
...
RETURN         // Ende Unterprogramm und Rücksprung

```

3.5.5 RETURN : Unterprogrammabschluss

Syntax : RETURN

Alternative Syntax : RT

Das RETURN schließt ein mit 'GOSUB' aufgerufenes Unterprogramm ab und bewirkt den Rücksprung zu der Anweisung, die im Programm nach dem 'GOSUB' steht.

Ein RETURN im Paralell-Betrieb beendet das aufgerufene Unterprogramm (Aufruf durch Signal an Starteingang und Adresse am I/O-Port). Die SERS wartet danach auf den nächsten Unterprogrammaufruf.

3.5.6 Programmierung von Positioniervorgängen

Eine Positionierung erfolgt durch die Parameter **Beschleunigung** (Syntax : **A=Wert**), **Geschwindigkeit** (Syntax : **V=Wert**) und **Weg/Position** (Syntax : **W=Wert**) und dem Befehl **'Starten Positionieren'** (Syntax : **E**).

Der Befehl **'E'** startet eine Positionierung mit den aktuellen (im Speicher der SERS vorhandenen) Werten der Parameter A, V und W.

Nach dem Einschalten der SERS gelten die im E²PROM abgespeicherten Parameter A und V. Der Parameter W ist nach dem Einschalten immer 0 (es sei denn P1117=1 und W wurde dadurch mit PSAVE wie die anderen Parameter im E²Prom abgespeichert).

Die Parameter A und V können jederzeit neu definiert werden. Wenn diese Parameter überschrieben werden, während die SERS positioniert, werden sie erst für den nächsten Positioniervorgang gültig.

Ein Überschreiben von W ist nur möglich, wenn der Antrieb steht, also bei POS=1, oder wenn der Antrieb mit konstanter Geschwindigkeit läuft (Konstantphase). Beim Überschreiben von W, während der Konstantphase, wird im relativen Positioniermodus der neue Wert W zum alten Wert W addiert und der Antrieb verfährt insgesamt den neuen Gesamtweg W. Im absoluten Positioniermodus fährt der Antrieb auf den neu geschriebenen Wert W. Die während der Konstantphase gesendeten Werte W dürfen nicht zu einer Drehrichtungsumkehr des Motors führen. In diesem Fall wird der gesendete Wert W mit einer Fehlermeldung quittiert und erst nach dem Fertig-Positionieren mit der nächsten Anweisung **'E'** (Starten Positionieren) berücksichtigt.

Zusätzlich zu den Standardparametern gibt es noch weitere Parameter, die die Positionierung beeinflussen :

- **Positioniermodus** P1014 (**Relativ** oder **Absolut Positionieren**)
- **Rampenform der Beschleunigung** :
 - P1032 (exponentielle oder Sinus-Rampenform)
 - P1005 (Beschleunigung Abschnitt 1)
 - P1006 (Beschleunigung Abschnitt 2)
 - P1007 (Geschwindigkeit Abschnitt 1)
 - P1008 (Geschwindigkeit Abschnitt 2)
- **Wichtungsarten** : P160 (Beschleunigung)
 - P44 (Geschwindigkeit)
 - P76 (Weg – Positionsdaten)

Alternativ zur Anweisung W=Wert gibt es die Kommandos:

WR=Wert : Positioniermodus Relativ (P1014=0) **und** W=Wert (z.B WR=1000)

WA=Wert : Positioniermodus Absolut (P1014=2) **und** W=Wert (z.B WA=1000)

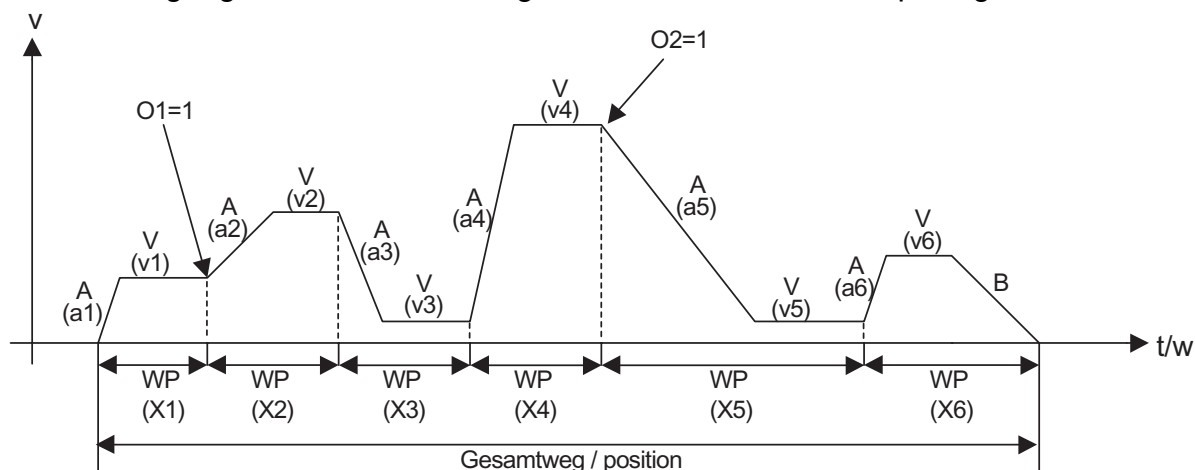
Mit folgenden Parametern kann der aktuelle Status eines Positioniervorgangs abgefragt werden:

- **P336** (alternativ **'POS'**) – POS = 1 wenn Positions-Ist-Wert = Positions-Soll-Wert
- **P1015 (Beschleunigungsphase)** = 1 während der Antrieb beschleunigt
- **P1016 (Konstantphase)** = 1 während der Antrieb mit konstanter Geschwindigkeit läuft

Im Programm-Ablaufbetrieb geht das Programm erst zur nächsten Anweisung weiter, wenn POS=1 ist. (Für den Masterbetrieb kann dies aber dem Parameter P1110 umgangen werden)

3.5.7 Positionieren mit Geschwindigkeitsprofilen - Polynomfahren

Es soll der Weg $W = X1 + X2 + X3 + X4 + X5 + X6$ positioniert werden. Dabei soll mit $a1$ auf die Geschwindigkeit $v1$ beschleunigt werden und mit $v1$ weiter gefahren werden bis der Wegabschnitt $X1$ beendet ist. Danach wird im Wegabschnitt $X2$ mit $a2$ auf die Geschwindigkeit $v2$ beschleunigt. Die Wegabschnitte $X3$ bis $X5$ werden analog zu $X1$ und $X2$ gefahren. Der letzte Abschnitt $X6$ wird durch die Beschleunigung $a6$, die Geschwindigkeit $v6$ und die Bremsrampe B gebildet.



