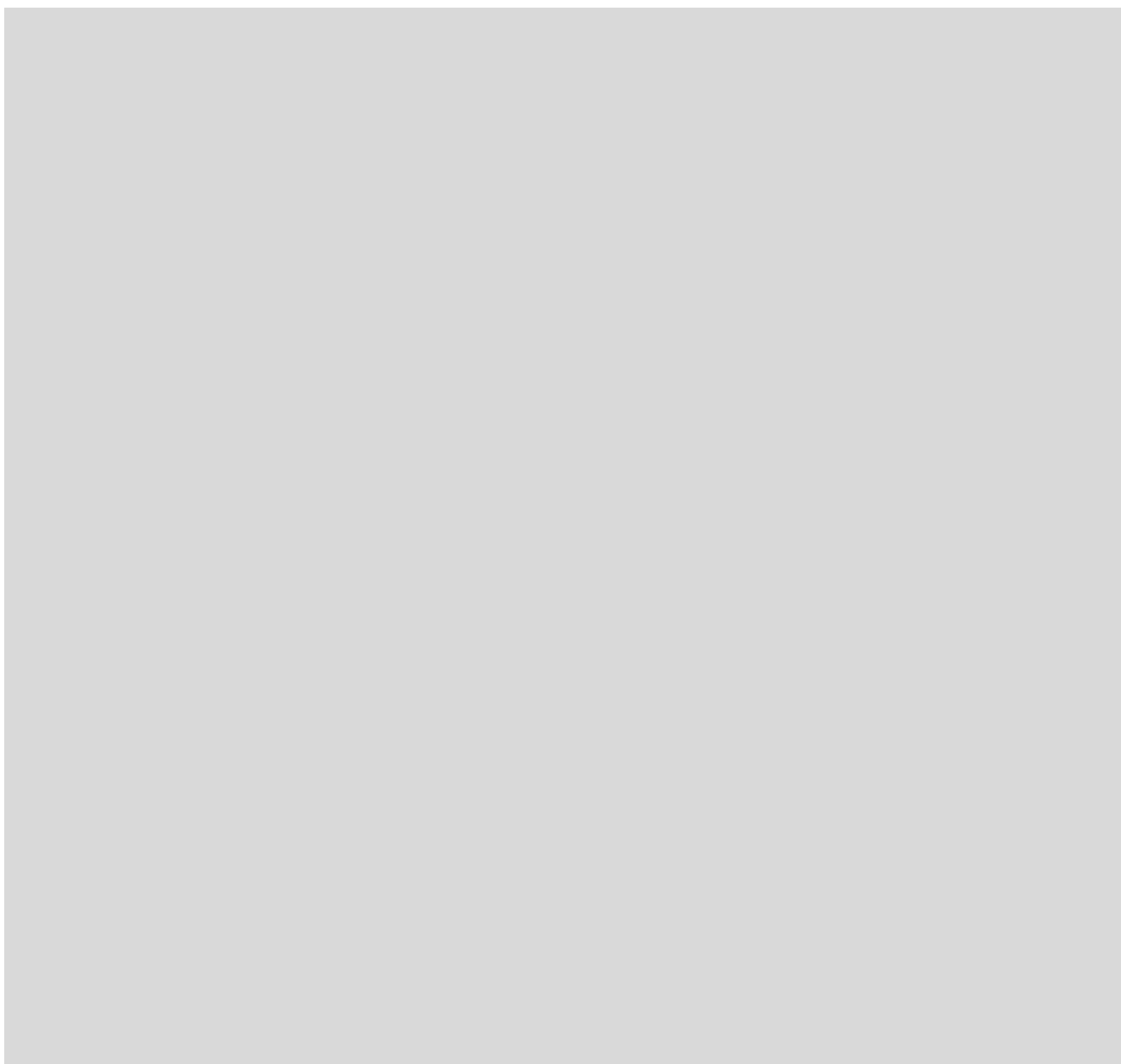


CL200 / CL350 / CL400 / CL500 / ICL700 / PCL

SFC Sequence Function Chart Software Manual



Edition

103

CL200 / CL350 / CL400 / CL500 / ICL700 / PCL

SFC Sequence Function Chart Software Manual

1070 072 186-103 (99.03) GB



Reg. Nr. 16149-03

© 1997–1999

by Robert Bosch GmbH, Erbach / Germany
All rights reserved, including applications for protective rights.
Reproduction or handing over to third parties are subject to our written permission.

Discretionary charge 20.00 DM

Contents

	Page
1	Safety Instructions
1.1	Standard operation 1–1
1.2	Qualified personnel 1–2
1.3	Safety instructions in this manual 1–3
1.4	Safety instructions concerning the described product 1–4
1.5	Documentation, version and trademarks 1–5
2	Introduction 2–1
2.1	Sequence control system structure 2–6
2.1.1	Sequence cascades in steps 2–6
2.1.2	Managing operating modes 2–7
2.1.3	Command output 2–8
2.1.4	Diagnostics and fault indication 2–8
2.2	Graphical display of process sequence 2–9
2.3	Sequence step in MADAP programming mode 2–11
2.4	Sequence step in step marker technique programming mode . . . 2–12
3	Sequence Cascade in MADAP Mode 3–1
3.1	Generating a MADAP-based sequence cascade 3–3
3.2	Specific MADAP markers 3–14
3.3	Specific operating mode bits 3–16
3.4	Step branching 3–19
3.5	Special-purpose cascades 3–20
4	Structural Elements of MADAP Sequence Cascade 4–1
4.1	Linear sequence cascade in MADAP mode 4–2
4.2	Alternative-branched sequence cascade in MADAP mode 4–4
4.3	Simultaneous-branched sequence cascade in MADAP mode . . . 4–6
4.4	Sequence cascade loop in MADAP mode 4–8
4.5	Jump in MADAP mode 4–10
5	Step Markers in Sequence Cascades 5–1
5.1	Generating a sequence cascade with step markers 5–2
6	Structural Elements of Sequences w/ Step Markers 6–1
6.1	Linear sequence cascade in Step Marker technique 6–2
6.2	Alternative-branched sequence cascade in Step Marker technique 6–4
6.3	Simultaneous-branched sequence cascade in Step Marker technique 6–6
6.4	Sequence cascade loop in Step Marker technique 6–8
6.5	Jump in sequence cascade in Step Marker technique 6–10

	Page
A	
A.1	
A.2	
Appendix	
Abbreviations	A-1
Index	A-2

1 Safety Instructions

Before you start working with programs that are programmed with the use of the Sequence Function Chart (SFC), we recommend that you thoroughly familiarize yourself with the contents of this manual. Keep this manual in a place where it is always accessible to all users.

1.1 Standard operation

This instruction manual presents a comprehensive set of instructions and information required for the standard operation of the described products. The described products are used for the creation of sequence cascades.

The products described hereunder

- were developed, manufactured, tested and documented in accordance with the relevant safety standards. In standard operation, and provided that the specifications and safety instructions relating to the project phase, installation and correct operation of the product are followed, there should arise no risk of danger to personnel or property.

Proper transport, handling and storage, placement and installation of the product are indispensable prerequisites for its subsequent flawless service and safe operation.

1.2 Qualified personnel

The relevant requirements of qualified personnel are based on the job specifications as outlined by the ZVEI and VDMA professional associations in Germany. Please refer to the following German-language publication:

Weiterbildung in der Automatisierungstechnik

Hrsg.: ZVEI und VDMA

MaschinenbauVerlag

Postfach 71 08 64

60498 Frankfurt

This instruction manual specifically addresses programmers, system commissioning and maintenance personnel. They require special knowledge about machine controllers.

Interventions in the hardware and software of our products which are not described in this instruction manual may only be performed by specially trained Bosch personnel.

Unqualified interventions in the hardware or software or non-compliance with the warnings listed in this instruction manual or indicated on the product may result in serious personal injury or damage of property.

Installation and maintenance of the products described hereunder is the exclusive domain of trained electricians as per VDE 1000-10 who are familiar with the contents of this manual.

Such personnel are

- those who, being well trained and experienced in their field and familiar with the relevant norms, are able to analyze the jobs being carried out and recognize any hazards which may have arisen.
- those who have acquired the same amount of expert knowledge through years of experience that would normally be acquired through formal technical training.

Please note our comprehensive range of training courses.

Our training center will be pleased to provide you with further information, telephone: +49 (0)6062 78-258.

1.3 Safety instructions in this manual

**DANGEROUS ELECTRICAL CURRENT**

This symbol is used to warn of the presence of **dangerous electrical current**. Insufficient or lacking compliance with these instructions can result in **personal injury**.

**DANGER**

This symbol is used wherever an insufficient or lacking compliance with instructions can result in **personal injury**.

**CAUTION**

This symbol is used wherever an insufficient or lacking compliance with instructions can result in **damage to equipment or data files**.

 This symbol is used to inform the user of special features.

1.4 Safety instructions concerning the described product

**DANGER**

Retrofits or modifications may adversely affect the safety of the products described!

The consequences may include severe injuries, damage to equipment, or environmental hazards. Possible retrofits or modifications to the system using third-party equipment therefore have to be approved by Bosch.

**DANGER**

Danger of life through inadequate EMERGENCY-STOP devices! EMERGENCY-STOP devices must be active and within reach in all system modes. Releasing an EMERGENCY-STOP device must not result in an uncontrolled restart of the system! First check the EMERGENCY-STOP circuit, then switch the system on

**DANGER**

Danger to persons and equipment!
Test every new program before operating the system!

1.5 Documentation, version and trademarks

Documentation

This manual contains information about the Sequence Function Chart (SFC).

Overview of instruction manuals:

Instruction manuals	Language	Order number
KETTE200 Software Module Description	English	1070 072 150
MMI-MADAP for Programmers and Project Designers	English	1070 072 168
BT-MADAP	D	1070 072 096

Special keys or key combinations are enclosed in pointed brackets:

- Special keys, e.g., <Enter>, <PgUp>, , etc.
- Key combinations (simultaneous pressing): e.g. <Ctrl> + <PgUp>, etc.

- ★ This symbol indicates that the description refers to an action that you are required to perform.

Version

- ☞ **This instruction manual applies to the following software version: WinSPS v2.3 and higher, Order number. 1070 077 925, for Microsoft Windows 95/98/NT4.0 operating systems**

Modifications

As this instruction manual has been newly released, there have been no modifications.

Trademarks

All trademarks referring to software that is installed on Bosch products when shipped from the factory represent the property of their respective owners.

At the time of shipment from the factory, all installed software is protected by copyright. Software may therefore be duplicated only with the prior permission of the respective manufacturer or copyright owner.

MS-DOS® and Windows™ are registered trademarks of Microsoft Corporation.

2 Introduction

The present documentation provides a comprehensive overview of procedures required for programming control sequences in Bosch PLC controllers with the use of the Sequence Function Chart (SFC). The sequence function chart creates programs based upon either the MADAP programming mode or step marker technique programming mode.

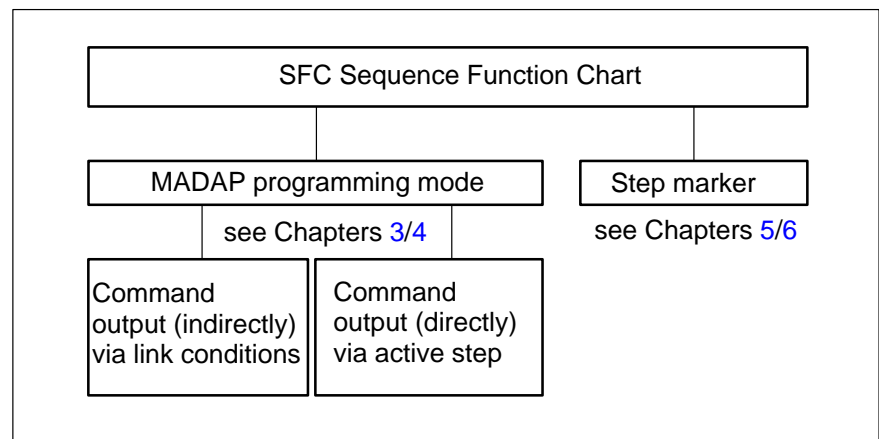
For the MADAP programming mode, the following PLC programme package variants are available:


- MMI-MADAP with MMI operator terminal
- BT-MADAP with BT-type operator terminal


The Help function of the Bosch WinSPS PLC Programming System provides additional online support for creating programs for process sequences. Supplementary information about the sequence function chart can be displayed via the **Direct Help** button or by means of the **Contents** and **Index** options on the **Help** menu.

The folder with the pathname C:\Bosch\WinSPS contains the sample file of a precess sequence, named WASCHTIP.AS.

Introduction to SFC Sequence Function Chart

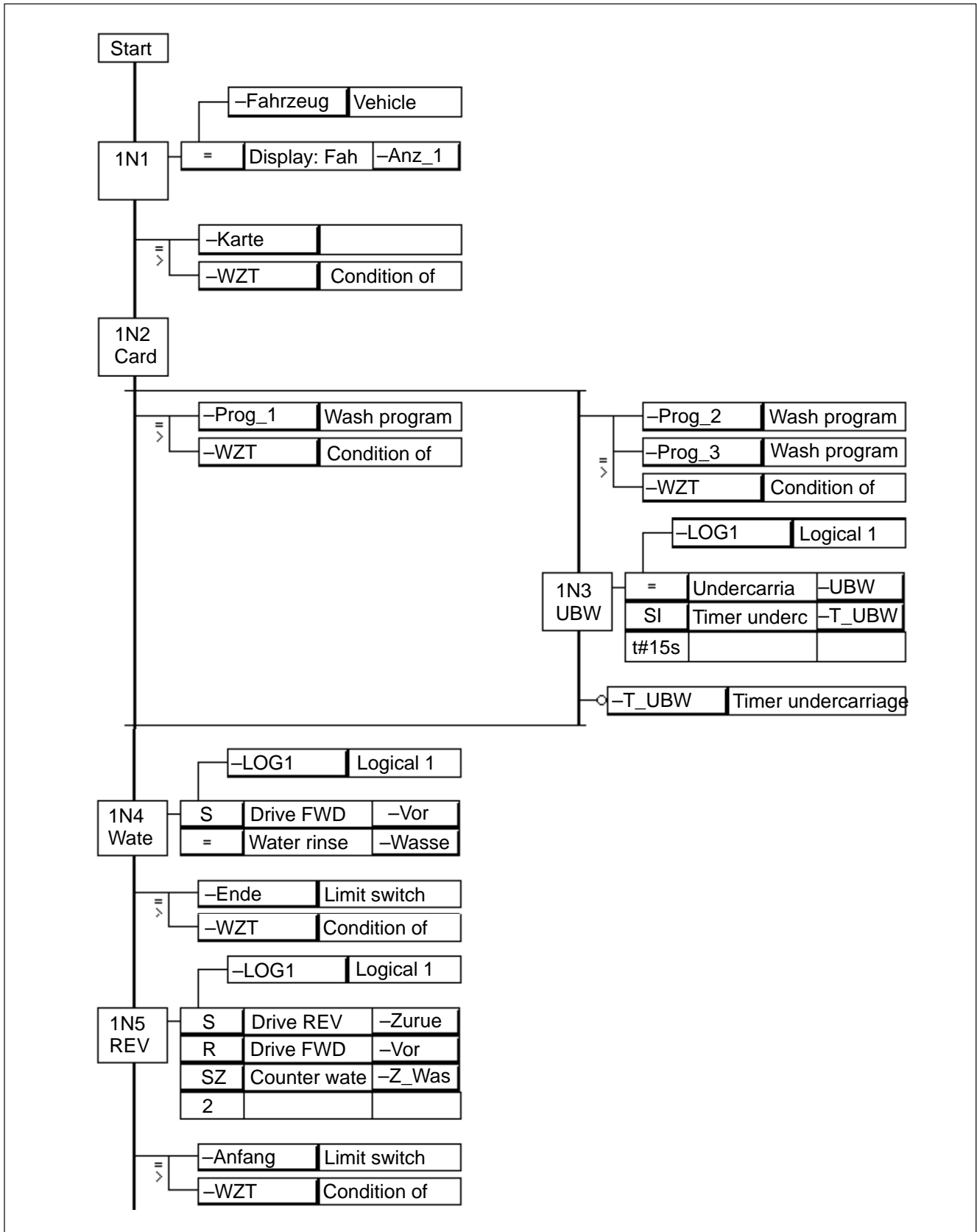


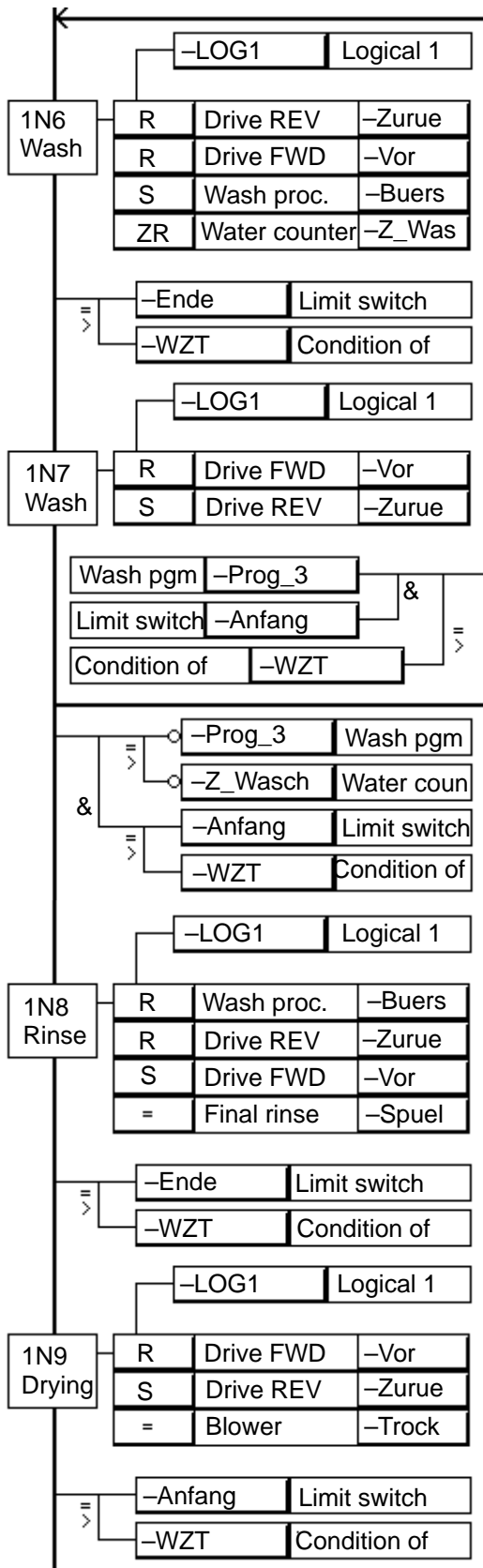
 **Expansion options through standard diagnostic and visualization software exist only in the case of programs that utilize the MADAP programming mode.**

