

SIMATIC

Process Control System PCS 7 SFC for SIMATIC S7 (V9.0 SP8)

Programming and Operating Manual

Security information	1
What's new in SFC?	2
Introduction	3
Getting started	4
Essentials of SFC	5
Starting and working	6
Create the project structure	7
Creating, configuring and managing SFCs	8
Adapting the properties of charts, types and instances	9
Configuring messages	10
Configuring the characteristics	11
Configuring sequential control systems	12
Standard interface and chart I/Os	13
Compiling SFCs and SFC types	14
Downloading to the AS	15
Behavior of the sequential control system in the AS	16
Testing and commissioning sequential control systems	17
Documenting programs	18
Configuring parameter controls	19
Tips & Tricks	20
Appendix	21

Legal information

Warning notice system

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

DANGER

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

WARNING

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

CAUTION

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

NOTICE

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

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

WARNING

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

Trademarks

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

Disclaimer of Liability

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

Table of contents

1	Security information	11
2	What's new in SFC?	13
3	Introduction	17
4	Getting started	21
4.1	How to work with SFC in overview	21
4.2	How to create an SFC in overview.....	23
4.3	How to create an SFC type in overview	26
4.4	How to create an SFC instance in overview	28
5	Essentials of SFC	29
5.1	SFC in the STEP 7 environment.....	29
5.2	SFC and plant hierarchy	30
5.3	SFC and other target systems	31
5.4	Configuration limits of the SFC	32
5.5	Uses and functionality of SFCs.....	33
5.5.1	What is an SFC?.....	33
5.5.2	What is an SFC type/SFC instance?	34
5.5.3	SFC and SFC type comparison	37
5.5.4	External view of the SFC.....	37
5.6	SFC elements	40
5.6.1	What is a sequencer?.....	40
5.6.2	What are sequencer elements?	41
5.6.3	What is a step?	42
5.6.4	What is a transition?.....	43
5.6.5	What is a text?	44
5.6.6	What is a sequence?.....	44
5.6.7	What is a simultaneous branch?	45
5.6.8	What is an alternative branch?.....	45
5.6.9	What is a loop?	46
5.6.10	What is a jump?	47
6	Starting and working	49
6.1	Working with the SFC Editor.....	49
6.2	How to start the SFC Editor	50
6.3	Multiuser engineering.....	51
6.4	Navigating in the SFC.....	52
6.4.1	How to navigate in the SFC	52
6.5	User interface and operator input.....	53

6.5.1	Elements of the user interface.....	53
6.5.2	Dialog boxes	56
6.6	Operator inputs with the mouse.....	58
6.7	Working with the keyboard	59
6.7.1	Working with the keyboard	59
6.7.2	Shortcut keys for menu commands	59
6.7.3	Shortcuts for menu commands	60
6.7.4	Navigating in the menu bar and menus with the keyboard.....	61
6.7.5	Navigating in dialog boxes with the keyboard	62
6.7.6	Navigating in texts with the keyboard.....	63
6.7.7	Selecting text with the keyboard	63
6.7.8	Keyboard shortcuts in the sequencer	64
6.7.9	Opening Help with the keyboard	64
6.7.10	Changing the window section	64
6.8	Default colors	65
6.9	Data backup in the SFC Editor.....	66
7	Create the project structure	67
7.1	How to create a project structure.....	67
8	Creating, configuring and managing SFCs	69
8.1	Overview of creating, configuring and managing SFCs.....	69
8.2	Creating an SFC	71
8.2.1	How to create an SFC	71
8.3	Creating an SFC type and creating an SFC instance	72
8.3.1	How to create an SFC type	72
8.3.2	How to create an SFC instance	73
8.4	Configuring an SFC	74
8.4.1	Basics for configuring an SFC.....	74
8.4.2	Configuration changes in the SFC.....	75
8.5	Configuring the SFC type and SFC instance	77
8.5.1	Type/instance concept of SFC	77
8.5.2	Configuring the SFC type.....	77
8.5.3	Configuration changes in the SFC type	79
8.5.4	Configuring the SFC instance.....	80
8.5.5	Configuration changes in the SFC instance	82
8.6	Opening	83
8.6.1	How to open SFCs or SFC types	83
8.6.2	How to open SFC instances	84
8.7	Copying.....	85
8.7.1	Copying and moving SFCs	85
8.7.2	Copying and moving SFC types	86
8.7.3	Copying and moving SFC instances	86
8.8	Deleting.....	88
8.8.1	How to delete SFCs and SFC types	88
8.8.2	How to delete SFC instances.....	88

9	Adapting the properties of charts, types and instances.....	89
9.1	How to adapt chart properties	89
9.2	How to adapt type properties	91
9.3	How to update SFC types	94
9.4	How to adapt instance properties	95
10	Configuring messages	97
10.1	How to configure messages in the SFC.....	97
11	Configuring the characteristics.....	99
11.1	Using the characteristics editor and the interface editor	99
11.2	Characteristics of the SFC type	100
11.3	Note on subsequent modifications to control strategies	102
11.4	How to configure note texts and position texts	103
11.5	Inputs/outputs for characteristics.....	104
11.6	Block contacts.....	105
11.7	Attributes for characteristics.....	107
12	Configuring sequential control systems	111
12.1	How to configure sequential control systems	111
12.2	Creating the sequencer topology.....	112
12.2.1	How to create a sequencer topology.....	112
12.2.2	How to configure multiple sequencers.....	113
12.2.3	Overview of inserting/creating SFC elements.....	116
12.2.4	How to create a sequence	117
12.2.5	How to create and edit a simultaneous branch	117
12.2.6	How to create and edit an alternative branch.....	119
12.2.7	How to create and edit a loop.....	119
12.2.8	How to create and edit a jump.....	121
12.2.9	How to create and edit a text box	122
12.3	Managing SFC elements.....	125
12.3.1	Selecting SFC elements	125
12.3.1.1	How to select with a mouse click.....	125
12.3.1.2	How to select with the keyboard	126
12.3.1.3	How to select with the lasso.....	126
12.3.1.4	How to select steps and transitions for editing.....	126
12.3.2	Copying, moving and deleting SFC elements	127
12.3.2.1	How to copy SFC elements	127
12.3.2.2	How to copy object properties of SFC elements	128
12.3.2.3	How to move SFC elements.....	129
12.3.2.4	Deleting SFC elements	130
12.4	Editing in the properties dialog box	131
12.4.1	How to edit sequencer properties.....	131
12.4.2	Editing the steps	132
12.4.2.1	How to edit a step.....	132

12.4.2.2	How to edit actions in a step	134
12.4.2.3	Technological actions - Basics and definitions of terminology	136
12.4.2.4	How to edit addresses in a step	137
12.4.2.5	Valid entries for addresses	139
12.4.2.6	How to filter block inputs/outputs	141
12.4.2.7	How to access structures	142
12.4.3	Editing transitions	144
12.4.3.1	How to edit the transition	144
12.4.3.2	Formulating the conditions of a transition	146
12.4.3.3	How to edit addresses in a transition	146
12.4.3.4	How to edit OS comments in a transition	148
12.4.4	Shared addresses and valid data types	149
12.4.4.1	Shared addresses	149
12.4.4.2	Absolute addressing	149
12.4.4.3	How to perform symbolic addressing	150
12.4.4.4	Valid data types	151
12.4.5	Calculations in steps and transitions of SFC	151
12.4.5.1	Overview of SFC calculations	151
12.4.5.2	Configuring a calculation in SFC	155
12.4.5.3	Scope of functions in SFC calculations:	158
12.4.5.4	Example for implementing a formula in a calculation for SFC	160
12.5	Specifying runtime properties	162
12.5.1	Runtime properties of an SFC	162
12.5.2	How to edit the run sequence	162
12.5.3	Runtime attributes for runtime groups and SFCs	163
13	Standard interface and chart I/Os	167
13.1	Standard interface of the SFC	167
13.1.1	Inputs/outputs of the SFC standard interface	167
13.1.2	Standard interface of the SFC	167
13.1.3	Meanings in the table	172
13.1.4	Inputs/outputs of the SFC sorted according to usage	172
13.2	Standard interface of the SFC type	174
13.2.1	Standard interface of the SFC type	174
13.2.2	Inputs/outputs of the SFC type standard interface	174
13.2.3	Inputs/outputs of the SFC type sorted according to usage	180
13.2.4	"Inputs/outputs" interface expansions	181
13.2.5	"Characteristics" interface parameter assignments	181
13.2.5.1	"Characteristics" interface parameter assignments	181
13.2.5.2	"Control strategies" characteristic	181
13.2.5.3	Characteristic for note texts	182
13.2.5.4	Characteristic for position texts	182
13.2.6	"Characteristics" interface expansions	183
13.2.6.1	"Characteristics" interface expansions	183
13.2.6.2	"Setpoints" characteristic	183
13.2.6.3	Meaning and usage of setpoints	189
13.2.6.4	"Process values" characteristic	192
13.2.6.5	"Control values" characteristic	192
13.2.6.6	"Parameters" characteristic	192
13.2.6.7	"Bit memory" characteristic	193
13.2.6.8	"Timers" characteristic	193

13.2.6.9	"Block contacts" characteristic	194
13.3	SFC inputs/outputs sorted by usage	195
13.3.1	Operating modes (SFC inputs/outputs)	195
13.3.2	Commands and operating states (SFC inputs/outputs)	196
13.3.3	Execution options (SFC inputs/outputs)	202
13.3.4	Group displays and group acknowledgements (SFC inputs/outputs)	204
13.3.5	Data from sequencers and steps to be processed (SFC inputs/outputs)	204
13.3.6	BATCH parameters (SFC inputs/outputs)	205
13.3.7	Continuous mode (SFC inputs/outputs)	206
13.3.8	Troubleshooting (SFC inputs/outputs)	206
13.3.9	Messages (SFC inputs/outputs)	209
13.3.10	Control words (SFC inputs/outputs)	210
13.3.11	Status words (SFC inputs/outputs)	210
13.3.12	System parameters (SFC inputs/outputs)	214
13.3.13	Reserves (SFC inputs/outputs)	215
14	Compiling SFCs and SFC types	217
14.1	Overview of compiling charts, types, and instances	217
14.2	Settings for compiling/downloading	219
14.3	How to compile	220
14.4	Overview of the blocks generated during compilation	221
14.5	Consistency check	223
15	Downloading to the AS	225
15.1	How to download programs	225
15.2	Reaction of the SFC at deactivation before downloading changes only	231
16	Behavior of the sequential control system in the AS	233
16.1	Sequential control systems in the AS	233
16.2	Specifying the runtime behavior	234
16.2.1	Runtime behavior of the sequential control system	234
16.2.2	Operating modes	234
16.2.3	Step control modes	235
16.2.4	Execution options	236
16.2.5	Operating states	238
16.2.5.1	Operating states	238
16.2.5.2	Operating state logic for SFC (SFC OSL)	239
16.2.5.3	Continuous mode	243
16.2.5.4	Diagram of the state changes for SFC OSL	245
16.2.5.5	Operating state logic for sequencers (sequencer OSL)	246
16.2.5.6	Diagram of the state changes for sequencer OSL	248
16.3	Processing the SFC in the AS	249
16.3.1	Processing an SFC	249
16.3.2	Processing an SFC after CPU stop and restart	251
16.3.3	Commands	253
16.3.4	Start requirements of an SFC (chart/instance)	254
16.3.5	Behavior when multiple commands are queued	255
16.3.6	Processing the SFC elements	256

16.3.6.1	Runtime phases of a step	256
16.3.6.2	Processing a step and transition	256
16.3.6.3	Processing a simultaneous branch	258
16.3.6.4	Processing an alternative branch	259
16.3.6.5	Processing a loop	260
16.3.6.6	Processing a jump	260
16.4	Changing the control strategy and setpoints.....	262
16.4.1	Changing the control strategy and setpoints for an SFC instance.....	262
16.4.2	Tracking the control strategy and setpoints in AUTO mode.....	264
17	Testing and commissioning sequential control systems	265
17.1	Operator control and monitoring during testing	265
17.2	Display in test mode.....	270
17.3	States of the steps and transitions	274
17.4	Icons for operating states	275
17.5	Properties of a step during testing	276
17.6	Properties of a transition during testing.....	279
17.7	Properties of a sequencer during testing.....	282
17.8	Calculations in steps and transitions during testing.....	284
18	Documenting programs.....	285
18.1	Documenting SFCs.....	285
18.2	Chart reference data	287
18.3	Logs	289
18.4	Defining footers	291
19	Configuring parameter controls	293
19.1	Parameter control	293
19.2	How to configure the runtime	294
19.3	Execution with different parameter sets.....	295
19.4	Example of a recipe data block	297
20	Tips & Tricks	299
20.1	Configuring SFC calls.....	299
20.2	Converting older projects	300
21	Appendix.....	303
21.1	Allocation request and allocation status	303
21.2	Data types	306
21.2.1	BOOL, BO.....	306
21.2.2	BYTE, BY	306
21.2.3	CHAR, C	306
21.2.4	DATE, D	307
21.2.5	DINT, DI	307

21.2.6	DWORD, DW	307
21.2.7	INT, I.....	308
21.2.8	REAL, R	308
21.2.9	S5TIME, T5	308
21.2.10	STRING, S.....	309
21.2.11	STRUCT, ST.....	309
21.2.12	TIME OF DAY, T	310
21.2.13	TIME, TI	310
21.2.14	WORD, W	310
Index		311

Security information

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

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

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

For additional information on industrial security measures that may be implemented, please visit

<https://www.siemens.com/industrialsecurity>.

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

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

<https://www.siemens.com/cert>.

What's new in SFC?

Enhancements/changes in V9.0 SP8

Compared with the previous version V9.0 SP7, version V9.0 SP8 includes the following enhancements or changes:

- A less restrictive download method is introduced for SFC types to benefit even more from typification of Equipment Modules (EM). Changing technological actions in a step no longer causes the SFC type to be aborted when downloading to the AS. In addition, for CPUs that support Type Change in RUN (TCiR method), that is CPU 410-5H PA, the changed I/O interface of SFC type (for example, adding a setpoint to the SFC type) can now be downloaded to the AS in RUN mode without aborting the sequencer. In this case, an abort can be prevented by using the TCiR method. Since topological changes in the active sequencer of an SFC type inevitably lead to an abort, it is recommended to add the required changes to the technological actions as conditional commands. This can completely avoid an abort. For more information, refer to "How to download programs (Page 225)".

Enhancements/changes in V9.0 SP7

Compared with the previous version V9.0 SP6, version V9.0 SP7 includes the following enhancements or changes:

- In the "Calculation for SFC" dialog box, you can now use "Open Type" context menu command to open the corresponding type (template) of the selected command or status.

Enhancements/changes in V9.0 SP6

Compared with the previous version V9.0 SP5, version V9.0 SP6 includes the following enhancements or changes:

- In the "Calculation for SFC" dialog box, you can now use "Go To" button to open the corresponding type (template) of the selected command or status.

Enhancements/changes in V9.0 SP5

Compared with the previous version V9.0 SP4, version V9.0 SP5 includes the following enhancements or changes:

- In V9.0 SP5, several minor corrections and improvements were made in the SFC.

Enhancements/changes in V9.0 SP4

Compared with the previous version V9.0 SP3, version V9.0 SP4 includes the following enhancements or changes:

- In V9.0 SP4, several minor corrections and improvements were made in the SFC.

Enhancements/changes in V9.0 SP3

Compared with Version V9.0 SP1, Version V9.0 SP3 includes the following enhancements or changes:

- New tabs for configuring the technological actions for preprocessing/postprocessing the sequencer of an SFC type
In an SFC type, the following two new tabs have been added in the "Properties" dialog box of a sequencer:
 - "Preprocessing (technological)"
 - "Postprocessing (technological)"

In these tabs, you can configure the technological actions for preprocessing or postprocessing the sequencer of an SFC type and can monitor them in the test mode.

Enhancements/changes in V9.0 SP1

Compared with Version V9.0, Version V9.0 SP1 includes the following enhancements or changes:

- The logging function for loading has been enhanced.
The "Archive" button in the "Load" tab of the "Logs" dialog box allows you to open a selection dialog for selecting and displaying log files with information about the loading process. In addition to the previous log files, the information about the loading process can now also be saved in XML format. These XML files with the information on the loaded changes can be compared in VXM if needed.
You can find additional information about this in the section "Change log and ES log" in the help or documentation *Process Control System PCS 7; CFC for SIMATIC S7*.

Enhancements/changes in V9.0

Version V9.0 contains the following enhancements or changes compared with version V8.2:

- Configuration of technological actions and conditions in SFC types
Technological actions in steps and technological conditions in transitions/start conditions can be configured in the SFC editor.
The current values of technological actions and conditions are displayed in the test mode of the SFC editor.
You can find additional information about this in the sections "How to edit a step (Page 132)" and "How to edit the transition (Page 144)".
- Classifications for SIMATIC BATCH
The classifications in the properties of an SFC type have been extended for the connection to SIMATIC BATCH via the equipment phase (EPH).
You can find additional information about this in the sections "How to adapt type properties (Page 91)" and "How to adapt instance properties (Page 95)".
- Global commands and status in the "SFC Library"
Pre-defined global commands and states are provided in the "SFC Library". These can be copied and used in the "sequential control systems" of equipment phases or equipment modules.

Enhancements/changes in V8.2

Version V8.2 contains the following enhancements or changes compared with version V8.1:

- **Calculations in SFC**

Calculations can be configured in steps, transitions or sequencers in the SFC Editor.

The values of the calculation results and the output variables are displayed in test mode of the SFC Editor.

You can find additional information about this in the section "Overview of SFC calculations (Page 151)".

Enhancements/changes in V8.1

Version V8.1 contains the following enhancements or modifications compared with version V8.0:

- **Block icons for process images in a style similar to that of the APL**

The templates for the block icons can be present in different versions. Variants "1" and "2" are already provided as standard.

- Variant "1" contains icons in the PCS 7 standard.

- Variant "2" contains icons in a style similar to that of the APL.

You can find additional information in section "Configuring SFC block icons" of the *SFC Visualization* manual or online help.

- **"Selective Download" function**

The new "Selective Download" function can be used to download one or more CFCs or SFCs to a CPU. You can use selective downloading as a user to specifically select which configured changes should be transferred to the CPU, enabled and tested.

You can find additional information in the section "Selective download of individual charts" of the *CFC for SIMATIC S7* manual or online help.

- **Type update in RUN mode**

CPU 410-5H PA supports type update in RUN mode. This makes it possible to update the instances and download them to the CPU in RUN mode after changing an interface at the block types. This is only possible in STOP mode in other automation systems.

You can find additional information about this in the section "How to download programs (Page 225)".

Introduction

What is SFC?

SFC (Sequential Function Chart) enables you to graphically configure and commission sequential control systems. The sequential control systems are transferred to an automation system and executed there. A sequential control system allows state-driven or event-driven execution of production processes based on sequencers.

You can use sequential control systems to describe the manufacturing specifications of products as event-driven processes (recipes), for example.

The sequential control system controls the basic automation functions created with CFC via operating and state changes and processes them selectively.

SFC provides two independent variants of sequential control systems for different application scenarios:

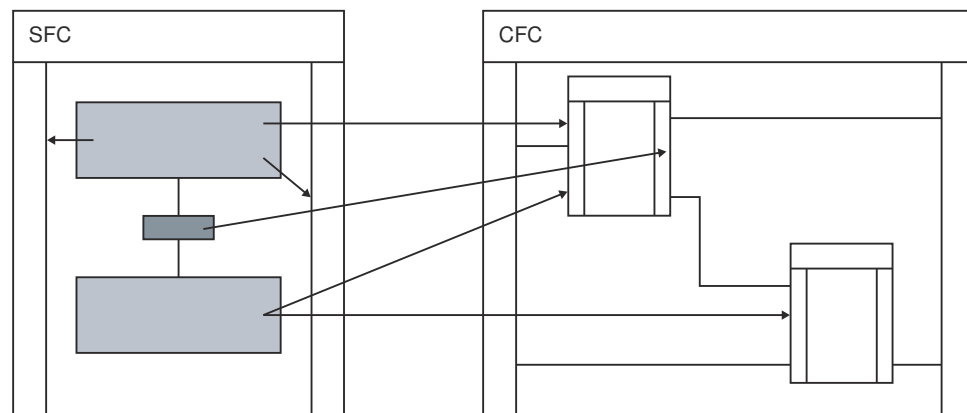
- SFC
- SFC type with SFC instances

SFC

An SFC contains the following:

- A standardized interface for controlling the SFC through the user program or by the operator
- Up to 8 sequencers for formulating the sequential control system ("gray sequencer")

The sequential control system accesses the basic automation blocks and signals directly and is therefore not reusable.

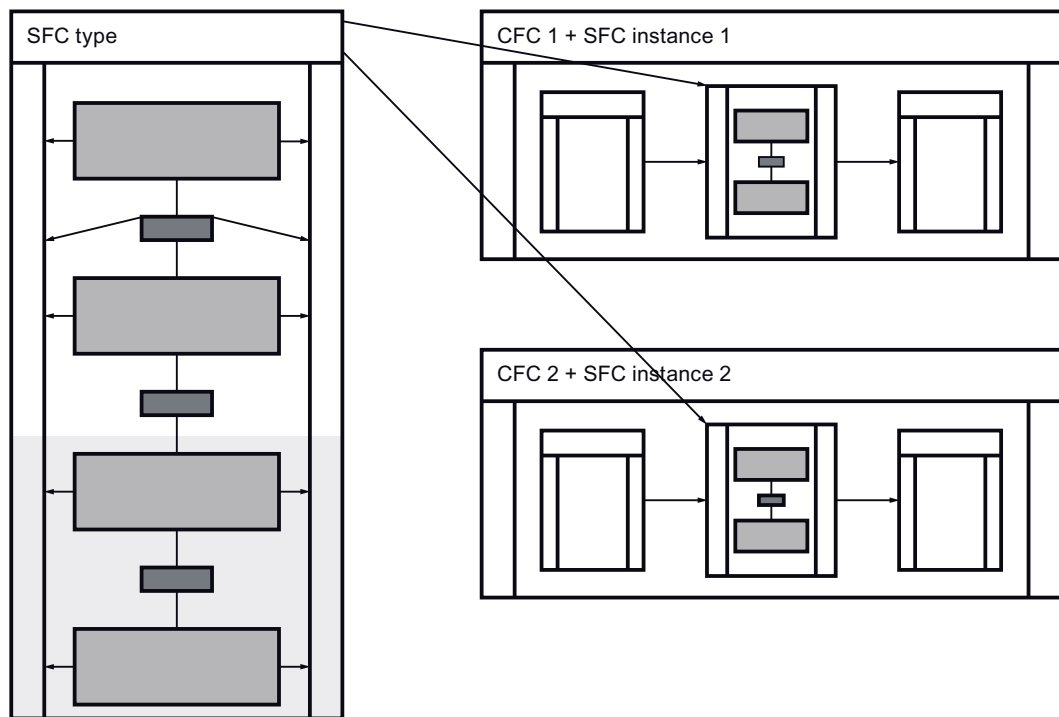


SFC type/SFC instance

An SFC type contains the following:

- A standardized interface for controlling the SFC through the user program or by the operator
- Up to 32 sequencers for formulating the sequential control system ("gray sequencer")

You can also expand the interface ("gray interface"). The SFC type accesses only its own interface and can therefore be used as often as required as an SFC instance.



An SFC instance is derived from an SFC type and initially has the same properties as the SFC type. You can adapt the SFC type to the SFC instance to a limited extent on an instance-specific basis through changes in the CFC or SFC.

Criteria for selecting SFC or SFC type

You use an **SFC** when you need a sequential control system that is to be used once and will control multiple sub areas of the production plant.

You use an **SFC type** when you need a sequential control system that is to be used more than once and has SFC instances that will each control their own sub area of the production plant. Central changes to an SFC type are automatically forwarded to all SFC instances.

You can use the following criteria in the order given to decide if an SFC or an SFC type is required:

- Sphere of action
- Reuse
- Changeability

Example of decision making:

If the sphere of action is large but the sequential control system is to be used more than once, you still may need to implement the sequential control system as an SFC in certain circumstances. Reuse is then only possible in the form of copies. In this case, you cannot make changes from a central location.

If the sphere of action is small, you can use either an SFC or an SFC type. Here, the decision depends on the ability to reuse and to change.

What is a sequential control system?

A sequential control system allows the structuring of production processes by breaking them down into consecutive steps. The steps define the actions to be executed and therefore the operations required in the production plant, for example:

- Switch on motor
- Open valve

The passage from one step to the next triggers a transition with a defined step enabling condition, for example:

- Temperature achieved
- Reactor empty

What are sequencers?

You can use sequencers as a higher-level structuring tool that defines state or event-triggered sub-sequences, for example:

- Sequence for production
- Sequence for holding
- Sequence for error handling

You specify the state or event that is to trigger execution of the sequencer in the start condition of the sequencer. The sequencer itself is formulated as a series of steps and transitions.

A sequential control system contains at least one sequencer. In the initial state, its start condition is set so that the sequencer is processed when the sequential control system is in the "active" state (condition: RUN = TRUE).

Where are sequential control systems used?

Typical applications of sequential control systems involve batch processing plants. But you can also use sequential control systems for continuously operating plants, for example, for:

- Startup and shutdown operations
- Operating point changes
- State changes when faults occur

You can use sequential control systems in the following plant levels:

- Device control level (for example, opening a valve or starting a motor)
- Group control level (for example, dosing, stirring, heating, or filling)
- Unit level (for example, tanks, mixers, scales, reactors)
- Plant level (synchronization of units and common resources, for example, routing)

You can use SFCs at all levels, while SFC types are generally used on the group control and unit levels because requirements for reusability at the higher levels are unlikely.

Note

The SFC editor is a tool for creating and testing sequential control systems.

In this documentation, the term "SFC" refers to the chart, type, instance or editor depending on the context.

Getting started

4.1 How to work with SFC in overview

Operating principle

1. You create your sequential control system using graphic tools in the SFC editor. You place the elements of the SFC in the sequencer according to predefined rules. You do not need to be aware of details such as algorithms or the assignment of machine resources but can concentrate solely on the technological aspects of your configuration.
2. After creating the sequencers as a series of steps and transitions, you proceed to configure the object properties of SFC, sequencers, steps and transitions, and formulate the respective properties there. You configure the following:
 - Operating parameters of the SFC
 - Start conditions of the sequencers
 - Actions of the steps
 - Step enabling conditions of the transitions
3. When this configuration is complete, you compile the executable machine code with SFC, download it to the AS, and test it with the SFC test functions.

SFC

If you have never before worked with the SFC editor, it is best to begin with an SFC. In this way, you will develop an understanding of the following topics:

- Creating, configuring, compiling and downloading a sequential control system
- How the sequential control system works in the AS
- How to use commands (such as Start and Abort) in a sequential control system in test mode

SFC type and SFC instance

Once you know how to implement a sequential control system with an SFC, you can begin to work with the SFC type and SFC instance. You can then decide which of the two sequential control systems is better suited for the specific task.

The SFC type and SFC instance introduce new aspects of working with sequential control systems. With the SFC, you can start working directly because an SFC can be generated and then compiled, downloaded and tested immediately.

With an SFC type, you create the sequential control system independent of a specific application in the form of a template so that it can be used more than once.

Then you create an SFC instance, adapt it to the individual application and interconnect it to the basic automation blocks.

4.1 How to work with SFC in overview

A standard interface must be available for the SFC type in order to interconnect an SFC instance with the basic automation. You can expand this interface to include new inputs/outputs or create the required inputs/outputs technologically as characteristics of the SFC type.

You then compile the program, download it to the AS, and test the SFC instance. This also tests the SFC type indirectly.

Additional information

For additional information on this topic, see the following sections:

How to create an SFC in overview (Page 23)

How to create an SFC type in overview (Page 26)

How to create an SFC instance in overview (Page 28)

4.2 How to create an SFC in overview

Introduction

The steps for configuring SFCs are presented in order below.

Procedure

1. **Create the project structure**

Create a project structure in the SIMATIC Manager in which you configure CFC/SFCs. You can find additional information about this in the section [How to create a project structure \(Page 67\)](#)

2. **Adapt the chart properties**

You can change the general properties (name, author, comment), adapt the operating parameters and change the version. By setting the operating parameters, you specify the behavior of the sequential control system, such as operating mode (MANUAL, AUTO), step control mode (T, O, T and C, etc.) and other execution options (cyclic operation, time monitoring or autostart). In the chart properties, you can change the option that the chart is to be transferred to the OS. You can find additional information about this in the section [How to adapt chart properties \(Page 89\)](#)

3. **Runtime properties**

The runtime properties of an SFC specify how this SFC fits in the time sequence of the processing within the overall structure of the target system. You can change the runtime properties in the window of the sequence editor for the CFC. You can find additional information about this in the section [Runtime properties of an SFC \(Page 162\)](#)

4. **Create the sequencers**

The newly created SFC already contains one sequencer (RUN). You create additional sequencers in the tab of the currently selected sequencer using the shortcut menu or you copy sequencers from the templates of the SFC Library. You can find additional information about this in the section [How to configure multiple sequencers \(Page 113\)](#)

5. **Configure the sequencer properties**

For each sequencer, you configure the start condition and, as an option, the action for preprocessing and postprocessing and the OS comment, if needed. You can find additional information about this in the section [How to edit sequencer properties \(Page 131\)](#)

6. **Create the topology of the sequencers**

You configure sequential control systems in the SFC editor by inserting the steps and transitions for one or more sequencers and adding additional elements as necessary. You can find additional information about this in the section [How to create a sequencer topology \(Page 112\)](#)

7. Configure the steps (in the object properties dialog box)

You formulate actions in steps. The actions contain statements that change the values of block inputs and of shared addresses or that enable and disable runtime groups or other SFCs. You can find additional information about this in the section [How to edit a step \(Page 132\)](#)

8. Configure the transitions (in the object properties dialog box)

You formulate conditions in the transitions to read the following:

- The values of the block I/Os and shared addresses.
- The state (such as on/off) of runtime groups or other SFCs.

When the conditions of a specified logic operation are met, the next step becomes active in the execution of the SFC and its actions are executed.

You can find additional information about this in the section [How to edit the transition \(Page 144\)](#)

9. Compile and download the program

When you compile the current chart folder, an executable user program is generated (compile entire program) that you can then download to the target system (CPU).

You can find additional information on this in the sections:

[Overview of compiling charts, types, and instances \(Page 217\)](#)

[How to download programs \(Page 225\)](#)

10. Test the program

After compiling and downloading, you test the program in process mode or in laboratory mode. Using the SFC test functions, you can run the sequential control system in various operating modes and step control modes and monitor and modify the values of addresses in the CPU. You can also influence the most important operating modes (e.g., STOP, Clear/Reset, RUN, for example) of the CPU.

You can find additional information about this in the section [Operator control and monitoring during testing \(Page 265\)](#).

Note

Prior to compiling (Item 9), you must interconnect the external view of the SFC or the SFC instance (of the SFC type) for AUTO mode in the CFC.

You can find additional information about this in the section: [External view of the SFC \(Page 37\)](#)

Tips

The following tips may help you:

- In the "SFC Library", you can access the "ChartStates" SFC with the menu command **SFC Library > Blocks+Templates > Templates**. This already contains several sequencers for state-oriented processing of the sequential control system. You can copy this chart and use it as an example.
- If you want to focus initially on learning the steps from creation to testing of a sequential control system, you should compile, download and test the "ChartStates" SFC (or a newly created SFC) without any modifications. The SFC is processed in MANUAL mode. In this way, you can monitor and influence the processing of the SFC in test mode with the available test functions.
- To stop the SFC from being processed "too quickly" in the automation system, you can move the SFC in the run sequence, for example, to the "OB32" task. Select the menu command **Debug > Test Settings** and select a suitable monitoring cycle for the test settings in the SFC.

4.3 How to create an SFC type in overview

Introduction

The steps for configuring SFC types are presented in order below:

Procedure

1. **Create the project structure**

Create a project structure in the SIMATIC Manager in which you can configure CFC/SFCs and SFC types.

You can find additional information about this in the section

How to create a project structure (Page 67)

Note:

You can also create an SFC type in a library.

2. **Create the SFC type**

In the Component View of SIMATIC Manager, insert an SFC type as a new object in the chart folder. The SFC type is entered in the block catalog of CFC.

Tip: In the "SFC Library", you can access the "TypeStates" SFC type with the menu command **SFC Library > Blocks+Templates > Templates**. The SFC type already contains several sequencers for state-oriented processing of the sequential control system. You will also find the "TypeCtrlStrategy" SFC type. It contains a control strategy-oriented processing of the sequential control system. You can copy these templates and change them to suit your purposes.

You can find additional information on this with the tips in the section:

Configuring the SFC type (Page 77)

3. **Adapt the SFC properties**

You can change the general properties (name, author, family, FB number, comment), adapt the operating parameters and change the version.

By setting the operating parameters, you specify the behavior of the sequential control system, such as operating mode (MANUAL, AUTO), step control mode (T, O, T and O, for example) and other execution options (such as cyclic operation, time monitoring or autostart).

You can find additional information about this in the section

How to adapt type properties (Page 91)

4. **Create the interface**

The SFC type already has a standard interface that usually must be extended for the interconnections of SFC instances with the basic automation.

You can find additional information in the SFC online help, in sections

Standard interface of the SFC type (Page 174)

"Inputs/outputs" interface expansions (Page 181)

"Characteristics" interface expansions (Page 183).

5. **Optional: Configuring the characteristics**

You use the Characteristics editor for the technological configuration of the interface. In the Characteristics editor, you define characteristics and insert control strategies, setpoints, process values or block contacts, for example.

You can find additional information about this in the section

Characteristics of the SFC type (Page 100)

6. Create the sequencers

Open the SFC type. The newly created SFC already contains one sequencer (RUN). You create additional sequencers in the tab of the currently selected sequencer with the shortcut menu or you copy sequencers from the templates in the SFC Library (SIMATIC Manager). You can copy these templates and adapt them to suit your purposes.

You can find additional information about this in the section
How to configure multiple sequencers (Page 113)

7. Configure the sequencer properties

For each sequencer, you configure the start condition and, as an option, the action for preprocessing and postprocessing and the OS comment, if needed.

You can find additional information about this in the section
How to edit sequencer properties (Page 131)

8. Create the topology of the sequencers

You configure sequential control systems in the SFC editor by inserting the steps and transitions for one or more sequencers and adding additional elements as necessary.

You can find additional information about this in the section
How to create a sequencer topology (Page 112)

9. Configure the steps (in the "Object Properties" dialog box)

You formulate actions in steps. The actions contain statements that change the values of inputs of the interface of the SFC type.

You can find additional information about this in the section
How to edit a step (Page 132)

10. Configure the transitions (in the "Object Properties" dialog box)

You formulate conditions in the transitions to read the values of I/Os from the interface of the SFC type. When the conditions of a specified logic operation are met, the next step becomes active in the execution of the SFC and its actions are executed.

You can find additional information about this in the section
How to edit the transition (Page 144)

11. Create the SFC instances

You can find additional information on this in the section:
How to create an SFC instance in overview (Page 28)

4.4 How to create an SFC instance in overview

Introduction

The steps for configuring SFC instances are presented in order below:

Procedure

1. **Open the CFC**
Open the CFC in which you want to interconnect an SFC instance to the basic automation blocks.
Place the SFC type in the chart from the CFC block catalog or from a library and thereby generate an SFC instance.
You can find additional information about this in the section
How to create an SFC instance (Page 73)
2. **Specify the SFC instance properties**
Change the general properties (name, comment) in the object properties of the SFC instance in the CFC, as required.
You can find additional information about this in the section
How to adapt instance properties (Page 95)
3. **Adapt the operating parameters and options of the instance**
In the CFC, open the SFC instance and adapt the operating parameters in the "Properties" dialog box. These parameters determine the runtime behavior in the AS.
As an option, select which of the control strategies specified by the SFC type are to be used for the SFC instance.
You can find additional information on this in the section:
How to adapt instance properties (Page 95)
4. **Assign parameters and interconnect the interface of the SFC instance**
You assign parameters for the SFC instance in CFC using the object properties or in SFC using the "I/Os" Interface editor.
In CFC, you interconnect the inputs/outputs of the SFC instance to the inputs/outputs of the CFC blocks or to shared addresses.
5. **Compile and download the program**
When you compile the current chart folder, an executable user program is generated (compile entire program) that you can then download to the target system (CPU).
For additional information on this topic, see the following sections:
Overview of compiling charts, types, and instances (Page 217)
How to download programs (Page 225)
6. **Test the program**
After compiling and downloading, you can test the program in process mode or in laboratory mode. Using the test functions, you run the sequential control system online in various operating modes and step control modes and monitor and modify the values of addresses. You can also influence the most important operating modes (STOP or RUN, for example) of the CPU.
You can find additional information about this in the section
Operator control and monitoring during testing (Page 265).

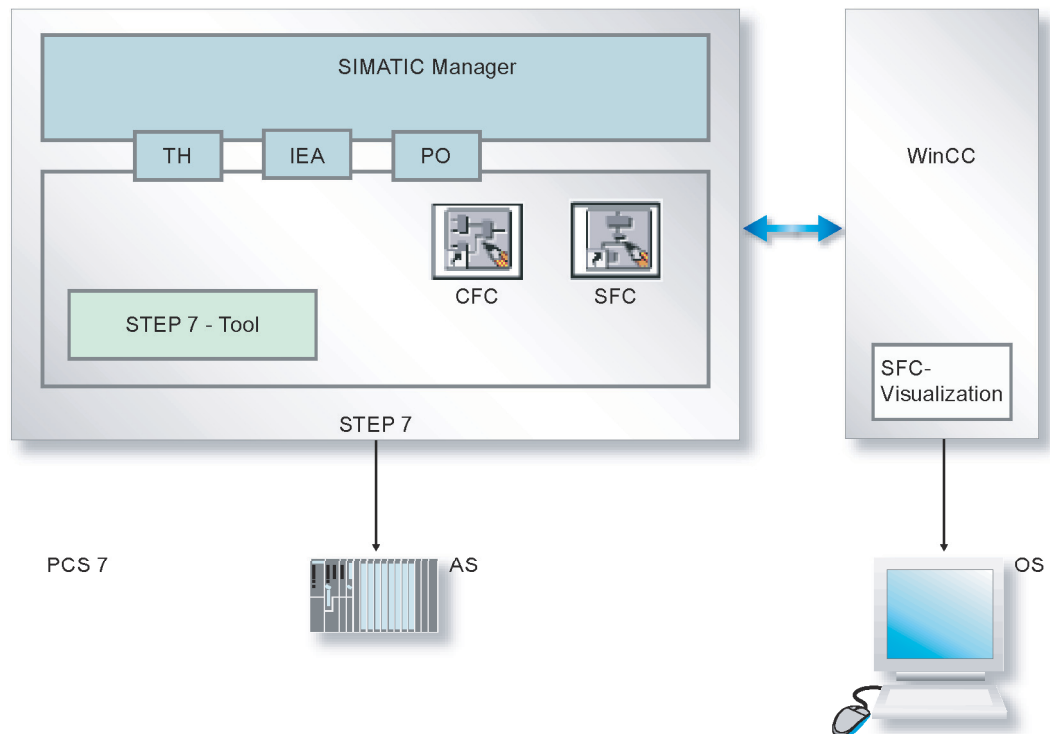
Essentials of SFC

5.1 SFC in the STEP 7 environment

Configuration environment

SIMATIC Manager is used as the graphic user interface for all target systems and it is used to coordinate the tools and objects. It manages tools and data and is used, among other things, to create and modify a project structure (CPU, CFC/SFCs) and to start the SFC editor.

The figure shows how SFC fits into the STEP 7 and PCS 7 environments.



Key:

Object	Meaning
PH (Plant Hierarchy)	Software packages of the process control system (PCS 7)
IEA (Import/Export Assistant)	
PO (process object view)	
WinCC	Operator control and monitoring system in PCS 7 (here with the add-on package for SFC Visualization)

5.2 SFC and plant hierarchy

Use of an SFC in the plant hierarchy

The plant hierarchy (PH) allows charts to be arranged and managed not only from the point of view of running them on a CPU, but also according to technological or plant criteria (for example, an SFC for device control, group control, or unit control).

If the SFC was assigned to a plant hierarchy folder, the path of the plant hierarchy is added to the chart name. You thus use the naming scheme of your plant as the criteria for arranging charts in the project.

Note

You cannot assign SFC types to a hierarchy folder in the plant view because they themselves are not relevant to execution (from the perspective of the process to be automated).

Additional information

You can find additional information on the plant hierarchy in the online help for PH.

5.3 SFC and other target systems

SFC and other target systems

This *SFC for S7* documentation contains a considerable amount of information that is only relevant for the S7 target system. To avoid needing to point this out in each individual case, the most important topics and functions that are irrelevant or are handled different for other target systems are listed below.

- Symbolic addressing
- Shared addresses
- Program-controlled enabling/disabling of charts

5.4 Configuration limits of the SFC

The SFC has the following configuration limits:

Object	Number
Sequencers per SFC	≤ 8
Sequencers per SFC type	≤ 32
Steps per sequencer	2 - 255
Instructions per action	≤ 50
Transitions per sequencer	1 - 255
Conditions per transition / start condition	≤ 16

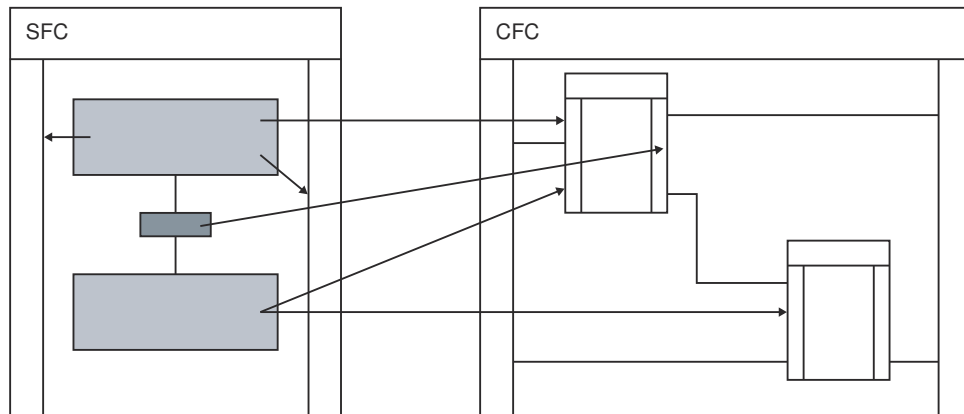
5.5 Uses and functionality of SFCs

5.5.1 What is an SFC?

Use and functionality

An SFC is a sequential control system that normally controls a larger section of a plant and is only used once because the same control task does not occur again in the plant.

To this end, the SFC has a set of properties and a standardized interface for controlling the SFC through the user program or by the operator. The sequential control system accesses the basic automation blocks and signals directly and is therefore not reusable.



To configure the sequential control system, you must be familiar with the mechanisms for processing an SFC and its sequencers in the AS.

You can find additional information on this in the following sections:
Sequential control systems in the AS (Page 233)

Runtime behavior of the sequential control system (Page 234)

Operating states (Page 238)

Operating state logic for SFC (SFC OSL) (Page 239)

Processing an SFC (Page 249)

Properties

The properties of the SFC include the following elements:

- Name, author and comment as descriptive data
- The operating parameters for the AS that determine the execution in the AS (for example, operating mode, step control mode, execution options).

You can find additional information on this in the following sections:
Operating modes (Page 234)

Step control modes (Page 235)

Execution options (Page 236)

Runtime behavior of the sequential control system (Page 234)

You can also specify whether operator control and monitoring of the SFC on the OS is to be possible ("Transfer chart to OS for visualization" check box). You require the "SFC Visualization" add-on package on the OS for this purpose.

You can configure the message properties and the footer data of the SFC as additional properties.

Control

The standard interface of the SFC contains the required inputs/outputs for the following tasks:

- Controlling the SFC in AUTO mode by means of the user program
- Operator control and monitoring of the SFC in MANUAL mode by the operator

Control of the SFC in AUTO mode is realized in the external view of the SFC.

You can find more detailed information on the external view in the section:
External view of the SFC (Page 37)

You can find more information on the SFC in the following sections of the online help:
Inputs/outputs of the SFC standard interface (Page 167)

Standard interface of the SFC (Page 167)

Inputs/outputs of the SFC sorted according to usage (Page 172)

Sequencers

The actual sequential control system is formulated with the sequencers.

The SFC allows configuration of up to 8 sequencers for formulating the control function (in the figure this is the "gray sequencer").

You can find more information on sequencers in the following sections:
What is a sequencer? (Page 40)

What is an SFC type/SFC instance? (Page 34)

SFC and SFC type comparison (Page 37)

5.5.2 What is an SFC type/SFC instance?

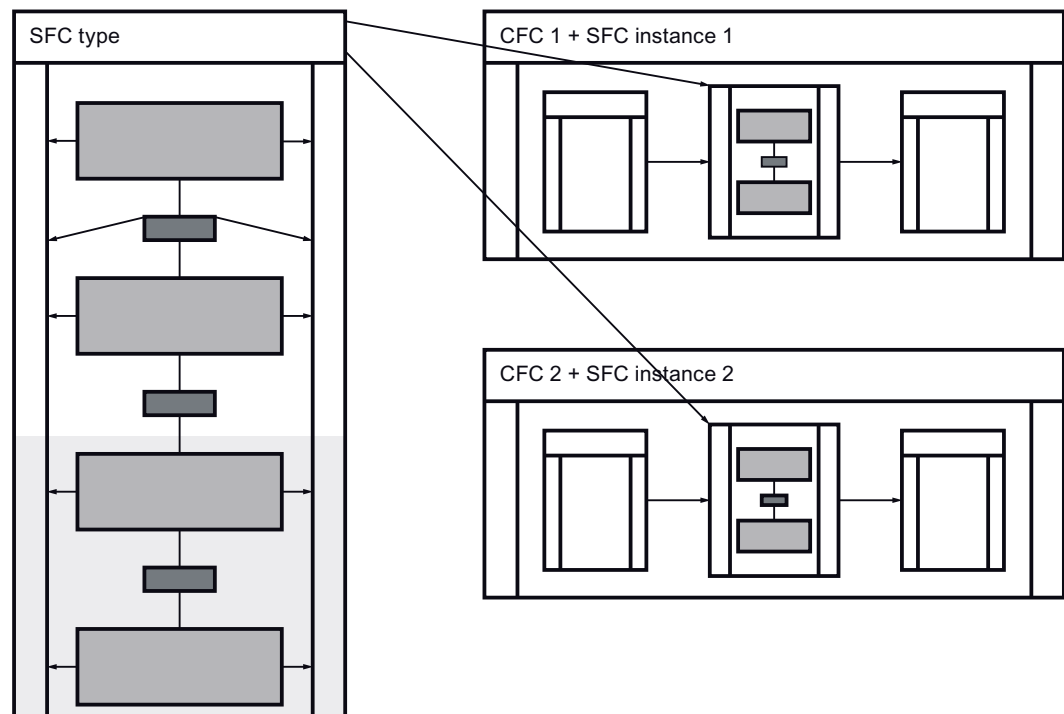
SFC type

An SFC type is a sequential control system that normally controls a smaller sub area of a plant (unit or section of a plant), whereby the control task occurs repeatedly in similar form in the plant. For this reason, with the SFC type the sequential control system is created as a type-defined template apart from a specific application. Once it is created, you can use the SFC type repeatedly by generating SFC instances from it.

The SFC type contains a standardized interface to allow control of the SFC from the user program or by the user and allows configuration of up to 32 sequencers for formulating the control function ("gray sequencer"). The interface can also be expanded by the user (interface with "gray background" in figure). The SFC type accesses only its own interface and can therefore be used as often as required as an SFC instance.

SFC instance

An SFC instance is derived from an SFC type and initially has the same properties as the SFC type. You can adapt the SFC instance to a limited extent on an instance-specific basis in CFC or SFC. Only after interconnecting the SFC instance with blocks and signals of basic automation do you have an executable sequential control system.



Properties of an SFC type and an SFC instance

The properties of the SFC type and SFC instance include the name and comment as descriptive data. The SFC type has the additional properties author, version and family and FB number as information for the compilation. Both for the SFC type and the SFC instance, you can specify the operating parameters for the AS that determine the execution in the AS (for example, operating mode, step control mode or execution options).

You will find more information on this subject in the following sections:

Operating modes (Page 234)

Step control modes (Page 235)

Execution options (Page 236)

Runtime behavior of the sequential control system (Page 234)

You can also set additional options and specify whether the SFC type and SFC instance are relevant for SIMATIC BATCH and which of the defined control strategies will be enabled for use with the SFC type or SFC instance. As additional properties, you can configure the message properties and the footer data of the SFC type and SFC instance.

SFC instances can be made available on the OS (default: Yes) so that operator control and monitoring of the SFC instances can be performed on the OS with the "SFC Visualization" add-on package. If you do not want SFC instances to be available on the OS, you need to clear the "OCM possible" check box in the object properties for the instance.

The standard interface of the SFC type includes the inputs/outputs needed to control an SFC instance from the user program in AUTO mode or for operator control and monitoring of the SFC instance by the user in MANUAL mode.

You will find detailed information on the standard interface in the section:

Standard interface of the SFC type (Page 174)

To interconnect an SFC instance with the basic automation, you usually need to expand the interface for the SFC type. You can do this by adding new inputs/outputs directly for the interface of the SFC type or by creating the necessary inputs/outputs technologically as characteristics of the SFC type. We recommend this procedure.

You will find more information on the interface in the sections:

"Characteristics" interface parameter assignments (Page 181)

"Characteristics" interface expansions (Page 183)

"Inputs/outputs" interface expansions (Page 181)

The actual sequential control system is formulated with the sequencers.

You can find more information on sequencers in the section:

What is a sequencer? (Page 40)

To configure the sequential control system, you must be familiar with the mechanisms for processing an SFC and its sequencers in the AS.

You can find additional information about this in the following sections:

Sequential control systems in the AS (Page 233)

Runtime behavior of the sequential control system (Page 234)

Operating states (Page 238)

Operating state logic for SFC (SFC OSL) (Page 239)

Processing an SFC (Page 249)

5.5.3 SFC and SFC type comparison

Common features

An SFC and an SFC type have the following common features:

- Standard interface for external control of the SFC (MANUAL/AUTO)
- Sequencers for formulating the control function of the SFC

Differences

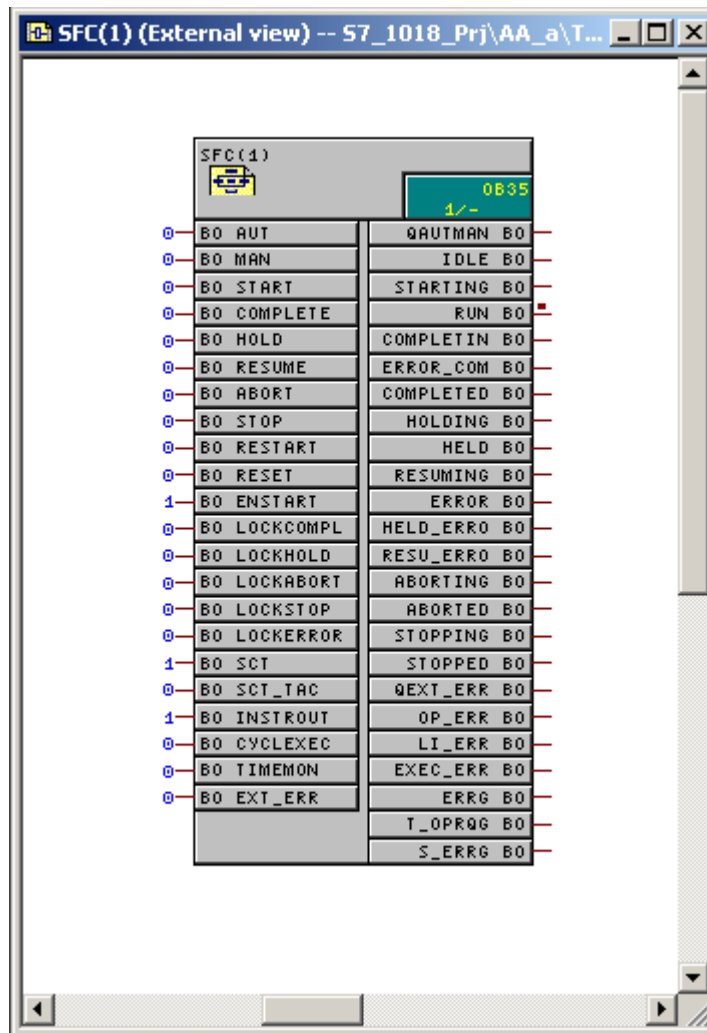
An SFC and an SFC type differ as follows:

SFC	SFC type
Direct access to basic automation	Access to basic automation by means of interface
Can be used once	Can be used more than once
Can be modified locally	Can be modified centrally
Interface cannot be expanded	Interface can be expanded
Maximum of 8 sequencers can be configured	Maximum of 32 sequencers can be configured

5.5.4 External view of the SFC

Display

The external view of the SFC represents the SFC in a CFC as a block with its interface. To distinguish it from CFC blocks and nested charts, the external view has the "SFC" icon  in the header.



Purpose

In the external view, you can assign parameters for and interconnect the interface of the SFC for AUTO mode so that the status of the SFC can be determined and the control signals can be derived from it for the SFC. This allows you to influence the processing of the SFC by means of the user program. If the SFC is used in MANUAL mode, only the inputs/outputs that are also processed in MANUAL mode are relevant in the external view.

Interconnection

You can provide the inputs/outputs with textual interconnection and/or interconnect them with compatible inputs/outputs of other objects or with shared addresses. You make all interconnections in the sheet bar. You cannot place any object, such as blocks, in this window.

Properties

You can open the object properties for the entire interface or for each individual I/O in the header of the external view. The Run Sequence window opens in the runtime properties box.

You can find more information on the interface in the sections:

Standard interface of the SFC (Page 167)

Inputs/outputs of the SFC sorted according to usage (Page 172)

5.6 SFC elements

5.6.1 What is a sequencer?

Sequencer

Sequencers enable status-dependent and event-driven execution in the SFC.

When it is created, each sequencer is given a consecutive number. This number is required for the "programmed target steps" and to interpret the outputs of the SFC in the CFC view.

An SFC can include a maximum of 8 sequencers and an SFC type a maximum of 32 sequencers, which can be controlled by defining different **start conditions**.

One sequencer at a time is displayed in the working window of the SFC. You switch to another sequencer via the tab at the lower edge of the window.

When you create a new SFC/type, a sequencer with the name "RUN" and the start condition RUN=TRUE is created automatically (Note: This corresponds to a chart of V5.x). The start conditions are formulated like transition conditions. You can find additional information in the section: What is a transition? (Page 43). An empty start condition – in contrast to the transition – is evaluated as FALSE, in other words, the sequencer is never executed.

In addition to this start condition, each sequencer also contains the "Priority" attribute (1 – 255) that is used to specify the start order if the conditions of several sequencers are satisfied simultaneously (in the "Start Condition" tab of the "Sequencer Properties" dialog box). If the conditions of multiple sequencers with the same priority are met simultaneously, the position of the sequencer in the tab determines the order of processing in the CPU (similar to the alternative branch; you will find information on this in the section Processing an alternative branch (Page 259)).

You can also configure an additional action for each sequencer. Each action consists of the following:

- **Preprocessing:** actions to be executed in every cycle after the sequencer has started and before the steps and transitions are processed
- **Postprocessing:** actions to be executed in every cycle after the steps and transitions are processed

This, for example, allows you to make pre-settings or to pass on the results of the sequencer execution.

You will find more information on configuring sequencers in the section How to configure multiple sequencers (Page 113)

5.6.2 What are sequencer elements?

Sequence path elements

An SFC consists of 1 to 8 sequencers (What is a sequencer? (Page 40)) and an SFC type of 1 to 32 sequencers, each with a series of sequence path elements (basic elements). This includes the following elements:

- Step
- Transition

There is another element outside a sequence (can be freely positioned):

- Text

The remaining elements are structures that are made up of different basic elements:

- Sequence
- Simultaneous branch
- Alternative branch
- Loop
- Jump

Identification of "Step" and "Transition"

The identification characteristic of the basic elements, step and transition, is a **name** that is unique within the sequencer. When the element is created, the editor enters a consecutive number that you can change to a name with a maximum of 16 characters. This name must not consist exclusively of numbers. The original number is retained for the element and is displayed in the object properties in the "Number" box. This is required for the "programmed target steps" and to interpret the outputs of the SFC in the CFC view.

You can use the optional **comment**, for example, to provide a text description of the respective functionality. A comment can take up several lines and contain up to 80 characters, however, only 16 characters are displayed to the right of the chart element.

If you position the cursor on the step or the comment, you can see a tooltip displaying the name with a maximum of 16 characters and a comment with up to 50 characters.

Resources

When programming and assigning parameters for steps, transitions, start condition and preprocessing and postprocessing in SFCs, you can access the interface of the SFC, the CFC block inputs/outputs, the runtime groups, other SFCs, and all shared addresses of the user program. With the textual interconnections, you can also access objects that do not exist yet. When you configure steps and transitions in SFC types, it is only possible to use the interface inputs/outputs of the SFC type in addition to the textual interconnections.

Screen display

The elements of an SFC are shown on the screen as follows:

- All the elements of a sequencer, including the links, are displayed in white with black print in the unselected and unedited state.
- Selected elements, including the links, are displayed in blue.
- Edited steps or transitions (whose object properties have been changed) are displayed in gray with black print.

Note

The colors indicated here are the default settings, some of which can be modified.

You can find additional information on this in the section: Default colors (Page 65).

5.6.3 What is a step?

Step

The step is a control entity for processing the assigned actions in the AS. You can configure up to three actions (Initialization, Processing, Termination) for each step.