Prinzip:

Ein Geschwindigkeitsprofil besteht aus einem oder mehreren Wegabschnitten. Jeder Wegabschnitt besteht aus einer Beschleunigung (Rampe) und einem Abschnitt mit konstanter Geschwindigkeit. Nur der letzte Wegabschnitt hat noch zusätzlich eine Bremsrampe B .

Ein Wegabschnitt beim Positionieren mit Geschwindigkeitsprofilen wird durch das Kommando "WP" definiert. Die aktuell gesetzte Beschleunigung A und die Geschwindigkeit V können für jeden Wegabschnitt neu definiert werden oder unverändert beibehalten werden.

Innerhalb der Definition eines Wegabschnittes muss zuerst das Kommando "WP" stehen.

Der letzte Wegabschnitt wird durch das Kommando "WPT" definiert.

Jede Wegabschnittsdefinition wird mit dem Kommando "E" abgeschlossen.

Wenn nach Beendigung eines Wegabschnittes SERS-Ausgänge, Merker oder beliebige andere Parameter gesetzt werden sollen (z.B. werden in obigem Diagramm die Ausgänge $O1$ und $O2$ gesetzt), **dann gilt:** Zuerst muss der nächste Wegabschnitt definiert werden (während der aktuelle Wegabschnitt noch verfahren wird), dann darf der Parameter, Ausgang oder/und Merker definiert werden, der nach Beenden des aktuellen Wegabschnittes gesetzt werden soll.

Vorzeitig beendet werden (ohne WPT) kann der Polynommodus mit "POS0" oder "POSR" - siehe Parameter P1043. Auch nach einem Fehler (z.B. durch eine falsche Polynomdefinition) bleibt der Polynommodus solange aktiv, bis er mit einem Kommando beendet wird

Obiges Beispiel mit 6 Wegabschnitten sieht als Ablaufprogramm folgendermaßen aus:

```
WP=100 A=1000 V=200 E WAIT P1123=1
WP=120 A=300 V=400 E WAIT P1123=1
O1=1
WP=100 A=1000 V=50 E WAIT P1123=1
WP=100 A=1500 V=700 E WAIT P1123=1
WP=200 A=500 V=100 E WAIT P1123=1
O2=1
WPT=130 A=1000 B=500 V=250 E WAIT P1123=1
```

Der Ausgang O1 wird nach Beendigung des ersten Wegabschnittes gesetzt (obwohl das Kommando O1=1 erst nach der Definition des zweiten Wegabschnittes steht !). Der Ausgang O2 wird nach Beendigung des vierten Wegabschnittes gesetzt (O2=1 steht im Programm aber nach der Definition des fünften Wegabschnittes !).

Die Definition der Beschleunigung innerhalb eines Wegabschnittes ist optional. Falls die Beschleunigung nicht explizit definiert wird, dann wird die aktuelle Beschleunigung A für alle Wegabschnitte verwendet (mit Ausnahme der letzten Bremsrampe → hier gilt Parameter B)

Wie beim einfachen Positionieren mit dem "W"-Parameter ist auch beim Positionieren mit Geschwindigkeitsprofilen der Parameter "WP" abhängig vom Positioniermodus P1014 (relativ oder absolut Positionieren).

Alternativ zu WP gibt es folgende Kommandos:

WAP : Wegabschnitt absolut (z.B. WAP=1000)

WRP : Wegabschnitt relativ (z.B. WRP=500)

Und alternativ zu WPT gibt es:

WAPT : letzter Wegabschnitt absolut (z.B. WAPT=1000)

WRPT : letzter Wegabschnitt relativ (z.B. WRPT=1000)

Wenn die Polynomabschnitte einzeln über die serielle Schnittstelle (RS232) zum SERS-Antrieb gesendet werden, dann zeigt die SERS über Parameter P1123 (P1123=1) an, wann der nächste Polynomabschnitt gesendet werden kann. Die SERS muss den nächsten Polynomabschnitt erhalten, bevor der aktuelle Abschnitt zu Ende gefahren wurde, andernfalls wird eine Fehlermeldung erzeugt und der Antrieb bleibt stehen !

3.5.8 WAIT Anweisung

Bei "WAIT" wird die Programmausführung gestoppt, solange das definierte Ereignis noch nicht eingetreten ist.

Syntax: WAIT *expression*

“**expression**“ ist definiert wie die IF-Abfrage in Kapitel 3.5.1 (Seite 30)

Beispiele:

WAIT I1=1 (das Programm wartet bis Eingang I1=1 wird)

WAIT ADC>0.5 (das Programm wartet bis der Wert am Analogeingang ADC>0.5 wird)



ACHTUNG:

Eine Anweisung mit D2=... GS oder D2=... GT... beendet die WAIT-Funktion (siehe Beschreibung Parameter P1164 - D2 – in Kapitel 3.6).

