CL500

Diagnostic Module DB500
Module Description

Edition

102

CL500

Diagnostic Module DB500
Module Description

1070 072 128-102 (92.10) GB

Discretionary charge 20.00 DM

Contents

Page

Contents

Contents

1 General Information

This description applies to the DB500 version 1.6 or index 105, in connection with the standard module MADAP500, version 105.

1.1 Requirements for operating the DB500

The DB500 is an intelligent system module with its own processor and its own user memory.

It is primarily operated in the CL500 basic unit for intelligent system modules and communicates with the control system via the system bus which is managed by the SK500 System Coordinator.

This module description should be used in conjunction with the following hardware, software and documentation:

- DB500 Hardware
- DB500 Utility program on diskette for communication between the programming unit and the DB500.
- Description of the **MADAP500** Module P.–Nr. 4142. The neces-sary TRANS500 and DIAG500 standard function modules are on the MADAP500 diskette.

1.2 Functions

Diagnosis

The diagnostic functions consist in monitoring the sequential step cascades of a machine or plant that are automatically diagnosed in the event of a fault.

The DB500 is able to simultaneously monitor up to 64 cascades with 128 steps each.

Monitoring of production sequences

This includes:

- Cycle time display
- Limit control
- Graphics display
- Alarm messages
- Logging of fault and operating messages

Status displays

The status display reports the signal status of the operands in the following forms of representation:

Operands	Form of representation
● Inputs	Binary, Hex, Dec, ASCII
● Extended inputs	Binary, Hex, Dec, ASCII
● Outputs	Binary, Hex, Dec, ASCII
● Extended outputs	Binary, Hex, Dec, ASCII
● Markers	Binary, Hex, Dec, ASCII
● Special markers	Binary, Hex, Dec, ASCII
● Data modules	Binary, Hex, Dec, ASCII
● Data buffers	Binary, Hex, Dec, ASCII
● Data field	Binary, Hex, Dec, ASCII
● Timers	Status, time value, time base
● Counters	Actual value

The binary representation can be with “0” and “1” or symbolic (□, ■).

Non−displayable ASCII characters are ignored and represented by the character “−”.

1.3 DB500 – Networking and Communications

The DB500 can communicate with

- the programming unit or printer/terminal
- a BAS monochrome monitor or PLASMA display
- a colour monitor (EGA standard) or TTL mono monitor

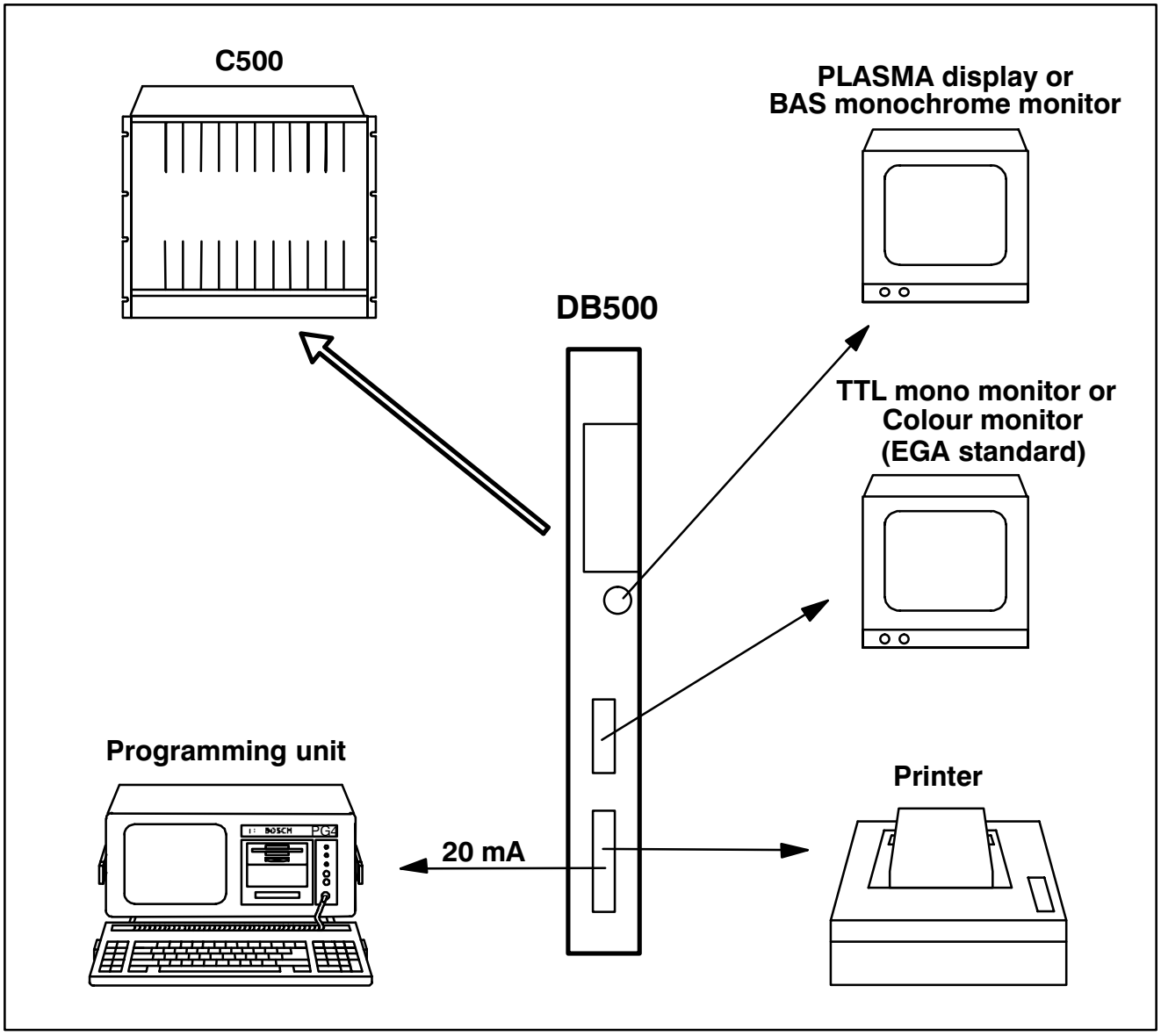


Fig. 1–1 DB500 – Networking and Communications

The programming unit of the DB500 is operated as an intelligent system module in the Bosch CL500.

The DB500 operates in conjunction with the standard function modules TRANS500 und DIAG500 which are incorporated in the user program of the CL500.

Networking with the Programming Unit

A great number of texts and screens for the status and fault messages are managed and processed in the DB500 and are input by the user by means of the programming unit.

Networking with monochrome BAS monitor

The monochrome monitor displays the internal screens and those edited by the user.

Networking with colour monitor Screens can be displayed on the colour monitor in a number of colours which are selected by means of attributes.

Networking with printer A printer can be used to log messages with date and time information. A hard copy of the momentary screen is also possible.

2 Equipment Description

2.1 Configuration

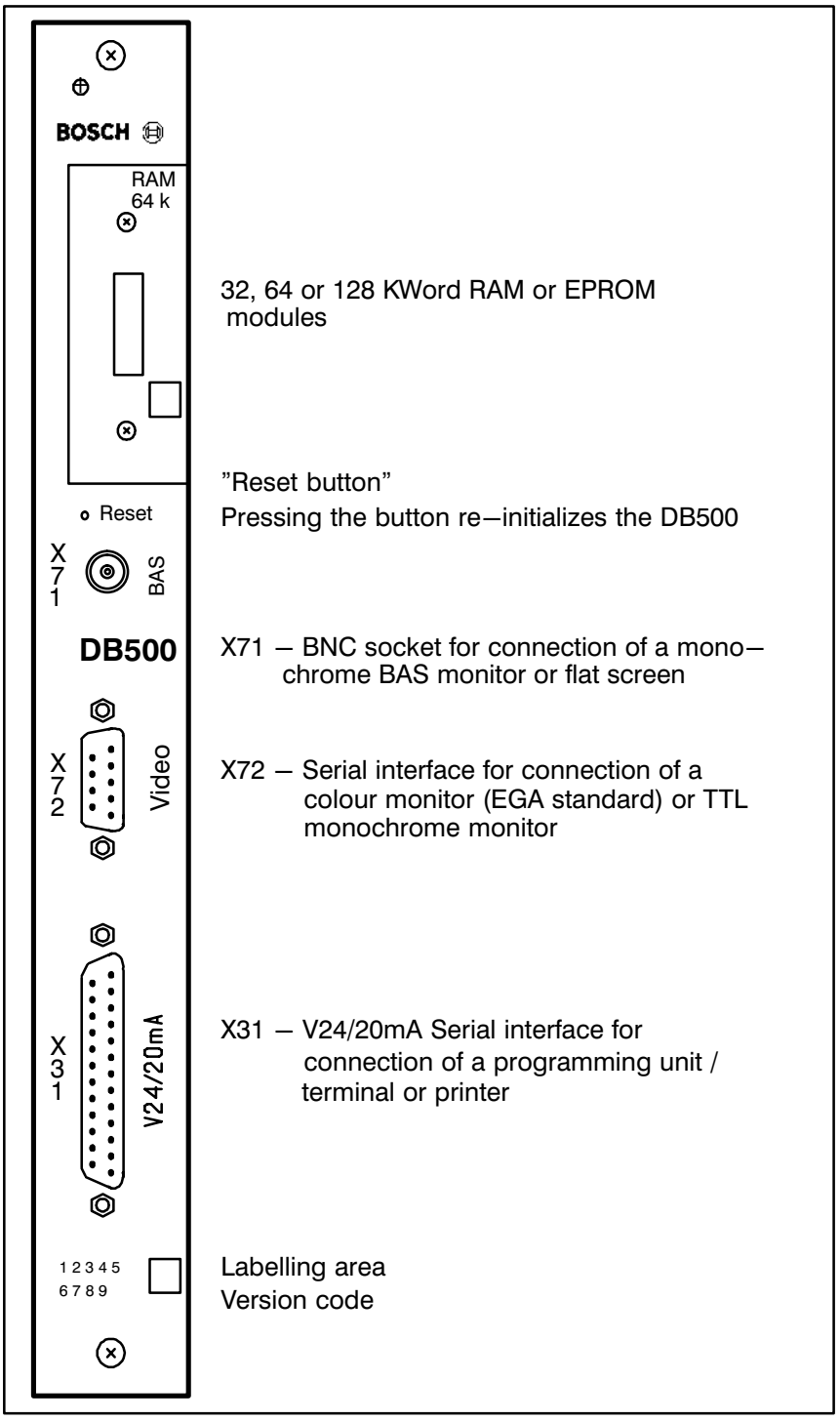


Fig. 2–1 DB500 Front Panel

2.2 Memory

Text and Screen Memories

The I/O/M/SM/T/C texts, cascade texts, customer specific messages, the texts for various types of display as well as the screens and function keys are stored in 32 KWord, 64 KWord or 128 KWord RAM or EPROM modules which can be plugged into the DB500 as required. The RAM module is buffered centrally by a battery located on the power supply module of the CL500.

Record memory

The DB500 incorporates a 32 KWord record memory which stores all diagnosis, DESI and customer messages.

The contents of the record memory can be sent to the display or printer as output devices, or cleared.

It is buffered centrally in the CL500 power supply module.

Note ☞

The DB500 is buffered centrally in the NT2/NT3 power supply module and monitored by the system coordinator. The data buffered in the RAM memory will be lost if the DB500 or the NT2/NT3 power supply module is unplugged or the battery removed.

2.3 Clock

The clock is incorporated in the SK500 system coordinator and is available to the DB500.

The clock provides data on the year, month, date, weekday, hour, minute and second.

The clock is set using system commands from the ZS500 (see Operation List P.–Nr. 4120), by MADAP (see description P.–Nr. 4142) or by means of the programmer function 'Setting clock' (Only as from SK version 3 and programmer software version 3.0).

2.4 Battery failure

Each time the control system is activated the software performs a battery test. A battery failure message is displayed on the monitor as an alarm message “Buffering interrupted”.

After a battery failure message, proceed as follows:

- Call function **KC=INIDB** < Enter >
  or press reset button on the front panel of the DB500.
- Reload DG5 File

Note ☞ **Battery replacement on the NT2/NT3 is described in the Manual P.–Nr. 4090 for the CL500.**

As from version 1.4 (EPROM 103), INIDB does not have to be called when no special settings (e.g. language, baud rate) are necessary.

2.5 Interfaces

The DB500 module has three different interfaces which are labelled on the front panel as follows:

- X31: Serial interface V.24/20mA for connecting a programming unit or printer
- X71: BAS signal at a BNC socket for connecting a monochrome monitor or flat screen
- X72: Interface for connecting a TTL colour or monochrome monitor

20mA Interface X31

The 20mA interface conforms to the requirements of VDI 2880, Part 2 (max. off−state voltage 27 V) for programmable logic controllers, process and data interfaces.

The 20mA interface can be wired actively, i.e. with current source, or passively depending on the plug layout.

Line states: Logical 1 −−−−−> 20 mA
Logical 0 −−−−−> no current

20 mA / active

Item	Designation	Connector No.	Signal direction
Shield	Shield	1	
Receive data +PLC	RxD +	12	<−−−
Receive data −PLC	RxD −	24	
Transmit data +PLC	TxD +	13	−−−>
Transmit data −PLC	TxD −	25	
Data Set Ready	DSR +	14	<−−−
Data Set Ready	DSR −	18	
Reader Control +	RDRCTL +	16	−−−>
Reader Control −	RDRCTL −	21	

Notes ☞

For active operation the connections 9 (12V IN) and 10 (12V OUT) must be bridged.

The 20 mA interface is used for connecting the programming unit. Connection to a printer is also possible.

20 mA / passive

Item	Designation	Connector No.	Signal direction
Shield	Shield	1	
Receive data +PLC	RxD +	22	<–––
Receive data –PLC	RxD –	12	
Transmit data +PLC	TxD +	23	–––>
Transmit data –PLC	TxD –	13	
Data Set Ready	DSR +	11	<–––
Data Set Ready	DSR –	14	
Reader Control +	RDRCTL +	19	–––>
Reader Control –	RDRCTL –	16	

V.24 (RS232) Interface X31

The signal voltage levels and the layout of the plug connector conform to the requirements of VDI 2880, Part 2 for programmable logic controllers, process and data interfaces.

Signal level (data line):

Logical 1 ––>	–25V	to – 3V
Logical 0 ––>	+ 3V	to +25V

Signal level (message and control line):

active ––>	+ 3V	to +25V
passive ––>	–25V	to – 3V

Item	Designation	Connector No.	Signal direction
Shield	Shield	1	
Transmit data	TxD	2	–––>
Receive data	RxD	3	<–––
Reference conductor (Signal Ground)		7	
Data Set Ready	DSR	6	<–––
Data Terminal Ready	DTR	20	–––>

2.5.1 Connector Cables and Cable Lengths

The connection must be made using a shielded and twisted cable. When running the cable, ensure that the data line does not run parallel with any power cables.

Characteristic data of connector cable for the V.24/20mA interface

The connector cable for the **V24/20mA interface** must not exceed the following characteristic data:

Line resistance: 138 Ω/km
Capacitance: 120 pF/m

Line lengths

The data is based on a shielded and twisted transmission cable 14 x 0.14 mm^2 with the following characteristic data:

Bit rate	Max. length (V24)	Max. length(20mA)
19200 Baud	15 m	75 m
9600 Baud	15 m	150 m
.		
.		
110 Baud	15 m	150 m

Note ☞ **With the V24 interface, note that the potential differences between transmitter and receiver should not exceed −2V < = V diff < = +2V.**

Characteristic data of BAS cable

Maximum length: 100 m
Surge impedance 73 ±3 Ω
Cable capacitance 70.5 pF/m

Characteristic data of TTL cable

Maximum length: 3 m

This length was determined with a shielded and twisted extension cable (14 x 0.14 mm^2) with the following characteristic data:

Line resistance:	138 Ω/km
Capacitance:	120 pF/m

The cable shield was attached to the housing of the D plug.

2.5.2 Plug layout

Plug X72 (TTL Video) — D socket, 9 pin

Pin	Mono	Colour
1	GND	GND
2	GND	GND
3	–	red
4	–	green
5	–	blue
6	intensity	GND
7	video	GND
8	HSYNC+/–	HSYNC+/–
9	VSYNC+/–	VSYNC+/–

Plug X31 (Serial Interface) — D socket HD20, 25 pin

Pin	V24	20mA active	20mA passive
1	Shield	Shield	Shield
2	TXD		
3	RXD		
4	–		
5	–		
6	DSR		
7	SIGGND		
8	–		
9		12 V IN	
10		12 V OUT	
11			DSR+
12		RX+	RX–
13		TX+	TX–
14		DSR+	DSR–
15			
16		RDRCTL+	RDRCTL–
17			
18		DSR–	
19			RDRCTL+
20	DTR		
21		RDRCTL–	
22			RX+
23			TX+
24		RX–	
25		TX–	

2.6 DB500 Settings

Note ☞ **The DB500 should be set before operation.**

Control signals, bit rate and transmission format for the X31 interface are set using DIP switch **S1**. A programmer or printer can be connected to this socket. DIP switch **S2** is used to setthe block address of the DB500.

The selection of the BAS monitor, flat screen (X71 interface), mono TTL monitor or colour monitor (X72 interface) is carried out with DIP switch **S4**.

The settings are kept when the DB500 is re−initialized (power **on**/**off**, reset button or KC command **'INIDB'**).

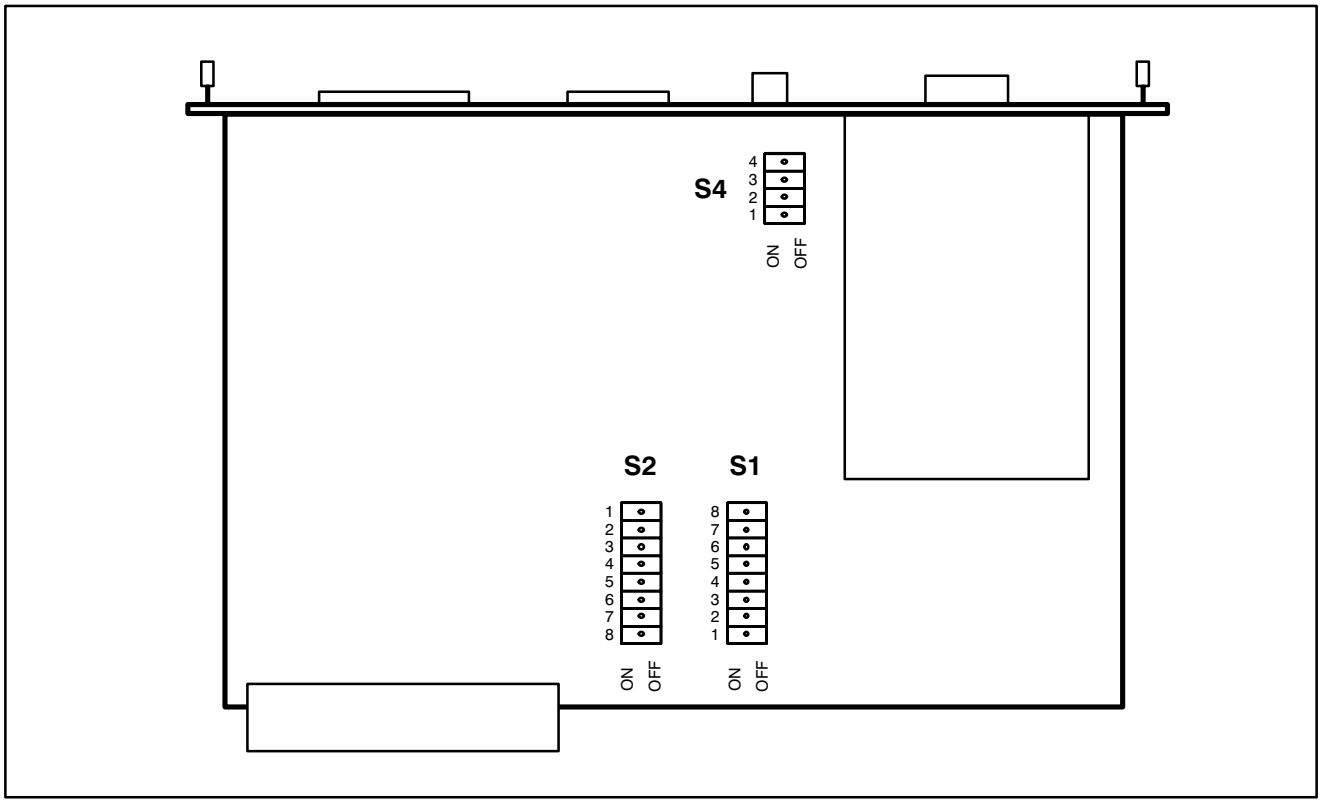


Fig. 2−2 Layout of switches on the DB500

2.6.1 Setting the parameters of interface X31 with DIP switch S1

2.6.1.1 General Information

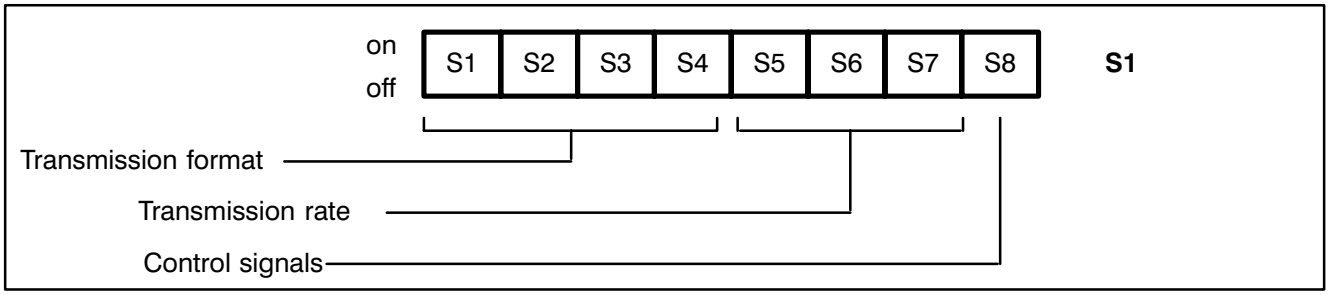


Fig. 2–3 Settings with the S1

Control signals

The scanning and non–scanning of the control signals is done using DIP switch **S1** with switch number 8, as follows:

Scanning control signals: Switch number 8 to “on”
Non–scanning control signals: Switch number 8 to “off”

Baud rate

The settings are as follows:

S5	S6	S7	Baud rate
on	on	on	19200 Baud
off	on	on	9600 Baud
on	off	on	4800 Baud
off	off	on	2400 Baud
on	on	off	1200 Baud
off	on	off	600 Baud
on	off	off	300 Baud
off	off	off	110 Baud

Transmission format

The settings are as follows:

S1:	1 Stop Bit	: on
	2 Stop Bits	: off
S2:	7 Information bits	: on
	8 Information bits	: off
S3:	Even parity	: on
	Uneven parity	: off
S4:	Parity	: on
	No parity	: off

The serial data signal is supplied to the 25−pin Cannon plug connector as a V24 signal and as a 20 mA signal. In both cases the potential separation is achieved using optical couplers.

2.6.1.2 Settings for the programming unit and printer

The **X31** interface can service either a programmer or a printer.

The parameters for the programming unit are set using DIP switch **S1**. These are the **first parameters** and are automatically activated after every **Power−On** and **Reset**.

The **second parameters** for the printer can be set for all outputs via the interface with the command **INIDB**. To return to the **first parameters** simply perform a **Reset** on the module.

Setting the first parameters for the programmer

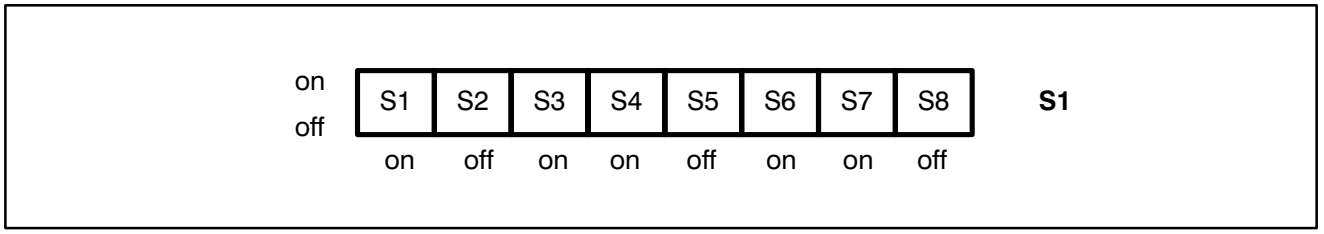


Fig. 2−4 Settings with S1 for the programming unit

Setting the printer parameters

If after the command **INIDB** the answer to the question **"Set parameters"** is **"Yes"**, they will apply to all outputs from that moment on. A printer can be connected after the input of the **second parameters**.

As a check, the newly changed parameters are issued again. They can be corrected by **"Change parameters" = Y**.