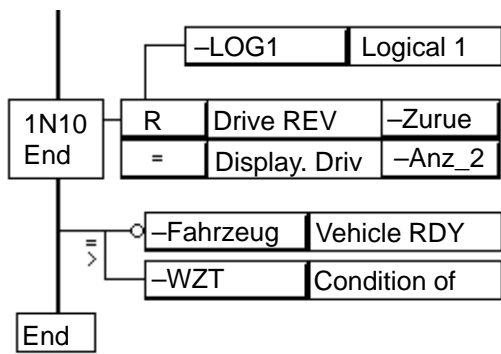
 **In a sequence cascade based upon the MADAP programming mode, only the program part of the active step is processed at any time. By contrast, in programs utilizing the step marker structure, the program part of all steps is always processed.**

With the application being equal, the MADAP programming structure always produces shortened program processing times and an optimized (shorter) machine cycle time.

The following pages show the example of a car wash plant that was created with the use of the SFC sequence function chart and the step sequence structure based upon the MADAP programming mode.







The functioning principle of sequence cascades is that of a positive-control (process or time-dependent) stepwise sequencing of a machine controller.

The sequencing from a step N to step n+1 depends upon meeting the conditions (transitions, end position attained in step n) inherent in step n.

An essential characteristic of a sequence cascade is the transparency of step programming and the unambiguous assignability of individual steps to specific machine functions.

At the project planning stage, significant time savings occur as a result of the use of preexisting standardized structures which merely require adaptation.

At the time a system is commissioned, the fact that a control function (or step) can be unambiguously assigned to a given machine movement proves a definite advantage. Designated link modifications are possible.

During machine operation, rapid troubleshooting of the sequence cascade is facilitated by the definite assignment that exists between control function and machine movement. The resulting diagnostics can be expanded to provide a diagnostic display for maintenance personnel without too much effort through the use of additional hardware and software.

A sequence cascade may consist of a linear step sequence and several step segments that are interlinked via AND (parallel) or OR (alternative) branchings. A jump instruction (skipping one or more steps) or programmed loop are also possible.

- Linear step sequence: Steps are processed in succession and in ascending order.
- Alternative branch: Steps or step sequences are subject to alternative (either/or) processing.
See also WinSPS Help, Index, "SFC: Alternative branch".
- Simultaneous branch: Steps or step sequences are processed concurrently (in parallel).
See also WinSPS Help, Index, "SFC: Simultaneous branch".
- Loop programming: Steps or step sequences are processed repeatedly.
See also WinSPS Help, Index, "SFC, Loop".
- Jump instruction: Steps or step sequences are skipped and not processed in one step cycle or are processed repeated and processed again.
See also WinSPS Help, Index, "SFC, Jump".

2.1 Sequence control system structure

The structure of a sequence control system is divided into the following main function blocks:

- Management of system operating modes.
- Graphical function display of sequence cascades, including steps and branches, jumps and loops.
- Command output.
- Diagnostics and error indication.
- User information.

2.1.1 Sequence cascades in steps

The sequence cascade is an essential component of the sequence control system. In this program part, the functional sequencing for the machine is programmed in separate process sequences. The sequence cascades are programmed in a separate program module (FC) with the use of the MADAP programming structure.

The steps are processed in succession while observing the various structural elements. An active step prepares the successive step. The command output occurs during the active step, provided that conditions have been met. The activation of each respective next step deactivates the preceding step.

In a linear sequence cascade, only one step is active at a given time. If parallel function sequences are required, several sequence cascades must be programmed.

Activating the processing of a step can also start a watchdog timer. If the watchdog interval elapses before the step processing has been concluded, this signals that a step was improperly executed, at the same time issuing the signal for fault diagnostics.

2.1.2 Managing operating modes

This MADAP program part processes the conditions for the various operating modes. The execution of operating modes is arranged in accordance with priorities. The highest, medium and lowest priorities are assigned to the Manual, Inching (single step) XX and Automatic operating modes, respectively.

The operating modes are selected via an operator terminal. Dependent upon the selected operating mode, the control of the sequence cascade is effected by the following signals:

- Operating mode selection
- Start, Stop, Reset
- Enable signals
- Command output
- Step-on conditions (transitions)
- Branch addresses
- Fault signal, etc.

A differentiation is made between the following operating modes of a sequence cascade:

Automatic mode

See also WinSPS Help, Index, "SFC: Automatic Sequence", "SFC: Operating Mode".

In "Automatic sequence" mode, the start signal is followed by the processing of individual step sequences that is either defined by programming or imposed by machine processes.

In some systems, the Machine Home position comprises one of the necessary prerequisites for the start of a given operating mode. Other systems permit the start of automatic sequencing from any system position, provided that the start was preceded by a successful synchronization between process sequences and machine position (does not apply to CL200).

Inching mode (single step)

See also WinSPS Help, Index, "SFC: Operating Mode".

In "Inching (single step)" mode, the process steps are processed in a sequential fashion, each step requiring the push of a button, there being no automatic step-on.

This operating mode facilitates program verification and commissioning and, at the same time, supports service personnel in the case of repair and maintenance procedures.

Semi-automatic mode

See also WinSPS Help, Index, "SFC: Operating Mode".

In "Semi-automatic" mode, the sequence cascade is processed automatically as long as a defined marker remains HIGH. This is used in "Inching (single step)" mode to effect automatic step-on for a number of steps within the sequence cascade, provided this becomes necessary as dictated by machine or function-specific requirements.

Manual (setup) mode

See also WinSPS Help, Index, "SFC: Manual", and "SFC: Operating Mode".

In "Manual (setup)" mode, the steps within the sequence cascade are freely selected, and the processing is started manually.

This operating mode facilitates system commissioning and provides for a direct function verification of a given actuator, e.g. subsequent to performing repair work. The function is also used for manual operation by the machine operator.

Besides the basic operating modes mentioned in the foregoing, there are also application-specific operating modes, such as Single cycle or Halt at the end of a cycle.

2.1.3 Command output

See also WinSPS Help, Index, "SFC: BEFA Condition".

MADAP programming structure

A program that is programmed on the basis of the MADAP programming structure uses a separate program module to handle the command output.

In the MADAP structure, in conjunction with the Enable signals that are dependent upon the respective operating mode, the command output is handled by comparing the number of the active step and assigning it to the step action.

Step marker structure

In a program utilizing the step marker programming mode, the command output occurs immediately subsequent to the programming of the step markers or, as a selectable option, in a separate output module.

2.1.4 Diagnostics and fault indication

This program part manages the messages and diagnostic indications and readies them for output to the operator terminal.

This is accomplished, for example, with multiline displays (BT5, BT20) or industrial-standard PCs (BT150, BT200).

2.2 Graphical display of process sequence

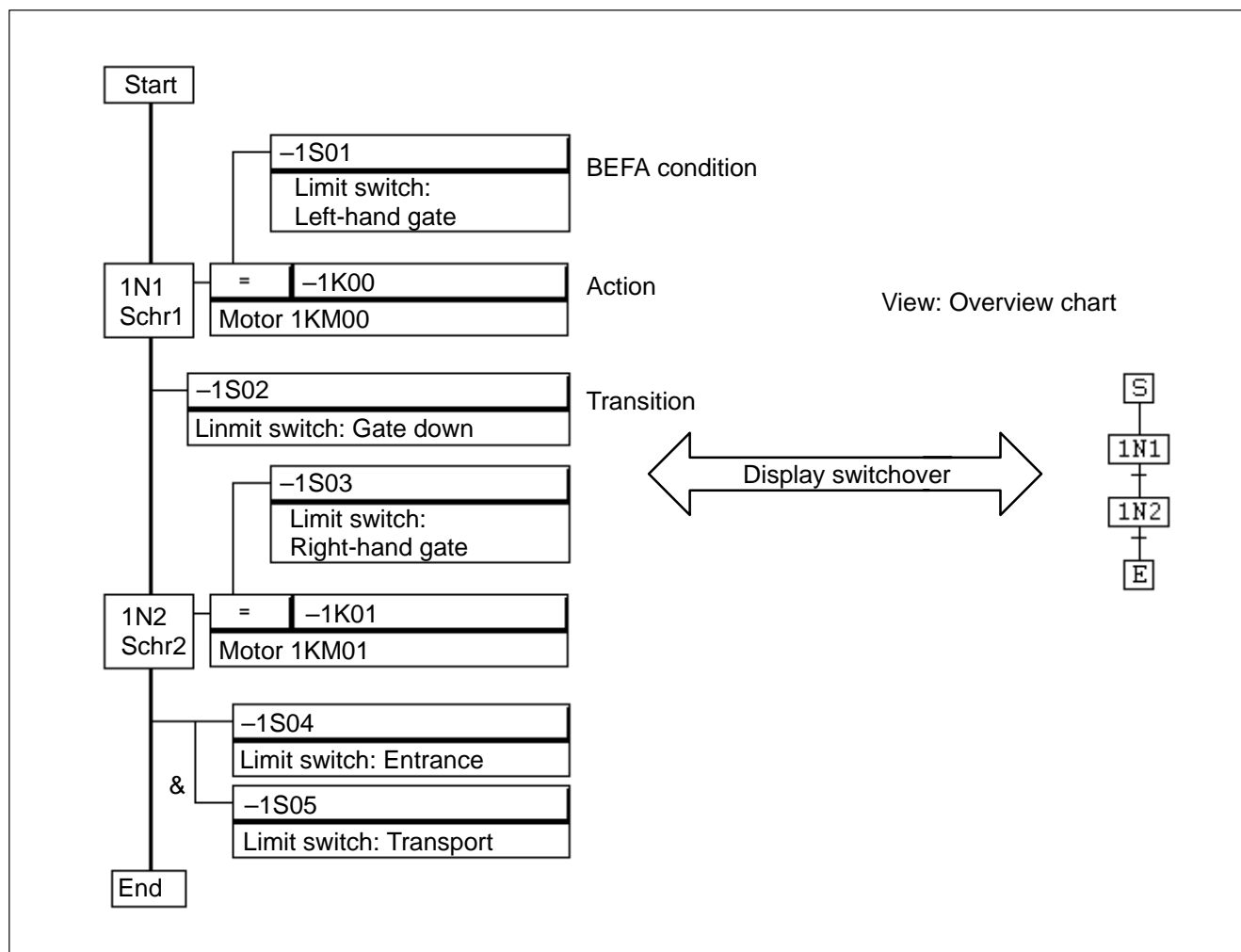
The assembly of a sequence cascade through arranging individual steps corresponds to the control sequences of the machine system. A machine-specific function is assigned to each step. However, blank steps that do not contain a control function may occur (see also WinSPS Help, Index, "SFC: Sequence Level").

Each step element features one or more input parameters (command output conditions) in an AND or OR link (also with negation). The command output represents the action leading up to the step.

The transition (satisfied step-on condition) causes the subsequent step to be initiated.

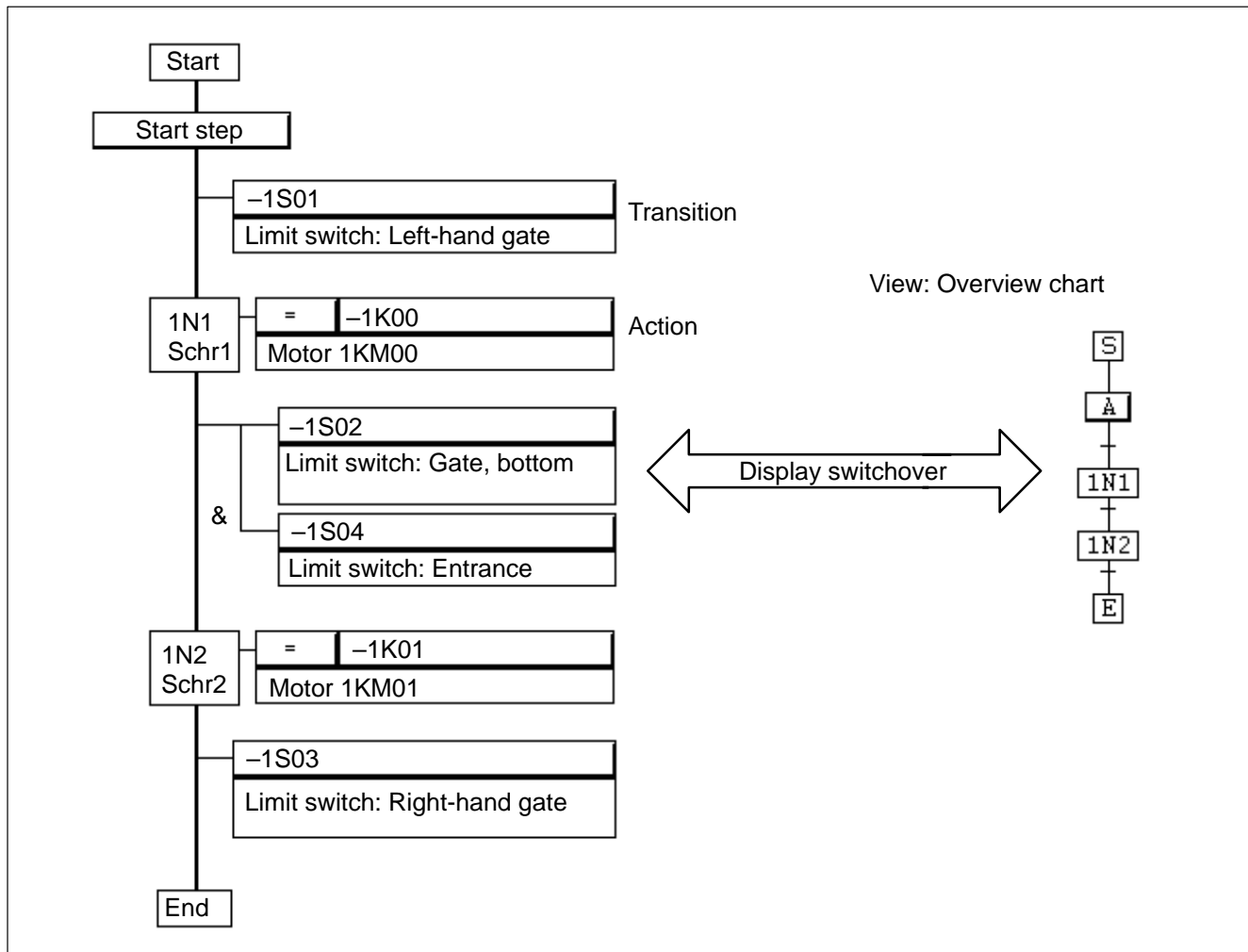
Example of MADAP programming structure

Linear sequence cascade comprising two steps, 1N1 and 1N2.



Example of step marker programming structure

Linear sequence cascade comprising three steps 1N0 (Start step), 1N1 and 1N2. The depiction of step 1N0 has been simplified.



