TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Hochschule
Zittau/Görlitz
UNIVERSITY OF APPLIED SCIENCES

CMS
Control of a Manipulator with SIMATIC S7

Osvald Modrlák, Lukáš Hubka
Petr Školník

Steffen Gärtner

Liberec 09/2012

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

Hochschule
Zittau/Görlitz
UNIVERSITY OF APPLIED SCIENCES

CMS

Control of a Manipulator with SIMATIC S7

Osvald Modrlák, Lukáš Hubka
Petr Školník

Steffen Gärtner

Liberec 09/2012

CONTENT

1 AIM OF THE EXPERIMENT

The aim of this experiment is to control a pneumatic-electric manipulator by the PLC SIMATIC S7-300. The control programs are programmed in the programming language STEP7 GRAPH and tested on the real manipulator. Fig. 1 shows the manipulator with a double magazine.

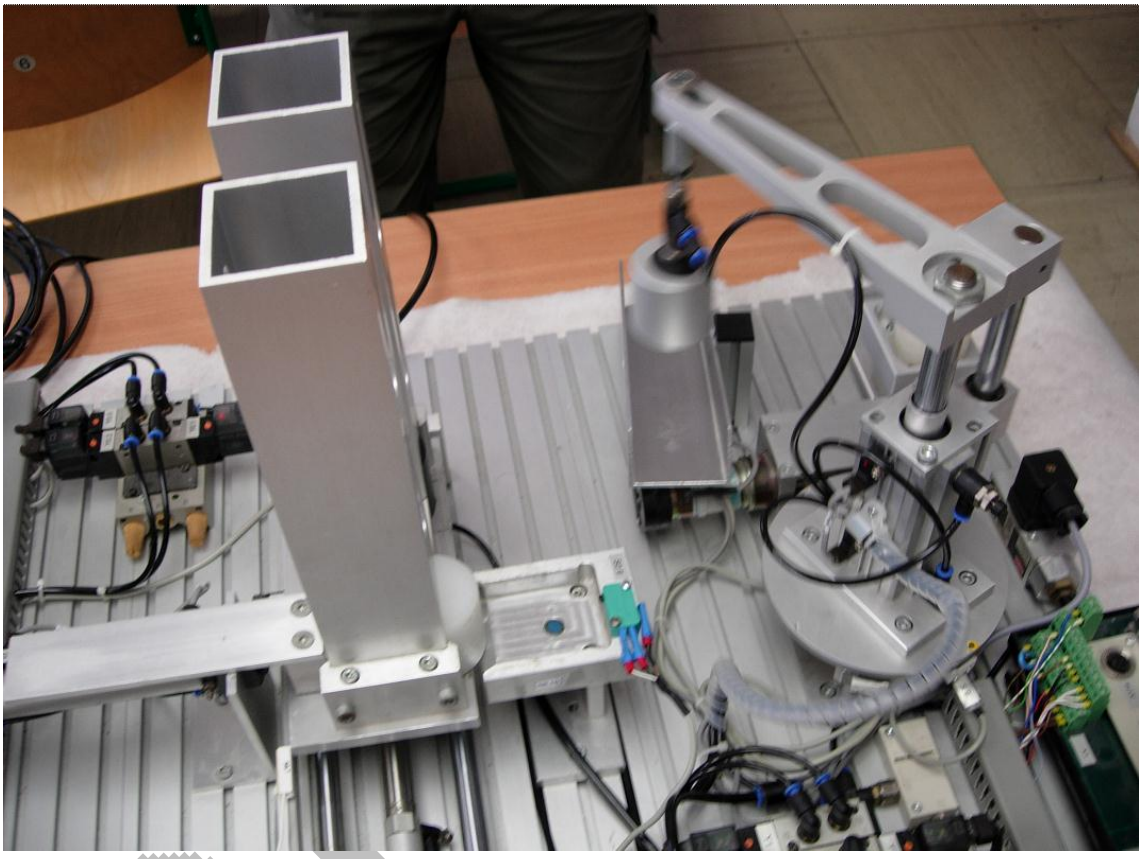


Fig. 1: Manipulator with double magazine

2 DESCRIPTION OF THE MANIPULATOR

The manipulator consists of a handling device (fig. 2) and the functional unit double magazine (fig. 3). The description of these components is done on the basis of the fig. 2 and fig. 3. (The buttons T2 and T3 are not shown.)

2.1 THE HANDLING DEVICE

With its sucker and arm the handling device (see fig. 2) is able to move objects both around and along the y-axis. The axis is the center of a turntable. The turntable is driven by a DC servo drive motor M0. The rotational position is freely selectable. The direction of rotation can be monitored by the rotation sensor S1. The rotation sensor S1 is not connected to the PLC. The following 3 positions of the turntable are detected by sensors

- basic position (barrel removal position/stack) sensor S8,
- output position 1 (dock) sensor S9,
- output position 2 (chute) sensor S10,

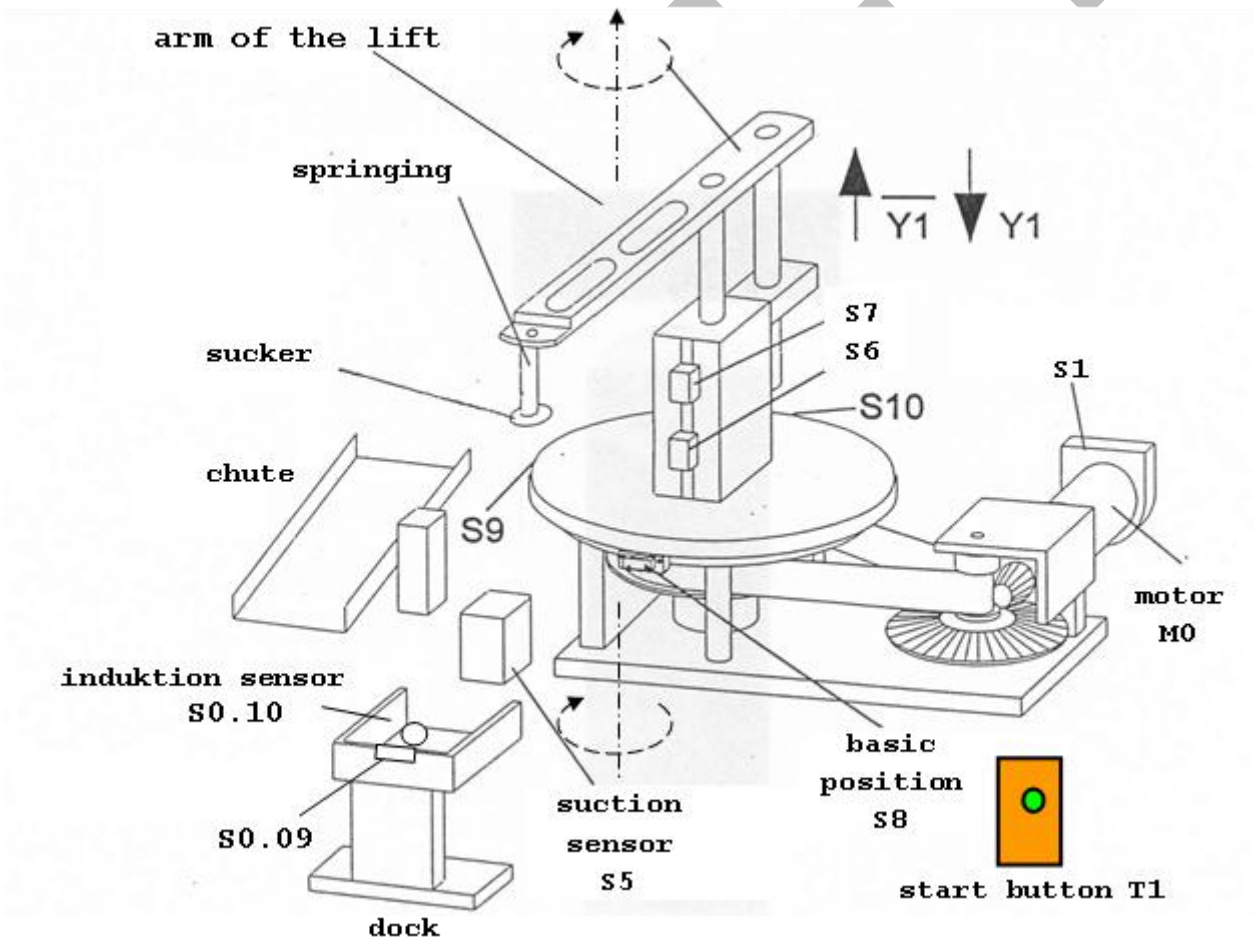


Fig. 2: Technological schema of the handling device

There is a suction cup (sucker) detached on the arm of a lift. The arm with the sucker is moved in vertical direction by a double-acting pneumatic lifting cylinder. The upper and lower arm end positions are detected by the micro-switches S6 and S7. During programming a certain time must be considered for performing of the following actions at the manipulator

- suction,
- lifting,
- rotation.

The description of the input and output signals is made according to the fig. 2 and fig. 3. The input signals provide information about the state of the individual units and elements of the manipulator. They are stored in the PLC input module (input register). The input signals are described in Tab. 1. Logical input signals are:

- start button signal T1,
- emergency stop button signal T2,
- auto/manual switch button signal T3,
- underpressure (sucker) monitoring sensor signal S5,
- signal of micro-switches on the lower and upper end position of the lifting cylinder signals S6 and S7,
- signals of micro-switches for the position of the turntable
  - basic position signal S8,
  - output position 1 (dock) signal S9,
  - output position 2 (chute) signal S10,
- sensor signal for the presence of a work piece in the removal position signal S0.9,
- induction sensor signal for the presence of a metallic barrel on the removal position signal S0.10.

The output signals control certain actions of the manipulator. They are described in tab. 2. Logical output signals are

- signal for the movement of the lifting cylinder signal Y1,
- signals for the activation and deactivation of the sucker signals Y2 and Y3,
- signals for the direction of rotation of the motor signals Y5 and Y6.

The actual input and output signals are processed as the logical input and output which are stored in the input and output module (input and output register) of the PLC system. Each input and output signal can be described by a variable name. For our task the input and output addresses of the input and output module (PLC registers) are determined by the wiring. In tab. 1 and tab. 2 there are shown the names and the descriptions of the signals, the input and output addresses of the input and output module (PLC-register) and the names of the variables.

