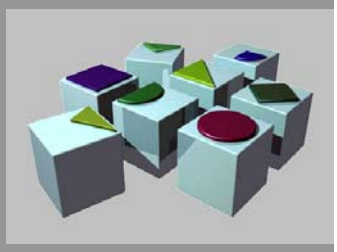
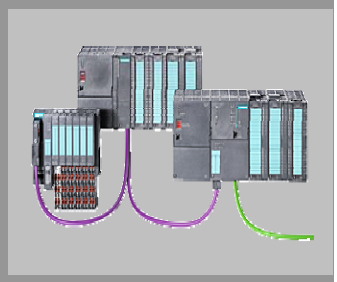
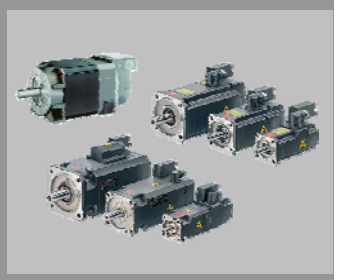






Manual

Visualization Operation Diagnostics
HMI Lite
Configuration Manual

Edition 2011

Solutions for Powertrain

TRANSLINE

SIEMENS

SIEMENS

Solutions for Powertrain

TRANSLINE - Visualization Operation Diagnostics HMI Lite

Manual

Manufacturer Documentation

2011 Edition

A5E40734099B AE

SINUMERIK® Documentation

Printing history

Brief details of this edition and previous editions are listed below.

The status of each edition is shown by the code in the „Remarks“ column.

Status codes in the „Remarks“ column:

A .... New documentation.
B .... Unrevised reprint with new Order No.
C .... Revised edition with new status.

Edition	Order No.	Remarks
03.04	A&D MC - Extranet	**A**
03.05	A&D MC - Extranet	**C**
01.07	A&D MC - Extranet	**C**
08.07	A&D MC - Extranet	**C**
2009	A&D MC - Extranet	**C**
2011	I IA&DT - E-Business Workplace	**C**

Further information is available on the Internet under:
http://www.automation.siemens.com

This publication was produced with WinWord V 7.0 and Designer V 6.0 using the documentation tool AutWinDoc.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

We have checked that the contents of this document correspond to the hardware and software described. Nonetheless, differences might exist. The information contained in this document is, however, reviewed regularly and any necessary changes will be included in the next edition. We welcome suggestions for improvement.

Subject to change without prior notice.

Available at: I IA&DT-E-Business Workplace
Printed in the Federal Republic of Germany

Siemens-Aktiengesellschaft.

0

Contents

Figures

Tables

Notes

1

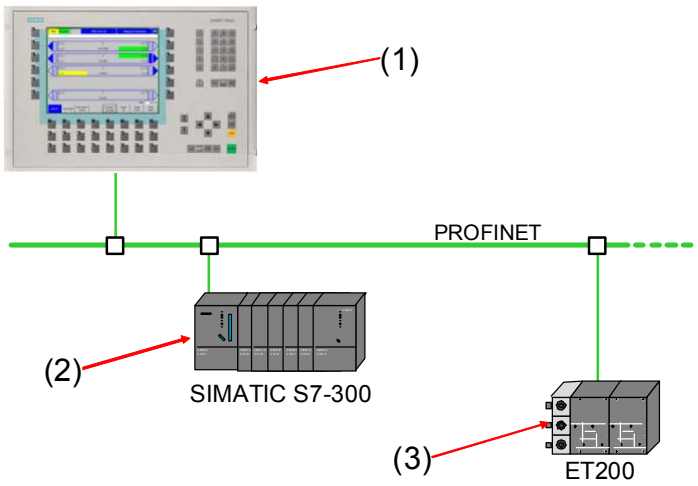
1 General Information

1.1 Product overview

HMI Lite is a user interface for the operator control and monitoring of machines. This user interface contains several screen forms for Windows CE operator panels from the SIMATIC product series and PLC blocks for supplying the screen forms.

The navigation within the individual screen forms is performed using a predefined menu structure, where the machine manufacturer also has the possibility to embed its own screens and so extend the existing menu structure.

HMI Lite is part of the "Solutions for Powertrain TRANSLINE" concept.



(1) Operator panel for displaying the HMI Lite screens
(2) SIMATIC S7-300 with the PLC program for supplying the screens
(3) External devices and I/O peripherals

Figure 1-1 System overview

1.2 Provided screens

Main menu: Preparation and setup

- Manual operation
- Power up condition
- Selecting/deselecting units
- Selecting/deselecting nut driver
- Selecting/deselecting nut driver groups
- Selecting/deselecting cycle types
- User operating screen

Main menu: Edit

- Machine overview
- Cycle times
- Workpiece count

Main menu: Diagnostics

- Messages
- Alarm Log
- Interface
- Version

Main menu: Hardware diagnostics

- PROFIBUS
- Drives
- MOBY-I
- AS-i
- Profinet
- Sinamics S120

Main menu: System

- System
- Panel Control
- Status variable

1.3 Basic knowledge

To commission the HMI Lite system, the following STEP 7 and WinCC flexible knowledge is required:

SIMATIC HMI (WinCC flexible)

- Installation of the WinCC flexible software
- Setup and operation of the SIMATIC HMI operator panels
- Configuring the interfaces and connections between HMI and the programmable controller
- Creation and parameterization of WinCC flexible objects
- Testing the HMI configurations

SIMATIC S7 (STEP 7)

- Installation of the STEP 7 software
- Handling the project archive files
- Working with programs that use several address types
- Working with symbolic addressing
- Creation and testing of application programs and troubleshooting
- Working with binary operations, timers, counters and comparators, and with arithmetic operations
- Development of programs that can reuse the same program block
- Working with data access functions
- Create data blocks
- Working with complex structures that contain parameters
- Including system functions (SFC) in a program
- Knowledge of the operation of SIMATIC S7 libraries
- Use of complex data structures for data storage

SIMATIC NET (STEP 7)

- Installation of the PROFIBUS peripherals and troubleshooting
- Installation of a PROFIBUS DP network and troubleshooting

1.4 Hardware requirements

HMI Lite is available for the SIMATIC HMI panels based on Windows CE listed in Table 1-1.

All screens support the touchscreen functionality and so can be converted to touch devices.

Operating devices

Description	Type of the installed screen	Touchscreen/keys
SIMATIC MP 277 10"	10.4" TFT display	Keys
SIMATIC MP 277 10"	10.4" TFT display	Touch screen
SIMATIC OP 177B 6”	5.7" LCD STN display	Keys
SIMATIC TP 177B 6"	5.7" LCD STN display	Touch screen

Table 1-1 Supported operator panels

Figure 1-2 Supported operator panels

Note

In addition to the configurations mentioned here, other project-specific configurations for other operator panels not listed here are also possible. The WinCC flexible configuration tool can be used to change the operator panel type (“Change Device Type…”) so it can also be used for other SIMATIC operator panels. The appropriate notes and restrictions for changing the device type are described in the WinCC flexible documentation.

1.5 Software requirements

1.5.1 Configuration and Programming Software / Licenses

Mandatory

Description	Version	Order no.
TRANSLINE HMI Lite	current	6FC5263-0PY11-0AG0
TRANSLINE HMI Lite	5.1	6FC5263-5PY11-1AG0

Description	Version
WinCC flexible	From 2008 + SP2
STEP 7	From 5.5

STEP 7 must be installed to edit the STEP 7 program and because of the fact that the WinCC flexible file is integrated in STEP 7.

Optional

Description	Version
S7-GRAPH	From 5.3
WINCC FLEXIBLE / PROAGENT FOR SIMATIC PANEL	From 2008 + SP2

The S7-GRAPH programming language can be used to graphically program machine sequences. Should an error occur, a ProAgent can be used to diagnose machine sequences programmed with S7-GRAPH.

This diagnostic capability means it is desirable to execute the manual functions using a S7-GRAPH sequencer. For this reason, HMI Lite contains a function block that can be used to execute the manual functions using a sequencer.

Note

The Service Packs can be downloaded from the Siemens product support at the following Internet address: http://www.ad.siemens.com/support

1.5.2 Runtime software / licenses

Mandatory

Description	Version	Order no.
HMI Lite copy license	current	6FC5263 0PY11 0AG1

Optional

Description	Version	Order no.
WINCC FLEXIBLE / PROAGENT FOR SIMATIC PANEL	2008	6AV6618-7DB01-1AB0

The license for ProAgent/MP Runtime is required only when the ProAgent process diagnostic screens are used.

Note

Further explanations for the PLC software requirements, for the supported operator panels and for the dependencies of other software packages are contained in the ProAgent manual.

■

2 Installation

2.1 Unpacking the source project

HMI Lite consists of a WinCC flexible project and STEP 7 blocks.

Both components are integrated in a STEP 7 project. This source project is supplied as a STEP 7 archive. To access the individual components, the STEP 7 archive must be unpacked with the Simatic Manager.

The unpacked source project from HMI Lite has the following folder structure:

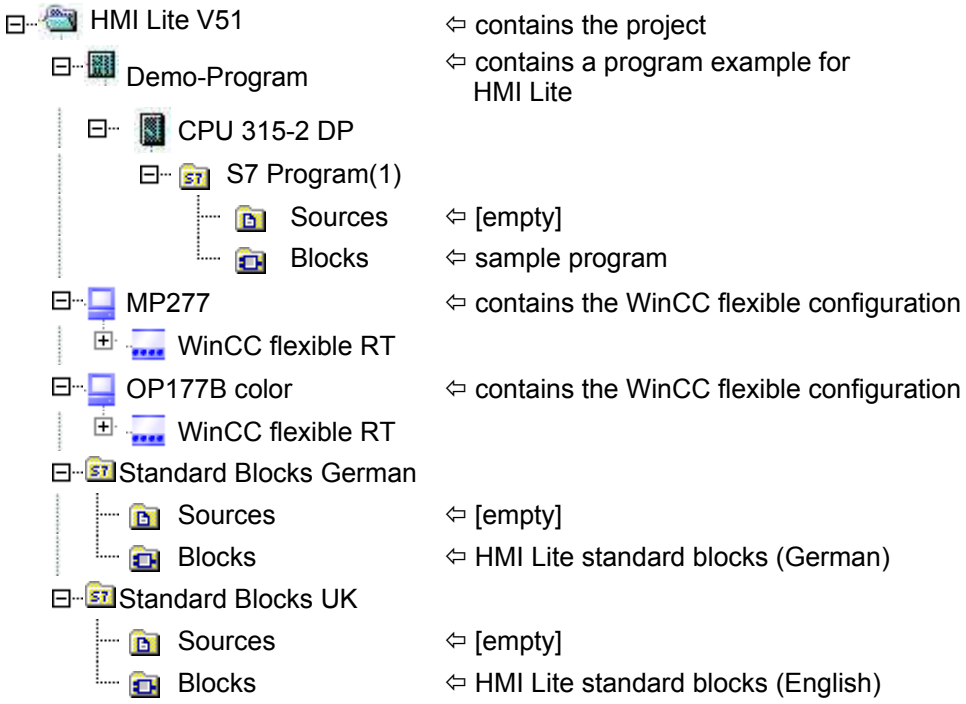


Figure 2-1 Structure of a HMI Lite S7 project

Note

A detailed description for unpacking a STEP 7 source project is contained in the Step7 documentation.

2.2 Proceed as follows

There are two ways of binding HMI Lite and the machine-specific program:

Use the Demo program of the HMI Lite S7 project as basis, adapt the parameterizations and configurations to the actual installation, and copy the machine-specific blocks in to the block folder.

Use the S7 project created by the OEM as basis and copy the HMI Lite blocks and the WinCC flexible configuration into this project. Note this procedure requires in the configuring that the operating unit and the CPU are connected using a network.

Note

The description for creating a network connection between the operating unit and the CPU is contained in the online help for Step7 and WinCC flexible.

2.3 Embedding your WinCC flexible project in STEP 7

After the HMI Lite archive file has been unpacked, two WinCC flexible configurations are integrated in the S7 project.

WinCC flexible projects	Description
MP277	WinCC flexible project for MP277 10" keys & touch
OP177B color	WinCC flexible project for OP177B 6" or TP177B 6"

Table 2-1 WinCC flexible projects from HMI Lite

The configuration appropriate for the used device must be copied into the user S7 project and connected with the controller.

Note

The procedure for the above-mentioned tasks is described in the online help for Step7 and WinCC flexible.

2.4 Requirements in the CPU Properties

The clock memory byte of the PLC must be activated for HMI Lite. The HMI Lite standard blocks use the clock memory byte for coordination tasks.

Note

The configuration of the clock memory byte of the CPU is described in the online help for Step7 (HW Config).

2.5 PROFIBUS configuration with direct keys option

For safety reasons, the direct keys of the operating unit should not be used for the manual functions.

For the direct keys functionality, the OP must be coupled to the CPU using PROFIBUS DP; this functionality is not available when an MPI network is used.

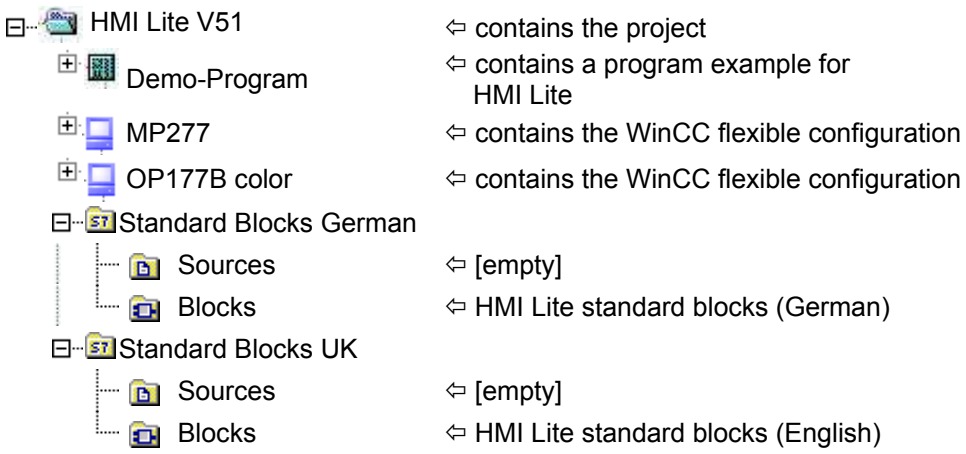
Note

The "Direct keys" option is available only when the operating unit is connected with the controller using über PROFIBUS. Further notes for configuring the PROFIBUS network and the direct keys function are contained in the "Manual Operation" section of this documentation.

2.6 PLC program blocks

2.6.1 PLC blocks from HMI Lite

All HMI Lite standard blocks are contained in the [HMI Lite] ⇨ [Blocks] folder of the source project (see display below):



The blocks must be copied into the STEP 7 user project.

Table 2-2 shows an overview of the blocks from HMI Lite.

Block numbers	Symbolic name of the blocks	Comment
DB67	DB_HMILITE_DATA	HMI Lite interface
DB68	DB_HMILITE_CONFIG	HMI Lite configuration
FC68	FC_HMILITE_BASIC	General PLC program
FC69	FC_HMILITE_MANUAL	PLC program for the operating screens
FB107 DB107	FB_HMILITE_S7G_MANUAL	Execution of manual functions using S7-GRAPH sequences
DB106	DB_HMILITE_S7G_CONFIG	Configuration for the FB107
DB69	DB_HMILITE_DEVICE_DIAG	Interface of the device diagnosis
FB96 DB96	FB_SIEM_DP_DIAG_DETAIL	Fetch the PROFIBUS configuration and the detailed data of a slave
FC96	FB_SIEM_DP_DIAG_OVERVIEW	Fetch the PROFIBUS configuration with detailed information
FC105	FC_HMILITE_PROFIBUS	Assignment of fetched diagnostic data from PROFIBUS
FC106	FC_HMILITE_MOBY	Moby diagnostic program code
FC107	FC_HMILITE_ASI	AS-i diagnostic program code
FC108	FC_HMILITE_COUNTER	"Workpiece counter" figure
FC109	FC_HMILITE_CYCLETIME	"Cycle times" figure
FC462	FC_HMILITE_DRIVE	Drive diagnostic program code
FC463	FC_HMILITE_DP_HISTORY	DP diagnostic history program code
DB463	DB_HMILITE_DP_HISTORY	DP diagnostic history interface
FB461 DB461	FB_HMILITE_SINAMICSCU3x0	HMI Lite SINAMICS CU 310/320 diagnostic
FB465 DB465	FB_SIEM_PNIODiag	PROFINET & PROFIBUS IO diagnotic

Table 2-2 PLC blocks from HMI Lite

2.6.2 Schema for calling the function blocks

Figure 2.2 shows how the HMI Lite blocks should be called (also see the Demo program). The program structure is an extract from the software guide of the "Solutions for Powertrain TRANSLINE 2000" project manual.

OB1, FC67 and FC151 are not part of HMI Lite and must be created by the machine manufacturer.

A detailed description of the functions and function blocks for HMI Lite can be found in those sections in which the appropriate screens are described.

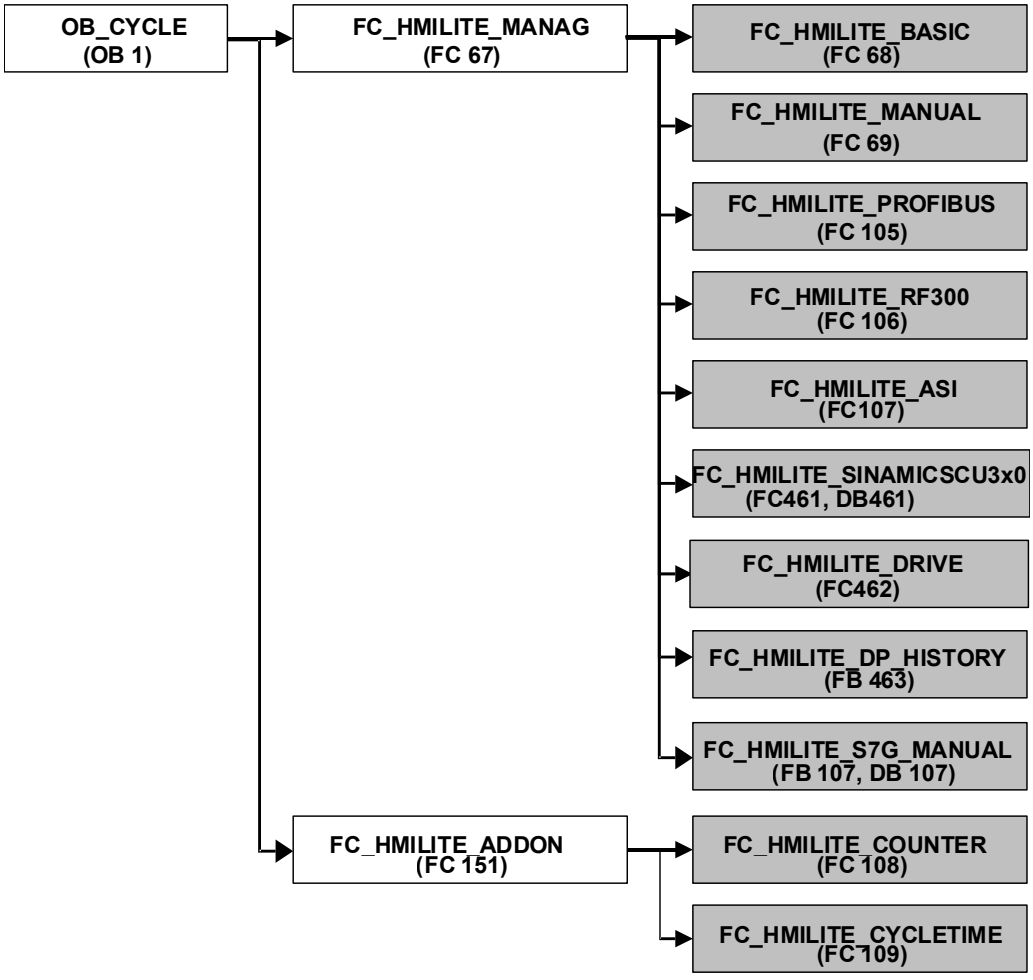


Figure 2-2 Calling hierarchy of the PLC blocks

2.7 Working with the data blocks

The two DB_HMILITE_DATA and DB_HMILITE_CONFIG data blocks form the interfaces between the HMI screens and the PLC program.

In contrast to the DB_HMILITE_DATA data block, the DB_HMILITE_CONFIG data block contains only the data for configuring the HMI masks and the PLC program.

The configuration settings for the machine must be made with Step7 in the DB_HMILITE_CONFIG.

2.7.1 Procedure for the configuring

The following table shows a simple possibility for entering the configuration data.

Step 6	Proceed as follows
1	Open the DB_HMILITE_CONFIG data block with STEP 7 in the declaration view.
2	Enter the appropriate values for the machine as initial values.
3	Change to the data view of the data block ([View] ⇨ [Data view] menu).
4	Accept the initial vlaues as current values using the ([Edit] ⇨ [Initialize data block]) command.
5	Save the data block ([File] ⇨ [Save] menu).
6	Load the data block into the controller ([Target system] ⇨ [Load] menu).
7	Testing the changed configuration.

Table 2-3 Editing the DB_HMILITE_CONFIG data block

Note

A detailed description for working with data blocks is contained in the Step7 online help.

■

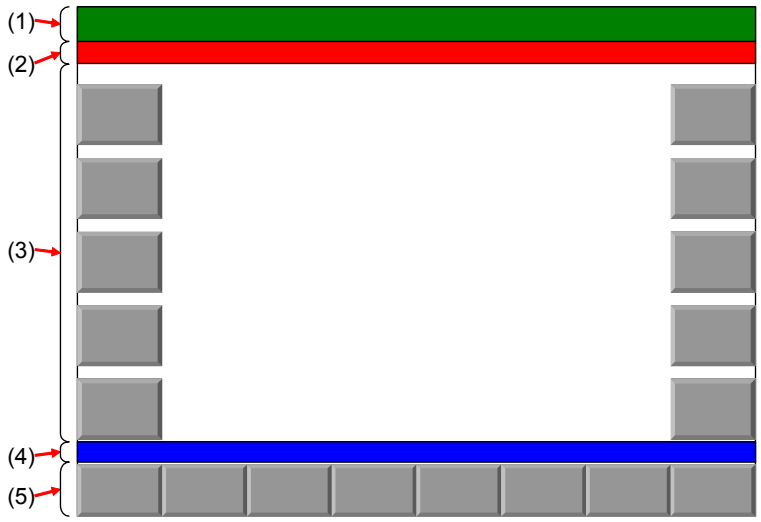
Notes

3

3 Global Settings and Functionality

3.1 Layout of the screens and basic screen elements

All screens have a standard structure (see Figure 3-1).



(1)	Header information - plant status
(2)	Message line for alarms and messages
(3)	Working area with vertical softkeys (optional)
(4)	Line for operator notes
(5)	Horizontal softkeys with screen-dependent functions

Figure 3-1 Screen elements

(1) Header

The upper area of each screen contains the header. It contains significant status information, such as operating mode, initial state, etc. This area also contains the message line for alarms and messages.

The header can be configured in two different types of representation. Whereas one of the representation types shows the status information as text, the other representation type shows the status information as graphic elements. Further details about the header are contained in Chapter 4 of this manual.

(2) Message line for alarms and messages

The message line is part of the header and so is visible in each screen. All fault and operational messages are displayed with number, time, status and message text. By default, the most recently occurring message is always displayed. However, the message settings in WinCC flexible can be changed so that the oldest associated message is always displayed.

(3) Working area

The working area shows the screen-dependent screen elements.

(4) Operator notes

Notes for the machine operation can be displayed in this line for the machine operator. The operator note is output as a single line of text.

(5) Horizontal softkeys

The horizontal softkeys are used primarily to select other screens and are always located in the lower screen area. Other than their use to call other screens, the function keys are used to scroll within the selected screen (e.g. page up / page down in the operator screens) or to activate special functions (e.g. to activate and deactivate the manual operation in the "PROFIBUS Diagnosis" screen).

By default, the menu structure is based on a two-level structure (main menu and submenu level). A third menu level is used only when a grouping of inter-related screen forms is required.

3.2 Menu structure

Although the menu structure of the HMI Lite standard project has the following form, it can at anytime be customized by the user for the specific project.

The menu structures differ according to the size of the operator panels:

Operator panels with 10" screen

Operator panels with 6" screen

The menu structures for these variants depend on the number of function keys of the devices.

The following structure figures show the form of the associated standard menus for the two operator panels:

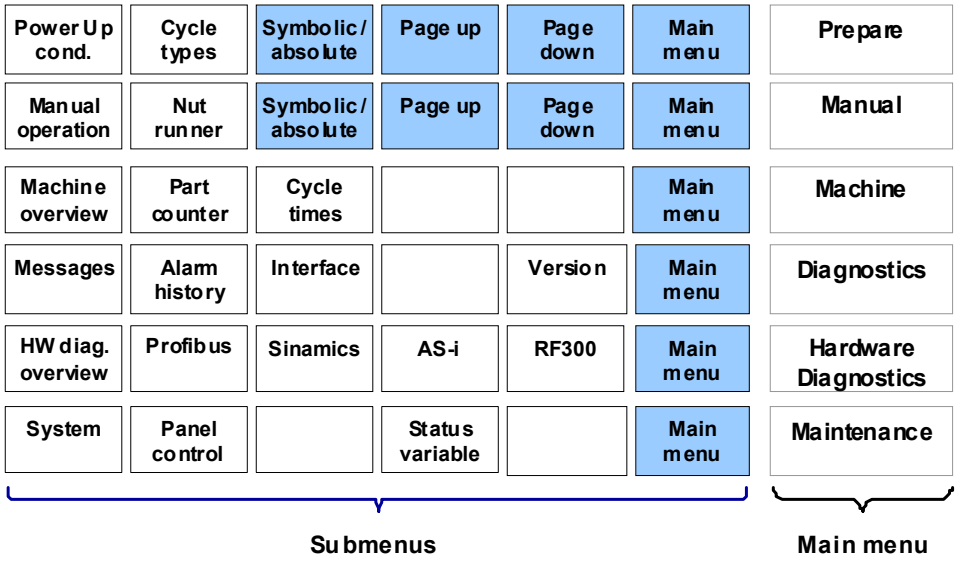


Figure 3-2 Structure of the menu for operator panels with 6" screen display

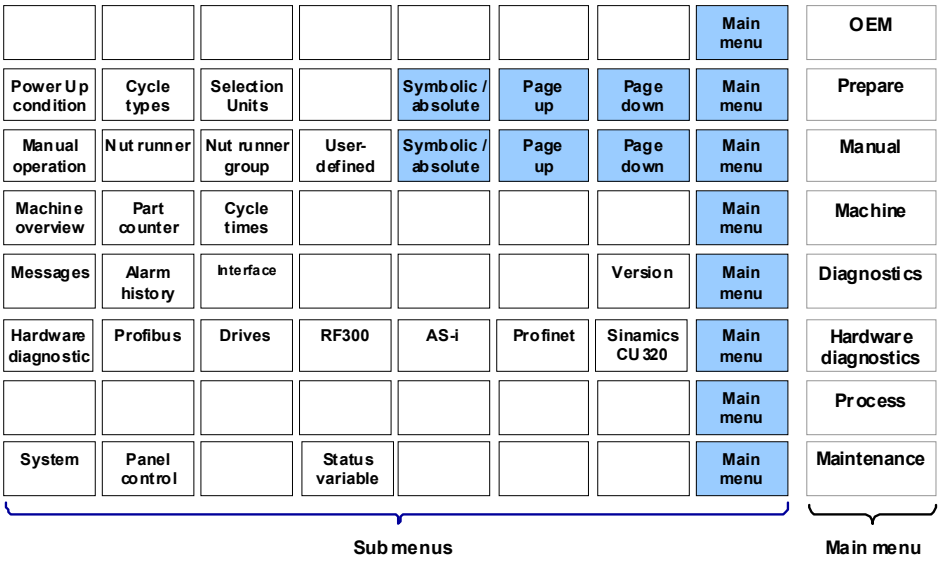


Figure 3-3 Structure of the menu for operator panels with 10" screen display

Screen forms of the machine manufacturer

The machine manufacturer should give the operator a graphic overview of the associated machine or plant in the HMI Lite "OEM ⇨ Plant overview" main screen. From this screen, the horizontal softkeys can be used to change to one of the six or eight main menus.

The 10-inch operator panels provide the machine manufacturer with the two main menus "OEM" and "Process" in which the machine manufacturer can include machine-specific screens and functions.

The limited number of softkeys for the 6" variant means only single screens can be included in the existing main menus. It is possible to create a third menu level.

Note

The menu structure shown here is the standard menu structure for HMI Lite. The menu structure can be customized for specific projects.

Navigation and function keys:

The gray-shaded buttons are navigation and function keys assigned to the individual screens in the corresponding submenus.

Click the "Main menu" button to return from the current menu to the main menu ("Plant overview" screen). The "Return" button is configured in the third menu level to return to the second menu level.

The "Page up"/"Page down" keys and the "Absolute/Symbolic" toggle keys required in the operating screens are described in Chapter 5.

3.3 "Template" screen

The "SS_90_Template" screen is used to add machine-specific screens while retaining the screen layout and the menu structure.

This results in the following procedure:

Duplicate the "SS_90_Template" screen

Rename the screen

Configure the screen

Include the screen in the menu structure

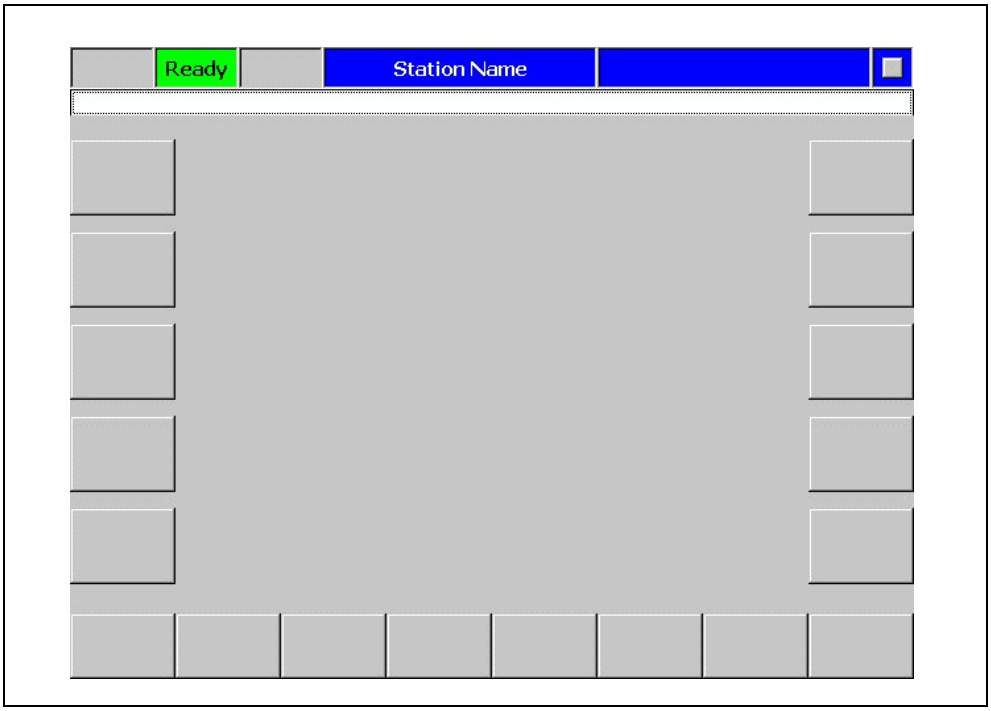


Figure 3-4 "SS_90_Template" screen for creating your own screens

3.4 Designation conventions

All WinCC flexible elements, such as screens, variables, graphics and symbol lists have been named using uniform designation conventions.

The designation structure must provide the following information:

Who created the associated element?

Who changed the element?

How are the individual elements linked with each other?

In other words: All WinCC flexible elements that can be changed by the user (configuring) are designated with "SO_". When the elements in WinCC flexible are sorted according to name, these elements appear at the start of the list.

In addition, the designations can be used to determine all elements assigned to a screen.

Designation convention syntax

All WinCC flexible elements, such as screens, variables, graphics and symbol lists must be named using these uniform designation conventions (see Table 3-1).

Name structure of the screen elements: **AB_XX_Name**	
Icon	Description
A	Who created the associated screen element? S: Siemens O: OEM (machine manufacturer) C: Customer
B	Who should change the screen element? S: Siemens O: OEM (machine manufacturer) C: Customer
XX	Assignment of the screen elements to each other (e.g. 62: means all elements of the "Workpiece counter" screen)
Name	Designation of the screen element (e.g. "PartCounter")

Table 3-1 Syntax of the designation convention for screen elements in Pro Tool

Example

S	S	_	51	_	ASIDiagnostic

The name of the screen form is "ASI-Diagnosis".
The number of the screen element is 51.
The screen form may only be changed by Siemens.
The screen form was created by Siemens.

All other elements only used in the "ASIDiagnostic" screen, such as variables or symbol lists, also have the identification 51.

E.g.: Variable: SS_51_ASIFlags
Variable: SS_51_SelectedMaster
Text list: SO_51_SelectedMasterIndex

Global screen elements (identification 00)

All screen elements not uniquely assigned to a specific screen have the identification 00 (e.g. the variables or symbol lists used in the header).

Screen element groups

In some cases, screen elements, e.g. variables, are used together by complete screen groups. A common group identification is then assigned to such screen elements.

For example, all screen elements used by all operating screens have the identification 10. Screen elements used only for a specific operating screen have the associated identification of the corresponding screen to which they are assigned (e.g. for the "SS_11_ManualMovement" screen).

3.5 Clock memory byte of the controller

The 8 bits of the clock memory byte change their binary value cyclically in the pulse pause ratio 1:1 with a period of 0.1 second to 2 seconds.

The clock memory byte is used by the HMI Lite blocks for internal, time-based trigger events (e.g. monitoring of the controller <> OP communication).

The clock memory byte must be transferred as input parameter to the "FC_HMILITE_BASIC". The FC_HMILITE_ BASIC function creates pulses of the individual clock signals and cyclically updates the variables of the data blocks.

3.6 PLC system time

3.6.1 System timer

To avoid using any timer of the CPU, all time functions within the HMI Lite blocks are realized using the CPU system time. The system time is fetched with the SFC64 and further processed.

3.6.2 System time and date

The SFC 1 ("READ_CLK") is used to read the date and time of the PLC from the "FC_HMILITE_BASIC" block. The system time will be written in the "AREAPOINTER.DATE_TIME_PLC [1..12]" data area of the "DB_HMILITE_DATA".

The "Controller date/time" area pointer is read from the panel with an acquisition cycle of 120 seconds and a synchronization with the OP clock is performed. If, for some reason, the synchronization of the clocks has been disabled, the "Controller date/time" area pointer must be deleted from the WinCC flexible configuration.

Note

The acquisition cycle of the "PLC date and time" area pointer should not be chosen too small, because it affects the communications capability of the operator panel. By default, 120 seconds (2 minutes) are set for the acquisition cycle.

The area pointer is located in WinCC flexible in the sub-tab of the Communication > "Connections" screen.

3.7 Identification of the selected screen

The information which screen is selected on the operator panel is made available in the WinCC flexible "SS_00_ScreenIdentification" variable. For the screen construction, the corresponding value is written in the variable; for screen removal, the variable is set to zero.

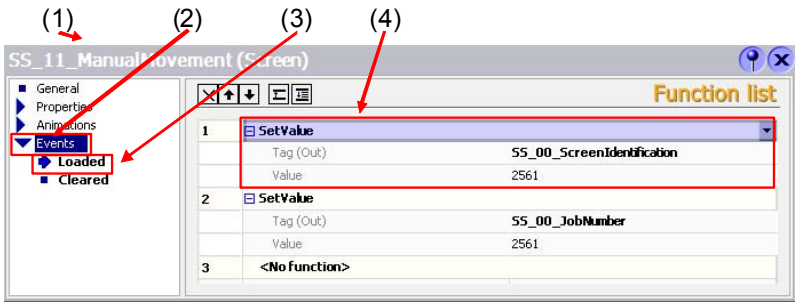
To keep the cycle time of the controller as small as possible, the program code for a specific screen should be executed only when the corresponding screen is selected.

The WinCC flexible "SS_00_ScreenIdentification" variable is defined as follows:

Tag:	SS_00_ScreenIdentification
Format:	WORD
PLC address:	"DB_HMILITE_DATA".GLOBAL.SCREEN_ID

Configuring screen events

For details of configuring screen events, see Figure 3-5.



(1) Properties dialog of a screen
(2) Register [events]
(3) Event [established] when the function is initiated
(4) Function [SetValue] to be performed

Figure 3-5 Configuring the screen event to identify the selected screen

Codes to identify the individual screens

WinCC flexible screen number	Designation of the system screen	Code to identify the screen form			
		High byte	Low byte	[dec.]	[hex.]
General screen forms					
1	SO_01_MainScreen	01	1	257	0x0101
2	SS_02_Status	01	2	258	0x0102
4	SS_04_SystemScreen	01	4	260	0x0104
5	SS_05_PanelControl	01	5	261	0x0105
6	SS_06_Version	01	6	262	0x0106
"Manual" screen forms					
11	SS_11_ManualMovement	10	1	2561	0x0A01
12	SS_12_PowerUpCondition	10	2	2562	0x0A02
13	SS_13_Unit	10	3	2563	0x0A03
14	SS_14_NutRunner	10	4	2564	0x0A04
15	SS_15_NutRunnerGroup	10	5	2565	0x0A05
16	SS_16_CycleTypes	10	6	2566	0x0A06
17	SS_17_UserDefine	10	7	2567	0x0A07
Simodrive 611 diagnosis					
21	SS_21_DriveStatus * SS_21_DriveStatusControlWord **	20	1	5121	0x1401
22	SS_22_DriveStatusStatusWord **	20	2	5122	0x1402
23	SS_23_DriveSignals * SS_23_DriveSignalsPositionStatusWord **	20	3	5123	0x1403
24	SS_24_DriveSignalsPositionSignals **	20	4	5124	0x1404
25	SS_25_DriveFaultsandWarnings * SS_25_DriveFaults **	20	5	5125	0x1405
26	SS_26_DriveWarnings **	20	6	5126	0x1406
Alarms and messages					
31	SS_31_Alarm	30	1	7681	0x1E01
32	SS_32_AlarmHistory	30	2	7682	0x1E02
35	SS_35_ProAgentAlarm ***	30	5	7685	0x1E05
36	SS_36_ProAgentUnit ***	30	6	7686	0x1E06
37	SS_37_ProAgentDetail ***	30	7	7687	0x1E07
Hardware diagnostics					
41	SS_41_HardwareDiagnostics*	40	1	10241	0x2801
42	SS_42_ProfibusOverview	40	2	10242	0x2802
43	SS_43_ProfibusDiagnosticsDetail * SS_43_ProfibusStandardDiagnosticsDetail **	40	3	10243	0x2803
44	SS_44_ProfibusExtendedDiagnosticsDetail **	40	4	10244	0x2804
45	SS_45_ProfibusDiagnosticHistory	40	5	10245	0x2805
51	SS_51_ASIDiagnostic * SS_51_ASIDiagnosticMaster **	50	1	12801	0x3201

WinCC flexible screen number	Designation of the system screen	Code to identify the screen form			
		High byte	Low byte	[dec.]	[hex.]
52	SS_52_ASIDiagnosticSlave **	50	2	12802	0x3202
53	SS_53_ASIDiag_Monitor	50	3	12803	0x3203
54	SS_54_ASIDiag_Monitor_Kreis_1 **	50	4	12804	0x3204
55	SS_55_ASIDiag_Monitor_Kreis_2**	50	5	12805	0x3205
57	SS_57_MobyDiagnostics * SS_57_MobyDiagnosticsStatus **	50	7	12807	0x3207
58	SS_58_MobyDiagnosticsError **	50	8	12808	0x3208
Machine information					
61	SO_61_MachineOverview	60	1	15361	0x3C01
62	SS_62_PartCounter * SS_62_PartCounterOverall **	60	2	15362	0x3C02
63	SS_63_PartCounterSpecific **	60	3	15363	0x3C03
65	SO_65_CycleTimes	60	5	15365	0x3C05
67	SS_67_Interlocks	60	7	15367	0x3C07
PNIO diagnostics					
71	SS_71_StationOverview1 *	70	1	17921	0x4601
72	SS_72_StationOverview2 *	70	2	17922	0x4602
73	SS_73_StationOverview3 *	70	3	17923	0x4603
74	SS_74_StationOverview4 *	70	4	17924	0x4604
75	SS_75_StationDetail *	70	5	17925	0x4605
76	SS_76_DiagnosticRepeater *	70	6	17926	0x4606
77	SS_77_HistoryTrigger *	70	7	17927	0x4607
78	SS_78_Legend *	70	8	17928	0x4608
79	SS_79_PNIODiagInfo *	70	9	17929	0x4609
SINAMICS CU3x0 diagnostics					
81	SS_81_ControlStatusword * SS_81_StatusControlword **	80	1	20481	0x5001
82	SS_82_StatusControlword **	80	1	20482	0x5001
83	SS_83_EPOSStatus * SS_83_EPOSStatusword **	80	1	20483	0x5001
84	SS_84_EPOSPositioning **	80	1	20484	0x5001
85	SS_85_FaultsAndWarnings * SS_85_Faults **	80	1	20485	0x5001
86	SS_86_Warnings **	80	1	20486	0x5001
Customer/project-specific screen forms					
70-79	Customer-specific screens ***	70	-	-	-
80-89	Customer-specific screens ***	80	-	-	-
OEM-specific screen forms					
90	OEM-specific screens (template) ***	90	-	-	-

WinCC flexible screen number	Designation of the system screen	Code to identify the screen form			
		High byte	Low byte	[dec.]	[hex.]
	* These screens are available only for the 10" variant. ** These screens are available only for the 6" variant. *** Placeholder for project- or customer-specific screens (not available in the standard version). **** These screens are not active for 10” variant.				

Table 3-2 Identification code of the individual screens

3.8 HMI Lite job mailbox

The job mailbox forms the primary interface between the HMI system and the control program for initiating an operator action.

Structure

The job mailbox has a defined length of four words. The structure is shown in the following table:

Address	Data type	Name	Description
n+0	WORD	JOBNUMBER	Job number
n+2	WORD	PARAMETER_1	1st parameter of the job
n+4	WORD	PARAMETER_2	2nd parameter of the job
n+6	WORD	PARAMETER_3	3rd parameter of the job

Table 3-3 Structure of the job mailbox

The first word always contains the job number. Depending on the associated control job, up to three parameters can be specified.

Number of the job and parameters

The job number corresponds to the screen identification number. This enables all actions initated by a specific screen to be determined exactly by the screen identification. The parameters specify the action to be performed. Details can be found in the descriptions of the associated screens.