Example of an input for the Second Parameters

KC=INIDB

Sprache/Language/Lengua/Taal

Deutsch : 1
English : 2
Espanol : 3
Nederlands : 4 [1]

Current parameter setting: — shows the current parameters

Character set : IBM
Baud rate : 9600 Baud
Parity : even
Data bits : 8
Stop bits : 1
Control signals : no

Change parameters (Y/N) ? : Y

Character set (I) BM / (A) SCII ? [I] : I — Standard or extended character set

1 = 19200
2 = 9600
3 = 4800
4 = 2400
5 = 1200
6 = 600
7 = 300
8 = 110

Baud rate	(1–8) [1] : 2	– Enter a number or confirm the prompt
Parity	(Y/N) [Y] : Y	
even	(Y/N) [Y] : Y	
Data bits	(7/8) [7] : 8	
Stop bits	(1/2) [1] : 1 or Return	
Control signals	(Y/N) [N] : N	

Current parameter setting:

Character set	:	IBM
Baud rate	:	9600 Baud
Parity	:	even
Data bits	:	8
Stop bits	:	1
Control signals	:	N

Change parameters	(Y/N) ? : N
Set parameters	(Y/N) ? : Y
Initialize DB500	(Y/N) ? : Y

Record memory cleared Monday, 24.04.1992 11:38:24

Printing parameters for hard copy printout

KC=PRINT

Hardcopy printout Monday, 24.02.1992 12:15:45

Parameters for hardcopy printout

First level	:	YES
Messages	:	YES
DESI	:	YES
Printout:		complete
Direction	:	LIFO

Change parameters	(Y/N)	:
First level	(Y/N)	:
Messages	(Y/N)	:
DESI	(Y/N)	:
Complete	(Y/N)	:
LIFO	(Y/N)	:

Example of the settings on printer ML 182 and layout of plug contacts of the cable connecting the printer to the DB500

Settings on printer ML 182

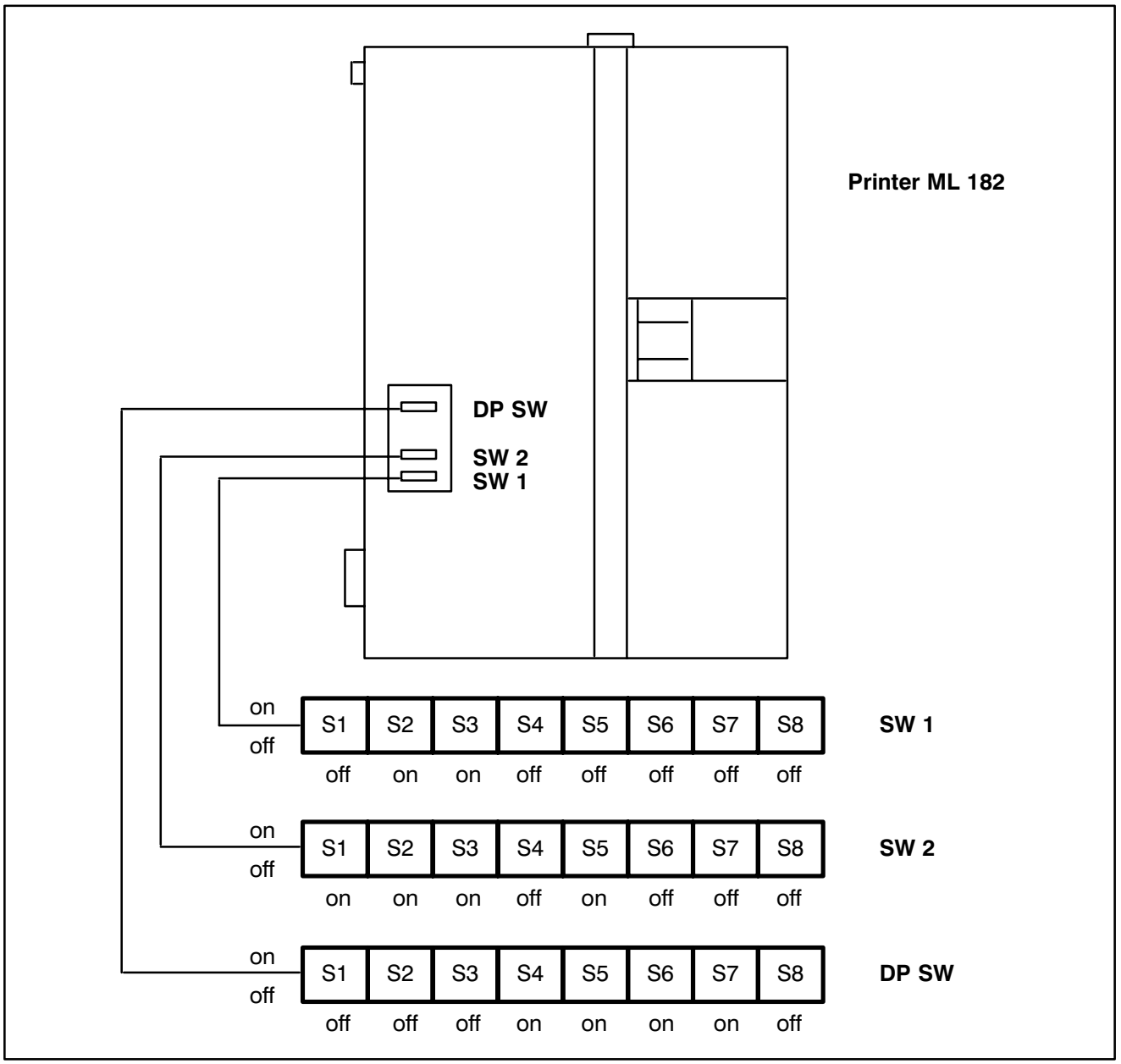
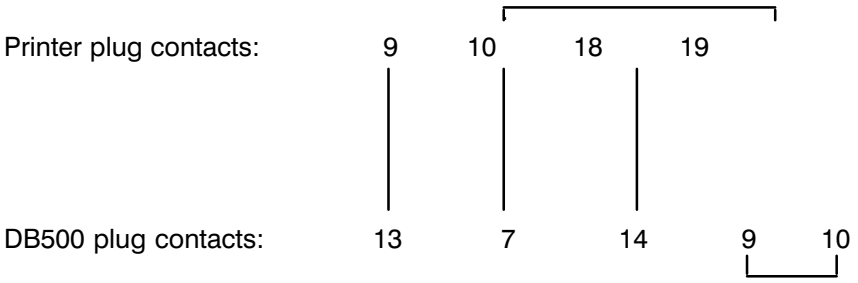


Fig. 2–5 Settings on printer ML 182

Layout of plug contacts of the cable connecting the printer to the DB500



Setting the block address of the DB500 with DIP switch S2

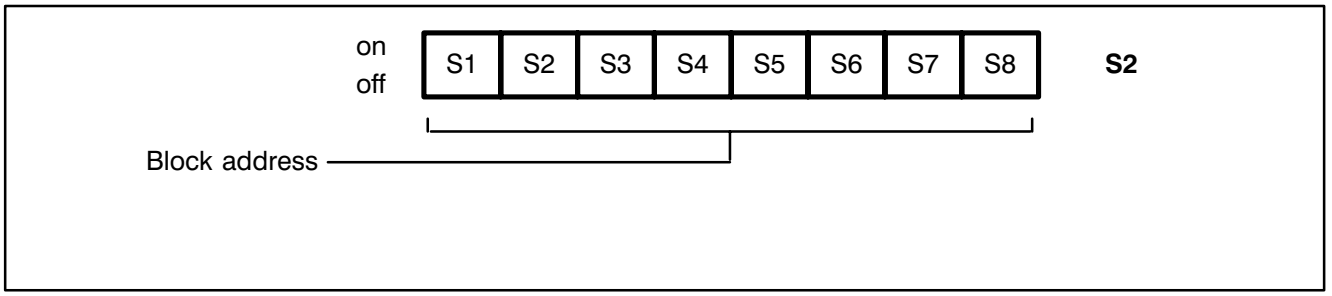


Fig. 2–6 Settings with switch S2

The DB500 occupies 4 blocks on the system bus, and so the card address (block address) must always be set as a multiple of 4.

The block address of the first DB500 depends on how many ZS500 are being operated in the CL500.

Since the block address of the ZS500 has to be a multiple of 8, the block address of the first DB500s to be addressed could begin with

8 for 1 ZS500,
16 for 2 ZS500,
24 for 3 ZS500,
32 for 4 ZS500, and continue in steps of 4.

Selecting displays and language by setting DIP switch S4

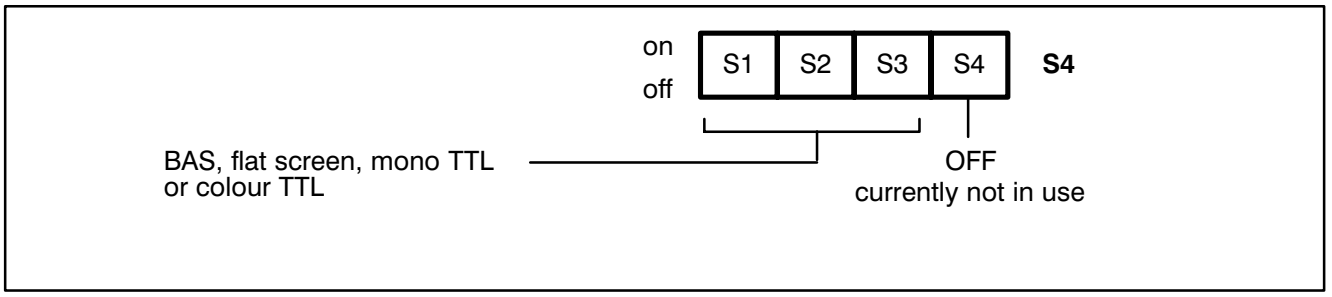


Fig. 2–7 Settings with switch S4

Selecting display

Switch number S1	S2	S3	Selected monitor
on	on	on	BAS monitor
off	off	on	Flat screen with EGA Timing via the BNC socket
off	on	on	Mono TTL monitor, positive VSYNC
on	off	on	Mono TTL monitor, negative VSYNC
on	on	off	Colour TTL monitor, positive VSYNC
on	off	off	Colour TTL monitor, negative VSYNC

In all 4 operating modes a maximum of 26 lines with 80 characters are represented on screen.

2.7 Screen modes

As mentioned above, the DB500 supports BAS, flat screen, mono and colour TTL monitors.

For operating the colour monitor, the DB500 has two memories (one character and one attribute memory). Each character on screen is assigned its own attribute (colour, flashing ...). All characters (text and attributes) can be simultaneously displayed. The BAS and TTL signals are optically separated and are externally available via a BNC socket and a 9–pin D subsocket. All signals at the D subsocket and the BNC socket are potential–linked (ISOGND 5V bridged with shield at panel).

2.7.1 BAS mode

Technical data of the BAS signal

In the BAS mode, the display works with a 9 x 10 character set.

Horizontal frequency	15,625 kHz (CCIR Euro TV Standard)
Vertical frequency	50 Hz
Line duration	64 µs
Visible line duration	50.6 µs
Frame rate	20 ms
Characters per line	80
Character size	9 x 10 Bildpunkte
Character lines per frame	26
Frame dots	720 x 260
Dot frequency	14.22 MHz
Signal level	BAS (CCIR Standard)
High–Intensity (Uss)	0.9 V (UHIGH – USYNC)
Low–Intensity (Uss)	0.6 V (ULOW – USYNC)
Sync–Level (Uss)	0.3 V (USYNC – UREF)
Sync–Low (with ref. to GND)	0.2 V (UREF)
High–Int (with ref.to sync)	1.2 V (UHIGH – UREF)

2.7.2 Flat screen mode

In this mode, the same attributes can be used as in the BAS mode. However, not all flat screens are able to process more than one brightness step.

Technical data of the flat screen

The flat screen mode works with an 8 x 15 character set.

In the flat screen mode the BAS signal with EGA timing is present at the BNC socket.

Horizontal frequency	21.85 kHz (EGA Standard)
Vertical frequency	52 Hz
Line duration	45.8 µs
Visible line duration	40.0 µs
Frame rate	19 ms
Characters per line	80
Character size	8 x 15 frame dots
Character lines per frame	26
Frame dots	640 x 390
Dot frequency	16 MHz
Signal level	BAS (CCIR Standard)
Resolution of the connected display	640 Dots x 400 Lines

2.7.3 Mono TTL Mode

In this mode, the same attributes can be used as with BAS operation.

Technical data of the mono TTL screen

The mono TTL mode works with an 8 x 13 character set.

The polarity of the VSYNC signal is selected via switch **S4**.

Horizontal frequency	18.432 kHz (Hercules Standard)
Vertical frequency	50 Hz
Line duration	54.3 µs
Visible line duration	45.0 µs
Frame rate	20 ms
Characters per line	80
Character size	8 x 13 frame dots
Character lines per frame	26
Frame dots	640 x 338
Dot frequency	14.22 MHz
Signal level	TTL
VSYNC	Negative (invertible)
HSYNC	Positive
Resolution of connected monitor	720 Dots x 350 Lines (Hercules Standard)

2.7.4 Colour TTL Mode

Technical data of the colour TTL screen

The colour TTL mode works with an 8 x 13 character set.

The polarity of the VSYNC signal is selected via switch **S4**.

The RGB signals are therefore EGA−compatible.

Horizontal frequency	21.85 kHz (EGA−Norm)
Vertical frequency	60 Hz
Line duration	45.8 µs
Visible line duration	40.0 µs
Frame rate	16.7 ms
Characters per line	80
Character size	8 x 13 frame dots
Character lines per frame	26
Frame dots	640 x 338
Dot frequency	16 MHz
Signal level	TTL
VSYNC	Negative (invertible)
HSYNC	Positive
Resolution of the connected monitor	640 Dots x 350 Lines (EGA Standard)

2.8 DB500 Slots

The DB500 can be operated in the CL500 basic unit within slots 5 to 14.

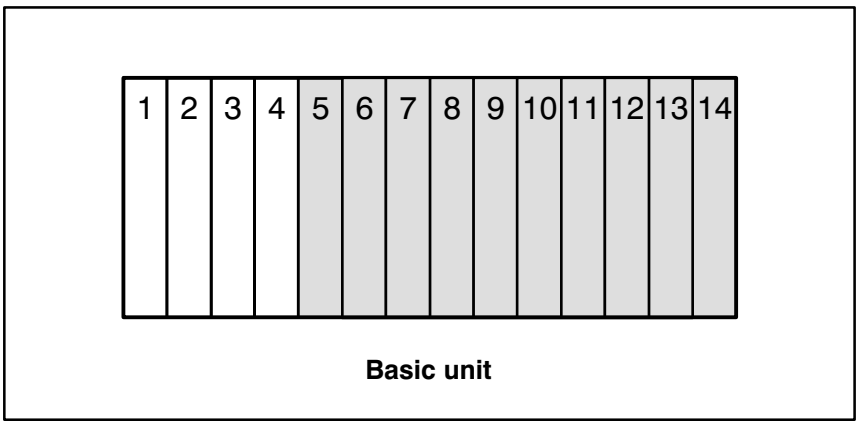


Fig. 2–8 Possible slots for the DB500 in the GG500 basic unit.

2.9 DB500 Operating Modes

All modes are selected by the user in the TRANS500 function module, which is described in detail in subsection 4.1.2.

The DB500 can only ever operate in one mode at a time. If more than one mode is set, the TRANS500 will decide which function the DB500 should perform in the following descending order of priority.

- Status display
- Screen display
- Cycle time display
- Limit control display
- Diagnostic display
- Display record memory
- Display text memory

2.9.1 Status display

Status displays facilitate rapid access to all PLC parameters.

Status displays are available for:

Operand	Format	Address area
Inputs	Byte	0 – 63
Extended inputs	Byte	0 – 63
Outputs	Byte	0 – 63
Extended outputs	Byte	0 – 63
Markers	Byte	0 – 255
Special markers	Byte	0 – 31
Timers		0 – 127
Counters		0 – 127
Data buffers	Words	0 – 511
Data words	Words	0 – 511
Data field	Words	0 – 24 k

Examples:

Inputs, outputs, markers, special markers

Inputs, outputs, markers and special markers are displayed in byte format. 32 bytes per screen can be simultaneously represented in binary, hex, decimal and ASCII form.

BOSCH “Diagnosis DB500” ZS1 >>S T A T U S – D I S P L A Y <<

>> INPUTS <<

Byte:	7 6 5 4 3 2 1 0	Hex:	Dec:	ASCII	Byte:	7 6 5 4 3 2 1 0	Hex:	Dec:	ASCII
I 0	0 0 0 0 0 0 0 0	0H	0D	–	I 16	0 0 0 0 0 0 0 0	0H	0D	–
I 1	0 0 0 0 0 0 0 0	0H	0D	–	I 17	0 0 0 0 0 0 0 0	0H	0D	–
I 2	0 0 0 0 0 0 0 0	0H	0D	–	I 18	0 0 0 0 0 0 0 0	0H	0D	–
I 3	1 0 0 0 0 0 0 0	80H	128D	–	I 19	0 1 0 0 0 0 0 1	41H	65D	A
I 4	1 0 0 0 0 1 1 0	86H	134D	–	I 20	0 1 0 0 1 0 0 0	48H	72D	H
I 5	1 0 0 0 0 1 0 1	85H	133D	–	I 21	0 1 0 1 0 0 0 0	50H	80D	P
I 6	0 0 0 0 1 0 0 1	9H	9D	–	I 22	0 1 0 1 1 0 0 0	58H	88D	X
I 7	0 0 0 1 0 0 0 1	11H	17D	–	I 23	0 1 1 1 0 0 0 0	70H	112D	p
I 8	0 0 0 1 0 0 1 0	12H	18D	–	I 24	0 1 1 1 0 1 0 0	74H	116D	t
I 9	0 0 0 1 0 0 0 1	11H	17D	–	I 25	0 0 1 0 1 0 0 0	28H	40D	(
I 10	0 0 0 1 0 1 1 0	16H	22D	–	I 26	0 0 1 1 0 0 0 0	30H	48D	0
I 11	0 0 0 0 0 0 0 0	0H	0D	–	I 27	0 0 1 1 1 1 0 1	3DH	61D	=
I 12	0 0 1 0 0 0 0 0	20H	32D		I 28	0 0 1 1 1 1 0 0	3CH	60D	<
I 13	0 0 0 0 0 0 0 0	0H	0D	–	I 29	0 0 0 0 0 0 0 0	0H	0D	–
I 14	0 0 0 0 0 0 0 0	0H	0D	–	I 30	0 0 0 0 0 0 0 0	0H	0D	–
I 15	**0 0 0 0 0 0 0 0**	0H	0D	–	I 31	**0 0 0 0 0 0 0 0**	0H	**0D**	–

Fig. 2–9 Status display of inputs

Counters

The counters are displayed as actual counter status in decimal values from 0 to 8191.

64 counter values can be displayed per screen in decimal format.

BOSCH "Diagnosis DB500" ZS1 >>S T A T U S – D I S P L A Y <<

>>COUNTER <<

Counter:	Counter:	Counter:	Counter:
C 0 = 8191	C 16 = 0	C 32 = 0	C 48 = 0
C 1 = 100	C 17 = 0	C 33 = 0	C 49 = 0
C 2 = 0	C 18 = 0	C 34 = 0	C 50 = 0
C 3 = 0	C 19 = 0	C 35 = 0	C 51 = 0
C 4 = 0	C 20 = 0	C 36 = 0	C 52 = 0
C 5 = 0	C 21 = 0	C 37 = 0	C 53 = 0
C 6 = 0	C 22 = 0	C 38 = 0	C 54 = 0
C 7 = 0	C 23 = 0	C 39 = 0	C 55 = 0
C 8 = 0	C 24 = 0	C 40 = 0	C 56 = 0
C 9 = 0	C 25 = 0	C 41 = 0	C 57 = 0
C 10 = 0	C 26 = 0	C 42 = 0	C 58 = 0
C 11 = 0	C 27 = 0	C 43 = 0	C 59 = 0
C 12 = 0	C 28 = 0	C 44 = 0	C 60 = 0
C 13 = 0	C 29 = 0	C 45 = 0	C 61 = 0
C 14 = 0	C 30 = 0	C 46 = 0	C 62 = 0
C 15 = 0	C 31 = 0	C 47 = 0	C 63 = 0

Fig. 2–10 Status display of counters

Timers

The timers are displayed as actual timer values in seconds. The time value and time base (10; 1; 0.1; 0.01 s) are also displayed.

32 timing circuits can be displayed per screen.

BOSCH "Diagnosis DB500" ZS1 >>S T A T U S – D I S P L A Y <<

>>TIMER <<

Timer:	Status:	Time:	Timebase:	Timer:	Status:	Time:	Timebase:
T 0	1	0.00 s	0.01	T 16	0	0.00 s	0.01
T 1	0 TH	0.00 s	0.01	T 17	0	0.00 s	0.01
T 2	0	2920.00 s	10.00	T 18	0	0.00 s	0.01
T 3	0	0.00 s	0.01	T 19	0	0.00 s	0.01
T 4	0	0.00 s	0.01	T 20	0	0.00 s	0.01
T 5	0	0.00 s	0.01	T 21	0	0.00 s	0.01
T 6	0	0.00 s	0.01	T 22	0	0.00 s	0.01
T 7	0	0.00 s	0.01	T 23	0	0.00 s	0.01
T 8	0	0.00 s	0.01	T 24	0	0.00 s	0.01
T 9	0	0.00 s	0.01	T 25	0	0.00 s	0.01
T 10	0	0.00 s	0.01	T 26	0	0.00 s	0.01
T 11	0	0.00 s	0.01	T 27	0	0.00 s	0.01
T 12	0	0.00 s	0.01	T 28	0	0.00 s	0.01
T 13	0	0.00 s	0.01	T 29	0	0.00 s	0.01
T 14	0	0.00 s	0.01	T 30	0	0.00 s	0.01
T 15	0	0.00 s	0.01	T 31	0	0.00 s	0.01

Fig. 2–11 Status display of timing circuits

Status: 0/1 corresponds to the result when the timer is polled (e.g.: A B T1)

TH means that the timer was stopped by the PLC command **"Timer Halt"**. The timer is stopped.

Time base: Programmed time–slot pattern in seconds.

Data modules / Data buffers / Data field

A maximum of 16 data words are displayed per screen.

Depending on the size of the data module, only the activated data words are displayed.

BOSCH "Diagnosis DB500" ZS1 >>S T A T U S – D I S P L A Y <<

>> DATA MODULE: 2 / DW: 0 – 30 <<

Word:	15 8	7 0	Hex:	Dec:	ASCII:
D 0	0 0 1 0 1 0 0 0	0 0 0 0 1 0 1 0	280AH	10250D	(.
D 2	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	2H	2D	..
D 4	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	1H	1D	..
D 6	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	2H	2D	..
D 8	0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0	2H	2D	..
D 10	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1	1H	1D	..
D 12	0 0 1 0 0 0 0 0	0 0 1 0 1 0 0 0	2028H	8232D	(
D 14	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..
D 16	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..
D 18	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..
D 20	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..
D 22	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..
D 24	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..
D 26	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..
D 28	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..
D 30	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0H	0D	..

> DIAGNOSIS – ACTIVE ! <

Fig. 2–12 Status display for data modules

Note ☞

If a diagnostic message comes up during a status display, this will be displayed in the "Alarm line". The messages are temporarily stored and automatically displayed in the manner described after changeover to diagnostic display.

2.9.2 Screen display

The screen display offers a wealth of possible representations in addition to the standard displays. A wide range of displays can be created by the user, PLC parameters (variables) can be monitored and plant operators, for example, can be guided/prompted.

A maximum of 511 screens can be supported. **The "0" screen is provided for clearing the screen.**

Screen display permits the simultaneous display of several screens. For example, screens can be compiled from a number of split screens or superimposed as the program dictates.

The use of variables allows the display of plant−specific data in a manner that is clear and concise.

Variables can be inputs, extended inputs, outputs, extended outputs, markers, special markers, timers, counters, data words, data buffers, data field, the date and time.

In the DB500 a maximum of 500 variables from data modules are possible. Up to 1000 variables per screen can be used for inputs, outputs and markers.

The display forms range from the simplest bit display to the complete text string. The display forms available for the different variables are described in more detail in "Variable displaying" in subsection 3.2.1.4.1.

The extended ASCII character set and the special DM characters are available.

The individual text and variable outputs on the screen are attributed. Each character on the screen can be assigned its own attribute (colour, flashing...etc.).

Note ☞ **The "Clear screen" command is taken as the background attribute for the complete screen, so long as an attribute was entered in the fist screen at address L1S1 (line 1, column 1).**

The screens are called by the TRANS500 function module. The DB500 can temporarily store up to 50 screen numbers, which means that a number of split screens can be combined before the actual screen display is enabled (screen selection).

This function also forms the basis of the **absolute display**, whereby a variable is represented by its address and not by its value. After the change-over from absolute to value display (or vice versa) the screen number memory is used to reconstruct the current screen.

Example:

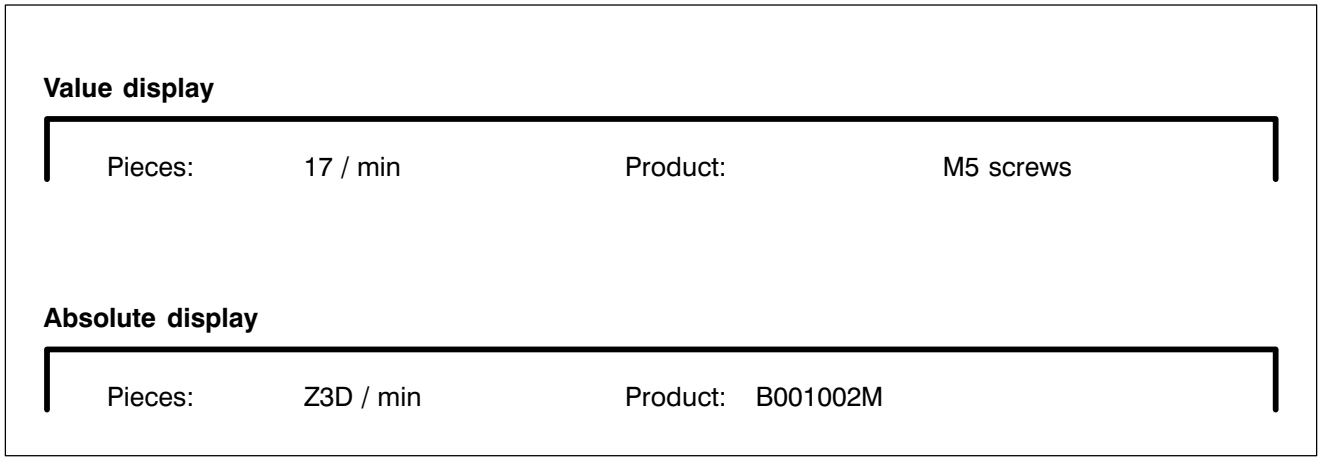


Fig. 2–13 Example of value and absolute displays

Note ☞

With many forms of representation the value display is smaller than the absolute display (B002003.2 <———> 1). The user should provide sufficient space in this instance, otherwise the absolute display will overwrite the text positions. No screen information is lost however, because the screen is reconstructed following the switchback.

Inserting attributes makes information clearer, e.g. by colour contrasting.

Attributes are modified directly when the screen is created as an integral part of the text information or by colour variables.

Softkeys

The **Softkey Screen** is a special kind of screen. Its only difference from standard screens is the fact that it is restricted to 2 lines. Lines 1 to 23 may not contain any information. As the DB500 contains the screens in condensed form, this is not a storage limitation.

Softkey screens can be activated for all standard tasks. They may contain variables and colour attributes.

Example of operating the record memory display using softkeys

Fig. 2–14 Operating the record memory display using softkeys

The texts of the screens can be stored in RAM or EPROM modules.

The screens are created on the programmer using the **DBG** utility program, then loaded into an EPROM module or directly into the module (RAM module).

2.9.3 Cycle time display

- Cycle in 1 s and 1/10 s
- The longest single cycle time is displayed inverted
- TC1 is ignored when processing the inverted display and has a special position, e.g. to display the total cycle time calculated by the user.

36 data words can be displayed as cycle times in seconds and tenths of a second per screen page.

The cycle time texts are up to 16 characters long.