3.5.9 “ : “ Anweisung - Handfahren bis STOP durch Eingang

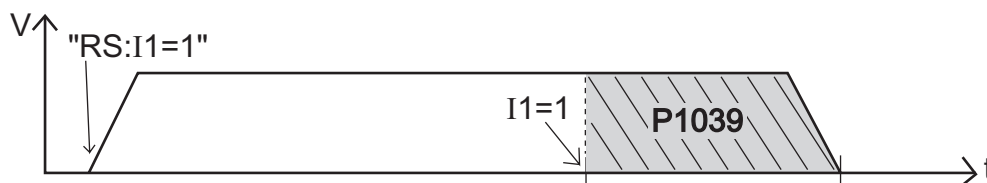
Handfahren bzw. Fahren mit konstanter Geschwindigkeit bis ein Ereignis eintritt. Dann wird der Antrieb sofort mit der Handfahrbeschleunigung P1018 bis zum Stillstand abgebremst.

Syntax: RS:*expression* (alternativ zu **RS** sind noch erlaubt : **LS**, **RF** und **LF**)

“**expression**“ ist definiert wie die IF-Abfrage in Kapitel 4.6.1 (Seite 32)

Beispiel: **RS:I1=1** → der Antrieb fährt im Handfahrmodus (rechts langsam - mit der Geschwindigkeit P1019) solange, bis Eingang I1 gesetzt wird.

Wenn der Parameter **P1039** <> 0 ist (ein **Nachlaufweg** ist definiert), dann fährt der Antrieb nach Eintreten des Ereignisses (Eingang wird aktiv, z.B. bei RS:I1=1 bzw. Eingang wird deaktiviert bei z.B. RS:I1=0) noch den definierten Nachlaufweg P1039 weiter und stoppt erst dann. Der Nachlaufweg beinhaltet den Bremsweg.



ACHTUNG:

Eine Anweisung mit D2=... GS oder D2=... GT... beendet die “ : “ - Funktion (siehe Beschreibung Parameter P1164 - D2 – in Kapitel 3.6).



ACHTUNG:

Vor Verwendung der “ : “ – Funktion muss vorher P1035=0 gesetzt worden sein (kann zu irgendeinem Zeitpunkt im Programm, vor der “ : “ – Funktion erfolgt sein).

Beispiel :

Z=1 P1035=0

RS:I1=1

3.5.10 Arithmetik / Rechnen mit der SERS

Ein SERS Programmer E²PROM Programm kann Arithmetikfunktionen enthalten.

Dazu stehen folgende Funktionen zur Verfügung:

- ein 32-Bit Akkumulator **X** dem alle Rechenoperationen zugewiesen werden müssen
- 100 Register (32-Bit Variablen) R0 bis R99, die frei verwendet werden können
- Folgende Rechenoperationen gibt es
 - Addieren '+', Subtrahieren '-', Multiplizieren '*', Dividieren '/'
 - Und '&', Oder '|', Exklusiv-Oder '^'
 - **NEG** : der Akkuinhalt wird negiert (Vorzeichen gewechselt)
- Die Operationen werden von links nach rechts ausgeführt (kein Punkt vor Strich).
- Rechenoperationen müssen immer mit dem Akkumulator X ausgeführt werden !
- Es können alle Parameter und 32-Bit Konstanten in Rechenoperationen verwendet werden.

Beispiele:

```
X=V*ADC+100 //Die momentan eingestellte Geschwindigkeit V wird mit dem Wert
V=X          //am Analogeingang ADC (0 – 100%) multipliziert, und der Wert
100 addiert
V=X          //

R1=2
X=V*ADC*3
R0=X
X=W+10000/R0*R1

X=W
NEG //Der Akku-Inhalt wird negiert (X=-X)
W=X
```

3.6 Parameter für erweiterten Programm-Modus

Parameter im SERS-Programmer:

P100 (alternative Syntax **C1**) Zähler 1
vorzeichenloser 16 bit Zähler (0 bis 65536)

P101 (alternative Syntax **C2**) Zähler 2
vorzeichenloser 16 bit Zähler (0 bis 65536)

P102 (alternative Syntax **C3**) Zähler 3
vorzeichenloser 32 bit Zähler (0 bis $4,295 \times 10^9$)

Mit Zählern und IF-Abfragen können z.B. Wiederholschleifen programmiert werden.

Bsp.: **#C1=20**
 #L1 Bei dem Beispiel wird der
 Anweisungsblock *Anweisungsblock* 20 mal
 #IF C1>1 GT 1 ausgeführt.

P1117: Optionen Parameter Abspeichern

Über das Kommando PSAVE (P1004=2) können alle Parameter (bis auf Register Rx und Merker Mx) in das E²PROM des SERS-Programmers bzw. der SERS-Antriebe gespeichert werden.

Bei einem vorhandenen Anzeigemenü im SERS-Programmer können Parameter editiert werden (siehe Kapitel 3.3.1 Syntax Programmierung Anzeige Menü, Steuerzeichen "!"). Das Editieren erfolgt über Taste "F1", wobei der neue Wert dann mit der "RET" Taste übernommen wird. Geändert (editiert) werden die Werte aber beim SERS-Programmer bzw. in den SERS-Antrieben nur im jeweiligen RAM (d.h. nach Ausschalten der Versorgungsspannung sind die Änderungen verloren).

Sollen die geänderten Werte dauerhaft gespeichert werden, dann muss das durch Abspeichern im E²PROM, über das Kommando PSAVE, erfolgen.

