SIEMENS

Solutions for Powertrain

System Manual HMI Lite (SIMATIC S7-1500)

2020 Edition

siemens.com/TRANSLINE

SIEMENS

Solutions for Powertrain

System Manual HMI Lite

Function Manual

Valid for:

Control
SIMATIC S7-1500

HMI Lite Version 15.1

2020

6FC5263-1PY11-5AG1

A5E50396413B AA

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

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.

Siemens AG
Digital Industries
Motion Control
P.O. Box 31 80
91050 ERLANGEN
GERMANY

Preface

SINUMERIK documentation

The SINUMERIK documentation is organized into the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

Additional information

You can find information on the following topics at the following address (https://support.industry.siemens.com/cs/de/en/view/108464614):

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and search in manuals/information)

If you have any questions regarding the technical documentation (e.g. suggestions, corrections), please send an e-mail to the following address (mailto:docu.motioncontrol@siemens.com).

mySupport/Documentation

At the following address (https://support.industry.siemens.com/My/ww/en/documentation), you can find information on how to create your own individual documentation based on Siemens' content, and adapt it for your own machine documentation.

Training

At the following address (http://www.siemens.de/sitrain) you can find information about SITRAIN (Siemens training on products, systems and solutions for automation and drives).

FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

You can find information about SINUMERIK at the following address (http://www.siemens.com/sinumerik).

Target group

The system manual is intended for:

- Project engineers
- Technologists (from machine manufacturers)
- System startup engineers (Systems/Machines)
- Programmers

Benefits

The function manual describes the functions so that the target group knows them and can select them. It provides the target group with the information required to implement the functions.

Standard version

This documentation only describes the functionality of the standard version. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

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

Further, for the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation or maintenance.

Note regarding the General Data Protection Regulation

Siemens observes standard data protection principles, in particular the principle of privacy by design. That means that

this product does not process / store any personal data, only technical functional data (e.g. time stamps). If a user links this data with other data (e.g. a shift schedule) or stores personal data on the same storage medium (e.g. hard drive) and thus establishes a link to a person or persons, then the user is responsible for ensuring compliance with the relevant data protection regulations.

Technical Support

Country-specific telephone numbers for technical support are provided on the Internet at the following address (https://support.industry.siemens.com/sc/ww/en/sc/2090) in the "Contact" area.

If you have any technical questions, use the online form in the "Support Request" area.

Table of contents

List of figures

List of tables

1

Fundamental safety instructions

1.1 General safety instructions

⚠ WARNING

Danger to life if the safety instructions and residual risks are not observed

If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Malfunctions of the machine as a result of incorrect or changed parameter settings

As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.

- Protect the parameterization against unauthorized access.
- Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

1.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

1.3 Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit
https://www.siemens.com/industrialsecurity (https://www.siemens.com/industrialsecurity).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under

https://www.siemens.com/industrialsecurity
(https://new.siemens.com/global/en/products/services/cert.html#Subscriptions).

Further information is provided on the Internet:

Industrial Security Configuration Manual
(https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ WARNING

Unsafe operating states resulting from software manipulation

Software manipulations, e.g. viruses, Trojans, or worms, can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners
- On completion of commissioning, check all security-related settings.

General

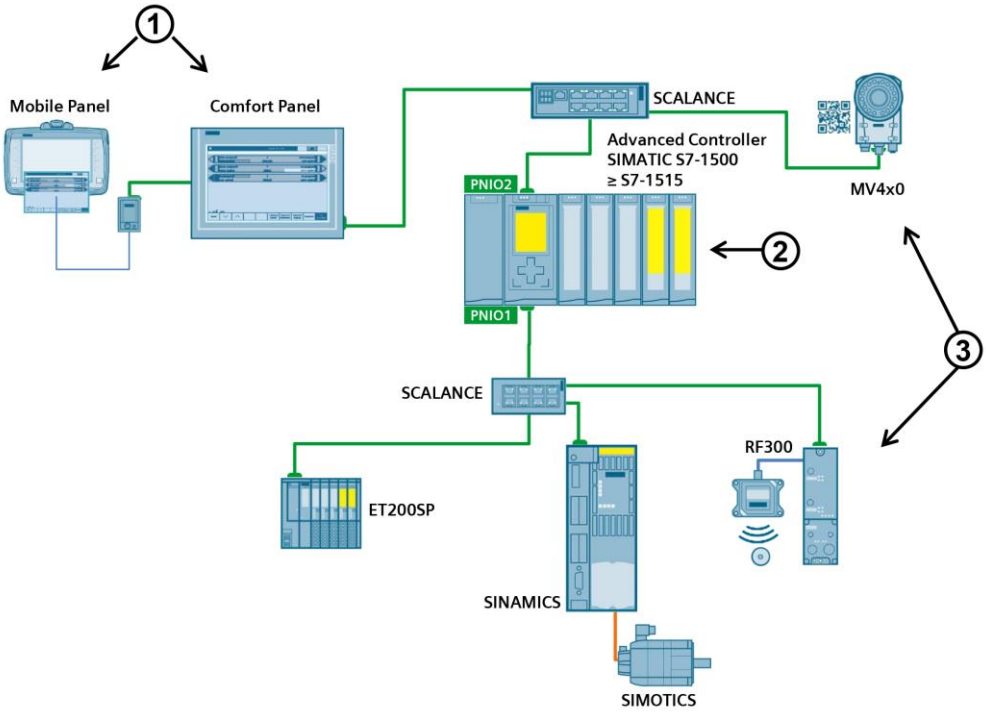
2

2.1 Product overview

HMI Lite is a user interface for the operator control and monitoring of machines. This user interface includes several screens (screen masks) for Comfort and Mobile Panels of the 2nd generation as well as for IPC477Es from the SIMATIC product series. The screens are supplied via PLC blocks in the SIMATIC S7-1500 as hardware or software version.

Navigation to the screens is effected through the HOME screen and further menu screens that the machine manufacturer can customize and extend. Meaning that the manufacturer can integrate own schemes into the navigation.

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



(1) HMI devices for displaying HMI Lite screens

(2) SIMATIC S7-1500 with the PLC program for supplying the screens

(3) External devices and I/O peripherals

Fig. 2-1: System overview example SIMATIC S7-1500

2.2 Available screens

Manual operation

- **Menu screen Function groups**
- Setup
- Power-up condition
- Selection of units
- Nut runners
- Nut runner groups
- Cycle types
- User operating screen

Production data

- Workpiece count
- Cycle times

EE@TRANSLINE

- EE consumption values
- EE measured values

Diagnostics

- **Menu screen Hardware diagnostics**

  System/CPU
  - System diagnostics
  - Web server
  - Safety
  - EKS

  RFID

    RFID

  SINAMICS
  - SINAMICS Status
  - SINAMICS Position
  - SINAMICS Alarm
  - SINAMICS SI Status

  Motor starter
  - Control/status
  - Measured values/statistics
  - Log book - Device errors
  - Log book - Triggering operations
  - Log book - Events
- Messages
- Alarm buffer
- Interface

System

- Version
- Panel Control
- System
- PLC system data

2.3 Prior 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 devices
- 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)

- STEP 7 programming
- Handling the project archive files
- Working with programs that use several address types
- Working with symbolic addressing
- Creation and testing of application programs as well as troubleshooting
- Working with binary operations, timers, counters and comparators, as well as well as arithmetic operations
- Development of programs that can use the same program block a multiple times
- Working with data access functions
- Creating data blocks
- Working with complex structures that contain parameters
- Including system functions in a program
- Using of complex data structures for data storage
- Working with the project library
- Working with global libraries

2.4 Hardware requirements

Control

The following minimum versions of the control are required:

SIMATIC S7-1500 from firmware release V2.1

SIMATIC S7-1500 software controller from firmware release V2.7

HMI devices - tested and released

HMI Lite has been tested and approved for the following SIMATIC HMI Panels and IPCs.

Table 2-1: Supported HMI devices - tested and released

Description	Display	Operator controls
SIMATIC HMI KTP900F Mobile	9" TFT widescreen display 800x480 pixel resolution	Touch screen and 10 tactile function keys, incl. LED
SIMATIC HMI TP1200 Comfort	12.1" TFT widescreen display 1280x800 pixel resolution	Touch screen
SIMATIC IPC477E 15" Multitouch	15" TFT color display 1366x768 pixel resolution	Touch screen, capacitive

SIMATIC HMI KTP900F Mobile

SIMATIC HMI TP1200 Comfort

SIMATIC IPC477E

Fig. 2-2: Supported HMI devices

HMI devices – compatible without any restrictions

The SIMATIC HMI panels and IPCs listed in the following table are compatible and HMI Lite can be run without any restrictions.

Table 2-2: Supported HMI devices – compatible without restrictions

Description	Display	Operator controls
SIMATIC HMI KTP900 Mobile	9" TFT widescreen display 800x480 pixel resolution	Touch screen and 10 tactile Function keys including LEDs
SIMATIC HMI TP1500 Comfort	15.4" TFT widescreen display, 1280x800 pixel resolution	Touch screen
SIMATIC HMI TP700 Comfort	7" TFT widescreen display, 800x480 pixel resolution	Touch screen
SIMATIC HMI TP900 Comfort	9" TFT widescreen display 800x480 pixel resolution	Touch screen
SIMATIC IPC477E 19" Multitouch	19" TFT color display 1366x768 pixel resolution	Touch screen, capacitive
SIMATIC IPC477E 19" Touch	19" TFT color display 1366x768 pixel resolution	Touch screen, resistive

HMI devices – compatible with some restrictions

The SIMATIC HMI panels and IPCs listed in the following table are compatible and HMI Lite can be run with some restrictions.

Table 2-3: Supported HMI devices – compatible with restrictions

Description	Display	Operator controls
SIMATIC HMI KP700 Comfort	7" TFT widescreen display, 800x480 pixel resolution	Membrane keyboard with 24 function keys + system keyboard
SIMATIC HMI KP900 Comfort	9" TFT widescreen display, 800x480 pixel resolution	Membrane keyboard with 26 function keys + system keyboard
SIMATIC HMI KP1200 Comfort	12.1" TFT widescreen display, 1280x800 pixel resolution	Membrane keyboard with 34 function keys + system keyboard
SIMATIC HMI KP1500 Comfort	15.4" TFT widescreen display, 1280x800 pixel resolution	Membrane keyboard with 36 function keys + system keyboard
SIMATIC HMI TP1900 Comfort	18.5" TFT widescreen display, 1366x768 pixel resolution	Touch screen
SIMATIC HMI TP2200 Comfort	21.5" TFT widescreen display, 1920x1080 pixel resolution	Touch screen
SIMATIC HMI KTP700 Mobile	7" TFT widescreen display, 800x480 pixel resolution	Touch screen and 8 tactile function keys, incl. LED
SIMATIC HMI KTP700F Mobile	7" TFT widescreen display, 800x480 pixel resolution	Touch screen and 8 tactile function keys, incl. LED
SIMATIC IPC477E 15" Touch	15" TFT color display 1280x800 pixel resolution	Touch screen, resistive
SIMATIC IPC477E 22" Multitouch	22" TFT color display 1920x1080 pixel resolution	Touch screen, capacitive
SIMATIC IPC477E 22" Touch	22" TFT color display 1920x1080 pixel resolution	Touch screen, resistive
SIMATIC IPC477E 24" Multitouch	24" TFT color display 1920x1080 pixel resolution	Touch screen, capacitive

Note

If you want to use HMI devices from Table 2-3: Supported HMI devices – compatible with restrictions, use the **Change device / version** function in the TIA Portal. Note that the resolution of your HMI device must not be less than the original station.
Under certain circumstances, it is possible that HMI Lite screens are not displayed to fill the complete screen for HMI devices that are only compatible with some restrictions.
If you adjust the resolution of the screens, check that all elements are displayed correctly.

2.5 Software requirements

2.5.1 Configuration

Software

Table 2-4: Current versions of the configuration software

Description	Version
TRANSLINE HMI Lite	V15.1
TIA STEP 7 Professional	From V15.1
TIA WinCC Comfort/Advanced	From V15.1

When using TIA WinCC Comfort, only HMI panels can be used, but not IPCs.

The 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 GRAPH sequencer. HMI Lite contains a corresponding function block to support this.

Note

Service Packs and updates for STEP 7 and WinCC are available under the following address from the Siemens Product Support on the Internet: http://support.automation.siemens.com.

Licenses

Table 2-5: Current version of the configuration software licenses

Description	Version
TIA STEP 7 Professional	V15.1
TIA WinCC Engineering Software Comfort/Advanced/Professional	V15.1

2.5.2 Runtime

Licenses

Table 2-6: Current version of the runtime licenses

Description	Version
TRANSLINE HMI Lite	V15.1

A separate license is created for each HMI device.

The following licenses are required when using SIMATIC S7-1500 software controllers and WinCC Runtime Advanced:

Table 2-7: Current version of the licenses

Description	Version
SIMATIC S7-1500 software controller CPU 150xS (F)	V2.0
WinCC Runtime Advanced (at least 2048 power tags)	V15.1

Optional

If you use PLCSIM Advanced V3.0, you require the corresponding PLCSIM Advanced 3.0 license.

If you use ProDiag, you require the corresponding ProDiag licenses.

2.5.3 HMI Lite MLFBs

Table 2-8: HMI Lite MLFBs

MLFB	Content	Version
6FC5263-0PY11-0AG0	Current software version + 1 runtime license	Current version of HMI Lite
6FC5263-1PY11-5AG1	Software V15.1 + 1 runtime license	HMI Lite V15.1
6FC5263-0PY11-0AG1	1 runtime license (without software)	Version-independent (HMI Lite copy license)

Installation

3

HMI Lite contains a project and a global library. For the initial installation use the project. To update existing projects use the library.

Note

Two control types are preconfigured in the project.

- **SIMATIC S7-1500:** subsequently, also called **PLC** or **S7-1500**
- **SIMATIC S7-1500 software controller:** subsequently, also called **software PLC** or **S7-1500 software controller**

3.1 Initial installation

Proceed as follows when you install HMI Lite for the first time:

1. Use the archived standard HMI Lite project as basis for your HMI Lite project. Unzip and open the HMI Lite TIA archive.
2. Select a **PLC** or **software PLC** control type by selecting one of the preconfigured **CPU 1515F-2 PN (PLC)** or **CPU 1507S F (software PLC)** controls and delete the control not required
3. Replace the CPU device type, if necessary.
   If you use an ET200 CPU, you have to add a new CPU and insert the blocks from the project library into the CPU.
4. Depending on the HMI device used you have to replace the HMI device type.
   See Section Station for the PLC or software PLC
5. Customize the number of HMI devices.
   After the initial installation, the SIMATIC S7-1500 is setup for two HMI devices.
   With WinCC Runtime Advanced, the SIMATIC S7-1500 Software Controller is simultaneously control and HMI device on the IPC.
   See Chapter 3.6 HMI devices
6. License the software.
   See Chapter 3.2 Licensing
7. Copy the blocks, tag tables, PLC data types and the other STEP 7 objects from your user program to the HMI Lite PLC station.
8. Copy the screens, HMI tags, text and graphic lists as well as the other WinCC objects from your user program to the station of the corresponding HMI device.

9. Assign parameters to the basic values of the HMI tags.
   - SO_00_000_index
   - SO_00_000_numberOfHomeScreen
10. Check the area pointers of the station of your HMI devices.

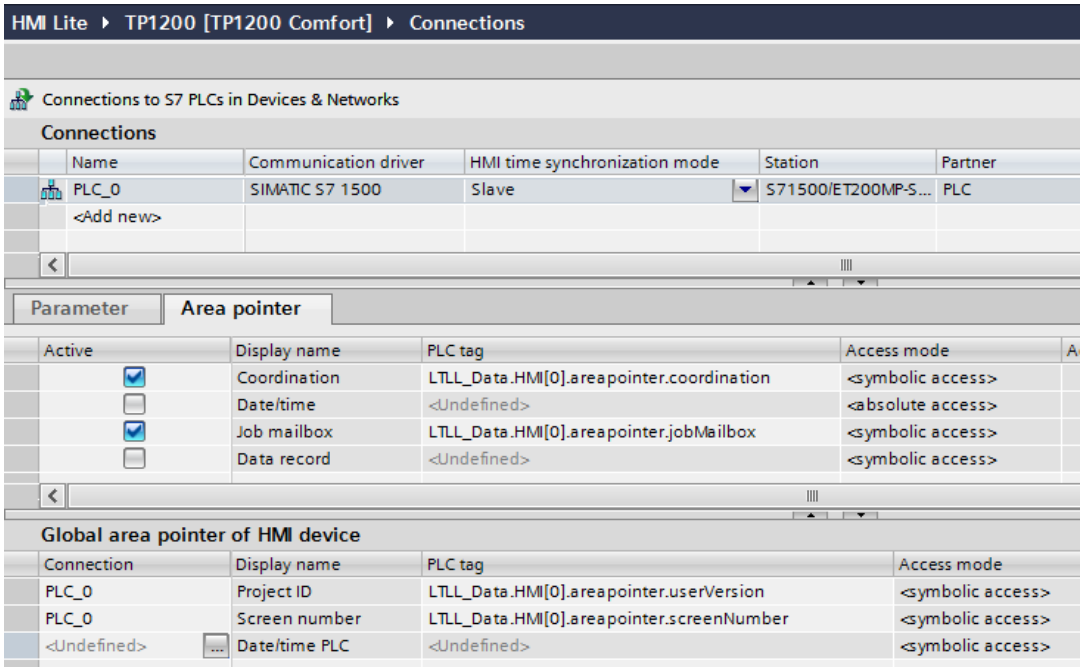


Fig. 3-1: Area pointer

HMI Lite project

An HMI Lite project comprises a STEP 7 program and a WinCC visualization.

By default it contains three stations: One station for the PLC and one station each for the KTP900F Mobile and TP1200 Comfort HMI devices.

Alternatively, a software PLC is configured on a IPC477E 15" Multitouch together with WinCC Runtime Advanced.

Station for the PLC or software PLC

The station for the PLC or software PLC contains blocks, PLC data types, tag tables and global constants.

The password-protected blocks for the PLC and the software PLC are identical. For technical reasons, the password-protected blocks for the software PLC are in folder **HMI Lite_CPU (software PLC protected blocks)**.

All other blocks and data types are in the folder **HMI Lite_CPU**.

Station KTP900F Mobile Panel

The station for a KTP900F Mobile Panel is the basis for the following HMI devices:

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

Station TP1200 Comfort Panel

The station for a TP1200 Comfort Panel is the basis for the following HMI devices:

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

Station IPC477E 15" Multitouch

The station for an IPC477E 15" Multitouch is the basis for the following HMI devices:

- IPC477E 19" Multitouch
- IPC477E 19" Touch
- IPC477E 15" Touch
- IPC477E 22" Multitouch
- IPC477E 22" Touch
- IPC477E 24" Multitouch

Note

The stations TP1200 Comfort Panel and IPC477E 15" Multitouch use the same screens. To ensure the correct resolution of the screens on the IPC477E when inserting them from the project library, in the TIA Portal select under **Options > Settings > Visualization > Resize screen** the radio button **Fit to screen**.

HMI Lite project structure

The HMI Lite project has the following folder structure.

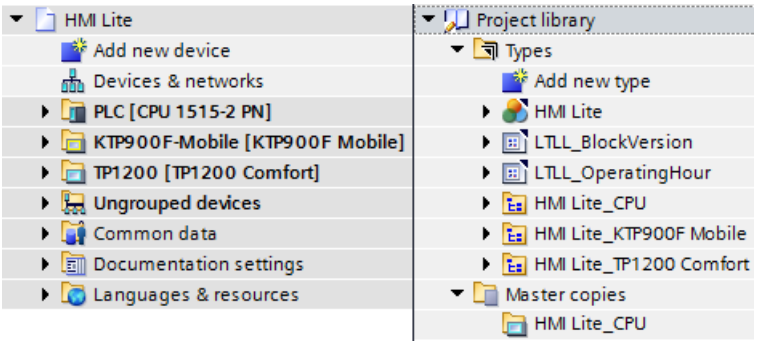


Fig. 3-2: HMI Lite project structure using a PLC with 2 panels as example

3.2 Licensing

A license is required to use HMI Lite. The license is required for each and every HMI device that uses HMI Lite. One license is included in the HMI Lite order. Licenses for additional HMI devices can be purchased under the HMI Lite copy license (without project).

One license key has to be generated per HMI device. Licensing is realized by entering the license number and the associated license key in the HMI Lite data block (**LTLL_Config**) in the HMI Lite project.

3.2.1 Creating the license key via the Internet

You generate the license key for HMI Lite via the Internet at http://www.siemens.com/automation/license.
Here the assignment of licenses to the hardware (access to the license database) is carried out via the **Web License Manager** in a standard Web browser.

1. Use the **Direct access**. The following screen page is displayed:

Fig. 3-3: Login for generating a license key

2. Enter the **License No.** and the **Dispatch note No.** These are printed on the **Certificate of License** (CoL) that you received together with the software.

3. Press the **Next** button.

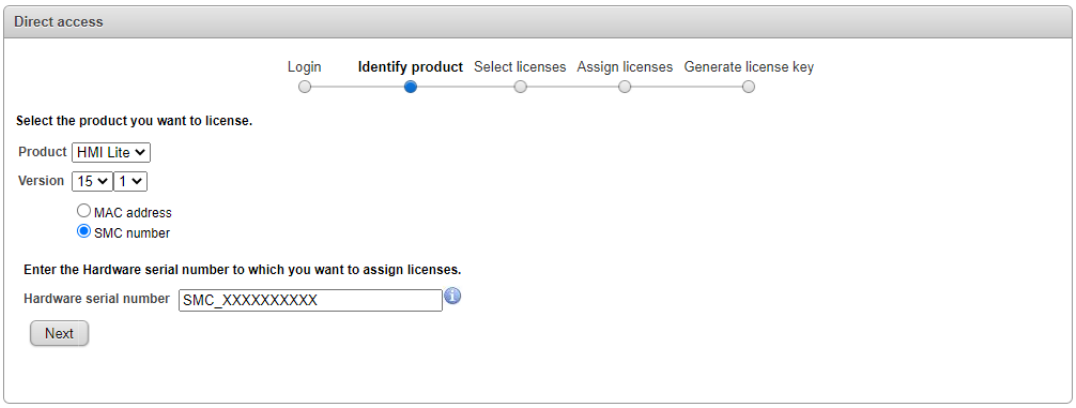


Fig. 3-4: Identify product: Entry of the data

4. Select HMI Lite at the **Product**.
5. For **Version,** select the HMI Lite version (V15.1).
6. Select the serial number to which you wish to link the license. You can select between the **MAC address** of the control and the **SMC number** of the memory card if the control has one.
   a. If you have selected a **MAC address**, enter the MAC address of the PLC that is used in the **MAC address** field.
   b. If you have selected the **SMC number**, enter the serial number of the SMC of the PLC (not the serial number of the HMI device) in the **Hardware serial number** field.

NOTICE
The license is permanently linked to the respective serial number on completion of the licensing procedure; a subsequent change is only possible under certain circumstances! We recommend that the license is linked to the MAC address of the control because the memory card can be changed without relicensing. If you link the license to the SMC, ensure that there is sufficient memory capacity; 256 MB is recommended. Additional information on this topic is provided in Chapter 3.2.5 Replacing the license key

7. Press the **Next** button.
   If licenses have already been assigned to the hardware, this is displayed.

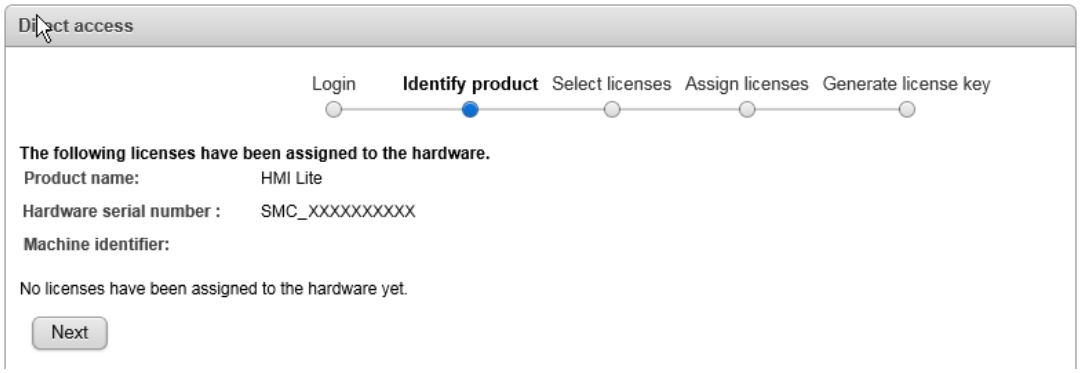


Fig. 3-5: Identify product: Licenses already assigned

8. Press the **Next** button.
   The licenses listed on the delivery note that are not yet assigned are displayed.

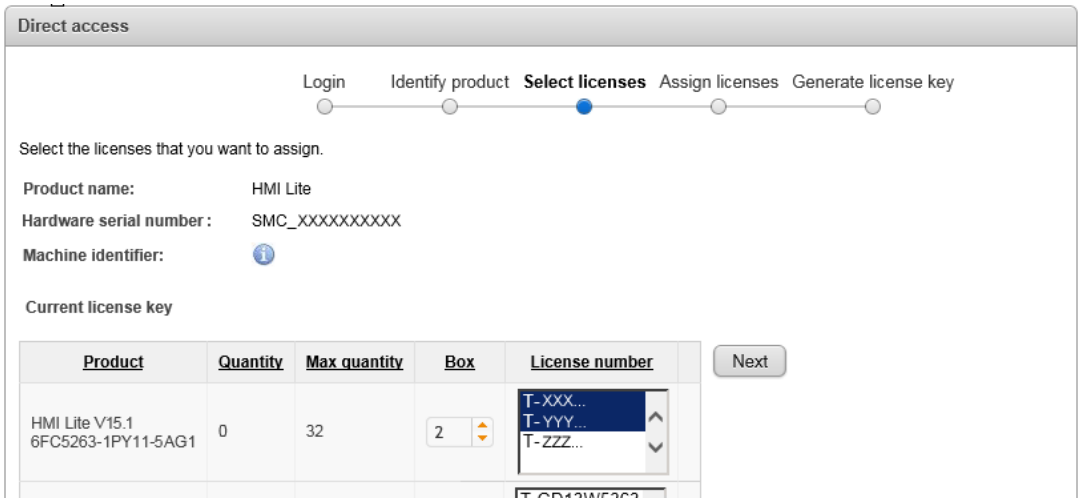


Fig. 3-6: Select licenses (example for 2 HMI devices)

9. In the line in which the HMI Lite product is displayed, select the required license in the **License number** column.
   If you are using several HMI devices, you must select a license for each HMI device (multiple selection).
10. Press the **Next** button.
    A summary of the selected licenses is displayed.
11. Check your selection.

12. Press the **Assign** button to assign the selected license(s).
Subsequently the generated license key(s) are displayed. A license key contains all the options that are assigned to the specified hardware. The assigned licenses are listed in the lower part of the screen.

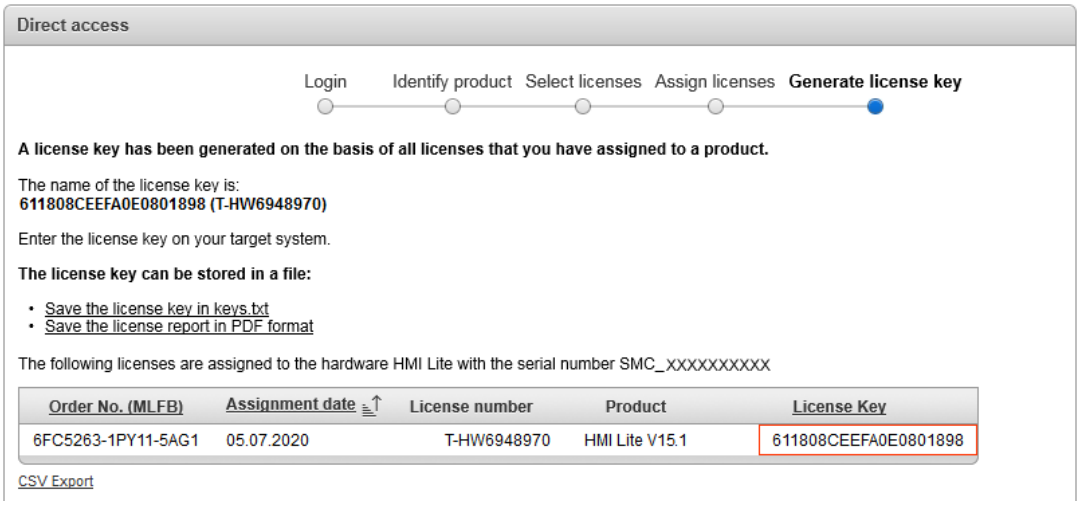


Fig. 3-7: Generate license key

13. Save the license key in a file by clicking
Save the license key in keys.txt
or
Save the license report in PDF format.

3.2.2 Entering the license number and license key in the HMI Lite project

Enter the license number(s) with the associated license key(s) in the **LTLL_Config** block at the following point:

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

[X] corresponds to one license.

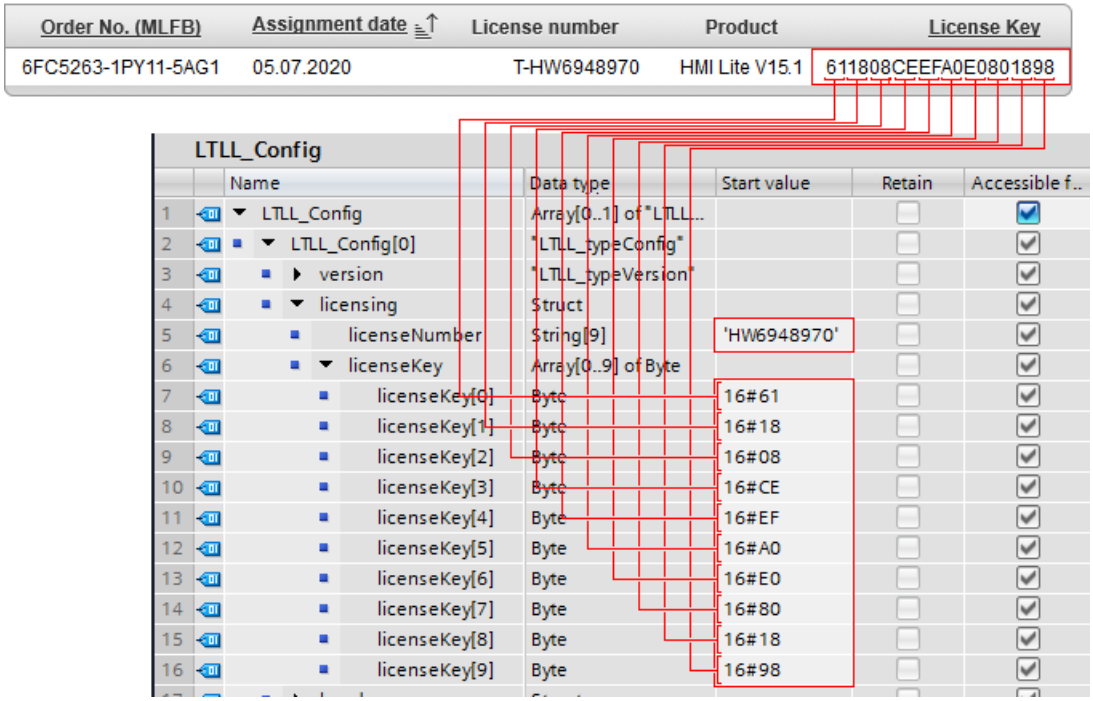


Fig. 3-8: Assigning a license key in LTLL_Config

3.2.3 Licensing for S7-PLCSIM Advanced V3.0

With a valid HMI Lite license, using the SMC number, you can also use HMI Lite with the S7-PLCSIM Advanced V3.0 simulation software.

1. To generate the license key and entry the license in the project, proceed as described in the previous sections.
2. Then enter the SMC number of the memory card in S7-PLCSIM Advanced V3.0 under **Virtual SIMATIC Memory Card > <*Station name*> > SIMATIC MC > sim_hwdb.ini** under **CPU Serial Number,** and enter two spaces at the end.

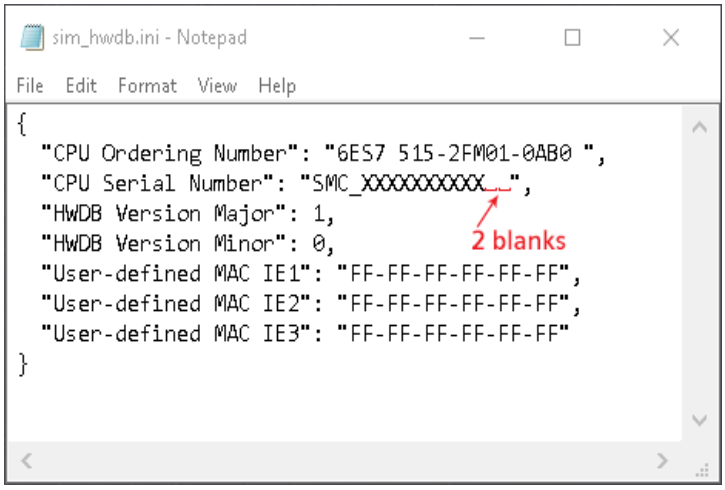


Fig. 3-9: Entering the SMC number in S7-PLCSIM Advanced

3. When calling function block **LTLL_Basic**, interconnect input **hwIDsmc** with the system constant **Local ~ Configuration**.

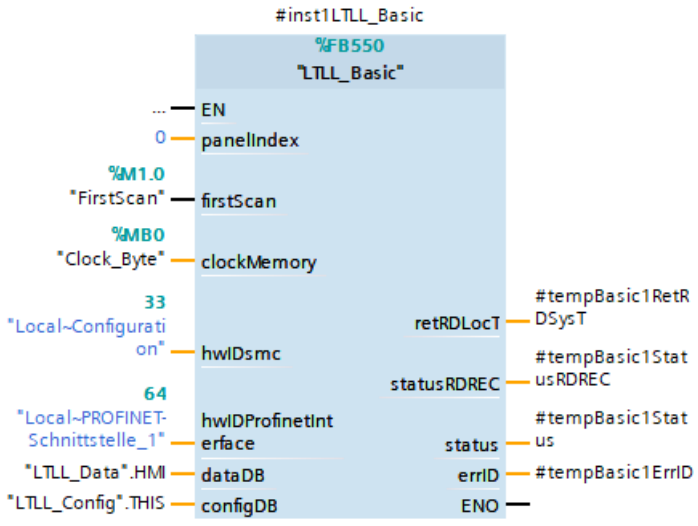


Fig. 3-10: Function block LTLL_Basic

Best use the license of the SMC that will be used later in the real system. When doing this, in the project ensure that you set input **hwIDsmc** back to **Local ~MC** before commissioning.