Tab. 1: Input signals of the handling device

Label	Description of the signal	PLC address	Name of variable
S5	Work piece (barrel) on sucker *log 1 = work piece is on the suction cup* *log 0 = work piece is not present on sucker*	I125.2	iBarrelOnSucker
S6	Arm is down *log 1 = lift cylinder is at the lower end position* *log 0 = lift cylinder is not at the lower end position*	I124.5	iArmDown
S7	Arm is up *log 1 = lift cylinder is at the upper end position* *log 0 = lift cylinder is not at the upper end position*	I124.6	iArmUp
S8	Arm above stack *log 1 = turntable (arm) is in the home position (stack)* *log 0 = turntable is not in the home position*	I124.7	iArmStack
S9	Arm above chute *log 1 = arm is above the chute* *log 0 = arm is not above the chute*	I125.0	iArmChute
S10	Arm above output dock *log 1 = arm is above the dock* *log 0 = arm is not above the dock*	I125.7	iArmDock
S0.9	New barrel arrives to output dock (stack) *pulse signal* *log 1 = new work piece is at the removal position available*	I124.4	iBarrelDock Pulse
S0.10	Metal barrel at the dock *log 1 = metal work piece is at the removal position available* *log 0 = metal work piece is not at the removal position available*	I125.4	iMetalBarrel
T1	Start button *log 1 = start button is pressed* *log 0 = start button is not pressed*	I125.6	iStartButton
T3	Manual/auto switch *log 1 = auto mode* *log 0 = manual mode*	I125.1	iManAuto Switch

Tab. 2: Output signals of the handling device

Label	Description of the signal	PLC address	Name of variable
Y1	Move arm down *log 1 = the coil of the lifting cylinder is activated, the arm of the lifting cylinder moves into the lower end position where it stops* *log 0 = the coil of the lifting cylinder is switched off, the arm of the lifting cylinder drives due to the spring force in the upper end position where it stops*	Q124.3	oArmDown
Y2	Set sucker on *log 1 = Vacuum is on. The coil of the suction cup is active. This open a valve for vacuum and the suction of the suction cup is produced.*	Q124.4	oSuckerOn
Y3	Set sucker off *log 1 = Vacuum is off. Thus, the formation of vacuum valve is closed. The suction force of the suction cup is zero.* ***Attention! The vacuum valve must be closed before producing the suction force of the suction cup.***	Q124.5	oSuckerOff
Y5	Rotate arm to right *log 1 = the arm is rotated to the right* *log 0 = no signal to the rotating actuator*	Q124.6	oRotateRight
Y6	Rotate arm to left *log 1 = the arm is rotated to the left* *log 0 = no signal to the rotating actuator*	Q124.7	oRotateLeft
H2	Green light *log 1 = green lamp light on* *log 0 = green lamp does not light*	Q125.0	oGreenLight
H3	White light *log 1 = white lamp light on* *log 0 = white lamp does not light*	Q125.1	oWhiteLight

2.2 THE DOUBLE MAGAZINE (STACK)

The double magazine (Fig. 3) consists of two removable stacks (A, B). Filling them outside the facility is possible without stopping the machine operation. Cylinders are intended to be the transported work pieces (barrels). The stacks are manually filled with the work pieces. The stacks (A, B) are pneumatically moved to the removal position by the slider (feeder) M2. Reaching the removal position of the stacks is detected by the micro-switches S0.7 and S0.8. There the work pieces are moved by the pneumatically driven feeder M1 from the stack into the removal position (working position). The initial position of this feeder is detected by the micro-switch S0.5. Another micro-switch S0.6 indicates whether the feeder is in the correct removal position. Work pieces at the removal position are detected by means of the micro-switches S.09 or S0.10. A certain time has to be considered for moving the work piece and for displacing the stacks during programming. The input signals to the PLC registers are shown in Tab. 3. The output signals by which the feeders M1 and M2 are controlled are shown in Tab. 4. It should be noted that as soon as the output signal Y2.0 is set to logic 1, the insertion of the work piece by the feeder M1 begins. Setting the signal to log 0 starts the backspacing of the feeder M1.

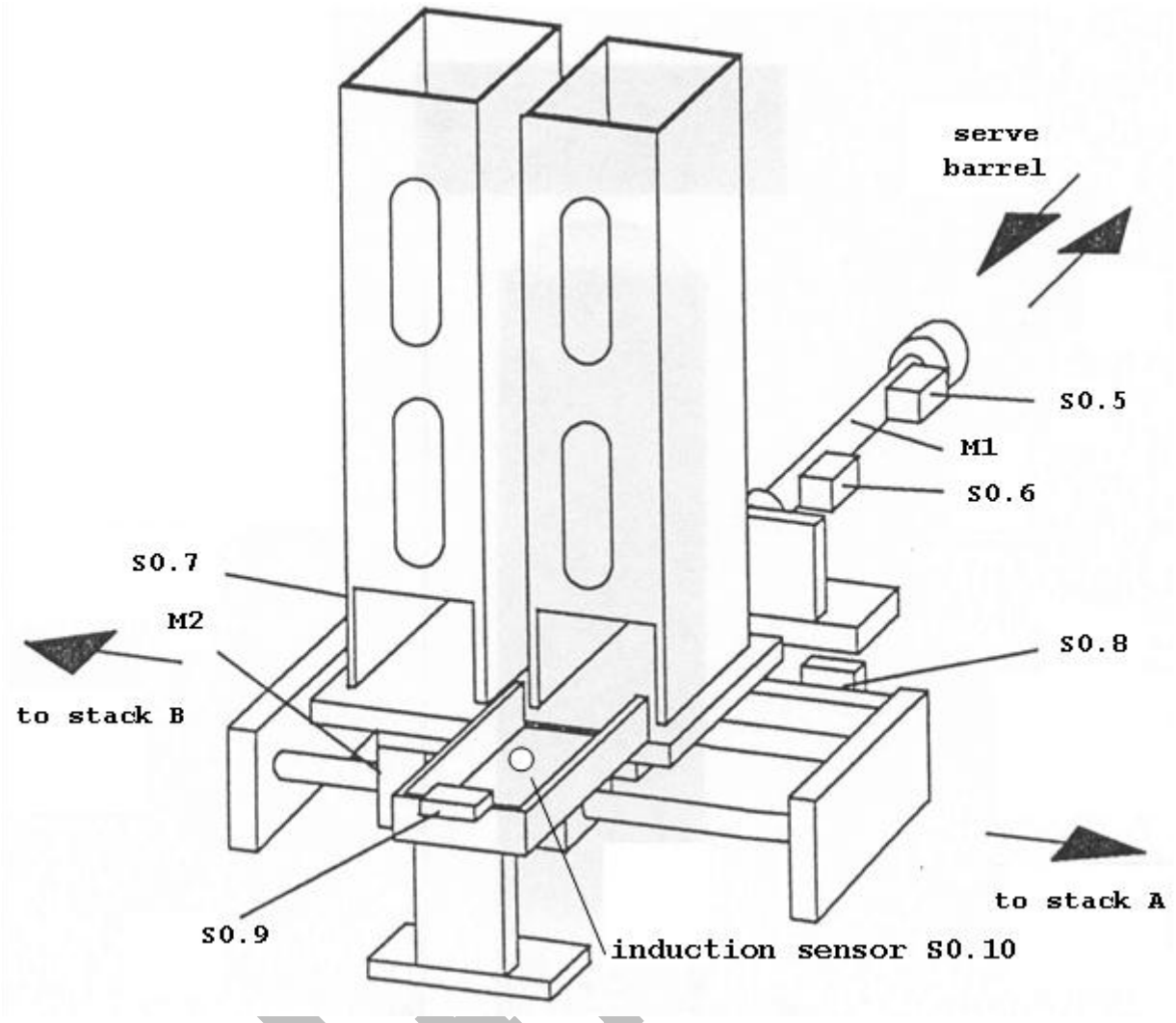


Fig. 3: Technological schema of the double magazine

The initial and end (removal) positions must be controlled. When the output signal Y2.1 is set to logic 1, the slider M2 starts the movement of the stack A in the removal position. When the output signal Y2.2 is set to logic 1, the movement of the stack B in the removal position starts. It is important to ensure that the removal position of the stacks is reached before an output signal is set.

Tab. 3: Input signals of the double magazine

Label	Description of the signal	PLC address	Name of variable
S0.5	Feeder M1 out of stack *log 1 = slider is in initial position (feeder M2 operation is possible)* *log 0 = slider is not in initial position (feeder M2 cannot operate)*	I124.0	iFeederOut
S0.6	Feeder M1 in stack *log 1 = slider is in end position (barrel is moved out of stack)* *log 0 = slider is not in end position*	I124.1	iFeederIn
S0.7	Feeder M2 in stack B position *log 1 = barrels can be served from stack B* *log 0 = stack B is not in the right position*	I124.3	iStackB
S0.8	Feeder M2 in stack A position *log 1 = barrels can be served from stack A* *log 0 = stack A is not in the right position*	I124.2	iStackA
S0.9	New barrel arrives to dock (stack) *pulse signal!* *log 1 = new work piece is at the removal position available*	I124.4	iBarrelDock Pulse
S0.10	Metal barrel at the dock *log 1 = metal work piece is at the removal position* *log 0 = metal work piece is not at the removal position*	I125.4	iMetalBarrel

Tab. 4: Output signals of the double magazine

Label	Description of the signal	PLC address	Name of variable
Y2.0	Feeder M1 serves barrel *log 1 = feeder serves barrel* *log 0 = feeder is going back to initial position*	Q124.0	oFeederServe
Y2.1	Feeder M2 is shifted to stack A *log 1 = slider M2 is shifting the stack A into serve position*	Q124.1	oStackA
Y2.2	Feeder M2 in shifted to stack B *log 1 = slider M2 is shifted the stack B into serve position*	Q124.2	oStackB

3 AUTOMATION SYSTEM SIMATIC S7-300/400

3.1 GENERAL DESCRIPTION

The SIMATIC S7-300 is a modularly designed programmable controller, which may consists of the following components:

- rack,
- power supply (PS),
- central processing unit (CPU),
- interface modules (IM),
- signal modules (SM),
- function modules (FM),
- communication module (CP)
- subnets.

An automation system (S7-300 station) may consist of several racks which are interconnected via bus cable. In the main rack there have to be the power supply, CPU and peripheral modules. Up to eight peripheral modules can be inserted in the central rack of the S7-300.

The used S7-300 station is set up in a simple configuration and is equipped with:

CPU 313C, V2 with digital input module DI24 and digital output module DO16, both on 24 V DC. Addresses of inputs are from the range I124.0 to I125.7, addresses of outputs are from the range Q124.0 to Q125.7.

Fig. 4 shows the used SIMATIC S7 station with CPU 313C-V2, rack, power supply, input and output modules.

Fig. 4: SIMATIC S7-300 – station of the experiment

3.2 ABSOLUTE AND SYMBOLIC ADDRESSES

An input/output signal is fed to an input/output module where it is connected to a particular terminal. This terminal has a peripheral address (absolute address). The CPU automatically copies the signal from the input/output module into the process image any time before the start of the program processing. The signal will be addressed as the operand 'input' in the input module or 'output' in the output module.

Example:

'New barrel arrive to dock' → on the absolute input address I124.4 is log 1.

You can enter a name for this input by allocating an alphanumerical symbol (symbolic address) to the absolute address in the table of symbols which corresponds to the meaning of this input signal.

