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 Programmer 2

Hand programming device for SERS-controls

Manual

Edition November 2011

Index

Index

1. General notes

1.1 SERS Programmer 2 and its version before SERS Programmer

The SERS Programmer 2, compared to its version before “SERS Programmer“ (and “SERS Programmer Version R“), additionally includes a programming mode, and an extended program mode. This enables programming customer specific display menus, and complete operational programs (for controlling SERS-controls) which are stored into the SERS Programmer 2.
This also allows using the SERS Programmer 2 as complete master in a system with one or multiple SERS-controls.

In the programming-mode an operational program is created for the extended program mode (via a connected Windows PC), and saved into the SERS Programmer 2. Furtheron parameter settings for the SERS Programmer 2 can be made in the programming mode.

They key pad with display is identical for both versions (new “SERS Programmer 2” and old version ”SERS Programmer”).

The visual difference between SERS Programmer 2 (and SERS Programmer 2-R) to the old version “SERS Programmer“ is the existing additional 2-pole connector (located beside the 9-pole D-Sub-connector), for supplying the SERSProgrammer 2 by 24VDC (only needed for the programming mode).

From beginning of chapter 1.3 within this manual the SERS Programmer 2 will be named only “SERS-Programmer“.

1.2 Standard mode and extended program mode

The SERS Programmer 2 can be operated in standard mode or in extended program mode.

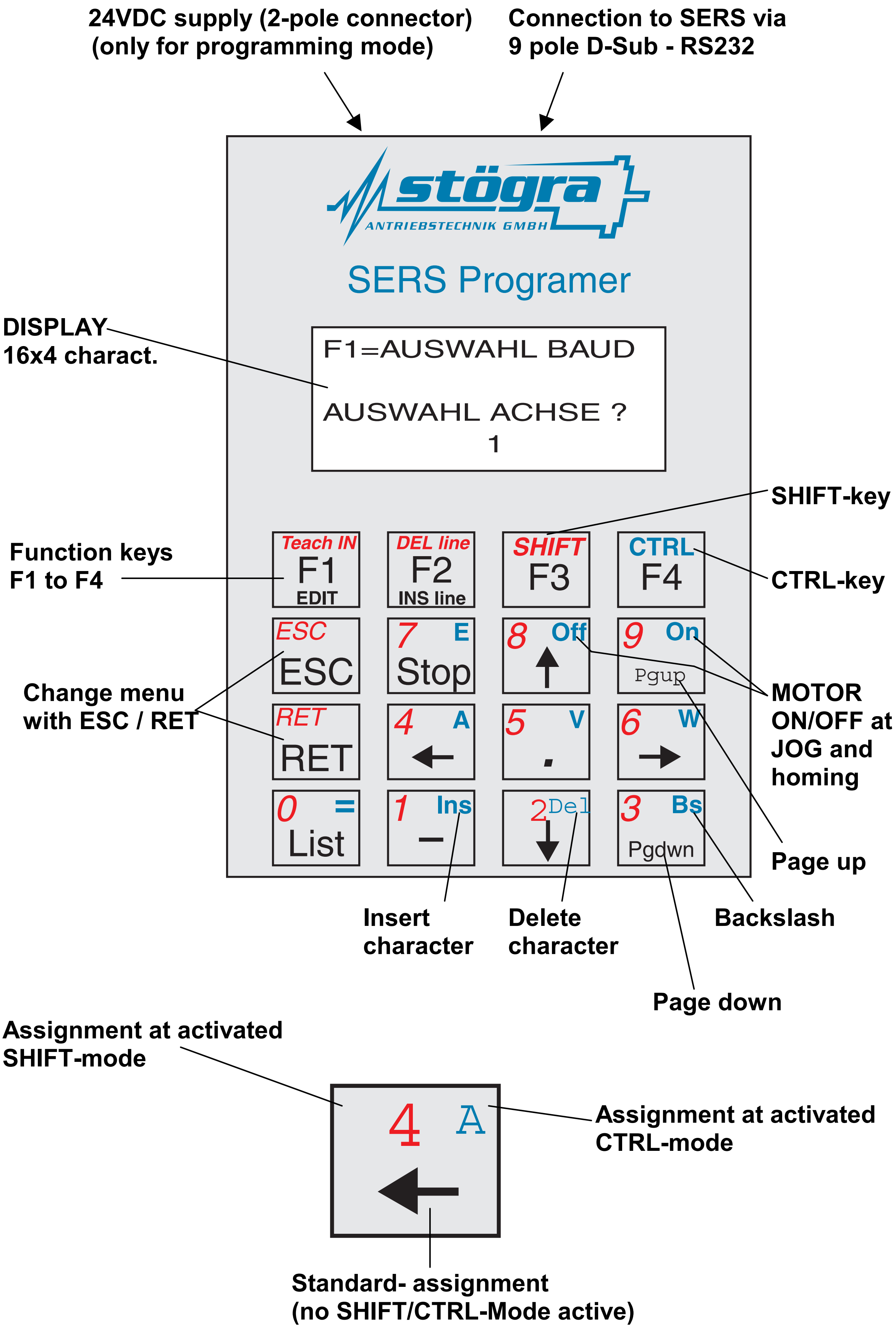
The standard mode includes a predefined menu (see chapter 2), and is identical to the “old” SERS-Programmer (version before SERS Programmer 2).
As long as there is no operational program saved into the SERS-Programmer 2 (as long as the SERS-Programmer 2 is not programmed in the programming mode for the extended program mode), at Power ON the SERS-Programmer 2 will be set automatically into the standard mode.

If an operational program exists in the SERS Programmer 2 (if it was programmed in the programming mode), then at Power ON the SERS Programmer 2 automatically will be operated with the extended program mode.

1.3 Keypad in extended program mode

In contrary to the standard mode the key assignment of the keypad can be defined free in the extended program mode.
The assignment of the keys is made in the operational program, respectively in the customer specific menu of the SERS-Programmer.

1.4 Keypad in standard mode



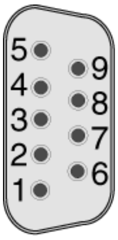
1.5 Connection SERS-Programmer to SERS-controls

SERS-Programmer and one SERS-control:

The Pins 2,3,5 and 9 of the 9-pole connector of the **SERS-Programmer** have to be connected **1:1** with the Pins 2,3,5 and 9 at the 9-pole connector of the **SERS-control** (**Attention: NO Zero-Modem connection**)

Configuration of pins of the 9-pole D-Sub-connector:

Pin
1 : not connected
2 : RXD
3 : TXD
4 : not connected
5 : GND

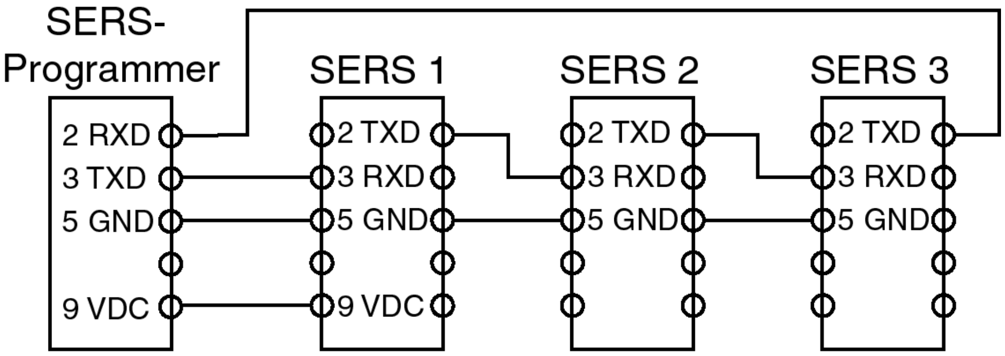


6 : not connected
7 : HSI (Hand Shake IN)
8 : HSO (Hand Shake Out)
9 : 9 VDC (supply SERS Programmer)

The handshake connnections (Pin 7 and 8) are connected inside the SERS-Programmer, but are not evaluated in standard version.

SERS-Programmer and multiple SERS-controls:

Connecting diagram:



ATTENTION:
The SERS-Programmer is supplied via Pin 9 at the 9-pole D-Sub-connector with 8 VDC by a SERS-control.
When connecting more than one SERS-controls, the SERS-Programmer should be supplied only by one SERS-control via Pin 9 !

1.6 Connecting the SERS-Programmer to a Windows-PC in the programming mode

In the programming mode (programming and saving an operational program for the extended program mode) the SERS-Programmer needs to be connected to a PC with a Microsoft Windows operating system (W95 to Vista).
Additionally to the connection to the RS232-interface of the PC (connection via 1:1 cable – see chapter 1.5), the SERS Programmer needs to be supplied externally by 24VDC (via the 2-pole connector).

1.7 Keyboard configurations

The keys of the SERS-Programmer are configured with one, two or three different characters/functions depending on the actual menu at the display !

At some menus only the keys F1, F2, F3, F4, ESC and RET with the standard configuration (black characters on the keys) are used.

The corresponding function for the function key is described in the actual display.
E.g.:

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			

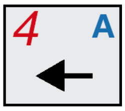
Pressing the ESC-key or RET-key will return to the menu before (higher level menu).

At some menus the keys include up to three different configurations.
The different configurations of the keys are marked with different colours (black, red and blue).

The standard configuration of a key is the black character / sign.

Pressing the red SHIFT-key (F3-key) once will enable the SHIFT-MODE and the RED characters / functions on the keys.
The enabled SHIFT-Mode is indicated via the character "▲" at the display of the SERS-Programmer in the last line, in the lines right end.
As long as this character is displayed, the red characters on the keys are enabled.

Example:



In the SHIFT-Mode the number "4" may be used during editing a program or when changing parameters in the parameter mode.

When pressing again the SHIFT-key, the SHIFT-Mode will be disabled and the character "▲" disappears at the display.

Pressing the blue CTRL-key (F4-key) once, will enable the CTRL-MODE and the BLUE characters / functions on the keys.
The enabled CTRL-Mode is indicated via the character "F" at the display of the SERS-Programmer in the right end of the last line.
As long as this character is displayed, the blue characters on the keys are enabled.

SHIFT-Mode and CTRL-Mode can be enabled only when changing parameters, when editing the operational program and in the Terminal-mode.

2. Operation and programming in standard mode
2.1 After Power ON

The SERS-Programmer starts operating immediately after connecting the supply voltage at the 9-pole D-Sub-connector.

After a 2 seconds initialisation phase the SERS-Programmer seeks for connected SERS-controls.
The baud rate and address, set at the SERS controls, will be recognised automatically (only drive addresses 1 to 3 are scanned automatically).

After finding a connected SERS-control following message will be displayed on the SERS programmer:

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							

In case of only one connected SERS the displayed drive address (corresponds to the set address at the DIP-switch on the SERS-control) has to be confirmed with the "RET"-key.

In case two or more SERS-controls are connected to the SERS-Programmer, via the arrow keys "↑" and "↓" there can be selected the address of SERS-control to be programmed.
All following operations with the SERS-Programmer are sent only to the SERS-control with the here selected drive address (exception function START ALL).

If later another SERS-control shall be programmed , first in the above menu the corresponding address has to be selected via the arrow keys.

How to return to the menu to select a drive address is described in the chapter "Address/language menu".

After confirming the selected drive address with the RET-key the display changes to the main menu.

