IndraStep - user-oriented step sequences with diagnoses

Application Manual

SYSTEM200

DOK-CONTRL-SPS*ISTEP**-AW01-EN-P

Title IndraStep - user-oriented step sequences with diagnoses

Type of Documentation Application Manual

Document Typecode DOK-CONTRL-SPS*ISTEP**-AW01-EN-P

Internal File Reference

- Doku/29006801
- Drawing number: 120-0400-B336-01

Purpose of Documentation This documentation describes …

- describing the functions of IndraStep and the application of INDRAMAT step sequence IndraStep

Record of Revisions

Description	Release Date	Notes
DOK-CONTRL-SPS*ISTEP**-AW01-EN-P	08/00	First edition

Published by Rexroth Indramat GmbH
Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main

Telephone 09352/40-0 • Tx 689421 • Fax 09352/40-4885

http://www.rexroth.com/indramat

Dept. ECS1 (MH/RW)

Note This document has been printed on chlorine-free bleached paper.

Inhaltsverzeichnis

1 Fundamentals

1.1 Purpose

The use of step sequences for SPS programming allows easier and clearer structuring of user programs. For this reason, users have been accepting step sequences more and more in the last years. Moreover, it has been integrated in the standard EN 6-1131-3. The generous support and the comfort in program creation are now a standard feature.

This support is also expected for

- commissioning and setup
- monitoring
- analyses and error localization.

Supporting users in these areas results in a saving of time and thus costs. For this reasons Rexroth INDRAMAT has developed "IndraStep". "IndraStep" are user-oriented step sequences that even go beyond the standard defined in EN 6-1131-3.

It also represents an extension of the SFC (sequence language, sequential function chart). The basis for understanding this manual is the knowledge of the fundamentals of the sequence language SFC and its representation in the Indramat System 200 (Fundamentals see DOK-CONTRL-SPS*PROVxx-ANW1-EN-P).

Advantages of Indramat control systems with WinHMI

Sytem200 and MT-CNC

The Rexroth INDRAMAT System200, or the MTC-CNC respectively, have proved successful with a tool management integrated in the NC process and the optimal communication for synchronization between NC processes and mechanisms (pure SPS processes), making complex SPS programs for control superfluous.

Fig. 1-1: Control and display devices

ProVi

This allows optimum diagnosis of the cyclical SPS program section necessary for the control of NC processes.

IndraStep

Step sequences with operating modes and automatic generation of diagnoses for sequence control.

WinHMI Optimum simultaneous operation and diagnosis:

- of NC processes
- of the cyclical SPS program section for control of NC processes with ProVi
- of the IndraStep step sequences for the control of SPS sequences

Overview of ProVi

The diagnosis and message system ProVi can be operated on HMI-devices. These belong to the device family MTC200. Two types of HMI devices exist.

The basic firmware and software on the devices has to be of version V18 or higher. As of version V01 the supplementary software WinHMI (for PC-based devices) as well as the screen manager (for small operating devices) support ProVi.

Origin of the visualization data Representations of the installation, diagnosis overviews, etc. offers rapid, comprehensive and sufficient information on the state of the installation. Information for the display is entered directly during program creation. To do so, predefined diagnosis methods are used. The diagnostic system automatically extracts the messages as visualization data from the SPS program. Thus, the entry of diagnosis data at other points is no longer required, so that the data consistency is assured.

Representation of visualization data ProVi is adapted to the application in interlinked installations, as well as in transfer lines. As visualization device a PC, e.g. BTV20, or a small operating device, e.g. BTV05, can be used. The visualization is divided in different levels, such as installation section overview, diagnosis overview, messages and similar. For automatic recognition of the installation structure defined directives must be fulfilled already during configuration and programming. The module configuration list contains the required data. It allows the automatic assignment of the information in the system to the "correct" points.

Overview of ProVi message types on WinHMI

Messages and faults, requested diagnosis data, are displayed on the PC-based variant of ProVi under WinHMI. A distinction is made between eight message types:

- System error messages
- Start requirements
- CNC faults
- Sequence faults (IndraStep)
- General faults (SPS)
- Messages (CNC/SPS)
- Safety information
- Setup diagnosis

ProVi diagnosis data are created both in the SPS and in the CNC.

System error messages, CNC faults and messages are generated by the system itself.

Defined diagnosis arrays and function blocks are provided for almost all SPS diagnosis data.

Sequence faults (IndraStep) are an exception.

An overview of IndraStep

The SPS controlled machine sequences are programmed in function blocks activated by the cyclical program.

By selecting IndraStep the function blocks are automatically assigned to a control and status interface. Several step sequences can be programmed in an SPS program. Each IndraStep step sequence has an interface which allows to control it practically like a CNC.

The programmer programs the individual machine movements as actions (e.g. one action for "Open door" and one action for "Close door") in IndraStep step sequence blocks. These actions will be activated by steps. Then, for example, the programmer only has to integrate the action "Open door" in the step "Open door to move workpiece in". In a later step, for example, "Open door to move workpiece out" the same action "Open door" can be integrated again (and thus the same logic).

In case of an error while moving in the workpiece, the diagnosis message "Action -Open door- in the step -Open door to move workpiece in- has failed" will be displayed. The fault messages can show the failed element with time of the error, i.e. the saved states, and online, i.e. the actual states, by means of a criteria analysis (more information on the WinHMI diagnosis screen).

The logic, required for the single step and manual operating modes, is also added automatically to the step sequence during compilation, thus reducing the need for further programming.

In this step sequence all motions are available in steps. For manual mode there is hardly any work left for the programmer to make it possible controlling these movements via visualization devices. If the manual mode requires other conditions than the automatic sequence, these will also be programmed in the movement action. As a result, these condition do not have to be programmed elsewhere.

Step-transition relation

A movement element is programmed in an action in relation to the movement, independent of the conditions of the sequence. The step sequence is responsible for the timing, according to which the movement element shall be executed within the entire sequence. Quitting a step is programmed in transitions. A transition, in which the contacts representing the upper end position of the door are inquired, is assigned to a door opening step. When the conditions of the transition are fulfilled, the step is completed before the transition and the next step is activated.

The IndraStep step sequence includes several diagnosis methods which allow unambiguous error determination without having to realize more programming.

1.2 Comparison of programming techniques

Generally, the mayor part of an SPS program for machine control consists of sequences and states. To create such a program the SFC editor (step sequence programming) is suited especially.

Programming without step sequences

If sequences shall be executed by means of ladder diagram, function block or command list programming, the user has to program the reaching of each state and the preparation of the next sequence individually by forming and using flags. Here, the realization of a faultless automatic mode involves a lot of configuration and programming work. There are additional activities required for the execution of operating modes and error evaluation which result in at least the same amount of work as the programming itself.

Programming with standard step sequences (SFC, EN 6-1131-3)

In the SPS programming instruction, the standard step sequence is described in the chapter "Sequence language, SFC, SEQUENTIAL FUNCTION CHART".

The SFC elements allow the creation of sequences in function blocks and programs. Sequences contain steps and transitions which are connected with each other through oriented lines. Moreover, alternative OR-branches and parallel AND-branches can be realized in the sequences.

Considerable savings in configuration and programming are the advantages of the application of SFC elements with standard step sequences. In addition, this procedure results in clearer structures, better legibility and handling during programming, error localization and setup. However, the programmer has to bear in mind the operating modes and the error evaluation inside the program when using standard step sequences. This presents the disadvantage that the programmer might not have integrated a possible error state, or entered an error text incorrectly or not at all.

User-oriented step sequences (IndraStep)

IndraStep is an extension of the standard step sequence SFC. IndraStep aims at keeping to the Standard EN 6-1131-3. For this purpose, the operating mode and the diagnosis/criteria analyses have been added to the standard step sequence.

Operating modes

The normal program execution is programmed by the programmer with the means of the standard step sequence in the standard SFC editor. The system automatically adds the operating mode control and the error evaluation to the program.

At the same time, data for WIN-HMI (standardized user interface for automated manufacturing) and ProVi (diagnosis and message system for HMI-devices) are created for status and error visualization and for handling. ProVi can also be run in combination with IndraStep.

Diagnosis/criteria analysis

Diagnosis data and texts are generated automatically during compilation compiling from step sequence contents . The diagnosis texts are taken from the program and allocation comments. Programmers who stick to the predefined programming technology and make use of the commenting do not have any additional work for the creation of the error evaluation. The diagnosis data are generated in connection with the WIN-HMI visualization and ProVi.

Advantages of IndraStep

- IndraStep is a programming technology which - compared with the standard step sequence - offers the programmer a method to focus on the essentials of programming.
- The operating modes are considered automatically. The user only has to program a correct automatic sequence.
- Step sequence and event monitoring has a direct effect on the operating mode control of the step sequence, as a result the programmer does not have to program an error management.

The diagnosis data for ProVi and IndraStep are extracted directly from the program. A programmer who uses the possibilities of program commenting does not only have the known documentiation advantages of this commenting, but also the advantage of having already entered the diagnosis texts. These advantages show effect, for example, when the program is modified during a setup under pressure of time.

1.3 Short description

The elements of the step sequence and their behavior in the different operating modes are described in short below these lines.

Step

Each step describes a machine status. At the beginning of a step the actions are activated. They will be processed until they can be considered fulfilled.

Action

Actions are activated by steps. In these steps, the output elements are activated depending on the enables and operating mode states.

Activation of actions depending on the operating modes

The activation of the action is realized according to the operating mode set in the step sequence. In automatic mode the activation is determined only by the state of the step, in manual mode by the active operating masks of the connected operating devices.

In an action only the conditions have to be programmed which are directly connected to the outputs to be controlled. The logic which controls the execution of the step sequence is programmed in the transition condition. Thus, the relation between activated steps and machine status will be established.

Behavior of the actions in dependence of the step sequence state

The outputs activated in the actions are influenced by the conditions programmed by the user in the operating mode control.

Program relation between end of action and transition result

Upon completion of an action, or switching the step on when the transition is fulfilled, the ladder diagram line of the output activation is separated by connection of the operating mode control. Thus, the direct relation to the transition is guaranteed, i.e. the activated output does not have to be reset when the transition is completed.

Transition

In the transition the user can program the conditions which introduce a transition between states. In case of step sequences with operating modes the transition conditions are directly combined with the actions actively executed at that moment. In the transition the conditions are programmed which complete the execution of the actions. It describes the result of a state, such as activation of a motor (action) and switching off of the motor when reaching the end position (transition).

Minimal step time

In addition to the conditions inserted directly by the user into transition generation, switching the transition on to the next step can be suppressed by a time-related to the step (T_min).

Maximum step time

In these freely definable conditions a maximum time monitoring (T_max) can be programmed for each step. Time values can be programmed as variable(s).

Dynamical waiting step

The maximum time can also be started with separate conditions.

Monitoring

Monitoring means the condition programmed by the user which shall switch the step sequence to a stopped state and at the same time transmit messages to the diagnosis system.

Global and local monitoring

General conditions and states can be monitored which are programmed globally once for the entire step sequence, or defined conditions which act step-specifically.

Step control signals

The control commands for the step sequence are programmed by the user by means of LD via a defined interface.

Starting/restarting the step sequence

Is effective in all operating modes. The positive edge is evaluated. This control signal serves for "restarting" when the step sequence is stopped and no error or stop query is active. In manual mode the stop state is deleted.

Stopping the step sequence

This function is effective in all operating modes and will be evaluated statically. The step sequence is brought to the stopped state when the command ’TRUE’ is activated.

Operating mode selection

Selection of the operating mode Auto, Single step, Manual, Synchronizing.

Deleting errors

The only effect is that error messages will be deleted.

Reset

Depending on the operating mode (Manual or Automatic) the effect of "RESET" differs. In the automatic mode the step sequence is set to the initial step. In manual mode the steps active at that moment will be deactivated.

Status messages of the step sequence

Faulted step

In case of an error the incorrect step will be displayed.

Fault

The step sequence is switched to an error state by an immediate stop monitoring or a time monitoring.

Internal error

An invalid step selection during synchronization has caused an internal error.

Stopped step sequence

The step sequence is stopped by the user or the program, or an error condition prevails.

Internal command and status signals

The step sequence can also be activated internally, except for the operating mode selection.

Internal control possibility of the step sequence

This logic, required for the single step mode and manual mode, is also added automatically to the step sequence during compilation, thus reducing further programming work.

The logic for step sequence synchronization can be programmed inone or several actions.

Synchronization

This functionality is used for example for a step sequence reverse and for monitoring tasks in which the machine is not only stopped with an error message, but in which the processing shall be continued with other steps, depending on the states. These are not linked directly by transitions to the active step. The combination of these steps by means of transitions is possible, but for reasons of clearness not always reasonable.

1.4 Operating mode behavior of the step sequence

In the following section the basic behavior of the IndraStep step sequence will be described. For a detailed description of the operating modes see chapter "6 IndraStep operating modes

Starting the step sequence

The step sequence is started by a positive edge at a control input and stopped by a positive edge on a control input. An active stop does not allow a start.

The switch-on state is, on the basis of the standard SFC, the started automatic mode. Steps will be processed actively until the subsequent transition condition is fulfilled. The fulfilled transition switches over to the next step by deactivating the actually active step and activating the step after the transition (via AND-branch various steps).

Immediate stop monitoring

All immediate stop monitorings programmed by the user are active.

Time monitoring

Minimal and maximal times are evaluated.

Single step

The step sequence functions like in the normal automatic mode. A change between steps only takes place when a start pulse is transferred to the step sequence in addition to the fulfilled transition condition.

Transition result saved

Since the contacts in the transition generation can not only be reached states but also time-limiteded signals the fulfillment of the transition is saved until the next start pulse.

Immediate stop monitoring

The monitoring functions are active like in the automatic mode.

Time monitoring

A programmed minimum time has the same effect as in normal automatic mode.

Maximum waiting time for fulfilled transition

While waiting for a start pulse the maximum time monitoring of the step is completed with the fulfillment of the transition.

Manual mode

When switching on the manual mode, all steps and the thus activated actions will be deactivated. The processing of steps can also be started by an external request. The external request is generally transmitted from one or several operating devices which are designed as key assignments in masks.

Realize several movements simultaneously

This function allows simultaneous activation of several steps with one request. This serves, for example, for opening all grippers of one clamping at the same time, which are activated in separate steps of a step sequence.

Monitoring in manual mode

The monitoring is activated by selecting steps.

Synchronization

Activation of steps in dependence of states. Is effective in the operating mode transition from manual to automatic mode. This allows the user to synchronize the step sequence to the machine status. Invalid step combinations cannot be activated. A preselection can be made out of an active step or action.

1.5 Diagnosis

Faulty states of the sequence must automatically generates a diagnosis. This diagnosis shows the cause of an error on ladder diagram level. This constitutes the initial error analysis of the faulted step. Thus, only one step can be faulty, even though several steps are active at the same time.

Diagnosis of the maximum time monitoring

The criteria analysis distinguishes according to not-fulfilled transition and according to not-fulfilled action. If both are not fulfilled, the action generation is analyzed. If the action is executed, the transition is analyzed.

Immediate stop monitoring

Immediate stop monitoring means monitoring conditions which constitute an error. In case of error, immediate stop monitoring networks will be visualized by means of the criteria analysis. The immediate stop monitoring can be created in general for the step sequence, or step-specifically in a step.

Executability indication (movement indication)

The executablity indication (movement indication) is shown in the manual mode by the steps selected in the operating buttons. The executability indication shows whether the conditions for the execution of an action are available.

The actuation of a movement key, although there is no executability indication, generates a criteria analysis in which the missing conditions for the activation of the output are displayed.

2 Types of step sequences

2.1 Selection possibility

The programmer has to select the programming language when opening a new program POU or function block POE (POU = program organization unit, e.g. program, function block or functions):

- SFC - standard step sequence according to the standard EN 6-1131-3, without automatic operating mode and diagnosis generation.
- LD - ladder diagram, FBS - function block language and IL - instruction list.

The support of IndraStep step sequences and ProVi is activated via the programming interface with the selection window Project\SFC template.

If no entries can be found in Project\SFC templates, it is possible to load the current version from Project\Filing\Fetch. For this purpose select HMI form the menu item "Fetch".

Standard step sequence (SFC according to EN 6-1131-3)

When opening a new program or function block, the programmer chooses a programming language. Here, the basic selection is made whether the SFC sequence structure is used for programming or not, or which functionalities to be included in the step sequence. Step sequences can be programmed on program level and on function block level. A POU may consist of one step sequence.

ProVi

PVPGxAxx PVPGxAxx is a template for the use of the ProVi message system. ProVi is a step sequence which serves for program structuring on program level. In this step sequence the operating mode functionality has not been implemented. During the compilation, data for visualization in WIN-HMI are generated.

Program structuring by means of SFC elements has proven advantageous for its improved legibility, however, in this case it does not serve as sequence control. In the ProVi program structuring step sequence, IndraStep can be used in form of function blocks.

IndraStep

ISFBxAxx ISFBxAxx is a template for the application of IndraStep. IndraStep is a step sequence on function block level with operating mode functionality and diagnosis data generation.

The functionality of the IndraStep step sequence is achieved by the interplay of step sequence system data, control function blocks and control actions in prescribed programming technology invisible for the user, which can be attached to our standard step sequence as of PCL-interface version 18.

This functionality is almost invisible for the user. However, the programming instruction for IndraStep step sequences must be observed.

2.2 Selecting step sequences

When opening a new program or function block the programmer can select between:

- Standard step sequence, ladder diagram, function block language and IL,

- ProVi - data generation for WIN-HMI as program,
- IndraStep - step sequence as function block.

Selection window SFC templates

In the menu 'Project' of the programming interface there is the column SFC templates. Here, the Indramat step sequence type, or visualization, to be opened is selected.

Program template PVPGxAxx ProVi - program, step sequence for program structuring without operating mode functionality with data generation for WIN-HMI.

FB-template ISFBxAxx IndraStep, step sequence (as function block) with operating mode and diagnosis data generation.

Note: This selection is, among other reasons, necessary since IndraStep requires more SPS ressources. than standard step sequences. The loaded templates already contain the necessary imports and allocations required for ProVi or IndraStep.

2.3 General information about the step sequences

Maximum step number up to 254 steps per step sequence.

Max. number of parallel steps identical to the maximum number of steps per step sequence, up to 254 steps.

Maximum number of actions in theory, up to 64,000 actions per step sequence.

Maximum number of transitions identical to the maximum number of steps per step sequence, up to 254 transitions.

In practice, the maximum number of steps, actions and transitions is limited by memory extension and program structure.

Program memory required

If an SPS card is provided, e.g. with 1 Mbytes memory, about 740 kbytes are available for program memory.

A step sequence without steps, actions and transitions (with IndraStep) requires about 6.4 kbytes of this memory space.

The step itself, without the actions used, requires about 12 bytes. An action or transition without the internal networks requires about 10 bytes. Since actions or transitions might be used several times, about 10 bytes are required for multiple use. The total program memory space required for an action depends strongly on the networks of these actions.

A step with, e.g., one action, one transition generation and one time monitoring action each, requires about 650 bytes program memory.

Data memory required

An SFC structure requires resources from the non-Boolean data memory, of which 32 kbytes remanent and non-remanent memory each are available.

The step itself, without the actions used, requires about 12 bytes. An action or transition requires about 10 bytes. Each multiple use of an action or a transition requires about 10 bytes. The total data memory required for an action depends strongly from the variables used in these actions.

A step with, e.g., one action, one transition generation and one time monitoring action each, requires about 40 bytes data memory and a time step (timer).

3 Interface of the IndraStep step sequence

Designation of function and data types

The designation of the used FB (e.g. ISFB3A01) and data dypes (e.g. ISTY3A01) are composed as follows:

ISFB = **I**ndra**S**tep **F**unction**B**lock

ISTY = **I**ndra**S**tep Data**TY**pes (data structures)

x = version number

A = distinctive feature (e.g. ISFBxAxx, ISFBxBxx etc.)

xx = release number

The version number and the release number must be identical within a step sequence. In an SPS program step sequences with different version and release numbers might occur.

IndraStep step sequences are programmed on the function block level. An IndraStep function block has an input iSFC (structure of type ISTYxAxx) for operating mode, Start, Stop etc. selection for step sequence control. The status messages, such as stopped, error etc. are issued via the output qSFC (structure of the type ISTYxBxx).

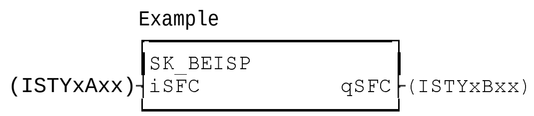


Fig. 3-2: Example of a step sequence function block

3.1 Control interface of the IndraStep function block

An IndraStep step sequence or the function block, which contains the step sequence, is controlled via the FB input iSFC (structure of type ISTYxAxx).

```
ISTYxAxx  STRUCT
            Start         BOOL   (*SFC start*)
            Stop          BOOL   (*SFC stop*)
            AutoStep      BOOL   (*SFC single block mode (automatic)*)
            Manual        BOOL   (*SFC manual mode*)
            SetStep       BOOL   (*forced steps (manual) for restart (auto)*)
            Reset         BOOL   (*reset SFC to initial step*)
            ClrError      BOOL   (*reset SFC error*)
            indOPNR       INT    (*HMI OP-area number OP2 - OP9*)
            indSCRNR      INT    (*HMI active screen-number*)
          END_STRUCT
```

Fig. 3-3: Structure of IndraStep - control signals iSFC

The control signals are evaluated statistically. The switch-on state of an IndraStep step sequence is the started automatic mode.

iSFC.Start

Restart of the step sequence.

After an error or stop iSFC.Start activates the deactivated command output of the action in the currently active steps. The step timers start again, and, as a consequence, the step time monitoring too.

If the step sequence has to wait in a defined step for the start input iSFC.Start, it has to be linked in the transition generation.

ISFC.Stop Stops the step sequence.

iSFC.Stop deactivates the command outputof the actions activated in the currently active steps. The behavior of the outputs managed in the step sequence are described in chapter 4 Control of the command output

In the automatic mode the stopped state can only be deleted with 'Restart'. In the manual mode the stopped state can also be reset with Delete Error.

iSFC.AutoStep switches the step sequence into the automatic single step mode.This is a part of the automatic mode. An active step is only being executed as long as the related transition is not fulfilled. By fulfilling the transition, the transition will be saved and the command output (see “Control of the command output”) will be disabled as of the next SPS cycle. Even if the transition should be 'FALSE' afterwards, the command output in the corresponding step remains disabled. With a positive edge at iSFC.Start or _SFC1.Start, the program is stepped on to the next step via the before fulfilled and saved transition(s) and the command output for this (these) step(s) will be enabled again.