Monitoring the connection

Because only status changes for keys and buttons can be transferred to the controller, the connection between the operator panel and the controller must be monitored for correct operation. This monitoring is performed using the sign-of-life bit of the operator panel from the "Coordination" area pointer. The sign-of-life bit is inverted by the operator panel in one second intervals.

The "FC_HMILITE_MANUAL" function checks cyclically whether the sign-of-life bit has been inverted to determine whether the connection to the operator panel still exists. If no inversion of the sign-of-life bit has been determined during a time interval, the job mailbox will be cleared. The time interval is defined by the following parameters:

DB_HMILITE_CONFIG.MANUAL_COMMON. SCREEN_ACTIVE_TIME

Important

Because the sign-of-life bit is not a real-time signal, depending on the data traffic on the network and the number of processes running on the operator panel, it can take longer than one second before the signal has changed its status. To ensure shorter response times and a faster shutdown of the manual operations, we recommend that the function keys of the operator panel are used as PROFIBUS DP direct keys.
The machine manufacturer is responsible for the reliable execution of the manual operation.

"Coordination" area pointer

The controller can use this data area to query the status of the operator panel, e.g. startup of the operator panel, current operating mode and ready for communication.

Structure of the "Coordination" area pointer

The structure of the "Coordination" area pointer with the length of one word:

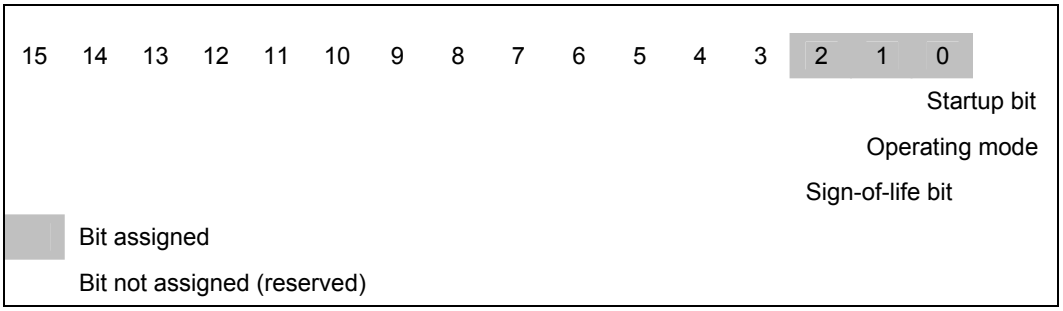


Figure 3-6 Layout of the "Coordination" area pointer

3.9 FC_HMILITE_BASIC

The basic functions of HMI Lite are realized using the "FC_HMILITE_BASIC" function. This FC is responsible for the coordination of the interface DBs and HMI screens.

Figure 3-7 shows the parameters of the FC; Table 3-4 describes its parameters.

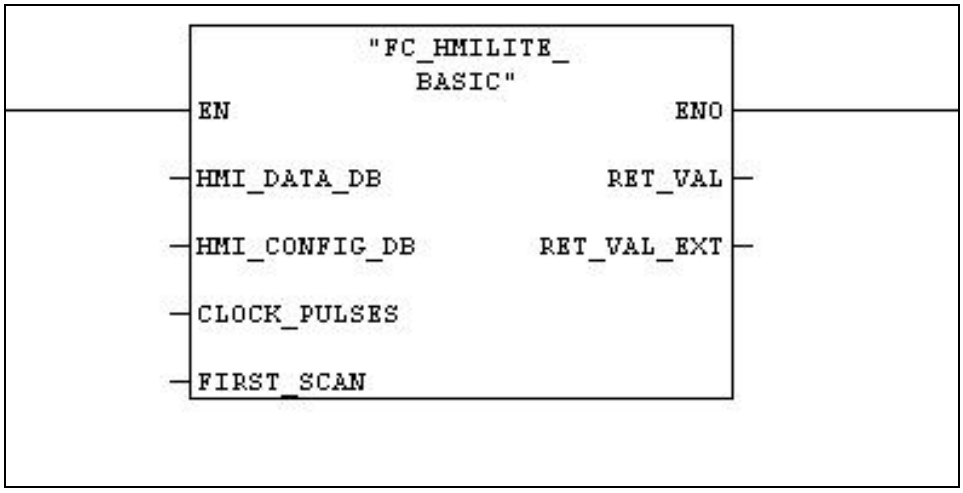


Figure 3-7 Interface of the "FC_HMILITE_BASIC" function

FC_HMILITE_BASIC parameters

Name	Type	Default	Example	Description
HMI_DATA_DB	INT	67	67	Number of the HMI runtime data block
HMI_CONFIG_DB	INT	68	68	Number of the HMI configuration data block
CLOCK_PULSES	BYTE	MB1	MB1	Clock memory byte, configured in the object properties of the CPU (HW Config)
FIRST_SCAN	BOOL	M0.5	M0.5	Restart flag, 1 – signal for the first cycle after CPU startup
RET_VAL	WORD	-	TEMP VARIABLE	Error message of the "FC_HMILITE_BASIC" function: W#16#80C1 => internal SFC1 (READ_CLK) call with error
RET_VAL_EXT	INT	-	TEMP VARIABLE	Error message from the SFC1 "READ_CLK", see online help for the SFC1

Table 3-4 Description of the FC_HMILITE_BASIC parameters

3.10 Connection of several operator panels to a controller

If several HMI Lite operator panels are to be connected with a controller, a new DB interface must be provided in the controller for the second and each additional operator panel. This requires the duplication of the HMI Lite DBs and the modification of the numbers in Step7 and WinCC flexible.

The following table lists the steps required to change the DB numbers.

Step	Proceed as follows
1	Duplicate the DB_HMILITE_DATA (DB67) as runtime interface for the second operator panel.
2	Duplicate the DB_HMILITE_CONFIG (DB68) for the configuration of the second operator panel.
3	The FC_HMILITE_BASIC (FC68) must be called once in the program for each operator panel. The "HMI_DATA_DB" and "HMI_CONFIG_DB" input parameters must be supplied with the appropriate DB numbers.
4	The FC_HMILITE_MANUAL (FC69) must be called once in the program for each operator panel. The "HMI_DATA_DB" and "HMI_CONFIG_DB" input parameters must be supplied with the appropriate DB numbers.
5	The number of the DB_HMILITE_DATA (DB67) is parameterized in the WinCC flexible "SO_00_DBNumberData" variable. This number must be changed in the corresponding WinCC flexible configuration. The new number of the duplicated DB_HMILITE_DATA must be entered as start value in the properties dialog of the WinCC flexible "SO_00_ DBNumberData" variable for the "initial settings".
6	The number of the DB_HMILITE_CONFIG (DB68) is parameterized in the WinCC flexible "SO_00_DBNumberData" variable. This number must be changed in the corresponding WinCC flexible configuration. The new number of the duplicated DB_HMILITE_CONFIG must be entered as start value in the properties dialog of the WinCC flexible "SO_00_DBNumberConfiguration" variable for the "initial settings".
7	The WinCC flexible area pointers with the DB67 (DB_HMILITE_DATA) absolute address must receive the duplicated DB_HMILITE_DATA as new absolute address.
8	The user-specific fault and operational messages must be assigned new addresses, unless the same messages should be displayed on both operator panels.

Table 3-5 Changing the numbers of the HMI Lite standard DBs.

Because at any one time only a single operator panel can access the hardware diagnosis, an operator panel change must be configured for this purpose.

! Important

Manual operations performed from both operator panels must be mutually interlocked. This remains the responsibility of the user.

■

Notes

4 Header and Operator Information

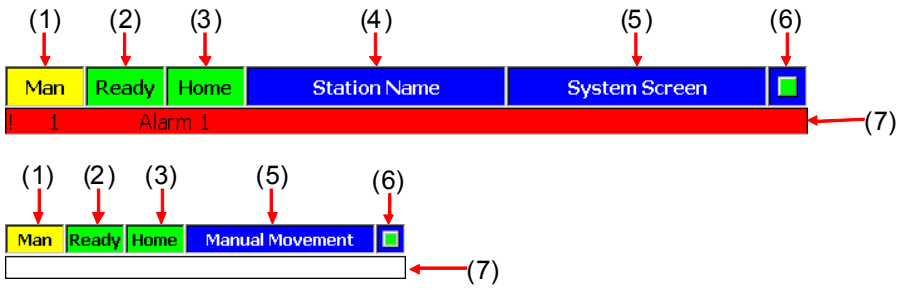
4

4.1 Header

4.1.1 Layout of the header

The HMI Lite header shows the operator general information about the machine status.

Figure 4-1 shows the structure of the header in HMI Lite.



(1)	Display of current operating mode
(2)	Plant status
(3)	Display of the basic position
(4)	Header – text field 1
(5)	Header – text field 2
(6)	CPU sign-of-life display
(7)	Alarm and message line

Figure 4-1 Layout of the header for the 6" and 10" operator panel

4.1.2 Display of current operating mode

The currently selected operating mode is displayed.

By default, the following operating modes are defined:

Display	Operating mode
[empty]	No operating mode selected
Auto	Linked mode
Cycle clock	Single mode
Step 6	Single-step mode
Manual	Setup

Table 4-1 Display of the operating modes in the header

Each of the operating modes listed above can be subdivided as follows:

- selected but not active (gray background)
- selected and active (green or yellow background)

"No operating mode" is displayed when:

- The operating mode selection switch is in an undefined position
- The operating mode is selected using keys but no key has been pressed

Operating modes that are selected but not active will be displayed with a gray background. Operating modes that are active will be displayed with a green or yellow background.

Operating mode selected		Operating mode activated	
Text	Icon	Text	Icon
Auto		Auto	
Cycle		Cycle	
Step		Step	
Man		Man	

Table 4-2 Selected and activated operating modes

"Operating mode selection" runtime interface

The associated operating mode is displayed using the interface bits in the DB_HMILITE_DATA data block, see Table 4-3 (the operating mode is displayed when the interface bit = "TRUE").

Text	Icon	Interface	Type

Auto		DB_HMILITE_DATA.HEADER.MODE.AUTOMATIC	BOOL
Cycle		DB_HMILITE_DATA.HEADER. MODE.CYCLE	BOOL
Step		DB_HMILITE_DATA.HEADER. MODE.STEP	BOOL
Man		DB_HMILITE_DATA.HEADER. MODE.MANUAL	BOOL

Table 4-3 – Display of the current operating mode

If no or several interface bits have the "TRUE" status, the "No operating mode" status will be displayed.

"Operating mode selected/active" runtime interface

If the "DB_HMILITE_DATA.HEADER. MODE.ACTIVE" interface bit is set to "TRUE", the operating mode will be displayd as active.

Configuration

No configuring required.

4.1.3 Status display

The following states are possible for the machine status display:

Text	Icon	Meaning	Description
Ready		Ready to run	No fault message or operation message is present.
Warn		Warning	One or more operation messages are present.
Alarm		Fault	One or more fault messages are present.

Table 4-4 – Status display

Runtime interface

The "Status display" is controlled with the following status bits in the DB_HMILITE_DATA data block:

Meaning	Interface	Type
Ready to run	--- (if no additional status bit has the status "1"– status = ready)	BOOL
Warning	DB_HMILITE_DATA.HEADER.STATUS.WARNING	BOOL
Interrupt	DB_HMILITE_DATA.HEADER.STATUS.ALARM	BOOL

Table 4-5 – Status display

By default, the status bits are not linked with other variables or objects (e.g. with alarm or message bits).

Configuration

No configuring required.

4.1.4 Display of the initial state

The following states are possible for the initial state display:

Text	Icon	Meaning	Description
		Empty	The machine is not in the "initial state"
Home		Initial state	The machine is in the "initial state"

Table 4-6 – "Initial state" display

Interface

The "initial state" display is made using the following bit in the DB_HMILITE_DATA data block:

Meaning	Interface	Type
Empty	---	
Initial state	DB_HMILITE_DATA.HEADER.POSITION.HOME	BOOL

Table 4-7 – "Initial state" display

The "initial state" status is displayed when the bit is "TRUE".

Configuration

No configuring required.

4.1.5 Text boxes

The user has two text fields available to display machine-specific text.

Runtime interface

The text is controlled using two variables in "DB_HMILITE_DATA". The text assigned to the value of the variable of the WinCC flexible – text list is displayed as text.

Address:	DB_HMILITE_DATA. HEADER.TEXTINDEX_1 DB_HMILITE_DATA. HEADER.TEXTINDEX_2
Format:	WORD
Range of values:	1..
Default setting:	W#16#0

Configuration

Text list	SO_00_HeaderText1 SO_00_HeaderText2
Display Format	Text Decimal
Value	Text
1	[Text to be displayed]
…	…

The "SO_00_HeaderText2" text list is preconfigured so that the screen name of the selected screen is displayed.

This requires that the screen number of the selected screen is transferred from the WinCC flexible "Screen number" area pointer into the "DB_HMILITE_DATA.HEADER.TEXTINDEX_2" variable.

For example:

```
L   DB_HMILITE_DATA.AREAPOINTER.SCREEN_NUMBER.SCREENNUMBER
T   DB_HMILITE_DATA.HEADER.TEXTINDEX_2
```

This is programmed in the FC67 in the Demo program.

The machine manufacturer must extend the "SO_00_HeaderText2" text list if new screens are to be added to the WinCC flexible configuration.

4.1.6 Sign-of-life of the CPU

The sign-of-life in the header displays the operating mode of the PLC. A periodic flashing in intervals of approximately one second indicates that the controller is in the "RUN" operating mode.

The "STOP" operating mode is indicated by a red rectangle.

Field	Interface
Flashing  or 	The PLC is in the "RUN" operating mode and the Panel – PLC communication occurs.
Static  or 	The communication with the PLC has been interrupted.
	The PLC is in the "STOP" operating mode

Table 4-8 – Display of the sign-of-life of the CPU

Runtime interface

Field	Interface
Sign-of-life	DB_HMILITE_DATA. HEADER.WATCHDOG

Table 4-9 – Display of the sign-of-life of the CPU

4.1.7 Changing the display of the status signals in the header

The machine status display in the header can be displayed as a symbol or as text.

The "SS_00_SelectHeader" screen provides both views.

The replacement of the header elements in the permanent window for WinCC flexible with the corresponding elements from the "SS_00_SelectHeader" screen can be used to change the display of the status signals.

4.2 Operator information

The operator information is a text output field used to display information for the operator. The text display is located above the horizontal buttons.

Note

For space reasons, the operator information field on 6-inch operator panels (OP177B & TP177B) is present only in the manual operation screens.

Runtime interface

Two runtime variables are used to control the dynamic behavior of the text output field.

The first variable, "DB_HMILITE_DATA. GLOBAL.PROMPT.INDEX", is used to select which text from the WinCC flexible text list is to be displayed.

The second variable, "DB_HMILITE_DATA.GLOBAL. PROMPT.ATTRIBUT", is linked with the attribute of the text output field. The variable so controls the color change of the flashing of the text output field.

Address:	DB_HMILITE_DATA. GLOBAL.PROMPT.INDEX
Format:	WORD
Range of values:	1..
Default setting:	W#16#0

Address:	DB_HMILITE_DATA. GLOBAL.PROMPT.ATTRIBUT
Format:	WORD
Range of values:	1..
Default setting:	W#16#0

Configuration

The WinCC flexible SO_00_OperatorPrompt text list contains the text to be displayed in the text field for operator information.

Text list		**SO_00_OperatorPrompt**
Display		Text
Type		Decimal
Value	[Text number]	[Text to be displayed]
…	…	…

■

Notes

5 Manual Operating

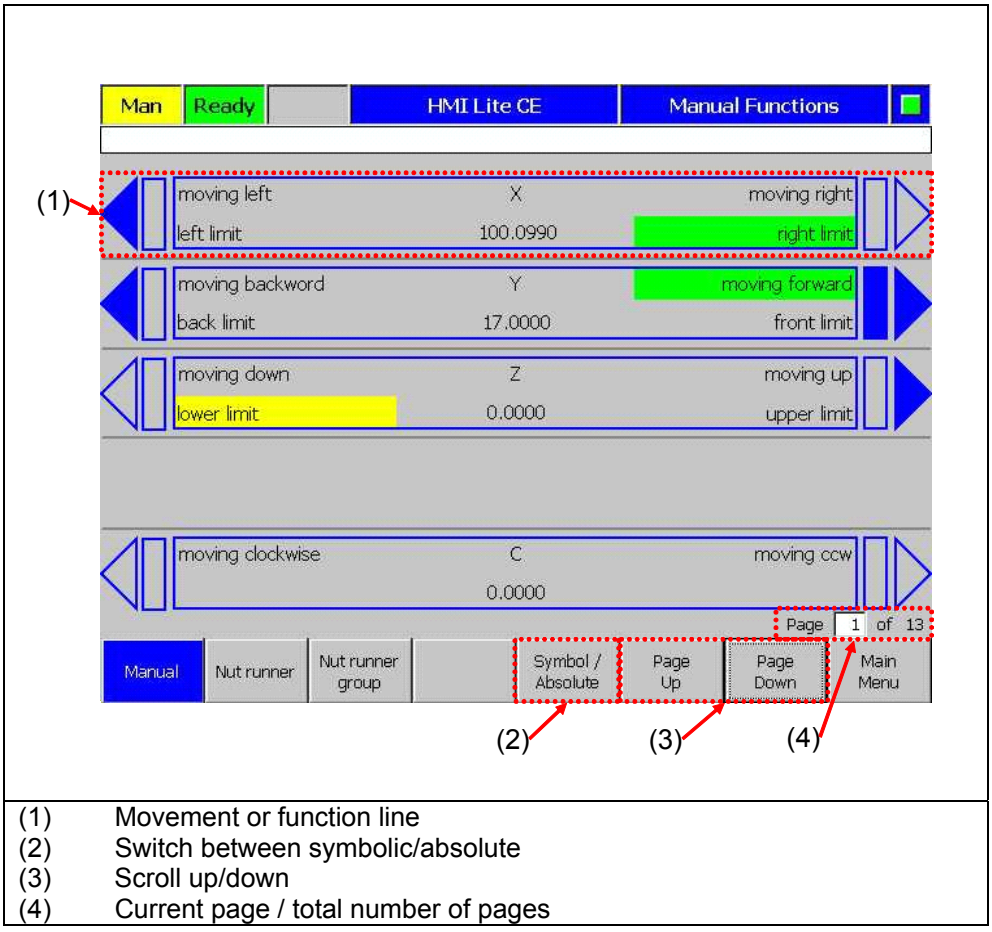
5

5.1 Overview

5.1.1 Layout and basic functionality of the manual operating screens

The operator can use the manual operating screens to perform movements, activate/deactivate machine elements, select cycle type and perform other actions for which a selection must be made.

All screens from the manual operation area have the same general structure (see Figure 5-1).



(1)	Movement or function line
(2)	Switch between symbolic/absolute
(3)	Scroll up/down
(4)	Current page / total number of pages

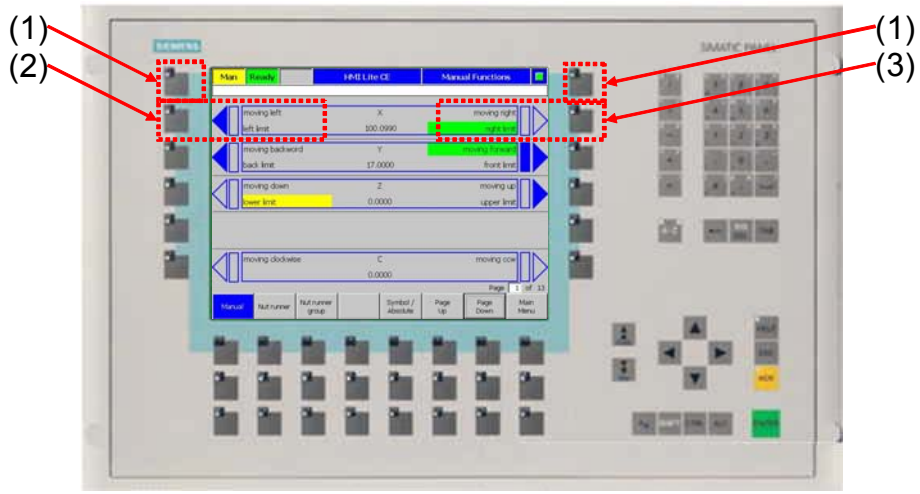
Figure 5-1 Layout of the manual operating screens

Movement and function line

Each movement and each function is represented by a line.

Every movement or function can be performed in two directions, such as input/output, open/close, up/down, forwards/backwards. One direction of the movement or function is shown on the left-hand side of the screen and the other direction is shown on the right-hand side of the screen.

The two keys to the immediate left or right of the function line are assigned to each movement as shown in Figure 5-2. These keys are used to activate the associated movement or function.



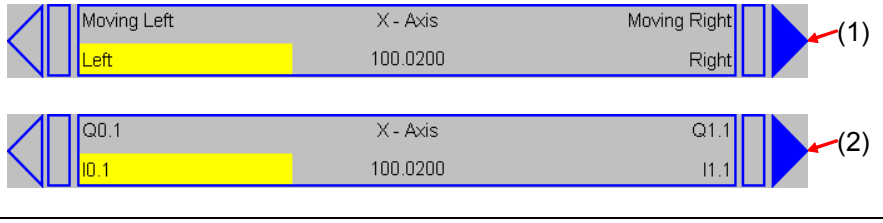
(1)	The two upper vertical softkeys are not assigned to any function line.
(2)	This button is assigned to the left-hand function in the first line.
(3)	This button is assigned to the right-hand function in the first line.

Figure 5-2 Assignment to the corresponding function lines

Absolute and symbolic view

The "symbolic/absolute" toggle key can be used to switch between the symbolic and the absolute designation of the inputs and outputs (e.g. I1.0, O1.0) that are assigned to the corresponding movements/functions (see Figure 5-3).

The represention in absolute form is displayed only for a parameterizable time. After this time has expired, the HMI Lite system switches automatically to the symbolic representation.



(1) Symbolic view
(2) Absolute view

Figure 5-3 Manual operating screens - absolute and symbolic view

Scrolling

A scroll function can be used to fetch all configured actions for a maximum display of five (10" device) or three (6" device) movements/functions per page.

When the scroll function is performed, the complete screen is always opened (all displayed function lines will be replaced by the function lines present on the next page).

When the last page is reached and the page-up key is pressed, the display changes back to the first page. Similarly, when the first page is reached and the page-down key is pressed, the display jumps to the last page.

You cannot change the screen (the page is locked) while a movement or function is being performed.

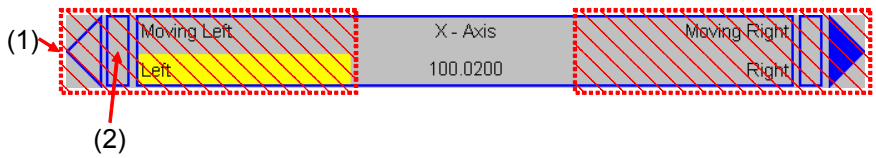
Selection and display of the page number

The current page number and the total number of pages are displayed at the bottom in each screen. The page number field, which is an input field, can be used to directly select a page by entering the appropriate page number from the keyboard or keypad.

Touch panel support

Other than the keys at the immediate left and right of each movement or function, the screens for the manual operation also support touch panel operation (for operator panels with touch functionality).

As shown in Figure 5-4, each movement/function can be initiated by touching the corresponding function in the appropriate areas.



(1)	Touch-sensitive area (button not visible)
(2)	Button activated / display pressed

Figure 5-4 Manual operating screens – touch panel support

To prevent the inadvertent initiation of a movement, the system is designed so that to initiate a movement, it must first be selected by touching the appropriate touch-sensitive surface. The selection of the movement is confirmed by a flashing square next to the movement arrow (see number (2) in Figure 5-4).

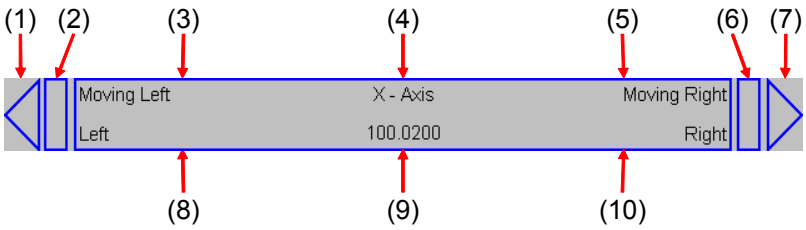
Once the movement has been confirmed, the movement can be initiated by subsequently touching the touch-sensitive area. As confirmation, the square next to the triangle becomes dark blue.

The function remains selected until one of the following events occurs:

- Another movement is selected.
- Scroll to another page.
- Select another screen.
- The sign-of-life monitoring deactivates the movement.

5.1.2 Elements of the movement/function line

Each movement line consists of the basic elements shown in Figure 5-5:



(1)	Executability, left-hand side
(2)	Key/button activated, left-hand side
(3)	Will be executed, left-hand side
(4)	Name of function
(5)	Will be executed, right-hand side
(6)	Key/button activated, right-hand side
(7)	Executability, right-hand side
(8)	Final status (end position), left-hand side
(9)	Position (optional)
(10)	Final status (end position), right-hand side

Figure 5-5 Manual operating screens - elements of the individual line

Designation (4)

The "Designation" object is the title for the movements/functions. Scrolling the screen updates the designations of the functions with the associated description of the associated line on the current page.

The text items are configured in text lists for WinCC flexible. The "Designation" element does not have any runtime interface.

Position (9)

The "Position" object can be used to display a numeric position value. The position field is optional and can be hidden for each movement or function.

Executability (1 and 7)

The "Executability" object indicates whether or not a movement can be performed.

- If the motion can be executed, the triangle is filled dark blue.
- If the movement cannot be performed because it is disabled or interlocked (e.g. target position reached), the triangle is displayed as an unfilled contour.

The status information must be supplied as binary signals via the runtime interface.

Final state (end position) (8 and 10)

The "Final state" element represents movement- or function-specific end positions in both directions (e.g. left/right, up/down, open/closed).

- If a target position has not yet been reached, the associated text of this position appears on a gray background.
- If the target position has been reached, the appropriate text is displayed with a yellow (left-hand side) or green (right-hand side) background.

Various text items for the symbolic and the absolute view can be displayed for each "Final state" element. The text items are configured in text lists for WinCC flexible.

The status information must be supplied as binary signals via the runtime interface.

Execution (3 and 5)

The "Execution" element shows the status of the output from which the associated movement/function is controlled.

- If the output is deactivated, the associated text appears with a gray background.
- If the output is activated, the appropriate text is displayed with a yellow (left-hand side) or green (right-hand side) background.

Various text items for the symbolic and the absolute view can be displayed for each "Execution" element. The text items are configured in text lists for WinCC flexible.

The status information must be supplied as binary signals via the runtime interface.

Key/button activated (2 and 6)

The "Key/button activated" element indicates whether or a key, button or a touch-sensitive area has been confirmed by the control program (processed). The status information is supplied by the corresponding function block from HMI Lite.

The "Key/button activated" element acts as follows for a key-operated panel:

- When a key is pressed and confirmed by the controller, the rectangle becomes dark blue.
- If a key has not been pressed or not confirmed by the PLC, the rectangle is represented as an unfilled contour.

The "Key/button activated" element confirms the preselection or selection of a function on the touch display with the following states (further details for the preselection are contained in the "Touch panel support" section):

- If a function has been preselected, this is indicated by the flashing of the rectangle (a dark-blue and contoured rectangle flashes in intervals of approximately 0.25 seconds).
- If the touch-sensitive surface of a function is activated a second time and this is confirmed by the controller, the rectangle becomes dark blue.

If the function is not active or preselected, the rectangle is displayed as an unfilled contour.

5.1.3 Assignment of the function numbers

Each assigned function line is assigned to a fixed function number. The first line is assigned function number 1, the second line is assigned function number 2, etc.

Lines that are not displayed (all elements hidden) do not interrupt the assignment.

Figure 5-6 shows the assignment of the function numbers over several pages.

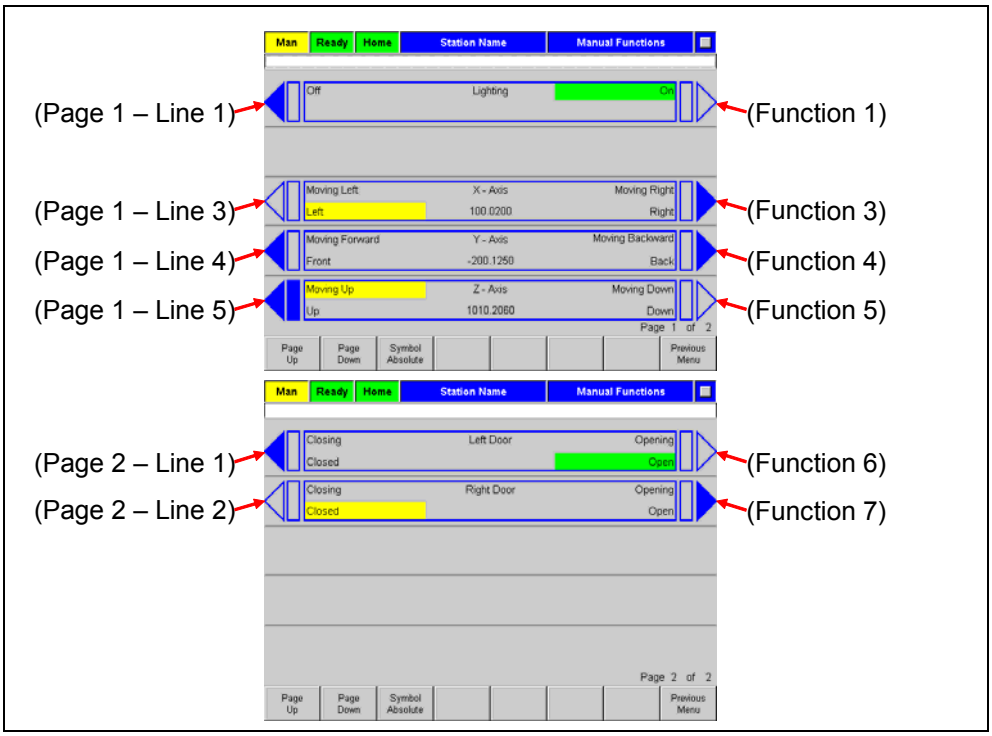


Figure 5-6 Manual operating screens – assignment of the function numbers

HMI Lite assigns the page layout of the movements or functions based on the total number of movements or functions configured in the selected screen and depending on how many movements or functions can be displayed on a page.

This means the movement-specific interfaces are page-dependent.

Note

The 10" display of a MP277 with header can display five function lines per screen page; the OP177B and other 6" devices can display three function lines per page.

5.2 Purpose of the individual manual operating screens

5.2.1 Manual operation

The manual operating screen contains a maximum of 64 movement or function lines. This allows special movements to be performed manually using keys. Every movement can be performed in two directions, such as input/output, open/close, up/down, forwards/backwards.

It is also possible to track each movement during its execution.

If there are more movements than can be displayed on the screen at the same time, the movements will be displayed on several pages. The individual pages can be grouped. This means each group forms its own manual operating screen for the operator and, for example, can be assigned to a specific plant section.

5.2.2 Power up condition

The "Power up condition" screen contains up to 64 function lines. This allows special power up conditions to be performed manually using keys. Each power up condition can be controlled in two directions, such as on/off, open/close.

It is also possible to track the status of each power up condition during its execution.

If there are more power up conditions than can be displayed on the screen at the same time, the power up conditions will be displayed on several pages

5.2.3 Selecting/deselecting units

The "Units" screen contains up to 32 function lines. Each line is assigned a machine unit that can be selected or deselected manually using keys.

If there are more units than can be displayed on the screen at the same time, the units will be displayed on several pages.

5.2.4 Selecting/deselecting nut runners

The "Nut runner" screen contains up to 32 function lines. Each line is assigned a nut runner that can be selected or deselected manually using keys.

If there are more nut runners than can be displayed on the screen at the same time, the nut runners will be displayed on several pages.

5.2.5 Selecting/deselecting nut driver groups

The "Nut runner groups" screen contains up to 32 function lines. Each line is assigned a nut runner group that can be selected or deselected manually using keys.

If there are more nut runner groups than can be displayed on the screen at the same time, the nut runner groups will be displayed on several pages.

5.2.6 Selecting/deselecting cycle type

The "Cycle type" screen contains up to 16 function lines. Each line is assigned a cycle type that can be selected or deselected manually using keys.

If there are more cycle types than can be displayed on the screen at the same time, the cycle types will be displayed on several pages.

5.2.7 User operating screen

The User operating screen is a freely-configurable manual operating screen that can be used for machine- or project-specific functions not assigned to any other manual operating screen.

5.3 Configuration and runtime interface

Each manual operating screen has its own text lists, parameter records and controller interface. These parameters and text lists have the same basic structure and are defined using the name of the associated screen.

The "DB_HMILITE_CONFIG" configuration DB and the "DB_HMILITE_DATA" runtime DB have their own data area for each screen; this data area is also defined by the designation of the associated screen. The name of the variable areas of each manual operating screen in the data blocks is shown in Table 5-1.

The name of the screen in WinCC flexible	**The name of the area in "DB_HMILITE_DATA" and "DB_HMILITE_CONFIG"**
SS_11_ManualMovement	SCREEN_MANUAL
SS_12_PowerUpCondition	SCREEN_POWERUP
SS_13_Unit	SCREEN_UNITS
SS_14_NutRunner	SCREEN_NUTRUNNER
SS_15_NutRunnerGroup	SCREEN_NUTRUNNER_GROUP
SS_16_CycleTypes	SCREEN_CYCLETYPE
SS_17_UserDefined	SCREEN_USER_DEFINED

Table 5-1 Manual operating screens – structure of the configuration interface

5.4 Configuration

Changes must be performed both in WinCC flexible and in STEP 7.

All text items are stored in text lists for WinCC flexible. Numeric parameters are stored in the HMI Lite DB_HMILITE_CONFIG configuration data block.

5.4.1 Global configurations

The "DB_HMILITE_CONFIG.MANUAL_COMMON" data area is used for the general configuration valid for all manual operating screens.

Display time of the absolute view

The time after which the absolute designation is switched back to the symbolic designation is stored in DB_HMILITE_CONFIG:

Address:	DB_HMILITE_CONFIG.MANUAL_COMMON. ABSOLUTE_DISPLAY_TIME
Format:	TIME
Range of values:	T#1MS…T#24D20H31M23S647MS
Default setting:	T#10S (10 seconds)

This parameter is independent of the screen.

"Touch operation preselection" timeout status

The time factor that determines how long a preselection initiated by touch remains active for a function is defined in DB_HMILITE_CONFIG in the following data address:

Address:	DB_HMILITE_CONFIG. MANUAL_COMMON. TOUCH_PRESELECTION_TIME
Format:	TIME
Range of values:	T#1MS…T#24D20H31M23S647MS
Default setting:	T#2S (2 seconds)

This parameter is required only for touch panels and does not depend on the screen (also refer to the "Touch panel support" section).

5.4.2 Number of movement or function lines

The number of required movement or function lines must be defined for each manual operating screen in the associated data block variable in DB_HMILITE_CONFIG.

Address:	DB_HMILITE_CONFIG. SCREEN_AAAAAA.NUMBER_OF_ROWS where AAAAAA = name of the screen (see Table 5-1)
Format:	INT
Range of values:	1…64 – for the "setup" and "power up condition" 1…32 – for units, nut runners and nut runner groups 1…16 – for the cycle type and the OEM screen
Default setting:	The maximum number of available lines

5.4.3 Grouping of the movement lines in the manual operating panel

To assign the manual operation to function groups, the manual operating screen can be selected more than once with different movement lines on the operator panel. When the screen is selected, the first and the last page number of the pages relevant for this screen selection must be entered for the variables specified below. This is done using the WinCC flexible "SetzeWert" function that is configured in addition to the "AktiviereBild" function on the key or button that selects the manual operating screen. Note that the page number of the last page must be assigned before the page number of the first page.

Address:	WinCC flexible variables: SS_10_NumberOfFirstPageAtManualScreen (first page) SS_10_NumberOfLastPageAtManualScreen (last page)
Format:	BYTE
Range of values:	1…Max – the maximum value depends on the number of movement lines (see Section 5.4.2) and on the number of lines per page (see Section 5.7). Example For 64 movement lines and 5 lines per page, this gives 13 pages with movement lines, consequently, the value range is 1...13.

WinCC flexible configuration of the screen selection of the manual operating screen in groups:

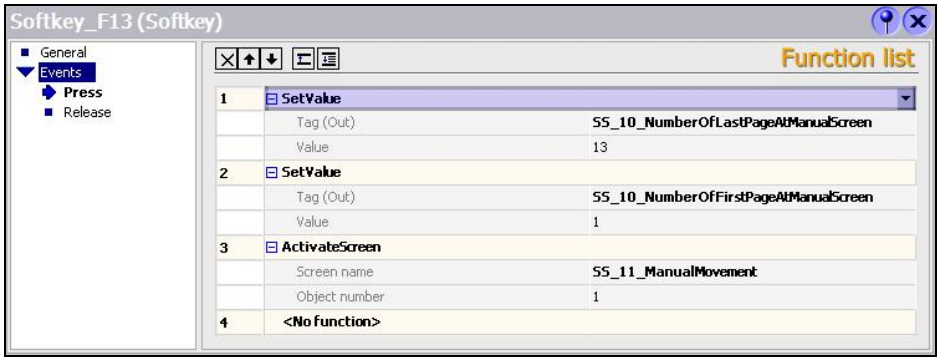


Figure 5-7 WinCC flexible configuration of the screen selection of the manual operating screen in groups

Note

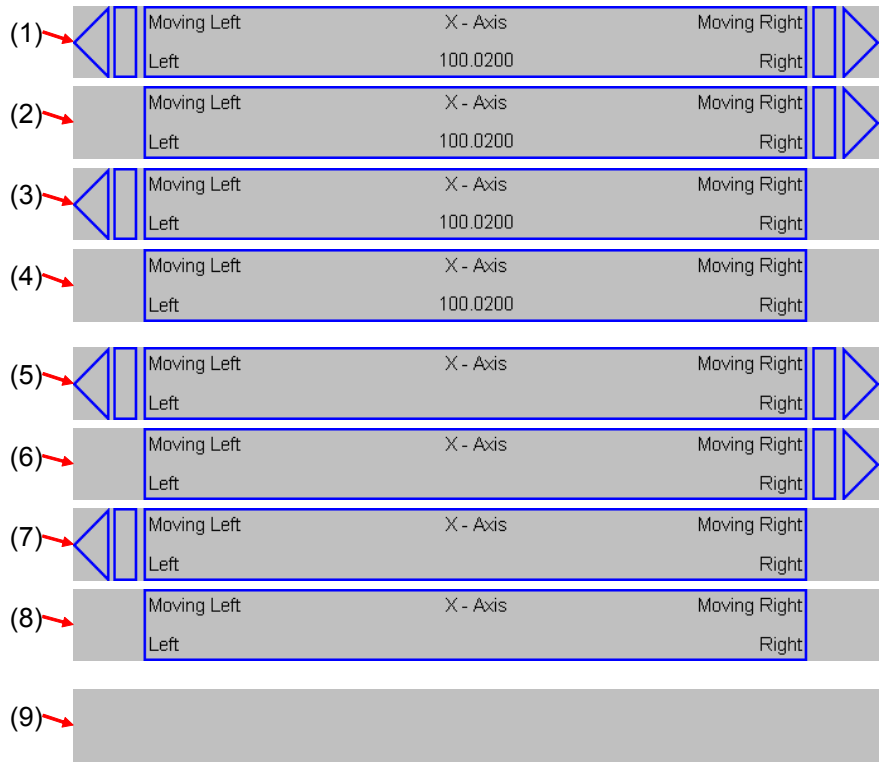
Note that the page number of the last page must be assigned before the page number of the first page.

5.4.4 Hiding elements of the function line

It is possible to hide elements from the function line (see Figure 5-8).

Depending on the associated configuration, the following elements can be hidden/displayed:

- Executability and "key pressed" on the left-hand side;
- Executability and "key pressed" on the right-hand side;
- "Position";
- All elements



(1)	All elements are visible	(Configuration = 0x0).
(2)	The left-hand side is hidden	(Configuration = 0x1).
(3)	The right-hand side is hidden	(Configuration = 0x2).
(4)	Both sides are hidden	(Configuration = 0x3).
(5)	The position is hidden	(Configuration = 0x4).
(6)	Position and left-hand side are hidden	(Configuration = 0x5).
(7)	Position and right-hand side are hidden	(Configuration = 0x6).
(8)	Position, left- and right-hand side are hidden	(Configuration = 0x7).
(9)	All elements are hidden	(Configuration = 0xF).

Figure 5-8 Manual operating screen – hiding screen elements

The individual function lines are configured at the following address in the DB_HMILITE_CONFIG as described above:

Address:	DB_HMILITE_CONFIG.SCREEN_AAAAAA.ROW_XX where AAAAAA = name of the screen (see Table 5-1) and XX is the number of the associated function line
Format:	BYTE
Range of values:	See Figure 5-9 and Figure 5-2
Default setting:	B#16#00

As shown in Figure 5-9, two configurations (configuration 1 and configuration 2) are possible for each movement/function.

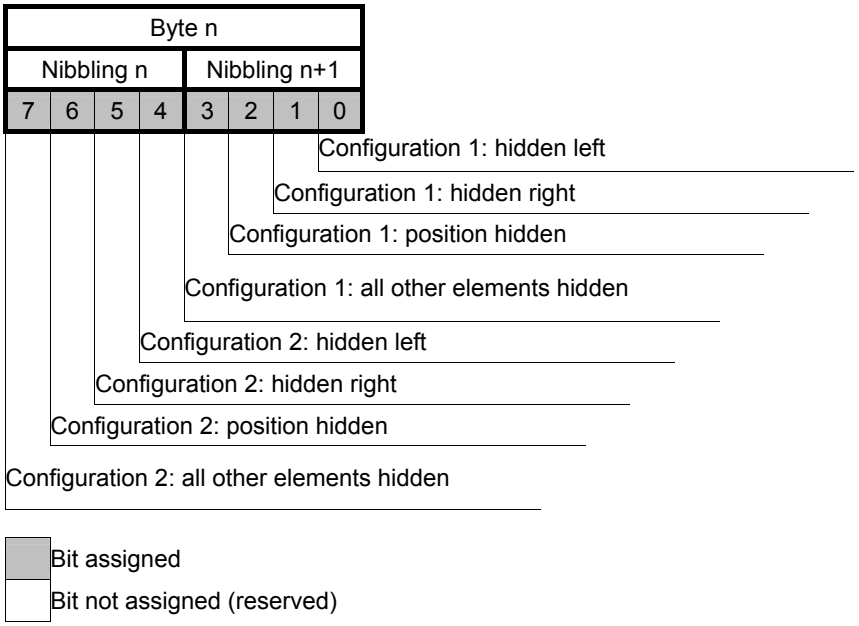


Figure 5-9 Manual operating screen – structure of the configuration interface

Only one configuration can be active for all movements or functions at any one time. The associated active configuration (configuration 1 or 2) can be selected dynamically using the "SELECT_CONFIG" input parameter for the "FC_HMILITE_MANUAL" function.

