SIEMENS

Create MyHMI /Automotive PLC blocks

Function Manual

10/2023
A5E53211851B 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
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E53211851B AA
Ⓟ 11/2023 Subject to change

Table of contents

Introduction 1

1.1 About SINUMERIK

From simple, standardized CNC machines to premium modular machine designs – the SINUMERIK CNCs offer the right solution for all machine concepts. Whether for individual parts or mass production, simple or complex workpieces – SINUMERIK is the highly dynamic automation solution, integrated for all areas of production. From prototype construction and tool design to mold making, all the way to large-scale series production.

Visit our website for more information SINUMERIK (https://www.siemens.com/sinumerik).

1.2 About this documentation

Standard scope

This documentation only describes the functionality of the standard version. This may differ from the scope of the functionality of the system that is actually supplied. Please refer to the ordering documentation only for the functionality of the supplied drive system.

It may be possible to execute other functions in the system which are not described in this documentation. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

For reasons of clarity, this documentation cannot include all of the detailed information on all product types. Further, this documentation cannot take into consideration every conceivable type of installation, operation and service/maintenance.

The machine manufacturer must document any additions or modifications they make to the product themselves.

Websites of third-party companies

This document may contain hyperlinks to third-party websites. Siemens is not responsible for and shall not be liable for these websites and their content. Siemens has no control over the information which appears on these websites and is not responsible for the content and information provided there. The user bears the risk for their use.

1.3 Documentation on the internet

1.3.1 Documentation overview SINUMERIK ONE

Comprehensive documentation about the functions provided in SINUMERIK ONE Version 6.13 and higher is provided in the Documentation overview SINUMERIK ONE (https://support.industry.siemens.com/cs/ww/en/view/109768483).

You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- User: Operating
- User: Programming
- Manufacturer/Service: Functions
- Manufacturer/Service: Hardware
- Manufacturer/Service: Configuration/Setup
- Manufacturer/Service: Safety Integrated
- Information and training
- Manufacturer/Service: SINAMICS

1.3.2 Documentation overview SINUMERIK 840D sl

Comprehensive documentation about the functions provided in SINUMERIK 840D sl Version 4.8 SP4 and higher is provided in the Documentation overview SINUMERIK 840D sl (https://support.industry.siemens.com/cs/ww/en/view/109766213).

You can display the documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- User: Operating
- User: Programming
- Manufacturer/Service: Functions
- Manufacturer/Service: Hardware
- Manufacturer/Service: Configuration/Setup
- Manufacturer/Service: Safety Integrated
- Manufacturer/Service: SINUMERIK Integrate/MindApp
- Information and training
- Manufacturer/Service: SINAMICS

1.3.3 Documentation overview SINUMERIK 828D

Comprehensive documentation about the functions provided in SINUMERIK 828D Version 4.8 SP4 and higher is provided in the 828D documentation overview (https://support.industry.siemens.com/cs/ww/en/view/109766724).

You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- User: Operating
- User: Programming
- Manufacturer/Service: Configuring
- Manufacturer/Service: Commissioning
- Manufacturer/Service: Functions
- Manufacturer/Service: Safety Integrated
- SINUMERIK Integrate/MindApp
- Info & Training

1.3.4 Documentation overview SINUMERIK MC

Comprehensive documentation about the functions provided in SINUMERIK MC Version 1.12 and higher is provided in the SINUMERIK MC documentation overview (https://support.industry.siemens.com/cs/ww/en/view/109771043).

You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- User: Operating
- User: Programming
- Manufacturer/Service: Commissioning
- Manufacturer/Service: Functions
- Manufacturer/Service: Hardware
- Manufacturer/Service: Safety Integrated
- Information and training

1.3.5 Documentation overview SINUMERIK 808D

Comprehensive documentation about the functions provided in SINUMERIK 808D Version 4.91 and higher is provided in the Documentation overview SINUMERIK 808D (https://support.industry.siemens.com/cs/ww/en/view/109780845).

You can display documents or download them in PDF format.

The documentation is divided into the following categories:

- User: Operating
- Manufacturer/Service: Configuring/Commissioning

1.3.6 Documentation overview SINUMERIK operator components

Comprehensive documentation about the SINUMERIK operator components is provided in the Documentation overview SINUMERIK operator components (https://support.industry.siemens.com/cs/document/109783841/technische-dokumentation-zu-sinumerik-bedienkomponenten?dti=0&lc=en-WW).

You can display documents or download them in PDF and HTML5 format.

The documentation is divided into the following categories:

- Operator Panels
- Machine control panels
- Machine Pushbutton Panel
- Handheld Unit / Mini handheld devices
- Further operator components

An overview of the most important documents, entries and links to SINUMERIK is provided at SINUMERIK Overview - Topic Page (https://support.industry.siemens.com/cs/document/109766201/sinumerik-an-overview-of-the-most-important-documents-and-links?dti=0&lc=en-WW).

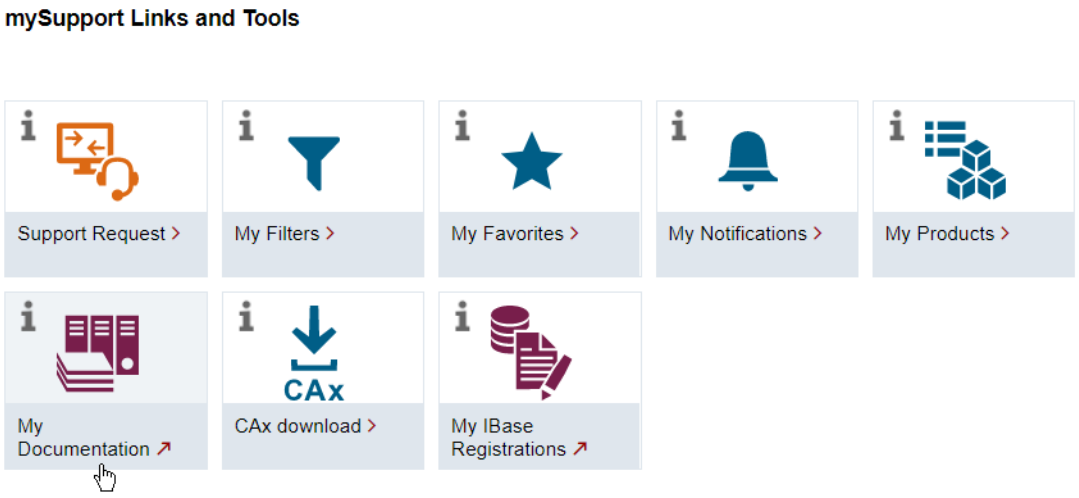
1.4 Feedback on the technical documentation

If you have any questions, suggestions or corrections regarding the technical documentation which is published in the Siemens Industry Online Support, use the link "Provide feedback" which appears at the end of the entry.

1.5 mySupport documentation

With the "mySupport documentation" web-based system you can compile your own individual documentation based on Siemens content, and adapt it for your own machine documentation.

To start the application, click on the "My Documentation" tile on the mySupport homepage (https://support.industry.siemens.com/cs/ww/en/my):



The configured manual can be exported in RTF, PDF or XML format.

Note

Siemens content that supports the mySupport documentation application can be identified by the presence of the "Configure" link.

1.6 Service and Support

Product support

You can find more information about products on the internet:

Product support (https://support.industry.siemens.com/cs/ww/en/)

The following is provided at this address:

- Up-to-date product information (product announcements)
- FAQs (frequently asked questions)
- Manuals
- Downloads
- Newsletters with the latest information about your products
- Global forum for information and best practice sharing between users and specialists
- Local contact persons via our Contacts at Siemens database (→ "Contact")
- Information about field services, repairs, spare parts, and much more (→ "Field Service")

Technical support

Country-specific telephone numbers for technical support are provided on the internet at address (https://support.industry.siemens.com/cs/ww/en/sc/4868) in the "Contact" area.

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

Training

You can find information on SITRAIN at the following address (https://www.siemens.com/sitrain).
SITRAIN offers training courses for automation and drives products, systems and solutions from Siemens.

Siemens support on the go

With the award-winning "Industry Online Support" app, you can access more than 300,000 documents for Siemens Industry products – any time and from anywhere. The app can support you in areas including:

- Resolving problems when implementing a project
- Troubleshooting when faults develop
- Expanding a system or planning a new system

Furthermore, you have access to the Technical Forum and other articles from our experts:

- FAQs
- Application examples
- Manuals
- Certificates
- Product announcements and much more

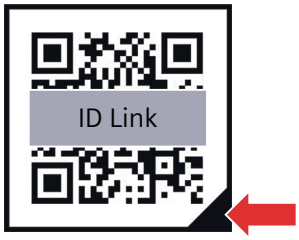
The "Industry Online Support" app is available for Apple iOS and Android.

Digital product information as QR code

The ID link according to IEC 61406 provides you with digital information about your product.

The ID link is a globally unique identifier and is located as a QR code on the product and the product packaging.

You can recognize the ID link by the frame with a black corner at the bottom right.



In addition to the digital nameplate you will find the following product information:

- Technical specifications
- FAQs
- Manuals
- Certificates
- Product announcements
- Application examples

1.7 OpenSSL

Using OpenSSL

This product can contain the following software:

- Software developed by the OpenSSL project for use in the OpenSSL toolkit
- Cryptographic software created by Eric Young.
- Software developed by Eric Young

You can find more information on the internet:

- OpenSSL (https://www.openssl.org)
- Cryptsoft (https://www.cryptsoft.com)

1.8 Compliance with the General Data Protection Regulation

Siemens observes standard data protection principles, in particular the data minimization rules (privacy by design).

For this product, this means:

The product does not process or store any personal data, only technical function data (e.g. time stamps). If the user links this data with other data (e.g. shift plans) or if he/she stores person-related data on the same data medium (e.g. hard disk), thus personalizing this data, he/she must ensure compliance with the applicable data protection stipulations.

Configuring Automotive PLC program plus blocks 2

2.1 Overview

Overview

NCU device interacts with automotive screens using the PLC blocks. The PLC blocks are responsible for coordination of the interface DBs and HMI. Each block has specific PLC data types (UDT). Use the specific UDT of the block to configure the block. For more information on using PLC data types, refer to Using PLC data types (UDT) in TIA Portal online help.

The following blocks are provided for SINUMERIK ONE:

- Manual operation
- PLC system data
- Tool life monitoring
- Process dashboard
- Hardware diagnostics
  - SINAMICS diagnostics
  - SINAMICS technology objects
  - Safety
  - Motor Starter
  - RFID
  - EKS
  - Interface
  - Energy

The following blocks are provided for PLC:

- Manual Operation
- PLC system data
- Tool life monitoring

- Process dashboard
- Hardware Diagnostics
  - SINAMICS diagnostics
  - SINAMICS technology objects
  - Safety
  - Motor Starter
  - RFID
  - EKS
  - Camera
  - Interface
  - Energy

2.2 Configuring PLC blocks

Procedure

Follow the below procedure to add block in cyclic OB:

1. In the cyclic block or main block, call "LAPP_Init" block and interface the input parameters. For more information, refer to LAPP_Init (Page 20).

   Note

   This is a mandatory block to be called.

2. Drag the LAPP_<block name> block from the "SINUMERIK ONE PLC blocks" and add it to the cyclic block or main block.
3. Create a data block and create the required PLC tags using UDT of the block. For more information on using UDT, refer to Using PLC data types (UDT) in TIA Portal online help.
4. Assign the necessary inputs and outputs of the block.

Note

- Runtime screens are inter-dependent and will not work as expected if you call wrong PLC programming block/functions.
- Refer to the subsequent topics to know more about the data types, input and output parameters of the library blocks.

Example

Using UDT for "LAPP_camera"

To use PLC data type within a data block as data type of a tag. Proceed as follows:

1. Double-click the "Add new block" command.
   The "Add new block" dialog box opens.
2. Click the "Data block (DB)" button.
3. Enter the name Demo and click "OK".
4. In Demo DB, create tag camera and select "LAPP_typeCameraIPConfig" as data type.
   or
   In Demo DB, create tag camera and select Array[0...2] of "LAPP_typeCameraIPConfig" as data type. (You can configure up to 3 cameras).
5. Create a "hmiIndex" tag with USInt as data type.
6. Call the "LAPP_camera" block and interface the parameters as shown below.

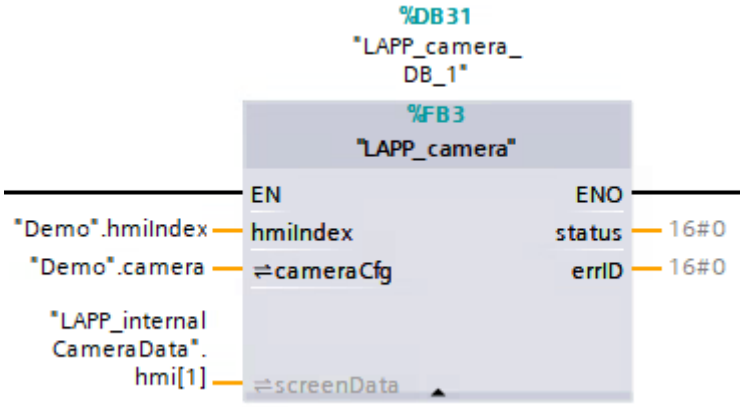


Figure 2-1 LAPP_camera block

2.3 LAPP_Init

2.3.1 Block interface - LAPP_Init for PLC

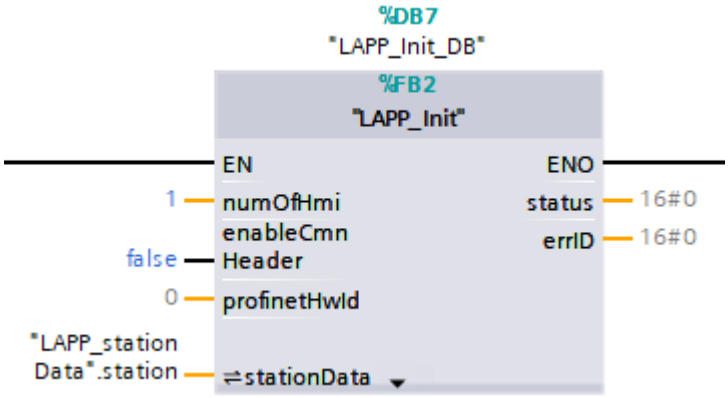


Figure 2-2 LAPP_Init block

Parameters

Table 2-1 Parameters of the LAPP_Init block

Name	Declaration	Type	Description
numOfHmi	Input	USInt	Determines number of HMI required.
enable-CmnHeader	Input	Bool	Enables the common header across all the HMIs configured. Displays the selected station details across all the HMIs. • 0 - Disable • 1 - Enable For example, Consider that you have 3 HMIs with common header enabled and station 4 is selected. • If you select station 10 on HMI 2, then the information of machine info, transline mode, basic position, plant status, and part ready of station 10 is displayed on all the 3 HMI's header. • If you select station 1 on HMI 2, then the information of machine info, transline mode, basic position, plant status, and part ready of station 10 is displayed on all the 3 HMI's header. **Note**: If common header is not enabled, then each HMI header will show the details of the respective station selected.
profinetHwId	Input	HW_IO	Interface the system constant value of the PROFINET port used for license generation.
stationData	InOut	Array of LAPP_typeStation	Interface the number of stations available. For more information, refer to Configuring station data (Page 23).

Name	Declaration	Type	Description
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information, refer to Error codes (Page 22).

Hardware ID of the PROFINET interface

In "Device view", select the PROFINET interface and navigate to "Properties > System constant" tab in the inspector window.

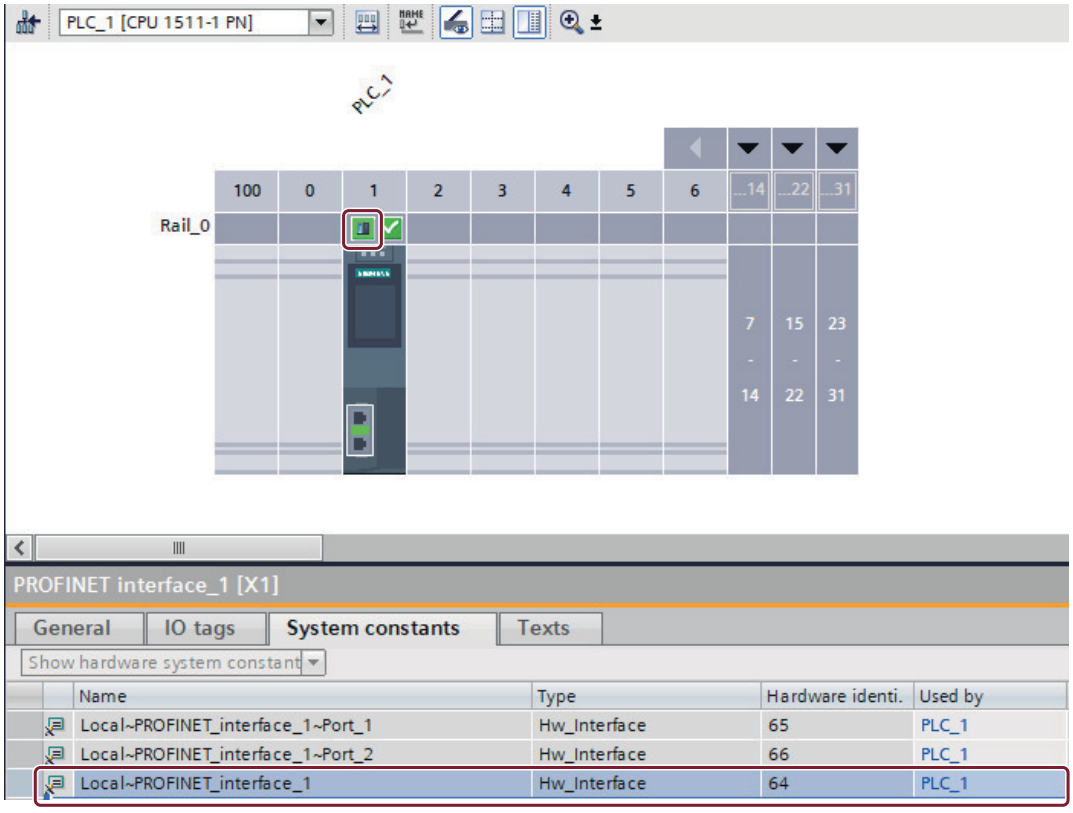


Figure 2-3 PROFINET interface hardware ID

2.3.2 Block interface - LAPP_Init for CNC

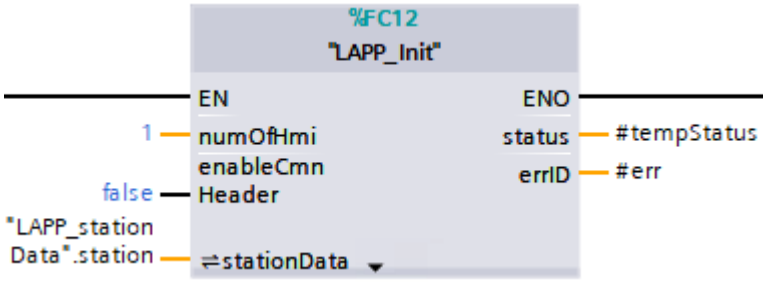


Figure 2-4 LAPP_Init block

Parameters

Table 2-2 Parameters of the LAPP_Init block

Name	Declaration	Type	Description
numOfHmi	Input	USInt	Determines number of HMI required.
enable-CmnHeader	Input	Bool	Enables the common header across all the HMIs configured. Displays the selected station details across all the HMIs. • 0 - Disable • 1 - Enable For example, Consider that you have 3 HMIs with common header enabled and station 4 is selected. • If you select station 10 on HMI 2, then the information of machine info, transline mode, basic position, plant status, and part ready of station 10 is displayed on all the 3 HMI's header. • If you select station 1 on HMI 2, then the information of machine info, transline mode, basic position, plant status, and part ready of station 10 is displayed on all the 3 HMI's header. **Note**: If common header is not enabled, then each HMI header will show the details of the respective station selected.
stationData	InOut	Array of LAPP_typeStation	Interface the number of stations available. For more information, refer to Configuring station data (Page 23).
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 22).

2.3.3 Error codes

List

Following are the error codes for "LAPP_Init".

Table 2-3 Error codes for LAPP_Init

Error code	Description
16#8201	Invalid HMI index.
16#8202	Invalid array value configuration for stations.

2.4 Configuring station data

2.4.1 Configuring header

Using "LAPP_Init" mandatory configuration block, you can configure header for different stations.

Configuring number of stations

Configure the number of stations required by modifying the array value in "LAPP_station.station".

Name of the station and message can be modified in the textlist (Page 107).

For example,
Scenario 1
To configure maximum stations to 10, then enter the value at "LAPP_station.station" as shown:

LAPP_stationData

	Name	Data type
1	Static	
2	station	Array[1..10] of "LAPP_typeStation"
3	station[1]	"LAPP_typeStation"
4	station[2]	"LAPP_typeStation"
5	station[3]	"LAPP_typeStation"
6	station[4]	"LAPP_typeStation"
7	station[5]	"LAPP_typeStation"
8	station[6]	"LAPP_typeStation"
9	station[7]	"LAPP_typeStation"
10	station[8]	"LAPP_typeStation"
11	station[9]	"LAPP_typeStation"
12	station[10]	"LAPP_typeStation"

Figure 2-5 LAPP_stationData

Result
In runtime, you can view as below:

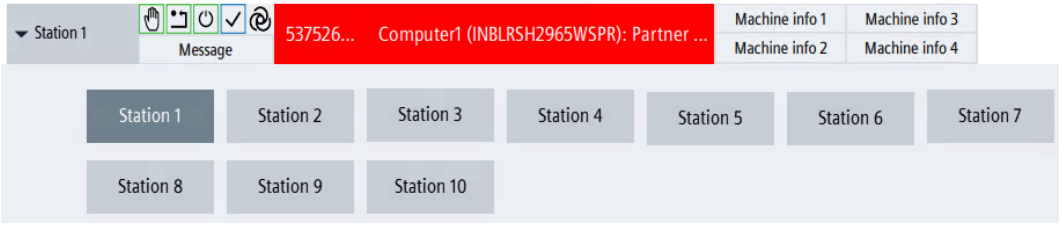


Figure 2-6 Header - Station list

Scenario 2
To configure maximum stations to 14, then enter the value at "LAPP_station.station" as shown:

LAPP_stationData

		Name	Data type
1		Static	
2		station	Array[1..14] of "LAPP_typeStation"
3		station[1]	"LAPP_typeStation"
4		station[2]	"LAPP_typeStation"
5		station[3]	"LAPP_typeStation"
6		station[4]	"LAPP_typeStation"
7		station[5]	"LAPP_typeStation"
8		station[6]	"LAPP_typeStation"
9		station[7]	"LAPP_typeStation"
10		station[8]	"LAPP_typeStation"
11		station[9]	"LAPP_typeStation"
12		station[10]	"LAPP_typeStation"
13		station[11]	"LAPP_typeStation"
14		station[12]	"LAPP_typeStation"
15		station[13]	"LAPP_typeStation"
16		station[14]	"LAPP_typeStation"

Figure 2-7 LAPP_stationData

Result
In runtime, you can view as below:

Figure 2-8 Header - Station list

2.4.2 Data type

LAPP_typeStation

Table 2-4 Structure of data type LAPP_typeStation

Name	Data type	Description
status	LAPP_typeStationStatus	You can configure the translation mode, initial position, plant, part ready, and MSO. For more information, refer to Base elements, SVG in TIA Portal online help.
Info	LAPP_typeInfo	You can configure the machine status information. For more information, refer to Editing Machine Information (Page 106).

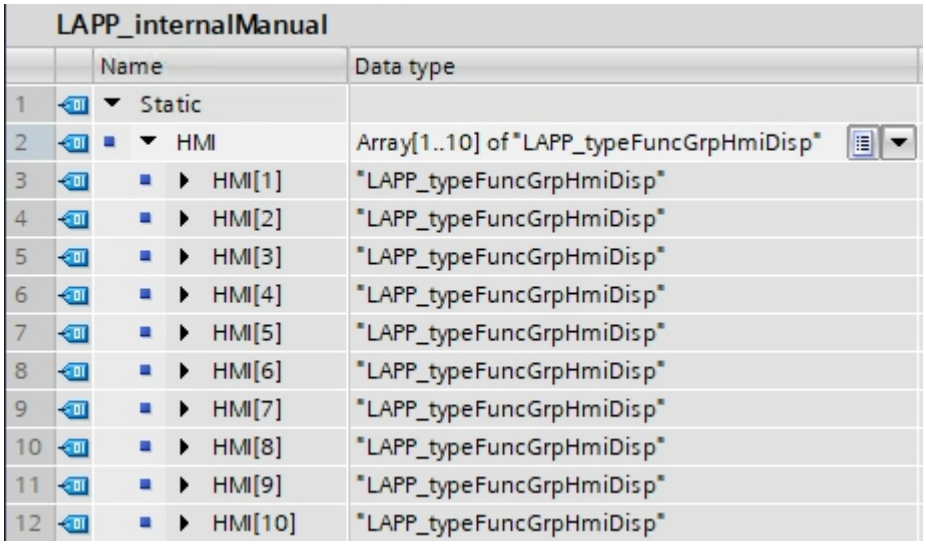
2.5 Configuring PLC blocks for different HMI

To get the data on multiple HMI's, you need to configure two blocks with different HMI index and screen data value in the block interface.

You need to add the number of HMI's required in LAPP_internalXXX DB for all the blocks. XXXX is name of the block.

Example for adding multiple HMI

Scenario 1
To configure maximum HMI to 10, then enter the array value at "HMI" as shown:

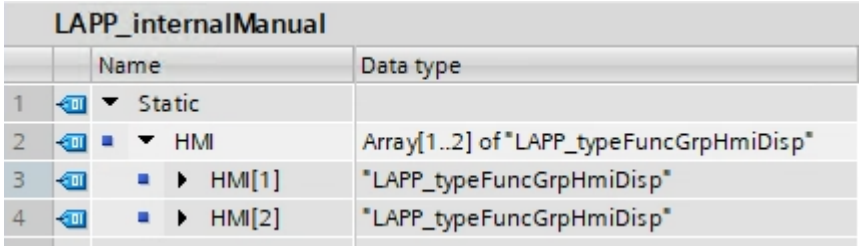


LAPP_internalManual

		Name	Data type
1		Static	
2		HMI	Array[1..10] of "LAPP_typeFuncGrpHmiDisp"
3		HMI[1]	"LAPP_typeFuncGrpHmiDisp"
4		HMI[2]	"LAPP_typeFuncGrpHmiDisp"
5		HMI[3]	"LAPP_typeFuncGrpHmiDisp"
6		HMI[4]	"LAPP_typeFuncGrpHmiDisp"
7		HMI[5]	"LAPP_typeFuncGrpHmiDisp"
8		HMI[6]	"LAPP_typeFuncGrpHmiDisp"
9		HMI[7]	"LAPP_typeFuncGrpHmiDisp"
10		HMI[8]	"LAPP_typeFuncGrpHmiDisp"
11		HMI[9]	"LAPP_typeFuncGrpHmiDisp"
12		HMI[10]	"LAPP_typeFuncGrpHmiDisp"

Figure 2-9 Internal DB

Scenario 2
To configure maximum HMI to 2, then enter the array value at "HMI" as shown:



LAPP_internalManual

		Name	Data type
1		Static	
2		HMI	Array[1..2] of "LAPP_typeFuncGrpHmiDisp"
3		HMI[1]	"LAPP_typeFuncGrpHmiDisp"
4		HMI[2]	"LAPP_typeFuncGrpHmiDisp"

Figure 2-10 Internal DB

Example for configuring data in multiple HMI

To view camera data on two different HMI's, configure the block as shown:

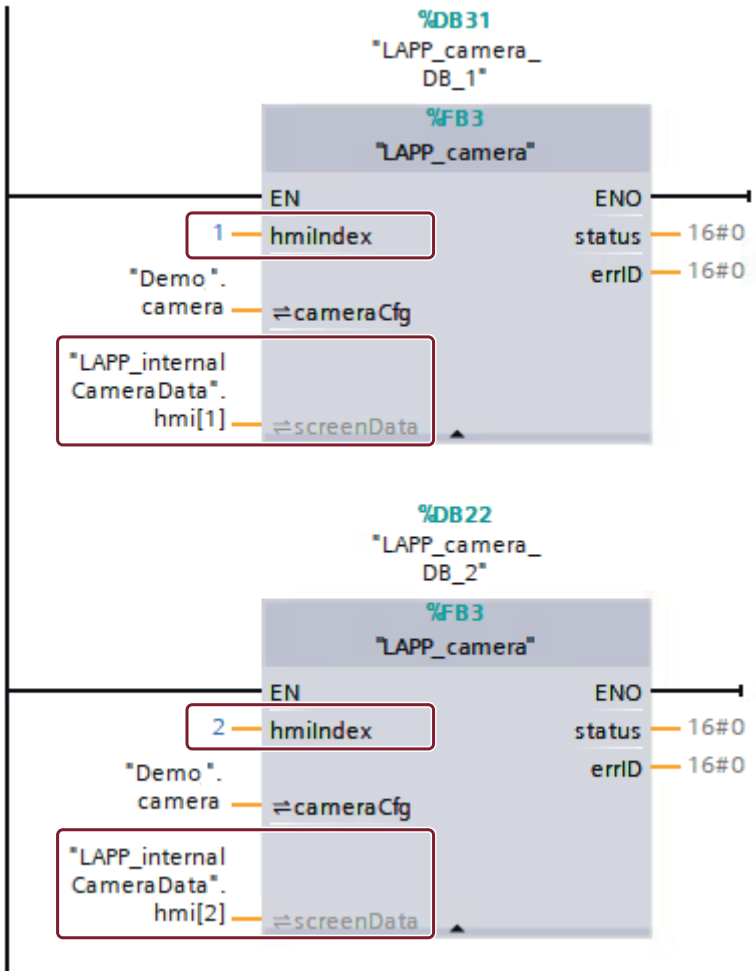


Figure 2-11 LAPP_camera block

Note

Ensure to interface the same HMI device number at "hmiIndex" and "screenData" value.

2.6 Updating the project with new library version

The libraries of a project needs be upgraded if you open the project in the higher version of "SINUMERIK ONE PLC Automotive Program Vxx.xx.xx.xx" .

Prerequisite

Save all the preconfigured data done by OEM along with modifications done to the Create MyHMI /Automotive libraries and screens before upgrading libraries.

Procedure

Follow the below procedure to upgrade the libraries:

1. In “Libraries” task card, navigate to “Global Libraries > SINUMERIK ONE PLC Automotive Program Vxx.xx.xx.xx > Master copies > SINUMERIK ONE PLC Program" folder.

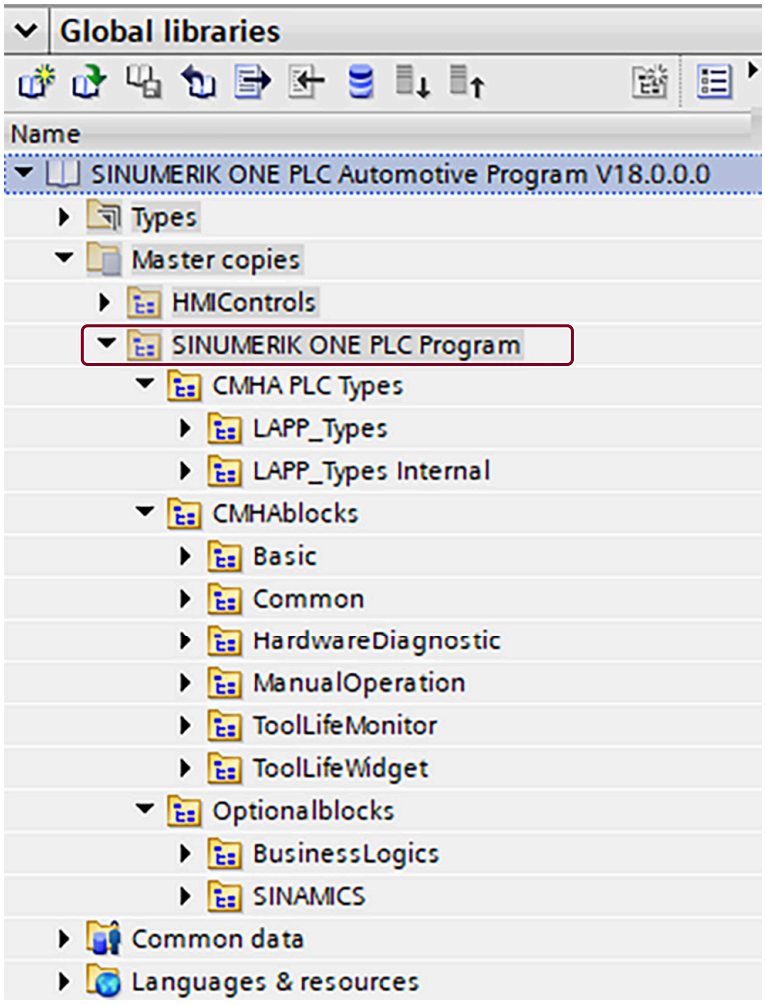


Figure 2-12 SINUMERIK ONE PLC Program

2. Drag "CMHAblocks" folder to "NCU [NCU/PPU xxx] > Program blocks".
3. Drag "CMHA PLC Types" folder to "NCU [NCU/PPU xxx] > PLC data types", as per the version of SINUMERIK ONE.

4. In “Libraries” task card, navigate to "Global Libraries > SINUMERIK ONE PLC Automotive Program Vxx.xx.xx.xx > Master copies > HMI Controls" folder.

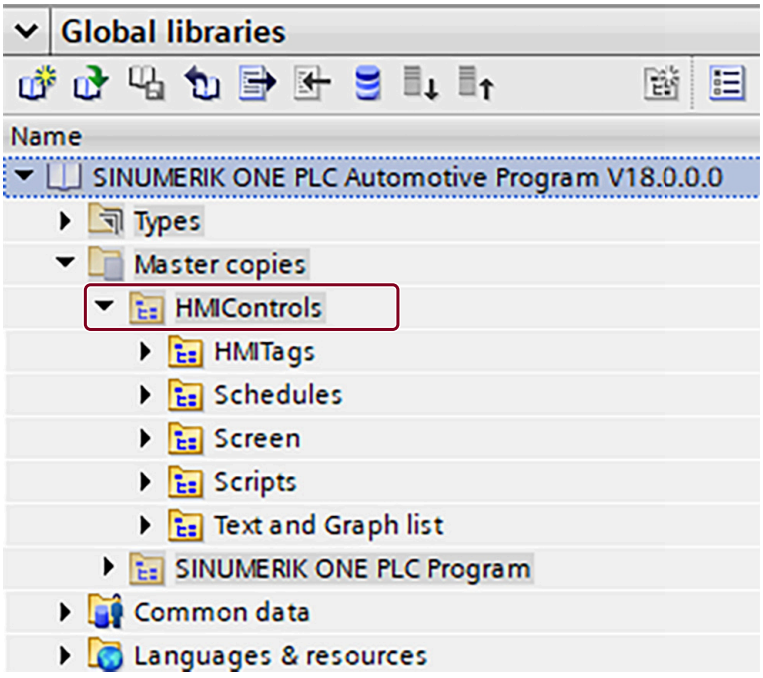


Figure 2-13 HMI Controls

5. Drag the "HMI tags", "Text and graphics lists", "Scripts", "Scheduled tasks" folder from "Global Libraries" to IPC.
6. Compile the software and hardware. For more information, refer to Compiling software and hardware in TIA Portal online help.
7. In “Libraries” task card, navigate to “Global Libraries > SINUMERIK ONE PLC Automotive Program Vxx.xx.xx.xx”.
8. Right-click "SINUMERIK ONE PLC Automotive Program Vxx.xx.xx.xx” and select "Update types > Project" command from the shortcut menu.
9. The "Update types in the project" dialog box opens.

10. Select either the entire project or individual devices for the update.
    The "Update instances in the project" check box is always selected during this process.

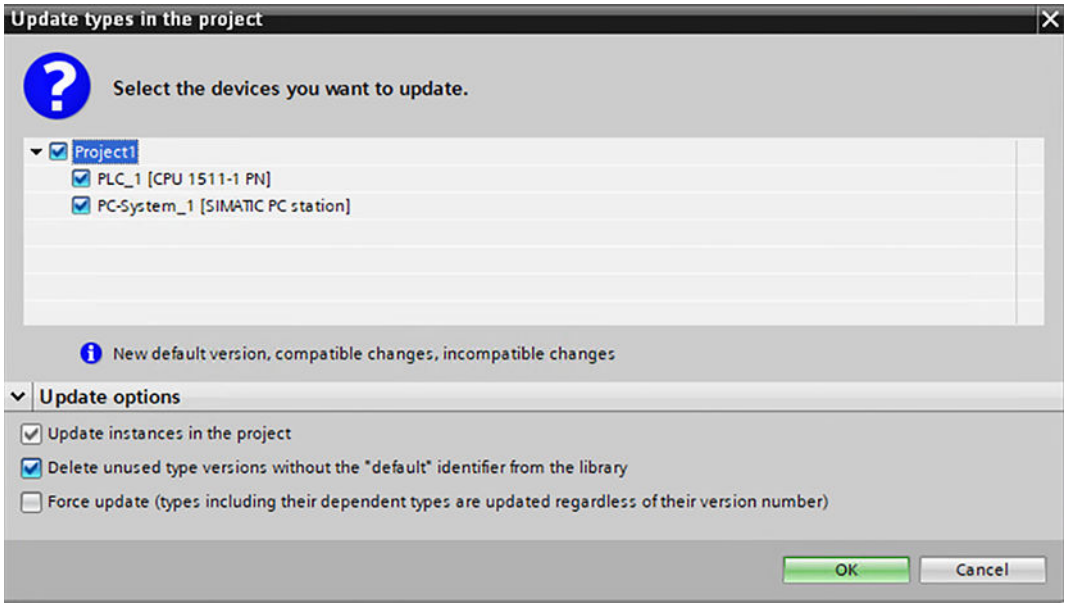


Figure 2-14 Update types in the project

Note

- By default, when the project is updated to the latest type instances, unused type instances are deleted from the project.
  To prevent deleting unused type instances from the project, under "General > Library settings" select the check box "Do not remove unused type version instances when updating the project from the library".
- If you do not delete unused type from the project, then you need to fix the incompatible changes manually. For more information to fix the inconsistent state, refer to TIA Portal online help (Information System > Using libraries > Using types and their versions > Working with types in the project library > Fix inconsistencies").

