SIEMENS

Application Example• 06/2015

Configuration Examples for HMI Faceplates

SIMATIC Comfort Panels, Runtime Advanced and WinCC (TIA Portal)

https://support.industry.siemens.com/cs/ww/en/68014632

Warranty and Liability

Note

The application examples are not binding and do not claim to be complete with regard to configuration, equipment or any contingencies. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for the correct operation of the described products. These application examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time and without prior notice. If there are any deviations between the recommendations provided in this Application Example and other Siemens publications – e.g. Catalogs – the contents of the other documents shall have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of fundamental contractual obligations ("wesentliche Vertragspflichten"). The compensation for damages due to a breach of a fundamental contractual obligation is, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these application examples or excerpts hereof is prohibited without the expressed consent of Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a productspecific newsletter. For more information, visit <u>http://support.industry.siemens.com</u>.

Table of Contents

Warr	anty and	Liability	2		
Preface					
1	Task		5		
	1.1 1.2	Overview Overview of the automation task	5 5		
2	Solutio	٦	7		
	2.1 2.2 2.3 2.4	Solution overview General notes on the application Hardware and software components used Supported operator panels and controllers	7 8 9 . 10		
3	Configu	iration and Settings	. 12		
	3.1 3.1.1 3.1.2 3.1.3 3.1.4 3.1.5 3.1.6 3.1.7 3.1.8 3.1.9 3.2 3.2.1 3.2.2	Configuration examples Sample_01: assigning a configurable function to a button Sample_02: creating a configurable "IO field" Sample_03: configuring font size at objects Sample_04: Switching objects invisible Sample_05: Configuring a color change at an object Sample_06: faceplate with user data type (HMI UDT) Sample_07: Faceplate with a PLC data type Sample_08: user authorizations at the faceplate Sample_09: Using a trend view in a faceplate Application Example Example 1 Example 2	. 12 . 13 . 16 . 23 . 26 . 29 . 39 . 46 . 50 . 55 . 55		
4	Operati	ng the Application	. 59		
	4.1 4.2 4.3	Overview "Example 1" picture "Example 2" picture	. 59 . 60 . 61		
5	Further	notes, tips and tricks	. 62		
6	Links &	Literature	. 63		
7	History.		. 63		

Preface

Objective of this application

The objective of this application is to provide you with an aid for generating faceplates.

If you are not yet familiar with the topic of "Faceplates", please view the documentation "Basic knowledge on the topic of HMI faceplates".

The documentation "Basic knowledge on the topic of HMI faceplates" discusses the following topics.

- Releasing the faceplate
- Updating faceplates
- HMI user data type (HMI UDT)
- PLC data types

Main topics of this application

Using several practice-relevant examples, the individual configuration steps are described in detail. The examples are available as compressed download.

Validity

- Software version as of WinCC Comfort V13 SP1
- All SIMATIC HMI operator panels that support faceplates (see chapter 2.4 "Supported operator panels).

Information on the function scope of the SIMATIC HMI operator panels is available in the Online Help of WinCC (TIA Portal) or FAQ <u>http://support.automation.siemens.com/WW/view/en/40227286</u>.

1.1 Overview

1 Task

1.1 Overview

Introduction

In plant and mechanical engineering in particular, the requirements for a uniform operator control and monitoring concept are high in order to guarantee short familiarization times or high operating security for the later operator of the plant.

A uniform operator control and monitoring concept is also important for the maintenance of plants. This can reduce down times during failures, maintenance works, or a plant expansion.

In serial machine construction, configured functions, such as drive on/off, switchover automatic/manual mode etc. are frequently used. These functions generally only need to be adjusted to the respective machine and its control tags. Using preprogrammed objects reduces engineering times and costs.

1.2 Overview of the automation task

The figure below provides an overview of the automation task.

You have different customers using the identical functions for controlling a pump or a drive.

Figure 1-1



1.2 Overview of the automation task

Description of the automation task

You have several customers whose plant areas have similar functions.

The plant sections consist of several drives for pumps and fans. It shall be possible to operate each drive via one "control panel".

The following functions shall be possible via the "control panel".

- Output of the current status of the drive
- Switchover manual/automatic mode
- Manual mode on/off
- Output e.g. of the current flow rate

Due to the different requirements for the HMI operator panel, it must be possible to use HMI operator panels with different display sizes.

Problem description

The following requirements for the operating concept must be fulfilled:

- Uniform appearance for all plants.
- Can be used several times even for different customers.
- Central changeability of the control panel.
- Automatic update of the control panels used in the project after performed modifications or expansions.
- Created control panel shall be available as a "library" so it can be used by different persons in the engineering process.
- A connection to a data block for data exchange between the control panel and the connected PLC controller. The data block of the PLC controller has a respective given structure.

2.1 Solution overview

2 Solution

2.1 Solution overview

For realizing the tasks described in chapter 1 it is advisable to use "faceplates".

Faceplates consist of a compilation of display and operator objects which you can manage and modify centrally in a library. Storing the faceplates in a library enables calling and using them easily in different projects.

The following figure shows the editor of a faceplate in progress. Don't be irritated by the "Lines", which in the beginning seem rather confusing. In the further course of the application, the correlations are explained step by step.

Figure	2-1
--------	-----

Project library	▶ Types ▶ 02_Face	plate_Pump [i	in progress]						- 7	∎×
✓ 🕆	B	IUSA	<u>A</u> ±≣±1	± <u>♣</u> ± <u>⊿</u> ±	≣±-3	: 📑 ± 🖾 ±	바비비	± 1 ± ⊲	't⊴± ⊑,±'	•
Fan: 00	0	X								^
Ð	Fault Ready for use		Cont	trol						≡
00	000 m3/h		pan	iel	· · · · · · · · · · ·					
Auto	Manual									
Manual	on Manual off									
4										~
Properties	Events Tags	Scripts	Text lists	Graphic li	ts Te:	kts Lang	juages			
Name > Button_Aut > Button_Mai > Button_Mai > Button_Mai > Button_X > EA-Field_Di > Rectangle_ > Rectangle	o_Mode nual_Off nual_Off nual_On splayValue_m3/h impNr. Fault Fault						ne Properties_f	aceplate	Туре	Interface
 Rectangle_ Rectangle_ Rectangle_ Symbol_Pu 	Frame Ready_to_use upper_lable mp	•								l

2.2 General notes on the application

2.2 General notes on the application

Advantages

This application offers the following advantages:

- The step-by-step instructions for creating a faceplate and the practical application examples reduce the familiarization time.
- Faceplates increase the efficiency. Storing the faceplate in a library enables calling and using them easily in different projects.
- Configuration instruction and example projects Nine practice-relevant examples are described in the documentation. The described faceplates are available ready configured in the project on hand.
- Avoiding errors through a simple connection to the STEP 7 control program.

Topics not covered in this application

This application does not include a description of

- the used Comfort Panel. This document describes only the steps that are necessary for this application.
- how to create, for example, a network connection between operator panel and configuration computer or PLC controller.
- the installation of the SIMATIC software WinCC (TIA Portal).

Basic knowledge of these topics is assumed.

Assumed knowledge

- Basic knowledge how to work with and how to operate a Comfort Panel is expected.
- Skills for using WinCC (TIA Portal).

2.3 Hardware and software components used

2.3 Hardware and software components used

The application was created with the following components:

Hardware components

Table 2-1

Component	Qty	Article number	Note
CPU 1515-2 PN	1	6ES7 515-2AM00-0AB0	Optional: the PLC is used for simulation purposes (data exchange between control panel and PLC).
TP1200 Comfort Panel	1	6AV2124-0MC01-0AX0	Alternatively: See Chapter 2.4.

Standard software components

Table -22

Component	No.	Article number	Note
WinCC Comfort V13 SP1	1	6AV2101-0AA01-0AA5	Alternatively: WinCC Advanced V13 SP1 or higher. WinCC Professional V13 SP1 or higher.
STEP 7 Professional V13 SP1	1	6ES7822-1AA01-0YA7	Optional: For the example configuration. Alternatively: higher version.

Sample files and projects

The following list includes all files and projects that are used in this example.

Table 2-3

Component	Note
68014632_Faceplates_instruction_CODE_v20.zip	This zip file contains the WinCC Comfort and STEP 7 project.
68014632_Faceplates_instruction_DOKU_v20_e.zip	Documentation

2.4 Supported operator panels and controllers

2.4 Supported operator panels and controllers

The following tables list those devices which support the faceplates.