Example:

BOSCH "Diagnosis DB500" ZS1 >>C Y C L E T I M E S <<

Station:	Value:		Station:	Value:	
TC 1 Text 1	2818.6	s	TC 19	0.0	s
TC 2 Text 2	0.1	s	TC 20	0.0	s
TC 3	0.1	s	TC 21	0.1	s
TC 4	0.1	s	TC 22	153.6	s
TC 5	0.2	s	TC 23	1.8	s
TC 6	0.1	s	TC 24	0.1	s
TC 7	0.3	s	TC 25	0.2	s
TC 8	0.1	s	TC 26	16.2	s
TC 9	0.4	s	TC 27	0.0	s
TC 10	0.1	s	TC 28	0.0	s
TC 11	0.5	s	TC 29	0.0	s
TC 12	0.1	s	TC 30	0.0	s
TC 13	0.6	s	TC 31	0.0	s
TC 14	0.2	s	TC 32	0.0	s
TC 15	**1239.3**	**s**	TC 33	0.0	s
TC 16	0.0	s	TC 34	0.0	s
TC 17	0.0	s	TC 35	0.0	s
TC 18	545.6	s	TC 36	0.0	s

> DIAGNOSIS – ACTIVE ! <

Fig. 2–15 Cycle time display

2.9.4 Limit control display

36 different limits can be monitored and displayed per screen page.
A 16−character−long text can be stored for every limit.

The display can be

- tabulated,
- as a bar chart, in which case the DB500 automatically calculates and displays the actual value as a % of the target value.

The display can be selected with or without an alert limit.

Examples: **Tabulated Display**

BOSCH "Diagnosis DB500" ZS1 >>LIMIT CONTROL <<

No.	Name:	Target:	Actual:	Warn:	No.	Name:	Target:	Actual:	Warn:
1	BIT 1234	64	8	56	19		0	0	0
2	TOOL 4	32	31	25	20		0	0	0
3	ELECTRODE 15	65	128	6	21		0	0	0
4		0	0	0	22		0	0	0
5		0	0	0	23		0	0	0
6		0	0	0	24		0	0	0
7		0	0	0	25		0	0	0
8		0	0	0	26		0	0	0
9		0	0	0	27		0	0	0
10		0	0	0	28		0	0	0
11		0	0	0	29		0	0	0
12		0	0	0	30		0	0	0
13		0	0	0	31		0	0	0
14		0	0	0	32		0	0	0
15		0	0	0	33		0	0	0
16		0	0	0	34		0	0	0
17		0	0	0	35		0	0	0
18		0	0	0	36		0	0	0

> DIAGNOSIS − ACTIVE <

Fig. 2−16 Tabulated display e.g. of toolchange monitoring

Display Response

- If the actual value is greater than the alert value, the alert value flashes.

- If the actual value is greater than the target value, both the target value and the alert values flash.

Display in Bar Chart Form

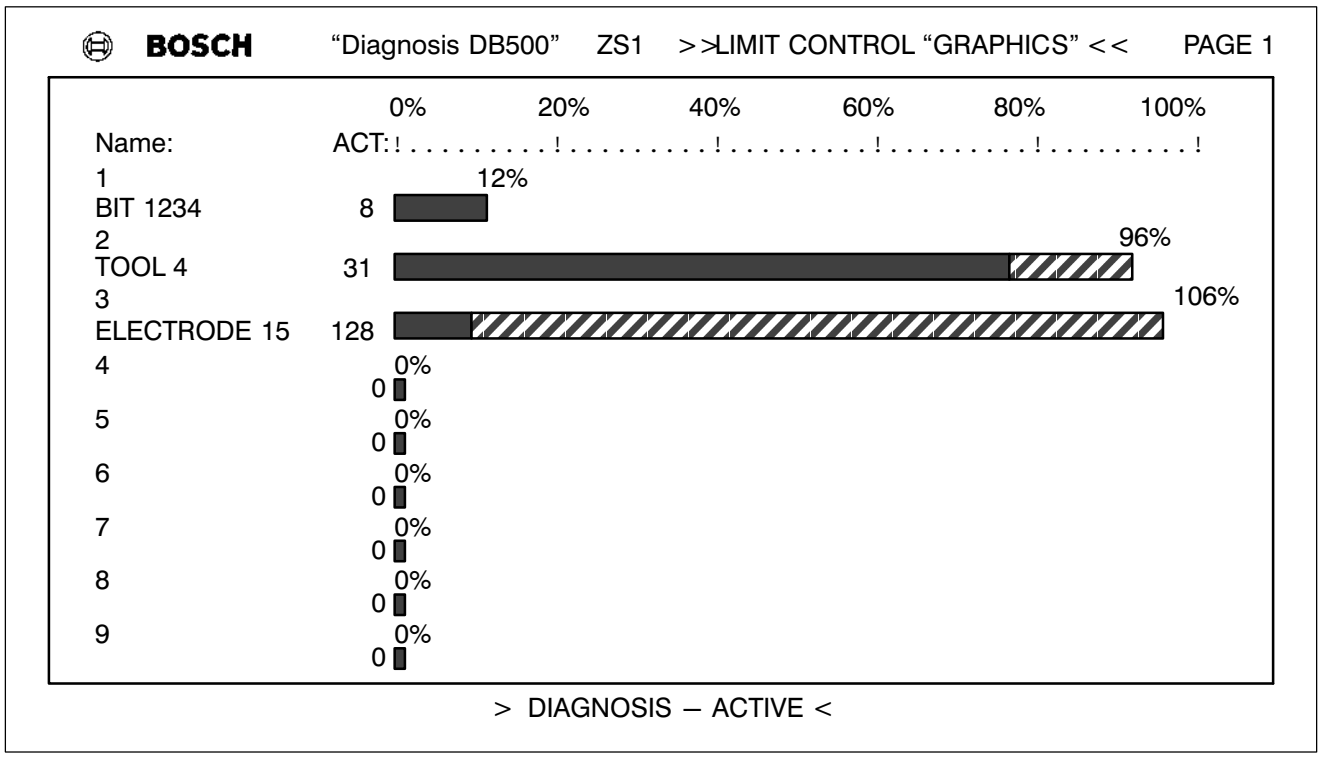


Fig. 2–17 Display in Bar Chart Form e.g. for toolchange monitoring

9 limits can be displayed per screen page. Page scrolling must be programmed in the PLC program via the TRANS500 function module.

Display Response

- If the actual value is greater than the alert value, then the bar is shown hatched from the alert limit upwards
- If the actual value is greater than the target value, the bar flashes.

2.9.5 Diagnostic display

A fundamental function of the DB500 is step cascade diagnosis. All steps which the DIAG500 function module identifies as faulty are decoded, logged, stored and – if necessary – displayed.

A total of 64 cascades each with 128 steps can be monitored per DB500. The DB500 performs a maximum of 64 step–on conditions (WSB).

If the control byte of the record memory is in the condition **"with first levels"**, then all identified step cascade errors are memorized with date and time.

The step–on conditions can also be logged on printer by the function **"and print"** independently of the storage function.

When an error status occurs the alarm line displays the message **"Diagnosis active!"** This message does not appear during the diagnostic display.

The DB500 can contain a 30–character–long text for every step–on condition and so it is easy to correlate error and machine. A text label of 16 characters is provided for each cascade.

Functions of the Diagnostic Display

The BEFA or WSB branches are displayed in 4 pages which each contain 16 step–on conditions with text and current status. The maximum number of possible pages and the current page can be identified by the page number in the uppermost screen line. Scrolling is done via the TRANS500 function module.

The status of the WSB is signalled in the form of a white field. A white field means the status is **"not true"**. This display is dynamic, i.e. it is always the current status which is displayed.

Cascade (“Kette”) Display

The display is on 4 pages each containing 16 cascades with mode, text, status and current step. The cascade page can be scrolled. When an error condition occurs the DB500 automatically displays the cascade page with the error cascade.

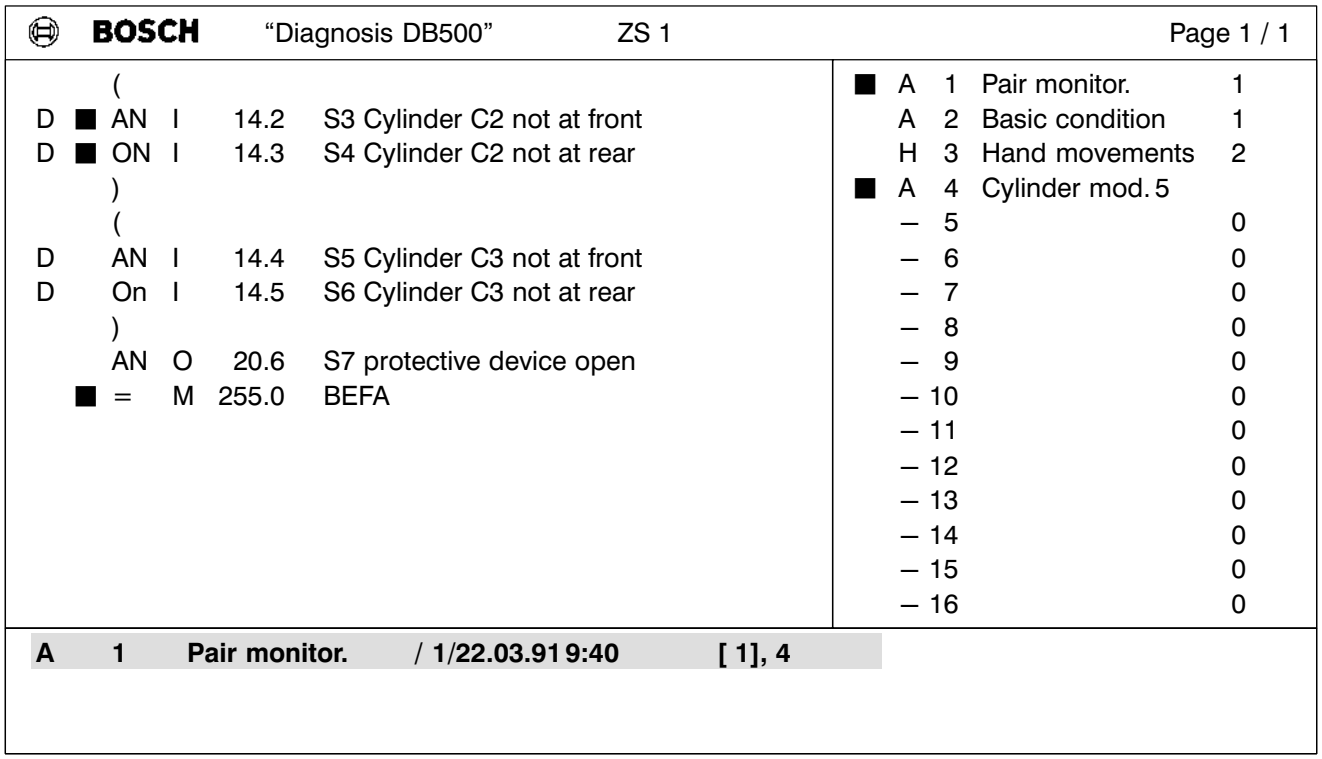


Fig. 2–18 Diagnostic screen

Functions via Operating Mode

The diagnostic screen offers information even when there is no error. It shows, for example, whether a cascade is in the “A”utomatic, “I”nching, or “M”anual mode or whether it is deactivated “–”.

A **"Cascade summary"** can be activated instead of the cascade text display. This summary shows the status of all cascades at a glance.

Example of a diagnostic screen with cascade summary

BOSCH " Diagnosis DB500" ZS1 Page 1 / 3

	(				1	17	33	49
	AN	I	6.5	Enable preselect	2	18	34	50
	A	I	2.4	Start manual	3	19	35	51
■	ON	I	2.4	Start auto	4	20	36	52
	)				5	21	37	53
	=	M	224.4		6	22	38	54
	A	I	3.1	Gripper up	7	23	39	55
	(				8	24	40	56
■	AN	I	3.2		9	25	41	57
	A	I	3.3		10	26	42	58
■	AN	I	3.4	Gripper down	11	27	43	59
	A	I	3.5		12	28	44	60
	AN	I	3.6		13	29	45	61
■	A	I	3.7		14	30	46	62
	O(	I	3.7		15	31	47	63
	AN	I	4.2		16	32	48	64

A 4 Cylinder mod. / 4 / 11.05.92 10:59 [4], 1

Fig. 2–19 Example of a Diagnostic Screen with Cascade Summary

The inverted numbers are faulty cascades.

First level Cascade

In the event of an error the faulty cascade, together with cascade number, mode, error step, date and time is superimposed in the lower half of the screen.

Cascade Error

Similarly to the cascade summary, a brief message is given as to whether other cascades are in an error condition. The bracketed cascade is the first level cascade.

DESI Errors

The I/O signals transferred to the DESI system are represented in detail.

Example of a diagnostic screen with DESI display rail and cascade page

BOSCH " Diagnosis DB500" ZS1 Page 1 / 3

	(				■	A	1	Pair monitoring 112
	AN	I	6.5	Enable preselect	■	H	2	Basic condition 3
	A	I	2.4	Start manual		A	3	Hand movement 1
■	ON	I	2.4	Start auto	■	A	4	Cylinder mod. 5
	)					–	5	0
	=	M	224.4			–	6	0
	A	I	3.1	Gripper up		–	7	0
	(					–	8	0
■	AN	I	3.2			–	9	0
	A	I	3.3			–	10	0
■	AN	I	3.4	Gripper down		–	11	0
	A	I	3.5			–	12	0
	AN	I	3.6			–	13	0
■	A	I	3.7			–	14	0
	O(	I	3.7			–	15	0
	AN	I	4.2			–	16	0

A 2 Basic condition / 112 / 11.05.92 10:59 [1], 3

Busm. / RING / Btn / Inp / Outp / Module / – Error| Desi–Stop| Toolset – Error / Modif.

Fig. 2–20 Example of a Diagnostic Screen with DESI Display Rail

The DESI rail is only displayed in the event of an error condition. The cause of the error is inverted and only displayed in capital letters. In Fig. 2–20, the cause of error is “RING”, which appears inverted in the display.

Example of a diagnostic screen with exact bit display

BOSCH "Diagnosis DB500" ZS 1 Page 1 / 1

```
      (
D ■ AN  I   14.2  S3 Cylinder C2 not at front          BT2  X1.X23  (1, 0C)
D ■ ON  I   14.3  S4 Cylinder C2 not at rear           BT2  X1.X24  (1, 0C)
      )
      (
D   AN  I   14.4  S5 Cylinder C3 not at front Cable br. BT2  X2.X21  (1, 0C)
D   On  I   14.5  S6 Cylinder C3 not at rear           BT2  X2.X22  (1, 0C)
      )
    AN  O   20.6  S7 Protective device open
  ■ =   M  255.0  BEFA
```

A 1 Pair monitor./ **1/22.03.91 9:40 [1], 4**

Busm. / Ring / Btn / **INP** / Outp / Module **Error** ¦ Desi−Stop ¦ Toolset Error / Modif.

Fig. 2−21 DESI Messages

Manual diagnosis

In addition to automatic diagnosis, every step of every cascade can also be checked for status. This is performed by the TRANS500 module (see subsection 4.1.2.3).

BOSCH "Diagnosis DB500" ZS 1 Page 1 / 1

■	AN	I	14.2	S3 Cylinder C2 not at front	■	A 1	Pair monitor. 1
■	AN	I	14.4	S5 Cylinder C3 not at front		**A 2**	**Basic condition 1**
	AN	O	20.6	S7 Protective device open		H 3	Hand movements 2
	=	M	255.1	WSB	■	A 4	Cylinder mod. 5
						– 5	0
						– 6	0
						– 7	0
						– 8	0
						– 9	0
						– 10	0
						– 11	0
						– 12	0
						– 13	0
						– 14	0
						– 15	0
						– 16	0

Manual diagnosis / Automatic branch
Cascade A2 Basic condition No: 1

Fig. 2–22 Diagnostic Screen with Cursor Supeimposition

2.9.5.1 Ladder Diagram

The ladder diagram (LD) representation is a useful addition to diagnostic display. Here relationships are shown more clearly than in the instruction list (IL).

The change from ladder diagram to diagnostic screen is performed by simple toggling (see 4.1.2.3).

In the ladder diagram, the representation is organized by branch and not by page. One branch comprises all the step−on conditions of an allocation.

Functions in the LD

Paging

Paging is done branchwise forwards and backwards.

Scrolling

Where there are more ladders than can be displayed on one screen page, the ladder diagram can be scrolled up and down. Arrows indicate where the LD overhangs the screen (see Fig. 2−23 Branch 2 / 2).

Symbols

The LD representation uses the following symbols:

——] [——	AND / OR
——] / [——	AND NOT / OR NOT
——( )——	equal to
——— |	Connection, line

Identifying text

The absolute identifying text is above the ladder.

Conditions “True / not true”

True conditions are displayed inverted.

—[/]— or ——] [——

A maximum of 7 chaining operations and one allocation are possible per line.

Example of a Ladder Diagram Display

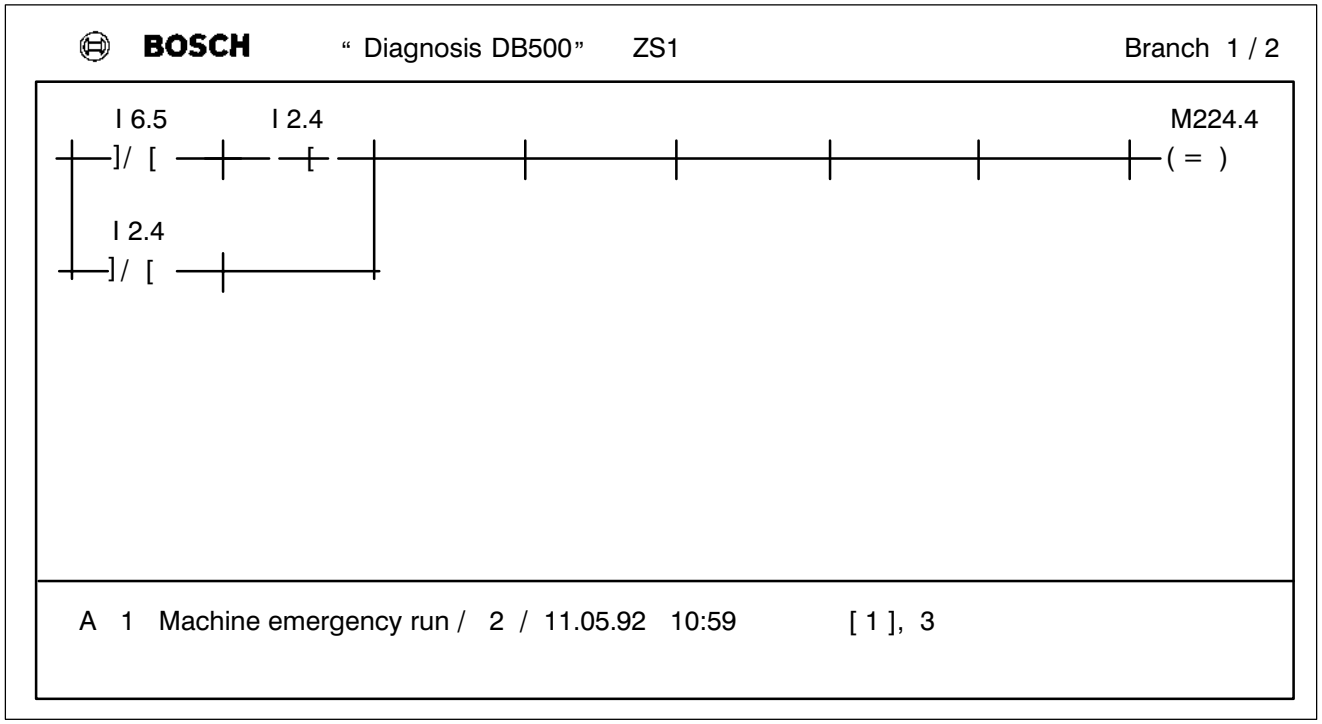


Fig. 2–23 Example of a ladder diagram display without true conditions – Branch 1 / 2

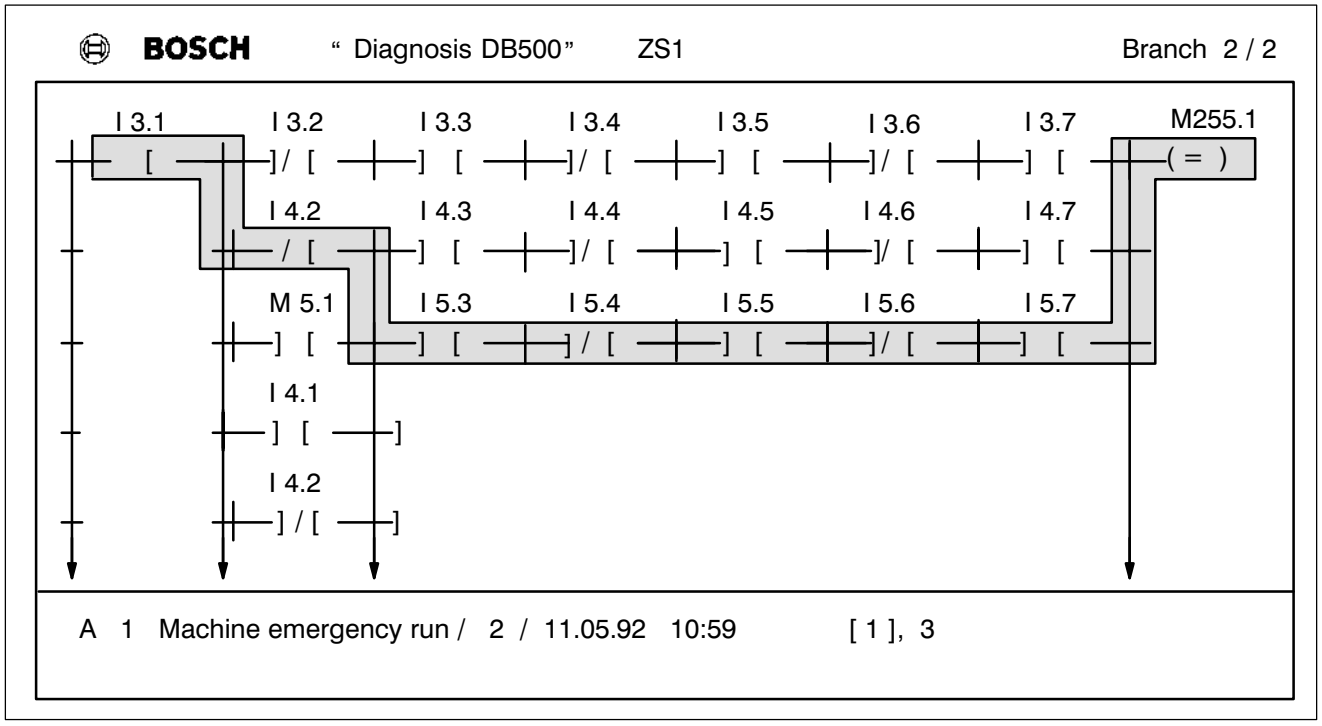


Fig. 2–23 Example of a ladder diagram display with true conditions – Branch 2 / 2

2.9.6 Record Memory

All diagnostic, DESI and customer messages can be filed in the record memory for later evaluation.

The entry can be individually decided for each message, and all entries are stored with the date and time of occurrence.

The memory operates as a loop memory, i.e. it is always possible to file a current message.

Once the memory's capacity is exhausted, the **"oldest"** message is overwritten in favour of the **"latest"** message.

The storage capacity of the record memory, depending on the type of message, is:

approx. 4300 diagnostic messages depending on the number of step–on conditions (I / O / SM / M / T / C – Points) per first level.

or

approx. 2300 customer messages

or

approx. 2000 DESI messages.

Any combination of the above types of message is possible.

Only first level messages can be entered in the record memory.

The record memory can be printed out or displayed on screen, thereby allowing the messages to be evaluated.

The record memory is buffered. It is controlled or cleared via the PLC or the terminal.

2.9.6.1 Display of the Record Memory

The standard display of the record memory contains a menu. By specifying certain codes the operator is able to display the contents of the record memory.

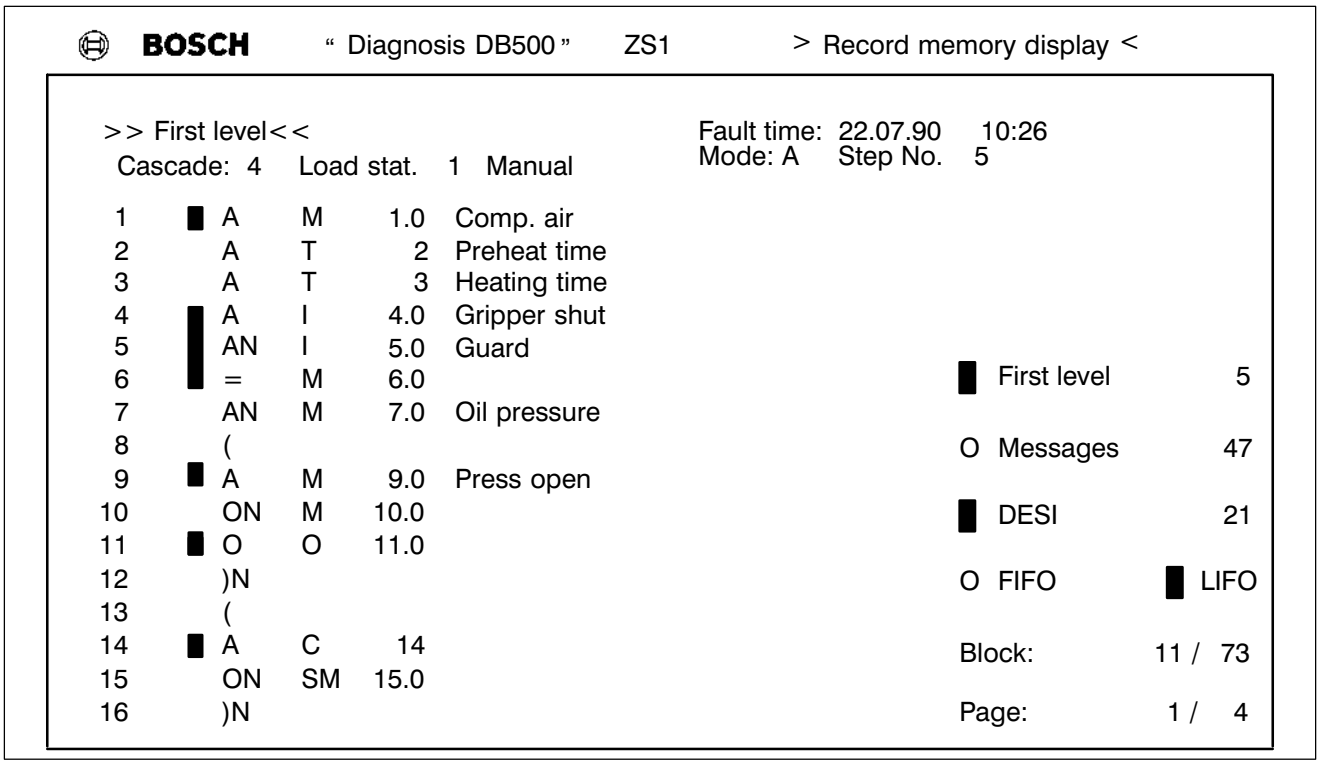


Fig. 2–24 Example of a Display of the Record Memory

Selection possibilities

Selection

First levels and/or DESI and/or customer messages can be individually called or turned off, thus making a particular message easy to locate.

