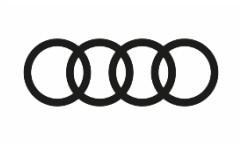
SIEMENS

VOLKSWAGEN

AKTIENGESELLSCHAFT



VASS V6 Standard

Standardization guideline

Safety Information

This documentation contains information which you must adhere to for your own personal safety and to avoid property damage. Notices referring to your personal safety are highlighted in the manual by a safety alert symbol; notices referring to property damage only have no safety alert symbol. Depending on the hazard level, the warnings are displayed in descending order as follows.

Danger

Indicates that death or severe bodily injury **will** result if proper precautions are not taken.

Warning

Indicates that death or severe bodily injury **may** result if proper precautions are not taken.

Caution

With a hazard alert symbol, indicates that minor bodily injury **may** result if proper precautions are not taken.

Caution

Without a hazard alert symbol, indicates that property damage **may** result if proper precautions are not taken.

Notice

Indicates that an unintended result or situation **may** occur if the corresponding information is not taken into account.

If multiple hazard levels may occur, the warning is always displayed with the highest possible level. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Typographical Conventions

Note

This symbol indicates information to facilitate understanding of machine processes and points to important information about the system or about handling the system.

Reference / note (to other documentation)

This symbol references information from other pieces of documentation that are connected with the existing system in a broader context.

In development, technical clarification / initial processing or revision

This symbol references a text segment in the documentation that has not been completed yet.

Qualified personnel

The device/system may only be set up and used according to this documentation. Only **qualified personnel** is allowed to install and work on a device/system. Qualified personnel as referred to in the safety guidelines in this documentation are those who are authorized to start up, earth and label units, systems and circuits in accordance with the relevant safety standards.

Intended use

Warning

This equipment may only be used for the applications described in the catalog and in the technical description, and only in conjunction with non-Siemens equipment and components recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

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

Disclaimer

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

Legal Notice

Document Identification

Project name:	VASS V6 Standard Standardization Guideline
Scope:	Volkswagen Group Projects with Siemens Control Technology and VASS Standard
Document type:	Standardization guideline
Document name:	VASS V6 Standard Standardization Guideline_V_1_0_EN.docx
Processed by:	Michael Stengel (ISO Software Systeme GmbH) commissioned by Siemens AG
Saved on:	October 15, 2019

Current version

Version		Editor	Reason for change	Comments
As of	Date			
1.0	23 July 2015	Michael Stengel	---	First version

Version history

Version		Editor	Reason for change	Comments
As of	Date			
1.0	15 October 2019	Michael Stengel	Adaptation to the VASS V6 Standard	Complete revision

Changes to the previous release

The following significant changes were made compared to the published version (VASS Standard Standardization Guideline_V_1_0 / July, 23, 2015):

Version 1.0 of the documentation **"VASS V6 Standard Configuration Guideline_V_1_0"** is a complete revision of the entire book of the previous version with regard to the **VASS V6 Standard**.

Therefore, individual changes, updates, deletions, etc. are not marked (highlighted in color) separately in this documentation, or listed here.

Siemens AG
Automation and Drives
P.O. Box 48 48
90437 Nuremberg
GERMANY

Volkswagen AG
Planning Automation technology
38436 WOLFSBURG
GERMANY

Contact persons

VOLKSWAGEN AG	**Technical**
Name:	Carsten Zickner
Department:	Automation Engineering / Control Technology Planning of manufacturing automation and digital production (PPD-A/F)
Location:	Wolfsburg
Phone:	+49 5361 9-993106
Mobile:	+49 151 16807985
Email:	carsten.zickner@volkswagen.de

VOLKSWAGEN AG	**Technical**
Name:	Frank Hennig
Department:	Automation Engineering / Control Technology Planning of manufacturing automation and digital production (PPD-A/F)
Location:	Wolfsburg
Phone:	+49 5361 9-123003
Mobile:	+49 1522 2923003
Email:	frank.hennig@volkswagen.de

AUDI AG	**Technical**
Name:	Felix Niederbacher
Department:	Planning of Automation Engineering (N/PG-D62)
Location:	Neckarsulm
Phone:	+49 7132 31-72489
Mobile:	+49 160 97295871
Email:	felix.niederbacher@audi.de

SEAT	**Technical**
Name:	Jesus Hurtardo
Department:	Automatizacion (SEA/PP-6)
City:	Martorell
Phone:	+34 937 733145
Mobile:	+34 636 213749
Email:	jesus.hurtardo@seat.es

SKODA	**Technical**
Name:	Jiri Nosek
Department:	Planovani Automatizace (PPF/1)
City:	Mlada Boleslav
Phone:	+420 326 840008
Mobile:	+420 734 298506
Email:	jiri.nosek5@skoda-auto.cz

ISO Software Systeme GmbH	**Document Creation**
Name:	Michael Stengel
Department:	ECM (Enterprise Content Management)
Location:	Nuremberg

Table of contents

Safety Information 2

Typographical Conventions 2

Qualified personnel 3

Intended use 3

Trademarks 3

Disclaimer 3

Legal Notice 3

Document Identification 4

Current version 4

Version history 4

Changes to the previous release 4

Contact persons 6

Table of contents 8

1 Introduction and Objectives 14

1.1 Introduction 14

1.1.1 Purpose of the documentation 14

1.1.2 Basic information on the objectives 14

1.1.3 Target groups 15

1.1.4 Qualification required 15

1.1.5 Follow-up versions 15

2 Basics 16

2.1 Designation structure and PROFINET device name 16

2.2 Cell production 18

2.3 Line production 19

2.4 General location name 20

2.4.1 E-Plan 20

2.4.2 STEP 7 21

2.5 Name of function elements 22

2.6 Device ID 23

2.6.1 Contactors (coils or variables) 23

2.6.2 Commands 24
2.6.3 Controls 25
2.6.4 Display elements 26
2.6.5 Operator controls and display elements 26

3 General overview of the VASS Standard Configuration Guideline 27
3.1 STEP 7 30
3.1.1 Basic project 30
3.1.1.1 GDS file 31
3.1.1.2 HW Config 31
3.1.2 Device catalog 32
3.1.3 Library 33
3.1.3.1 Blocks 34
3.1.3.2 Data structures 34
3.1.3.3 PLC tags 34
3.2 WinCC Advanced 35
3.2.1 Faceplates (BB) 35
3.2.2 Faceplates for detailed screens 36
3.2.3 Languages and project texts 37
3.3 SiVArc 38
3.4 Documentation 39
3.4.1 Block help 39
3.4.2 Hardware configurations 40
3.4.3 Operating and visualization information 40
3.4.4 Device parameterization 40

4 Specifications for brand and plant-specific standards 41

5 Development of a VASS module 46
5.1 Working in the VASS basic project 46
5.1.1 Integrating the GSDML file into the HW Configuration 47
5.1.2 Setting up a device in HW Config 48
5.2 Working in the device catalog 49
5.3 Working in the TIA library 55
5.3.1 Creating symbols 56
5.3.2 Creating module FBs in SCL 57
5.3.2.1 Information on general structures 57
5.3.2.2 Add new block 59
5.3.2.3 Pop-up "Add new block" 60
5.3.2.4 Properties (General) 61
5.3.2.5 Properties information 61

5.3.2.6 Creating a module component folder....62
5.3.2.7 Rename new group....63
5.3.2.8 Moving the module component to the module component folder....64
5.3.3 Changing a module component to a library type....65
5.3.3.1 Copying a block from the project navigation to the project library (create library type)65
5.3.3.2 "Add type" dialog....66
5.3.3.3 Creating folders in the project library....67
5.3.3.4 Moving a block in the project library....68
5.3.4 Release block type for editing....69
5.3.4.1 Requirements....69
5.3.4.2 Release for editing....69
5.3.5 Enable type version dialog....70
5.3.5.1 Tag declaration....71
5.3.5.2 Introduction to SCL....77
5.3.5.2.1 SCL - Structured Control Language....77
5.3.5.2.2 Block structure....77
5.3.5.2.3 Identifier in SCL....78
5.3.5.2.4 Display of numbers....78
5.3.5.2.5 Operators/operands/expressions....79
5.3.5.2.6 Comment....79
5.3.5.2.7 Slice function....80
5.3.5.2.8 Input and output parameters....82
5.3.5.3 Structure of a module FB in SCL....83
5.3.5.3.1 Structure of the FB....83
5.3.5.3.2 Change journal....84
5.3.5.3.3 Explanation of the configuration bits....85
5.3.5.3.4 Checking buttons of the visualization....86
5.3.5.3.5 Program section with control of the outputs....87
5.3.5.3.6 Evaluation of messages....93
5.3.5.3.7 Assignment of supervisions for ProDiag....94
5.3.5.3.8 Preparation of the symbol tags for visualization....98
5.3.5.3.9 Animation navigation....101
5.3.6 Interconnecting structures....104

6 WinCC Advanced and SiVArc....105

6.1 Working in WinCC - Basic project....105

6.1.1 General observation on visualization elements....105
6.1.1.1 Colors in WinCC Advanced....105
6.1.2 Creation of a faceplate....106
6.1.2.1 General information on faceplates....107

6.1.2.2 Creating an HMI-UDT ... 109
6.1.2.3 Individual objects in the faceplate ... 110
6.1.2.4 Documentation ... 122
6.1.3 Creation of a faceplate for a detailed screen ... 123
6.1.3.1 General information ... 123
6.1.3.2 Storage in the project ... 123
6.1.3.3 Creating a detailed screen ... 124
6.1.3.3.1 Layout (screen template) for detailed screens ... 124
6.1.3.3.2 Object names of objects in the faceplate for a detailed screen ... 127
6.1.3.3.3 Assignment of objects to layers ... 128
6.1.3.3.4 Supply text fields ... 130
6.1.3.3.5 HMI UDT ... 131
6.1.3.4 Tags for the detailed screen ... 132
6.1.3.5 Faceplate for calling the detailed screen ... 133
6.1.4 Creation of a faceplate for a detailed screen (Technology) ... 137
6.1.4.1 Faceplate for calling the detailed screen ... 140
6.2 VASS library ... 144
6.3 Faceplate texts and multilingualism ... 145
6.3.1 Type-specific texts ... 145
6.3.2 Instance-specific texts ... 147
6.4 SiVArc rules ... 149
6.4.1 Plug-Ins SiVArc properties in faceplates ... 149
6.4.1.1 Faceplate BB_Ventil_03 name - HMI-UDT and texts ... 149
6.4.1.2 Faceplate BB_Ventil_04_A - Name, HMI-UDT and texts ... 149
6.4.2 SiVrc tag rules ... 150
6.4.2.1 Definition of tag rules tag group hierarchy with "SiVArc_" ... 150
6.4.2.2 Generated tag names with "_HMI-UDT" and data type HMI-UDT ... 151
6.4.2.3 Faceplate interface automatically supplied with data type HMI-UDT ... 151
6.4.3 SiVArc Plug-Ins ... 152
6.4.3.1 Plug-Ins SiVArc properties in screen templates - Screen 003 ... 152
6.4.3.2 Plug-Ins SiVArc properties in screen templates - Screen 004 ... 152
6.4.3.3 Use of the parameter "Maske" or "Cfg" ... 153
6.4.4 SiVArc text lists and text list rules ... 155
6.4.4.1 Creation of new SiVArc text lists as master copies ... 155
6.5 "Screen rules" editor ... 157
6.6 Creating screen rules ... 159
6.7 ASCII modulations with SiVArc expressions ... 164
6.7.1 Differences in syntax for PLC and HMI tags ... 164
6.7.1.1 Syntax example PLC tag ... 164
6.7.1.2 Syntax example HMI tag ... 164

6.7.2 FB642 FB_S7G_Control with parameter S7G_Control_Ext Var.A ... 165
6.7.3 FB642 FB_S7G_Control with parameter S7G_Control_Ext Var.B ... 166
6.7.4 FB_Vorwahl TextFields ... 167
6.7.4.1 FB_Vorwahl TextField1 ... 168
6.7.4.2 FB_Vorwahl TextField3 ... 168
6.7.4.3 Result in screen Vorwahlen_013 ... 168

7 Block help ... 169
7.1 Storage location ... 169
7.2 Premises ... 170
7.2.1 File name ... 170
7.2.2 File type ... 170
7.2.3 Header ... 170
7.2.4 Continuous document ... 170
7.2.5 Screens / screenshots ... 171
7.2.6 Interconnection examples ... 171
7.3 Notes ... 171
7.3.1 Hyperlinks ... 171
7.4 Description of the content of a block help ... 172
7.4.1 Brief description ... 172
7.4.2 Interconnecting example ... 173
7.4.3 Input parameters ... 174
7.4.4 Output parameters ... 175
7.4.5 Input and output parameters ... 175
7.4.6 External tag references ... 175
7.4.7 Dependencies ... 176
7.4.8 Operating principle ... 176
7.4.9 Alarm system ... 179
7.4.10 Visualization ... 181
7.4.11 Faceplates ... 182
7.4.12 Operation/visualization information ... 183
7.5 Conversion of the block help from MS WORD into PDF format ... 184
7.5.1 General ... 184
7.5.2 Handling ... 184

8 Transfer of a VASS module ... 186
8.1 Library ... 187
8.1.1 Brand or plant library ... 187
8.1.2 Library for a single module ... 194
8.1.3 Saving and archiving a global library ... 200
8.2 Block help ... 202

8.3 Assignment list 202

List of figures 203

List of tables 209

Abbreviations / glossary 211

1 Introduction and Objectives

1.1 Introduction

1.1.1 Purpose of the documentation

This documentation, "VASS V6 Standard Standardization Guideline", serves those involved in the creation of the VASS standard for the Volkswagen Group as a specification and a guideline for implementing standardized modules.

1.1.2 Basic information on the objectives

The documentation "VASS V6 Standard Standardization Guideline" is intended to establish the basis for the creation of new objects (module FBs or faceplates).

The following points are described in detail:

- Structure of module FBs
- Structure of faceplates
- Structure of screen rules
- Operating principle of the SiVArc
- Ensuring a high-quality overall result
- Ensuring high levels of availability and flexibility of the overall system
- Minimizing investment, operating and expansion costs that are required
- Implementing optimal work contents and working conditions for the user group

Note

Detailed items that were not included in the documentation "VASS V6 Standard Standardization Guideline" or in additional standard specifications must be discussed with the contact person in charge at the Volkswagen Group.

For this purpose, the project partner should prepare suggestions to be submitted to the contact person in charge at the Volkswagen Group, based on the existing documentation.

1.1.3 Target groups

The guideline is intended for all software designers and programmers who need to create or edit elements in the context of the VASS standard.

1.1.4 Qualification required

- Simatic Step 7 Programming 1-3 in the TIA Portal
- Simatic Programming 1-2 with S7-SCL in the TIA Portal
- Simatic WinCC Advanced in the TIA Portal
- Options for automatic visualization generation (TIA-SiVArc)
- SIMATIC Safety Integrated in the TIA Portal
- VASS-OEE beginners' workshop

Note

The creation and editing of the standard requires the person to have completed the necessary TIA training courses at Siemens (or comparable e.g. "Volkswagen Academy") and have sufficient knowledge of the VASS standard, creating projects in accordance with the configuration guideline and general knowledge of Volkswagen production technology.

1.1.5 Follow-up versions

If changes and additions of the present documentation should become necessary in the context of the document maintenance by the Volkswagen Group during the project execution, they will be incorporated in the relevant follow-up versions of the document.

Notice

The documentation "VASS Standard Standardization Guideline" does not release the system supplier from its responsibility to ensure that the version of systems and documentation to be supplied is professional and meets the local regulations.

2 Basics

2.1 Designation structure and PROFINET device name

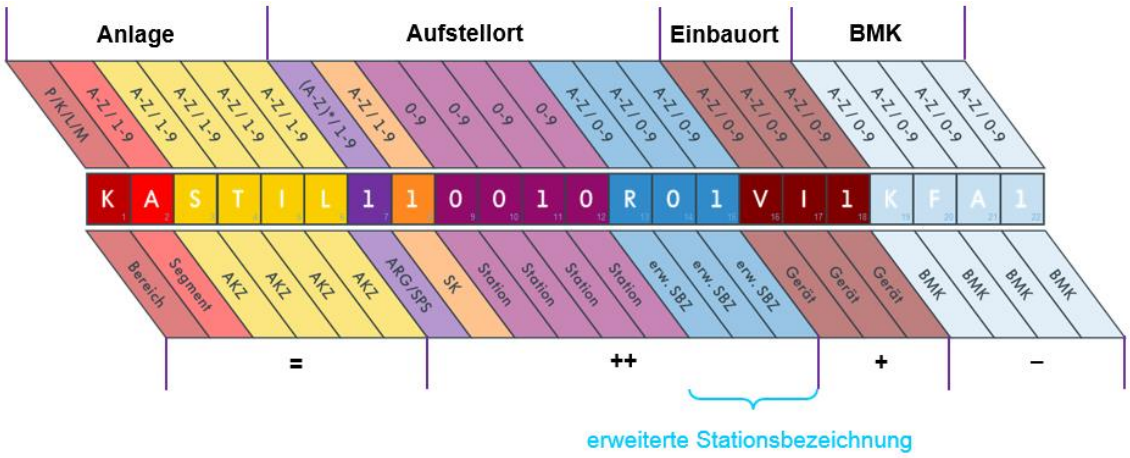


Fig. 2.1 PROFINET device name

	HID	=	Higher level designation
	ARG	=	Workgroup
	SK	=	Protective circuit
	SN	=	Station name
	SRD	=	Standard reference designation

Note

- Spaces must be filled with minus signs ("-").
- If the last digit (no. 22) is not used, this digit must be filled with an "X".
- The device names within a location (plant) must be unique.

Example of PROFINET device name:

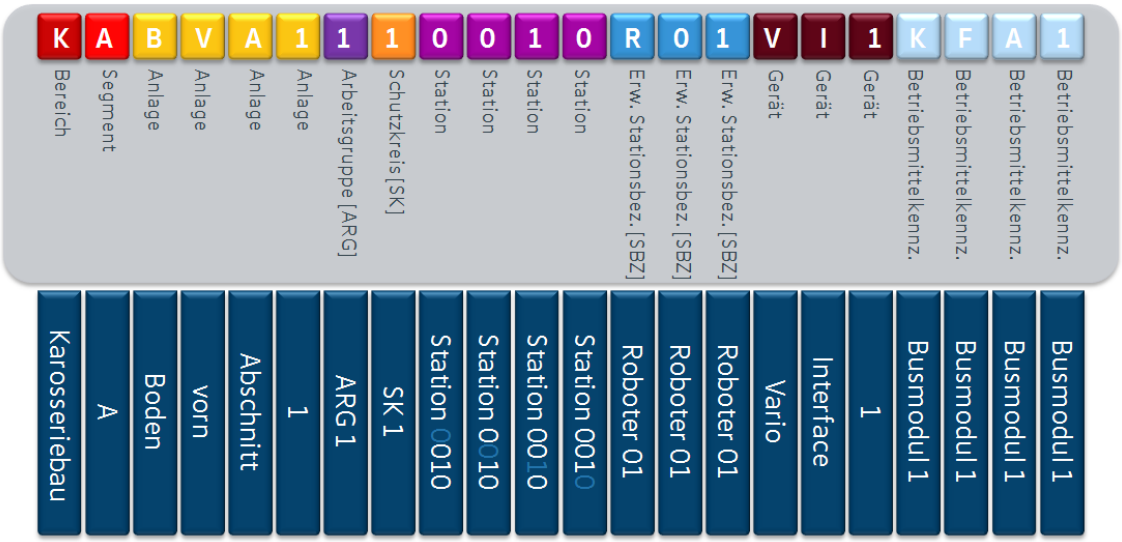


Fig. 2.2 Example of PROFINET device name

2.2 Cell production

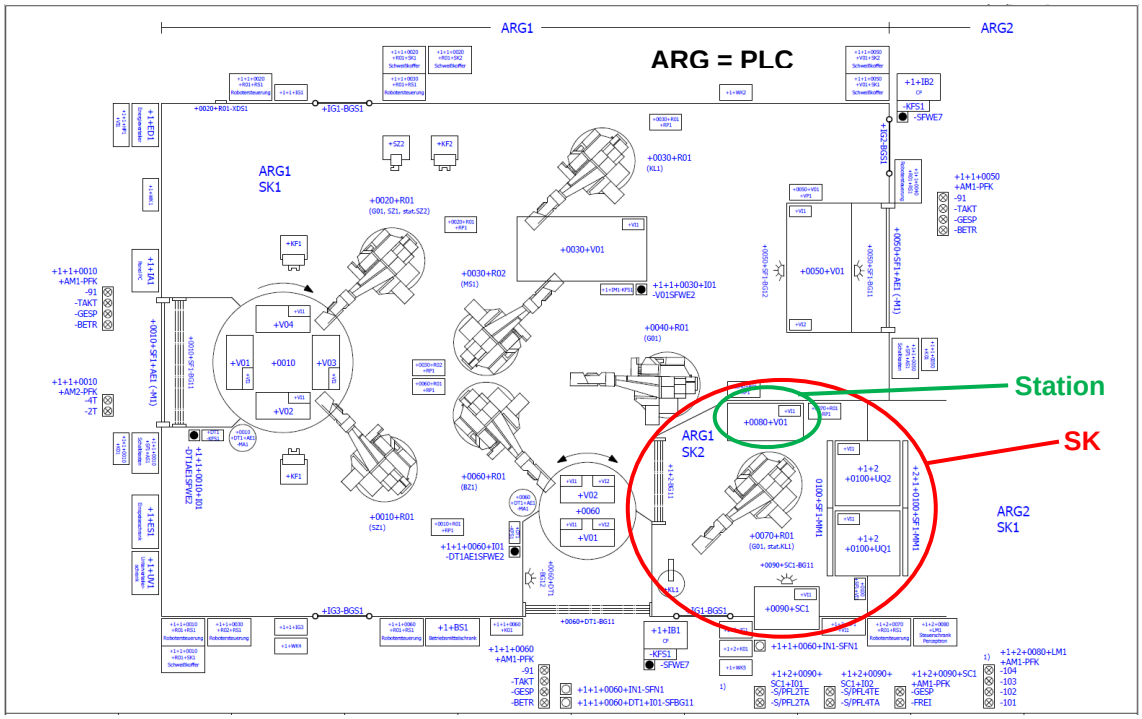


Fig. 2.3 Cell production

"ARG" = A workgroup is the area which is controlled by a PLC.

"SK" = The protective circuit defines an individual safety area which can be separately shut down.

"Station" = A place where a component can be located is referred to as a station.

2.3 Line production

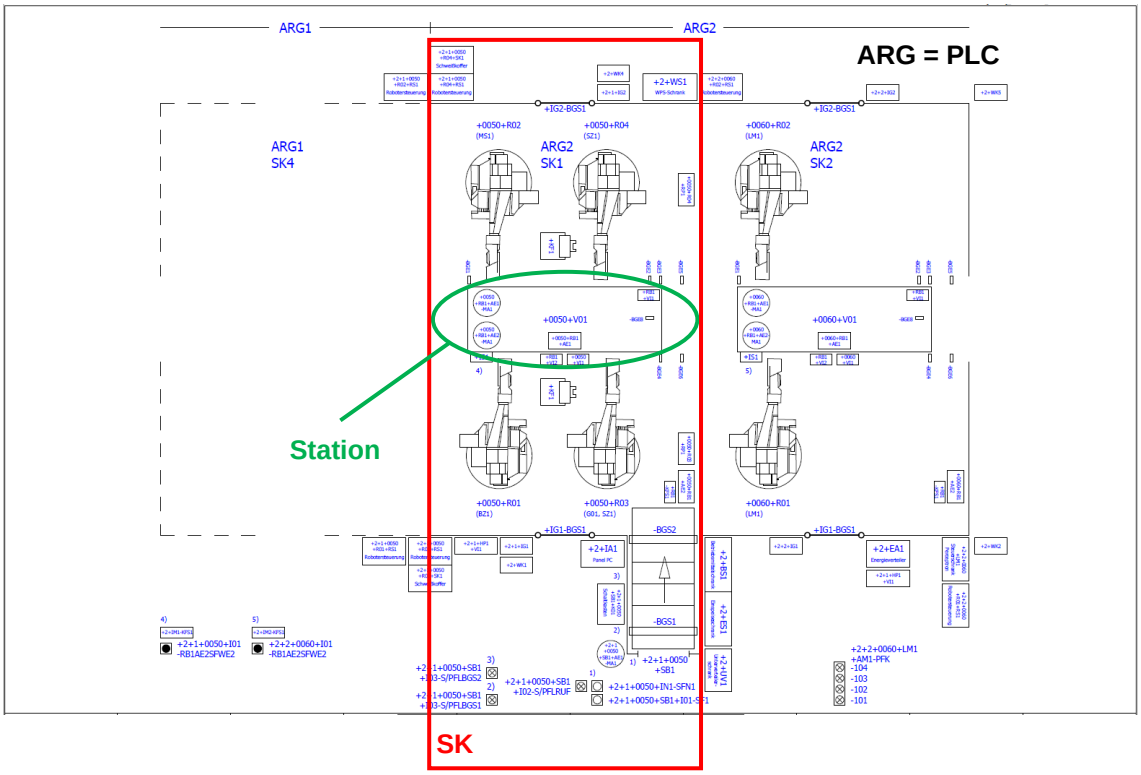


Fig. 2.4 Line production

"ARG" = A workgroup is the area which is controlled by a PLC.

"SK" = The protective circuit defines an individual safety area which can be separately shut down.

"Station" = A place where a component can be located is referred to as a station.

2.4 General location name

2.4.1 E-Plan

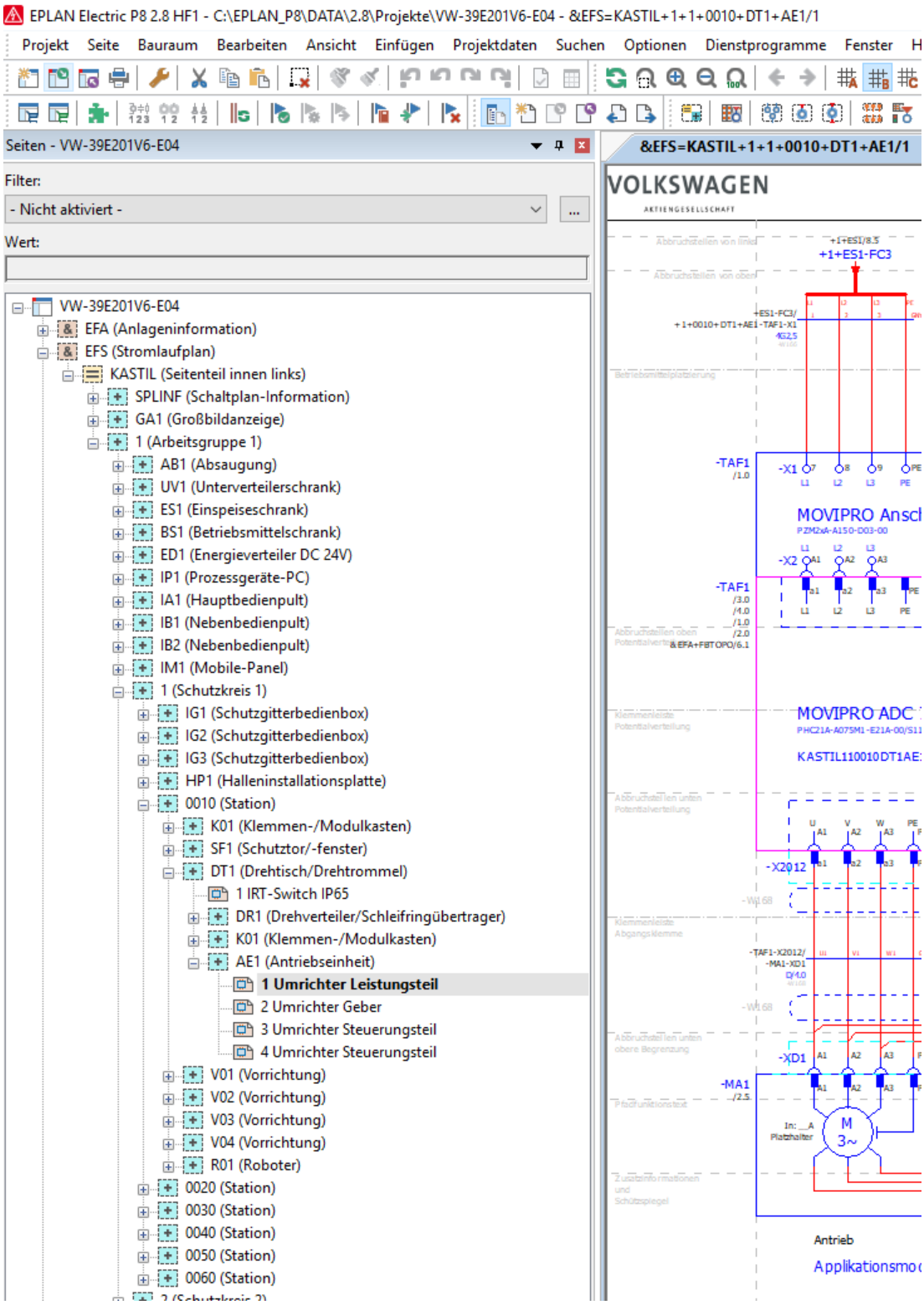


Fig. 2.5 E-Plan

2.4.2 STEP 7

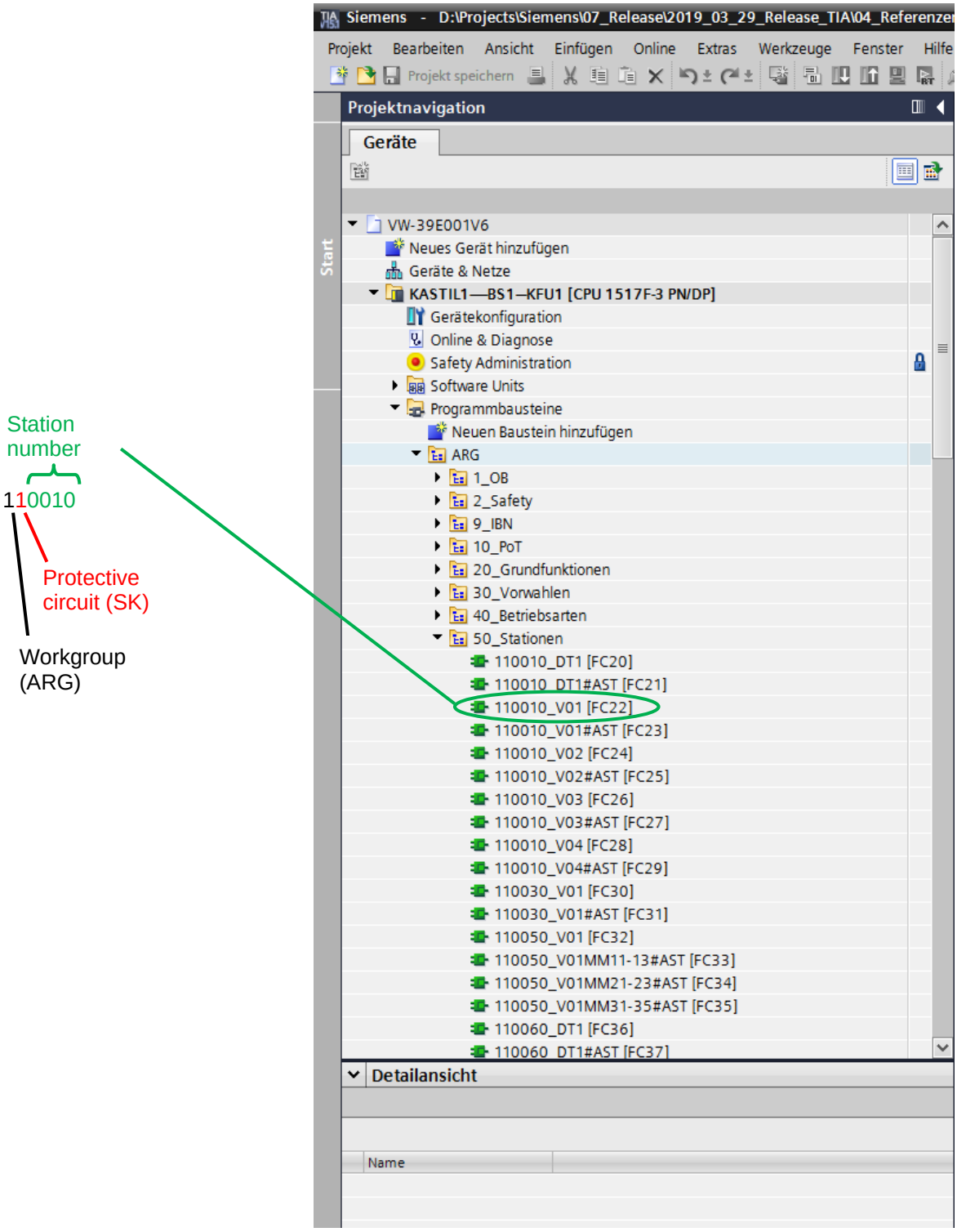


Fig. 2.6 General location name

Explanation of the function structure:

X	(1-digit): Function or signal of a workgroup	(e.g. 1)
XX	(2-digit): Function or signal of a protective circuit	(e.g. 11)
XXXXXX	(6-digit): Function or signal of a station	(e.g. 110010)

2.5 Name of function elements

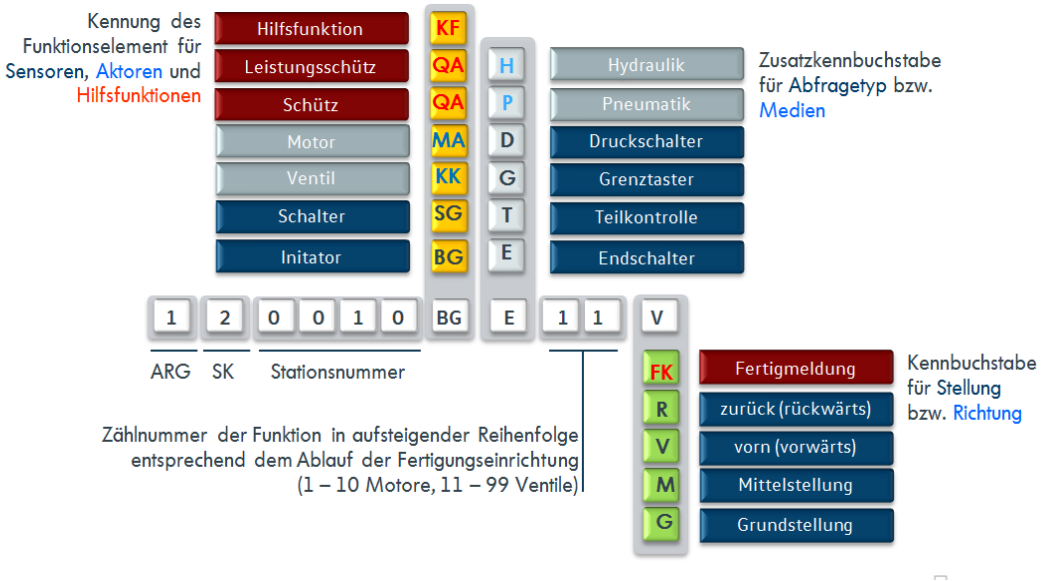


Fig. 2.7 Name of function elements

Note

The name of the functional units is specified by the electrical department.

2.6 Device ID

Some key device IDs are listed below.

2.6.1 Contactors (coils or variables)

Identification letter	No.	Explanation
K	0	Control voltage on
K	4	Emergency stop circuit
RK	4	Emergency stop robot
FK	4	Emergency stop external system
K	5	Two-hand console
K	6	Stop in initial position
K	8	Preselection of individual operation
K	9	Preselection of linked operation
K	10	Initial state
K	16	Emergency stop entire system
K	20	Operating speed
K	23	Pre-selection of single movement
K	25	Enable single movement (manual)
K	25W	Enable single movement tool
K	25MP	Enable single movement via mobile panel
K	26	Enable individual operation / linked operation (automatic)
K	26W	Automatic enable tool
K	26ROB	Automatic enable robot
K	25/Z8	Enable individual movement stop cylinder
K	26/Z8	Automatic enable stop cylinder
K	27	Automatic enable with part
K	28	Automatic enable clamped with part
K	36	Protective circuit
K	36PS	Protective circuit process systems
K	40	Start On
K	48	Control device control
K	51	Controls
K	51U	Controls bridgeable
K	51N	Controls not bridgeable
K	52	Starting requirements
K	61	Drives on
K	81	Change of electrode cap
K	82	Warning stud welding
K	83	Tool life
K	90	Enable access requirement
K	91	Jamming
K	92	Simulation mode

Identification letter	No.	Explanation
K	93	Emptying
K	94	Line stop
K	95	Range stop
K	100	Enable actuator start
K	101	Start single pass
K	111	Enable emergency off and protective circuit
K	121	Bridging lock
K	122	Bridging protective device
K	WE7	Preselection E7
K	WE2	Preselection E2

Tab. 2.1 Explanation of terms "Contactors"

2.6.2 Commands

Identification letter	No.	Explanation
MK	10	Conveyor is at the front
MK	11	Station is ready for conveyor backward
MK	12	Conveyor is back
MK	13	Station is ready for conveyor forward
MK	14	Start linked operation from station (K40;K40b)
MK	16	Enable emergency stop circuit
MK	16G	Enable emergency stop circuit entire system
MK	18	Enable linked operation
MK	40	Station ready for transfer
MK	41	Station ready for takeover
MK	42	Enable runout / manual
MK	44	Part taken over
MK	45	Part transferred
MK	50	Emergency stop und protective circuit closed
MK	51	Protective circuit closed
MK	52	Station ready for open insert protective device
MK	61	Drives on
MK	62	Feedback signal from the drives on

Tab. 2.2 Explanations of terms "Commands"

2.6.3 Controls

Identification letter	No.	Explanation
QB	1	Main switch
QF	1-n	Circuit breaker
FCM	1-n	Motor circuit breaker
FCM	1-n	Motor circuit breaker controller
SFN	1-n	Emergency stop
SF	1M	Motor 1 center
SF	2R	Motor 2 forward
SF	2V	Motor 2 back
SF	11V	Function 11 forward
SF	11R	Function 11 back
SF	WE2 WE22	Keyswitch E2/E22 bridging personal safety
SF	WE7	Keyswitch E7 bridging lock (machinery safety)
SFLT		Lamp test
SFK	0A	Control voltage off
SFK	0E	Control voltage on
SFK	9	Automatic
SFK	10	Pause (stop in initial state)
SFK	21	Switchover "slow"
SFK	22	set up / inching
SFK	23	Manual (single movement)
SFK	40A	Stop
SFK	40E	Start
SFK	100A	Load voltage off
SFK	100E	Load voltage on
SFQUIT		Acknowledge fault
SFFrgK	9	Enable automatic
SFFrgK	22	Enable set up / inching
SFFrgFS		Enable remote control
SFFrgBA		Enable operating mode selection
SFFrgxx		other operator control enables
SFHA	SHxxA	Hydraulics off
SFHE	SHxxE	Hydraulics on
SFSA	SSxxA	Lubrication off
SFSE	SSxxE	Lubrication on
SFFrgS		Enable lubrication selection

Tab. 2.3 Explanation of terms "Operator controls"

2.6.4 Display elements

Identification letter	No.	Explanation
PWE	2	Bridging personal safety
PWE	7	Bridging lock
PK	0	Control on
PK	6	Stop at end of clock
PK	9	Automatic
PK	16	Emergency stop (group display)
PK	20	Fast
PK	21	Slow
PK	22	Setting up
PK	23	Manual
PK	36	Protective grille (group display)
PK	40	Start On
PK	48	Control device control (Bfg)
PK	51	Controls
PK	81	Change of electrode cap
PK	82	Warning stud welding
PK	83	Warning tool life
PK	91	Jamming display
PK	100	Load voltage on
PQFFA		Fuse blown
Pstoe		General fault
PFrgFs		Enable remote control
PFrgBa		Enable operating modes
PFrgxx		Other operator control enables selected
PHA	PHxxA	Hydraulics off
PHE	PHxxE	Hydraulics on
PSN		Emergency stop (single emergency stop)
PSG		Protective grille (single protective grille)

Tab. 2.4 Explanations of terms "Display elements"

2.6.5 Operator controls and display elements

Identification letter	No.	Explanation
KFP		Operator panel in protective grille (SG) - operator box
KFM		Multi - Panel / Mobile - Panel
KFP		Panel - PC

Tab. 2.5 Explanation of terms "Operator controls and display elements"

3 General overview of the VASS Standard Configuration Guideline

Engineering tools used for the VASS standard with Windows 7 / 64-bit or Windows 10 / 64-bit

The utilized tools are listed below.

The versions for editing the standard are based on the version defined in the project.

Program	Version	Function	Use
SIMATIC STEP 7 Professional	V15.1	Configuration S7 controller	Programming device, operator panel
SIMATIC STEP 7 Safety Advanced	V15.1	Configuration S7 controller	Programming device, operator panel
SIMATIC WinCC Advanced	V15.1	Configuration of operator panels	Programming device, operator panel
SIMATIC Visualization Architect	V15.1	Generation of operator interfaces	Programming device, operator panel
TIA Openness	V15.1	Configuration S7 controller	Programming device, operator panel
SIMATIC WinCC Runtime Advanced 8192 PowerTags	V15.1	Runtime environment for WinCC	Programming device, operator panel
SIMATIC ProDiag for WinCC Runtime Advanced	---	Diagnostics for the user interface	Programming device, operator panel
SIMATIC WinCC Logging for Runtime Advanced	---	Configuration of operator panels	Programming device, operator panel
SINAMICS Startdrive	V15.1	Configuration of SINAMICS drives S7 controller	Programming device, operator panel
SINAMICS DCC (no license required)	V15.1	Configuration of SINAMICS drives S7 controller S120	Programming device, operator panel

Tab. 3.1 Engineering tools for VASS standard with Windows 7 / 64-bit or Windows 10 / 64-bit

Notice

If older software versions are used, the functionality of the control software cannot be guaranteed.

When setting up the SW tools, the program versions released in the respective software workshop of the projects must be observed.

Elements in a VASS module:

- **STEP 7**
  - GSD files
  - Symbols
  - Module FB
  - Visualization interface HMI-UDT
  - Interconnecting structure STB
  - Internal block structure ST
- **VISU**
  - Faceplate incl. SiVArc Plug-ins entries
  - Faceplate for detailed screens incl. SiVArc Plug-ins entries (including layout)
  - Scripts, tags, structures
- **SiVArc**
  - Screen rules
  - Text list rules
  - Tag rules
- **Docu**
  - Block help
  - Device documentation

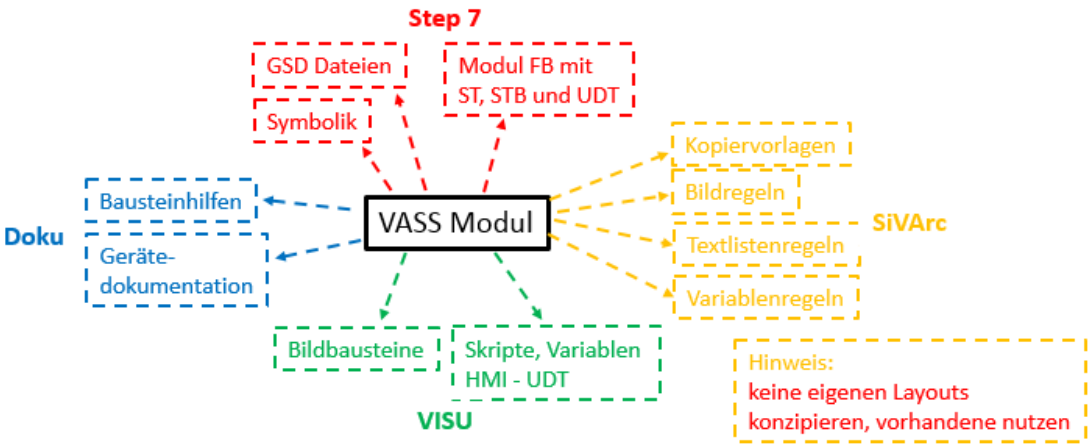


Fig. 3.1 Elements in a VASS module

Overview of the VASS standard:

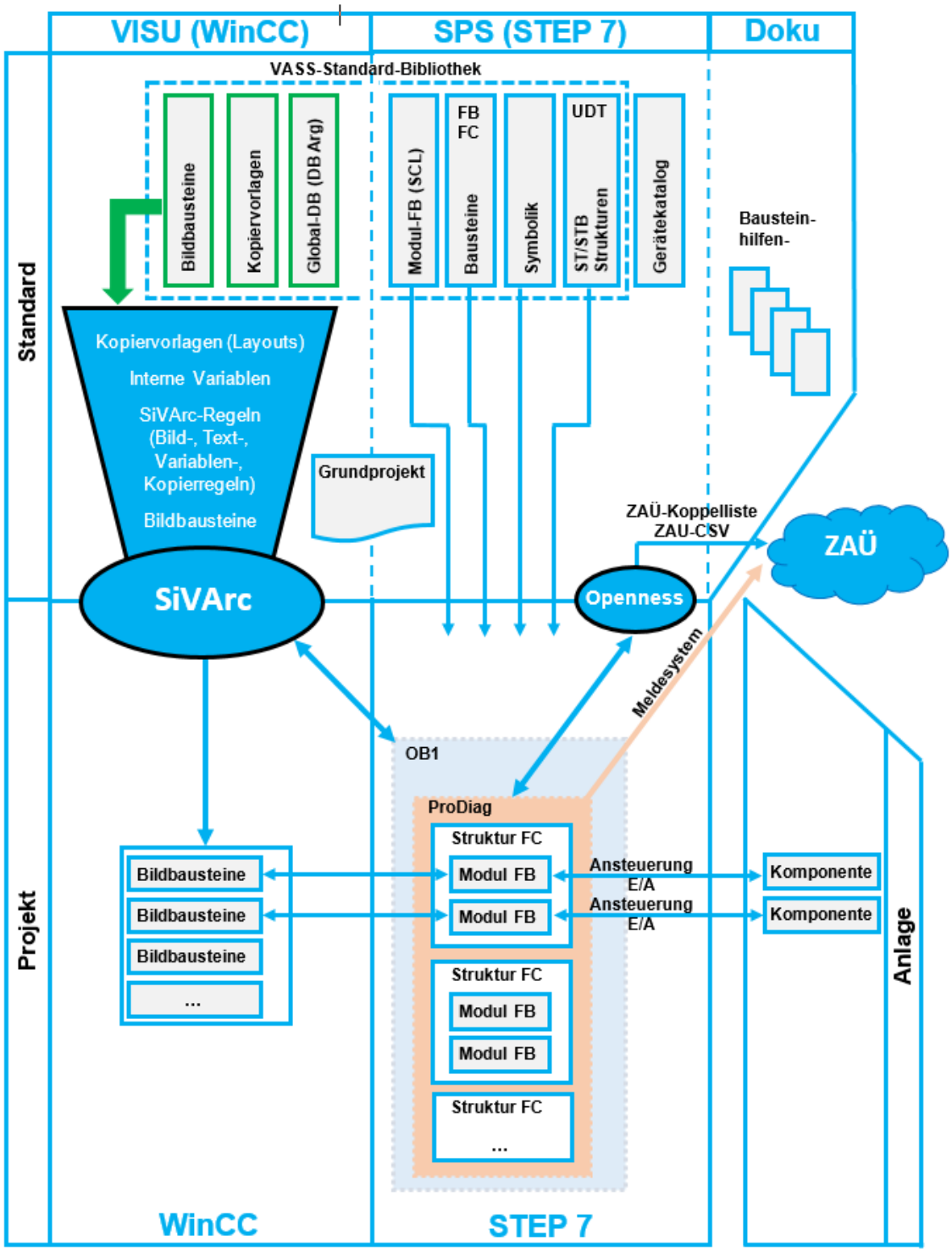


Fig. 3.2 Overview of the VASS standard

3.1 STEP 7

3.1.1 Basic project

The basic project is used in the standard to specify all STEP 7 related basic settings and to provide the GSD files of the PROFINET devices and represent the devices in the hardware configuration once as parameterized examples.

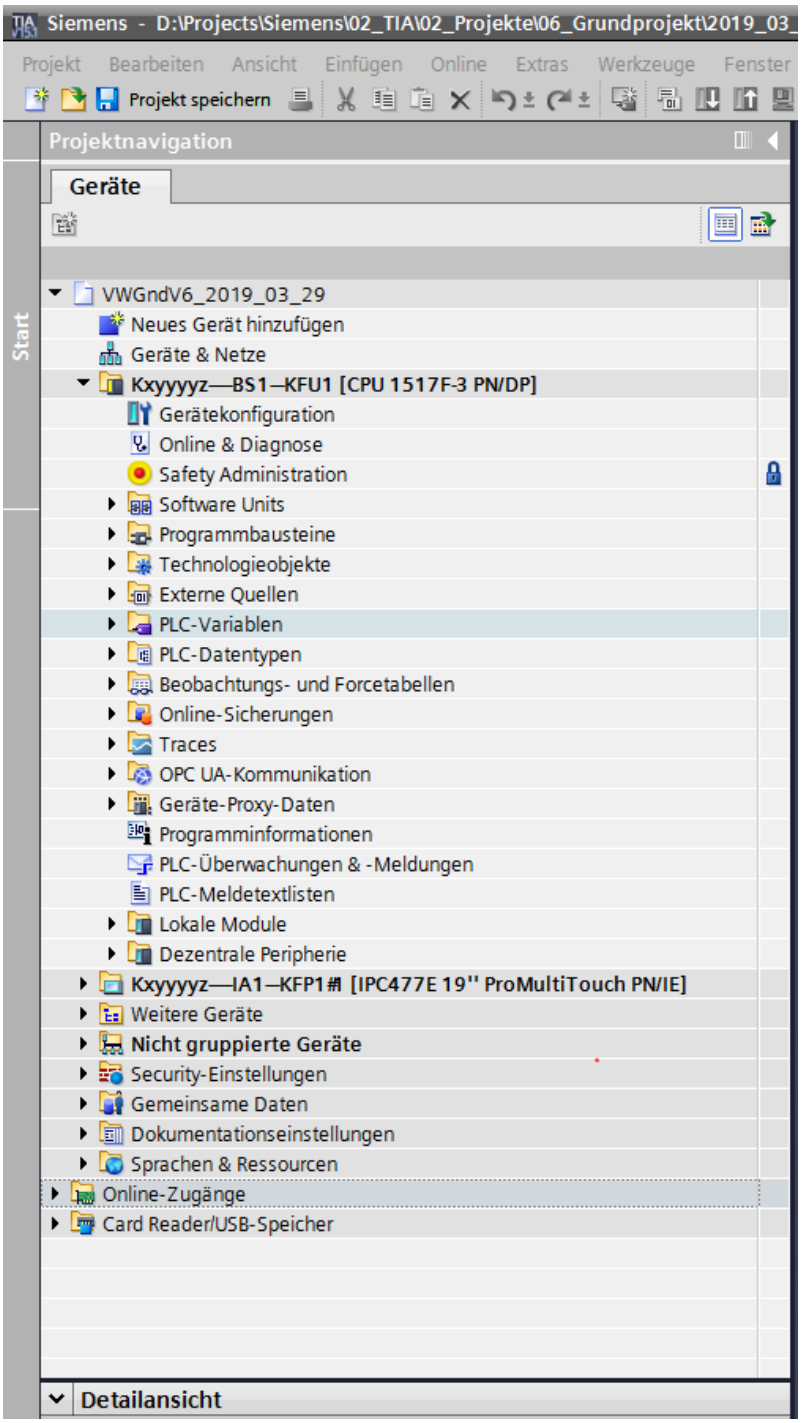


Fig. 3.3 Basic project

3.1.1.1 GDS file

If new PROFINET devices are to be used in the project, their GSD file must be installed once in STEP 7.

The device is available for configuration.

3.1.1.2 HW Config

The device must be incorporated in the HW Config once as an example to represent the parameterization as an example and to be able to transfer the GSD file to STEP 7 installations on other computers via the basic project.

3.1.2 Device catalog

Separate device catalogs are created in the standard in which only the devices that have been released for the standard are located.

They are stored in categories by manufacturer.

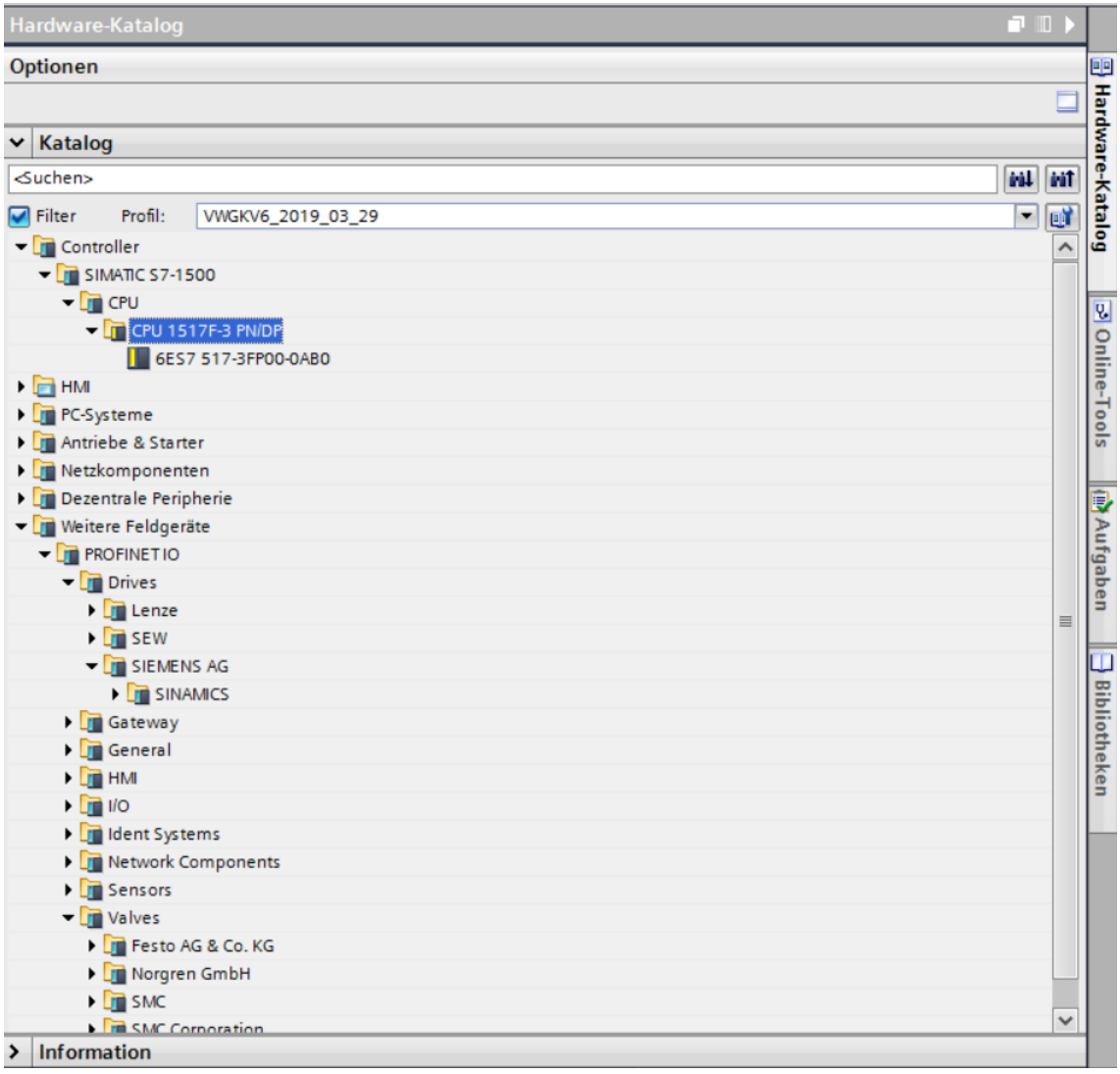


Fig. 3.4 Device catalog

3.1.3 Library

In the context of the VASS standard, the TIA Portal contains a library where ready-made elements are stored. This library contains the name "VASS_V6_VW_yyyy_mm_dd" and is generally referred to as "VASS library".

VASS library:

The folders "VASS_Safety_V6" and "VASS_V6" provide elements according to functional or technological categories.

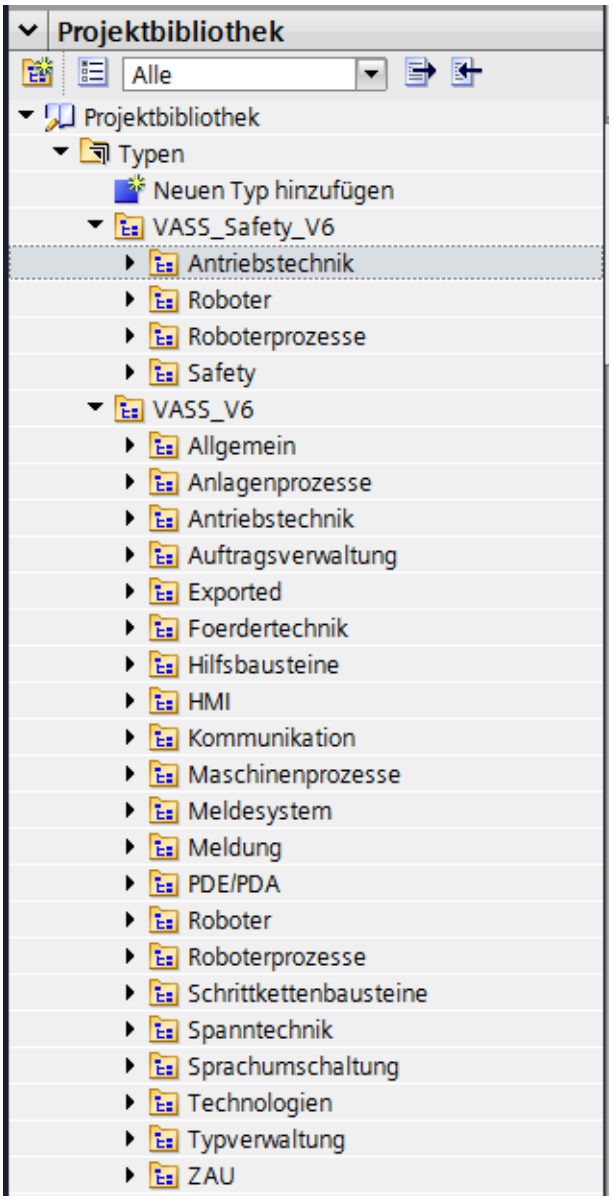


Fig. 3.5 Library

3.1.3.1 Blocks

Function blocks are programmed in the standard that execute a certain function for controlling function modules for actuators and units.

3.1.3.2 Data structures

UDTs are created in the standard that are used in the projects as:

- global structures for passing on information between the blocks (e.g. ST_BA).
  Global structures begin with the prefix "ST_".
- interconnecting structures for external switching of the blocks (e.g. STB_Valve).
  interconnecting structures begin with the prefix "STB_".
- visualization structures for connecting faceplates and FBs (e.g. HMI-UDT_Valve).
  Visualization structures begin with the prefix "HMI-UDT_".

3.1.3.3 PLC tags

The folder "Default tag table" contains all the symbols that are required for each project in the standard.

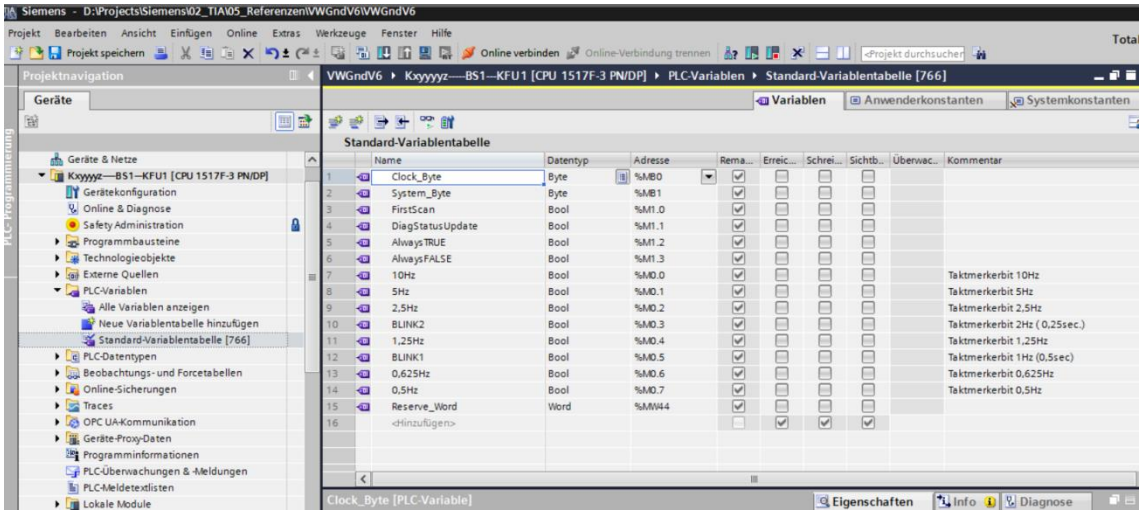


Fig. 3.6 Default tag table

3.2 WinCC Advanced

3.2.1 Faceplates (BB)

If a module component requires a display or operating option, an associated faceplate must be created for it, which is then stored in the project library.

The faceplate communicates with the module component via a defined interface (HMI-UDT).

Texts and message texts are stored directly in the faceplates in several languages.

Tags are exchanged between the visualization and the control program via static tags that are stored and updated in the instance data block belonging to the module FB call.

This is a 1:1 relationship between module component (FB) and faceplate (FP).

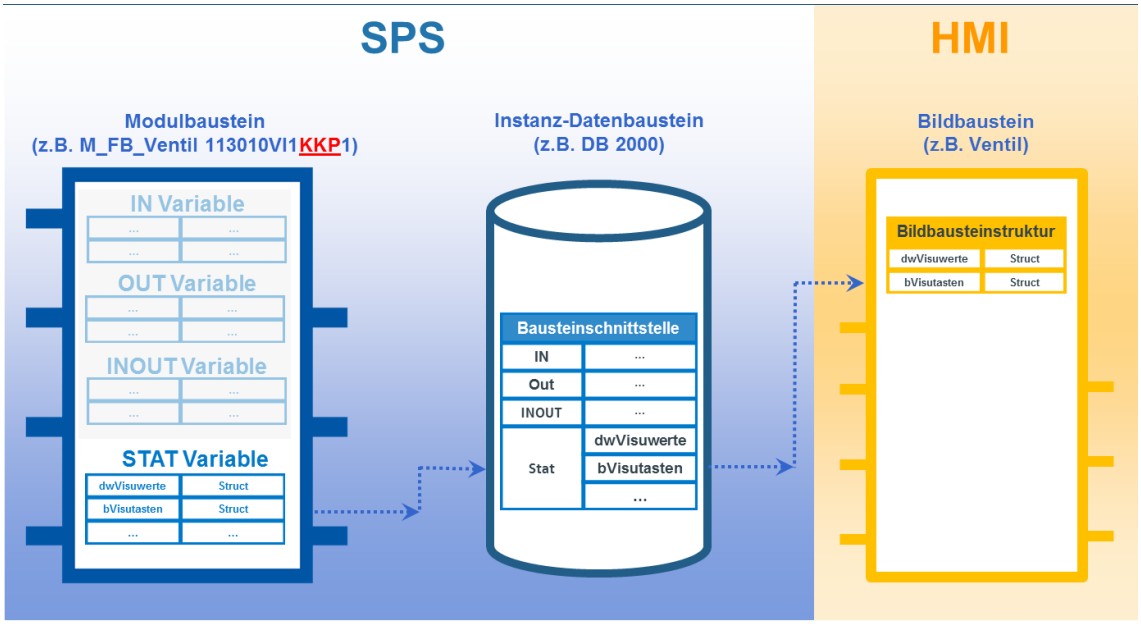


Fig. 3.7 Principle of data exchange between PLC and HMI

3.2.2 Faceplates for detailed screens

Faceplates for detailed screens should be provided for displaying detailed information as an addition to "03-pages or root screen", for example, drives and reading points.

If a module component requires a complex display and operating option, an associated faceplate must be created for it for a detailed screen, which is stored in the project library.

Texts and message texts are stored directly in the faceplates in several languages.

Tags are exchanged between the visualization and the control program via static tags that are stored in the instance data block belonging to the module FB call.

The connection to the corresponding instance data block is made at runtime, by transferring the link to the corresponding instance faceplate when the calling faceplate opens the detailed screen.

This is a 1:1 relationship between module component (FB) and faceplate (FP).

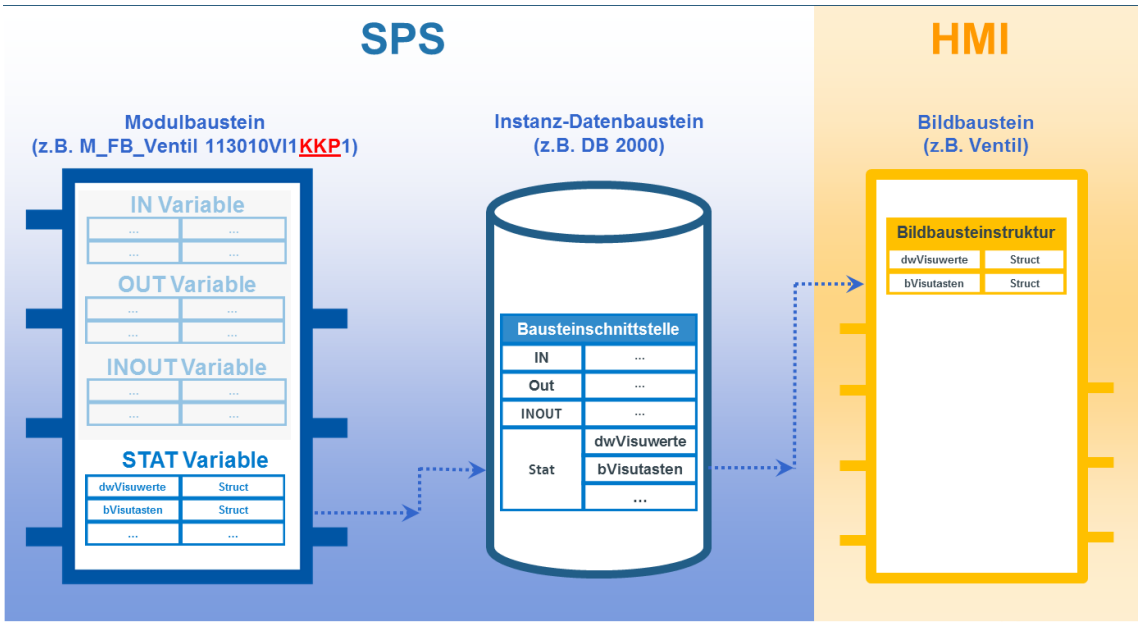


Fig. 3.8 Relationship between instance data block and detailed screen

3.2.3 Languages and project texts

The following languages must be provided for the texts and project texts:

Language
German (Germany)
Chinese (PRC)
English (US)
Polish (Poland)
Portuguese (Brazil)
Russian (Russia)
Slovak (Slovakia)
Spanish (international)
Czech (Czech Republic)
Turkish (Turkey) ???
Bulgarian (Bulgaria) ???
French (France) / without translation
Dutch (Belgium) / without translation
Hungarian (Hungary) / without translation

Tab. 3.2 Languages in the faceplates

The item "project texts" is used in the navigation tree for translation. The translations of the languages can be entered here in the individual columns.

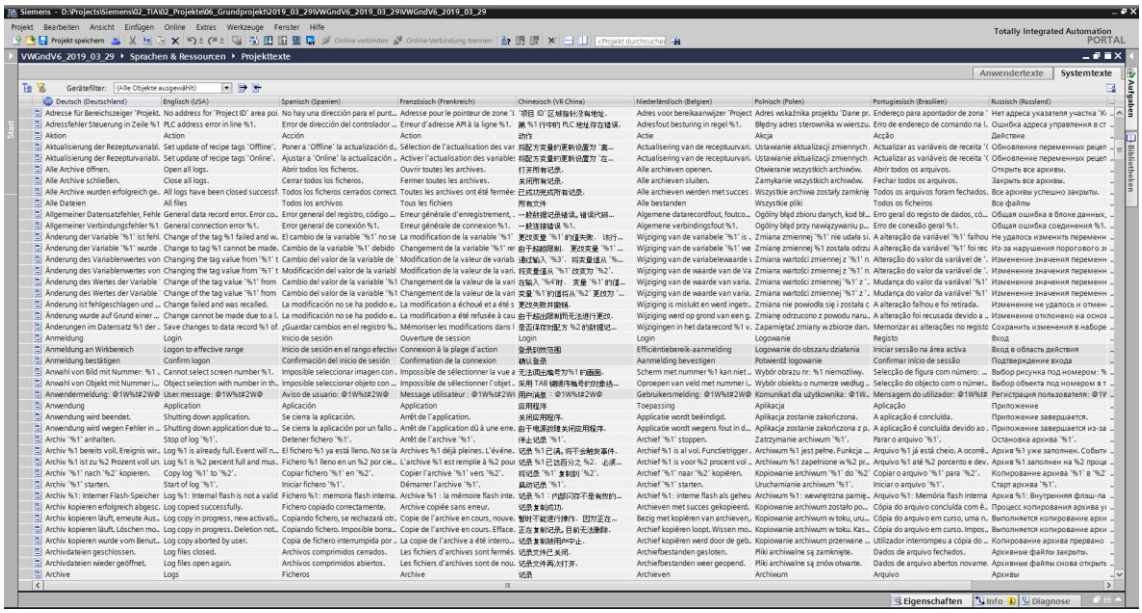


Fig. 3.9 Overview of the project texts

3.3 SiVArc

Screens for the operating and diagnostic pages are shown in the visualization. The individual pages are generated largely autonomously by the SiVArc (SiVArc = **Si**emens **V**isualization **Arc**hitect).

The basis for this are visualization objects in the form of faceplates and layouts that are generated via screen rules.