Example:

'New barrel arrive to the dock' → 'iBarrelDockPulse'.

4 INTRODUCTION TO S7-GRAPH

Siemens serves several possibilities how to build up the program and which program tool to use. The base program package contains the programming languages 'Statement List' (STL), 'Ladder Diagram' (LD) and 'Function Block Diagram' (FBD). It also includes 'S7-SCL' (Structured Control Language), 'S7-GRAPH' (Sequencer chain planning) and 'S7-HiGraph' (state chart).

Programming the control of a pneumatic-electric manipulator shall serve as an introduction to the possibilities of SIMATIC S7-300. The emphasis is placed on the creation and testing of a control program as a sequence chain. The programing is focused only to S7-GRAPH.

4.1 S7-GRAPH STRUCTURE OF THE SEQUENCE CHAIN

Before programming in S7-GRAPH the task should be broken down into its individual steps in the design phase. Basis for the concept design are the technology images (fig. 2 and fig. 3), the input and output signals of the handling device (tab. 1 and tab. 2) and the input and output signals of the double magazine (tab. 3 and tab. 4). The operation of the manipulator is described by S7-GRAPH in the form of a sequence chain. A sequence chain can be defined as follows:

A sequence chain represents a sequence of single steps and conditions (transitions). The steps are executed in a fixed order and the transitions control how the process moves to the next single step (see fig. 5).

What are 'single step', 'active step' and 'transition'?

Single step: The control task is divided into single steps. The actions which must be performed by the controller in a particular state are described in steps.

Active step: Actions of this step are actually processed. A step is activated,

- **if the conditions of the preceding transition are fulfilled or**
- **if it is defined as an initial step and the sequencer was initialized or**
- **if it is called by an event-driven action.**

Transition: A set of conditions which must be fulfilled to change the actual state of the program to the new one.

Structure and processing of a sequence chain.

The processing of a sequence chain begins with an initial step or with several initial steps.

An active step is exited when all possible occurring disturbances are resolved and the transition to the following step is fulfilled. The next step following the fulfilled transition becomes active.

At the end of the sequence chain is

- **a jump to any step of the sequence chain (cyclic operation of the sequencer) or**
- **a chain end (program ends and is stopped).**

To determine the structure of the sequence chain, follow these steps.

1) Break down the manipulator operation process into steps and specify the order of the steps (fig. 5)!
2) Specify for each step which actions should be performed within it!
3) Specify for each step which conditions have to be met to jump into the next step!

An alternative branch (fig. 6) or jump (fig. 7) can also be used.

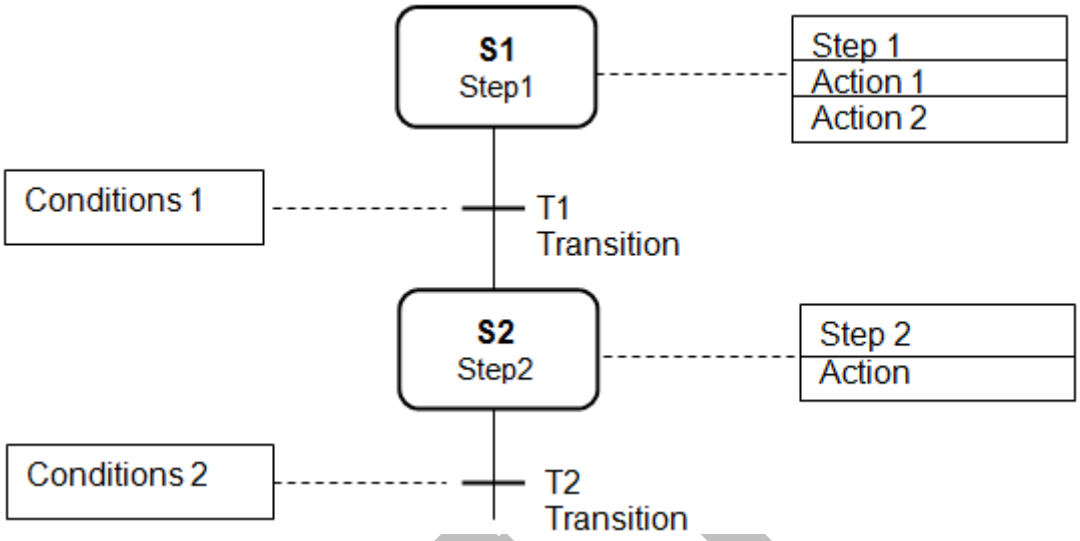


Fig. 5: The structure of the sequence chain in S7-GRAPH

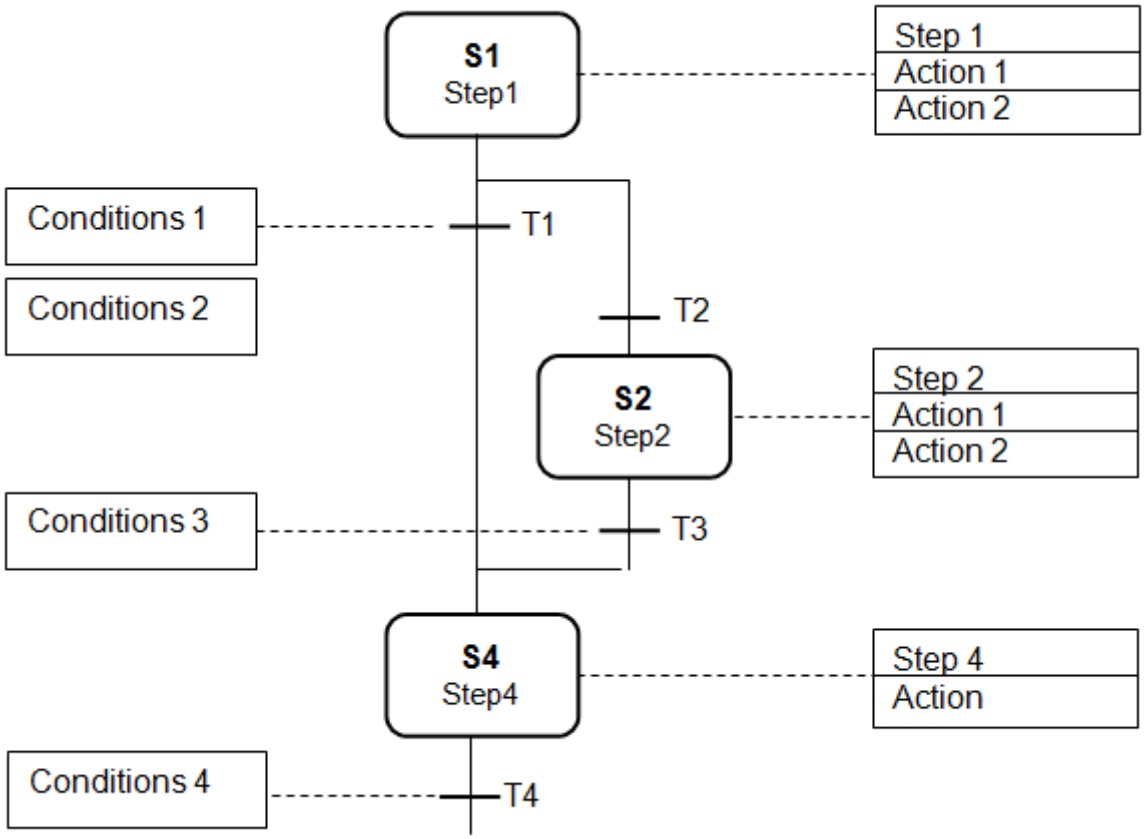


Fig. 6: S7-GPARH program structure with an alternative way

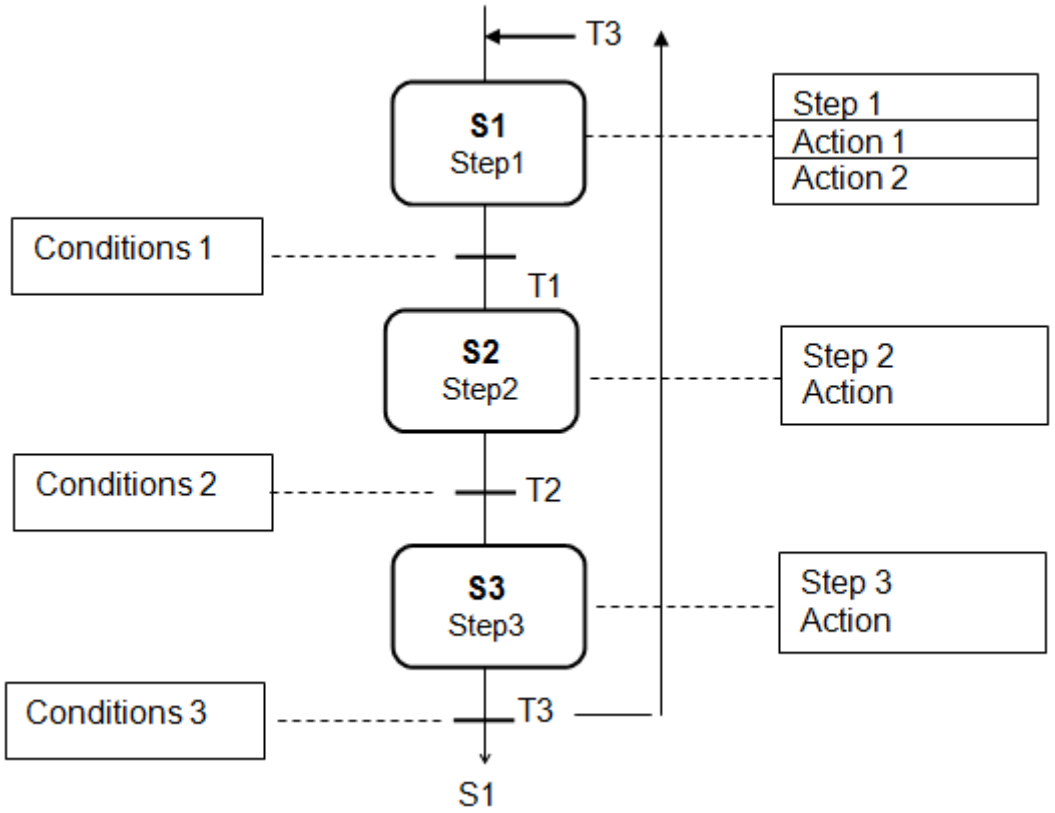


Fig. 7: S7-GRAPH program structure with a jump

5 CREATING PROGRAMS IN THE SIMATIC MANAGER

The SIMATIC Manager (fig. 8) is a software tool that works with the objects in the STEP7 environment. These logical objects correspond to real objects in the technical equipment of the plant.