Sequence cascade structures:

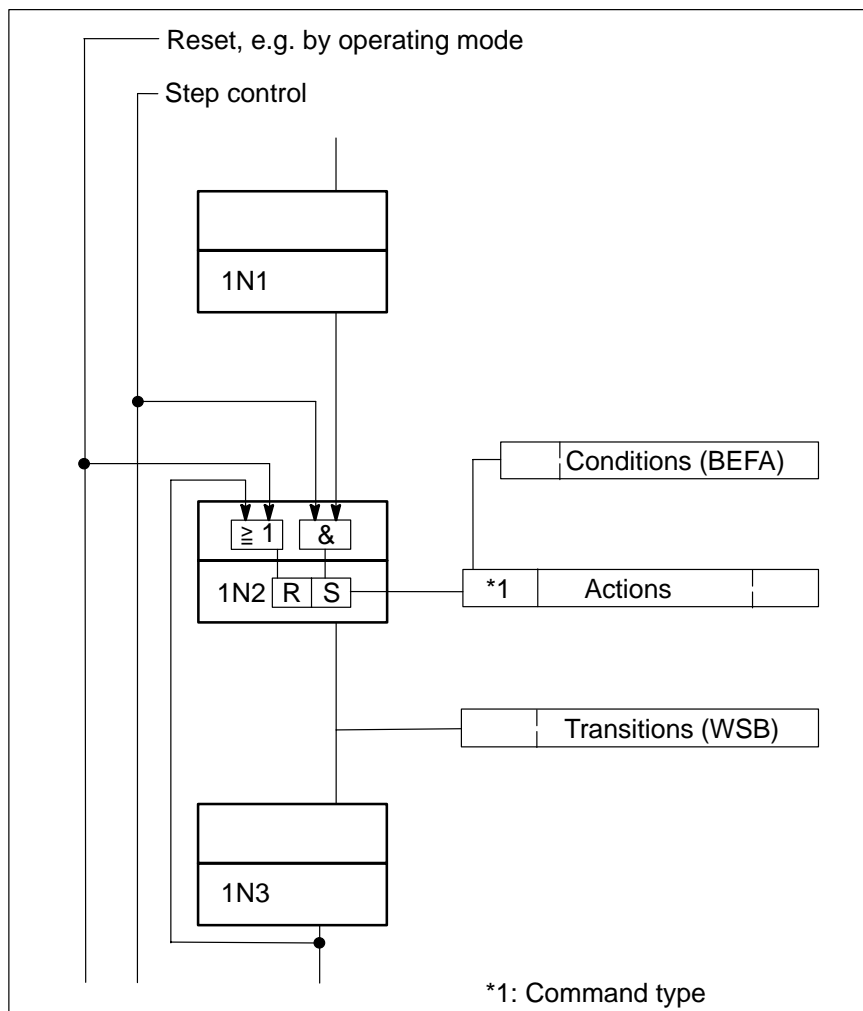
- Linear
- Alternative branches
- Simultaneous branches
- Step loops
- Branching jumps

See also WinSPS Help, Index, "SFC: Branch", "SFC: Loop", and "SFC: Jump."

2.3 Sequence step in MADAP programming mode

The action of a step is influenced by several factors:

- The operating mode management
- The preceding step
- The subsequent step



The "Conditions" field (BEFA) contains the symbolic or absolute operand addresses for the command output of a given step. Several AND or OR links (also with negation) are possible.

The "Actions" field contains the absolute or symbolic addresses of the action operand. Several parallel actions are possible.

The command output occurs with the active step and when all conditions have been satisfied.

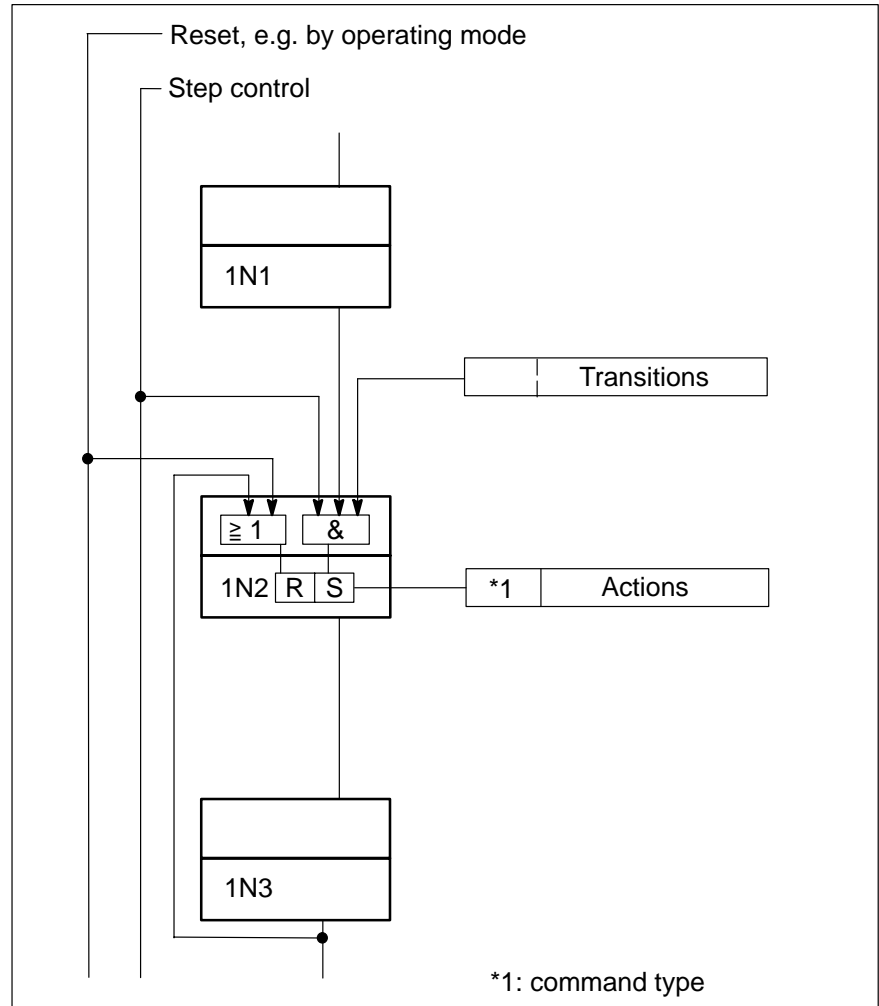
The "Transitions" (step-on conditions) field contains the absolute or symbolic addresses of the transitions, e.g. of end positions of a movement. Several transitions with AND and/or OR links (also with negation) are possible.

See also WinSPS Help, Index: "SFC, MADAP".

2.4 Sequence step in step marker technique programming mode

The action of a step is influenced by several factors:

- The operating mode management
- The preceding step
- The subsequent step



The "Transitions" (step-on conditions) field contains the absolute or symbolic addresses of the transitions, e.g. of end positions of a movement. Several transitions with AND and/or OR links (also with negation) are possible.

The "Actions" field contains the absolute or symbolic addresses of the action operand. Several parallel actions are possible.

See also WinSPS Help, Index: "SFC, Step Marker Technique".

3 Sequence Cascade in MADAP Mode

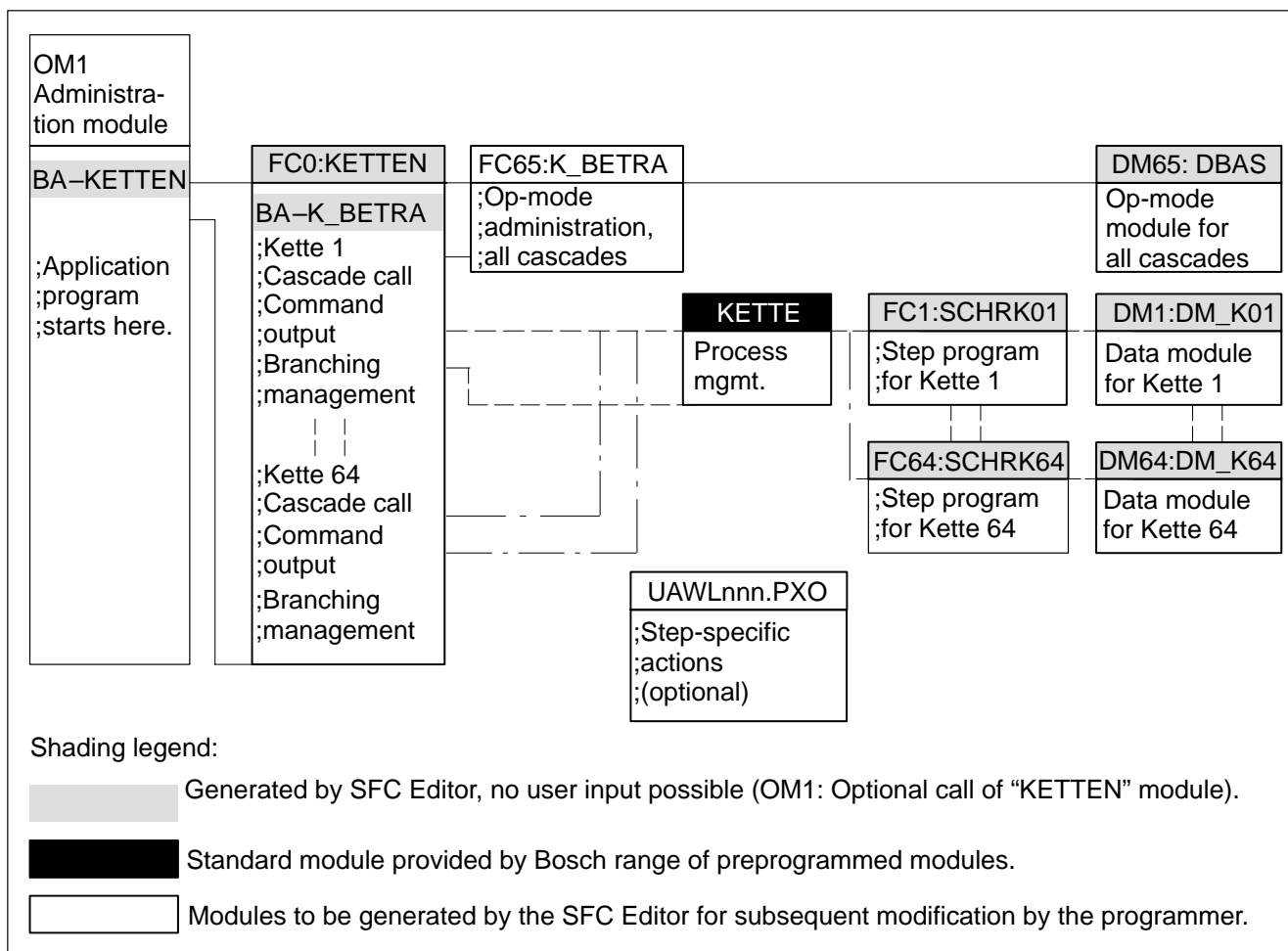
Sequence cascade programming on the basis of the MADAP control and operating concept provides significant advantages when compared to the conventional sequence cascade programming mode using the step marker technique.

Advantages of MADAP structure

- Use of standardized function modules.
- Expandability to include automatic diagnostics and operator guidance.
- Shorter program processing time, thus shorter machine cycle times.
- Synchronization between machine function and processing step (except CL200).

The sequential order of module calls for the MADAP program package is shown in the diagram below.

Using the example of a CL400/500, the module structure generated by the SFC Editor appears as follows:



Besides the sequence cascade modules of the MADAP program package, standardized program modules are available for the following functions:

- Machine operation with preprogrammed screen masks
- Diagnostics and fault indication in the event of cascade faults
- Indication and control of operand status
- Indication of controller information
- Machine usage indication (MMI-MADAP only)
- System configuration display (MMI-MADAP only)
- Display of user screens
- Display of threshold values (MMI-MADAP only)
- Protocol buffer (MMI-MADAP only)

3.1 Generating a MADAP-based sequence cascade

The sequence function chart (SFC) generates the program and data modules for the sequence cascades.



CAUTION

Once generated, the modules can be modified manually for testing purposes. The “SFC, Create Sequence” command will again overwrite the modifications with the function initially programmed under SFC.

MADAP sequence cascade modules

See also WinSPS Help, Index, “SFC, Create Sequence”.

Absolute module address	Symbolic module address	Functions
FC0	-KETTEN.PXO	Branching management, operating mode control, command output.
FC1 thru FCnn	-SCHRK01.PXO thru -SCHRKnn.PXO	Sequence cascades with manual and automatic branches.
FC65	-K_BETRA.PXO	Sequence cascade operating modes. Module is created only if not yet present. An existing module will not be overwritten.
FC200	-KETTE.PXL – KETTE200.PXL (CL200) -KETTE700.PXL (ICL700)	Cascade organization, limited to entry in symbol file.
DM1 thru DMnn	-DB_K01 thru -DB_Knn	Sequence cascade data.
DM65	-DBAS.PXD	Operating mode data.
DM120	-DB120.PXD	Transfer of diagnostic data (CL200 only).

The program module named UAWLnnn.PXO is created through the use of the “Insert” PLC command; it is not, however, entered into the symbol file.

Symbol file

With the exception of the UAWLnnn.PXO module, the modules belonging to the sequence cascade structure are written into the symbol file.



CAUTION

Once generated, the module entries in the symbol file can be modified manually. The “SFC, Create Sequence” command will restore the MADAP structure (module addresses).

Symbol file

Operand addresses for CL200

Absolute module address	Symbolic module address	Functions
M188	-VERZWADR	Branching address
M191.0	-BEFA	Command output
M191.1	-WSB	Step-on condition
M191.2	-StoeM	Fault marker
M191.4	-HalbAuto	Automatic step-on during Inching mode
M191.5	-WZT_Halt	Stop wait timer
M191.6	-WZT	Wait time marker state
M191.7	-UETKONTR	Stop monitoring watchdog timer

Operand addresses for CL350/CL400/CL500/PCL/ICL700

Absolute module address	Symbolic module address	Functions
M242	-VERZWADR	Branching address
M255.0	-BEFA	Command output
M255.1	-WSB	Step-on condition
M255.2	-StoeM	Fault marker
M255.4	-HalbAuto	Automatic step-on during Inching mode
M255.5	-WZT_Halt	Stop wait timer
M255.6	-WZT	Wait time marker status
M255.7	-UET_Halt	Stop monitoring watchdog timer

KETTEN.PXO cascade module

See also WinSPS Help, Index, "SFC, Create Sequence".

As required by the user, the call for the cascade module (CM-KETTEN) can be generated automatically in the OM1 or entered manually.

**CAUTION**

When using the automatic generating function, the original programming in the OM1 organization module will be overwritten.

As an alternative, the call (for the sequence cascades) can also be inserted manually at a suitable program position. The functions of this module apply to all sequence cascades.

For all sequence cascades, the sequence function chart generates the –

- Call for the sequence operation mode module
- Call for the cascade management module
- Management of simultaneous branches
- Command output
- Management of multiple assignments

**CAUTION**

Once generated, the KETTEN.PXO program module can be modified manually for testing purposes. The "SFC, Create Sequence" command will again overwrite the modifications with the function initially programmed under SFC.

The KETTEN module is divided into networks, with each network containing a separate function block of the associated cascade.

The control of parallel-branching subcascades is effected by the Reset and Start operating mode bits.

SCHRKnn.PX0 step module

See also WinSPS Help, Index, "SFC, Create Sequence".

The call for the step module is issued from within the cascade organization module, e.g. KETTE.PXL.

By default, analogous with the cascade number, an FC with a corresponding number (cascade 1 = absolute address of FC1 = symbolic address SCHRK01.PX0) is used and entered into the symbol file.

A separate step module is generated for each sequence cascade.



CAUTION

Once generated, the SCHRKnn.PX0 step module can be modified manually for testing purposes. The "SFC, Create Sequence" command will again overwrite the modifications with the function initially programmed under SFC.

The SCHRKnn.PX0 step module contains the PLC program matching the sequence cascade, including all command output conditions, step-on conditions (transitions) and branching jumps.

The sequence cascades are listed in the sequence of their graphical representation. The steps are listed in the jump instruction at the start of the module, and are followed by the individually programmed steps.

The sequence steps are divided into manual and automatic branches. The determination of which branch is to be processed is derived through loading the reported operating mode ("control mode") from the cascade data module belonging to a respective sequence cascade.

In the event that the conditions for Manual and Automatic mode are identical, no specific branches will be created for the operating modes.

The branches contain –BEFA (command output) and –WSB (step-on condition) markers which are defined for each type of control unit. The referred –BEFA und –WSB markers have the same addresses in each step and in each sequence cascade of a given control unit type.

The conditions from the automatic branch can be copied into the manual branch. In this manner, programming the manual branches can be accomplished through copying and subsequent modification under conditions that are almost identical to those provided by automatic programming.

The diagnostic function accesses these branches for display purposes while analyzing the links at the same time.