The SiVArc creates the individual screens using the automatic analysis of the S7 project.

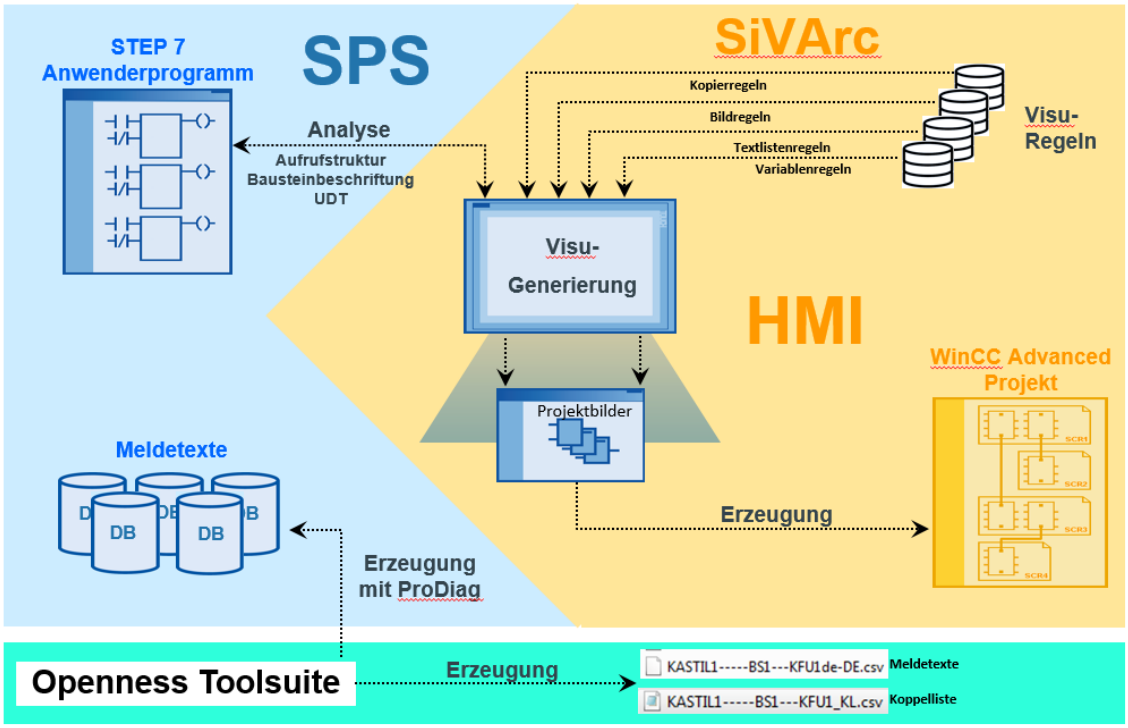


Fig. 3.10 SiVArc screen generation overview

3.4 Documentation

3.4.1 Block help

A directory with the structure **"…\03_Dokumentation\de-DE\…"** must be created so that corresponding online help is available during the configuration of the S7 logic for the project-specific function blocks via the function keys **"[Shift] + [F1]"**.

Corresponding subfolders are required for further languages.

The languages German (de-DE), English (en-US), Spanish (es-ES) and Chinese (zh-CN) are currently provided.

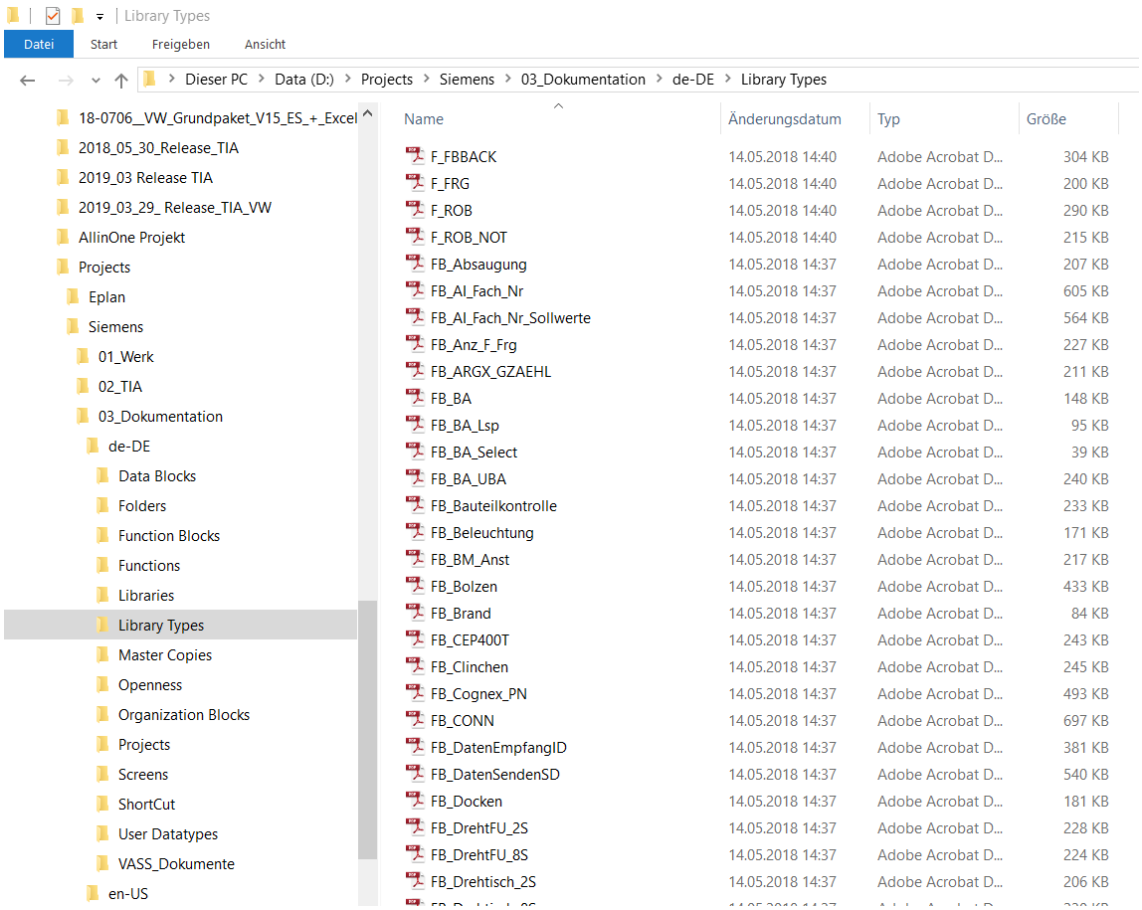


Fig. 3.11 VASS_Online_help

Note

Please observe the conventions in the TIA Portal for the creation and use of the user-defined documentation.

Open the TIA Portal help, enter the search term "Provide user-defined documentation" and go to the conventions using the link.

3.4.2 Hardware configurations

All default settings for the devices used are described in the document "Hardware configurations".

Fig. 3.12 Hardware configurations

3.4.3 Operating and visualization information

If there is a new faceplate and a new detailed screen, the operating functionality must be described in the document "Operating and visualization information".

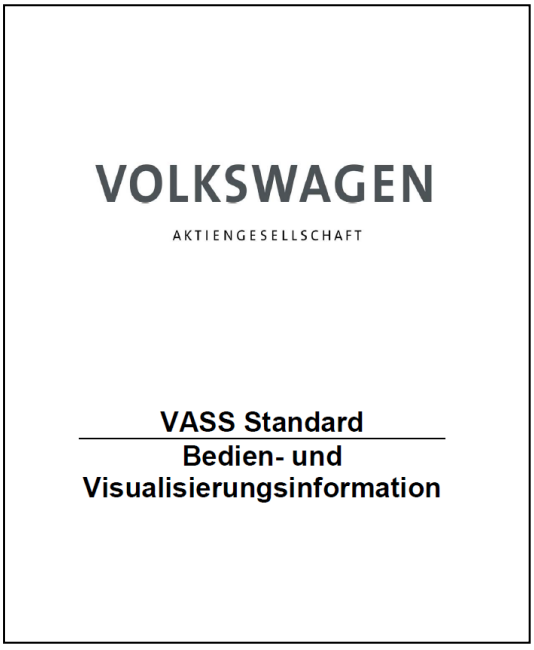


Fig. 3.13 Operating and visualization information

3.4.4 Device parameterization

If the module components created are dependent on a specific parameterization or configuration of the device you are controlling, the settings required on the device must be described in a document.

4 Specifications for brand and plant-specific standards

The VASS standard is designed in such a way that separate supplemental packages can be created for the VASS package of the Volkswagen brand as well as individual plants.

They must be created in the same structure so that they can be smoothly integrated into the projects to be created.

This also enables the possibility to transfer to the VASS package.

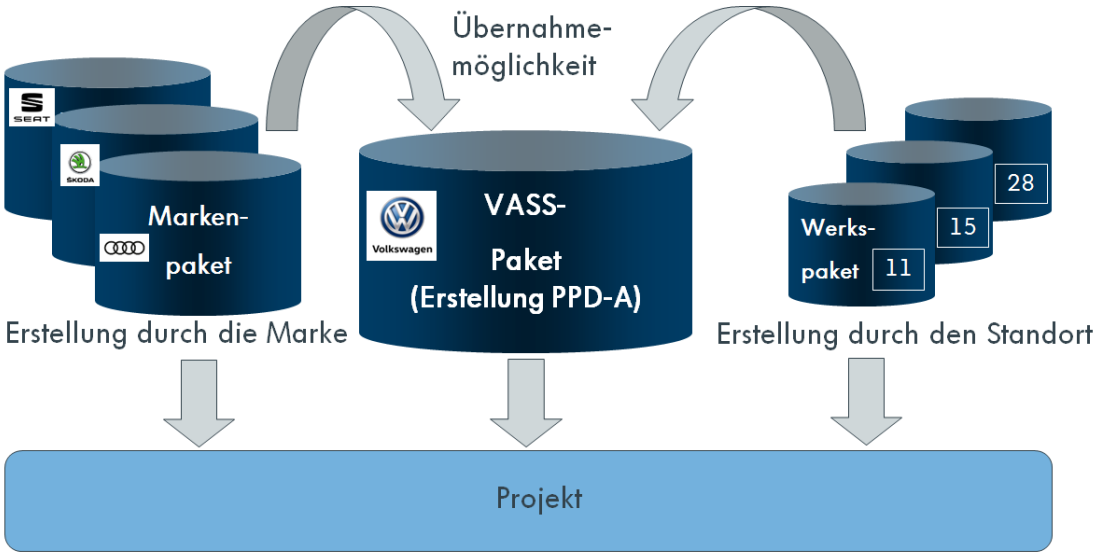


Fig. 4.1 VASS data

Identifiers

Unique names must be used for the packages and individual elements.

Number ranges

Number ranges are assigned for the individual brands and plants to prevent double assignments.

Note

During implementation, consultation between the brands and sites with the standardizing electrical engineering department of the Volkswagen brand is required!

Designation system

	Group / brand VW
TIA library	VASS_V6_VW_yyyy_mm_dd
TIA library minicontroller	VASS_V6_VW_7Z_yyyy_mm_dd
TIA basic project	VWGndV6_yyyy_mm_dd
TIA symbols	VWSymbolikV6_yyyy_mm_dd
TIA device catalog	VWGKV6_yyyy_mm_dd
TIA Block Help	VWBausteinhilfenV6_DE_yyyy_mm_dd

Tab. 4.1 Designation system "Group / Brand VW"

Abbreviation for other brands:

- AU = AUDI
- PO = Porsche
- SE = SEAT
- SK = SKODA
- VN = Volkswagen Nutzfahrzeuge

	Brand [1)]
TIA library	VASS_V6_SK_yyyy_mm_dd
TIA library minicontroller	VASS_V6_SK_7Z_yyyy_mm_dd
TIA basic project	SKGndV6_yyyy_mm_dd
TIA symbols	SKSymbolikV6_yyyy_mm_dd
TIA device catalog	SKGKV6_yyyy_mm_dd
TIA Block Help	SKBausteinhilfenV6_DE_yyyy_mm_dd

Tab. 4.2 Designation system "Brand"

TIA library	Plant [2)]
TIA library minicontroller	VASS_V6_11_yyyy_mm_dd
TIA basic project	VASS_V6_11_7Z_yyyy_mm_dd
TIA symbols	11GndV6_yyyy_mm_dd
TIA device catalog	11SymbolikV6_yyyy_mm_dd
TIA Block Help	11GKV6_yyyy_mm_dd
TIA library	11BausteinhilfenV6_DE_yyyy_mm_dd

Tab. 4.3 Designation system "plant"

1) other brands, in this case, for example SKODA (SK)

2) in this example Wolfsburg (11)

Note

If individual modules without a direct reference to the Group, brand or site must be created, the designation is based on the module to be created.

The blocks for all standardized functional units and technologies which are used universally are provided via the **VASS library**.

They are in the range **FB100 - FB799**.

As **additional blocks** may be required, **related to the project in question**, for individual brands and the individual VW sites, which are developed locally, **separate number ranges** are provided for this.

This enables the blocks to be exchanged without conflict throughout the Group as required.

Group, SEAT and SKODA

				Group	**Group**	**SEAT**	**SKODA**
FB	Standard range	Library	General	100 … 199	3100 … 3199	4100 … 4199	5100 … 5199
			Robot	200 …249	3200 … 3249	4200 … 4249	5200 … 5249
			Robot processes	250 … 299	3250 … 3299	4250 … 4299	5250 … 5299
			Machine processes	300 … 349	3300 … 3349	4300 … 4349	5300 … 5349
			Drive technology	350 … 399	3350 … 3399	4350 … 4399	5350 … 5399
			Clamping technology	400 … 449	3400 … 3449	4400 … 4449	5400 … 5449
			Internal conveyor equipment	450 … 499	3450 … 3499	4450 … 4499	5450 … 5499
			Technologies	500 … 549	3500 … 3549	4500 … 4549	5500 … 5549
			Type management	550 … 599	3550 … 3599	4550 … 4599	5550 … 5599
			ZAU	600 … 619	3600 … 3619	4600 … 4619	5600 … 5619
			PDD/PDE	620 … 639	3620 … 3639	4620 … 4639	5620 … 5639
			Sequencer blocks	640 … 649	3640 … 3649	4640 … 4649	5640 … 5649
			Alarm system	650 … 659	3650 … 3659	4650 … 4659	5650 … 5659
			Tool blocks	660 … 699	3660 … 3699	4660 … 4699	5660 … 5699
			External conveying equipment	700 … 749	3700 … 3749	4700 … 4749	5700 … 5749
			Assemblies	750 … 799	3750 … 3799	4750 … 4799	5750 … 5799
		System		800 … 899			
	F range	Library		930 … 999			
		System		1000			

Tab. 4.4 Reserve, SEAT and SKODA

VWN, Porsche and Audi

				VWN	Porsche	Audi
FB	Standard range	Library	General	6100 … 6199	7100 … 7199	8100 … 8199
			Robot	6200 … 6249	7200 … 7249	8200 … 8249
			Robot processes	6250 … 6299	7250 … 7299	8250 … 8299
			Machine processes	6300 … 6349	7300 … 7349	8300 … 8349
			Drive technology	6350 … 6399	7350 … 7399	8350 … 8399
			Clamping technology	6400 … 6449	7400 … 7449	8400 … 8449
			Internal conveyor equipment	6450 … 6499	7450 … 7499	8450 … 8499
			Technologies	6500 … 6549	7500 … 7549	8500 … 8549
			Type Management	6550 … 6599	7550 … 7599	8550 … 8599
			ZAU	6600 … 6619	7600 … 7619	8600 … 8619
			PDD/PDE	6620 … 6639	7620 … 7639	8620 … 8639
			Sequencer blocks	6640 … 6649	7640 … 7649	8640 … 8649
			Alarm system	6650 … 6659	7650 … 7659	8650 … 8659
			Tool blocks	6660 … 6699	7660 … 7699	8660 … 8699
			External conveying equipment	6700 … 6749	7700 … 7749	8700 … 8749
			Assemblies	6750 … 6799	7750 … 7799	8750 … 8799
		System				
	F range	Library				
		System				

Tab. 4.5 VWN, Porsche and Audi

VW sites

				VW sites
FB		Library	VW Wolfsburg	2000 … 2049
			VW Emden	2100 … 2149
			VW Zwickau	2250 … 2299
			VW Slovakia	2300 … 2349
			VW Pamplona	2350 … 2399
			VW Mexico	2400 … 2449
			VW Brazil	2450 … 2499
			Reserve	2500 … 2549
			VW Chattanooga	2550 … 2599
			VW South Africa	2600 … 2649
			VW India	2650 … 2699
			SVW	2700 … 2749
			Reserve	2750 … 2799
			FAW	2800 … 2849
			Reserve	2850 … 2899
			Reserve	2900 … 2949
			Reserve	2950 … 2999

Tab. 4.6 Designation system "VW sites"

The number ranges are managed by the individual brands and VW sites.

Regular feedback to the standardizing electrical engineering department of the VW brand must be provided.

Note

If a block is to be created by a system supplier within the framework of his project, he must assign it in the number range for users (10 – 99).

5 Development of a VASS module

5.1 Working in the VASS basic project

In the VASS basic project, all required extensions must be made to ensure the full functioning of the VASS module.

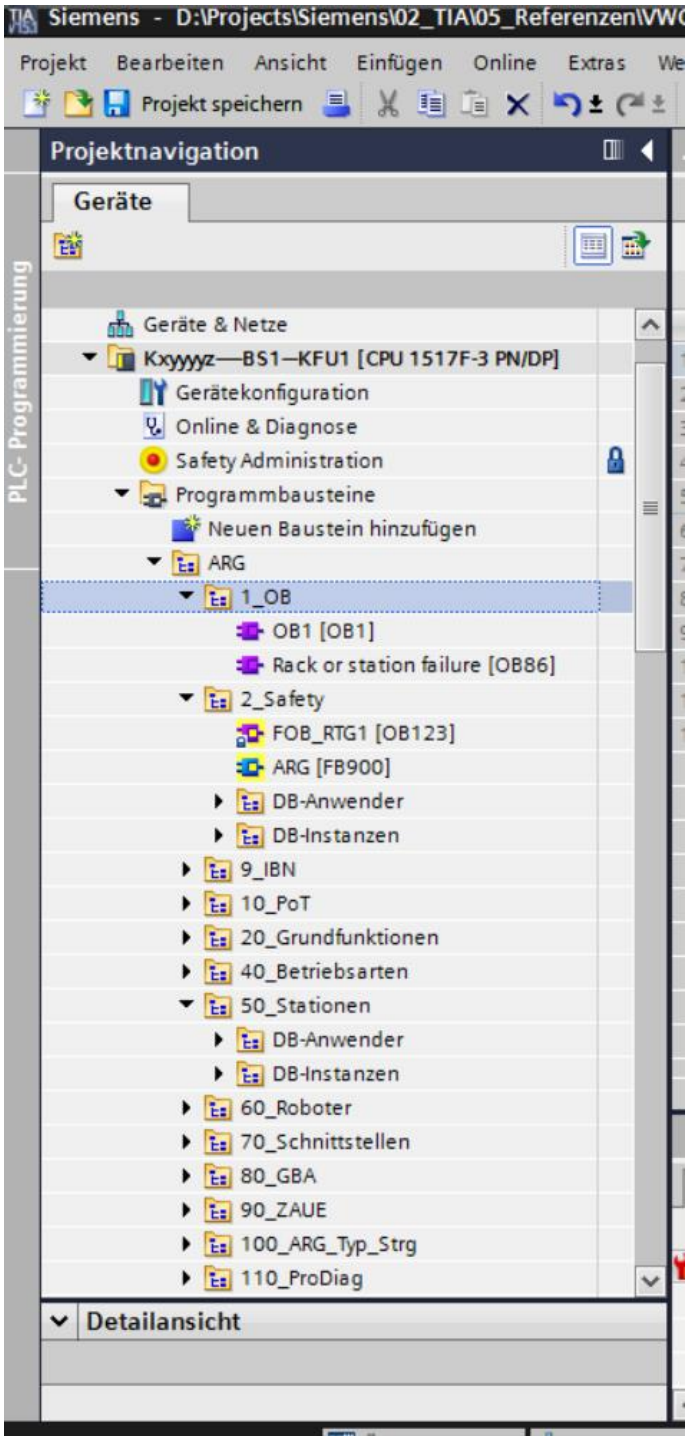


Fig. 5.1 VASS basic project

5.1.1 Integrating the GSDML file into the HW Configuration

The GSDML file is supplied by the device manufacturer or can be downloaded from the service page.

xxxx		GSDML-V23-Siemens-Sinamics_G120D-20111221.zip (17 KB)
6SL3544-0FB2x-1Fx0	V4.6	GSDML_V225_V23_CU240D-PN_V4-6.zip (33 KB)

Fig. 5.2 GSDML file

For integration in the device catalog, call the application "Manage general station description files (GSD)" via "Options" in the TIA Portal.

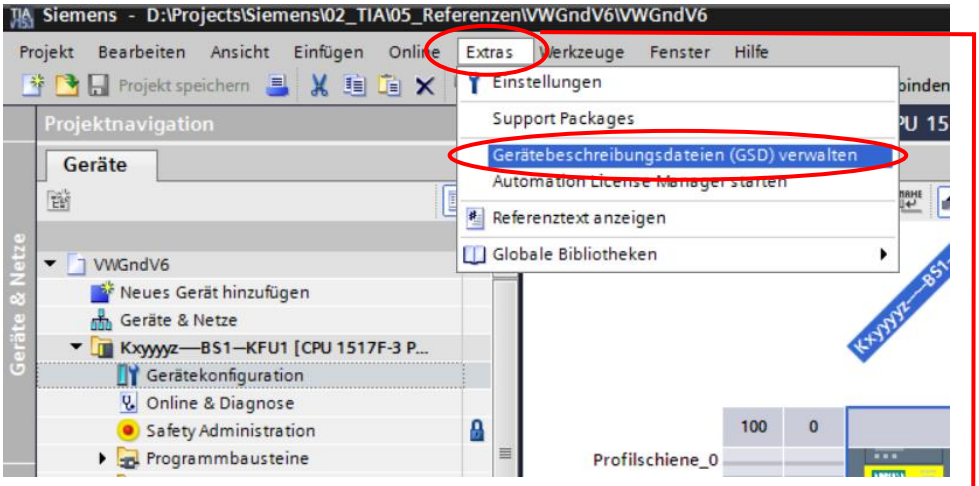


Fig 5.3 Calling the application "Manage general station description files (GSD)"

In the "Installed GSDs" tab, the corresponding "GSDML file" must be selected and then installed.

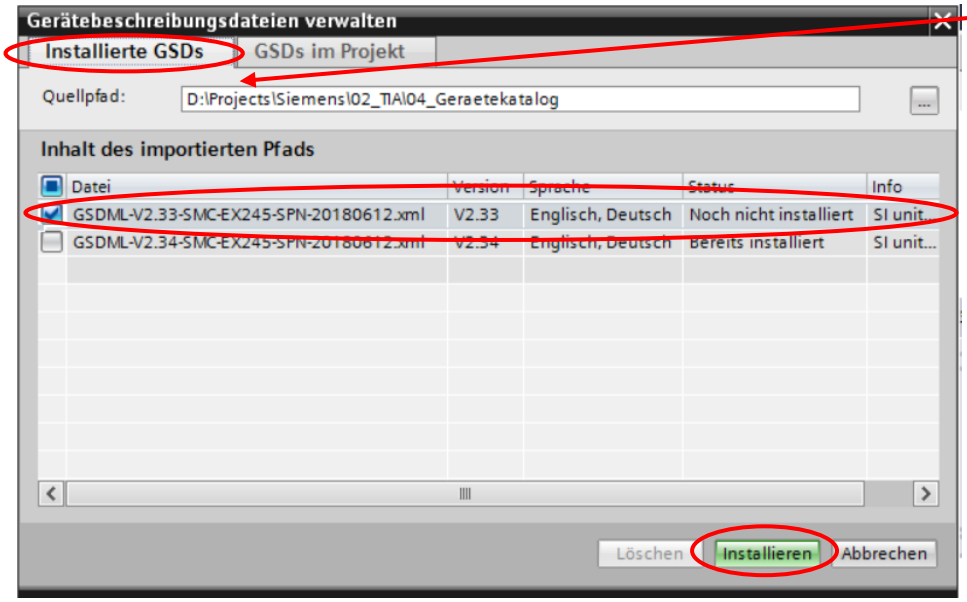


Fig. 5.4 Select and install "GSDML file"

5.1.2 Setting up a device in HW Config

The device is selected in the catalog and then dragged to the PROFINET line with drag and drop.

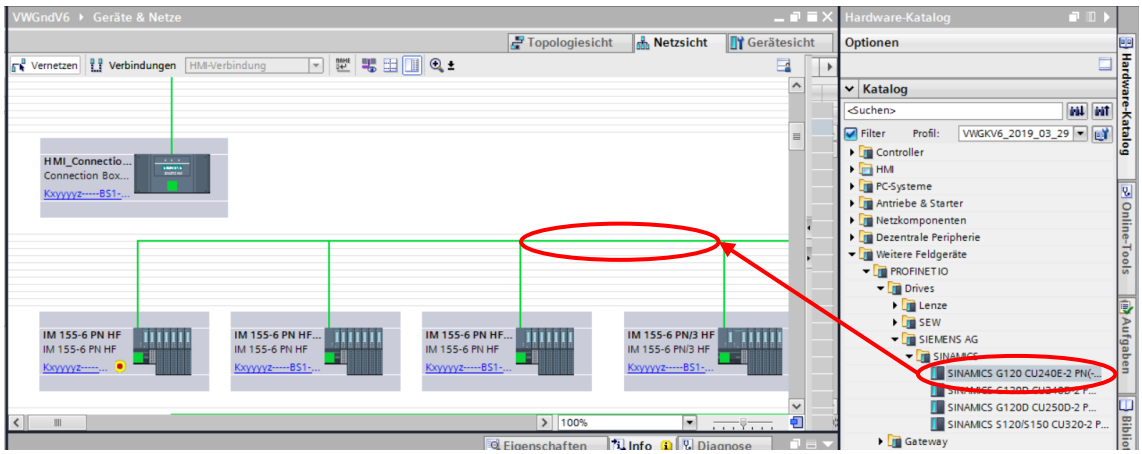


Fig. 5.5 Setting up a device in HW Config (1)

The defined IP address is then set.

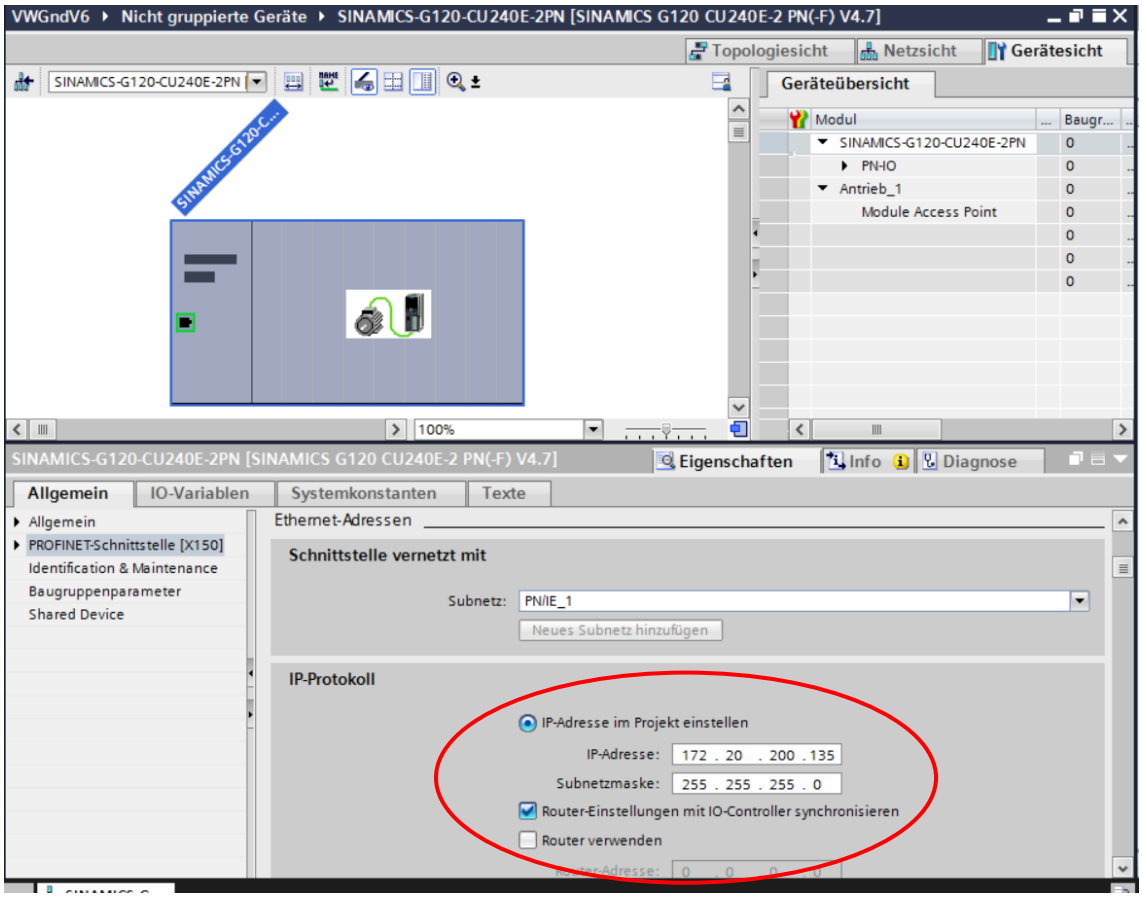


Fig. 5.6 Setting up a device in HW Config (2)

All further parameters (I/O - addresses, device name, device number, etc.) can then be edited:

5.2 Working in the device catalog

The desired catalog profile can then be selected in the "Devices & Networks" project navigation.

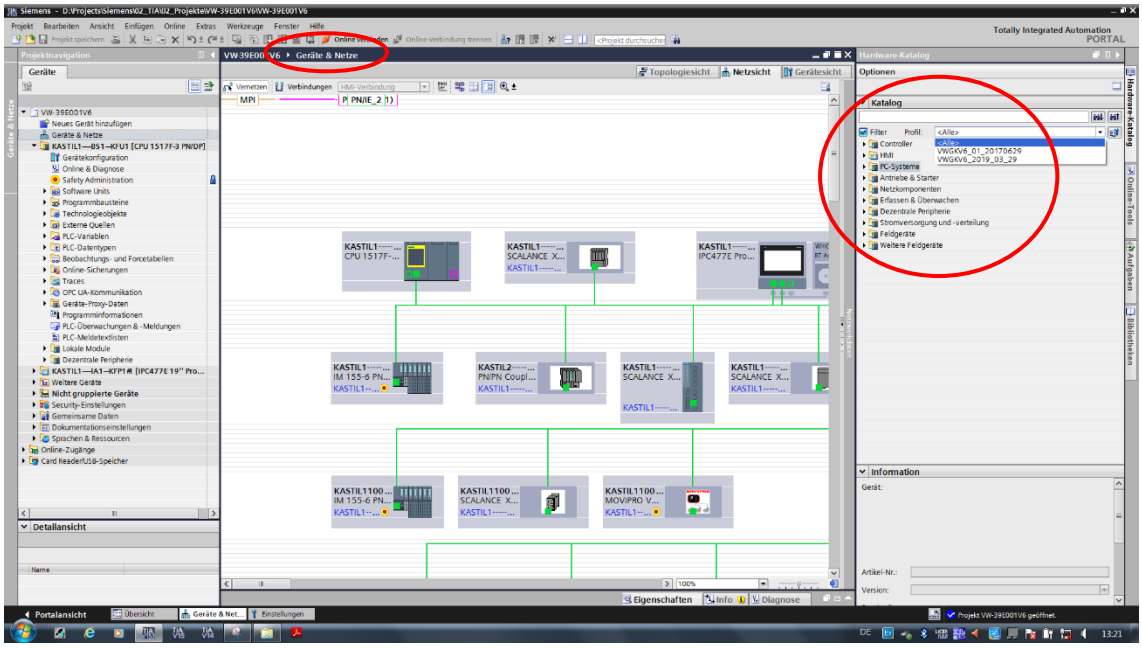


Fig. 5.7 Select the desired catalog under "Devices & Networks"

To create a separate profile, a new profile must first be created.

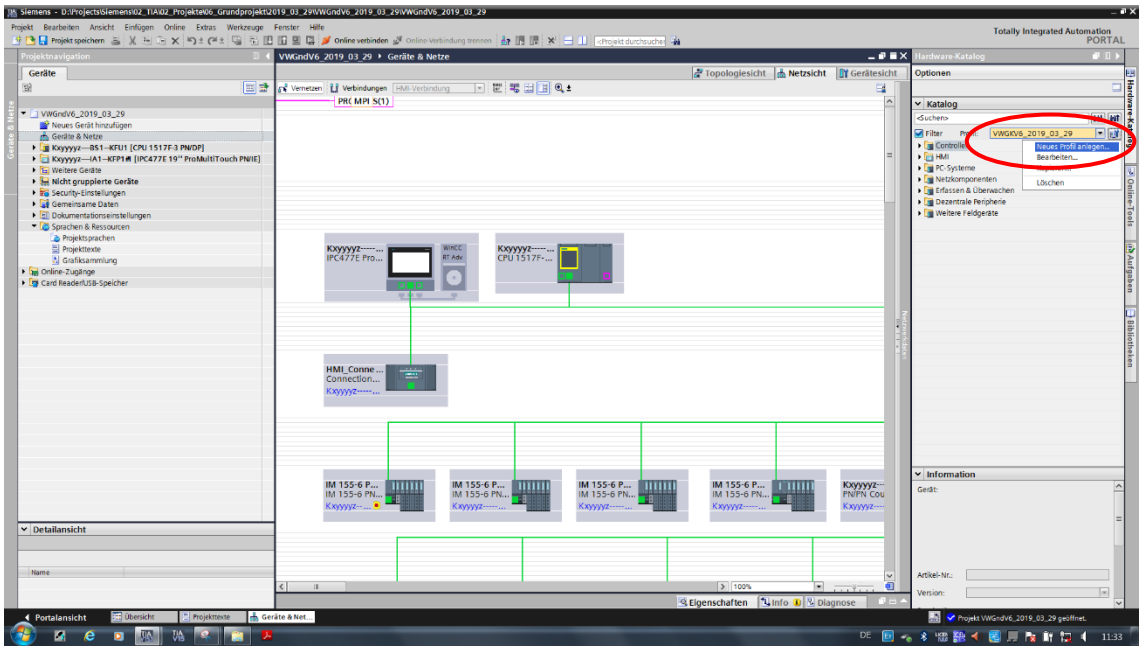


Fig. 5.8 Create new catalog profile

The necessary elements must be added to the new profile.

It is important that the elements as described in Chapter 5.1 "Working in the VASS basic project" were installed before they can be added to the profile.

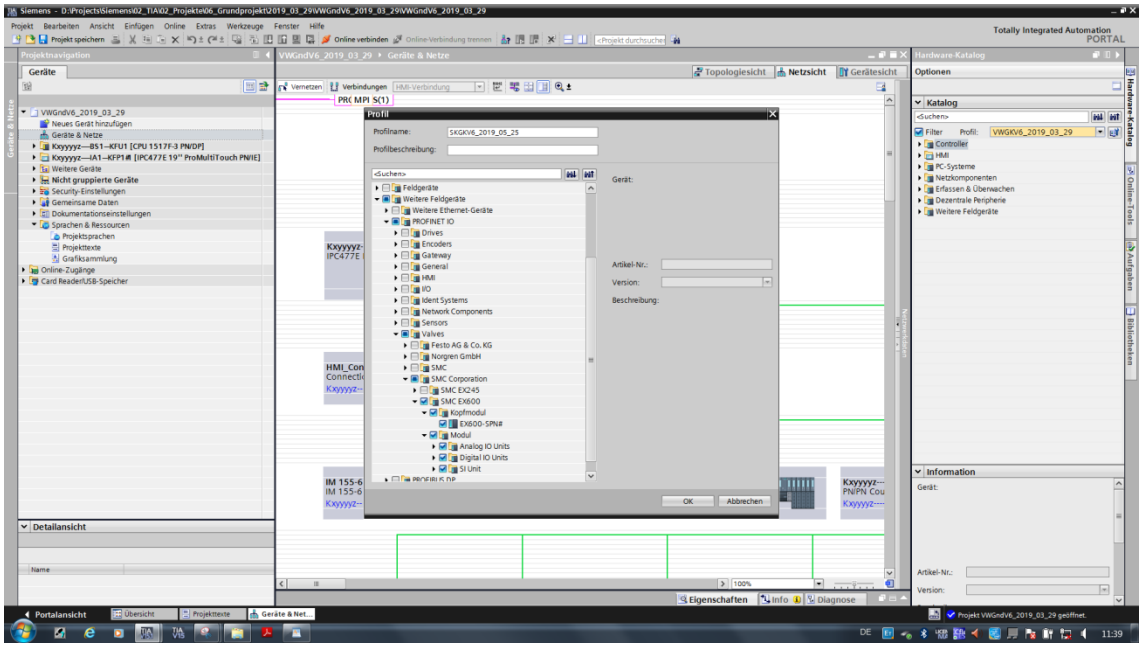


Fig. 5.9 Add elements

The profile name must be assigned to the new profile in accordance with the convention in Chapter 4 "Specifications for brands and plant-specific standards",

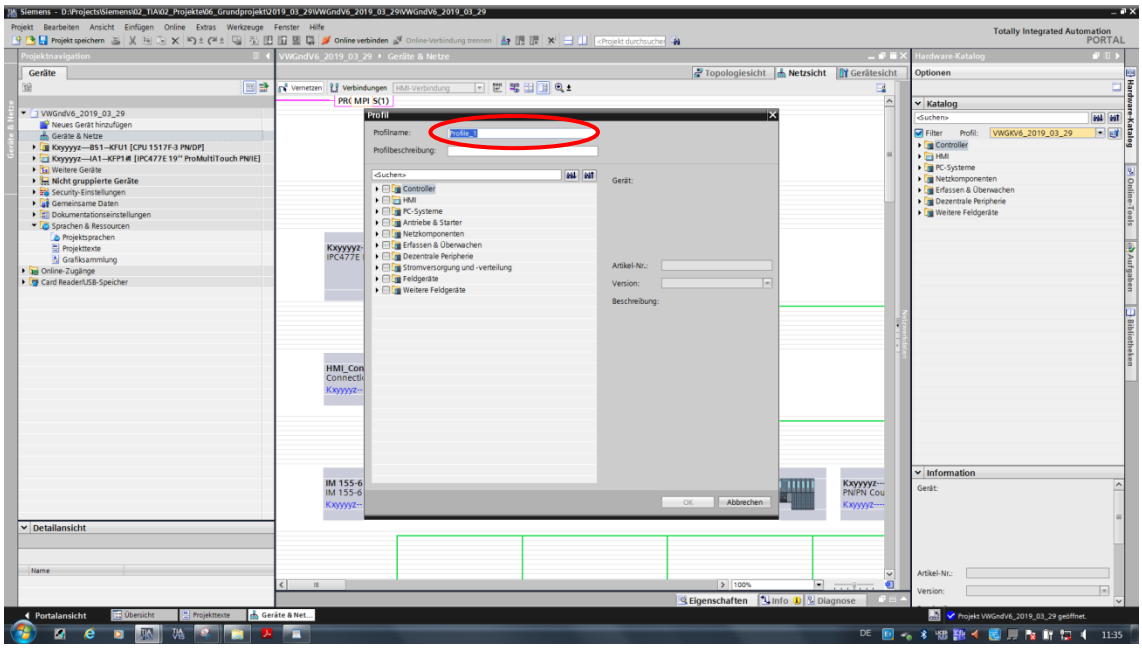


Fig. 5.10 Assign catalog profile name (1)

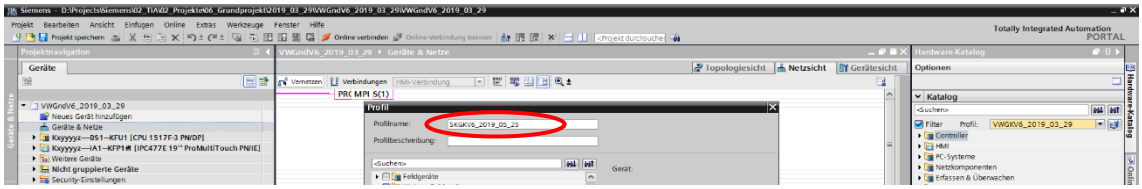


Fig. 5.11 Assign catalog profile name (2)

The dialog is then closed with "OK".

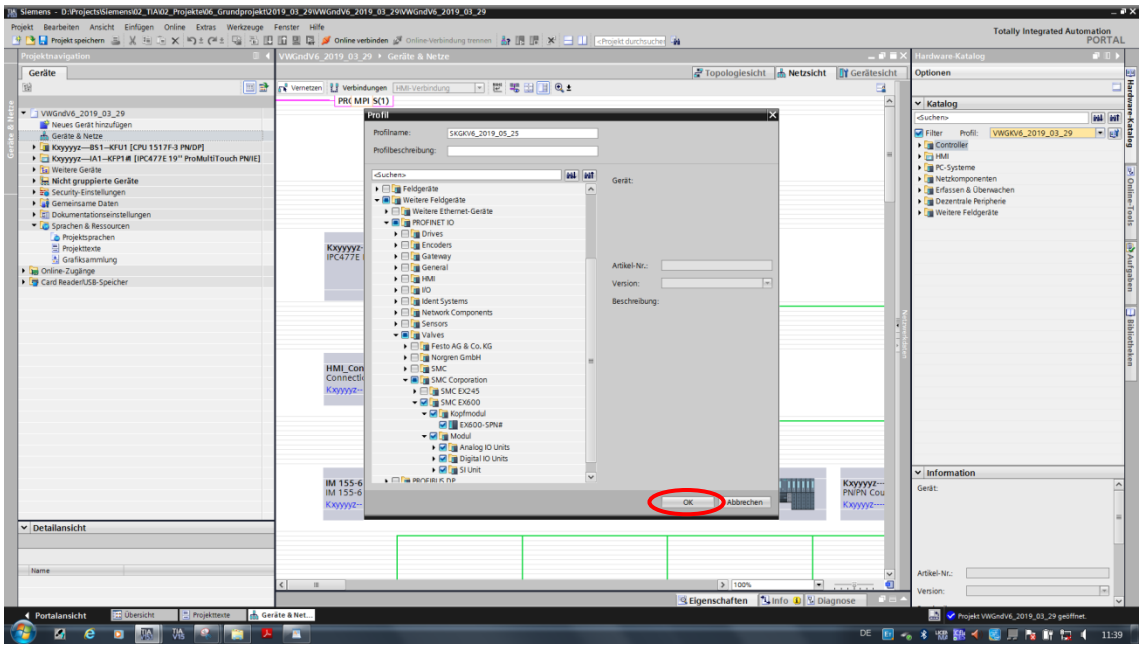


Fig. 5.12 End creation of catalog profile

As the profile is initially only available locally, it still has to be imported.

For this, open the "Settings" under "Options".

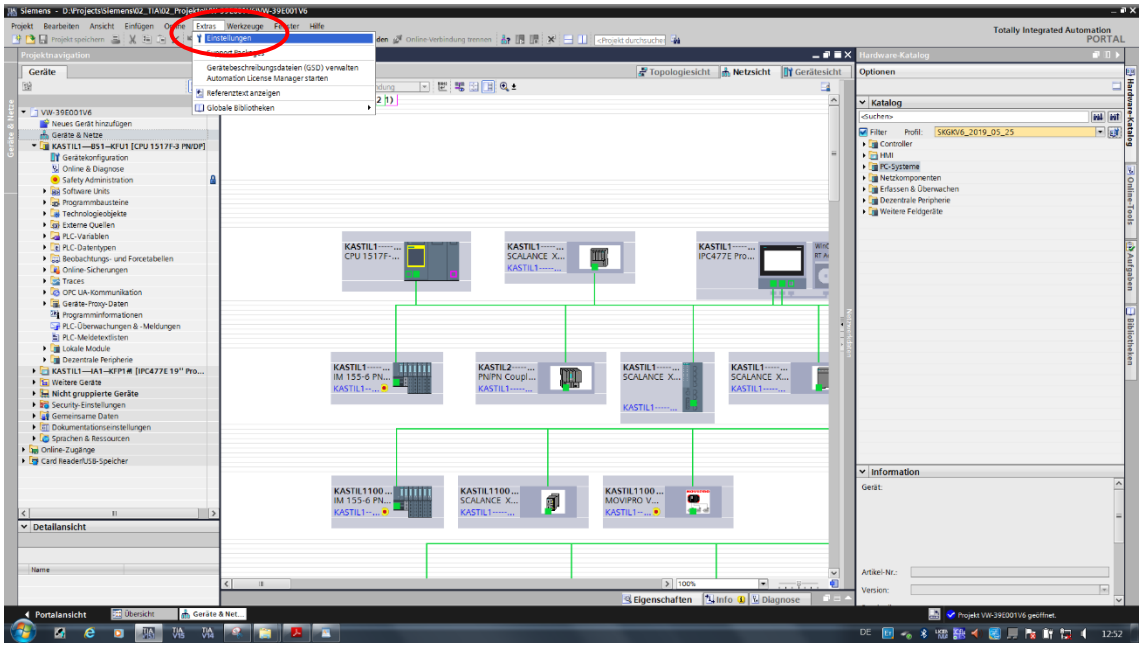


Fig. 5.13 Open settings for TIA

The option "Export settings" must be selected in the area "Import/Export settings".

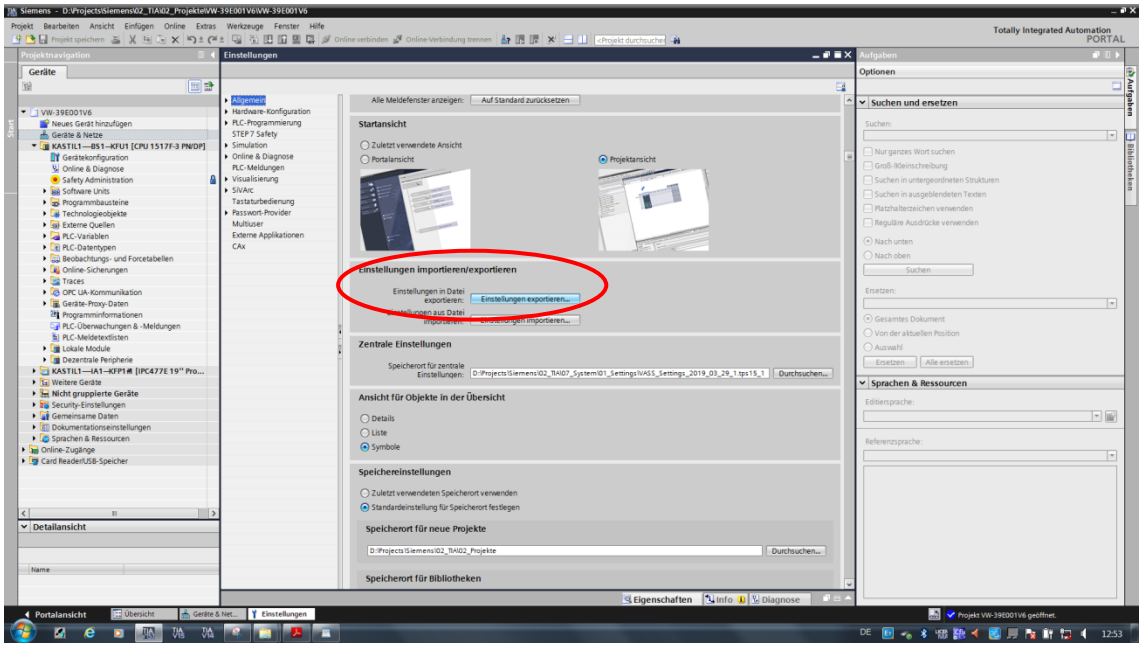


Fig. 5.14 TIA settings export dialog

In the dialog window that opens, only the profile of the device catalog created may be selected; all other check boxes must be deselected.

The profile name of the device catalog is entered in the field "File name" and the file is exported with "Export".

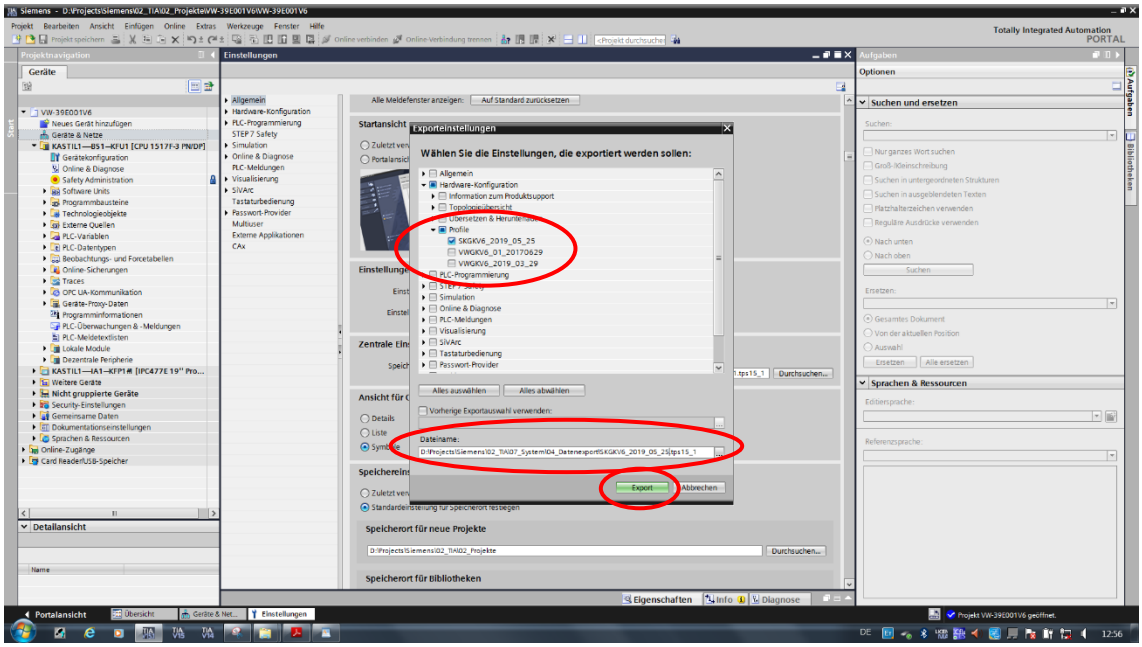


Fig. 5.15 Selecting a catalog profile for export

The successful export is then acknowledged.

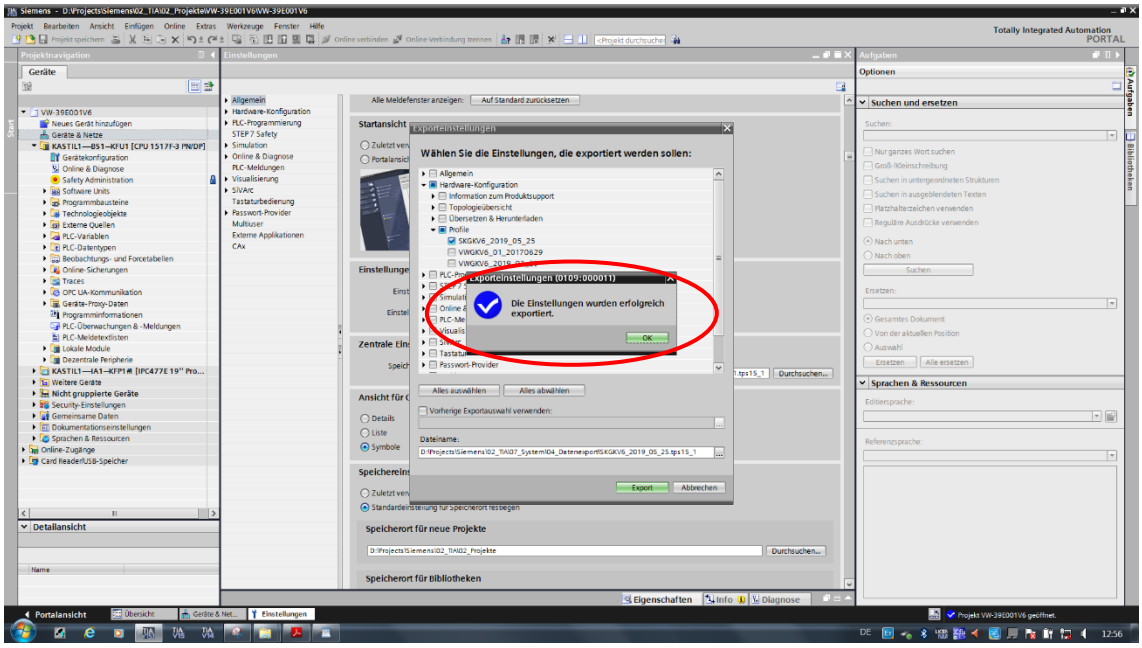


Fig. 5.16 Acknowledging successful export

The exported device catalog can then be accessed under the path "D:/Projects/Siemens/02_TIA/07_System/04_Datenexport".

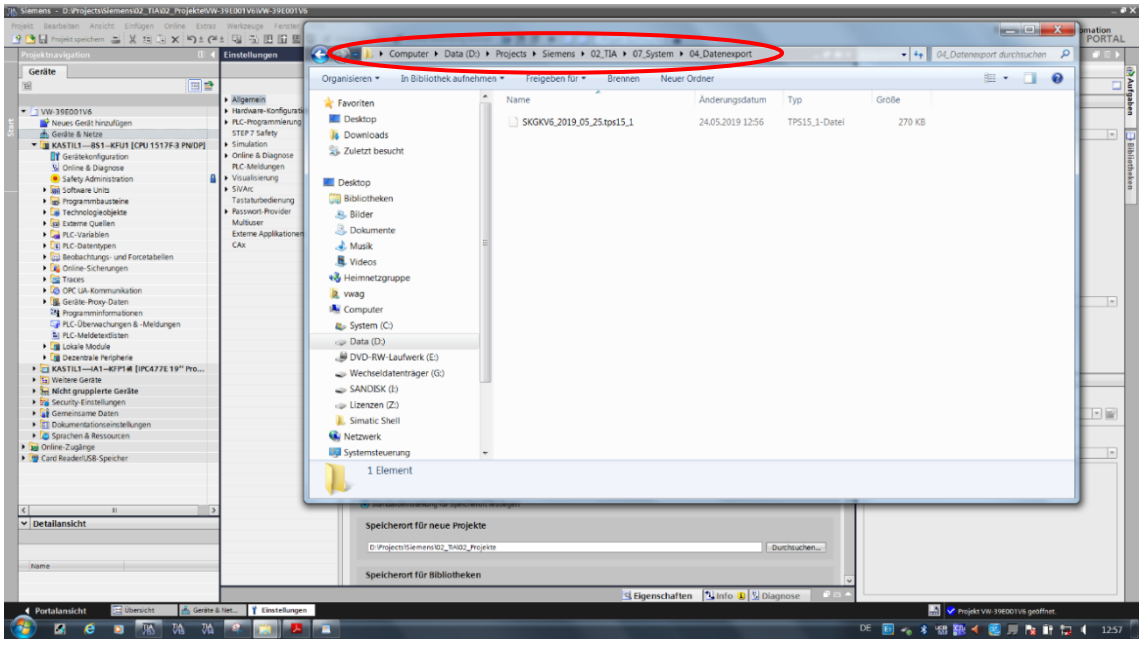


Fig. 5.17 Storage path of export file

5.3 Working in the TIA library

Before a block can be created for a new module, the following specifications must be made:

- block and, if necessary, UDT number
- Block name
- Block family name
- Block version number
  Numbering system for the block version number is as follows:
  Version x.y.z
  x = main version (is incremented by the standardizer in the event of a complete version change due to function extension)
  y = subversion (is incremented by the standardizer in the event of changes which were made as part of a bug fix)
  z = change index (is incremented in the event of code changes on the site, if required, by the commissioning company)

Note

In the event of the distribution of a newly created block, the third position (z) must therefore always be versioned with 0.

5.3.1 Creating symbols

This is only required for blocks which work with address areas at the input or output parameters.

This can be done in the form of a definition of an I/O data type and/or as symbol table in the "XXSymbolikV06".

If the sequence and name of the input and output tags are always identical, a data type must be used. Only if flexible tags are found within the area do the inputs and outputs have to be created individually in the symbol table (XXSymbolikV06). Otherwise, only the data type should be entered.

A spreadsheet must be created for each device.

The following syntax rule should be observed for this:

"structure_technology_input/output"

Examples:

Symbol	Datentyp	Adresse	Kommentar
G120D BIN			
<<Station>>AE<<Nummer>>_A	ST_SINAMICS_BIN_A	A 0.0	Treiber SINAMICS BIN (SPS Ausgänge) PA
<<Station>>AE<<Nummer>>_E	ST_SINAMICS_BIN_E	E 0.0	Treiber SINAMICS BIN (SPS Eingänge) PE
Symbol	**Datentyp**	**Adresse**	**Kommentar**
G120D TR			
<<Station>>AE<<Nummer>>_A	ST_SINAMICS_TR_A	A 0.0	Treiber SINAMICS TR (SPS Ausgänge) PA
<<Station>>AE<<Nummer>>_E	ST_SINAMICS_TR_E	E 0.0	Treiber SINAMICS TR (SPS Eingänge) PE
Symbol	**Datentyp**	**Adresse**	**Kommentar**
G120D RB			
<<Station>>AE<<Nummer>>_A	ST_SINAMICS_RB_A	A 0.0	Treiber SINAMICS TR (SPS Ausgänge) PA
<<Station>>AE<<Nummer>>_E	ST_SINAMICS_RB_E	E 0.0	Treiber SINAMICS TR (SPS Eingänge) PE
Symbol	**Datentyp**	**Adresse**	**Kommentar**
SINAMICS SAFETY TELEGRAMM 30			
<<Station>>AE<<Nummer>>_A_F	ST_SINAMICS_SAFETY30_A	A 0.0	Treiber SINAMICS BIN (SPS Ausgänge) PA (Safety)
<<Station>>AE<<Nummer>>_E_F	ST_SINAMICS_SAFETY30_E	E 0.0	Treiber SINAMICS BIN (SPS Eingänge) PE (Safety)
Symbol	**Datentyp**	**Adresse**	**Kommentar**
SINAMICS SAFETY TELEGRAMM 900			
<<Station>>AE<<Nummer>>_A_F	ST_SINAMICS_SAFETY900_A	A 0.0	Treiber SINAMICS BIN (SPS Ausgänge) PA (Safety)
<<Station>>AE<<Nummer>>_E_F	ST_SINAMICS_SAFETY900_E	E 0.0	Treiber SINAMICS BIN (SPS Eingänge) PE (Safety)

Fig. 5.18 Creating symbols

5.3.2 Creating module FBs in SCL

5.3.2.1 Information on general structures

Data structures at which (all) the blocks are docked, e.g. "ST_Betriebsarten" (operating modes structure) are used as the "backbone" for passing on standardized information.

They are used to supply all blocks with central pre-selections and enables.

The symbol for these structures begins with "ST_".

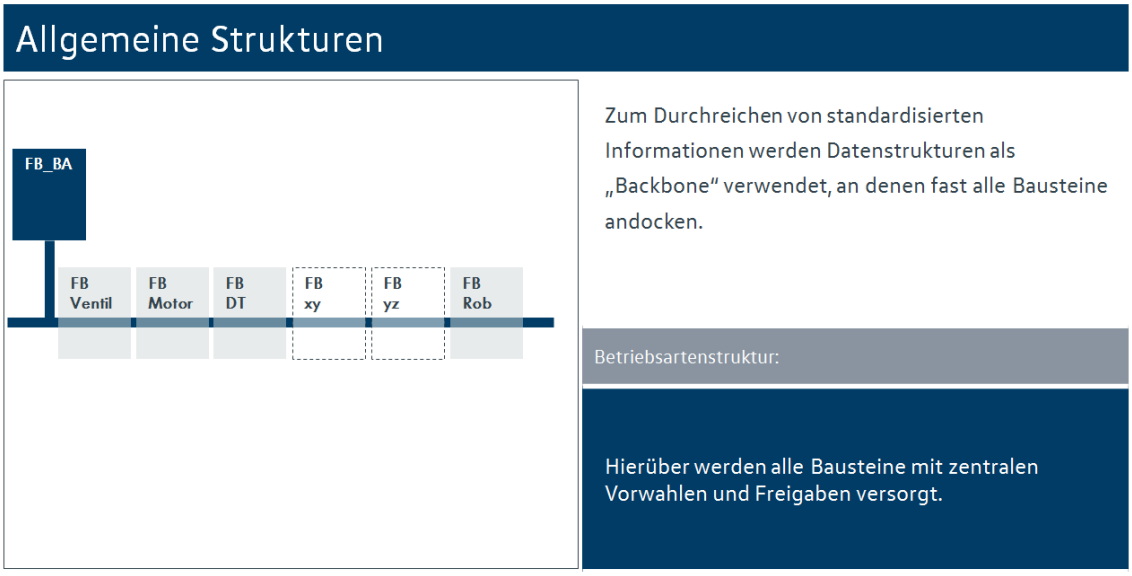


Fig. 5.19 General structures

Example:

Use of the UDT "ST_Betriebsarten" as formal operand

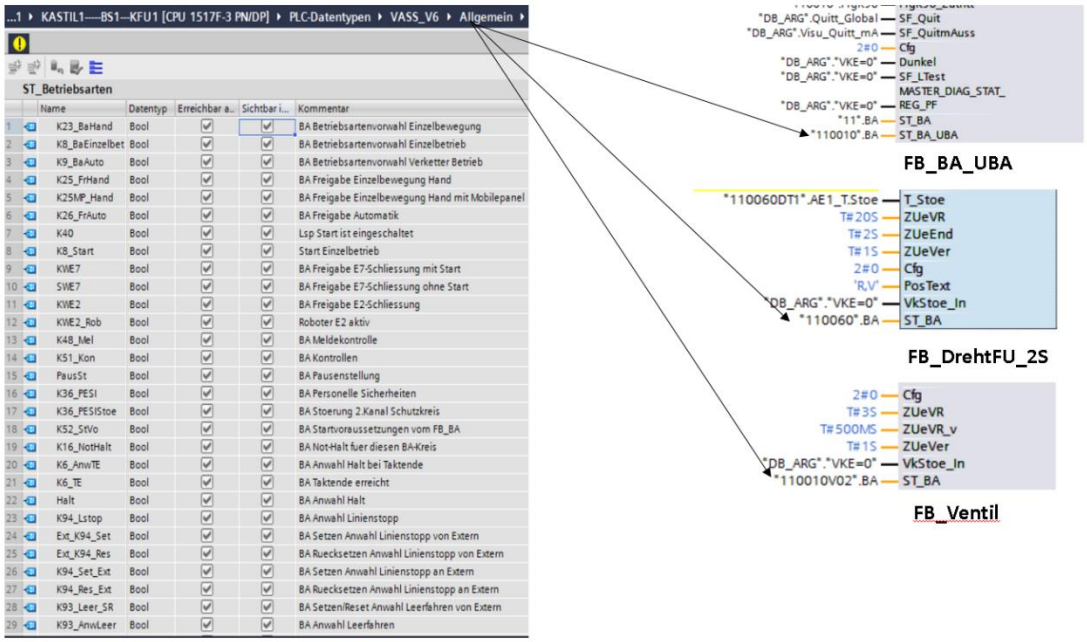


Fig. 5.20 Create UDT

5.3.2.2 Add new block

Search for one of the folders corresponding to the technology in the project navigation under "Program blocks → VASS_V6" (in this example) and select "Add new block" with a right click.

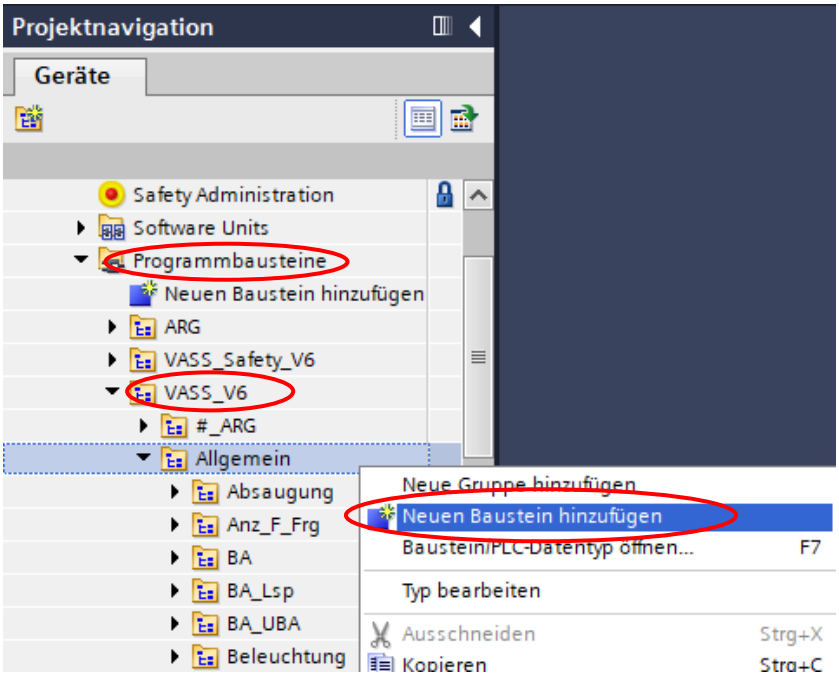


Fig. 5.21 Add new block

5.3.2.3 Pop-up "Add new block"

In the pop-up, assign the name with the prefix "FB_".

Select "SCL" as the language.

Set the type to "Function block", change the numbering from "Automatic" to "Manual" and assign the number as specified (Chapter 4 "Specifications for brand and plant-specific standards").

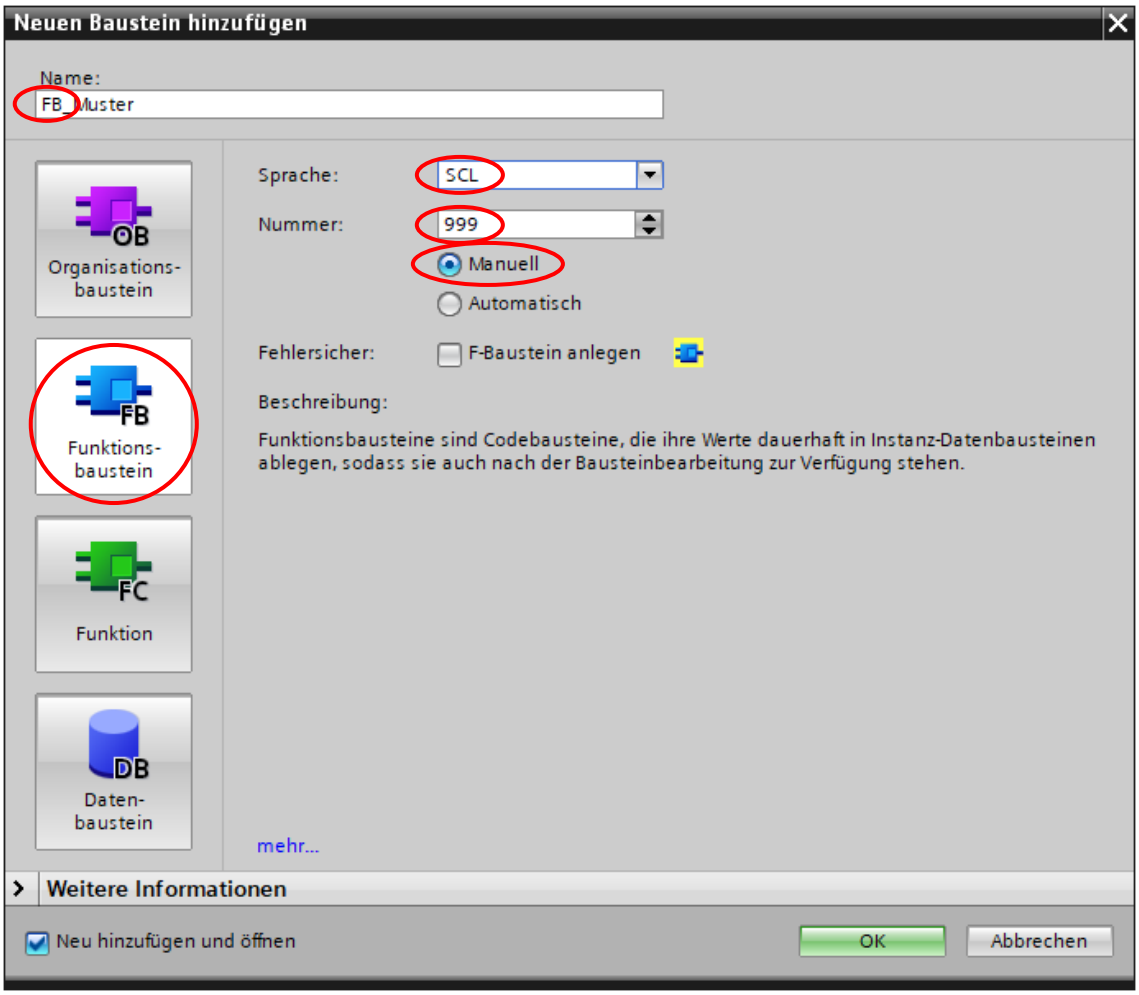


Fig. 5.22 Pop-up "Add new block"

5.3.2.4 Properties (General)

The data which was displayed previously in the pop-up window can be changed under "Properties → General" in the Inspector window of the project view.

The "type" and the "language" can, however, no longer be changed.