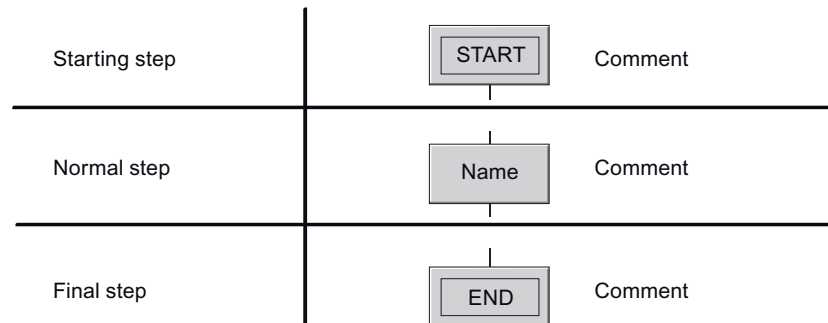
Action

An action is a series of statements formulated as follows:

- Assignments for assigning parameters of CFC blocks or shared resources, for example:
Setpoint temp:=100
XYZ.pump.on :=TRUE
- Enabling or disabling of an SFC or a runtime group, for example:
SFC_1.INTONOFF := TRUE
ABL_1.EN := FALSE

Step types

You can use one start step, a maximum of 253 normal steps, and one final step in each sequencer. When a sequencer is created, one start step, one final step and one transition are created automatically. These three basic elements form the initial state of a sequencer that you can then adapt by adding further chart elements.



The start step is activated at the start of the sequencer and the associated actions are executed according to the status of the successor transition. The final step does not have a successor transition; all actions are processed exactly once.

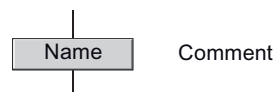
You cannot copy, cut or delete the start step or final step. This ensures that a sequencer always contains one (and only one) start step and final step.

All steps besides the start step and final step are normal steps.

5.6.4 What is a transition?

Transition

A transition contains the conditions with which a sequential control system passes control from one step to the next. Several conditions can be logically combined using Boolean operators. The result of the logic operation determines whether the next step is enabled.



Transition condition

The result of a transition condition in an SFC is derived from a Boolean expression that is formed by logic operations on the values of the following elements:

- Shared addresses
- Textual interconnections
- CFC block inputs/outputs
- Runtime group state
- SFC status

In SFC types, only the inputs/outputs of the SFC type interface can be used in addition to the textual interconnections.

During compilation, an empty transition is pre-assigned the default value TRUE. This pre-assignment is necessary because while the formulation of a condition is optional, the AS requires a defined value as a step enabling condition.

If several transitions are valid simultaneously, for example, in the alternative branch, for a loop or jumps, the system automatically assigns priority from left to right in descending order.

5.6.5 What is a text?

Text

You can use the "Text" chart element to insert any static text (free text) you require in a chart.

The text box can contain one-line or multiple-line character strings. For example, this allows you to insert descriptive texts into the SFC during the analysis phase that can later be replaced by automation functions.

Free texts are not embedded in the topological sequencer structure. If you change the topology, the text boxes are not repositioned; they maintain their position.

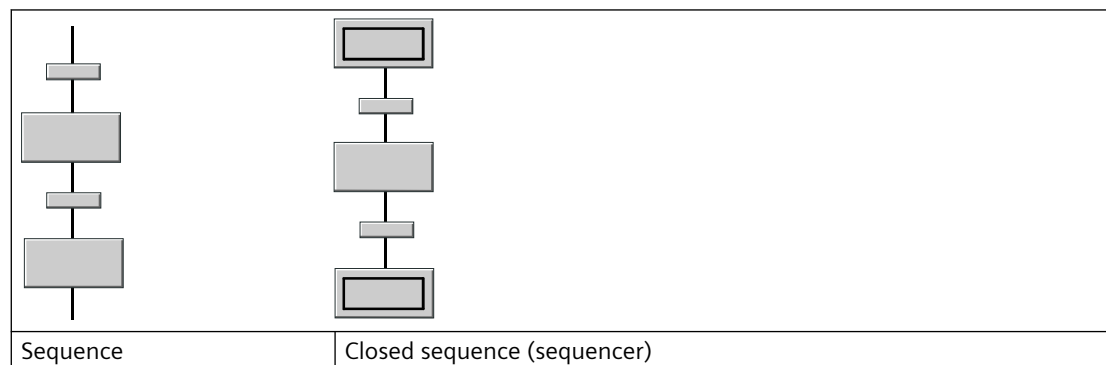
5.6.6 What is a sequence?

Sequence

A sequence is a series of steps and transitions that can be created with a selectable length and inserted in the sequential control system.

Sequencer

A closed sequence forms one sequencer in a sequential control system, for example between the divergence and convergence of a simultaneous or alternative branch. The entire sequencer is also designated as a sequencer in SFC. It runs from the start step to the final step.



5.6.7 What is a simultaneous branch?

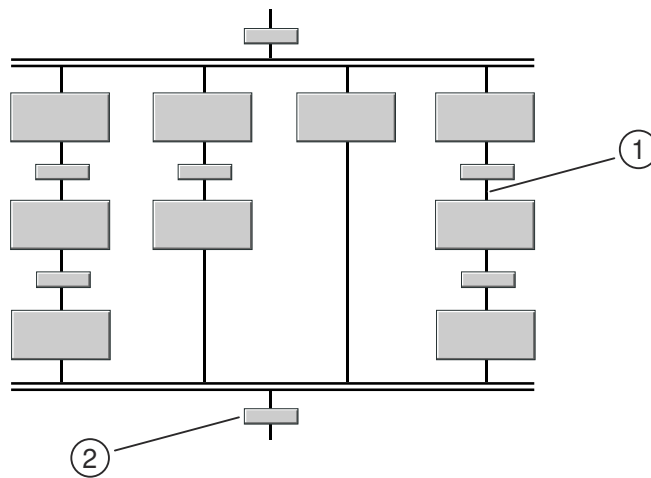
Simultaneous branch

If the execution is to be split into two or more sequencers that are to be executed at the same time, the simultaneous branch is used.

A simultaneous branch consists of at least two sequencers that are processed at the same time.

A simultaneous branch is always preceded by a transition or an alternative branch. The simultaneous branches end in a simultaneous convergence that is always followed by a transition or alternative branch.

The successor transition executes only when all actions of the steps at the end of every associated sequencer have been executed (apart from the "termination" action) and the step enabling condition is fulfilled (synchronization).



Key:	
(1)	Sequencer
(2)	Successor transition

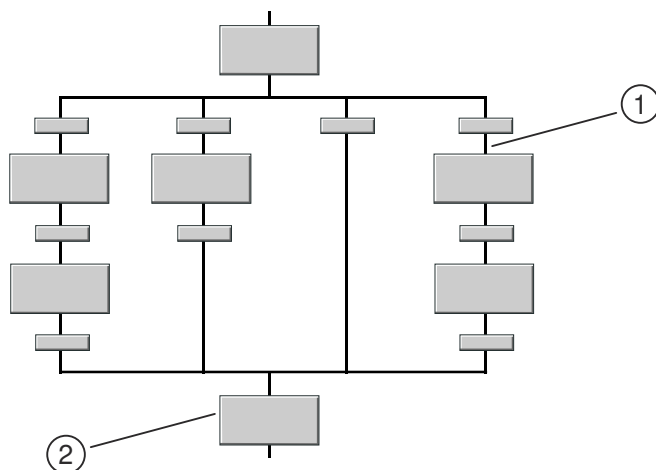
5.6.8 What is an alternative branch?

Alternative branch

If the execution is to be split up into two or more sequencers but only one of these sequencers is to be executed, an alternative branch is used.

An alternative branch consists of at least two sequencers, of which only one will be processed depending on the state of the first transitions in the individual sequencers. The sequencer whose transition is fulfilled first is selected. If more than one transition is fulfilled simultaneously, the sequencer furthest left with a fulfilled transition is executed.

The alternative branch can only be preceded and followed by a step or a simultaneous branch.



Key:

(1) Sequencer

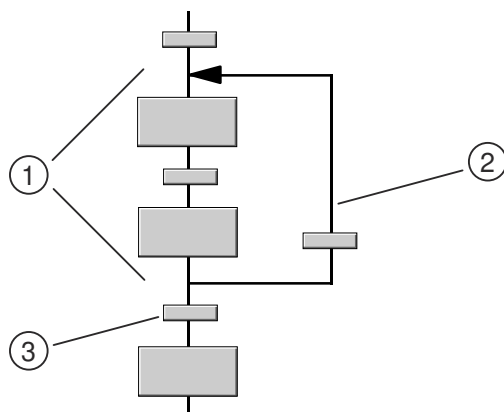
(2) Next step

5.6.9 What is a loop?

Loop

If a section of the sequencer is to be executed again, contingent on a transition, a loop is used.

A loop consists of a sequence and a return path with a transition that encloses a sequence. The start of the loop must be immediately following a step and the return path must converge again immediately before a step.



Key:

(1) Sequence

(2)	Return path
(3)	Successor transition

The transition of the return path is processed chronologically after the successor transition.

If the successor transition and the return path transition are fulfilled at the same time, the step or simultaneous branch located after the successor transition is processed.

Note

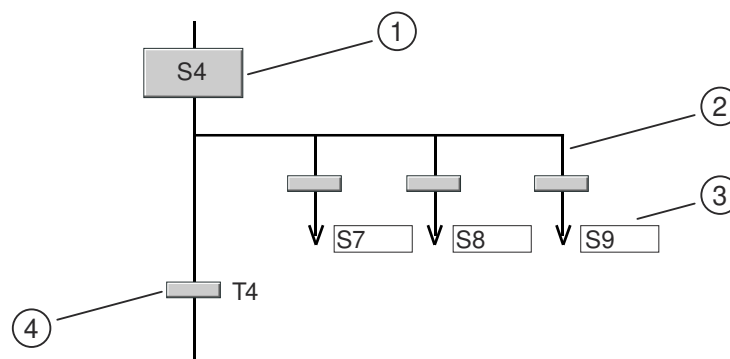
Return paths out of or into simultaneous or alternative branches are not possible.

5.6.10 What is a jump?

Jump

A jump can be used, contingent on a transition condition, to continue the execution of the sequencer at any step within the same sequencer.

A jump always leads immediately out of the sequence following a step (origin of the jump). Several jumps are also possible.



Key:	
(1)	Jump origin
(2)	Jump
(3)	Jump destination
(4)	Successor transition

A jump consists of an initiating transition and an arrow with the jump destination information. The jump destination specification is the name of the step at which the execution is continued when the transition is fulfilled. If the jump destination is still unknown, the name is "??".

The transition of the jump is processed chronologically after the successor transition.

If the successor transition and one or more jump transitions are fulfilled at the same time, the step (or simultaneous branch) located after the successor transition is executed.

The origin and destination of a jump must always be a step.

Note

Be aware of the possible consequences during execution in the AS when working with jumps to or from a sequencer of a simultaneous branch.

You can find additional information in the section:
How to create and edit a jump (Page 121).

Starting and working

6.1 Working with the SFC Editor

SFC Editor

You create your sequential control system using graphic tools in the SFC Editor by specifying actions and step enabling conditions. The elements of the SFC are positioned in the sequencer according to predefined rules. You do not need to be aware of details such as algorithms or the assignment of machine resources; you can concentrate solely on the technological aspects of your configuration.

After creating the sequencers as a series of steps and transitions, you proceed to configure the object properties of SFC, sequencers, steps and transitions. You configure the operating parameters of the SFC, the start conditions of the sequencers as well as the actions of the steps and the step enabling conditions of the transitions.

After configuration, you compile the executable machine code with SFC, download it to the CPU, and test it with the SFC test functions.

6.2 How to start the SFC Editor

Introduction

You can call the SFC editor directly or via the SIMATIC Manager.

Starting via the SIMATIC Manager:

Double-click the icon of the SFC you want to open.
The SFC Editor starts and opens the chart.

Starting via the Windows Desktop:

Double-click on the icon of the desired SFC on your desktop.

or

Select the program listing in the Start menu with the menu command **Start > SIMATIC > STEP 7 > SFC**.

6.3 Multiuser engineering

Configuring in the network

Multiuser engineering of projects or multiprojects is basically possible and allows you to configure, test and commission target systems from different remote locations or from a PC network (multiuser engineering).

You can find detailed information about multiuser engineering and its consequences in the section "Configuring in the network" of the *CFC for SIMATIC S7* documentation.

6.4 Navigating in the SFC

6.4.1 How to navigate in the SFC

Introduction

There are several ways in which you can navigate in an SFC.

Changing the display size

- Select one of the menu commands **View > Zoom > Zoom In / Zoom Out / Normal Size / Zoom Factor**
This allows you to change the size of the displayed SFC elements - and therefore also the visible section of the window.
or
- Use the icon for "Zoom In" and "Zoom Out" in the toolbar.



Changing the displayed section

You can change the displayed section as follows:

- Using the scroll bars
- With the **View > Center** menu command or by clicking the button
A square button icon with a circular arrow and a small square in the center, representing the 'Center' function.
- By dragging the SFC elements over the window border (automatic scrolling of the window section)
- Changing the window section (Page 64)

Switching between sequencers

If the SFC consists of several sequencers, you can switch from one sequencer to another as follows:

- Using the tabs in the bottom horizontal scroll bar
- Using the menu commands **Edit > Go To > Next Sequencer**
or **Edit > Go To > Previous Sequencer**

6.5 User interface and operator input

6.5.1 Elements of the user interface

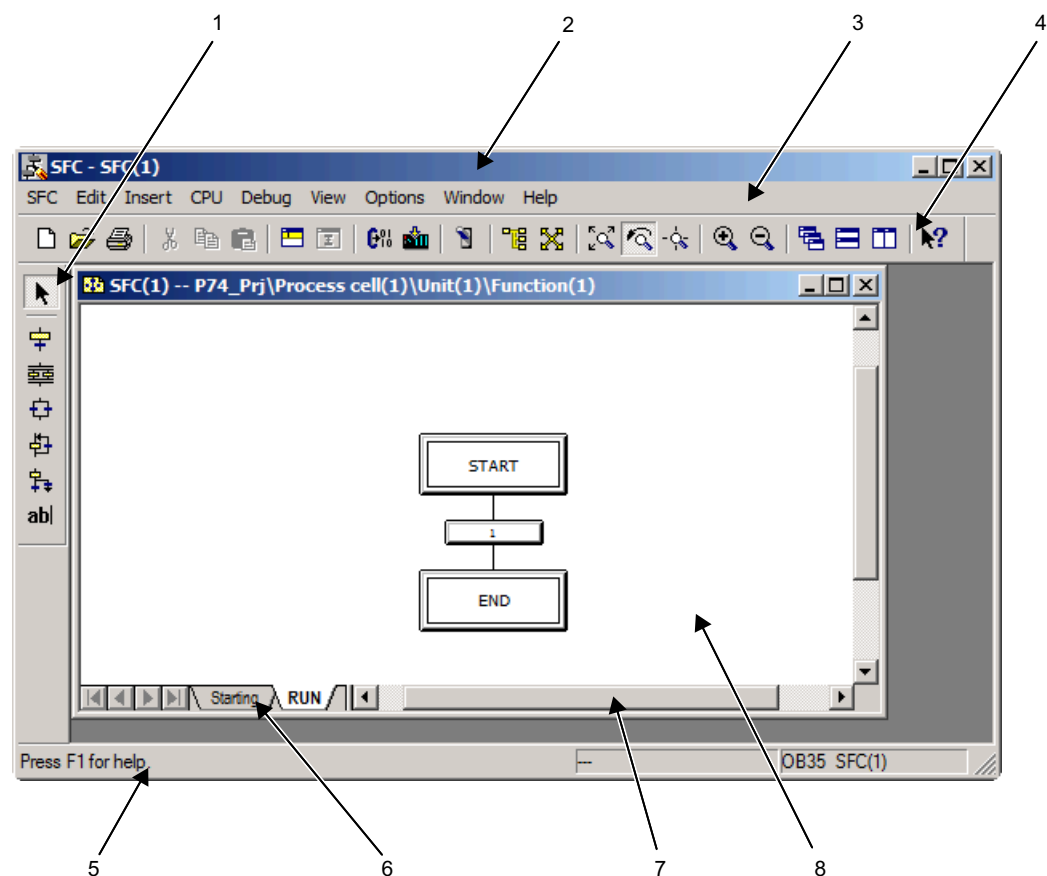
SFC Editor

The SFC Editor is a Windows application consisting of the following elements:

- Frame window with title bar
- Menu bar
- Toolbar
- Status bar
- At least one working window

You execute functions and operator inputs using menu commands (menu bar, shortcut menu) and in dialog boxes as needed.

It is generally advisable and more efficient to manipulate functions and objects using the mouse. Most operations can be also performed using the keyboard.



Key:	
(1)	Element bar
(2)	Title bar
(3)	Menu bar
(4)	Toolbar
(5)	Status bar
(6)	Tabs
(7)	Scroll bar
(8)	Working window

Windows

Within the SFC Editor, you can open as many windows as you wish (within the limits of Windows). Each window contains the view of an SFC (chart or type). You can also open several windows for the same SFC. The window in the foreground is always the active working window. This means that the menus and buttons in the toolbar always apply to this window.

A scroll bar is located at the lower border of the window, which you can use to move the content in the visible area of the window.

This bar also contains the tabs with which you can switch between the sequencers of the chart.

Title bar

The title bar of the SFC window contains the name "SFC" and the buttons that you normally find in Windows applications. If you display the SFC maximized to the full screen display, the name also appears in the title bar of the SFC window, because the title bar of the working window is no longer displayed. The title bar of the online window can be distinguished from that of the offline window by a different background and/or text color (can be set in SIMATIC Manager using the menu command **Options > Settings > "View" tab**).

Menu bar in the SFC window

The menu bar of the SFC window is always visible below the title bar. There are two different combinations of menus depending on whether or not an SFC is open.

- If no SFC is open, the window is displayed with an empty editing area and the menu bar is limited to four menus (**SFC, View, Options, Help**).
- If at least one SFC is open, the complete menu bar is displayed with all menu titles.
- If several charts are open, the commands of the menu bar relate to the active window.

Menu

A menu is a group of functions that can be selected and activated in the menu bar. Menu commands that are not applicable in a particular situation are displayed in gray and cannot be selected.

Opening a menu

You open menus by clicking the menu title or, using the keyboard, by typing the underlined character in the menu title while holding down the ALT key.

If you hold down the left mouse button, you can move through the menu bar and open and close the menus one after the other (browsing).

Shortcut menu

By right-clicking in the current context, you can display a shortcut menu containing frequently needed commands from the menu bar. You obtain a different shortcut menu with the following objects:

- Free area
- SFC element
- "Properties" dialog box in the editing window
- "Properties" dialog box for the selected line outside the editing window

Icons in the toolbar

The icons in the toolbar can be used to start some of the most important functions from the menus.

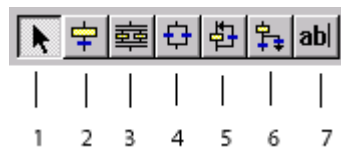
Depending on the current state of the editor, only the command icons that can be applied in a specific situation are activated.

If you hover the mouse pointer over an icon in the toolbar for a short time, the tooltip for the button is displayed. The status bar displays a more detailed description of the function of the button.

When you click the question mark icon, the cursor changes its appearance. You can now click a button, a menu command or an object in a chart. Help on the selected element will appear.

Icons in the element bar

The icons on the element bar represent the functions of the "Insert" menu.



Key:	
(1)	Activate selection
(2)	Insert step + transition
(3)	Insert simultaneous sequence
(4)	Insert alternative sequence
(5)	Insert loop

(6)	Insert jump
(7)	Insert text

Positioning the toolbar and element bar

You can freely position the toolbar and element bar any place on the desktop, as is usual in Windows. Press the left mouse button at any free space on the bar and drag it to the desired location. The arrangement of the icons is either horizontal or vertical depending on the position of the bar.

If you position the toolbar or element bar in the working area away from the edge, it is displayed as a dialog box with a "Close" button.

If you close the toolbar or element bar, you can display it again with the menu command **View > Toolbar** or **View > Element Bar**.

Status bar

The status bar is located at the bottom edge of the SFC window and displays important information.

The left part of the status bar displays contextual information, for example explanations of menu commands, operator prompts or error messages.

The right part of the status bar displays current information, such as the chart installation pointer. In test mode (online), the operating state of the CPU is displayed and also visualized by a color (red = STOP, green = RUN).

Optional layout

While the menu bar is always visible, you can toggle the display of the toolbar, element bar and status bar on or off, for example, to make more room on your screen. You can specify what is displayed by selecting the appropriate command in the "View" menu.

6.5.2 Dialog boxes

Operator control

When you activate some functions, a dialog box opens. Here, you can make settings and selections for the function you have activated. In the dialog boxes of modal dialogs, you cannot make selections or entries outside the box. You must first close the dialog box by clicking "OK" or "Cancel". The dialog boxes for "Properties" and "Browse", on the other hand, remain on the user interface while you continue to edit the sequencer topology or select other menu functions (except for activating test mode; here, the dialog boxes are closed).

Layout

Some dialog boxes contain tabs. This means that several pages (tabs) are cascaded on the screen. You can navigate from tab to tab as necessary.

For easy configuring, some dialog boxes have not only the standard buttons (e.g. OK, Cancel, Help) but also other buttons with useful functions.

In the "Object Properties" dialog box, you can use the buttons listed below within the sequencer to access the next element of the selected element type (step or transition) without having to open and close the box again each time. The jumps to the base elements take place in logical order and not in geometrical order.

Icon	Function
	You jump within a simultaneous or alternative branch from left to right to the first element of the next sequencer in each case.
	You jump from right to left to the last element of the next sequencer on the left in each case.
	You go from top to bottom within a sequencer. When a simultaneous or alternative branch is reached, only the left sequencer is executed.
	You go from bottom to top within a sequencer. When a simultaneous or alternative branch is reached, only the right sequencer is executed.

6.6 Operator inputs with the mouse

Mouse and pointer

The main tool for working with the SFC Editor is the mouse. You can perform most operations with the mouse, in particular those for selecting, moving, and inserting chart elements. In some situations, the mouse pointer changes its appearance and may take the shape of an arrow, hand or hourglass.

In addition to the standard pointer in insert mode, the SFC Editor also uses the icons of the chart elements to be inserted, as they are displayed in the toolbar or element bar. The icons also have a small cross to enable exact positioning (top left). This additional cross is referred to as a positioning crosshair in the following.

For move or copy functions, the positioning crosshair also appears along with a hand icon when a position is reached where the element is permitted.

Autoscrolling

When moving elements in a chart, you can scroll the displayed window if the insertion position of the element is not currently visible. Holding down the left mouse button, drag the object to the edge of the window so that the positioning crosshair is on the inner (horizontal or vertical) line of the window frame. The window is then scrolled until the limit of the chart is reached or until you pull the positioning crosshair away from the edge of the window.

Shortcuts using the mouse

- Object properties: Double-click on a step or transition
- Move: Select SFC elements + drag
- Copy and paste: Select SFC elements + <Ctrl> + drag
- Select all elements in the lasso: "Lasso" (press mouse button and drag)

Tooltips for steps and transitions

Instead of opening the object properties to get specific information, you can also do the following:

- Step: If you position the mouse pointer on a step, the name, number and – when available – the run times, comments and acknowledgement information are displayed.
- Transition: If you position the mouse pointer on a transition, the name, number and – when available – the comments are displayed.

6.7 Working with the keyboard

6.7.1 Working with the keyboard

Working with the keyboard

You can also perform most functions with the keyboard. You can find information about this in the following sections:

- Shortcut keys for menu commands (Page 59)
- Shortcuts for menu commands (Page 60)
- Navigating in the menu bar and menus with the keyboard (Page 61)
- Opening Help with the keyboard (Page 64)
- Navigating in dialog boxes with the keyboard (Page 62)
- Changing the window section (Page 64)
- Navigating in texts with the keyboard (Page 63)
- Selecting text with the keyboard (Page 63)
- Keyboard shortcuts in the sequencer (Page 64)

6.7.2 Shortcut keys for menu commands

Shortcut keys for menu commands

You can call every menu command via an appropriate key combination with the <Alt> key.

Press the following keys in the order shown

- <Alt>
- The letter underlined in the menu name you require (for example, <Alt+S> for the "SFC" menu - if the "SFC" menu is included in the menu bar). The menu opens.
- The underscored letter in the name of the required menu command (for example, <N> for the menu command "New").
If this involves a command with additional submenus, these will also be opened. Repeat the actions above until you have selected a complete menu command by typing in the relevant letters.

The menu command is executed after you have entered the last letter of the shortcut key.

Examples

Menu commands	Shortcut key
SFC > New	<Alt+S+N>
Options > Customize > Colors	<Alt+O+C+C>

6.7.3 Shortcuts for menu commands

Shortcuts for menu commands

Menu command	Menu	Function	Shortcut key
New	SFC	Create a new chart	<Ctrl+N>
Open		Open chart	<Ctrl+O>
Close		Close chart	<Ctrl+F4>
Check consistency		Check consistency of charts in the chart folder	<Ctrl+Alt+K>
Compile		Compile charts in the chart folder	<Ctrl+B>
Print		Print chart	<Ctrl+P>
Exit		Exit the SFC Editor	<Alt+F4>
Cut	Editing	Cut selected objects	<Ctrl+X>
Copy		Copy selected objects	<Ctrl+C>
Paste		Paste copied/cut objects	<Ctrl+V>
Delete		Delete selected objects	
Copy object properties		Copy object properties for step/transition	<Ctrl+Shift+C>
Paste object properties		Paste copied object properties in step/transition	<Ctrl+Shift+V>
Select All		Select all objects in the chart	<Ctrl+A>
Find		Search for SFC elements	<Ctrl+F>
Object properties		Open the "Properties – Step/Transition" dialog	<Alt+Return>
Go to, Next sequencer		Go to the sequencer that is after the current sequencer	<Ctrl+PgDn>
Go to, Previous sequencer		Go to the sequencer that is before the current sequencer	<Ctrl+PgUp>
Run sequence		Open "Run Sequence"	<Ctrl+F11>
Select	Insert	Switch on selection mode (cursor as arrow)	<Esc>
Step + [Transition		Activate insert mode for step + transition	<Ctrl+1>
Simultaneous branch		Activate insert mode for simultaneous branch	<Ctrl+2>

Menu command	Menu	Function	Shortcut key
Alternative branch		Activate insert mode for alternative branch	<Ctrl+3>
Loop		Activate insert mode for loop	<Ctrl+4>
Jump		Activate insert mode for jumps	<Ctrl+5>
Text field		Activate insert mode for text fields	<Ctrl+6>
Download	Target system	Download current program to the target system	<Ctrl+L>
Module Information		Display status of the current CPU	<Ctrl+D>
Operating mode		Display/change the operating mode of the CPU	<Ctrl+I>
Test mode	Debug	Activate/deactivate test mode	<Ctrl+T>
Overview	View	Select overview display	<Ctrl+U>
Detailed view		Select display at the last set zoom level	<Ctrl+Shift+U>
Zoom, Zoom in		Enlarge size of display	<Ctrl+Num + *>
Zoom, Zoom out		Reduce size of display	<Ctrl+Num - *>
Zoom, Normal size		Display in normal size (100%)	<Ctrl+Shift+N>
Update			<F5>
Settings, Display	Options	Open the "Layout" dialog box	<Ctrl+Alt+E>
Chart reference data		Open "Chart reference data"	<Ctrl+Alt+R>
Symbol table		Open the symbol table	<Ctrl+Alt+T>
Arrange, Cascade	Windows	Cascade windows	<Shift+F5>
Arrange, Horizontally		Tile windows from top to bottom	<Shift+F2>
Arrange, Vertically		Tile windows from left to right	<Shift+F3>
Context-sensitive help	Help	If there is a current context, for example, a selected menu command, the relevant help topic is opened. Otherwise the table of contents is displayed.	<F1>

*) "Num +" / "Num -" indicates the plus/minus keys on the numeric keypad

6.7.4 Navigating in the menu bar and menus with the keyboard

Navigating in the menu bar and menus with the keyboard

Function	Shortcut key
Move to the menu bar	<F10>
Move to the pop-up menu	<Shift+F10>

Function	Shortcut key
Select the menu containing the underscored character X	<Alt+X>
Select a menu command	Underlined character in the menu command
Move one menu command to the left	<Left>
Move one menu command to the right	<Right>
Move one menu command up	<Up>
Move one menu command down	<Down>
Activate the selected menu command	<ENTER>
Deselect menu or return to text	<Esc>

6.7.5 Navigating in dialog boxes with the keyboard

Navigating in dialog boxes with the keyboard

Function	Shortcut key
Move to the next text box (left to right and top to bottom)	<Tab>
Move one text box back	<SHIFT+Tab>
Move to the next line	<Ctrl+Down>
Move to the previous line	<Ctrl+Up>
Previous address	<Ctrl+Left>
Next address or in steps in the last address of the last line: Next line on next page	<Ctrl+Right>
Focus on line with this digit (single)	<Ctrl+Number>
Next page (with transitions and start condition of the sequencers)	<PAGE DOWN>
Previous page (with transitions and start condition of the sequencers)	<PAGE UP>
Move to the next tab	<Ctrl+PAGE DOWN>
Move to the previous tab	<Ctrl+PAGE Up>
To first address	<Ctrl+Home>
To last address	<Ctrl+End>
Activate/deactivate OS comment in line with this digit (single)	<Ctrl+Alt+Number>
Move to text box that contains the underscored character X	<Alt+X>
Move the cursor in a selection list	Arrow keys
Open a selection list	<Alt+Down>
Select or deselect an object in a list	<Space>
Confirm the entries and close the dialog box ("OK" button)	<ENTER>
Close the dialog box without saving changes ("Cancel" button)	<Esc>

6.7.6 Navigating in texts with the keyboard

Navigating in texts with the keyboard

Function	Shortcut key
One line up or one character to the left in a text consisting of only one line	<Up>
One line down or one character to the right in a text consisting of only one line	<Down>
One character to the right	<Right>
One character to the left	<Left>
One word to the right	<Ctrl+Right>
One word to the left	<Ctrl+Left>
To the beginning of the line	<Home>
To the end of the line	<End>
To the previous page	<Page Up>
To the next page	<Page Down>
To the beginning of the text	<Ctrl+Home>
To the end of the text	<Ctrl+End>

6.7.7 Selecting text with the keyboard

Selecting text with the keyboard

Function	Shortcut key
One character to the right	<Shift+Right>
One character to the left	<Shift+Left arrow>
One word to the right	<Ctrl+Shift+Right>
One word to the left	<Ctrl+Shift+Left arrow>
Text to beginning of line	< Shift+Home>
Text to end of line	<Shift+End>
One line up	< Shift+Up>
One line down	< Shift+Down>
To the previous page	<Shift+Page Up>
To the next page	<Shift+Page Down>
To the beginning of a file	<Ctrl+Shift+Home>
To the end of a file	<Ctrl+Shift+End>

6.7.8 Keyboard shortcuts in the sequencer

Keyboard shortcuts in the sequencer

- <Up> selects the previous SFC element in the sequence
- <Down> selects the next SFC element in the sequence
- <Left> selects the SFC element further left in the sequence
- <Right> selects the SFC element further right in the sequence

6.7.9 Opening Help with the keyboard

Opening Help

Function	Shortcut key
Opens Help	<F1> If there is a current context, for example, a selected menu command, the relevant help topic is opened. Otherwise the table of contents is displayed.
Closes the help window, back to SFC Editor.	<Alt+F4>

6.7.10 Changing the window section

Changing the window section

Function	Shortcut key
Scroll window section up	<Page Up>
Scroll window section down	<Page Down>
Scroll window section right	<Ctrl+Page Up>
Scroll window section left	<Ctrl+Page Down>
To top of window	<Ctrl+Home>
To bottom of window	<Ctrl+End>
To left edge of window	<Home>
To right edge of window	<End>

6.8 Default colors

Overview

Defaults (Change these defaults with the menu command Options > Customize > Colors...).	
Selection	Blue
Step/transition [transition] parameters assigned [object]	Gray
Step/transition parameters assigned [name]	Black
Step/transition no parameters [object]	White
Step/transition no parameters [name]	Black
Step/transition [comment]	Black
Step/transition selected [name]	White
Step ACTIVE [object]	Green
Step ACTIVE [name]	Black
Step EXECUTED [object]	Dark green
Step EXECUTED [name]	White
Step HELD [object]	Yellow
Step HELD [name]	Black
Step ERROR [Object]	Red
Step ERROR [name]	Yellow
Transition FALSE [object]	Red-brown
Transition FALSE [name]	White
Transition TRUE [object]	Dark green
Transition TRUE [name]	White
Operator prompt	Gray
Error confirmation	Red
Defaults that cannot be changed:	
Textual interconnection [object]	Yellow
Textual interconnection [name]	Black
Step INACTIVE [object]	*)
Step INACTIVE [name]	*)
Transition INACTIVE [object]	**)
Transition INACTIVE [name]	**)
Windows default: (Modify with the right mouse button on Desktop: Properties > Display > Extended > Element: Selected elements)	
Background of the tab ID for tabs with content	Blue
*) Color is identical to the color for the unselected step and cannot be changed	
**) Color is identical to the color for the unselected transition and cannot be changed	

6.9 Data backup in the SFC Editor

Data backup in the SFC Editor

All changes made in the SFC Editor are saved immediately - there is therefore no other save option in SFC. This means that you can no longer undo or cancel changes in the SFC Editor by closing the editor without saving. This does not apply for the configuration of sequences, steps, and transitions in the properties dialog box. There you can "discard" changes.

To back up your data, copy the entire program folder to a backup project. You can then revert to old versions at any time. This also allows you to archive your complete configuration.

Create the project structure

7.1 How to create a project structure

Optional procedures

Use the SIMATIC Manager to create a project. The following options are available in the SIMATIC Manager for creating a project:

- **PCS 7 Wizard**
With the PCS 7 Wizard, you create a single project or multiproject in the Plant view and in the Component view. In addition to the actual project, a multiproject also includes a master data library.
- **STEP 7 Wizard**
With this wizard you create a STEP 7 project with the following elements:
 - The SIMATIC station
 - The CPU
 - The S7 program
 - The block folder
- **Menu commands**
With the menu commands you create a new project and insert all the required components.

We recommend the procedure using the **PCS 7 Wizard** (point 1) because it is both faster and is less liable to errors.

You can select the PCS 7 Wizard or the STEP 7 Wizard in the SIMATIC Manager in the "Wizards" tab with the menu command **Options > Settings.....**

Creating a project with the PCS 7 Wizard

1. Start the PCS 7 Wizard (if it does not start automatically) with the menu command **File > 'New Project' Wizard**.
2. Follow the steps in the wizard.
In the dialog "Which objects will be used in the project?", the "SFC" option in the "AS objects" box is set by default.

The hardware configuration is made largely automatically and only the CPUs permitted for PCS 7 are available to you.

Creating, configuring and managing SFCs

8.1 Overview of creating, configuring and managing SFCs

Overview

You can create new charts and types, open them for editing, and modify their properties in SIMATIC Manager and in the SFC Editor. You can copy and delete charts and types only in the SIMATIC Manager. You copy and delete instances in the relevant CFC.

The SFC can be displayed in CFC in its external view, allowing the SFC to be controlled by CFC interconnections. The external view displays the SFC like a block, which means with the standard interface.

Requirement for creating SFCs and SFC types

In the SIMATIC Manager, the desired project structure must be created.

Each newly created chart or type is assigned a default name, which you can later change. If you change the name, the system checks to ensure that it is unique in the CPU. Note the following when assigning names:

- The names of SFCs can contain a maximum of 22 characters.
- The names of SFC types can contain a maximum of 16 characters. Although you can enter 24 characters in the properties, when the instances are created, only 16 characters are permitted.
- Illegal characters are: " % . / \

You will find more information on creating project structures, charts and types in the sections:

How to create a project structure (Page 67)

How to create an SFC (Page 71)

How to create an SFC type (Page 72)

Requirements for creating SFC instances

You create an SFC instance from an SFC type. The following requirements must be met:

- The SFC type that you want to use as an instance is located in the chart folder of the current S7 program.
- The CFC in which you want to insert and interconnect the instance is open.
- The same naming convention as that for the SFC type applies, but you can only use 16 characters for the instance name.

You can find more information on creating instances in the section:

How to create an SFC instance (Page 73)

8.1 Overview of creating, configuring and managing SFCs

You can find more information on handling charts in the sections:

How to adapt chart properties (Page 89)

How to adapt type properties (Page 91)

How to adapt instance properties (Page 95)

Copying and moving SFCs (Page 85)

Copying and moving SFC types (Page 86)

Copying and moving SFC instances (Page 86)

How to delete SFCs and SFC types (Page 88)

How to delete SFC instances (Page 88)

8.2 Creating an SFC

8.2.1 How to create an SFC

Introduction

You create a new SFC in the following ways:

- in the Component view, Plant view or Process Object view of the SIMATIC Manager
- in the SFC Editor

Creating a chart in the SIMATIC Manager

1. In the Component view of the SIMATIC Manager, select the project (the SIMATIC station, the CPU, the S7 program) and the chart folder.
If the chart is to be assigned to a plant hierarchy folder, select the project and the hierarchy folder from the Plant view or Process Object view.
2. Open a chart folder in the Component view or select a hierarchy folder in the Plant view or Process Object view.
3. Insert a chart into the chart folder or the hierarchy folder with the menu command **Insert > S7 Software > SFC**.
The chart is given a default name by the system, for example SFC(1), which you can change.

Creating a chart in the SFC Editor

1. Select the menu command **SFC > New..** in the SFC.
The "New" dialog box opens.
2. Select the project (the SIMATIC station, the CPU, the S7 program) and the chart folder in the dialog box.
3. If the chart is to be assigned to a plant hierarchy folder, select the project and the hierarchy folder in the "Plant view" or "Process Object view".
4. Select "SFC" from the drop-down list in the "Object type" box.
5. Enter a chart name in the "Object name" box.
6. Click "OK".

A new window opens with the chart (1 sequencer, initial state).

8.3 Creating an SFC type and creating an SFC instance

8.3.1 How to create an SFC type

Introduction

You create a new SFC type in the following ways:

- In the Component view of the SIMATIC Manager
- in the SFC Editor

Creating a type in the SIMATIC Manager

1. Open the **chart folder** in the **Component view** of the SIMATIC Manager .
2. Select the menu command **Insert > S7 Software > SFC Type**.
The SFC type is inserted into the chart folder.

Creating a type in the SFC Editor

1. Select the menu command **SFC > New**.
The "New" dialog box opens.
2. Select the project (the SIMATIC station, the CPU, the S7 program) and the chart folder in the dialog box.
3. Select "SFC type" from the drop-down list in the "Object type" box.
4. Enter a chart name in the "Object name" box.
5. Click "OK".
A new window opens with the SFC type (a sequencer in the initial state).

Result

The type is given a default name by the system, for example SFC_Type(1), which you can change. The next free FB number is automatically reserved for the SFC type and the type template (FB 247) is copied to the block folder with this number. This allows you to configure messages and instances for the type once it is created without having to compile the type. The FB number can be modified later in the object properties dialog box.

When you first create an SFC type, the blocks required for compilation and for execution in the AS are copied to the current program and then managed in the ES. The blocks are included in the supplied block library.

Note

SFC types cannot be assigned to a hierarchy folder in the plant view because they are not relevant to execution (from the perspective of the process to be automated).

8.3.2 How to create an SFC instance

Introduction

The SFC types available in the chart folder are displayed in the CFC block catalog. They are either in the "All blocks" directory and in the directory of the family, if they are assigned to a family; otherwise, they are in the "Other blocks" directory.

You can also insert SFC types in the CFC from the libraries (for example, SFC library or master data library).

Procedure

In CFC, drag the SFC type into the CFC from the block catalog or from a library.

Result

The SFC instance is represented like a CFC instance block. If there is not enough space to position the SFC instance, it is displayed as an "overlapping block" (light gray and without visible inputs/outputs). After moving them to a free position in the chart, the overlapping blocks are displayed as normal blocks again.

You can rename, assign parameters, and interconnect the SFC instance in the CFC.

8.4 Configuring an SFC

8.4.1 Basics for configuring an SFC

Introduction

In SFC, there is initially the "SFC" object type along with the "SFC type" object type. The SFC allows the definition of sequential control systems using its predefined interface and any process signals of the plant being automated. The SFC must first be compiled and then downloaded to the automation system where it is executed.

Basics of configuration

The SFC has an interface that is created when the SFC is generated and includes the standard interface derived from the SFC template (block @SFC_RTS). This is required to provide SFC system functionality (operating modes, operating states, step control modes, etc.) at the interface of the SFC.

You cannot move or delete the elements of the standard interface. Neither can you add new elements to the standard interface. You can edit the initial value, comments, and attributes.

During configuration, you can use the interface I/Os and any required process signals to formulate step assignments or the transition and start conditions. Here, textual interconnections are also possible.

The SFC is therefore not self-contained and can only be used once. If you wish to reuse the SFC, copy the SFC and adapt it fully to the "new environment" because the process signals used must usually be replaced by others.

You can find more information on creating an SFC in the section: How to create an SFC in overview (Page 23).

You can set or change the following in the "Properties" dialog box:

- The "General" properties (name, author, comment, OS comment)
- The "AS operating parameters" (default settings: Step control mode, Operating mode, Command Output, Cyclic operation, Time monitoring and the start options: Autostart, Use default operating parameters when SFC starts)
- The option whether or not the SFC is available on the OS for operator control and monitoring

Runtime properties

An SFC is inserted in the run sequence in order to determine when the SFC is to be processed in the automation system.

You can find additional information on this in the section: Runtime properties of an SFC (Page 162)

Messages

You have the option of configuring seven messages requiring acknowledgment for an SFC in the SFC Editor (menu command **SFC > Message...**). The SFC itself needs the remaining available messages (one message requiring acknowledgment and one that does not).

You can find additional information on this in the section: How to configure messages in the SFC (Page 97)

Footer

You can configure an optional footer for an SFC.

You can find additional information on this in the section: Defining footers (Page 291)

Copying, moving, deleting

You can copy, move, or delete SFCs in the SIMATIC Manager.

You can find additional information about this in the following sections:

Copying and moving SFCs (Page 85)

How to delete SFCs and SFC types (Page 88)

The attributes required for the SFC (for example, S7_m_c) are preset.

Compiling and downloading

The SFC is compiled in the "Compile program" context and downloaded in the "Download to CPU" context.

You can find additional information about this in the following sections:

Overview of compiling charts, types, and instances (Page 217)

How to download programs (Page 225)

Testing

In the SFC test mode, the execution of the SFC is made dynamic and can be influenced with the test functions.

You can find additional information on this in the section: Operator control and monitoring during testing (Page 265)

8.4.2 Configuration changes in the SFC

Description

You can change the topology (step/transition sequences, jump destinations) or the configured steps/transitions in the SFC. The changes do not take effect until the chart has been compiled and downloaded.

8.4 Configuring an SFC

Following a topology change, you can download inactive sequencers at any time. You need to disable the SFC to download active sequencers.

You can download changes in the step and transition configuration at any time even if the SFC is currently being processed in the automation system.

The basic rule is that changes in the SFC that prevent or restrict a download of changes in RUN mode can be made only after the user has confirmed a prompt.

Further procedure

After completing the changes, you must compile and download the OS to ensure that the current data is available on the OS.

8.5 Configuring the SFC type and SFC instance

8.5.1 Type/instance concept of SFC

Overview of SFC type and SFC instance

The type/instance concept was introduced with SFC V6.0. This makes it possible to create sequential control system types that generate SFC instances when placed in a CFC.

The SFC type is not executable on its own. Like a function block type, an SFC type must be placed in a CFC to obtain a sequence-relevant object, in this case, an SFC instance. To execute an SFC instance, both the SFC type and the SFC instance are downloaded to the automation system.

You can find additional information in the section "Configuring the SFC type (Page 77)".

8.5.2 Configuring the SFC type

Introduction

The SFC contains the "SFC type" object type, in addition to the "SFC" object type. The SFC type allows the definition of sequential control systems including an interface. The sequential logic of the SFC type is based only on the interface I/Os of the SFC type. This is why the SFC type cannot access all process signals, in contrast to the SFC.

Overview of configuring the SFC type

The SFC type has an interface corresponding to the SFC. It is created when an SFC type is generated and already contains the SFC type standard interface, derived from the SFC type template "@SFC_TYPETEMPLATE" (FB 247). The standard interface is required to provide SFC system functionality (for example, operating modes, operating states, step control modes, etc.) at the interface of the SFC type.

You can add additional inputs/outputs to the standard interface in the Interface editor and additional characteristics in the Characteristics editor.

Note: The number of inputs is limited to 800 for the SFC type.

You can find more information on characteristic in the section: Using the characteristics editor and the interface editor (Page 99)

You cannot move or delete the elements of the standard interface or the characteristics. You can change the initial value, comments, and attributes.

Explicitly created elements can be manipulated as required.

During configuration, you can only use the interface inputs/outputs to formulate step assignments or the transition and start conditions. Addresses in assignments or conditions are therefore always references to inputs/outputs of the interface. Here, textual

interconnections are also possible. This means the SFC type is self-contained because there is no external access from the SFC type that bypasses the interface.

There are two ways of creating and changing an SFC type:

- **Creating and changing in a library**
This has the advantage that the master for the SFC type is always in the library and that the test project can still run until a new version of the SFC type is adopted.
- **Creating and changing in a project**
This has the advantage that every change to the SFC type can be checked immediately because you are working directly with the master.

You can find additional information in the section: How to create an SFC type in overview (Page 26)

Configuring runtime properties

An SFC type does not have any runtime properties, because it is not relevant to execution of the program. An SFC type **cannot** be inserted in the run sequence.

Configuring messages

You can configure seven messages that must be acknowledged and five that do not require acknowledgment for an SFC type. The SFC type itself requires the remaining available messages (one message for each message type and 10 status messages for SIMATIC BATCH).

You can find additional information on this in the section: How to configure messages in the SFC (Page 97)

Configuring footers

You can configure one footer for an SFC type.

You can find additional information on this in the section: Defining footers (Page 291)

Compiling/downloading

The SFC type is compiled in the "Compile program" context and downloaded to the CPU in the "Download to CPU" context.

You will find more information on this subject in the following sections:

Overview of compiling charts, types, and instances (Page 217)

How to download programs (Page 225)

Tips

The following tips may help you:

- In the "SFC Library", you can find the "TypeStates" SFC type with the menu command **SFC Library > Blocks+Templates > Templates**. It already contains several sequencers for state-oriented processing of the sequential control system. You will also find the "TypeCtrlStrategy" SFC type containing control strategy-oriented processing of the sequential control system. You can copy these templates and change them to suit your purposes.
- When creating an SFC type for **SIMATIC BATCH**, remember the following points:
 - In **Properties > Operating Parameters AS** for the SFC type, you need to select the "AUTO" mode, otherwise the control commands from SIMATIC BATCH will have no effect (this also applies to the SFC instances).
 - In **Properties > Options**, you have to set one of the categories, "EOP" or "EPH", otherwise the SFC type remains "invisible" to SIMATIC BATCH.
 - If you wish to use the "Continuous mode" of the SFC, you have to set inputs/outputs ENASTART=1 and SELFCOMP=0. You must take into account the READY_TC input/output when configuring the start condition for sequencers (this also applies to SFC instances). For additional information, refer to the section: Start requirements of an SFC (chart/instance) (Page 254)

8.5.3 Configuration changes in the SFC type

Effects of the modified SFC type

Modifications to the interface of the SFC type are transferred to the SFC instances immediately. Changes to the interface include adding and deleting I/Os and changing data types. You can only download the SFC type and its instances during RUN mode of the automation system if all the SFC instances of this SFC type are disabled or if you temporarily disable the SFC instances during downloading. The instances are disabled during the download following operator confirmation and restarted after the download, again following operator confirmation. The execution of the instance then depends on the process state and on the configuration of the instances (especially the start conditions).

Note

If you perform the download with the SIMATIC Manager function "Compile and download objects" and the SFCs need to be disabled for this, the SFCs will not be disabled. In this case, the download will not be performed and an error message is entered in the log.

While changes are being downloaded, the SFC instances cannot be processed in the automation system, and the SFC instances cannot be accessed via the interconnections in the CFC.

You do not have to disable the SFC instances to rename I/Os and characteristics or to change units and limits.

Changes in the topology (step/transition sequence, jump destination) and the step or transition configurations are made in the SFC type and do not become effective in

the SFC instances until they are compiled and downloaded. You can download inactive sequencers following changes to the topology at any time. If you want to download active sequencers following changes to the topology, you must first disable the SFC instances before downloading. You can download changes to the step and transition configuration at any time even if SFC instances of the SFC type are currently being processed in the automation system.

After changing the configuration, you need to compile the OS to ensure that the current data is available on the OS.

Effects on SFC instances in the CFC

If you change an SFC type and the instances belonging to it are open in the CFC, you will have to update (**View > Update** or **F5**) the CFC and therefore the SFC instances after compiling and downloading the changes. If the CFC is in test mode, you must leave this mode before performing the update.

Effects on SFC instances in the SFC

If you want to change an SFC type and the instances belonging to it are opened in the SFC and are in test mode, you must leave this mode before changing the type. After making the change and compiling and downloading the change, you will have to update the SFC instances (**View > Update** or **F5**). You can then re-enable test mode.

8.5.4 Configuring the SFC instance

Introduction

An SFC instance is derived from an SFC type. The SFC type is first inserted into a CFC similar to a function block type in the CFC. The SFC instances are therefore always assigned to one CFC and are addressed using the chart. SFC instances are displayed as blocks, same as CFC instances, which means their interface is visible in the CFC.

SFC instances are not displayed in the SIMATIC Manager, because they can only be addressed by means of the CFC. With the assignment of the CFC to the plant hierarchy, the SFC instances contained are also indirectly assigned to the plant hierarchy.

Creating and handling

SFC instances are created by placing an SFC type in the CFC. The SFC types in the chart folder are displayed in the CFC "Blocks" catalog (in "All blocks" and in the folder of the family if they are assigned to a family, otherwise in the "Other blocks" folder). You can also insert SFC types from the libraries (for example, SFC library, master data library) in the CFC.

You can configure and interconnect an SFC instance similar to a CFC instance.

SFC instances are handled in the same way as CFC instances in the CFC in which they are placed.

Runtime properties

The runtime properties of an SFC instance are identical to the runtime properties of CFC block instances.

Messages

You can configure the messages of an SFC instance in SFC (menu command **SFC > Message...**) and in CFC using the "Object Properties" dialog box.

Opening and testing

You can open SFC instances in the CFC. The SFC opens and the first sequencer is displayed. You cannot change the sequencer. In SFC test mode, the execution of the instance is made dynamic and can be influenced in the same way as in SFCs.

Copying, moving, deleting

You can copy/move SFC instances in the CFC or between CFCs.

You delete SFC instances in the CFC or indirectly by deleting the CFC in the SIMATIC Manager.

You will find more information on this subject in the following sections:

Copying and moving SFC instances (Page 86)

How to delete SFC instances (Page 88)

Configuration

The configuration options for SFC instances in the SFC are limited compared to SFC types. The interface and the sequencers cannot be changed in the SFC instance because they are specified identically for all instances based on the SFC type. However, you can change the parameters (initial value and comments) of the interface I/Os for each specific instance.

You can make the following settings in the "Properties" dialog box:

- "General" properties (name, comment).
- "AS operating parameters" (these are the default settings: Step control mode, Operating mode, Command Output, Cyclic operation, Time monitoring and the start options: Autostart, use default operating parameters when SFC starts).
- The "options" for SIMATIC BATCH (category, allow operator instructions) cannot be modified here; the configured control strategies can be enabled or disabled here for the instance.

You can open the "Properties" dialog box as follows:

1. Select the SFC instance in the CFC.
2. Open the shortcut menu and select the menu command **Open**
The SFC instance opens in the SFC.
3. Select the menu command **SFC > Properties...**
The "Properties SFC Instance" dialog box opens.

Note

When using SIMATIC BATCH, SFC instances may not be changed at a later time.

Compiling/downloading

SFC instances are compiled in the "Compile program" context and downloaded to the CPU in the "Download to CPU" context.

You will find more information on this subject in the following sections:

Overview of the blocks generated during compilation (Page 221)

Overview of compiling charts, types, and instances (Page 217)

How to download programs (Page 225)

Tips:

- You can use the "SFC Visualization" add-on package on the OS for operator control and monitoring of the SFC instances (and SFCs). You also use "SFC Visualization" to perform the necessary configuration steps for SFC operator control and monitoring. You can find additional information on SFC Visualization in the *SFC Visualization for S7* manual or the "Options" folder of the WinCC online help.

8.5.5 Configuration changes in the SFC instance

Description

Due to the limited configuration options for SFC instances, any changes you make directly in the SFC instances can be downloaded at any time, even if the SFC instances are currently being processed in the automation system.

Further procedure

After completing the changes, you must compile and download the OS to ensure that the current data is available on the OS.

8.6 Opening

8.6.1 How to open SFCs or SFC types

Introduction

You can open an SFC or an SFC type in both the SIMATIC Manager and SFC Editor.

Opening an SFC or SFC type in the SFC Editor

1. Select the menu command **SFC > Open** in the SFC Editor.
2. Select the project (the SIMATIC station, the CPU, the S7 program) and the chart folder containing the SFC or SFC type you want to open.
3. Select the "SFC" object type for the chart or the "SFC type" for the type.
4. Click the chart or type you want to open in the right window of the dialog box.
5. Click "OK".
The window with the SFC or SFC type opens.

Note

The "SFC" menu displays the last four objects edited (SFCs and/or SFC types) as menu commands. If you select one of these menu commands, the corresponding object opens.

Opening an SFC plan or SFC type in the SIMATIC Manager

1. Select the menu command **View > Component View** in the SIMATIC Manager and then select the project and the chart folder in the S7 program.
2. Open the chart folder.
3. Double-click on the SFC or the SFC type in the right window.
The SFC Editor starts (if not already started) and the chart or type is displayed in an SFC Editor window.

Note

If the project is assigned a plant hierarchy, you can also open SFCs in the **Plant view** or **Process Object view** in the SIMATIC Manager. In this case, you open the hierarchy folder that contains the chart instead of the chart folder.

SFC types are not included in the plant hierarchy because they are not relevant to execution from a technological perspective.

8.6.2 How to open SFC instances

Introduction

You can only open SFC instances in the CFC.

Procedure

1. Select the SFC instance in the respective CFC.
2. Select the **Edit > Open** menu command.
The SFC Editor starts (if it is not already started) and the SFC instance is displayed in a window of the SFC Editor. The displayed topology is determined by the associated SFC type.

8.7 Copying

8.7.1 Copying and moving SFCs

Overview of copying and moving

You can copy charts with the SIMATIC Manager . You can transfer tested partial or complete structures from one CPU to another CPU or even within a single CPU. Existing references are not lost if the relevant charts are copied together at one time.

You can not only copy individual charts but also an entire chart folder with all the charts it contains.

You can also move charts across CPUs. The information contained in the section "Copying/moving across CPUs" applies to moving.

Copying within the CPU (in the chart folder)

Note the following when copying within the CPU:

- Interconnections and cross-accesses between CFCs and between SFC and CFCs are also copied if the charts involved are copied at the same time.
- Chart names must be unique in a chart folder; it may be necessary to rename files.
- Special considerations when copying SFCs:
 - All SFC statements and SFC transition conditions that access uncopied blocks in CFCs will continue to access the original blocks.
 - All SFC statements and SFC transition conditions that access global resources (via symbolic addresses of the symbol list) will continue to access these resources.

Copying/moving across CPUs

When copying/moving charts across CPUs, remember the following points:

- Interconnections
Interconnections to shared addresses and to runtime groups are not copied unless you copy a complete program folder including user program, chart folder and symbol table.
- Block types
Unless all the block types are identical in the destination CPU (name, block I/Os), the copy/move function is rejected. You must then first copy the relevant block types to the block folder of the destination CPU and import them into a CFC of the destination chart folder.
- Task
A task with the same name is used to insert the copied/moved blocks. The system does not check whether the tasks with the same name on both CPUs actually have the same properties. If a task with the same name does not exist on the destination CPU, copying/moving is rejected. You must then configure a task with the same name. If the destination system is an S7 CPU, the task is an OB number; any name in the symbol table will be considered as a comment and has no relevance.

8.7 Copying

- Chart names, block names
Because CFC block names are unique within a chart, it is not necessary to rename them. It may be necessary to rename charts, however.
- Special consideration when copying/moving SFCs
All SFC statements and SFC transition conditions that access global resources (by means of symbolic addresses of the symbol list) access the same resources on the target CPU. If these resources do not exist on the destination CPU, they are automatically converted to textual interconnections (see above).

You can find more information on textual interconnections in the section "Working with textual interconnections" of the *CFC for SIMATIC S7* documentation.

8.7.2 Copying and moving SFC types

Copying SFC types

The runtime objects associated with the SFC type are also copied. If the SFC type is not up-to-date because the time stamp of the FB is older than the time stamp of the SFC type, you receive a message. If there is an SFC type with an identical name in the destination folder, it is overwritten following an acknowledgment prompt and any deviating properties for the existing type are passed to the SFC instances.

Moving SFC types

You can only move an SFC type when there are no SFC instances for the SFC type in the source. The runtime objects associated with the SFC type are also moved. If there is an SFC type with an identical name in the destination folder, it is overwritten following an acknowledgment prompt and any deviating properties for the existing type are passed to the SFC instances.

8.7.3 Copying and moving SFC instances

Introduction

You can copy or move SFC instances within a CFC, between CFCs or indirectly by copying/moving the CFC in the SIMATIC Manager. The runtime objects associated with the SFC instance are also copied or moved.

Copying SFC instances

You can copy an SFC instance within a CFC or between CFCs in the same chart folder. If you copy a CFC within a chart folder, the SFC instance is also copied. The runtime objects associated with the SFC instance are also copied.

If you copy an SFC instance between CFCs from different chart folders or copy a CFC to a different chart folder, the SFC type is also copied.

Moving SFC instances

If you move an SFC instance between CFCs in the same chart folder, the SFC instance is moved and the associated runtime objects are retained.

If you move an SFC instance between CFCs from different chart folders or move a CFC to a different chart folder, the SFC type is also copied or moved.

8.8 Deleting

8.8.1 How to delete SFCs and SFC types

Deleting an SFC or SFC type

You delete SFCs and SFC types in the SIMATIC Manager.

