Project 2

Sequential control for a coin plant
with the language SIMATIC S7-GRAPH

List of contents

1 Objectives of the project

2 Introduction

2.1 Short description and characterization of a sequential control

2.2 Overview of the programming language S7-GRAPH for Programmable Logic Controllers (PLC) S7-300/400

2.2.1 Blocks of the sequential control

2.2.2 Elements of a sequencer

- Initial step
- Linear succession
- Alternative branch
- Simultaneous branch
- Jump to a step

3 Hard- and Software

4 Procedure for setting up and programming a sequential control with S7-GRAPH

5 Problem description of a coin plant

5.1 Technology schematics

5.2 Problem description

6 Assignment list for a coin plant

7 Tasks during the execution of the project

8 Appendix:

C1: Non-event-controlled actions

C2: Actions with a step becoming active / being deactivated

1 Objectives of the project

Main objective: Mode of operation and programming of a Programmable Logic Controller: sequential control

Learning objectives:

- Characteristics of a sequential control (synchronous, asynchronous)
- Becoming acquainted with available function blocks (structured programming)
- Statement of step enabling conditions
- Creating a sequential function chart in conformity with DIN 40719, part 6
- Selection of the necessary hardware units
- Programming
- Simulation
- Starting-up procedure, testing

2 Introduction

2.1 Short description and characterization of a sequential control

Controlling sequences of motions, e.g. of a machine or installation, the chronological order is of decisive importance, as the next motion may only be started after the preceding one has been finished. Such problems with an inevitable step-by-step run can be solved by sequential controls. The progression from one step to the following one happens according to step enabling conditions. Those either depend on check-back signals or are only time-oriented. As sequential controls are - as logic controls - binary controls, the same symbols are used for the description of the functions (see figure 1). Supplementary there is only a simplified description of the sequence steps.
The smallest functional unit of the sequential control is the progression to next step. Several steps in succession make a sequencer. The steps are set one behind the other by suitable conditions. Each step outputs one or more commands, resets the preceding step and prepares the next step to be set.

Process-oriented stepping-conditions depend on check-back-signals which signalize a certain process status and in most cases the execution of previously given commands.
Time-oriented stepping-conditions depend only on time-conditions, i.e. waiting time. They are applied when detecting of a status, i.e. a check-back-signal, is technologically hard to realize or impossible.

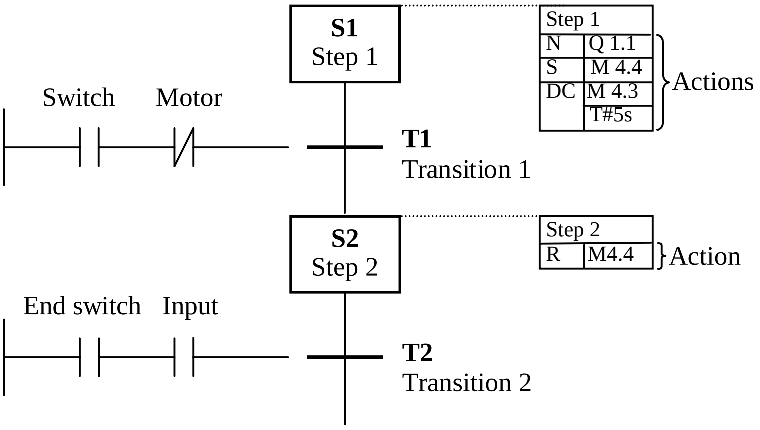


Figure 1: Example: Sequential control system with steps and transitions

2.2 Overwiew of the S7-GRAPH programming language for Programmable Logic Controllers (PLC) S7-300/400

The S7-GRAPH programming language extends the range of functions of STEP 7 by allowing you to program sequential controls graphically.
With S7-GRAPH you can configure and write programs to control sequential processes with a SIMATIC Programmable Logic Controller (PLC).
S7-GRAPH for the S7-300/400 complies with the sequential controlled language "Sequential Function Chart" specified in the DIN EN 61131-3 (IEC 1131-3) standard.

S7-GRAPH allows a description of the system by dividing the structure into

- steps with actions and
- transitions (step enabling conditions).

With S7-GRAPH a sequential control can be programmed containing up to eight sequencers.