ERROR MESSAGES:

If after Power On the display shows **"*** kein Echo"**, then check the connections (Pins 2 and 3 must be connected 1:1 - see chapter 1.2).

If after confirming a drive address with the key RET, the message **"*** kein Antrieb adressiert"** is displayed, then verify the addresses set at the SERS-controls via the address switches and the selected address at "AUSWAHL Adresse ?" ("SELECTED AXIS ?").

2.2 General notes to the menus

The menus of the SERS-Programmer are designed as tree structure (illustrated in chapter 3 "menu tree structure").

Pressing the function keys F1 to F4, described at the menus on the display leads to the submenus.

Pressing the "ESC"-key or the "RET"-key results in jumping back to the higher level menu.

2.3 Main menu

The Main menu can be entered via confirming the selected drive address with RET or ESC after Power On (see 2.1 "After Power On") or from all other menus via jumping back to the higher level menus with ESC or RET until reaching the main menu.

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				

F1 → Jump to MANUAL MODE-menu (see 2.7)
F2 → Jump to PROGRAM MODE-menu (see 2.14)
F3 → Jump to START Program (see 2.5)
F4 → Jump to START All (see 2.6)
ESC or RET → Jump to Address/language menu

2.4 Address/language menu

When pressing ESC or RET in the main menu the Address/language menu is displayed.

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 → Automatically scanning of connected SERS-controls
- Baud rate and address (1 to 3) are scanned automatically
- Menu on display described at "2.1 After Power On"

F2 → Jump into the menu, as described at "2.1 After Power On"
- Baud rate and address are not scanned automatically (manual setting of the address as at "2.1 After Power On" and there with F1 possibility to select a baud rate manually - via the arrow keys)

F3 → Selection of German language (for SERS-Programmer and SERS-control)

F4 → Selection of English language (for SERS-Programmer and SERS-control)

ESC or RET → Jump to Main menu

2.5 START Program

Pressing F3 in the main menu enters the "START Program" - mask.

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

Following SERS-diagnostics are displayed here:

- After "IN" in the 2. line on the display the inputs are shown (digital inputs I1 to I8 binary coded).
- The state of the outputs are shown at O1 to O4.
- In the 3. line below "IN" (in the 2. Line) are shown error / warning messages.
  - If the space is empty, then there is no warning or error at the moment.
  - warnings are indicated with "W" and the content of P12 (see manual SERS). e.g. "W2" indicates a temperature warning.
  - errors are indicated with "E" and the content of P11 (see manual SERS)
- In the 4th line the actual position is indicated
- An "-ok" after the position shows that the motor stands still

Pressing "F1" starts the actual SERS-Program (operational program in the EEProm of the SERS-control).
Pressing "F4" or the "STOP"-key interrupts or terminates (see description of parameter P1033 in manual SERS) the program.

Pressing "ESC" or "RET" returns to the main menu.

2.6 START All

Pressing F4 in the main menu enters the "START All" - Mask.

In case of two or more connected SERS-controls to the SERS-Programmer, with the "START ALL"-function all axis can be started simultaneously.

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

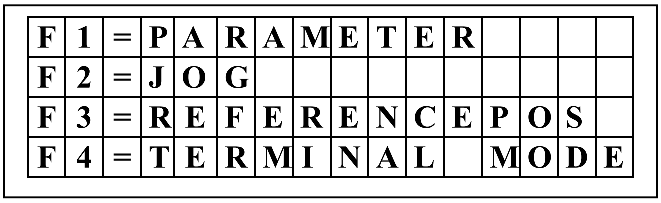
F1 → The command START-program will be sent to all connected SERS-controls ("RUN-command with broadcast address "*" : #*RUN)

F4 → The command STOP will be sent to all connected SERS-controls ("STOP-command with broadcast address "*" : #*S)

ESC or RET → Return to main menu

2.7 MANUAL Mode - Menu

Pressing F1 in the main menu enters the "MANUAL Mode" - Menu.



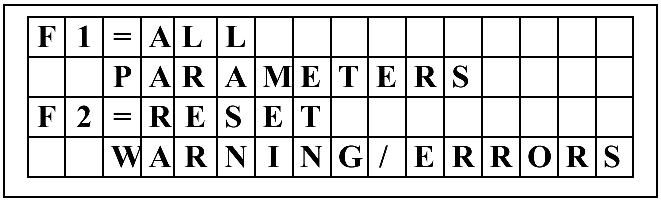
```
F1=PARAMETER
F2=JOG
F3=REFERENCEPOS
F4=TERMINAL MODE
```

F1 → Parameter-Menu (see 2.8)
F2 → JOG-Menu (see 2.11)
F3 → HOMING-Menu (see (2.12)
F4 → Terminal Mode (see 2.13)

ESC or RET → Return to main menu

2.8 PARAMETER - Menu

Pressing F1 in the MANUAL Mode - Menu enters the PARAMETER - menu.



```
F1=ALL
  PARAMETERS
F2=RESET
  WARNING/ERRORS
```

F1 → Parameter Mask (see 2.9)
F2 → RESET Menu (see 2.10)

ESC or RET → Return to main menu

2.9 PARAMETER - Mask

Pressing F1 in the PARAMETER - Menu enters the PARAMETER - Mask.
If no password was requested yet (e.g. when entered the PROGRAM Mode Menu) then now a password is requested.
The password is defined in the SERS-control.
Each connected SERS-control requires a password once after a Power-On-Reset, before the parameter and program section may be entered.

P	A	S	S	W	O	R	T	?							

The input of a 4-digit number is expected (switch assignments of keys to red letters/numbers - SHIFT Mode of keys).

After confirming the password with RET the PARAMETER-Mask is shown.

If the input password is correct, then all SERS-Parameters are shown.

If the input password is not correct, then only these parameters are shown, which are enabled in the SERS also without password (parameters P1060 to P1065 - see manual SERS).

NOTE: In case no password is defined in the SERS:
When asked for the password in the SERS-Programmer only the RET-key may be pressed (no input of a password) to enable the display of all parameters.

If by mistake a wrong password was input:
The SERS-control must be restarted (Power-ON-Reset required) to be able to input a password again.

PARAMETER Mask:

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

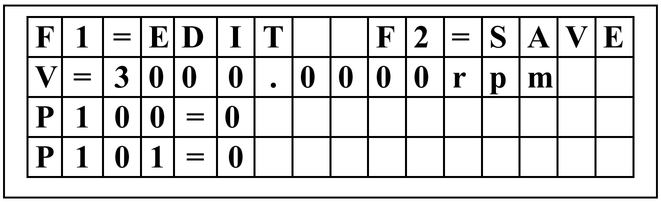
With help of the arrow keys "↑" and "↓" and the Page up/down keys "Pgup" and "Pgdwn" the lines can be scrolled and the cursor can be moved to the different parameters.

F1 → The parameter marked by the cursor (2. line in the display) can be changed.
ATTENTION: For the input of numbers first the SHIFT-mode with the red key assignment must be enabled (pressing one time the SHIFT-key).
If characters have to be deleted or inserted, then the CTRL-mode with the blue assignment must be activated (pressing one time the CTRL-key and the key "Ins" for inserting of digits and "Del" for deleting characters)

No scaling is allowed at the input.
(e.g. V=3000 is correct - V=3000 **rpm** is **wrong**)

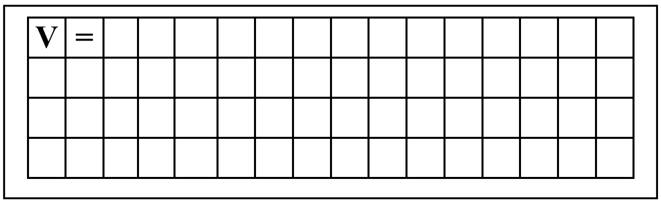
Pressing **"RET"** confirms the **changed value**, and **"ESC"** will **keep the old value**.

E.g.: Changing of the velocity - parameter "V" - The display must show the following lines:



```
F1=EDIT  F2=SAVE
V=3000.0000rpm
P100=0
P101=0
```

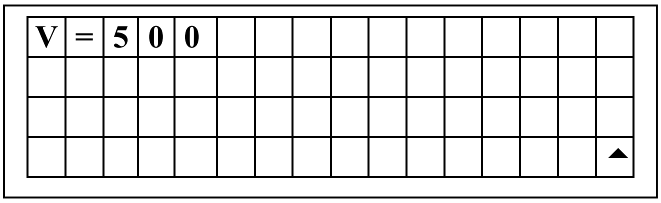
The velocity "V" shall be changed to 500 rpm.
For changing "V" the key F1 has to be entered. Then the display will be as follows:



```
V=
```

For being able to input numbers, the SHIFT-Mode must be enabled - press SHIFT-key . The enabled SHIFT-mode is indicated via the character "▲" in the display in line 4 at the right end of line

Now the number "500" may be input



```
V=500


               ▲
```

and confirmed with "RET".

<u>IMPORTANT NOTE:</u>
For saving the changes of the parameters permanently into the SERS EEPROM the key F2 (SAVE) has to be pressed. If the changes were not saved with F2, then after the next Power-ON-Reset of the SERS-control all changes will be lost..

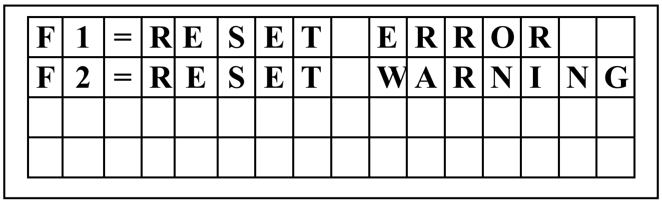
After pressing F2 the SERS-control returns an "ok" to the SERS-Programmer.
"ESC" or "RET" returns to the Parameter mask.

Another ESC or RET → Return to the PARAMETER Menu

ESC oder RET → Zurück zum PARAMETER Menü

2.10 RESET - Menu

When pressing the key Taste F2 in the PARAMETER - Menu the RESET-Menu will be displayed.



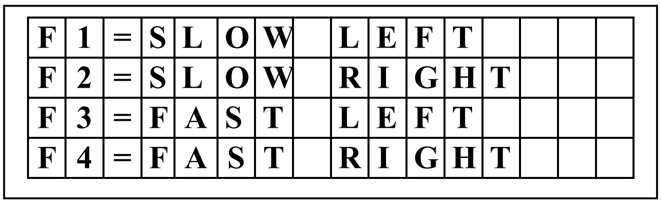
```
F1=RESET ERROR
F2=RESET WARNING
```

F1 → in the SERS actual error messages will be reset (P11=0)
F2 → in the SERS actual warning messages will be reset (P12=0)

ESC or RET → back to the PARAMETER Menu

2.11 JOG - Menu

When pressing the key F2 in the "MANUAL Mode" - Menu the JOG - Menu will be displayed.



```
F1=SLOW LEFT
F2=SLOW RIGHT
F3=FAST LEFT
F4=FAST RIGHT
```

Before the drive can be moved, the motor phase current must be ON. When the motor current is ON, the 7-Segment-display at the SERS-control shows a "5" (in case of no warning or error).
When pressing the "ON" key the motor current will be switched ON, when pressing the "OFF" key the motor current will be switched OFF. (**Note:** the CTRL-mode does not need to be active)

F1 → Drive runs slow left
F2 → Drive runs slow right
F3 → Drive runs fast left
F4 → Drive runs fast right

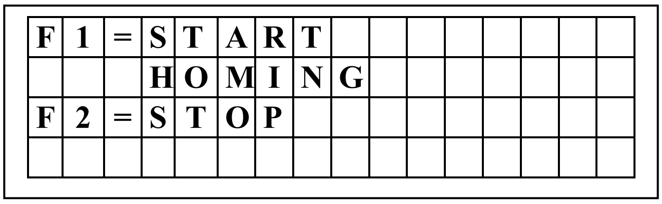
NOTE:
Whether the motor runs into the left direction when pressing F1 or F3, or if the motor in reality runs into the right direction, depends on the way of connection of the motor phases with the SERS-control (exchanging the both connections within one motor phase will change the running direction) !