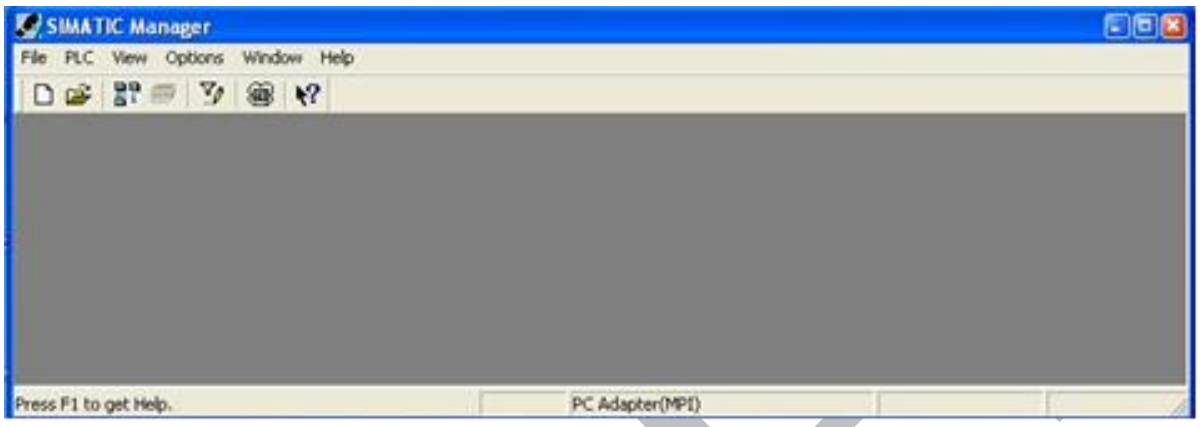


Fig. 8: SIMATIC Manager

A project can contain several stations which are interconnected. One station contains a CPU, which may include a program in STEP7. This program is again a container for other objects, such as the 'Blocks' object, which contains among others the compiled blocks (see fig. 13). The STEP7 objects are connected in a tree structure.

An object is selected by clicking on it. An object is called when you click on the name of the selected object or if you select the highlighted item from the menu 'Edit' > 'Object Properties'. An object is opened with a double click on the object.

It is not possible to give detailed information about the SIMATIC Manager or programming in STEP7 in this instruction manual. The goal is to give all information to be able to create a STEP7 program (S7-Graph language) for the manipulator.

Processing a new project in STEP7 comprises the following steps

- creating a New Project,
- configuring the hardware (SIMATIC station, at least the CPU),
- configuring connections,
- programing (creating symbols, sources, blocks),
- testing/debugging of the program.

5.1 CREATING A PROJECT AND A STEP7 PROGRAM

The STEP7 wizard is recommended to use to create a new project. Select the used CPU and the wizard creates a project with the S7 station, the selected CPU and a STEP7 program container with selected organization blocks.

Start the Project Wizard by 'File' > STEP7 wizard 'New Project'. The wizard has 4 steps (see fig. 9 to fig. 12).

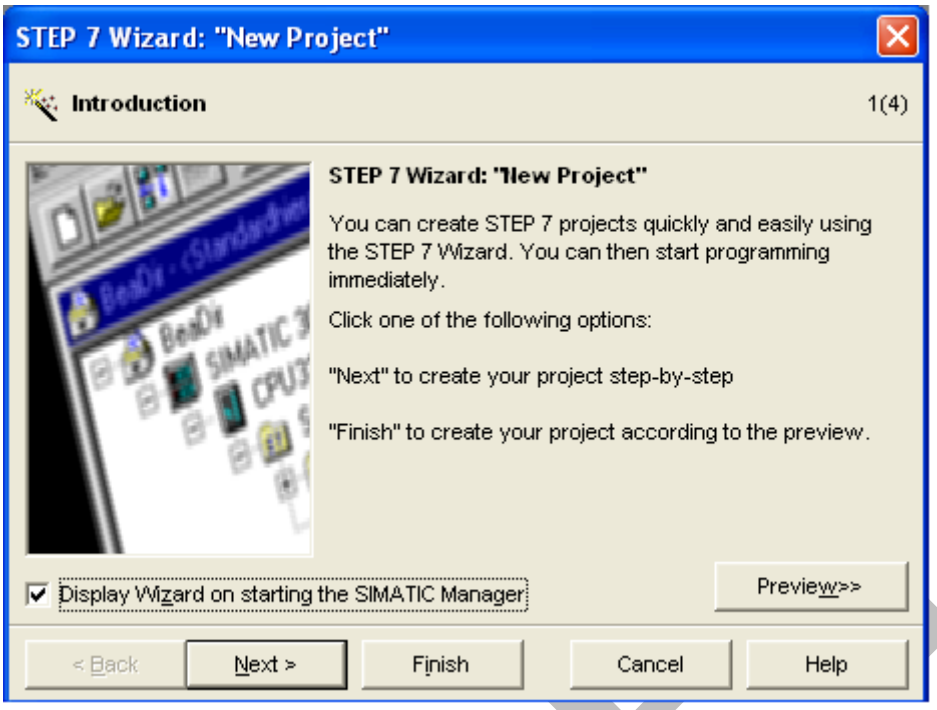


Fig. 9: New project wizard – Step 1

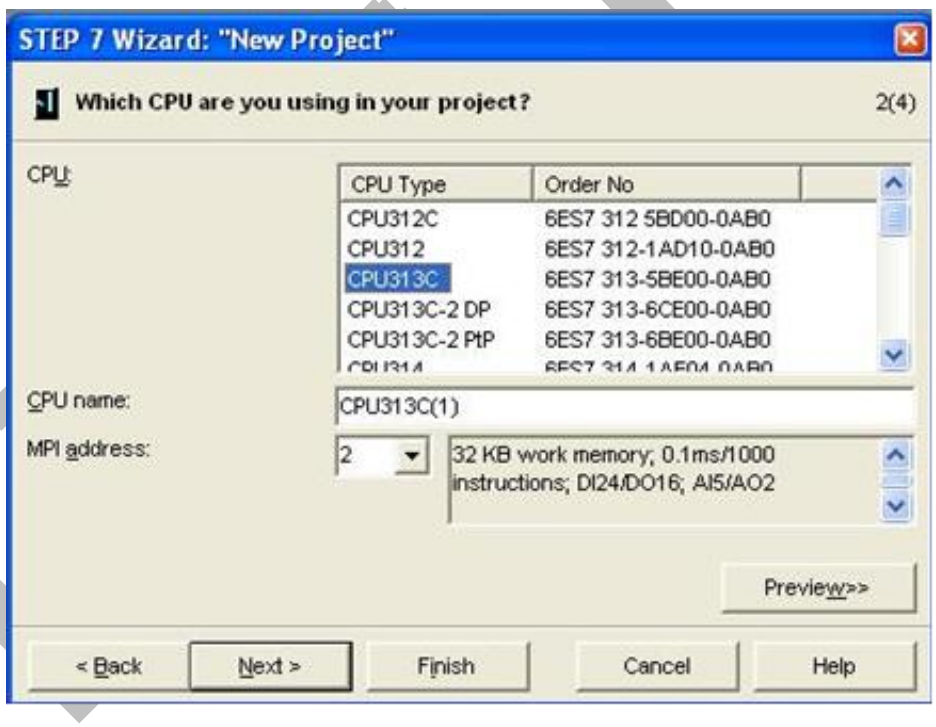


Fig. 10: New project wizard – Step 2

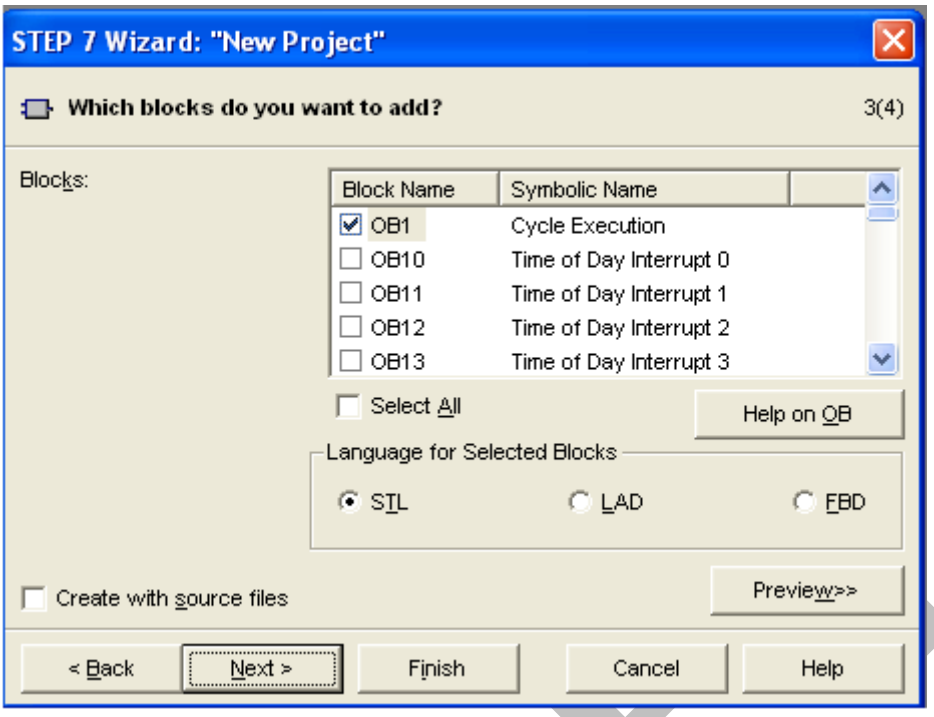


Fig. 11: New project wizard – Step 3

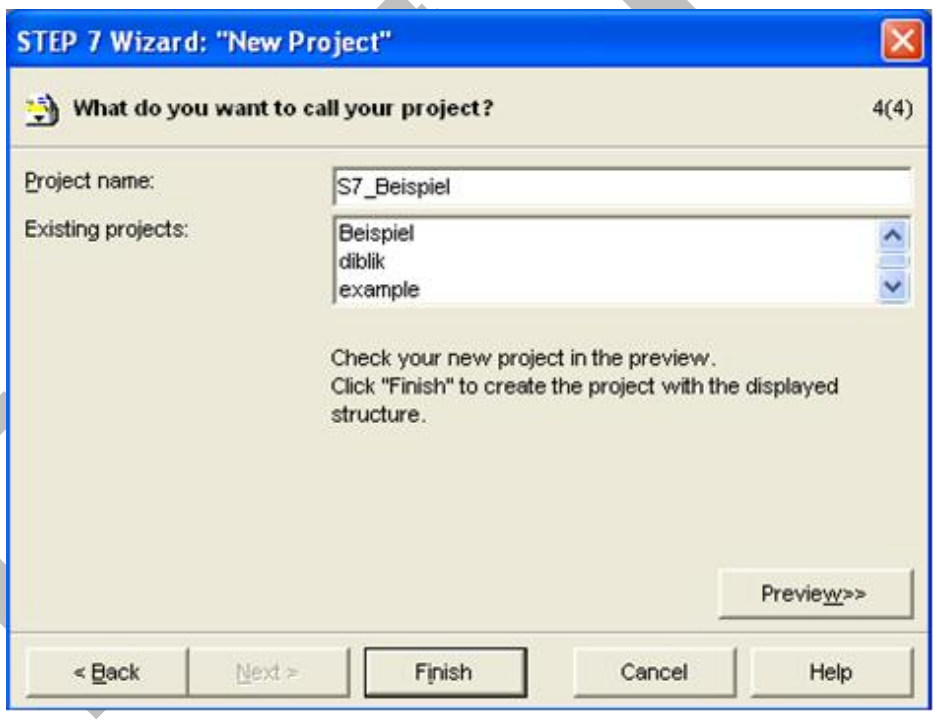


Fig. 12: New project wizard – Step 4

The result of the activities of the STEP7 Wizard is shown in fig. 13. You can see the SIMATIC 300 station, the CPU 313C and the S7 program folder.