A sequencer can consist of:

- upstream permanent instructions
- steps with actions controlling the outputs of S7-300/400 or calling the STEP7 code-blocks (FC)
- transitions containing the step enabling conditions for the next step
- branches, jumps, sequence end
- downstream permanent instructions

2.2.1 Blocks of the sequential control

A runnable program consists at least of:

- an organization block (OB) calling the S7-GRAPH-FB block.
  The OB is programmed with STEP-7, e.g. in KOP, TUP or AWL
- a function block (FB) containing the sequencer
- an instance-data-block (DB) with the data for the sequencer

The initialization data are read in during the data block generation.

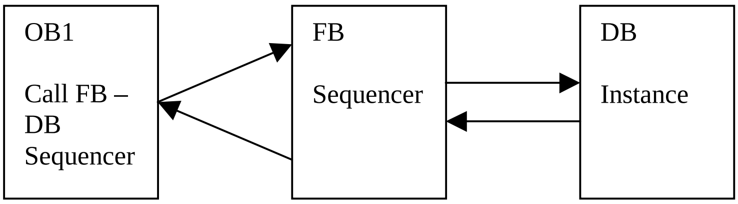


Figure 2: Minimum block structure of a sequencer

S7-GRAPH allows selecting the stepping property from one of four possible modes:

- Automatic mode:
  In the automatic mode, the next step is enabled when a transition is satisfied.
- Manual mode:
  In contrast to the automatic mode, in the manual mode the next step is not enabled when the transition is satisfied. The steps are selected and deselected manually.
- Inching mode step to next:
  The inching mode corresponds to the automatic mode with an additional step enabling condition. Not only must the transition be satisfied, but there must also be a rising edge at the T_PUSH parameter before control passes to the next step.
- Automatic or step to next:
  In the "automatic or step-by-step" mode, control is passed to the next step when the transition is satisfied or when there is a rising edge at the T_PUSH parameter.

For selecting the mode of the sequential control, the standard parameter entry of S7-GRAPH-FB is necessary, as shown in figure 3.

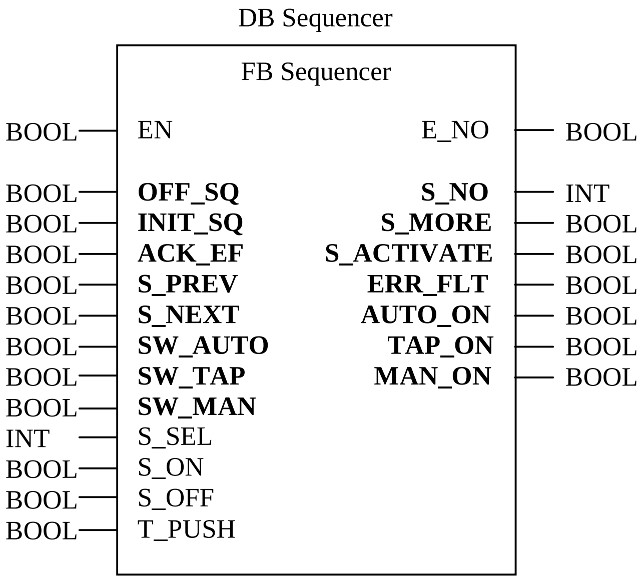


Figure 3: FB-call with standard parameter set

Input parameters of the S7-GRAPH-FB

The FB reacts to the rising edge of the input parameter (exception: EN)