The step sequence has to be started with iSFC.Start or _SFC1.Start when it is switched back from automatic single step mode to automatic (change of operating modes).

iSFC.Manual deactivates active steps, saves the step constellation of the step sequence and switches to the manual mode. In this operating mode it is possible to force steps and actions. If the switching back is realized manually (by changing from iSFC.Manual to FALSE), the step sequence is in stopped automatic mode.

iSFC.SetStep attempts to take over the step combination set in the manual mode to the automatic mode. The step constellation saved before when switching the operating mode manually will be rejected. It is not possible to quit the manual mode in case of a step constellation which cannot be executed in automatic mode. The stauts signal 'gSFC.Manual' remains set, the status signals 'fSFC.Error' and 'gSFC.Manual' will be set.

iSFC.Reset In the automatic mode iSFC.Reset sets the step sequence to the initial step. In manual mode the forced steps will be deactivated. The step constellations saved during the change from automatic to manual mode is not deleted by Reset.

iSFC.ClrError deletes an active error of the step sequence, indicated by the status signal 'gSFC.Error'.

iSFC.indOPNR corresponds to the selected OP-level (HMI-buttons)

iSFC.indSCRNR corresponds to the selected button (HMI-buttons)

3.2 Status interface of the IndraStep function block

The status signals of the step sequence for program application are issued via the FB ouput gSFC (structure of type ISTY2B00).

```
ISTYxBxx   STRUCT
            ErrorStep    STRING[9]  (*faulted step*)
            Error        BOOL       (*Machining error is present*)
            ErrSetStp    BOOL       (*Error SetStep take over in Auto*)
            Stop         BOOL       (*SFC stopped*)
            Manual       BOOL       (*Manual mode is active*)
           END_STRUCT
```

Fig. 3-4: Structure of IndraStep - status signals gSFC

qSFC.ErrorStep displays the error text with the name of the faulted step. The status signal qSFC.ErrorStep refers to the first error occurred if several steps are processed simultaneously (initial error display).

qSFC.Error displays an error state. The step sequence can be switched to an error state by the following actions:

- via the event monitoring by setting '_Action.MonError' or '_SFC1.MonError' (see 5.4 Immediate stop monitoring (event monitoring)).
- or in automatic mode by the end of the maximum step time monitoring (see 5.2 Step sequence structure
- or by a step constellation that cannot be applied in the automatic mode.

qSFC.ErrSetStp indicates that a preselected step constellation cannot be realized in the automatic mode.

qSFC.Stop indicates that the step sequence is in stopped state. 'Stop' can only be deleted with 'Start'.

The error state (qSFC.Error = TRUE) deactivates the command output of active actions and sets the step sequence to the stopped state (qSFC.Stop = TRUE).

An error message can only be acknowledged and deleted with 'ClrError'.

In the automatic mode the stopped state can only be deleted by 'Restart'. In the manual mode the stopped state is reset by Delete Error.

qSFC.Manual indicates that the step sequence is in manual mode.

Note: A step sequence cannot be switched to 'Automatic', if a step constellation that cannot be run was preselected with the input SetStep or in the step sequence with set _SFC1.SetStep, or _SFC1.GoStep (see iSFC.SetStep, _SFC1.SetStep and _SFC1.GoStep).

3.3 Data and contorl structure of the IndraStep function block

The IndraStep Data and control structure _SFC1 (type ISTXxCxx) serves for controlling the IndraStep step sequence. This structure includes the following areas:

- iCOMMAND - internal step sequence control elements
- iSTATUS - internal step sequence status elements
- COMMAND - extended step sequence control elementes
- STATUS - extended step sequence status elementes
- INTERN - internal markers for step sequence control
- TMP - temporary step data

The programmer can use areas **COMMAND** and **STATUS** in the step sequence for the control of the step sequence.

Elements of the IndraStep control structure _SFC1 (ISFBxCxx)

_SFC1.SET_HAND	iCOMMAND:	Manual mode	BOOL
_SFC1.MODE_AUTO	iSTATUS :	Automatic mode is activ	BOOL
_SFC1.RESET	iCOMMAND:	SFC reset	BOOL
_SFC1.RESTORE	iCOMMAND:	SFC restore steps	BOOL
_SFC1.ERRORFLG	iSTATUS :	SFC error flag	BOOL
_SFC1.ERRORNR	iSTATUS :	SFC error number	USINT
_SFC1.ERRORTYP	iSTATUS :	SFC error type	INT

Elements to be used for step sequence programming:

_SFC1.Auto	STATUS :	SFC mode auto	BOOL
_SFC1.Manual	STATUS :	SFC mode manual	BOOL
_SFC1.SumError	STATUS :	SFC summary error flag	BOOL
_SFC1.TmrError	STATUS :	Step time monitor error	BOOL
_SFC1.ManDiagRq	STATUS:	Manual diagnosis request	BOOL
_SFC1.AutoStep	STATUS:	SFC in single step (auto)	BOOL
_SFC1.MonError	COMMAND:	Monitoring error	BOOL
_SFC1.Start	COMMAND :	SFC start	BOOL
_SFC1.Stop	COMMAND:	SFC stop	BOOL
_SFC1.SetStep	COMMAND:	forced steps (man.) for restart	BOOL
_SFC1.GoStep	COMMAND:	go to forced steps	BOOL

- **STATUS**-flag inform about the operating state and may only be requested.
- **COMMAND** flags can be used for step sequence control.

_SFC1.SfcCycl	INTERN :	Internal cycle flag	BOOL
_SFC1.ErrorStop	INTERN :	Stop by error	BOOL
_SFC1.StopReqw	INTERN :	Stop by _iSFC.Stop	BOOL
_SFC1.EdError	INTERN :	Edge diagnostic error	BOOL
_SFC1.EdStart	INTERN :	Edge start SFC	BOOL
_SFC1.EdManuChg	INTERN :	operating mode change	BOOL
_SFC1.EdModeChg	INTERN :	operating mode change	BOOL
_SFC1.ChangeStp	INTERN :	step change in jog mode	BOOL
_SFC1.StartReq	INTERN :	Start request	BOOL
_SFC1.SetStpReq	INTERN :	SetStep request	BOOL
_SFC1.X	TMP :	AUTO:Stp. act. / MANU:Stp. force	BOOL
_SFC1.T	TMP :	STATUS : step active time	TIME
_SFC1.S	TMP :	STATUS : Transition to stp. ndd.	BOOL
_SFC1.T_enable	TMP :	STATUS : Transition enable	BOOL
_SFC1.T_min	TMP :	Min. step time is running	BOOL
_SFC1.T_max	TMP :	Max. step time has run out	BOOL

Note: The elements iCOMMAND, iSTATUS, INTERN and TMP may only be influenced by the step sequence operating mode control.

3.4 Physical external interface of the IndraStep function block

The inputs and outputs related to the application can be allocated as usual in the area VAR_INPUT and VAR_OUTPUT. However, it is possible to allocate them as global agreement VAR_EXTERNAL.

```
                 Example
               ┌────────────────────┐
               │SK_BEISP            │
   (ISFC3A01)──┤iSFC            qSFC├──(ISFC3B01)
       (BOOL)──┤Input_1     Output_1├──(BOOL)
       (BOOL)──┤Input_2     Output_2├──(BOOL)
       (BOOL)──┤Input_3     Output_3├──(BOOL)
               └────────────────────┘
```

Fig. 3-5: Example of a step sequence function block

Imports on program level

System-immanent type imports on program level for the IndraStep step sequence are already entered when opening a new programs using ProVi, so the programmer only has to enter the IndraStep FB’s.

```
IMPORT        Comment
TYPE
(***************************************************************************)
(*** Imports which are necessary for the Indramat step sequence ***)
 ISTYxAxx      (*Structure control inputs of the step sequence*)
 ISTYxBxx      (*Structure status outputs of the step sequence*)
 ISTYxGxx     (*Structure global action control data     *)
(*********** End of import of Indramat step-specific data **********)
(***************************************************************************)
FUNCTION

FUNCTION_BLOCK
 SK_BEISP      (*Example of an IndraStep application*)
```

Fig. 3-6: Example of import section on program level for the application of IndraStep.

Allocations on program level

For step sequence control, a separate name has to be allocated to the structure ISTYxAxx for the application in the program.

For the application of the step sequence state in the program, a separate name has to be allocated to the structure ISTYxBxx.

An internal communication between the operating mode control is realized with the globally agreed structure ISTYxGxx. This has to be name ‘_Action’ in the area VAR and VAR_GLOBAL. This entry is available in case of new programs with ProVi and applies to one or several IndraStep step sequences.

Inputs and outputs of the periphery or flags on program level can be transmitted to the step sequence function block either traditionally or as global variable.

```
Identifier   AT            TYPE        :=          Comment
PROGRAM      STATTEST
VAR_INPUT
 I_WkstP1    %I112.3.0     BOOL                    (*Input workpiece at transfer position*)
 .
 .
 I_GBoh1u    %I113.3.4     BOOL                    (*Input screw tap 1 bottom*)
END_VAR

VAR_OUTPUT
 Q_Band1     %Q112.1.0     BOOL                    (*Output workpiece position1*)
 .
 .
 Q_PART_OK   %Q113.1.3     BOOL                    (*Output part machining completed*)
END_VAR

VAR
 _Action                   ISTYxGxx                (*Global temporary action control*)
 iSFC1                     ISTYxAxx                (*Control inputs IndraStep*)
 qSFC1                     ISTYxBxx                (*Status outputs IndraStep*)
 Beispiel                  SK_BEISP                (*Example of an IndraStep step sequence*)

END_VAR

VAR RETAIN

END_VAR
VAR_GLOBAL
 _Action                   ISTYxGxx                (*Global temporary action control*)
 (* Globalized inputs from the periphery *)
 I_WkstP1    %I112.3.0     BOOL                    (*Input workpiece at transfer position*)
 .
 I_GBoh1u    %I113.3.4     BOOL                    (*Input screw tap 1 bottom*)
 (* Globalized inputs from the periphery *)
 Q_Band1     %Q112.1.0     BOOL                    (*Output workpiece position1*)
 .
 Q_PART_OK   %Q113.1.3     BOOL                    (*Output part machining completed*)
END_VAR
```

Fig. 3-7: Example of allocation section of the use of IndraStep in the program

4 Control of the command output

The command output is the assignment of a logic result to Boolean flags, outputs and jumps.

4.1 Control of the command output of a standard step sequence

A standard step sequence does not have a higher-level control of the command output. If a step is active, the programmed actions are activated by this. Networks in active actions are processed. The system support of a standard step sequence is limited to the elements for which the user has information on whether a step or an action is processed actively, or is in post-processing. Operating modes and error management with error evaluation are not included in this support.

A data structure with the Boolean elements <AktName.Q> and <AktName.A> is created in the system with each new non-Boolean action. These can be used in the implementation like normal Boolean structure elements.

The element Q of this structure (AktName.Q) is ‘TRUE’ as long as the action is active and is not reprocessed. The element A (AktName.A) is 'TRUE' during the entire action processing.

By linking before command output the post-processing of an action can be defined with the element <AktName.Q>.

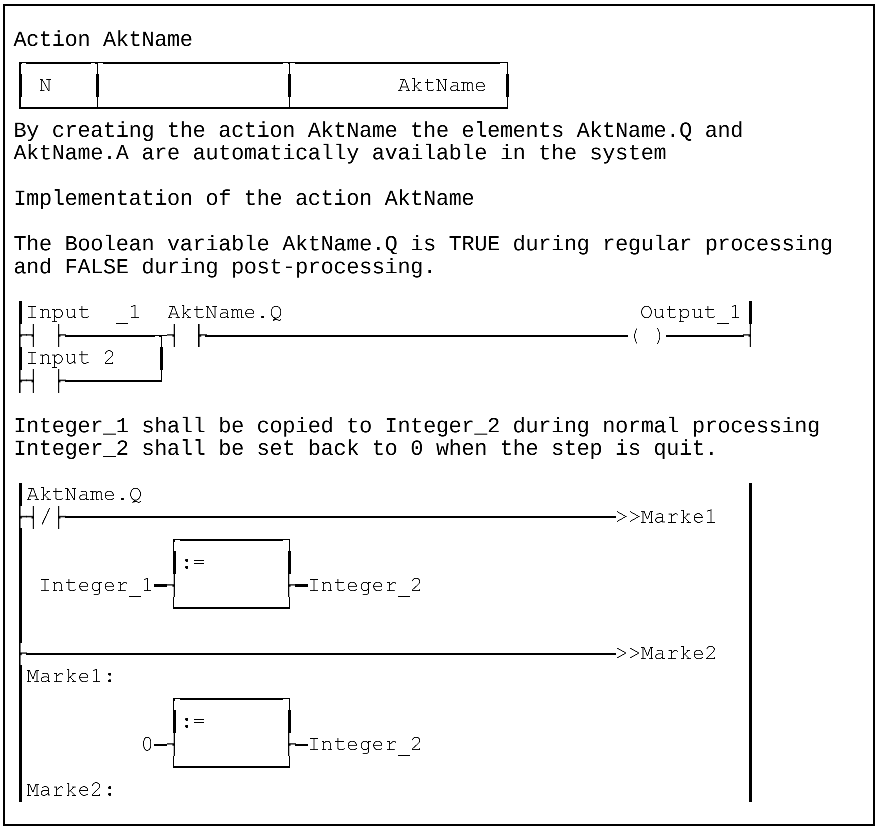


Fig. 4-8: Control of the action post-processing of a standard step sequence

Rexroth Indramat

4.2 Control of the command output of an IndraStep step sequence

The standard action elements are not sufficient for an IndraStep step sequence. The different conditions for disabling the command output of an activated action are taken into account by the function block action (type ISFBxBxx). This FB is integrated automatically during step sequence compilation prior to the command output. Thus, the application of the operating mode control is simplified.

Function block Action (FB ISFBxBxx)

When opening a new IndraStep step sequence this FB has already been imported automatically (type ISFBxBxx) and is allocated with the name ‘Action’. This function block must not be deleted.

The control of the command output is realized during the automatic insertion in the user interfaces during compilation. The system automatically inserts the function block only in the Boolean operations. In case of subsequently modified operations, the compiler correctsthe network position or the wiring.

Note: In case of Online-editing no automatic insertion prior to the command output takes place. In this case the user has to program the FB 'Action' himself prior to the command output and connect it with the action name.

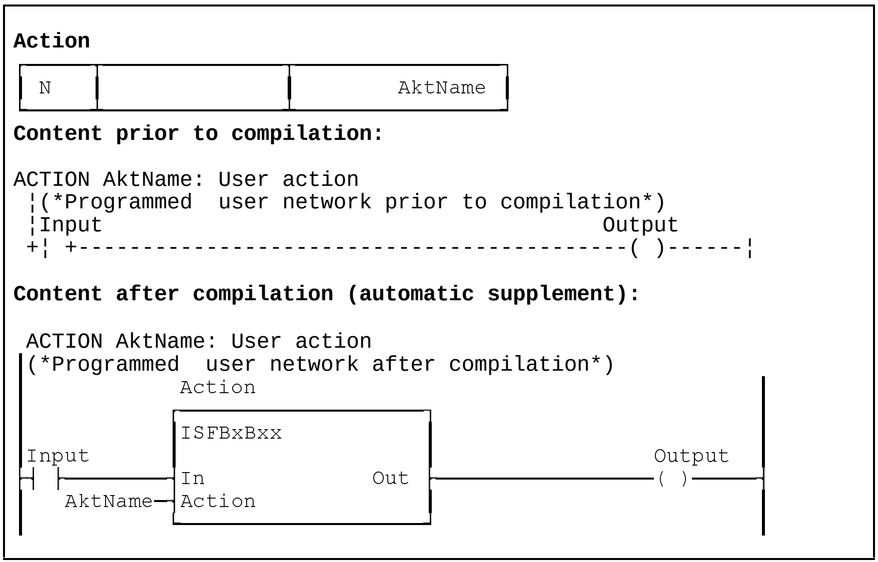


Fig. 4-9: Example of the network with Boolean command output

Functionality of the FB Action (type ISFBxBxx)

FB Action interrupts the operation:

- when the step sequence is in stopped state
- in case of error
- in the automatic single step mode with fulfilled transition but not yet switched on step
- during action post-processing
- when the action is in forced state. Regarding the processing principle, forced actions are in the post-processing phase since there is no step relation.

Note: FB Action use the global variable '_Action'. On program level this has to be imported as ISTYxGxx and allocated _Action type ISTYxGxx in the area VAR and VAR_GLOBAL.

Non-Boolean operations

The FB 'Action' (function block type ISFBxBxx) is not inserted automatically.

The programmer has to define the reaction of non-Boolean operations . If the process shall be realized analogously to Boolean networks, this can be controlled e.g. via jumps over those networks.

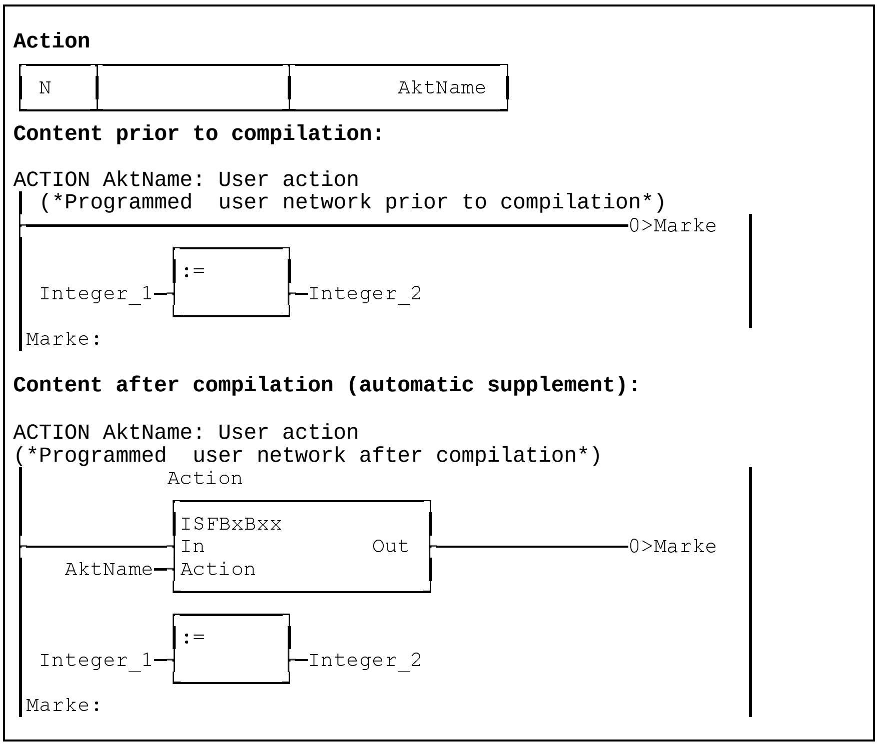


Fig. 4-10: Example of a network with non-Boolean command output

Controlling of automatic inserting of the FB Action (ISFBxBxx)

Generally, the FB Action is inserted in every network with Boolean result. However, the programmer can control the automatic insertion and the diagnosis generation.

Action names which do not lead to the insertion of FB Action

Action name Always

generally, no intervention in the action 'Always' takes place.

Note: The action 'Always' must always be integrated in the initialization step of the step sequence with the action qualifier R and must not be renamed or deleted.

Action names aManual and aMonError

If the action names aManual and aMonError are assigned to actions, the FB's Action will not be inserted in the actions for the control of the command output.

Action name aMonError

The monitoring (limit switch pair monitoring, etc.) valid for the entire step sequence shall be programmed in the action aMonError. This action is analyzed and monitored independently.

Variable names do not lead to the insertion of FB Action

Variables _SFC1, _TMP and _Action

If a network result is formed on the basis of a Boolean element of the variable or the structure element with the allocation name _SFC1 (ISTYxCxx), _Action (ISTYxGxx) or _TMP, no intervention of FB Action takes place.

The variable name _TMP serves for declaring a structure with any elements which do not lead to the network result FB Action insertion.

Note: Only when the variable elements _SFC1, _TMP and _Action are used exclusively as network result, no intervention of FB Action takes place.

Labels between which FB Action will not be inserted

Label CtrlOf

After this label FB Action will not be inserted in a network until end of action or until the label CtrlOn is set.

Label CtrlOn

as of this label the insertion will be continued if the intervention of FB Action has been switched off with the label CtrlOf.

Status signals for internal step sequence evaluation

_SFC1.Auto

Automatic mode

Step sequence is in the automatic operating mode. This flag is "TRUE" when the step sequence is in automatic mode.

_SFC1.Manual

Manual mode

Step sequence is in the manual operating mode. This flag is "TRUE" when the step sequence is in manual mode.

_SFC1.SumError

Summary error

This flag is "TRUE" when a maximum time is exceeded or an immediate stop monitoring error is active.

_SFC1.TmrError

Step time monitoring

This flag is "TRUE" when a maximum time is exceeded (<Step name>.T_max has expired).

_SFC1.AutoStep

Automatic-single-step

This flag is "TRUE" when the step sequence is in automatic single step mode.

Note: These STATUS flags may only be requested. They allow the programmer to realize links within the step sequence which require information on these states. Writing on STATUS flags can lead to uncontrolled behavior of the step sequence.

Command signals for internal step sequence control

_Action.MonError, _SFC1.MonError

Generates an error state.

By setting _Action.MonError, or _SFC1.MonError the step sequence can be switched to an error state by an event within the step sequence (see 5.4 Immediate stop monitoring (event monitoring).

_SFC1.Start

Restart of the stopped step sequence.

By setting _SFC1.Start a start or restart of the stopped step sequence can be released in automatic or automatic single step mode by an event in the step sequence. The step times will be reset and the monitoring times are started again. The marker is evaluated in parallel to iSFC.Start (see 3.1 Control interface of the IndraStep function block).

_SFC1.Stop

Stopping the step sequence.

By setting _SFC1.Stop the step sequence can be stopped by an event in the step sequence. The marker is evaluated in parallel to iSFC.Stop (see 3.1 Control interface of the IndraStep function block).

_SFC1.SetStep

Application of a step constellation to the automatic mode.

By setting _SFC1.SetStep it is possible to define by an event in the step sequence that a step constellation set in manual mode is taken over to the automatic mode when switching over. The marker is evaluated in parallel to iSFC.SetStep (see 3.1 Control interface of the IndraStep function block).

_SFC1.GoStep

Unconditional continuation of step sequence processing in any step (in automatic mode).

By setting _SFC1.GoStep steps can be selected in the step sequence (<Step name>.X) at which the step sequence processing shall be continued. The <Step name>.X flag of the steps to be selected must be set over 3 SPS cycles so the system can take over the steps. When the selection is completed the step sequence control resets _SFC1.GoStep.

If an impossible step constellation is selected for further processing, the step sequence switches to the error state (the outputs qSFC.Error and qSFC.ErrSetStp are set). Upon acknowledgement of this error and the subsequent restart the step sequence processing is continued with the step constellation active at the time of setting _SFC1.GoStep.

```
Action
| N |                |        AktName |

Contents of the action:
ACTION AktName: User action
  (*Within one step, if the flag Event is set TRUE *)
  (*the step Step shall be selected and the step sequence *)
  (*processing shall be continued there.      *)
 CtrlOf:
 Event                                          Step.X
-| |-----+-------------------------------------+-(S)---
         |                                     |
 _SFC1.GoStep                                  |_SFC1.GoStep
-| |-----+                                     +-( )---
```

Fig. 4-11: Example of an application of _SFC1.GoStep

 Rexroth Indramat

5 Programming instruction for IndraStep step sequence

5.1 Diagnosis texts and criteria analysis

The diagnosis texts are taken from the program and allocation comments. During the compilation of the IndraStep step sequence function blocks, the operating mode control is completed and the diagnosis data and texts are extracted. Programmers who use the commenting will not have any extra work for error evaluation creation.

The language-dependent management of extracted diagnosis data and texts is made available in WIN-HMI.

5.2 Step sequence structure

During processing in correct automatic mode IndraStep step sequences are, in principle, like standard step sequences. Thus, all functionality features of the normal SFC editor can be used for programming, such as alternative branches, parallel branches and jumps.