You can only delete SFC types if there are no SFC instances for the SFC type.
If there are instances for an SFC type, a message will appear indicating this.

The runtime objects associated with the SFC type are thereby also deleted.

Procedure

1. Select the SFC or SFC type in the Component view of the SIMATIC Manager .
2. Press the key.

8.8.2 How to delete SFC instances

Deleting an SFC instance

You delete SFC instances in the associated CFC or indirectly by deleting the CFC in the SIMATIC Manager.

The runtime objects associated with the SFC instance are thereby also deleted.

Procedure

1. Select the SFC instance in the CFC.
2. Press the key.

Adapting the properties of charts, types and instances

9.1 How to adapt chart properties

Introduction

You can view and change the properties of the current SFC (SFC Editor) or any SFC in the chart folder (in the SIMATIC Manager). The chart properties are displayed in a dialog box with several tabs.

Opening the dialog box of the chart properties

In the SFC Editor

1. Open the SFC.
2. Select the menu command **SFC > Properties....**
The dialog box of the properties opens.

In the SIMATIC Manager

1. Select the SFC in the SIMATIC Manager .
2. Select the menu command **Edit > Object Properties....**
The dialog box of the properties opens.

Tabs

You can modify the chart properties in the following tabs.

- **General information**
Use this tab to enter or modify the chart name, the author and the comment and to enable or disable the write protection. When write protection is enabled, the chart cannot be unintentionally changed with the SFC Editor or interface editor.
- **Operating parameters AS**
You can set the following options in this tab.
 - Defaults for the initial state of the chart: "Step control mode", "Operating mode", "Command output", "Cyclic operation" and "Time monitoring".
 - SFC startup following a CPU restart
The options are "Initialize SFC" or "Retain SFC state." This allows you to determine if the SFC should start with the data it had before the CPU stop or if it should be initialized again following a restart of the CPU.
 - Start options of the chart: "Autostart" and "Use default operating parameters when SFC starts".

9.1 How to adapt chart properties

- **OS**

If the "Transfer chart to OS for visualization" check box is selected, the SFC is transferred to the OS automatically with the next OS compilation.

- **Version**

The major and minor versions can be set separately with the cursor keys. Version change is disabled for write-protected charts. You can set versions from 0.0001 to 255.4095. A new chart always starts with version number 0.0001. No number less than the previously saved version can be set.

Use "Data version" to obtain information on the software version used to create the chart.

When versioning is enabled in the properties of the project, the dialog box is opened automatically with the "Version" tab after a chart has been changed and the chart has been closed.

Data version:

This tab displays the software version used to create or most recently modify the program. The data version does not depend on the product version. The data version is determined by the database and shows the current version of your data structure.

9.2 How to adapt type properties

Opening the dialog box

In the SFC Editor, select the menu command **SFC > Properties...** or the shortcut menu command **Object Properties...** in the chart folder of the SIMATIC Manager for the selected SFC type.

The dialog box of the properties opens.

You can change the type properties in the following tabs.

"Common" tab

Use this tab to enter or modify the type name, the author, the family, the FB number, the comment and to enable or disable the write protection. When write protection is enabled, the chart cannot be unintentionally changed with the SFC Editor or the characteristics and interface editors.

"Operating parameters AS" tab

You can perform the following actions in this tab:

- Set the defaults for the initial state of the SFC instances that are generated from this type: "Step control mode", "Operating mode", "Command output", "Cyclic operation" and "Time monitoring".
- Enable or disable the start options for the SFC instances: "Autostart" and "Use default operating parameters when SFC starts".
- SFC startup after CPU restart
This includes the "Initialize SFC" and "Retain SFC state" options. This allows you to determine if the SFC should start with the data it had before the CPU stop or if it should be initialized again following a restart of the CPU.

"Options" tab

You can perform the following actions in this tab:

- SIMATIC BATCH area
 - Classification of the SFC type for SIMATIC BATCH in the "Category" box:

<none>	There is no classification for BATCH.
EOP	The SFC type is classified as "EOP".
EPH	The SFC type is classified as "EPH".
EPH (derived)	The SFC type is derived from an SFC type with the BATCH category "EPH interface". If the "EPH (derived)" option is selected, the name of the interface must be entered in the "Derived from interface" field. Only then is the derivative from a base class provided.
EPH (interface)	An SFC type declared in the BATCH category as an "EPH interface" serves as a base class and thus as a generic interface for the integration of the recipe phase (RPH) of SIMATIC BATCH. An SFC type with the BATCH category "EPH interface" only contains an interface description with setpoints, control strategies, etc. but no implementation. The programming of sequencers with the steps and transitions is no longer offered by the SFC editor in this case.

- Allow operator instructions to enable input of values in the operator dialog on the OS.
- "Derived from interface" selection box
You can select the interface derived from the SFC type in this selection box.
When the "EPH (derived)" option is selected in the "Category" box, the name of the interface must be entered in this box. Only then is the derivative from a base class provided.
- Control strategy selection
This box contains all control strategies configured for the SFC type (maximum of 32).
Select the respective check box to enable control strategies of the SFC type as default for the SFC instances. This applies to all SFC instances that are already in the project - as long as the default value has not been changed in the instance - and for all the SFC instances yet to be generated. The control strategy selection can be modified for the individual instances.
- SIMATIC IT
Use the "MES-relevant" check box to decide whether or not the information of the SFC instances is transferred to the enterprise levels MIS/MES when requested.

"Version" tab

The major and minor versions can be set separately with the cursor keys. Version change is disabled for write-protected SFC types. You can set versions from 0.0001 to 255.4095. A new SFC type always starts with version number 0.0001. No number lower than the previously saved version can be set.

Use "Data version" to obtain information on the software version used to create the chart.

When versioning is enabled in the properties of the project, the dialog box is opened automatically with the "Version" tab after a type has been changed and the window has been closed.

Data version:

This tab displays the software version used to create or most recently modify the program. The data version does not depend on the product version. The data version is determined by the database and shows the current version of your data structure.

See also

How to adapt instance properties (Page 95)

9.3 How to update SFC types

Updating in multiproject

SFC types can be updated in a multiproject. The procedure is the same as when updating block types of CFC.

You can find additional information in the section "How to update block types/SFC types" of the *CFC for SIMATIC S7* documentation.

9.4 How to adapt instance properties

Opening the dialog box

1. Select the SFC instance in the CFC and then select the menu command **Edit > Open**.
The SFC instance opens in the SFC editor.
2. Select the menu command **SFC > Properties...** for the opened SFC instance in the SFC Editor.
The dialog box of the properties opens.

Tabs

You can modify the SFC instance properties in the following tabs.

- **General information**
This tab is used to enter or modify the instance name and the comment. All other properties (see SFC type) can be read but not modified. The "Write-protected" check box indicates if the CFC is write-protected. In this case, you have read-only rights for the SFC instance.
- **Operating parameters AS**
In this tab, you can change the operating parameters for the SFC instance (see SFC type).

- **Options**

In this tab, you can view and change the options set for the SFC type:

- SIMATIC BATCH "Category" (view only)
The field displays the classification configured for the type:
You can find a description of the classification in the section "How to adapt type properties (Page 91)".
 - "none"
 - "EOP"
 - "EPH"
 - "EPH (derived)"
 - "EPH (interface)"
- SIMATIC BATCH "Allow operator instructions" (view only)
Values can be input in the operator dialog box on the OS when this option is set.
- "Derived from interface" (read only) SIMATIC BATCH selection box
If configured, the interface from which the SFC type is derived is displayed in this selection box.
- Control strategy (cannot be changed if write protection is enabled)
This box contains all control strategies configured for the SFC type (maximum of 32).
The control strategies adopted from the SFC type are selected. You can change the selection for each SFC instance. You can select a new control strategy or deselect an existing assignment.
Note: If you change the control strategy selection for an SFC instance, all subsequent changes to the control strategy selection for the SFC type are no longer applied automatically for this SFC instance.
- SIMATIC IT
The "MES-relevant" check box determines whether or not the information of this SFC instance is transferred to the MIS/MES enterprise levels when requested.

- **Version**

In "Version:" box, you can always see the version of the associated SFC type. The version cannot be changed in the SFC instance.

Data version:

This tab displays the software version used to create or most recently modify the program. The data version does not depend on the product version. The data version is determined by the database and shows the current version of your data structure.

Configuring messages

10.1 How to configure messages in the SFC

Configuring messages in the SFC

Select the menu command **SFC > Message...** in the SFC Editor to configure the messages.

You can configure specific message texts for each SFC/SFC type/SFC instance. You can change the message texts in a dialog box (for example, to distinguish messages in different charts/types).

Configuration limits

You can configure the following number of messages for one **SFC**:

- Seven messages requiring acknowledgment

You can configure the following number of messages for one **SFC type** or one **SFC instance**:

- Seven messages requiring acknowledgment
- Five message that do not require acknowledgment

The SFC type itself requires the remaining available messages (one message for each message type and 10 status messages for SIMATIC BATCH).

The reserved message events have default texts:

- "Step runtime exceeded"
- "Operator prompt"
- 10 status messages for SIMATIC BATCH (only with SFC type/instance)
 - "Run"
 - "Completed"
 - "Held"
 - "Aborted"
 - "Ready to complete"
 - "Stopped"
 - "Error"
 - "MANUAL"
 - "Not released for SIMATIC BATCH"
 - "Runtime exceeded"

Step runtime exceeded

The "step error" event is a process control message that must be acknowledged and is signaled along with three associated values using the message block ALARM_8P. The remaining seven available messages (SFC type/instance only) and associated values can be assigned as required.

There are also inputs/outputs at the standard interfaces that can be used to trigger the messages (SIG_2 ... SIG_8), including the associated values AUX_PR04 ... AUX_PR10 (for SFC type/instance).

You can use these inputs/outputs with interconnections in the actions of the steps or with direct block interconnections.

Operator prompt

The operator prompt is a message that does not require acknowledgment and that is signaled in an SFC with the NOTIFY message block and in an SFC type with NOTIFY_8P message block.

You can find a table of the I/Os used for the messages in the interface under: Messages (SFC inputs/outputs) (Page 209)

Note

When configuring messages, remember that the messages of ALARM_8P can only be assigned message classes requiring acknowledgment.

Configuring the characteristics

11.1 Using the characteristics editor and the interface editor

Characteristics editor/interface editor

You use the characteristics editor for the technological configuration of the interface for an SFC type. You can only work with the editor if an SFC type is open in SFC.

You define the interface inputs/outputs of the SFC type as follows:

- Directly in the interface editor:
Call this editor with the menu command **View > Inputs/Outputs** or with the button



- In the characteristics editor
Call with the menu command **View > Characteristics** or the button



The interface editor and the characteristics editor box are displayed alternatively. This means that it is not possible to define technological characteristics and interface inputs/outputs at the same time. You can define characteristics and inputs/outputs in any order.

Note

You define individual inputs/outputs in the interface editor. These are available in addition to the inputs/outputs defined in the characteristics editor. These inputs/outputs are unknown to the characteristics editor and cannot be handled there.

Setpoints, for example, are created and assigned values during the definition of characteristics. From this definition, the characteristics editor generates the required inputs/outputs, which are displayed in the interface editor after it is opened and can be used to configure the sequential logic. You cannot change the I/Os generated by the characteristics editor in the interface editor (except system attributes, initial value, comment).

You can group inputs/outputs only in the characteristics editor. These groups are taken into account during compilation in the applicable plausibility checks and status and message processing. The values (such as setpoints) required for visualization/operator control on an OS station that are displayed with the SFC type faceplate are configured in the characteristics editor.

You can find additional information on the characteristics editor in the section: Characteristics of the SFC type (Page 100)

11.2 Characteristics of the SFC type

Overview

A characteristic is the technology-oriented functionality of an SFC type that is described by a set of attributes (for example, I/O name, data type, initial value or high limit).

The following characteristics are available for creating the type:

Characteristic	Meaning
Control strategies	Control strategies are used for structuring of an SFC type as required for process control. You define control strategies in the characteristics editor in order to use them in configuring sequencers (for heating or cooling, for example). The control strategy can be set by the operator or by a higher-level controller (for example, SIMATIC BATCH). Control strategies are optional. You can find additional information in the section: Note on subsequent modifications to control strategies (Page 102)
Setpoints	Setpoints control the SFC type or are used by it to control underlying automation functions. Setpoints can be set by the operator or by a higher-level controller (SIMATIC BATCH, for example). A setpoint is assigned to the control strategies that exist when it is created. You can change the assignment individually for each control strategy.
Process values	Process values control the SFC type based on process signals (for example, the value for the fill level).
Control values	Control values are used by the SFC type to control external logic.
Parameters	Parameters modify the behavior of the SFC type. For example, you can use parameters to select alternative branches or to configure start conditions.
Bit memory	Bit memory is the local data memory of the SFC instances belonging to the SFC type. You can temporarily store data here and reuse it later. The data are retained until they are overwritten or deleted.
Timers	Timer objects are used for configuring time-controlled sequences in an SFC type. Timer objects can be used locally for a step or used across multiple steps and sequencers. Timed routines are started in steps, stopped, and the elapsed time queried in transitions or start conditions. You can vary timed sequences with a variety of modes (for example, pulse, extended pulse, on delay, retentive on delay, off delay).
Note texts	Note texts are used to display information relating to execution that may require action on the part of the operator. A note text is displayed on the OS in the SFC faceplate and can be acknowledged there. You can find additional information about this in the section: How to configure note texts and position texts (Page 103)
Block contacts	Block contacts represent blocks of the basic automation. When a block contact is created, inputs/outputs of the block to be represented are created in the interface of the SFC type. When configuring the SFC type, you can use these inputs/outputs to represent the block that will later be interconnected with an SFC instance. You can find additional information in the sections: Inputs/outputs for characteristics (Page 104) Block contacts (Page 105).
Position texts	Position texts are used to display the current progress in execution or the status of execution of the SFC. A position text is displayed on the OS in the SFC faceplate. You can find additional information in the section: How to configure note texts and position texts (Page 103)

Additional information

You can find additional information in the following sections:

Inputs/outputs for characteristics (Page 104)

Attributes for characteristics (Page 107)

11.3 Note on subsequent modifications to control strategies

Notes

Control strategies already used, e.g. in:

- Start conditions
- Transitions
- Assignment to setpoints
- Control strategy release

Should not be deleted or moved in the characteristics editor, otherwise the control strategies will be renumbered. This also applied for control strategies that you copy from one SFC type into a different SFC type.

The numbers of the control strategies in the existing utilization are not adapted. For this reason, accesses and other mechanisms that reference these numbers no longer function as originally intended.

Example

You no longer want to use the control strategy with number "2" and delete it.

The control strategies are renumbered. The previous control strategy "3" is given the number "2", control strategy "4" is given the number "3", etc.

If you no longer want to use a control strategy, do **not** delete the control strategy; rather, rename it as "not used", for example. The numbering and functionality of the other control strategies is then retained.

11.4 How to configure note texts and position texts

Procedure from configuration up to using the note texts

1. The note texts are configured in the characteristics editor and each text is assigned a unique number. When the text is created, the editor first assigns the next available number but this number can be changed to any other number (as long as it is unique among all the note texts) (value range: 1 to 32767). The numbering can also include gaps.
Before a note text can be displayed in the SFC faceplate, the OPTIPNO I/O containing the relevant note text number must be assigned in a step suitable for the note text.
2. The OS can be activated after compiling and downloading the AS and compiling and downloading the OS.
3. You can prepare and start the SFC instance ("Prepared values" view of the faceplate) once the faceplate for the SFC instance is open.
When the step used in item 1 is executed, the faceplate displays the note text.
4. The note text can be acknowledged with the "O" button, whereby the dialog box for operator confirmation is displayed.
5. When the operator action is confirmed (by clicking the "Yes" button), an operating message is generated on the OS and the OPTIPNO output is reset to "0" in the block.
The value change of OPTIPNO from "x" to "0" can be interpreted as an acknowledgment if the SFC type does not trigger this change itself by assigning a value to OPTIPNO in a step.

Note: The displayed text is either the comment for the note text or, if the comment was not configured, the name.

Procedure from configuration to using the position texts

The procedure for position texts corresponds to the procedure for note texts with the following differences:

- The associated block I/O is called "POSINO".
- Steps 4 and 5 are omitted.

Notes

Note the following:

- The display of note and position texts does not cause a message to be generated and is not documented in the message system. The acknowledgment of a note text, on the other hand, is documented as an operating message.
- When copying instruction texts and position texts from one SFC type and inserting it in a different SFC type, the numbers are reassigned. If usages already exist in steps, transitions and sequencers then you must check these numbers and correct them if necessary.

11.5 Inputs/outputs for characteristics

Inputs/outputs for characteristics

There are predefined descriptions of the required inputs/outputs for use in defining the interface inputs/outputs or I/O elements of a characteristic.

The required interface inputs/outputs or terminal elements for the "Control strategies", "Note texts" and "Position texts" characteristics are already contained in the standard interface of an SFC type.

For the other characteristics, the names of the individual I/O elements are made up of the I/O names configured in the characteristic and a fixed part.

Additional information

Additional information is available in the sections:

"Characteristics" interface parameter assignments (Page 181)

"Characteristics" interface expansions (Page 183)

Standard interface of the SFC type (Page 174)

Block contacts (Page 105)

Attributes for characteristics (Page 107)

11.6 Block contacts

Block contacts

If you want to use block contacts to connect the basic automation blocks, specify the inputs/outputs relevant for the connection to an SFC type in the block types.

To do this, the "S7_contact=true" system attribute must be set for the block I/O in the creation language of the block type. The technological blocks from the PCS 7 Advanced Process Library (APL) are prepared accordingly. You can adapt the inputs/outputs in the supplied block types to your project conditions.

Connections between SFC types by means of block contacts are not supported.

Note

If you subsequently add or change system attribute "S7_contact" for the I/O of a block type that is used as a block contact in the SFC type, this has no effect on the block inputs/outputs of the SFC type.

If you want these changes to be adopted for the SFC type, note the following:

- Import the changed block type in the CFC with the menu command **Options > Block Types**. All block instances of the type are updated during the import.
 - Update the block contacts in the SFC with the menu command **Options > Block Contacts**. This menu command applies the changes to the interface of the SFC type and to all SFC instances.
-

By defining an instance of the "block contact" characteristic in the characteristics editor, the selected block inputs/outputs are created accordingly in the SFC type. The IN I/Os of the block type are created as OUT I/Os in the SFC type. The OUT I/Os of the block type are created as IN I/Os in the SFC type. The IN_OUT I/Os of the block type are created as OUT I/Os in the SFC type.

This allows you to interconnect a specific CFC instance of the block type to the corresponding inputs/outputs of an SFC instance. An instance of the "block contact" characteristic is therefore always assigned to exactly one block type. You must enter the relevant block type in the "Block" column in the right window pane. The block type must be present in the ES data management.

This procedure enables you to configure, commission, and visualize basic automation blocks independently. As a result, you can configure SFC types based on block contacts and interconnect them later to the basic automation blocks.

Affect of the "block contacts" characteristic on interconnections of an SFC instance

The "block contacts" characteristic of an SFC type affects the creation of an interconnection at the instance of the SFC type if the following condition is fulfilled:

- The interconnection should be created at a block that is configured in the "Block contacts" characteristic of the SFC type, for example, "PIDConL".

The "block contacts" characteristic creates block I/Os of a particular block (for example, PIDConL) in the SFC type.

11.6 Block contacts

When one of these block I/Os of the SFC instance is interconnected with the block in the chart ("PIDConL" in the example), all block I/Os of the SFC instance that were created using the "Block contacts" characteristic are interconnected with the associated I/Os at the block ("PIDConL" here). This means that not only will this interconnection be created, but all other block contact interconnections between the SFC instance and the block will be automatically completed as well.

11.7 Attributes for characteristics

Attributes for characteristics

In the following table you can see the attributes available for configuring the characteristics along with their meaning.

Assignment of the attributes to characteristics

Characteristics →	Control strat- egy	Set points	Process values	Control values	Para- meters	Bit mem- ory	Timers	Note texts	Block contacts	Posit- ion texts
Attributes ↓										
Name	x	x	x	x	x	x	x	x	x	x
Display name	x	x			x				x	
Number	x							x		x
Standard	x									
Data type		x	x	x	x	x				
I/O name		x	x	x	x	x	x		x	
Comment	x	x	x	x	x	x	x	x	x	x
<Setpoint>	x									
Initial value		x	x	x	x	x	x			
Text length		x	x	x	x	x				
Precision		x	x	x	x	x				
Unit		x	x	x	x					
Low limit		x								
High limit		x								
Text0		x								
Text1		x								
Enumeration		x								
Archive		x								
Test		x								
Setpoint ID		x								
Proc val ID		x								
Material		x								
Tracking ID		x								
Block									x	
Faceplate					x					
Transfer ID		x								
Transfer coun- ter		x								

Meaning of the attributes

Attribute	Meaning
Name	The name is a designator for the characteristic that must be unique among all characteristics of a type. It can contain a maximum of 24 characters and must not contain spaces or special characters (exception: "_"). The name is irrelevant for the interface I/Os.
Display name	By default, the display name is the same as the name and is transferred to WinCC when you compile the OS. The display names can only be modified on the type. The display name cannot be translated until after the display name has been changed and is then included in the transfer to WinCC for the respective language when the OS is compiled.
Number	<p>The number uniquely identifies the corresponding characteristic.</p> <p>With control strategies, the number is managed by the system and assigned consecutively from 1 to 32 according to the position in the characteristics editor. The control strategy <no> is assigned bit <no-1> in the SELCS I/O of the SFC type. This bit is set if the corresponding control strategy has been configured. With an SFC type and SFC instance, the bit is deleted when the corresponding control strategy is not to be used (selection with the menu command SFC > Properties > Options: Control strategy selection).</p> <p>The next available number is used for note and position texts. However, you can replace the number with any other number of your choice as long as it is unique and in the range of 1 to 32767.</p>
Standard	"Standard" indicates the control strategy that will be used when starting the SFC instance with prepared values.
Data type	<p>Permitted data types for characteristics are BOOL, INT, DINT, REAL, and STRING. For setpoints, the data types PI and PO are also available and basically represent a REAL setpoint with the addition of the "Material" and "Tracking ID" attributes. Enumerations can be assigned to the data types DEST, SOURCE, VIA and TKEY.</p> <p>Depending on the characteristic and its data type, other boxes can either be edited or are locked.</p>
I/O name	<p>The I/O name is required to generate the interface I/Os belonging to the characteristic. The I/O name is derived from the characters of the name. You can change the I/O name. The following limits apply:</p> <p>Setpoints and counters: max. 16 characters Block contacts: max. 10 characters All other characteristics: max. 24 characters</p> <p>When the interface I/Os are created, several I/Os are generated depending on the characteristic. A suffix is appended to the I/O name so that the I/Os are named uniquely.</p> <p>The maximum lengths of the I/O names result from the different ways of generating interface I/Os. With setpoints and timers, the system limits the suffix to a maximum of 8 characters. With block contacts, the length of the suffix depends on the block being used. No suffix is appended to the other characteristics so that the I/O name can contain a maximum of 24 characters.</p> <p>Name length:</p> <p>The names of the roles are also used for generating the I/O names. An I/O name is made up of the role name and the name of the I/O of the lower-level object, separated by "_". The length of a name for an I/O to the SFC is limited to 24 characters.</p> <p>For the name of the role of an EMT on an EPH, a maximum of 7 characters is allowed, because the I/O with the longest name and "s7_contact" attribute on the EMT is "ERROR_COMPLETING". This already has 16 letters. This leaves 7 letters (24 – 16 – 1=7) for the role name.</p> <p>For the same reason, the I/O name of a setpoint of the lower-level EM must not be longer than 24 – 7 {MaxRoleName} – 4 {"_ERR"} - 1 = 12 characters.</p> <p>For the roles of the CMT on the EMT, this cannot be said generally because it is unknown which I/O names are used on the technological block of the CMT. In general (e.g. APL library), these are much shorter so this restriction comes into play less often here.</p>
Comment	You can use comments to describe the characteristic in greater detail. The comment can contain a maximum of 80 characters and include any special characters.

Attribute	Meaning
<Setpoint>	<p>Setpoint selection for the "Control strategy" characteristic. Name of each created setpoint is displayed as a column. This allows you to assign the required setpoint to the control strategy by selecting the corresponding column.</p> <p>The assignment of setpoint to control strategy is stored in the "<setpoint-I/O name>_CS" I/O for each setpoint. The coding is the same as for the SELCS I/O; in other words, control strategy <no> is assigned bit <no-1>. This bit is set if the setpoint is assigned to the control strategy; in other words, it is required to run the control strategy.</p>
Initial value	The initial value corresponds to the value of the characteristic when no current value is available. The attribute can be changed in the SFC instance.
Text length	With the STRING data type, the text length defines the maximum length of the string (number of characters: 1 to 254).
Precision	With the REAL, PI, and PO data types, the precision defines the number of decimal places to be displayed (0 to 7).
Unit	<p>With the INT, DINT, REAL, PI, and PO data types, a unit can be defined. This is stored as system attribute "S7_unit" for the interface I/Os. The attribute can be changed in the SFC instance.</p> <p>The units are included as a basic set in the ES data management and can be added to or modified in SIMATIC Manager as "shared declarations".</p>
Low limit	With the INT, DINT, REAL, PI, PO, DEST, SOURCE, VIA, and TKEY data types, a low limit can be defined for the range of values. This is stored in the "<I/O name>_LL" I/O. The attribute can be changed in the SFC instance.
High limit	With the INT, DINT, REAL, PI, PO, DEST, SOURCE, VIA, and TKEY data types, a high limit can be defined for the range of values. This is stored in the "<I/O name>_HL" I/O. The attribute can be changed in the SFC instance.
Text0	With the BOOL data type, this is used to specify the value identifier for the value FALSE. This attribute is stored as system attribute "S7_string_0" for the interface I/Os. The text box is only available when the data type is BOOL and when <none> is set in the "Enumeration" column. The attribute can be changed in the SFC instance.
Text1	With the BOOL data type, this is used to specify the value identifier for the value TRUE. This attribute is stored as system attribute "S7_string_1" for the interface I/Os. The text box is only available when the data type is BOOL and when <none> is set in the "Enumeration" column. The attribute can be changed in the SFC instance.
Enumeration	<p>With the BOOL, INT, DINT, DEST, SOURCE and VIA data types, you can assign an enumeration. This is stored as system attribute "S7_enum" for the interface I/Os. The enumeration is defined in the SIMATIC Manager in the "Shared declarations". You can select the name of the enumeration for the attribute from a drop-down list. The attribute can be changed in the SFC instance.</p> <p>If a listing is assigned to a block parameter of an SFC instance, you can select any number of additional listings from the shared declarations later.</p> <p>Note: If "S7_enum" is used, the "S7_string_0" and "S7_string_1" system attributes are ignored.</p>

Attribute	Meaning
Archive	<p>From the drop-down list, you can select whether the current value of the "<I/O name>_AO" output is to be archived in WinCC, is not to be archived, or is to be transferred to the long-term archive. The attribute can be changed in the SFC instance.</p> <p>An archive variable is created when the OS is compiled so that the value is transferred to the archive.</p> <p>Note:</p> <p>In the case of an EMT, the instance is implicitly created with the SFC type at once. After defining a setpoint within the SFC type, this setpoint is propagated to the SFC instance but the archiving attribute (internal UDA S7_archive) at the instance is initially set to "No Archiving". Changing the archiving attribute afterwards at the SFC type has generally no effect on the instance.</p> <p>If the system attribute S7_archive is changed on EMT in the project library and a type synchronization is performed, all instances of the EMT will inherit this attribute. If the attribute was changed in the instances, the attribute will be overwritten with the synchronization.</p> <p>If the system attribute S7_archive is changed on standalone SFC types in the project library and a block type update is performed, the existing SFC instances remain unchanged regarding this attribute. New instances of the SFC will inherit this attribute.</p> <p>In general, the archiving behavior is determined at the SFC instances. Therefore the archiving setting must be configured in the SFC instance and not in the SFC type.</p>
Test	This selects the characteristic that is to be provided for manual testing (manufacturing instruction) in the batch report of SIMATIC BATCH.
Setpoint ID Proc val ID	You use the ID number to specify if the characteristic value is to be saved for logging. If the ID is greater than 0, the setpoint or process value is also made available for an external program. You can make processing in an external application easier if you assign unique ID numbers to the characteristic values. You can assign ID numbers from 0 to 32767.
Material	With the PI and PO data types, you can enter a material identifier.
Tracking ID	With the data types PI, PO, you can enter a numeric ID to identify the material.
Block	This attribute contains the name of the block type represented by the block contact. When the interface I/Os are generated, the interface I/Os for which system attribute "S7_contact=true" is set are adopted in the SFC type from the block type. The names of the interface I/Os of the block type are appended as suffixes to the I/O names of the block contact. Inputs and in/outs of the block type become outputs of the SFC type. Outputs of the block type become inputs of the SFC type.
Faceplate	Here you can select the parameters that will be displayed in the parameter view of the SFC faceplate.
Transfer ID	This ID is the unique ID of an individual transfer in SIMATIC BATCH. It is part of the transfer key (data type, TKEY) and assigned by an MES application.
Transfer counter	This attribute shows the loop repetition number of the transfer phases in SIMATIC BATCH. It is part of the transfer key (data type TKEY) and is assigned by an MES application.

Additional information

You can find additional information about the I/Os and characteristics in the following sections:

"Characteristics" interface parameter assignments (Page 181)

"Characteristics" interface expansions (Page 183)

You can find additional information about the system attributes in the STEP 7 online help in the section "System attributes for block I/Os".

Configuring sequential control systems

12.1 How to configure sequential control systems

Requirements

Before you can configure sequential control systems, you must first create the required basic automation functions with CFC and/or STEP 7 tools. At the same time, the AS blocks to be used in the SFCs or SFC types are also inserted. Automation functions that do not yet exist can be added as needed and then used in the SFC.

Procedure

When configuring a sequential control system (SFC or SFC type), proceed as follows:

- Create the sequencer topology with the required number of sequencers and the required layout of SFC elements.
- Configure the start condition, the preprocessing and the postprocessing in the "Properties" dialog box for the sequencers.
- Configure the actions and conditions in the "Properties" dialog box for the steps and transitions.

Color ID

The objects of the sequential control system are displayed in different colors depending on their current state. The elements of an unselected sequential control system, for example, are displayed in "white" (parameters not assigned) or "gray" (parameters assigned) and in "blue" when they are selected.

You can select your own color scheme for specific elements using the menu command **Options > Customize > Colors....**

When you open the object properties, the tab IDs of the tabs without entries have the standard background color of the dialog box. For a tab with content, the background of the tab ID is marked in the color that is set in the Windows properties for the display (right mouse button on desktop: **Properties > Display > Extended > Element: Selected Element**).

Textual interconnections

If you delete CFC blocks that access the SFC, these accesses will be configured as textual interconnections (text of the address is marked in yellow).

You can complete a textual interconnection as soon as the connection partner is available again in the chart folder. The textual interconnection will then become a real interconnection.

To complete a textual interconnection, select either the menu command **Options > Make Textual Interconnection** or select the "Make textual interconnection" check box in the "Compile" dialog box.

12.2 Creating the sequencer topology

12.2.1 How to create a sequencer topology

Sequencer display

In its initial state, the new SFC consists of one sequencer; however, this can be extended up to a maximum of 8 (SFC) or 32 (SFC type) sequencers. You create each sequencer in its own working window. A tab is created for every working window at the bottom of the screen. The tab shows the name of the sequencer (for example, RUN or SEQ1). You can switch to the individual sequencers using these tabs.

A newly created sequencer (menu command **Insert > Sequencer > ...**) is inserted at a selected position in the SFC in its initial state consisting of an initial step, transition and final step.

You can find additional information in the section: How to configure multiple sequencers (Page 113)

If you insert or delete SFC elements in the sequencer, its layout is changed automatically according to predefined rules. These rules determine the spacing between elements, the extension of steps and transitions, or the alignment of alternative branches. You can change these layout rules at any time using the menu command **Options > Settings > Layout....**

You can center the entire sequencer topology in the window. You can zoom into and zoom out of the display as required using the zoom functions. The changes are made in percentage increments that are determined by the zoom factor.

Adding elements

If you want to add additional elements to the SFC, select the icon of the required element to be created in the element bar.

The cursor changes its appearance from an arrow to the selected icon with a positioning crosshair. Position the cross at the desired location to insert the sequencer element. The position of insertion is marked by a green line. Left click on the insertion position. The inserted elements are highlighted in color.

Syntax rules

The sequencer topology is formed by the sequences of steps and transitions. The fundamental rule of the sequencer topology is that a step (S) must be followed by a transition (T) and a transition must always be followed by a step (sequence: **S-T-S** or **T-S-T**). The editor automatically adheres to the rules.

Example:

If you insert a simultaneous branch after a transition and before a step in a sequencer, a transition is automatically created between the simultaneous branch and the step.

12.2.2 How to configure multiple sequencers

Introduction

An SFC can contain several sequencers that can be used for different applications. By specifying different start conditions, you can arrange for a specific sequencer to start when a particular event occurs. You can, for example, configure a separate sequencer for every operating state (idle, active, or error, for example) or for every control strategy (heating, cooling, or tempering, for example).

Note

Predefined sequencers are already available for various standard scenarios. These sequencer templates are available in the "SFC Library".

You can copy these templates and adapt them to suit your purposes.

Grouping of sequences into a sequencer

If the processing sequences for the "Run" state and the "Holding", "Held" and "Resuming" states are to be grouped into one sequencer as alternative branches, you must set the RUNHOLD input of the SFC to FALSE. After the "Hold" command, the SFC sequencer is aborted (the currently active step and the END step are run through) and following each status change, the SFC sequencer is run through again. The SFC sequencer is started again following the "Resume" command.

If RUNHOLD = TRUE, the sequencer change mechanism no longer works because the SFC sequencer is held. The held SFC sequencer cannot be (re)started in the statuses "Holding", "Held" and "Resuming". The held SFC sequencer is resumed in the held step following the "Resume" command.

Start conditions of the sequencers

The first sequencer of an SFC has condition "<SFCName>.RUN=Run" for the SFC or "RUN=Run" for the SFC type. The start condition of any other newly added sequencer is empty and therefore unfulfilled; in other words, it will never be processed. In contrast to newly created transitions that are always fulfilled, a new sequencer must always be given a defined start condition with the menu command **Sequencer Properties > "Start Condition" tab**.

Because it is possible that several start conditions can be satisfied at the same time, you can assign different priorities to the individual sequencers with the menu command **Sequencer Properties > "General" tab**, Priority: 1 to 255).

Formulate the start condition of a sequencer so that the status of the operating state logic is checked and the appropriate sequencer is executed when the SFC is in the corresponding state. You can also formulate any other conditions to suit your purposes.

When **configuring start conditions**, follow the same procedure as for configuring transitions.

You will find detailed information on this in the sections:

Formulating the conditions of a transition (Page 146)

How to edit addresses in a transition (Page 146)

How to edit OS comments in a transition (Page 148)

Examples

Example 1:

Configure a sequencer with a start condition that queries one of the control strategies of the SFC. The formulation is, for example, "QCS=Heat". If the SFC is set to this control strategy, the sequencer is executed and is not dependent on the operating state of the SFC.

Example 2:

Configure a sequencer with the start condition "<SFCName>.IDLE=Idle". This sequencer is executed if the operating state is "Idle".

Example 3:

Any process status can also be queried for the start condition.

To do this, interconnect the process status to the external signal "LOCKERROR" (input of the SFC). The SFC changes to "Error" state if the problem occurs. Also configure a sequencer for handling the problem whose start condition is "<SFCName>.ERROR=Error AND <process status>=1", for example.

Example 4:

As an alternative to example 3, error handling is also possible without a state change. To achieve this, you configure a sequencer with the start condition "<process status>=1" and assign it a high priority. This sequencer is always executed when the problem occurs and the sequencer that is currently being processed has a lower priority than the sequencer for handling the problem. In this case, do not interconnect the process status to the "LOCKERROR" input as this would mean a change to "Error" state.

Note

With a chart, the address "<SFCName>.I/O" is specified and with a type only "I/O" is specified.

How do I create a sequencer?

You can insert a new sequencer consisting of a start step, a transition, and a final step with the menu commands:

Insert > Sequencer > Before Current Sequencer

or

Insert > Sequencer > At End

How do I move/copy a sequencer?

You can move or copy a sequencer within an SFC window and reinsert it at another position.

Procedure for moving within the current SFC window

1. Select the menu command **Edit > Move Sequencer...**
The "Move Sequencer" dialog box opens.
2. Check if the "Create copy" check box is cleared in the dialog box (default: check box cleared).

3. From the list, select the sequencer before which you want to insert the moved sequencer or select the line "(put at end)".
4. Click "OK".

Procedure for copying within the current SFC window

1. Select the menu command **Edit > Move Sequencer....**
The "Move Sequencer" dialog box opens.
2. Select the "Create Copy" check box (check mark).
3. From the list, select the sequencer before which you want to insert the copied sequencer or select the line "(put at end)".
4. Click "OK".

The copied sequencer is renamed automatically so that the names of the sequencers in the SFC window are always unique. If the last position of the name is a number, it is incremented, otherwise a free number is appended to the name.

Procedure for copying/moving and inserting outside the current SFC window

1. Select the menu command **Edit > Copy Sequencer** or **Edit > Cut Sequencer**.
2. Switch to the other SFC window.
3. Select the menu command **Edit > Insert Sequencer**.
The copied/cut sequencer is then renamed and inserted before the current sequencer.

Note

Pasting also works within the same SFC window. The copied/cut sequencer is then renamed and inserted before the current sequencer.

How do I delete a sequencer?

You delete the current sequencer as follows:

1. Select the menu command **Edit > Delete Sequencer**.
2. Click "Yes" to acknowledge the prompt for confirmation.

Notes on configuration

Note the following:

- The names of the steps and transitions must be unique within a sequencer; you can use the same names in different sequencers.
- You can also configure an additional action for each sequencer. Each action consists of the following:
 - **Preprocessing**: actions to be executed in every cycle after the sequencer has started before processing the steps and transitions
 - **Postprocessing**: actions to be executed in every cycle after processing the steps and transitions This, for example, allows you to make pre-settings or to pass on the results of the sequencer execution.

You configure both parts in the "Sequencer Properties" dialog box. The dialog box provides the "Preprocessing" and "Postprocessing" tabs for this purpose, whose structure corresponds to the processing phases of steps.

12.2.3 Overview of inserting/creating SFC elements

Inserting/creating SFC elements

You select the SFC element you want to insert with the menu commands **Insert > Step/Transition**, **> Simultaneous Branch**, **> Alternative Branch**, **> Loop**, **> Jump** or **> Text**. The cursor changes its appearance and you change to insert mode. As an alternative to the menu commands, you can also click the corresponding icon in the element bar.

When you now move the mouse over the sequencer, the shape of the cursor indicates whether you can insert the selected element (icon of the SFC element and <+> key) or whether you cannot (icon depicting a prohibited sign).

If you move the mouse pointer to a "permitted" insertion point in the sequencer, you will see a **horizontal green line**.

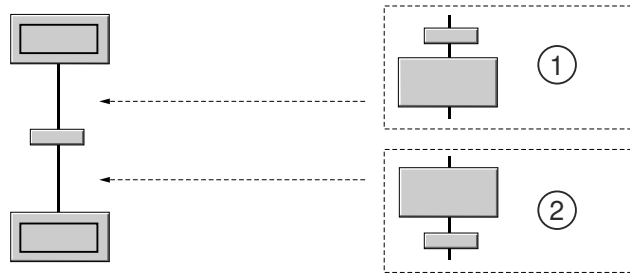
Within the simultaneous branch (in the vicinity of the upper simultaneous branch or lower simultaneous branch), the **vertical green line** indicates that you are inserting an additional sequence. If, for example, you insert an alternative branch beside a sequence, an additional step is created before and after the sequence to maintain the correct syntax. If, for example, you insert a simultaneous branch beside a sequence, an additional transition is created before and after the sequence.

After insertion, the inserted SFC elements are highlighted in blue.

12.2.4 How to create a sequence

Introduction

When generating a sequence, a transition-step-sequence (TS) or an step-transition-sequence (ST) (see figure) is generated depending on the position.



Key:	
(1)	TS sequence
(2)	ST sequence

Procedure

1. Click the step/transition button in the element bar



2. Click the vertical line of the sequencer between a step and transition (or between a transition and step).
3. If you move the cursor to a "permitted" insertion point in the sequencer, you will see a horizontal green line.
4. If you also want to define the length of the sequence, hold down the left mouse button and drag the lasso in the vertical direction.
The current length that will be inserted (number of ST/TS pairs) is displayed as a number at the starting point of the lasso.

Result

The sequence is inserted at the desired position.

12.2.5 How to create and edit a simultaneous branch

Introduction

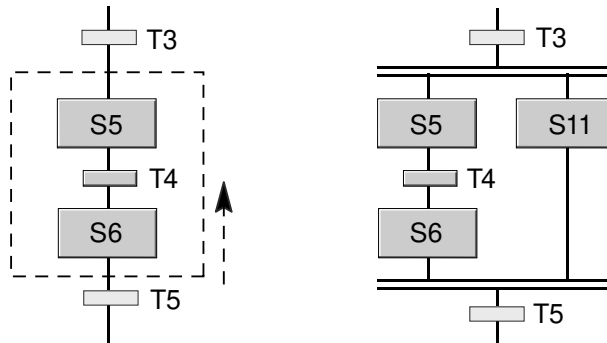
When you create a simultaneous branch, two sequences are generated, each consisting of one step. Depending on the point of insertion, a further transition is added automatically before or after the simultaneous branch to ensure that the syntax is maintained.

Procedure

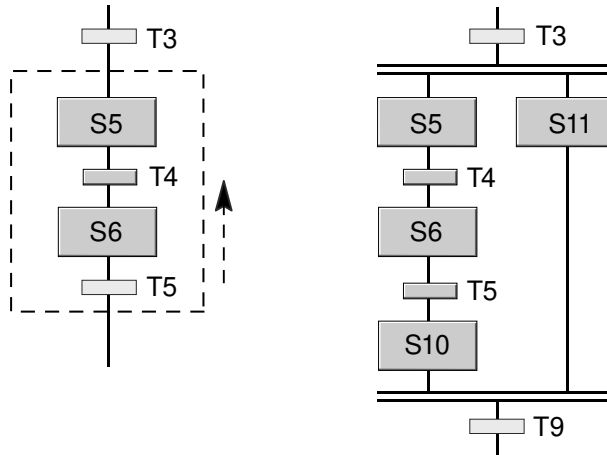
1. Click the simultaneous branch button in the element bar.



2. Then click the position at which the simultaneous branch should be inserted.
If you keep the left mouse key pressed while entering and drag open a lasso around elements of a sequence (here S5 to S6), the marked elements become part of the left sequence of the generated simultaneous branch.



If you drag a lasso so that instead of a pure step sequence (here: S5 to S6) a step-transition sequence is captured (here: S5 to T5), due to syntactic reasons an additional step is created (here: S10) in the left sequence and a transition (here: T9) after the simultaneous branch.



3. If you want to expand a simultaneous branch, switch to the desired insertion mode (for example, alternative branch) and click using the positioning cursor at the desired position in the vicinity of the upper or lower branch line.

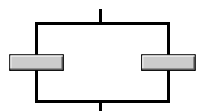
Editing a simultaneous branch

You can add additional sequences, simultaneous branches or alternative branches to a simultaneous branch, delete them or insert them in a different sequence. You can move sequences within a simultaneous branch or to any other position in the sequencer (except in the return branch of a loop). If you delete the next to last sequence, the remaining sequence is simply integrated in the surrounding structure and the simultaneous branch is eliminated.

12.2.6 How to create and edit an alternative branch

Introduction

When you create an alternative branch, two sequences are generated, each consisting of a transition. Depending on the point of insertion, a further step is added automatically before or after the alternative branch to ensure that the syntax is maintained.



Procedure

1. Click the alternative branch button in the element bar.



2. Then click the position at which the simultaneous branch should be inserted.
If you hold down the left mouse button and open a lasso around elements of a sequence, these elements become part of the left sequence of the resulting alternative branch.
3. If you want to expand an alternative branch, switch to the desired insertion mode (for example, simultaneous branch) and click the positioning cursor at the desired position in the vicinity of the upper or lower branch line.

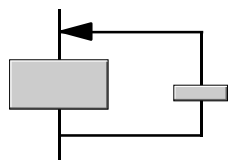
Editing an alternative branch

You can add additional sequences to an alternative branch or delete them and also insert them in a different sequence. You can move sequences within the alternative branch or to any other position in the chart. If you delete the next to last sequence, the remaining sequence is simply integrated in the surrounding structure and the alternative branch is eliminated.

12.2.7 How to create and edit a loop

Introduction

When you create a loop, a sequence, which can consist of a single step, and a return path with a transition are generated.



Procedure

1. Click the loop button in the element bar.



2. Click the vertical line at the position where the loop should begin.
3. Keep the mouse button pressed and release it on the vertical line at the desired end position.

Result

The loop is created around the elements contained between the start position and end position. The syntax is maintained by adding elements as necessary.

Moving a loop

If you want to move a loop, first select the loop with the elements it contains. Keep the mouse button pressed and place the loop at the desired position on the vertical line of the sequence.

Changing the return jump destination

To change the return jump destination later, select the horizontal line of the arrow tip and holding down the left mouse button, move it to any syntactically correct position in the sequencer above the starting point.

If you drag the arrow point below the starting point of the loop, the starting point will be moved instead of the return jump destination.

Note

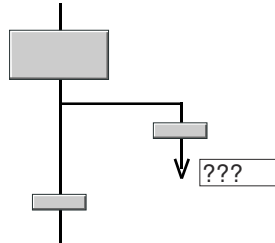
Loops leading into or out of alternative or simultaneous branches are not permitted. Due to the block-oriented sequencer topography, only entire chart elements, including alternative and simultaneous branches, can exist within a loop.

The return path of a loop can and must contain only one transition.

12.2.8 How to create and edit a jump

Introduction

When you insert a jump, a transition with an arrow and jump destination information is generated.



Procedure

1. Click the jump button in the element bar.



2. Create the jump directly after a step on the vertical line of the sequence:
 - If you want to create a jump with an undefined jump destination, click the vertical line after the step.
The jump is created. The jump destination is displayed as question marks (???). You can now change the jump destination.
 - If you want to create a jump with a defined jump destination, drag from the jump origin to the desired step.
The jump is created. The name of the step is entered in the jump destination.
 - If you want to create a jump with several jump destinations, repeat the previous action.

Note

When a step used as a jump destination is deleted, all jumps referring to it become undefined.

If the name of a step used as a jump destination is subsequently changed, all the jumps referencing it are automatically adapted.

3. Double-click the jump destination information.
The "Select Jump Destination" dialog box opens. All the available steps of the chart—except the start step—are shown in a sortable list.
4. Select a step as the new jump destination and click "OK".

Jump to destination

1. Select the jump.
2. Select the shortcut menu command "Go to Jump Destination".
Result: The jump destination is displayed as selected.

You can use the function in the edit and test modes.

Deleting a jump

Select the jump-transition and press the "Del" key.

Note

Be aware of the possible consequences during execution in the AS when working with jumps to or from a sequence of a simultaneous branch.

Jumps to or from a simultaneous branch should be used with special care: the respective jump always refers to the sequence in the simultaneous branch in which the source or destination is located. All other sequences will continue to be processed regardless of what happens.

If you jump from one of the sequences, you should make sure that

- There is a return jump to the same sequence

or

- All other sequences are also exited with a jump.

The user is responsible for using jumps carefully and being aware of the consequences to prevent unexpected runtime behavior in the AS.

Recommendation: Avoid using jumps into or out of simultaneous branches.

12.2.9 How to create and edit a text box

Introduction

You can insert, delete, copy and move a text box to any free position in the chart. You cannot move text objects across charts using drag-and-drop.

Procedure

1. Click on the text button in the element bar.



2. Click using the positioning cursor at the point in the window where you want to position the text box.

A text box is inserted in the window and opens. The text cursor is active.

3. Enter text in the text box.
The text automatically wraps at the end of the line.
4. Finally, click outside the text box to exit edit mode and to close the text box.

Note

When positioning, make sure that the text box does not cover elements of the sequencer topology. If this happens, the unselected text box will be displayed as a frame with a transparent surface (without content). The chart elements below it remain visible.

Changing the size of a text box

If you enter more text than can be displayed in the box, the size of the box is not increased automatically; instead the text moves out of the visible area. You can make the entire text visible by increasing the size of the box manually. If you change the width of the box, the length of the text lines is automatically adapted.

Procedure:

1. Click on the text box to open it.
2. Grab the hatched frame with the cursor and drag the box until the required size is reached.

Moving a text box

To move a text box to another location **within the window**, proceed as follows:

1. **Left click** on the text box.
The text box opens.
2. Grab the text box with the cursor at the non-highlighted point of the frame and drag it to another location within the window.

To move the text box **to another window**, follow the procedure described under "Copying or cutting and inserting a text box".

Copying or cutting and inserting a text box

1. Open the shortcut menu of the text box.
2. Select the menu command **Cut Text Box** if you want to move the text box or **Copy Text Box** if you want to copy it.
3. Switch to the required window.
This can be the same window or another window of a sequencer.
4. Open the shortcut menu and select the menu command **Insert**.
The cursor changes to a hand (with an empty object following cutting, with "+" object after copying).
5. **Left click** at the desired insertion position.
The text box is inserted at the cursor position.

Deleting a text box

1. **Right click** on the text box.
The shortcut menu opens.
2. Select the menu command **Delete Text Box**.
The dialog box with the prompt for confirmation opens.
3. Click "Yes".
The text box is deleted.

12.3 Managing SFC elements

12.3.1 Selecting SFC elements

12.3.1.1 How to select with a mouse click

Single selection

You can select in the following ways:

- You can select steps and transitions by simply left-clicking them.
- You select sequences by clicking the vertical line between a step and transition. Provided that the sequence is not in a simultaneous branch, alternative branch, or a loop, the entire sequencer is selected.
- You can select sequences by clicking on the upper or lower horizontal line.
- You can select loops by clicking the upper or lower horizontal line of the return path (the entire loop is selected) or on the vertical link (only the return path with the transition is selected).
- You can select jumps by clicking the horizontal line of the jump or the vertical line below the step of the origin of the jump (the origin and the jump are both selected). If you click the jump destination information, the associated transition is also selected.
- You can select the entire sequencer by clicking the vertical link of a sequence containing the start or final step.
- If you click an element, any other elements selected at the time are deselected.
- If you right click, the shortcut menu appears with the currently available menu commands.

Multiple selection

If you want to select more than one SFC element, hold down the <Ctrl> key when clicking with the mouse. The clicked SFC element is selected without deselecting elements you have already selected. If you have inadvertently selected an element, you can deselect it by clicking again while holding down the <Ctrl> key.

If you have selected an entire sequence and click a single element while holding down the <Ctrl> key, the single element is deselected and the entire selection effectively becomes a selection of single elements.

12.3.1.2 How to select with the keyboard

Selecting with the keyboard

You can select SFC elements with the <Up>, <Down>, <Left> and <Right> keys of the keyboard (single selection). The selection made depends on the currently selected element. If no SFC element is currently selected, when you first press a key, the last selected element is selected again.

- <Up> selects the previous SFC element in the sequence
- <Down> selects the next SFC element in the sequence
- <Left> selects the SFC element to the left in the sequence
- <Right> selects the SFC element to the right in the sequence

If you also hold down the <Ctrl> key, you select the SFC elements without deselecting those already selected (multiple selection).

12.3.1.3 How to select with the lasso

Selecting with the lasso

If you drag the mouse while holding down the left mouse button, you open a lasso or frame. All the SFC elements located completely within the lasso are selected when you release the button. SFC elements that were already selected are deselected if they were outside the lasso.

If you also hold down the <Ctrl> key, you select the SFC elements without deselecting those already selected outside the lasso.

If you drag a lasso around a group of selected and unselected elements while holding down the <Ctrl> key, the selected elements are deselected and the unselected elements are selected.

12.3.1.4 How to select steps and transitions for editing

Selecting for editing steps and transitions

You can edit the properties of steps and transition as follows:

1. Double-click a step or transition.
2. Select the menu command **Edit > Object Properties....**

A dialog box with tabs opens. Once this dialog box is open, you can edit additional steps or transitions as follows:

3. Click the following buttons in the dialog box:

	(previous step or transition)
	(next step or transition)
	(step or transition further to the left)
	(step or transition further to the right)

4. Click the desired step or transition in the SFC.

12.3.2 Copying, moving and deleting SFC elements

12.3.2.1 How to copy SFC elements

Overview of copying

You can insert copied elements of a sequencer to a syntactically correct position within the same sequencer or in another sequencer of the same SFC or a different SFC on the same CPU or a different CPU. If necessary, new names are assigned automatically to the copied elements. The copies contain the same actions or conditions as the originals.

If you paste a copied step or a copied transition in an SFC on a different CPU, textual interconnections may be created if there are accesses to objects that have not been copied (for example, blocks in the CFC).

When you copy, the syntax is automatically corrected, for example, by adding an empty step or an empty transition.

Copying with the mouse

In addition to the menu commands **Copy** and **Paste**, you can also copy SFC elements with the mouse (exception: text boxes).

Procedure:

1. Select the desired elements and hold down the left mouse button.
2. Press the <Ctrl> key at the same time.
3. Drag the positioning cursor to the desired location in the sequencer in the same window or another window and then release the mouse button (drag-and-drop).

The cursor adopts the shape of a prohibited sign if copying is not permitted at the current position.

Note on copying jumps

Note the following:

- If you copy a sequence that contains a jump and the step of the jump destination, the jump destination is adapted accordingly in the copy.
- If you copy a sequence that contains a jump, but the jump destination is outside of the copied objects, the jump destination becomes undefined.

Copying text boxes

Follow the steps outlined below to copy text boxes:

1. Open the shortcut menu of the text box.
2. Select the **Copy Text Box** menu command.
3. Switch to the required window.
This can be the same window or another window of a sequencer.
4. Open the shortcut menu and select the menu command **Insert**.
The cursor changes to a hand (with an empty object following cutting, with "+" object after copying).
5. Left click at the desired insertion position.
The text box is inserted at the cursor position.

Note

Multiple selected text boxes cannot be copied at the same time.

Copy object properties

Instead of SFC elements, you can also copy and insert the object properties of steps and transitions.

You can find additional information on this in the section:
How to copy object properties of SFC elements (Page 128)

12.3.2.2 How to copy object properties of SFC elements

Object properties of steps and transitions

You can copy the complete object properties of steps and transitions and insert them in other steps and transitions. This also applies to the start and end step.

Procedure

1. Select the SFC element whose object properties you want to copy.
2. In the shortcut menu, select the menu command **Copy object properties**.
3. Select the SFC element that is to adopt the copied object properties.

4. In the shortcut menu, select the menu command **Insert object properties**.
A message will appear that prompts you to confirm this action.
5. Acknowledge with "Yes".
The selected SFC element will adopt the copied object properties.

Note

The message appears prior to each inserting action. If you tick the box "Do not show this message in the future", the message will not be shown anymore.

If you open the "Customize" dialog box with the menu command **"Options > Customize..."** in the SIMATIC Manager, you can reactivate the system messages in the "General" tab.

12.3.2.3 How to move SFC elements

Overview of moving

You can paste cut elements of a sequencer to a syntactically correct position within the same sequencer or in another sequencer of the same SFC or a different SFC on the same CPU or a different CPU. If necessary, new names are assigned automatically to the copied elements. The moved elements retain their actions or conditions.

When you insert, the syntax is automatically corrected by adding an empty step or an empty transition as needed.

Moving with the mouse

In addition to the menu commands **Cut** and **Paste**, you can also move SFC elements with the mouse. If you move SFC elements, the elements are implicitly cut and pasted again.

Procedure

1. Select the desired elements and hold down the left mouse button.
2. Keeping the mouse button pressed, drag the positioning cursor to the desired location in the sequencer in the same window or another window and then release the mouse button (drag-and-drop).

The cursor adopts the shape of a prohibited sign if copying is not permitted at the current position.

Moving text boxes

To move a text box to another location **within the window**, proceed as follows:

1. Left click on the text box.
The text box opens.
2. Grab the text box with the cursor at the non-highlighted point of the frame and drag it to another location within the window.

If you want to move the text box **to another window**, proceed as follows:

1. Open the shortcut menu of the text box.
2. Select the **Cut Text Box**.
3. Switch to the required window.
This can be the same window or another window of a sequencer.
4. Open the shortcut menu and select the menu command **Insert**.
The cursor changes to a hand (with an empty object following cutting, with "+" object after copying).
5. Left click at the desired insertion position.
The text box is inserted at the cursor position.

Note

Multiple selected text boxes cannot be moved at the same time.

12.3.2.4 Deleting SFC elements

Deleting SFC elements

You can delete selected SFC elements after a prompt for confirmation. The following exceptions and special consideration apply:

- You cannot completely delete the start step or final step. If you delete the start or final step, the parameter assignments for these steps are deleted but not the SFC element itself.
- If you delete only one element from a syntactical unit, the syntax is immediately restored by entering a new element to suit the syntax (this has no parameters assigned). This means that you have only deleted the parameter settings for the object.
- You cannot delete the last step of a sequence in a simultaneous branch. To delete the sequence that now consists of only a single step, you must select the sequence. You select the sequence by clicking the vertical line.
The same applies to the alternative branch.
- You can delete a text box by right-clicking and selecting the **Delete Text Box** command from the shortcut menu. You can delete several text boxes by selecting them (with a lasso, for example) and then right clicking on one of the objects to select the **Delete** shortcut menu command.

12.4 Editing in the properties dialog box

12.4.1 How to edit sequencer properties

Opening the dialog box

1. Select the menu command **Edit > Sequencer Properties...** or position the cursor on the sequencer name in the tab at the bottom of the window and select the shortcut menu command **Sequencer Properties....**
2. The dialog box of the sequencer properties opens.
3. Specify the properties by formulating the actions and conditions.

