

Manual

Visualization, Operation, Diagnostics
HMI Lite

Edition 2016



Solutions for Powertrain

TRANSLINE

SIEMENS

Solutions for Powertrain

TRANSLINE - Visualization, Operation, Diagnostics HMI Lite

Manual

Valid for:
HMI Lite V8.0 and SIMATIC S7-1500

General Information	1
Installation	2
Global Settings and Functionality	3
Procedure to be applied when creating new screens	4
Header and Operator Information	5
Manual Operating	6
Productions Data Screens	7
Diagnostics	8
Hardware Diagnostics	9
System Screens	10
Appendix	A

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
2016	DF MC – E-Business Workplace	A

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Contents

1 General Information	1-13
1.1 Product overview	1-13
1.2 Provided screens	1-14
1.3 Basic knowledge	1-15
1.4 Hardware requirements	1-16
1.5 Software requirements	1-18
1.5.1 Configuration and Programming Software/Licenses	1-18
1.5.2 Runtime software/licenses	1-18
2 Installation	2-19
2.1 HMI Lite TIA-project	2-19
2.2 Licensing	2-21
2.3 Creating an HMI Lite project	2-25
2.4 Requirements in the CPU Properties	2-27
2.5 Direct keys option	2-28
2.6 PLC program blocks	2-29
2.6.1 HMI Lite Standard blocks	2-29
2.6.2 Schema for calling the function blocks	2-30
2.7 Operator panels	2-31
2.8 Working with the data blocks	2-33
3 Global Settings and Functionality	3-35
3.1 Layout of the screens and basic screen elements	3-35
3.2 Menu structure	3-37
3.3 Clock memory byte of the control	3-39
3.4 PLC system time	3-40
3.4.1 System timer	3-40
3.4.2 System time and date	3-40
3.5 Job mailbox HMI Lite	3-41

3.6 LTLL_Basic block.....	3-43
4 Procedure to be applied when creating new screens.....	4-45
4.1 The Template screen.....	4-45
4.2 Designation conventions.....	4-46
4.3 Identification of the selected screen	4-48
4.4 Style elements	4-51
5 Header and Operator Information	5-53
5.1 Header	5-53
5.1.1 Layout of the header.....	5-53
5.1.2 Display of current operating mode	5-54
5.1.3 Status display	5-55
5.1.4 Display of the initial state	5-56
5.1.5 Text boxes	5-57
5.1.6 Sign-of-life of the PLC.....	5-58
5.1.7 Changing the display of the status signals in the header	5-58
5.2 Operator information.....	5-59
6 Manual Operating.....	6-61
6.1 Overview	6-61
6.1.1 Layout and basic functionality of the manual operating screens	6-61
6.1.2 Elements of the movement/function line	6-64
6.1.3 Assignment of the function numbers	6-66
6.2 Purpose of the individual manual operating screens.....	6-68
6.2.1 Setup screen.....	6-68
6.2.2 Power up condition	6-68
6.2.3 Selecting/deselecting units	6-68
6.2.4 Selecting/deselecting nut runners	6-68
6.2.5 Selecting/deselecting nut driver groups	6-69
6.2.6 Selecting/deselecting cycle type	6-69
6.2.7 User operating screen	6-69
6.3 Configuration and runtime interface.....	6-70
6.4 Configuration.....	6-71
6.4.1 Global configurations.....	6-71
6.4.2 Number of movement/function lines	6-72
6.4.3 Grouping of the movement lines in the setup screen	6-73
6.4.4 Hiding elements of the function line	6-74
6.4.5 Display text	6-77
6.5 Runtime interface.....	6-80
6.5.1 Interface for information about the binary state.....	6-80

6.6 Control interface.....	6-83
6.6.1 Job mailbox.....	6-83
6.6.2 Binary control interface.....	6-85
6.7 LTLL_Manual block.....	6-86
6.8 LTLL_ManualControl block.....	6-89
6.9 LTLL_ManualGraph block	6-91
7 Production Data Screens	7-97
7.1 Cycle times screen.....	7-97
7.1.1 Layout and functionality.....	7-97
7.1.2 Runtime interface (LTLL_Cycletime)	7-99
7.1.3 Configuration	7-101
7.2 Workpiece counter screen	7-102
7.2.1 Layout and functionality.....	7-102
7.2.2 Runtime interface (LTLL_Counter)	7-106
7.2.3 Configuration	7-109
8 Diagnostics.....	8-113
8.1 Alarm screen and Alarm buffer screen	8-113
8.1.1 Layout and functionality.....	8-113
8.1.2 Runtime interface	8-114
8.1.3 Configuration	8-114
8.2 Interface screen	8-116
8.2.1 Layout and functionality.....	8-116
8.2.2 Runtime interface	8-117
8.2.3 Configuration	8-118
9 Hardware Diagnostics	9-119
9.1 System diagnostics.....	9-121
9.2 Webserver.....	9-122
9.3 SINAMICS diagnostics	9-124
9.3.1 SINAMICS status screen.....	9-124
9.3.2 SINAMICS alarms screen.....	9-125
9.3.3 SINAMICS position screen.....	9-126
9.3.4 SINAMICS Safety Integrated Status screen.....	9-127
9.3.5 Configuring of the WinCC screens	9-128
9.3.6 Configure the drive object (LTLL_SinamicsCFG).....	9-129
9.3.7 Runtime interface (LTLL_Sinamics)	9-129
9.4 Motor starter diagnostics	9-131
9.4.1 Runtime interface (LTLL_Motorstarter)	9-132
9.5 RFID diagnostics.....	9-135
9.5.1 Layout and functionality.....	9-135

9.5.2 Supported ident devices	9-136
9.5.3 Configuring the WinCC screens	9-136
9.5.4 Runtime interface (LTLL_RFID)	9-137
9.6 Safety screen	9-139
10 System Screens	10-141
10.1 Version screen	10-141
10.2 Panel Control screen	10-142
10.3 System screen	10-144
10.4 PLC system data screen.....	10-146
A Appendix	A-147
A.1 Abbreviations.....	A-147
A.2 Change log	A-148
A.2.1 Edition 2016.....	A-148

Tables

Table 1-1	Supported operator panels – tested and approved	1-16
Table 1-2	Supported operator panels - compatible without restrictions	1-16
Table 1-3	Supported operator panels - compatible with restrictions	1-17
Table 2-2	Overview of the HMI Lite blocks	2-29
Table 3-1	Button styles	3-38
Table 3-2	Structure of the job mailbox	3-41
Table 3-3	Description of the LTLL_Basic parameters	3-43
Table 4-1	Designation of the convention syntax for the WinCC screen elements	4-46
Table 4-2	Identification code of the individual screens	4-49
Table 5-1	Display of the operating modes in the header	5-54
Table 5-2	Display of operating modes (selected, active/not active)	5-54
Table 5-3	Display of the current operating mode – interface bits	5-55
Table 5-4	Machine status display	5-55
Table 5-5	Status display – interface bits	5-56
Table 5-6	Initial state display – possible states	5-56
Table 5-7	Initial state display – interface bits	5-56
Table 5-8	Display of the sign-of-life of the PLC	5-58
Table 6-1	Manual operating screens – structure of the configuration interface	6-70
Table 6-2	Manual operating screens – structure of the text lists	6-77
Table 6-3	Manual operating screens – display texts example	6-78
Table 6-4	Manual operating screens – text list example	6-79
Table 6-5	Operating screens - code for identifying the screen in the job mailbox	6-84
Table 6-6	Parameters of the LTLL_Manual block	6-87
Table 6-7	Parameters of the LTLL_ManualControl block	6-90
Table 6-8	Parameters of the LTLL_ManualGraph block	6-92
Table 7-1	Parameters of the LTLL_Cycletime block	7-100
Table 7-2	Time parameters of the LTLL_Counter block	7-107
Table 7-3	WinCC text list SO_04_011_PartCounterType	7-110
Table 8-1	Selection window for the interlocks – screen caption of the text list	8-118
Table 8-2	Designation of the input/outputs	8-118
Table 9-1	Text list for the axis designations	9-127
Table 9-2	Drive object structure in data block LTLL_SinamicsCFG	9-128
Table 9-3	Parameters of the LTLL_Sinamics block	9-129
Table 9-4	Parameters of the LTLL_Motorstarter block	9-132
Table 9-5	Text list for the designations of the motor starters	9-133
Table 9-6	Text list for the designations of the ident devices	9-135
Table 9-7	Parameter of the LTLL_RFID block	9-137
Table 9-8	Parameter of the LTLL_Safety block	9-139

Figures

Fig. 1-2	Supported operator panels	1-16
Fig. 2-1	Structure of a HMI Lite project	2-20
Fig. 2-2	Dialog for generating a license key	2-21
Fig. 2-3	Identify product: Entering the data	2-22
Fig. 2-4	Identify product: Already assigned licenses	2-22
Fig. 2-5	Select licenses (example for 2 operator panels)	2-22
Fig. 2-6	Generate License Key	2-23
Fig. 2-7	Entering the license keys into LTLL_Config	2-24
Fig. 2-8	Area pointers	2-25
Fig. 2-9	Call interface of the LTLL_Basic block	2-27
Fig. 2-10	Field in the LTLL_Data block	2-31
Fig. 2-11	Field in the LTLL_Config block	2-31
Fig. 3-2	Structure of the menu for operator panels with 12,1" screen display (TP1200 Comfort)	3-37
Fig. 3-3	Structure of the menu for operator panels with 9" screen display (KTP900F Mobile)	3-37
Fig. 3-4	Layout of the Coordination area pointer	3-42
Fig. 3-5	Interface of the LTLL_Basic block	3-43
Fig. 4-2	Configuring the screen event to identify the selected screen	4-48
Fig. 4-3	Style elements	4-51
Fig. 5-2	Assignment of the 2nd text list SO_00_000_HeaderTextlist_2	5-57
Fig. 6-2	Manual operation – Selection/actuation of a function line	6-62
Fig. 6-3	Manual operating screens - absolute and symbolic view	6-63
Fig. 6-4	Manual operating screens - elements of the individual line	6-64
Fig. 6-5	Manual operating screens – assignment of the function numbers	6-66
Fig. 6-6	WinCC configuration of the screen selection of the setup screen in groups	6-73
Fig. 6-7	Manual operating screen – hiding screen elements	6-74
Fig. 6-8	Manual operating screens – text lists	6-77
Fig. 6-9	Manual operating screens – example for the configuration of a text	6-79
Fig. 6-10	Manual operating screens – dynamic movement elements	6-80
Fig. 6-11	Call interface of LTLL_Manual block	6-86
Fig. 6-12	Call interface of LTLL_ManualControl block	6-89
Fig. 6-13	Call interface of LTLL_ManualGraph block	6-91
Fig. 6-14	Adding a variable of the type LTLL_ManualGraphExt	6-95
Fig. 6-15	Data transfer and call of the instruction GetInstanceName	6-96
Fig. 6-16	Data transfer to LTLL_ManualGraphControl. call[x].output	6-96
Fig. 7-2	Call interface of the LTLL_Cycletime block	7-99
Fig. 7-3	Workpiece counter screen	7-102
Fig. 7-4	Workpiece counter – procedure for confirming the reset	7-104
Fig. 7-5	Call interface of the LTLL_Counter block	7-106
Fig. 8-2	Integrating the PLC code display	8-114
Fig. 8-3	PLCCodeViewer activation	8-115
Fig. 8-4	Setting for the alarm display object in the Alarm buffer screen	8-115
Fig. 8-5	Interface screen SS_04_031_Interlocks	8-116
Fig. 9-2:	System diagnostics screen	9-120
Fig. 9-3	WebServer screen	9-121
Fig. 9-4	Webserver: Changing URL	9-122
Fig. 9-5	SINAMICS status	9-123
Fig. 9-6	SINAMICS alarms	9-124
Fig. 9-7	SINAMICS position	9-125
Fig. 9-8	SINAMICS Safety Integrated Status	9-126
Fig. 9-9	How to assign a text list to a drive object	9-127

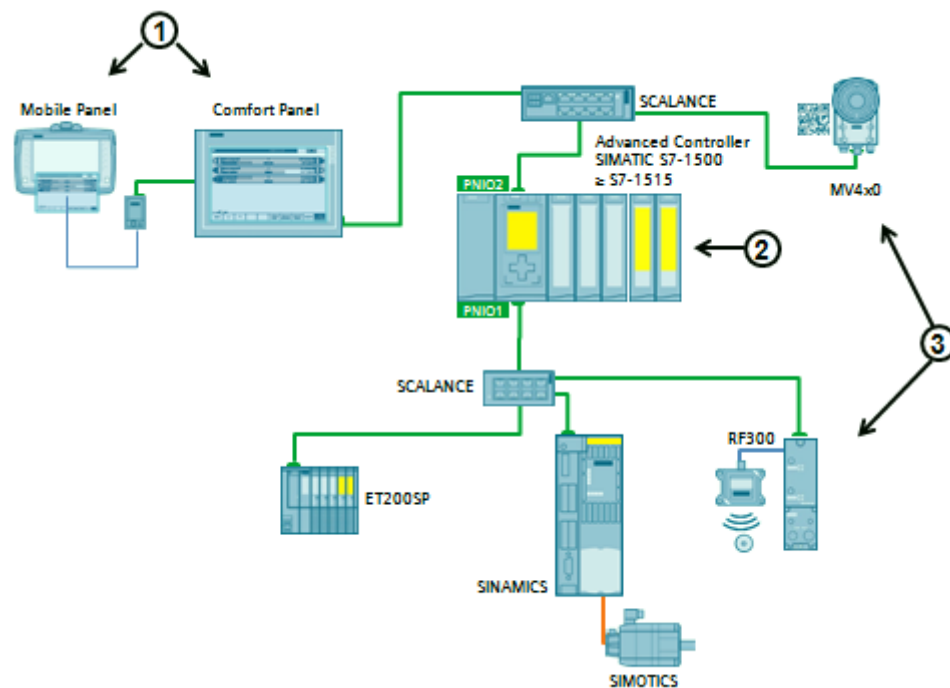
Fig. 9-10	Call interface of the LTLL_Sinamics blocks.....	9-128
Fig. 9-11	Motor starter: Control signals, motor protection, status word	9-130
Fig. 9-12	Motor starter: Measured values, statistical data	9-130
Fig. 9-13	Motor starter: Example for logbook: Device error logbook.....	9-131
Fig. 9-14	Call interface of the LTLL_Motorstarter block.....	9-131
Fig. 9-15	RFID diagnostics	9-134
Fig. 9-16	Call interface of the LTLL_RFID block.....	9-136
Fig. 9-17	Safety screen	9-138
Fig. 9-18	Call interface of the LTLL_Safety block	9-139
Fig. 10-2	Panel Control screen	10-142
Fig. 10-3	System screen.....	10-144
Fig. 10-4	PLC system data screen	10-146

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 Comfort - and Mobile Panels of the 2nd Generation from the SIMATIC product series and PLC blocks for supplying the screen forms. The navigation within the individual screens 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 panels for displaying the HMI Lite screens
- (2) SIMATIC S7-1500 with the PLC program for supplying the screens
- (3) External devices and I/O peripherals

Fig. 1-1 System overview

1.2 Provided screens

Main menu: Preparation and setup

- Setup screen
- 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
- Interlocks

Main menu: Diagnostics

- Messages
- Alarm Log
- Interface

Main menu: Hardware diagnostics

- System Diagnostics
- Webserver
- SINAMICS
- Motor starter
- RFID
- Safety

Main menu: System

- Version
- Panel Control
- System
- PLC System data

1.3 Basic knowledge

To commission the HMI Lite system, the following knowledge is required:

Visualization WinCC (TIA)

- TIA Portal visualization (WinCC)
- 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 objects
- Testing the HMI configurations
- Working with the project library

Programming STEP 7 (TIA)

- Programming STEP 7
- 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
- Working with the project library

1.4 Hardware requirements

Control

The following minimum control versions are required:
S7-1500 as of firmware version V2.0

Operating devices - tested and approved

HMI Lite is available for the SIMATIC HMI Comfort panels listed in Table 1-1.

Table 1-1 Supported operator panels – tested and approved

Description	Type of the installed screen	Touchscreen/keys
SIMATIC HMI KTP900F Mobile	9" TFT Wide-screen display, 800x480 Pixel resolution	Touch screen and 10 tactile function keys, including LEDs
SIMATIC HMI TP1200 Comfort	12,1" TFT Wide-screen display, 1280x800 Pixel resolution	Touch screen

SIMATIC HMI KTP900F Mobile



SIMATIC HMI TP1200 Comfort



Fig. 1-2 Supported operator panels

Operating devices – compatible without restrictions

The SIMATIC HMI Panels listed in the following table are compatible and HMI Lite is executable without restrictions.

Table 1-2 Supported operator panels - compatible without restrictions

Description	Type of the installed screen	Touchscreen/keys
SIMATIC HMI KTP900 Mobile	9" TFT Wide screen display, 800x480 Pixel resolution	Touch screen and 10 tactile function keys, including LEDs
SIMATIC HMI TP1500 Comfort	15,4" TFT Wide screen display, 1280x800 Pixel resolution	Touch screen

Operating devices – compatible with restrictions

The SIMATIC HMI Panels listed in the following table are compatible and HMI Lite is executable with restrictions.

Table 1-3 Supported operator panels - compatible with restrictions

Description	Type of the installed screen	Touchscreen/keys
SIMATIC HMI KP700 Comfort	7" TFT Wide screen display, 800x480 Pixel resolution	Membrane keyboard with 24 function keys and system keyboard
SIMATIC HMI KP900 Comfort	9" TFT Wide screen display, 800x480 Pixel resolution	Membrane keyboard with 26 function keys and system keyboard
SIMATIC HMI KP1200 Comfort	12,1" TFT Wide screen display, 1280x800 Pixel resolution	Membrane keyboard with 34 function keys and system keyboard
SIMATIC HMI KP1500 Comfort	15,4" TFT Wide screen display, 1280x800 Pixel resolution	Membrane keyboard with 36 function keys and system keyboard
SIMATIC HMI TP700 Comfort	7" TFT Wide screen display, 800x480 Pixel resolution	Touch screen
SIMATIC HMI TP900 Comfort	9" TFT Wide screen display 800x480 Pixel resolution	Touch screen
SIMATIC HMI TP1900 Comfort	18,5" TFT Wide screen display, 1366x768 Pixel resolution	Touch screen
SIMATIC HMI TP2200 Comfort	21,5" TFT Wide screen display, 1920x1080 Pixel resolution	Touch screen
SIMATIC HMI KTP700 Mobile	7" TFT Wide screen display, 800x480 Pixel resolution	Touch screen and 8 tactile function keys, including LEDs
SIMATIC HMI KTP700F Mobile	7" TFT Wide screen display, 800x480 Pixel resolution	Touch screen and 8 tactile function keys, including LEDs

Note

If you want to use operator panels from the **Compatible with restrictions** table, use the **Change operator panel type** function in the TIA Portal. Note that the resolution of your operator panel must not be less than the original station. The HMI Lite screens are somewhat smaller than the screen of the respective panel and there is therefore an empty space on the edge of the screen. If you adjust the resolution of the screens, check that all elements are displayed correctly.

1.5 Software requirements

1.5.1 Configuration and Programming Software/Licenses

Mandatory

Description	Version
TRANSLINE HMI Lite	8.0

Description	Version
TIA STEP 7 Professional	V14
TIA WinCC Engineering Software Comfort/Advanced/Professional	V14

The S7-GRAPH programming language can be used to graphically program machine sequences.
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://support.automation.siemens.com>

1.5.2 Runtime software/licenses

Mandatory

Description	Version
HMI Lite V8.0 License one license per panel	current



2

2 Installation

2.1 HMI Lite TIA-project

A HMI Lite project consists of the STEP 7 program and a WinCC visualization. It contains per default three stations: One station for the PLC and one station each for the KTP900F Mobile and TP1200 Comfort operator panels.

PLC station

The station for the PLC contains the PLC blocks, variable tables and global constants.

KTP900F Mobile Panel station

The station for a KTP900F Mobile Panel is the base for the following panels:

- KP700 Comfort
- KP900 Comfort
- TP700 Comfort
- TP900 Comfort
- TP1900 Comfort
- KTP700 Mobile
- KTP700F Mobile
- KTP900 Mobile

The station for a TP1200 Comfort Panel is the base for the following panels:

- KP1200 Comfort
- KP1500 Comfort
- TP1500 Comfort
- TP2200 Comfort

HMI Lite Project structure

A HMI Lite project has the following structure.

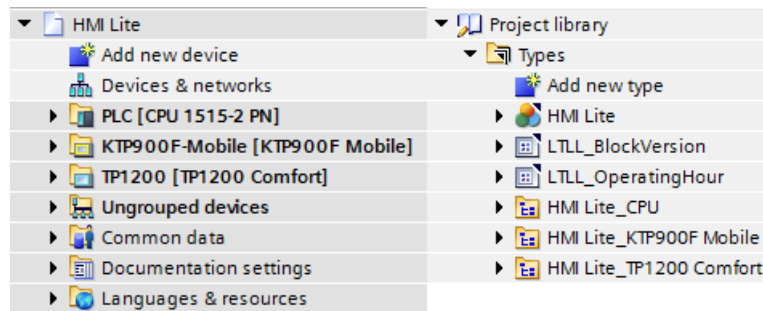


Fig. 2-1 Structure of a HMI Lite project

2.2 Licensing

The use of HMI Lite requires a license. For each panel which is used in HMI Lite, a license is required. An order of HMI Lite includes one license. You can purchase licenses for additional panels as HMI Lite Copy license (without purchasing a project).

You must generate a license key for each panel. Licensing is realized by entering the license number and the license key in the HMI Lite data block (LTLL_Config) in the TIA project.

Licensing via the internet

You must generate the license key for HMI Lite via the internet under

<http://www.siemens.com/automation/license>.

By using the **Web License Manager**, you can assign licenses to the hardware (access to the license database) in a standard web browser.

1. Log in via **Direct Access**. The following screen is displayed:

Fig. 2-2 Dialog for generating a license key

2. Enter the **license number** and the **dispatch note number**. They are imprinted on the Certificate of License you have got together with the software.

3. Press the **Next** button.

Direct access

Login Identify product Select licenses Assign licenses Generate license key

Enter the serial number of the hardware to which you want to assign licenses.

Product: HMI Lite

Version: 8.0

Hardware serial number: SMC_XXXXXXXXXX

Next

Fig. 2-3 Identify product: Entering the data

4. Select the version of HMI Lite at **Version** (V8.x).
5. Enter the serial number of the PLC's SMC card in the **Hardware serial number** field (not the serial number of the operator panel).
6. Select HMI Lite at **Product**.
7. Press the **Next** button.
If the hardware has already been assigned licenses, this is displayed.

Direct access

Login Identify product Select licenses Assign licenses Generate license key

The following licenses have been assigned to the hardware.

Product name: HMI Lite

Hardware serial number : SMC_XXXXXXXXXX

Machine identifier:

No licenses have been assigned to the hardware yet.

Next

Fig. 2-4 Identify product: Already assigned licenses

8. Press the **Next** button.
The licenses from the dispatch note which are not yet assigned are displayed.

Direct access

Login Identify product Select licenses Assign licenses Generate license key

Select the licenses that you want to assign.

Product name: HMI Lite

Hardware serial number : SMC_XXXXXXXXXX

Machine identifier: ⓘ

Current license key

Product	Quantity	Max quantity	Box	License number
HMI Lite V8.0 6FC5263-8PY11-0AG0	0	32	2	T-XXX... T-YYY... T-ZZZ...

Next

Fig. 2-5 Select licenses (example for 2 operator panels)

9. In the line in which the HMI Lite V8.0 is displayed, select the required license in the **License number** column.
If you are using several operator panels, you must select a license for each operator panel (multiple selection).
10. Press the **Next** button.
A summary of the selected licenses is displayed.
11. Control your selection.
12. Press the **Assign** button to assign the selected license(s) to the hardware.
After that, the generated license key(s) is (are) displayed. A license key includes all options assigned to the specified hardware. The assigned licenses are listed below on the screen.