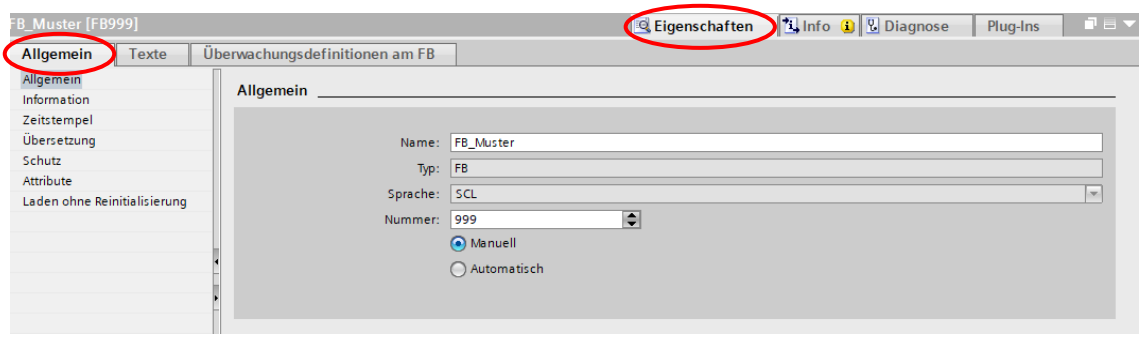


Fig. 5.23 Properties (General)

5.3.2.5 Properties information

The name of the block without the prefix "FB" should be entered under "Family" and "User-defined ID".

The creator should be entered as the author.

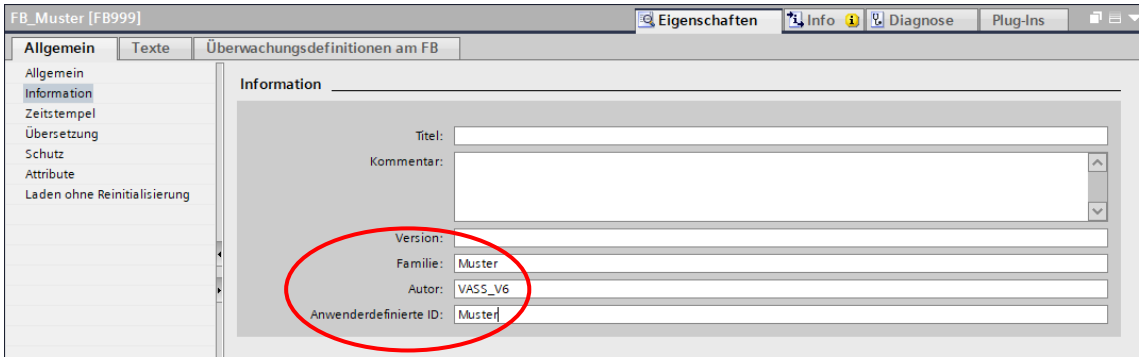


Fig. 5.24 Properties information

5.3.2.6 Creating a module component folder

A corresponding block folder must be created for the module component. This is done by right-clicking on the associated technology folder with "Add new group".

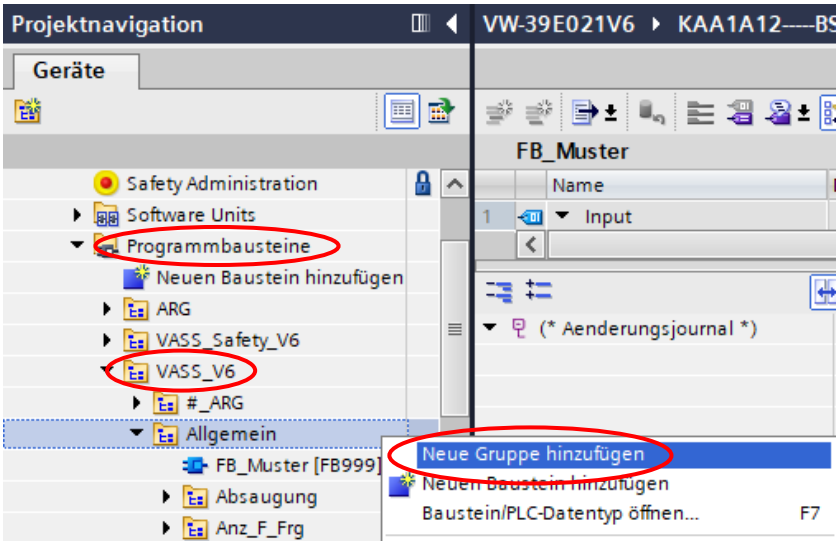


Fig. 5.25 Add new group

5.3.2.7 Rename new group

Right-click on the new new group and click "Rename".

Then enter the name of the FB without the prefix "FB_".

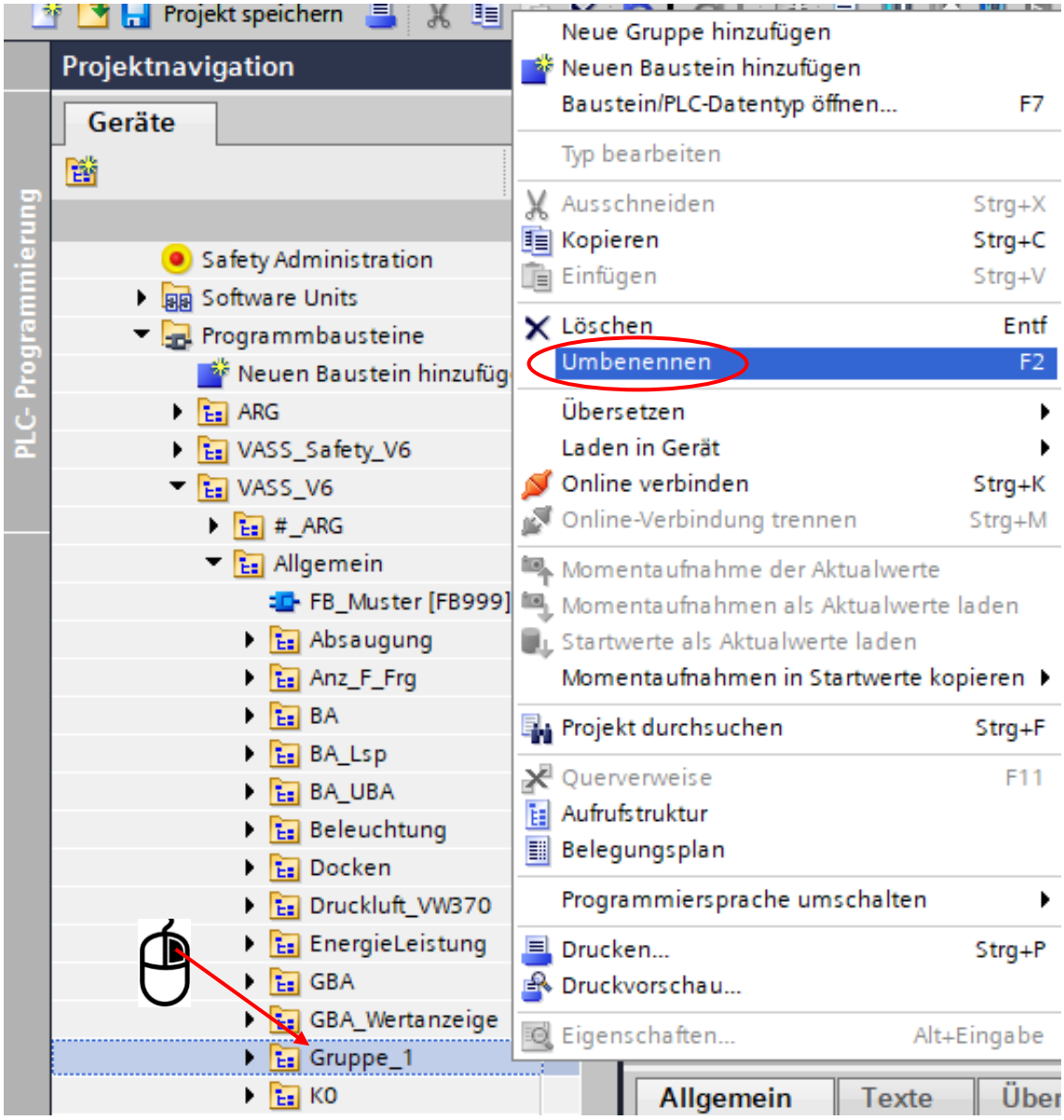


Fig. 5.26 Rename new group

5.3.2.8 Moving the module component to the module component folder

The module component created can be dragged to the new folder using "drag & drop".

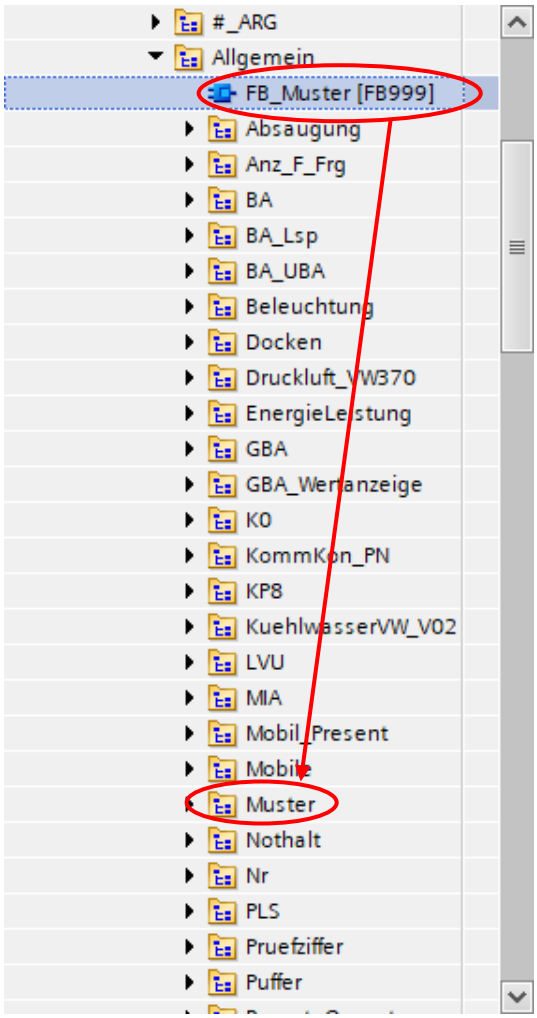


Fig. 5.27 Moving a block in the project tree

5.3.3 Changing a module component to a library type

5.3.3.1 Copying a block from the project navigation to the project library (create library type)

Use a "drag-and-drop" operation to move FB_Muster to the folder in the project library.

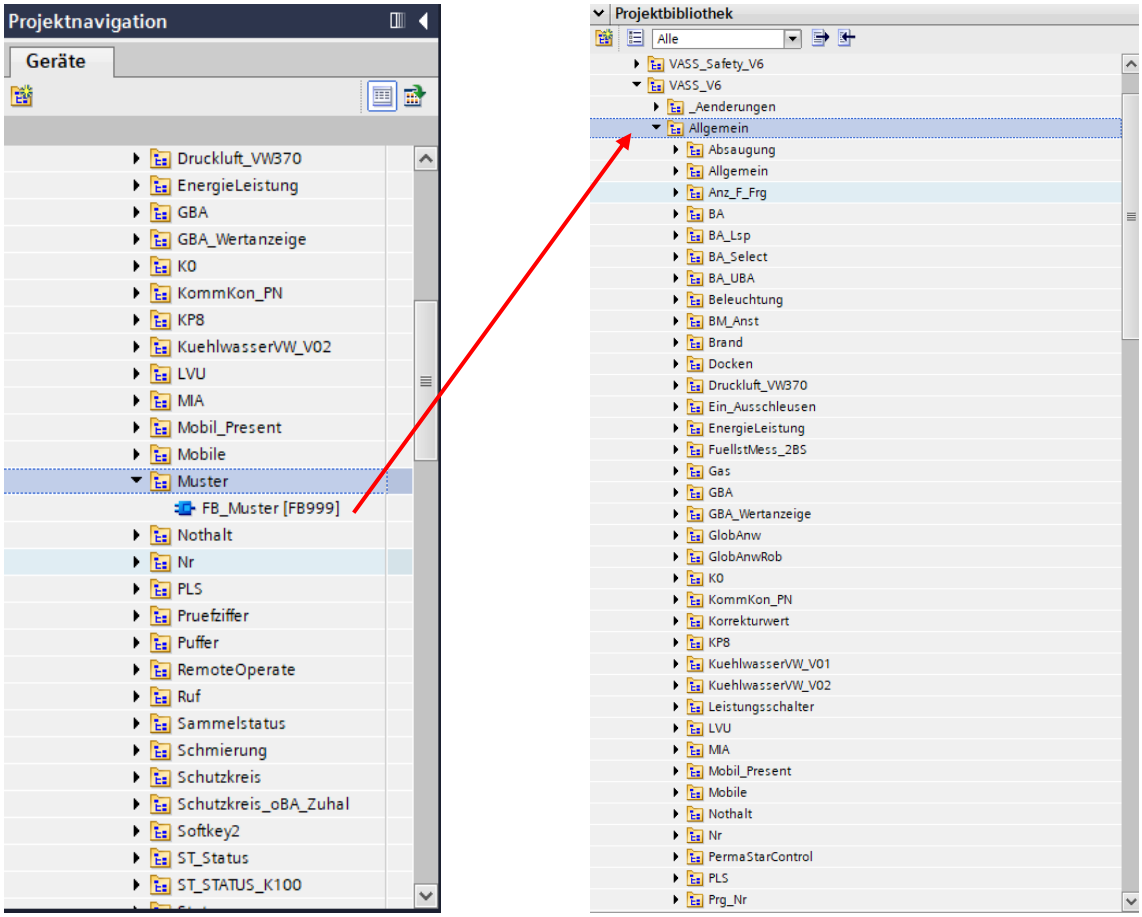


Fig. 5.28 Create library type

Note

Only single module components can be moved (typified) to the project library, not folder structures.

A pop-up window is displayed informing you that the block is being compiled.

If errors occur here, they must be rectified with the help of the Inspector window (Info).

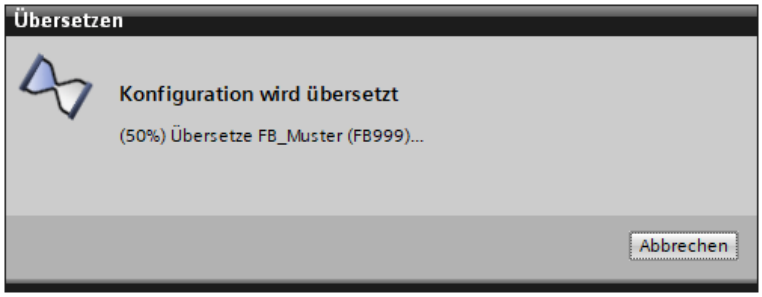


Fig. 5.29 Pop-up window informing you that the block is being compiled

5.3.3.2 "Add type" dialog

The first time the block is typified, the "Add type" dialog is displayed.

The type name (block name with the prefix "FB_") must be entered here.

The version number of the initial creation is "1.1.0" (see figure).

The same person as in the properties of the block should be entered here as the author.

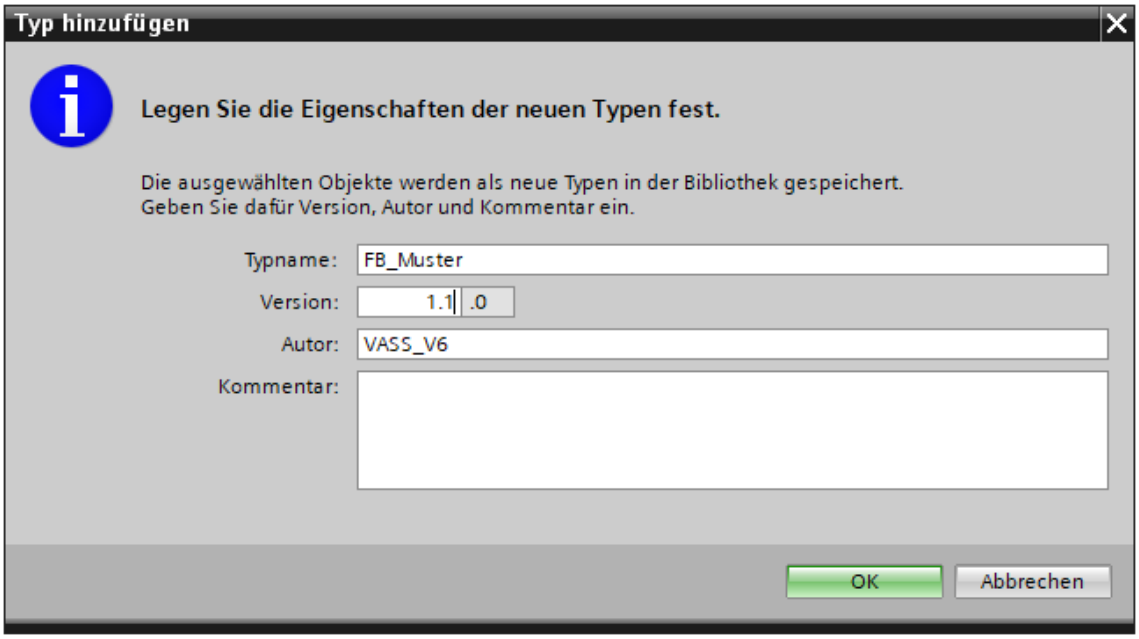


Fig. 5.30 "Add type" dialog

5.3.3.3 Creating folders in the project library

Select an appropriate technology folder within the project library and right-click on "Add folder".

The folder should have the same structure as in the project navigation.

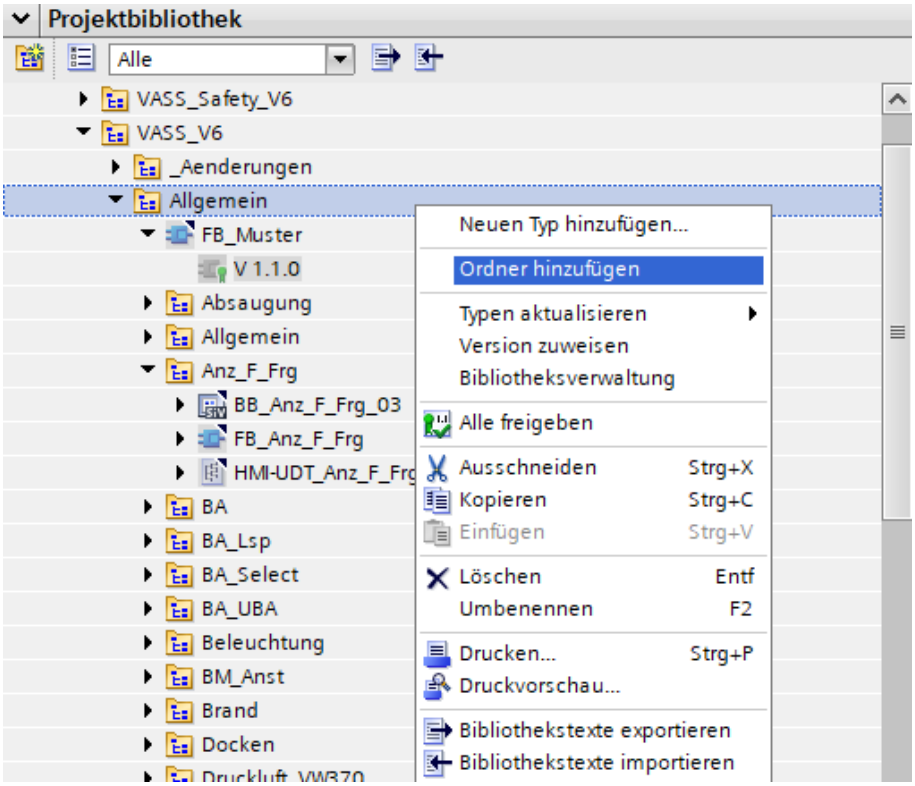


Fig. 5.31 Creating folders in the project library

5.3.3.4 Moving a block in the project library

Move the module component created (FB-Muster) in the project library to the new folder (Muster) using "drag & drop".

The block has now been stored as a type in the project library.

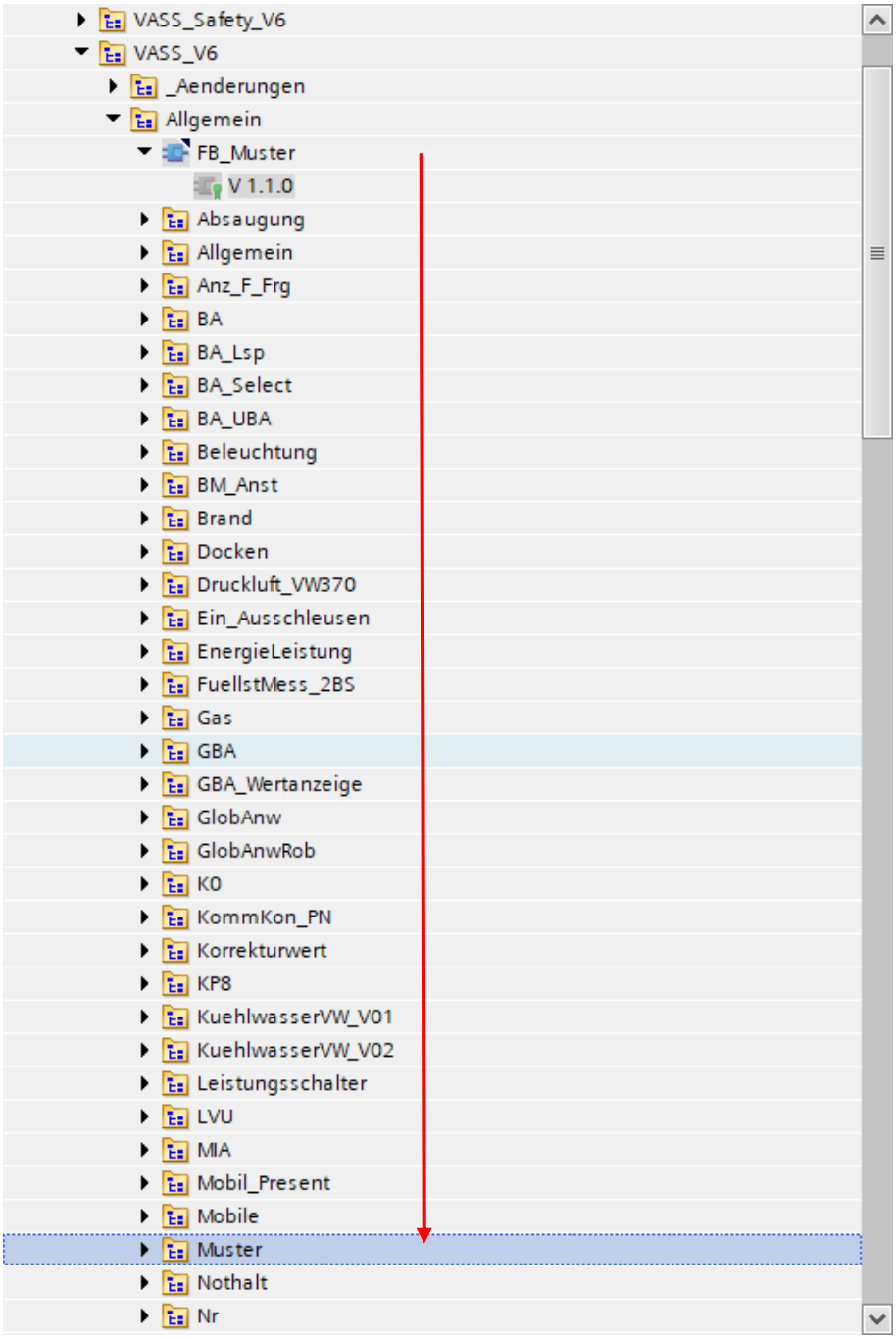


Fig. 5.32 Moving a block in the project library

5.3.4 Release block type for editing

5.3.4.1 Requirements

The block must be available for editing in both the project library and the project navigation.

5.3.4.2 Release for editing

A non-released type with a display is shown in the work area of the TIA Portal:

The editor is write-protected because it is connected to a type in the library.
To carry out changes, you must "edit the type".

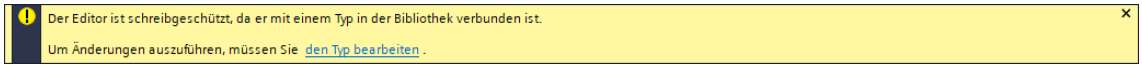


Fig. 5.33 Display "Editor is write-protected"

Here, you must click on "**edit the type**" to change the write-protected block to edit mode.

The info bar should then look as follows:

This object is connected to a type in the library and is currently
in the "In test" state.
Any change to this test instance is mirrored in the version of the type in the test state
: You can "release the version" or "discard the changes and
delete the version".

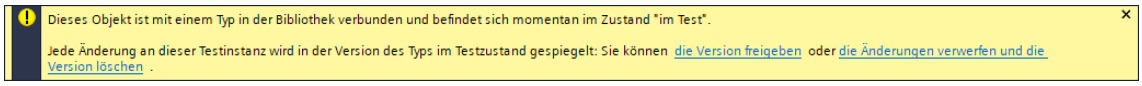


Fig. 5.34 Display "edit the type"

If changes were made, you can accept them by clicking on the text "release the version".

If no changes were made, please click "Discard changes and delete version".

5.3.5 Enable type version dialog

In the dialog, the version number is incremented automatically at the last position. This enables changes in the field to be recognized.

The checkmark "Delete all unused type versions from the library" must be set so that multiple versions of a block do not exist within a project.

Confirm the dialog with "OK".

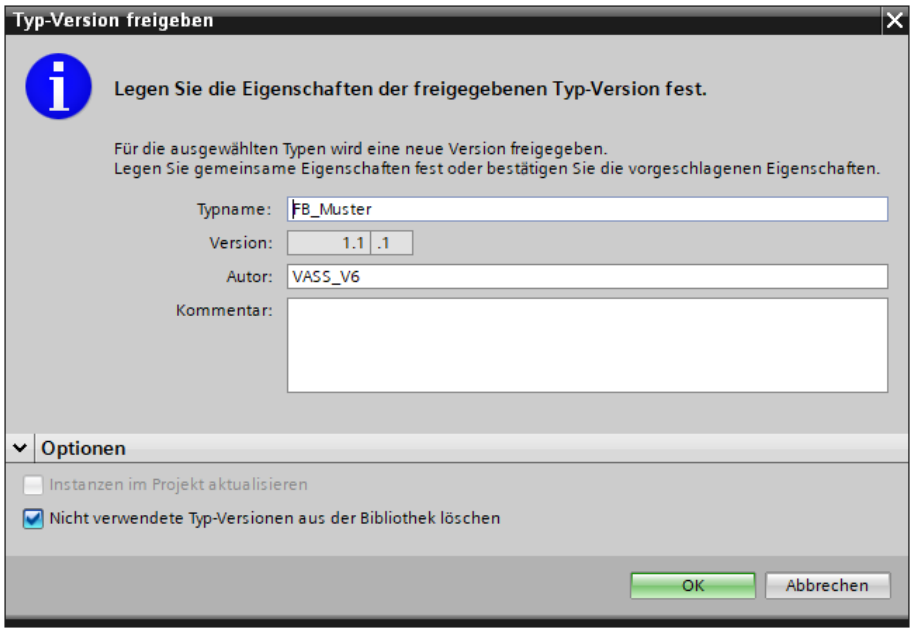


Fig. 5.35 Release dialog type version

5.3.5.1 Tag declaration

① Accessible from HMI/OPC UA

The property must be disabled by default for input parameters, output parameters and input/ouput parameters.

② Writable from HMI/OPC UA

The property must be disabled by default for input parameters, output parameters and input/ouput parameters.

③ Visible in HMI engineering

The property must be disabled by default for input parameters, output parameters and input/ouput parameters.

④ Set value

The set value is not applicable in the VASS standard.

⑤Supervision

Supervision is available for the system-side signaling system ProDiag and are assigned in the static range.

Input parameters

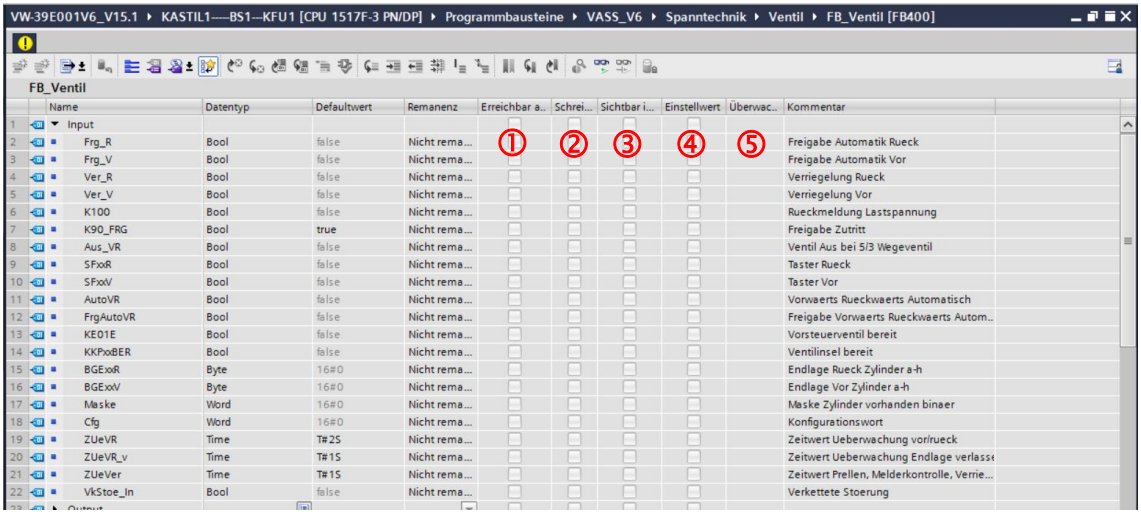
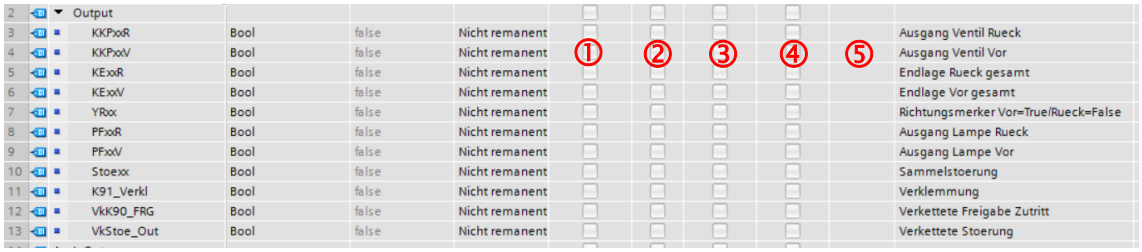


Fig. 5.36 Block: Input parameters

Output parameters



2	Output										
3	KKPxxR	Bool	false	Nicht remanent							Ausgang Ventil Rueck
4	KKPxxV	Bool	false	Nicht remanent	①	②	③	④	⑤		Ausgang Ventil Vor
5	KExxR	Bool	false	Nicht remanent							Endlage Rueck gesamt
6	KExxV	Bool	false	Nicht remanent							Endlage Vor gesamt
7	YRxx	Bool	false	Nicht remanent							Richtungsmerker Vor=True/Rueck=False
8	PFxxR	Bool	false	Nicht remanent							Ausgang Lampe Rueck
9	PFxxV	Bool	false	Nicht remanent							Ausgang Lampe Vor
10	Stoexx	Bool	false	Nicht remanent							Sammelstoerung
11	K91_Verkl	Bool	false	Nicht remanent							Verklemmung
12	VkK90_FRG	Bool	false	Nicht remanent							Verkettete Freigabe Zutritt
13	VkStoe_Out	Bool	false	Nicht remanent							Verkettete Stoerung

Fig. 5.37 Block: Output parameters

Input and output parameters

3	InOut										
4	ST_BA	"ST_Betriebsarten"			①	②	③	④	⑤		Betriebsartenstruktur

Fig. 5.38 Block: Input and output parameters

Temporary tags

Temporary tags are used as buffers for the current block call. If a block with temporary tags is opened, the states of the tags are initially undefined. The tags are only assigned a defined value, but only for this block call by the assignment or the transfer of values. Interconnecting structures (STB_Ventil) are called in the temporary tag area so that the interconnecting structure is bound to the FB.

Static tags

A static tag is a local tag whose value is retained across all block cycles (block memory).

It is used to store values of a function block.

First, the declaration of the HMI-UDT is made, then the ProDiag supervision follows.

FB_Ventil

	Name	Datentyp	Defaultwert	Remanenz	Erreichbar a..	Schrei...	Sichtbar i...	Einstellwert	Überwac..	Kommentar
39	Static									
40	HMI-UDT	"HMI-UDT_Ventil"		Nicht re...	☑	☑	☑	☐		
41	001	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Freigabe
42	002	Bool	false	Nicht rema...	☐	☐	☐	☐		Verriegelungsfehler Rueck
43	003	Bool	false	Nicht rema...	☐	☐	☐	☐		Verriegelungsfehler Vor
44	004	Bool	false	Nicht rema...	☐	☐	☐	☐		Befehlsgeberkontrolle Rueck
45	005	Bool	false	Nicht rema...	☐	☐	☐	☐		Befehlsgeberkontrolle Vor
46	006	Bool	false	Nicht rema...	☐	☐	☐	☐		Endlage Rueck verlassen ohne Anst.
47	007	Bool	false	Nicht rema...	☐	☐	☐	☐		Endlage Vor verlassen ohne Anst.
48	008	Bool	false	Nicht rema...	☐	☐	☐	☐		Endlage Rueck nicht verlassen
49	009	Bool	false	Nicht rema...	☐	☐	☐	☐		Endlage Vor nicht verlassen
50	010	Bool	false	Nicht rema...	☐	☐	☐	☐		Zeitueberwachung Bewegung Rueck
51	011	Bool	false	Nicht rema...	☐	☐	☐	☐		Zeitueberwachung Bewegung Vor
52	012	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Zyl. a
53	013	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Zyl. b
54	014	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Zyl. c
55	015	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Zyl. d
56	016	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Zyl. e
57	017	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Zyl. f
58	018	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Zyl. g
59	019	Bool	false	Nicht rema...	☐	☐	☐	☐		Stoerung Zyl. h
60	020	Bool	false	Nicht rema...	☐	☐	☐	☐		Ventilspule Unterbrechung,Kurzschluss
61	021	Bool	false	Nicht rema...	☐	☐	☐	☐		Warnung Zyl.a
62	022	Bool	false	Nicht rema...	☐	☐	☐	☐		Warnung Zyl.b
63	023	Bool	false	Nicht rema...	☐	☐	☐	☐		Warnung Zyl.c
64	024	Bool	false	Nicht rema...	☐	☐	☐	☐		Warnung Zyl.d
65	025	Bool	false	Nicht rema...	☐	☐	☐	☐		Warnung Zyl.e
66	026	Bool	false	Nicht rema...	☐	☐	☐	☐		Warnung Zyl.f
67	027	Bool	false	Nicht rema...	☐	☐	☐	☐		Warnung Zyl.g
68	028	Bool	false	Nicht rema...	☐	☐	☐	☐		Warnung Zyl.h
69	029	Bool	false	Nicht rema...	☐	☐	☐	☐		Endlage Prellen Rueck
70	030	Bool	false	Nicht rema...	☐	☐	☐	☐		Endlage Prellen Vor
71	255	Bool	false	Nicht rema...	☐	☐	☐	☐		@FB_Ventil@0.0
72	xF_Frg	Bool	false	Remanent	☐	☐	☐	☐		Fehler Freigabe
73	xF_BFG_R	Bool	false	Remanent	☐	☐	☐	☐		Fehler Melderkontrolle Endlagen R
74	xF_BFG_V	Bool	false	Remanent	☐	☐	☐	☐		Fehler Melderkontrolle Endlagen V
75	xF_EndR_voA	Bool	false	Remanent	☐	☐	☐	☐		Fehler Endlage Rueck verlassen ohne Ansteuerung
76	xF_EndV_voA	Bool	false	Remanent	☐	☐	☐	☐		Fehler Endlage Vor verlassen ohne Ansteuerung

Fig. 5.39 Block: Static tags

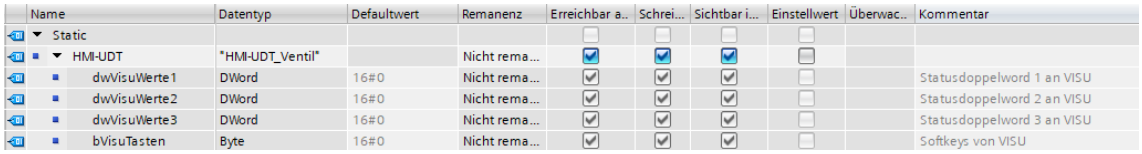
For visualization

Create HMI-UDT with the required tags

The name of the UDT is written such that HMI-UDT is fixed and the FB name follows separated by a "_". Example HMI-UDT_Valve

The visualization interface is defined in the following order:

- VisuValues → status bits / values for visualization
- …..
- VisuTasten → Control bits of the visualization



Name	Datentyp	Defaultwert	Remanenz	Erreichbar a...	Schrei...	Sichtbar i...	Einstellwert	Überwac...	Kommentar
Static									
HMI-UDT	"HMI-UDT_Ventil"		Nicht rema...	☑	☑	☑	☐		
dwVisuWerte1	DWord	16#0	Nicht rema...	☑	☑	☑	☐		Statusdoppelword 1 an VISU
dwVisuWerte2	DWord	16#0	Nicht rema...	☑	☑	☑	☐		Statusdoppelword 2 an VISU
dwVisuWerte3	DWord	16#0	Nicht rema...	☑	☑	☑	☐		Statusdoppelword 3 an VISU
bVisuTasten	Byte	16#0	Nicht rema...	☑	☑	☑	☐		Softkeys von VISU

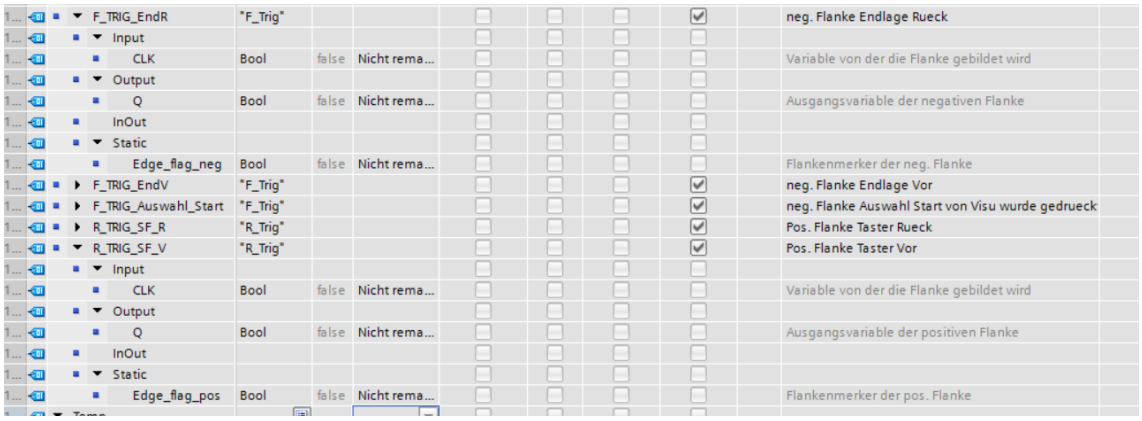
Fig. 5.40 Block: Static tags Visu 1

The SiVArc only recognizes the string "HMI-UDT". No distinction is made between upper and lower case.

Note

There may be only one block per structure name as the SiVArc would only recognize the first block.

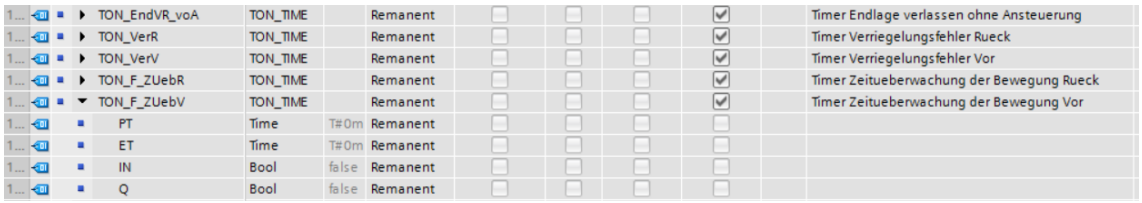
For edge formation



F_TRIG_EndR	"F_Trig"							☑		neg. Flanke Endlage Rueck
Input										
CLK	Bool	false	Nicht rema...							Variable von der die Flanke gebildet wird
Output										
Q	Bool	false	Nicht rema...							Ausgangsvariable der negativen Flanke
InOut										
Static										
Edge_flag_neg	Bool	false	Nicht rema...							Flankenmerker der neg. Flanke
F_TRIG_EndV	"F_Trig"							☑		neg. Flanke Endlage Vor
F_TRIG_Auswahl_Start	"F_Trig"							☑		neg. Flanke Auswahl Start von Visu wurde gedrueck
R_TRIG_SF_R	"R_Trig"							☑		Pos. Flanke Taster Rueck
R_TRIG_SF_V	"R_Trig"							☑		Pos. Flanke Taster Vor
Input										
CLK	Bool	false	Nicht rema...							Variable von der die Flanke gebildet wird
Output										
Q	Bool	false	Nicht rema...							Ausgangsvariable der positiven Flanke
InOut										
Static										
Edge_flag_pos	Bool	false	Nicht rema...							Flankenmerker der pos. Flanke

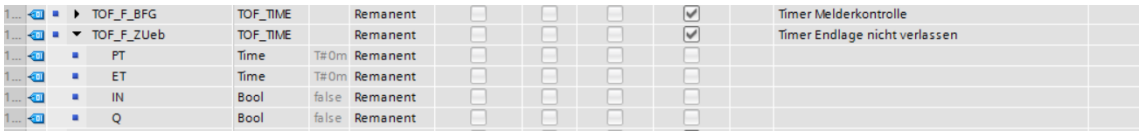
Fig. 5.41 Block: Static tags edge formation

For timer



TON_EndVR_voA	TON_TIME		Remanent				☑		Timer Endlage verlassen ohne Ansteuerung
TON_VerR	TON_TIME		Remanent				☑		Timer Verriegelungsfehler Rueck
TON_VerV	TON_TIME		Remanent				☑		Timer Verriegelungsfehler Vor
TON_F_ZUebR	TON_TIME		Remanent				☑		Timer Zeitueberwachung der Bewegung Rueck
TON_F_ZUebV	TON_TIME		Remanent				☑		Timer Zeitueberwachung der Bewegung Vor
PT	Time	T#0m	Remanent						
ET	Time	T#0m	Remanent						
IN	Bool	false	Remanent						
Q	Bool	false	Remanent						

Fig. 5.42 Block: Static tags ON delay



TOF_F_BFG	TOF_TIME		Remanent				☑		Timer Melderkontrolle
TOF_F_ZUeb	TOF_TIME		Remanent				☑		Timer Endlage nicht verlassen
PT	Time	T#0m	Remanent						
ET	Time	T#0m	Remanent						
IN	Bool	false	Remanent						
Q	Bool	false	Remanent						

Fig. 5.43 Block: Static tags OFF delay

Other static tags

FB_Ventil

		Name	Datentyp	Def...	Remanenz	Erreichbar a..	Einstellwert	Übe...	Kommentar
29		255	Bool	false	Nicht re...				@FB_Ventil@0.0
30		bF_BFG	Byte	16#0	Remanent				Fehlerbyte Melderkontrolle
31		xF_Frg	Bool	false	Remanent				Fehler Freigabe
32		xF_BFG_R	Bool	false	Remanent				Fehler Melderkontrolle Endlagen R
33		xF_BFG_V	Bool	false	Remanent				Fehler Melderkontrolle Endlagen V
34		xF_EndR_voA	Bool	false	Remanent				Fehler Endlage Rueck verlassen ohne Ansteuerung
35		xF_EndV_voA	Bool	false	Remanent				Fehler Endlage Vor verlassen ohne Ansteuerung
36		xF_VerR	Bool	false	Remanent				Verriegelungsfehler Rueck
37		xF_VerV	Bool	false	Remanent				Verriegelungsfehler Vor
38		xF_EndR	Bool	false	Remanent				Fehler Endlage Rueck nicht verlassen
39		xF_EndV	Bool	false	Remanent				Fehler Endlage Vor nicht verlassen
40		xF_ZuebR	Bool	false	Remanent				Fehler Zeitueberwachung Bewegung Rueck
41		bF_EndR	Byte	16#0	Remanent				Stoerbyte Endlage vor nicht verlassen
42		bF_EndV	Byte	16#0	Remanent				Stoerbyte Endlage Rueck nicht verlassen
43		bF_EndR_voA	Byte	16#0	Remanent				Stoerbyte Endlage Vor nicht verlassen
44		bF_EndV_voA	Byte	16#0	Remanent				Stoerbyte Endlage Vor nicht verlassen
45		bF_Stoe	Byte	16#0	Remanent				Fehlerbyte Sammelstoerung
46		bF_ZUebR	Byte	16#0	Remanent				Stoerbyte Zeitueberwachung Bewegung Rueck
47		bF_ZUebV	Byte	16#0	Remanent				Stoerbyte Zeitueberwachung Bewegung Vor
48		xF_ZuebV	Bool	false	Remanent				Fehler Zeitueberwachung Bewegung Vor
49		xHM_EndR	Bool	false	Remanent				Hilfsmerker Endlage rueck
50		xHM_EndV	Bool	false	Remanent				Hilfsmerker Endlage vor
51		xF_KKPxxBER	Bool	false	Remanent				Fehler Ventilspule
52		xEndR	Bool	false	Remanent				Sammelmeldung Endlage rueck
53		xEndV	Bool	false	Remanent				Sammelmeldung Endlage vor
54		xSF_R	Bool	false	Nicht rema...				Taster Vorwaerts von Visu
55		xSF_V	Bool	false	Nicht rema...				Taster rueckwaerts von VISU

Fig. 5.44 Block: Other static tags

5.3.5.2 Introduction to SCL

5.3.5.2.1 SCL - Structured Control Language

SCL is a structured programming language.

Unlike statement lists (STL), ladder diagrams (LAD) and function block diagrams (FBD), it involves "high-level programming".

5.3.5.2.2 Block structure

The block structure comprises the following components and is subdivided by regions:

- input parameters — Input
- Output parameters — Output
- Input/output parameters — InOut
- Static tags — Static
- Temporary tags — Temp
- Statement section

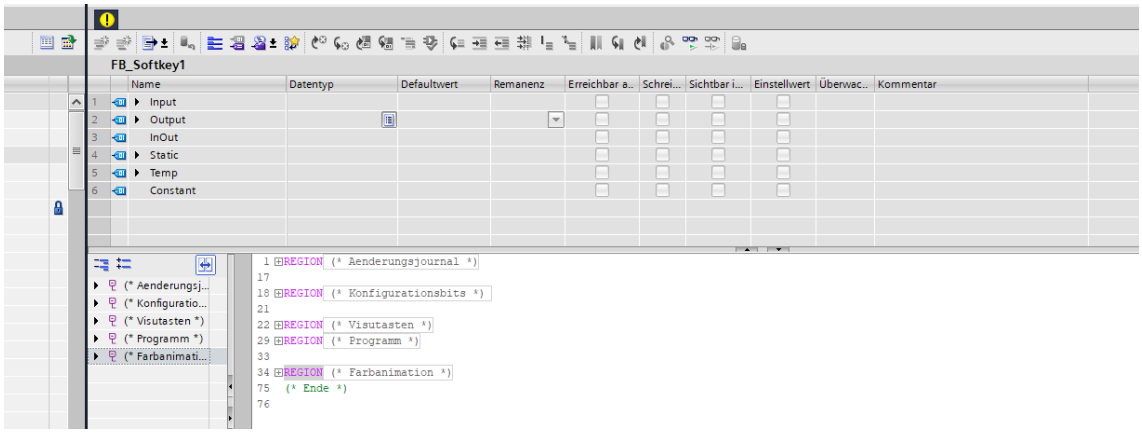


Fig. 5.45 FB_Softkey1

5.3.5.2.3 Identifier in SCL

Identifiers are, for example, names of constants, tags and FBs, with a maximum length of 24 characters. Umlauts or reserved characters and words are not allowed.

The first character must be a letter or underscore. No distinction is made between upper and lower case text.

The identifier indicating the data type must be placed in front of the tag name.

x → Bool
b → Byte
w → Word
dw → Double word

i → Integer
di → Double integer
r → Real

p → Pointer

ar → Arrays
st → Structure
t → Time

Examples:

dwVisuWerte1	: DWORD ;	//status double word 1 at VISU
bVisutasten	: BYTE ;	//softkey of VISU
xF_Ver	: BOOL;	//Lock error

5.3.5.2.4 Display of numbers

Numbers are displayed as follows:

TYPE#SYSTEM#NUMBER

TYPE: Byte, Word, Dword

SYSTEM: Binary, dec, hex

NUMBER: 0~9, A~F

Example:

15	Binary	**Byte#2#0000 1111**
	Decimal	**#15**
	Hex	**Byte#16#0F**

5.3.5.2.5 Operators/operands/expressions

There must always be a space between operators and operands.

Example:

```
IF #iSollWert < #iIstWert THEN

    #iSollWert := 5 + #iIstWert;

END_IF;
```

Fig. 5.46 Blank spaces between operators and operands

5.3.5.2.6 Comment

- Line comment " // " is used for:
  - Input and output parameters
  - Declaring variables
  - Formation of static tags

- Comment block " (* *) "or " (/* */) " is used for:
  - Detailed explanation in the program
  - Commenting out large program sections

Note

The syntax " (/* */) " indicates multilingual comment blocks and should be used with version TIA V16 and higher so that you can also use the option to translate comments in SCL.

5.3.5.2.7 Slice function

Addressing areas of a tag with slice access

Description:

You have the option to specifically address areas within declared tags. You can access areas of the 1-bit, 8-bit, 16-bit, or 32-bit widths. The division of a memory area (e.g. BYTE or WORD) into a smaller memory area (e.g. BOOL) is also referred to as a "slice".

Structures, constants and tags overlaying AT cannot be addressed with slice access.

Syntax:

The following syntax is used for addressing:

<Tag>.X<Bit number>
<Tag>.B<BYTE number>
<Tag>.W<WORD number>
<Tag>.D<DWORD number>

The syntax has the following components:

Part	Description
<Variable>	Tag that you access. The tag must be of the "Bit string" or "Integer" data type. With SCL, you can only program slice access to tags of the "Integer" data type when the IEC check is disabled.
X	ID for the access width "Bit (1Bit)"
B	ID for the access width "Byte (8 Bit)"
W	ID for the access width "Word (16 Bit)"
D	ID for access width "DWord (32 bit)"
<BIT number>	Bit number within <tag> that is accessed. Number 0 accesses the least significant BIT.
<BYTE number>	Byte number within <tag> that is accessed.
<WORD number>	Word number within <tag> that is accessed. The number 0 accesses the least significant WORD.
<DWORD number>	DWord number within <tag> that is accessed. The number 0 accesses the least significant DWORD.

Tab. 5.1 Syntax of slice function

Example of slice access:

The following examples show the addressing of slices in bit, byte, word and double word modes:

Addressing	Explanation
"Engine".Motor.X0 … "Engine".Motor.X7	"Motor" is a tag of the BYTE, WORD, DWORD or LWORD data type in the global data block "Engine". X0 addresses the bit address 0; X7 the bit address 7 within "Motor".
"Engine".Speed.B0 … "Engine".Speed.B1	"Speed" is a tag of the WORD, DWORD or LWORD data type in the global data block "Engine". B0 addresses the byte address 0; B1 the byte address 1 within "Speed".
"Engine".Fuel.W0 … "Engine".Fuel.W1	"FUEL" is a tag of the DWORD or LWORD data type in the global data block "Engine". W0 addresses the word address 0, W1 the word address 1 within "Fuel".
"Engine".Data.D0 … "Engine".Data.D1	"Data" is a tag of the LWORD data type in the global data block "Engine". D0 addresses the double word address 0; D1 the double word address 1 within "Data".

Tab. 5.2 Example of slice access

Note

You can find a detailed example in the Siemens Industry Online Support:
https://support.industry.siemens.com/cs/ww/en/view/57374718

5.3.5.2.8 Input and output parameters

Declaration In/Out / "In_Out parameters"

If the block is to be assigned a parameter for configuration or the enable is to be provided with a bit mask, the block should be declared as a data type WORD. This enables the binary form to be used during the call to supply the parameter. (2#0000_0000_1111_1111).

Typical examples of this are the parameters "_Cfg" and "Mask" at the "FB_Ventil" (FB400).

Input in binary format:

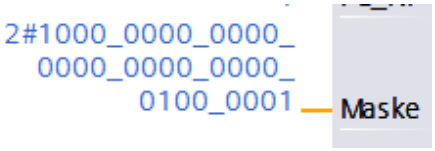


Fig. 5.47 Input in binary format

5.3.5.3 Structure of a module FB in SCL

The module FBs are stored in the VASS library.

Note

Generally, changes or new standard module FBs are permitted only after consultation and with the approval of the electrical engineering department in charge.

Module FBs are blocks for controlling function modules for actuators and units or other FBs that execute a certain function.

Blocks that are not available in the VASS library must be created by the system supplier in consultation with the standardizing electrical engineering department in charge.

The module components are usually created in the programming language S7-SCL.

5.3.5.3.1 Structure of the FB

The subdivision into the following regions must be adhered to in the block:

- Change journal
- Explanation of the configuration bits
- Checking buttons of the visualization
- Program section with control of the outputs
- Evaluation of messages
- Assignment of supervisions for ProDiag
- Preparation of the symbol tags for visualization
- Animation navigation

Note

Changes and deviations from the block structure must be coordinated with the standardizing electrical engineering department.

The figures in this section are from the module FB:

FB 400: FB_Ventil Module FB valve

If functions, which are to be shown as an example, are not contained in FB 400, figures from other blocks are used.

5.3.5.3.2 Change journal

In the VASS standard, the change journal of a module FB comprises the following:

- Copyright information
- Change information

```
REGION (* Aenderungsjournal *)
(*
*****************************************************************************************
**  Copyright  2016             SIEMENS AG D-90475 Nuernberg                            **
**  All Rights Reserved         VOLKSWAGEN AG, 38436 Wolfsburg                          **
**                              AUDI AG, D-85045 Ingolstadt                             **
*****************************************************************************************
    Datum       Version     Autor       Beschreibung
-----------------------------------------------------------------------------------------
    13.03.19    1.4.00      VASS_V6     Meldetexte angepasst
    12.04.18    1.1.00      VASS_V6     TIA Startversion

*****************************************************************************************
*)
END_REGION
```

Fig. 5.48 Change journal

When a block is created or changed, in addition to the date, version and editor, the description of the change and a comment on incorporation in the standard should be entered in the change information.

The change to the VISU interface should be marked separately, by entering a "V" at the end of the line.

Please note that the most recent entry must always be at the beginning of the list, i.e. at the top.

5.3.5.3.3 Explanation of the configuration bits

```
REGION (* Konfigurationsbits *)
(*
BIT     Kommentar
X0      0= 5/2-Wegeventil bistabil ,1= 5/2-Wegeventil unistabil bzw. 5/3-Wegeventil
X1      1= Befehlsgeberkontrolle ausblenden
X2      1= Bewegungszeitueberwachung ausblenden
X3      Reserve
X4      Reserve
X5      Reserve
X6      1= ohne Kontrolle "NOT AutoVR" (z.B.Elektrodenzylinder)
X7      1= Grundstellung vor
X8      1= ohne Einschaltventil KE01E
X9      Reserve
X10     1= Prellen ausblenden
X11     1= mit Verzoegerung Befehlsgeberkontrolle
X12     Reserve
X13     Reserve
X14     Reserve
X15     Reserve
*)
END_REGION
```

Fig. 5.49 Explanation of the configuration bits

5.3.5.3.4 Checking buttons of the visualization

```
(* Tasten von Visu pruefen *)
IF "DB_ARG".PC_AKTIV = DWORD#16#00000000 THEN
    #"HMI-UDT".bVisuTasten := Byte#0;
END_IF;

#xSF_R := (#"HMI-UDT".bVisuTasten = Byte#1) OR #SFxxR AND NOT #SFxxV;
#xSF_V := (#"HMI-UDT".bVisuTasten = Byte#2) OR #SFxxV AND NOT #SFxxR;
```

Fig. 5.50 Block: Check VISU buttons

Value range bVisuTasten:	1 = Backward,	2 = Forward
Optional hardware button:	SFxxR	SFxxV

PC_ACTIVE: Data DWORD, in this case a bit is set to TRUE for each PC as soon as an operator control is pressed. This serves as a plausibility check of the PC (2nd channel) and for locking the PC in the event of special operation, e.g. "E2".

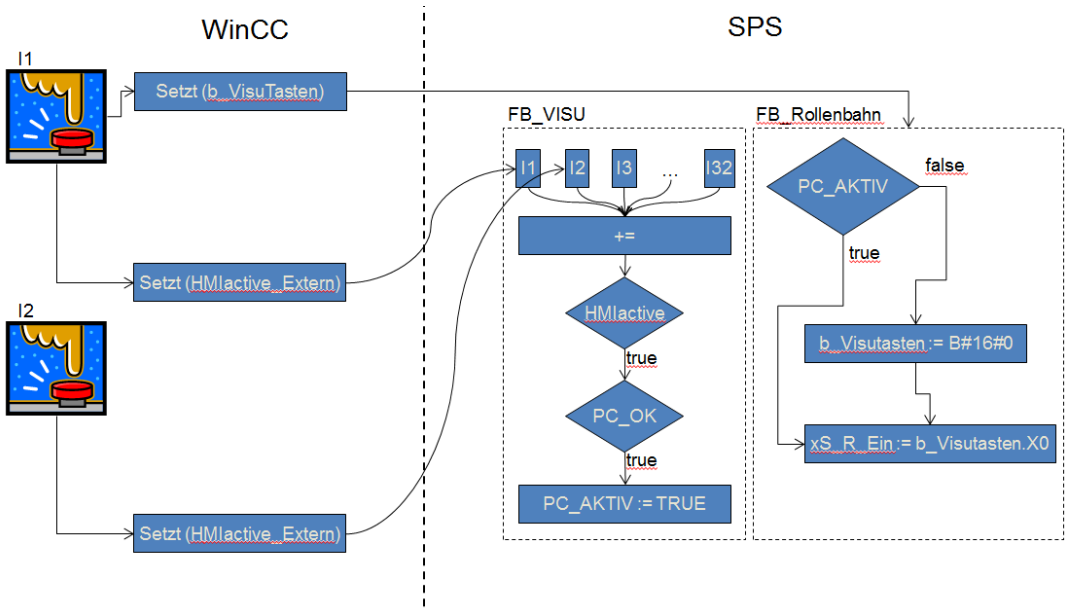


Fig. 5.51 PC_ACTIVE

5.3.5.3.5 Program section with control of the outputs

Description

The program consists of a brief description of the block function. The full block description can be seen in the help function of the block.

```
(*Dieser Baustein realisiert die Ansteuerung und Überwachung für ein Ventil mit-----*)
(*8 Zylindern*)
```

Fig. 5.52 Block: Program description

Programming commands for program control

```
REGION (/* program */)
    #KKPxxR := #xSF_R;
    #KKPxxV := #xSF_V;

    //Counting loop
    FOR #i := 0 TO 100 DO
        #arrVariable[#i] := FALSE; // Statement section FOR
    END_FOR;

    //counting loop with step width
    FOR #i := 0 TO 100 BY 2 DO
        #arrVariable[#i] := FALSE; // Statement section FOR
    END_FOR;

    //Create multiway branch
    CASE #j OF
        1:              // Statement section case 1
            ;
        2..4:           // Statement section case 2 to 4
            ;
        ELSE               // Statement section ELSE
            ;
    END_CASE;

    // execute condition
    IF #Cfg.%X0 THEN      // Condition
        #xStatement := #SFxxR;     // Statement section IF
    END_IF;

    // Branch condition
    IF #Cfg.%X1 THEN      // Condition
        #xStatement := TRUE;    // Statement section IF
    ELSE
        #xStatement := FALSE;      // Statement section ELSE
    END_IF;

    // Branch condition multiple times
    IF #Cfg.%X2 THEN      // Condition
        #xStatement := TRUE;    // Statement section IF
    ELSIF #Cfg.%X3 THEN         // Condition
        #xStatement := FALSE;      // Statement section ELSIF
    ELSE            // Statement section ELSE
        #xStatement := FALSE;
    END_IF;

    // Recheck loop condition
    // If i<5, the line " #arrVariable[#i] := TRUE;"
    // is not executed as CONTINUE
```

```
    //skips these lines and the program loop is executed again.
    // If i>5, CONTINUE is not executed and the line "
    // #arrVariable[#i] := TRUE;"
    // is processed
    FOR #i := 1 TO 15 BY 2 DO
        IF (#i < 5) THEN
            CONTINUE;
        END_IF;
        #arrVariable[#i] := TRUE;
    END_FOR;

    // Exit loop immediately'
    // If I>5, the loop is exited with EXIT
    FOR #i := 1 TO 15 BY 2 DO
        IF (#i > 5) THEN
            EXIT;
        END_IF;
        #arrVariable[#i] := TRUE;
    END_FOR;

END_REGION

WHILE .. DO .. is not allowed.
REPEAT .. UNTIL .. is not allowed.
GOTO is not allowed.
RETURN is not allowed.
```

For edge evaluation

Positive edge:

```
#R_TRIG_SF_R(CLK := #xSF_R);
#R_TRIG_SF_V(CLK := #xSF_V);

IF "DB_ARG".S_AUSWAHL THEN
    IF #R_TRIG_SF_R.Q THEN
        #xSF_R_Ein := NOT #xSF_R_Ein;
        #xSF_V_Ein := False;
        #xSF_R := False;
    ELSIF #R_TRIG_SF_V.Q THEN
        #xSF_V_Ein := NOT #xSF_V_Ein;
        #xSF_R_Ein := False;
        #xSF_V := False;
```

Fig. 5.53 # edge evaluation positive edge

Negative edge:

```
#F_TRIG_Auswahl_Start(CLK := "DB_ARG".S_AUSWAHL_START);

IF #F_TRIG_Auswahl_Start.Q OR NOT "DB_ARG".S_AUSWAHL THEN
    #xSF_R_Ein := False;
    #xSF_V_Ein := False;
```

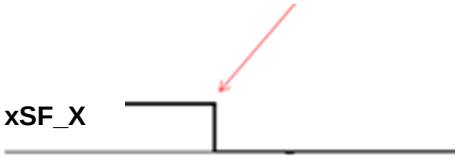


Fig. 5.54 # edge evaluation negative edge

For the timers

On delay:

```
TON_Zueb_X(IN := StartSignal, PT := Zeitangabe);
...
TON_ZUeb_X.Q == TRUE
```

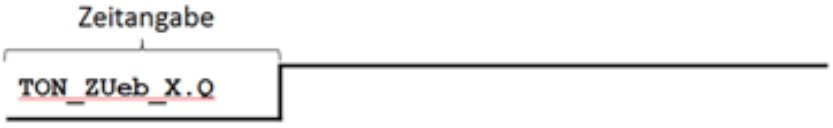


Fig. 5.55 # ON delay (1)

```
(* Einschaltverzoegerung Endlage verlassen ohne Ansteuerung *)
        #TON_EndVR_voA(IN := (#xKExxR AND #xYR_R) OR (#xKExxV AND #xYR_V),
                       PT := #ZUeVer);
```

Fig. 556 ON delay (2)

OFF delay:

```
(*-------------------------------------------------------------------------------*)
(* Zeitueberwachung Endlagen nicht verlassen *)
        #TOF_F_ZUeb(IN := ((#xYR_R AND (#BGExxV = BYTE#0)) OR (#xYR_R AND #xHM_EndR))
                    OR ((#xYR_V AND (#BGExxR = BYTE#0)) OR (#xYR_V AND #xHM_EndV)),
                    PT := #ZUeVR_v);
```

Fig. 5.57 Switch off delay

Evaluation of input parameters

```
(* Freigabe Einschaltventil *)
    #xFrgKE01E := #KE01E OR #Cfg.%X8;
```

Fig. 5.58 Block: Evaluation of input parameters

Error evaluation

```
(* Stoerung Befehlsgeberkontrolle *)
        #TOF_F_BFG(IN := (#BGExxR & #BGExxV & #Maske) = BYTE#0,
                   PT := #ZUeVer);
        IF NOT #TOF_F_BFG.Q AND #xYR_V AND #ST_BA.EN_Stoe AND NOT #Cfg.%X1 THEN
            #xF_BFG_R := TRUE;
            #bF_BFG := #BGExxR & #Maske.%B0;
        END_IF;
        IF NOT #TOF_F_BFG.Q AND #xYR_R AND #ST_BA.EN_Stoe AND NOT #Cfg.%X1 THEN
            #xF_BFG_V := TRUE;
            #bF_BFG := #BGExxV & #Maske.%B0;
        END_IF;
```

Fig. 5.59 Block: Error evaluation

```
(*-------------------------------------------------------------------------------*)
(* Sammelstoerung *)
        #xStoexx := #xF_Frg
        OR #xF_BFG_R
        OR #xF_BFG_V
        OR #xF_EndR_voA
        OR #xF_EndV_voA
        OR #xF_VerR
        OR #xF_VerV
        OR #xF_EndR
        OR #xF_EndV
        OR #xF_ZuebR
        OR #xF_ZuebV
        OR #xF_KKPxxBER;
        #Stoexx := #xStoexx;
```

Fig. 5.60 Block: Forming a general fault

```
(*-------------------------------------------------------------------------------*)
(* Ruecksetzen Stoerung Bewegungszeitueberwachung *)
        IF #ST_BA.Quit OR NOT #ST_BA.EN_Stoe OR #Cfg.%X2 THEN
            #xF_ZuebR := FALSE;
            #bF_ZUebR := Byte#0;
            #xF_ZuebV := FALSE;
            #bF_ZUebV := Byte#0;
        END_IF;
```

Fig. 5.61 Block: Reset fault

The signal "Acknowledge" from the Visu is formed in the structure "BA".

`ST_BA.Quit`

Interconnection of the output parameters

```
(*-------------------------------------------------------------------------------*)
(* Ausgaenge Schreiben *)
    #xKKPxxR := (#xHM_KKPxxR OR (#xKExxR AND NOT #Aus_VR AND #Cfg.%X0)) AND NOT #xHM_KKPxxV;
    #xKKPxxV := (#xHM_KKPxxV OR (#xKExxV AND NOT #Aus_VR AND #Cfg.%X0)) AND NOT #xHM_KKPxxR;
    #KKPxxR := #xKKPxxR;
    #KKPxxV := #xKKPxxV;

    #xPFxxR_DL := #xHM_EndR;
    #xPFxxR_BL := #xF_Frg OR #xF_EndR_voA OR #xF_VerR;
    #xPFxxR_BS := #xF_BFG_R OR #xF_EndR OR #xF_ZuebR;
(* Ausgang Lampe Rueck *)
    #PFxxR := ("DB_ARG".BLINK_S AND #xPFxxR_BS)
    OR ("DB_ARG".BLINK_L AND (#xPFxxR_BL AND NOT #xPFxxR_BS))
    OR (#xPFxxR_DL AND NOT #xPFxxR_BL AND NOT #xPFxxR_BS)
    OR "DB_ARG".LTEST;

    #xPFxxV_DL := #xHM_EndV;
    #xPFxxV_BL := #xF_Frg OR #xF_EndV_voA OR #xF_VerV;
    #xPFxxV_BS := #xF_BFG_V OR #xF_EndV OR #xF_ZuebV;
(* Ausgang Lampe Vor *)
    #PFxxV := ("DB_ARG".BLINK_S AND #xPFxxV_BS)
    OR ("DB_ARG".BLINK_L AND (#xPFxxV_BL AND NOT #xPFxxV_BS))
    OR (#xPFxxV_DL AND NOT #xPFxxV_BL AND NOT #xPFxxV_BS)
    OR "DB_ARG".LTEST;
```

Fig. 5.62 Block: Interconnecting output parameters

5.3.5.3.6 Evaluation of messages

```
REGION (* Ueberwachungen *)
    #"001" := #xF_Frg;              // 'STE/Stoerung Freigabe'
    #"002" := #xF_VerR;             // 'STE/Verriegelungsfehler Rueck'
    #"003" := #xF_VerV;             // 'STE/Verriegelungsfehler Vor'
    #"004" := #xF_BFG_R;            // 'STE/Befehlsgeberkontrolle Rueck'
    #"005" := #xF_BFG_V;            // 'STE/Befehlsgeberkontrolle Vor'
    #"006" := #xF_EndR_voA;         // 'STE/Endlage Rueck verlassen ohne Anst.'
    #"007" := #xF_EndV_voA;         // 'STE/Endlage Vor verlassen ohne Anst.'
    #"008" := #xF_EndR;             // 'STE/Endlage Rueck nicht verlassen'
    #"009" := #xF_EndV;             // 'STE/Endlage Vor nicht verlassen'
    #"010" := #xF_ZuebR;            // 'STE/Bewegungszeitueberwachung Rueck'
    #"011" := #xF_ZuebV;            // 'STE/Bewegungszeitueberwachung Vor'
    #"012" := #wF_StoeWarn.%X0;     // 'STE/Stoerung Zyl. a'
    #"013" := #wF_StoeWarn.%X1;     // 'STE/Stoerung Zyl. b'
    #"014" := #wF_StoeWarn.%X2;     // 'STE/Stoerung Zyl. c'
    #"015" := #wF_StoeWarn.%X3;     // 'STE/Stoerung Zyl. d'
    #"016" := #wF_StoeWarn.%X4;     // 'STE/Stoerung Zyl. e'
    #"017" := #wF_StoeWarn.%X5;     // 'STE/Stoerung Zyl. f'
    #"018" := #wF_StoeWarn.%X6;     // 'STE/Stoerung Zyl. g'
    #"019" := #wF_StoeWarn.%X7;     // 'STE/Stoerung Zyl. h'
    #"020" := #xF_KKPxxBER;         // 'STE/Fehler Ventilspule'
    #"021" := #wF_StoeWarn.%X8;     // 'MTE/Warnung Zyl.a'
    #"022" := #wF_StoeWarn.%X9;     // 'MTE/Warnung Zyl.b'
    #"023" := #wF_StoeWarn.%X10;    // 'MTE/Warnung Zyl.c'
    #"024" := #wF_StoeWarn.%X11;    // 'MTE/Warnung Zyl.d'
    #"025" := #wF_StoeWarn.%X12;    // 'MTE/Warnung Zyl.e'
    #"026" := #wF_StoeWarn.%X13;    // 'MTE/Warnung Zyl.f'
    #"027" := #wF_StoeWarn.%X14;    // 'MTE/Warnung Zyl.g'
    #"028" := #wF_StoeWarn.%X15;    // 'MTE/Warnung Zyl.h'
    #"029" := #xF_PreR;             // 'MTE/Endlage Prellen Rueck'
    #"030" := #xF_PreV;             // 'MTE/Endlage Prellen Vor'
END_REGION
```

Fig. 5.63 Evaluation of messages

5.3.5.3.7 Assignment of supervisions for ProDiag

In the VASS standard, the system-side signaling system ProDiag is used for messages for onsite visualization and for the ZAÜ. "Operand" supervision is thus used within function blocks.