The "Properties" dialog is divided into five tabs for editing the sequencer:

- "General"
- "Start condition (technological)" (for SFC type only)
- "Start condition"
- "OS Comment"
- "Preprocessing"
- "Postprocessing"

"General" tab

You change the name and comment of the sequencer in the "General" tab. The "Number" box contains the sequencer number generated by SFC. This cannot be changed. In The "Priority" box, you can assign the sequencer a priority of 1 to 255. Here, 1 represents the lowest priority and 255 the highest priority. The priority decides which sequencer of an SFC is started when the start conditions of several sequencers are met simultaneously.

"Start condition (technological)" tab

This tab is only available and can be edited for the SFC type. The tab is read-only for an SFC instance.

In this tab, you can define the technological condition for the SFC type that will cause the sequencer to start.

Only a technological condition can be defined in this tab.

The "Condition (technological)" button is used to create or edit the technological condition.

The 'Corresponding row in the "Condition" tab' selection box is used to assign the technological condition configured here one of the 16 conditions in the "Start condition" tab.

"Start Condition" tab

In this tab, you define the conditions for the SFC/SFC type that will cause the sequencer to start (for example, "<SFCName>.RUN = Run" starts the sequencer when the SFC is in "Run" operating state).

The other editing steps in this tab are identical to those for transition conditions. You can find additional information on this in the section "Formulating the conditions of a transition (Page 146)".

"OS Comment" tab

Editing steps in this tab are the same as for the OS comments for the transitions. You can find information on this in the section "How to edit OS comments in a transition (Page 148)".

"Preprocessing" and "Postprocessing" tabs

In these tabs, you can configure the following actions for the SFC/SFC type:

- Action that is to be executed in every cycle after the sequencer has started but before the steps and transitions are processed (preprocessing).
- Action that is to be executed in every cycle after the steps and transitions are processed (postprocessing).

The editing steps in this tab are identical to those for formulating actions of the steps. You can find additional information on this in the section "How to edit a step (Page 132)".

As in steps and transitions, it is also possible to configure calculations in these two tabs. You can find additional information about this in the section "Overview of SFC calculations (Page 151)".

12.4.2 Editing the steps

12.4.2.1 How to edit a step

Opening the "Properties" dialog of the step

You open the dialog box by double-clicking the step to be edited or by selecting the menu command **Edit > Object Properties...** for a selected step.

A dialog box opens where you can edit the properties and formulate the actions.

The dialog box is divided into four tabs:

- "General"
- "Actions (technological)" (for SFC type only)
- "Initialization"
- "Processing"
- "Termination"

"General" tab

On the "General" tab, you can enter or change the step name, the run time and the comments for the step and the OS.

The number was assigned by SFC when the step was generated and is unique in this sequencer. It is read-only and cannot be modified. The number sequence has no gaps, which means when a step is deleted and a new one is later inserted, the new step is assigned the number of the previously deleted step.

You can assign a flag to the step by clicking the "Confirmation" check box. This flag determines the behavior of the step during execution in the AS in the "T/T and O" modes. Follow-on transitions after steps with this flag will only become active (enable the next step) once they have been met and confirmed by the operator (as in "T and O" mode). Without these flags, the successor transitions enable the next steps once their conditions are satisfied (as in the "T" mode).

With "Minimum:" you can set the minimum time that a step should remain active regardless of whether the successor transition is already satisfied.

With "Maximum:" you can specify a duration for the time monitoring that specifies the maximum time for which the step may be active.

In the "Comment" and "OS Comment" boxes, you can enter texts, for example, a brief description of the step. The comment for the step can contain up to 80 characters and the OS comment up to 512 characters. The OS comment is used in process control for visualizing the step.

Note

The run times are based on the CPU time and not on the number of OB cycles. Cyclic interrupt OBs in the CPU are started at the same intervals, but the time when they are processed can change due to jobs with higher priority, temporary overload of the CPU or shorter run times of the blocks.

Example: in a 1-second cycle, the configured time can be up to 1 second longer than the set time (2 s). To maintain a maximum run time of 2 cycles, you will have to set a run time of 1.5 seconds here.

"Actions (technological)" tab

This tab is only available and can be edited for the SFC type. The tab is read-only for an SFC instance.

In this tab, you can configure the following technological actions for the steps of the SFC type to be performed: Up to 50 technological actions can be configured.

A technological action can be created or edited for each line using the "Action (technological)" button.

Technological actions have the "VOID" data type.

For more information, refer to "Technological actions - Basics and definitions of terminology (Page 136)".

"Initialization", "Processing", "Termination" tabs

The tabs for the processing phases (actions) "Initialization", "Processing" and "Termination" all have the same layout. This is where you configure the statements that control the actions that take place during initial processing, normal processing and termination processing of the step.

The statements are adopted as OS comments if the check box is activated (check mark) in the relevant line.

With the menu command **Options > Edit OS Comment...**, you can later specify in a dialog box how the OS comments should be used in each processing phase:

- Leave unchanged
- Use all statements
- Use no statements

You can also specify how any edits to the OS comments are to apply in this dialog box:

- The entire chart folder
- The current chart
- The selected steps

Documenting the configuration

You can document the configured actions of the step. By clicking "Print" in the Object Properties dialog, you obtain a log of the step with information about the properties and statements for initialization, processing and termination.

Changes to the configuration

If you change the configuration of a step (name, comment, OS comment, attributes, assignments), you can compile changes only (**SFC > Compile...**, range: **Changes** menu command) and download while the CPU is in RUN mode (**CPU > Download...**, **Download mode: Changes**) without first deactivating the active SFC.

Detailed information on editing the step can be found in the sections:

- How to edit actions in a step (Page 134)
- How to edit addresses in a step (Page 137)
- How to copy object properties of SFC elements (Page 128)

12.4.2.2 How to edit actions in a step

Introduction

You enter the statements for actions in a formatted dialog box.

Editing actions

Note

Note that an SFC type can only access its own interface and not external objects.

For each step, you can formulate up to 50 statements per action. Ten of these are visible in the dialog box. You can change the visible section with the scroll bar.

Each statement line consists of the following elements:

- The button with the line number for selecting the line
- The check box that you can use to select the statement as an OS comment
- A text box for the left (first) address, the operator, and the right (second) address.
- The "f(x)" button for creating or editing a calculation.
Depending on the tab, the button for the left and/or right text box of a statement line may be present.
For configuring a calculation, the associated text box in the statement line must be empty. Otherwise, the associated "f(x)" button is deactivated.

You can use the number buttons at the start of each line to can select a statement in order to copy it, delete it or insert a previously copied statement before it. Once you have selected a button, you can right-click to access the available menu commands.

To copy actions, you can completely copy the statements of an action with the shortcut menu command **Copy/Paste Action** and paste them in another action. The line to be copied does not have to be selected for copying. However, a line must be selected when pasting. Otherwise, the action cannot be pasted. In this way, you can easily copy the statements from the "Initialization" action to the "Termination" action, for example.

If you have placed the text cursor in an address box, the shortcut menu only displays the functions required for editing the contents.

Texts with a yellow background reference objects that no longer exist or do not yet actually exist (block inputs/outputs, charts, runtime groups).

This reference is a textual interconnection that you can make with the menu command **Options > Make Textual Interconnection** if the interconnection partner actually exists. The textual interconnection is displayed on a yellow background in the line of the address. The same highlighting is used if a previously real interconnection becomes a textual interconnection due to deletion of the interconnection partner.

Textual interconnections that cannot be made are ignored during compilation. A corresponding warning is entered in the log indicating which input/output still has a textual interconnection. An unmade textual interconnection is also tolerated in the download.

Note

Interconnections cannot be set up if the target of the interconnection is not unique, in other words, its name occurs more than once. The interconnection is also displayed as a textual interconnection in this case and it cannot be made.

Example: The object name of an SFC is the same as the symbolic name of a DB.

You can find additional information on editing the step in the sections:

- How to edit addresses in a step (Page 137)
- Valid data types (Page 151)
- Overview of SFC calculations (Page 151)

12.4.2.3 Technological actions - Basics and definitions of terminology

Overview

Programming of steps, as well as their pre- and postprocessing of the sequencer, can be done quickly and easily on a technological language level (referred to as "technological actions").

For this purpose, programming elements such as functions, commands, statuses, and language elements (IF, IF_ELSE, and STATEMENT_LIST) have been implemented. Multiple commands can also be added sequentially using the STATEMENT_LIST language element, referred to as "command sequence". A command or command sequence can also be conditionally used in the IF and IF_ELSE language elements (that is, below the THEN and ELSE branches), referred to as "conditional commands".

Similar to the programming of a step in the SFC editor, a command can be freely programmed in the CFC editor. A command or command sequence can therefore also be referred to as "sub-step". Free programming for a command is done by opening the command object in the CFC editor and configuring the command properties using the so-called "assignment statements" in the "Initialization", "Processing", and "Termination" tabs of the command properties window. Commands which have assignment statements either only in the "Initialization" tab or only in the "Processing" tab of the command properties window can be used in the pre- and postprocessing of a sequencer.

Behavior of the SFC runtime system from the user view:

- Conventional programming of a step in the "Initialization", "Processing", and "Termination" tabs is still supported. If a technological action is also added, the commands therein are executed depending on the programmed tabs in the respective "Initialization", "Processing", or "Termination" phase before the conventional programming takes place.
- A command or command sequence can be understood as a sub-step abstractly represents an atomic operation, i.e. all programmed tabs (I, P, T) of the command properties window in the CFC editor are considered by the runtime system and each is fully executed. Regarding a step, a command sequence without any IF or IF_ELSE condition is executed similar to the conventional assignment statements in the "Initialization", "Processing", and "Termination" tabs.
- For conditional commands or command sequences, the runtime system manages an IF statement with a condition and a THEN branch or an ELSE branch, if any.
- IF statements nested in one another will be supported up to a maximum nesting depth of seven. Note that the STATEMENT_LIST language element is also considered in the calculation of the nesting depth.
- All conditions of IF statements within a technological action are calculated once prophylactically in each cycle at the top level before the execution of the conditional commands.

- If the condition of the IF statement is fulfilled (=TRUE), the commands below the THEN branch will be managed by the runtime system as to be executed. The corresponding assignment statements of the commands in the THEN branch are executed in this cycle and the subsequent cycles by the runtime system depending on the programmed tabs.
- If the condition of the IF statement is unfulfilled (=FALSE), the commands below the ELSE branch, if any, will be managed by the runtime system as to be executed. The corresponding assignment statements of the commands in the ELSE branch are executed in this cycle and the subsequent cycles by the runtime system depending on the programmed tabs.
- Conditional commands of the technological tabs and the assignment statements of the commands are always executed one after the other in separate cycles, never simultaneously. If the assignment statements of the affected command or command sequence have not all been executed, a step change or processing change between the THEN branch and ELSE branch is disabled.
- If the calculation result of the condition of the IF statement changes (from TRUE to FALSE or vice versa), then the THEN branch and the ELSE branch, if any, will be executed one after the other in the subsequent cycles. The change of the calculation result for the condition also triggers a termination (execution of the assignment statements in the "Termination" tab) of the affected command or command sequence of the THEN branch or ELSE branch.
- In the context of conditional commands, there may be the circumstance that the condition is only fulfilled when the runtime system is already in the "Processing" phase and the "Initialization" phase has already been run through. In this case, the runtime system executes all assignment statements of the command sequence from the "Initialization" tab substitutionally in the "Processing" phase. If the conditional command or the command sequence also contains assignment statements in the "Processing" tab, the runtime system must subsequently remain in the "Processing" phase for at least one more cycle, regardless of the condition. The alternative command sequence (THEN branch or ELSE branch) is not considered in this cycle, i.e. its execution is omitted even if the condition is fulfilled. If the transition condition is fulfilled, the switching of the transition is suspended in this case.
- If the transition condition is fulfilled, all assignment statements of active commands or command sequences from the "Termination" tab are executed in the course of the step change, even in the case of conditional commands.
- When using conditional commands (using IF, IF_ELSE language elements) in steps, there is always an increased storage capacity requirement (code memory) compared to conventional programming with alternative branches.
Please note that the increased storage capacity requirements must be taken into consideration when calculating the reserves in these cases. In particular, sufficient reserves must be maintained for program extensions.

12.4.2.4 How to edit addresses in a step

Selecting addresses

Note

Note that an SFC type can only access its own interface and not external objects.

Edit or process the addresses for the statements as follows:

1. Drag the inputs/outputs or characteristics from the interface editor or characteristics editor to the address box.
Note: Several inputs/outputs can exist for a characteristic, for example, for setpoints, timers, and block contacts. In this case, a list of available inputs/outputs is displayed before you insert an address and you can select the required I/O.
2. Drag the block inputs/outputs from CFCs directly to the address box.
3. Click "Browse".
The "Browse" dialog box opens.
The "Browse" dialog has four different tabs:

- "Plant View"
- "Component View"
- "Runtime Groups"
- "Icons"

When you browse (through CFCs in the Plant or Component view, for example), all the available objects of the chart folder are found and displayed.

You limit the list of inputs/outputs by clicking "Filter...". Then, only the inputs/outputs relating to the current configuration are shown. You can find additional information on filters in the section: How to filter block inputs/outputs (Page 141)

4. Select the desired chart and then select the block and the I/O from it.
5. Click "Apply" to enter your selection in the active address box (or drag it to the address box).
The dialog box remains open until you explicitly close it with the "Close" button or you close the "Object Properties" dialog.

If you make the selection by means of the "Browse" dialog box or using a drag-and-drop operation from the CFC, the entries are expanded to include the path of the plant hierarchy (if this exists). The hierarchy path and the chart name are separated by double backslashes (\\).

Notes

- Accesses to CFC blocks can also be modified in the relevant CFC. This modification, however, is restricted to "rewiring" SFC accesses; in other words, moving the access to another block I/O (with <Alt> + drag-and-drop).
- Do not write any values to outputs of blocks and charts (CFC and SFC). Generally, these are overwritten again by the processing of the block or chart.

Example: Turning an SFC on and off

The statement "<SFC_chart>.INTONOFF := TRUE", activates an SFC.

The statement "<SFC_chart>.INTONOFF := FALSE" deactivates an SFC; that is, its final step is processed before the chart is terminated.

Selecting value identifiers

To enable the value identifiers to be displayed in SFC, select the **Options > Customize > Layout...** menu command and then, in the dialog that opens, check the "Parameters: Value identifier" check box.

If you insert a block I/O with a value identifier in the address box, the value identifier is also entered in the right (empty) address box. If there are several value identifiers for this I/O, you can open the list box with the value identifiers in the right address box and select the desired value identifier for this address from the drop-down list.

Editing addresses

When making textual entries, make sure that names are consistent. A symbol that does not exist in the symbol list (or a referenced symbol that is subsequently renamed) cannot be checked in the editor and is set up as a textual interconnection.

Notes

- When you enter statements, they are checked for consistency to ensure the correct syntactic and semantic formulation. The program, for example, checks if the data types of the logically combined addresses are compatible.
- In the addresses, you can enter accesses to CFC block inputs/outputs whose block does not yet really exist in the CFC. These statements are displayed in the statement line on a yellow background and are set up as textual interconnections.

You can find more information on textual interconnections in help for the CFC editor or in the section "Working with textual interconnections" of the *CFC for SIMATIC S7* documentation.

Go to point of use

If you want to see the point of use of an address, place the cursor in the address box and click "Go To". The corresponding chart opens. If the address is an input or output of a block in the CFC, the I/O of the relevant block is selected. You can also move to the point of use by double-clicking on the address.

With shared addresses, you jump to HW Config if the address is known there. Otherwise, an error message is displayed.

The "Go To" function cannot be used for runtime groups.

12.4.2.5 Valid entries for addresses

Left address

SFC

The following entries are permitted in the SFC for the left address:

- A block input of a block in a CFC
Syntax: <ChartName>\<BlockName>.<I/OName>
- An SFC (switch on/off)
Syntax: <ChartName>.INTONOFF
- An interface I/O of the SFC/CFC (own or another SFC)
Syntax: <ChartName>.<I/OName>
- A runtime group (switch on/off)
Syntax: <GroupName>.EN
- A shared address to which write access is permitted
Syntax: <SymbolicName>
- A text interconnection
Syntax: <any characters> or
path reference to an object that does not yet exist (syntax as for block I/O)
- Structures

Syntax	Structure:	<I/O_name>
	Substructures:	<I/O_name>.[<Substructure>...]<Substructure>
	Structure element:	<I/O_name>.[<Substructure>...]<Element>

SFC type

The following entries are permitted in the SFC type for the left address:

- Interface elements (inputs/outputs, structures, substructures, structure elements)

Syntax	Input/output:	<I/O_name>
	Structure:	<I/O_name>
	Substructures:	<I/O_name>.[<Substructure>...]<Substructure>
	Structure element:	<I/O_name>.[<Substructure>...]<Element>

- A text interconnection
Syntax: <any characters> or
path reference to an object that does not yet exist (syntax as for block I/O)

Right address

The following entries are permitted in the SFC for the right address:

SFC

- A block I/O of a block in a CFC
Syntax: <ChartName>\<BlockName>.<I/OName>
- An SFC (on/off)
Syntax: <ChartName>.BUSY
- An interface I/O of the SFC/CFC (own or another SFC)
Syntax: <ChartName>.<I/OName>

- A runtime group (on/off)
Syntax: <GroupName>.EN
- A shared address
Syntax: <SymbolicName>
- A text interconnection
Syntax: <any characters> or
path reference to an object that does not yet exist (syntax as for block I/O)
- A constant
Syntax: depends on data type
For the data types BOOL, BYTE, INT, DINT, WORD, DWORD, instead of the absolute numeric values, you can also specify their symbolic identifiers (value identifiers).
- Structures

Syntax	Structure:	<I/O_name>
	Substructures:	<I/O_name>.[<Substructure>...]<Substructure>
	Structure element:	<I/O_name>.[<Substructure>...]<Element>

SFC type

The following entries are permitted in the SFC type for the right address:

- Interface elements (inputs/outputs, structures, substructures, structure elements)

Syntax	Input/output:	<I/O_name>
	Structure:	<I/O_name>
	Substructures:	<I/O_name>.[<Substructure>...]<Substructure>
	Structure element:	<I/O_name>.[<Substructure>...]<Element>

- A text interconnection
Syntax: <any characters> or
path reference to an object that does not yet exist (syntax as for block I/O)

12.4.2.6 How to filter block inputs/outputs

Procedure

In the "CFC Charts" tab of the "Browse" dialog, you can use a filter to display only the block I/Os that are relevant for assignment in the current address box. In the "Filter – I/Os" dialog box, you can specify the criteria that decide which I/Os of the selected block are displayed or not displayed.

Filter criteria

The following filter criteria are ANDed with each other.

Filter criterion	Meaning
List boxes	
Name: Comment:	In the list boxes, you can enter the text to identify an I/O name or I/O comment. You do not have to enter the complete name/comment. Instead, initial letters can be entered (in order without gaps).
Visible in CFC: <All>	An input/output can be visible in the CFC or, although present, invisible (S7_visible := 'false').
Listed: <All>	Listed means that these I/Os (attribute: S7_edit := 'param' or S7_edit := 'signal') can be edited in a list in the process object view in the SIMATIC Manager without having to open the relevant chart (menu command Options > Process Objects > Select I/Os...).
Operator control and monitoring:	These are block I/Os intended for operator control and monitoring on the OS (attribute: S7_m_c := 'true').
Interconnected in CFC:	As a practical matter, SFC can only use the I/Os interconnected in CFC for read accesses. If block inputs are write-accessed, these values would be overwritten by the values coming from the CFC interconnection.
Write access in SFC:	The SFC cannot write to one block I/O more than once. These I/Os can therefore be hidden if they have been write-accessed.
Check boxes	
Inputs (IN) Outputs (OUT) In/outs (IN_OUT)	If you reset a check box, no I/Os of this I/O type are displayed.
Suitable I/Os only:	'Suitable I/Os only' means that the assignment for this address must be suitable for the entry for the other address. Example: If you have assigned an I/O with data type INT in the first address, only I/Os of data type INT are displayed in the filter when you click in the box of the second address. Filtering is performed only when criteria can be explicitly defined; in other words, if an address is a constant, for example, all I/Os are displayed unfiltered and can be selected for the second address.

You can reset all the changed settings using the "Default" button. The default for the list boxes is <All> and all check boxes are selected.

12.4.2.7 How to access structures

Introduction

In the **step properties**, you can configure structure assignments. Structures, substructures or structure elements are possible according to the following syntax:

Structure:	<I/O_name>
Substructures:	<I/O_name>.[<Substructure>...]<Substructure>
Structure element:	<I/O_name>.[<Substructure>...]<Element>

You can enter this information directly in the statement lines or by means of the "Browse" dialog box. In the "Browse" dialog box, you can enter the desired structure in the statement line by double-clicking or by selecting the structure and clicking "Apply" or by selecting the

shortcut menu command **Apply I/O**. You apply substructures or structure elements in the "Structure" dialog box.

In the **Transition Properties**, it is only possible to compare structure elements.

Note on SFC types

The use of structures of I/O type IN_OUT is not permitted. Although SFC tolerates the assignment, this is detected as an error when the S7 program is compiled.

Selecting a structure

1. Select the desired structure in the "Browse" dialog box.
2. Select the menu command **Open Structure** from the shortcut menu.
The "Structure" dialog box opens.
The name of the structure is specified in the title bar. The table of the dialog box contains the same columns as found in the "I/Os" column of the "Browse" dialog box.
3. Double-click the table line or click "Apply".
The substructure, structure, or structure element is entered in the "Properties" dialog box.

Note

Structures and substructures are not entered in the "Properties" dialog of the transition.

Meaning of the columns in the "Structure" dialog box

Name

Shows the names of the structure elements of the entire structure, for example:

var01_char		
var04_struct		
	var02_bool	
	var02_int	
	var06_struct	
		var03_word
		var03_date
var01_time		

Data type

Shows the data type of the structure element (BOOL, DINT or CHAR) or STRUCT for substructures.

I/O

Shows the I/O type of the structure element or substructure (IN, OUT, IN_OUT).

CFC interconnection

If the structure is interconnected in CFC, this is indicated here by an 'x' (entire column).

SFC access (writing)

An existing (write) access by the SFC to a structure element is indicated by an "x".

Comment

Shows the comment of the structure element, for example: "1=CPU DPV1 mode".

12.4.3 Editing transitions

12.4.3.1 How to edit the transition

Opening the "Properties" dialog box of the transition

1. Double-click the transition to be edited or select the menu command **Edit > Object Properties....**
The dialog box opens.
2. Edit the properties (name, comment), formulate the conditions and enter the OS comment.

The dialog box is divided into three tabs:

- "General"
- "Condition (technological)" (for SFC type only)
- "Condition"
- "OS Comment"

"General" tab

In the "General" tab, you can enter or change the name and comment.

The number was assigned by SFC when the transition was generated and is unique in this sequencer. It is read-only and cannot be modified. The number sequence has no gaps. If a transition is deleted and a new one inserted later, the new transition is given the number of the previously deleted transition.

"Condition (technological)" tab

This tab is only available and can be edited for the SFC type. The tab is read-only for an SFC instance.

In this tab, you can configure a technological condition of the SFC type.

Only a technological condition can be defined in this tab.

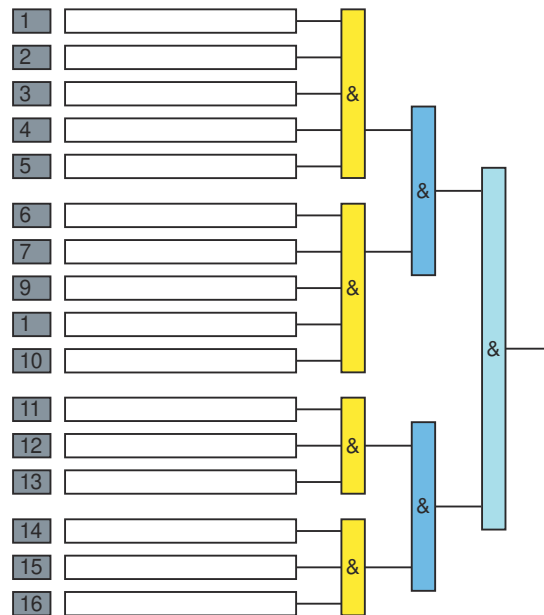
The "Condition (technological)" button is used to create or edit the technological condition.

The 'Corresponding row in the "Condition" tab' selection box is used to assign the configured technological condition here one of the 16 conditions in the "Condition" tab.

"Condition" tab

In the "Condition" tab, you can combine 2 x 5 and 2 x 3 conditions using three-stage transition logic to form a Boolean expression.

As an alternative to addresses or textual interconnections, a calculation can be configured in the statement line of a condition using the "f(x)" button.



"OS Comment" tab

In the "OS Comment" tab, you can enter a text for every condition. This text will be displayed on the OS during process control. The text of the condition is entered by default.

Documenting the configuration

You can document the configured conditions of the transition. By clicking "Print" in the Object Properties dialog, you obtain a log of the transition with information about the properties and parameters of the conditions.

Changes to the configuration

If you change the configuration of a transition (name, comment, condition, OS comment), you can compile changes only (**SFC > Compile...**, range: **Changes** menu command) and download them while the CPU is in RUN mode (**CPU > Download...**, **Download mode: Changes** menu command) without having to first deactivate the active SFC.

You can find additional information on editing the transition in the sections:

- Formulating the conditions of a transition (Page 146)
- How to edit addresses in a transition (Page 146)
- How to edit OS comments in a transition (Page 148)

- How to copy object properties of SFC elements (Page 128)
- Overview of SFC calculations (Page 151)

12.4.3.2 Formulating the conditions of a transition

Introduction

You specify the step control conditions for the selected transition in the "Condition" tab.

Note

Note that an SFC type can only access its own interface and not external objects.

Formulating a transition

You formulate the transition as a Boolean expression, which can consist of 2 x 5 and 2 x 3 conditions (expression segments). The conditions are combined using three-stage transition logic.

The Boolean operators are designed as buttons. By clicking the operator, you can change it from "AND (&)" to "OR (≥ 1)". To make a "NAND" from an "AND" and a "NOR" from an "OR", click the output of the operator. The negation is displayed by a bold dot on the output line.

Alternative operation: If you position the cursor over an operator, you can right-click to open the shortcut menu, which displays all the available Boolean operators. After you select the menu command, the operator is changed accordingly.

You can find additional information on editing the transition in the sections:

- How to edit addresses in a transition (Page 146)
- Valid data types (Page 151)

12.4.3.3 How to edit addresses in a transition

Selecting addresses

You can edit the addresses or define them in the "Browse" dialog box.

You can find additional information about this in the section "How to edit addresses in a step (Page 137)".

The advantage of browsing is that the addresses are then unique. As a third possibility, you can use a drag-and-drop operation to move the block inputs/outputs from CFCs directly to the box for the address.

Note

Note that an SFC type can only access its own interface and not external objects.

As when selecting addresses for the steps, you can also enter the value identifiers of the inputs/outputs for the transitions.

If you make the selection by means of the "Browse" dialog box or using a drag-and-drop operation from the CFC, the entries are expanded to include the path of the plant hierarchy (if this exists). The hierarchy path and the chart name are separated by double backslashes (\\).

Texts with a yellow background reference objects that no longer exist or do not yet actually exist (block inputs/outputs, charts, runtime groups).

This reference is a textual interconnection that you can complete during compiling or explicitly using the menu command **Options > Make Textual Interconnection** if the interconnection partner actually exists. The textual interconnection is displayed on a yellow background in the line of the address. The same display is used if a former actual interconnection becomes a textual interconnection due to subsequent deletion of the interconnection partner.

Textual interconnections that cannot be made are ignored during compilation. A corresponding warning is entered in the log indicating which input/output still has a textual interconnection. An unmade textual interconnection is also tolerated in the download.

Notes

- Accesses to CFC blocks can also be modified in the relevant CFC. This modification is, however, restricted to "rewiring" SFC accesses; in other words, moving the access to another block I/O (with <Alt> + drag-and-drop).
- Access to structures is possible but is restricted to the structure elements. In the "Browse" dialog, you can open the structure by right-clicking the structure and selecting the shortcut menu command **Open Structure** and then select the desired structure element.
- Interconnections cannot be set up if the target of the interconnection is not unique, in other words, its name occurs more than once. The interconnection is then displayed as a textual interconnection and cannot be made. Example: The object name of an SFC is the same as the symbolic name of a DB.
- When you enter conditions, they are checked for consistency to ensure the correct syntactic and semantic formulation of the conditions. The program also checks if the data types of the logically combined addresses are compatible.
- The predefined logic is normally sufficient. If you require more complex formulations, you can create them with blocks in a CFC and enter the results that were calculated in the chart as the address in the transition conditions.

Go to point of use

If you want to see the point of use of an address, place the cursor in the address box and click "Go To". The corresponding chart opens. If the address is an input or output of a block in the CFC, the I/O of the relevant block is selected. You can also move to the point of use by double-clicking on the address.

With shared addresses, you jump to HW Config if the address is known there. Otherwise, an error message is displayed.

The "Go To" function cannot be used for runtime groups.

12.4.3.4 How to edit OS comments in a transition

Entering an OS comment

In the "OS Comment" tab, you can enter a text for every condition. This text will be displayed on the OS during process control. The text of the condition is entered by default.

You cannot change the Boolean operators in this part of the dialog box; the buttons are only for visualizing the existing logic operations.

If the comment lines are marked with the following character, the comment is derived from the condition (default).



Calculations:

If a calculation is configured in a statement line in the transition, the name of the calculation is displayed in the associated line on the "OS Comment" tab.

Notes

- The OS comment for the SFC type and SFC instance can be different. If the SFC type has an I/O that is interconnected in the SFC instance, the interconnection partner is indicated for the instance. If this is an output with multiple interconnections, only one of the interconnections is displayed.
If the inputs/outputs are not interconnected, the OS comment of the SFC type and SFC instance are identical.

Example:

	SFC type	SFC instance
Interconnected	RUN = TRUE	chart\block.io = TRUE
Not interconnected	RUN = TRUE	chart\sfc_instance.RUN = TRUE

- Provided you do not edit the OS comment, it will be adapted automatically whenever the compare condition is changed. The OS comment is also automatically adapted when the referenced CFC blocks are renamed, copied, moved and deleted.
- If you change the condition and a user-edited comment text is associated with this condition, this comment text is not automatically adapted. In this case, a dialog box appears indicating that you should check the comment text. You can cancel the change in this dialog box by clicking "Cancel".

- You can initialize an individual OS comment in the SFC Editor by deleting the OS comment. The text of the compare condition is then used again as the OS comment and from this time onwards is automatically adapted whenever the condition is changed, providing you do not edit the OS comment.

Alternative action: You enter the original OS comment (text of the condition) again using the shortcut menu command **Use Default**.

- You can edit the OS comments later.
The menu command **Options > Edit OS Comments...** opens a dialog box in which you specify how you want to use the OS comment for the transitions:
 - Leave unchanged
 - Use all conditions

You can also specify how any edits to the OS comments are to apply in this dialog box:

- The entire chart folder
- The current chart
- The selected steps/transitions

12.4.4 Shared addresses and valid data types

12.4.4.1 Shared addresses

Shared addresses

Shared addresses are connection partners that are outside the CFC/SFCs. Shared addresses have read and write access. Note that the data type of the shared address and of the accessing object (address of a statement or condition in SFC, block I/O in CFC) must match.

You can specify the addresses in symbolic or absolute form.

You can find additional information in the following sections:

- How to perform symbolic addressing (Page 150)
- Absolute addressing (Page 149).

12.4.4.2 Absolute addressing

Rules

The address of a value is used for absolute addressing.

Follow these rules:

- The address must not be located in the area reserved for compiling the CFC menu command (**Options > Settings for Compilation...**).
- The address must not be located in a value range that does not exist for the current CPU.

- The address must be compatible with the data type of the accessing object (address of a statement or condition in SFC, block I/O in CFC). If the case of a peripheral input word (PIW), the first interconnection determines the type.
- The data width must accommodate the address and I/O.

Examples

The following examples illustrate how to specify addresses in absolute form:

Address	Remark
I5.1	Access to input 5.1
Q5.1	Access to output 5.1
M6.7	Access to memory bit 6.7
MW10	Access to memory word 10
DB10.DW20	Access to data block 10, data word 20
DB20.DX2.1	Access to data block 20, bit 2.1

12.4.4.3 How to perform symbolic addressing

Symbolic addressing

With symbolic addressing, you specify the address to be processed as a symbol. Symbols and addresses are assigned in the symbol table. The menu command **Options > Symbol Table** opens the symbol table where you can add or edit entries.

When you make your entries in the symbol table, remember to adhere to the STEP 7 conventions.

Checking the syntax

You can use the menu command **Chart > Check Consistency...** to check that all the entries are present in the symbol table before you compile. You can display the log of the consistency check with the menu command **Options > Logs...**

Examples

Below, you can see several examples of possible entries. The following examples illustrate how addresses can be specified symbolically.

Address	Remark
Limitswitch	For example access to an input bit. The absolute address is defined in the symbol table.
Recipe.setpoint	Access to a data block (recipe). The structure or the type of the data block determines the data block element (value).
DB17.setpoint	As above, but with absolute addressing of the data block

Address	Remark
Recipe.DW5	As above, but with absolute addressing of the data word
"I5.1"	Access to the symbol with the name E5.1. To avoid the absolute address with the same name being addressed instead of the symbol, the symbol is enclosed in quotation marks.

12.4.4.4 Valid data types

Valid data types

The following data types are permitted for formulating actions and conditions in SFC:

Data type	Permitted in steps	Permitted in transitions
BOOL, BO (Page 306)	Yes	Yes
BYTE, BY (Page 306)	Yes	Yes
CHAR, C (Page 306)	Yes	Yes
DATE, D (Page 307)	Yes	Yes
DINT, DI (Page 307)	Yes	Yes
DWORD, DW (Page 307)	Yes	Yes
INT, I (Page 308)	Yes	Yes
REAL, R (Page 308)	Yes	Yes
S5TIME, T5 (Page 308)	Yes	No
STRING, S (Page 309)	Yes	No
STRUCT, ST (Page 309)	Yes	Yes *)
TIME, TI (Page 310)	Yes	Yes
TIME OF DAY, T (Page 310)	Yes	Yes
WORD, W (Page 310)	Yes	Yes

*) only to structure elements

12.4.5 Calculations in steps and transitions of SFC

12.4.5.1 Overview of SFC calculations

Overview

Calculations can be configured in steps, transitions or sequencers in the SFC Editor. The calculations are displayed with current values in test mode for checking.

Calculations can be configured and edited in an SFC or SFC type, but not in an SFC instance.

Configuring

Configuring is carried out in the "Calculation for SFC" dialog box of the SFC Editor.

The dialog box is opened with the "f(x)" button of a statement line in the following properties dialog boxes:

- Of steps ("Initialization", "Processing", "Termination" tabs)
- Of transitions ("Conditions" tab)
- Of sequencers (menu command "Edit > Sequencer Properties"; "Start condition", "Preprocessing", "Postprocessing" tab)

For configuring a calculation, the corresponding field in the statement line must be empty. Otherwise, the "f(x)" button is deactivated.

You can find additional information on this in the description below.

Test mode

In test mode, the current values of the function inputs and calculation results are displayed in their own column in the "Calculation for SFC" dialog box.

You can find additional information about this in the section "Calculations in steps and transitions during testing (Page 284)".

Error handling in the formula

There is no error handling in the formula, such as in case of overflow, division by zero, etc.

Printing

In the SFC Editor, all conditions configured for steps and transitions can be recorded in a printout if test mode is not active.

If a calculation is configured in a statement line, the name of the calculation is included in the printout. This is the same description that is displayed in the statement line of the step or transition.

For long names of calculations, the shortened form with leading dots "..." is displayed to make the printout easier to read.

To completely document the calculation, the content of the calculation is also included in the printout at the end of the condition block.

Calculations in the SFC Visualization

The current values of the function inputs and calculation results of the calculations can be displayed in the SFC visualization in the process pictures of the operator station. You can find additional information on this in the documentation for SFC Visualization.

Overview of configuration

The "Calculation for SFC" dialog box contains the following objects:

- Table
The calculation is configured in the table.
- "Formula" field:
The "Formula" field is used for the textual representation of the currently configured calculation and cannot be edited.
- Various buttons with functions as in the properties dialog boxes of the steps and transitions.

Checking data types during configuration

When the calculation is configured, the compatibility of the data types is automatically checked. This occurs, for example, when inserting functions or configuring addresses in the "Interconnection" column.

- If the data types are incompatible, a message occurs and the selected action or data type is rejected.
- If the check was successful and the data types are compatible, the number of possible data types for the function inputs and the function result may decrease.
If only one data type remains after the check, the corresponding line is shown with "gray" background color. In other words, the data type is explicitly assigned. You can find additional information in the description of the "Data Type" column below.
- When inserting the first function in a calculation, the data type compatibility with the corresponding address is checked in the properties dialog box of the step or transition.
- Variables, constants or lower-level functions:
When configuring variables or constants and when inserting lower-level functions, a type conversion is performed if needed when the data type compatibility is checked. The type conversion is allowed only to a data type with a larger range of values.
- Constants:
When a constant is configured as an address, the possible data types of the specified constant string are determined during the check of the data type. For example, string "1" is compatible with data types BOOL, BYTE, WORD, DWORD, INT, DINT and REAL. Thereafter, it is checked whether at least one of these data types is included in the set of possible data types for the function input.
- Textual interconnection:
When a textual interconnection is configured as an address, no checks are performed for the data type compatibility.
In this case, the check for data type compatibility takes place during the "Close textual interconnections" operation instead. Textual interconnections are shown with a yellow background color, as in the properties dialog box of a step or transition.

Structure of the table

The calculation is configured in the table. The desired functions are inserted in the "Calculation" column for this.

You can find the possible functions and data types in the section "Scope of functions in SFC calculations: (Page 158)".

"Calculation" column

- The name of the calculation is displayed in the top line of the column and can be edited there. This name is transferred to the higher-level properties dialog box of the step or transition in which the calculation is configured.
When you leave this cell, a plausibility check is performed, for example, for uniqueness, permissible characters and character length. Non-unique names are made unique when the data is entered in the higher-level properties dialog box, for example, by adding "(1)".
- Deleting the calculation:
To delete the entire calculation, you can use the "Delete calculation" menu command in the shortcut menu for the top cell.
- All the lines underneath display the configured functions for the calculation in a tree structure. The calculation result of a sub-function is represented as a "root", and the associated function inputs are represented as "branches" below it.

- The calculation result of the uppermost function in this column is simultaneously the overall result of this calculation.
- You can insert and delete functions or function inputs in the column or replace functions.
- When a function is deleted, all associated function inputs are deleted. Subordinate functions that are configured for one of the function inputs to be deleted are also deleted.
- Nesting depth in calculations:
Functions can be inserted at the function inputs of a function. These inserted functions are then a subordinate level. The maximum nesting depth is limited to 8 subordinate levels.

"Data Type" column

- The data types for the function inputs and the function result are configured and displayed in the column. If multiple data types are possible, you can select the desired data type in a selection box.
- Background color of a cell is "Gray" or "White":
If only one data type is possible, the associated cell in the table is shown with "Gray" background color. If, on the other hand, multiple data types are possible, the desired data type can be selected from a drop-down list and the background color of the cell is "White".
- Background color "Dark gray" when data type is specified manually:
The status of the data types is a requirement for the "Apply" button. This means that an explicit data type must be specified for all function inputs/results in the calculation in order for the calculation data to be transferred to the properties dialog box of the step or transition. Thus, no cell in the "Data Type" column is permitted to have "White" background color. If there are still cells with "White" background color, you must manually specify the data type. To specify the data type, you select the "Data Type" cell of a function input that has "White" background color. The shortcut menu for the selected cell contains the menu commands for specifying the data type and canceling the specification. After manual specification of the data type, the cell is displayed with "Dark gray" background color.

"Interconnection" column

- The addresses, constants or textual interconnections for the function inputs are configured and displayed in this column.
For function inputs for which a subordinate function is configured, no addresses can be selected or configured.
- The following options are available for configuring the address:
 - Manual entry:
For manual entry, such as for configuring constants or textual interconnections, there are various functions available in the shortcut menu for the cell.
 - When a calculation is configured in an SFC:
The desired variable or block I/O can be moved from an open CFC to the "Interconnection" cell of the calculation using drag-and-drop.
Alternatively, the "Browse..." button can be used to open a dialog box for selecting an address for the selected function input.
 - When a calculation is configured in an SFC type:
Only the interface of the SFC type can be accessed for selecting addresses. In this case, a characteristic can be moved from the characteristic list or an I/O from the I/O list of the SFC type to the "Interconnection" cell of the calculation using drag-and-drop.

You can find additional information on configuring in the section "Configuring a calculation in SFC (Page 155)".

12.4.5.2 Configuring a calculation in SFC

Introduction

The description below shows how to create a new calculation in SFC in one of the following properties dialog boxes:

- Of steps ("Initialization", "Processing", "Termination" tabs)
- Of transitions ("Conditions" tab)
- Of sequencers (menu command "Edit > Sequencer Properties"; "Start condition", "Preprocessing", "Postprocessing" tab)

A description for implementing a formula in a calculation in SFC is available in the section "Example for implementing a formula in a calculation for SFC (Page 160)".

Requirements

- The SFC or SFC type in which you want to configure a calculation is open in the SFC Editor and not write-protected! A calculation cannot be created or edited in an SFC instance.
- The project must be a "Multiproject" type.

Procedure

1. Open the "Properties" dialog box of the desired step or transition/sequencer in the SFC Editor. In the "Properties" dialog box, change to the tab in which you want to configure the calculation in a statement line, e.g. "Condition" tab.
2. In the statement line next to the field in which you want to insert the calculation, click the associated "f(x)" button. The "Calculation for SFC" dialog box opens.
For configuring a calculation, the associated text box in the statement line must be empty. Otherwise, the associated "f(x)" icon is deactivated.
In this case, delete the content of the field and then click the associated "f(x)" button to configure the calculation.
3. The uppermost cell in the "Calculation" column contains the name of the calculation. If the calculation is being newly created, the uppermost cell contains the text "Please insert a function".
The top cell can only be edited when a function has been added.
Select the shortcut menu command "Insert function" and one of the displayed functions for this. The selected function is inserted in the "Calculation" column. This uppermost function that was inserted first always represents the highest level of the structure or calculation. If necessary, this function can also be replaced later.
You can find additional information about the functions and data types available for selection in the section "Scope of functions in SFC calculations: (Page 158)".
4. Enter a name for the calculation in the top cell.
Various editing functions are available in the shortcut menu of the cell. The name is transferred to the higher-level properties dialog box of the step or transition.
When you leave this cell, a plausibility check is performed, for example, for uniqueness, permissible characters and character length. Non-unique names are made unique when the data is entered in the higher-level properties dialog box, for example, by adding "(1)".
5. Complete the configuration of the first inserted function.
After you insert a function, the following associated data must be configured at each function input:
 - The desired data type in the "Data type" column (see step 6).
 - An interconnection, constant or lower-level function in the "Interconnection" column (see step 7).

The order in which these configuration steps are performed may vary. For example, if you want to insert lower-level functions, it makes sense to define the data types afterwards.
6. Check for the desired data type in the "Data Type" column for the associated function inputs and change the data type, if necessary.
If only one data type is possible for a function input, the associated cell in the table is shown with "Gray" background color. If, on the other hand, multiple data types are possible, the desired data type can be selected from a drop-down list and the background color of the cell is "White".

7. At a function input, select the associated cell in the "Interconnection" column. The address for the function input is configured in this cell.

The following options are available for configuring an address:

- Manual entry:
For manual entry, such as for configuring constants or textual interconnections, there are various functions available in the shortcut menu for the cell.
- When a calculation is configured in an SFC:
The desired variable or block I/O can be moved from an open CFC to the "Interconnection" cell of the calculation using drag-and-drop.
The "Browse..." button can be used to open a dialog box for selecting an address for the selected function input.
- When a calculation is configured in an SFC type:
Only the interface of the SFC type can be accessed for selecting addresses. In this case, a characteristic can be moved from the characteristic list or an I/O from the I/O list of the SFC type to the "Interconnection" cell of the calculation using drag-and-drop.

8. If you want to integrate an additional function in the calculation, select a function input in the "Calculation" column and select the new function in the shortcut menu. A lower-level function can be inserted even for an interconnected function input. The previous entry in this cell is then replaced.

If needed, you can also add further function inputs for certain functions via the shortcut menu.

After this function is inserted, it becomes a subordinate level. The maximum nesting depth is limited to 8 subordinate levels.

The uppermost function that was inserted first is always the highest level of the structure. The second level and all other levels can then also constitute multiple functions that are configured at the function inputs of the next higher level in each case.

9. You can also replace or delete functions, if required. To do so, select the desired function in the "Calculation" column and select the corresponding menu command in the shortcut menu.

When a function is being replaced, the selection displayed in the shortcut menu depends on the new function having similar properties to the function to be replaced.

10. Check the "Data Type" column. An explicit data type must be specified for all function inputs/ results in the calculation in order for the calculation data to be transferred to the properties dialog box of the step or transition. Thus, no cell of the "Data Type" column may have "White" background color. If there are still cells with "White" background color, you must manually specify the data type.
To do so, select the "Data Type" cell of a function input whose data type is not yet explicitly specified and therefore has "White" background color.
Select the desired data type in the drop-down list of the cell.
The shortcut menu for this selected cell contains the menu commands for specifying the data type and canceling the specification.
Select the shortcut menu command "Specify data type". After manual specification of the data type, the cell is displayed with "Dark gray" background color. If you want to change the data type again, you can cancel the specification using the corresponding shortcut menu command.
Repeat the step for all other cells whose data type is not specified and whose background color is "White".
11. When all desired functions are configured and the data types are explicitly specified, the calculation data can be transferred to the properties dialog box of the step or transition.
To do this, click the "Apply" button. The "Calculation for SFC" dialog box closes.
The calculation is displayed in the statement line of the properties dialog box of the step or transition with the name entered in the calculation.

Note

Saving the calculation

The calculation has now been transferred to the statement line of the "Properties" dialog box but is not yet saved.

Saving occurs when the "Apply" button is clicked in the "Properties" dialog box of the step or transition.

Result

A new calculation has been created in a step, transition, or sequencer of SFC.

12.4.5.3 Scope of functions in SFC calculations:

Overview

The following functions and data types are available in SFC for configuring calculations in steps, transitions or sequencers.

The table shows the available functions and the possible data types of the inputs and outputs.

Function (operation)	Icon	Data type Input	Number of inputs	Data type Output (function result)
Addition (ADD)		INT, DINT, REAL	2 ... n	Same as input type
Subtraction (SUB)		INT, DINT, REAL	2	Same as input type
Multiplication (MUL)		INT, DINT, REAL	2 ... n	Same as input type
Division (DIV)		INT, DINT, REAL	2	Same as input type
Negation (NEG)		INT, DINT, REAL	1	Same as input type
Comparison "Equal to" (EQ)		INT, DINT, REAL	2	BOOL
Comparison "Not equal to" (NE)		INT, DINT, REAL	2	BOOL
Comparison "Less than" (LT)		INT, DINT, REAL	2	BOOL
Comparison "Greater than" (GT)		INT, DINT, REAL	2	BOOL
Comparison "Less than or equal to" (LE)		INT, DINT, REAL	2	BOOL
Comparison "Greater than or equal to" (GE)		INT, DINT, REAL	2	BOOL
Bitwise AND (AND)		BOOL, BYTE, WORD, DWORD	2 ... n	Same as input type
Bitwise OR (OR)		BOOL, BYTE, WORD, DWORD	2 ... n	Same as input type
Bitwise Exclusive OR (XOR)		BOOL, BYTE, WORD, DWORD	2 ... n	Same as input type
Bitwise NOT AND (NAND)		BOOL, BYTE, WORD, DWORD	2 ... n	Same as input type
Bitwise NOT OR (NOR)		BOOL, BYTE, WORD, DWORD	2 ... n	Same as input type
Bitwise negation (NOT)		BOOL	1	BOOL
Square root (SQRT)		REAL	1	REAL
Absolute value (ABS)		INT, DINT, REAL	1	Same as input type

Note

Separate objects are created in the "Charts" folder of the master data library for functions listed in the table. These objects are only used for calculations in SFC and are not identical to the library blocks that are available in CFC editors, for example.

12.4.5.4 Example for implementing a formula in a calculation for SFC

Introduction

The example below shows the principle for implementing a formula in a calculation for SFC.

The procedure described here also prevents any configuration effort.

Example formula: $a + b - c * d / e$

A detailed description of the configuration is available in the section "Configuring a calculation in SFC (Page 155)".

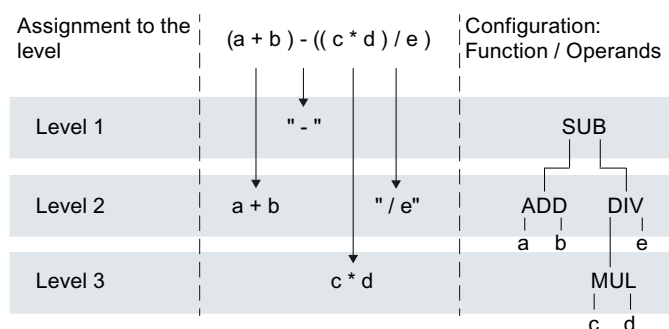
Implementation

The implementation and configuration of the formula " $a + b - c * d / e$ " in SFC generally takes place in the following steps:

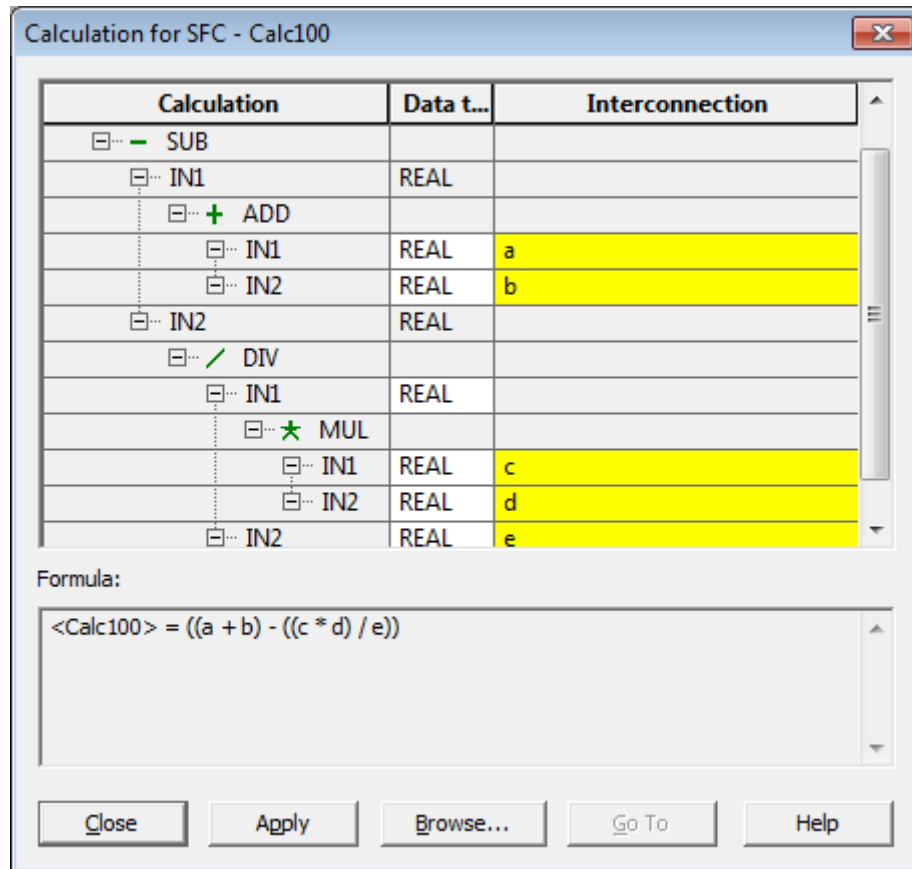
1. Setting the parentheses in the source formula
The parentheses are required to form the correct intermediate calculations in the formula. The implementation of the source formula for configuring the calculation in SFC is prepared simultaneously.

Intermediate step	Formula
Source formula	$a + b - c * d / e$
Parentheses 1	$(a + b) - (c * d / e)$
Parentheses 2 This parenthesis is required for the intermediate result of the product " $c * d$ ".	$(a + b) - ((c * d) / e)$

2. Arrangement of the individual mathematical intermediate calculations in levels
The individual intermediate calculations are assigned to specific levels as required for the subsequent configuration of the mathematical functions.



3. Configuration of the functions in the "Calculation for SFC" dialog box
 During configuration, the functions determined in step 2 are added in the "Calculation" column in the order "Level 1" to "Level 3".
 The current configuration of the calculation is displayed in the "Formula" dialog box so that it can be checked.
 The following figure shows how the calculation example "Calc100" after adding the functions.
 The operands "a" to "e" are entered as placeholders in the "Interconnection" column to provide a better overview.



4. Configuring the interconnections
 The required interconnections or constants and the data type are configured for the operands "a" to "e" at the inputs of the functions in the "Interconnection" column.

12.5 Specifying runtime properties

12.5.1 Runtime properties of an SFC

Runtime properties

The runtime properties of an SFC or SFC instance determine how the SFC is included in the chronological execution sequence within the overall structure in the CPU. These properties determine the behavior of the target system with respect to response times, dead times, or the stability of time-dependent structures, such as control loops.

Insertion in the run sequence

Each SFC is inserted in a run sequence by default. Specifically, it is inserted in a task that corresponds to an organization block (OB) in S7. SFCs can be inserted in runtime groups and thereby given the attributes of the runtime group (reduction ratio, phase offset).

An SFC type does not have any runtime properties, because it is not relevant to execution of the program. An SFC type cannot be inserted in the run sequence.

You edit the run sequence with the run sequence editor of CFC.

SFC instances are inserted in runtime groups of the associated CFC by default.

You can find additional information on editing the step in the sections:

How to edit the run sequence (Page 162)

Runtime attributes for runtime groups and SFCs (Page 163)

12.5.2 How to edit the run sequence

Editing the run sequence

You can edit the run sequence of SFCs/SFC instances in several ways. The run sequence includes the insertion position of the SFCs in the tasks and the runtime attributes "reduction ratio" and "phase offset" by means of the attributes of the runtime group.

Changing the run sequence

1. Select the button in the toolbar



or

select the **Edit > Run Sequence** menu command.

The SFC starts and the "Run Sequence" window opens. If the catalog also opens in SFC, you can close it.

2. Select the task containing the SFC you want to upgrade in the left window.
3. Select the SFC and select the "Cut" function.
4. Select the new position in the run sequence (task or object in the task) and the select "Insert".
If you have selected a task, the SFC is inserted at the beginning of the task.
If you have selected an object within the task, the SFC is inserted after the object.

As an alternative to Cut/Paste, you can also use a drag-and-drop operation to move an SFC from an open task to another task.

Inserting an SFC in other tasks

Each SFC (chart/instance) must be inserted in at least two tasks:

- In OB 100 for the startup behavior
- In **one** cyclic interrupt OB (for example, OB 32) for cyclic operation

Note

You **cannot** insert the same SFC more than once in the same task. Inserting an SFC in several cyclic tasks is not permitted because this can result in unpredictable behavior. No provision is made for insertion of the SFC in other non-cyclic tasks (except in OB 100).

Removing an SFC from a task

Select the SFC in the task and delete it by selecting the menu command **Delete** or by pressing the key.

The SFC is deleted from the task following a prompt for confirmation.

You can find additional information in the section:

Runtime attributes for runtime groups and SFCs (Page 163)

12.5.3 Runtime attributes for runtime groups and SFCs

Introduction

You can insert SFCs in runtime groups if they have the "reduction ratio" and/or "phase offset" attributes. You set the attributes in the object properties of the runtime group. All charts of the runtime group then have the same "reduction ratio" and "phase offset".

Inserting and editing a runtime group

You can generate a runtime group in the run sequence editor as follows:

1. Select the menu command **Edit > Run Sequence...**

or

click the following button



2. Select the insertion position in the run sequence.
3. Select the menu command **Insert > Runtime Group....**
4. You can enter the name and any desired comment in the "Insert Runtime Group" dialog box. You can also change the default setting of the runtime attributes for the reduction ratio and phase offset here (see information below).
5. Click "OK".

If you want to change the runtime attributes later, select the runtime group in the window of the run sequence and select the menu command **Edit > Object Properties**.

Note

SFCs can run in different cyclic tasks when they are inserted in runtime groups. From a technological point of view, this enables you to structure the project better, which will result in considerably better performance when configuration changes are made (including reducing the time needed for compilation).

Enable attribute of the SFC

The Enable attribute (EN) enables (EN=1) or disables (EN=0) the execution of the SFC. As long as EN is set to 0, the SFC will not be processed regardless of any other conditions.

You can make the enable attribute dynamic. In this case, the value of a block output or the statement of a step decides whether or not the SFC is processed.

Runtime attributes of the runtime group

You can also use the enable attribute to enable and disable a runtime group, same as an SFC.

You cannot assign the following attributes to an SFC directly; they are assigned by inserting them into a runtime group:

- Scan rate
- Phase offset

The SFC receives the attributes set for this runtime group.

An SFC that is not inserted in a runtime group has the following defaults:

- "Reduction ratio = 1"
- "Phase offset = 0"

If you want SFCs in a CPU to run with different runtime attributes, you must insert these charts in different runtime groups.

- **Scan rate**

The reduction ratio specifies whether the SFC is processed each time the task is executed or only every n th time the task is executed. " n " is an integer ($n=2^t$, where $0 \leq t \leq 15$). The increments are a multiple of the basic cycle clock of the task.

Example:

Basic cycle of a cyclic interrupt (OB 33): 500 ms

Possible clock cycles with reduction ratio: 1 s, 2 s, 4 s, 8 s, 16 s, etc.

If a chart is not installed in a runtime group, the number 1 is preset; in other words, the SFC is processed in each run (every 500 ms).

- **Phase offset**

Phase offset allows a uniform distribution of load within the CPU. It must be considered in conjunction with " n ", the reduction ratio. The SFC is executed as often as specified by " n ", in each case offset by " m " units of the task cycle. " m " is an integer, where $0 \leq m \leq (n-1)$.