Runtime

Table 2-4

No.	Operator panel
1.	WinCC Runtime Advanced
2.	WinCC Runtime Professional (this documentation only discusses the panels and WinCC Runtime Advanced)

Comfort Panels

Table 2-5		
No.	Operator panel	
3.	KP400 Comfort	
4.	KTP400 Comfort	
5.	KP700 Comfort	
6.	TP700 Comfort	
7.	KP900 Comfort	
8.	TP900 Comfort	
9.	KP1200 Comfort	
10.	TP1200 Comfort	
11.	KP1500 Comfort	
12.	TP1500 Comfort	
13.	TP1900 Comfort	
14.	TP2200 Comfort	

Panels

Table 2-6

No.	Operator panel
1.	TP 277
2.	OP 277

2.4 Supported operator panels and controllers

Mobile Panels

Table 2-7	
No.	Operator panel
1.	Mobile Panels
2.	Mobile Panel 277
3.	Mobile Panel 277 IWLAN V2
4.	Mobile Panel 277F IWLAN V2
5.	Mobile Panel 277F IWLAN (RFID Tag)

Multi Panels

Table 2-8	3
-----------	---

No.	Operator panel			
1.	MP 277			
2.	MP 377			

SIMATIC controllers

— Т:	ah	P	2-9	

No.	SIMATIC Controller
1.	S7-300/400 controllers
2.	S7-1200 controllers
3.	S7-1500 controllers

3 Configuration and Settings

In this chapter, details on the configuration of a faceplate are described. The examples are based on one another.

3.1 Configuration examples

Using typical application examples, it is shown how to realize the described tasks with a faceplate.

The examples are based on one another. The function scope increases from example to example.

General notes on the example projects

An S7-1500 controller was used here as controller.

Apart from example "Sample_07" (faceplate with a PLC data type), the examples can also be used for an "S7-300/S7-400" controller.

The figure below shows the library with the stored faceplates.

Figure 3-1

✓ Project library	boy
🞽 🗄 All 🔽	<u> </u>
▼ 💭 Project library	·7
▼ 🔄 Types	A
📑 Add new type	lin .
D2_Faceplate_Sample_S7_1500	ati
HMI_UDT_Pump_S7_1500	ons
Sample_01_S7_1500	
Sample_02_S7_1500	232
Sample_03_S7_1500	
Sample_04_S7_1500	ayo
Sample_05_S7_1500	ut
Sample_06_S7_1500	
Sample_07_57_1500	*
Sample_08_S7_1500	5
Sample_09_S7_1500	str
UserDataType_Faceplate	lict
Master copies	ion
	0

3.1.1 Sample_01: assigning a configurable function to a button

Assigning a configurable function to a button

With two buttons it shall be possible for the user to assign any system function at the "Press" event, such as "SetBit" for example.

Figure 3-2

•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	
•	1	•	•	•	•	•	•	•	•	•	-21	•	12	•	•	•	•	•	•	•	•	•	21	
1	i.				C)	n				I	÷	1				C)f	ff				L	1
•	ь	_	_	_	_	_		_	_	_	4	÷	Ŀ,	_	_	_	_	_		_	_	_	4	
•	·	·	·	·	·	·	·	·	·	·	·	÷	÷	·	·	·	·	·	·	·	·	·	·	•

Table 3-1

No.	Action
1.	Inserting objects
	 Open the faceplate editor. Add two buttons to the workspace. Assign a label to the buttons in the inspector window in "Events > General". In this example: Text -> On Text -> Off Assign a name to the buttons in the inspector window in "Events > Miscellaneous", which the button can be identified by. In this example: Button_1 -> Button_On Button_2 -> Button_Off
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages Name Dynamizat. Name Dynamizat. Appearance Configure a connection with drag-and-drop. Security Miscellaneous Security Miscellaneous
	Button_On 🖳 Properties 🗓 Info 🕦 🗓 Diagnostics 💿 🗉 🥆
	Properties Animations Events Property list Miscellaneous General Object Appearance Name: Button_On Layout Text format Flashing Miscellaneous Security

No.	Action
2.	Assigning properties
	 Select the "Events" tab in the configuration area (1). In the "Contained objects" list you open the properties of the button (2). Select the event for which the function shall be performed (3). Via drag&drop you configure a connection between the selected "Event" and the "Interface" list (4). In this example: Button_Off -> "Press"
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages
	Name • Button_Off • Button_Off Click Deactivate • Button_On • Button_On • Button_On • Button_On
	 In the "Interface" list of the newly created property, you assign a name you can later use for identifying the property at the faceplate (1). In this example: Press -> Event_Button_Off
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages
	Name Button_Off Activate Change Click Deactivate Press Release Button_On
	 Repeat the previously described steps for the second button. The required settings are thus completed.
3.	Releasing the faceplate
	For the performed configurations to become effective, the created faceplate must be released.
	The faceplate is updated.

No.	Action
4.	Faceplate view: Assigning a function
	Call up the created faceplate in a picture.
	The subsequent figure shows the newly created faceplate with the two buttons " On " and " Off " (1).
	In the "Properties > Events" tab of the faceplate (2), the two created properties "Event_Button_Off" and "Event_Button_On" are displayed (3). From the dropdown list (4), you can assign the familiar functions such as "SetBit" to the events "Event_Button_Off" and "Event_Button_On".
	Note: The name of the properties was selected so they can easily be assigned to the buttons.
	Sample_01_1 Sample_01_1 Sample_01_1
	Properties Interface Animations Events (2)
	Image: Event_Button_Off Event_Button_On 3 3 Tag (Input/output) <add function=""></add>

3.1.2 Sample_02: creating a configurable "IO field"

Creating a configurable "IO field"

Process values shall be input and output at two IO fields. It shall be possible to assign the HMI tags to the IO fields. Figure 3-3



No.	Action
1.	Inserting objects
	Open the faceplate editor.
	Add two IO fields to the workspace.
	 In the inspector window in "Events > Miscellaneous" you assign a name to the IO field which the button can be identified by. In this example:
	I/O field_1 -> IO_FIEld_CurrentSpeed
	 In the inspector window, you make further settings for the IO fields such as settings at the
	"Layout".
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages
	Name Dyn
	Image: Second secon
	Po_Field_SpeedSetting Configure a Configure a Connection with Field_SpeedSetting
	The second secon
	IO_Field_SpeedSetting 📴 Properties 🚺 Info 🖳 Diagnostics 💿 🗖 🗖 🤝
	Properties Animations Events
	Property list Miscellaneous
	General Deligation
	Appearance Users IO Sidd GrandSetting
	Characteristics Iname: IO_Field_SpeedSetting
	Text format
	Flashing
	Limits
	Miscellaneous Security

No.	Action
2.	Assigning properties
	 Select the "Properties" tab (1)
	 In the "Contained objects" list you open the properties of the IO field (2).
	• Open the "General" entry from the list and mark the "Process value" property (3).
	 Via drag&drop you configure a connection between the selected "Property" and the "Interface" list (4) Drag the mouse pointer to an existing estagent. (The name of the estagent can be
	specified/modified individually and may deviate from the names used in the picture).
	In this example: IO Field CurrentSpeed -> "Process value" → "Properties Faceplate"
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages
	Name Type VIO_Field_CurrentSpeed Froperties_Faceplate
	2 Appearance
	Flashing General
	2 Display format Configure a
	Field length
	E Show leading zeros
	5 Layout
	Miscellaneous
	Security Text
	IO_Field_SpeedSetting
	 In the "Interface" list of the newly created property, you assign a name you can later use for identifying the property at the faceplate (1)
	In this example:
	Process_value -> Processvalue_CurrentSpeed Select the data type from the drop-down list (2)
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages
	Name Type ✓ IO_Field_CurrentSpeed ✓ Properties_Faceplate
	Appearance ProcessValue_CurrentSpeed @ Int Characteristics
	Flashing
	Display format
	Process value Image: Constraint of the second sec
	E Show leading zeros 123
	O ► Layout
	Limits Miscellaneous
	Security
	► IO_Field_SpeedSetting

No.	Action
	Repeat the previously described steps for the second IO field.
	The required settings are thus completed.
3.	Releasing the faceplate
	For the performed configurations to become effective, the created faceplate must be released.
	The faceplate is updated.
4.	Faceplate view; assigning an HMI tag
	Call up the created faceplate in a picture.
	The subsequent figure shows the created faceplate with the two IO fields (1). In the "Properties > Interface" tab of the faceplate (2), the two created properties "ProcessValue_CurrentSpeed" and "ProcessValue_SpeedSettings" are displayed (3). An HMI tag can be assigned to the "ProcessValue_CurrentSpeed" and "ProcessValue_SpeedSettings" properties via the drop-down list (4).
	Note: The name of the properties was selected so they can easily be assigned to the IO fields.
	Sample_02_1 Properties Linto & Diagnostics I I
	Name Static value Dynamization
	✓ Properties_Faceplate
	ProcessValue_CurrentSpeed N 3 M1_CurrentSpeed II. 4

3.1.3 Sample_03: configuring font size at objects

Configurable font size at the object

A faceplate shall be used for operator panels of different sizes. The font size of the objects used in the faceplate (two IO fields, two buttons and texts) shall be specified here.