11. Click "OK".
    If you have made compatible changes, the upgrade is performed.
    If you have made incompatible changes, a warning appears due to an inconsistent state of the project library. For more information to fix the inconsistent state, refer to TIA Portal online help (Information System > Using libraries > Using types and their versions > Working with types in the project library > Fix inconsistencies).

Note

Once the libraries are updated, if the duplicate HMI alarms are created with the names “1_1”, 2_2”, “3_3” and so on.
Delete the duplicate alarms for the libraries to work as intended.

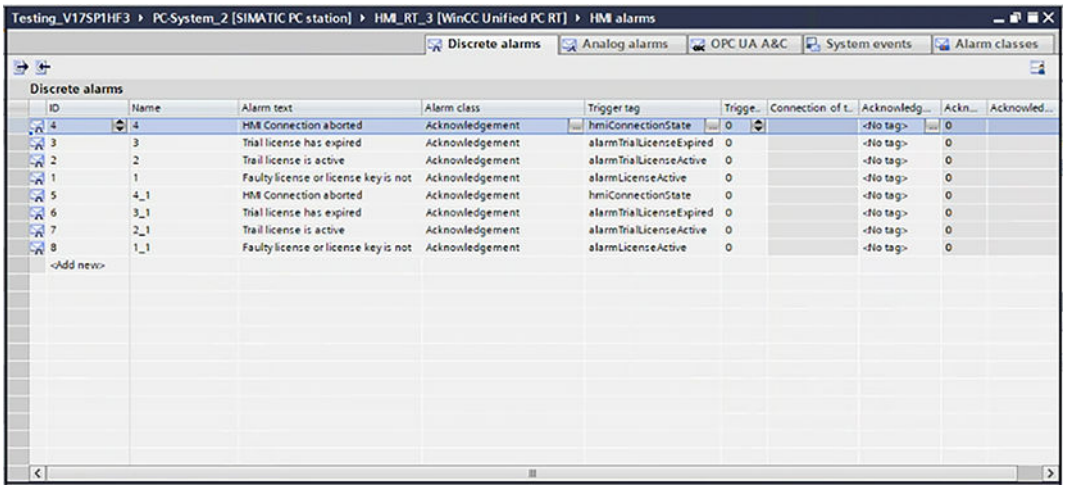


Figure 2-15 Duplicate HMI alarms

Note

- Once the libraries are updated, copy the OEM screens or PLC blocks which is saved before upgrade, so that all the changes are reflected.
- Similarly, you can update the HMI screens also.

Result

All the older versions are deleted from the project and updated to the most recent versions.

2.7 Manual operation

2.7.1 Manual Operation as a screen

2.7.1.1 Overview

Manual operation that can be handled in automotive manufacturing for production assembly lines and can be programmed using Manual Operation block. For example, you can program power-up conditions, drive movements, handling nut runners, handling cycles etc.

The below image gives the workflow details of manual operation.

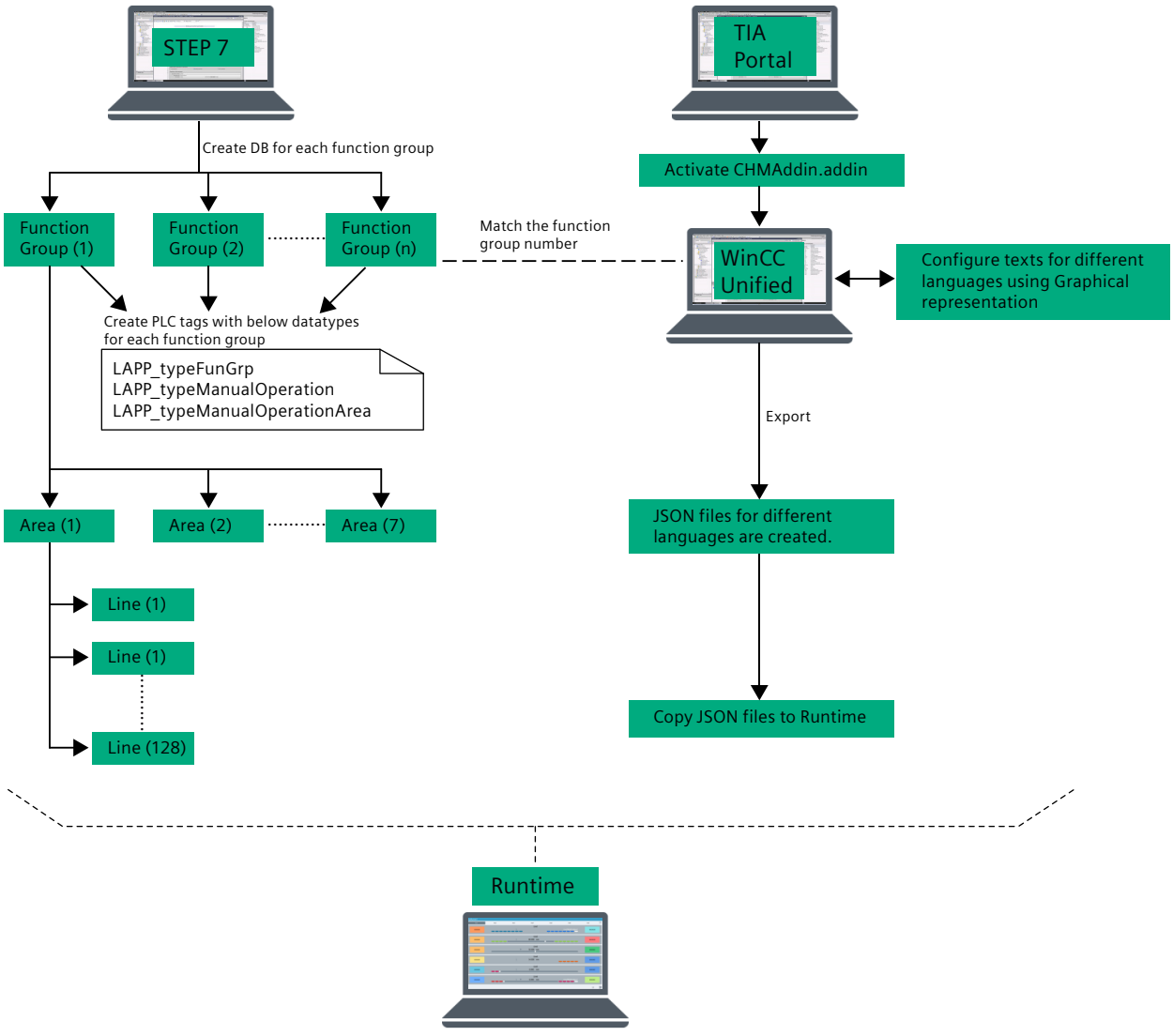


Figure 2-16 Manual Operation configuration workflow

Note

Maximum of 7 areas can be configured in individual function group (for each type of functions like nutrunner, cycles, powerup and so on).

Based on the requirement, you can create individual function groups for each program. You can group the manual operation lines into seven different areas under each function group.

You need to define the number of lines (Page 44) required for each function group. You can group the lines into different areas which can be represented and operated using the screens during runtime.

The following data types (Page 43) are used to configure manual operation block:

- "LAPP_typeFuncGrp"
- "LAPP_typeManualOperationArea"
- "LAPP_typeManualOperation"

You need to configure the following for manual operation screen to work as intended:

- Block configuration in step 7
  - "LAPP_manualOperationMain" - configure HMI index.
  - "LAPP_funcGrpConfig (Page 37)" - configure the number of lines, areas, and the HMI index required for the function group.
  - "LAPP_manualOperation (Page 38)" - configure the mode of operation and the manual operation line parameters.

  For more details on configuring the blocks, refer to Configuring PLC blocks (Page 18).
- Text list and screen configuration (Page 109) in WinCC Unified - activate the "CHMAddin" add-in, configure the line properties.

In runtime, you can view the manual operation screen as shown:

Figure 2-17 Manual operation screen

Note

All the buttons, end positions, position indicator and actor set is configurable, you can adapt the manual operation as per the requirement and configure the lines for powerup, movement, setup, cycle times, nut runners and so on.

For more information on runtime behavior, refer to Manual operation in TIA Portal online help.

2.7.1.2 Manual operation as a screen

For illustration purpose, the following PLC tag names are used:

- lines
- funcGrp
- areas

Procedure

1. Follow the steps from Configuring PLC blocks (Page 18) to configure the mandatory block and create DB. For example, for first function group create a DB "NutRunner1"

Note

You can create different data block for different function groups.

2. In the DB, create 3 tags with data types "LAPP_typeFunGrp", "LAPP_typeManualOperation", and "LAPP_typeManualOperationArea".
For example, In "NutRunner1" DB, create the tags as shown:

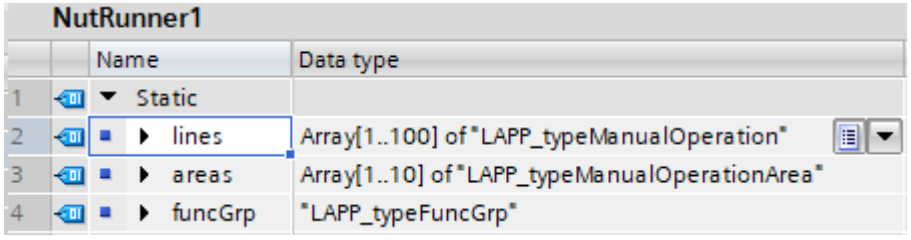


Figure 2-18 NutRunner1 data block

Note

Ensure to start array value from 1.

3. In "NutRunner1.lines", define the number of lines required for this function group by modifying the array value. For more information, refer to Examples (Page 44).

Note

OEM can decide the number of lines required based on the type of the S7 1500 PLC.

4. In "NutRunner1.funcGrp.funcGrpNum", enter the function group ID. For example, 100 as shown below:

NutRunner1

	Name	Data type	Start value
1	Static		
2	lines	Array[1..168] of "LAPP_typeManualOperation"	
3	areas	Array[1..16] of "LAPP_typeManualOperationArea"	
4	funcGrp	"LAPP_typeFuncGrp"	
5	funcGrpNum	UInt	100
6	actvArea	UInt	0

Figure 2-19 FunctiongroupID

5. In "NutRunner1.areas", define the number of areas required for this function group by modifying the array value. For more information, refer to Examples (Page 44).
6. In "NutRunner1.areas[x].displayedLines", enter the line numbers to be displayed as shown for each area:
   x - line number

NutRunner1

	Name	Data type	Start value
1	Static		
2	lines	Array[1..168] of "LAPP_typeManualOperation"	
3	areas	Array[1..16] of "LAPP_typeManualOperationArea"	
4	areas[1]	"LAPP_typeManualOperationArea"	
5	areaNum	UInt	101
6	displayedLines	String	'1-10,3,1;'
7	hmiPage	"LAPP_typePageDetails"	
8	areas[2]	"LAPP_typeManualOperationArea"	
9	areaNum	UInt	102
10	displayedLines	String	'18-60,102-78,44;'
11	hmiPage	"LAPP_typePageDetails"	
12	areas[3]	"LAPP_typeManualOperationArea"	

Figure 2-20 NutRunner1 data block

Note

- Ensure to enter the value within single quotation mark and end it with semicolon ('...;').
- You can enter multiple line numbers by segregating them using comma (1,10)
- You can enter the range of line numbers by using hyphen (1 - 10).
- You can enter the lines to be displayed in any order as mentioned below:
  - incremental range of line values (1 to 10)
  - decremental range of line values (102-78)
  - single line number (1)
  - repeated line number ('1-10, 3, 1;')
- Ensure to follow the rules to enter the line numbers for the block to work as intended. Else the PLC can go into error mode.

7. From "Program blocks", navigate to "SINUMERIK ONE PLC blocks" and drag the "LAPP_manualOperationMain" block to the cyclic block or main block. Interface the input and output parameters.

Note

You need call this mandatory block only once in cyclic block/ main block.

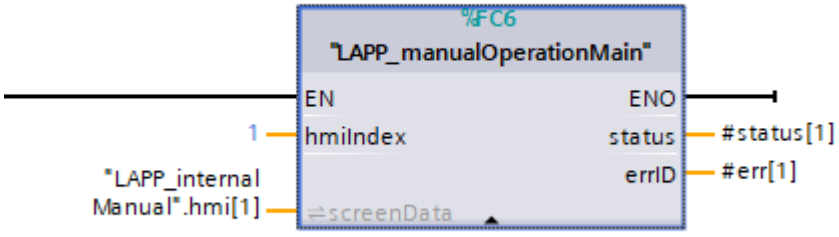


Figure 2-21 LAPP_manualOperationMain

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
screenData	InOut	LAPP_typeFuncGrpH-miDisp	Interface the HMI tag from "LAPP_internal-Manual.hmi[y]". **Note**: "LAPP_internalManual" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information, refer to Error codes (Page 47).

y- indicates HMI device number.

8. From "Program blocks", navigate to "SINUMERIK ONE PLC blocks" and drag the "LAPP_funcGrpConfig" block to the cyclic block or main block.
9. Interface the input and output parameters for the "LAPP_funcGrpConfig" block. For more information, refer to Block interface - LAPP_funcGrpConfig (Page 37).

10. From “Program blocks”, navigate to “SINUMERIK ONE PLC blocks” and drag the "LAPP_manualOperation" block to the cyclic block or main block.

Note

- Refer to the subsequent topics to know more about the types of Manual Operation blocks in this section.
- It is mandatory to configure the manual operation lines using the types of Manual Operation blocks to view the line in runtime.

11. Interface the input and output parameter to the added block interface. For more information, refer to Block interface - LAPP_manualOperation (Page 38).
12. Configure the text list and screen in WinCC Unified. For more information, refer to Configuring LAHP_ManualOperation (MCP/Touch) (Page 109).

2.7.1.3 Block interface - LAPP_funcGrpConfig

Block interface

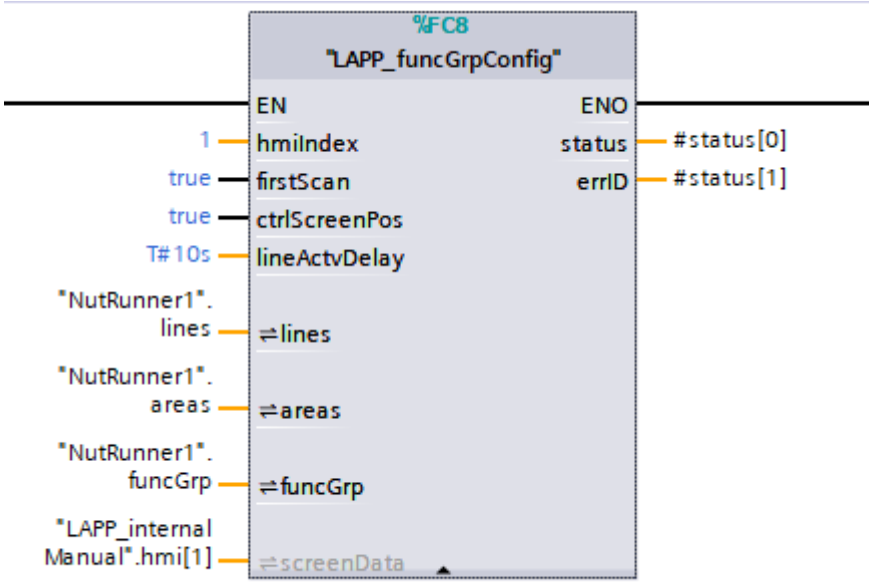


Figure 2-22 LAPP_funcGrpConfig

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
firstScan	Input	Bool	Startup bit.

Name	Declaration	Type	Description
ctrlScreenPos	Input	Bool	Determines the position of the control screen for all the lines in the function group: • 0 - displays on top of the line • 1 - displays below the line
lineActvDelay	Input	Time	Set the line active delay for the function group.
lines	InOut	Array of LAPP_typeManualOperation	Interface the total number of lines in the function group.
areas	InOut	Array of LAPP_typeManualOperationArea	Interface the total number of areas in the function group.
funcGrp	InOut	LAPP_typeFuncGrp	Interface the function group ID.
screenData	InOut	LAPP_typeFuncGrpHmiDisp	Interface the HMI tag from "LAPP_internalManual.hmi[y]". **Note**: "LAPP_internalManual" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 47).

y- indicates HMI device number.

Note

Ensure to interface the same HMI device number at "hmiIndex" and "screenData" value.

2.7.1.4 Block interface - LAPP_manualOperation

"LAPP_ManualOperation" block is used to configure Touch Operated and MCP mode for Manual Operation.

In case of touch operated mode, you can move the device along the left (negative) and right (positive) direction of the axes using buttons.

The actual data comes from field and the touch operation signals are sent as feedback to PLC. You can also edit the text on these buttons. For more information refer to Editing text (Page 109).

In case of MCP operated mode. The actual data goes as output field to WinCC and the MCP signals comes as input to HMI.

Block details

This block is used for data handling and configurations.

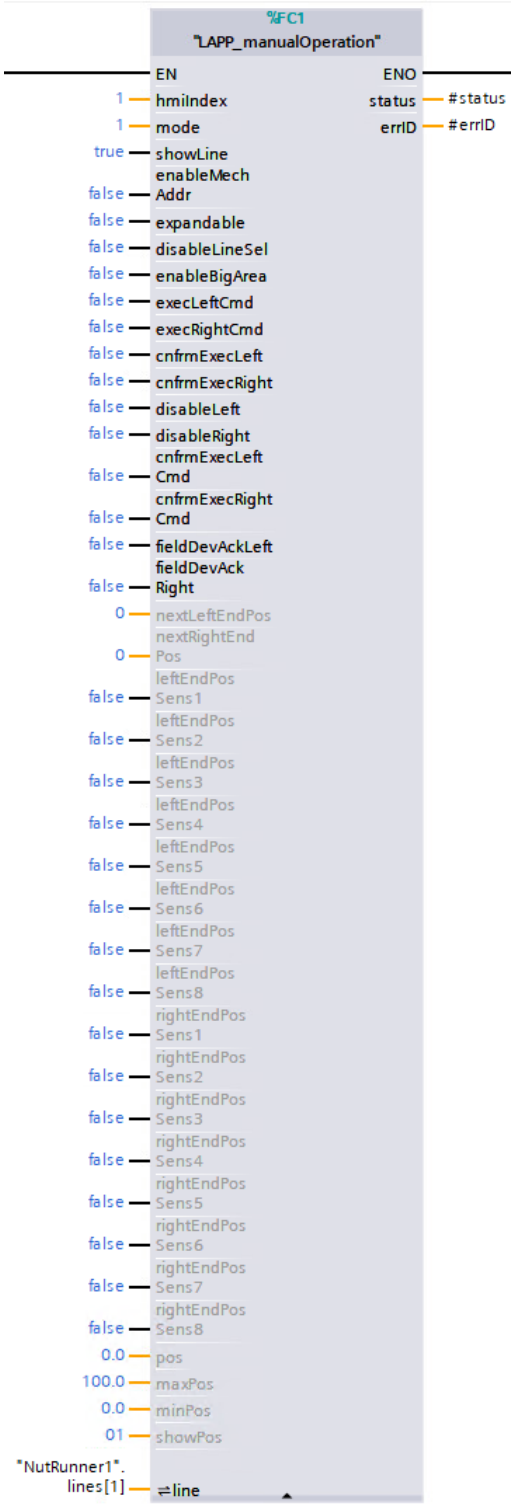


Figure 2-23 LAPP_manualOperation

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
mode	Input	USInt	Determines the operating mode of the line configured: • 1 - Touch operated mode • 2 - MCP operated mode
showLine	Input	Bool	Determines the line visibility in runtime: • 0 - Visible • 1 - Invisible For more information, refer to Example (Page 45). **Note**: The "showLine" parameter works only for the lines configured in screen.
enableMechAddr	Input	Bool	Enables mechanical addressing mode on the screen: • 0 - Disable • 1 - Enable **Note**: The mechanical addressing mode is optional. If enabled, then OEM needs to configure the mechanical address also.
expandable	Input	Bool	Determines the visibility of "▶" icon on the line to view the expand area: • 0 - Invisible • 1 - Visible
disableLineSel	Input	Bool	Determines the line selection possibility to operate the commands: • 0 - Enables line selection • 1 - Disables line selection **Note**: If line selection is disabled, then "enableBigArea" is also disabled.
enableBigArea	Input	Bool	Enables the operating area of the line to whole screen for 5 seconds: • 0 - Disable • 1 - Enable For more information, refer to Example (Page 46).
execLeftCmd	Input	Bool	Status of left command execution. **Note**: Only available for MCP mode. Set the value to "false" for touch operation mode.
execRightCmd	Input	Bool	Status of right command execution. **Note**: Only available for MCP mode. Set the value to "false" for touch operation mode.
cnfrmExecLeft	Input	Bool	Confirms executability for the left button: • 0 - not executable • 1 - executable

Name	Declaration	Type	Description
cnfrmExecRight	Input	Bool	Confirms executability for the right button: • 0 - not executable • 1 - executable
disableLeft	Input	Bool	Disables left button. For example: Enable the button when the "cnfrmExecLeft" is executable.
disableRight	Input	Bool	Disables right button. For example: Enable the button when the "cnfrmExecRight" is executable.
cnfrmExecLeftCmd	Input	Bool	Confirms the execution of "execLeftCmd": • 0 - not confirmed • 1 - confirmed
cnfrmExecRightCmd	Input	Bool	Confirms the execution of "execRightCmd": • 0 - not confirmed • 1 - confirmed
fieldDevAckLeft	Input	Bool	Acknowledgement from field device for left command: • 0 - not acknowledged • 1 - acknowledged
fieldDevAckRight	Input	Bool	Acknowledgement from field device for right command: • 0 - not acknowledged • 1 - acknowledged
nextLeftEndPos	Input	USInt	Configure the next left end position to be reached.
nextRightEndPos	Input	USInt	Configure the next right end position to be reached.
leftEndPosSens1	Input	Bool	Sensor input for left position 1 (field input) • 0 - not sensed • 1 - sensed
leftEndPosSens2	Input	Bool	Sensor input for left position 2 (field input): • 0 - not sensed • 1 - sensed
leftEndPosSens3	Input	Bool	Sensor input for left position 3 (field input): • 0 - not sensed • 1 - sensed
leftEndPosSens4	Input	Bool	Sensor input for left position 4 (field input): • 0 - not sensed • 1 - sensed
leftEndPosSens5	Input	Bool	Sensor input for left position 5 (field input) • 0 - not sensed • 1 - sensed
leftEndPosSens6	Input	Bool	Sensor input for left position 6 (field input) • 0 - not sensed • 1 - sensed

Name	Declaration	Type	Description
leftEndPosSens7	Input	Bool	Sensor input for left position 7 (field input): • 0 - not sensed • 1 - sensed
leftEndPosSens8	Input	Bool	Sensor input for left position 8 (field input) • 0 - not sensed • 1 - sensed
rightEndPosSens1	Input	Bool	Sensor input for right position 1 (field input) • 0 - not sensed • 1 - sensed
rightEndPosSens2	Input	Bool	Sensor input for right position 2 (field input) • 0 - not sensed • 1 - sensed
rightEndPosSens3	Input	Bool	Sensor input for right position 3 (field input) • 0 - not sensed • 1 - sensed
rightEndPosSens4	Input	Bool	Sensor input for right position 4 (field input): • 0 - not sensed • 1 - sensed
rightEndPosSens5	Input	Bool	Sensor input for right position 5 (field input): • 0 - not sensed • 1 - sensed
rightEndPosSens6	Input	Bool	Sensor input for right position 6 (field input): • 0 - not sensed • 1 - sensed
rightEndPosSens7	Input	Bool	Sensor input for right position 7 (field input): • 0 - not sensed • 1 - sensed
rightEndPosSens8	Input	Bool	Sensor input for right position 8 (field input): • 0 - not sensed • 1 - sensed
pos	Input	Real	Actual position of the axis.
maxPos	Input	Real	maximum limit of the position along the line.
minPos	Input	Real	minimum limit of the position along the line.
showPos	Input	USInt	Determines the position of the setup line visible: • 00 – both, graphical representation, and numeric value is invisible • 01 – both, graphical representation, and numeric value is visible • 02 – only the numeric value is visible You can set the value to 00 and use the line for Powerup condition.

Name	Declaration	Type	Description
line	InOut	LAPP_typeManua-lOperation	Assign manual operation line data. For example, "NutRunner1.lines[x]".
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information, refer to Error codes (Page 47).

x - line number

2.7.1.5 Data type

LAPP_typeFunGrp

Table 2-5 Structure of data type LAPP_typeFunGrp

Name	Data type	Description
funcGrpNum	UInt	Displays the specific ID for the function group.
actvArea	UInt	Displays the active area.

LAPP_typeManualOperationArea

Table 2-6 Structure of data type LAPP_typeManualOperationArea

Name	Data type	Description
areaNum	UInt	Displays the area number
displayedLines	String	The list of lines to be displayed in the area

LAPP_typeManualOperation

Refer LAPP_typeManualOperation (Page 38) for more information on the data type.

2.7.1.6 Examples

Configuring number of lines

Example

Scenario 1
To configure maximum lines to 100, then enter the value at "NutRunner1.lines" as shown:

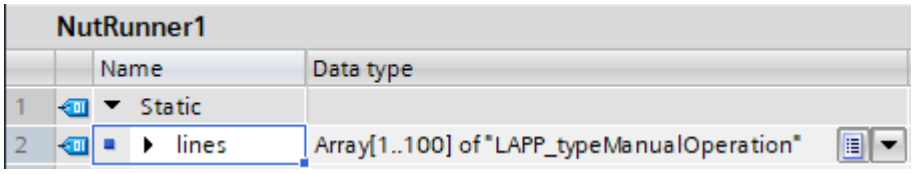


Figure 2-24 NutRunner1 data block

Scenario 2

To configure maximum lines to 168, then enter the value at "NutRunner1.lines" as shown:

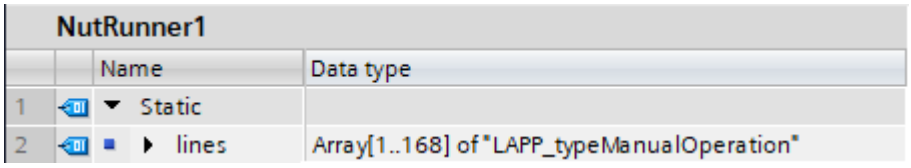


Figure 2-25 NutRunner1 data block

Configuring number of areas

Example

Scenario 1
To configure maximum areas to 10, then enter the value at "NutRunner1.areas" as shown:

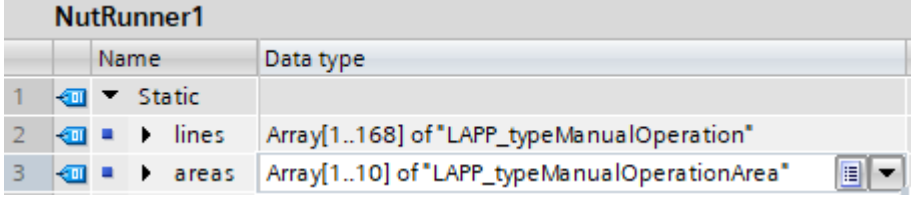


Figure 2-26 NutRunner1 data block

Scenario 2

To configure maximum lines to 16, then enter the value at "NutRunner1.areas" as shown:

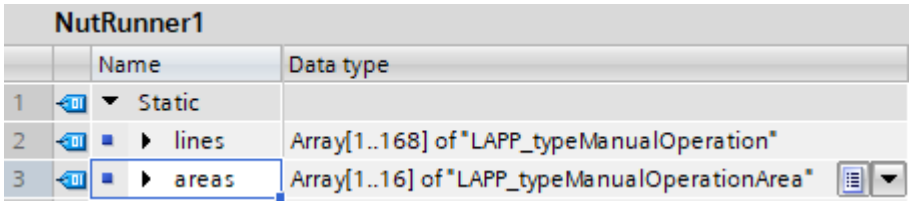


Figure 2-27 NutRunner1 data block

Configuring showline

Example to Show line

To show the line in manual operation screen, set the parameter values as shown below:

Input parameter	Value	Description
"line"	"Line2"	In runtime, line 2 will be shown.
"showLine"	1	If value is 1, then the line is visible during runtime.

Figure 2-28 Line visible

Example to hide line

To hide the line in manual operation screen, set the parameter values as shown below:

Input parameter	Value	Description
"line"	"Line2"	In runtime, Line2 will be shown.
"showLine"	0	If value is 0, then the line is visible during runtime.

Figure 2-29 Line2 invisible

Note

You can hide more than one line.

Configuring big area

Example to enable big area

To enable the line operating area to full screen in manual operation screen, set the parameter values as shown below:

Input parameter	Value	Description
"line"	"Line2"	In runtime, line 2 will be shown.
"disableLineSel"	0	Line selection is enabled.
"enableBigArea"	1	If value is 1, then the line operating area to full screen.

Figure 2-30 "enableBigArea" enabled

Example to disable big area

To disable the full screen operating area of the line in manual operation screen, set the parameter values as shown below:

Input parameter	Value	Description
"line"	"Line2"	In runtime, line 2 will be shown.
"disableLineSel"	0	Line selection is enabled.
"enableBigArea"	0	If value is 0, then the line operating area is restricted to the button area.

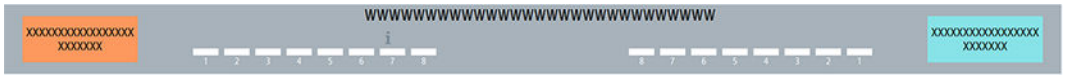


Figure 2-31 "enableBigArea" disabled

2.7.1.7 Error codes

List

Following are the error codes for "LAPP_funcGrpConfig".

Table 2-7 Error codes for function group

Error code	Description
16#8400	No connection to HMI.
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	Input at displayed lines is empty.
16#8203	The specified range of lines are not in the required format.
16#8204	Incorrect line numbers entered in the input parameters of the displayed line.
16#8205	The number of active lines exceeds the maximum limit (128).

Following are the error codes for "LAPP_manualOperation".

Table 2-8 Error codes for MCP and Touch operated modes

Error code	Description
16#8400	No connection to HMI.
16#8200	Faulty license.
16#8201	Invalid HMI index.

2.7.2 Manual Operation as faceplate

2.7.2.1 Manual Operation as faceplate

You can create manual operation lines using faceplates. You can use "LAPP_manualOperationFaceplate" and add up to 6 lines. To add faceplate on multiple screens, add a trigger to execute the block.
For example,

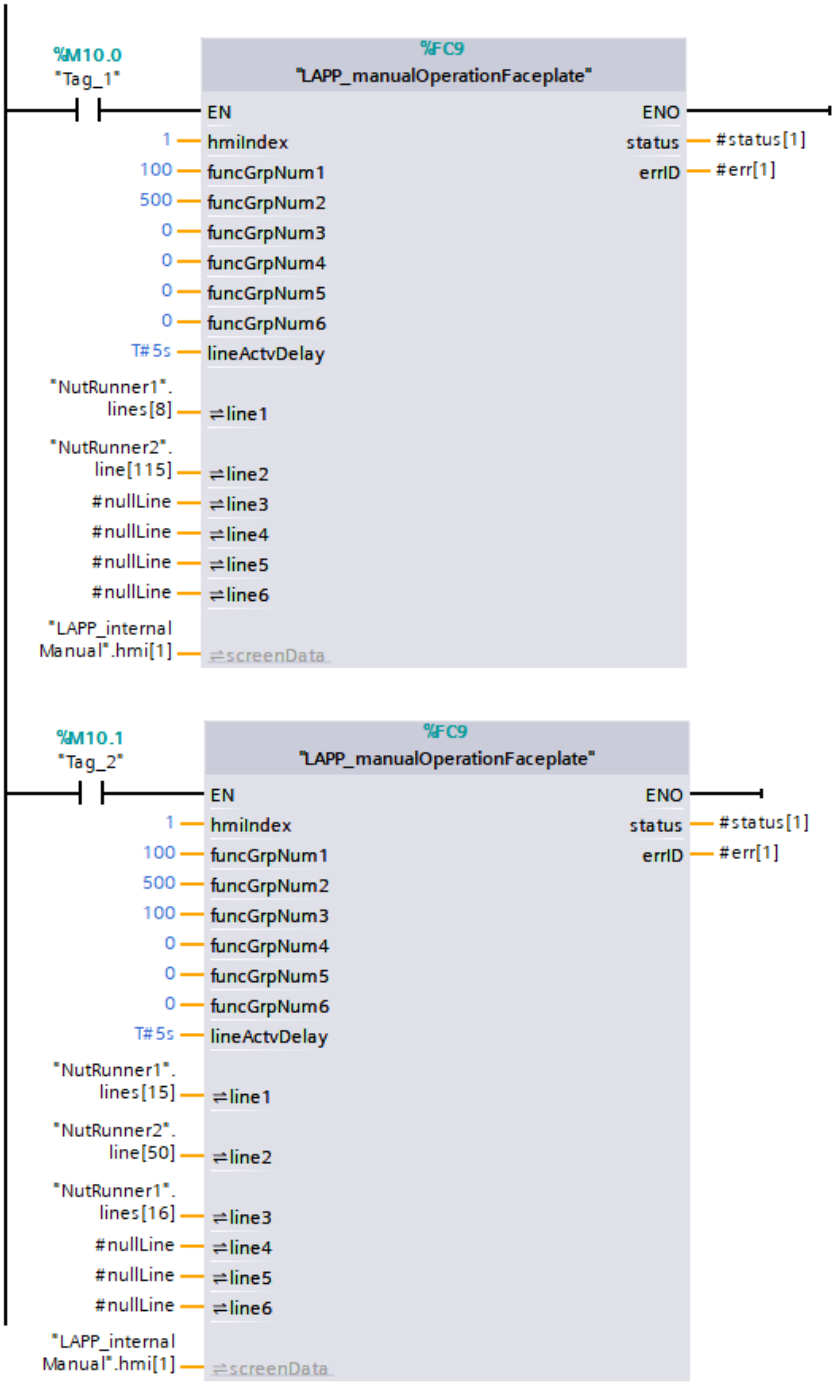


Figure 2-32 LAPP_manualOpeartionFaceplate block

Procedure

1. From "Program blocks", navigate to "SINUMERIK ONE PLC blocks" and drag the "LAHP_ManualOperationFacePlate".
2. Interface the input and output parameters. For more information on block interface, refer to Block interface - LAPP_manualOperationFaceplate (Page 51).

Note

You can use indirect addressing for the faceplate to minimize the number of power tags in HMI. For more information, refer to Addressing tags indirectly on TIA Portal online help.

2.7.2.2 Block interface - LAPP_manualOperationFaceplate

Block interface

This block is used for data handling and configurations.

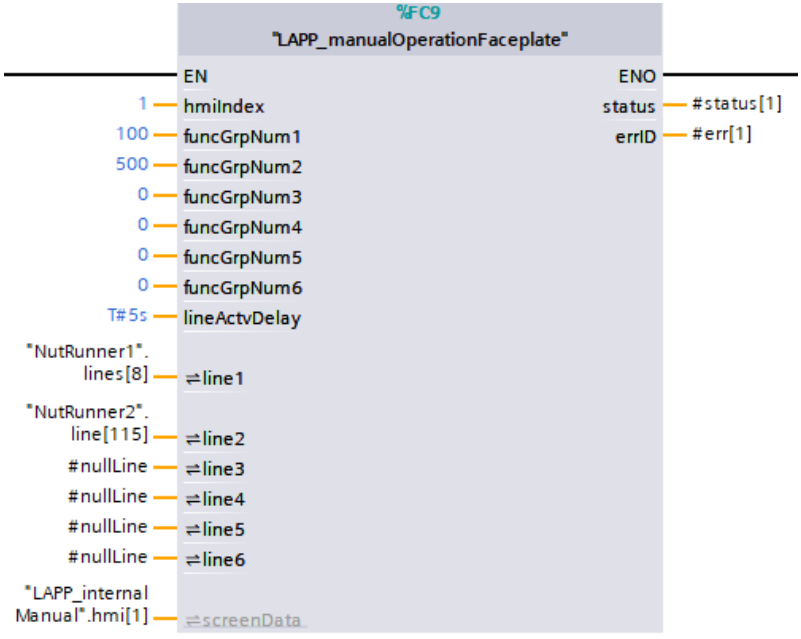


Figure 2-33 LAPP_manualOperationFaceplate

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
funcGrpNum1	Input	UInt	Interface the function group number of the line you want to interface at "line1" parameter.
funcGrpNum2	Input	UInt	Interface the function group number of the line you want to interface at "line2" parameter.