Direct access

Login Identify product Select licenses Assign licenses **Generate license key**

A license key has been generated on the basis of all licenses that you have assigned to a product.

The name of the license key is:
1D ... (T-XXX...), 28 ... (T-YYY...)

Enter the license key on your target system.

The license key can be stored in a file:

- [Save the license key in keys.txt](#)
- [Save the license report in PDF format](#)

The following licenses are assigned to the hardware HMI Lite with the serial number SMC_XXXXXXXXXX

Order No. (MLFB)	Assignment date	License number	Product	License Key
6FC5263-8PY11-0AG0	13.12.2016	T-XXX...	HMI Lite V8.0	28 ...
6FC5263-8PY11-0AG0	13.12.2016	T-YYY...	HMI Lite V8.0	1D ...

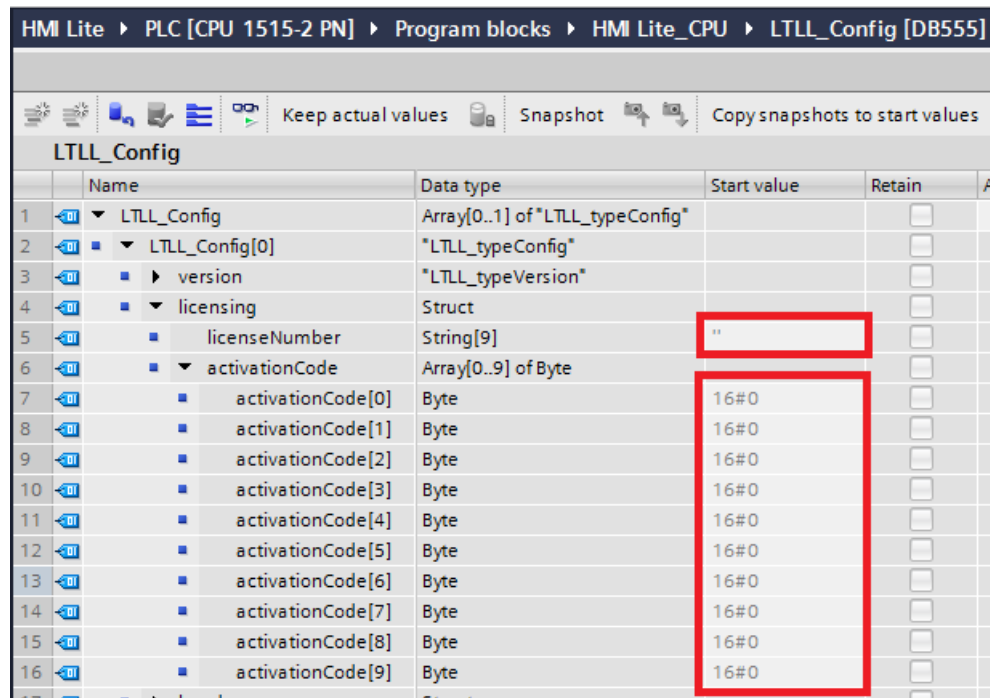
Fig. 2-6 Generate License Key

13. Save the license key in a file via right-clicking the mouse on
Save the license key in keys.txt
or
Save the license report in PDF format.

Enter license number and license key in the HMI Lite project

Enter the license numbers and the corresponding license keys in the **LTLL_Config** block at the following position:

- License number: **LTLL_Config.THIS[X].licensing.licenseNumber**
- License key: **LTLL_Config.THIS[X].licensing.licenseKey**



HMI Lite ▸ PLC [CPU 1515-2 PN] ▸ Program blocks ▸ HMI Lite_CPU ▸ LTLL_Config [DB555]					
LTLL_Config					
	Name	Data type	Start value	Retain	A
1	LTLL_Config	Array[0..1] of "LTLL_typeConfig"		☐	
2	LTLL_Config[0]	"LTLL_typeConfig"		☐	
3	version	"LTLL_typeVersion"		☐	
4	licensing	Struct		☐	
5	licenseNumber	String[9]		☐	
6	activationCode	Array[0..9] of Byte		☐	
7	activationCode[0]	Byte	16#0	☐	
8	activationCode[1]	Byte	16#0	☐	
9	activationCode[2]	Byte	16#0	☐	
10	activationCode[3]	Byte	16#0	☐	
11	activationCode[4]	Byte	16#0	☐	
12	activationCode[5]	Byte	16#0	☐	
13	activationCode[6]	Byte	16#0	☐	
14	activationCode[7]	Byte	16#0	☐	
15	activationCode[8]	Byte	16#0	☐	
16	activationCode[9]	Byte	16#0	☐	

Fig. 2-7 Entering the license keys into LTLL_Config

2.3 Creating an HMI Lite project

As basis for an HMI Lite project, use the HMI Lite project from the HMI Lite V8.0 Project DVD.

Integrating a user project in HMI Lite

If you want to integrate your user program in the HMI Lite project, proceed as follows in the TIA Portal:

1. Change the device type of the PLC to the required S7-CPU.
2. Copy the blocks, variable tables, PLC data types and the other STEP 7 objects from the user program to the HMI Lite PLC station.
3. Copy the screens, HMI variables, text and graphic lists as well as the other WinCC objects from the user program to the appropriate panel station.
4. Check the area pointers of your panel stations.

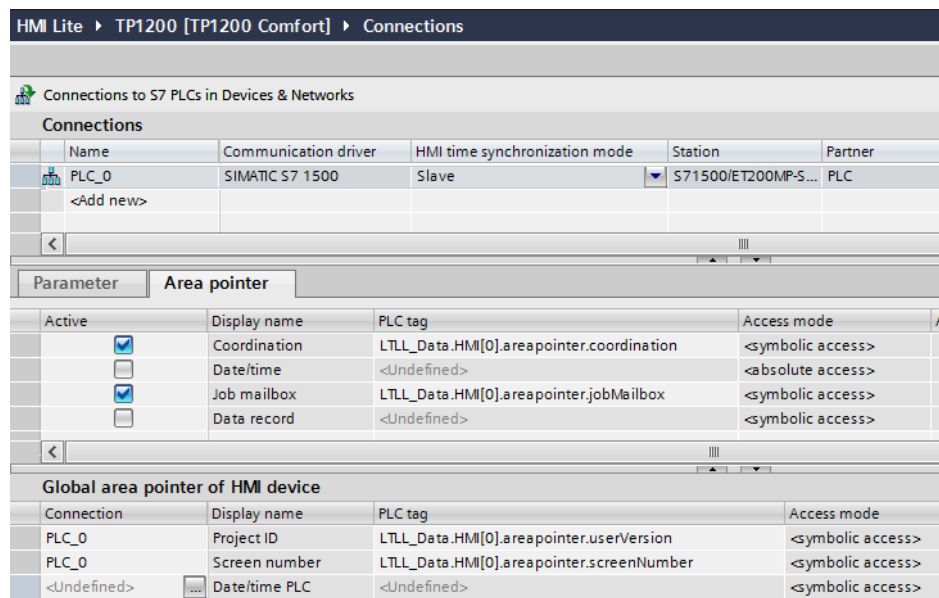


Fig. 2-8 Area pointers

Restrictions

- The HMI Lite blocks are encrypted and therefore cannot be simulated with PLCSIM.
- The encrypted HMI Lite blocks must not be changed.
- The HMI Lite PLC data types must not be changed (**LTLL_type**....). A change may result in that you can no longer compile the HMI Lite blocks.
- The HMI Lite function blocks (FBs) and PLC data types are typified in the project library. The connection to the type must not be cancelled as otherwise the update of the HMI Lite objects cannot be guaranteed.
- Screens with the SS... identifier must not be changed.
- The HMI Lite faceplates must not be changed.
- HMI variables with the SS_... identifier must not be changed.
Exceptions are:
SS_02_001_setupScreenNumberOfLastPage
SS_02_001_setupScreenNumberOfFirstPage
See Section 6.4.3.
- Text and graphic lists with the SS_... identifier must not be changed.
- The connection of the screens to the types in the project library must not be cancelled as otherwise the update of the HMI Lite screens cannot be guaranteed.
- Do not change the directory structure of the HMI Lite objects.

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.

The clock memory byte is an input variable of the **LTLL_Basic** block.

Call interface

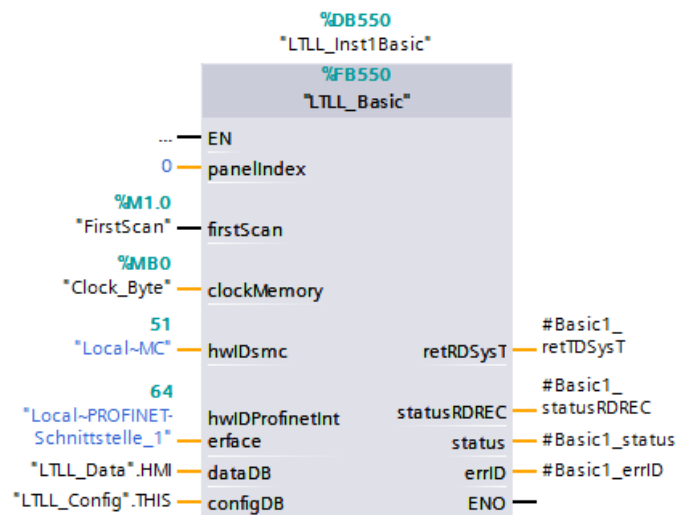


Fig. 2-9 Call interface of the **LTLL_Basic** block

2.5 Direct keys option

For safety reasons, the direct keys of the operating unit should be used for the manual functions.

The direct keys functionality is available in the block **LTLL_Manual**.

The input word of the direct keys should be assigned to the input variable **keyButton** in **LTLL_Manual**.

Note

Further notes for configuring the direct keys function are contained in the online help of the TIA Portal.

Further notes for configuring the manual screens are contained in the "Manual Operating" chapter of this documentation.

2.6 PLC program blocks

2.6.1 HMI Lite Standard blocks

All HMI Lite standard blocks are contained in the HMI Lite project.

Table 2-1 Overview of the HMI Lite blocks

Symbolic name of the blocks	Comment
LTLL_Data	HMI Lite interface
LTLL_Config	HMI Lite configuration
LTLL_Basic	General PLC program
LTLL_Manual LTLL_ManualControl	PLC program for the operating screens
LTLL_ManualGraph LTLL_ManualGraphExt	Execution of manual functions using S7-GRAPH sequences
LTLL_ManualGraphConfig LTLL_ManualGraphControl	Configuration for the FB107
LTLL_DeviceDiag	Interface of the device diagnostics
LTLL_RFID	Moby diagnostic program code
LTLL_Counter LTLL_CounterData	Workpiece counter screen
LTLL_Cycletime	Cycle time screen
LTLL_Sinamics LTLL_SinamicsCFG	HMI Lite SINAMICS diagnostics Data block with SINAMICS objects
LTLL_Safety	HMI Lite Safety Diagnostics
LTLL_Motorstarter	HMI Lite ET200pro Motor starter diagnostics
LTLL_PLCSYSTEMData	PLC System data screen

2.6.2 Schema for calling the function blocks

LTLL_Basic **LTLL_Manual**

The **LTLL_Basic** and **LTLL_Manual** blocks must be called once cyclically for each operator panel.

LTLL_ManualGraph

The **LTLL_ManualGraph** block must be called once cyclically per step sequence for each operator screen.

More detailed information can be found in Section 6.9.

LTLL_Sinamics **LTLL_Motorstarter** **LTLL_RIFD** **LTLL_Safety** **LTLL_PLCSYSTEMData**

The blocks for the hardware diagnostics (**LTLL_Sinamics**, **LTLL_Motorstarter**, **LTLL_RIFD**, **LTLL_Safety**, **LTLL_PLCSYSTEMData**) must be called once cyclically.

LTLL_Counter **LTLL_Cycletime**

The **LTLL_Counter** and **LTLL_Cycletime** blocks must be called for each workpiece counter or for each cycle time. (1 workpiece counter = 1 call, 3 workpiece counters = 3 calls, etc.)

LTLL_Safety

The **LTLL_Safety** block must be called once cyclically if you want to use the safety diagnostics.

Important

If you do not use Safety, delete the call of the **LTLL_Safety** block and the **LTLL_Safety** block from your project (not from the project library).

2.7 Operator panels

If several HMI Lite operator panels are to be connected with a control, a new DB interface must be provided in the control for the second and each additional operator panel.

Proceed as follows if you wish to connect several operator panels to a control:

1. Expand the array in the **LTLL_Data** block by one element. (The array size mirrors the number of operator panels on a control system.)

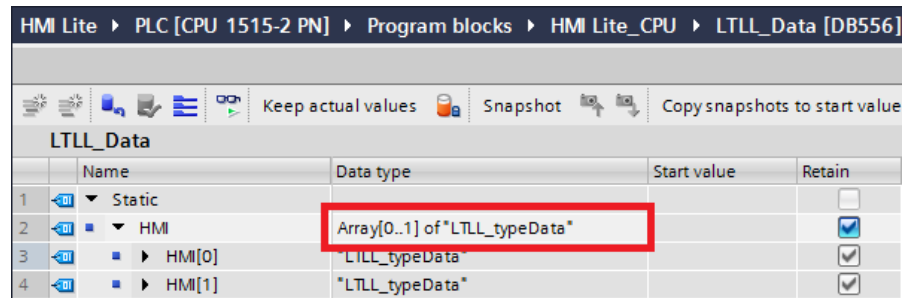


Fig. 2-10 Field in the **LTLL_Data** block

2. Expand the array in the **LTLL_Config** block by one element. (The array size mirrors the number of operator panels on a control system.)

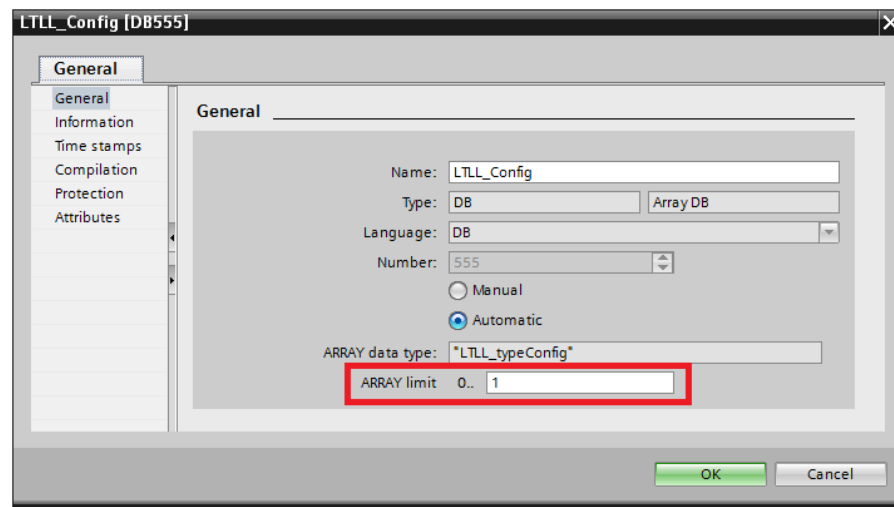


Fig. 2-11 Field in the **LTLL_Config** block

3. The **LTLL_Basic** block must be called once in the program for each operator panel. The **dataDB** and **configDB** input parameters must be supplied with the appropriate array elements of the data blocks.
4. The **LTLL_Manual** block must be called once in the program for each operator panel. The **dataDB** and **configDB** input parameters must be supplied with the appropriate array elements of the data blocks.

5. The array index in the data blocks must be entered in the **SO_00_000_index** HMI variable as start value in the basic settings.
6. The area pointers of the operator panel must be adjusted to the corresponding data areas in the DBs.
7. The user-specific fault and operating messages must be assigned new addresses, unless the same messages should be displayed on both operator panels.

Auf die Hardware-Diagnose kann zu einem Zeitpunkt nur von einem Bediengerät aus zugegriffen werden. Deshalb muss hierfür eine Bediengeräte-Umwahl projektiert werden.



Important

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

2.8 Working with the data blocks

The two **LTLL_Data** und **LTLL_Config** data blocks form the interfaces between the HMI screens and the PLC program.

In contrast to the **LTLL_Data** data block, the **LTLL_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 in the **LTLL_Config**.

Procedure for the configuring

Adapt the number of arrays in the **LTLL_Data** and **LTLL_Config** data blocks to the number of operator panels that you are using (1 panel = array[0..0], 2 panels = Array [0..1], etc.).

The following describes a simple option for entering the configuration data. This is recommended as the correct values are taken over after a CPU restart and a load is not placed on the cycle time.

To edit the **LTLL_Config** data block:

1. Open the **LTLL_Config** data block in the TIA Portal.
2. Enter the start values corresponding to the machine.
3. Take over the start values as actual values with the **Load start values as actual values** command.
4. Load the data block to the control system.
5. Test the changed configuration.
8. The array index in the data blocks must be entered in the **SO_00_000_index** HMI variable as start value in the basic settings.
9. The area pointers of the operator panel must be adjusted to the corresponding data areas in the DBs.
10. The user-specific fault and operating messages must be assigned new addresses, unless the same messages should be displayed on both operator panels.

Note

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



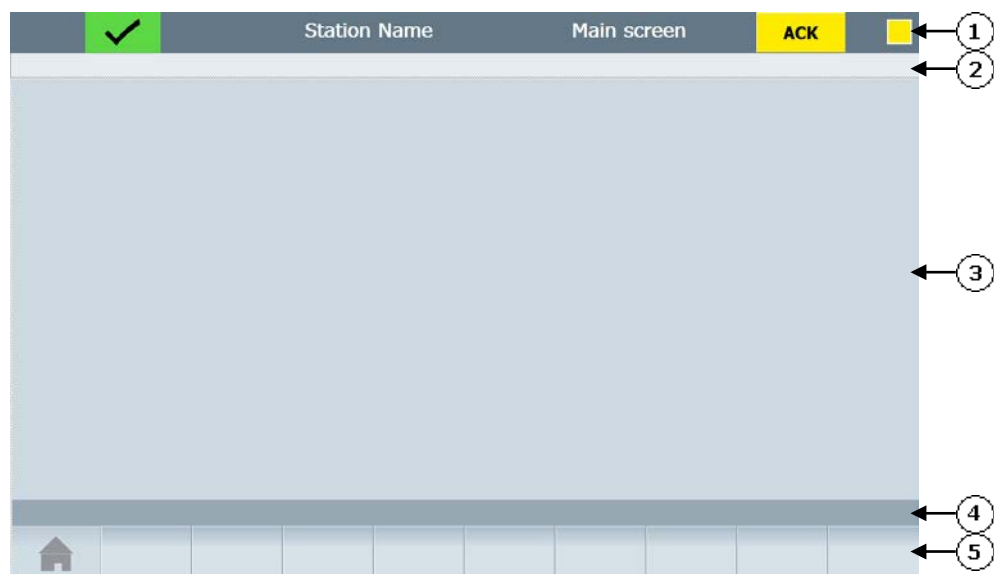
Notes

3

3 Global Settings and Functionality

3.1 Layout of the screens and basic screen elements

All screens have a standard structure.



- (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

Fig. 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. You find further details about the header in Chapter 5 "Header and Operator Information" 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 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. for resetting a workpiece counter).

By default, the menu structure is based on a two-level structure of main menu and submenu level. A third menu level is used if a grouping of inter-related screen forms is required. A third level is also existing for hardware diagnostics.

3.2 Menu structure

The menu structure of the HMI Lite standard project must not be customized.
The menu structure differs according to the size of the operator panels:

- Operator panels with 12,1" screen (TP1200 Comfort)
- Operator panels with 9" screen (KTP900F Mobile)

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

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

	OEM	Prepare	Manual	Machine	diagnostics	Hardware Diagnostics	Process		System	Main menu
	Power Up Condition	Cycle Types	Select Units			Symbol / Absolute	Previous page	Next page	Back	Submenu Prepare
	Manual	Nut runner	Nut runner group	User Defined		Symbol / Absolute	Previous page	Next page	Back	Submenu Manual
	Machine Overview	Part Count	Cycle Time						Back	Submenu Machine
	Alarm	alarm buffer	Interlock				Activate PLC code viewer	Acknowledge message	Back	Submenu Diagnostics
	Hardware Diagnostics	System diagnostics	Webserver	RFID	SINAMICS	Motor starter	Safety		Back	Submenu Hardware
	Version	Panel Control	System	PLC system data				Software Version	Back	Submenu System

Fig. 3-2 Structure of the menu for operator panels with 12,1" screen display (TP1200 Comfort)

	OEM	Prepare	Manual	Machine	diagnostics	Hardware Diagnostics	System			Main menu
	Power Up Condition	Cycle Types	Select Units	Symbol / Absolute	Previous page	Next page	Back			Submenu Prepare
	Manual	Nut runner	Nut runner group	User Defined	Symbol / Absolute	Previous page	Next page			Submenu Manual
	Machine Overview	Part Count	Cycle Time				Back			Submenu Machine
	Alarm	alarm buffer	Interlock		PLC code viewer	Acknowledge message	Back			Submenu Diagnostics
	Hardware Diagnostics	System diagnostics	Webserver	RFID	SINAMICS	Motor starter	Safety			Submenu Hardware
	Version	Panel Control	System	PLC System data	Version blocks	Software Version	Back			Submenu System

Fig. 3-3 Structure of the menu for operator panels with 9" screen display (KTP900F Mobile)

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 main screen **Overview (SO_01_001_MainScreen)**. From this screen, the horizontal softkeys can be used to change to one of the 7 or 9 main menus.

In the 12,1" variant the two standard main menus **OEM** and **Process** plus a disengaged main menu are available for the machine manufacturer for including machine-specific screens and functions.

In the 9" variant only the **OEM** standard main menu is available.

It is possible for both variants to create a third menu level.

Navigation and function keys:

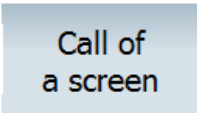
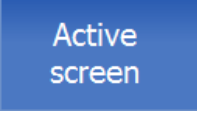
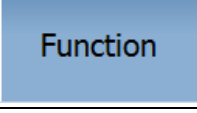
The gray-shaded buttons are navigation and function keys assigned to the individual screens in the corresponding submenus.

Click the **Back** button to return from the current menu to the previous menu. The **Home** button is configured to return to the HMI Lite main screen **Overview (SO_01_001_MainScreen)**.

Further buttons are described in the corresponding chapters.

Button styles

Table 3-1 Button styles

Button	Explanation
	Button for selecting a screen
	Button of the current screen
	Button for calling a function within the current screen

3.3 Clock memory byte of the control

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 control <> OP communication).

The clock memory byte must be transferred as input parameter to the **LTLL_Basic** block. The **LTLL_Basic** creates pulses of the individual clock signals and cyclically updates the variables of the data blocks.

3.4 PLC system time

3.4.1 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.

3.4.2 System time and date

The **LTLL_Basic** block reads the local date and the local time of the PLC via **RD_LOC_T** command. The system time will be written into the data area **LTLL_Data.HMI[X].areapointer.dateTimePLC**.

The time of the panel will be synchronized automatically via the time synchronization setting of the TIA portal.

Note

Please note that you must adapt the panel settings for winter and summer time every six month. You must not use the area pointers **date/time** and **date/time control**.

You can also use another time synchronization.

3.5 Job mailbox HMI Lite

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:

Table 3-2 Structure of the job mailbox

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

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 initiated 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 control, the connection between the operator panel and the control 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 function block **LTLL_Basic** 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:

LTLL_Config.THIS[X].manualCommon.screenActiveTime



Important

The sign-of-life bit is not a real-time signal. Therefore it can take longer than one second before the signal has changed its status. This is depending on the data traffic on the network and the number of processes running on the operator panel.