Step processing in SCHRKnn.PXO step module

Step processing commences at the following point:

- Subsequent to a Reset of the sequence cascade, with the first step.
- Subsequent to successful synchronization for Automatic mode, with the first step after synchronization.
- With the Manual mode step that was selected by the movement key.
- The command output occurs if the –BEFA conditions of a given step have been met.
- In Automatic mode, once the –WSB step-on conditions of a step have been met, processing of the next step is started. In Inching mode, after the –WSB step-on conditions of a step have been met, a wait interval for data bit D6.4 (S+1) is inserted before the step-on can occur.
- In Automatic mode, once the –WSB step-on conditions of the last step have been met, the first step is again processed. In Inching mode, once the –WSB step-on conditions of the last step have been met, a wait interval for data bit D6.4 (S+1) is inserted before the step-on can to Step 1 can occur.

Manual mode permits random access to any step of the sequence cascade. The step processing is started by the movement key on the operator terminal.

K_BETRA.PXO cascade mode module

See also WinSPS Help, Index, "SFC, Create Sequence".

The cascade mode module is called from within the KETTEN.PXO cascade module.

The K_BETRA.PXO cascade mode module handles the transfer or operating modes to all sequence cascades.

The sequence function chart generates the K_BETRA.PXO cascade mode module as a program basis for an extension allowing operating mode control by the user.

The user is required to use the K_BETRA.PXO cascade mode module to program the operating mode and to write the same into the DBAS.PXD data module.

**CAUTION**

If this module does not yet exist, it will be created. The user will then modify this module in accordance with his requirements. The use of the "SFC, Create Sequence" command will not cause an existing cascade mode module (K_BETRA.PXO) to be modified.

In the DBAS.PXD cascade mode data module, a data word is provided for the operating mode of each sequence cascade, i.e., data word 2 for cascade 1, data word 4 for cascade 2, and so forth.

```
;Example of Automatic operating mode with start (CL400):
BA      -DBAS                ;Automatic operating mode
L       D2,A                ;Operating mode, 1st cascade
A       -AUTO               ;Automatic selected?
=       A.2                 ;Automatic selected

;START:
A       -E_START            ;Operating mode started?
=       A.3                 ;Start of operating mode

;Operating mode and start for cascade 1 as per DBAS/D2
T       A,D2                ;Transfer of operating mode and start of DBAS
```

In the case of parallel branching of cascade sequences, bit 12 of the operating mode (no step-on in same cycle, refer to tables below) must be LOW.

Significance of operating mode bits for CL200, excerpt from DBAS data module.

Bit	Explanation
0	Manual mode (H)
1	Inching mode (T)
2	Automatic mode (A)
3	Start (s)
4	Execute next step, S+1
5	Set step
6	Acknowledge fault (D7.7=1)
7	Reset
8	Stop
9	reserved
10	reserved
11	reserved
12	reserved
13	reserved
14	reserved
15	Fault acknowledgement is mandatory

Significance of operating mode bits for CL350/CL400/CL500/ICL700/PCL, excerpt from DBAS data module.

Bit	Explanation
0	Manual mode (H)
1	Inching mode (T)
2	Automatic mode (A)
3	Start (s)
4	Execute next step, S+1
5	Set step
6	Acknowledge fault (D7.7=1)
7	Reset (r)
8	Stop (h)
9	Synchronize
10	reserved
11	Generate cascade data module
12	No step-on in same cycle
13	Diagnostics if WSB step-on condition=HIGH
14	WSB does not reset BEFA, manual mode only
15	Fault acknowledgement is mandatory

Sequence organization modules

- KETTE200.PXL (CL200)
- KETTE.PXL (CL350/400/500)
- KETTE700.PXL (ICL700)
- KETTEPCL.PXL (SoftSPS)

See also WinSPS Help, Index, "SFC, Create Sequence".

The cascade organization module is called from within the KETTEN.PXO cascade module.

The referred cascade organization modules are supplied in the MADAP software packages and may also be purchased separately. They are entered into the symbol file and are not generated. The function of cascade organization modules is controlled by appropriate parameterization.

The module named KETTE200.PXL calls the "sequence cascade" function in the CL200.

UAWLnnn.PXO Module-specific step programming

See also WinSPS Help, Index, "SFC, Rebuild all".

Using the functions below, the sequence function chart generates the module as a template, and supplements the action through the PLC-Edit action element:

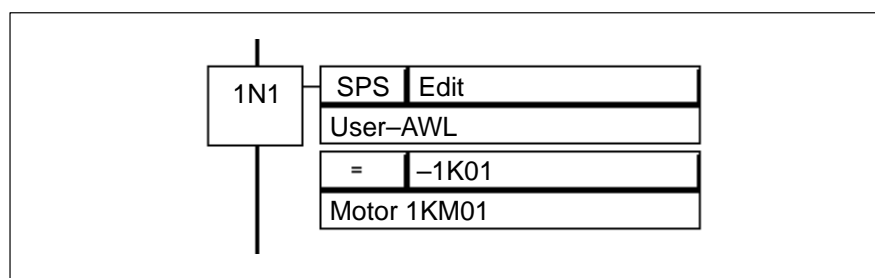
- Language elements
- Insert action
- PLC commands

An existing UAWLnnn.PXO module will not be overwritten, and an additional module named UAWLnnn+1.PXO is created. The module must be modified by the user. It is not entered into the symbol file.

The module can be edited with the use of the "EDIT" command.

Independent of the MADAP structure, the module permits commands that are additional to the action of a given step.

PLC-Edit action element



The programming inherent in the UAWLnnn.PXO module is taken into account when the step action is generated.

Program in UAWLnnn.PXO module

```

;Example:
A      -I S10
=      M23.0
    
```

Linking with action, step -1N1

```

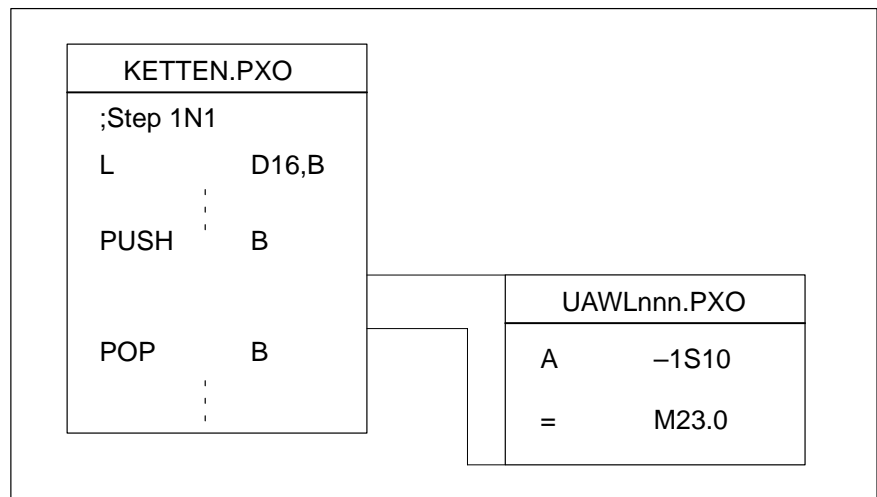
;Example: Linking of UAWLnnn.PXO with action
;Schritt 1N1

;Read active step
L      D16,B

;Step 1N1
CPLA   1,B
A      -ZERO           ;1N1 command output?
PUSH   B              ;Save register B

;Start PLC commands UAWLnnn
A      -I S10         ;Command output linked
=      M23.0         ;with UAWLnnn.PXO program

;END PLC commands UAWLnnn
POP    B              ;Load register B
    
```



DB_K01.PXD ... DB_Knn.PXD sequence data modules

See also WinSPS Help, Index, "SFC, Create Sequence".

The sequence function chart generates the data modules for the sequence cascades.

**CAUTION**

If this module does not yet exist, it will be created. The user will then modify this module in accordance with his requirements. The use of the "SFC, Create Sequence" command will not cause an existing cascade mode module to be modified.

The data modules of the sequence cascades essentially contain the following entries (example of CL400):

- D0 Fault and/or status entries about sequence cascade
- D2 Cascade number
- D4 Number of steps
- D6 Selected operating mode
- D8 Reported operating mode
- D10 Step number of completed step
- D12 Step number of current step
- D14 Step number of prepared step
- D16 Command output via step number
- D18 Actual value, wait time
- D20 Actual value, monitoring time
- D48 thru D62 Synchronized steps
- D64 thru D78 Active step
- D80 thru D94 Binary command output for active step

The data is entered by means of the "Generate cascade data module" (D7.3) function or the "Generate sequence cascade" function (CL200).

The "Generate cascade data module" function (except for CL200) is processed on the following occasions:

- Subsequent to each program load
- Upon power-up of the PLC

The diagnostic function utilizes a segment of this data for the display.

Detailed information about the data bits can be taken from the module descriptions of the respective descriptions of the MADAP software packages.

DBAS.PXD data module for sequence function chart

See also WinSPS Help, Index, "SFC, Create Sequence".

The sequence function chart generates the DBAS.PXD data module.

The DBAS.PXD data module serves to control the operating module of the sequence cascades. For information on operating mode bits, refer to page 3-9.

The user-specific program in the K_BETRA.PXO module provides the data words with operating mode information.

With the exception of the CL200, the operating mode is read from the DBAS.PXD in the KETTEN.PXO module and transferred to the cascade organization module by means of a parameter.

If the CL200 is used, the operating mode is directly taken from the DBAS.PXD module and entered into data word D6 of the cascade data module.

DB120.PXD diagnostics data module (CL200)

The sequence function chart for the CL200 generates the DM120 data module.

**CAUTION**

If this module does not yet exist, it will be created. The user will then modify this module in accordance with his requirements. The use of the "SFC, Create Sequence" command will not cause an existing module to be modified.

During operation, information about the active first-value error of a sequence cascade is automatically collected in the transfer module. As an option, the data referring to occurring consequential errors may be entered into an additional range.

An operator terminal accesses this data module for diagnostics and display purposes.

3.2 Specific MADAP markers

The MADAP software package makes use of marker addresses that are subject to controller-specific definition.

–BEFA

In the sequence cascades, the assignment of the –BEFA command output marker is programmed via the command output conditions.

The step action is carried out when the marker goes HIGH.

In each sequence cascade and individual step in both Manual and Automatic branches, the same marker with a defined address is used. The marker assignment to the proper sequence cascade and step is handled by the sequence organization module and by a corresponding entry in the cascade data module. The program-controlled command output concerning a given step action is effected by a compare operation in the KETTEN.PXO module.

–WSB

The assignment of the –WSB step-on condition marker is programmed via the conditions governing the step-on action. If this marker goes HIGH, processing is ended for the current and started for the subsequent step. At this point, there exist options for programmer intervention (e.g. linking with time-out or wait time marker).

Step-on action occurs when the marker goes HIGH.

In each sequence cascade and individual step in both Manual and Automatic branches, the same marker with a defined address is used. The marker assignment to the proper sequence cascade and step is handled by the sequence organization module.

–StoeM (fault marker)

The –StoeM fault marker starts diagnostics and fault indication independent of the monitoring time. For example, in the event that a protective door is opened in the system, this will result in the immediate return of a fault indication and user information.

The diagnostics commence when the marker goes HIGH.

–VERZWADR (branch address)

The branch address in a marker word is used to transfer, within a given step, a step number for the next step processing (branch) to the sequence organization module. As the step-on condition must be fulfilled before the branching can take place, a step is always fully processed before branching occurs.

This method can be used to skip processing steps (branching jump) or to implement OR branches and loops.

With the use of the sequence function chart, this marker word address is automatically tagged with the proper destination step number in the SCHRKnn.PXO step module.

-UET_Halt (monitoring time HALT)

In the absence of freely selectable conditions, the -UET_Halt "Monitoring Time Halt" marker is used to stop the monitoring interval, an action that causes the automatic diagnostics to be disabled.

No diagnostics occur when the marker is HIGH.

-WZT (wait time marker status)

The value for the wait time is either transferred to the sequence organization module in the form of a parameter or, in the case of the CL200, is written into a data word of the cascade data module.

Independent of the foregoing, the wait time can be adapted for each step through step-specific programming.

For example, the step-on action can be delayed by linking the wait time marker within the step programming of the Automatic branch.

Marker LOW: Wait time is running.

Marker HIGH: Wait time has elapsed.

-WZT_Halt (wait time HALT)

The wait time can be halted by means of freely selectable conditions.

For example, the step-on action can be delayed by linking the wait time marker within the step programming of the Automatic branch.

When the marker goes HIGH, the monitoring time is halted.

3.3 Specific operating mode bits

The functions in the operating mode word of the cascade data modules (DM_K01 through DM_Knn) are selected and/or executed when the respective bit attains HIGH status. Individual functions are described below.

D6.0 Manual mode

This bit selects Manual operating mode and possesses the highest priority ranking in the selection of operating modes.

The freely selectable step execution under Manual conditions is effected by manipulating movement keys on the operator terminal. In normal conditions, the D6.3 Mode start is effected by pressing the movement key.

D6.1 Inching mode

This bit selects Inching mode and possesses medium priority ranking in operating mode selection.

Once Start (D6.3) has been enabled, step execution under Automatic conditions requires selecting the S+1 key (D6.4) on the operator terminal.

The steps are executed in the sequence prescribed by Automatic mode.

D6.2 Automatic mode

This bit selects Automatic mode and possesses the lowest priority ranking in operating mode selection.

Once Start (D6.3) has been enabled, the step execution under Automatic mode conditions is system-controlled.

D6.3 Mode start

This bit starts the selected operating mode. It is valid for all operating modes.

To enable the "Synchronize" function in Automatic mode, the bit must be LOW.

D6.4 S+1

- Inching mode, execute next step; –WSB step-on condition of current step must be met
- Semiautomatic mode, the next steps are executed as long as the –HALBAUTO semiautomatic marker is HIGH; prerequisites: –WSB step-on conditions of steps are met.

D6.5 Set step

Adopt preselected step number and execute in Manual mode.

D6.6 Halt acknowledgement

Manual acknowledgement of a diagnosis; only when D7.7 "Mandatory Fault Acknowledgment" marker is HIGH.

D6.7 Reset

Deletes active step from processing, reinitializes step processing and prepares step 1.

D7.0 Halt

Halts sequence cascade. Processing of current step continues. Command output continues.

D7.1 Synchronize

(Not for CL200.)

- **Automatic**

In the SCHRKnn.PXO step module, the Automatic branch is searched for fulfilled –BEFA command output conditions and nonfulfilled –WSB step–on conditions. This is done to check which line of step programming matches the current machine position.

In the event that the exact programmed step matching the current machine position is found, this step is prepared for the start of operation in Automatic mode.

- **Manual**, no operating mode bit

The steps matching the current machine position are determined and provided for execution in the movement screens. The respective function is started by “Releasing” the movement key (transition from HIGH to LOW state).

D7.3 Generate cascade data module

(Not for CL200.)

The sequence organization module writes the cascade data, e.g. the number of steps, to the DM_Knn.PXD cascade data module. To do so, the start bit must be LOW.

This function is also performed:

- Subsequent to program loading
- Upon energizing the PLC

D7.4 Step-on action

(Not for CL200.)

The D7.4 operating mode bit controls the step-on action within the same PLC cycle.

Automatic mode:

- **LOW state:**
With the step-on condition fulfilled, the next step is processed in the same PLC cycle, whereby program processing time is shortened. The cycle time of the machine is thus optimized.
- **HIGH state**
With the step-on condition fulfilled, the next step is processed in the subsequent PLC cycle.

D7.6 WSB does not reset BEFA

(Not for CL200.)

Manual mode only:

- LOW
With the -WSB step-on condition fulfilled, the corresponding -BEFA command output is reset.
- HIGH
With the -WSB step-on condition fulfilled, the corresponding -BEFA command output is not reset.

D7.7 Mandatory fault acknowledgement

- HIGH
A diagnosis must be acknowledged by the D6.6 "Halt acknowledgement" bit; even with the -WSB step-on condition fulfilled, no automatic step-on action occurs.
- LOW
A diagnosis does not require acknowledgement. With the -WSB step-on condition fulfilled, automatic step-on occurs.

3.4 Step branching

In the course of sequence cascade processing, steps may be omitted or skipped. The relevant actions are controlled via the branching marker word.

The branching marker word represents an address that is defined in a controller-specific fashion.

Example of step branching for a CL400:

In Branching Step 2, the step-on operands I0.5 and I0.6 determine whether Step 3 or Step 4 is to be processed next.

```

;Branching Step 2

1N2:

;OR decision
DEF      M230.0,-Zweig1Auto_1
DEF      M230.1,-Zweig2Auto_2

;Alternative branch: Step 3
A        I0.5
=        -Zweig1Auto_1

;Alternative branch: Step 4
A        I0.6
=        -Zweig2Auto_2

;Transfer of step number
L        3,A
A        -Zweig1Auto_1
JPC      AlternativVerzweigungSchritt2

L        4,A
A        -Zweig2Auto_2
JPC      AlternativVerzweigungSchritt2

L        2,A

AlternativVerzweigungSchritt2:
T        A,-VERZWADR

```

3.5 Special-purpose cascades

Monitoring sequence cascades

Sequence cascades programmed on the basis of the MADAP structure provide the option of automatic cascade fault diagnostics.