Name	Declaration	Type	Description
funcGrpNum3	Input	UInt	Interface the function group number of the line you want to interface at "line3" parameter.
funcGrpNum4	Input	UInt	Interface the function group number of the line you want to interface at "line4" parameter.
funcGrpNum5	Input	UInt	Interface the function group number of the line you want to interface at "line5" parameter.
funcGrpNum6	Input	UInt	Interface the function group number of the line you want to interface at "line6" parameter.
lineActvDelay	Input	Time	Set the line active delay for the function group.
line1	InOut	LAPP_typeManualOperation	Represents the number of manual operation lines configured. For example, Assign manual operation lines from "NutRunner1.lines[x]". **Note**: If the line is not configured, create a tag for null line with data type "LAPP_typeManualOperation" and interface.
line2	InOut	LAPP_typeManualOperation	Represents the number of manual operation lines configured. For example, Assign manual operation lines from "NutRunner1.lines[x]". **Note**: If the line is not configured, create a tag for null line with data type "LAPP_typeManualOperation" and interface.
line3	InOut	LAPP_typeManualOperation	Represents the number of manual operation lines configured. For example, Assign manual operation lines from "NutRunner1.lines[x]". **Note**: If the line is not configured, create a tag for null line with data type "LAPP_typeManualOperation" and interface.
line4	InOut	LAPP_typeManualOperation	Represents the number of manual operation lines configured. For example, Assign manual operation lines from "NutRunner1.lines[x]". **Note**: If the line is not configured, create a tag for null line with data type "LAPP_typeManualOperation" and interface.
line5	InOut	LAPP_typeManualOperation	Represents the number of manual operation lines configured. For example, Assign manual operation lines from "NutRunner1.lines[x]". **Note**: If the line is not configured, create a tag for null line with data type "LAPP_typeManualOperation" and interface.
line6	InOut	LAPP_typeManualOperation	Represents the number of manual operation lines configured. For example, Assign manual operation lines from "NutRunner1.lines[x]". **Note**: If the line is not configured, create a tag for null line with data type "LAPP_typeManualOperation" and interface.
screenData	InOut	LAPP_typeFuncGrpHmiDisp	Interface the HMI tag from "LAPP_internalManual.hmi[y]". **Note**: "LAPP_internalManual" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).

Name	Declaration	Type	Description
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information, refer to Error codes (Page 53).

x- indicates line number.

y- indicates HMI device number.

Note

- All the actual data must be programmed by OEM.
- Ensure to configure null line if the line is not configured for the block, else the block will not work as intended.
- Ensure to interface the same HMI device number at "hmiIndex" and "screenData" value.

2.7.2.3 Error codes

List

Following are the error codes for "LAPP_manualOperationFaceplate".

Error code	Description
16#8400	No connection to HMI.
16#8200	Faulty license.
16#8201	Invalid HMI index.

2.8 Tool life monitor

2.8.1 Overview

Tool life monitor is used to indicate the life expectancy of the tools used in machining. It is a database provided to monitor tool used in production. You can also view the status of the tool - if the tool is in active state, if there are any alarms on the tool and/or if the life of the tool has expired.

The LAHP_toolLifeMonitor runtime screen is shown below:

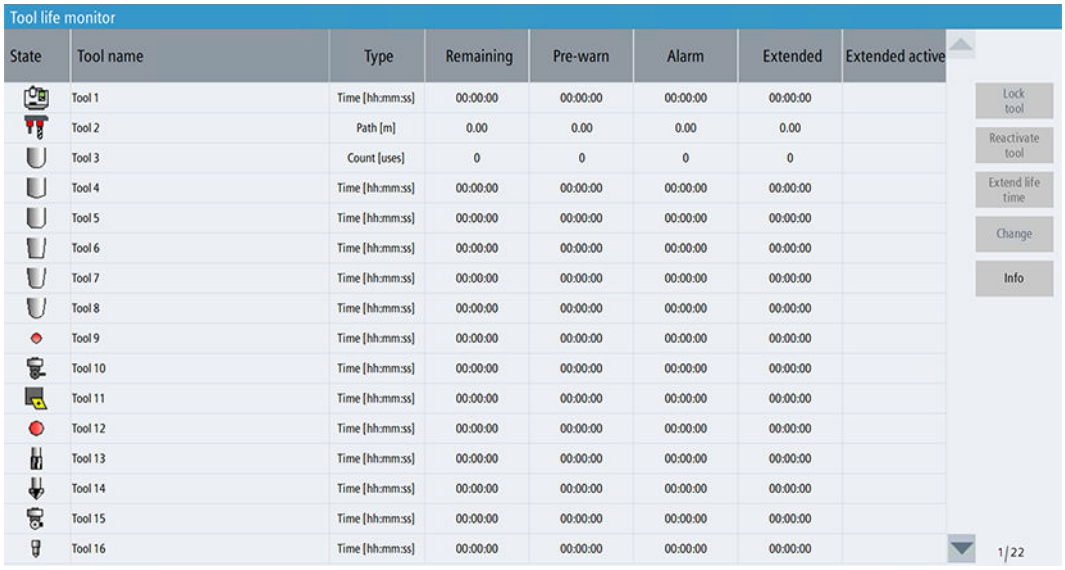


Figure 2-34 Tool life monitor screen

Tool life monitor is available in three modes.

- Time - The tool can be monitored based the duration for which it is operated.
- Length - The tool can be monitored based on the length of the tool.
- Count - The tool can be monitored based the number of times it is used, or the number pieces it manufactures.

The LAPP_typeToolLifeMonitor data type is used to configure the tool life blocks.

You need to configure the following for tool life screen to work as intended:

- Block configuration in step 7
  - "LAPP_toolLifeMain" (Page 55) - Configure this block once in the cyclic block or main block and interface the number of tools and the data to be fetched to display on the screen.
  - "LAPP_toolLifeMonitor (Page 57)" - Configure this block for each tool and interface the required input and output parameters.

  For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).
- Text and graphic list configuration (Page 131) to modify the displayed tool name and the tool information in WinCC Unified.

2.8.2 Configuring number of tools

Procedure

1. Create new data block. For example: Demo
2. Create a tag with array[1.....X] of "LAPP_typeToolLifeMonitor" as data type.
   X - enter the maximum tools in the array value
3. Interface the tag to "tools" parameter of the "LAPP_toolLifeMain" block.

For example,
Scenario 1

To configure maximum tools to 30, create a tag with array value as [1..30] as shown:

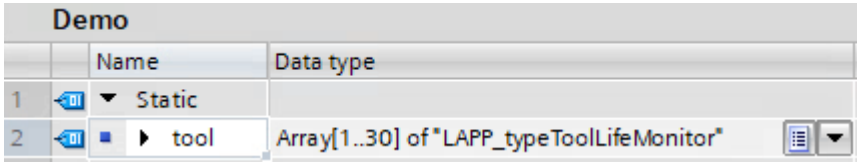


Figure 2-35 Demo DB

Scenario 2

To configure maximum tools to 300, create a tag with array value as [1..300] as shown:

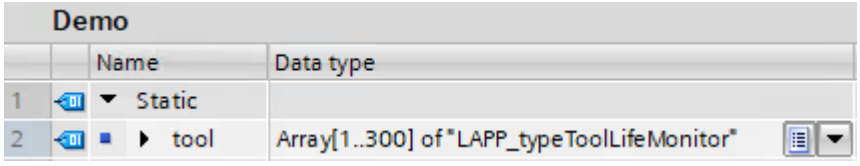


Figure 2-36 Demo DB

Result

You have configured the number of tools for monitoring the tool life.

2.8.3 Block interface - LAPP_toolLifeMain

The "LAPP_toolLifeMain" block is used to configure number of tools used.

Note

You need call this mandatory block only once in cyclic block/ main block.

Block interface

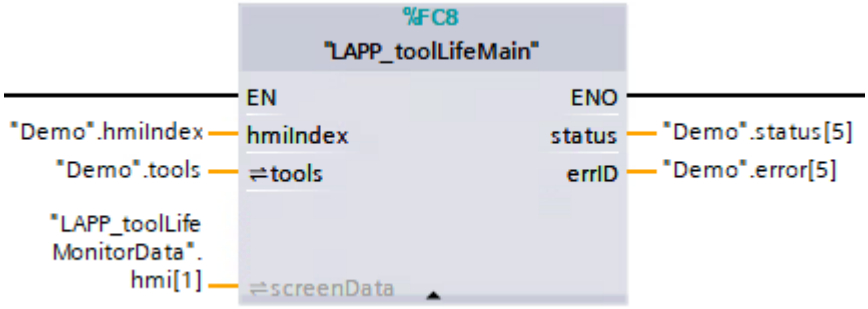


Figure 2-37 LAPP_toolLifeMain

Parameters

Table 2-9 Parameters for "LAPP_toolLifeMain"

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note:** HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
tools	InOut	Array of LAPP_typeToolLifeMonitor	Interface the PLC tag with "LAPP_toolLifeMain" data type. You are interfacing number of tools defined for use.
screenData	InOut	LAPP_typeHmiToolScreenData	Interface the HMI tag from "LAPP_toolLifeMonitorData.hmi[y]". **Note**: "LAPP_toolLifeMonitorData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 60).

y- indicates HMI device number.

Note

To view the data in multiple HMI, you need to configure multiple blocks with same configuration and different "hmiIndex" value.

2.8.4 Block interface - LAPP_toolLifeMonitor

Based on the mode selection, interface the parameters of the "LAPP_toolLifeMonitor" block.

You need to configure this block for each tool.

For example, if you need to configure 10 tools, use this block ten times to configure 10 tools. Each block is configured with one tool data.

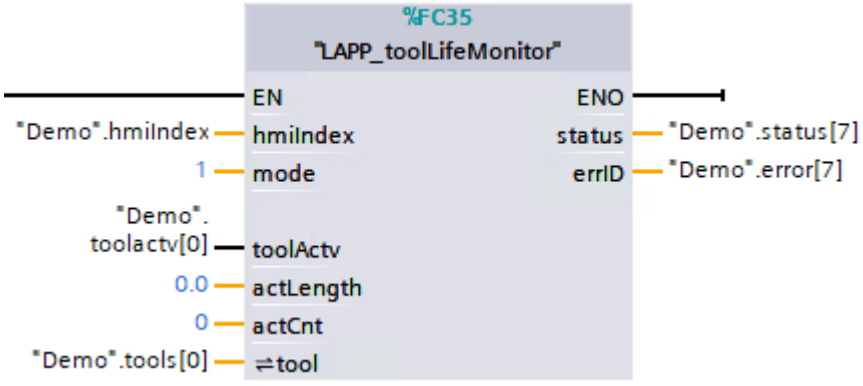


Figure 2-38 LAPP_toolLifeMonitor time mode

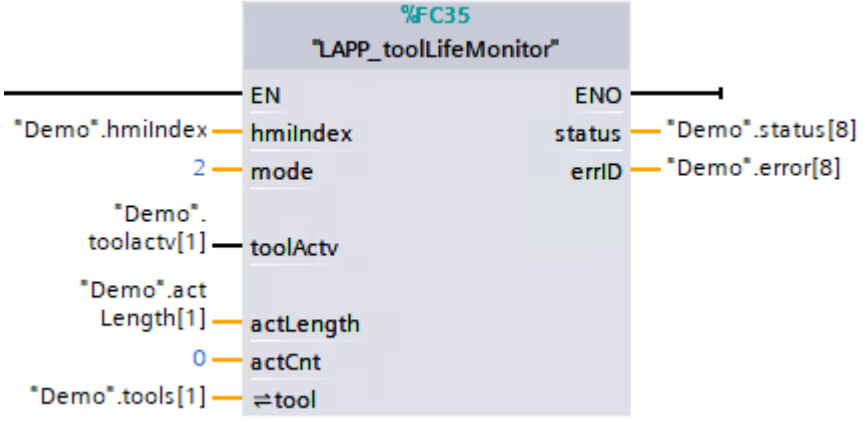


Figure 2-39 LAPP_toolLifeMonitor length mode

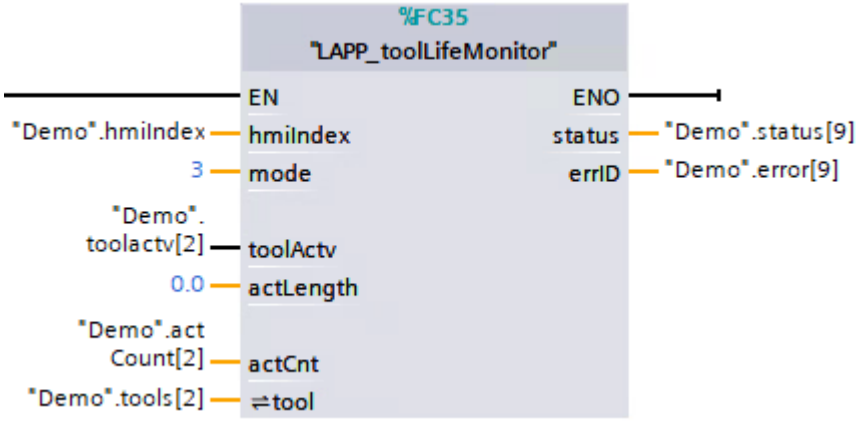


Figure 2-40 LAPP_toolLifeMonitor count mode

⚠ WARNING

Dynamization of the modes

Do not change the mode in runtime. If you change, you need to reconfigure the tool from UI and reactivate the tool from PLC.

Parameters

Table 2-10 Parameters for "LAPP_toolLifeMonitor"

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
mode	Input	USInt	Operating mode of the tool: • 1 - Time • 2 - Length • 3 - Count
toolActv	Input	BOOL	Enable to activate and start the timer of the tool: • 0 - inactive • 1 - active
actLength	Input	Real	Actual length of the tool (to be entered if the tool is in length mode)
actCnt	Input	DInt	Actual count of the tool (to be entered if the tool is in count mode)
tool	InOut	LAPP_typeToolLifeMonitor	Tool life structure

Name	Declaration	Type	Description
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 60).

2.8.5 Data type

LAPP_typeToolLifeMonitor

This data type is used to configure the details of the tool in use.

Table 2-11 Structure of data type LAPP_typeToolLifeMonitor

Name	Type	Description
mode	USInt	Operating mode of the tool: • 1 - Time • 2 - Length • 3 - Count
toolIndex	UInt	Tool index number
state	USInt	State of the tool: • 0 - Ok • 1 - Prewarn • 2 - End of life • 3 - Locked
extndLifeActv	Bool	Extended life: • 0 - Not active • 1 - Active
edgeNum	UInt	Edge tool number
sisterToolNum	UInt	sister tool number
settings	LAPP_typeToolLifeSet-tings	Limits of tool settings

Table 2-12 Structure of data type LAPP_typeToolLifeSettings

Name	Type	Description
lifeSpan	DWord	Life of the tool
remaining	DWord	Remaining measure of the tool
preWarn	DWord	Pre-warning limit to measure the tool operation

Name	Type	Description
alarm	DWord	Alarm limit to measure the tool operation
extndAlarm	DWord	Extended alarm limit to measure the tool operation

2.8.6 Error codes

List

The following are the error codes for LAPP_toolLifeMain block:

Error Code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8400	No connection to HMI.

The following are the error codes for LAPP_toolLifeMonitor block:

Error Code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	Shows the mode of the tool. • 1 - Time • 2 - Length • 3 - Count
16#8203	Alarm time entered from runtime is greater than pre-warn time.
16#8204	The tool life span entered is less than pre-warn time.
16#8205	The pre-warning limit entered is zero or not entered.
16#8206	The alarm limit entered is zero or not entered.
16#8400	No connection to HMI.

2.9 Process Dashboard

2.9.1 Overview

The process dashboard is used to view the quality metrics, check details of the workpiece, current workpiece in production, tool life details, and add notes to next shift operator.

The "LAPP_TypeWidgetPartDetails", "LAPP_TypeWidgetProductionSts", "LAPP_TypeWidgetQualityMetrics", "LAPP_TypeWidgetWpiece" UDT and "LAPP_typeWidgetToolLife" block is used to parametrize the process dashboard screen during runtime.

You need to configure the following for process dashboard screen to work as intended:

- Create tags using the UDT and configure the "LAPP_typeWidgetToolLife" (Page 64) block in step 7.
  For more information on creating a data block to use in "LAHP_ProcessDashboard" screen, refer to Creating data block (Page 61).
  For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).
- HMI tag creation and text list (Page 137) configuration to modify the displayed tool name and the tool information in WinCC Unified.

2.9.2 Creating data block

To configure the process dashboard block, you need to create data block.

Procedure

To create data block for process dashboard:

1. Right-click "Program blocks" and click "Add new block"

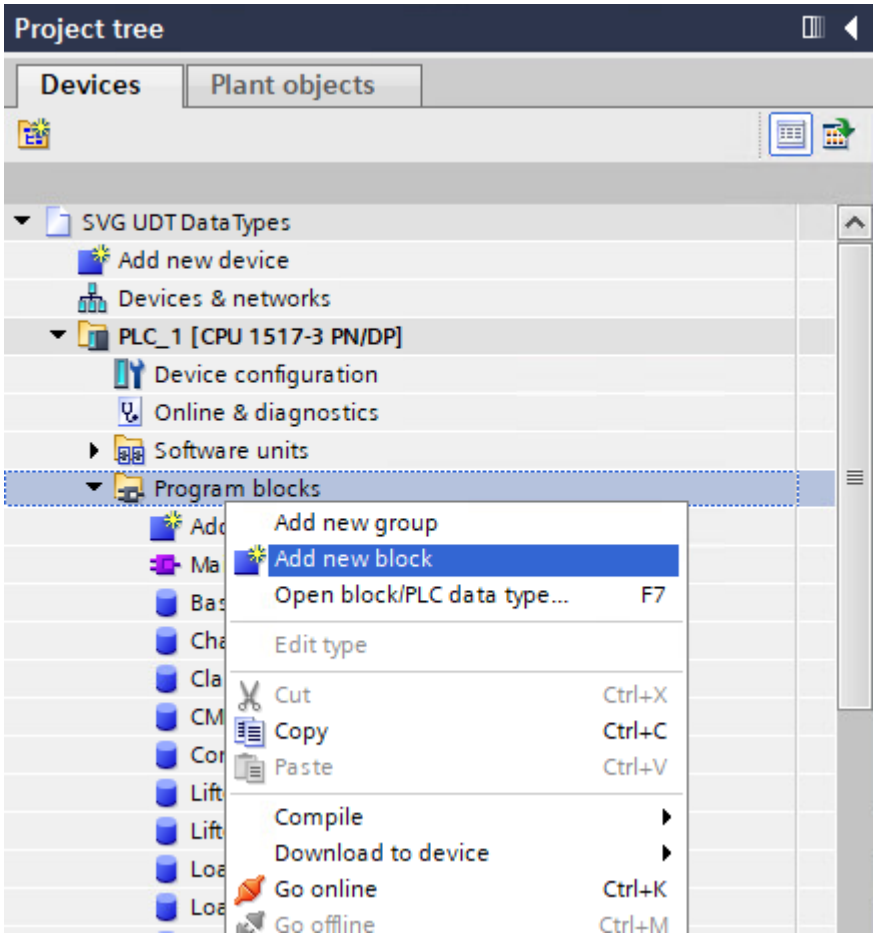


Figure 2-41 Add new block

"Add new block" window opens.

2. Click "Data block".

3. In "Type" field, select "LAPP_typeWidgetPartDetails", "LAPP_typeWidgetProductionSts", "LAPP_typeWidgetQualityMetrics" or "LAPP_typeWidgetWpiece" to configure the part details, production status, quality metrics or the workpiece details respectively.

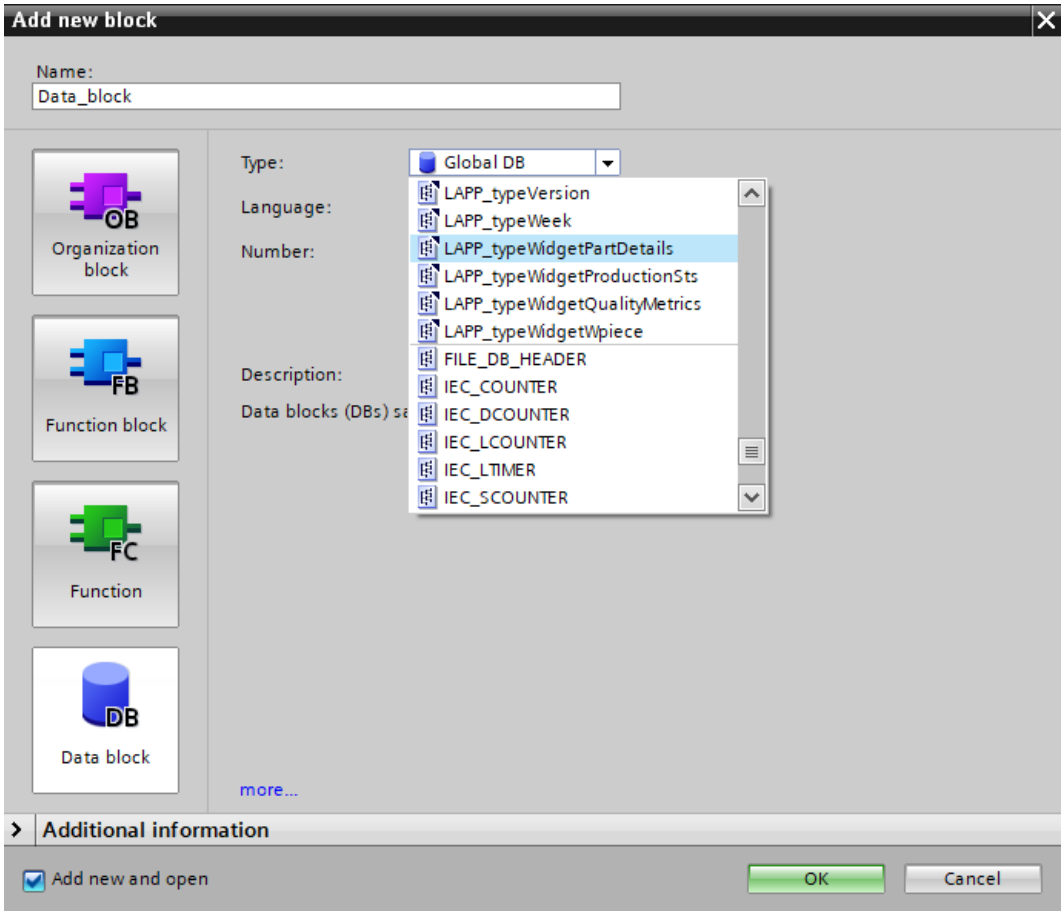


Figure 2-42 Data block creation

4. Click "OK".

Result

Data block for dashboard gets created.

2.9.3 Block interface - LAPP_widgetToolLife

Block interface

This block is used to configure the tool life widget in the LAHP_ProcessDashboard runtime screen.

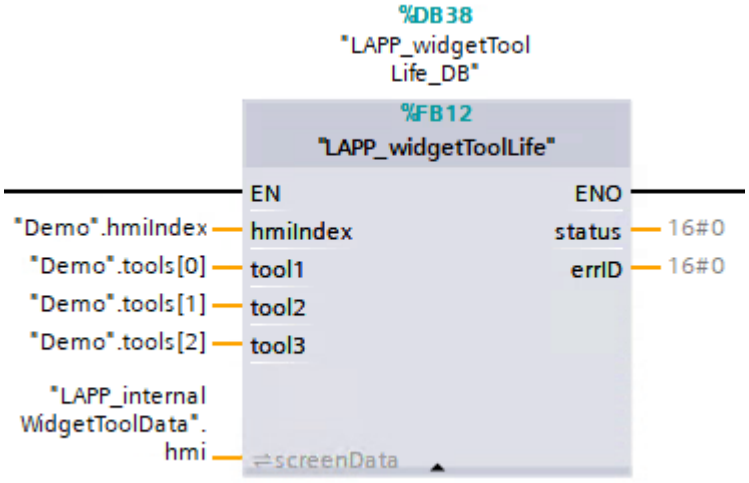


Figure 2-43 LAPP_widgetToolLife

Parameters

Table 2-13 Parameters for "LAPP_widgetToolLife"

Name	Declaration	Type	Description
hmiIndex	Inout	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
tool1	Input	LAPP_typeToolLifeMonitor	Details of tool 1 to be displayed on the dashboard screen
tool2	Input	LAPP_typeToolLifeMonitor	Details of tool 2 to be displayed on the dashboard screen
tool3	Input	LAPP_typeToolLifeMonitor	Details of tool 3 to be displayed on the dashboard screen
screenData	InOut	LAPP_internalWidgetToolData	Interface the HMI tag from "LAPP_internalWidgetToolData.hmi[y]". **Note**: "LAPP_internalWidgetToolData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).

Name	Declaration	Type	Description
status	Output	Word	Displays the status of the block.
errID	Output	Word	Displays the error codes of the block. For more information, refer to Error codes (Page 67).

y- indicates HMI device number.

LAPP_typeToolLifeMonitor

This data type is used to configure the details of the tool in use. For more information on the structure, refer to Data type (Page 59).

2.9.4 Data type

LAPP_typeWidgetWpiece

This data type is used to fetch the details of the workpiece in progress.

Table 2-14 Structure of data type LAPP_typeWidgetWpiece

Name	Data type	Description
currWpieceName	String	Name of the current workpiece in production
currWpiecePartID	String	Part ID of the current workpiece in production
currWpieceProgram	String	Name of the program used for current workpiece
setTime	LTime	Runtime set for the current workpiece
actTime	LTime	Actual machining time of the current workpiece
residualTime	LTime	Residual runtime of the current workpiece

LAPP_typeWidgetPartDetails

This data type is used to fetch the details of the workpiece count.

Table 2-15 Structure of data type LAPP_typeWidgetPartDetails

Name	Data type	Description
actWpieceTarget	UInt	Number of workpieces set for production
actWpieceGoodParts	UInt	Actual number of good parts produced
actWpieceBadParts	UInt	Actual number of bad parts produced
actShiftTarget	UDInt	Number of workpieces set for production in the current shift
actShiftGoodParts	UDInt	Actual number of good parts produced in the current shift

Name	Data type	Description
actShiftBadParts	UDInt	Actual number of bad parts produced in the current shift
dayTarget	UDInt	Number of workpieces set for production in a day
dayGoodParts	UDInt	Actual number of good parts produced in a day
dayBadParts	UDInt	Actual number of bad parts produced in a day

LAPP_typeWidgetQualityMetrics

This data type is used to fetch the details on the quality of the workpiece.

Table 2-16 Structure of data type LAPP_typeWidgetQualityMetrics

Name	Data type	Description
lastWpieceStatus	Bool	Status of the quality of the workpiece checked: • 0 = Not ok • 1 = ok
nextWpieceCheck	UInt	Program the counter to check for the next workpiece quality.

LAPP_typeWidgetProductionSts

This data type is used to fetch the details on the quality of the workpiece.

Table 2-17 Structure of data type LAPP_typeWidgetProductionSts

Name	Data type	Description
status1	Bool	Production status 1
status2	Bool	Production status 2
status3	Bool	Production status 3
status4	Bool	Production status 4
status5	Bool	Production status 5
stacklightCount	USInt	Stacklight count for production status

2.9.5 Error codes

List

Table 2-18 Description of error codes "LAPP_widgetToolLife".

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	The mode of tool entered is incorrect.
16#8203	The tool nuber is zero or invalid.
16#8400	No connection to HMI.

2.10 Hardware diagnostics

2.10.1 SINAMICS

2.10.1.1 Ovierview

The SINAMICS blocks supplies the details for the SINAMICS screens. There are two types of SINAMICS diagnostic blocks:

- SINAMICS diagnostic blocks using EPOS (Page 69): This block is used to read the drive parameters and actual values directly from drive.
  The LAPP_typeSINAMICSConfig data type is used to configure the SINAMICS EPOS blocks. The SINAMICS EPOS runtime screen is shown below.

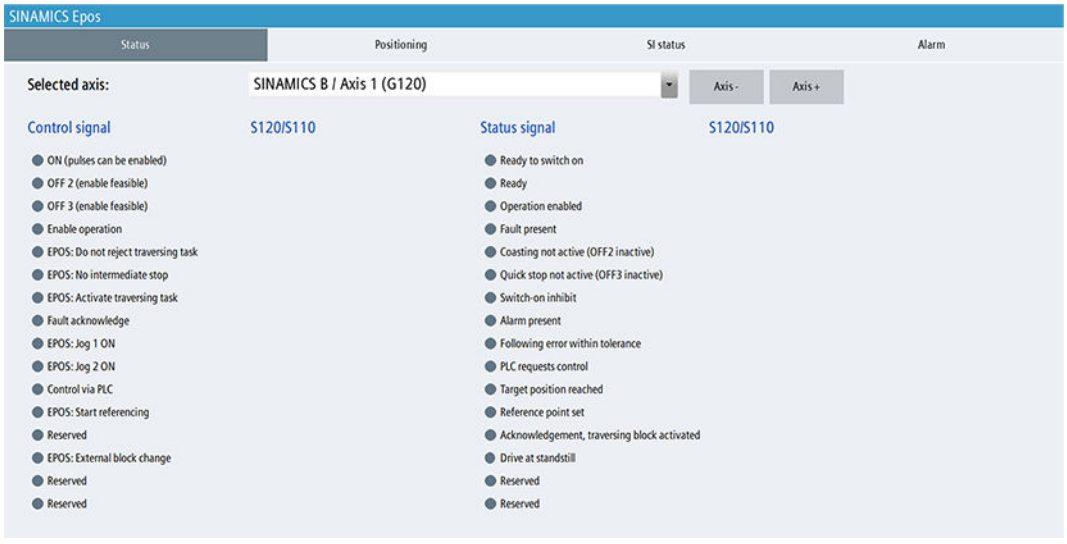


Figure 2-44 SINAMICS EPOS

- SINAMICS diagnostic block using technology objects (Page 71): This block is used to read data of the technology objects. The technology objects that can be used are speed axis, position axis and synchronous axis. The SINAMICS TO runtime screen is shown below.

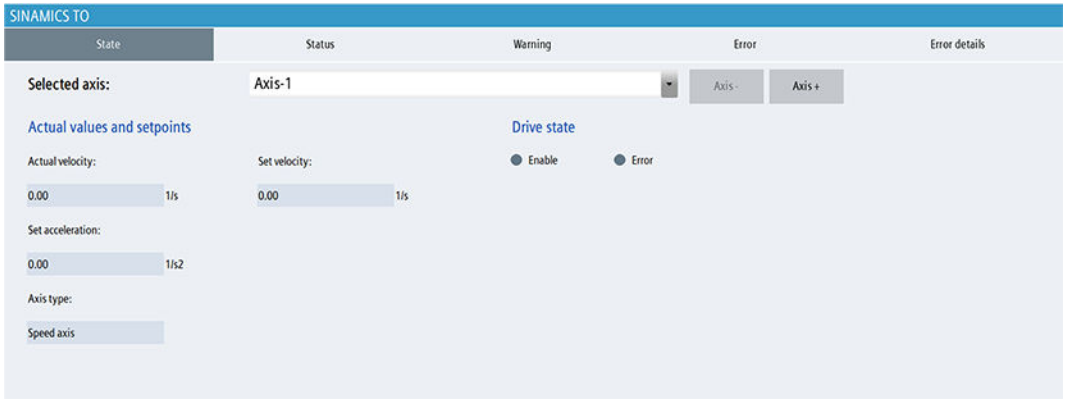


Figure 2-45 SINAMICS TO

The function block must be called once cyclically. The FB call must be enabled via the "driveEnable" parameter.

For more details on configuring the block, refer to Configuring PLC blocks (Page 18).

2.10.1.2 SINAMICS diagnostic blocks using EPOS

Block interface - LAPP_SINAMICSEpos

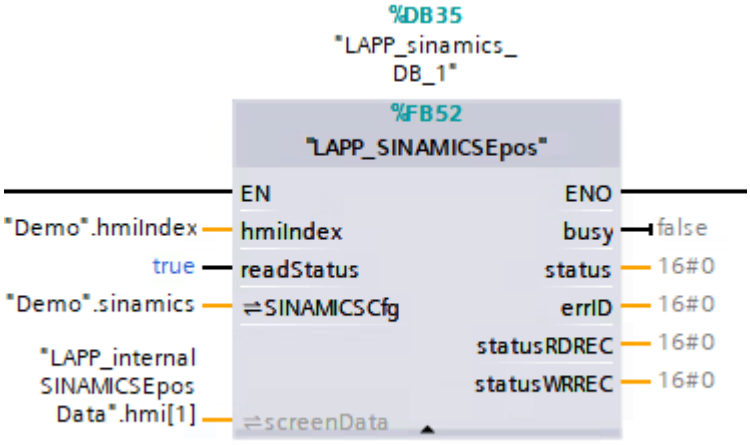


Figure 2-46 LAPP_SINAMICSEpos

Table 2-19 Parameters of the "LAPP_SINAMICSEpos"

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
readStatus	Input	Bool	"TRUE" enables the communication of the block with the drive.
SINAMICSCfg	InOut	Array of LAPP_typeSINAMICS-Config	The drive objects are configured.
screenData	InOut	LAPP_typeHmiSINAMICSEpos	Interface the HMI tag from "LAPP_internalSINAMICSEposData.hmi[y]". Note: "LAPP_internalSINAMICSEposData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer Configuring PLC blocks for different HMI (Page 25).
busy	Output	Bool	"TRUE" communication with the drive.

Name	Declaration	Type	Description
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 70).
statusRDREC	Output	DWord	Status of the RDREC instruction. For more information on instruction, refer to RDREC: Read data record in TIA Portal online help.
statusWRREC	Output	DWord	Status of the WRREC instruction. For more information on instruction, refer to WRREC: Write data record in TIA Portal online help.

y- indicates HMI device number.

Data type

LAPP_typeSINAMICSConfig

Table 2-20 Structure of data type LAPP_typeSINAMICSConfig

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 3 = SINAMICS S210 without position 4 = SINAMICS S210 with position
hwID	HW_IO	Hardware identification of the DP slave, taken from "Devices & networks"
axis	BYTE	Device number from the hardware configuration

Error codes

List

Table 2-21 Description of error codes of "LAPP_SINAMICS"

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	Invalid drive object type.
16#8203	Invalid input array configuration.
16#8204	Block disabled.
16#8600	Status of the write record block.

Error code	Description
16#8601	Status of the read record block.
16#8602	Invalid order reference.
16#8400	No connection to HMI.

2.10.1.3 SINAMICS diagnostic block using technology objects

Block interface - LAPP_SINAMICSTO

Prerequisite

Use the compatible version of technology object blocks.

Note

If older version TO block is used, you get a compilation error.
Use the correct version for the blocks to work as intended.

You need to select the compatible version of technology block in the area highlighted in the below screen.

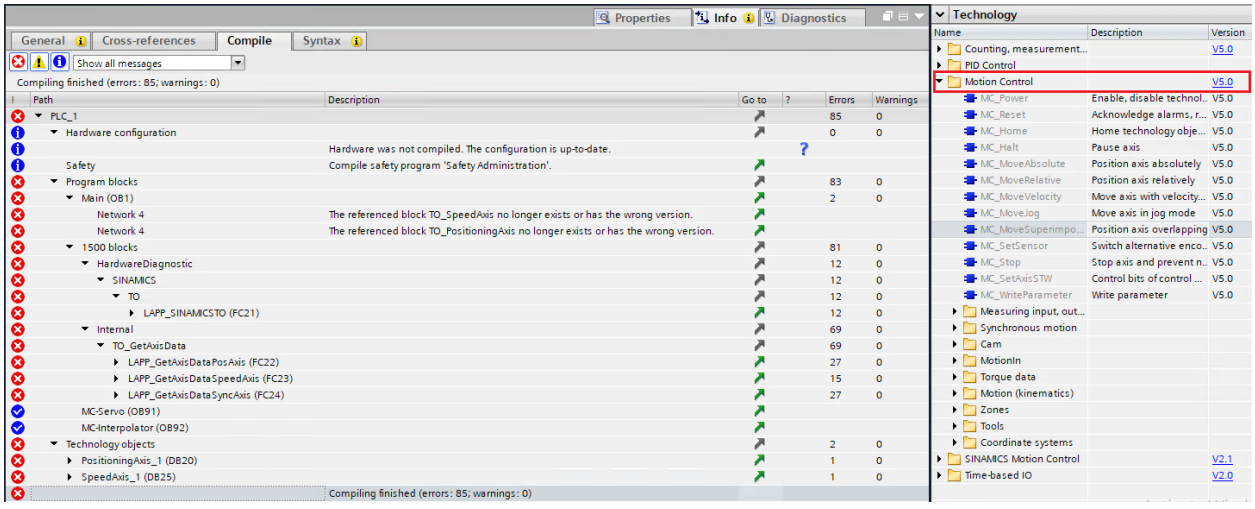


Figure 2-47 Compile error and TO version selection

Block details

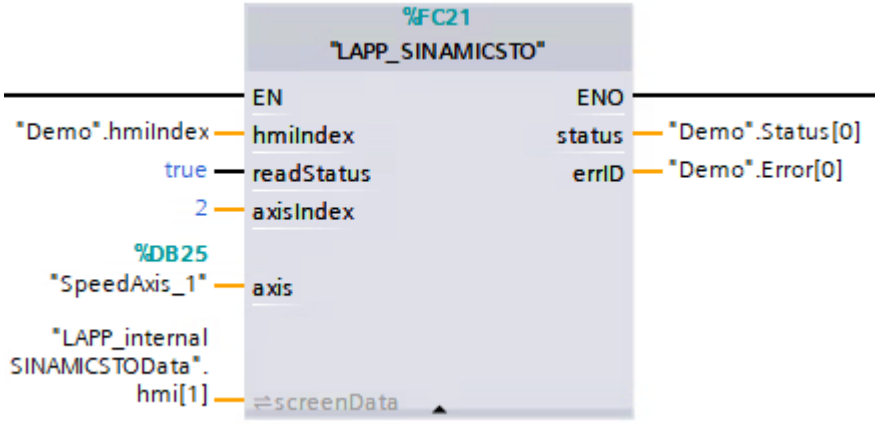


Figure 2-48 LAPP_SINAMICSTO

Parameters

Table 2-22 Parameters of the "LAPP_SINAMICSTO" block

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
readStatus	Input	Bool	"TRUE" enables the communication of the block with the technology object
axisIndex	Input	Int	HMI index for displaying the selected technology object data in the screen
axis	Input	DB_ANY	Interface for the technology object data block. Note: Ensure to interface the correct TO data-block, and verify if the type of axis (speed, position, or synchronous axis) for the entered "axisIndex" in the text list (Page 147) is same.
screenData	InOut	LAPP_typeHmiAxisData	Interface the HMI tag from "LAPP_internalSINAMICSTOData.hmi[y]". **Note**: "LAPP_internalSINAMICSTOData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).