The dynamic changing of the configuration settings allows movement or function elements to be displayed or hidden depending on the associated machine status (e.g. machine in manual operation). Typical examples are:

- Hide the "Executability" and "Key pressed" elements in automatic operation
  manual operation ⇨ configuration 1 ⇨BIN 0000, HEX 0
  automatic operation ⇨ configuration 2 ⇨BIN 0011, HEX 3
  parameter value: BIN 0011 0000 = HEX 03 = B#16#30
- Hide the "Position" display in automatic operation
  manual operation ⇨ configuration 1 ⇨BIN 0000, HEX 0
  automatic operation ⇨ configuration 2 ⇨BIN 0100, HEX 4
  parameter value: BIN 0100 0000 = HEX 40 = B#16#40

Table 5-2 shows all possible parameter values for the various configurations.

All others	Item	Right-hand side	Left-hand side:	Value		
				BIN	DEC	HEX
				0000	0	0x0
			yes	0001	1	0x1
		yes		0010	2	0x2
		yes	yes	0011	3	0x3
	yes			0100	4	0x4
	yes		yes	0101	5	0x5
	yes	yes		0110	6	0x6
	yes	yes	yes	0111	7	0x7
yes	yes	yes	yes	1111	15	0xF

Table 5-2 Manual operating screens – values of the configuration interfaces

Example configurations

Several configuration examples follow:

The "Position" element is hidden for both configuration settings:
all types of machine elements (e.g. pumps, valves) that do not supply any confirmation of the position.

The "Executable" and "Key pressed" elements are hidden for both configuration settings: Machine elements that are not controlled from the operator panel; only the status needs to be displayed here (e.g. the "on/off" state controlled by the pushbutton).

The "Executable" and "Key pressed" elements are hidden for one configuration setting: Machine elements that can be controlled only in manual operation (e.g. machine axis); only the status of these elements (e.g. "axis moves left" ⇨ "execution" and "axis has reached the left-hand limit switch" ⇨ "end state") is displayed in automatic operation.

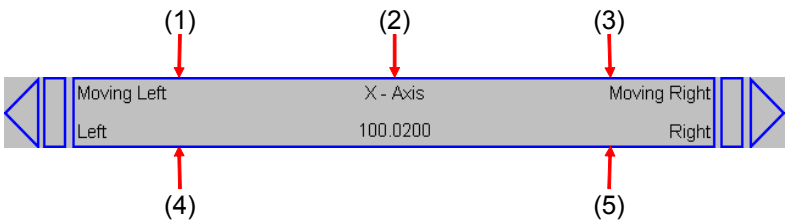
All elements are hidden for both configuration settings:
if this setting is made, a blank line results so that the movement or function groups (e.g. axis – blank line – clamping – blank line – lubrication) can be separated from each other.

Note

HMI Lite does not interlock the output signals with the "hidden" configuration settings. This means, even when the "Executable" and "Key pressed" elements are configured as hidden, the output signals will still be initiated by pressing the keys to the left or right of the movement or function or by touching the buttons. The machine-specific control program must realize any required interlock functionality.

5.4.5 Display text

All text items displayed in the manual operating screens are configured in the WinCC flexible text lists. Each screen has its own text list, where the text can be specified for each element. The text lists have the structure shown in Figure 5-10.



(1) Left confirmation
(2) Name
(3) Right confirmation
(4) Left final state
(5) Right final state

Figure 5-10 Manual operating screens – text lists

All text lists have the same structure (see Table 5-3).

Text list		**SO_11_Manual** **SO_12_PowerUp** **SO_13_Unit** **SO_14_Nutrunner** **SO_15_NutrunnerGroup** **SO_16_CycleType** **SO_17_UserDefined**
Display Format		Text Decimal
Value	10	Line #1 - function name – symbolic
Value	11	Line #1 - function name – absolute
Value	12	Line #1 – left confirmation – symbolic
Value	13	Line #1 – left confirmation – absolute
Value	14	Line #1 – left final state – symbolic
Value	15	Line #1 – left final state – absolute
Value	16	Line #1 – right confirmation – symbolic
Value	17	Line #1 – right confirmation – absolute
Value	18	Line #1 – right final state – symbolic
Value	19	Line #1 – right final state – absolute
Value	20	Line #2 - function name – symbolic
Value	21	Line #2 - function name – absolute
	…	

Table 5-3 Manual operating screens – structure of the text lists

Each screen element is always assigned two text list positions:

- The first position contains the text for the symbolic representation.
- The second position specifies the text for the absolute view.

Example

The following examples show all required steps for configuring the display text for a movement to be displayed in the third line on the first screen page.

The movement to be specified is a numeric axis that is to move left or right.

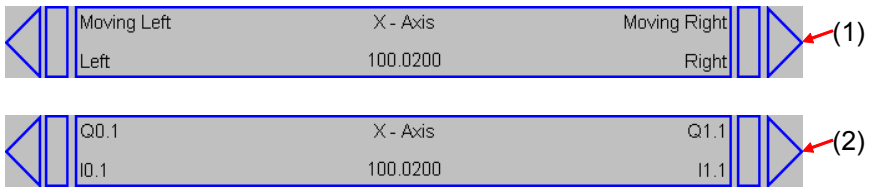
The movement is initiated by the Q0.1 and Q1.1 outputs. The movement is limited by limit switches connected to the I0.1 and I0.1 inputs. Consequently, the display text is as follows:

Text element of the movement line	Text to be displayed
"Designation" text for the symbolic view:	"X axis"
"Designation" text for the absolute view:	"X axis"
"Execution left" text for the symbolic view:	"Moves left"
"Execution left" text for the absolute view:	"Q0.1"
"Final state left" text for the symbolic view:	"Left"
"Final state left" text for the absolute view:	"I0.1"
"Execution right" text for the symbolic view:	"Moves right"
"Execution right" text for the absolute view:	"Q1.1"
"Final state right" text for the symbolic view:	"Right"
"Final state right" text for the absolute view:	"I1.1"

The following tables show all required positions of the text lists based on the above assignment:

Text list		**SO_11_Manual**
Value	30	X axis
Value	31	X axis
Value	32	Moves left
Value	33	Q0.1
Value	34	Left
Value	35	I0.1
Value	36	Moves right
Value	37	Q1.1
Value	38	Right
Value	39	I1.1

The specified movement is displayed as follows:

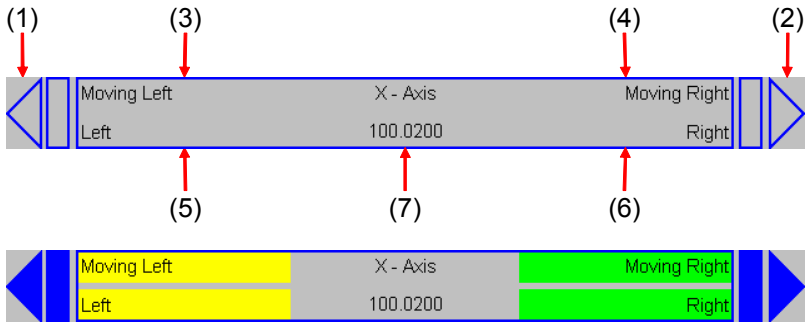


(1) Symbolic view
(2) Absolute view

Figure 5-11 Manual operating screens – example for the configuration of a text

5.5 Runtime interface

Color changes show the details of the binary state of a movement or function (see Figure 5-12). Only the "Position" element shows a numeric position value.



(1)	Executable (interlock), left-hand side
(2)	Executable (interlock), right-hand side
(3)	"Will be performed" / "Moving" confirmation, left-hand side
(4)	"Will be performed" / "Moving" confirmation, right-hand side
(5)	Final status / end position, left-hand side
(6)	Final status / end position, right-hand side
(7)	Item

Figure 5-12 Manual operating screens – dynamic movement elements

5.5.1 Interface for information about the binary state

The data addresses in the DB_HMILITE_DATA data block control the details concerning the binary state of a movement or function.

Address:	"DB_HMILITE_DATA".SCREEN_AAAAAA.ROW[XX] where AAAAAA = name of the screen (see Table 5-1) and XX is the number of the associated function
Format:	BYTE
Range of values:	See Figure 5-13 and Figure 5-14
Default setting:	---

Each grouping element represents a movement or function.

Figure 5-13 describes the structure of the interface for the function byte.

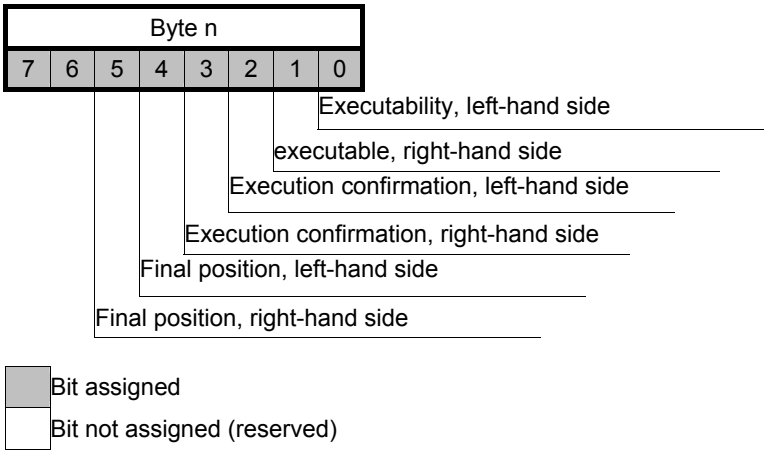


Figure 5-13 Manual operating screens – structure of the runtime interface

Bits 0…5 control the display of the information concerning the binary status (the function for bits 6 ... 7 is described in Section 8.6).

! Important

Bits 6 and 7 of the runtime data interface are used as control signals (operator panel ⇨ controller). This means the bits that supply information about the status must be addressed individually. If all status information was written concurrently with a single byte transfer command, the control signals would be overwritten and falsified.

Guidelines

The information items that provide the binary status are not mutually interlocked so that a real representation of the input and output signals is produced.

The following guidelines, however, provide a general statement of how the information items that provide the binary status can be used in practice:

- The two "Execution" displays should never be active concurrently for a single movement. Otherwise this would provide the impression that the movement would be performed at the same time in both directions.
- The two "Final status" displays should never be active concurrently for a single movement. This would provide the impression that the movement had reached both end positions (at opposite directions) at the same time.
- The two "Executable" and "Final status" displays should never be active concurrently for a single movement. Otherwise this would provide the impression that the movement is executable although the final position has already been reached.

- The "Execution" and "Final status" displays should never be active concurrently for a single movement. This would indicate that a movement or function is currently active although the final position has already been reached.

5.5.2 Display the actual position

In contrast to the interface for the binary status, the "Position" interface does not have its own data address for the individual movements or functions.

The following data interface is shared in the "SS_11_ManualMovement" screen:

Address:	DB_HMILITE_DATA.MANUAL_COMMON. POSITION[1] … POSITION[8]
Format:	REAL
Range of values:	Maximum 18 digits are displayed (4 decimal places)
Default setting:	---

The machine-specific program must copy the movement-specific position value into the variable for the position display.

Example

A position value should be displayed for the manual movement no. 3 (display in the "SS_11_ManualMovement" screen, page 1, line 3).

Consequently, the machine-specific program must implement the following logic:

IF (active screen == "SS_11_ManualMovement" screen) AND
IF (current page == 1), THEN
DB_HMILITE_DATA.MANUAL_COMMON.POSITION[3] = value

The selected screen and the active page can be determined using the following data addresses in the DB_HMILITE_DATA data block:

Selected screen

Address:	DB_HMILITE_DATA. GLOBAL.SCREEN_ID
Format:	WORD
Range of values:	W#16#0000 … W#16#FFFF (1…65535) for the idenitification of the screen, see Table 5-4
Default setting:	---

Current page

Address:	DB_HMILITE_DATA. MANUAL_COMMON.CURRENT_PAGE
Format:	BYTE
Range of values:	B#16#01…B#16#13 (1…13)
Default setting:	---

First and last visible line

Address:	DB_HMILITE_DATA.MANUAL_COMMON. ROW_VISIBLE_FIRST DB_HMILITE_DATA.MANUAL_COMMON. ROW_VISIBLE_LAST
Format:	WORD
Range of values:	W#16#0001…W#16#0040 (1…64)
Default setting:	---

The "first and last line" details can be used as an alternative method to determine whether the movement is currently being displayed.

! Important

The following DB_HMILITE_DATA data addresses:
GLOBAL.SCREEN_ID
MANUAL_COMMON.CURRENT_PAGE
MANUAL_COMMON.ROW_VISIBLE_FIRST
MANUAL_COMMON.ROW_VISIBLE_LAST
contain internal data for HMI Lite.
Like all other addresses, these variables should only be used read-only.

5.6 Control interface

A movement or function can be initiated using one of the following operator actions:

- with the keys indicated by the corresponding triangle icon;
- by touching the appropriate button for the corresponding movement.

HMI Lite provides two different interfaces that the machine-specific program can use to evaluate these operator commands.

The "job mailbox" is used as data interface to send jobs from the operator panel to the controller. A job to be performed by the control program is then initiated with an operator input. The "job mailbox" is used by all HMI Lite screens.

The other interface is screen-specific and, in contrast to the "job mailbox", uses binary signals.

Either the "job mailbox" or the "binary control interface" can be used to initiate a movement or function.

5.6.1 Job mailbox

The data addresses of the "job mailbox" belong to the DB_HMILITE_DATA.GLOBAL and defined as follows:

Address:	DB_HMILITE_DATA. GLOBAL.JOB.NUMBER GLOBAL.JOB.PARAMETER_1 GLOBAL.JOB.PARAMETER_2 GLOBAL.JOB.PARAMETER_3
Format:	WORD
Range of values:	W#16#0000…W#16#FFFF
Default setting:	---

When the operator panel initiates a movement or function (for example, an operator presses a key at the left or right of the movement), the following information will be displayed in the "job mailbox":

Job number:	Screen identification code, e.g. W#16#0A01 for the "SS_11_ManualMovement" screen (see Table 5-4)
Parameter 1:	Number of the movement/function (e.g. W#16#0001 for the first movement)
Parameter 2:	Direction of motion W#16#0001: "right" direction (bit 0) W#16#0002: "left" direction (bit 1)
Parameter 3:	Reserved for internal use

The code for identifying the screen ("job number" parameter in the job mailbox) for the manual operating screens is described in Table 5-4:

Screen	Identification code of the associated screen
SS_11_ManualMovement	W#16#0A01
SS_12_PowerUpCondition	W#16#0A02
SS_13_Unit	W#16#0A03
SS_14_NutRunner	W#16#0A04
SS_15_NutRunnerGroup	W#16#0A05
SS_16_CycleTypes	W#16#0A06
SS_17_UserDefined	W#16#0A07

Table 5-4 Operating screens - code for identifying the screen in the "job mailbox"

Example

The "SS_11_ManualMovement" screen is active and displays the first screen page.

When the operator presses the left key that shows the triangle of the second movement line (function number 2), the following data will be displayed in the "job mailbox":

Job number: W#16#0A01 ⇨ "SS_11_ManualMovement" screen
Parameter 1: W#16#0002 ⇨ "Second movement line" function
Parameter 2: W#16#0001 ⇨ "Left" direction

When the operator releases the key, the values for parameter 1 and parameter 2 will be cleared (value W#16#0000).

Important

The "job number" will not be cleared when the operator releases a key used to initiate a movement.
The "job number" will be set when one of the operating screens becomes active.

The machine-specific user program must analyze the "job mailbox" data and initiate the required commands for performing the movement or function.

5.6.2 Binary control interface

The "binary control interface" is a interface that uses binary signals. Each movement or function is assigned two binary signals that represent a possible direction of the associated movement or function.

Address:	DB_HMILITE_DATA.SCREEN_AAAAAA.ROW[XX] where AAAAAA = name of the screen (see Table 5-1) and XX is the number of the associated function
Format:	BYTE
Range of values:	See Figure 5-13 and Figure 5-14
Default setting:	---

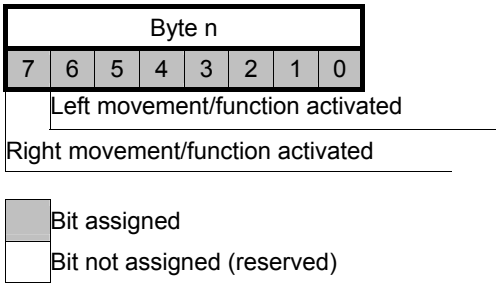


Figure 5-14 Manual operating screens – structure of the control interface

When a movement or function is initiated from the operator panel (for example, when the operator presses a key assigned to a function line), the control bits will be set; the control bit will be reset when the function key is released.

Example

A manual operating screen is active and displays the first screen page.

When the operator presses the left key that shows the triangle of the second movement line (function number 2), bit 6 for DB_HMILITE_DATA.SCREEN_MANUAL.ROW[2] will be set.

The bit will be reset when the operator releases the key.

5.7 FC_HMILITE_MANUAL

The FC_HMILITE_MANUAL function provides the following functionality:

- Scroll in the manual operating screens when more movements/functions have been specified than can be displayed on the screen.
- Switch between the symbolic and the absolute representation.
- Switch between the first and the second configuration of the movement or function line.
- Representation of the key signals on the control interfaces.
- Monitor the connection between the operator panel and the controller.
- Interlock the signals for the key-operated panel or the interfaces of the touch operated panels or the direct keys.

FC_HMILITE_MANUAL must be called cyclically.

FC_HMILITE_MANUAL call interface

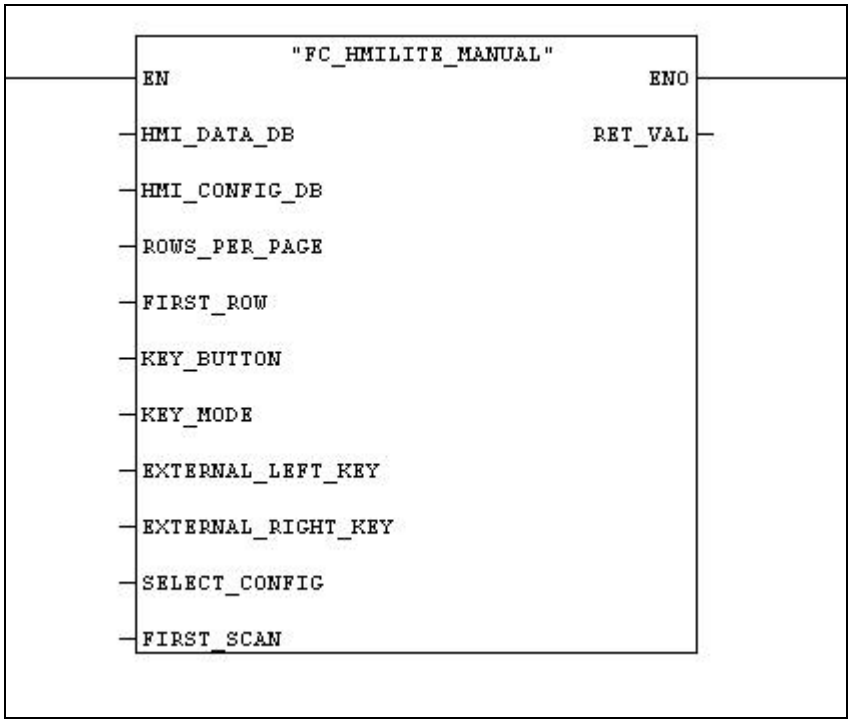


Figure 5-15 FC_HMILITE_MANUAL – interface for calling the function

FC_HMILITE_MANUAL parameters

Name	Type	Default	Example	Description
HMI_DATA_DB	Int	67	67	The number of the HMI runtime data block.
HMI_CONFIG_DB	Int	68	68	The number of the HMI configuration data block.
ROWS_PER_PAGE	Int	---	B#16#5	The number of function lines that can be displayed on the screen at the same time. Three lines for the 6-inch operator panel with header. Five lines for the 10-inch operator panel with header.
FIRST_ROW	Int	2	2	The function keys located at the side to be used for the first movement line. 1 = first movement will be performed by F1 and F2; optional for operation without header. 2 = first movement will be performed by F3 and F4; optional for operation without header.
KEY_BUTTON	Word	--	IW 5	Input word of the PROFIBUS DP direct keys. Assignment: Bit 0: line 5, left key Bit 1: line 5, right key Bit 2: line 6, left key Bit 3: line 6, right key Bit 4-7: Reserve Bit 8: line 1, left key Bit 9: line 1, right key Bit 10: line 2, left key Bit 11: line 2, right key Bit 12: line 3, left key Bit 13: line 3, right key Bit 14: line 4, left key Bit 15: line 4, right key If no direct keys are used, the value W#16#FFFF must be specified.
KEY_MODE	Int	---	0	Mode for executing the operation. 0: Function active while a key remains pressed. 1: Function active while a button remains pressed. 2: Function active after the button has been clicked twice. 3: Function active while an external key remains pressed; selection of the function by the function keys located at the side. 4: Function active while an external key remains pressed; selection of the function by the function buttons.
EXTERNAL_LEFT_ KEY	Bool	---	---	Only relevant in the 3 and 4 key modes. Performs the left command of the selected function.

Name	Type	Default	Example	Description
EXTERNAL_RIGHT_ KEY	Bool	---	---	Only relevant in the 3 and 4 key modes. Performs the right command of the selected function.
SELECT_CONFIG	Bool	---	M5.3	Switch between the two configurations for hiding of individual elements of the function line.
FIRST_SCAN	Bool	---	---	Restart flag, 1 – signal for the first cycle after CPU startup
RET_VAL	Word	---	MW2	Error message: 0000: No error 7000: No operating screen selected, block will not be processed currently 7002: No sign-of-life from the OP, operation is disabled 8091: Invalid screen ID DB_HMILITE_DATA.GLOBAL.BILD_ID is invalid 8092: Invalid key mode TASTEN_MODUS < 0 or > 4

Table 5-5 Description of the FC_HMILITE_MANUAL parameters

Note

An additional safety function must be programmed for the parameterization of key mode 3 and 4 (use of external keys) for performing movements.

The program must set the DBX296.0 bit in the DB67 (DB_HMILITE_DATA) when the selection of a movement is to be disabled. For example, this can be implemented by activating a key switch.

The DB67.DBX296.1 bit causes the program code to reset the selection and re-release the selection of other movements.

5.8 FB_HMILITE_S7G_MANUAL

The FB_HMILITE_S7G_MANUAL function provides the following functionality:

- Display the executability of the movements displayed on the panel.
- Activate a configured S7 Graph step for the selection of a movement by pressing a key on the operating screen.

FB_HMILITE_S7G_MANUAL must be called cyclically.

Syntax of the function call

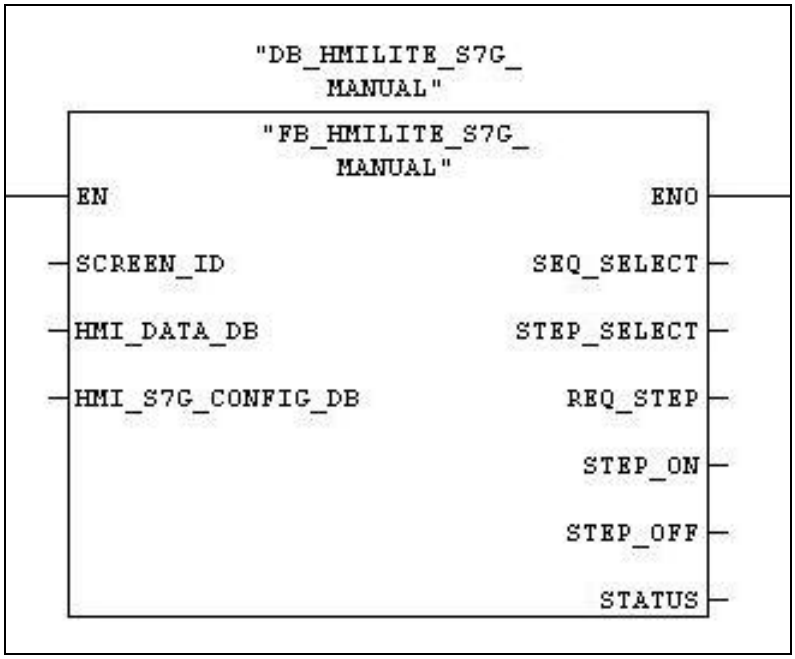


Figure 5-16 FB_HMILITE_S7G_MANUAL – interface for calling the function

FB_HMILITE_S7G_MANUAL parameters

Name	Type	Default	Example	Description
SCREEN_ID	WORD	W#16#A01	W#16#A01	Screen ID of the operating screen for which the FB call is to be valid
HMI_DATA_DB	INT	67	67	The number of the HMI runtime data block (DB67)
HMI_S7G_ CONFIG_DB	INT	106	106	The number of the S7GRAPH configuration data block (DB106)
SEQ_SELECT	INT	---	MW180	Output the sequencer DB number of the selected movement
STEP_SELECT	INT	---	MW182	Output the step number of the selected movement
REQ_STEP	BOOL	---	M184.0	Step activation/deactivation running
STEP_ON	BOOL	---	M184.1	Switch on step (activate)

Name	Type	Default	Example	Description
STEP_OFF	BOOL	---	M184.2	Switch off step (deactivate)
STATUS	WORD	---	MW186	Fault or status message of the selected movement (see following tables)

Table 5-6 Description of the FB_HMILITE_S7G_MANUAL parameters

Status information of the Status parameter

Value (W#16# ….)	Description	S7-Graph parameter
0001	Initialize step	
0002	Select step number	S_SEL
0003	Step number correct	S_NO = S_SEL
0004	Activate step	S_ON = True
0005	Step active	S_ACTIVE = True > S_ON = False
0006	Deactivate sequencer	OFF_SQ = True
0007	Sequencer off	OFF_SQ = False

Fault information of the Status parameter

Value (W#16# ….)	Explanation
8001	No DB number entered in "HMILITE_S7G_CONFIG_DB"
8002	DB does not exist
8003	DB too short
8004	No S7-Graph DB
8005	S7-Graph version not V4.0 .. V5.9
8006	S7-Graph FB version > V5.9
8007	S7-Graph FB parameter not "Standard" or "Maximum"
8008	Step number not present
8011	Sequencer not in manual operation (MAN_ON = False)

Operation of the FB:

If an operating screen is selected on the operator panel, the FB uses the SCREEN_ID to check whether the operating screen is valid for calling the FB.

If the call is valid, it copies the parameterization from the HMILITE_S7G_CONFIG_DB (DB106) for the movements displayed in the screen into its instance DB. It also checks the sequencer DB and the presence of the specified step number in the sequencer. If it detects a fault, the error number for the corresponding parameterization will be entered in the instance DB. The error number corresponds to the fault information for the Status parameter.

If the parameterization is correct, the block assigns the executability (interlock) of the movements in the HMI_DATA_DB (DB67) so that it is displayed on the screen. The triangle on the right- and left-hand edge of the movement line (refer to the description of the operating screens) shows on the screen the executability.

If a movement is initiated by pressing a key (button), the FB activates the parameterized step in the corresponding sequencer. The FB deactivates the step when the key (button) is released. This all assumes the correct parameterization.

Requirements:

1. The sequencer must be in the manual operating mode.

   The activation of a step is possible only in the manual operating mode (MAN_ON = TRUE). The operating mode is checked before the step is activated. If the sequencer is not in manual operation, the step will not be activated and an error number (STATUS = W#16#8011) will be issued.

2. The "OFF_SQ", "S_ON" and "S_SEL" sequencer parameters must not be overwritten by the user program.

   The function uses the "OFF_SQ", "S_ON" and "S_SEL" sequencer parameters directly. These parameters must not be overwritten by the user program while the step is being activated. To ensure this, the assignments of these data bits should be bypassed using the "REQ_STEP" output signal of the FB. In addition, these parameters must not be supplied directly on the sequencer FB.

   If it is not possible to bypass the assignments of the sequencer parameters, they must be connected with the "SEQ_SELECT", "STEP_SELECT", "STEP_ON" and "STEP_OFF" output signals of the FB 107 so that the function is still provided.

3. Prior to activating a step, all other steps should be deactivated.

   It is not permitted for several steps to be active concurrently in a sequential sequencer. Consequently, the sequencer FB does not permit a second step to be activated for an active step.

4. Only the "Standard" or "Maximum" FB parameter types may be used.

   The FB_HMILITE_S7G_MANUAL (FB107) checks the setting of the FB parameters of the sequencer. If a different parameter type is selected, an error number (STATUS = W#16#8007) will be output and the processing of this sequencer terminated.

5. The "Permanent processing of all interlocks in manual operation" block setting must be selected.

   To ensure that the executability (interlock) of a movement is displayed correctly, the sequencer FB must update all interlocks in manual operation.

Tips and tricks:

The FB_HMILITE_S7G_MANUAL (FB107) can be used for all operating screens. It must be called with different instance DBs and a unique HMILITE_S7G_CONFIG_DB (DB106) must be created for each operating screen.

The sequencer DB number and the step number are parameterized in the "HMILITE_S7G_CONFIG_DB" (DB106)

For each configured line in the operating screen, the following variables must be parameterized in the HMILITE_S7G_CONFIG_DB (DB106) data block:

Address:	HMILITE_S7G_CONFIG_DB.ROW_XX.LEFT.SQ_DB_NO (where XX = the number of the line of the operating screen)
Format:	INT
Range of values:	The data block number of the sequencer in which the corresponding line (XX) of the step is to be activated when the left key (button) is pressed.
Default setting:	0

Address:	HMILITE_S7G_CONFIG_DB.ROW_XX.LEFT.STEP_NO (where XX = the number of the line of the operating screen)
Format:	INT
Range of values:	The step number of the step in which the corresponding line (XX) of the step is to be activated when the left key (button) is pressed.
Default setting:	0

Address:	HMILITE_S7G_CONFIG_DB. ROW_XX.RIGHT.SQ_DB_NO (where XX = the number of the line of the operating screen)
Format:	INT
Range of values:	The data block number of the sequencer in which the corresponding line (XX) of the step is to be activated when the right key (button) is pressed.
Default setting:	0

Address:	HMILITE_S7G_CONFIG_DB.ROW_XX.RIGHT.STEP_NO (where XX = the number of the line of the operating screen)
Format:	INT
Range of values:	The step number of the step in which the corresponding line (XX) of the step is to be activated when the right key (button) is pressed.
Default setting:	0

5.9 Step-by-step procedure

The following table summarizes the steps required to create a manual operating screen.

All other operating screens differ only in the listed data addresses and the WinCC flexible tex tlists.

Step 6	Proceed as follows
1	Use the data block editor in STEP 7 (declaration view) to open the DB_HMILITE_CONFIG data block.
2	Specify the display time for the absolute view: HANDBEDIENUNG_ALLGEMEIN.ZEIT_ABSOLUTE_ANZEIGE:
3	Specify the number of movements/functions for the manual operating screen: BILD_HANDBEDIENUNG.ANZAHL_ZEILEN
4	Specify which elements are to be hidden for all function lines: BILD_HANDBEDIENUNG.ZEILE_01 … ZEILE_64
5	Now switch to the data view and initialize the data block.
6	Save and close the DB_HMILITE_CONFIG data block.
7	Use STEP 7 to open the FC_HMILITE_MANAG function.
8	Call FC_HMILITE_MANUAL and set the parameters appropriately for your requirements.
10	Load all changed blocks into the controller.
11	Use WinCC flexible to open the WinCC flexible project from HMI Lite.
12	Open the SO_11_Manual text list and define for the corresponding function lines the display text for the symbolic and the absolute view.
13	Close the text list.
15	Save and generate the WinCC flexible project, and transfer it to the operator panel.
16	To display the status of the movement/function, the corresponding signals must be assigned in the DB_HMILITE_DATA to the interface area of the operating screens.
17	The user program can use two interfaces to perform movements/functions. Binary interface: DB_HMILITE_DATA.BILD_HANDBEDIENUNG.ZEILE_01 … 64 Or job mailbox: DB_HMILITE_DATA.GLOBAL.AUFTRAEGE.AUFTRAGSNUMMER DB_HMILITE_DATA.GLOBAL.AUFTRAEGE.PARAMETER_1 DB_HMILITE_DATA.GLOBAL.AUFTRAEGE.PARAMETER_2

Table 5-7 Procedure for creating a manual operating screen

■

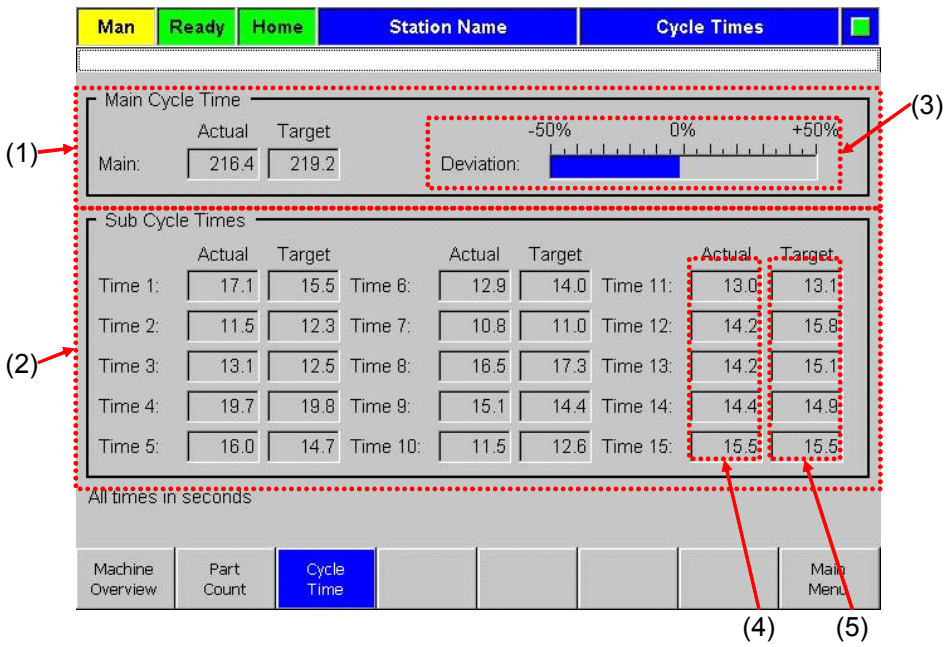
6

6 "Production Data" Screens

6.1 "Cycle times" screen

6.1.1 Layout of the screen and functionality

The "Clock times" screen displays the total clock time and the sub-clock times of the machine.



(1)	"Total clock time" area
(2)	"Sub-clock time" area (can optionally be hidden from the display)
(3)	Deviation of the total clock time as percentage (± 50%) from the set clock time
(4)	Actual clock time values
(5)	Set clock time values

Figure 6-1 "Clock times" screen

Display elements

This screen is subdivided into the following two main areas:

- Main clock time
- Sub-clock time

The "Main clock time" area displays the values for the actual clock time and the set clock time. The deviation between the the actual clock time and the set clock time is also specified as percentage. The range of the clock time deviation is limited to ± 50%. If the deviation lies outside this range, this will be indicated by arrows at the far left- or right-hand side of the bar.

Note

The deviation is calculated using the following equation:
deviation = actual clock time / set clock time

Procedure for the clock times

The clock time must be calculated with the start and the end signal of a clock pulse or single clock pulse. This value represents the actual clock time and will be updated when it is redefined. The interruption of a clock time is possible. The evaluation of a signal (binary, change from 0 to 1) causes an interruption. The change from 1 to 0 (falling edge) causes the counting of the clock time to be continued.

Range of values

The values for the clock times are entered in 16-bit unsigned integer variables. The values are displayed in seconds with one decimal place. The displayed clock time resolution corresponds to a tenth of a second.

The maximum displayed value is 6553.5, which corresponds to 1 hour, 49 minutes, 13 seconds and 500 milliseconds.

The accuracy of the timer depends on the type of the associated controller. The documentation for the S7 CPU data provides more detailed information.

"Clock times" screen with reduced display functions

The "Single clock times" area can be hidden from the display. This function is controlled using the HIDE_SPECIFIC configuration parameter in the DB_HMILITE_CONFIG configuration data block. In this case, the complete control field with the single clock times will be hidden from the display (see Figure 6-1 Element (2)).

6.1.2 Runtime interface (FC_HMILITE_CYCLETIME)

The calculation of the clock times is realized with the FC_HMILITE _CYCLETIME (FC109) function. A total of sixteen clock times can be recorded. Each clock time acquisition can be started and stopped independent of each other clock time acquisition. The first acquisition is used for the total clock time. The other fifteen clock time acquisitions are used for the single clock times. No timers are used.

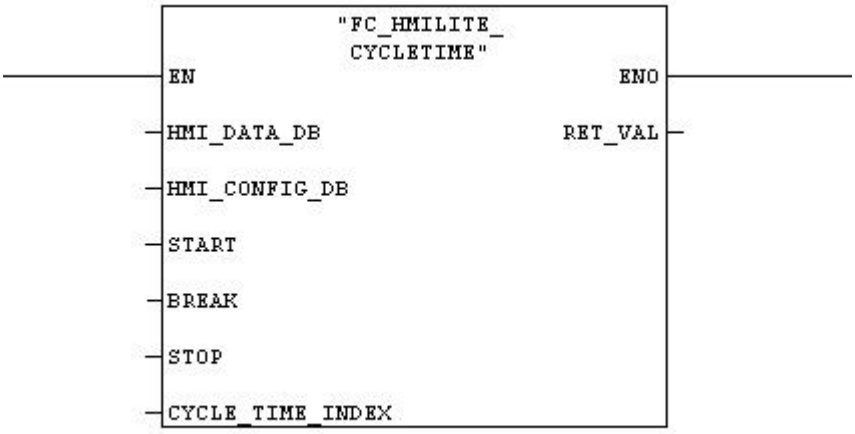


Figure 6-2 Call interface of the FC_HMILITE_CYCLETIME function

Name	Type	Default	Example	Description
HMI_DATA_DB	INT	67	67	The number of the HMI runtime data block
HMI_CONFIG_DB	INT	68	68	The number of the HMI configuration data block
START	BOOL	---	M0.5	A positive edge starts the clock time selected by the INDEX parameter
BREAK	BOOL	---	M0.4	A positive edge interrupts the counting, a negative edge continues the clock time measurement
STOP	BOOL	---	M0.5	A positive edge stops the clock time selected by the INDEX parameter
CYCLE_TIME_ INDEX	BYTE	---	MB6	Select the clock time to be measured. Index = 0 means total clock time Index = 1 – 15 means single clock time 1 - 15
RET_VAL	WORD	---	MW2	Function return value

Table 6-1 Parameters of the FC_HMILITE_CYCLETIME (FC109) function

Note

The simultaneous measurement of different clock times is possible by calling the block several times within a cycle.

6.1.3 Configuration

DB_HMILITE_CONFIG

The area for the single clock times can be hidden by setting the following variables:

Address:	DB_HMILITE_CONFIG. SCREEN_CYCLETIMES.HIDE_SPECIFIC
Format:	BOOL
Range of values:	FALSE = single clock times will be displayed TRUE = single clock times with be hidden
Default setting:	FALSE

A set clock time can be defined for the total clock time:

Address:	DB_HMILITE_CONFIG. SCREEN_CYCLETIMES.MAIN.TARGET
Format:	BOOL
Range of values:	INT
Default setting:	1.. 65535 (0.1 seconds)

A set clock time can be defined for each single clock time:

Address:	DB_HMILITE_CONFIG. SCREEN_CYCLETIMES.SUB.TARGET_XX (where XX is the number of the corresponding single clock time: 1..15)
Format:	INT
Range of values:	1.. 65535 (seconds).
Default setting:	0

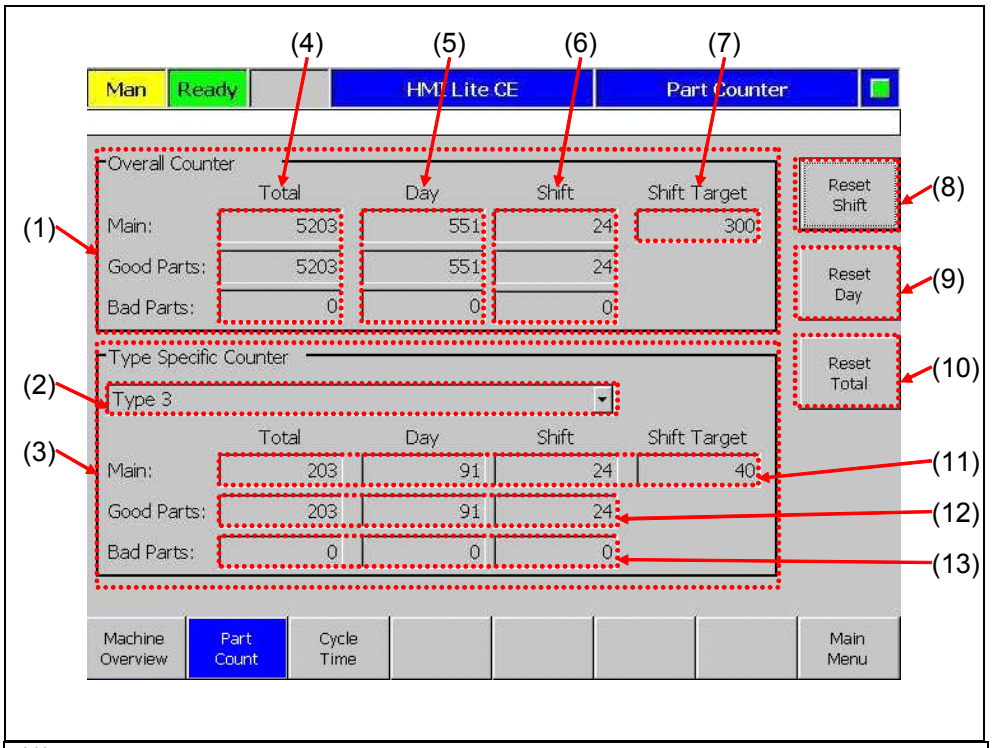
6.1.4 Step-by-step procedure

Step	Proceed as follows
1	Use the data block editor in STEP 7 to open the "DB_HMILITE_CONFIG" data block in the declaration view.
2	Set the SCREEN_CYCLETIMES.HIDE_SPECIFIC variable to the value: FALSE = when the single clock times are to be displayed FALSE = when the single clock times are to be hidden
3	Now edit the SCREEN_CYCLETIMES.MAIN.TARGET and SCREEN_CYCLETIMES.SUB.TARGET_01 to 15 variables to define the values for the set clock times.
4	Now switch to the data view and initialize the data block.
5	Save and close the DB_HMILITE_CONFIG data block.
6	Use the STEP 7 program editor to open the FC_HMILITE_ADDON (FC151) function.
7	Call FC_HMILITE_CYCLETIME (FC109) and assign the required parameters.
8	Now save and close the FC_HMILITE_ADDON (FC151) function.
9	Load all changed blocks into the controller.