Parameter	Data Type	Description
EN	BOOL	Controls execution of the FB (enable input). If EN is not connected, the FB is always executed.
OFF_SQ	BOOL	OFF_SEQUENCE: Sequencer off, in other words deactivate all steps
INIT_SQ	BOOL	INIT_SEQUENCE: Activate initial steps (reset sequencer)
ACK_EF	BOOL	ACKNOWLEDGE_ERROR_FAULT: Acknowledgement of a disturbance, force switching to next step
S_PREV	BOOL	PREVIOS_STEP: Automatic mode: Pages back through the currently active steps. The step number is indicated in S_NO. Manual mode: Indicate previous step (next lower number) in S_NO.
S_NEXT	BOOL	NEXT_STEP: Automatic mode: Page forwards through the currently active steps. The step number is indicated in S_NO. Manual mode: Indicates the number of the next step (next higher number) in S_NO.
SW_AUTO	BOOL	SWITCH_MODE_AUTOMATIC: Mode change to automatic mode
SW_TAP	BOOL	SWITCH_MODE_TRANSITION_AND_PUSH: Mode change to Inching mode („semi-automatic“)
SW_TOP	BOOL	SWITCH_MODE_TRANSITION_OR_PUSH: Mode change to automatic or switch to next
SW_MAN	BOOL	SWITCH_MODE_MANUAL: Mode change to manual mode (automatic execution is not triggered)
S_SEL	INT	STEP_SELECT: Selects a specific step for the output parameter S_NO. Activate/deactivate in the manual mode with S_ON, S_OFF.
S_ON	BOOL	STEP_ON: Manual mode: Activate the displayed step.
T_PUSH	BOOL	PUSH_TRANSITION: Transition switches when the condition is satisfied and T_PUSH (edge). Requirement: Inching (SW_TAP) or automatic or step-by-step (SW-TOP) mode.

Output parameters of the S7-GRAPH-FB

Parameter	Data Type	Description
ENO	BOOL	Enable output. When the FB is active and no error has occured, ENO has the value 1, otherwise 0.
S_NO	INT	STEP_NUMBER: Display step number
S_MORE	BOOL	MORE_STEPS: Other steps are active
S_ACTIVE	BOOL	STEP_ACTIVE: Displayed step is active
ERR_FLT	BOOL	IL_ERROR_OR_SV_FAULT: Group disturbance
AUTO_ON	BOOL	AUTOMATIC_IS_ON: Indicates the automatic mode
TAP_ON	BOOL	T_AND_PUSH_IS_ON: Indicates the inching mode
TOP_ON	BOOL	T_OR_PUSH_IS_ON: Display SW_TOP mode
MAN_ON	BOOL	MANUAL_IS_ON: Indicates the manual mode

A sequencer can either be processed cyclically, i.e. by jumping from the sequence end to the sequence start or it can be executed only once, stopping at the sequence end.

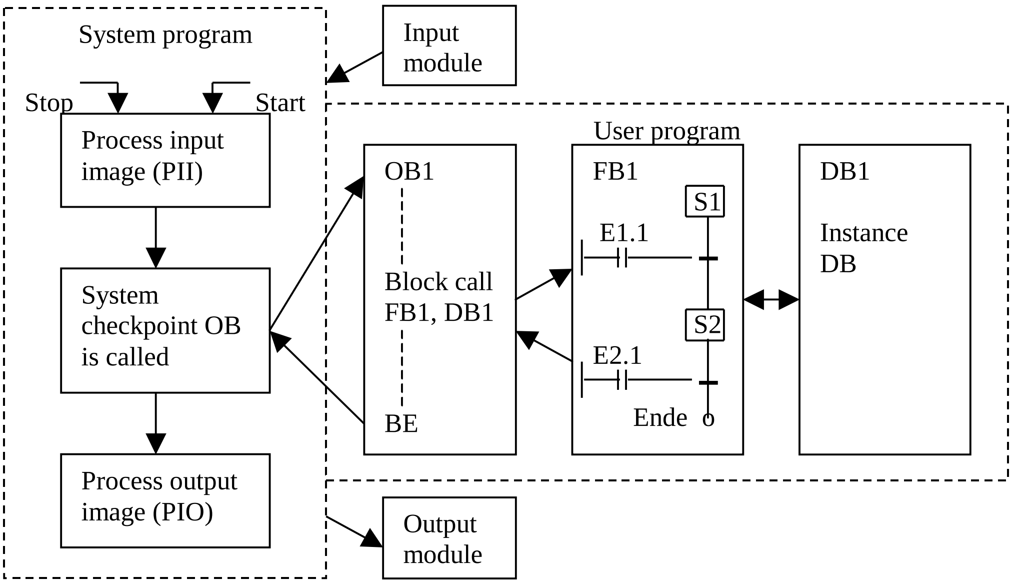
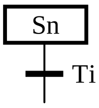


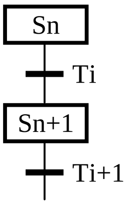
Figure 4: Example for a GRAPH program flow

2.2.2 Elements of the sequencer

- Initial step
  Every sequencer starts with an initial step, which becomes active if the S7-GRAPH-FB parameter INIT_SQ = 1 or if the conditions of the previous transition are satisfied.