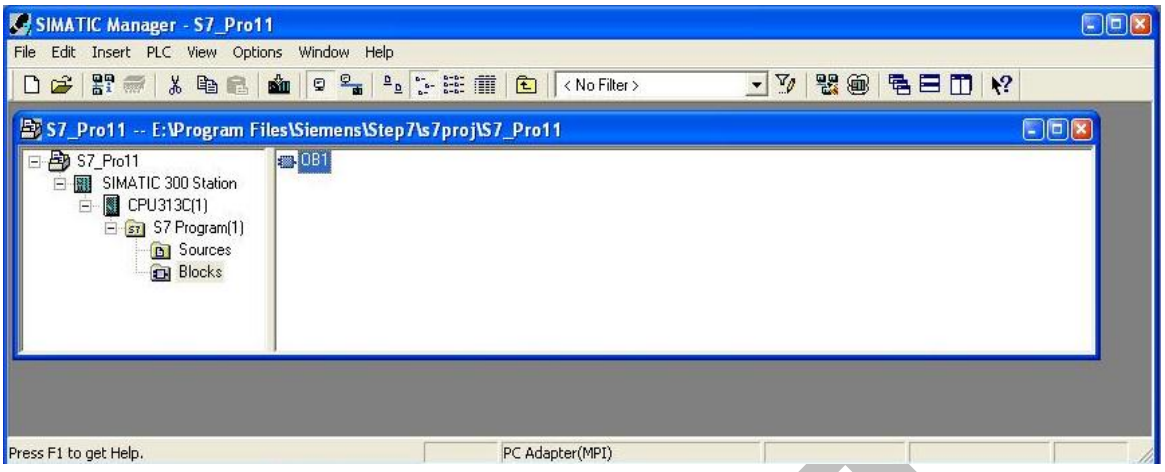


Fig. 13: SIMATIC Manager window with new project

It is now the task to create a sequential program with GRAPH7. The application program is created under the item 'S7 Program'. This object can be allocated to a CPU within the project structure or it can be created independently from the CPU. It contains the object **Symbols**, the container **Sources** and **Blocks**.

The implementation involves the following steps in the main menu. Click on 'Insert' > 'S7 Block' > 'Function Block' (see fig. 14). This opens the 'Properties' window, which must be completed. Set the right program language to 'GRAPH' (see fig. 15). By clicking 'OK' open the newly created window 'SIMATIC Manager – S7_Beispiel \ S7 Program'. Click on the function block FB1 (fig. 16).

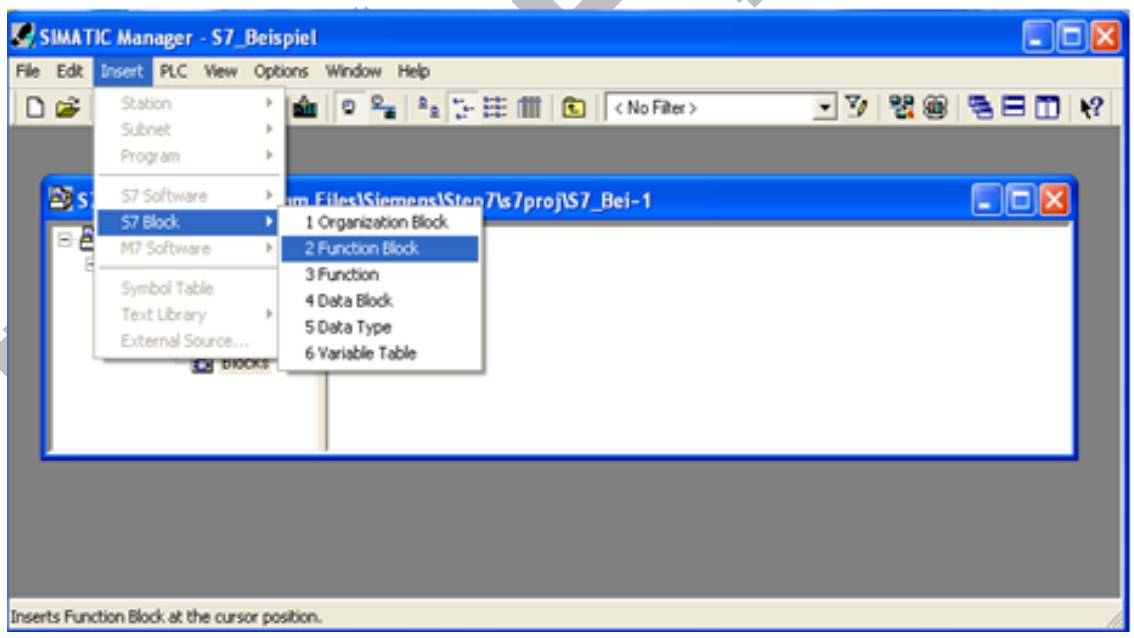


Fig. 14: Insert new function block

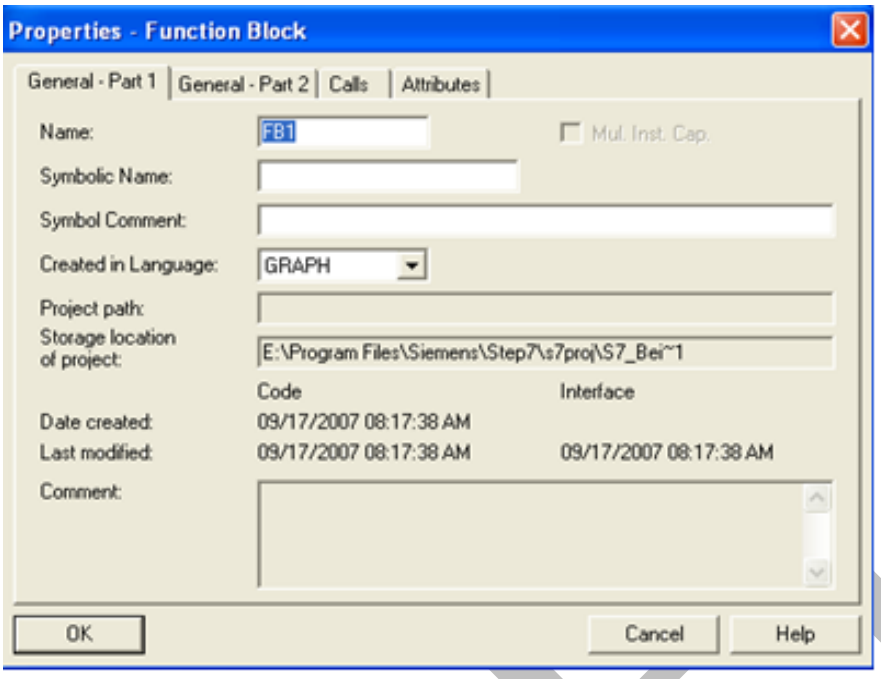


Fig. 15: Function Block Properties window

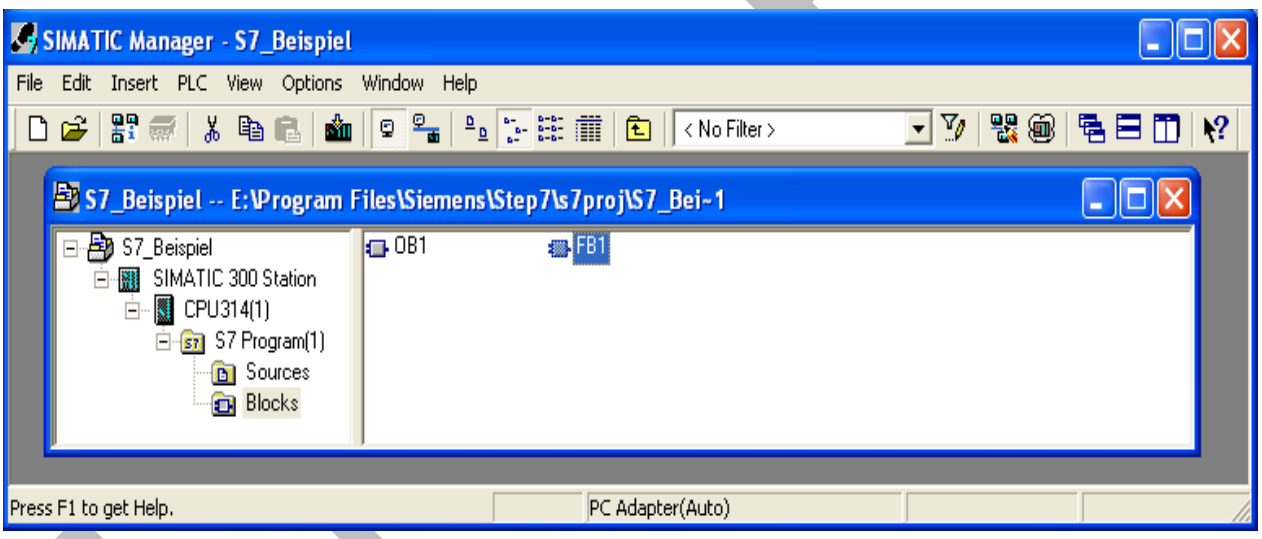


Fig. 16: Project with one function block (FB1)

5.2 CREATING A SYMBOL TABLE

Symbolic addressing in STEP7 was introduced earlier (chapter 3.2). Within the process of creating an S7 program, the SIMATIC Manager also creates an empty table of symbols named 'Symbols'. If you want to work with symbols, then open it and allocate symbols to the absolute addresses. This is carried out with the following steps

1. Open the folder 'S7 Program' by double click (fig. 17).
2. Open the symbol table in the 'S7 Program' by double clicking on 'Symbols' (Fig. 18).
3. Edit the symbols table.
4. Save the symbol table by 'Symbol Table' > 'Save'.

Determine during editing:

- symbolic address/Symbol (variable name, e.g.: 'X0' or 'iArmDown'),
- absolute address (e.g.: I124.0 or E124.0),
- data type (e.g.: BOOL for binary variable),
- comment (optional).

Note! Depending on the language version, the inputs are defined with 'I' or 'E' and the outputs with 'Q' or 'A'!