To diagnose additional machine functions that are extraneous to a sequence cascade, the referred functions can be monitored in a separate sequence cascade. Such cascades mostly comprise a single step.

Examples:

- Basic start conditions
- Tandem limit switch monitoring

In the event of a fault occurrence, there will be an automatic diagnostic display for these functions also.

Command output is not normally required for these special-purpose cascades.

Manual sequence cascades

These comprise sequence cascades that are executed in Manual mode.

They are programmed with Automatic mode conditions, however.

4 Structural Elements of MADAP Sequence Cascade

To program a sequence cascade on the basis of the MADAP programming mode, the following structural elements are available:

- Linear structure
- Alternative branch
- simultaneous branch
- Loop
- Branch jump

4.1 Linear sequence cascade in MADAP mode

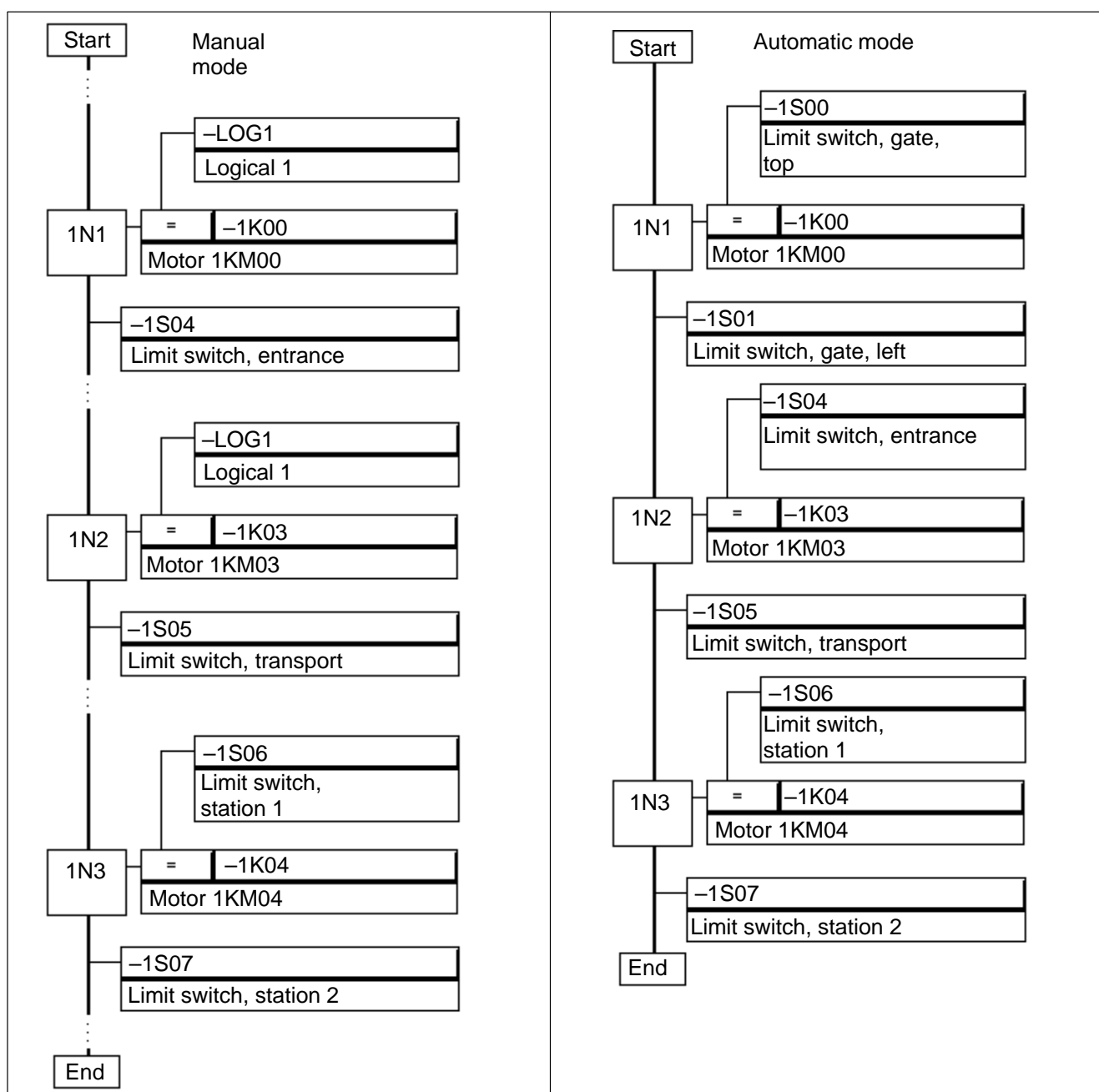
See also WinSPS Help, Index, "SFC, MADAP".

The linear sequence cascade processes the steps, in accordance with their graphical diagram, from the first step to the last in a contiguous ascending sequence. The command output occurs accordingly.

The sequence function chart generates the program organization for a sequence cascade.

Example: CL400

Linear sequence cascade 1 with steps 1N1 (start step), 1N2, 1N3 and differentiation of Manual and Automatic mode.



Creating linear sequence cascade in MADAP mode

See also WinSPS Help, Index, "SFC, Create Sequence".

For the example on the preceding page, the "Create Sequence" function generates the following modules, which are also written into the symbol file:

- KETTEN.PXO Operating mode control & command output module
- K_BETRA.PXO Cascade operating modes
- SCHRK01.PXO Step programming for sequence cascade
- DB_K01.PXD Data module for sequence cascade
- DBAS.PXD Data module for operating modes

4.2 Alternative-branched sequence cascade in MADAP mode

See also WinSPS Help, Index, "SFC, Alternative".

The alternative-branched sequence cascade processes the steps in accordance with their graphical diagram. In alternative branches, only one branch is processed at any time. The –WSB step-on conditions at the start of the alternative branch define which branch is to be processed.

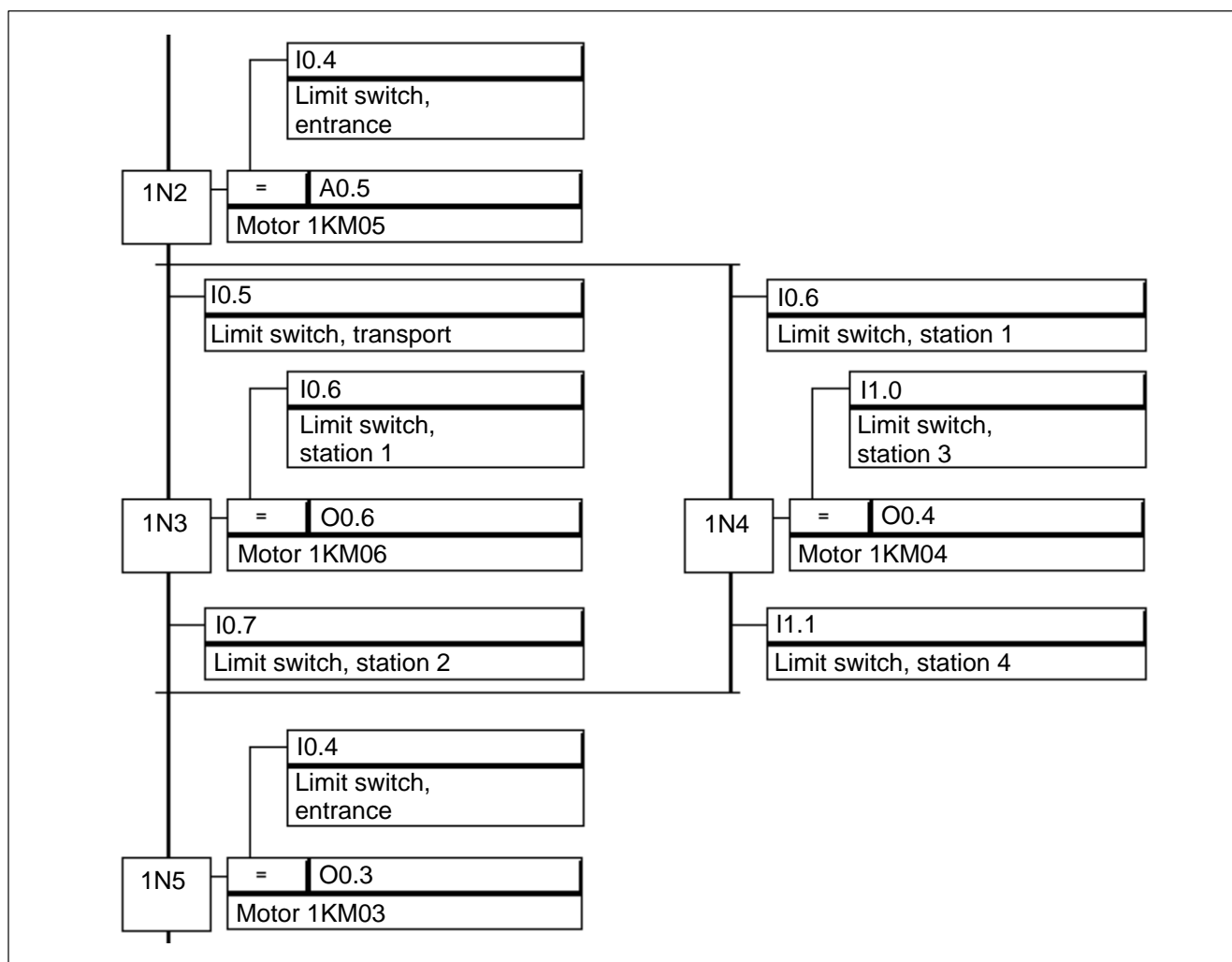
The difference between the command output for a sequence cascade with alternative branches and that for a linear sequence cascade is that no command output occurs for the steps in the branch that is not being processed.

The sequence function chart generates the program organization for a sequence cascade.

The alternative branch is denoted by single horizontal lines.

Example: CL400

Section of alternative-branched sequence cascade 1 with Step 1N2, alternative step 1N3, alternative step 1N4 and step 1N5.



Subsequent to processing Step 1N2, the alternative branching causes processing to continue with Steps 1N3 or 1N4.

Step-on condition I0.5 initializes step 1N3; alternatively, step-on condition I0.6 initializes step 1N4.

In step 1N2 the step-on conditions for step 1N3 and/or step 1N4 define the step number that will be written into the –VERZWADR branching marker word. The step processing will occur accordingly.

The branching management is controlled by the contents of the –VERZWADR branching marker word which are subject to controller-specific definition.

If the state of both step-on conditions is HIGH, the I0.5 step-on condition will have priority (step-on to 1N3) over its I0.6 counterpart.

Step 1N5 is initialized by either step 1N3 or 1N4.

The sequence organization module (e.g. KETTE.PXL) controls, via the contents of Register A, the step processing in the SCHRK01.PXO step module. In accordance with the decision in step 1N2 (stored in branching marker word), a specific alternative branch per sequence cascade is not processed.

Creating alternative-branched sequence cascade in MADAP mode

See also WinSPS Help, Index, "SFC, Create Sequence".

For the example on the preceding page, the "Create Sequence" function generates the following modules, which are also written into the symbol file:

- KETTEN.PXO Operating mode control & command output module
- K_BETRA.PXO Cascade operating modes
- SCHRK01.PX0 Step programming for sequence cascade and branching management
- DB_01.PXD Data module for sequence cascade
- DBAS.PXD Data module for operating modes

 **In the case of alternative branching, only one sequence cascade is ever generated, regardless of the width of the branch.**

4.3 Simultaneous-branched sequence cascade in MADAP mode

See also WinSPS Help, Index, "SFC, Branch".

The simultaneous-branched sequence cascade is processed in accordance with its graphical structure.

In a simultaneous branch, all simultaneous branches are processed without exception.

For each parallel branch, the command output is handled by its own sequence cascade.

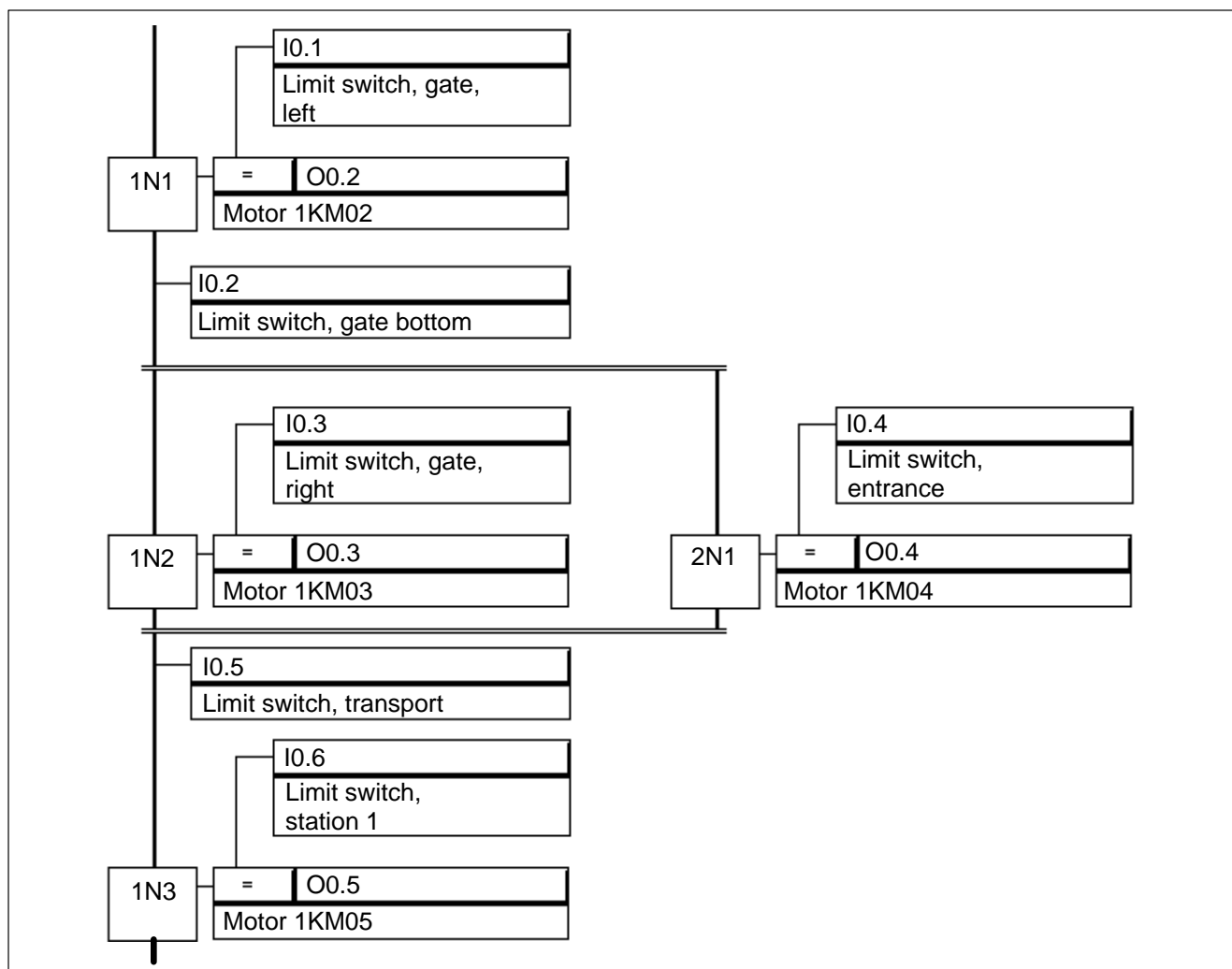
The sequence function chart generates the program organization for several sequence cascades.

For each simultaneous branch, the program organization of a separate simultaneous sequence cascade is generated.

The simultaneous branch is denoted by parallel horizontal lines.

Example: CL400

Section of simultaneous-branched sequence cascade. Main cascade with steps 1N1, 1N2 and 1N3. Simultaneous cascade 2 with step 2N1.



The simultaneous branch starts after step 1N1.

The user is required to use the application-specific program in the K_BETRA.PXO cascade operating module to program the operating modes, and to write the same into the DBAS.PXD data module.

This data is again loaded in the KETTEN.PXO sequence module and transferred, as parameter P1 in both the Network 1 (main sequence) and Network 2 (simultaneous sequence), to the KETTE.PXO module.

In the example, the module call (CM KETTE,4) occurs twice with the following parameters:

- Number of respective sequence cascade
- Operating mode of respective sequence cascade
- Time value for monitoring interval of respective sequence cascade
- Time value for wait time (from step to step) of respective sequence cascade

If the main sequence is engaged in step processing prior to the simultaneous branch, the simultaneous sequence will be reset.

If the main sequence is engaged in processing the first step in the simultaneous branch, the simultaneous sequence will be started.

If the main sequence is engaged in step processing after the simultaneous branch, the simultaneous sequence will be reset.