LIFO / FIFO

This sets the output direction **(Last In – First Out, First In – First Out)**. Depending on this setting, the **first** or **last** message is run when the record display is selected, and the direction for paging is determined.

Paging

Paging by message or by page is controlled by the TRANS500 function module.

Block

Block selection causes the current position within the memory to be displayed. For example, 22 / 163 means the current message 22 out of a total of 163 message blocks.

Page

The page display is activated when more information is available than can be displayed on screen. When paging in FIFO the next page is automatically selected.

2.9.7 Text Memory

The customer−specific texts are filed in plug−in RAM or EPROM memory modules. These texts are:

- Designation texts of operands – max. 30 characters
- Cascade texts – max. 16 characters
- Cycle time display texts – max. 16 characters
- Texts for limit display – max. 16 characters
- Customer−specific messages – max. 64 characters
- Screens and softkeys

These texts are created on the programming unit. The created screens and the text can then be loaded direct by the programming unit into an EPROM module or into the DB500 (RAM module). The RAM module is buffered centrally by the CL500.

2.9.7.1 Display of text memory

The text memory display is called via the PLC (see Description MADAP500, P.−Nr.4142) or terminal with one of these commands

KC=CONTENT or INH or **KC=DIRECTORY or DIR**

The module size and memory layout (RAM or EPROM) can be displayed.

The number in the text display assigned to the individual text types stands for the number of texts stored for that text type.

When called via **Modem** after a timer supervision or after the ESCAPE key is activated, the screen is enabled again for the function selected by the PLC.

Example:

BOSCH " Diagnosis DB500 " ZS1 > Text memory – Display <

RAM–Module	128 kB
Inputs	15
Outputs	2
Markers	8
Special markers	3
Timers	2
Counters	1
Cascades	10
Messages	7
Cycle times	2
Limits	4
Screens	12
Total	28 kB

Fig. 2–25 Example of a Text Memory Display

Module sizes

The following modules can be used as text memories:

Memory module	Capacity DB500
RAM 32 kW	64 kB
RAM 64 kW	128 kB
RAM 128 kW	256 kB
EPROM 32 kW	64 kB
EPROM 64 kW	128 kB
EPROM 128 kW	256 kB

2.10 Documentation Printout

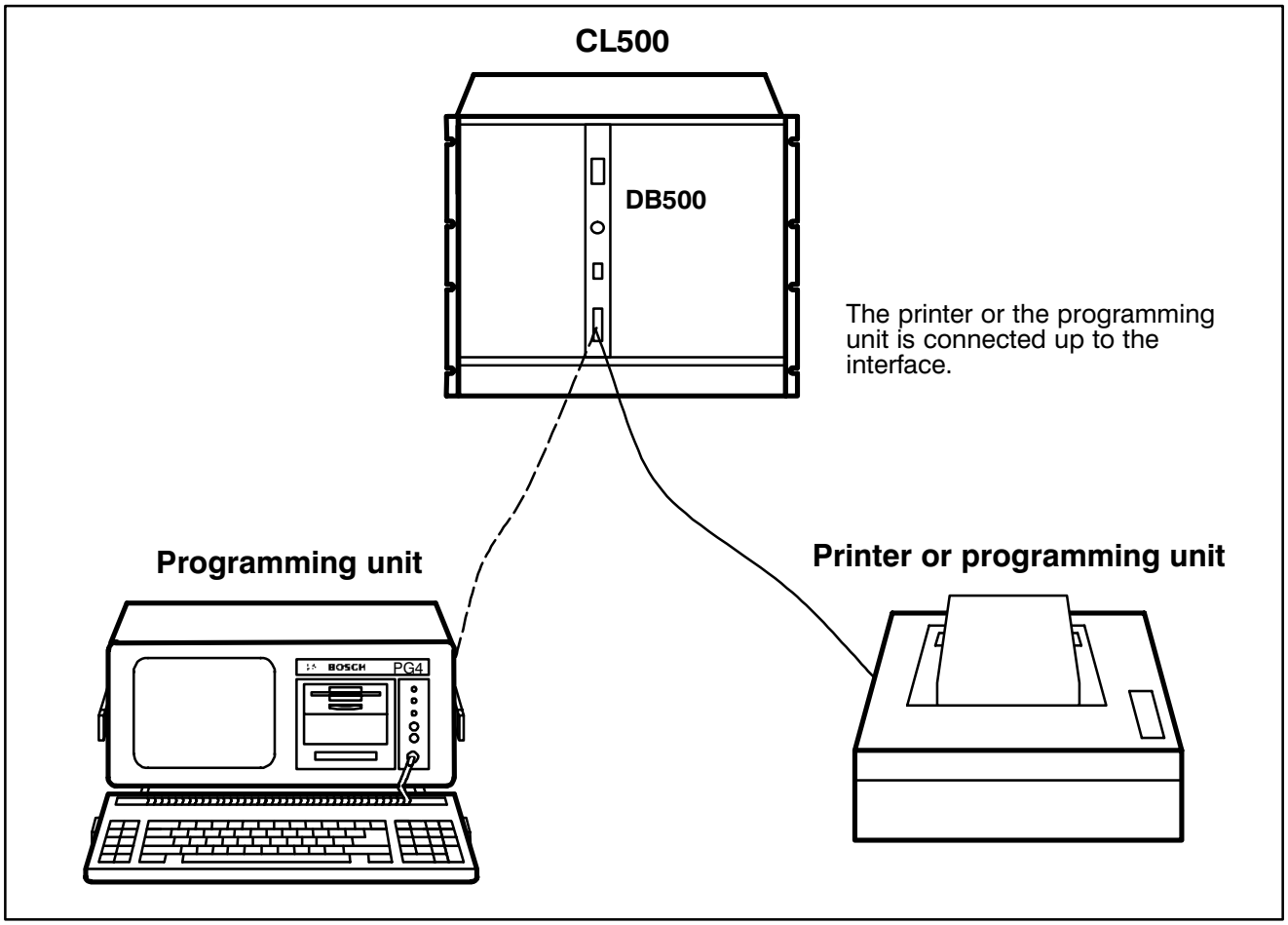


Fig. 2–26 Example of connection of programmer and printer

The following printouts are possible:

- Record memory with diagnostic messages, customer–specific messages and DESI messages
- Print screen
- General messages

Note ☞ **Remember that the printer must be IBM–compatible in order to be able to print the full character set, with the exception of special characters.**

2.10.1 Record Printout

The record printout is call via PLC parameters (see Description MADAP500, P.−Nr.4142) or terminal with the command

KC=PRINT or P.

Both calls (PLC or terminal) are of equal value.

The record printout can also be initiated from the terminal with different parameters from those set in the PLC (see page 2−12).

All other requests are disabled while a printout is running. Other messages can be entered in the memory during the record memory output. The memory cannot be cleared until after the output with the command

KC=ERASE or ERA or PLC.

The printout must be told the output direction (LIFO / FIFO), the type of message and the new−value output.

LIFO / **FIFO**

This defines the output direction.

Message type

The printout can be restricted to DESI messages, for example, by turning off the first level and customer messages (PLC parameters or terminal inquiry).

New values

With each print operation the record memory register the printout of a “specific message”. This message is allocated an identifier. At the next printout − and depending on the parameter setting or terminal inquiry − either **“new”** or **“all”** messages are output.

Parameters

When the record output starts, the current date, printout time and parameter selection are output.

Changes

If the parameters are changed while printout is active then the modified parameters will apply as from the current item. The parameter selection is output every time the parameters are changed.

Abort

The printout is aborted by changing the parameter selection of the PLC to **"without"** first level + **"without"** DESI + **"without"** customer messages.

If the printout was initiated from the terminal, it can be aborted by entering **ESCAPE**.

Example of a Record Printout:

```
********* Record printout                          Tuesday, 24.07.90      18:10:45 *******

Parameters for record printout:

        First level     :       YES
        Messages        :       YES
        DESI            :       NO
        Printout        :       complete
        Direction       :       LIFO


        >> MESSAGE  coming  !  <<      Message No.:   4        24.07.90     08:00

        Plant start-up after service


        >> First level <<                    Fault time:          24.07.90     09:24

        Cascade:  2  Basic condition          Mode A               Step No.  3

              1     A      SM    2.3
              2     AN     M     2.3      Low oil pr.
              3     A      I    55.5      Comp. air
              4     AN     SM    2.3
              5     =      M   255.3
              6     A      M   255.0      BEFA
              7     (
              8     AN     O     2.3
              9     O      O     2.4
             10     )N
             11     =      M   222.2
             12     =      M   222.2
             13     A      O     2.3
             14     =      M   255.1      WSB

        >> MESSAGE  going  !  <<      Message No:   6        24.07.90     09:34

        Plant start-up after service

********* E N D   record printout                 Tuesday, 24.07.90      18:10:49 *******
```

Fig. 2-27 Example of a Record Printout

2.10.2 Print Screen Function

The current screen can be printed out either by using the appropriate parameter of the TRANS500 function module (see Description MADAP500, P.−Nr.4142) or with the command

KC=HARDCOPY or H.

2.10.3 Alarm Line

An alarm is activated in all displays in addition to the normal information. The plant operator is immediately notified of conditions such as a breakdown in the data exchange between PLC and the DB500.

The alarm line can also be used to output customer−specific messages, which are described below in greater detail.

Where a number of alarm messages occur simultaneously, they are output in order of priority, with the high−priority message being displayed.

Internal messages take priority over customer−specific messages.

Summary of Internal Messages

Message	Cause
Diagnostics not active !	Currently no data exchange between the DB500 and the PLC (e.g. when the text memory is being loaded)
ZS transmitting no data !	No response by ZS to data inquiry. ZS may be in STOP
No data exchange with FM possible !	FM and DB500 not compatible e.g. wrong versions.

Message	Cause
Memory module !?	No functional module identified.
EPROM !	Loading not possible. An attempt was made to load texts into an EPROM module.
DIAGNOSIS – ACTIVE !	Diagnostic message has arrived.
DESI – Message active !	DESI reporting error.
OVERFLOW! Diagnostic messages	More than 64 WSB have been transferred to the DB500.
Data overflow text memory !	More texts than storage capacity.
Interface not ready !	No character could be trans-ferred inside the monitoring time during attempt to output via interface.
Variable not displayable !	A variable requested by the screen display or the softkey rail cannot be activated by the PLC.

Customer–specific messages

Up to 511 customer–specific messages can be stored in the DB500.

These messages are activated by the PLC and are either entered in the record memory with date and time, output to printer or displayed on screen in the alarm line, as required.

The output/display is controlled by the user describing a word parameter in the TRANS500 function module.

When displayed in the alarm line, the insertion of screen attributes is similar to the screen display.

Example

^ G This is an alarm text

It means: G – Hexcode 47
4 – Background colour yellow
7 – Foreground colour white (see Fig. 3–10)

Internal alarm messages (no connection with ZS) have priority over the customer–specific messages, and are displayed immediately.

The user has no control over this order of priority. Once the alarm message is cleared the customer message is automatically activated.

It is always the last message with the Trans parameter **"Screen"** which is visible in the alarm line. Message No.: **"0"** clears the customer mes-sage from the screen (see Description MADAP500 P.–Nr.4142).

The messages **"coming** / **going"** indicate whether a message is occurring or clearing.

Example of customer–specific messages:

Record memory display

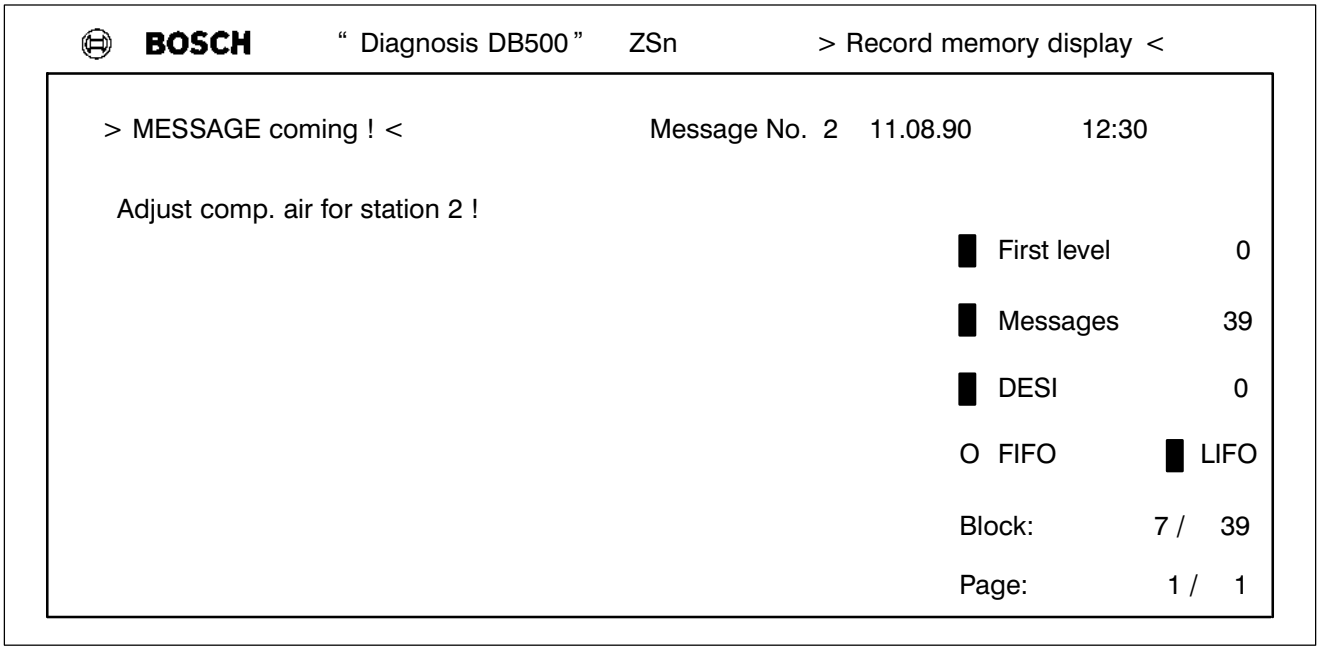


Fig. 2–28 Example of a customer–speicific message in the record memory

Alarm Line

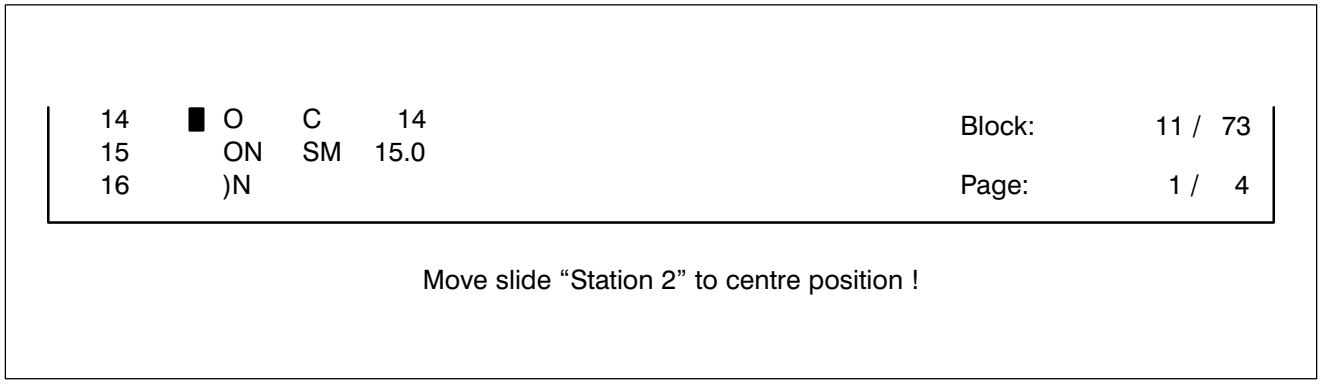


Fig. 2–29 Example of a customer–specific message in the alarm line

Record printout

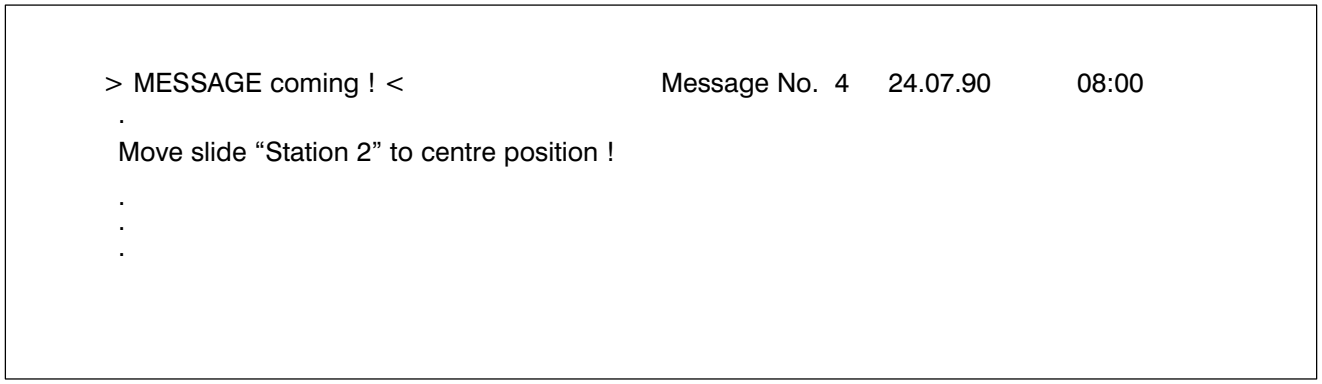


Fig. 2–30 Example of a record printout with a customer message

Printout of the message (and print)

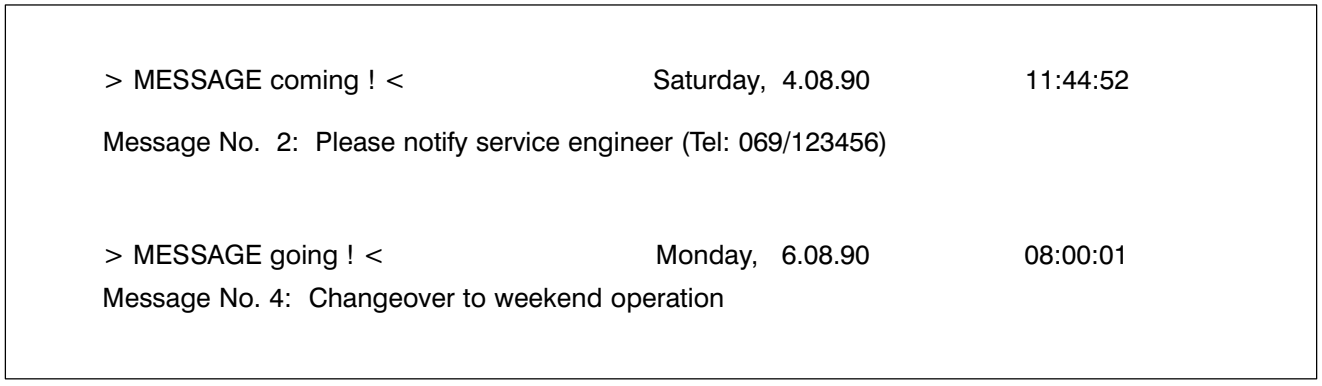
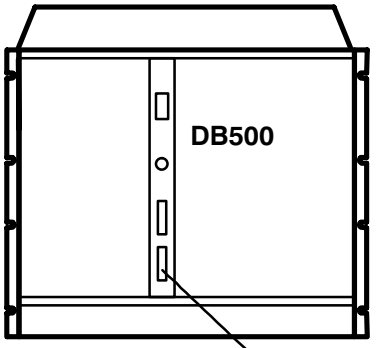


Fig. 2–31 Example of a printout of the message

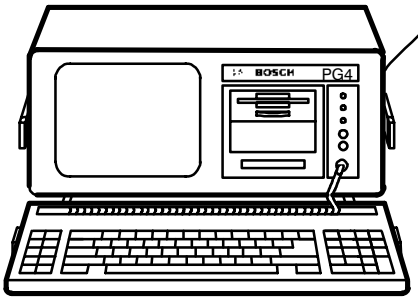
3 Entries (Communication between DB500 and Programming Unit)

3.1 Summary of Possible Entry Functions



X31 – Interface V.24/20mA

CL500



Programming unit

Possible entry functions:

- Edit
- Create graphics

} the same function keys

- Modem operation
- Load
- EPROM–Programming
- End
- Enter symbol file

Fig. 3–1 Possible entries into the DB500

3.1.1 Installing and Calling the DBG Utilities

The DBG Utilities include the following files:

- INSTALL.BAT
- DBG.BAT
- DB500.EXE
- DB5—100.OVL
- PG4MESD.TXT
- PG4TEXTD.TXT

Installing the Software

Calling the file “INSTALL.BAT” from the diskette creates the path C:\DBG on hard disk and copies all required files from the diskette to this path. When this procedure is complete the path is validated.

Calling the Program Package

The complete program package is called by entering

DBG < Enter >

The DB500 main menu is then displayed on screen together with the underlaid program modules.

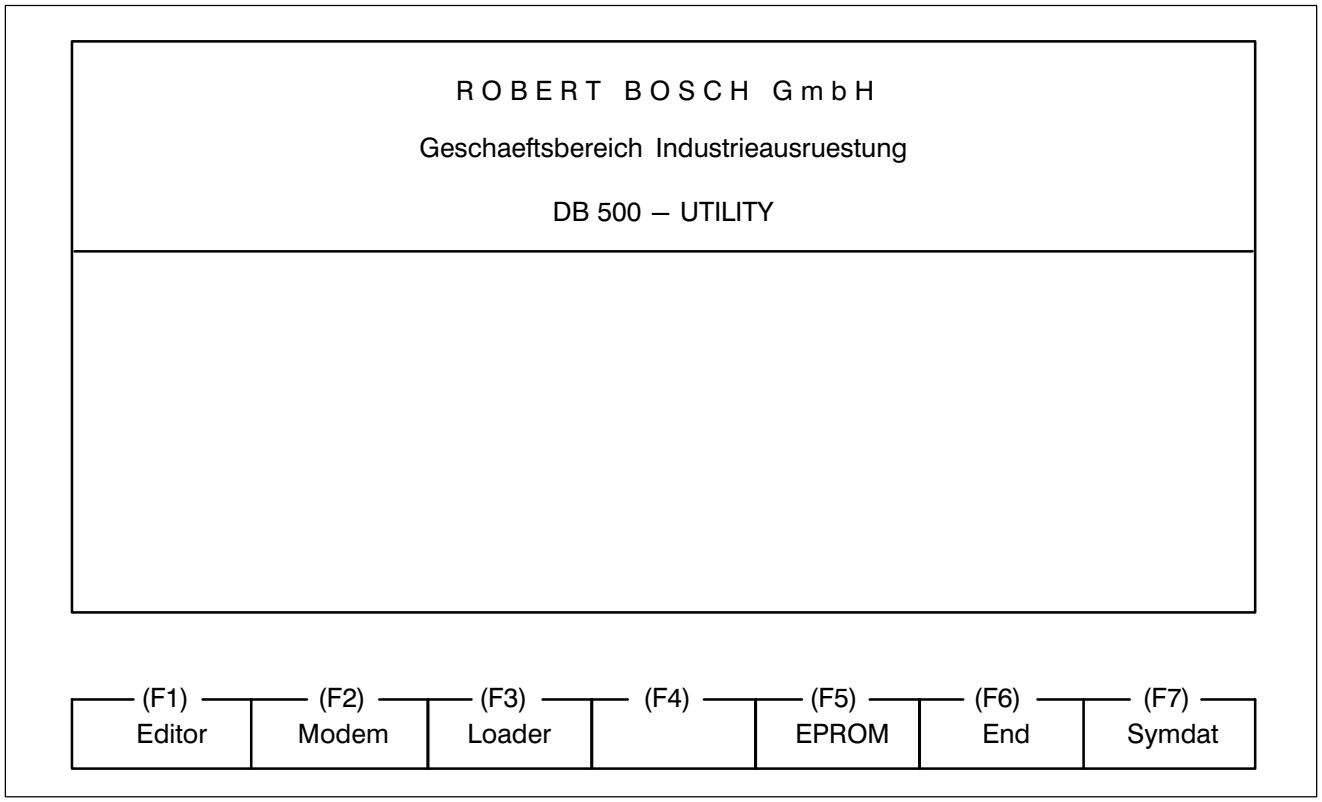


Fig. 3–2 DBG Main Menu

Exiting the program package

Pressing the function key **"F6"** (End) in the main menu exits the program package "DBG Utilities" without further prompts.

3.2 Program modules

The main menu contains a program module bar which offers the following program modules:

- Editor
- Modem
- Loader
- EPROM
- End
- Symdat(symbol file)

The individual program modules have the following tasks:

3.2.1 Editor

The **Editor** is used to edit documentation, batch files, graphics etc.