Name	Declaration	Type	Description
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 73).

y- indicates HMI device number.

Error codes

List

Table 2-23 Description of error codes of "LAPP_SINAMICSTO"

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	Block is disabled.
16#8203	Axis is not selected.
16#8204	The selected axis type is not supported.
16#8400	No connection to HMI.

2.10.2 LAPP_motorStarter

2.10.2.1 Overview

The motor starter blocks supply the WinCC screens for the motor starter diagnostics. You must call the "LAPP_motorStarter" block.

The "LAHP_MotorStarter" runtime screen is shown below.

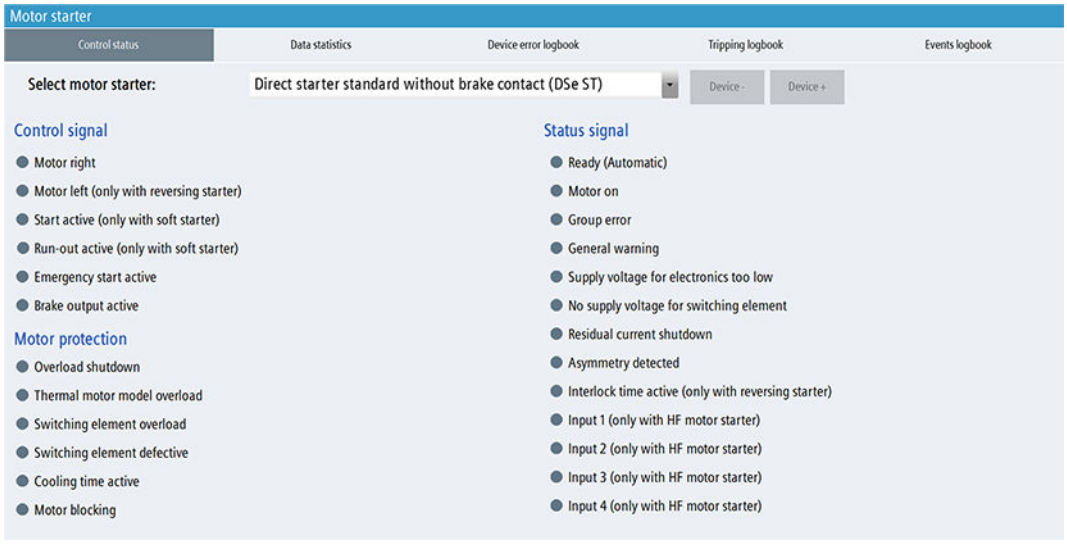


Figure 2-49 Motor starter

The LAPP_typeStarterConfig data type is used to configure the motor starter blocks.

You need to configure the following for "LAHP_MotorStarter" screen to work as intended:

- Block configuration (Page 74) in Step 7
  For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).
- Text list configuration (Page 147) in WinCC Unified

2.10.2.2 Block interface - LAPP_motorStarter

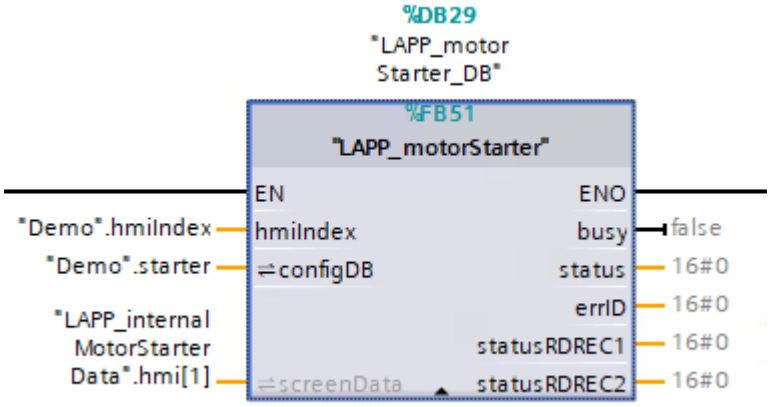


Figure 2-50 LAPP_motorStarter

Parameters

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
configDB	InOut	Array of LAPP_typeStarterConfig	The database interfaced to configure the motorstater device
screenData	InOut	LAPP_typeHmiMotorStarter	Interface the HMI tag from "LAPP_internalMotorStarterData.hmi[y]". **Note**: "LAPP_internalMotorStarterData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
busy	Output	Bool	"TRUE" communication with the drive
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the blocks. For more information refer to Error codes (Page 76).
statusRDREC1	Output	DWord	Status of the RDREC instruction. For more information, refer to RDREC: Read data record TIA Portal online help.
statusRDREC2	Output	DWord	Status of the RDREC instruction. For more information, refer to RDREC: Read data record TIA Portal online help.

y- indicates HMI device number.

2.10.2.3 Data type

LAPP_typeStarterConfig

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

Name	Type	Description
deviceType	Byte	Motor starter type. 0 - ET200pro 1 - ET200SP
hwID	HW_IO	Hardware identification of the DP slave, taken from "Devices & networks"

2.10.2.4 Error codes

List

Table 2-24 Description of the output parameter status of "LAPP_motorStarter" block

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	Invalid device type.
16#8203	Motor starter diagnostic sceens not active.
16#8204	Invalid input array configuration.
16#8600	Status of the read record block.
16#8601	Status of the read record block.
16#8400	No connection to HMI.

2.10.3 LAPP_safety

2.10.3.1 Overview

Safety block is used for monitoring the failsafe functions of the PLC.

For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).

2.10.3.2 Block interface

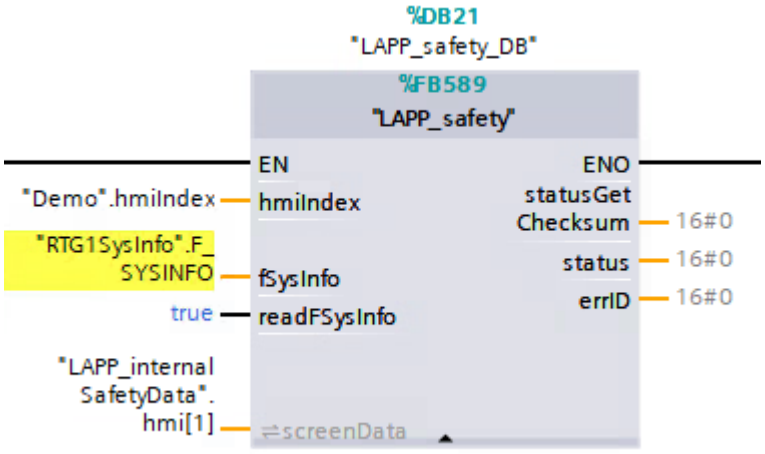


Figure 2-51 LAPP_safety block

Parameters of the "LAPP_safety" block

Name	Declaration	Type	Standard	Description
hmiIndex	Input	USInt	-	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
fSysInfo	Input	F_SYSINFO	RTG1SysInfo.F_ SYSINFO	S_SYSINFO parameter of the F-runtime group info DB For more information, refer F-runtime group information DB (S7-1200, S7-1500) in TIA Portal online help. You can view "Step 7 Safety" blocks only when safety function is configured for PLC.
readFSysInfo	Input	Bool	-	Positive edge copies the current data to the acceptance data
screenData	InOut	LAPP_typeHmi-Safety	-	Interface the HMI tag from "LAPP_internalSafetyData.hmi[y]". **Note**: "LAPP_internalSafetyData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
statusGetChecksum	Output	Word	-	Status of the GetChecksum instruction

Name	Declaration	Type	Standard	Description
status	Output	Word	-	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	-	Displays the error codes of the block. For more information refer to Error codes (Page 78).

y- indicates HMI device number.

2.10.3.3 Error codes

List

Following are the error codes for "LAPP_safety".

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8400	No connection to HMI.

2.10.4 LAPP_RFID

2.10.4.1 Overview

The LAPP_RFID block supplies the WinCC screens for the RFID diagnostics. The displayed data is read in via the interface. You need to call this block cyclically once for each configured ident device, whereby the selected device parameter corresponds to the corresponding values from the WinCC text list LAHP_IdentDeviceName.

The "LAHP_RFID" runtime screen is shown below.

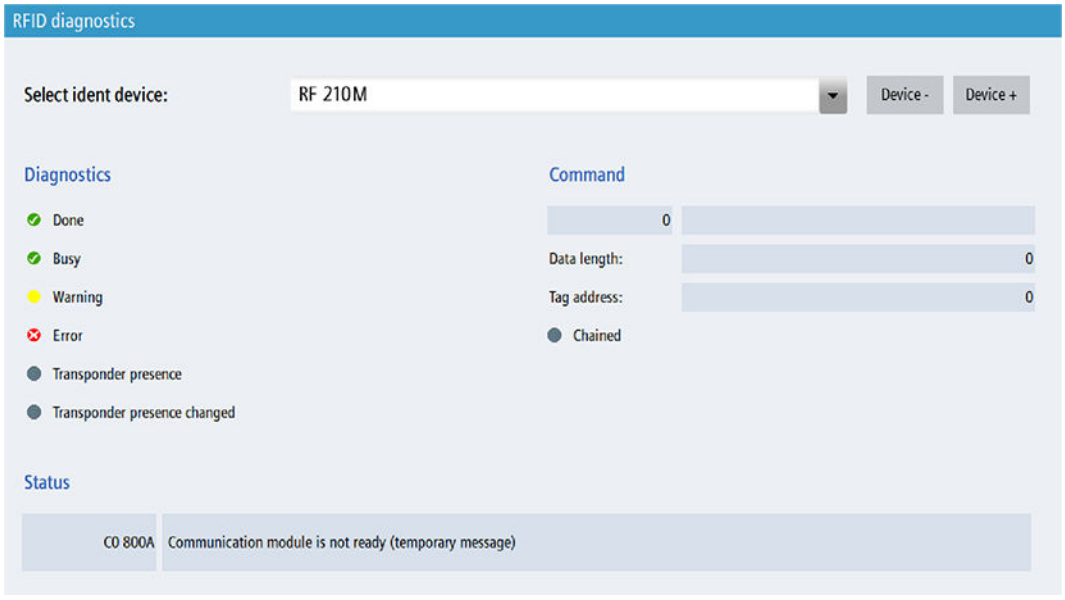


Figure 2-52 RFID diagnostics

You need to configure the following for "LAHP_RFID" screen to work as intended:

- Block configuration (Page 80) in step 7
  For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).
- Text list configuration (Page 149) in WinCC Unified.

2.10.4.2 Block interface

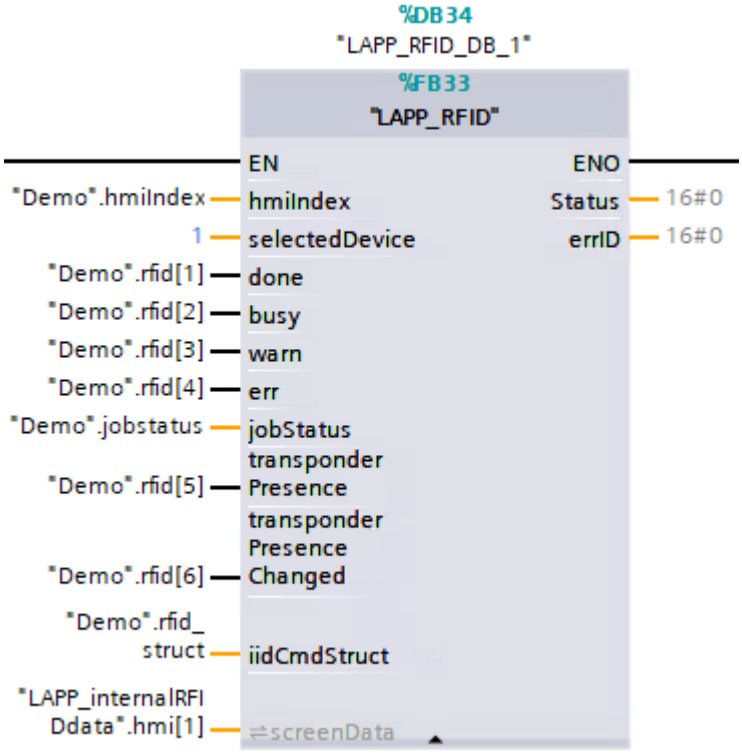


Figure 2-53 LAPP_RFID block

Parameters

Table 2-25 Parameters of the LAPP_RFID block

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
selectedDevice	Input	UInt	Selecting the ident device: Value from the text list LAHP_IdentDeviceName
done	Input	Bool	Assign the output parameter "done" of the ident blocks or of the ident profile
busy	Input	Bool	Assign the output parameter "busy" of the ident blocks or of the ident profile
warn	Input	Bool	Assign the output parameter "warning" of the ident blocks or of the ident profile
err	Input	Bool	Assign the output parameter "error" of the ident blocks or of the ident profile
jobStatus	Input	DWord	Assign the output parameter "status" of the ident blocks or of the ident profile

Name	Declaration	Type	Description
transponderPresence	Input	Bool	Assign the output parameter "presence" of the ident blocks or of the ident profile
transponderPresence-Changed	Input	Bool	Assign the output parameter "tpc" of the ident blocks or of the ident profile
iidCmdStruct	Input	IID_CMD_ STRUCT	Assign the input parameter of the current command of the ident blocks or of the ident profile
screenData	InOut	LAPP_typeHmiRFID	Interface the HMI tag from "LAPP_internalRFIDdata.hmi[y]". **Note**: "LAPP_internalRFIDdata" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errId	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 81).

y- indicates HMI device number.

2.10.4.3 Error codes

List

Table 2-26 Description of error codes of "LAPP_RFID"

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	Device in the screen not selected.
16#8400	No connection to HMI.

2.10.5 LAPP_EKS

2.10.5.1 Overview

The "LAPP_EKS" function block is used to read the key data of Electronic Key System (EKS). The function block has to be called in the cyclic program once per EKS adapter (key recording) with different instance DBs.

The "LAHP_EKS" runtime screen is shown below.

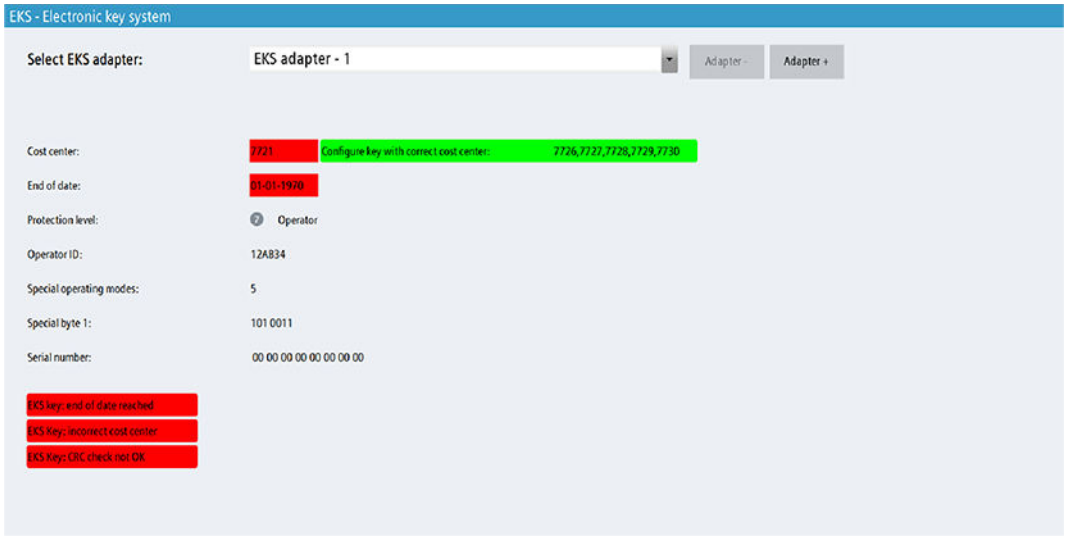


Figure 2-54 EKS - Electronic key system

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

Note

The expiry date of the key is compared with the system clock of the S7 CPU. Ensure 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 the "CostCenter" input parameter.

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.

The LAPP_typeEKSIntf data type is used to configure the EKS blocks.

You need to configure the following for "LAHP_EKS" screen to work as intended:

- Block configuration (Page 83) in Step 7
  For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).
- Tag creation, text and graphic list (Page 81) configuration in WinCC Unified

2.10.5.2 Block interface

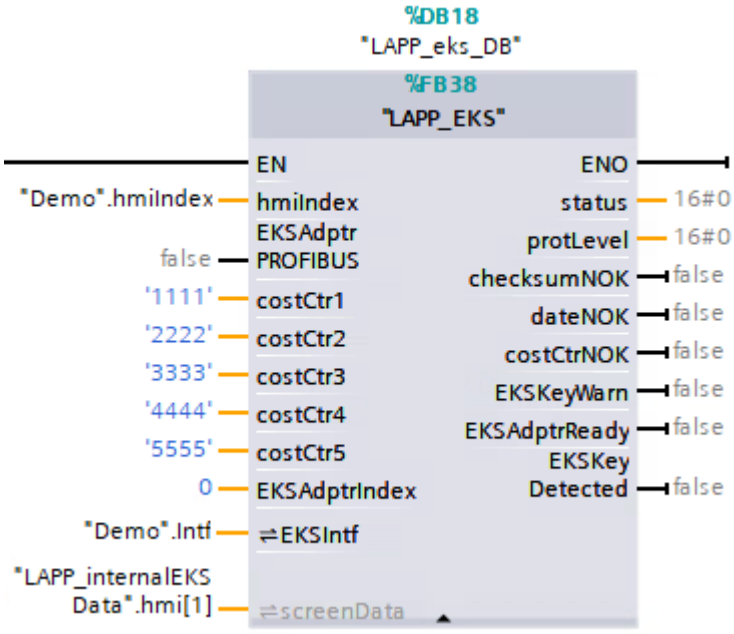


Figure 2-55 LAPP_Eks block

Parameters

Table 2-27 Parameters of the "LAPP_EKS" block

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
EKSAdptrPROFIBUS	Input	Bool	Interface of the EKS adapter: PROFINET=FALSE PROFIBUS=TRUE
costCtr1	Input	STRING[4]	Cost center number
costCtr2	Input	STRING[4]	Cost center number
costCtr3	Input	STRING[4]	Cost center number
costCtr4	Input	STRING[4]	Cost center number
costCtr5	Input	STRING[4]	Cost center number
EKSAdptrIndex	Input	UInt	HMI index for displaying the selected key data in the screen "EKS - electronic"
EKSIntf	InOut	LAPP_typeEKSIntf	Interface to the EKS adapter; PLC tag with "LAPP_typeEksIntf" data type.

Name	Declaration	Type	Description
screenData	InOut	LAPP_typeHmiEKSData	Interface the HMI tag from "LAPP_internal-EKSData.hmi[y]". **Note**: "LAPP_internalEKSData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
protLevel	Output	Byte	Output of the protection level
checksumNOK	Output	Bool	Checksum of the key not OK
dateNOK	Output	Bool	Expiry date of the key elapsed
costCtrNOK	Output	Bool	Cost center does not match parameterized cost center
EKSKeyWarn	Output	Bool	EKS key will become invalid in less than 30 days
EKSAdptrReady	Output	Bool	EKS adapter ready to operate
EKSKeyDetected	Output	Bool	EKS key recognized

y- indicates HMI device number.

For more information, refer to Configuring PLC blocks (Page 18)

2.10.5.3 Data types

LAPP_typeEKSIntf

Table 2-28 Structure of data type LAPP_typeEKSIntf

Name	Type	Description
keyData	Array [0..127] of byte	Configure the key data to display on screen.

2.10.5.4 Authorization levels concept

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

Table 2-29 Authorization levels concept

Standard Siemens authorization levels	Authorization	Your EKS
Authorization level 1 (machine manufacturer)	Manufacturer, service, maintenance engineer	Authorization level 1 Default: Red key
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 Default: Green key
Authorization level 5 (qualified operator)	Not used	Not used
Authorization level 6 (trained operator)	Operator	Authorization level 6 Default: Black key
Authorization level 7 (semi-skilled operator)	No authorization	No key Default: Grey key

Note

- You can modify authorization names as per the requirement in "LAHP_EksProtectionLevel" text list.
  For more information, refer to Editing EKS authorization names (Page 154).
- You can modify the default key as per the requirement in "LAHP_User_FullHD" or "LAHP_User_HDReady" or "LAHP_User_NonHD" graphic list.
  For more information, refer to Editing graphics (Page 151).

2.10.5.5 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.

Data structure

Write to the EKS key, considering 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

The function block described here "LAPP_EKS" 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. To use an EKS key with this function block, it has to be written in the following format as of byte 84:

Table 2-30 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		Spare
106 - 113	8 bytes	Hexadecimal	Special bytes	Spare

2.10.5.6 Error codes

List

Table 2-31 Description of error codes of "LAPP_EKS"

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8400	No connection to HMI.

2.10.6 LAPP_camera

2.10.6.1 Overview

You can view live image from camera.

- The camera image allows you to track remote processes and monitor difficult-to-access areas.
- You can create and configure the required number of cameras.

The "LAHP_Camera" runtime screen is shown below.

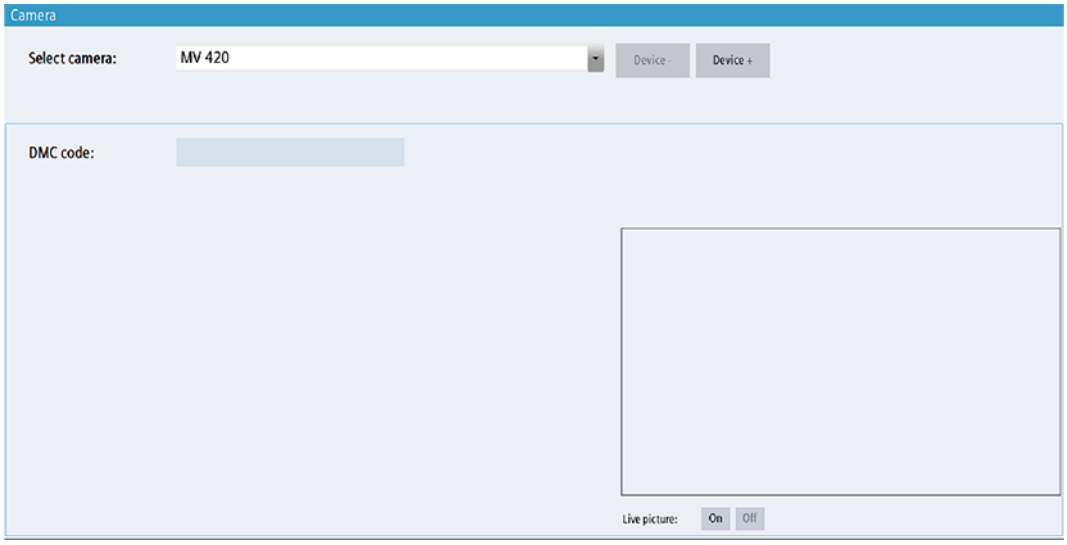


Figure 2-56 Camera

The LAPP_typeCameraIPConfig data type is used to configure the motor starter blocks.

You need to configure the following for "LAHP_Camera" screen to work as intended:

- Block configuration (Page 87) in Step 7
  For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).
- Text list configuration (Page 156) in WinCC Unified

2.10.6.2 Block interface

This block is used for data handling and configurations.

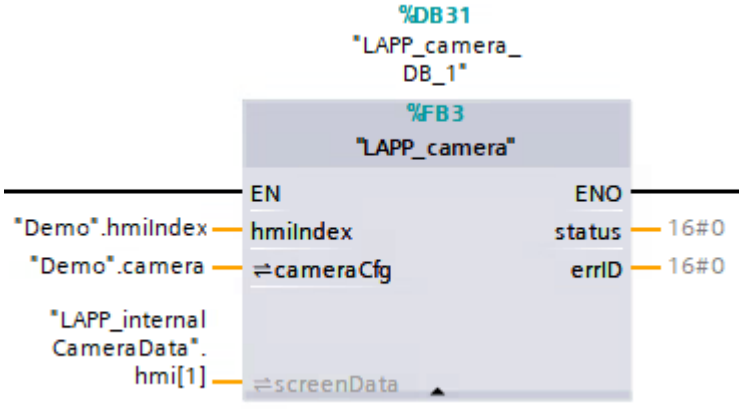


Figure 2-57 LAPP_camera block

Parameters

Table 2-32 Parameters of "LAPP_camera" block

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
cameraCfg	InOut	Array of LAPP_typeCameraIP-Config	Interface the PLC tag with data type "LAPP_typeCameraIPConfig"
screenData	InOut	LAPP_typeHmiCamera	Interface the HMI tag from "LAPP_internalCameraData.hmi[y]". Note: "LAPP_internalCameraData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information, refer to Error codes (Page 89).

y- indicates HMI device number.

See also

Configuring PLC blocks (Page 18)

2.10.6.3 Data type

LAPP_typeCameraIPConfig

This data type is used for configuring the IP address.

Table 2-33 Structure of data type LAPP_typeCameraIPConfig

Name	Type	Description
cameraIP	String[15]	Configure the IP address of the camera

2.10.6.4 Error codes

List

Table 2-34 Description of error codes of "LAPP_camera"

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	Invalid IP address of the camera.
16#8400	No connection to HMI.

2.10.7 LAPP_EETransfer

2.10.7.1 Overview

The energy block is used for measuring the energy consumption for different machine states. The "LAHP_EnergyEfficiency" screen is shown below.

Figure 2-58 Energy efficiency

Runtime interface (LAPP_EETransfer)

In the PLC, four blocks are provided for the use of **EE@TRANSLINE**. The **LAPP_EE** energy efficiency data block is the interface to the energy efficiency screens.

The **LAPP_EETransfer** block supplies the **LAPP_EE** block with the data relevant for the screen. The measuring points for the energy efficiency (e.g. electrical energy and compressed air) are transferred to **LAPP_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 **LAPP_EE** block. **LAPP_EETransfer** also calls the **EnS_EEm_Calc** block internally. **EnS_EEm_Calc** is responsible for the scaling, overflow check, 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 **LAPP_EEmMachine** data block.

All configurations are performed and operations controlled in the **LAPP_EEmMachine** block. The measuring points are already pre-assigned according to the **EE@TRANSLINE** concept. For this reason, the following variables of the **LAPP_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.

For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).

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

Table 2-35 **Pre-assigned measuring points of the energy efficiency**

No.	Energy type	Measured variable	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	kvarh	Yes
4	Compressed air	Maximum volume flow	normally m^3	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	Axis energy		kWh	Yes

2.10.7.2 Block interface

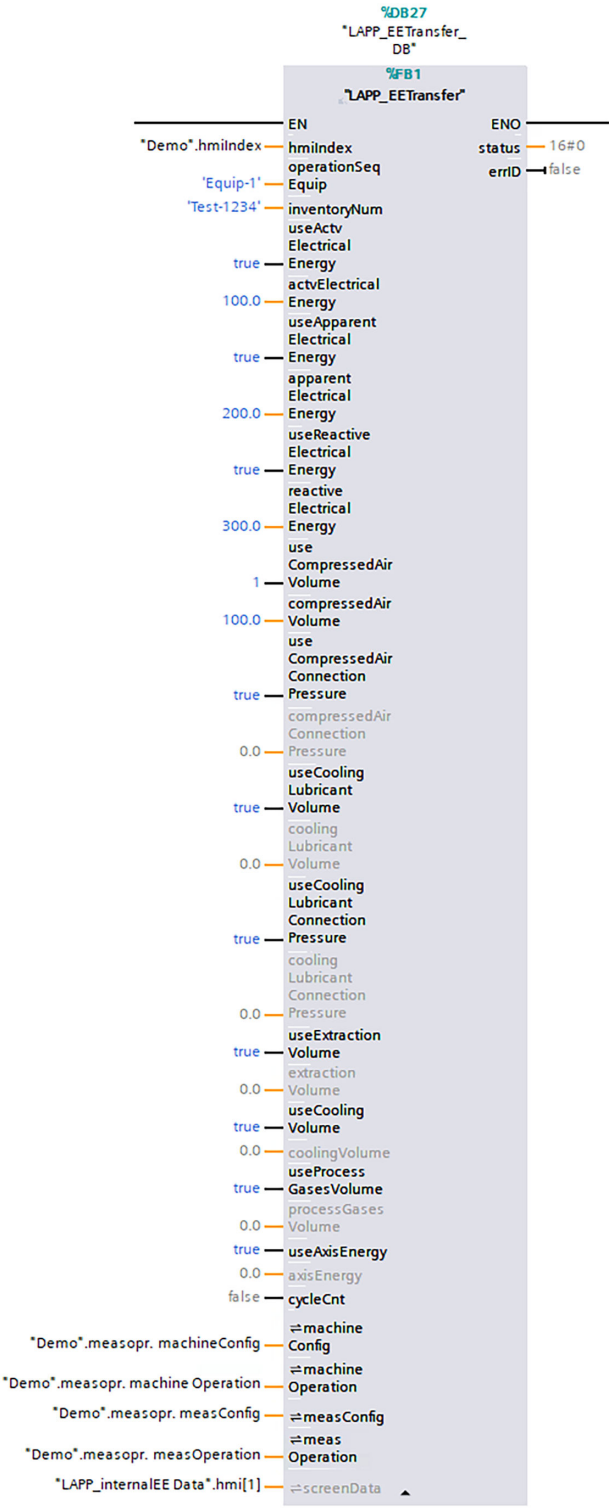


Figure 2-59 LAPP_EETransfer block

Parameters

Table 2-36 Parameters of "LAPP_EETransfer" block

Name	Declaration	Type	Standard	Description
hmiIndex	Input	USInt	---	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
operationSeqEquip	Input	WString[10]	---	Operation sequence of the equipment
inventoryNum	Input	WString[20]	---	Inventory number
useActvElectricalEnergy	Input	Bool	---	Activate electrical energy - active energy
actvElectricalEnergy	Input	LReal	---	Electrical energy - active energy [kWh]
useApparentElectricalEnergy	Input	Bool	---	Activate electrical energy - apparent energy
apparentElectricalEnergy	Input	LReal	---	Electrical energy - apparent energy [kVAh]
useReactiveElectricalEnergy	Input	Bool	---	Activate electrical energy - reactive energy
reactiveElectricalEnergy	Input	LReal	---	Electrical energy - reactive energy [kvarh]
useCompressedAirVolume	Input	Bool	---	Activate compressed air - volume
compressedAirVolume	Input	Real	---	Compressed air - volume [normally m^3]
useCompressedAirConnectionPressure	Input	Bool	---	Activate compressed air - supply pressure
compressedAirConnectionPressure	Input	Real	---	compressed Air Connection Pressure
useCoolingLubricantVolume	Input	Bool	---	Activate cooling lubricant - volume
coolingLubricantVolume	Input	LReal	---	Cooling lubricant - volume [l]
useCoolingLubricantConnectionPressure	Input	Bool	---	Activate cooling lubricant - supply pressure
coolingLubricantConnectionPressure	Input	Real	---	Cooling lubricant - supply pressure [bar]
useExtractionVolume	Input	Bool	---	Activate extraction - volume
extractionVolume	Input	LReal	---	Extraction - volume [l]
useCoolingVolume	Input	Bool	---	Activate cooling - volume
coolingVolume	Input	LReal	---	Cooling - volume [l]

Name	Declaration	Type	Standard	Description
useProcessGasesVolume	Input	Bool	---	Activate process gases - volume
processGasesVolume	Input	LReal	---	Process gases - volume [m³]
useAxisEnergy	Input	Bool	---	Activate axis energy
axisEnergy	Input	LReal	---	Axis energy
cycleCnt	Input	Bool	---	Increase cycle clock for rising edge
machineConfig	InOut	EnS_ EEm_ typeConfig	EEmMachine. machine Configuration	Configuration data of the machine Note: You can create an array of this datatype.
machineOperation	InOut	EnS_ EEm_typeOperation	EEmMachine. machineOperation	Operating data of the machine Note: You can create an array of this datatype.
measConfig	InOut	Array of EnS_EEm_typeMeasConfig	EEmMachine. measurements Configuration	Operating data of the machine Note: You can create an array of this datatype.
measOperation	InOut	Array of EnS_EEm_typeMeasOperation	EEmMachine. measurements Operation	Operating data of the measuring points Note: You can create an array of this datatype.
screenData	InOut	LAPP_typeHmiEEData	---	Interface the HMI tag from "LAPP_internalEEData.hmi[y]". **Note**: "LAPP_internalEEData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
status	Output	Word	---	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Bool	---	Displays the error codes of the block. For more information refer to Error codes (Page 94).

y- indicates HMI device number.

For more information on the description of all other values refer the status parameter of **EnS_Em_Calc** in the TIA Portal online help.

2.10.7.3 Error codes

List

Table 2-37 Description of error codes of "LAPP_EETransfer"

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	Invalid number of energy types.
16#8203	Invalid array of lower bounds.
16#8204	Invalid array of upper bounds.
16#8400	No connection to HMI.

2.10.8 LAPP_plcSystemData

2.10.8.1 Overview

The "LAPP_PLCSystemData" block reads the PLC system data on the first scan of PLC startup.

The "LAHP_PLCSystemData" runtime screen is shown below.

Figure 2-60 PLC system data

For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).

2.10.8.2 Block interface

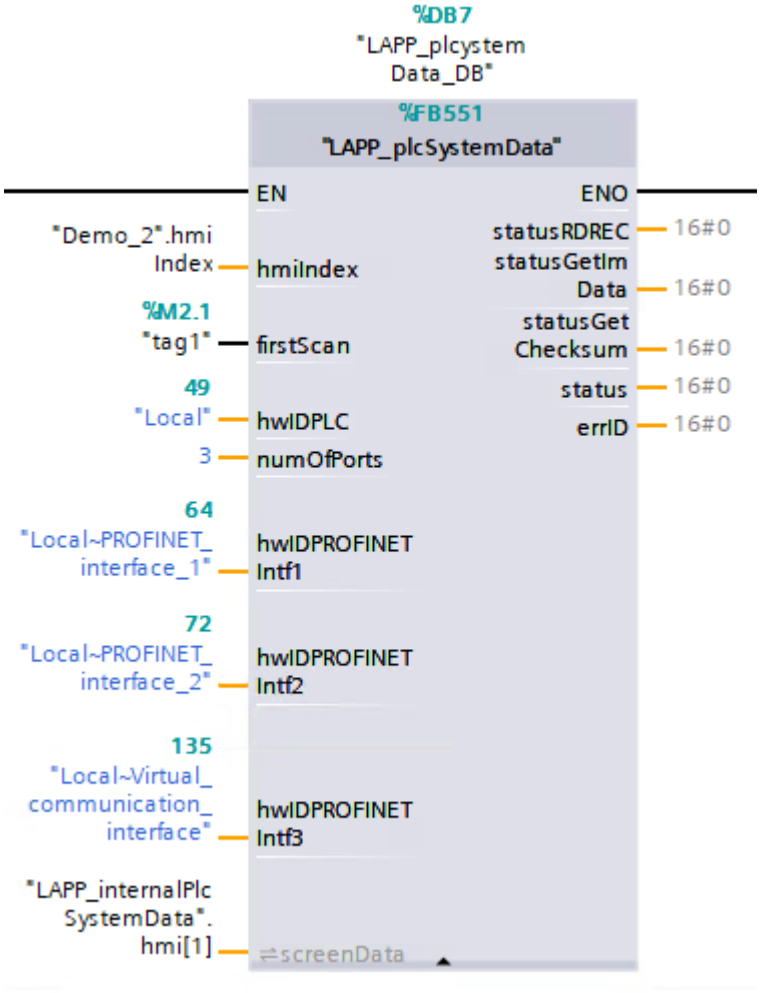


Figure 2-61 LAPP_plcSystemData block

Parameters

Table 2-38 Parameters of the "LAPP_plcSystemData" block

Name	Declaration	Type	Description
hmiIndex	Input	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
firstScan	Input	Bool	Startup bit. For more information to enable first scan, refer to Enabling first scan (Page 96).
hwIDPLC	Input	HW_IO	PLC hardware ID

Name	Declaration	Type	Description
numOfPorts	Input	Int	Number of ports configured (Minimum 1 and maximum 3)
hwIDProfinetIntf1	Input	HW_IO	Hardware ID of the PROFINET interface 1
hwIDProfinetIntf2	Input	HW_IO	Hardware ID of the PROFINET interface 2
hwIDProfinetIntf3	Input	HW_IO	Hardware ID of the PROFINET interface 3
screenData	InOut	LAPP_typeHmiPLCSystemData	Interface the HMI tag from "LAPP_internalPlcSystemData.hmi[y]". **Note**: "LAPP_internalPlcSystemData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
statusRDREC	Output	DWord	Status of the RDREC instruction. For more information on RDREC instruction, refer to RDREC: Read data record in TIA Portal online help.
statusGetImData	Output	Word	Status of the GET_IM_DATA instruction. For more information on GET_IM_DATA instruction, refer Get_IM_Data: Reading identification and maintenance data in TIA Portal online help.
statusGetChecksum	Output	Word	Status of the GetChecksum instruction. For more information on GetChecksum instruction, refer GetChecksum: Read out checksum in TIA Portal online help.
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 97).

y- indicates HMI device number.