Ob PSAVE sich auf dem SERS-Programmer oder einen SERS-Antrieb bezieht, hängt von der aktuellen gültigen Zuordnung über "Z=..." ab.

Z=0 → SERS-Programmer, Z=1 → SERS-Antrieb mit Adresse 1, Z=2 ... usw.

Mit folgenden Optionen können auch die Register Rx über Kommando PSAVE abgespeichert werden, und das Kommando PSAVE kann automatisch beim Editieren (Taste "F1") von Werten im Anzeigemenü erzeugt werden.

Bit 0 (dez 1) : Bei der Anweisung P1004=2 (Kommando "PSAVE") werden zusätzlich zu den Standardparametern die Register Rx (im SERS-Programmer "R0" bis "R99" und in SERS-Antrieben "R0" bis "R11") im E²PROM abgespeichert. PSAVE wird aber nicht automatisch erzeugt bei Edit über Taste "F1"

Bit 1 (dez 2) : Das Kommando PSAVE, zum Abspeichern der Standardparameter im E²PROM, wird automatisch erzeugt, nach Editieren über Taste "F1"

Bit 2 (dez 4) : PSAVE mit Abspeichern auch der Register Rx (im SERS-Programmer "R0" bis "R99" und in SERS-Antrieben "R0" bis "R11") im E²PROM, und PSAVE wird automatisch erzeugt, nach Editieren über Taste "F1"

Das Speichern aller 100 Register (R0 bis R99) im E²PROM des SERS-Programmers dauert ca 1.2 Sekunden.

Während dieser Zeit wird "save..." am SERS-Programmer angezeigt.

P1162: letzter Tastenwert (Codierung siehe Kapitel 3.7), flankengetriggert, d.h. der Tastendruck seit der letzten Abfrage (z.B. mit X=P1162) wird gespeichert und nach einer Abfrage gelöscht.

Achtung: Bei programmierten Schleifen, in denen P1162 bei jedem Schleifendurchlauf abgefragt wird, geht ein Tastendruck somit verloren, wenn er nicht zwischengespeichert wird.

P1162 kann auch beschrieben werden, und beschreibt auch den Tastaturwert für die Menüauswahl, der extra gespeichert wird. Damit kann verhindert werden, dass die gleiche Taste mehrfach ausgewertet wird.

P1100: (alternative Syntax D) Zeitverzögerung

Einheit 100ms, Bereich 0.0 bis 6553.5 (=655 Sekunden)

Kleinste programmierbare Wartezeit: 10ms (D=0.1)

Beispiel.: **D=20** → 2s Wartezeit oder **D=0.5** → 50 ms Wartezeit

Die Programmfortsetzung wird für die Zeit **D** angehalten.

P1164: (alternative Syntax: D2) Wartezeit

Einheit 100ms, Bereich 0.0 bis 6553.5 (=655 Sekunden)

Kleinste programmierbare Wartezeit: 10ms (D=0.1)

Die Wartezeit D2 (=Timer D2) läuft im Hintergrund, parallel zum Programm, ab, und setzt bei Ablauf das Bit DB (=P1165).

Folgende Abfragen werden wahr, wenn die Wartezeit D2 abgelaufen ist:
"IF DB", "IF DB=1", "IF D2=0"

Wenn dem D2 ein GT oder GS folgt, dann wird der Sprung ausgeführt wenn die Wartezeit abgelaufen ist (z.B. D2=10 GT30 W=360 E ...)

Das Programm geht ohne Sprung nach dem GT oder GS weiter. D.h. in obigem Beispiel wird "W=360 E" (und alle nachfolgenden Kommandos) sofort nach "D2=10" ausgeführt. Der Programmsprung "GT30" wird erst nach Ablauf der Wartezeit D2 ausgeführt.

Die Wartezeit kann auch mit D2=Wert nachgeladen (verlängert) werden solange der Sprung nicht ausgeführt wurde.

ACHTUNG: Nach dem Sprung durch Ablauf der Wartezeit D2 werden alle der folgenden "Blockaden" zurückgesetzt:

- 1) Wartezeit D (P1110) wird auf 0 gesetzt.
- 2) "WAIT" und Stop (:) ... werden zurückgesetzt und es wird nicht weiter gewartet.
- 3) Auf von einem Slave abgefragte Werte wird nicht weiter gewartet (z.B. Z=3 X=P51 oder Z=3 IF POS=1).

Bei einem Unterprogrammaufruf durch eine abgelaufene Wartezeit (z.B. D2=10.0 GS 99) ist zu beachten, dass die Unterbrechung zu jedem beliebigen Zeitpunkt erfolgen kann, und das Programm auch an dieser Stelle fortgesetzt wird. Zum Fortsetzungszeitpunkt sind aber auch alle Blockaden (Fälle 1 bis 3 - siehe oben) aufgehoben.

P1165: (alternative Syntax: DB) Wartezeit Bit wird gesetzt bei Ablauf der Wartezeit D2



P1182: Parameter-Zeilenummer

Bei programmierten Anzeigemenüs, die mit "\a" beginnen, entspricht der Wert für P1182, nach dem Einschalten des SERS-Programmers, der Zeilenummer (zwischen 1 und 4), die den ersten editierbaren Parameter enthält.

Bei Listen, die mit "\b" beginnen, ist der Cursor nach dem Einschalten immer in der ersten Zeile und damit P1182=1.

Mit den "hoch" / "runter" (↑ / ↓)Tasten wird der Cursor im Anzeigemenü, und damit auch P1182 entsprechend erhöht bzw. erniedrigt.

Durch Beschreiben von P1182 (z.B. P1182=3) kann der Cursor im Anzeigemenü auf eine beliebige Zeile gesetzt werden.