In the "static" part of the block header, the messages must be generated via the number sequence 001 - 254 under any existing HMI-UDT.

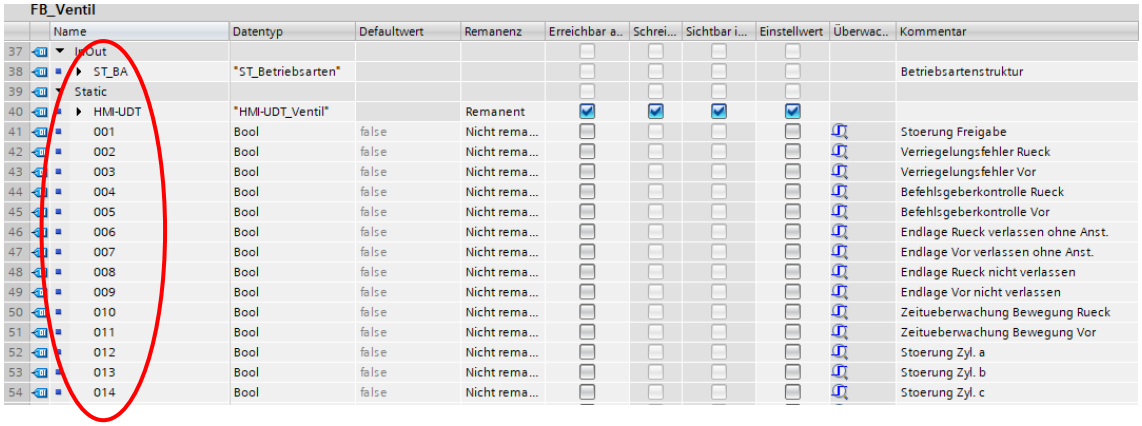


Fig. 5.64 Generation of messages in the "Static" part of the block header

"255" is an ID for the ZAÜ system. "@FB_block_name@0.0" must be entered (multilingual) in the comment here.

The structure of the comment must be observed as otherwise the ZAÜ cannot assign it.

Example "Valve block":

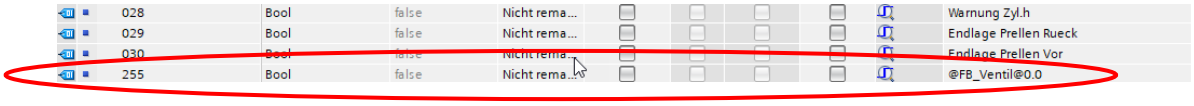


Fig. 5.65 ID for the ZAÜ system

Note

The comment for messages 001 - 254 must be translated and entered in all the languages defined for the VASS standard.

The chapter "Multilingualism" in the configuration guidelines should be consulted.

To create a new supervision, you must open the shortcut menu with a right click in the "Supervision" column of the message in question and insert "Add new supervision" with a left click.

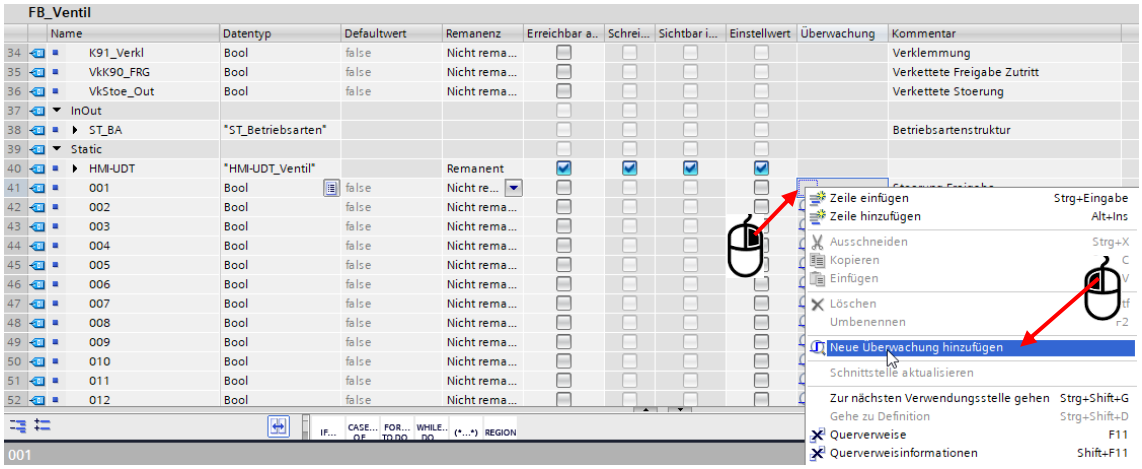


Fig. 5.66 Creating an "operand" supervision

You can make the following settings under the "Supervisions" tab in the Inspector window:

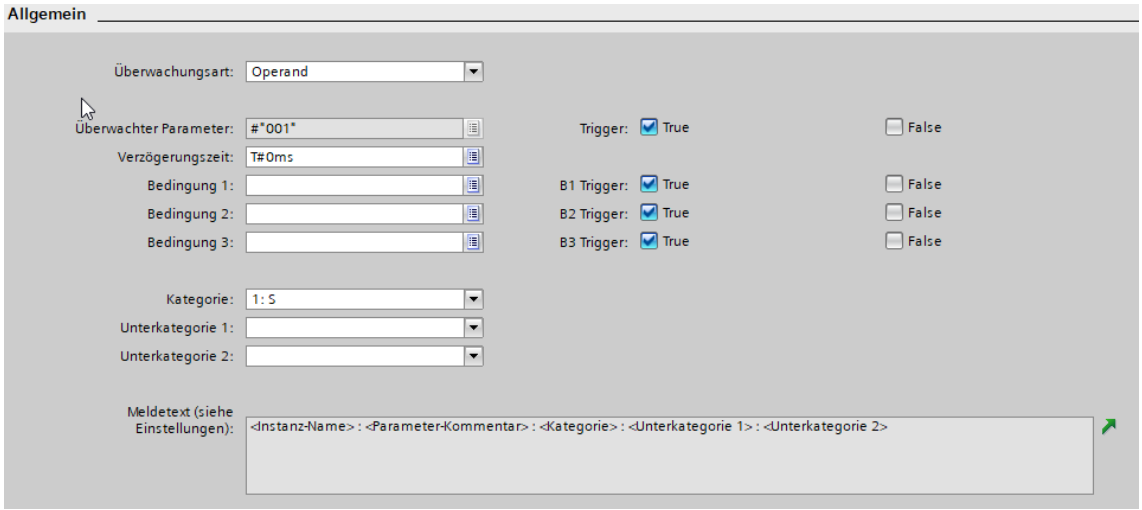


Fig. 5.67 Settings under the "Supervisions" tab in the Inspector window

Note

In the VASS standard, "delay time" and "condition 1 - 3" are not used.

The trigger settings should be left as "True".

The category and the subcategory must be defined for a message.

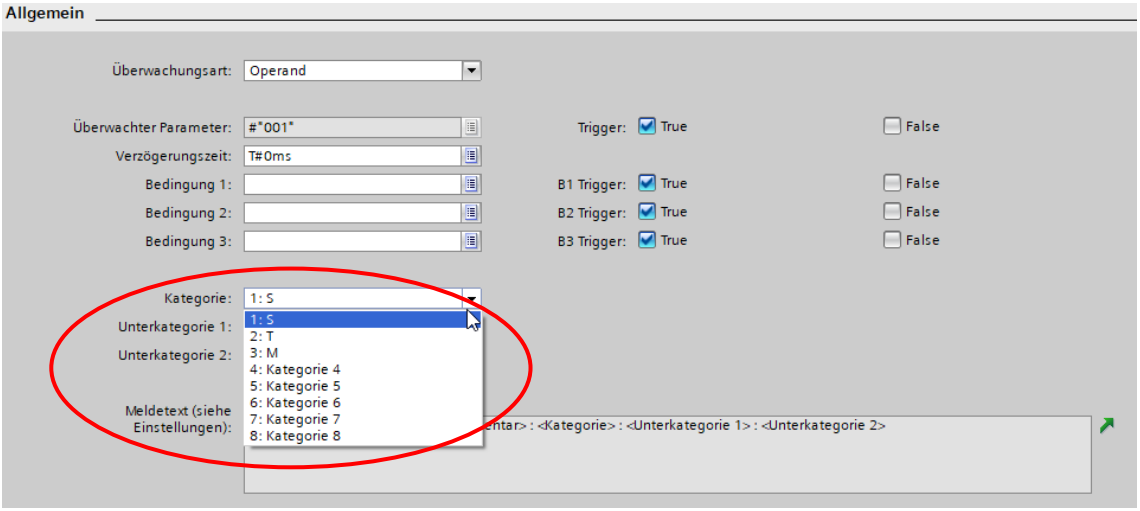


Fig. 5.68 Selection of categories

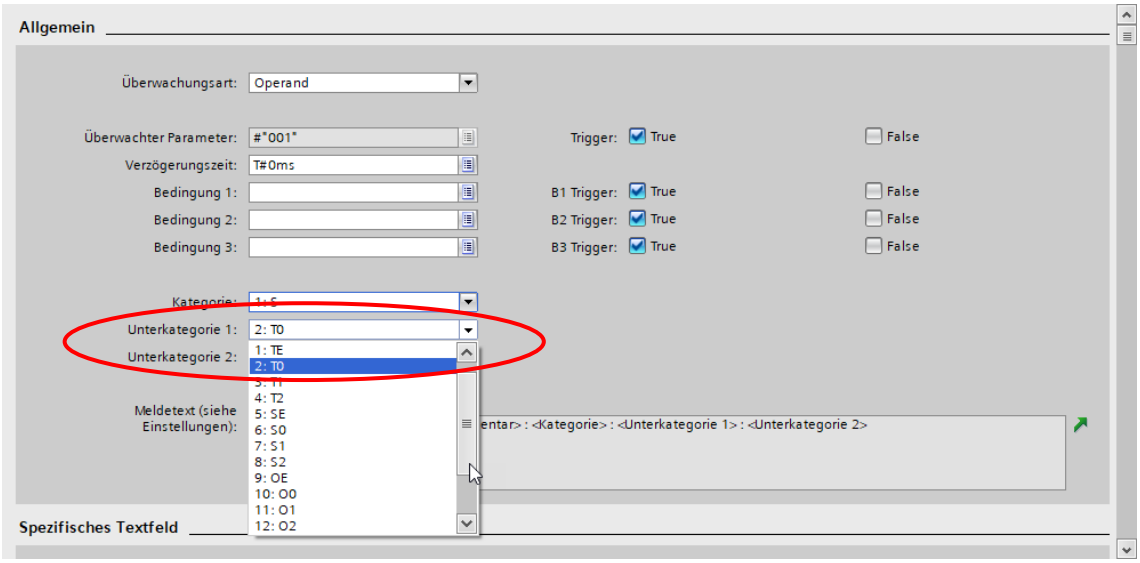


Fig. 5.69 Selection of subcategories

Definitions of terms

Term	Definition
Faults	Result in immediate system standstill
Messages	do not immediately result in system standstill
TPM	"Total Productive Maintenance" (preventative maintenance)
Technical	Anything that is caused by technical components in the system itself does not have a direct influence on the plant
System-related	Fault / message caused by external systems or by process-related wear
Organizational	Fault / message caused by the plant operator
EW	Initial value identification. First fault / message in the PLC after free from errors

Tab. 5.3 Definition of terms for the signaling system

The abbreviations to be used for entering the categories are defined in the table below.

1st character	Alarm class (category)	
	S	Fault
	M	Message
	T	TPM
2nd character	**Message type (subcategory)**	
	T	Technical
	S	System-related
	O	Organizational
3rd character	**Radio relevance**	
	E	Forwarding with initial value
	0	No forwarding
	1	General forwarding
	2	General forwarding; no entry in the ZAÜ database
4th to 7th character	**Radio number**	
	An optional radio number can be entered here.	

Tab. 5.4 Abbreviation for entries

5.3.5.3.8 Preparation of the symbol tags for visualization

NIBBLE

A nibble is a volume of data comprising the 4 bits. 1 BYTE has 2 nibbles

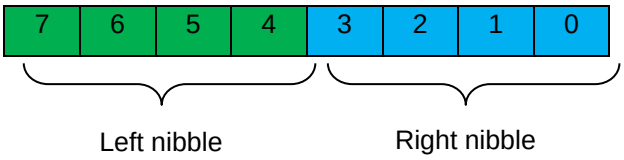


Fig. 5.70 Structure of a nibble

COLOR ANIMATION OF AREAS/FLAGS

Areas / flags have 4 bits for color animation. The permitted range is from 0…F (hexadecimal).

Program example:

```
dwVisuWert1.B0 := BYTE#16#00; (* Delete Visu value *)
IF #Condition1 THEN // Describe the example of a right nibble
dwVisuWert1.B0 := dwVisuWert1.B0 OR BYTE#16#02; (*02: Area GN   Text BK*)
END_IF;
IF #Condition2 THEN // Describe the example of a left nibble
dwVisuWert1.BO := dwVisuWert1.B0 OR SHL (BYTE#16#02 ,N:=4) (*02: Area GN   Text BK*)
END_IF;
```

COLOR ANIMATION of FRAME

Frames consist of a byte with 2 nibbles

The right nibble contains the color animation. The permitted range is from 0…F (hexadecimal).

The left nibble indicates which frame is to be visible (0 = frame 1 / F = frame 16)

If only one frame is used, the left nibble is not needed.

Program example

```
dwVisuWert1.B1 := BYTE#16#00; (* Delete Visu value *)
IF #Condition1 THEN // Describe the example of a right nibble color
dwVisuWert1.B0 := dwVisuWert1.B0 OR BYTE#16#02; (*02: Area GN   Text BK*)
END_IF;
IF #Condition2 THEN // Describe the example of a left nibble visibility
dwVisuWert1.BO := dwVisuWert1.B0 OR SHL (BYTE#16#00 ,N:=4) (*00 frame 1 is visible*)
END_IF;
```

COLOR ANIMATION of BUTTON

A button consists of a byte

Bit 0 to 6 → Color animation and visibility

Bit 7 → Enable display

Program example

```
dwVisuWert1.B1 := BYTE#16#00; (* Delete Visu value *)
IF #Condition1 THEN // Describe the example bit 0..6 color and visibility
dwVisuValue1.B1 := dwVisuValue1.B1 OR BYTE#16#02; (*02: Area GN   Text BK*)
END_IF;
IF #Condition2 THEN // Describe the example bit 7 enable display
dwVisuWert1.B1 := dwVisuWert1.B1 OR 128; (*128: Bit 7 is controlled*)
END_IF;
```

Color code table:

```
(*********Farbcodetabelle**********)
(* BK=Black/Schwarz, BU=Blue/Blau,*)
(* GN=Green/Gruen, GR=Gray/Grau,  *)
(* LT=Light/Hell, WH=White/Weiss, *)
(* YE=Yellow/Gelb, RD=Red/Rot,    *)
(* st=Statisch, bl=blinkend       *)
(* HEX Flaechenfarben  Textfarben *)
(* 00: Flaeche GR   st Text BK st *)
(* 01: nicht sichtbar             *)
(* 02: Flaeche GN   st Text BK st *)
(* 03: Flaeche BU   st Text WH st *)
(* 04: Flaeche RD   st Text WH st *)
(* 05: Flaeche LTBU st Text WH st *)
(* 06: Flaeche YE   st Text BK st *)
(* 07: Flaeche LTBU st Text YE st *)
(* 08: Flaeche GR   st Text GN st *)
(* 09: Flaeche GR   st Text BU st *)
(* 0A: Flaeche GR   st Text YE st *)
(* 0B: Flaeche GN   bl Text BK st *)
(* 0C: Flaeche BU   bl Text WH bl *)
(* 0D: Flaeche RD   bl Text WH bl *)
(* 0E: Flaeche LTBU bl Text WH bl *)
(* 0F: Flaeche YE   bl Text BK st *)
(* 10: Flaeche GR   st Text GN bl *)
(* 11: Flaeche GR   st Text BU bl *)
(* 12: Flaeche GR   st Text YE bl *)
(* 13: Flaeche GN   st Text YE bl *)
(* 14: Flaeche BU   st Text YE bl *)
(* 15: Flaeche LTBU st Text YE bl *)
(* ******************************)
```

Fig. 5.71 Color code table

Color animation

Wert	Fläche grau	Fläche grün	Fläche blau	Fläche rot	Fläche hellblau	Fläche gelb	Text schwarz	Text weiss	Text grün	Text blau	Text gelb	sichtbar	Text blinkend	Fläche blinkend	Buttons	Flächen Rahmen	Flags	Bemerkung
0	X						X					X			X	X	X	Grundzustand
1															X	X	X	ausgeblendet
2		X					X					X			X	X	X	
3			X					X				X			X	X	X	
4				X				X				X			X	X	X	
5					X			X				X			X	X	X	
6						X	X					X			X	X	X	
7					X						X	X			X	X	X	
8	X								X			X			X	X		
9	X									X		X			X	X		
10	X										X	X			X	X		
11		X					X					X		X	X	X		
12			X					X				X	X	X	X	X		
13				X				X				X	X	X	X	X		
14					X			X				X	X	X	X	X		
15						X	X					X		X	X	X		
16	X								X			X	X		X			
17	X									X		X	X		X			
18	X										X	X	X		X			
19		X									X	X	X		X			
20			X								X	X	X		X			
21					X						X	X	X		X			
22															X			
23															X			
24															X			
25															X			
26															X			
27															X			
28															X			
29															X			
30															X			
31															X			

Fig. 5.72 Color animation

COLOR ANIMATION special case end positions A - H

Bit distribution in "dwVisuValues":

Color	End position A	End position B	End position C	End position D	End position E	End position F	End position G	End position H
Area GN Text BK	0	1	2	3	4	5	6	7
Area BU Text WH	8	9	10	11	12	13	14	15
Area RD Text WH flashing	16	17	18	19	20	21	22	23
Visibility	24	25	26	27	28	29	30	31

Tab. 5.5 Bit distribution in "dwVisuWerte"

5.3.5.3.9 Animation navigation

For improved user guidance, the messages should also then be assigned, depending on the use, (station, robot, robot process device, machinery process device).

```
REGION (/* Animation Navigation */)
    // Nicht benötigte sind zu löschen

    //- Vorlage für Arbeitsgruppenglobale Bausteine (Navigationszeile, Grundbildbutton)
    //-- Störung
    "DB_ARG".VisuSS.Global_Stoer := "DB_ARG".VisuSS.Global_Stoer OR #xStoe;
    //-- Warnung
    "DB_ARG".VisuSS.Global_Warn := "DB_ARG".VisuSS.Global_Warn OR #xWarn;
    //-- Wartung
    "DB_ARG".VisuSS.Global_Wart := "DB_ARG".VisuSS.Global_Warn OR #xWart;

    //- Vorlage für Stationsbezogene Bausteine (Navigationszeile, Stationsbutton)
    //-- Störung
    "DB_ARG".VisuSS.Station_Stoer := "DB_ARG".VisuSS.Station_Stoer OR #xStoe;
    //-- Warnung
    "DB_ARG".VisuSS.Station_Warn := "DB_ARG".VisuSS.Station_Warn OR #xWarn;
    //-- Wartung
    "DB_ARG".VisuSS.Station_Wart := "DB_ARG".VisuSS.Station_Warn OR #xWart;

    //- Vorlage für Roboter Bausteine
    //-- Störung
    "DB_ARG".VisuSS.Rob_Stoer := "DB_ARG".VisuSS.Rob_Stoer OR #xStoe;
    //-- Warnung
    "DB_ARG".VisuSS.Rob_Warn := "DB_ARG".VisuSS.Rob_Warn OR #xWarn;
    //-- Wartung
    "DB_ARG".VisuSS.Rob_Wart := "DB_ARG".VisuSS.Rob_Warn OR #xWart;

    //- Vorlage für Roboterprozessgeräte Bausteine
    //-- Störung
    "DB_ARG".VisuSS.RobPG_Stoer := "DB_ARG".VisuSS.RobPG_Stoer OR #xStoe;
    //-- Warnung
    "DB_ARG".VisuSS.RobPG_Warn := "DB_ARG".VisuSS.RobPG_Warn OR #xWarn;
    //-- Wartung
    "DB_ARG".VisuSS.RobPG_Wart := "DB_ARG".VisuSS.RobPG_Warn OR #xWart;

    //- Vorlage für Maschinenprozessgeräte Bausteine
    //-- Störung
    "DB_ARG".VisuSS.MaPG_Stoer := "DB_ARG".VisuSS.MaPG_Stoer OR #xStoe;
    //-- Warnung
    "DB_ARG".VisuSS.MaPG_Warn := "DB_ARG".VisuSS.MaPG_Warn OR #xWarn;
    //-- Wartung
    "DB_ARG".VisuSS.MaPG_Wart := "DB_ARG".VisuSS.MaPG_Warn OR #xWart;

END_REGION
```

Fig. 5.73 Animation navigation

To this end, ""DB_ARG".VisuSS" contains the following bits which can be set:

DB_ARG

	Name	Datentyp	Startwert	Remanenz	Erreichbar a..	Schrei...	Sichtbar i...	Einstellwert
16	GlobAnwMeldungen	"HMI-UDT_GlobAnw"		☑	☑	☑	☑	☑
17	Proz_Nr_Rob	"HMI-UDT_Rob_Korr		☑	☑	☑	☑	☑
18	VisuSS	"ST_Visu_FB114"		☑	☑	☑	☑	☑
19	Global_Stoer	Bool	false	☑	☑	☑	☑	☐
20	Global_Warn	Bool	false	☑	☑	☑	☑	☐
21	Global_Wart	Bool	false	☑	☑	☑	☑	☐
22	Station_Stoer	Bool	false	☑	☑	☑	☑	☐
23	Station_Warn	Bool	false	☑	☑	☑	☑	☐
24	Station_Wart	Bool	false	☑	☑	☑	☑	☐
25	Rob_Stoer	Bool	false	☑	☑	☑	☑	☐
26	Rob_Warn	Bool	false	☑	☑	☑	☑	☐
27	Rob_Wart	Bool	false	☑	☑	☑	☑	☐
28	RobPG_Stoer	Bool	false	☑	☑	☑	☑	☐
29	RobPG_Warn	Bool	false	☑	☑	☑	☑	☐
30	RobPG_Wart	Bool	false	☑	☑	☑	☑	☐
31	MaPG_Stoer	Bool	false	☑	☑	☑	☑	☐
32	MaPG_Warn	Bool	false	☑	☑	☑	☑	☐
33	MaPG_Wart	Bool	false	☑	☑	☑	☑	☐

Fig. 5.74 Bits that must be set in the ""DB_ARG".VisuSS"

An animation for "Station" is programmed out in the example "Ventil".

```
640 REGION (* Animation Navigation *)
641 (* Anzeige fuer Stationsbezogene Bausteine [HMI Template Taste der Station] *
642     #xStoe := #"001"
643     OR #"002"
644     OR #"003"
645     OR #"004"
646     OR #"005"
647     OR #"006"
648     OR #"007"
649     OR #"008"
650     OR #"009"
651     OR #"010"
652     OR #"011"
653     OR #"020";
654     "DB_ARG".VisuSS.Station_Stoer := "DB_ARG".VisuSS.Station_Stoer OR #xStoe;
655
656     #xWarn := #"029"
657     OR #"030";
658
659     "DB_ARG".VisuSS.Station_Warn := "DB_ARG".VisuSS.Station_Warn OR #xWarn;
660
661     #xWart := FALSE;
662
663     "DB_ARG".VisuSS.Station_Wart := "DB_ARG".VisuSS.Station_Wart OR #xWart;
664 END_REGION
```

Fig. 5.75 Programmed out animation for "Station"

Use of templates for creating blocks

Even if the creation of a functional block was explained in the previous chapters, it is generally useful to use the available structure block as a template.

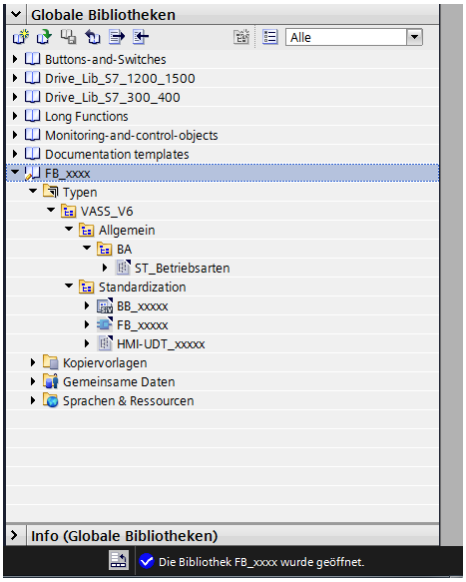


Fig. 5.76 Structure block

5.3.6 Interconnecting structures

Interconnecting structures are used by the user for interconnecting the parameters of an FB call in the user program.

Interconnecting structures reduce the tag declarations and simplify parameterization.

Interconnecting structures must always be defined if a large number of tags are repeatedly required in the same way for the external circuit of a block.

The symbol for these interconnecting structures begins with "STB_", e.g. "STB_Ventil" (interconnecting structure for the FB_Ventil).

The UDTs are stored in the VASS library in the folder of the module together with the module FB.

With regard to the designation of the UDT, the name must match that of the associated module FB.

Both a tag structure of the data type UDT and a single tag from the structure can be used as a formal operand of an FC/FB.

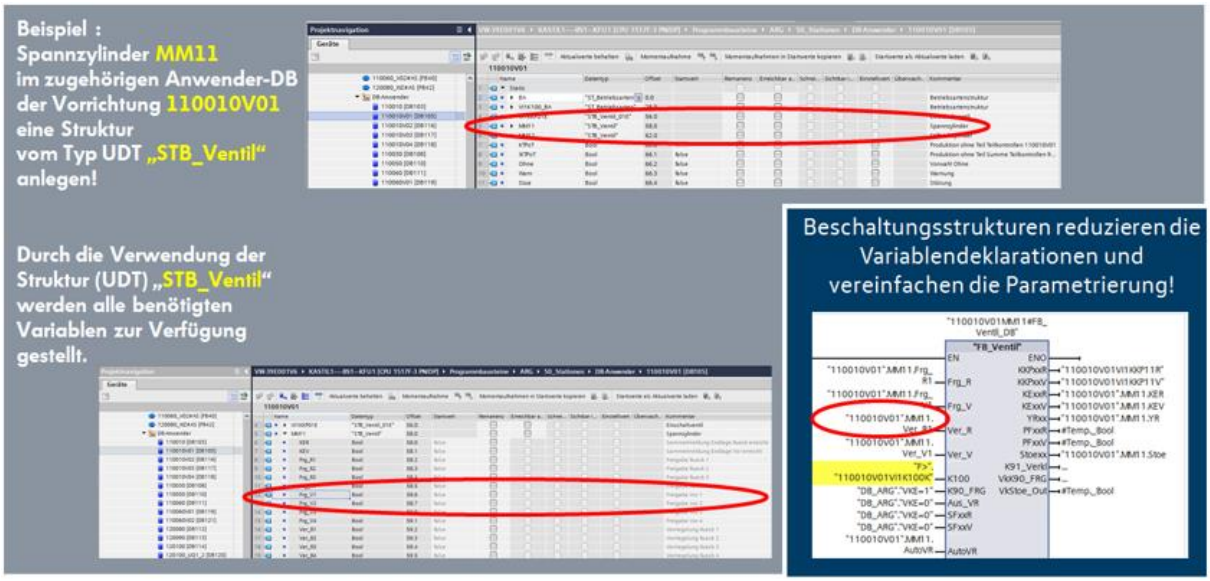


Fig. 5.77 Interconnecting structures

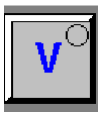
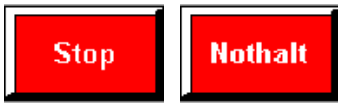
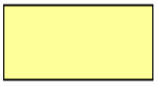
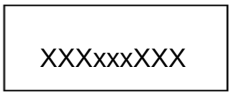
6 WinCC Advanced and SiVArc

6.1 Working in WinCC - Basic project

6.1.1 General observation on visualization elements

6.1.1.1 Colors in WinCC Advanced

	Colors	View		
Permitted colors		Color code RGB		
		Red	Green	blue
Red		255	0	0
Yellow		255	255	0
Light yellow		255	255	153
Blue		0	0	255
Light blue		60	140	255
Green		0	255	0
Light gray		192	192	192
Dark gray		128	128	128
Background for Button text	Light gray (inactive) Black (inactive)			
Animated background for button text R Animated button text R	Green (end position) Black (end position)			
Animated background for button text R Animated button text R	Green (end position) Yellow (user guidance)			
Animated background for button text R Animated button text R	Light gray Flashing green (control)			
Animated background for button text V Animated button text V	Blue (end position) White (end position)			
Animated background for button text V Animated button text V	Blue (end position) Yellow (user guidance)			

Animated background for button text V Animated button text V	Light gray flashing blue (control)	
Animated background for button text Pre-selections/Maintenance Animated button text pre-selections/maintenance	Light blue (selected) White (selected)	
Animated background for button text BA_Uebersicht (emergency stop, stop) Animated button text BA_Uebersicht (emergency stop, stop)	Red (active) White (active)	
Field for static faceplate texts	Light yellow	
Animated interlock display	Yellow	
Entry field text	White Black	

Tab. 6.1 Colors in WinCC Advanced

6.1.2 Creation of a faceplate

Before a faceplate can be created for a new module, the following specifications must be made:

- Faceplate and, if necessary, HMI-UDT number
- Block name
- Faceplate family name
- Faceplate version numbering system for the block version number is as follows: Version x.y.z:
  - <x = main version (is incremented by the standardizer in the event of a complete version change due to function extension)
  - y = subversion (is incremented by the standardizer in the event of changes that are adopted as part of a bug fix)
  - z = change index (is incremented in the event of code changes on the site, by the commissioning company, if necessary)

Note

In the event of the distribution of a newly created faceplate, the third position (z) must therefore always be versioned with 0.

6.1.2.1 General information on faceplates

When creating faceplates, the following size specifications relating to frame, buttons, background, etc. must be observed.

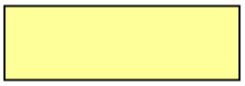
Width (pixels)	Height (pixels)	Symbol
250	70	
311	70	
372	70	
433	70	
494	70	
555	70	
616	70	

Tab. 6.2 Indication of sizes for faceplates

Note

The width of the frame is increased with each additional button by 61 (pixels).

The basis is the width 250 (pixels) with two buttons accordingly for "Ventil_03".

	Width (pixels)	Height (pixels)	Symbol
Faceplate frame	250	70	
Polygon for button 3D effect	60	59	
Background for button	60	60	
Animated background for button text	51	50	
Animated interlock display	15 x 15		
Animated button text	24	36	
Invisible button	62	61	
Background frame for static faceplate texts and end positions	123	60	
Frame for static faceplate texts	119	38	
Frame for end positions	119	17	
Static faceplate texts	119 119	15 15	
Animated background for end positions	119	17	
End position text	11	14	

Tab. 6.3 Indication of sizes for different areas of faceplates

6.1.2.2 Creating an HMI-UDT

The data type "HMI-UDT" is created for a faceplate. The structure corresponds to the visualization interface declared in the module FB in the static area of the DB.

1st stat. tag and keyword **VISUWERT**.

The address sequence must be strictly adhered to. The data types may be different within the same address range, e.g.:

DWORD ←→ WORD0, Byte2, Bit07 … Bit00. **BB Ventil_03**

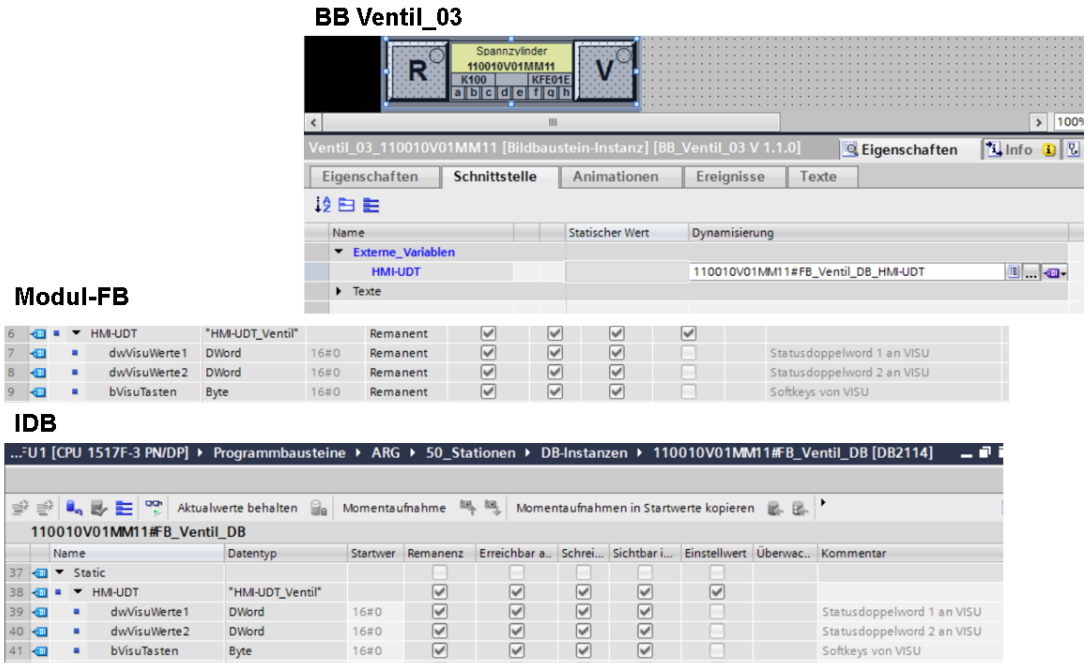


Fig. 6.1 Create an HMI-UDT

6.1.2.3 Individual objects in the faceplate

Objects in the screen layers

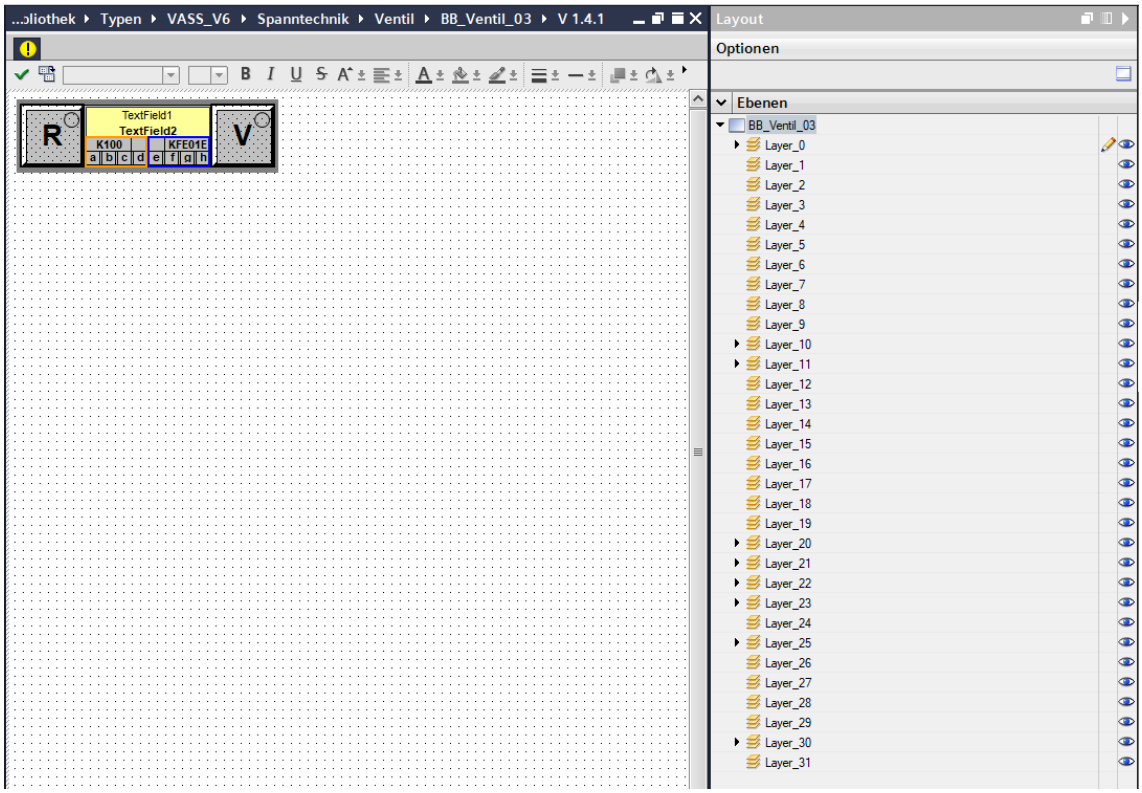


Fig. 6.2 Selection of layers

Layer 10: Ventil_03_Rahmen (rectangle)

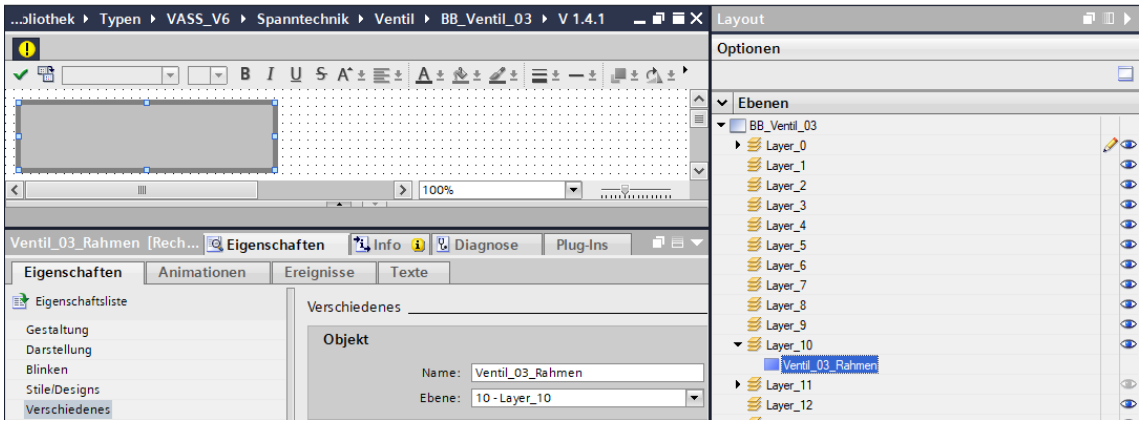


Fig. 6.3 Frame

Layer 11: Ventil_03_Innenrahmen

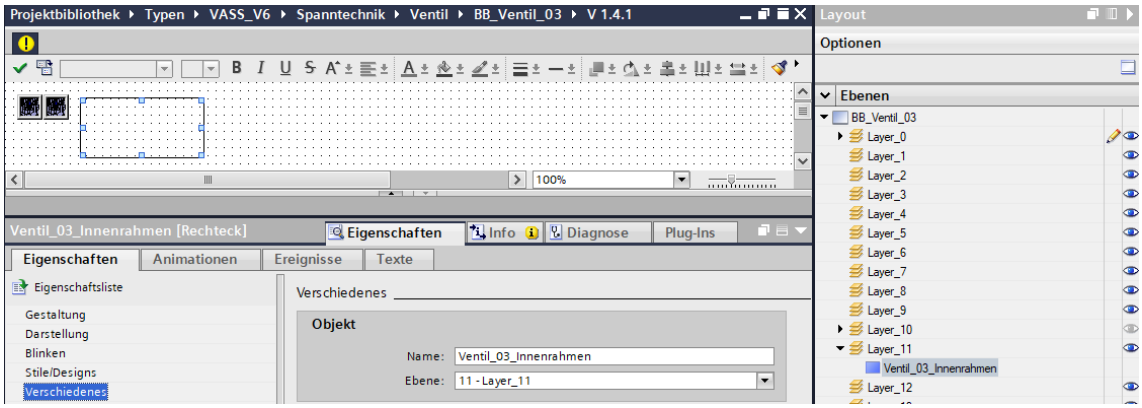


Fig. 6.4 Gutter frame

Layer 20: Ventil_03_Button_R_Dsw (rectangle)

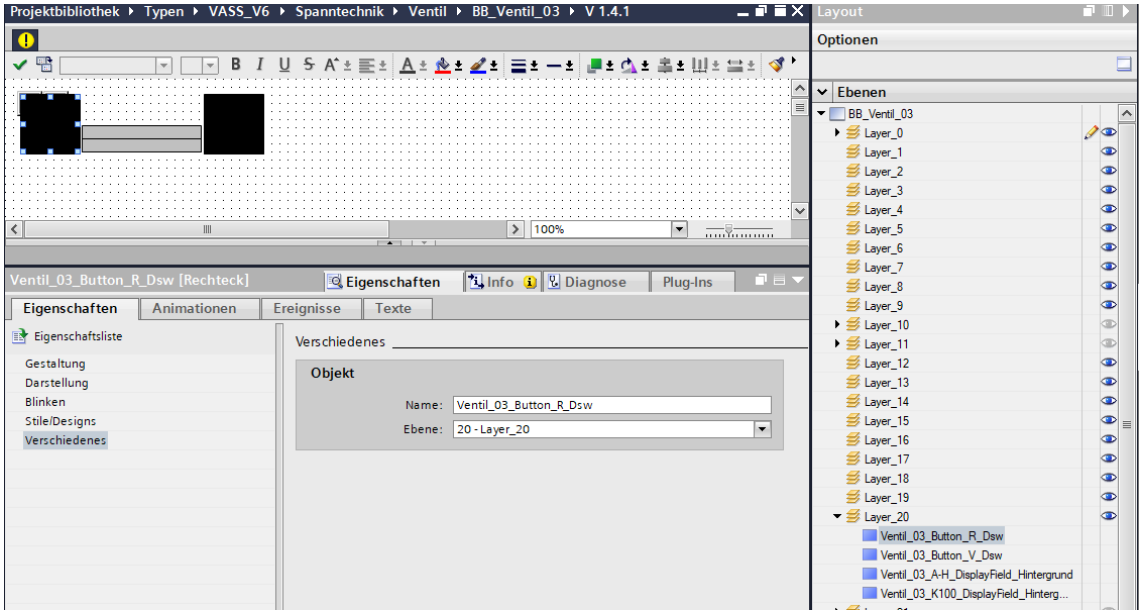


Fig. 6.5 Button Dsw

Layer 21 Ventil_03_Button_R_Dws (polygon)

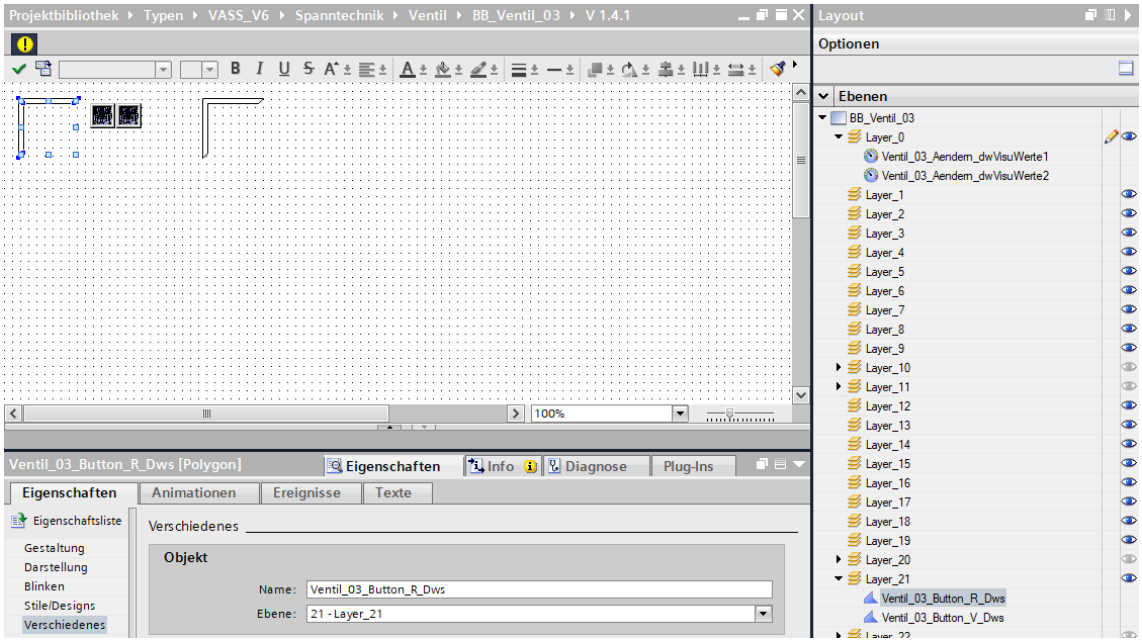


Fig. 6.6 Button Dws

Layer 22: Ventil_03_R_Button_AnimArea (text field)

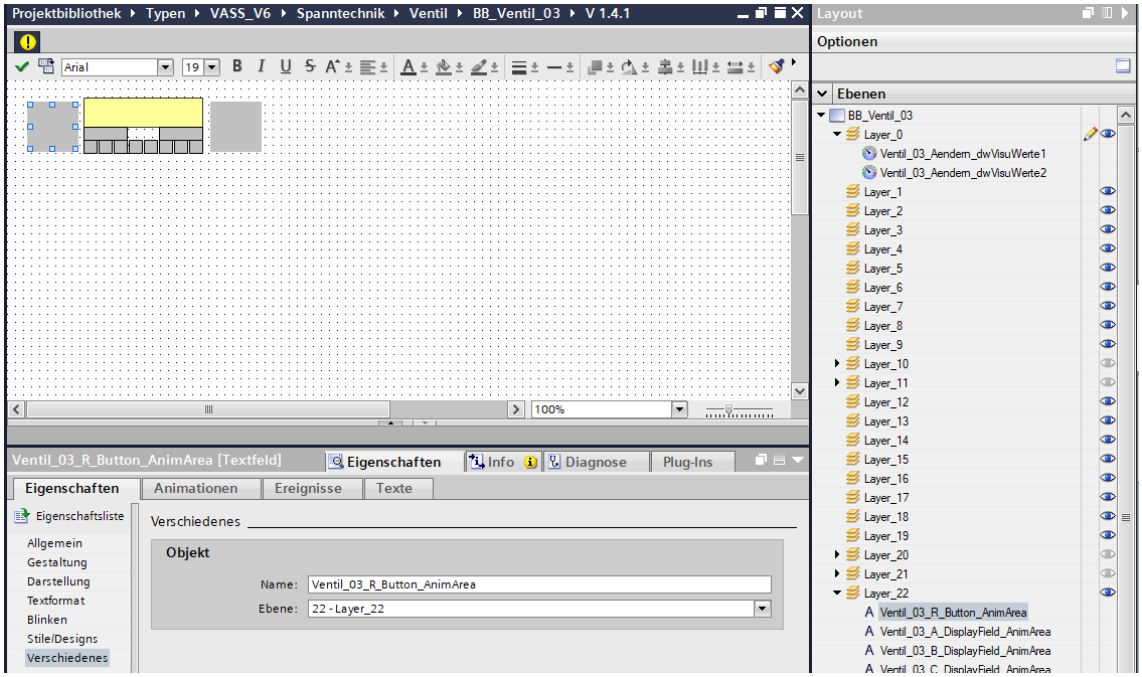


Fig. 6.7 Button AnimFlaeche

Layer 22: Ventil_03_TextField_AnimArea (text field)

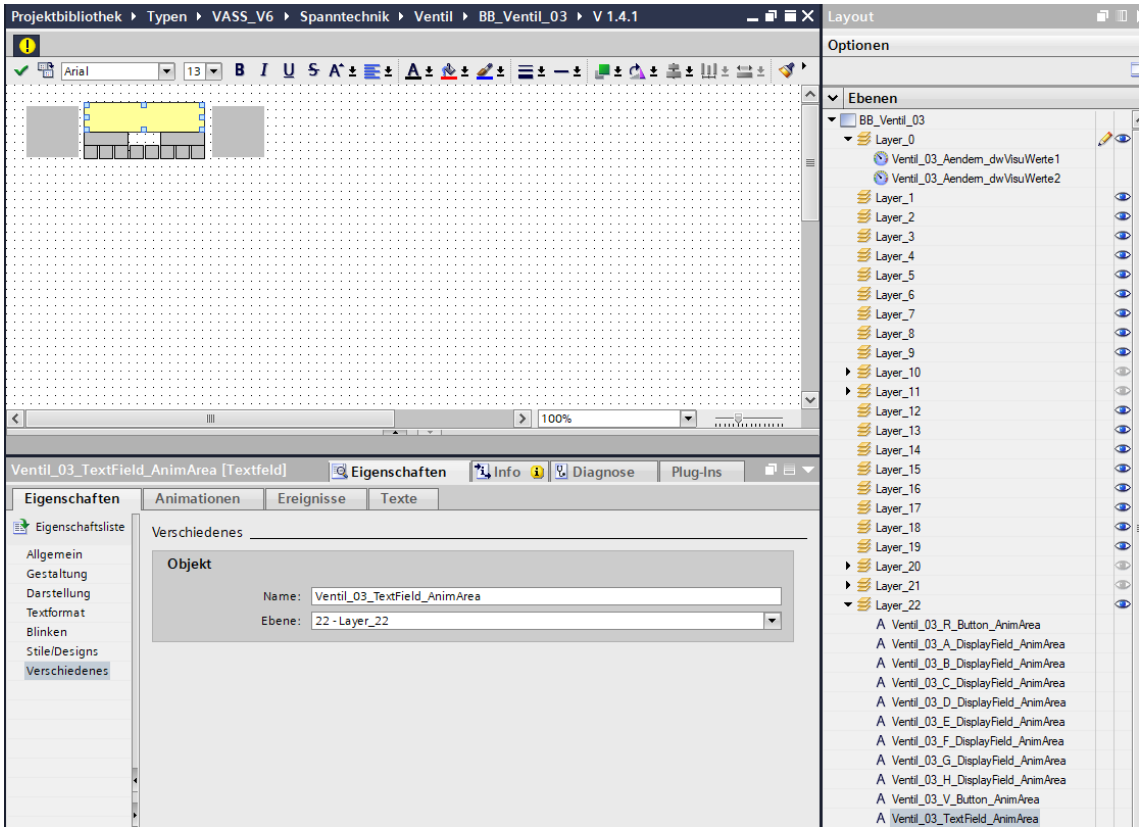


Fig. 6.8 Button AnimFlaeche

Layer 22: Ventil_03_A_DisplayField_AnimArea (text field)

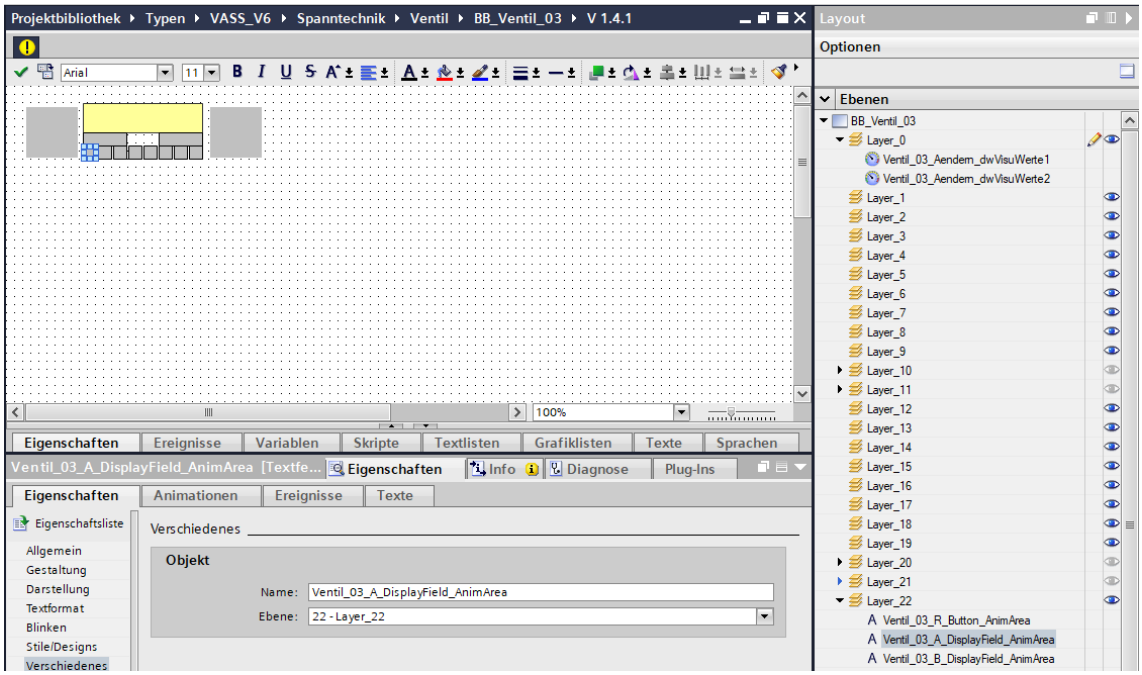


Fig. 6.9 Button AnimFlaeche

Layer 23: Ventil_03_R_Button_AnimText (text field)

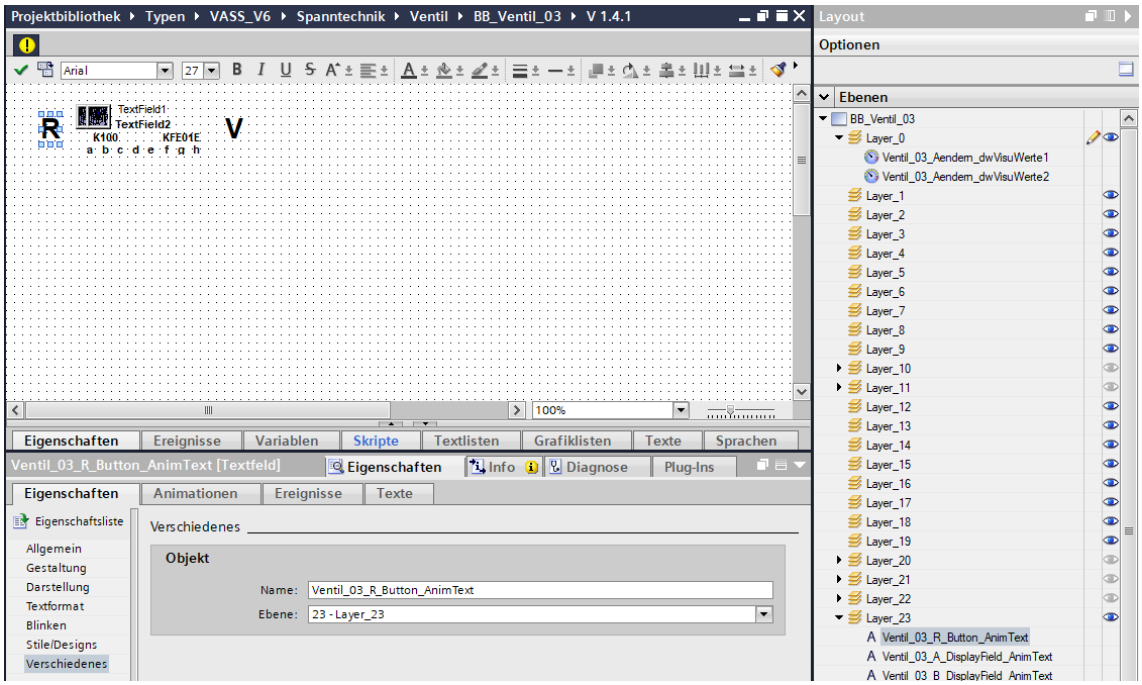


Fig. 6.10 Button AnimText

Layer 23: Ventil_03_A_DisplayField_AnimText (text field)

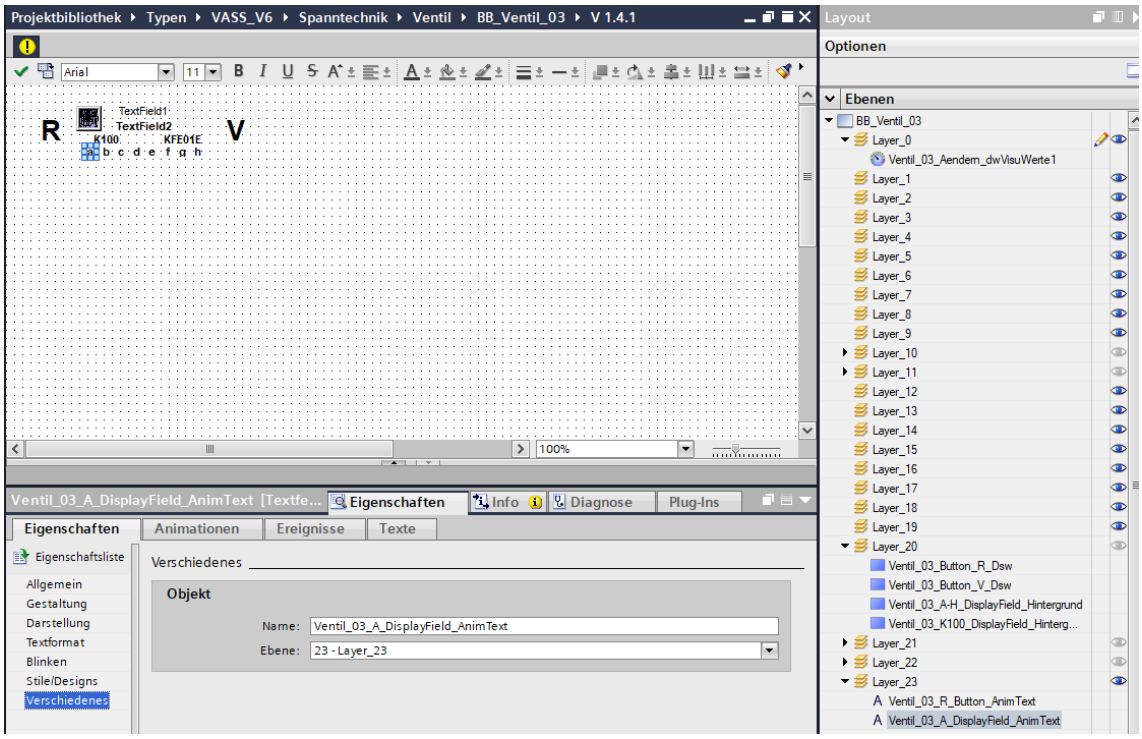


Fig. 6.11 Button AnimText

Layer 23: Ventil_03_TextField1_AnimText (text field)

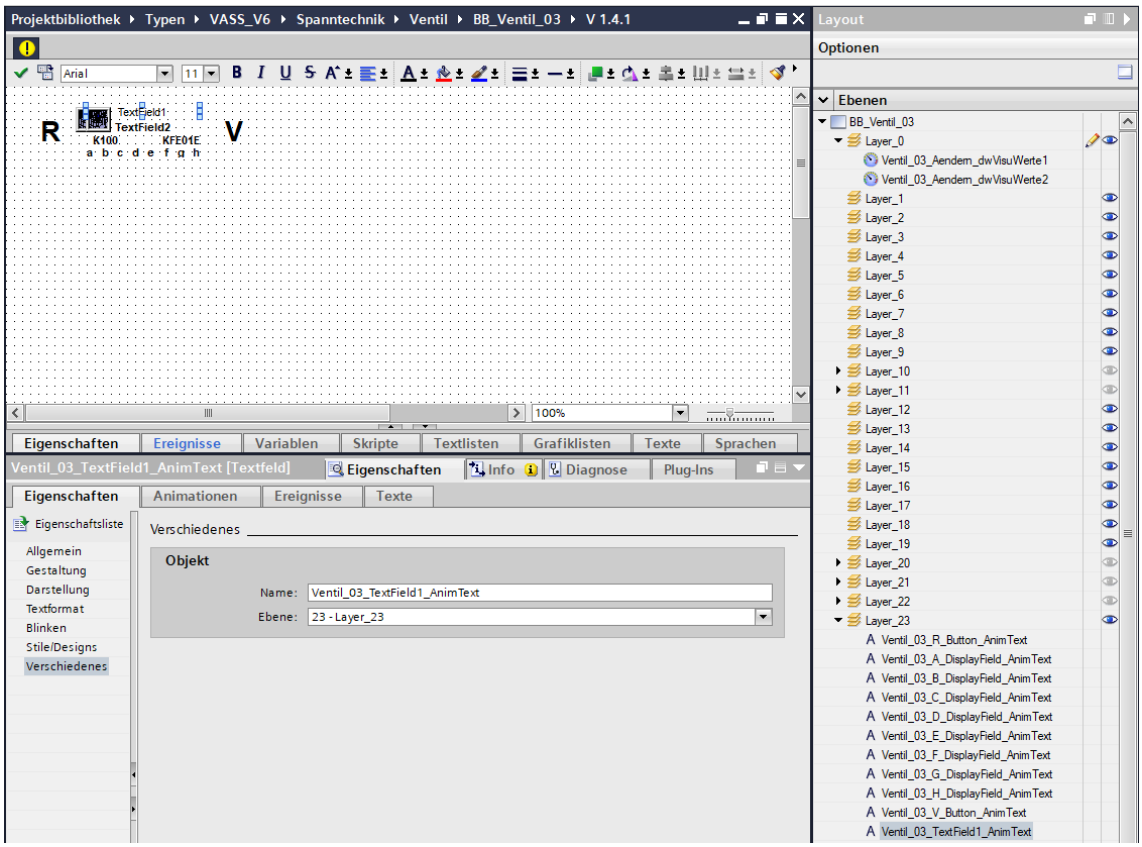


Fig. 6.12 Button AnimText

Layer 25: Ventil_03_Button_V_Verriegelung (circle)

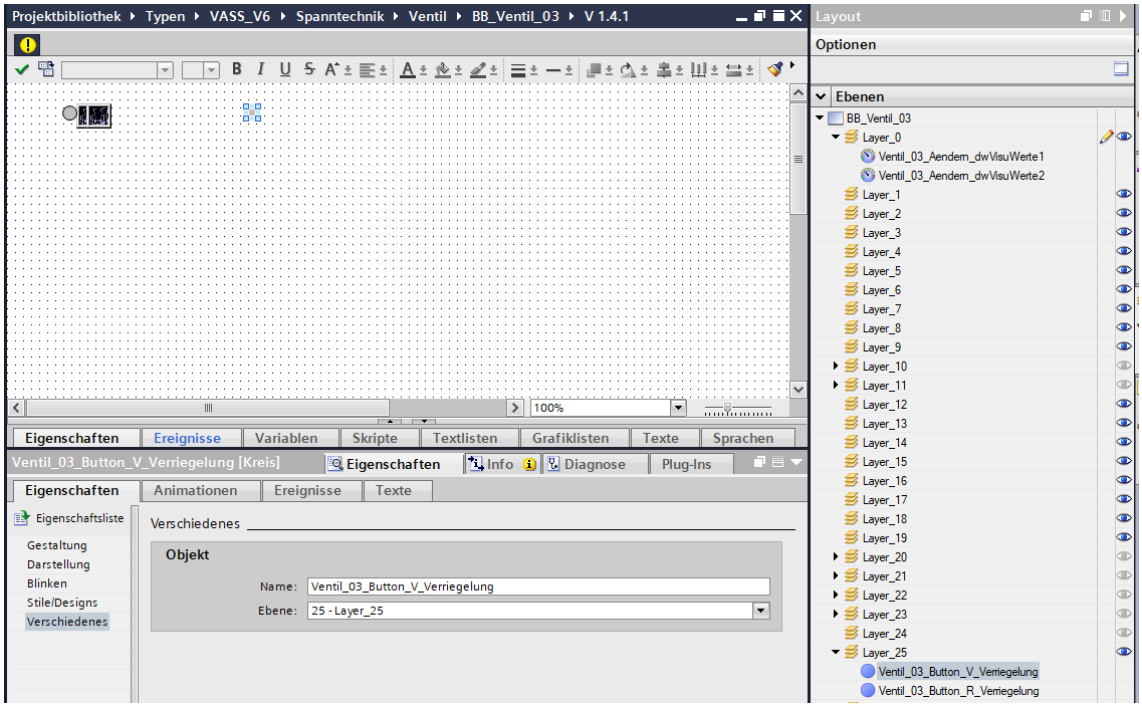


Fig. 6.13 Button lock

Layer 30: Ventil_03_Button_R_Schaltflaeche (button)

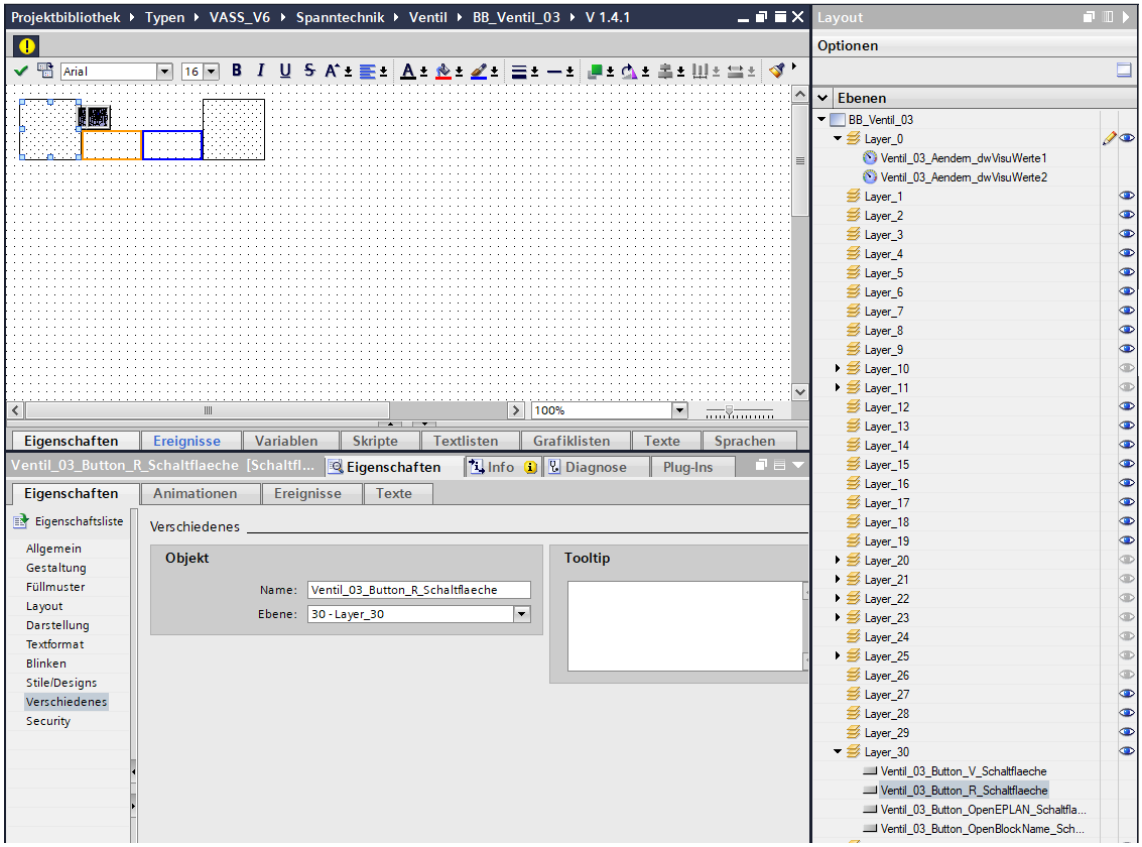


Fig. 6.14 Button button

Layer 30: Ventil_03_Button_OpenEPLAN_Schaltflaeche (button)

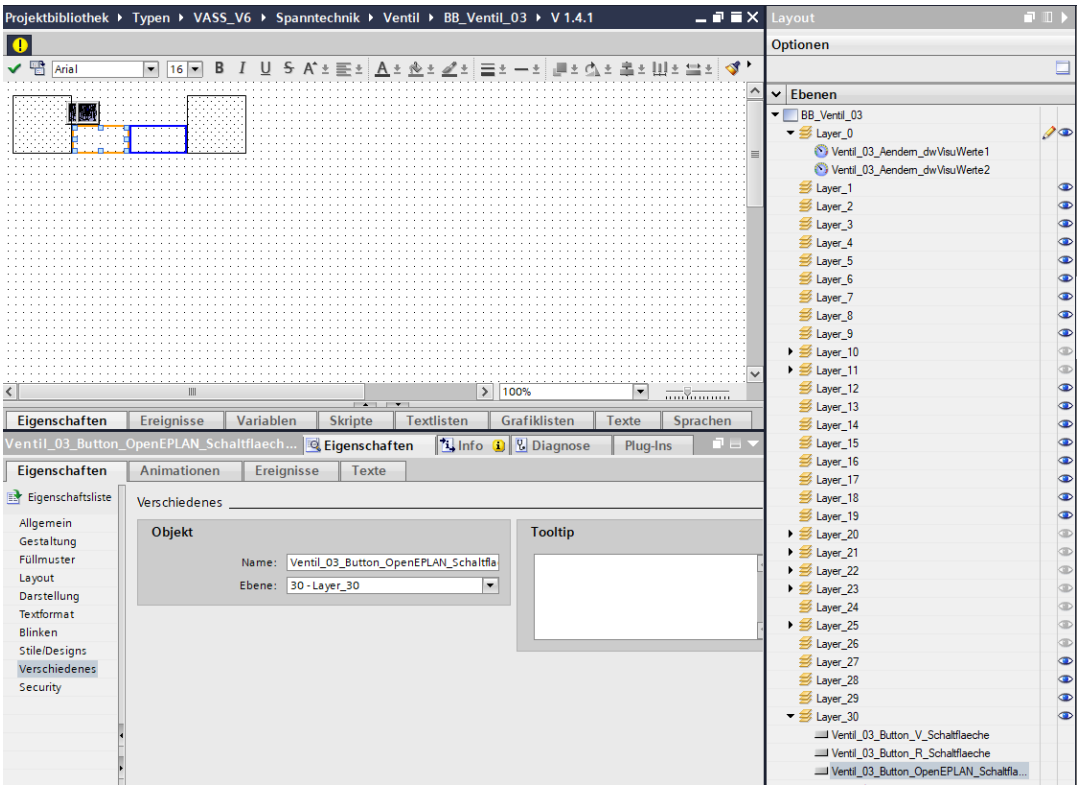


Fig. 6.15 Button button

Layer 30: Ventil_03_Button_OpenBlockName_Schaltflaeche (button)

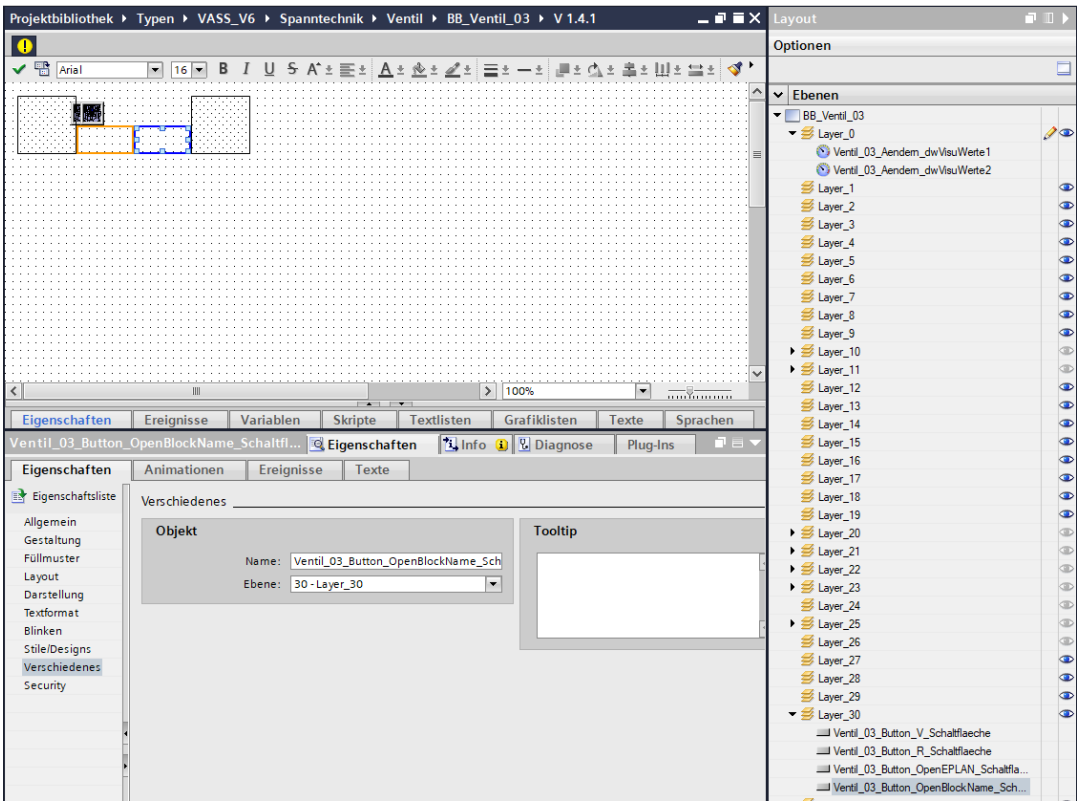


Fig. 6.16 Button button

Layer 00: Ventil_03_ Aendern_dwVisuWerte1 (pointer instrument)

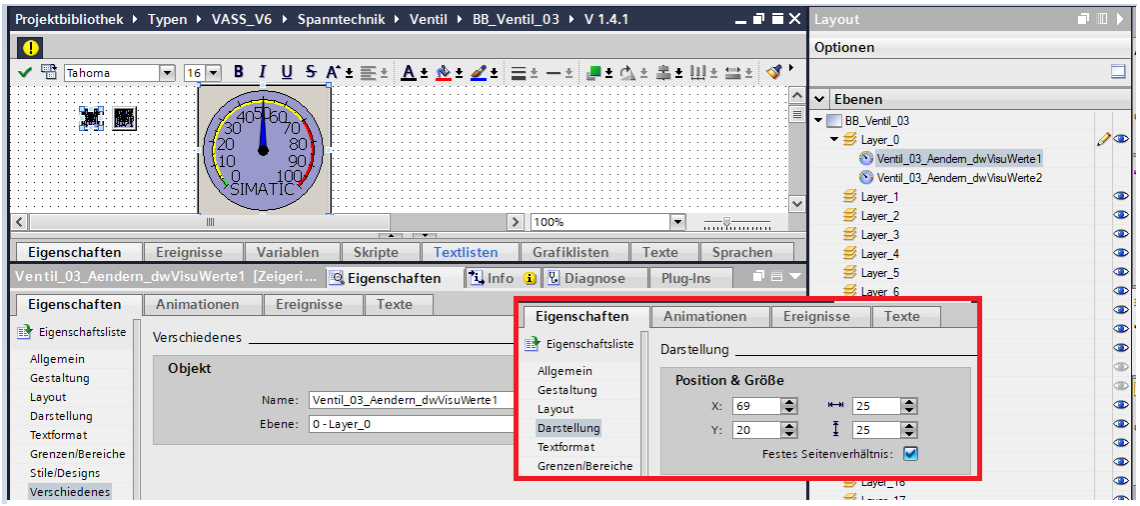


Fig. 6.17 Button pointer

One pointer instrument is required in each case for the update of the tags used in the structure (dwVisuWerte).