No.	Action
1.	Inserting objects
	 Open the faceplate editor. Enter the required objects such as buttons, IO fields etc., into the workspace. For this purpose, look at the table section 1 in chapters <u>3.1.1</u> and <u>3.1.2</u>. Additionally, add two text fields into the workspace. In the inspector window in "Events > Miscellaneous" you assign a name to the text field which the button can be identified by. In this example: Text field_1 -> Text_Field_CurrentSpeed Text field_2 -> Text_Field_SpeedSetting In the inspector window, you make further settings for the individual objects, if necessary such as settings at the "Layout". For optical reasons, the objects are integrated into a frame (1).
	Current Speed: 000000 Speed Setting: 000000 Image: Control of the set of t
	Properties Events Tags Scripts Text lists Graphic lists Texts Image: Configure a connection with drag-and-drop. Name Name Type Name Type Name
	Button_On 📴 Properties 🚺 🗓 Diagnostics 🖬 🖃 🥆
	Properties Animations Events
	Property list General Appearance Design Layout Text format Flashing Miscellaneous

No.	Action
2.	Assigning a property for the first object
	 Select the "Properties" tab (1). In the "Contained objects" list you open the properties of the first object (2). Open the "Text" entry from the list and mark the "Font" property (3). Configure a connection between the selected "Property" and the "Interface" list via drag&drop (4). Drag the mouse pointer to an existing category. (The name of the category can be specified/modified individually and may deviate from the names used in the picture). In this example: Button_Off -> "Text" -> "Font type" → "Properties_Objects"
	Properties Events Tags Scripts Text lists Graphic lists Texts Image: Configure a connection with drag-and-drop. Name Dynamizations • Button_Off • Appearance • Properties_Objects • <
	 In the "Interface" list of the newly created property, you assign a name you can later use for identifying the property at the faceplate (1). In this example: Font -> Font_AllObjects The data type is given automatically (2). Properties Events Tags Scripts Text lists Graphic lists Texts of Font Properties. Dynamications Properties Events Tags Scripts Text lists Graphic lists Texts of Font Properties. Dynamications Properties Events Tags Scripts Text lists Graphic lists Texts of Font Properties. Dynamications Properties Events Tags Scripts Text lists Graphic lists Texts of Font Properties. Dynamications Properties Events Tags Scripts Text lists Graphic lists Texts of Font Properties. Dynamications Properties Events Tags Scripts Text lists Graphic lists Texts of Font Properties. Dynamications Dynamications Properties. Dynamications Dynamications Properties. Dynamications Properties. Dynamications Properties. Dynamications Dynamications Properties. Dynamications Dynamications Properties. Dynamications Dynamications Properties. Dynamications Properties. Dynamications Dynamications Properties. Dynamications Properties. Dynamications Properties. Dynamications Properties. Dynamications Properties. Properties. Dynamications Properties. Dynamications Properties. Dynamications Properties. Properties. Dynami

Action
Assigning further properties to the "Interface" list
 Note: You can assign several properties from the "Contained objects" list to one property from the "Interface" list. In the "Contained objects" list you open the properties of the next object (1). Open the "Text" entry from the list and mark the "Font" property (2). Configure a connection between the selected "Property" and the "Interface" list via drag&drop. Drag the mouse pointer to the already existing property (3). In this example: Button_On -> "Text" -> "Font type" → "Font_AllObjects"
Imperiates Events rags Stripts rext ists Graphic lists rexts Languages Imperiates Design Imperiates Imperiates

No.	Action
4.	Releasing the faceplate
	For the performed configurations to become effective, the created faceplate must be released.
	The faceplate is updated.
5.	Faceplate view; assigning the text format
	Call up the created faceplate in a picture.
	The subsequent figure shows the created faceplate (1).
	In the "Properties > Interface" tab of the faceplate (2) the created property "Font_AllObjects" is
	The font can be assigned to the "Font_AllObjects" property via the selection box (4).
	In this example:
	Note: The name of the tags was selected so the stored function can be easily recognized
	The name of the tags was selected so the stored function can be easily recognized.
	Current Speed: 000000 Speed Setting: 000000
	On Off
	· · · · · · · · · · · · · · · · · · ·
	Sample_03_1 2 Sa
	Properties Interface Animations Events
	Name Static value Dynamization
	✓ Properties_Objects 3
	Font_AllObjects M Tahoma, 16px 4

3.1.4 Sample_04: Switching objects invisible

Switching objects invisible

A faceplate cannot be switched invisible directly via a system function. It shall be possible to switch all objects in a faceplate invisible via a function/tag.

No.	Action
1.	Inserting objects
	 Open the faceplate editor. Enter all objects such as buttons, IO fields, text fields etc., into the workspace you require. The structure of the faceplate from chapter <u>3.1.3</u> is used as a template for this example.
	Current Speed:000000Speed Setting:000000
	Cff
2.	Creating tags
	 In the "Interface" list of the configuration area, you add a new tag and specify data type "Bool" in the "Type" column via the dropdown menu. In this example: Name of the property: FaceplateVisible.
	Type: Bool.
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages Name Dynamizatio Name Type Name Type Name Dynamizatio Name Type Properties_Faceplate Button_CloseFaceplate Sutton_Off FaceplateVisible Bool Interface Framework_01 Configure a connection with drag-and-drop. 1 1 Interface Io_Field_CurrentSpeed Text_Field_SpeedSetting Interface Interface Interface

No.	Action
3.	Creating an animation
	 In the workspace you select all objects (1). In the inspector window you open the "Properties > Animations" tab. Add the "Visibility" animation in the area navigation at "Display" (2). Selecting a tag: The properties/tags from the "Interface" list are displayed to you via the symbol (3). Select the "FaceplateVisible" tag from the list. Selecting the tag automatically connects all marked object with the "Visibility" animation. You can see this in the "graphic" area at the newly created connections (4). Specify the area in which the faceplate shall be visible. In this example: Area: 0 to 0. Visibility: invisible.
	Current Speed: 000000 Speed Setting: 000000 On Off
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages Name Dynamizations Name Type Name Type Button_Off Button_On Properties_Faceplate Bool FaceplateVisible Bool Properties_Field_CurrentSpeed Io_Field_CurrentSpeed Io_Field_SpeedSetting 4 Io_Field_SpeedSetting Io_Field_SpeedSetting
	Framework_01 Supervises Diagnostics I = -
	Visibility
	Overview Display Add new a Visibility Tag: Tag: FaceplateVisible To: To: Single bit Visibility Visibility Visible Visible V
	The required settings are thus completed.
4.	Releasing the faceplate
	For the performed configurations to become effective, the created faceplate must be released.
	The faceplate is updated.

No.	Action						
5.	Faceplate view; tag assignment						
	Call up the created faceplate in a picture.						
	The subsequent figure shows the created faceplate (1). In the "Properties > Interface" tab of the faceplate (2) the created tag "FaceplateVisible" is displayed (3). Via the dropdown menu you can assign an HMI tag to the "FaceplateVisible" tag via which the						
	Taceplate can be switched invisible (4). Note: The name of the tags was selected so the stored function can be easily recognized.						
	Current Speed: 000000 Speed Setting: 000000 On Off						
	Sample 04.1 (2) Sample 04.1 (2)						
	Sample_04_1 2 Properties Linfo Diagnostics						
	Name 3 Static value Dynamization ✓ Properties_Faceplate FaceplateVisible M1_FaceplateVisible						

3.1.5 Sample_05: Configuring a color change at an object

Configuring a color change at an object

Pressing a button shall change the background color of the pressed button.