3.2.4 Test license

You can activate a test license for HMI Lite once as follows:

1. Set the correct time, time zone and summer/winter time in the control.
2. Enter any arbitrary 9-digit license number in DB **LTLL_Config**.THIS[X].licensing.licenseNumber.
   The test license is activated for 1000 hours.
   Once the trial license has expired, it can no longer be activated.
   You can see the remaining time of the test license in the HMI Lite version screen on the HMI device.

Fig. 3-11: Test license

3.2.5 Replacing the license key

After activation, the license is permanently linked to the respective medium (memory card or control) and is not intended for later transfer to a different medium.

Therefore, note the following:

- A memory size of 256 MB is recommended when using an SMC. However, there should be at least 24 MB.
  The memory size selected always depends on the project. It is advisable to plan a buffer as projects grow or are expanded.
- If it already appears probable that the memory card will be changed, then it is better to link the license to the MAC address of the control. In this case, the memory card can be changed as required.
- Licensing on the SMC makes sense if the memory card is to be frequently used in various controls.

Under certain circumstances, it is possible to replace the license key: It must be ensured that the medium used can no longer be used, e.g. in the event of a defect.

If the SMC used becomes too small as the project expands, and is to be replaced by a larger one, then the following rules apply depending on the situation:

1) **You want to continue using the smaller memory card without HMI Lite**

   Send us the smaller memory card. Depending on the availability, you will receive an empty card with the same size without an HMI Lite license. We then transfer your license of the smaller card to your new, larger memory card.

2) **You no longer require the smaller memory card, or this is defective, for example**

   You prove that the smaller card no longer functions. You can do this as follows:

   - You send the smaller card to the Returns Center for formatting, and then send us your confirmation.
   - You cut the smaller card and send us a photograph of the destroyed card with readable SMC number.

   We then transfer your license of the smaller card to your new, larger memory card.

3) **You use the smaller memory card for test purposes or smaller projects/systems with HMI Lite or in all other cases:**

   You order a new HMI Lite copy license for the larger memory card.

To discuss the details of the first two rules mentioned above, you can contact us via e-mail transline-support.sdw.iiadt.ger@siemens.com or via a ticket obtained from Siemens Support.

The following data is mandatory when requesting a replacement license key:

- Detailed explanation as to why a replacement license key must be generated
- In case of situation 2: Photograph of the destroyed card (e.g. cut) with readable serial number
- License number (see the license certificate)
- Previous serial number of the memory card
- New serial number of the memory card
- Name
- Machine manufacturers
- End user
- HMI Lite version (see the HMI Lite version screen)

3.3 Update

Updating a project library

Follow these steps to update HMI Lite:

1. Retrieve and open the HMI Lite library.
2. Update your project library with the global TIA Portal library by right-clicking on **Types** and selecting the menu command **Update > Library** in the **Global libraries** window.
   The "Update library" dialog is displayed.

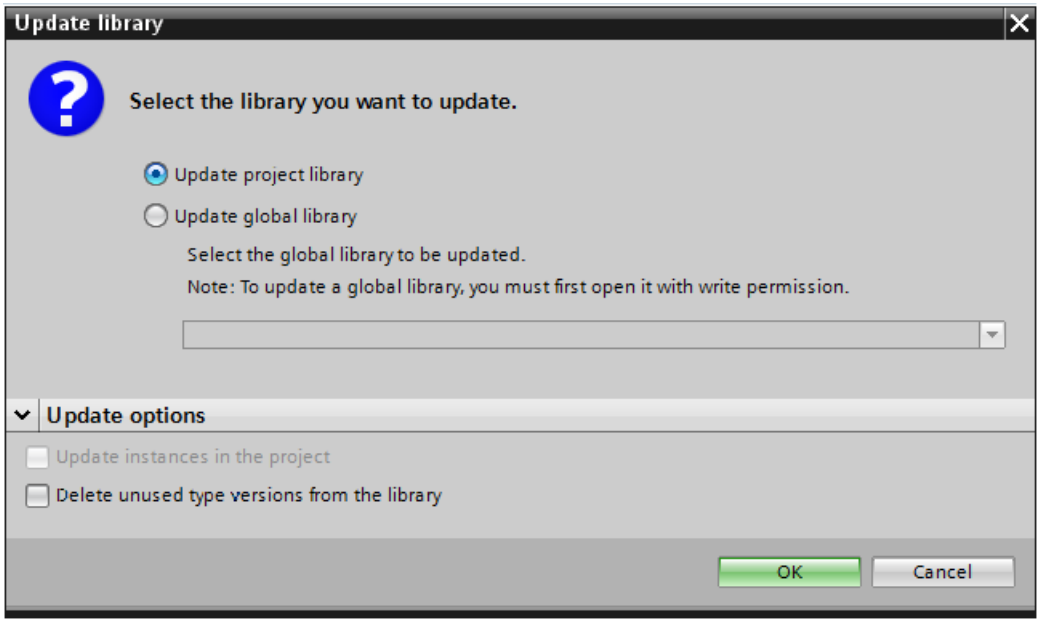


Fig. 3-12: Updating the library

3. Ensure that no check box is selected at the updating options and confirm the dialog box with **OK**.
4. Drag & drop all the elements of the **Master copies** folder from the global library into your project library to **Master copies**.

Updating CPUs

Follow these steps to update the CPUs:

1. Update your project with the project library by right-clicking on **Types** and selecting the menu command **Update > Project** in the **Project library** window.
   The "**Update project**" dialog is displayed.

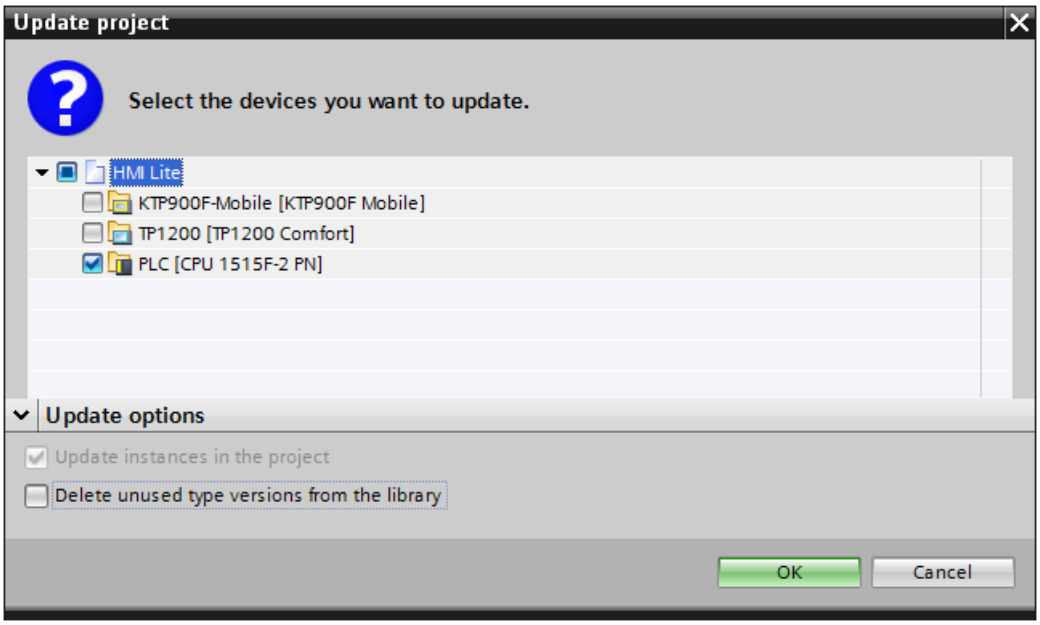


Fig. 3-13: Update the project using the example of a PLC with 2 panels

2. Confirm the dialog with "**OK**".
3. Drag & drop the **Types > HMI Lite_CPU** from your project library into your project under *Control name* > **Program blocks**, and also under *Control* **name> PLC data types**.

4. Adapt the version of your GRAPH blocks in the properties under **General > Block > Version** to V5.0.

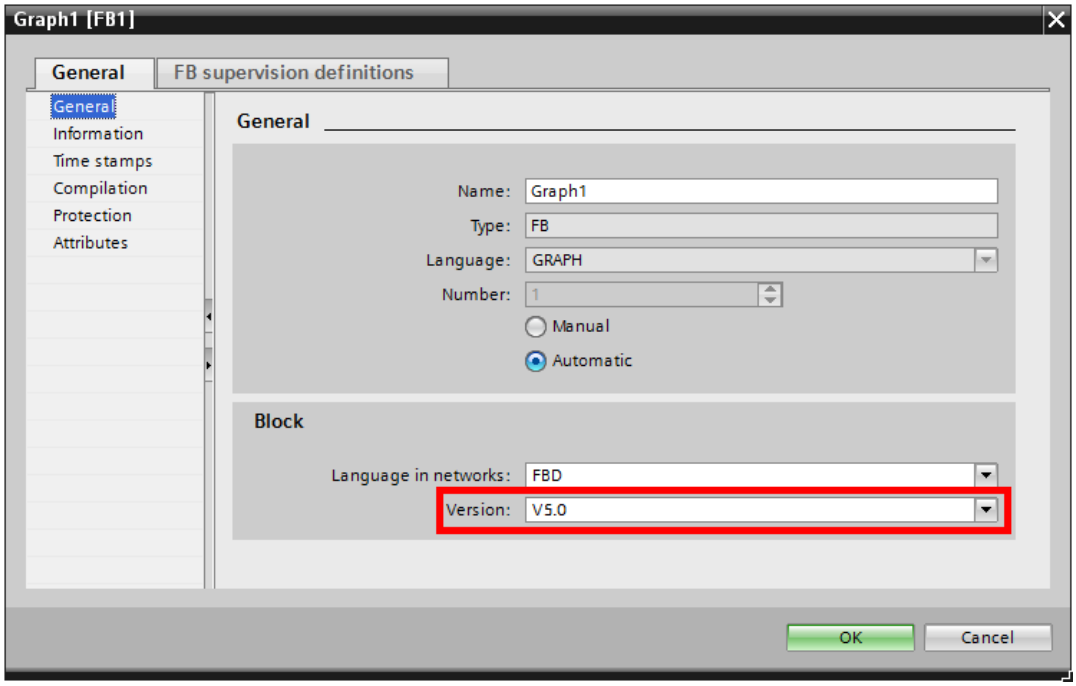


Fig. 3-14: Version of the GRAPH blocks

5. Delete all the duplicates of the **LTLL_ManualGraphControl** data block.
   This block is only required once.
6. Delete all the multiple calls of the **LTLL_ManualGraph** function block that exist on the basis of several step sequencers within a manual faceplate. The **LTLL_ManualGraph** function block only has to be called once for each configured manual faceplate that controls step sequencers.
7. Call up the **LTLL_ManualGraphInterlock** function with the parameters that you use at **MOVE** in your step sequencer FBs that use the HMI Lite sequencer control in the downstream permanent instructions.
   Requirement is that the **LTLL_ManualGraphInterlock** block from the updated project library has been integrated into the project.
8. Delete the **MOVE** instructions that fill the **LTLL_ManualGraphControl** data block.
9. Drag & drop the **Master copies > HMI Lite_CPU** from the project library into your project to **[CPU] system > Program blocks**.
10. Update the call of the **LTLL_PLCSystemData** block and assign parameters to the block.
    see Chapter 11.4 PLC system data

11. Harmonize your project with the project library to have names and paths corrected automatically.
    Objects from the copy templates are excluded from this.
    a. Select the **Types** folder in the project library.
    b. Click the symbol to open the library management.
    c. Select the **Types** folder in the library management.
    d. Click the symbol to start harmonization.
    The **Harmonize project** dialog box opens.

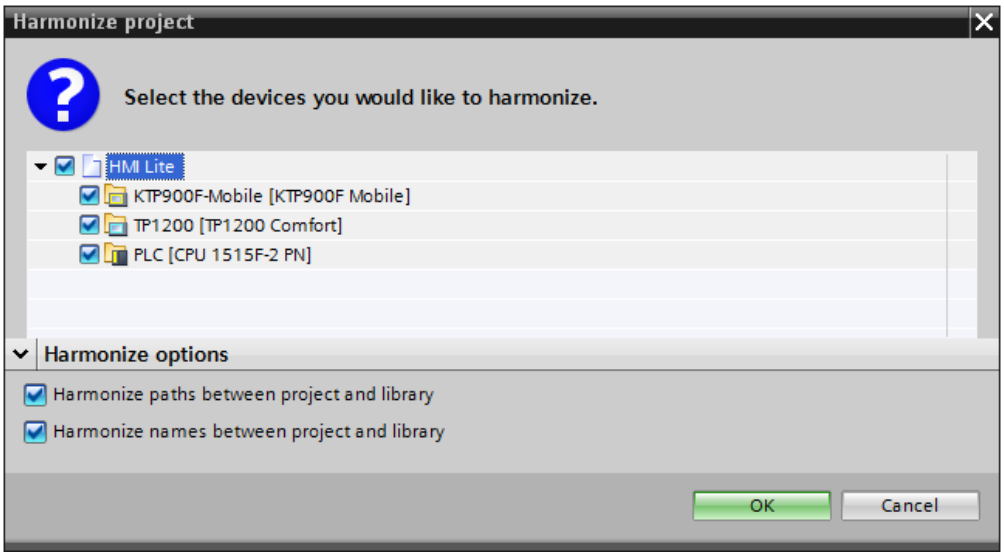


Fig. 3-15: Harmonize the project using the example of a PLC with two panels

    e. Activate the desired checkboxes and click **OK**.

Updating screens

1. Copy the HMI device(s) from **Project library > Master copies** into your project
2. Correct the connections in **Devices & networks**.
3. Copy the screens, HMI tags, text and graphic lists as well as additional WinCC objects from your user program to the station of the corresponding HMI device.
4. Accept, in as far as required, all the text lists from your user project with the following exceptions:
   - Text lists whose names begin with **SS_**...
   - Text list **SO_00_000_HeaderTextlist_2**,
     if you use own texts there.
     If required, manually add the missing texts.
5. Assign parameters to the basic values of the HMI tags.
   - SO_00_000_index
   - SO_00_000_numberOfHomeScreen
6. Check the area pointers of the station of your HMI devices.

3.4 Direct key options for HMI devices with keys

For safety reasons, the direct keys of the HMI device should be used for the manual functions.

The direct key functionality is available in the **LTLL_Manual** block. The input word of the direct keys must be handed over to the **keyButton** input parameter of the **LTLL_Manual**.

Note

For more detailed information on configuring the direct key function please refer to the TIA Portal online help.
Additional information about configuring the manual faceplates is available in Chapter 7 Manual Operation.

3.5 PLC program blocks

3.5.1 HMI Lite standard blocks

All HMI Lite standard blocks are included in the HMI Lite project

Table 3-1: Overview of the HMI Lite standard blocks

Symbolic block name	Comment
LTLL_Data	HMI Lite interface
LTLL_Config	HMI Lite configuration
LTLL_Basic	HMI Lite general
LTLL_Manual LTLL_ManualControl	PLC program for the operating screens Parameter assignment for an operating cell
LTLL_ManualGraph LTLL_ManualGraphExt	Execution of manual functions using GRAPH sequencers
LTLL_ManualGraphConfig LTLL_ManualGraphControl	Configuration and interface for the sequencer control
LTLL_Counter LTLL_CounterData	"Workpiece counter" screen Data for type unit counter
LTLL_Cycletime	"Cycle times" screen
LTLL_DeviceDiag	Interface of the device diagnostics
LTLL_Eks LTLL_EksHmiData	HMI Lite EKS diagnostics Block with EKS data
LTLL_RFID	Program code RF300 diagnostics
LTLL_Sinamics LTLL_SinamicsCFG	HMI Lite SINAMICS diagnostics Block with SINAMICS objects
LTLL_Safety	HMI Lite Safety diagnostics
LTLL_Motorstarter	HMI Lite ET200pro motor starter diagnostics
LTLL_PLCSystemData	PLC system data screen
LTLL_EETransfer	EE@TRANSLINE transfer block

3.5.2 Procedure for calling the function blocks

Call sequence

LTLL_Basic must be called as the first block.

LTLL_Manual must be called before **LTLL_ManualGraph**.

Call scheme

Table 3-2: Block call procedure

Block	Description
LTLL_Basic	Must be called once cyclically for each HMI device
LTLL_Manual	
LTLL_PLCSystemData	
LTLL_ManualGraph	Must be called once cyclically for each manual operating screen More detailed information can be found in Section 6.9.
LTLL_Eks	Must be called once for each EKS adapter
LTLL_Sinamics	Must be called once cyclically
LTLL_Motorstarter	
LTLL_RFID	
LTLL_EETransfer	
LTLL_Safety	Must be called for each workpiece counter or for each cycle time. 1 workpiece counter = 1 call 3 workpiece counters = 3 calls …
LTLL_Counter	
LTLL_Cycletime	Must be called once cyclically, if the safety diagnostics is used
LTLL_Safety	

NOTICE
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).

3.6 HMI devices

HMI Lite is supplied in the **SIMATIC S7-1500** configuration with two HMI devices, as well as in the **SIMATIC S7-1500 software controller** configuration running on IPC477E with WinCC Runtime Advanced as integrated HMI device. You must make some changes if you wish to change the number of HMI devices.

Since exactly one DB interface is required for each HMI device, you have to reduce or create these accordingly, if required. Follow these steps if you want to remove or add an HMI device at a control:

1. Reduce or extend the array in the **LTLL_Data** block by one element. (The array size mirrors the number of HMI devices on a control)

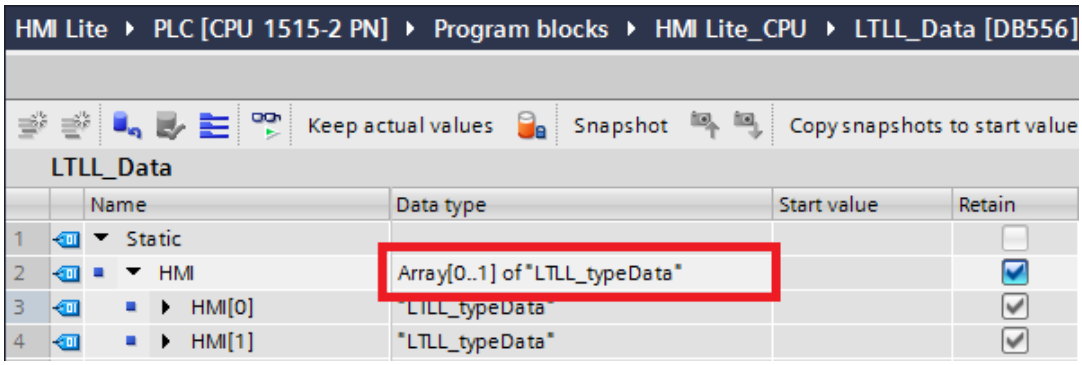


Fig. 3-16: Field in block LTLL_Data

2. Reduce or extend the array in the **LTLL_Config** block by one element. (The array size mirrors the number of HMI devices on a control)

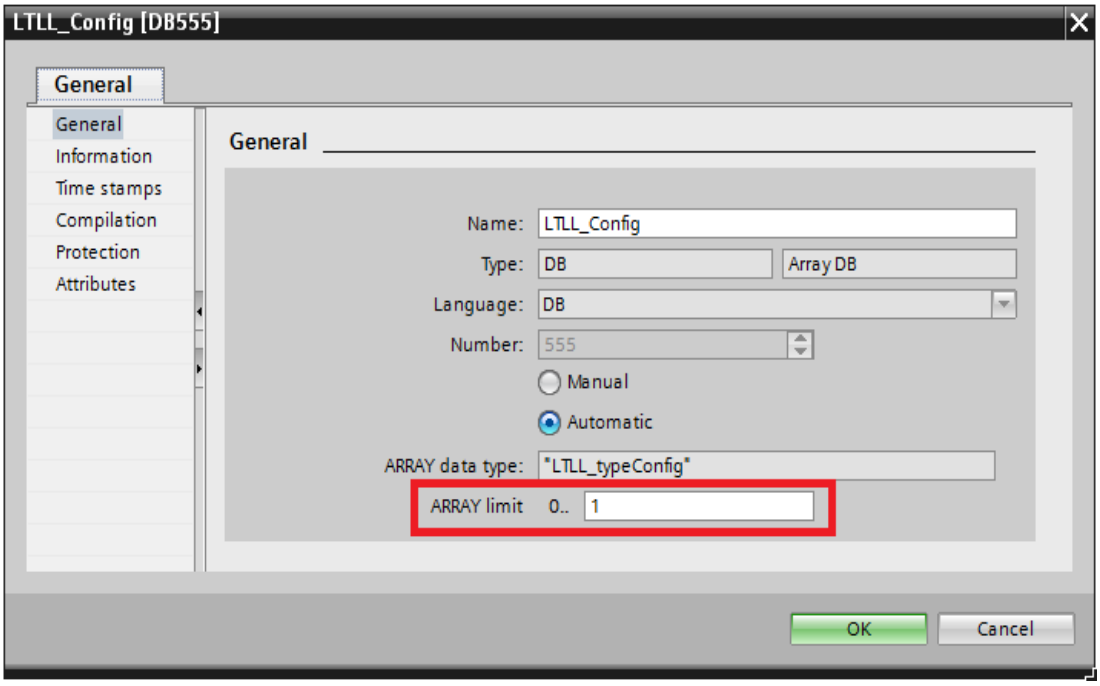


Fig. 3-17: Field in block LTLL_Config

3. The **LTLL_Basic** block must be called once in the program for each HMI device. 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 HMI device. 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 tag as the start value in the basic settings.
6. The area pointers of the HMI device 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 HMI devices.

Only a single HMI device can access the hardware diagnostics at any one time. Therefore an HMI device changeover must be configured for this purpose.

⚠ WARNING
If manual operations can be carried out from both HMI devices, these must be mutually interlocked. This is the responsibility of the user!

3.7 Working with data blocks

The two data blocks **LTLL_Data** and **LTLL_Config** are the interfaces between HMI screens and the PLC program. In contrast to the **LTLL_Data** data block, the **LTLL_Config** data block only contains data for the configuration of HMI screens and of the PLC program. The configuration settings for the machine have to be carried out in the **LTLL_Config**.

Procedure for the configuration

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

Note

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

3.8 Restrictions

- It is not permissible that the encrypted HMI Lite blocks are modified.
- It is not permissible that HMI Lite PLC data types are modified (**LTLL_type**….). A change may result in you no longer being able to 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 an update of the HMI Lite objects cannot be guaranteed.
- Screens with the SS... identifier must not be changed.
- It is not permissible that HMI Lite faceplates are changed.
- HMI tags with the SS_... identifier must not be changed.
  Exceptions are:
  SS_02_001_setupScreenNumberOfLastPage
  SS_02_001_setupScreenNumberOfFirstPage
  See chapter 7.4.3 Grouping of the movement lines in the setup screen
- 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.
- It is not permissible that the directory structure of HMI Lite objects is modified.

3.9 Modifying protected screens

If you modify protected screens, please note the following:

Revoking the connection to the type

After the connection to the library type has been revoked, the screen is no longer write-protected and can be modified correspondingly.
Please note that the screen cannot be refreshed by an HMI Lite update.

Editing the type

When you edit the type, a version with the status **Being tested** is created. The screen in this version is no longer write-protected and can be modified.
When releasing the version ensure that only the third digit (Vx.x.**x**) is modified. This guarantees an HMI Lite update.

Global settings and functionality

4

4.1 Layout of the screens and basic screen elements

All screens have a standard structure.

(1) Header, 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. 4-1: Screen elements

Header

The upper area of each screen contains the header. It contains significant status information, such as the function mode, basic position, etc. This area also contains two buttons used to access the HOME screen and the higher-level screen.

The header can be configured in two different representation variants. One variant shows the status information as text, the second by means of graphic elements.

Additional information about the header is available in Chapter 6 Header and Operating note.

Message line

The message line is part of the header and is therefore visible in each screen. All messages are displayed with number, time, status and message text. In the standard case the message that occurred last (most recently) is displayed. This can be changed in the message settings of WinCC so that the message that occurred first (oldest) is always displayed.

Work area

Texts and screen elements of the selected screen are displayed in the work area.

Operating instructions

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

Horizontal softkeys

The horizontal softkeys are located in the lower screen area. They are used to select screens (for HMI devices with keys), to scroll within the selected screen (for example page up / page down in the operator screens) or to activate special functions (for example for resetting a workpiece counter).

4.2 Menu structure

Both touch HMI devices and HMI devices with keys are available for operation in HMI Lite.

An optimized operation is available for each of the two HMI device variants.

Navigation and function keys

The menus contain the navigation and function keys that are assigned to the individual screens in the corresponding submenus.

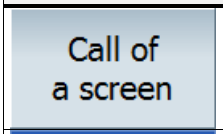
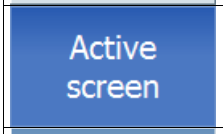
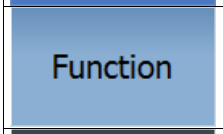
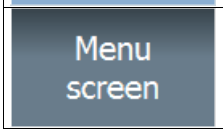
The **Previous menu/Back** button is used to return from the current menu to the previous one.

With the **Home** button you always return to the HMI Lite screen **HOME** (**SO_01_101_HomeScreen**).

Additional buttons are described in the relevant sections.

Button styles

Table 4-1: Button styles

Button	Meaning
	Button for calling a screen
	Button of the currently selected screen
	Button for calling a function within the current screen
	Button for calling a menu screen

4.2.1 Touch HMI device

The following menu screens are included in the supply:

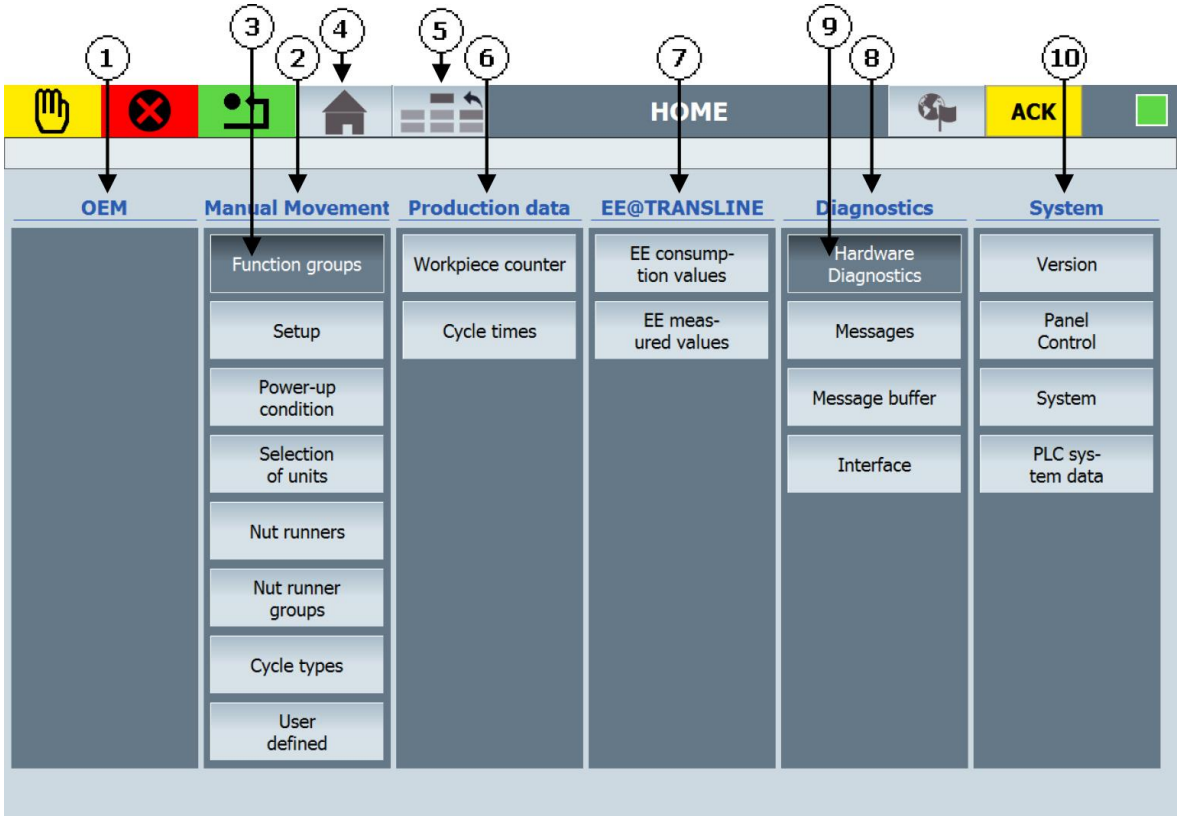
Table 4-2: Menu screens

Designation	Screen name	Description
HOME	SO_01_101_HomeScreen	Home menu screen through which screens are linked directly or via a menu screen
Hardware diagnostics	SO_01_102_MenuHardwareDiagnostic	Menu screen through which the hardware diagnostics screens are linked directly
Function groups	SO_02_101_FunctionGroups	Menu screen through which the manual faceplates are linked directly

1. Define the **SO_01_101_HomeScreen** screen as the start screen.
2. Enter the WinCC screen number of the **SO_01_101_HomeScreen** screen in the **SO_00_000_numberOfHomeScreen** tag under **Properties > Values > Start value**. By default this is: 1101.
3. If required, customize the start screen and additional supplied menu screens.

HOME

You can also use the HOME screen as the start screen for touch HMI devices. You can modify and extend it. In the supplied version it contains buttons to directly call HMI Lite screens as well as buttons for calling the menu screens **Function groups** and **Hardware diagnostics**.



(1) Area for buttons of the OEM screens

(2) Area for buttons of the manual faceplate screens

(3) Button for the "Function groups" menu screen

(4) Header button for the HOME screen

(5)Header button for the higher-level screen

(6) Area for buttons of the production data screens

(7) Area for buttons of the EE@TRANSLINE screens

(8) Area for buttons of the diagnostics screens

(9) Button for the "Hardware diagnostics" menu screen

(10) Area for buttons of the system screens

Fig. 4-2: HOME screen

Menu screens

Menu screens are a collection of buttons for directly calling screens that belong together thematically. The scope of delivery already encompasses the menu screens **Function groups** and **Hardware diagnostics**.

You can customize the supplied menu screens to your requirements and create your own menu screens.

Follow these steps to create your own menu screens:

1. Generate a new menu screen by copying and customizing a supplied screen or by creating your own screen from scratch.
2. Enter the WinCC screen number of the higher-level screen under **Events > Loaded > SetTag > Value** in the properties of the menu screen.

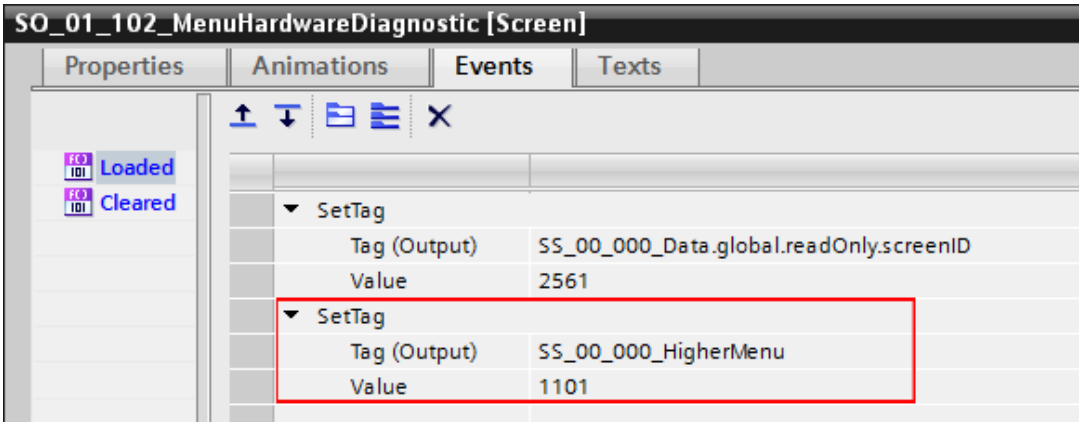


Fig. 4-3: Higher-level menu - Loaded > SetTag

3. Enter the WinCC screen number of the generated menu screen under **Events > Click > SetTag > Value** in the properties of the buttons with which screens are called.

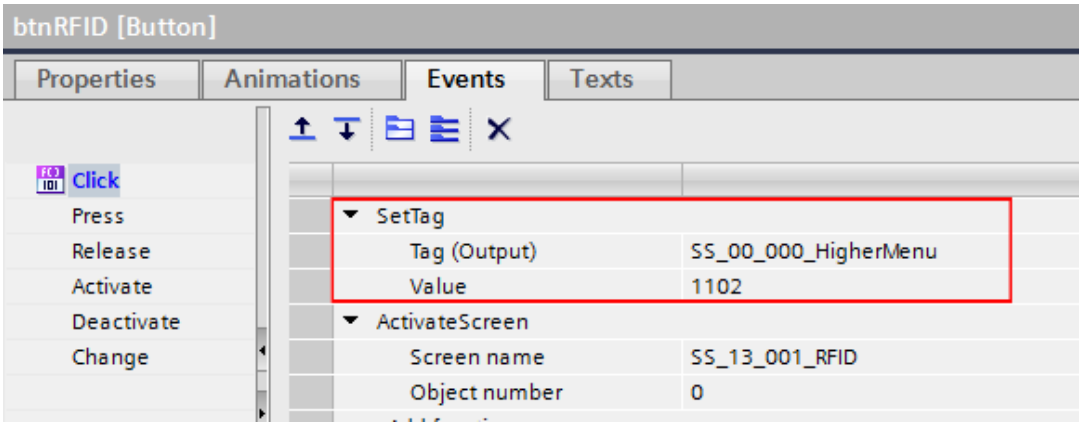


Fig. 4-4: Higher-level menu - Click > SetTag

4.2.2 HMI device with keys

1. Define the **SO_01_001_MainScreen** as the start screen.
2. Enter the WinCC screen number of the **SO_01_001_MainScreen** in the **SO_00_000_numberOfHomeScreen** HMI tag under **Properties > Values > Start value**. By default this is: 1001.
3. Remove the two buttons in the header under **Screen management > Permanent area**.

Screens of the machine manufacturer

The machine manufacturer should give the operator a graphic overview of the associated machine or plant in the HMI Lite **Overview** (**SO_01_001_MainScreen**) main screen. From here the horizontal softkeys can be used to change to one of the 7 or 9 main menus.

In the 12.1" or 15" version, the two standard main menus **OEM** and **Process** as well as a free main menu are available into which your own machine-specific screens and functions can be integrated.

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

In both variants it is possible to create a third menu level.

4.3 Clock memory byte of the control

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

The clock memory byte is used by the HMI Lite blocks for internal, time-based trigger events (for example monitoring the communication between control and OP).

It has to be transferred as an input parameter to the **LTLL_Basic** block. **LTLL_Basic** generates pulses of the individual clock signals and cyclically updates the tags of the data blocks.