Step

The step programming corresponds to the standard programming and is not subject to specific restrictions.

```
sMoveOut   Example of a step

Data for the step (Type ISTYxDxx) which are available
automatically in the system when programming the step:

 sMoveOut.X         BOOL AUTO : Step active  / MANU : Force step
 sMoveOut.T         TIME STATUS  : Step active time
 sMoveOut.X_        BOOL STATUS  : Step active temporarily
 sMoveOut.S         BOOL STATUS  : Transition to step completed
 sMoveOut.ActionIn  BOOL STATUS  : Action In
 sMoveOut.ManEnable BOOL COMMAND : Enable Action in Manual-Mode
 sMoveOut.T_min     BOOL Minimum step time is running
 sMoveOut.T_max     BOOL Maximum step time has run out
```

Fig. 5-12: Example of a step

The step name and the step comment is used for visualization by WIN-HMI.

The steps of the IndraStep step sequence are ISTYxDxx type steps. If a new step is programmed, the following eight variables are added automatically to this step:

<Step>.X - Step active

Step active, forcing steps

If this variable is TRUE, the step is processed. In the automatic mode of the step sequence the variable is activated only by the step sequence control. Forcing by the user program does not influence the step sequence control. In manual mode this variable can be forced (even several steps at a time!), leading to the processing of the step(s) and the executability indication.

<Step>.T - current step time

Current step time

This variable is a TIME type variable. As long as the step is active, this timer counts. If the step is deactivated, the timer stops with the last value.

Only by activating the step again this timer will be reset by the step sequence control.

<Step>.X_ - Step active during processing time

This variable is only TRUE within the active step processing and thus in the actions activated in this step.

Note: <Step>.X_ is an operating mode variable and must not be influenced by the SPS program.

<Step>.S - transition for step fulfilled

This variable is used by the operating mode control for saving the positive transition result.

Note: <Step>.S is an operating mode variable and must not be influenced by the SPS program.

<Step>.ActionIn - executability indication

The executability indication and the function block Action (ISFBxBxx) work together. An activated step "collects" information on the used and active actions as well as the control blocks integrated. If at least one network result is fulfilled at the input "In" of the control block "Action", this variable is set to TRUE, or to FALSE respectivly, when all results are FALSE.

<Step>.ManEnable - movement enable in manual mode

The action(s) controlled via the function block in the manual mode can be enabled with this variable.

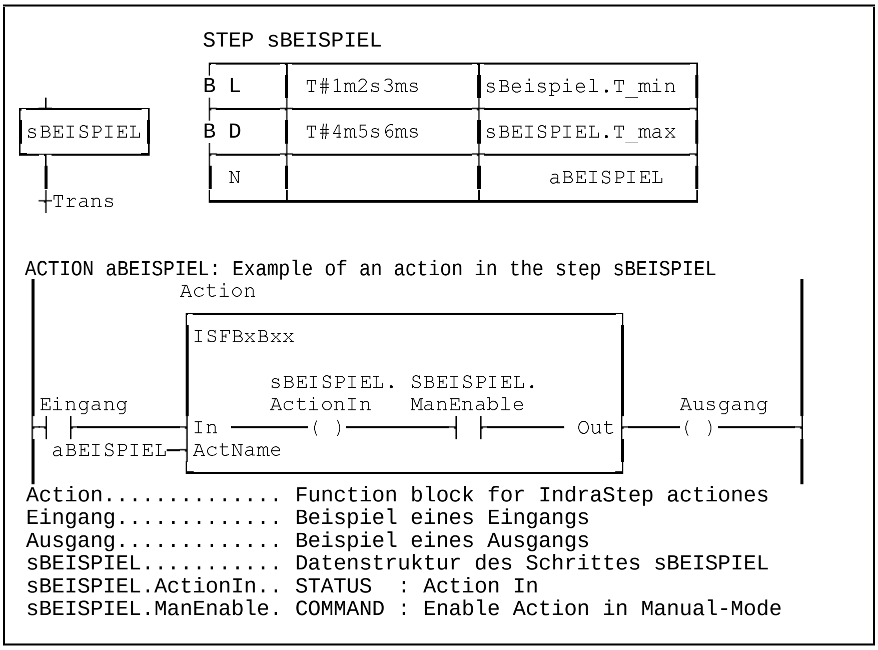


Fig. 5-13: Example of the effect of ActionIn and ManEnable

<Step>.T_min - minimal step time

The programmer enters this variable in the step to generate a minimum step time as Boolean time-limited (L) action.

By activating the step this time action is started. During this time (<Step>.T_min = TRUE) the step is not switched on.

```
B L | T#1m2s3ms | Schritt.T_min | (*Minimum step time*)
```

Fig. 5-14: Example of an action minimum step time

<Step>.T_max - maximum step time

The programmer enters this variable in the step to generate a maximum step time as Boolean time-delayed (D) action.

By activating the step this time action is started. If this time is over and the step is still active (<Step>.T_max = TRUE), the error state "Maximum step time" is released.

```
B D | T#1m2s3ms | Schritt.T_max | (*Maximum step time*)
```

Fig. 5-15: Example of an action maximum step time

Variable step times

Since the time monitoring are controlled by means of the standard actions L and D, the step times can also be programmed with variables (see Standard programming instructions).

```
B D | TxVAR | Schritt.T_max | (*Maximum step time*)
```

Fig. 5-16: Example of the variable step time

Action

```
N |  | aBEISPIEL | (*Beispiel einer Aktion*)

Data for the action (Type ISTYxExx) which are available
automatically in the system when programming this action:

 aBEISPIEL.Q  BOOL  STATUS  : Action active
 aBEISPIEL.A  BOOL  STATUS  : Action being processed
 aBEISPIEL.X  BOOL  COMMAND : force Action
```

Fig. 5-17: Example of an action

The actions of the IndraStep step sequence are ISTYxExx type actions. If a new non-Boolean action is programmed, the following three variables are added automatically to this action:

<Action name>.Q - normal action processing

normal action processing

If this variable is TRUE, the action is processed normally and is not in the post-processing phase.

<Action name>.A - general action processing

Action post-processing

If this variable is TRUE, the action is processed. The difference to the variable <Action name>.Q is that this variable is also TRUE during the post-processing of the action.

<Action name>.X - forcing an action

Forcing an action

If this variable is set TRUE, the action is processed. During the time the variable is TRUE, the action is in the post-processing phase.

Note: Actions are forced independent of the automatic mode, i.e. also in the automatic mode!

```
STEP Band_1: Conveyor 1 move workpiece in and open door 1
B D | T#20s | Band_1.T_max | (**)
  N |       | aBAND1       | (*Workpiece to posit.1*)
  F                           forced action
  N |       | aTUERo1      | (*Open door 1*)
```

Fig. 5-18: Example of a forced action

Limitations when using actions

The programming of actions corresponds to the standard programming. Limitations are the applicable qualifiers and action types.

Admissible action qualifiers are N, P, P0, P1, L, D. The step time monitoring is programmed by actions with the action qualifier L (<Step>.T_min) and D (<Step>.T_max).

Note: Application actions with saving qualifiers or qualifiers effective for longer than the step time cannot be controlled by the operating mode control.

Multiple use of actions is admissible.

Boolean actions are generally not admissible since they cannot be controlled by the operating mode control. Actions for minimal and maximal step time are the exception to this.

Control of the command output is realized via a function block, allocated "Action" of the type ISFBxBxx which is inserted automatically during the step sequence compilation (see 4 Control of the command output).

Transition

For controllability and diagnosis purposes, the step transition sequence must be a program-technical unit. Therefore, the transition generation is realized within the step and prior to the transition. For this purpose a Boolean element of the transition (<Transition name>.X) is programmed in the step. Transition conditions programmed directly in the transition (like in the standard step sequence without operating modes) are overwritten by a general control logic during step sequence compilation.

```
    .
    .
 |________|
   |
  -+Pos_2     Machining position reached
   |
Data for the transition (Type ISTYxFxx) which are available
automatically in the system when programming this transition:

Pos_2.X    BOOL  COMMAND : Transition complete
Pos_2.X_   BOOL  STATUS  : Transition complete
```

Fig. 5-19: Example of a transition

The transitions of the IndraStep step sequence are ISTYxFxx type transitions. If a new transition is programmed, the following two variables are added automatically to this transition:

<Transition>.X

Generation of a transition result

If the transition is fulfilled, this flag is used for saving the transition result in the step. This means that within the steps active before the transition result has to be formed on this flag.

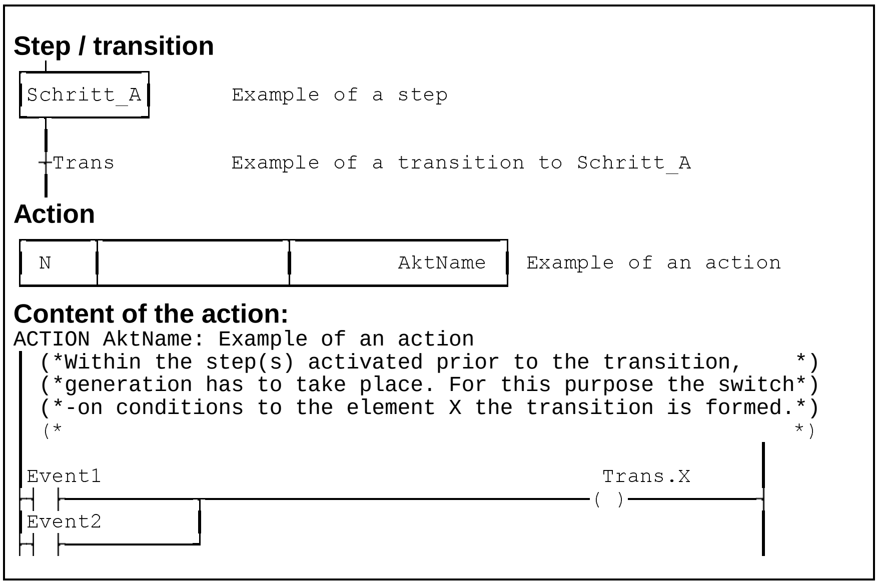


Fig. 5-20: Example of the generation of a transition result

<Transition>.X_

This is formed by the operating mode control. In dependence of the operating mode, the transition result is transmitted to the step sequence control with this flag.

Note: <Transition>.X is an operating mode variable and must not be influenced by the user SPS program.

5.3 Transition generation

In order to guarantee the controllability and the diagnostic capability of the step sequence the transition generation has to be executed within the step. Transition conditions programmed directly in the transition are overwritten by a general control logic during step sequence compilation.

Transition generation of the Indramat step sequence

When a new step is opened, the cursor positions in the new step. After having entered the step name, the cursor automatically positions at the transition and a transition name has to be entered.

Boolean variables as transition name

are not allowed for Indramat step sequences with operating modes. In contrast to the standard step sequences, existing Boolean variables must notbe entered here.

<Transition>.X

A variable <Transition>.X is automatically available for transition generation when entering the new transition name. This variable has to be programmed in an active step before the transition. The switch-on condition is met when this variable is set TRUE.

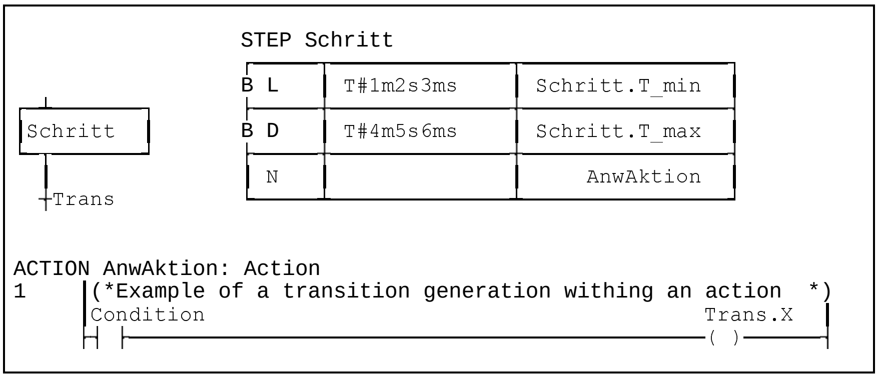


Fig. 5-21: Example of a transition programming

The transition generation can also be attached to a normal application network.

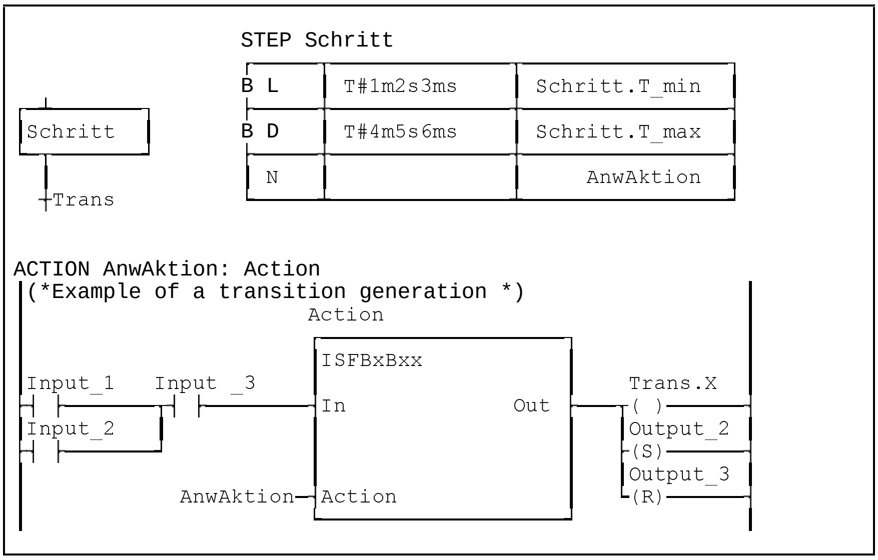


Fig. 5-22: Example of a transition programming to an application network

If other variables are programmed in parallel at the same time with the transition flag <Transition>.X, the compiler inserts the action control block automatically while compiling.

5.4 Immediate stop monitoring (event monitoring)

_Action.MonError / _SFC1.MonError

By setting the IndraStep system variable _Action.MonError the step sequence can be switched to an error state by an event within the step sequence. At the step input the variable '_Action.MonError' is always set to 'FALSE' by the operating mode control. This allows the user to diagnosis conditions over several networks, but not unintentionally beyond the step limits.

Note: The event monitoring can only be released by an action which is in an active step (<Step name>.X = TRUE). No error can be released during the post-processing phase.

If the VKE (= Verknüpfungsergebnis, operation result) of a network, in which _Action.MonError is programmed, is 'True', an immediate stop monitoring error message is generated at once. The immediate stop monitoring error message cannot be deleted in an active step, as long as the error state is related to the _Action.MonError.

Difference between _Action.MonError and _SFC1.MonError

In the criteria analysis the _Action.MonError is displayed as _SFC1.MonError. The flag _Action.MonError is part of the globalized data structure _Action and is thus used by all step sequences. Therefore, this flag is always reset by the operating mode control to the step input and will always indicated FALSE in the status display. The regrouping of the the flags _Action.MonError and _SFC1.MonErr at the step output is realized automatically.

As a result, the control block "Action" can disable the subsequent command outputs in the same SPS cycle in case of an monitor error, even before the operating mode control has registered the error request and disables the command output by the sum error _SFC1.SumError.

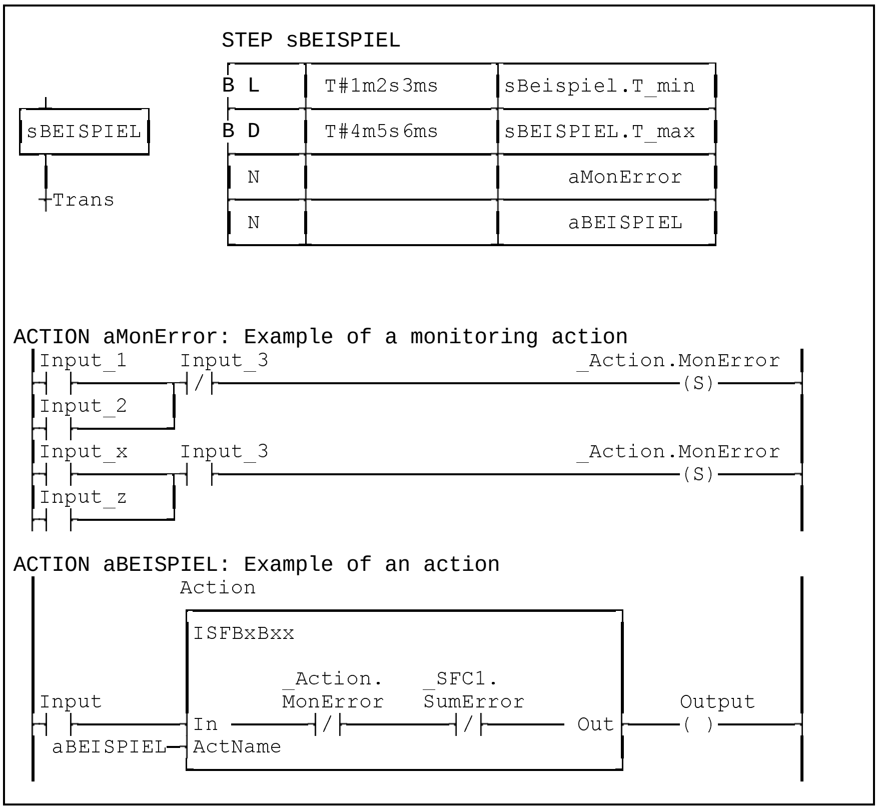


Fig. 5-23: Example of the effect of the event monitoring

5.5 Time monitoring

In automatic and in single step mode the step times can be monitored with respect to a minimum and a maximum time.

Minimum dwell time in the step

This time defines the minimum dwell time of the step, during which the step does not switch on despite fulfilled transition. The state "transition fulfilled" with minimum time not expired is a normal operating state, which is not considered an error.

<Step name>.T_min This variable is entered in the step to generate a minimum step time as Boolean time-limited (L) action.

B L	T#1m2s3ms	Schritt.T_min	(*Minimum step time*)

Fig. 5-24: Example of an action minimum step time

Maximum dwell time in the step

This time defines the maximum dwell time of the step, during which the transition to the step has to be fulfilled. If this time expires without the transition to the step being fulfilled, the step sequence is switched to the error state.

<Step name>.T_max

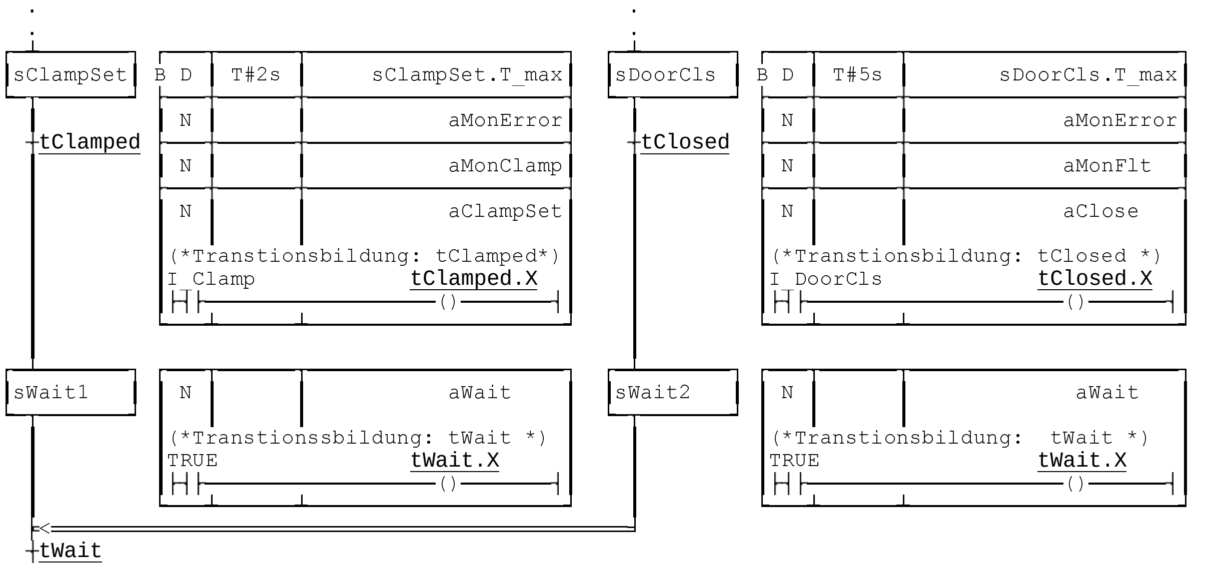
This variable is entered in the step to generate a maximum step time as Boolean time-delayed (D) action.

B D	T#1m2s3ms	Schritt.T_max	(*Maximum step time*)

Fig. 5-25: Example of an action maximum step time

Transition generation for AND-branch junction

The time monitoring of a step can only produce the intended effect when a step with the programmed time monitoring is followed a transition generation. The junction of AND-branches is realized before a transition. As a result, this transition is responsible for all steps of the AND branch before the junction. This transition has to be fulfilled and the steps have to be active simultaneously before the junction, so that the processing of the step sequence is continued at that point (Standard of the EN 6-1131-3). Therefore, the junctions have to synchronized via waiting steps, since otherwise the monitoring of the steps does not produce the intended effect. Moreover, wrong diagnoses might be caused by this.



The two steps sClampSet and sDoorCls must both have a transition of their own so the maximum time monitoring can be effective separately for each step. Therefore, the steps sWait1 and sWait2 follow. As soon as both transitions of the steps sClampSet und sDoorClsare fulfilled and thus the steps sWait1 and sWait2 are active, the step sequence continues functioning in a branch junction, since the transition tWait and TRUE are bond firmly

Fig. 5-26: Example of a parallel branch junction with time monitoring

6 IndraStep operating modes

An IndraStep function block has an input iSFC (structure of type ISTYxAxx) for operating mode selection, Start, Stop etc. for step sequence control. The status messages, such as stopped, error etc. are issued via the output qSFC (structure of the type ISTYxBxx).

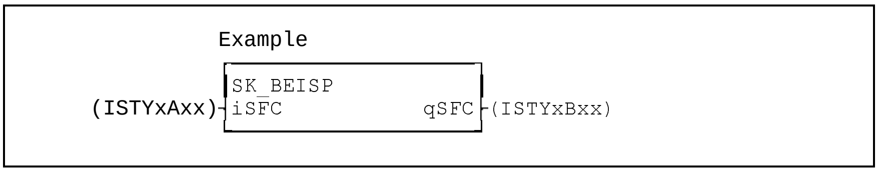


Fig. 6-27: Example of a step sequence function block

6.1 Automatic

iSFC.Manual = FALSE

The step sequence is started by a positive edge at a control input and stopped by a positive edge on a control input <iSFC.Start>. An active stop does not allow a start.

The switch on-state is, on the basis of the step sequences without operating modes, the started automatic mode. Steps will be processed actively until the subsequent transition condition is fulfilled. The fulfilled transition switches over to the next step by deactivating the actually active step and activating the step following the transition (via AND-branch various steps).

Immediate stop monitoring

All monitorings programmed by the user are active.

Time monitoring

Minimal and maximal times are evaluated.

6.2 Automatic single step

iSFC.AutoStep = TRUE and iSFC.Manual = FALSE

The step sequence functions like in the normal automatic mode. A change between steps only takes place when a start pulse is transferred to the step sequence in addition to the fulfilled transition condition.