Beim Schreiben von P1182=0 wird kein Cursor dargestellt, für eine ungestörte LCD Anzeige. Die Ausgabe des Anzeigemenüs muss aber nach P1182=0 mindestens noch einmal erfolgen.

Mit P1182 lässt sich die zuletzt angezeigte Parameternummer sichern und für neue Parameterlisten initialisieren. Auch eine Begrenzung der erreichbaren Parameter ist damit möglich.

P1183: FB Option, default: 0

Bit 0 (dez 1): Reset Freigabe (Tasten "ESC" und "." gleichzeitig auf der FB)
P1183=1 gibt den Reset frei

→ P1183=1 ist sinnvoll für den Test-Betrieb, während der Entwicklungsphase. Dadurch kann einfach zwischen Programmier-Modus und erweitertem Programm-Modus gewechselt werden (nur über Tastendruck, ohne Spannung Aus/Ein)

Bit 1 (dez 2): Wenn D2 abgelaufen ist und im Ablaufprogramm ein Label angesprungen werden soll (durch D2=..GT.. oder D2=..GS..), dann wird nicht gewartet auf die folgenden Aktionen, die das Programm anhalten:

- a) die Bestätigung von Fehlerquittierungen
 - b) Ablauf von Timeout-Zeiten der seriellen Schnittstelle (der Timeout wird auf 20ms verkürzt)
 - c) Beendigung von vordefinierten Menüs (P1187)
- bei P1183=2 wird sofort gesprungen (nach Ablauf D2)

Bit 2 (dez 4): nach dem Quittieren einer Fehlermeldung wird die Programm-Zeilenummer im LCD angezeigt:

Z.B. "Zeile:0114" wenn in Zeile 114 ein Syntaxfehler existiert.

Dadurch ist das Auffinden der verursachenden Programmstelle möglich.

P1184: aktuellen Tastenwert (Codierung siehe Kapitel 3.7) lesen
braucht ca. 10ms Zeit, da die Entprellung durchlaufen wird
Für Tastaturauswertungen mit Flanke sollte P1162 verwendet werden.

P1185: Antriebsadresse

Vorbelegung durch den Adressenscan am Start (von Adresse 9 bis 1, die niedrigste vorhandene wird gewählt). Das Adressmenü (aufrufbar durch P1187=2) weist P1185 auch zu. P1185 ersetzt das "a" (z.B. "#aW?") für die Adressierung im Textstring, gelesen wird die aktuelle Adresse.

P1185 kann aber auch beschrieben werden, um so die folgenden Kommandos dem entsprechenden SERS-Antrieb zuzuweisen (z.B. P1185=2)

P1186: Baudrate SERS-Programmer:

1=1200 2=2400 3=4800 4=9600 5=19200 (default) 6=38400

alle anderen Werte: 19200 Baud bei Programmier-Modus, und automatischer Scan (der angeschlossenen SERS-Antriebe) bei Ausführung (im erweiterten Programm-Modus)

P1187: Menü-Aufruf (siehe auch Kapitel 3.8 "Vordefinierte Menüs")

Wertebereich 0..9

Durch Zuweisung der Menünummer wird das vordefinierte Menü ausgeführt (z.B. P1187=2 Adressmenü).

Das Programm wird während der Ausführung des Menüs angehalten.

Die Menüs funktionieren wie beim SERS-Programmer im Standardmodus, jedoch wird beim Verlassen des Menüs durch ESC oder RET das Programm weiter ausgeführt, und nicht in das übergeordnete Menü gesprungen.

Wenn etwas gesendet wird, dann immer an die gewählte Antriebsadresse (die in P1185 steht).

Aufrufbare Menüs sind in Kapitel 3.8 beschrieben.

P1188: pro Adresse wird vom SERS-Programmer ein Bit gesetzt wenn der Antrieb vorhanden ist:

Antrieb 1: 1	Antrieb 2: 2	Antrieb 3: 4	Antrieb 4: 8
Antrieb 5: 16	Antrieb 6: 32	Antrieb 7: 64	Antrieb 8: 128
Antrieb 9: 256			

Beispiel: wenn Antriebe 1, 3 und 7 vorhanden sind: P1188=69

P1189: Fehler-Maske für L65 Aufruf (default: 3)

Label L65 wird nach der Quittierung eines Fehlers (mögliche Fehler sind die unter den untenstehenden drei Bits beschriebenen) nur dann aufgerufen, wenn in P1189 die entsprechenden Bits gesetzt sind.

Bit 0 (dez 1): Timeout der seriellen Schnittstelle (nach ca. 4 s)

Bit 1 (dez 2): Im Adressmenü wurde eine ungültige Adresse ausgewählt

Bit 2 (dez 4): Antrieb meldet Fehler zurück

Das Ablaufprogramm bleibt so lange stehen, bis der Anwender die Fehleranzeige durch Betätigen einer Taste quittiert hat. Danach wird L65 aufgerufen (wenn das entsprechende Bit in P1189 gesetzt ist), um einen geordneten Weiterlauf zu ermöglichen.

Die Zieladresse "Z" wird vor dem Sprung zu dem Fehlerlabel "L65" auf die eigene Adresse (Z=0) gesetzt.

Wenn L65 nicht definiert ist, dann wird das Programm neu gestartet.

R0 bis R99: frei verfügbare Register

R12 bis R99 sind bei Konstantenzuweisung auf 2 Nachkommastellen begrenzt
Beispiel: R12=20.5682 → es wird nur R12=20.56 übernommen

M1 bis M20: frei verfügbare Merker (Wertebereich: "0" und "1")

Merker können nicht im E²PROM abgespeichert werden, und werden daher nur zur Laufzeit im RAM gespeichert.