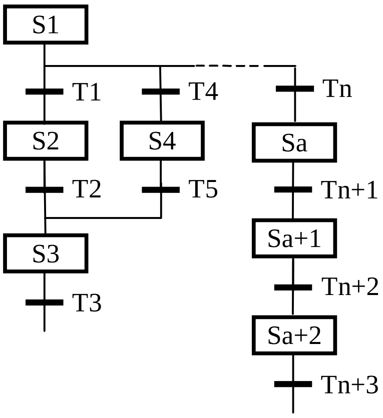


- Linear succession of the sequencer
  A linear sequencer consists of a succession of steps and transitions. If the sequencer is closed by a sequence end, the sequencer is executed only once.

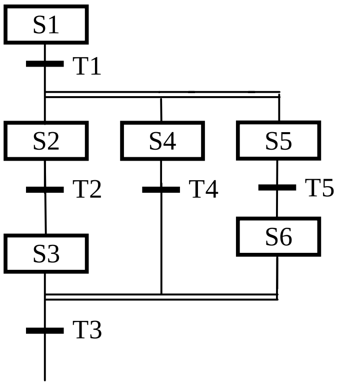


- Alternative Branch
  An alternative branch consists of more than one parallel path. Each path in an alternative branch begins with a transition. Only the branch path whose transition switches first is executed. An alternative branch therefore corresponds to an OR operation in which only one path can be active.
  Each path in an alternative branch ends with a transition and can be closed by a branch stop or a jump.

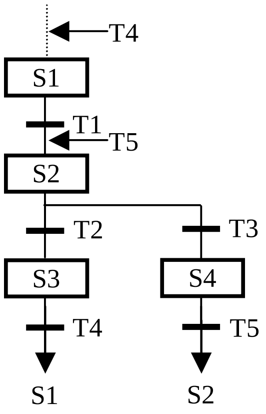
  If more than one transition is satisfied at the beginning of various paths, the transition furthest to the left, in other words the transition located immediately below the previous step, has highest priority. The priority of all other paths of the alternative branch is decided by the transition numbers. In this case, the transition with the lowest number has the highest priority.



- Simultaneous branch
  A simultaneous branch consists of more than one parallel path each of which starts with a step. The paths are executed simultaneously. A simultaneous branch corresponds to an AND branch.
  The transition before the simultaneous branch activates the first steps of the individual simultaneous branch paths. Every path in a simultaneous branch ends with a step and is completed by a successor transition.



- Jump to a step
  A jump is a transition to a step without graphical link. A jump can occur within a sequencer or to another sequencer in the same FB.



3 Hard- and Software

The execution of the sequential control requires the following hard- and software components:

a) Programmable Logic Controller S7-300 with components:

- mounting channel as a mounting rack	
- Power supply module PS 3075A	(6ES7-307-1EA00-0AA0)
- Central processing unit CPU 315-2DP	(6ES7-315-2AF03-0AB0)
- Digital input module 16xDC 24V	(6ES7-321-1BH01-0AA0)
- Digital output module 16xDC 24V, 0.5A	(6ES7-322-1BH01-0AA0)

b) MPI interface cable for the connection CPU / PC

c) PC with a minimum configuration: Pentium processor, Windows 95/98/NT, 32MB RAM, hard disc 3 GB, CD-ROM drive, colour display

d) Software package SIMATIC STEP 7 with GRAPH 7

4 Procedure for setting up and programming a sequential control system with S7-GRAPH

(🖱 = left mouse click)

SIMATIC Manager - Start
Wizard - Cancel
Menu field: File → New → 🖱
Name (Project) enter „**OK**“
Menu field: Insert → Station → SIMATIC 300 🖱
→ ⊞ - Project-Name 🖱
└─⊞ SIMATIC 300 → Hardware 🖱🖱
↑ 🖱