Call interface

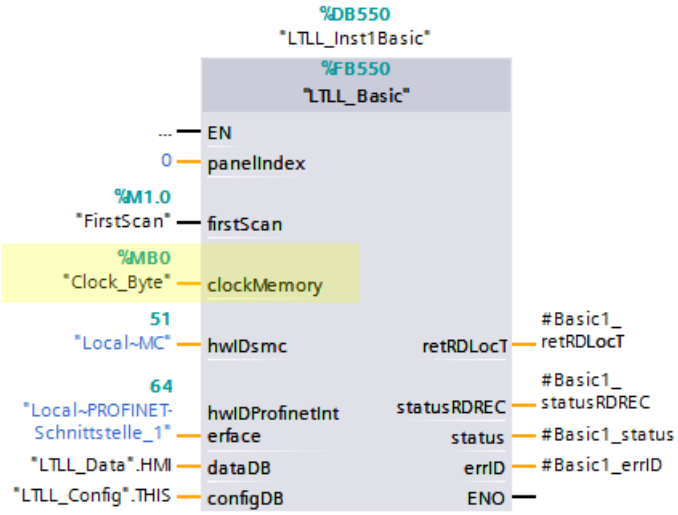


Fig. 4-5: Call interface of block LTLL_Basic

4.4 PLC system time

4.4.1 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.

4.4.2 System time and date

The **LTLL_Basic** block reads the local date and the local time of the PLC by using the instruction **RD_LOC_T**. The system time is written into the data area **LTLL_Data.HMI[X].areapointer.dateTimePLC**.

The time of the HMI device is synchronized using the automatic time synchronization setting of the TIA Portal, not by using the area pointer.

Note

Ensure that you have configured the correct time zone in the control and in the HMI device.

You can also use a different time synchronization function.

4.5 HMI Lite job mailbox

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

Structure

The job mailbox has a defined length of 4 words. The structure is shown in the table below:

Table 4-3: Structure of the job mailbox

Address	Data type	Name	Description
n+0	WORD	jobnumber	Job number
n+2	WORD	parameter_1	Parameter of the job
n+4	WORD	parameter_2	Parameter of the job
n+6	WORD	parameter_3	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.

Job number and parameters

The job number corresponds to the screen identification number. Therefore all actions that are initiated by a specific screen can 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 softkeys and buttons can be transferred to the control, the connection between the HMI device and the control must be monitored for correct operation. This monitoring is performed using the sign-of-life bit of the HMI device from the **Coordination** area pointer. The sign-of-life bit is inverted by the HMI device in 1 second intervals.

The **LTLL_Basic** function block checks cyclically whether the sign-of-life bit has been inverted to determine whether the connection to the HMI device still exists. If no inversion of the sign-of-life bit has been determined during a time interval, the job mailbox is cleared. The time interval is defined by the following parameters:

LTLL_Config.THIS[X].manualCommon.screenActiveTime

⚠ WARNING
The sign-of-life bit is not a real-time signal. Therefore it can take longer than one (1) second before the signal has changed its status. This depends on the data traffic on the network and the number of processes running on the HMI device. Using the function keys of the HMI device as I/O device-direct keys, ensures shorter response times and faster execution in the manual mode. If a touch HMI device is used, and external key module has to be used to ensure short response times and faster execution of the manual operation. The machine manufacturer is responsible for the reliable execution of the manual operation.

Coordination range pointer

The control can use this data area to query the status of the HMI device, for example startup of the HMI device, current operating mode and ready for communication.

Structure of the Coordination area pointer

The **Coordination** area pointer with a length of one word has the following structure:

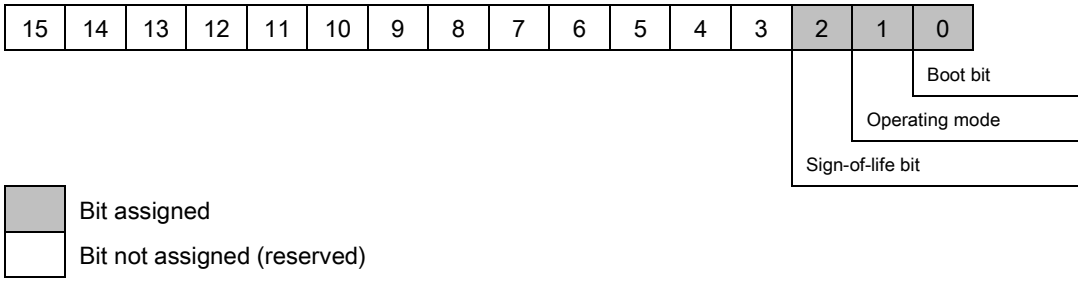


Fig. 4-6: Structure of the Coordination area pointer

4.6 LTLL_Basic block

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

Call interface

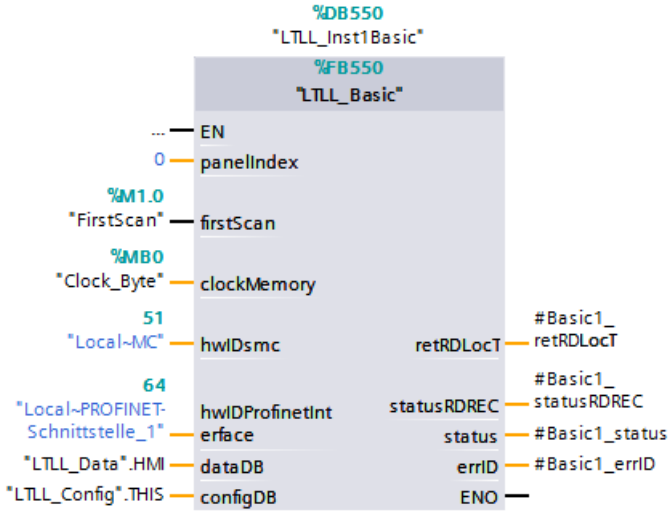


Fig. 4-7: Call interface of block LTLL_Basic

Parameters

Table 4-4: Description of the LTLL_Basic parameters

Name	Declaration	Type	Standard	Description
panelIndex	Input	INT	-	Index of the HMI device (0-based)
firstScan	Input	BOOL	FirstScan	Restart flag, startup bit
clockMemory	Input	BYTE	Clock_Byte	Clock memory byte, configured in object properties of the CPU (device configuration)
hwIDsmc	Input	HW_IO	Local~MC	System constant of the SMC card of the control
hwIDProfinetInterface	Input	HW_IO	-	System constant of the PROFINET interface
dataDB	InOut	Array[*] of LTLL_typeData	LTLL_Data.HMI	HMI Lite Runtime data DB
configDB	InOut	Array[*] of LTLL_type Config	LTLL_Config.THIS	HMI Lite Configuration DB
retRDLocT	Output	INT	-	Return value of RD_LOC_T
StatusRDREC	Output	DWord	-	Status of RDREC
status	Output	Word	-	Block status
errID	Output	Word	-	Local error ID

Output parameter status

Table 4-5: Description of the output parameter status of LTLL_Basic

Error code (W#16# ….)	Description
16#8200	No license key entered
16#8201	Invalid activation code
16#8202	Activation code invalid for version
16#8203	Input parameter **hwIDsmc** connected incorrectly
16#8204	Invalid PanelIndex
16#8205	License key incomplete

Procedure for creating new screens

5

5.1 The Template screen

The **SS_00_000_Template** screen serves as a template for inserting machine-specific screens while retaining the screen layout and the menu structure.

Proceed as follows:

1. Duplicate the **SS_00_000_Template** screen.
2. Rename the screen.
3. Configure the screen.
4. Integrate the screen into the menu structure.

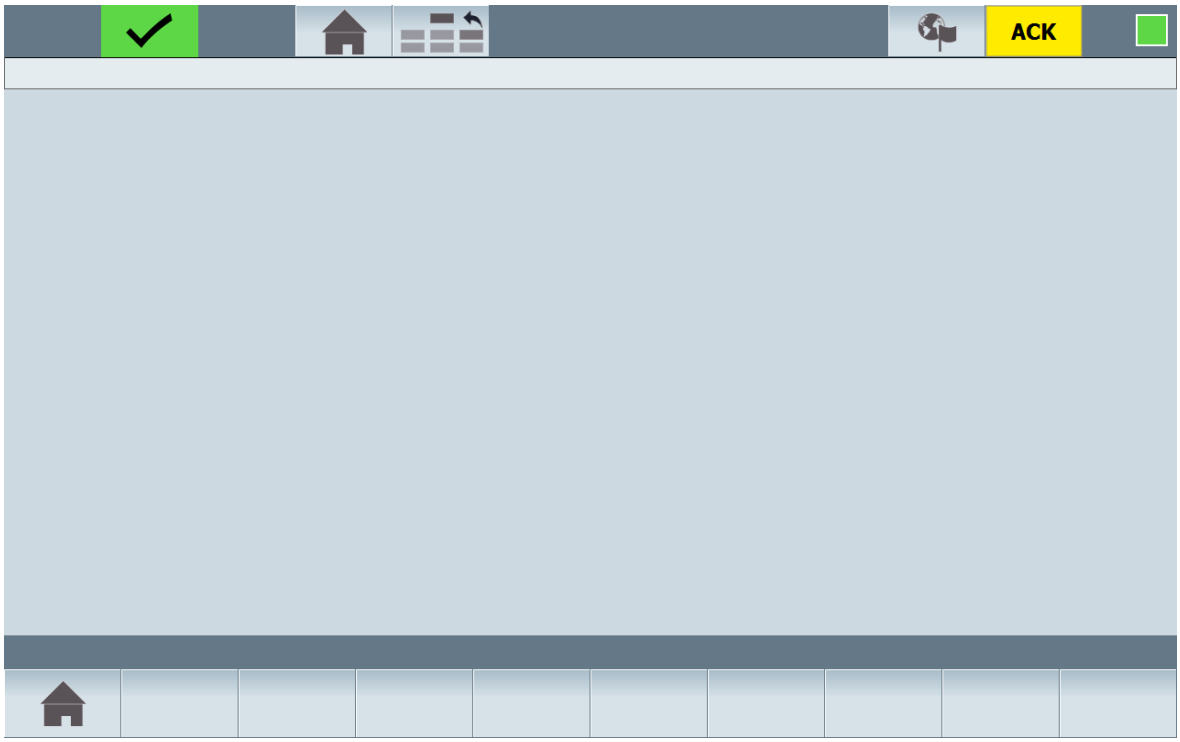


Fig. 5-1: Screen SS_00_000_Template as template for creating your own screens

5.2 Designation conventions

All WinCC elements, such as screens, tags, 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 may modify 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 by their name, these elements appear at the top of the list.

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

Designation convention syntax

The WinCC elements screens, tags, symbol lists and graphics have to be named according to these standard naming conventions.

Table 5-1: Syntax of the designation convention for screen elements in WinCC

Name structure of the screen elements: AB_XX_XXX_Name	
Symbol	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 may modify 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

P	S	_	29_021	_	Recipes
					The name of the screen is **Recipes**.
			The screen number is 29021.		
	The screen may only be modified by Siemens.				
The mask was created by Siemens for a specific project.					

All other elements that are only used in the **Recipes** screen, such as tags or symbol lists, also have the identification 29_021.

e.g.: Tag: PS_29_021_Index

Tag: PS_29_021_SelectedMaster

Text list: PO_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 tags or symbol lists used in the header).

Screen element groups

In some cases, screen elements, such as tags, are used in common by complete screen groups. A common group identification is then assigned to such screen elements.

For example, all screen elements that are used by all operating screens have the identifier 02_000.

Screen elements used only for a specific operating screen have the identification of the corresponding screen to which they are assigned (for example for the **SS_02_001_Setup** screen).

5.3 Identification of the selected screen

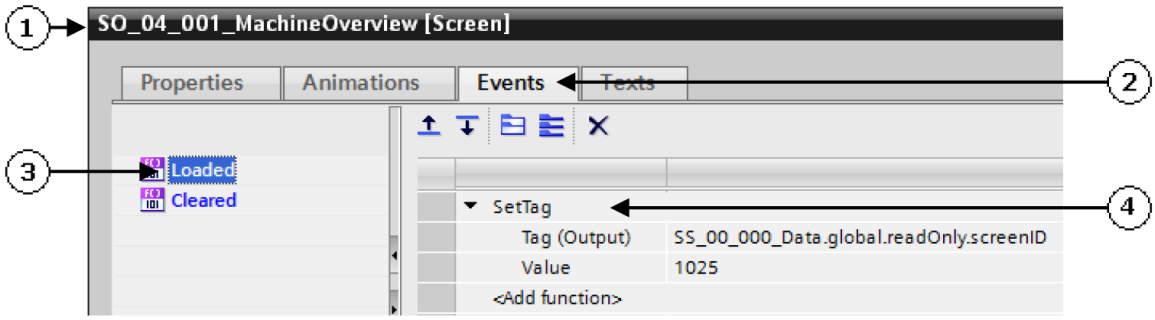
The information which screen is selected on the HMI device is made available in the WinCC **SS_00_000_Data.global.readOnly.screenID** tag. During the screen setup the corresponding value is written into the tag. When the screen is removed, the tag is set to zero.

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 tag **SS_00_000_Data.global.readOnly.screenID** 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



(1) Properties of a screen

(2) **Events** tab

(3) Event **Loaded** when the function is initiated

(4) Function **SetTag** that is to be executed

Fig. 5-2: Configuring the screen event to identify the selected screen

Codes to identify the individual screens

Table 5-2: Identification code for individual screens

WinCC Screen number	Designation of the system screen	Code to identify the screen			
		High byte [dec.]	Low byte [dec.]	[dec.]	[hex.]
General screens					
1101	SO_01_101_HomeScreen	01	101	357	0x0165
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 operation					
2101	SO_02_101_MenuFunctionGroups	02	101	613	0x0265
2011	SO_02_011_FunctionGroups*****	02	011	523	0x020B
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

WinCC Screen number	Designation of the system screen	Code to identify the screen			
		High byte [dec.]	Low byte [dec.]	[dec.]	[hex.]
Machine information					
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
EE@TRANSLINE					
5001	SS_05_001_EnergyEfficiencyEconomy	05	001	1281	0x0501
5002	SS_05_002_EnergyEfficiencyMeasurement	05	002	1282	0x0502
Hardware diagnostics					
10101	SO_10_101_MenuHardwareDiagnostic	10	101	2661	0x0A65
10001	SO_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

WinCC Screen number	Designation of the system screen	Code to identify the screen			
		High byte [dec.]	Low byte [dec.]	[dec.]	[hex.]
SINAMICS diagnostics					
11001	SS_11_001_ControlStatusword*	11	001	2817	0x0B01
11002	SS_11_002_ControlWord**	11	002	2818	0x0B02
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
Motor starter diagnostics					
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

WinCC Screen number	Designation of the system screen	Code to identify the screen			
		High byte [dec.]	Low byte [dec.]	[dec.]	[hex.]
RFID diagnostics					
13001	SS_13_001_RFID	13	001	3329	0x0D01
Safety					
14001	SS_14_001_Safety	14	001	3585	0x0E01
Siemens project-specific screens					
29000	PP_29_yyy_ScreenName PP stands for project-specific screens	29	yyy = 0-255	-	-
OEM-specific screens					
30000	OO_30_yyy_ScreenName****	30	yyy = 0-255	-	-
Customer-specific/project-specific screens					
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-specific or customer-specific screens (not available in the standard version)

***** Is only used for HMI devices with keys

5.4 Style elements

As of HMI Lite V8, a WinCC style is used so that all elements have the same look and feel.

Activate **Properties** > **Styles/Designs** > **Style/Design settings** in your WinCC objects and select the matching style element under **Design style element**.

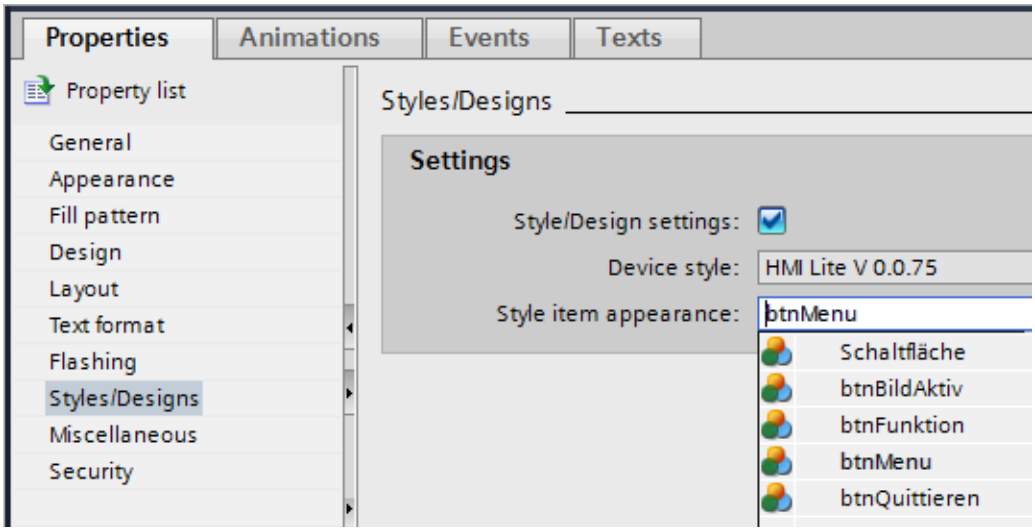


Fig. 5-3: Style elements

Several style elements may exist for WinCC objects. Different style elements are available for buttons, for example:

- Style element **Schaltfläche**: Calling other screens
- Style element **btnBildAktiv**: Current screen
- Style element **btnFunktion**: Function within the current screen
- Style element **btnMenu**: Calling a menu screen

Additional objects, such as I/O fields and text fields, are available with different style elements. These are, for example, text alignment, transparent background, headings, colored background.

See also Chapter 4.2 Menu structure > Button styles

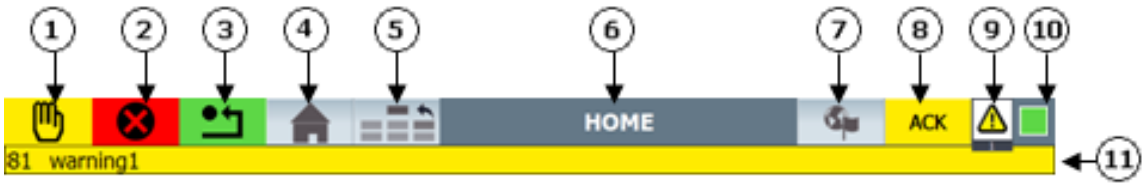
Header and operating note

6

6.1 Header

6.1.1 Header layout

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



(1) Display of current operating mode
(2) System state
(3) Display of basic position
(4) Button for calling the HOME screen
(5) Button for calling the higher-level menu
(6) Header text field 2
(7) Button for changing the language
(8) Acknowledgment button for the current message in the message line
(9) Message indicator
(10) Sign-of-life bit
(11)Alarm and message line

Fig. 6-1: Layout of the header

6.1.2 Display of current function mode

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

Table 6-1: Display of the function modes in the header

Display	Function mode
[empty]	No function mode selected
Auto	Interlinked operation
Cycle	Single mode
Step	Single-step mode
Manual	Setup

Every function mode can be displayed as follows:

- Gray background: Function mode is selected but not active
- Green or yellow background Function mode is selected and active

No function mode is displayed when:

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

Table 6-2: Display of the function modes (selected, active/not active)

Function mode selected		Function mode activated	
Text	Symbol	Text	Symbol
Auto		Auto	
Cycle		Cycle	
Step		Step	
Manual		Manual	

Runtime interface function mode selection LTLL_Data.HMI[X].header.mode

Display of the respective function mode is realized using the interface bits in the **LTLL_Data** data block. The function mode is displayed if the interface bit = **TRUE**.

Table 6-3: Display of the actual function mode - interface bits

Text	Symbol	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 function mode** status is displayed.

Runtime interface Function mode selected / active

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

Configuration

Configuration is not required.

6.1.3 Status display

The following plant states are possible:

Table 6-4: System status display

Text	Symbol	Meaning	Description
OK		Ready to operate	No fault or warning is present
Warn		Warning	One or more warnings are present
Alarm		Fault	One or more faults are active

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

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

Table 6-5: Status display - interference bits

Meaning	Interface	Type
Ready to operate	--- (if 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 tags or objects (e.g. with alarm or message bits).

Configuration

Configuration is not required.

6.1.4 Display of the basic position

The following states are possible for the basic position display:

Table 6-6: Basic position display – possible statuses

Text	Symbol	Meaning	Description
		Empty	The machine is not in the basic position.
Home		Basic position	The machine is in the basic position.

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

The **basic position** is displayed using the following bit in the **LTLL_Data** data block:

Table 6-7: Basic position display – interface bit

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

The **basic position** status is displayed when the bit is **TRUE**.

Configuration

Configuration is not required.

6.1.5 Text fields

One text field is available for displaying the machine-specific texts.

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

The text is controlled using one tag in the **LTLL_Data** block. The text assigned to the value of the tags in the WinCC text list is displayed.

Address:	LTLL_Data.HMI[X].header.textindex2
Format:	WORD
Value range:	0-65535
Default setting:	W#16#0

Configuration

Text list:	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 numbers of the selected screen are transferred from the WinCC **Screen number** area pointer to the **LTLL_Data.HMI[X].header.textindex2** tag.

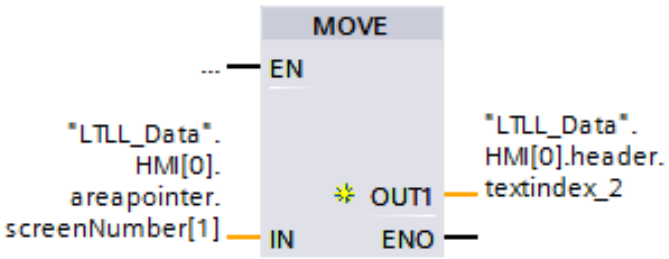


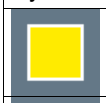
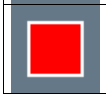
Fig. 6-2: Supply of the 2nd text list SO_00_000_HeaderTextlist_2

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

6.1.6 Sign-of-life of the PLC

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

Table 6-8: Display sign of life of the PLC

Field	Interface
 Cyclic flashing at intervals of approximately 1 second	The PLC is in **RUN** mode. Communication active between the HMI device and the PLC.
	Communication with the PLC has been interrupted.
	The PLC is in **STOP** mode.

6.1.7 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 toggled by means of:

LTLL_Config.THIS[X].header.useTextHeader

FALSE = Symbolic header

TRUE = Text header

6.2 Operating note

The operating note 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 tags are used to control the dynamic behavior of the text output field. The **LTLL_Data.HMI[X].global.prompt.index** tag is used to select which text from the WinCC text list is to be displayed.

The **LTLL_Data.HMI[X].global.prompt.attribut** tag is linked to the **Appearance** animation of the operating note. It controls the color marking and/or the flashing of the operating note.

Configuring the operating note

1. Select your HMI device in the project navigation.
2. There select **Screen management > Templates > Template**.
3. Select the **seaUserNote** object in the **Template** screen.
4. Define your own appearance in the properties of **seaUserNote** under **Animations > Display > Appearance**.

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

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

Configuration

The WinCC text list **SO_00_000_OperatorPrompt** contains all the texts that can be displayed in the text field for operating notes.

Table 6-9: WinCC text list SO_00_000_OperatorPrompt

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

Manual operation

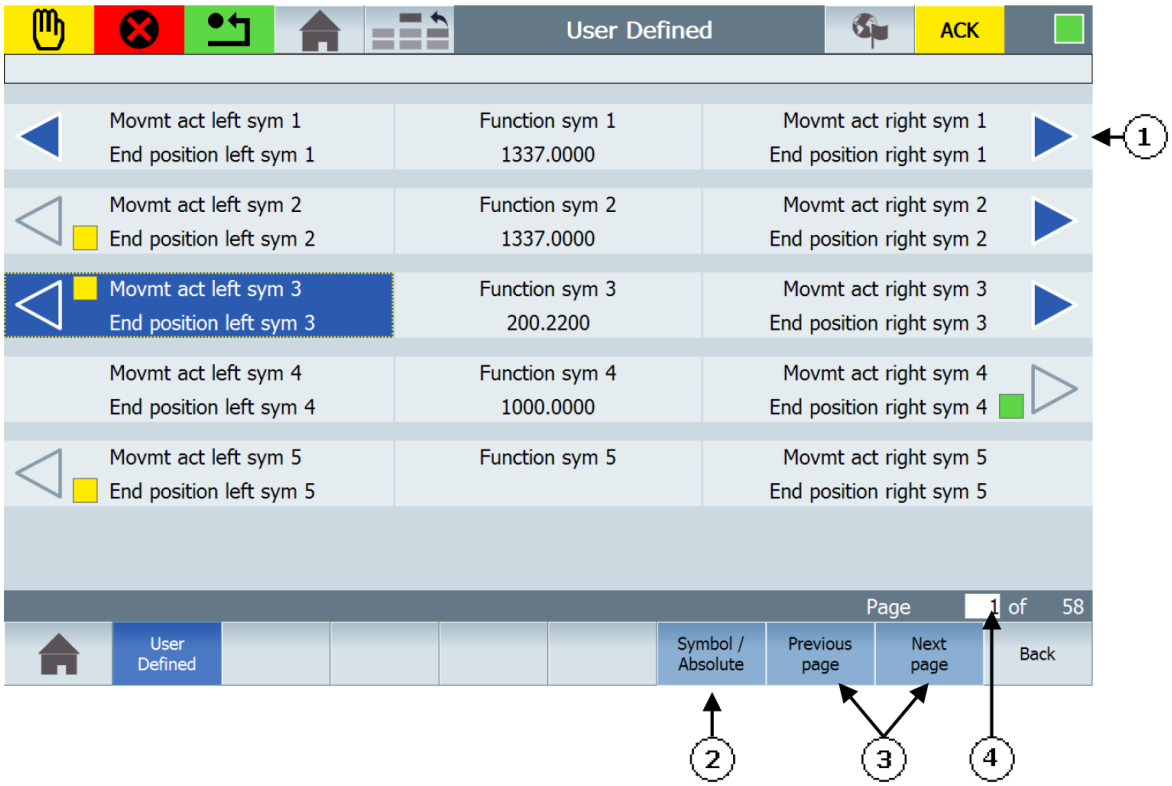
7

7.1 Overview

7.1.1 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 types and perform other functions for which a selection must be made.

All screens from the manual operation area have the same general structure.



(1) Movement/function line

(2) Switchover **Symbols/Absolute**

(3) Scroll to previous / next page

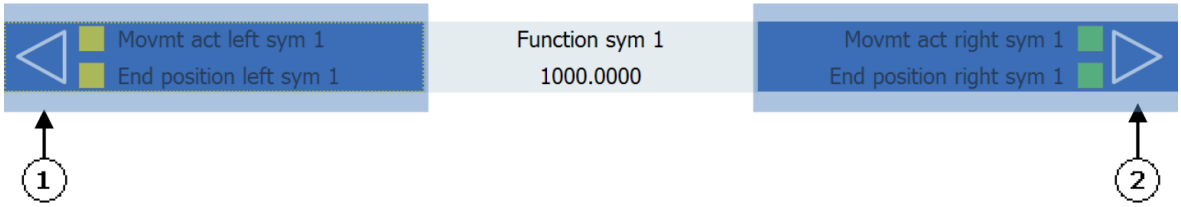
(4) Current page / total number of pages

Fig. 7-1: Structure of screens for manual operation

Movement and function line

Every movement or function is displayed in a separate line and can be performed in two directions, such as input/output, open/close, up/down, forwards/backwards. One direction of the movement/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.

Each movement/function can be initiated or selected by touching the respective areas.



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

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

Fig. 7-2: Manual operation – selection/actuation of a movement/function line

There are five different modes for selecting movement/function lines

- **Touch direct**:
  The function is active as long as the button is pressed.

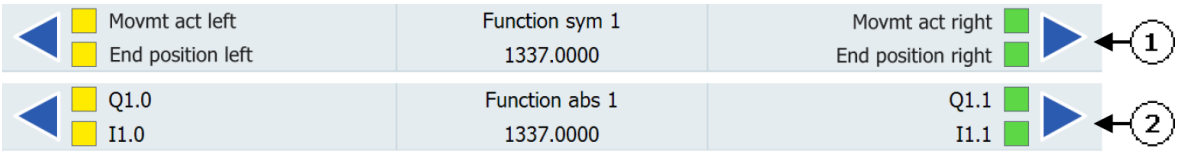
- **Touch pre-selection**:
  To prevent the inadvertent initiation of a movement, the movement that is to be executed must first be selected by touching the appropriate touch-sensitive surface. The selection of the movement is confirmed by blue flashing on the movement side. Once the movement has been confirmed, the movement can be initiated by subsequently touching the touch-sensitive area. The movement side is permanently marked with blue as confirmation.
- **Touch external:**
  The function is selected through the button. The function is enabled through an external key module.
  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 bit deactivates the movement due to a communication problem between the HMI device and the control.
- **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 through an external key module.
  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 bit deactivates the movement due to a communication problem between the HMI device and the control.

* Only in the case of 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 designations of the inputs and outputs (e.g. I1.0, Q1.0) that are assigned to the corresponding movements/functions.



(1) Symbolic view

(2) Absolute view

Fig. 7-3: Screens for manual operation - absolute and symbolic Representation

Scrolling

A scroll function can be used to call all configured movements/functions for a maximum display of six (12.1" or 15" device) or four (9" device) movements/functions per page.

When the scroll function is performed, all displayed function lines are replaced by the function lines of the next page.

If the **Next page** button is pressed on the last page, the first page is displayed. If the **Previous page** button is pressed on the first page, the last page is displayed.

The screen cannot not be changed while a movement/function is being carried out. The page is locked.

Current page/Total number of pages

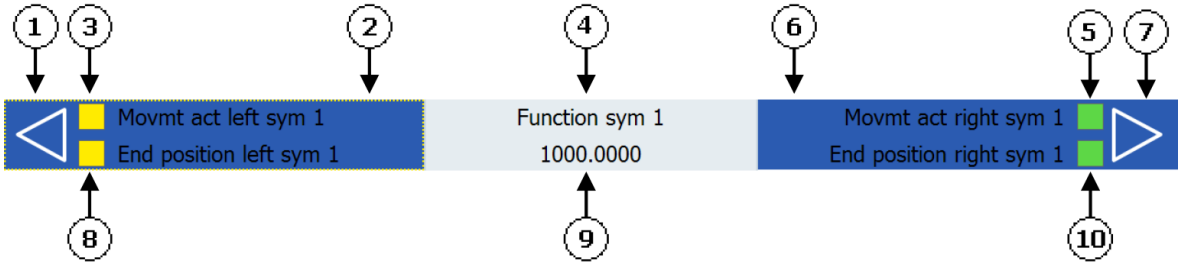
The current page number and the total number of pages are displayed at the bottom of the screen. A page can be selected directly by entering the page number of the keyboard or on a key pad.

Note

When the setup screen is grouped in function groups, the page numbers refer to the function groups and not to the setup screen itself.

7.1.2 Elements of the movement/function line

Each movement/function line consists of the following basic elements:



(1) Executability, left-hand side

(2) Key/button/touch activated, left-hand side

(3) Execution, left-hand side

(4) Designation of the movements/functions

(5) Execution, right-hand side

(6) Key/button/touch 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. 7-4: Screens for manual operation – elements of a movement/function line

Designation (4)

The **Designation** element is the title for the movements/functions. The text items are configured in text lists of WinCC. The "Designation" element does not have a runtime interface.

Position (9)

The **Position** element 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** element indicates whether or not a movement can be performed.
If the movement 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 in the form of binary signals via the runtime interface.

If the movement line controls GRAPH steps, the executability is controlled automatically over the interlock.

Final status (end position) (8, 10)

The **Final status** element represents movement-specific or function-specific end positions in both directions (e.g. left/right, up/down, open/closed).
A square is not displayed as long as the target position has not yet been reached. When the target position is reached, the square is displayed in yellow (left-hand side) or green (right-hand size).

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

The status information must be supplied in the form of binary signals via the runtime interface.

Execution (3, 5)

The **Execution** element shows the status of the output that controls the respective movement/function.
No square is displayed if the output is disabled. When the output is enabled, the square is displayed in yellow (left-hand side) or green (right-hand size).

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

The status information must be supplied in the form of binary signals via the runtime interface.

Key/button/touch activated (2, 6)

The **Key/button/touch activated** elements indicate whether or not a key, button or the corresponding touch-sensitive area has been confirmed (processed) by the control program. The status information is supplied by the corresponding function block from HMI Lite.

- The **Key/button activated** element acts as follows for an HMI device with keys:
  When a key has been pressed and confirmed by the control, the movement side turns blue.
  The movement side remains gray if no button was pressed or if pressing of the button was not confirmed by the control.

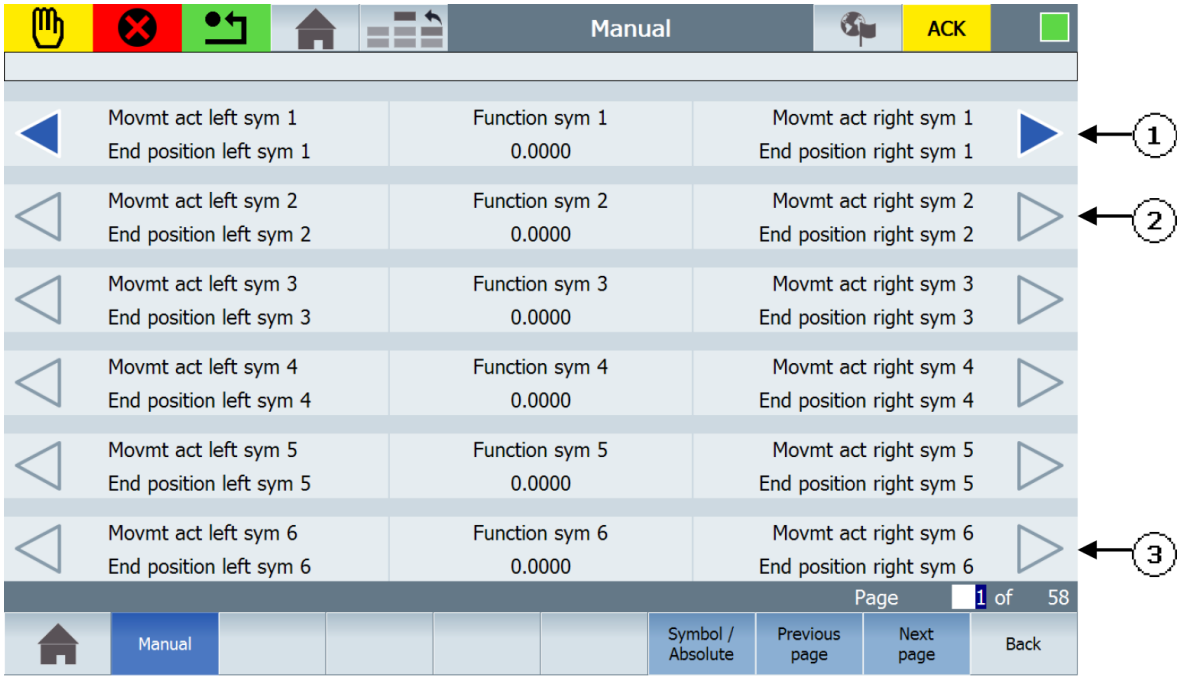
- The **Touch activated** element confirms the preselection or selection of a movement/function on the touch display with the following states:
  If a function has been preselected, this is indicated by the flashing of the **movement area**.
  If the touch-sensitive surface of a movement/function is activated a second time and this is confirmed by the control, the movement area is marked in blue.
  The movement side remains gray if a movement/function is not active or preselected.