6.2 "Workpiece counter" screen

6.2.1 Layout of the screen and functionality

The "Workpiece counter" screen is used to display the produced workpieces.



(1)	Total workpiece counter
(2)	Select the workpiece type
(3)	Workpiece-related counter
(4)	Total number of completed workpieces
(5)	Total number of workpieces produced on this day
(6)	Total number of workpieces produced during this shift
(7)	Planned workpiece count during a shift
(8)	Reset shift counter
(9)	Reset day counter
(10)	Reset total counter
(11)	Total number of produced parts
(12)	Number of produced parts that are correct
(13)	Number of produced parts that are faulty

Figure 6-3 "Workpiece counter" screen

Display elements

This screen is subdivided into the following two main areas:

- Number of total counters
- Number of workpiece-related counters.

Each area contains separate values for the shift, day and total counters.

These subareas are subdivided into the following counter values:

- Total workpiece counter (good and bad parts)
- Workpiece counter (good parts)
- Workpiece counter (bad parts)

If a setpoint is specified for the shift counter (value greater than 0), the output field for the planned workpiece number will be displayed. Otherwise the shift setpoint fields will be hidden.

The values for the type-specific workpiece counters can be selected using a selection list. A maximum of fifteen workpiece-related parts counters can be used. The text for the designaton of the workpiece must be edited by the machine manufacturer in a text list.

Procedure for counting

Depending on the machine cycle time, the user program must determine the number of produced good and bad parts.

Once these values have been determined, the counter variables in the "DB_HMILITE_DATA" data block must be updated using the following equation:

total workpieces = total workpieces old + number of produced parts

total bad parts = total bad old + number of produced bad parts

This must be performed at the same time for the shift, day and total counter of the total unit counter and the workpiece-related unit counter.

If no workpiece-related unit counters are required, only the total unit counter needs to be updated.

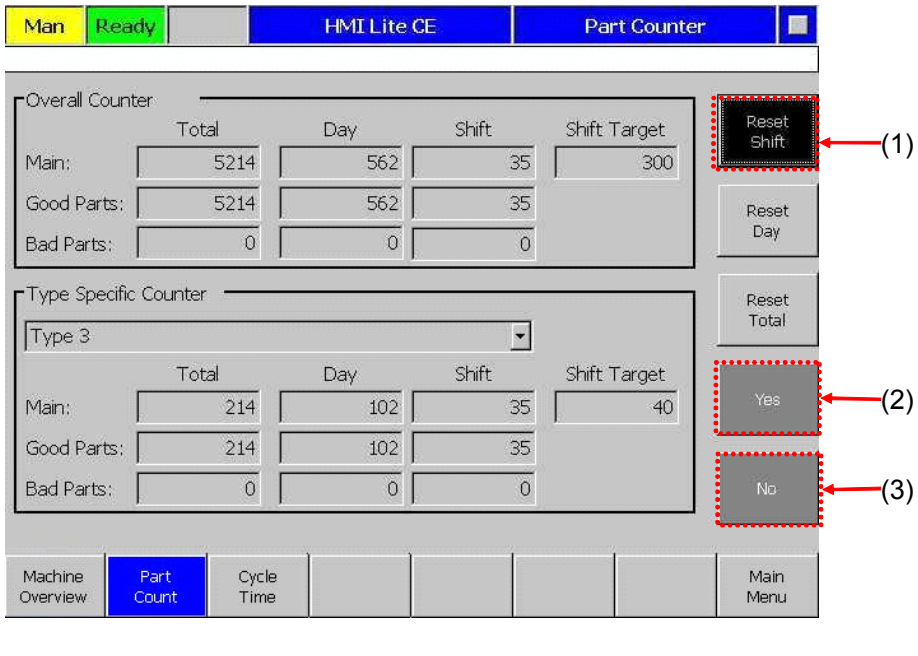
Procedure for resetting

In contrast to the procedure for counting, the procedure for resetting the counters is performed for the specific shift, day and total counters.

This means, for example, resetting the shift counter resets all shift-specific counters, the total unit counter and all workpiece-related unit counters.

The reset procedure must be initiated using the machine-specific logic. The RESET keys can also be used to initiate a manual reset. If required, the RESET keys can be hidden by setting the appropriate configuration bits in the DB_HMILITE_CONFIG.

Pressing a RESET key initiates the provided confirmation procedure.



(1) The RESET key that was pressed assumes a different color
(2) Button for confirming the RESET operation
(3) Button for canceling the RESET operation

Figure 6-4 Workpiece counter – procedure for confirming the reset

Range of values

Value ranges of the counters:

Total – sum of good/bad parts:	0 … 4.294.967.295
Total - bad parts:	0 … 65.535
Day – sum of good/bad parts:	0 … 65.535
Day - bad parts:	0 … 65.535
Shift – sum of good/bad parts:	0 … 65.535
Shift - bad parts:	0 … 65.535

The good parts counters are calculated by the operator panel.

Formula: [good parts] = [total parts] - [bad parts].

Workpiece counter screen with reduced display functions

The "Workpiece-related parts counter" area can be hidden from the display. This function is controlled using the HIDE_TYPE_SPECIFIC configuration parameter in the DB_HMILITE_CONFIG configuration data block. Hiding applies to the complete Workpiece-related parts counter (see number (3) in Figure 6-3).

Configuration parameter HIDE_TYPE_SPECIFIC has no effect in the 6" variant. Because the contents of two screens are split, reduction must be implemented by deleting the display with the single cycle times. The screen calls from screen "SS_62_PartCounterOverall" must be cleared.

6.2.2 Runtime interface (FC_HMILITE_COUNTER)

The "FC_HMILITE_COUNTER" (FC108) function uses directly the unit counter variables in the "DB_HMILITE_DATA" data block.

The user program can also access these variables (e.g. save the values for further processing or archiving before a RESET is performed).

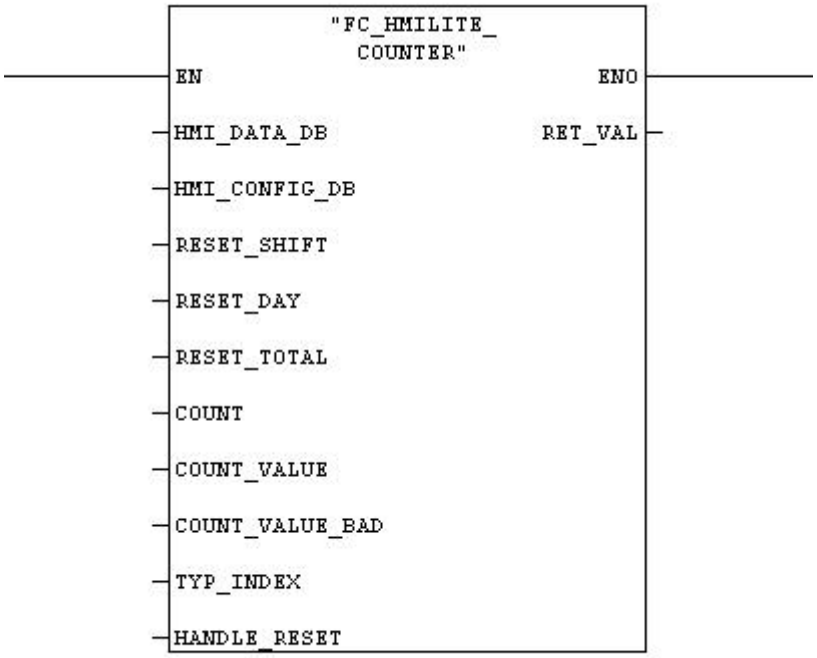


Figure 6-5 Call interface of the FC_HMILITE_COUNTER function

Name	Type	Default	Example	Description
HMI_DATA_DB	INT	67	67	The number of the HMI runtime data block
HMI_CONFIG_DB	INT	68	68	The number of the HMI configuration data block
RESET_SHIFT	BOOL	---	M5.0	An increasing edge resets the shift counters
RESET_DAY	BOOL	---	M5.1	An increasing edge resets the day counters
RESET_TOTAL	BOOL	---	M5.2	An increasing edge resets the total counters
COUNT	BOOL	---	M5.3	An increasing edge updates the counter
COUNT_VALUE	BYTE	---	MB6	The number of the total parts to be counted (good + bad parts)
COUNT_VALUE_ BAD	BYTE		MB7	Number of bad parts to be counted
TYP_INDEX	BYTE	---	MB8	The index of the workpiece type to be counted. If value = 0, only the type-independent unit counter will be processed. Values less than 0 or greater than 15 will cause an error message to be issued
HANDLE_RESET	BOOL	---	M5.4	Release the unit counters – Reset keys and buttons in the screen (workpiece counter), the Reset function in the screen will be performed only for a 1–signal of this parameter
RET_VAL	WORD	---	MW2	Function return value 0000: no error 8001: TYP_INDEX > 15 8002: TYP_INDEX < 0

Table 6-2 Time parameters of the FC_HMILITE_COUNTER function

An increasing edge of the COUNT parameter initiates a counting action. The total and bad parts counters are incremented using the following equation:

[new counter value] = [old counter value] + [counter value].

The counter value is defined by the COUNT_VALUE (good and bad parts) and COUNT_VALUE_BAD (bad parts) parameters. The values for the "good parts" are calculated using the following equation:

[good parts value] = [total parts value] - [bad parts value].

The shift, day and total counters are incremented by the same counter value.

The INDEX parameter specifies which workpiece-related counter will be updated. Up to 15 workpiece-related counters can be selected. The workpiece-independent total counter is always updated. If a value 0 is specified for the "INDEX" parameter, only the total counter will be updated.

The "Reset counter" parameters always reset all workpiece-related counters and the total counter. For example, the RESET_SHIFT function resets all workpiece-related counters (total, good and bad parts counter) and the total shift counter (total, good and bad parts counter). If during a cycle both an increasing edge at the "RESET" parameter and at the "COUNT" parameter is detected, the counter function and then the RESET function will be performed. Within a cycle, it is possible to reset the total counter, the day counter and shift counter. Using the controller-internal RESET functions and the reset functions of the user interface do not preclude each other (e.g. the RESET_SHIFT parameter can also be used when the HANDLE_RESET parameter specifies the 1–signal at the same time).

Note

The different workpieces can be counted within a cycle by calling the function several times.

6.2.3 Configuration

DB_HMILITE_CONFIG

The area for the workpiece type can be hidden by setting the following variables:

Address:	DB_HMILITE_CONFIG. SCREEN_COUNTER.HIDE_TYPE_SPECIFIC
Format:	BOOL
Range of values:	FALSE = workpiece type-dependent area will be displayed FALSE = workpiece type-dependent area will be hidden
Default setting:	FALSE

The RESET keys can be displayed and deactivated by setting the following variables. For example for the case when the reset is to be performed automatically by the user program:

Address:	DB_HMILITE_CONFIG. SCREEN_COUNTER.HIDE_RESET_SHIFT DB_HMILITE_CONFIG. SCREEN_COUNTER. HIDE_RESET_DAY DB_HMILITE_CONFIG. SCREEN_COUNTER. HIDE_RESET_TOTAL
Format:	BOOL
Range of values:	FALSE = the corresponding RESET key is active and displayed. FALSE = the corresponding RESET key is not active and hidden.
Default setting:	FALSE

The number of parts to be produced in the current shift (planned) must be configured in DB_HMILITE_CONFIG. The complete planned count (in accordance with the sum of the setpoints for all parts) and the workpiece-related planned count can be specified for each individual workpiece type. The addresses have the following form:

Address:	DB_HMILITE_CONFIG. SCREEN_COUNTER.OVERALL.SHIFT_TARGET
Format:	INT
Range of values:	0 .. 65535
Default setting:	0

Address:	DB_HMILITE_CONFIG. SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_ TARGET_XX (where XX is the number of the corresponding workpiece type: 1..15)
Format:	INT
Range of values:	0 .. 65535
Default setting:	0

It can be specified in the following variables how long the buttons for confirmation and cancel of the Reset function are to be visible and active.

Address:	DB_HMILITE_CONFIG. SCREEN_COUNTER.TIME_VALUE_HIDE_RESET
Format:	TIME
Range of values:	T#1MS…T#24D20H31M23S647MS
Default setting:	T#5S (5s)

The RESET function will be cancelled after the specified time has expired.

Configuring the text list in WinCC flexible

This SS_50_PartType text list contains the designations of the workpiece types to be displayed in the selection window.

Text list		**SO_62_PartCounterType**
Display		Text
Format		Decimal
Value	1	Workpiece 1 designation
Value	2	Workpiece 2 designation
…	…	…
Value	15	Workpiece 15 designation

Table 6-3 WinCC flexible text list SS_62_PartCounterType

6.2.4 Step-by-step procedure

Step	Proceed as follows
1	Use the data block editor in STEP 7 to open the "DB_HMILITE_CONFIG" data block in the declaration view.
2	Set the SCREEN_COUNTER.HIDE_TYPE_SPECIFIC variable to the value: FALSE = when the type-specific counter is to be displayed TRUE = when the type-specific counter is to be hidden
3	Specify the shift setpoint by editing the variables for OVERALL.SHIFT_TARGET (total counter) and TYPE_SPECIFIC.SHIFT_TARGET_01 to TYPE_SPECIFIC.SHIFT_TARGET_15
4	Now switch to the data view and initialize the data block.
5	Save and close the DB_HMILITE_CONFIG data block.
6	Use the STEP 7 program editor to open the FC_HMILITE_ADDON (FC151) function.
7	Call FC_HMILITE_COUNTER (FC108) and assign the required parameters.
8	Now save and close the FC_HMILITE_ADDON (FC151) function.
9	Load all changed blocks into the controller.
10	Use WinCC flexible to open the WinCC flexible file from HMI Lite.
11	Edit the SO_62_PartCounterType text list and enter the meaningful designations for the workpiece types at the appropriate positions. Delete all text entries that are not used.
12	Save and generate the WinCC flexible project, and transfer it to the operator panel.
13	Create a machine-specific logic for the counting of the workpieces by dynamically changing the parameter of the "FC_HMILITE_COUNTER" (FC108) function: COUNT_VALUE: Total count of the parts to be produced per pulse (good and bad parts) COUNT_VALUE_BAD: Bad parts to be counted TYP_INDEX: Index of the workpiece type to be counted; if only total counter, then "0" COUNT: Counting pulse (increasing edge 0⇨1).
14	Create, if required or necessary, a machine-specific logic for resetting the unit counter. The HMI screen provides the possibility for the manual reset of the workpiece counter.

■

7

7 Diagnostics

7.1 "Messages" screen and "Message archive" screen

7.1.1 Layout of the screen and functionality

The alarm screens display the alarm messages, general messages and system messages in tabular form. The "messages" screen displays the messages and alarms currently pending; the "message archive" screen displays the contents of the alarm buffer.

Alarm events are saved to an internal, non-volatile buffer. The size of this alarm buffer depends on the HMI device type.

The screen structure of the two "messages" and "message archive" screen forms is identical.

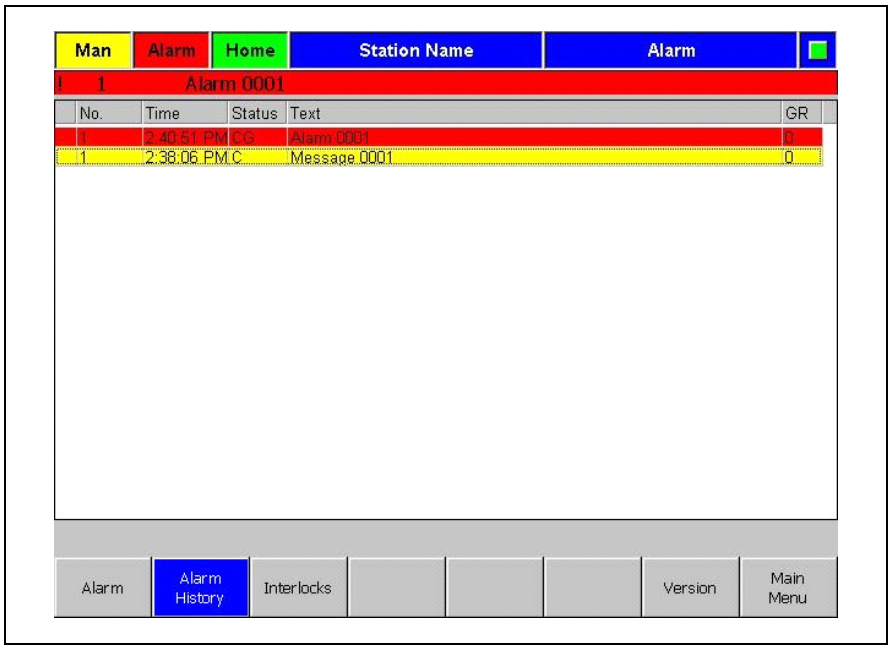


Figure 7-1 "Alarms and messages" screen

The following information is displayed in tabular form:

- whether the message can be diagnosed (yes = * / no = no entry);
- the message number;
- the time stamp of the message;
- the message status (K: arrived, G: sent, Q: acknowledged);
- the message text;
- the acknowledge group.

7.1.2 Runtime interface

The "interface for alarms and messages" data block is defined by the WinCC flexible "SO_00_Alarm" and "SO_00_Message" variables for fault messages and operational messages, respectively.

7.1.3 Configuration

The alarm or message text is configured in the area of the message display object. For further information, refer to the documentation for WinCC flexible.

Settings of the alarm archive

The message display of the alarm view displays selected message events from the alarm buffer. The configuration specifies which events are displayed. The alarm window shows the alarm events selected in the properties dialog.

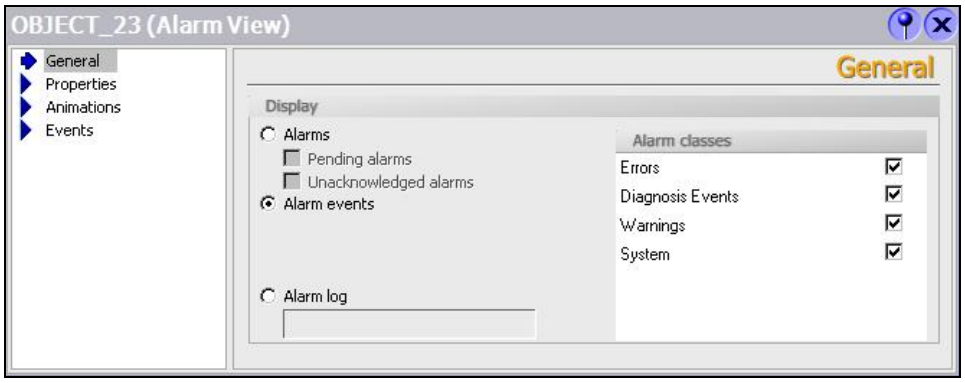
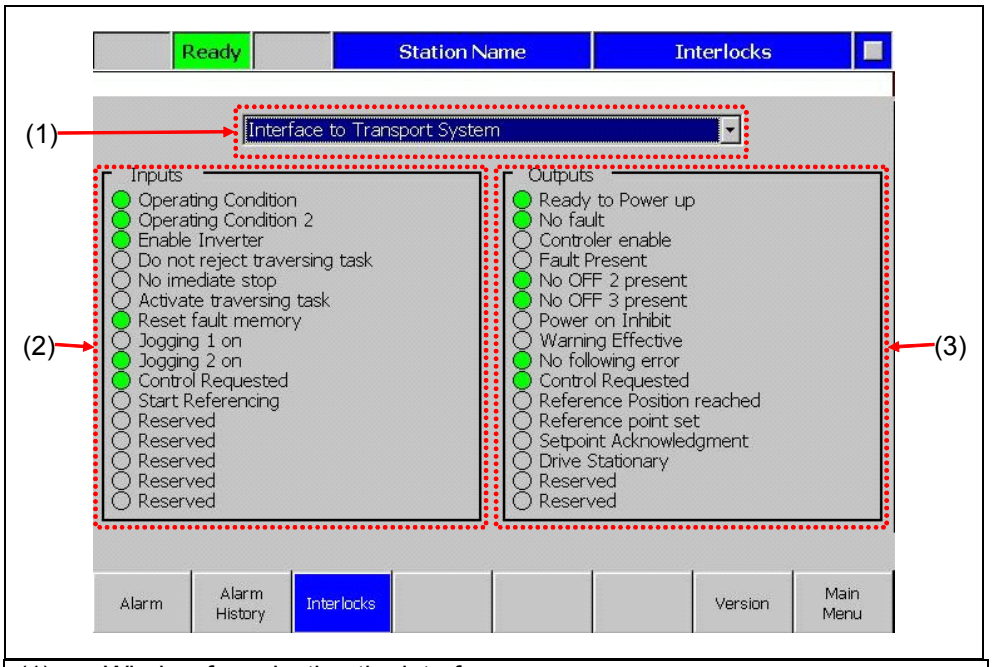


Figure 7-2 Setting for the alarm display object in the "Alarm archive" screen

7.2 "Interface" screen

7.2.1 Layout of the screen

The "Interface" screen can be used to display for diagnostic purposes the signals transferred between the controller and the external devices. Up to sixteen diagnostic interfaces with freely configurable names can be created. Each diagnostic interface can display 16 inputs and 16 outputs. The interface can be selected in a selection window.



(1)	Window for selecting the interface
(2)	Display of the states of the associated inputs
(3)	Display of the states of the associated outputs

Figure 7-3 "Interface" screen

7.2.2 Runtime interface

The runtime interface for the "Interface" screen consists of three variables. The first, DB_HMILITE_DATA.SCREEN_INTERLOCK.SELECTION, represents the current interface selected from the selection window.

Address:	DB_HMILITE_DATA. SCREEN_INTERLOCK.SELECTION
Format:	INT
Range of values:	1..16
Default setting:	1

The inputs/outputs to be visualized must then be copied to the following addresses depending on the currently selected interface:

Address:	DB_HMILITE_DATA SCREEN_INTERLOCK.SIGNALS.INPUT
Format:	WORD
Range of values:	The status of each bit is displayed in the screen by the associated LED element.
Default setting:	-

Address:	DB_HMILITE_DATA SCREEN_INTERLOCK.SIGNALS.OUTPUT
Format:	WORD
Range of values:	The status of each bit is displayed in the screen by the associated LED element.
Default setting:	-

7.2.3 Configuration

Up to sixteen interface descriptions can be defined and selected from a selection window. A WinCC flexible text list can be used to specify a name for each of these interfaces:

Text list		**SO_67_InterlocksSelection**
Display Format		Text Decimal
Value	01	Name for interface no. 1
Value	02	Name for interface no. 2
		…
Value	16	Name for interface no. 16

Table 7-1 Selection window for the interlocks – screen caption of the text list

The following text lists can be used to specify a designation for each input and output of all the interfaces:

Text list		**SO_67_InterlocksInput** **SO_67_InterlocksOutput**
Display Format		Text Decimal
Value	01	Name for input/output #1 of interface #1
Value	02	Name for input/output #2 of interface #1
		…
Value	16	Name for input/output #16 of interface #1
Value	17	Name for input/output #1 of interface #2

Table 7-2 Designation of the input/outputs

7.3 "Version" screen

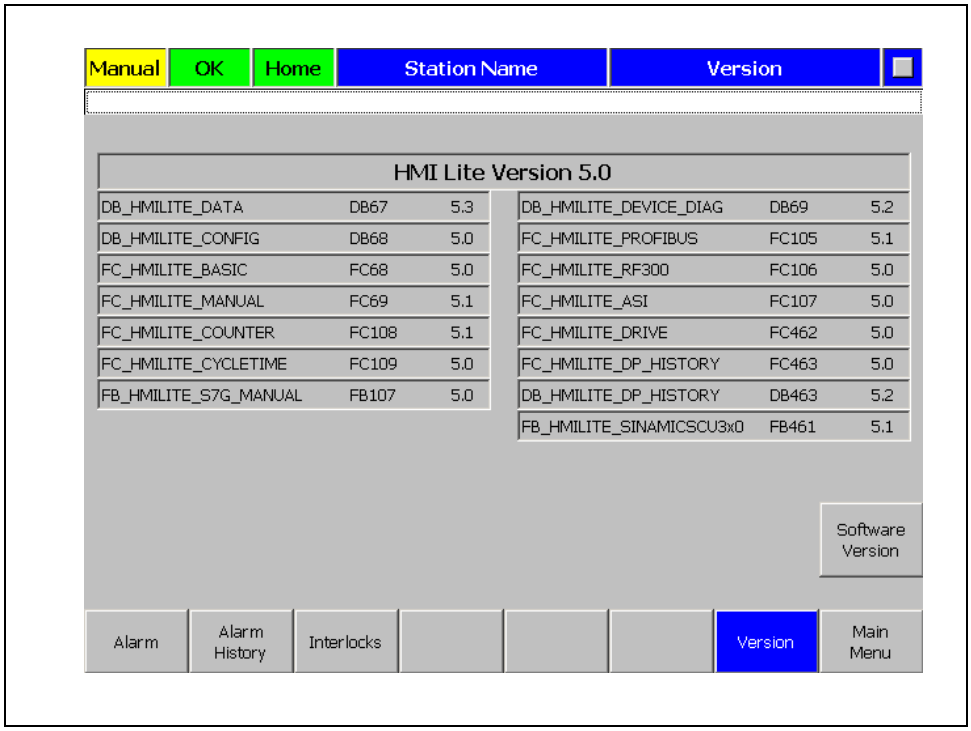


Figure 7-4 "Version display" screen

The "Version" screen displays for diagnostic purposes the associated version of the WinCC flexible screens, the data blocks, the functions and the function blocks of HMI Lite.

The "Software version" button is used to display a window with the version of the WinCC flexible runtime system files (not available for 6" devices!).

No configuring required.

■

8 Hardware Diagnostics

You can branch into the individual diagnostics screens from the hardware diagnostics overview screen. Depending on size, these may again be divided into further substructures.

In the case of PROFIBUS and PROFINET diagnostics, the state of the systems is already shown in the overview selection. The state of the networks is indicated by the color of the keys: green = all nodes OK, yellow at least one node has an error, and red = at least one node is out of operation.

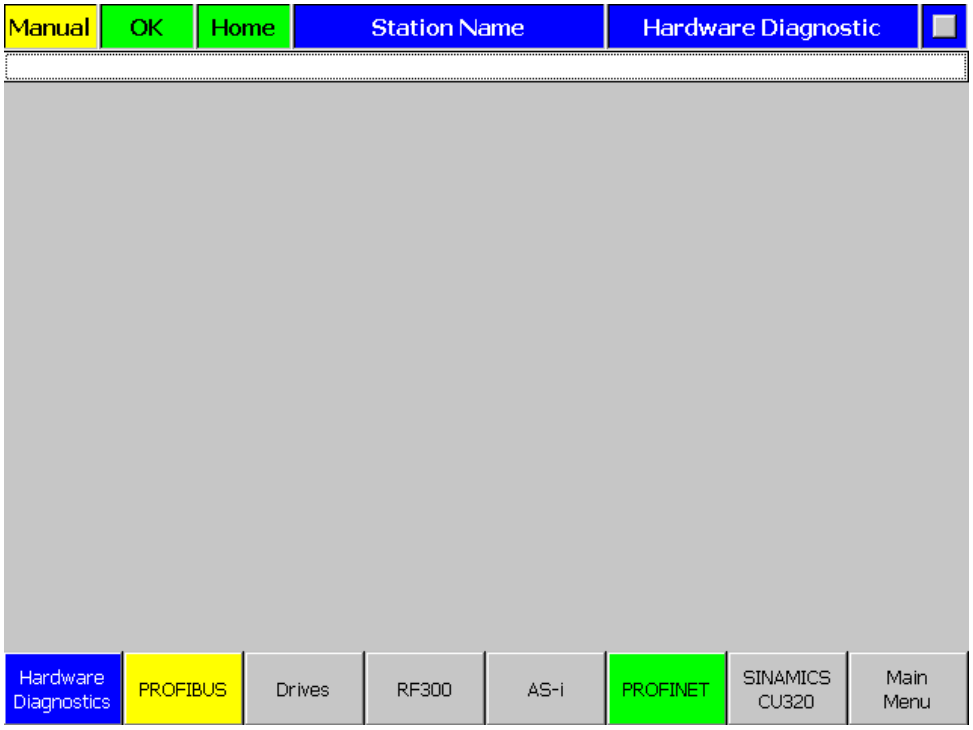


Figure 8-1 Hardware Diagnose

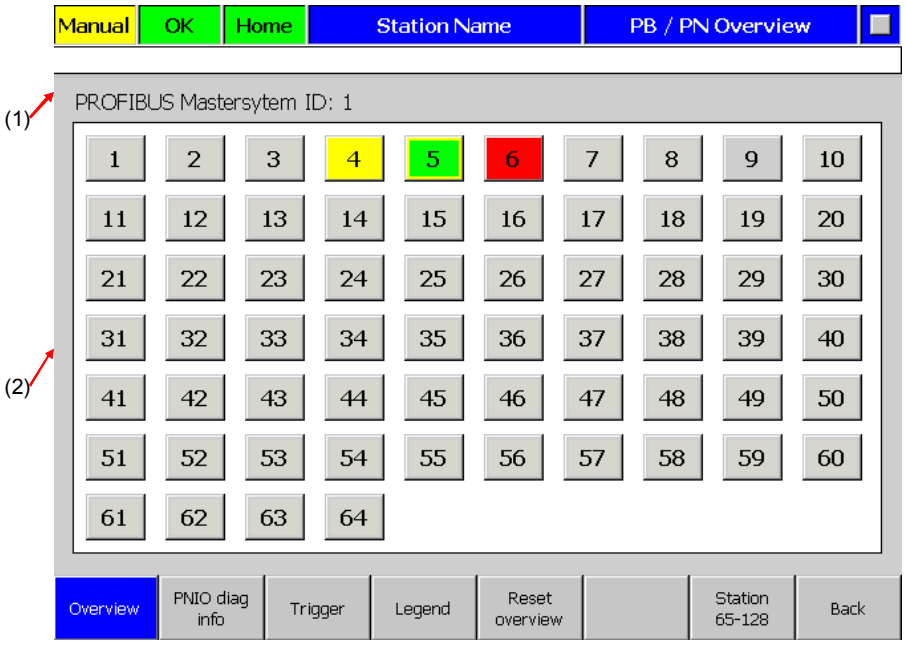
The following sections describe the hardware diagnosis functions in more detail.

8.1 PROFINET/PROFIBUS diagnosis

The "PROFINET/PROFIBUS diagnosis" screens are used to perform a diagnosis of a PROFINET IO or PROFIBUS DP network. They are present only in the 10-inch configuration.

The diagnosis is based on the PNIODiag standard diagnostic package that can be downloaded from the article ID: 26996747 in the Siemens product support. This article ID also provides more detailed documentation.

8.1.1 "PROFINET/PROFIBUS overview" screen



(1)	Select a PROFINET or PROFIBUS station
(2)	Network overview with the configured stations

Figure 8-2 "PROFINET/PROFIBUS overview" screen

The "PROFINET/PROFIBUS overview" screen provides an overview of which configured stations are present on the bus and what state they have.

In each case, an overview page shows a maximum of 64 stations. You can navigate between the individual screens of the station overview using the relevant keys. If screens have to be divided into blocks of 64 stations each, this results in two overview screens for PROFIBUS networks, and a maximum of four for PROFINET networks.

Every station can be selected directly from the nodes recognized on the bus from the hardware configuration (SIMATIC manager) via touch device or the tab key (on the key version), in order to show detailed diagnostic data.

Further PNIO diagnostics screens are also selected from here. The trigger screen, the legend, and the information page branch from the screen call hierarchy from the overview.

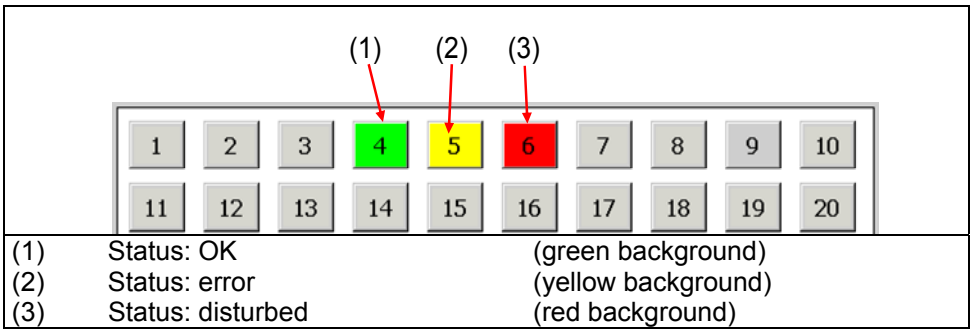
The station overview is updated by pressing the "Reset Overview" key, the status of the system is read out again and the event memory is erased.

8.1.2 Overview: Diagnostics network overview

The status of all configured PN devices of the IO system or DP slaves in the DP master system are displayed in the "Network overview" area of the PROFINET / PROFIBUS overview.

A maximum of 256 devices and maximum 126 slaves can be displayed. Each bus node is represented by its device number (PROFINET) and its address (PROFIBUS) in a status field.

The status is indicated by the background color.



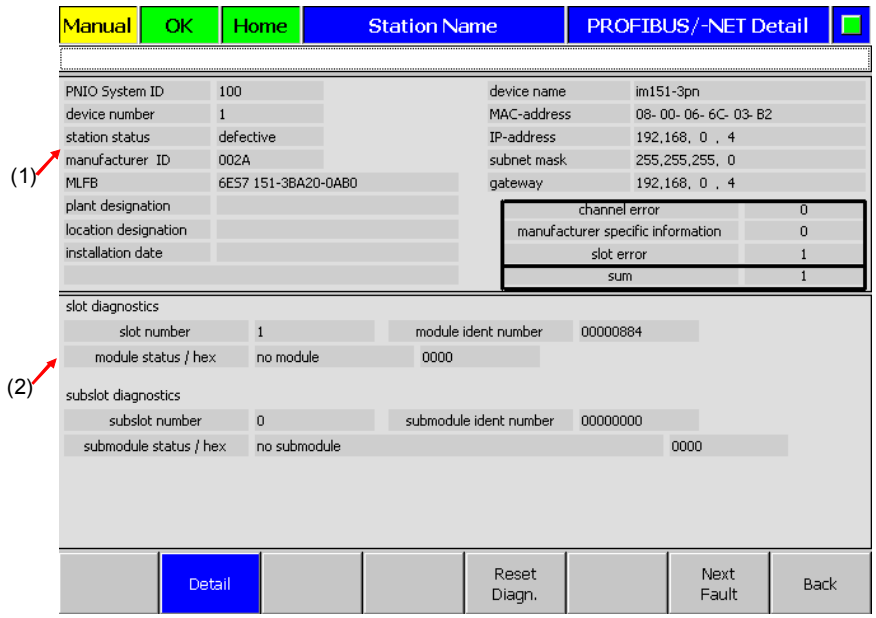
(1)	Status: OK	(green background)
(2)	Status: error	(yellow background)
(3)	Status: disturbed	(red background)

Figure 8-3 PROFINET/PROFIBUS diagnostics - Network overview

The status displays of the bus nodes also contain information about the history of the node. The "previous" status is indicated by a yellow or red colored border. This indicates the status from which the station entered the present status. When the overview is reset, the information is also reset/cleared.

8.1.3 "Detailed diagnosis" screen

The "Detailed diagnosis" screen displays detailed diagnostic information for the selected station and groups them into two main areas.



(1) Station status
(2) Detailed fault information

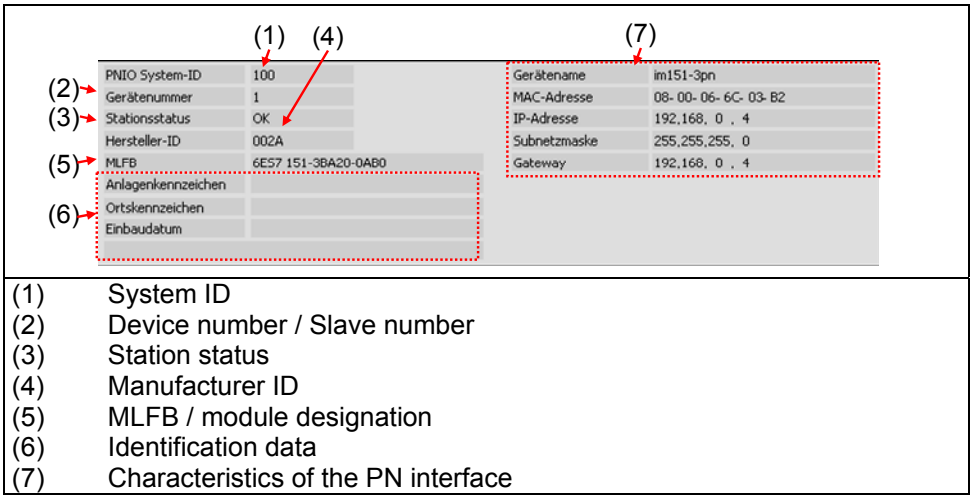
Figure 8-4 "PROFIBUS detailed diagnosis" screen

The "detailed diagnosis" screen opens if a station is selected from the station overview via its number. The screen is divided into several sections that are displayed subject to the status of the station.

If, for example, the station is defective and a diagnosis alarm is triggered, the type of defect and the number of errors are displayed in the top section. The individual errors are described in more detail in the lower section.

Only one diagnosis is displayed at a time. If several diagnoses exist, it is possible to switch between the pending messages using the "Next Fault" softkey. The display and the pending diagnosis data can be updated with the "Reset Diagnosis" key.

Station diagnosis



(1)	System ID
(2)	Device number / Slave number
(3)	Station status
(4)	Manufacturer ID
(5)	MLFB / module designation
(6)	Identification data
(7)	Characteristics of the PN interface

Figure 8-5 PROFINET devices in the detailed diagnosis – station data without errors

The station diagnosis of the screen shows the basic information for the selected station.

This information is:

- ID of the system to which the station is connected
- Device number (station number)
- Station status
- Manufacturer identification
- Module identification
- Number of faults present, sorted according to fault type and total number of faults

Only general information be will shown if no diagnoses are currently present.

As from a particular firmware version, devices on PROFINET and PROFIBUS support the option of entering identification data in the hardware configuration and loading them onto the device. The identification data can then also be read out and displayed via the PNIODiag. The following information is displayed in the Identification screen section:

- System identifier
- Location identifier
- Date of installation
- Additional information.

Note
This data must be loaded from the hardware configuration explicitly with the "Load module identification...“ command from the "Target system" menu, separately for each device (select device).

This procedure must be triggered again if a device is replaced.

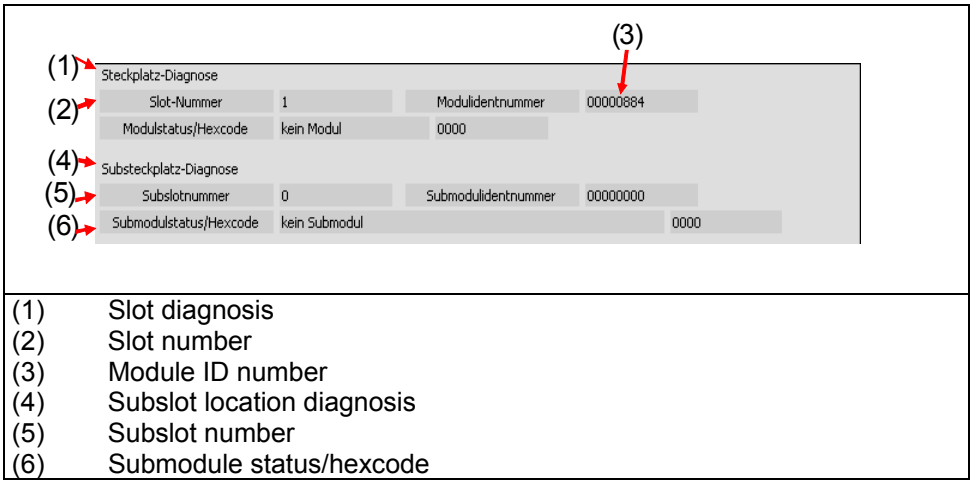
On PROFINET nodes, additional information is visualized.

All settings for the PN interface of the device are displayed:

- Device name
- MAC address
- IP address
- Subnetwork mask
- Gateway.

In addition, each device in the network must be uniquely identified.

Detailed fault information



(1)	Slot diagnosis
(2)	Slot number
(3)	Module ID number
(4)	Subslot location diagnosis
(5)	Subslot number
(6)	Submodule status/hexcode

Figure 8-6 PROFINET devices in a detail diagnosis with errors

The information actually displayed in the "Detailed fault information" screen area depends on the diagnostic capabilities of the selected station.

The fault information supplies the fault location (slot number, subslot number, channel) and the designation of the fault (wire breakage). Additional details, such as the data format and the channel type, are also displayed. The fault information is also displayed in hexadecimal notation.

If required, the device manual can be consulted to obtain the meaning of the hexadecimal value.