Example:

Basic cycle of a cyclic interrupt: 500 ms

Reduction ratio: 16 (which means the SFC is processed every 8 s)

phase offset: 3. The SFC is thus processed after 1.5 s; 9.5 s; 17.5 s, etc.

If a chart is not installed in a runtime group, the number 0 is preset; in other words, the SFC is processed without phase offset.

Note

You should only use reduction ratio and phase offset in the tasks that are executed in defined cycles; in other words, in (cyclic) interrupt tasks. In all other tasks you should use extreme caution, particularly in process interrupt tasks and special tasks. Here, you should not change the default reduction ratio=1 and phase offset=0.

Standard interface and chart I/Os

13.1 Standard interface of the SFC

13.1.1 Inputs/outputs of the SFC standard interface

Standard interface of the SFC

The standard interface of the SFC includes the necessary inputs/outputs for controlling the SFC from the user program or by means of operator control and monitoring.

For more information on the I/Os, refer to the following sections:

Standard interface of the SFC (Page 167)

Inputs/outputs of the SFC sorted according to usage (Page 172)

13.1.2 Standard interface of the SFC

Standard interface of the SFC

The following table lists the inputs/outputs of the standard interface for the SFC in alphabetical order.

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	OCM
ABORT	Automatic command "Abort"	BOOL	0	I	
ABORTED	Operating state "Aborted"	BOOL	0	O	
ABORTING	Operating state "Aborting"	BOOL	0	O	
AUT	Change to AUTO (process mode)	BOOL	0	I	
BA_ADDSTATE	BATCH: Additional status word	DWORD	0	O	
BA_EN	BATCH: Enable	BOOL	1	I	+
BA_ID	BATCH: Consecutive batch number	DWORD	0	I	+
BA_NA	BATCH: Batch name	STRING [32]	''	I	+
BA_STATE	BATCH status word	DWORD	0	O	+
BUSY	Status "Busy" (= neither "Idle" nor "Completed" nor "Aborted" nor "Stopped")	BOOL	0	O	
COMPLETE	Automatic command "Complete"	BOOL	0	I	
COMPLETED	Operating state "Completed"	BOOL	0	O	

13.1 Standard interface of the SFC

I/O	Meaning	Data type	Default	Type	OCM
COMPLETING	Operating state "Completing"	BOOL	0	O	
CONT	Continuous mode	BOOL	0	I	
CPU_RESTART	Restart of the CPU	BOOL	0	O	
CUSEQ	No. of current sequencer	BYTE	0	O	+
CUSTEP	No. of current step	WORD	0	O	
CUSTEPACTSTATE	Active state of current step	BOOL	0	O	
CUSTEPCOUNT	Number of active steps	BYTE	0	O	
CUSTEPERRSTATE	Error state of current step	BOOL	0	O	
CUSTEPHELDSTATE	Held state of current step	BOOL	0	O	
CUSTEPMAXRT	Max. run time of the current step	TIME	0	O	
CUSTEPMINRT	Min. run time of the current step	TIME	0	O	
CUSTEPRT	Run time of the current step	TIME	0	O	
CUTRANS	Number of the current transition	WORD	0	O	
CUTRANSCOUNT	Number of active transitions	BYTE	0	O	
CUTRANSRESULT	Result of first active transition	BOOL	0	O	
CYCLEXEC	Cyclic operation (only in AUTO)	BOOL	0 1)	I	
DIS_START_STATE	Reason for start disturbance	DWORD	0	O	+
ENABORT	Enable "Abort"	BOOL	1	I	
ENASTART	Enable "Start in Run"	BOOL	0	I	
ENAUT	Enable "Change to AUTO"	BOOL	0	I	
ENCOMPLETE	Enable "Complete"	BOOL	1	I	
ENHOLD	Enable "Hold"	BOOL	1	I	
ENMAN	Enable "Change to MANUAL"	BOOL	0	I	
ENRESET	Enable "Reset"	BOOL	1	I	
ENRESTART	Enable "Restart"	BOOL	1	I	
ENRESUME	Enable "Resume"	BOOL	1	I	
ENSTART	Enable "Start"	BOOL	1	I	
ENSTOP	Enable "Stop"	BOOL	1	I	
ENTARGETSTEP	Enable "Set target steps"	BOOL	1	I	
ERRG	Group error (EXT_ERR OP_ERR LI_ERR S_ERRG)	BOOL	0	O	
ERROR	Operating state "Error"	BOOL	0	O	
ERROR_COMPLETING	Operating state "Error (Completing)"	BOOL	0	O	
EXEC_ERR	Runtime error, e.g., step error, CPU restart	BOOL	0	O	
EXT_ERR	External error	BOOL	0	I	
HELD	Operating state "Held"	BOOL	0	O	
HELD_ERROR	Operating state "Held (Error)"	BOOL	0	O	
HELDSEQ	No. of held sequencer	BYTE	0	O	+
HELDSTEP	No. of held step	WORD	0	O	
HELDSTEPCOUNT	For future use	BYTE	0	O	
HELDSTEPERRSTATE	For future use	BOOL	0	O	
HELDSTEPHELDSTATE	For future use	BOOL	0	O	

I/O	Meaning	Data type	Default	Type	OCM
HELDSTEPRT	For future use	TIME	0	O	
HELDTRANS	For future use	WORD	0	O	
HELDTRANSCOUNT	For future use	BYTE	0	O	
HELDTRANSRESULT	For future use	BOOL	0	O	
HOLD	Automatic command "Hold"	BOOL	0	I	
HOLDING	Operating state "Holding"	BOOL	0	O	
IDLE	Operating state "Idle"	BOOL	0	O	
INSTROUT	Command Output (only in AUTO)	BOOL	1 2)	I	
INTABORT	Internal command "Abort"	BOOL	0	IO	
INTCOMPLETE	Internal command "Complete"	BOOL	0	IO	
INTERROR	Internal command "Error"	BOOL	0	IO	
INTHOLD	Internal command "Hold"	BOOL	0	IO	
INTONOFF	Internal command "Switch SFC on/off": 0: Switch off, 1: Switch on	BOOL	0	IO	
INTRESET	Internal command "Reset"	BOOL	0	IO	
INTRESTART	Internal command "Restart"	BOOL	0	IO	
INTRESUME	Internal command "Resume"	BOOL	0	IO	
INTSTART	Internal command "Start"	BOOL	0	IO	
INTSTOP	Internal command "Stop"	BOOL	0	IO	
IORES1...6	Reserved for internal use	BOOL	0	IO	
IRES1...5	Reserved for internal use	BOOL	0	I	
LASTSEQ	No. of last active sequencer	BYTE	0	O	
LASTSTEP	No. of last active step	WORD	0	O	
LI_ERR	Parameter assignment error (e.g., invalid combination at inputs)	BOOL	0	O	
LI_ERR_STATE	Status word for parameter error	DWORD	0	O	+
LOCKABORT	External interlock "Abort"	BOOL	0	I	
LOCKCOMPLETE	External interlock "Complete"	BOOL	0	I	
LOCKERROR	External interlock "Error"	BOOL	0	I	
LOCKHOLD	External interlock "Hold"	BOOL	0	I	
LOCKSTOP	External interlock "Stop"	BOOL	0	I	
MAN	Change to MANUAL (operator control mode)	BOOL	0	I	
MODE	Processing mode: 0: Cyclic processing in cyclic interrupt OB 1: One-time processing in startup OB	BOOL	0	I	
MSG_EVID	Message number (ALARM_8P)	DWORD	2	I	
MSG_LOCK	Lock messages: 0: Messages not locked 1: Messages locked	BOOL	0	I	+
MSG_SUP	Message suppression (ALARM_8P)	BOOL	0	O	+
NMSG_EVID	Message number (NOTIFY)	DWORD	1	I	
OCCUPIED	BATCH: "Occupied" ID 0 = not occupied, 1 = occupied	BOOL	0	I	+

13.1 Standard interface of the SFC

I/O	Meaning	Data type	Default	Type	OCM
OP_ERR	Operator error	BOOL	0	O	
OP_ERR_STATE	Status word for operator error	DWORD	0	O	+
ORES_BY	Reserved for internal use	BYTE	0	O	
ORES1...15	Reserved for internal use	BOOL	0	O	
QAUTMAN	Current operating mode: 0: MANUAL, 1: AUTO	BOOL	0 4)	O	
QBA_EN	BATCH: Enable	BOOL	0	O	
QBA_ID	BATCH: Consecutive batch number	DWORD	0	O	
QBA_NA	BATCH: Batch name	STRING [32]	''	O	
QDIS_START	"Start" not possible (e.g. LI_ERR, OP_ERR, ...)	BOOL	0	O	
QENAUT	Enable "Change to AUTO"	BOOL	0	O	
QENMAN	Enable "Change to MANUAL"	BOOL	0	O	
QEXT_ERR	External error	BOOL	0	O	
QFORCEMAN	Changeover to MANUAL without enable (= forced manual, for 1 cycle)	BOOL	0	O	
QOCCUPIED	BATCH: "Occupied" identifier	BOOL	0	O	
QSCC	Step enabling with operator confirmation	BOOL	0	O	
QSCT	Step enabling with transition	BOOL	0	O	
QSCT_TAC	Step enabling with transition and step-specific operator confirmation	BOOL	0	O	
QSCTAC	Step enabling with transition and Confirmation by operator	BOOL	0	O	
QSCTOC	Step enabling with transition or operator confirmation	BOOL	0	O	
QSTEP_NO	BATCH: Step number	DWORD	0	O	
READY_TC	"Ready to complete" state	BOOL	0	O	
RESET	Automatic command "Reset"	BOOL	0	I	
RESTART	Automatic command "Restart"	BOOL	0	I	
RESU_ERROR	Operating state "Resuming (Error)"	BOOL	0	O	
RESUME	Automatic command "Resume"	BOOL	0	I	
RESUMING	Operating state "Resuming"	BOOL	0	O	
RUN	Operating state "Run"	BOOL	0	O	
RUNCOUNT	Number of runs in "cyclic operation"	INT	0	O	
RUNHOLD	Response of the RUN sequencer to the "Hold" command 0: Hold sequencer, 1: Abort sequencer	BOOL	0	I	
S_ERRCA	Confirm all step errors (only in AUTO)	BOOL	0	I	
S_ERRG	Group display "Step error"	BOOL	0	O	
SCT	Step enabling with transition (only in AUTO)	BOOL	1 5)	I	
SCT_TAC	Step enabling with transition and step-specific operator confirmation (only in AUTO)	BOOL	0 5)	I	

I/O	Meaning	Data type	Default	Type	OCM
SELFCOMP	SFC changes automatically from "Run" to "Completing"	BOOL	1	I	
SELFRESET	SFC changes automatically from "Completed/Aborted/Stopped" to "Idle" (only in MANUAL)	BOOL	0	I	
SFC_ADDSTATE	Additional SFC status word	DWORD	0	O	+
SFC_CONTROL	SFC control word (internal interface)	DWORD	0	IO	+
SFC_INIT	Initialize SFC	BOOL	0	IO	
SFC_STATE	SFC status word	DWORD	0	O	+
SIG_2...8	Generate message 2...8 (ALARM_8P)	BOOL	0	IO	
START	Automatic command "Start"	BOOL	0	I	
STARTING	Operating state "Starting"	BOOL	0	O	
STEP_NO	BATCH: Step number	DWORD	0	I	+
STOP	Automatic command "Stop"	BOOL	0	I	
STOPPED	Operating state "Stopped"	BOOL	0	O	
STOPPING	Operating state "Stopping"	BOOL	0	O	
T_OPRQCA	Confirmation of all operator prompts (only in AUTO)	BOOL	0	I	
T_OPRQG	Group display "Operator prompt at transition"	BOOL	0	O	
TARGETSEQ	Sequencer number for "Set target step"	BYTE	0	IO	
TARGETSTEP	Step number for "Set target step"	WORD	0	IO	
TIMEMON	Time monitoring (only in AUTO)	BOOL	0 3)	I	
TRIG_CPU_RESTART	Start trigger for sequencer after CPU restart	BOOL	0	O	

Key	
1)	Depends on "SFC properties/operating parameters AS/cyclic operation"
2)	Depends on "SFC properties/operating parameters AS/Command Output"
3)	Depends on "SFC properties/operating parameters AS/time monitoring"
4)	Depends on "SFC properties/operating parameters AS/operating mode"
5)	Depends on "SFC properties/operating parameters AS/step control mode"

13.1.3 Meanings in the table

Explanations of the table

- In the "I/O" column, text in **bold** print signifies that the attribute S7_visible = 'true' is set. The I/O is visible.
- The following abbreviations are used in the "Type" column
 - I = IN (input)
 - IO = IN_OUT (in/out)
 - O = OUT (output)
- The following abbreviations are used in the "Attr." column
 - "Q" means that attribute S7_link = 'true' is set. The I/O is interconnectable.
 - "M" means that this is a MESSAGE ID for a message block (e.g., ALARM_8P). Parameters cannot be assigned for the I/O; the ID is assigned by the message server.
 - "B" means operator controllable (only via a faceplate). Write access to this I/O is enabled at the OS. Hidden in the CFC.
- In the "OCM" column, "+" means that attribute S7_m_c = 'true' is set. The I/O can be controlled and monitored by the operator.

You can find the meanings of the individual system attributes in "Help on STEP 7" under the topic "System attributes for block I/Os".

13.1.4 Inputs/outputs of the SFC sorted according to usage

Inputs/outputs of SFC

The inputs/outputs of the standard interface for the SFC are sorted according to their use and summarized in the following sections:

Operating modes (SFC inputs/outputs) (Page 195)

Commands and operating states (SFC inputs/outputs) (Page 196)

Execution options (SFC inputs/outputs) (Page 202)

Group displays and group acknowledgements (SFC inputs/outputs) (Page 204)

Data from sequencers and steps to be processed (SFC inputs/outputs) (Page 204)

BATCH parameters (SFC inputs/outputs) (Page 205)

Continuous mode (SFC inputs/outputs) (Page 206)

Troubleshooting (SFC inputs/outputs) (Page 206)

Messages (SFC inputs/outputs) (Page 209)

Control words (SFC inputs/outputs) (Page 210)

Status words (SFC inputs/outputs) (Page 210)

System parameters (SFC inputs/outputs) (Page 214)

Reserves (SFC inputs/outputs) (Page 215)

13.2 Standard interface of the SFC type

13.2.1 Standard interface of the SFC type

Standard interface of the SFC type

The standard interface of the SFC type includes the necessary inputs/outputs for the user program for controlling the SFC instances via the user program or for operator control and monitoring of the SFC instances.

Additional information

You can find additional information about the inputs/outputs in the following sections:

Inputs/outputs of the SFC type standard interface (Page 174)

Inputs/outputs of the SFC type sorted according to usage (Page 180)

"Characteristics" interface parameter assignments (Page 181)

"Characteristics" interface expansions (Page 183)

Changing the control strategy and setpoints for an SFC instance (Page 262)

"Inputs/outputs" interface expansions (Page 181)

13.2.2 Inputs/outputs of the SFC type standard interface

SFC type: Inputs/outputs of the standard interface

The following table lists the inputs/outputs of the standard interface of the SFC type in alphabetical order.

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	OCM
ABORT	Automatic command "Abort"	BOOL	0	I	
ABORTED	Operating state "Aborted"	BOOL	0	O	
ABORTING	Operating state "Aborting"	BOOL	0	O	
AC_AcquireId	Allocation ID for type "Takeover (TakeOver)"	DWORD	0	I	
AC_Priority	Priority	BYTE	0	I	
AC_Request	Type of allocation request For more information, refer to Allocation request and allocation status (Page 303).	BYTE	0	I	
AC_SafeNotMsk	AND_NOT logic operation for allocation change with type "Force (safe) (ForceSafe)"	DWORD	0	I	

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
AC_SafeOrMsk	OR logic operation for allocation change with type "Force (safe) (ForceSafe)"	DWORD	0	I	
AC_WaitTmMax	Maximum wait time [s]	REAL	0.0	I	
ACQ_ID	Allocation ID for type "Takeover (TakeOver)"	DWORD	0	I	
AF_ReqPrev	Previous type of allocation request	BYTE	0	O	
AF_State	Allocation status For more information, refer to Allocation request and allocation status (Page 303).	BYTE	0	O	
AF_TimeStamp	Time stamp of the allocation request (Unix time)	DINT	0	O	
AF_WaitTmAct	Current value of the wait time [s]	REAL	0.0	O	
AUT	Change to AUTO (process mode)	BOOL	0	I	
AUX_PR04...10	Associated value 4..10	ANY	0	IO	
BA_ADDSTATE	BATCH: Additional status word	DWORD	0	O	
BA_CONTROL	BATCH control word (internal interface)	DWORD	0	IO	+
BA_EN	BATCH: Enable	BOOL	1	I	+
BA_ID	BATCH: Consecutive batch number	DWORD	0	I	+
BA_NA	BATCH: Batch name	STRING [32]	' '	I	+
BA_STATE	BATCH status word	DWORD	0	O	+
BUSY	Status "Busy" (= neither "Idle" nor "Completed" nor "Aborted" nor "Stopped")	BOOL	0	O	
COMPLETE	Automatic command "Complete"	BOOL	0	I	
COMPLETED	Operating state "Completed"	BOOL	0	O	
COMPLETING	Operating state "Completing"	BOOL	0	O	
CONT	Continuous mode	BOOL	0	I	
CONT_T	Max. run time for "Continuous mode" [s]	DINT	0	I	
CPU_RESTART	Restart of the CPU	BOOL	0	O	
CS	AUTO: Prepared control strategy (apply at next "Start")	INT	0	I	+
CS_HL	Control strategy "high limit"	INT	0 1)	I	
CS_LL	Control strategy "low limit"	INT	0 1)	I	
CSP_DEFAULT	Initial value of the prepared control strategy	INT	0	I	+
CSP_OP	MANUAL: Prepared control strategy by operator (apply at next "Start")	INT	0	IO	+
CSSPACCEPT	Control strategy+ setpoints applied (1 cycle long)	BOOL	0	O	
CUSEQ	No. of current sequencer	BYTE	0	O	+
CUSTEP	No. of current step	WORD	0	O	
CUSTEPACTSTATE	Active state of current step	BOOL	0	O	
CUSTEPCOUNT	Number of active steps	BYTE	0	O	
CUSTEPERRSTATE	Error state of current step	BOOL	0	O	
CUSTEPHELDSTATE	Held state of current step	BOOL	0	O	
CUSTEPMAXRT	Max. run time of the current step	TIME	0	O	

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
CUSTEPMINRT	Min. run time of the current step	TIME	0	O	
CUSTEPRT	Run time of the current step	TIME	0	O	
CUTRANS	Number of the current transition	WORD	0	O	
CUTRANSCOUNT	Number of active transitions	BYTE	0	O	
CUTRANSRESULT	Result of first active transition	BOOL	0	O	
CYCLEXEC	Cyclic operation (only in AUTO)	BOOL	0 2)	I	
DIS_START_STATE	Reason for start disturbance	DWORD	0	O	+
ENABORT	Enable "Abort"	BOOL	1	I	
ENACQUIRE	1 = Can be allocated with an equipment phase (EPH).	BOOL	1	I	
ENASTART	Enable "Start in Run"	BOOL	0	I	
ENAUT	Enable "Change to AUTO"	BOOL	0	I	
ENCOMPLETE	Enable "Complete"	BOOL	1	I	
ENCSP	Enable for "prepare control strategy"	BOOL	1	I	+
ENFORCEMAN	Enable change to MANUAL if ENMAN = 0	BOOL	1	I	
ENHOLD	Enable "Hold"	BOOL	1	I	
ENMAN	Enable "Change to MANUAL"	BOOL	0	I	
ENRESET	Enable "Reset"	BOOL	1	I	
ENRESTART	Enable "Restart"	BOOL	1	I	
ENRESUME	Enable "Resume"	BOOL	1	I	
ENSTART	Enable "Start"	BOOL	1	I	
ENSTOP	Enable "Stop"	BOOL	1	I	
ENTARGETSTEP	Enable "Set target steps"	BOOL	1	I	
ERRG	Group error (EXT_ERR OP_ERR LI_ERR S_ERRG)	BOOL	0	O	
ERROR	Operating state "Error"	BOOL	0	O	
ERROR_COMPLETING	Operating state "Error (Completing)"	BOOL	0	O	
EXEC_ERR	Runtime error, e.g., step error, CPU restart	BOOL	0	O	
EXT_ERR	External error	BOOL	0	I	
HELD	Operating state "Held"	BOOL	0	O	
HELD_ERROR	Operating state "Held (Error)"	BOOL	0	O	
HELDSEQ	No. of held sequencer	BYTE	0	O	+
HELDSTEP	No. of held step	WORD	0	O	
HELDSTEPCOUNT	For future use	BYTE	0	O	
HELDSTEPERRSTATE	For future use	BOOL	0	O	
HELDSTEPHELDSTATE	For future use	BOOL	0	O	
HELDSTEPRT	For future use	TIME	0	O	
HELDTRANS	For future use	WORD	0	O	
HELDTRANSCOUNT	For future use	BYTE	0	O	
HELDTRANSRESULT	For future use	BOOL	0	O	
HOLD	Automatic command "Hold"	BOOL	0	I	
HOLDING	Operating state "Holding"	BOOL	0	O	

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
IDLE	Operating state "Idle"	BOOL	0	O	
INSTROUT	Command Output (only in AUTO)	BOOL	1 3)	I	
INTABORT	Internal command "Abort"	BOOL	0	IO	
INTCOMPLETE	Internal command "Complete"	BOOL	0	IO	
INTERERROR	Internal command "Error"	BOOL	0	IO	
INTHOLD	Internal command "Hold"	BOOL	0	IO	
INTRESET	Internal command "Reset"	BOOL	0	IO	
INTRESTART	Internal command "Restart"	BOOL	0	IO	
INTRESUME	Internal command "Resume"	BOOL	0	IO	
INTSTART	Internal command "Start"	BOOL	0	IO	
INTSTOP	Internal command "Stop"	BOOL	0	IO	
IORES_BY	Reserved for internal use	BYTE	0	IO	
IORES0...7	Reserved for internal use	BOOL	0	IO	
IRES1...3	Reserved for internal use	BOOL	0	I	
LASTSEQ	No. of last active sequencer	BYTE	0	O	
LASTSTEP	No. of last active step	WORD	0	O	
LI_ERR	Parameter assignment error (e.g., invalid combination at inputs)	BOOL	0	O	
LI_ERR_STATE	Status word for parameter error	DWORD	0	O	+
LOCKABORT	External interlock "Abort"	BOOL	0	I	
LOCKCOMPLETE	External interlock "Complete"	BOOL	0	I	
LOCKERROR	External interlock "Error"	BOOL	0	I	
LOCKHOLD	External interlock "Hold"	BOOL	0	I	
LOCKSTOP	External interlock "Stop"	BOOL	0	I	
MAN	Change to MANUAL (operator control mode)	BOOL	0	I	
MODE	Processing mode: 0: Cyclic processing in cyclic interrupt OB 1: One-time processing in startup OB	BOOL	0	I	
MSG_ACK	Acknowledge messages (ALARM_8P)	WORD	0	O	
MSG_ERR	Message error (ALARM_8P)	BOOL	0	O	
MSG_EVID	Message number (ALARM_8P)	DWORD	0	I	
MSG_LOCK	Lock messages: 0: Messages not locked 1: Messages locked	BOOL	0	I	+
MSG_STAT	Message status (ALARM_8P)	WORD	0	O	
MSG_SUP	Message suppression (ALARM_8P)	BOOL	0	O	+
NMSG_EVID1	Message number 1 (NOTIFY_8P)	DWORD	0	I	
NMSG_EVID2	Message number 2 (NOTIFY_8P)	DWORD	0	I	
NMSG_STAT1	Message status 1 (NOTIFY_8P)	WORD	0	O	
NMSG_STAT2	Message status 2 (NOTIFY_8P)	WORD	0	O	
NSIG_12...16	Generate event message 12...16 (NOTIFY_8P)	BOOL	0	IO	
OCCUPIED	BATCH: "Occupied" identifier	BOOL	0	IO	+

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
OP_ERR	Operator error	BOOL	0	O	
OP_ERR_STATE	Status word for operator error	DWORD	0	O	+
OPTIPNO	Information number for operator	INT	0	O	+
ORES_BY1...3	Reserved for internal use	BYTE	0	O	
ORES1...24	Reserved for internal use	BOOL	0	O	
PARAM	Check control strategy + setpoints	BOOL	0	I	
POSINO	Position text number	INT	0	O	+
QACQFAILED	1 = Allocation request failed (group error)	BOOL	0	O	
QAUTMAN	Current operating mode: 0: MANUAL, 1: AUTO	BOOL	0 5)	O	
QBA_EN	BATCH: Enable	BOOL	0	O	
QBA_ID	BATCH: Consecutive batch number	DWORD	0	O	
QBA_NA	BATCH: Batch name	STRING[32]	''	O	
QCONT	"Continuous mode"	BOOL	0	O	
QCONT_T	Current run time for "Continuous mode" [s]	DINT	0	O	
QCONT_T_ERR	Max. run time for "Continuous mode" exceeded	BOOL	0	O	
QCS	Current control strategy	INT	0	O	+
QCS_0	1: No control strategy	BOOL	0	O	
QCS_1...32	1: Current control strategy 1 - 32	BOOL	0	O	
QCSP	Prepared control strategy	INT	0	O	+
QDIS_START	"Start" not possible (e.g., faulty control strategy and/or setpoints, LI_ERR, OP_ERR, etc.)	BOOL	0	O	
QENAUT	Enable "Change to AUTO"	BOOL	0	O	
QENACQUIRE	1 = Can be allocated with an equipment phase (EPH).	BOOL	0	O	
QENMAN	Enable "Change to MANUAL"	BOOL	0	O	
QEXT_ERR	External error	BOOL	0	O	
QFORCEMAN	Change to MANUAL without enable (= force manual, 1 cycle long)	BOOL	0	O	
QISACQUIRED	1 = Is allocated by an equipment phase (EPH).	BOOL	0	O	
QOCCUPIED	BATCH: "Occupied" identifier	BOOL	0	O	
QPARAM	Control strategy + setpoints were checked and are ok	BOOL	0	O	
QREFRESH	BATCH: Read in setpoints/process values again	BOOL	0	O	
QSCC	Step enabling with operator confirmation	BOOL	0	O	
QSCT	Step enabling with transition	BOOL	0	O	
QSCT_TAC	Step enabling with transition and step-specific op- erator confirmation	BOOL	0	O	
QSCTAC	Step enabling with transition and operator confir- mation	BOOL	0	O	
QSCTOC	Step enabling with transition or operator confir- mation	BOOL	0	O	
QSTEP_NO	BATCH: Step number	DWORD	0	O	

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
QSTEP_T	BATCH: Current step run time [s]	DINT	0	O	+
QTAKESP	Adopt setpoints immediately	BOOL	0	O	
READY_TC	"Ready to complete" state	BOOL	0	O	
REFRESH	BATCH: Read in setpoints/process values again	BOOL	0	IO	
RESET	Automatic command "Reset"	BOOL	0	I	
RESTART	Automatic command "Restart"	BOOL	0	I	
RESU_ERROR	Operating state "Resuming (Error)"	BOOL	0	O	
RESUME	Automatic command "Resume"	BOOL	0	I	
RESUMING	Operating state "Resuming"	BOOL	0	O	
RUN	Operating state "Run"	BOOL	0	O	
RUNCOUNT	Number of runs in "cyclic operation"	INT	0	O	
RUNHOLD	Response of the RUN sequencer to "Hold" 0: Hold sequencer, 1: Abort sequencer	BOOL	0	I	
RUNUPCYC	Number of run-up cycles	INT	3	I	
S_ERRCA	Confirm all step errors (only in AUTO)	BOOL	0	I	
S_ERRG	Group display "Step error"	BOOL	0	O	
SCT	Step enabling with transition (only in AUTO)	BOOL	1 6)	I	
SCT_TAC	Step enabling with transition or step-specific operator confirmation (only in AUTO)	BOOL	0 6)	I	
SELCS	Enable control strategies: Bit 0...31: Control strategy 1...32	DWORD	0 1) 7)	I	+
SELFCOMP	SFC changes automatically from "Run" to "Completing"	BOOL	1	I	
SELFRESET	SFC changes automatically from "Completed/Aborted/Stopped" to "Idle" (only in MANUAL)	BOOL	0	I	
SFC_ADDSTATE	Additional SFC status word	DWORD	0	O	+
SFC_CONTROL	SFC control word (internal interface)	DWORD	0	IO	+
SFC_INIT	Initialize SFC	BOOL	0	IO	
SFC_STATE	SFC status word	DWORD	0	O	+
SIG_2...8	Generate message 2...8 (ALARM_8P)	BOOL	0	IO	
START	Automatic command "Start"	BOOL	0	I	
STARTING	Operating state "Starting"	BOOL	0	O	
STEP_NO	BATCH: Step number	DWORD	0	I	+
STEP_T	BATCH: Max. step run time [s]	DINT	0	I	+
STOP	Automatic command "Stop"	BOOL	0	I	
STOPPED	Operating state "Stopped"	BOOL	0	O	
STOPPING	Operating state "Stopping"	BOOL	0	O	
T_OPRQCA	Confirmation of all operator prompts (only in AUTO)	BOOL	0	I	
T_OPRQG	Group display "Operator prompt at transition"	BOOL	0	O	
TAKESP	Adopt setpoints immediately	BOOL	0	I	
TARGETSEQ	Sequencer number for "Set target step"	BYTE	0	IO	

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
TARGETSTEP	Step number for "Set target step"	WORD	0	IO	
TEC_NAME	Technological name	STRING [64]	0	O	
TIMEMON	Time monitoring (only in AUTO)	BOOL	0 4)	I	
TRIG_CPU_RESTART	Start trigger for sequencer after CPU restart	BOOL	0	O	
USTATUS	Status word in VSTATUS (can be set as required by user)	WORD	0	I	
VSTATUS	Status word	DWORD	0	O	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "SFC properties/operating parameters AS/cyclic operation"
3)	Depends on "SFC properties/operating parameters AS/Command Output"
4)	Depends on "SFC properties/operating parameters AS/time monitoring"
5)	Depends on "SFC properties/operating parameters AS/operating mode"
6)	Depends on "SFC properties/operating parameters AS/step control mode"
7)	Depends on "SFC properties/options/control strategy selection"

13.2.3 Inputs/outputs of the SFC type sorted according to usage

The inputs/outputs of the standard interface for the SFC type are sorted according to their use and summarized in the following sections:

Operating modes (SFC inputs/outputs) (Page 195)

Commands and operating states (SFC inputs/outputs) (Page 196)

Execution options (SFC inputs/outputs) (Page 202)

Group displays and group acknowledgements (SFC inputs/outputs) (Page 204)

Data from sequencers and steps to be processed (SFC inputs/outputs) (Page 204)

BATCH parameters (SFC inputs/outputs) (Page 205)

Continuous mode (SFC inputs/outputs) (Page 206)

Troubleshooting (SFC inputs/outputs) (Page 206)

Messages (SFC inputs/outputs) (Page 209)

Control words (SFC inputs/outputs) (Page 210)

Status words (SFC inputs/outputs) (Page 210)

System parameters (SFC inputs/outputs) (Page 214)

Reserves (SFC inputs/outputs) (Page 215)

13.2.4 "Inputs/outputs" interface expansions

Valid inputs/outputs

The "Inputs/Outputs" interface editor is available for interface expansions. The following data types are permitted for the inputs/outputs in the IN, OUT and IN_OUT sections:

INPUT (IN section):

BOOL, BYTE, CHAR, WORD, DWORD, INT, DINT, REAL, S5TIME, TIME, DATE, TIME_OF_DAY, STRING

OUTPUT (OUT section):

BOOL, BYTE, CHAR, WORD, DWORD, INT, DINT, REAL, S5TIME, TIME, DATE, TIME_OF_DAY, STRING

IN_OUT (IN_OUT section):

BOOL, BYTE, CHAR, WORD, DWORD, INT, DINT, REAL, S5TIME, TIME, DATE, TIME_OF_DAY, STRING

13.2.5 "Characteristics" interface parameter assignments

13.2.5.1 "Characteristics" interface parameter assignments

"Characteristics" interface parameter assignments

Characteristics are only relevant for the SFC type.

The inputs/outputs of the following characteristics are contained in the standard interface:

- "Control strategies" characteristic (Page 181)
- Characteristic for note texts (Page 182)
- Characteristic for position texts (Page 182)

13.2.5.2 "Control strategies" characteristic

"Control strategies" characteristic

You can find explanations for the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	OCM
CS	AUTO: Prepared control strategy (apply at next "Start")	INT	0	I	+
CS_LL	Control strategy "low limit"	INT	0 1)	I	
CS_HL	Control strategy "high limit"	INT	0 1)	I	
CSP_DEFAULT	Initial value of the prepared control strategy	INT	0	I	+

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
CSP_OP	MANUAL: Prepared control strategy by operator (apply at next "Start")	INT	0	IO	+
CSSPACCEPT	Control strategy and setpoints applied (1 cycle long)	BOOL	0	O	
ENCSP	Enable for "prepared control strategy"	BOOL	1	I	+
QCS	Current control strategy	INT	0	O	+
QCS_0	1: No control strategy	BOOL	0	O	
QCS_1...32	1: Current control strategy 1 - 32	BOOL	0	O	
QCSP	Prepared control strategy	INT	0	O	+
SELCS	Enable control strategies: Bit 0..31: Control strategy 1..32	DWORD	0 1) 2)	I	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "SFC properties/options/control strategy selection"

You can find additional information in the section:
Changing the control strategy and setpoints for an SFC instance (Page 262)

13.2.5.3 Characteristic for note texts

Characteristic for note texts

Using the assignment of a note text number in a step action, the corresponding note text is displayed in the SFC faceplate.

You can find explanations for the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	OCM
OPTIPNO	Note text number for operator	INT	0	O	+

13.2.5.4 Characteristic for position texts

Characteristic for position texts

When a position text number is assigned in a step action, the corresponding position text is displayed in the SFC faceplate.

You can find explanations for the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	OCM
POSINO	Position text number	INT	0	O	+

You can find additional information in the section:
How to configure note texts and position texts (Page 103)

13.2.6 "Characteristics" interface expansions

13.2.6.1 "Characteristics" interface expansions

"Characteristics" interface expansion

Characteristics are only relevant for the SFC type.

The inputs/outputs for the following characteristics are not included in the standard interface but can be configured as needed:

"Setpoints" characteristic (Page 183)

"Process values" characteristic (Page 192)

"Control values" characteristic (Page 192)

"Parameters" characteristic (Page 192)

"Bit memory" characteristic (Page 193)

"Timers" characteristic (Page 193)

"Block contacts" characteristic (Page 194)

Note

In the following tables of characteristics, "name" stands for "I/O name". It is not the name of a characteristic.

13.2.6.2 "Setpoints" characteristic

"Setpoints" characteristic

You can find detailed information on I/Os for setpoints in the section:
Meaning and usage of setpoints (Page 189)

You can find explanations for the table entries in the section:
Meanings in the table (Page 172)

"BOOL" data type

I/O	Meaning	Data type	Default	Type	OCM
"name"	Setpoint automatic input	BOOL	0	I	+
"name"_AI	Process value input	BOOL	0	I	+
"name"_AO	Process value output	BOOL	0	O	+

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
"name"_CS	Enable control strategies: Bit 0..31 = control strategy 1..32	DWORD	0 1) 2)	IO	+
"name"_EN0OP	Enable operator control of setpoint on "0"	BOOL	1	IO	+
"name"_EN1OP	Enable operator control of setpoint on "1"	BOOL	1	IO	+
"name"_EN0OPP	Enable operator control of prepared setpoint on "0"	BOOL	1	IO	+
"name"_EN1OPP	Enable operator control of prepared setpoint on "1"	BOOL	1	IO	+
"name"_ERR	Setpoint error	BOOL	0	O	
"name"_OP	Setpoint operator input	BOOL	0	IO	+
"name"_OPP	Prepared setpoint operator input	BOOL	0	IO	+
"name"_Q	Setpoint output	BOOL	0	O	+
"name"_QP	Prepared setpoint output	BOOL	0	O	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "<Setpoint>/Properties/Control strategy assignment"

"INT" data type

I/O	Meaning	Data type	Default	Type	OCM
"name"	Setpoint automatic input	INT	0	I	+
"name"_AI	Process value input	INT	0	I	+
"name"_AO	Process value output	INT	0	O	+
"name"_CS	Enable control strategies: Bit 0..31 = control strategy 1..32	DWORD	0 1) 2)	IO	+
"name"_ENOP	Enable operator control of setpoint	BOOL	1	IO	+
"name"_ENOPP	Enable operator control of prepared setpoint	BOOL	1	IO	+
"name"_ERR	Setpoint error	BOOL	0	O	
"name"_LL	Low limit setpoint	INT	0	I	+
"name"_HL	High limit setpoint	INT	100	I	+
"name"_OP	Setpoint operator input	INT	0	IO	+
"name"_OPP	Prepared setpoint operator input	INT	0	IO	+
"name"_Q	Setpoint output	INT	0	O	+
"name"_QP	Prepared setpoint output	INT	0	O	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "<Setpoint>/Properties/Control strategy assignment"

"DINT", "SOURCE", "DEST" and "VIA" data types

I/O	Meaning	Data type	Default	Type	OCM
"name"	Setpoint automatic input	DINT	0	I	+
"name"_AI	Process value input	DINT	0	I	+
"name"_AO	Process value output	DINT	0	O	+
"name"_CS	Enable control strategies: Bit 0..31 = control strategy 1..32	DWORD	0 1) 2)	IO	+
"name"_ENOP	Enable operator control of setpoint	BOOL	1	IO	+
"name"_ENOPP	Enable operator control of prepared setpoint	BOOL	1	IO	+
"name"_ERR	Setpoint error	BOOL	0	O	
"name"_HL	High limit setpoint	DINT	DINT: 100, SOURCE, DEST, VIA: 2147483647	I	+
"name"_LL	Low limit setpoint	DINT	0	I	+
"name"_OP	Setpoint operator input	DINT	0	IO	+
"name"_OPP	Prepared setpoint operator input	DINT	0	IO	+
"name"_Q	Setpoint output	DINT	0	O	+
"name"_QP	Prepared setpoint output	DINT	0	O	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "<Setpoint>/Properties/Control strategy assignment"

"REAL" data type

I/O	Meaning	Data type	Default	Type	OCM
"name"	Setpoint automatic input	REAL	0.0	I	+
"name"_AI	Process value input	REAL	0.0	I	+
"name"_AO	Process value output	REAL	0.0	O	+
"name"_CS	Enable control strategies: Bit 0..31 = control strategy 1..32	DWORD	0 1) 2)	IO	+
"name"_ENOP	Enable operator control of setpoint	BOOL	1	IO	+
"name"_ENOPP	Enable operator control of prepared setpoint	BOOL	1	IO	+
"name"_ERR	Setpoint error	BOOL	0	O	
"name"_HL	High limit setpoint	REAL	100.0	I	+
"name"_LL	Low limit setpoint	REAL	0.0	I	+
"name"_OP	Setpoint operator input	REAL	0.0	IO	+
"name"_OPP	Prepared setpoint operator input	REAL	0.0	IO	+
"name"_Q	Setpoint output	REAL	0.0	O	+
"name"_QP	Prepared setpoint output	REAL	0.0	O	+

13.2 Standard interface of the SFC type

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "<Setpoint>/Properties/Control strategy assignment"

"STRING" data type

I/O	Meaning	Data type	Default	Type	OCM
"name"	Setpoint automatic input	STRING	"	I	+
"name"_AI	Process value input	STRING	"	I	+
"name"_AO	Process value output	STRING	"	O	+
"name"_CS	Enable control strategies: Bit 0..31 = control strategy 1..32	DWORD	0 1)	IO	+
"name"_ENOP	Enable operator control of setpoint	BOOL	1	O	+
"name"_ENOPP	Enable operator control of prepared setpoint	BOOL	1	O	+
"name"_ERR	Setpoint error	BOOL	0	O	
"name"_OP	Setpoint operator input	STRING	"	O *)	+
"name"_OPP	Prepared setpoint operator input	STRING	"	O *)	+
"name"_Q	Setpoint output	STRING	"	O	+
"name"_QP	Prepared setpoint output	STRING	"	O	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "<Setpoint>/Properties/Control strategy assignment"
*)	The parameters are implemented as outputs because no IN_OUT parameters are permitted with the "STRING" data type!

"PI" data type (process input = parameter for input materials)

I/O	Meaning	Data type	Default	Type	OCM
"name"	Setpoint automatic input	REAL	0.0	I	+
"name"_AI	Process value input	REAL	0.0	I	+
"name"_AO	Process value output	REAL	0.0	O	+
"name"_B	Setpoint tracking ID	DINT	0	I	+
"name"_BAI	Process value input	DINT	0	I	+
"name"_BAO	Process value output	DINT	0	O	+
"name"_BOP	Operator input	DINT	0	IO	+
"name"_BOPP	Prepared operator input	DINT	0	IO	+
"name"_BQ	Setpoint	DINT	0	O	+
"name"_BQP	Prepared setpoint	DINT	0	O	+
"name"_CS	Enable control strategies: Bit 0..31 = control strategy 1..32	DWORD	0 1) 2)	IO	+

I/O	Meaning	Data type	Default	Type	OCM
"name"_ENOP	Enable operator control of setpoint	BOOL	1	IO	+
"name"_ENOPP	Enable operator control of prepared setpoint	BOOL	1	IO	+
"name"_ERR	Setpoint error	BOOL	0	O	
"name"_HL	High limit setpoint	REAL	100.0	I	+
"name"_LL	Low limit setpoint	REAL	0.0	I	+
"name"_M	"Material" setpoint	STRING[16]	"	I	+
"name"_MAI	"Material" process value input	STRING[16]	"	I	+
"name"_MAO	"Material" process value output	STRING[16]	"	O	+
"name"_MOP	"Material" operator input	STRING[16]	"	O	+
"name"_MOPP	Prepared "Material" operator input	STRING[16]	"	O	+
"name"_MQ	"Material" setpoint	STRING[16]	"	O	+
"name"_MQP	Prepared "Material" setpoint	STRING[16]	"	O	+
"name"_OP	Setpoint operator input	REAL	0.0	IO	+
"name"_OPP	Prepared setpoint operator input	REAL	0.0	IO	+
"name"_Q	Setpoint output	REAL	0.0	O	+
"name"_QP	Prepared setpoint output	REAL	0.0	O	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "<Setpoint>/Properties/Control strategy assignment"

"PO" data type (process output = parameter for main products, by-products, and waste products)

I/O	Meaning	Data type	Default	Type	OCM
"name"	Setpoint automatic input	REAL	0.0	I	+
"name"_AI	Process value input	REAL	0.0	I	+
"name"_AO	Process value output	REAL	0.0	O	+
"name"_B	Setpoint tracking ID	DINT	0	I	+
"name"_BAI	Process value input	DINT	0	I	+
"name"_BAO	Process value output	DINT	0	O	+
"name"_BOP	Operator input	DINT	0	IO	+
"name"_BOPP	Prepared operator input	DINT	0	IO	+
"name"_BQ	Setpoint	DINT	0	O	+
"name"_BQP	Prepared setpoint	DINT	0	O	+
"name"_CS	Enable control strategies: Bit0..31 = control strategy 1..32	DWORD	0 1) 2)	IO	+
"name"_ENOP	Enable operator control of setpoint	BOOL	1	IO	+
"name"_ENOPP	Enable operator control of prepared setpoint	BOOL	1	IO	+
"name"_ERR	Setpoint error	BOOL	0	O	
"name"_LL	Low limit setpoint	REAL	0.0	I	+

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
"name"_HL	High limit setpoint	REAL	100.0	I	+
"name"_M	"Material" setpoint	STRING[16]	"	I	+
"name"_MAI	"Material" process value input	STRING[16]	"	I	+
"name"_MAO	"Material" process value output	STRING[16]	"	O	+
"name"_MOP	"Material" operator input	STRING[16]	"	O	+
"name"_MOPP	Prepared "Material" operator input	STRING[16]	"	O	+
"name"_MQ	"Material" setpoint	STRING[16]	"	O	+
"name"_MQP	Prepared "Material" setpoint	STRING[16]	"	O	+
"name"_OP	Setpoint operator input	REAL	0.0	IO	+
"name"_OPP	Prepared setpoint operator input	REAL	0.0	IO	+
"name"_Q	Setpoint output	REAL	0.0	O	+
"name"_QP	Prepared setpoint output	REAL	0.0	O	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "<Setpoint>/Properties/Control strategy assignment"

"TKEY" data type

I/O	Meaning	Data type	Default	Type	OCM
"name"	Setpoint automatic input	DINT	0	I	+
"name"_AI	Process value input	DINT	0	I	+
"name"_AO	Process value output	DINT	0	O	+
"name"_CS	Enable control strategies: Bit 0..31 = control strategy 1..32	DWORD	0 1) 2)	IO	+
"name"_ENOP	Enable operator control of setpoint	BOOL	1	IO	+
"name"_ENOPP	Enable operator control of prepared setpoint	BOOL	1	IO	+
"name"_ERR	Setpoint error	BOOL	0	O	
"name"_LL	Low limit setpoint	DINT	0	I	+
"name"_HL	High limit setpoint	DINT	DINT: 100, SOURCE, DEST, VIA: 2147483647	I	+
"name"_OP	Setpoint operator input	DINT	0	IO	+
"name"_OPP	Prepared setpoint operator input	DINT	0	IO	+
"name"_Q	Setpoint output	DINT	0	O	+
"name"_QP	Prepared setpoint output	DINT	0	O	+
"name"_ID	"Transfer ID" automatic input	STRING[16]	"	I	+
"name"_IDAI	"Transfer ID" process value input	STRING[16]	"	I	+
"name"_IDAO	"Transfer ID" process value output	STRING[16]	"	O	+
"name"_IDQ	"Transfer ID" setpoint output	STRING[16]	"	O	+

I/O	Meaning	Data type	Default	Type	OCM
"name"_IDQP	"Transfer ID" prepared setpoint output	STRING[16]	"	O	+
"name"_IDOP	Setpoint operator input	STRING[16]	"	O	+
"name"_IDOPP	Prepared setpoint operator input	STRING[16]	"	O	+
"name"_C	"Transfer COUNTER" automatic input	DINT	0	I	+
"name"_CAI	"Transfer COUNTER" process value input	DINT	0	I	+
"name"_CAO	"Transfer COUNTER" process value output	DINT	0	O	+
"name"_CQ	"Transfer COUNTER" setpoint output	DINT	0	O	+
"name"_CQP	"Transfer COUNTER" prepared setpoint output	DINT	0	O	+
"name"_COP	Setpoint operator input	DINT	0	IO	+
"name"_COPP	Prepared setpoint operator input	DINT	0	IO	+

Key	
1)	Depends on the number of control strategies in the characteristics editor
2)	Depends on "<Setpoint>/Properties/Control strategy assignment"

13.2.6.3 Meaning and usage of setpoints

Inputs/outputs of setpoints

The following explanations apply to all setpoints regardless of their data type. Specific explanations are indicated as such.

Process values

The process value input is used to read out the actual process value from the plant. The process value can be a value from the process or a calculated value. The value is output unmodified at the output.

Data type	I/O	Meaning
	Input	
all	..._AI	Actual Value Input
PI	..._BAI	Tracking ID Actual Value Input
PI	..._MAI	Material Actual Value Input
TKEY	..._IDAI	Transfer ID Actual Value Input
TKEY	..._CAI	Transfer Counter Actual Value Input
	Output	
all	..._AO	Actual Value Output
PI	..._BAO	Tracking ID Actual Value Output

13.2 Standard interface of the SFC type

Data type	I/O	Meaning
PI	..._MAO	Material Actual Value Output
TKEY	..._IDAO	Transfer ID Actual Value Output
TKEY	..._CAO	Transfer Counter Actual Value Output

Prepared setpoints

At the setpoint inputs for AUTO or MANUAL mode, the word that will be used when the SFC instance is next started is "prepared".

You can assign parameters for the MANUAL mode inputs in the "Prepared values" view of the SFC faceplate. The prepared values are sent to the outputs.

Data type	I/O	Meaning
	Input	
all	...	Setpoint Automatic Input
all	..._OPP	Setpoint Operator Input Prepare
PI, PO	..._BOPP	Tracking ID Setpoint Operator Input Prepare
PI, PO	..._B	Tracking ID Setpoint Automatic Input
PI, PO	..._M	Material Setpoint Automatic Input
PI, PO	..._MOPP	Material Setpoint Operator Input Prepare
TKEY	..._ID	Transfer ID Setpoint Automatic Input
TKEY	..._IDOPP	Transfer ID Setpoint Operator Input Prepare
TKEY	..._C	Transfer Counter Setpoint Automatic Input
TKEY	..._COPP	Transfer Counter Setpoint Operator Input Prepare
	Output	
all	..._QP	Valid Setpoint Prepare
PI, PO	..._BQP	Tracking ID Valid Setpoint Prepare
PI, PO	..._MQP	Material Valid Setpoint Prepare
TKEY	..._IDQP	Transfer ID Valid Setpoint Prepare
TKEY	..._CQP	Transfer Counter Valid Setpoint Prepare

Setpoints that will be applied immediately

A value that will be applied immediately is set on the setpoint inputs for AUTO or MANUAL mode. In AUTO mode, the TAKESP input must also be set.

The inputs of the operating mode MANUAL can be parameterized in the "Current values" view of the SFC faceplate. The current values are sent to the outputs.

Data type	I/O	Meaning
	Input	
all	...	Setpoint Automatic Input
all	..._OP	Setpoint Operator Input
PI, PO	..._B	Tracking ID Setpoint Automatic Input
PI, PO	..._BOP	Tracking ID Setpoint Operator Input
PI, PO	..._M	Material Setpoint Automatic Input
PI, PO	..._MOP	Material Setpoint Operator Input
TKEY	..._ID	Transfer ID Setpoint Automatic Input

Data type	I/O	Meaning
TKEY	..._IDOP	Transfer ID Setpoint Operator Input
TKEY	..._C	Transfer Counter Setpoint Automatic Input
TKEY	..._COP	Transfer Counter Setpoint Operator Input
	Output	
all	..._Q	Active Setpoint
PI, PO	..._BQ	Tracking ID Active Setpoint
PI, PO	..._MQ	Material Active Setpoint
TKEY	..._IDQ	Transfer ID Active Setpoint
TKEY	..._CQ	Transfer Counter Active Setpoint

Operator enables (preparation and modification of setpoints in MANUAL)

The preparation or modification of the setpoint in MANUAL mode is only possible if the corresponding operator enables are set.

Data type	I/O	Meaning
	Input	
BOOL	..._EN0OP	Enable setpoint "0" operator input
BOOL	..._EN1OP	Enable setpoint "1" operator input
BOOL	..._EN0OPP	Enable setpoint "0" operator input prepare
BOOL	..._EN1OPP	Enable setpoint "1" operator input prepare
all except BOOL	..._ENOP	Enable Setpoint Operator Input
all except BOOL	..._ENOPP	Enable Setpoint Operator Input Prepare

Checking limit values

Modified limit values will only be applied at the appropriate output if the new value is within the defined limits of the setpoint. If this is not the case, the limit value violation is indicated as an error at the <sw>_ERR output and the output affected remains unchanged.

Data type	I/O	Meaning
	Input	
all except BOOL, STRING	..._LL	Low Limit
all except BOOL, STRING	..._HL	High Limit
	Output	
all except BOOL, STRING	..._ERR	Setpoint Input Error

Checking control strategy assignment

The value of the <sp>_CS input defines the assignment of the setpoint to the existing control strategies. This assignment is necessary for checking and applying setpoint changes and for adapting the SFC faceplate to the currently selected control strategy. Assign parameters for the input <sw>_CS in the "Object Properties" dialog for a setpoint.

13.2.6.4 "Process values" characteristic

"Process values" characteristic

I/O	Meaning	Data type	Default	Type	OCM
"name"	Process value input	BOOL	0	I	
		INT	0		
		DINT	0		
		REAL	0.0		
		STRING	"		

You can find explanations for the table entries in the section: Meanings in the table (Page 172)

13.2.6.5 "Control values" characteristic

"Control values" characteristic

I/O	Meaning	Data type	Default	Type	OCM
"name"	Control value output	BOOL	0	O	
		INT	0		
		DINT	0		
		REAL	0.0		
		STRING	"		

You can find explanations for the table entries in the section: Meanings in the table (Page 172)

13.2.6.6 "Parameters" characteristic

"Parameters" characteristic

I/O	Meaning	Data type	Default	Type	OCM
"name"	Parameter input	BOOL	0	I	
		INT	0		
		DINT	0		
		REAL	0.0		
		STRING	"		

You can find explanations for the table entries in the section: Meanings in the table (Page 172)

Note

The input parameters of the "Parameters" characteristic can be used for a direct access to the own SFC type or for an indirect access to a lower-level SFC type. Therefore, these parameters are not assigned the S7_contact attribute by default. If a parameter from the "Parameters" characteristic is to be used by a higher-level EPH or EM, the S7_contact attribute must be set manually.

13.2.6.7 "Bit memory" characteristic

"Bit memory" characteristic

I/O	Meaning	Data type	Default	Type	OCM
"name"	Bit memory (static VAR area)	BOOL INT DINT REAL STRING	0 0 0 0.0 "		

Note

Bit memory is an internal variable that does not appear as a visible I/O on the interface. As a result, for example, you cannot use the "Browse" dialog box or the interface editor to incorporate this variable in the addresses of the step/transition properties. You have to enter the I/O name for the bit memory with the keyboard.

13.2.6.8 "Timers" characteristic

"Timers" characteristic

I/O	Meaning	Data type	Default	Type	OCM
SAMPLE_T	Task sampling time in sec (preset to the relevant cycle time by the code generator)	REAL	0.0	I	
"name"_MODE	Operating mode: 0: Start timer as pulse "Pulse" 1: Start timer as extended pulse "ExtP" 2: Start timer with on delay "OnDel" 3: Start timer with retentive on delay "RetOn-D" 4: Start timer with off delay "Off-D"	INT	2	I	

13.2 Standard interface of the SFC type

I/O	Meaning	Data type	Default	Type	OCM
"name"_TIME0	Time in s	REAL	1.0	IO	
"name"_RESET	Reset	BOOL	0	IO	
"name"_I0	Input pulse	BOOL	0	IO	
"name"_QERR	Error	BOOL	0	O	
"name"_Q0	Output pulse	BOOL	0	O	
"name"_PTIME	Time remaining in s	REAL	0.0	O	

You can find explanations for the table entries in the section: Meanings in the table (Page 172)

13.2.6.9 "Block contacts" characteristic

"Block contacts" characteristic

You can find information on this in the section: Block contacts (Page 105).

13.3 SFC inputs/outputs sorted by usage

13.3.1 Operating modes (SFC inputs/outputs)

SFC inputs/outputs of the operating modes

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	O&M
ENAUT	Enable "Change to AUTO"	BOOL	0	I	
QENAUT	Enable "Change to AUTO"	BOOL	0	O	
AUT	Change to AUTO (process mode)	BOOL	0	I	
Operator and display section "AUTO"	Change to AUTO (process mode) with button				
ENMAN	Enable "Change to MANUAL"	BOOL	0	I	
QENMAN	Enable "Change to MANUAL"	BOOL	0	O	
MAN	Change to MANUAL (operator control mode)	BOOL	0	I	
Operator and display section "MANUAL"	Change to MANUAL (operator control mode) with button				
QAUTMAN	Current operating mode: 0: MANUAL, 1: AUTO	BOOL	0 *)	O	
QFORCEMAN	Change to MANUAL without enable (= force manual, for 1 cycle)	BOOL	0	O	

Key	
*)	Depends on "SFC properties/operating parameters AS/operating mode"

Switching the operating mode in the interface

Previous operating mode: 0 = MANUAL 1 = AUTO	Enable MANUAL		Enable AUTO		Changeover -> MAN -> AUTO		Next operating mode: 0 = MANUAL 1 = AUTO
QAUTMAN	ENMAN	QENMAN	ENAUT	QENAUT	MAN	AUT	QAUTMAN
0	x	x	0 <-> 1	0 <-> 1	0	0	0
0	x	x	0	0	0	1	0
0	x	x	1	1	0	1	0 -> 1
1	0 <-> 1	0 <-> 1	x	x	0	0	1
1	0	0	x	x	1	0	1
1	1	1	x	x	1	0	1 -> 0

Switching the operating mode via operator input (SFC test mode/SFC visualization)

Previous operating mode	Enable MANUAL/AUTO	Changeover -> MAN -> AUTO		Next operating mode	
Display	Display + "Lock" button	MANUAL button	AUTO button	Display	
MANUAL	Closed + raised	Inactive	Click	After AUTO warning box	
MANUAL	Open + depressed	Inactive	Click	AUTO	
AUTO	Closed + raised	Click	Inactive	After MANUAL warning box	
AUTO	Open + depressed	Click	Inactive	MANUAL	

The "padlock" enable button also shows the status of QENMAN or QENAUT.

The QFORCEMAN output is set for one cycle when there is a changeover from AUTO to MAN if no enable was set.