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 control 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:

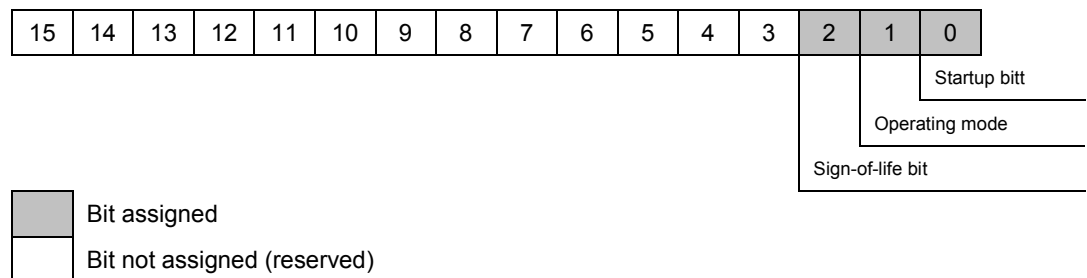


Fig. 3-4 Layout of the **Coordination** area pointer

3.6 LTLL_Basic block

The basic functions of HMI Lite are realized using the function block LTLL_Basic. This FB is responsible for the coordination of the interface DBs and HMI screens.

Call interface

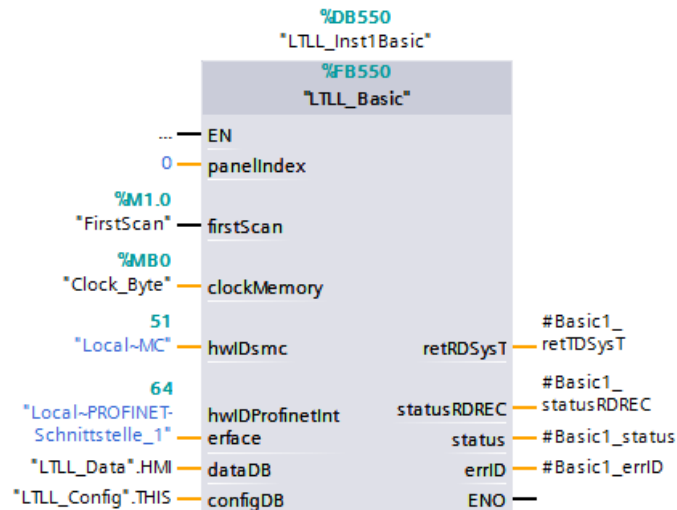


Fig. 3-5 Interface of the **LTLL_Basic** block

Parameters

Table 3-3 Description of the **LTLL_Basic** parameters

Name	Type	Default	Example	Description
panelIndex	INT	-	-	Panel index (0-based)
firstScan	BOOL	FirstScan	FirstScan	New start marker, restart bit
clockMemory	BYTE	Clock_Byte	Clock_Byte	Clock memory byte, configured in the object properties of the CPU (HW Config)
hwIDsmc	HW_IO	Local~MC	Local~MC	System constant of the SMC card of the control
hwIDProfinet Interface	HW_IO	-	-	System constant of the PROFINET interface
dataDB	Array[*] of LTLL_typeData	LTLL_Data.HMI	LTLL_Data.HMI	HMI Lite runtime data DB
configDB	Array[*] of LTLL_typeConfig	LTLL_Config.THIS	LTLL_Config.THIS	HMI Lite configuration DB
retRDSysT	INT	-	-	Return value of RD_SYS_T
statusRDREC	DWord	-	-	Status of RDREC
status	Word	-	-	Block status
errID	Word	-	-	Local error ID



Notes

4

4 Procedure to be applied when creating new screens

4.1 The Template screen

The **SS_00_000_Template** screen is used to add machine-specific screens while retaining the screen layout and the menu structure.

This results in the following procedure:

1. Duplicate the **SS_00_000_Template** screen
2. Rename the screen
3. Configure the screen
4. Include the screen in the menu structure



Fig. 4-1 **SS_00_000_Template** screen for creating your own screens

4.2 Designation conventions

All WinCC 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 is allowed to change the element?
- How are the individual elements linked with each other?

All WinCC elements that can be changed by the user (configuring) are designated with **SO_**. When the elements in WinCC 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 elements, such as screens, variables, graphics and symbol lists must be named using these uniform designation conventions.

Table 4-1 Designation of the convention syntax for the WinCC screen elements

Name structure of the screen elements: AB_XX_XXX_Name	
Icon	Description
A	Who created the associated screen element? S: Siemens (HMI Lite Standard) P: Siemens project specific (not HMI Lite standard) O: OEM (machine manufacturer) C: Customer
B	Who is allowed to change the screen element? S: Siemens P: Siemens project specific (not HMI Lite standard) O: OEM (machine manufacturer) C: Customer
XX_XXX	Assignment of the screen elements to each other (e.g. 11_XXX means all elements of the SINAMICS diagnostics)
Name	Designation of the screen element (e.g. PartCounter)

Example

D	S	_	29_021	_	Recipes
					The name of the screen form is "Recipes".
					The number of the screen element is 29021.
					The screen form may only be changed by Siemens.
					The screen form was created project specific by Siemens.

All other elements only used in the "Recipes" screen, such as variables or symbol lists, also have the identification 29_021.

E.g.: Variable: SS_29_021_Index
 Variable: SS_29_021_SelectedMaster
 Text list: SO_29_021_SelectedMasterIndex

Global screen elements (identification 00_000)

All screen elements not uniquely assigned to a specific screen have the identification 00_000 (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 02_000. 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_02_001_Setup** screen).

4.3 Identification of the selected screen

The information which screen is selected on the operator panel is made available in the WinCC **SS_00_000_Data.global.readOnly.screenID** variable. For the screen construction, the corresponding value is written in the variable; for screen removal, the variable is set to zero.

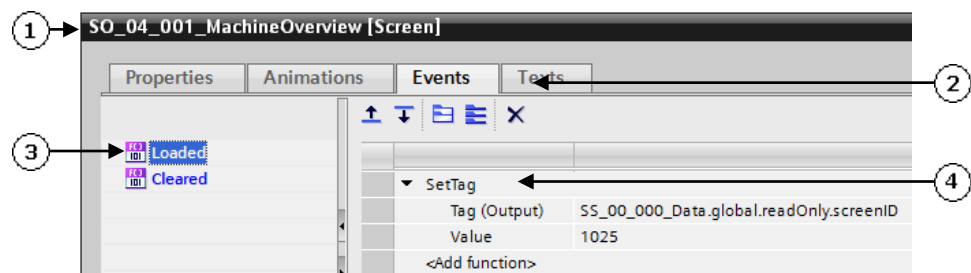
To keep the cycle time of the control as small as possible, the program code for a specific screen should be executed only when the corresponding screen is selected.

The WinCC **SS_00_000_Data.global.readOnly.screenID** variable is defined as follows:

Tag	SS_00_000_Data.global.readOnly.screenID
Format	WORD
PLC address	LTLL_Data.HMI[X].global.readOnly.screenID

Configuring screen events

For details of configuring screen events, see Figure 4-2.



- (1) Properties dialog of a screen
- (2) Register Events
- (3) Event Loaded when the function is initiated
- (4) Function SetTag to be performed

Fig. 4-2 Configuring the screen event to identify the selected screen

Codes to identify the individual screens

Table 4-2 Identification code of the individual screens

WinCC screen number	Designation of the system screen	Code to identify the screen form			
		High- Byte [dec.]	Low- Byte [dec.]	[dec.]	[hex.]
General screens					
1001	SO_01_001_MainScreen	01	001	257	0x0101
1011	SS_01_011_Version*	01	011	273	0x0111
1012	SS_01_012_Version1**	01	012	274	0x0112
1013	SS_01_013_Version2**	01	013	275	0x0113
1014	SS_01_014_PanelControl	01	014	276	0x0114
1015	SS_01_015_SystemScreen	01	015	277	0x0115
1016	SS_01_016_PLCSystemDaten	01	016	278	0x0116
1017	SS_01_017_EKS	01	017	279	0x0117
Manual screens					
2001	SS_02_001_Setup	02	001	513	0x0201
2002	SS_02_002_PowerUpCondition	02	002	514	0x0202
2003	SS_02_003_Unit	02	003	515	0x0203
2004	SS_02_004_NutRunner	02	004	516	0x0204
2005	SS_02_005_NutRunnerGroup	02	005	517	0x0205
2006	SS_02_006_CycleTypes	02	006	518	0x0206
2007	SS_02_007_UserDefine	02	007	519	0x0207
Alarms and messages					
3001	SS_03_001_Alarm	03	001	769	0x0301
3002	SS_03_002_AlarmHistory	03	002	770	0x0302
Machine information					
4001	SO_04_001_MachineOverview	04	001	1025	0x0401
4011	SS_04_011_PartCounter*	04	011	1035	0x040B
4012	SS_04_012_PartCounterOverall**	04	012	1036	0x040C
4013	SS_04_013_PartCounterSpecific**	04	013	1037	0x040D
4021	SO_04_021_CycleTimes	04	021	1045	0x0415
4031	SS_04_031_Interlocks	04	031	1055	0x041F
Hardware diagnostics					
10001	SS_10_001_HardwareDiagnostic	10	001	2561	0x0A01
10011	SS_10_011_Systemdiagnose	10	011	2571	0x0A0B
10012	SS_10_012_Webserver	10	012	2572	0x0A0C
10013	SS_10_013_PlcCodeViewer***	10	013	2573	0x0A0D
SINAMICS diagnostics					
11001	SS_11_001_ControlStatusword*	11	001	2817	0x0B01
11002	SS_11_002_ControlWord**	11	002	2818	0x0B02

WinCC screen number	Designation of the system screen	Code to identify the screen form			
		High- Byte [dec.]	Low- Byte [dec.]	[dec.]	[hex.]
11003	SS_11_003_StatusWord**	11	003	2819	0x0B03
11011	SS_11_011_EPOSStatus*	11	011	2827	0x0B0B
11012	SS_11_012_EPOSStatusWord**	11	012	2828	0x0B0C
11013	SS_11_013_EPOSPositioning**	11	013	2829	0x0B0D
11021	SS_11_021_FaultsAndWarnings*	11	021	2837	0x0B15
11022	SS_11_022_Faults**	11	022	2838	0x0B16
11023	SS_11_023_Warnings**	11	023	2839	0x0B17
11031	SS_11_031_SafetyStatusword	11	031	2847	0x0B1F
<i>Motor starter diagnostics</i>					
12001	SS_12_001_ControlStatus*	12	001	3073	0x0C01
12002	SS_12_002_Control**	12	002	3074	0x0C02
12003	SS_12_003_Status**	12	003	3075	0x0C03
12011	SS_12_011_DataStatistics*	12	011	3083	0x0C0B
12012	SS_12_012_MeasuredData**	12	012	3084	0x0C0C
12013	SS_12_013_Statistics**	12	013	3085	0x0C0D
12021	SS_12_021_LogbookDeviceError	12	021	3093	0x0C15
12022	SS_12_022_LogbookTrippingOperations	12	022	3094	0x0C16
12023	SS_12_023_LogbookEvents	12	023	3095	0x0C17
<i>RFID diagnostics</i>					
13001	SS_13_001_RFID	13	001	3329	0x0D01
<i>Safety</i>					
14001	SS_14_001_Safety	14	001	3585	0x0E01
<i>Siemens end customer-screens</i>					
29000	PP_29_yyy_ScreenName PP stands for project specific screens	29	yyy = 0-255	-	-
<i>OEM-specific screens</i>					
30000	OO_30_yyy_ScreenName****	30	yyy = 0-255	-	-
<i>Customer/project-specific screens</i>					
31000	CC_31_yyy_ScreenName****	31	yyy = 0-255	-	-

* only for TP1200 Comfort

** only for KTP900F Mobile

*** Optionally available in the project library

**** Placeholder for project or customer specific screen (not available in the standard version)

4.4 Style elements

A WinCC style has been created for HMI Lite V8.0 which ensures that all elements have the same Look and Feel.

In your WinCC objects, activate **Properties > Styles/Designs > Style/Design settings** and select the appropriate style element under **Style item appearance**

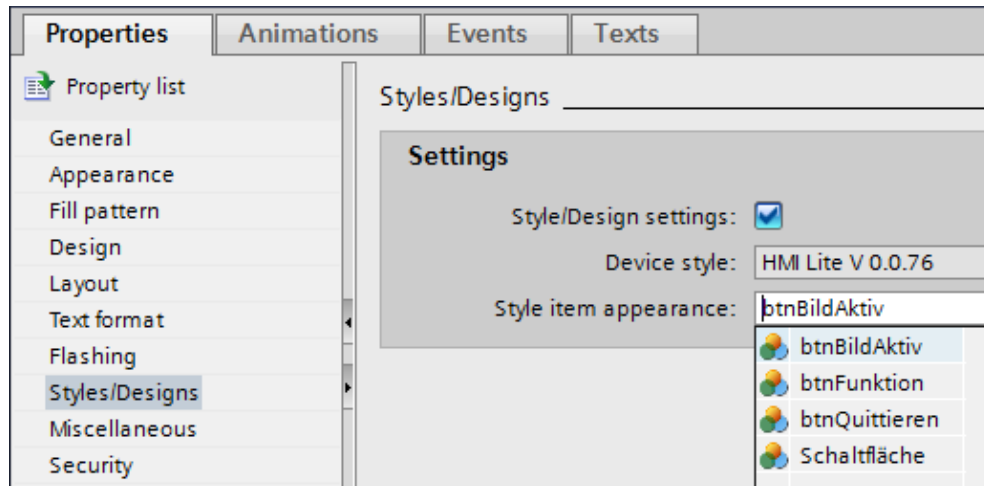


Fig. 4-3 Style elements

In some cases, several style elements are provided for WinCC objects. For example, three decisive style elements are available for buttons:

- Style element **Schaltfläche**: Call-up of other screens
- Style element **btnBildAktiv**: Current screen
- Style element **btnFunktion**: Function including the current screen

Further objects, for example, I/O fields and text fields with various style elements are provided. These include e.g. text alignment, transparent background, headers, colored background.

See also chapter 3.2 Button styles

Notes



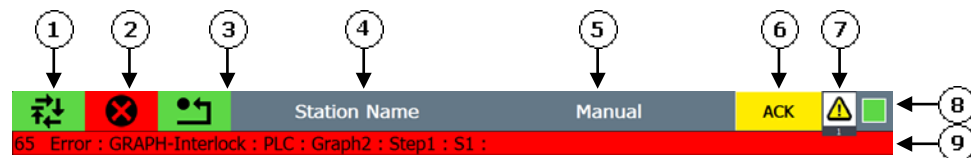
5

5 Header and Operator Information

5.1 Header

5.1.1 Layout of the header

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



- (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) Acknowledge pushbutton for the current message in the message line
- (7) Message indicator
- (8) Sign-of-life bit
- (9) Alarm and message line

Fig. 5-1 Layout of the HMI Lite header

5.1.2 Display of current operating mode

The currently selected operating mode is displayed.
By default, the following operating modes are defined:

Table 5-1 Display of the operating modes in the header

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

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.

Table 5-2 Display of operating modes (selected, active/not active)

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

Operating mode selection runtime interface **LTLL_Data.HMI[X].header.mode**

The associated operating mode is displayed using the interface bits in the **LTLL_Data** data block. The operating mode is displayed when the interface bit = **TRUE**.

Table 5-3 Display of the current operating mode – interface bits

Text	Icon	Interface	Type

Auto		LTLL_Data.HMI[X].header.mode.automatic	BOOL
Cycle		LTLL_Data.HMI[X].header.mode.cycle	BOOL
Step		LTLL_Data.HMI[X].header.mode.step	BOOL
Manual		LTLL_Data.HMI[X].header.mode.manual	BOOL

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

Operating mode runtime interface selected/active

If the **LTLL_Data.HMI[X].header.mode.active** interface bit is set to **TRUE**, the operating mode will be displayed as active.

Configuration

No configuring is required.

5.1.3 Status display

The following states are possible for the machine status display:

Table 5-4 Machine status display

Text	Icon	Meaning	Description
OK		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

LTLL_Data.HMI[X].header.status runtime interface

The **Status display** is controlled with the following status bits in the **LTLL_Data** data block:

Table 5-5 Status display – interface bits

Meaning	Interface	Type
Ready to run	--- (no additional status bit has the status "1" – status = ready)	BOOL
Warning	LTLL_Data. HMI[X].header.status.warning	BOOL
Alarm	LTLL_Data. HMI[X].header.status.alarm	BOOL

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

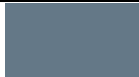
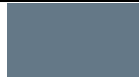
Configuration

No configuring is required.

5.1.4 Display of the initial state

The following states are possible for the initial state display:

Table 5-6 Initial state display – possible states

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

LTLL_Data. HMI[X].header.position runtime interface

The **Initial state** display is made using the following bit in the **LTLL_Data** data block:

Table 5-7 Initial state display – interface bits

Meaning	Interface	Type
Empty	---	
Initial state	LTLL_Data. HMI[X].header.position.home	BOOL

The **Initial state** status is displayed when the bit is **TRUE**.

Configuration

No configuring is required.

5.1.5 Text boxes

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

LTLL_Data.HMI[X].header.textindex runtime interface

The text is controlled using two variables in the **LTLL_Data** block. The text assigned to the value of the variable of the WinCC text list is displayed as text.

Address	LTLL_Data.HMI[X].header.textindex1 LTLL_Data.HMI[X].header.textindex2
Format	WORD
Range of values	0-65535
Default setting	W#16#0

Configuration

Text list	SO_00_000_HeaderTextlist_1 SO_00_000_HeaderTextlist_2
Display	Text
Format	Decimal
Value	Text
1	[Text to be displayed]
...	...

The **SO_00_000_HeaderTextlist_2** 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 **Screen number** area pointer into the **LTLL_Data.HMI[X].header.textindex2** variable.

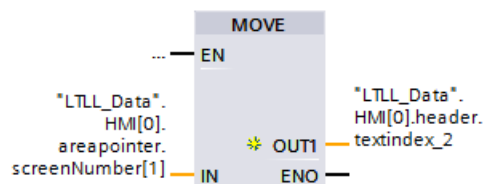


Fig. 5-2 Assignment of the 2nd text list **SO_00_000_HeaderTextlist_2**

The machine manufacturer must extend the **SO_00_000_HeaderTextlist_2** text list if new screens are to be added to the WinCC configuration.

5.1.6 Sign-of-life of the PLC

The sign-of-life in the header displays the operating mode of the PLC.

Table 5-8 Display of the sign-of-life of the PLC

Field	Interface
Periodic flashing in intervals of approximately one second  	The PLC is in the RUN operating mode. The Panel – PLC communication occurs.
	The communication with the PLC has been interrupted.
	The PLC is in the STOP operating mode

5.1.7 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 display is changed using the following:

LTL Config.THIS[X].header.useTextHeader

FALSE = Symbolic header

TRUE = Text-based header

5.2 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.

Runtime interface **LTLL_Data.HMI[X].global.prompt**

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

The first variable, **LTLL_Data.HMI[X].global.prompt.index**, is used to select which text from the WinCC text list is to be displayed.

The second variable, **LTLL_Data.HMI[X].global.prompt.attribut**, is linked with the attribute of the text output field. The variable controls the color change of the flashing of the text output field.

Address:	LTLL_Data.HMI[X].global.prompt.index
Format:	WORD
Range of values:	1..
Default setting:	W#16#0

Address:	LTLL_Data.HMI[X].global.prompt.attribut
Format:	WORD
Range of values:	1..
Default setting:	W#16#0

Configuration

The WinCC **SO_00_000_OperatorPrompt** text list contains the text to be displayed in the text field for operator information.

Text list		SO_00_000_OperatorPrompt
Display		Text
Type		Decimal
Value	[Text number]	[Text to be displayed]
...	...	...



Notes

6

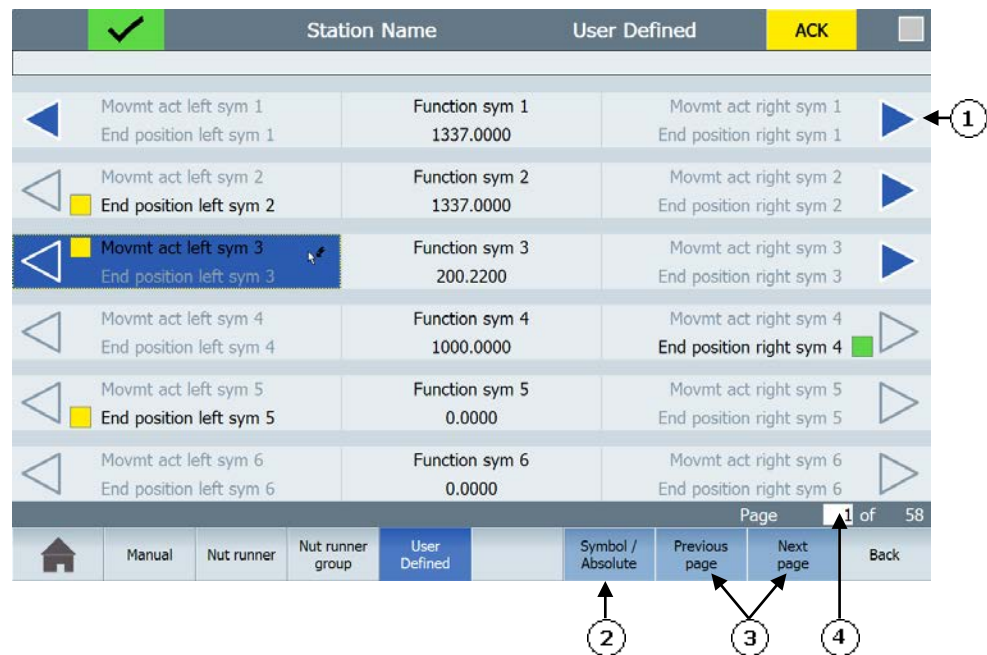
6 Manual Operating

6.1 Overview

6.1.1 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.



- (1) Movement or function line
- (2) Switch between Symbolic/Absolute
- (3) Scroll to previous page/next page
- (4) Current page/total number of pages

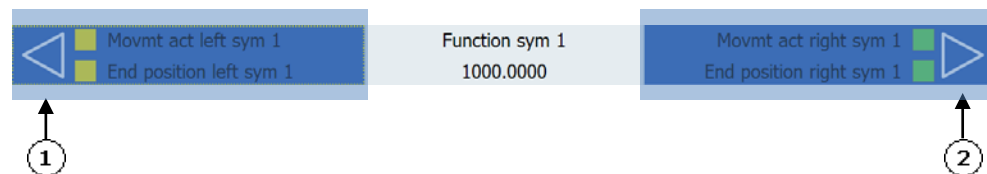
Fig. 6-1 Layout of the manual operating screens

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.

As shown in the following, each movement/function can be initiated or selected in the respective areas by touching.



- (1) Touch-sensitive area left (button not visible)
- (2) Touch-sensitive area right (button not visible)

Fig. 6-2 Manual operation – Selection/actuation of a function line

There are five different modes for selecting function lines

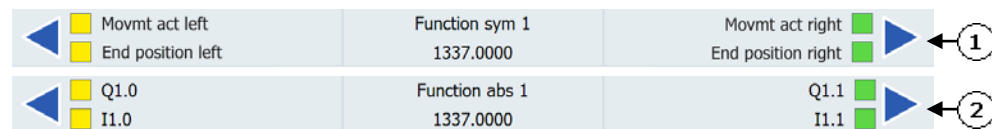
- Touch direct:
The function is active as long as the button is pressed.
- Touch pre-selection:
To prevent the inadvertent initiation of a movement, the system is designed such that to initiate a movement, it must first be selected by touching the appropriate touch-sensitive surface. The selection of the movement is confirmed by blue flashing on the movement side. Once the selection of the movement has been confirmed, it can be initiated by subsequently touching the touch-sensitive area. The movement side is permanently marked with blue as confirmation.
The function remains selected until one of the following events occurs:
Another movement is selected.
You scroll to another page.
Another screen is selected.
The sign-of-life monitoring deactivates the movement.
- Touch external:
The function is selected by the buttons. The function is enabled via external buttons.
- Softkeys direct*:
The function is enabled by pressing the softkeys on the side.
- Softkeys external *: The function is selected via the softkeys on the side.
The function is enabled via external buttons.