- If an external key module is used, the **Key/button/touch** element flashes when a movement line has been selected.

For more details about the pre-selection please refer to Chapter 7.1.1 Layout and basic functionality of the manual operating screens > Movement and function line.

7.1.3 Assignment of the function numbers

Each displayed 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 across several pages.

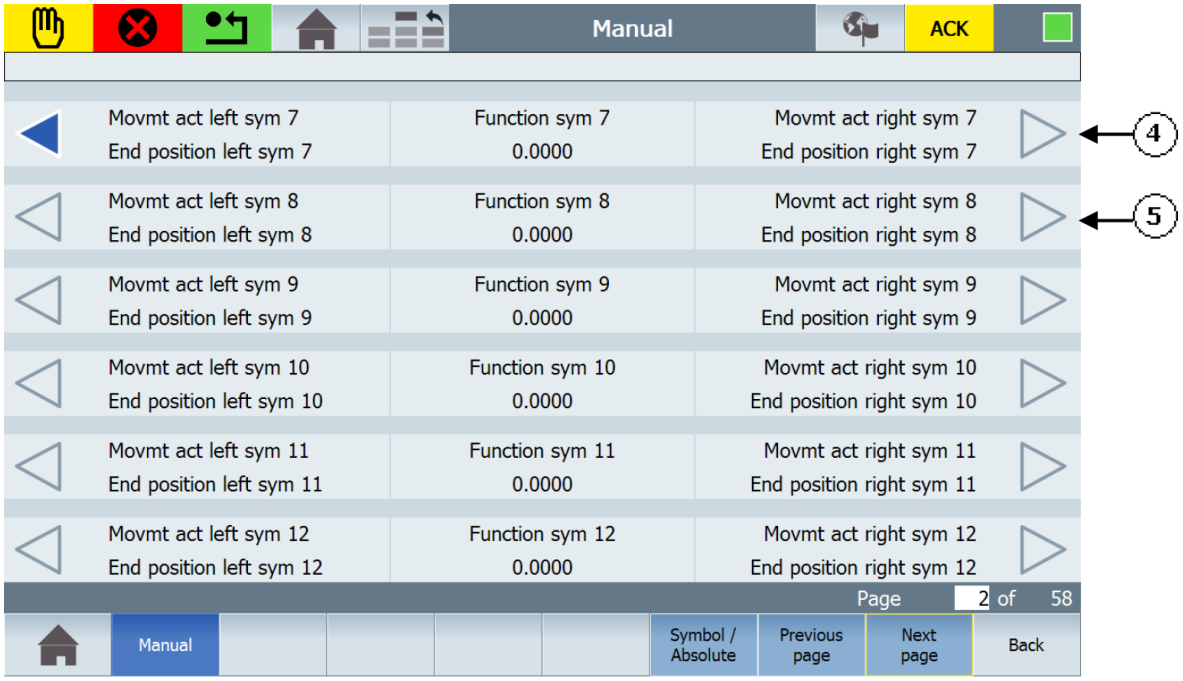


(1) Page 1, Line 1, Function 1

(2) Page 1, Line 2, Function 2

(3) Page 1, line 6, function 6

Fig. 7-5: Screens for manual operation - assignment of function numbers



(4) Page 2, line 1, function 7

(5) Page 2, line 2, function 8

Fig. 7-6: Screens for manual operation - assignment of function numbers

The page layout of the movements/functions is based on the following factors of HMI Lite:

- Total number of movements/functions that are configured in the selected screen
- Number of movements/functions that can be displayed on a page

Note

On a 12.1" or 15" HMI device, 6 movement/function lines can be displayed per screen page, on a 9” HMI device, 4.

7.2 Function of the manual operation screens

Setup

The **Setup screen** contains a maximum of 348 movement/function lines. This allows special movements to be performed manually using keys or touch. 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, for example at which position the movement currently is.

If more movements are configured than can be displayed on the screen at the same time, the movements are 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.

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 are displayed on several pages

Selection of units

The **Selection of 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 are displayed on several pages.

Nut runners

The "Nut runner" 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 runners than can be displayed on the screen at the same time, the nut runners are displayed on several pages.

Nut runner 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 are displayed on several pages.

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 are displayed on several pages.

User operating screen

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

7.3 Configuration and runtime interface

Each manual operating screen has its own text lists, parameter records and control interface. These parameters and text lists have the same basic structure and are defined using the name of the respective 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 7-1: Screens for manual operation – Assignment of the screens to the interface in the blocks

Name of the screen in WinCC	Name of the area in LTLL_Data and LTLL_Config
SS_02_001_Setup	screenSetup
SS_02_002_PowerUp	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

7.4 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.

7.4.1 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**: If **LTLL_Config.THIS[X].manualCommon.absoluteDisplayTime** is configured with 0, there is no automatic return to the symbolic view.

Address:	LTLL_Config.THIS[X].manualCommon.absoluteDisplayTime
Format:	TIME
Value range:	T#1MS…T#24D20H31M23S647MS
Default setting:	T#10S (10 s)

Touch operation preselection timeout status

The period 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
Value range:	T#1MS…T#24D20H31M23S647MS
Default setting:	T#2S (2 s)

7.4.2 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 tag in **LTLL_Config**.

Address:	LTLL_Config.THIS[X].screenAAAAAA.numberOfRows AAAAAA = name of the screen (see Table 7-1: Screens for manual operation – Assignment of the screens to the interface in the blocks)
Format:	INT
Value range:	1…348 for all manual operation screens
Default setting:	The maximum number of available lines

7.4.3 Grouping of the movement lines in the setup screen

In order to divide the **Setup** screen into function groups it is possible to configure the screen several times, each with different pages.

When the screen is selected, the first and the last relevant page must be entered for the tags specified below. This is done using the WinCC **SetValue** function that is configured in addition to the **ActivateScreen** function on the key or button that selects the setup screen.

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

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

Address:	WinCC tags: SS_02_001_setupScreenNumberOfLastPage (last page) SS_02_001_setupScreenNumberOfFirstPage (first page)
Format:	BYTE
Value range:	1…Max The maximum value depends on the number of movement lines and on the number of lines per page. See Chapter 7.4.2 Number of movement/function line Chapter 7.7 LTLL_Manual block Example: For 348 movement lines and 6 lines per page, this results in 58 pages with movement lines, consequently, the value range is 1...58.

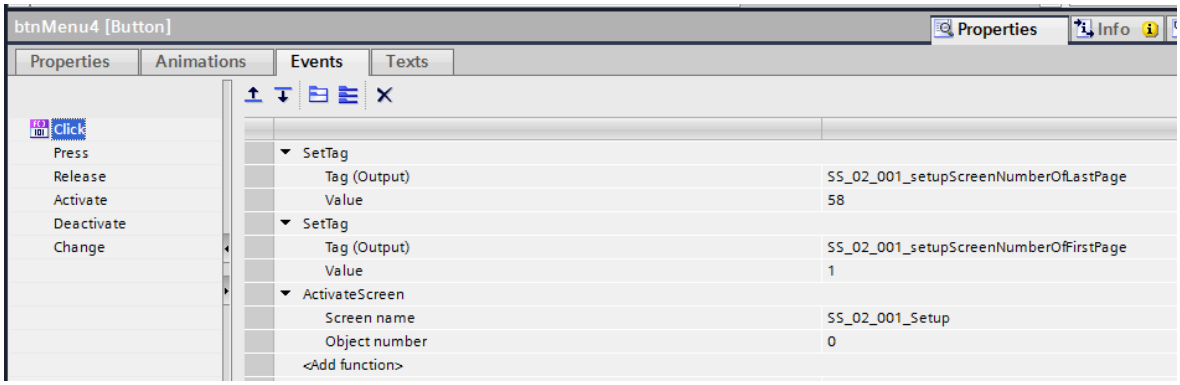


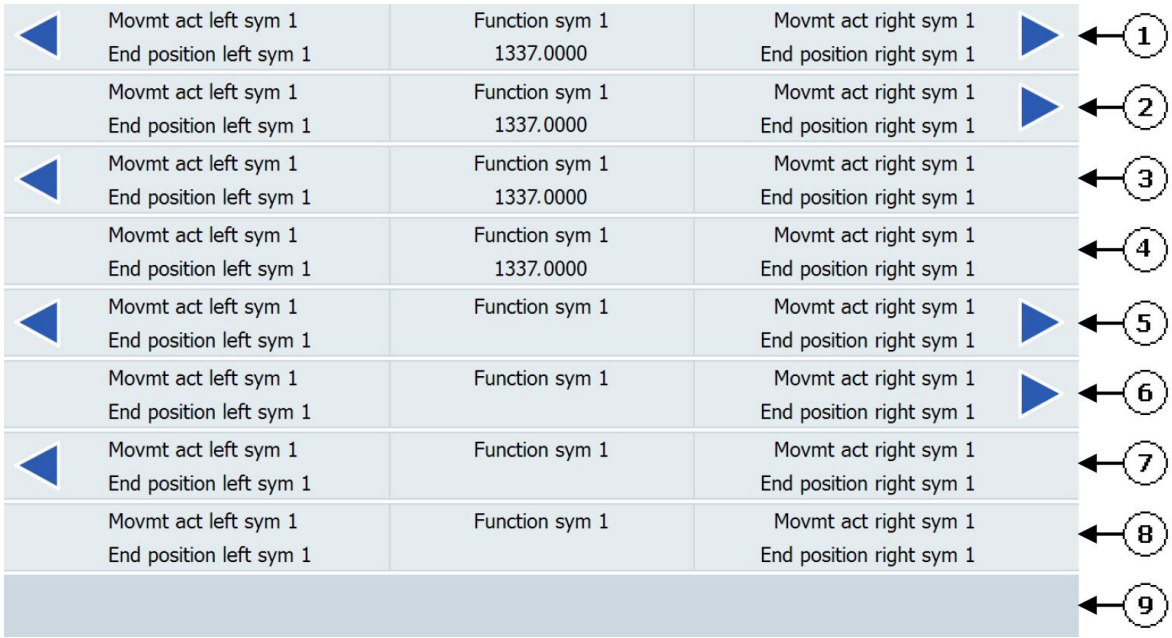
Fig. 7-7: WinCC configuration of the screen selection of the setup screen in groups

7.4.4 Hiding elements of the function line

It is possible to hide elements from the function line.

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

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



(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-hand and right-hand side are hidden.

(9) All elements are hidden.

Fig. 7-8: Screen for manual operation - 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] AAAAAA = Name of the screen (see Table 7-1: Screens for manual operation – Assignment of the screens to the interface in the blocks) Y = Number of the respective function line
Format:	LTLL_typeManualConfig
Value range:	-
Default setting:	-

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

configs[0].hiddenLeft	Configuration 1: Hidden left
configs[0].hiddenRight	Configuration 1: Hidden right
configs[0].hiddenPosition	Configuration 1: Position hidden
configs[0].hiddenAllOther	Configuration 1: All other elements hidden
configs[1].hiddenLeft	Configuration 2: Hidden left
configs[1].hiddenRight	Configuration 2: Hidden right
configs[1].hiddenPosition	Configuration 2: Position hidden
configs[1].hiddenAllOther	Configuration 2: All other elements hidden

Only one configuration can be active for all movements/functions at any one time. The associated active configuration (Configuration 1 or 2) can be selected dynamically using the **selectConfig** input parameter at 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).

Configuration examples

Several configuration examples follow:

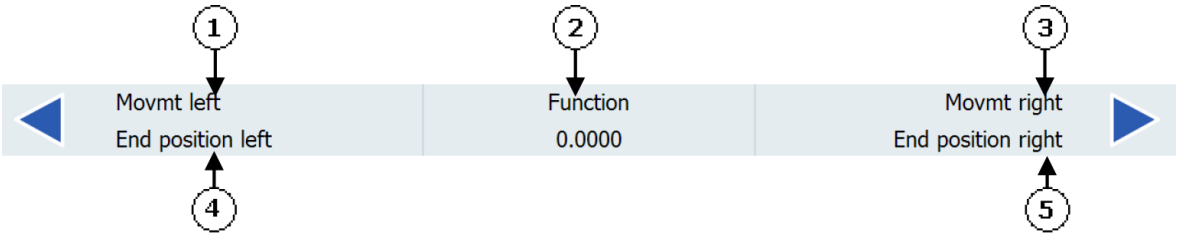
- The **Position** element is hidden for both configuration settings:
  Example: All types of machine elements (e.g. pumps, valves) that do not supply any confirmation of the position.
- The **Executable** element is hidden for both configuration settings:
  Example: Machine elements that are not controlled from the HMI device
  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 only be controlled in manual operation (for example machine axis)
  Only the status of these elements (for example **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.

⚠ WARNING

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. You have to realize any interlocking functionalities by means of the user program.

7.4.5 Display texts

All text items displayed in the manual operating screens are configured in the WinCC text lists. In this case each screen has its own text list. The text can be configured for each element.



(1) Feedback signal left

(2) Name

(3) Feedback signal right

(4) Final state left

(5) Final state right

Fig. 7-9: Screens for manual operation – text lists

All text lists have the same structure.

Table 7-2: Screens for manual operation - structure of 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 – feedback message left – symbolic
Value	13	Line #1 – feedback message left – absolute
Value	14	Line #1 – final state left – symbolic
Value	15	Line #1 – final state left – absolute
Value	16	Line #1 – feedback message right – symbolic
Value	17	Line #1 – feedback message right – absolute
Value	18	Line #1 – final state right – symbolic
Value	19	Line #1 – final state right – absolute
Value	20	Line #2 - function name – symbolic
Value	21	Line #2 - function name – absolute
	…	

Two text list positions are assigned to each screen element:

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 configured 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. Correspondingly the display texts are as follows:

Table 7-3: Screens for manual operation – example for display texts

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	Move active left sym 1
"Execution left" text for the absolute view	Q1.0
"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	Move active 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 values of the text lists for the manual operating screens have the following structure:

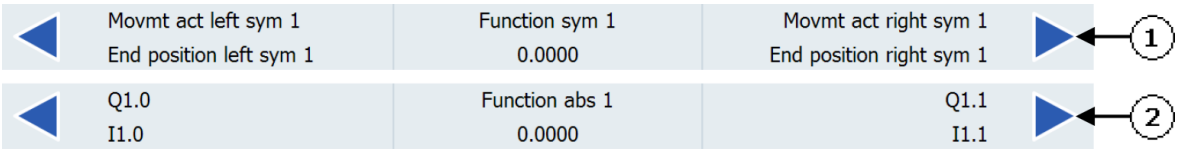
Tens, hundreds, thousands digit: Movement/function line

Units digit: Identifier of the movement/function text

Table 7-4: Screens for manual operation – example of a text list

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

The configured movement is displayed as follows:



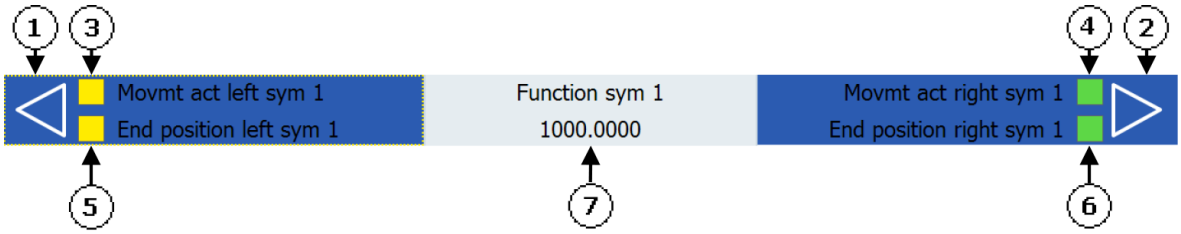
(1) Symbolic view

(2) Absolute view

Fig. 7-10: Screens for manual operation - example for configuring a text

7.5 Runtime interface

Color changes show the details of the binary status of a movement/function. The **Position** element shows a numeric position value.



(1) Executable (interlock), left-hand side

(2) Executable (interlock), right-hand side

(3) **Being performed/Moving** feedback message, left-hand side

(4) **Being performed/Moving** feedback message, right-hand side

(5) Final state/end position, left-hand side

(6) Final state/end position, right-hand side

(7) Position

Fig. 7-11: Screens for manual operation – dynamic movement elements

Information about the binary state

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

Address:	LTLL_Data.HMI[X].screenAAAAAA.rows[Y] AAAAAA = Name of the screen (see Table 7-1: Screens for manual operation – Assignment of the screens to the interface in the blocks) Y = Number of the associated function
Format:	LTLL_typeManualData
Value range:	-
Default setting:	-

Each grouping element represents a movement/function.

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

⚠ WARNING

The bits in the data interface under controlInterface are used as control signals (HMI device > Control). 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 may never be active concurrently for a single movement. Otherwise this would give the impression that the movement would be performed at the same time in both directions.
- The two **End status** displays may never be active concurrently for a single movement. This would give the impression that the movement had reached both end positions (at opposite directions) at the same time.
- The two **Executability** and **End status** displays may never be active concurrently for a single movement.
  Otherwise this would give the impression that the movement is executable although the final position has already been reached.
- The **Executability** and **End status** displays may never be active concurrently for a single movement.
  This would indicate that a movement/function is currently active although the final position has already been reached.

Selected screen

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

Address:	LTLL_Data.HMI[X].global.readOnly.screenID
Format:	WORD
Value range:	W#16#0000 … W#16#FFFF (0…65535) For the identification of the screen (see Table 5-2: Identification code for individual screens)
Default setting:	-

Current page

The active page can be determined using the following data address in the **LTLL_Data** data block:

Address:	LTLL_Data.HMI[X].manualCommon.readOnly.currentPage
Format:	UInt
Value range:	0 to 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
Value range:	W#16#0001…W#16#015C (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.

NOTICE
The tags under a **readOnly** structure are internal tags and may only be used with read access.

7.6 Controller interface

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

- By using the keys indicated by the corresponding triangle symbol
- 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 HMI device to the control. 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/function.

7.6.1 Job mailbox

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

Address:	LTLL_Data.HMI[X].global.job
	-number
	-parameter_1
	-parameter_2
	-parameter_3
Format:	WORD
Value range:	W#16#0000…W#16#FFFF
Default setting:	-

When the HMI device initiates a movement/function (for example, an operator presses a key at the left or right of the movement), the following information is displayed in the **job mailbox**:

Order number:	Screen identification code (see Table 5-2: Identification code for individual screens)
Parameter 1:	Number of the movement/function
Parameter 2:	Direction of movement: W#16#0001: Movement **to the right** (bit 0) W#16#0002: Movement **to the left** (bit 1)
Parameter 3:	Reserved for internal use

The code for identifying the screen (**job number** parameter in the job mailbox) is described below for the manual operating screens:

Table 7-5: Operating screens - code for identifying the screen in the job mailbox

Screen	Identification code of the respective 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 is displayed in the **job mailbox**:

Order number:	W#16#0201	Screen **SS_02_001_Setup**
Parameter 1:	W#16#0002	**Second movement line** function
Parameter 2:	W#16#0002	Direction **to the left**

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

NOTICE
The job number is not cleared when the operator releases a key used to initiate a movement. The job number is set as soon as 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.

7.6.2 Binary controller interface

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

Address:	LTLL_Data.HMI[X].screenAAAAAA.rows[Y].controlInterface AAAAAA = Name of the screen (see Table 7-1: Screens for manual operation – Assignment of the screens to the interface in the blocks) Y = Number of the associated function
Format:	LTLL_typeManualDataControl
Value range:	-
Default setting:	-

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

When a movement or function is initiated from the HMI device (for example, when the operator presses a key assigned to a function line), the control bits are set. The control bit is reset when the function key is released.

7.7 LTLL_Manual block

The **LTLL_Manual** block includes the following functionality:

- Scrolling in the manual operating screens when more movements/functions have been specified than can be displayed on the screen.
- Switching between the symbolic and the absolute representation
- Switching between the first and the second configuration of the movement/function line
- Representation of the key signals on the control interfaces
- Monitoring of the connection between the HMI device and the control.
- Interlocking the signals for the HMI device with keys or the interfaces of the touch HMI device or the direct keys.

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

Call interface

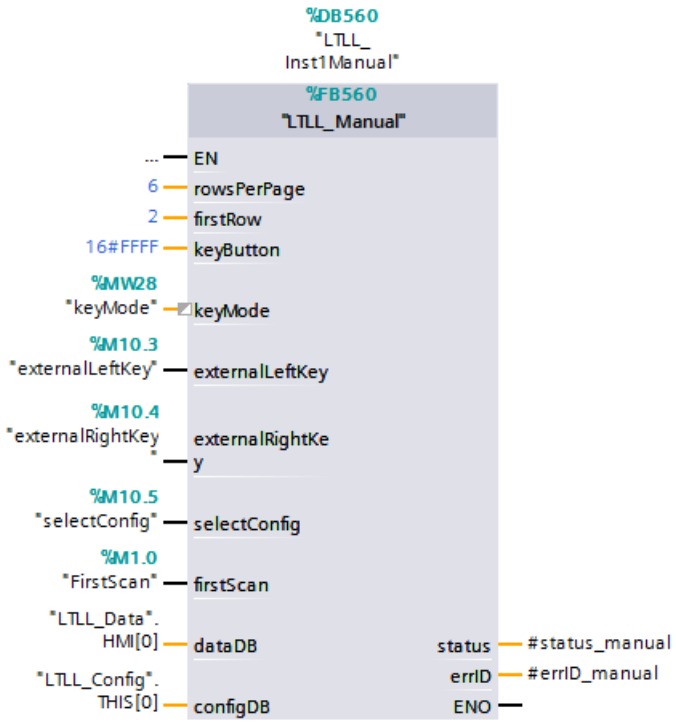


Fig. 7-12: Call interface for block LTLL_Manual

Parameters

Table 7-6: Description of the parameters of LTLL_Manual

<table>
<tr><th>Name</th><th>Declaration</th><th>Type</th><th>Standard</th><th>Description</th></tr>
<tr><td>rowsPerPage</td><td>Input</td><td>USInt</td><td>6 or 4</td><td>Number of function lines that can be displayed on the screen at the same time.

4 lines for the 9" HMI device with header

6 lines for 12.1" or 15" HMI devices with header</td></tr>
<tr><td>firstRow</td><td>Input</td><td>USInt</td><td>2</td><td>Function keys located at the side to be used for the first movement line

1 = first movement is performed by F1 and F2.
(optional) Operation without header

2 = first movement is performed by F3 and F4.
Operation with header</td></tr>
<tr><td>keyButton</td><td>Input</td><td>WORD</td><td>-</td><td>Assignment of the input word of the I/O device direct keys:
<table><tr><th>Bit</th><th>Line</th><th>Key</th></tr><tr><td>0</td><td>1</td><td>Left</td></tr><tr><td>1</td><td>1</td><td>Right</td></tr><tr><td>2</td><td>2</td><td>Left</td></tr><tr><td>3</td><td>2</td><td>Right</td></tr><tr><td>4</td><td>3</td><td>Left</td></tr><tr><td>5</td><td>3</td><td>Right</td></tr><tr><td>6</td><td>4</td><td>Left</td></tr><tr><td>7</td><td>4</td><td>Right</td></tr><tr><td>8</td><td>5</td><td>Left</td></tr><tr><td>9</td><td>5</td><td>Right</td></tr><tr><td>10</td><td>6</td><td>Left</td></tr><tr><td>11</td><td>6</td><td>Right</td></tr></table>If no direct keys keys are used, the value W#16#FFFF must be specified.</td></tr>
</table>

Name	Declaration	Type	Standard	Description
keyMode	Input	USInt	-	Mode for executing the operation see 7.1.1 Layout and basic functionality of the manual operating screens 0: Softkeys direct Function active while a key remains pressed 1: Touch direct Function active while a button remains pressed 2: Touch pre-selection Function active after 2nd press of the button 3: Softkeys external Function active while an external key remains pressed; selection of the function by the function keys located at the side 4: Touch external: Function active while an external key remains pressed; selection of the function by the function keys located at the side
externalLeftKey	Input	BOOL	-	Only relevant in the 3 and 4 key modes. Performs the left command of the selected function.
externalRightKey	Input	BOOL	-	Only relevant in the 3 and 4 key modes. Performs the right command of the selected function.
selectConfig	Input	BOOL	-	Switch between the two configurations for hiding of individual elements of the function line. FALSE = Configuration 1 TRUE = Configuration 2
firstScan	Input	BOOL	-	Restart flag 1-signal for the first cycle after CPU startup

Name	Declaration	Type	Standard	Description
dataDB	InOut	LTLL_typeData	LTLL_Data.HMI[0]	HMI Lite Runtime data DB
configDB	InOut	LTLL_typeConfig	LTLL_Config.THIS[0]	HMI Lite Configuration DB
status	Output	WORD	-	Block status
errId	Output	WORD	-	Local error handling

Output parameter status

Table 7-7: Description of the output parameter status of LTLL_Manual

Error code (W#16# ….)	Description
16#8200	HMI Lite licensing failed

Parameter for external key mode

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

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

Address: LTLL_Data.HMI[X].manualCommon.closedSelectedRow

Format: BOOL

Value range: -

Default setting: -

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

Address: LTLL_Data.HMI[X].manualCommon.resetSelectedRow

Format: BOOL

Value range: -

Default setting: -

7.8 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. 7-13: Call interface for block LTLL_ManualControl

Parameters

Table 7-8: Description of the parameters of LTLL_ManualControl

Name	Declaration	Type	Standard	Description
useGraphForManualRow	Input	BOOL	-	TRUE: Manual operation line controls the GRAPH step FALSE: Manual operation line does not control a GRAPH step
hiddenLeft1	Input	BOOL	-	Hide left configuration 1
hiddenRight1	Input	BOOL	-	Hide right Configuration 1
hiddenPosition1	Input	BOOL	-	Hide position configuration 1
hiddenAllOther1	Input	BOOL	-	Hide all other elements configuration 1
scaleNumber1	Input	USInt	-	Not used
hiddenLeft2	Input	BOOL	-	Hide left configuration 2
hiddenRight2	Input	BOOL	-	Hide right Configuration 2:
hiddenPosition2	Input	BOOL	-	Hide position configuration 2
hiddenAllOther2	Input	BOOL	-	Hide all other elements configuration 2
scaleNumber2	Input	USInt	-	Not used
executabilityLeft	Input	BOOL	-	Executability left
executabilityRight	Input	BOOL	-	Executability right
confirmExecuteLeft	Input	BOOL	-	Left movement active
confirmExecuteRight	Input	BOOL	-	Right movement active
finalPositionLeft	Input	BOOL	-	End position left
finalPositionRight	Input	BOOL	-	End position right
position	Input	LReal	-	Position
dataDB	InOut	LTLL_typeData	LTLL_Data. HMI[0]	HMI Lite Runtime data DB
rowConfig	InOut	LTLL_typeManual Config	-	Configuration of a manual operation line
rowData	InOut	LTLL_typeManual Data	-	Runtime data of a manual operation line
leftFunctionActive	Output	BOOL	-	Left function activated

Name	Declaration	Type	Standard	Description
rightFunctionActive	Output	BOOL	-	Right function activated
scaleNumberOK	Output	BOOL	-	Not used
errID	Output	WORD	-	Local error ID
Return	Return	WORD		Return value for block

Function return value

Table 7-9: Description of the return value of LTLL_ManualControl

Error code (W#16# ….)	Description
16#8200	HMI Lite licensing failed

7.9 LTLL_ManualGraph block

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

- Display of the executability of the movements, which are 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

Block **LTLL_ManualGraphExt** requires Graph version 5.0.

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

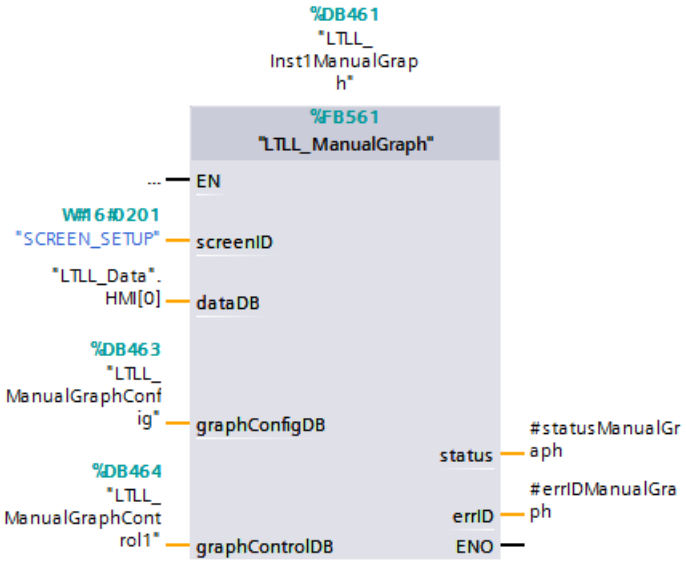


Fig. 7-14: Call interface for block LTLL_ManualGraph

Parameters

Table 7-10: Description of the parameters of LTLL_ManualGraph

Name	Declaration	Type	Standard	Description
screenID	Input	WORD	W#16#0201	Screen ID of the operating screen for which the FB call is to be valid
dataDB	InOut	LTLL_typeData	LTLL_Data. HMI[0]	HMI Lite Runtime data DB
graphConfigDB	InOut	LTLL_typeManual GraphConfig	LTLL_Manual GraphConfig	HMI Lite sequencer operating screen configuration
graphControlDB	InOut	LTLL_typeManual GraphControl	LTLL_ManualGraph Control	Interface DB for sequencer control in HMI Lite manual operating screens
status	Output	WORD	-	Block status
errID	Output	WORD	-	Local error ID

Output parameter status

Table 7-11: Description of the output parameter status of LTLL_ManualGraph

Error code (W#16# ….)	Description
8200	HMI Lite licensing failed
8201	Invalid screen ID

Functionality of the FB

If an operating screen is selected on the HMI device, 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 chapter 7.1.1 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 tags must be parameterized in the **LTLL_ManualGraphConfig** data block:

Address:	LTLL_ManualGraphConfig.row[X].left.dbInstanceName X = the number of the line of the operating screen
Format:	WString[125]
Value range:	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 X = the number of the line of the operating screen
Format:	Int
Value range:	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 X = the number of the line of the operating screen
Format:	WString[125]
Value range:	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 X = the number of the line of the operating screen
Format:	Int
Value range:	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. Enter block **LTLL_ManualGraphExt** in the properties of the GRAPH FBs under **General > Attributes > Extension block name**.