Transition result saved

Since the contacts in the transition generation are not only achieved states but also time-limiteded signals, the fulfillment of the transition is saved until the next start pulse.

Immediate stop monitoring

The monitoring functions are active like in the automatic mode.

Time monitoring

A programmed minimum time is effective like in the normal automatic mode.

Maximum waiting time for fulfilled transition

While waiting for a start pulse with the fulfillment of the transition the maximum time monitoring of the step is completed .

6.3 Synchronization

iSFC.SetStep = TRUE if iSFC.Manual changes from TRUE to FALSE.

Is effective in the operating mode transition from manual to automatic mode. This allows the user to synchronize the step sequence to the machine status. Invalid step combinations cannot be activated. A preselection can be made out of an active step or action.

6.4 Manual

iSFC.Manual = TRUE and iSFC.SetStep = FALSE

In manual mode basically all steps and all movement executing actions are deactivated. The processing of movement executing steps can be started by an external request. The external request is generally transmitted from one or several operating devices which are designed as key assignments in masks .

Realize several movements simultaneously

This function allows simultaneous activation of several steps with one key. This serves, for example, for opening all grippers of one clamping at the same time, which are activated in separate steps of a step sequence.

Monitoring in manual mode

The basic step sequence monitoring still has to be activated. Step-specific monitoring is activated by selecting the corresponding steps.

Differences to the automatic mode

- In manual mode all transitions are permanently inactive. As a consequence they are not processed and cannot switch.
- All steps of the SPS cycle in which the SFC is processed first in the manual mode are deactivated.
- The action processing is still realized according to the rules of automatic mode. Active steps at the switch-over time are completed as if the related transition switched on, only the next step(s) are not activated. Set and save time-limited actions remain active, or can be activated.

Note: If saved actions shall be deactivated while switching over to the manual mode, these must be reset by means of a step activated in a defined way in manual mode.

Forcing steps

```
   |
+----------+
| sMoveOut |  Example of a step
+----------+
   |

sMoveOut.X          BOOL AUTO : Step active  / MANU : Force step
sMoveOut.T          TIME STATUS  : Step active time
sMoveOut.X_         BOOL STATUS  : Step active temporarily
sMoveOut.S          BOOL STATUS  : Transition to step completed
sMoveOut.ActionIn   BOOL STATUS  : Action In
sMoveOut.ManEnable  BOOL COMMAND : Enable Action in Manual-Mode
sMoveOut.T_min      BOOL Minimum step time is running
sMoveOut.T_max      BOOL Maximum step time has run out
```

Fig. 6-28: Data element of a step

Step activation

<Step>.X

Steps can only be forced in manual mode. By switching over to the manual mode the currently active steps are deactivated, i.e. all <Step>.X-flags are set to FALSE by the operating mode control in the first active manual mode-SPS cycle. As of the second manual mode-SPS cycle the step flags (<Step name>.X) can be set to TRUE and the steps processing can be started by direct forcing (Ctrl+F4), or via a button inWinHMI. The networks in the forced steps controlled by the operating mode control are enabled by the step flag <Step name>.ManEnable, which is evaluated in the function block "Action".

Executability indication (movement indication))

The executability indication (movement indication) is shown in the manual mode by the steps selected in the operating buttons. The executability indication shows whether the conditions for the execution of an action are available.

The actuation of a movement key, although there is no executability indication, generates a criteria analysis in which the missing conditions for the activation of the output are displayed.

<Step>.ActionIn

The functionality of the executability indication is provided in combination with the function block Action (ISFBxBxx). The function block represents the executablity indication on the step flag <Step>.ActionIn. If the network is fulfilled until the function block the step variable <Step>.ActionIn = TRUE. Since one step might contain several actions the executability indication is "collected". If at least one network result is fulfilled at the input "In" of the control block "Action" in one step, this variable is set TRUE. Only if all results are FALSE, this variable will be FALSE.

Movement enabling in the manual mode

<Step>.ManEnable

The actions of the active step activated in the manual mode via the function block "Action" can be enabled with the variable <Step>.ManEnable.

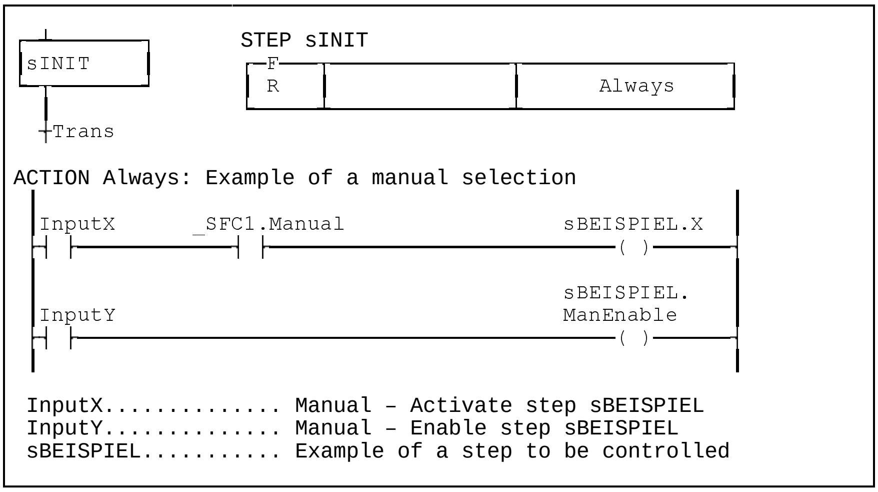


Fig. 6-29: Example of a step selection and activation

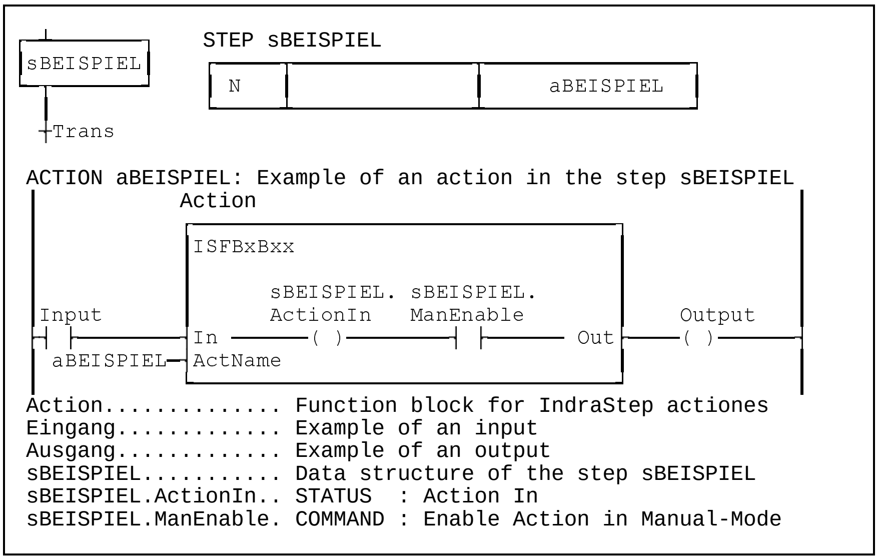


Fig. 6-30: Example of the effect of a step selected in manual mode

Forcing actions

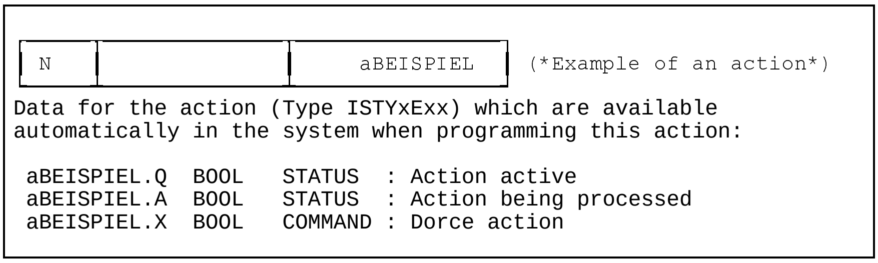


Fig. 6-31: Data elements of an action

<Aktion>.X

A data structure of the type ISTYxExx with the name of the action is created with each newly entered action. This structure contains the three Boolean elements Q, A and X.

There are two ways to force actions independently of the operating mode:

- in an active action by means of writing in <Action>.X within the step sequence.
- via the connected visualization by forcing the <Action>.X

For this purpose, the step sequence management (structure interpreter) realizes the following behavior.

- In the first step of a step sequence execution actions are forced (corresponds to post-processing).
- During this process the states of the action force flag <Action>.X of all actions are checked. With <Action>.X, <Action>.A is set as well. An ascending edge at <Action>.X leads to forcing and a descending edge leads to a cancellation of the forcing of an action.
- Then, the post-processing is realized for all actions the forcing of which has been cancelled and which are not active due to a normal execution. This means that in case of a descending edge of the flag <Action>.X, the output <Action>.Q = FALSE, the flag <Action>.A remains set for one further SPS cycle and thus the action will be reprocessed.

Then, the structures programmed by the user are processed.

The following table contains the possible assignments of the action system variables:

xxx.Q	xxx.A	xxx.X	
FALSE	FALSE	FALSE	no action processing
FALSE	TRUE	FALSE	action post-processing
FALSE	TRUE	TRUE	action processing (forced)
TRUE	TRUE	FALSE	action processing (activated)
TRUE	TRUE	TRUE	action processing (forced and activated)

For every action a post-processing identification can be realized by evaluating these flags.

Note: The evaluation of the action data structure is realized by the function block Action type ISFBxBxx for the IndraStep step sequence.

6.5 SFC operating mode error

_SFC1.ERRORFLG
_SFC1.ERRORNR
_SFC1.ERRORTYP

_SFC1.ERRORFLG, _SFC1.ERRORNR and _SFC1.ERRORTYP are system variables, which are available automatically with each step sequence. _SFC1.ERRORFLG indicates whether the last operating mode switch-over has been executed successfully. If _SFC1.ERRORFLG = ''TRUE', the feature of the SFC step status of the last operating mode switch-over which has led to the cancellation can be defined by means of the flags _SFC1.ERRORNR and _SFC1.ERRORTYP.

SFC operating mode switch-over error

_SFC1. ERRORTYP	Cause	Troubleshooting
-11101	No step active in SFC	Operating mode switch-over will be cancelled.
-11102	The SFC contains branches with too many flags (jam).	SFC is processed in manual mode.
-11103	The SFC contains branches with too many flags.	

Note: The operating mode switch-over does not have any direct effect on the action processing. The actions are only influenced indirectly via the activation or deactivation of the steps. As a consequence, all saving actions remain active e.g. during the switch-over from automatic to manual mode!

7 Step sequence diagnosis via WIN-HMI

7.1 Step sequence diagnosis

General information

The diagnosis display always refers to a module of the selected control. In case of networked machines the diagnosis can also be displayed by other control panels.

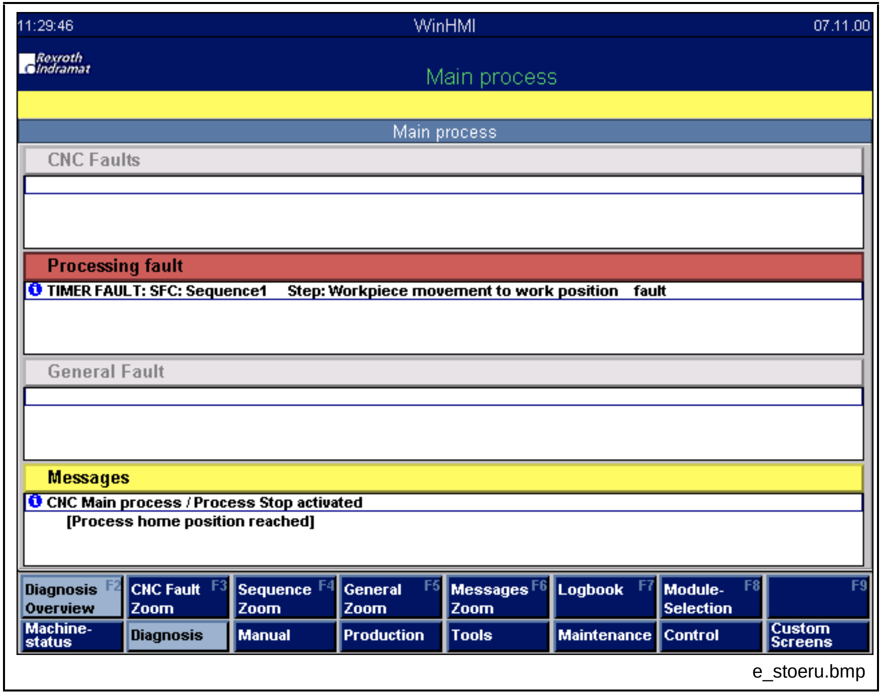


Abb. 7-32: Diagnoses Overview

Message types

- CNC faults: CNC faults of the processes
- Sequence faults: Faults of the IndraStep step sequence
- General faults: ProVi messages which were configured as general faults.
- Messages: ProVi messages which were configured as messages and CNC messages of the processes.

User guidance

If fault messages are active in the selected control, the "OP3“ key starts to flash red.

If the diagnosis overview is already on the screen and the fault has occurred in a module that is not displayed the "F8“ key flashes red. The faulty element is marked red in the module selection.

Multi-language function

- CNC messages/faults: multi-language texts of DOS
- ProVi messages/step sequence faults: Input in the message integrator

Info texts

Info texts can be created for the CNC faults, for general errors and for CNC/SPS messages. If an info text exists for a message, this is symbolized by a white 'i' on blue ground in the message line. An info text can be displayed with the INFO key.

- CNC messages/faults:
  info texts of the DOS interface
- ProVi messages:
  Input in the message integrator

Rexroth Indramat

- Sequence faults:
  Criteria analysis formed automatically from the IndraStep step sequence

NC indications If an NC comment is issued in the NC for a process, this comment is displayed in the second line of the corresponding NC message in square brackets.

ZOOM The window in which the cursor is positioned can be magnified via the Zoom in key. Individual fields can be zoomed directly via the keys "F3" - "F6".

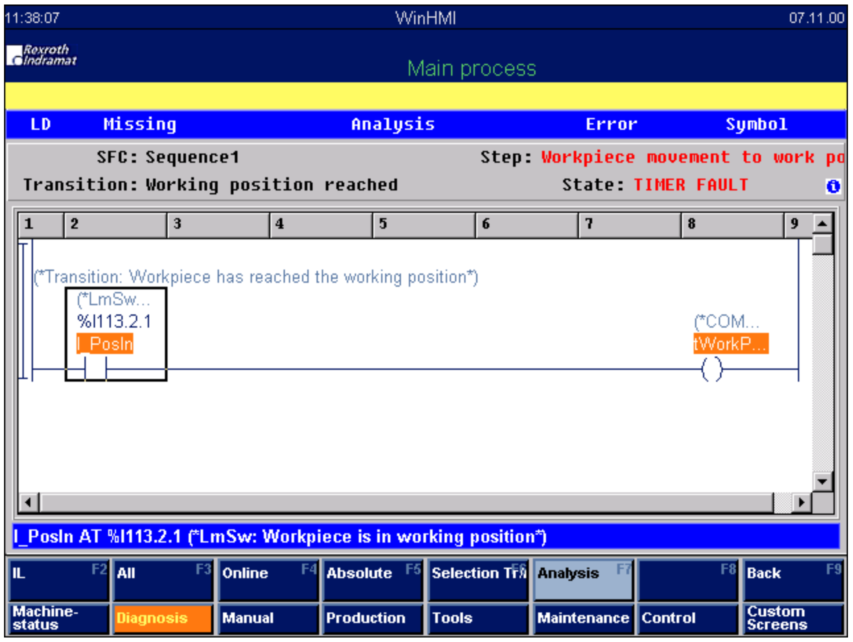
F3 - CNC fault ZOOM

F5 - general fault ZOOM

F6 - messages ZOOM

The step sequence diagnosis is generated automatically out of IndraStep function blocks and can be called out of the diagnosis overview via the INFO key.

In the step sequence diagnosis the currently active IndraStep-function block program code on contact level in the control is displayed.



E_SFCDIAGKOP.bmp

Abb. 7-33: Criteria analysis in LD

Displayed data

The display is subdivided into several areas:

Header Display of the current display options.

Step and sequence information (upper area)

- Step sequence names (name of the FB or the text from the MI (Message integrator))
- Step sequence operating mode (Auto, Manual, Fault...)
- Comment of the displayed step or text from the MI
- Comment of the displayed action or transition or text from the MI

SPS Code in IL or LD

- SPS Code of the displayed action or transition
- Comment on SPS variables
- I/O address of the SPS variables in case of global I/O variable

- Status of the SPS variables
- Logic result
- Not-fulfilled conditions (are displayed red)

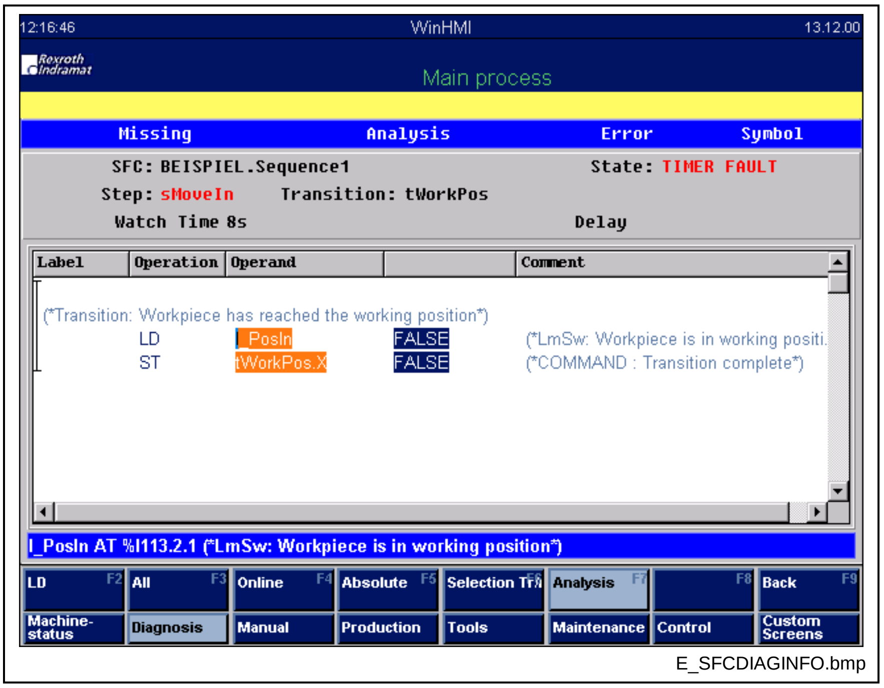
Information on the variable selected with the cursor (blue bar on the bottom)

- Variable name
- I/O address of the SPS variables in case of global I/O variable
- Comment of the SPS variables

Additional information

By pressing the INFO key, the upper part of the display can be switched over. In this area further information on faults are provided.

- Step sequence name (entity name of the FB)
- Step sequence operating mode (Auto, Manual, Fault...)
- Step name
- Action or transition name
- Qualifier (only for action display)
- Monitoring time of the step
- Waiting time of the step



E_SFCDIAGINFO.bmp

Abb. 7-34: Criteria analysis in IL with additional information

Display options

Online representation: In the online representation the current state of the variables will be displayed. (Switch-over with F4)

Frozen state In frozen state the state of the variable of the faulted step is displayed in the cycle in which the fault occurred. If the fault is acknowledged the diagnosis switches to the Online representation of the current step. (Switch-over with F4)

Switch-over between step, transition and action During the diagnosis start, the first transition is displayed when the step is fulfilled. If the condition of the step is not fulfilled, the step will be displayed. If there are several actions the first not-fulfilled action will be displayed.

An action is fulfilled when the VKE is TRUE at the end of the action.

Then, the displayed action or transition can be selected.

(Select with F6)

Types of representation

The SPS code can be represented in IL or in LD. When opening a window the type of representation selected last is selected again. The user can switch over between these two types of representation at any time.

(Switchover with F2)

E_SFCDIAGAWL.bmp

Abb. 7-35: Criteria analysis in IL

Types of criteria analysis

(Switchover with F3)

- All conditions: All SPS code lines are displayed independent of whether the contact is fulfilled or not.
- Missing conditions: Only the variables will be displayed which are required for the fulfilling of the condition such as:

All conditions		**Missing conditions**	
SPS Code	Status	SPS Code	Status
LD E1	TRUE	LD E2	FALSE
AND(		OR E4	FALSE
LD E2	FALSE	ST A1	FALSE
AND E3	TRUE		
OR E4	FALSE		
)			
ST A1	FALSE		

Fig. 7-36: Criteria analysis

Absolute or symbolic representation

The display can be switched over between the variable name and the absolute address. (Switch-over with F5)

7.2 Installation

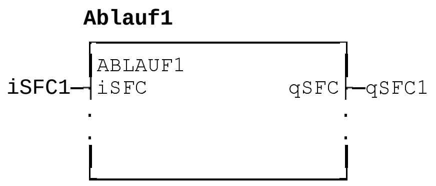
Prerequisites

Prerequisite for the use of the step sequence diagnosis of IndraStep is the WinHMI user interface. General installation and handling descriptions of the WIN-HMI user interface are contained in the document "Standardized user interface WIN-HMIWIN-HMI for automated manufacturing",
DOK-MTC200-WIN-HMI*Vxx-FK02-DE-P.

The operating mode functionality of IndraStep can also be used without WinHMI.

Log-on of the IndraStep step sequence in WinHMI

In order to carry out an indraStep step sequence diagnosis in WinHMI 'OP 3 Diagnosis', the corresponding step sequence has to be entered with the allocation name in the file Moduldef.INI.




```
;Default Moduldefinition MUI/GUI version 18Vxx

[ModulGeneral]

[ModulGeneral\ModulDescription]
DE = Einheit
EN = Unit

;[DeviceAddr0]
;SetupDiags =
;Warnings =
;StartConditions =

[DeviceAddr0\Modul1]
ModulName = Testen der Schrittketten
;Errors =
;Messages =

;[DeviceAddr0\Modul1\Process]
;Entry1 =

[DeviceAddr0\Modul1\Sfc]
Entry1= Ablauf1
Entry2= .....
```

Fig. 7-37: Example of an entry of a step sequence in the file Moduldef.INI.

This entry activates the diagnosis of the IndraStep step sequence by WIN-HMI.

7.3 Diagnoses in a case of error

Sequence faults

Errors occurring during the step sequence processing are displayed as sequence faults. This display is extracted from the allocation name of the step sequence, in this example it is 'Ablauf1' and the step comment in which the error occurred.

Program in the control:

```
FB Ablauf1

 sMoveOut          Move workpiece to transfer position

  tMoveOut         Waiting for clamping and door is open
```

Diagnosis display (OP3):

Sequence fault:

Sequence: **Ablauf1** Step: **Move workpiece to the transfer position** faulty

Fig. 7-38: Example sequence fault - application of the step comment

Criteria analysis

The analysis of the faulted step by WIN-HMI can be continued by means of a criteria analysis. In this process immediate stop monitoring or time monitoring errors are distinguished.

Immediate stop monitoring fault

The cause for the error was caused in a network in which '_Action.MonError' was set 'TRUE'. If an error occurs, the action, in which this flag was programmed, is analyzed for positive network results.