No.	Action
1.	Inserting objects
	 Open the faceplate editor. Enter all objects such as buttons, IO fields, text fields etc., into the workspace you require. The structure of the faceplate from chapter <u>3.1.4</u> is used as a template for this example.
	Current Speed: 000000 Speed Setting: 000000
2.	Creating tags
	 In the "Interface" list of the configuration area you add two new tags and via the dropdown menu you specify data type "Bool" in the "Type" column (1). In this example: Name of the property: Status_Button_On Status_Button_Off Type: Bool.
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages
	Name Dynamizatio > Button_Off Name > Button_On Properties_Faceplate > Framework_01 Status_Button_Off > 10_Field_CurrentSpeed Imega > Text_Field_CurrentSpeed Imega > Text_Field_SpeedSetting Imega

No.	Action
3.	Creating an animation
	 Select the "On" button in the workspace (1). In the inspector window you open the "Properties > Animations" tab. Add the "Appearance" animation in the navigation in "Display" (2). Selecting a tag: The properties/tags from the "Interface" list are displayed to you via the symbol (3). Select the "Status_Button_On" tag from the list. Selecting the tag automatically creates a connection between the property of the "On" button (Button_On) in "Contained objects" list and the "Status_Button_On" tag in the "Interface" list. You can see this in the "oraphic" area (4).
	 Specify the area and the background color (5).
1	Value 0: gray
1	Value 1: green
	Current Speed: 000000 Speed Setting: 000000 Image: Optimized Setting: 000000 Image: Optimized Setting: 000000
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages
	Name Type > Button_Off > Properties_Faceplate > Button_Off Bool > Framework_01 > Status_Button_Off > IO_Field_CurrentSpeed > Text_Field_SpeedSetting
	Button_On Properties Linfo 😧 Diagnostics 🗈 🖃 🗸
	Properties Animations Events Appearance Appearance Overview Tag Tag connections Name: Status_Button_On Add new ani Address: ScreenModules Appearance Multiple bits Single bit Image Range Background color Program 0 Add news 0,0,0 Image 0,255,0 Image Add news
	 Repeat the steps for the second "Off" button. The required settings are thus completed.

No.	Action					
4.	Releasing the faceplate					
	For the performed configurations to become effective, the created faceplate must be released.					
	The faceplate is updated.					
5.	Faceplate view; tag assignment					
	Call up the created faceplate in a picture.					
	The subsequent figure shows the created faceplate (1).					
	In the "Properties > Interface" tab of the faceplate (2) the created tags "Status_Button_Off" and					
	Via the dropdown list you can assign an HMI tag to the "Status Button Off" and					
	"Status_Button_On". Depending on the value of the tag, the background color of both buttons changes (4)					
	Note: The name of the tags was selected so the stored function can be easily recognized.					
	Current Speed: 000000 Speed Setting: 000000 1					
	Sample 05 1 (2) Properties Hinfo V Diagnostics					
	Properties Interface Animations Events					
	Name Static value Dynamization Transition Image: Comparison for the state of					
	Status_Button_Off N 3 Status_ButtonOff 4					
	Status_Button_On N Status_ButtonOn II					

3.1.6 Sample_06: faceplate with user data type (HMI UDT)

Faceplate with a user data type (HMI UDT)

Note A description on the subject of "HMI UDT" is given in the document "Basic knowledge on the topic of HMI faceplates".

Task

The drive for a fan shall be operated and monitored via a configured control panel. Due to the number of functions, the tag is connected via an HMI UDT.

The following functions and status information shall be executed and evaluated.

Status information

- "Automatic mode" state active
- "Manual mode" state active
- "Manual mode ON" state active
- "Manual mode OFF" state active
- "Ready to use" state
- "Error" state
- "OK" state

Functions

- Output of fan number
- Output value of the current flow rate
- Switchover to automatic mode
- Switchover to manual mode
- Manual mode "On"
- Manual mode "Off"

Faceplate view

Figure 3-4

Fan:	000		X
Ŕ	尸;	ault Ready for use	
	0000	m3/h	
Aut	to	Manual	
Manua	al on	Manual o	off

Note As of WinCC (TIA Portal) V13 SP1 and when using an S7-1200 or S7-1500 controller, you do no longer need to use a PLC data type.

Creating data block, user data type and HMI tag for the data exchange

No.	Action					
1.	Creating a data block					
	Fo	r realizii	ng the described Sample_	06, the following	, data blo	ck "DB10" was created. The
	dis	played	data block contains the st	ructure for three	faceplate	es.
	1	ءِ 🛫 🗧	• F 14 6 6 6 6 6	= 🕼 😤		
		DB10	FacePlate HMI UDT			
		Nam	1e	Data type	Offset	Start value
	1		Static			
	2		FacePlate_01	Struct	0.0	
	З		New Kes DI	DIPT	17.0	
	4		 FacePlate_02 	Struct	16.0	
	5		Status_Automatic_Mode	Bool	0.0	false
	6		Status_Manual_Mode	Bool	0.1	false
	7		Status_Manual_On	Bool	0.2	false
	8		 Status_Manual_Off 	Bool	0.3	false
	9		Res05	Bool	0.4	false
	10		Status_Ready_To_Use	Bool	0.5	false
	11		Status_Fault	Bool	0.6	false
	12		Res08	Bool	0.7	false
	13		Status_Pump	Int	2.0	0
	14		Res10	Int	4.0	0
	15		Pump_Number	Int	6.0	0
	16		Pump_Flow_Rate	Int	8.0	0
	17		Set_Automatic_Mode	Bool	10.0	false
	18		Set_Manual_Mode	Bool	10.1	false
	19	- •	Set_Manual_On	Bool	10.2	false
	20		Set_Manual_Off	Bool	10.3	talse 3
	21	- -	FacePlate 03	Struct	32.0	
	23	-	New Res. 05	Dint	44.0	0
		-	-			
	_					
	(1)	Str	ucture for the first facenlat	`		
	(י) (2)	Str	ucture (onened) for the se	cond facentato		
	(2)	500	ucture (opened) for the se			
	(3)	Str	ucture for a third faceplate	;		
	NO	te:	laak aan ha sutsudsul d			
	In	e data k	block can be extended at v	WIII.		

No.	Action							
2.	Configuring a user data type							
	Docu data t	ment "Basic knowlec	lge on the topic of H	MI facep	ates" describ	es how to	o configure a	user
		.) [
	The f	iqure below shows th	e created user data	type with	the respectiv	ve taos. T	he structure	
	(struc	ture) corresponds to	that of the "DB10" of	data bloci	κ.			
	Pleas	e ensure that the us	er data block is "rel	eased" af	ter creating it			
					ter ereating t	•		
	Mana	af the user date tur						
	Name	e of the user data typ		Imp				
	HN	/II user data types						
	Con	nmunication driver: SIMATIC	\$7 1500	v	Device family	Panels / Wi	inCC Runtime Adva	nced 👻
		Name	Communication driver	Data type	Length	Start value	Offset	Bit offset
	ŧ	Status_Automatic_Mode	SIMATIC S7 1500	Bool	1		0	0
	E	Status_Manual_Mode	SIMATIC S7 1500	Bool	1		0	1
	Ē	Status_Manual_On	SIMATIC S7 1500	Bool	1		0	2
	Ē	Status_Manual_Off	SIMATIC S7 1500	Bool	1		0	3
	Ē	Res05	SIMATIC S7 1500	Bool	1		0	4
	Ē	Status_Ready_To_Use	SIMATIC S7 1500	Bool	1		0	5
	E	Status_Fault	SIMATIC S7 1500	Bool	1		0	6
	E	Res08	SIMATIC S7 1500	Bool	1		0	7
Status_Pump SIMATIC S7 1500 Int 2 2					2	0		
	E	Res10	SIMATIC S7 1500	Int	2		4	0
Pump_Number SIMATIC S7 1500 Int 2				6	0			
	Ē	Pump_Flow_Rate	SIMATIC S7 1500	Int	2		8	0
	Ē	Set_Automatic_Mode	SIMATIC S7 1500	Bool	1		10	0
	Ę	Set_Manual_Mode	SIMATIC S7 1500	Bool	1		10	1
	E	Set_Manual_On	SIMATIC S7 1500	Bool	1		10	2
	(F	Set_Manual_Off	SIMATIC S7 1500	Bool	1		10	3