<u>Hardware configuration</u>
Toolbar: Catalog 🖱
→ ⊞ - SIMATIC 300 🖱
→ ⊞ - RACK 300 🖱
→ └──[] - Rail 🖱🖱
→ ⊞ - PS300 🖱
→ └──[] - PS3075A 🖱🖱 (6ES7 307-1EA00-0AA0)
→ ⊞ - CPU 300 🖱
→ └──[] - CPU 315-2DP 🖱
→ └─⊞ - 6ES7 315-1AF03-0AB0 🖱
→ └──[] V1.1 🖱🖱
Properties-Profibus interface DP master (RO/2.1)$^{+}$ „**OK**“
Insert (2775:790) **„No“**
Insert (13:4242) **„No“**
Column 3 in the configuration table should be left empty, column 4 clicked
→ ⊞ - SM 300 🖱
→ ⊞ - DI 300 🖱
└──[] - SM 321 DI 16xDC 24V 🖱🖱 (6ES7 321-1BH01-0AA0)
→ ⊞ - DO 300 🖱
└──[] - SM322 DO 16xDC24V/0,5A 🖱🖱 (6ES7 322-1BH01-0AA0)
Configuration Save: → Station → Save 🖱
Configuration Exit: → Station → Exit 🖱

In Project → ⊞ - SIMATIC 300 🖱
→ └─⊞ - CPU315-2DP 🖱
→ └─⊞ - S7 Program 🖱
├── Sources
→ └── Block 🖱🖱
Menu field: Insert → S7 Block → Organization block Menu field: 🖱
Properties-Function block General-Part 1:
Name: OB1
Created in Language: LAD „**OK**“
Menu field: Insert → S7 Block → Funktion block 🖱
Properties-Function block General-Part 1:
Name: FB1
Created in Language: GRAPH „**OK**“

In Project -□- FB1
↑🖱🖱

Programming Sequential Control Systems
By clicking on □ (Step + Transition) and other symbols, a sequencer is built up.
Insert a leap by:
- Menu field: insert a leap 🖱
  - click to an end of transition T9 with a mouse
    and click at the end of branch (S1).

When a sequencer is constructed, all actions and transitions should be entered
e.g. S1 - Step1 clicked → action described then „OK“
→ File → Save 🖱
In project: -□- OB1 → Toolbar → Program elements → FB 🖱 → FB1 🖱
↑🖱🖱

Assignment of the FB sequencer
→ File → Save 🖱

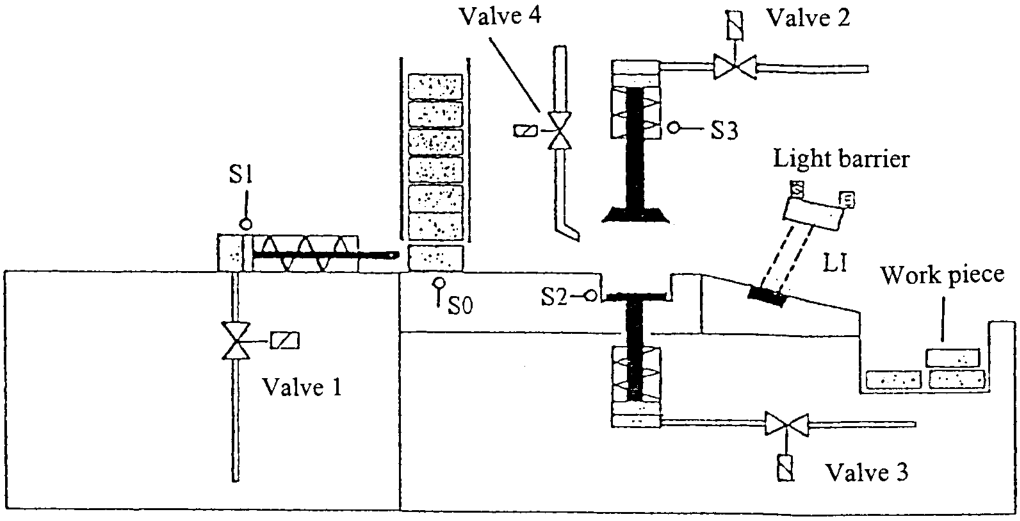
OB1, FB1 and all other blocks with Ctrl 🖱
Menu field: View → Load (mode On-line) 🖱
Toolbar: Simulation 🖱

<u>Start the simulation</u>: S7-PLCSIM
Toolbar: Insert input, Output, Bit memory, Timer, Counter 🖱
→ Input, Bit memory, Timer, put down
CPU 300/400 → RUN 🖱
Simulation performed.