Network 1 of the KETTEN.PXO module contains:

- Operating mode control for main sequence
- Operating mode control for simultaneous sequences
- Command output for main sequence

The command output to the main sequence is followed by operating mode control for simultaneous sequence, including Start and Reset.

Network 2 of the KETTEN.PXO module contains:

- Operating mode control for simultaneous sequence
- Command output for simultaneous sequence

In the SCHRK01.PXO and SCHRK02.PXO modules, the sequence steps are denoted in the succession of their graphical representation. In each sequence, the succession of step processing corresponds to that of a linear sequence.

Creating simultaneous-branched sequence cascade in MADAP mode

See also WinSPS Help, Index, "SFC, Create Sequence".

For the example on the preceding page, the "Create Sequence" function generates the following modules, which are also written into the symbol file:

- KETTEN.PXO Operating modes, command outputs, and simultaneous sequence operating modes
- K_BETRA.PXO sequence operating modes
- SCHRK01.PXO Main sequence
- SCHRK02.PXO Simultaneous sequence
- DB_K01 Data module for main sequence
- DB_K02 Data module for simultaneous sequence
- DBAS Data module for operating modes

4.4 Sequence cascade loop in MADAP mode

See also WinSPS Help, Index, "SFC, Loop".

Loop processing is controlled by the loop step-on condition. It causes repeated processing of identical steps or step sections.

The sequence function chart generates the program organization for a sequence cascade.

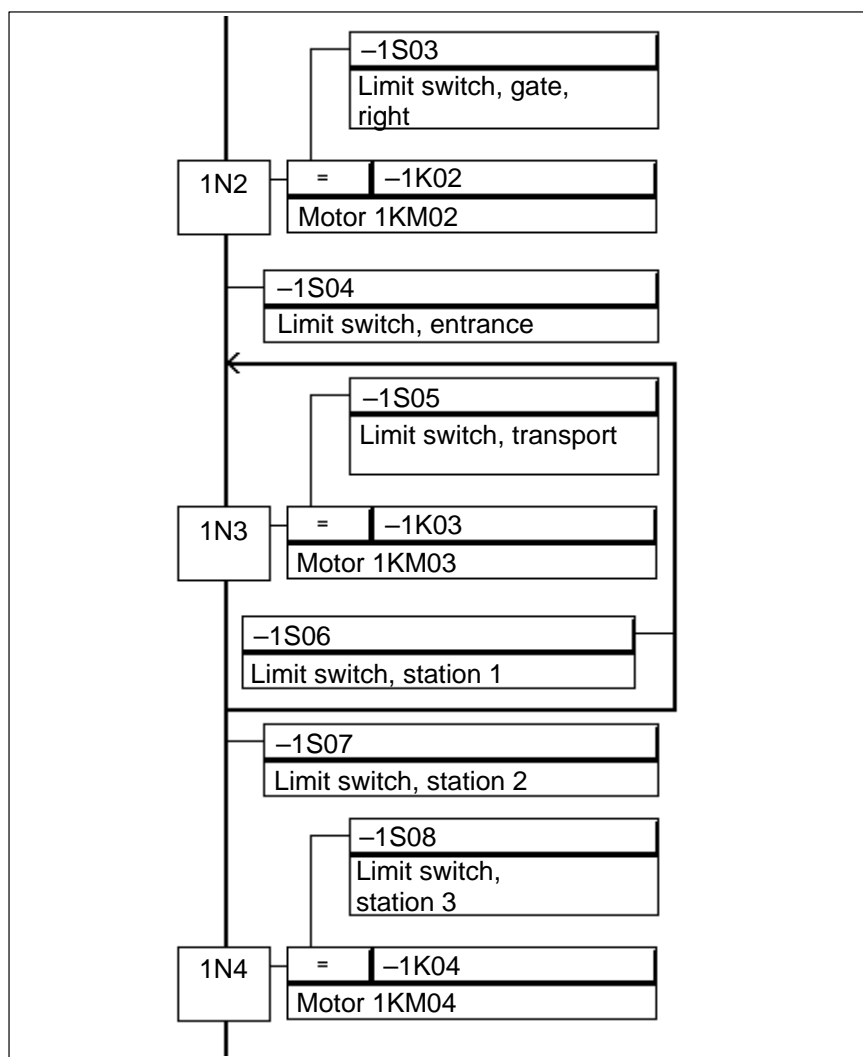
The loop (reverse loop) is represented by an arrow connecting the end of a step with the start of a preceding step.

The command output of a sequence cascade containing a loop differs from that of a linear sequence cascade in that multiple command outputs occur to the steps within the loop.

The loop processing within a sequence represents a reverse jump in step processing.

Example: CL400

Section of loop in sequence cascade; step 1N3 is processed repeatedly.



Subsequent to step 1N3, loop processing is controlled by step-on conditions –1S06 and –1S07. The step-on conditions also define the entry in the branching marker word (–VERZWADR).

The loop will be processed until the step-on condition operand for –1S06 has attained LOW state, and the step-on condition operand for –1S07 has gone HIGH.

If both are HIGH, the –1S07 input will have priority, meaning that loop processing will either be terminated or not executed.

The sequence organization module (e.g. KETTE.PXL) controls, via the contents of Register A, the step processing in the SCHRK01.PXO step module. In accordance with the decision in step 1N3, loop processing is either executed or omitted.

Creating sequence cascade with loop in MADAP mode

See also WinSPS Help, Index, "SFC, Create Sequence".

For the example on the preceding page, the "Create Sequence" function generates the following modules, which are also written into the symbol file:

- KETTEN.PXO Operating mode control, command output
- K_BETRA.PXO sequence operating modes
- SCHRKnn.PXO Step, programming, loop management
- DB_Knn.PXD Data module for sequence cascade
- DBAS.PXD Data module for operating modes

4.5 Jump in MADAP mode

See also WinSPS Help, Index, "SFC, Jump".

The jump is used to advance directly to any desired step of a given sequence. The jump is also used to exit closed structures, such as loops or alternative-branched sequences.

As a consequence of a jump, processing of steps or step sections may be skipped or repeated.

In that case, the jump will be controlled via the step-on conditions.

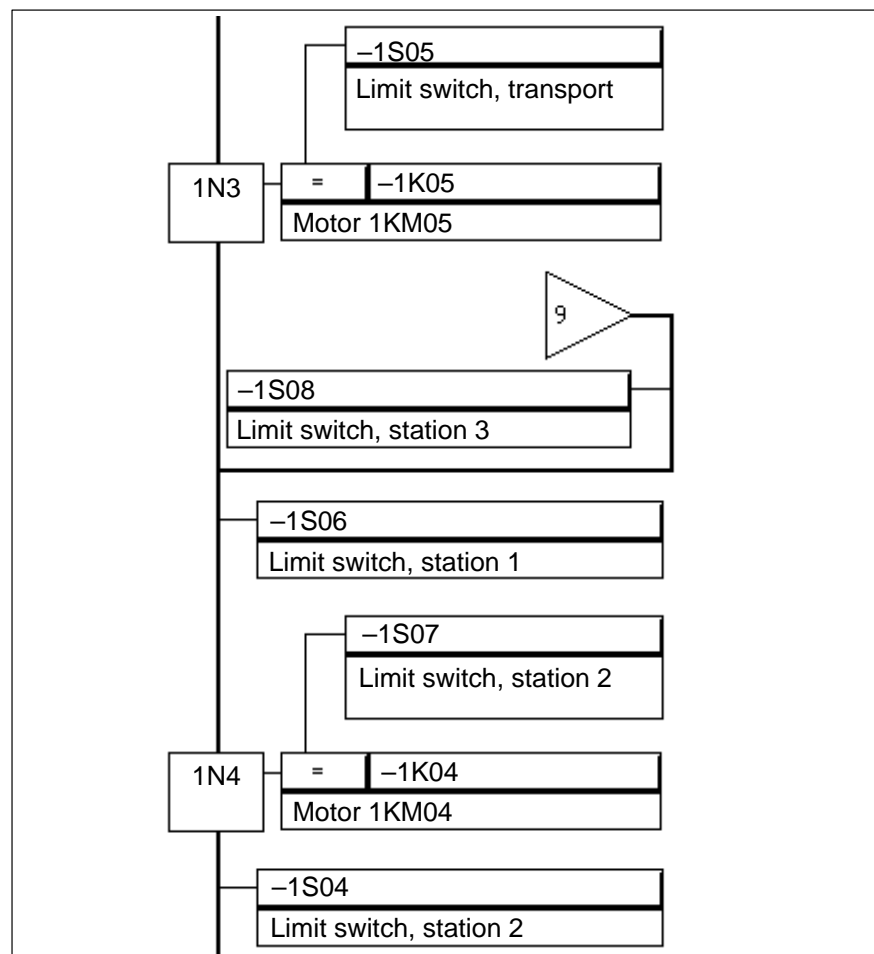
The sequence function chart generates the program organization for a sequence cascade.

The jump is represented by a triangular symbol that is placed between the action and step-on condition of a step.

The command output of a sequence cascade containing a jump differs from that of a linear sequence cascade in that no command output is provided to the skipped steps (steps excluded from processing), and repeated command output is provided to those steps that are repeated.

Example: CL400

Section showing jump from step 1N3 to step 1N9.




Subsequent to step 1N3, the execution of the jump is controlled by step-on conditions –1S06 and –1S08 via the contents of the –VERZWADR branching marker word.

The jump is executed when the state of the –1S06 step-on condition is LOW and that of the –1S08 step-on condition is HIGH.

If both are HIGH, the –1S06 step-on condition will have priority, meaning that the jump will not be executed.

The sequence organization module (e.g. KETTE.PXL) controls, via the contents of Register A, the step processing in the SCHRK01.PXO step module. In accordance with the decision in step 1N3, the jump is either executed or omitted.

 **As the specified destination step number may have changed due to step insertions or deletions in the program, the programmer is required to keep track of the jump destination step number.**

Creating jump in MADAP mode

See also WinSPS Help, Index, “SFC, Jump”.

For the example on the preceding page, the “Create Sequence” function generates the following modules, which are also written into the symbol file:

- KETTEN.PXO Operating mode control, command output
- K_BETRA.PXO Cascade operating modes
- SCHRKnn.PXO Step programming, loop management
- DB_Knn.PXD Data module for sequence cascade
- DBAS.PXD Data module for operating modes

Notes:

5 Step Markers in Sequence Cascades

Step markers (also: "step flags") comprise the major element in conventional sequence cascade programs adhering to the standardized notation as per DIN EN61131. Analogous to their programming, step markers are processed in succession. Processing commences in the first step. Once the last step has been processed, an entire system cycle (plant cycle) is completed.

The state (HIGH / LOW) of the step marker is assigned by the various transitions. In a given sequence cascade, only one step (marker) is active.

The state of the step marker is influenced as follows:

- The step marker of a given step is prepared by the preceding step marker.
- The fulfilled transitions cause the status of the step marker of the step that is thus prepared to go HIGH.
- The activation of the subsequent step again sets the preceding step marker status to LOW.

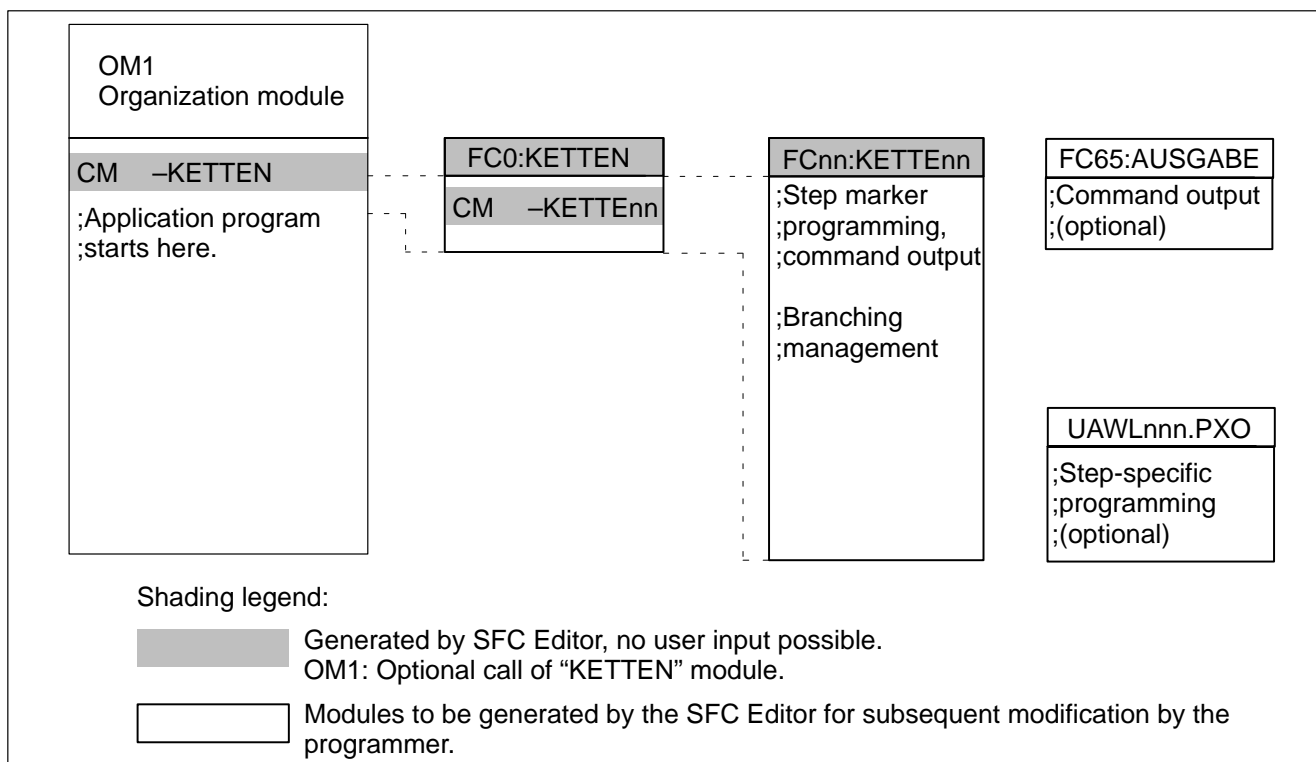
The step marker state can be used either directly for command output functions and/or for a variety of other bit operations.

Example of step marker designation:

- 2N3 identifies the step marker for step 3 in sequence cascade 2.

Modules and step marker structure

Module structure generated by the sequence function chart (SFC).



5.1 Generating a sequence cascade with step markers

The sequence function chart (SFC) generates the program organization for a sequence cascade.



CAUTION

Once generated, the modules can be modified manually for testing purposes. The “SFC, Create Sequence” command will again overwrite the modifications with the function initially programmed under SFC.

Modules and step marker sequence cascade

See also WinSPS Help, Index, “SFC, Create Sequence”.

Absolute module address	Symbolic module address	Functions
FC0	-KETTEN.PXO	Branching management, operating mode control, command output (optionally, separate command output module).
FC1 bis FCnn	-KETTE01.PXO bis -KETTEnn.PXO	Sequence cascades with steps and command output.
FC65	-AUSGABE.PXO	Command output for all sequence cascades. Generated as a selectable option.

The program module named UAWLnnn.PXO is created through the use of the “Insert” PLC command; it is not, however, entered into the symbol file.

Symbol file

With the exception of the UAWLnnn.PXO module, the modules belonging to the sequence cascade structure are written into the symbol file.



CAUTION

Once generated, the module entries in the symbol file can be modified manually. The “SFC, Create Sequence” command will restore the step marker structure (module addresses).

Operand addresses, sample excerpt from CL200

Absolute module address	Symbolic module address	Function
M4.0 thru MX.X	1N0 thru 1NXX	Step marker
M0.0	-LOG0	LOW (logical 0)
M0.1	-LOG1	HIGH (logical 1)

Operand addresses, sample excerpt from CL400

Absolute module address	Symbolic module address	Function
M4.0 thru MX.X	1N0 thru 1NXX	Step marker
SM30.3	-LOG0	LOW (logical 0)
SM31.1	-LOG1	HIGH (logical 1)

KETTEN.PXO cascade module

The KETTEN.PXO module contains the module call for the KETTEnn.PXO sequence modules in accordance with the step marker structure.

Varying conditions in different operating modes are handled in a corresponding number of KETTEnn.PXO sequence modules.

As an option, the module call for the KETTEN.PXO module can be entered into the OM1 organization module at the time of module generation.



CAUTION

When using the automatic generating function, the original programming in the OM1 organization module will be overwritten.

KETTEnn.PXO sequence cascade module

The KETTEnn.PXO module governs all programmed sequence cascades and is invoked by the KETTEN.PXO module.

The KETTEnn.PXO sequence cascade modules contain the instruction list (IL) for the sequence cascades, including all transitions, step markers, actions (command output) and branches.

By default, analogous to a given sequence number an FC with a corresponding number is used (Kette 1 = absolute address; FC1 = symbolic address in KETTE01.PX0).