No.	Action
3.	Creating an HMI tag
	 The data exchange between the PLC and the user data type used in the faceplate requires an HMI tag. Create an HMI tag with a controller connection. The name can be selected by the user (1). Select the user data type used in the faceplate as the data type (2). In this case "HMI_UDT_Pump_S7_1500".
	01 TagTable Esceptate
	Name A Data type Connection PLC
	Assigning a start address to the HMI tag The "DB10" data block displayed in the table section "1" contains the structure for three faceplates. The start address for the first faceplate is "DB10.DBX0.0" The start address for the second faceplate is "DB10.DBX16.0"
	- The start address for a third faceplate is "DB10.DBX32.0".
	Data type Address Connection PLC HMI_UDT_Pump_S7_1500 V MDB10.DBX0.0 IMI_connection_2 PLC
	Operand identifier: DB DB number: 10 Operand type: DBX
	Address: 0 Bit number: 0

No.			Action			
4.	View of t	he HMI tag with assigned user dat	a type			
	The symbol for reducing / expanding the visible entries enables viewing the complete structure of the created HMI tags (1). The structure of the assigned user data type was automatically assigned to HMI tag "DataExchange_HMI_UDT_Faceplate_01".					
\bigcirc	🥩 🖻	🗄 🔁				
Ľ	01_T	agTable_Faceplate				
	No	me 🔺	Data type	Address		
	 • 	DataExchange_HMI_UDT_Faceplate_01	HMI_UDT_Pump_S7_1500 V 0.0.3	%DB10.DBX0.0		
	- 10	Status_Automatic_Mode	Bool	%DB10.DBX0.0		
		Status_Manual_Mode	Bool	%DB10.DBX0.1		
		Status_Manual_On	Bool	%DB10.DBX0.2		
		Status_Manual_Off	Bool	%DB10.DBX0.3		
		Res05	Bool	%DB10.DBX0.4		
		Status_Ready_To_Use	Bool	%DB10.DBX0.5		
		Status_Fault	Bool	%DB10.DBX0.6		
		Res08	Bool	%DB10.DBX0.7		
		Status_Pump	Int	%DB10.DBW2		
	- 💷 =	Res10	Int	%DB10.DBW4		
	- 💷 =	Pump_Number	Int	%DB10.DBW6		
		Pump_Flow_Rate	Int	%DB10.DBW8		
		Set_Automatic_Mode	Bool	%DB10.DBX10.0		
	💷 -	Set_Manual_Mode	Bool	%DB10.DBX10.1		
		Set_Manual_On	Bool	%DB10.DBX10.2		
		Set_Manual_Off	Bool	%DB10.DBX10.3		
	- 	DataExchange_HMI_UDT_Faceplate_02	HMI_UDT_Pump_S7_1500 V 0.0.3	%DB10.DBX16.0		
		DataExchange_HMI_UDT_Faceplate_03	HMI_UDT_Pump_S7_1500 V 0.0.3	%DB10.DBX32.0		

Integrating a user data type (HMI UDT) into a faceplate

No.		Action				
1.	Inse	ting objects				
	• (• ;	Dpen the faceplate editor. Enter all objects such as buttons, IO fields, text fields etc., into the workspace required according to the plant description.				
	The	picture shows the completed faceplate.				
	Fan: 000 X Fault Image: Constraint of the second se					
		0000 m3/h				
		Auto Manual				
	ľ	1anual on Manual off				
2.	Adjusting the names of the inserted objects					
	Adjust the names of the used objects in the " Properties > Contained objects " tab so you car easily identify these objects. For this purpose, look at the previously described chapters <u>3.1.1</u> to <u>3.1.5</u> . The figure below shows the edited names in the "Contained objects" list.					
		Name Dynamizations				
		Button Auto Mode				
		Button Manual Mode				
		Button Manual Off				
		Button Manual On				
		Button x				
		► EA-Field DisplavValue m3/h				
	cts	► EA-Field PumpNr.				
	bje	Rectangle Fault				
		Rectangle Frame				
	ne	Rectangle Ready to use				
	Itai	Rectangle_upper_lable				
	ē	▶ Symbol_Pump				
		Textfeld Ready to use				
		Textfield_Fault				
		Textfield Pump				
		▶ Textield_Unit_m3/h				

No.			Action			
3.	Using the user data type	in the faceplate				
	In the " Properties > Inter user data type as a "Type"	ace" tab of the conf via the dropdown n	iguration a nenu.	area y	ou add a tag and s	elect the created
	In this example: Name of the user data typ	e: HMI_UDT_Pu	mp			
				•[
	Name	Tuna				
	- Proportion Eccoplate	туре				
	Properties_Faceplate					
	Property_UD1	HMI_UDI_Pump	▼			
		01_HMI_UDT_Pum UInt UDInt	p_1			
	_	Timer USInt				
	4	HMI UDT Pump	N			
		Dint	2			
	• _	WString				
		Int		ľ		
			- 11			
	The symbol for reducing / the created tags (1). The figure below shows th and the "Interface" list.	e already created co	e entries e	enable ons be	es viewing the comp tween the "Contain	olete structure of ed objects" list
		<u></u>				
	Properties Events Tags	Scripts Text lists G	iraphic lists	Texts	Languages	
	No.	i-				7
	Button Auto Mode	namiz			Properties Faceplate	Туре
	Button_Manual_Mode				 Property_UDT 	HMI_UDT_Pump_S7_1500 \
	Button_Manual_Off Button_Manual_Off	Alt			Status_Automatic_Mode	- Bool
	Button_Manual_On Button_x				Status_Manual_Mode Status_Manual On	- Bool
	EA-Field_DisplayValue_m3/h				Status_Manual_Off	- Bool
	EA-Field_PumpNr.				Res05	- Bool
	Rectangle_Fault Bectangle_Frame				Status_Ready_To_Use Status_Fault	≪⊡ Bool ≪⊡ Bool
	Rectangle_Ready_to_use			-	Res08	a Bool
	Rectangle_upper_lable	•			Status_Pump	🕢 Int
	E Symbol_Pump				Res10 Pump_Number	- Int - Int
	Textfield_Fault				Pump_Flow_Rate	int
	Textfield_Pump				Set_Automatic_Mode	- Bool
	 Textield_Unit_m3/h 				Set_Manual_Mode	- Bool
					Set_Manual_Off	Bool
				-	Faceplate_Visible	- Bool
	< IIII	>				III
	The details for assigning the procedure is principal	ne individual configu y the same as desc	rations an	e des iously	cribed below. 9 for chapters <u>3.1.1</u>	to <u>3.1.5</u> .

No.	Action
5.	Configuring a color change at objects
	 A color change has been stored at the following objects (animation). Button: Button_Auto_Mode Button: Button_Manual_Mode Button: Button_Manual_On Button: Button_Manual_Off Rectangle: Rectangle_Ready_to_use Rectangle: Rectangle_Fault Display: Symbol_Pump
	The procedure is described in chapter <u>3.1.5</u> .
	The allocation can be controlled by means of the graphic representation.
6.	 Creating configurable IO fields The following IO fields can be configured. Input/output field: IO-Field_DisplayValue_m3/h Output field: IO-Field_PumpNr The procedure is described in the chapter <u>3.1.2</u>. The assignment via drag&drop in this case occurs directly to the respective tag at the interface.
	Example: Input/output field: IO-Field Display//alue m3/h
7.	Assigning an event to the buttons
	An event has been configured at the following objects. Button: Button_Auto_Mode Button: Button_Manual_Mode Button: Button_Manual_On Button: Button_Manual_Off Button: Button_x
8	Switching objects invisible
5.	For the enclosed application example, there is the option to switch the used faceplate invisible. The procedure is described in the chapter <u>3.1.4</u> .

No.			Actio	n		
9.	View of the complete	d faceplat	e in the faceplate	editor		
	The "Interface" list only contains the tags for the user data type and the tag for switching the faceplate invisible. In this example: Property_UDT -> HMI_UDT_Pump_S7_1500 Faceplate_Visible -> Bool					
	The "Connections" sho The reference behind i	wn in the g t should no	graphic area are n ow no longer be "c	ow more onfusing	e extensive in this e j" for you.	example application.
	Properties Events Tag	s Scripts	Text lists Graphic	ists Tex	ts Languages	
	Name	Dynamiz			Name	Туре
	Button_Auto_Mode				 Properties_Faceplate 	
	Button_Manual_Mode			\rightarrow	Property_UDT	HMI_UDT_Pump_S7_1500
	Button_Manual_Off				Faceplate_Visible	📶 Bool
	Button_Manual_On Button x					
	EA-Field_DisplayValue_m3/h			/		
	EA-Field_PumpNr.					
	S Rectangle_Fault				_	
	Rectangle_Frame Restangle_Ready to use		·///////		4	
	Rectangle_ready_to_use Rectangle_upper lable					
	Symbol_Pump		/////		-	
	Textfeld_Ready_to_use		////			
	Textfield_Fault		I///			
	 Textfield_Pump Textfield_Unit_m3/b 					
	i icxiicid_oni(_inoin					
10	Releasing the facenta	te				
10.	Releasing the hacepit					
	For the performed cont	igurations	to become effecti	ve, the c	reated faceplate m	ust be released.
	The ferenlets is an date	- -			-	
	The faceplate is update	eu.				