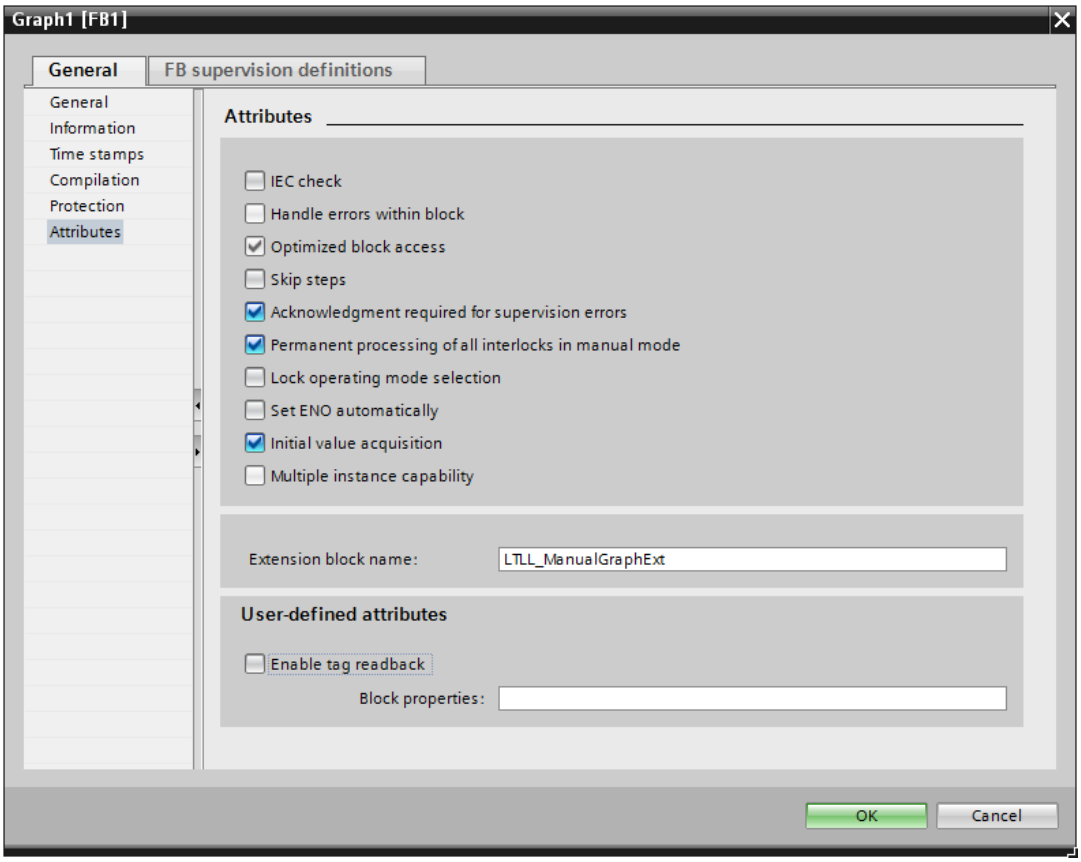


Fig. 7-15: Specify extension block LTLL_ManualGraphExt

2. Add a tag of the type **LTLL_ManualGraphExt** in the static parameters of the sequencer FBs.

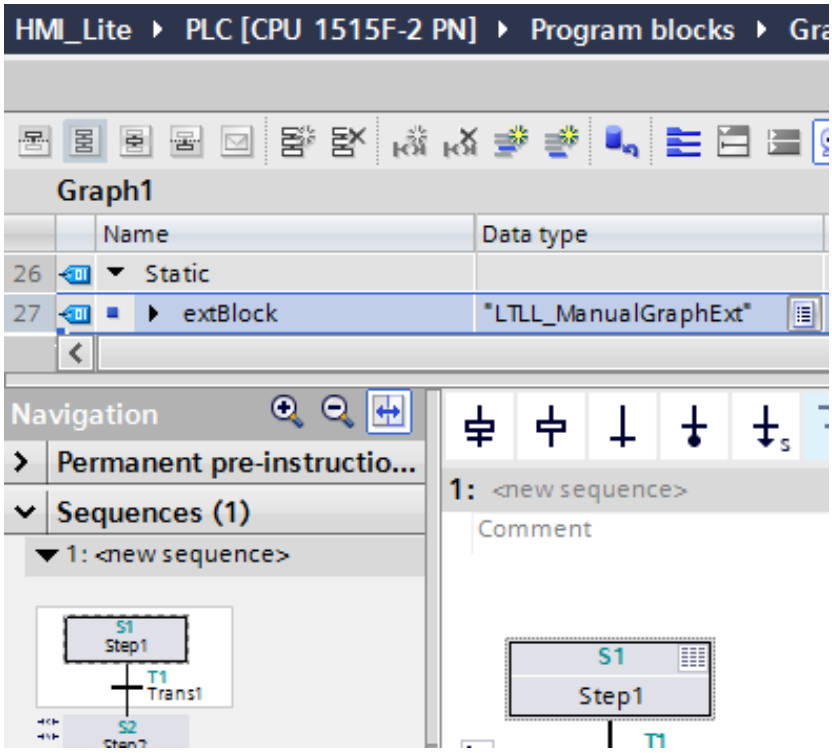


Fig. 7-16: Adding a tag, type LTLL_ManualGraphExt

3. In the upstream permanent instructions of the Graph sequencer, route the data of the **LTLL_ManualGraphControl** to the static tag **Input** of the tag of the type **LTLL_ManualGraphExt** created above.
Make sure that you are using the correct **screen** area:

screen[0]: screenSetup (SS_02_001_Setup)

screen[1]: screenPowerUp (SS_02_002_PowerUpCondition)

screen[2]: screenUnit (SS_02_003_Unit)

screen[3]: screenNutRunner (SS_02_004_NutRunner)

screen[4]: screenNutRunnerGroup (SS_02_005_NutRunnerGroup)

screen[5]: screenCycleType (SS_02_006_Cycletype)

screen[6]: screenUserDefine (SS_02_007_UserDefine)

4. Call the instruction **GetInstanceName** and transfer the result to the tag of the type **LTLL_ManualGraphExt.input.instanceName**

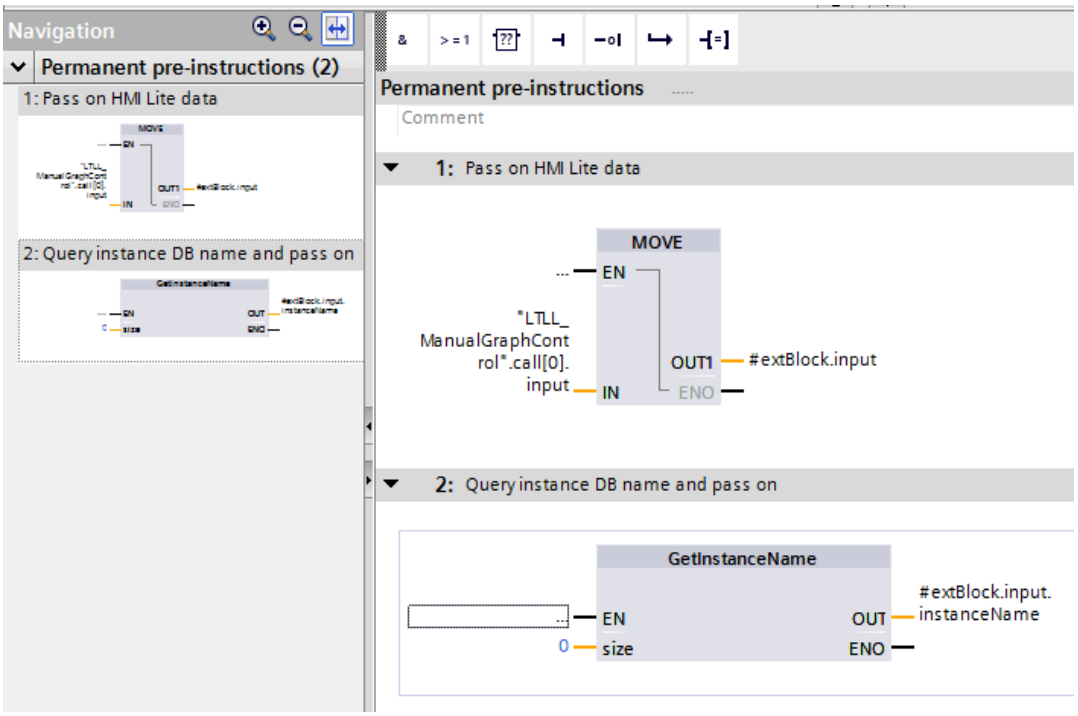


Fig. 7-17: Data transfer and call instruction GetInstanceName

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

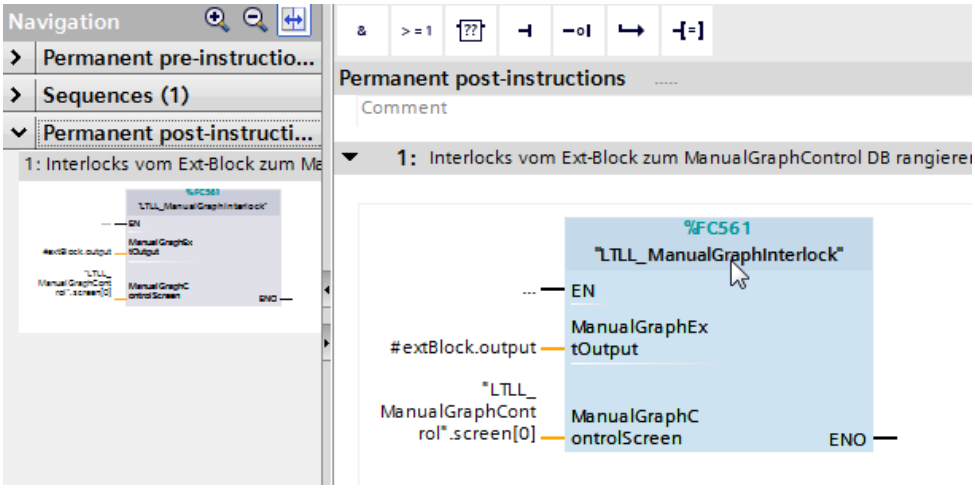


Fig. 7-18: Data transfer to LTLL_ManualGraphControl.call[x].output

6. Repeat the steps for all GRAPH sequencers in which steps are controlled by the manual operating screens.

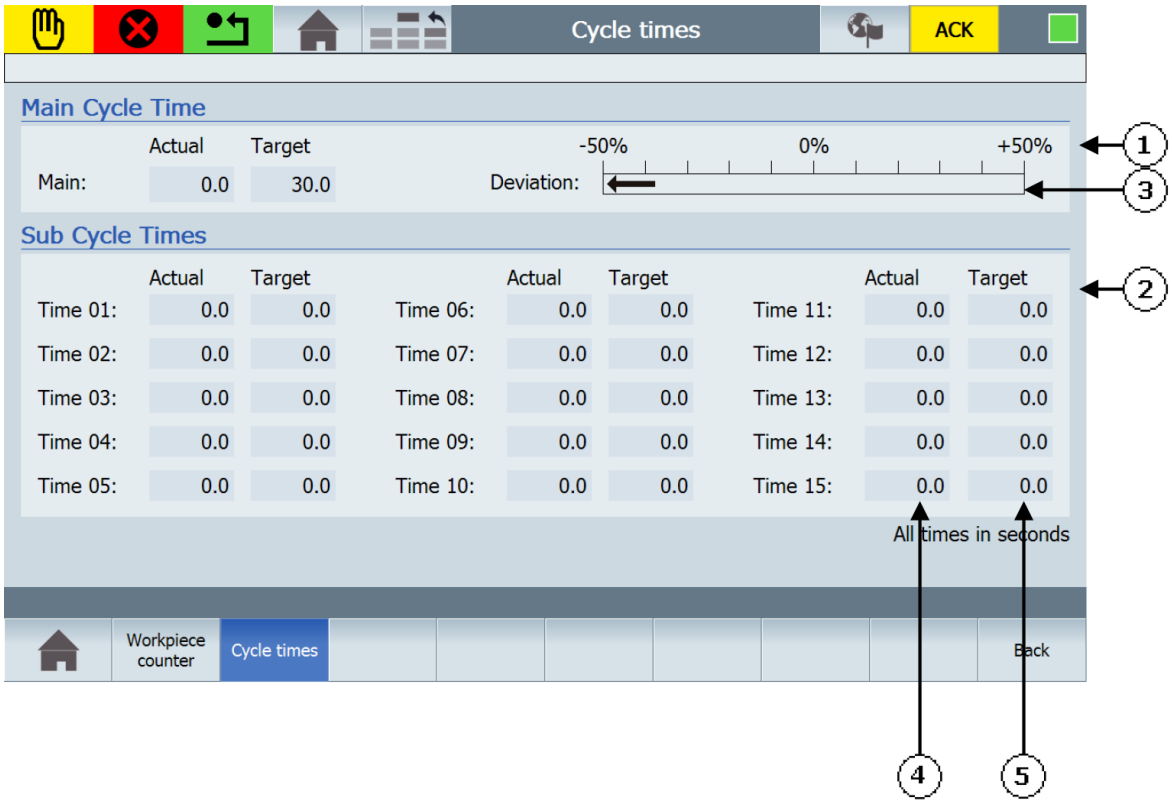
Production data screens

8

8.1 Cycle times

8.1.1 Layout and functionality

The **Cycle times** screen displays the total cycle time and the single cycle times of the machine.



(1): Total clock time area

(2): Single cycle times area (can be hidden)

(3): Deviation of the total cycle time as percentage (± 50%) from the set cycle time

(4): Actual clock time values

(5): Set clock time values

Fig. 8-1: Cycle times (SS_04_021_CycleTimes)

Screen elements

The screen is subdivided into the following two main areas:

- Total cycle time
- Single cycle times

The **Total cycle time** area displays the values for the actual cycle time and the setpoint cycle time. In addition, the deviation between the actual and setpoint cycle time is output as a percentage. The range of the cycle time deviation is limited to ± 50%. If the deviation lies outside this range, this is indicated by arrows at the left-hand or right-hand side of the bar.

Note

The deviation is calculated using the following equation:
Deviation = Actual cycle time / Setpoint cycle time
Only the first 6 single cycle times are displayed on the KTP900F Mobile.

Procedure for the cycle times

The cycle time is to be calculated with the start and the end signal of a cycle or single cycle. This value represents the actual cycle time and is updated when it is redefined. The interruption of a cycle 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 cycle time to be continued.

Value range

The values of the cycle times are entered in 32-bit integer tags with sign. The values are displayed in seconds with one decimal place. The displayed cycle time resolution corresponds to a tenth of a second.

The max. indicated value is 214,748,364.7 seconds.

The accuracy of the timer depends on the type of the control used. You will find more detailed information in the documentation for the S7 CPU data.

Reduced display functions

The **Single cycle times** area can be hidden. This function is controlled by using the configuration parameter **LTLL_Config.THIS[X].screenCycletime.hideSpecific**. In this case, the complete control field with the single cycle times is hidden.

8.1.2 Runtime interface (LTLL_Cycletime)

Calculation of the cycle times is realized with the **LTLL_Cycletime** block. A total of 16 cycle times can be acquired. Each cycle time acquisition can be started and stopped independently of other cycle time acquisitions. The first acquisition is used for the total cycle time. The other 15 are used for the single cycle times. No timers are used.

Call interface

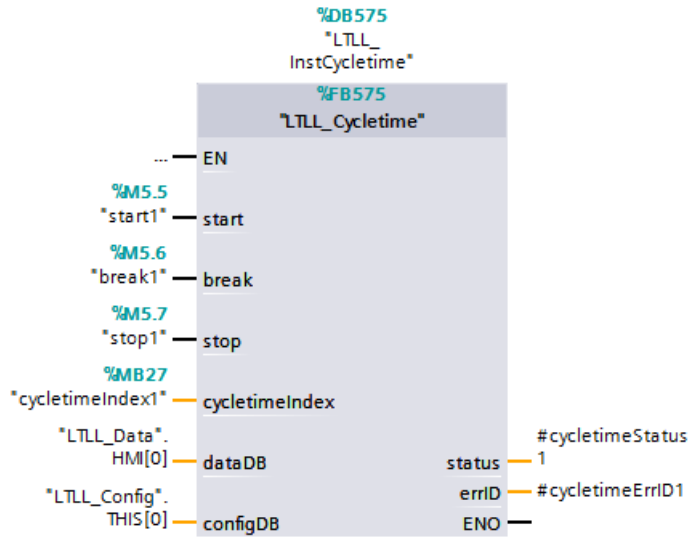


Fig. 8-2: Call interface of block LTLL_Cycletime

Parameters

Table 8-1: Parameters of block LTLL_Cycletime

Name	Declaration	Type	Standard	Description
start	Input	BOOL	---	A positive edge starts the cycle time selected by the **cycletimeIndex** parameter
break	Input	BOOL	---	A positive edge interrupts the counting, a negative edge continues the cycle time measurement
stop	Input	BOOL	---	A positive edge stops the cycle time selected by the **cycletimeIndex** parameter
Cycletime index	Input	USInt	---	Select the cycle time to be measured. cycletimeIndex=0 : Total cycle time cycletimeIndex=1- 15: Single cycle time 1 - 15
dataDB	InOut	LTLL_typeData	LTLL_Data. HMI[0]	HMI Lite Runtime data DB
configDB	InOut	LTLL_typeConfig	LTLL_Config. THIS[0]	HMI Lite Configuration DB
status	Output	WORD	---	Block status
errId	Output	WORD	---	Local error handling

Output parameter status

Table 8-2: Description of the output parameter status of LTLL_Cycletime

Value (V#16# ….)	Description
16#8200	HMI Lite licensing failed
16#8201	Invalid cycletimeIndex

Note

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

8.1.3 Configuration

LTLL_Config

The area for the single cycle times can be hidden by setting the following tags:

Address:	LTLL_Config.THIS[X].screenCycletime.hideSpecific
Format:	BOOL
Value range:	FALSE: Single cycle times are displayed
	TRUE: Single cycle times are hidden
Default setting:	FALSE

A setpoint cycle time can be defined for the total cycle time:

Address:	LTLL_Config.THIS[X].screenCycletime.main.target
Format:	DINT
Value range:	-2_147_483_648 to +2_147_483_647
Default setting:	300 (30.0 seconds)

A setpoint cycle time can be defined for each single cycle time:

Address:	LTLL_Config.THIS[X].screenCycletime.sub.target[XX] (where XX is the number of the corresponding single cycle time: 1..15)
Format:	DINT
Value range:	-2_147_483_648 to +2_147_483_647
Default setting:	300 (30.0 seconds)

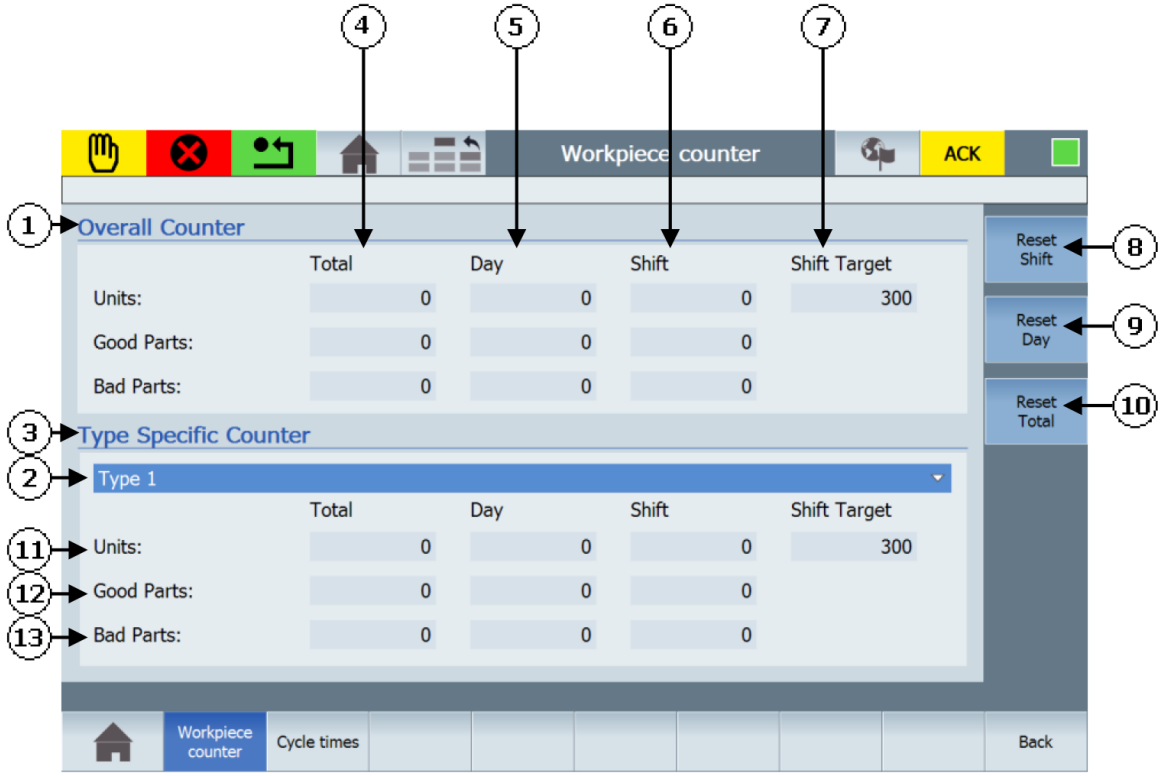
How to configure **LTLL_Config**:

1. Open the **LTLL_Config** data block.
2. Set the tag **screenCycletime.hideSpecific** to
   FALSE when the single cycle times are to be displayed
   TRUE when the single cycle times are to be hidden
3. Specify the values for the setpoint cycle times by editing the tags **screenCycletime.main.target** and **screenCycletime.sub.target[XX]**.
4. Load the data block to the control.
5. Save and close the **LTLL_Config** data block.
6. Open the **LTLL_HMILite** organization block.
7. Call the **LTLL_Cycletime** block and assign the required parameters.
8. Save and and close the **LTLL_HMILite** block.
9. Load all changed blocks to the control.

8.2 Workpiece counter

8.2.1 Layout and functionality

The **Workpiece counter** screen is used to display produced workpieces.



(1)Total workpiece counter
(2) Select the workpiece type
(3) Workpiece type-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) Setpoint workpiece count during a shift
(8) Reset shift counter
(9) Reset day counter
(10) Reset total counter
(11) Total number of produced parts
(12) Total number of produced parts that are OK
(13) Total number of produced parts that are not OK

Fig. 8-3: Workpiece counter (SS_04_011_PartCounter)

Screen elements

The screen is subdivided into the following two main areas:

- Total Workpiece Counter
- Type Workpiece Counter

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

These subareas are subdivided into the following counter values:

- Units:
  Total workpiece counter (good and bad parts)
- Good Parts:
  Workpiece counter good parts
- Bad Parts:
  Workpiece counter bad parts

If setpoints are specified for the shift counter (value greater than 0), these are displayed in the output fields **Shift setpoint** for the planned workpiece number. Otherwise these are hidden.

The values for the type-specific workpiece counters can be selected using a selection list. Up to 3500 workpiece-related part counters can be configured. 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 tags in the **LTLL_Data** data block must be updated using the following equations:

Total workpieces = Total workpieces old + Number of produced parts

Total bad parts = Total bad parts old + Number of produced bad parts

This has to be carried out for the total, day and shift counters of the total unit counter as well as workpiece-related unit counters at the same time.

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

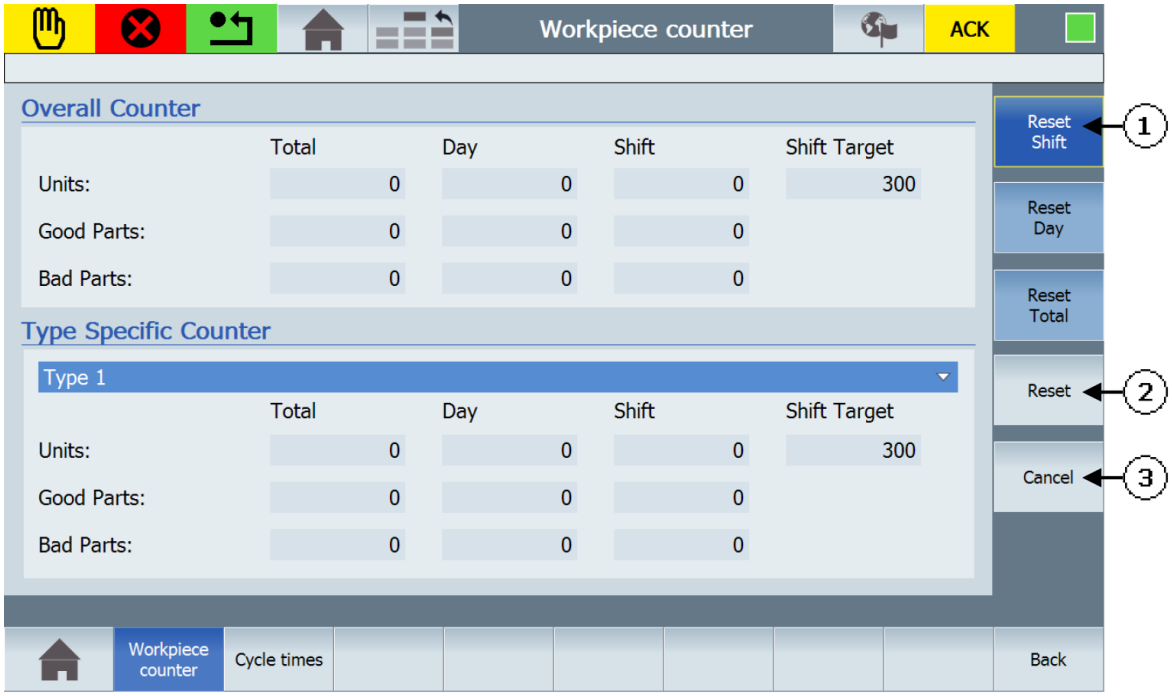
Procedure for resetting

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 buttons can also be used to initiate a manual reset. If required, the Reset buttons can be hidden by setting the appropriate configuration bits in the **LTLL_Config**.

Pressing a Reset button initiates the provided confirmation procedure.



(1)The reset button that was activated is marked in color.

(2) Button for confirming the reset operation

(3) Button for canceling the reset operation

Fig. 8-4: Workpiece counter – procedure for confirming the reset

Value range

Value ranges of the counters:

Total – sum of good/bad parts:	0 to 18,446,744,073,709,551,615
Total - bad parts:	0 to 4,294,967,295
Day – sum of good/bad parts:	0 to 4,294,967,295
Day - bad parts:	0 to 4,294,967,295
Shift – sum of good/bad parts:	0 to 4,294,967,295
Shift - bad parts:	0 to 4,294,967,295

Unit counter with reduced display functions

The **Type Workpiece Counter** area can be hidden. This function is controlled by means of the **hideTypeSpecific** configuration parameter in the **LTLL_Config** configuration data block. Hiding applies to the complete workpiece-type-related parts counter.

8.2.2 Runtime interface (LTLL_Counter)

The **LTLL_Counter** block uses the workpiece counter tags of the data blocks **LTLL_Data** and **LTLL_CounterData**.

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

Call interface

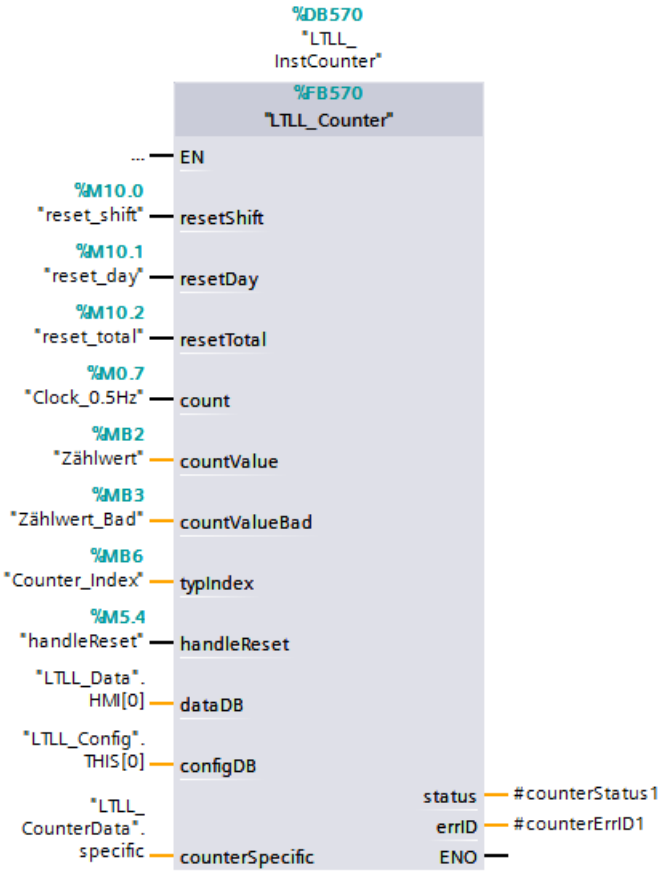


Fig. 8-5: Call interface of block LTLL_Counter

Parameters

Table 8-3: Time parameter of LTLL_Counter

Name	Declaration	Type	Standard	Description
resetShift	Input	BOOL	---	A rising edge resets the shift counters.
resetDay	Input	BOOL	---	A rising edge resets the day counters.
resetTotal	Input	BOOL	---	A rising edge resets the total counters.
count	Input	BOOL	---	A rising edge updates the counter.
countValue	Input	BYTE	---	The number of the total parts to be counted (good + bad parts)
countValueBad	Input	BYTE		Number of bad parts to be counted
typIndex	Input	WORD	---	The index of the workpiece type to be counted. If value = 0, only the total unit counter is processed. Values less than 0 will cause an error message.
handleReset	Input	BOOL	---	Enable of the unit counter Reset buttons and buttons in the screen (workpiece counter) The Reset function is only carried out with a 1-signal of the parameter.
counterSpecific	InOut	Array[*] of LTLL_typeCounter	LTLL_CounterData.specific	DB in which the Type Workpiece Counter is stored
dataDB	InOut	LTLL_typeData	LTLL_Data.HMI[0]	HMI Lite Runtime data DB
configDB	InOut	LTLL_typeConfig	LTLL_Config.THIS[0]	HMI Lite Configuration DB
status	Output	WORD	---	Block status
errId	Output	WORD	---	Local error handling

Output parameter status

Table 8-4: Description of the output parameter status of LTLL_Counter

Value (V#16# ….)	Description
16#0001	Only total counter
16#8200	HMI Lite licensing failed
16#8201	Invalid typIndex
16#8400	Invalid selection in the screen

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

[New counter value] = [Old counter value] + [Counter value].

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

[Good parts value] = [Total parts value] - [Bad parts value].

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

The **typIndex** parameter specifies which workpiece-related counter is 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 is updated.

The **resetDay, resetShift, resetTotal** reset parameters 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 a rising edge at the reset parameters and at the **count** parameter is detected, the counter function and then the Reset function is performed. Within a cycle, it is possible to reset the total counter, the day counter and shift counter. Usage of the control-internal Reset functions as well as the reset functions of the user interface is not mutually exclusive. (For example it is possible to use the **resetShift** parameter even when the **handleReset** parameter carries a 1-signal at the same time.)

Note

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

8.2.3 Configuration

LTLL_Config

The area for the workpiece-type-specific unit counter can be hidden by setting the following tags:

Address:	LTLL_Config.THIS[X].screenCounter.hideTypeSpecific
Format:	BOOL
Value range:	FALSE: The **Type Unit Counter** is displayed
	TRUE: The **Type Unit Counter** is hidden
Default setting:	FALSE

The Reset buttons can be displayed and deactivated by setting the following tags, for example, when resetting is to be carried out automatically by means of the user program.

Address:	LTLL_CONFIG.THIS[X].screenCounter.hideResetShift
	LTLL_CONFIG.THIS[X].screenCounter.hideResetDay
	LTLL_CONFIG.THIS[X].screenCounter.hideResetTotal
Format:	BOOL
Value range:	FALSE = The corresponding Reset button is active and displayed.
	FALSE = The corresponding Reset button is not active and hidden.
Default setting:	FALSE

The number of parts to be produced in the current shift (setpoint) is to be configured in **LTLL_Config.THIS[X]** and **LTLL_CounterData**. The total setpoint (sum of the setpoints for all parts) and the workpiece-related setpoint 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
Value range:	0 to 4,294,967,295
Default setting:	300

It can be specified in the following tags 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
Value range:	T#1MS…T#24D20H31M23S647MS
Default setting:	T#5S (5s)

The reset function is 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 8-5: 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

Step-by-step procedure for configuring a workpiece counter:

1. Open the **LTLL_Config** data block.
2. Set the **screenCounter.hideTypeSpecific** tag to
   FALSE when the type-specific counters are to be displayed
   or to
   TRUE when the type-specific counters are to be hidden.
3. Specify the shift setpoint by editing the tags for
   LTLL_Config.THIS[X].screenCounter.overall.shiftTarget (total counter) and
   LTLL_CounterData.specific[x].shiftTarget.
4. Load the data blocks to the control.
5. Save and close the data block.
6. Open the **LTLL_HMILite** organization block.
7. Call the **LTLL_Counter** block and assign the required parameters.
8. Save and and close the **LTLL_HMILite** block.
9. Load all changed blocks to the control.
10. Use WinCC to open the WinCC file from HMI Lite.
11. Edit the **SO_04_011_PartCounterType** text list.
12. Enter meaningful designations for the workpiece types at the corresponding positions.
13. Delete all the text entries that are not used.
14. Save the WinCC project.
15. Compile the WinCC project and transfer it to the HMI device.
16. Create a machine-specific logic for the counting of the workpieces by dynamically changing the parameter of the **LTLL_Counter** block:

countValue:	Total number of the parts to be counted per Pulse (OK and NOK parts)
countValueBad:	Number of bad parts to be counted
typIndex:	Index of the workpiece type to be counted; if only total counter, then "0"
count:	Count pulse (rising edge 0 > 1)

17. Create, if required or necessary, a machine-specific logic for resetting the unit counter. The HMI LITE screen provides the possibility for the manual reset of the workpiece counter.

Diagnostics

9

9.1 Messages and message buffers

9.1.1 Layout and functionality

The screen structure of the two **Messages** and **Message buffer** screens is identical. All messages are displayed in tables in the screens. The currently active messages are displayed in the **Messages** screen. The **Message buffer** screen displays the contents of the message buffer.

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

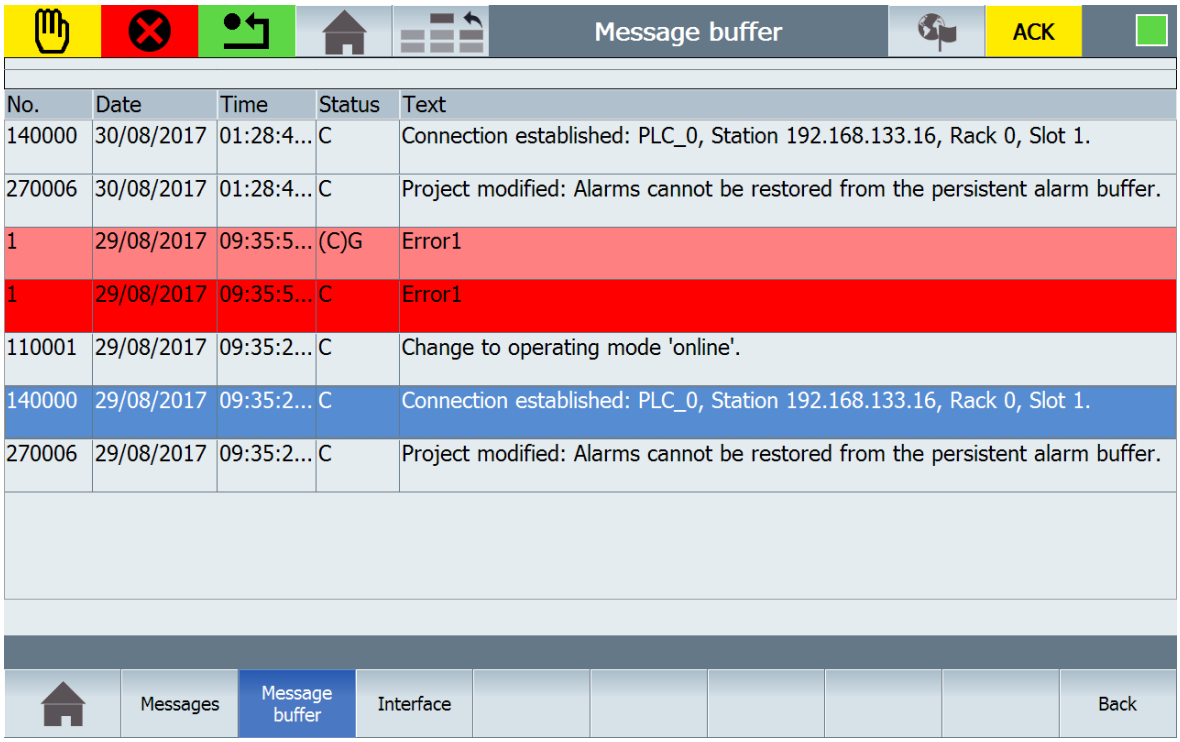


Fig. 9-1: Message buffer (SS_03_002_AlarmHistory)

The following information is displayed in a table:

- Message number
- Time stamp of the message
- Message status (K: incoming, G: outgoing, Q: acknowledged)
- Message text

9.1.2 Runtime interface

Preconfigured bit messages exist in HMI Lite.

The data block interface for these messages is defined by the WinCC tags **SO_00_000_fault** for faults and **SO_00_000_warnings** for warnings.

9.1.3 Configuration

The message texts are configured under **HMI messages > Bit messages**. Additional information in this regard may be found in the WinCC documentation.