The velocity and the acceleration of the running in the JOG mode can be adjusted via the parameters P1018 to P1020 (see manual SERS)

ESC or RET → Return to MANUAL Mode - Menu

2.12 HOMING - Menu

When pressing the key F3 in the "MANUAL Mode" - Menu, the HOMING-Menu will be displayed



```
F1=START
   HOMING
F2=STOP
```

Before starting homing the motor phase current must be ON.
(7-Segment-display at the SERS-control shows "4" → Motor current is OFF,
"5" → Motor current is ON)
Pressing the "ON" key will switch the motor current ON, pressing the "OFF" key will switch the motor current OFF. (**Note:** the CTRL-mode does not need to be enabled)

F1 → Homing of the drive will be started
(See also manual SERS - chapter homing)
F2 → the drive will be stopped

ESC or RET → Return to the MANUAL Mode - Menu

2.13 TERMINAL Mode

Pressing the key F4 in the "MANUAL Mode" - Menu enables the TERMINAL Mode.

Here all characters available at the keyboard (and the characters selectable via the function / key "LIST" - see chapter 2.18, page 20) may be sent direct to the SERS drive.

ATTENTION: Each new line sent to the SERS-control must begin with the character "#". Also the drive address must be sent once to the SERS for the SERS recognising, that the following characters are addressed to it.

E.g.: #1ON
The phase current of the drive with drive address "1" will be switched on.

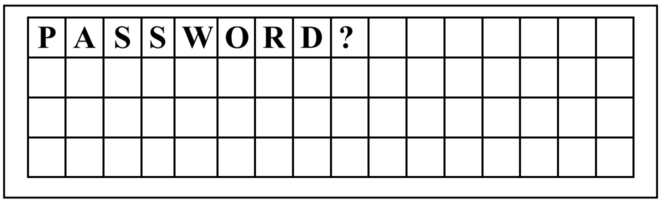
NOTE: The Terminal Mode can be entered only with valid password.

ESC or RET → Return to the MANUAL Mode – Menu

2.14 PROGRAM Mode - Menu

When pressing the key F2 in the main menu then the "PROGRAM Mode" - Menu will be displayed.

If there was no request for the password yet (e.g. before the PARAMETER mask) then the input of the password is required now.
The password is defined in the SERS-control.
Each connected SERS-control requires the input of the password once after a Power-On-Reset, before the parameters and the program section can be accessed.



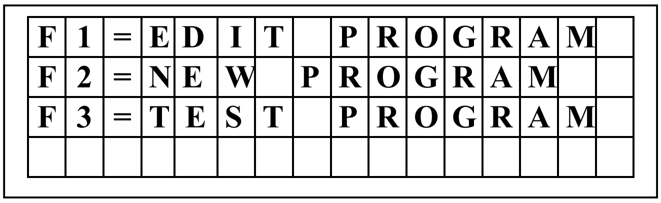
```
PASSWORD?
```

The input of a 4-digit number is expected here (press SHIFT-key to enable the red numbers on the keys).

NOTE: If there is no password defined in the SERS-control:
When asked for the password in the SERS-Programmer only the RET-key may be pressed (no input of a password) to enable the display of the total program.

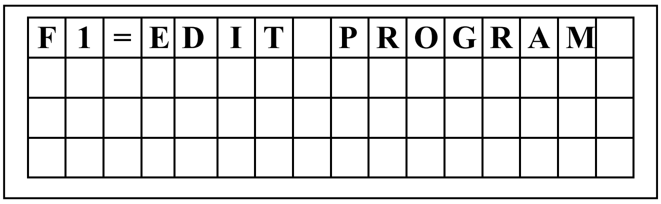
If by mistake a wrong password was input:
The SERS-control must be restarted (Power-ON-Reset required) to be able to input a password again.

In case of a correct password or if there is no password defined, then after pressing the RET-key following menu will be displayed:



```
F1=EDIT PROGRAM
F2=NEW PROGRAM
F3=TEST PROGRAM
```

In case of a wrong password or if there is a password defined and only the RET-key was pressed (without input of a password), then following menu will be displayed:



```
F1=EDIT PROGRAM
```

F1 → EDIT PROGRAM - mask (see 2.15)
F2 → NEW PROGRAM - mask (see 2.16)
F2 → TEST PROGRAM - mask (see 2.17)
ESC or RET → Return to main menu

2.15 EDIT PROGRAM - mask

When pressing the key F1 in the PROGRAM Mode Menu then the EDIT PROGRAM - Mask will be enabled.

IF a wrong password or no password (in case a password is defined in the SERS) was input before, then only the program lines are displayed, which are enabled in the parameters P1070 to P1074.

E.g.:

V	E	L	O	C	I	T	Y		A	X	I	S		1	
V	=	1	0	0	0	.	0	0	0	0					
D	I	A	M	E	T	E	R		H	O	L	E			
W	=	1	0	.	0	0	0	0							

All displayed value/parameter assignments may be changed now. To do that, the cursor must be positioned to the assignment to be changed by using the arrow keys and /or page up/page down keys.
After pressing F1 the assigned value can be changed
- by overwriting with a new value or
- by changing the single digits by using the arrow keys "↑" and "↓".

Pressing the RET-key will take the changed value into the program.
Pressing the ESC-key will keep the old value in the program.

In case the input password is correct the complete program will be displayed.

E.g.:

1	:	O	N												
2	:	L	1												
3	:	"	V	E	L	O	C	I	T	Y		A	X	I	S
	1		"												

By using the arrow keys "↑" , "↓", "→" and "←" and the page keys "Pgup" and "Pgdwn" all program lines can be reached.

Via the function keys F1 and F2 all together 4 functions can be called.

- **"EDIT"** → edit parameter assignments (e.g.: in program line 4:V=1000.0000 the value for "V" may be edited after pressing F1 - overwriting with new value or changing the single digits to higher or lower values by using the arrow keys "↑" and "↓" (**only parameter assignments may be edited with "EDIT" !**).
  Pressing the RET-key will take the changed value into the program and pressing the ESC-key will keep the old value.
- **"Teach IN"** (SHIFT-KEY and afterwards F2-key) → The display changes to the MANUAL Mode Menu and the drive can be moved by using the manual drive functions to the desired position. Pressing the RET-key this position will be taken into the program, by inserting automatically a new program line after the actually by the cursor indicated line : "W=' actual motor position'" (actual motor position is the position where the motor was moved with the manual drive functions)

NOTE to Teach IN: Pay attention to the correct positioning mode
(absolute or relative positioning - parameter P1014)
With ESC the function will be truncated and no line will be inserted.

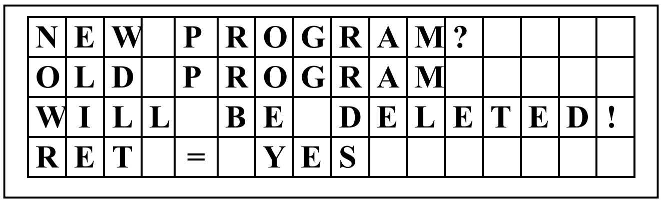
- **"INS line"** → after the by the cursor actual indicated program line there will be inserted an new line - any new command or assignment may be inserted into the program - all characters on the keys (black, blue via CTRL and red via SHIFT) and the additional characters via the "LIST"-key (see chapter 2.18 page 20) are available.
  The RET-key will take the new line into the program, the ESC-key will truncate the "Insert"-function.
  NOTE: For inserting a line before the first program line, the cursor must be positioned to the left side to the line numbering "1".
  If there are more than one commands or assignments in one program line, and a new program line is to be inserted between two commands in a line or before the next line, then the cursor must be positioned to the correct command within the line by using the arrow keys "→" and "←" , and after that the new command may be inserted **(Inserting a new line is done always after the command marked by the cursor !)**
- **"DEL line"** → The command actually marked by the cursor will be deleted. Before deleting the operator will be asked, if he really wants to delete the command (RET will delete the command, all other keys will truncate the "delete"-function)

NOTE: In contrary to the parameter changes, the program changes will be saved immediately permanently into the EEProm of the SERS-control (In case of parameter changes the saving of the new values into the EEProm will be done only after pressing the key F2 - see chapter 2.9)

ESC or RET → Return to PROGRAM Mode Menu

2.16 NEW PROGRAM - Mask

When Pressing the key F2 in the PROGRAM Mode Menu the
NEW PROGRAM - Mask will be displayed.



```
NEW PROGRAM?
OLD PROGRAM
WILL BE DELETED!
RET = YES
```

ATTENTION !
Pressing the RET-key will delete the actual program in the SERS !
After that a new program can be input.
All functions as described in chapter 2.15 are available (in case of correct password)

ESC or RET → Return to PROGRAM Mode Menu

2.17 TEST PROGRAM - Mask

Pressing the key F3 in the PROGRAM Mode Menu will enable the TEST PROGRAM - Mask.

Here the program can be executed step by step (command by command).

E.g.:

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. line : last executed command in the program
3. line : next to be executed command
4. line : actual drive position (P51) and a "-ok" in case the drive does not move.

F1 → Next command (shown in 3. line on display) will be executed.
F2 → Actual command and drive will be stopped.

ESC or RET → Return to PROGRAM Mode Menu

2.18 Additional characters in TERMINAL mode and EDIT Program / NEW Program

When editing a program and in the terminal modus via the key "**List**" there areavailable additional characters.

0 = List

With the arrow keys a character in the list can be selected and inserted into the program or the terminal with the key RET.

With the keys "Pgup" and "Pgdwn" three different pages with characters may be selected.