3.1.7 Sample_07: Faceplate with a PLC data type

Note A description on the subject of "PLC data types" is given in the document "Basic knowledge on the topic of HMI faceplates".

Task Description

The task corresponds to the example from the previous chapter 3.1.6.

The drive for a fan shall be operated and monitored via a configured control panel. Due to the number of functions, the tag is connected via a PLC data type.

Status information

- "Automatic mode" state active
- "Manual mode" state active
- "Manual mode ON" state active
- "Manual mode OFF" state active
- "Ready to use" state
- "Error" state
- "OK" state

Functions

- Output of fan number
- Output value of the current flow rate
- Switchover to automatic mode
- Switchover to manual mode
- Manual mode "On"
- Manual mode "Off"

Faceplate view

Figure 3-5

Fan:	000		X
Ŕ	尸;	ault Ready for use	
	0000	m3/h	
Aut	to	Manual	
Manua	al on	Manual o	off

3.1 Configuration examples

Creating PLC data types

No.		Action						
1.	Creating PLC data types							
	The following structure was used for realizing the task. The structure is the same for all faceplates. Document "Basic knowledge on the topic of HMI faceplates" describes how to configure a PLC data type.							
	Not	es						
	•	Ple	ase note that you are using an S	7-1200 or S7-15	00 controller			
	•	The	e PLC data type can be extended	at will.				
		1		こ 11 🕾				
		PLO	UDT Pump S7 1500					
			Name	Data type	Default value	Accessible f	Visible in	
	1	-	Status_Automatic_Mode	Bool 🔳	false			
	2		Status_Manual_Mode	Bool	false			
	з	-00	Status_Manual_On	Bool	false			
	4	-00	Status_Manual_Off	Bool	false			
	5	-00	05_Res.	Bool	false			
	6	-00	Status_Ready_To_Use	Bool	false			
	7	-	Status_Fault	Bool	false		N	
	8	-00	08_Res.	Bool	false			
	9	-00	Status_Pump	Int	0			
	10	-00	10_Res.	Int	0			
	11	-00	Pump_Number	Int	0			
	12	-00	Pump_Flow_Rate	Int	0			
	13	-00	Set_Automatic_Mode	Bool	false		N	
	14	-00	Set_Manual_Mode	Bool	false		N	
	15	-00	Set_Manual_On	Bool	false		N	
	16	-	Set_Manual_Off	Bool	false			

3.1 Configuration examples

Integrating a PLC data type into a data block

No.	Action					
1.	Creating a d	ata block				
	For realizing the described task, the following data block "DB11" was created. The data block contains the data for three faceplates. For each faceplate, a tag was stored and then assigned to the "PLC data type" tag.					
	🥩 🔮 🐁	🛃 🔛 🖶 🖶 🖶 🗮				
	Name	iceriate_rtc_001	Data tura	0#	Start value	
	1 an Valle	atic	Data type		Start value	
	2	FacePlate_01	"PLC_UDT_Pump_S7_1500"	0.0	0	
		EacoPlate 02	PLC UDT Rump S7 1500	16.0	0	
	5	Status Automatic Mode	Rool	0.0	falce	
	5 -	Status_Automatic_Mode	Bool	0.0	falce	
	7 - 10	Status_Manual_Mode	Bool	0.7	false	
	8 📶 🔳	Status_Manual_Off	Bool	0.3	false	
	9 📶 🔹	05 Res.	Bool	0.4	false	
	10 📶 🔹	Status Ready To Use	Bool	0.5	false	
	11 📶 🔹	Status Fault	Bool	0.6	false	
	12 📶 🔳		Bool	0.7	false	
	13 📶 🔳	Status_Pump	Int	2.0	0	
	14 🕣 🔹	10_Res.	Int	4.0	0	
	15 🕣 🔹	Pump_Number	Int	6.0	0	
	16 📶 🔳	Pump_Flow_Rate	Int	8.0	0	
	17 📶 🔳	Set_Automatic_Mode	Bool	10.0	false	
	18 📶 🛛 🔳	Set_Manual_Mode	Bool	10.1	false	
	19 📶 🛛 🔳	Set_Manual_On	Bool	10.2	false	
	20 📶 🔹	Set_Manual_Off	Bool	10.3	false	
	21 📶 💻	New_Res_02	Dint	28.00	0	
	22 📶 🕨 🕨	FacePlate_03	"PLC_UDT_Pump_S7_1500"	32.0		
	23 📶 🗖	New_Res. 03	Dint	44.0	0	
	 (1) Struct (2) Struct (3) Struct Note:	ure for the first faceplate ure (opened) for the second ure for a third faceplate	faceplate			
	The data bloc	ck can be extended at will.				

Integrating a PLC data type into a faceplate

No.		Action
1.	Inserti	ng objects
	• O	pen the faceplate editor.
	• Er	nter all objects such as buttons, IO fields, text fields etc., into the workspace required
	ac	cording to the plant description.
	The at	tached faceplate from example "Sample 06" serves as a template.
	Eant	000 ×
	1 an.	
	/	Fault
	T_	
	7	
		0000 m3/h
		Auto Manual
	D.d.	number Manual off
	I*Ia	
2.	Adjust	ing the names of the inserted objects
	Adiust	the nerves of the used chiests in the "Drementice > Contained chiests" tob so you can
	easilv i	identify these objects.
	For th	is purpose, look at the previously described chapters $3.1.1$ to $3.1.5$.
	The fig	ure below shows the edited names in the "Contained objects" list.
	Ne	ame Dynamizations
		Button_Auto_Mode
	, in the second se	Button Manual Off
		Button Manual On
		Button x
	, in the second s	EA-Field Display/Jalue m3/b
		EA-Field PumpNr
	je –	Pertande Fault
	- E	Pertande France
	- je	Postenale Postu to use
	tair –	Postangle_vener_lable
	- ē	Curched Burge
		Symbol_rump
	•	rextreme_keady_to_use
	•	TextGeld Press
		TextileId_Pump
		Textield_Unit_m3/h

No.		Action				
3.	Using PLC data types in the faceplate					
	In the " Properties > Inte PLC data type as a "Type	face " tab of the configuration area you add a tag and select the created " via the dropdown menu.				
	In this example: Name of the user data types the second seco	pe PLC_UDT_Ump_S7_1500				
	Name Properties_Faceplate Property_PLC_UDT Faceplate_Visible	Type PLC_UDT_Pump_S7_1500 V 0.0.1 Image: MM_UDT_Pump_S7_1500 V 0.0.2				
		IReal PLC_UDT_Pump_57_1500 V 0.0.1 Real Sint Time_Of_Day Timer UDInt UDInt USInt WString				
4.	View of the configured	raceplate				
	After assigning the user of displayed to you via the of The symbol for reducing a the created tags (1). The figure below shows t and the "Interface" list.	ata type in the "Interface" list, the structure of the user data type is reated tag. "expanding the visible entries enables viewing the complete structure of ne already created configurations between the "Contained objects" list				
	Properties Events Tags St Name > Button_Auto_Mode > Button_Manual_Off > Button_Manual_Off > Button_Manual_Off > Button_Manual_Off > Button_Manual_Off > Button_Kanual_On > EA-Field_PumpNr. > Rectangle_Ready_to_use > Textfield_Ready_to_use > Textfield_Ready_to_use > Textfield_Pump > Textfield_Unit_m3th	ipts Text lists Graphic lis Texts Languages Improve the second				

No.	Action
5.	 Details for assigning the individual configurations The configurations of the following points corresponds to those of the previous examples ("Sample_06") and are not further discussed here. Configuring a color change at objects (Link) Creating configurable IO fields (Link) Assigning an event to the buttons (Link) Switching objects invisible (Link)
6.	View of the completed faceplate in the faceplate editor The "Interface" list only contains the tags for the PLC data type and the tag for switching the faceplate invisible. In this example: Property_UDT -> PLC_UDT_Ump_S7_1500 Faceplate_Visible -> Bool The "Connections" shown in the graphic area are now more extensive in this example application. The reference behind it should now no longer be "confusing" for you.
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages Name
7.	Releasing the faceplate For the performed configurations to become effective, the created faceplate must be released.
	The faceplate is updated.