The program makes the distinction between text and graphics automatically, i.e. it recognizes whether a line is a text or a graphics line. The **Editor** works with a maximum line length of 80 characters

The **Editor** is called with the function key

and is exited with function key

Note ☞ **The Editor does not monitor the text lengths and screen sizes defined for the DB500. Corrections and “Trimming” to the displayable sizes are performed by the program module “Loader”. During this operation no character is lost in the editable file.**

Screen after calling the Editor with Function Key < F1 >

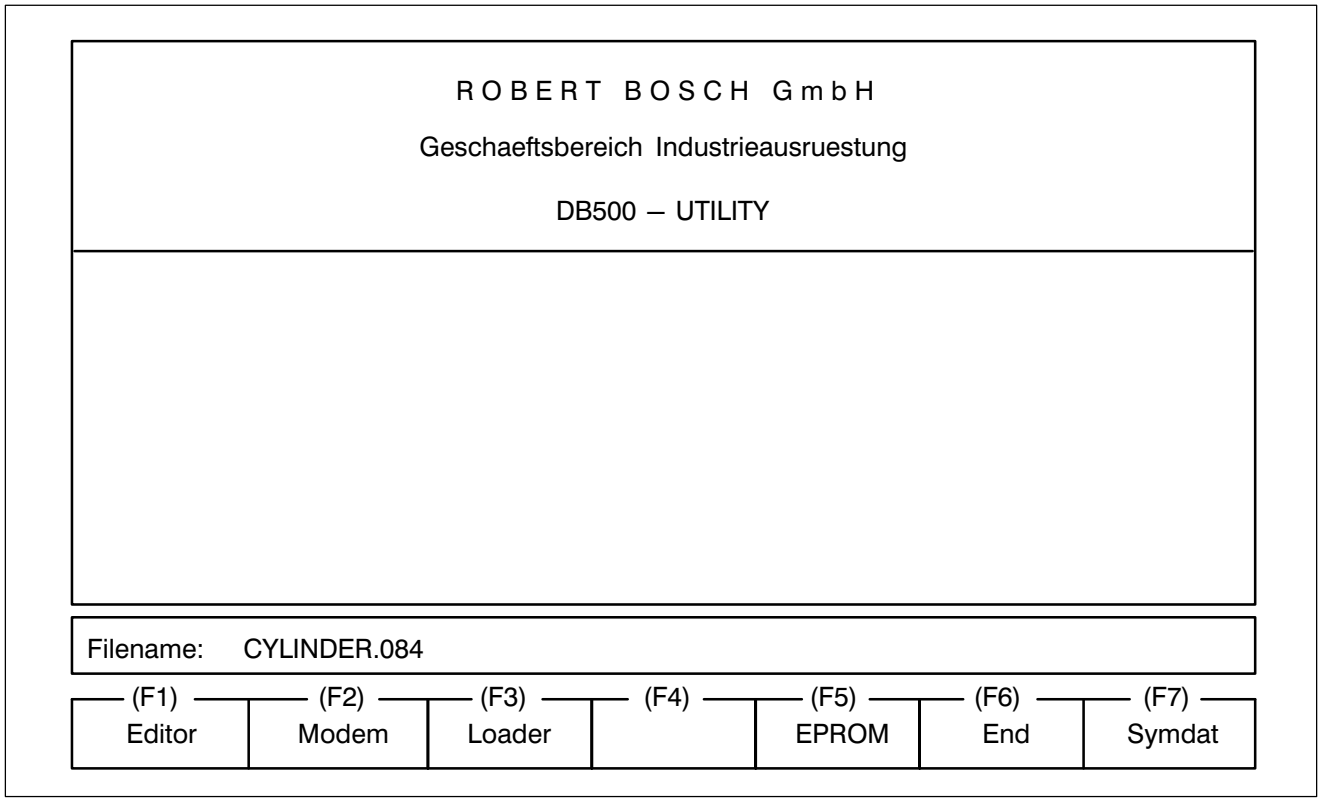


Fig. 3–3 Screen after calling Editor

After keying the desired filename, e.g. **CYLINDER.084**, followed by < Enter > the file is loaded and displayed on the monitor.

Screen after Filename Entry

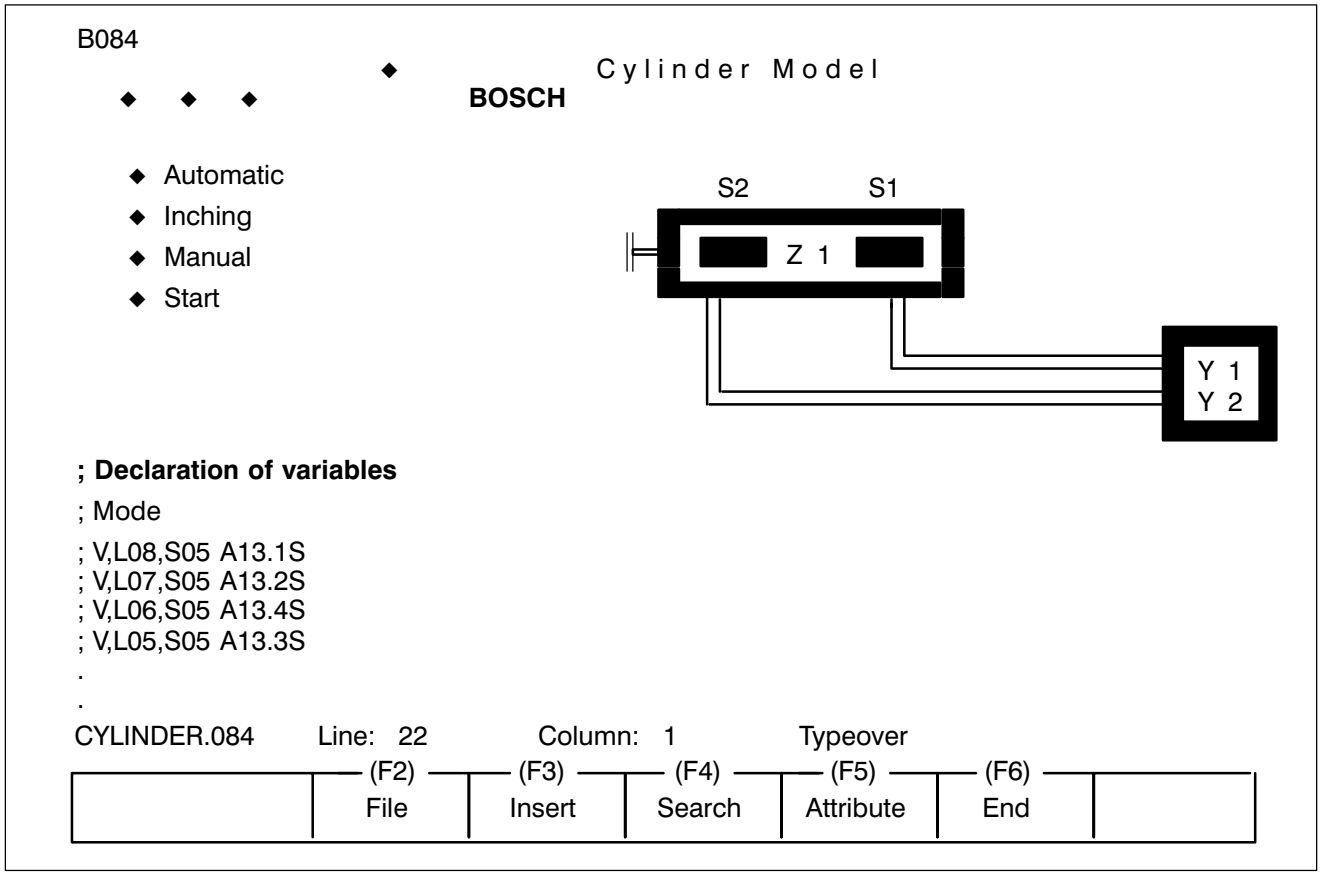


Fig. 3–4 Screen after Filename Entry

Working with **Editor** is achieved using the following functions:

1. Use of function key **< F3 >** for inserting.

   This key generates a variable. If the cursor position is already occupied by a special character or a variable then repeated pressing of the **< F3 >** key will toggle between **Special characters/Varaible line** and the **associated position** on the screen.

2. Editing in this screen can **only** be done by typeover. The **< INS >** key does not toggle between **"Insert" and "Typeover"** but between the normal screen and the attribute screen.

3. Variable lines are edited using the function key **< F3 >**.

4. To distinguish between unoccupied screen positions (character **00**) and **blanks**, the **blank** is replaced by a placeholder (¬). This placeholder reverts to a **blank** on **loading** to the DB500.

Each of these characters (placeholders) occupies one memory byte.

In the **Editor** the placeholder for variables is a lozenge (◆) and for special characters a laughing face.

Note ☞

Screens are programmed as they are meant to appear later on the monitor.

The screen numbers (1 — 512) must be inserted at the left-hand margin (column 1) as the start of screen identifier.

The screen lines are then numbered starting with line 1 and ending at line 25.

The variables are defined as a list following the last screen line.

When converting the screens of the DB600 and DB301 on the DB500, unoccupied positions on the screen will be filled by the placeholder (¬). Each of these characters occupies one byte in the text and graphic memory. Depending on application, these characters can be cleared and this storage space recovered.

3.2.1.1 Editable Text Types

Text types	No. of characters, max. text length	No. of texts
I0.0B – I63.7B	30	512
O0.0B – O63.7B	30	512
M0.0B – M255.7B	30	2048
SM0.0B – SM31.7B	30	256
T0 – T127	30	128
C0 – C127	30	128
Cascade texts	16	K01 – K64
Message texts	64	S001 – S511
Limit texts	16	W01 – W36
Cycle time texts	16	U01 – U36

I/O/M/SM/T/C/– Designation texts

The designation or criteria texts are used to verbally describe the untrue conditions defined by the “DIAGNOSIS” and displayed on the screen in standard diagnostic form.

The texts can be entered in the field provided with up to 30 characters per operand.

Cascade texts K01.....K64

The cascade texts provide an abbreviated description of the step cascades in the standard diagnostic screen.

The cascade number (1–64) is entered in the field provided for the address, with an identifier starting with the letter “K” (K01, K02, ...K64). The order is random. The texts can be entered with up to 16 characters per cascade.

Message texts S001.....S511

The message texts are operating messages independent from the diagnostics and can be recorded on printer or terminal.

The message number (1 — 511) is entered in the field provided for the address, with an identifier starting with th letter “S” (S001, S002, ...S511). The order of the message texts is random. Up to 64 characters can be entered in the message text field.

Cycle time texts U01.....U36

The cycle time texts are used to designate stations in the cycle time screen.
The station number (1 — 36) is entered in the field provided for the address with an identifier starting with the letter “U” (U01, U02, ...U36). The order of cycle time texts is random. Up to 16 characters can be entered in the cycle time text field.

Limit texts W01.....W36

The limit texts designate the stations in the limit control screen.

The station numbers (1 — 36) are entered in the field provided for the address with an identifier starting with the letter “W” (W01, W02, ...W36). The order of the limit texts is random.

Up to 16 characters can be entered in the field provided for the limit text.

Notes ☞

The edited line lengths are controlled, which means it is not possible to edit more than the acceptable number of characters.

The acceptability of a line is also checked for the correctness of the first character. Comment lines must begin with a semicolon and can be freely edited.

3.2.1.2 Command Groups in Editor

The **Editor** program module contains the following command groups:

- File
- Insert
- Search
- Attribute

"File" command group

Calling the command group **"File"** with the function key

opens a window containing the option **"Save"**.

Using function key

the user can now save the file without exiting the **Editor**.

“Save” Option

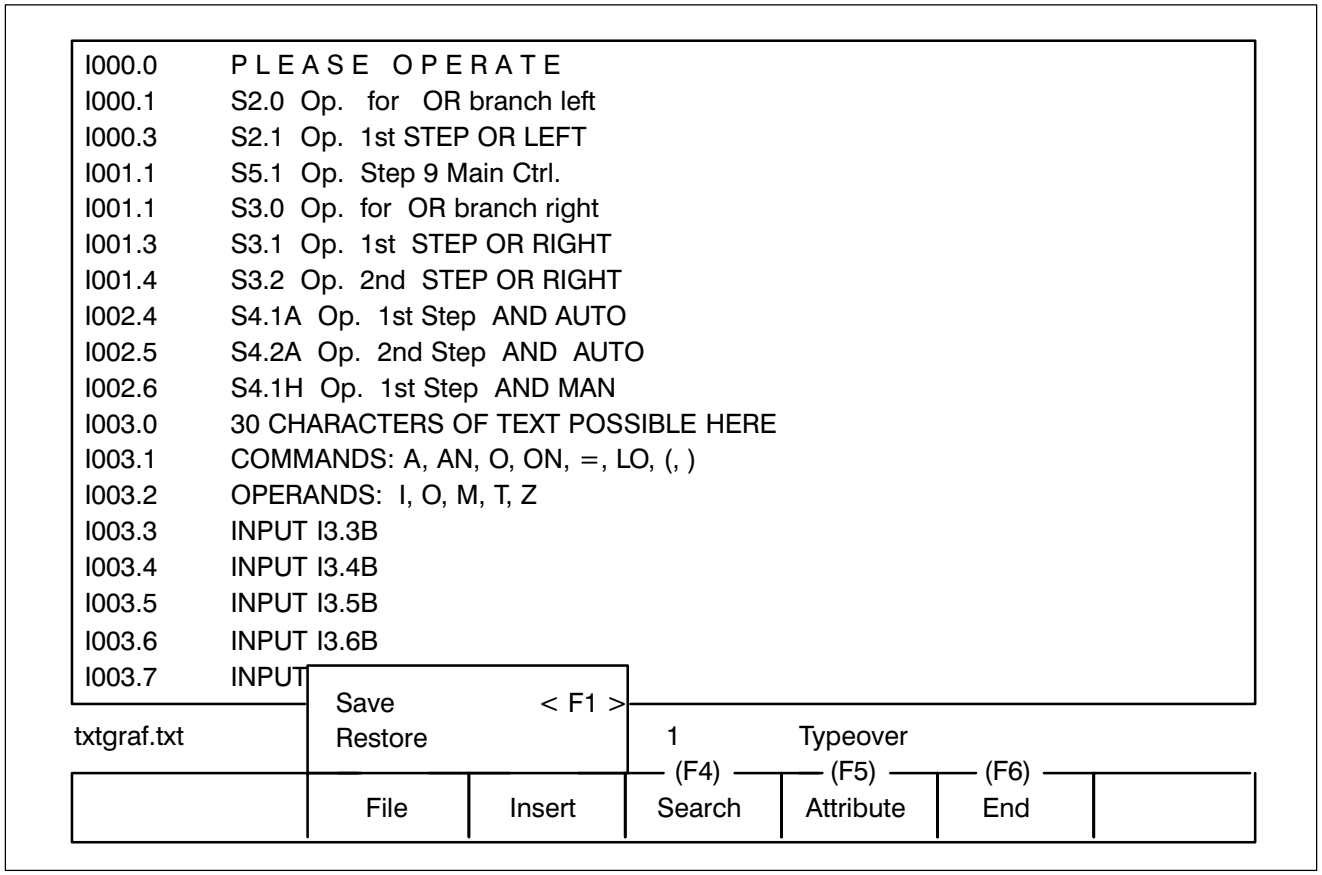


Fig. 3–5 Screen after calling command group “File”

“Insert” command group

When the command group **“Insert”** is called with the function key

the software differentiates between the following possibilities:

1. Outside a screen (incl. lines Bxxx and BE) a new screen with the screen number B??? is inserted.

2. Inside a screen, either a variable line is generated (if there was no variable at the corresponding screen line) or the screen jumps to and fro between the variable line and the corresponding screen line. In this way, it is possible to find the right variable line for every variable and vice versa.

On screen the place of the variable is occupied by a placeholder in the form of a lozenge (◆), which can neither be overwritten nor deleted by the **Backspace or DEL** keys.

A variable can only be removed using **Shift—DEL** which deletes it from both the screen and the variable line.

Note ☞ **The edited variables are not checked for acceptability. Unacceptable variables are ignored by the DB500.**

"Insert" Option

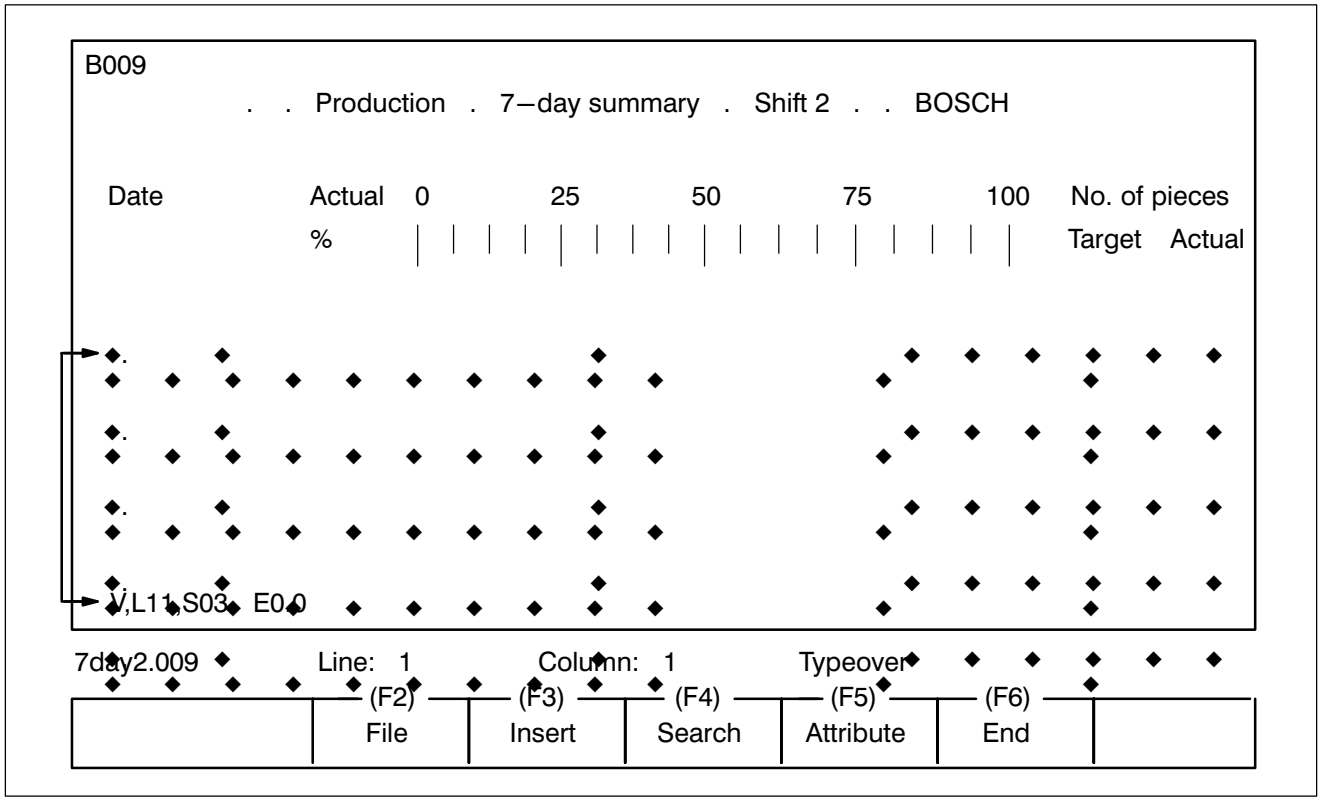


Fig. 3—6 Possible screen after calling the command group "Insert"

"Search" Command Group

Calling the command group **"File"** with the function key

opens a window containing the following options:

- Data header – Key F1 (Jump to start)
- File end – Key F2 (Jump to end)
- Character sequence – Key F3 (see below)
- Next (repeat) – Key F4 (repeat operation)

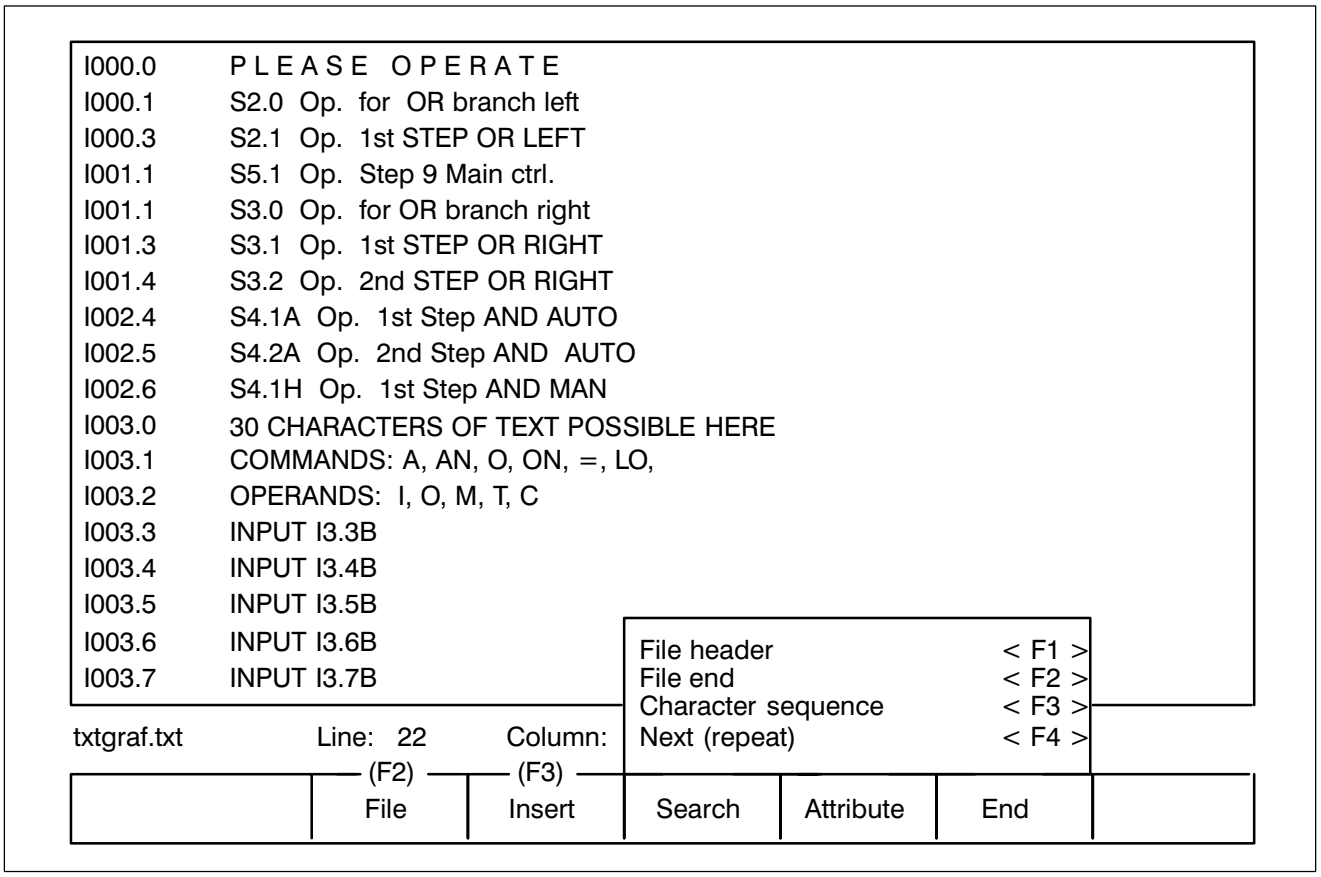


Fig. 3–7 Screen after calling command group "Search"

If the

key is used to select a sequence of character, the desired sequence must be entered first. When the entry is terminated the user is offered three options (see Fig. 3–8).

Searching a character sequence

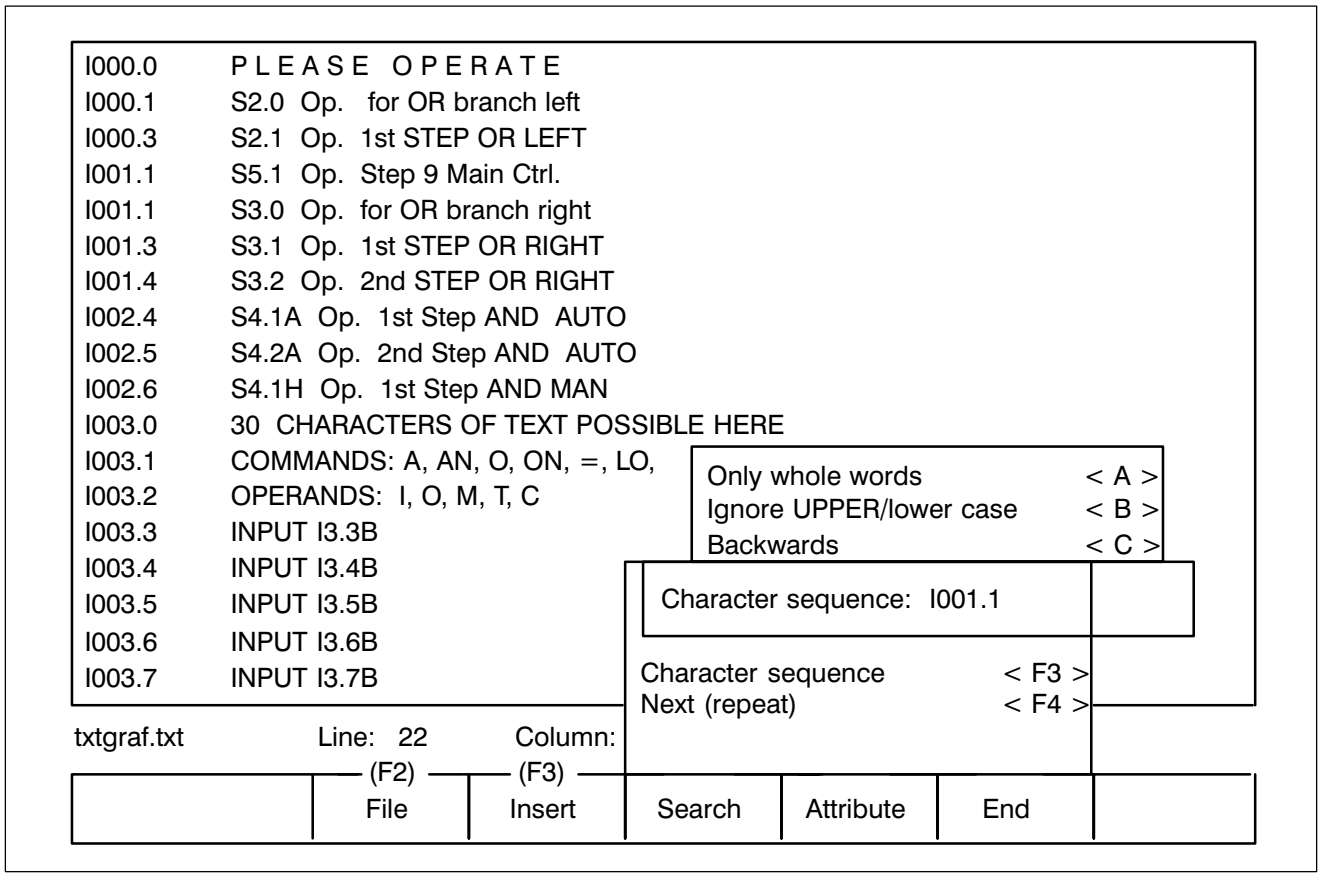


Fig. 3–8 Screen after calling the "Character sequence" option with key F3

“Attribute” Command Group

The DB500 can be operated in both mono and colour mode.

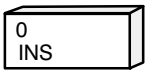
In the colour mode each screen line can be assigned

- a foreground colour,
- a background colour,
- flash or not flash,
- underscore, yes/no

The following attributes are possible in mono mode:

- flashing,
- inverse,
- underscore,
- double bright.

The attributes are edited either by switching to the attribute screen from the Editor with key



and entering the appropriate identifier, or direct with function key

INS switches from the **Edit screen** condition to the corresponding **Attribute screen**, where the attributes can be edited at will or from the Attribute screen to the Edit screen.

When editing the attributes with function key

the desired attribute qualities are selected in three windows (see Fig. 3–9).