1. page:

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			|	
!		*		?		=		#		/		+		&	

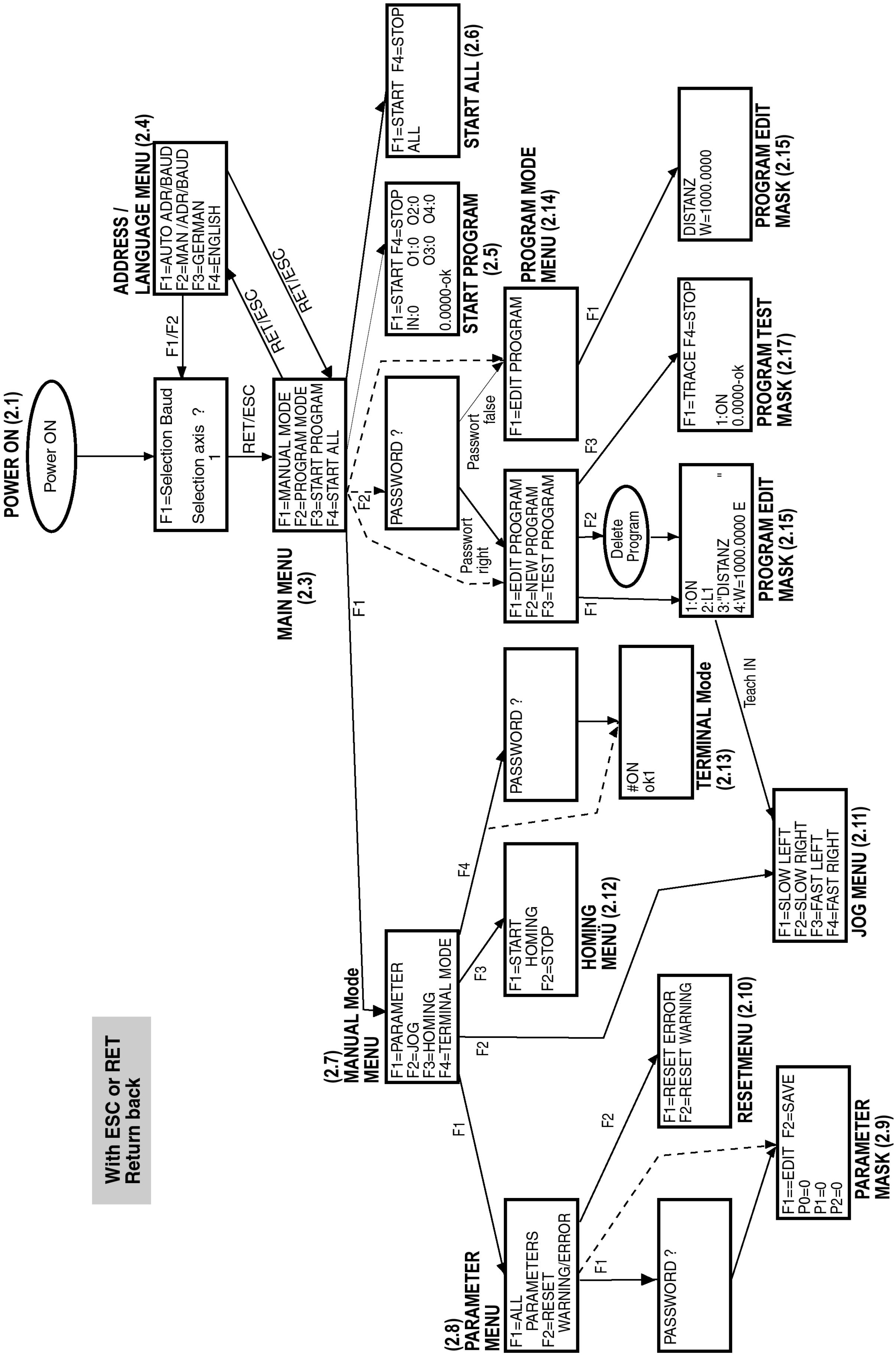
2. page:

"		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. page:

:		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 Menu - tree structure in standard mode



3. Operation and programming in the extended program mode
3.1 Programming mode

For saving a program for the extended program mode into the SERS-Programmer, or for setup of parameters, the SERS-Programmer must be set into the prgramming mode. Also it needs to be connected via a 1:1 connecting cable to the RS232-interface of a PC.
Furtheron the SERS-Programmer must supplied externally by 24VDC (via the 2-pole connector, which is located beside the 9-pole RS232 D-Sub connector).

Immedeately after (or already before) supplying the 24VDC voltage, the key “F1“ at the SERS-Programmer key pad must be pressed until the display shows:

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

Now the SERS Programmer is in the programming mode and the STÖGRA programming software may be started at the PC, for communicating with the SERS-Programmer. At the right upper corner of the PC software window there will be shown the found baudrate (19200 Baud) and the found address (200) of the connected SERS-programmer.
The set address of the SERS-Programmer always is “0“ (master address). But for programming it, it must behave like a slave. And that is the case if the SERS-programmer will be contacted by address “200“.

If the software window shows “no drive“, close the software, and verify the connection (correct 1:1 cable and SERS-Programmer is ready with above display) and restart the software.
If the problem continues (“no drive“), then try, if available, another PC.

The STÖGRA programming software allows to setup an operational program with customer specific display, and store it into the SERS-Programmer.
The SERS Programmer includes 8Kbyte program memory (E^{2}Prom), what corresponds to aproximately 1200 instructions.

The Button “program“ in the PC programming software opens the program window, where an operational program can be programed.
The Button “save program into SERS“ will save the operational program into the SERS-Programmer.
The syntax for an operational program and the (display) menu is described in chapter 3.3.

By pressing the button “parameter“ in the PC programming software, the actual parameter settings of the SERS programmer will be shown.
The parameters can be modified according to your needs.
Button “save parameter into E^{2}Prom“ will save the parameter modifications permanently into the E^{2}Prom of the SERS-Programmer.
Explanations to the relevtant parameters can be found in chapter 3.6 (pages 38 - 41).

After saving an operational program into the SERS-Programmer, and after a restart (power OFF/ON, or switch off 24VDC and connect the SERS-programmer to a SERS control), the SERS programmer will be started automatically in the extended program mode, with the new customer specific (display) menu.

3.2 Operation in extended program mode

In case of an existing operational program in the SERS-Programmer (a program was created in the programming mode), then the SERS-Programmer starts automatically in the extended program mode, after a Power-ON-Reset.
This means also starting the operational program automatically.

Before starting the operational program an automatical address scan is executed (the SERS programmer is scanning connected SERS/WSERS drives with addresses 1 to 9).
For each exisiting SERS/WSERS drive a corresponding bit is set in parameter P1188 (see also description of P1188).
The lowest address found will be written in parameter P1185.

In case P1186=0 or P1186>7, also the baudrate of the connected SERS/WSERS drives is tested automatically (in case of multiple drives connected, the baudrate must be identical for all drives).
Otherwise (0 < P1186 < 7) the baudrate defined in P1186 will be used.

The address of the SERS-Programmer always is “0“ (only in the programming mode the SERS programmer will be called by address “200“).

3.3 Syntax extended program mode
3.3.1 Syntax programming display menu

Programming of the display menu is done by control characters set between quotation marks “ “ .
That means the quotation marks “ “ need to be programmed explicitly.
Below control characters are defined:

"\a": beginning of the menu, for menus with up to max. 4 lines

"\b": beginning of the menu, for menus with more than 4 lines (scroll menus)

"\e": end of the menu (here the key pad is evaluated)

"\c": erase LCD display and start character output in the left upper corner

"\n" or "\r": begin new line (line break - CR/LF)
a line break also will be created automatically at the end of each line, if the line was not terminated by character “\%“
In case of "\n" or "\r" also all characters, coming after "\n" or "\r" until the end of the line, will be erased

"\%": do not begin a new line (no automatical line break), but output of the next characters directly behind the last character defined

"\1": character output in the 1st line

"\2": character output in the 2nd line

"\3": character output in the 3rd line

"\4": character output in the 4th line

"#x": “x“ may contain the letter “a“ or a number (SERS-slave address)
“#a“ (x = “a“): the parameter following “#a“ refers to the actual SERS-address (SERS address set in P1185 – see P1185 at page 40)
“#1“ (x = “1“): the parameter following “#a“ refers to the SERS with address “1“
all characters following "#x", until "?" or "!", are parameters which are only displayed, or which are displayed and can be edited (depending on which character follows - "?" or "!" – see definitions below).

"?": write parameter (see above “#x“) in LCD-display
parameter (value) will be only displayed

"!" : write parameter (see above “#x“) in LCD-display
parameter (value) will be displayed and can be edited via key "F1"

“text“: *text* may be any characters or any expression, with exception of above defined control characters

ATTENTION:
In case the display menu contains parameters, which can be edited (with control character “!“), then the menu must be called repetitive (e.g. by using a program loop). Only when calling the corresponding line, it will be checked wether the key “F1“ was pressed. And only then the corresponding parameter can be edited.

3.3.2 Syntax programming operational programs

in this chapter the general syntax definitions for an operational program will be described. The syntax is identically with the syntax for SERS controls (operational programs stored into SERS controls). All expressions and characters, which are not described more detailed, must be written as they are.
Cursive written expressions are described more detailed in the following lines.
Expressions set in **[ ]** are optionally.
All key words may be written in small or big letters.
Explanations for the syntax are behind the comment characters ***//*** .

instruction line

# [*drive address*] [*instruction list*] *end of line*

drive address

decimal constant // allowed is 0 - 127
* // addressing all drives – e.g. for synchronically starting all drives - broadcast address

When using the STÖGRA programming software then “ **# [*Achsnummer*]** “ will be admended autoamtically at each line. Therefore for setting up operational programs with the STÖGRA programming software, use only following expression:
[*instruction list*] *end of line*

instruction list

instruction
instruction [instruction list]

instruction

"character list" // Text for displaying in the SERS-Programmer
assignment
command
operator operand // for arithmetic functions
unary_operator // for arithmetic functions
IF [!] *address* [*condition decimal constant*] // conditional execution - see chapter 3.5.1
WAIT [!] *address* [*condition decimal constant*] // Wait for event
: [!] *address* [*condition decimal constant*] // manual driving until input is (not) active
GOTO *decimal constant* // jump to label number decimal constant
GOSUB *decimal constant* // jump to subroutine at label number decimal const.
GT *decimal constant* // equivalent to GOTO
GS *decimal constant* // equivalent to GOSUB
RETURN, RT // return from subroutine – target address is the line after the last GOSUB command
RS // manual drive right slow (**R**ight **S**low)
RF // manual drive right fast (**R**ight **F**ast)
LS // manual drive left slow (**L**eft **S**low)
LF // manual drive left fast (**L**eft **F**ast)
L *decimal constant* // label number
PE // end of program
PSAVE // parameters in the SERS will be saved permanently – save values from RAM of SERS into E^{2}Prom of SERS

POSSAVE	// save P51 (actual-position) into E^{2}Prom
POS0	// move to electrical “0“-position (every 7,2°)
POSR	// move to position command value (see P1043)
RUN	// equivalent to P0=1 – start program in E^{2}Prom
VER	// shows program version
ON	// equivalent to P134=7 – turn on phase current of motor
OFF	// equivalent to P134=0 – turn off phase current of motor
//	// comment character – all characters in a line from here will be not interpreted

condition

< <= = > >= <>

character list

character
character-list character

character

// any character except quotation marks “ and 0x00 (ASCII-character with code 00)

assignment

address = *data*	
X = operand	// accumulator for calculating
address ?	// shows parameter data (value - content)
address ??	// shows parameter identification

operator

+	// adding to the accumulator
–	// subtracting from the accumulator
*	// multiplying with accumulator, in case ‘∗‘ shall be used at begin of line, then the address of the drive must be set ahead, for the SERS not interpreting the character ‘∗‘ as broadcast address
/	// dividing accumulator
&	// “AND“ accumulator
|	// “OR“ accumulator
^	// “EXCLUSIV OR“ accumulator

unary operator

NOT	// inverting accumulator bit by bit
NEG	// inverting sign of accumulator

operand

address
decimal constant

address

A	// equivalent P138 – acceleration
B	// equivalent P1096 – deceleration at polynom positioning
ADC	// equivalent P1046 – analogue Input
C1, C2, C3	// equivalent P100, P101, P102 - counter
D	// equivalent P1100 – delay time in $^1/_{10}$ seconds
D2	// equivalent P1164 – delay time in $^1/_{10}$ seconds
DB	// equivalent P1166 – delay time bit
M1bis M20	// marker M1 to M20
I1 to I8	// inputs I1 to I8 from P1300