13.3.2 Commands and operating states (SFC inputs/outputs)**SFC inputs/outputs of the commands and operating states**

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

"Start" (enables, commands, new operating state)

I/O	Meaning	Data type	Default	Attr.
ENSTART	Enable "Start"	BOOL	1	Q
ENASTART	Enable "Start in Run"	BOOL	0	Q
QDIS_START	"Start" not possible (e.g. LI_ERR, OP_ERR, ...)	BOOL	0	
START	Automatic command "Start"	BOOL	0	Q
"Start" control and display section	"Start" button for manual command			
INTSTART	Internal command "Start"	BOOL	0	
STARTING	New operating state "Starting" (transitional state)	BOOL	0	
RUN	New operating state "Run" (after "Starting")	BOOL	0	

"Hold" (enables, commands, new operating state)

I/O	Meaning	Data type	Default	Type
ENHOLD	Enable "Hold"	BOOL	1	I
HOLD	Automatic command "Hold"	BOOL	0	I

I/O	Meaning	Data type	Default	Type
"Hold" control and display section	"Hold" button for manual command			
INTHOLD	Internal command "Hold"	BOOL	0	IO
LOCKHOLD	External interlock "Hold"	BOOL	0	I
HOLDING	New operating state "Holding" (transitional state)	BOOL	0	O
HELD	New operating state "Held" (after "Holding")	BOOL	0	O

"Resume" (enables, commands)

I/O	Meaning	Data type	Default	Type
ENRESUME	Enable "Resume"	BOOL	1	I
RESUME	Automatic command "Resume"	BOOL	0	I
"Resume" control and display section	"Resume" button for manual command			
INTRESUME	Internal command "Resume"	BOOL	0	IO

"Resume" in "Held" operating state (new operating state)

I/O	Meaning	Data type	Default	Type
RESUMING	New operating state "Resuming" (transitional state)	BOOL	0	O
RUN	New operating state "Run" (after "Resuming")	BOOL	0	O

"Resume" in "Held (Error)" operating state (new operating state)

I/O	Meaning	Data type	Default	Type
RESU_ERROR	New operating state "Resuming (Error)" (transitional state)	BOOL	0	O
RUN	New operating state "Run" (after "Resuming (Error)")	BOOL	0	O

"Abort" (enables, commands, new operating state)

I/O	Meaning	Data type	Default	Type
ENABORT	Enable "Abort"	BOOL	1	I
ABORT	Automatic command "Abort"	BOOL	0	I
"Abort" control and display section	"Abort" button for manual command			

13.3 SFC inputs/outputs sorted by usage

I/O	Meaning	Data type	Default	Type
INTABORT	Internal command "Abort"	BOOL	0	IO
LOCKABORT	External interlock "Abort"	BOOL	0	I
ABORTING	New operating state "Aborting" (transitional state)	BOOL	0	O
ABORTED	New operating state "Aborted" (after "Aborting")	BOOL	0	O

"Complete" (enables, commands, new operating state)

I/O	Meaning	Data type	Default	Type
ENCOMPLETE	Enable "Complete"	BOOL	1	I
COMPLETE	Automatic command "Complete"	BOOL	0	I
"Complete" control and display section	"Complete" button for manual command			
INTCOMPLETE	Internal command "Complete"	BOOL	0	IO
LOCKCOMPLETE	External interlock "Complete"	BOOL	0	I
SELFCOMP	SFC changes automatically from "Run" to "Completing"	BOOL	1	I
COMPLETING	New operating state "Completing" (transitional state)	BOOL	0	O
COMPLETED	New operating state "Completed" (after "Completing")	BOOL	0	O

"Stop" (enables, commands, new operating state)

I/O	Meaning	Data type	Default	Type
ENSTOP	Enable "Stop"	BOOL	1	I
STOP	Automatic command "Stop"	BOOL	0	I
"Stop" control and display section	"Stop" button for manual command			
INTSTOP	Internal command "Stop"	BOOL	0	IO
LOCKSTOP	External interlock "Stop"	BOOL	0	I
STOPPING	New operating state "Stopping" (transitional state)	BOOL	0	O
STOPPED	New operating state "Stopped" (after "Stopping")	BOOL	0	O

"Restart" (enables, commands, new operating state)

I/O	Meaning	Data type	Default	Type
ENRESTART	Enable "Restart"	BOOL	1	I
RESTART	Automatic command "Restart"	BOOL	0	I

I/O	Meaning	Data type	Default	Type
"Restart" control and display section	"Restart" button for manual command			
INTRESTART	Internal command "Restart"	BOOL	0	IO
STARTING	New operating state "Starting" (transitional state)	BOOL	0	O
RUN	New operating state "Run" (after "Held/Held (Error)")	BOOL	0	O

"Reset" (enables, commands, new operating state)

I/O	Meaning	Data type	Default	Type
ENRESET	Enable "Reset"	BOOL	1	I
RESET	Automatic command "Reset"	BOOL	0	I
"Reset" control and display section	"Reset" button for manual command			
INTRESET	Internal command "Reset"	BOOL	0	IO
SELFRESET	SFC changes automatically from "Completed/Aborted/Stopped" to "Idle" (only in MANUAL)	BOOL	0	I
IDLE	New operating state "Idle"	BOOL	0	O

"Error" (enables, commands)

I/O	Meaning	Data type	Default	Type
"Error" control and display section	"Error" button for manual command			
INTERROR	Internal command "Error"	BOOL	0	IO
LOCKERROR	External interlock "Error"	BOOL	0	I

"Error" not in operating state "Completing" (new operating state)

I/O	Meaning	Data type	Default	Type
ERROR	New operating state "Error" (transitional state)	BOOL	0	O
HELD_ERROR	New operating state "Held (Error)" (after "Error")	BOOL	0	O

"Error" in operating state "Completing" (new operating state)

I/O	Meaning	Data type	Default	Type
ERROR_COMPLETING	New operating state "Error (Completing)" (transitional state)	BOOL	0	O
COMPLETING	New operating state "Completing" after "Error (Completing)" (transitional state)	BOOL	0	O

Explanations of the commands and operating state changes

The "Start,... " MANUAL commands are entered from the control and display section by means of the buttons and control word SFC_CONTROL and reset after they have been executed.

Whether or not the buttons are enabled is derived from the status word SFC_STATE and depends on the particular "EN..." enable, the operating mode, and the operating state:

- In AUTO mode, all buttons for the commands are disabled.
- In MANUAL mode, the buttons for the commands are enabled or disabled depending on the enable and the operating state.

The command inputs "START," are evaluated in AUTO mode depending on the "ENSTART," enable. Whether they are permitted depends on the current operating state.

You can find additional information about this in the section: Operating states (Page 238).

The LOCK commands are evaluated in AUTO and MANUAL modes regardless of the enable. Whether they are permitted depends on the current operating state.

The INT commands are evaluated in AUTO and MANUAL modes regardless of the enable. They are used in step actions and reset after they have been executed. Whether they are permitted depends on the current operating state.

Note the special conditions for the "Start" command:

QDIS_START

"Start" is possible only when QDIS_START is set to 0 (in IDLE, RUN, COMPLETED, ABORTED).

You can find additional information in the section: Troubleshooting (SFC inputs/outputs) (Page 206)

ENASTART

In RUN mode, "Start" is possible only when ENASTART is set to 1 and the following conditions are also met:

AUTO mode: CONT = 1 (Continuous mode) and

READY_TC = 1 (RUN sequencer is completed).

or after action "READY_TC:= 1")

You can find additional information in the section: Start requirements of an SFC (chart/instance) (Page 254)

Special INT command (for SFC only)

I/O	Meaning	Data type	Default	Type
INTONOFF	Internal command "Switch SFC on/off", 1: Switch on, 0: Switch off	BOOL	0	IO

This command exists to allow compatibility with SFC V5.2 (<SFCname>.EN) and is used in step actions.

Note

The I/O INTONOFF is not an available **SFC type** in V6.1 and higher. It has been changed to a reserve (IORES0). Existing accesses to INTONOFF are converted to textual interconnections.

Enable SFC (INTONOFF=1)

I/O	Meaning	Data type	Default	Type
ENSTART	Enable "Start"	BOOL	1	I
STARTING	New operating state "Starting" (transitional state)	BOOL	0	O
RUN	New operating state "Run" (after "Starting")	BOOL	0	O

Disable SFC (INTONOFF=0)

I/O	Meaning	Data type	Default	Type
ENABORT	Enable "Abort"	BOOL	1	I
ABORTING	New operating state "Aborting" (transitional state)	BOOL	0	O
ABORTED	New operating state "Aborted" (after "Aborting")	BOOL	0	O

Special states

I/O	Meaning	Data type	Default	Type
READY_TC	"Ready to complete" state	BOOL	0	O
BUSY	Status "Busy" (= neither "Idle" nor "Completed" nor "Aborted" nor "Stopped")	BOOL	0	O

READY_TC

With the READY_TC (Ready to complete) output of the standard interface, the SFC provides a signal for synchronization between the SFC and the higher-level controller for the transition from "Run" to "Completing." The SFC sets the READY_TC to "1" if "Run" has finished executing, in other words, the start conditions for a sequencer are not fulfilled or a started sequencer has been executed completely. Regardless of the reason, READY_TC can also be set in increments. This enables the actual SFC execution completion to be signaled. A higher-level controller issues the "Complete" command to the SFC only when the READY_TC output is set.

If "Run" operating state is exited (for example, due to a "Complete", "Hold," or "Error") or an active change in the control strategy takes place, READY_TC = 0 is set. If the SFC has been

placed in "Hold" state and then resumed, READY_TC is not automatically reset to the previous value; rather, it remains set to 0 because the SFC cannot decide whether the conditions for READY_TC = 1 have been fulfilled again when a resumption occurs. The project engineer is responsible for proper handling of READY_TC in this case.

BUSY

BUSY = 1 indicates that the SFC is "busy," i.e., has been started. This signal replaces the signal <SFCname.EN> used for transitions in SFC V5.2 and lower.

Note

If several commands are queued simultaneously (for example, external and internal commands), an interconnection error (LI_ERR=1) is usually displayed. The commands will not be executed or only one of the commands will be executed.

Programmed targets steps / "Set target step" enable

I/O	Meaning	Data type	Default	Type
ENTARGETSTEP	Enable "Set target steps"	BOOL	1	I
TARGETSEQ	Sequencer number for "Set target step"	BYTE	0	IO
TARGETSTEP	Step number for "Set target step"	WORD	0	IO

With ENTARGETSTEP, you can enable or disable the manual setting of target steps in MANUAL mode (relevant only in MANUAL, in contrast to other "EN..."). ENTARGETSTEP has no effect on programmed target steps.

With TARGETSEQ and TARGETSTEP, only one target step can be set in each sequencer. Target steps already set are reset.

See also

Behavior when multiple commands are queued (Page 255)

13.3.3 Execution options (SFC inputs/outputs)

SFC inputs/outputs of the execution options

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type
CYCLEEXEC	Cyclic operation (only in AUTO)	BOOL	0 2)	I
"Cyclic Operation" control and display section	Option: Cyclic Operation (only in MANUAL)		0 2)	
RUNCOUNT	Number of runs in "cyclic operation"	INT	0	O

13.3 SFC inputs/outputs sorted by usage

I/O	Meaning	Data type	Default	Type
INSTROUT	Command Output (only in AUTO)	BOOL	1 3)	I
"Command Output" control and display section	Option: Command Output (only in MANUAL)		1 3)	
TIMEMON	Time monitoring (only in AUTO)	BOOL	0 4)	I
"Time Monitoring" control and display section	Option: Time Monitoring (only in MANUAL)		0 4)	
SCT	Step enabling with transition (only in AUTO)	BOOL	1 6)	I
SCT_TAC	Step enabling with transition and step-specific operator confirmation (only in AUTO)	BOOL	0 6)	I
"Step control mode" control and display section	MAN T:: Step enabling with transition T or O: Step enabling with transition or operator confirmation T and O: Step enabling with transition and operator confirmation O: Step enabling with operator confirmation T/T and O: Step enabling with transition and step-specific Confirmation by operator		T 6)	
QSCC	Step enabling with operator confirmation	BOOL	0	O
QSCT	Step enabling with transition	BOOL	0	O
QSCT_TAC	Step enabling with transition and step-specific operator confirmation	BOOL	0	O
QSCTAC	Step enabling with transition and operator confirmation	BOOL	0	O
QSCTOC	Step enabling with transition or operator confirmation	BOOL	0	O
RUNHOLD	Response of the RUN sequencer to the "Hold" Command 0: Hold sequencer, 1: Abort sequencer	BOOL	0	I
SELFCOMP	SFC changes automatically from "Run" to "Completing"	BOOL	1	I
SELFRESET	SFC changes automatically from "Completed/Aborted/Stopped" to "Idle" (only in MANUAL)	BOOL	0	I

Key	
2)	Depends on "SFC properties/operating parameters AS/cyclic operation"
3)	Depends on "SFC properties/operating parameters AS/Command Output"
4)	Depends on "SFC properties/operating parameters AS/time monitoring"
6)	Depends on "SFC properties/operating parameters AS/step control mode"

13.3.4 Group displays and group acknowledgements (SFC inputs/outputs)

SFC inputs/outputs of the group displays and group acknowledgements

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type
S_ERRCA	Confirm all step errors (only in AUTO)	BOOL	0	I
S_ERRG	Group display "Step error"	BOOL	0	O
T_OPRQCA	Confirmation of all operator prompts (only in AUTO)	BOOL	0	I
T_OPRQG	Group display "Operator prompt at transition"	BOOL	0	O
"Confirm all" control and display section	Confirm all step errors and all operator prompts with button			

13.3.5 Data from sequencers and steps to be processed (SFC inputs/outputs)

SFC inputs/outputs of the data from sequencers and steps to be processed

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	OCM
CUSEQ	No. of current sequencer	BYTE	0	O	+
CUSTEP	No. of current step	WORD	0	O	
CUSTEPACTSTATE	Active state of current step	BOOL	0	O	
CUSTEPCOUNT	Number of active steps	BYTE	0	O	
CUSTEPERRSTATE	Error state of current step	BOOL	0	O	
CUSTEPHELDSTATE	Held state of current step	BOOL	0	O	
CUSTEPMAXRT	Max. run time of the current step	TIME	0	O	
CUSTEPMINRT	Min. run time of the current step	TIME	0	O	
CUSTEPRT	Run time of the current step	TIME	0	O	
CUTRANS	Number of the current transition	WORD	0	O	
CUTRANSCOUNT	Number of active transitions	BYTE	0	O	
CUTRANSRESULT	Result of first active transition	BOOL	0	O	
HELDSEQ	No. of held sequencer	BYTE	0	O	+
HELDSTEP	No. of held step	WORD	0	O	
HELDSTEPCOUNT	For future use	BYTE	0	O	
HELDSTEPERRSTATE	For future use	BOOL	0	O	
HELDSTEPHELDSTATE	For future use	BOOL	0	O	
HELDSTEPRT	For future use	TIME	0	O	
HELDTRANS	For future use	WORD	0	O	
HELDTRANSCOUNT	For future use	BYTE	0	O	

I/O	Meaning	Data type	Default	Type	OCM
HELDTRANSRESULT	For future use	BOOL	0	O	
LASTSEQ	No. of last active sequencer	BYTE	0	O	
LASTSTEP	No. of last active step	WORD	0	O	

Note

For step or transition related values (CUSTEPx, CUTTRANSx, HELDSTEPx, HELDTRANSx, LASTSTEP) only the value of the first step or the first active transition will be indicated.

The number of active steps/transitions is indicated by xCOUNT.

13.3.6 BATCH parameters (SFC inputs/outputs)

SFC inputs/outputs of the BATCH parameters

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	O&M
BA_CONTROL *)	BATCH control word (internal interface)	DWORD	0	IO	+
BA_EN	BATCH: Enable	BOOL	1	I	+
BA_ID	BATCH: Consecutive batch number	DWORD	0	I	+
BA_NA	BATCH: Batch name	STRING[32]	"	I	+
BA_STATE	BATCH status word	DWORD	0	O	+
OCCUPIED	BATCH: "Occupied" identifier	BOOL	0	I *) IO..**)	+
QBA_EN	BATCH: Enable	BOOL	0	O	
QBA_ID	BATCH: Consecutive batch number	DWORD	0	O	
QBA_NA	BATCH: Batch name	STRING[32]	"	O	
QOCCUPIED	BATCH: "Occupied" identifier	BOOL	0	O	
QREFRESH *)	BATCH: Read in setpoints/process values again	BOOL	0	O	
QSTEP_NO	BATCH: Step number	DWORD	0	O	
QSTEP_T *)	BATCH: Current step run time [s]	DINT	0	O	+
REFRESH *)	BATCH: Read in setpoints/process values again	BOOL	0	IO	
STEP_NO	BATCH: Step number	DWORD	0	I	+
STEP_T *)	BATCH: Max. step run time [s]	DINT	0	I	+

13.3 SFC inputs/outputs sorted by usage

Key	
*)	Only for SFC type
**)	Only for SFC

13.3.7 Continuous mode (SFC inputs/outputs)

SFC connections for continuous mode

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type
CONT	"Continuous mode"	BOOL	0	I
CONT_T *)	Max. run time for "Continuous mode" [s]	DINT	0	I
QCONT *)	"Continuous mode"	BOOL	0	O
QCONT_T *)	Current run time for "Continuous mode" [s]	DINT	0	O
QCONT_T_ERR *)	Max. run time for "Continuous mode" exceeded	BOOL	0	O
ENASTART	Enable "Start in Run"	BOOL	0	I

Key	
*)	Only for SFC type

For an explanation of the cooperation between user controls and SFCs, refer to the section: Continuous mode (Page 243)

13.3.8 Troubleshooting (SFC inputs/outputs)

SFC inputs/outputs of the troubleshooting

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type
ERRG	Group error (EXT_ERR OP_ERR LI_ERR S_ERRG)	BOOL	0	O
EXEC_ERR	Runtime error, e.g., step error, CPU restart	BOOL	0	O
EXT_ERR	External error	BOOL	0	I
LI_ERR	Parameter assignment error (e.g. invalid combination at inputs)	BOOL	0	O
OP_ERR	Operator error	BOOL	0	O

I/O	Meaning	Data type	Default	Type
QEXT_ERR	External error (= EXT_ERR)	BOOL	0	0
QDIS_START	"Start" not possible (e.g. LI_ERR, OP_ERR, ...)	BOOL	0	0

Calculation of LI_ERR and LI_ERR_STATE

AUTO

- Control strategy (SFC type only)
The control strategy is checked when QPARAM = 1:
Bit 0 - if control strategy high limit is violated ($CS > CS_HL$)
Bit 0 - if control strategy low limit is violated ($CS < CS_LL$)
Bit 0 - if control strategies exist ($CS_LL > 0$) and the prepared control strategy (CS) is not enabled (SELCS)
- Setpoint (SFC type only)
The setpoint is checked when QPARAM = 1:
Bit 1 - if setpoint high limit is violated ($\langle \text{setpoint} \rangle > \langle \text{setpoint} \rangle_HL$)
Bit 1 - if setpoint low limit is violated ($\langle \text{setpoint} \rangle < \langle \text{setpoint} \rangle_LL$)
Bit 1 - if setpoint high limit < low limit ($\langle \text{setpoint} \rangle_HL < \langle \text{setpoint} \rangle_LL$)
- Automatic commands:
Bit 2 - more than one command at the same time (e.g., START and HOLD, etc.)
Bit 3 - simultaneous changeover to MANUAL and AUTO (MAN = 1 and AUT = 1)
- Execution options:
Bit 4 - SCT = 0 and SCT_TAC = 0
Bit 4 - SCT = 1 and SCT_TAC = 1

MANUAL

- Execution options:
Bit 5 - SELFRESET = 1 and "Cyclic Operation" control and display section
(neither SELFRESET nor "Cyclic Operation" is executed)
- Internal commands and external interlocks**
Bit 6 - more than one internal command at the same time (e.g., INTABORT and INTCOMPLETE) (1 cycle)
Bit 7 - more than one external interlock at the same time (e.g., LOCKABORT and LOCKCOMPLETE)
Bit 8 - internal commands and external interlock at the same time (INT...+LOCK...) (1 cycle)

Command execution and LI_ERR

The following commands are **not** executed if LI_ERR is set to 1:

AUTO

- START (see also QDIS_START), COMPLETE, HOLD, RESUME, RESET

AUTO + MANUAL

- INTCOMPLETE, INTHOLD, INTRESUME, INTRESET, INTSTART
- LOCKCOMPLETE, LOCKHOLD

The following commands are **also** executed if LI_ERR is set to 1:

AUTO + MANUAL

- INTERERROR, INTSTOP, INTABORT
- LOCKERROR, LOCKSTOP, LOCKABORT

Calculation of OP_ERR and OP_ERR_STATE (SFC type only)

The following checks in MANUAL mode result in OP_ERR (for one AS execution cycle):

- Control Strategy
 - Bit 0 - if control strategy high limit is violated ($CSP_OP > CS_HL$) (1 cycle)
 - Bit 0 - if control strategy low limit is violated ($CSP_OP < CS_LL$) (1 cycle)
 - Bit 0 - if control strategies exist ($CS_LL > 0$) and the prepared control strategy (CSP_OP) is not enabled (SELCS).
- Setpoints
 - Bit 1 - if setpoint high limit is violated ($\langle setpoint \rangle_OP > \langle setpoint \rangle_HL$) (1 cycle)
 - Bit 1 - if setpoint low limit is violated ($\langle setpoint \rangle_OP < \langle setpoint \rangle_LL$) (1 cycle)
 - Bit 1 - if setpoint high limit < low limit ($\langle setpoint \rangle_HL < \langle setpoint \rangle_LL$)

Calculation of QDIS_START and DIS_START_STATE

AUTO

- Bit 0 (SFC type only) Control strategy and setpoint have not yet been checked ($QPARAM = 0$) or are bad
- Bit 1 In "Run" mode, the "Start in Run" enable is not set ($ENASTART = 0$) and/or continuous mode is not set ($QCONT = 0$)

MANUAL

- Bit 2 In "Run" mode, the "Start in Run" enable is not set ($ENASTART = 0$)
- Bit 3 There is an OP_ERR

AUTO + MANUAL

- Bit 4 There is an LI_ERR
- Bit 5 No "Start" enable ($ENSTART = 0$)
- Bit 6 There is an external interlock (LOCKERROR, LOCKHOLD, LOCKABORT, LOCKCOMPLETE, LOCKSTOP)
- Bit 7 There is the internal command INTERERROR
- Bit 8 (SFC type only) If control strategies are configured but none is selected ($QCSP = 0$)
- Bit 9 The transition to STARTING has been executed
- Bit 10 An AUTO/MANUAL changeover has just been executed (1 cycle long)

13.3.9 Messages (SFC inputs/outputs)

SFC inputs/outputs of the messages

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	O&M
AUX_PRO4..10 *)	Associated value 4..10	ANY	0	IO	
MSG_ACK *)	Acknowledge messages (ALARM_8P)	WORD	0	O	
MSG_ERR *)	Message error (ALARM_8P)	BOOL	0	O	
MSG_EVID	Message number (ALARM_8P)	DWORD	0	I	
MSG_LOCK	Lock messages: 0: Messages not locked 1: Messages locked	BOOL	0	I	+
MSG_STAT	Message status (ALARM_8P)	WORD	0	O	
MSG_SUP	Message suppression (ALARM_8P)	BOOL	0	O	+
NMSG_EVID **)	Message number (NOTIFY)	DWORD	0	I	
NMSG_EVID1 *)	Message number 1 (NOTIFY_8P)	DWORD	0	I	
NMSG_EVID2 *)	Message number 2 (NOTIFY_8P)	DWORD	0	I	
NMSG_STAT1 *)	Message status 1 (NOTIFY_8P)	WORD	0	O	
NMSG_STAT2 *)	Message status 2 (NOTIFY_8P)	WORD	0	O	
NSIG_12..16 *)	Generate event message 12..16 (NOTIFY_8P)	BOOL	0	IO	
RUNUPCYC *)	Number of run-up cycles	INT	3	I	
SIG_2..8	Generate message 2..8 (ALARM_8P)	BOOL	0	IO	
BA_ID	Associated value 1 (ALARM_8P, NOTIFY, NOTIFY_8P)	DWORD	0	I	+
BA_NA	Associated value 2 (ALARM_8P, NOTIFY, NOTIFY_8P)	STRING[32]	"	I	+
STEP_NO	Associated value 3 (ALARM_8P, NOTIFY, NOTIFY_8P)	DWORD	0	I	+

Key	
*)	Only for SFC type
**)	Only for SFC

13.3.10 Control words (SFC inputs/outputs)

SFC inputs/outputs of the control words

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	O&M
BA_CONTROL *)	BATCH control word (internal interface)	DWORD	0	IO	+
SFC_CONTROL	SFC control word (internal interface)	DWORD	0	IO	+

Key

*) Only for SFC type

Use of the inputs/outputs

- SIMATIC BATCH uses **BA_CONTROL** to send commands to the SFC (in AUTO).
- **SFC_CONTROL** allows the control and display section to send commands to the SFC (in MANUAL).

13.3.11 Status words (SFC inputs/outputs)

Status words of the SFC inputs/outputs

You can find explanations of the table entries in the section "Meanings in the table (Page 172)".

I/O	Meaning	Data type	Default	Type	OCM
BA_ADDSTATE	Additional BATCH status word	DWORD	0	O	
BA_STATE	BATCH status word	DWORD	0	O	+
SFC_STATE	SFC status word	DWORD	0	O	+
USTATUS *)	Status word in VSTATUS (can be set as required by user)	WORD	0	I	
VSTATUS *)	Status word	DWORD	0	O	+

Key

*) Only for SFC type

BA_ADDSTATE

Bit	Meaning	Source
0	Identifier "Self-completing"	SELFCOMP

1	Identifier "Self-resetting"	SELFRESET
2	Enable "Start in Run"	ENASTART
3	No trigger for sequencer in the "...ING" state after BA_CONTROL command "FastSFC"	-
4	The start command is at the SFC interface (START) or a start command has been activated via BA_CONTROL. The display is reset to the STARTING state.	-
5	Restart of the CPU	CPU_RESTART
6	1 = CPU restart after stop during block processing	-
7	1 = SFC Startup after CPU Restart > Option: Initialize SFC	-
8	1 = SFC block is being processed	-

BA_STATE

Bit	Meaning	Source
0	Operating state "Idle"	IDLE
1	Operating state "Run"	RUN / HELD_ERROR **)
2	Operating state "Completed"	COMPLETED
3	Operating state "Held"	HELD
4	Operating state "Aborted"	ABORTED
5	"Ready to complete" state	READY_TC
6	Operating state "Stopped"	STOPPED
7	Reserved for S88 "Paused"	-
8 *)	Identifier "Monitoring time for continuous mode exceeded"	QCONT_T_ERR
9	Operating state "Starting"	STARTING
10	Operating state "Resuming"	RESUMING / RESU_ERROR **)
11	Operating state "Completing"	COMPLETING / ERROR_COMPLETING **)
12	Operating state "Holding"	HOLDING
13	Operating state "Aborting"	ABORTING
14	Operating state "Stopping"	STOPPING
15	Reserved for S88 "Pausing"	-
16	Default MANUAL/AUTO mode	SFC properties/ operating parameters AS/ "Mode"
17	MANUAL/AUTO mode	QAUTMAN
18	BATCH: Enable	BA_EN/QBA_EN
19 *)	BATCH: Identifier "Run time exceeded"	QSTEP_T > STEP_T
20	Group display "Operator prompt transition"	T_OPRQG
21 *)	BATCH: Read in setpoints/process values again	REFRESH / QREFRESH
22	Identifier "Start lock"	QDIS_START
23	ID "Continuous mode" (Continuous)	CONT & RUN & READY_TC

13.3 SFC inputs/outputs sorted by usage

24	BATCH: "Occupied" identifier	OCCUPIED/QOCCUPIED
25	Operating state "Error"	ERROR / ERROR_COMPLETING **)
26	"Execution" error	EXEC_ERR
27	Group display "Step run time exceeded"	S_ERRG
28	"Parameter assignment/interconnection" error	LI_ERR
29	"Operator input" error	OP_ERR
30	"External" error	EXT_ERR/QEXT_ERR
31	BATCH: Group error	Bit25-30 ORed

Key	
*)	Only for SFC type
**	The state that was in effect before the error state is also displayed.
)	

SFC_STATE

Bit	Meaning	Source
0-4	Operating state bit 0-4 (states 0...4 are identical to the coding of SFC V5.2)	0: IDLE 1: RUN 2: COMPLETED 3: HELD 4: ABORTED 5: STARTING 6: COMPLETING 7: ERROR_COMPLETING 8: HOLDING 9: RESUMING 10: ERROR 11: HELD_ERROR 12: RESU_ERROR 13: ABORTING 14: STOPPING 15: STOPPED 16-31: free
5	"Continuous mode" state	CONT & RUN & READY_TC
6	"Ready to complete" state	READY_TC
7	AUTO/MANUAL mode	QAUTMAN (0=MANUAL, 1=AUTO)
8	Input error	OP_ERR
9	Interconnection/parameter assignment error	LI_ERR
10	"AUTO" operator enable	ENAUT/QENAUT
11	"MANUAL" enable	ENMAN/QENMAN
12	"Start" operator enable	ENSTART& *)

Bit	Meaning	Source
13	"Complete" operator enable	ENCOMPLETE& *)
14	"Hold" operator enable	ENHOLD& *)
15	"Resume" operator enable	ENRESUME& *)
16	"Abort" operator enable	ENABORT& *)
17	"Stop" operator enable	ENSTOP& *)
18	"Restart" operator enable	ENRESTART& *)
19	"Reset" operator enable	ENRESET& *)
20	"Error" operator enable	*)
21	Adopt default operating parameters when SFC starts	SFC properties/ operating parameters AS/ "Use default operating parameters..."
22	Group display "Operator prompt"	T_OPRQG
23	Group display "Step error"	S_ERRG
24-26	Control mode bit 0...2	0: QSCT 1: QSCTOC 2: QSCTAC 3: QSCC 4: QSCT_TAC
27	CPU restart after stop with active SFC	CPU-RESTART (consistent data)
28	CPU restart after stop during module processing	CPU-RESTART (inconsistent data)
29	Command output	MANUAL: Control and display section / "Command Output" AUTO:INSTROUT
30	Cyclic operation	MANUAL: Control and display section "Cyclic Operation" AUTO:CYCLEXEC
31	Time monitoring	MANUAL: Control and display section "Time Monitoring" AUTO:TIMEMON
*)	Depending on the operating state (see: Operating state logic for SFC (SFC OSL) (Page 239))	

USTATUS (SFC type only)

Bit	Meaning
0-15	User-assigned

VSTATUS (SFC type only)

Bit	Meaning	Source
0	Operating state "Idle"	IDLE
1	Operating state "Starting / Run"	RUN / STARTING / RESU_ERROR
2	Operating state "Completing / Completed"	COMPLETED / COMPLETING / ERROR_COMPETING

13.3 SFC inputs/outputs sorted by usage

3	Operating state "Held / Error"	HELD / HOLDING / ERROR / HELD_ERROR
4	Operating state "Aborted"	ABORTED / ABORTING
5	Operating state "Ready to complete"	READY_TC
6	Operating state "Stopped"	STOPPED / STOPPING
7	1 = EPHs and EMs 0 = All other SFC instances	-
8	Operating state "Error"	ERROR / ERROR_COMPETING
9	Transitional state "...ING"	"...ING" / ERROR
10	MANUAL/AUTO mode	QAUTMAN (0 = MANUAL, 1 = AUTO)
11	ID "Continuous mode" (Continuous)	CONT & RUN & READY_TC
12	SFC group error	ERRG
13	Allocation enable	QENACQUIRE
14	Allocation by equipment phase (EPH) active	QISACQUIRED
15	Allocation group error	QACQFAILED
16-31	USTATUS Bit 0-15	-

13.3.12 System parameters (SFC inputs/outputs)

SFC inputs/outputs of the system parameters

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

Special

I/O	Meaning	Data type	Default	Type
MODE	Processing mode: 0: Cyclic processing in cyclic interrupt OB 1: One-time processing in startup OB	BOOL	0	I
SFC_INIT	Initialize SFC	BOOL	0	IO

- The MODE parameter is preset by the SFC code generator for insertion of the SFC instance in the startup OB or cyclic interrupt OB and must not be modified.
- The SFC_INIT parameter is used to initialize the SFC instance for test purposes and must not be used.

13.3.13 Reserves (SFC inputs/outputs)

SFC inputs/outputs of the reserves

The reserve parameters fill offset gaps and are intended for future expansions and must therefore not be used.

You can find explanations of the table entries in the section: Meanings in the table (Page 172)

I/O	Meaning	Data type	Default	Type	Attr.	O&M
SFC						
IORES1..6	Reserved for internal use	BOOL	0	IO		
IRES1..5	Reserved for internal use	BOOL	0	I		
ORES_BY	Reserved for internal use	BYTE	0	O		
ORES1..15	Reserved for internal use	BOOL	0	O		
SFC type						
IORES_BY	Reserved for internal use	BYTE	0	IO		
IRESX1...8	Reserved for internal use	BOOL	0	IO		
IRES1...3	Reserved for internal use	BOOL	0	I		
IRESOX1...9	Reserved for internal use	BOOL	0	I		
IRESIO1	Reserved for internal use	INT	0	I		
IRESX10...17	Reserved for internal use	BOOL	0	I		
ORES_BY1..3	Reserved for internal use	BYTE	0	O		
ORES1..27	Reserved for internal use	BOOL	0	O		
ORESX01...07	Reserved for internal use	BOOL	0	O		

Compiling SFCs and SFC types

14.1 Overview of compiling charts, types, and instances

Compiling

When compiling (scope: entire program), all charts, SFC types, and SFC instances of the current chart folder are converted to the source language and then compiled. If you have modified an SFC, SFC type or SFC instance, you have the option of compiling the changes only.

When you compile, first all SFC types then all SFC instances and then all SFCs are compiled. Finally, the CFCs are compiled.

The SFC type is compiled only if a compilation-relevant change in the type has been made since the last compilation.

A function block type containing the I/Os defined in the interface and the configured sequencers is generated from the SFC type provided with an interface. As with the SFC, an action FC and a transition FC are also generated.

During compilation, a consistency check is performed automatically. You can start this test regardless of the compiling process using the menu command **SFC > Check Consistency**.

Logs are created during the consistency check and when compiling and downloading. You can view the logs in the SIMATIC Manager with the menu command **Options > Charts > Logs...** and in the SFC and CFC with the menu command **Options > Logs....**

After compilation, you can download the user program to the target system and perform testing and commissioning.

CPU comparison

Select the menu command **CPU > Compare** if you want to compare the time stamp of the most recent changes before downloading.

A dialog box opens listing the date and time of the following changes:

- Last download-relevant change
- Last compilation
- Compilation of the loaded program

If the time stamp of the last download-relevant change is older than the last offline program change, this has no effect on the execution of the program in the CPU. You do not need to download the program again.

If the time stamp of the "Last compilation" is older than the time stamp of the most recent modification relevant to the download, you must compile the charts and download them to the CPU so that they match.

If the time stamp of the "Compilation of the loaded program" is older than the time stamp of the "Last compilation," you must download the user program from the PC or programming device to the CPU so that they match.

Additional information

You can find additional information about compiling in the following sections:

Settings for compiling/downloading (Page 219)

How to compile (Page 220)

Overview of the blocks generated during compilation (Page 221)

Consistency check (Page 223)

Logs (Page 289)

14.2 Settings for compiling/downloading

Making the settings

Select the menu command **Options > Customize > Compile/Download....**

A dialog box opens in which you can find information about the resources related to compiling:

- You can specify which warning limits are to apply so that possible dangers are detected before you download.
- You can specify which resources should remain unused during compilation of the current chart folder.
This can be useful, for example, if you want to solve an automation task partly with charts and partly by programming (for example, STL, LAD or SCL programs) and when you have functions (FCs) or data blocks (DBs) from other sources in your user program.
- You can see the statistics showing how many resources (DBs, FCs) are available for compiling and how many are already being used.
- You can specify whether an image of the loaded program will be generated for the comparison. The image is created as an XML document and assigned to the program. With it you can determine which differences exist between the data to be loaded and the loaded data exist prior to reloading.

Note

If you work only with CFC and SFC in your program, you can leave the default compilation settings unchanged.

14.3 How to compile

Compiling the chart folder

1. Select the menu command **SFC > Compile...**
A dialog box opens where you can select the relevant radio button to specify compilation of the "Entire program" (all objects are compiled) or "Changes" (only the objects changed since the last compilation are compiled).
A dialog box for configuring the compilation settings opens.
2. Select whether to compile the "Entire program" or "Changes".
3. Optional: Deactivate the check box "Generate Module Driver" if for example your hardware is still incomplete for this compiling process. This deactivation only applies for this compile, for the next compile the check box is reactivated.
4. Optional: Activate the check box "Generate SCL Source" if you want to view a certain part of the program, for reasons of better understanding for example or for troubleshooting in the SCL code.
An SCL source file will be generated and saved in the source folder only if the check box is selected. This source file is not needed for compiling the program.
The deactivated check box is only valid for one compile; at the next compile it is deactivated again.
5. Click "OK". The dialog box closes and the compilation starts.

Result

The charts of the current program (chart folder) are checked for consistency and then compiled. The messages from the consistency check are written to a log file. If "errors" and "warnings" occur, you can jump to the relevant SFC by double-clicking the message.

During compilation, a dialog box with a progress bar is displayed. You can stop the compilation at any time by clicking "Cancel".

Viewing and printing logs

Select the menu command **Options > Logs...** to view and print messages from the consistency check or compilation.

Note

You can also perform a consistency check without compiling. To do so, select the menu command **SFC > Check Consistency...**

You can find additional information under: Consistency check (Page 223).

14.4 Overview of the blocks generated during compilation

Introduction

Understanding the following relationships can be helpful if you want to control the use of resources in the CPU.

Overview

The structure you have configured in SFCs is mapped to the S7 objects function block, function (FC) and data block (DB). All charts (CFC and SFC) of a chart folder are always compiled.

Each SFC in the chart folder is mapped as follows during compilation:

SFC	1 instance DB + n sequencer DBs + 2 FCs
SFC type	1 FB + 2 FCs
SFC instance	1 instance DB + n sequencer DBs

In order for the SFCs/SFC instances to be executed in the AS, the following blocks are also present in the block folder (depending on the configuration):

Object name	Symbolic name	For Chart (C) / Type (T)	Meaning
FB 245	@SFC_BZL	C / T	Operating state logic
FB 246	@SFC_ESM	C / T	Sequencer execution
FB 300	@SFC_RTS	P (V6.0 and higher)	Runtime system
FB 300	@SFC_INTP	P (up to V5.x)	Runtime system
FC 240	@SFC_OPI	T	Auxiliary block for INT setpoint
FC 241	@SFC_OPDI	T	Auxiliary block for DINT setpoint
FC 242	@SFC_OPR	T	Auxiliary block for REAL setpoint
FC 243	@SFC_OPB	T	Auxiliary block for BOOL setpoint
FC 244	@SFC_OPS	T	Auxiliary block for STRING setpoint
FC 250	@SFC_INDCALL	P	Auxiliary block for block calls
SFB 35	ALARM_8P	C / T	Messages with associated values for eight signals
SFB 36	NOTIFY	P	Messages without acknowledgement display
SFB 31	NOTIFY_8P	T	Messages without acknowledgement display for eight signals

If a "time" characteristic exists, the following block is also required:

FB 5	TIMER_P	T	Pulse generator
------	---------	---	-----------------

When you create a new SFC or SFC type, the required blocks are automatically copied to the block folder. For an SFC type, an additional copy of FB 247 is created (FB number is the first available number, for example, FB 1025).

14.4 Overview of the blocks generated during compilation

Furthermore, additional FCs are required in order to call the blocks processed in the CFC according to the configured run sequence, as follows:

- One FC for each OB used
- One FC for each runtime group

DBs are used by the CFC as follows:

- An instance DB is created for each inserted block of a block type (FB).
- Pool DBs are created for storing interim results of FCs, for example. One pool DB is required per data type. When the maximum length (4 Kbytes) is reached, another DB is created.

The quantity of FCs and DBs available varies according to the CPU. You must allocate these as follows:

- FCs for block types
- FCs and DBs for the structures created with the STL, LAD and SCL programming languages
- FCs and DBs for compiled charts

Calculation of SFC resources

The section "Notes on Usage" in the SFC Readme contains an Excel file "SFC resources" that you can use to calculate your approximate system resource requirements.

14.5 Consistency check

Scope of the check

Prior to the actual compilation, the system automatically performs the following consistency checks:

- Check to determine whether the block types in the user program match the types imported in the CFC
Inconsistencies can occur if block types in the user program have been changed or deleted following the block type import into the CFC.
- Check to determine whether symbolic references to shared addresses are entered in the symbol table
Inconsistencies can occur if the relevant entries have not yet been made in the symbol table or if they have been changed or deleted.
- Check to determine whether the interconnected data blocks (DBs) are present in the user program
Inconsistencies can occur if data blocks in the user program have been subsequently changed or deleted.
- Check to determine whether in/out parameters or block outputs of type "ANY", "STRING", "DATE_AND_TIME" or "POINTER" have been assigned (i.e., interconnected)
Check to determine whether all blocks accessed by conditions and instructions from the SFC are present.

You can also perform this consistency check without compiling, for example, if you want to check whether there is any point at all in compiling.

Procedure

1. To do so, select the menu command **Chart > Check Consistency**.
All charts of the current folder are checked. When the check is completed, the result is shown in the "Logs" dialog box.
2. You can also read or print the log at a later time.
3. In SFC, select the menu command **Options > Logs...**, or in the SIMATIC Manager, select the menu command **Options > Chart > Logs...**

Additional information pertaining to the logs is available under: Logs (Page 289)

Downloading to the AS

15.1 How to download programs

Introduction

In order to commission graphically generated charts/types for a CPU, the program must first be compiled and downloaded to the target system. The charts are downloaded to the CPU to which the user program containing the current chart folder is assigned.

Note

Programs created with SFC must always be downloaded to the CPU from SFC or CFC, because only this download function guarantees the consistency of the configuration data with the CPU data. The same download function is also used if you select the menu command **CPU > Compile and Download Objects...** in the SIMATIC Manager .

On the other hand, copying blocks in the "Offline Block Folder" to insert them in the "Online Block Folder" is not permitted.

Change log

The download to the CPU is - similar to working in test mode - a secure, logged function for S7 if SIMATIC Logon Service is installed and access protection and change logging are activated.

In addition to the change log of the SIMATIC Manager, in the ES the actions and the time stamp can be listed in the ES log when loading, if the current chart folder is activated for the ES log. Activate the selected chart folder in the dialog box of the object characteristics in the tab "ES Log".

You can find additional information in the section "The Change log" in the documentation or online help of the *CFC for SIMATIC S7*.

Requirements for downloading

Before you can download, the following requirements must be met:

- There must be a connection between the CPU and your programming device or PC.
- Edit mode is set.
- The program is compiled.

Procedure

1. Select the **CPU > Download** menu command.
The "Download" dialog box opens.
2. Select the download mode to specify if the "Entire program" or only "Changes" are to be downloaded to the CPU. You can also select the entire program in a "Download to test CPU".
3. Click "OK" to download the program.

If you have made download-relevant changes in the configuration and have not compiled since you made the changes, you will be prompted to compile before you download. If the compilation is free of errors, the download will be started automatically when the compilation is complete.

Download: Downloading the entire program

The following is performed with the "Entire program" download mode:

- The CPU is set to "STOP" after a prompt for confirmation.
- All blocks in the CPU are deleted.
- All charts in the current chart folder are downloaded to the CPU.

Notes on downloading the entire program

- Compiling the entire program does not necessarily mean that a complete download is necessary. If the program was already downloaded to the CPU prior to compiling, it is possible to download the changes only.
- If you cancel a full download, no download of changes is possible until a full download has been completed. Reason: The blocks were deleted on the CPU prior to the download.

Download: Downloading changes

In the "Changes" download mode with CPU in "RUN-P" mode, you can download changes in your configuration to the AS without having to change the CPU to STOP mode. With this type of download, you only download changes that have been made since the last download.

Notes on downloading changes

- If the sequencer topology has been changed in the SFCs/SFC types, for example, steps or transitions have been added, deleted, copied, moved or jump destination has been changed, and if the modified sequencer is active in the AS, these charts or all instances of the type must be disabled. Disabling before the download and enabling after the download is handled by the download program following positive acknowledgment of a prompt. Otherwise, the download is aborted.
Since topological changes in the active sequencer of an SFC type inevitably lead to an abort, it is recommended to add the required changes to the technological actions as conditional commands. This can completely avoid an abort.
If you perform the download with the SIMATIC Manager function "Compile and Download Objects", the SFCs are not automatically disabled. In this case, the download will not be performed and an error message is entered in the log.
- Modifications to the interface of the SFC type are transferred to the SFC instances immediately. For this reason, the SFC instances must be disabled during the downloads and execution must be suspended in the CPU. Disabling before the download and enabling after the download is handled by the download program following positive acknowledgment of a prompt. Otherwise, the download is aborted.
However, in the case of CPUs that support type change in RUN mode (TCiR function), that is CPU 410-5H PA, changed I/O interface of the SFC type can be loaded in RUN mode using the TCiR instance DB recovering method without disabling the SFC type instances.
- For changed technological actions within SFC types, you can download the changes to the CPU in the RUN mode after compilation without disabling the changed SFC.
- For changed SFCs/types/instances (SFC properties, sequencer properties, object properties of the step/transitions) whose sequencer topology have not changed, you can download the changes to the CPU in the RUN after compilation without disabling the changed SFC.
- If you have not changed the SFC itself, but only the objects that are accessed (for example, a symbol in the symbol table, runtime groups, block I/O), you do not have to disable the SFC before downloading changes.
- Following a download of changes, a disabled SFC with the "Autostart: on" property is not started automatically but must be started again by the operator (MAN) or using the external view/SFC instance (AUTO).

You can find additional information about status transitions when disabling the SFC in the section "Reaction of the SFC at deactivation before downloading changes only (Page 231)".

Download: Download changes to CPU in RUN mode

This option is only available with a CPU 410-5H PA, because this CPU supports type update in RUN mode. The operating modes for the various download functions differ with the CPU 410-5H PA from those of other CPUs.

You can find additional information about this in the section "Special considerations during download to a CPU 410-5H PA" of the *CFC for SIMATIC S7* manual or online help.

Notes on H CPUs

- If the H CPU is in solo mode, for example, due to the failure of a CPU, and a CPU failover has taken place, when there is an online access, (in this case, downloading changes), a selection dialog box opens. In this dialog box, you can select the required CPU. This dialog box does not appear in redundant mode.
- If you download program changes to a CPU that is operating in solo mode and then perform a "Switchover with modified configuration" using the menu command **CPU > Operating Mode...**, these changes are lost and you can only download the entire program.
Remedy: Download in redundant mode. In this case, you must make sure that the operating mode remains unchanged until the download is complete.

Note on F systems

Downloading changes in programs with a modified F component requires the entry of an F password. Without this legitimating, downloading is aborted.

Including user data blocks when downloading changes

The "Modified user DBs relevant to loading" option is set as the default setting and only relevant when you download changes. All blocks including the user data blocks are always downloaded when the entire program is downloaded.

The effects of this check box depend on whether the change in the user DB in the offline program is relevant to download or not:

- Changes in a user DB relevant to download:
A change that is relevant to download occurs when the interface of the user DB is changed, that is:
 - A DB variable is added or removed.
 - The data type of a DB variable is changed.
 - The value of a DB variable is changed.
- Changes in a user DB not relevant to download:
A change that is not relevant to download occurs when the user DB is changed but the interface of the user DB is retained. For example:
 - The author of the DB is changed.
 - The comment of the DB is changed.
 - The comment of the DB variable is changed.
 - The name of the DB variable is changed.

If this option is activated, the user data blocks only with the loading-relevant changes are downloaded to the CPU:

- If the user DB exists in the offline program and doesn't exist in the CPU, the user DB in the offline program is downloaded to the CPU.
- If the user DB exists in the offline program and also in the CPU, the content of the DB in the offline program overwrites the content of the corresponding DB in the CPU.
- If the user DB no longer exists (deleted) in the offline program but exists in the CPU, the corresponding user DB is deleted from the CPU after the download.

If this option is deactivated, the modified user data blocks are ignored when downloading changes. However, if the changes in the user DB in the offline program are relevant to download, a warning is displayed for the modified data blocks that were not downloaded.

- If the user DB exists in the offline program but doesn't exist in the CPU, downloading will be aborted with an error message.
- If the user DB no longer exists (deleted) in the offline program but exists in the CPU, a warning is displayed.

Note

Program change

Note the following after a program change:

- The data of the automation system and operating system may be inconsistent for AS-OS communication after downloading your changes. In this case, you need to transfer the data again with the menu command **Options > Wizard: Compile Multiple OSs > Start...**
 - Inconsistencies related to cross-CPU communication (AS-AS) may occur after downloading your changes.
-

Note

Note that there is no absolute guarantee that the CPU will be prevented from changing to STOP mode when you download changes.

You can find additional information in the section "System support for avoiding causes of CPU Stop" of the *CFC for SIMATIC S7* manual or online help.

Downloading a modified program to a test CPU

In the "Download to test CPU (Entire program)" download mode, you can download a modified program to a different CPU or to PLCSIM for testing purposes without losing the ability to download changes to the original CPU.

The entire program is downloaded to the test CPU (or to PLCSIM) without loss of the download identifiers and without the comparison time stamp being written to the ES data management.

You can find detailed information about downloading to the test CPU in the section "Downloading a modified program to a test CPU" of the *CFC for SIMATIC S7* manual or online help.

Displaying modifications before loading

The "Show Changes" function is only available if the Version Cross Manager (VXM) add-on package is installed and if an image of the downloaded program was generated.

Generating an image of the downloaded program

Assuming you have selected "Generate image of downloaded program for comparison" in the "Settings for Compilation/Download" dialog box, the image will be generated as an XML document and assigned to the program following a successful download operation.

Comparing programs

If you generated an image of the downloaded program, you can click "Show Changes" in the "S7 Download" dialog box to run a comparison between the XML file and the program you now want to download before it is actually downloaded.

The VXM is then called and runs the comparison. The comparison will enable you to see which data have changed in relation to the program that was downloaded previously. You can then decide whether or not to download the latest version.

15.2 Reaction of the SFC at deactivation before downloading changes only

Status transitions at SFC deactivation

Old state	Action	New state
Starting	Running sequencer is aborted	Aborting
Running	"	"
Completing	"	"
Error (completing)	"	"
Completed	"	"
Holding	"	"
Held	"	"
Resuming	"	"
Error	"	"
Held (error)	"	"
Resuming (error)	"	"
Stopping	"	"
Stopped	"	"
Aborting	Wait until sequencer is completed (running sequencer is not aborted!)	Aborted
Idle	Running sequencer is aborted	Idle (loader can load)
Completed	"	"
Stopped	"	"
Aborted	"	"

Example 1

A sequencer is running with the "Error" state and this sequencer should be loaded.

The following sequences occur:

Old state	Action	New state
Error	Running sequencer is aborted	Aborting
Aborting	Wait until sequencer is completed (running sequencer is not aborted!)	Aborted
Aborted	Running sequencer is aborted	Idle
Idle	Running sequencer is aborted	Idle (loader can load)

Example 2

A sequencer is running with the "Aborting" state and this sequencer should be loaded.

15.2 Reaction of the SFC at deactivation before downloading changes only

The following sequences occur:

Old state	Action	New state
Aborting	Wait until sequencer is completed (running sequencer is not aborted!)	Aborted
Aborted	Running sequencer is aborted	Idle
Idle	Running sequencer is aborted	Idle (loader can load)

Note

A sequencer is running in the "Aborting" state, cannot be deactivated by the loader if user confirmation is required by the configured step control mode

Example 3

A sequencer is running with the "Idle" state and this sequencer should be loaded.

The following sequences occur:

Old state	Action	New state
Idle	Running sequencer is aborted	Idle (loader can load)

You can find additional information about the effects of a modified SFC type under:

Configuration changes in the SFC type (Page 79)

Behavior of the sequential control system in the AS 16

16.1 Sequential control systems in the AS

Requirements

- The SFC created with the SFC Editor has been specified with the following:
 - The sequencers (start condition and pre/postprocessing) and their sequencer topologies
 - The action functions of the steps
 - The conditions of the transitions
 - The runtime properties
- The program has been compiled and its blocks have been downloaded to the AS.

Changing the state of the sequential control system

After the program has been downloaded to the AS, the sequential control system is in a defined state. You specify this defined state with the SFC Editor (default setting: "Idle"). By assigning start conditions and processing sequences for the sequencers, different sequences are executed in the AS depending of the SFC status.

You can change the runtime behavior as follows.

- You can change operating parameters with the SFC Editor for testing and commissioning or on the OS.
Example: the switch from "Transition (T)" to "Confirmation by operator (O)" step control mode or
- Switch the time monitoring from "off" to "on".

In order to achieve a desired operating state, for example, the SFC is influenced using commands that are input by the operator or are applied by the program.

Interaction with basic automation

The sequential control system in the AS relates to the basic automation using the action and transition functions. With parameter control, there are also relationships to the parameter data.

Each SFC is assigned a certain runtime behavior. The basic automation, including the blocks located in CFCs, may have a different runtime behavior from that of the SFC. This can be controlled by placing the external view of an SFC in the CFC. In the same way, an SFC instance is controlled by the blocks placed in the CFC.

The structure of the runtime system allows the sequential control system and the blocks of the basic automation to run in different cycles, thereby reducing the cycle load. In the same way, SFCs can be inserted in runtime groups and can thus be given different reduction ratios / phase offsets.

16.2 Specifying the runtime behavior

16.2.1 Runtime behavior of the sequential control system

Basics of runtime behavior

The behavior of a sequential control system depends on operating parameters "Operating state", "Operating mode", and "Step control mode" and the execution options.

You can set the operating parameters when testing and commissioning or on the OS (SFC Visualization). Exception: The "Autostart" parameter can only be set in the properties dialog box for the SFC, SFC type or SFC instance in the "AS Operating Parameters" tab.

The default settings for the operating parameters are as follows:

"Defaults" group		
Step control mode	T	Transition (process-driven)
Operating mode	MANUAL	Operator mode
Command output	On	
Cyclic operation	Off	
Time monitoring	Off	
"Start options" group		
Autostart	Off	
Use default operating parameters when SFC starts	Off	

The following operating state is set after the CPU starts:

Idle	(if Autostart = off)
Starting	(if Autostart = on)

You can find additional information on the AUTO/MAN changeover in the section "Operating modes (Page 234)".

16.2.2 Operating modes

SFC operating modes

The operating mode decides whether the running of the program is controlled by the operator or automatically controlled by the AS program.

SFC recognizes the following operating modes:

- **AUTO (Process mode)**
The execution is controlled by the AS program.
The program uses the parameter assignment and interconnections of inputs in the external view of the SFC or SFC instance set that you specified in the CFC to perform control.
- **MANUAL (operator mode)**
Execution is controlled by operator commands or by changing the execution options, such as commissioning or SFV.

You can enable the changeover from MANUAL to AUTO or AUTO to MANUAL with an operator input in test mode or in SFC Visualization or in the interface of the chart (external view) or the SFC instance. The ENAUT and ENMAN inputs/outputs are provided for this.

You can find additional information on the I/Os in the section "Operating modes (SFC inputs/outputs) (Page 195)".

16.2.3 Step control modes

Overview

The various step control modes change the behavior of the SFC when passing control from active steps to successor steps.

You can change the step control mode in all operating states. The individual step control modes are mutually exclusive. SFC recognizes the following step control modes:

Code for the step control modes	Control using...
T	Transition
T or O	Transition or confirmation by operator
T and O	Transition and confirmation by operator
O	Confirmation by operator
T/T and O	Step-specific confirmation by operator

Step control modes

T

Transition:

The sequential control system is process-driven (running automatically). When a transition is fulfilled, control is passed by disabling the predecessor steps and enabling the successor steps.

Step control mode "T" is possible in MANUAL mode (operator mode) or AUTO mode (process mode).

T or O

With transition **or** confirmation by operator:

The process or the operator controls the runtime of the sequential control system. An

16.2 Specifying the runtime behavior

operator prompt is set for each successor transition of an active step, and the next step is enabled after operator input is made. If the transition is fulfilled before the operator input, the next step is enabled without operator input (automatically).

Step control mode "T or O" is only possible in MANUAL mode (operator mode).

T and O

With transition **and** confirmation by operator:

The process and the operator control the runtime of the sequential control system. When the successor transition of an active step is fulfilled, an operator prompt is set and the next step is enabled only after operator input has been made.

Step control mode "T and O" is only possible in MANUAL mode (operator mode).

O

Confirmation by operator:

Only the operator controls the runtime of the sequential control system. The transitions do not need to be fulfilled. An operator prompt is set for all successor transitions of each active step, and the next step is enabled after the operator input.

Step control mode "O" is only possible in MANUAL mode (operator mode).

T/T and O

Step-specific confirmation by operator:

The "Confirmation" label is set or reset on a step-specific basis in the "Properties" dialog box for the step. The sequential control system runs in the following modes:

- **Process-controlled** in steps **without** the "confirmation" option.
Each completed transition following a step without this identifier passes on control without operator intervention (corresponds to "T").
- **Controlled by the operator** for steps **with** the "Confirmation" option.
When an active step with the "Confirmation" option becomes true, the system prompts the operator to acknowledge the next step and then continues the sequencer (corresponds to "T and O").

Step control mode "T /T and O" is possible in MANUAL mode (operator mode) or AUTO mode (process mode).

Note

In step control modes "O" and "T or O", the **minimum run time** of the step can be overridden by the operator.

16.2.4 Execution options

Introduction

You use the execution options to specify the behavior of the sequential control system. The individual execution options can be combined.

You can set the execution options in the "AS Operating Parameters" tab of the "Properties" dialog box. You open the dialog box with the menu command **SFC > Properties....**

Execution options of the SFC

- **Command output**

If "Command Output: *on*" is set, the actions of active steps are processed; if "Command Output: *off*" is set, they are not processed.

During commissioning, or if errors occur, blocking the command output in conjunction with suitable step control modes ("C", "T or C") can bring the sequential control system to a defined state without affecting the process.

- If command output is off, the minimum run time of a step has no effect.
- Changing the command output first takes effect after transition to the next step. Therefore, either all actions or no actions of a step are executed.

- **Cyclic operation**

If "Cyclic Operation: *on*" is set, execution is resumed automatically with "Starting" state after "Completed" state.

A sequencer that is to be processed in "Completed" state is exited again immediately in cyclic operation. Only the start step and final step are processed.

- If you are working in MANUAL mode and using SFC inputs/outputs for state control, SELFRESET = 0 must be set in cyclic operation.
- If you are working in AUTO mode, PARAM = 1 must be set (so that start disable QDIS_START = 0 can be set).