For the simulation the time is given in minutes, in the sequence of the chain it can be watched better.

5 Problem description of a coin plant

5.1 Technology schematics



5.2 Problem description

The cylinder of valve 1 pushes the work piece from the magazin (position switch S0 is operated) to the stamping place and moves back when position switch S2 is not operated. Afterwards the cylinder of the coinstamp should extend (valve 2) until position switch S3 is operated. When the position switch operates, the cylinder should run in again.
Further the work piece should be raised up by the cylinder of valve 3, S2=0.
In 3 seconds valve 4 should open to blow an airstream at the coin and move it over the slope to the catching reservoir. On the way the coin passes a light barrier LI, which switches off valves 3 and 4.
The procedure can repeat itself after a 4 second pause, when the precondition for the home position is fulfilled (end switch S0, S1 and LI are closed, S2 and S3 are opened).

6 Assignment list for a coin plant

Symbol	Inputs	Comment
S0	I 0.0	Position switch, workpiece is under the magazine
S1	I 0.1	Position switch, cylinder OF valve 1 is not driven out
S2	I 0.2	Position switch, workpiece is in the coinposition
S3	I 0.3	Position switch, Coin stamp is driven out
LI	I 0.4	Photo sensor

Inputs for the setup of the FB1 block

I 1.0	Sensing device, sequence consists are switched off- all steps are deactivated (Off_SQ)
I 1.1	Sensing device, sequence consists are initialized (INT_SQ)
I 1.2	Sensing device, trouble receipted, refresh forced (ACK_EF)
I 1.3	Switch, operating mode automatic/manual (SW_AUTO/SW_MAN)
I 1.4	Switch, inching mode (SW_TAP)
I 1.5	Sensing device, transition switched, when condition fulfilled and T_PUSH (edge) Precondition: inching mode (SW_TAP)

Outputs

Q 4.1	Valve 1 (slide cylinder)
Q 4.2	Valve 2 (coinage cylinder)
Q 4.3	Valve 3 (raise cylinder)
Q 4.4	Valve 4 (airstream)
Q 4.5	Valve 3 (edge for WinCC)
Q 5.0	Signal- FB active (EDO)
Q 5:1	Signal- automatic mode (AUTO_ON)
Q 5.2	Signal- manual mode (MAN_ON)
Q 5.3	Signal- inching mode (TAP_ON)
Q 5.7	Normal state of the system

Memory bits

M 50.0	Waiting time 3 sec. (T#3s), time delay for valve 4
M 50.1	Waiting time 4 sec. (T#4s), time delay for repeating the sequence cascade
MW1	Signal, step number (S_NO)

Program structure

FB1 – GRAPH function block (sequence cascade)
OB1 – Organization block

Literature

Handbook for SIMATIC Software STEP 7 by Siemens.

7 Tasks during the execution of the project

- Structuring the sequencer
  (Dividing the embossing process into steps with corresponding actions, stepping conditions – transitions)
- Programming the sequencer according to the defined procedure
- Executing the simulation on the PC
- Loading the program into the PLC, testing mode