A station diagnosis can be fetched for each detected station. The format of the displayed data depends on the station type:

- For PROFIBUS DP slaves, the channel diagnosis is output in text form with the device-specific diagnosis shown in hexadecimal notation.
- The diagnosis of PROFINET IO stations differentiates between direct PROFINET IO devices and PROFIBUS slaves connected with a PROFINET IO system using an IE/PB link:

  ➢ PROFINET device:

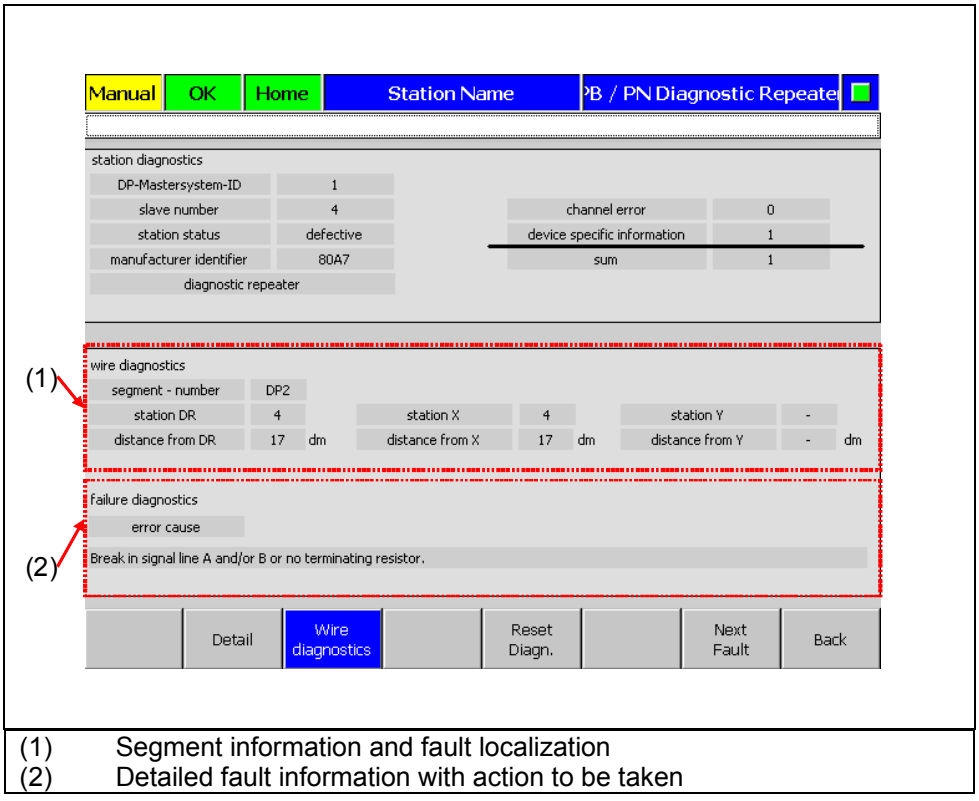
  The channel and slot diagnosis is output in text form. The manufacturer-specific diagnostics are displayed in hexadecimal notation.

➢ PROFIBUS slaves on the IE/PB link:

The diagnostic data (structure, see EN 50 170 Volume 2, PROFIBUS) is shown only in hexadecimal format. Only the status of the module is displayed for an IE/PB link itself.

8.1.4 “Wire diagnostics" display

The "wire diagnostics" display shows detailed diagnostics information in the PROFIBUS network for diagnostic repeaters. The upper section of the display shows the same information as the detail display, in the lower section, the cause of the error and information about the location of the error are shown for the individual diagnosable segments.



(1)	Segment information and fault localization
(2)	Detailed fault information with action to be taken

Figure 8-7 "Wire diagnostics" display

Segment information

The wire diagnostics contains information about the location of the fault for each segment as seen from the diagnostic repeater. The affected segment (segment number), the PROFIBUS address of the diagnostic repeater (Station DR), and the distance of the fault from the diagnostic repeater are displayed.

Fault localization

In data block Station X, information is provided about which station is the last station before the location of the fault, as seen from the diagnostic repeater. The station number (station X) and the distance between the station and the fault location (distance X) are displayed.

Data block Station Y contains information about the station detected behind the fault location. The station number (station Y) and the distance between the location of the fault and station Y (distance Y) help to locate the defect. If no station is detected behind the fault, no data is displayed.

All distances are given in decimeters (dm).

Fault diagnosis with action to be taken

A possible cause of the fault is displayed below the fault diagnosis, together with action for eliminating the fault.

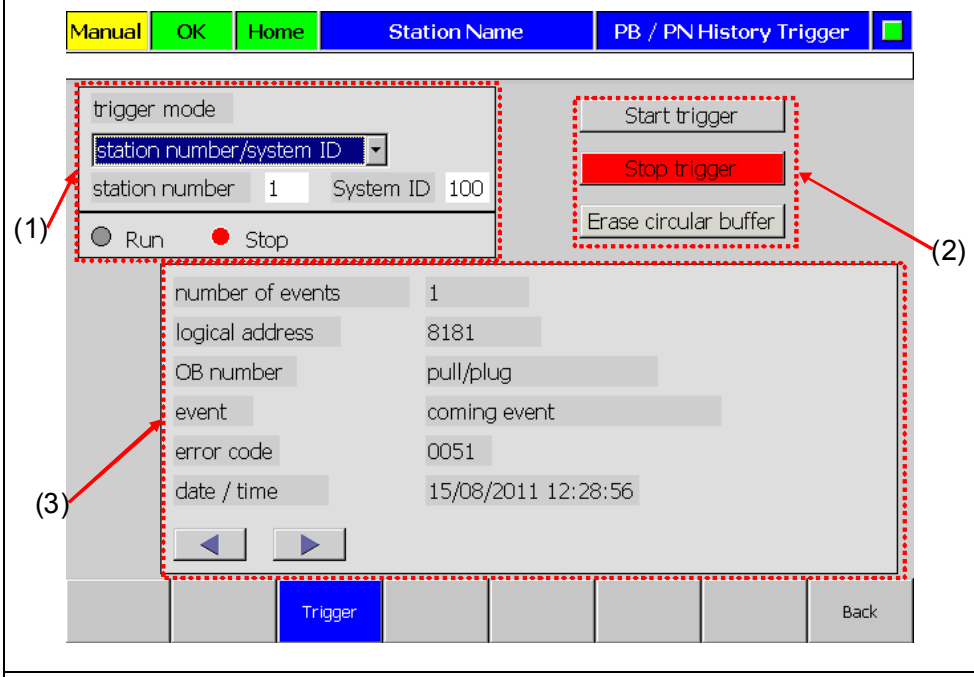
Note

Please also refer to the descriptions given in the manual about the diagnosis repeater. The diagnosis repeater will only provide accurate information about the fault location if it is initialized during installation while the plant or system is running without error (ascertain topology).

8.1.5 "Trigger history" screen

You can set in the "Trigger history" screen which station in which system is to be monitored in order to monitor transient events and also view the recorded events.

This is particularly useful when the events are too fast for them to be evaluated in the station diagnosis. Recording is performed in a ring buffer and can always be started for only one device at a time. Up to 30 events can be stored in the ring buffer.



(1) Set the trigger mode
(2) Operate the trigger
(3) Events display

Figure 8-8 "PROFIBUS detailed diagnosis" screen

Setting the trigger mode

First select the station to be monitored. Trigger mode is set to the selected station and the events are logged in real time. The station is selected either via its logic address or via the station number and subsystem ID. The logic address is the diagnostic address of the device in the hardware config of the SIMATIC manager. The station number corresponds to the PROFIBUS address and the device number of the PROFINET I/Os. The corresponding system ID has the standad value 1 on PROFIBUS networks and 100 on PROFINET networks.

The address can only be entered in the stop state and only one trigger event may be active at a time.

Operating the trigger

Press "Start trigger" to activate the trigger. The state remains active even for a change to a different screen and continues to run in the background. The trigger is running when the background is green.

Press "Stop trigger" to deactivate the trigger after which the events can be viewed. The trigger is stopped when the background is red.

Press "Clear ring buffer" to clear the ring buffer. To prevent data falsification, clear the ring buffer before using the trigger and when the address is reset. Clearing the ring buffer places the trigger in the stopped state should it be active.

Displaying the events

In this window you can display the recorded events.

If several events are present, the arrows can be used to scroll left and right. The arrows are enabled only when the trigger is in the stopped state.

8.1.6 "Legend" screen

The "Legend" screen shows the states that a station can assume in the network overview.

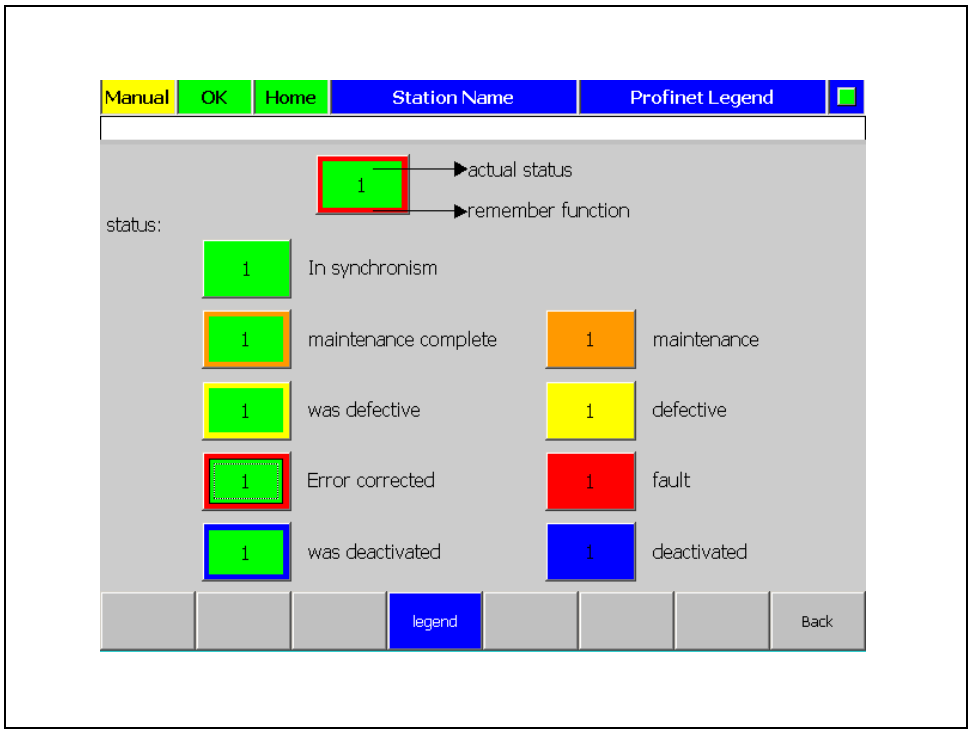


Figure 8-9 "PROFINET/PROFIBUS overview" legend screen

8.1.7 PNIODiag Info Screen

The "PNIODiag" info screen shows information about and the version number of the diagnostics package. In addition, a message window is supplied with messages that only refer to the PNIODiag. They can be cleared again with the "confirm" key at the bottom of the screen.

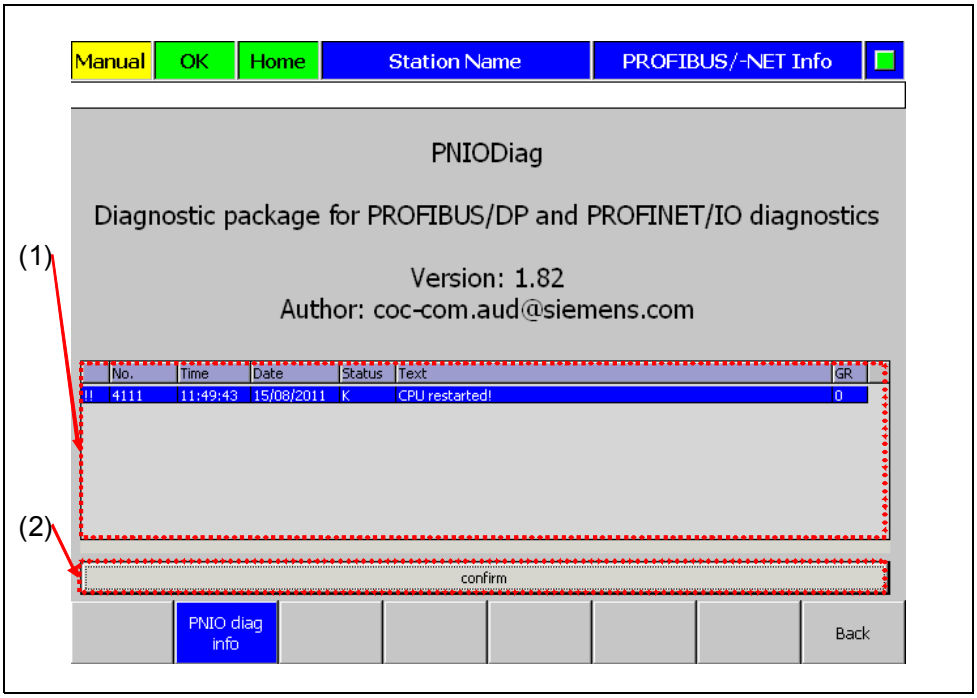
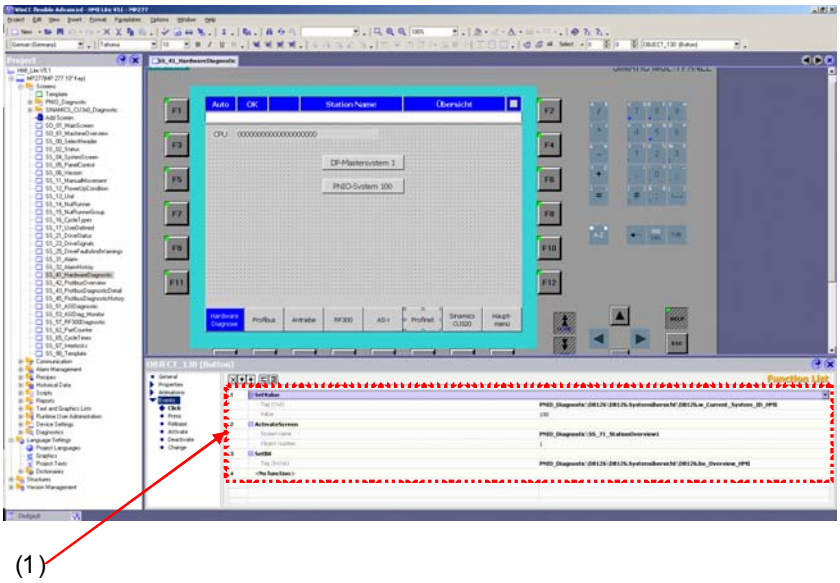


Figure 8-10 Screen „PNIO Diag Info“

8.1.8 Configuring the WinCC flexible screens

Selecting the PROFINET/PROFIBUS network for the diagnosis



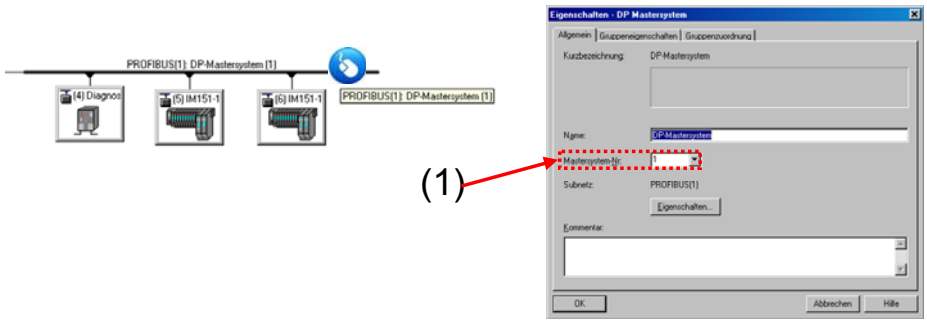
(1)	Functions for selecting the ROFINET or PROFIBUS diagnosis

Figure 8-11 Functions for selecting the PROFINET or PROFIBUS diagnosis

The PROFINET/PROFIBUS diagnostic screens can be fetched for various PROFINET or PROFIBUS networks.

The WinCC flexible "PNIO_Diagnostic\DB126\DB126.System overview\DB126.w_Current_System_ID_HMI" variable specifies the associated network for the call. This variable is set for a PROFINET network to the IO system number or for a PROFIBUS network to the master system number. These numbers are assigned in the Step7 hardware configuration. The default value of the master system number for PROFIBUS and PROFINET is 1 or 100, respectively.

When screens are called into the PNIO-Diag, the variable under Events must be supplied with the configured value from the hardware config.



(1) Auswahl der System-ID

Figure 8-12 Defining the DP master system in the hardware configuration

.

Address:	WinCC flexible variables: PNIO_Diagnostic\DB126\DB126.System overview\DB126.w_Current_System_ID_HMI
Format:	WORD
Values range:	For PROFINET: IO system number (default value = 100) For PROFIBUS: master system number (default value = 1) The values are configured in the hardware configuration of the SIMATIC station.

8.1.9 Runtime interface (FB_PNIODiag)

The PNIODiag function block can be used for the diagnosis of PROFINET IO and PROFIBUS DP systems. For this purpose, the PNIODiag function block determines general system status information and, if required, detailed diagnostic information. This information is then displayed using the visualization.

The PNIODiag function block provides the following functions for PROFIBUS DP and PROFINET IO systems:

- Obtain an overview of the state of the individual systems
- Show the station states as overview diagnosis
- Detailed diagnosis of the individual stations
- Save the transient state changes
- Record the diagnostic events
- Deactivate and activate PROFIBUS DP slaves and PROFINET IO devices

The PNIODiag function block call must be made in the OB1, OB82, OB83, OB86 and OB100 (OB101, OB102) execution levels. The order and content of the 20-byte temporary local data of the organization blocks must not be changed.

The IN0 input parameter specifies the DB number 466 for saving the diagnostic data. Because the creation of the data block for the data storage is specified in OB100, the DB number must also be specified in the OB100. If no value is specified, the DB200 data block will be created automatically.

Should an error occur while processing the block, the "OUT1" output parameter will contain an error code that helps the Support department with troubleshooting.

Note

The required S7 data blocks and the S7 function block do not contain any symbol information. An interface description for the S7 modules is not available. The S7 modules should be used only in conjunction with the provided visualization.

Note

The download of the PNIO standard package contains more detailed documentation for the FB_PNIODIAG (see Introduction).

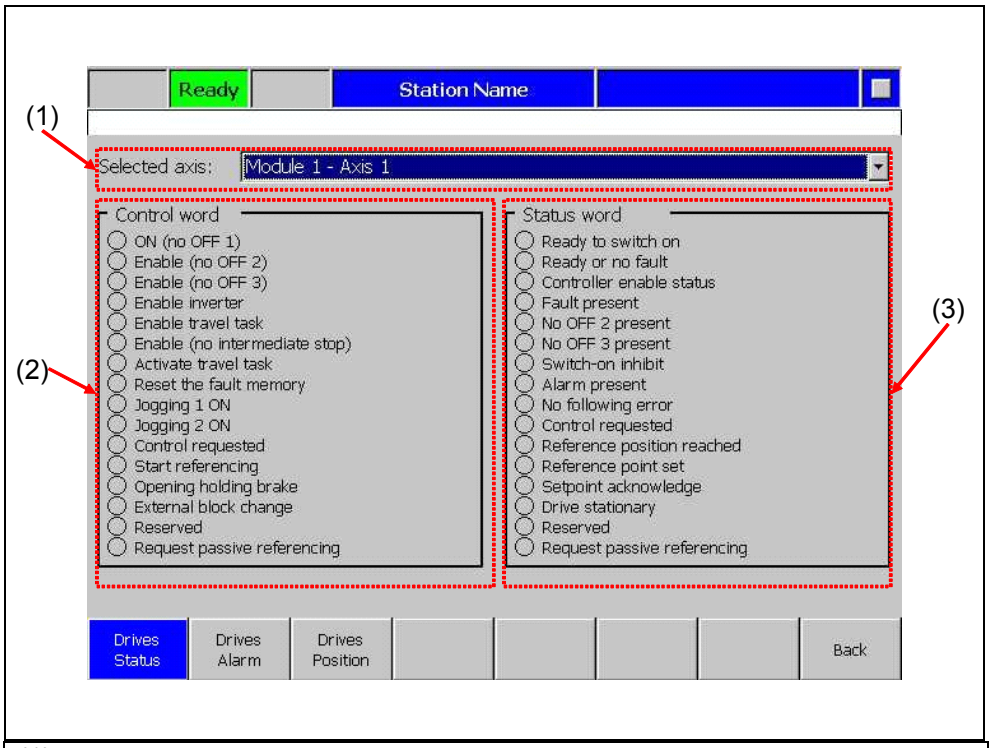
Note

In the PNIO standard package, the FB_PNIODIAG is also provided as a variant for CPUs with a maximum block size of 16 KB.

8.2 "Drive" screen

8.2.1 "Drive status" screen

The "Drive status" screen displays the control and status signals of the axis selected from the selection window.

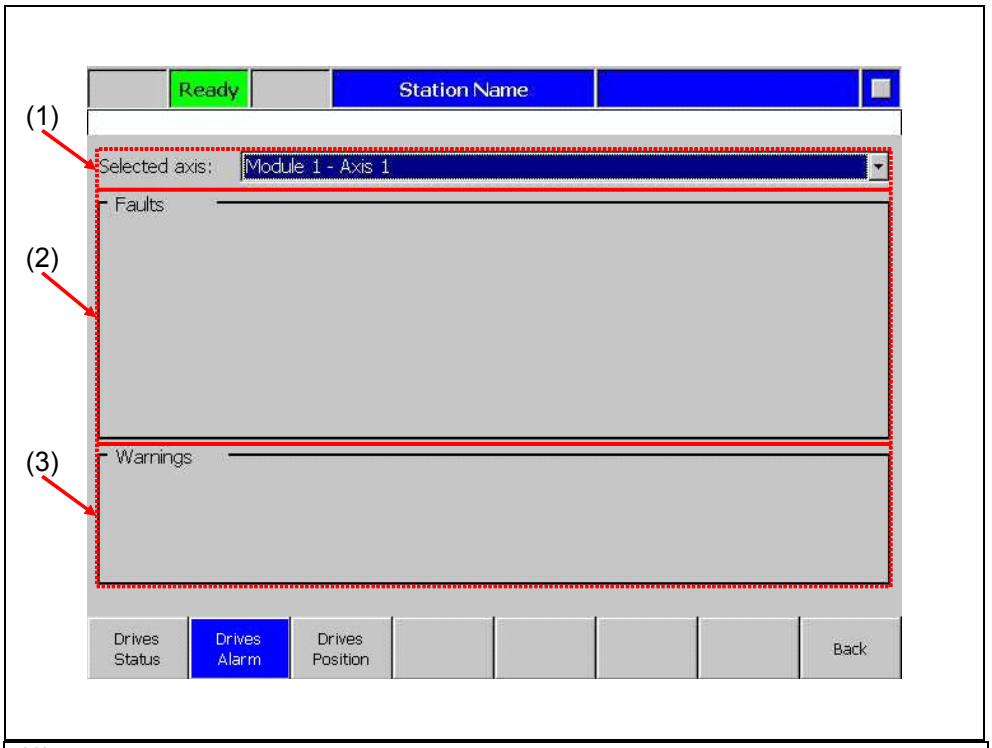


(1)	Selection of the axis (selection window)
(2)	Control signals of the selected axis
(3)	Status signals of the selected axis

Figure 8-13 "Drive status" screen

8.2.2 "Drive alarms" screen

The "Drive alarms" screen displays the malfunctions and warnings of the selected axis.

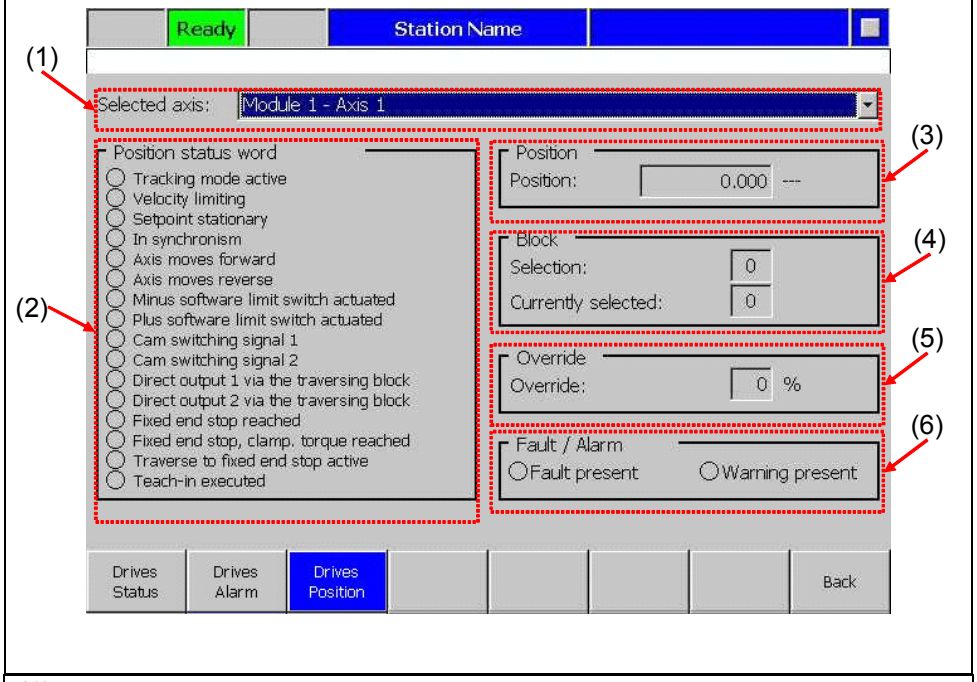


(1)	Selection of the axis (selection window)
(2)	Area for displaying the malfunctions
(3)	Area for displaying the warnings

Figure 8-14 "Drive alarms" screen

8.2.3 "Drive position" screen

The "Drive position" screen displays the Positioning and Positioning data status signals, such as position, block and override of the selected axis. The data, however, is available only for the axes when they are operated as positioning axes.



(1)	Selection of the axis (selection window)
(2)	Positioning status signals
(3)	Display of the axis position
(4)	Number of selected block
(5)	Override
(6)	Display of any pending fault/warning

Figure 8-15 "Drive position" screen

Note

When a Micro Master is used, the information will not be supplied to the Position screen. Only the 611U drive supports the functionality and displays the data. The data will be cleared for other drives.

8.2.4 Configuring the WinCC flexible screens

Configuring the text list in WinCC flexible

The designation of the axes must be specified. The text items are stored in the WinCC flexible "SO_20_AxisName" text list. Each configured axis must have a position in the text list.

The "SO_20_AxisName" text list has the following structure:

Text list		**SO_20_AxisName**
Display Format		Text Decimal
Value	1	Designation of the first axis
Value	2	Designation of the second axis
…	…	…

Table 8-1 Text list for the axis designations

! Important

Designation text items for missing axes must be deleted!

8.2.5 Runtime interface (FC_HMILITE_DRIVE)

The FC_HMILITE_DRIVE supplies data to the WinCC flexible screens for the drives. The displayed data is fetched directly from the drive using the PKW interface.

The user must call this function once cyclically in the OB1 for each configured axis, where ACHS_INDEX matches the corresponding values from the WinCC flexible "SO_20_AxisName" text list (see sample program).

A message frame type with PKW interface must be specified for each axis in the HW Config. The input and output address of an axis must be identical.

When positioning axes are used on a Simodrive 611U, the following additional data must be entered in the PZD interface:

- The selected block number (setpoint) and the current block number (actual value) must be transferred to the second word.
- The positioning status word must be transferred to the third word.

This corresponds to the standard assignment of the 101, 108 and 109 message frames in the "positioning" operating mode.

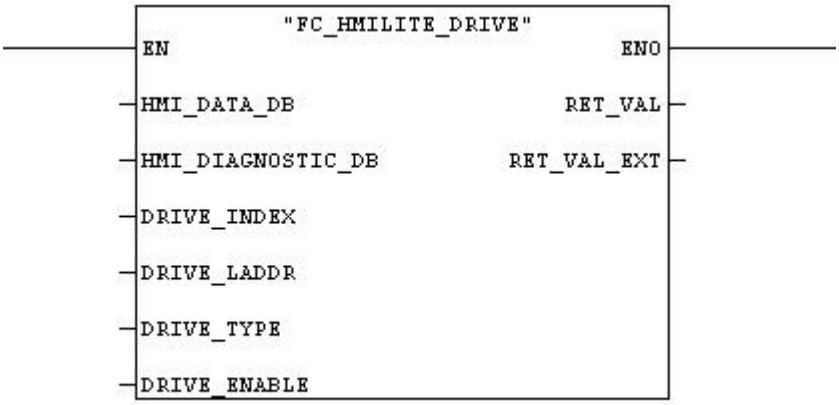


Figure 8-16 Call interface of the FC_HMILITE_DRIVE function

Name	Type	Default	Example	Description
HMI_DATA_DB	INT	67	67	The number of the HMI runtime data block
HMI_ DIAGNOSTIC_DB	INT	69	69	The number of the HMI diagnostic data block
DRIVE_INDEX	INT	--	--	Selection of the axis (the value comes from the "SO_20_611UAxisName" text list)
DRIVE_LADDR	INT	---	---	Address of the PKW interface (the input and the output address must be identical)
DRIVE_TYPE	INT	---	---	Drive type> 0: Simodrive 611U 1: Micromaster
DRIVE_ENABLE	BOOL	TRUE	---	"TRUE" releases the communication of the block with the drive using the PKW interface.
RET_VAL	WORD	---	---	Function return value 0000: no error 7000: block not processed 8096: invalid axis type 8102: invalid response from the drive 8103: drive reports error, error no. in RET_VAL_EXT 8104: run time error in the data transmission to the drive 82xx: internal error, call the screen again 8Cyy: SFC yy error, error no. in RET_VAL_EXT
RET_VAL_EXT	INT	---	---	Return value of a system function or error number of the drive (also see RET_VAL)

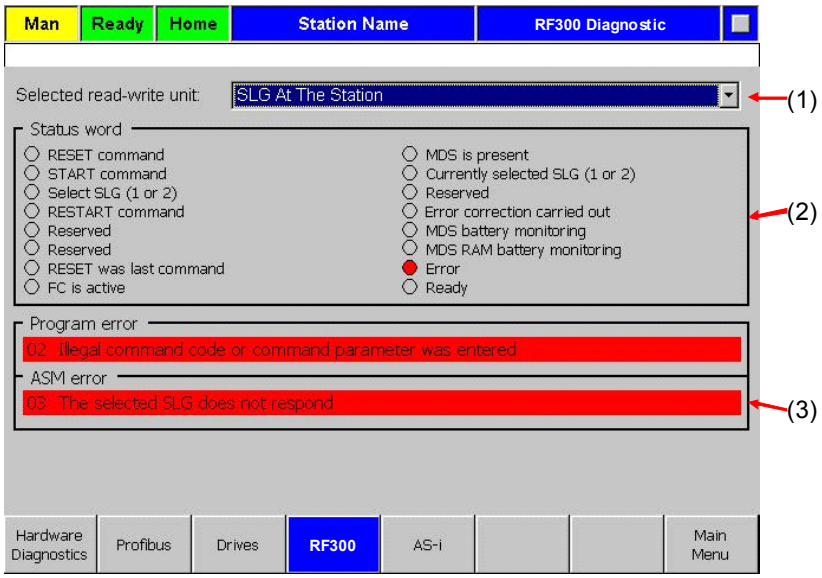
Table 8-2 Parameters of the FC_HMILITE_ANTRIEB function

8.3 "RF300" screen

The "RF300 diagnostics" screen shows the status signals and error messages of a read-write device (SLG). The data is fetched and assigned by a standard block from the Moby interface module (ASM).

The signals and error messages from several interface modules can be displayed in the screen.

8.3.1 Layout of the screen and functionality



(1)	Select the RF300 read/write device (selection window)
(2)	Status signals of the RF300 SLG
(3)	Error messages of the RF300 SLG

Figure 8-17 "RF300" screen

Selection of the RF300 SLG

You can select several RF300 SLGs in the selection field. Each RF300 SLG provides its own interface.

Status of the SLG

"Status of the SLG" area displays the states of the selected SLG.

RF300 SLG error messages

The "RF300 SLG error messages" area displays RF300-specific error messages.

8.3.2 Supported RF300 interface modules

The following table contains a list of all supported RF300 interface modules with the associated FC:

Interface Module	Block
ASM 450	FC 44
ASM 452 ASM 456 ASM 473 ASM 475	FC 45
ASM 452 - file handler	FC 46

Table 8-3 Supported RF300 interface modules

The data exchange between the controller and the ASM is performed using the FC44, FC45 or FC46 function, depending on the ASM type. The control and feedback signals to and from the ASM, and the error messages can be found in the so-called command data block.

The "RF300 diagnostics" screen displays the information contained in these command data blocks.

The structure of the command data block depends on the associated block.

8.3.3 Configuring the WinCC flexible screens

Configuring the text list in WinCC flexible

The designations of the read-write devices must be specified. The text items are stored in the WinCC flexible "SO_57_RF300SLGName" text list. Each configured RF300 SLG must have an entry in the text list.

The "SO_57_RF300SLGName" text list has the following structure:

Text list		SO_57_RF300SLGName
Display Format		Text Decimal
Value	1	Designation of the first RF300 read-write device
Value	2	Designation of the second RF300 read-write device
…	…	…

Table 8-4 Text list for the designations of the RF300 SLGs

Important

The text items for non-configured (unused) RF300 SLGs must be deleted.

8.3.4 Runtime interface (FC_HMILITE_RF300)

The FC_HMILITE_RF300 supplies data to the WinCC flexible screens for the RF300 diagnostics. The displayed data is fetched from the command data block.

The user must call this function once cyclically in the OB1 for each configured RF300 SLG, where RF300_INDEX matches the corresponding values from the WinCC flexible "SO_57_ RF300SLGName" text list (see sample program).

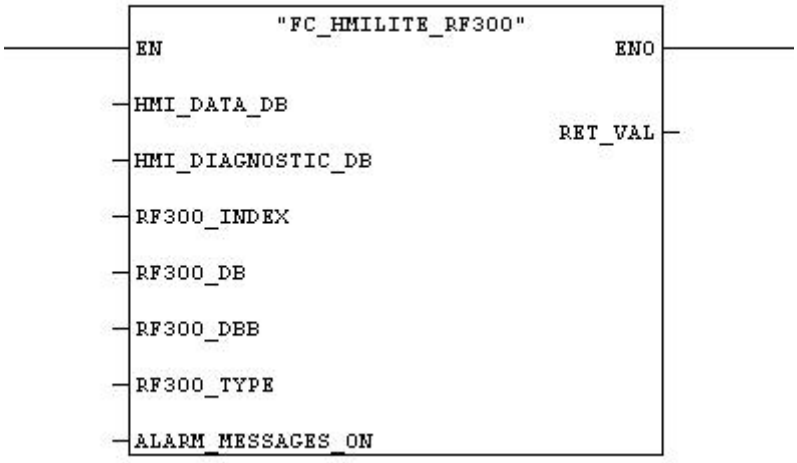


Figure 8-18 Call interface of the FC_HMILITE_RF300 function

Name	Type	Default	Example	Description
HMI_DATA_DB	INT	67	67	The number of the HMI runtime data block
HMI_ DIAGNOSTIC_DB	INT	69	69	The number of the HMI diagnostic data block
RF300_INDEX	INT	--	--	Selection of the SLG (the value comes from the "SO_57_RF300SLGName" text list)
RF300_DB	INT	---	---	The number of the RF300 command data block
RF300_DBB	INT	0	0	The start address of the SLG in the command data block
RF300_TYPE	INT	---	---	RF300 block type: 1: FC44 - Word 2: FC45 3: FC46 5: FC44 - Byte
ALARM_ MESSAGES_ON	BOOL	---	---	When TRUE, the FC generates a message when a fault is present at the SLG.
RET_VAL	WORD	---	---	Function return value 0000: no error 7000: block not processed 8096: invalid RF300 type

Table 8-5 Parameters of the FC_HMILITE_RF300 function

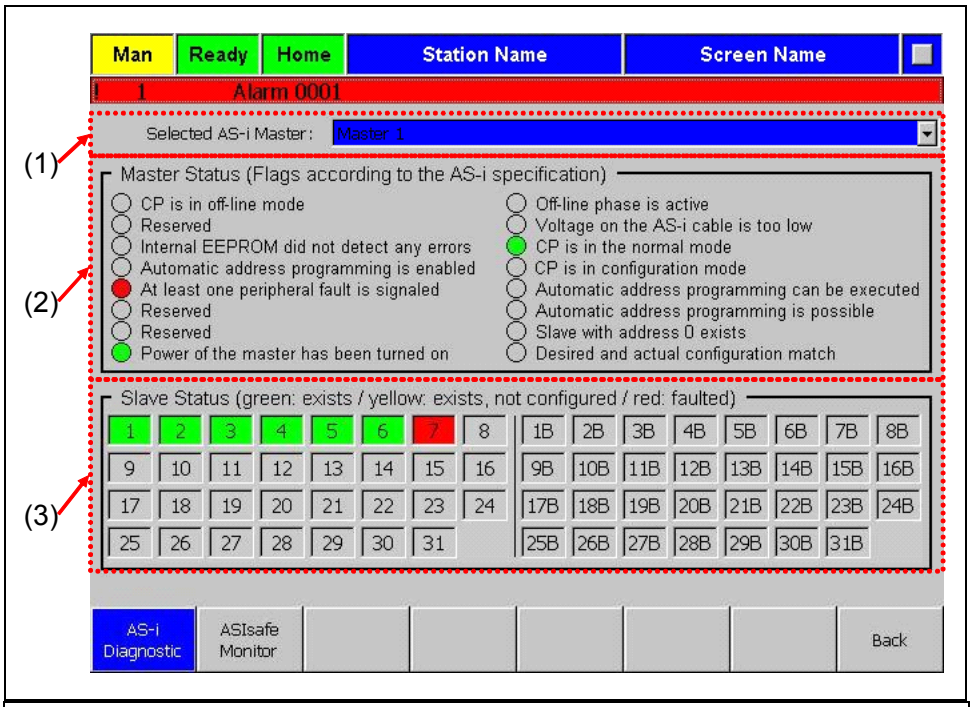
8.4 "AS-i" screen

8.4.1 "AS-i diagnostics" screen

The "AS-i diagnostics" screen provides a user interface for performing diagnostics for an AS-i master.

The screen shows the status signals and the AS-i slaves connected to the bus. The data is fetched from the AS-i master and assigned.

The screen is divided into three areas:



(1)	Selection of the AS-i master
(2)	Status of the AS-i master
(3)	Overview and states of the slaves connected to the bus

Figure 8-19 "AS-i diagnostics" screen

Selection of the master

Several AS-i master modules can be selected in the selection field. Each AS-i master provides its own bus.

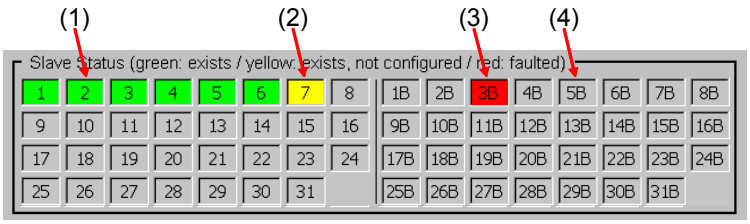
Status of the master

The states of the communications processor of the selected AS-i master are displayed in the "Status of the master" area. The status signals conform to the AS-i specification.

Status of the slaves

The "Status of the slaves" displays an overview of the addresses of the configured slaves. A maximum of 62 A/B slaves can be displayed (see AS-i description, 2.1). Each bus node is represented by its address in a status field.

The slave status is indicated using various background colors.



(1)	Slave status: present (green background)
(2)	Slave status: present but not configured (yellow background)
(3)	Slave status: faulty (red background)
(4)	Slave status: not configured (gray background)

Figure 8-20 Status of the slave on the AS-i master

8.4.2 Configuring the WinCC flexible screens

Configuring the text list in WinCC flexible

The designation of the communications processors of the AS-i must be configured. The text items are stored in the WinCC flexible "SO_51_ASIIndex" text list. Each configured AS-I master must have a position in the text list.

Important

Designation text items for missing communications processors must be deleted.

Structure of the "SO_51_ASIMasterName" text list:

Text list		**SO_51_ASIIndex**
Display Format		Text Decimal
Value	1	Designation of the first AS-I communications processor
Value	2	Designation of the second AS-I communications processor
…	…	…

Table 8-6 Text list for the designations of the AS-I masters

8.4.3 Runtime interface (FC_HMILITE_ASI)

The FC_HMILITE_ASI supplies data to the WinCC flexible screens for the AS-i diagnostics. The displayed data is fetched directly from the AS-i master.

The user must call this function just once cyclically in the OB1 for each AS-i master, where ASI_INDEX matches the corresponding values from the WinCC flexible "SO_51_ASIIndex" text list (see sample program).

For the error diagnostics, the FC must also be called once in the OB82. In this case, the "HMI_DATA_DB" and "HMI_DIAGNOSTIC_DB" parameters must be specified appropriately for their function. The "OB82_CALL" parameter must also have the fixed value "1".

! **Important**

The FC_HMILITE_ASI must be called directly in the OB82 and must not be called indirectly by another FC or FB.

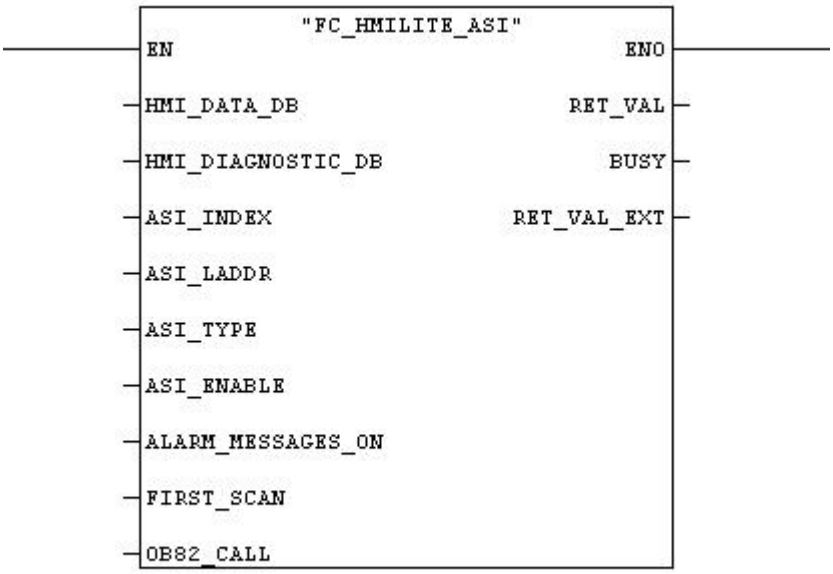


Figure 8-21 Call interface of the FC_HMILITE_ASI function

Name	Type	Default	Example	Description
HMI_DATA_DB	INT	67	67	The number of the HMI runtime data block
HMI_ DIAGNOSTIC_DB	INT	69	69	The number of the HMI diagnostic data block
ASI_INDEX	INT	--	--	Selection of the AS-i master (the value comes from the "SO_51_ASIIndex" text list)
ASI_LADDR	INT	---	---	The start address of the AS-i master; the input and the output address must be identical
ASI_TYPE	INT	---	---	Type of the AS-i master: 1: CP142-2 (ET200X) 2: CP342-2 (S7-300, ET200M) 3: CP343-2 (S7-300, ET200M) 4: DP/AS-i Link 20E 5: DP/AS-i Link 65
ASI_ENABLE	BOOL	TRUE	---	"TRUE" releases the communication of the block with the AS-i master.
ALARM_ MESSAGES_ON	BOOL	---	---	When TRUE, the FC generates a message when a fault is present at the SLG.