If "Cyclic Operation: *Off*" is set, the sequential control system remains in "Completed" state. If there is no command to exit the state, "Completed" is always processed cyclically.

This behavior applies to all operating states that can be exited only by means of commands. You can find information on operating modes in the section:

Operating state logic for SFC (SFC OSL) (Page 239)

16.2 Specifying the runtime behavior

- **Time monitoring**

If "Time Monitoring: *on*" is set, after a step has been enabled the active time ("Current run time") of the step is continuously compared with the monitoring time ("Maximum run time") and a step error is reported if the time is exceeded.

An acknowledgment of the step error does not complete this step.

If the active sequencer is stopped and RUNHOLD = 0, then the following applies:

- The time monitoring for this step continues.
- A step error is reported if a timeout occurs.

If "Time Monitoring: *off*" is set, there is no comparison of the active time and monitoring time.

- **Autostart**

When an SFC has the property "Autostart: *on*", the SFC goes immediately to "Starting" state after a restart in the CPU without any operator intervention. An automatic start does not occur after downloading changes (CPU does not go to STOP). The SFC must be restarted by the operator or via the program (for example, from the external view in the CFC).

If "Use default operating parameters when SFC starts : *on*" is set, all the defaults/options set in the runtime properties that, for example, have been changed in test mode are effective again when the SFC starts.

The setting for "Autostart" can only be changed in the **SFC > Properties > "Operating Parameters AS"** dialog box.

16.2.5 Operating states

16.2.5.1 Operating states

Operating state of the sequential control system

The operating state of the sequential control system indicates the current execution status and the operating behavior. For example, you can see if operator intervention is necessary for continued operation or which commands are available for switching to a different operating state.

You can influence the operating state as follows:

- In operator mode (MANUAL) using commands in test mode and in SFC Visualization.
- In process mode (AUTO) using interconnections with the external view of the SFC or with the SFC instance.

Operating state logic

The operating state logic of SFC (SFC OSL) describes the following:

- The possible states of an SFC or an SFC instance
- The transitions that are possible in a particular state
- The events that bring about a state change

In addition to the SFC OSL, there is also a separate simpler operating state logic, the sequencer OSL, for the sequencers configured in an SFC. It describes the following:

- The possible states of a sequencer
- The transitions that are possible in a particular state
- The events that bring about a state change

The relationship between SFC OSL and sequencer OSL results from the possibility of being able to execute one (of several) sequencers of the SFC in every state of the SFC OSL.

You will find more information on operating states in the sections:

- Operating state logic for SFC (SFC OSL) (Page 239)
- Operating state logic for sequencers (sequencer OSL) (Page 246)
- Commands (Page 253)

16.2.5.2 Operating state logic for SFC (SFC OSL)

Changing the operating states

The current operating state of the SFC OSL can be changed through the following events:

- Commands (e.g., Start, Resume, Hold) in "MANUAL" or "AUTO" mode
- External signals (inputs of the SFC or commands from another SFC, for example)
- Internal signals (commands of own sequencers, from test mode or SFC Visualization).
- Implicit state change

Diagram of the state changes

The operating state logic of an SFC is defined by the diagram of the state changes.

You can find additional information about this in the section "Diagram of the state changes for SFC OSL (Page 245)".

Notes on diagram

The SFC OSL contains some state changes that have been retained in order to ensure compatibility with older projects. These are displayed with **dashed blue lines** in the diagram.

The **numbers** in the diagram are the identifiers for the individual operating states whose meaning is listed in the following tables.

Operating states (SFC OSL)

No.	State	Meaning
1	Idle	Initial state; waiting for Start command.
2	Starting	Start processing after Start command.

16.2 Specifying the runtime behavior

No.	State	Meaning
3	Run	Normal processing after completion of the start processing.
4	Completing	Completion of processing after a Complete command or implicit completion.
5	Error (Completing)	Error processing during completion of processing.
6	Completed	Completion of processing is finished; waiting for Reset or Start command.
7	Holding	Hold processing after Hold command.
8	Held	Hold processing has completed; waiting for Resume command.
9	Resuming	Resume processing after Resume command.
10	Error	Error processing following error.
11	Held (Error)	Error processing is complete and no more errors are present; waiting for Resume command.
12	Resuming (Error)	Resume processing after Resume command.
13	Aborting	Abort processing after Abort command.
14	Aborted	Abort processing is complete; waiting for Reset or Start command.
15	Stopping	Stop processing after Stop command.
16	Stopped	Stop processing is complete; waiting for Reset command.

The following tables describe the transitions between the states (Source State No. / Destination State No.) as well as their triggers.

X = Possible from several states.

State changes using commands (SFC OSL)

Source/ Target	Command MANUAL (AUTO)	Meaning
X/2	START	Triggers start of processing by changing to the "Starting" state
3/4	COMPLETE	Triggers completion of processing by changing to the "Completing" state
2/7 3/7	HOLD	Triggers holding of processing by changing to the "Holding" state
8/9 11/12	RESUME	Triggers resumption of processing by changing to the "Resuming" or "Resuming (error)" state
X/10 4/5	ERROR	Triggers error processing by changing to the "Error" or "Error (Completing)" state
X/13	ABORT	Triggers abortion of processing by changing to the "Aborting" state
X/15	STOP	Triggers stop processing by changing to the "Stopping" state
X/2	RESTART	Triggers start of processing by changing to the "Starting" state
X/1	RESET	Changes to the "Idle" state

State changes using external signals (SFC OSL)

The external signals are inputs of the SFC that enable other blocks (for example, interlock blocks) to influence the transitions between states of the OSL.

However, the external signals work only within the limits of the OSL. Therefore, provision must be made for the state change in the state logic in order for this state change to occur.

Source/ Target	External signal	Meaning
3/4	LOCKCOMPLETE	Triggers completion of processing by changing to "Completing" state
2/7 3/7	LOCKHOLD	Triggers holding of processing by changing to "Holding" state
X/10 4/5	LOCKERROR	Triggers error processing by changing to "Error" or "Error (Completing)" state. The error processing is executed completely and the state then changes to "Held (Error)". If an error is pending, the state changes back to "Error" immediately.
X/13	LOCKABORT	Triggers abortion of processing by changing to "Aborting" state
X/15	LOCKSTOP	Triggers stop processing by changing to "Stopping" state

State changes using internal signals (SFC OSL)

Internal signals are SFC internal commands that are set in steps to cause a state change in the SFC. The signals are checked by the OSL and reset automatically after a state change.

The internal signals only work within the limits of the OSL. Therefore, provision must be made for the state change in the state logic in order for this state change to occur.

Source/ Target	Command	Meaning
X/2	INTSTART	Triggers start of processing by changing to the "Starting" state
3/4	INTCOMPLETE	Triggers completion of processing by changing to the "Completing" state
2/7 3/7	INTHOLD	Triggers holding of processing by changing to the "Holding" state
8/9 11/12	INTRESUME	Triggers resumption of processing by changing to the "Resuming" or "Resuming (error)" state
X/10 4/5	INTERROR	Triggers error processing by changing to the "Error" or "Error (Completing)" state
X/13	INTABORT	Triggers abortion of processing by changing to the "Aborting" state
X/15	INTSTOP	Triggers stop processing by changing to the "Stopping" state
X/2	INTRESTART	Triggers start of processing by changing to the "Starting" state
X/1	INTRESET	Changes to the "Idle" state

Implicit state changes (SFC OSL)

The states "Starting", "Completing", "Holding", "Resuming", "Error", "Resuming (Error)", "Aborting", and "Stopping" are transitional states. When a transitional state has been completely executed—in other words, there is no start condition fulfilled for a sequencer or a started sequencer has been completely executed—the state changes to the successor state defined in the OSL (implicit change).

The "Run" state is also treated as a transitional state if SFC input SELFCOMP= 1 is set (default for chart and type). Otherwise, the "Run" state can only be exited explicitly with the "Complete" command.

Note

Switching from SELFCOMP=0 to SELFCOMP=1 only has an effect when the "Run" and "Ready to Complete" state have not yet occurred.

Transitional states can, of course, also be exited due to an external problem or with the "Abort" and "Stop" commands. In this case, the appropriate change is executed, and not the implicit change.

Source/ Target	SFC Status	Meaning
2/3	Sequencer finished	Start processing finished; change to "Run"
3/4	Sequencer finished	Normal processing finished; change to "Completed" (prerequisite: external signal SELFCOMP=1)
4/6	Sequencer finished	Completion of processing finished; change to "Completed"
7/8	Sequencer finished	Hold processing finished; change to "Held"
x/3	Sequencer finished	Resume processing finished; change to "Run"
13/14	Sequencer finished	Abort processing finished; change to "Aborted"
15/16	Sequencer finished	Stop processing finished; change to "Stopped"

State control using SFC inputs/outputs (SFC OSL)

SELFCOMP, SELFRESET, RUNHOLD and CONT are configurable inputs of the SFC and influence the behavior of the OSL.

SELFCOMP is effective in "MANUAL" and "AUTO" modes.

Source/ Target	External signal	Meaning
3/4	SELFCOMP	SELFCOMP=1 (SFC self-completing): SFC changes by implication from the "Run" state to the "Completing" state after normal processing is completed. SELFCOMP=0 (SFC not self-completing): SFC remains in the "Run" state until the "Complete" command is issued. If there is a state change using the "Complete" command, normal processing is aborted if it was not yet completed.

SELFRESET is effective only in "MANUAL" mode.

Source/ Target	External sig- nal	Meaning
X/1	SELFRESET	SELFRESET=1: In the "Completed", "Aborted" and "Stopped" states, SFC makes an automatic and immediate change to "Idle" state. The result is that a sequencer is started and immediately aborted in the above states. SELFRESET=0: If the "Completed", "Aborted" and "Stopped" states have been completely processed, the SFC remains in the corresponding state until the "Reset" or "Start" command is issued (does not "Start" when "Stopped"). Note: In cyclic operation, SELFRESET = 1 must not be set. This causes an error (LI_ERR).

RUNHOLD is effective in "MANUAL" and "AUTO" modes.

Source/ Target	External sig- nal	Meaning
3/7	RUNHOLD	RUNHOLD=1: When there is a change from "Run" to "Holding", the previous sequencer is aborted and the new sequencer is started. RUNHOLD=0: When there is a change from "Run" to "Holding", the previous sequencer is held and the new sequencer is started.

You can find information on the operating mode transitions in the paragraph above "Diagram of the state changes" and in the section Processing an SFC (Page 249)" in the paragraphs "SFC operating mode changed" and "Notes on the operating mode transitions".

CONT

You can find more information on CONT in the section "Continuous mode (Page 243)".

16.2.5.3 Continuous mode

CONT status flag

In AUTO mode, you can use the additive status flag (CONT input and QCONT output) to allow cooperation between a user control system (or SIMATIC BATCH) and SFC. It allows you to restart an SFC without having to first switching it off. The OSL of the SFC remains in "Run" state and sets output QCONT = 1 if RUN = 1 and READY_TC = 1. If QCONT = 1 is set, QCONT_T_ERR = 0 is set.

Commands are executed regardless of the status flag.

16.2 Specifying the runtime behavior

The SFC can be restarted when `READY_TC = 1` is set. This is set by the SFC after processing the final step of the sequencer to be processed in "Run" state. If the same SFC is restarted or aborted, status flag `QCONT = 0` is set.

If the SFC has not been restarted within a configured time (`CONT_T # 0 s`), the error output `QCONT_T_ERR = 1` is set. This output allows individual reactions to the error.

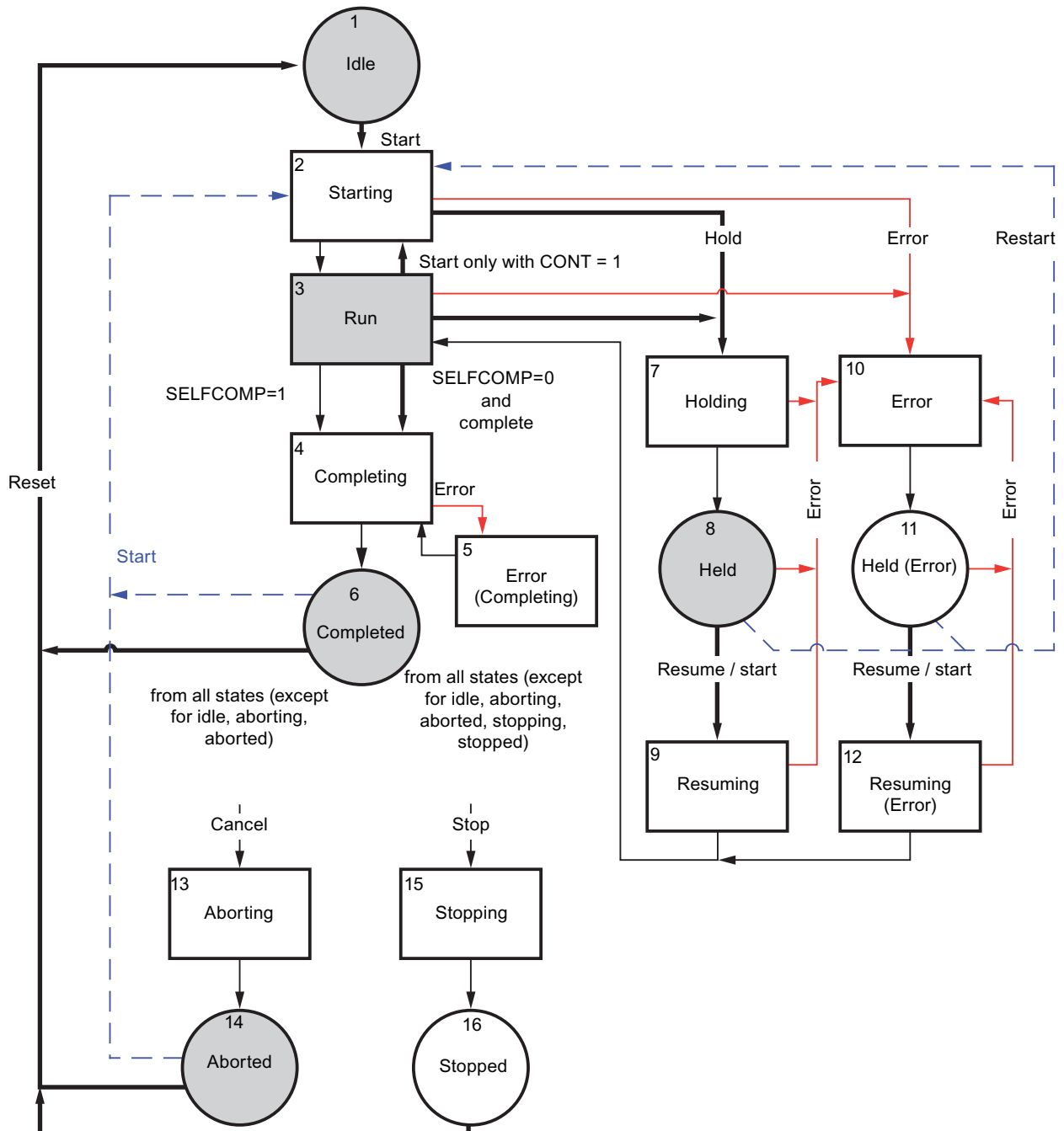
You will find more information on the continuous mode in the sections:

Start requirements of an SFC (chart/instance) (Page 254)

Continuous mode (SFC inputs/outputs) (Page 206)

16.2.5.4 Diagram of the state changes for SFC OSL

State changes for SFC OSL



Key

	States that are exited through events
	Transition states that are implicitly exited
	States applied from OSL for SFC V5.x
	Events: Commands / Conditions / External Signals / Internal Signals
	Event: Error
	Results applied from OSL for SFC V5.x
	Implicit transitions that are triggered from SFC, if the active sequencer has been processed to completion or if there is not sequencer to process.

16.2.5.5 Operating state logic for sequencers (sequencer OSL)**Overview**

The sequencer OSL controls processing of sequencers.

The operating state logic of the sequencer is defined by the diagram of the state changes. You can find additional information on this in the section: Diagram of the state changes for sequencer OSL (Page 248)

The sequencer OSL is executed independent of the SFC-OSL when a sequencer is processed. This means that the sequencer has a state which differs from the SFC state. For example, the state of the SFC OSL can be "Holding", whereas the state of the sequencer OSL is "Run" (due to the processing of the sequencer to the "Holding" state). The processing of the sequencer OSL is subordinate to the SFC OSL. This means that the state change in the SFC OSL usually effects the state change in the sequencer OSL.

States of the sequencer OSL

No.	State	Meaning
1	Idle	Initial state; waiting for Start command
2	Run	Normal processing
3	Completed	Normal processing finished; waiting for Start command
4	Held	Waiting for Resume command
5	Aborted	Waiting for Start command

State changes by means of commands (sequencer OSL)

The commands for the sequencer OSL are internal commands of the SFC runtime system.

The following tables describe the transitions between the states (Source State No. / Destination State No.) as well as their triggers.

Source/ Target	Command	Meaning
X/2	Start	Triggers sequencer execution by changing to the "Run" state
2/4	Hold	Holds sequencer execution by changing to the "Held" state
4/2	Resume	Resumes sequencer execution by changing to the "Run" state
4/2	Restart	Restarts sequencer execution by changing to the "Run" state
X/5	Cancel	Aborts sequencer execution by changing to the "Aborted" state

X = Possible from several states.

Implicit state changes (sequencer OSL)

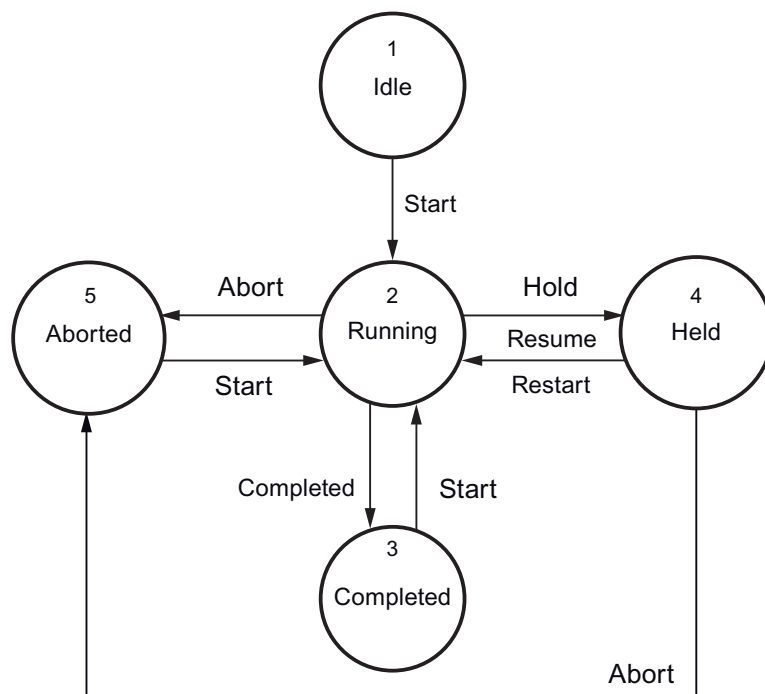
The "Run" state is a transitional state. An implicit state change to "Completed" state occurs if "Run" and, thus, the sequencer including the final step has been processed completely.

"Run" can, of course, also be exited by means of the "Abort" and "Hold" commands. In this case, the appropriate transition triggered by the command is executed and not the implicit transition.

Source/ Target	Sequencer status	Meaning
2/3	Completed	Sequencer execution is complete

16.2.5.6 Diagram of the state changes for sequencer OSL

State changes for sequencer OSL



Key:

○	States that are changed due to commands/operations
➡	Operator commands
→	Implicit transitions triggered by the SFC

16.3 Processing the SFC in the AS

16.3.1 Processing an SFC

Introduction

In the following description of the basic principles of cyclic operation of an SFC, it is assumed that the SFC contains several sequencers with user-defined start conditions.

How it works

The SFC is executed based on the input signals pending in the SFC interface and the internal status of the SFC. After an entire chart folder has been downloaded to the automation system, all the SFCs are in "Idle" state or "Starting" state if Autostart = 1 is set. The following is checked:

- The operating mode, commands, and external and internal signals are checked. This determines the state that the SFC should take based on the input signals.
- All start conditions of the sequencers are checked. This determines which sequencer is to be processed based on the preceding checks.

SFC operating mode unchanged

If the SFC status is unchanged, the sequencer whose start condition is fulfilled and has the highest priority of all sequencers with a fulfilled start condition is executed. If there are several sequences with a fulfilled start condition and the same priority, then the sequencer with the tab furthest left in the editor is executed (similar to the execution of alternative branches). If the sequencer that was processed up to now is different from the sequencer that is to start processed now, the previous sequencer is aborted and the new sequencer starts.

The start condition triggers the start of a sequencer. While the sequencer is being processed, it is not necessary for the start condition to remain fulfilled. The sequencer is processed until it is completed or until a sequencer with a higher priority and fulfilled start condition has to be processed or the sequencer is aborted or held due to a operating mode transition.

In the "Idle", "Completed", "Aborted", "Stopped", "Held", "Held (Error)", "Error" and "Run" states (when SELFCOMP=0), sequencers are processed until you exit the state by means of a command. A sequencer can, therefore, be repeated any number of times when the corresponding start condition is fulfilled and no command is pending. To prevent this behavior, you can configure the final transition of the sequencer with an unfulfilled condition. This causes the sequencer to "hang" at this transition and to exit only following a command.

SFC operating mode changed

If the SFC operating mode has changed, the SFC processes the previous sequencer and the new sequencer according to the state change that has taken place:

When an operating mode transition from "Run" to "Holding" is performed, the active sequencer is held (with RUNHOLD = 0) or aborted (with RUNHOLD = 1) and the new sequencer is started.

If the previous sequencer has been completely processed, the state changes from "Resuming" or "Resuming (Error)" to "Run". The new sequencer is resumed (with RUNHOLD = 0) or started (with RUNHOLD = 1) at the transition from "Resuming" and is started at the transition from "Resuming (Error)".

If there is an implicit state change, the change is executed when the sequencer of the first transitional state has been processed completely and is therefore terminated. If there is no sequencer with a fulfilled start condition, the implicit change is executed immediately and the new sequencer starts.

With all other state changes resulting from commands, external or internal signals, the previous sequencer is aborted if it has not yet been processed completely and the new sequencer starts.

Notes on operating mode transitions

- When a sequencer is held, the running step is held immediately after the execution phase. The following when resuming the sequencer:
 - When the transition is fulfilled, the step is resumed **after** the execution phase.
 - When the transition is not fulfilled, the resumption occurs **with** the execution phase of the step.
- When a sequencer aborts or when there is an active change in the control strategy, the active steps are always processed completely followed by the final step. The completion of the active steps and the initialization or processing of the final step are processed in one cycle.
- A sequencer can be processed in an SFC even if the SFC has not received a start command. This is then the case when, for example, in the status "Idle" the start condition of one or several sequencers is fulfilled or sequencers exist, whose starting conditions are indicated independently of states.

You can find more information on this in the sections "Inputs/outputs of the SFC type standard interface (Page 174)" and "Operating state logic for SFC (SFC OSL) (Page 239)".

Coordination of sequencers and target steps

You can influence the start behavior of a sequencer that will become active in the future by setting a target step for this sequencer in step actions of the current sequencer (for example `<SFCName>.TARGETSEQ:=2; <SFCName>.TARGETSTEP:=5`). This corresponds to setting a target step manually during testing and commissioning; however, the difference in this case is that all other target steps of the sequencer including those set manually are reset. This means that you can specify the required start step for the sequencer to be started at any time.

TARGETSEQ = 0 is set when the step action has been executed.

Note that target steps cannot be set in simultaneous branches.

Example: By setting `<SFCName>.TARGETSEQ:=2` and `<SFCName>.TARGETSTEP:=0`, all the target steps in the sequencer with number 2 are cleared.

A target step is taken into account when a sequencer starts or resumes and is then deleted.

Because the required execution and the target step both generally depend on the prior execution of the SFC, you can query the last active sequencer and the last active step in a

transition condition (for example <SFCName>.LASTSEQ:=3; <SFCName>.LASTSTEP:=2) to set different target steps depending on the result of the query.

Sequences and steps are identified by their number. These are displayed in the "Properties" dialog boxes and can be used to configure target steps.

Cooperation between operator control or SIMATIC BATCH and SFC

If an SFC instance is being used by SIMATIC BATCH, the following process occurs:

- SIMATIC BATCH sets the input OCCUPIED = 1.
- The SFC sets the output QOCCUPIED = 1 and the occupied code (bit 24) for BA_STATE.
- OCCUPIED and QOCCUPIED are set with SFC = 0 for a reset (transition to "Idle").
When OCCUPIED is deleted by SIMATIC BATCH itself, the QPARAM = 0 output is also set. This means that any pending, incorrect setpoint is no longer checked, which means an error is not displayed (LI_ERR = 0).

16.3.2 Processing an SFC after CPU stop and restart

Startup characteristics

When the CPU goes into the stop state, the running sequencer stops at the step it is currently processing. When the CPU is restarted, the SFC is initialized and the data valid before the stop occurred is lost. The property of the SFC is the default state.

If you want the SFC to retain its state when the CPU restarts, you need to set the AS operating parameter defaults correspondingly (**Chart > Properties > tab: AS Operating Parameters > SFC Startup after CPU Restart > Option: Retain SFC state**).

The operator can decide how the SFC is to be further processed based on the SFC state and process state. A changeover from AUTO to MANUAL may need to be performed for this.

Another factor to be considered is whether the CPU went to stop during block processing or processing was completed. If the first case is true, the data are inconsistent. The following procedure needs to be performed as a result:

Procedure after restart

Requirement:

- The SFC is not in the "Idle" state.
or
- The SFC is in the "Idle" state and a sequencer is active.

CPU stop occurs during block processing (data are inconsistent)	
Command	Processing
Reset	SFC continues in the "Idle" operating state, whereby all sequencers are initialized.

CPU stop does not occur during block processing (data are consistent)	
Command	Processing
Resume	SFC resumes in the current operating state
Cancel	SFC aborts according to the OSL
Stop	SFC stops according to the OSL

The "Resume" command is always allowed regardless of the operating state, but the "Abort" and "Stop" commands are only permitted when they conform to the status transitions of the operating state logic. You can find additional information on this in the section: Operating state logic for SFC (SFC OSL) (Page 239)

The state symbols are displayed and the "CPU_RESTART" and "EXEC_ERR" outputs are set until one of the commands is executed (Resume, Abort, Stop, Reset).

You can use the TRIG_CPU_RESTART output for the start condition of the sequencers that should be executed after the CPU restart. This output is only set by the system and must be reset by a configured actions (such as step action TRIG_CPU_RESTART = 0).

Note

After the CPU restart, the ENRESUME, ENABORT, ENSTOP and ENRESET are not considered for the buttons Resume, Abort, Stop and Reset.

Display in test mode

In test mode, the steps for a CPU stop and after a CPU restart are accordingly identified.

CPU stop	
CPU restart and SFC with consistent data	
CPU restart and SFC with inconsistent data	

The symbols for "CPU restart and SFC with consistent data" and "CPU restart and SFC with inconsistent data" are also displayed in the operator control and display section.

Additional information

You can find additional information on this in the following sections:

- "States of the steps and transitions (Page 274)"
- "Display in test mode (Page 270)"

16.3.3 Commands

Introduction

Regarding the commands for changing the operating mode, there is a distinction between commands for "MANUAL" and "AUTO" modes.

Commands for "MANUAL" mode

In "MANUAL" mode, you can set or modify the operating states using the buttons of the operator interface in SFC test mode or in SFC Visualization (you can find information about this in the "MANUAL commands for the SFC" table below). The ability for the operator to issue commands depends on the operating state and the command enables "EN.....".

Commands for "AUTO" mode

In "AUTO" mode, depending on the command enables "EN....", the commands are issued using the interface inputs "START," by assigning parameters or interconnecting to a higher-level automatic controller.

Commands for both operating modes

Regardless of the operating mode and the command enables, you can issue interlock commands via the "LOCKCOMPLETE," interface inputs.

Regardless of the mode and the command enables, you can issue the commands via the (IN_OUT) "INTSTART," interface inputs in the object properties of the steps. These commands are reset after they have been executed, in other words, when the state changes.

You can find additional information on this in the section: Commands and operating states (SFC inputs/outputs) (Page 196) under "Explanations of the commands and operating state changes".

Note

The commands arranged as buttons in the control section of the online window (Test mode) are effective for the SFC or SFC instance only and not for sequencers.

MANUAL commands for the SFC

Button	Command	Meaning
	Start	Triggers start of processing by changing to the "Starting" state
	Hold	Triggers holding of processing by changing to the "Holding" state
	Resume	Triggers resumption of processing by changing to the "Resuming" or "Resuming (Error)" state
	Cancel	Triggers abortion of processing by changing to the "Aborting" state

Button	Command	Meaning
	Exit	Triggers completion of processing by changing to the "Completing" state
	Stop	Triggers stop processing by changing to the "Stopping" state
	Restart	Triggers restart of processing by changing to the "Starting" state
	Reset	Changes to the "Idle" state
	Error	Triggers error processing by changing to the "Error" or "Error (Completing)" state

The effects of commands are graphically illustrated in the following section: Diagram of the state changes for SFC OSL (Page 245)

16.3.4 Start requirements of an SFC (chart/instance)

General requirements

An SFC is started only when the requirements for starting are fulfilled. The start enable must be set (ENSTART = 1) and the SFC must be in a state in which starting is permitted. The following conditions must also be met:

- There is no interlock error (LI_ERR = 0).
- None of the signals INTERERROR, LOCKERROR, LOCKCOMPLETE, LOCKHOLD, LOCKABORT or LOCKSTOP is active at the same time.
- There is no operator error in MANUAL mode (OP_ERR = 0).

If you are using control strategies for an SFC instance, you must have selected one of the defined control strategies (CS=<defined control strategy>). With CS=0 or CS>CS_HL, it is not possible to start the SFC instance. If you have not configured any control strategies, the CS, CS_LL, CS_HL inputs/outputs are assigned the value 0 during compilation and must not be modified.

If you are using setpoints, you must specify them within the respective limits. Otherwise it is not possible to start the SFC instance.

With an SFC instance in AUTO mode, you must have set the parameter transfer (PARAM=1) before starting. The transferred control strategies and setpoints are then checked and the start disable reset (QDIS_START=0) if the parameters are valid. The start disable remains set until a PARAM is set by a parameter transfer. Following a successful start, the start disable remains set until the next parameter transfer.

If one of the conditions above is not fulfilled, the start disable (QDIS_START=1) is set and the start is not executed. The "Start" button is not enabled in this case.

Starting in the "Run" state

To be able to start in "Run" state, you must have set the additional start enable (ENASTART=1) and deactivated the self termination (SELFCOMP=0).

In AUTO mode, you must have also activated continuous mode (CONT=1).

The start condition of the sequencer that is to be processed in the "Run" state must also ensure that the sequencer has not already been processed completely (READY_TC=0). This prevents the sequencer from being repeated cyclically under certain circumstances because the start condition is satisfied.

Example of the start condition:

RUN=TRUE AND READY_TC=FALSE.

When an SFC starts or an active change in the control strategy takes place, the finished ID is reset (READY_TC=0) and after complete execution of the sequencer in "Run" state it is set again (READY_TC=1).

Restart in the "Held / Held (Error)" state

ENSTART = 1 and ENRESTART = 1 must be set to be able to restart in the "Held / Held (Error)" state.

You can find additional information about multiple simultaneously pending commands in the sections:

- Behavior when multiple commands are queued (Page 255)
- Continuous mode (Page 243)

16.3.5 Behavior when multiple commands are queued

Behavior when multiple commands are queued

If several commands are queued simultaneously (for example, external and internal commands), an interconnection error (LI_ERR=1) is usually displayed. The commands will not be executed or only one of the commands will be executed.

Additional information

You can find additional information in the section:
Start requirements of an SFC (chart/instance) (Page 254)

16.3.6 Processing the SFC elements

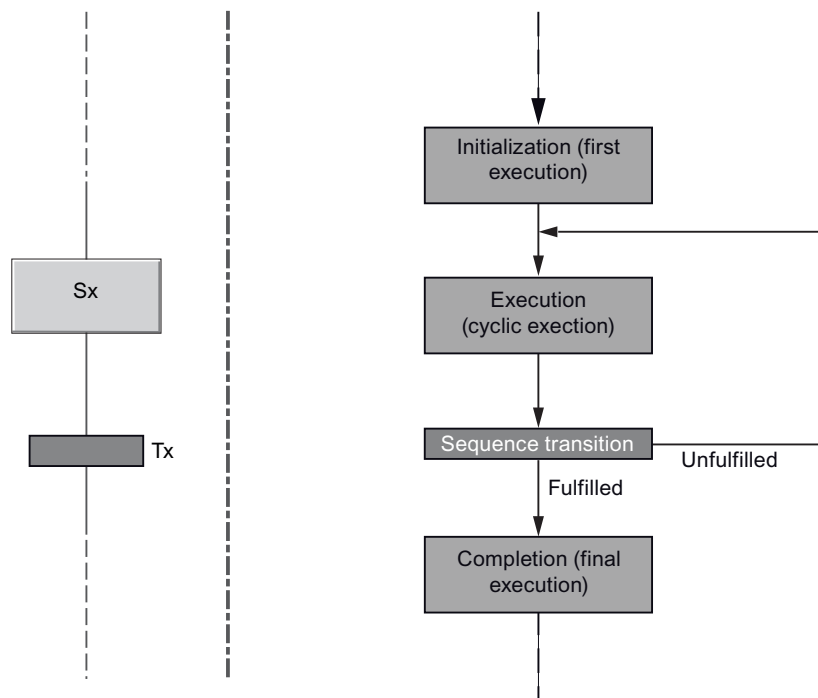
16.3.6.1 Runtime phases of a step

Runtime phases

Each step is divided into three phases (actions):

- Initialization: Action for initial processing
- Processing: Action for cyclic processing
- Termination: Action for final processing

The following figure shows the runtime phases of a step in conjunction with a successor transition: On the left are the elements of the sequencer topology and on the right the corresponding runtime phases.



16.3.6.2 Processing a step and transition

Processing steps and transitions

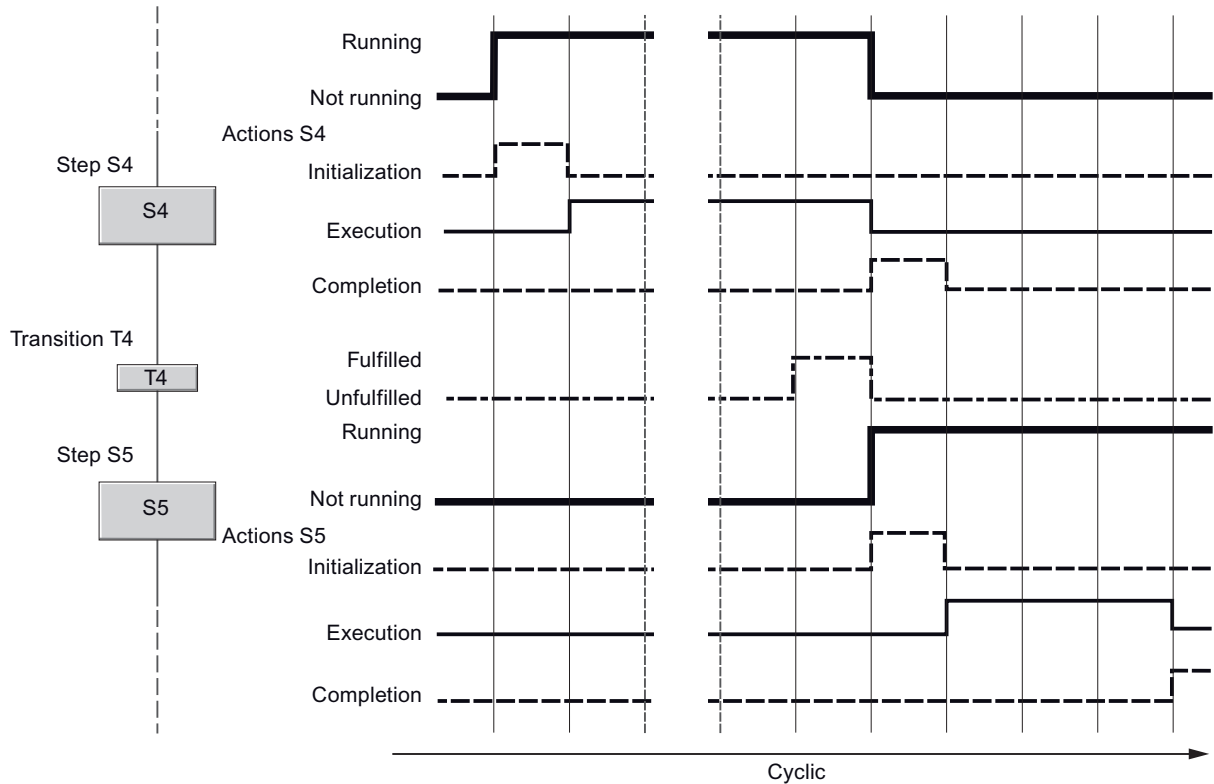
The **Start step** is activated without checking conditions and its actions are performed when the sequential control system is started.

A (normal) **step** has the states "active" and "inactive". A step becomes active after the preceding transition passes control to it. The actions are then triggered and controlled. A step becomes inactive after the successor transition is fulfilled.

If there is an abort or an active change in the control strategy, the completion of processing of the previously active step is still executed and the initialization of the final step is activated (overlapping in the same cycle).

A **transition** has the states "FALSE" and "TRUE". The state of the successor transition of the active step is checked. If a successor transition is fulfilled and the step enabling condition is true, the previous step is disabled and the next step is enabled. The transition is not checked in relation with the step control mode until any configured minimum run time has expired.

The actions of the **final step** are performed once only.



When the chart progresses from one step to the next, the termination action is executed in the same cycle as the first action (initialization or processing) of the next step.

This allows the "non-latching behavior" specified in IEC 1131 - 3 to be achieved.

Example:

In step S4, the processing phase opens a valve and the termination phase closes it again. If the same valve is opened again in the first action of the next step (S5), the overlapping of the two actions (both in one cycle) means that the valve is not closed.

Special situations

The example in the schematic shows the time behavior when all three actions of a step are configured.

Other combinations are also possible:

- If no "initialization" action is configured, the "processing" action begins immediately when the step is enabled.
- If no "termination" action is configured, the step is disabled immediately as soon as the transition is fulfilled.

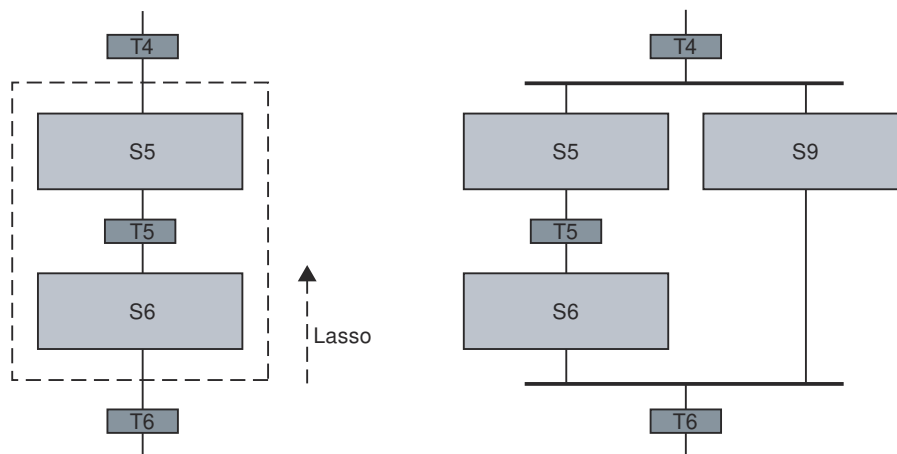
The minimum time that a step is active depends on the number of configured actions. This is one or two actions with a normal step, and up to three actions with a final step.

If a minimum run time is set for the step, the step remains active for at least this time even if the transition condition is fulfilled earlier.

16.3.6.3 Processing a simultaneous branch

Processing a simultaneous branch

Simultaneous branches are processed in a single cycle, and the parallel sequences run independently of one another.

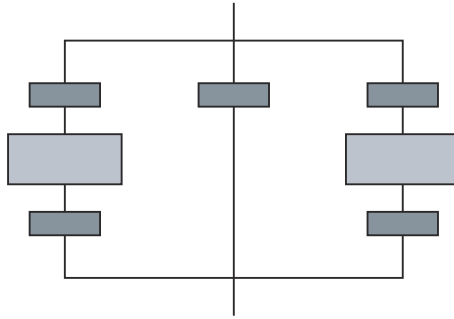


The transition after the simultaneous branch becomes true when all steps at the end of the sequencers are active and the conditions are fulfilled.

16.3.6.4 Processing an alternative branch

Processing an alternative branch

The sequencer that is processed in an alternative branch is the one whose transition condition is fulfilled first.



If several conditions are fulfilled at the same time, the transition that is located furthest left in the sequencer topology is activated.

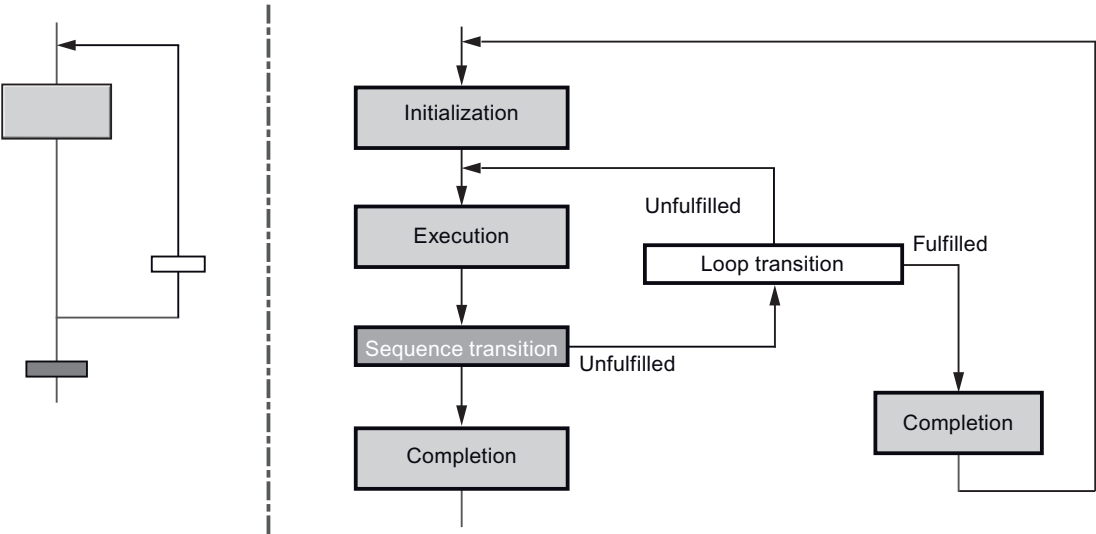
Note

An alternative branch cannot contain an unprogrammed transition at the start of a sequencer. Reason: Unprogrammed transitions are always set to TRUE by default, and are thus automatic fulfilled. This means that they are always fulfilled **before** a programmed transition.

16.3.6.5 Processing a loop

Processing a loop

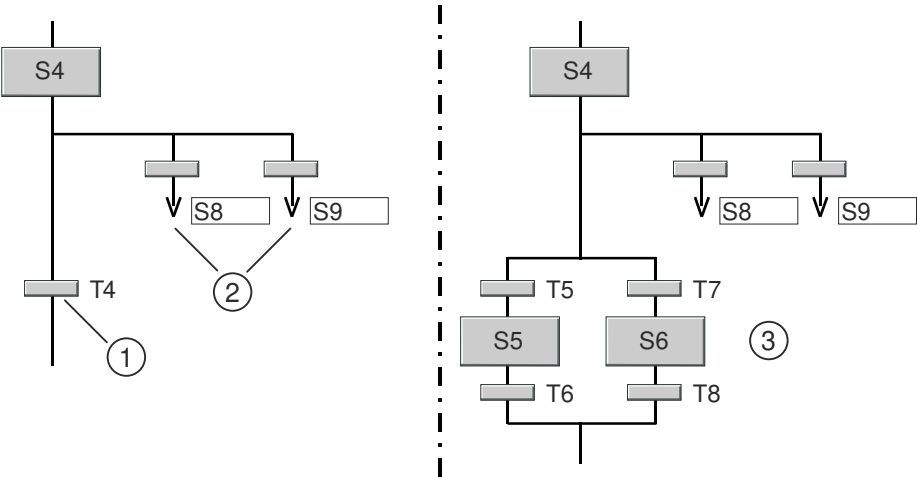
The figure below shows the runtime phases of a loop: On the left are the elements of the sequencer topology and on the right the corresponding runtime phases.



16.3.6.6 Processing a jump

Processing a jump

The jump is executed when the transition of the jump is satisfied.



Key	
1	Successor transition

2	Jumps
3	Alternative branch

Example on the left in the figure: If there is more than one jump following the origin of a jump (S4), then—similar to processing in an alternative branch—the jump whose transition is fulfilled first is executed. If several transitions are fulfilled at the same time, the transition that is located furthest left is activated.

Example on the right in the figure: If an alternative branch follows instead of a successor transition (such as T4 in the left configuration) in the sequencer, all the transitions of the alternative branch (T5 and T7) are queried before the transitions of the jumps.

16.4 Changing the control strategy and setpoints

16.4.1 Changing the control strategy and setpoints for an SFC instance

Introduction

You can change the control strategy and/or setpoints before restarting an SFC instance or change the setpoints and apply them immediately.

Changing control strategy and setpoints

There are two mechanisms for changing the control strategy and/or setpoints:

1. Changing the control strategy and/or setpoints and applying them at the next start

In **AUTO mode**, proceed as follows:

Use the PARAM and START inputs/outputs. The CS inputs for the control strategy and <setpoint> for the setpoints are assigned the new values and then PARAM input = 1 is set. The QPARAM output signals that a change of control strategy and/or setpoints at the next start has been requested; the output is reset after the start. When the SFC instance starts, the control strategy and/or setpoints are applied.

In **MANUAL mode**, follow the steps outlined below:

Set the appropriate enables for changing the control strategy ENCSP or the setpoints <setpoint>_ENOPP, to allow changes to be performed. This enables the corresponding input options in the faceplate for SFC instances.

The inputs CSP_OP for the control strategy and <setpoint>_OPP for the setpoints are assigned the new values. You can do this in CFC Test mode directly in the SFC instance or in the faceplate of the SFC instance in the "Prepared values" tab. The control strategy and/or setpoints are applied when the start is executed.

If the values are valid, they are displayed at the outputs QCSP for the control strategy and <setpoint>_QP for the setpoints in **both modes**. If the limits for the control strategy or setpoints are exceeded, this is displayed at the corresponding outputs <setpoint>_ERR. Invalid values are not applied to the outputs QCSP or <setpoint>_QP. After the start, the current control strategy is displayed at the QCS output and the current setpoints at the <setpoint>_Q outputs (QCS = QCSP, "sw"_Q = "sw"_QP).

2. Changing setpoints and applying them immediately

This mechanism cannot be used to change the control strategy.

In **AUTO mode**, proceed as follows:

Use the TAKESP input/output. The inputs <setpoint> for the setpoints are assigned the new value and the TAKESP input is set. The setpoints are adopted immediately if the TAKESP input is set and no error has been found in the setpoints (high/low limit). The QTAKESP output signals that an immediate change of setpoints was requested.

The PARAM signal must always be set when setpoint changes are made because the setpoints are checked and adopted in the prepared values only when PARAM is set. Only checked values are applied at START or TAKESP.

In **MANUAL mode**, follow the steps outlined below:

You must set the appropriate enables for changing the setpoints <setpoint>_ENOP to allow changes to be made. This enables the corresponding input options in the faceplate for SFC instances.

The <setpoint>_OP inputs for the setpoints are assigned the new values. You can do this in CFC Test mode directly in the SFC instance or in the faceplate of the SFC instance in the "Current values" tab. The setpoints are applied immediately.

Note on working in the faceplate: The control of setpoints must be enabled on the "Current values" page. The "'Current setpoints" can be used page' property must be selected in the "@pg_@sfc_type_actuasp.pdf" faceplate subimage.

You can find additional information on this in the online help of *SFC Visualization*.

Assuming the values are valid, they are displayed at the <setpoint>_Q outputs for the setpoints in **both modes**. If the limits for the control strategy or setpoints have been exceeded (error case), this is displayed at the corresponding outputs <setpoint>_ERR. Invalid values are not applied at the <setpoint>_Q outputs.

Summary

In both mechanisms, the output CSSPACCEPT indicates whether or not the requested changes were applied. This output is set after applying the control strategy and/or setpoints by the SFC instance at the relevant outputs. The CSSPACCEPT output is set for one processing cycle (for 1.) or remains set until the TAKESP request is reset (for 2.).

Output LI_ERR indicates whether or not an error occurred when attempting to apply value changes with the result that the values could not be completely applied. If LI_ERR is set, QDIS_START is not reset and a start and, thus, the adoption of the control strategy and/or setpoints is prevented (for 1.).

16.4.2 Tracking the control strategy and setpoints in AUTO mode

Tracking the control strategy and setpoints

"Tracking" involves writing back of the current control strategy and setpoint values to the corresponding operator inputs (only in AUTO mode). As a result, the setpoints of the last active control strategy used is immediately available at the next start.

Control strategy tracking:			
MANUAL	At start if CSP_DEFAULT > 0 : CSP_OP = CSP_DEFAULT		
AUTO	if CSP_DEFAULT > 0 : CSP_OP = CSP_DEFAULT		
	if CSP_DEFAULT = 0 : CSP_OP = QCSP		
Setpoint tracking:	<Setpoint>_OP	=	<Setpoint>_Q
	<Setpoint>_OPP	=	<Setpoint>_QP

Testing and commissioning sequential control systems

17

17.1 Operator control and monitoring during testing

Overview

Debug functions are integrated in the SFC Editor for the following tasks to provide support for commissioning:

- Monitoring the work process of the sequential control system in the AS
- Influencing the operating modes
- Changing setpoints

Test modes

You can start test mode in either of the following test operating modes:

- Process mode
- Laboratory mode

You can select the test operating mode while in edit mode using the menu commands in the "Debug" menu. You cannot switch once you are in test mode.

In process mode, the communication for the online dynamic display of the SFCs and SFC instances is restricted to keep the load on the CP and bus to a minimum. If overload occurs in this test operating mode, a message is displayed indicating that the limit of the bus load has been reached. In this case, stop the testing of the SFCs that are not absolutely necessary for the test.

In laboratory mode, communication for the online dynamic updates of the SFCs is unrestricted. Use laboratory mode to perform convenient and efficient testing and commissioning.

Setting the test environment

With the menu command **Debug > Test Settings...**, you can open a dialog box in which you can change the monitoring cycle for the current program (default: 2 s).

The cycle time is uniformly stored with the CFC in the chart folder. This means that the same monitoring cycle is valid for both applications (SFC and CFC).

Note

If the cycle time is changed in the SFC, this affects charts/instances with dynamic properties. A change in the CFC affects only newly registered SFCs/instances.

Requirements

- The sequential control system (chart or instance) to be tested including the necessary basic automation functions (CFCs) have been compiled without errors and downloaded to the CPU.
- The chart is open in SFC or SIMATIC Manager, the SFC instance is open in the CFC.

Note

With S7, loading into the CPU—and working in test mode—are functions that are logged if the SIMATIC Logon Service is installed and the access protection and change log are activated.

You can find additional Information on this in the CFC documentation under "Change log and ES log".

Activating test mode

Click the following icon in the toolbar or select the menu command **Test > Test Mode**.



You change from edit mode to test mode.

You can switch back to edit mode at any time during testing.

Note

Make sure that the sequential control system is not waiting for operator input when you switch to edit mode.

The mode switch always relates to the currently active SFC. The overview display of this SFC is dynamically and cyclically updated.

After you have switched to test mode, the current state of the sequential control system is displayed. This does not mean that a sequential control system that has already started can necessarily be monitored or controlled from the start. For example, this is the case with sequential control systems that are started immediately after they are downloaded to the AS without requiring any operator command (autostart).

Note

If the H CPU is being operated in solo mode, for example, due to CPU failure, and a CPU failover has taken place, if there is online access (here: Activate Test Mode), a selection dialog box is displayed. In this dialog box, you can select the required CPU. In redundant operation, this dialog box does not appear.

Note

Update ("F5" key) not possible in test mode.

Operator control and monitoring

You operate and monitor the sequential control system in the overview display of the SFC. Here, you can change the operating states, operating modes, step control modes, and the execution options as required.

Open the properties if you wish to see and/or modify the values of the individual steps.

If you double-click on a step or a transition, a dialog box appears similar to the object properties dialog box in edit mode.

The actions of the steps can be viewed in two different modes, which means the tabs exist twice. In normal view, the interconnecting information is displayed, in the additional view the OS comment is also shown. In the Object Properties of the transition, you will see in the formulated condition in the "Current Cond." tab and the OS comment of the current condition in the "OS Comment" tab.

You can also open the object properties for the step and transition at the same time. To do this, select the required transition and double-click a step to open both dialog boxes (or in the opposite order: select a step and double-click a transition). The step and transition do not need to be associated in order to be able to see both dialog boxes at the same time.

A selected element in the sequencer is indicated by a blue background.

Calculation:

If a calculation is configured in a statement line, the so-called description of the calculation is displayed in the address field and the current value of the calculation result is displayed in the output field. If you want to influence the calculation result, this is only possible in the "Calculation in SFC" dialog box, which is opened with the "f(x)" button.

Tooltips for steps and transitions

Instead of opening the object properties to get specific information, you can also do the following:

- Step: If you position the mouse pointer on a step, the name, number, run times, comments and acknowledgement information are displayed.
- Transition: If you position the mouse pointer on a transition, the name, number, run times and comments are displayed.

Acknowledging in the sequencer and in the object properties dialog box

If a button is displayed for operator prompt acknowledgement or error acknowledgement for the monitored step or transition in the sequencer, the object properties dialog box also has the corresponding button or buttons added to it.

In case of a step runtime error, the step is returned to the state which it had before the error occurred (active = "green", for example) after the error has been acknowledged.

Acknowledgement information

In the object properties of the step you can assign parameters for acknowledgement information. In step control mode "Step-specific confirmation by operator (T/T and O)", this acknowledgement information is displayed at the associated acknowledgement button for an operator prompt. The text can be positioned as required using the mouse pointer, it remains connected with the button over a connecting line.

Changes during testing (SFC only, not SFC instance)

The step attributes (confirmation, minimum/maximum run time, constant in assignments) and transition attributes (constant in conditions) that can be modified in test mode are adopted in the AS and in the ES data management when the change is made and do not require recompilation and download.

Note

If you want to modify existing SFC instances, you must exit test mode and then change the corresponding SFC type. After compiling and downloading the changes, all the instances are automatically adapted.

Changing characteristics during testing (SFC instance only)

The "Control strategy" and "Setpoint values" characteristics can be changed as described in the section: Changing the control strategy and setpoints for an SFC instance (Page 262).

You can influence the "process values", "parameters", "timers" and "block contacts" characteristics in test mode of the CFC at the corresponding I/Os of the SFC instance if the I/Os are not already interconnected with blocks.

The "control values" and "bit memory" characteristics cannot be modified in test mode.

Trace

In test mode, you can use the **Debug > Trace** menu command to specify that the active sequencer is always displayed automatically. If this menu command is not set, the explicitly selected sequencer is displayed.

Note

When tracing is activated, you can only open the object properties of a step or transition that is currently active. If you open object properties of a step or transition that is not active, a dialog box appears asking if you wish to disable the tracing.

Testing a single sequencer

If you cannot test a single sequencer without its functions affecting other sequencer, you can use the following procedure:

1. Copy the sequencer to a separate SFC or SFC type.
2. Adapt the start conditions (for example, RUN = TRUE).
3. If the sequencer is in an SFC type: Generate an SFC instance.
After starting, the sequencer is executed immediately.
4. After making any necessary corrections and testing again, copy the sequencer back to the original SFC or SFC type.