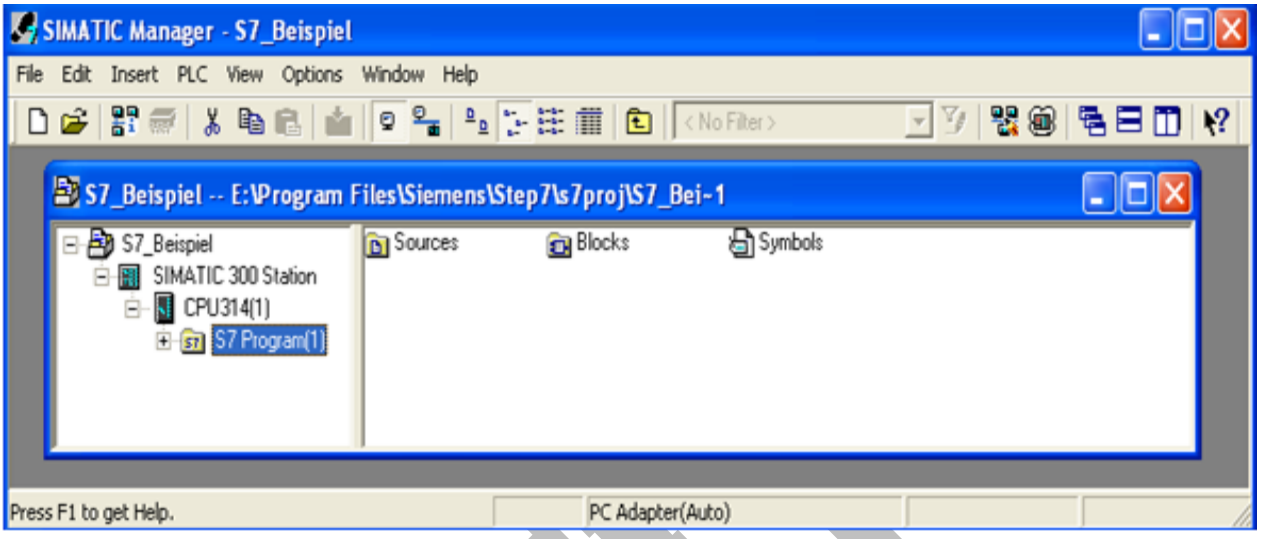


Fig. 17: 'Symbols' library in the SIMATIC Manager

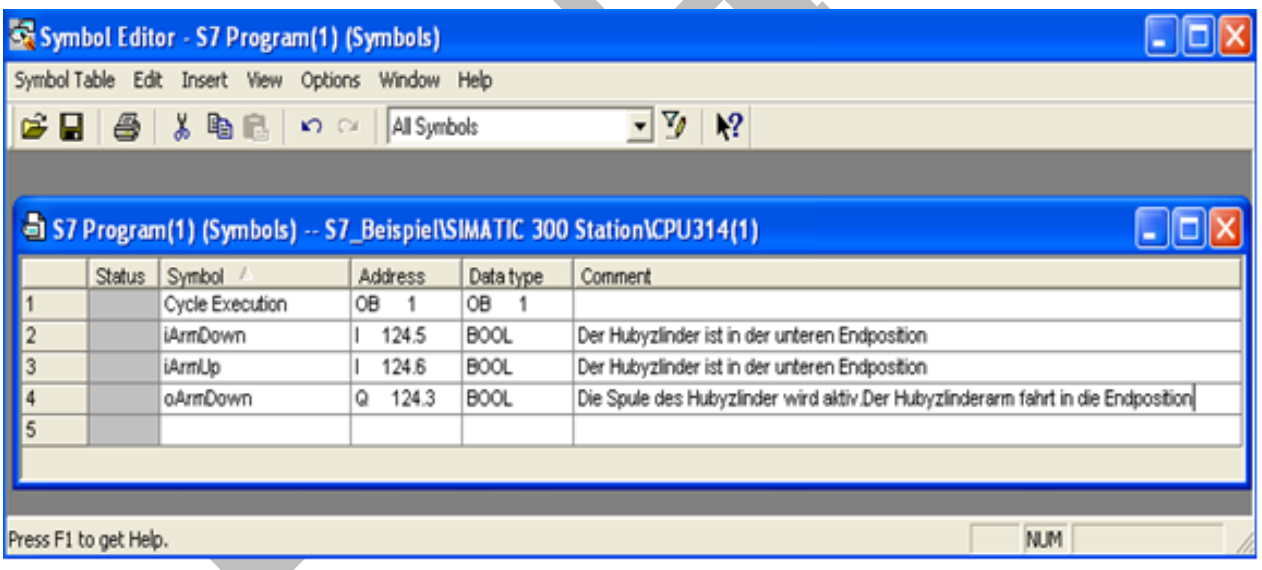


Fig. 18: Symbol Editor window

5.3 CREATING A SEQUENCE CHAIN WITH A S7-GRAPH FUNCTION BLOCK

Starting point is the SIMATIC Manager. By double clicking FB1 the first step 'Step1' and the first transition 'Trans1' are enabled (see fig. 19). The structure can be created both as 'Overview' or in the 'Single sheet representation'. Go to 'View' > 'Display With' > 'Conditions and Actions' to display the conditions and actions which are enabled.

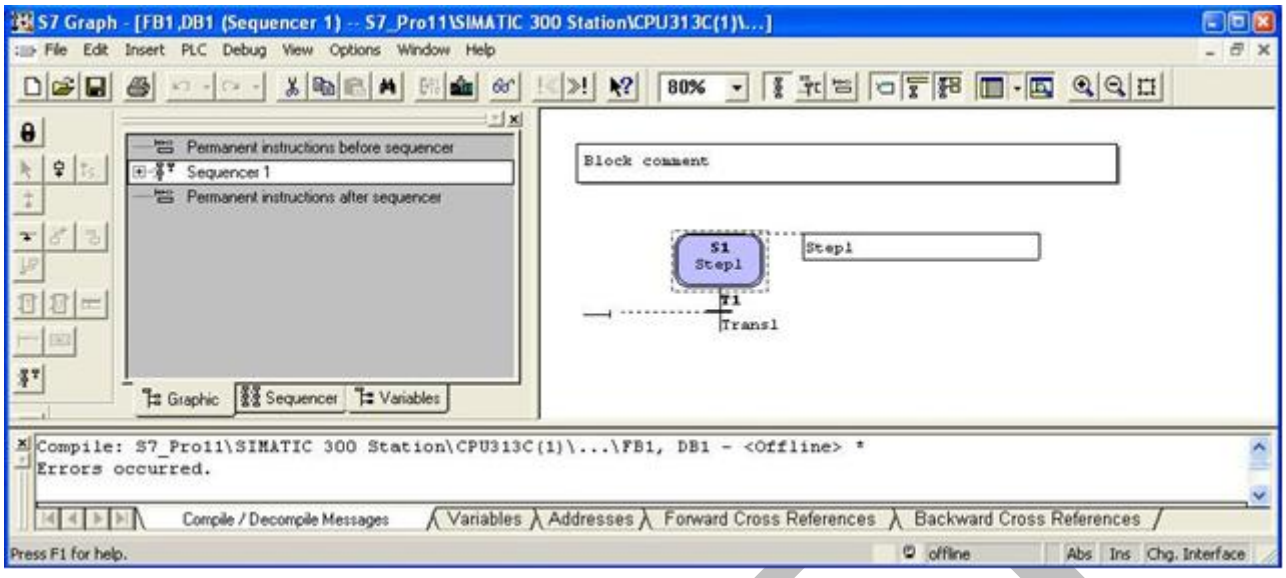
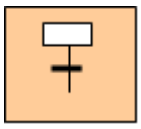


Fig. 19: S7 GRAPH window – start of the new program

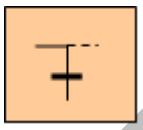
The S7-GRAPH FB editor has a toolbar on the left side with steps, transitions, jumps, alternative branches, etc. You can use the mouse and the toolbar 'Sequence' to move the necessary blocks from the end of the sequence to the beginning. For this purpose, either the 'Direct' mode or the 'Drag and Drop' mode can be used.

5.3.1 'Direct' mode

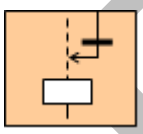
Select the menu command: 'Insert' > 'Direct'



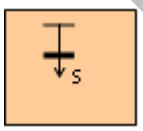
Mark the selected transition and click this icon 'Insert Step + Transition' until you arrive at the appropriate step.



Mark the selected step and click this icon 'Open an alternative branch' to create (open) a new alternative branch.



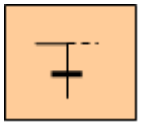
Click this icon 'Close alternative branch' and select the desired transitions (e.g. 'Trans1' as start and 'Trans2' as end) to close the branch.



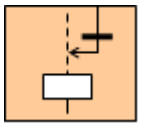
Select this icon 'Insert jump' and mark the position from where the jump starts and then the position where the jump is directed to.

5.3.2 'Drag and Drop' mode

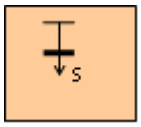
Select the menu command 'Insert' > Drag and Drop'.



Select this icon 'Open an alternative branch' and open the alternative branch by mouse click on the appropriate step.



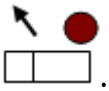
Select this icon 'Close alternative branch'. First select initial the transition (starting point for alternative branch) and then select the final transition (closing point for alternative branch).



Select this icon 'Insert jump' and mark the position from where the jump starts and then the position where the jump is directed to.

5.4 STEP ACTION PROGRAMMING

The following explanations describe the 'Drag and Drop' mode. Use menu instruction 'Insert' > 'Drag-and-Drop'.

1. Select the menu command 'Insert' > Action'. Result is that the mouse pointer takes on the following form  .
2. By clicking on the 'Paste Special box' an empty action line is added.
3. Enter the action.

Alternatively it is possible to use an easier way. Click the right mouse button and realize the following command sequence (from local menu):

'Step 1' > Insert new element' > 'Action'.

An action consists of instruction and an address (operand). The example task requires

- **S** set output (log 1) to operand,
- **R** reset output (log 0) to operand,
- **N** non holding (log 1 as long as the step is active),
- **D** delay (the operand is set for a defined time and then reset).

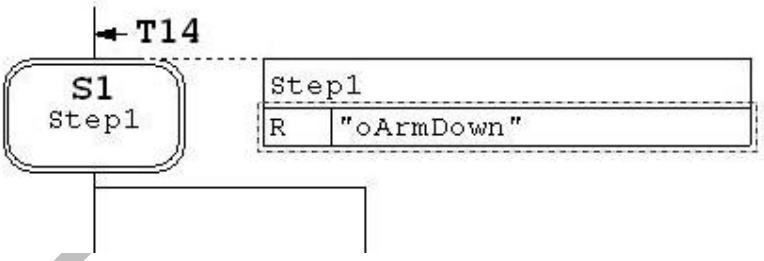
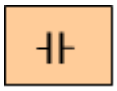


Fig. 20: Step action

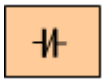
5.5 TRANSITION PROGRAMMING

The bit logic **NO** (normally open contact), **NC** (normally closed contact) and **Comparator** are used for the conditions in the transitions. To program transitions proceed as follows

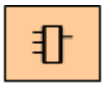
1. Adjust the view 'LAD' and select the appropriate icons in the toolbar 'LAD / FBD'.