Name	Type	Default	Example	Description
FIRST_SCAN	BOOL	---	---	Indicates a restart of the controller. Must have the "TRUE" value in the first cycle.
OB82_CALL	BOOL	---	---	"TRUE" must be set when the call is made from the OB82.
RET_VAL	WORD	---	---	Function return value 0000: no error 7000: block not processed 8096: invalid ASI type 8101: unexpected AS-i master status 82xx: internal error, call the screen again 8Cyy: SFC yy error, error no. in RET_VAL_EXT
BUSY	BOOL	---	---	Indicates that the FC is communicating with the AS-i master
RET_VAL_EXT	INT	---	---	Return value of a system function or error number of the drive (also see RET_VAL)

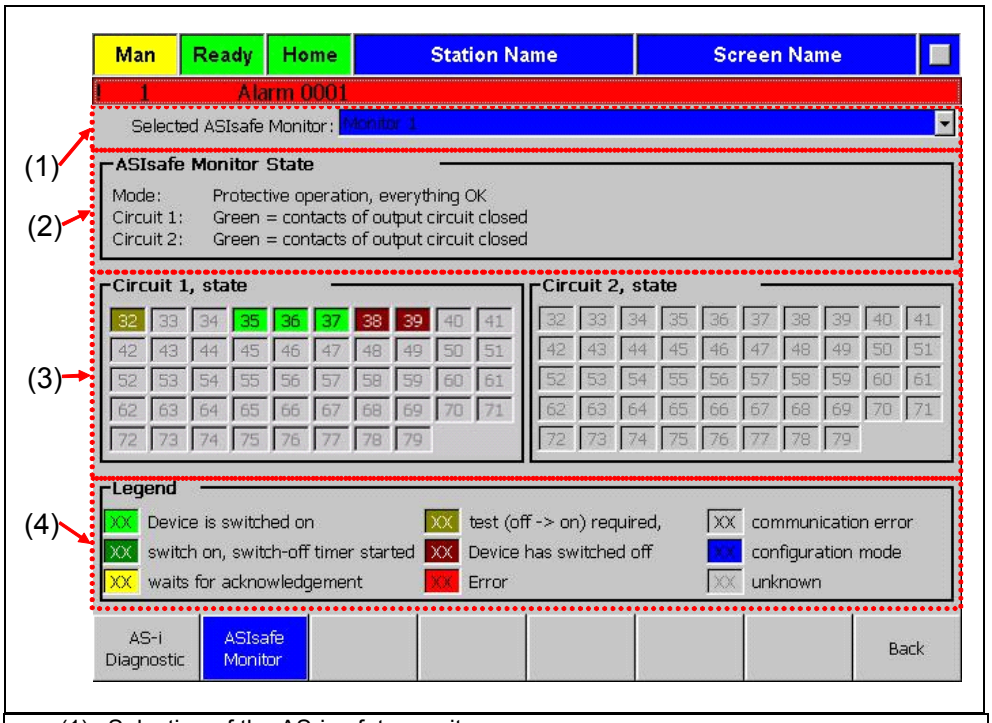
Table 8-7 Parameters of the FC_HMILITE_ASI function

8.4.4 "ASIsafe monitor" screen

The "AS-i safety monitor" screen provides a user interface for performing diagnostics of an AS-i safety monitor.

The screen displays the operating mode of the safety monitor and the status of the individual devices. The data is fetched from the AS-i safety monitor and assigned.

The screen is divided into four areas:



(1) Selection of the AS-i safety monitor
(2) Status of the safety monitor
(3) Status of the individual devices
(4) Legend

Figure 8-22 "ASIsafe Monitor" screen

Selection of the safety monitor

Several AS-i safety monitors can be selected in the selection field.

Status of the safety monitor

The "Status of the safety monitor" area displays the operating mode and the status of the two safety circuits of the selected safety monitor.

Status of the individual devices

The "Status of the individual devices" area visualizes an overview of the 32 – 79 devices of a safety monitor. Each device is represented by its address in a status field.

The slave status is indicated using various background colors.

Legend

The "Legend" area is used to specify the assignment of the color combination to the status of the device.

8.4.5 Configuring the WinCC flexible screens for ASIsafe Monitor

Configuring the text list in WinCC flexible

The designations of the various safety monitors of the AS-interface must be configured. The text items are stored in the "SO_53_ASIMonitor" text list. Each safety monitor that exists must have a position in the text list.

! Important

Designation text items for missing safety monitors must be deleted.

Structure of the "SO_53_ASIMonitor" text list:

Text list		SO_53_ASIMonitor
Display Format		Text Decimal
Value	1	Designation of the first safety monitor
Value	2	Designation of the second safety monitor
…	…	…

Table 8-8 Text list for the designations of the safety monitors

8.4.6 Runtime interface (FB_ASIMON2D)

The diagnostic data of the AS-i safety monitor is fetched using the "ASIMON2D" FB. The user must call this function block once cyclically in the OB1 for each configured AS-i safety monitor.

Depending on the safety monitor selected in the screen, the diagnostic data of the FB_ASIMON2D ("memory_copy_diagnosis" parameter) must be copied into the "DB_ASI_DIAGNOSE" (DB69) data block in the "DIAGNOSE_DATEN" (DBW154) DB area.

In the example, this is done using SFC 20 (BLKMOV).

The copying of the diagnostic data, however, can also be enabled using the following FB parameters.

- "enable_copy_diagnosis" input parameter:

  The input enables the copying of the diagnostic data to the "DB_ASI_DIAGNOSE" DB. It must have the TRUE value when the valid safety monitor for this FB call is selected in the screen.

- "indirect_copy_diagnosis" input parameter:

  The input specifies how the address is stored at the "memory_copy_diagnosis" input in the "DB_ASI_DIAGNOSE" DB. Because this is specified directly, "false" must always be specified here.

- "memory_copy_diagnosis" input parameter:

  A pointer to the data area of the diagnostic data in the "DB_ASI_DIAGNOSE" (DB69) DB must be specified here. The data area (DIAGNOSE_DATEN) starts at address 154 in the DB and is 182 bytes long.

Note

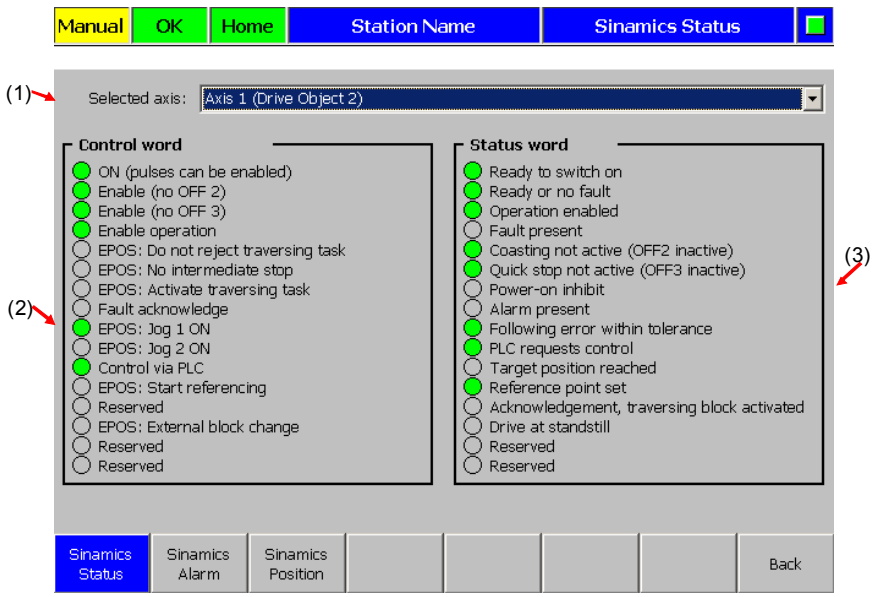
The project is contained on the Software CD with the title "Tools for Control & Distribution - AS-Interface Safety at Work". The archive file in Version 1.0 is stored with the file name "ASiMon2D-v10.zip".

The detailed description of the FB_ASIMON2D is contained in the documentation "ASIMON2D_DIAG_v1.0_deu_eng.pdf".

8.5 "SINAMICS" screen

8.5.1 "SINAMICS status" screen

The "SINAMICS status" screen shows the control and status signals of the SINAMICS axis selected from the selection window.

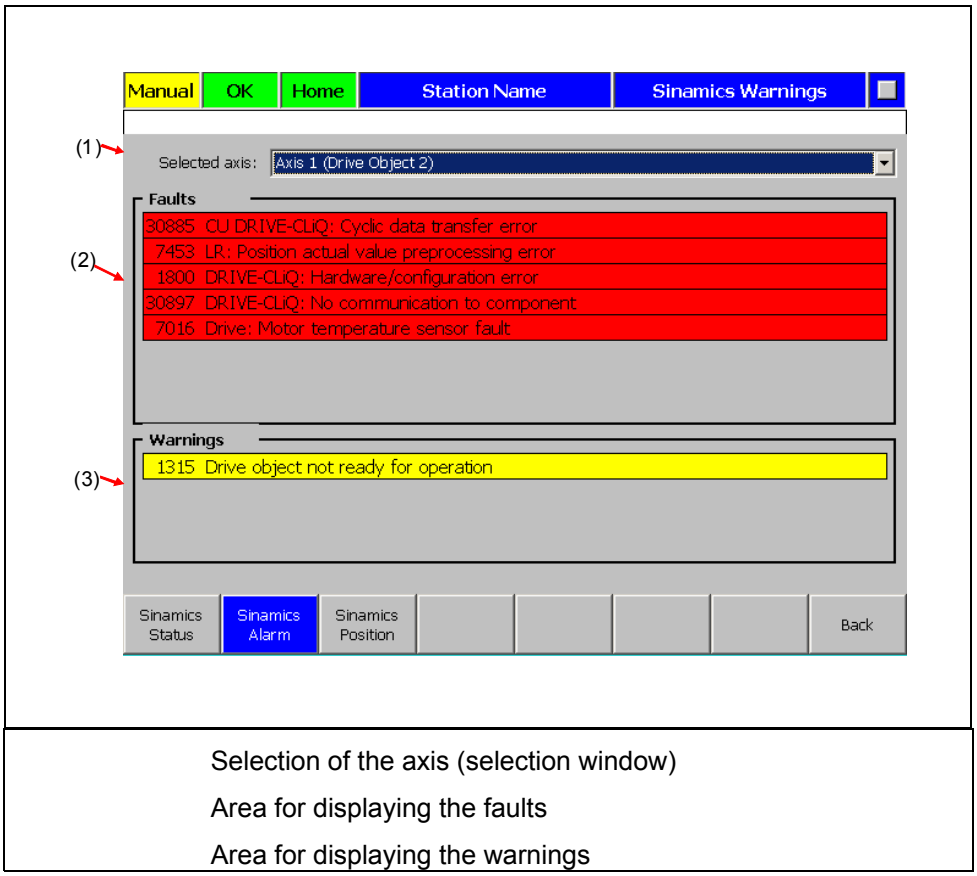


(1) Selection of the axis (selection window)
(2) Control signals of the selected axis
(3) Status signals of the selected axis

Figure 8-23 "SINAMICS status" screen

8.5.2 "SINAMICS alarms" screen

The "SINAMICS alarms" screen shows the faults and warnings of the selected SINAMICS axis.

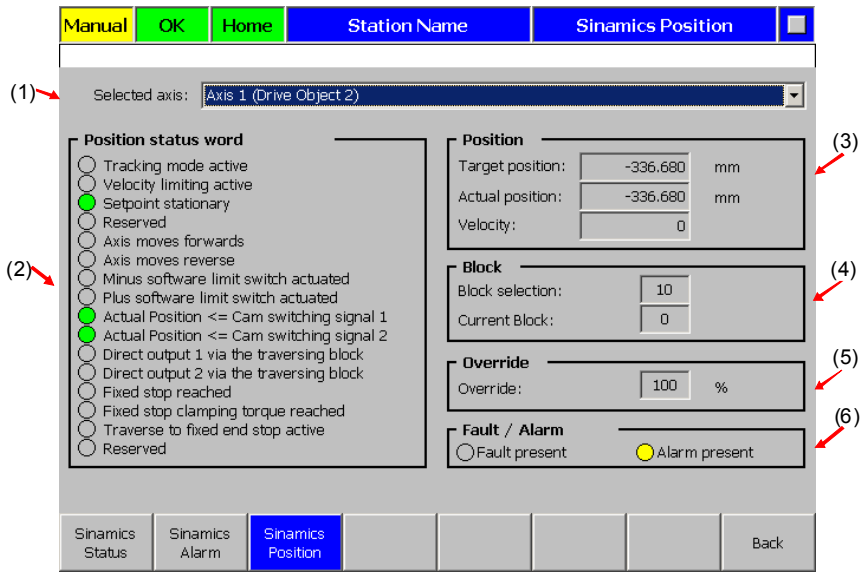


	Selection of the axis (selection window)
	Area for displaying the faults
	Area for displaying the warnings

Figure 8-24 "SINAMICS alarms" screen

8.5.3 "SINAMICS position" screen

The "SINAMICS position" screen shows the positioning status signals and positioning data, such as position, block and override of the selected axis. The data, however, is available only for those SINAMICS axes operated as positioning axes (EPOS).



1. Selection of the axis (selection window)
2. Positioning status signals
3. Display of the axis position
4. Number of the selected block
5. Override
6. Display of any pending fault/warning

Figure 8-25 "SINAMICS position" screen

Note

The "SINAMICS position" screen does not contain any information when a SINAMICS S120 without positioning functionality is used.

8.5.4 Configuring of the WinCC flexible screens

Configuring of the text list in WinCC flexible

The designations of the axes must be configured. The text items are stored in the "SO_80_SinamicsAxis" WinCC flexible text list. A position in the text list must be present for each configured axis. The text list that must be created for each drive group is displayed or hidden depending on the selected CU.

The "SO_80_SinamicsAxis" text list has the following structure:

Text list		**SO_80_SinamicsAxis**
Display		Text
Format		Decimal
Value	2	Designation of the first axis (value = drive object number)
Value	3	Designation of the second axis (value = drive object number)
…	…	…

Table 8-9 Text list for the axis designations

! Important

The designation text for missing axes must be deleted!

! Important

The value of the text list entry must match the drive object number of the drive group.

8.5.5 Runtime interface (FB_HMILITE_SINAMICSCU3x0)

The FB_HMILITE_SINAMICSCU3x0 supplies the WinCC flexible screens for the SINAMICS diagnostic screens. The displayed data is fetched directly from the drive as parameter requests using acyclical communications services.

The function block must be fetched once cyclically with the appropriate diagnostic address ("DRIVE_DIAGADDR" parameter) for each CU. The "DRIVE_ENABLE" parameter assigned to the selected CU (drive group) must be used to enable the FB call. In addition, the text list with the associated drive object numbers assigned to the drive group must be displayed.

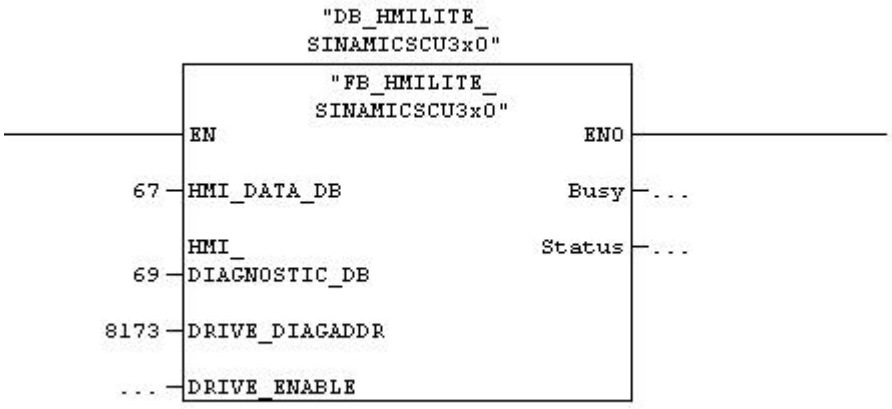


Figure 8-26 Call interface of the FB_HMILITE_SINAMICSCU3x0 function

Name	Type	Default	Example	Description
HMI_DATA_DB	INT	67	67	Number of the HMI runtime data block
HMI_ DIAGNOSTIC_DB	INT	69	69	Number of the HMI diagnostic data block
DRIVE_DIAGADDR	INT	---	---	SINAMICS diagnostic address (see HW Config)
DRIVE_ENABLE	BOOL	TRUE	---	"TRUE" enables the communication of the block with the drive
BUSY	BOOL	TRUE	---	"TRUE": communication with the drive is running
STATUS	WORD	---	---	Block status (0x8001 = SFB53 error, 0x8002 = SFB52 error)

Table 8-10 Parameters of the FB_HMILITE_SINAMICSCU3x0 function

8.6 "PROFIBUS" screen

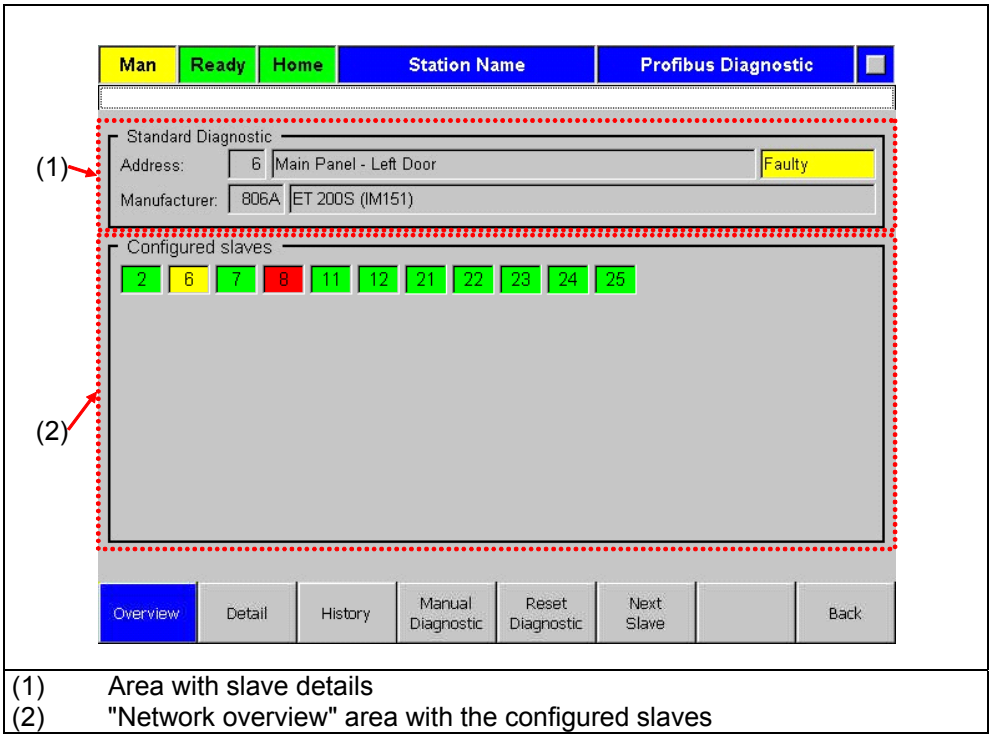
Note

The PROFIBUS Diagnostic screens are only active in the 6" variant. In the 10" variant, the screens are still to be found in the WinCC flexible configuration, but are no longer contained in the screen call-up structure because they are being replaced with the new PNIODiagnose screens.

If there are reasons why PNIODiagnose cannot be used with FB 465 (for description see Chapter 8.1), the previous version of PROFIBUS Diagnosis with FB 96 can continue to be used.

This adjustment can be made by adapting the screen calls.

8.6.1 Layout of the "PROFIBUS overview" screen



(1)	Area with slave details
(2)	"Network overview" area with the configured slaves

Figure 8-27 "PROFIBUS overview" screen

The "PROFIBUS diagnostics" screens are used to diagnose a PROFIBUS DP network.

The "PROFIBUS overview" screen provides an overview of the configured DP slaves are on the bus and the status they have.

The screen also displays detailed information of the selected DP slave.

The PROFIBUS DP diagnosis is based on the standard diagnostic FB for SIMATIC S7 available in the Siemens Internet. This diagnostic block monitors the PROFIBUS DP network and fetches the diagnostic data of the DP slave.

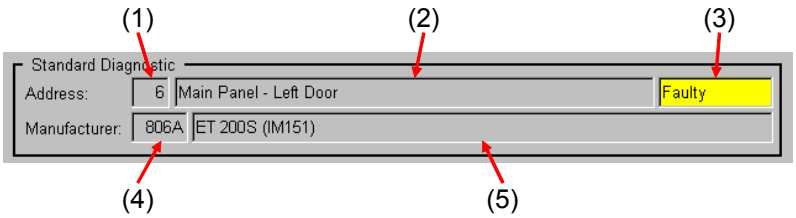
The overview displays the first faulty slave. This information appears in a higher-level window and can be used for troubleshooting where it is very useful.

The display must be confirmed by the operator, otherwise the display field will not be closed. An automatic close is performed once all slaves have returned to a fault-free state.

Figure 8-28 Display of slave address of first faulty slave

8.6.2 Overview: Diagnostic data of the slave

The "Slave details" area (standard diagnosis) displays information about the selected slave.



(1)	Slave address
(2)	Description of the slave location designation
(3)	Slave status
(4)	Manufacturer identification of the slave as hexadecimal number
(5)	Manufacturer identification of the slave as text

Figure 8-29 PROFIBUS diagnostics – details about the slave

Apart from the slave address (bus node) and the slave status, the manufacturer identification and the slave location are displayed as text.

Manufacturer identification

The manufacturer identification contains a code that describes the type of the DP slave (the product). Both the identification code as 4-digit hexadecimal number and the product name are displayed.

Slave location

The location designation of the slave. This information must be entered by the machine manufacturer.

Selection of the slave

The selection of the slave depends on the operating mode of the PROFIBUS diagnosis. Pressing the "Manual diagnosis" key switches between the two operating modes, auto and manual.

Automatic operation

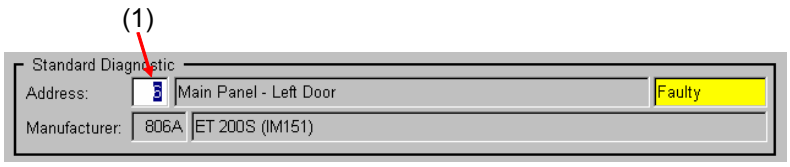
The "Next slave" key can be used in automatic operation of the PROFIBUS diagnosis to step to the next incorrect or faulty slave.

If there are no incorrect or faulty slaves, the node address "0" will be displayed (manufacturer identification "0000", slave status "OK").

Manual operation

In the manual operation of the PROFIBUS diagnosis, it is possible to enter from the keyboard any slave address that has been configured to view the detailed information.

Once "manual operation" has been selected, enter the slave address in the slave address field that has been activated as input screen.



(1) Input field for the slave address

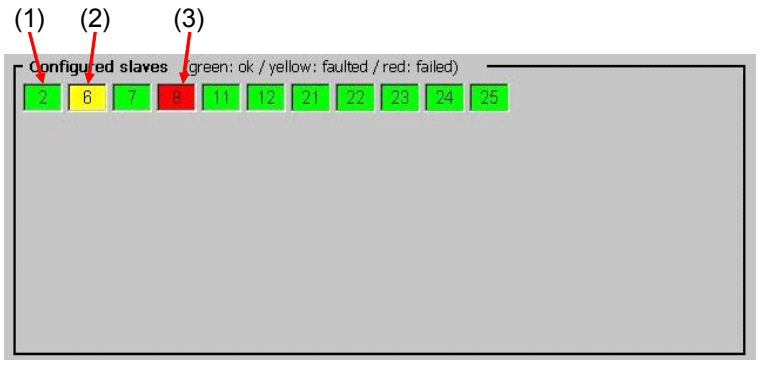
Figure 8-30 PROFIBUS diagnostics – manual selection of the slave

8.6.3 Overview: Diagnosis network overview

The "Network overview" area of the PROFIBUS overview displays the status of all DP slaves configured in the DP master system.

A maximum of 126 slaves can be displayed. Each bus node is represented by its address in a status field.

The slave status is indicated using various background colors.

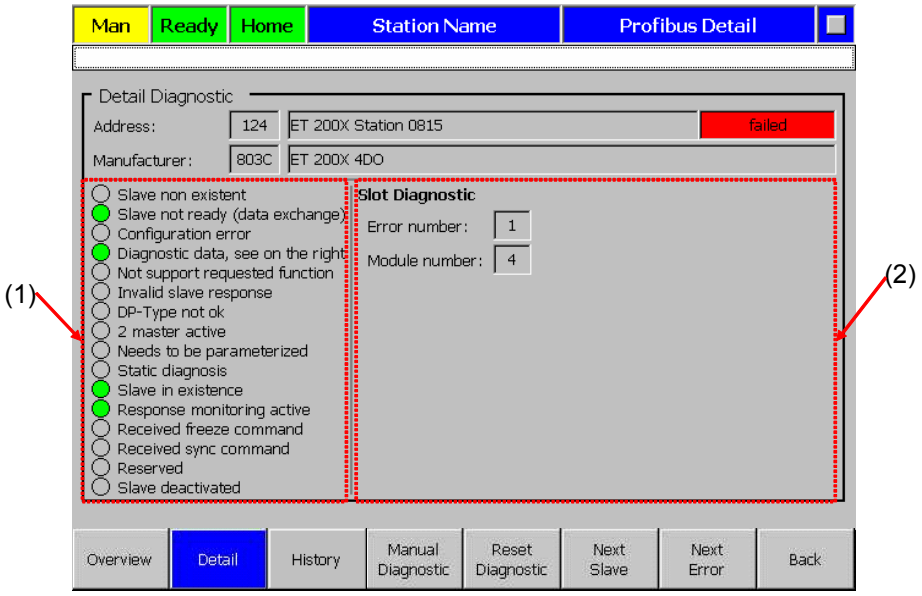


(4)	Slave status: OK	(green background)
(5)	Slave status: Incorrect	(yellow background)
(6)	Slave status: Faulty	(red background)

Figure 8-31 PROFIBUS diagnostics – network overview

8.6.4 Detailed diagnostics

The "PROFIBUS detailed diagnostics" screen shows detailed diagnostic information of the selected slave subdivided into two main areas.



(1)	Station status
(2)	Enhanced slave diagnostics

Figure 8-32 "PROFIBUS detailed diagnostics" screen

Station status

This area shows the station status of the PROFIBUS slave diagnostics.

The station status provides an overview of the status of the selected DP slave and is prescribed in the PROFIBUS standard for each standard PROFIBUS slave.

Enhanced diagnostics

The information displayed in the "Enhanced diagnostics" screen area depends on the diagnostic capabilites of the selected slave. Not only information in accordance with the PROFIBUS standard, but also special SIMATIC S7 diagnostic data is displayed. Figure 8-5 shows the line diagnostics of a diagnostic repeater.

Diagnostic type

A PROFIBUS DP slave can use the supplied diagnostic data to report several errors. The information shown for "diagnostic type" depends on the type of the fault or malfunction. The possible combinations are listed in the following table.

Diagnostic type	Number of the module	Module status	Channel number	Channel type	Channel faulty
Slot diagnostics	yes				
Module status	yes	yes			
Channel diagnostics	yes		yes	yes	yes
S7 diagnostics	yes		yes	yes	yes
Device diagnostics	yes				yes

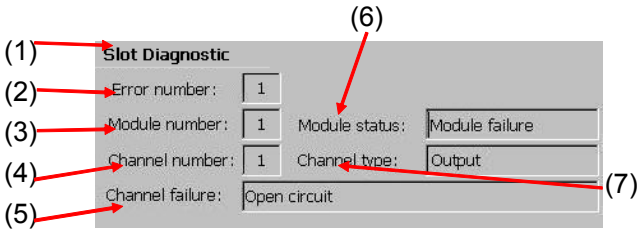
Table 8-11 PROFIBUS diagnostics – types of the enhanced diagnosis

Slot diagnostics, module status and channel diagnostics are diagnostic types in accordance with the PROFIBUS standard.

S7 diagnostics and device diagnostics are two enhanced diagnostic types provided by the SIMATIC S7 DP slaves and S7 modules.

Display of errors

The "Next error" key can be used to scroll through all reported errors of the selected slave. Depending on the diagnostic type of the error, different information fields will be displayed (for example, "module status" or "channel number").



(1) Diagnostic type
(2) Error number
(3) Number of the module
(4) Channel number
(5) Error text
(6) Module status
(7) Channel type

Figure 8-33 PROFIBUS diagnostics – error display

The displayed variable text items are stored in the following WinCC flexible text lists:

Text list	**SS_43_PBModuleStatus**
Display	Text
Format	Decimal

Table 8-12 PROFIBUS diagnostics – text list for the module status

Text list	**SS_43_PBChannelType**
Display	Text
Format	Decimal

Table 8-13 PROFIBUS diagnostics – text list for the channel type

Text list	**SS_43_PBChannelError**
Display	Text
Format	Decimal

Table 8-14 PROFIBUS diagnostics – text list for the channel errors

Because these text lists already contain all required text items, a configuring is not necessary.

8.6.5 DPD history

The "PROFIBUS DP history“ screen provides a tabular overview of the detailed diagnostic slave information stored. The latest message is displayed on top. The information is identical to the detailed information displayed and stored in chronological order only.

The history stores the last eight messages which are also displayed on MP277. On OP177B, the last five messages are displayed in a table.

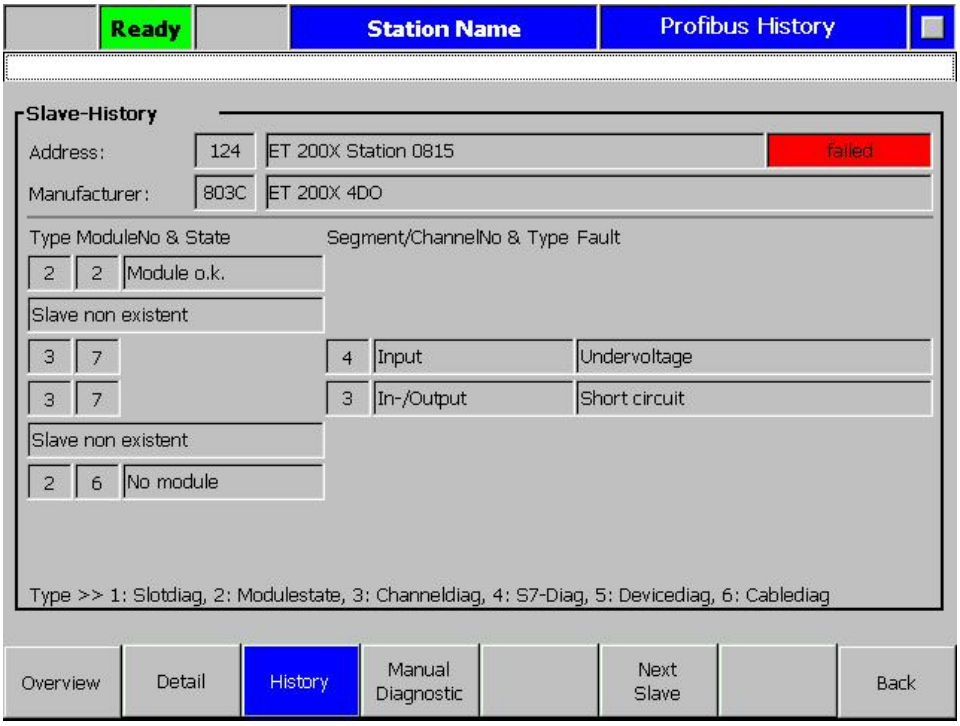


Figure 8-34 “PROFIBUS history" screen

Type

The first column comprises the type of error occurred. The following types are differentiated for diagnostics:

1: Slot diagnostics
2: Module status
3: Channel diagnostics
4: S7 diagnostics
5: Unit diagnostics
6: Cable diagnostics

If a slave completely fails, i. e. it is no longer a bus node, the following message is output: “Slave not on bus“.

Enhanced diagnostic information and diagnostic types

All further data displayed in the table correspond to the information included in the "Enhanced diagnostics" area from the detail display which has been evaluated in the current error case. These data is stored in chronological order for the relevant slave, allowing error tracking for each slave. This supplies conclusive information about the cause and frequency of the errors occurring facilitates filtering of frequently recurring error profiles and allows for structured troubleshooting.

Navigation

Navigation within the “History“ screen is analog to the diagnostic displays. The “Next slave“ selection option allows to leaf through slaves. If “manual diagnostics“ is active, all slaves are leafed through. If “manual diagnostics“ is not activated, only the currently faulty slaves are leafed through. When selecting the manual function, you can directly enter a slave address in the address field.

8.6.6 Configuring the WinCC flexible screens

Location designation of the slave

The text items for the "location designation of the slave" must be specified. The text items are stored in the WinCC flexible "SO_40_PBLocation" text list. Each configured slave address must have an entry in the text list. The location designations for non-configured slaves can be deleted. "SO_40_PBLocation" has the following structure:

Text list		**SO_40_PBLocation**
Display Format		Text Decimal
Value	0	Reserved [empty]
Value	1	Location designation of the slave with the node address 1
…	…	…
Value	126	Location designation of the slave with the node address 126

Table 8-15 PROFIBUS diagnostics – text list for the location designation of the slaves

Manufacturer identification

The manufacturer identifications are stored in the WinCC flexible following text lists:

Text list		**SS_40_PBSlaveIdent_0000_7FFF**
Display Format		Text Decimal
Value	0	[empty]
Value	1	Product designation for the manufacturer identification 0x0001
…	…	…
Value	32767	Product designation for the manufacturer identification 0x7FFF

Table 8-16 Text list for the product designations of the manufacturer identifications (0x0000...0x7FFF)

Text list		**SS_40_PBSlaveIdent_8000_FFFF**
Display Format		Text Decimal
Value	0	Product designation for the manufacturer identification 0x8000
Value	1	Product designation for the manufacturer identification 0x8001
…	…	…
Value	32767	Product designation for the manufacturer identification 0xFFFF

Table 8-17 Text list for the product designations of the manufacturer identifications (0x8000…0xFFFF)

The Siemens PROFIBUS components are already predefined in the text lists. The machine manufacturer can add further product designations.

Use of the simple PROFIBUS diagnostics

The simple PROFIBUS DP diagnosis is based on the standard diagnostic FC for SIMATIC S7 available in the Siemens internet. Although this diagnostic block monitors the PROFIBUS DP network, it does not evaluate any detailed information of the slaves.

When the function is used, the display is restricted to the "PROFIBUS overview" screen. Because no detailed information is available, the screens for the detailed diagnostics can be deleted and the buttons for the call must be deleted. The manual operation is active for this functionality. The slaves can only be selected using the address field activated as input field; this means that the button for switching must be removed.

The following elements on the 10" device must be deleted:

- Screen 43, SS_43_ProfibusDiagnosticDetail
- In screen 42, SS_42_ProfibusOverview, the "Details" and "Manual diagnostics" buttons, and when a keyboard device is used, the functions for keys F14 and F16

The following elements on the 6" device must be deleted:

- Screen 43, SS_43_ProfibusStandardDiagnosticsDetail, and 44, SS_44_ProfibusExtendedDiagnosticsDetail.
- In screen 42, SS_42_ProfibusOverview, the "Details" and "Manual diagnostics" buttons, and when a keyboard device is used, the functions for keys F9 and F10

8.6.7 Runtime interface (FC_HMILITE_PROFIBUS)

The FC_HMILITE_PROFIBUS supplies data to the WinCC flexible screens of the PROFIBUS diagnostics. To analyze the bus system, the function uses the FB_SIEM_DP_DIAG_DETAIL (FB96) standard function block for SIMATIC S7. The FB96 (originally FB125) can be downloaded from the Siemens Internet for the product support pages. A detailed description of the module is also available there. This document does not provide a more detailed description because only the DP_MASTERSYSTEM (number of the master system, see HW Config) and EXTERNAL_DP_INTERFACE (= FALSE) input parameters may be parameterized.

The user must call this function block in the OB1, OB82 and OB86 (see sample program).

! Important

Only the DP_MASTERSYSTEM and EXTERNAL_DP_INTERFACE input parameters may be parameterized with the appropriate values. The other input parameters will be controlled from the HMI Lite system!

Further details for the program structure, the block designations and the block numbers are contained in the software guide of the "Solutions for Powertrain TRANSLINE 2000" manual.

The FC_SIEM_DP_DIAG_OVERVIEW function can be used as alternative to the FB_SIEM_DP_DIAG_DETAIL. In this case however, no detailed diagnosis of the PROFIBUS system is possible.
If the "FCs „FC_SIEM_DP_DIAG_OVERVIEW" is used, the WinCC flexible configuration must be modified. For further details, see:

Chapter 8.6.5 "Configuration of the WinCC flexible Screens“.

If the FC_SIEM_DP_DIAG_OVERVIEW is used instead of the FB_SIEM_DP_DIAG_DETAIL, approximately 6.7 KB main memory in the controller is saved.

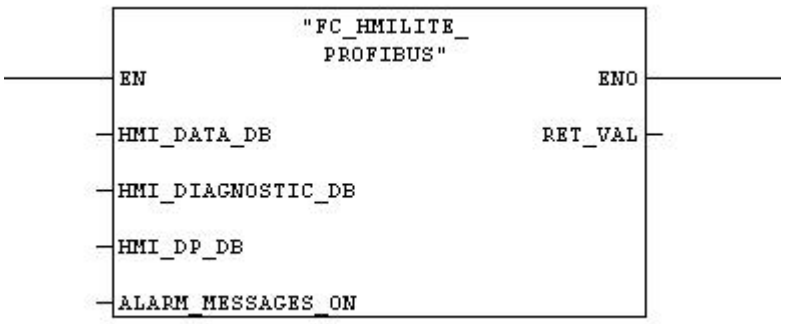


Figure 8-35 Call interface of the FC_HMILITE_PROFIBUS (FC105) function

Name	Type	Default	Example	Description
HMI_DATA_DB	INT	67	67	The number of the HMI runtime data block
HMI_ DIAGNOSTIC_DB	INT	69	69	The number of the HMI diagnostic data block
HMI_DP_DB	INT	96	96	The number of the instance data block for FB_SIEM_DP_DIAG_DETAIL. If the "FC_SIEM_DP_DIAG_OVERVIEW" FC is used, a 0 must be parameterized here.
ALARM_MESSAGES_ ON	BOOL	---	---	When TRUE, the FC generates a message when a DP slave has failed or is faulty.
RET_VAL	WORD	---	---	Function return value 0000: no error 7000: Block not processed

Table 8-18 Parameters of the FC_HMILITE_PROFIBUS (FC105) function

8.6.8 Runtime interface (FC_SIEM_DP_DIAG_OVERVIEW (FC96))

The FC96 block determines the failed and faulty slaves cyclically or event-controlled. It should be used when only the overview diagnosis of a bus system is to be displayed.

If the FC96 is processed successfully, the BIE bit will be set to "1". The BIE bit is deleted when the processing of the FC96 failed.

The DP evaluation can be activated in the OB1 cyclically or event-controlled. Bits can be set in the OB82 (diagnosis), OB86 (station failure) and OB100 (restart) that are linked in the OB1 to an activation condition (CHECK_AKTIV).

Technical specifications of the FC96:
Execution time for CHECK_ACTIV=0: approx. 0 ms
Execution time for CHECK_ACTIV=1: < 2 ms (depending on the DP master)
User memory space in CPU: 1.3 KB

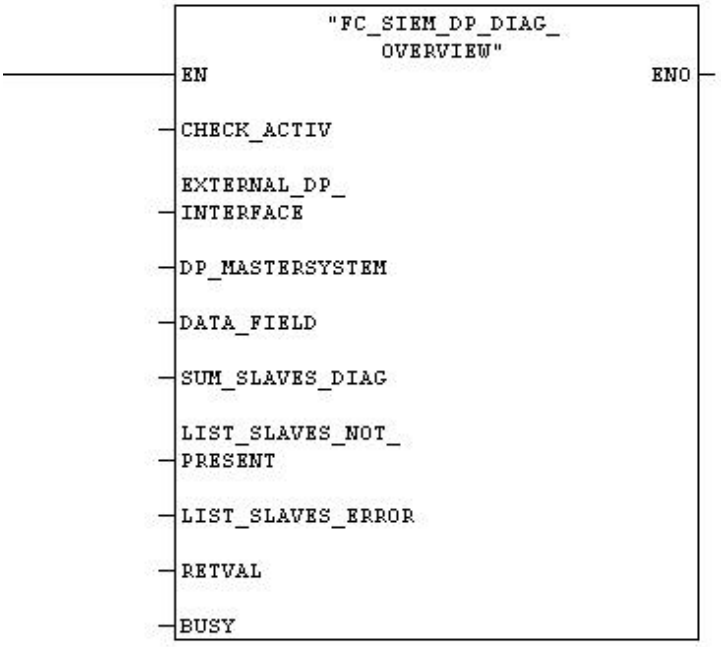


Figure 8-36 Call interface of the FC_SIEM_DP_DIAG_OVERVIEW (FC96) function

Name	Type	Default	Example	Description
CHECK_ACTIV	BOOL	M0.1	M0.1	This is used by the user to activate the evaluation. The activation bit can be set cyclically or in some other block (e.g. B82, OB86, OB100 and thus event-controlled).
EXTERNAL_DP_ INTERFACE	BOOL	FALSE	FALSE	Information whether the DP interface of the master is an integrated (Master-CPU = 0) or an external DP interface (CP/IM =1).
DP_MASTERSY STEM	INT	1	1	The number of the DP master system configured with STEP 7 must be specified here.

Name	Type	Default	Example	Description
DATA_FIELD	ANYL	P#DB69. DBX302.0 BYTE 50	P#DB69. DBX302.0 BYTE 50	Data field for internal block processing. Only Byte is permitted as data type. The field size must be at least 50 bytes. The data field is not relevant for the user.
SUM_SLAVES_ DIAG	INT	---	---	The number of the affected DP slaves (failed or faulty).
LIST_SLAVES_ NOT_PRESENT	ANY	P#DB69. DBX286.0 DWORD 4	P#DB69. DBX286.0 DWORD 4	List of the failed DP slaves. Each DP bus node is assigned to a bit in the list. The the following lengths must be specifies as length values: 128 bits or 16 bytes or 8 words or 4 Dwords.
LIST_SLAVES_ ERROR	ANY	P#DB69. DBX270.0 DWORD 4	P#DB69. DBX270.0 DWORD 4	List of the faulty DP slaves. Each DP bus node is assigned to a bit in the list. The following lengths must be specifies as length values: 128 bits or 16 bytes or 8 words or 4 Dwords.
RETVAL	INT	DB69.DB W354	DB69.DB W354	Return value with the following error codes: 0000 = no error during parameterization of the block 8080 = incorrect data type at least one parameter of the ANY type 8081 = incorrect area length at least one parameter of the ANY type 8082 = no evaluation possible because of missing CPU information.
BUSY	BOOL	DB69.DB X356.0	DB69.DB X356.0	This parameter specifies that the FC is currently evaluating the DP system.