Note

External tags are only updated on use.

The pointer instrument **must** have a minimum size of 10x10 pixels.

The faceplates are supplied with internal tags. The internal tags are connected to the data of the external tags via scripts. Due to the connection via scripts, the tags are not automatically updated as there is no event. To achieve this, a pointer instrument is integrated, whose event serves as the trigger for the update.

To reduce the cycle time, the color animation is edited in the S7 blocks. The internal tags must be created accordingly.

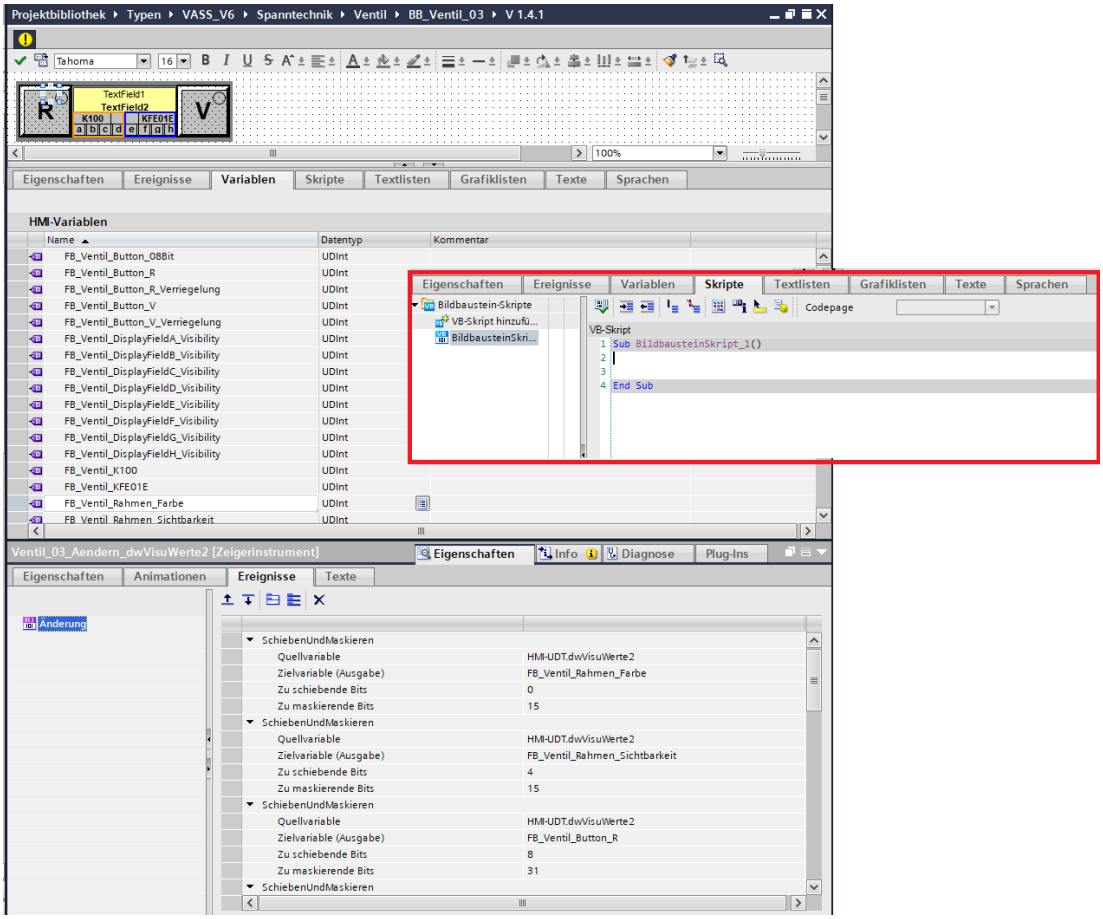


Fig. 6.18 Reduction of cycle time

WinCC Advanced - "Moving and masking" function in the VASS standard

"Shifting and masking" is used to break up a double word.

The double word is broken down on the PLC page. On the HMI side, the "shifting and masking" function enables the broken up double word to be directly addressed (e.g. animation of screen elements).

Example:

1 button (8 bits) 1 frame (8 bits) 4 areas (4 bits each)

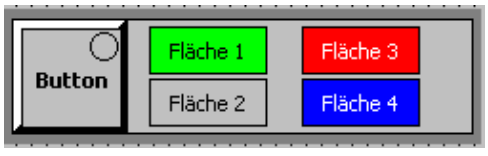


Fig. 6.19 "Shifting and masking" in the VASS standard

Position in double word (dwVisuWerte)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Area 4				Area 3				Area 2				Area 1				Frame								Button							

Tab. 6.4 Position in double word (dwVisuValues)

Note

The position of the elements is not fixed and can be individually assigned.

"Moving and masking" for buttons

- "Moving" is not necessary as the button begins at BIT 0.
- "Masked" is 255, as the entry is shown in "decimal".

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mask BYTE = DEC 255																								128	64	32	16	8	4	2	1
Target tag button																								7	6	5	4	3	2	1	0

Tab. 6.5 "Moving and masking" for buttons

"Moving and masking" for frames

- "Move" by 8 bits and "mask" with 255

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Move by 8 bits																															
Mask BYTE = DEC 255																								128	64	32	16	8	4	2	1
Target tag frame																								7	6	5	4	3	2	1	0

Tab. 6.6 "Move" by 8 bits and "mask" with 255

- "Move" by 16 bits and "mask" with 255

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Move by 16 bits																															
Mask BYTE = DEC 15																												8	4	2	1
Target tag area 1																												3	2	1	0

Tab. 6.7 "Move" by 16 bits and "mask" with 255

- "Move" by 20 bits and "mask" with 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Move by 20 bits																															
Mask BYTE = DEC 15																												8	4	2	1
Target tag area 2																												3	2	1	0

Tab. 6.8 "Move" by 20 bits and "mask" with 15

- "Move" by 24 bits and "mask" with 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Move by 24 bits																															
Mask BYTE = DEC 15																												8	4	2	1
Target tag area 3																												3	2	1	0

Tab. 6.9 "Move" by 24 bits and "mask" with 15

- "Move" by 28 bits and "mask" with 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Move by 28 bits																															
Mask BYTE = DEC 15																												8	4	2	1
Target tag area 4																												3	2	1	0

Tab. 6.10 "Move" by 28 bits and "mask" with 15

Note

If you want to "mask" words, you have to enter the mask = 65535.

6.1.2.4 Documentation

If a faceplate is released in the project library, the name of the library must be entered in the Author field as specified in Chapter 4 "Specifications for brands and plant-specific standards" .

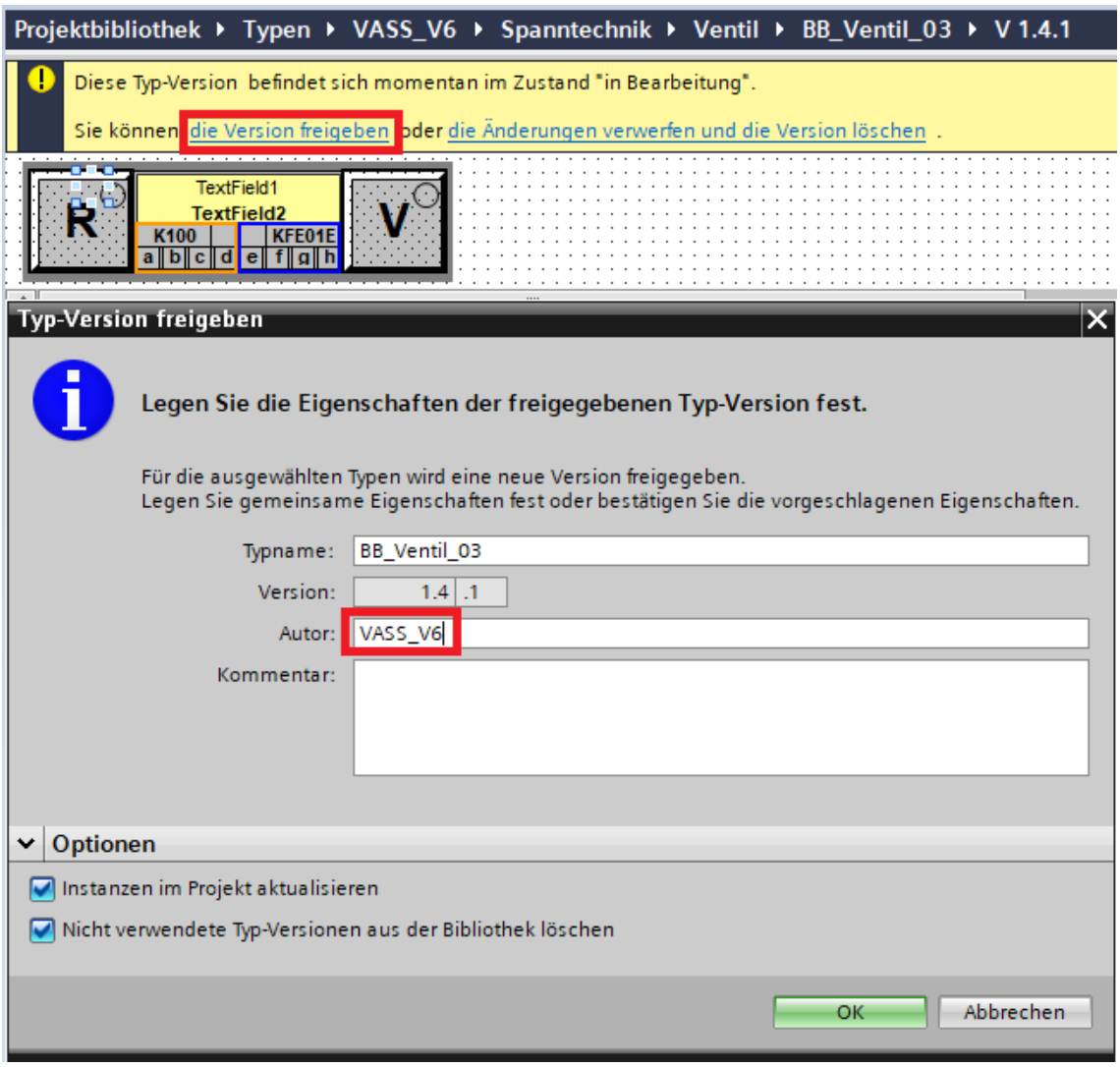


Fig. 6.20 Documentation

Note

In the event of a change or extension of the HMI-UDT in the project tree, it must also be manually updated under the associated faceplate.

6.1.3 Creation of a faceplate for a detailed screen

The standard contains detailed screens. Generally, new detailed screens can also be developed.

Faceplates for detailed screens should be provided for displaying detailed information as an addition to "03-pages or root screen", for example, drives and reading points.

If a module component requires a complex display and operating option, an associated faceplate must be created for it for a detailed screen, which is stored in the project library.

6.1.3.1 General information

For the sake of simplicity, the faceplate for "FB_SEW_AMA_Bin", for example, has been created for the description of the initial creation of the detailed screen.

Faceplates for detailed screens are named "BB_Detail_**xxxx**". The technology of the name of the associated module FB is used for **xxxx**.

In this case, for "FB_**SEW_AMA_Bin**", the faceplate for the detailed screen is "BB_Detail_**SEW_AMA_Bin**".

6.1.3.2 Storage in the project

The faceplates for detailed screens should be stored in the folder of the technology with the module_FB in the project library.

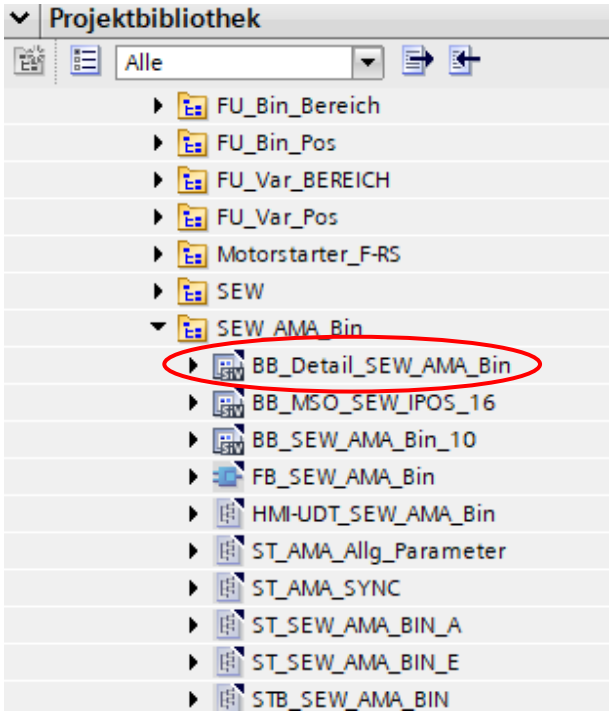


Fig. 6.21 Folders in the project library

6.1.3.3 Creating a detailed screen

Note

Generally, changes or new standard faceplates are permitted only after consultation and with the approval of the standardizing electrical engineering department in charge.

For the creation or extension of detailed screens, it is usually recommended to access existing faceplates for detailed screens or detailed screen objects of the current VASS library.

6.1.3.3.1 Layout (screen template) for detailed screens

The placement of the faceplates for detailed screens is done using screen rules for detailed screens. For each detailed screen an associated layout is created in which the faceplate is placed for the detailed screen via screen rules.

The layouts also have unique names for the respective detailed screen.

For the faceplate "BB_Detail_SEW_AMA_Bin", the layout has the name "Layout_Detail_SEW_AMA_Bin_19Z".

The layouts are in the project library in the "Master copies" folder. Newly created layouts are sorted under Master copies in a separate folder (as specified in Chapter 4 "Specifications for brand and plant-specific standards")

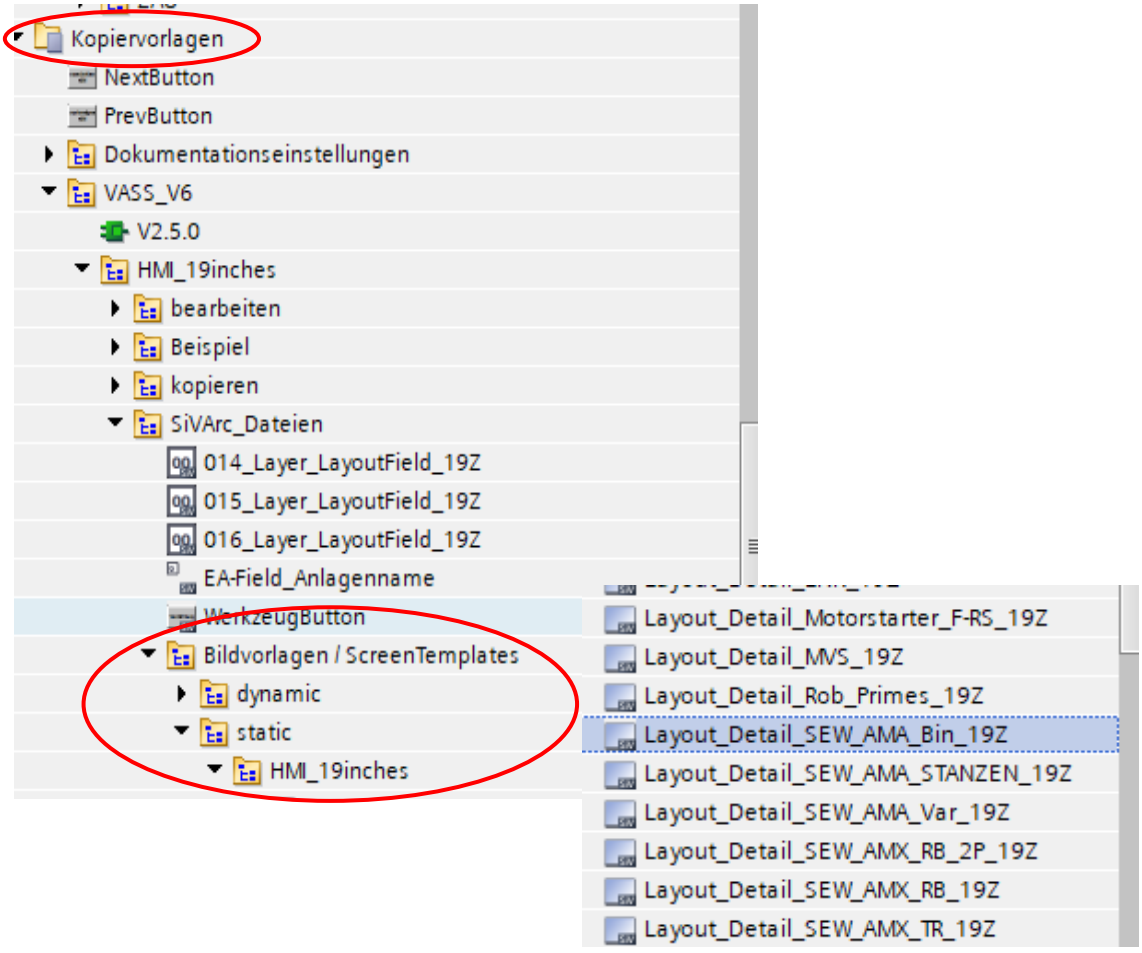


Fig. 6.22 Master copies folder in the project library

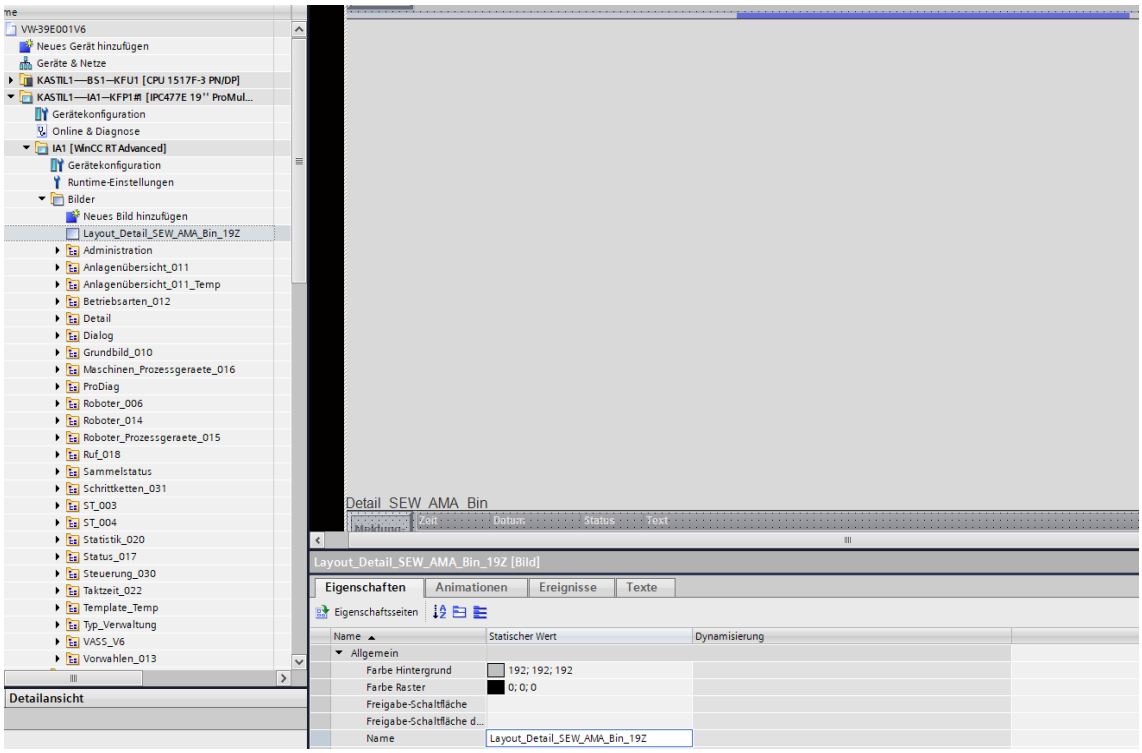


Fig. 6.23 Layout for detailed screen "SEW_AMA_Bin"

This describes how the name of the screen is formed and in which screen group it is generated as a screen.

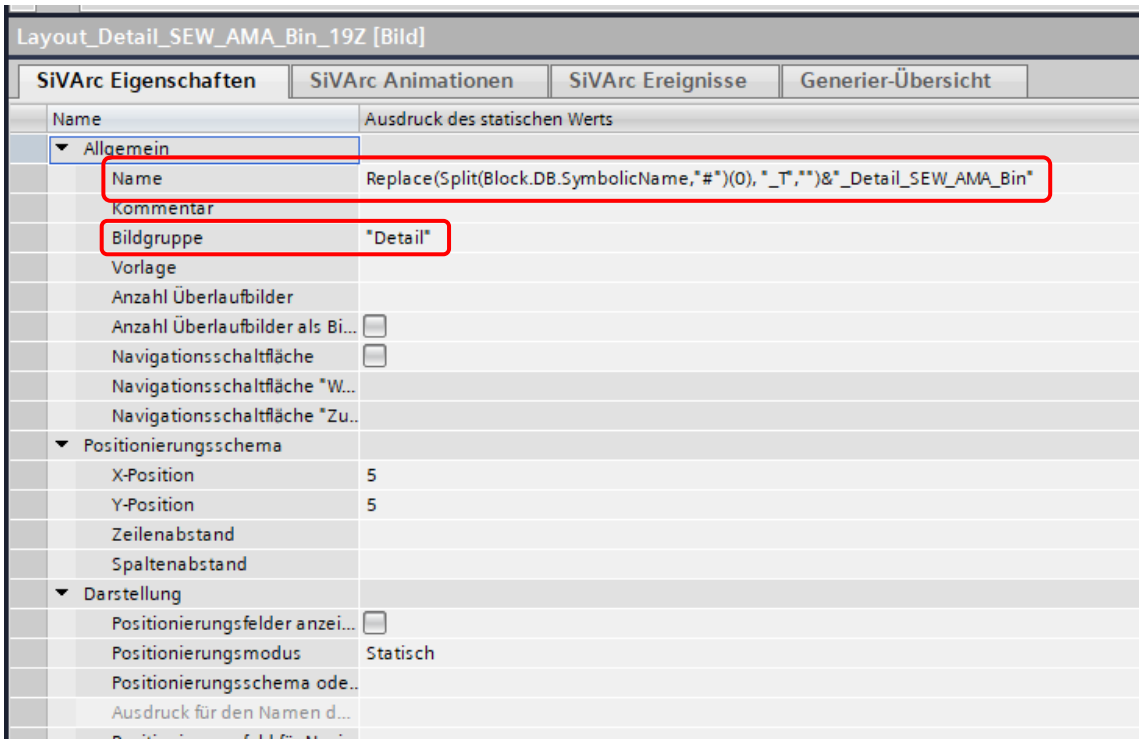


Fig. 6.24 SiVArc properties in "Layout__Detail_SEW_AMA_Bin"

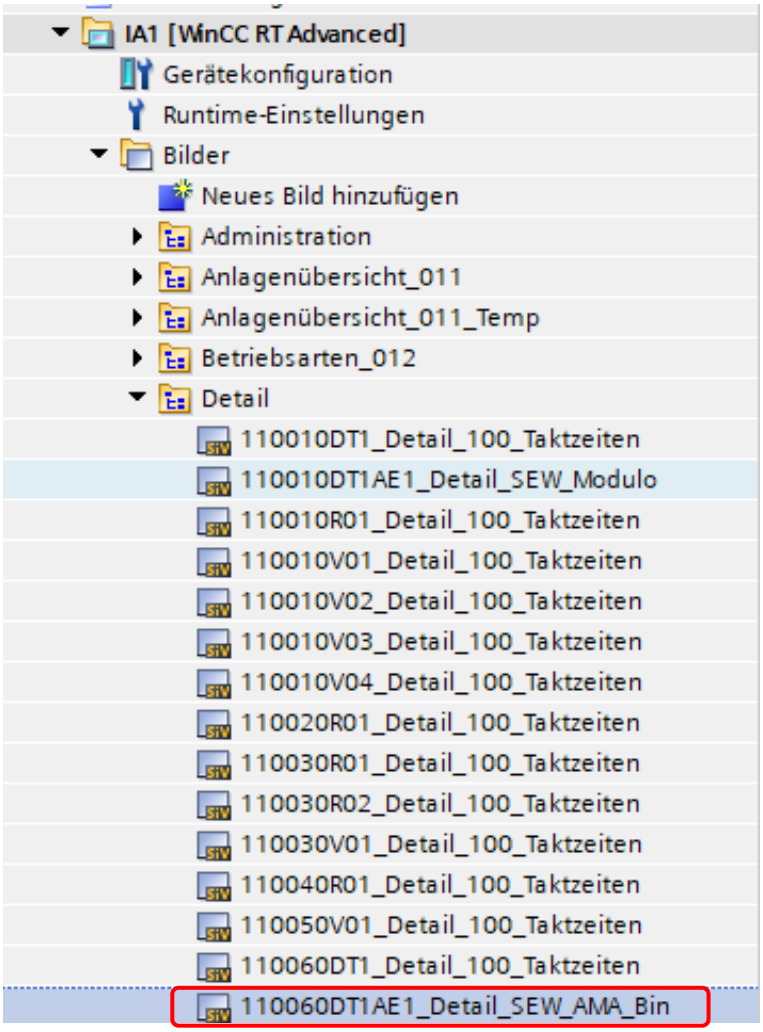


Fig. 6.25 Detailed screen for "SEW_AMA_Bin" after generation

6.1.3.3.2 Object names of objects in the faceplate for a detailed screen

A faceplate for a detailed screen consists of several objects such as text fields, buttons, graphic objects.

For a better overview, the object names of all objects begin with the keyword "Detail", followed by the faceplate name and the object name or more detailed description.

To keep the description of the creation of a detailed screen as simple and clear as possible, the faceplate is generated for the "SEW_AMA_Bin".

Example of an animated text of the button "Inch Mode" for SEW_AMA_Bin in the detailed screen

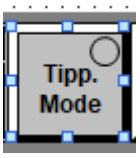


Fig. 6.26 Example of an animated text

For the text field "Inch Mode" in the detailed screen "SEW_AMA_Bin", the object name "Detail_SEW_AMA_Bin_Button_Tipp.Mode_AnimText" is specified.

The specification or renaming of the object name is done via the object properties.

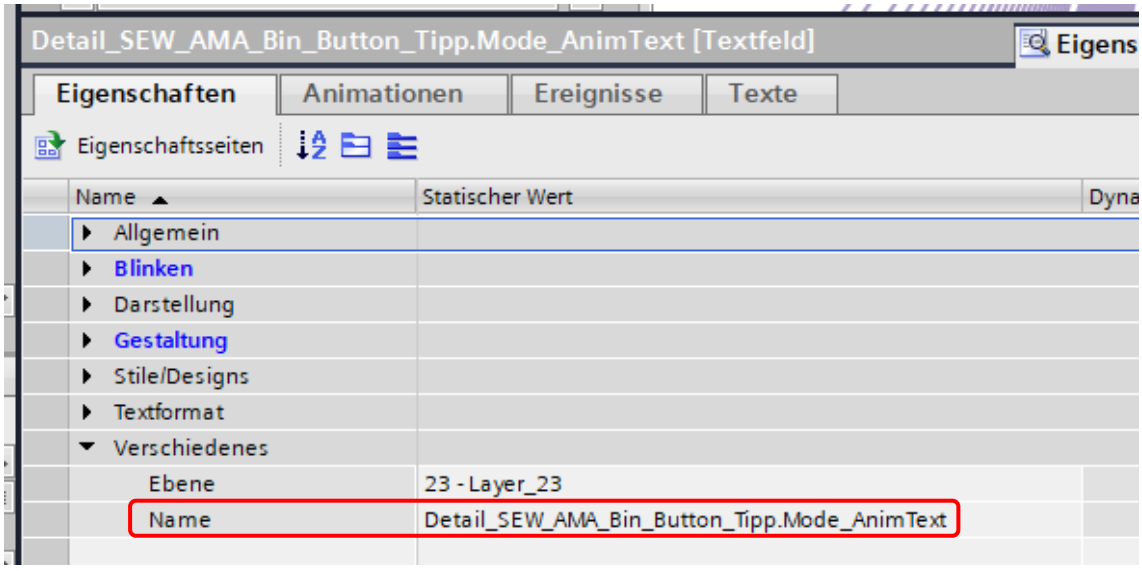


Fig. 6.27 Specification of object name

6.1.3.3.3 Assignment of objects to layers

Objects must be assigned to different layers.

It is therefore possible to show or hide objects in the detailed screen.

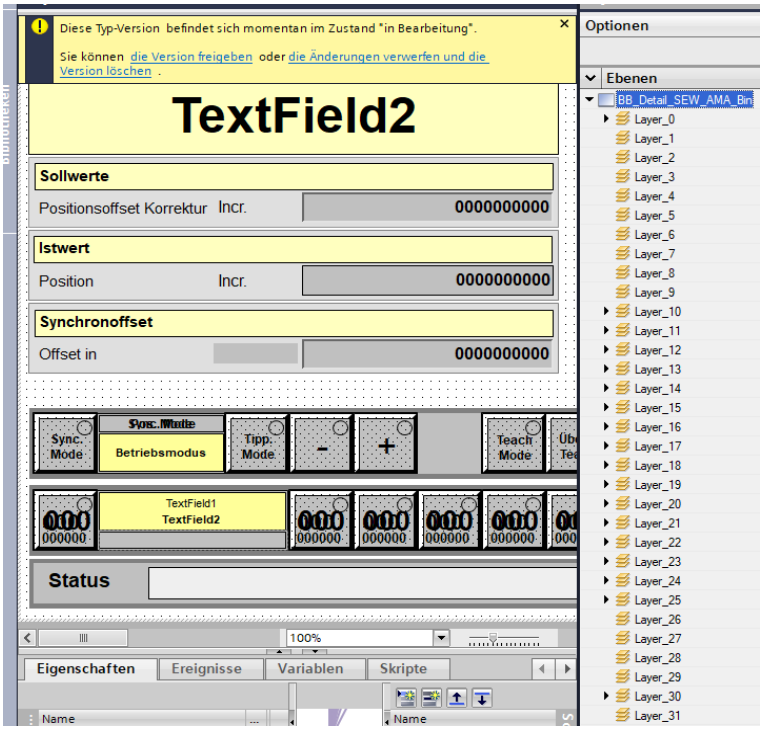


Fig. 6.28 Faceplate for detailed screen with display of all layers

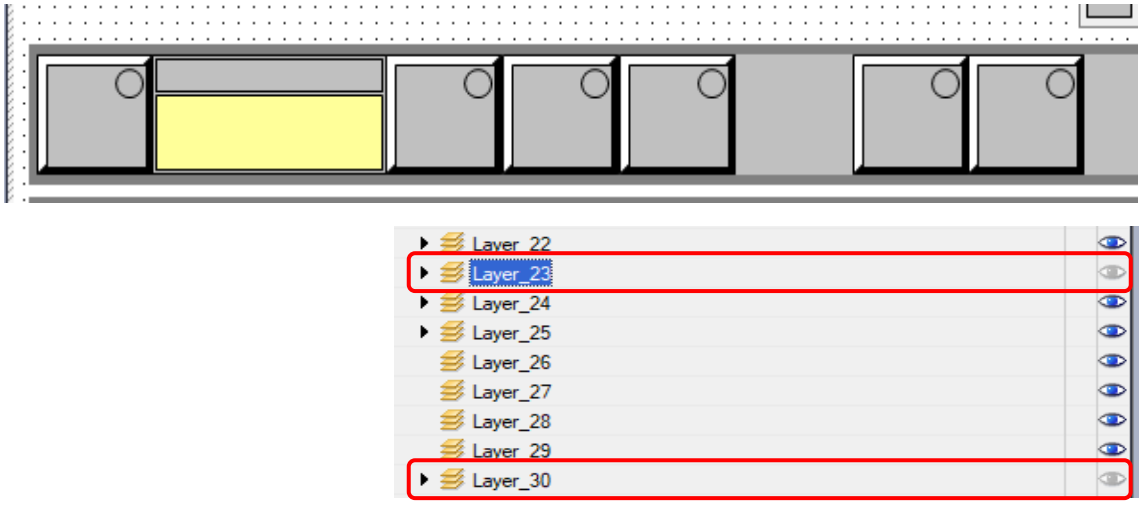


Fig. 6.29 Detailed screen with hiding of the layers 23 (texts) and 30 (buttons)

The configuration of the objects in the specified layers should always be adhered to.

- Layer 00: Change dwVisuValues (pointer instrument)
- Layer 10: Outer frame
- Layer 11: Inner frame
- Layer 20: Frame element rectangle of an operating button (Dws)
  End position at the back (rectangle)
- Layer 21: Frame element polygon of an operating button (Dws)
- Layer 22: Animated areas
- Layer 23: Animated text
- Layer 25: Operator control enables on the button
- Layer 30: Buttons

The specification as to the layer in which the object is displayed, is defined via the object properties.

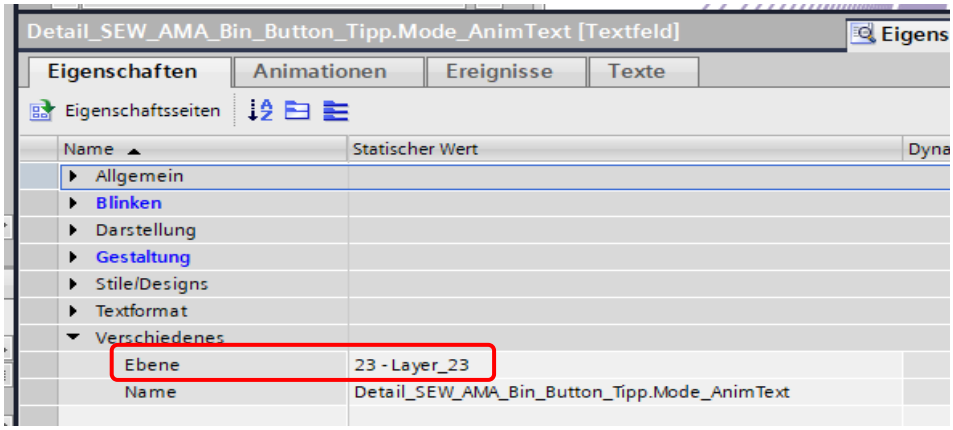
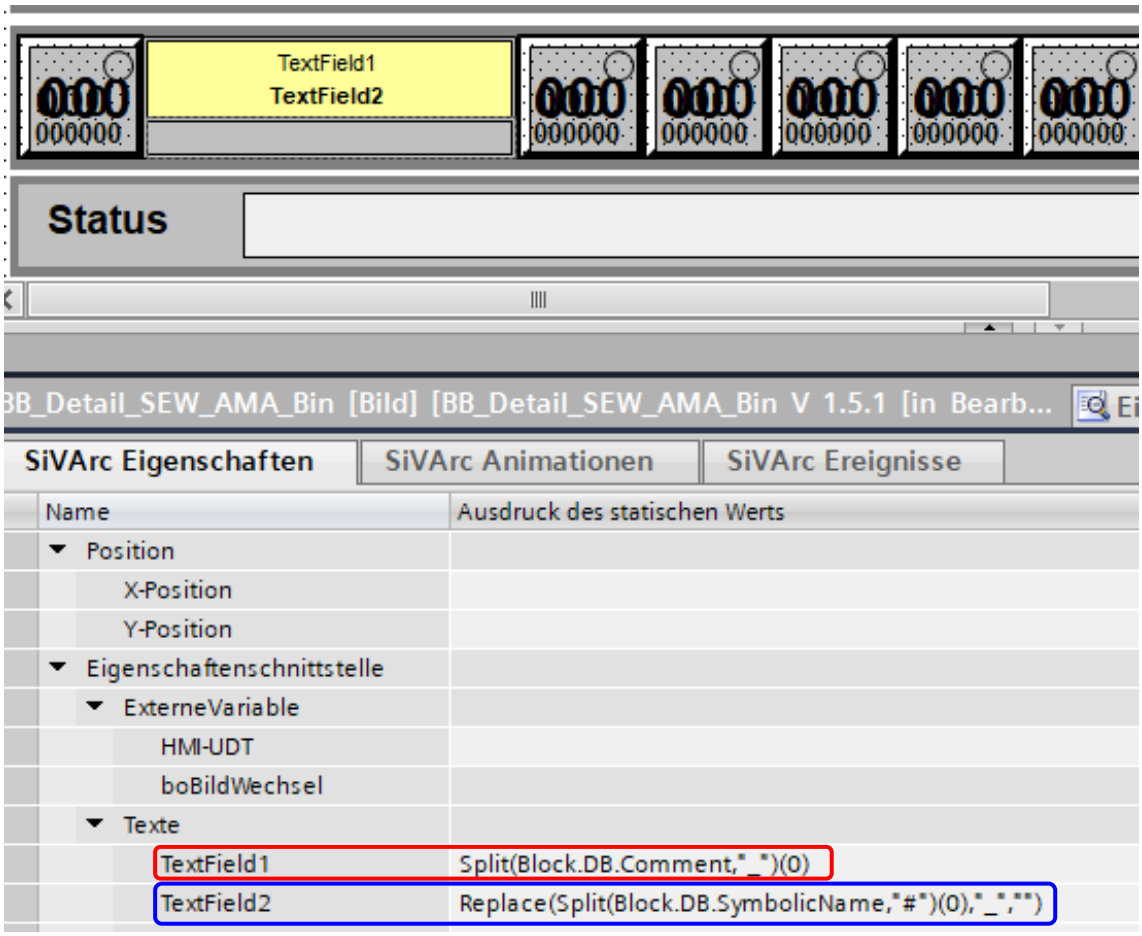


Fig. 6.30 Setting of layer using a text field as an example

The remaining layers can be used for further meaningful structuring of faceplates.

6.1.3.3.4 Supply text fields



- "TextField1" comes from the symbol comment
- "TextField2" comes from the symbol name

Fig. 6.31 Supply of text fields in the faceplate for detailed screen

6.1.3.3.5 HMI UDT

For a new detailed screen, the faceplate needs an HMI-UDT which must be generated for the module FB.

The HMI-UDT is used for data exchange between the controller and the visualization.

The sequence and the data type of the individual tags within this HMI-UDT must correspond to the associated module component.

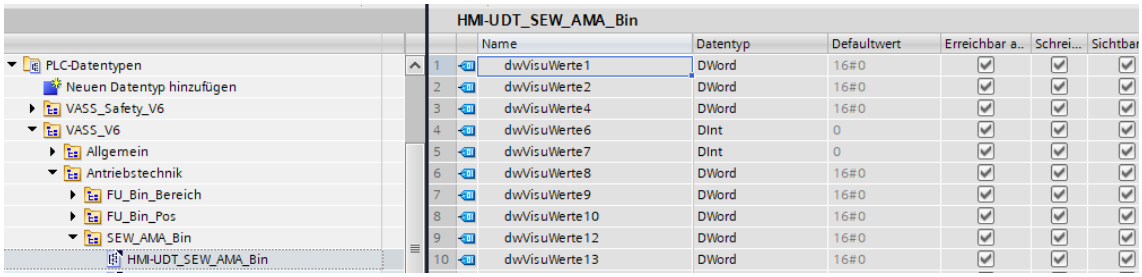


Fig. 6.32 Example declaration of the visualization interface "HMI-UDT_SEW_AMA_Bin"

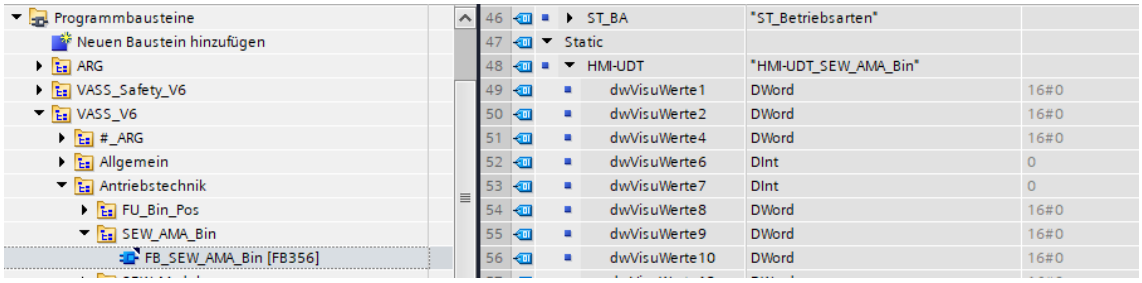


Fig. 6.33 Example assignment of the "HMI-UDT_SEW_AMA_Bin" in the module FB

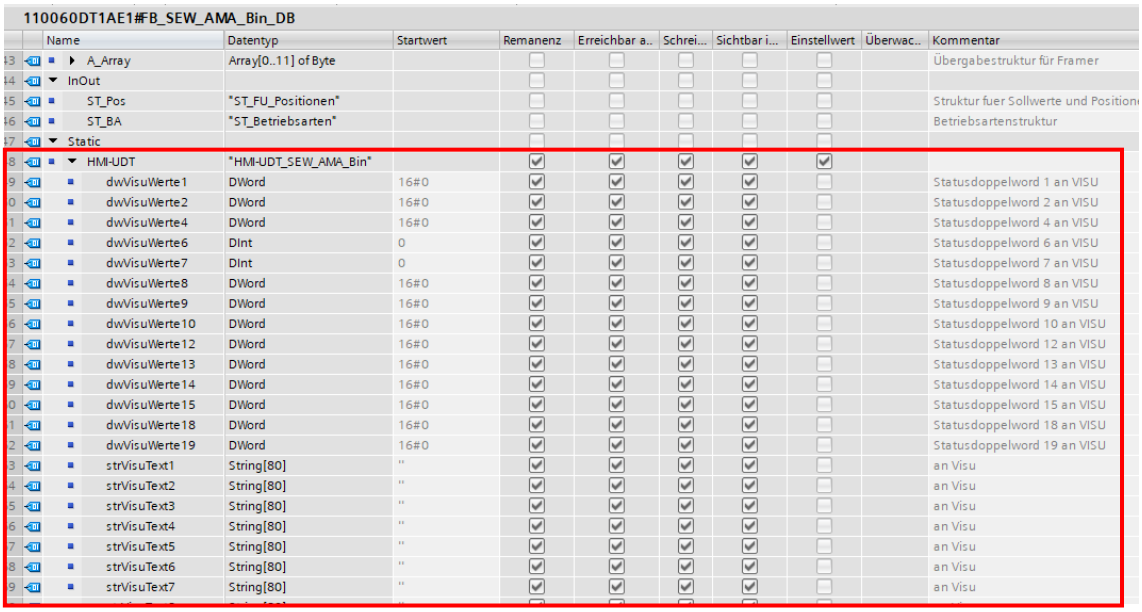


Fig. 6.34 Representation of visualization interface in the instance DB

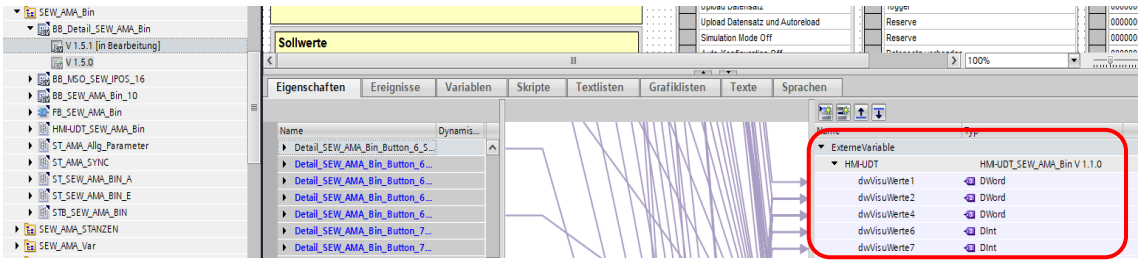


Fig. 6.35 Example of the associated HMI-UDT for the faceplate "BB_Detail_SEW_AMA_Bin"

6.1.3.4 Tags for the detailed screen

For each faceplate of a detailed screen, the internal tags that are switched from the external tags to the internal tags with the "Move and mask" function must be created in the faceplate (in this case, for example, "BB_Detail_SEW_AMA_Bin").

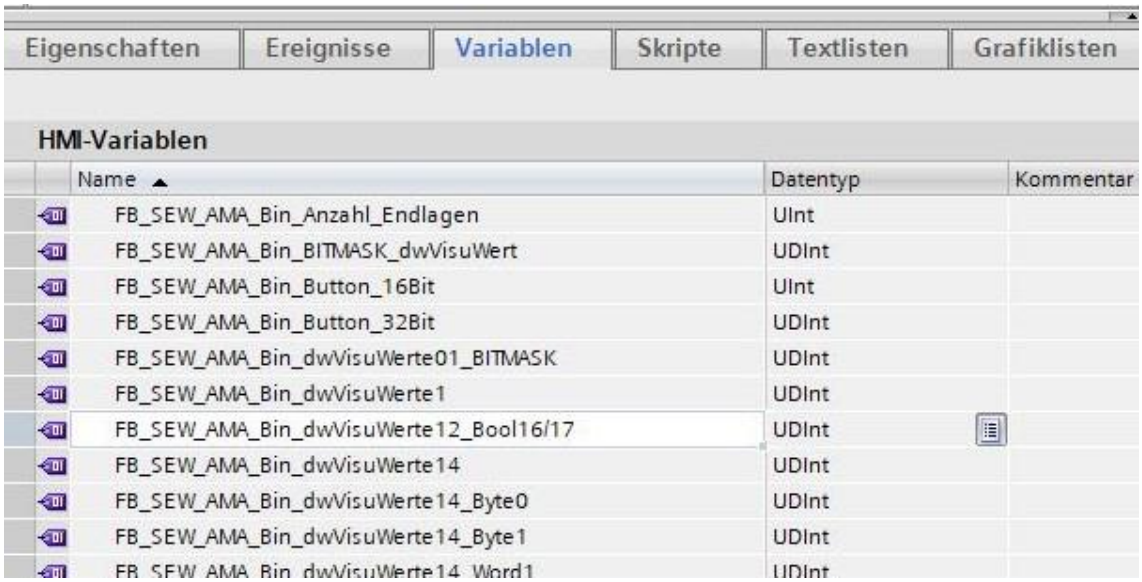


Fig. 6.36 Example of internal tag directory

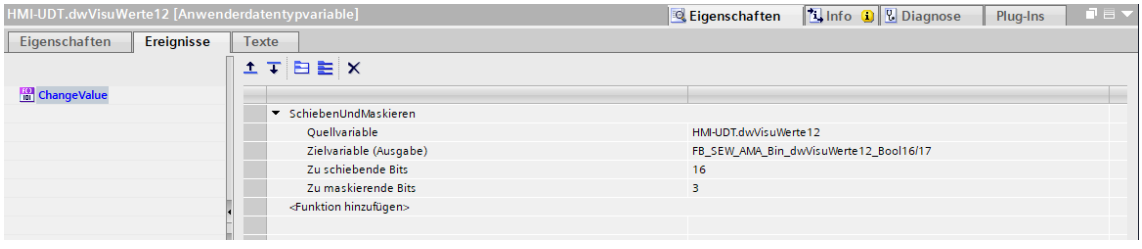


Fig. 6.37 Switching dwVisuWerte to internal tags

6.1.3.5 Faceplate for calling the detailed screen

For every detailed screen there is a faceplate which is placed by the SiVArc and via which the detailed screen is called.

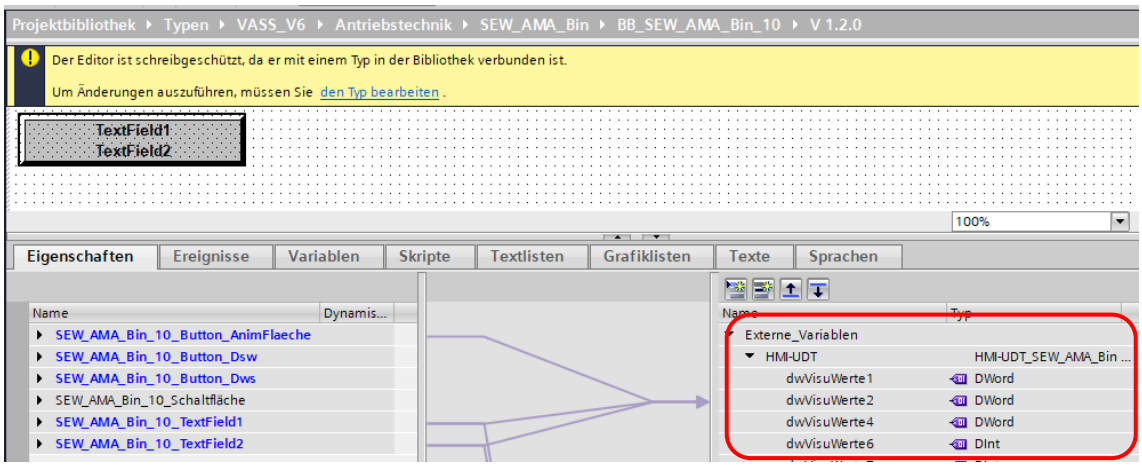


Fig. 6.38 Faceplate properties "BB_SEW_AMA_Bin_10" for the root screen

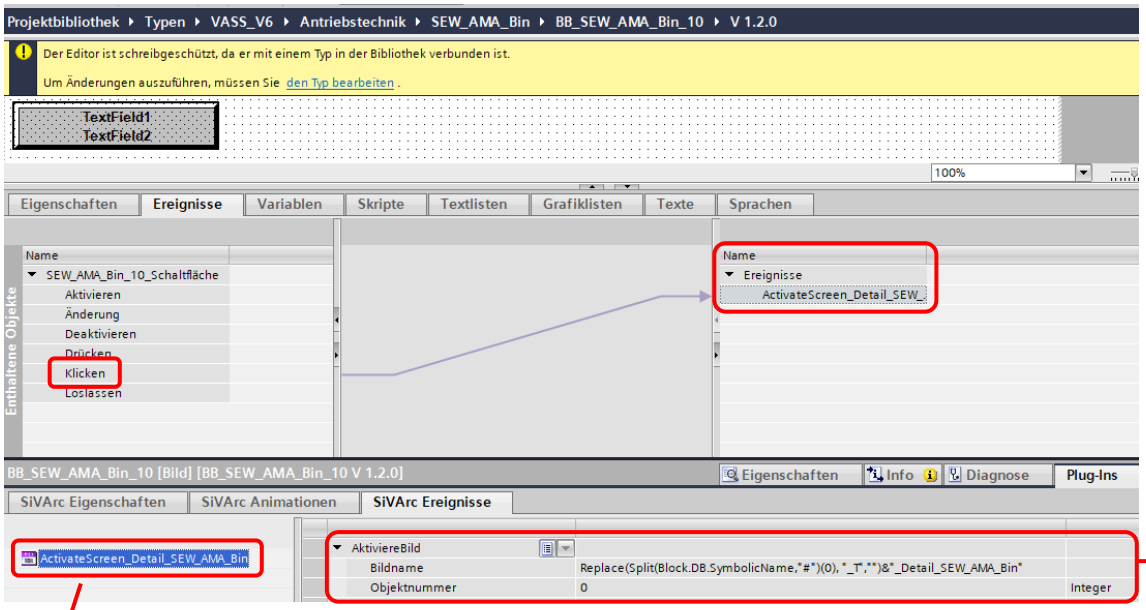


Fig. 6.39 Events faceplate "BB_SEW_AMA_Bin_10" for calling the detailed screen from the root screen

Event for the call of the detailed screen

Name of the detailed screen determined by the SiVArc

Example:

"FB_DrehtFU_2S" called as an instance in the PLC in a station with the drive type depending on the keyword "Antrieb" "SEW_AMA_BIN" as instance.

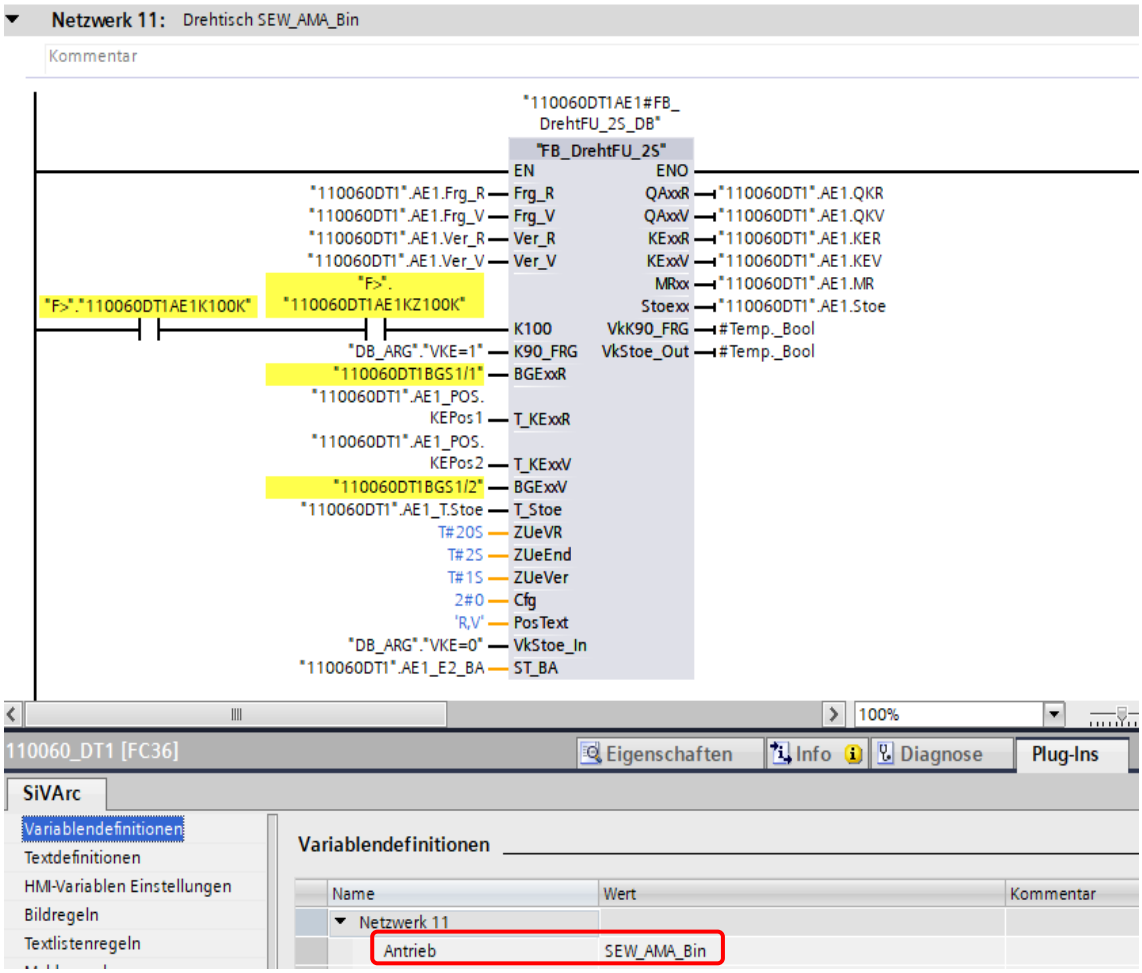


Fig. 6.40 Events faceplate "FB_DrehtFU_2S"

Example:

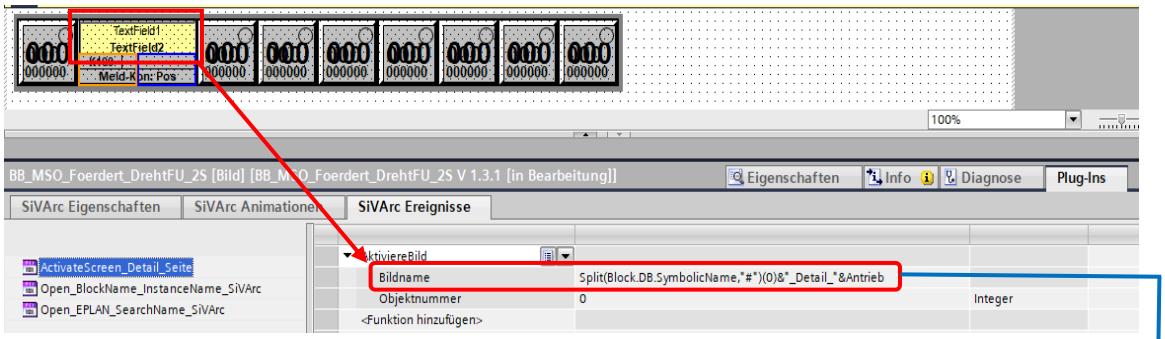


Fig. 6.41 "FB_DrehtFU_2S" with the SiVArc event "ActivateScreen"

The screen name is determined from the symbols of the instance DB (the first entry, "SymbolicName") and "_Detail_" is added as a fixed text, the value of the network comment comes from the Plug-ins.

Here as an example: Screen name of the detailed screen '110060DT1AE1_Detail_SEW_AMA_Bin"

Name of the instance DB | fixed text | Drive type via keyword

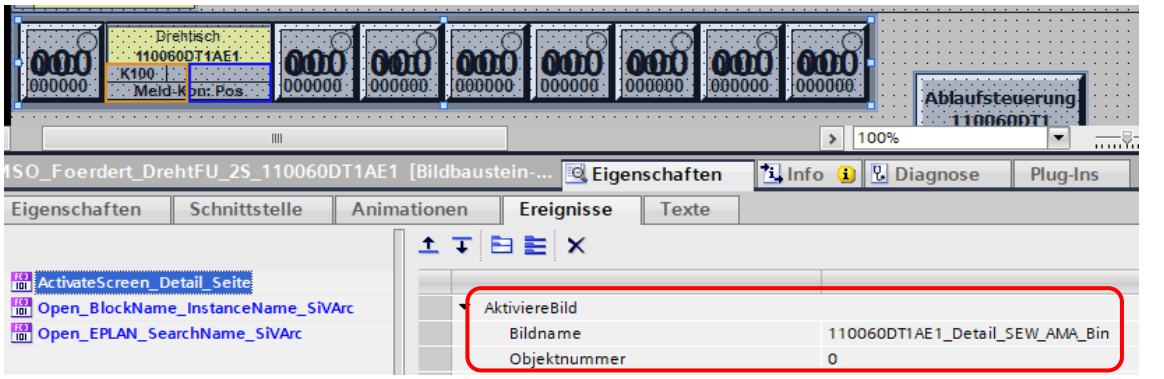


Fig. 6.42 The faceplate for "FB_DrehtFU_2S" is placed in the station screen

Note

When a new faceplate is created for a detailed screen, the keyword of an existing technology group must be used if the new module also corresponds to this technology. Otherwise a new keyword must be created.

Keywords are not translated because they are a fixed element of the SiVArc Plug-Ins properties.

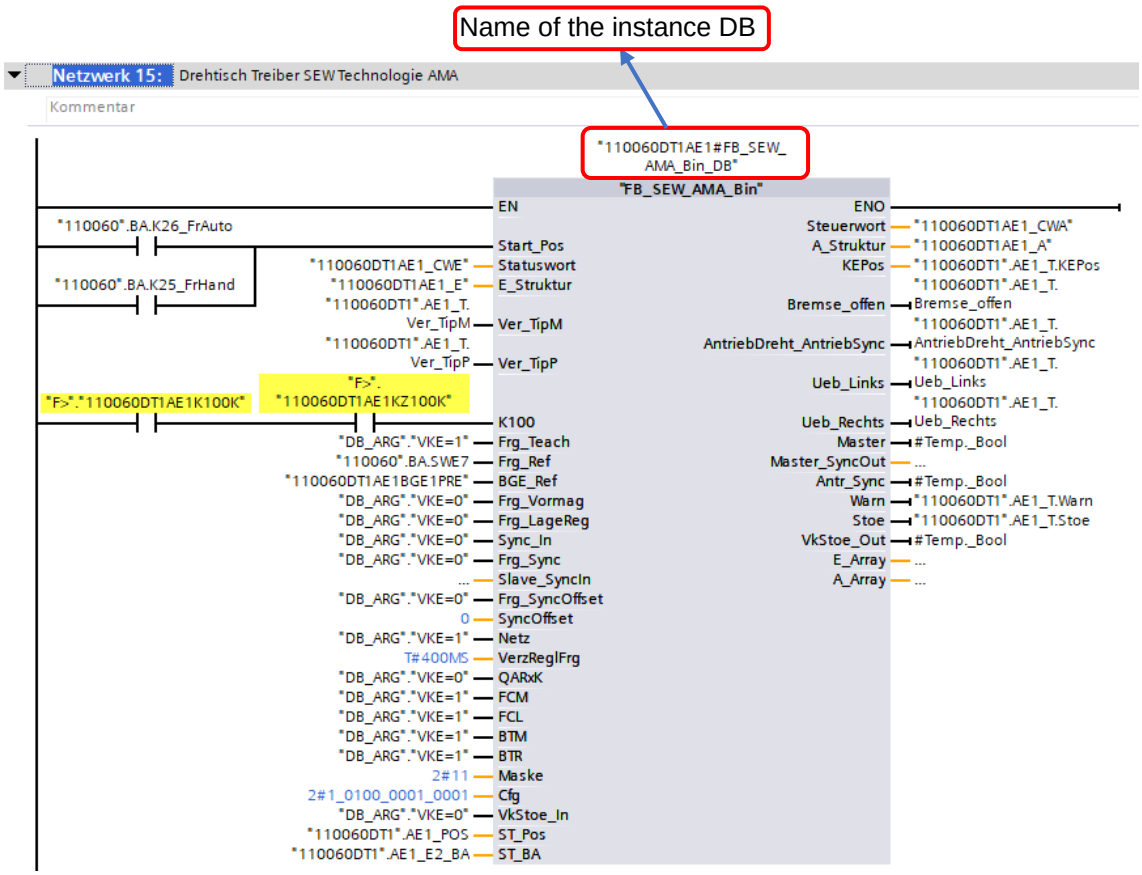


Fig. 6.43 The call of the drive "FB_SEW_AMA_Bin" as an instance in the PLC in the station

Calling the "FB_SEW_AMA_Bin" in the PLC logic places a button on the root screen via which the detailed screen can be called.

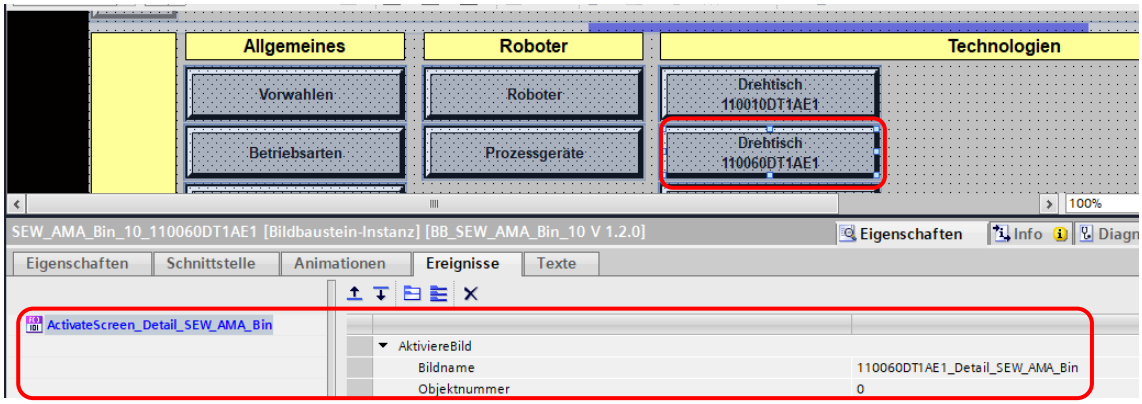


Fig. 6.44 The faceplate "SEW_AMA_Bin_10" is placed on the root screen by SiVArc

Calling the "FB_SEW_AMA_Bin" in the PLC logic generates the associated detailed screen by the SiVArc.