Integrating the PLC code display

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

1. To do so, move the screen **SS_10_013_PlcCodeViewer** from the project library to the project tree.
   - 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:

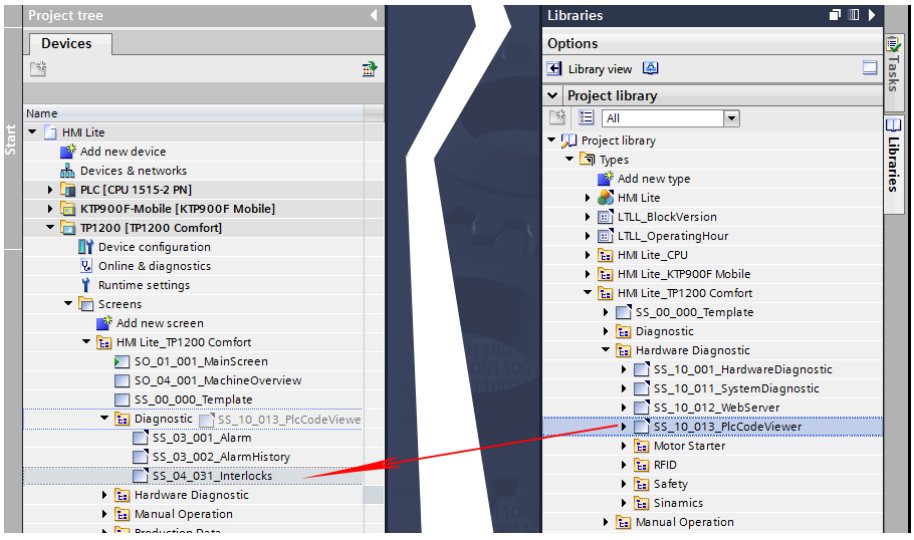


Fig. 9-2: Integrating the PLC code display

2. Open the **SS_03_001_Alarm** screen and insert the following function in the properties of the **PLC-Code Viewer** button in the **Events > Click** tab: **ActivatePLCCodeViewer** with the parameters:
   - Screen name: SS_10_013_PlcCodeViewer
   - Screen object: plcCodeViewer

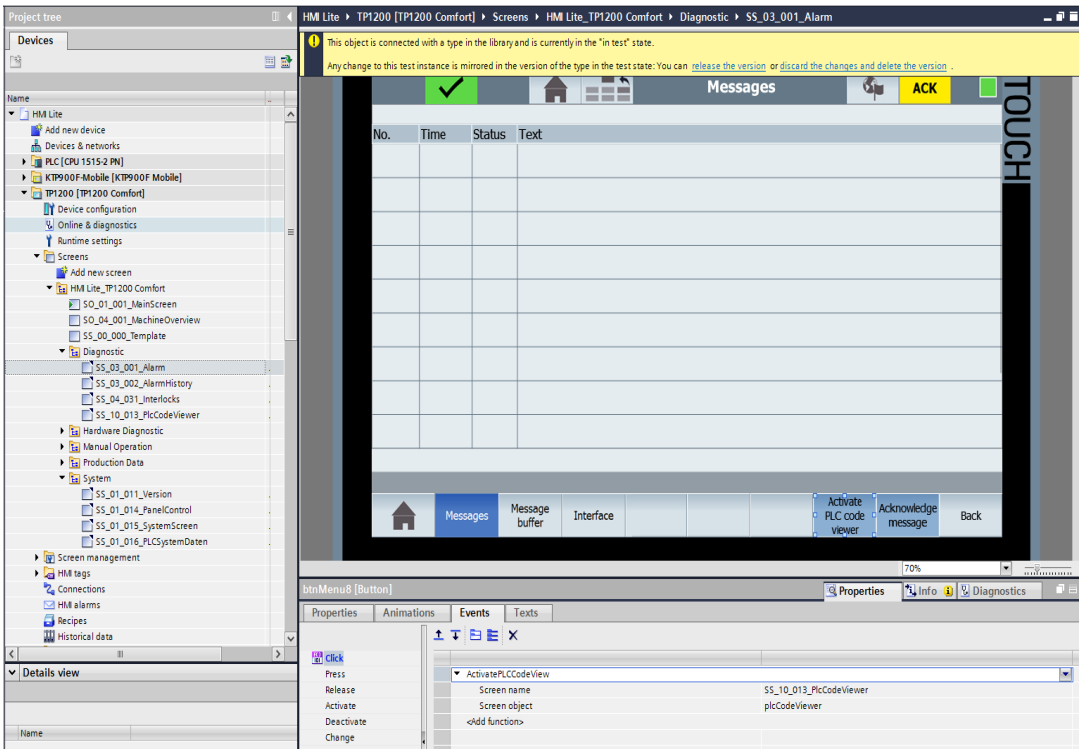


Fig. 9-3: Activating PLCCodeViewer

Settings of the message buffer

The **Message buffer** displays selected message events from the message buffer. The configuration specifies which events are displayed, meaning that the message window displays the message events selected in the properties.

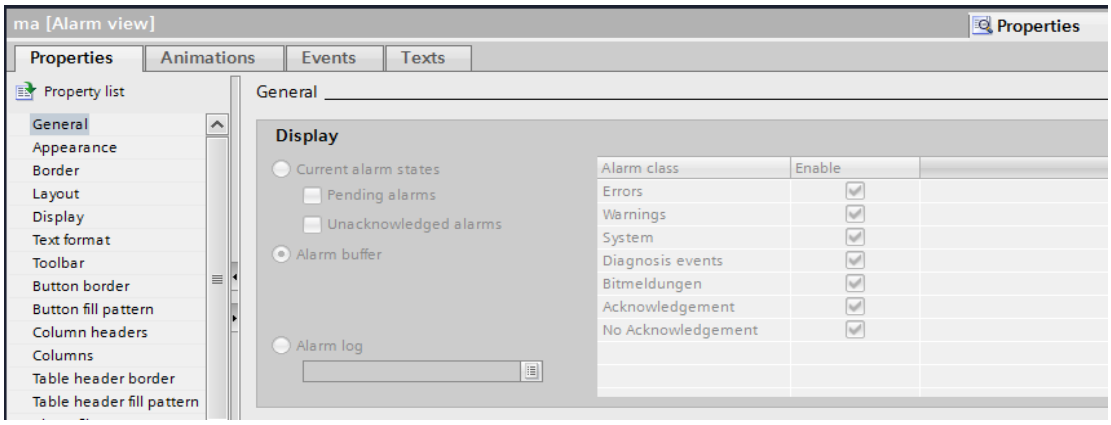
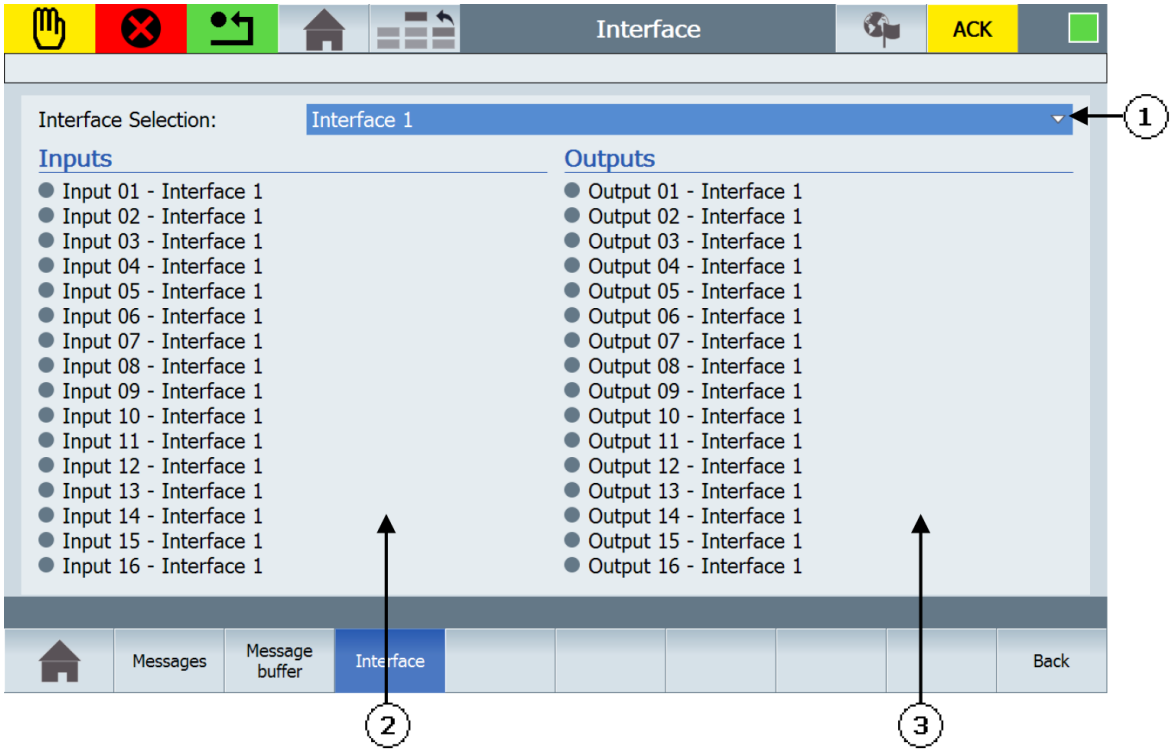


Fig. 9-4: Setting for the message display object in the Message buffer screen

9.2 Interface

9.2.1 Layout and functionality

The **Interface** screen can be used to diagnose the interface signals between the control and 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 desired interface can be selected from a drop-down list.



(1) Selection list for selecting the diagnostic interface

(2) Status display of the inputs

(3) Status display of the outputs

Fig. 9-5: Interface (SS_04_031_Interlocks)

9.2.2 Runtime interface

The runtime interface for the **Interface** screen consists of three tags. The **LTLL_Data.HMI[X].screenInterlock.selection** tag represents the current interface that was selected from the drop-down list.

Address	LTLL_Data.HMI[X].screenInterlock.selection
Format	INT
Value range	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
Value range	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
Value range	The status of each bit is displayed in the screen by the associated LED element.
Default setting	-

9.2.3 Configuration

Up to 218 interface descriptions can be defined and selected in the drop-down list. A name can be configured for each of these interfaces in a WinCC text list.

Table 9-1: Selection window for the interlocks – screen caption of the text list

Text list		**SO_04_031_InterlockSelection**
Display Format		Text Decimal
Value	01	Name for Interface No. 1
Value	02	Name for Interface No. 2
		…
Value	218	Name for Interface No. 218

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

Table 9-2: Designation of the input/outputs

Text list		**SO_04_031_InterlocksInputs** **SO_04_031_InterlocksOutputs**
Display Format		Text Decimal
Value	01	Name for input/output #1 of interface #1
Value	02	Name for input/output #2 of interface #1
		…
Value	16	Name for input/output #16 of interface #1
Value	17	Name for input/output #1 of interface #2

Hardware diagnostics

10

From the **Hardware diagnostics** screen you can branch into the individual diagnostics screens. Depending on their scope, these are, in turn, divided into their own substructures.

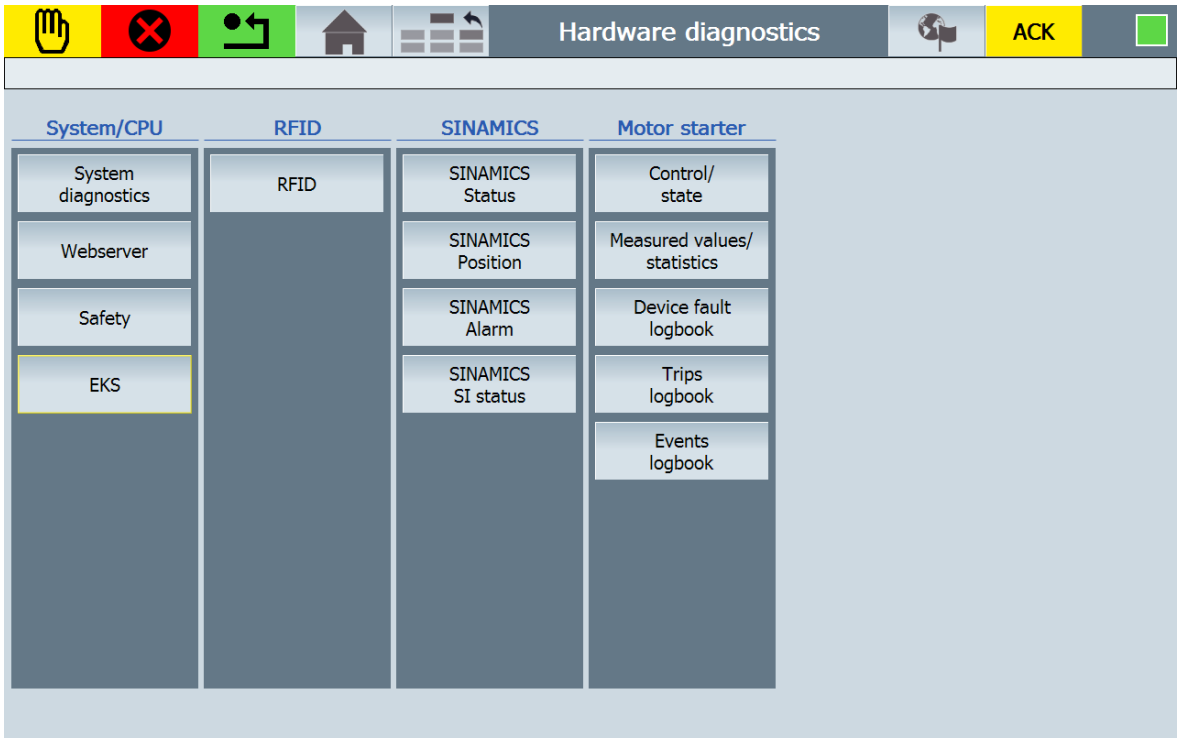


Fig. 10-1: Hardware diagnostics (SO_10_001_HardwareDiagnostic)

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

10.1 System diagnostics

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

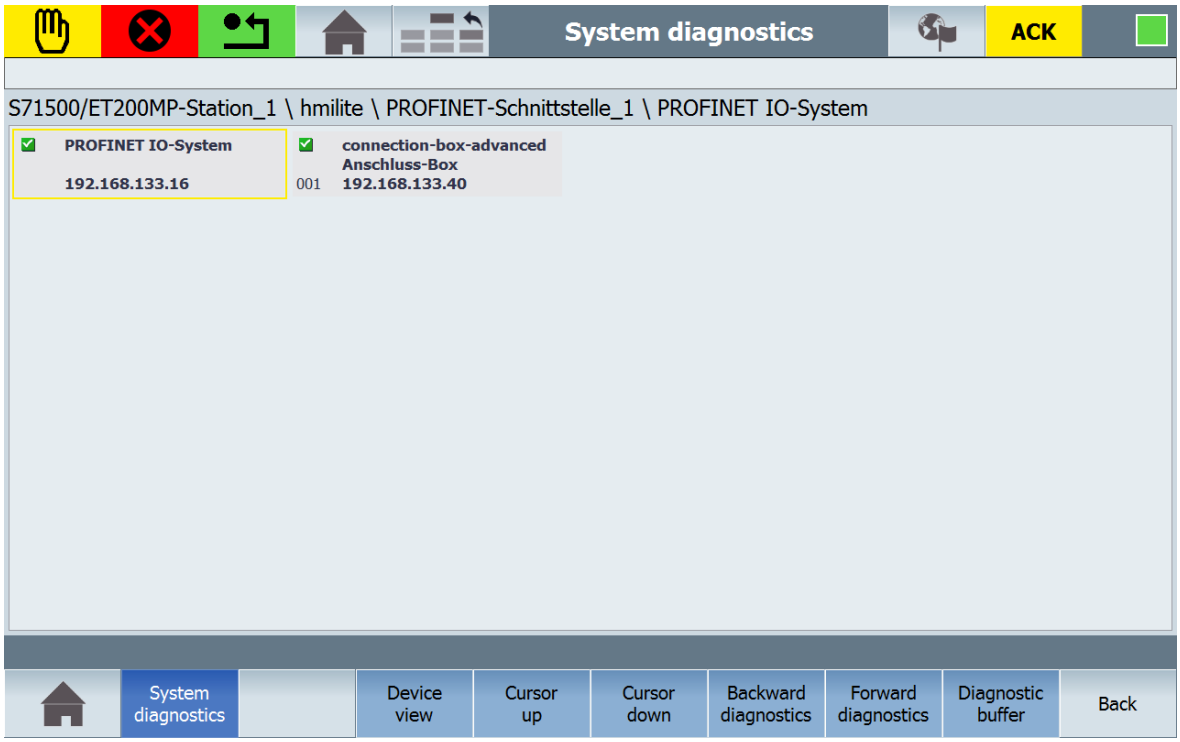


Fig. 10-2: System diagnostics (SS_10_011_SystemDiagnostic)

10.2 Web server

An HTML browser object is integrated in the HMI Lite screen **Web server**.

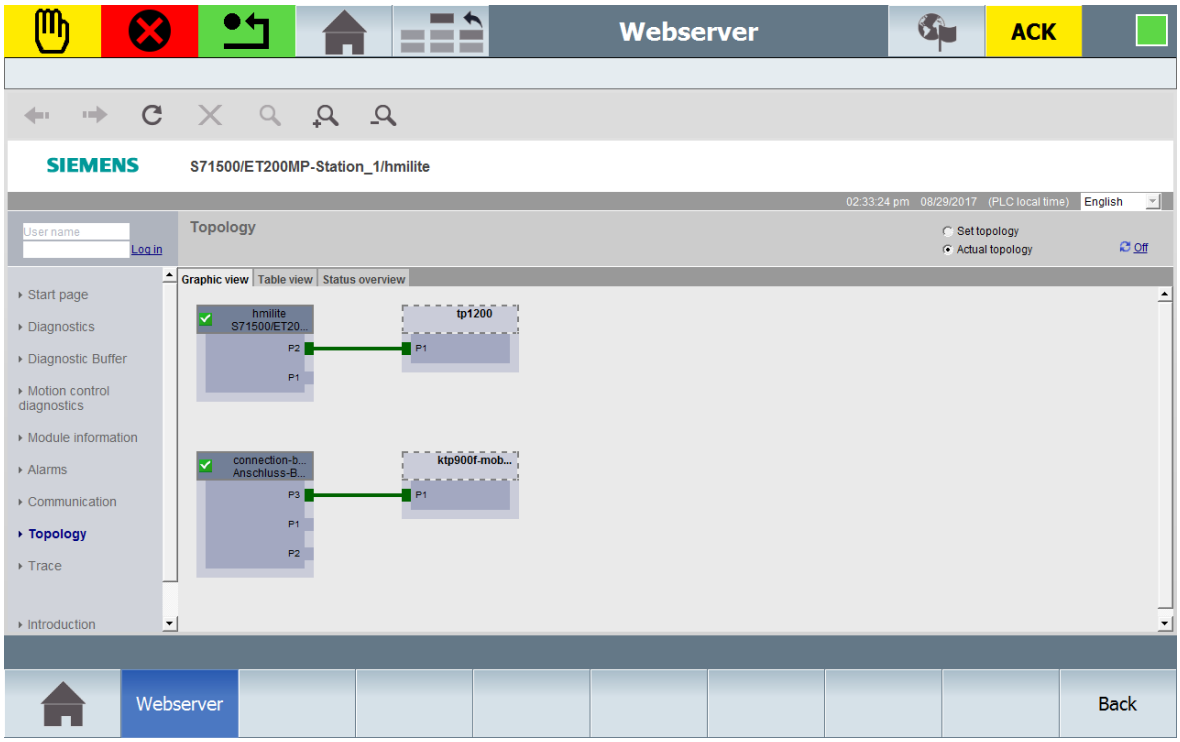


Fig. 10-3: Web server (SS_10_012_WebServer)

The URL of the Web server is read out during the start by the **LTLL_Basic** block and is stored in the tag **LTLL_Data.HMI[X].global.readOnly.webServerAddress**.

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

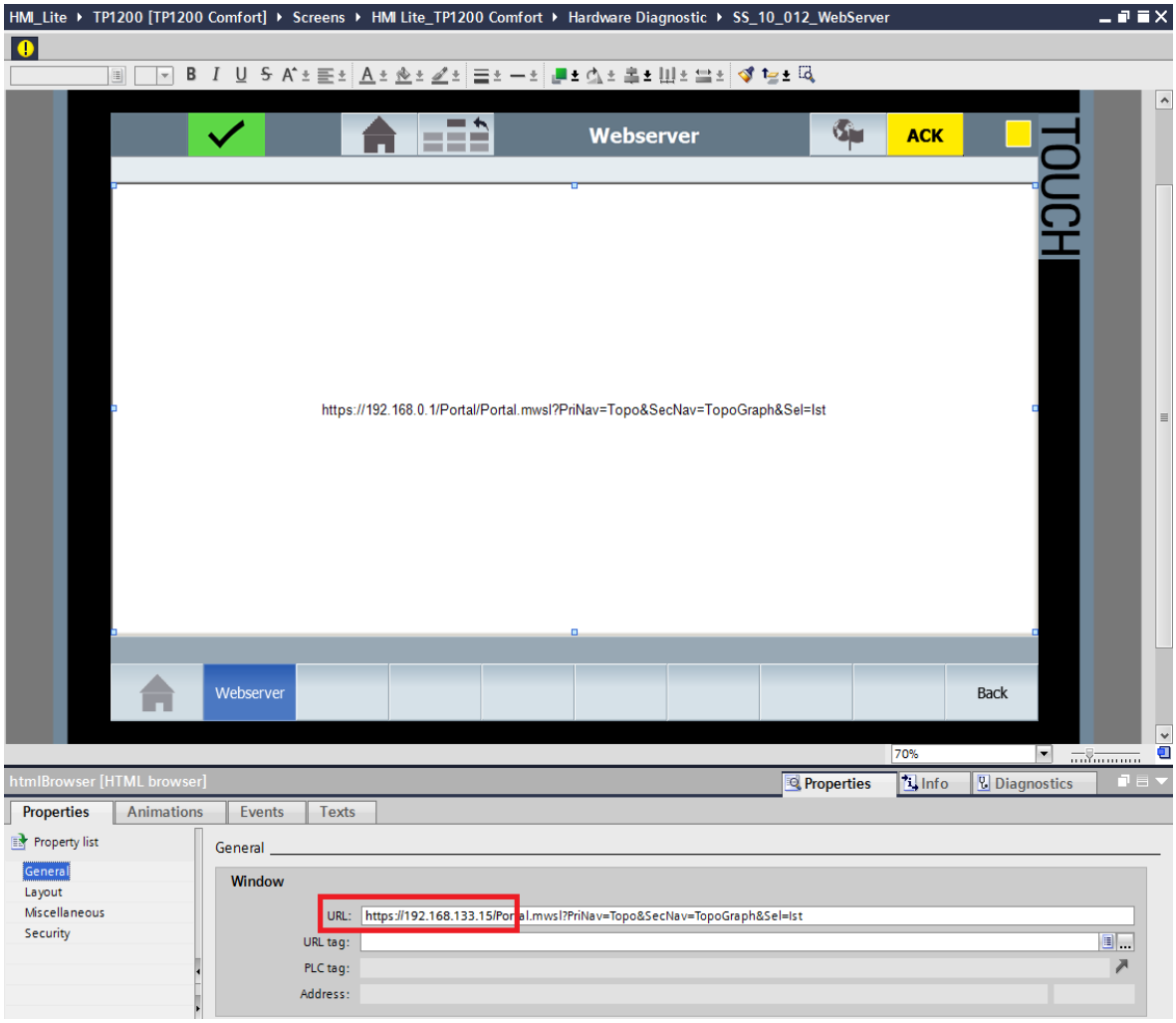


Fig. 10-4: Web server: Changing the URL

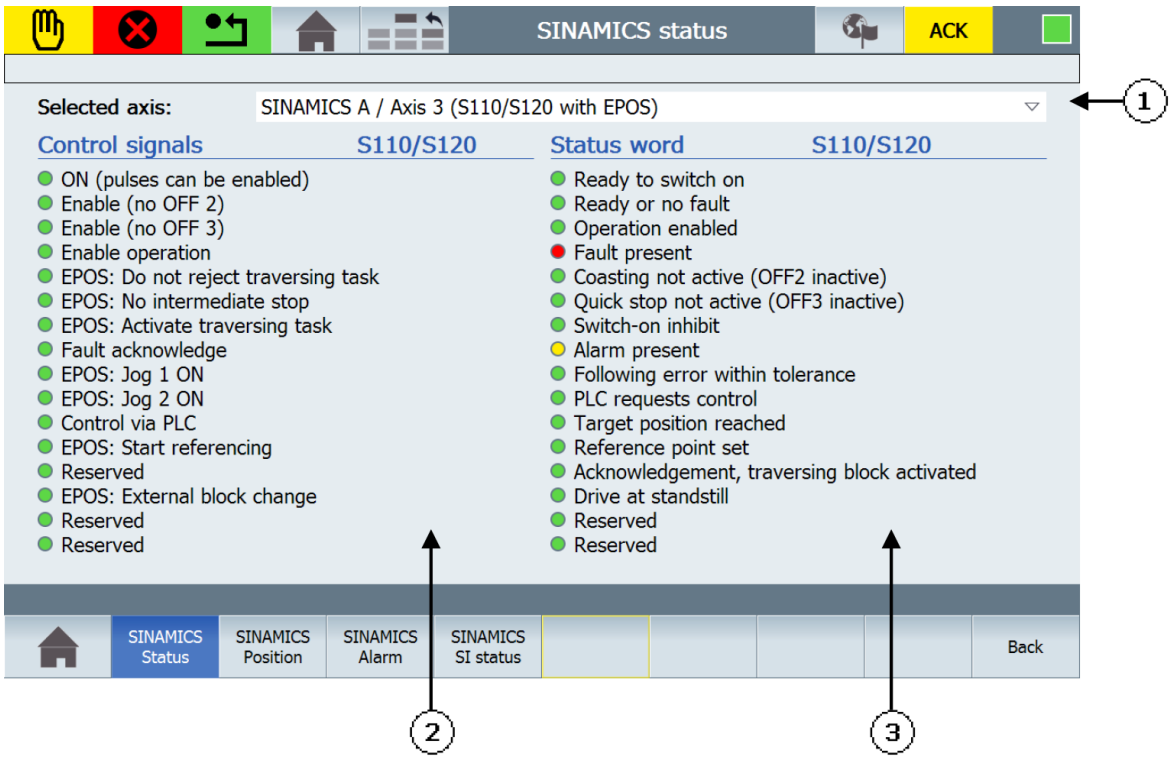
10.3 SINAMICS diagnostics

SINAMICS diagnostics is integrated in HMI Lite for the following drives and their variants:

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

10.3.1 SINAMICS Status

The **SINAMICS Status** screen displays the control and status signals of the SINAMICS axis that was selected from the drop-down list.



(1) Selection of the axis

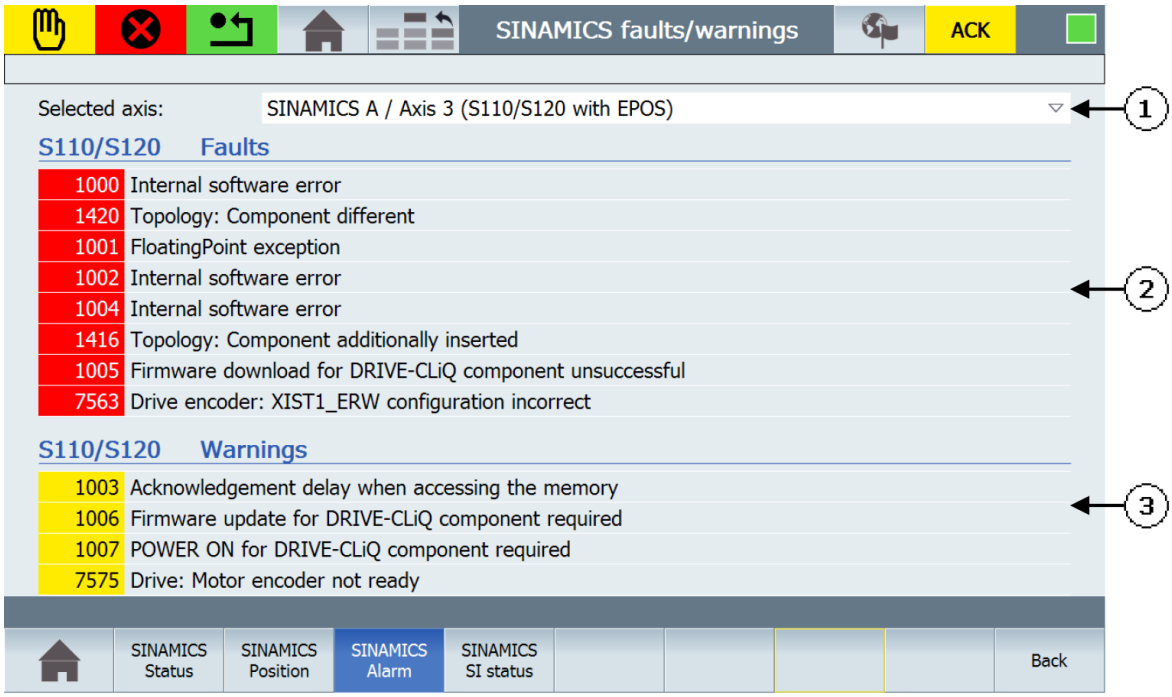
(2) Control signals of the selected axis

(3) Status signals of the selected axis

Fig. 10-5: SINAMICS status (SS_11_001_ControlStatusword)

10.3.2 SINAMICS Alarms

The **SINAMICS Alarms** screen displays the faults and warnings of the selected SINAMICS axis.



(1) Selection of the axis

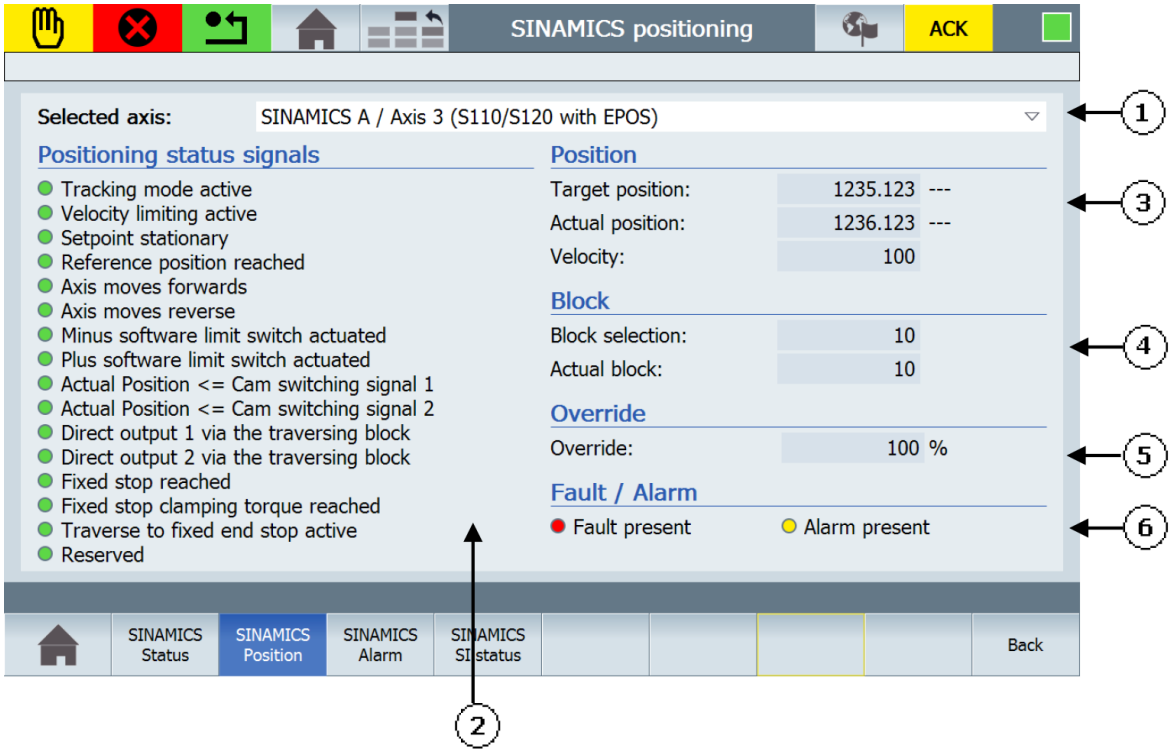
(2) Display of faults

(3) Display of warnings

Fig. 10-6: SINAMICS alarms (SS_11_021_FaultsAndWarnings)

10.3.3 SINAMICS Position

The **SINAMICS Position** screen displays the **Positioning status signals** and **Positioning data**, such as the Position, Block and Override of the selected axis. The data is only available for SINAMICS axes that are operated as positioning axes (EPOS).



(1) Selection of the axis

(2) Position of the status signals

(3) Display of the axis position

(4) Number of the selected block

(5) Override

(6) Display of any active fault/warning

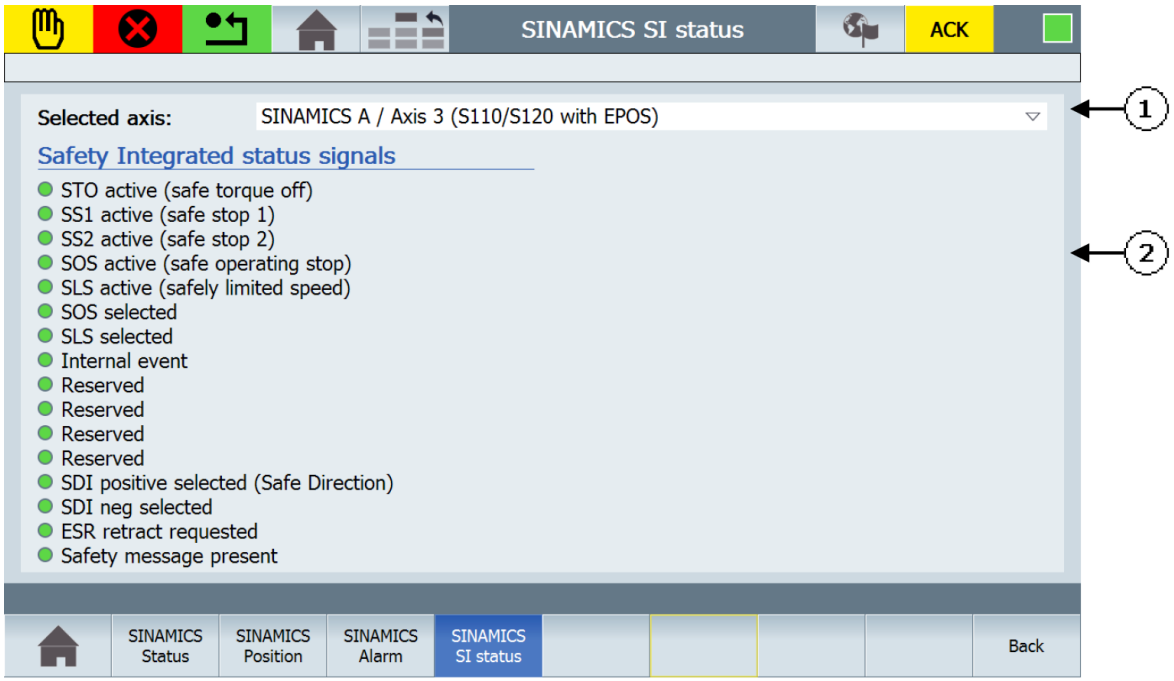
Fig. 10-7: SINAMICS position (SS_11_011_EPOSStatus)

Note

If SINAMICS S120 without positioning functionality as well as the SINAMICS G110/G120 are used, the information in the **SINAMICS Position** screen is not supplied.

10.3.4 SINAMICS SI Status

The **SINAMICS SI Status** screen displays the safety status signals of the SINAMICS axis that was selected from the drop-down list.



(1) Selection of the axis

(2) Safety Integrated status signals of the selected axis

Fig. 10-8: SINAMICS SI status (SS_11_031_SafetyStatusword)

10.3.5 Configuration of the WinCC screens

Configuring the text list in WinCC

For every configured designation in the text list a drive object has to be configured in the **LTLL_SinamicsCFG** data block.

The designation of the text list entry is completely free.

The value of the text list entry has to agree with the index of the drive object in the **LTLL_SinamicsCFG** data block. A text list entry is assigned to a drive object through the value.