Time fault

The time T_max has expired without the transition being fulfilled. In case of an error, the action in which this transition generation was programmed is analyzed for negative network results. In this case a distinction is made between two possibilities.

Time fault transition

If the network of the transition generation was not fulfilled, but the network of the action was fulfilled, the network of the transition generation will be analyzed,

or

Time fault action

If the networks of the transition generation and the action are not fulfilled, the action network will be analyzed.

Controlling the diagnosis generation

50 Boolean and 10 non-Boolean states can be saved for each step for initial error analysis. In case of an error these states are evaluated by the criteria analysis. Networks which are required for the control of the step sequence can be deactivated for criteria analysis. For this purpose, the command <DiagOf> is used to activate the diagnosis data generation, and <DiagOn> to deactivate it.

aMonError

The same number of states is available for the action <aMonError> in which the general step sequence monitoring shall be realized. This means that in one step 50 Boolean and 10 non-Boolean states can be saved in one step fo the actions <aBEISPIEL> and <aMonError> (see Fig. 7-39: Example for deactivation of the diagnosis generation

DiagOf

<DiagOf> as network comment switches off the diagnosis generation for the related action block.

DiagOn

<DiagOn> as network comment switches the diagnosis generation for the related action block on again.

The deactivation of the diagnosis generation by means of <DiagOf> is only effective until the end of the corresponding action block.

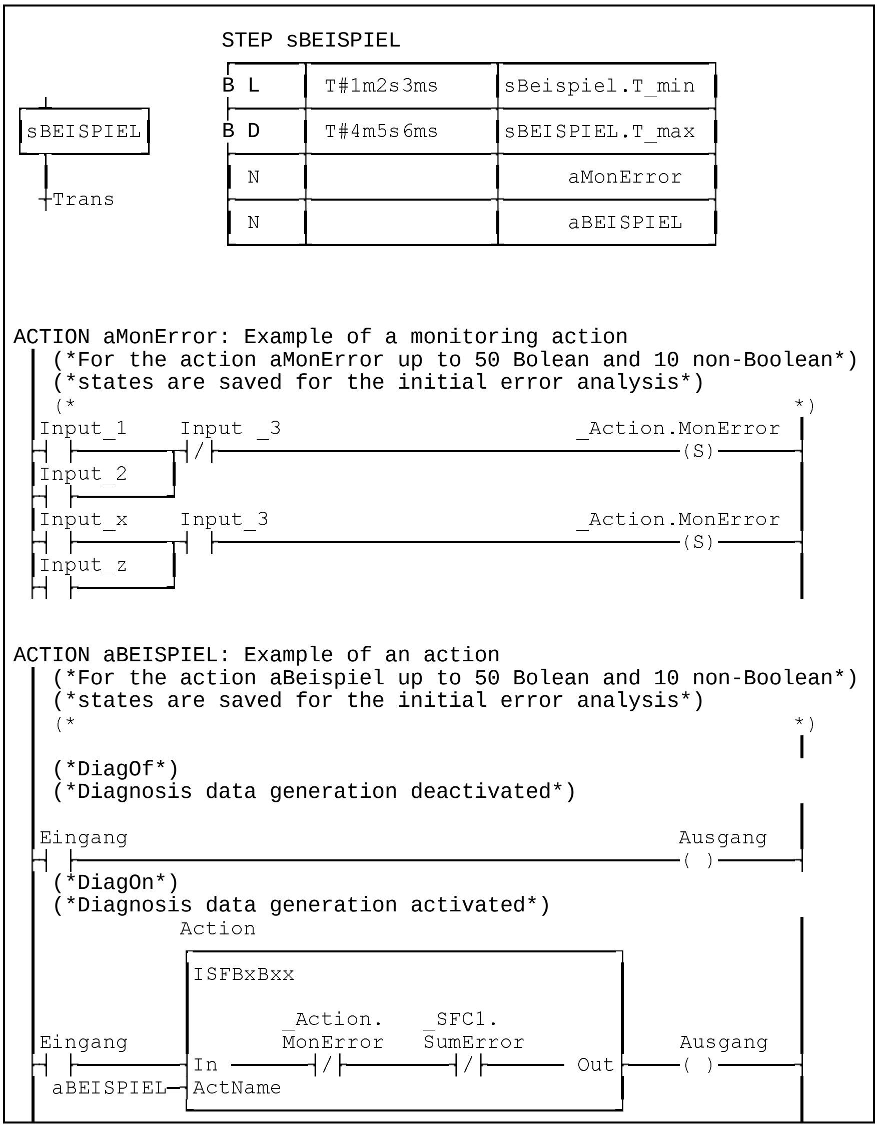


Fig. 7-39: Example for deactivation of the diagnosis generation

8 Example of an application

8.1 Program

Import section

```
IMPORT        Comment
TYPE
(***************************************************************************)
(*** Imports which are necessary for the Indramat step sequence ***)
 ISTYxAxx      (*Structure control inputs of the step sequence*)
 ISTYxBxx      (*Structure status outputs of the step sequence*)
 ISTYxGxx      (*Structure global action control data     *)
(*********** End of import of Indramat step-specific data **********)

FUNCTION


FUNCTION_BLOCK
 IS_DOOR1      (*Example IndraStep-SFC door control  *)
```

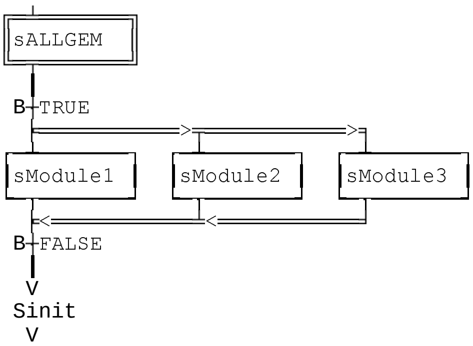
Fig. 8-40: Import section of the program

Allocation section

```
PROGRAM     BEISPIEL

VAR_INPUT
 I_Door1o   %I113.2.0                      (*LmSw: Door 1 is open*)
 I_Safety1  %I113.2.3                      (*Sw  : Door 1 safety bar*)
 I_Door1c   %I113.2.6                      (*LmSw: Door 1 closed*)
 .
 PbClrErr   %I112.3.3                      (*Pb  : SFC-clear error*)
 PbCtrlRes  %I112.3.6                      (*Pb  : SFC-reset to Init-Step*)
END_VAR

VAR_OUTPUT
 Q_Door1o   %Q113.0.0                      (*Q :Motor Open door 1*)
 Q_Door1c   %Q113.0.6                      (*Q :Motor Close door 1*)
 L_Door1o   %Q113.0.1                      (*LPb :Door 1 is open*)
 .
 L_Reset    %Q112.1.6                      (*LPb: SFC-reset*)

END_VAR

VAR
(***************************************************************************)
(*     IndraStep SFC                                                      *)
(*Standard-Variables (don't delete this area)                             *)
(***********************************************************************)
 _Action                  ISTY2G00         (*Global temporary action control*)
(*Variables of the general user program section, here edge determination *)
(*of the control panel*)
 Eg_Auto                  BOOL             (*Edge evaluation of the keys*)
 Eg_Manu                  BOOL             (*Edge evaluation of the keys*)
 Eg_Sync                  BOOL             (*Edge evaluation of the keys*)
 .
(***************************************************************************)
(************************ Step sequences *********************************)
(***************************************************************************)
 iSFC1                    ISTY2A00         (*Control inputs IndraStep*)
 qSFC1                    ISTY2B00         (*Status outputs IndraStep*)
 Door1                    IS_DOOR1         (*Example of door control with IndraStep*)
END_VAR

VAR_GLOBAL
(******************** Global temporary action control *****************)
 _Action                  ISTY2G00         (*Global temporary action control*)
 I_Door1o   %I113.2.0     BOOL             (*LmSw: Door 1 is open*)
 I_Safety1  %I113.2.3     BOOL             (*Sw  : Door 1 safety bar*)
```

Fig. 8-41: Allocation section of the program

Implementation




```
sALLGEM   General program section
sModule1  Program section for station 1
sModule2  Program section for station 2
sModule3  Program section for station 3
```

STEP sALLGEM: General program section

S		aBEDIEN	(*Control panel*)

STEP sModule1: Program section for station 1

N		aBEISPIEL	(*Example IndraStep*)

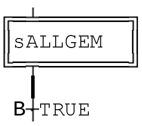
STEP sModule2: Program section for station 2

N		aSTAT02	(*Program section station 2*)

STEP sModule3: Program section for station 3

N		aSTAT03	(*Program section station 3*)

Fig. 8-42: Step sequence implementation for program structuring



STEP sALLGEM: General program section

S		aBEDIEN	(*Control panel*)

ACTION aBEDIEN: Control panel

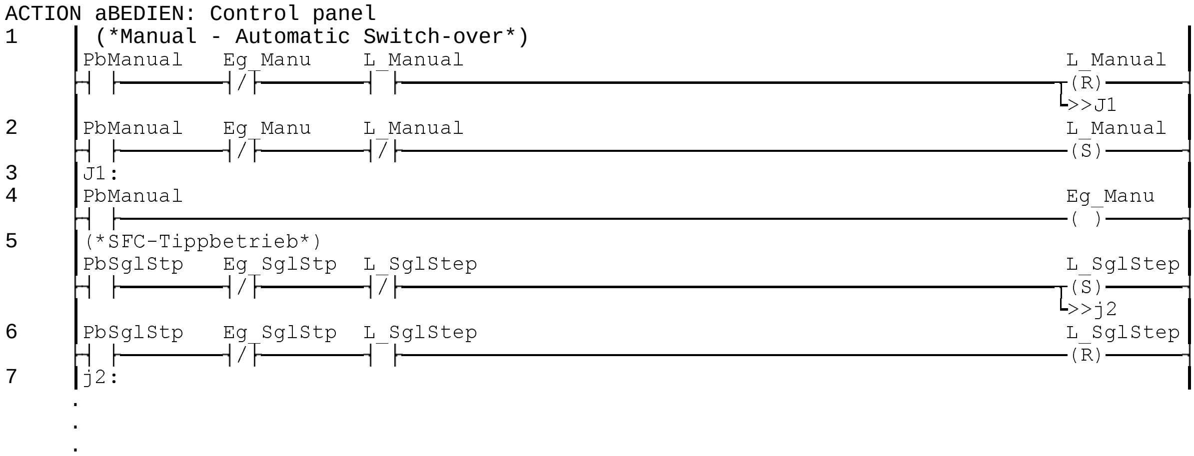


Fig. 8-43: Implementation of the general program section

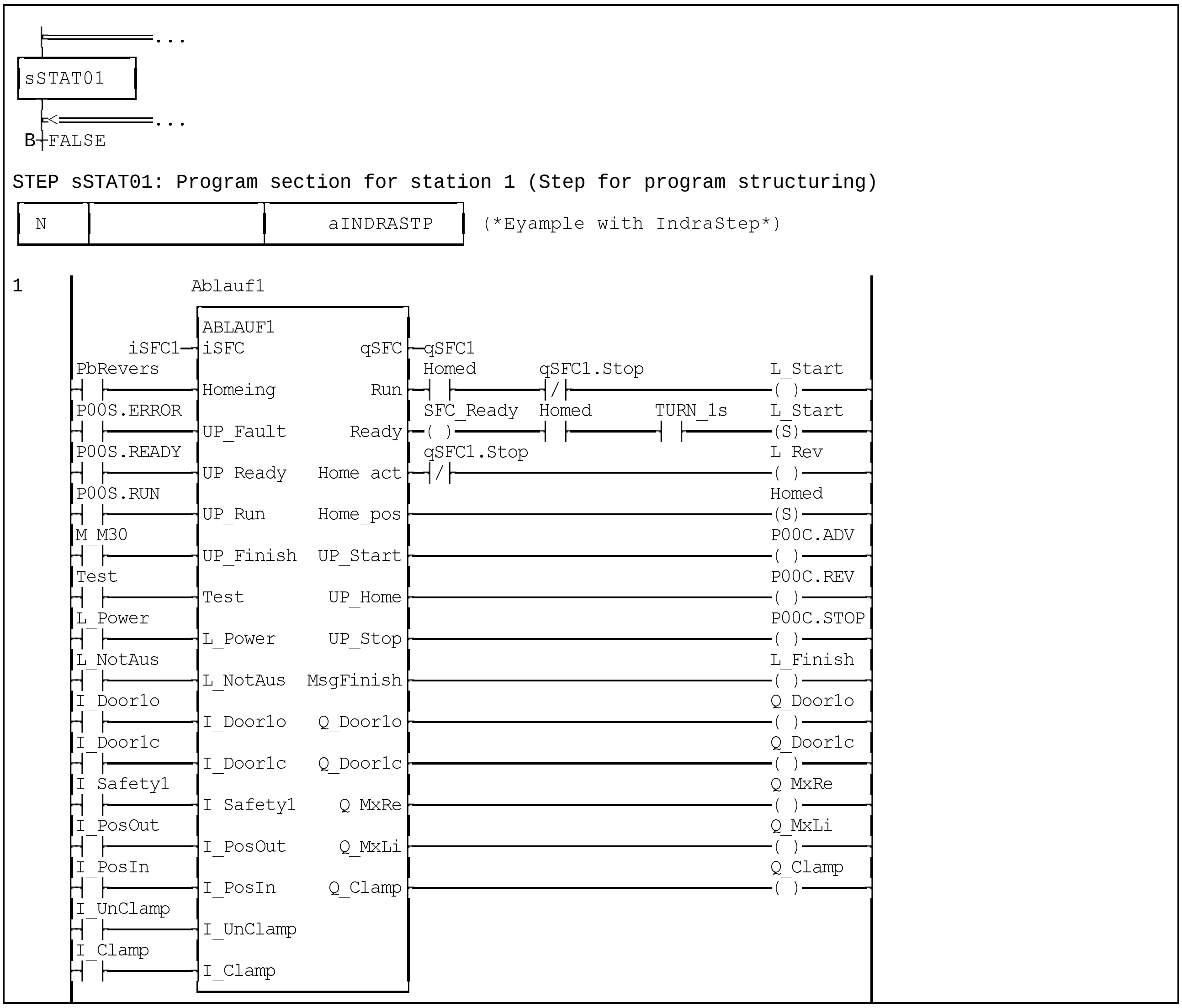


Fig. 8-44: Function block implementation with IndraStep (activation of step sequence function blocks)

8.2 IndraStep function block

Import

```
IMPORT        Comment
TYPE
(*************************************************************)
(********************** IndraStep SFC ***********************)
(* Standard Types of IndraStep SFC        (do not clear)****)
 ISTYxAxx      (*commands of SFC          (do not clear)*)
 ISTYxBxx      (*status of SFC            (do not clear)*)
 ISTYxCxx      (*SFC data                 (do not clear)*)
 ISTYxDxx      (*Step data                (do not clear)*)
 ISTYxExx      (*Action data              (do not clear)*)
 ISTYxFxx      (*Transition data          (do not clear)*)
 ISTYxGxx      (*Global data action control(do not clear)*)
 ISTYxHxx      (*Diagnosis-values Boolean  (do not clear)*)
 ISTYxIxx      (*Diagnosis-values non bool.(do not clear)*)
(************************************************************)
 ISTYxXxx      (*Example of temporary Flags*)
FUNCTION
FUNCTION_BLOCK
(**********************************************************)
(****************** IndraStep SFC **********************)
(* Standard FB of IndraStep SFC           (do not clear)*)
 ISFBxBxx      (*FB controls application output*)
(*************************************************************************)
```

Fig. 8-45: Import of IndraStep function block

Allocation

```
Identifier  AT              TYPE        :=          Comment
FUNCTION_BLOCK              ABLAUF1
VAR_INPUT
(****************************************************************************)
(*************************** IndraStep SFC ********************************)
(************ Command structure of SFC (do not clear this area) ***********)
 iSFC                       ISTYxAxx                (*Command structure of IndraStep SFC*)
(********************************************************************)
 Homeing                    BOOL                    (*Home position*)
 UP_Fault                   BOOL                    (*Subprocess FAULT*)
 UP_Ready                   BOOL                    (*Subprocess READY*)
 UP_Run                     BOOL                    (*Subprocess RUN*)
 UP_Finish                  BOOL                    (*Subprocess processing COMPLETED*)
 (********************************** Inputs *******************************)
 L_Power                    BOOL                    (*LPb: Power On*)
 L_NotAus                   BOOL                    (*LPb: Emergency stop active*)
 I_Door1o                   BOOL                    (*LmSw: Door 1 is open*)
 I_Door1c                   BOOL                    (*LmSw: Door 1 is closed*)
 I_Safety1                  BOOL                    (*Sw  : Door 1 safety bar*)
 I_PosOut                   BOOL                    (*LmSw: Workpiece in transfer position*)
 I_PosIn                    BOOL                    (*LmSw: Workpiece in machining position*)
 I_UnClamp                  BOOL                    (*LmSw: Workpiece is not clamped*)
 I_Clamp                    BOOL                    (*LmSw: Workpiece is clamped*)
END_VAR

VAR_OUTPUT
(****************************************************************************)
(************************* IndraStep SFC **********************************)
(****** Status structure of step chain (do not clear this area) ***********)
 qSFC                       ISTYxBxx                (*Status structure of IndraStep SFC*)
(****************************************************************************)
 Run                        BOOL                    (*Sequence is active*)
 Ready                      BOOL                    (*Ready for start*)
 Home_act                   BOOL                    (*Homeing is active*)
 Home_pos                   BOOL                    (*Homeing reached*)
 UP_Start                   BOOL                    (*Subprocess START*)
 UP_Home                    BOOL                    (*Subprocess HOME*)
 UP_Stop                    BOOL                    (*Subprocess STOP*)
 MsgFinish                  BOOL                    (*Workpiece machined at transfer position*)
(******************************* Outputs ********************************)
 Q_Door1o                   BOOL                    (*Q : Open door 1*)
 Q_Door1c                   BOOL                    (*Q : Close door 1*)
 Q_MxRe                     BOOL                    (*Q : Move in workpiece*)
 Q_MxLi                     BOOL                    (*Q : Move out workpiece*)
 Q_Clamp                    BOOL                    (*Q : Clamp workpiece*)
END_VAR

VAR
(****************************************************************************)
(****************************** IndraStep SFC ****************************)
(************* Standard variables of SFC (do not clear this area) *********)
(*For this FB the global variable _Action (type ISFC3Gxx) must be entered*)
(*in level program.                                                       *)
 Action                     ISFBxBxx                (*Function block for IndraStep actions*)
 isty3h00                   ISTYxHxx                (*diagnosis-values for Boolean*)
 isty3i00                   ISTYxIxx                (*diagnosis-values for non-Boolean*)
 isty3h01                   ISTYxHxx                (*diagnosis-values for Boolean*)
 isty3i01                   ISTYxIxx                (*diagnosis-values for non-Boolean*)
 tmpStpP                    ^ISTYxDxx               (*Temporary SFC pointer*)
(****************************************************************************)
 _TMP                       ISTYxXxx                (*Example for temporary Flags*)
 btv05NO                    USINT                   (*BTV05 screen number*)
 ton1                       TON
 T1_1s                      BOOL                    (*Limit switch overlapping for longer than 1s*)
END_VAR

VAR RETAIN
END_VAR

VAR_EXTERNAL
 Trigger                    BOOL                    (*Trigger*)
(****************************************************************************)
(****************************** IndraStep SFC ****************************)
(*This global variable _Action (type ISFC3Gxx) must be entered in level   *)
(*program.*)
 _Action                    ISTYxGxx                (*Global variable for action control*)
(****************************************************************************)
END_VAR
```

Fig. 8-46: Allocation of IndraStep function block

Implementation

```
Template of the step sequence:       ISFBxAxx
Allocation of the step sequence:     _SFC1
Type of the step sequence:           ISTYxCxx
Type of steps:                       ISTYxDxx
Type of transitions:                 ISTYxFxx
Type of actions:                     ISTYxExx
System action Step-In:               _aIn1
System action Step-Out:              _aOut1
System action Diag:                  _aDiag1

_STEP1:                             ^ISTYxDxx
```

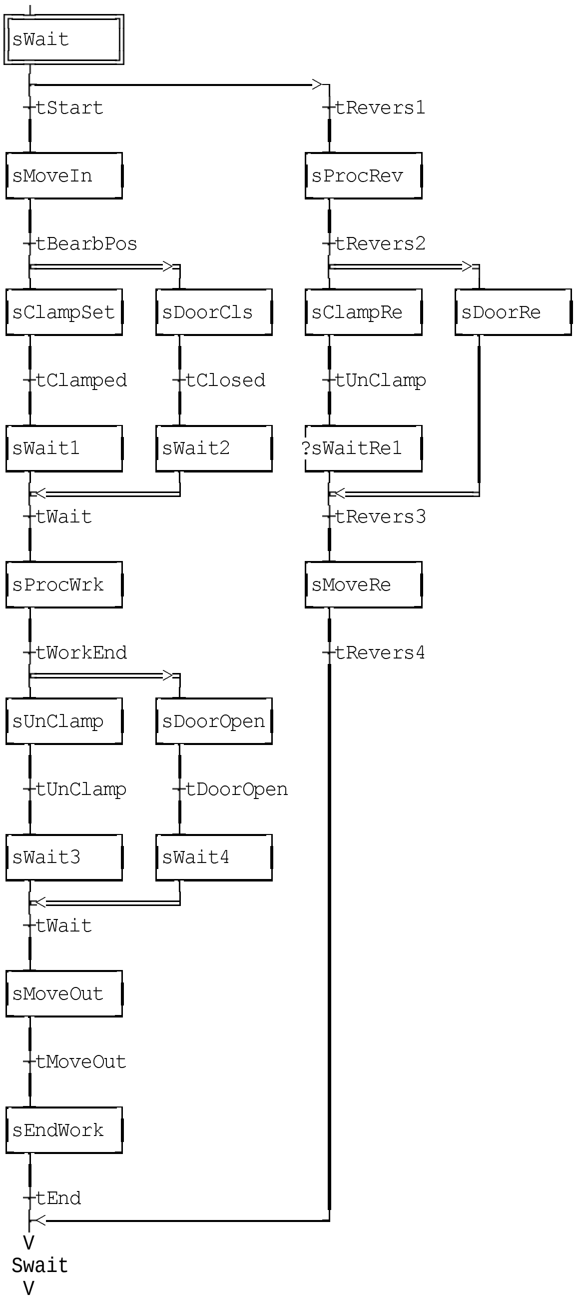
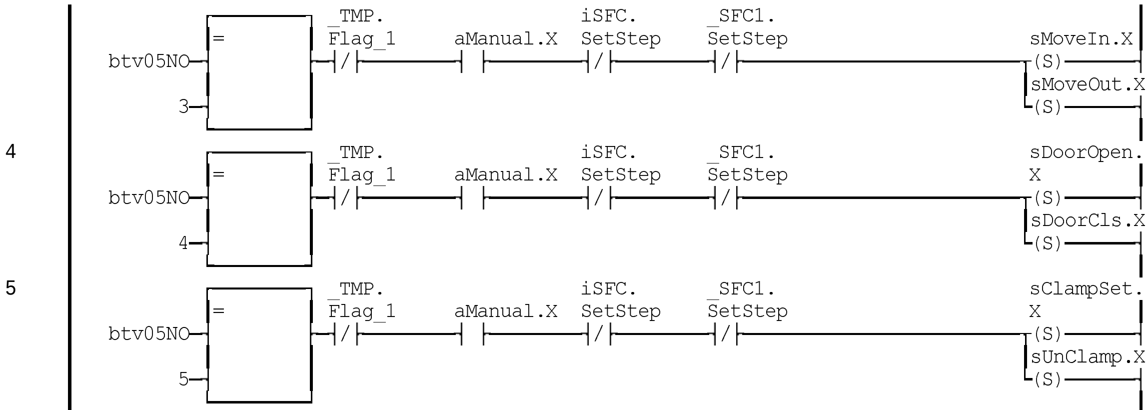
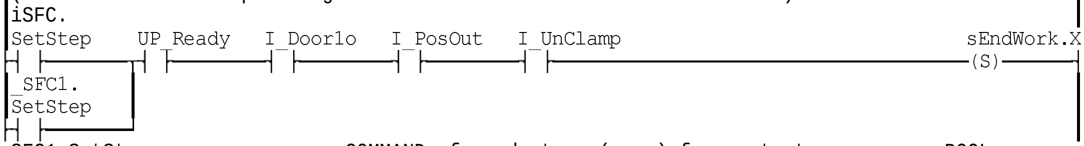