2.10.8.3 Enabling first scan

Procedure

To activate first scan bit, follow the below steps:

1. In the "Network view", click PLC.
2. Navigate to "Properties > General" tab in the inspector window.

3. Click "System and clock memory".
4. Enable the "Enable the use of system memory byte" checkbox in "System memory bits" as shown below:

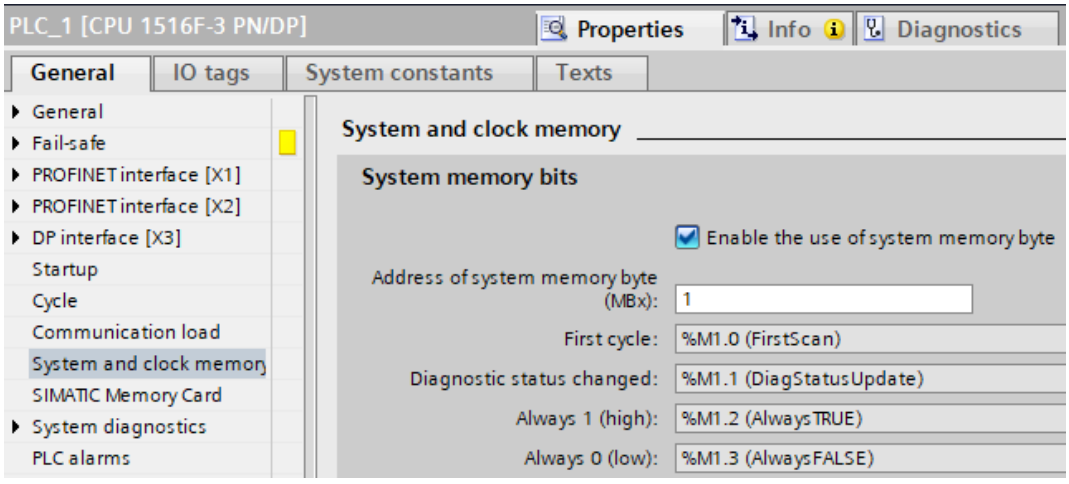


Figure 2-62 System memory bits

2.10.8.4 Error codes

List

Table 2-39 Description of error codes "LAPP_plcSystemData"

Error code	Description
16#8200	Faulty license.
16#8201	Invalid HMI index.
16#8202	PLC model not found.
16#8203	PLC configuration port error.
16#8400	No connection to HMI.

2.10.9 LAPP_interface

2.10.9.1 Overview

You can use "LAHP_Interface" screen to view the status of the inputs and outputs for the selected interface as shown:

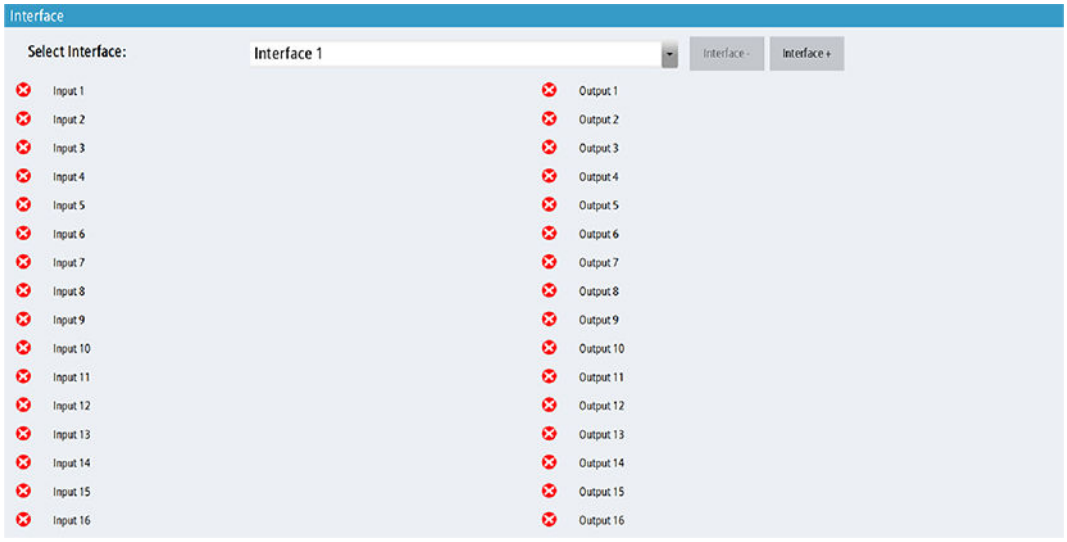


Figure 2-63 Interface

You need to map the IO parameters to "LAPP_interface" block for the "LAHP_Interface" screen to work as intended.

You need to configure the following for "LAHP_interface" screen to work as intended:

- Block configuration (Page 99) in Step 7

For more information on the configuring the block, refer to Configuring PLC blocks (Page 18).

- Text list configuration (Page 158) in WinCC Unified

2.10.9.2 Block interface

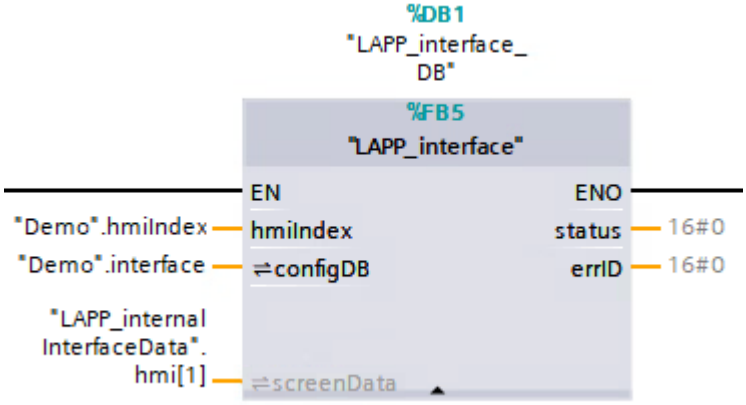


Figure 2-64 LAPP_interface

Parameters

Table 2-40 Parameters of the "LAPP_interface" block

Name	Declaration	Type	Description
hmiIndex	InOut	USInt	Configure the HMI Index which is defined while setting up the HMI device. **Note**: HMI index value cannot be 0. It should match with HMI index value used in License DB (LAPP_license).
ConfigDB	InOut	Array of LAPP_typeIntf-Config	Interface the PLC tag with LAPP_typeIntfConfig data type.
screenData	InOut	LAPP_typeHmiIntf	Interface the HMI tag from "LAPP_internalInterfaceData.hmi[y]". **Note**: "LAPP_internalInterfaceData" is an internal DB. Modify the array value to increase the HMI device support. For more information, refer to Configuring PLC blocks for different HMI (Page 25).
status	Output	Word	Displays the status of the block. • 16#7000: Block is in initialized state. • 16#7002: Block is in execution state and has no errors.
errID	Output	Word	Displays the error codes of the block. For more information refer to Error codes (Page 101).

y- indicates HMI device number.

2.10.9.3 Data type

LAPP_typeIntfConfig

Table 2-41 Structure of data type LAPP_typeIntfConfig

Name	Type	Description
in	Word	Input interface
out	Word	Output interface

Example

In the below image, "interface" tag is created with "LAPP_typeIntfConfig" with array value [0..9].

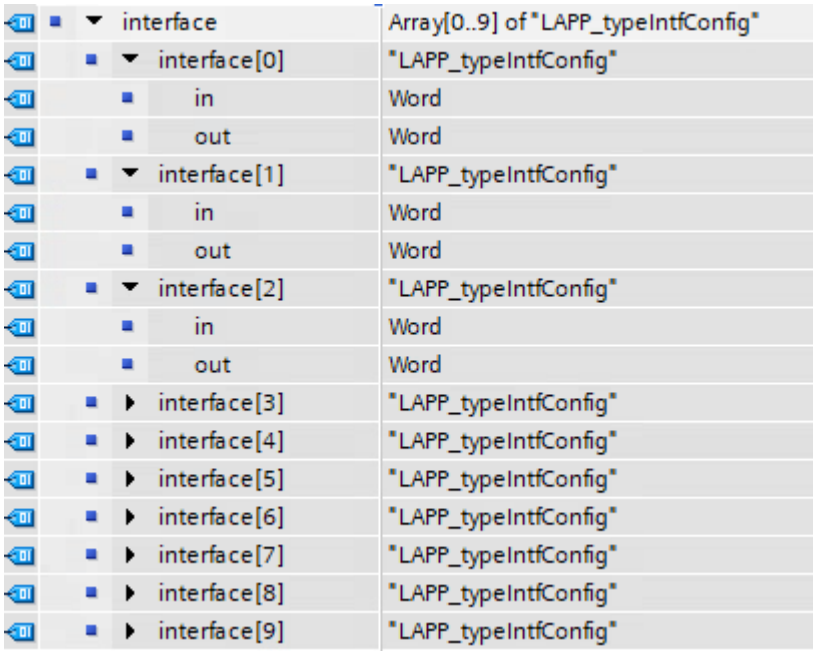
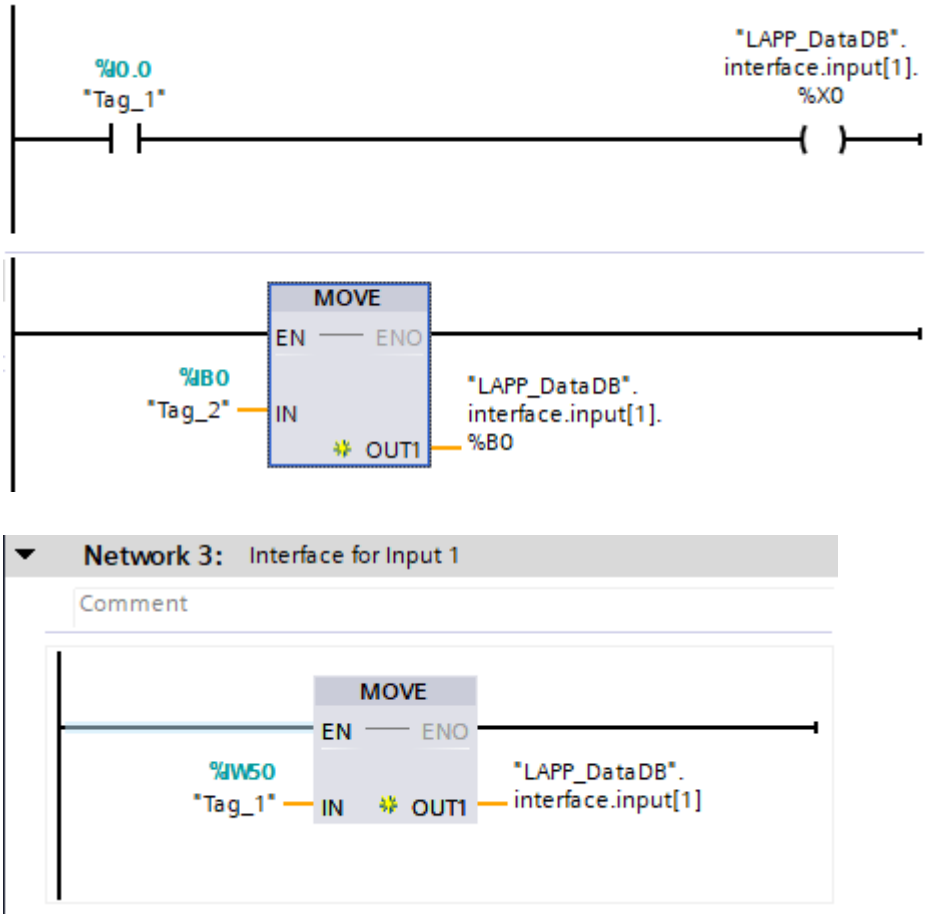


Figure 2-65 Interface tag

- interface[0] - Input and output for interface 1
- interface[1] - Input and output for interface 2
  .
  .
- interface[9] - Input and output for interface 10

2.10.9.4 Example

You can assign the inputs to the interface screen as shown



2.10.9.5 Error codes

List

Table 2-42 Description of error codes of "LAPP_interface"

Error code	Description
16#8200	Faulty license
16#8201	Invalid HMI index.
16#8400	No connection to HMI.

Automotive HMI plus screens

3

3.1 Overview

Header and Footer screens here are predefined screens provided by SIEMENS.

You can also add your own navigation screens, logos, layouts, and other customizations here.

⚠ CAUTION
Important It is not recommended to modify the standard screens with prefix "LAHP", delivered from Siemens, as this could lead to data loss and malfunctioning.

3.2 Configuring projects backup for different HMIs

You need to create separate project backup for each HMI. You need to define the HMI number in "hmiIndex" in "LAHP_Common".
For PLC, you can view the LAHP_Common" tag table as shown:

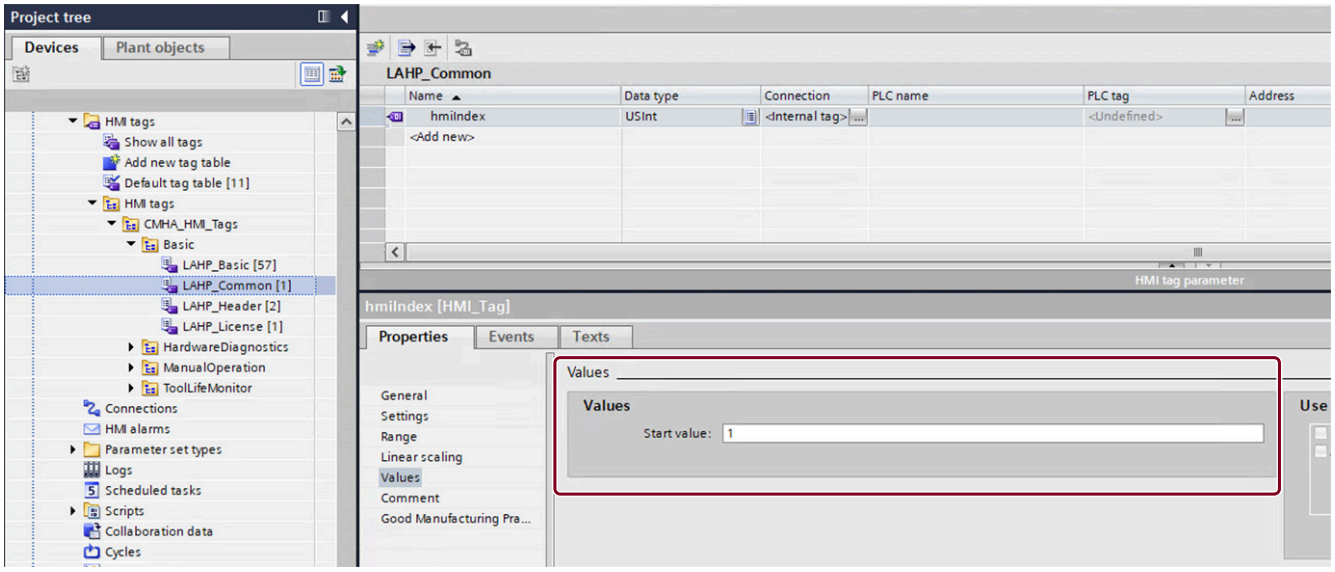


Figure 3-1 LAHP_Common

For CNC, you can view the "LAHP_Common" tag table as shown:

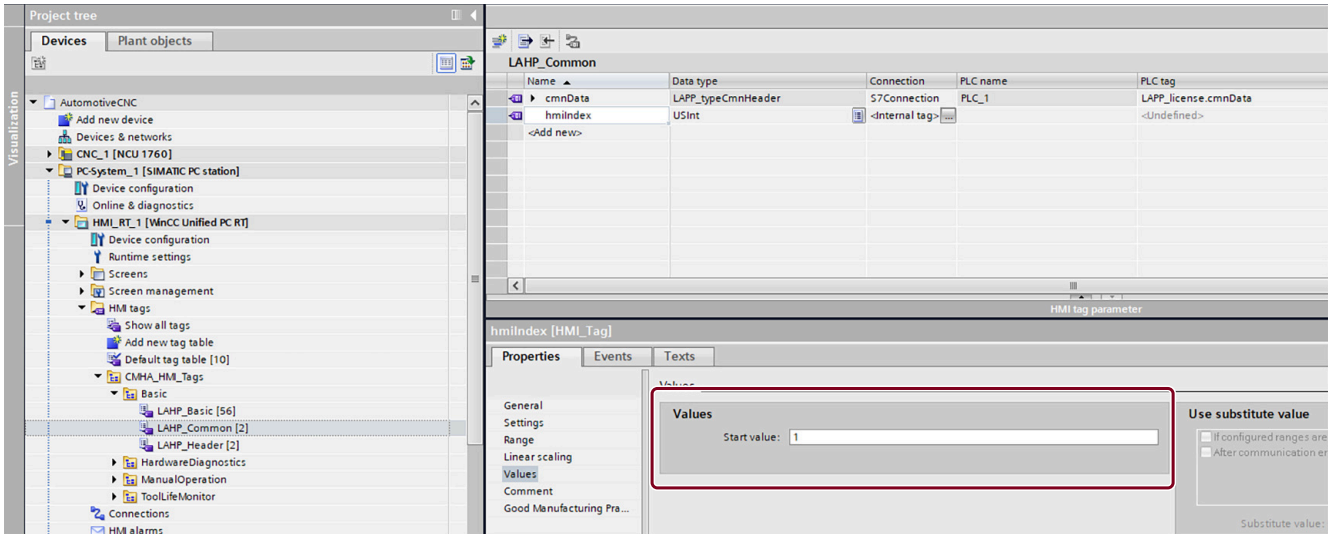


Figure 3-2 LAHP_Common

3.3 Header

3.3.1 Header

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

Note

- You can change the SIEMENS logo, displayed on the header.
- Header is displayed from WinCC.

- You have the option to modify,
  - the logo provided in the Header (Page 105)
  - machine information (Page 106) provided on the header
  - station details (Page 107) (station name and information) provided on the header

For more information on configuring the station status SVG icons, refer to StationStatus SVG in TIA Portal online help.

For more information on runtime behavior of the header, refer to Header in TIA Portal online help.

3.3.2 Editing logo

Procedure

You can edit the logo to be displayed on the header. To change the logo:

1. Navigate to “Project tree” >> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Navigate to "Graphic lists" window.
3. Select "LAHP_Logo" from the "Graphic list" window.

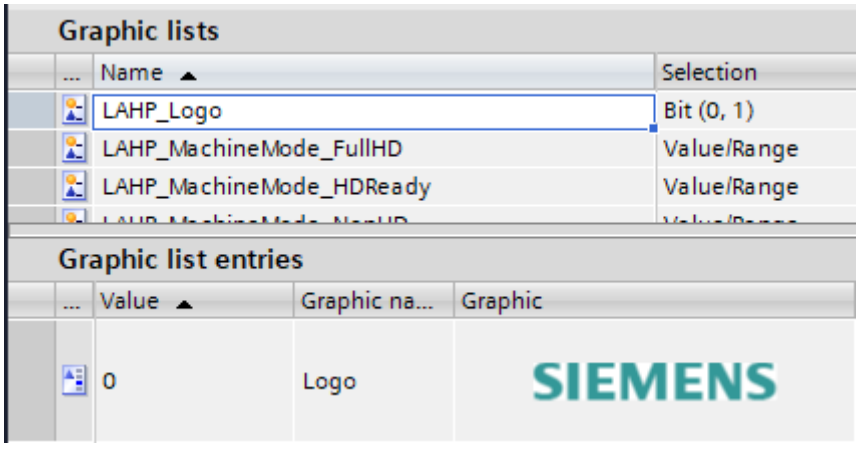


Figure 3-3 Graphic list

4. In "Graphic list entries " window, select the required logo.

Result

The logo is updated.

3.3.3 Editing Machine Information

Procedure

To edit the machine info:

1. Navigate to “Project tree” >> IPC device >> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAPP_Machine1Info", in the "Text list entries" window, modify the text in “Text” field.

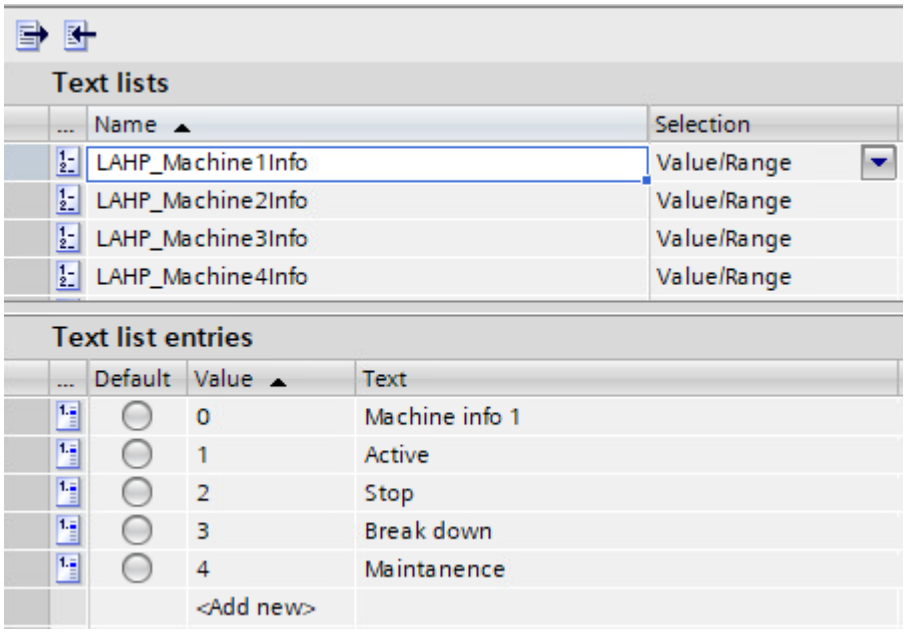


Figure 3-4 LAHP_Machine1Info

Note

Follow the same steps to edit the required information for other machines.

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

Result

In runtime, the machine information appears as configured.

3.3.4 Editing Station details

Procedure

To edit the station name:

1. Navigate to “Project tree” >> IPC device >> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAPP_StationName", in the "Text list" window.

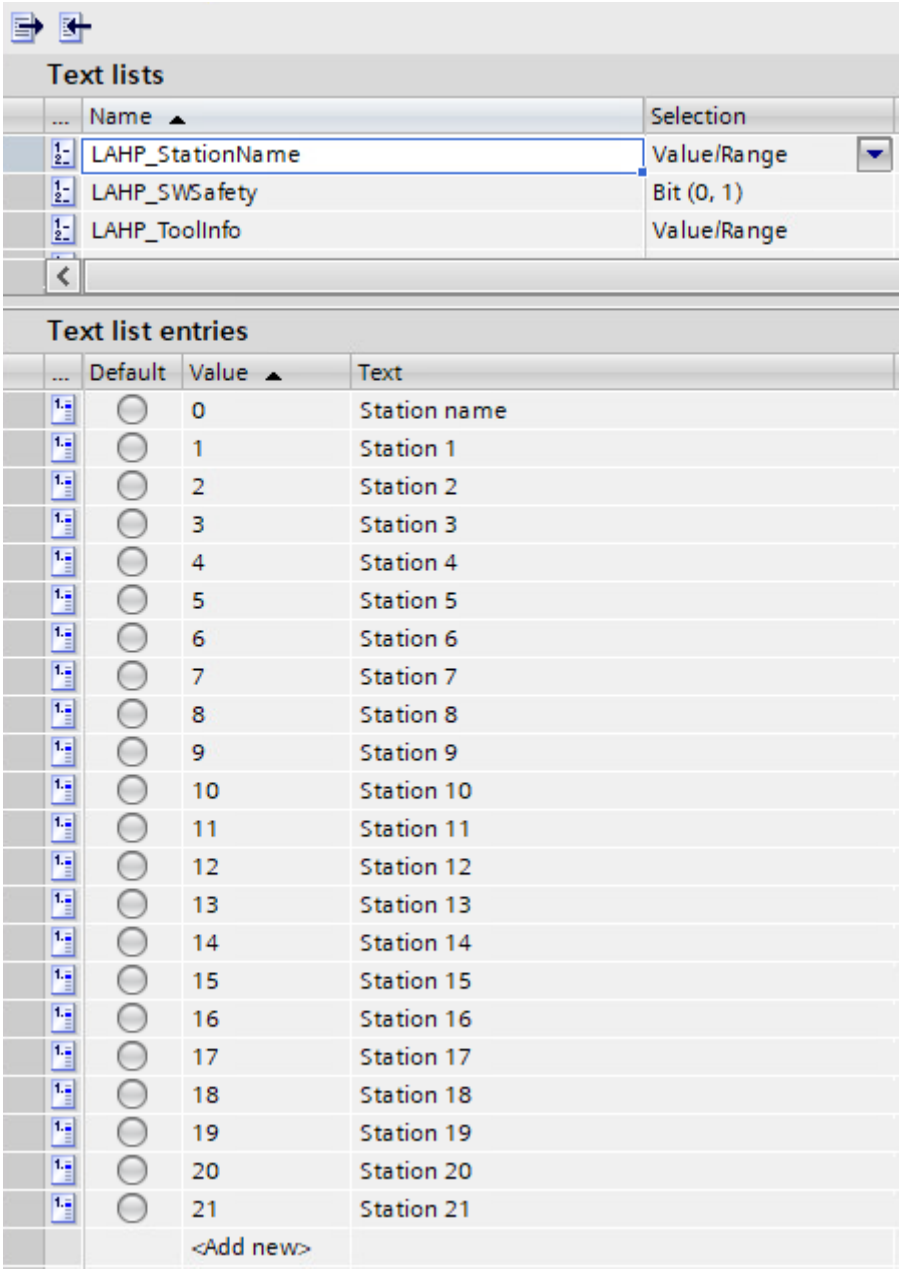


Figure 3-5 StationName

3. In "text list entries" window, select the required value and modify the text as per the requirement.

To edit the station info:

1. Navigate to “Project tree” >> IPC device >> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAPP_StationInfo", in the "Text list" window.

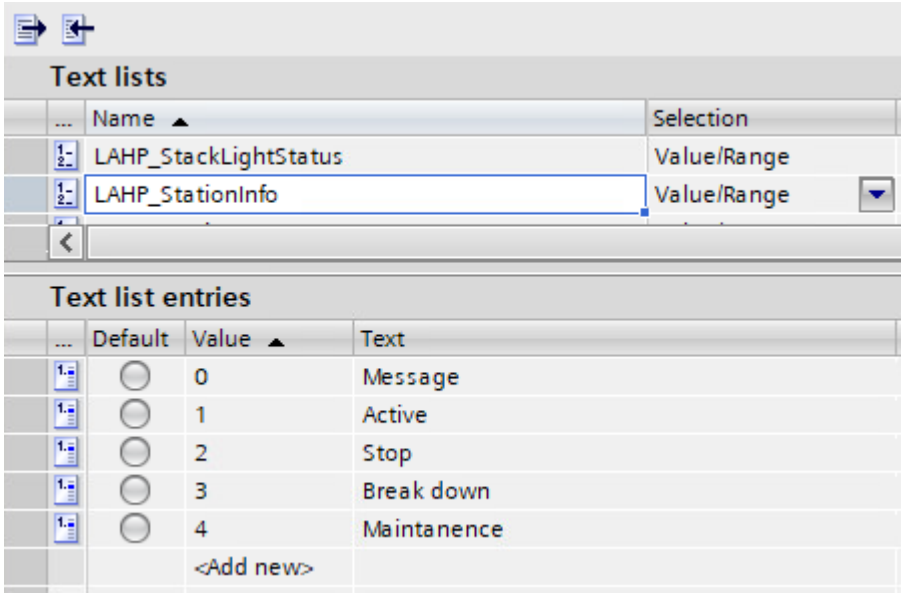


Figure 3-6 StationInfo

3. In "text list entries" window, select the required value and modify the text as per the requirement.

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

Result

In runtime, the station name and information appears as configured.

3.4 Footer

Footer

You can configure your own buttons and define new functionalities as per requirement for different operations in WinCC Unified and use them in runtime. The SVGs can be configured as footer icons. The SVGs available to configure the footer icons are:

- CMHFooter
- OEMFooter

3.5 Configuring LAHP_ManualOperation (MCP/Touch)

3.5.1 Configuring as a screen

3.5.1.1 Creating manual operation screen

Procedure

To create manual operation screen:

1. From "Global library > S71500 PLC Automotive Program > Types > Faceplates_PLC > Common", add the libraries to "Project library" as shown:

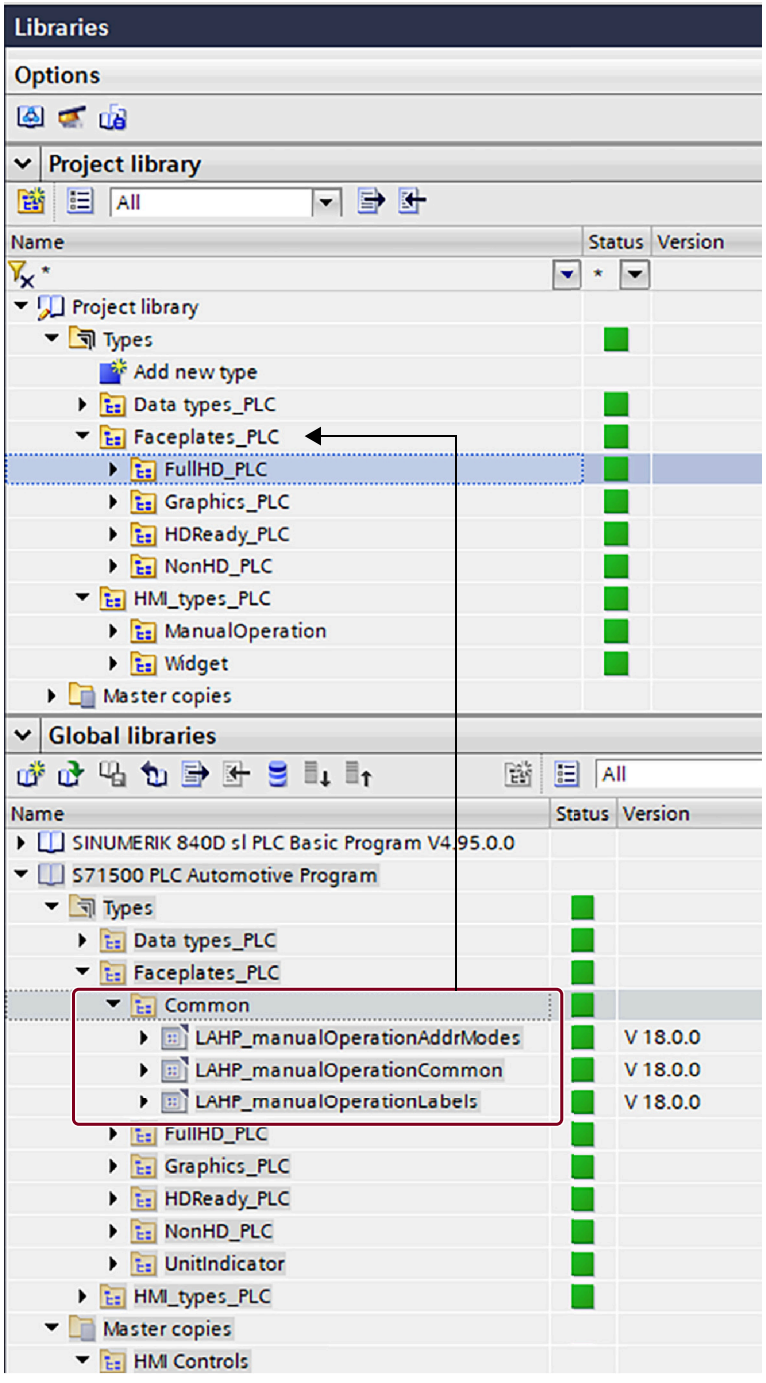


Figure 3-7 Libraries

2. In "Add-Ins" task card, navigate to "Add-Ins" and search for "CMHAddin.addin".

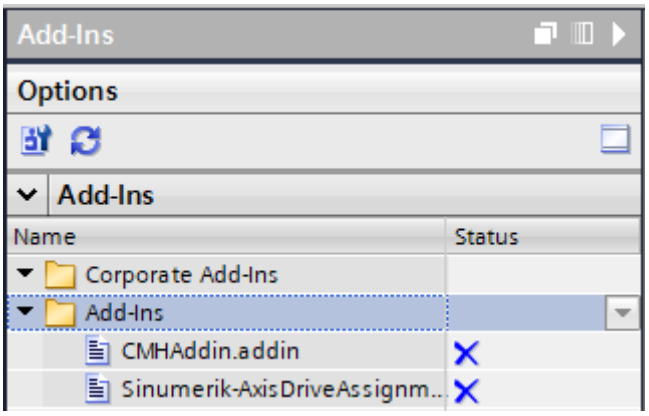


Figure 3-8 Add-Ins

3. Right-click on "CMHAddin.addin" and then click on "Activate".

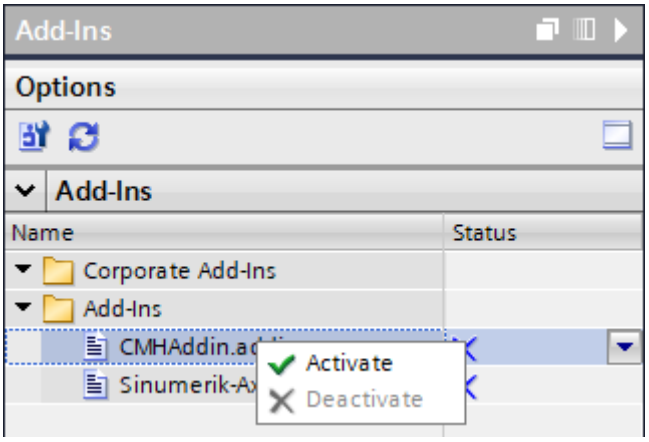


Figure 3-9 Add-Ins

The "Add-in activation" window opens.

4. Click "Yes", to apply the permission.

5. In "Project tree", right-click on HMI device and then navigate to "CHM Addin" >> "Setup" >> click on "Create screen".

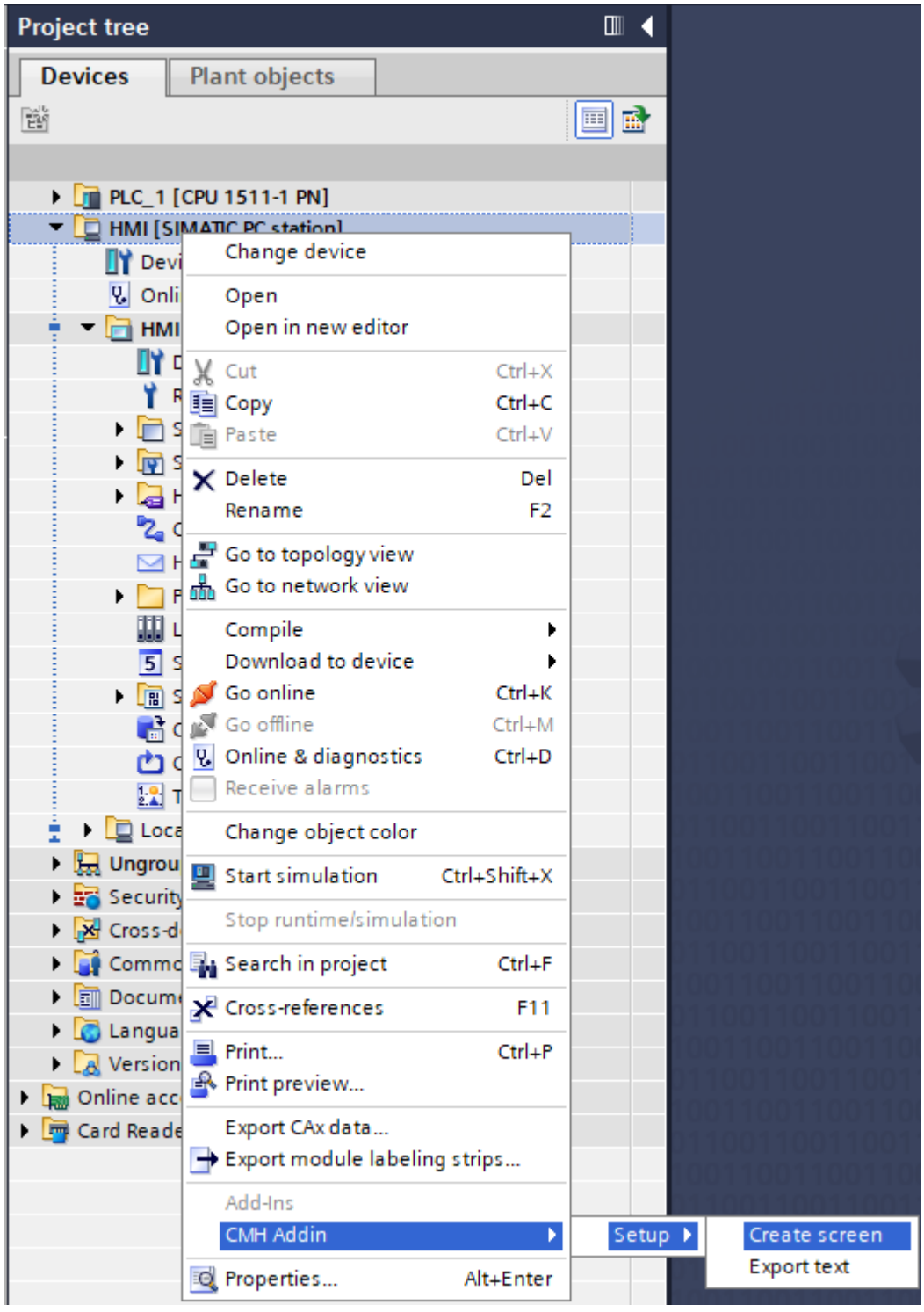


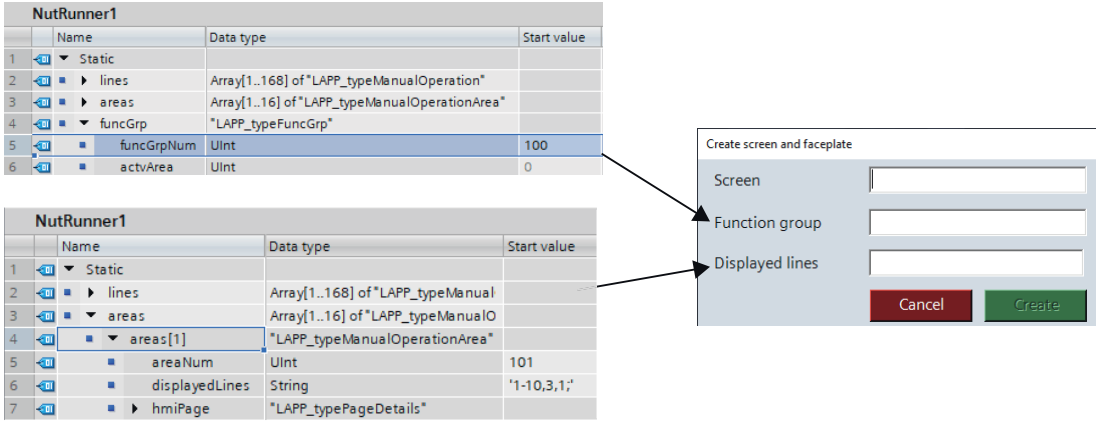
Figure 3-10 Project tree

The "Create screen and faceplate" pop-up window opens.