IN	// equivalent P1300 (for syntax with IF and WAIT)
I9 to I16	// equivalent P1301 – inputs I9 to I16
J1	// home switch for requests with IF, WAIT and ":"
J2	// limit swith left for requests with IF, WAIT and ":"
J3	// limit swith right for requests with IF, WAIT and ":"
J4	// STOP-switch for requests with IF, WAIT and ":"
J5	// service switch for requests with IF, WAIT qnd ":"
J6	// zero point, at encoder input and encoder with zero point
O1 until O16	// equivalent P1201 until P1216 / output 01 until O16
P *decimal constant*	// parameter – see parameter description in SERS/WSERS-manual, and here in chapter 3.4
POS	// equivalent P336 – status In-Position
R0 bis R99	// register for free usage
V	// equivalent P91 - velocity
W	// equivalent P47 – distance to move or position
WA	// as W but additionally P1014=2 (absolute positioning)
WR	// as W but additionally P1014=0 (relative positioning)
WP	// first positioning section in mode polynom positioning
WAP	// as WP but additionally P1014=2 (absolute positioning)
WRP	// as WP but additionally P1014=0 (relative positioning)
WPT	// last positioning section in mode polynom positioning
WAPT	// as WPT but additionally P1014=2 (absolute positioning)
WRPT	// as WPT but additionally P1014=0 (relative positioning)
X	// equivalent P1047 - accumulator for calculating
Z	// equivalent P1 - destination address for master mode

<u>command</u>

E	// start actual positioning job – actual value set in W (P47) will be executed
H	// start homing
S	// stop – the motor will be stopped - see also parameter P1033 – continue after stop

<u>data</u>

[-] *decimal constant*

<u>end of line</u>

Carriage Return	// RETURN-key of keyboard
/r	// Return character – ASCII-Code 13 (decimal)
/n	// Linefeed character – ASCII-Code 10 (decimal)

<u>decimal constant</u>

decimal digit	
. decimal digit	// e.g. **. 5** (= 0.5)
decimal constant decimal digit	// e.g. **10.75**

<u>decimal digit</u>

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

<u>Separator</u>

"blank character"	
,	// comma
;	// semicolon, prevents a carriage return at program input in the actual line
"tab"	// tabulator character

3.3.3 Destination address at control characters, parameters and requests

Control characters for programming the display automatically always refer to the SERS Programmer (address “0“).
Instructions, commands, parameter and requests, in an operational program, need to be assigned excplicitely to a SERS drive or the SERS programmer, by “**Z=...**“.

Example:
Z=0 means commands are dedicated for the SERS-Programmer
Z=2 means commands are dedicated for the SERS drive with address 2

But there are some instructions / commands / parameter, which would not make sense in case of assigning to the SERS-Programmer (e.g. start positioning or start homing – these commands only make sense for a SERS drive).

Following instructions / commands / parameter therefore will be always assigned automatically to the SERS drive address, which is defined in P1185 (the lowest drive address found at the automatical address scan, after Power-ON). That means here a command “Z=...“ is not necessary:

"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"
Homing velocity: P41, Homing acceleration: P42
Actual position 1: P51
Feedrate override: P108
Homing parameter: P147, Homing velocity slow: P1003
Commutating table: P1009
Drives status and limit switches: P1013
Positioning mode: P1014
Acceleration phase: P1015, Constant phase: P1016
Jog acceleration: P1018
Jog velocity slow : P1019, Jog velocity fast: P1020
Ramp shape: P1032
Limit switches and Digin: P1056
Load next polynom section: P1123
Digin: P1300, P1301

3.4 Limitations for E^{2}Prom programs

Limitations for a E^{2}Prom program are as follows :

- program storage: the number of possible program lines depends on the way of programming - 8K Byte are available, as longer as the single lines are, as less lines may be programmed. When using only one instruction per line, then approx. 1200 Zeilen may be programmed.
- maximum 60 characters per program line
- maximum 128 labels – L1 until L128 - (with “special“ label L65, which is the destination address after a program interruption)
- interlacing levels for subroutines : max. 4 (interlaced program loops)

Syntax examples:

Example 1:
“\a“
“#0R2?“
“\e“

Register value R2 of axis 0 (SERS-Programmer) will be displayed.
e.g. if R2 = 100
SERS-Programmer display:

100

Example 2:
“\a“
”PROGRAM EXAMPLE ”
“REG2: #0R2!“
“\e“

In the first line of the SERS-Programmer the text “PROGRAM EXAMPLE “ will be displayed.
In the next line the text “REG2 “ will be displayed and behind that (still in the same line) the value of register R2 (of axis 0 - SERS-Programmer).
R2 may be edited (because of the character “!“ in the program).
E.g. if R2 = 100
SERS-Programmer display:

PROGRAM EXAMPLE
REG2: 100

Example 3:
“\a“
”PROGRAM EXAMPLE ”
“REG2: #0R2!“
“POS: #1P51?“
“\e“

In the first line of the SERS-Programmer the text “PROGRAM EXAMPLE “ will be displayed.
In the next line there will be the text “REG2 “ and behind that (still in the in same line) the value of register R2 (of axis 0 - SERS-Programmer).
R2 may be edited (because of the character “!“ in the program).
In the third line of the SERS-Programmer display the text “POS: “ and behind that the motor position of axis 1 will be displayed.
E.g. if R2 = 100, and actual motor position of axis 1 = 360.5000
SERS-Programmer display:

PROGRAM EXAMPLE
REG2: 100
POS: 360.5000

3.5 Functions for programming operational programs

3.5.1 IF : conditional execution

Syntax : **IF** *expression*

or **IF** *expression*
THEN ... command or multiple commands ...
[**ELSE]** ... command or multiple commands ...
END

expression: *parameter* condition *parameter* | *fixed point constant*
example: P11<P12 X>=V M1=O1 X<=-123.456 V>3000.0 O1<>O2

parameter: all SERS parameter and digital inputs (I1 to I16, and J1 to J6) are allowed. Also Parameter numbers and their alternative syntax may be used (e.g. “V“ or “P91“, which both mean the same parameter).

condition: < <= = <> > >=

fixed point constant: - 2147483.639 to +2147483.639

IF-structure without “THEN ...“:
If the “**expression**“ is true, then the first command after “expression” will be executed
If the “**expression**“ is false, then the first command after “expression” will be ignored.
All commands coming afterwards, will be executed independent of the IF-command.

IF-structure with “THEN ...“:
If the “**expression**“ is true, then all commands between “THEN“ and ELSE will be executed. In case there is no “ELSE“ (structure only IF ... THEN ... END), then all commands between “THEN“ and END will be executed.

If “ELSE“ exists, and “**expression**“ is false, then all commands between “ELSE“ and “END“ will be executed.

Following conventions are made for the **digital inputs**:

- **I1 to I8** : if the input is set, then the event is true.
  e.g.: IF I5=1 If input I5 is set, then execute next instruction,
  otherwise skip / ignore next instruction
  IF I5=0 If input I5 is not set, then execute next instruction,
  otherwise skip / ignore next instruction
- **IN0 to IN255** : IF IN=*digit* requests all inputs, where the values of the inputs are binary coded → (I1=1, I2=2, I3=4, I4=8, I5=16, I6=32, I7=64, I8=128)
  e.g.: IF IN5 → If I1 and I3 are set then execute next instruction
  (I1 + I3 = 1 + 4 = 5)
  For negative events use “!” → e.g. IF **!**IN5 → If not IN5 (I1=0 or I3=0)
- **J1** - Home switch (IF J1=1 → if the drive is on the home switch - opener)
- **J2** - Limit switch left (IF J2=0 → if the drive is on the limit switch left - closer)
- **J3** - Limit switch right
- **J4 -** STOP switch
- **J5** - Service switch external
- **J6** - Zero point (for encoders with zero point)

IF-request with counters C1, C2 and C3 (P100, P101 and P102):

Every IF-request will decrement the counter by 1
IF C1>1 GT 20 → C1=C1-1 and in case C1>1 execute the next command (GT 20)
By using this structure counter loops can be realised (see below example 3)

Typical requests of SERS-drives:

IF POS=0 ... Verify if the drive is still moving (e.g. after positioning of a SERS-drive was started by command "E")
POS=0 → drive is still moving, POS=1 → drive is stopped

IF RUN ... Verify if a program (started by command "RUN") is still running
RUN → Program is still running

Examples for IF-requests:

Example 1:

```
IF P12<>0 O1=1
GT 20
```

In case P12 <> 0 (a warning is active) then first output O1=1 will be set (O1=1 is the next command after the IF-request), and then a jump to Label L20 follows (GT20). In case P12 = 0 ("P12<>0" is wrong), then the next command O1=1 will not be executed, but skipped, and only the command after that (GT 20) will be executed.

Following program is identical with above program:

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

Please note: It is not important, if the command following the "IF-request" is located in the same program line (as the "IF-request"), or in the next program line.

Example 2:

```
IF P12<>0 THEN
O1=1 O2=0 O3=1
ELSE
O1=0 O2=1 O3=0
END
GT 20
```

In case P12 <> 0, the comands (coming after THEN) "O1=1", "O2=0" and "O3=1" will be executed. In case P12 = 0, the commands (coming after ELSE) O1=0 O2=1 O3=0 will be executed.

Example 3:

```
C1=10
L1
WAIT I1=1
WR=180 E
IF C1>1 GT1
```

The block with comands "WAIT I1=1" and "WR=180 E" will be executed 10 times.

Example 4:

```
IF ADC>R0 THEN
X=ADC*1000 V=X WR=3600 E
END
```

In case the value at the analogue input ADC is higher than R0, then the commands "X=ADC*1000 V=X WR=3600 E" are executed.

PLEASE NOTE:
A maximum of 64 structures with THEN .. ELSE .. END are allowed in an operational program!

3.5.2 Label : program lables

Syntax : L *decimal constant*
decimal constant is a value between 1 and 128 (e.g. L1 or L28)

Labels are used as jump destinations, for creating program loops or subroutines. Each label number may be defined only once in a program !

In case of P1033=2: The program jumps to Label L65 if the E^2PROM-program is terminated suddenly, e.g. by an external Stop-command or a drive error (e.g. error temperature). This enables certain actions to be executed in case of an error.
In case Label L65 does not exist, then the program will be terminated after a drive error or a ‘stop‘.

3.5.3 GOTO, GT : jumps

Syntax : GOTO *Label* - Label must be defined anywhere in the program
Alternative syntax : GT *Label*

Program jump to a label (backwards or forwards)

```
        L1
        ...
        GOTO 1              // equivalent :  GT 1
or
        GOTO 1
        ...
        L1
```

3.5.4 GOSUB : call of subroutine

Syntax : GOSUB *Label* - Label must be defined anywhere in the program
Alternative Syntax : GS *Label*

Call (jump to) a subroutine
– Each subroutine contains a label at its begin and a ‘RETURN‘ at its end
e.g.:

```
        ...
        GOSUB 12            // equivalent :  GS 1
        ...
        L12                 // start subroutine
        ...
        RETURN              // end of subroutine and continue with instruction after last
                               'GOSUB'
```

3.5.5 RETURN : terminating subroutine

Syntax : RETURN
Alternative Syntax : RT

The instruction RETURN terminates a subroutine, which was called with ‘GOSUB‘ and initiates a jump to the instruction, which follows the ‘GOSUB‘.