Fig. 8-47: Step sequence overview

```
sWait

STEP sWait: Initial step - warting for start commmand
| R |  | Always     | (**)
| R |  | aManual    | (**)
| N |  | aMonStart  | (*Monitoring start command*)
| N |  | Wait       | (*Waiting for start command or home positon.*)

ACTION Always: This action is always processed
1   (*DiagOf*)
    (*******  Rung IndraStep control : do not clear  *******)
                                                        _SFC1.
                                                        SfcCycl
    ----------------------------------------------------(R)----
2   (***** End of rung IndraStep control : do not clear ******)
    (**********************************************************************)
    (***************************** Manual mode *****************************)
    (**********************************************************************)
    (*Machining the aManual-action in the operating mode manual : *)
    _SFC1.
    Manual                                              aManual.X
    -| |------------------------------------------------( )----
3    (*Release UP-Stop, when the step sequence is stopped:*)
    qSFC.Stop                                           UP_Stop
    -| |------------------------------------------------( )----
4   (*                    External homeing request               *)
    (*Switch immeadiatly to Revers branch, when step sequence is not yet  *)
    (*active in the Revers-branch:*)
                                                        _SFC1.
    Homeing      Home_act                               GoStep
    -| |---------|/|--------+---------------------------+-( )----
    _SFC1.                  |                           |
    GoStep                  |                           | sProcRev.X
    -| |--------------------+                           +-(S)----
5   (*Restart stopped reverse program by Homeing request:*)
                                                        _SFC1.
    Homeing     qSFC.Stop  Home_act                     Start
    -| |---------| |--------| |-------------------------( )----
   _SFC1.GoStep................ COMMAND: go to forced steps....................... BOOL
   _SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
   _SFC1.SfcCycl............... INTERN : Internal cycle flag...................... BOOL
   _SFC1.Start................. COMMAND: SFC start................................ BOOL
   aManual.X................... COMMAND : force Action............................ BOOL
   Home_act.................... Homeing is active................................. BOOL
   Homeing..................... Home position.................................... BOOL
   QSFC.Stop................... SFC stopped...................................... BOOL
   SProcRev.X.................. AUTO : Step active  / MANU : Force step........... BOOL
   UP_Stop..................... Subprocess STOP.................................. BOOL

ACTION aManual
1   (*DiagOf*)
    (*Selecting the step via the BTV20 HMI buttons in manual mode       *)
    (*only when the botton for the step sequence has been selected and no*)
    (*synchronizations request is active:                              *)
                                                        _TMP.
          iSFC.   |=  |                                 Flag_1
       indOPNR----|   |---------------------------------( )----
              4---|   |
2                          _TMP.              iSFC.      _SFC1.         sDoorOpen.
          iSFC.   |=  |    Flag_1   aManual.X  SetStep   SetStep        X
      indSCRNR----|   |----| |-------| |--------|/|--------|/|---------+-( )----
                  |   |                                                | sDoorCls.X
            802---|   |                                                +-( )----
                                                                       | sMoveIn.X
                                                                       +-( )----
                                                                       | sMoveOut.X
                                                                       +-( )----
                                                                       | sClampSet.
                                                                       | X
                                                                       +-( )----
                                                                       | sUnClamp.X
                                                                       +-( )----
3   (*Alternative activation of the steps in manual mode by e.g. external*)
    (*manual operating device:*)
```

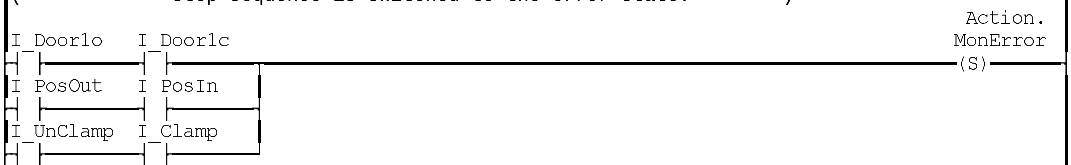



```
6    (*Example of step sequence synchronization:*)
     (*When the step sequence is in home position status, it is possible to synchronize*)
     (*directly from the manual mode to the step ready for transfer,    *)
     (*e.g. to transfer the end of machining signal to a higher process*)
     (*in the automatic operating mode.                                 *)
```

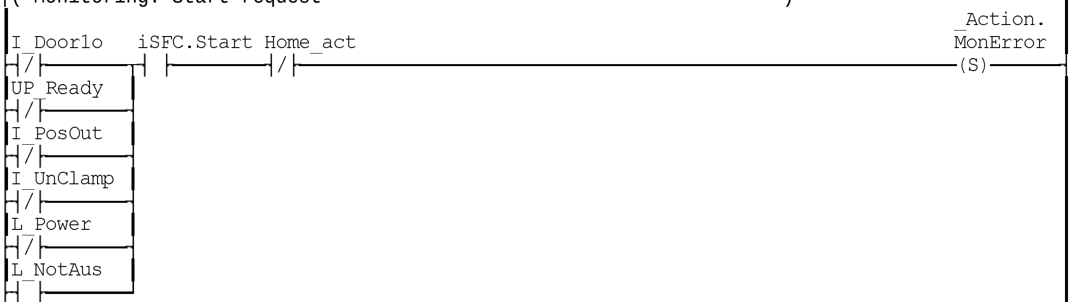



```
_SFC1.SetStep............... COMMAND: forced steps (man.) for restart .......... BOOL
_TMP.Flag_1................. temp. flag any Boolean............................ BOOL
aManual.X................... COMMAND : force Action............................ BOOL
btv05NO..................... BTV05 Screen number............................... USINT
I_Door1o.................... LmSw: Door 1 is open.............................. BOOL
I_PosOut.................... LmSw: Workpiece in transfer position.............. BOOL
I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
ISFC.indOPNR................ HMI OP-area number OP2 - OP9...................... INT
ISFC.indSCRNR............... active screen-number.............................. INT
ISFC.SetStep................ forced steps (manual) for restart (auto)........... BOOL
SClampSet.X................. AUTO : Step active  / MANU : Force step........... BOOL
SDoorCls.X.................. AUTO : Step active  / MANU : Force step........... BOOL
SDoorOpen.X................. AUTO : Step active  / MANU : Force step........... BOOL
SEndWork.X.................. AUTO : Step active  / MANU : Force step........... BOOL
SMoveIn.X................... AUTO : Step active  / MANU : Force step........... BOOL
SMoveOut.X.................. AUTO : Step active  / MANU : Force step........... BOOL
SUnClamp.X.................. AUTO : Step active  / MANU : Force step........... BOOL
UP_Ready.................... Subprocess READY............... ................. BOOL
```

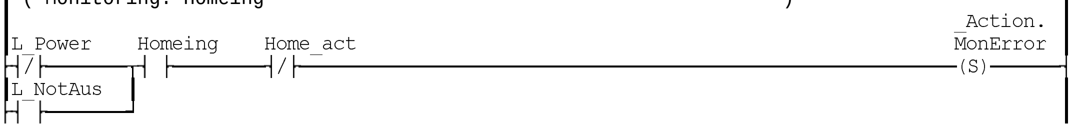
```
ACTION aMonStart: Monitoring start command
1    (*Monitoring: when the limit switches respond simultaneously, the *)
     (*            step sequence is switched to the error state.       *)
```




```
2    (*Monitoring: Start request                                       *)
```




```
3     (*Monitoring: Homeing                                            *)
```



Rexroth Indramat

```
_Action.MonError............. COMMAND : monitoring failure........................... BOOL
     Home_act.................... Homeing is active................................ BOOL
     Homeing..................... Home position.................................... BOOL
     I_Clamp..................... LmSw: Workpiece is clamped........................ BOOL
     I_Door1c.................... LmSw: Door 1 is closed............................ BOOL
     I_Door1o.................... LmSw: Door 1 is open.............................. BOOL
     I_PosIn..................... LmSw: Workpiece in machining position............. BOOL
     I_PosOut.................... LmSw: Workpiece in transfer position.............. BOOL
     I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
     ISFC.Start.................. SFC start........................................ BOOL
     L_NotAus.................... LPb: Emergency stop active........................ BOOL
     L_Power..................... LPb: Power On.................................... BOOL
     UP_Ready.................... Subprocess READY................................. BOOL

ACTION Wait: Warting for start command or homeing
1     (*CtrlOf - deactivating the intervention of FB Action *)
      (*Issueing step sequence active (when the waiting step is not active) and*)
      (*rest Home-Position-message.*)
                 Action
            +-------------------+
            |ISFBxBxx           |
      Wait.Q|                   |                                            Run
      -|/|--|In              Out|------------------------------------------+--( )--
            |                   |                                          | Home_pos
      Wait--|ActName            |                                          +--(R)--
            +-------------------+
2     (*Output of the message Ready (home position) and thus the ready signal is  *)
      (*available. This message should be displayed in all operating modes*)
      (*except for the case when there is an error *)
                                                  Action
                                                +-------------------+
                                                |ISFBxBxx           |
                                       _SFC1.   |                   |
      I_Door1o   UP_Ready   I_PosOut   I_UnClamp   SumError      |                   |      Ready
      -| |--------| |--------| |--------| |--------|/|--------|In              Out|------( )--
                                            Wait--|ActName            |
                                                +-------------------+

3     (*Transition generation: Sequence start                                     *)
      (*                     The transition only switches in correct automatic- *)
      (*                     mode and therefore does not require further *)
      (*                     disables.                                     *)
      iSFC.Start Ready                                                     tStart.X
      -| |--------| |------------------------------------------------------( )--
4     (*Transitionsbildung : Grundstellung fahren*)
      Homeing     qSFC.Stop                                                tRevers1.X
      -| |--------|/|------------------------------------------------------( )--
     _SFC1.SumError.............. STATUS : SFC summary error flag................... BOOL
     Action...................... Function block for IndraStep actiones............. ISFBxBxx
     Home_pos.................... Homeing reached.................................. BOOL
     Homeing..................... Home position.................................... BOOL
     I_Door1o.................... LmSw: Door 1 is open.............................. BOOL
     I_PosOut.................... LmSw: Workpiece in transfer position.............. BOOL
     I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
     ISFC.Start.................. SFC start........................................ BOOL
     QSFC.Stop................... SFC stopped...................................... BOOL
     Ready....................... Ready for start.................................. BOOL
     Run......................... Sequence is active............................... BOOL
     TRevers1.X.................. COMMAND : Transition complete..................... BOOL
     TStart.X.................... COMMAND : Transition complete..................... BOOL
     UP_Ready.................... Subprocess READY................................. BOOL
     Wait........................ Waiting for start command or home position ........ ISTYxExx
     Wait.Q...................... STATUS  : Action active................ .......... BOOL
```

Fig. 8-48: Step 'sWait'

```
| N |            |      aMonError |(*Move workpiece to machining position*)

ACTION aMonError: Monitoring
1     (*Aktion Monitoring: In case of limit switch overlapping, active Emergency stop or *)
      (*                   a drop in power, the step sequence is switched to             *)
      (*                   the error state.                                              *)
                                                                                  _Action.
       I_Door1o   I_Door1c                                                        MonError
      -| |--------| |----------------------------------------------------------------(S)-
       I_PosOut   I_PosIn    T1_1s
      -| |--------| |--------| |---
       I_UnClamp  I_Clamp
      -| |--------| |---
       UP_Fault
      -| |---
       L_Power
      -|/|---
       L_NotAus
      -| |---
2     (*CtrlOf - Deactivation of the intervention of the FB Action                      *)
      (*DiagOf - Diagnosis generation for this network can be suppressed*)
      (*         since this part is only used for controlling the monitoring error message.*)
      (*                                                                                 *)
      (**)
      (*Suppressing monitor error messages when no automatic or in manual mode no step is- *)
      (*activated.*)
                 _STEP1^.                                                         _Action.
       _SFC1.Auto ManEnable                                                        MonError
      -|/|--------|/|--------------------------------------------------------------(R)-
                 _SFC1.
                 Manual
                 -|/|-
3     (*For time delay of limit switch overlapping the timer can be programmed for the  *)
      (*criteria analysis outside the proper diagnosis.*)
      (*For this purpose and in this example the timer ton1 and the marker*)
      (*T1_1s are crated.                                                  *)
                           ton1
                          +-----------+
                          |TON        |
       I_PosOut   I_PosIn |           |                                        T1_1s
      -| |--------| |-----|IN_      Q_|----------------------------------------( )-
                   T#1s---|PT_     ET_|-
                          +-----------+
    _Action.MonError............ COMMAND : monitoring failure...................... BOOL
    _SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
    _SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
    _STEP1^.ManEnable........... .................................................. BOOL
    I_Clamp..................... LmSw: Workpiece is clamped........................ BOOL
    I_Door1c.................... LmSw: Door 1 is closed............................ BOOL
    I_Door1o.................... LmSw: Door 1 is open.............................. BOOL
    I_PosIn..................... LmSw: Workpiece in machining position............. BOOL
    I_PosOut.................... LmSw: Workpiece in transfer position.............. BOOL
    I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
    L_NotAus.................... LPb: Emergency stop active........................ BOOL
    L_Power..................... LPb: Power On..................................... BOOL
    T1_1s....................... Limit switch overlapping for longer than 1s........ BOOL
    ton1........................ .................................................. TON
    UP_Fault.................... Subprocess ERROR.................................. BOOL
```

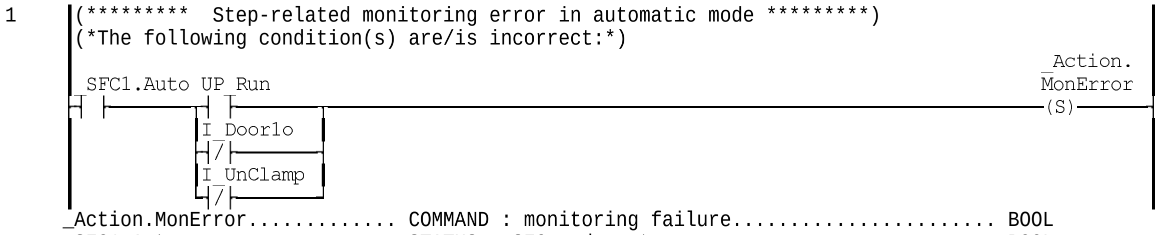
Fig. 8-49: Action 'aMonError'

sMoveIn

STEP sMoveIn: Move workpiece from transfer position to machining position

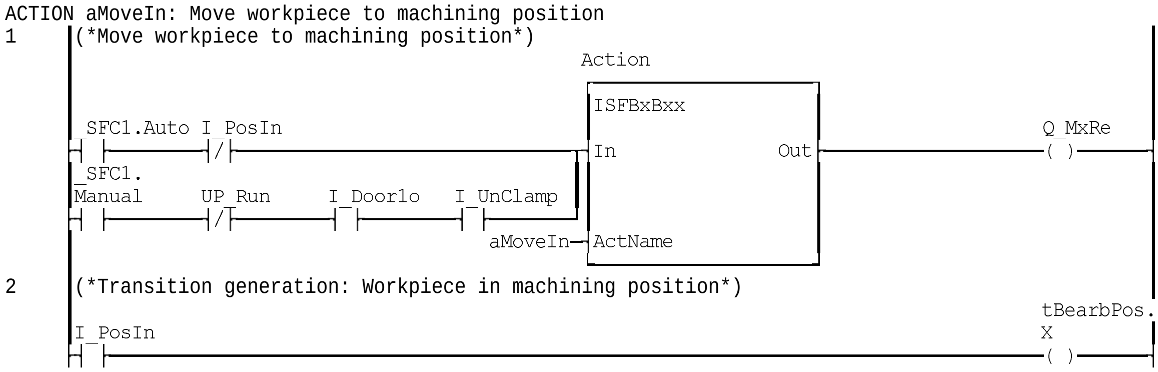
B D	T#8s	»In.T_max	(*Move workpiece to machining position*)
N		aMonError	(*Move workpiece to machining position*)
N		aMonMove	(*Monitoring*)
N		aMoveIn	(*Move workpiece to machining position*)

ACTION aMonMove: Monitorin




```
_Action.MonError............ COMMAND : monitoring failure...................... BOOL
_SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
I_Door1o.................... LmSw: Door 1 is open.............................. BOOL
I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
UP_Run...................... Subprocess RUN.................................... BOOL
```

ACTION aMoveIn: Move workpiece to machining position




```
_SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
_SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
Action...................... Function block for IndraStep actiones............. ISFBxBxx
AMoveIn..................... Move workpiece to machining position.............. ISTYxExx
I_Door1o.................... LmSw: Door 1 is open.............................. BOOL
I_PosIn..................... LmSw: Workpiece in machining position............. BOOL
I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
Q_MxRe...................... Q : Move in workpiece............................. BOOL
TBearbPos.X................. COMMAND : Transition complete...................... BOOL
UP_Run...................... Subprocess RUN.................................... BOOL
```

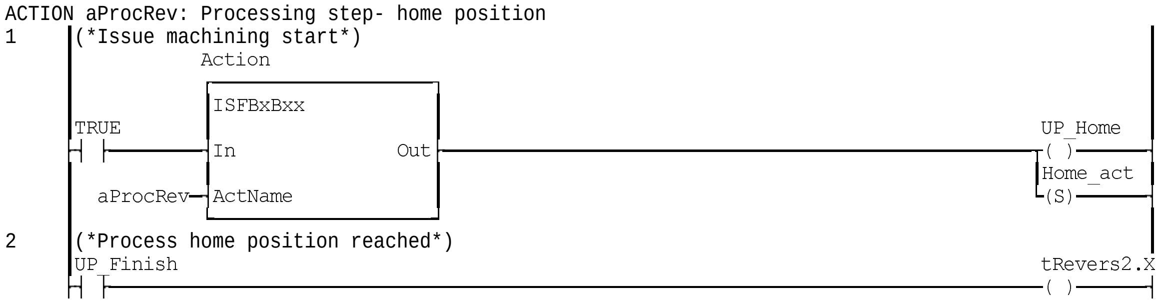
Fig. 8-50: Step 'sMoveIn'

sProcRev

STEP sProcRev: Processing step Revers

N		aMonError	(*Processing step - home position*)
N		aProcRev	(*Processing step - home position*)

ACTION aProcRev: Processing step- home position




```
Action....................... Function block for IndraStep actiones............. ISFBxBxx
AProcRev..................... Machining process - home position.................. ISTYxExx
Home_act..................... Homeing is active................................. BOOL
TRevers2.X................... COMMAND : Transition complete...................... BOOL
UP_Finish.................... Subprocess processing COMPLETED.................... BOOL
UP_Home...................... Subprocess HOME.................................... BOOL
```

Fig. 8-51: Step 'sProcRev'

```
sClampSet
STEP sClampSet: Clamp workpiece
```

B D	T#2s	»et.T_max	(*Maximum time close clamping*)
N		aMonError	(*Clamp workpiece*)
N		aMonClamp	(*Monitoring clamping not possible*)
N		aClampSet	(*Clamp workpiece*)

```
ACTION aMonClamp: Monitoring step-related: Clamping
1     (*********  Step-related monitoring error in automatic mode *********)
      (*The following condition(s) is(are) incorrect:*)
                                                                              _Action.
       _SFC1.Auto I_PosIn                                                     MonError
      --| |--------|/|--------------------------------------------------------(S)---
     _SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
     I_PosIn..................... LmSw: Workpiece in machining position............. BOOL
     _Action.MonError............ COMMAND : monitoring failure....................... BOOL

ACTION aClampSet: Move workpiece to machining position
1     (**************************** Workpiece clamping ***************************)
      (* Automatic : As long as the limit switch I_Clamp reports not clamped     *)
      (*             the output is activated normally. Only when the limit       *)
      (*             position is reached, the output is set to saving via the    *)
      (*             transitions network.                                        *)
      (* Manuell   : The output is activated in dependence of the allowed machine*)
      (*             status. In this case too the output is set to saving via    *)
      (*             the transition network when reaching the limit position of  *)
      (*             the output.                                                 *)
                           Action
                        ISFBxBxx
       _SFC1.Auto I_Clamp                                                     Q_Clamp
      --| |--------|/|-------In             Out-------------------------------( )---
       _SFC1.
      Manual      I_PosIn
      --| |--------| |--
                  I_PosOut
                  -| |--
              aClampSet--ActName

2     (***************** Transition generation workpiece clamping *****************)
      (*             When reaching the end position the tranisiton marker            *)
      (*             is generated and the output is set to saving. To avoid          *)
      (*             that the action post-processing immediately switches off the output*)
      (*             when quitting the step, the intervention of the FB Action       *)
      (*             has to be suppressed for this Logig in                          *)
      (*             this network with CtrlOf                                        *)
      (*CtrlOf*)
      I_Clamp                                                                 tClamped.X
      --| |-------------------------------------------------------------------( )---
                                                                              Q_Clamp
                                                                              (S)---
     _SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
     _SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
     I_Clamp..................... LmSw: Workpiece is clamped........................ BOOL
     I_PosIn..................... LmSw: Workpiece in machining position............. BOOL
     I_PosOut.................... LmSw: Workpiece in transfer position.............. BOOL
     Q_Clamp..................... Q : Clamp workpiece............................... BOOL
     Tclamped.X.................. COMMAND : Transition complete..................... BOOL
```

Fig. 8-52: Step 'sClampSet'