Alle Parameter für SERS-Antriebe sind in den Handbüchern "wsers_v....pdf" bzw. "sers_v...pdf" beschrieben.

3.7 Tastenwerte bei Tastaturabfrage

Tastenwerte (für Abfrage mit P1162 und P1184)

F1 = 240

F2 = 241

F3 = 242

F4 = 243

ESC = 244

Stop = 245

↑ = 246

Pgup = 247

RET = 248

← = 249

. = 46

→ = 251

List = 252

– = 45

↓ = 254

Pgdwn = 255

3.8 Vordefinierte Menüs

Über Zuweisungen der entsprechenden Nummer an den Parameter P1187 können diese Menüs aufgerufen werden.

"**LCD:**" zeigt den Bildschirmaufbau nach dem Aufruf.

"**Tasten:**" benennt die aktiven Tasten und deren Reaktion.

Wenn etwas gesendet wird, dann immer an die zuletzt gewählte, bzw. über den Autoscan gefundene Antriebsadresse (P1185).

P1187=

0: Z_MENU_MASTER (kein vordefiniertes Menü)

Anzeige und Auswahl von Parametern entsprechend dem Ablaufprogramm.

LCD:

Die Anzeige wird durch das Anwenderprogramm vorgegeben. (z.B. Ausgabe zwischen "\a..." und "\e").

Der Cursor wird auf die ausgewählte Parameter-Zeilenummer positioniert.

Tasten: (werden durch die Anweisung "\e" ausgewertet)

"hoch" verringert die Parameter-Zeilenummer (P1182) um 1 (wenn möglich)

"runter" erhöht die Parameter-Zeilenummer (P1182) um 1 (wenn möglich, maximal Anzahl der Zeilen zwischen "\a" und "\e")

F1 editiert den ausgewählten Parameter

1: Z_MENU_SPRACHE

LCD: **F1=AUTO ADR/BAUD**
 F2=MAN ADR/BAUD
 F3=DEUTSCH
 F4=ENGLISH

Tasten:

ESC oder RET verlässt das Menü, Sprung zu Z_MENU_MASTER

STOP sendet "s"

F1 sucht die Baudrate und danach die Antriebsnummer (nur 0..3)

Wenn keine Baudrate ermittelt werden kann, wird zu Z_MENU_BAUDRATE verzweigt

F2 verzweigt zu Z_MENU_ADRESSE

F3 sendet "p265=0psave"

F4 sendet "p265=1psave"

2: Z_MENU_ADRESSE

LCD: **F1=AUSWAHL BAUD**

AUSWAHL ACHSE ?

3

Tasten:

ESC oder RET liest die Sprache (P265) aus dem Antrieb und verlässt das Menü, Sprung zu Z_MENU_MASTER

F1 verzweigt zu Z_MENU_BAUDRATE

"hoch" erhöht die Antriebsadresse (P1185)

"runter" verringert die Antriebsadresse (P1185)

3: Z_MENU_BAUDRATE

LCD: **AUSWAHL**
 BAUDRATE
 ?
 38400

Tasten:

ESC oder RET verlässt das Menü, Sprung zu Z_MENU_MASTER

"hoch" wählt die nächst höhere Baudrate aus

"runter" wählt die niedrigere Baudrate aus

Die Schnittstelle wird nach dem Verlassen auf die neue Baudrate initialisiert.

4: Z_MENU_RESET_ERROR

LCD: **F1=RESET FEHLER**
 F2=RESET WARNUNG

Tasten:

ESC oder RET verlässt das Menü, Sprung zu Z_MENU_MASTER ASTER

F1 sendet "p11=0" und zeigt einen Quittierungstext

F2 sendet "p12=0" und zeigt einen Quittierungstext

5: Z_MENU_PARAMETER

LCD:
F1=EDIT F2=SAVE
P0=0
Z=0
P2=0
usw. ...

Tasten:

ESC oder RET verlässt das Menü, Sprung zu Z_MENU_MASTER

STOP sendet "s"

F1 editiert den angewählten Parameter

F2 sendet "psave"

"Pgup" verringert die Parameter-Zeilenummer um 10 (wenn möglich)

"hoch" verringert die Parameter-Zeilenummer um 1 (wenn möglich)

"Pgdn" erhöht die Parameter-Zeilenummer um 10

"runter" erhöht die Parameter-Zeilenummer um 1

6: Z_MENU_HAND

LCD: **F1=LANGSAM LINKS**
 F2=LANGSAM RECHT
 F3=SCHNELL LINKS
 F4=SCHNELL RECHT

Tasten:

ESC oder RET verlässt das Menü, Sprung zu Z_MENU_MASTER

STOP sendet "s"

F1 sendet "P11=0 P1035=1" und "ls"

(jeweils mit Wiederholung solange die Taste gedrückt bleibt, danach "P1031=0")

F2 sendet "P11=0 P1035=1" und "rs"

F3 sendet "P11=0 P1035=1" und "lf"

F4 sendet "P11=0 P1035=1" und "rf"

OFF sendet "off"

ON sendet "on p11=0 p12=0"

7: Z_MENU_REFERENZ

LCD: **F1=START**
 REFERENZFAHRT
 F2=STOP

Tasten:

ESC oder RET verlässt das Menü, Sprung zu Z_MENU_MASTER

F1 sendet "h"

F2 oder STOP sendet "s"

OFF sendet "off"

ON sendet "on p11=0 p12=0"

8: Z_MENU_START (Programmstart)

LCD: **F1=START F4=STOP**
 IN:255 O1:0 O2:0
 O3:0 O4:0
 0.0000-ok

Tasten:

ESC oder RET verlässt das Menü, Sprung zu Z_MENU_MASTER

F1 sendet "run"

F4 oder STOP sendet "s"

OFF sendet "off"

ON sendet "onp11=0p12=0"

9: Z_MENU_TERMINAL

LCD:

wird gelöscht und dann jedes empfangene Zeichen angezeigt.