A RETURN in the parallel mode terminates the called subroutine (called via a signal at the start input and a address at the I/O-Port). Then the SERS waits for the next subroutine call.

3.5.6 Programming positioning jobs

A positioning job is specified by the parameters **acceleration** (syntax : **A=*value***), **velocity** (syntax : **V=*value***), and **travel distance/position** (syntax : **W=*value***).
The command **'execute positioning job'** (syntax : **E**) starts the positioning job.

For executing a positioning job with the SERS, only the execute command '**E**' has to be sent to the SERS or '**E**' must be programmed in the executable program in the E^2Prom.
The parameters A, V and W of an executed positioning job are kept in the memory of the SERS, and, if not overwritten, they can be used for the next positioning job.

After Power-On of the SERS the parameters A and V, stored in the E^2PROM, are valid. The parameter W after Power-ON always is "0" (except P1117=1 and W was saved, as all other parameters, into the E^2PROM by writing PSAVE, or W was saved into the E^2PROM by using command 'POSSAVE')

The parameter A and V may be redefined any time. If these parameters are overwritten, during the SERS is executing a positioning job, then they will be valid only for the next positioning job.

Changing W is only possible at standstill of the drive (POS=1), or when the drive runs with constant velocity (constant phase) .
When overwriting W in the constant phase, then in relative positioning mode the new value W will be added to the previous value W and the drive will execute the total new distance W. In absolute positioning mode the drive will position to the new value W. The value W sent during the constant phase may not result in a change of the motor direction. In that case the sent value W will be returned with an error message, and only after finishing the actual positioning job, W will be executed with the next instruction 'E' (start positioning).

Additionally to the standard parameters there are some more parameters, which influence the positioning jobs :

- **Positioning mode** P1014 (**relative or absolute positioning**)
- **Ramp form of the acceleration** : P1032 (exponential or sinus-ramp form)
  P1005 (acceleration section 1)
  P1006 (acceleration section 2)
  P1007 (velocity section 1)
  P1008 (velocity section 2)
- **Scaling** : P160 (acceleration)
  P44 (velocity)
  P76 (travel distance / position data)

Alternatively to the assignment W=*value* there are the commands:
WR=*value* : positioning mode is set to relative (P1014=0) **and** W=*value* (e.g. WR=1000)
WA=*value* : positioning mode is set to absolute (P1014=2) **and** W=*value*

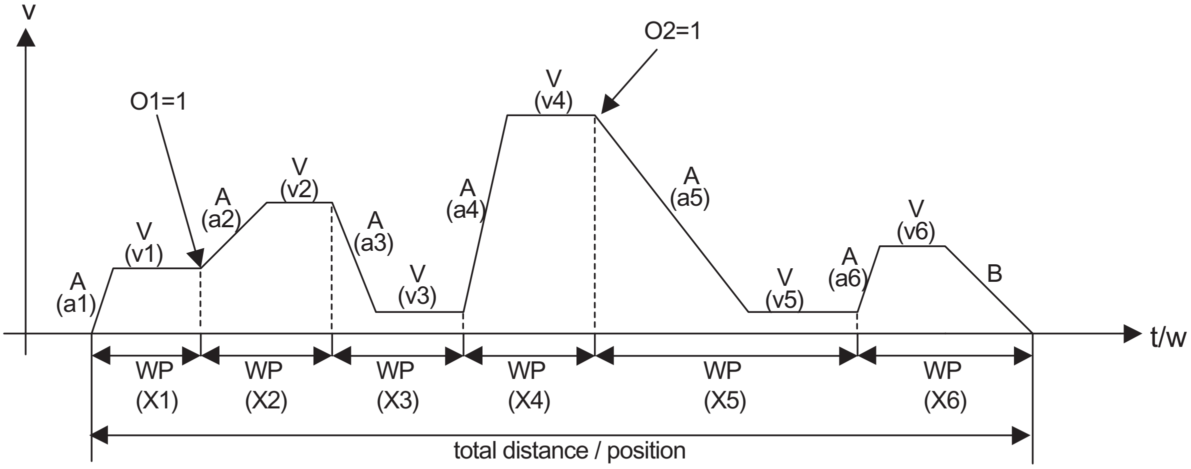
Following parameters show the actual status of a positioning job:

- **P336** (alternative syntax **'POS'**) –
  POS = 1→ actual position value = position command value
  (motor reached its destination, motor is not moving)
- **P1015 (acceleration phase)** = 1 during accelerating of the drive
- **P1016 (constant phase)** = 1 when the drive runs with constant speed

In the executable program mode the program only continues with the next instruction if POS=1 is fullfilled. But with parameter P1110 also immediate execution of the next instruction independently of 'POS' can be selected (P1110=0, usable e.g. for the master mode).

3.5.7 Positioning with velocity profiles (polynom positioning)

The drive shall move the distance W = X1 + X2 + X3 + X4 + X5 + X6.
First the drive must accelerate with the value a1 to the velocity v1 and then move with constant velocity v1 until the section W1 is completed. Then the drive accelerates with a2 to velocity v2 in the distance section X2. The sections X3 until X5 are as sections X1 and X2.
The last section X6 is defined with the acceleration a6, the velocity v6 and the decelerating ramp B.



Function principle:
A velocity profile consists of a single or multiple distance sections. Each distance section consists of an accelerating ramp and a section with constant velocity. The last distance section includes additionally the decelerating ramp B. When positioning with velocity profiles a distance section will be defined with the command “WP“. The acceleration A and velocity V may be redefined for each distance section or the actual values set may be used (no redefini-tion of acceleration and/or velocity value in the new distance section).
Within the definition of a distance section first the command “WP“ must be defined.
The last distance section must be defined with the command “WPT“.
Each distance section must be terminated with “E“.

In case SERS-outputs, marker or any parameters shall be set/defined after terminating a distance section within a velocity profile, (e.g. in the diagram above the outputs O1 and O2), proceed as follows: at first the next coming distance section has to be defined (during still executing the actual = last defined and started distance section), and after that the parameter, output or/and marker to be set after the still running distance section, may be defined.

Terminating the polynom mode without using “WPT” must be done with by using “POS0“ or “POSR“ - see parameter P1043. After an error (e.g. because of a wrong definition of polynom sections) the polynom mode stays active, until it will be terminated by a command.

The example / diagram above with 6 distance sections must be programmed as follows:

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

The output O1 will be set after terminating the first distance section (but note: the command O1=1 comes only after the definition of the second distance section !). The output O2 will be set after terminating the fourth distance section (O2=1 is defined after the definition of the fifth distance section !).

The definition of the acceleration within a distance section is optionally. If the acceleration is not defined within the single distance sections, then the last set acceleration
A is used (with exception of the last decelerating ramp → here parameter B is used).

The command “WP” depends on the positioning mode (P1014 – relative or absolute positioning mode) in the same way as the standard positioning command (positioning/distance definition) “W“.

Alternatively to WPT it can be used:
WPA : distance section absolut (e.g. WPA=1000)
WPR : distance section relativ (e.g. WPR=500)

and alternatively to WPT it can be used:
WPTA : terminating distance section absolute (e.g. WPTA=1000)
WPTR : terminating distance section relative (e.g. WPTR=1000)

If the single polynom distance sections are sent via RS232 to the SERS drives, then the SERS indicates via parameter P1123 (P1123=1), when the next polynom section must be sent. The SERS needs to receive the next polynom section definition, before the actual polynom section is terminated by the motor, else there will be created an error message and the motor will be stopped !

3.5.8 WAIT instruction

A "WAIT" instruction stops the execution of an operational program until the defined event comes true.

Syntax: WAIT *expression*
“**expression**“ is defined as the IF-request in chapter 3.5.1 (page 30)

Examples:
WAIT I1=1 (the program waits until input I1=1)
WAIT ADC>0.5 (the program waits until the value at the analogue input ADC is >0.5)

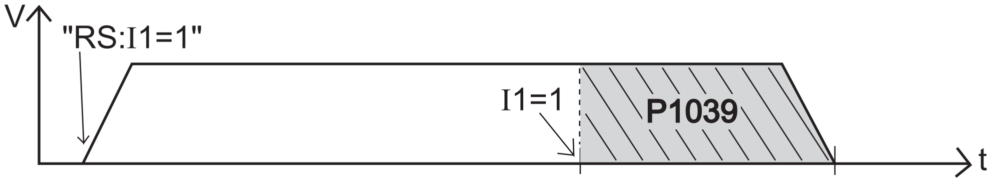
PLEASE NOTE:
An instruction using D2=... GS or D2=... GT... will terminate the WAIT-function (see description of parameter P1164 - D2 – in chapter 3.6).

3.5.9 “ : “ instruction – jog drive until STOP at an input

Manual driving (jog) with constant velocity until a specified event comes true (then the motor will be decelerated with the jog acceleration value P1018 and then stopped).

Syntax: RS:*expression* (alternatively to **RS** are allowed : **LS**, **RF** and **LF**)
“**expression**“ is defined as the IF-request in chapter 3.5.1 (page 30)
e.g. **RS : I1=1** → the motor is driving in jog mode (right slow – with the velocity P1019) until input I1 is set.

If parameter **P1039** <> 0 (a driving distance after stop is defined), the drive will move the relative distance defined in P1039 after the event came true (input is active e.g. at RS:I1=1 or input is deactivated at e.g. RS:I1=0) and then stops. The driving distance after stop (P1039) includes the distance moved during decelerating until motor stop.



PLEASE NOTE:
An instruction using D2=... GS or D2=... GT... will terminate the WAIT-function (see description of parameter P1164 - D2 – in chapter 3.6).

PLEASE NOTE:
Before using the “ : “ – function, the parameter P1035=0 must be set (may be done any time in the program, <u>**before**</u> programming the “ : “ – function).

Example :
Z=1 P1035=0
RS:I1=1

3.5.10 Arithmetical functions

A SERS Programmer E^{2}PROM program may include arithmetical functions.
There are following arithmetical functions:

- a 32-bit accumulator **X** - all arithmetical operations have to be assigned to the accumulator
- 100 Registers R0 to R99 (32-bit variables), which may be used for any assignments
- following arithmetical operations exist
  - Adding ‘**+**‘ , Subtracting ‘**–**‘ , Multiplying ‘*****‘ , Dividing ‘/‘
  - And ‘**&**‘ , Or ‘**|**‘ , Exclusive-Or ‘**^**‘
  - **NEG** : the accumulator’s sign will be inverted (positive values will be changed to negative, negative values will be changed to positive)
- all operations are executed from left to right (no point before line).
- arithmetical operations always have to be done via the accumulator X !
- there may be used all parameters and 32-bit constants in arithmetical operations.