*: (only for HMI devices with vertical softkeys)

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, Q1.0) that are assigned to the corresponding movements/functions (see fig. 6-3).

The representation 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

Fig. 6-3 Manual operating screens - absolute and symbolic view

Scrolling

A scroll function can be used to fetch all configured actions for a maximum display of six (12,1" device) or three (9" device) movements/functions per page.

When the scroll function is performed, all displayed function lines will be replaced by the function lines present on the next page.

When the last page is reached and the **Next page** key is pressed, the display changes back to the first page. Similarly, when the first page is reached and the **Previous page** key is pressed, the display jumps to the last page.

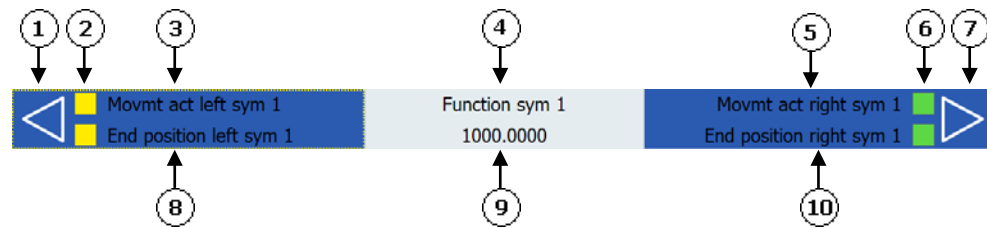
You cannot change the screen (the page is locked) while a movement/function is being performed.

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.

6.1.2 Elements of the movement/function line

Each movement line consists of the basic elements shown in the following figure:



- (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

Fig. 6-4 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. 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/function.

Executability (1, 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 a contour.

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

Final state (end position) (8, 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 respective text of this position is gray and the colored square is not displayed

If the target position has been reached, the appropriate text is black and a yellow (left-hand side) or green (right-hand side) square is displayed.

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

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

Execution (3, 5)

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

If the output is disabled, the corresponding text is gray and the colored square is not displayed.

If the output is enabled, the text is black and the yellow (left-hand side) or green (right-hand side) square is displayed.

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.

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

Key/button activated (2, 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 movement side becomes blue.

If a key has not been pressed or not confirmed by the PLC, the movement side remains gray.

The **Key/button activated** element confirms the preselection or selection of a function on the touch display with the following states:

If a function has been preselected, this is indicated by the flashing of the **movement area**.

If the touch-sensitive surface of a function is activated a second time and this is confirmed by the controller, the movement area is marked in blue.

If the function is not active or preselected, the movement area becomes gray.

For further details on preselection, also refer to the section on **Movement and function line**.

6.1.3 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. The following figure shows the assignment of the function numbers over several pages.

Station Name		Manual	ACK
◀	Movmt act left sym 1 End position left sym 1	Function sym 1 0.0000	Movmt act right sym 1 End position right sym 1 ▶
◀	Movmt act left sym 2 End position left sym 2	Function sym 2 0.0000	Movmt act right sym 2 End position right sym 2 ▶
◀	Movmt act left sym 3 End position left sym 3	Function sym 3 0.0000	Movmt act right sym 3 End position right sym 3 ▶
◀	Movmt act left sym 4 End position left sym 4	Function sym 4 0.0000	Movmt act right sym 4 End position right sym 4 ▶
◀	Movmt act left sym 5 End position left sym 5	Function sym 5 0.0000	Movmt act right sym 5 End position right sym 5 ▶
◀	Movmt act left sym 6 End position left sym 6	Function sym 6 0.0000	Movmt act right sym 6 End position right sym 6 ▶
Page 1 of 58			
Home	Manual	Nut runner	Nut runner group
User Defined	Symbol / Absolute	Previous page	Next page
Back			

◀	Movmt act left sym 7 End position left sym 7	Function sym 7 0.0000	Movmt act right sym 7 End position right sym 7 ▶
◀	Movmt act left sym 8 End position left sym 8	Function sym 8 0.0000	Movmt act right sym 8 End position right sym 8 ▶
◀	Movmt act left sym 9 End position left sym 9	Function sym 9 0.0000	Movmt act right sym 9 End position right sym 9 ▶
◀	Movmt act left sym 10 End position left sym 10	Function sym 10 0.0000	Movmt act right sym 10 End position right sym 10 ▶
◀	Movmt act left sym 11 End position left sym 11	Function sym 11 0.0000	Movmt act right sym 11 End position right sym 11 ▶
◀	Movmt act left sym 12 End position left sym 12	Function sym 12 0.0000	Movmt act right sym 12 End position right sym 12 ▶
Page 2 of 58			
Home	Manual	Nut runner	Nut runner group
User Defined	Symbol / Absolute	Previous page	Next page
Back			

- (1) Page 1, Row 1, Function 1
- (2) Page 1, Row 3, Function 3
- (3) Page 1, Row 4, Function 4
- (4) Page 1, Row 5, Function 5
- (5) Page 2, Row 1, Function 6
- (6) Page 2, Row 2, Function 7

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

HMI Lite assigns the page layout of the movements/functions based on the total number of movements/functions configured in the selected screen and depending on how many movements/functions can be displayed on a page. This means the movement-specific interfaces are page-dependent.

Note

The 12,1" display can display six function lines per screen page; the 9" devices can display four function lines per page.

6.2 Purpose of the individual manual operating screens

6.2.1 Setup screen

The **Setup screen** contains a maximum of 348 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 setup screen for the operator and, for example, can be assigned to a specific plant section.

6.2.2 Power up condition

The **Power up condition** screen contains up to 348 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

6.2.3 Selecting/deselecting units

The **Units** screen contains up to 348 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.

6.2.4 Selecting/deselecting nut runners

The **Nut runner** screen contains up to 348 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.

6.2.5 Selecting/deselecting nut driver groups

The **Nut runner groups** screen contains up to 348 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.

6.2.6 Selecting/deselecting cycle type

The **Cycle type** screen contains up to 348 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.

6.2.7 User operating screen

The User operating screen is a freely-configurable manual operating screen that can be used for machine- or project-specific functions. It has 348 function lines.

6.3 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 **LTLL_Config** configuration DB and the **LTLL_Data** runtime DB have their own data area for each screen; this data area is also defined by the designation of the associated screen.

Table 6-1 Manual operating screens – structure of the configuration interface

Name of the screen in WinCC	Name of the area in LTLL_Data and LTLL_Config
SS_02_001_Setup	screenSetup
SS_02_002_PowerUpCondition	screenPowerup
SS_02_003_Unit	screenUnit
SS_02_004_NutRunner	screenNutrunner
SS_02_005_NutRunnerGroup	screenNutrunnerGroup
SS_02_006_CycleTypes	screenCycletype
SS_02_007_UserDefine	screenUserDefine

6.4 Configuration

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

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

6.4.1 Global configurations

The **LTLL_Config.THIS[X].manualCommon** 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 **LTLL_Config**:

Address	LTLL_Config.THIS[X].manualCommon. absoluteDisplayTime
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 **LTLL_Config** in the following data address:

Address	LTLL_Config.THIS[X].manualCommon. touchPreselectionTime
Format	TIME
Range of values	T#1MS...T#24D20H31M23S647MS
Default setting	T#2S (2 seconds)

6.4.2 Number of movement/function lines

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

Address	LTLL_Config.THIS[X]. screenAAAAAA.numberOfWorks where AAAAAA = name of the screen (see table 6-1)
Format	INT
Range of values	1...348 – for all manual operating screens
Default setting	The maximum number of available lines

6.4.3 Grouping of the movement lines in the setup screen

To assign the manual operation to function groups, the setup screen can be configured more than once with different pages.
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 **SetTag** function that is configured in addition to the **ActivateScreen** function on the button that selects the setup screen.

Note

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

This function is only available on the **setup** screen (**SS_02_001_Setup**).

Address	WinCC variables: SS_02_001_setupScreenNumberOfLastPage (last page) SS_02_001_setupScreenNumberOfFirstPage (first page)
Format	BYTE
Range of values	1...Max – the maximum value depends on the number of movement lines (see Section 6.4.2) and on the number of lines per page (see Section 6.7). Example: For 348 movement lines and 6 lines per page, this gives 58 pages with movement lines, consequently, the value range is 1...58.

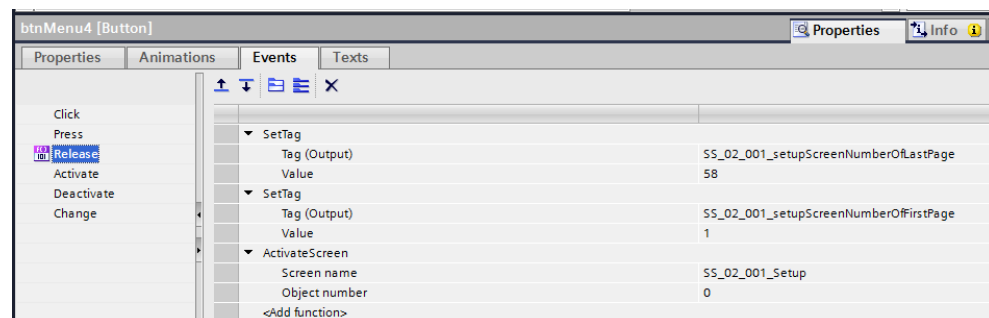


Fig. 6-6 WinCC configuration of the screen selection of the setup screen in groups

6.4.4 Hiding elements of the function line

It is possible to hide elements from the function line (see Fig. 6-7 Manual operating screen – hiding screen elements).

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

- **Executability** on the left-hand side;
- **Executability** on the right-hand side;
- **Position**;
- All elements

▶	Movmt act left sym 1 End position left sym 1	Function sym 1 1337.0000	Movmt act right sym 1 End position right sym 1	▶	1
	Movmt act left sym 1 End position left sym 1	Function sym 1 1337.0000	Movmt act right sym 1 End position right sym 1	▶	2
▶	Movmt act left sym 1 End position left sym 1	Function sym 1 1337.0000	Movmt act right sym 1 End position right sym 1		3
	Movmt act left sym 1 End position left sym 1	Function sym 1 1337.0000	Movmt act right sym 1 End position right sym 1		4
▶	Movmt act left sym 1 End position left sym 1	Function sym 1	Movmt act right sym 1 End position right sym 1	▶	5
	Movmt act left sym 1 End position left sym 1	Function sym 1	Movmt act right sym 1 End position right sym 1	▶	6
▶	Movmt act left sym 1 End position left sym 1	Function sym 1	Movmt act right sym 1 End position right sym 1		7
	Movmt act left sym 1 End position left sym 1	Function sym 1	Movmt act right sym 1 End position right sym 1		8
					9

- (1) All elements are visible
- (2) The left-hand side is hidden
- (3) The right-hand side is hidden
- (4) Both sides are hidden
- (5) The position is hidden
- (6) Position and left-hand side are hidden
- (7) Position and right-hand side are hidden
- (8) Position, left- and right-hand side are hidden
- (9) All elements are hidden

Fig. 6-7 Manual operating screen – hiding screen elements

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

Address	LTLL_Config.THIS[X].screenAAAAAA.rows[Y] where AAAAAA = name of the screen (see table 6-1) and Y is the number of the associated function line
Format:	LTLL_typeManualConfig
Range of values	-
Default setting	-

Two configurations (configuration 1 and configuration 2) are possible for each movement/function:

configs[0].hiddenLeft	configuration 1: left-hand side is hidden
configs[0].hiddenRight	configuration 1: right-hand side is hidden
configs[0].hiddenPosition	configuration 1: position is hidden
configs[0].hiddenAllOther	configuration 1: all other elements are hidden
configs[1].hiddenLeft	configuration 2: left-hand side is hidden
configs[1].hiddenRight	configuration 2: right-hand side is hidden
configs[1].hiddenPosition	configuration 2: position is hidden
configs[1].hiddenAllOther	configuration 2: all other elements are hidden

Only one configuration can be active for all movements/functions at one time. The associated active configuration (configuration 1 or 2) can be selected dynamically using the **selectConfig** input parameter of the **LTLL_Manual** block. The dynamic changing of the configuration settings allows movement/function elements to be displayed or hidden depending on the associated machine status (e.g. machine in manual operation).

Example configurations

Nachfolgend finden Sie einige Projektierungsbeispiele:

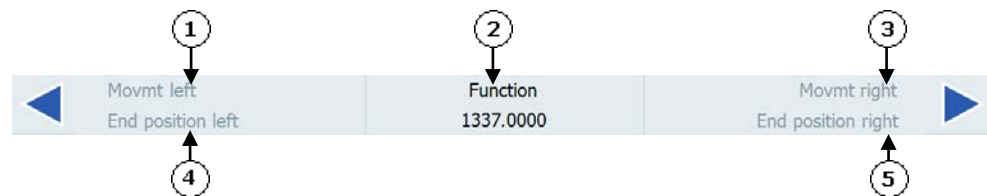
- The **Position** element is hidden for both configuration settings:
Example: all types of machine elements (e.g. pumps, valves) that do not supply any confirmation of the position
- The **Executable** element is hidden for both configuration settings:
Example: Machine elements that are not controlled from the operator panel
Only the status needs to be displayed here (e.g. the **on/off** state controlled by the pushbutton).
- The **Executable** element is hidden for one configuration setting:
Example:
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/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. This means that the output signals are initiated by pressing the keys to the left or right of the movement/function or by touching the buttons, even if **movement** items are hidden. The machine-specific control program must realize any required interlock functionality.

6.4.5 Display text

All text items displayed in the manual operating screens are configured in the WinCC text lists. Each screen has its own text list, where the text can be specified for each element.



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

Fig. 6-8 Manual operating screens – text lists

All text lists have the same structure.

Table 6-2 Manual operating screens – structure of the text lists

Text list		SO_02_001_Setup SO_02_002_PowerUpCondition SO_02_003_Unit SO_02_004_Nutrunner SO_02_005_NutrunnerGroup SO_02_006_CycleType SO_02_007_UserDefine
Display		Text
Format		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
	...	

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 Q1.0 and Q1.1 outputs. The movement is limited by limit switches connected to the I1.0 and I1.1 inputs. Consequently, the display text is as follows:

Table 6-3 Manual operating screens – display texts example

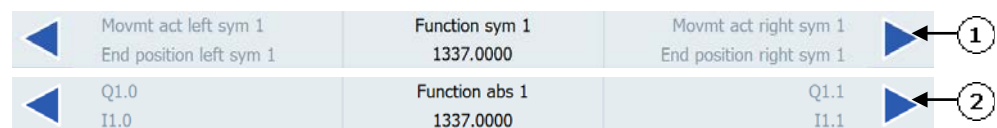
Text element of the movement line	Text to be displayed
Designation text for the symbolic view:	Function sym 1
Designation text for the absolute view:	Function abs 1
Execution left text for the symbolic view:	Movmt act left sym 1
Execution left text for the absolute view:	Q 1.0
Final state left text for the symbolic view:	End position left sym 1
Final state left text for the absolute view:	I1.0
Execution right text for the symbolic view:	Movmt act right sym 1
Execution right text for the absolute view:	Q1.1
Final state right text for the symbolic view:	End position right sym 1
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:

Table 6-4 Manual operating screens – text list example

Text list		SO_02_001_Setup
Value	10	Function sym 1
Value	11	Function abs 1
Value	12	Movmt act left sym 1
Value	13	Q1.0
Value	14	End position left sym 1
Value	15	I1.0
Value	16	Movmt act right sym 1
Value	17	Q1.1
Value	18	End position right sym 1
Value	19	I1.1

The specified movement is displayed as follows:



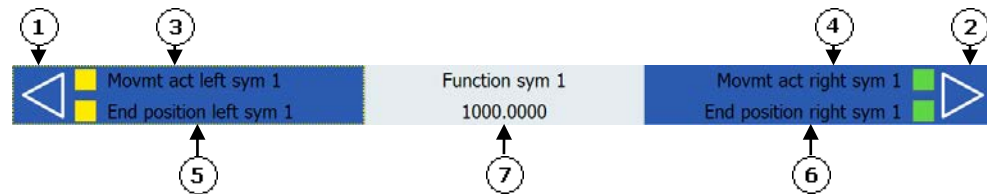
(1) Symbolic view

(2) Absolute view

Fig. 6-9 Manual operating screens – example for the configuration of a text

6.5 Runtime interface

Color changes show the details of the binary state of a movement or function. 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) Position

Fig. 6-10 Manual operating screens – dynamic movement elements

6.5.1 Interface for information about the binary state

The data addresses in the **LTLL_Data** data block control the details concerning the binary state of a movement/function.

Address	LTLL_Data.HMI[X].screenAAAAAA.rows[Y] where AAAAAA = name of the screen (see table 6-1) and Y is the number of the associated function
Format	LTLL_typeManualData
Range of values	-
Default setting	-

Each grouping element represents a movement/function.

runtimeInterface.executabilityLeft	Executable, left side
runtimeInterface.executabilityRight	Executable, right side
runtimeInterface.confirmExecuteLeft	Execution confirmation left side
runtimeInterface.confirmExecuteRight	Execution confirmation right side
runtimeInterface.finalPositionLeft	End position left side
runtimeInterface.finalPositionRight	End position right side
runtimeInterface.position	Position
controlInterface.leftFunctionActive	Reserved (must not be written to)
controlInterface.rightFunctionActive	Reserved (must not be written to)

**Important**

The bits in the data interface under `controlInterface` are used as control signals (HMI device ⇒ controller). The bits that supply information about the status must therefore be addressed individually or via `LTLL_typeManualDataRuntime`. If all status information was written concurrently with a single `LTLL_typeManualData` 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 must 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 must 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 must 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 must 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.

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

Selected screen

Address	LTLL_Data.HMI[X].global.readOnly.screenID
Format	WORD
Range of values	W#16#0000 ... W#16#FFFF (0...65535) for the identification of the screen, see table 6-4
Default setting	---

Current page

Address	LTLL_Data.HMI[X].manualCommon.readOnly.currentPage
Format	UInt
Range of values	0 bis 65535
Default setting	---

First and last visible line

Address	LTLL_Data.HMI[X].manualCommon.readOnly. rowVisibleFirst LTLL_Data.HMI[X].manualCommon.readOnly. rowVisibleLast
Format	WORD
Range of values	W#16#0001...W#16#0040 (1...348)
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 variables under a **readOnly** structure are internal variables and may only be used with read access.

6.6 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.

6.6.1 Job mailbox

The data addresses of the job mailbox belong to the **LTLL_Data.HMI[X].global.job** and are defined as follows:

Address	LTLL_Data.HMI[X].global.job number parameter_1 parameter_2 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#0201 for the SS_02_001_Setup screen (see table 6-4)
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 6-4:

Table 6-5 Operating screens - code for identifying the screen in the job mailbox

Screen	Identification code of the associated screen
SS_02_001_Setup	W#16#0201
SS_02_002_PowerUpCondition	W#16#0202
SS_02_003_Unit	W#16#0203
SS_02_004_NutRunner	W#16#0204
SS_02_005_NutRunnerGroup	W#16#0205
SS_02_006_CycleTypes	W#16#0206
SS_03_007_UserDefined	W#16#0207

Example

The **SS_02_001_Setup** 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#0201 ⇒ **SS_02_001_Setup** screen
 Parameter 1: W#16#0002 ⇒ **Second movement line** function
 Parameter 2: W#16#0002 ⇒ **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.

6.6.2 Binary control interface

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

Address	LTLL_Data.HMI[X].screenAAAAAA.rows[Y].controlInterface where AAAAAA = name of the screen (see table 6-1) and Y is the number of the associated function
Format	LTLL_typeManualDataControl
Range of values	-
Default setting	-

controlInterface. leftFunctionActive	Left movement/function activated
controlInterface. rightFunctionActive	Right movement/function activated

When a movement/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.

6.7 LTLL_Manual block

The **LTLL_Manual** block 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/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

The **LTLL_Manual** block must be called cyclically.

Call interface

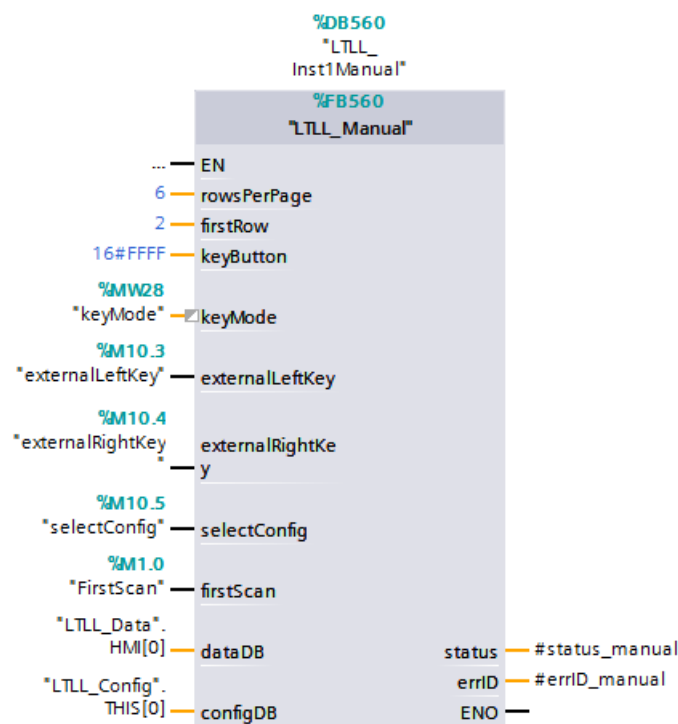


Fig. 6-11 Call interface of **LTLL_Manual** block

Parameters

Table 6-6 Parameters of the **LTLL_Manual** block

Name	Type	Default	Example	Description
dataDB	LTLL_type Data	LTLL_Data. HMI[0]	LTLL_Data. HMI[0]	HMI Lite runtime data block
configDB	LTLL_type Config	LTLL_Config. THIS[0]	LTLL_Config. THIS[0]	HMI Lite configuration data block
rowsPerPage	USInt	6 or 4	6	The number of function lines that can be displayed on the screen at the same time. 4lines for the 9" operator panel with header. 6 lines for the 12,1" operator panel with header.
firstRow	USInt	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 with header.
keyButton	Word	-	EW 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.
keyMode	USInt	-	0	Mode for executing the operation See chapter 6.1.1 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.

Name	Type	Default	Example	Description
externalLeftKey	Bool	-	-	Only relevant in the 3 and 4 key modes Performs the left command of the selected function.
externalRightKey	Bool	-	-	Only relevant in the 3 and 4 key modes Performs the right command of the selected function.
selectConfig	Bool	-	M5.3	Switch between the two configurations for hiding of individual elements of the function line FALSE = configuration 1 TRUE = configuration 2
firstScan	Bool	-	-	Restart flag, 1 – signal for the first cycle after CPU startup
status	Word	-	MW2	16#g8200 HMI Lite llicenseing failed
errId	Word	-	-	Local error handling

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

LTLL_Data.HMI[X].manualCommon.closedSelectedRow bit when the selection of a movement is to be disabled. For example, this can be implemented by activating a key switch.

The **LTLL_Data.HMI[X].manualCommon.resetSelectedRow** bit causes the program code to reset the selection and re-release the selection of other movements.

6.8 LTLL_ManualControl block

With the **LTLL_ManualControl** function, an individual manual operation line can be parameterized and the binary control interface can be queried.