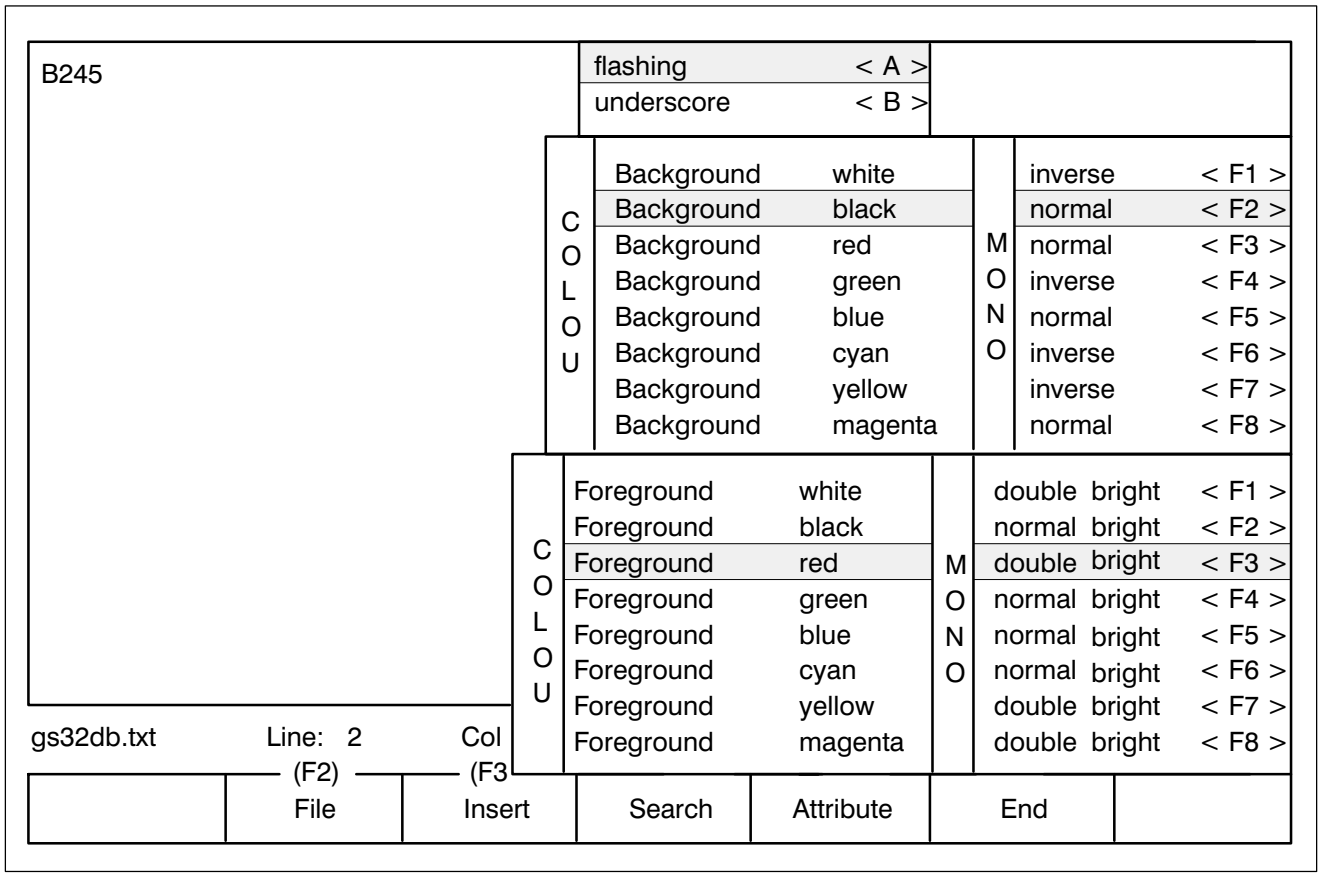


Fig. 3–9 Screen after calling the "Attribute" command group with the key F5

The values revealed when the windows are opened correspond to the underlaid attribute.

The first two windows (foreground and background) are each split and describe the screen statement for a DB500 in colour and mono modes. The attributes can be edited using

on either the Edit screen or the Attribute screen.

Attributes of the DB500

Hex Number	Meaning	Colour	Monochrome
80	Flash		Flash
08	Underscore		Underscore
00	Foreground colour	black	normal bright
01	Foreground colour	red	double bright
02	Foreground colour	green	normal bright
03	Foreground colour	yellow	double bright
04	Foreground colour	blue	normal bright
05	Foreground colour	magenta	double bright
06	Foreground colour	cyan	normal bright
07	Foreground colour	white	double bright
00	Background colour	black	normal
10	Background colour	red	normal
20	Background colour	green	inverse
30	Background colour	yellow	inverse
40	Background colour	blue	normal
50	Background colour	magenta	normal
60	Background colour	cyan	inverse
70	Background colour	white	inverse

The total attribute is achieved by addition.

Examples:	Foreground colour red	01	Flash	80
	Background colour white	70	Underscore	08
	Flashing	80		
	Total attribute	F1		88

Fig. 3–10 Attributes of the DB500

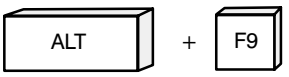
Note ☞ **The attribute < FF > (white on white, flashing and underscored) is changed to attribute < FM > (yellow on white, flashing and underscored) when loading. No such change to the screen image occurs on mono.**

3.2.1.3 Editing Special Characters and Special Key Configurations

The existing **Editor** facilitates the entry of characters from the complete ASCII character set from **00H to FFH.**

The entry of graphic characters is activated with key

or with the key combination



.

The difference between these two calls is that calling with **< F9 >** generates a window with a list of all **ASCII characters** on the left−hand side of the screen; with **< ALT > + < F9 >** this is on the right−hand side. The window can be scrolled with an inverted element until the desired character is selected.
Pressing the **< Return >** key supplies the value of this character as a return parameter.

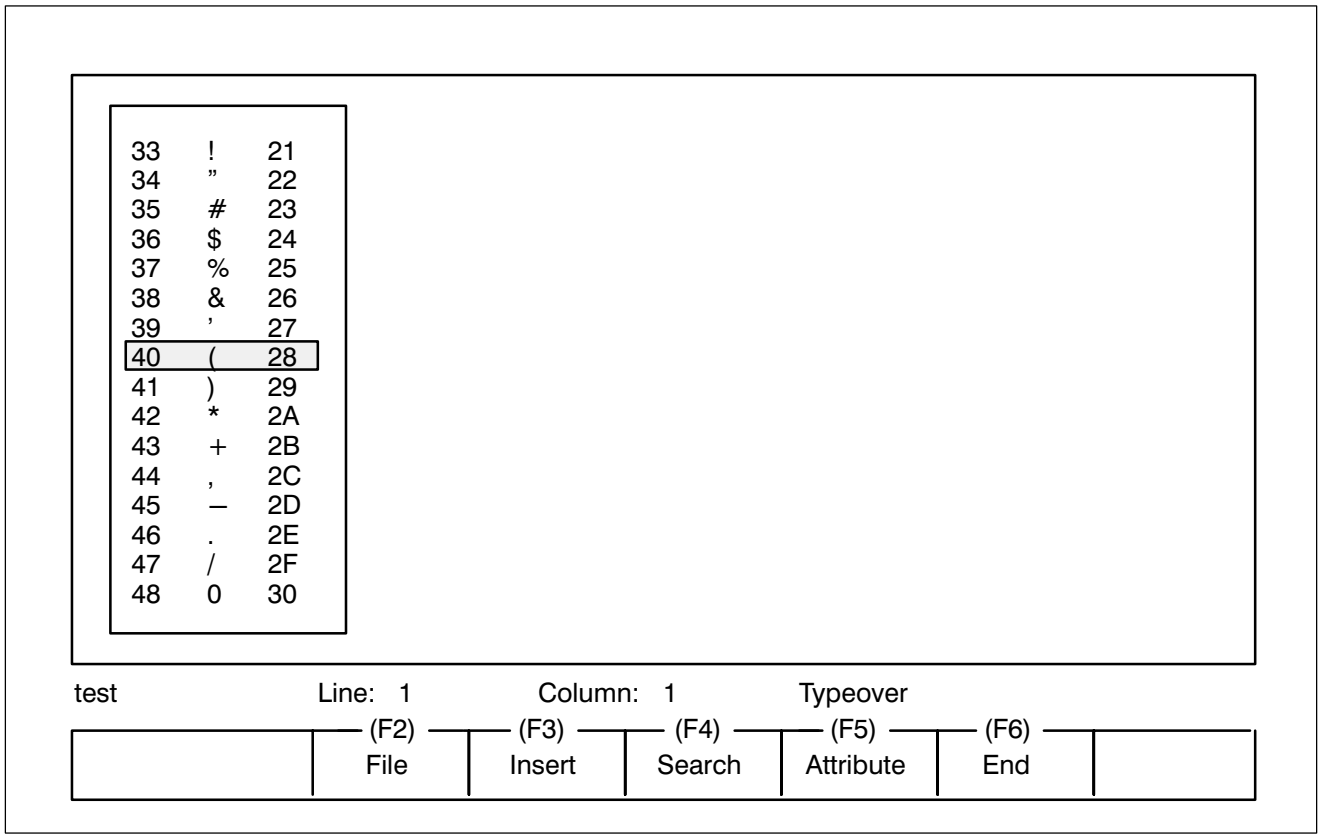


Fig. 3–11 Example of the ASCII Character Set called with key < **F9** >

Pressing the **< F7 >** key displays the last edited special characters on screen without the need to open a window.

This does not apply to characters less than **20H** or the characters **00H – 1FH and 7FH – ACH.** As these characters in the DB500 differ from the ASCII list, a direct entry of these characters into the screen is disabled. They can only be displayed in hex in the form

S, Lxx, Syy ZZ .

xx = line number
yy = column number
ZZ = character number.

The display is automatically selected by the software, i.e. if a character less than 20H is edited a special character line is generated and a placeholder (a laughing face) is inserted in the screen, which can neither be over-written nor deleted by means of the **Backspace key or DEL**.

A special character can only be deleted using **< Shift—Del >** which removes it from both the screen and the special character line.

Use the edit keys to reach the individual characters in the list:

Pressing this key shifts the current field up a line.
Display scrolls when window edge is reached.

Pressing this key shifts the current field down a line.
Display scrolls when window edge is reached.

Jump to first character in list

Jump to last character in list

Window is paged back a page

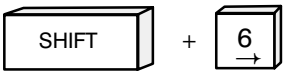
Window is paged forward a page

Pages window back 5 pages

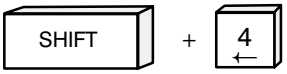
Pages window forward 5 pages

The character **FF** in the special character editor has a special function. The character **FF** stands for a “transparent” place in the screen, i.e. where the information of any screens previously called is visible. If the user is in the attribute screen, however, **FF** is displayed as an attribute in the form of a special character line.

Specal Key Configurations



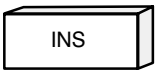
Jump to the end of the chosen line. On screen – to the end of the screen line or attribute line.



Jump to the beginning of the chosen line. On screen – to the beginning of the screen line or attribute line.

Cursor function only, no editing

Cursor is moved 8 places left.



Toggles between **"Insert"**

and **"Typeover"**. Within a screen, toggles between **Screen** and **Attribute**.

Converting screens created with Diagnostic Modules DB301 and DB600

Screens created with DB301/DB600 software are automatically identified by the Editor when the file is retrieved, and are converted to DB500 screens in the mono mode.

3.2.1.4 Using the Editor to Create Graphics

Connection of Possible Communications Units to the DB500

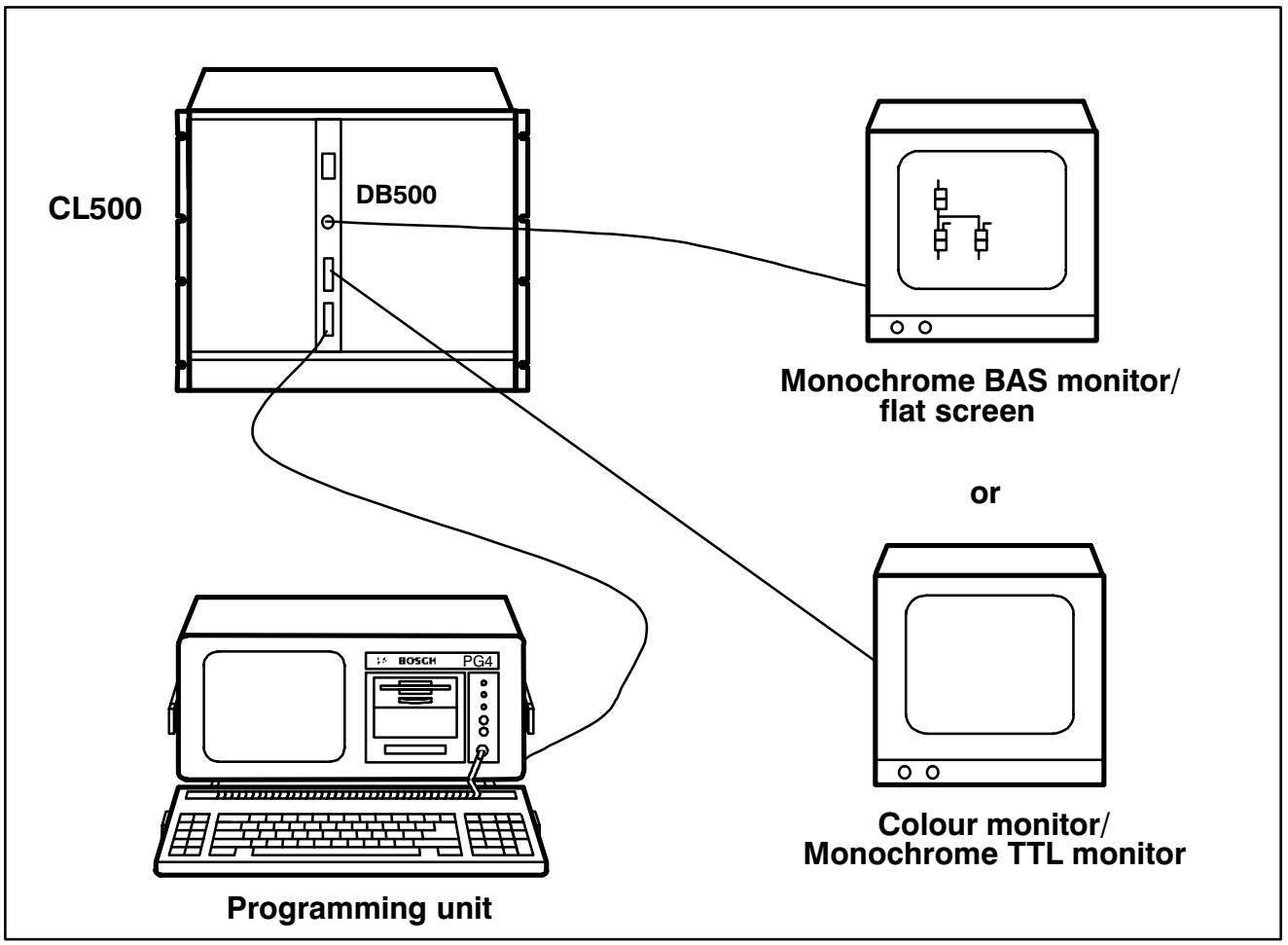


Fig. 3–12 Connecting the equipment used for graphics creation

In addition to the standard display of diagnostic and status messages, the user is also able to create plant–specific graphics and screens as an additional aid to see–at–a–glance diagnostics and plant monitoring. 511 out of the 512 screens that can be created can be addressed. The first screen with No. 000 is not editable and is used to clear the screen display.

The DB500 has a special character set for graphic symbols (see Fig. 3–13).

Each screen can contain texts and variables as well as graphics symbols. The following variables are available:

- I/O/M/SM/EI/EO
- Data buffers
- Data field
- Data words
- Timers
- Counters
- Date
- Time
- Colour variable

The manner in which individual variables are displayed is shown in Fig. 3−14 “Displaying of Variables”.

When working with the Editor the programming unit need not necessarily be connected up to the DB500. The texts and screens are generated after calling the program module **Editor** using function key

The screens are entered in the **Insert** command group.

The sequence of screens in a file can be random.

A beginning− or end−of−screen identifier does not have to be edited − this is carried out automatically when a new screen number is entered.

Note ☞ **The screen no. 000 cannot be edited. It is used for clearing the screen display.**

Entering Variables

Variables are defined as a list following the last screen line and are automatically entered from column 1 according to the following definition in command group **Insert** of the **Editor** program module:

V, Lnn, Snn (Operand) with

line numbers (Lnn) 1 to 25 and column numbers (Snn) 1 to 80 are possible.

Screen programming

It is advisable always to set the screen number as a multiple of 10 of the line numbering , as this makes it easier to count the permitted lines (25 per screen).

3.2.1.4.1 Character Set of the DB500

Dec	➧	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
⬙	Hex-Dec	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0			Blank Space	0	@	P	‘	p					L		α	≡
1	1			!	1	A	Q	a	q							ß	±
2	2	⌒		″	2	B	R	b	r							Γ	≥
3	3			#	3	C	S	c	s							π	≤
4	4	▽		$	4	D	T	d	t							Σ	⌠
5	5			%	5	E	U	e	u							σ	⌡
6	6		↑	&	6	F	V	f	v							μ	÷
7	7		↓	′	7	G	W	g	w							τ	≈
8	8		←	(	8	H	X	h	x							Φ	○
9	9		→	)	9	I	Y	i	y							Θ	●
10	A			*	:	J	Z	j	z							Ω	•
11	B			+	;	K		k								δ	√
12	C			,	<	L	\	l								∞	n
13	D			—	=	M	{	m				i				∅	²
14	E			•	>	N	^	n	~			<<				⟨	
15	F			/	?	O	_	o				>>				⟧	(FF)

Fig. 3–13 DB500 Character Set

3.2.1.4.2 Displaying of Variables

Variable Range	Identifiers German	Identifiers English	Format	Display type/ Number range	Example: Call	Remarks
Inputs 0.0 – 63.7	E	I	Bit	Figure (0/1) or Symbols □ / ■ Text up to 30 charac.	I4.2	Bit
Outputs 0.0 – 63.7	A	O			I4.2S	Bit, Symbol
Markers 0.0 – 255.7	M	M			I4.2Y	Bit, Symbol
Special markers 0.0 – 31.7	SM	SM			I4.2T	Bit, Text
EI 0.0 – 63.7	EO	EI			IEI4.2	Bit
EO 0.0 – 63.7	AZ	EO				
Inputs 0 – 63	E	I	Byte	Decimal (D) 0 – 255	I4B	Byte, Decimal
Outputs 0 – 63	A	O		Hexadecimal (H) 0 – FF	I4BD	Byte, Decimal
Markers 0 – 255	M	M		Binary (B) 00000000 – 11111111	I4BH	Byte, Hex
EI 0 – 63	EZ	EI		Binary symbol (S) □□□□□□□□ - ■■■■■■■■	I4BB	Byte, Binary
EO 0 – 63	AZ	EO		ASCII – set #1	I4BA	Byte, ASCII character
				Status (Z) Null = 0 # Null =1	I4BS	Byte, Binary symbol
				Status symbol (Y) Zero = □ # Zero = ■	I4BZ	Byte, Status
					I4BY	Byte, Status symbol
Inputs 0 – 62	E	I	Word	Decimal (D) 0 – 65535	I4W	Word, Decimal
Outputs 0 – 62	A	O		Hexadecimal (H) 0 – FFFF	I4WD	Word, Decimal
Markers 0 – 254	M	M		Binary (B) 00000000 00000000 – 11111111 11111111	I4WH	Word, Hex
EI 0 – 62	EZ	EI		Binary symbol (S) □□□□□□□□ □□□□□□□□ - ■■■■■■■■ ■■■■■■■■	I4WB	Word, Binary
EO 0 – 62	AZ	EO		ASCII – set #2	I4WS	Word, Binary symbol
				Status (Z) Null = 0 # Null =1	I4WZ	Word, Status
				Status symbol (Y) Zero = □ # Zero = ■	I4WA	Word, ASCII character
					I4WY	Word, Status symbol
					I4WF	Word, Colour variable

Fig. 3–14 Displaying Variables – Sheet 1

Variable Range	Identifiers German	Identifiers English	Format	Display type/ Number range	Example: Call	Remarks
Data module 0 – 255	B	B	Bit	Figure (0/1) or Symbols □ / ■	B001004.2	Bit
					B001004.2S	Bit, Symbol
Data buffer 0 – 510	P	P			B001004.2Y	Bit, Symbol
Data field 0 – 24 k	DF	DF			B001004.2	Bit
Data module 0 – 255	B	B	Byte	Decimal (D) 0 – 255		
				Hexadecimal (H) 0 – FF	B001004B	Byte, Decimal
Data buffer 0 – 510	P	P		Binary (B) 00000000 – 11111111	B001004BD	Byte, Decimal
				Binary symbol (S) □□□□□□□□ - ■■■■■■■■	B001004BH	Byte, Hex
Data field 0 – 24 k	DF	DF			B001004BB	Byte, Binary
				ASCII – set #1	B001004BA	Byte, ASCII
				Status (Z) Null = 0 # Null =1	B001004BS	Byte, Binary symbol
					B001004BZ	Byte, Status
				Status symbol (Y) Zero = □ # Zero = ■	B001004BY	Byte, Status symbol
Data module 0 – 255	B	B	Wort	Decimal (D) 0 – 65535	B001004W	Word, Decimal
					B001004WD	Word, Decimal
				Hexadecimal (H) 0 – FFFF	B001004WH	Word, Hex
Data buffer 0 – 510	P	P			B001004WB	Word, Binary
				Binary (B) 00000000 00000000 – 11111111 11111111	B001004WS	Word, Binary symbol
Data field 0 – 24 k	DF	DF			B001004WZ	Word, Status
					B001004WY	Word, Status symbol
				Binary symbol (S) □□□□□□□□ □□□□□□□□ - ■■■■■■■■ ■■■■■■■■	B001004WF	Word, Colour variable
					B001004WM	Word, Message
				ASCII – set #2	B001004WA	Word, ASCII
				Status (Z) Null = 0 # Null =1		Contents can be displayed as ASCII Text display up to 80 char. (40 + 1 data word)
				Status symbol (Y) Zero = □ # Zero = ■		00H must be entered as text end when the text is less than 80 characters
				Message 1 word attribute + 80 char. text		The first data word is interpreted as an attribute

Fig. 3–14 Displaying Variables – Sheet 2

Variable Range	Identifiers German	Identifiers English	Format	Display type/ Number range	Example: Call	Remarks
Timers 0 – 127	T	T	Word	Decimal (D) 0.01 s – 9990 s		The current time in seconds is always displayed
				Status (Z) Null = 0 # Null =1	T1	Timer, Time
				Status symbol (Y) Zero = □ # Zero = ■	T1Z	Timer, Status
				Text up to 30 char.	T1Y	Timer, Symbol
					T1T	Timer, Text
Counters 0 – 127	Z	C	Word	Decimal (D) 0 – 65535	C1	Counter, Decimal
				Hexadecimal (H) 0 – FFFF	C1D	Counter, Decimal
				Binary (B) 00000000 00000000 – 00011111 11111111	C1H	Counter, Hex
				Binary symbol (S) □□□□□□□□ □□□□□□□□ - ■■■■■■■■ ■■■■■■■■	C1B	Counter, Binary
				ASCII – set #2	C1S	Counter, Binary symbol
				Status (Z) Null = 0 # Null =1	C1Z	Counter, Status
				Status symbol (Y) Zero = □ # Zero = ■	C1Y	Counter, Status symbol
				Text up to 30 char.	C1T	Counter, Text
					C1F	Counter, Colour variable
					C1A	Counter, ASCII
Date	D	D		8 Places: TT.MM.JJ.	D	Date
Time	C	X		8 Places: HH.MM.SS	C	German Version
					X	English Version

Fig. 3–14 Displaying Variables – Sheet 3

3.2.1.4.3 Explanation of the Variable Display “STATUS” and “COLOUR VARIABLE”

“Status” Display

Polling the status of a variable evaluates it as **“not equal Zero”**.

If, for example, the counter is still active the status is **“1”** or a solid box **“■”**. If the counter has counted out, a **“0”** or hollow box **“□”** represents the status.

The following are available: I / O / M / SM / T / C / DB / DP / DF / EI / EO

Display as byte/word:

Display: 1 character **0** / **1** or ■ / □

Example:

I0BZ	Byte, status	1
I0BY	Byte, status symbol	■
I0WZ	Word, status	1
I0WY	Word, status symbol	■

Displaying “Colour variable”

The colour variable is used for special effects on the screen. The colour variable can be used to change the attribute of a text output, for example. The user’s own alarm message can change colour by PLC program, for instance.

The following are available: I / O / M / SM / C / DB / DP / DF / EI / EO

Display as word:
The highbyte of the colour variable determines the number of characters whose attribute is changed. The lowbyte contains the new attribute.

Example: F = Colour variable (not visible)
The following entries are made consecutively in the colour variable (e.g. a data word):

Entry:	**Effect:**
Colour variable	F
Call screen	This is a text
F = 1108H (17 characters, BAS underscore)	This is a text
F = 0809H (8 characters, BAS bright+underscore)	**This is** a text
F = 0508H (5 characters, BAS normal+underscore)	This **is** a text

3.2.2 Program Module (Terminal Emulation Program) “Modem”

The Terminal Emulation Program **“Modem”** is called from the DBG utility software with the function key

and is used to operate the DB500 in On−Line mode.

This operating mode facilitates − among other things − the initializing of the module and the setting of printer parameters.

The connection between the programmer and the DB500 is made via the EP/AG connection module in the programmer.

Parameters of the EP/AG Interface

The terminal program initializes the serial interface of the EP/AG as follows:

Baud rate:	9600 Baud
Parity:	even
Data bits:	8
Stop bits:	1
Transfer mode:	20 mA − current loop

This setting cannot be changed by the utility program. The EP/AG is deactivated when the emulation program is exited.

Screen Display

When the terminal program is called the entire screen is cleared and the cursor is placed in the top left−hand corner. The echo is made via the DB500.

The terminal program is exited using the function key

.

3.2.2.1 Operating the DB500 from the Terminal

KC Commands

Apart from loading texts/screens and printer control the interface is also used to operate the module independently of the PLC.

As the DB500 has its own processor, a record memory printout can be started independently of the function module, e.g. during startup or program stop.

The following is a list of terminal commands and their meaning:

KC=XXXXXX The entry always starts with the command request **KC=** followed by the command. The underscored letters are enough to identify the commands, and there is no need for spaces. The commands are entered in upper case letters.

KC=INIDB Initializing of the DB500, setting internal parameters for record memory, creating the PLC—independent printer parameters, etc.

KC=ERASE Clears the record memory.

KC=INHALT Displays text memory.

KC=DIRECTORY Displays text memory.

KC=RESET Equivalent to Reset key on module.

KC=HARDCOPY Print screen.

KC=PRINT Starts a record memory printout following conformation or modification of the current parameters.

Modem operation is exited using the key **F6**.

3.2.3 “Loader” Program Module

The **“Loader”** program module is called with the function key

.

After the call the following screen is displayed and prompts the user to enter the filename.