Insert normally open contact (NO).



Insert normally closed contact (NC).



Insert the comparator.

2. Place the symbol to the appropriate location by clicking on the transitions. The insert mode can be terminated at any time by pressing the ENTER button.
3. Enter the operands (addresses). Select the corresponding text box with a mouse click on its spacers '??.?'. Then write an absolute or symbolic address (e.g. 'I125.0' or 'iArmChute').
4. You can add a comment to the sequence chain. You find the comment in each display top left and it can be opened with a mouse click.

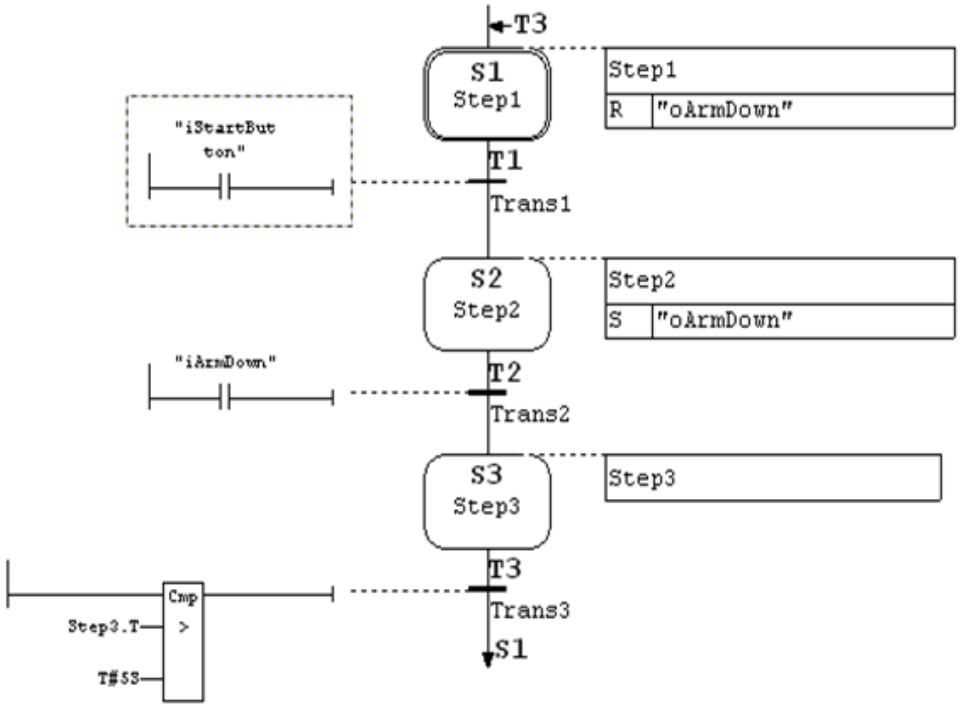


Fig. 21: Transition programming

5.6 SAVE AND CLOSE THE SEQUENCER

After completion of the sequence chain, it is saved with File' > 'Save'. Here the compilation is done automatically. Result is that the 'Select Instance DB' dialog box with the default instance DB1 is opened. The setting is accepted by clicking 'OK'. The sequencer is closed with 'File' > 'Close'.

5.7 EMBEDDING THE SEQUENCE CHAIN INTO THE PROJECT

The program is called and started from the organization block (OB). In general, you can create OB1 in LAD (Ladder Diagram), FBD (Function Block Diagram) or STL/SCL (Instruction List). The whole procedure is not described here, only the process of creation of the main command (to allow calling of the function and running the program) is presented.

1. Open the folder 'Block' in the SIMATIC Manager.
2. Start the editor for LAD/STL/FBD by double click on OB1.

3. Select the STL in the 'View' menu.
4. Enter the command '**CALL FB1, DB1**' (see fig. 22).
5. Finally you can finish programming the OB1 with 'File' > 'Save' and 'File' > 'Close'.

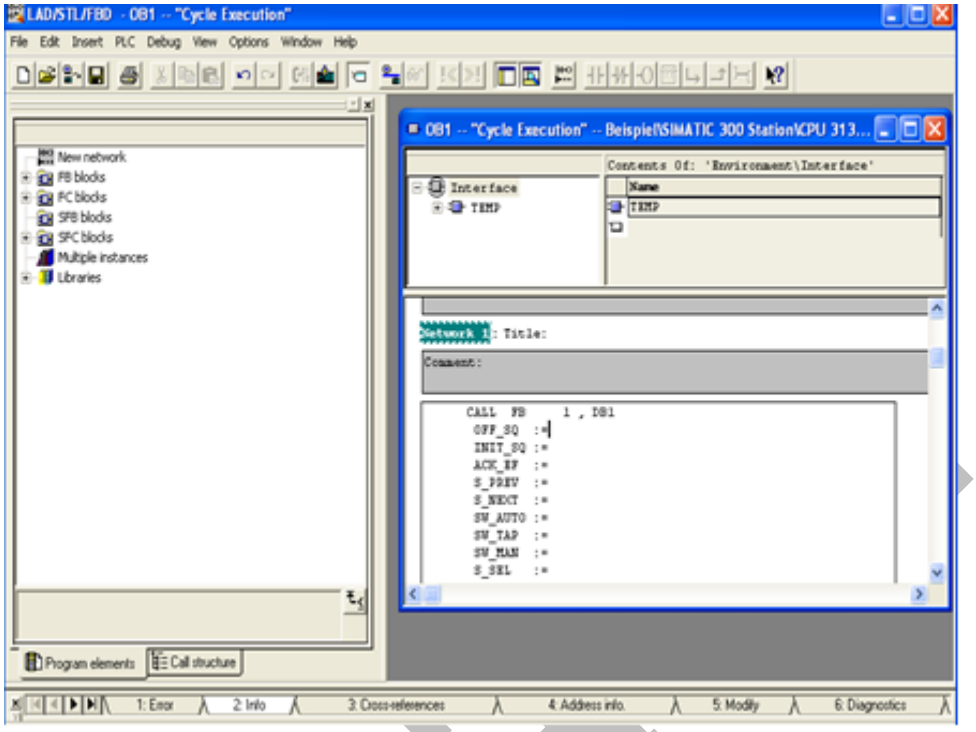


Fig. 22: Embedding the calling of the created function block from main cycle OB1

5.8 SUPPLEMENT TO THE CPU DESCRIPTION

The page 'Which CPU are you testing in your Project' in the STEP 7 Wizard does not include the used CPU 313C_V2 (new version). Therefore it still must be set correctly before loading into the automation system.

1. Open the SIMATIC Manager program and select 'SIMATIC 300 Station' (fig. 23).
2. Double click the library icon 'Hardware' (fig. 23).
3. Select the CPU 'CPU 313C/V2.0' (fig. 24).

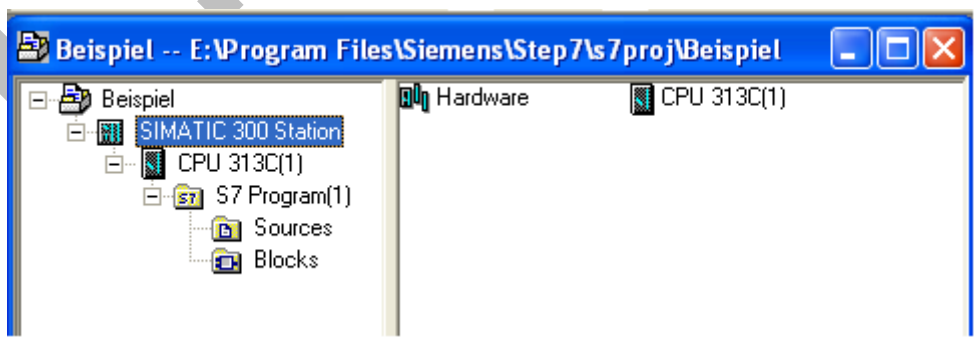


Fig. 23: Selecting the hardware

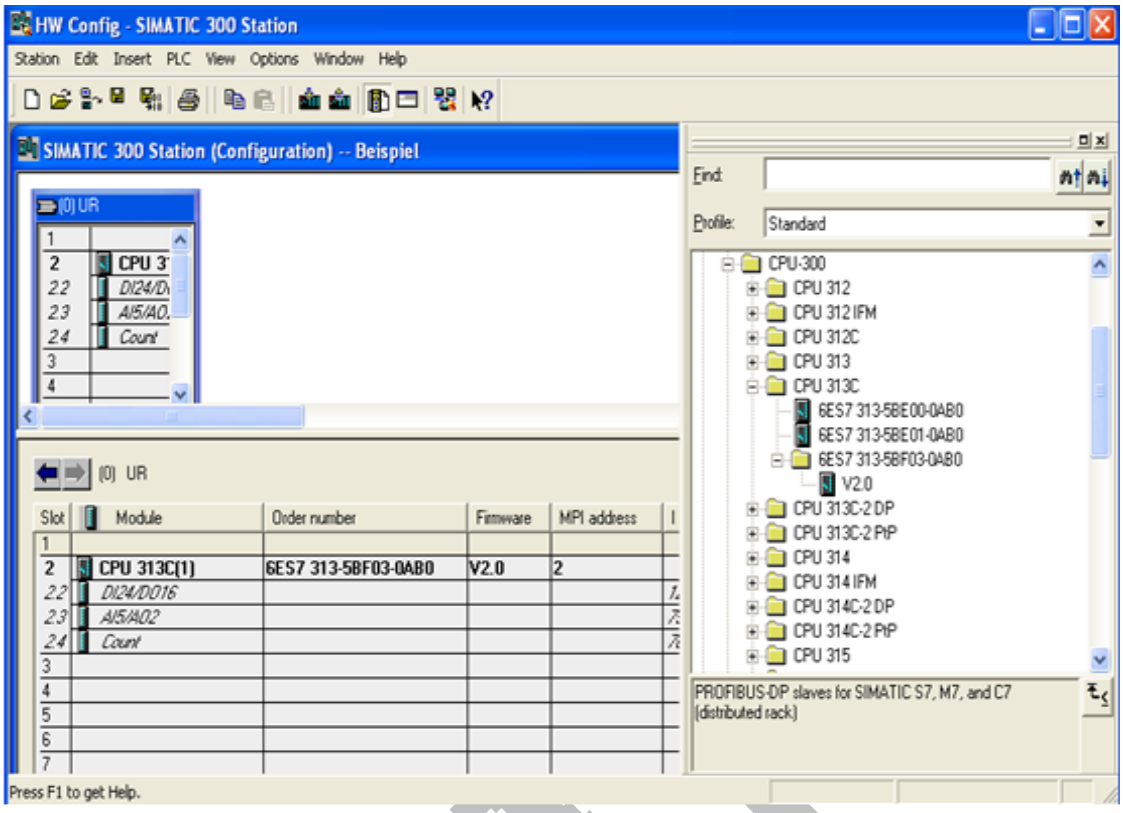


Fig. 24: Selecting the right type of PLC's CPU

5.9 DOWNLOADING THE PROGRAM TO CPU

To test the program and control the technical system it must be downloaded from the computer to the PLC. Follow these steps.