8 Appendix

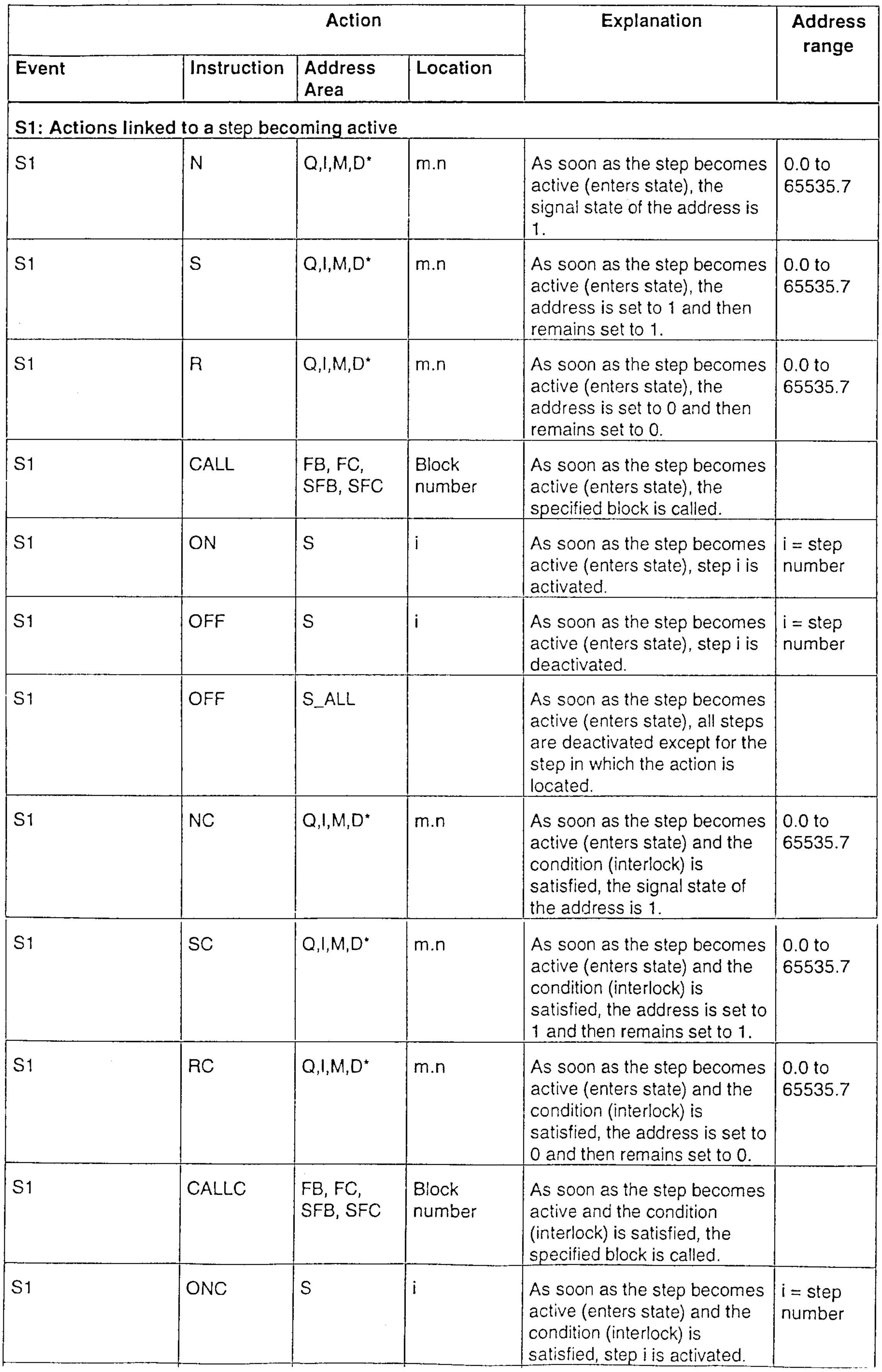
C1: Non-event-controlled-action

Action					Explanation	Address range
Event	In-struction	Address Area	Location	Time Constant		
	N	Q,I,M,D*	m.n		As long as the step is active, the signal state of the address is 1.	0.0 to 65535.7
	S	Q,I,M,D*	m.n		As long as the step is active, the address is set to 1 and then remains set to 1.	0.0 to 65535.7
	R	Q,I,M,D*	m.n		As long as the step is active, the address is set to 0 and then remains set to 0.	0.0 to 65535.7
	D	Q,I,M,D*	m.n	T#<const>	n seconds after step activation the signal state of the address is 1 for the duration of the step activation. This does not apply if the step is active for a time shorter than n seconds.	0.0 to 65535.7
	L	Q,I,M,D*	m.n	T#<const>	If the step is active, the address has signal state 1 for n seconds.	0.0 to 65535.7
	CALL	FB, FC, SFB, SFC	Block number		As long as the step is active, the specified block is called.	
	NC	Q,I,M,D*	m.n		As long as the step is active and the condition (interlock) is satisfied, the signal state of the address is 1.	0.0 to 65535.7
	SC	Q,I,M,D*	m.n		As long as the step is active and the condition (interlock) is satisfied, the address is set to 1 and then remains set to 1.	0.0 to 65535.7
	RC	Q,I,M,D*	m.n		As long as the step is active and the condition (interlock) is satisfied, the address is set to 0 and then remains set to 0.	0.0 to 65535.7
	DC	Q,I,M,D*	m.n	T#<const>	n seconds after step activation and as long as the step is active and the condition (interlock) is satisfied, the signal state of the address is 1. If the step is not active, the signal state of the address is 0.	0.0 to 65535.7
	LC	Q,I,M,D*	m.n	T#<const>	If the step is active and the condition (interlock) is satisfied, the address has signal state 1 for n seconds. If the step is not active, the signal state of the address is 0.	0.0 to 65535.7
	CALLC	FB, FC, SFB, SFC	Block number		As long as the step is active and the condition (interlock) is satisfied, the specified block is called.	