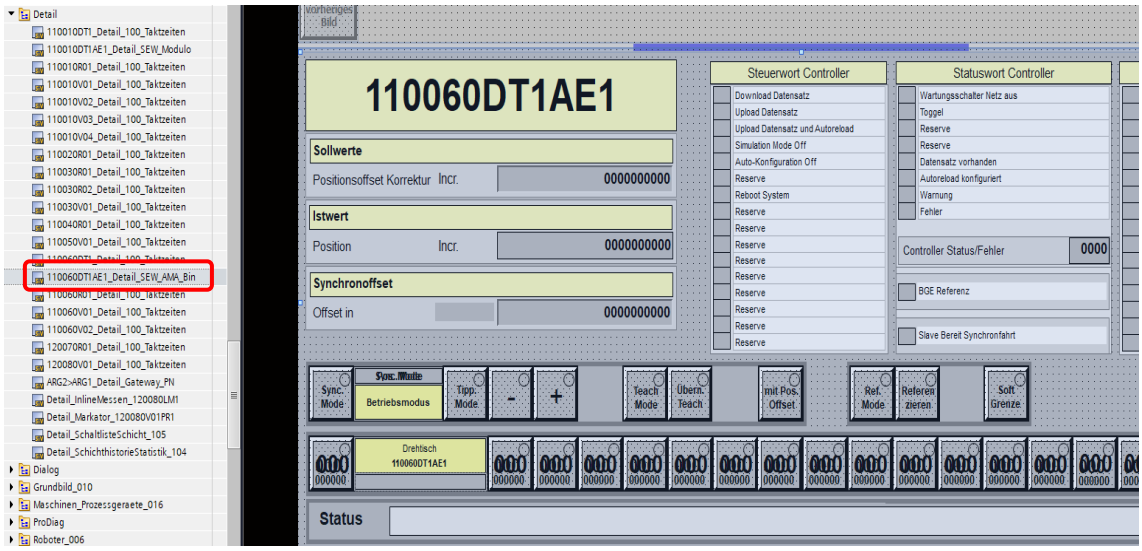


Fig. 6.45 The detailed screen "110060DT1AE1_Detail_SEW_AMA_Bin" is generated in the folder "Detail" by SiVArc

6.1.4 Creation of a faceplate for a detailed screen (Technology)

For the faceplate "BB_Detail_InlineMessen", the layout has the name "Layout_106_Detail_InlineMessen_19Z".

The layouts are in the project library in the "Master copies" folder.

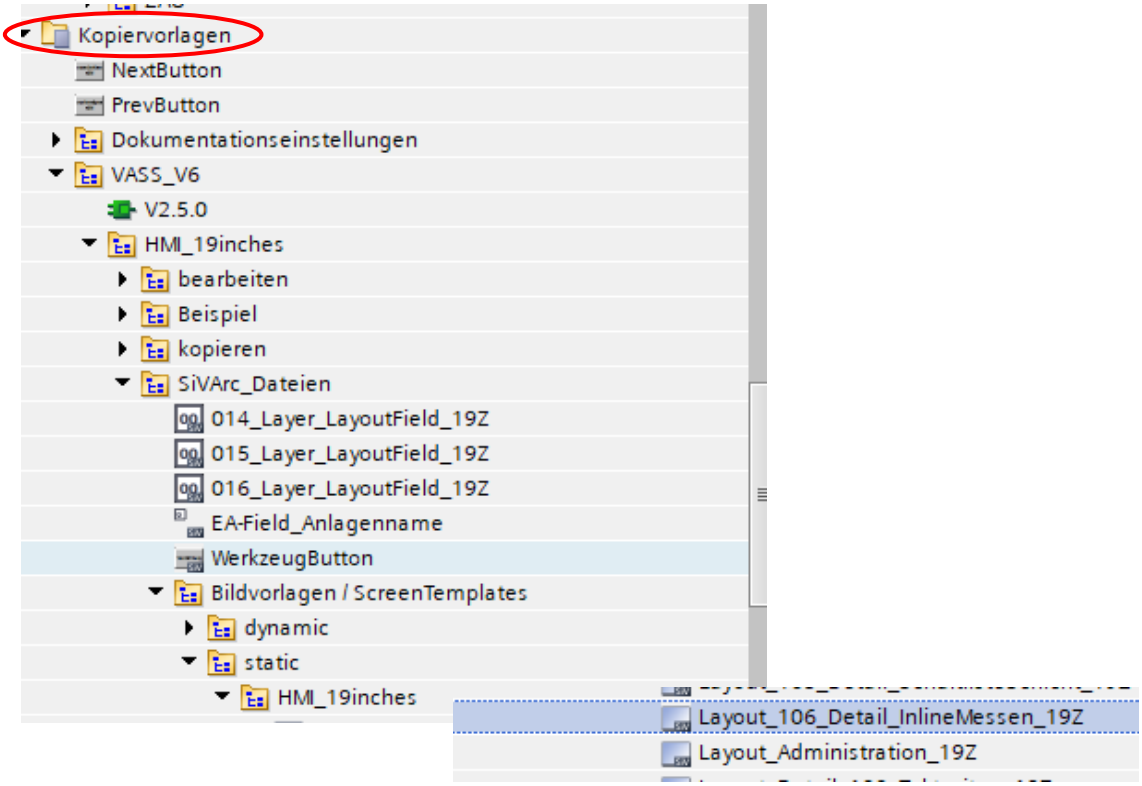


Fig. 6.46 Master copies folder in the project library

This shows the SiVArc properties in "Layout_106_Detail_InlineMessen", how the name of the screen is formed and in which screen group it is generated as a screen.

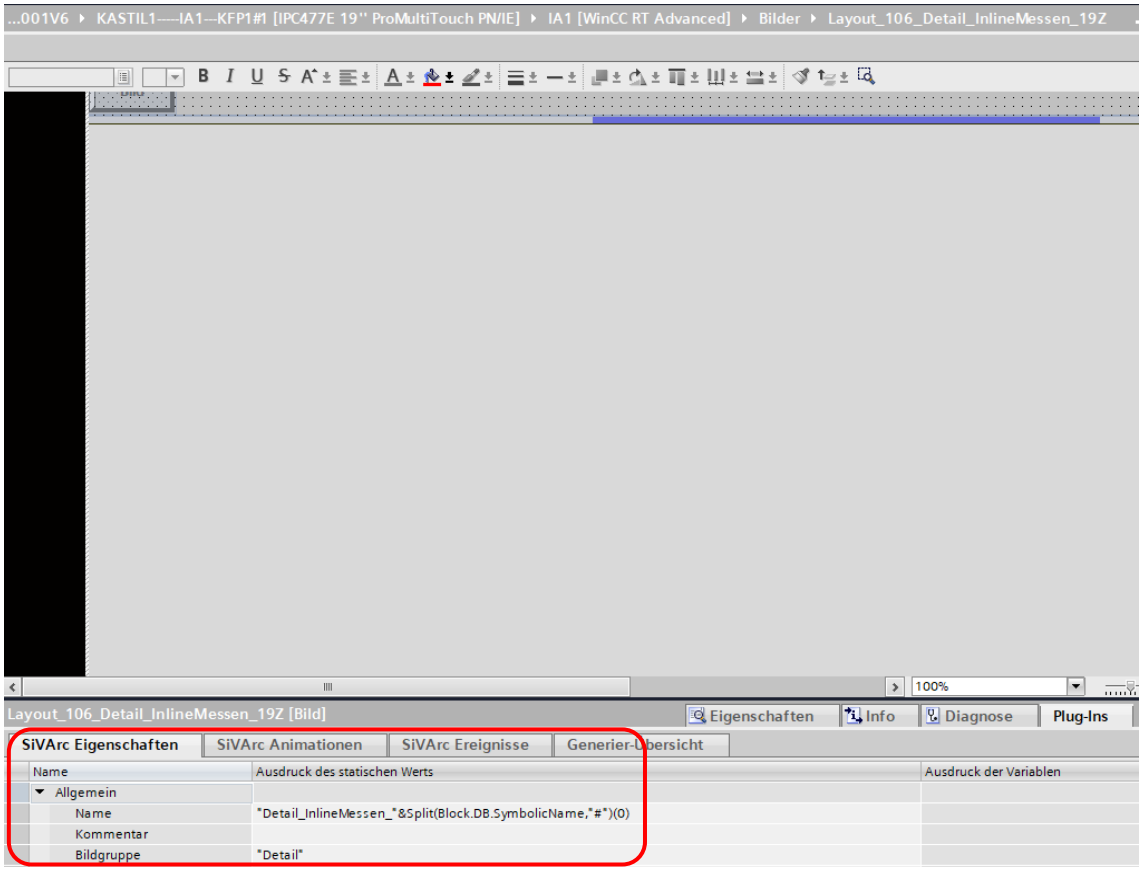


Fig. 6.47 SiVArc properties in "Layout_106_Detail_InlineMessen"

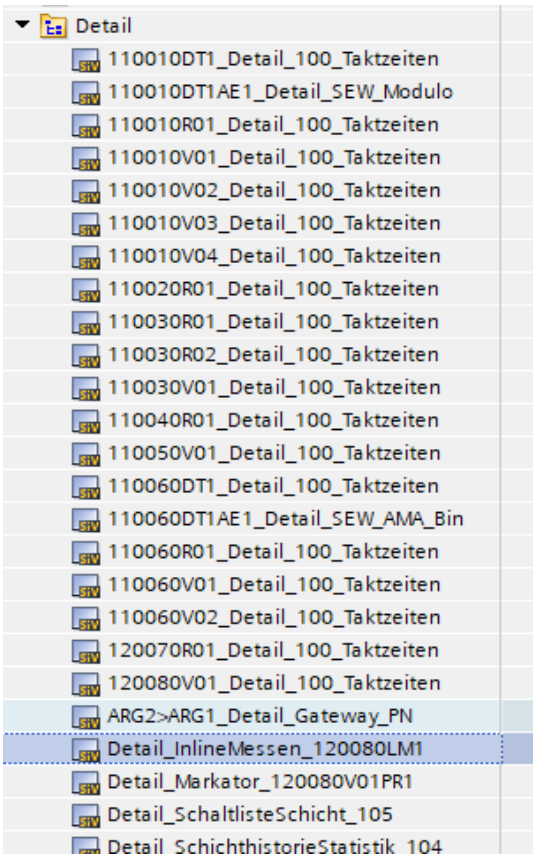


Fig. 6.48 Detailed screen "Detail_InlineMessen_120080LM1" after generation

6.1.4.1 Faceplate for calling the detailed screen

For every detailed screen there is a faceplate which is placed by the SiVArc and via which the detailed screen is called.

Name of the detailed screen determined by the SiVArc.

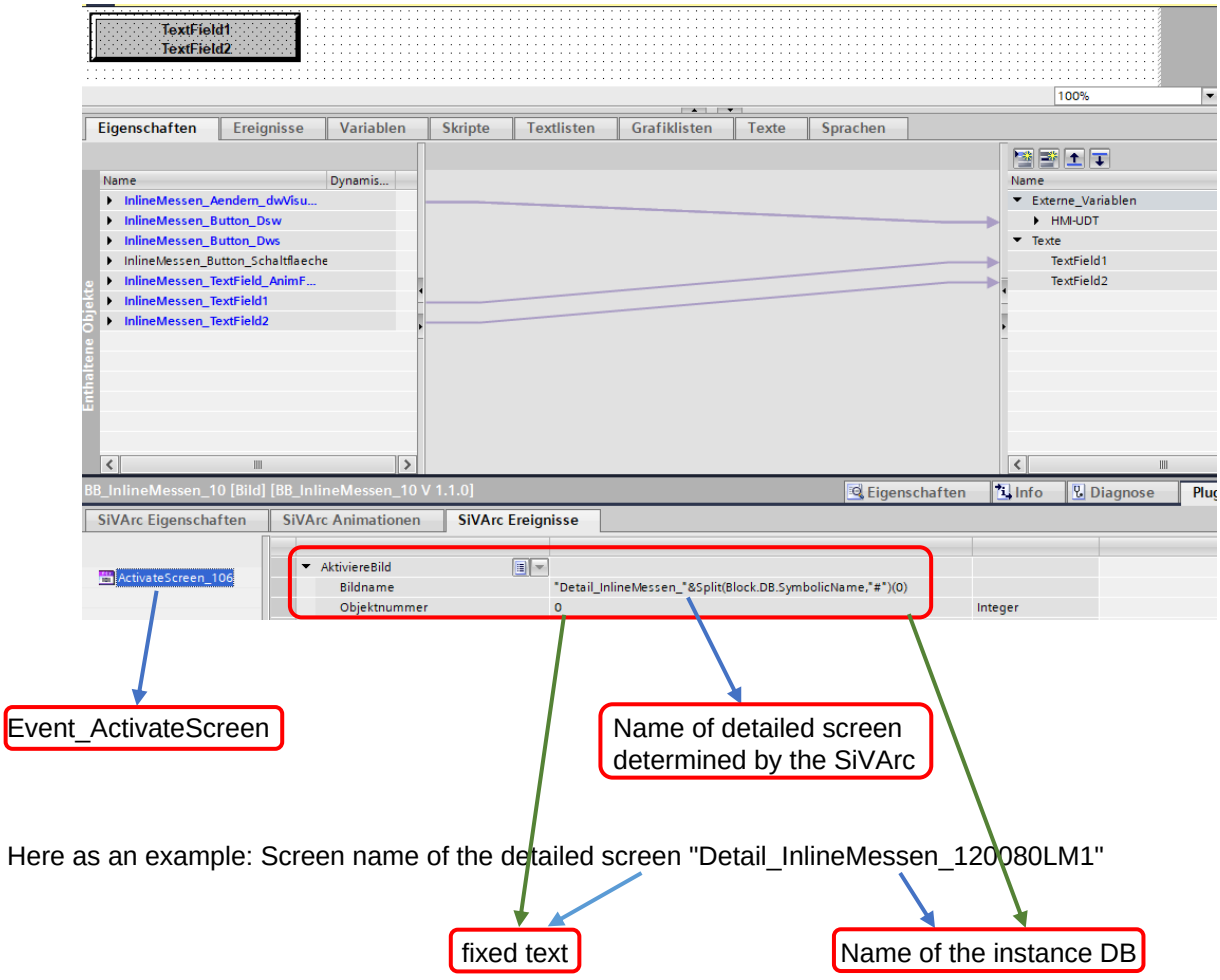


Fig. 6.49 Faceplate properties "BB_InlineMessen_10" for the root screen

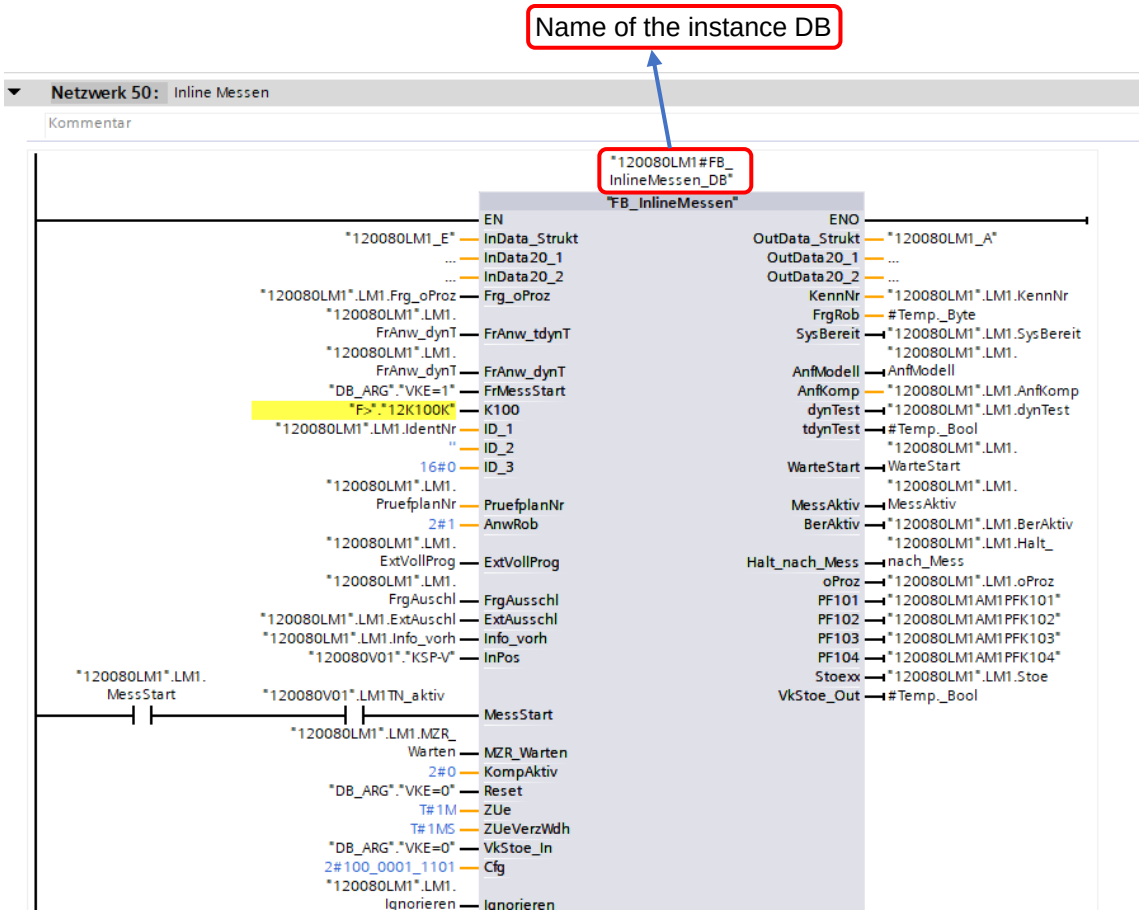


Fig. 6.50 Calling "FB_InlineMessen" as an instance in a station

Detailed screens for technologies, without different characteristics, as is the case for drives, do not have a keyword.

The names of the detailed screen are generated by the SiVArc by a fixed text, e.g. "Detail_InlineMessen_" and the first entry of the instance DB name (up to #) e.g. "120080LM1" of the instance DB for "Detail_InlineMessen_120080LM1".

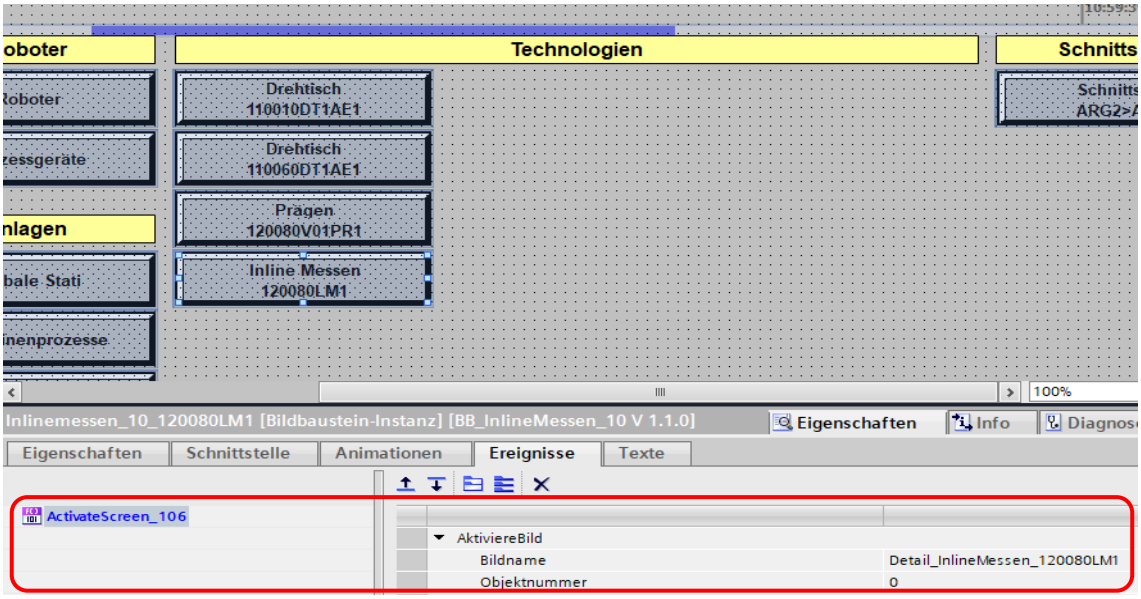


Fig. 6.51 Placement of "BB_InlineMessen_10" on the root screen

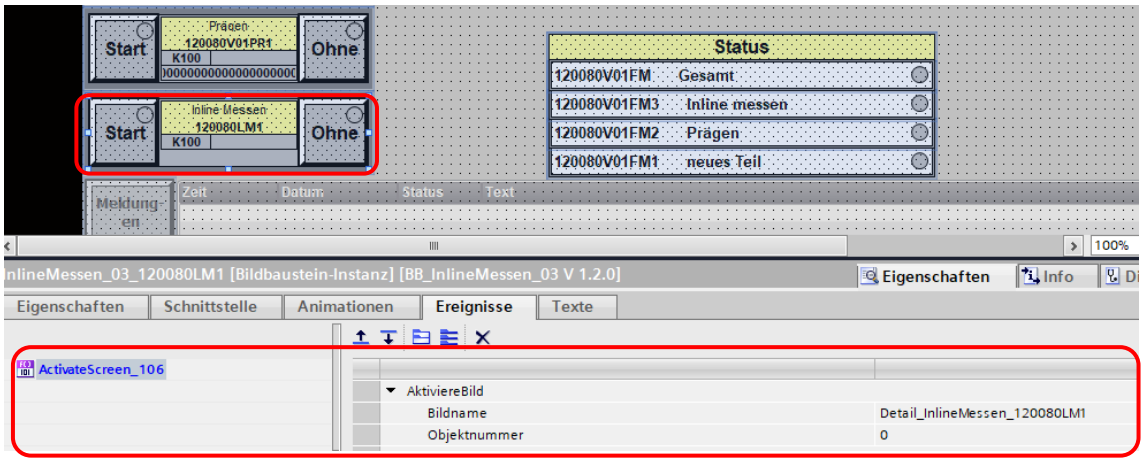


Fig. 6.52 Placement of "BB_InlineMessen_03" on the station screen

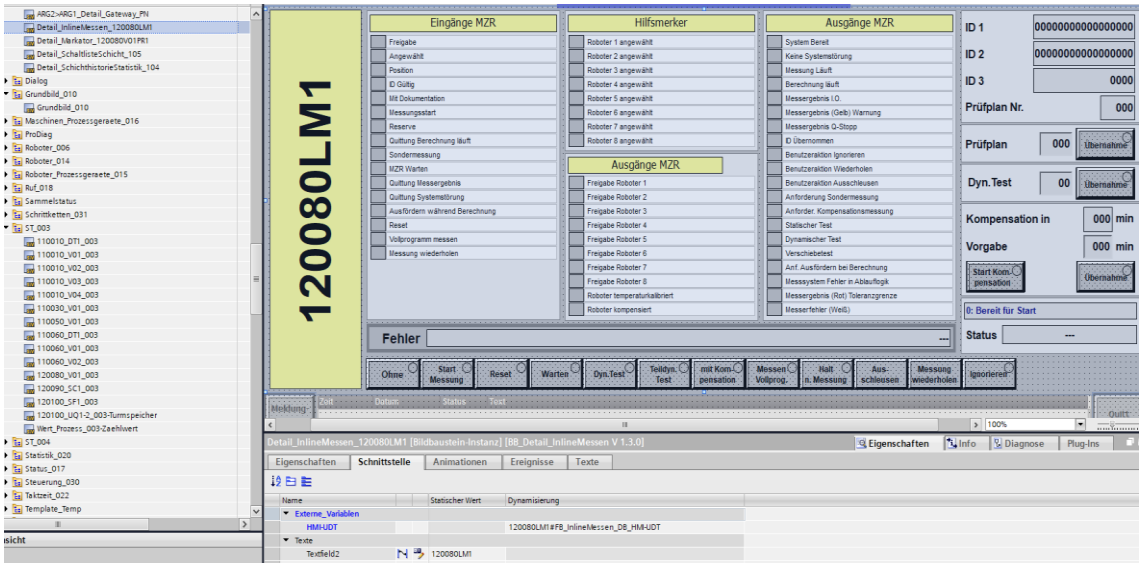


Fig. 6.53 Detailed screen "Detail_InlineMessen_120080LM1"

6.2 VASS library

The storage location for the faceplates is the folder "VASS_V6". Each new faceplate belongs to one of the technology groups.

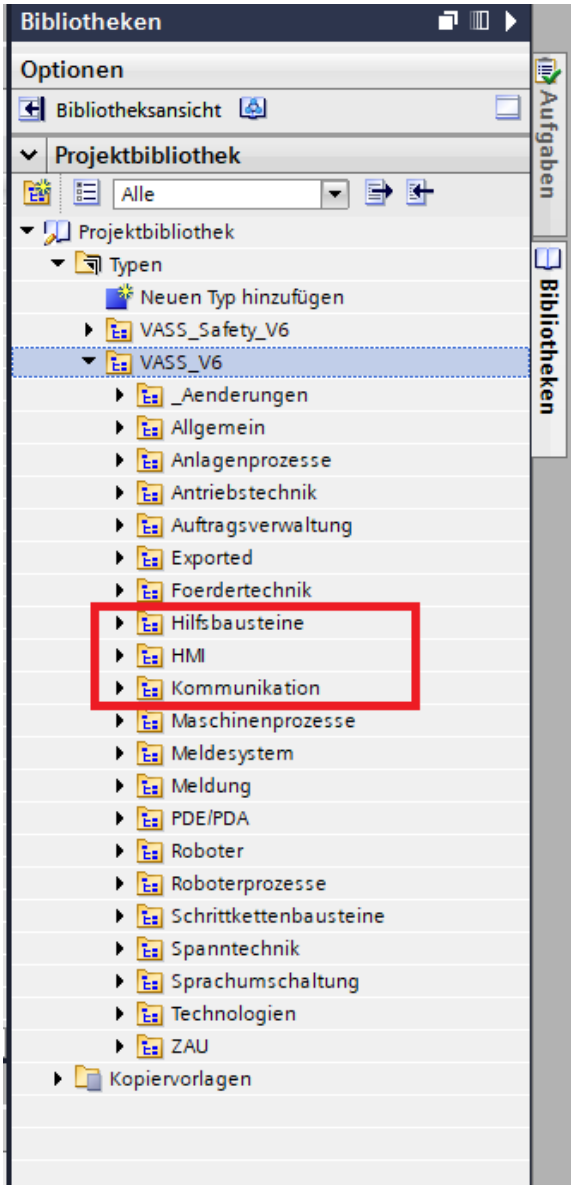


Fig. 6.54 Technology groups of the storage location for the faceplates

Note

Faceplates do not belong in the technology group marked in red (tool blocks, HMI and communication).

6.3 Faceplate texts and multilingualism

6.3.1 Type-specific texts

- These texts are stored within the faceplate
- are identical for all instances of a faceplate
- can be defined in several languages
- are edited in the faceplate editor
- are created and modified only manually

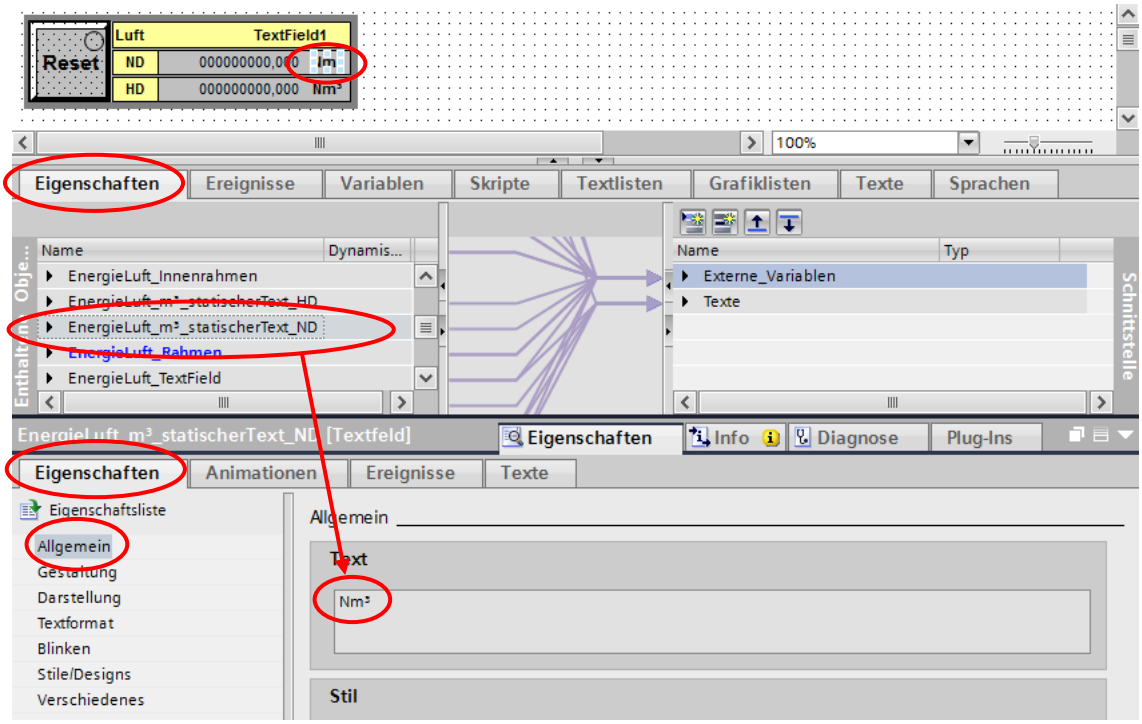


Fig. 6.55 Type-specific texts

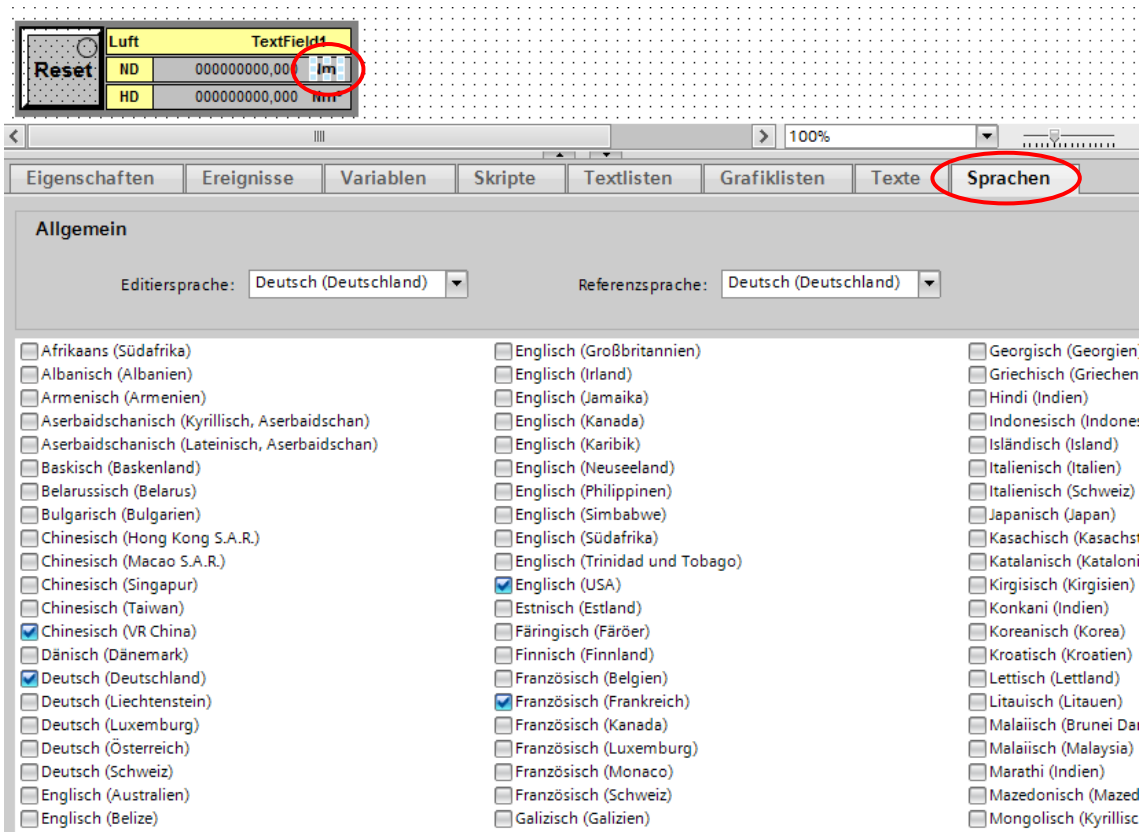


Fig. 6.56 Select languages

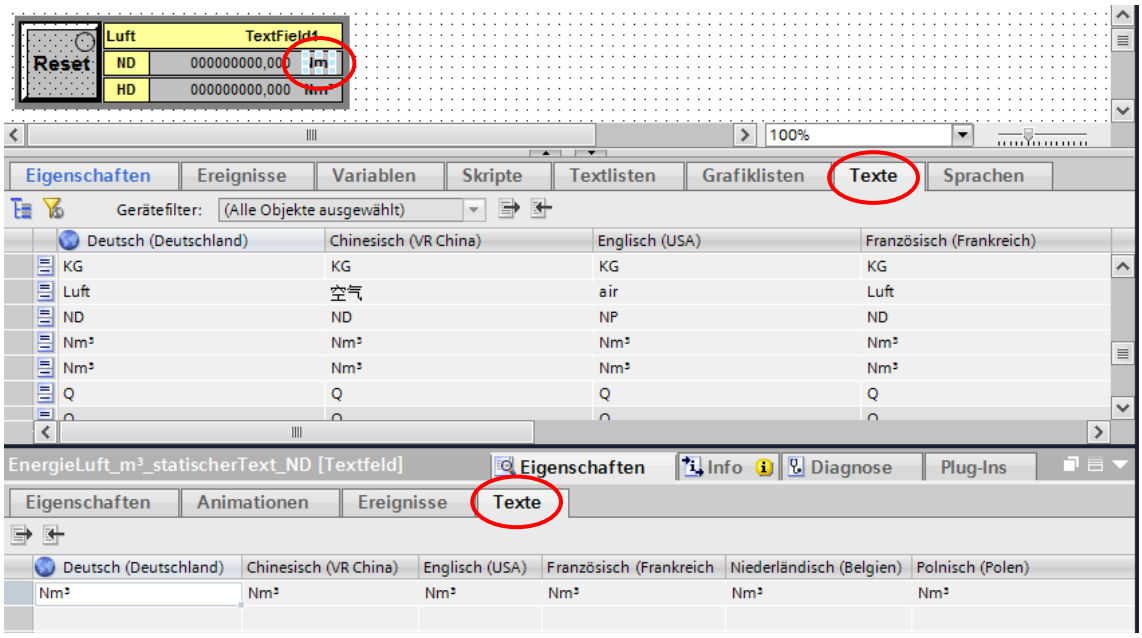


Fig. 6.57 Select languages

6.3.2 Instance-specific texts

- These texts are stored at the faceplate interface
- may be different per instance of a faceplate
- can be defined in several languages
- are edited in the faceplate editor and under SiVArc properties (BIock may not be marked, you have to click in the "empty field")
- are automatically generated by SiVArc.

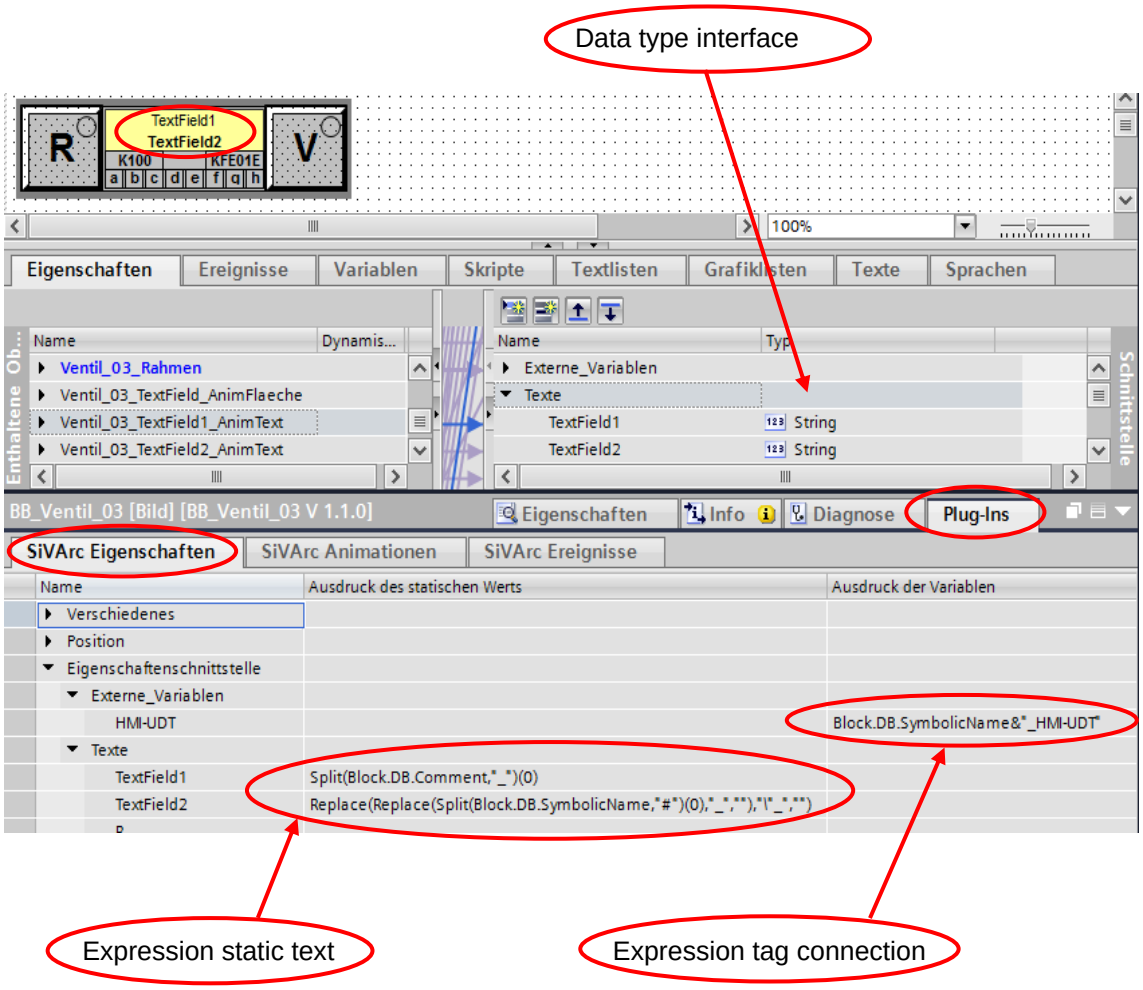


Fig. 6.58 Instance-specific texts

The name of the interface, e.g. "TextField1", must be entered in the interface properties under texts for all languages.

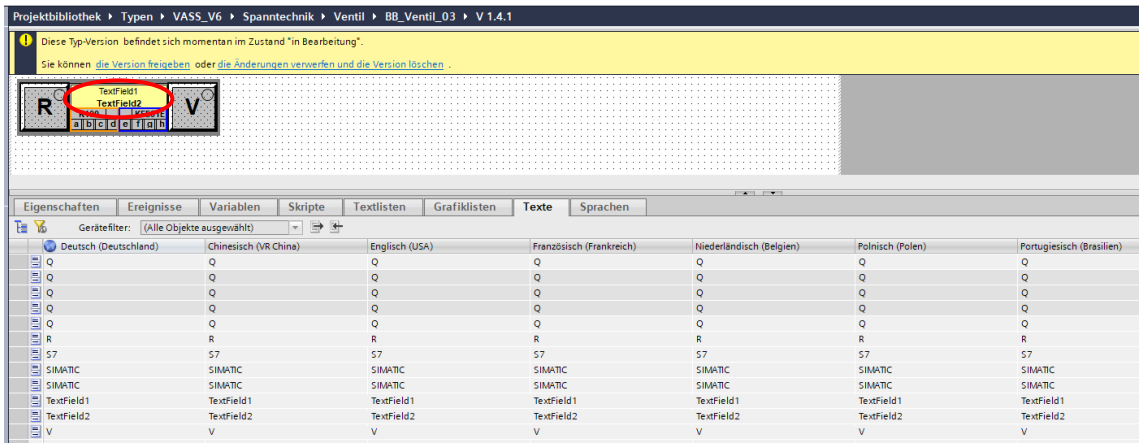


Fig. 6.59 Multilingual texts

6.4 SiVArc rules

6.4.1 Plug-Ins SiVArc properties in faceplates

6.4.1.1 Faceplate BB_Ventil_03 name - HMI-UDT and texts

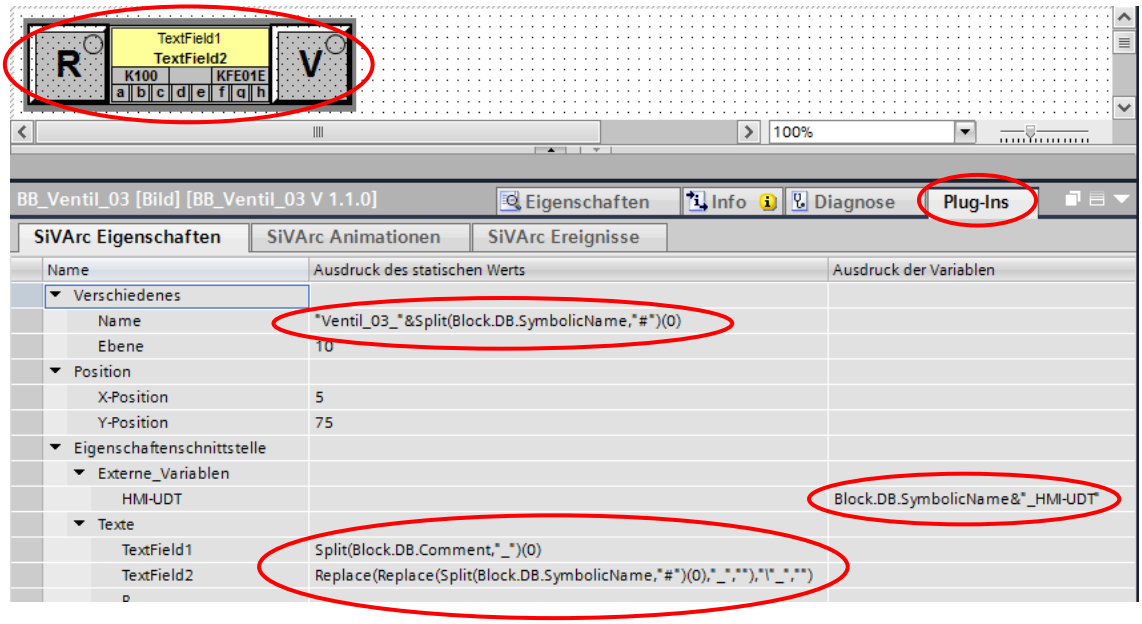


Fig. 6.60 Faceplate BB_Ventil_03 Name - HMI-UDT and texts

6.4.1.2 Faceplate BB_Ventil_04_A - Name, HMI-UDT and texts

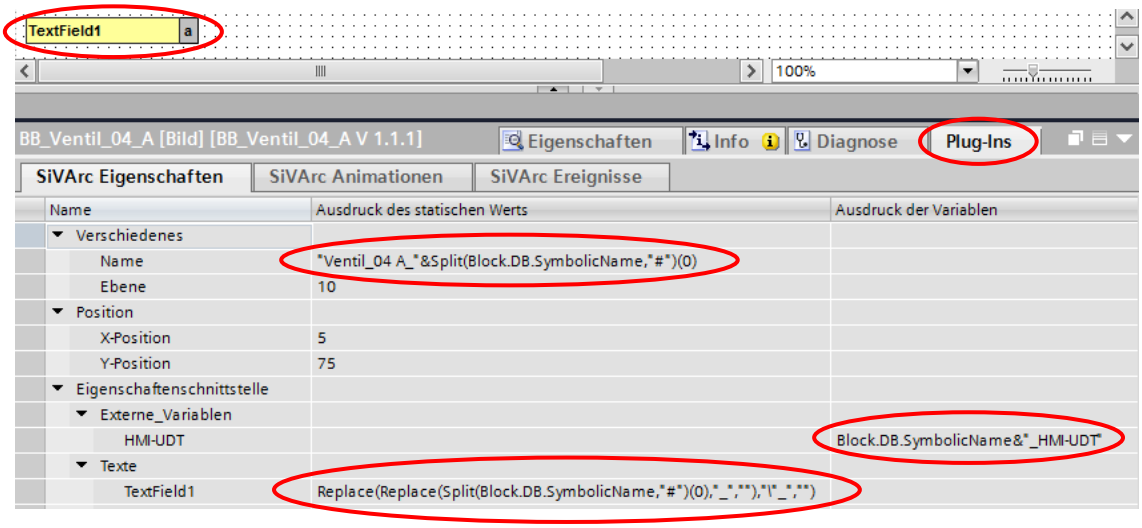


Fig. 6.61 Faceplate BB_Ventil_04_A - name, HMI-UDT and texts

6.4.2 SiVArc tag rules

6.4.2.1 Definition of tag rules tag group hierarchy with "SiVArc_"

Fig. 6.62 Definition of tag rules tag group hierarchy with "SiVArc_"

Warning

Tag rules are fixed and must not be changed.

6.4.2.2 Generated tag names with "_HMI-UDT" and data type HMI-UDT

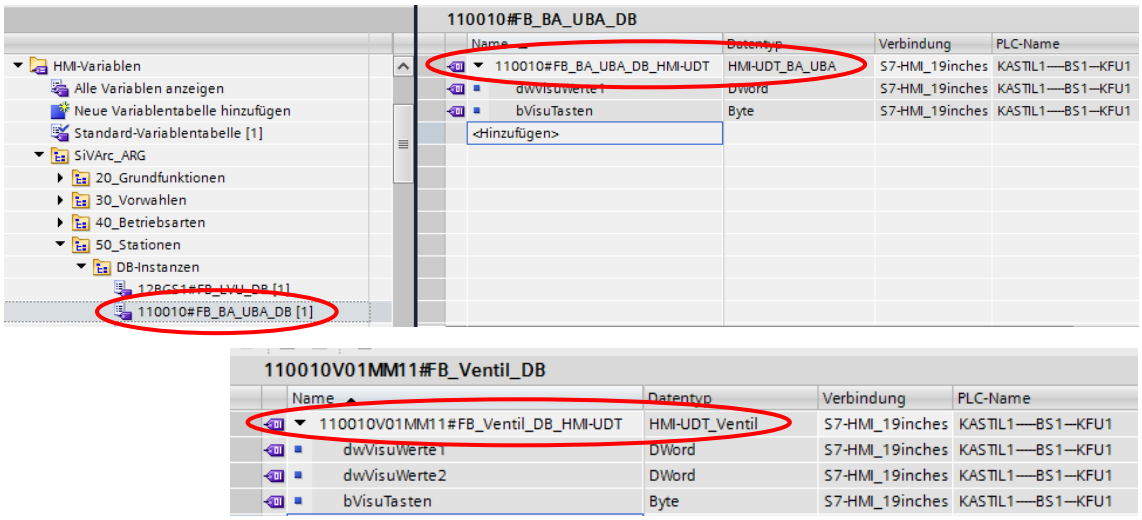


Fig. 6.63 Generated tag names with"_HMI-UDT" and data type HMI-UDT

6.4.2.3 Faceplate interface automatically supplied with data type HMI-UDT

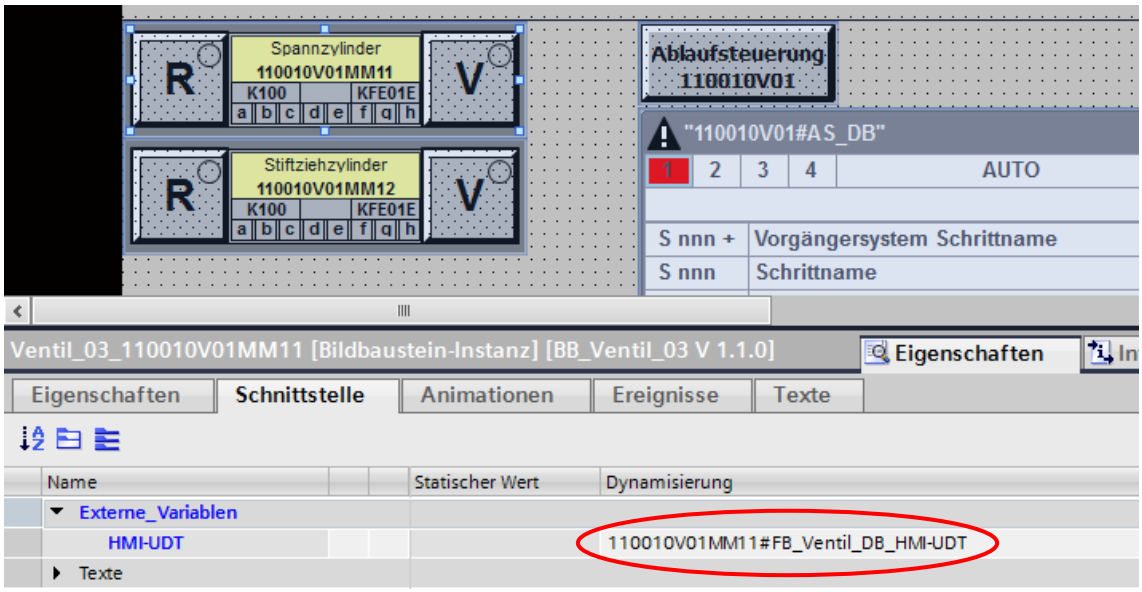


Fig. 6.64 Faceplate interface automatically supplied with data type HMI-UDT

6.4.3 SiVArc Plug-Ins

Warning

The layouts are fixed and must not be changed.

6.4.3.1 Plug-Ins SiVArc properties in screen templates - Screen 003

The figure is meant to show how the name of the "003 pages" is generated.

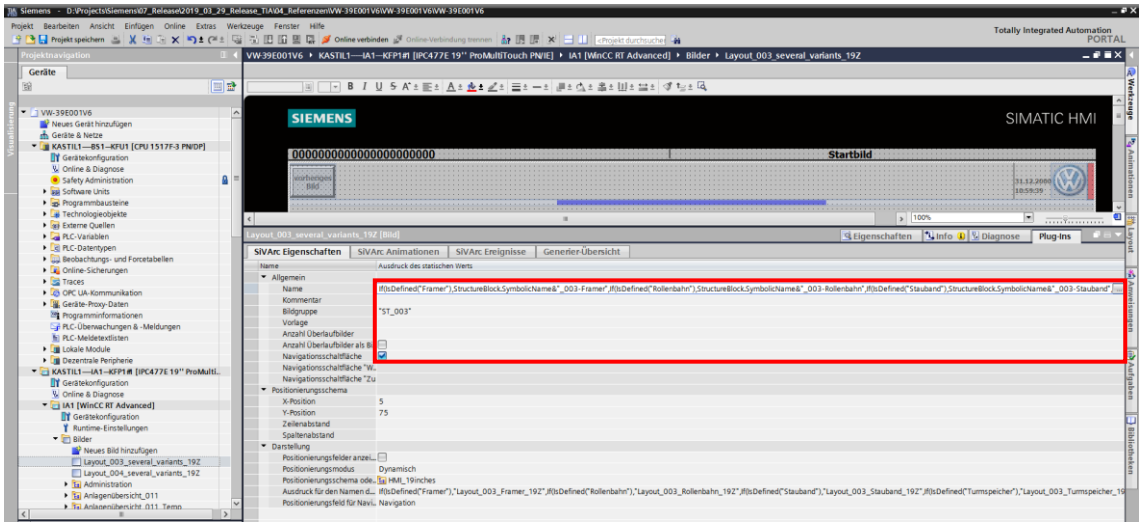


Fig. 6.65 Plug-Ins SiVArc properties in screen templates - Screen 003

6.4.3.2 Plug-Ins SiVArc properties in screen templates - Screen 004

The figure is meant to show how the name of the "004 pages" is generated.

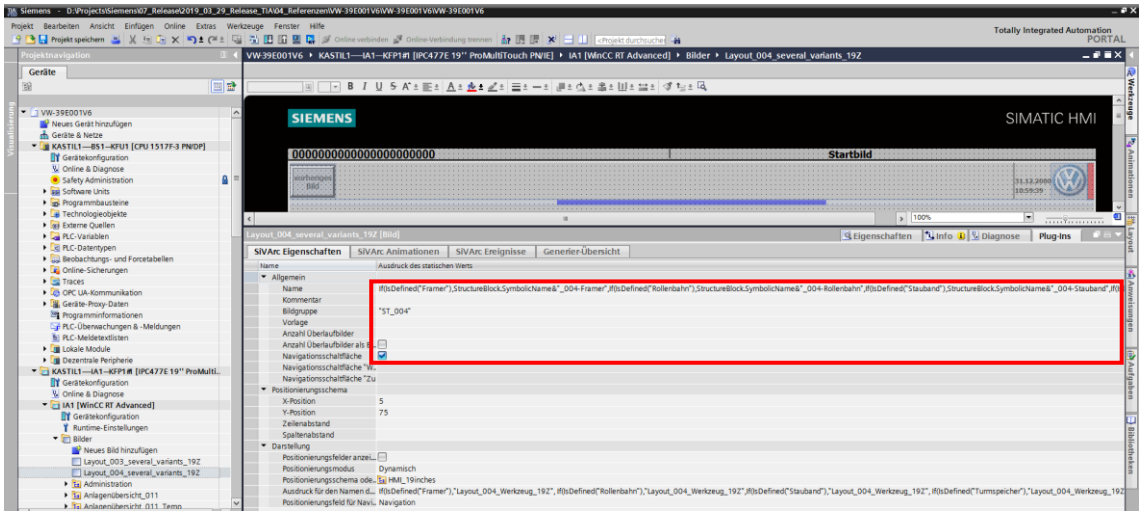


Fig. 6.66 Plug-Ins SiVArc properties in screen templates - Screen 004

6.4.3.3 Use of the parameter "Maske" or "Cfg"

If a technology comprises several subfunctions for which there are also different faceplates, the placement of the faceplates can be controlled via the parameters "Maske" or "Cfg". These parameters must be stored in the screen rules.

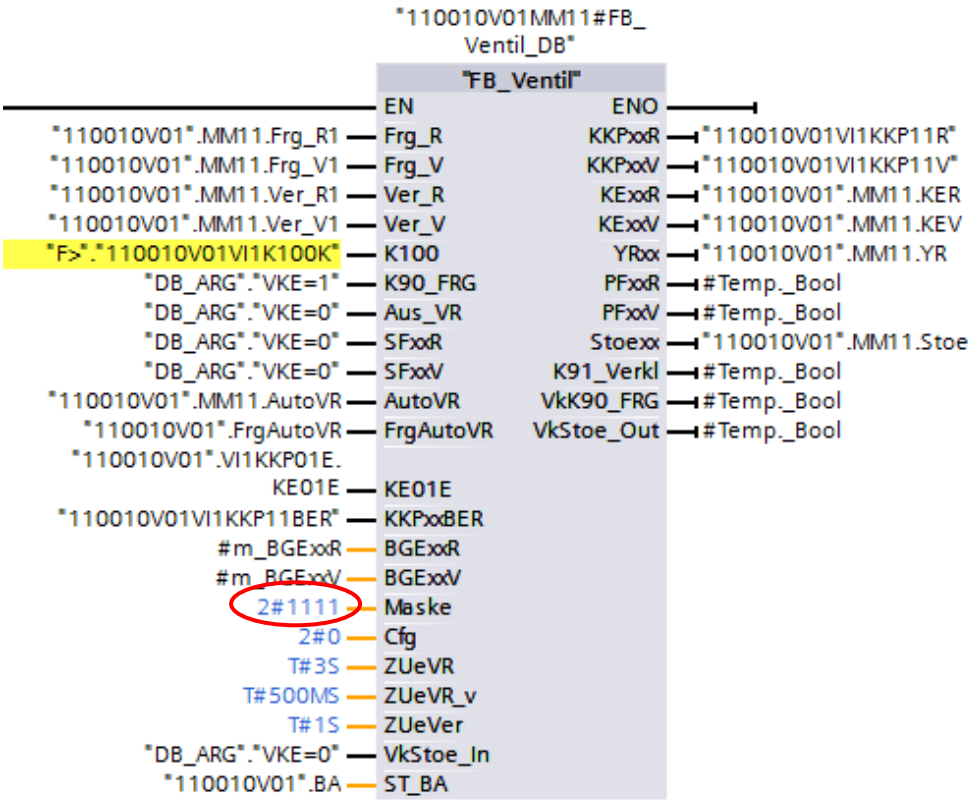


Fig. 6.67 Parameter assignment at the FB_Ventil parameter "Maske"

With this assignment of the parameter "Maske" at the FB_Ventil, the end positions "A, B, C, D" are placed on the tool screen (screen 04) after SiVArc generation.

Bit0 = End position_A

Bit1 = End position_B

Bit2 = End position_C

Bit3 = End position_D

Bit4 = End position_E

Bit5 = End position_F

Bit6 = End position_G

Bit7 = End position_H

The system for a screen rule with the parameter "Maske" or "Cfg" is illustrated using the following example:

Conditions for faceplate Ventil_04 with cylinder end positions A - H:

((Block.Parameters("Maske").Assigned And Block.Parameters("Maske").Value) And **2#1) = 1**
cylinder end position A

((Block.Parameters("Maske").Assigned And Block.Parameters("Maske").Value) And **2#10) = 2**
cylinder end position B

etc.

The value of the mask is accessed via an additional condition and a faceplate is therefore selected for placement.

These conditions must be entered in the "screen rules" under "Condition".

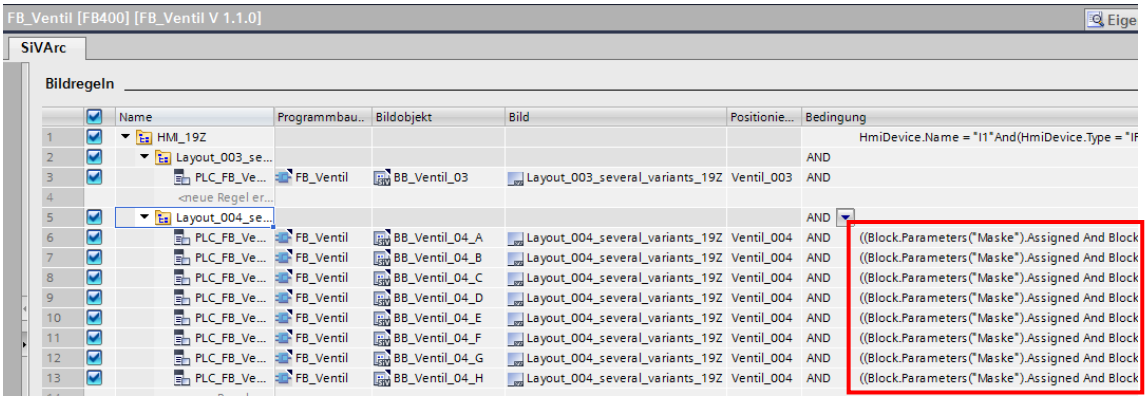


Fig. 6.68 SiVArc screen rules for valve faceplate in screens 003 and 004

SiVArc generated screen 004 filled with faceplates

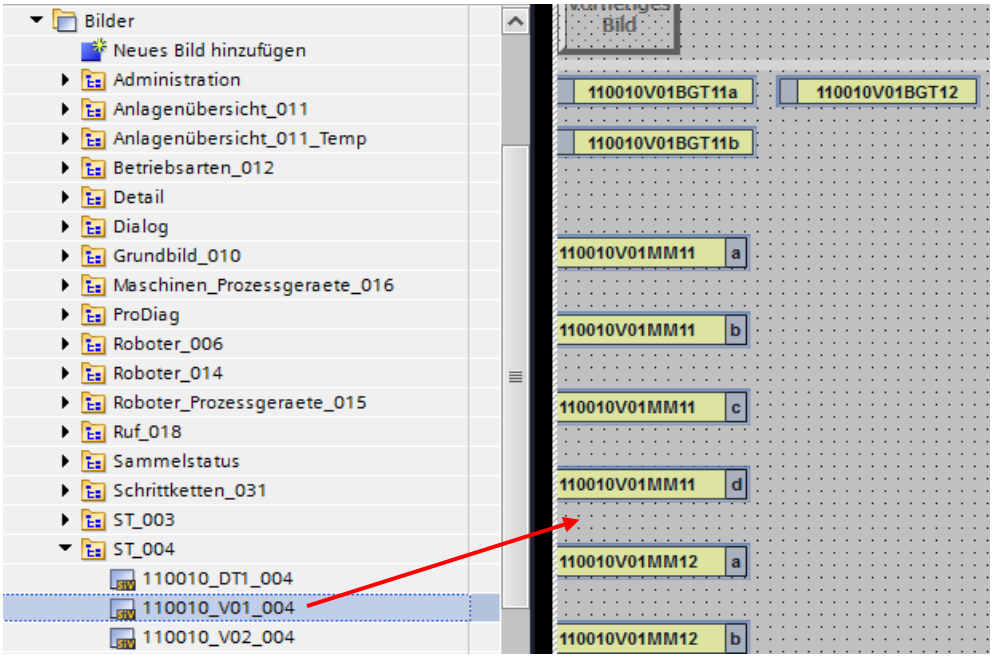


Fig. 6.69 SiVArc generated screen 004 filled with faceplates

6.4.4 SiVArc text lists and text list rules

If screen elements are to be automatically filled with IO comments, text lists must be created.

6.4.4.1 Creation of new SiVArc text lists as master copies

To create a new text list, a new text list must be created and assigned a name in the navigation tree under test and graphic lists.

Entries must then be made under "Plug-Ins → SiVArc properties" for these text list entries.

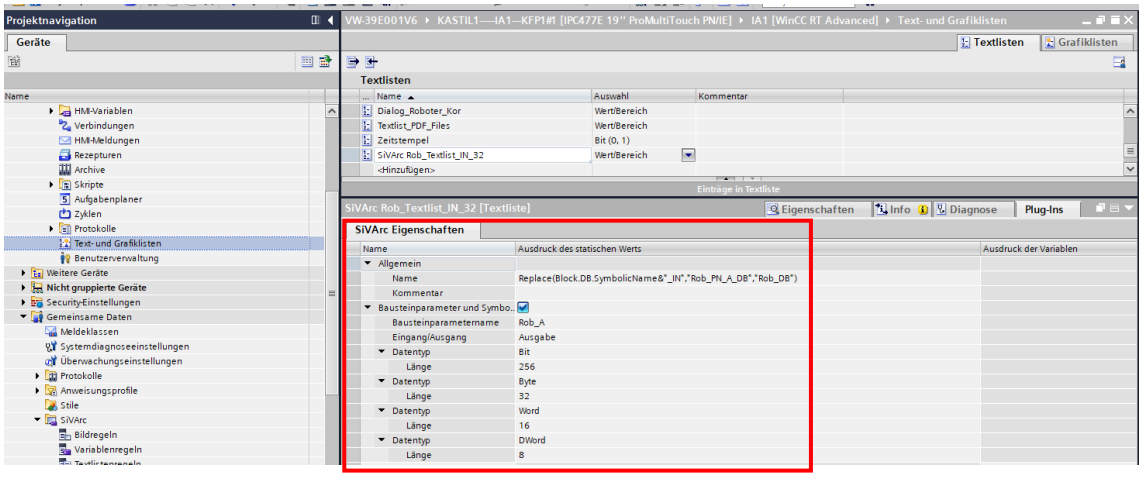


Fig. 6.70 Creation of new SiVArc text lists as master copies

The completed text list must then be stored in the project library under "Master copies → VASS_V6 → HMI_19Inches → SiVArc_Files → Textlisten / Textlists".

Newly created text lists are sorted under Master copies in a separate folder (as specified in Chapter 4 "Specifications for brand and plant-specific standards")

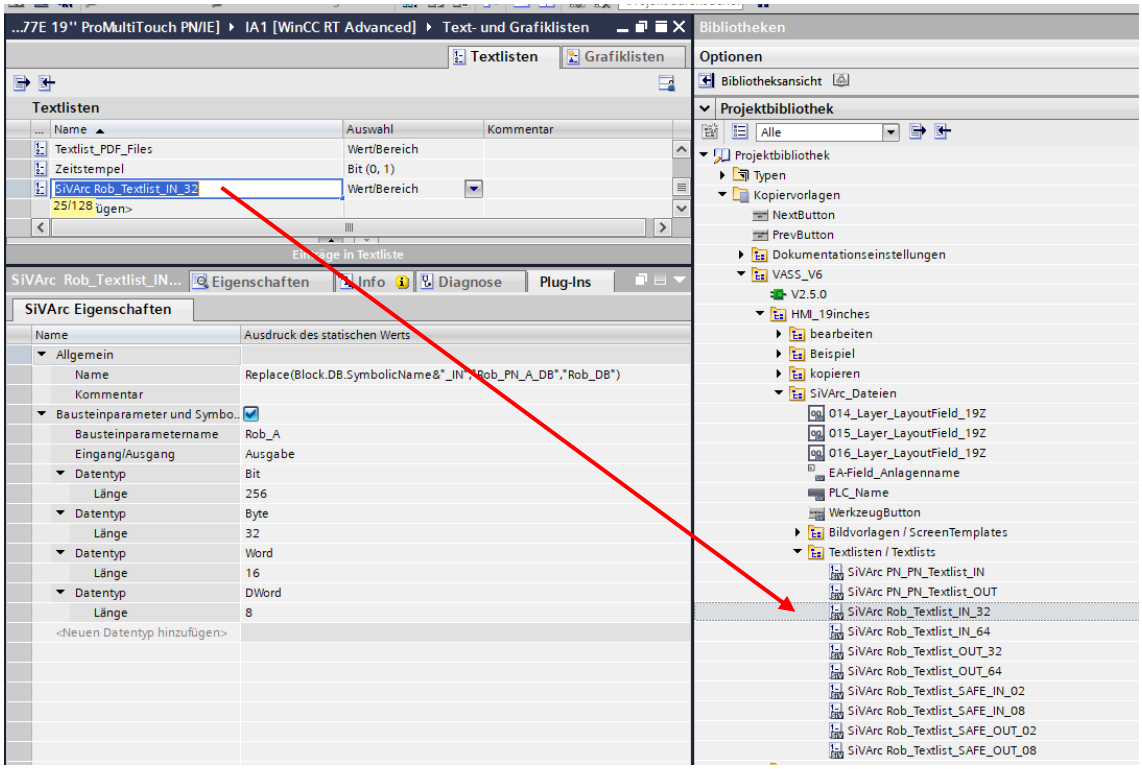


Fig. 6.71 Storage of new text list

When the newly created text list has been copied to the master copies (project library), the text list must be deleted on the left-hand side in the project.

Only then is it possible to create text list rules for the SiVArc.

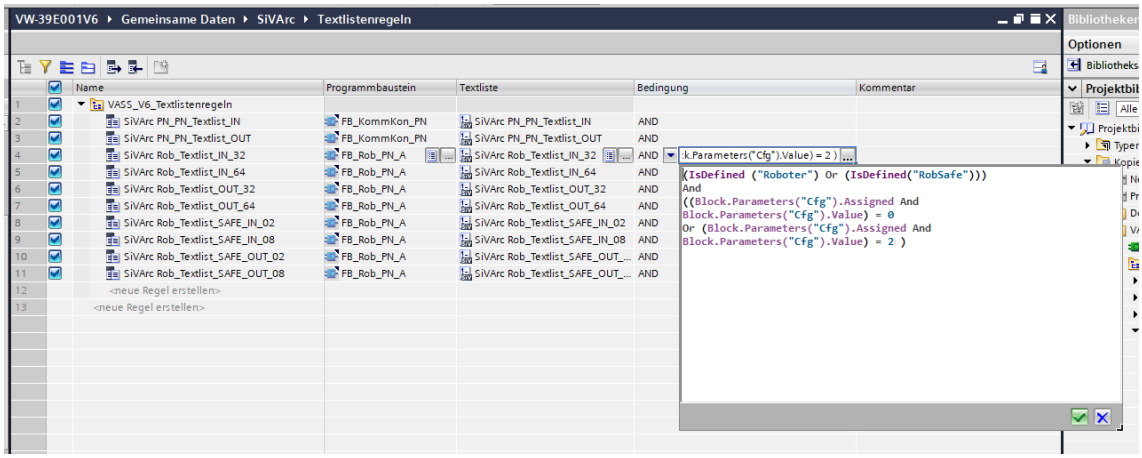


Fig. 6.72 Creation of new text list rules

6.5 "Screen rules" editor

In the "Screen rules" editor, you define the screen rules according to which SiVArc faceplates are generated in screens for various technologies. A rule is made up as follows:

- **Name**
  Unique name of the screen rule.
- **Program block** (module FB)
  Module FB that is called at any position in the user program.
- **Screen object**
  Master copy or type of faceplate that is generated. The master copy or the type must be stored in the project library.
- **Screen** (layout)
  Generation template of the screen on which the faceplate is generated. The generation template must be stored in the project library.
- **Positioning field**
  Positioning field that is included in the positioning scheme of the screen. Use the positioning field to specify the positioning of the faceplate to be generated.
- **Condition** (optional)
  SiVArc expression that is evaluated when processing this screen rule. If no condition is specified, the screen rule is always executed. The condition applies collectively for a rule group. You can refine the condition for individual rules of the rule group.
- **Comment** (optional)
  Individual comment for screen rule

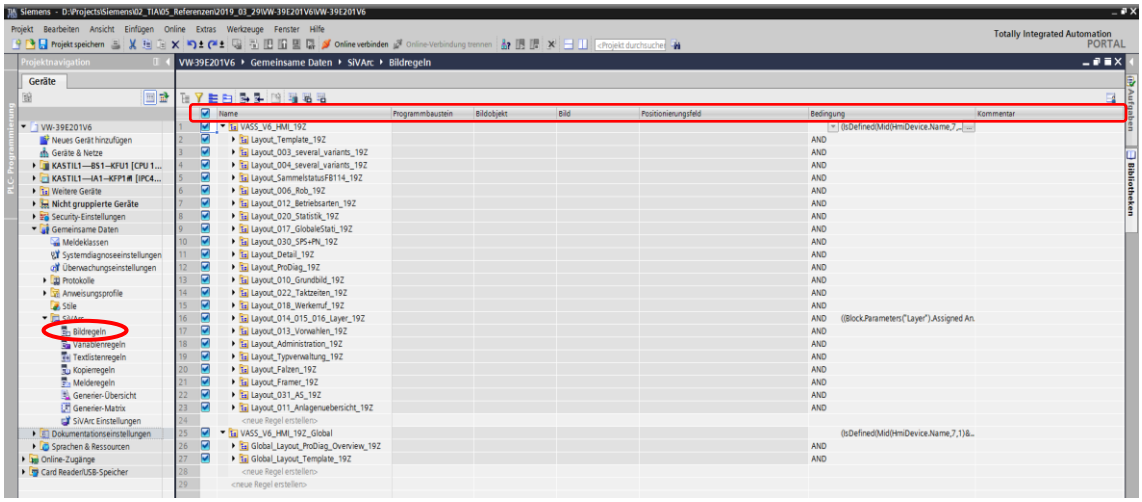


Fig. 6.73 "Screen rules" editor

Display the following columns as required via the icons in the toolbar:

- ① **PLC (CPU)**
  The screen rule is executed for the selected controllers. This point should generally be activated in the VASS standard.
- ② **HMI device**
  The screen rule is executed for the selected HMI devices. This point should generally be activated in the VASS standard.
- ③ **HMI device type**
  If multiple HMI devices of the same type are available in your project, you can also select types of HMI devices. This point should generally be activated in the VASS standard.