In accordance with the graphical sequence cascade programming, a separate step marker, which is set or reset, is assigned to each step.

The graphically represented transitions always precede the step markers.

The module contains the command output. As an option, the command output can also be handled by a separate AUSGABE.PXO output module.

The step markers are used to control the action (command output).

Sequence cascade processing

Example: Sequence cascade containing three steps and step markers:

```
;Start step

;Initializing step markers
A      I0.0
S      M4.0           ;Step marker 1N0
R      M4.1           ;Step marker 1N1
R      M4.2           ;Step marker 1N2
R      M4.3           ;Step marker 1N3

;Transition from 1N0 to 1N1
A      I0.1
A      M4.0
R      M4.0           ;Step marker 1N0
S      M4.1           ;Step marker 1N1

;Action of step 1N1
A      M4.1
=      O0.1

;Transition from 1N1 to 1N2
A      E0.2
A      M4.1
R      M4.1           ;Step marker 1N1
S      M4.2           ;Step marker 1N2

;Action of step 1N2
A      M4.2
=      O0.2

;Transition from 1N2 to 1N3
A      I0.3
A      M4.2
R      M4.2           ;Step marker 1N2
S      M4.3           ;Step marker 1N3

;Action of step 1N3
A      M4.3
=      O0.4

;Transition from 1N3 to 1N0
A      I0.4
A      M4.3
R      M4.3           ;Step marker 1N3
S      M4.0           ;Step marker 1N0
```

AUSGABE.PXO output module

The generation of this module is optional. It contains the command output shared by all sequence cascades. The assignment of step actions is handled by the step markers.

The module call for the AUSGABE.PXO output module is not generated automatically. It must be inserted manually into the program.

UAWLnnn.PXO module-specific step programming

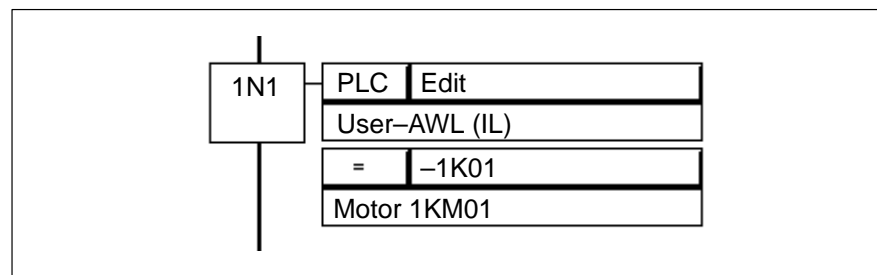
See also WinSPS Help, Index, "SFC, Create Sequence".

The module allows for instructions that are additional to the action of a given step.

Using the **Toolbox / Insert Action / PLC Instructions** menu sequence, the sequence function chart generates, as a template, the UAWLnnn.PXO module. The step action is thus supplemented by the PLC Edit action element.

Because an existing UAWLnnn.PXO module is not overwritten, a new UAWLnnn+1.PXO module is created. The module is subject to suitable adaptation by the user. It is not entered into the symbol file.

The module can be modified by means of the EDIT command.

Example: PLC Edit action element

The program from the UAWLnnn.PXO module is linked with the step marker 1N1 in the KETTE01.PXO module.

Example of UAWLnnn.PXO

```
A      -1S10
=      M23.0
```

Link to step

```
;Action step 1N1
A      -1N1
A      -1S10
=      M23.0
```

6 Structural Elements of Sequences w/ Step Markers

To construct a sequence cascade with the use of the step marker technique, the following structural elements are available:

- Linear structure
- Alternative branch
- Parallel branch
- Loop
- Jump

6.1 Linear sequence cascade in Step Marker technique

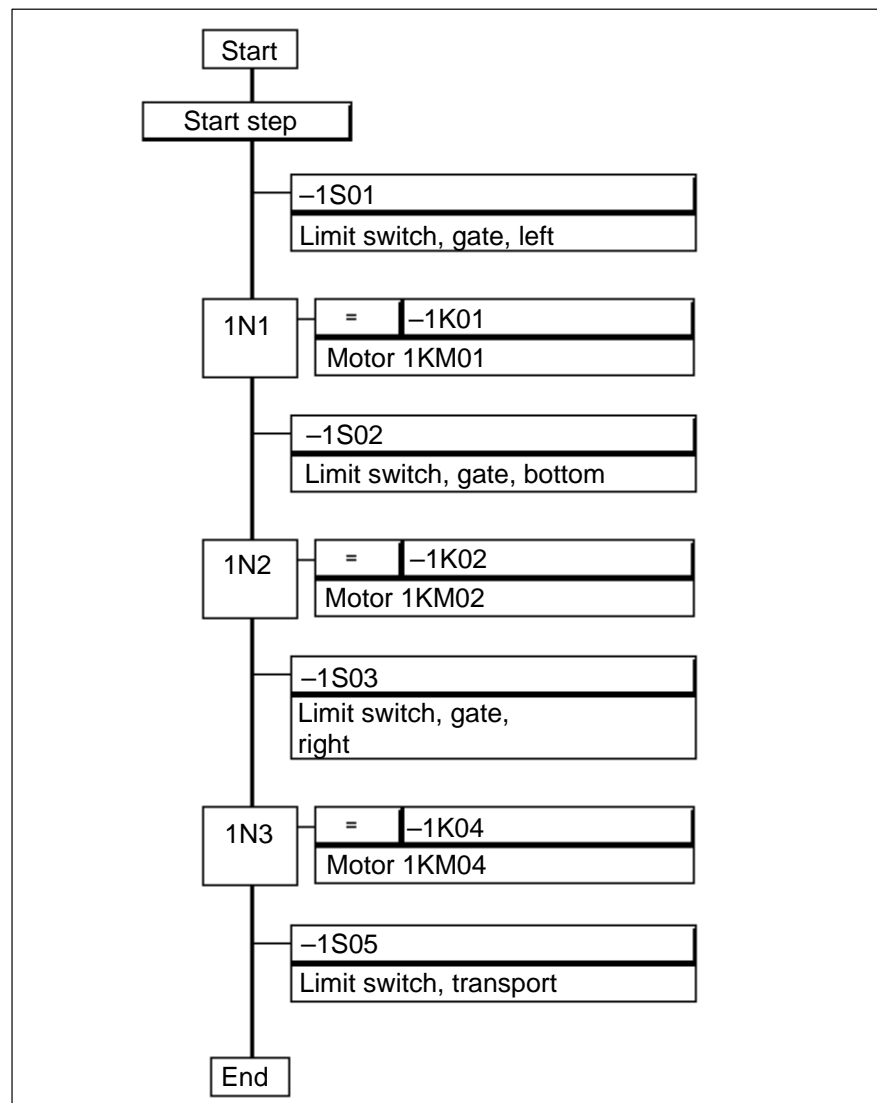
See also WinSPS Help, Index, "SFC, Step Marker Technique".

The linear sequence cascade processes the steps, in accordance with their graphical diagram, from the first step to the last in a contiguous ascending sequence.

The sequence function chart generates the program organization for a sequence cascade.

Example

Linear sequence cascade 1 with start step 1N0, den Schritten 1N1, 1N2 und 1N3.



Creating linear sequence cascade in Step Marker technique

See also WinSPS Help, Index, "SFC, Create Sequence".

For the example on the preceding page, the "Rebuild All" function generates the following modules, which are also written into the symbol file:

- KETTE01.PXO Step programming and command output
- KETTEN.PXO Call for sequence cascade
- AUSGABE.PXO Command output in separate program module, selectable as an option.e-branched

6.2 Alternative-branched sequence cascade in Step Marker technique

See also WinSPS Help, Index, "SFC, Branch".

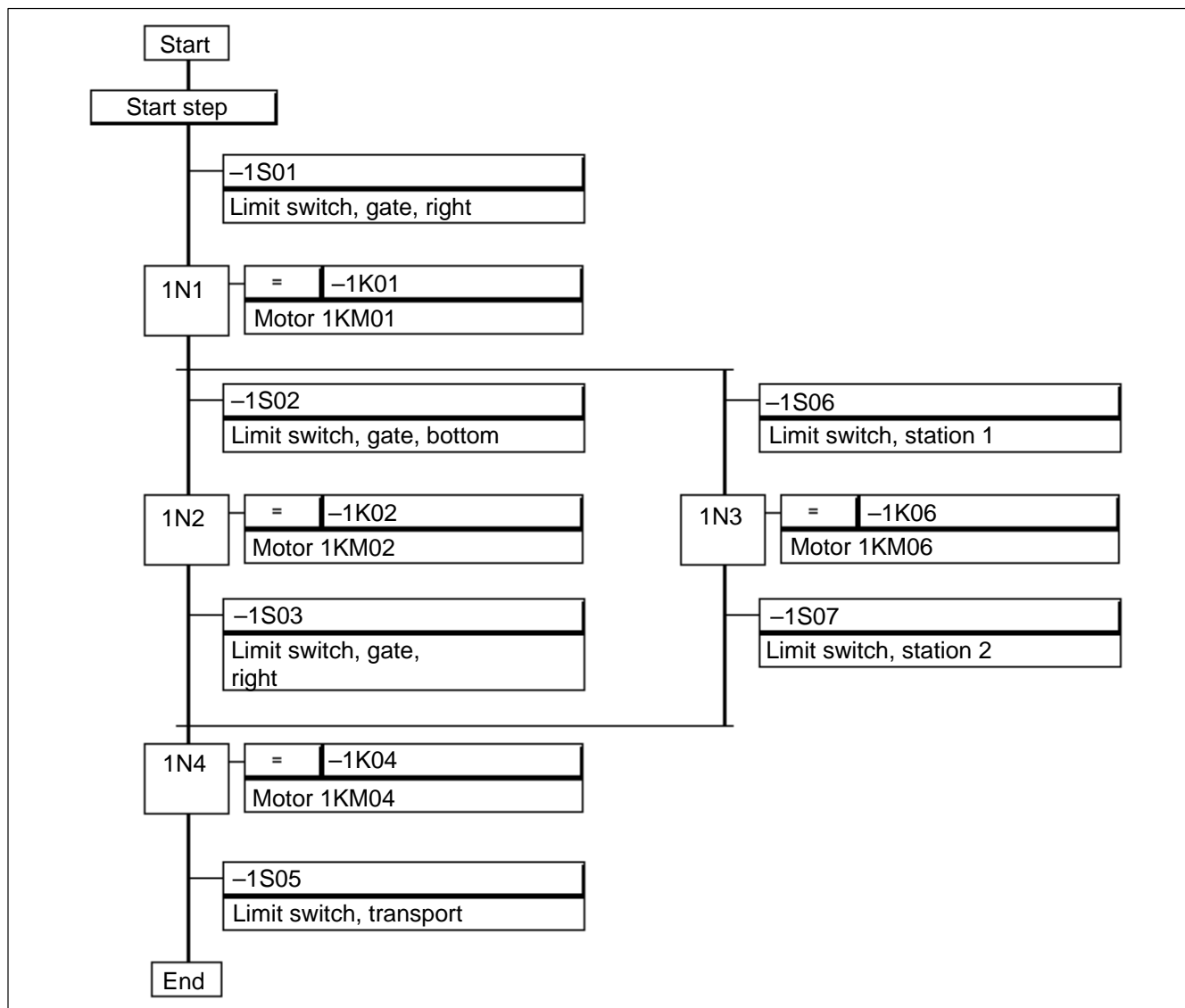
The sequence function chart generates the program organization for a sequence cascade.

In alternative branches, only one branch is processed at any time. The transitions within the alternative branch define which branch is to be processed.

The alternative branch is denoted by single horizontal lines.

Example

Alternative-branched sequence cascade w/ alternative steps 1N2 and 1N3.



Subsequent to processing Step 1N1, the alternative branching causes processing to continue with Steps 1N2 or 1N3.

Step marker 1N1, in conjunction with transition –1S02, initializes either step 1N2 or, together with transition –1S06, it initializes step 1N3.

If the state of both transitions is HIGH, the initialization of step 1N2 will have priority.

Step 1N4 is initialized either by step 1N2 in conjunction with transition –1S03 or by step 1N3 in conjunction with transition –1S07.

Creating alternative-branched sequence cascade in Step Marker technique

See also WinSPS Help, Index, "SFC, Create Sequence".

For the example on the preceding page, the "Create Sequence" function generates the following modules, which are also written into the symbol file:

- KETTE01.PXO Step programming and command output
- KETTEN.PXO Module call for sequence cascade
- AUSGABE.PXO Command output in separate program module as a selectable option

6.3 Simultaneous-branched sequence cascade in Step Marker technique

See also WinSPS Help, Index, "SFC, Branch".

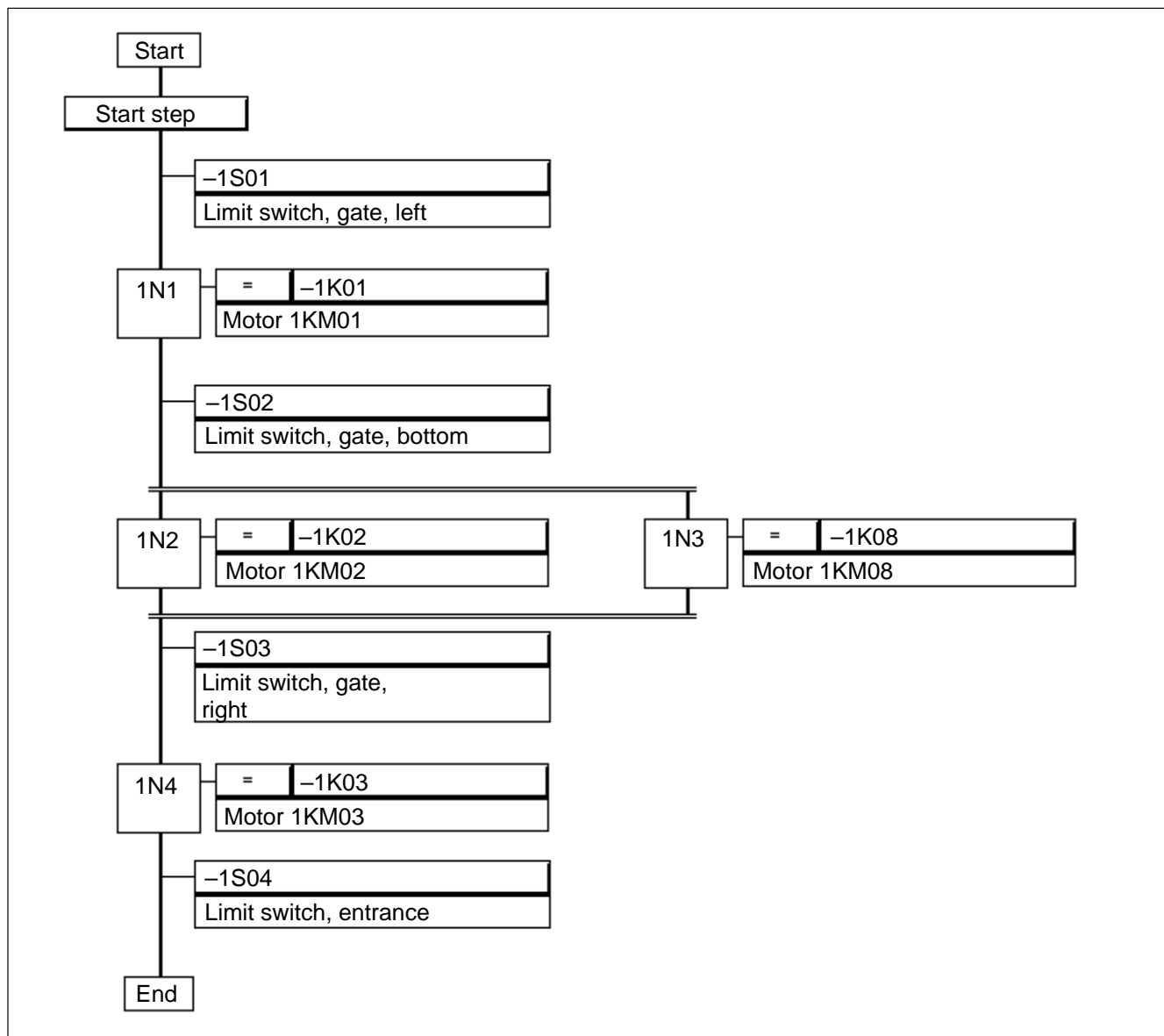
In a simultaneous branch, all simultaneous branches are processed without exception.

The sequence function chart generates the program organization for a sequence cascade, adding the required number of simultaneous branches.

The simultaneous branch is denoted by parallel horizontal lines.

Example

Simultaneous-branched sequence cascade with simultaneous branches 1N2 and 1N3.



Subsequent to step 1N1, the simultaneous branch commences with steps 1N2 und 1N3.

The step marker 1N1 simultaneously initializes steps 1N2 and 1N3; the simultaneous branch is processed at the same time.