6. Enter the following details
   - Screen name
   - Function group number configured in PLC
   - The lines to be displayed. Define the lines as per the requirement which belongs to same function group.

Note

- Ensure to enter the value within single quotation mark and end it with semicolon ('...;').
- You can enter multiple line numbers by segregating them using comma (1,10)
- You can enter the range of line numbers by using hyphenation (1 - 10).
- You can enter the lines to be displayed in any order as mentioned below:
  - incremental range of line values (1 to 10)
  - decremental range of line values (102-78)
  - single line number (1)
  - repeated line number ('1-10, 3, 1;')



7. Click "Create".
   A screen gets created with number of lines defined.

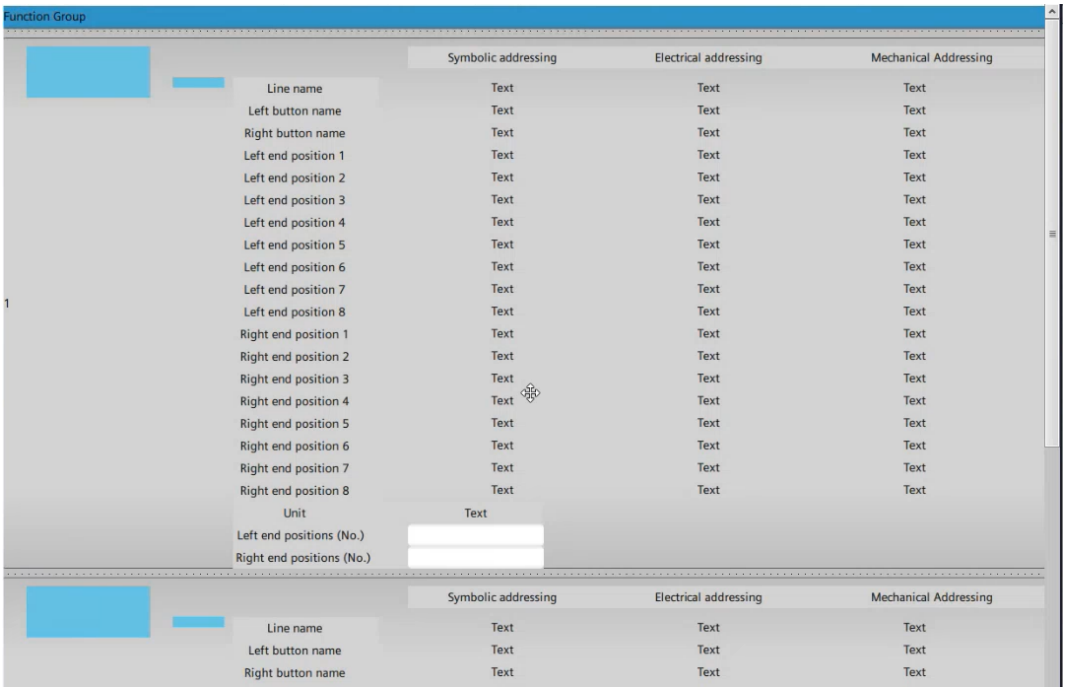


Figure 3-11 Manual operation Add-in screen

Note

This screen is not meant for manual operation in runtime, it is graphical representation to create text in JSON files.

8. Select the button to configure the color of the button required for the line. For more information on color with respect to different states of the button, refer to Touch-operated mode.

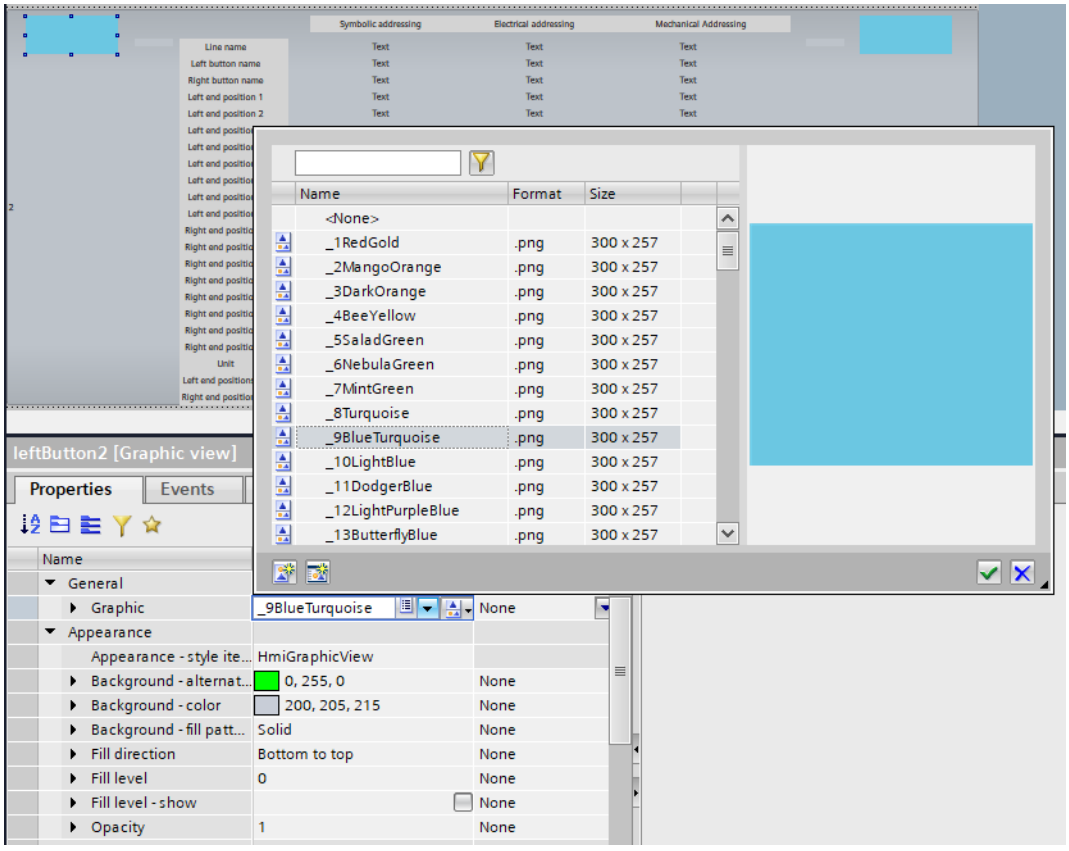


Figure 3-12 Properties

9. Select the end position and set the required color.

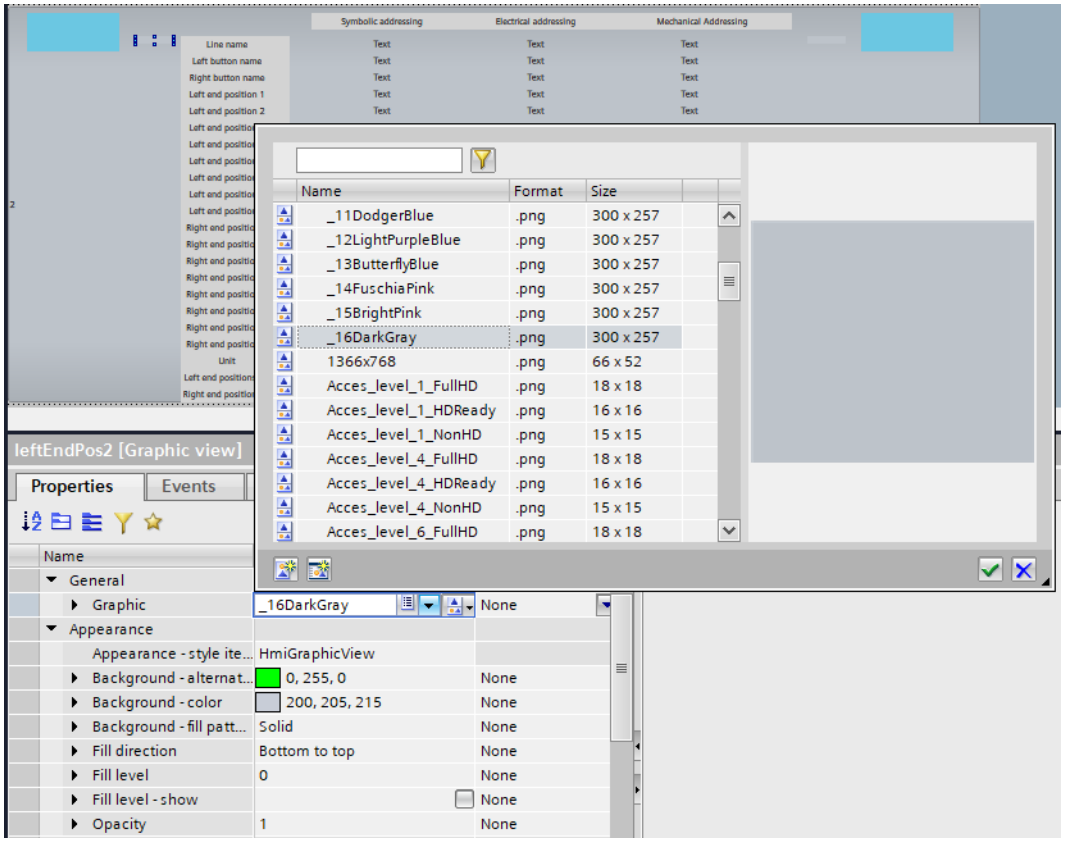


Figure 3-13 Properties

10. Select the end position graphic and edit the value as per the required end position in properties of "numOfLeftEndPos" and "numOfRightEndPos".

Note

Enter the number of end positions between 0 to 8.

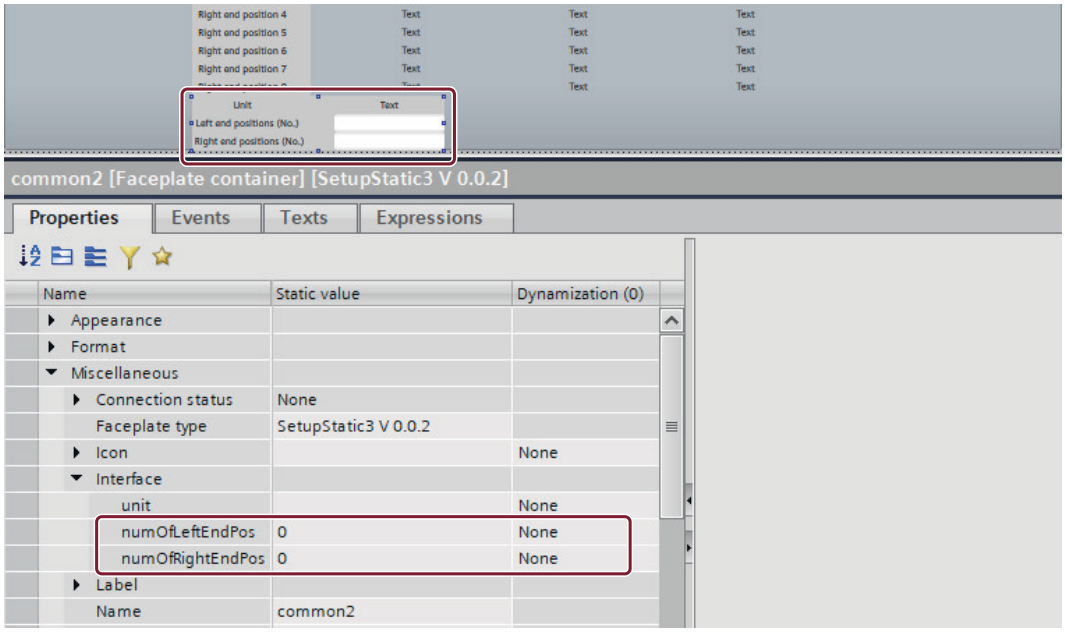


Figure 3-14 Properties

11. Enter the name of line, buttons, left and right end position.

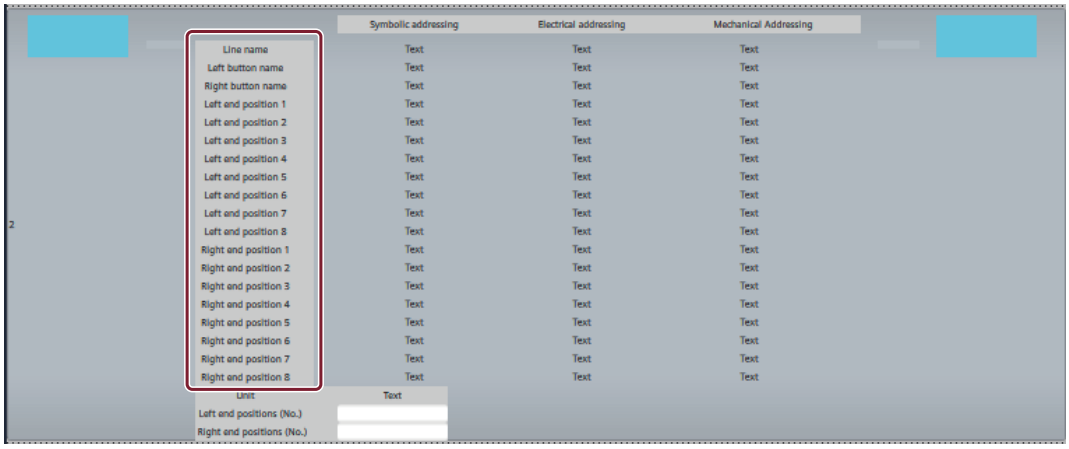


Figure 3-15 Edit text

Note

Maximum number of strings allowed are:

- for line name: 45 (If big area is enabled, then 27)
- for button names: 24 (If big area is enabled or if you enter text in two rows, then 22)
- For end position text: 34 (If big area is enabled, then 18)

12. Enter the symbolic addressing / absolute addressing / mechanical addressing. The Mechanical addressing is an optional addressing method, which is displayed in runtime only when "enableMechAddr" is enabled in PLC blocks.

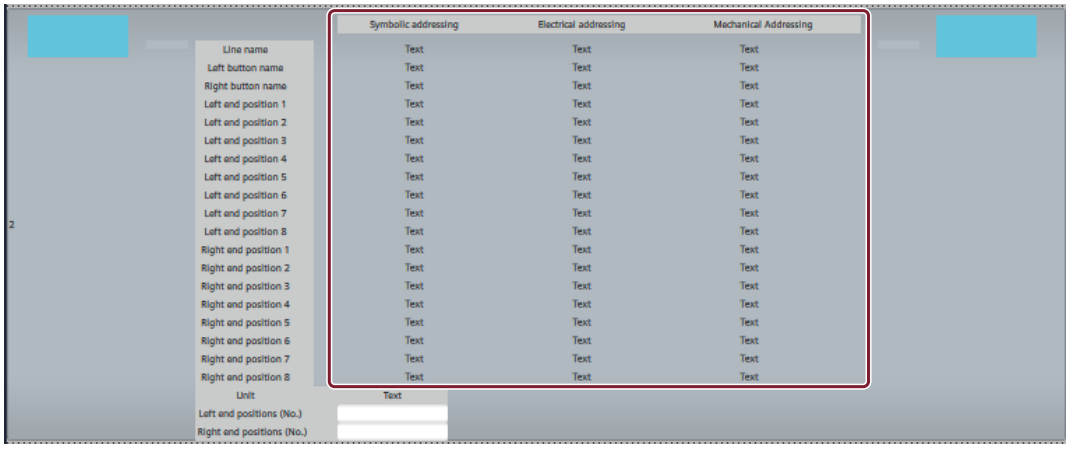


Figure 3-16 **Addressing modes**

13. Select the whole screen, navigate to "Properties" >> "Texts" tab and then export the screen texts. Get the texts translated into required languages.

14. Select the whole screen, navigate to "Properties" >> "Texts" tab, import the translated files back to the screen.

15. In "HMI tags", Navigate to "ManualOperation" >> "LAHP_manualOperation" tag table.

16. Modify the "manualOperationFilePath" tag value to save the JSON files.

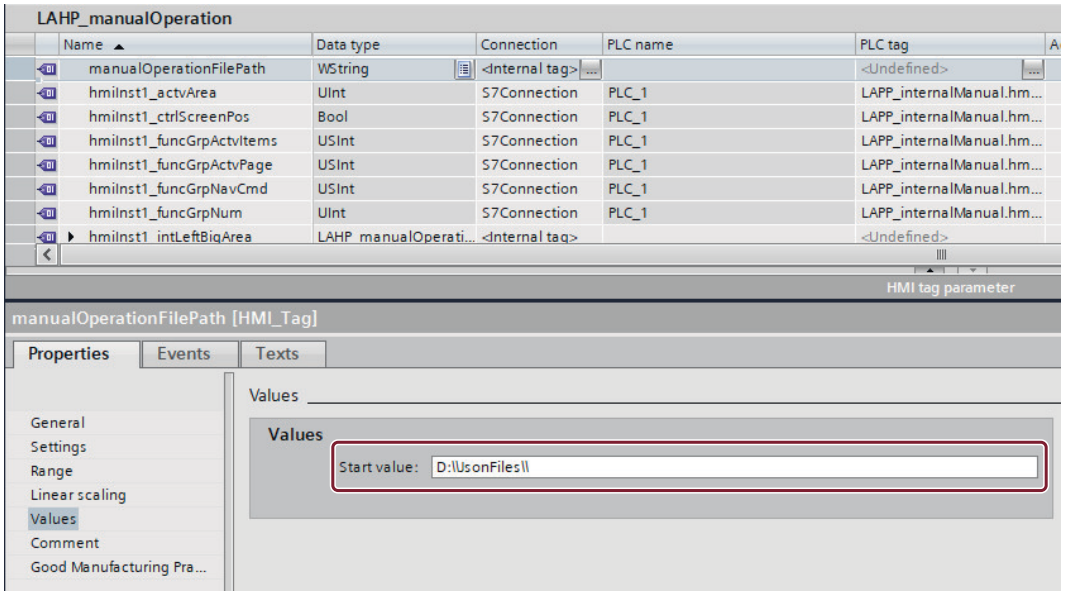


Figure 3-17 LAHP_manualOperation

17. In "Project tree", right-click on HMI device and then navigate to "CHM Addin" >> "Setup" >> click on "Export".
    The "Export to file" pop-up opens.

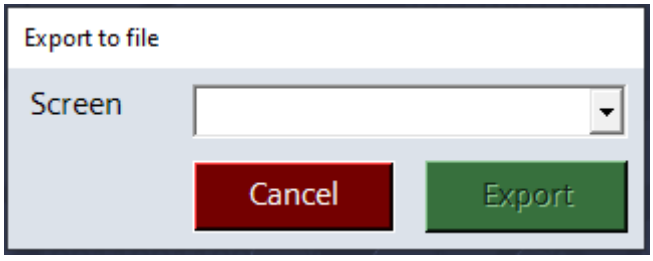


Figure 3-18 Export to file

18. In "Screen" field, select the screen you want to export and then click "Export".
    The JSON files with screen data for all the languages are saved in the path configured.
19. In "LAHP_manualOperation" tag table, modify the "manualOperationFilePath" tag value to the runtime path and load the project to runtime.
20. Copy the JSON files to the runtime path set in Step 18.

3.5.1.2 Configuring navigation to manual operation screen

Procedure

1. In "LAHP_Navigation" screen, select the function group button.
2. Navigate to "Properties > Events > Release > ManualOperation.LAHP_getFuncGrp > funGrpNum" and enter the function group number configured in PLC data block.

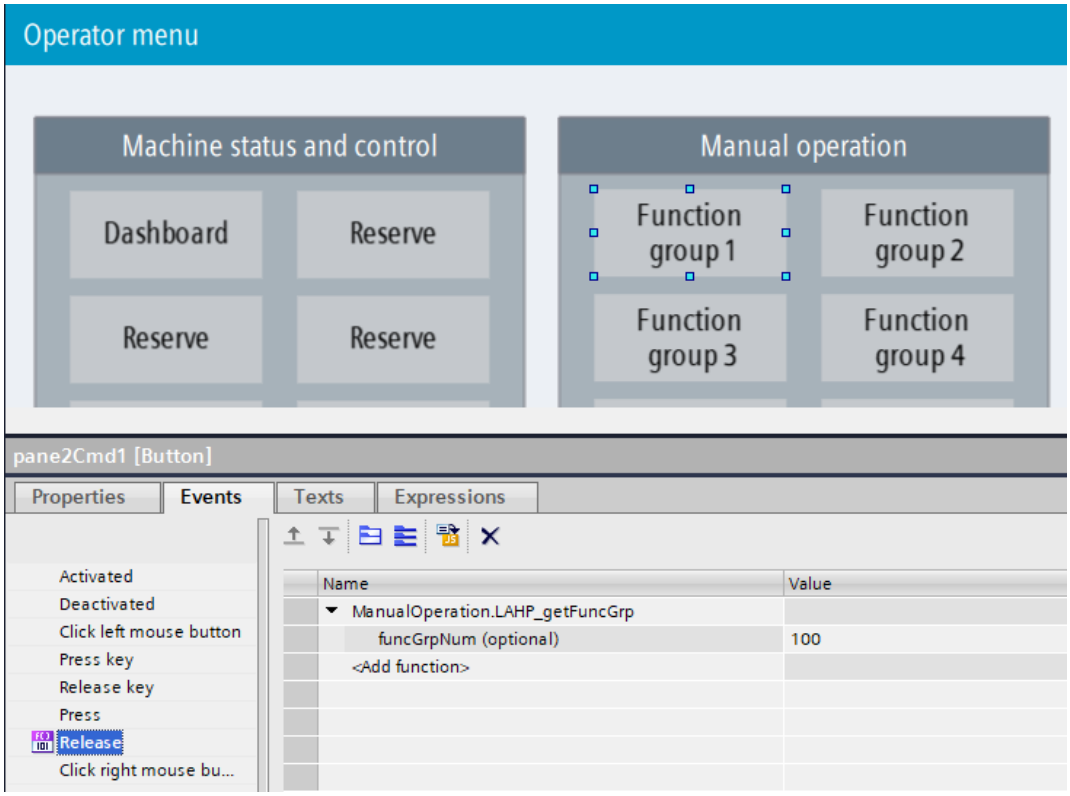


Figure 3-19 Operator menu

3.5.1.3 Configuring function group and area names

You can give the required names for the manual operation screens and areas. You need to add the names for function groups in "LAHP_FunctionGroups" text list.

Note

Do not the edit the name of the "LAHP_FunctionGroups" textlist.

For area names to be displayed on the screen, you need to create a textlist with naming convention of the text list must be "FunctionAreas_(functiongroupID)"

Editing function group textlist

To add the names for function groups:

1. Navigate to “Project tree” >> IPC device >> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAHP_FunctionGroups", in the "Text list entries" window, select the required area to rename and modify the text in “Text” field.

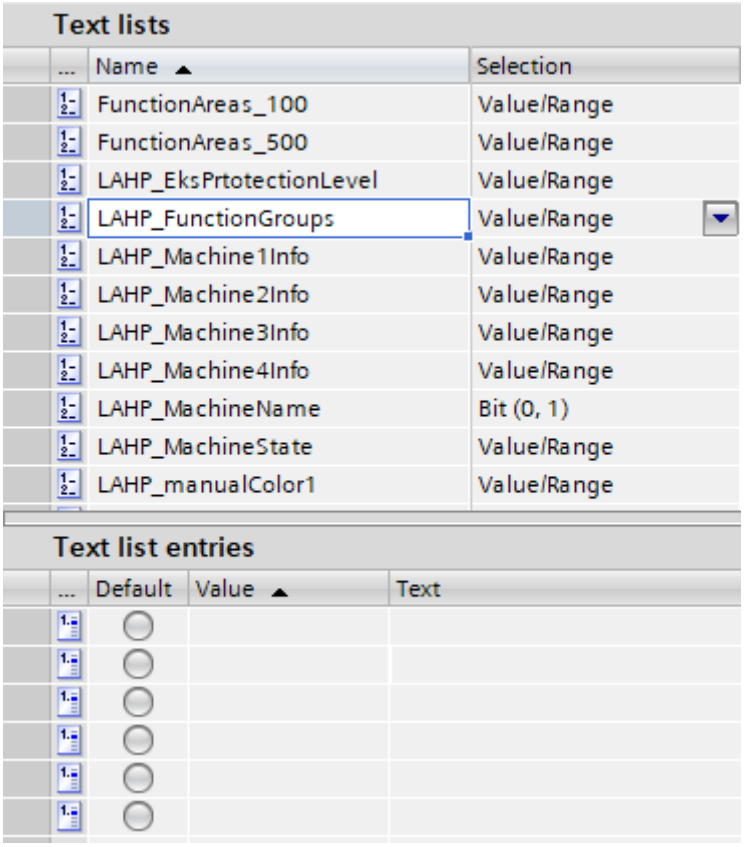


Figure 3-20 LAHP_FunctionGroups

3. Enter the function group ID and function group name in the "Text list entries".
   For example,
   Match the function group ID entered in "NutRunner1.funcGrp.funcGrpNum".

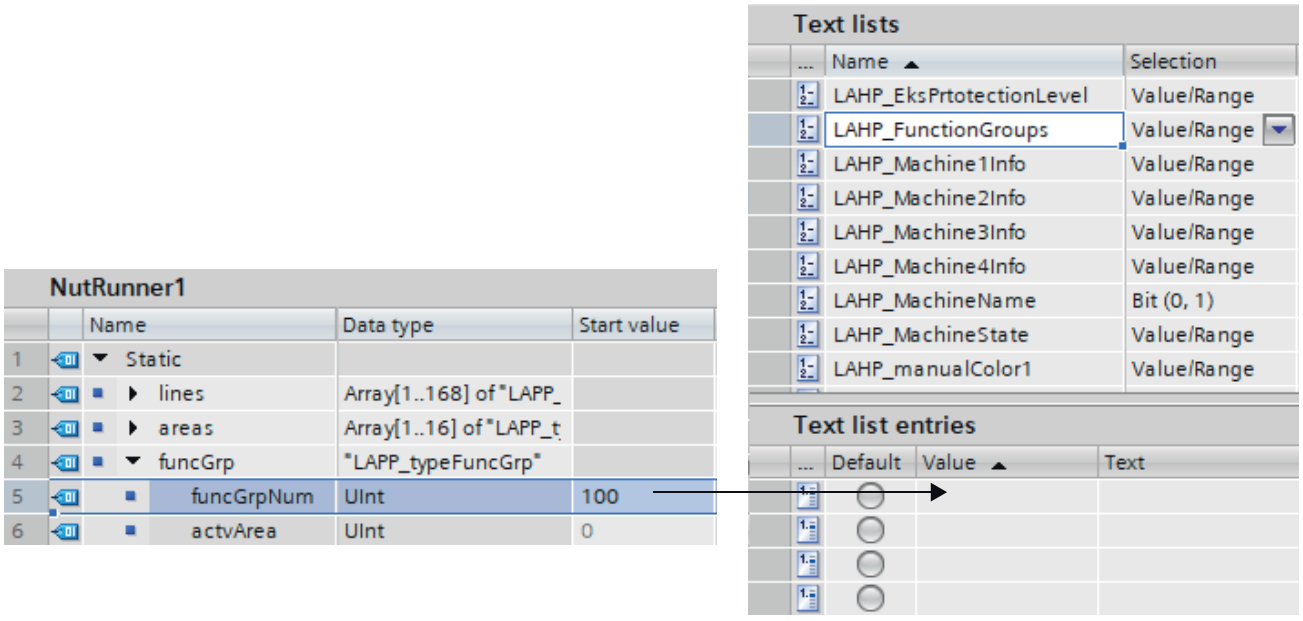


Figure 3-21 Text list entries

The function group ID given in the PLC tags should match the ID entered for text list.

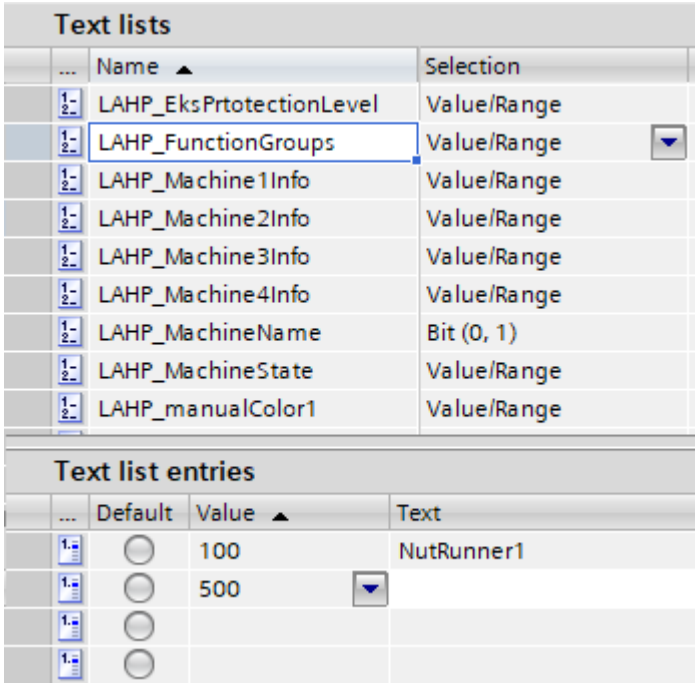


Figure 3-22 LAHP_FunctionGroups

Note

Based on the value in "Value" column the text is fetched during runtime to display on the screen title. Ensure to match the "Value" column with function group number declared in PLC DB.

Creating area text list

To add names for different areas of the function group:

1. Create a new text list. For more information, refer Creating a text list in TIA portal Online help.
2. Name the text list as "FunctionAreas_(functiongroupID).
   For example:
   - For "NutRunner1" function group with function group ID as 1, the areas text list should be named as "FunctionAreas_100".
   - For 2nd function group, the areas text list value is to be named as "FunctionAreas_500".

   Note

   You can define your own function group ID.

3. Add the "Value" and "Text" of the area in "Text list entries".
   Area values are in the incremental order subsequent to the function group ID.
   For example,
   - If the value of "NutRunner1.funcGrp.funcGrpNum" is 100, the area IDs are as follows:
     Area1 - 101
     Area2 - 102
     .
     .
     .
     Area10 - 110
   - If the value of "NutRunner1.funcGrp.funcGrpNum" is 1, the area IDs are as follows:
     Area1 - 2
     Area2 - 3
     .
     .
     .
     Area10 - 11

Text lists

...	Name	Selection
	FunctionAreas_100	Value/Range
	LAHP_EksPrtotectionLevel	Value/Range
	LAHP_FunctionGroups	Value/Range
	LAHP_Machine1Info	Value/Range
	LAHP_Machine2Info	Value/Range
	LAHP_Machine3Info	Value/Range
	LAHP_Machine4Info	Value/Range
	LAHP_MachineName	Bit (0, 1)
	LAHP_MachineState	Value/Range
	LAHP_manualColor1	Value/Range

Text list entries

...	Default	Value	Text
		101	Area1
		102	Area2
		103	Area3
		104	Area4
		105	Area5
		106	Area6
		107	Area7
		108	Area8
		109	Area9
		110	Area10

Figure 3-23 Text list entries

Result

In runtime the manual operation screen name is displayed as configured.

3.5.1.4 Control screen

New control screen

The pre-configured control screens are already available in “IPC (xxxxx)CNC [WinCC Unified PC]” >> "Screens" >> "SIEMENS" >> “ManualOperation” >> “SampleControlScreens”.
Naming conventions of the control screen is ControlScreen_y_x, where y represents the function group number and x represents the line number.

Note

- For the screens to work as intended, naming convention must be followed when naming the control screens.
- You can add only one control screen for one line.

Procedure

To create control screen, follow the below steps:

1. Create a control screen or copy the pre-configured control screen.
2. Name the control screen with respective function group number and line number.

For example,

- To add a control screen for line 18 which belongs to function group number 100, you need to create a control screen with the name "ControlScreen_100_18".
- To add a control screen for line 100 which belongs to function group number 500, create a control screen with the name "ControlScreen_500_100".

You can also modify the height and width of the control screen assigned for each line.

1. Select the ControlScreen_y_x.
2. Navigate to “Properties” >> “Size and position”.
3. Populate the “Static value” field for “Height” and “Width”.

Note

- The height of the control screen can be a desirable value.
- If the width of the control screen exceeds the width of the device resolution, there appears a side-scroll bar. Hence it is suitable to assign the width of the control screen equal to the device resolution or slightly lesser.

3.5.2 Configuring as faceplate

Procedure

To create faceplate:

1. Create tag and link to PLC tag to trigger the block execution. For example, create a tag "M10_0" with PLC address as "Tag_1".

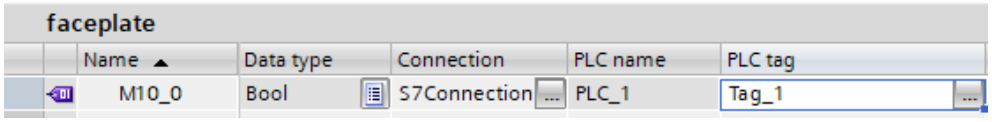


Figure 3-24 Tag table

Note

Ensure to use the same PLC tag for block execution. For more information, refer to Manual Operation as faceplate (Page 49).

2. In automotive library, navigate to "HMI controls" and select "LAHP_manualOperationMain_(required resolution)" faceplate from the required resolution folder and add it to the required screen.

Note

You can view the sample faceplate created in "manualOperationFaceplate" screen. You can copy the faceplate and do the configuration.

3. Select the faceplate, navigate to "Properties > Properties" tab and configure the following:
   – In "Format > Size-fit", select "Fit window to screen"

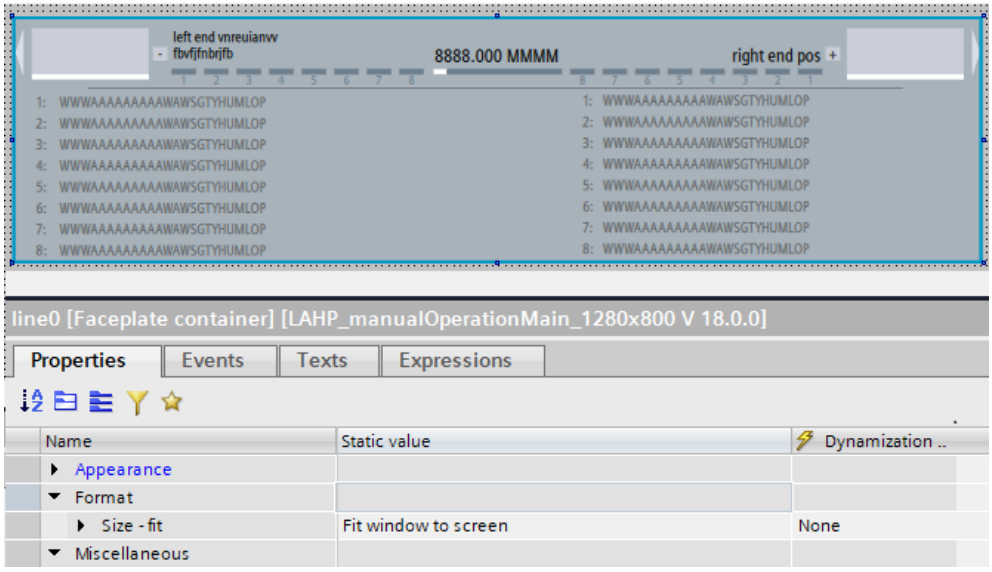


Figure 3-25 Properties

– In "Miscellaneous > Interface", interface the value for the highlighted properties from "LAHP_manualOperationFaceplate" tag table.