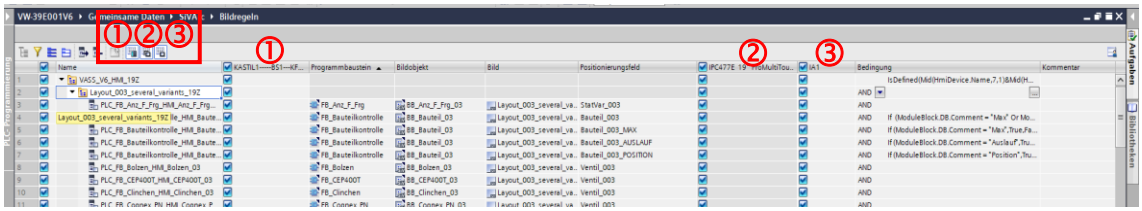


Fig. 6.74 Screen rules and division of the columns

When the program block (module FB_) is processed by SiVArc when the faceplates are generated, SiVArc evaluates the condition of each screen rule.

6.6 Creating screen rules

Requirement

- A test program is created with the call of the module FB and the faceplate.
- The generation template (layout) of a screen has been created (existing layouts should be used).
- The "Screen rules" editor is open.
- The columns "PLC (CPU)" and "HMI device" are visible.

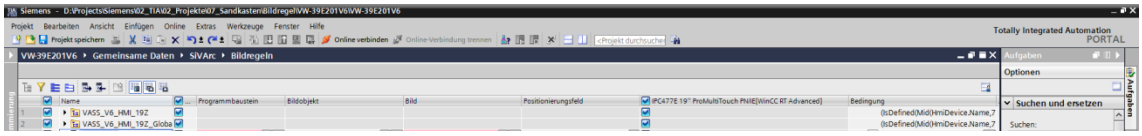


Fig. 6.75 Screen rules editor

Procedure

To define a screen rule for generating a display and operating element, follow these steps:

(in this case, using the example "FB_Ventil", "BB_Ventil_03", layout for 003 pages "Layout_003_several_variants_19Z"):

- Create a screen rule by double-clicking on "<Create new rule>".
  To this end, first minimize any existing folder structure .

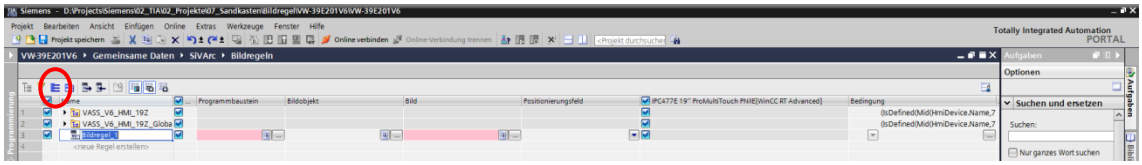


Fig. 6.76 Create new screen rules

- Assign a unique name to the rule. The name is assigned according to the syntax "PLC_FB_FB_name_HMI_faceplate_name_layout_page", for example "PLC_FB_Ventil_HMI_Ventil_03".
  In the case of a screen rule for the field Assembly, the syntax is "PLC_FB_FB_Name_Assembly_HMI_Faceplate_Name_Layout_Page_Assembly".

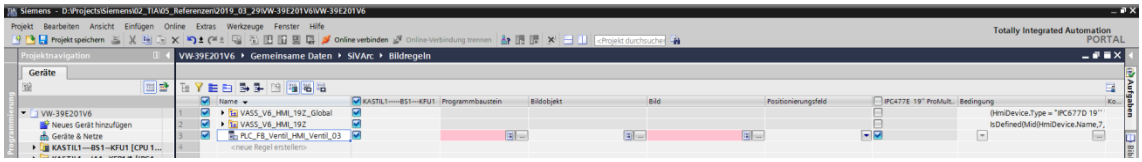


Fig. 6.77 Screen rule: Assigning a name

- Under "PLC", select the controllers for which the screen rule is to apply.
- This point should generally be activated in the VASS standard.
- Select the program block (module FB) for which the faceplate is generated.

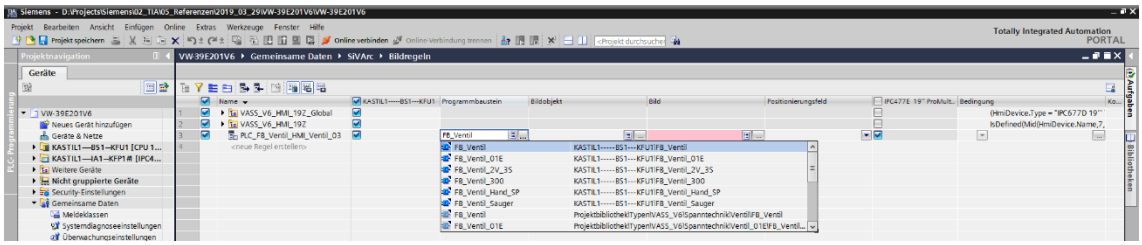


Fig. 6.78 Screen rule: Assigning a program block (module FB)

- Under "Screen object", select the generation template of the display and operating element.

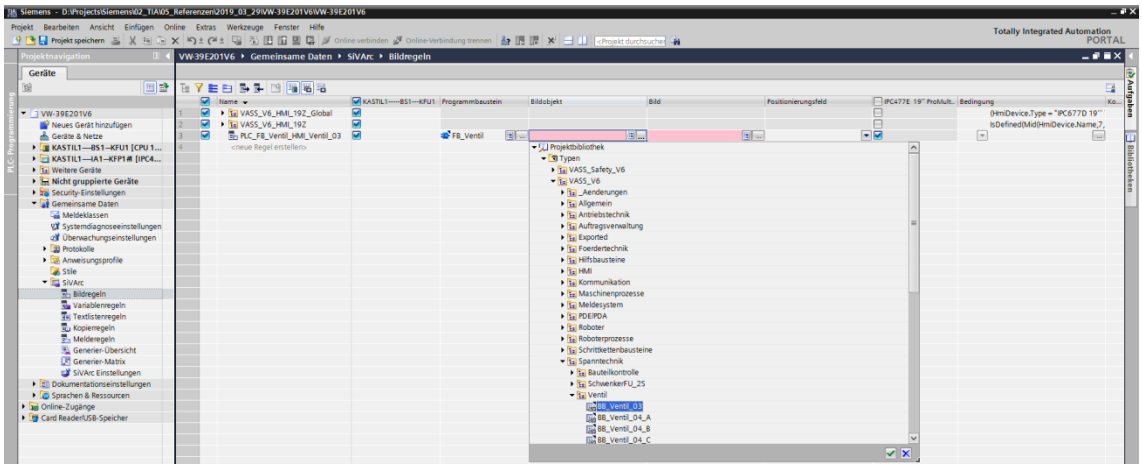


Fig. 6.79 Screen rule: Assigning a program block (module FB)

- Under "Screen", select the generation template of the screen in which the object is generated.

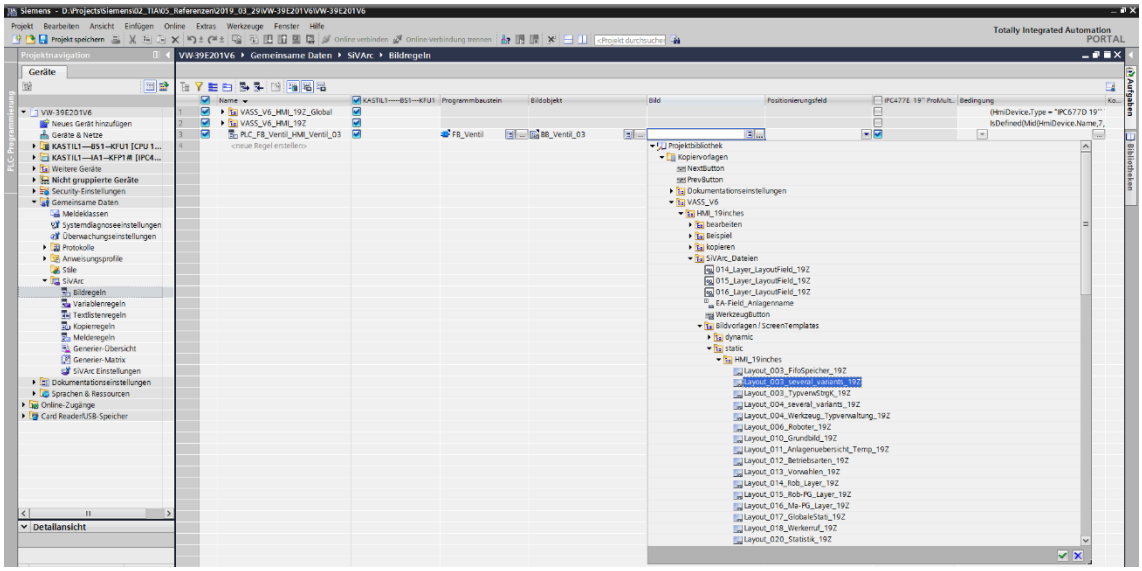


Fig. 6.80 Screen rule: Assigning a generation template (screen)

- If a position schema (layout field) is stored for the generation template, select the placement area under "Positioning field". If you do not specify a placement area, the generated faceplate object is positioned in the screen according to the SiVArc positioning scheme.

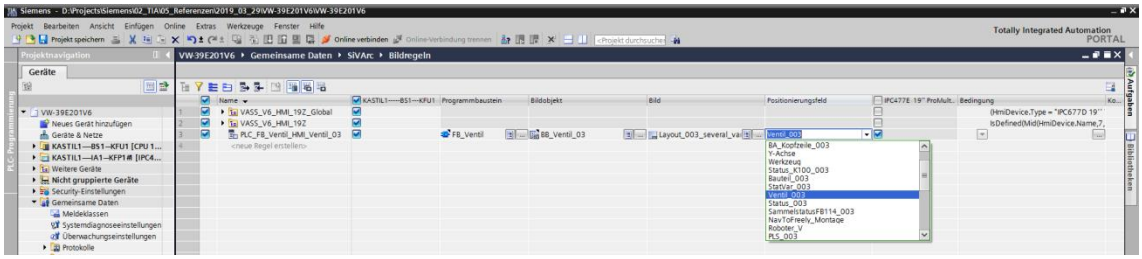


Fig. 6.81 Screen rule: Assigning a generation template (screen)

- Under "HMI device", select the HMI devices for which the screen rule is to apply. Use the toolbar icons to show or hide the device types of the HMI devices.
  If you select no HMI device, the screen rule applies to all HMI devices that are connected to the selected controller.
  If the rule is only to be executed for objects or program blocks (module FB) that meet a specific condition, program the corresponding expression under "Condition" with SiVArc scripting.

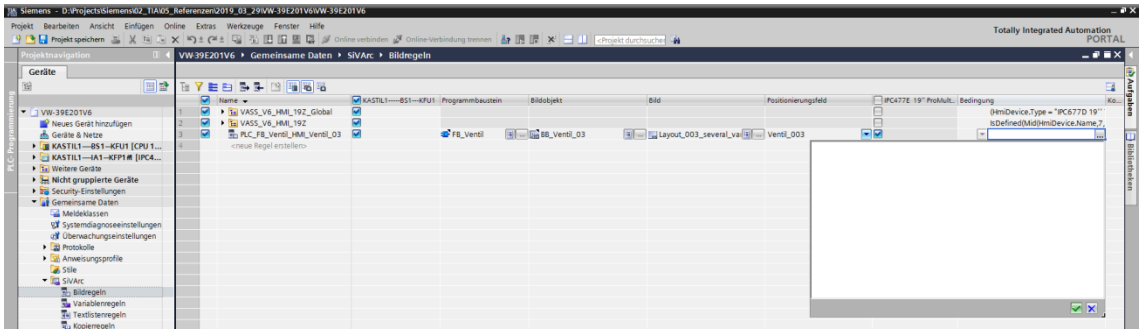


Fig. 6.82 Screen rule: Assigning operation

For the sake of clarification, in this case using the example of the "FB_Ventil" for the screen "Layoutseite 004".

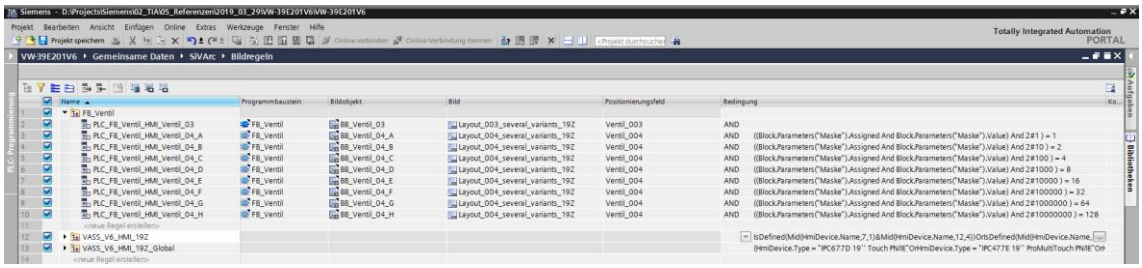


Fig. 6.83 Condition at FB_Ventil for layout page 004

- To combine the individual screen rules to form a group, all rules are marked and grouped together by right-clicking "Add new rule group"

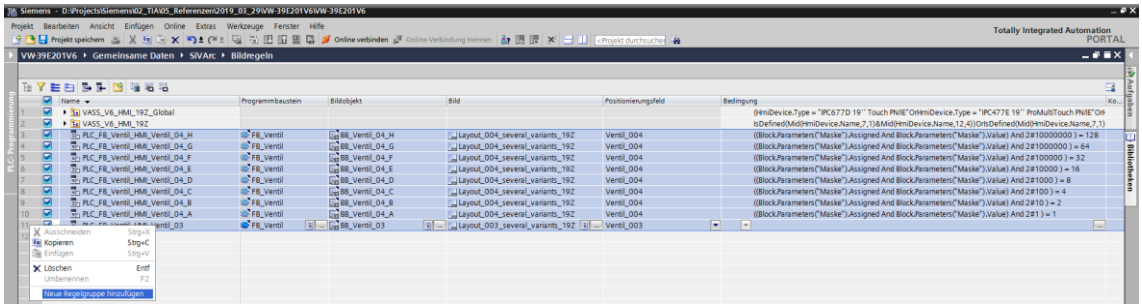


Fig. 6.84 Combining screen rules to form a rule group

- The name of the screen rule group is the same as the module FB name.

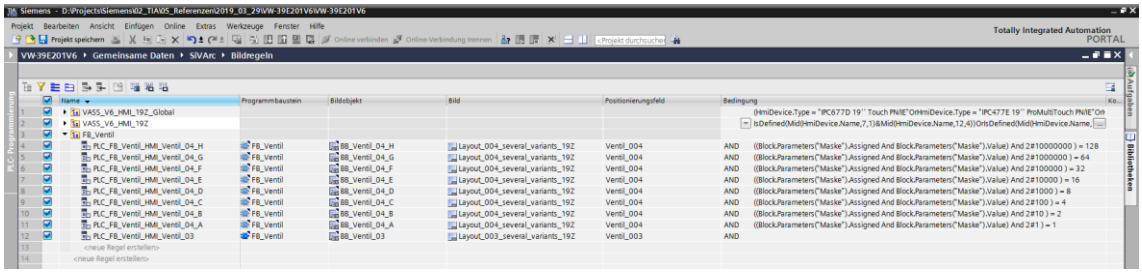


Fig. 6.85 Assign name of rule group

- The screen rule group is then moved to the master copies in the project library.

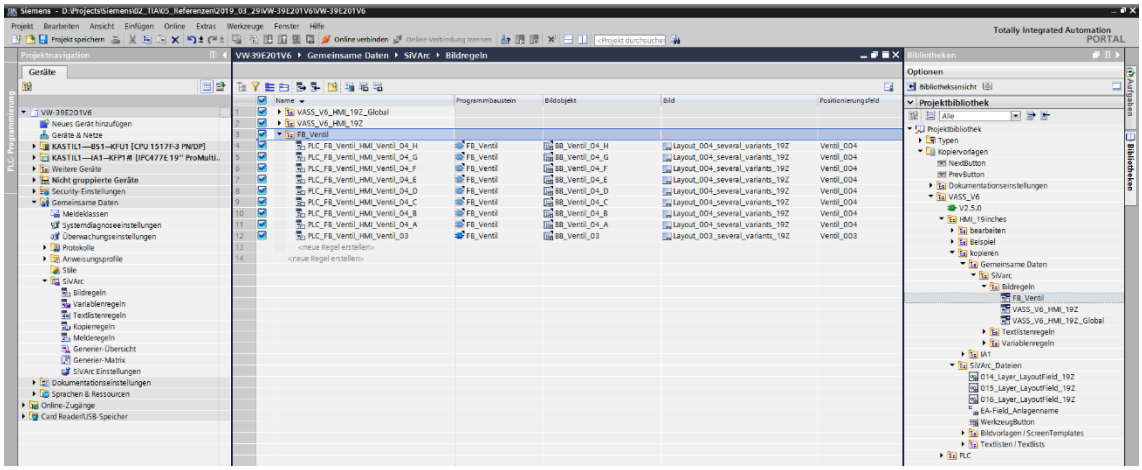


Fig. 6.86 Create copy template

6.7 ASCII modulations with SiVArc expressions

For use in SiVArc / in HMI, the tag names and instance DB comments / instance DB names must be adapted.

6.7.1 Differences in syntax for PLC and HMI tags

In contrast to the HMI tag, the PLC tag contains **"**, or **.** instead of _ !

6.7.1.1 Syntax example PLC tag

"110010DT1#AS_DB**".**OFF_SQ

6.7.1.2 Syntax example HMI tag

110010DT1#AS_DB_OFF_SQ

6.7.2 FB642 FB_S7G_Control with parameter S7G_Control_Ext Var.A

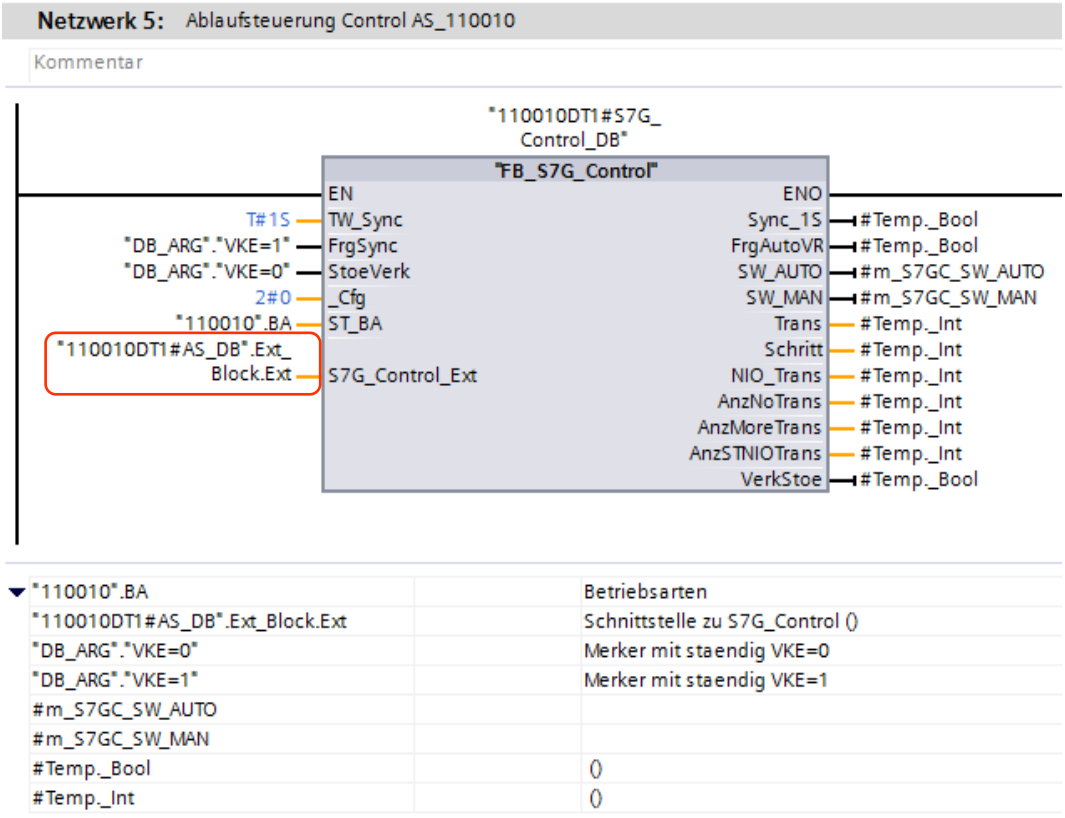


Fig. 6.87 FB642 FB_S7G_Control with parameter S7G_Control_Ext Var.A

Source: FB642 = "FB_S7G_Control" with instance DB and input parameters "S7G_Control_Ext"

Name in PLC: **"110010DT1#AS_DB".Ext_Block.Ext**

Target: "BB_Button_ActivateSequence_Layer" TextField2

SiVArc expression:

```
Replace(Replace(Split(Block.Parameters("S7G_Control_Ext").Value,"_")(0),"\"",""),"#AS","")
```

Results:

"110010DT1#AS
110010DT1#AS
110010DT1

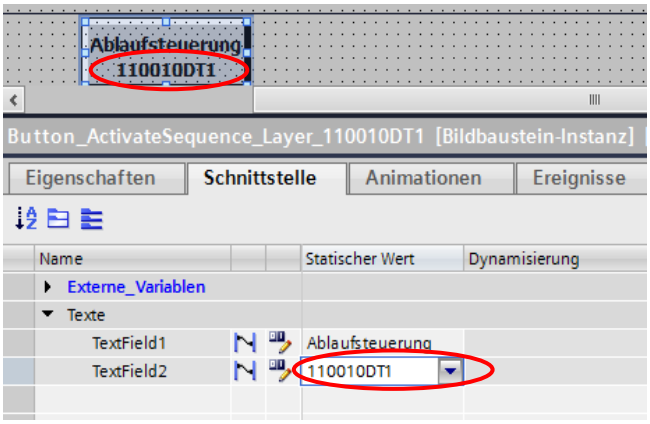


Fig. 6.88 Result "Sequential control 110010DT1"

6.7.3 FB642 FB_S7G_Control with parameter S7G_Control_Ext Var.B

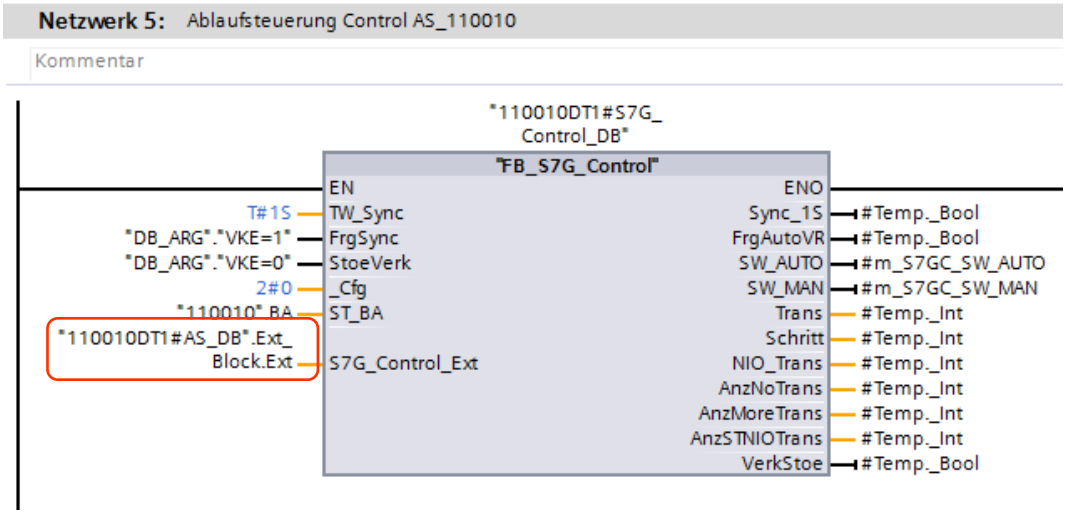


Fig. 6.89 FB642 FB_S7G_Control with parameter S7G_Control_Ext Var.B

Source: FB642 = "FB_S7G_Control" with instance DB and input parameters "S7G_Control_Ext"

Name in PLC: **"110010DT1#AS_DB".Ext_Block.Ext**

Target "GRAPH overview"Assigned GraphDB tag

SiVArc expression:

```
Replace(Replace(Block.Parameters("S7G_Control_Ext").Value,"\".Ext_Block.Ext",""),"\"","")&"_OFF_SQ"
```

Results:

"110010DT1#AS_DB
110010DT1#AS_DB
110010DT1#AS_DB_OFF_SQ

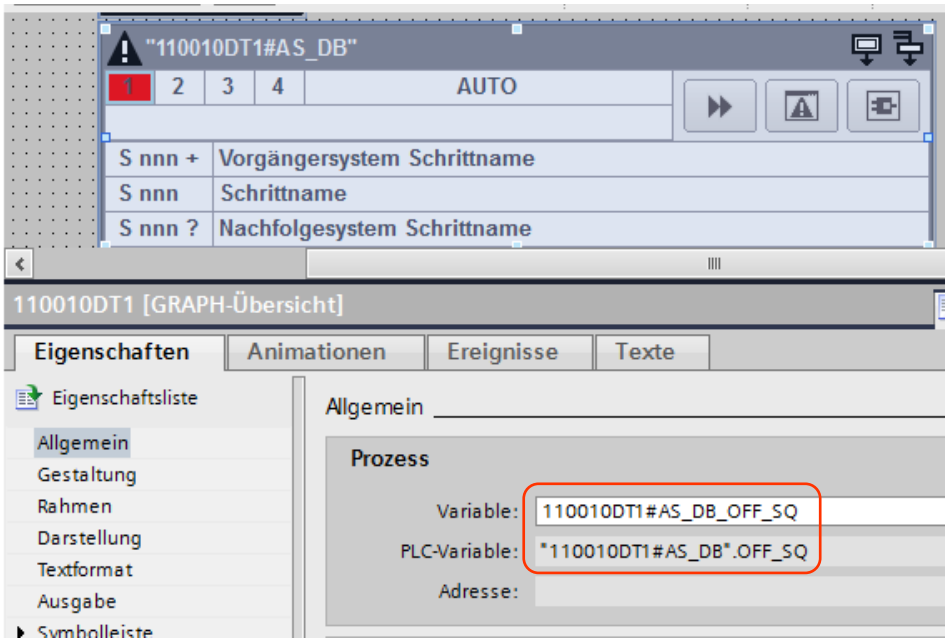


Fig. 6.90 Result "110010DT1#AS_DB_OFF_SQ "

6.7.4 FB_Vorwahl TextFields

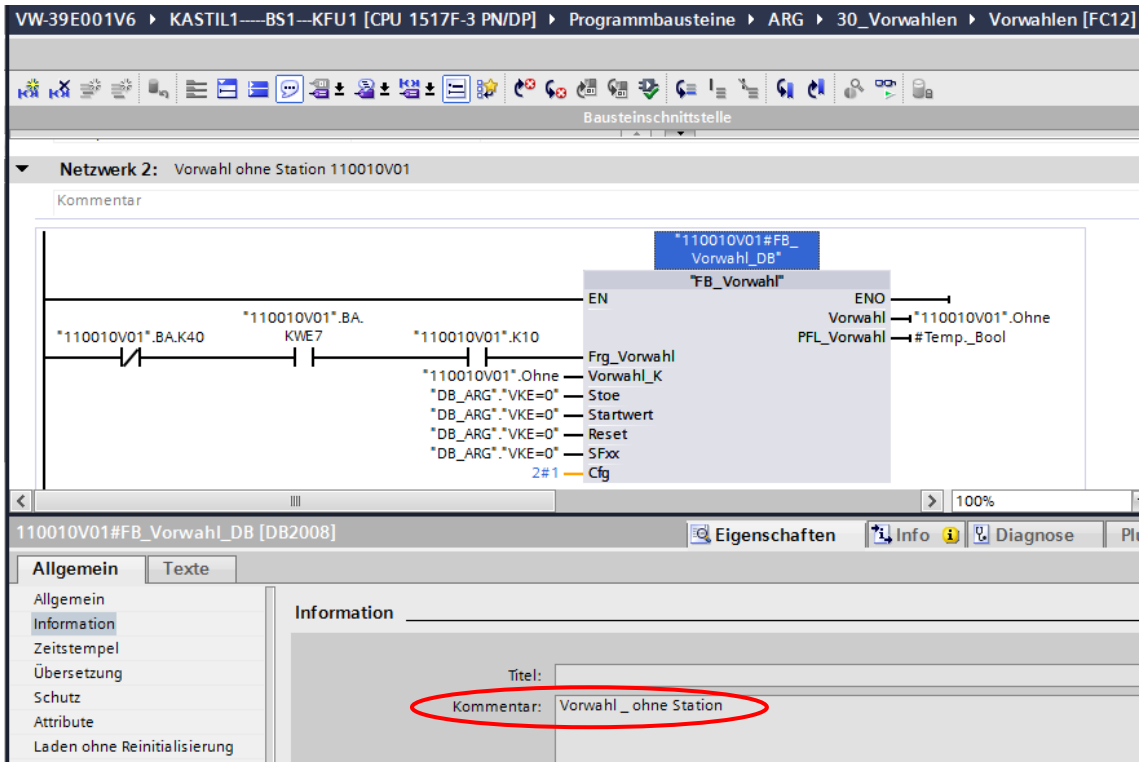


Fig. 6.91 FB_Vorwahl TextFields (1)

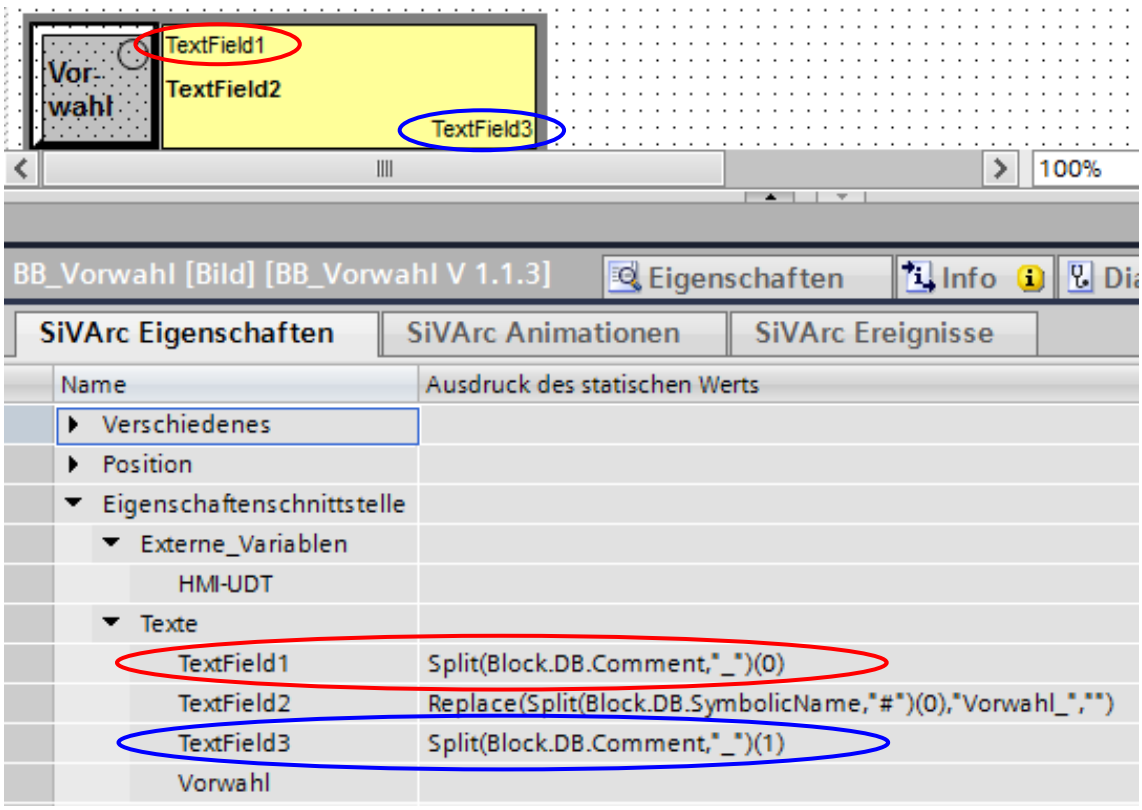


Fig. 6.92 FB_Vorwahl TextFields (2)

6.7.4.1 FB_Vorwahl TextField1

Source: FB145 = "FB_Vorwahl"
Instance DB comment: Pre-selection _ without station

Target: Preselection

SiVArc Expression: (BB_Vorwahl TextField1) **Split(Block.DB.Comment,"_")(0)**

Result: **Preselection (Vorwahl)**

6.7.4.2 FB_Vorwahl TextField3

Source: FB145 = "FB_Vorwahl"
Instance DB comment: Pre-selection _ without station

Destination: without station

SiVArc Expression: (BB_Vorwahl TextField3) **Split(Block.DB.Comment,"_")(1)**

Result: **without station (ohne Station)**

6.7.4.3 Result in screen Vorwahlen_013

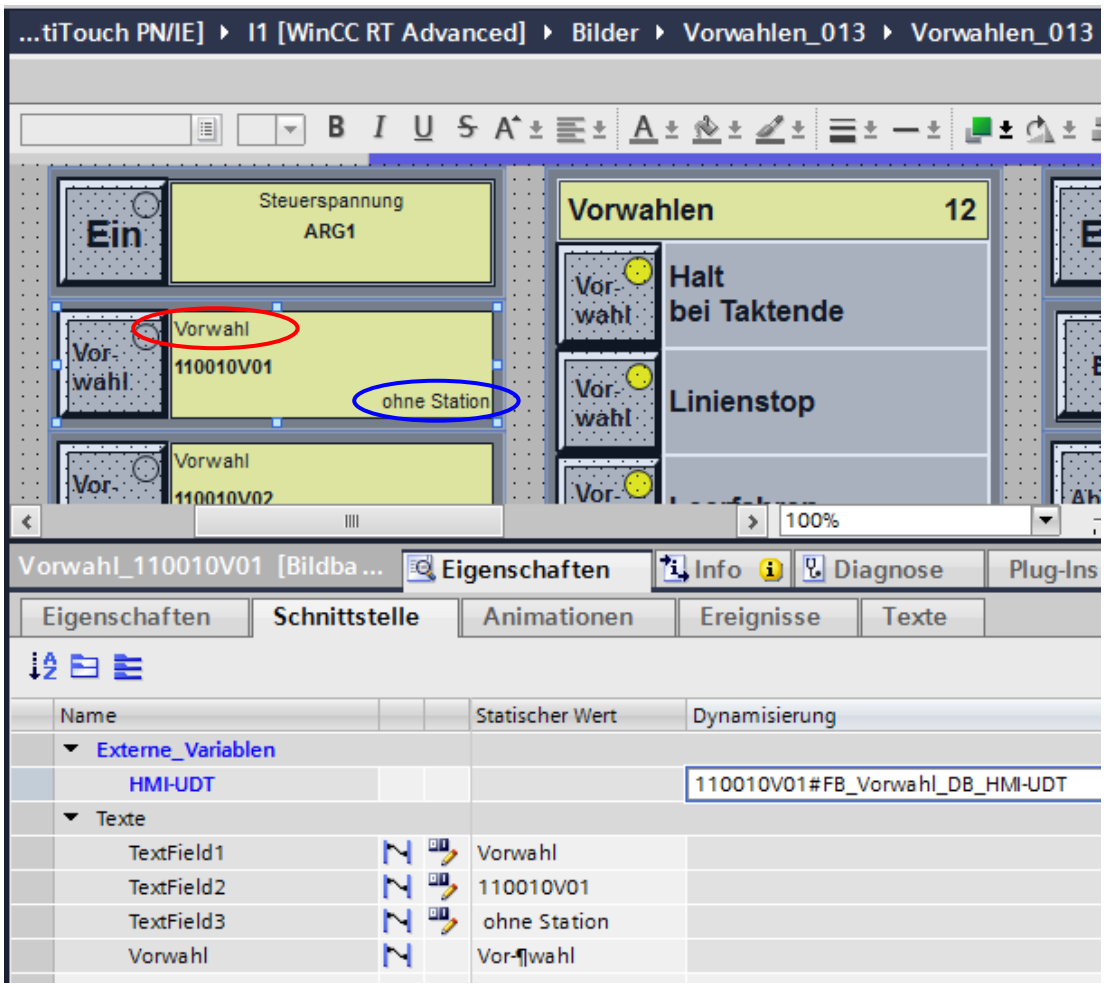


Fig. 6.93 Result in screen Vorwahlen_013

7 Block help

Using the key combination **"[Shift] + [F1]"** it is possible to display a corresponding help text for each S7 block.

A separate document should be created for each S7 block.

The procedure for creating such a document is created step by step below.

7.1 Storage location

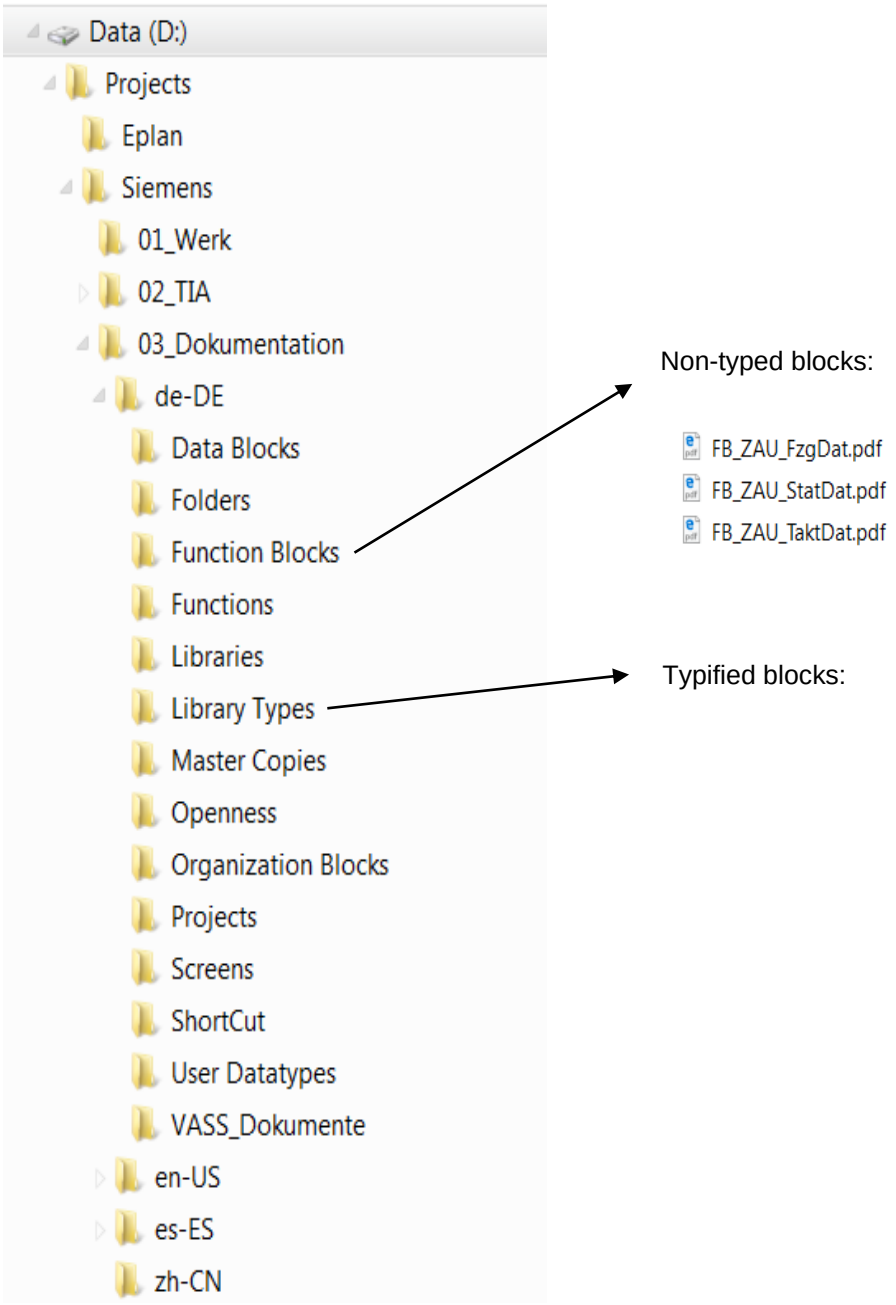


Fig. 7.1 Storage location

7.2 Premises

To ensure the quick and correct conversion of the created Word files, these premises must be observed during creation.

7.2.1 File name

The file name must be identical to the module component name. Storage is in the folder path of the corresponding language.

The file names may not contain Ä, Ö, Ü.

Example:

- FB_BA.pdf

7.2.2 File type

The block help may be of the type "*.doc" or "*.docx".

7.2.3 Header

The header (title, version, date) should be adopted as in the sample document.

Example:

FB xxx FB_yyy

Version 1.1.0

Date of help: xx.yy.2018

7.2.4 Continuous document

The block help must be written as a continuous document. No print preview.

No page breaks or section breaks needed.

Superfluous paragraph marks should be avoided.

7.2.5 Screens / screenshots

The inserted screens should be inserted in a sufficiently high quality.

7.2.6 Interconnection examples

The interconnection examples should always be shown with symbols under the block.

7.3 Notes

7.3.1 Hyperlinks

Hyperlinks are not supported.

7.4 Description of the content of a block help

The content of a block help can be seen by way of example in FB 120 "FB_Druckluft_VW370".

The blue texts and the graphics in the blue frames that appear in the following chapters 7.4.1 to 7.4.12 are the original texts from the "[Shift] + [F1]" help on block FB 120.

Note

The chapters shown in the example must always be created and described.

FB 120 FB_Druckluft_VW370
Version 1.2.0

Date of help file: August 6, 2019

7.4.1 Brief description

This block is used to actuate a valve cluster (central distribution board) with a "KKP01E" switch-on valve and a "KKP01D" pneumatic valve. The switch-on valve is used to enable the control air for operating the valves in the valve cluster (main pneumatic valve and main cooling water valve). The block also measures compressed air and air use (for compressed air meters).

This block is usually accessed with a dedicated instance data block.

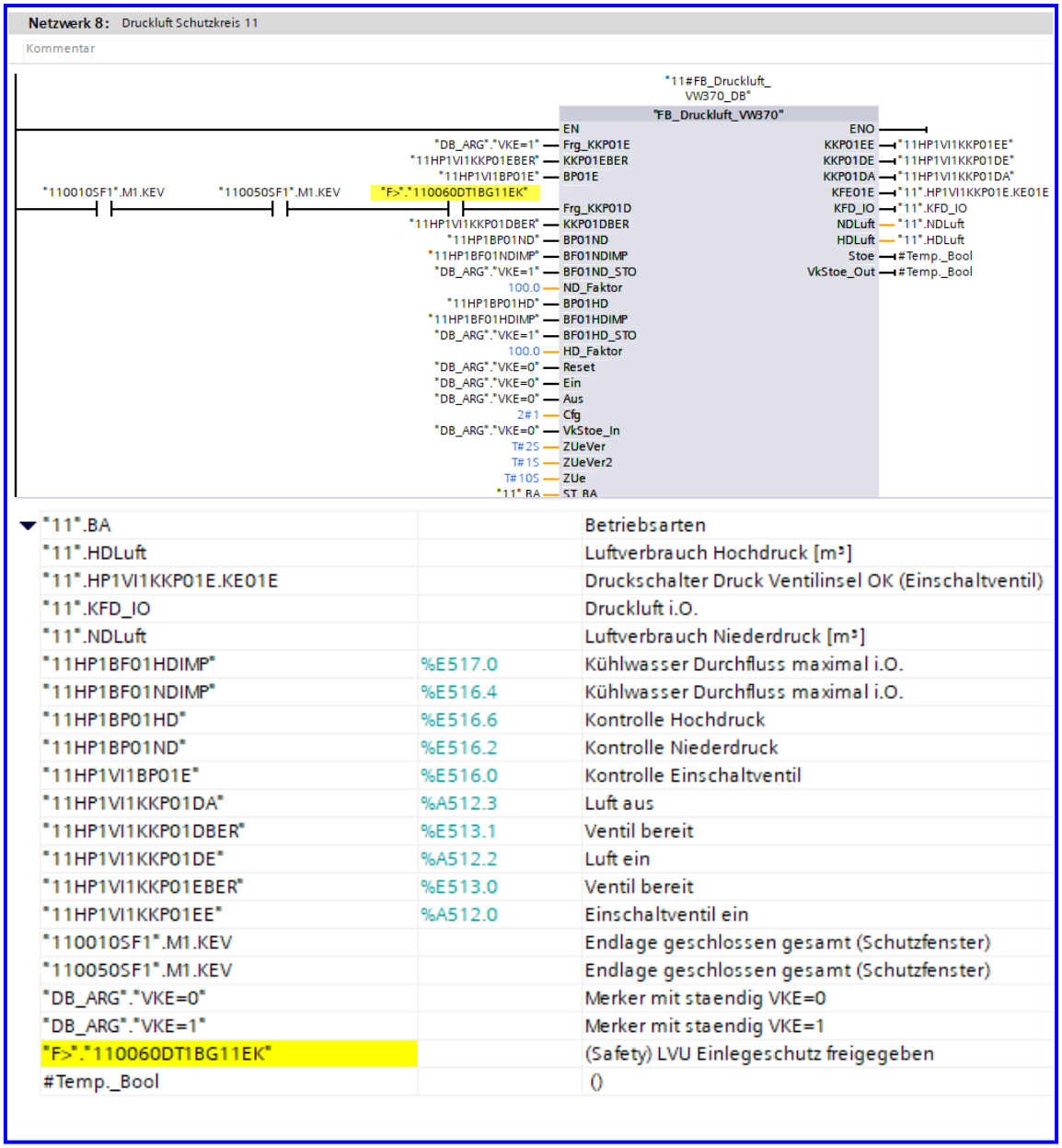
The "symbolic name" of the instance data block must appear as follows:

xx#FB_Druckluft_VW370_DB e. g. 11#FB_Druckluft_VW370_DB

The following should be entered for "symbol comment":

Compressed air VW370

7.4.2 Wiring example



"11".BA		Betriebsarten
"11".HDLuft		Luftverbrauch Hochdruck [m³]
"11".HP1VI1KKP01E.KE01E		Druckschalter Druck Ventilinsel OK (Einschaltventil)
"11".KFD_IO		Druckluft i.O.
"11".NDLuft		Luftverbrauch Niederdruck [m³]
"11HP1BF01HDIMP"	%E517.0	Kühlwasser Durchfluss maximal i.O.
"11HP1BF01NDIMP"	%E516.4	Kühlwasser Durchfluss maximal i.O.
"11HP1BP01HD"	%E516.6	Kontrolle Hochdruck
"11HP1BP01ND"	%E516.2	Kontrolle Niederdruck
"11HP1VI1BP01E"	%E516.0	Kontrolle Einschaltventil
"11HP1VI1KKP01DA"	%A512.3	Luft aus
"11HP1VI1KKP01DBER"	%E513.1	Ventil bereit
"11HP1VI1KKP01DE"	%A512.2	Luft ein
"11HP1VI1KKP01EBER"	%E513.0	Ventil bereit
"11HP1VI1KKP01EE"	%A512.0	Einschaltventil ein
"110010SF1".M1.KEV		Endlage geschlossen gesamt (Schutzfenster)
"110050SF1".M1.KEV		Endlage geschlossen gesamt (Schutzfenster)
"DB_ARG"."VKE=0"		Merker mit staendig VKE=0
"DB_ARG"."VKE=1"		Merker mit staendig VKE=1
"F>"."110060DT1BG11EK"		(Safety) LVU Einlegeschutz freigegeben
#Temp._Bool		()

Fig. 7.2 Screenshot of "FB_Druckluft_VW370"

7.4.3 Input parameters

Name	Type	Description
Frg_KKP01E	BOOL	Switch-on valve enable
KKP01EBER	BOOL	Valve cluster ready, switch-on valve
BP01E	BOOL	Pressure switch, switch-on valve (control air active)
Frg_KKP01D	BOOL	Compressed air enable (additional enables to connect/disconnect compressed air, e.g. protective window closed)
KKP01DBER	BOOL	Valve cluster ready, compressed air
BP01ND	BOOL	Air low pressure
BF01NDIMP	BOOL	Pulse input, air consumption, low pressure
BF01ND_STO	BOOL	Air consumption, low pressure, no fault
ND_Faktor	REAL	Low-pressure factor (e.g. 1 pulse = 0.01 Nm³ → 10)
BP01HD	BOOL	Air, high pressure
BF01HDIMP	BOOL	Pulse input, air consumption, high pressure
BF01HD_STO	BOOL	Air consumption, high pressure, no fault
HP_Faktor	REAL	High-pressure factor (e.g. 1 pulse = 0.01 Nm³ → 10)
Reset	BOOL	Reset all air meters
On	BOOL	Switching on
Off	BOOL	Switching off
Cfg	WORD	Configuration bits (see Configuration bits table)
VkStoe_In	BOOL	Linked fault
ZUeVer	TIME	Time until pressure reached, pilot valve
ZUeVer2	TIME	Time until pressure falls, pilot valve
ZUe	TIME	Time for high pressure, low pressure switch-on and switch-off control

Tab. 7.1 Input parameters

Configuration bits

Parameter	Bit	Status	Description
Cfg	00	True False	With low pressure and with high pressure With low pressure and without high pressure
	01	True	Output NDair and HDair with factor * 10 (one decimal place remains, measuring range is reduced from 32767 to 3276.7) Display value on the visualization remains unchanged
	02..15		n.c.

n.c. = not connected (not used)

Tab. 7.2 Configuration bits

7.4.4 Output parameters

Name	Type	Description
KKP01EE	BOOL	Output, switch-on valve On
KKP01DE	BOOL	Output, pneumatic valve On
KKP01DA	BOOL	Output, pneumatic valve Off
KFE01E	BOOL	Control air enable
KFD_IO	BOOL	Compressed air on and no errors
NDLuft	INT	Air consumption, low pressure (in Nm³)
HDLuft	INT	Air consumption, high pressure (in Nm³)
Stoe	BOOL	Group fault
VkStoe_Out	BOOL	Linked fault

Tab. 7.3 Output parameters

7.4.5 Input and output parameters

Name	Type	Description
ST_BA	ST_Betriebsarten	Operating mode structure, provides all operating mode information such as manual, auto, etc., to the block.

Tab. 7.4 Input and output parameters

7.4.6 External variable references

Global variables

These variables are taken straight from the global variable declarations.

Name	Type	Description
PC_AKTIV	DWORD	Each bit in the least significant nibble of this byte signals a valid connection and a key operation on a PC: Bit 00: PC 1 Bit 01: PC 2 Bit 02: PC 3 Bit 03: PC 4
STEU_EIN	BOOL	Control voltage is switched on

Tab. 7.5 External tag references

7.4.7 Dependencies

ST_Betriebsarten

7.4.8 Operating principle

The switch-on valve output "KKP01EE" is actuated with the switch-on valve enable input "Frg_KKP01E" and, if there are no faults, with the compressed air switch-on valve.

With control air enable "KFE01E", the emergency stop enabled, the safety circuit closed, start Off and compressed air enable "Frg_KKP01D", compressed air can be switched on and off with the visu or the "On" and "Off" inputs. The edge is evaluated for the inputs.

Compressed air is still switched off after 10 seconds if there is no compressed air pending at the "BP01ND" or "BP01HD" pressure switch. (Manual operation of ball valve)

The states compressed air on, compressed air off, selection, high pressure, low pressure, group faults and the activation conditions are processed for visualization. Air consumption can be measured with the air consumption pulse inputs " BF01NDIMP / BF01HDIMP " and displayed on the visu in normal cubic meters (Nm^3). The air meters can be reset with the reset button + SWE7 or with the "Reset" input. Air consumption in m^3 can be further processed with the "NDLuft / HDLuft" outputs.

Example:

Central distribution board valve cluster with "KKP01E" switch-on valve, "KKP01D" pneumatic valve and "KKP01W" cooling water valve.

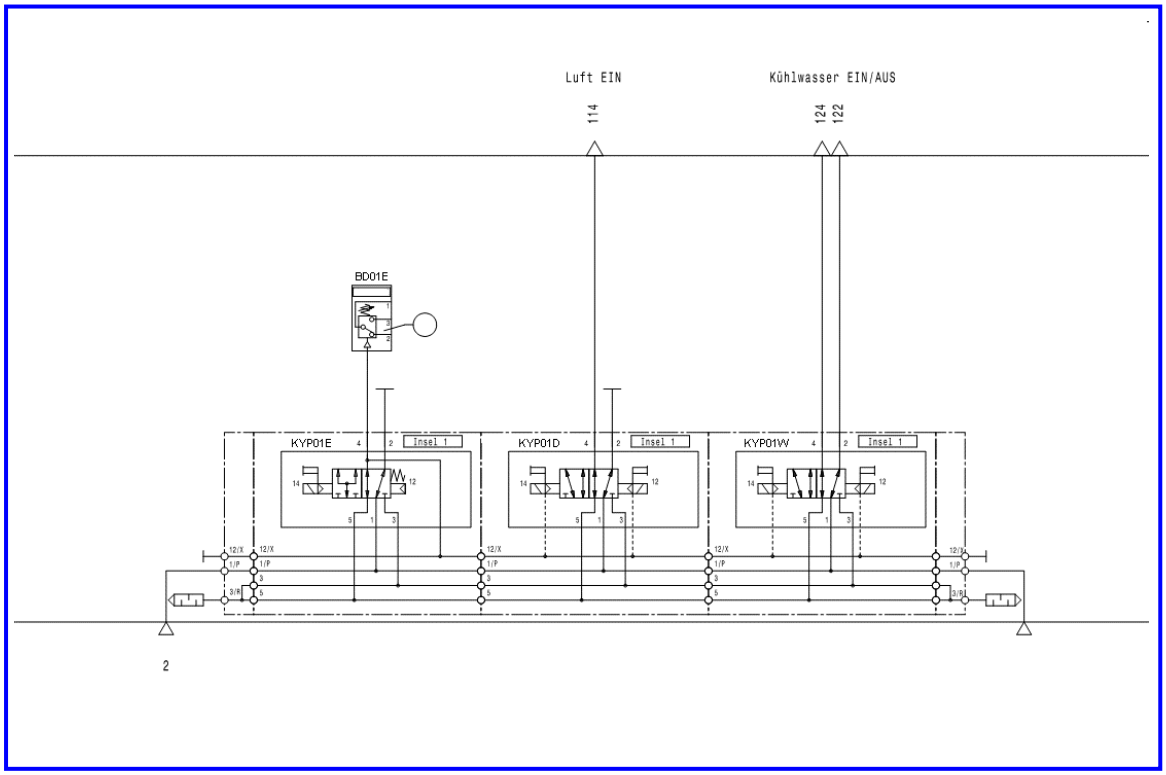


Fig. 7.3 HIP valve terminal

Example:

Central distribution board, pneumatics, air pressure and cooling water

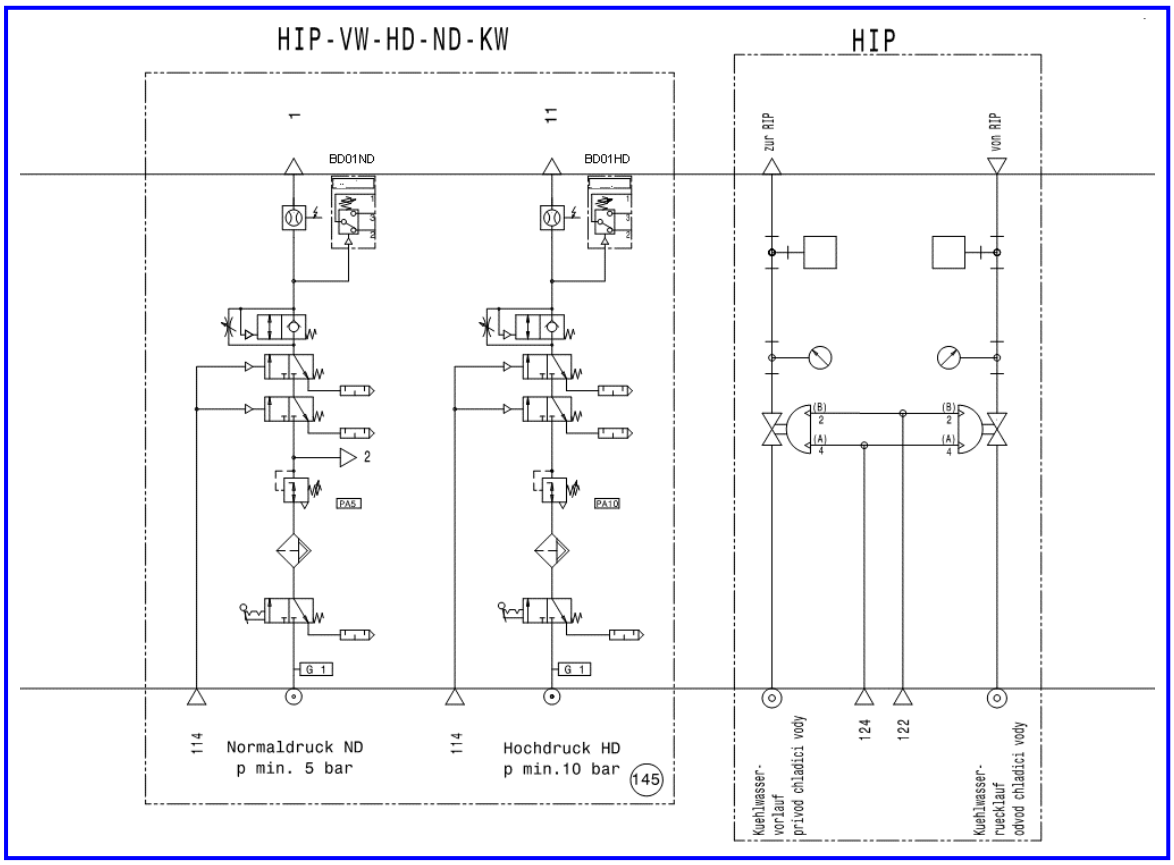


Fig. 7.4 HIP Pneumatics

7.4.9 Alarm system

Message text	Prioritization	Cause / *Solution*
Pressure switch fault, switch-on valve	STE	After pilot valve enable, the pressure switch does not output "Pressure reached" ('BP01E') within the configured delay time (ZUeVer) The delay time may be too short. → Increase the delay time Check the pressure switch or pressure switch feedback. Check the valves or compressed air system.
Actuator fault, switch-on valve	STE	After the switch-on valve is switched off, the pressure switch is still outputting "Pressure reached" ('BP01E') within the configured delay time (ZUeVer2) The delay time may be too short. → Increase the delay time Check the pressure switch or pressure switch feedback. Check the valves or compressed air system.
Fault, valve coil interruption or short-circuit, switch-on valve	STE	The valve cluster signals a interruption or short-circuit in the valve coil of the switch-on valve Check the valve coil for a short-circuit / check valve coil installation.
Fault, switch-on, low pressure	ST0	After compressed air activation, the pressure switch does not output "Pressure reached" ('BP01ND') within the configured delay time (ZUe) The delay time may be too short → Increase the delay time Check the pressure switch or pressure switch feedback. Check the valves or compressed air system.
Fault, switch-on, high pressure	ST0	After compressed air activation, the pressure switch does not output "Pressure reached" ('BP01HD') within the configured delay time (ZUe) The delay time may be too short. → Increase the delay time Check the pressure switch or pressure switch feedback. Check the valves or compressed air system.

Fault, switch-off, low pressure	ST0	After compressed air switch-off, the pressure switch continues to output "Pressure reached" ('BP01ND') within the configured delay time (ZUe) The delay time may be too short → Increase the delay time Check the pressure switch or pressure switch feedback. Check the valves or compressed air system.
Fault, switch-off, high pressure	ST0	After compressed air switch-off, the pressure switch continues to output "Pressure reached" ('BP01HD') within the configured delay time (ZUe) The delay time may be too short. → Increase the delay time Check the pressure switch or pressure switch feedback. Check the valves or compressed air system.
No compressed air	STE	The compressed air is switched off or no pressure switch is signaling "Pressure reached" Switch on compressed air.
Pressure switch fault, low pressure	STE	Compressed air is switched on: The pressure switch signal ('BP01ND') "Pressure reached" switches from True to False. Compressed air is switched off: The pressure switch signal ('BP01ND') "Pressure reached" switches from False to True. Check the pressure switch or pressure switch feedback. Check the valves or compressed air system.
Pressure switch fault, high pressure	STE	Compressed air is switched on: The pressure switch signal ('BP01HD') "Pressure reached" switches from True to False. Compressed air is switched off: The pressure switch signal ('BP01HD') "Pressure reached" switches from False to True. Check the pressure switch or pressure switch feedback. Check the valves or compressed air system.

Fault, valve coil interruption or short-circuit, pneumatic valve	STE	The valve cluster signals a interruption or short-circuit in the valve coil of the pneumatic valve(s) Check the valve coil for a short-circuit / check valve coil installation.
Fault, air meter, low pressure	ST0	The air meter does not signal no fault. Check the air meter.
Fault, air meter, high pressure	ST0	The air meter does not signal no fault. Check the air meter.

Tab. 7.6 Error messages

7.4.10 Visualization

Visu variables

These variables are used by visualization.

The visu variables are declared in the VAR area.

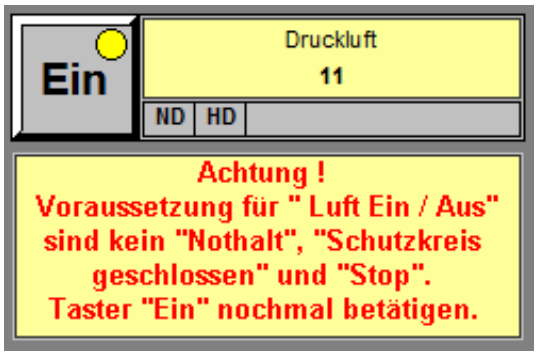
Name	Type	Description

dwVisuWerte1	DWORD	Change of color for the symbol bit assignment
		dwVisuWerte1.X0: = False;
		dwVisuWerte1.X1: = xF_Einschalt_ND Or xF_Einschalt_HD;
		dwVisuWerte.X2: = xF_Ausschalt_ND Or xF_Ausschalt_HD;
		dwVisuWerte.X3: = xAnw_Ein;
		dwVisuWerte.X4: = KKP01DE;
		dwVisuWerte.X5 := KKP01DA;
		dwVisuWerte.X7 := Stoe;
		dwVisuWerte.X8 := xDruckluftEin And BP01ND;
		dwVisuWerte.X9 := xDruckluftEin And (BP01HD Or Not xCFG_0);
		dwVisuWerte.X10: = xHM_FrgAnwahl ;
		dwVisuWerte.X11: = xVisuHinweis;
		dwVisuWerte.X12: = False;
		dwVisuWerte.X13: = BP01ND;
		dwVisuWerte.X14: = xF_Einschalt_ND Or xF_Ausschalt_ND Or xF_ND;
		dwVisuWerte.X15: = Not xCFG_0;
		dwVisuWerte.X16: = BP01HD;
		dwVisuWerte.X17:= xF_Einschalt_HD Or xF_Ausschalt_HD Or xF_HD;
		
		dwVisuWerte.X24: = KKP01EE;
		dwVisuWerte1.X25: = KFE01E;
		dwVisuWerte1.X26: = xStoe_KKP01E;
		
		dwVisuWerte1.X27: = ST_BA.SFWE7;
		dwVisuWerte1.X28: = xF_BF01NDSTO Or xF_BF01HDSTO;

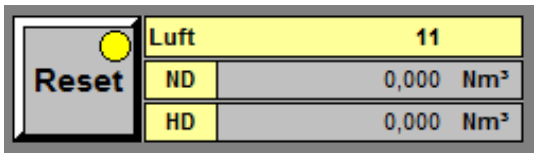
bVisuTasten	BYTE	Visualization key number The value range is defined as follows: 001 = "Selection" key activated

Tab. 7.7 Visualization

7.4.11 Faceplates



Druckluft (in 001 Allgemein)



EnergieLuft (in 001 Allgemein)

Fig. 7.5 Faceplates in the project library

7.4.12 Operation/visualization information

Media pre-selection

The media pre-selections include "compressed air" and "cooling water".

Example media pre-selection compressed air

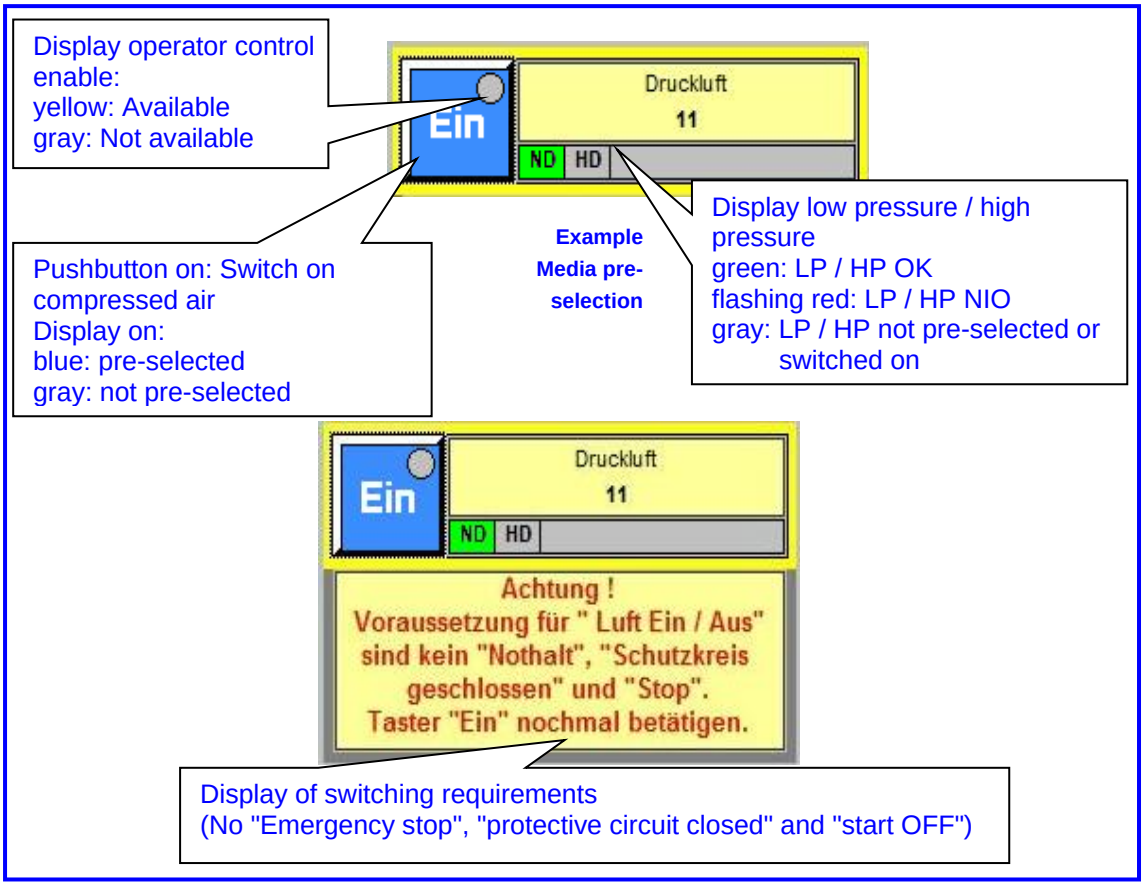


Fig. 7.6 Operation / visualization information

Important:

To obtain the operator control enable for this object, the "on" button must be actuated. A text field is now shown in which the requirements for switching on/off are listed. If these are already met, the operator control enable is already available and the switching operation can now be carried out after re-actuation of the "on" button. If no switching operation is carried out within 5 seconds, the test field is automatically hidden again.

7.5 Conversion of the block help from MS WORD into PDF format

7.5.1 General

As the call of the block help should preferably be in PDF format from the TIA Portal, the Word documents must also be stored in PDF format after creation.

7.5.2 Handling

To create the PDF file, the MS-Word function "Save as" should be used and the following procedure is used.

- Select "Save as…"
- Select file type PDF "*.pdf"
- Click on "Options…"

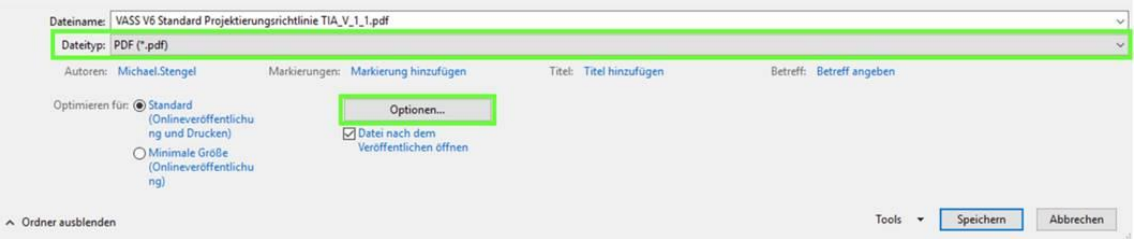


Fig 7.7 Create PDF file

- Under "Options" remove the two check marks under "Include non-printing information".