Table 8-19 Parameters of the FC_SIEM_DP_DIAG_OVERVIEW (FC96) function

8.6.9 Runtime interface (FC_HMILITE_DP_HISTORY)

The FC_HMILITE_DP_HISTORY supplies data to the WinCC flexible history screen for Profibus diagnostics.

The user must call up this function block in FC67 and OB86 (see sample program). When being called in OB86, the input parameter “OB86_CALL“ must be assigned to TRUE. If FC172 is called in FC67, the parameter must be assigned to FALSE.

Data are stored in the corresponding DB172 which has to be integrated in the program. This data block is responsible for completing the table for the relevant slave and storing failure and fault consequences.

The data block comprises a memory location structure for the slave history. By default, this is limited to a maximum number of 20 slaves (for the entire system). If the hardware configuration comprises more slaves, you have to increase the number in the array field. Due to reduced memory space, you can only enter a limited number which is sufficient for most systems.

+98.0	SLAVE_HISTORY	ARRAY[1..20]	
*0.0		STRUCT	
+0.0	SLAVE_ADR	BYTE	B#16#0
+1.0	SLAVE_STATUS	BYTE	B#16#0
+2.0	RES_20	BYTE	B#16#0
+4.0	Error	ARRAY[1..8]	
*0.0		STRUCT	
+0.0	ERROR_TYP	BYTE	B#16#0
+1.0	MODULE_NO	BYTE	B#16#0
+2.0	MODULE_STATE	BYTE	B#16#0
+3.0	CHANNEL_NO	BYTE	B#16#0
+4.0	CHANNEL_TYP	BYTE	B#16#0
+5.0	SLAVE_OUT_OF_ORDER	BOOL	FALSE
+6.0	CHANNEL_ERROR_CODE	INT	0
=8.0		END_STRUCT	
=68.0		END_STRUCT	
=1458.0		END_STRUCT	

Figure 8-37 DB_HMILITE_DP_HISTORY (DB172) slave number adaptation

Important

The FC_HMILITE_DB_HISTORY (FC172) only works in combination with FB_SIEM_DP_DIAG_DETAIL (FB96). It does not work with the FC version of the Siemens PROFIBUS standard diagnostic FC_SIEM_DP_DIAG_OVERVIEW (FC96)! The data included in DB96 are required to generate the history table.

The WinCC flexible configuration need not be adapted since this has already been done for detailed diagnostics.

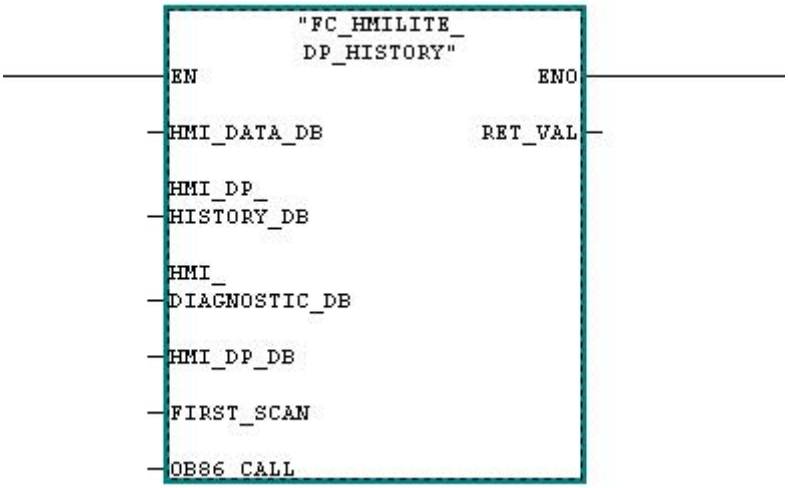


Figure 8-38 Calling interface of the function FC_HMILITE_DP_HISTORY (FC172)

Name	Type	Standard	Exam-ple	Description
HMI_DATA_DB	INT	67	67	Number of the HMI-Runtime data block
HMI_ HISTORY_DB	INT	172	172	Number of the HMI history block
HMI_ DIAGNOSTIC_DB	INT	69	69	Number of the HMI diagnostic data block
HMI_DP_DB	INT	96	96	Number of the instance data block for FB_SIEM_DP_DIAG_DETAIL.
RESTART_FLAG	BOOL	---	---	Displays a controller restart. Must have the value “TRUE“ within the first cycle.
OB86_CALL	BOOL	---	---	Must be set to “TRUE“ if the call was initiated from OB86.
RET_VAL	WORD	---	---	Function return value 0000: No fault 8001: Function has not been executed since DB172 is too short to make entries

Table 8-20 Parameters for the function FC_HMILITE_DP_HISTORY (FC172)

■

9

9 System Screens

9.1 "System" screen

9.1.1 Layout of the screen and functionality

The "System" screen contains general system functions, such as password and user administration, the language setting for the user interface, lamp test and clear the alarm and message archive.

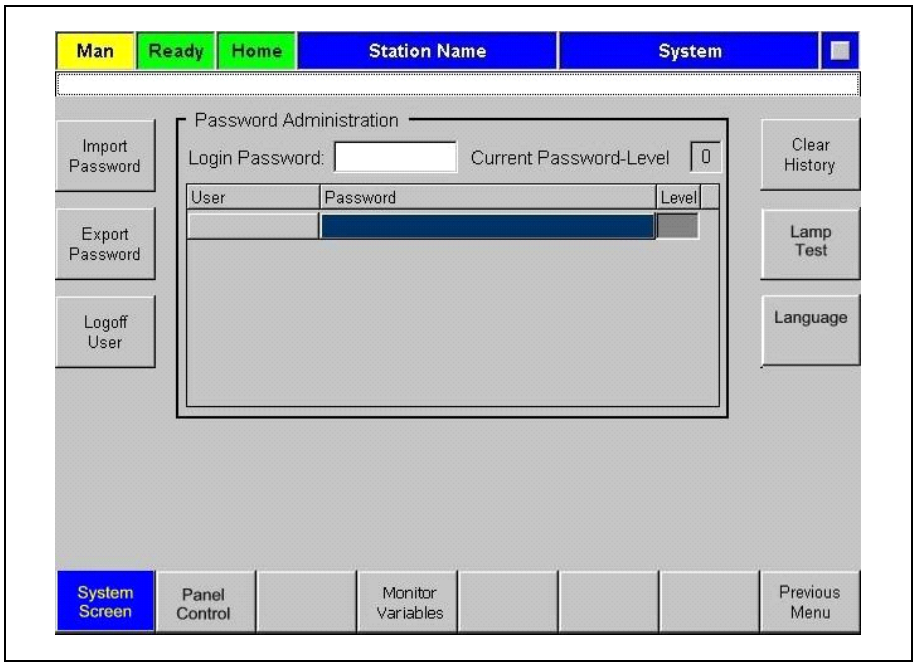


Figure 9-1 "System" screen

Import/export password

This function can be used to export the password list to a memory card or import the password list from a memory card. This makes it possible to specify the password list for only one machine and then transfer it to the other machines.

User logoff

This function is used to reset the current password level to level 0 (user without any special rights).

Delete message buffer

This function is used to clear the message buffer. This includes all messages that have occurred until this time.

Lamp test

The DB_HMILITE_DATA.GLOBAL.LAMPTEST variable has the 1-signal while the key or button remains pressed.

This variable must be further processed by the machine manufacturer.

Address:	DB_HMILITE_DATA.GLOBAL.LAMPTEST
Format:	BOOL
Range of values:	1-signal when the "Lamp test" key on the operator panel remains pressed.
Default setting:	False

Changing the language

The "Change language" button can be used to switch between the languages installed on the operator panel.

A maximum of three languages can be loaded onto the operator panel. The standard project for HMI Lite is delivered in five languages:

- German
- English (United Kingdom)
- French
- Italian
- Spanish (international)

Other languages can be implemented for specific projects.

9.2 "Panel Control" screen

9.2.1 Layout of the screen and functionality

The "SS_05_PanelControl" screen provides a number of functions associated with the maintenance and the setting of the operator panel.

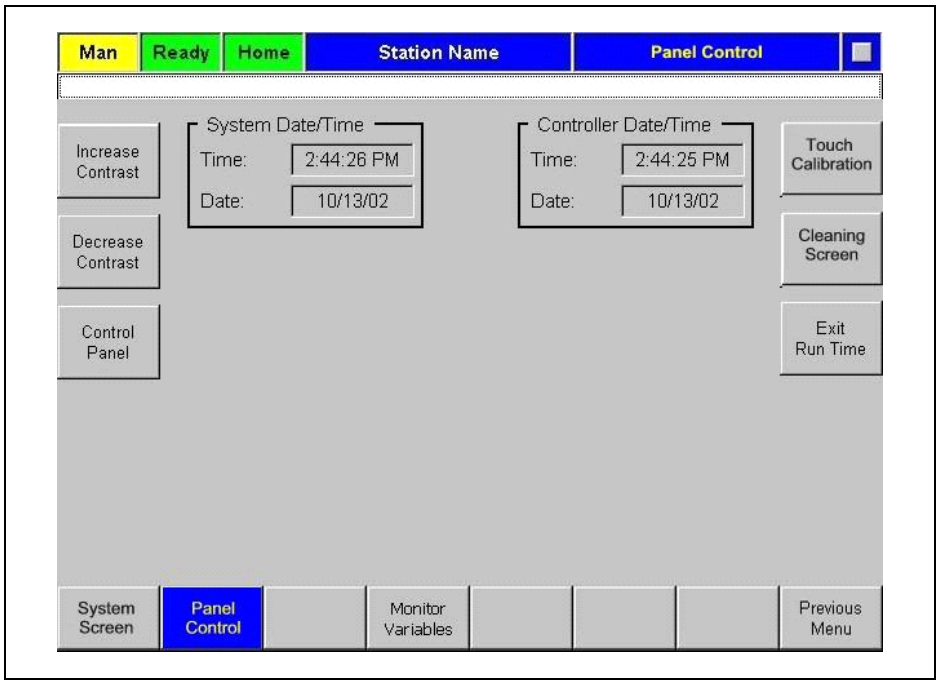


Figure 9-2 "Panel Control" screen

Increase/reduce contrast

These function keys are used to set the contrast of the operator panel.

System control

Pressing this function key opens the window for the system control of the operating system.

Clean screen (only for touchscreen operator panels)

After pressing the "Clean screen" function key, the operator panel user interface switches for a parameterizable time to an empty screen page on which the touch function is deactivated. During this time, it is possible to clean the screen without the danger of inadvertently initiating some function.

Touch calibration (only for touch screen operator panels)

Pressing the "Touch calibration" button starts the calibration of the touch screen.

Exit runtime

Pressing this button exits the WinCC flexible runtime environment and switches to the operating system level.

9.3 "Status Variable" screen

9.3.1 Layout of the screen and functionality

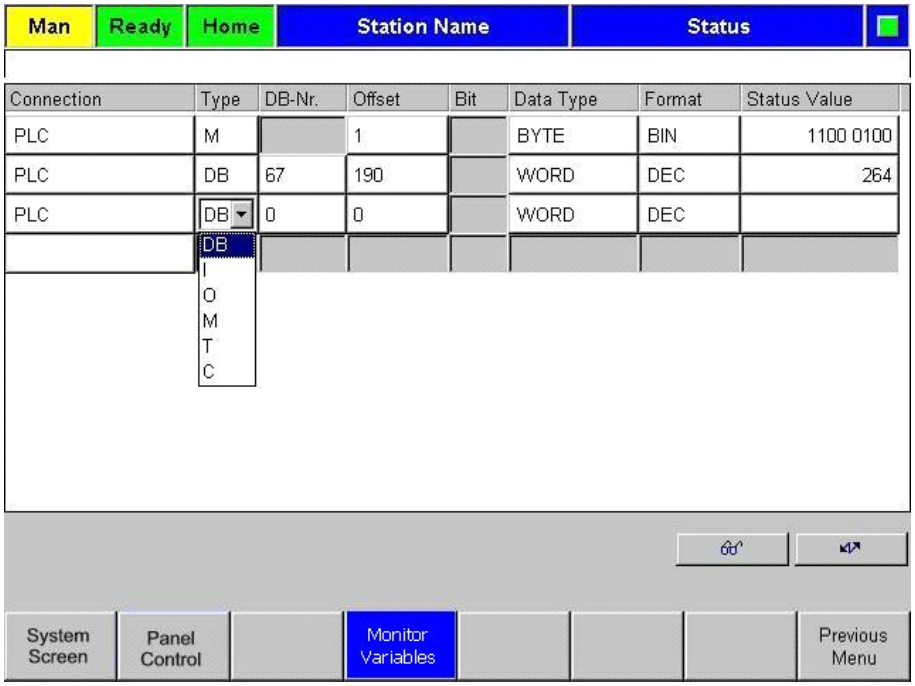


Figure 9-3 "Status Variable" screen

The user can use this screen to monitor the control variables for diagnostic or maintenance purposes.

■

Notes

A

A Appendix

A.1 Abbreviations

The list of abbreviations for the HMI Lite TRANSLINE can be found in Solutions for Powertrain TRANSLINE Manual – Appendix.

A.2 Bibliography

Description	Type
Working with STEP 7 First Steps	Manual
Programming with STEP 7 Manual	Manual
Configuration Hardware and Data Exchange Connections with STEP 7 Manual	Manual
Ladder Diagram (LAD) for Programming S7-300 and S7-400 Reference Manual	Manual
Statement List (STL) for Programming S7-300 and S7-400 Reference Manual	Manual
Function Block Diagram (FBD) for Programming S7-300 and S7-400 Reference Manual	Manual
Standard Software for S7 and M7 STEP 7 User's Guide	Manual
Standard Software for S7-300 and S7-400 Standard Functions, Part 2 Reference Manual	Manual
System Software for S7-300/400, System and Standard Functions Reference Manual	Manual
S7 Graph for S7-300/400 Programming of Sequence Controls Manual	Manual
SIMATIC NET Actuator/Sensor Interface Master Module CP 342-2	Manual
SIMATIC NET CP 343–2 Actuator/Sensor Interface Master Manual	Manual
PLC S7-300, CPU Descriptions; CPU 312 IFM to CPU 318-2 DP Reference Manual	Manual

Description	Type
S7-300 Command List – CPU Descriptions CPU 312C, 313C, 313C-2 PtP, 313C-2 DP, 314C-2 PtP, 314C-2 DP, 312, 314, 315-2 DP	Manual
Simatic HMI WinCC flexible 2008 Compact / Standard / Advanced User Manual	Manual
SIMATIC HMI WinCC flexible 2008 Communication Part 1 User Manual	Manual
SIMATIC HMI WinCC flexible 2008 Communication Part 2 User Manual	Manual
SIMATIC HMI MP 277 Operator Panel (WinCC flexible) Instruction Manual	Manual
SIMATIC HMI TP 177A, TP 177B, OP 177B Operator Panels (WinCC flexible) Instruction Manual	Manual
PROFIBUS DP FB125, FC125 Diagnostic Blocks for SIMATIC S7	Download
PROFIBUS diagnostic package for SIMATIC S7 and TD/OP and WinCC	Download
PROFINET IO diagnostic package	
SIMODRIVE 611U / POSMO SI/CD/CA V5.2 Software Package	Download

Table A-1 Bibliography

A.3 Structure of the DB_HMILITE_DATA data block

Address	Name	Type
Software version of the DB		
0	RELEASE.MAJOR	BYTE
1,0	RELEASE.MINOR	BYTE
Area pointer		
2,0	AREAPOINTER.COORDINATION[1..16]	BOOL
4,0	AREAPOINTER.USER_VERSION	WORD
6,0	AREAPOINTER.SCREEN_NUMBER.SCREENTYPE	INT
8,0	AREAPOINTER.SCREEN_NUMBER.SCREENNUMBER	INT
10,0	AREAPOINTER.SCREEN_NUMBER.SPARE_1	INT
12,0	AREAPOINTER.SCREEN_NUMBER.FIELDNUMBER	INT
14,0	AREAPOINTER.SCREEN_NUMBER.SPARE_2	INT
16,0	AREAPOINTER.JOB_MAILBOX.JOBNUMBER	INT
18,0	AREAPOINTER.JOB_MAILBOX.PARAMETER_1	INT
20,0	AREAPOINTER.JOB_MAILBOX.PARAMETER_2	INT
22,0	AREAPOINTER.JOB_MAILBOX.PARAMETER_3	INT
24,0	AREAPOINTER.DATE_TIME_OP[1..12]	BYTE
48,0	AREAPOINTER.SPARE[1..22]	BYTE
Job		
70,0	GLOBAL.JOB.NUMBER	INT
72,0	GLOBAL.JOB.PARAMETER_1	INT
74,0	GLOBAL.JOB.PARAMETER_2	INT

Address	Name	Type
76,0	GLOBAL.JOB.PARAMETER_3	INT
Operator Information		
78,0	GLOBAL.PROMPT.INDEX	WORD
80,0	GLOBAL.PROMPT.ATTRIBUT	WORD
Global internal variables		
82,0	GLOBAL.INTERNAL.BUTTON.PRESSED_HORZ	WORD
84,0	GLOBAL.INTERNAL.BUTTON.RESERVED_01	WORD
86,0	GLOBAL.INTERNAL.BUTTON.PRESSED_VERT	WORD
88,0	GLOBAL.INTERNAL.BUTTON.SELECTED_VERT	WORD
90,0	GLOBAL.INTERNAL.BUTTON.RESERVED_02	WORD
92,0	GLOBAL.INTERNAL.BUTTON.RESERVED_03	WORD
94,0	GLOBAL.INTERNAL.BUTTON.RESERVED_04	WORD
96,0	GLOBAL.INTERNAL.BUTTON.BUTTON_RELEASED_VERT	WORD
98,0	GLOBAL.INTERNAL.BUTTON.RESERVED_05	WORD
100,0	GLOBAL.INTERNAL.BUTTON.RESERVED_06	WORD
102,0	GLOBAL.INTERNAL.BUTTON.RESERVED_07	WORD
104,0	GLOBAL.INTERNAL.BUTTON.KEY_PRESSED_LAST_VERT	WORD
106,0	GLOBAL.INTERNAL.BUTTON.PRESSED_PULSE_HORZ	WORD
108,0	GLOBAL.INTERNAL.BUTTON.PRESSED_PULSE_VERT	WORD
110,0	GLOBAL.INTERNAL.BUTTON.PRESSED_PREVIOUS_HORZ	WORD
112,0	GLOBAL.INTERNAL.BUTTON.PRESSED_PREVIOUS_VERT	WORD
114,0	GLOBAL.INTERNAL.BUTTON.TIMER_SELECTED_VALUE	TIME
118,0	GLOBAL.INTERNAL.BUTTON.TIMER_SELECTED_STATUS	BOOL
118,1	GLOBAL.INTERNAL.BUTTON.PRESSED_LEFT	BOOL
118,2	GLOBAL.INTERNAL.BUTTON.PRESSED_RIGHT	BOOL
118,3	GLOBAL.INTERNAL.BUTTON.PRESSED_BOTTOM	BOOL
118,4	GLOBAL.INTERNAL.BUTTON.LOCKED	BOOL
118,5	GLOBAL.INTERNAL.BUTTON.RESET_SELECTED	BOOL
118,6	GLOBAL.INTERNAL.BUTTON.SELECTED	BOOL
118,7	GLOBAL.INTERNAL.BUTTON.DOUBLE_CLICK	BOOL
119,0	GLOBAL.INTERNAL.BUTTON.PREVIOUS_KEY_MODE	BYTE
Global variables		
120,0	GLOBAL.SCREEN_ID	WORD
122,0	GLOBAL.SCREEN_CHANGED	BOOL
122,1	GLOBAL.MANUAL_SCREEN_ACTIVE	BOOL
122,2	GLOBAL.ENABLE_TOUCH	BOOL
124,0	GLOBAL.TIMER_VALUE	TIME
128,0	GLOBAL.SCREEN_PREVIOUS	WORD
130,0	GLOBAL.CLOCK.MEMORY	BYTE
131,0	GLOBAL.CLOCK.PREVIOUS	BYTE
132,0	GLOBAL.CLOCK.PULSE	BYTE
134,0	GLOBAL.OP_ALIVE.TIMER_VALUE	TIME
138,0	GLOBAL.OP_ALIVE.MANUAL_SCREEN_SELECTED	BOOL
138,1	GLOBAL.OP_ALIVE.MANUAL_SCREEN_ACTIVE	BOOL
138,2	GLOBAL.OP_ALIVE.SIGN_OF_LIFE_FP	BOOL
138,3	GLOBAL.OP_ALIVE.SIGN_OF_LIFE_FN	BOOL
138,4	GLOBAL.OP_ALIVE.DELAY_MAN_SCREEN_INACTIV	BOOL

Address	Name	Type
140,0	GLOBAL.BUTTON_PRESSED_VERT	WORD
142,0	GLOBAL.KEY_PRESSED_VERT	WORD
144,0	GLOBAL.LAMPTEST	BOOL
Version IDs		
146,0	GLOBAL.PROGRAM_VERSION.FC_BASIC.HI	BYTE
147,0	GLOBAL.PROGRAM_VERSION.FC_BASIC.LO	BYTE
148,0	GLOBAL.PROGRAM_VERSION.FC_MANUAL.HI	BYTE
149,0	GLOBAL.PROGRAM_VERSION.FC_MANUAL.LO	BYTE
150,0	GLOBAL.PROGRAM_VERSION.FC_PROFIBUS.HI	BYTE
151,0	GLOBAL.PROGRAM_VERSION.FC_PROFIBUS.LO	BYTE
152,0	GLOBAL.PROGRAM_VERSION.FC_RF300.HI	BYTE
153,0	GLOBAL.PROGRAM_VERSION.FC_RF300.LO	BYTE
154,0	GLOBAL.PROGRAM_VERSION.FC_ASI.HI	BYTE
155,0	GLOBAL.PROGRAM_VERSION.FC_ASI.LO	BYTE
156,0	GLOBAL.PROGRAM_VERSION.FC_COUNTER.HI	BYTE
157,0	GLOBAL.PROGRAM_VERSION.FC_COUNTER.LO	BYTE
158,0	GLOBAL.PROGRAM_VERSION.FC_CYCLETIME.HI	BYTE
159,0	GLOBAL.PROGRAM_VERSION.FC_CYCLETIME.LO	BYTE
160,0	GLOBAL.PROGRAM_VERSION.FC_DRIVES.HI	BYTE
161,0	GLOBAL.PROGRAM_VERSION.FC_DRIVES.LO	BYTE
162,0	GLOBAL.PROGRAM_VERSION.FB_S7GRAPH.HI	BYTE
163,0	GLOBAL.PROGRAM_VERSION.FB_S7GRAPH.LO	BYTE
164,0	GLOBAL.PROGRAM_VERSION.DB_DIAGNOSTIC.HI	BYTE
165,0	GLOBAL.PROGRAM_VERSION.DB_DIAGNOSTIC.LO	BYTE
166,0	GLOBAL.PROGRAM_VERSION.FC_HISTORY.HI	BYTE
167,0	GLOBAL.PROGRAM_VERSION.FC_HISTORY.LO	BYTE
168.0	GLOBAL.PROGRAM_VERSION.DB_HISTORY.HI	BYTE
169.0	GLOBAL.PROGRAM_VERSION.DB_HISTORY.LO	BYTE
170.0	GLOBAL.PROGRAM_VERSION.FB_SINAMICS.HI	BYTE
171.0	GLOBAL.PROGRAM_VERSION.FB_SINAMICS.LO	BYTE
172,0	GLOBAL.RESERVE[100..123]	BYTE
Alarms		
196,0	GLOBAL.ALARM.ALARM_09	BOOL
196,1	GLOBAL.ALARM.ALARM_10	BOOL
196,2	GLOBAL.ALARM.ALARM_11	BOOL
196,3	GLOBAL.ALARM.ALARM_12	BOOL
196,4	GLOBAL.ALARM.ALARM_13	BOOL
196,5	GLOBAL.ALARM.ALARM_14	BOOL
196,6	GLOBAL.ALARM.ALARM_15	BOOL
196,7	GLOBAL.ALARM.ALARM_16	BOOL
197,0	GLOBAL.ALARM.FAILED_DP_SLAVE	BOOL
197,1	GLOBAL.ALARM.FAULTY_DP_SLAVE	BOOL
197,2	GLOBAL.ALARM.FAILED_ASI_SLAVE	BOOL
197,3	GLOBAL.ALARM.FAULTY_RF300	BOOL
197,4	GLOBAL.ALARM.ALARM_05	BOOL
197,5	GLOBAL.ALARM.ALARM_06	BOOL
197,6	GLOBAL.ALARM.ALARM_07	BOOL

Address	Name	Type
197,7	GLOBAL.ALARM.ALARM_08	BOOL
198,0	GLOBAL.ALARM.MANUAL_SCREEN_INACTIVE	BOOL
198,1	GLOBAL.ALARM.ASI_DISCONNECTED	BOOL
198,2	GLOBAL.ALARM.DRIVE_DISCONNECTED	BOOL
198,3	GLOBAL.ALARM.SINAMICS_BLOCK_DISABLE	BOOL
198,4	GLOBAL.ALARM.ALARM_29	BOOL
198,5	GLOBAL.ALARM.ALARM_30	BOOL
198,6	GLOBAL.ALARM.ALARM_31	BOOL
198,7	GLOBAL.ALARM.ALARM_32	BOOL
199,0	GLOBAL.ALARM.ALARM_17	BOOL
199,1	GLOBAL.ALARM.ALARM_18	BOOL
199,2	GLOBAL.ALARM.ALARM_19	BOOL
199,3	GLOBAL.ALARM.ALARM_20	BOOL
199,4	GLOBAL.ALARM.ALARM_21	BOOL
199,5	GLOBAL.ALARM.ALARM_22	BOOL
199,6	GLOBAL.ALARM.ALARM_23	BOOL
199,7	GLOBAL.ALARM.ALARM_24	BOOL
Header		
200,0	HEADER.MODE.AUTOMATIC	BOOL
200,1	HEADER.MODE.CYCLE	BOOL
200,2	HEADER.MODE.STEP	BOOL
200,3	HEADER.MODE.MANUAL	BOOL
...		
201,0	HEADER.MODE.ACTIVE	BOOL
...		
202,0	HEADER.STATUS.WARNING	BOOL
202,1	HEADER.STATUS.ALARM	BOOL
...		
204,0	HEADER.POSITION.HOME	BOOL
...		
206,0	HEADER.TEXTINDEX_1	WORD
208,0	HEADER.TEXTINDEX_2	WORD
210,0	HEADER.CLOCK_MEMORY	BYTE
211,0	HEADER.WATCHDOG	BOOL
211,1	HEADER.WATCHDOG_PLC_STOP	BOOL
212,0	HEADER.RESERVE[1..7]	BYTE
Manual operation screens - general		
220,0	MANUAL_COMMON.POSITION[1]	REAL
224,0	MANUAL_COMMON.POSITION[2]	REAL
228,0	MANUAL_COMMON.POSITION[3]	REAL
232,0	MANUAL_COMMON.POSITION[4]	REAL
236,0	MANUAL_COMMON.POSITION[5]	REAL
240,0	MANUAL_COMMON.POSITION[6]	REAL
244,0	MANUAL_COMMON.POSITION[7]	REAL
248,0	MANUAL_COMMON.POSITION[8]	REAL
252,0	MANUAL_COMMON.NUMBER_OF_PAGE	BYTE
253,0	MANUAL_COMMON.CURRENT_PAGE	BYTE

Address	Name	Type
254,0	MANUAL_COMMON.ROW_VISIBLE_FIRST	WORD
256,0	MANUAL_COMMON.ROW_VISIBLE_LAST	WORD
258,0	MANUAL_COMMON.INTERNAL.CONFIGURATION_ADDR_BYTE	WORD
260,0	MANUAL_COMMON.INTERNAL.CONFIGURATION_ADDR_BIT	WORD
262,0	MANUAL_COMMON.INTERNAL.TEXT_INDEX_ROW [1..8]	WORD
278,0	MANUAL_COMMON.INTERNAL.INTERFACE_ADDR_BYTE	WORD
280,0	MANUAL_COMMON.INTERNAL.OFFSET_INTERFACE	WORD
282,0	MANUAL_COMMON.INTERNAL.OFFSET_CONFIGURATION	WORD
284,0	MANUAL_COMMON.INTERNAL.TIMER_VIEW_VALUE	TIME
288,0	MANUAL_COMMON.INTERNAL.TIMER_VIEW_STATUS	BOOL
288,1	MANUAL_COMMON.INTERNAL.PAGE_LOCKED	BOOL
288,2	MANUAL_COMMON.INTERNAL.PAGE_CHANGED	BOOL
288,3	MANUAL_COMMON.INTERNAL.VIEW_ABSOLUT	BOOL
288,4	MANUAL_COMMON.INTERNAL.MANUAL_SCREEN_ACTIVE	BOOL
289,0	MANUAL_COMMON.INTERNAL.SPARE_305	BYTE
290,0	MANUAL_COMMON.INTERNAL.SPARE_306	WORD
292,0	MANUAL_COMMON.INTERNAL.SPARE_307	WORD
294,0	MANUAL_COMMON.INTERNAL.SELECTED_ROW	INT
296,0	MANUAL_COMMON.CLOSED_SELECTED_ROW	BOOL
296.1	MANUAL_COMMON.RESET_SELECTED_ROW	BOOL
298,0	MANUAL_COMMON.SPARE [1..2]	BYTE
Manual operation screen		
300,0	SCREEN_MANUAL.ROW [1..64, 0..7]	BOOL
364,0	SCREEN_MANUAL.RESERVED [1..6]	BYTE
370,0	SCREEN_MANUAL.ACTUAL_PAGE	BYTE
371,0	SCREEN_MANUAL.PREVIOUS_ACTUAL_PAGE	BYTE
372,0	SCREEN_MANUAL.FIRST_PAGE	BYTE
373,0	SCREEN_MANUAL.LAST_PAGE	BYTE
374,0	SCREEN_MANUAL.SPARE_74	BYTE
375,0	SCREEN_MANUAL.PREVIOUS_FIRST_PAGE	BYTE
376,0	SCREEN_MANUAL.SPARE [1..4]	BYTE
Power up condition		
380,0	SCREEN_POWERUP.ROW [1..64, 0..7]	BOOL
444,0	SCREEN_POWERUP.RESERVED [1..6]	BYTE
450,0	SCREEN_POWERUP.ACTUAL_PAGE	BYTE
451,0	SCREEN_POWERUP.PREVIOUS_ACTUAL_PAGE	BYTE
452,0	SCREEN_POWERUP.SPARE [1..8]	BYTE
Units		
460,0	SCREEN_UNITS.ROW [1..32, 0..7]	BOOL
492,0	SCREEN_UNITS.RESERVED [1..3]	BYTE
496,0	SCREEN_UNITS.ACTUAL_PAGE	BYTE
497,0	SCREEN_UNITS.PREVIOUS_ACTUAL_PAGE	BYTE
498,0	SCREEN_UNITS.SPARE [1..8]	BYTE
Nut runner		
506,0	SCREEN_NUTRUNNER.ROW [1..32, 0..7]	BOOL
538,0	SCREEN_NUTRUNNER.RESERVED [1..3]	BYTE
542,0	SCREEN_NUTRUNNER.ACTUAL_PAGE	BYTE

Address	Name	Type
543,0	SCREEN_NUTRUNNER.PREVIOUS_ACTUAL_PAGE	BYTE
544,0	SCREEN_NUTRUNNER.SPARE [1..8]	BYTE
Nut runner groups		
552,0	SCREEN_NUTRUNNER_GROUP.ROW [1..32, 0..7]	BOOL
584,0	SCREEN_NUTRUNNER_GROUP.RESERVED [1..3]	BYTE
588,0	SCREEN_NUTRUNNER_GROUP.ACTUAL_PAGE	BYTE
589,0	SCREEN_NUTRUNNER_GROUP.PREVIOUS_ACTUAL_PAGE	BYTE
590,0	SCREEN_NUTRUNNER_GROUP.SPARE [1..8]	BYTE
Cycle types		
598,0	SCREEN_CYCLETYPE.ROW [1..16, 0..7]	BOOL
614,0	SCREEN_CYCLETYPE.RESERVED [1..5]	BYTE
620,0	SCREEN_CYCLETYPE.ACTUAL_PAGE	BYTE
621,0	SCREEN_CYCLETYPE.PREVIOUS_ACTUAL_PAGE	BYTE
622,0	SCREEN_CYCLETYPE.SPARE [1..8]	BYTE
User operator screen		
630,0	SCREEN_USER_DEFINED.ROW [1..16, 0..7]	BOOL
646,0	SCREEN_USER_DEFINED.RESERVED [1..5]	BYTE
652,0	SCREEN_USER_DEFINED.ACTUAL_PAGE	BYTE
653,0	SCREEN_USER_DEFINED.PREVIOUS_ACTUAL_PAGE	BYTE
654,0	SCREEN_USER_DEFINED.SPARE [1..8]	BYTE
Cycle times		
662,0	SCREEN_CYCLETIMES.MAIN.LAST	INT
664,0	SCREEN_CYCLETIMES.SUB.LAST [1..15]	INT
694,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[0].START_EDGE	BOOL
694,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[0].STOP_EDGE	BOOL
694,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[0].BREAK_P_EDGE	BOOL
696,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[0].TIMER_START	TIME
700,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[0].TIME_BUFFER	INT
702,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[1].START_EDGE	BOOL
702,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[1].STOP_EDGE	BOOL
702,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[1].BREAK_P_EDGE	BOOL
704,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[1].TIMER_START	TIME
708,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[1].TIME_BUFFER	INT
710,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[2].START_EDGE	BOOL
710,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[2].STOP_EDGE	BOOL
710,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[2].BREAK_P_EDGE	BOOL
712,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[2].TIMER_START	TIME
716,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[2].TIME_BUFFER	INT
718,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[3].START_EDGE	BOOL
718,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[3].STOP_EDGE	BOOL
718,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[3].BREAK_P_EDGE	BOOL
720,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[3].TIMER_START	TIME
724,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[3].TIME_BUFFER	INT
726,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[4].START_EDGE	BOOL
726,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[4].STOP_EDGE	BOOL
726,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[4].BREAK_P_EDGE	BOOL
728,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[4].TIMER_START	TIME

Address	Name	Type
732,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[4].TIME_BUFFER	INT
734,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[5].START_EDGE	BOOL
734,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[5].STOP_EDGE	BOOL
734,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[5].BREAK_P_EDGE	BOOL
736,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[5].TIMER_START	TIME
740,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[5].TIME_BUFFER	INT
742,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[6].START_EDGE	BOOL
742,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[6].STOP_EDGE	BOOL
742,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[6].BREAK_P_EDGE	BOOL
744,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[6].TIMER_START	TIME
748,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[6].TIME_BUFFER	INT
750,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[7].START_EDGE	BOOL
750,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[7].STOP_EDGE	BOOL
750,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[7].BREAK_P_EDGE	BOOL
752,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[7].TIMER_START	TIME
756,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[7].TIME_BUFFER	INT
758,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[8].START_EDGE	BOOL
758,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[8].STOP_EDGE	BOOL
758,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[8].BREAK_P_EDGE	BOOL
760,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[8].TIMER_START	TIME
764,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[8].TIME_BUFFER	INT
766,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[9].START_EDGE	BOOL
766,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[9].STOP_EDGE	BOOL
766,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[9].BREAK_P_EDGE	BOOL
768,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[9].TIMER_START	TIME
772,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[9].TIME_BUFFER	INT
774,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[10].START_EDGE	BOOL
774,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[10].STOP_EDGE	BOOL
774,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[10].BREAK_P_EDGE	BOOL
776,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[10].TIMER_START	TIME
780,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[10].TIME_BUFFER	INT
782,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[11].START_EDGE	BOOL
782,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[11].STOP_EDGE	BOOL
782,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[11].BREAK_P_EDGE	BOOL
784,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[11].TIMER_START	TIME
788,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[11].TIME_BUFFER	INT
790,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[12].START_EDGE	BOOL
790,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[12].STOP_EDGE	BOOL
790,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[12].BREAK_P_EDGE	BOOL
792,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[12].TIMER_START	TIME
796,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[12].TIME_BUFFER	INT
798,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[13].START_EDGE	BOOL
798,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[13].STOP_EDGE	BOOL
798,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[13].BREAK_P_EDGE	BOOL
800,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[13].TIMER_START	TIME
804,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[13].TIME_BUFFER	INT
806,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[14].START_EDGE	BOOL

Address	Name	Type
806,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[14].STOP_EDGE	BOOL
806,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[14].BREAK_P_EDGE	BOOL
808,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[14].TIMER_START	TIME
812,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[14].TIME_BUFFER	INT
814,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[15].START_EDGE	BOOL
814,1	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[15].STOP_EDGE	BOOL
814,2	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[15].BREAK_P_EDGE	BOOL
816,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[15].TIMER_START	TIME
820,0	SCREEN_CYCLETIMES.INTERNAL.CYCLETIME[15].TIME_BUFFER	INT
822,0	SCREEN_CYCLETIMES.SPARE [1.0.10]	BYTE
Interface		
832,0	SCREEN_INTERLOCK.SELECTION	INT
834,0	SCREEN_INTERLOCK.SIGNALS.INPUTS	WORD
836,0	SCREEN_INTERLOCK.SIGNALS.OUTPUTS	WORD
838,0	SCREEN_INTERLOCK.SPARE [1..9]	WORD
Workpiece counter screen		
856,0	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_RESET_SHIFT	BOOL
856,1	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_RESET_DAY	BOOL
856,2	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_RESET_TOTAL	BOOL
856,3	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_COUNT	BOOL
856,4	SCREEN_COUNTER.INTERNAL.SHOW_RESET	BOOL
856,5	SCREEN_COUNTER.INTERNAL.REQ_RESET_SHIFT	BOOL
856,6	SCREEN_COUNTER.INTERNAL.REQ_RESET_DAY	BOOL
856,7	SCREEN_COUNTER.INTERNAL.REQ_RESET_TOTAL	BOOL
857,0	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_SHOW_YESNO	BOOL
857,1	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_YES_OP	BOOL
857,2	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_NO_OP	BOOL
857,3	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_RESET_SHIFT_OP	BOOL
857,4	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_RESET_DAY_OP	BOOL
857,5	SCREEN_COUNTER.INTERNAL.EDGE_FLAG_RESET_TOTAL_OP	BOOL
858,0	SCREEN_COUNTER.INTERNAL.TIME_VALUE_SHOW_YESNO	TIME
862,0	SCREEN_COUNTER.INTERNAL.SPARE [1..4]	BYTE
866,0	SCREEN_COUNTER.SELECTED_COUNTER	WORD
868,0	SCREEN_COUNTER.SPARE [1..8]	BYTE
876,0	SCREEN_COUNTER.OVERALL.TOTAL	DINT
880,0	SCREEN_COUNTER.OVERALL.TOTAL_BAD	INT
882,0	SCREEN_COUNTER.OVERALL.DAY	INT
884,0	SCREEN_COUNTER.OVERALL.DAY_BAD	INT
886,0	SCREEN_COUNTER.OVERALL.SHIFT	INT
888,0	SCREEN_COUNTER.OVERALL.SHIFT_BAD	INT
890,0	SCREEN_COUNTER.SPECIFIC[1].TOTAL	DINT
894,0	SCREEN_COUNTER.SPECIFIC[1].TOTAL_BAD	INT
896,0	SCREEN_COUNTER.SPECIFIC[1].DAY	INT
898,0	SCREEN_COUNTER.SPECIFIC[1].DAY_BAD	INT
900,0	SCREEN_COUNTER.SPECIFIC[1].SHIFT	INT
902,0	SCREEN_COUNTER.SPECIFIC[1].SHIFT_BAD	INT
904,0	SCREEN_COUNTER.SPECIFIC[2].TOTAL	DINT

Address	Name	Type
908,0	SCREEN_COUNTER.SPECIFIC[2].TOTAL_BAD	INT
910,0	SCREEN_COUNTER.SPECIFIC[2].DAY	INT
912,0	SCREEN_COUNTER.SPECIFIC[2].DAY_BAD	INT
914,0	SCREEN_COUNTER.SPECIFIC[2].SHIFT	INT
916,0	SCREEN_COUNTER.SPECIFIC[2].SHIFT_BAD	INT
918,0	SCREEN_COUNTER.SPECIFIC[3].TOTAL	DINT
922,0	SCREEN_COUNTER.SPECIFIC[3].TOTAL_BAD	INT
924,0	SCREEN_COUNTER.SPECIFIC[3].DAY	INT
926,0	SCREEN_COUNTER.SPECIFIC[3].DAY_BAD	INT
928,0	SCREEN_COUNTER.SPECIFIC[3].SHIFT	INT
930,0	SCREEN_COUNTER.SPECIFIC[3].SHIFT_BAD	INT
932,0	SCREEN_COUNTER.SPECIFIC[4].TOTAL	DINT
936,0	SCREEN_COUNTER.SPECIFIC[4].TOTAL_BAD	INT
938,0	SCREEN_COUNTER.SPECIFIC[4].DAY	INT
940,0	SCREEN_COUNTER.SPECIFIC[4].DAY_BAD	INT
942,0	SCREEN_COUNTER.SPECIFIC[4].SHIFT	INT
944,0	SCREEN_COUNTER.SPECIFIC[4].SHIFT_BAD	INT
946,0	SCREEN_COUNTER.SPECIFIC[5].TOTAL	DINT
950,0	SCREEN_COUNTER.SPECIFIC[5].TOTAL_BAD	INT
952,0	SCREEN_COUNTER.SPECIFIC[5].DAY	INT
954,0	SCREEN_COUNTER.SPECIFIC[5].DAY_BAD	INT
956,0	SCREEN_COUNTER.SPECIFIC[5].SHIFT	INT
958,0	SCREEN_COUNTER.SPECIFIC[5].SHIFT_BAD	INT
960,0	SCREEN_COUNTER.SPECIFIC[6].TOTAL	DINT
964,0	SCREEN_COUNTER.SPECIFIC[6].TOTAL_BAD	INT
966,0	SCREEN_COUNTER.SPECIFIC[6].DAY	INT
968,0	SCREEN_COUNTER.SPECIFIC[6].DAY_BAD	INT
970,0	SCREEN_COUNTER.SPECIFIC[6].SHIFT	INT
972,0	SCREEN_COUNTER.SPECIFIC[6].SHIFT_BAD	INT
974,0	SCREEN_COUNTER.SPECIFIC[7].TOTAL	DINT
978,0	SCREEN_COUNTER.SPECIFIC[7].TOTAL_BAD	INT
980,0	SCREEN_COUNTER.SPECIFIC[7].DAY	INT
982,0	SCREEN_COUNTER.SPECIFIC[7].DAY_BAD	INT
984,0	SCREEN_COUNTER.SPECIFIC[7].SHIFT	INT
986,0	SCREEN_COUNTER.SPECIFIC[7].SHIFT_BAD	INT
988,0	SCREEN_COUNTER.SPECIFIC[8].TOTAL	DINT
992,0	SCREEN_COUNTER.SPECIFIC[8].TOTAL_BAD	INT
994,0	SCREEN_COUNTER.SPECIFIC[8].DAY	INT
996,0	SCREEN_COUNTER.SPECIFIC[8].DAY_BAD	INT
998,0	SCREEN_COUNTER.SPECIFIC[8].SHIFT	INT
1000,0	SCREEN_COUNTER.SPECIFIC[8].SHIFT_BAD	INT
1002,0	SCREEN_COUNTER.SPECIFIC[9].TOTAL	DINT
1006,0	SCREEN_COUNTER.SPECIFIC[9].TOTAL_BAD	INT
1008,0	SCREEN_COUNTER.SPECIFIC[9].DAY	INT
1010,0	SCREEN_COUNTER.SPECIFIC[9].DAY_BAD	INT
1012,0	SCREEN_COUNTER.SPECIFIC[9].SHIFT	INT
1014,0	SCREEN_COUNTER.SPECIFIC[9].SHIFT_BAD	INT