Call interface



Fig. 6-12 Call interface of **LTLL_ManualControl** block

Parameters

Table 6-7 Parameters of the **LTLL_ManualControl** block

Name	Type	Default	Example	Description
dataDB	LTLL_type Data	LTLL_Data. HMI[0]	LTLL_Data. HMI[0]	HMI Lite Runtime data DB
rowConfig	LTLL_type Manual Config	-	LTLL_Config. THIS[0].screen Powerup. rows[1]	Configuration of a manual operation line
rowData	LTLL_type ManualData	-	LTLL_Data. HMI[0].screen Powerup.rows [1]	Runtime data of a manual operation line
useGraphForManualRow	Bool	-	-	TRUE: Manual operation line controls Graph steps FALSE: Manual operation line does not control any Graph steps
hiddenLeft1	Bool	-	-	Hide left configuration 1
hiddenRight1	Bool	-	-	Hide right configuration 1
hiddenPosition1	Bool	-	-	Hide position configuration 1
hiddenAllOther1	Bool	-	-	Hide all other elements configuration 1
scaleNumber1	USInt	-	-	Protection level configuration 1
hiddenLeft2	Bool	-	-	Hide left configuration 2
hiddenRight2	Bool	-	-	Hide right configuration 2
hiddenPosition2	Bool	-	-	Hide position configuration 2
hiddenAllOther2	Bool	-	-	Hide all other elements configuration 2
scaleNumber2	USInt	-	-	Protection level configuration 2
executabilityLeft	Bool	-	-	Executability left
executabilityRight	Bool	-	-	Executability right
confirmExecuteLeft	Bool	-	-	Left movement active
confirmExecuteRight	Bool	-	-	Right movement active
finalPositionLeft	Bool	-	-	End position left
finalPositionRight	Bool	-	-	End position right
position	LReal	-	-	Position
leftFunctionActive	Bool	-	-	Left function activated
rightFunctionActive	Bool	-	-	Right function activated
scaleNumberOK	Bool	-	-	Protection level sufficient
errID	Word	-	-	Local error ID
Return	Word			16#8200 HMI Lite licensing failed

6.9 LTLL_ManualGraph block

The **LTLL_ManualGraph** function block provides the following functionality:

- Display the executability of the movements displayed on the HMI device.
The executability is read from the interlock from GRAPH.
- Activate a configured GRAPH step for the selection of a movement by pressing a key on the operating screen.

Note

The **LTLL_ManualGraphExt** block requires GRAPH Version 3.0.

The **LTLL_ManualGraph** block must be called cyclically for each operator screen. An **LTLL_ManualGraphConfig** data block is needed for each manual operating screen.

Only one **LTLL_ManualGraphControl** is required.

Example:

When there are two sequencers which are to be used in the **Setup** screen and in the **Switch-on conditions** screen, the block must be called 2 times.

Call interface

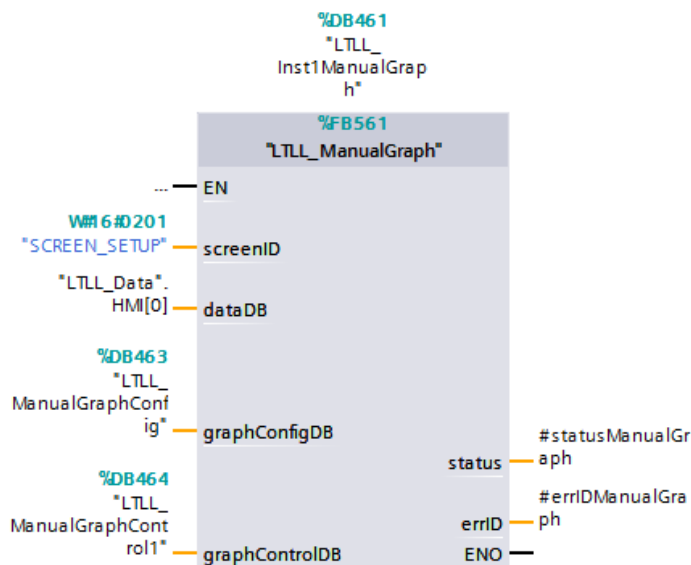


Fig. 6-13 Call interface of **LTLL_ManualGraph** block

Parameters

Table 6-8 Parameters of the **LTLL_ManualGraph** block

Name	Type	Default	Example	Description
screenID	Word	W#16#0201	SCREEN_SET UP	Screen ID of the operating screen for which the FB call is to be valid
dataDB	LTLL_typeData	LTLL_Data. HMI[0]	LTLL_Data. HMI[0]	HMI Lite runtime data block
graphConfigDB	LTLL_typeManualGraphConfig	LTLL_ManualGraphConfig	LTLL_ManualGraphConfig	HMI Lite sequencer operating diagram configuration
graphControlDB	LTLL_typeManualGraphControl	LTLL_ManualGraphControl1	LTLL_ManualGraphControl1	Interface DB for sequencer control in HMI Lite manual operating screens
status	Word	-	-	Block status
errID	Word	---	-	Local error ID

Output parameter status

Value (W#16#)	Description
8200	HMI Lite licensing failed
8201	Invalid screen ID

Functionality of the FB

If an operating screen is selected on the operator panel, the function block uses the **screenID** to check whether the operating screen is valid for calling the function block.

If the call is valid, it copies the parameterization of the **LTLL_ManualGraphConfig** for the movements displayed in the screen into its instance DB. Furthermore, it prepares the data for the block **LTLL_ManualGraphExt**. This block is responsible for controlling the sequencer and for the read-out of the interlock network status. The block **LTLL_ManualGraphExt** provides the interlock data for the block **LTLL_ManualGraph**. This, in turn, routes the data to the **LTLL_Data** block so that it is displayed on the screen. The triangle on the right-hand and left-hand edge of the movement line shows the executability on the screen.

See Section 6.1.1 Layout and basic functionality of the manual operating screens
If a movement is initiated by pressing a button, the FB activates the parameterized step in the corresponding sequencer. It deactivates the step when the button is released.

Requirements

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.

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

The function block uses the **OFF_SQ**, **S_ON**, **S_OFF** and **S_SEL** sequencer parameters. These parameters must not be overwritten by the user program while the step is being activated.

Prior to activating a step, all other steps must 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.

To ensure that the executability (interlock) of all movements is displayed correctly, the **Permanent processing of all interlocks in manual mode** checkbox must be enabled for the sequencer FB.

Tips and tricks

The **LTLL_ManualGraph** block can be used for all operating screens. It must be called with different instance DBs and a unique **LTLL_ManualGraphConfig** must be created for each operating screen.

Use the following call sequence:

- I. LTLL_Basic
- II. LTLL_Manual
- III. LTLL_ManualGraph
- ...
- X. Sequencer FB (user-specific)

Parameterization of the sequencer DB name and the step number in LTLL_ManualGraphConfig

For each line in the movement screen which is to control Graph steps, the following variables must be parameterized in the **LTLL_ManualGraphConfig** data block:

Address:	LTLL_ManualGraphConfig.row[X].left.dbInstanceName (where X = the number of the line of the operating screen)
Format:	WString[125]
Range of values:	The instance data block name of the sequencer in which the corresponding line (X) of the step is to be activated when the left button is pressed. The name must be specified in double inverted commas.
Default setting:	-

Address:	LTLL_ManualGraphConfig.row[X].left.stepNumber (where X = the number of the line of the operating screen)
Format:	USInt
Range of values:	The step number of the step which is to be activated when pressing the left button of the corresponding line (X)
Default setting:	0

Address:	LTLL_ManualGraphConfig.row[X].right.dbInstanceName (where XX = the number of the line of the operating screen)
Format:	WString[125]
Range of values:	The instance data block name of the sequencer in which the corresponding line (X) of the step is to be activated when the right button is pressed. The name must be specified in double inverted commas.
Default setting:	-

Address:	LTLL_ManualGraphConfig.row[X].right.stepNumber (where X = the number of the line of the operating screen)
Format:	USInt
Range of values:	The step number of the step which is to be activated when pressing the right button of the corresponding line (X)
Default setting:	0

Configuration of the Graph sequencer(s)

You must route the prepared data between the blocks **LTLL_ManualGraph** and **LTLL_ManualGraphExt** in your Graph sequencer as follows:

1. Add a variable of the type **LTLL_ManualGraphExt** in the static parameters of the sequencer FB.

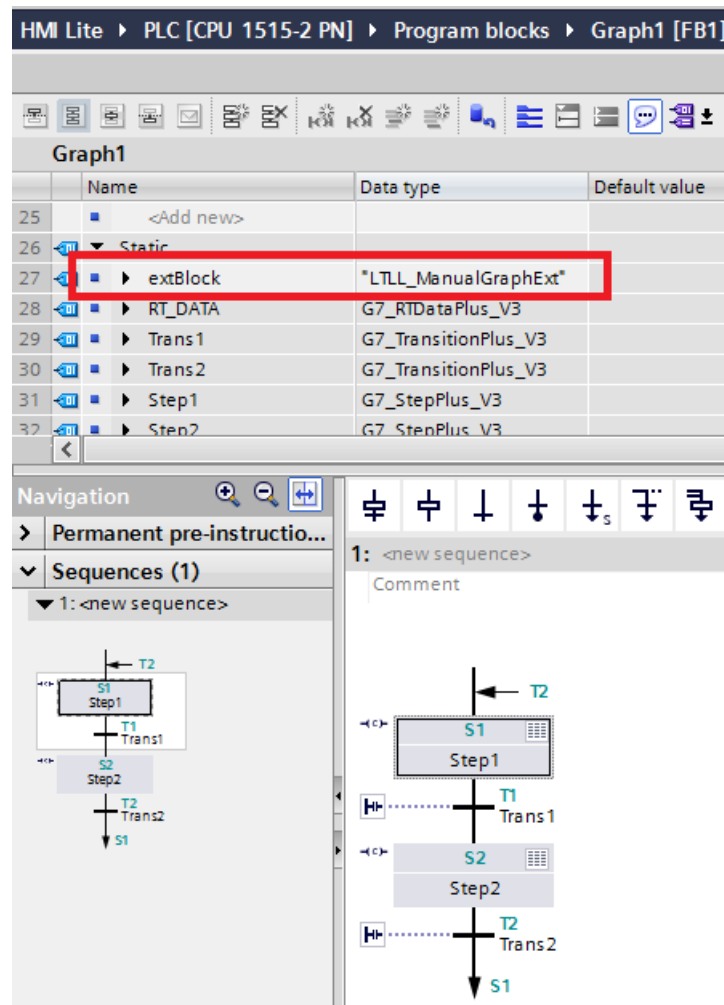


Fig. 6-14 Adding a variable of the type **LTLL_ManualGraphExt**

2. In the upstream permanent instructions of the Graph sequencer, route the data of the **LTLL_ManualGraphControl** into the static variable **Input**, of the variable of the type **LTLL_ManualGraphExt** created above. Make sure that you are using the correct **call** area:

call[0]	screenSetup (SS_02_001_Setup)
call[1]	screenPowerUp (SS_02_002_PowerUpCondition)
call[2]	screenUnit (SS_02_003_Unit)
call[3]	screenNutRunner (SS_02_004_NutRunner)
call[4]	screenNutRunnerGroup (SS_02_005_NutRunnerGroup)
call[5]	screenCycleType (SS_02_006_Cycletype)
call[6]	screenUserDefine (SS_02_007_UserDefine)

3. Call the instruction **GetInstanceName** and transfer the result to the variable of the type **LTLL_ManualGraphExt.input.instanceName**

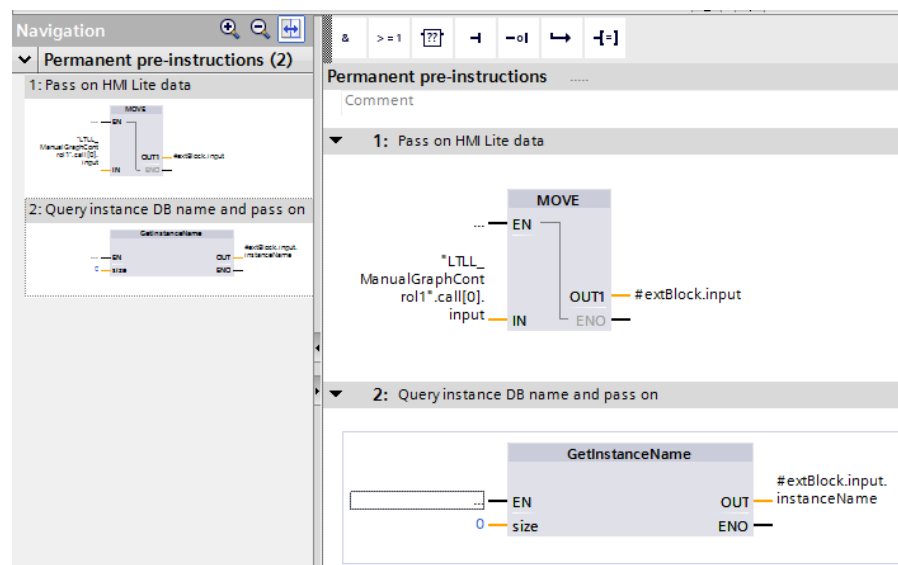


Fig. 6-15 Data transfer and call of the instruction **GetInstanceName**

4. Call the **LTLL_ManualGraphInterlock** block that switches the data of the tag of the type **LTLL_ManualGraphExt.output** to the **call[x].ouput** area of the **LTLL_ManualGraphControl** DB in the downstream permanent instructions of your GRAPH sequencer.

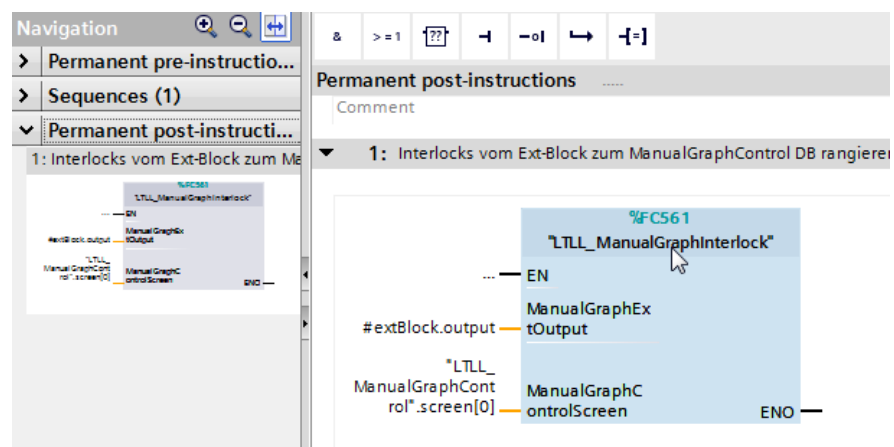


Fig. 6-16 Data transfer to **LTLL_ManualGraphControl.call[x].ouput**

5. Repeat the steps for all S7-GRAPH sequencers in which steps are controlled by the manual operating screens. ■

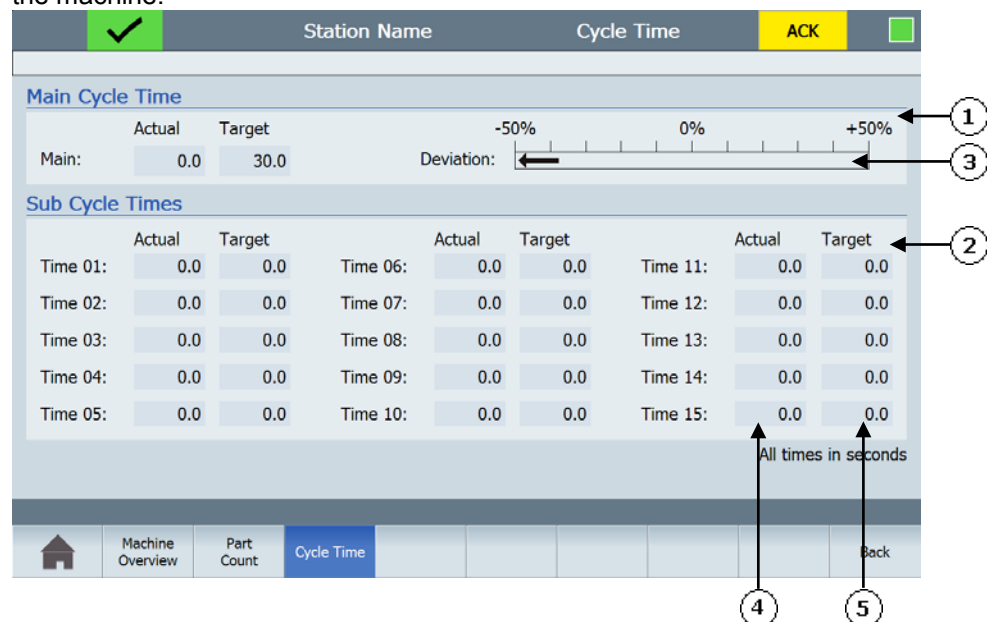
7

7 Production Data Screens

7.1 Cycle times screen

7.1.1 Layout 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 ($\pm 50\%$) from the set clock time
- (4) Actual clock time values
- (5) Set clock time values

Fig. 7-1 **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 $\pm 50\%$. 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:

$\text{deviation} = \text{actual clock time} / \text{set clock time}$

On the KTP900F Mobile, only the first 6 single clock times are displayed.

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 32-bit signed 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 214.748.364,7 seconds.

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 `LTL_Config.THIS[X].screenCycletime.hideSpecific` configuration parameter. In this case, the complete control field with the single clock times will be hidden from the display (see Fig. 7-1 **Clock times** screen Element (2)).

7.1.2 Runtime interface (LTLL_Cycletime)

The calculation of the clock times is realized with the **LTLL_Cycletime** block. 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.

Call interface

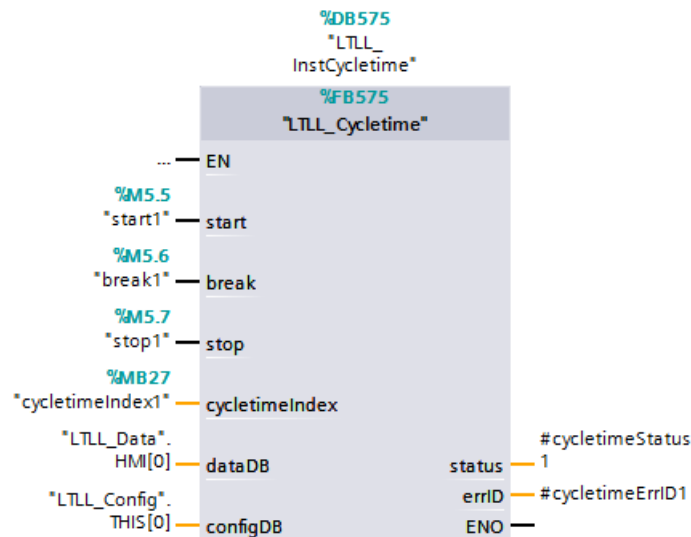


Fig. 7-2 Call interface of the **LTLL_Cycletime** block

Parameters

Table 7-1 Parameters of the **LTLL_Cycletime** block

Name	Type	Default	Example	Description
dataDB	LTLL_type Data	LTLL_Data. HMI[0]	LTLL_Data. HMI[0]	HMI Lite runtime DB
configDB	LTLL_type Config	LTLL_Config .THIS[0]	LTLL_Config .THIS[0]	HMI Lite configuration DB
start	BOOL	---	M0.5	A positive edge starts the clock time selected by the cycletimeIndex 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 cycletimeIndex parameter
cycletimeIndex	USInt	---	MB6	Select the clock time to be measured. cycletimeIndex = 0 means total clock time cycletimeIndex = 1 – 15 means single clock time 1 - 15
status	WORD	---	MW2	Block status
errId	WORD	-	-	Local error handling

Output parameter status

Value (W#16#)	Description
16#8200	HMI Lite licensing failed
16#8201	Invalid cycletimeIndex

Note

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

7.1.3 Configuration

LTLL_Config

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

Address	LTLL_Config.THIS[X].screenCycletime.hideSpecific
Format	BOOL
Range of values	FALSE = single clock times will be displayed TRUE = single clock times will be hidden
Default setting	FALSE

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

Address	LTLL_Config.THIS[X].screenCycletime.main.target
Format	BOOL
Range of values	-2_147_483_648 bis +2_147_483_647
Default setting	300 (30,0 Sekunden)

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

Address	LTLL_Config.THIS[X].screenCycletime.sub.target[XX] (where XX is the number of the corresponding single clock time: 1..15)
Format	DINT
Range of values	-2_147_483_648 bis +2_147_483_647
Default setting	300 (30,0 Sekunden)

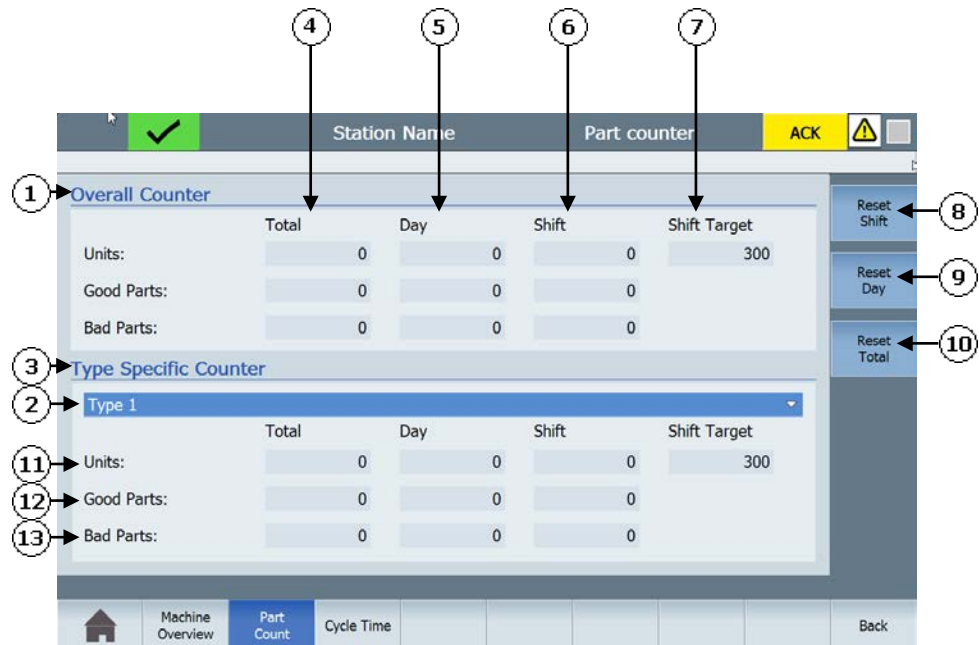
Configure **LTLL_Config** as follows:

1. Open the **LTLL_Config** data block.
2. Set the **screenCycletime.hideSpecific** variable to the value:
FALSE = when the single clock times are to be displayed
TRUE = when the single clock times are to be hidden
3. Edit the **screenCycletime.main.target** and **screenCycletime.sub.target[XX]** variables to define the values for the set clock times.
4. Download the data block to the controller.
5. Save and close the **LTLL_Config** data block.
6. Open the **LTLL_HMILite** organization block
7. Call **LTLL_Cycletime** and assign the required parameters.
8. Save and close the **LTLL_HMILite** block.
9. Load all changed blocks into the controller.

7.2 Workpiece counter screen

7.2.1 Layout 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

Fig. 7-3 **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 3500 workpiece-related parts counters can be used.

The text for the designation 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 **LTLL_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 **LTLL_Config**.

Pressing a RESET key initiates the provided confirmation procedure.

The screenshot shows the 'Part counter' screen with a 'Station Name' field and an 'ACK' button. Below this, there are two main sections: 'Overall Counter' and 'Type Specific Counter'. Each section contains a table with columns for 'Total', 'Day', 'Shift', and 'Shift Target'. The 'Overall Counter' table shows 'Units: 0', 'Good Parts: 0', 'Bad Parts: 0', and 'Shift Target: 300'. The 'Type Specific Counter' table shows 'Type 1' selected, with 'Units: 0', 'Good Parts: 0', 'Bad Parts: 0', and 'Shift Target: 300'. On the right side of the screen, there are five buttons: 'Reset Shift', 'Reset Day', 'Reset Total', 'Reset', and 'Cancel'. Numbered callouts indicate: (1) 'Reset Shift' button, (2) 'Reset' button, and (3) 'Cancel' button.