Examples:

```
X=V*ADC+100   // the actual selected velocity V is multiplied with the value at the
V=X           // analogue input ADC (0 – 100%), and the value 100 is added

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

X=W
NEG    //The accumulator ‘s sign is inverted (X=–X)
W=X
```

3.6 Parameter for extended program mode

Parameter in the SERS-Programmer:

P100 (alternative syntax **C1**) counter 1
unsigned 16 bit counter (0 to 65536)

P101 (alternative syntax **C2**) counter 2
unsigned 16 bit counter (0 to 65536)

P102 (alternative syntax **C3**) counter 3
unsigned 32 bit counter (0 to 4,295 x 10^9)

Counters together with IF-requests can be used for programming loops.

E.g.: **#C1=20**
#L1
Instruction block
#IF C1>1 GT 1

In the example the *instruction block* will be executed 20 times.

P1117: Options Parameter Saving

With command PSAVE (P1004=2) all parameters (with the exception of registers Rx and marker Mx) can be saved into the E^2PROM of the SERS-Programmer, or in case PSAVE is sent to a SERS-drive, the they will be saved into the SERS-drive.
In case of an existing display menu in the SERS-Programmer, parameters can be edited here (see also chapter 3.3.1 syntax programming display menu, control character "!"). Editing may be done via the key “F1“, and the new value will be valid after confirmation with the “RET“ key. But the values are modified (editied) in the SERS-Programmer respectively in the SERS-drievs only in the corresponding RAM (that means after switching off the power supply the modified values are lost).
If the modified values shall be saved permanently, then this must be done by saving them into the E^2PROM, by using the command PSAVE.
Wether PSAVE refers to the SERS-Programmer or a SERS-drive, depends on the actual valid assignment via “Z=...“.
Z=0 → SERS-Programmer, Z=1 → SERS-drive with address 1, Z=2 ... .
Following options enable saving also registers Rx via command PSAVE, and command PSAVE can be executed automatically during editing (key “F1“) of values in the display menu.

Bit 0 (decimal 1) : when executing P1004=2 (command “PSAVE”), then additionally to the standard parameters the registers Rx (in SERS-Programmer "R0" to "R99" and in SERS-drives "R0" to "R11") will be saved into the E^2PROM. PSAVE will not be executed automatically when editing via key “F1“

Bit 1 (decimal 2) : Command PSAVE (for saving all standard parameters into the E^2PROM) will be executed automatically when editing via key ”F1”

Bit 2 (decimal 4) : PSAVE saves standard parameters and registers Rx (in SERS-Programmer "R0" to "R99" and in SERS-drives "R0" to "R11") into the E^2PROM, and PSAVE will be executed automatically when editing via key ”F1”

Saving all 100 registers (R0 to R99) into the E^2PROM of the SERS-Programmer needs approximately 1.2 seconds.
During this time "save..." will be displayed at the SERS-Programmer.

P1162: Last Key Value (key codes see chapter 3.7), edge triggered, that means the first pressed key since last time requesting the key pad (e.g. via X=P1162) will be stored, and after being requested it will be deleted.
Attention: in case of program loops, where P1162 is requested in every loop, a key press is lost, if not saved in a buffer memory.
P1162 also may be written, and also describes the key value for the menu selection, which is saved extra. With that a multiple evaluation of the same key can be evitated.

P1100: (alternative syntax D) Time Delay
$^1/_{10}$ seconds = 100ms (D=1), range 0.0 to 6553.5 (=655 seconds)
Smallest programmable delay: 10ms (D=0.1)
Example.: **D=20** → 2 seconds delay or **D=0.5** → 50ms delay
A program running is interrupted for the time delay **D**.

P1164: (alternative syntax: D2) Time Delay D2
$^1/_{10}$ seconds = 100ms (D=1), range 0.0 to 6553.5 (=655 seconds)
Smallest programmable delay: 10ms (D=0.1)

In contrary to to **D**, with **D2** the operational program will not be interrupted.
After a command with “D2=...“ the program continues immedeately.
After D2 expired, the Bit DB (P1165) will be set, and may be requested by IF-commands.
Following requests will be true, after D2 expired: "IF DB","IF DB=1","IF D2=0"
If there is a GT or GS command, directly after a D2 command, then the jump will be executed only after expiration of D2.

E.g. D2=10 GT30 W=360 E …:
The program first continues without the jump defined after the GT or GS command. In above example “W=360 E“ (and all commands after that) will be executed immedeately after the command “D2=10“. The program jump “GT30“ will be executed only after expiration of the Delay Time D2.

D2 also may be reloaded (extended) with command D2=value, as long as the jump was not executed.

ATTENTION: After the jump, because of expiration of Delay Time D2, also any of the following blockades will be reset:

1) Delay Time D (P1110) will be rest to zero (“0“).
2) "WAIT" and Stop (:) ... will be reset, and the program does not continue waiting
3) The program does not continue waiting for values requested from SERS drives (slaves) (e.g. Z=3 X=P51 or Z=3 IF POS=1).

Please note:
In case of a subroutine call, because of an expired Delay Time D2 (e.g. D2=10.0 GS 99), the actual program running may be interrupted at any program location, and it will continue also exactly at that point. But at the time it continues, also all blockades (see cases 1 to 3 above) will be canceled.

P1165: (alternative syntax: DB) Delay Time Bit
Will be set after expiration of Delay Time D2

P1182: Parameter line number
For programmed display menus, starting with "\a", after switching on the SERS-Programmer, the value in P1182 corresponds to the line number (between 1 and 4) which consists of the first parameter which can be edited. The cursor will be in this line.
In case of lists, which start with "\b", after switching on the SERS - Programmer, the cursor is always in the first line, and therefore P11182=1.
The “up“ / “down“ (↑ / ↓) keys move the cursor up / down in the display menu, and with that P1182 will be increased / decreased.
Writing P1182 (e.g. P1182=3) will set the cursor to the correspnding line in the display menu.
When writing P1182=0 the cursor will be hidden. But the output of the display menu needs to be executed at least one time after P1182=0.
With P1182 the last displayed parameter number can be stored, and initialised for new parameter lists. Also a limitation of the parameters which can be reached, is possible by writing P1182.

P1183: FB Option, default: 0

Bit 0 (decimal 1): enable Reset (pressing keys “ESC“ and “.“ simultaneously)
P1183=1 enables the Reset
→ P1183=1 is useful for a test-operation, during developing a program. This enables easy switching between programming mode and extended program mode (just via pressing keys, without power off/on)

Bit 1 (decimal 2): After expiration of D2, and in case the program shall jump to a label (by programming D2=..GT.. or D2=..GS..), the program will not wait for the actions as follows (which normally cause the program waiting):
a) the acknlowdgement (“Return” key) of error messages
b) Expiration of timeout at the serial interface (the timeout will be shortened to 20ms)
c) termination of pre-defined menus (P1187)
→ P1183=2 means the program will jump immedeately (after expiration of D2)

Bit 2 (decimal 4): after acknowledgment of an error message the actual program line number will be shown in the LCD (display):
e.g. "line:0114" if in line 114 there is a syntax error.
This makes it easy, finding the errors position, within the program (useful for test operation during developing a program in the SERS programmer)

P1184: read actual key value (codes see chapter 3.7)
Needs about 10ms time, because debouncing must be executed first
For evaluating with edges of the key pad, P1162 should be used.

P1185: Drive address
P1185 is pre-set from the address scan after power on (scan from addresses 9 to 1, the lowest found will be written to P1185).
The address menu (may be called by P1187=2) assigns the value to P1185.
P1185 also substitutes the "a" (e.g. "#aW?") for addressing in a text string.
The actual address will be read.

P1186: Baudrate SERS-Programmer:
1=1200 2=2400 3=4800 4=9600 5=19200 (default) 6=38400
all other values: 19200 Baud in the programming mode, and automatical scan (of the connected SERS-drives) when executing a program (in the extended program mode)

P1187: Menu call (see also chapter 3.8 "pre-defined menus")
value range 0..9
By assigning the menu number the corresponding pre-defined menu will be executed (e.g. P1187=2 → address menu will be executed).
The program will be stopped during execution of the menu.
The menus work as in the standard mode of the SERS-Programmer, but after exiting the menu with key ESC or RET, the (stopped) program will be continued, and there will be no jump into the higher ranking menu (as it would be in the standard menu mode).
If anything will be transmitted to a SERS drive, then always to the actual choosen address (actual stored in P1185).
Menus (which can be called with P1187) are described in chapter 3.8.

P1188: for each (SERS drive) address existing, one bit will be set in P1187:
address 1: 1 address 2: 2 address 3: 4 address 4: 8
address 5: 16 address 6: 32 address 7: 64 address 8: 128
address 9: 256

Example: existing drive addresses 1, 3 and 7 → P1188=69

P1189: Error mask for L65 call (default: 3)
Label L65 will be called, after acknowledgementing an error (possible errors are described below at Bit 0 to Bit 2), only if at least one of following Bits in P1198 are set:

Bit 0 (decimal 1): timeout at the serial interface (after about 4 seconds)
Bit 1 (decimal 2): if in the adress menu an invalid address was choosen
Bit 2 (decimal 4): drive (SERS) returns an error message

The executable program will be stopped, until the user acknowledged the error message by pressing a key. After that L65 will be called (if the corresponding Bit is set in P1189), for enabling a correct program continuation.
The destination address "Z" will be set "Z=0", before jumping to the error label L65.
If L65 is not defined in the program, the the program will be restarted.

R0 to **R99**: free usable registers
R12 to R99 are limited to 2 decimal points in case of assignment of constants
Example: R12=20.5682 → only R12=20.56 will be stored

M1 to **M20**: free usable markers (range value : "0" and "1")
marker cannot be saved into the E^2PROM, and therefore are kept only in the RAM during program operation.

All parameters for SERS drives are described in the manuals "sers_v....pdf" and "wsers_v...pdf".

3.7 Key values in case of key pad requests

Key values (for requests with P1162 and 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 Predefined menus

Pre-defined menus can be called by assigning the corresponding numbers to parameter P1187.
"**LCD:**" shows the screen after calling the menu.
"**Keys:**" describes the active keys and the reaction after pressing them.
If any characters will be sent, then always to the address which was called recently, respectively to the last selected address, respectively to the address which was found by autoscan (P1185).

P1187=

0: Z_MENU_MASTER (= no pre-defined menu)
Display and selection of parameters defined by the operational program.

LCD:
The display will be defined by the user program (e.g. output between "\a..." and "\e").
The cursor will be set to the selected parameter line number.

Keys: (evaluation by the command "\e")
"up" decrements the parameter line number (P1182) by 1 (if possible)
"down" increments the parameter line number (P1182) by 1 (if possible, maximum counts of the lines between "\a" and "\e")
F1 edits the selected parameter

1: Z_MENU_LANGUAGE

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

Keys:
ESC or RET exits the menu, jump to Z_MENU_MASTER
STOP will send the character "s"
F1 searches for the baudrate and then the drive address (only 0..3)
If no Baudrate could be found, then the program jumps to menu Z_MENU_BAUDRATE
F2 jumps to menu Z_MENU_ADRESSE
F3 sends the string "p265=0psave"
F4 sends the string "p265=1psave"

2: Z_MENU_ADDRESS

LCD: **F1=AUSWAHL BAUD**

AUSWAHL ACHSE ?
3

Keys:
ESC or RET reads the language (P265) from the drive and exits the menu, jump to Z_MENU_MASTER
F1 jumps to Z_MENU_BAUDRATE
"up" increments the drive address (P1185)
"down" decrements the drive address (P1185)

3: Z_MENU_BAUDRATE

LCD: **AUSWAHL**
BAUDRATE
?
38400

Keys:
ESC or RET exits the menu, jump to Z_MENU_MASTER
"up" selects the next higher Baudrate
"down" selects the next lower Baudrate
The RS232 interface will be initialised with the new Baudrate after exiting the menu.