Address	Name	Type
1016,0	SCREEN_COUNTER.SPECIFIC[10].TOTAL	DINT
1020,0	SCREEN_COUNTER.SPECIFIC[10].TOTAL_BAD	INT
1022,0	SCREEN_COUNTER.SPECIFIC[10].DAY	INT
1024,0	SCREEN_COUNTER.SPECIFIC[10].DAY_BAD	INT
1026,0	SCREEN_COUNTER.SPECIFIC[10].SHIFT	INT
1028,0	SCREEN_COUNTER.SPECIFIC[10].SHIFT_BAD	INT
1030,0	SCREEN_COUNTER.SPECIFIC[11].TOTAL	DINT
1034,0	SCREEN_COUNTER.SPECIFIC[11].TOTAL_BAD	INT
1036,0	SCREEN_COUNTER.SPECIFIC[11].DAY	INT
1038,0	SCREEN_COUNTER.SPECIFIC[11].DAY_BAD	INT
1040,0	SCREEN_COUNTER.SPECIFIC[11].SHIFT	INT
1042,0	SCREEN_COUNTER.SPECIFIC[11].SHIFT_BAD	INT
1044,0	SCREEN_COUNTER.SPECIFIC[12].TOTAL	DINT
1048,0	SCREEN_COUNTER.SPECIFIC[12].TOTAL_BAD	INT
1050,0	SCREEN_COUNTER.SPECIFIC[12].DAY	INT
1052,0	SCREEN_COUNTER.SPECIFIC[12].DAY_BAD	INT
1054,0	SCREEN_COUNTER.SPECIFIC[12].SHIFT	INT
1056,0	SCREEN_COUNTER.SPECIFIC[12].SHIFT_BAD	INT
1058,0	SCREEN_COUNTER.SPECIFIC[13].TOTAL	DINT
1062,0	SCREEN_COUNTER.SPECIFIC[13].TOTAL_BAD	INT
1064,0	SCREEN_COUNTER.SPECIFIC[13].DAY	INT
1066,0	SCREEN_COUNTER.SPECIFIC[13].DAY_BAD	INT
1068,0	SCREEN_COUNTER.SPECIFIC[13].SHIFT	INT
1070,0	SCREEN_COUNTER.SPECIFIC[13].SHIFT_BAD	INT
1072,0	SCREEN_COUNTER.SPECIFIC[14].TOTAL	DINT
1076,0	SCREEN_COUNTER.SPECIFIC[14].TOTAL_BAD	INT
1078,0	SCREEN_COUNTER.SPECIFIC[14].DAY	INT
1080,0	SCREEN_COUNTER.SPECIFIC[14].DAY_BAD	INT
1082,0	SCREEN_COUNTER.SPECIFIC[14].SHIFT	INT
1084,0	SCREEN_COUNTER.SPECIFIC[14].SHIFT_BAD	INT
1086,0	SCREEN_COUNTER.SPECIFIC[15].TOTAL	DINT
1090,0	SCREEN_COUNTER.SPECIFIC[15].TOTAL_BAD	INT
1092,0	SCREEN_COUNTER.SPECIFIC[15].DAY	INT
1094,0	SCREEN_COUNTER.SPECIFIC[15].DAY_BAD	INT
1096,0	SCREEN_COUNTER.SPECIFIC[15].SHIFT	INT
1098,0	SCREEN_COUNTER.SPECIFIC[15].SHIFT_BAD	INT

Table A-2 Structure of the DB_HMILITE_DATA data block

A.4 Structure of the DB_HMILITE_CONFIG data block

Address	Name	Type
Software Version		
0,0	RELEASE.MAJOR	BYTE
1,0	RELEASE.MINOR	BYTE
Global variables		
2,0	GLOBAL.SPARE [1.0.18]	BYTE
Header		
20,0	HEADER.SPARE [1.0.20]	BYTE
Manual operation screens - general		
40,0	MANUAL_COMMON.SPARE_0	INT
42,0	MANUAL_COMMON.TOUCH_PRESELECTION_TIME	TIME
46,0	MANUAL_COMMON.SCREEN_ACTIVE_TIME	TIME
50,0	MANUAL_COMMON.ABSOLUTE_DISPLAY_TIME	TIME
54,0	MANUAL_COMMON.SPARE [1.0.10]	BYTE
Manual operation screen		
64,0	SCREEN_MANUAL.ROW_01	BYTE
65,0	SCREEN_MANUAL.ROW_02	BYTE
...		
127,0	SCREEN_MANUAL.ROW_64	BYTE
...		
134,0	SCREEN_MANUAL.NUMBER_OF_ROWS	INT
136,0	SCREEN_MANUAL.OFFSET_DB_CONFIG	WORD
138,0	SCREEN_MANUAL.OFFSET_DB_DATA	WORD
140,0	SCREEN_MANUAL.SPARE [1.0.6]	BYTE
Power up condition		
146,0	SCREEN_POWERUP.ROW_01	BYTE
147,0	SCREEN_POWERUP.ROW_02	BYTE
...		
209,0	SCREEN_POWERUP.ROW_64	BYTE
...		
216,0	SCREEN_POWERUP.NUMBER_OF_ROWS	INT
218,0	SCREEN_POWERUP.SPARE [1.0.10]	BYTE
Unit selection/deselection		
228,0	SCREEN_UNITS.ROW_01	BYTE
229,0	SCREEN_UNITS.ROW_02	BYTE
...		
259,0	SCREEN_UNITS.ROW_32	BYTE
...		
264,0	SCREEN_UNITS.NUMBER_OF_ROWS	INT
266,0	SCREEN_UNITS.SPARE [1.0.10]	BYTE
Nut runner selection/deselection		
276,0	SCREEN_NUTRUNNER.ROW_01	BYTE
277,0	SCREEN_NUTRUNNER.ROW_02	BYTE
...		

Address	Name	Type
307,0	SCREEN_NUTRUNNER.ROW_32	BYTE
...		
312,0	SCREEN_NUTRUNNER.NUMBER_OF_ROWS	INT
314,0	SCREEN_NUTRUNNER.SPARE [1.0.10]	BYTE
Nut runner groups selection/deselection		
324,0	SCREEN_NUTRUNNER_GROUP.ROW_01	BYTE
325,0	SCREEN_NUTRUNNER_GROUP.ROW_02	BYTE
...		
355,0	SCREEN_NUTRUNNER_GROUP.ROW_32	BYTE
...		
360,0	SCREEN_NUTRUNNER_GROUP.NUMBER_OF_ROWS	INT
362,0	SCREEN_NUTRUNNER_GROUP.SPARE [1.0.10]	BYTE
Cycle types		
372,0	SCREEN_CYCLETYPE.ROW_01	BYTE
373,0	SCREEN_CYCLETYPE.ROW_02	BYTE
...		
387,0	SCREEN_CYCLETYPE.ROW_01	BYTE
...		
394,0	SCREEN_CYCLETYPE.NUMBER_OF_ROWS	INT
396,0	SCREEN_CYCLETYPE.SPARE [1.0.10]	BYTE
User operator screen		
406,0	SCREEN_USER_DEFINED.ROW_01	BYTE
407,0	SCREEN_USER_DEFINED.ROW_02	BYTE
...		
421,0	SCREEN_USER_DEFINED.ROW_16	BYTE
...		
428,0	SCREEN_USER_DEFINED.NUMBER_OF_ROWS	INT
430,0	SCREEN_USER_DEFINED.SPARE [1.0.10]	BYTE
Cycle times		
440,0	SCREEN_CYCLETIMES.MAIN.TARGET	INT
442,0	SCREEN_CYCLETIMES.SUB.TARGET_01	INT
444,0	SCREEN_CYCLETIMES.SUB.TARGET_02	INT
...		
470,0	SCREEN_CYCLETIMES.SUB.TARGET_15	INT
472,0	SCREEN_CYCLETIMES.HIDE_SPECIFIC	BOOL
474,0	SCREEN_CYCLETIMES.SPARE [1.0.10]	BYTE
Workpiece counter		
484,0	SCREEN_COUNTER.HIDE_TYPE_SPECIFICFALSE	BOOL
484,1	SCREEN_COUNTER.HIDE_RESET_SHIFTTRUE	BOOL
484,2	SCREEN_COUNTER.HIDE_RESET_DAYTRUE	BOOL
484,3	SCREEN_COUNTER.HIDE_RESET_TOTALTRUE	BOOL
486,0	SCREEN_COUNTER.NUMBER_OF_COUNTER16	INT
488,0	SCREEN_COUNTER.TIME_VALUE_HIDE_RESET	TIME
492,0	SCREEN_COUNTER.SPARE[1]	BYTE
493,0	SCREEN_COUNTER.SPARE[2]	BYTE
494,0	SCREEN_COUNTER.OVERALL.SHIFT_TARGET	INT
496,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_01	INT

Address	Name	Type
498,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_02	INT
500,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_03	INT
502,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_04	INT
504,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_05	INT
506,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_06	INT
508,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_07	INT
510,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_08	INT
512,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_09	INT
514,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_10	INT
516,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_11	INT
518,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_12	INT
520,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_13	INT
522,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_14	INT
524,0	SCREEN_COUNTER.TYPE_SPECIFIC.SHIFT_TARGET_15	INT

Table A-3 Structure of the DB_HMILITE_CONFIG data block

A.5 Structure of the DB_DEVICE_DIAGNOSE data block

Address	Name	Type
0,0	BILD_PROFIBUS_POINTER	DINT
PROFIBUS diagnostics		
4,0	SCREEN_PROFIBUS.DP_SLAVE_IDENT	WORD
6,0	SCREEN_PROFIBUS.DP_SLAVE_IDENT_OP	WORD
8,0	SCREEN_PROFIBUS.SLAVE_ADDRESS_IN	BYTE
9,0	SCREEN_PROFIBUS.SLAVE_ADDRESS_OUT	BYTE
10,0	SCREEN_PROFIBUS.ACTUAL_SLAVE_STATUS	BYTE
11,0	SCREEN_PROFIBUS.NEXT_SLAVE	BOOL
11,1	SCREEN_PROFIBUS.SPARE_71	BOOL
11,2	SCREEN_PROFIBUS.MANUAL_MODE	BOOL
11,3	SCREEN_PROFIBUS.NO_DETAIL_MODE	BOOL
11,4	SCREEN_PROFIBUS.SPARE_74	BOOL
11,5	SCREEN_PROFIBUS.SPARE_75	BOOL
11,6	SCREEN_PROFIBUS.SPARE_76	BOOL
11,7	SCREEN_PROFIBUS.RESET	BOOL
12,0	SCREEN_PROFIBUS.FIRST_SLAVE_FAILURE	BYTE
13,0	SCREEN_PROFIBUS.CABLE_DIAGNOSTIC_CASE	BYTE
14,0	SCREEN_PROFIBUS.SLAVE_ NO[1] ... SLAVE_NO[126]	BYTE
140,0	SCREEN_PROFIBUS.SLAVE_STATUS[1] ... SLAVE_STATUS[126]	BYTE
266.0	SCREEN_PROFIBUS.ALARM_NUMBER.FAULTY_SLAVE	INT
268.0	SCREEN_PROFIBUS.ALARM_NUMBER.FAILED_SLAVE	INT
270.0	SCREEN_PROFIBUS.INTERNAL.FAULTY_SLAVES [1] ... [4]	DWORD
286.0	SCREEN_PROFIBUS.INTERNAL.FAILED_SLAVES [1] ... [4]	DWORD
302.0	SCREEN_PROFIBUS.INTERNAL.DATA_FIELD [1] ... [50]	BYTE
352.0	SCREEN_PROFIBUS.INTERNAL.SUM_SLAVES_DIAG	INT

Address	Name	Type
354.0	SCREEN_PROFIBUS.INTERNAL.RETVAL	INT
356.0	SCREEN_PROFIBUS.INTERNAL.BUSY	BOOL
356.1	SCREEN_PROFIBUS.INTERNAL.RISING_EDGE_MANUAL_MODE	BOOL
356.2	SCREEN_PROFIBUS.INTERNAL.SCAN_SLAVES_DONE	BOOL
356.3	SCREEN_PROFIBUS.INTERNAL.EDGE_NEXT_SLAVE	BOOL
356.4	SCREEN_PROFIBUS.INTERNAL.HISTORY_COORDINATION	BOOL
356.5	SCREEN_PROFIBUS.INTERNAL.SPARE_863 ... SPARE_877	BOOL
AS-i diagnostics		
358,0	SCREEN_ASI.SELECTED_CP	BYTE
359,0	SCREEN_ASI.SPARE_ 10 … SPARE_17	BOOL
360,0	SCREEN_ASI.SLAVE_STATUS[1] … SLAVE_STATUS[64]	BYTE
424,0	SCREEN_ASI.ASI_FLAGS	WORD
426,0	SCREEN_ASI.SPARE_68	INT
428,0	SCREEN_ASI.INTERNAL.OB82_MDL_ADDR	INT
430,0	SCREEN_ASI.INTERNAL.SEND[0] ... SEND[1]	BYTE
432,0	SCREEN_ASI.INTERNAL.RECV[0] ... RECV[31]	BYTE
464,0	SCREEN_ASI.INTERNAL.STATUS [0] ... STATUS[7]	DWORD
496,0	SCREEN_ASI.INTERNAL.MODE	BYTE
497,0	SCREEN_ASI.INTERNAL.ASI_TYPE	BYTE
498,0	SCREEN_ASI.INTERNAL.ASI_LADDR	WORD
500,0	SCREEN_ASI.INTERNAL.RECORD_LENGTH	WORD
502,0	SCREEN_ASI.INTERNAL.TYPE_SLAVES_AB	BOOL
502,1	SCREEN_ASI.INTERNAL.FLAGS_ASI_SPEC	BOOL
502,2	SCREEN_ASI.INTERNAL.ALIVE	BOOL
502,3	SCREEN_ASI.INTERNAL.ALIVE_STATUS	BOOL
502,4	SCREEN_ASI.INTERNAL.PULSE_1SEC	BOOL
502,5	SCREEN_ASI.INTERNAL.BUSY	BOOL
502,6	SCREEN_ASI.INTERNAL.CANCEL	BOOL
502,7	SCREEN_ASI.INTERNAL.OB82_FAULT	BOOL
503,0	SCREEN_ASI.INTERNAL.ASI_INDEX	BYTE
504,0	SCREEN_ASI.SCREEN_ASI_MONITOR.SELECTED_MONITOR	BYTE
505,0	SCREEN_ASI.SCREEN_ASI_MONITOR.SPARE_10	BYTE
506,0	SCREEN_ASI.SCREEN_ASI_MONITOR.RETVAL_FB102	INT
	SCREEN_ASI.SCREEN_ASI_MONITOR.DIAGNOSTIC_DATA...	
508,0	...diag.DataSelect_All_Devices	BOOL
508,1	...diag.data_doublecheck	BOOL
508,2	...diag.data_ready_tripped_ch_1	BOOL
508,3	...diag.data_ready_tripped_ch_2	BOOL
508,4	...diag.diagnosis_done	BOOL
508,5	...diag.diagnosis_error	BOOL
508,6	...diag.Channel_1_off	BOOL
508,7	...diag.Channel_2_off	BOOL
509,0	...diag.status_monitor	BYTE
510,0	...diag.status_channel[1]	BYTE
511,0	...diag.status_channel[2]	BYTE
512,0	...diag.quantity[1]	BYTE
513,0	...diag.quantity[2]	BYTE

Address	Name	Type
514,0	...device[32-79].channel_info	BYTE
515,0	...diag.device[32-79].status	BYTE
610,0	...info.max_device_in_channel[1].index	BYTE
611,0	...info.max_device_in_channel[1].status	BYTE
612,0	...info.max_device_in_channel[2].index	BYTE
613,0	...info.max_device_in_channel[2].status	BYTE
614,0	...info.retval	BYTE
616,0	...info.Config_Mode	BOOL
616,1	...info.Startup_Phase	BOOL
616,2	...info.Com_Error	BOOL
618,0	...time_stamp.actual_time	TIME
626,0	...time_stamp.OSSD_1_off	TIME
634,0	...time_stamp.OSSD_1_on	TIME
642,0	...time_stamp.OSSD_2_off	TIME
650,0	...time_stamp.OSSD_2_on	TIME
658,0	...time_stamp.Config_Mode_asimon	TIME
666,0	...time_stamp.Startup_Phase_asimon	TIME
674,0	...time_stamp.Fault_or_Error_FB	TIME
682,0	...time_stamp.Startup_Diagnosis_FB	TIME
Drive diagnostics		
690,0	SCREEN_DRIVE.INDEX	BYTE
691,0	SCREEN_DRIVE.DRIVE_TYPE	BYTE
692,0	SCREEN_DRIVE.STATUS_WORD	WORD
694,0	SCREEN_DRIVE.CONTROL_WORD	WORD
696,0	SCREEN_DRIVE.POS_STATUS_WORD	WORD
698,0	SCREEN_DRIVE.POS_ACT_VALUE	WORD
702,0	SCREEN_DRIVE.OVERRIDE	DWORD
706,0	SCREEN_DRIVE.BLOCK_SELECT	WORD
708,0	SCREEN_DRIVE.BLOCK_CURRENT	WORD
710,0	SCREEN_DRIVE.FAULT_CODE [1] … FAULT_CODE[8]	WORD
726,0	SCREEN_DRIVE.ALARM_CODE [1] … ALARM_CODE[4]	WORD
734,0	SCREEN_DRIVE.DIMENSION	WORD
736,0	SCREEN_DRIVE.ALARM_NUMBER.FAULTY_AXIS	INT
738,0	SCREEN_DRIVE.ALARM_NUMBER.SPARE_2	INT
740,0	SCREEN_DRIVE.INTERNAL.PCV_WRITE.PCA	WORD
742,0	SCREEN_DRIVE.INTERNAL.PCV_WRITE.IND	WORD
744,0	SCREEN_DRIVE.INTERNAL.PCV_WRITE.PVA	DWORD
748,0	SCREEN_DRIVE.INTERNAL.PCV_READ.PCA	WORD
750,0	SCREEN_DRIVE.INTERNAL.PCV_READ.IND	WORD
752,0	SCREEN_DRIVE.INTERNAL.PCV_READ.PVA	DWORD
756,0	SCREEN_DRIVE.INTERNAL.MODE	BYTE
757,0	SCREEN_DRIVE.INTERNAL.TASK	BYTE
758,0	SCREEN_DRIVE.INTERNAL.TASK_LIST [1] … TASK_LIST[10]	BYTE
768,0	SCREEN_DRIVE.INTERNAL.TASK_PTR	DWORD
772,0	SCREEN_DRIVE.INTERNAL.DEST_PTR	DWORD
776,0	SCREEN_DRIVE.INTERNAL.INDEX_PREV	BYTE
777,0	SCREEN_DRIVE.INTERNAL.FAULT_PRESENT	BOOL

Address	Name	Type
777,1	SCREEN_DRIVE.INTERNAL.ALARM_PRESENT	BOOL
777,2	SCREEN_DRIVE.INTERNAL.ALIVE	BOOL
777,3	SCREEN_DRIVE.INTERNAL.ALIVE_STATUS	BOOL
777,4	SCREEN_DRIVE.INTERNAL.PULSE_1SEC	BOOL
777,5	SCREEN_DRIVE.INTERNAL.SPARE_375	BOOL
777,6	SCREEN_DRIVE.INTERNAL.SPARE_376	BOOL
777,7	SCREEN_DRIVE.INTERNAL.SPARE_377	BOOL
778,0	SCREEN_DRIVE.INTERNAL.ALARM_CODE	WORD
780,0	SCREEN_DRIVE.INTERNAL.ALARM_NUMBER	WORD
MOBY diagnostics		
782,0	SCREEN_MOBY.SELECTED_SLG	BYTE
783,0	SCREEN_MOBY.MOBY_TYPE	BYTE
784,0	SCREEN_MOBY.STATUS	WORD
786,0	SCREEN_MOBY.ASM_ERROR	INT
788,0	SCREEN_MOBY.PROG_ERROR	INT
790,0	SCREEN_MOBY.BUS_ERROR	INT
792,0	SCREEN_MOBY.INTERNAL.FAULTY_SLG	BYTE
793,0	SCREEN_MOBY.INTERNAL.SPARE_1	BYTE
794,0	SCREEN_MOBY.INTERNAL.SPARE_2	INT
796,0	SCREEN_MOBY.INTERNAL.SPARE_3	INT
798,0	SCREEN_MOBY.INTERNAL.SPARE_4	INT
MOBY diagnostics		
800.0	SCREEN_SINAMICS.SELECTED_DRIVEOBJECT	BYTE
801.0	SCREEN_SINAMICS.DRIVE_MODE	BYTE
802.0	SCREEN_SINAMICS.CONTROL_WORD	WORD
804.0	SCREEN_SINAMICS.STATUS_WORD	WORD
806.0	SCREEN_SINAMICS.POS_STATUS_WORD	WORD
808.0	SCREEN_SINAMICS.POS_SET_VALUE	DINT
812.0	SCREEN_SINAMICS.POS_ACT_VALUE	DINT
816.0	SCREEN_SINAMICS.SPEED_ACT_VALUE	DINT
820.0	SCREEN_SINAMICS.OVERRIDE	REAL
824.0	SCREEN_SINAMICS.BLOCK_SELECT	INT
826.0	SCREEN_SINAMICS.BLOCK_CURRENT	DINT
830.0	SCREEN_SINAMICS.FAULT_CODE[1..8]	WORD
846.0	SCREEN_SINAMICS.ALARM_CODE [1..4]	WORD
854.0	SCREEN_SINAMICS. DIMENSION	WORD

Table A-4 Structure of the DB_DEVICE_DIAGNOSE data block

A.6 Change log

A.6.1 Edition 03/2003

First Issue

A.6.2 Changes from 03/2003 edition to 03/2004 edition

Complete revision of the manual
Chapters renumbered
Division into manual and configuration guide

A.6.3 Changes from 03/2004 edition to 05/2005 edition

Chapter 1 Introduction

Section 1.4 Hardware requirements
Formal revision

Section 1.5 Software requirements
Software versions corrected

Chapter 2 Installation

Section 2.2 Procedure
Formal revision

Section 2.3 Inclusion of your ProTool project in STEP 7
Formal revision
OP270_6 added

Section 2.5 PROFIBUS configuration with direct key option
Formal revision
Note for safety grounds that the direct keys should be used

Section 2.6 PLC program blocks
Corrected for Version 4.0

Section 2.7 Working with data blocks
Formal revision

Chapter 3 Global Settings and Functionality

Section 3.1 Layout of the screens and basic screen elements
Formal revision

Section 3.2 Menu structure
Formal revision

Section 3.3 Screen "template"

Formal revision

Section 3.4 Designation conventions
Formal revision

Section 3.5 Clock memory byte of the controller
Formal revision

Section 3.6 PLC system time
Formal revision

Section 3.7 Identification of the selected screen
Formal revision

Section 3.8 HMI Lite job mailbox
Formal revision

Section 3.9 FC_HMICE_BASIC
Formal revision
Input parameters revised

Section 3.10 Connection of several operator panels to a controller
New

Chapter 4 Header and Operator Information

Section 4.1 Header
Formal revision
Section 4.1.7 "Change of the display of the status signals in the header“ added

Chapter 5 Manual Operation

Section 5.1 Overview
Formal revision

Section 5.2 Purpose of the individual manual operation screens
Formal revision

Section 5.3 Configuration and runtime interface
Formal revision

Section 5.4 Configuring
Formal revision
Description of the "Touch-panel support activation" variable removed
Section 5.4.3 "Grouping of the movement lines in the manual operation screen" added, new function of the HMI Lite Version 4.0; this means all subsequent sections are displaced

Section 5.5 Runtime interface
Formal revision

Section 5.6 Control interface
Formal revision

Section 5.7 FC_HMICE_MANUAL
Formal revision
Description of the input and output parameters revised

Section 5.8 FB_HMICE_S7GRAPH_MANUAL
New

Section 5.9 Step-by-step procedure
Formal revision

Chapter 6 Production Data Screens

Section 6.1 "Clock times" screen
Formal revision

Section 6.2 "Workpiece counter" screen
Formal revision

Chapter 7 Diagnosis

Section 7.3 "Version" screen
Formal revision
Graphic for the version screen of the HMI Lite Version 4.0 added

Chapter 8 Hardware Diagnostics

Section 8.1 "PROFIBUS" screen
Formal revision
Graphic for the HMI Lite Version 4.0 added
Description for the "Use of the simple PROFIBUS diagnostics" added
Section 8.1.6 Runtime interface (FC_HMICE_PROFIBUS) added
Section 8.1.7 Runtime interface (FC_SIEM_DP_DIAG_OVERVIEW (FC96)) added

Section 8.2 "Drive" screen
Formal revision
"Configuring the DB_HMICE_CONFIG" description removed
Section 8.2.5 "Runtime interface (FC_HMICE_DRIVE)" added

Section 8.3 "MOBY-I" screen
Formal revision
Section 8.3.2 "Supported MOBY interface modules" extended with the ASM 452 - Filehandler
"Configuring the DB_HMICE_CONFIG" description removed
Section 8.3.4 Runtime interface (FC_HMICE_MOBY) added

Section 8.4 "AS-i" screen
Formal revision
"Configuring the DB_HMICE_CONFIG" description removed
Section 8.4.3 Runtime interface (FC_HMICE_ASI) added

Section 8.5 FC_HMICE_DIAGNOSIS and FC_DEVICE_DIAGN
Section removed

Chapter 9 System Screens

Section 9.1 "System" screen
Formal revision

Section 9.2 "Panel Control" screen
Formal revision

Section 9.3 "S7-CPU diagnostics" screen
Formal revision

Section 9.4 "Status Variable" screen
Formal revision

Main chapter

Registers reassigned

Chapter A Appendix

Appendix A.3 Structure of the DB_HMICE_DATA data block
Updated to Version 4.0

Appendix A.4 Structure of the DB_HMICE_CONFIG data block
Updated to Version 4.0

Appendix A.5 Structure of the DB_DEVICE_DIAGNOSE data block
New

Appendix A.6 Change log
New

A.6.4 Changes from 05/2005 edition to 03/2007 edition

Change of the designations, etc. through all chapters
(Function: Find - Replace)

Old designation	*New designation*
HMI Lite CE	HMI Lite
HMICE	HMILITE
ProTool	WinCC flexible
(old operator panel types)	MP277 and OP177B (TP177B)

Chapter 1 Introduction

Section 1.2 Provided screens
Notes (* and **) for the incomplete configuring in the HMI project removed;
menu tile changed and corrected (rename and delete)

Section 1.4 Hardware requirements
Table 1-1 changed for the new operator panel types
Screen 1-2 (Chp1_Panels.ppt) new and the representation of new device types added
Information text changed to the new menu text in WinCC flexible ("Change Device Type...")

Section 1.5 Software requirements

Software versions changed
Special order number added
Service Packs – table corrected

Chapter 2 Installation

Section 2.1 Unpacking the source project

Project structure in Figure 2-1 changed to the new structure (incl. WinCC flexible)

Section 2.3 Inclusion of your WinCC flexible project in STEP 7

Table 2-1 changed for WinCC flexible (name assignment, new devices and icons)

Section 2.6 PLC program blocks

Section 2.6.1 "PLC blocks from HMI Lite" adapted to the project structure
Section 2.6.2 "Schema for calling the function blocks" in Figure 2-2 (names and designation) changed for Version 5.0

Chapter 3 Global Settings and Functionality

Section 3.2 Menu structure

Figure 3-2 (Chp3_MneuStructure_6Inch.ppt) and Figure 3-3 (Chp3_MneuStructure.ppt) changed

Section 3.7 Identification of the selected screen

Configuring of screen events in Figure 3-5 (Chp3_SS_00_ScreenIdentification.ppt) changed to represent the WinCC flexible user interface
Table 3-2 changed and extended (affected screen numbers 53/54/55/57/58)

Section 3.9 FC_HMILITE_BASIC

Figure 3-7 Block representation (Chp3_FC_HMILITE_BASIC.jpg) changed

Chapter 4 Header and Operator Information

Section 4.1 Header

Section 4.1.1 "Layout of the header" changed for 6" devices from mono to color

Chapter 5 Manual Operation

Section 5.1 Overview

Figure 5-2 not replaced, old user interface for the operator panel remains the same in the basic structure, no change other than the colors

Section 5.4 Configuring

Section 5.4.3 "Grouping of movement lines in the manual operation screen" Figure 5-7 (Chp5_KeyConfig.jpg) changed to reflect the WinCC flexible user interface

Section 5.7 FC_HMILITE_MANUAL

Block representation in Figure 5-15 (Chp5_FC_HMILITE_MANUAL.jpg) changed
Note for the handling of the lock for external key mode added

Section 5.8 FB_HMILITE_S7G_MANUAL

Block representation in Figure 5-16 (Chp5_FB_HMILITE_S7G_MANUAL.jpg) changed

Chapter 6 Production Data Screens

Section 6.1 "Clock times" screen

Section 6.1.1 "Layout of the screen and functionality" Figure 6-1 (Chp6_ScreenCycleTime.ppt) key sequence changed
Text in the "Procedure for the clock time" section extended with the functionality that the measurement of the clock time can be interrupted
Section 6.1.2 "Runtime interface (FC_HMILITE_CYCLETIME)" block representation in Figure 6-2 (Chp6_FC_HMILITE_CYCLETIME.jpg) changed and Table 6-1 extended with the "BREAK" entry

Section 6.2 "Workpiece counter" screen

Section 6.2.1 "Layout of the screen and functionality" Figure 6-3 (Chp6_ScreenPartCount.ppt) and Figure 6-4 (Chp6_PartCountYesNo.ppt) key sequence changed
Section 6.2.2 "Runtime interface (FC_HMILITE_COUNTER)" block representation in Figure 6-5 (Chp6_FC_HMILITE_COUNTER.jpg) changed

Chapter 7 Diagnosis

Section 7.1 "Messages" screen and "Alarm archive" screen

Section 7.1.1 "Layout of the screen and functionality", change of the text because the alarm structure in WinCC flexible differs (no storage medium)
Section 7.1.3 "Configuration" new text, Figure 7-2 for the creation of the alarm display object in the Alarm archive screen added

Section 7.3 "Version" screen

Figure 7-4 (previously 7-3) (Chp7_ScreenVersion.ppt) changed to reflect HMI Lite Version 5.0

Chapter 8 Hardware Diagnostics

Section 8.1 "PROFIBUS" screen

Section 8.1.6 "Runtime interface (FC_HMILITE_PROFIBUS)" Figure 8-7 (Chp8_FC_HMILITE_PROFIBUS.jpg) changed
Section 8.1.7 "Runtime interface (FC_SIEM_DP_DIAG_OVERVIEW (FC96))" Figure 8-8 (Chp8_FC_SIEM_DP_DIAG_OVERVIEW.jpg) changed

Section 8.2 "Drive" screen

Section 8.2.3 "Drive position" screen, note added that the Position screen is supplied with data only for 611
Section added "Runtime interface (FC_HMILITE_DRIVE)" Figure 8-12 (Chp8_FC_HMILITE_DRIVE.jpg) changed

Section 8.3 "MOBY-I" screen

Section 8.3.2 "Supported MOBY interface modules" extended with the ASM 456
Section 8.3.4 "Runtime interface (FC_HMILITE_MOBY)" Figure 8-14 (Chp8_FC_HMILITE_MOBY.jpg) changed

Section 8.4 "AS-i" screen

Section 8.4.1 "AS-i diagnosos" screen, title changed, and Figure 8-15 (Chp8_ScreenASIDiagnose.ppt) changed because the screen selection changed with an addition
Section 8.4.3 "Runtime interface (FC_HMILITE_ASI)" Figure 8-17 (Chp8_FC_HMILITE_ASI.jpg) changed
New: Section 8.4.4 "ASIsafe Monitor" screen added
New: Section 8.4.5 Configuring the WinCC flexible screens for ASIsafe Monitor added
New: Section 8.4.6 Runtime interface (FB_ASIMON2D) added

Chapter 9 System Screens

Section 9.1 "System" screen

Section 9.1.1 "Layout of the screen and functionality" Figure 9-1 (Chp9_ScreenSystemGeneral.ppt) changed, the screen selection for S7-CPU diagnostics removed
Extension of the language scope and thus also the functionality for the "Change language" item

Section 9.2 "Panel Control" screen

Formal revision
Section 9.2.1 "Layout of the screen and functionality" Figure 9-2 (Chp9_ScreenPanelControl.ppt) changed, the screen selection for S7-CPU diagnostics removed

Section 9.3 "S7-CPU diagnostics" screen

deleted

Section 9.3 (previously 9.4) "Status Variable" screen

Section 9.3.1 "Layout of the screen and functionality" Figure 9-3 (previously 9.4) (Chp9_ScreenS7VarStatus.ppt) changed, the screen selection for S7-CPU diagnostics removed

Chapter A Appendix

Appendix A.3 Structure of the DB_HMILITE_DATA data block

Updated to Version 5.0

Appendix A.4 Structure of the DB_HMILITE_CONFIG data block

Updated to Version 5.0 (only the title, contents same as V4.0)

Appendix A.5 Structure of the DB_HMILITE_DEVICE_DIAG data block

Updated to Version 5.0
Appendix A.6.4 Changes from 05/2005 edition to 03/2007 edition
New

A.6.5 Changes from 03/2007 edition to 08/2007 edition

Chapter 2 Installation

Section 2.6 PLC program blocks

Section 2.6.1 “PLC blocks from HMI Lite“ in Table 2-2 extended by DP history blocks (FC & DB)
Section 2.6.2 “Schema for calling the function blocks“ in Figure 2-2, entry FC_HMILITE_DP_HISTORY added

Chapter 3 Global Settings and Functionality

Section 3.7 Identification of the selected screen

Table 3-2 extended by the new screen entry SS_45_ProfibusDiagnosticHistory (screen number 45)

Chapter 5 Manual Operation

Section 5.4 Configuring

Section 5.4.3 “Grouping of movement lines in the manual operation screen“, Figure 5-7 (Chp5_KeyConfig.jpg) updated

Chapter 7 Diagnosis

Section 7.3 "Version" screen

Figure 7-4 (Chp7_ScreenVersion.ppt) extended by the blocks FC_HMILITE_DP_HISTORY (FC172) and DP_HMILITE_DP_HISTORY (DB172)

Chapter 8 Hardware Diagnostics

Section 8.1 "PROFIBUS" screen

Section 8.1.1 Figure 8-1 (Chp8_ScreenProfibus.ppt) updated
Note regarding the display > first faulty slave (Chp8_Failed_slave.JPG) integrated
Section 8.1.4 “Detailed diagnostics“, Figure 8-6 (Chp8_ScreenProfibusDetail.ppt) updated
New: Section 8.1.5 “DP history“, new section added to latching PROFIBUS diagnostics
Section 8.1.6 (formerly 8.1.5), Section 8.1.7 (formerly 8.1.6) und Section 8.1.8 (formerly 8.1.7), numbers shifted due to new section entry for the DP history (8.1.5)
New: Section 8.1.9 “Runtime interface (FC_HMILITE_DP_HISTORY)“

Chapter A Appendix

Section A.3 Structure of the DB_HMILITE_DATA data block

Block version IDs for the latching PROFIBUS diagnostics have been updated

Section A.5 Structure of the DB_HMILITE_DEVICE_DIAG data block

Updated by the additional functions for the latching PROFIBUS diagnostics (DP history) regarding BILD_PROFIBUS
Section A.6.5 Changes from 03/2007 edition to 08/2007 edition
Newly created

A.6.6 Changes from 08/2007 edition to 2009 edition

Chapter 1 General

Section 1.2 Screens provided
Extended by the hardware diagnostics screens for Profinet and Sinamics

Section 1.5.1 Configuring and programming software / licenses
Updated to Version 5.1 and order number changed.

Chapter 2 Installation

Section 2.1 Dearchiving of the source project
Updated to Version 5.1

Section 2.6 PLC programming blocks
Updated to Version 5.1
Blocks for Profinet and Sinamics diagnosis entered in the table and call-up scheme

Chapter 3 Global settings and functionality

Section 3.2 Menu structure
Menu structure extended by the hardware diagnostics screens for Profinet and Sinamics

Section 3.7 Identification of the screen selected
Table 3-2 extended by the SINAMICS and PROFINET diagnostics screens

Chapter 7 Diagnosis

Section 7.3 "Version" screen
Figure 7-4 (Chp7_ScreenVersion.ppt) extended by the block FB_HMILITE_SINAMICSCU3x0 (FB461) and block numbers adapted.

Chapter 8 Hardware diagnosis

Section 8.3 "RF300" screen
Term Moby replaced by RF300.

Section 8.4 "ASIsafeMonitor" screen
Selection text list names changed.

Section 8.5 "SINAMICS" screen
Newly created.

Section 8.6 "PROFINET / PROFIBUS diagnosis" screen
Newly created.

Chapter A Appendix

Section A.3 Structure of the data block DB_HMILITE_DATA
Section version IDs updated for the blocks for Sinamics diagnosis. alarm “SINAMICS_BLOCK_DISABLE“ added.

Section A.5 Structure of the data block DB_HMILITE_DEVICE_DIAG
Extended by the section SINAMICS
Section A.6.6 Changes from 08/2007 edition to 2009 edition
Newly created

A.6.7 Changes from edition 2009 to edition 2011

Chapter 1 General

Section 1.1 Product overview
In screen “Chp1_System_overview” link changed from PROFIBUS to PROFINET

Section 1.2 Screens offered
Extended by the machine overview in the machining area

Subsection 1.5.1 Configuration and programming software / licenses
Version of configuration and programming software changed to current versions, plus order number for ProAgent license adapted.

Chapter 2 Installation

Section 2.6 PLC program blocks
Subsection 2.6.1 “PLC blocks from HMI Lite” in Table 2-2, FB 465 (DB 465) renamed to FB_SIEM_PNIODiag with the comment that blocks cover PROFINET & PROFIBUS IO Diagnostics
Subsection 2.6.2 “Scheme for calling function blocks” in Fig. 2-2 slightly adjusted

Chapter 3 Global Settings and Functionality

Section 3.7 Identification of the selected screen
Table 3-2 extended by another screen of PROFINET Diagnostics, plus note that PROFIBUS Diagnostics screens in the 10" variant are no longer active

Chapter 6 Production Data Screens

Subsection 6.2.1 Screen layout and functionality
Note supplemented for implementing reduced display functions in the workpiece counter screen for 6" devices.

Chapter 8 Hardware Diagnosis

Overview screen of the HW diagnosis (Chp8_ScreenHWDiagnose.ppt) inserted with short text for explanation together with status display of PROFIBUS and PROFINET from the overview already.

Section 8.1 "PROFINET / PROFIBUS diagnosis" screen

Previously Chapter 8.6

Fully revised, new screens, extended texts
Screens inserted or changed (Chp8_ScreenPNIODiagUebersicht.ppt, Chp8_ScreenPNIODiagLegende.ppt, Chp8_ScreenPNIODiagDetail.ppt, Chp8_ScreenPNIODiagLeitungsdiagnose.ppt, Chp8_ScreenPNIODiagTrigger.ppt, Chp8_ScreenPNIODiagInfo.ppt), explanations described in screen shots (Chp8_PNIODiagZustaende.ppt, Chp8_PNIODiagTNInformation.ppt, Chp8_PNIODiagDetaildiagnose.ppt) and configuration instructions added with note on the network ID (Chp8_HWKonfigMastersystem).

Subsection 8.2.5 Runtime interface (FC_HMILITE_DRIVE)

Formal review

Section 8.3 Screen "RF300"

Subsection 8.3.1 “Layout of screen and functionality” Fig. 8-17 (Chp8_ScreenRF300.ppt) changed

Subsection 8.3.4 Runtime interface (FC_HMILITE_RF300)

Formal review

Subsection 8.4.3 Runtime interface (FC_HMILITE_ASI)

Formal review

Subsection 8.4.6 Runtime interface (FC_HMILITE_ASIMON2D)

Formal review

Subsection 8.5.5 Runtime interface (FC_HMILITE_SINAMICSCU3x0)

Formal review

Section 8.6 "PROFIBUS" screen

Previously Section 8.1
Note added that screens can be used only in the 6" variant. In the 10" variant, the diagnosis is covered through the new PNIODiag

Chapter A Appendix

Subsection A.6.7 Changes from the 01/2009 Edition to the 09/2011 Edition

Newly created

To
Siemens AG

TRANSLINE Support
P. O. Box 106026

D-70049 Stuttgart

Tel. +49 (0) 711 / 137 - 3964

Fax +49 (0) 711 / 137 - 2838

Email: transline-support.ad.sdw.rd@siemens.com

Suggestions

Corrections

For Publication/Manual:

Solutions for Powertrain

TRANSLINE - Visualization Operation Diagnostics HMI Lite

Manufacturer Documentation

From

Name:

Company/Dept.

Address: _____

_____

Telephone: _____

Telefax: _____

Manual

Available at:
I IA&DT-E-Business Workplace
2011 Edition

Should you come across any printing errors when reading this publication, please notify us on this sheet. Suggestions for improvements are also welcome.

Suggestions and/or corrections

Siemens AG
Industry Sector
Drive Technologies, MC MT
Postfach 3180
91050 ERLANGEN
GERMANY

Reserved for changes
Available at
I IA&DT E-Business Workplace
Printed in Germany

www.workplace.automation.siemens.com/