An instance DB is required with CALL[C] FB/SFB.

S7-GRAPH for S7 300/400 Programming Sequential Control Systems
C79000-G7076-C526-01

C2: Akion with arriving/leaving step



Action				Explanation	Address range
Event	**Instruction**	**Address Area**	**Location**		
S1: Actions linked to a step becoming active					
S1	N	Q,I,M,D*	m.n	As soon as the step becomes active (enters state), the signal state of the address is 1.	0.0 to 65535.7
S1	S	Q,I,M,D*	m.n	As soon as the step becomes active (enters state), the address is set to 1 and then remains set to 1.	0.0 to 65535.7
S1	R	Q,I,M,D*	m.n	As soon as the step becomes active (enters state), the address is set to 0 and then remains set to 0.	0.0 to 65535.7
S1	CALL	FB, FC, SFB, SFC	Block number	As soon as the step becomes active (enters state), the specified block is called.	
S1	ON	S	i	As soon as the step becomes active (enters state), step i is activated.	i = step number
S1	OFF	S	i	As soon as the step becomes active (enters state), step i is deactivated.	i = step number
S1	OFF	S_ALL		As soon as the step becomes active (enters state), all steps are deactivated except for the step in which the action is located.	
S1	NC	Q,I,M,D*	m.n	As soon as the step becomes active (enters state) and the condition (interlock) is satisfied, the signal state of the address is 1.	0.0 to 65535.7
S1	SC	Q,I,M,D*	m.n	As soon as the step becomes active (enters state) and the condition (interlock) is satisfied, the address is set to 1 and then remains set to 1.	0.0 to 65535.7
S1	RC	Q,I,M,D*	m.n	As soon as the step becomes active (enters state) and the condition (interlock) is satisfied, the address is set to 0 and then remains set to 0.	0.0 to 65535.7
S1	CALLC	FB, FC, SFB, SFC	Block number	As soon as the step becomes active and the condition (interlock) is satisfied, the specified block is called.	
S1	ONC	S	i	As soon as the step becomes active (enters state) and the condition (interlock) is satisfied, step i is activated.	i = step number

Action				Explanation	Address range
Event	Instruction	Address Area	Location		
S1	OFFC	S	i	As soon as the step becomes active (enters state) and the condition (interlock) is satisfied, step i is deactivated.	i = step number
S1	OFFC	S_ALL		As soon as the step becomes active (enters state) and the condition (interlock) is satisfied, all steps are deactivated except for the step in which the action is located.	
S0: Actions linked to a step being deactivated					
S0	N	Q,I,M,D*	m.n	As soon as the step is deactivated (leaves state), the signal state of the address is 1.	0.0 to 65535.7
S0	S	Q,I,M,D*	m.n	As soon as the step is deactivated (leaves state), the address is set to 1 and then remains set to 1.	0.0 to 65535.7
S0	R	Q,I,M,D*	m.n	As soon as the step is deactivated (leaves state), the address is set to 0 and then remains set to 0.	0.0 to 65535.7
S0	CALL	FB, FC, SFB, SFC	Block number	As soon as the step is deactivated (leaves state), the specified block is called.	
S0	ON	S	i	As soon as the step is deactivated (leaves state), step i is activated.	i = step number
S0	OFF	S	i	As soon as the step is deactivated (leaves state), step i is also deactivated.	i = step number