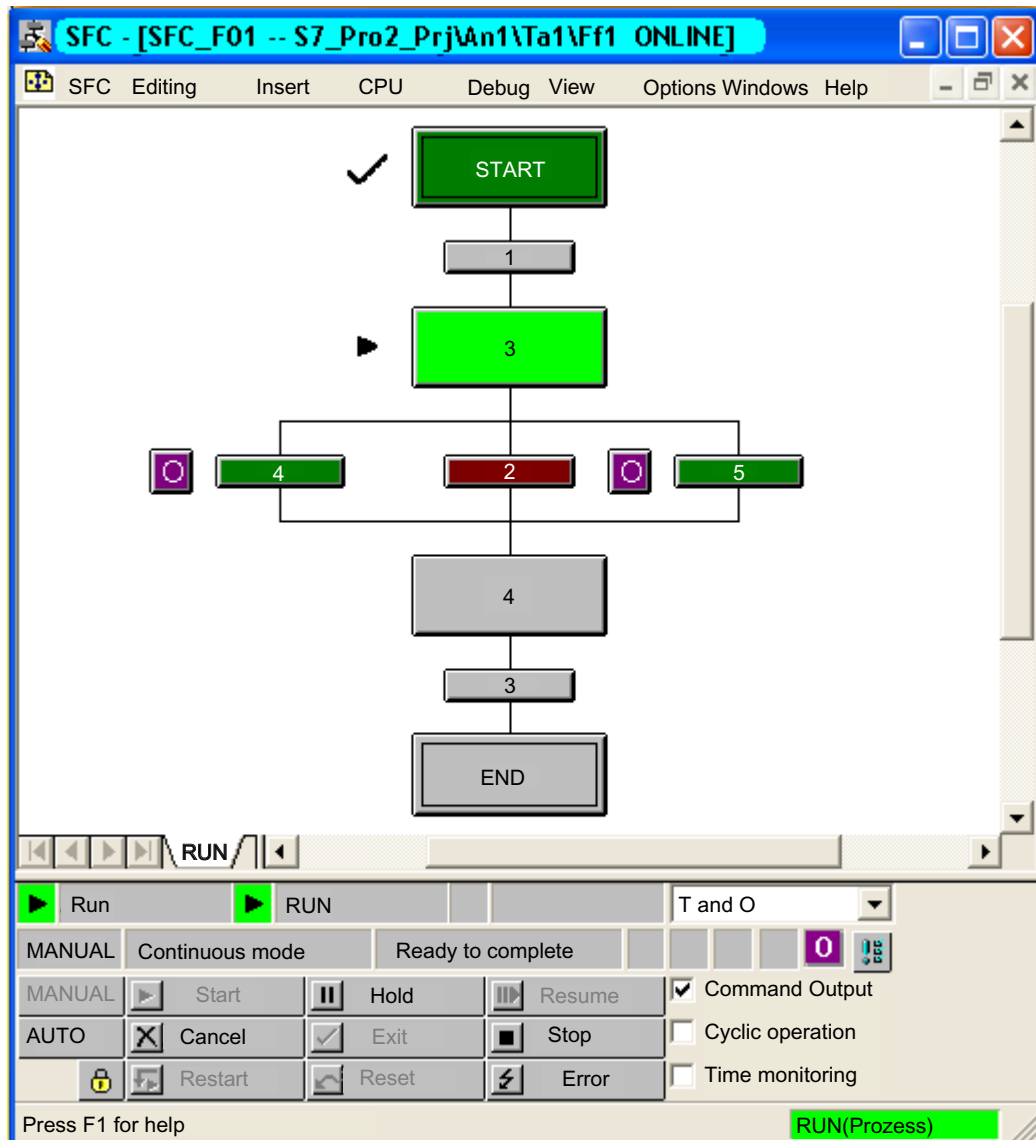
You can find additional information about test mode in the sections:

- Display in test mode (Page 270)
- Properties of a step during testing (Page 276)
- Properties of a transition during testing (Page 279)
- Properties of a sequencer during testing (Page 282)

17.2 Display in test mode

Layout of the SFC window

The window of the SFC or SFC instance in test mode has an extra control and display section at the bottom edge of the window compared to edit mode. All operator inputs relate to the chart or the instance (not to the sequencer).



The following elements are shown in the control and display section (from left to right):

- First row:
 - Symbol and name of the SFC mode
 - Status display and name of the active sequencer
 - Status display and name of the held sequencer
 - Box with drop-down list for displaying and changing the step control modes
- Second row:
 - The current operating mode (MANUAL/AUTO)
 - Status flag "Continuous mode" for bumpless switching to AUTO mode (for example, to avoid an SFC having to be switched off before it is restarted). This is displayed when output QCONT = 1.
 - Status display "Ready to complete", when the SFC is not self-completing (SELFCOMP = 0) and is waiting in the run state for the "Complete" command (display, when READY_TC=1).

The display		at a CPU restart with consistent data
		or
The display		at a CPU restart with inconsistent data
		for an interconnection error (or empty box)
		for an operator error (or empty box)
		for a step error (or empty box)
The button		for an operator prompt (or empty box)
		for the group acknowledgment

- The buttons:
 - For selecting "MANUAL" or "AUTO" mode
 - For enabling changeover to "AUTO"



After the changeover enable, the icon changes to the state:



- The buttons of the Commands (Page 253):

	Start		Hold		Resume
	Cancel		Exit		Stop
	Restart		Reset		Error

- The check boxes for enabling and disabling the execution options "Command Output", "Cyclic Operation", and "Time Monitoring".

Operator prompt

An operator prompt (not with "T") is displayed by the following button beside the transition icon.



After this button or the subsequent "Group acknowledgment" button is clicked and execution is resumed, the operator prompt is hidden.



When an acknowledgement was configured for the step, this is displayed in the step control mode "Step-specific acknowledgement by operator (T/T and O)" at the associated acknowledgement button. The text can be positioned as required using the mouse pointer, it remains connected with the button over a connecting line.

Runtime error



This button exists only when a runtime error is indicated for the current step with the following symbol:



You click this button (or the "Group acknowledgment" button) to acknowledge the pending errors. After acknowledgment, the appearance of the step returns to the state it had before the error occurred (active = "green", for example).

Display of the states

In test mode of the SFC, both the SFC and the sequencers with their start conditions are made dynamic. The name and status of the SFC and the name and status of the sequencer that is currently being executed are displayed in the title bar of the SFC window. The start condition and the execution of the sequencer are also displayed.

The various states of the sequential control system, the sequencer, the steps and the transitions are visualized by different colors or icons.

In addition to the color frame of the steps, a status display also appears. The icon of the status display is therefore an additional visualization of the current operating state in the chart in case the colors cannot be clearly distinguished. You cannot change the colors of the state indicators.

You will find more information on displaying the states in the sections:
Icons for operating states (Page 275)

States of the steps and transitions (Page 274)

Display of the CPU operating mode

The operating mode of the CPU is displayed in the status bar (right information box): green + RUN, red + STOP. Test mode is also indicated: (Laboratory) or (Process).

Display of the sequencer

If the SFC contains more than one sequencer, the following is displayed based on the setting in the "Debug" menu:

- The active sequencer is always displayed automatically (menu command **Debug > Trace** set)
- or
- The explicitly selected sequencer is displayed (menu command **Debug > Trace** not set).

17.3 States of the steps and transitions

Overview

Step state	Step color	Icon
Inactive, not executed	Gray	
Inactive, executed	Dark green	✓
Run	Light green	▶
Held	Yellow	⏸
Error	Red	⚡
CPU stop (not with "inactive, not executed")	Red	⛔
After CPU restart (consistent data)	Magenta	▶▶▶
After CPU restart (inconsistent data)	Magenta	↶
Transition state	Transition color	
Inactive	Gray	
Fulfilled	Dark green	
Unfulfilled	Dark red	

17.4 Icons for operating states

Overview

Icon	State	Meaning
	Aborting	Processing after Abort command.
	Aborted	Cancellation processing is complete; waiting for Reset or Start command.
	Run	Processing after ending the processing in the "Starting" state.
	Held	Cancellation processing in "Held" is complete; waiting for "Resume", "Abort", or "Stop" command.
	Held (Error)	Error processing is complete no more errors pending; waiting for "Resume", "Abort", or "Stop" command.
	Holding	Processing after "Hold" command.
	Completing	Processing after "Complete" command or after implicit complete.
	Completed	Processing in "Completing" state is complete; waiting for "Reset", "Start", "Abort", or "Stop" command.
 (empty)	Idle	Processing in initial state; waiting for "Start" command.
	Error	Processing after "Error" command.
	Error (Completing)	Processing after "Error" command in "Completing" state.
	Resuming	Processing after command "Resume" or "Start".
	Resuming (Error)	Processing after command "Resume" or "Start".
	Stopped	Processing in "Stopping" state completed, waiting for command "Start" or "Reset" or Abort.
	Starting	Processing after command "Start" or "Restart".
	Stopping	Processing after command "Stop".
Only operating states displayed in bold letters apply to sequencers.		

17.5 Properties of a step during testing

Tabs of the "Properties" dialog

The "Properties" dialog box of the step is divided into various tabs:

- **"General" tab**

The "Name" box has a frame, its color corresponds to the state of the step and is updated continuously. All colors that occur are listed in the following table: Default colors (Page 65). If you select the "Confirmation" check box, you set a flag that is evaluated in the "T/T and C" step control mode (step-specific confirmation by operator). This can only be modified in the SFC. The sequential control system runs as follows:

- **Process-driven** for steps without the "Confirmation" option.
Each fulfilled successor transition of a step without the "Confirmation" option enables the next step without operator intervention (corresponds to "T").
- **Controlled by the operator** for steps with the "Confirmation" option.
When an active step with the "Confirmation" option becomes true, the system prompts the operator to acknowledge the next step and then continues the sequencer (corresponds to "T and O").

If you select the "Target step" check box, the current step is selected as the target step. This is marked in the sequencer with the following symbol to the left of the step:



This option can only be modified for the SFC and the SFC instance if they are not in the sequencer "Run" state.

Setting the target step means that:

- The inactive sequencer starts at the selected target step instead of at the start step when it is next processed.
- The held sequencer resumes at the target step when the continuation follows proper processing of the interrupted steps.

The target step marker is valid only for the next "Start" or "Resume" command. When these commands are executed or when the CPU is restarted, the target step marker is cleared. You can also select several steps as the target step. The user is responsible for selecting the target steps in a proper way so as to ensure their processing without the formation of blockades or loops. You can find detailed information on this under: Processing an SFC (Page 249)

Note

Please note the following:

- If you use "programmed target steps", the target steps sent by the operator in the corresponding sequences are deleted.
- Target steps set in "MANUAL" mode are not deleted when there is a changeover to "AUTO".
- Setting or deleting target steps is possible only when ENTARGETSTEP = 1 is set.

In the "Run times" area, you can change the "Minimum" and "Maximum" parameters for SFCs. If you click in the text box, an additional dialog box opens in which you can enter a new time. Any changes you make are entered in the ES data management when you click "OK" and take effect in the next processing cycle in the AS.

You can use the "Current", "Minimum" and "Maximum" boxes to monitor the run time. If no values were configured for the runtimes (time = 0), the individual boxes display "- - -". You will see the comment of the step in the "Comment" field.

In the "Acknowledgement information" box, you see the text displayed to the operator as information in the mode "Step-specific acknowledgement by the operator (T/T and O)". The information appears in test mode or in the SFC visualization on the corresponding acknowledgement button. The text can be positioned anywhere in the window but remains linked to the button with a connection line.

- **"Actions (technological)" tab**

The technological actions configured in a step of a sequencer of an SFC type are shown in this tab.

This tab is only displayed for the SFC type or an SFC instance, not for an SFC.

Each statement line consists of the following elements:

- Display field with the name of the configured technological action
- "Action (technological)" button



The configured technological action can be opened with this button to display the current process values.

- **"Initialization", "Processing", "Termination" tabs**

On the left next to the first address there is a box displaying the current value of the address. The box on the right next to the second address contains the configured value that you can change (only in the SFC). If you click in the box, you open the "Change Value" dialog box, in which you can enter the new value.

After you close the dialog box, the changed value is written to the ES data management (and in the CPU) and becomes effective in the next processing cycle.

- **Register "OS comment" (initialization), (processing), (termination)**

In these registers you can see the initialization, processing, or termination actions. The middle column does not show the interconnection information, however, as it does in the Initialization, Processing and Termination tabs. Here it shows the configured OS comment. All other details are the same.

Buttons

If a timeout occurs during time monitoring of the step, the following button for acknowledging the step runtime error is displayed next to the affected step:



The following button is also displayed in the dialog box.



17.5 Properties of a step during testing

This allows you to acknowledge the error from the dialog box.

Note

In case of a step runtime error, the step is returned to the state which it had before the error occurred (active = "green", for example) after the error has been acknowledged.

You can use the "Go To" button of an SFC to jump from the current address box to its point of use, for example, to the block in the CFC or to the I/O address in HW Config (not possible for an SFC instance).

17.6 Properties of a transition during testing

Tabs of the "Properties" dialog

The "Properties" dialog box of the transition is divided into various tabs:

- **General**
The "Name" box has a frame; its color corresponds to the state of the transition and is updated continuously. All colors that occur are listed in the following table: Default colors (Page 65).
- **Curr. cond. (technological)**
The technological condition configured in a transition of a sequencer is shown in this tab.
- **Current Cond.**
This tab shows the current state of the conditions.
- **OS Comment**
This tab shows the current values and the state of the transition logic, as shown on the "Current Cond." tab. The difference to the "Current Cond." tab here is that there is a column with the OS comments instead of columns with the formulated conditions. All other details are the same.
- **Last Cond.**
This tab shows the state of the conditions of the previous processing cycle.
- **Cond. after Error**
This tab shows the state of the conditions that led to an error.

"Curr. cond. (technological)" tab

The technological condition configured in a transition of a sequencer is shown in this tab.

This tab is only displayed for the SFC type or an SFC instance, not for an SFC.

Only a technological condition can be defined in this tab.

The technological condition consists of the following elements:

- Left address field; shows the name of the configured technological condition.
- "Condition (technological)" button



The configured technological condition can be opened via this button to display the current process values.

- Operator field; shows the configured operator.
- Middle address field
- Right address field
- Line that shows the configured technological condition.

Status of the logic operation:

17.6 Properties of a transition during testing

In test mode, the color of the line indicates the binary logic result of the technological condition.

- A green line means "TRUE" (fulfilled).
- A red line means "FALSE" (unfulfilled).
- A black line means "UNDEF" (undefined).

Background colors:

- The background color such as "light green" in a row indicates that a technological condition is configured.
- The background color such as "Turquoise" in a row indicates a forced value.

The colors can be set in the CFC editor.

"Current Cond." tab

There are boxes with the current value of the address on the left next to the first address and on the right next to the second address. In the SFC, you can change the content of both boxes (not possible with an SFC instance). If you click in one of the boxes, the "Change Value" dialog box opens where you can enter the new value for the address.

After you close the dialog box, the changed value is written to the ES data management (and to the CPU) and becomes effective in the next processing cycle.

The results of the Boolean logic operations for the conditions are visualized as colored connecting lines of varying thickness.

- A wide, green line means "fulfilled".
- A thin, red line means "unfulfilled".
- A thin, black line means "inactive".

"Last Cond." and "Cond. after Error" tab

The content of these tabs is not automatically updated; they therefore have an additional "Update" button. The state of the transition is entered here as it was at the time when the object properties were opened. You can display the current state in a permanently open dialog box by clicking "Update". The values of the addresses cannot be changed.

Buttons

If an operator prompt is displayed for the monitored transition in the sequencer, the following button is displayed beside the transition:



The following button is also displayed in the dialog box.



This allows you to acknowledge the operator prompt from the dialog box.

You can use the "Go To" button of an SFC to jump from the current address box to its point of use, for example, to the block in the CFC or to the I/O address in HW Config (not possible for an SFC instance).

17.7 Properties of a sequencer during testing

Tabs of the "Properties" dialog

The "Properties" dialog box of the sequencer is divided into various tabs:

- **General**

The "Name" box has a frame; its color corresponds to the state of the transition and is updated continuously. All colors that occur are listed in the following table: Default colors (Page 65).

The "Comment" box displays the configured comment for this sequencer.

You can see the priority of the sequencer in the "Priority" box. The priority decides which sequencer of a chart is started when the start conditions of several sequencers are met simultaneously.

- **Start condition (technological)**

This tab is only displayed for the SFC type or an SFC instance, not for an SFC.

Only a technological condition can be defined in this tab.

The technological condition consists of the following elements:

- Left address field; shows the name of the configured technological condition.
- "Condition (technological)" button



The configured technological condition can be opened via this button to display the current process values.

- Operator field; shows the configured operator.
- Middle address field
- Right address field
- Line that shows the configured technological condition.

Status of the logic operation:

In test mode, the color of the line indicates the binary logic result of the technological condition.

- A green line means "TRUE" (fulfilled).
- A red line means "FALSE" (unfulfilled).
- A black line means "UNDEF" (undefined).

Background colors:

- The background color such as "light green" in a row indicates that a technological condition is configured.
- The background color such as "Turquoise" in a row indicates a forced value.

The colors can be set in the CFC editor.

- **Start Condition**

Each line represents a condition. Of the maximum 16 conditions, 2 x 5 are visible on the first page and 2 x 3 on the second page. You can move to the second page by clicking "Arrow" on the last operator.

There are boxes with the current value of the address on the left next to the first address and on the right next to the second address. The values can only be changed for the SFC.

- **OS Comments (Start Condition)**

The difference to the "Start Condition" tab here is that there is a column with the OS comments instead of columns with the formulated conditions. All other details are the same.

- **Preprocessing**

This tab contains the actions for preprocessing of the current sequencer. The values can only be changed for the SFC.

Each line represents one statement. Up to 50 statements are possible. The section of displayed statements can be moved with the scroll bar at the right of the window.

Each statement consists of a left address, an operator, and a right address.

There are boxes with the current value of the address on the left next to the first address and on the right next to the second address. If you click in one of the boxes, the "Change Value" dialog box opens where you can enter the new value for the address. After you close the dialog box, the changed value is written to the CPU and takes effect in the next processing cycle. A constant in the right address box is also written to the ES data management.

- **OS Comments (Preprocessing)**

The difference to the "Preprocessing" tab here is that there is a column with the OS comments instead of columns with the formulated conditions. All other details are the same.

- **Postprocessing**

This tab shows the actions for postprocessing of the current sequencer. The values can only be changed for the SFC. The structure of the tab is identical to the "Preprocessing" tab (see relevant section).

- **OS Comments (Postprocessing)**

In contrast to the "Postprocessing" tab, this has a column with the OS comments instead of columns with the formulated conditions. All other details are the same.

17.8 Calculations in steps and transitions during testing

Overview

A calculation in a step or transition that was created in the SFC Editor can be tested in test mode.

The "Calculations in SFC" dialog box is opened for this in test mode with the "f(x)" button in the properties dialog box of a step or transition.

The "f(x)" button in a statement line is only active if a calculation was configured in edit mode. In this case, the statement line contains the name of the calculation and the current value of the calculation result.

Structure of the dialog box

The structure of the dialog box in test mode is the same as in edit mode except that the "Value" column is added to the table.

The current value is displayed and cyclically updated in the "Value" column:

- For the overall result of the calculation
- For all function inputs with Interconnection to addresses
- For intermediate results, that is, function inputs with interconnection to lower-level functions.

Only the cells of the "Value" column can be edited in test mode. All other table columns are not editable.

Changing the value of an address

1. To change a value of the calculation, select the "Change value" menu command in the shortcut menu for the relevant cell.
2. The "Change Value" dialog box opens. You can enter the desired new value in the "New value" field.
3. Clicking the "OK" button closes the dialog box and transfers the new value to the AS and the CFC data management.
4. In the "Calculations in SFC" dialog box, you can now check the effect of the new value in the calculation.

Note

Error handling in the formula

There is no error handling in the formula, such as in case of overflow, division by zero, etc.

Documenting programs

18.1 Documenting SFCs

Overview

The documentation of SFCs/types/instances includes the following:

- Printout of the SFC in a variety of layouts
- Parameter assignments
- Properties
- Chart reference data

Specifying the page layout

If you want to change the layout, proceed as follows:

1. Select the menu command **SFC > Page Setup....**
A dialog box opens.
2. Select the paper format (e.g. "A4", "A4 with margin") in the drop-down list.

Previewing the printout

1. Select the menu command **SFC > Print Preview...**
The pages to be printed are displayed on the screen.
2. Check the layout.
You can start the printout from this preview.

Printing an SFC

1. Click the button



in the toolbar or
select the menu command **SFC > Print....**
A dialog box opens.

2. Here, you make settings for the print scope and the layout:
Print What:

- Properties
- Interface inputs/outputs
- External view (for SFC) or characteristics (for SFC type/instance)

Sequencers:

- Properties
- Normal size
- Overview (you can select either normal size or overview, but not both).
- Steps/transitions

Options (only with "Normal size"):

- Alternative branch aligned left
- Comment/text

18.2 Chart reference data

Starting the application

1. Select the menu command **Options > Chart Reference Data...** or click the button



The chart reference data are opened with an empty window.

2. In the "View" menu, select the list containing the information you want to display or click the corresponding icon in the toolbar.
The list with the current reference data opens.

You do not need to close the window with the chart reference data if you want to continue working in SFC. You can view the generated lists while continuing your work in SFC.

Chart reference data

You can display and print the following chart reference data:

Run Sequence

The graphical display shows the entire run sequence of a CPU.

Cross-References Chart Element -> Address

The list displays all shared addresses used in the project and the elements which access them.

Cross-References SFC > Chart Element

The list shows all existing accesses from SFCs to the inputs/outputs of CFC elements.

Cross-References Chart Element > Runtime Group

The list displays all points at which CFC and SFCs reference runtime groups.

Block Interconnections

The list displays all block interconnections in the project.

Accesses in SFC Types

The list displays all read and write accesses within an SFC type.

Block Types

The list displays the block types used and the points (CFC) at which they are used.

S7 Resource Allocation

The list shows the assignment between CFC configuration objects and S7 resources.

Local Data

The list displays all the OBs in the program along with the calculated local data requirements and the local data size of the individual priority classes configured offline and actually existing online.

Block Call Hierarchy

The graphical display shows the call hierarchy of all blocks in the current program.

Textual Interconnections

The list shows all textual interconnections (path reference to the interconnection target) with the chart name and chart element of the interconnection source.

Statistics

The graphical display shows the quantity of utilized CFC and SFC objects and S7 resources, the time stamp of the current program and the process objects of the project.

You can find a detailed description of the lists and the meanings of the columns in the section "Display of chart reference data" of the CFC online help "Chart Reference Data".

Export

You can save the generated lists as a file in CSV format (for use in Microsoft EXCEL, for example) by selecting the menu command **Reference Data > Create Export File....**

18.3 Logs

Saving and Printing Logs

The menu command **Options > Logs...** opens a dialog box with several tabs.

Some tabs are only available when the corresponding functions have been previously executed.

Saving the log file and printing always relates to the currently open tab.

Tabs

"Compile" tab

List of the messages that occurred during compilation (including the compiler messages); for example, if the project contains a program but no station. Example message: "The program is not assigned to an actual CPU".

"Check Consistency" tab

List of the messages that occurred during the consistency check.

"Download" tab

List of the messages that occurred during a download, for example, if the download was free of errors: Example message: "0 errors and 0 warnings found".

The "Archive" button allows you to open a selection dialog for selecting and displaying log files with detailed information about the loading process. If the change log is disabled, the "Archive" button is disabled.

You can find additional information about this in the section "Change log and ES log" in the help or documentation *Process Control System PCS 7; CFC for SIMATIC S7*.

"ES Log" tab

Log of all protected actions (download, test mode). This requires that the SIMATIC Logon Service is installed.

"Step Processing" tab

After checking the SFC runtime behavior with the menu command **Options > Check Step Processing**, this tab lists the steps that use the same address in the terminating action of one step and in the initializing action or processing action of the next step. The log shows how many SFCs were checked and how many accesses were found and the SFCs with an unchanged runtime behavior.

"Convert Format" tab

After converting charts of older versions to V5.1 or later, the charts that no longer have the same properties are listed here. These include, for example, SFCs that had the attributes "scan rate" and "phase offset" in the old version. The information includes the insertion location (task) and the values for the scan rate and phase offset for every chart affected. Example: "SFC1: Task OB32 scan rate 4 phase offset 2".

"Make Textual Interconnections" tab

The menu command **Options > Make Textual Interconnections** is used to make all textual interconnections that are assigned to a specific interconnection partner in the current chart folder. The textual interconnections made with this action are displayed along with error messages indicating the textual interconnections that could not be made for one reason or another.

"Block types" tab

This tab lists the messages that occurred when the block types were checked (during import).

18.4 Defining footers

Overview

The menu command **SFC > Footers...** opens a dialog box where you can enter the text to be displayed in the footers on every printed page.

With the DOCPRO add-on package, you can print the SFC/type together with footer data. A distinction is made between global data and specific data (local data) for footers.

You can enter the global data for the project using DOCPRO or SIMATIC Manager ; the specific data must be entered with the SFC Editor. Remember that the specific data overwrites the entries of the global data for the particular SFC.

You can enter specific data even if you have not installed the DOCPRO add-on package. These data are saved and can be printed later when DOCPRO is available for print jobs.

Specific footer data

You can enter the SFC-specific footer data in the active tabs "Part 1" to "Part 4" and "Available Fields". These include document type, date created, document number, dates of changes or free texts.

Keywords in footers

In the global footers, you can enter keywords that will be replaced by current texts in the printout. Below, you can see the available keywords and their meaning:

Keyword	Designation	Meaning	DOCPRO key
\$\$CN\$\$	Name	Name as entered in the properties.	\$54
\$\$CC\$\$	Comment	Comment text as entered in the properties.	\$60
\$\$A\$\$	Author	Name as entered in the properties.	\$55
\$\$DC\$\$	Date created	Date as entered in the properties.	\$56
\$\$DM\$\$	Last modified	Date as entered in the properties.	\$57
-----	-----	-----	-----
\$\$CH\$\$	Project path	Path, as entered in the properties	
\$\$PP\$\$	Storage location of project	Where the project is physically stored, as entered in the properties.	

Using DOCPRO keys

If you are using DOCPRO V5.1 and create new projects, you can define the keys using DOCPRO keys. This means that you do not have to enter the keywords in SFC in the footers; these are, however, still supported. You still have to make the entries in older projects.

Note

If you print other objects (for example STL blocks or the DOCPRO table of contents) with the keywords used in SFC, the keywords themselves appear in the printout for these objects and not the replacement. This problem no longer exists if you use DOCPRO keys.

When using DOCPRO keys, you must change the standard layout in DOCPRO. You can replace the default footers with specific footers in the "Modify Layout" dialog box. You can find information about this in the DOCPRO online help or in the *DOCPRO manual: Creating Documentation in Conformance with Standards*.

The keys for "Project path" and "Storage location of project" (in the lower part of the table) must continue to be used because there are no DOCPRO keys for them.

Configuring parameter controls

19.1 Parameter control

Parameter control

In addition to fixed sequential control systems, parameter controls are also used in batch processes. Parameter controls are sequential control systems with variable parameters.

These variable parameters are compiled in a shared data block (recipe data block). You can give the recipe data block a symbolic name, for example, "RecParDB".

The variables of the recipe data block are values assigned to the parameters of the basic automation during configuration.

You can find additional information on parameter control in the sections:

How to configure the runtime (Page 294)

Execution with different parameter sets (Page 295)

Example of a recipe data block (Page 297)

19.2 How to configure the runtime

Procedure

You configure the runtime of the parameter control with the SFC Editor. The procedure is similar to the configuration of the sequential control system. You assign the values from the recipe data block to the parameters of the basic automation in the "Object Properties" dialog box.

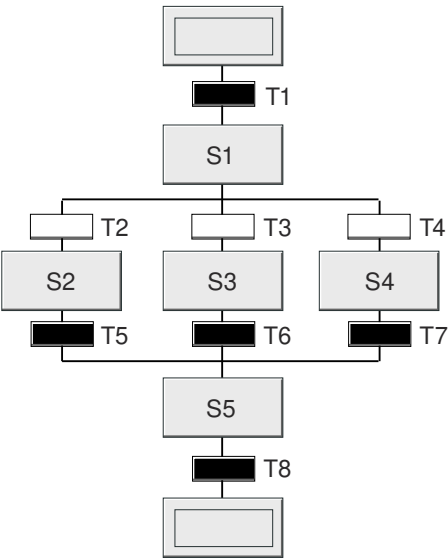
Example: Configuration of a step

The following data is entered in the basic automation (example) from the example recipe data block (Page 297) "RecParDB" in the "Object Properties" dialog for the step:

```
ctrlr_1.w := "RecParDB".etmp
ctrlr_4.w := "RecParDB".itmp
```

Example: Configuration of a transition

In the "Object Properties" dialog box for the transition, a recipe parameter is used to specify the sequencer of an alternative branch that is to be executed:



Transition: Querying recipe data

Reg. figure:

Condition in T2	"RezParDB".altzgw	= 1
Condition in T3	"RezParDB".altzgw	= 2
Condition in T4	"RezParDB".altzgw	= 3

19.3 Execution with different parameter sets

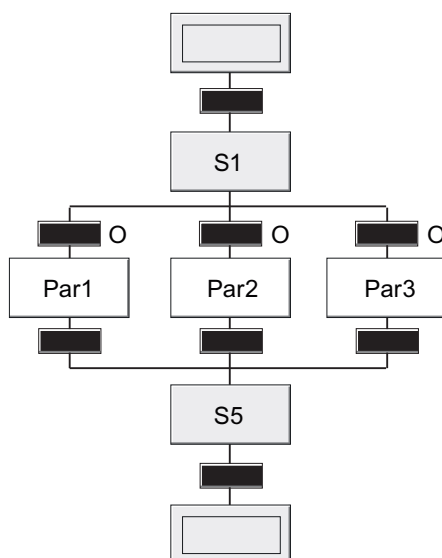
Changing on the OS

If you change the content of the recipe data block, parameter control can also run with different parameter sets. You change the parameters on the OS.

Examples

The examples below show the various ways in which you can change the content of the recipe data block.

- You can integrate the variables of the recipe data block as modifiable variables in the OS process pictures and specify the current values of the variable on the OS before starting the parameter control.
- You store various recipe parameter sets in the AS. The alternative statement of a recipe parameter set for the recipe data block can result in an alternative branch in the "T and O or "O" step control mode.

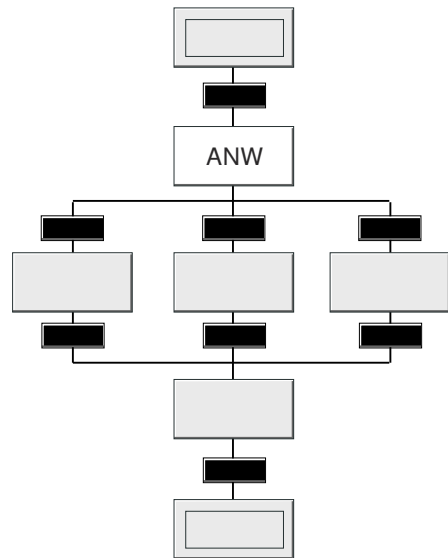


Step: Alternative application of recipe data from different data blocks

Reg. figure:

Instruction in Par1	"RezParDB".chbez	:=	"RezParDB1".chbez
	"RezParDB".ansmng	:=	"RezParDB1".ansmng
Instruction in Par2	"RezParDB".chbez	:=	"RezParDB2".chbez
	"RezParDB".ansmng	:=	"RezParDB2".ansmng
Instruction in Par3	"RezParDB".chbez	:=	"RezParDB3".chbez
	"RezParDB".ansmng	:=	"RezParDB3".ansmng

- In the OS process picture, you have included the modifiable variables "Batch name" and "Amount". By calling an appropriate user block in the parameter control, the parameters in the recipe data block "RecParDB" are adapted relative to a standard batch.



Step: Activate a user block for adapting amounts

Reg. figure:

Instruction in ANW	ANW.EN	:= 1
--------------------	--------	------

19.4 Example of a recipe data block

Recipe Block "RecParDB"

The table contains an example of a "RecParDB" recipe block.

Tag Name	Type	Value	Designation
	STRUCT		
Bantam	STRING[8]	'ch_4711'	Batch name
vol	INT	2000	Amount in liters
reapr	INT	3	Reactor pressure in bar
etmp	INT	90	External temp in deg. C
itmp	INT	125	Internal temp. in deg. C
tott	INT	110	Total reaction time in min.
stvol	INT	1000	Standard amount in liters
altseq	INT	3	Alternative branch
	END_STRUCT		

Tips & Tricks

20.1 Configuring SFC calls

Tasks

You can use the SFC to perform the following tasks:

- Configuring hierarchical SFC calls
or
- Coordinating processing of multiple SFCs in the automation system

Configuring hierarchical SFC calls

You configure hierarchical SFC calls in the SFC coordination chart ("SFC Coord" chart controls "SFC Slave1" and "SFC Slave2") as follows:

- Configure the preceding transition to the next step (to coordinate SFC Coord with SFC slave1; in other words, SFC Coord waits for the termination of SFC Slave1, if this is still running).
Enter the condition in the "Properties" dialog box of the transition:
SFC Slave1.BUSY = OFF
- Configure the step for activating SFC Slave1.
Enter the statement in the "Initialization" tab of the "Properties" dialog box for the step:
the step: SFC Slave1.INTONOFF := ON
- Configure the successor transition for the step above (to coordinate SFC Coord with SFC Slave1, in other words SFC Coord waits for termination of SFC Slave1).
Enter the condition in the "Properties" dialog box of the transition:
SFC Slave1.BUSY = OFF

SFC Slave2 is now controlled by SFC Coord.

No special configuration is necessary for SFC Slave1 and SFC Slave2.

Coordinating the SFCs

You coordinate multiple SFCs (SFC1 and SFC2) using data cells (such as bit memory, data block elements) that are set in SFC1 (statement in a step) and read in SFC2 to enable the next step (condition in a transition).

20.2 Converting older projects

Using the new SFC runtime system

Before the new functions of the SFC runtime system can be used, the data is converted the first time you write to projects of an earlier version and the FB 300 SFC runtime system is replaced following a prompt for confirmation.

You then compile and download the program as follows:

1. Select the menu command **SFC > Compile...**
The program is compiled.
2. Select the menu command **CPU > Download...** and then "Download: Changes" with the automation system in RUN (or STOP) mode.
The program is downloaded. Switch off all active charts in the corresponding dialog box.
3. Switch all active charts back on in the corresponding dialog box.

For additional information on converting older projects to SFC of the current version, refer to the manuals for *PCS 7 Software Updates*.

SFC control block

When V5.x ES data are converted to $\geq$ V6.x ES data, the SFC control block (SFC_CTRL) is eliminated and replaced by the external view of the SFC. All the positioned SFC control blocks are deleted and replaced in each case by the external view of the SFC that displays the interface of the SFC as a block. The settings of the SFC control block parameters and its interconnections are applied to the external view. The runtime behavior of the SFC is not changed.

The position in the run sequence is now also no longer important (in the past it was necessary to ensure that SFC_CTRL was installed before the SFC in the run sequence).

Control strategies and/or setpoint changes for projects created with SFC < V6.1 SP1

To make control strategy and/or setpoint changes effective, the block FB 245 must be copied from the SFC library to the block folder prior to the change followed by compilation of the entire program and download of the changes.

You can find additional information on this in the section:
Changing the control strategy and setpoints for an SFC instance (Page 262)

System attributes on standard interface

When older projects are converted, the system attributes of the block connections of the standard interface are transferred from FB247 or FB300 of the current SFC Library. During this process, the attributes for the block connections of SFCs or block connections of the standard interface of SFC types / SFC instances that have been changed by the user are lost. In block connections, attributes of parameters that were created by the characteristics editor or interface editor are retained.

Possible procedure when you want to accept attribute changes at the standard interface during conversion:

- SFC type
In FB247 (@SFC_TYPTemplate) in the current SFC Library, change the attributes that were changed on the standard interface of SFC types.
Note that this changes the attributes for all SFC types and SFC instances following the conversion.
- SFC
In FB300 (@SFC_RTS) in the current SFC Library, change the attributes that were changed on the standard interface of SFCs.
Note that this changes the attributes for all SFCs following the conversion.

Note

You must change the attributes again after every installation of SFC for SIMATIC S7.

Appendix

21.1 Allocation request and allocation status

Introduction

With equipment phases, the lower-level equipment modules are not permanently assigned, but instead requested at runtime, allocated and then released again.

The allocation characteristics are configured using an "allocation request" in the step of the SFC.

The current status of allocation at runtime is shown with the "Allocation status".

You can find additional information about this in the documentation *Process Control System PCS 7; CFC for SIMATIC S7* in the section "Configuring the allocation of an equipment phase".

Allocation request and allocation status

- The allocation requirements are defined in the master data library under "Shared Declarations > Enumerations > ES_AcquireTypes".
The following table describes the available allocation requests sorted by their priorities.
The lowest priority is at the top.

Allocation request (Type according to ES_AcquireTypes)	Priority	Network share	Description
No request (NoRequest)	0	-	Default value of the allocation variable Is also used to withdraw an allocation request.
Preliminary (wait) (PrelimWait)	1	Implicit	Similar to "Wait", only with implicit release in a subsequent, higher-level allocation request.
Preliminary (now) (PrelimNow)	2	Implicit	Allocation must be immediately successful or it fails. Implicit release for a subsequent, higher-level allocation request.
Wait (Wait)	3	Explicit	If no immediate allocation is possible, there is a wait for the release. If the passed wait time is zero, the wait may last indefinitely. This allocation request fails when a wait time greater than zero has been configured and this time has expired. Explicit release required.
Immediate (Now)	4	Explicit	Allocation must be immediately successful or it fails. Explicit release required.
Takeover (TakeOver)	5	Implicit	With the same allocation identifier (for example, unit name), an immediate change of ownership takes place. With regard to errors and wait time; otherwise like the "Wait" type.
Force (safe) (ForceSafe)	6	Explicit	Once the equipment module (EM) to be allocated is at an end position (IDLE, ABORTED, etc. – ORing possible as well) that can be determined by parameter, it changes ownership.
Force (now) (ForceNow)	7	Explicit	Immediate, unconditional change of ownership by the requesting object.

- The allocation states (plural) are defined in the master data library under "Shared Declarations > Enumerations > ES_AcquireStates".
The following table describes the available allocation states.

21.1 Allocation request and allocation status

Allocation status (identifier in accordance with ES_AcquireStates)	Comment
Idle	Free Equipment module (EM) is available.
Owner	Owner Allocation has succeeded.
Waiting	Waiting Request was accepted, but equipment module (EM) is not yet available.
NotAvailable	Not available Equipment module (EM) is not available at the time of the request. Can only occur with the allocation requests "Now (Now)" and "Preliminary (now) (PrelimNow)".
Deactivated	Disabled The requested equipment module (EM) can generally not be allocated according to its own information.
Timeout	Maximum wait time has expired
Snatched	"Snatched" Equipment module (EM) has been forcibly taken over by another location.
ConfigError	Configuration error Only applies to the allocation type "Force (safe) (ForceSafe)". Occurs when the description masks for a safe state do not match, for example, bits should be included per OR logic operation and simultaneously excluded per AND_NOT logic operation.

21.2 Data types

21.2.1 BOOL, BO

BOOL data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
BO	BOOL; logical number	0-1	0; 1; False; True	FALSE; TRUE; FALSE; TRUE

21.2.2 BYTE, BY

BYTE data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
BY	BYTE; Sequence of 8 bits	0 ... 255, (0 ... FF)	28; 16#2a; 10#123; 2#10110011	16#1C; 16#2A; 16#7B 16#B3

21.2.3 CHAR, C

CHAR data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
C	CHAR; individual character	Depending on the configured Windows character set, "Single byte" or "Multibyte"	'A'; 'a'; '0'; '%';	'A'; 'a'; '0'; '%'

21.2.4 DATE, D

DATE data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
D	DATE; Date	1990-1-1 ... 2168-12-31	1996-04-29	D#1996-04-29

21.2.5 DINT, DI

DINT data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
DI	DINT; double integer	-2147483648 ... 2147483647	12345; -17385267; 16#3BC9; 10#123456789; 2#1011010101010101	12345; -17385267; 15305; 123456789 46421

21.2.6 DWORD, DW

DWORD data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
DW	DWORD; string of 32 bits	0 ... 4294967295, (0 ... FFFFFFFF)	1234567689; 16#1a2b3c4d; 10#1234567890; 2#10010010010010010010	16#499602D2; 16#1A2B3C4D 16#499602D2 16#92492

21.2.7 INT, I

INT data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
I	INT; integer	-32768 ... 32767	4099; -30123; 16#1AC5; 10#12345; 2#0010110010101110	4099; -30123; 6853; 12345; 11438

21.2.8 REAL, R

REAL data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
R	REAL; floating-point number	-3.40282e+38 ... -1.1755e-38 ... 0 ... 1.1755e-38 ... 3.40282e+38	22.78; -1234522.456789; -3.456e-3; 2.573e19	2.278e1; -1.2345224e6; -3.456e-3; 2.573e19

21.2.9 S5TIME, T5

S5TIME data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
T5	S5TIME; duration in S5 format	0h_0m_0s to 2h_46m_30s; 0 ... 9990ms in 10ms-, 100ms to 99900ms in 100ms- 1s to 999s in 1s- and 10s to 9990s in 10s steps	1h_30m_0s; 1234567ms; 2h; 32m_5s	T#1h_30m; T#20m_34s_567ms T#2h; T#32m_5s

Note

Accesses to S5TIME addresses in transitions are not possible.

21.2.10 STRING, S

STRING data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
W	STRING; character string	Depending of the configured Windows character set, "Single byte" or "Multibyte" (at least 127 characters)	'Charge_127';	'Charge_127';

Note

The STRING data type is supported as an address in the statements of the steps. Both constant STRINGS (in quotation marks) and inputs and outputs of blocks (FB instances, shared DBs) are possible as addresses, for example:

- CFC\block.inpstring := 'conststring'
- CFC\block.inpstring := CFC\block.outstring

The following restrictions apply:

- Accesses to FB in/out parameters (IN_OUT) of the STRING type are not possible.
- Accesses to FC inputs, FC outputs, FC in/out parameters of the STRING type are not possible.
- Accesses to STRING addresses are not possible in transitions.
- Constant STRINGS cannot be modified in test mode. They can be modified in the SFC Editor and downloaded to the AS after compiling.

21.2.11 STRUCT, ST

STRUCT data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
ST	STRUCT;	--.	--	Name and value of the first elementary data type

21.2.12 TIME OF DAY, T

TIME OF DAY data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
T	TIME_OF_DAY or TOD; time of day	0:0:0.0 ... 23:59:59.999	12:45:18.012	TOD#12:45:18.012

21.2.13 TIME, TI

TIME data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
TI	TIME; duration	... 24d_20h_31m_23s_647ms (... 2147483647ms)	12d_12h_12m_34s_789ms; 123456789ms; 2h;	T#12d_12h_12m_34s; T#3h_25m_45s_678ms T#2h;

21.2.14 WORD, W

WORD data type

Abb.	Keyword, Type	Value range from ... to	Sample entries	Display in the chart
W	WORD; string of 16 bits	0 ... 65535, (0 ... FFFF)	16#bAc1; 10#12345 2#1000011101011010	16#BAC1; 16#3039 16#875A

Index

A

- Absolute addressing, 149
- Accessing
 - Help with keyboard, 64
 - To structures, 142
- Adapting, 89, 91, 95
 - Chart properties, 89
 - Instance properties, 95
 - Type properties, 91
- Addresses, 139
 - Valid entries, 139
- Addressing, 149, 150
 - Absolute, 149
 - Symbolic, 150
- ALARM_8P, 97
- Alternative branch, 45
 - Creating, 119
 - in the AS, 259
- Attributes
 - for characteristics, 107

B

- BATCH parameters, 205
 - SFC inputs/outputs, 205
- Bit memory, 100
- Block contacts, 100, 104, 105
- Block I/O
 - Filtering, 141
- Blocks, 221
 - Generated during compilation, 221
- BO, 306
- BOOL, 306
- BY, 306
- BYTE, 306

C

- C, 306
- Calculation for SFC
 - Compatibility of the data types, 153
 - Configuring a new calculation, 155
 - Configuring in SFC type, 157
 - Configuring in the SFC, 157
 - Data type specification, 154
 - Delete calculation, 153

- Error handling, 152
- Functions and data types, 158
- Implementation of a formula, 160
- Name of the calculation, 153
- Nesting depth of the functions, 154
- Overview, 151
- Printing, 152
- Test mode, 284
- Changes to the configuration, 75, 79, 82
 - of the SFC, 75
 - of the SFC instance, 82
 - of the SFC type, 79
- Changes to the interface, 79
- Changes to the topology, 80
- Changing, 262
 - Control strategies subsequently, 102
 - Display size in the SFC, 52
 - the control strategy and setpoints for an SFC instance, 262
 - Window section, 64
- CHAR, 306
- Character string, 309
- Characteristic
 - Attributes, 107
 - Bit memory, 193
 - Block contacts, 105, 194
 - Control strategies, 102, 181
 - Control values, 192
 - Editor, 99
 - Inputs/outputs, 104
 - Note texts, 182
 - Parameters, 192
 - Position texts, 182
 - Process values, 192
 - Setpoints, 183
 - SFC type, 100
 - Timers, 193
- Characteristics interface expansions [SFC inputs/outputs], 183
- Characteristics interface parameter assignments, 181
- Chart properties, 89
 - Adapting, 89
- Chart reference data, 287
 - Documenting, 287
- Commands, 196
 - During execution in the AS, 253
 - Priorities, 253
 - SFC inputs/outputs, 196
- Commands and operating states, 196

- Commissioning, 265, 276, 279, 282
- Common features, 37
 - SFC and SFC type, 37
- Comparison, 37
 - SFC and SFC type, 37
- Compiling, 219, 220
 - Settings, 219
 - SFC, 217
 - SFC instance, 217
 - SFC type, 217
- Configuration, 74, 77, 80, 299
 - SFC, 74
 - SFC calls, 299
 - SFC instance, 80
 - SFC type, 77
- Configuration limits of the SFC, 32
- Configuring, 69, 97, 111, 112, 125, 126, 127, 128, 129, 130, 137, 149, 162, 163
 - Messages in the SFC, 97
 - Multiple sequencers, 113
 - Note text, 103
 - Position text, 103
 - Sequential control system, 111, 112, 116, 117, 119, 121, 122, 125, 126, 127, 128, 129, 130, 131, 132, 134, 137, 144, 146, 148, 149, 162, 163
 - SFC, 69
- Configuring in the network, 51
- Consequence
 - of 16 bits, 310
 - of 32 bits, 307
 - of 8 bits, 306
- Consistency check, 223
- Continuous, 243
- Continuous mode, 206, 243
 - SFC inputs/outputs, 206
- Control strategies, 100, 181
 - Changing subsequently, 102
- Control strategies and setpoints, 262
 - Changes in an SFC instance, 262
- Control strategy and setpoints, 264
 - Tracking, 264
- Control values, 100
- Control words, 210
 - BA_CONTROL, 210
 - SFC inputs/outputs, 210
 - SFC_CONTROL, 210
- Conversion, 300
 - from older projects, 300
- Coordination, 249
 - Sequencer, 249
- Copy, 127
 - SFC elements, 127

- Copying, 85, 86, 128
 - Object properties, 128
 - SFC, 85
 - SFC instance, 86
 - SFC type, 86
- Creating, 23, 26, 28, 67, 69, 71, 72, 73, 112
 - Alternative branch, 119
 - Jump, 121
 - Loop, 119
 - Project structure, 67
 - Sequence, 117
 - Sequencer topology, 112
 - SFC, 23, 69, 71
 - SFC instance, 28, 73
 - SFC type, 26, 72
 - Simultaneous branch, 117
 - Text field, 122

D

- D, 307
- Data backup, 66
- Data from sequencers and steps to be processed (SFC inputs/outputs), 204
- Data type
 - BOOL, 306
 - BYTE, 306
 - CHAR, 306
 - DATE, 307
 - DINT, 307
 - DWORD, 307
 - INT, 308
 - REAL, 308
 - S5TIME, 308
 - STRING, 309
 - STRUCT, 309
 - TIME, 310
 - TIME OF DAY, 310
 - WORD, 310
- Data types, 151
 - for SFC, 151
 - Valid, 151
- Date, 307
- DATE, 307
- Default
 - Colors, 65
- Default colors, 65
- Deleting, 88, 130
 - SFC, 88
 - SFC element, 130
 - SFC instance, 88
 - SFC type, 88

DI, 307
 Diagram
 of the state changes for sequencer OSL, 248
 of the state changes for SFC OSL, 245
 Dialog boxes
 Layout, 56
 Operator control, 62
 Differences, 37
 SFC and SFC type, 37
 DINT, 307
 Display
 in test mode, 270
 Display size in the SFC
 Changing, 52
 Documentation, 285, 287, 291
 Chart reference data, 287
 Footers, 291
 Logs, 289
 SFCs, 285
 Documenting SFCs, 285
 Double integer, 307
 Download, 225
 Download of changes in RUN, 227
 Programs, 225
 Downloading, 219
 Settings, 219
 DW, 307
 DWORD, 307

E

Editing, 162
 Jump, 121
 Loop, 119
 Properties of the sequencer, 131
 Run sequence, 162
 Step, 132
 Text field, 122
 Transition, 144
 Editing steps/transitions, 126
 Selecting, 126
 Elements, 53
 User interface, 53
 Example, 297
 Recipe data block, 297
 Execution options, 202, 236
 SFC inputs/outputs, 202
 External view, 37
 SFC, 37

F

Filtering
 Block I/O, 141
 Footer, 291
 Defining, 291

G

Generated blocks, 221
 SFC, 221
 Getting started, 21
 Group acknowledgements, 204
 SFC inputs/outputs, 204
 Group displays, 204
 SFC inputs/outputs, 204

H

Handling, 69, 71
 Charts, 71
 SFC, 69
 Help
 opening with keyboard, 64
 How it works, 294
 Configuring, 294

I

I, 308
 I/O groups, 104
 Icons
 Operating states, 275
 Individual character, 306
 Inputs/outputs, 172, 180
 for characteristics, 104
 SFC, 172, 180
 Inputs/outputs of the standard interface, 174
 SFC type, 174
 Inserting/creating
 SFC elements, 116
 Instance properties, 95
 Adapting, 95
 INT, 308
 Integer, 308
 INTEGER, 308
 Interface editor, 99
 Interface expansions for inputs/outputs, 181

J

Jump, 47
 Creating, 121
 Editing, 121
 in the AS, 260

K

Keyboard, 59

L

Layout
 Dialog box, 56
LI_ERR, 206
Logical number, 306
Loop, 46
 Creating, 119
 Editing, 119
 in the AS, 260

M

MANUAL commands, 253
Meaning and usage of inputs/outputs for setpoints, 189
Memory requirements, 221
Menu
 Edit, 60
 Help, 60
 Operator control, 61
 Options, 60
 Paste, 60
 SFC, 60
 Target system, 60
 Test, 60
 View, 60
 Window, 60
Menu commands
 Shortcuts, 60
Messages, 209
 SFC inputs/outputs, 209
Messages in the SFC, 97
 Configuring, 97
Migration from older projects, 300
Moving, 85, 86, 129
 SFC, 85
 SFC element, 129

 SFC instance, 86
 SFC type, 86
Multiuser engineering, 51

N

Navigating
 in the SFC, 52
Note texts, 100
 Characteristics, 182
 Configuring, 103
NOTIFY, 97

O

Object properties, 128
 Copying, 128
OCCUPIED, 249
OP_ERR, 206
Opening, 83, 84
 SFC instance, 84
 SFC/SFC type, 83
Operating modes, 195, 234
 SFC inputs/outputs, 195
Operating state logic, 246
 for sequencers, 246
Operating state logic for SFC, 239
Operating states, 196
 Icons, 275
 S7, 238
 SFC inputs/outputs, 196
Operator control
 Dialog boxes, 62
 Menu, 61
Operator control and monitoring during testing, 265
Operator input, 58
 With the keyboard, 59
 With the mouse, 58
OSL, 246
 for sequencers, 246
 for SFC, 239
OSL for sequencers, 248
 Diagram, 248
OSL for SFC, 245
 Diagram, 245
Overview, 221
 Blocks generated during compilation, 221

P

- Parameter control, 293, 295
 - Configuring the runtime, 294
 - Example of a recipe data block, 297
 - Execution with different parameter sets, 295
- Parameter sets, 295
 - Different, 295
- Parameters, 100
- Position texts, 100, 103
 - Characteristics, 182
- Postprocessing, 40, 113
- Preprocessing, 40, 113
- Printing
 - Logs, 289
- Priority
 - Commands, 253
- Process values, 100
- Processing, 249
 - Alternative Branch in the AS, 259
 - Jump in the AS, 260
 - Loop in the AS, 260
 - Sequencer, 249
 - Sequencers, 251
 - SFC, 249, 251
 - SFC after a CPU stop, 251
 - Simultaneous Branch in the AS, 258
 - Step and transition in the AS, 256
- Programmed target steps, 196
- Programs, 225
 - Download, 225
- Project structure, 67
 - Creating, 67
- Properties, 279, 282
 - Sequencer during testing, 282
 - Step during testing, 276
 - Transition during testing, 279
- Purpose
 - Note text, 103
 - Position text, 103

Q

- QDIS_START, 206

R

- R, 308
- REAL, 308
- Real number, 308

- Recipe data block, 297
- Reserves, 215
 - SFC inputs/outputs, 215
- Resources needed, 221
- Run sequence, 162
 - Editing, 162
- Runtime attributes, 163
- Runtime behavior, 234
- Runtime phases
 - Step, 256
- Runtime properties, 162
 - SFC, 162

S

- S5 duration, 308
- S5TIME, 308
- Saving
 - Logs, 289
- Saving/printing logs
 - Documenting, 289
 - printing, 289
 - saving, 289
- Selecting, 63, 125, 126
 - Text with keyboard, 63
 - when editing steps/transitions, 126
 - with the keyboard, 126
 - with the lasso, 126
 - with the mouse, 125
- Selection criteria
 - SFC, 17
 - SFC type, 17
- Sequence, 44
 - Creating, 117
- Sequence path elements, 41
- Sequencer, 17, 40, 44, 131, 249
 - Copying/moving, 113
 - Creating, 113
 - Deleting, 113
 - Editing sequencer properties, 131
 - Keyboard shortcuts, 64
 - Processing, 249
- Sequencer OSL, 246
- Sequencer template, 113
- Sequencer topology, 112
 - Creating, 112
- Sequencers, 249

- Sequential control system, 17, 111, 112, 125, 126, 127, 128, 129, 130, 137, 162, 163
 - Configuring, 111, 112, 116, 117, 119, 121, 122, 125, 126, 127, 128, 129, 130, 131, 132, 134, 144, 146, 148, 162, 163
 - Copying, 137
- Sequential control systems in the AS, 233
- Setpoints, 100
- Settings, 219
 - Compiling/downloading, 219
- SFC, 21, 23, 29, 30, 31, 33, 37, 69, 71, 74, 75, 83, 85, 88, 162, 167, 172, 180, 217, 233
 - and other target systems, 31
 - and plant hierarchy, 30
 - Changes to the configuration, 75
 - Compiling, 217
 - Configuration, 74
 - Configuration limits, 32
 - Configuring, 69
 - Copying/moving, 85
 - Creating, 23, 69, 71
 - Deleting, 88
 - External view, 37
 - Handling, 69
 - in the AS, 233, 234, 235, 236, 238, 253, 256, 258, 259, 260
 - in the STEP 7 environment, 29
 - Inputs/outputs, 172, 180
 - Introduction, 17
 - Navigating, 52
 - Opening, 83
 - Operating principle, 21
 - Parameter control, 293
 - Runtime properties, 162
 - Selection criteria, 17
 - Standard interface, 167
 - Standard interface I/Os [alphabetical order], 167
 - Type/instance concept, 77
- SFC and SFC type, 37
 - Compared, 37
- SFC calls, 299
 - Configuration, 299
- SFC Editor, 49, 50
 - start, 50
 - working, 49
- SFC element, 42, 43, 44, 45, 46, 47, 129, 130
 - Deleting, 130
 - Moving, 129
- SFC elements, 127
 - Copying, 127
 - Inserting/creating, 116
- SFC inputs/outputs, 181, 183, 195, 196, 202, 204, 205, 206, 209, 210, 214, 215
 - BATCH parameters, 205
 - Characteristics interface expansions, 183
 - Characteristics interface parameter assignments, 181
 - Commands and operating states, 196
 - Continuous mode, 206
 - Control words, 210
 - Data from sequences and steps to be processed, 204
 - Execution options, 202
 - Group displays + group acknowledgements, 204
 - Messages, 209
 - Operating modes, 195
 - Reserves, 215
 - Status words, 210
 - System parameters, 214
 - Troubleshooting, 206
- SFC instance, 17, 28, 34, 73, 80, 82, 84, 86, 88, 217, 262
 - Changes to the configuration, 82
 - Changing control strategy and setpoints, 262
 - Compiling, 217
 - Configuration, 80
 - Copying/moving, 86
 - Creating, 28, 73
 - Deleting, 88
 - Opening, 84
- SFC OSL, 239, 245
- SFC processing after a CPU stop, 251
- SFC type, 26, 34, 72, 77, 83, 86, 88, 174, 217
 - Compiling, 217
 - Configuration, 77
 - Copying/moving, 86
 - Creating, 26, 72
 - Deleting, 88
 - Inputs/outputs of the standard interface, 174
 - Opening, 83
 - Selection criteria, 17
 - Standard interface, 174
 - Update, 94
- SFC/instance
 - start, 254
- Shared addresses, 149, 150
- Shortcut keys
 - for menu commands, 59
- Shortcuts for menu commands, 60
- Simultaneous branch, 45
 - Creating, 117
 - in the AS, 258
- ST, 309

Standard interface, 167, 174
 of the SFC, 167
 of the SFC type, 174
 Start, 50
 SFC, 254
 SFC Editor, 50
 SFC instance, 254
 Start conditions, 40, 113
 Sequencer, 40
 States
 Steps, 274
 Transitions, 274
 Status words
 BA_STATE, 210
 SFC_STATE, 210
 USTATUS, 210
 VSTATUS, 210
 Step, 42, 137
 Access to structures, 142
 Editing, 132
 Editing actions, 134
 Editing addresses, 137
 Editing general properties, 134
 Runtime phases, 256
 STEP 7, 29
 Step and transition
 in the AS, 256
 Step control modes, 235
 STRING, 309
 STRUCT, 309
 Structure, 309
 Structures, 142
 Symbolic addressing, 150
 System attributes [parameters], 107
 System attributes for parameters, 172
 System parameters, 214
 SFC inputs/outputs, 214

T

T, 310
 T5, 308
 Target step, 249, 276
 Target steps, 196
 Programmed, 196
 TARGETSEQ, 196, 249
 TARGETSTEP, 196, 249
 Test mode
 Display, 270
 Text, 44
 Selecting with keyboard, 63

Text field
 Creating, 122
 Editing, 122
 TI, 310
 TIME, 310
 Time duration, 310
 Time of day, 310
 TIME OF DAY, 310
 Timers, 100
 Tips & Tricks, 299
 Trace, 265
 Tracking, 264
 Control strategies and setpoints, 264
 Transition, 43, 142
 Access to structures, 142
 Edit OS comments, 148
 Editing, 144
 Editing addresses, 146
 Formulating conditions, 146
 Troubleshooting, 206
 SFC inputs/outputs, 206
 Type properties, 91
 Adapting, 91
 Type/instance concept, 77
 of the SFC, 77

U

User interface, 53, 56, 58, 59, 60, 61, 62, 63, 64
 Elements, 53
 Using the keyboard, 59, 61, 62, 63, 64

V

Valid data types, 151
 Valid entries for addresses, 139

W

W, 309, 310
 Window section
 Changing, 64
 WORD, 310
 Working, 49
 with the SFC Editor, 49
 Working with SFC, 21
 principle, 21