Tasten:

ESC oder RET verlässt das Menü, Sprung zu Z_MENU_MASTER

senden das entsprechende Zeichen oder Zeichenfolge über die serielle Schnittstelle (funktioniert auch mit Zeichenliste Auswahl "List")

3.9 Hinweise zur Programmierung im erweiterten Programm-Modus

Es gibt folgende typische Fehlermöglichkeiten:

Anzeige (LCD)

Damit die LCD Anzeige nicht flimmert, sollte die Anweisung "\c" für Clear Screen (LCD löschen) bei zyklischer Anzeige nicht verwendet werden.

Stattdessen sollte die LCD gezielt beschrieben werden, z.B. mit "\1" für die Ausgabe in der ersten Zeile.

Die Anweisung "\n" löscht bis zum Zeilenende alle Zeichen und sorgt für einen korrekten Bildaufbau bei unterschiedlich langen Positionsausgaben.

Tastatur

Damit keine Tastbetätigungen verloren gehen, darf die Tastatur über den P1162 nur ein Mal pro Schleifendurchgang gelesen werden. Eine Tastbetätigung wird bis zur Abfrage mit P1162 gespeichert und nach der Abfrage gelöscht. Wenn also der P1162 das zweite Mal in einer Schleife abgefragt wird, ist der Tastenwert 0.

Für die Auswertung mit mehreren "IF" Vergleichen muss mit X=P1162 zwischengespeichert werden.

Zieladresse

Die Zieladresse "Z" muss bei der Steuerung von mehreren Antrieben und bei Zuweisungen und Abfragen von Parametern beim SERS-Programmer und SERS-Antrieben in einem Ablaufprogramm, richtig eingestellt werden.

Wenn z.B. die entsprechende Zuweisung Z=0 bei einer IF-Abfrage eines Parameters des SERS-Programmers vergessen wurde - z.B. P1184 zum Lesen des aktuellen Tastenwertes - (und kurz davor mit Z=1 ein Parameter eines SERS-Antriebs zugewiesen wurde), dann gibt es die Fehlermeldung "unbekanntes if Ereignis" (weil der Parameter im Zielantrieb nicht vorhanden ist - P1184 existiert in einem SERS-Slave nicht, da es sich hier um einen reinen SERS-Programmer Parameter handelt).

Zuweisungen

Die einzelnen Anweisungen werden intern in 16 Bit und 32 Bit Worte umgesetzt und gespeichert. Dadurch braucht ein Programm wenig Speicherplatz und läuft schnell ab. Eine Einschränkung besteht jedoch in der Anzahl verwendbarer Parameter pro Anweisung. In Zuweisungen kann jeweils nur ein Parameter und eine Konstante oder der Rechenakku "X" verwendet werden.

Möglich sind z.B.:

P51=360 P51=X X=P51 X=R0
R0=X X=360 R0=360

Nicht möglich sind z.B.:

P51=P138 R1=R0

Zwei Parameteradressen haben im internen 16 Bit Opcode keinen Platz

Hier muss der Rechenakku X verwendet werden:

X=P51 P138=X X=R0 R1=X

Rechenoperationen verwenden immer den Rechenakku "X" als Ergebnisspeicher und als ersten Operanden.

Z.B. $W = ADC * (P51 + P41)$

muss codiert werden als:

$X = P51 + P41 * ADC$

W=X

3.10 Programmbeispiele für erweiterten Programm-Modus

Beispiel 1 –Kabelablängmaschine

Anzeige 1. Zeile : **F1=EDIT F2=START**

Anzeige 2. Zeile: **LAENGE: xxx**

Anzeige 3. Zeile: **ANZAHL xxx**

Anzeige 4. Zeile: **ANZ:xxx POS:xxx**

Parameter (änderbare Werte):

Länge (LAENGE), Stückzahlzähler ("ANZAHL")

Anzeige (nur angezeigte Werte):

fertig bearbeitete Stücke ("ANZ"), aktuelle Motorposition ("POS")

F1=EDIT F2=START

LAENGE: 360

ANZAHL: 20

ANZ:15 POS:1800.