The **SO_11_000_SinamicsAxis** text list has the following structure:

Table 10-1: Text list for the axis designations

Text list		**SO_11_000_SinamicsAxis**
Display Format		Text Decimal
Value	0	Designation of the first axis (value = Drive object index in the **LTLL_SinamicsCFG**)
Value	1	Designation of the second axis (value = Drive object index in the **LTLL_SinamicsCFG**)
…	…	…

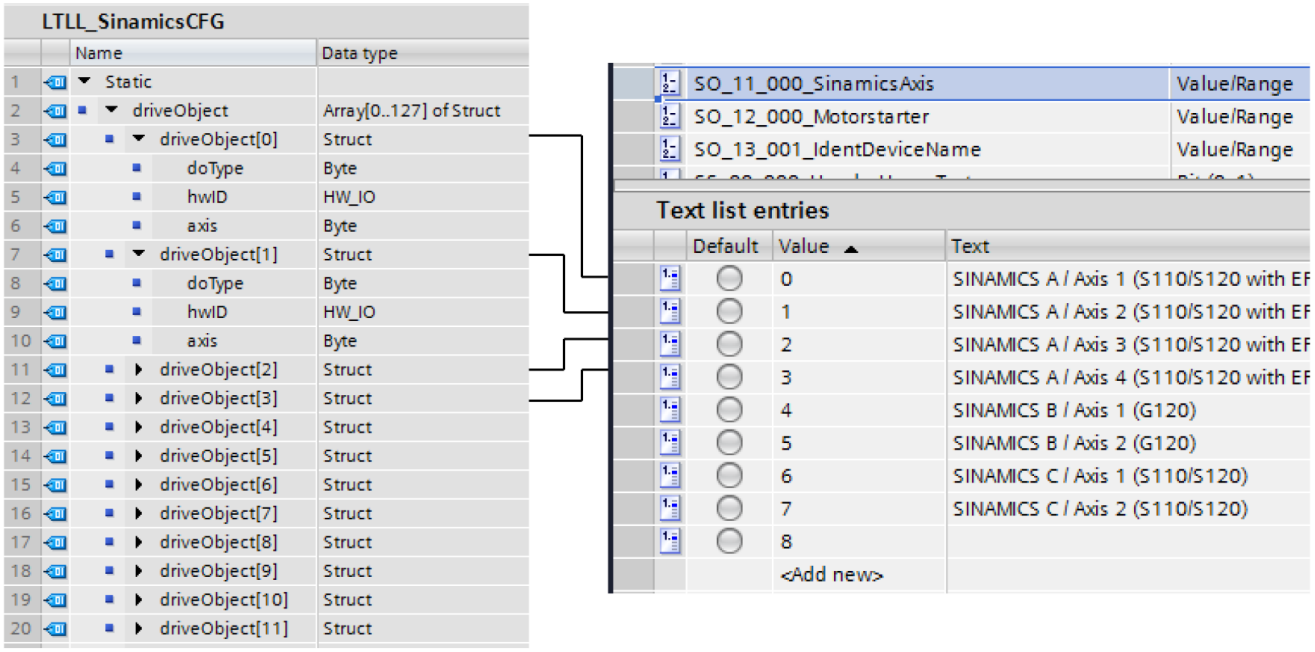


Fig. 10-9: Assignment of the text list entry to the drive object

NOTICE
Designation text items for missing axes must be deleted! The value of the text list entry has to agree with the index of the drive object that was configured in the **LTLL_SinamicsCFG**.

10.3.6 Configuration of a drive object (LTLL_SinamicsCFG)

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

Table 10-2: Structure of a drive object in the LTLL_SinamicsCFG

Name	Type	Description
doType	BYTE	Drive object type. 0 = SINAMICS S110/S120 with positioning functionality (EPOS) 1 = SINAMICS S110/S120 without positioning functionality (EPOS) 2 = SINAMICS G110/G120
hwID	HW_IO	Hardware identification of the DP slave, taken from "Devices & networks".
axis	BYTE	Drive object ID

10.3.7 Runtime interface (LTLL_Sinamics)

The **LTLL_Sinamics** block supplies the WinCC screens for the SINAMICS diagnostics screens. The displayed data is read directly from the drive by parameter jobs via acyclic communication services.

The function block has to be called once cyclically. The FB call has to be enabled via the **driveEnable** parameter.

Call interface

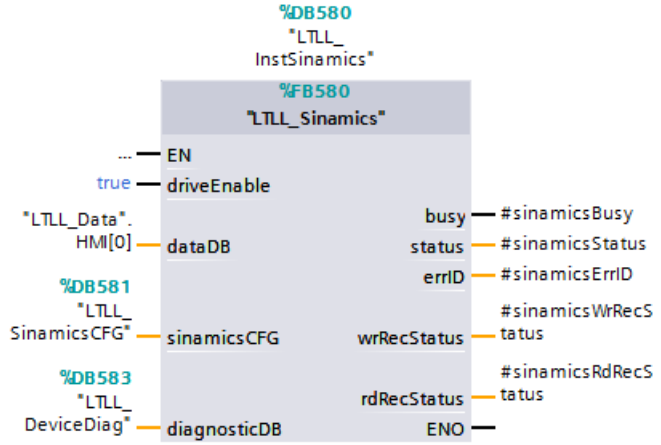


Fig. 10-10: Call interface for block LTLL_Sinamics

Parameters

Table 10-3: Parameters of function LTLL_Sinamics

Name	Declaration	Type	Standard	Description
driveEnable	Input	BOOL	TRUE	"TRUE" enables the communication of the block with the drive.
dataDB	InOut	LTLL_typeData	LTLL_Data.HMI [0]	HMI Lite Runtime data DB
diagnosticDB	InOut	LTLL_typeDevice Diag	LTLL_DeviceDiag	HMI Lite diagnostics data block
sinamicsCFG	InOut	LTLL_typeSinamicsCFG	LTLL_Sinamics CFG	HMI Lite block in which the drive objects are configured
busy	Output	BOOL	TRUE	"TRUE" communication with the drive
status	Output	WORD	---	Block status
errId	Output	WORD	-	Local error handling
wrRecStatus	Output	DWord	-	Status of the WRREC instruction
rdRecStatus	Output	DWord	-	Status of the RDREC instruction

Output parameter status

Table 10-4: Description of the output parameter status of LTLL_Sinamics

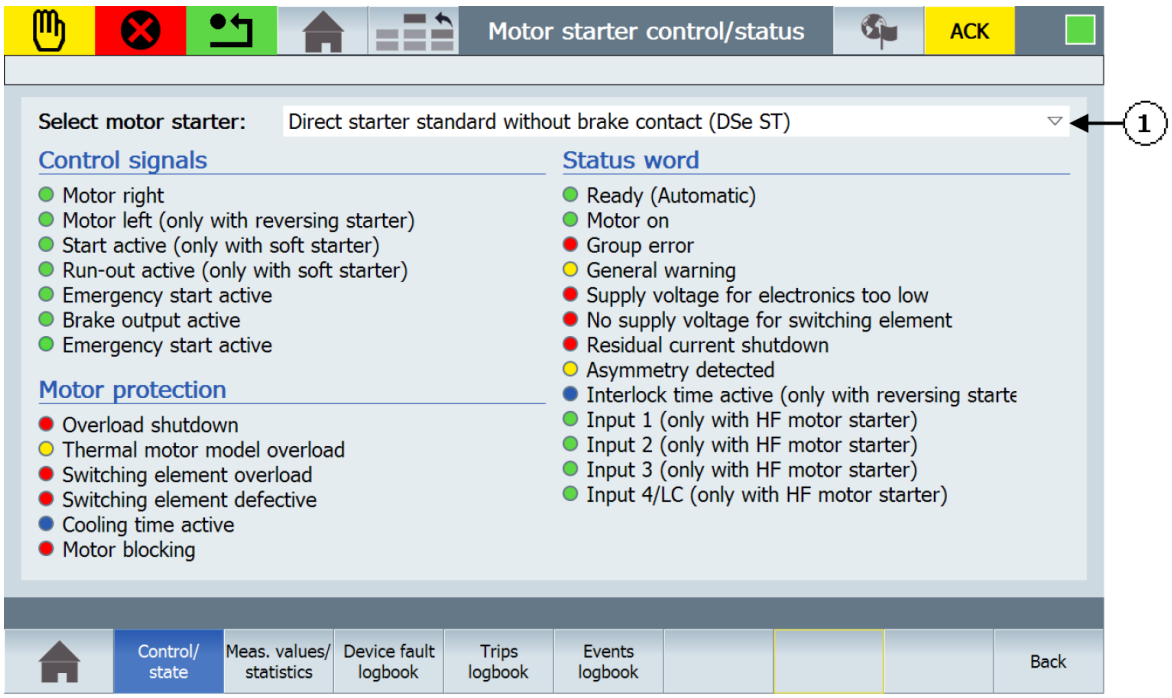
Error code (W#16# ….)	Description
16#8200	HMI Lite licensing failed
16#8201	Invalid drive object
16#8600	Error in instruction WRREC
16#8601	Error in instruction RDREC
16#8602	Invalid job reference

10.4 Motor starter control/status

The motor starter diagnostics consists of the following diagnostics screens:

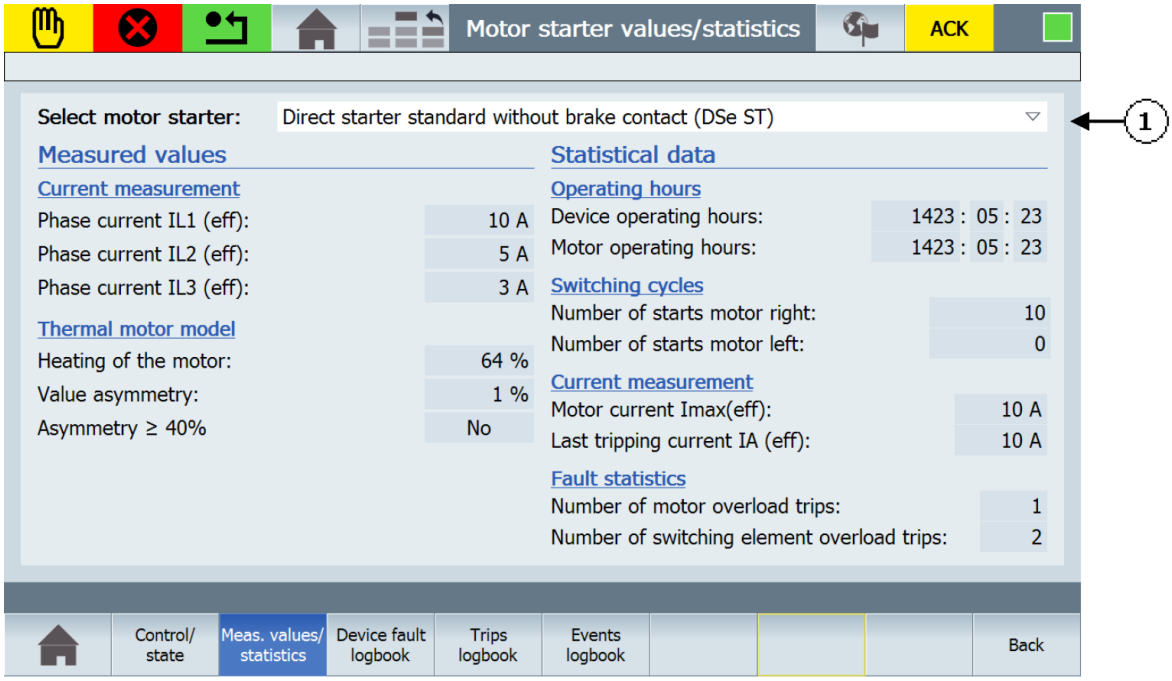
- Control/status
  Control signals, motor protection, status signals
- Measured values/statistics
  Measured values, statistical data
- Log book - Device errors
- Log book - Triggering operations
- Log book - Events

10.4.1 Layout and functionality



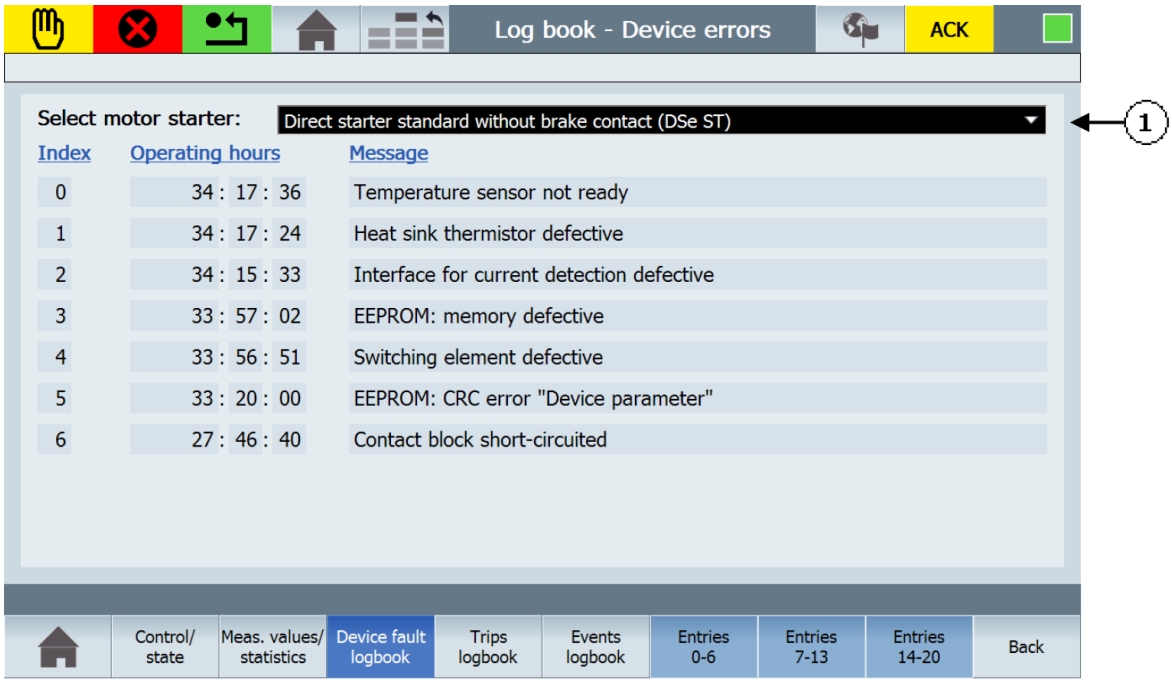
(1) Selecting the motor starter

Fig. 10-11: Motor starter control/status (SS_12_001_ControlStatus): Control signals, motor protection, status signals



(1) Selecting the motor starter

Fig. 10-12: Motor starter measured val./statistics (SS_12_011_DataStatistics): Measured values, statistical data



(1) Selecting the motor starter

Fig. 10-13: Logbook device error (SS_12_021_LogbookDeviceError)

10.4.2 Runtime interface (LTLL_Motorstarter)

The **LTLL_Motorstarter** block supplies the WinCC screens for the motor starter diagnostics. You have to call this block once cyclically.

Call interface

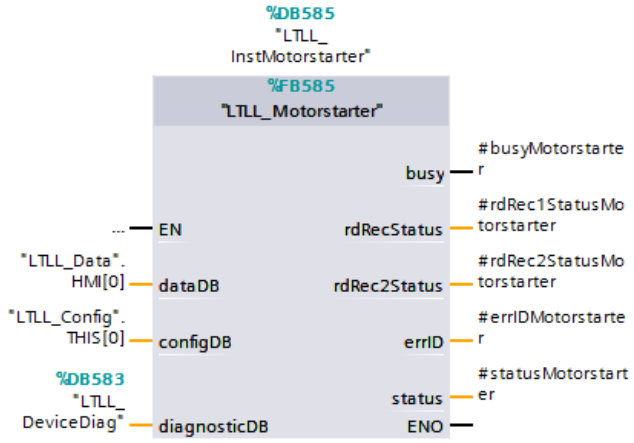


Fig. 10-14: Call interface for block LTLL_Motorstarter

Parameters

Table 10-5: Parameters of block LTLL_Motorstarter

Name	Declaration	Type	Standard	Description
dataDB	InOut	LTLL_typeData	LTLL_Data.HMI[0]	HMI Lite Runtime data DB
configDB	InOut	LTLL_typeConfig	LTLL_Config.THIS[0]	HMI Lite Configuration DB
diagnosticDB	InOut	LTLL_typeDeviceDiag	LTLL_DeviceDiag	Number of the HMI diagnostic data block
busy	Output	BOOL	-	Job running
rdRecStatus	Output	DWord	---	Status of RDREC1
rdRec2Status	Output	DWord	-	Status of RDREC2
status	Output	WORD	---	Block status
errId	Output	WORD	-	Local error handling

Output parameter status

Table 10-6: Description of the output parameter status of LTLL_Motorstarter

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

Configuring in LTLL_Config

You have to specify the hardware address of the motor starter that you want to diagnose in the **LTLL_Config** block. Up to 128 motor starters can be entered. The index of the field corresponds to the index of the text list in WinCC.

Address	LTLL_Config.THIS[X].screenMotorstarter.hwID[Y] (Y corresponds to the number of the motor starter selected in the screen)
Format	HW_IO
Value range	-
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**. Each motor starter to be diagnosed must have an entry in the text list.

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

Table 10-7: Text list for the designations of the motor starter

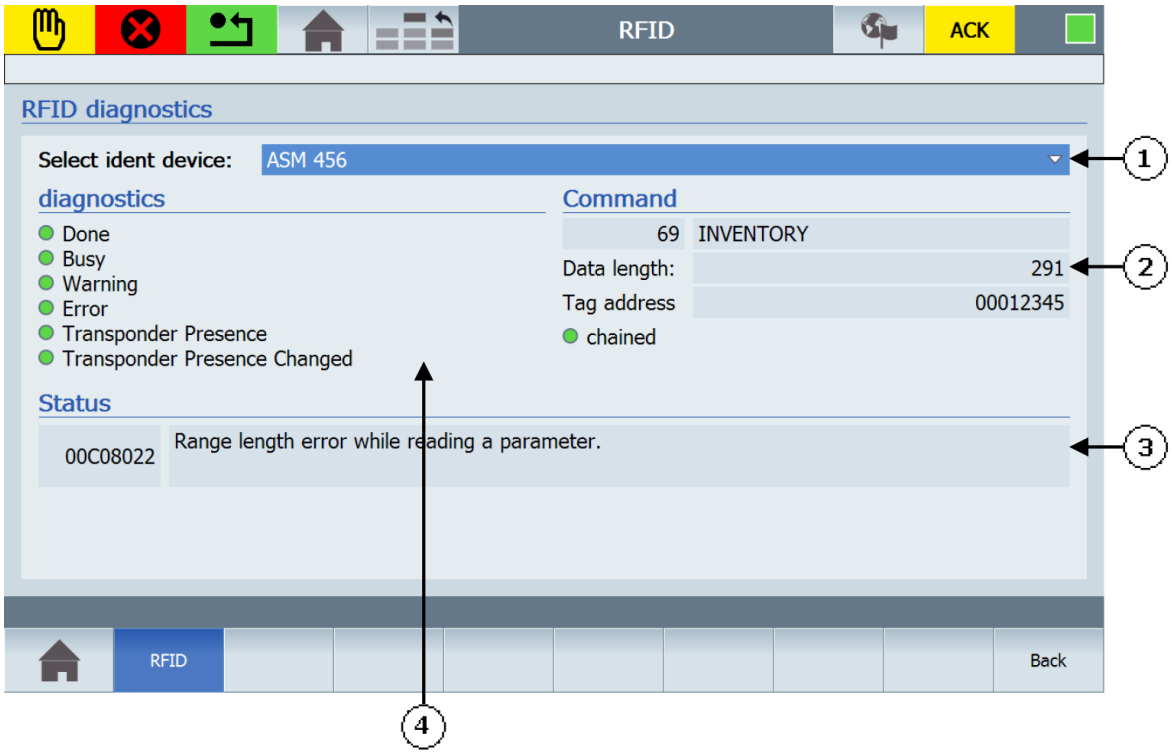
Text list		SO_12_000_Motorstarter
Display Format		Text Decimal
Value	0	Designation of the first motor starter
Value	1	Designation of the second motor starter
…	…	…

10.5 RFID

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

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

10.5.1 Layout and functionality



(1) Selecting the identification device

(2) Command area

(4) Error messages of the identification device

(4) Diagnostics bits

Fig. 10-15: RFID (SS_13_001_RFID)

Selection of the ident device

Select an ident device from the drop drop-down list. Each ident device represents a separate interface.

Diagnostics

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

Command

The data of the current command is displayed in this area.

Status

Error messages of the selected ident device are displayed in this area.

10.5.2 Supported identification devices

The following Ident devices are supported:

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

Data exchange between the control and ident devices is effected either through the ident blocks or through the ident profile.

10.5.3 Configuration of the WinCC screen

Configuring the text list in WinCC

The designations of the ident devices have to be configured. The text items are stored in the WinCC text list **SO_13_001_IdentDeviceName**. Each configured ident device must have an entry in the text list.

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

Table 10-8: Text list for the designations of the identification 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
…	…	…

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

10.5.4 Runtime interface (LTLL_RFID)

The **LTLL_RFID** block supplies the WinCC screens for the RFID diagnostics. The displayed data is read in via the interface.

You have to call this block cyclically once for each configured ident device, whereby the **selectedDevice** parameter corresponds to the corresponding values from the WinCC text list **SO_13_001_IdentDeviceName**.

Call interface

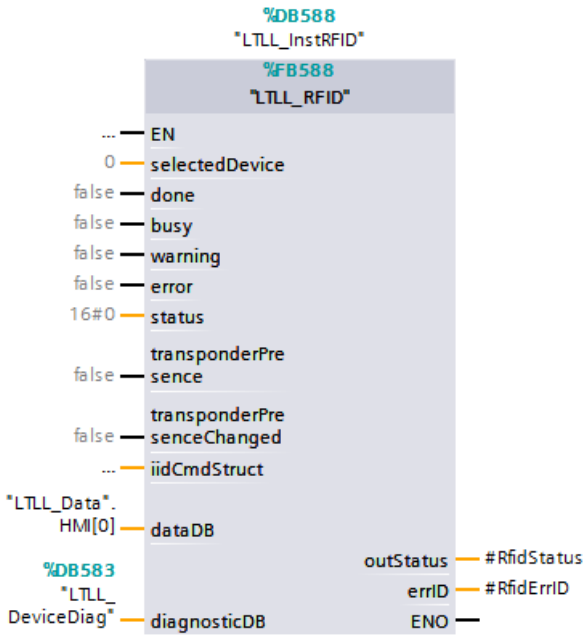


Fig. 10-16: Call interface of block LTLL_RFID

Parameters

Table 10-9: Parameters of block LTLL_RFID

Name	Declaration	Type	Standard	Description
selectedDevice	Input	UInt	---	Selecting the ident device: Value from the text list SO_13_001_IdentDeviceName
done	Input	BOOL	---	Output parameter **done** of the ident blocks or of the ident profile
busy	Input	BOOL	---	Output parameter **busy** of the ident blocks or of the ident profile
warning	Input	BOOL	---	Output parameter **warning** of the ident blocks or of the ident profile
error	Input	BOOL	---	Output parameter **error** of the ident blocks or of the ident profile
status	Input	DWord	---	Output parameter **status** of the ident blocks or of the ident profile
Transponder Presence	Input	BOOL	---	Output parameter **presence** of the ident blocks or of the ident profile
Transponder Presence Changed	Input	BOOL	---	Output parameter **tpc** of the ident blocks or of the ident profile
iidCmdStruct	Input	IID_CMD_ STRUCT	---	Input parameter of the current command of the ident blocks or of the ident profile
dataDB	InOut	LTLL_typeData	LTLL_Data. HMI[0]	HMI Lite runtime data DB
diagnosticDB	InOut	LTLL_typeDeviceDiag	LTLL_ DeviceDiag	Number of the HMI diagnostic data block
outStatus	Output	WORD	---	Block status
errId	Output	WORD	---	Local error handling

Output parameter outStatus

Table 10-10: Description of the output parameter outStatus of LTLL_RFID

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

10.6 Safety

The upper area of the screen displays the generation time and collective signature of the current generation approved by the safety program.

This allows deviations of the current collective signature from the approved one to be determined in 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.

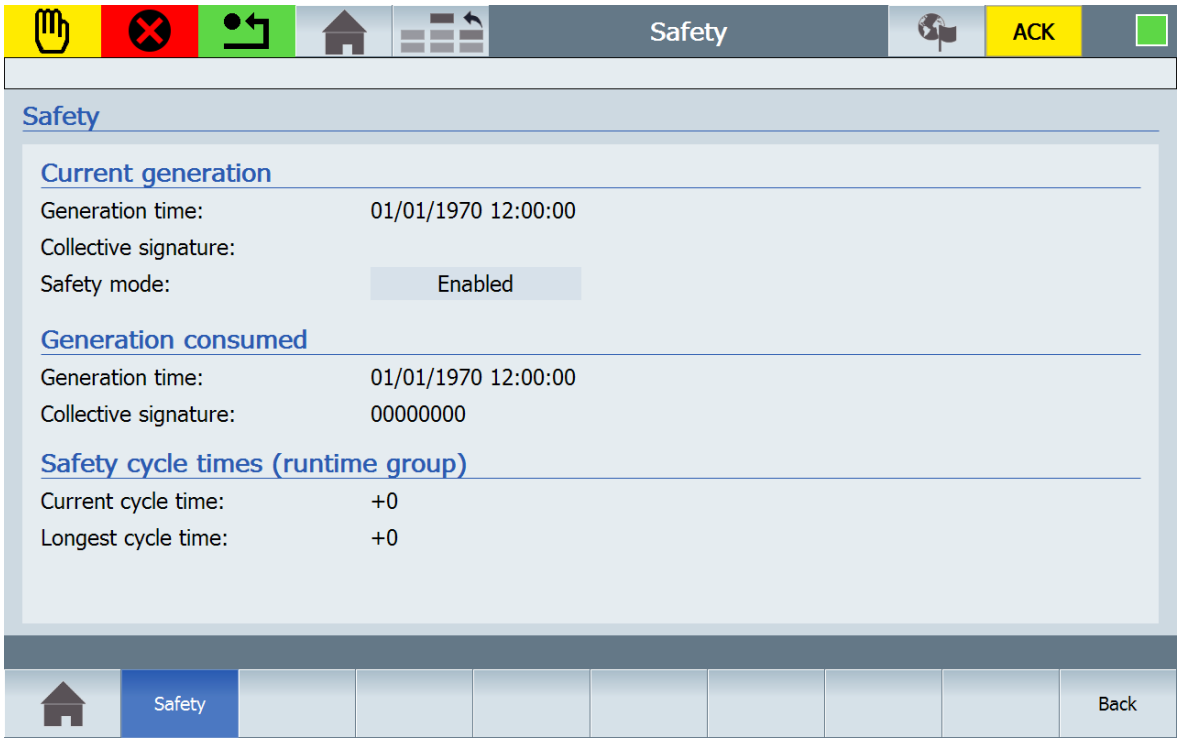


Fig. 10-17: Safety (SS_14_001_Safety)

Configuration

NOTICE
If you are not using an F-PLC or a safety program, you have to delete the call of the LTLL_Safety block and the block itself from the project. The block may be retained in the project library.

At the **fSysInfo** input parameter, transfer the **F_SYSINFO** parameter of the F-runtime group info data block (**Program blocks > System blocks > STEP 7 safety**).

You can set the current safety data as acceptance data via the **set** input parameter.

Call interface

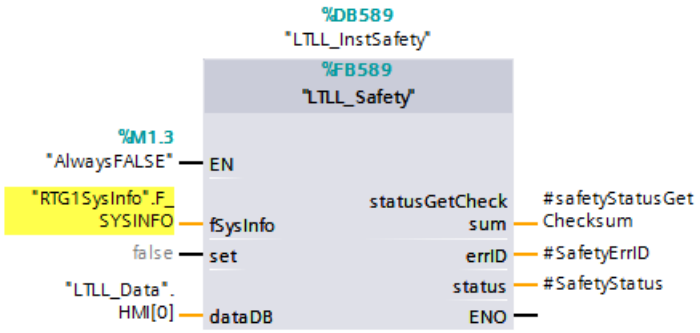


Fig. 10-18: Call interface for block LTLL_Safety

Parameters

Table 10-11: Parameters of block LTLL_Safety

Name	Declaration	Type	Standard	Description
fSysInfo	Input	F_SYSINFO	RTG1SysInfo.F_SYSINFO	S_SYSINFO parameter of the F-runtime group info DB
set	Input	BOOL	---	Positive edge copies the current data to the acceptance data
dataDB	InOut	LTLL_typeData	LTLL_Data.HMI[0]	HMI Lite Runtime data DB
statusGetChecksum	Output	WORD	---	Status of the GetChecksum instruction
status	Output	WORD	---	16#8200 HMI Lite licensing failed
errId	Output	WORD	---	Local error handling

Output parameter status

Table 10-12: Description of the output parameter status of LTLL_Safety

Error code (W#16# ….)	Description
16#8200	HMI Lite licensing failed

10.7 EKS

In this screen the relevant key data of an EKS (Electronic Key Systems) is displayed. The inserted key is read out by the **LTLL_EKS** function block.

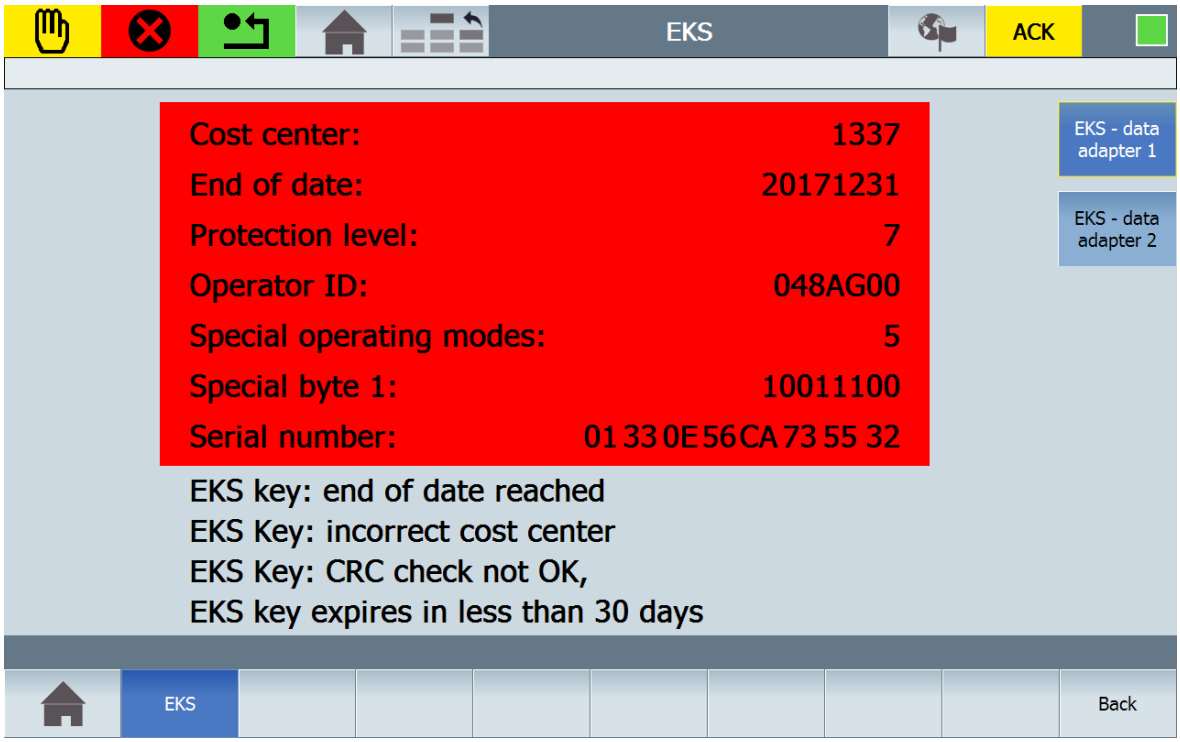


Fig. 10-19: EKS (SS_15_001_EuchnerKeySystem)

The screen displays the relevant key data. If no key is inserted, no data is displayed.

An unsuccessful check of the key data is indicated by a red marking, as well as by a text output of the cause.

If the EKS adapter (key holder) is not ready, this fact is pointed out in the screen.

If several EKS adapters are connected to the CPU, these can be selected by using the function keys on the right-hand side.

Note

If required, the 16-digit serial number of the plugged EKS key can be read out in the corresponding instance DB of the **LTLL_EKS**: Tags **KeyData.SerialNumber[1]** to **KeyData.SerialNumber[8]** in hexadecimal format.

10.7.1 Authorization levels concept

The following table shows the authorization levels concept when the EKS (Electronic Key Systems) is used:

Table 10-13: Authorization levels concept

Standard Siemens authorization levels	**Authorization**	**Euchner EKS**
Authorization level 1 (machine manufacturer)	Manufacturer, service, maintenance engineer	Authorization level 1 Red key Blue key (OEM)
Authorization level 2 (commissioning engineer, service)	Not used	Not used
Authorization level 3 (end user)	Not used	Not used
Authorization level 4 (programmer, machine setter)	Programmer, machine setter	Authorization level 4 Green key
Authorization level 5 (qualified operator)	Not used	Not used
Authorization level 6 (trained operator)	Operator	Authorization level 6 Black key
Authorization level 7 (semi-skilled operator)	No particular authorization	No key

10.7.2 Format of the EKS key

Use suitable software to read and write to the EKS key, e.g. Electronic Key Manager of the Euchner company.

Write to the EKS key, taking into account the following data structure:

- Memory size of the EKS key:
  124 bytes
- Area that can be written to:
  Bytes 0 - 115
- Area with serial number that cannot be changed:
  Bytes 116 - 123
- Function block LTLL_EKS described here verifies the checksum starting at byte 84.

Since the key allocation described here is also used for SINUMERIK-based machines with HMI PRO, some of the written data is configured for machine tools. The data is not relevant for use with HMI Lite.

In order to use an EKS key with this function block, it has to be written in the following format as of byte 84:

Table 10-14: Data of the EKS key

Data area of the EKS key	Size	Data format	Content	Further information
84-87	4 bytes	ASCII	Cost center	
88-95	8 bytes	ASCII	End date	End of the validity of the EKS key
96	1 byte	Hexadecimal	Authorization level	Protection levels 1-7
97-103	7 bytes	ASCII	Machine operator identification	
104	1 byte	Hexadecimal	Safe operating modes	MSO 1-5
105.0	1 bit	BOOL	Special bit, reworking	1=rework is enabled
105.1	1 bit	BOOL	Special bit, operating system	1=access to PC OS enabled
105.2	1 bit	BOOL	Special bit, quality data	0 = not OK 1 = OK
105.3-105.7	5 bits	BOOL		Reserve
106-113	8 bytes	Hexadecimal	Special bytes	Reserve

10.7.3 Configuration in WinCC

The corresponding **EKSAdapterHMIIndex** (input parameter of the FB **LTLL_EKS**) has to be entered in the WinCC tag **LTLL_EKS_HMI_DATA.EKSAdapterIndex**.

for the EKS adapter to be displayed.