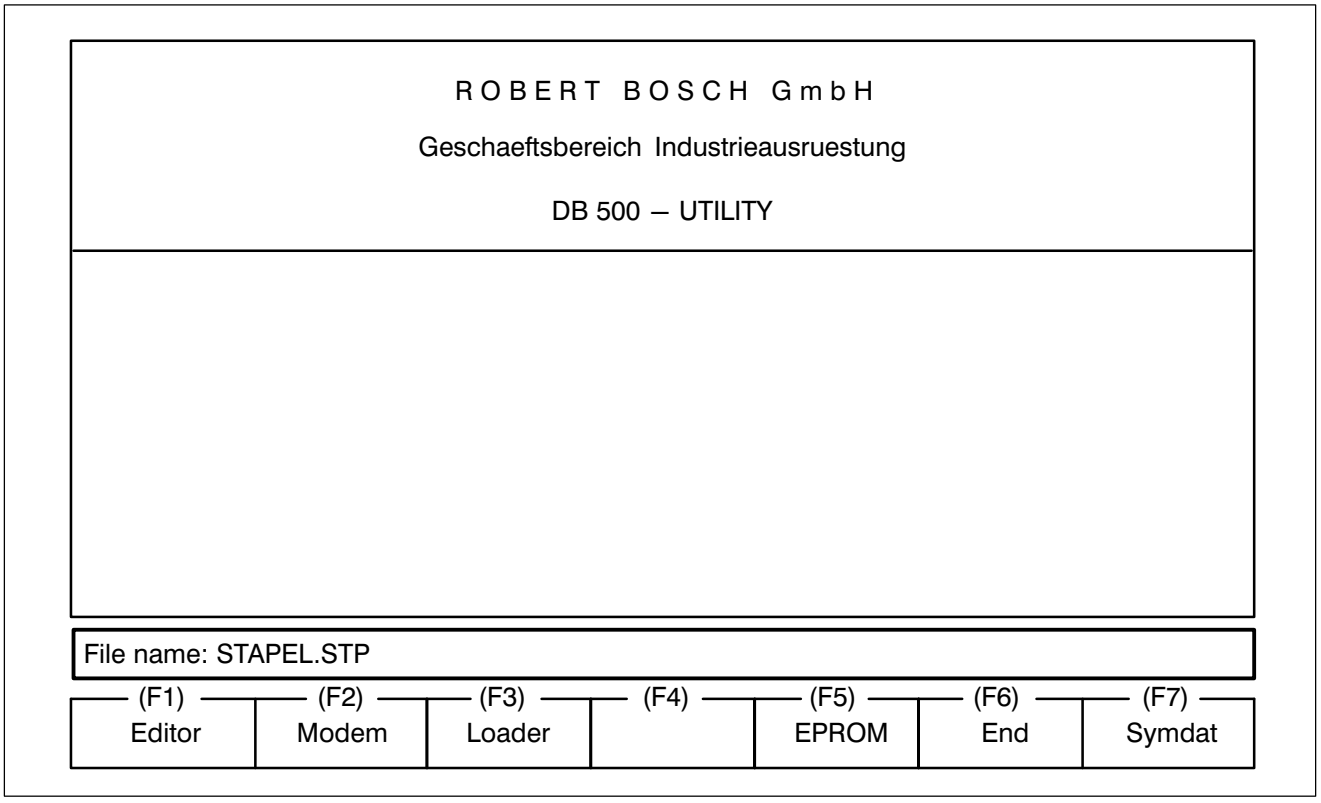


Fig. 3–15 Screen after calling the “Loader” Program Module

The **“Loader”** performs the transmission of any desired data (max. 128 kW) from the programming unit to the DB500.

The transfer format is based on the transmission convention BUEP19 (see Description P.–Nr.4130).

The transmission is effected from the hard disk drive of the programming unit. The program is run using the programming unit’s operator prompts.

When preparing the file to be loaded, depending on the extension, the program differentiates between

.STP – file and
.DG5 – file.

If the entered file has the **.STP** extension, then this file is interpreted as a batch file. The lines are called as individual files and converted. Text and graphics can be combined within these files.

If the entered file has the **.DG5** extension, then it is identified as a loadable file that can be loaded immediately, since no conversion is neces-sary.

With all other extensions the file is converted, filed as a .DG5 file and then loaded.

Special functions with /S and /N (only applies to batch files)

If there is a space smaller than 4 characters between 2 ASCII characters, this is filled with blanks by entering /**S**.

Example: This is a text.

Inverting without /**S:** This is a text

Inverting with /**S:** This is a text

/**N** cancels /**S**.

Program handling

```
STAPEL.STP
/S                      ;Converting all screens with space bridging
GRUND.001
EINSCH.002
.
SKTABELL.154
or
GRUND.001 /S      ;Converting this screen with space bridging
EINSCH.002
.
SKTABELL.154 /S   ;Converting this screen with space bridging
```

As from DB500 Version 102, the functions /S and /N are meaningless!

Automatic Language Adaptation (valid as from Version 1.7 and only usable in the batch file)

As from the DBG−SW Version 1.7, existing DB500 screens can be loaded automatically into foreign−language DB500 modules.

Procedure for automatic language conversion

− All screens are loaded independently of the set DB500 language and are displayed on the DB500 monitor, as long as they start with the beginning−of−screen identifier **Bxyz** or **Gxyz** and end with the end−of−screen identifier **BE or GE**.

− The variable identifiers are converted into other languages with system definitions.

/K Converts all variable identifiers into German
e.g. Input I −−−> Eingang E

/C Converts all variable identifiers into English
e.g. Ausgang A −−−> Output O

Example

STAPEL.STP

```
/C                    ;Converting all screens into English
GRUND.001             ;Variable identifiers
EINSCH.002
.
.
SKTABELL.154
```

or

```
GRUND.001  /K         ;Converting this screen into German
                      ;Variable identifiers
EINSCH.002            ;This screen stays in the original
.
.
SKTABELL.154 /K       ;Converting this screen into German
                      ;Variable identifiers
```

When converting variables it must be noted that the variable identifier **'C'** has different meanings depending on the language.

If **C** or **C000** (Time Display) is defined as a variable, the English designation **'X'** is entered during the conversion /**C**.

If **C0** or **C00** (Counter Display) is defined as a variable, the German designation **'Z'** is entered during the conversion /**K**.

3.2.3.1 Loading the DB500

To load text and graphics to the DB500 the programming unit must be connected to the X31 interface (20mA) of the DB500.

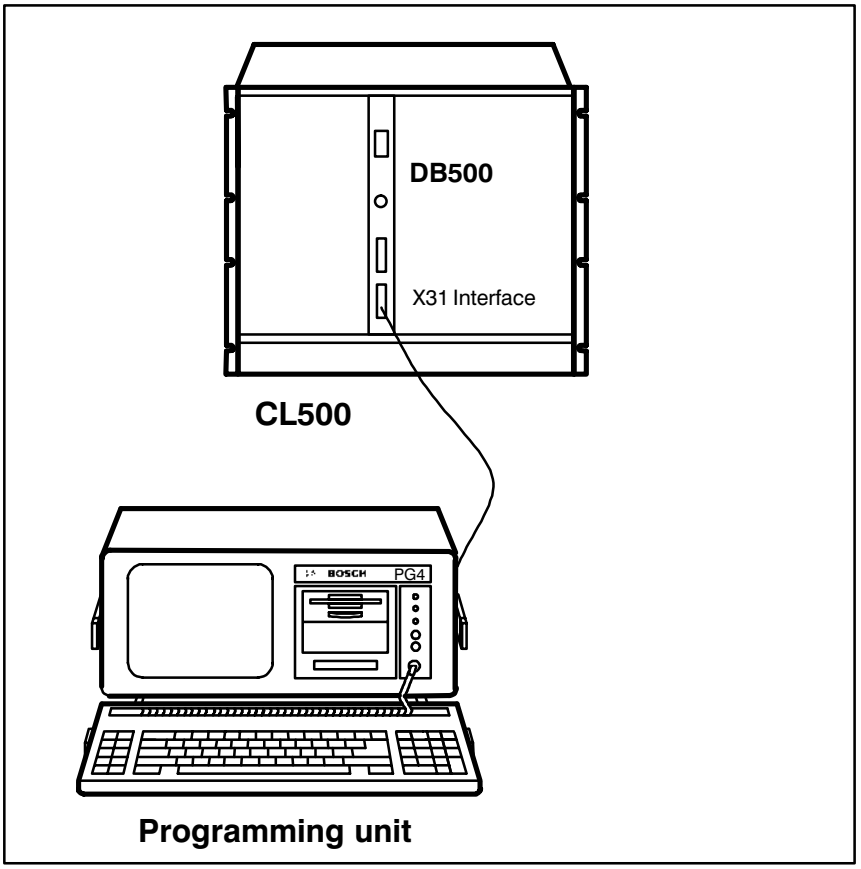


Fig. 3–16 Connecting the programming unit to the DB500

The following messages can appear during tansmission:

- Transmission interrupted

  Action: Check cable and DB500

- Insufficient text memory. The file is larger than the memory area of the plugged–in module.

  Action: Reduce texts or screens or fit larger memory module.

Transmission from the programming unit to the DB500

The program module **"Loader"** transmits texts and screens in condensed form and ordered sequence via the X31 interface (20 mA) to the user memory of the DB500.

The text files and screen files are contained in the .STP file created by the user.

The .STP file is converted into a DG5 file when the loading process is called.

Note ☞ **Only the DG5 file generated after conversion can be loaded into the DB500.**

3.2.4 “EPROM” Program Module

The “EPROM” program module makes it possible to transfer the file to EPROM modules of the DB500.

The hardware and software of the DB500 permits the use of EPROM modules of the CL500 for storing texts and graphics. The EPROM programming software converts the texts and graphics created with the DBG Editor into a format which the DB500 can read, and writes this file to an EPROM module.

The program module is called with the function key

and the user is prompted to enter the filename.

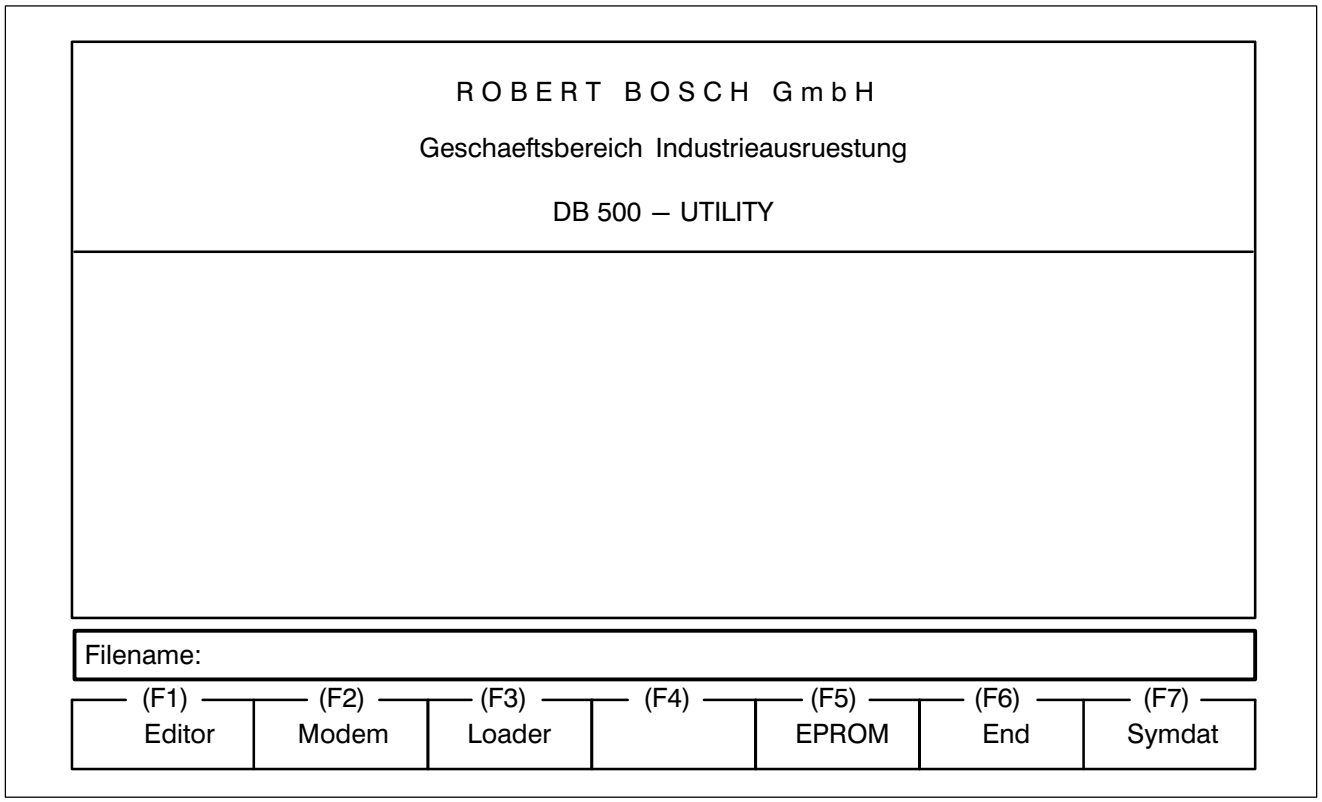


Fig. 3–17 Screen after calling the “EPROM” Program Module

Screen Display

After entering the filename followed by **< Enter >** a conversion attempt is executed, and then the following screen appears on the monitor.

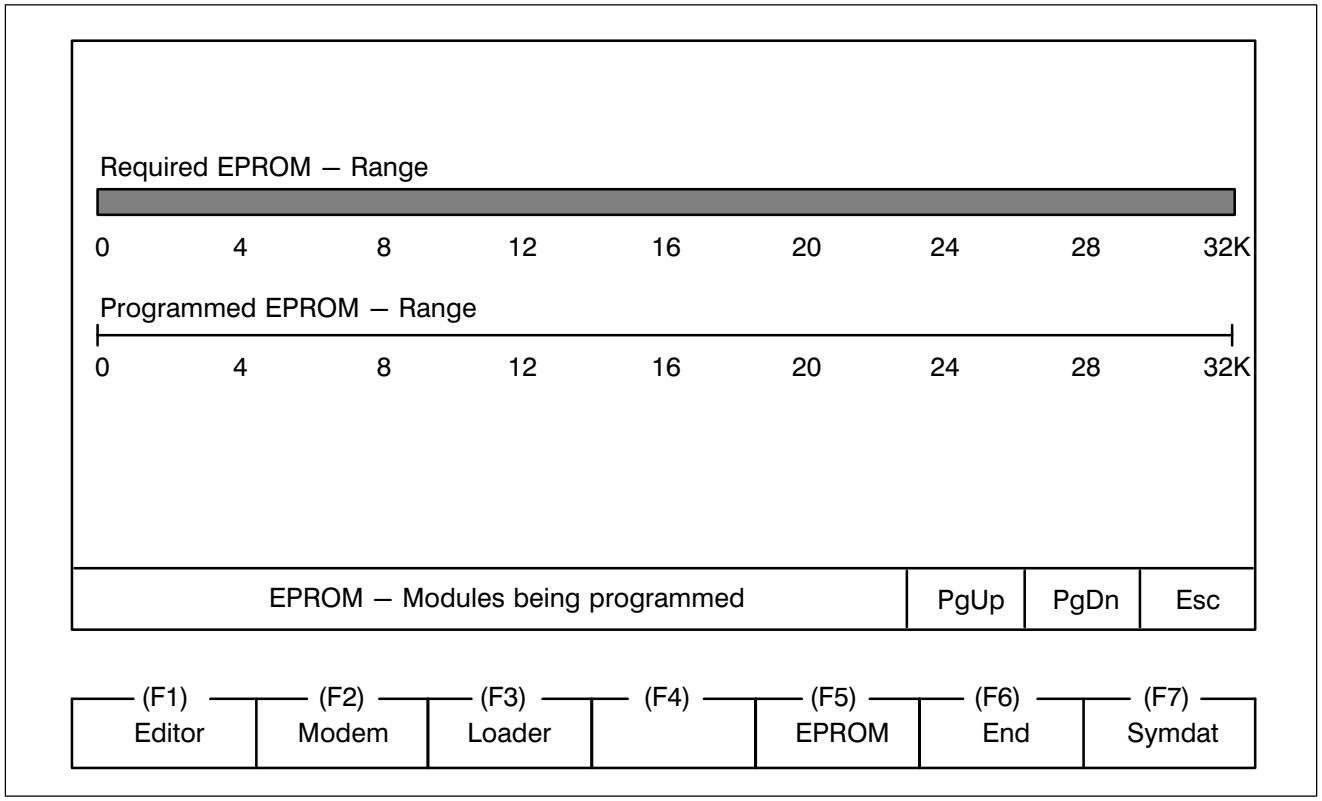


Fig. 3–18 Screen after entering filename

The bars displayed on the screen have the following meanings:

1st Bar: Required EPROM range
– Shows the size of the graphics file to be programmed. The bar reduces from left to right as the transfer proceeds.

2nd Bar: Programmed EPROM range before and during the programming operation.
– Shows the size of the file programmed so far.

3.2.5 “End” Program Module

The **“End”** program module terminates the program. After calling the program module with the function key

the program is exited and no further inquiring is performed.

3.2.6 “Symdat” Program Module

The **“Symdat”** program module makes it possible to enter comments from a symbol file.

The program module is called with the function key

and prompts the user to enter the symbol filename.

Only the I/O/M/T/C comments are entered.

These comments and their associated operands are created in a file of the same filename but with the extension **.TXT**.

If the user specifies a filename without an **.S5S** extension, the message

Wrong file type

is given. If a file of the same name alreay exists with the extension **.TXT**, then the message

Overwrite “Path and Filename” Module Yes/No

appears. After the filename is entered the user is prompted

Conversion including symbols Yes/No

The user can choose whether or not the symbol name edited in the symbol file is to be part of the text in the text file to be created. After conversion is complete the user is prompted

Load text file? Yes/No

“Yes” triggers the jump to the **“Editor”** with the created **.TXT File.**

“No” returns the screen to the main menu.

4 PLC Software (Communication between PLC and DB500)

The software for communications between the DB500 and the CL500, and between the programming unit and the DB500 consists of the “DIAG500” and “TRANS500” function modules.

To be able to create a diagnosable user program it is also necessary to use the “KETTE” function module for sequential step cascades. This is located in the **MADAP500** software package.

Note ☞ **The “MADAP500” software package is not described here. This is detailed separately in P.–Nr. 4142, which should be consulted when using the DB500.**

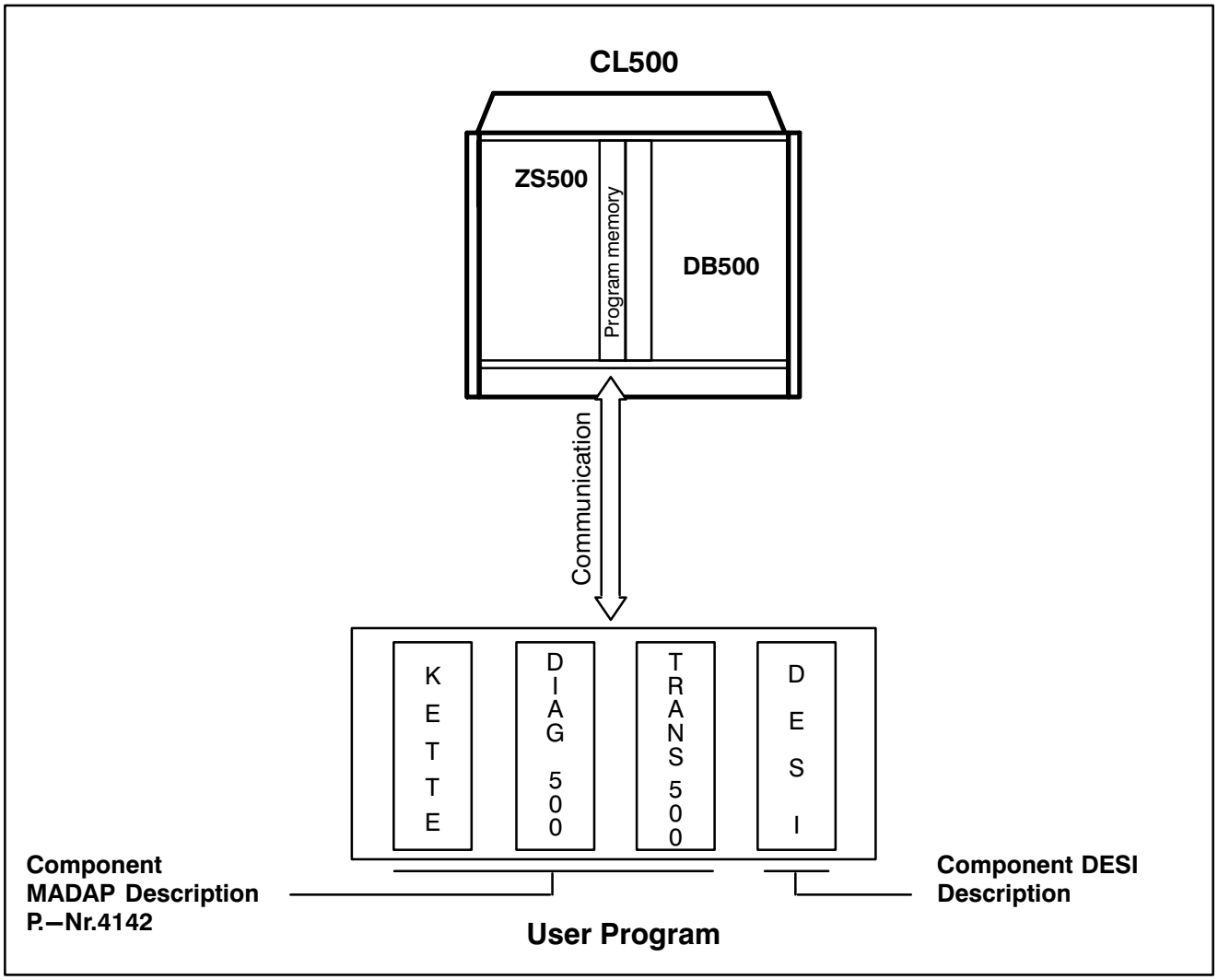


Fig. 4–1 DB500 Software

4.1 Standard Function Modules for Diagnostics

General

DIAG500 and TRANS500

- The DIAG500 monitors the sequencing process steps and checks the “Step Module” for the step—on conditions that are programmed in it. It displays missing cascade criteria.

- The TRANS500 is responsible for operating the DB500.

The function modules have permanently defined parameters which the user must assign specific operands, data words or constants in the associated program module.

The Standard Function Modules must be incorporated in the user program by means of symbol tables and module lists.

4.1.1 "DIAG500" Function Module

4.1.1.1 General

The DIAG500 function module is used to monitor the sequencing process steps of a machine or plant that is programmed with step cascade technology. In the event of a fault the monitored process steps are diagnosed for untrue step–on conditions. The first level error message is displayed at the DB500. The follow–on errors are displayed by highlighting the cascade number.

With one DB500 and the DIAG500 module up to 64 step cascades with a maximum of 128 steps can be monitored.

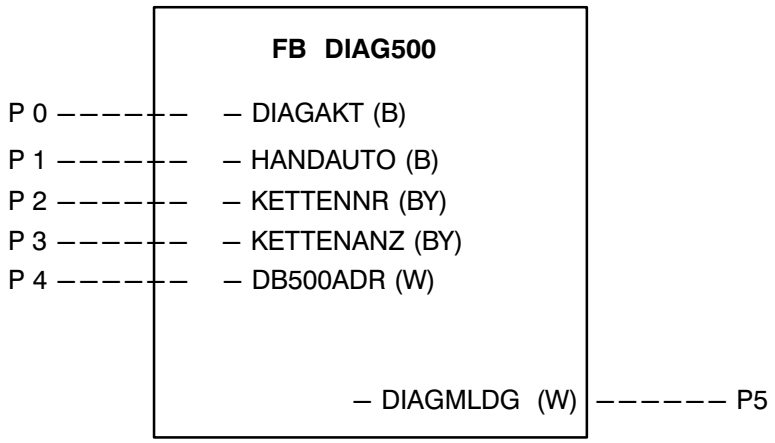
The module must be continuously run, but diagnosis is only performed when a error message comes up.

Parametering

The DIAG500 module can be called by any of the PLC program modules OM or PM. This gives the modules the parameters for controlling the diagnostic module.

The DIAG500 function module has 5 input parameters and one output parameter.

Call module



4.1.1.2 Input and Output Parameters

	Par.	Name	Format	Explanation of Parameter
Input Parameters	P0	–DIAGAKT	(B)	Changeover between manual and automatic diagnosis. Manual diagnosis is only performed when this parameter is 1.
	P1	–HANDAUTO	(B)	Changeover of diagnostic display between manual and automatic branch with P1 = 1 for automatic branch P1 = 0 for manual branch.
	P2	–KETTNR	(BY)	Number of cascade to be diagnosed in manual (Number of data module assigned to the cascade).
	P3	–KETTANZ	(BY)	Number of cascades to be processed
	P4	–DB500ADR	(W)	Address of diagnostic module to be operated

Note ☞ **Only cascade numbers for which the data modules are defined may be entered.**

Output Parameters	P5	–DIAGMLDG	(W)	The error messages from the function module are output, see subsection 4.1.1.3

4.1.1.3 Error Messages

In output parameter P5 the FM DIAG500 outputs the errors which it has detected (parametering errors or errors in the PLC program).

Bit	Cause of error
15	Group error: at least 1 of the following bits is active
14	DB500 with invalid version number
13	Diagnosis module not initialized
12	Data module for cascade not available or too short
11	No free memory in data field
10	Step number larger than the defined number of steps of the cascade to be diagnosed
9	Cascade number not permitted for manual diagnosis
8	Number of cascades not permitted
7	
6	Structure error in jump distributor, e.g. not only jump com−mands are programmed in jump bar of cascade
5	
4	More than 64 step−on conditions in operation branch
3	Unacceptable command in operation branch
2	
1	
0	Warning: cascade number = 0 (no group error message)

About Bit 11

For the TRANS500 and DIAG500 function modules the internal data field (not programmable by user) is divided into 12 freely usable blocks.

Of these, the DIAG500 module occupies 1 block for each call and the TRANS500 module 2 blocks.

The error occurs when too many function modules are contained in the program. The distribution of blocks is only executed on control startup.

4.1.1.4 Criteria Analysis

In displaying the diagnostic result a difference is made between **Manual and Automatic operation**. In the "MANUAL" mode the first untrue branch of the currently active step in the selected cascade is diagnosed and displayed (BEFA = command output branch WSB = step−on condition). In the "AUTOMATIC" mode the end of module (EM) is selected at the end of the step. The FM then searches for the start of the automatic branch and analyzes just as with manual operation. This diagnostic technique necessitates the following step module structure.