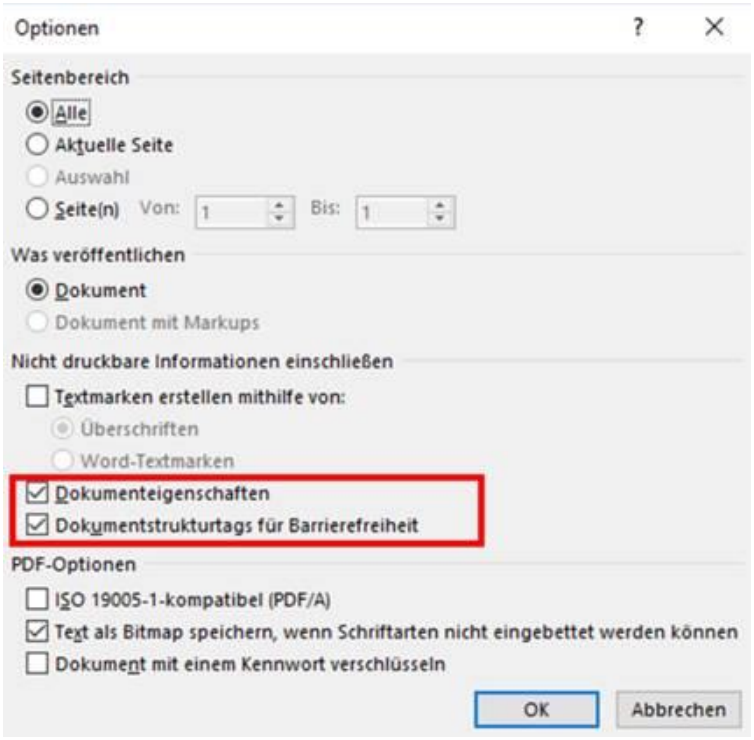


Fig. 7.8 Checkmarks must be removed

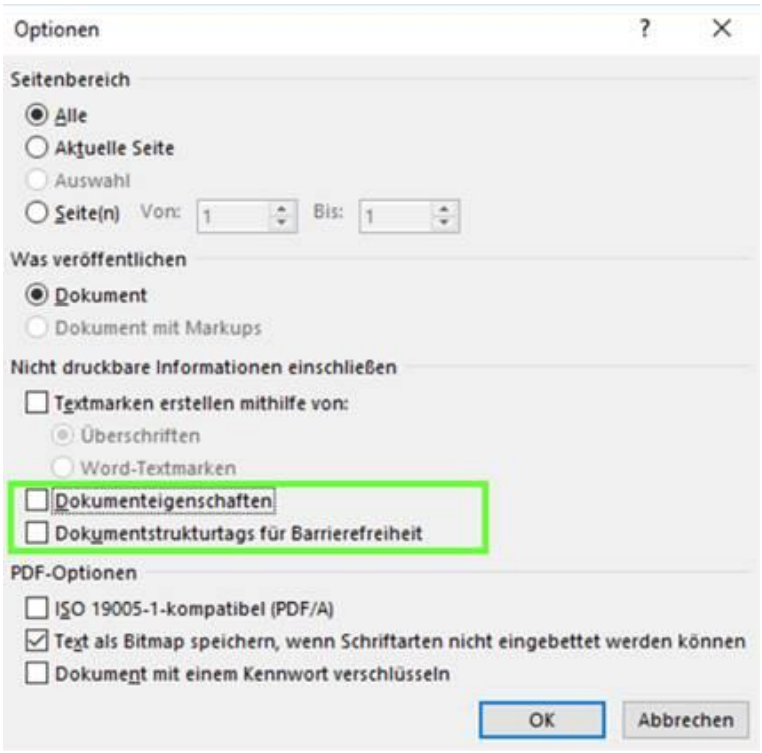


Fig. 7.9 Checkmarks were removed

The PDF file must be stored in the project as described in Chapter 7.1 "Storage location".

Note

The created PDF document must be checked to ensure that graphics, tables, etc. have been correctly created. This may have to be corrected accordingly in the Word document.

8 Transfer of a VASS module

So that a VASS module is always transferred with all its components, a folder structure is defined for this in which the individual parts have to be stored.

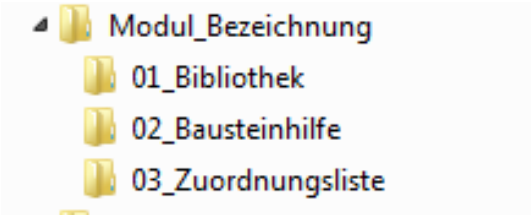


Fig. 8.1 Folder structure (transfer of a VASS module)

Example FB_Ventil:

Fig. 8.2 Folder structure using FB_Ventil as an example

8.1 Library

The library file (global library) exported from the TIA Portal is stored in the folder "01_Bibliothek". A library usually consists of a module FB, a faceplate and a screen rule.

8.1.1 Brand or plant library

Libraries which are specially created for a brand or a plant (and not just from a module FB and associated faceplate(s)) are referred to accordingly as brand or plant libraries.

The folders and subfolders in the project library must be structured in the same way as the structure of the VASS library.

Only the structure elements which are needed to find the required cross connections to the VASS group library are to be taken into account.

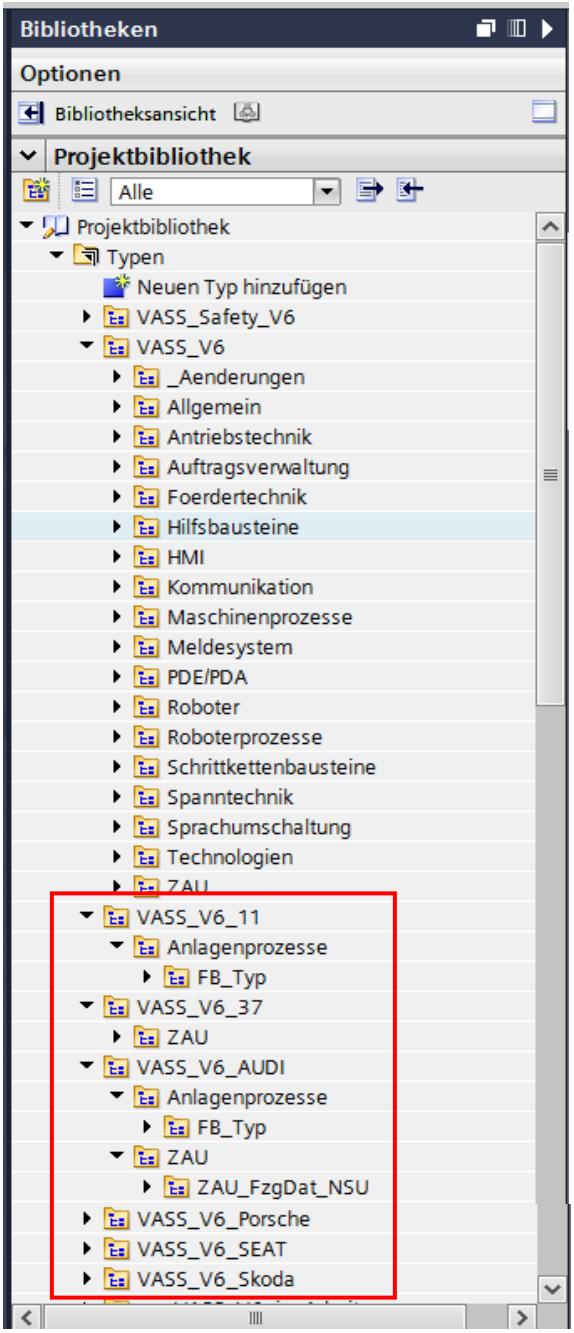


Fig. 8.3 Structure types project library

All elements that are not typified (e.g. screen rules) are located under the main folder "Master copies".

To ensure version control and automatic management via TIA Openness, an FC without content is dragged to the folder with the version as the FC name.

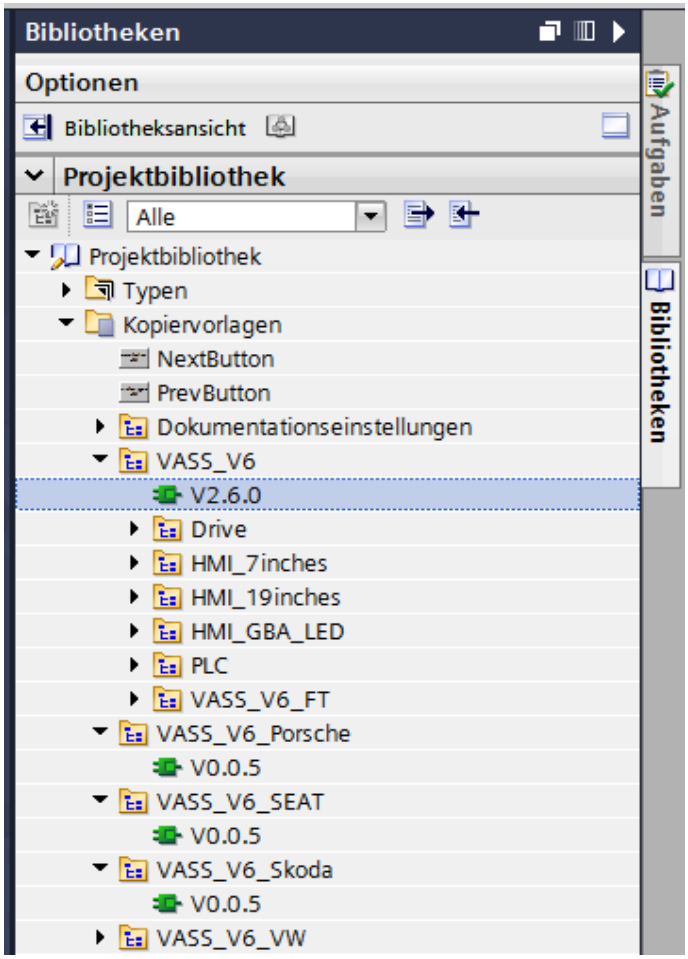


Fig. 8.4 Structure of master copies project library

The global library must then be created as follows:

- Creating a new global library and assignment of the library names in accordance with the information in Chapter 4 "Specifications on brands and plant-specific standards" (e.g. Wolfsburg site (11) "VASS_V6_11_2019_05_25").

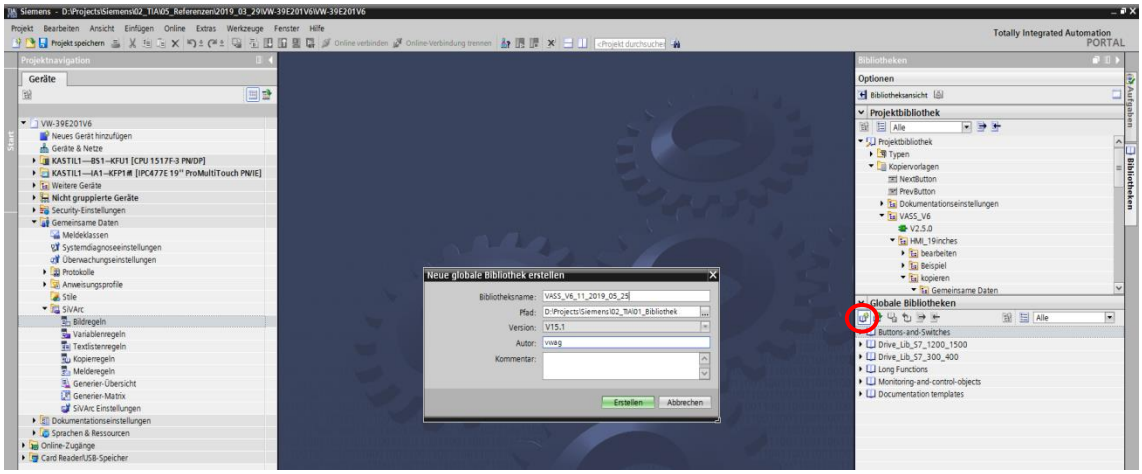


Fig. 8.5 Creation of a global library

- The generated and typified elements (module FBs, faceplates, structures) are copied from the project library to the newly created global library by right-clicking on the "Module folder → Update types → Library".

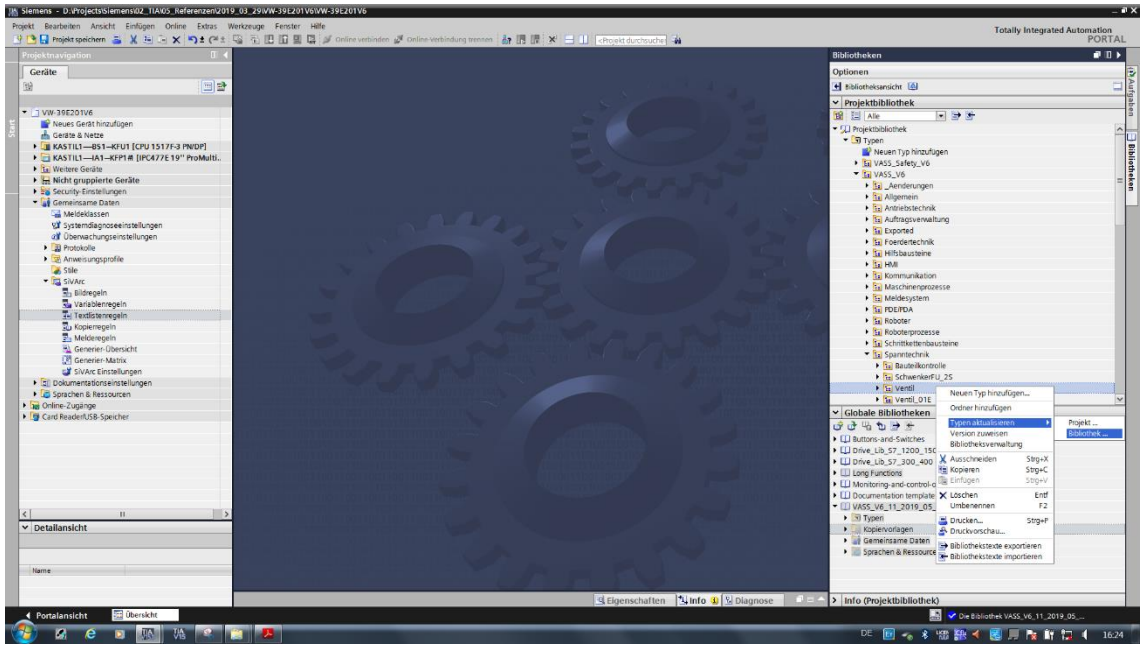


Fig. 8.6 Update types in the library

- The library must be selected again in the dialog that opens.

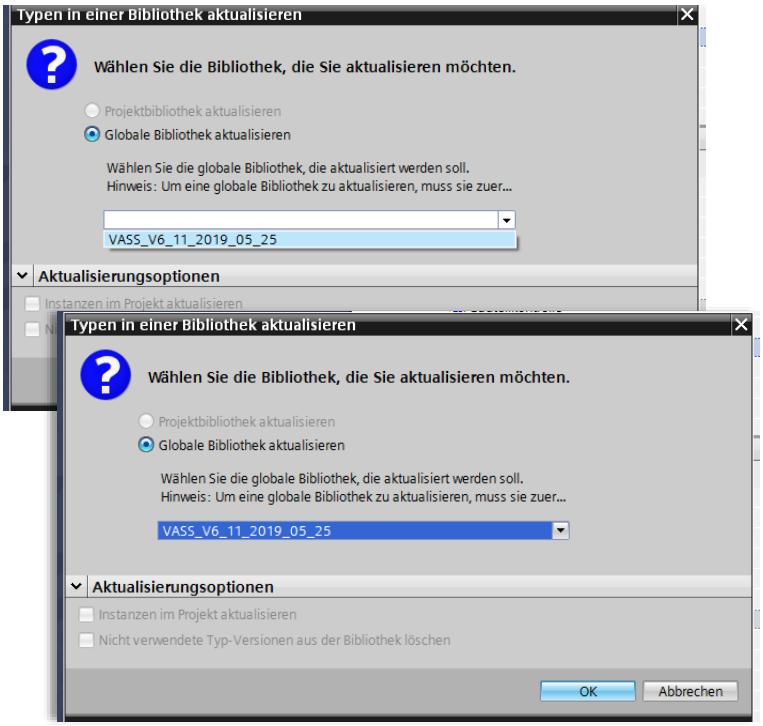


Fig. 8.7 Selection of library

- This step must be repeated until all typified modules have been copied or all modules can be selected at the same time and copied to the global library in one operation.

- When the typified modules are copied, modules which are used by the module FBs (such as tool blocks and structures) are also automatically copied.
  In addition, the folder structure is created in the same way as the project library.

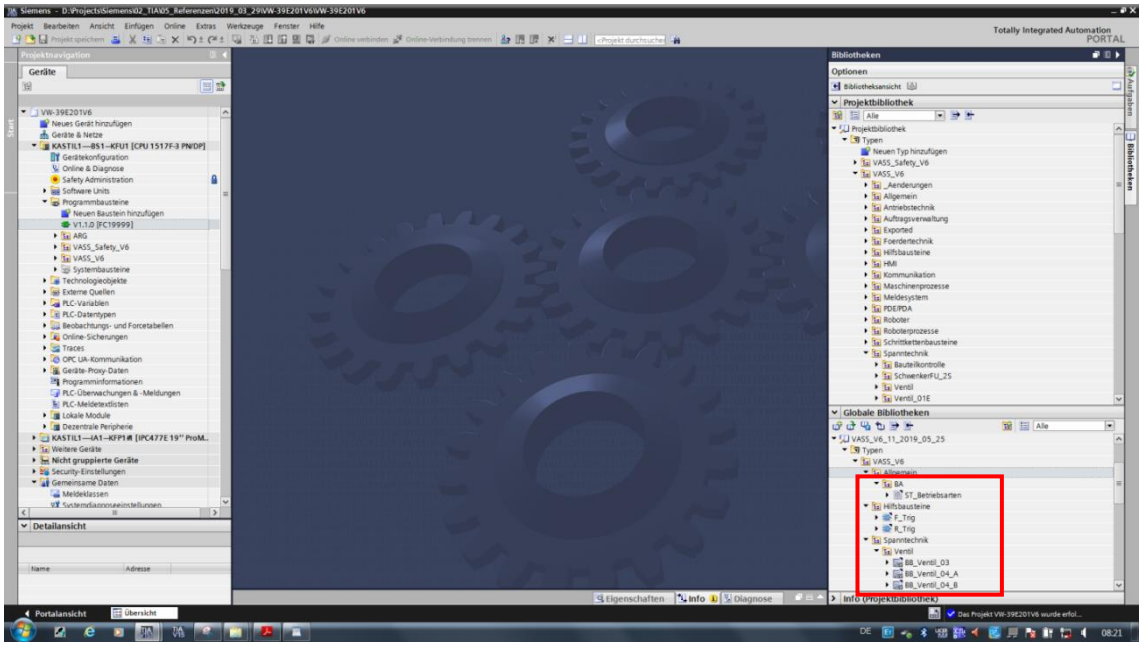


Fig. 8.8 Result: Updating types

- Screen rules are not typified and must be manually copied from the project library to the global library. As a result, the subfolder structure must be manually created under Master copies for the relevant categories.
- The screen rule can easily be dragged to the created folder using "drag & drop".

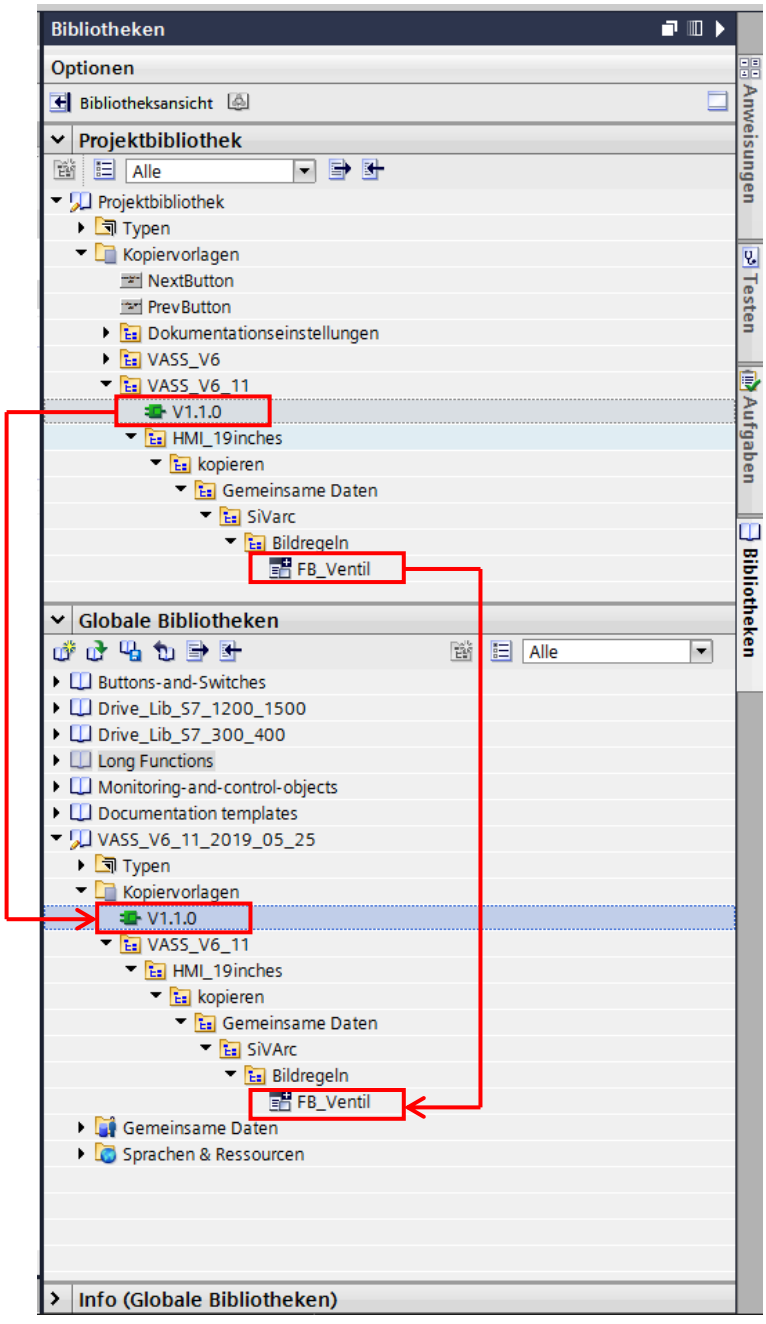


Fig. 8.9 Copying master copies to the global library

If all master copies are copied to the global library, they can be saved and archived in accordance with Chapter 8.1.3 "Saving and archiving the global library".

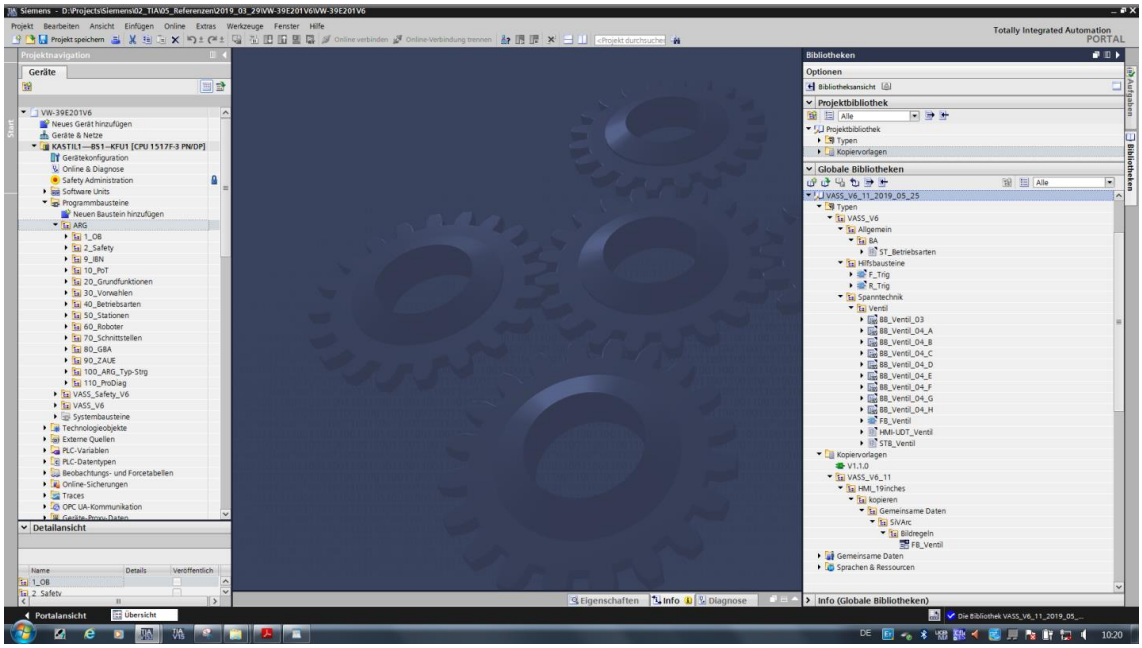


Fig. 8.10 Completed global library

8.1.2 Library for a single module

For the event that only one single module is to be transferred, the library is either a brand or a plant library as described in Chapter 8.1.1 "Brand or plant library".

Structuring is carried out as follows.

The folders and subfolders in the project library must be structured in the same way as the structure of the VASS library (applicable categories).

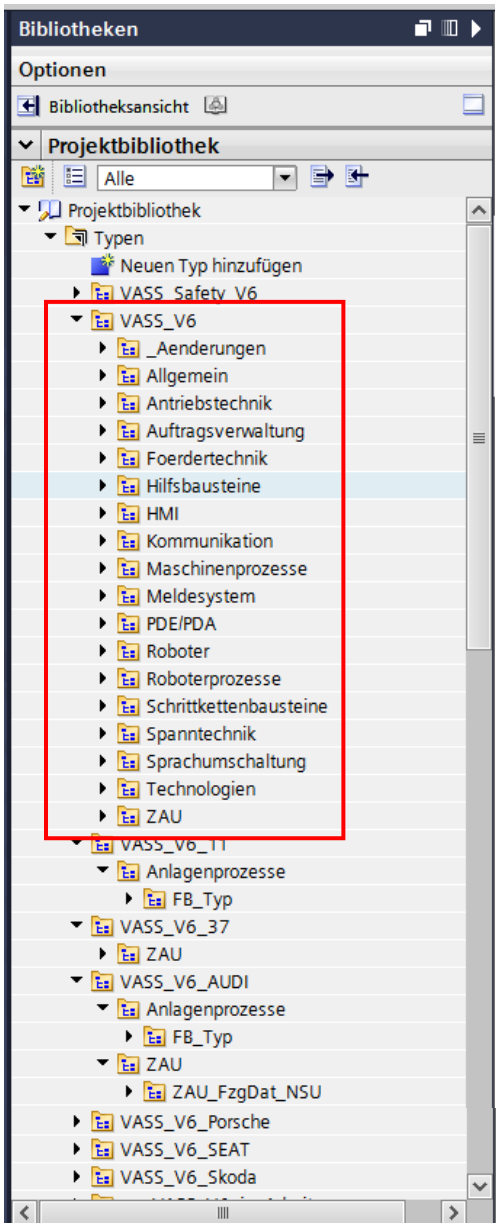


Fig. 8.11 Structure types project library

All elements that are not typified (e.g. screen rules) are located under the main folder "Master copies".

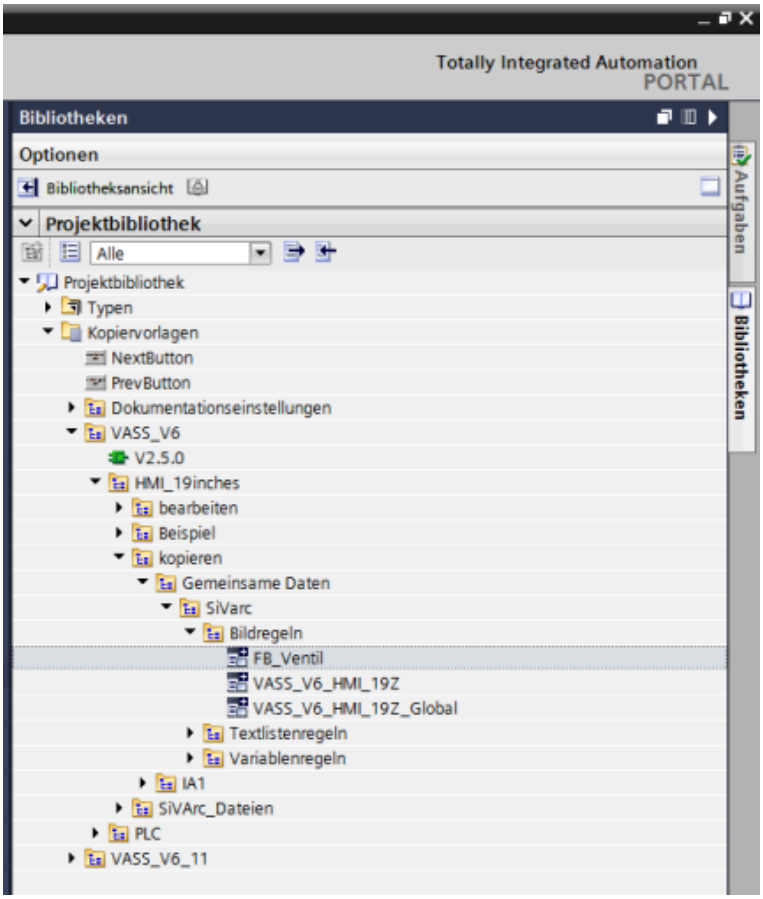


Fig. 8.12 Structure of master copies project library

The global library must then be created as follows:

- Creation of a new global library and assignment of library names.
  It consists of the "module FB name" and the block version (e.g.: FB_Ventil_V_1_1_0).

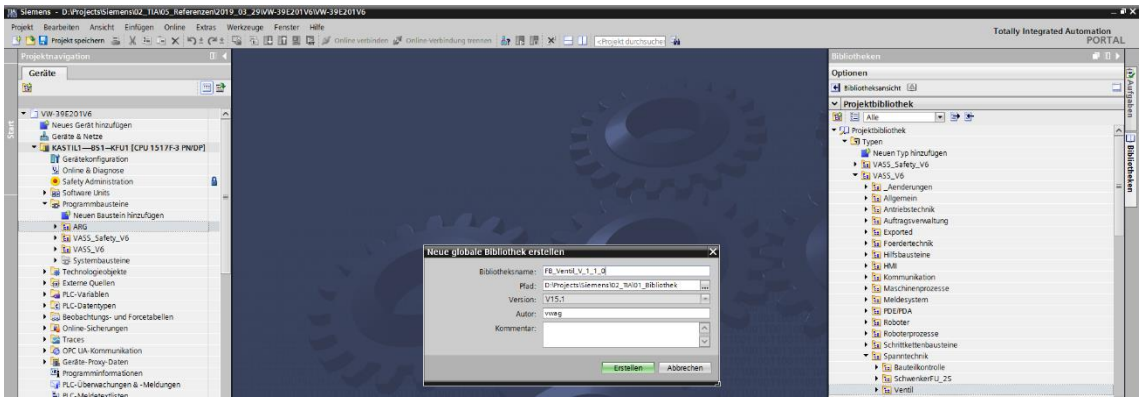


Fig. 8.13 Creation of a global library

- The generated and typified elements (module FB, faceplates, structures) are copied from the project library to the newly created global library by right-clicking on the "Module folder → Update types → Library".

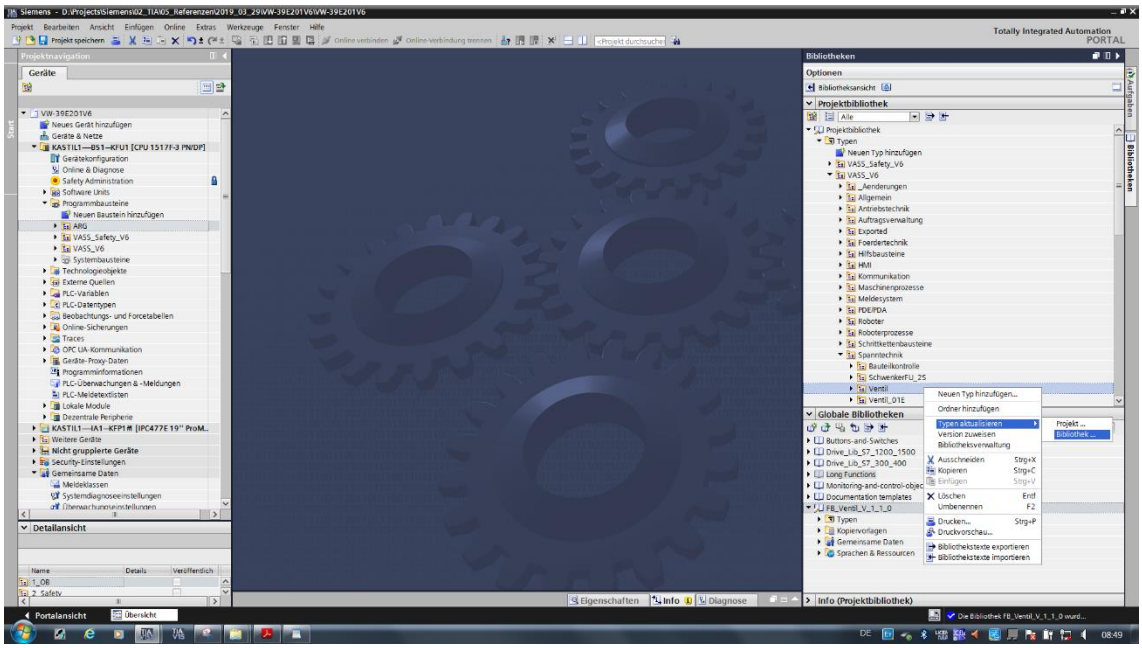


Fig. 8.14 Update types in the library

- The library must be selected again in the dialog that opens.

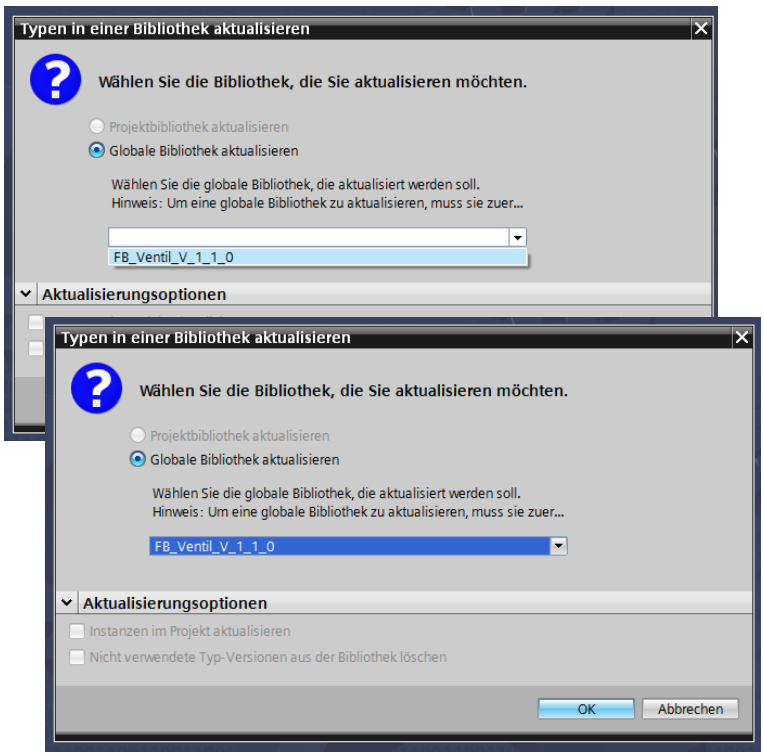


Fig. 8.15 Selection of library

- When the typified modules are copied, modules which are used by the module FBs (such as tool blocks and structures) are also automatically copied.
  In addition, the folder structure is created in the same way as the project library.

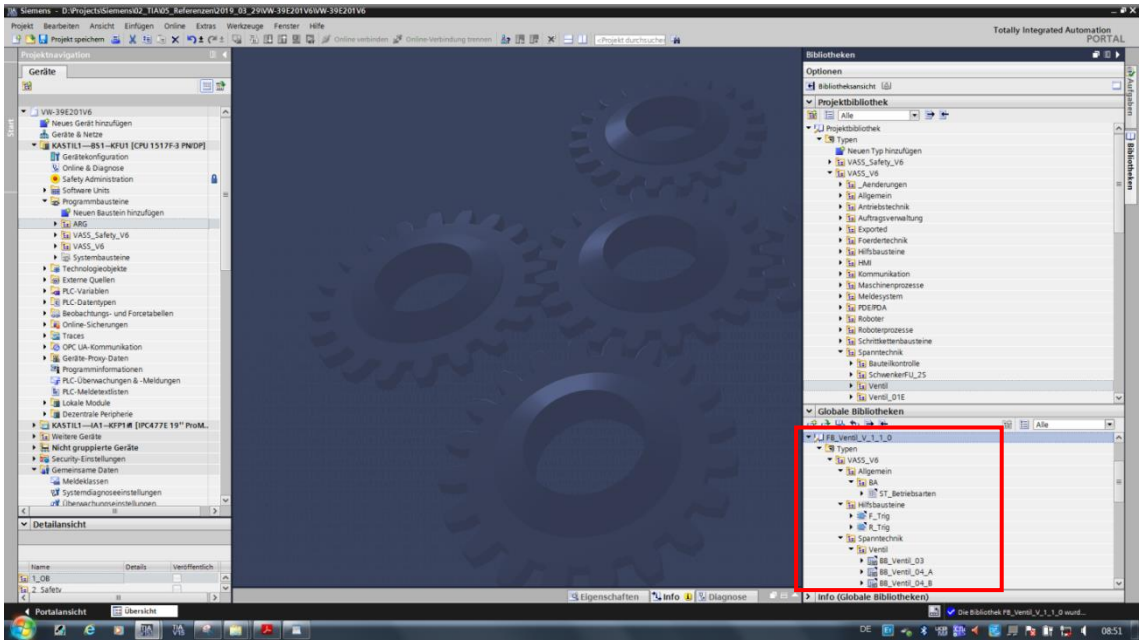


Fig. 8.16 Result

- Screen rules are not typified and must be manually copied from the project library to the global library. As a result, the subfolder structure must be manually created under Master copies for the relevant categories.
- The screen rule can easily be dragged to the created folder using "drag & drop".

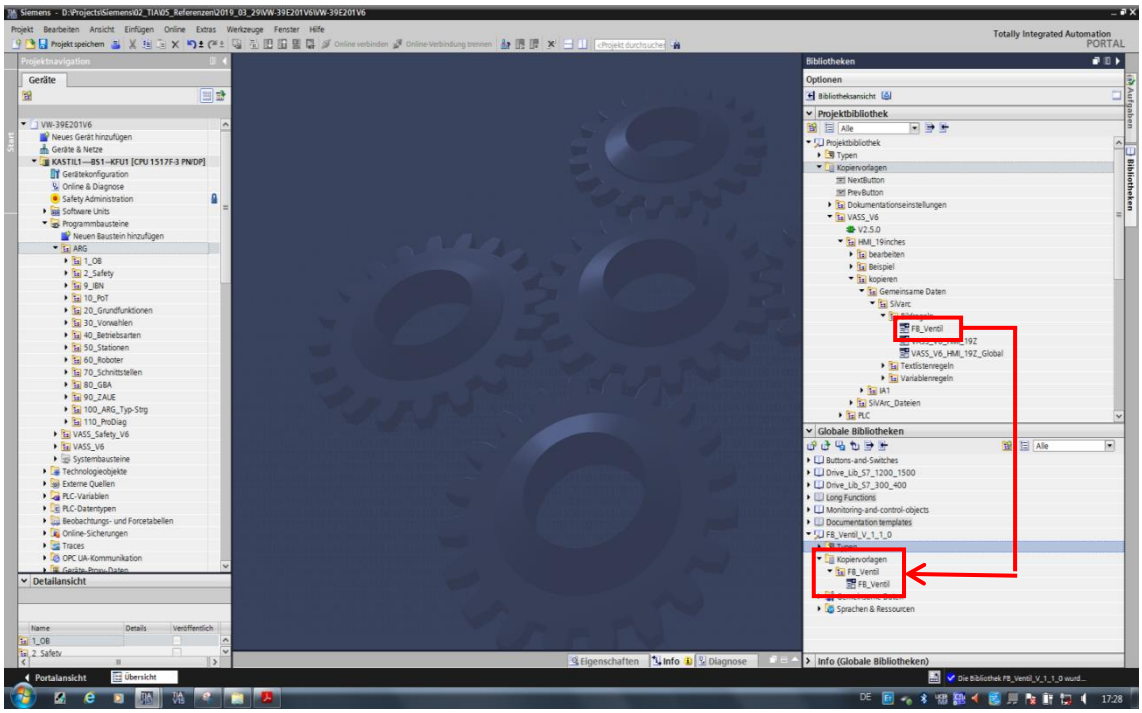


Fig. 8.17 Copying master copies to the global library

If all master copies are copied to the global library, they can be saved and archived in accordance with Chapter 8.1.3 "Saving and archiving the global library".

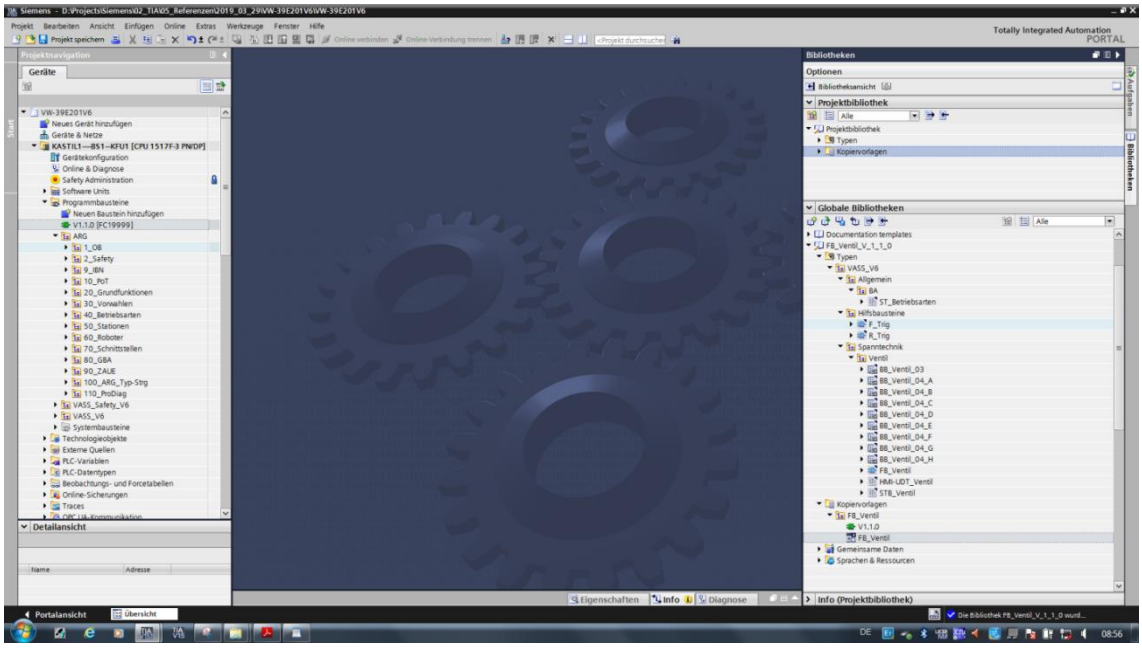


Fig. 8.18 Completed global library

8.1.3 Saving and archiving a global library

The global library must be saved before archiving.

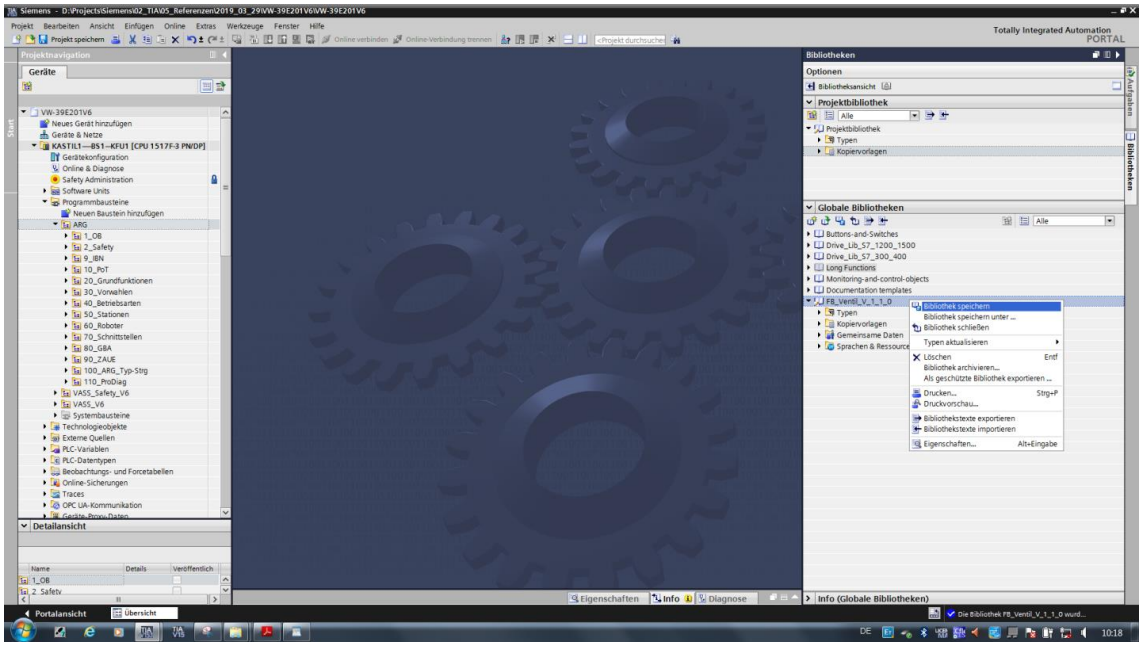


Fig. 8.19 Saving a global library

The global library can then be archived.

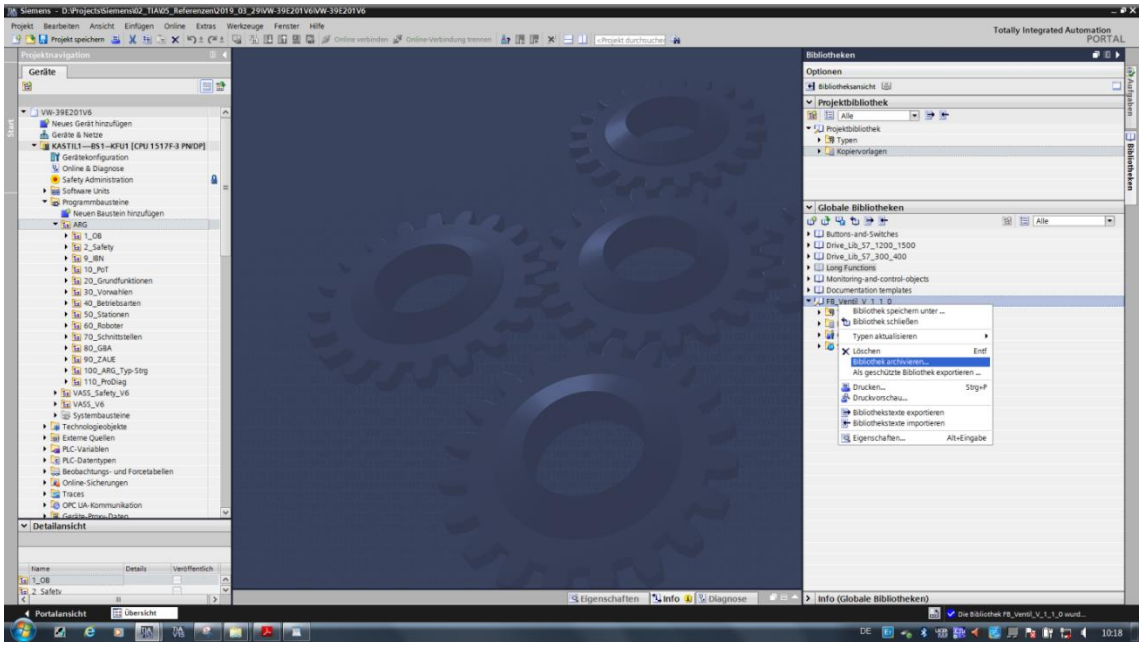


Fig. 8.20 Archiving a global library

In the dialog that opens, the check marks "Archive as compressed file" and "Add date and time to the target name" must be set.

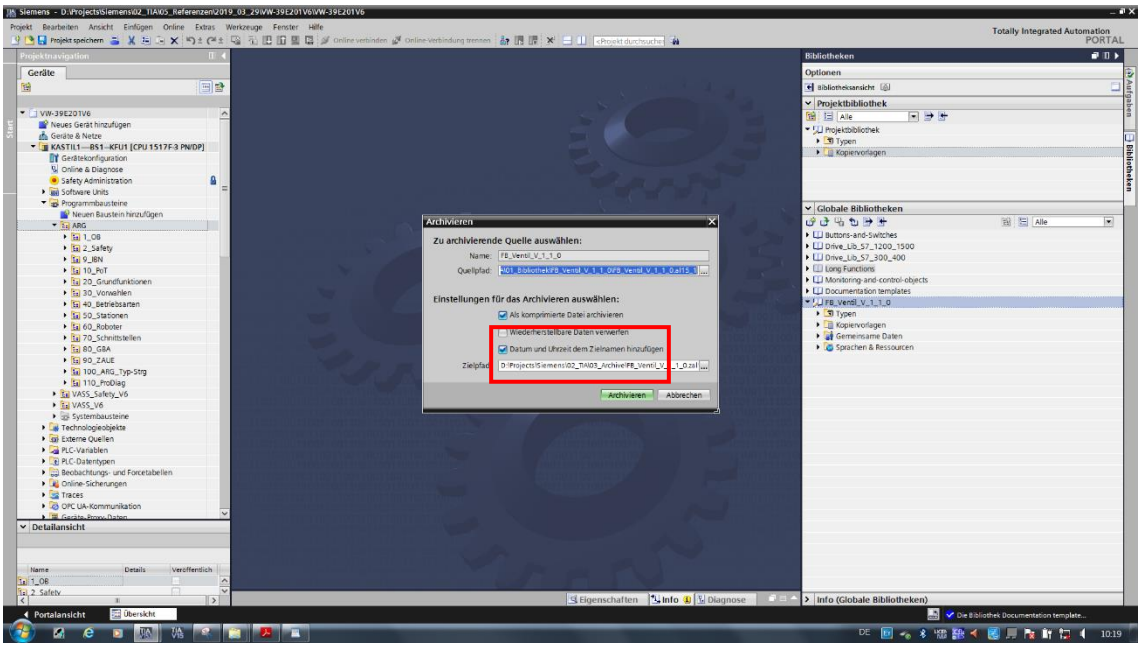


Fig. 8.21 Dialog window "Archive“ of the global library

8.2 Block help

The block help created in accordance with Chapter 7 "Block help" must be provided in both Word and PDF format.

8.3 Assignment list

If a module uses FB I/O parameters with address ranges, the declaration of the I/O symbols must be supplied as a separate spreadsheet in an Excel file.

The design and format should be provided in the same manner as the "VWSymbolikV06.xls".

List of figures

Fig. 2.1 PROFINET device name ... 16
Fig. 2.2 Example of PROFINET device name ... 17
Fig. 2.3 Cell production ... 18
Fig. 2.4 Line production ... 19
Fig. 2.5 E-Plan ... 20
Fig. 2.6 General location name ... 21
Fig. 2.7 Name of function elements ... 22
Fig. 3.1 Elements in a VASS module ... 28
Fig. 3.2 Overview of the VASS standard ... 29
Fig. 3.3 Basic project ... 30
Fig. 3.4 Device catalog ... 32
Fig. 3.5 Library ... 33
Fig. 3.6 Default tag table ... 34
Fig. 3.7 Principle of data exchange between PLC and HMI ... 35
Fig. 3.8 Relationship between instance data block and detailed screen ... 36
Fig. 3.9 Overview of the project texts ... 37
Fig. 3.10 SiVArc screen generation overview ... 38
Fig. 3.11 VASS_Online_help ... 39
Fig. 3.12 Hardware configurations ... 40
Fig. 3.13 Operating and visualization information ... 40
Fig. 4.1 VASS data ... 41
Fig. 5.1 VASS basic project ... 46
Fig. 5.2 GSDML file ... 47
Fig 5.3 Calling the application "Manage general station description files (GSD)" ... 47
Fig. 5.4 Select and install "GSDML file" ... 47
Fig. 5.5 Setting up a device in HW Config (1) ... 48
Fig. 5.6 Setting up a device in HW Config (2) ... 48
Fig. 5.7 Select the desired catalog under "Devices & Networks" ... 49
Fig. 5.8 Create new catalog profile ... 50
Fig. 5.9 Add elements ... 50
Fig. 5.10 Assign catalog profile name (1) ... 51
Fig. 5.11 Assign catalog profile name (2) ... 51
Fig. 5.12 End creation of catalog profile ... 52
Fig. 5.13 Open settings for TIA ... 52
Fig. 5.14 TIA settings export dialog ... 53
Fig. 5.15 Selecting a catalog profile for export ... 53
Fig. 5.16 Acknowledging successful export ... 54

Fig. 5.17 Storage path of export file 54
Fig. 5.18 Creating symbols 56
Fig. 5.19 General structures 57
Fig. 5.20 Create UDT 58
Fig. 5.3 Add new block 59
Fig. 5.3 Pop-up "Add new block" 60
Fig. 5.23 Properties (General) 61
Fig. 5.3 Properties information 61
Fig. 5.3 Add new group 62
Fig. 5.26 Rename new group 63
Fig. 5.27 Moving a block in the project tree 64
Fig. 5.28 Create library type 65
Fig. 5.29 Pop-up window informing you that the block is being compiled 66
Fig. 5.30 "Add type" dialog 66
Fig. 5.31 Creating folders in the project library 67
Fig. 5.32 Moving a block in the project library 68
Fig. 5.33 Display "Editor is write-protected" 69
Fig. 5.34 Display "edit the type" 69
Fig. 5.35 Release dialog type version 70
Fig. 5.36 Block: Input parameters 71
Fig. 5.37 Block: Output parameters 72
Fig. 5.38 Block: Input and output parameters 72
Fig. 5.39 Block: Static tags 73
Fig. 5.40 Block: Static tags Visu 1 74
Fig. 5.41 Block: Static tags edge formation 75
Fig. 5.42 Block: Static tags ON delay 75
Fig. 5.43 Block: Static tags OFF delay 75
Fig. 5.44 Block: Other static tags 76
Fig. 5.45 FB_Softkey1 77
Fig. 5.46 Blank spaces between operators and operands 79
Fig. 5.3 Input in binary format 82
Fig. 5.48 Change journal 84
Fig. 5.3 Explanation of the configuration bits 85
Fig. 5.50 Block: Check VISU buttons 86
Fig. 5.51 PC_ACTIVE 86
Fig. 5.52 Block: Program description 87
Fig. 5.53 # edge evaluation positive edge 89
Fig. 5.54 # edge evaluation negative edge 89
Fig. 5.55 # ON delay (1) 90
Fig. 556 ON delay (2) 90

Fig. 5.3 Switch off delay ... 90
Fig. 5.58 Block: Evaluation of input parameters ... 90
Fig. 5.59 Block: Error evaluation ... 91
Fig. 5.60 Block: Forming a general fault ... 91
Fig. 5.61 Block: Reset fault ... 91
Fig. 5.62 Block: Interconnecting output parameters ... 92
Fig. 5.63 Evaluation of messages ... 93
Fig. 5.64 Generation of messages in the "Static" part of the block header ... 94
Fig. 5.65 ID for the ZAÜ system ... 94
Fig. 5.66 Creating an "operand" supervision ... 95
Fig. 5.67 Settings under the "Supervisions" tab in the Inspector window ... 95
Fig. 5.68 Selection of categories ... 96
Fig. 5.69 Selection of subcategories ... 96
Fig. 5.70 Structure of a nibble ... 98
Fig. 5.71 Color code table ... 99
Fig. 5.72 Color animation ... 100
Fig. 5.73 Animation navigation ... 101
Fig. 5.74 Bits that must be set in the ""DB_ARG".VisuSS" ... 102
Fig. 5.75 Programmed out animation for "Station" ... 102
Fig. 5.76 Structure block ... 103
Fig. 5.77 Interconnecting structures ... 104
Fig. 6.1 Create an HMI-UDT ... 109
Fig. 6.2 Selection of layers ... 110
Fig. 6.3 Frame ... 111
Fig. 6.4 Gutter frame ... 111
Fig. 6.5 Button Dsw ... 112
Fig. 6.6 Button Dws ... 112
Fig. 6.7 Button AnimFlaeche ... 113
Fig. 6.8 Button AnimFlaeche ... 113
Fig. 6.9 Button AnimFlaeche ... 114
Fig. 6.10 Button AnimText ... 114
Fig. 6.11 Button AnimText ... 115
Fig. 6.12 Button AnimText ... 115
Fig. 6.13 Button lock ... 116
Fig. 6.14 Button button ... 116
Fig. 6.15 Button button ... 117
Fig. 6.16 Button button ... 117
Fig. 6.17 Button pointer ... 118
Fig. 6.18 Reduction of cycle time ... 119
Fig. 6.19 "Shifting and masking" in the VASS standard ... 119

Fig. 6.20 Documentation ... 122
Fig. 6.21 Folders in the project library ... 123
Fig. 6.22 Master copies folder in the project library ... 124
Fig. 6.23 Layout for detailed screen "SEW_AMA_Bin" ... 125
Fig. 6.24 SiVArc properties in "Layout__Detail_SEW_AMA_Bin" ... 125
Fig. 6.25 Detailed screen for "SEW_AMA_Bin" after generation ... 126
Fig. 6.26 Example of an animated text ... 127
Fig. 6.27 Specification of object name ... 127
Fig. 6.28 Faceplate for detailed screen with display of all layers ... 128
Fig. 6.29 Detailed screen with hiding of the layers 23 (texts) and 30 (buttons) ... 128
Fig. 6.30 Setting of layer using a text field as an example ... 129
Fig. 6.31 Supply of text fields in the faceplate for detailed screen ... 130
Fig. 6.32 Example declaration of the visualization interface "HMI-UDT_SEW_AMA_Bin" ... 131
Fig. 6.33 Example assignment of the "HMI-UDT_SEW_AMA_Bin" in the module FB ... 131
Fig. 6.34 Representation of visualization interface in the instance DB ... 131
Fig. 6.35 Example of the associated HMI-UDT for the faceplate "BB_Detail_SEW_AMA_Bin" ... 132
Fig. 6.36 Example of internal tag directory ... 132
Fig. 6.37 Switching dwVisuWerte to internal tags ... 132
Fig. 6.38 Faceplate properties "BB_SEW_AMA_Bin_10" for the root screen ... 133
Fig. 6.39 Events faceplate "BB_SEW_AMA_Bin_10" for calling the detailed screen from the root screen ... 133
Fig. 6.40 Events faceplate "FB_DrehtFU_2S" ... 134
Fig. 6.41 "FB_DrehtFU_2S" with the SiVArc event "ActivateScreen" ... 135
Fig. 6.42 The faceplate for "FB_DrehtFU_2S" is placed in the station screen ... 135
Fig. 6.43 The call of the drive "FB_SEW_AMA_Bin" as an instance in the PLC in the station ... 136
Fig. 6.44 The faceplate "SEW_AMA_Bin_10" is placed on the root screen by SiVArc ... 136
Fig. 6.45 The detailed screen "110060DT1AE1_Detail_SEW_AMA_Bin" is generated in the folder "Detail" by SiVArc ... 137
Fig. 6.46 Master copies folder in the project library ... 137
Fig. 6.47 SiVArc properties in "Layout_106_Detail_InlineMessen" ... 138
Fig. 6.48 Detailed screen "Detail_InlineMessen_120080LM1" after generation ... 139
Fig. 6.49 Faceplate properties "BB_InlineMessen_10" for the root screen ... 140
Fig. 6.50 Calling "FB_InlineMessen" as an instance in a station ... 141
Fig. 6.51 Placement of "BB_InlineMessen_10" on the root screen ... 142
Fig. 6.52 Placement of "BB_InlineMessen_03" on the station screen ... 142
Fig. 6.53 Detailed screen "Detail_InlineMessen_120080LM1" ... 143
Fig. 6.54 Technology groups of the storage location for the faceplates ... 144
Fig. 5.3 Type-specific texts ... 145
Fig. 6.56 Select languages ... 146
Fig. 6.57 Select languages ... 146

Fig. 6.58 Instance-specific texts 147
Fig. 6.59 Multilingual texts 148
Fig. 6.60 Faceplate BB_Ventil_03 Name - HMI-UDT and texts 149
Fig. 6.61 Faceplate BB_Ventil_04_A - name, HMI-UDT and texts 149
Fig. 6.62 Definition of tag rules tag group hierarchy with "SiVArc_" 150
Fig. 6.62 Generated tag names with"_HMI-UDT" and data type HMI-UDT 151
Fig. 6.62 Faceplate interface automatically supplied with data type HMI-UDT 151
Fig. 6.65 Plug-Ins SiVArc properties in screen templates - Screen 003 152
Fig. 5.3 Plug-Ins SiVArc properties in screen templates - Screen 004 152
Fig. 6.67 Parameter assignment at the FB_Ventil parameter "Maske" 153
Fig. 6.68 SiVArc screen rules for valve faceplate in screens 003 and 004 154
Fig. 5.3 SiVArc generated screen 004 filled with faceplates 154
Fig. 5.3 Creation of new SiVArc text lists as master copies 155
Fig. 6.71 Storage of new text list 156
Fig. 6.72 Creation of new text list rules 156
Fig. 6.73 "Screen rules" editor 157
Fig. 6.74 Screen rules and division of the columns 158
Fig. 6.75 Screen rules editor 159
Fig. 6.76 Create new screen rules 159
Fig. 6.77 Screen rule: Assigning a name 160
Fig. 6.78 Screen rule: Assigning a program block (module FB) 160
Fig. 6.79 Screen rule: Assigning a program block (module FB) 160
Fig. 6.80 Screen rule: Assigning a generation template (screen) 161
Fig. 6.81 Screen rule: Assigning a generation template (screen) 161
Fig. 6.82 Screen rule: Assigning operation 162
Fig. 6.83 Condition at FB_Ventil for layout page 004 162
Fig. 6.84 Combining screen rules to form a rule group 162
Fig. 6.85 Assign name of rule group 163
Fig. 6.86 Create copy template 163
Fig. 6.87 FB642 FB_S7G_Control with parameter S7G_Control_Ext Var.A 165
Fig. 6.88 Result "Sequential control 110010DT1" 165
Fig. 6.89 FB642 FB_S7G_Control with parameter S7G_Control_Ext Var.B 166
Fig. 6.90 Result "110010DT1#AS_DB_OFF_SQ " 166
Fig. 6.91 FB_Vorwahl TextFields (1) 167
Fig. 6.92 FB_Vorwahl TextFields (2) 167
Fig. 6.93 Result in screen Vorwahlen_013 168
Fig. 7.1 Storage location 169
Fig. 7.2 Screenshot of "FB_Druckluft_VW370" 173
Fig. 7.3 HIP valve terminal 177
Fig. 7.4 HIP Pneumatics 178

Fig. 7.5 Faceplates in the project library ... 182
Fig. 7.6 Operation / visualization information ... 183
Fig 7.7 Create PDF file ... 184
Fig. 7.8 Checkmarks must be removed ... 184
Fig. 7.9 Checkmarks were removed ... 185
Fig. 8.1 Folder structure (transfer of a VASS module) ... 186
Fig. 8.2 Folder structure using FB_Ventil as an example ... 186
Fig. 8.3 Structure types project library ... 187
Fig. 8.4 Structure of master copies project library ... 188
Fig. 8.5 Creation of a global library ... 189
Fig. 8.6 Update types in the library ... 189
Fig. 8.7 Selection of library ... 190
Fig. 8.8 Result: Updating types ... 191
Fig. 8.9 Copying master copies to the global library ... 192
Fig. 8.10 Completed global library ... 193
Fig. 8.11 Structure types project library ... 194
Fig. 8.12 Structure of master copies project library ... 195
Fig. 8.13 Creation of a global library ... 196
Fig. 8.14 Update types in the library ... 196
Fig. 8.15 Selection of library ... 197
Fig. 8.16 Result ... 197
Fig. 8.17 Copying master copies to the global library ... 198
Fig. 8.18 Completed global library ... 199
Fig. 8.19 Saving a global library ... 200
Fig. 8.20 Archiving a global library ... 200
Fig. 8.21 Dialog window "Archive“ of the global library ... 201

List of tables

Tab. 2.1 Explanation of terms "Contactors" 24
Tab. 2.2 Explanations of terms "Commands" 24
Tab. 2.3 Explanation of terms "Operator controls" 25
Tab. 2.4 Explanations of terms "Display elements" 26
Tab. 2.5 Explanation of terms "Operator controls and display elements" 26
Tab. 3.1 Engineering tools for VASS standard with Windows 7 / 64-bit or Windows 10 / 64-bit 27
Tab. 3.2 Languages in the faceplates 37
Tab. 4.1 Designation system "Group / Brand VW" 42
Tab. 4.2 Designation system "Brand" 42
Tab. 4.3 Designation system "plant" 42
Tab. 4.4 Reserve, SEAT and SKODA 43
Tab. 4.5 VWN, Porsche and Audi 44
Tab. 4.6 Designation system "VW sites" 45
Tab. 5.1 Syntax of slice function 80
Tab. 5.2 Example of slice access 81
Tab. 5.3 Definition of terms for the signaling system 97
Tab. 5.4 Abbreviation for entries 97
Tab. 5.5 Bit distribution in "dwVisuWerte" 100
Tab. 6.1 Colors in WinCC Advanced 106
Tab. 6.2 Indication of sizes for faceplates 107
Tab. 6.3 Indication of sizes for different areas of faceplates 108
Tab. 6.4 Position in double word (dwVisuValues) 120
Tab. 6.5 "Moving and masking" for buttons 120
Tab. 6.6 "Move" by 8 bits and "mask" with 255 120
Tab. 6.7 "Move" by 16 bits and "mask" with 255 121
Tab. 6.8 "Move" by 20 bits and "mask" with 15 121
Tab. 6.9 "Move" by 24 bits and "mask" with 15 121
Tab. 6.10 "Move" by 28 bits and "mask" with 15 121
Tab. 7.1 Input parameters 174
Tab. 7.2 Configuration bits 174
Tab. 7.3 Output parameters 175
Tab. 7.4 Input and output parameters 175
Tab. 7.5 External tag references 175
Tab. 7.6 Error messages 181
Tab. 7.7 Visualization 181
Tab. 8.1 Abbreviations / Glossary 214

Abbreviations / glossary

	Abbreviation	Full text	Comment
A	ARG	Workgroup	
B	BB	Faceplate	
	BEB	Conditional block end	
C	CommCon	Communication control	
	CRC	Cyclic Redundancy Check	Safety checksum
D	DB	Data block	
	DI	Digital input	
	DO	Digital output	
E	ES	Engineering system	
	ET	Electronic terminal	
F	F-…	Fail-safe	
	FB	Function block	
	FBD	Function block diagram	
	FC	Function	
	FM	Completion report	
	FRG	Enable	
	FU	Frequency converter	
G	GBA	Large-scale display	
	GSD	Device master data	
H	HF	Hotfix	
	HMI	Human Machine Interface	
I			
J			
K			

L	LAD	Ladder diagram	
	LSP	Load voltage	
	LVU		Operator safety light barrier
M			
N			
O	OB	Organization block	
	OEM	Original Equipment Manufacturer	
	OP	Operator panel	
P	PC	Personal computer	
	PDA	Production data display	
	PDE	Production data entry	
	PG	Programming device	
	PLC	Programmable logic controller	
	PLS	Operator protection laser scanner	
	PN	PROFINET	
	POF	Polymer Optical Fiber	
	PoT	Production without parts	
	PS	Process system	
Q			
R	RLO	Result of logic operation	
	RO	RemoteOperate	
	Rob	Robot	
	RT	Runtime	
S	SCL	Structured Control Language	Structured text
	SCS	Sequential control system	
	SDB	System Data Block	
	SFM	Report system errors	

	SiVArc	Siemens Visualization Architect	The system analyzes STEP 7 user projects and automatically generates visualizations from them for WinCC Advanced. The SiVArc system is subdivided into two software packages: - Update Manager - Screen Generator
		SiVArc	The SiVArc generates the WinCC Advanced visualization from a STEP 7 user project and the rule sets. - Automatic generation of WinCC Advanced visualization projects - Flexible selection of HMI device type - Preview of the auto-import result - Creating multilingual message texts - Automatic access to the Volkswagen message text database - Automatic import into WinCC Advanced - Detailed instructions of manual post-editing - Convenient analysis of the result through plain text log files
	SK	Protective circuit	
	SK	Sequencers	
	SP	Service pack	
	SSL	System state list	
	ST	Station	
	STB	Interconnecting structure	
	STL	Statement list	
T	TIA	Totally Integrated Automation	

U	UBA	Sub-mode	
		Update Manager (SiVArc)	The Update Manager distributes all the data required for the generation of the WinCC Advanced projects to the Engineering System. - Storage and update of all project data - Libraries - Documentation - Reference projects - SiVArc setup - Rule set - Data distribution that is easily configurable - Distribution of the STEP 7 block help
	UDT	User-defined data type	
V	V	Version	
	VASS	Volkswagen AUDI SEAT SKODA	VASS standard: Software standard for Volkswagen AUDI SEAT SKODA (currently for body manufacturing)
W	WCF	WinCC Advanced	
X			
Y			
Z	ZAÜ / ZAU / ZAUE	Central System Monitoring	

Tab. 8.1 Abbreviations / Glossary