3.1.8 Sample_08: user authorizations at the faceplate

Assigning the user authorization to a faceplate

Switching from automatic mode to manual mode as well as the subsequent operation in manual mode shall only be performed by an authorized person (service staff).

A user administration is required for this.

Table 3-11 No. Action 1. Inserting objects Open the faceplate editor. ٠ Enter all objects such as buttons, IO fields, text fields etc., into the workspace you require. • The attached faceplate from example "Sample_07" serves as a template. 000 Fan: Х Fault Ready for use m3/h 0000 Manual Auto Manual on Manual off 2. Adjusting the names of the inserted objects Adjust the names of the used objects in the "Properties > Contained objects" tab so you can easily identify these objects. For this purpose, look at the previously described chapters 3.1.1 to 3.1.5. The figure below shows the edited names in the "Contained objects" list. Dynamizations Name Button_Auto_Mode Button_Manual_Mode Button_Manual_Off Button_Manual_On Button x EA-Field_DisplayValue_m3/h ▶ EA-Field_PumpNr. Rectangle_Fault Rectangle_Frame Rectangle_Ready_to_use Rectangle_upper_lable Symbol_Pump ۰. Textfeld_Ready_to_use Textfield_Fault Textfield_Pump

Textield_Unit_m3/h

No.	Action
3.	Assign a property to the first button in the "Interface" list
	 Select the "Properties" tab (1). In the "Contained objects" list you open the properties of the "Button_Auto_Mode" button (2). Open the "Security" entry from the list and mark the "Authorization" property (3). Via drag&drop you configure a connection between the selected "Property" and the "Interface" list (4). Drag the mouse pointer to an existing category. (In this example a further category named "SecurityLevel" was created).
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages Name Uppartize Importive Importive
	 In the "Interface" list of the newly created property, you assign a name you can later use for identifying the property at the faceplate (1). In this example: Authorization -> Security_Button The data type is given automatically (2).
	The required settings for the first button are thus completed.

No.	Action		
4.	Assign further buttons in the "Interface" list		
	 Note: You can assign several properties from the "Contained objects" list to one property from the "Interface" list. However, you can also store a separate "Property" in the "Interface" list for each button. In this example, the properties of all five buttons are assigned to a "Property" in "Interface" list. In the "Contained objects" list you open the properties of the button (1). Open the "Security" entry from the list and mark the "Authorization" property (2). Configure a connection between the selected "Property" and the "Interface" list via drag&drop. Drag the mouse pointer to the already existing "Security_Button" property (3). 		
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages		
	Name Dynamizations > Button_Auto_Mode 1 > Appearance > Security_Button > Design > Security_Button > Flashing > General > Layout > Security_Button > Miscellaneous 2 Authorization 10 Authorization 2 Allow operator control 10 > Text > Button_Manual_Off > Button_Manual_On > > Et4-Field_Displad/alue_m3(b) V		
	Repeat the previously described steps for all listed elements. The picture below shows the complete assignment of the properties.		
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages Name Dynamiza Dynamiza Image: Construct a structure of the structure of		
	The required settings are thus completed.		

No.	Action		
5.	Releasing the faceplate		
	For the performed configurations to become effective, the created faceplate must be released.		
	The faceplate is updated.		
6.	Faceplate view; assigning the authorization		
	Call up the created faceplate in a picture.		
	The subsequent figure shows the created faceplate (1). In the "Properties > Interface" tab of the faceplate (2) the created property "Security_Button" is displayed (3). The respective authorization can be assigned to the "Security Button" property via the selection box (4).		
	In this example: Operating		
	Note: The name of the tags was selected so the stored function can be easily recognized.		
	The name of the tags was selected so the stored function can be easily recognized.		
	Security_Button 3 N P Operating 4		

3.1.9 Sample_09: Using a trend view in a faceplate

Using a trend view in a faceplate

The current flowrate of air over a period of one minute shall additionally be output via the trend view in the control station.

Figure 3-6

Fan:	000)	
P	F	Fault	14 12 10
	0000	m3/h	6
Au	to	Manual	4-
Manu	al on	Manual off	0 10:58:39 AM 10:59:09 AM 10:59:39 AM 12/31/2000 12/31/2000 12/31/2000

The created control station in "Sample_07" serves as a template.

Table 3-12

No.		ction	
1.	Duplicate control stations		
	 Open the project library and right-click to s "Sample_07". 	t the faceplate you wish to	o duplicate. In this case
	 Select the "Duplicate type" item from the oppoperties of the new faceplate, such as the 	ext menu. A window opens ew name, for example. Co	where you can specify the nfirm the entries with "OK".
	▼ 💭 Project library	3	
	▼ 🔄 Types	A	
	Add new type		
	01_Faceplate_Sample_S7_300	lat	
	02_Faceplate_Sample_S7_1500	0n	
	HMI_UDT_Pump_S7_1500		
	Sample_01_57_1500		
	Sample_02_57_1500	1 2	
	Sample_03_57_1500	ayr	
	Sample_04_57_1500	Suff Control of Contro	
	Sample_05_S7_1500		
	Sample_07_57 <u>_1500</u>	*	
	Sample_08_57 Open		
	Sample_09_S7 Edit type	stri	
	UserDataType_I Duplicate type	lat.	
	Master copies Update	lon l	
	Assign version	S	
	Library management		
	Cut Ctrl+X		
	💼 Copy Ctrl+C	as	
	Paste Ctrl+V	ks	
	X Delete Del		
	Rename F2		
		li l	
	Print Ctrl+P Print preview	rarie	
1	Roperties Alt+Enter		
l			



No.		Action	
3.	Display of the trend view		
	Select die trend view and in the "Properties" of the inspector window, you create the view of the trend		
	view as before. With r	egards to the example, the following points were adapted:	
	- Table		
	- Time axis		
	Value avia loft/	/right	
		ngnt	
	Ready for	/ use 14	
	0000		
	0000 m3/		
	Auto Ma		
	Manual on Man		
		10:58:39 AM 10:59:39 AM 10:59:39 AM	
	· ·L	12/31/2000 n 12/31/2004	
	Trend_View [Trend view]	🖸 Properties 🔄 🗓 Info 🕦 💟 Diagnostics 🚽 💷 🔽	
	Properties Animation	s Events Texts	
	Property list		
	Tread	Left value axis	
	Irend	Show left Yavis	
	Appearance Border		
	Lavout	Range	
	Text format	Auto-size	
	Toolbar		
	Button border	Axis start: U	
	Button fill pattern	Begin, tag:	
	Table	Axis end: 15	
	Table header border	End, tag:	
	Table header fill pattern 🖣	Show help lines	
	Time axis	Help line value: 50	
	Left value axis	The print to be.	
	Right value axis	label	
	Style/Designs		
	Miscellaneous	Label: 🗹	
	Wiscenarieous	Marks: 🗹	
		Label length: 2	
		Increment: 1	
		Marks: 2	

No.	Action
4.	Animation "Visibility"
	Trend view shall be shown and hidden together with the other objects. To do this, the "Visibility" animation is used here.
	 Select the trend view in the workspace (1). In the inspector window, you open the "Properties > Animations" tab (2). Add the "Visibility" animation in the area navigation at "Display" (3). Selecting a tag: The properties/tags from the "Interface" list are displayed to you via the symbol. Select the "Faceplate_Visible" tag from the list. Specify the area in which the trend view shall be visible. In this example: Area: 0 to 0. Visibility: invisible.
	Auto Manual 4 2 Manual on Manual off 0 10:58:39 AM 10:59:09 AM 10:59:39 AM 12/31/2000 12/31/2000 12/31/2000
	Properties Events Tags Scripts Text lists Graphic lists Texts Languages
	Properties Animations Events Texts
	Overview I Tag connections Display Add new animation Visibility No Range From: 0 To: 0 Single bit 0 Single bit 0 Compared Single Single Sing
	The required settings are thus completed.