	PLC Step Module	Remarks
1	JP (A)	Jump distributor
2	JP −SCHRITT n JP −SCHRITT n + 1 JP −SCHRITT m	Absolute jumps to the step−on conditions m = 128
3	−SCHRITT n L D8,A A A.4 JPC −AUTO ; 1st BEFA operation block = −BEFA ; 2nd BEFA operation block A −BEFA = −BEFA ; 1st WSB operation block = −WSB EM −AUTO ; 1st BEFA operation block = −BEFA ; 2nd BEFA operation block A −BEFA = −BEFA ; 1st WSB operation block = −WSB ; 2nd operation block A −WSB = −WSB EM	Step programming Input operation mode Automatic or inching active ? Go to auto branch Manual mode Several BEFA branches possible Looping through possible Step−on conditions Several WSB branches possible End of manual mode Automatic mode Several BEFA branches possible Looping through possible Step−on conditions Several WSB branches possible Looping through possible End of manual mode

The following commands are acceptable for creating the step—on conditions:

Command		Display	Condition true	Condition not —true
A	AND	A	1	0
AN	AND NOT	AN	0	1
O	OR	O	1	0
ON	OR NOT	ON	0	1
=	EQUAL TO	=		
(	LEFT BRACKET	(		
)	RIGHT BRACKET	)		
)N	RIGHT BRACKET WITH NEGATION	)N in instruction list only		
NOP0		no display		
NOP1		no display		
S	SET			
R	RESET			

The **S** and **R** commands should be programmed, so long as the following criteria is complied with:

- S and R may not be linked to BEFA or WSB.
- As with equal—to signs, S and R can only take place directly before the assignment to BEFA or WSB.
- Only outputs and markers are permitted as operands.
- The use of scratch markers is not permitted.
- Only the status of the output or marker on which the S or R command is effected is displayed in the diagnostic screen of the DB500.
- Bit operations with register bits are not permitted.

For the operation commands A, AN, O, ON, the following operands are acceptable:

- Inputs
- Outputs
- Markers
- Special markers
- Timers
- Counters

4.1.1.5 Diagnostic Fadeout

If the step−on condition is true (WSB = 1) the step concerned is not diagnosed. The diagnosis of untrue BEFA branches is also overridden.

If this is required, however, the PM **"Kette"** must be changed accordingly (see **MADAP500 Description** P.−Nr. 4142).

First level message in the data module

On the arrival of a first level message the function module enters the detected messages in the DB120 data module as CL500 Opcode with date and time. For this to happen, the DB120 must be integrated in the user program with at least 72 words.

Control flags are filed in data word D0 of the DB120. D0.0 is set by the function module to 1 when a first level message was entered. This bit should be polled by the user program and reset to zero after the mes-sage is read out.

Data word D0.1 controls the response of the function module whenever there is a first level message without the readout of the old message having been acknowledged by resetting bit D0.0.

When D0.1 = 0, the first message is always overwritten by the new first level.

When D0.1 = 1, the new message is not entered in the data module and is lost.

Structure of the DB120 Data Module

DW	Highbyte Contents	Lowbyte Contents
D000	Flags	
D002	Day	Month
D004	Year	Hour
D006	Minute	Second
D008	Weekday (0 = Sunday)	Spare
D010	Cascade number	Step number
D012	Module type	Module number
D014	Cascade status	Number of messages
D016	1st Opcode	
D018	2nd Opcode	
D020	3rd Opcode	
D022	4th Opcode	
.		
.		
D140	63rd Opcode	
D142	64th Opcode	

Date format

The date and time of the first level message are entered by the function module into data words D2 to D8 as hex data. Weekdays are identified as follows:

0 = Sunday
1 = Monday
2 = Tuesday
3 = Wednesday
4 = Thursday
5 = Friday
6 = Saturday

Number of Messages

Here the function module enters the number of operations that belong to the current first level. The display is hexadecimal. Only the first 64 operations are filed in the data module. If the step−on condition consists of more than 64 conditions then the number of messages is set to 65.

Module Type

This byte indicates the module type of the faulty cascade. Here, the value 1 = program module is entered in the CL500.

Cascade Status

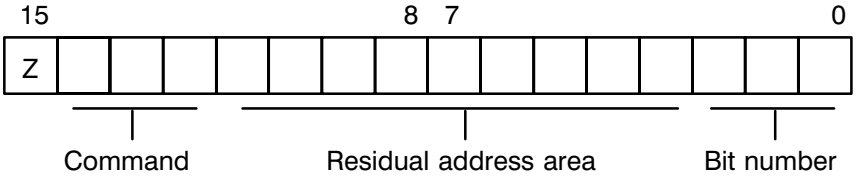
This byte indicates the mode of the faulty cascade at the time of the first level message:

1 = Cascade in manual
2 = Cascade in inching
4 = Cascade in automatic.

Opcode

The instructions of the faulty operation block are filed by the function module in the data module as CL500 machine code. In addition to the Opcode the current status (true/not true) of the step−on condition is displayed.

Format of Direct Bit Commands



Z = Status of operation

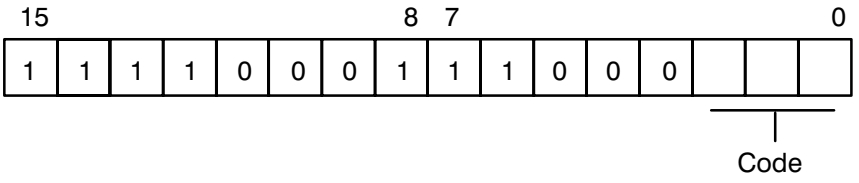
1 = Instruction true
0 = Instruction not true

Command:

Bit 14	13	12	
0	0	0	A (And)
0	0	1	AN (And not)
0	1	0	O (Or)
0	1	1	ON (Or not)
1	0	0	not permitted in the operation block
1	0	1	not permitted in the operation block
1	1	0	= (equal to)
1	1	1	not permitted in the operation block

Area address				Residual address:						Hex–	Operand:	
Bit	11	10	9	8	7	6	5	4	3	adr.		
from	0	0	0	0	0	0	0	0	0	00H	C	(Counter status)
to	0	0	0	0	0	1	1	1	1	0FH		
from	0	0	0	0	1	0	0	0	0	10H	T	(Timer status)
to	0	0	0	0	1	1	1	1	1	1FH		
from	0	0	0	1	0	0	0	0	0	20H	SM	(Special markers)
to	0	0	0	1	1	1	1	1	1	3FH		
from	0	1	0	0	0	0	0	0	0	40H	not permitted in	
to	0	0	1	1	1	1	1	1	1	7FH	operation block	
from	0	1	0	0	0	0	0	0	0	80H	I	(Inputs)
to	0	1	0	1	1	1	1	1	1	BFH		
from	0	1	1	0	0	0	0	0	0	C0H	O	(Outputs)
to	0	1	1	1	1	1	1	1	1	FFH		
from	1	0	0	0	0	0	0	0	0	100H	M	(Markers)
to	1	1	1	1	1	1	1	1	1	1FFH		

Format of Bit Special Commands



Code:			Command-code	Command	
Bit	2	1 0			
	0	0 0	F1C0H	(	(left bracket)
	0	0 1	F1C1H	O(	(or left bracket)
	0	1 0	F1C2H	)	(right bracket)
	0	1 1	F1C3H	)N	(right bracket with negation
	1	x x		not permitted in the operation block	

The NOPO and NOP1 commands permitted within the operation block are entered as an instruction in the data module.

4.1.1.6 Module Characteristics

Module name:	DIAG500.P50
Module length:	2213 words
Operation time:	
– without diagnosis	4.7 – 8.9 ms (depending on the number of cascades)
– with diagnosis (2 conditions)	7.7 – 9.1 ms with 64 cascades
Called Module:	DM120
Used data words:	D0 – D144
Used markers	M240.0 – M255.7

4.1.2 “TRANS500” Function Module

4.1.2.1 General

The “TRANS500” function module manages data traffic between the ZS500 and the DB500.

It handles the following functions:

- Selection of DB500 mode
- Screen driving
- Transfer of current time from the SK500 to the DB500
- Setting the target values and alert limits for limit control
- Transfer of screen numbers for displaying function key rails
- Cursor control for step cascade selection
- Display of DESI error messages
- Diagnosis with ladder diagram display
- Transfer of set values to the designated data modules
- Maintaining priority of operating modes

Parametering of the TRANS500

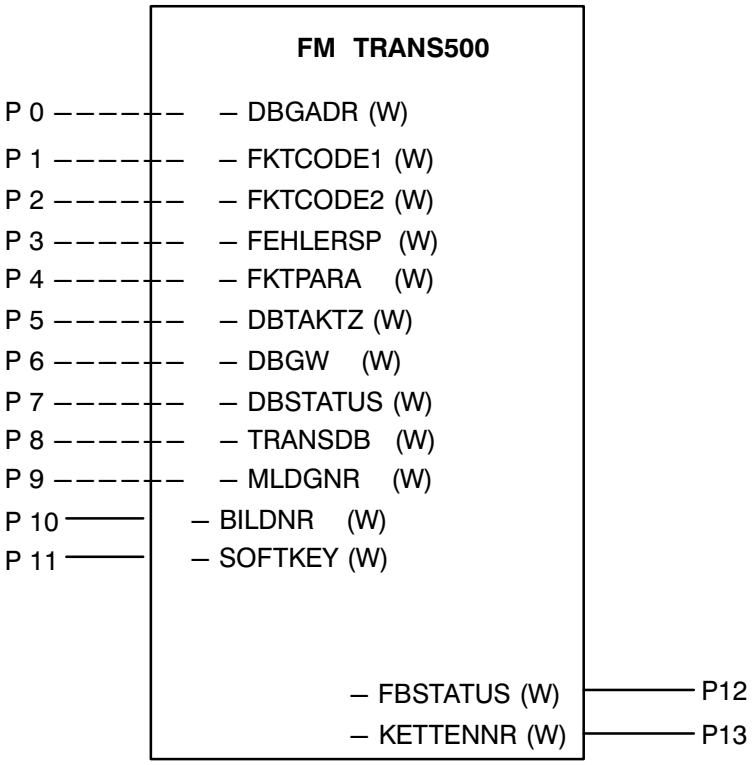
The “TRANS500” module can be called by any desired program or organisation module (PM, OM) of the PLC program. At the same time the module is provided with the parameters for controlling the diagnostic module.

The module uses the scratch marker M240.0 – M255.7 from the PLC range. Apart from the parameter transfer there is no other exchange of data between the function module and the user program.

Call module

CM –TRANS500,14

The TRANS500 module has 12 input parameters and 2 output parameters.



4.1.2.2 Input and Output Parameters

Input Parameters

No. of P	Name	Format	Explanation
P0	–DBGADR	(W)	Block address of the DB500 to be addressed. The entry is the address switch position on the DB500.
P1	–FKTCODE1	(W)	Function code 1 Mode selection (display) of the DB500
P2	–FKTCODE2	(W)	Function code 2 Mode selection (display) of the DB500
P3	–FEHLERSP	(W)	Definition of the mode of the DB500 record memory for error management and error output.
P4	–FKTPARA	(W)	Function parameters Transfer of data belonging to an operating mode selected by the function code (e.g. page number, cursor position)
P5	–DBTAKTZ	(W)	Cycle time display Data module number for cycle time-display
P6	–DBGW	(W)	Limit display Data module number for limit display
P7	–DBSTATUS	(W)	Status display Data module number for status display
P8	–RESERVE	(W)	Spare
P9	–MLDGNR	(W)	Message number Number and attribute of a user message that is logged via printer, screen or record memory

P10	–BILDNR	(W)	Screen number Number of user screen to be displayed
P11	–SOFTKEY	(W)	Softkey Number of the screen which is to be displayed as a softkey rail.

Output Parameters

No.	Name	Format	Explanation of the parameter
P12	–FBSTATUS	(W)	Acknowledgements from the TRANS500 to the user program
P13	–KETTENNR	(W)	Cascade number Cascade number selected by the cursor for manual diagnosis or entry of the cursor position when turning on the cascade cursor with P1 Bit 8.

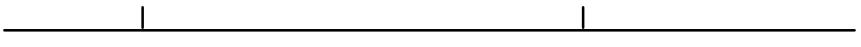
Parameter Acceptance

In order to save transit time and reduce the number of system bus access operations, the individual functions are divided into different PLC cycles. This means that the module parameters do not change during their operation.

The FM thus ensures that it only reads in and temporarily stores the input parameters at a specified time. The function module displays the time of acceptance in the output parameter FBSTATUS (P12) by setting bit 8 (synchronization). This bit is reset at the next module call, and the user program can take on new parameters.

Synchronization

Parameter



Parameter acceptance

4.1.2.3 Function Code and Function Parameters

In the function code (Parameters P1 and P2) operating modes programmed by the user are transferred to the diagnostic module.

Any necessary additions are transferred in function parameter (P4).

A function is executed in accordance with a given set bit (stat) or with a 0 ––––> 1 (flk) flank in the function code.

Parameter P1 / Function code 1

Bit No.	Explanation
15 flk	Hardcopy
14 stat	Scroll direction in LD
13 flk	Branch step–on in LD
12	
11 stat	Display mode in diagnosis screen Bit 1
10 stat	Display mode in diagnosis screen Bit 0
9 stat	IL diagnosis screen (0) LD diagnosis screen (1)
8 stat	Turn on cursor in cascade field = 1 Turn off cursor in cascade field = 0
7 stat	Change limit fast = 1 Change limit slow = 0
6 flk	Decrement limit at limit change
5 flk	Increment limit at limit change
4 flk	Cursor down (disagnosis and limit display)
3 flk	Cursor up (diagnosis and limit display)
2 flk	Page in cascade field
1 flk	Page in diagnosis field / Shift to LD
0 stat	Screen bright with diagnosis display = 0 Screen dimmed with diagnosis display = 1

Function code 2 / Parameter P2

Bit No.	Function parameter P4	Explanation
15		
14		
13		
12 stat		Status display numeric = 0 Status display symbolic = 1
11 stat		Display contents of text memory
10 stat	Op. code Op. address	Status display
9 stat		Display alert limit = 0 Not display alert limit = 1
8 stat	Page number in Bit 0 and 1	Graphic limit display (Bar chart)
7		
6 stat		Change target = 0 Change alert limit = 1
5 stat		Change limit
4 stat		Tabular limit display
3 stat		Cycle time display
2 stat		Variable address display = 0 Variable value display = 1
1 stat		Screen display
0 stat		Diagnosis

Diagnosis Display

For individual parameters the abbreviation Px(Bit) will be used.

Example: P1(2) stands for Bit 2 im Parameter P1.

The **"Diagnosis"** standard screen is selected by setting Bit P2(0). In this mode, the screen can only be switched to dim by P1(0). In the display **IL**, P1(9) = 0, there is a choice of three additional displays in the right−hand side of the screen:

P1(11)	P1(10)	Display
0	0	**Cascade display** 16 cascades with mode and step number are displayed. Paging within the chain display is effected via P1(2).
0	1	**Cascade summary** A summary of all 64 chains is displayed. Faulty cascades are marked.
1	0	**DESI display** When the CL500 is equipped with the DESI system, this display represents the arrangement between PLC I/O addresses and DESI module point.

In order to display a diagnosis the Bit "0" of the parameter P1 must show the status "0" (screen bright), as this function has a higher priority. Taking account of the above priorities, the user is free to decide which criteria (malfunctions, plant modes, etc.) should form the basis for the calling of screen displays, and paging and transmitting screens.

Ladder Diagram Display

In the diagnosis display, P1(9) permits toggling between the **IL** and **LD** display modes. P1(9) = 1 applies to LD. The display of step conditions is executed branchwise in LD. With a positive flank at parameter bit P1(13) the branches can be moved further (direction forwards only).

In the LD the DB500 displays a maximum of 5 parallel operations on one screen page. If the step to be diagnosed has more parallel contacts, the screen shift can take place through the branch. The scrolling direction is determined by P1(14). The following applies:

Parameter P1(14) = 1: Shift screen down
Parameter P1(14) = 0: Shift screen up
Scrolling is executed by 0 −−−> 1 flank in P1(1).

Limit change

A limit change can only be executed within the tabular limit display.

The value to be changed is determined by the cursor key. The bits 5, 6 and 7 in parameter P1 determine the direction and step size of the change. The following applies:

P1(5) : Limit is incremented at 0 ———> 1 flank
P1(6) : Limit is decremented at 0 ———> 1 flank
P1(7) : 0 ———> step size = 1
1 ———> step size = 100

The limit change is effected in the data module.

A limit display only takes place when at least 108 data words are integrated in the data module addressed by P6. With shorter module lengths all values are set to zero, an error message appears in the FM status word and P12(10) is set.

4.1.2.4 Operand Flags and Operand Addresses

When the status display is selected the function module must be told the operands and their addresses in encoded form as hex values in parameter P4. Only the operands that belong to the ZS500 in which the function module is incorporated are displayed.

Entry as hex value xyH where x = operand flag
y = operand address

x	Operand	x	Operand
0	no display	8	Data buffer
1	Inputs I/O field / Ext.fld.	9	Data field
2	Outputs I/O field / Ext.fld.	A	DESI Equipment
3	Markers	B	
4	Timers	C	
5	Counters	D	
6	Special markers	E	
7	Data words	F	

y	I/O Byte/ Ext. fld.	M Byte	Timers	Counters	Data module/ Data buffer Data field
0	0 – 31	0 – 31	0 – 31	0 – 63	0 – 30
1	32 – 63	32 – 63	32 – 63	64 – 127	32 – 62
2	Z0 – 31	64 – 95	64 – 95		64 – 94
3	Z32 – 63	96 – 127	96 – 127		96 – 126
4		128 – 159			128 – 158
5		160 – 191			160 – 190
6		192 – 223			192 – 222
7		224 – 255			224 – 254
8					256 – 286
9					288 – 318
A					320 – 350
B					352 – 383
C					384 – 414
D					416 – 446
E					448 – 478
F					480 – 510

No address need be given to display the special markers since they can all be displayed on a single screen page.

4.1.2.5 Data Modules for Standard Screens

With the standard display screens of data module, cycle time and limit, the contents of data modules are displayed. The necessary data module numbers are transferred in the following parameters

- P5 for cycle time display,
- P6 for limit display, and
- P7 for status display.

(For P5 and P6 see MADAP Description P.–Nr.4142, subsection 3.2.5)

Data field display

The TRANS500 module manages the data field like a data module. For this, the start address of the data field area to be displayed is transferred in parameter P7 (address area 0 .... 24576). Via the operand address it is then possible to select the following data field words without changing the start address.

4.1.2.6 Screen Display and Screen Transfer

In the function code (parameter P2) bit 1 switches on the screen display of the DB500. The function module carries on a dialog with the DB500 to ensure that no screen number is overread.

This is why the PLC program cannot change the screen number until bit 1 for **"enable screen change"** is set to zero in FM output parameter P12.

To display a softkey rail, parameter P11 transfers the number of the screen whose lines 24 and 25 should be displayed as a softkey rail.

P11 = 00 means no softkey display.

4.1.2.7 Operating Mode of the Record Memory (P3)

The P3 defines which data the diagnostic module collects in its record memory and how the memory contents are displayed.

When the bit is set (stat) or when the flank changes from 0 ---> 1 (flk) in the function code, the function is performed.

Parameter P3

Bit Number	Explanation
15 flk	Clear record memory
14 flk	Output record memory to printer
13 stat	Output record memory to screen
12	
11	
10 flk	Page record memory (direction thru bit 9)
9 stat	Output as FIFO = 1, Output as LIFO = 0
8 stat	Print all messages = 1 Print new messages only = 0
7	
6	
5	
4 stat	Record memory entry on printer
3 stat	Output record memory with DESI messages
2 stat	Output record memory with user messages
1	
0 stat	Output record memory with diagnostic messages

4.1.2.8 Output Record Memory to Printer

The print output of the record memory is started by P3(14). At the same time, bits 0, 2 and 3 define which messages should be printed. Printing can be recorded or terminated during printout by setting or resetting the appropriate bits. Each change of the printer parameters is logged in the printout.

The user program can recognize the duration of printout by the set output bit P12(3). This bit is also set when the printout of the record memory was started by a **KC Command** in modem operation.

4.1.2.9 User Message (P9)

In this parameter the user gives the DB500 a user−specific message selected with the message number. In addition to the message number, the module must be told the message attributes (e.g. message coming, message going).

To ensure that no message number is overread the user program should only store a new message when the appropriate stop bit, P12(2), is reset in the output parameter P12 of the function module.

Bit 15 indicates whether a user message is active (message coming P12(15) = 0) or whether the cause of the message has been remedied (message going P12(15) =1).

With selected screen output, the message is only displayed if no higher priority message is present. Higher priority messages are those which are generated independently by the diagnostic module, e.g. **“DESI message active”** or **“ZS transmits no data”**.

A displayed user message contains no directional information (coming/going); it can be cleared by transmitting the message number **Zero**.

Parameter P9

Bit Number	Explanation
15 stat	Message coming = 0 Message going = 1
14 stat	Message displayed on screen
13 stat	Message is entered in record memory with date and time
12 stat	Message is output to printer with date and time
11	
10	
9	
0 – 8	Message number: Message active = 1 – 511 No current message = 0

The message is only displayed on the monitor when the bit P9(14) is set and no higher priority messages (e.g. diagnostic active) are present.

4.1.2.10 Output Parameter P12

In output parameter P12 the TRANS500 FM transmits the internally detected program and systems errors and handshake bits to the user program for dialog with the DB500.

Parameter P12

Bit Number	Explanation
15	DB500 not found (wrong address setting)
14	DB500 not initialized
13	No operation of DB500 by TRANS500
12	DB500 firmware with invalid version number
11	Internal hardware error in DB500
10	Faulty variable in screen or status display
9	System command in operation
8	
7	Parameter acceptance at 0 ———> 1 flank
6	Language display bit 2
5	Language display bit 1
4	Language display bit 0
3	Record memory being printed (1)
2	Message coming/going enabled = 0 Message coming/going disabled = 1
1	Screen change enabled = 0 Screen change disabled = 1
0	

Status Message Bit 13

For the TRANS500 and DIAG500 function modules the internal data field (not programmable by user) is divided into 12 freely usable blocks.

Of these, the DIAG500 module occupies 1 block for each call and the TRANS500 2 blocks.

The error occurs when too many function modules are integrated in the program. The division of the blocks is only performed on control startup.

Status Message Bit 10

The error message bit P12(10) is concerned exclusively with the current screen display and is set by the function module in the following instances:

- **Status display**
  - An unavailable operand type has been transferred in parameter P4.
  - The data module to be displayed is not integrated in the PLC program.
- **Limit display**
  The data module entered in parameter P6 is not integrated or has less than 108 data words.
- **Cycle time display**
  The data module entered in parameter P5 is not integrated or has less than 36 data words.
- **Screen display**
  The user screen contains a data word variable from a data module which is not integrated or is too short.

Language display Bits 4, 5, 6

These three bits display the language selected on the DB500 module:

P9(6)	P9(5)	P9(4)	Language
0	0	0	German
0	0	1	English
0	1	0	Spanish
0	1	1	Dutch
1	x	x	not yet occupied

4.1.2.11 Output Parameter P13

In the “Diagnostic display” mode the user can use a cursor on a cascade number in the cascade field. The cascade number indicated by the cursor is output via the P13 and can then be used as an input parameter for manual diagnosis.

Note ☞

When the cursor position is set to “0” and the cascade cursor is activated (P1(8) changes from 0 to 1), the function module treats parameter P13 like an input parameter. The TRANS500 reads and displays the number transferred in the parameter as a new cascade position.

4.1.2.12 Priority Control

The DB500 can only operate in one mode at a time. Where several modes are set in the function code, the TRANS500 will decide which function the DB500 should execute.

The function module makes its choice of mode according to the following descending order of priority:

- Status display
- Call screens
- Cycle time display
- Limit control display
- Diagnosis
- Record memory display
- Display of table of contents

4.1.2.13 System Clock

The DB500 receives the current system time from the system coordinator SK500 via the FM TRANS; the time is written in the DB500 memory. In its initializing phase the function module sets the command “LCC” (load clock cyclical) with target address in the system area. SK500 then writes the current time in seconds in the specified system area. To keep bus traffic to a minimum, all other PLC modules which require the time should access this storage area. The function module sets bit 0 in the clock’s status word to indicate that the cyclical freshening of the time has been activated.

Setting the system clock is performed via the user program.

System Area for Date and Time Transfer

Address	Highbyte	Lowbyte
S500	Acknowledgement word for system command	
S502	Clock status word for PLC program	
S504	Minute (hex)	Second (hex)
S506	Hour (hex)	Day (hex)
S508	Month (hex)	Year (hex)
S510		Weekday (1 — 7)

Clock Status Word

Bit Number	Explanation
15	Error in time processing detected = 1
14	
13	
12	
11	
10	
9	
8	
7	
6	
5	
4	
3	
2	
1	
0	LCC system command from FM TRANS500 activated = 1

4.1.2.14 Module Characteristics

Module name:	TRANS500
Module length:	3595 words
Operation time: Standard screens:	1.0 – 4.5 ms
Screen display with variables (with 510 data word variables):	1.0 – 6.9 ms
Markers used:	M240 – M255.7

5 Technical Data

Memory: Text and Graphics Memory	RAM Modules 32, 64 and 128 KWords EPROM Modules 32, 64 and 128 KWords
Record memory	32 KWords
Interfaces	BAS signal at a BNC socket for connection to a monochrome monitor or flat screen. Floating earth potential 75 Ω Interface for connection of a colour monitor V24/20 mA as per VDI 2880 Part 2 for connection of a programming unit or printer
Baud rate	110 Baud .....19200 Baud
Non−operating temperature Operating temperature	−20 0C to +70 0C 0 0C to +55 0C
Enclosure class	IP 20 as per DIN 40050
Humidity class	F as per DIN 40050
Dimensions	Dual Eurocard size (160 mm x 234 mm)

Your Notes:

6 Re—Order Information

Item	Order No
DB500	062839
Memory modules:	
RAM 32 KWords	056768
RAM 64 KWords	062365
RAM 128 KWords	066845
EPROM 32 KWords	056769
EPROM 64 KWords	062366
EPROM 128 KWords	066848
Cable K7 to connect DB500 to Bosch PLC interface on programming unit	054334
Accessory Kit B3 to connect DB500 to Printer ML182:	
Connector set	054563
Cable	910152*
Accessory Kit B4 to connect DB500 to Printer ML294/321	
Connector set	054564
Cable	910152*
Accessory Kit B6 to connect DB500 to Bosch PLC interface on programming unit	
Connector set	054565
Cable	910152*
Utility program on diskette	063672
MADAP incl. extended functions (KETTE, DIAG500, TRANS500)	063696
Extension package, language—dependent program modules, symbol files, screens:	
German	069073
English	069075
Dutch	069074
Spanish	069022

* = Minimum order quantity 20 m

Your notes:

7 Alterations

The edition P.–Nr. 4125/E3 has been completely revised.

It is therefore unnecessary to go into individual alterations in detail.

1070 072 128-102 (92.10) GB · HB SP· AT/PLS · Printed in Germany