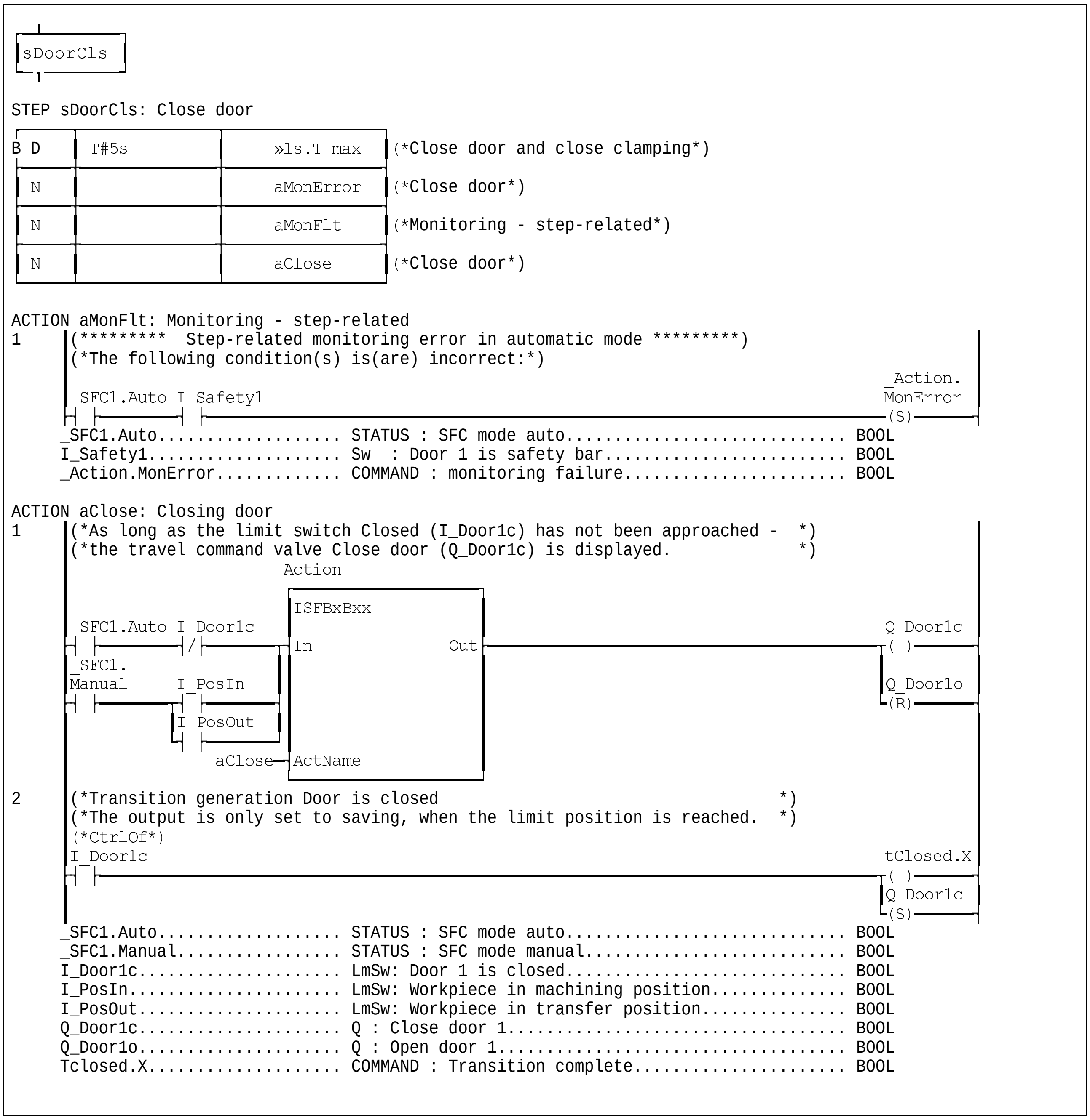


Fig. 8-53: Step 'sDoorCls'

Rexroth Indramat

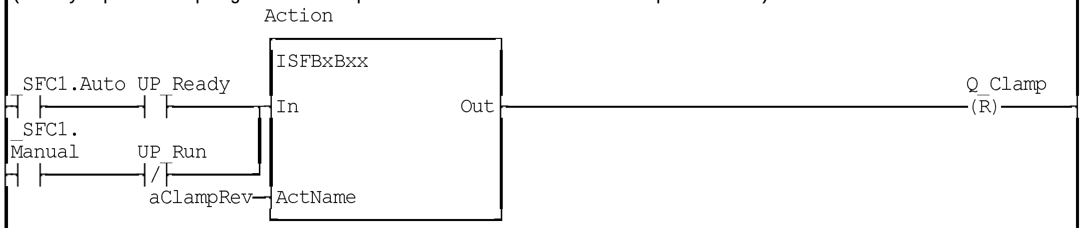
sClampRe

STEP sClampRe: Open clamping - home position –

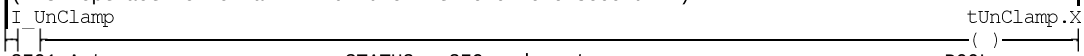
B L	T#1s	»Re.T_min		(*Minimum time Open clamping*)
N		aMonError		(*Open clamping- Move to home position*)
N		aClampRev		(*Open clamping- Move to home position*)

ACTION aClampRev: Open clamping- Move to home position

1 (*Only open clamping when the process has moved to home position*)



2 (*Transition generation Clamping open. *)
(**)
(*The L-action sClampRe.T_min 1s in the step sClampRe this action*)
(*is kept active for a minimum dwell time of one second. *)




```
_SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
_SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
aClampRev................... Open clamping- Move to home position............... ISTYxExx
Action...................... Function block for IndraStep actiones............. ISFBxBxx
I_UnClamp................... LmSw: Workpiece is not clamped..................... BOOL
Q_Clamp..................... Q : Clamp workpiece............................... BOOL
TunClamp.X.................. COMMAND : Transition complete..................... BOOL
UP_Ready.................... Subprocess READY.................................. BOOL
UP_Run...................... Subprocess RUN.................................... BOOL
```

Fig. 8-54: Step 'sClampRe'

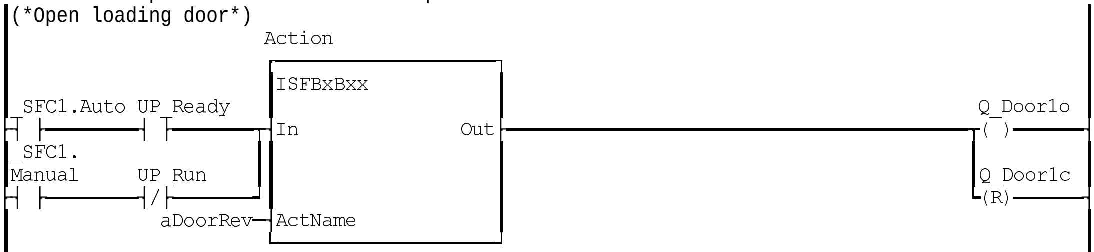
sDoorRe

STEP sDoorRe: Open loading door - Home position -

N		aDoorRev	(*Open door - Move to home position*)

ACTION aDoorRev: Open door - Move to home position

1 (*Open loading door*)



2 (*Transitions: Door is open*)
(*Ctrl0f*)

3 (*Since the waiting step after Open clamping has to be active,*)
(*here, only the transition condition of the door has to be taken into*)
(*account.*)

```
_SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
_SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
I_Door1o.................... LmSw: Door 1 is open.............................. BOOL
Q_Door1c.................... Q : Close door 1.................................. BOOL
Q_Door1o.................... Q : Open door 1................................... BOOL
Trevers3.X.................. COMMAND : Transition complete..................... BOOL
UP_Ready.................... Subprocess READY.................................. BOOL
UP_Run...................... Subprocess RUN.................................... BOOL
```

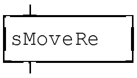
Fig. 8-55: Step 'sDoorRe'

```
sWait1

STEP sWait1: Waiting step for step sDoorCLS
| N |   | aWait |  (*Transition generation for waiting steps*)

ACTION aWait: Transition generation for waiting steps
1     TRUE                                                                    tWait.X
      -| |--------------------------------------------------------------------( )--
     tWait.X...................... COMMAND : Transition complete...................... BOOL
```

Fig. 8-56: Step 'sWait1'

```
sProcWrk

STEP sProcWrk: Machine workpiece (Start subprocess)
B D | T#30s | »rk.T_max  |  (*Workpiece machining*)
| N |       | aMonError  |  (*Workpiece machining*)
| N |       | aMonWork   |  (*Monitoring for workpiece machining*)
| N |       | aProcWrk   |  (*Workpiece machining*)

ACTION aMonWork: Monitoring for workpiece machining
1     (*********  Step-related monitoring error in automatic mode *********)
      (*The following condition(s) is(are) incorrect:*)
                                                                               _Action.
      I_Clamp     _SFC1.Auto                                                   MonError
      -|/|--------| |------------------------------------------------------------(S)--
      I_PosIn  |
      -|/|-----|
      I_Door1c |
      -|/|-----|
     I_Clamp...................... LmSw: Workpiece is clamped......................... BOOL
     _SFC1.Auto................... STATUS : SFC mode auto............................ BOOL
     _Action.MonError............. COMMAND : monitoring failure....................... BOOL
     I_PosIn...................... LmSw: Workpiece in machining position............. BOOL
     I_Door1c..................... LmSw: Door 1 is closed............................ BOOL

ACTION aProcWrk
1     (*Issuing start of machining*)
      _SFC1.                                                                   _TMP.
      Manual     I_Door1c   I_PosIn    I_Clamp    UP_Run                       Manual
      -| |--------| |--------| |--------| |--------| |--------------------------( )--
2                Action
                 ISFBxBxx
      _SFC1.Auto                                                               UP_Start
      -| |--------In              Out-------------------------------------------( )--
      _TMP.    |
      Manual   |
      -| |-----|
        aProcWrk-ActName
3     (*Transition generation stop machining normally*)
      UP_Finish                                                                tWorkEnd.X
      -| |---------------------------------------------------------------------( )--

     _SFC1.Auto................... STATUS : SFC mode auto............................ BOOL
     _SFC1.Manual................. STATUS : SFC mode manual.......................... BOOL
     _TMP.Manual.................. temp. flag for manual rung......................... BOOL
     I_Clamp...................... LmSw: Workpiece is clamped......................... BOOL
     I_Door1c..................... LmSw: Door 1 is closed............................ BOOL
     I_PosIn...................... LmSw: Workpiece in machining position............. BOOL
     TWorkEnd.X................... COMMAND : Transition complete...................... BOOL
     UP_Finish.................... Subprocess processing COMPLETED.................... BOOL
     UP_Run....................... Subprocess RUN..................................... BOOL
     UP_Start..................... Subprocess START................................... BOOL
```

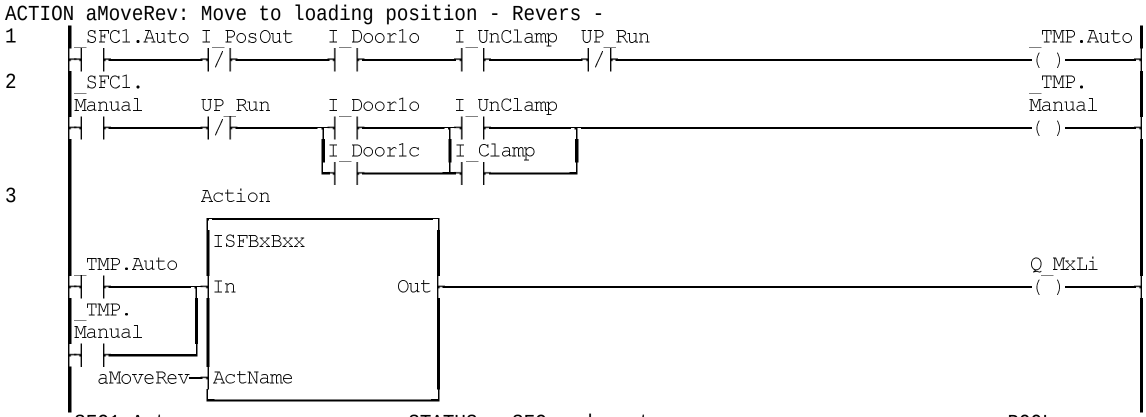
Fig. 8-57: Step 'sProcWrk'



STEP sMoveRe: Move to loading position - Home position -

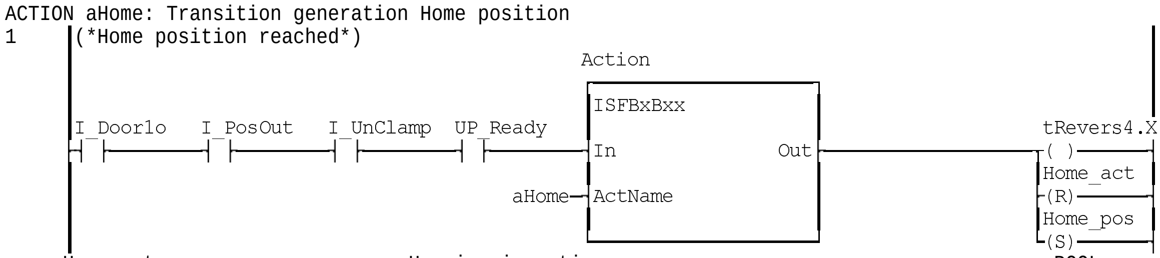
N		aMonError	(*Move to loading position – Revers -*)
N		aMoveRev	(*Move to loading position – Revers -*)
N		aHome	(*Transition generation Home position*)

ACTION aMoveRev: Move to loading position - Revers -



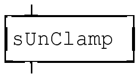

```
_SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
_SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
_TMP.Auto................... temp. flag for automatic rung..................... BOOL
_TMP.Manual................. temp. flag for manual rung........................ BOOL
I_Clamp..................... LmSw: Workpiece is clamped........................ BOOL
I_Door1c.................... LmSw: Door 1 is closed........................... BOOL
I_Door1o.................... LmSw: Door 1 is open............................. BOOL
I_PosOut.................... LmSw: Workpiece in transfer position.............. BOOL
I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
Q_MxLi...................... Q : Move out workpiece............................ BOOL
UP_Run...................... Subprocess RUN................................... BOOL
```

ACTION aHome: Transition generation Home position




```
Home_act.................... Homeing is active.................................... BOOL
Home_pos.................... Homeing reached...................................... BOOL
I_Door1o.................... LmSw: Door 1 is open................................. BOOL
I_PosOut.................... LmSw: Workpiece in transfer position.................. BOOL
I_UnClamp................... LmSw: Workpiece is not clamped........................ BOOL
TRevers4.X.................. COMMAND : Transition complete......................... BOOL
UP_Ready.................... Subprocess READY..................................... BOOL
```

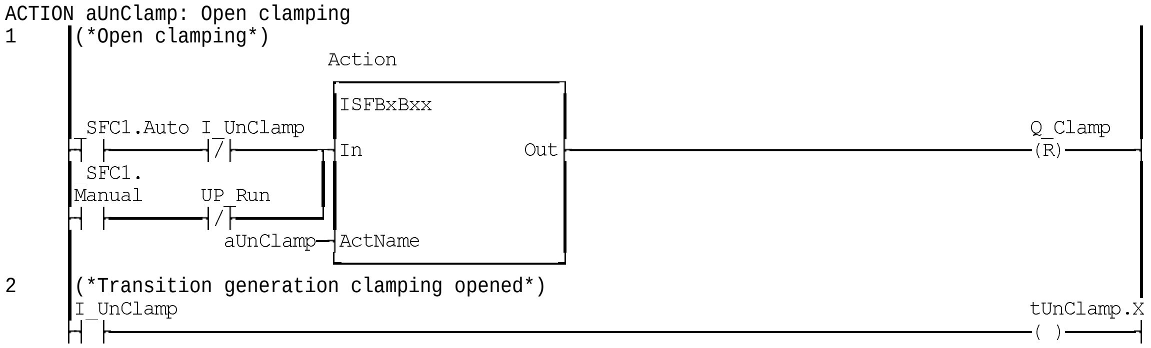
Fig. 8-58: Step 'sMoveRe'




```
STEP sUnClamp: Open clamping
+--------------------------------------+
B D   ¦ T#2s           ¦      »mp.T_max  ¦(*Open clamping*)
+-----+----------------+----------------¦
¦ N   ¦                ¦      aMonError  ¦(*Open clamping*)
+-----+----------------+----------------¦
¦ N   ¦                ¦      aUnClamp   ¦(*Open clamping*)
+--------------------------------------+
```

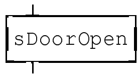
B D	T#2s	»mp.T_max	(*Open clamping*)
N		aMonError	(*Open clamping*)
N		aUnClamp	(*Open clamping*)

ACTION aUnClamp: Open clamping




```
_SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
_SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
Action...................... Function block for IndraStep actiones............. ISFBxBxx
AUnClamp.................... Open clamping.................................... ISTYxExx
I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
Q_Clamp..................... Q : Clamp workpiece............................... BOOL
TUnClamp.X.................. COMMAND : Transition complete...................... BOOL
UP_Run...................... Subprocess RUN.................................... BOOL
```

Fig. 8-59: Step 'sUnClamp'

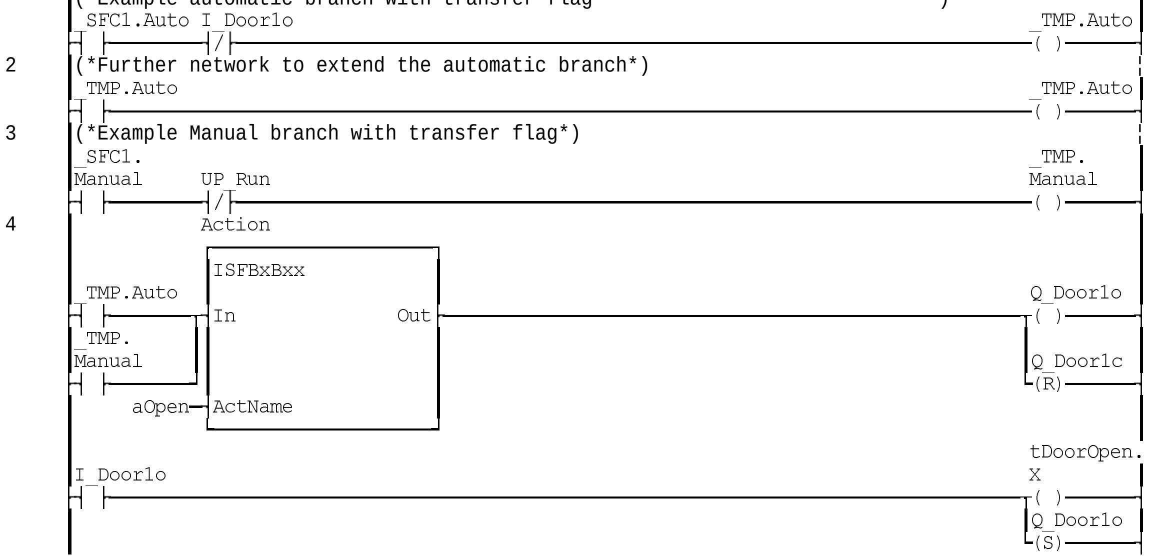


STEP sDoorOpen: Open door

B D	T#5s	»en.T_max	(*Maximum time monitoring Open door*)
N		aMonError	(*Monitoring - Limit switch pair monitoring*)
N		aOpen	(*Open door*)

ACTION aOpen: Open door

```
1  (***************************************************************************)
   (*In order to avoid exceeding the maximum width of representation of the  *)
   (*ladder diagram it is possible to use temporary flags. To avoid that     *)
   (*the FB Action is also switched in the intermediate flags,               *)
   (*no intervention takes place in case of variables starting with _TMP     *)
   (*formed as a simple network result. The structure _TMP (Typ ISTYxXxx)    *)
   (*is an example structure and can be modified, if necessary.              *)
   (*To allow an operating mode independent difference between the network branches *)
   (*In order differentiate the network branches operating mode dependently  *)
   (*the step sequence data structure _SFC1 flags are integrated for auto,   *)
   (*auto-single step mode (_SFC1.Auto;_SFC1.AutoTipp), manual (_SFC1.Manual)*)
   (*and for errors discrimination (_SFC1.SumError; _SFC1.TmrError).         *)
   (***************************************************************************)
   (*As long as the limit switch open (I_Door1o) is not approached -         *)
   (*The travel command Valve Open door (Q_Door1o) is issued.                *)
   (*Example automatic branch with transfer flag                             *)
```




```
_SFC1.Auto.................. STATUS : SFC mode auto........................... BOOL
_SFC1.Manual................ STATUS : SFC mode manual......................... BOOL
_TMP.Auto................... temp. flag for automatic rung..................... BOOL
_TMP.Manual................. temp. flag for manual rung........................ BOOL
I_Door1o.................... LmSw: Door 1 is open............................. BOOL
Q_Door1c.................... Q : Close door 1................................. BOOL
Q_Door1o.................... Q : Open door 1.................................. BOOL
TDoorOpen.X................. COMMAND : Transition complete..................... BOOL
UP_Run...................... Subprocess RUN................................... BOOL
```

Fig. 8-60: Step 'sDoorOpen'

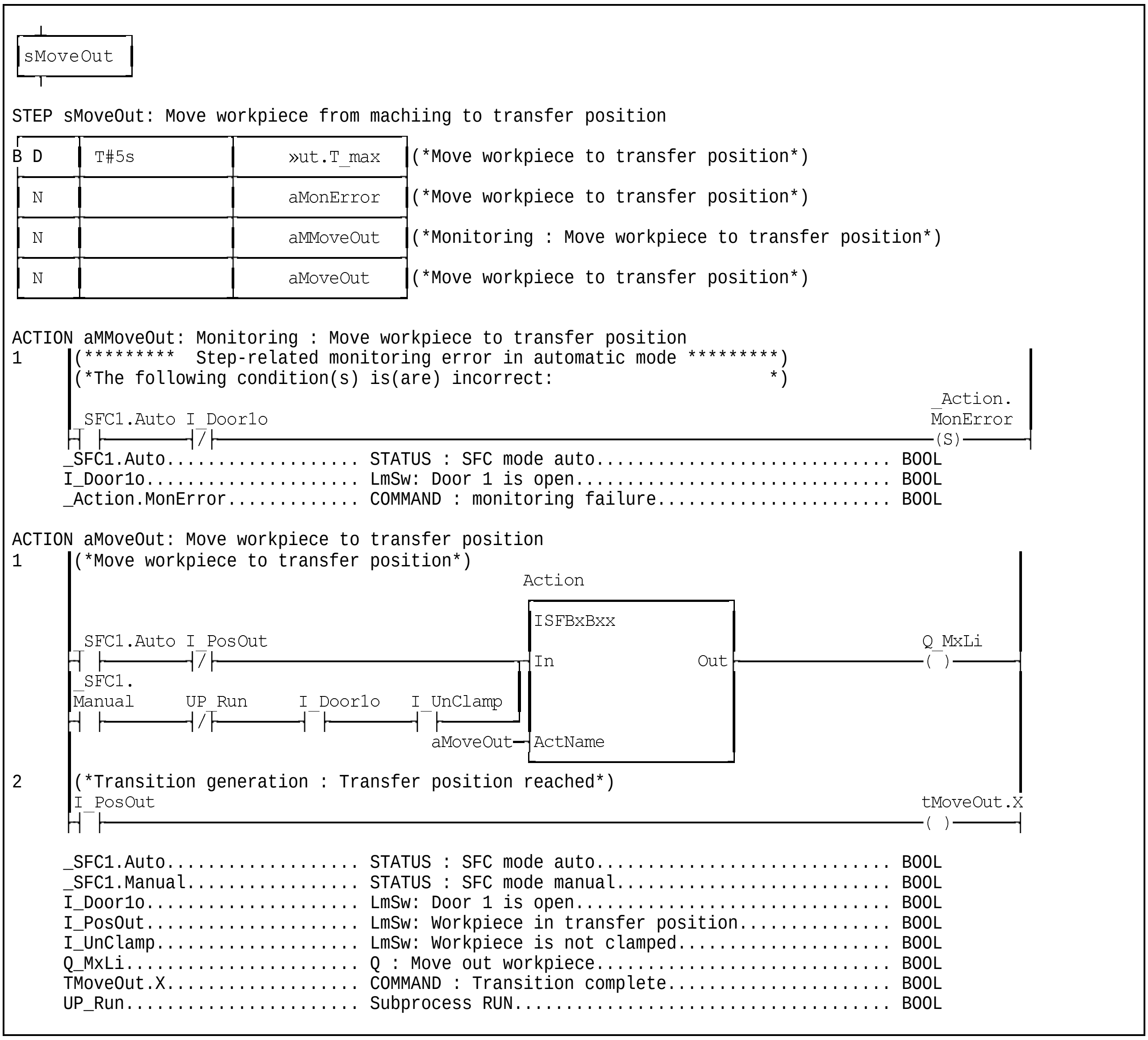


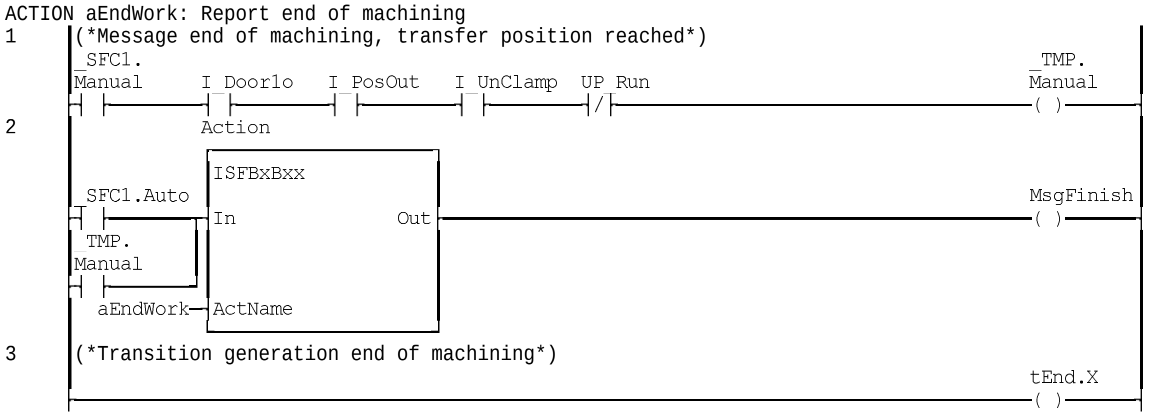
Fig. 8-61: Step 'sMoveOut'

sEndWork

STEP sEndWork: Report end of machining generate ready message

B L	T#500ms	»rk.T_min	(*Report end of machining*)
N		aEndWork	(*Report end of machining*)