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

Fig. 7-4 **Workpiece counter** – procedure for confirming the reset

Range of values

Value ranges of the counters:

Total – sum of good/bad parts: 0 bis 18.446.744.073.709.551.615

Total - bad parts: 0 bis 4.294.967.295

Day – sum of good/bad parts: 0 bis 4.294.967.295

Day - bad parts: 0 bis 4.294.967.295

Shift – sum of good/bad parts: 0 bis 4.294.967.295

Shift - bad parts: 0 bis 4.294.967.295

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 **hideTypeSpecific** configuration parameter in the **LTLT_Config** configuration data block. Hiding applies to the complete Workpiece-related parts counter (see number (3) in Fig. 7-3 **Workpiece counter** screen). Configuration parameter **HIDE_TYPE_SPECIFIC** has no effect in the 9" variant.

7.2.2 Runtime interface (LTLL_Counter)

The **LTLL_Counter** block uses directly the unit counter variables in the **LTLL_Data** and **LTLL_CounterData** data blocks.
The user program can also access these variables (e.g. save the values for further processing or archiving before a RESET is performed).

Call interface

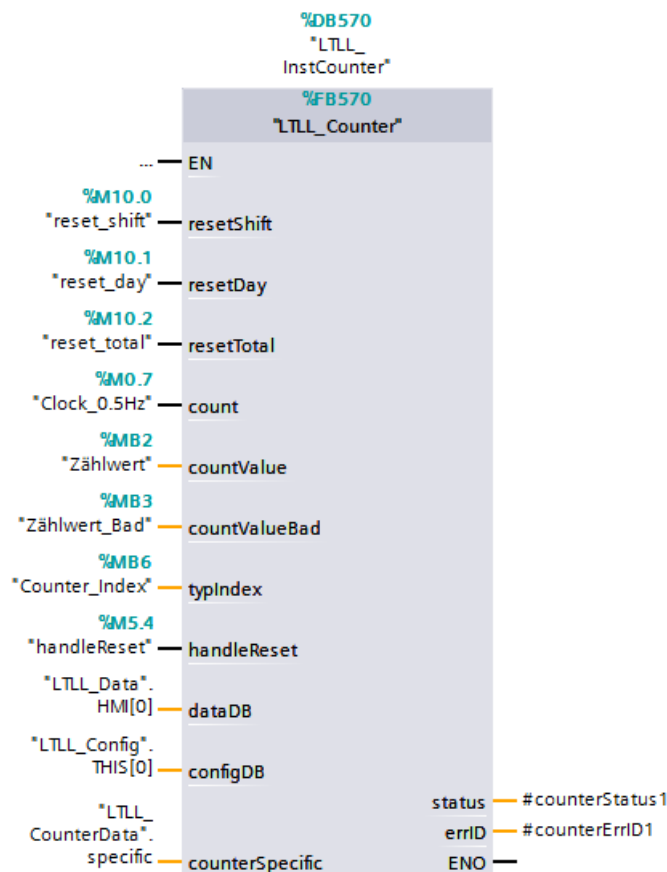


Fig. 7-5 Call interface of the **LTLL_Counter** block

Parameters

Table 7-2 Time parameters of the **LTLL_Counter** block

Name	Type	Default	Example	Description
dataDB	LTLL_typeData	LTLL_Data.HMI[0]	LTLL_Data.HMI[0]	HMI Lite DB
configDB	LTLL_typeConfig	LTLL_Config.THIS[0]	LTLL_Config.THIS[0]	HMI Lite configuration DB
resetShift	BOOL	---	M5.0	An increasing edge resets the shift counters
resetDay	BOOL	---	M5.1	An increasing edge resets the day counters
resetTotal	BOOL	---	M5.2	An increasing edge resets the total counters
count	BOOL	---	M5.3	An increasing edge updates the counter
countValue	BYTE	---	MB6	The number of the total parts to be counted (good + bad parts)
countValueBad	BYTE		MB7	Number of bad parts to be counted
typIndex	WORD	---	MW8	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 will cause an error message to be issued
HandleReset	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
CounterSpecific	Array[*] of LTLL_typeCounter	LTLL_Counter Data.specific	LTLL_Counter Data.specific	DB where the part counters are specified
status	WORD	---	MW2	Block status
errId	WORD	-	-	Local error handling

Output parameter status

Value (W#16#)	Description
16#0001	Only total parts counter
16#8200	HMI Lite licensing failed
16#8201	Invalid type index
16#8400	Invalid selection in the screen

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

$$[\text{new counter value}] = [\text{old counter value}] + [\text{counter value}].$$

The counter value is defined by the **countValue** (good and bad parts) and **countValueBad** (bad parts) parameters. The values for the **good parts** are calculated using the following equation:

$$[\text{good parts value}] = [\text{total parts value}] - [\text{bad parts value}].$$

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

The **typIndex** parameter specifies which workpiece-related counter will be updated. Up to 3500 workpiece-related counters can be selected. The workpiece-independent total counter is always updated. If a value 0 is specified for the **typIndex** parameter, only the total counter will be updated.

The reset parameters **resetDay**, **resetShift**, **resetTotal** always reset all workpiece-related counters and the total counter. For example, the **resetShift** 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 parameters 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 **resetShift** parameter can also be used when the **handleReset** 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.

7.2.3 Configuration

LTLL_Config

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

Address	LTLL_Config.THIS[X]. screenCounter.hideTypeSpecific
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	LTLL_CONFIG.THIS[X]. screenCounter.hideResetShift LTLL_CONFIG.THIS[X]. screenCounter.hideResetDay LTLL_CONFIG.THIS[X]. screenCounter.hideResetTotal
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 **LTLL_Config.THIS[X]** und **LTLL_CounterData**. 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	LTLL_CONFIG.THIS[X]. screenCounter.overall.shiftTarget LTLL_CounterData.specific[X].config.shiftTarget
Format	UDINT
Range of values	0 bis 4.294.967.295
Default setting	300

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	LTLL_CONFIG.THIS[X]. screenCounter.TimeValueHideReset
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

This **SO_04_011_PartCounterType** text list contains the designations of the workpiece types to be displayed in the selection window.

Table 7-3 WinCC text list **SO_04_011_PartCounterType**

Text list		SO_04_011_PartCounterType
Display		Text
Format		Decimal
Value	1	Workpiece 1 designation
Value	2	Workpiece 2 designation
...	...	...
Value	3500	Workpiece 3500 designation

Configure the text list as follows:

1. Open the **LTLL_Config** data block.
2. Set the **screenCounter.hideTypeSpecific** 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
LTLL_Config.THIS[X].screenCounter.overall.shiftTarget (total counter)
and **LTLL_CounterData.specific[x].shiftTarget**.
4. Download the data block to the controller.
5. Save and close the data block.
6. Open the **LTLL_HMILite** organization block.
7. Call **LTLL_Counter** block and assign the required parameters.
8. Save and close the **LTLL_HMILite** block.
9. Load all changed blocks into the controller.
10. Open the WinCC file in HMI Lite via WinCC.
11. Edit the **SO_04_011_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 compile the WinCC 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 LTLL_Counter block:
countValue: Total count of the parts to be produced per pulse (good and bad parts)
countValueBad: Bad parts to be counted
typIndex: Index of the workpiece type to be counted; if only total counter, then "0"
count: Counting pulse (increasing edge $0 \Rightarrow 1$)
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.



Notes

8

8 Diagnostics

8.1 Alarm screen and Alarm buffer screen

8.1.1 Layout and functionality

The alarm screens display the alarm messages, general messages and system messages in tabular form. The **Alarm** screen displays the messages and alarms currently pending. The **Alarm buffer** 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 **Alarm** and **Alarm buffer** screens is identical.

Station Name		Message buffer		ACK	
No.	Date	Time	Status	Text	
110001	2016-12-15	12:24:15	PMC	Change to operating mode 'online'.	
140000	2016-12-15	12:24:15	PMC	Connection established: PLC_0, Station 192.168.133.15, Rack 0, Slot 1.	
270006	2016-12-15	12:24:06	PMC	Project modified: Alarms cannot be restored from the persistent alarm buffer.	

Fig. 8-1 Alarm and Alarm buffer screen SS_03_001_Alarm and SS_03_002_AlarmHistory

The following information is displayed in tabular form:

- message number
- time stamp of the message
- message status (K: arrived, G: sent, A: acknowledged)
- message text

8.1.2 Runtime interface

The interface data block is defined by the WinCC **SO_00_000_fault** variable for fault messages and **SO_00_000_warnings** variable for operational messages.

8.1.3 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.

Integrating the PLC code display

When using a Graph or ProDiag overview object, you can use the preconfigured screen **SS_10_013_PlcCodeViewer**.

- For this, move the screen **SS_10_013_PlcCodeViewer** from the project library into the project navigation.
 - Path in the project library for TP1200 Comfort:
[Project library]/[Types]/HMI Lite_TP1200 Comfort/Diagnostic/SS_10_013_PlcCodeViewer
 - Path in the project library for KTP900F Mobile:
[Project library]/[Types]/HMI Lite_KTP900F Mobile/Diagnostic/SS_10_013_PlcCodeViewer

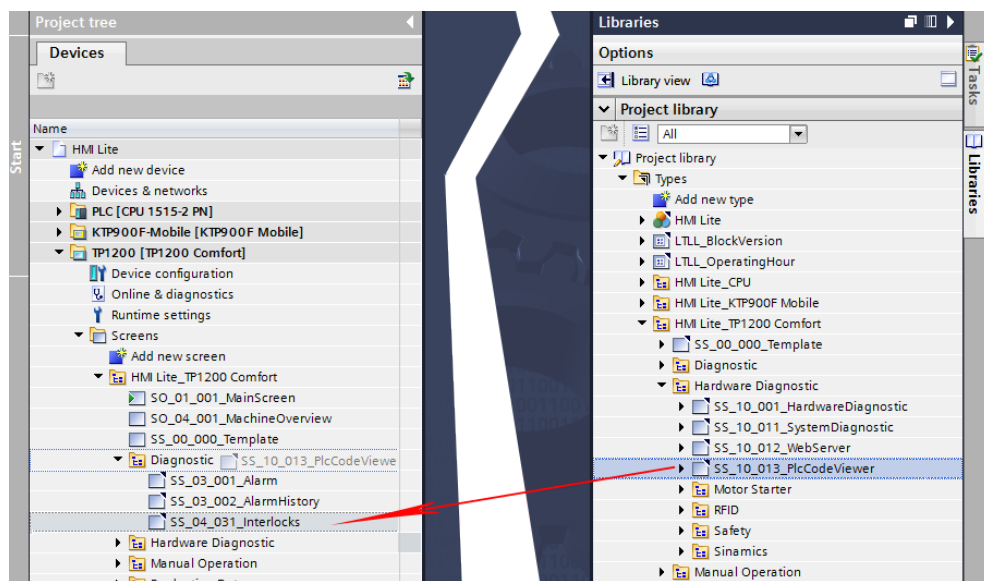


Fig. 8-2 Integrating the PLC code display

2. Open the screen **SS_03_001_Alarm** and insert the following function in the results of the button selection **Activate PLC Code Viewer: Activate PLC Code Display** with the parameters:
 - Screen name: SS_10_013_PlcCodeViewer
 - Screen object: plcCodeViewer

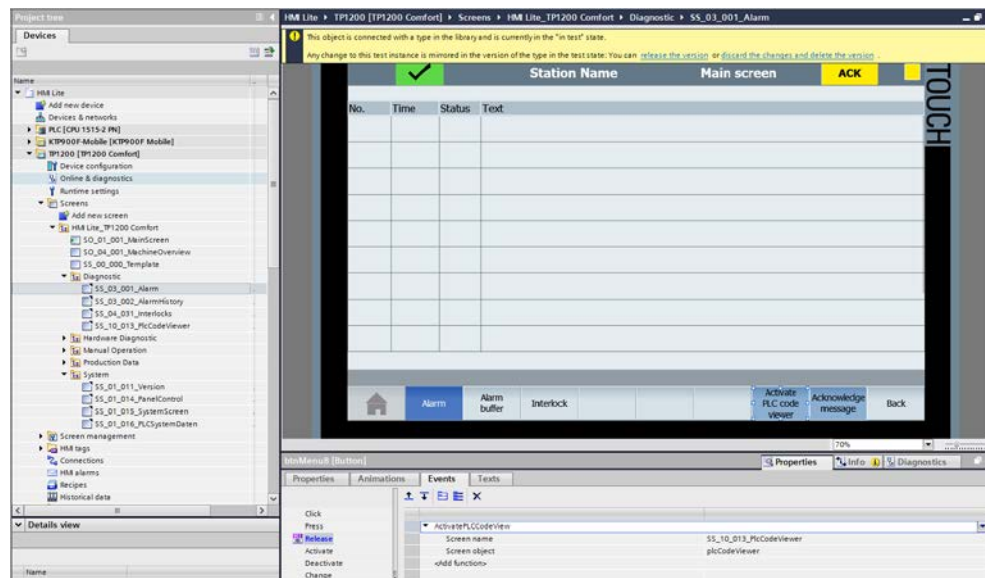
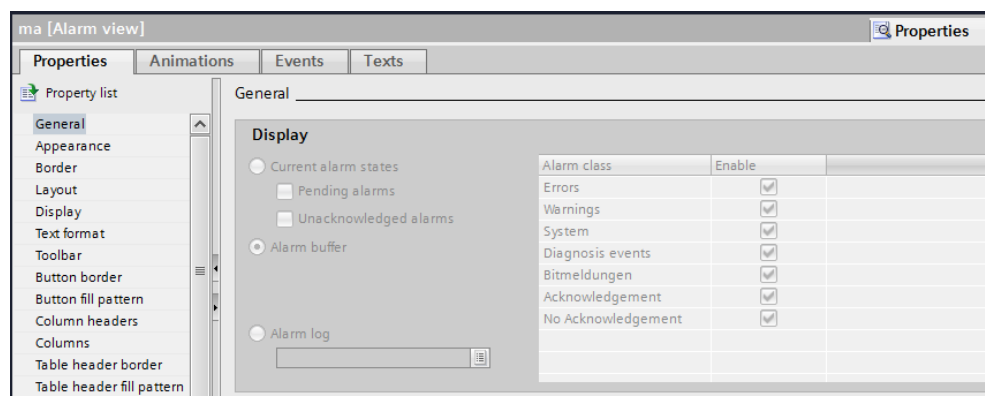


Fig. 8-3 PLCCodeViewer activation

Settings of the alarm buffer

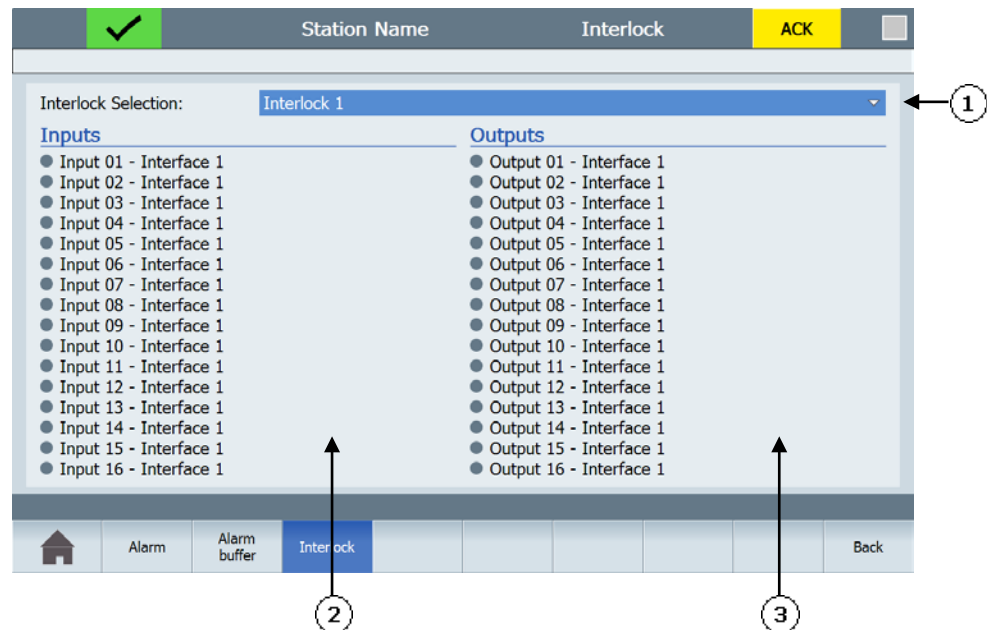
The **alarm buffer** screen displays selected message events from the alarm buffer. The configuration specifies which events are displayed. The alarm window shows the alarm events selected in the properties dialog.

Fig. 8-4 Setting for the alarm display object in the **Alarm buffer** screen

8.2 Interface screen

8.2.1 Layout and functionality

The **Interface** screen can be used to display for diagnostic purposes the signals transferred between the controller and the external devices. Up to 218 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 list box.



- (1) List box for selecting the interface
- (2) Display of the states of the inputs
- (3) Display of the states of the outputs

Fig. 8-5 Interface screen **SS_04_031_Interlocks**

8.2.2 Runtime interface

The runtime interface for the **Interface** screen consists of three variables. The **LTLL_Data.HMI[X].screenInterlock.selection** variable represents the current interface selected from the list box.

Address	LTLL_Data.HMI[X].screenInterlock.selection
Format	INT
Range of values	1..218
Default setting	1

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

Address	LTLL_Data.HMI[X].screenInterlock.signals.inputs
Format	WORD
Range of values	The status of each bit is displayed in the screen by the associated LED element.
Default setting	-

Address	LTLL_Data.HMI[X].screenInterlock.signals.outputs
Format	WORD
Range of values	The status of each bit is displayed in the screen by the associated LED element.
Default setting	-

8.2.3 Configuration

Up to 218 interface descriptions can be defined and selected from a list box. A WinCC text list can be used to specify a name for each of these interfaces:

Table 8-1 Selection window for the interlocks – screen caption of the text list

Text list		SO_04_031_InterlockSelection
Display		Text
Format		Decimal
Value	01	Name for interface no. 1
Value	02	Name for interface no. 2
		...
Value	16	Name for interface no. 16

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

Table 8-2 Designation of the input/outputs

Text list		SO_04_031_InterlocksInputs SO_04_031_InterlocksOutputs
Display		Text
Format		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



9

9 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.



Fig. 9-1 **Hardware Diagnostics** screen

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

9.1 System diagnostics

The HMI Lite screen **System diagnostics** uses the WinCC standard control **System diagnostics view**.

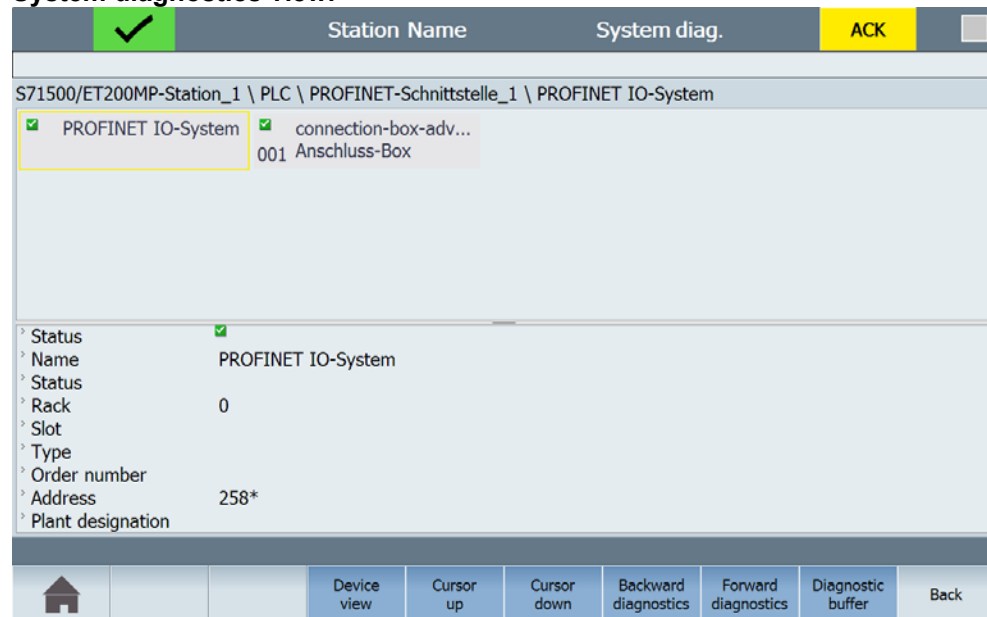


Fig. 9-2: **System diagnostics** screen

9.2 Webserver

The HMI Lite **Webserver** screen includes an HTML browser object.

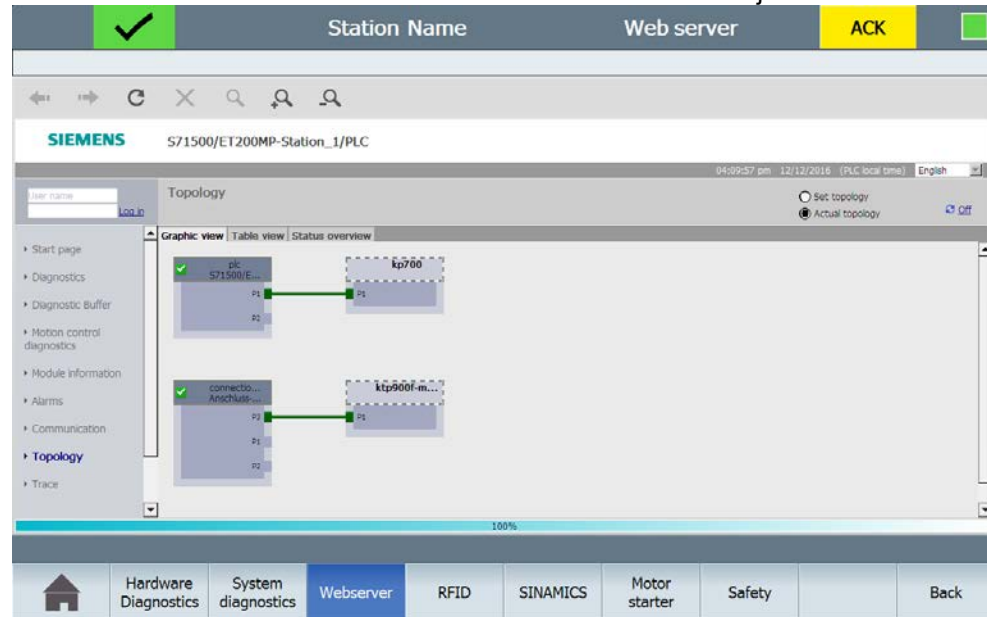


Fig. 9-3 **WebServer** screen

During startup of the PCL, the **LTLL_Basic** block reads the Webserver URL and stores it in the **LTLL_Data.HMI[X].global.readOnly.webServerAddress** variable.

If the object is not correctly displayed, you can change the URL in the **General** tab under **Properties**.