4: Z_MENU_RESET_ERROR

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

Keys:
ESC or RET exits the menu, jump to Z_MENU_MASTER
F1 sends the string "p11=0" and shows an acknowledgement text
F2 sends the string "p12=0" and shows an acknowledgement text

5: Z_MENU_PARAMETER

LCD:
F1=EDIT F2=SAVE
P0=0
Z=0
P2=0
...
P1301=0

Keys:
ESC or RET exits the menu, jump to Z_MENU_MASTER
STOP sends character "s"
F1 edits the selected parameter
F2 sends the string "psave"
"Pgup" decrements the parameter line number by 10 (if possible)
"up" decrements the parameter line number by 1 (if possible)
"Pgdwn" increments the parameter line number by 10 (if possible)
"down" increments the parameter line number by 1 (if possible)

6: Z_MENU_HAND

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

Keys:
ESC or RET exits the menu, jump to Z_MENU_MASTER
STOP sends character "s" (stop motor motion and program in SERS drive)
F1 sends strings "P11=0 P1035=1" and "ls" (jog left slow)
(each repeated as long as the key is pressed, after that "P1031=0" is sent)
F2 sends "P11=0 P1035=1" and "rs" (jog right slow)
F3 sends "P11=0 P1035=1" and "lf" (jog fast left)
F4 sends "P11=0 P1035=1" and "rf" (jog fast right)
OFF sends "off" (motor phase current off)
ON sends "on p11=0 p12=0" (motor phase current on, and reset errors / warnings)

7: Z_MENU_HOMING

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

Keys:
ESC or RET exits the menu, jump to Z_MENU_MASTER
F1 sends the character "h" (start homing)
F2 or STOP sends "s"
OFF sends "off"
ON sends "on p11=0 p12=0"

8: Z_MENU_START (program start)

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

Keys:
ESC or RET exits the menu, jump to Z_MENU_MASTER
F1 sends the string "run" (start program in SERS drive)
F4 or STOP sends the character "s"
OFF sends "off"
ON sends "onp11=0p12=0"

9: Z_MENU_TERMINAL

LCD:
First it will be erased and then each character received will be displayed

Keys:
ESC or RET exits the menu, jump to Z_MENU_MASTER
sends the corresponding character or string via the serial interface
(also works with selection of characters → via the key "List", see chapter 2.13 Terminal mode in standard mode, pages 15 - 16)

3.9 Notes on programming in the extended program mode

There are some typical error possibilities as follows:

Display (LCD)
For avoiding a flickering LCD display, the command "\c" should not be used for Clear Screen (LCD), in case of a cyclic display.
Instead of it the LCD should be written explicitely, e.g. by using "\1" for the output of the first line.
The command "\n" deletes all characters until end of line, and results in building up a correct display also in case of position outputs with different lengths.

Key pad
For not loosing keypresses, the key pad may be read via P1162 only once per iteration loop. A key press will be saved until the request via P1162, and after that it will be deleted. In case P1162 will be requested a second time within one iteration loop, then the key value will be “0” (what means the keypress is lost).
For evaluation with multiple "IF" comparations, command X=P1162 must be used for buffering the keypress.

Destination address
The destination address "Z" must be set correctly, in case of controling multiple SERS drives, and in case of assignments and requests of parameters at the SERS-Programmer and SERS-drives in an operational program.
E.g. if the corresponding assignment ”Z=0” for an IF-request of a parameter of the SERS-Programmers was forgotten - e.g. P1184 for reading the actual keypress value (and shortly before there was assigned a parameter to a SERS-drive by “Z=1”), then there will be an error message "unknown if event" (because the parameter does not exist in the destination drive - P1184 does not exist in a SERS-Slave, because it is a pure SERS-Programmer parameter).

Assignments / Instructions
The single instructions will be converted in 16 Bit and 32 Bit words and saved.
This results in less program space, and the program will run more quickly by that.
But a limitation exists for the number of usable parameters per assignment.
Assignments may include only one parameter and one constant, or one parameter and the accumulator "X".

Allowed examples:

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

Not allowed examples:

```
P51=P138   R1=R0
```

Two parameter addresses do not have sufficient space in the internal 16 Bit Opcode
Here the accumulator “X“ must be used:

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

Arithmetical operations always use the accumulator "X" for storing the result and as first operand.
e.g.   W = ADC * (P51 + P41)
must be programmed as:

```
X=P51 +P41 *ADC
W=X
```

3.10 Program examples for the extended program mode

Example 1 – cable cut machine
Display 1. line: **F1=EDIT F2=START**
Display 2. line: **LENGTH: xxx**
Display 3. line: **COUNT: xxx**
Display 4. line: **CNT:xxx POS:xxx**

Parameter (variable values) which can be edited:
Length (LENGTH), piece counter (“COUNT“)

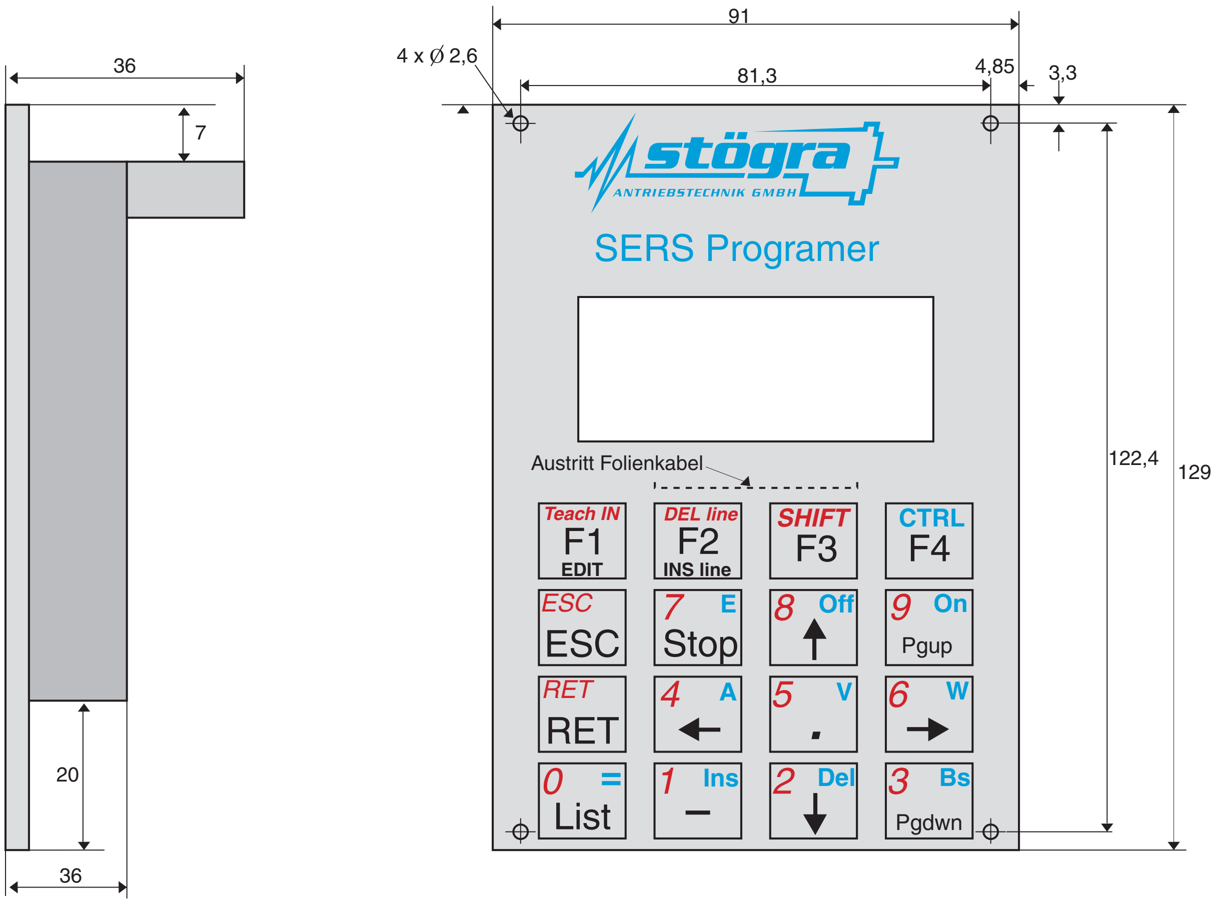
Display (only displayed values):
Pieces finished (“CNT“), actual motor position (“POS“)

F1=EDIT F2=START
LENGTH: 360
COUNT: 20
CNT:15 POS:1800.

Operational program (explanations after “//“, SP = SERS Programmer)

```
R2=0 R3=0 P51=0        // initialising of the used parameter
L1                     // program-Label – jump destination L1
GS20                   // subroutine call Label L20 (here menu display)
X=P1162 IF X<>241 GT 1 // request key pad for key “F2“
X=R1 C1=X              // assignment R1 (in SP) to counter C1 (in SP)
X=R0 WR=X V=50         // assignment X (of SP) to WR (of drive 1), speed (V) = 50
R2=0 R3=0 P51=0        // register R2 and R3 = 0, and actual position (P51) = 0
GS20                   // subroutine call label L20 (here menu display)
L2                     // counter loop via C1 and request of C1 (see below)
WAIT I1=1              // wait for input I1 of drive 1
O1=0 E                 // reset output O1 (drive 1) and start drive 1
GS 10                  // subroutine call label L10
IF C1>1 GT 2           // end counter loop (return to L2 if C1>1)
GT 1                   // return to begin (jump to label L1)
L10                    // program label, begin of subroutine L10
GS20                   // subroutine call Label L20 (here menu display)
IF POS=0 GT 10         // as long as drive 1 is positioning, return to L10
O1=1                   // output O1=1 at drive 1
X=R2+1 R2=X            // R2=R2+1 (for display of finished pieces)
GS20                   // subroutine call Label L20 (here menu display)
RT                     // end subroutine L10 and return
L20                    // program label, begin of subroutine L20
"\a"                   // begin display menu (display SERS-Programmer)
"F1=EDIT F2=START"     // only output of text “F1=EDIT F2=START”
"LENGTH:#0R0!"         // “LENGTH“ + R0 (of SP), R0 is editable (because of “!“)
"COUNT:#0R1!"          // “COUNT“ + R1 (of SP), R1 is editable (because of “!“)
"CNT:#0R2? POS:#1P51?"  // “CNT:“ + C1 (of SP)  + “POS:“ + P51“ (of drive 1)
"\e"                   // end display menu
RT                     // end subroutine L20 and return
```

4. Dimensions



5. Technical specifications

Program memory: 8KByte E^2PROM

Voltage supply via Pin 9 (of 9-pole D-Sub): 8VDC (6 – 12VDC, ripple < 10%)
- the 8VDC will be supplied by a connected SERS-drive

Voltage supply via 2-pole connector: 24VDC (20 – 28 VDC, ripple < 10%)
- needed only in the programming mode (for programming the SERS Programmer via a PC)

Current load : 30 mA

Protection : without housing : IP00 (front side IP65)
with housing : IP41 (front side IP65)