10.7.4 Configuration in STEP 7 (LTLL_EKS function block)

The **LTLL_EKS** function block supplies the WinCC screen **SS_15_001_EuchnerKeySystem** to display the EKS key data.

The function block has to be called in the cyclic program once per EKS adapter (key recording) with different instance DBs (see example program).

The block includes the following functions:

Reading out the EKS key

Calculating the checksum of the key and comparing it with the checksum of the key

Checking the expiry date of the key

Checking the cost center of the key

NOTICE
The expiry date of the key is compared with the system clock of the S7 CPU. For this reason it must be ensured that the date and time of the S7-CPU are set correctly.

The cost center of the key is compared with the cost center parameterized at input parameters **CostCenter1** – **CostCenter5** .

If all checks have been carried out successfully, the protection level of the key is output at the **ProtectionLevel** output parameter.

The key data is deleted when the key is removed.

Configuring in the STEP7 hardware configuration

NOTICE
Only the module **Read/Write:128/120 Byte I/O** (DP identifier 192) is permitted for the EKS adapter with PROFIBUS interface. All other modules are not supported by the **LTLL_EKS** function block.

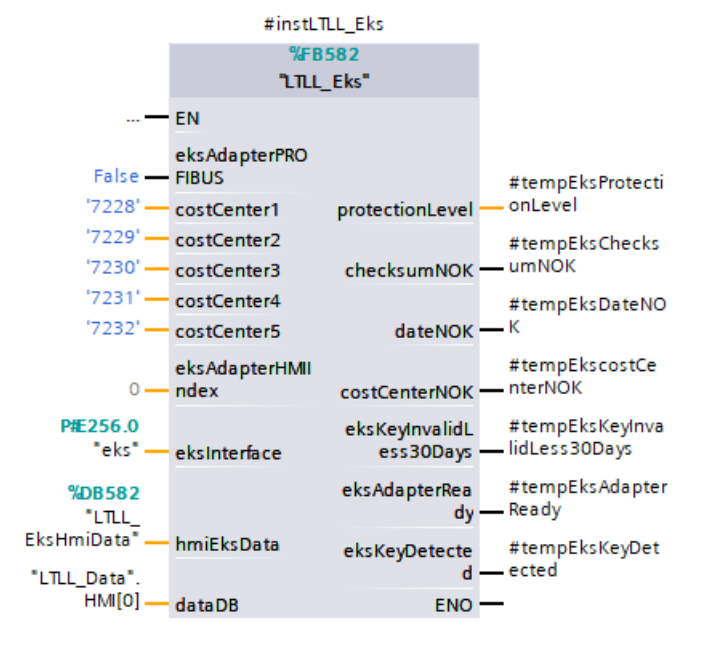


Fig. 10-20: Function block LTLL_Eks

Table 10-15: Parameter description LTLL_Eks

Name	Declaration	Type	Standard	Description
EKSAdapter PROFIBUS	Input	BOOL	-	Interface of the EKS adapter: PROFINET=FALSE PROFIBUS=TRUE
CostCenter1	Input	STRING[4]	-	Cost center 1
CostCenter2	Input	STRING[4]	-	Cost center 2
CostCenter3	Input	STRING[4]	-	Cost center 3
CostCenter4	Input	STRING[4]	-	Cost center 4
CostCenter5	Input	STRING[4]	-	Cost center 5
EKSAdapterHMI index	Input	UINT	-	HMI index for displaying the selected key data in the screen **SS_15_001_EuchnerKey System**
EKS_Interface	InOut	type_EKSInterface	-	Interface to the EKS adapter; PLC tag with data type type_EKSInterface
Hmi_Eks_Data	InOut	type_Hmi_Eks_ Data	-	DB LTLL_EKS_HMI_DATA for displaying the key data in the screen SS_15_001_EuchnerKey System
data_DB	InOut	LTLL_typeData	LTLL_ Data.HMI	HMI Lite Runtime data DB
ProtectionLevel	Output	INT	-	Output of the protection level
ChecksumNOK	Output	BOOL	-	Checksum of the key not OK
DateNOK	Output	BOOL	-	Expiry date of the key elapsed
CostCenterNOK	Output	BOOL	-	Cost center does not match parameterized cost center
EKSKeyInvalid Less30Days	Output	BOOL	-	EKS key will become invalid in less than 30 days
EKSAdapter Ready	Output	BOOL	-	EKS adapter ready to operate
EKSKeyDetected	Output	BOOL	-	EKS key recognized

System screens

11

11.1 Version

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

The **Software version** button is used to display a window with the versions of the WinCC Runtime system files.

This screen does not have to be configured.

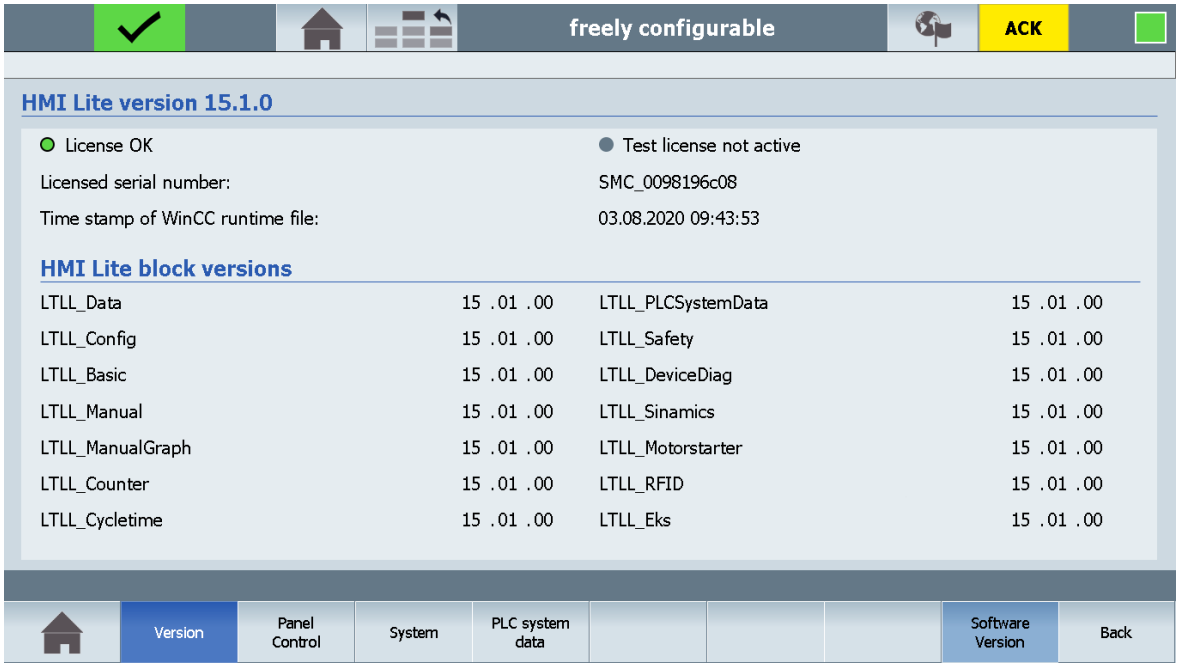


Fig. 11-1: Version (SS_01_011_Version)

11.2 Panel control

The **PanelControl** screen provides a number of functions associated with the maintenance and the setting of the HMI device. The range of functions is limited by the use of an IPC instead of a panel, which means that the fewer functions are available in the screen.

The **SS_01_014_PanelControl** with a limited range of functions is available in the standard project in the **HMI Lite_TP1200 Comfort _IPC477E 15 "MT > System (IPC replacements)** folder.

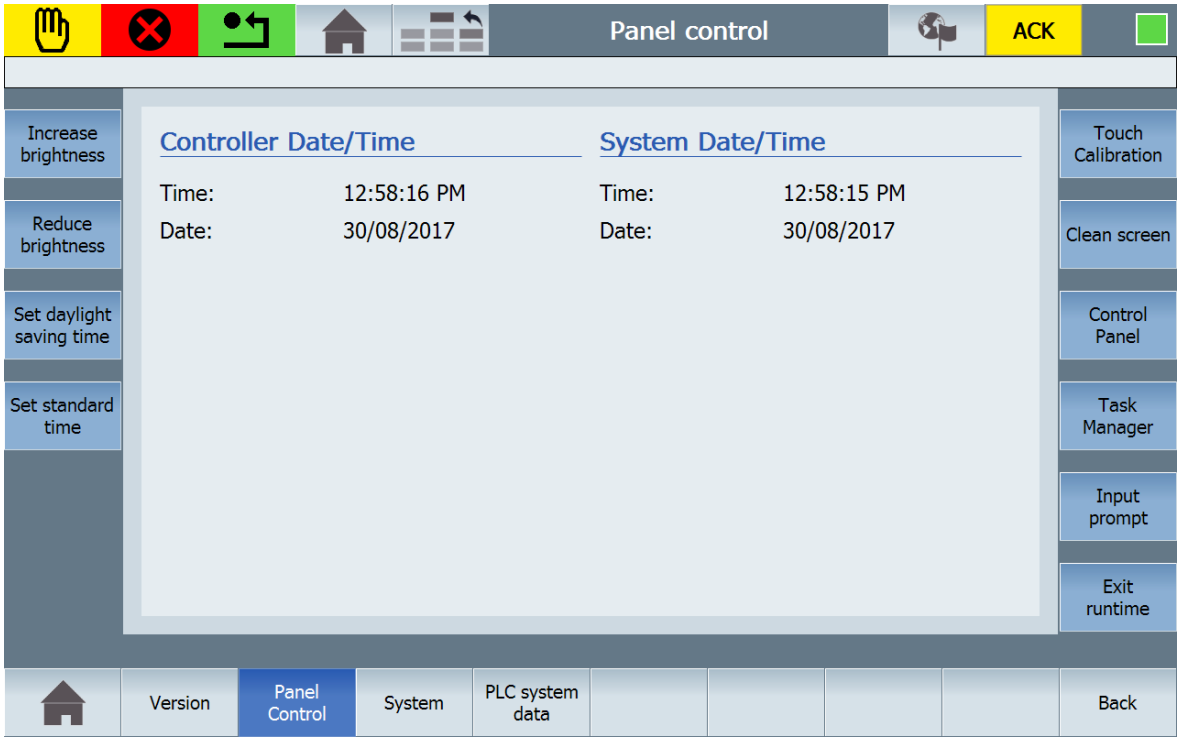


Fig. 11-2: Panel control (SS_01_014_PanelControl)

Increase/decrease brightness (available for panels only)

These softkeys are used to set the contrast of the HMI device.

Set daylight saving time (available for panels only)

When this softkey is pressed, the setting in the HMI device is changed to Daylight saving time.

Set standard time (available for panels only)

When this softkey is pressed, the setting in the HMI device is changed to Standard time.

Touch calibration (only available for panels with touch screen)

When the **Touch calibration** button is pressed the calibration of the touch screen is started.

Clean screen (only for touchscreen HMI devices)

After the **Clean screen** button has been pressed, the HMI device 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 (available for panels only)

When this button is pressed, the window for the control panel of the operating system opens.

Task Manager

When this button is pressed, the window for the Task Manager of the operating system opens.

Prompt to make an entry (available for panels only)

When this button is pressed, the window for the command prompt of the operating system opens.

Exit runtime

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

11.3 System

The **System** screen contains general system functions for the configuration of the system.

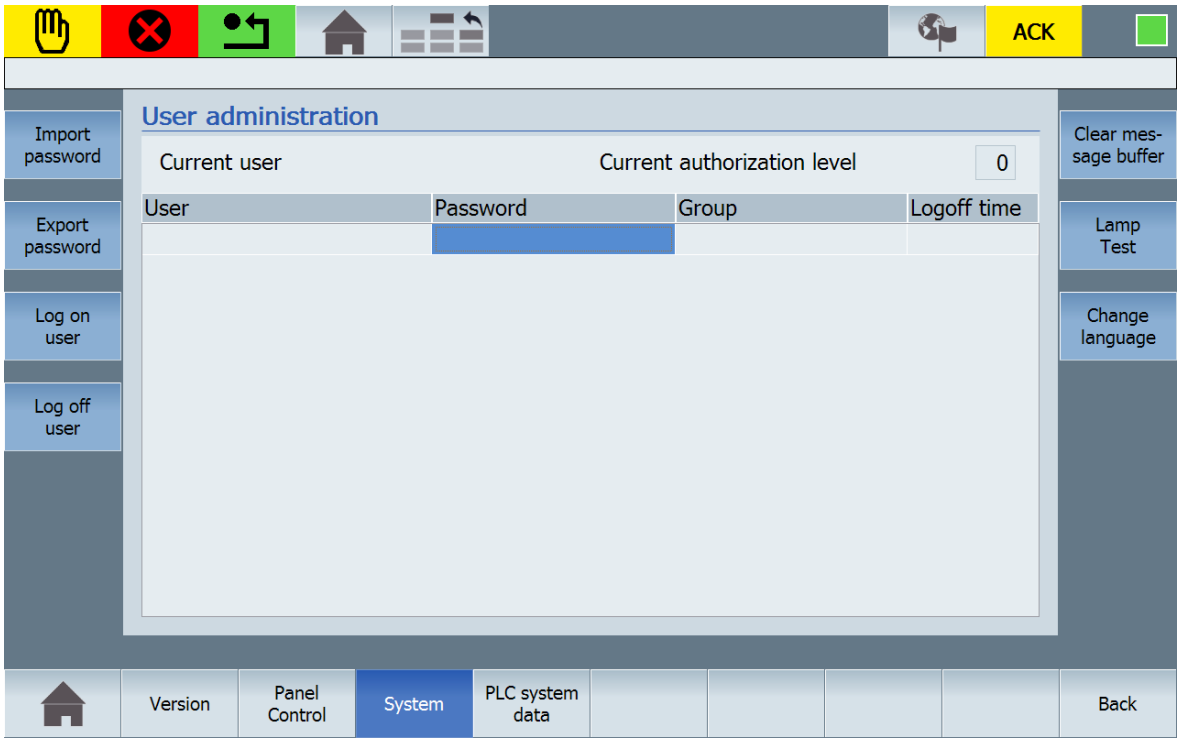


Fig. 11-3: System (SS_01_015_SystemScreen)

Export/import 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 enter the password list on one machine and then transfer it to other machines.

User logon

This function opens the user logon dialog of WinCC. There the user and password can be entered.

User logoff

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

Clear message buffer

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

Lamp test

As long as the softkey or the button is pressed, the tag **LTLL_Data.HMI[X].global.lamptest** has the signal 1.

This tag has to be processed further by the machine manufacturer.

Address	LTLL_Data.HMI[X].global.lamptest
Format	BOOL
Value range	1-signal when the "Lamp test" softkey on the HMI device has been pressed.
Default setting	False

Change language

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

The HMI Lite project is available in thirteen languages:

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

Additional languages can be implemented for specific projects.

11.4 PLC system data

11.4.1 Layout and functionality

The screen shows the PLC cycle time, PLC settings and the network configuration as well as the identification and maintenance data.

The data of the interface that is parameterized in the **LTLL_Basic** function block is displayed in the network configuration area.

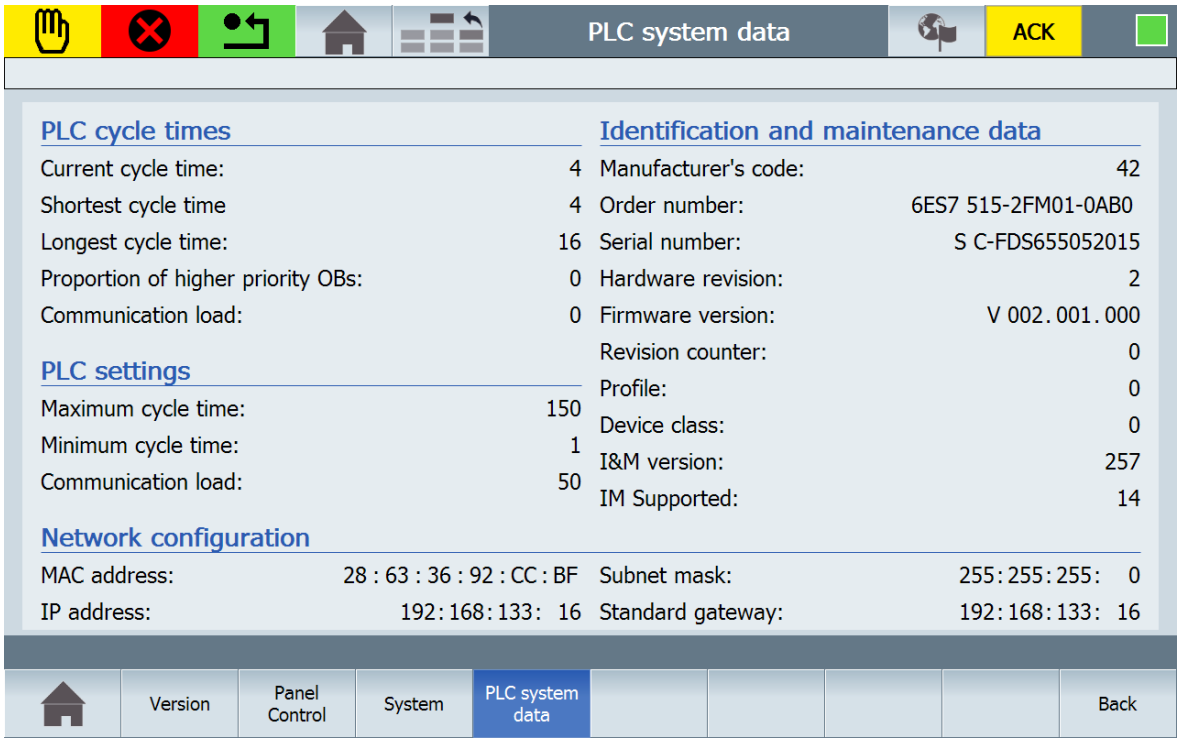


Fig. 11-4: PLC system data (SS_01_016_PLCSystemDaten)

11.4.2 Runtime Interface (LTLL_PLCSystemData)

Block **LTLL_PLCSystemData** supplies the WinCC **PLC system data** screen. The block must be called cyclically once per HMI device.

Call interface

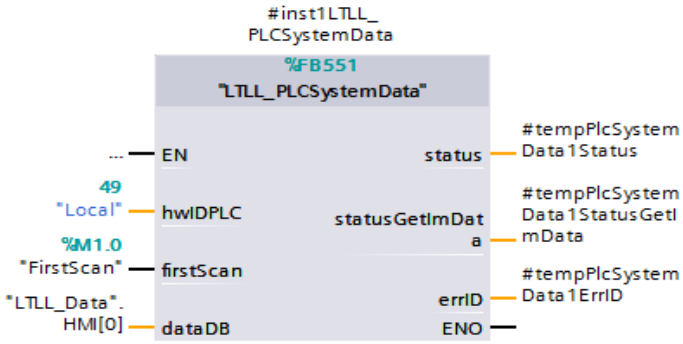


Fig. 11-5: Call interface of block LTLL_PLCSystemData

Parameters

Table 11-1: Parameters of block LTLL_PLCSystemData

Name	Declaration	Type	Standard	Description
hwIDPLC	Input	HW_IO	"Local"	Hardware ID of the PLC
firstScan	Input	BOOL		Startup bit
dataDB	InOut	LTLL_typeData	LTLL_Data.HMI[0]	HMI Lite Runtime data DB
status	Output	WORD	-	Block status
statusGetImData	Output	WORD	-	Status of instruction GET_IM_DATA
errId	Output	WORD	-	Local error handling

Output parameter status

Table 11-2: Description of the output parameter status of LTLL_PLCSystemData

Error code (W#16# ….)	Description
16#8200	HMI Lite licensing failed

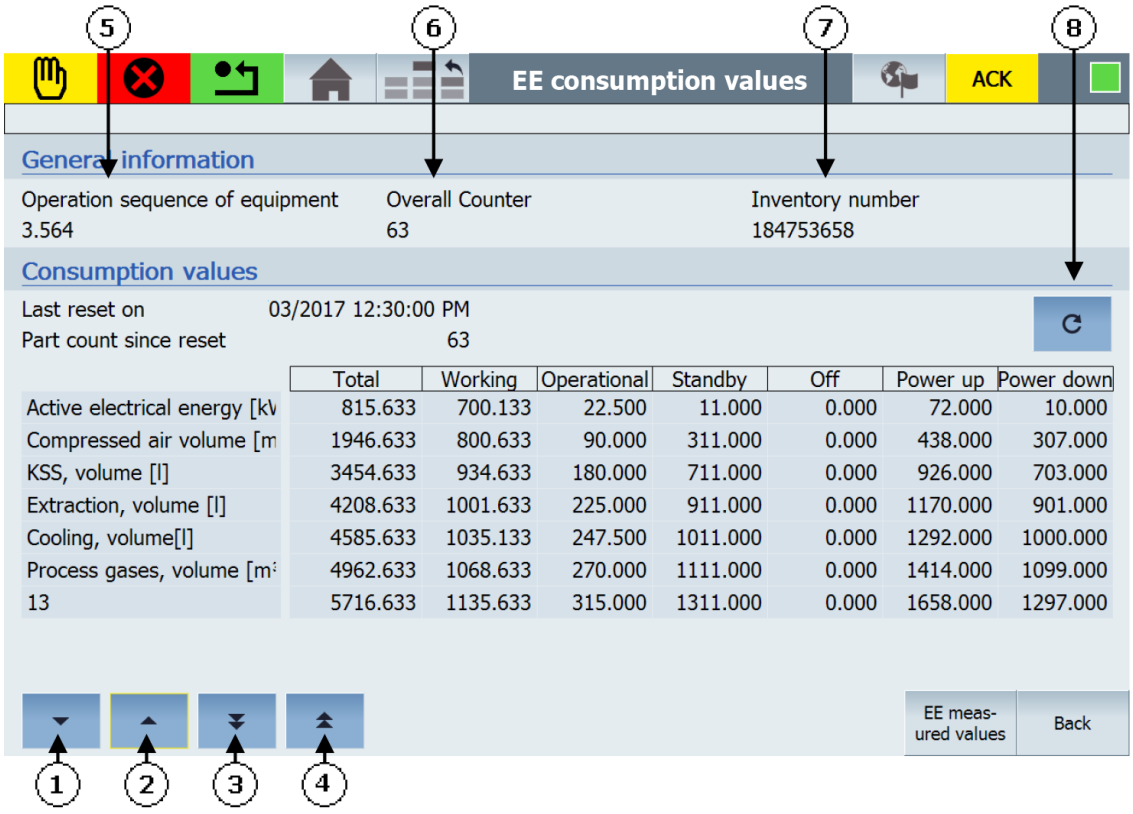
Energy_Efficiency@TRANSLINE

12

EE@TRANSLINE in HMI Lite comprises the two screens **Energy efficiency consumption values** and **Energy efficiency measured values**, as well as the blocks to supply the screens and calculations of the measuring points.

12.1 Energy efficiency consumption values

It is possible to display the energy consumption for different machine states since the last counter reset on this screen. The display is categorized by the predefined energy forms of the energy efficiency data block.



(1) Button to move down by 1 line

(2) Button to move up by 1 line

(3) Button to move down by 4 lines

(4) Button to move up by 4 lines

(5) Maximum 10-character alphanumeric designation of the operation sequence

(6) Total number of produced TRANSLINE workpieces

(7) Machine inventory number

(8) Button for resetting the consumption values

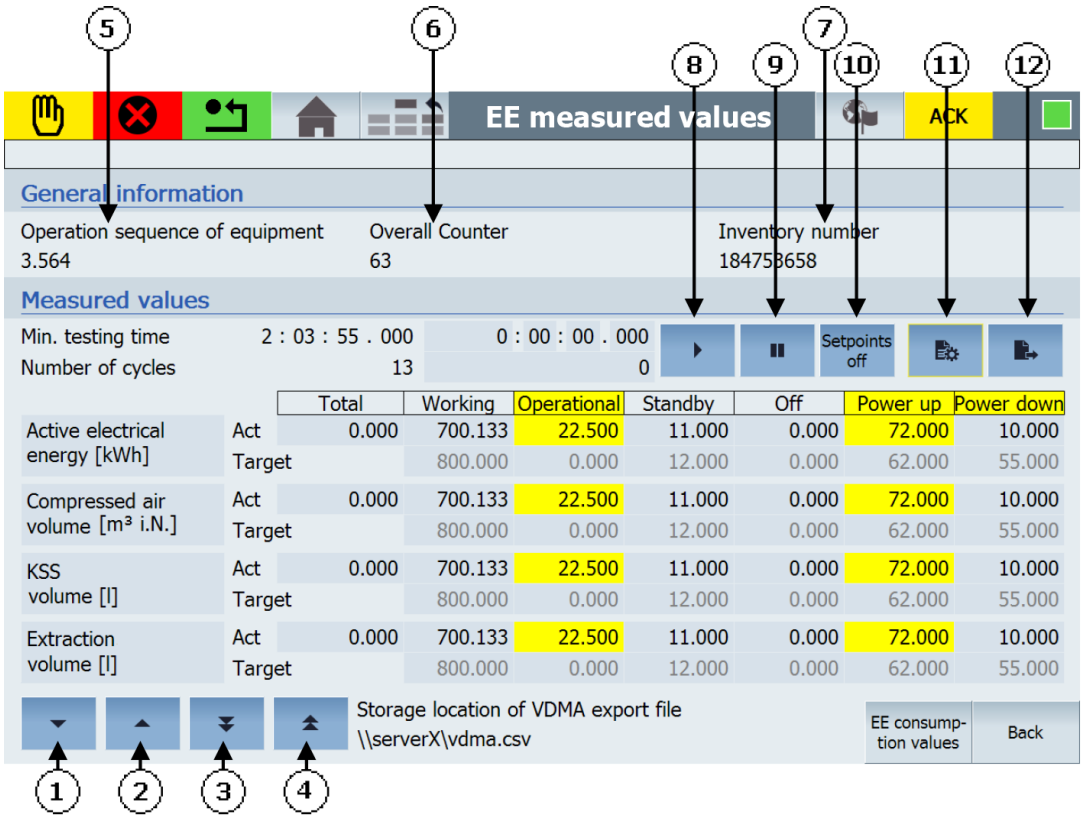
Fig. 12-1: Energy efficiency consumption values (SS_05_001_EnergyEfficiencyEconomy)

Reset

The consumption values of all energy forms can be reset by pressing the button for resetting the consumption values.

12.2 Energy efficiency measured values

It is possible to measure and display the energy consumption for different machine states for a specific time interval and display it on this screen. The display is categorized by the predefined energy forms of the energy efficiency data block. The measured values can be exported as a .csv file.



(1) Button to move down by 1 line

(2) Button to move up by 1 line

(3) Button to move down by 4 lines

(4) Button to move up by 4 lines

(5) Maximum 10-character alphanumeric designation of the operation sequence

(6) Total number of produced TRANSLINE workpieces

(7) Machine inventory number

(8) Button for stopping and starting measurement

(9) Button for pausing measurement

(10) Button for displaying and hiding the setpoints

(11) Button for setting the target path for the VDMA export

(12) Button for exporting the measured values

Fig. 12-2: Energy efficiency measured values (SS_05_002_EnergyEfficiencyMeasurement)

Start measurement

This button starts the energy measurement for a specific period.

The following entries are required:

- Period that should be specified as minimum for the measurement, in the format: hh:mm:ss (Input field **Minimum measuring time**).
- Number of cycles that should be specified as minimum for the measurement (Input field **Number of cycles**).

Measurement is started manually by pressing the button for starting the measurement and is terminated automatically when both the **Minimum measuring time** and the **Number of cycles** have been reached.

The measured values are displayed in the table. If the measured values (Actual) are greater than the desired values (Target), they are marked in yellow.

Table 12-1: Buttons for measuring energy efficiency

Button	Description
	Start measurement
	Pause measurement
	Stop measurement

Targets

Target setpoints can be modified and also be displayed and hidden.

VDMA export

The measured values are exported to a CSV file called *Inventory number* _YYYY-MM-DD_hh:mm.ss.csv A target path to any location has to be specified for the VDMA export. The export of the measured values can then be started.

12.3 Runtime interface (LTLL_EETransfer)

In the PLC, four blocks are provided for the use of EE@TRANSLINE.

The **LTLL_EE** energy efficiency data block is the interface to the energy efficiency screens.

The **LTLL_EETransfer** block supplies the **LTLL_EE** block with the data relevant for the screen.

LTLL_EETransfer must be called in a cyclic interrupt OB (Cyclic interrupt OB).

The measuring points for the energy efficiency (e.g. electrical energy and compressed air) are transferred to **LTLL_EETransfer**. Measuring points that are independent of the energy state (e.g. supply pressure), are not processed via the **EnS_EEm_Calc** block, but instead are written directly to the **LTLL_EE** block.

LTLL_EETransfer also calls the **EnS_EEm_Calc** block internally.

EnS_EEm_Calc is responsible for the scaling, overflow check, and measurements as well as for the distribution of the measuring points according to energy states.

The database for the **EnS_EEm_Calc** block is the **EEmMachine** data block.

All configurations are performed and operations controlled in the **EEmMachine** block.

In HMI Lite, the measuring points are already pre-assigned according to the EE@TRANSLINE concept. For this reason, the following tags of the **EEmMachine** block must not be changed:

- machineConfiguration.eStateName
- measurementsConfiguration[0-7, 9].name
- measurementsOperation[0-7, 9].in

All information for the **EnS_EEm_Calc** and **EEmMachine** blocks can be found in the online help of the TIA Portal.

Information on the pre-assigned measuring points of the **LTLL_EE** data block can be found in the following table:

Table 12-2: Pre-assigned energy efficiency measuring points

No.	Energy type	Measured tag	Unit in the interface data block	HMI
0	Electrical energy	Active energy	kWh	Yes
1	Electrical energy	Apparent energy	kVAh	No
2	Electrical energy	Reactive energy	kvarh	No
3	Compressed air	Volume	normal m^3	Yes
4	Compressed air	Maximum volume flow	normal m^3/h	No
5	Compressed air	Supply pressure	bar	No
6	Compressed air	Maximum supply pressure	bar	No
7	Cooling lubricant	Volume	l	Yes
8	Cooling lubricant	Supply pressure	bar	No
9	Extraction	Volume	l	Yes
10	Cooling	Volume	l	Yes
11	Process gases	Volume flow	m^3	Yes
12-18	Free for parametrization			
19	Axis energy		kWh	Yes

Call interface

Fig. 12-3: Call interface of the LTLL_EETransfer block

Parameters

Table 12-3: Parameters of block LTLL_ EETransfer

Name	Declaration	Type	Standard	Description
operationSeq Equipment	Input	String[10]	---	Operation sequence of the equipment
inventoryNumber	Input	String[20]	---	Inventory number
useActive ElectricalEnergy	Input	BOOL	---	Activate electrical energy - active energy
activeElectrical Energy	Input	LReal	---	Electrical energy - active energy [kWh]
useApparent ElectricalEnergy	Input	BOOL	---	Activate electrical energy - apparent energy
Apparent ElectricalEnergy	Input	LReal	---	Electrical energy - apparent energy [kVAh]
useReactive ElectricalEnergy	Input	BOOL	---	Activate electrical energy - reactive energy
reactiveElectrical Energy	Input	LReal	---	Electrical energy - reactive energy [kvarh]
useCompressed AirVolume	Input	BOOL	---	Activate compressed air - volume
compressedAir Volume	Input	Real	---	Compressed air - volume [normal m³]
useCompressed AirConnection Pressure	Input	BOOL	---	Activate compressed air - supply pressure
compressedAir Connection Pressure	Input	Real	---	Compressed air - supply pressure [bar]
useCooling LubricantVolume	Input	BOOL	---	Activate cooling lubricant - volume
coolingLubricant Volume	Input	LReal	---	Cooling lubricant - volume [l]
useCooling Lubricant Connection Pressure	Input	BOOL	---	Activate cooling lubricant - supply pressure
coolingLubricant Connection Pressure	Input	Real	---	Cooling lubricant - supply pressure [bar]
useExtraction Volume	Input	BOOL	---	Activate extraction - volume

Name	Declaration	Type	Standard	Description
extractionVolume	Input	LReal	---	Extraction - volume [l]
useCooling Volume	Input	BOOL	---	Activate cooling - volume
coolingVolume	Input	LReal	---	Cooling - volume [l]
useProcess GasesVolume	Input	BOOL	---	Activate process gases - volume
processGases Volume	Input	LReal	---	Process gases - volume [m³]
useAxisEnergy	Input	BOOL	---	Activate axis energy
axisEnergy	Input	LReal	---	Axis energy
useUserEnergy1-useUserEnergy7	Input	BOOL	---	Activate user specified energy form 1-7
UserEnergy1-UserEnergy7	Input	Real	---	User specified energy form 1-7
cycleCount	Input	BOOL	---	Increase cycle count in the case of a rising edge
machine Configuration	InOut	EnS_ EEm_ typeConfig	EEmMachine. machine Configuration	Configuration data of the machine
machine Operation	InOut	EnS_ EEm_type Operation	EEmMachine. machineOperation	Operating data of the machine
measurements Configuration	InOut	Array[*] of EnS_ EEm_type MeasConfig	EEmMachine. measurements Configuration	Configuration data of the measuring points
measurements Operation	InOut	Array[*] of EnS_ EEm_type Meas Operation	EEmMachine. measurements Operation	Operating data of the measuring points
ee	InOut	LTLL_typeEE	LTLL_EE	Connection block for the HMI
error	Output	BOOL	---	Block error
status	Output	WORD	---	Block status

Output parameter status

Table 12-4: Description of the output parameter status of LTLP_EETransfer

Value (V#16# ….)	Description
16#8200	Invalid number of measuring points
16#8201	A lower limit of the transferred arrays in not equal to 0
16#8202	The array upper limits do not match the array upper limits of the machine configuration

Note

For a description of all other values, please refer to the Online Help of the TIA Portal under Parameter **status** of the **EnS_EEm_Calc**.

Appendix

A

A.1 List of abbreviations

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

A.2 Change index

A.2.1 2020 Edition

Newly created/first edition for HMI Lite V15.1 and SIMATIC S7-1500

Get more information

Solutions for Powertrain:
www.siemens.com/TRANSLINE

Solutions for Powertrain Extranet:
https://support.industry.siemens.com/cs/ww/en/view/109744084

Automotive Manufacturing:
www.siemens.com/automotive

SINUMERIK CNC Automation System:
www.siemens.com/sinumerik

SINAMICS Drive Technology:
www.siemens.com/sinamics

Motion Control Systems and Solutions for Production Machines and Machine Tools:
www.siemens.com/motioncontrol

Industry Online Support (Service and Support):
www.siemens.com/online-support

IndustryMall:
www.siemens.com/industrymall

Siemens AG
Digital Industries
Motion Control
Postfach 3180
91050 Erlangen
Germany

Scan the QR-Code for information regarding Solutions for Powertrain