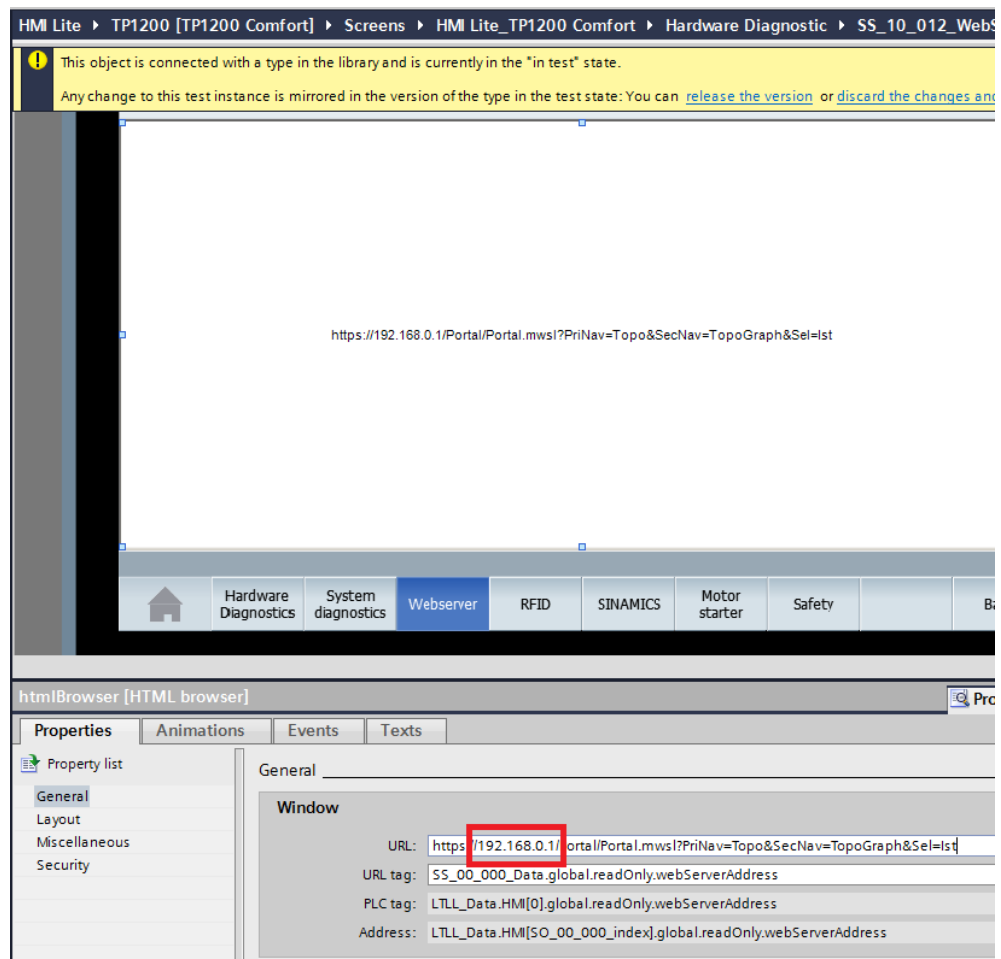


Fig. 9-4 Webserver: Changing URL

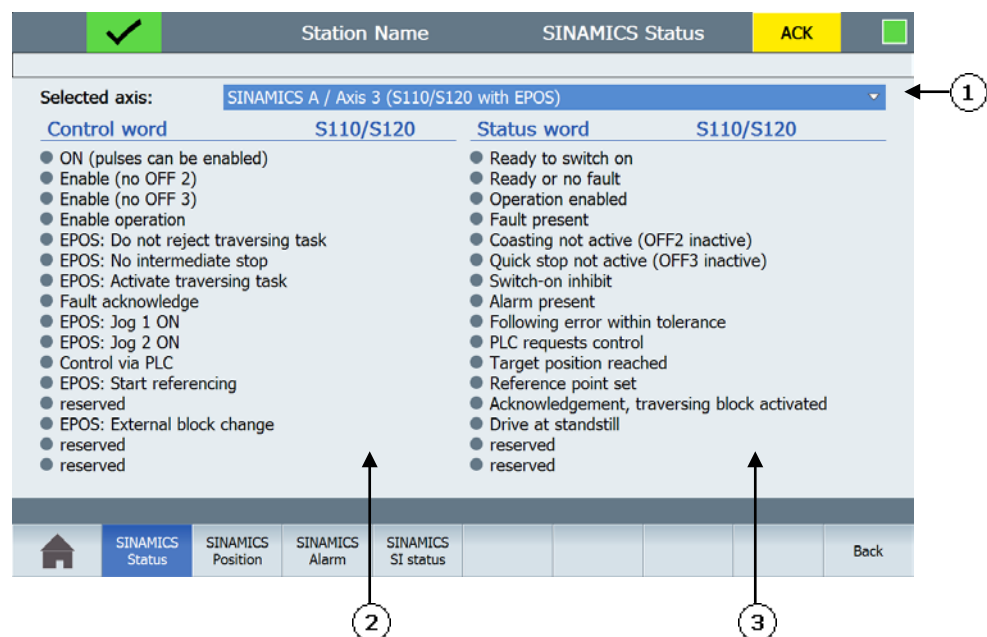
9.3 SINAMICS diagnostics

HMI Lite includes SINAMICS diagnostics for the following drives and variants:

- SINAMICS S110
- SINAMICS S120
- SINAMICS G110
- SINAMICS G120

9.3.1 SINAMICS status screen

The **SINAMICS status** screen shows the control and status signals of the SINAMICS axis selected from the list box.

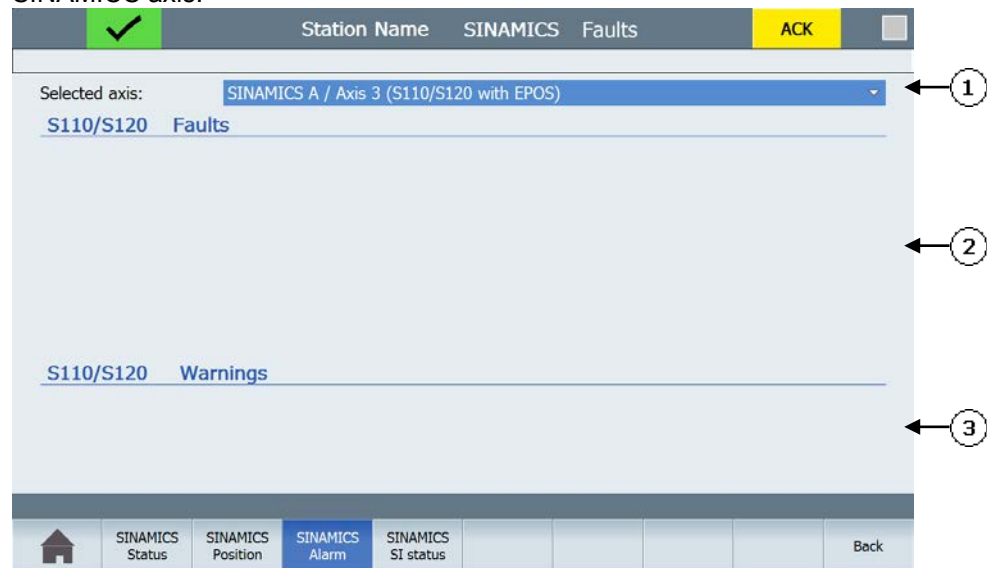


- (1) Selection of the axis
- (2) Control signals of the selected axis
- (3) Status signals of the selected axis

Fig. 9-5 SINAMICS status

9.3.2 SINAMICS alarms screen

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

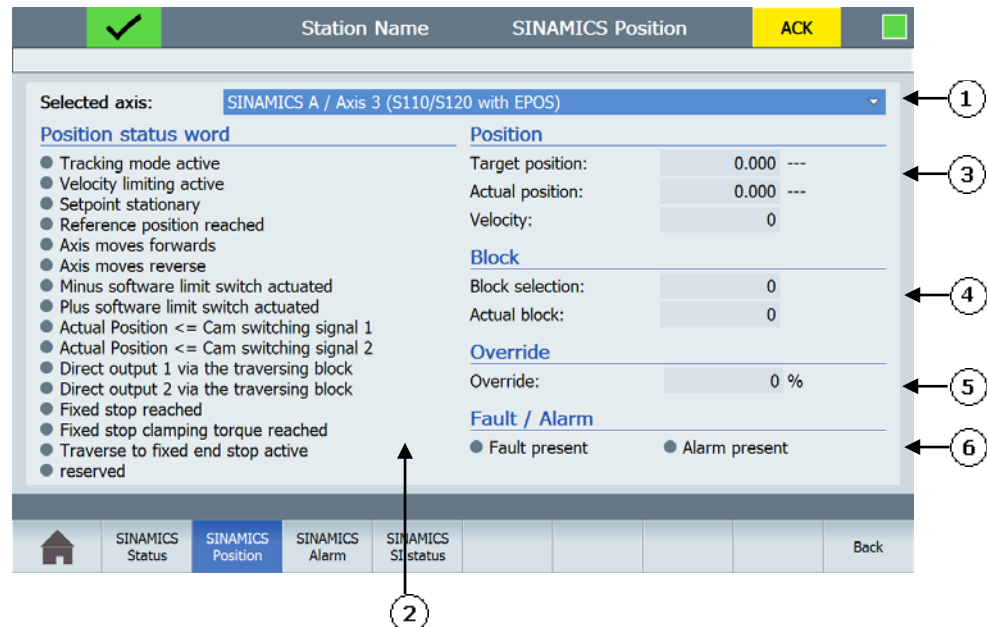


- (1) Selection of the axis
- (2) Area for displaying the faults
- (3) Area for displaying the warnings

Fig. 9-6 SINAMICS alarms

9.3.3 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
- (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

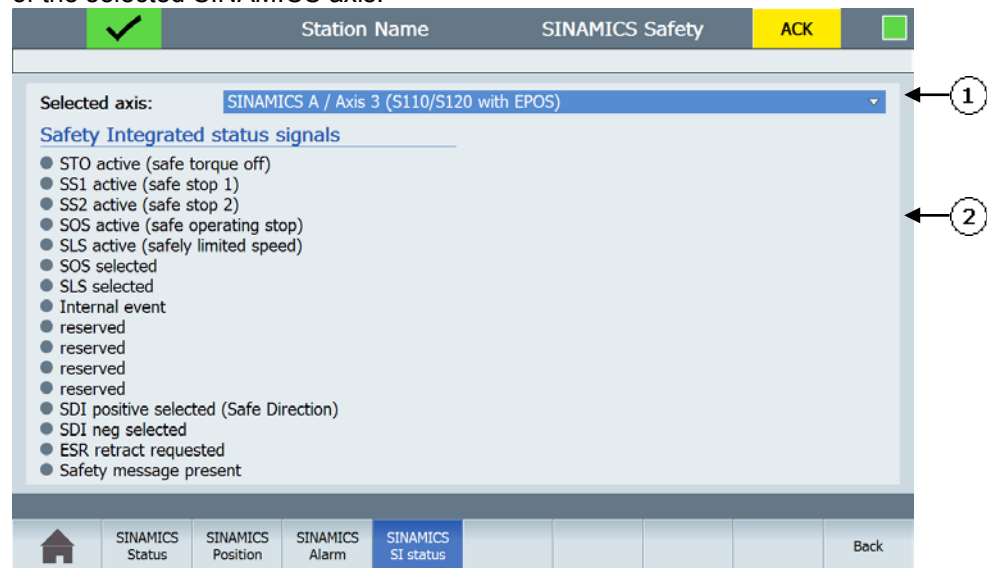
Fig. 9-7 SINAMICS position

Note

The **SINAMICS position** screen does not contain any information when a SINAMICS S120 without positioning functionality or SINAMICS G120 is used.

9.3.4 SINAMICS Safety Integrated Status screen

The **SINAMICS Safety Integrated Status** screen displays the safety status signals of the selected SINAMICS axis.



- (1) Selection of the axis (selection window)
(2) Safety Integrated Status signals of the selected axis

Fig. 9-8 SINAMICS Safety Integrated Status

9.3.5 **Configuring of the WinCC screens**

Configuring of the text list in WinCC

For each configured designation in the text list there must be a drive object configured in the **LTLL_SinamicsCFG** block.
The designations of the text list entry can be chosen freely.
The value of the text list entry must match index of the drive object in the **LTLL_SinamicsCFG** data block. The text list entry is assigned to the drive object through the value.
The **SO_11_000_SinamicsAxis** text list has the following structure:

Table 9-1 Text list for the axis designations

Text list		SO_11_000_SinamicsAxis
Display		Text
Format		Decimal
Value	2	Designation of the first axis (value = index of the driveObject in LTLL_SinamicsCFG)
Value	3	Designation of the second axis (value = index of the driveObject in LTLL_SinamicsCFG)
...	...	...

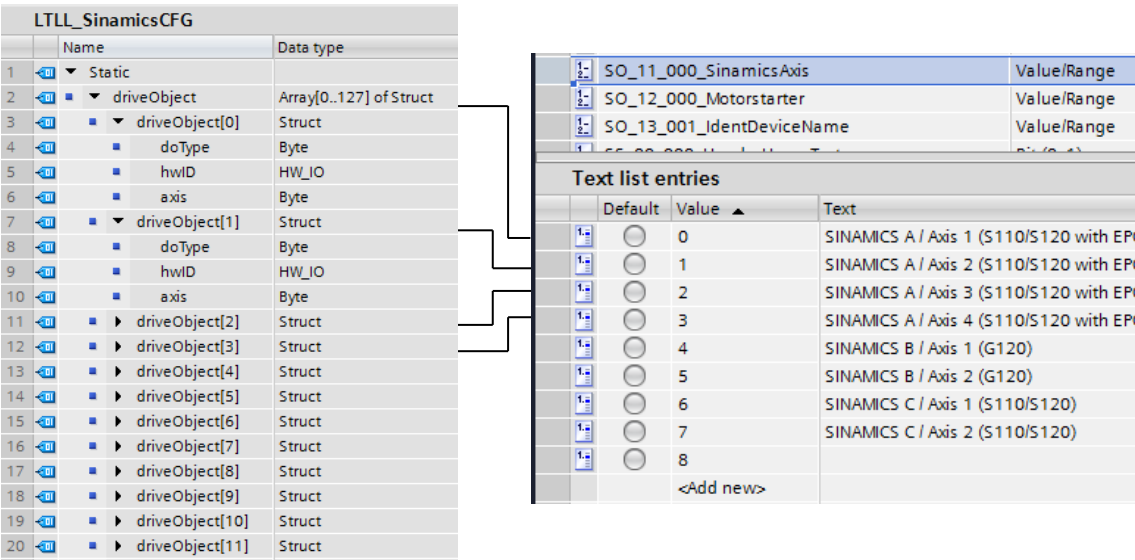


Fig. 9-9 How to assign a text list to a drive object

!

Important

The designation text for missing axes must be deleted!

The value of the text list entry must match the drive object index (order) in **SO_11_000_SinamicsAxis**.

9.3.6 Configure the drive object (LTLL_SinamicsCFG)

Each text list entry in the text list is assigned through value to a drive object which is configured in the **LTLL_SinamicsCFG** data block. A drive object is configured in a structure:

Table 9-2 Drive object structure in data block **LTLL_SinamicsCFG**

Name	Type	Description
do_Type	BYTE	Drive object type 0 = SINAMICS S110/S120 with positioning functionality (EPOS) 1 = SINAMICS S110/S120 without positioning functionality (EPOS) 2 = SINAMICS G120
hwID	INT	Diagnostic address of the DP-Slave from "Devices & Networks"
axis	BYTE	Drive object ID

9.3.7 Runtime interface (LTLL_Sinamics)

The **LTLL_Sinamics** supplies the WinCC 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 called once cyclically. The **driveEnable** parameter must be used to enable the FB call.

Call interface

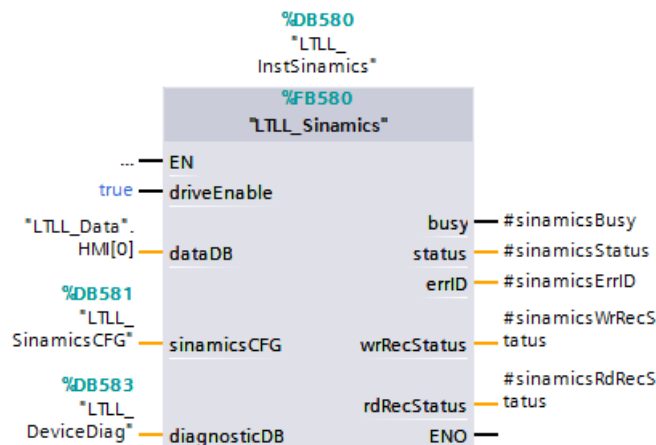


Fig. 9-10 Call interface of the **LTLL_Sinamics** blocks

Parameters

Table 9-3 Parameters of the **LTLL_Sinamics** block

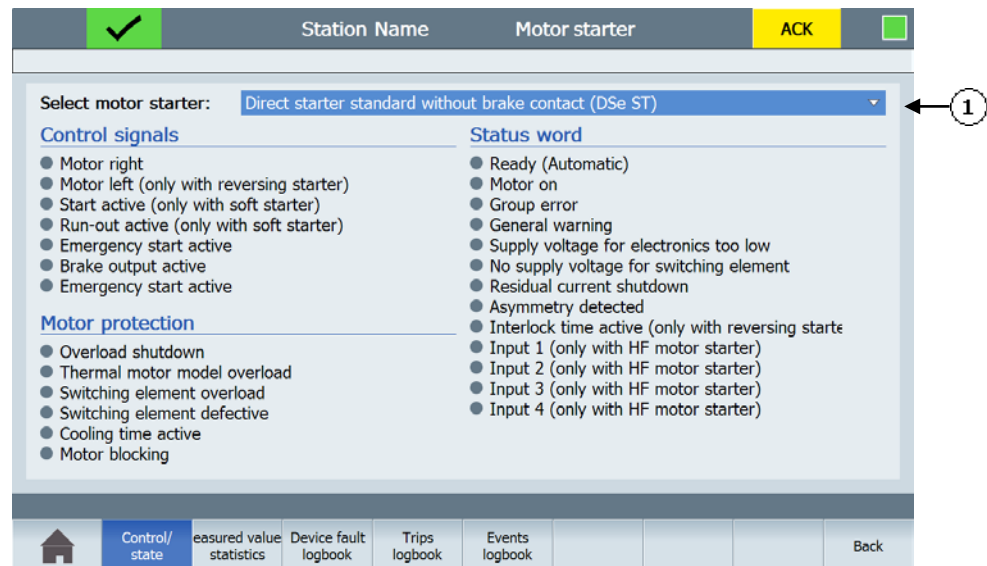
Name	Type	Default	Example	Description
dataDB	LTLL_typeData	LTLL_Data.HMI[0]	LTLL_Data.HMI[0]	Number of the HMI runtime data block
diagnosticDB	LTLL_typeDeviceDiag	LTLL_DeviceDiag	LTLL_DeviceDiag	Number of the HMI diagnostic data block
sinamicsCFG	LTLL_typeSinamicsCFG	LTLL_SinamicsCFG	LTLL_SinamicsCFG	HMI Lite block which contains the configuration of the drive objects
driveEnable	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
errId	Word	-	-	Local error handling
wrRecStatus	DWord	-	-	Status of the WRREC instruction
rdRecStatus	DWord	-	-	Status of the RDREC instruction

Status information parameter status

Value (W#16#)	Description
16#8200	HMI Lite licensing failed
16#8201	Invalid drive object
16#8600	Error in WRREC instruction
16#8601	Error in RDREC instruction
16#8602	Invalid order reference

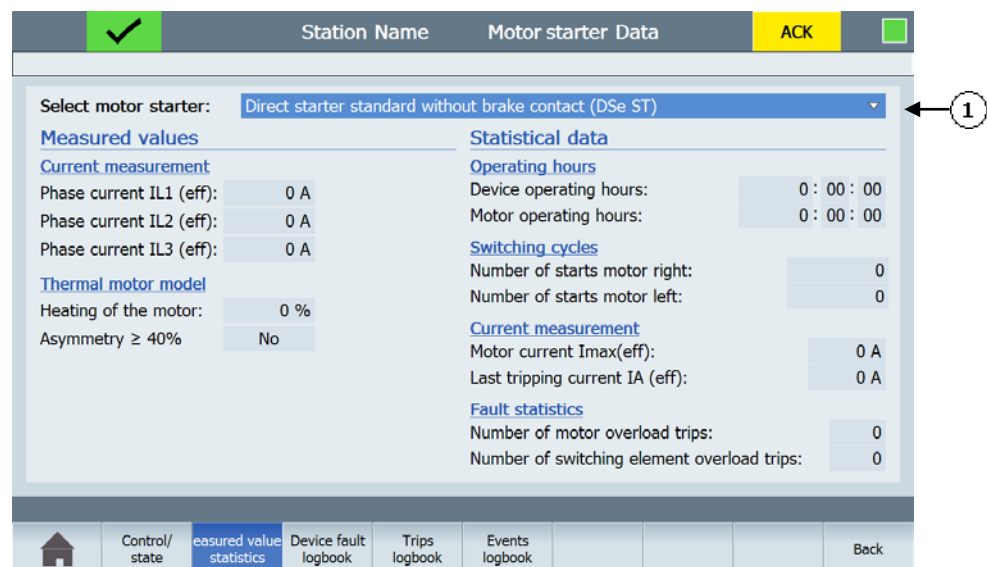
9.4 Motor starter diagnostics

The Motor starter diagnostics consists of the two diagnostics screen control signals, motor protection, status word (Control/status) and measured values, statistical data (Measured/statistic) and three logbook screens(Device error logbook, trips logbook, events logbook).



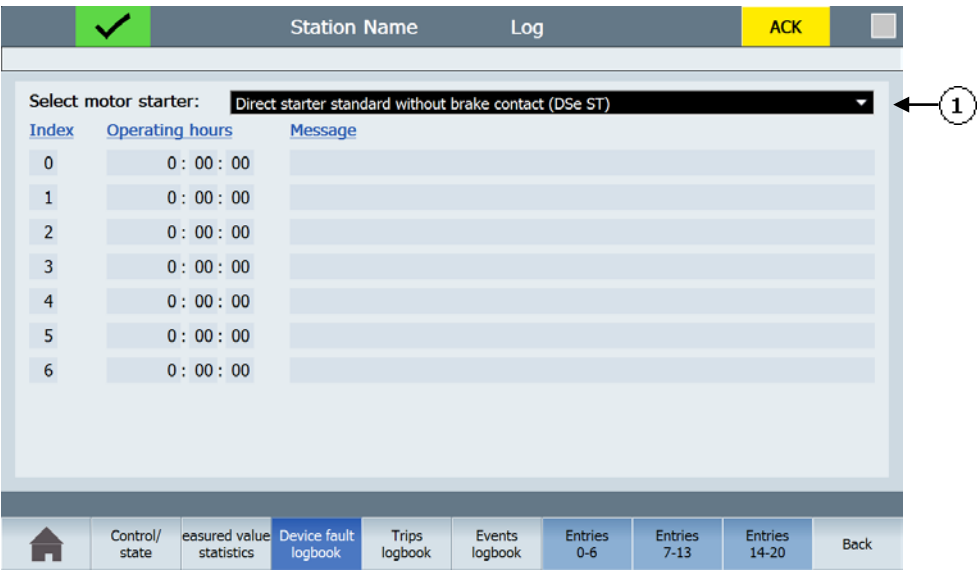
(1) Selection of the motor starter

Fig. 9-11 Motor starter: Control signals, motor protection, status word



(1) Selection of the motor starter

Fig. 9-12 Motor starter: Measured values, statistical data



(1) Selection of the motor starter

Fig. 9-13 Motor starter: Example for logbook: Device error logbook

9.4.1 Runtime interface (LTLL_Motorstarter)

The **LTLL_Motorstarter** block supplies the WinCC screens for the Motor starter Diagnostics. You must call the block once cyclically.