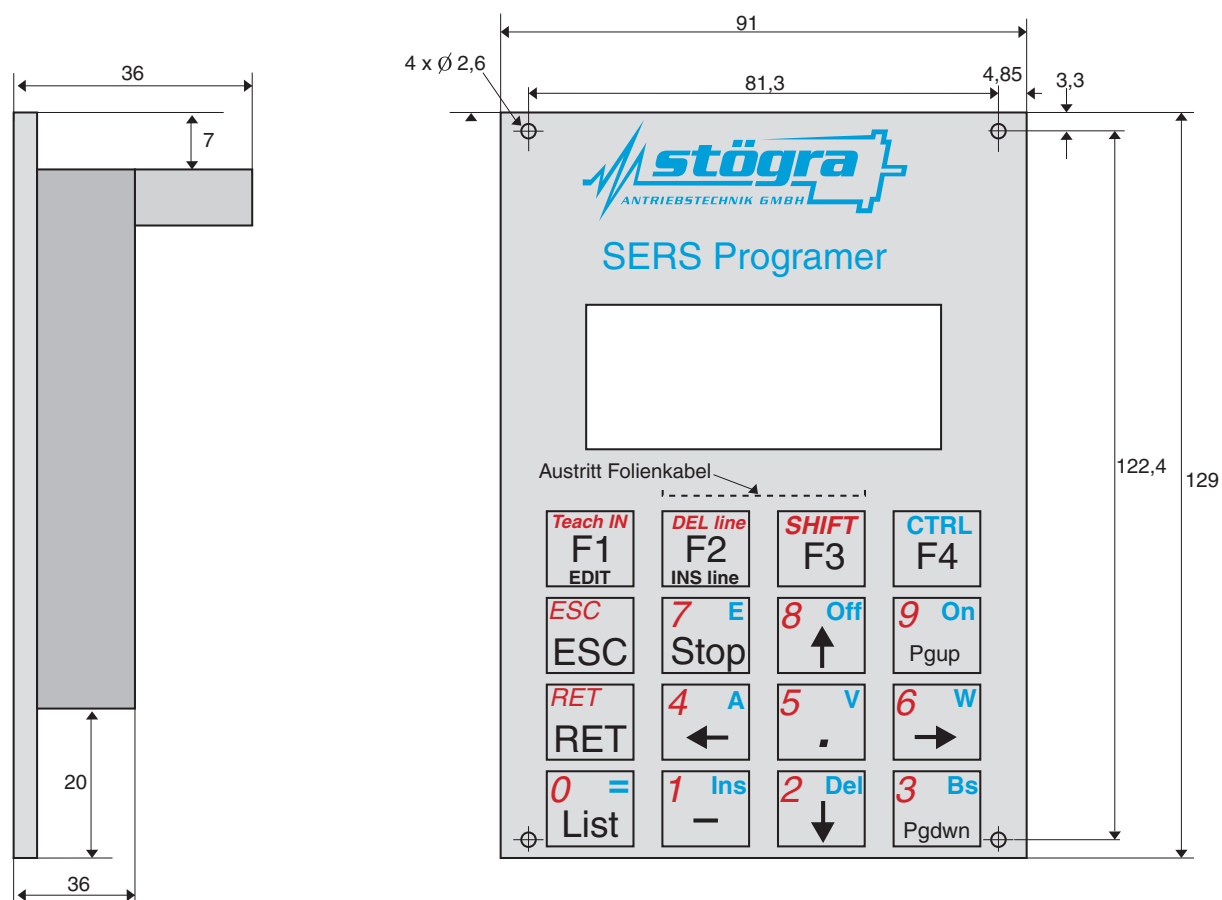
Ablaufprogramm (Erläuterungen nach "///", SP = SERS Programmer)

```

R2=0 R3=0 P51=0      // Initialisieren der verwendeten Parameter
L1                    // Programm-Marke – Sprungziel L1
GS20                  // Aufruf Unterprogramm bei Label L20 (hier Menüanzeige)
X=P1162 IF X<>241 GT 1 // Abfrage Tastatur auf Taste "F2"
X=R1 C1=X             // Zuweisung R1 (von SP) auf Zähler C1 (von SP)
X=R0 WR=X V=50       // Zuweisung von X (von SP) auf WR (von Achse 1)
R2=0 R3=0 P51=0 GS20 // Aufruf L20 (hier Menüanzeige)
L2                    // Zählerschleife über C1 und Abfrage C1 (siehe unten)
WAIT I1=1            // Warten auf Eingang I1 an Achse 1
O1=0 E                // Setzen Ausgang O1 (Achse 1) und Start Achse 1
GS 10                 // Aufruf Unterprogramm bei Label L10
IF C1>1 GT 2          // Ende Zählerschleife (zurück zu L2 falls C1>1)
GT 1                  // zurück zum Anfang (Sprung zu L1)
L10                   // Programm-Marke, hier Beginn Unterprogramm L10
GS20                  // Aufruf L20 (hier Menüanzeige)
IF POS=0 GT 10        // Solange Achse 1 noch positioniert, Sprung zurück zu
O1=1                  // Ausgang O1=1 bei Achse 1
X=R2+1 R2=X           // R2=R2+1 (für Anzeige bearbeitete Stücke)
GS20                  // Aufruf Unterprogramm bei Label L20
RT                    // Ende Unterprogramm L10 und Rücksprung
L20                   // Programm-Marke, hier Beginn Unterprogramm L20
"a"                   // Anfang Anzeigemenü (Anzeige SERS-Programmer)
"F1=EDIT F2=START"   // nur Text "F1=EDIT F2=START"
"LAENGE:#0R0!"        // "LAENGE" + R0 (von SP), R0 ist editierbar (wegen "!")
"ANZAHL:#0R1!"        // "ANZAHL" + R1 (von SP), R1 ist editierbar (wegen "!")
"ANZ:#0R2? POS:#1P51?" // "ANZ:" + C1 (von SP) + "POS:" + P51" (von Achse 1)
"e"                   // Ende Anzeigemenü
RT                    // Ende Unterprogramm L20 und Rücksprung

```

4. Abmaße



5. Technische Spezifikationen

Programmspeicher: 8KByte E²PROM

Versorgungsspannung über Pin 9 (9-pol. D-Sub): 8VDC (6 – 12 VDC, Ripple < 10%)
- die 8VDC werden von einer angeschlossenen SERS-Steuerung geliefert

Versorgungsspannung über 2-pol. Stecker : 24VDC (20 – 28VDC, Ripple < 10%)
- wird nur im Programmiermodus benötigt (bei Programmierung des SERS Programmer V2 über einen PC)

Stromaufnahme : 30mA

Schutzart : Ohne Gehäuse : IP00
Mit Gehäuse : IP41 (bzw. nur Frontseite IP65)