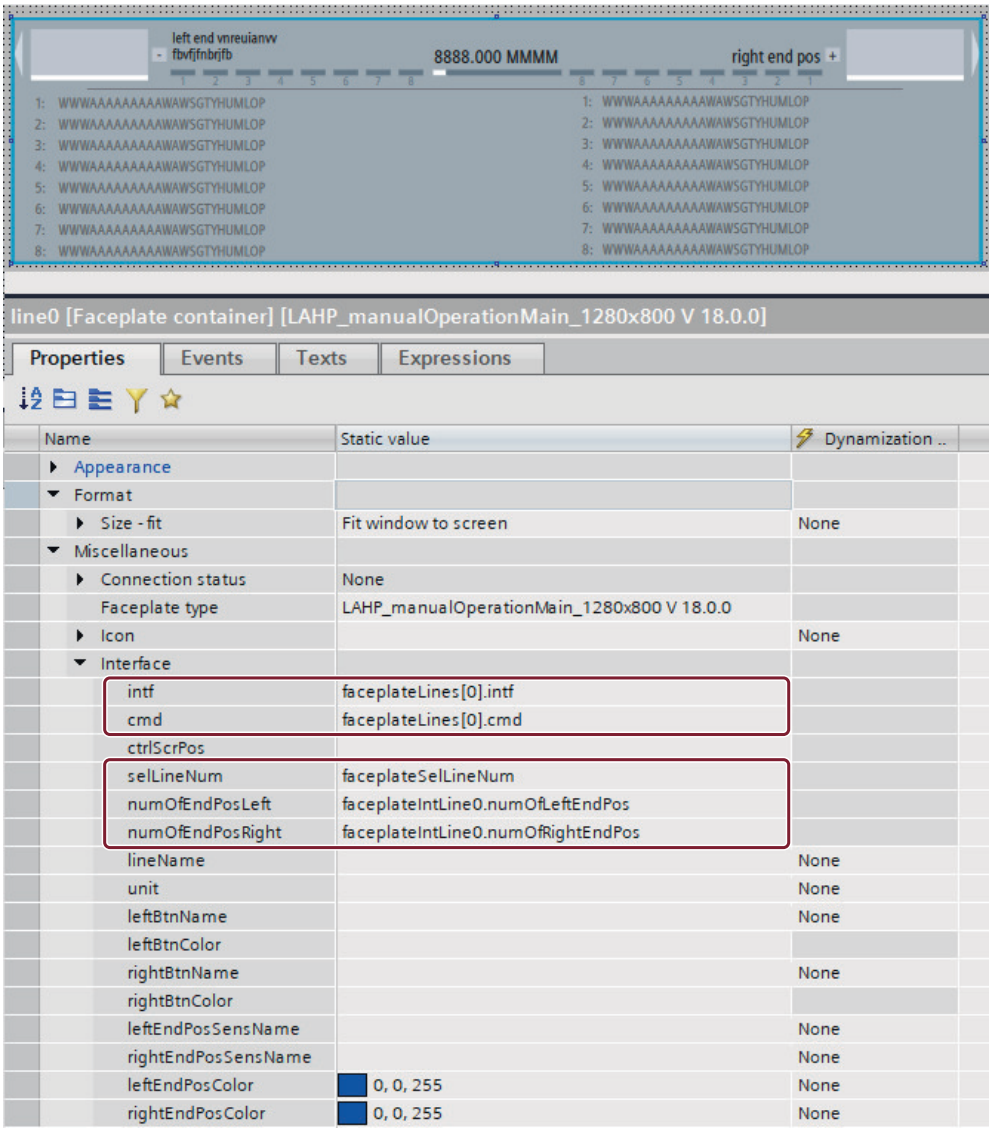


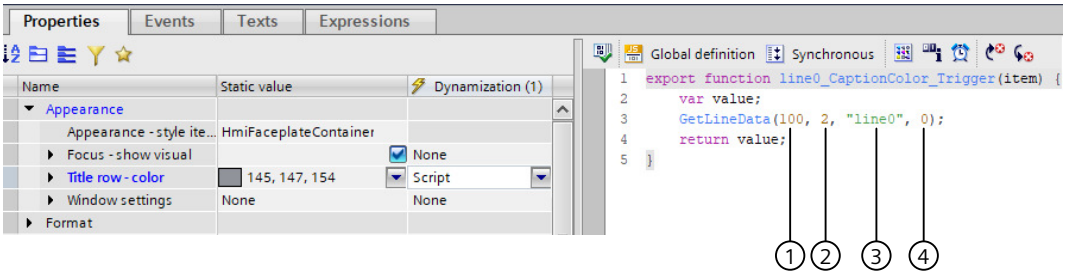
Figure 3-26 Properties

Note

- You can create six faceplates for each screen.
- Interface the respective line parameter for each screen:
  - First line: "faceplateLines[0]"
  - Second line: "faceplateLines[1]"
    .
  - Sixth line: "facplateLines[5]"
- For "Intf", "cmd", "numOfEndPosLeft", and "numOfEndPosRight" property interface the respective faceplate parameter.
- For "selLineNum" property for all the lines interface "faceplateSelLineNum" tag.

– Enter the name for the faceplate in "Properties > Miscellaneous > Name".

4. Select the line, and in any of the property > "Dynamization" column set the value to "Script" and add the script as shown:



1 Function group number to which the line belongs.

2 It is the line number configured in the function group. For example, 2 indicates that, you want to view the data of the second line from the function group (function group number 100)

3 The name for faceplate. Ensure to enter the same name entered in "Properties > Miscellaneous > Name".

4 The line index number of the faceplate on the screen. For example, for

- first faceplate on the screen, enter the number 0
- second faceplate on the screen, enter the number 1
  .
  .
- sixth faceplate on the screen, enter the number 5

Figure 3-27 Script

Note

Ensure to enter the respective line details in the script to avoid errors.

5. Click on icon, the "Add trigger" window opens.

6. In "Trigger" field, select "Tags" and add the following triggers as shown:

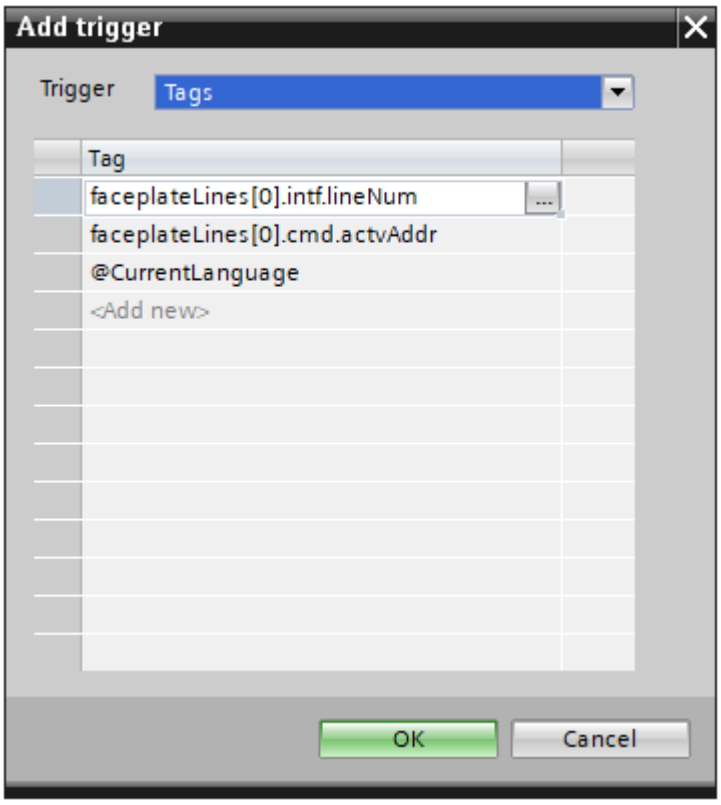


Figure 3-28 Add trigger

Note

- Add the respective line "faceplateLines[X].intf.lineNum" and "faceplateLines[X].cmd.actvAddr" tags. X- line index number
- Select the "@CurrentLanguage" tag from "Default tag table".

7. Select the whole screen and navigate to "Properties > Events" tab.
   – In "Loaded", configure as shown to ensures that upon opening the screen, the respective "LAPP_manualOperationFaceplate" block execution happens.

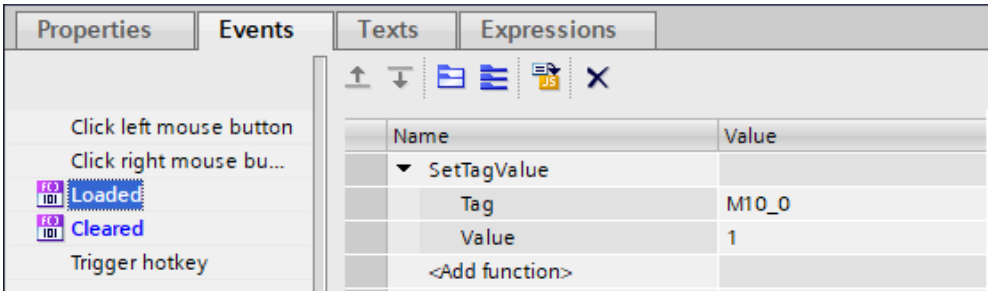


Figure 3-29 Events

– In "Cleared", configure as shown

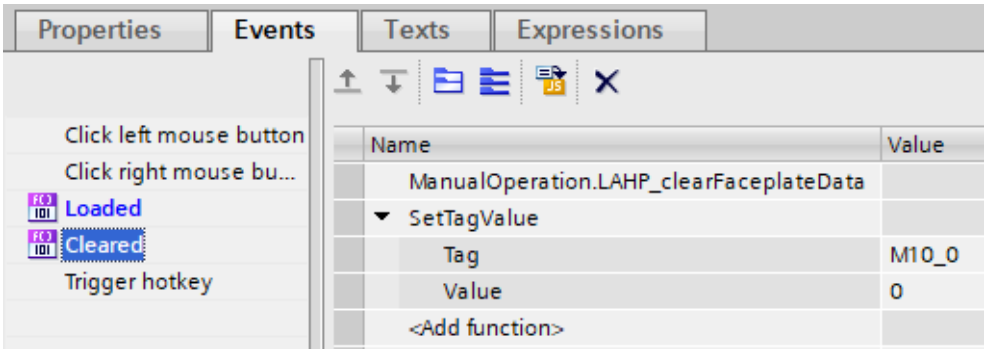


Figure 3-30 Events

Note

Ensure to interface the same tag to "LAPP_manualOperationFaceplate" block, for the faceplate to work as intended.

Note

- Expand ON and Expand visible parameters are available for Manual Operation as faceplate. OEM needs to configure as per the requirement.
- You can also use indirect addressing for the faceplate to minimize the number of power tags in HMI.

3.6 HMI tags for LAHP_ToolLifeMonitor

3.6.1 Editing graphics

Procedure

You can edit the tool graphics displayed in runtime for each tool. To change the graphics:

1. Navigate to “Project tree” >> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Navigate to "Graphic lists" window.

3. Select "LAHP_ProcessDashboardToolList" from the "Graphic lists" window.

Graphic lists

...	Name	Selection	Comment
	LAHP_MachineState1	Value/Range	
	LAHP_MachineState2	Value/Range	
	LAHP_ManualOperationActive	Value/Range	
	LAHP_ProcessDashBoardToolList	Value/Range	
	LAHP_ShiftPlanningActive	Value/Range	
	LAHP_ToolLifeActive	Value/Range	
	LAHP_ToolLifeState	Value/Range	
	LAPP_RFIDActiveStatus	Bit (0, 1)	
	LAPP_RFIDError	Bit (0, 1)	
	LAPP_RFIDWarning	Bit (0, 1)	

Graphic list entries

...	Default	Value	Graphic name	Graphic
		1	to_mill_machine_1	
		2	to_mill_magazine_1	
		3	to_poly_3d_tool_110_down_1	
		4	to_poly_3d_tool_111_down_1	
		5	to_poly_3d_tool_121_down_1	
		6	to_poly_3d_tool_155_down_1	
		7	to_poly_3d_tool_156_down_1	
		8	to_poly_3d_tool_157_down_1	

Figure 3-31 Tool life graphic list

4. In "Graphic list entries " window, select the required tool to change the graphic.

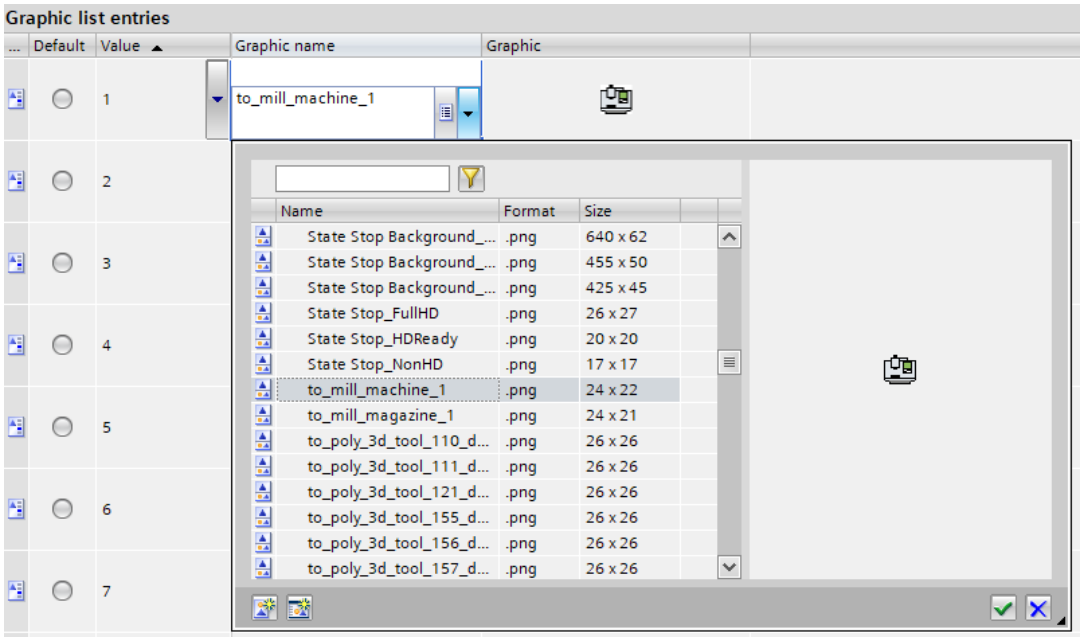


Figure 3-32 Graphic list entries

5. Click , to add a new graphics
   or
   Select the required graphic from the list.
6. Click .

Note

- The graphics set here for the tool is displayed in "LAHP_Processdashboard" and "LAHP_Toollifemonitor" screens.
- Ensure to assign the correct graphic against the tool index (in data block) mentioned in the "Value" column.

Result

Graphic is updated for the selected tool.

3.6.2 Configuring text lists

You can configure the required tool name and information in "LAHP_ToolName", "LAHP_ToolInfo" and "LAHP_ToolTypes" text list as follows:

Procedure to configure "LAHP_ToolName"

1. In Project tree, navigate to “Devices” >> IPC device >> “Text and graphic lists”.
2. Select “LAHP_ToolName” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default tools added.

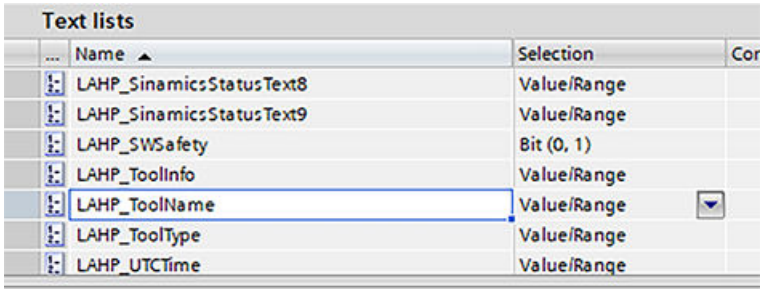


Figure 3-33 LAHP_ToolName

3. Add the names for the tools in "Text list entries".

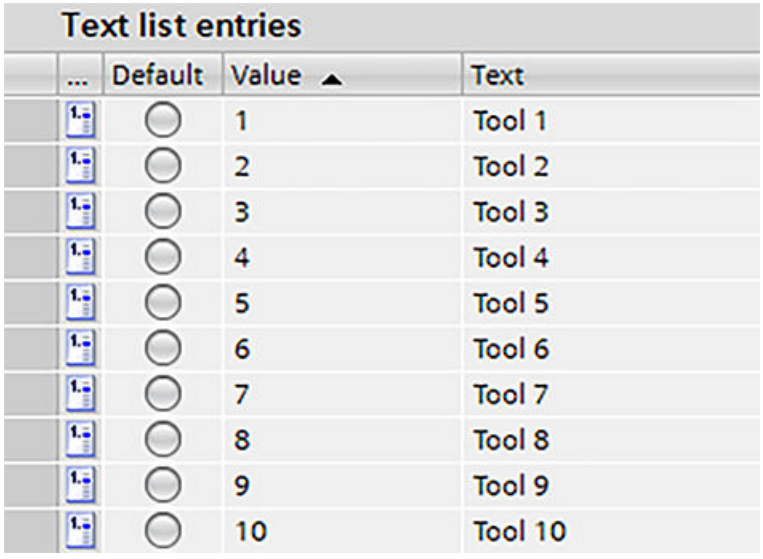


Figure 3-34 Text list entries (LAHP_ToolName)

Procedure to configure "LAHP_ToolInfo"

1. In Project tree, navigate to “Devices” >> IPC device >> “Text and graphic lists”.
2. Select “LAHP_ToolInfo” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default tools added.

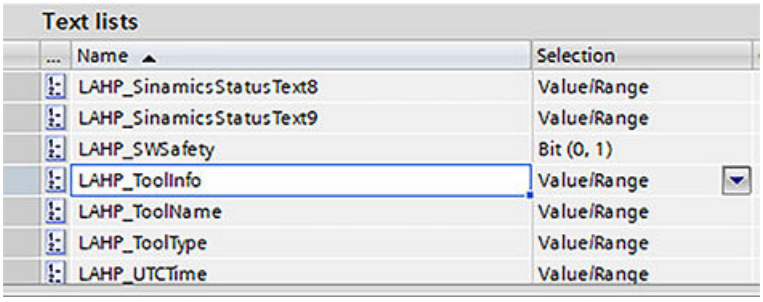


Figure 3-35 LAHP_ToolInfo

3. Add the required information for the tools in "Text list entries".

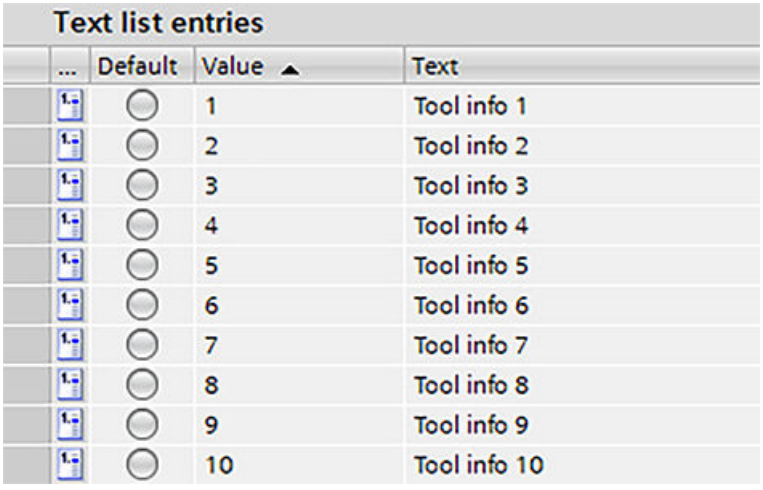


Figure 3-36 Text list entries (LAHP_ToolInfo)

Procedure to configure "LAHP_ToolType"

1. In Project tree, navigate to “Devices” >> IPC device >> “Text and graphic lists”.
2. Select “LAHP_ToolType” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default tools added.

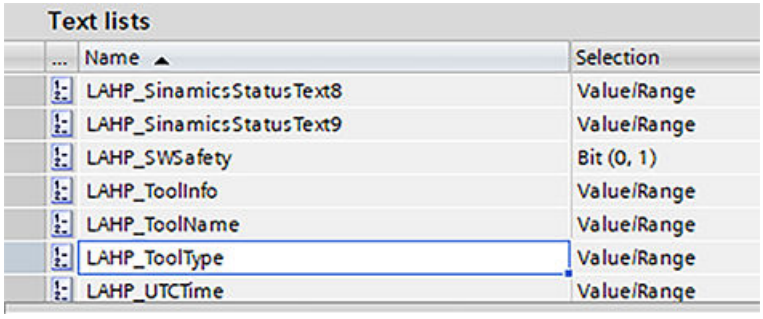


Figure 3-37 LAHP_ToolType

3. Add the required units for the tools in "Text list entries".

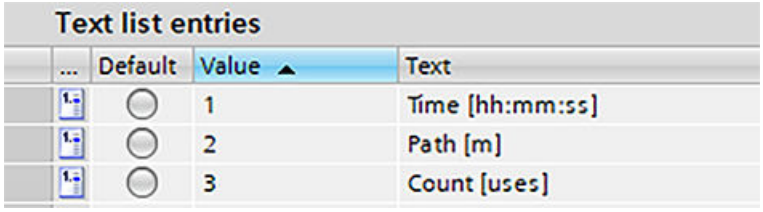


Figure 3-38 Text list entries (LAHP_ToolType)

Note

You can modify the units depending on the requirement.

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

3.7 Process dashboard

3.7.1 Configuring text lists for LAHP_ProcessDashboard

Text list for production status

You can configure the required status of the stack lights in "LAHP_StackLightStatus" text list as follows:

1. In Project tree, navigate to “Devices” >> IPC device >> “Text and graphic lists”.
2. Select “LAHP_StackLightStatus” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default status added.

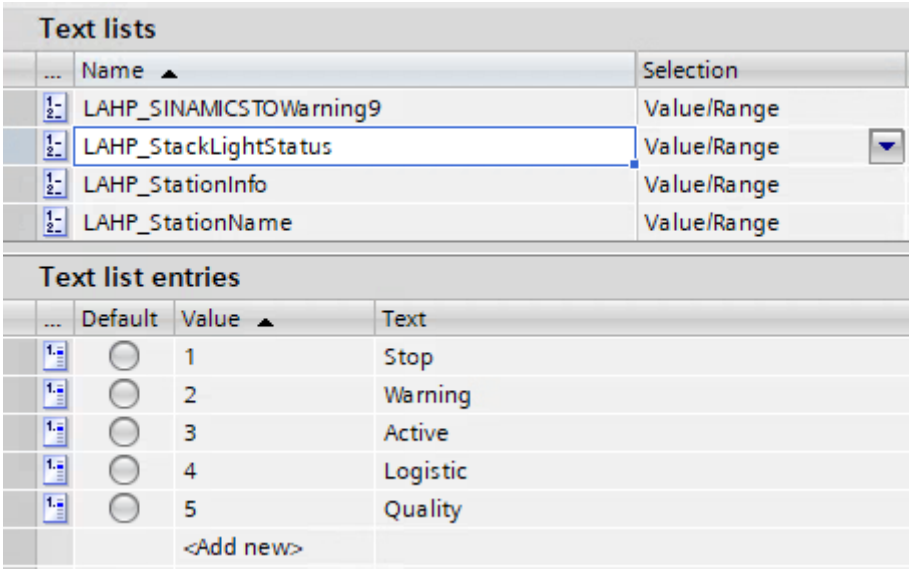


Figure 3-39 LAHP_StackLightStatus

3. Add the required number of stack light status in "Text list entries".

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

Note

The above steps can be followed to configure the text lists for

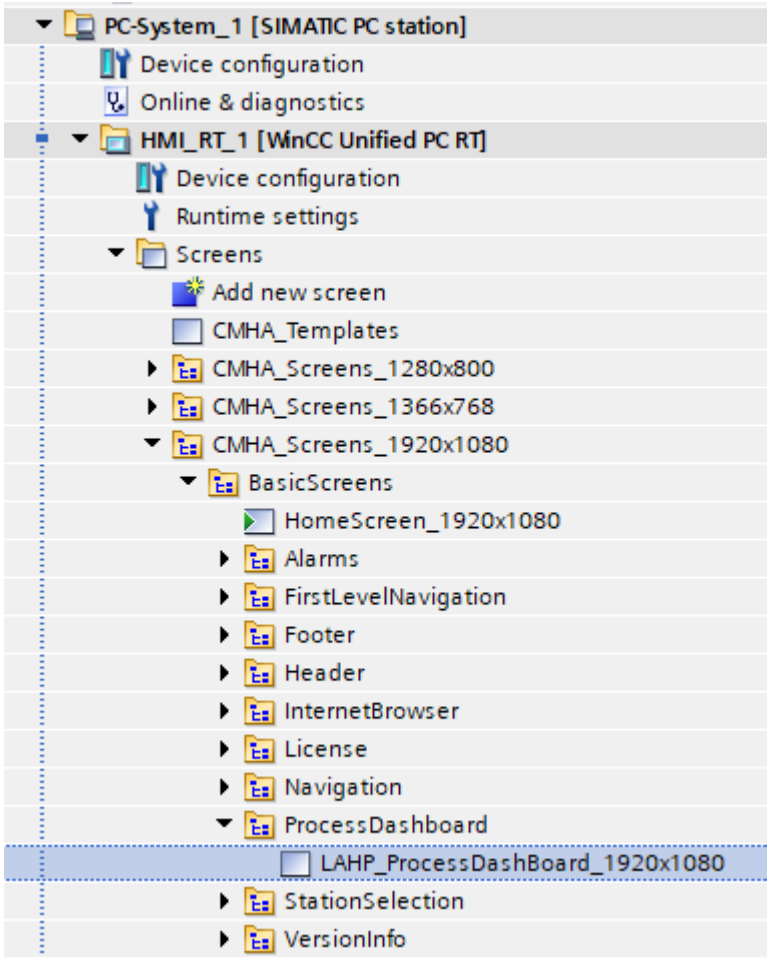
- Quality metrics (LAHP_WidgetQualityStatic)
- Edge tool (LAHP_ToolWidgetEdgeTool)
- Sister tool (LAHP_WidgetToolSisterTool)

3.7.2 Process dashboard as faceplate

3.7.2.1 Creating the faceplate

Procedure

1. Create HMI tags for each widget. For more information, refer HMI tags for faceplate (Page 142).
2. Navigate to the HMI device > “Screens". Select the screen based on the requirement.



Note

You can select the type of device based on the requirement. It is categorized into

- Full HD CNC
- Full HD PLC
- HD Ready
- Non-HD CNC
- Non-HD PLC devices.

3. Navigate to “Libraries”>> “Project library” >> “Types” and drag the faceplate of the required resolution based on the device to the screen.

4. Select the faceplate and then navigate to "Properties" tab >> "Miscellaneous" >> "Interface" and configure the interface properties.
5. To modify the stacklight color for the production status, select the "Status" widget, navigate to "Properties" tab >> "Miscellaneous" >> "Interface" and configure the colors as per the requirement.

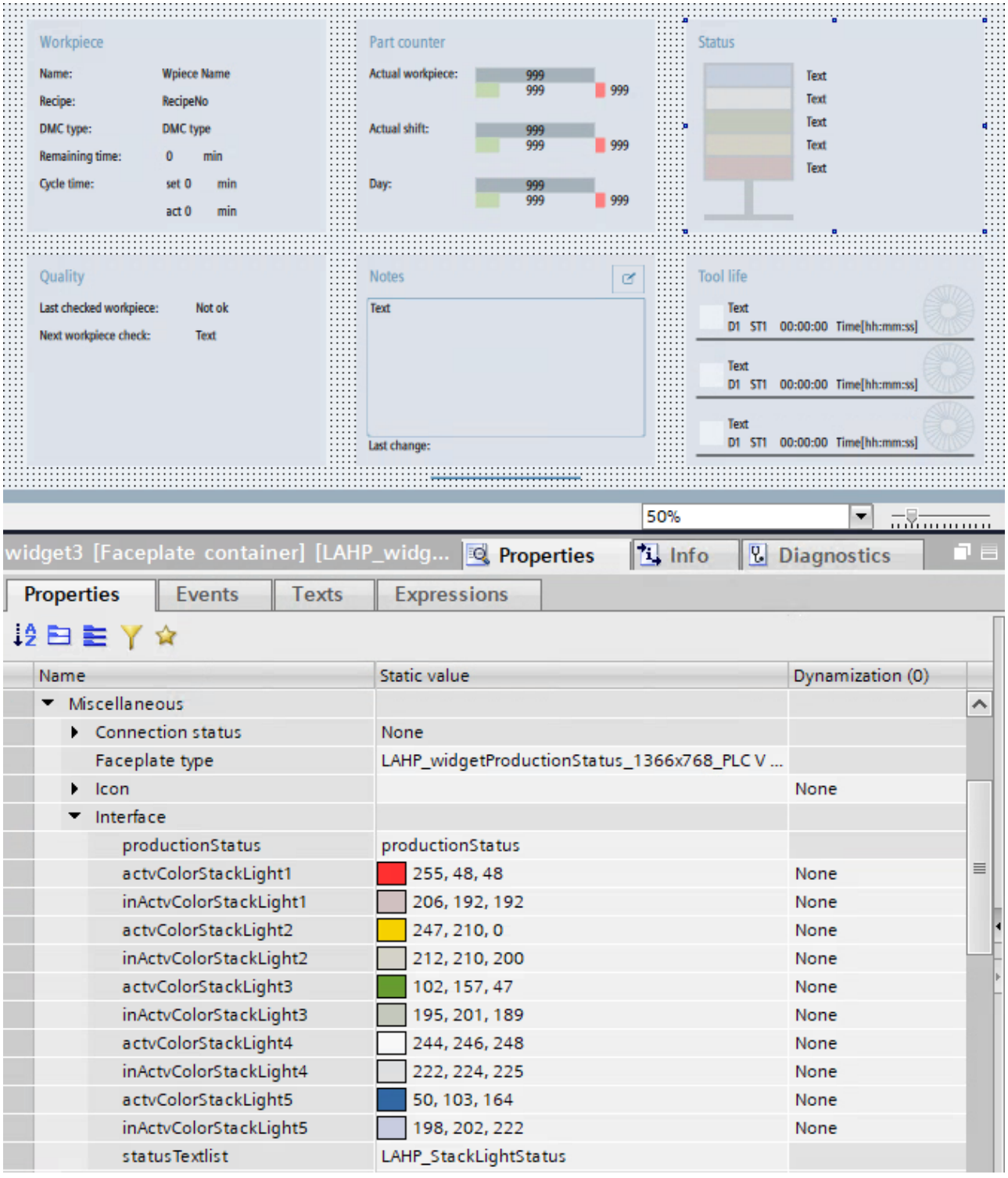


Figure 3-40 Process dashboard interface

Result

The faceplate for process dashboard is created.

3.7.2.2 HMI tags for faceplate

Procedure

Perform the following steps to view/add values to the HMI tags (workpiece, part details, production status and quality) for LAHP_ProcessDashboard:

1. Navigate to the “IPC (xxxx) CNC [WinCC unified PC RT]”>> “HMI tags” and click “Add new tag table”.
2. Navigate to the tag table on the right side and click “Add new”.

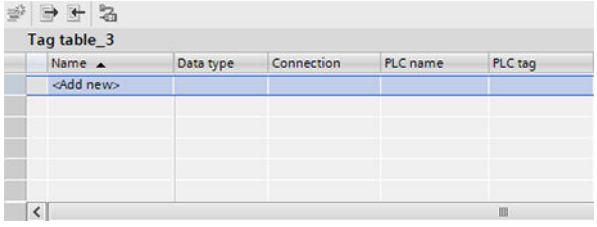


Figure 3-41 Tag table

3. In the “PLC tag” field browse and select the required datatype.

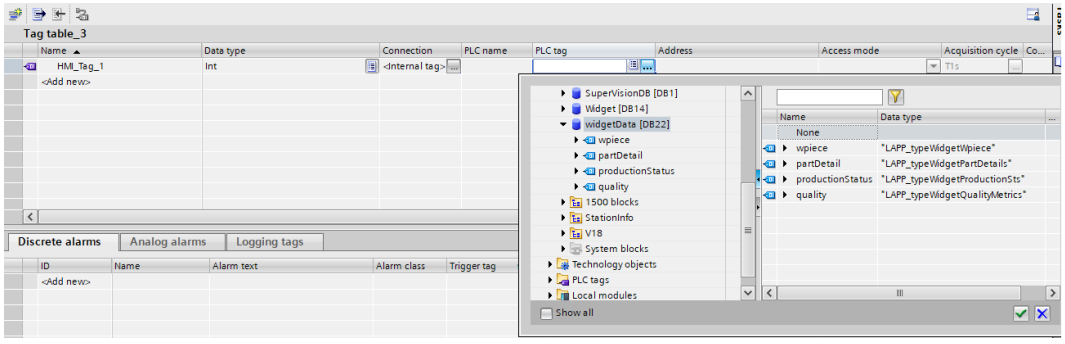


Figure 3-42 Tag selection

4. Click ✓. The required HMI tag is created.

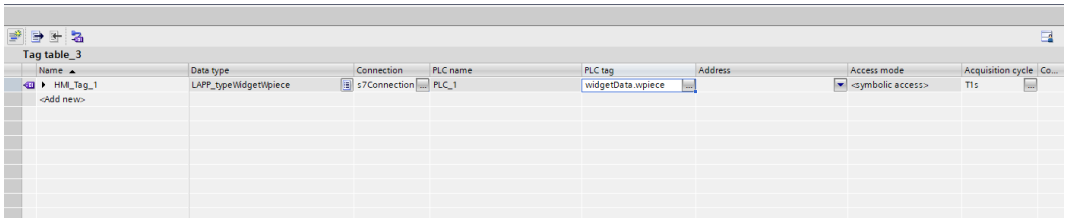


Figure 3-43 Tag created

Perform the following steps to view/add values to the HMI tags (notes) for LAHP_ProcessDashboard:

1. Navigate to the “IPC (xxxx) CNC [WinCC unified PC RT]”>> “HMI tags” and click “Add new tag table”.
2. Navigate to the tag table on the right side and click “Add new”.

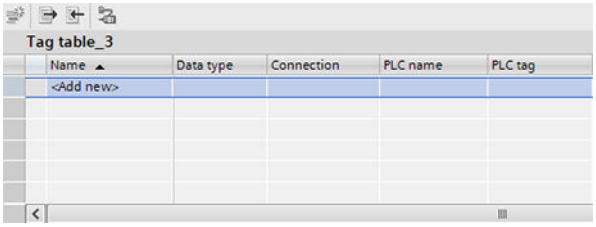


Figure 3-44 Tag table

3. In the “Data type” field browse and select "LAPP_typeWidgetNotes".

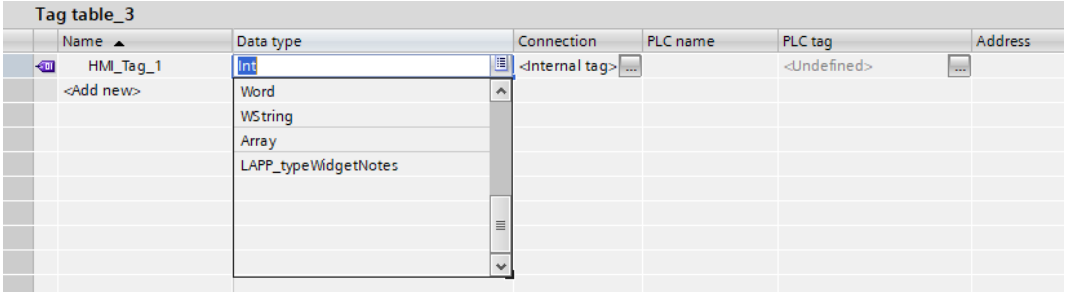


Figure 3-45 Tag selection

4. The required HMI tag is created.

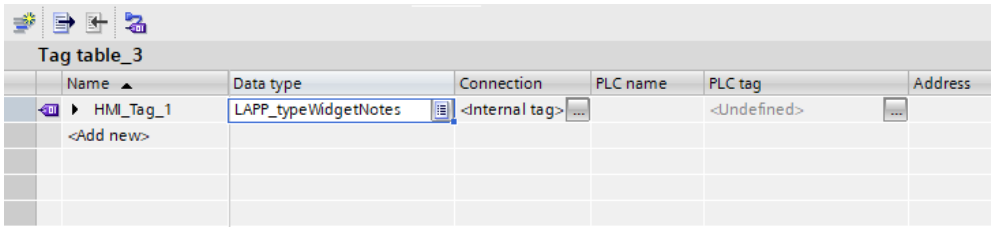


Figure 3-46 Tag selected (notes)

5. Navigate to "Properties" >> "Settings" and enable "Persistence for internal tags" to retrieve the value when you turn-off the runtime.

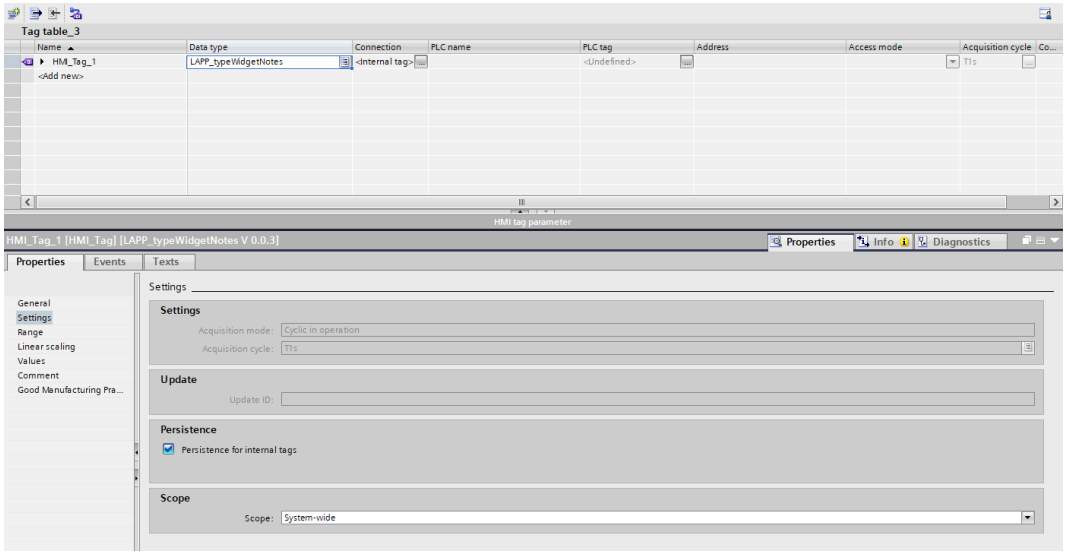


Figure 3-47 Configure properties

3.8 SINAMICS

3.8.1 Configuring text list for SINAMICS EPOS

Procedure

You can configure the required number of axes for status, SI status and Alarm in "LAHP_SINAMICSEPOSAxis" text list as follows:

1. In Project tree, navigate to “Devices” >> "IPC device" >> “Text and graphic lists”.
2. Select “LAHP_SINAMICSEPOSAxis” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default axes added.