ACTION aEndWork: Report end of machining




```
_SFC1.Auto.................. STATUS : SFC mode auto............................ BOOL
_SFC1.Manual................ STATUS : SFC mode manual.......................... BOOL
_TMP.Manual................. temp. flag for manual rung........................ BOOL
I_Door1o.................... LmSw: Door 1 is open.............................. BOOL
I_PosOut.................... LmSw: Workpiece in transfer position.............. BOOL
I_UnClamp................... LmSw: Workpiece is not clamped.................... BOOL
MsgFinish................... Workpiece machined at transfer position ........... BOOL
TEnd.X...................... COMMAND : Transition complete..................... BOOL
UP_Run...................... Subprocess RUN.................................... BOOL
```

Fig. 8-62: Step 'sEndWork'

9 Index

C

D

E

F

G

I

L

M

N

O

P

Q

R

S

Rexroth Indramat

T

U

10 List of figures

11 Service & Support

11.1 Helpdesk

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main steht Ihnen mit Rat und Tat zur Seite. Sie erreichen uns

- Telefonisch: **+49 (0) 9352 40 50 60**
  über Service-Call Entry Center Mo-Fr 07:00-18:00
- per Fax: **+49 (0) 9352 40 49 41**
- per e-Mail: **service@indramat.de**

Our service helpdesk at our headquarters in Lohr am Main, Germany can assist you in all kinds of inquiries. Contact us

- by phone: **+49 (0) 9352 40 50 60**
  via Service-Call Entry Center Mo-Fr 07:00 am -6:00 pm
- by fax: **+49 (0) 9352 40 49 41**
- by e-mail: **service@indramat.de**

11.2 Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service direkt ansprechbar unter

+49 (0) 171 333 88 26
oder **+49 (0) 172 660 04 06**

After helpdesk hours, contact our service department directly at

+49 (0) 171 333 88 26
or **+49 (0) 172 660 04 06**

11.3 Internet

Weitere Hinweise zu Service, Reparatur und Training finden Sie im Internet unter

www.indramat.de

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit Ihrem lokalen Ansprechpartner auf. Die Adressen sind im Anhang aufgeführt.

Additional notes about service, repairs and training are available on the Internet at

www.indramat.de

Please contact the sales & service offices in your area first. Refer to the addresses on the following pages.

11.4 Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

detaillierte Beschreibung der Störung und der Umstände.

Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.

Tel.-/Faxnummern und e-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind.

For quick and efficient help, please have the following information ready:

1. Detailed description of the failure and circumstances.
2. Information on the type plate of the affected products, especially type codes and serial numbers.
3. Your phone/fax numbers and e-mail address, so we can contact you in case of questions.

11.5 Kundenbetreuungsstellen - Sales & Service Facilities

Verkaufsniederlassungen	sales agencies
Niederlassungen mit Kundendienst	offices providing service

Deutschland – Germany

vom Ausland: (0) nach Landeskennziffer weglassen!
from abroad: don't dial (0) after country code!

Vertriebsgebiet Mitte Germany Centre Rexroth Indramat GmbH Bgm.-Dr.-Nebel-Str. 2 97816 Lohr am Main **Kompetenz-Zentrum Europa** Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885	SERVICE **CALL ENTRY CENTER** **MO – FR** **von 07:00 - 18:00 Uhr** **from 7 am – 6 pm** **Tel. +49 (0) 9352 40 50 60** service@indramat.de	SERVICE **HOTLINE** **MO – FR** **von 17:00 - 07:00 Uhr** **from 5 pm - 7 am** + SA / SO **Tel.: +49 (0)172 660 04 06** **oder / or** **Tel.: +49 (0)171 333 88 26**	SERVICE **ERSATZTEILE / SPARES** verlängerte Ansprechzeit - extended office time - ♦ nur an Werktagen - only on working days - ♦ von 07:00 - 18:00 Uhr - from 7 am - 6 pm - **Tel. +49 (0) 9352 40 42 22**
Vertriebsgebiet Süd Germany South Rexroth Indramat GmbH Ridlerstraße 75 80339 München Tel.: +49 (0)89 540138-30 Fax: +49 (0)89 540138-10 indramat.mue@t-online.de	Gebiet Südwest Germany South-West Mannesmann Rexroth AG Vertrieb Deutschland – VD-BI Geschäftsbereich Rexroth Indramat Regionalzentrum Südwest Ringstrasse 70 / Postfach 1144 70736 Fellbach / 70701 Fellbach Tel.: +49 (0)711 57 61–100 Fax: +49 (0)711 57 61–125	Vertriebsgebiet Ost Germany East Rexroth Indramat GmbH Beckerstraße 31 09120 Chemnitz Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333	Vertriebsgebiet Nord Germany North Mannesmann Rexroth AG Vertriebsniederlassung Region Nord Gesch.ber. Rexroth Indramat Walsroder Str. 93 30853 Langenhagen Tel.: +49 (0) 511 72 66 57-0 Fax: +49 (0) 511 72 66 57-95
Vertriebsgebiet West Germany West Mannesmann Rexroth AG Vertrieb Deutschland Regionalzentrum West Borsigstrasse 15 40880 Ratingen Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406	Vertriebsgebiet Mitte Germany Centre Mannesmann Rexroth AG Gesch.ber. Rexroth Indramat Lilistraße 14-18 63067 Offenbach Tel.: +49 (0) 69 82 00 90-0 Fax: +49 (0) 69 82 00 90-80	Vertriebsgebiet Ost Germany East Mannesmann Rexroth AG GB Rexroth Indramat GmbH Holzhäuser Str. 122 04299 Leipzig Tel.: +49 (0)341 86 77-0 Fax: +49 (0)341 86 77-219	Vertriebsgebiet Nord Germany North Rexroth Indramat GmbH Kieler Straße 212 22525 Hamburg Tel.: +49 (0) 40 81 955 966 Fax: +49 (0) 40 85 418 978

Europa – Europe

vom Ausland: (0) nach Landeskennziffer weglassen, from abroad: don't dial (0) after country code,

Italien: 0 nach Landeskennziffer mitwählen Italy: dial 0 after country code

Austria - Österreich	Austria - Österreich	Belgium - Belgien	Denmark - Dänemark
Mannesmann Rexroth Ges.m.b.H. Gesch.ber. Rexroth Indramat Hägelingasse 3 1140 Wien Tel.: +43 (0)1 9852540-400 Fax: +43 (0)1 9852540-93	Mannesmann Rexroth G.m.b.H. Gesch.ber. Rexroth Indramat Industriepark 18 4061 Pasching Tel.: +43 (0)7221 605-0 Fax: +43 (0)7221 605-21	Mannesmann Rexroth N.V.-S.A. Gesch.ber. Rexroth Indramat Industrielaan 8 1740 Ternat Tel.: +32 (0)2 5830719 Fax: +32 (0)2 5830731 indramat@rexroth.be	BEC AS Zinkvej 6 8900 Randers Tel.: +45 (0)87 11 90 60 Fax: +45 (0)87 11 90 61
Czech Republic - Tschechien	England	Finland - Finnland	France - Frankreich
Mannesmann-Rexroth, spol.s.r.o. Hviezdoslavova 5 627 00 Brno Tel.: +420 (0)5 48 126 358 Fax: +420 (0)5 48 126 112	Mannesmann Rexroth Ltd. Rexroth Indramat Division Broadway Lane, South Cerney Cirencester, Glos GL7 5UH Tel.: +44 (0)1285 863000 Fax: +44 (0)1285 863030	Rexroth Mecman Oy Rexroth Indramat division Ansatie 6 017 40 Vantaa Tel.: +358 (0)9 84 91-11 Fax: +358 (0)9 84 91-13 60	Mannesmann Rexroth S.A. Division Rexroth Indramat Parc des Barbanniers 4, Place du Village 92632 Gennevilliers Cedex Tel.: +33 (0)141 47 54 30 Fax: +33 (0)147 94 69 41 Hotline: +33 (0)608 33 43 28
France - Frankreich	France - Frankreich	Hungary - Ungarn	Italy - Italien
Mannesmann Rexroth S.A. Division Rexroth Indramat 270, Avenue de Lardenne 31100 Toulouse Tel.: +33 (0)5 61 49 95 19 Fax: +33 (0)5 61 31 00 41	Mannesmann Rexroth S.A. Division Rexroth Indramat 91, Bd. Irène Joliot-Curie 69634 Vénissieux – Cedex Tel.: +33 (0)4 78 78 53 65 Fax: +33 (0)4 78 78 53 62	Mannesmann Rexroth Kft. Angol utca 34 1149 Budapest Tel.: +36 (1) 364 00 02 Fax: +36 (1) 383 19 80	Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Via G. Di Vittoria, 1 20063 Cernusco S/N.MI Tel.: +39 02 2 365 270 Fax: +39 02 700 408 252378
Italy - Italien	Italy - Italien	Italy - Italien	Italy - Italien
Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Via Borgomanero, 11 10145 Torino Tel.: +39 011 7 50 38 11 Fax: +39 011 7 71 01 90	Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Via del Progresso, 16 (Zona Ind.) 35020 Padova Tel.: +39 049 8 70 13 70 Fax: +39 049 8 70 13 77	Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Via Mascia, 1 80053 Castellamare di Stabia NA Tel.: +39 081 8 71 57 00 Fax: +39 081 8 71 68 85	Mannesmann Rexroth S.p.A. Divisione Rexroth Indramat Viale Oriani, 38/A 40137 Bologna Tel.: +39 051 34 14 14 Fax: +39 051 34 14 22
Netherlands - Niederlande/Holland	Netherlands - Niederlande/Holland	Norway - Norwegen	Poland - Polen
Rexroth B.V. Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0)411 65 19 51 Fax: +31 (0)411 65 14 83 indramat@hydraudyne.nl	Rexroth Hydrocare B.V. Kruisbroeksestraat 1 (P.O. Box 32) 5281 RV Boxtel Tel.: +31 (0)411 65 19 51 Fax: +31 (0)411 67 78 14	Rexroth Mecman AS Rexroth Indramat Division Berghagan 1 or: Box 3007 1405 Ski-Langhus 1402 Ski Tel.: +47 (0)64 86 41 00 Fax: +47 (0)64 86 90 62	Mannesmann Rexroth Sp.zo.o. Biuro Poznan ul. Dabrowskiego 81/85 60-529 Poznan Tel.: +48 061 847 67 99 Fax: +48 061 847 64 02
Rumania - Rumänien	Russia - Russland	Spain - Spanien	Spain - Spanien
Mannesmann Rexroth Sp.zo.o. Str. Drobety nr. 4-10, app. 14 70258 Bucuresti, Sector 2 Tel.: +40 (0)1 210 48 25 +40 (0)1 210 29 50 Fax: +40 (0)1 210 29 52	Tschudnenko E.B. Arsenia 22 153000 Ivanovo Tel.: +7 093 223 96 33 oder/or +7 093 223 95 48 Fax: +7 093 223 46 01	Mannesmann Rexroth S.A. Divisiòn Rexroth Indramat Centro Industrial Santiga Obradors s/n 08130 Santa Perpetua de Mogoda Barcelona Tel.: +34 9 37 47 94 00 Fax: +34 9 37 47 94 01	Goimendi S.A. División Rexroth Indramat Parque Empresarial Zuatzu C/ Francisco Montagne no.2 20018 San Sebastian Tel.: +34 9 43 31 84 21 - service: +34 9 43 31 84 56 Fax: +34 9 43 31 84 27 - service: +34 9 43 31 84 60 satindramat-goimendi@adegi.es
Sweden - Schweden	Slowenia - Slowenien	Switzerland East - Schweiz Ost	Switzerland West - Schweiz West
Rexroth Mecman Svenska AB Rexroth Indramat Division Varuvägen 7 125 81 Stockholm Tel.: +46 (0)8 727 92 00 Fax: +46 (0)8 647 32 77	Rexroth Indramat elektromotorji d.o.o. Otoki 21 64 228 Zelezniki Tel.: +386 64 61 73 32 Fax: +386 64 64 71 50	Mannesmann Rexroth Schweiz AG Gesch.ber. Rexroth Indramat Gewerbestraße 3 8500 Frauenfeld Tel.: +41 (0)52 720 21 00 Fax: +41 (0)52 720 21 11	Mannesmann Rexroth Suisse SA Département Rexroth Indramat Rue du village 1 1020 Renens Tel.: +41 (0)21 632 84 20 Fax: +41 (0)21 632 84 21
Turkey - Türkei			
Mannesmann Rexroth Hidropar A..S. Fevzi Cakmak Cad No. 3 34630 Sefaköy Istanbul Tel.: +90 212 541 60 70 Fax: +90 212 599 34 07			

Rexroth Indramat

Africa, Asia, Australia – incl. Pacific Rim

vom Ausland: (x) nach Landeskennziffer weglassen!
from abroad: don't dial (x) after country code!

Australia - Australien AIMS - Australian Industrial Machinery Services Pty. Ltd. Unit 3/45 Horne ST Campbellfield , VIC 3061 Melbourne Tel.: +61 (0)3 93 59 02 28 Fax: +61 (0)3 93 59 02 86	Australia - Australien Mannesmann Rexroth Pty. Ltd. No. 7, Endeavour Way Braeside Victoria, 31 95 Melbourne Tel.: +61 (0)3 95 80 39 33 Fax: +61 (0)3 95 80 17 33 mel@rexroth.com.au	China Shanghai Mannesmann Rexroth Hydraulics & Automation Ltd. Wai Gaoqiao Free Trade Zone No.122, Fu Te Dong Yi Road Shanghai 200131 - P.R.China Tel.: +86 21 58 66 30 30 Fax: +86 21 58 66 55 23	China Mannesmann Rexroth (China) Ldt. 15/F China World Trade Center 1, Jianguomenwai Avenue Beijing 100004, P.R.China Tel.: +86 10 65 05 03 80 Fax: +86 10 65 05 03 79
China Mannesmann Rexroth (China) Ldt. A-5F., 123 Lian Shan Street Sha He Kou District Dalian 116 023, P.R.China Tel.: +86 411 46 78 930 Fax: +86 411 46 78 932	China Mannesmann Rexroth (China) Ldt. Guangzhou Repres. Office Room 1014-1016, Metro Plaza, Tian He District, 183 Tian He Bei Rd Guangzhou 510075, P.R.China Tel.: +86 20 8755-0030 +86 20 8755-0011 Fax: +86 20 8755-2387	Hongkong Rexroth (China) Ldt. 1/F., 19 Cheung Shun Street Cheung Sha Wan, Kowloon, Hongkong Tel.: +852 22 62 51 00 Fax: +852 27 41 33 44	India - Indien Mannesmann Rexroth (India) Ltd. Rexroth Indramat Division Plot. A-58, TTC Industrial Area Thane Turbhe Midc Road Mahape Village Navi Mumbai - 400 701 Tel.: +91 (0)22 7 61 46 22 Fax: +91 (0)22 7 68 15 31
India - Indien Mannesmann Rexroth (India) Ltd. Rexroth Indramat Division Plot. 96, Phase III Peenya Industrial Area Bangalore - 560058 Tel.: +91 (0)80 8 39 73 74 Fax: +91 (0)80 8 39 43 45	Indonesia - Indonesien PT. Rexroth Wijayakusuma Jl. Raya Bekasi Km 21 Pulogadung Jakarta Timur 13920 Tel.: +62 21 4 61 04 87 +62 21 4 61 04 88 Fax: +62 21 4 60 01 52	Japan Rexroth Automation Co., Ltd. Service Center Japan Yutakagaoka 1810, Meito-ku, NAGOYA 465-0035, Japan Tel.: +81 (0)52 777 88 41 +81 (0)52 777 88 53 +81 (0)52 777 88 79 Fax: +81 (0)52 777 89 01	Japan Rexroth Automation Co., Ltd. Rexroth Indramat Division 1F, I.R. Building Nakamachidai 4-26-44, Tsuzuki-ku YOKOHAMA 224-0041, Japan Tel.: +81 (0)45 942 72 10 Fax: +81 (0)45 942 03 41
Korea Mannesmann Rexroth-Korea Ltd. Rexroth Indramat Division 1500-12 Dadae-Dong- Saha-Ku Pusan, 604-050 Republic of South Korea Tel.: +82 (0)51 26 00 741 Fax: +82 (0)51 26 00 747 gyhan@rexrothkorea.co.kr	Malaysia Mannesmann Rexroth SDN.BHD. Head Office No. 3, Block B, Jalan SS 13/5 Subang Jaya Industrial Estate 47500 Petaling Jaya - Selangor Tel.: +60 (0) 3 73 44 870 Fax: +60 (0) 3 73 44 864	South Africa - Südafrika TECTRA Automation (Pty) Ltd. 28 Banfield Road,Industria North RSA - Maraisburg 1700 Tel.: +27 (0)11 673 20 80 Fax: +27 (0)11 673 72 69	Taiwan Rexroth Uchida Co., Ltd. No.17, Lane 136, Cheng Bei 1 Rd., Yungkang, Tainan Hsien Taiwan, R.O.C. Tel.: +886 (0)6 25 36 565 Fax: +886 (0)6 25 34 754
Thailand NC Advance Technologies Co. Ltd. 59/76 Moo 9 Soi Ramintra 34 Ramintra Road, Tharang, Bangkhen Bangkok 10220 Tel.: +66 2 943 70 62 +66 2 943 71 21 Fax: +66 2 509 23 62 sonkawin@hotmail.com			

Nordamerika – North America

USA Hauptniederlassung - Headquarters	USA Central Region - Mitte	USA Southeast Region - Südwest	USA SERVICE-HOTLINE
Mannesmann Rexroth Corporation Rexroth Indramat Division 5150 Prairie Stone Parkway Hoffman Estates, IL 60192-3707 **Competence Centre America** Tel.: +1 847 6 45 36 00 Fax: +1 847 6 45 62 01 service@indramat.com	Mannesmann Rexroth Corporation Rexroth Indramat Division Central Region Technical Center Auburn Hills, MI 48326 Tel.: +1 248 3 93 33 30 Fax: +1 248 3 93 29 06	Mannesmann Rexroth Corporation Rexroth Indramat Division Southeastern Technical Center 3625 Swiftwater Park Drive Suwanee, Georgia 30174 Tel.: +1 770 9 32 32 00 Fax: +1 770 9 32 19 03	- 7 days x 24hrs - **+1-800-860-1055**
USA Northeast Region – Nordost	**USA Northeast Region – Nordost**	**Canada East - Kanada Ost**	**Canada West - Kanada West**
Mannesmann Rexroth Corporation Rexroth Indramat Division Charlotte Regional Sales Office 14001 South Lakes Drive Charlotte, North Carolina 28273 Tel.: +1 704 5 83 97 62 +1 704 5 83 14 86	Mannesmann Rexroth Corporation Rexroth Indramat Division Northeastern Technical Center 99 Rainbow Road East Granby, Connecticut 06026 Tel.: +1 860 8 44 83 77 Fax: +1 860 8 44 85 95	Basic Technologies Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Tel.: +1 905 335 55 11 Fax: +1 905 335-41 84	Basic Automation Burnaby 5345 Goring St. Burnaby, British Columbia Canada V7J 1R1 Tel. +1 604 205-5777 Fax +1 604 205-6944 dave.gunby@basic.ca

Südamerika – South America

Argentina - Argentinien	Argentina - Argentinien	Brazil - Brasilien	Brazil - Brasilien
Mannesmann Rexroth S.A.I.C. Division Rexroth Indramat Acassusso 48 41/7 RA - 1605 Munro (Buenos Aires) Tel.: +54 (0)11 4756 01 40 Fax: +54 (0)11 4762 6862 mannesmann@mannesmannsaic.com.ar	NAKASE Servicio Tecnico CNC Calle 49, No. 5764/66 RA - 1653 Villa Balester Prov. - Buenos Aires Tel.: +54 (0) 11 4768 36 43 Fax: +54 (0) 11 4768 24 13 nakase@usa.net nakase@infovia.com.ar	Mannesmann Rexroth Automação Ltda. Divisão Rexroth Indramat Rua Georg Rexroth, 609 Vila Padre Anchieta BR - 09951-270 Diadema-SP [Caixa Postal 377] [BR-09901-970 Diadema-SP] Tel.: +55 (0)11 4075 90 60 +55 (0)11 4075 90 70 Fax: +55 (0)11 4075 90 50 awittwer@rexroth.com.br	Mannesmann Rexroth Automação Ltda. Divisão Rexroth Indramat R. Dr.Humberto Pinheiro Vieira, 100 Distrito Industrial BR - 89220-390 Joinville - SC [Caixa Postal 1273] Tel./Fax: +55 (0)47 473 58 33 Mobil: +55 (0)47 974 66 45 prochnow@zaz.com.br
Mexico			
Mannesmann Rexroth Mexico S.A. de C.V. Calle Neptuno 72 Unidad Ind. Vallejo MEX - 07700 Mexico, D.F. Tel.: +52 5 754 17 11 +52 5 754 36 84 +52 5 754 12 60 Fax: +52 5 754 50 73 +52 5 752 59 43			

Notizen - Notes

Printed in Germany