No.	Action		
5.	Releasing the faceplate		
	For the performed configurations to become effective, the created faceplate must be released.		
	The faceplate is updated.		
6.	Faceplate view		
	Configuring trends		
	Call up the created faceplate in a picture.		
	The subsequent figure shows the new created faceplate. In the "Properties > Interface" tab of the faceplate, the created property " Trend " is displayed (1).		
	 To configure the trend view, click on the selected button (2) in the "Static value" column. The "Trend" window opens. In the window you perform the configuration of the trends as before. 		
	Fan: 000 X Fault Fault Ready for use 14 12 10 14 12 10 8 4 2 0 0000 m3/h 10:59:39 AM 10:58:39 AM 10:59:39 AM 12/31/2000 12/31/2000 3 12/31/2000 12/31/2000 3 * rend * 100 • • * 100 • • * * * 100 • • * * * 100 • • * * * * * * * * * * * * * * * * * * * * * * * * <		

3.2 Application Example

3.2 Application Example

3.2.1 Example 1

In picture "001_Application > Topic_001.1" of the attached HMI configuration, an example is listed of how the created faceplate can be integrated into a plant picture.

Task

In a ventilation system two fan drives shall be operated and monitored via the HMI operator panel. The control elements for controlling the fan drives shall only be visible or called on demand.

Solution

The following picture shows the control elements (faceplates) in a plant picture.

The used "functions" of the faceplates are described in detail for chapter 3.1.

Clicking on one of the "fan symbols" (1) calls the respective control element or closes it again.

The control element can also be closed again via the "X" button at the control element.



Figure 3-7

The call of the faceplates for the "Topic_001.1" picture is described on the following page.

3.2 Application Example

Configuration

In the attached HMI project, the used faceplate is stored as an example (Sample_06).

Faceplate as well as user data type are available in the "Libraries" task card.

No.	Action		
1.	Configuring the faceplate		
Add the "Sample_06" faceplate into the plant picture. In the "Properties > Interface" tab of the faceplate both tags "Faceplate_Visible" and "Property_UDT" are displayed (1). Assign the respective HMI tag to this tag (2). In this example: Faceplate_Visible -> Faceplate_02_Visible Property_UDT -> DataExchange_FP_02_PLC_HMI_UDT			
	Fan: 000 X Manual on		
	Fault Ready for use		
	0000 m3/h		
	Auto Manual		
Manual	Manual on Manual off		
	Sample 06 S7 1500 2 [Screen module instance] [Sample 06 S7 1500 V 0.0.5]		
	Properties Interface Animations Events Texts		
	Name Dynamization		
	Properties_Faceplate Faceplate Visible Faceplate 02 Visible		
	Property_UDT DataExchange_HMI_UDT_Faceplate_02		

3.2 Application Example



3.2 Application Example

3.2.2 Example 2

The "001_Application > Topic_001.2" picture contains the second example.

In the second example, a faceplate is displayed in connection with a trend view. Faceplate "Sample_09" is used as a template.

A PLC program is stored for the example. The operation of the faceplate can be simulated via the PLC program.

The functions can be recreated with PLC Sim and the Runtime simulation.

igure 3-8			-
Simulation: Har	dling faceplate		
← ≁ 🔺	Example 01 Example (12	Ξ
Overview	Show faceplate	Simulate status output	
	Fan: 120 X	Display Fault (0-1): Simulate fault	
	Fault Ready for use	<u> </u>	
	10 m3/h		
	Auto Manual 4 Manual on Manual off	1:56:38 PM 1:57:38 PM	
		11/12/2014 11/12/2014	

4.1 Overview

4 Operating the Application

4.1 Overview

The enclosed example configuration helps you get an impression of how the faceplate can be applied.

For testing the example configuration, you can, whenever necessary, also use the simulation integrated in WinCC (TIA Portal).

Calling the example application

Table 4-1

No.	Action	Pictures
1.	Calling the example application After starting Runtime, the following screen will be displayed. Here you select "concrete example". This takes you to the "Overview" screen.	SIEMENS SIMATIC HMI
2.	"Overview" screen The "Overview" screen enables you to navigate to the example application. To do this, click the button " Example 1 " or " Example 2 ".	Application description SEEMENS Cample 02 Cample 02 Overview The application describes the configuration steps to create faceplates. The application describes the configuration steps to create faceplates. The application of sectional applications are leated. Example 1: Example 1: Example 1: Example 1: Example 2: Example 1: Example 3: Example 1: Example 4: Example 2: Example 2: Example 2: Example 3: Example 3: Example 4: Example 4: Example 4: Ex

4.2 "Example 1" picture

4.2 "Example 1" picture

Showing or hiding the faceplates

The picture shows an application case for using faceplates.

The control stations for the fan drives can alternatively be shown or hidden. The picture remains clearer and provides space for further objects.

Table 4	1-2
---------	-----

No.	Action	Pictures
1.	Hiding faceplates In this case, the two control stations are hidden.	Valve 02 Fan 01x Fan 22x Fan 22x
	Apart from the fan symbol, a yellow point is also displayed so the operator can see behind which object a there is a "function" (button). (Clearly apparent in Runtime).	Fan 01x
2.	Faceplate shown	Fan: 12 X
	In this case, a control station has been called. The call is performed via the invisible button located above the fan symbol. Renewed actuation of the invisible button,	Valve 02 Valve 02 Fan 01x Fan 02x Fan 02x Fan 02x
	station, hides the control station again.	
3.	Both faceplates are shown In this case, both control stations have been called. The call is performed via the invisible button located above the fan symbol. Renewed actuation of the invisible button, or pressing the "X" button at the control station, hides the control station again.	Valve 01 Valve 02 Fan 01x Fant in interval for use interval for use in interval for use in interval for

4.3 "Example 2" picture

4.3 "Example 2" picture

Functional view of a faceplate

A PLC program is stored for example 2.

The following functions are evaluated or controlled via the PLC program.

Table 4-3

No.	Action	Pictures
1.	Simulation of the faceplate	
	Showing / hiding the faceplate You can show or hide the button (1) via the faceplate.	Show faceplate 2 Simulate status output Display Fault (0-1): Simulate fault
	You can also hide the control station again via the "X" button at the control station.	Fault Ready for use 114 10 m3/h
	Buttons, fan icon and status display are represented with a background color that depends on the selected function.	Auto Manual Manual on Manual off 156:38 PM 1:57:38 PM 11/12/2014 11/12/2014

5 Further notes, tips and tricks

Tips and help for trouble-shooting

Why is the configured tag with data type "HMI UDT" displayed "red" at the interface of the faceplate?

- Check the used communication driver in the HMI UDT (e.g. S7 300/400 or S7 1500).
- Check the used connection to the HMI tag you assigned the "HMI UDT" as data type to.
- The communication driver at the "HMI UDT" and the used "connection" must match.

What do you need to observe when changing over from a "HMI UDT" to a PLC data type?

- When recreating the structure of the HMI UDT 1.1 in the PLC data type, the connections are adopted when "exchanging" the HMI UDT. Otherwise you need to recreate the connections.

How do you know the version number of the faceplate in the picture?

Call the properties of the faceplate. The version number is displayed in the "header" of the properties.

Why are functions executed in the faceplate (for example, an object is "shown/hidden") even though no parameter exists at the interface?

- Open the faceplate
- In the inspector window you view the properties of the individual objects.

Do all of the parameters of a "HMI UDT" or a PLC data type need to be switched?

No, not all of the stored parameters need to be switched.
 You can, for example, create a "universal" UDT which covers all possible functions and can thus be used for different for faceplates.

What do you need to observe when editing/expanding a PLC data type retrospectively?

- After the changes have been performed, the project must first be restarted. Then, the faceplate to which the PLC data type is linked must be adjusted.

6 Links & Literature

Internet link specifications

This list is by no means complete and only presents a selection of suitable information.

Table 6-1

	Торіс	Title
\1\	Link to this document	https://support.industry.siemens.com/cs/ww/en/view/68014632
\2\	Siemens Industry Online Support	Industry Online Support https://support.industry.siemens.com
131	FAQ	What are the functional differences between the different SIMATIC panels? https://support.industry.siemens.com/cs/ww/en/view/40227286
\4\	FAQ	How can you create faceplates in WinCC (TIA Portal) with user authorizations? https://support.industry.siemens.com/cs/ww/en/view/57434982
\5\	Application	Application: Sample blocks for STEP 7 and WinCC flexible https://support.industry.siemens.com/cs/ww/en/view/36435784
\6\	Application	Faceplates for the Visualization of Sentron PAC Power Meters. https://support.industry.siemens.com/cs/ww/en/view/67318600

7

History

Table 7-1

Version	Date	Modifications
V1.0	01/2013	First version
V2.0	06/2015	Update to WinCC V13 SP1 (new function – PLC data types)