Call interface

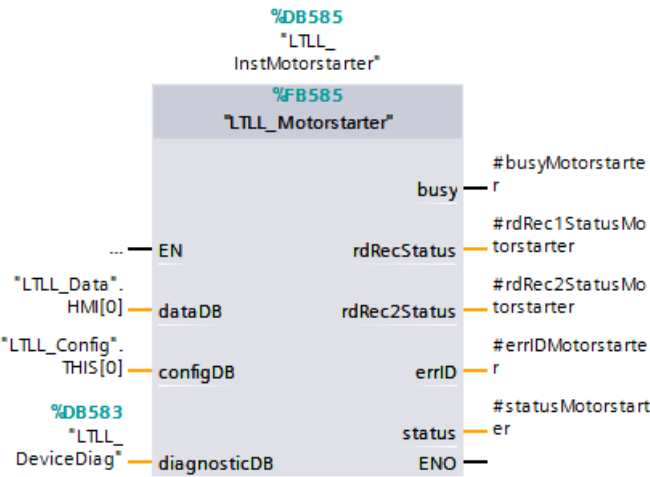


Fig. 9-14 Call interface of the **LTLL_Motorstarter** block

Parameters

Table 9-4 Parameters of the **LTLL_Motorstarter** block

Name	Type	Default	Example	Description
dataDB	LTLL_typeData	LTLL_Data.HMI[0]	LTLL_Data.HMI[0]	HMI Lite runtime DB
configDB	LTLL_typeConfig	LTLL_Config.THIS[0]	LTLL_Config.THIS[0]	HMI Lite configuration DB
diagnosticDB	LTLL_typeDeviceDiag	LTLL_DeviceDiag	LTLL_DeviceDiag	Number of the HMI Lite diagnostics DB
busy	Bool	-	-	Job is running
rdRecStatus	DWord	---	---	Status of RDREC1
rdRec2Status	DWord	-	-	Status of RDREC2
status	Word	---	-	Block status
errId	Word	-	-	Local error handling

Output parameter status

Value (W#16#)	Description
16#8200	HMI Lite licensing failed
16#8600	Error in RDREC1
16#8601	Error in RDREC2

Configuring in LTLL_Config

In the **LTLL_Config** block, you must specify the hardware address of the motor starter you wish to diagnose. You can enter up to 128 motor starters. The index of the field corresponds to the index of the text list in WinCC.

Address	LTLL_Config.THIS[X].screenMotorstarter.hwID[Y] (Y: number of the selected motor starter in the screen)
Format	HW_IO
Range of values	-
Default setting	-

Configuring the text list in WinCC

The designations of the motor starter devices are configured in the WinCC text list **SO_12_000_Motorstarter**. For each motor starter to be diagnosed, an entry in the text list must exist.

The **SO_12_000_Motorstarter** text list has the following structure:

Table 9-5 Text list for the designations of the motor starters

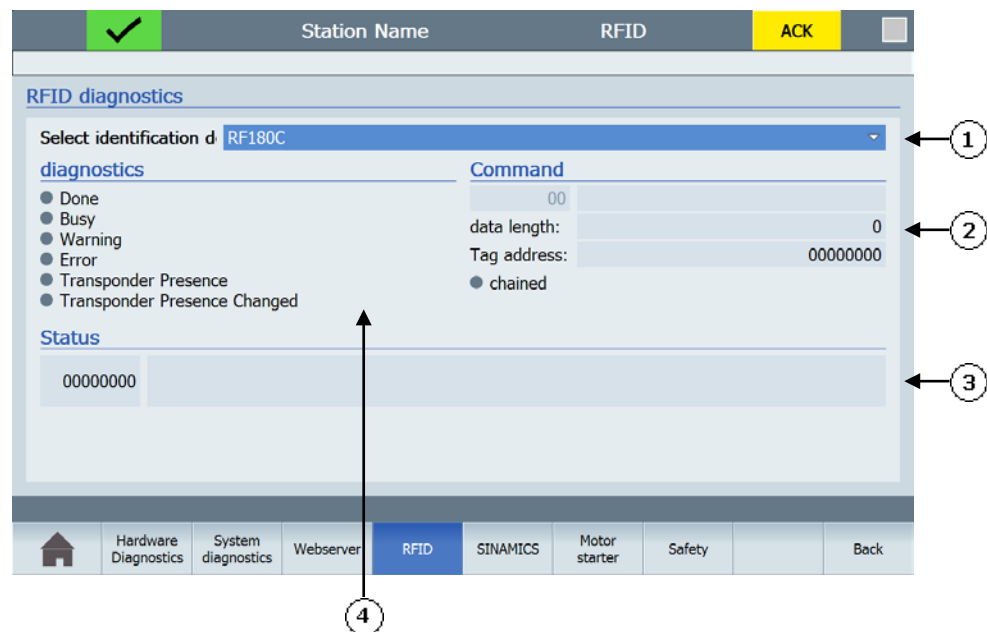
Text list		SO_12_000_Motorstarter
Display		Display
Format		Format
Value	Value	Designation of the first motor starter
Value	Value	Designation of the second motor starter
...	...	...

9.5 RFID diagnostics

The **RF300 diagnostics** screen shows the status signals and error messages of an ident device. The data must be transferred to the **LTLL_RFID** block as input parameters.

The signals and error messages from several ident devices can be displayed in the screen.

9.5.1 Layout and functionality



- (1) Selection of the ident device
- (2) Command area
- (3) Error messages of the ident device
- (4) Diagnostic bits

Fig. 9-15 RFID diagnostics

Selection of the ident device

Select an ident device in the list box. Each ident device provides its own interface.

Diagnostics

The diagnostic bits show the status of the selected ident device.

Command

In this area, data of the current command are displayed.

Status

In this area, error messages of the selected ident device are displayed.

9.5.2 Supported ident devices

The following ident devices are supported:

- ASM 450
- SIMATIC RF120C
- SIMATIC RF170C
- SIMATIC RF180C
- SIMATIC RF680R
- SIMATIC RF685R
- SIMATIC MV420
- SIMATIC MV440

The data exchange between the controller and the ident devices is performed using the ident blocks or the ident profile.

9.5.3 Configuring the WinCC screens

Configuring the text list in WinCC

The designations of the ident devices must be specified. The text items are stored in the WinCC **SO_13_001_IdentDeviceName** text list. For each configured ident device an entry in the text list must exist.

The **SO_13_001_IdentDeviceName** text list has the following structure:

Table 9-6 Text list for the designations of the ident devices

Text list		SO_13_001_IdentDeviceName
Display		Text
Format		Decimal
Value	1	Designation of the first ident device
Value	2	Designation of the second ident device
...	...	...



Important

The text items for non-configured (unused) ident devices must be deleted.

9.5.4 Runtime interface (LTLL_RFID)

The **LTLL_RFID** block supplies data to the WinCC screens for the RFID diagnostics. The displayed data is read from the interface.
The user must call this function once cyclically for each configured ident device, where **selectedDevice** parameter matches the corresponding values from the WinCC **SO_13_001_IdentDeviceName** text list (see sample program).

Call interface

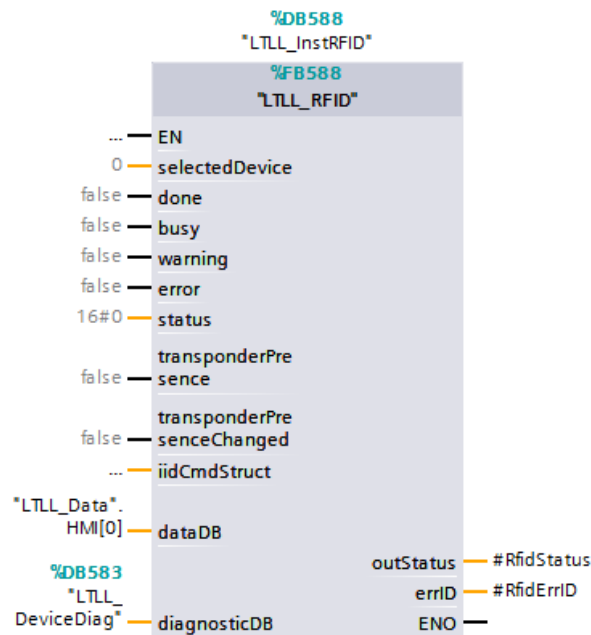


Fig. 9-16 Call interface of the **LTLL_RFID** block

Parameters

Table 9-7 Parameter of the **LTLL_RFID** block

Name	Type	Default	Example	Description
dataDB	LTLL_typeData	LTLL_Data.HMI[0]	LTLL_Data.HMI[0]	HMI Lite runtime DB
diagnosticDB	LTLL_typeDeviceDiag	LTLL_DeviceDiag	LTLL_DeviceDiag	The number of the HMI diagnostic data block
selectedDevice	UInt	-	-	Selection of the SLG the value comes from the SO_13_001_IdentDeviceName text list
done	Bool	---	---	Output parameter done of the ident blocks or the ident profile
busy	Bool	-	-	Output parameter busy of the ident blocks or the ident profile
warning	Bool	---	---	Output parameter warning of the ident blocks or the ident profile
error	Bool	-	-	Output parameter error of the ident blocks or the ident profile
status	DWord	-	-	Output parameter status of the ident blocks or the ident profile
transponderPresence	Bool	-	-	Output parameter presence of the ident blocks or the ident profile
transponderPresenceChanged	Bool	-	-	Output parameter tpc of the ident blocks or the ident profile
iidCmdStruct	IID_CMD_STRUCT	-	-	Input parameter current command of the ident blocks or the ident profile
outStatus	WORD	---	-	Block status
errId	Word	-	-	Local error handling

Status information of parameter status

Value (W#16#)	Description
16#0001	Device not selected in the screen
16#8200	HMI Lite licensing failed

9.6 Safety screen

In the upper section of the screen, the current generation time and the current collective signature as well as the generation time and collective signature at the acceptance of the safety program are displayed. If the current collective signature differs from the accepted collective signature, this can be seen on the screen. The mode of the safety operation is also displayed. The cycle times of the parameterized runtime group are displayed in the lower section.

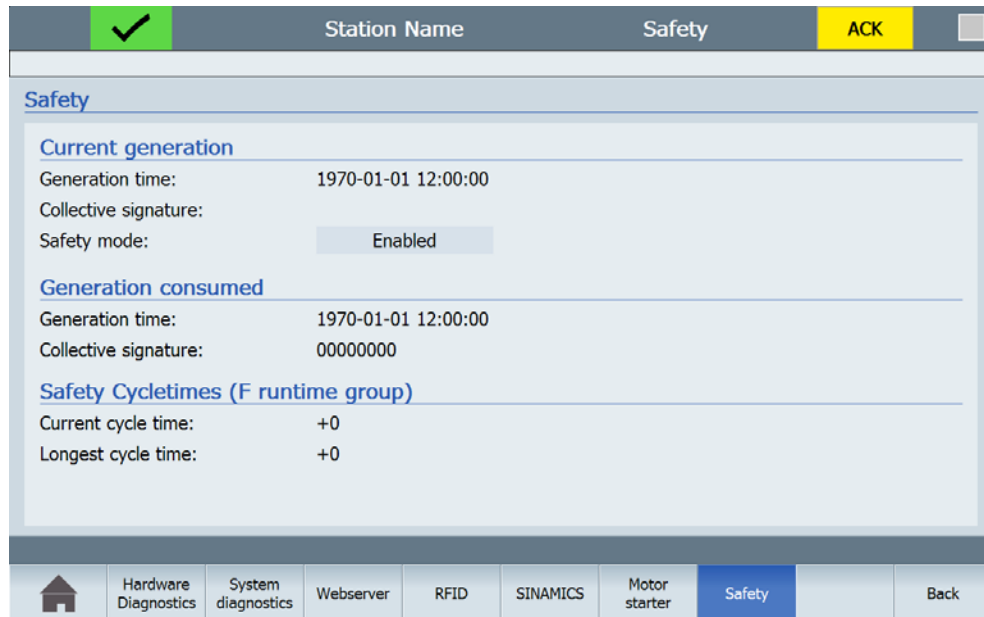


Fig. 9-17 Safety screen

Configuration



Important

If you are not using an F-PLC or a safety program:

Delete the call of the LTLL_Safety block and the block from your project. The block must be retained in the project library

At the **fSysInfo** input parameter, transfer the **F_SYSINFO** parameter of the F-runtime group info DB (**Program blocks → System blocks → STEP 7 safety**). You can set the current safety data as acceptance data via the **set** input parameter.

Call interface

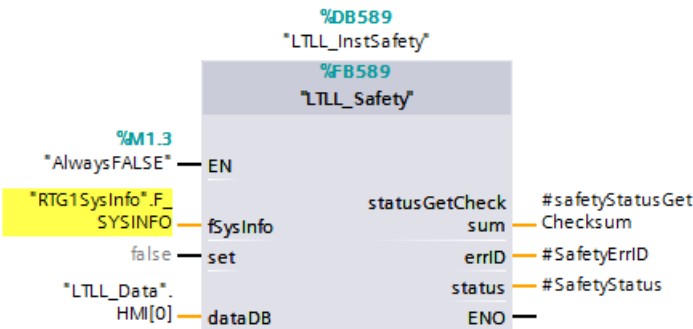


Fig. 9-18 Call interface of the **LTLL_Safety** block

Parameters

Table 9-8 Parameter of the **LTLL_Safety** block

Name	Type	Default	Example	Description
dataDB	LTLL_typeData	LTLL_Data.HMI[0]	LTLL_Data.HMI[0]	HMI Lite runtime DB
fSysInfo	F_SYSINFO	RTG1SysInfo.F_SYSINFO	RTG1SysInfo.F_SYSINFO	S_SYSINFO parameter of the F-runtime group info DB
set	Bool	-	-	Positive edge copies the current data to the acceptance data
statusGetChecksum	Word	-	-	Status of the GetChecksum command
status	Word	---	-	16#8200 HMI Lite licensing failed
errId	Word	-	-	Local error handling



Notes

10

10 System Screens

10.1 Version screen

The **Version** screen displays for diagnostic purposes the associated version of the WinCC screens, the data blocks, the functions and the function blocks of HMI Lite. In addition, the licensing state of HMI Lite is displayed.

The **Software version** button is used to display a window with the version of the WinCC runtime system files (not available for 9" devices!).

No configuring required.

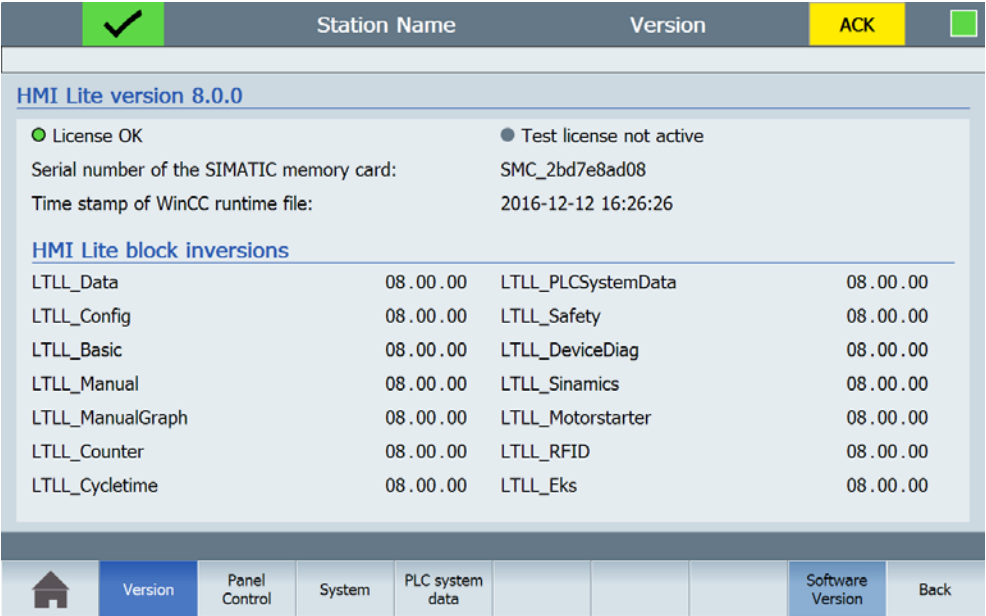


Fig. 10-1 **Version** screen

10.2 Panel Control screen

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

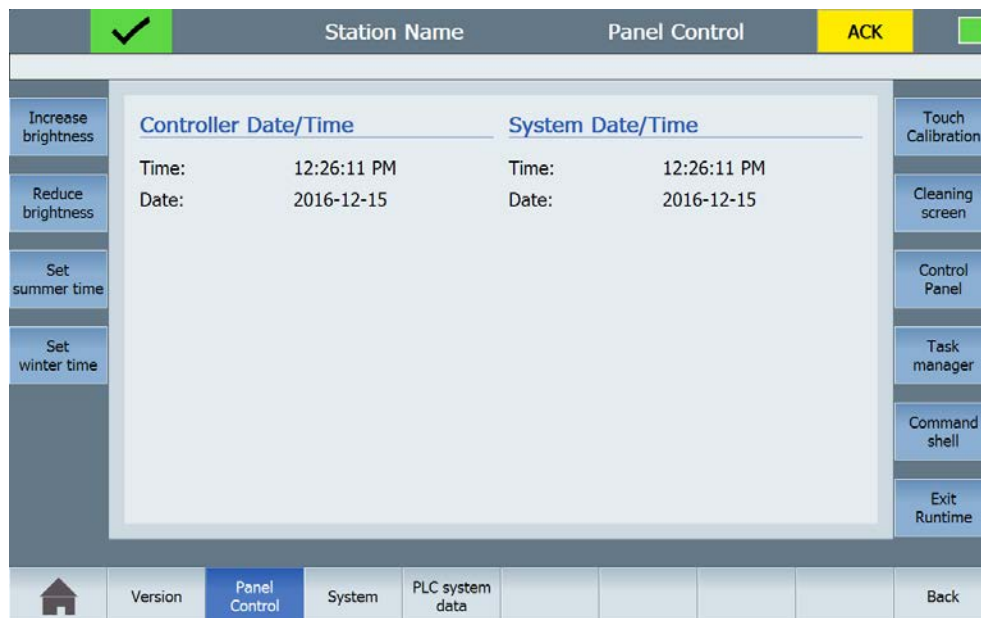


Fig. 10-2 Panel Control screen

Increase/reduce brightness

These softkeys are used to set the contrast of the operator panel.

Set summer time

Press this softkey to change the setting to summer time.

Set winter time

Press this softkey to change the setting to winter time.

Touch calibration (only for touch screen operator panels)

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

Cleaning 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.

Control panel

Pressing this softkey opens the window for the system control of the operating system.

Task manager

Press this softkey to open the Task Manager screen of the operating system.

Command shell

Press this softkey to open the command shell of the operating system.

Exit runtime

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

10.3 System screen

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.

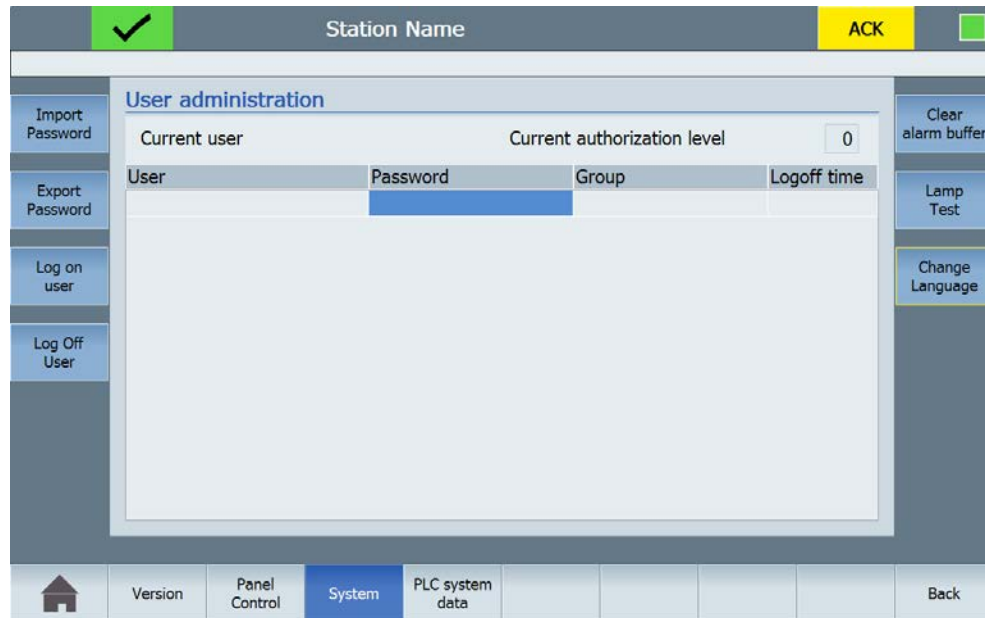


Fig. 10-3 System screen

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.

Log on user

This function opens the WinCC user log-in dialog, where you can enter the user and the password.

Log off user

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

Clear alarm buffer

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

Lamp test

The **LTLL_Data.HMI[X].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	LTLL_Data.HMI[X].global.lamptest
Format	BOOL
Range of values	1-signal when the Lamp test key on the operator panel remains pressed.
Default setting	False

Change language

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

The standard project for HMI Lite is delivered in twelve languages:

- Czech
- Chinese
- English (United Kingdom)
- French
- German
- Hungarian
- Italian
- Polish
- Portuguese (Brazil)
- Romanian
- Russian
- Spanish
- Swedish

Other languages can be implemented for specific projects.

10.4 PLC system data screen

The screen is provided as from HMI Lite Version 8.1.
The screen is provided as typified object. As from HMI Lite Version 8.1, it can be updated automatically via Project Update.

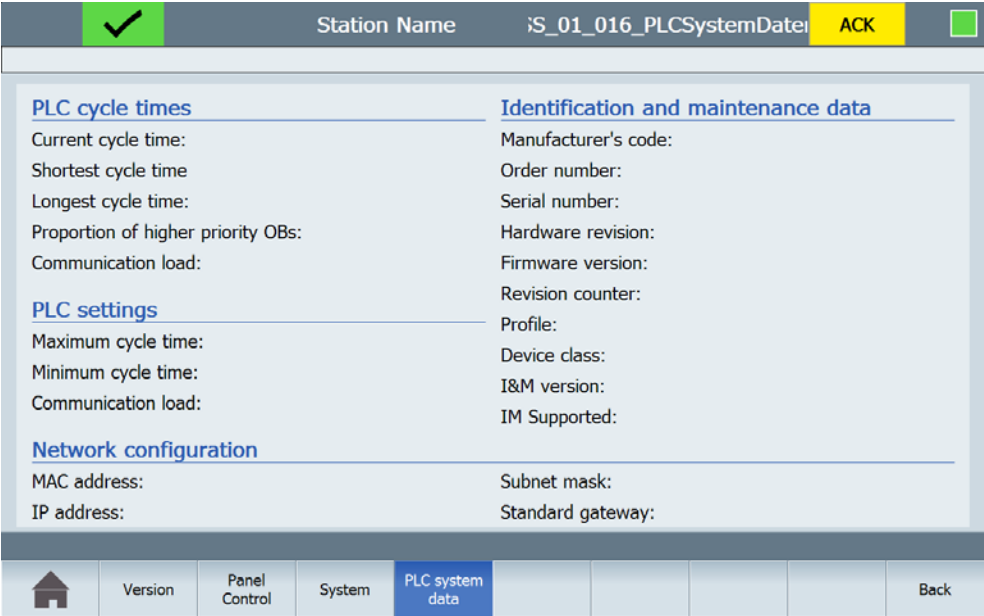


Fig. 10-4 PLC system data screen



A

A Appendix

A.1 Abbreviations

PLCSIM	Simulation of a control
RFID	Radio-frequency identification
TIA	Totally Integrated Automation

A.2 Change log

A.2.1 Edition 2016

New documentation/first Issue for HMI Lite V8.0 and SIMATIC S7-1500

To
Siemens AG

TRANSLINE Support
P. O. Box 10 60 26

D-70049 Stuttgart (Germany)

Fax +49 (0) 711 / 137 - 2838 [Documentation]

<mailto:transline-support.sdw.iadt.ger@siemens.com>

For Publication/Manual:

Solutions for Powertrain

TRANSLINE - Visualization, Operation,
Diagnostics HMI Lite

Manufacturer-Documentation

From

Name:

Company/Dept.

Address: _____

Telephone: _____ / _____

Telefax: _____ / _____

Manual

Available at:
DF MC – E-Business Workplace

2016 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 Division
Motion Control Systems
Postfach 3180
91050 ERLANGEN
GERMANY

Reserved for changes
Available at
DF MC E-Business Workplace

Printed in Germany
© Siemens AG 2016

<https://workplace.automation.siemens.com>