The simultaneous branch is again merged in step 1N4 by the AND-query of step markers 1N2 and 1N3. Step 1N4 will be initialized only if both simultaneous branches have been processed and transition –1S03 fulfilled.

Creating simultaneous-branched sequence cascade in Step Marker technique

See also WinSPS Help, Index, "SFC, Create Sequence".

For the example on the preceding page, the "Create Sequence" function generates the following modules, which are also written into the symbol file:

- KETTE01.PXO Step programming and command output
- KETTEN.PXO Module call for sequence cascade
- AUSGABE.PXO Command output in separate program module as a selectable option

6.4 Sequence cascade loop in Step Marker technique

See also WinSPS Help, Index, "SFC, Loop".

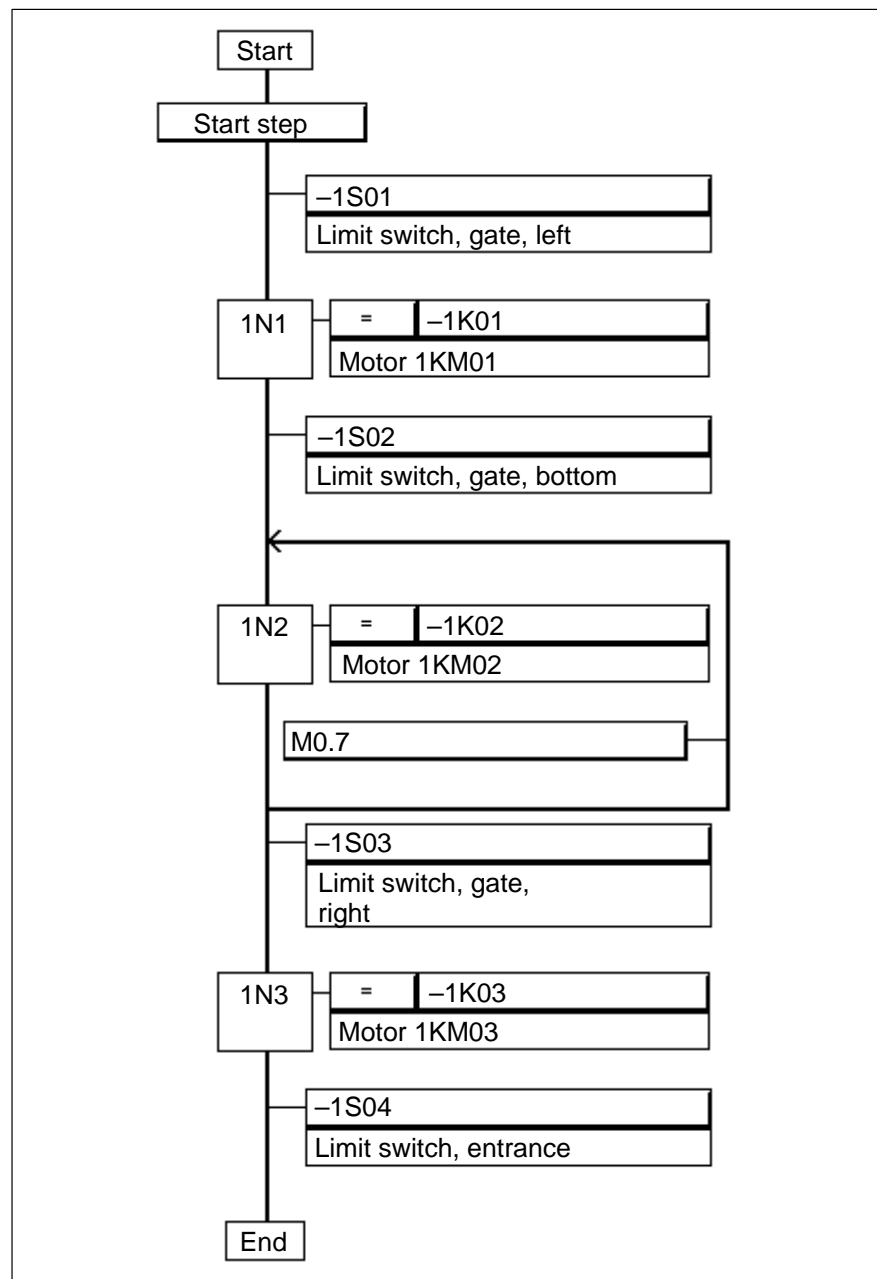
Loop processing is controlled by the loop transition. It causes repeated processing of identical steps or step sections.

The sequence function chart generates the program organization for a sequence cascade.

The loop is represented by an arrow connecting the end of a step with the start of a step.

Example

Loop in sequence cascade with loop step 1N2.



The loop processing for step 1N2 is controlled by the transitions of marker M0.7 and input –1S03.

Subsequent to step 1N1, the program part begins with loop processing of step 1N2. Step 1N2 is followed by checking the state of the loop condition.

If marker M0.7 is HIGH and input –1S03 is LOW, the loop will be processed. The step marker of the last step in the loop is reset, and the step marker of the first step in the loop is initialized. The loop cycle begins.

If the state of the transitions of both M0.7 and –1S03 is HIGH, transition M0.7 will have priority.

If the state of marker M0.7 is LOW, no additional loop cycles are processed, step marker 1N3 is initialized by step 1N2, which concludes loop processing.

Creating sequence cascade with loop in Step Marker technique

See also WinSPS Help, Index, "SFC, Create Sequence".

For the example on the preceding page, the "Create Sequence" function generates the following modules, which are also written into the symbol file:

- KETTE01.PXO Step programming and command output
- KETTEN.PXO Module call for sequence cascade
- AUSGABE.PXO Command output in separate program module as a selectable option

6.5 Jump in sequence cascade in Step Marker technique

See also WinSPS Help, Index, "SFC, Jump".

The jump is used to advance directly to any desired step of a given sequence. The jump is also used to exit closed structures, such as loops or alternative-branched sequences.

As a consequence of a jump, processing of steps or step sections may be skipped or repeated.

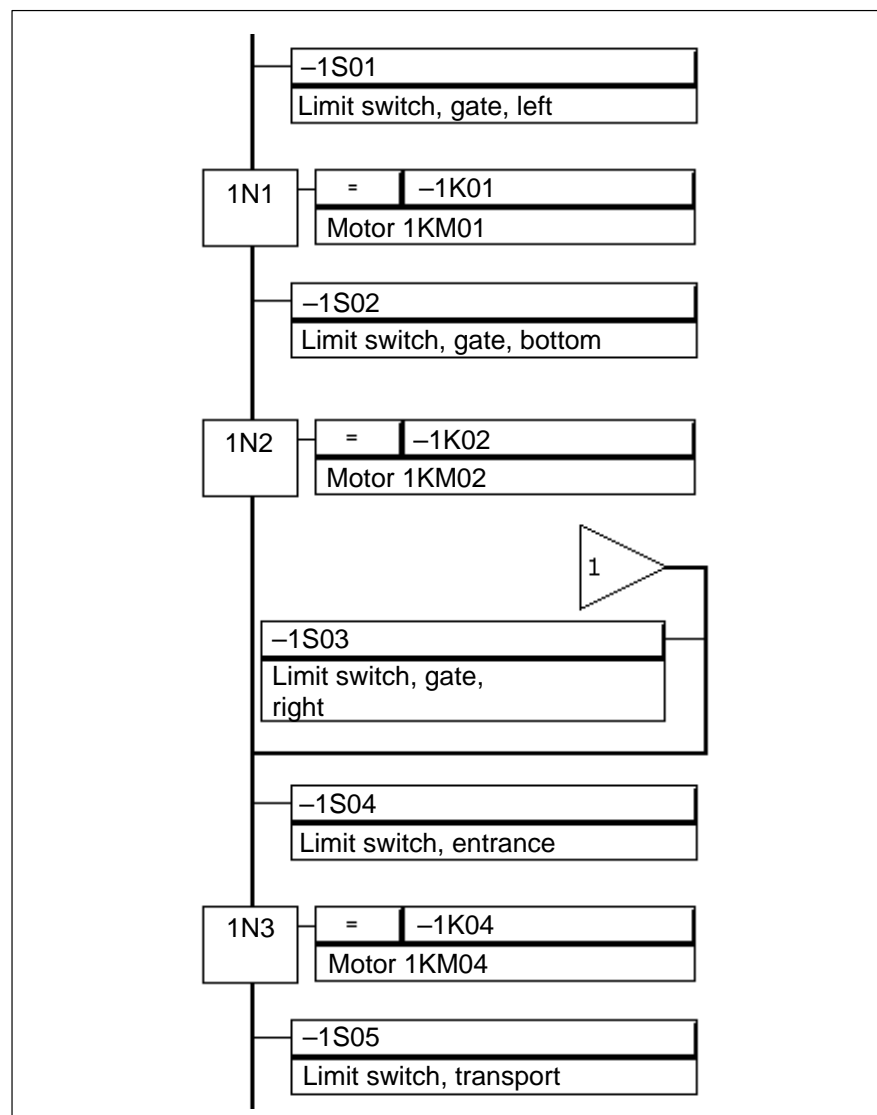
The jump will be controlled by the transition.

The sequence function chart generates the program organization for a sequence cascade.

The jump is represented by a triangular symbol that is placed between the action and transition.

Example

Jump from step 1N2 to step 1N1.




The jump is programmed in step 1N2. The state of the transition to the jump is checked.

If the state of the condition –1S03 is HIGH, step 1N1 is initialized. Step marker 1N2 is reset.

If the state of condition –1S03 is LOW and that of –1S04 is HIGH, the step marker of step 1N3 is set, and no jump is executed. This causes the step marker 1N2 to be reset.

If the state of the transitions of both –1S03 and –1S04 is HIGH, transition –1S03 will have priority.

 **As the specified destination step number may have changed due to step insertions or deletions in the program, the programmer is required to keep track of the jump destination step number.**

Creating sequence cascade with jump in Step Marker technique

See also WinSPS Help, Index, “SFC, Create Sequence”.

For the example on the preceding page, the “Create Sequence” function generates the following modules, which are also written into the symbol file:

- KETTE01.PXO Step programming and command output
- KETTEN.PXO Module call for sequence cascade
- AUSGABE.PXO Command output in separate program module as a selectable option

Notes:

A Appendix

A.1 Abbreviations

Abbr.	Explanation
1N1	Step designation
BEFA	MADAP command output marker
BT	Operator terminal
DM	Data module
FC	Program module
LOG1	Logical 1 (high)
LOG0	Logical 0 (low)
OM	Organization module
S+1	Step-on designation
SCHRK	Step programming for sequence cascade
SFC	Sequential function chart (graphical sequence programming)
StoeM	Fault marker
UAWL	User Instruction list
UET_Halt	Halt monitoring time
VERZWADR	MADAP branch address
WinSPS	PLC programming software
WSB	MADAP step-on condition marker
WZT	Wait interval timer
WZT_Halt	Halt wait time

A.2 Index

- A**
 Action, Edit, 3-10, 5-6
 Active step, 3-12
 Alternative branch, 2-5
 AUSGABE.PXO, 5-2, 5-6
 Output module, 5-4
 Automatic mode, 2-7, 3-16
- B**
 Basic start conditions, 3-20
 BEFA, 3-4, 3-6, 3-7, 3-14, 3-18
 Branch address, 3-14
 Branching address, 3-4
 BT-MADAP, 2-1
- C**
 Cascade data module, Generate, 3-17
 Cascade mode module, 3-8
 Cascade module, 3-5, 5-4
 Cascade number, 3-6
 Command output, 2-8, 2-9, 3-4, 3-14
- D**
 Data module
 Diagnostics, 3-13
 Sequence function chart, 3-13
 DB_Knn.PXD, 3-12
 DB120.PXD, 3-13
 DBAS.PXD, 3-8, 3-13
 Diagnostics, 2-8
 CL200, 3-13
 Documentation, 1-5
- E**
 EMERGENCY-STOP devices, 1-4
- F**
 Fault diagnostics, 2-6
 Fault indication, 2-8
 Fault marker, 3-14
 Function modules, 3-1
- H**
 HalbAuto, 3-4
 Halt, 3-17
 Halt acknowledgement, 3-16
- I**
 Inching mode, 3-16
 Insert action, 3-10
- J**
 Jump, 2-5
- K**
 K_BETRA.PXO, 3-3, 3-8
 KETTE.PXL, 3-3
 KETTE200.PXL, 3-3
 KETTE700.PXL, 3-3
 KETTEN.PXO, 3-5, 5-4
 KETTEnn.PXO, 5-4
- L**
 Linear step sequence, 2-5
 Loop programming, 2-5
- M**
 MADAP
 Alternative branch, 4-4
 Branch jump, 4-10
 Generated modules, 3-3
 Jump, 4-10
 Linear sequence cascade, 4-2
 Loop, 4-8
 Parallel branching, 4-6
 Program structure, 3-1
 Programming structure, 2-1
 Sequence step, 2-11
 Software package, 3-10
 Specific markers, 3-14
 Standardized function modules, 3-2
 Structural elements, 4-1
 Symbol file, 3-4
 UAWLnnn.PXO, 3-10
 Mandatory fault acknowledgement, 3-18
 Manual mode, 3-16
 MMI-MADAP, 2-1
 Mode start, 3-16
 Modifications, 1-5
 Monitoring time, 3-12, 3-14
 Monitoring time HALT, 3-15
 Movement key, 3-7
- N**
 Number of steps, 3-12
- O**
 Operand addresses, Entry in symbol file, 3-4, 5-3
 Operating mode, 2-7, 3-12
 Automatic Sequence, 2-7
 Manual (setup), 2-8
 Priority, 2-7
 Semi-automatic, 2-7
 Single step, 2-7
 Operating mode bits, 3-9, 3-16
- P**
 Parallel branch, 2-5
 PLC commands, 3-10
 Process sequence, Graphical representation, 2-9
 Process sequences, 2-6
- Q**
 Qualified personnel, 1-2

R

Reset, 3-17

S

S+1, 3-16

Safety instructions, 1-3

SCHRKnn.PX0, 3-6

Sequence cascade

Call-up, 3-5

Data module, 3-12

Fault entries, 3-12

Principle, 2-5

Sequence cascades, Status entries, 3-12

Sequence control system, 2-6

Sequence organization modules, 3-10

Set step, 3-16

Special-purpose cascades, 3-20

Standard operation, 1-1

Step, 2-6

Step , Branching, 3-19

Step element, 2-9

Step flag, 5-1

Step marker, 5-3

Sequence step, 2-12

Structural elements, 6-1

Symbol file, 3-3

Step markers, 5-1

Alternative branch, 6-4

Example, 5-5

Generated modules, 5-2

HIGH/LOW state, 5-1

Jump, 6-10

Linear sequence cascade, 6-2

Loop, 6-8

Module structure, 5-1

Output module, 5-6

Processing example, 5-5

Sequence cascade module, 5-4

Simultaneous branch, 6-6

State, 5-3

Symbol file, 5-2

UAWLnnn.PX0, 5-6

Step module, 3-6

Step number, 3-12

Step processing, 3-7

Step-on action, 3-17

Step-on condition, 2-9

StoeM, 3-4, 3-14

StoeM, fault marker, 3-4

Synchronize, Automatic, 3-17

Synchronized steps, 3-12

T

Tandem limit switch monitoring, 3-20

Trademark, 1-5

Transition, 2-9

U

UAWLnnn.PX0, 3-3, 3-10, 5-2, 5-6

UET_Halt, 3-4, 3-15

UETKONTR, 3-4

V

Version, 1-5

VERZWADR, 3-4, 3-14

W

Wait time, 3-12

Wait time HALT, 3-15

Wait time marker, 3-15

Watchdog timer, 2-6

WSB, 3-4, 3-6, 3-7, 3-14, 3-18

WSB does not reset BEFA, 3-18

WZT, 3-4, 3-15

WZT_Halt, 3-4, 3-15

Notes:

Bosch-Automationstechnik

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Industriehydraulik
Postfach 30 02 40
D-70442 Stuttgart
Telefax (07 11) 8 11-18 57

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Fahrzeughydraulik
Postfach 30 02 40
D-70442 Stuttgart
Telefax (07 11) 8 11-17 98

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Pneumatik
Postfach 30 02 40
D-70442 Stuttgart
Telefax (07 11) 8 11-89 17

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Montagetchnik
Postfach 30 02 07
D-70442 Stuttgart
Telefax (07 11) 8 11-77 77

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Telefax (0 60 62) 78-4 28

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Schraub- und Einpreßsysteme
Postfach 11 61
D-71534 Murrhardt
Telefax (0 71 92) 22-1 81

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Entgrattechnik
Postfach 30 02 07
D-70442 Stuttgart
Telefax (07 11) 8 11-34 75

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Schulung AT/VSZ
Berliner Straße 25
D-64711 Erbach
Telefax (0 60 62) 78-8 33

Technische Änderungen vorbehalten

Ihr Ansprechpartner

BOSCH



Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Telefax (0 60 62) 78-4 28