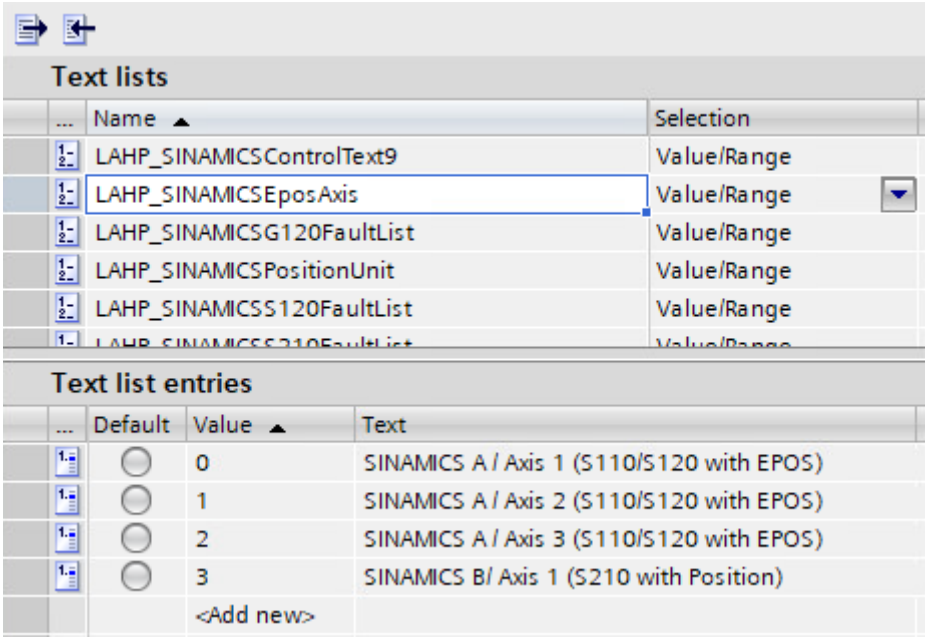


Figure 3-48 LAHP_SINAMICSEPosAxis

3. Add the required number of axes in "Text list entries".

You can configure the required number of axes for positioning in "LAHP_SINAMICSAxis" text list as follows:

1. In Project tree, navigate to “Devices” >> "IPC device" >> “Text and graphic lists”.
2. Select “LAHP_SINAMICSAxis” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default axes added.

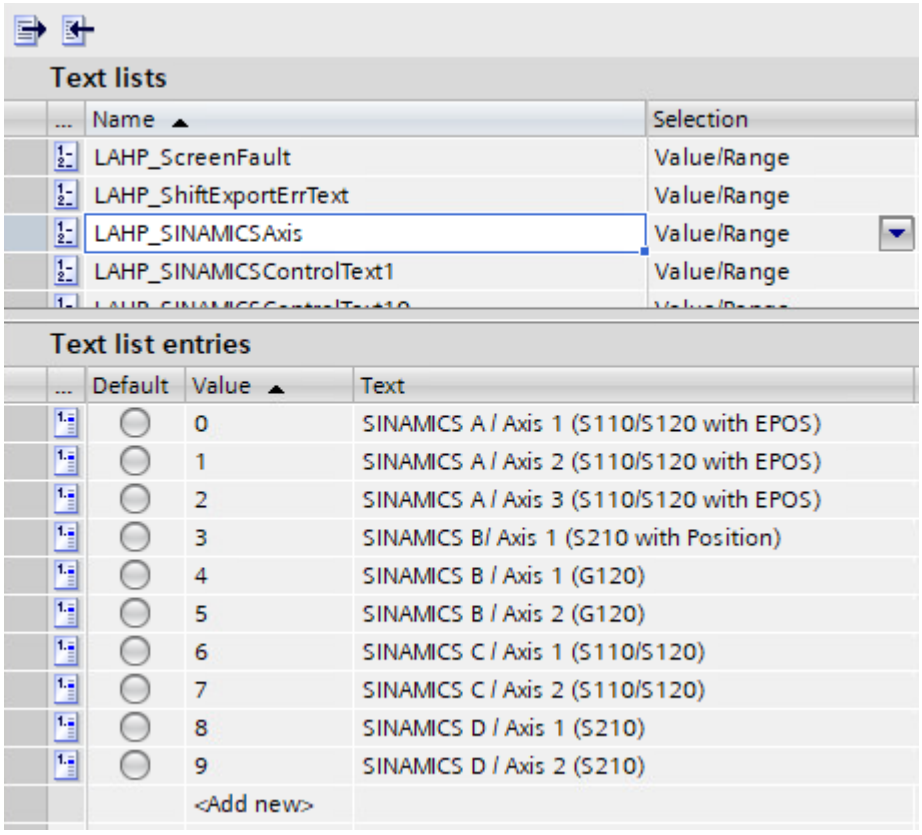


Figure 3-49 LAHP_SINAMICSAxis

3. Add the required number of axes in "Text list entries".

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

3.8.2 Configuring text lists for SINAMICS Technology Object

Procedure

You can configure the required number of axes in "LAHP_SINAMICSTOAxis" text list as follows:

1. In Project tree, navigate to “Devices” >> IPC device >> “Text and graphic lists”.
2. Select “LAHP_SINAMICSTOAxis” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default axes added.

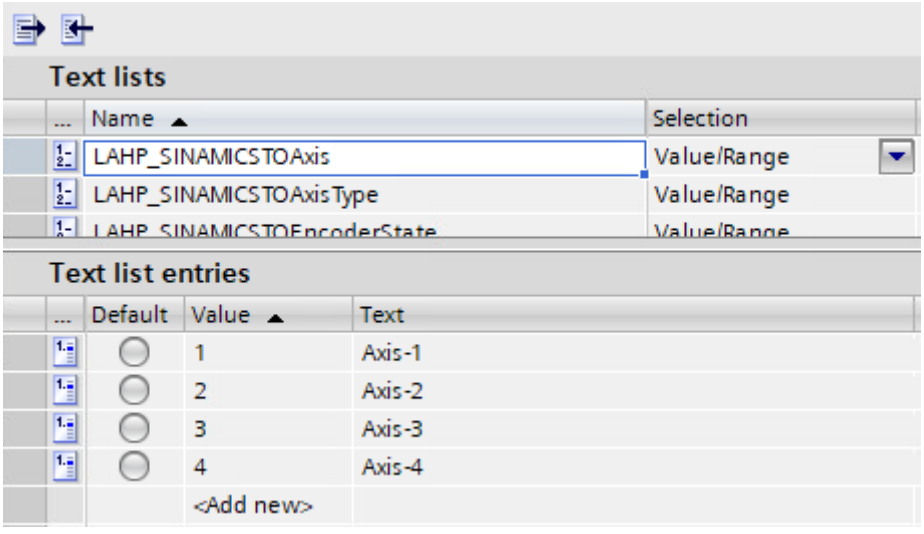


Figure 3-50 LAHP_SINAMICSTOAxis

3. Add the required number of axes in "Text list entries".

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

3.9 Configuring the text list for Motor starter

The designations of the motor starter devices are configured in the WinCC text list "LAHP_Motorstarter". Each motor starter to be diagnosed must have an entry in the text list.

Procedure

You can configure the required motor starter device in "LAHP_Motorstarter" text list as follows:

1. Navigate to “Project tree” >> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAHP_Motorstarter" from the "Text lists" window.

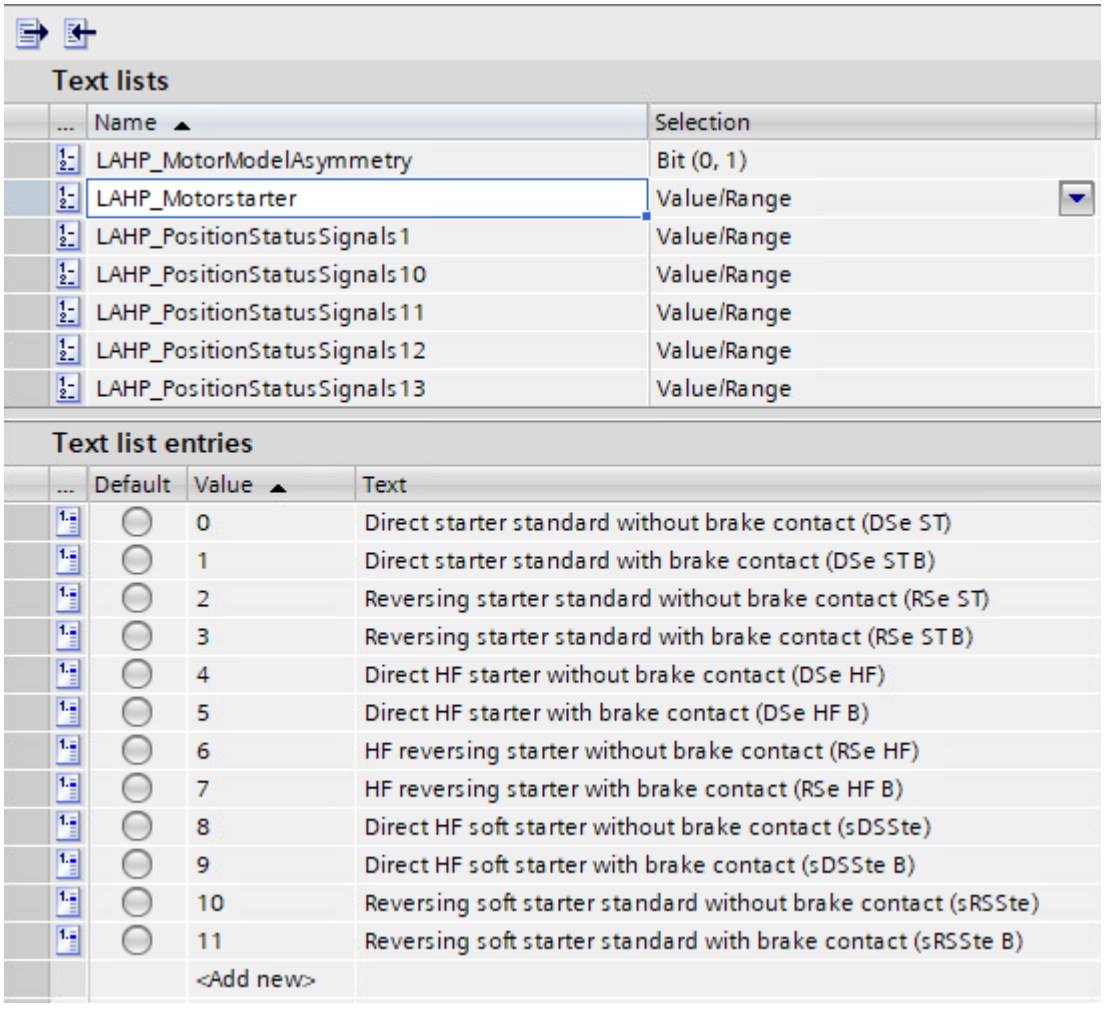


Figure 3-51 LAHP_Motorstarter

3. In "Text list entries" window, select the required value and modify the text as per the requirement.

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

3.10 Configuring text list for LAHP_RFID

Description

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

The "LAHP_IdentDeviceName" text list has the following structure:

Table 3-1 Text list for the designations of the ident devices

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

Procedure

You can configure the required motor starter device in "LAHP_IdentDeviceName" text list as follows:

1. Navigate to “Project tree” >> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAHP_IdentDeviceName" from the "Text lists" window.

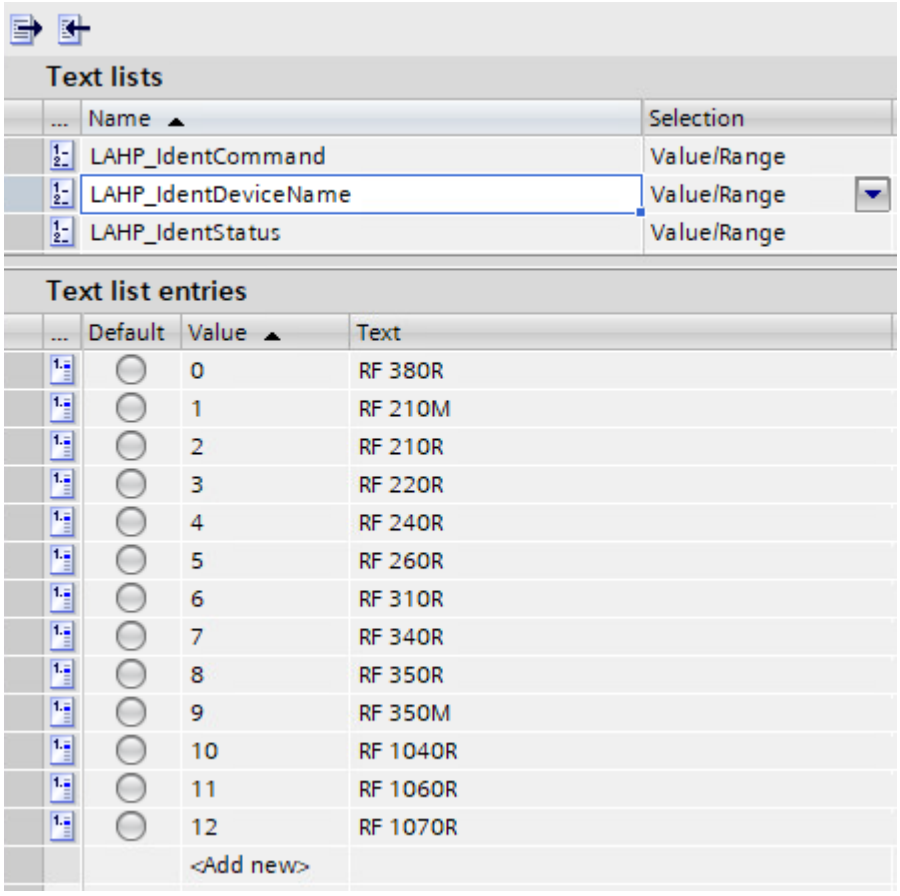


Figure 3-52 LAHP_IdentDeviceName

3. In "Text list entries" window, select the required value and modify the text as per the requirement.

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

3.11 LAHP_EKS

3.11.1 Configuring tag for LAPP_EKS

The corresponding **eksAdptrIndex** (input parameter of the FB **LAPP_EKS**) has to be entered in the WinCC tag **LAPP_internalEKSData.hmi[1].EKSAdptrIndex** for the EKS adapter to be displayed.

3.11.2 Editing graphics

You can modify the graphic lists to modify the default key graphics. For different resolutions use the respective lists:

- "LAHP_User_FullHD"
- "LAHP_User_HDReady"
- "LAHP_User_NonHD"

Procedure

You can edit the key graphics displayed in runtime for each key. To change the graphics:

1. Navigate to “Project tree” >> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Navigate to "Graphic lists" window.

3. Select "LAHP_User_FullHD" from the "Graphic lists" window.

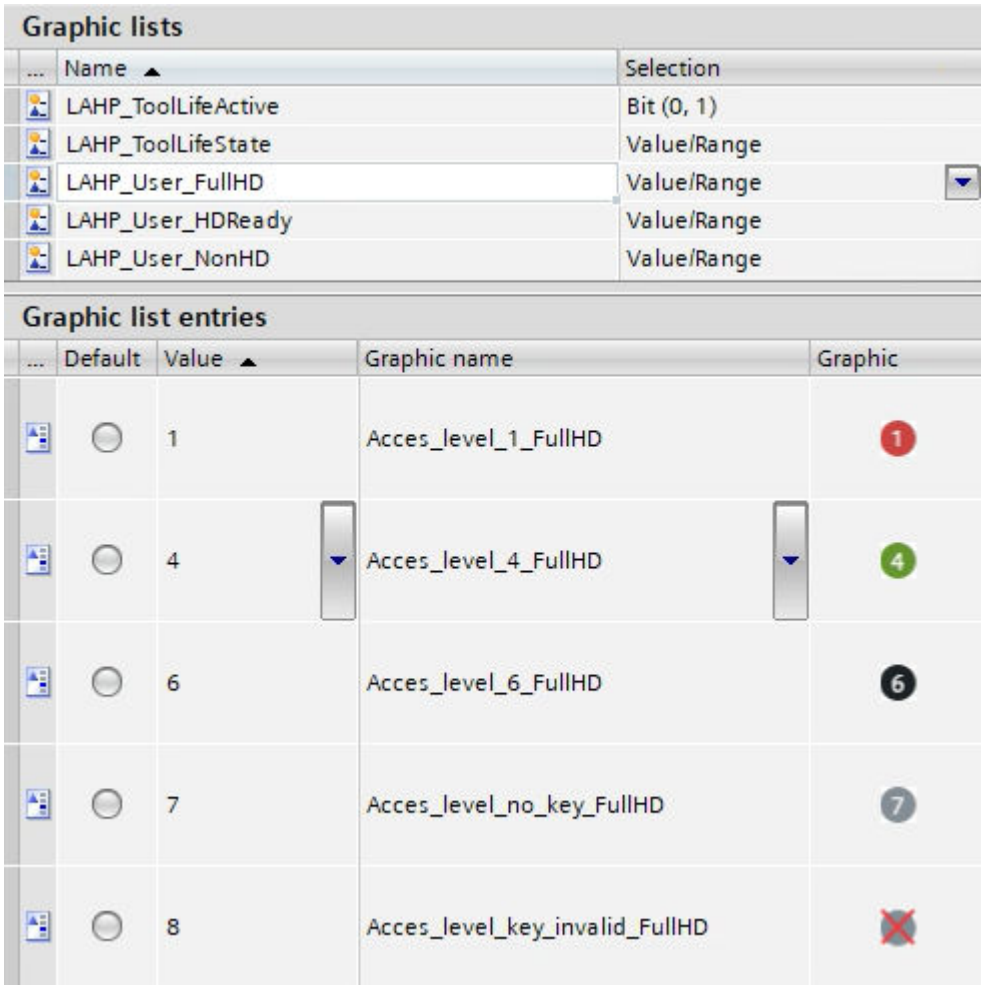


Figure 3-53 EKS graphic list

Note

By default, the EKS levels 1, 4, 6, 7, and 8 are configured, the user can also configure 2, 3, and 5.

4. In "Graphic list entries" window, select the required graphic name to change the graphic.

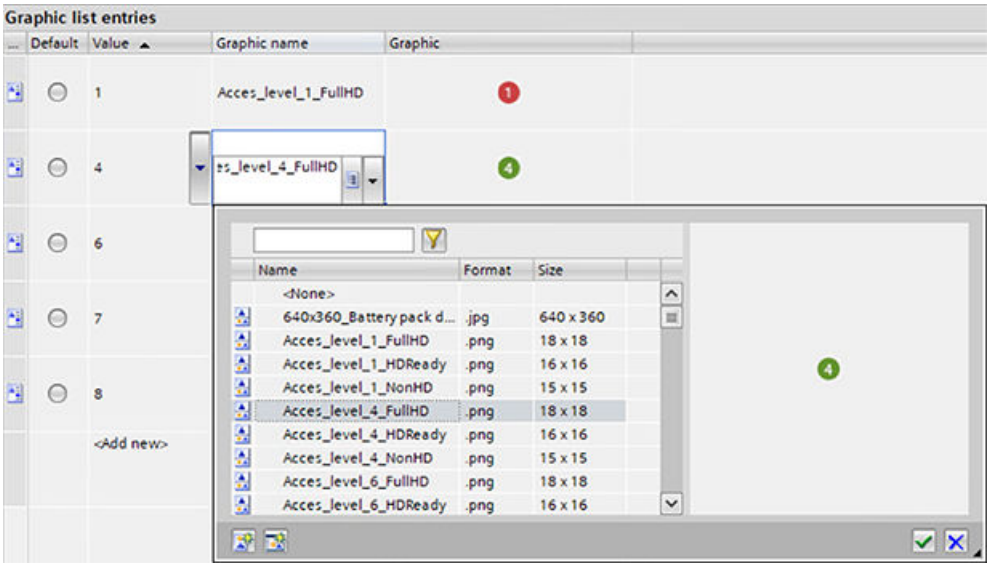


Figure 3-54 Graphic list entries

5. Click , to add a new graphics
   or
   Select the required graphic from the list.
6. Click .

3.11.3 Editing text list

Procedure

You can configure the required EKS authorization name in "LAHP_EksProtectionLevel" text list as follows:

1. In Project tree, navigate to “Devices” >> IPC device >> “Text and graphic lists”.
2. Select “LAHP_EKSProtectionLevel” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default names of EKS authorization.

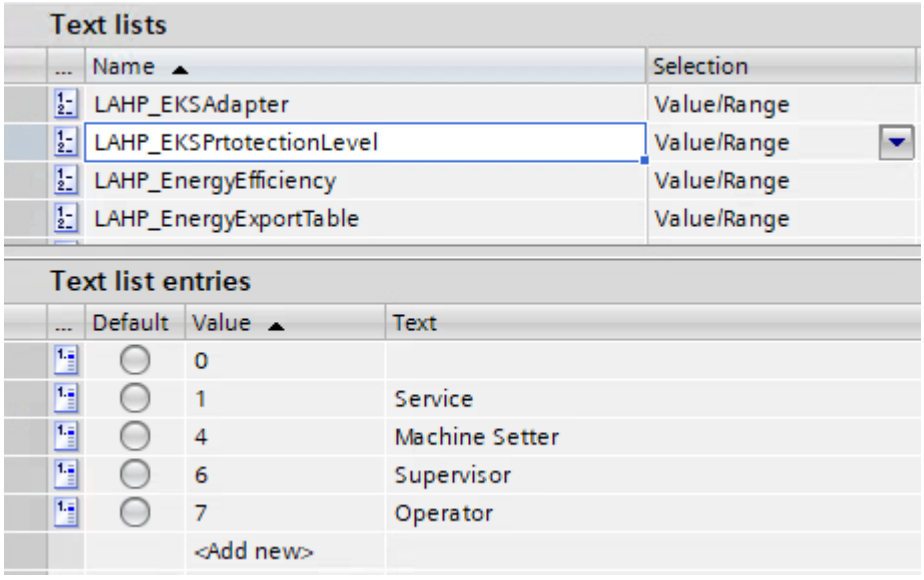


Figure 3-55 EKS protection level text list

3. In "Text list entries" window, select the required value and modify the text as per the requirement.

Note

By default, EKS protection level 1, 4, 6, and 7 are configured. You can also configure 2, 3, and 5.

You can configure the name of the EKS adapter in "LAHP_EKSAdapter" text list as follows:

1. Navigate to “Project tree” >> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAHP_EKSAdapter" from the "Text lists" window.

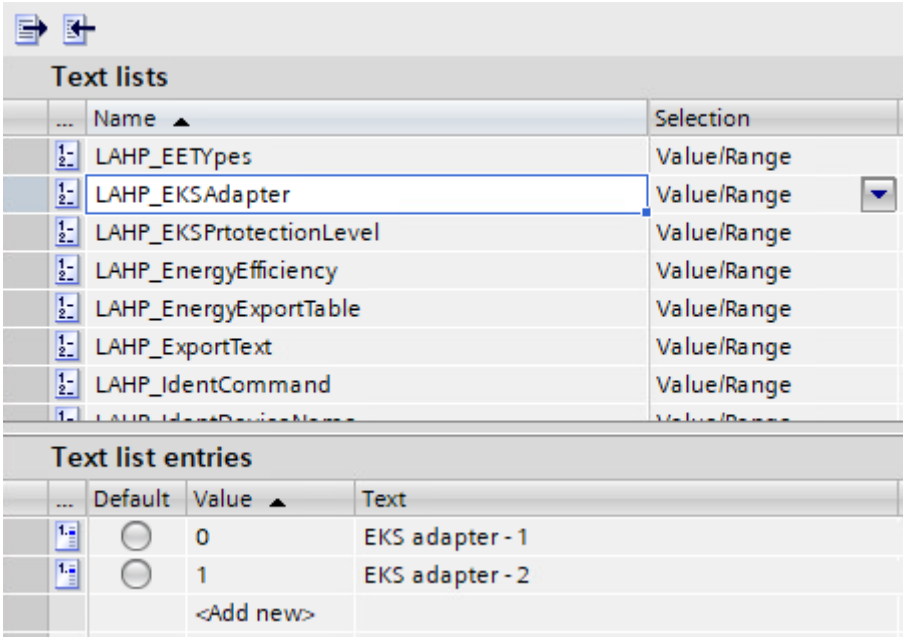


Figure 3-56 LAHP_EKSAdapter

3. In "Text list entries" window, select the required value and modify the text as per the requirement.

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

3.12 Configuring text list for camera

Procedure

You can configure the required camera device in "LAHP_CameraDevices" text list as follows:

1. In Project tree, navigate to “Devices” >> IPC device >> “Text and graphic lists”.
2. Select “LAHP_CameraDevices” from the “Text lists” screen. “Text list entries” screen appears below the screen, which lists default devices added.

Text lists

...	Name ▲	Selection
	LAHP_Browser	Value/Range
	LAHP_CameraDevices	Value/Range
	LAHP_ChangeToolData	Value/Range
	LAHP_DeviceState	Value/Range
	LAHP_EKSAdapter	Value/Range
	LAHP_IdentCommand	Value/Range
	LAHP_IdentDeviceName	Value/Range
	LAHP_IdentStatus	Value/Range

Figure 3-57 LAHP_CameraDevices

3. Add the required number of devices in "Text list entries".

Text list entries

...	Default	Value ▲	Text
		0	MV 420
		1	MV 540 S
		2	MV 540 H
		<Add new>	

Figure 3-58 Text list entries

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

3.13 LAHP_Browser

3.13.1 Overview

A web control object is integrated in the WinCC Unified WebServer screen.

By configuring the URL and the IP address of the devices in "LAPH_Browser" text list.

Note

Ensure that all the required devices are accessible from HMI network.

Browser is available for PLC only.

For CNC, you can use standard browser from SINUMERIK Operate.

3.13.2 Editing text

Procedure

You can modify the URL and IP address. Follow the below steps to change:

1. Navigate to “Project tree” >> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAHP_Browser" from the "Text lists" window.

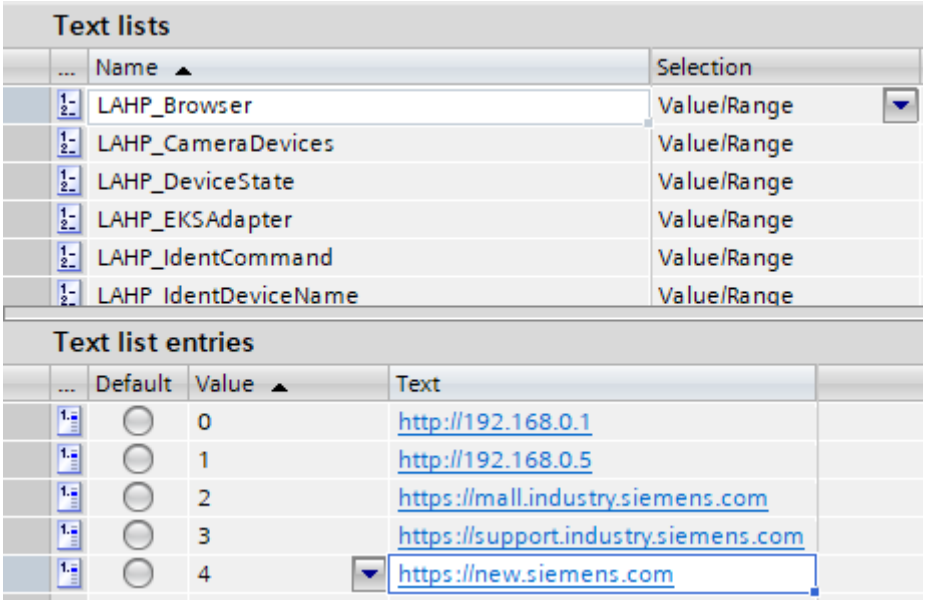


Figure 3-59 Text lists

3. In "Text list entries " window, select the required value and modify the text as per the requirement.

For more information, refer "Visualizing process with Runtime Unified >> Configuring text lists and graphics list" section in TIA Portal online help.

3.14 LAHP_Interface

3.14.1 Editing text

Procedure

Editing interface names

1. Navigate to “Project tree”>> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAHP_Interface", in the “Text list entries” window, modify the interface name in "Text" field.

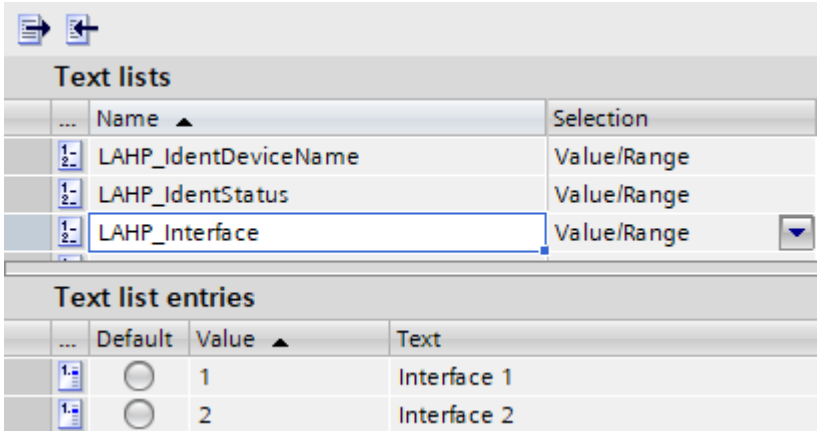


Figure 3-60 LAHP_Interface

Editing input interface names

1. Navigate to “Project tree”>> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAHP_InputInterface0", in the “Text list entries” window, modify the interface name in "Text" field.

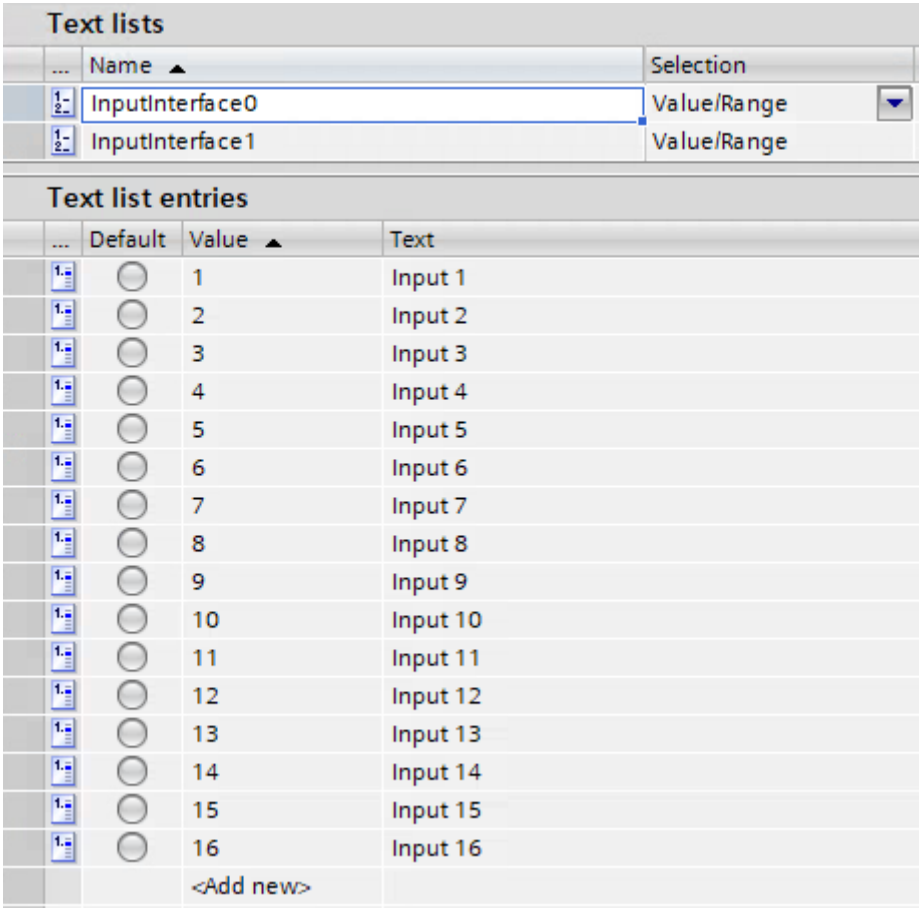


Figure 3-61 Input interface

Note

Follow the same steps to edit the required texts for other input interfaces.

Editing output interface names

1. Navigate to “Project tree”>> IPC device>> “Text and graphic lists”. “Text lists” window opens.
2. Select "LAHP_OutputInterface0", in the “Text list entries” window, modify the interface name in "Text" field.

Text lists

...	Name	Selection
	OutputInterface0	Value/Range
	OutputInterface1	Value/Range

Text list entries

...	Default	Value	Text
		1	Output 1
		2	Output 2
		3	Output 3
		4	Output 4
		5	Output 5
		6	Output 6
		7	Output 7
		8	Output 8
		9	Output 9
		10	Output 10
		11	Output 11
		12	Output 12
		13	Output 13
		14	Output 14
		15	Output 15
		16	Output 16
		<Add new>	

Figure 3-62 Output interface

Note

Follow the same steps to edit the required texts for other output interfaces.

Abbreviations

4

A	
ACK	Acknowledgement
ACTV	Active
ACTVCHN	Active Channel
ACT	Actual
ADPTR	Adapter
ASYM	Asymmetry
ATTR	Attribute
ATTRIB	Attributes
AUTHZ	Authorization
AUTO	Automatic
AVG	Average

B	
BCD	Binary Coded Decimals
BI	Bidirectional
BL	Business logic

C	
CALC	Calculate
CF/CONFIG	Configuration
CLK	Clock
CLR	Color
CMD	Command
CMH	Create MyHMI
CMHA	Create MyHMI /Automotive
CMPLT	Complete
CNFRM	Confirm
CNT	count
COMM	Communication
CORR	Corrected
COST CTR	Cost Center
CT	Cycletime
CTR	Center
CTRL	Control
CURR	Current (electrical)

D	
D4	Display 4
DB	Data Block
DEF	Definition
DESCR	Description
DIAG	Diagnostics
DIFF	Difference
DISBL	Disable
DISP	Display
DM	Data Matrix
DO	Drive Object
DTL/DT	Date and Time

E	
EE	Energy Efficiency
EKS	Electronic Key Switch
EM	Emergency
EMSTOP	Extended Stop
ENBL	Enable
ENC	Encoder
EPOS	Basic positioner
EQUIP	Equipment
ERR	Error
ERRID	ErrorID
ET	Elapsed time
EXEC	Execute
EXP	Expiry
EXTN	Extension
EXTND	Extended

F	
FCT	Function
FCTS	Functions
FP	Flag pulse
FREQ	Frequency

G	
GRP	Group

H	
HDR	Header
HMI	Human Machine Interface
HR	Hour
HW	Hardware

I	
ID	Identity
IM	Identification and Maintenance
IMM	Immediate
IN	Input
INC	Increment
IND	Index
INFO	Information
INST	Instance
INTF	Interface
IP	Internet Protocol
IPC	Industrial PC

L	
LAHP	Library automotive HMI plus
LAPP	Library automotive PLC program plus
LWR	Lower

M	
MACH	Machine
MAN	Manual
MAX	Maximum
MEAS	Measurement
MECH	Mechanical
MIN	Minimum
MLFB	Machine Readable Product Identifier
MO	Mode of Operation
MON	Monitoring

N	
NAV	Navigation
NCU	Numerical Control Unit
NOK	Not Ok
NUM	Number

O	
OBJ	Object
OEM	Original Equipment Manufacturer
OPHRS	Operating Hours
OPTR	Operator
OUT	Output
OVLD	Overload

P	
PARAM	Parameterization
PCT	Percent/Percentage
POS	Position
PRV/PREV	Previous
PRIO	Priority
PN	PROFINET
PLC	Programmable Logic Controller
PROT	Protection

R	
RFID	Radio Frequency Identification
RDREC	ReadRecord
REC	Record
REG	Registry
REL	Release
REQ	Request
RST	Reset
RT	Runtime

S	
SEL	Selection/select
SENS	Sensor
SIGN	Signature
SW	Software
STD	Standard
STATS	Statistics
SPC	Statistical process control
SCHELE	Switching element
SVG	Scalable vector graphics
SSM	Single step mode
SYS	System

T	
TO	Technology Object
TOT	Total
TL	Transline
TRG	Trigger

U	
UCP	Unified Comfort Panel
URL	Uniform Resource Locator
UPR	Upper
USR	User

V	
VAR	Variable
VIB	Vibration

W	
WARN	Warning
WPIECE	Work Piece
WRREC	WriteRecord

Appendix A

A

A.1 Rulesets

This topic is about the deviations from programming guidelines. Following are the list of deviated rules:

Settings in TIA portal

Rule No.	Standard Rulesets	Deviated Rulesets
8	R: tab characters are not permitted in the source text. Indentations must be realized with two space characters.	Indentations are realized with four space characters.

Identifier

Rule No.	Standard Rulesets	Deviated Rulesets
14	R: only meaningful identifiers are used.	Identifiers used are based on recommendation, as specific review is difficult.

PLC programming

Rule No.	Standard Rulesets	Deviated Rulesets
26	R: within a block, only local tags are used.	In exceptional cases, for security reasons local tags are not used everywhere.
34	R: data exchange between blocks is performed exclusively via the block interfaces.	In exceptional cases, for security reasons data exchange between blocks is not via the block interfaces.

For more information on programming guidelines, refer to Programming Guidelines and Programming Styleguide for SIMATIC S7-1200 and S7-1500 and WinCC (TIA Portal) (https://support.industry.siemens.com/cs/document/81318674/programming-guidelines-and-programming-styleguide-for-simatic-s7-1200-and-s7-1500-and-wincc-(tia-portal)?dti=0&lc=en-WW).

Index

M

T

U