1. Open the SIMATIC Manager and select the folder with your project.
2. Select 'PLC' > 'Download'.

Note: The download of the S7-GRAPH blocks is best done in STOP mode.

5.10 TESTING THE PROGRAM

The online connection has to be prepared to test the user program. Follow these steps:

1. Open the project window in the SIMATIC Manager.
2. Open the sequencer by double click to your function block (FB1).
3. Open the SIMATIC Manager and select the folder 'Block'.
4. Select the menu command 'Debug' > 'Monitor'.

5.11 CHANGING THE PROGRAM

If there are unexpected results due to tests, carry out changes by this way:

1. After double click to the error you to get right into the 'Step'.
2. End the online monitor status with 'Debug' > 'Monitor'.
3. Make necessary changes.
4. Save the sequencer and end with the menu commands: 'File' > 'Save' and 'File' > 'Close'.

6 EXAMPLE

As an example of creating a sequence chain is shown the 'Variant 1' from the next chapter. Create a program for this task as a sequence chain and verify the correctness of the program on the manipulator!

6.1 TASK DESCRIPTION

The lifting cylinder of the handling device should be driven from the upper end position to the lower end position (removal position) by short push of the start button. After reaching the lower end position it shall be reset back to the upper end position after 5 seconds. This cycle shall be repeated only after pushing the start button. The simplest sequence chain is to design!

For this example, only the arm is moved by the double acting pneumatic cylinder in the vertical direction and the start button T1 is used. The upper and lower end position can be detected by the micro-switches S6 and S7. To start the switch T1 will be applied.

6.2 GENERAL CONDITIONS

The cylinder has two defined end positions:

1. The lifting cylinder is at the upper end position. This state is messaged through sensor S7 = log 1. If the lifting cylinder is not in the upper end position the sensor S7 = log 0.
2. The lifting cylinder is in the lower end position. This state is messaged through sensor S6 = log 1. If the lifting cylinder is not in the lower end position the sensor S6 = log 0.

6.3 DIVIDING THE TASK INTO STEPS

1. Step 1: The cylinder is at the upper end position.
   Transition1: Reset button is pressed.
2. Step 2: The cylinder is going to the lower end position.
   Transition2: The lifting cylinder is at the lower end position.
3. Step 3: The cylinder is waiting at the lower end position for 5 seconds.
   Transition3: The waiting time of 5 seconds has elapsed.

If you want to repeat the cycle continuously then add a jump from the end to the Step1.

6.4 INPUT AND OUTPUT DEFINITION

Now, as it was described, it is necessary to define the corresponding input and output signals. The basis for this is the technological scheme (fig. 2) and the electrical wiring of the sensor to the terminals of the input and output modules. The inputs and outputs are given by absolute addresses, which are defined according to the existing wiring. For the names of the variables English abbreviations were chosen.

There are three inputs (tab. 5)

- lifting cylinder is at the upper end position — sensor S7,
- lifting cylinder is at the lower end position — sensor S6,
- start button is pressed — switch T1

and one output (tab. 6)

- the coil of the lifting cylinder is getting active sensor Y1.

Tab. 5: Inputs on the example

Label	Description of the signal	PLC address	Name of variable
S6	Arm is down *log 1 = lifting cylinder is at the lower end position* *log 0 = lifting cylinder is not at the lower end position*	I124.5	iArmDown
S7	Arm is up *log 1 = lifting cylinder is at the upper end position* *log 0 = lifting cylinder is not at the upper end position*	I124.6	iArmUp
T1	Start button *log 1 = start button is pressed* *log 0 = start button is not pressed*	I125.6	iStartButton

Tab. 6: Output signals in the example

Label	Description of the signal	PLC address	Name of variable
Y1	Move arm down *log 1 = the coil of the lift cylinder is activated, the arm of the lifting cylinder moves into the lower end position where stops* *log 0 = the coil of the lifting cylinder is switched off, the arm of the lifting cylinder drives due to the spring force in the upper end position where it stops*	Q124.3	oArmDown

6.5 CREATING THE SEQUENCER PROGRAM

The sequencer is carried out as it was described. The sequencer is shown in fig. 25.

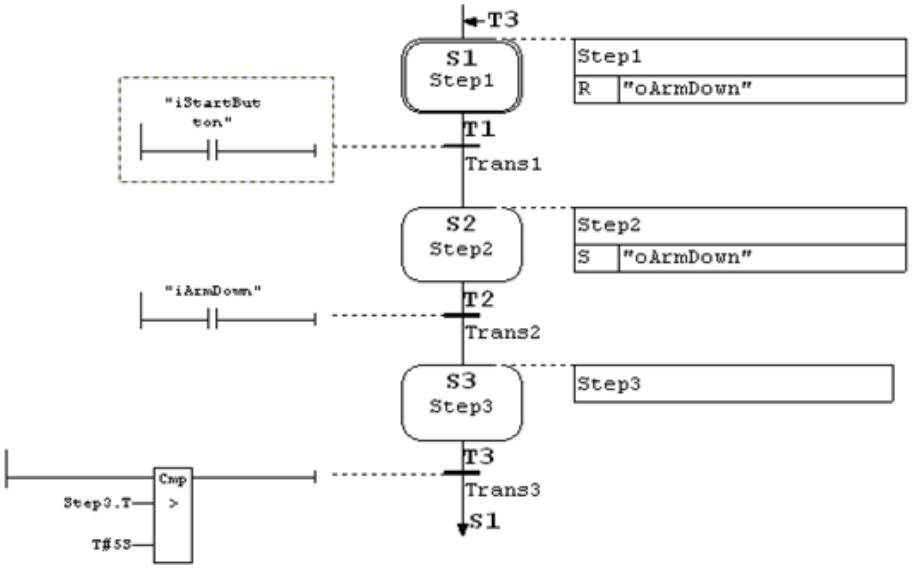


Fig. 25: Sequencer program of the example

Note: The waiting time is realized with a comparator, where the elapsed time of the 'Step3' is compared with the given waiting time of 5 s ('T#5S').

7 EXPERIMENTAL TASKS

The whole task (final operation) shall be solved in six steps. Each step may correspond to a partial program that is referred as a separate variant. Consider carefully the entire structure of the program. It is recommended to keep the names of the variables in English.

7.1 TASK 'PREPARATION' – VARIANT 0

- The states of the output signals Y2/Y3 (set sucker on/off), Y5/Y6 (rotate arm to right/left) and Y2.1/Y2.2 are not completely defined. Test the function of all possible signal state combinations and determine which ones are allowed!
- Examine the behavior of the signal S0.9 (new work piece arrives to dock).

7.2 TASK 'CYLINDER OPERATION' – VARIANT 1

- After pushing the start button the lifting cylinder of the handling device should be driven from the upper end position to the lower end position. After reaching the lower end position, the cylinder shall be returned to the upper end position after 5 seconds.
- This cycle is to be repeated by pushing the start button.

Do not forget to test the initial position of the cylinder! If the initial position is not the upper end position, firstly transfer the cylinder to the upper end position!

7.3 TASK 'VACUUM TESTING' – VARIANT 2

- The cylindrical work piece (barrel) is manually loaded into the removal position!
- After pushing the start button the lifting cylinder of the handling device should be driven from the upper end position to the lower end position only in the case the work piece is present at the dock. If the dock is free (work piece is not served at the dock), wait until the work piece is served and light on the white lamp. The white lamp is lighted off after the work piece is served.
- 1 second after reaching the lower end position, the work piece should be grasped and held with the suction cup (vacuum on).
- After 1 second, the cylinder with the work piece should be returned to the upper end position.
- The work piece should be kept for 2 seconds in the upper end position.
- After these 2 seconds the suction cup should release the load (vacuum off).
- This cycle is to be repeated by pushing the start button.

Do not forget to test the initial position of the cylinder and the initial function of the vacuum! If the initial position is not the upper end position, firstly transfer the cylinder to upper end position!

7.4 TASK 'ROTATION' – VARIANT 3

- If the work piece is at the dock, the lifting cylinder of the handling device should be driven from the upper end position to the lower end position after pushing the start button. If the dock is free (work piece is not served at the dock), wait until the work piece is served and light on the white lamp. The white lamp is lighted off after the work piece is served.
- 1 second after reaching the lower end position, the work piece should be grasped and held with the suction cup.

- After 1 second, the cylinder with the work piece should be returned to the upper end position.
- In the upper end position, the cylinder should rotate with the work piece into the output position 2 (chute position).
- 2 seconds after the output position 2 is reached the suction cup should drop the work piece into the chute.
- Then the lifting cylinder should turn back to the initial position.
- This cycle is to be repeated by pushing the start button.

Again do not forget to test the initial position of the cylinder and initial function of the vacuum!

7.5 TASK 'MAGAZINE' – VARIANT 4

- Variant 4 corresponds to the variant 3 only with the difference that the work piece is automatically served from the prepared stack (do not move the magazine).

7.6 TASK 'FULL OPERATION' – VARIANT 5

- Variant 5 corresponds to the variant 4 only with the difference that after the first stack is emptied, the second stack is used.
- After the first pushing of the start button start the operation and continue automatically in the never ending loop until the start button is pressed.
- In the case of any error or if both stacks are empty light on the white lamp.

8 LITERATURE

[1] Borlebach, K.H.; Kraemer, G; Mock, W. und Nows, E.: Steuerungstechnik mit speicherprogrammierbaren Steuerungen SPS, Verlag Europa-Lehrmittel, 5657 HaanGruiten, 1992, ISBN :3-8085-3144-4

[2] Blau, R.; Proske, D; Werner, F.: Regelungstechnik Binäre Steuerungen Teil 1: Kombinatorische Systeme, Technische Fachhochschule Berlin, Fernstudium, 1993.

[3] Blau, R.; Proske, D; Werner, F.: Regelungstechnik Binäre Steuerungen Teil 1: Sequentielle Systeme, Technische Fachhochschule Berlin, Fernstudium, 1993.

[4] Proske, D.: Steuerungstechnisch interpretierte Petri-Netze (SIPN), Hochschule Zittau/Görlitz (FH), FB Elektro-und Informationstechnik, Steuerungstechnik Lehrbrief ST-3, Ausgabe November 2003.

[5] Berger, H.: Automatisieren mit SIMATIC S5-115U, Berlin, München: Siemens - Aktiengesellschaft. 1991.